

TOSHIBA

東芝 オリジナル 32 ビット RISC マイクロプロセッサ

ARM Core Family

TMPA901CMXBG

Not Recommended
for New Design

株式会社 **東芝** セミコンダクター社

お客様各位

2021-9-1

東芝デバイス&ストレージ株式会社
東芝デバイスソリューション株式会社

〒212-8520 神奈川県川崎市幸区堀川町 580-1

Tel: 044-548-2200

Fax: 044-548-8965

非同期シリアル通信機能に関する誤記について

平素より東芝マイクロコントローラーをご使用頂き、誠にありがとうございます。

弊社マイコンに内蔵されております非同期シリアル通信機能(UART、または FUART)、50%デューティモード付き非同期シリアル通信回路(UART)の送信割り込み発生タイミングで、データシート、リファレンスマニュアルの記載に誤記が発見されました。

大変ご迷惑をおかけ致しますが、本文章をご確認頂きますようお願い申し上げます。

本件のご不明な点につきましては、弊社営業担当までお問い合わせいただきますようお願い申し上げます。

—記—

1. 対象製品

TMPM342FYXBG TMPM343F10XBG TMPM343FDXBG TMPM366F20AFG TMPM366FWFG TMPM366FYFG TMPM366FDFG TMPM366FWXBG TMPM366FYXBG TMPM366FDXBG TMPM367FDFG TMPM367FDXBG TMPM368FDFG TMPM368FDXBG TMPM369FDFG TMPM369FDXBG TMPM36BF10FG TMPM36BFYFG TMPM381FWDFG TMPM381FWFG TMPM383FSEFG TMPM383FSUG TMPM383FWEFG TMPM383FWUG TMPM3V4FSEFG TMPM3V4FSUG TMPM3V4FWEFG TMPM3V4FWUG TMPM3V6FWDFG TMPM3V6FWFG	TMPM440FEXBG TMPM440F10XBG TMPM461F10FG TMPM461F15FG TMPM462F10FG TMPM462F15FG TMPM46BF10FG TMPM4G6FDFG TMPM4G6FEFG TMPM4G6F10FG TMPM4G7FDFG TMPM4G7FEFG TMPM4G7F10FG TMPM4G8FDFG TMPM4G8FDXBG TMPM4G8FEFG TMPM4G8FEXBG TMPM4G8F10FG TMPM4G8F10XBG TMPM4G8F15FG TMPM4G8F15XBG TMPM4G9FDFG TMPM4G9FDXBG TMPM4G9FEFG TMPM4G9FEXBG TMPM4G9F10FG TMPM4G9F10XBG TMPM4G9F15FG TMPM4G9F15XBG	TMPA900CMXBG TMPA901CMXBG TMPA910CRAXBG TMPA910CRBXXBG TMPA911CRXBG TMPA912CMXBG TMPA913CHXBG
---	---	---

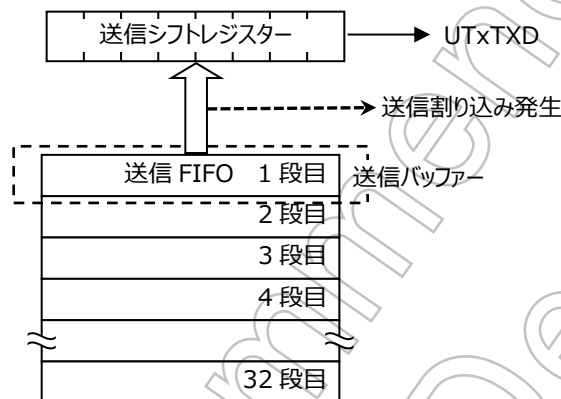
2. 詳細

送信割り込みの発生タイミングは以下となります。

なお、送信割り込み発生タイミング誤記につきましては"送信 FIFO 未使用時"に限られ、"送信 FIFO 使用時"ではデータシートの記載のとおりとなります。

2.1. 送信 FIFO 未使用時

送信バッファ(送信 FIFO 1 段目)から送信シフトレジスタにデータが転送されたときに(送信バッファに空きができたとき)送信割り込みが発生します。



2.1.1. 送信割り込み発生タイミング

送信 FIFO 未使用時の送信割り込みは、次データに対する送信バッファへの書き込みタイミングを通知するため、送信バッファが空になったタイミングで発生します。送信割り込みは送信バッファに次のデータが書き込まれると自動的にクリアされるため、連続的にデータを送信し続ける場合はソフトウェアによる送信割り込みのクリアは必要ありません(UARTxICR<TXIC> = "1"設定)。

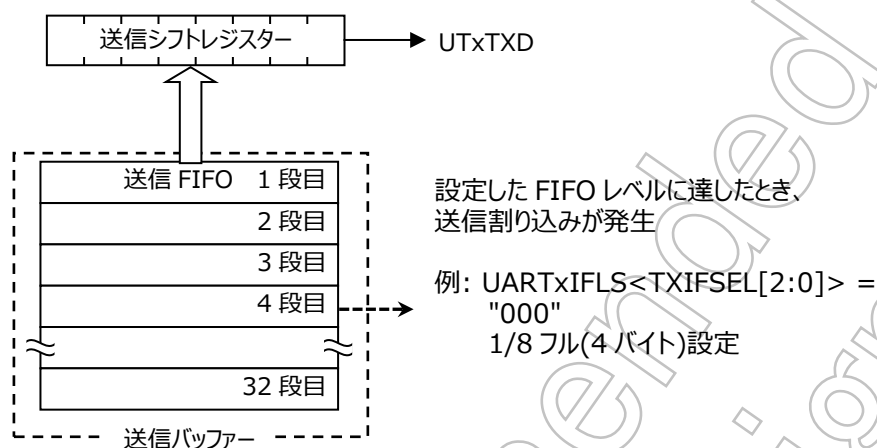
また、送信を終了する場合は、最終送信データがシフトレジスタに転送され、送信バッファが空になった際に最後の送信割り込みが発生します。送信バッファに次のデータを書き込まない場合は、割り込みハンドラー内でソフトウェアによる割り込みクリア(UARTxICR<TXIC> = "1"設定)を実行することで送信割り込みを意図的にクリアすることができます。

なお、データ送信中にソフトウェアで送信割り込みクリア(UARTxICR<TXIC> = "1"設定)を実行した場合、送信完了時の STOP ビット発生と同時のタイミングで送信バッファにデータの書き込みを行うと、送信割り込みは発生しません。確実に送信割り込みを発生させる場合は、データ送信中にソフトウェアで送信割り込みをクリアしないで送信バッファにデータを書き込むか、送信が停止している状態(UARTxFR<BUSY> = "0"のとき)で送信バッファにデータを書き込んでください。

連続してデータを送信する場合は、次項の送信 FIFO を利用したデータ転送を推奨致します。

2.2. 送信 FIFO 使用時

送信動作により送信 FIFO の格納段数が $\text{UARTxIFLS} < \text{TXIFSEL}[2:0] >$ であらかじめ設定した FIFO レベルに達すると送信割り込みが発生します。



2.2.1. 送信割り込み発生タイミング

送信 FIFO 使用時は、設定した FIFO レベルに達したときに送信割り込み発生します。

例えば、 $\text{UARTxIFLS} < \text{TXIFSEL}[2:0] > = \text{"000"}$ (1/8 フル 4 バイト設定) の場合、送信 FIFO に格納されたデータが 4 段目に達したときに送信割り込みが発生します。

送信割り込みは設定した FIFO レベルを超えるデータが送信 FIFO に格納されるとクリアされ、設定した FIFO レベルに達すると再度発生します。

3. 誤記内容

製品により送信割り込み発生タイミングの記載内容が異なり、各製品に対する誤記掲載箇所の章番号を以下表に示します。なお、送信割り込み発生タイミング誤記につきましては"送信 FIFO 未使用時"に限られ、"送信 FIFO 使用時"ではデータシートの記載のとおりとなります。

誤記に対する修正、追記内容は、以降の「4. 誤記修正・追記内容」で説明し、全ての対象製品で共通の記載内容となります。

3.1. 記載タイプ A

3.1.1. 対象製品と対象箇所の章番号

製品名	記載箇所章番号
TMPM342FYXBG	16.4.7
TMPM366F20AFG(注)	15.4.7
TMPM366FWFG、TMPM366FYFG、TMPM366FDFG、TMPM366FWXBG、 TMPM366FYXBG、TMPM366FDXBG	16.4.7
TMPM367FDFG、TMPM367FDXBG、TMPM368FDFG、TMPM368FDXBG、 TMPM369FDFG、TMPM369FDXBG	13.4.7
TMPM36BFYFG、TMPM36BF10FG	13.4.7
TMPA900CMXBG、TMPA901CMXBG、TMPA910CRAXBG、TMPA910CRBXBG、 TMPA911CRXBG、TMPA912CMXBG、TMPA913CHXBG	3.13.1.1 (7)

注) 非同期シリアル通信機能(UART)章です。

タイプ A

修正が必要な記載箇所(赤枠)

割り込み種類	割り込み発生タイミング
オーバーランエラー発生	過剰データの STOP ビット受信後
ブレークエラー 割り込み	STOP ビット受信後
パリティエラー 発生	パリティデータ受信後
フレーミングエラー発生	フレームオーバーとなる Bit データを受信した後
受信タイムアウト割り込み	受信 FIFO にデータを取り込んでから、Baud16 の 511 クロック後
送信割り込み	最終 DATA (MSB データ)を送信した後
受信割り込み	STOP ビット受信後

3.2. 記載タイプ B(1)

3.2.1. 対象製品と対象箇所の章番号

製品名	記載箇所章番号
TMPM461F10FG、TMPM461F15FG、TMPM462F10FG、TMPM462F15FG	14.4.6.2

タイプ B(1)

修正が必要な記載箇所(赤枠)

割り込み種類	割り込み発生タイミング
オーバランエラー割り込み	FIFO がフルになったときの STOP ビット受信後
ブレークエラー割り込み	STOP ビット受信後
パリティエラー割り込み	パリティデータ受信後
フレーミングエラー割り込み	フレームオーバーとなる Bit データを受信した後
受信タイムアウト割り込み	受信 FIFO にデータを取り込んでから、Baud16 の 511 クロック後
送信割り込み	FIFO 未使用時: 送信許可後、1 バイト目は START ビット送信開始時と STOP ビット送信開始時、2 バイト目以降は、STOP ビット送信開始時(それぞれの割り込み発生でデータ書き込みを行い割り込みクリアした場合)
	FIFO 使用時: STOP ビット送信開始時(MSB データ転送後)に FIFO 内が設定された FIFO レベルのデータ数となったとき
受信割り込み	FIFO 未使用時: STOP ビット受信後
	FIFO 使用時: 設定した FIFO がフルとなるまでの STOP ビット受信後

3.3. 記載タイプ B(2)

3.3.1. 対象製品と対象箇所の章番号

製品名	記載箇所章番号
TMPM343FDXBG、TMPM343F10XBG、TMPM366F20AFG(注)	16.4.6.2
TMPM381FWFG、TMPM381FWDFG、 TMPM383FSUG、TMPM383FSEFG、TMPM383FWUG、TMPM383FWEFG、 TMPM3V4FSUG、TMPM3V4FSEFG、TMPM3V4FWUG、TMPM3V4FWEFG、 TMPM3V6FWFG、TMPM3V6FWDFG	11.4.6.2
TMPM440FEXBG、TMPM440F10XBG	26.4.6.2

注) 50%デューティモード付き非同期シリアル通信回路(UART)章です。

タイプ B(2)

修正が必要な記載箇所(赤字)

割り込み種類	割り込み発生タイミング
オーバランエラー発生	FIFO がフルになった時の STOP ビット受信後
ブレークエラー 割り込み	STOP ビット受信後
パリティエラー 発生	パリティデータ受信後
フレーミングエラー発生	フレームオーバーとなる Bit データを受信した後
受信タイムアウト割り込み	受信 FIFO にデータを取り込んでから、Baud16 の 511 クロック後
送信割り込み	FIFO 未使用時: 送信許可設定後、1 バイト目は START ビット送信開始時と STOP ビット送信開始時、2 バイト目以降は、STOP ビット送信開始時 (それぞれの割り込み発生でデータ書き込みを行い割り込みクリアした場合)
	FIFO 使用時: STOP ビット送信開始時(MSB データ転送後)に FIFO 内に設定された FIFO レベルのデータ数となった時
受信割り込み	FIFO 未使用時: STOP ビット受信後
	FIFO 使用時: 設定した FIFO レベルがフルとなるデータの STOP ビット受信後

3.4. 記載タイプ B(3)

3.4.1. 対象製品と対象箇所の章番号

製品名	記載箇所章番号
TMPM4G6FDFG、TMPM4G6FEFG、TMPM4G6F10FG、TMPM4G7FDFG、 TMPM4G7FEFG、TMPM4G7F10FG、TMPM4G8FDFG、TMPM4G8FDXBG、 TMPM4G8FEFG、TMPM4G8FEXBG、TMPM4G8F10FG、TMPM4G8F10XBG、 TMPM4G8F15FG、TMPM4G8F15XBG、TMPM4G9FDFG、TMPM4G9FDXBG、 TMPM4G9FEFG、TMPM4G9FEXBG、TMPM4G9F10FG、TMPM4G9F10XBG、 TMPM4G9F15FG、TMPM4G9F15XB	リファレンスマニュアル(注) 高精度非同期シリアル通信 回路(FUART-B) 3.8.2

注) 本文章の UARTxIFLS を **[FURTxIFLS]** に、UARTxICR を **[FURTxICR]** に、UARTxFR を **[FURTxFR]** に読み替えてください。

タイプ B(3)

修正が必要な記載箇所(赤枠)

割り込み種類	割り込み発生タイミング
オーバランエラー発生	FIFO がフルになったときの STOP ビット受信後
ブレークエラー割り込み	STOP ビット受信後
パリティエラー発生	パリティデータ受信後
フレーミングエラー発生	フレームオーバとなる Bit データを受信した後
受信タイムアウト割り込み	受信 FIFO にデータを取り込んでから、転送クロックの 511 クロック後
送信割り込み	1 バイト保持レジスタ時(FIFO 未使用時) 送信許可設定後、1 バイト目は START ビット送信開始時と STOP ビット送信開始時、2 バイト目以降は、STOP ビット送信開始時(それぞれの割り込み発生でデータ書き込みを行い割り込みクリアした場合) FIFO 使用時 STOP ビット送信開始時(MSB データ転送後)に FIFO 内が設定された FIFO レベルのデータ数となったとき
受信割り込み	1 バイト保持レジスタ時(FIFO 未使用時) STOP ビット受信後 FIFO 使用時 設定した FIFO レベルがフルとなるデータの STOP ビット受信後

3.5. 記載タイプ C

3.5.1. 対象製品と対象箇所の章番号

製品名	TD 記載箇所章番号
TMPM46BF10FG	19.4.6.2

タイプ C	
修正が必要な記載箇所(赤枠)	
割り込み種類	割り込み発生タイミング
オーバーランエラー割り込み	FIFO がフルになった時の STOP ビット受信後
ブレークエラー割り込み	STOP ビット受信後
パリティエラー割り込み	パリティデータ受信後
フレーミングエラー割り込み	フレームオーバーとなる Bit データを受信した後
受信タイムアウト割り込み	受信 FIFO にデータを取り込んでから、Baud16 の 511 クロック後
送信割り込み	最終 DATA の MSB データを送信した後
受信割り込み	STOP ビット受信後

4. 誤記修正・追記内容

製品により送信割り込みの割り込み発生タイミングの記載が異なりますが、共通して正しい記載内容は以下となります。

4.1. 送信割り込み発生タイミング

送信 FIFO 未使用時の送信割り込みは、次データに対する送信バッファへの書き込みタイミングを通知するため、送信バッファが空になったタイミングで発生します。送信割り込みは送信バッファに次のデータが書き込まれると自動的にクリアされるため、連続的にデータを送信し続ける場合はソフトウェアによる送信割り込みのクリアは必要ありません(UARTxICR<TXIC> = "1"設定)。

また、送信を終了する場合は、最終送信データがシフトレジスタに転送され、送信バッファが空になった際に最後の送信割り込みが発生します。送信バッファに次のデータを書き込まない場合は、割り込みハンドラー内でソフトウェアによる割り込みクリア(UARTxICR<TXIC> = "1"設定)を実行することで送信割り込みを意図的にクリアすることができます。

なお、データ送信中にソフトウェアで送信割り込みクリア(UARTxICR<TXIC> = "1"設定)を実行した場合、送信完了時の STOP ビット発生と同時のタイミングで送信バッファにデータの書き込みを行うと、送信割り込みは発生しません。確実に送信割り込みを発生させる場合は、データ送信中にソフトウェアで送信割り込みをクリアしないで送信バッファにデータを書き込むか、送信が停止している状態(UARTxFR<BUSY> = "0"のとき)で送信バッファにデータを書き込んでください。

以上

Not Recommended
for New Design

ARM, ARM Powered, AMBA, ADK, ARM9TDMI, TDMI, PrimeCell, RealView, Thumb, Cortex, Coresight, ARM9, ARM926EJ-S, Embedded Trace Macrocell, ETM, AHB, APB, and KEIL はARM LimitedのEUおよびその他の国における登録商標または商標です。

ARM

はじめに～ 本仕様書での表記に関する注意点

各周辺回路(IP)には、SFR(Special Function Register)と呼ばれる制御レジスタが準備されています。本仕様書では、以下のルールに従って表現しています。

a) IP の一覧表

- IP の一覧表では、レジスタ名称、アドレス、簡単な説明が表現されています。
- すべてのレジスタには、32bit で表現されるユニークなアドレスが割り振られており、下記のように『 **base アドレス + (固有)アドレス** 』のように分割して表現されています。

Register Name	Address (base+)	Description
SAMPLE	0x0001	Sample レジスタ
...	...	...

base アドレス= 0x0000_0000

注 1) 本レジスタ(SAMPLE)の場合、00000000 番地(hex)+0001 番地(hex)のため、00000001 番地

注 2) 本レジスタは記述説明用のサンプルです。本マイコンには存在しません。

b) 各 SFR(レジスタ)の説明

- 各々のレジスタは、基本的にすべて 32bit のレジスタで構成されています(一部例外有)。
- 各レジスタの説明では、対象ビット、ビットシンボル、タイプ、リセット後の初期値、機能説明が表現されています。

Address =(0x0000_0000) + 0x0001

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	—	—	Undefined	Read as undefined. Write as zero.
[7:6]	SAMPLE76	R/W	0y00	Sample 設定 0y00: サンプルモード 0 に設定 0y01: サンプルモード 1 に設定 0y10: サンプルモード 2 に設定 0y11: サンプルモード 3 に設定
...	...	...	...	...

注 1) 基本のタイプは、基本的に
 R/W (READ WRITE) : 読み出し/書き込み可能
 RO (READ ONLY) : 読み出しのみ可能
 WO (WRITE ONLY) : 書き込みのみ可能
 の 3 種類に分かれています。

一部、例外のタイプ(USB デバイスコントローラ、SD ホストコントローラ)が存在します。詳細は、各章の説明を参照してください。

注 2) ビットの状態を示す表現は
 16 進表記を 0x00FF = 255 (10 進表記)
 2 進表記を 0y0101 = 5 (10 進表記)
 で表現しています。

注 3) 1 ワード(Word)は 32 ビットと定義します。

32 ビット RISC マイクロプロセッサ TMPA901CMXBG

1. 概要と特長

TMPA901CM は、ARM9TM_cpu コアを内蔵した、32 ビット RISC マイクロプロセッサです。

TMPA901CMXBG は 177 ピン BGA パッケージ製品です。

特長は次のとおりです。

- (1) ARM 社製 ARM926EJTM-S を使用
 - ・データキャッシュ 16kbyte
 - ・インストラクションキャッシュ 16kbyte
- (2) 最大動作周波数 : 200 MHz(@0~70℃) / 150MHz(@-20~85℃)
- (3) 7-Layer のマルチバスシステムを採用
 - ・Bus Master1:CPU データ
 - ・Bus Master2:CPU 命令
 - ・Bus Master3:LCD コントローラ
 - ・Bus Master4:LCD データプロセスアクセラレータ
 - ・Bus Master5:DMA コントローラ 1
 - ・Bus Master6:DMA コントローラ 2
 - ・Bus Master7:USB デバイスコントローラ
- (4) メモリアクセス
 - ・内蔵 RAM: 32K バイト (プログラム、データ、表示メモリ用として使用可能)
 - ・内蔵 ROM: 16K バイト (Boot 専用メモリ)
 - USB から内蔵 RAM へのロードが可能。
 - ・4G バイトのリニアアクセス空間 (有効空間は約 1.7G バイト)
 - ・セパレートバスシステム:
 - 外部アドレス 24bit : A0-A23
 - 外部データバス 16bit : D0-D15
- (5) メモリコントローラ
 - ・チップセレクト出力: 2 チャンネル
 - ・DRAM 専用チップセレクト: 1 チャンネル
 - ・外部端子の選択により、SDR(Single Data Rate)の SDRAM と、DDR(Double Data Rate)の LVCMOS I/O を持つ、SDRAM を使用可能 (SSTL_IO の DDR SDRAM は使用できません)。
 - ・非同期 Static メモリをサポート、同期式 Static メモリはサポートしません。
- (6) 16 ビットタイマ: 6 チャンネル
 - ・内 4 チャンネルのタイマを使って、2 系統 PWM 出力に対応
- (7) 同期式シリアルバスインタフェース: 1 チャンネル
 - ・SPI モード/MicroWire モードをサポート
- (8) I²C バスインタフェース: 1 チャンネル
- (9) UART : 2 チャンネル
 - ・チャンネル 0 : TXD/RXD のみをサポート / IrDA1.0 モードをサポート
 - ・チャンネル 1 : TXD/RXD のみをサポート

(10) USB デバイス コントローラ: 1 チャンネル

- ・ USB(REV2.0)に対応
- ・ High Speed (480Mbps)対応(Low Speed 非対応)
- ・ 4 エンドポイントに対応
 - エンド・ポイント 0: コントロール 64 バイト× 1-FIFO
 - エンド・ポイント 1: バルク(Device→Host: IN 転送) 512 バイト× 2-FIFO
 - エンド・ポイント 2: バルク(Host→Device: OUT 転送)512 バイト× 2-FIFO
 - エンド・ポイント 3: 割り込み 64 バイト× 1-FIFO

(11) USB Host コントローラ: 1 チャンネル

- ・ フルスピード(12Mbps)に対応

(12) I²S(Inter-IC Sound)インターフェース: 2 チャンネル

- ・ チャンネル 0(受信用: 32Byte FIFO×2)
- ・ チャンネル 1(送信用: 32Byte FIFO×2)

チャンネル 0、1 共通使用端子あり

(13) LCD コントローラ

- ・ 800×480 ピクセルのサイズをサポート
- ・ TFT/STN パネルをサポート
- ・ STN はモノクロ 4/15/64 階調、カラーは 256/3375 階調をサポート
- ・ TFT は、16bit カラーに対応

(14) LCD データプロセスアクセラレータ

- ・ スケーラ機能(拡大/縮小)
- ・ フィルタ機能(バイキュービック方)
- ・ 画像 BLEND 機能(Font ブレンド対応)

(15) RTC(リアルタイムクロック)

(16) メロディアラームジェネレータ

- ・ 8 種類のアラームパターンを出力可能

(17) キーオンウェイクアップ(キー入力割込み)

(18) 10 ビット AD コンバータ (サンプルホールド回路内蔵): 4 チャンネル

(19) タッチスクリーンインターフェースに対応

- ・ 低抵抗のスイッチを内蔵し、縦横切り替え用の外付け部品を削除可能

(20) ウォッチドッグタイマ

(21) 周波数検知回路

- ・ 高周波発振フェイルセーフ

(22) 割り込み機能: 21 種類

- ・ 外部 (7 本) 3 種類 外部割込み(エッジ: 立ち上がり、立ち下がり/レベル: High、Low)およびキー入力
- ・ 内部 18 種類 16bit Timer × 3、RTC × 1、A/D コンバータ × 1
LCDC × 1、NANDFC × 1、UART × 2、
I²C × 1、SSP × 1、USB Device × 1、USB Host × 1、I²S × 1、
LCDDA × 1、DMAC × 2、WDT × 1

(23) 入出力ポート: 43

(24) DMA コントローラ: 8 チャンネル

(25) NAND - Flash メモリのインターフェース:2 チャンネル

- ・ NAND - Flash メモリへの接続が容易
- ・ 2LC(2 値)、4LC(4 値)両タイプに対応
- ・ Data Bus 8Bit、Page Size 512/2048Byte に対応
- ・ 4-address 訂正、5-address 以上のエラー検出が可能な、Reed Solomon 演算回路内蔵

(26) スタンバイ機能

- ・ スタンバイモード時における各端子の状態をビット単位にて設定可能
- ・ リーク(漏れ)電流を対策するための電源管理回路(PMC)を内蔵

(27) クロック制御機能

- ・ クロック逡倍回路(PLL)を2ブロック内蔵し、外付け 10~25MHz の発振子から下記の各々クロックを供給します。

0~70℃時 : USB デバイス用クロック 480MHz と CPU へのクロック 200 MHz の供給が可能
(USB 使用時、CPU クロックは 192MHz)

-20~85℃時 : USB デバイス用クロック 480MHz と CPU へのクロック 150MHz の供給が可能
(USB 使用時、CPU クロックは 144MHz)

- ・ クロックギア機能: 高周波クロック $f_c \sim f_c/8$ まで切り替え可能
- ・ 時計用クロック ($f_s = 32.768 \text{ kHz}$)

(28) 周波数検知回路(OFD : Oscillation Frequency Detector)内蔵

(29) 動作電圧

- ・ 内部 DVCC1A、DVCC1B= $1.5\text{V} \pm 0.1\text{V}$
- ・ 高周波発振器、PLL 用電源 DVCC1C= $1.5\text{V} \pm 0.1\text{V}$
- ・ メモリ用外部 I/O DVCCM= $3.0\text{V} \sim 3.6\text{V}$ または $1.8\text{V} \pm 0.1\text{V}$
- ・ 一般用外部 I/O DVCC3IO= $3.0\text{V} \sim 3.6\text{V}$
- ・ AD コンバータ用外部 I/O AVCC3AD= $3.0\text{V} \sim 3.6\text{V}$
- ・ USB Device 用外部 I/O AVDD3T/C= $3.3\text{V} \pm 0.15\text{V}$
- ・ USB Host 用外部 I/O AVCC3H= $3.0\text{V} \sim 3.6\text{V}$

(30) DSU(JTAG)機能

- ・ ARM9 コアの ICE 機能を、JTAG にてサポート

(31) パッケージ

- ・ 177 ピン FBGA : P-FBGA177-1313-0.8C4

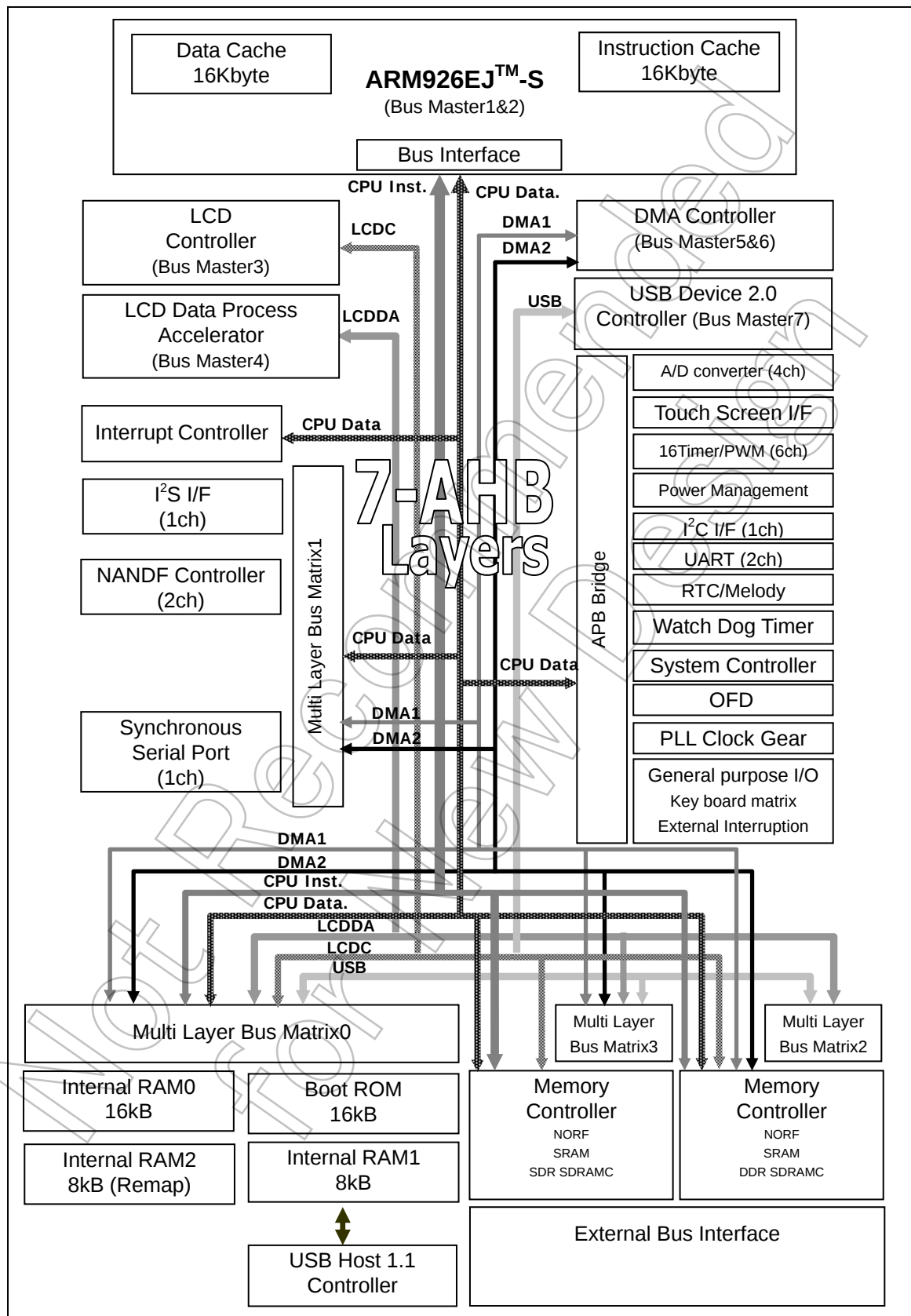


図 1.1 TMPA901CM ブロック図

2. ピン機能

TMPA901CM の入出力ピンの名称と概略機能を示します。

2.1 ピンの配置図 (Top View)

TMPA901CMのピン配置図は、図 2.1.1のとおりです。

TMPA901CM
FBGA177-P-1313-0.8C4
TOP VIEW
(上面からの透視図)

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15
B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15
C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15
D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15
E1	E2	E3	E4	E5							E12	E13	E14	E15
F1	F2	F3	F4								F12	F13	F14	F15
G1	G2	G3	G4								G12	G13	G14	G15
H1	H2	H3	H4								H12	H13	H14	H15
J1	J2	J3	J4								J12	J13	J14	J15
K1	K2	K3	K4								K12	K13	K14	K15
L1	L2	L3	L4								L12	L13	L14	L15
M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15
N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11	N12	N13	N14	N15
P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15
R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15

図 2.1.1 ピン配置図

表 2.1.1 ピン構成

	1	2	3	4	5	6	7	8
A	A1 DVSSCOM	A2 SM3/XT2	A3 SM2/XT1	A4 PU3/NDD3/LD3	A5 PU2/NDD2/ LD2	A6 PU1/NDD1/ LD1	A7 PU0/NDD0/ LD0	A8 SE5/A5
B	B1 SP0/TCK	B2 PC2/PWE	B3 PC3/MLDALM/PWM 1OUT	B4 PU7/NDD7/LD7	B5 PU6/NDD6/ LD6	B6 PU5/NDD5/ LD5	B7 PU4/NDD4/ LD4	B8 SF3/A11
C	C1 SP4/RTCK	C2 SP1/TMS	C3 PC4/FSOUT/PWM3 OUT	C4 PV3/NDCLE/LD11	C5 PV2/NDAL /LD10	C6 PV1/NDWE _n /LD9	C7 PV0/NDRE _n /LD8	C8 SG0/A16
D	D1 SP5/TDO	D2 SP2/TDI	D3 PC6/I2C0CL/USBP ON	D4 PV7/LD15	D5 PV6/NDRB/ LD14	D6 PV5/NDCE1 _n /LD13	D7 PV4/NDCE0 _n /LD12	D8 SG4/A20
E	E1 DVCC3IO	E2 SP3/TRST _n	E3 PC7/I2C0DA/INT9	E4 DVCC3IO	E5 DVSSCOM			
F	F1 DVCC1B	F2 DVCC3IO	F3 DVCC3IO	F4 DVCC3IO				
G	G1 DVSSCOM	G2 DVSSCOM	G3 DVSSCOM	G4 DVSSCOM				
H	H1 DVCC1A	H2 DVCC1A	H3 DVCC1A	H4 DVCC1A				
J	J1 AVCC3AD	J2 VREFH	J3 VREFL	J4 DVCC1B				
K	K1 PD4/AN4/MX	K2 PD5/AN5/MY	K3 AVSS3AD	K4 DVCC3IO				
L	L1 PD6/INTA(INTT S1)/AN6	L2 PD7/INTB/AN7	L3 DVCC3IO	L4 SM6/AM0				
M	M1 DVCC3IO	M2 DVCC3IO	M3 PA0/KI0	M4 PA2/KI2	M5 DVSSCOM	M6 AVSS3C	M7 DVCC1A	M8 DVCC3IO
N	N1 SM4/RESET _n	N2 PN0/U0TXD/SIR00 UT	N3 PA1/KI1	N4 PA3/KI3	N5 DVSSCOM	N6 AVDD3C	N7 AVDD3T1	N8 AVDD3T0
P	P1 PN1/U0RXD/SI R0IN	P2 SM7/AM1	P3 DVCC1C	P4 DVSS1C	P5 DVSSCOM	P6 SR3/REXT	P7 AVSS3T2	P8 AVSS3T1
R	R1 DVSSCOM	R2 SM0/X1	R3 SM1/X2	R4 DVCC1C	R5 SR4/VSNS	R6 AVSS3T3	R7 SR1/DDM	R8 SR0/DDP
	1	2	3	4	5	6	7	8

表 2.1.2 ピン構成

9	10	11	12	13	14	15	
A9 SE4/A4	A10 SE3/A3	A11 SE2/A2	A12 SE1/A1	A13 SE0/A0	A14 SL2/DMCAP	A15 DVSSCOM	A
B9 SG7/A23	B10 SF2/A10	B11 SF1/A9	B12 SF0/A8	B13 SE7/A7	B14 SE6/A6	B15 SL1/DMCD CLKN	B
C9 SF7/A15	C10 SG6/A22	C11 SF6/A14	C12 SF5/A13	C13 SF4/A12	C14 SK0/DMCSDQM0/D MCDDM0	C15 SL0/DMCD CLKP/DMC SCLK	C
D9 SG3/A19	D10 SG2/A18	D11 SG5/A21	D12 SG1/A17	D13 SK4/SMCWE _n	D14 SK1/DMCSDQM1/D MCDDM1	D15 SL6/DMCCL KIN	D
			E12 SK5/SMCBE1 _n	E13 SJ5/DMCBA1	E14 SB7/D15	E15 SB6/D14	E
			F12 SJ6/DMCCKE	F13 SJ4/DMCBA0	F14 SB5/D13	F15 SB4/D12	F
			G12 DVCCM	G13 SJ3/DMCCAS _n	G14 SB3/D11	G15 SB2/D10	G
			H12 DVCCM	H13 SJ2/DMCRAS _n	H14 SB1/D9	H15 SB0/D8	H
			J12 DVCCM	J13 SJ1/DMCWE _n	J14 SL5/DMCDDQS1	J15 SL4/DMCD DQS0	J
			K12 DVC01A	K13 SJ0/SMCOE _n	K14 SA7/D7	K15 SA6/D6	K
			L12 DVCC1B	L13 SH7/DMCCS _n	L14 SA5/D5	L15 SA4/D4	L
M9 DVCC3IO	M10 SN2/SELJTAG	M11 AVCC3H	M12 SN1/SELDVCCM	M13 SH4/SMCCS1 _n	M14 SA3/D3	M15 SA2/D2	M
N9 PB2/KO2/LC LFP	N10 PB1/KO1/LCLAC	N11 PT2/SP0DO /I2S0DATI	N12 PT4/U1TXD/USBPON	N13 SH3/SMCCS0 _n	N14 SA1/D1	N15 SA0/D0	N
P9 SN0/SELMEM C	P10 PB0/KO0/LCLCP	P11 PT6/U1CTS n/I2S0DATO	P12 PT1/SP0CLK/I2S0CL K	P13 PT0/SP0FSS/I2S 0WS	P14 PT3/SP0DI/I2S0MC LK	P15 SH2/SMCB E0 _n	P
R9 AVSS3T0	R10 PB3/KO3/LCLLP	R11 PT7/X1USB	R12 PT5/U1RXD/USBOC	R13 SN7/HDM	R14 SN6/HDP	R15 DVSSCOM	R
9	10	11	12	13	14	15	

2.2 ピン名称と機能

入出力ピンの名称と機能は次のとおりです。

メモリ関連の端子は、外部端子「SELMEMC」の状態によって 2 種類の MPMC(MPMC0/1)を選択し、端子が切り替わります。

表 2.2.1 ピン名称と機能(1/6)

ピン名称	ピン数	入出力	機能	備考
SA0~SA7 D0~D7	8	入出力	データ: データバス D0~D7	MPMC0/1 兼用
SB0~SB7 D8~D15	8	入出力	データ: データバス D8~D15	MPMC0/1 兼用
SE0~SE7 A0~A7	8	出力	アドレス: アドレスバス A0~A7	MPMC0/1 兼用
SF0~SF7 A8~A15	8	出力	アドレス: アドレスバス A8~A15	MPMC0/1 兼用
SG0~SG7 A16~A23	8	出力	アドレス: アドレスバス A16~A23	MPMC0/1 兼用
SH2 SMCBE0n	1	出力	NORF/SRAM/MROM 用、バイトイネーブル信号(D0~D7)	MPMC0/1 兼用
SK5 SMCBE1n	1	出力	NORF/SRAM/MROM 用、バイトイネーブル信号(D8~D15)	MPMC0/1 兼用
SH3 SMCCS0n	1	出力	NORF/SRAM/MROM 用、チップセレクト信号 0	MPMC0/1 兼用
SH4 SMCCS1n	1	出力	NORF/SRAM/MROM 用、チップセレクト信号 1	MPMC0/1 兼用
SH7 DMCCSn DMCCSn	1	出力 出力	SDR_SDRAM 用、ライトイネーブル信号 DDR_SDRAM 用、ライトイネーブル信号	MPMC0 使用時 MPMC1 使用時
SJ0 SMCOEn	1	出力	NORF/SRAM/MROM 用、アウトイネーブル信号	MPMC0/1 兼用
SJ1 DMCWEn DMCWEn	1	出力 出力	SDR_SDRAM 用、ライトイネーブル信号 DDR_SDRAM 用、ライトイネーブル信号	MPMC0 使用時 MPMC1 使用時
SJ2 DMCRASn DMCRASn	1	出力 出力	SDR_SDRAM 用、Row アドレスストローブ信号 DDR_SDRAM 用、Row アドレスストローブ信号	MPMC0 使用時 MPMC1 使用時
SJ3 DMCCASn DMCCASn	1	出力 出力	SDR_SDRAM 用、Column アドレスストローブ信号 DDR_SDRAM 用、Column アドレスストローブ信号	MPMC0 使用時 MPMC1 使用時

注) 各々のピンに対する「SA0~SA7、...SR0~SR4」の名前は、便宜上のシンボルであり、汎用ポート機能を示す「PA0~PA7、...PV0~PV7」とは異なり、機能はありませんので御注意ください。

表 2.2.1 ピン名称と機能(2/6)

ピン名称	ピン数	入出力	機能	備考
SJ4 DMCBA0 DMCBA0	1	— 出力 出力	— SDR_SDRAM 用、BANK0 ストローブ信号 DDR_SDRAM 用、BANK0 ストローブ信号	— MPMC0 使用時 MPMC1 使用時
SJ5 DMCBA1 DMCBA1	1	— 出力 出力	— SDR_SDRAM 用、BANK1 ストローブ信号 DDR_SDRAM 用、BANK1 アドレスストローブ信号	— MPMC0 使用時 MPMC1 使用時
SJ6 DMCCKE DMCCKE	1	— 出力 出力	— SDR_SDRAM 用、クロックイネーブル信号 DDR_SDRAM 用、クロックイネーブル信号	— MPMC0 使用時 MPMC1 使用時
SK0 DMCSDQM0 DMCDDM0	1	— 出力 出力	— SDR_SDRAM 用、バイトイネーブル信号(D0~D7) DDR_SDRAM 用、データマスク信号(D0~D7)	— MPMC0 使用時 MPMC1 使用時
SK1 DMCSDQM1 DMCDDM1	1	— 出力 出力	— SDR_SDRAM 用、バイトイネーブル信号(D8~D15) DDR_SDRAM 用、データマスク信号(D8~D15)	— MPMC0 使用時 MPMC1 使用時
SK4 SMCWEn	1	— 出力	— NORF/SRAM/MROM 用、ライトイネーブル信号	— MPMC0/1 兼用
SL0 DMCSCLK DMCDCLKP	1	— 出力 出力	— SDR_SDRAM 用、クロック信号 DDR_SDRAM 用、正位相クロック信号	— MPMC0 使用時 MPMC1 使用時
SL1 — DMCDCLKN	1	— — 出力	— 未使用 DDR_SDRAM 用、逆位相クロック信号	— MPMC0 使用時 MPMC1 使用時
SL2 DMCAP DMCAP	1	— 出力 出力	— SDR_SDRAM 用、アドレス／プリチャージ信号 DDR_SDRAM 用、アドレス／プリチャージ信号	— MPMC0 使用時 MPMC1 使用時
SL4 — DMCDDQS0	1	— — 入出力	— 未使用 DDR_SDRAM 用、データストローブ信号(D0~D7)	— MPMC0 使用時 MPMC1 使用時
SL5 — DMCDDQS1	1	— — 入出力	— 未使用 DDR_SDRAM 用、データストローブ信号(D8~D15)–	— MPMC0 使用時 MPMC1 使用時
SL6 DMCCLKIN	1	— 入力	— SDR/DDR_SDRAM 用、FB クロック	— MPMC0/1 兼用

注) 各々のピンに対する「SA0~SA7、…SR0~SR4」の名前は、便宜上のシンボルであり、汎用ポート機能を示す「PA0~PA7、…PV0~PV7」とは異なり、機能はありませんので御注意ください。

表 2.2.1 ピン名称と機能(3/6)

ピン名称	ピン数	入出力	機能	備考
SM0 X1	1	入力	高周波発振器接続入力端子	
SM1 X2	1	出力	高周波発振器接続出力端子	
SM2 XT1	1	入力	低周波発振器接続入力端子	
SM3 XT2	1	出力	低周波発振器接続出力端子	
SM4 RESETn	1	入力	リセット: TMPA901CM が初期化されます (シュミット入力, プルアップ抵抗付き)	
SM6~SM7 AM0~AM1	2	入力	起動モード入力端子	
SN0 SELMEMC	1	入力	メモリコントローラ選択端子	
SN1 SELDVCCM	1	入力	メモリ関連動作電圧選択端子	
SN2 SELJTAG	1	入力	パウンダリスキャン切換え端子	
SN6 HDP	1	入出力	USB ホスト用データ端子(D+)	
SN7 HDM	1	入出力	USB ホスト用データ端子(D-)	
SP0 TCK	1	入力	JTAG 用クロック端子	
SP1 TMS	1	入力	JTAG 用端子	
SP2 TDI	1	入力	JTAG 用データ入力端子	
SP3 TRSTn	1	入力	JTAG 用リセット端子	
SP4 RTCK	1	出力	JTAG 用クロック出力端子	
SP5 TDO	1	出力	JTAG 用データ出力端子	
SR0 DDP	1	入出力	USB デバイス用データ端子(D+)	
SR1 DDM	1	入出力	USB デバイス用データ端子(D-)	
SR3 REXT	1	入力	VSNS 端子と 12kΩの抵抗で接続します	
SR4 VSNS	1	入力	REXT 端子と 12kΩの抵抗で接続します	

注) 各々のピンに対する「SA0~SA7、…SR0~SR4」の名前は、便宜上のシンボルであり、汎用ポート機能を示す「PA0~PA7、…PV0~PV7」とは異なり、機能はありませんので御注意ください。

表 2.2.1 ピン名称と機能 (4/6)

ピン名称	ピン数	入出力	機能	
PA0~PA3 KI0~KI3	4	入力 入力	ポート A0~A3: 入力ポート キー入力 KI0~KI3: キーオンウェイクアップ 0~4 用端子 (シュミット入力, プルアップ抵抗付き)	
PB0 KO0 LCLCP	1	出力 出力 出力	ポート B0: 出力ポート キー出力 KO0: キーアウト端子(オープンドレイン設定可能) LCD ドライバ出力端子	
PB1 KO1 LCLAC	1	出力 出力 出力	ポート B1: 出力ポート キー出力 KO1: キーアウト端子(オープンドレイン設定可能) LCD ドライバ出力端子	
PB2 KO2 LCLFP	1	出力 出力 出力	ポート B2: 出力ポート キー出力 KO2: キーアウト端子(オープンドレイン設定可能) LCD ドライバ出力端子	
PB3 KO3 LCLLP	1	出力 出力 出力	ポート B3: 出力ポート キー出力 KO3: キーアウト端子(オープンドレイン設定可能) LCD ドライバ出力端子	
PC2 PWE	1	出力 出力	ポート C2: 出力ポート 外部電源制御出力: 外部電源の ON/OFF を制御するための端子です。通常動作時は"H"レベルを出力し、スタンバイモード時には"L"レベルを出力します。	
PC3 MLDALM PWM0OUT	1	出力 出力 出力	ポート C3: 出力ポート メロディアラーム出力端子 タイマの PWM 出力端子	
PC4 FSOUT PWM2OUT	1	出力 出力 出力	ポート C4: 出力ポート 低周波出力クロック端子 タイマの PWM 出力端子	
PC6 I2C0CL USBPON	1	入出力 入出力 出力	ポート C6: 入出力ポート I ² C クロック入出力 USB ホスト電源 ON 信号出力	
PC7 I2C0DA INT9 USBOCn	1	入出力 入出力 入力 入力	ポート C7: 入出力ポート I ² C データ入出力 割り込み要求端子 9: 立ち上がり/立ち下がりエッジをプログラム可能な割り込み要求端子 USB ホスト過電流の検出 USB OC 入力	
PD4 AN4 MX	1	入力 入力 出力	ポート D4: 入力ポート アナログ入力 4: AD コンバータ 入力端子 X-マイナス: タッチパネル用 X-接続端子	
PD5 AN5 MY	1	入力 入力 出力	ポート D5: 入力ポート アナログ入力 4: AD コンバータ 入力端子 Y-マイナス: タッチパネル用 Y-接続端子	
PD6 AN6 PX INTA(INTTSI)	1	入力 入力 出力 入力	ポート D6: 入力ポート アナログ入力 6: AD コンバータ 入力端子 X-プラス: タッチパネル用 X-接続端子 割り込み要求端子 A: 立ち上がり/立ち下がりエッジをプログラム可能な割り込み要求端子	
PD7 AN7 PY INTB	1	入力 入力 出力 入力	ポート D7: 入力ポート アナログ入力 7: AD コンバータ 入力端子 Y-プラス: タッチパネル用 Y-接続端子 割り込み要求端子 B: 立ち上がり/立ち下がりエッジをプログラム可能な割り込み要求端子	

表 2.2.1 ピン名称と機能 (5/6)

ピン名称	ピン数	入出力	機能	備考
PN0 U0TXD SIR0OUT	1	入出力 出力 出力	ポート N0: 入出力ポート UART 機能 0 送信 データ IrDA1.0 用データ出力端子	
PN1 U0RXD SIR0IN	1	入出力 入力 入力	ポート N1: 入出力ポート UART 機能 0 受信 データ IrDA1.0 用データ入力端子	
PT0 SP0FSS I2S0WS	1	入出力 入出力 入出力	ポート T0: 入出力ポート SSP0 用 FSS 端子 I ² S0 ワードセレクト 入力/出力	
PT1 SP0CLK I2S0CLK	1	入出力 入出力 入出力	ポート T1: 入出力ポート SSP0 用クロック端子 I ² S0 シリアルクロック 入力/出力	
PT2 SP0DO I2S0DATI	1	入出力 出力 入力	ポート T2: 入出力ポート SSP0 用データ出力端子 I ² S0 受信シリアルデータ入力	
PT3 SP0DI I2S0MCLK	1	入出力 入力 出力	ポート T3: 入出力ポート SSP0 用データ入力端子 I ² S0 受信回路のマスタクロック出力	

表 2.2.1 ピン名称と機能 (6/6)

ピン名称	ピン数	入出力	機能	備考
PT4 U1TXD USBPON	1	入出力 出力 出力	ポート T4: 入出力ポート UART 機能 1 送信 データ USB ホスト電源 ON 信号出力	
PT5 U1RXD USBOCn	1	入出力 入力 入力	ポート T5: 入出力ポート UART 機能 1 受信 データ USB ホスト過電流の検出 USB OC 入力	
PT6 U1CTSn I2S1DATO	1	入出力 出力 出力	ポート T6: 入出力ポート UART 機能 1 データ送信可能 I ² S1 送信のシリアルデータ出力	
PT7 X1USB	1	入出力 入力	ポート T7: 入出力ポート USB 用クロック入力端子	
PU0~PU7 NDD0~NDD7 LD0~LD7	8	入出力 入出力 入出力	ポート U: 入出力ポート NANDF メモリ用データバス LCD ドライバ用データバス	
PV0 NDREn LD8	1	入出力 出力 出力	ポート V0: 入出力ポート NAND-Flash 用リードイネーブル LCD ドライバ用データバス	
PV1 NDWEen LD9	1	入出力 出力 出力	ポート V1: 入出力ポート NAND-Flash 用ライトイネーブル LCD ドライバ用データバス	
PV2 NDALE LD10	1	入出力 出力 出力	ポート V2: 入出力ポート NAND-Flash 用アドレスラッチイネーブル LCD ドライバ用データバス	
PV3 NDCLE LD11	1	入出力 出力 出力	ポート V3: 入出力ポート NAND-Flash 用コマンドラッチイネーブル LCD ドライバ用データバス	
PV4 NDCE0n LD12	1	入出力 出力 出力	ポート V4: 入出力ポート NAND-Flash0 チップセレクト LCD ドライバ用データバス	
PV5 NDCE1n LD13	1	入出力 出力 出力	ポート V5: 入出力ポート NAND-Flash1 チップセレクト LCD ドライバ用データバス	
PV6 NDRB LD14	1	入出力 入力 出力	ポート V6: 入出力ポート NAND-Flash Ready(1)/Busy(0)入力 LCD ドライバ用データバス	
PV7 LD15	1	入出力 出力	ポート V7: 入出力ポート LCD ドライバ用データバス	

ピン名称	ピン数	電源	機能	備考
DVCC1A	6	電源	内部 主エリア用 VCC 電源	
DVCC1B	3	電源	内部 B/U エリア用 VCC 電源	
DVCC1C	2	電源	高周波クロック/PLL 回路用 VCC 電源	
DVSS1C	1	電源	高周波クロック/PLL 回路用 VSS 電源	
DVCC3IO	11	電源	外部 I/O 用 VCC 電源 (一般)、(LCD)	
DVCCM	3	電源	外部 I/O 用 VCC 電源 (メモリ用)	
AVCC3AD	1	電源	外部 I/O 用 VCC 電源 (A/D)	
AVSS3AD	1	電源	外部 I/O 用 VSS 電源 (A/D)	
VREFH	1	入力	A/D コンバータ用リファレンス電圧	
VREFL	1	入力	A/D コンバータ用リファレンス電圧	
AVDD3Tx	2	電源	外部 I/O 用 VDD 電源 (USB Device)	
AVSS3Tx	4	電源	外部 I/O 用 VSS 電源 (USB Device)	
AVDD3C	1	電源	外部 I/O 用 VDD 電源 (USB Device)	
AVSS3C	1	電源	外部 I/O 用 VSS 電源 (USB Device)	
AVCC3H	1	電源	外部 I/O 用 VCC 電源 (USB Host)	
DVSSCOM	12	電源	共通 VSS 電源 (GND)	

- 各端子の対応電源、機能および、初期値一覧表 1 (DVCCM 系統)

対応電源	代表名	兼用機能	兼用機能	Pull up/Down	入力 Buffer	RESET 後初期値 機能 / 端子状態
DVCCM	SA0~SA7	D0~D7	—	—	ON	D0~D7 / Hz*
	SB0~SB7	D8~D15	—	—	ON	D8~D15 / Hz*
	SE0~SE7	A0~A7	—	—	—	Address out / “L”出力
	SF0~SF7	A8~A15	—	—	—	Address out / “L”出力
	SG0~SG7	A16~A23	—	—	—	Address out / “L”出力
	SH2	SMCBE0n	—	—	—	SMCBE0n out / “H”出力
	SK5	SMCBE1n	—	—	—	SMCBE1n out / “H”出力
	SH3	SMCCS0n	—	—	—	SMCCS0n out / “H”出力
	SH4	SMCCS1n	—	—	—	SMCCS1n out / “H”出力
	SH7	DMCCSn	—	—	—	DMCCSn out / “H”出力
	SJ0	SMCOEn	—	—	—	SMCOEn out / “H”出力
	SJ1	DMCWEn	—	—	—	DMCWEn out / “H”出力
	SJ2	DMCRASn	—	—	—	DMCRASn out / “H”出力
	SJ3	DMCCASn	—	—	—	DMCCASn out / “H”出力
	SJ4	DMCBA0	—	—	—	DMCBA0n out / “L”出力
	SJ5	DMCBA1	—	—	—	DMCBA1n out / “L”出力
	SJ6	DMCCKE	—	—	—	DMCCKEn out / “H”出力
	SK0	DMCSDQM0	DMCDDM0	—	—	SELMEMC = 0 の時 DMCSDQM0 out / “L”出力 SELMEMC = 1 の時 DMCDDM0 out / “L”出力
	SK1	DMCSDQM1	DMCDDM1	—	—	SELMEMC = 0 の時 DMCSDQM1 out / “L”出力 SELMEMC = 1 の時 DMCDDM1 out / “L”出力
	SK4	SMCWEn	—	—	—	SMCWEn out / “H”出力

注1) 各々のピンに対する「SA0~SA7、…SU0~SU6」の名前は、便宜上のシンボルであり、汎用ポート機能を示す「PA0~PA7、…PV0~PV7」とは異なり、機能はありませんので御注意ください

注2) 入力 Buffer が「ON」と書かれた端子は、入力許可状態が初期値となりますので、必要に応じて、外部で端子処理をしてください。また、データバス (SA0~SA7、SB0~SB7、SC0~SC7、SD0~SD7)は、常に入力許可状態のため、貫通電流対策として、外部にて Pull_UP/Down などのレベル固定の処理が必要です。

- 各端子の対応電源、機能および、初期値一覧表 2 (DVCCM 系統)

対応電源	代表名	兼用機能	兼用機能	Pull up/Down	入力 Buffer	RESET 後初期値 機能 / 端子状態
DVCCM	SL0	DMCSCLK	DMCDCLKP	—	—	SELMEMC = 0 の時 DMCSCLK out / CLK 出力 SELMEMC = 1 の時 DMCDCLKP out / CLK 出力
	SL1	DMCDCLKN	—	—	—	SELMEMC = 0 の時 無効信号 / “H”出力 SELMEMC = 1 の時 DMCDCLKN out / 反転 CLK 出力
	SL2	DMCAP	—	—	—	DMCAP out / “L”出力
	SL4	DMCDDQS0	—	—	ON	DMCDDQS0 / Hz*
	SL5	DMCDDQS1	—	—	ON	DMCDDQS1 / Hz*
	SL6	DMCCLKIN	—	—	ON	DMCCLKIN input / Hz

注1) 各々のピンに対する「SA0~SA7、…SR0~SR4」の名前は、便宜上のシンボルであり、汎用ポート機能を示す「PA0~PA7、…PV0~PV7」とは異なり、機能はありませんので御注意ください

注2) 入力 Buffer が「ON」と書かれた端子は、入力許可状態が初期値となりますので、必要に応じて、外部で端子処理をしてください。また、DDR_SDRAM 使用時の DQS 信号 (DMCDDQS0、DMCDDQS1)は、常に入力許可状態のため、貫通電流対策として、外部にて Pull_UP/Down などのレベル固定の処理が必要です。

- 各端子の対応電源、機能および、初期値一覧表 3 (DVCC3IO 系統)

対応電源	代表名	兼用機能	兼用機能	Pull up/Down	入力 Buffer	RESET 後初期値 機能 / 端子状態
DVCC3IO	SM2	XT1	—	—	—	発振状態
	SM3	XT2	—	—	—	発振状態
	SM4	RESETn	—	PU	ON	RESETn input / “H”出力
	SM6	AM0	—	—	ON	AM0 input / Hz
	SM7	AM1	—	—	ON	AM1 input / Hz
	SN0	SELMEMC	—	—	ON	SELMEMC input / Hz
	SN1	SELDVCCM	—	—	ON	SELDVCCM input / Hz
	SN2	SELJTAG	—	—	ON	SELJTAG input / Hz
	SP0	TCK	—	—	ON	TCK input / Hz
	SP1	TMS	—	—	ON	TMS input/ Hz
	SP2	TDI	—	—	ON	TDI input / Hz
	SP3	TRSTn	—	—	ON	TRSTn input / Hz
	SP4	RTCK	—	—	—	RTCK out / CLK 出力
	SP5	TDO	—	—	—	TDO out / TDO 出力

注1) 各々のピンに対する「SA0~SA7、…SU0~SU4」の名前は、便宜上のシンボルであり、汎用ポート機能を示す「PA0~PA7、…PV0~PV7」とは異なり、機能はありませんので御注意ください。

注2) 入力 Buffer が「ON」と書かれた端子は、入力許可状態が初期値となりますので、必要に応じて、外部で端子処理をしてください。

- 各端子の対応電源、機能および、初期値一覧表 4 (DVCC3IO 系統)

対応電源	代表名	兼用機能	兼用機能	Pull up/Down	入力 Buffer	RESET 後初期値 機能 / 端子状態
DVCC3IO	PA0 to PA4	KI0 to KI4	—	PU	ON	PA0~PA4 input / “H”出力
	PB0 to PB3	KO0 to KO3	LCLxx	—	—	PB0~PB3 out / “H”出力
	PC2	PWE	—	—	—	PWE out / “H”出力
	PC3	MLDALM	PWM0OUT	—	—	PC3 out / “H”出力
	PC4	FSOUT	PWM2OUT	—	—	PC4 out / “L”出力
	PC6	I2C0CL	—	—	ON	PC6 input / Hz
	PC7	I2C0DA	INT9	—	ON	PC7 input / Hz
	PN0	U0TXD	SIR0OUT	—	ON	PN0 input / Hz
	PN1	U0RXD	SIR0IN	—	ON	PN1 input / Hz
	PT0	SP0FSS	I2S0WS	—	ON	PT0 input / Hz
	PT1	SP0CLK	I2S0CLK	—	ON	PT1 input / Hz
	PT2	SP0DO	I2S0DATI	—	ON	PT2 input / Hz
	PT3	SP0DI	I2S0CLK	—	ON	PT3 input / Hz
	PT4	U1TXD	USBPON	—	ON	PT4 input / Hz
	PT5	U1RXD	USBOC	—	ON	PT5 input / Hz
	PT6	U1CTSn	I2S1DATO	—	ON	PT6 input / Hz
	PT7	X1USB	—	—	ON	PT7 input / Hz
	PU0~PU7	NDD0~NDD7	LD0 to LD7	—	ON	PU0~PU7 / Hz
	PV0	NDREn	LD8	—	ON	PV0 input / Hz
	PV1	NDWEn	LD9	—	ON	PV1 input / Hz
	PV2	NDALE	LD10	—	ON	PV2 input / Hz
	PV3	NDCLE	LD11	—	ON	PV3 input / Hz
	PV4	NDCE0n	LD12	—	ON	PV4 input / Hz
	PV5	NDCE1n	LD13	—	ON	PV5 input / Hz
	PV6	NDRB	LD14	—	ON	PV6 input / Hz
	PV7	—	LD15	—	ON	PV7 input / Hz

注1) 各々のピンに対する「SA0~SA7、…SUR~SR4」の名前は、便宜上のシンボルであり、汎用ポート機能を示す「PA0~PA7、…PV0~PV7」とは異なり、機能はありませんので御注意ください。

注2) 入力 Buffer が「ON」と書かれた端子は、入力許可状態が初期値となりますので、必要に応じて、外部で端子処理をしてください(外部 NAND Flash メモリを使用する場合は、PV4(NDCE0n), PV5(NDCE1n), など端子, 必要に応じて、Pull_UP などのレベル固定を処理してください)。

- 各端子の対応電源、機能および、初期値一覧表 5 (AVCC3AD 系統)

対応電源	代表名	兼用機能	兼用機能	兼用機能	Pull up/Down	入力 Buffer	RESET 後初期値 機能 / 端子状態
	PD4	AN4	MX	—	—	OFF	AN4 input / Hz
	PD5	AN5	MY	—	—	OFF	AN5 input / Hz
	PD6	AN6	PX	INTA(INTTSI)	PD*	ON	AN6 input / Hz
	PD7	AN7	PY	INTB	—	ON	AN7 input / Hz

注1) 各々のピンに対する「SA0~SA7、…SR0~SR4」の名前は、便宜上のシンボルであり、汎用ポート機能を示す「PA0~PA7、…PV0~PV7」とは異なり、機能はありませんので御注意ください。

注2) 入力 Buffer が「ON」と書かれた端子は、入力許可状態が初期値となりますので、必要に応じて、外部で端子処理をしてください。

注1) PD6 の Pull down 抵抗は、RESET 後は OFF 状態です。

- 各端子の対応電源、機能および、初期値一覧表 6 (USB Device)

対応電源	代表名	兼用機能	兼用機能	Pull up/Down	入力 Buffer	RESET 後初期値 機能 / 端子状態
AVDD3C/T	SR0	DDP	—	PD	ON	DP input / “L”出力
	SR1	DDM	—	PD	ON	DM input / “L”出力
	SR3	REXT	—	—	—	REXT input / Hz
	SR4	VSENS	—	—	—	VSENS input / Hz

注1) 各々のピンに対する「SA0~SA7、…SU0~SU4」の名前は、便宜上のシンボルであり、汎用ポート機能を示す「PA0~PA7、…PV0~PV7」とは異なり、機能はありませんので御注意ください。

注2) 入力 Buffer が「ON」と書かれた端子は、入力許可状態が初期値となりますので、必要に応じて、外部で端子処理をしてください。USB の D+、D-信号は PHY 内部に PullDown 抵抗が内蔵されています。

- 各端子の対応電源、機能および、初期値一覧表 7 (USB Host)

対応電源	代表名	兼用機能	兼用機能	Pull up/Down	入力 Buffer	RESET 後初期値 機能 / 端子状態
AVCC3H	SN6	HDP	—	—	ON	HDP input / Hz
	SN7	HDM	—	—	ON	HDM input / Hz

注1) 各々のピンに対する「SA0~SA7、…SU0~SU4」の名前は、便宜上のシンボルであり、汎用ポート機能を示す「PA0~PA7、…PV0~PV7」とは異なり、機能はありませんので御注意ください。

- 各端子の対応電源、機能および、初期値一覧表 8 (OSC)

対応電源	代表名	兼用機能	兼用機能	Pull up/Down	入力 Buffer	RESET 後初期値 機能 / 端子状態
DVCC1C	SM0	X1	—	—	—	発振状態
	SM1	X2	—	—	—	発振状態

注1) 各々のピンに対する「SA0~SA7、…SU0~SU4」の名前は、便宜上のシンボルであり、汎用ポート機能を示す「PA0~PA7、…PV0~PV7」とは異なり、機能はありませんので御注意ください。

注2) 入力 Buffer が「ON」と書かれた端子は、入力許可状態が初期値となりますので、必要に応じて、外部で端子処理をしてください。

3. 動作説明

TMPA901CM の CPU 回路に関する簡易説明をいたします。

3.1 CPU

TMPA901CM の CPU に関する基本動作についてブロックごとに説明します。

尚、本書では CPU ブロックの概要のみ説明しています。動作の詳細は ARM 社にお問い合わせください。

TMPA901CM には ARM 社の 32 ビット RISC プロセッサ ARM926EJ-S™ が内蔵されています。

下記に ARM926EJ-S™ コアの概略図を載せます。

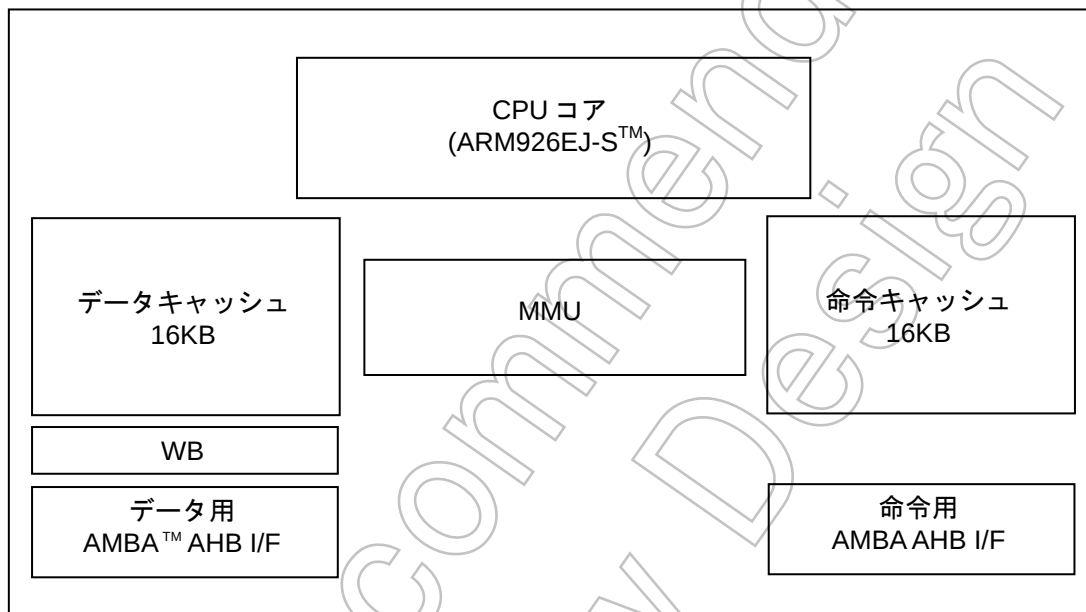


図 3.1.1 ARM926EJ-S™ コア

オリジナルの ARM926EJ-S™ と異なるハードウェアのサポートを以下に示します。

1. Coprocessor I/F はサポートしていません
2. Embedded ICE-RT はサポートしていません
3. TCM I/F はサポートしていません。
4. ETM9™ I/F はサポートしていません。
5. CPU の Version は、ARM 社で管理されており、Version は "r0p5" です。

3.1.1 リセット動作

TMPA901CM にリセットをかけるには、電源電圧が動作範囲内であり、かつ、内部発振器の発振が安定した状態で、少なくとも 20 システムクロック ($0.8\mu\text{s}$ @ $X1 = 25\text{MHz}$)、RESETn 入力端子を “L” レベルにしてください。

リセットで PLL は停止し PLL 出力は非選択となり、クロックギアは TOP(1/1) にセットされます。

よってシステムクロックは 25MHz ($X1 = 25\text{MHz}$) で動作します。

またリセットが受け付けられると、内蔵 I/O および入出力ポートおよびその他の端子は、初期化されます。

内蔵 I/O のレジスタを初期化

(初期値は、Port および Pin の章を参照して下さい。)

注 1) 本 LSI は RAM を内蔵していますが、リセット動作により内蔵 RAM のデータは保持されない場合があります。リセット後、内蔵 RAM のデータを初期化してください。

注 2) 本 LSI には、一部の電源 (DVCC1A, DVCC1C, AVDD3Tx, AVDD3Cx, AVCC3H) を遮断してスタンバイ電流を低減する機能 (PCM 機能) を持っていますが、電源供給カット対象の電源 (DVCC1A, DVCC1C, AVDD3Tx, AVCC3Cx, AVCC3H) を供給しない状態で、リセット動作を行うと内部に貫通電流を流すことがあります。リセットする場合は、まず、電源供給カット対象の電源 (DVCC1A, DVCC1C, AVDD3Tx, AVDD3Cx, AVCC3H) を供給し、十分に電源が安定した後にリセットするようにしてください。

オリジナル ARM926EJ-STM は、リセット直後のベクトル位置と、エンディアンを選択することが出来ますが、本 LSI では以下の設定に固定されています。

エンディアン	起動ベクトル
リトルエンディアン	0x00000000

3.1.2 例外

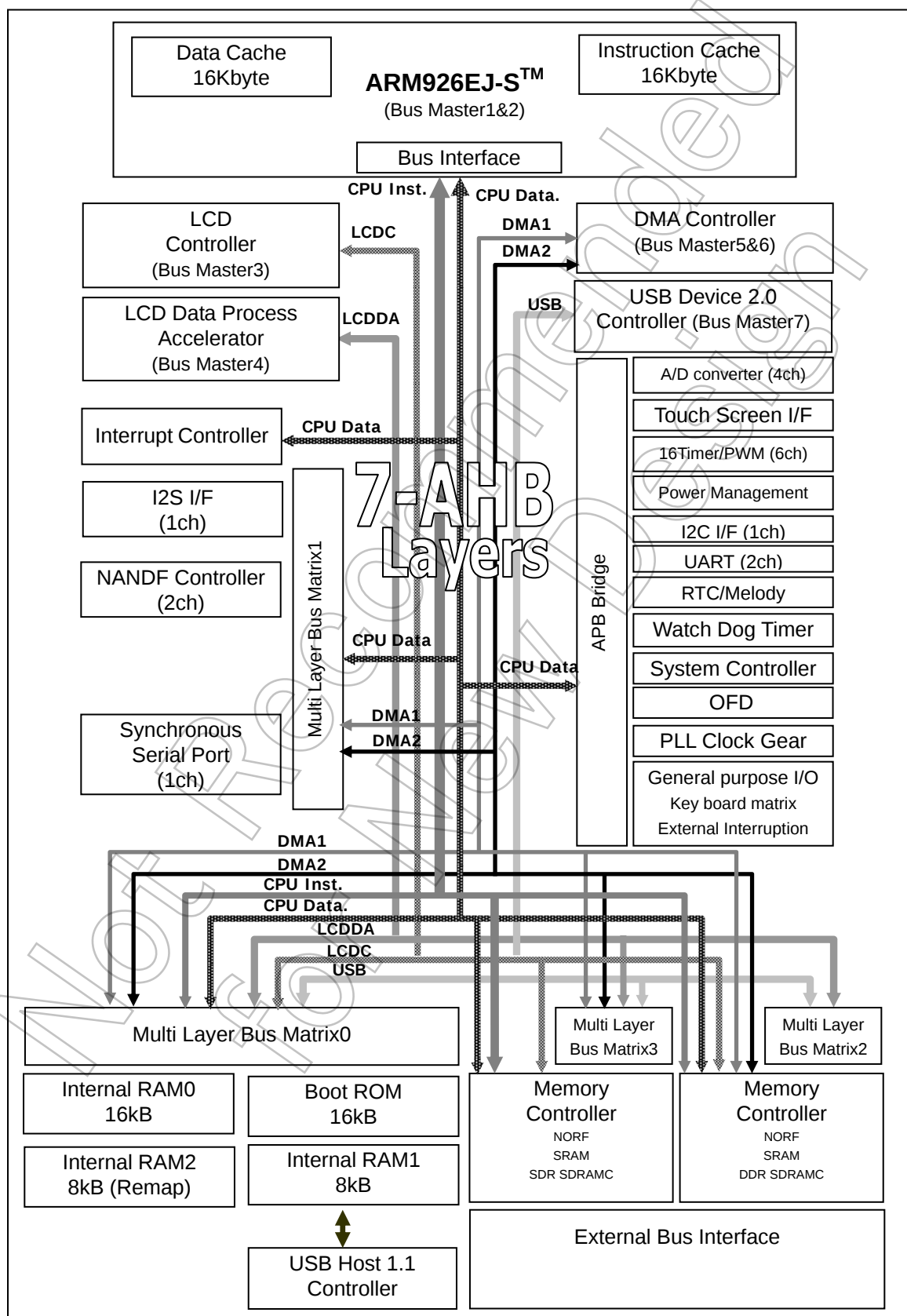
本 LSI には 7 種類の例外があり、それぞれに対応する特権処理モードがあります。
例外のタイプは次の 7 つです。下記の表に示します。

例外	アドレス	備考
リセット	0x00000000	
未定義命令の実行	0x00000004	
ソフトウェア割り込み (SWI) 命令	0x00000008	オペレーティングシステムコールに使用されます。
プリフェッチアポート	0x0000000C	命令フェッチメモリアポート
データアポート	0x00000010	データアクセスメモリアポート
IRQ	0x00000018	通常割り込み
FIQ	0x0000001C	高速割り込み

Not Recommended
for New Design

3.1.3 Multi Layer AHB

本 LSI では、7Layer 構造の Multilayer AHB バスシステムを採用しています。



3.2 JTAGインタフェース

3.2.1 仕様概要

TMPA901CMXBG には、Joint Test Action Group (JTAG) 規格に適合するインタフェースが用意されています。このインタフェースには業界標準の JTAG プロトコル (IEEE Std 1149.1・1990(Includes IEEE Std 1149.1a・1993)) が使われています。本章では、バウンダリスキャン、インタフェースで使われるピンと信号に触れながら、このインタフェースについて説明しています。

1) JTAG 規格バージョン

IEEE Std 1149.1・1990(Includes IEEE Std 1149.1a・1993)

2) JTAG 命令

標準命令(BYPASS、SAMPLE/PRELOAD、EXTEST) + HIGHZ 命令, CLAMP 命令を実装

3) IDCODE

なし

4) バウンダリスキャンレジスタ(BSR)対象外端子

a) 発振回路(SM0~3)

b) USB(SR0,SR1,SR3,SR4,SN6,SN7)

c) JTAG 制御端子(SN2、SP0~5)

d) 電源/GND(VREFH,VREFL を含む)

e) A/D 端子(PD4~7)

f) Touch Panel PX,PY (PD6,PD7)

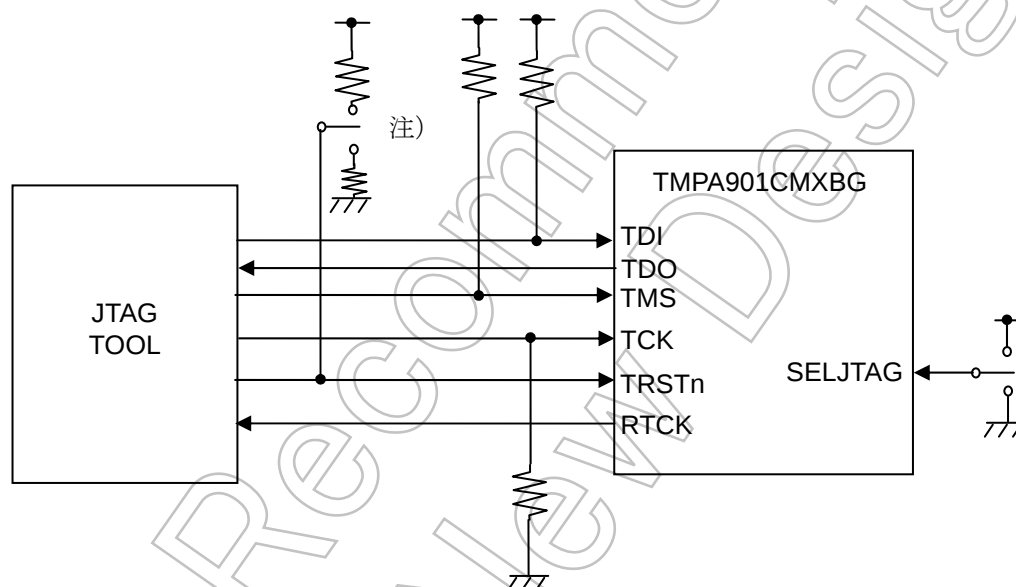
注) PR2 端子は、入出力端子ですが、出力用の BSR が接続されているため、SAMPLE/PRELOAD 命令にて端子の状態をキャプチャすることはできません。

3.2.2 信号の要約と接続例

JTAG インタフェース信号は次のとおりです。

- TDI JTAG シリアルデータ入力
- TDO JTAG シリアルデータ出力
- TMS JTAG テストモード選択
- TCK JTAG シリアルクロック入力
- TRSTn JTAG テストリセット入力
- RTCK JTAG テストフィードバックシリアルクロック出力
- SELJTAG ICE/JTAG テスト選択入力(準拠イネーブル信号) 0:ICE 1:JTAG

JTAG 対応の開発ツールを、JTAG インタフェースに接続し、デバッグをサポートします。
デバッグに関しては、使用する開発ツールの仕様を確認してください。



注) JTAG ツールを使用しない場合：TRSTn を永続的に“Low”に接続してください。

JTAG ツールを使用する場合：TRSTn を“Low”に保ってから“High”に遷移させてください。

使用する JTAG ツールによっては pull-up が内蔵されている場合があるので、外部の pull-down 抵抗値は開発ツールの仕様に合わせて設計してください。

モード設定端子	動作モード
SELJTAG	
0	Boundary Scan Mode 以外の時は 0 にしてください。 通常の Debug Mode として使用できます。 注) ただし、AM1 = 1, AM0 = 1 の内部 BOOT 起動時は Debug 出来ません。
1	Boundary Scan Mode として使用できます。

図 3.2.1 JTAG 開発ツールとの接続例

3.2.3 バウンダリスキャンの概要

絶えず高密度化していく集積回路 (IC)、表面実装デバイス、プリント回路基板 (PCB) に両面実装されるコンポーネント、および埋め込み穴の発達によって、内部基板とチップの接続という物理的接触に依存する内部回路テストはしだいに使いにくくなってきました。IC が高度に複雑化してきたため、こうしたチップをすみずみまで実行するテストは大規模化し、作成が難しくなってきました。

この難しさに対する解決策の 1 つとして開発されたのが、「バウンダリスキャン」回路です。バウンダリスキャン回路とは、各ピンとピンに接続されている IC の内部回路との間に設けられる一連のシフトレジスタのことです。通常、それらのバウンダリスキャンセルはバイパスされますが、IC がテストモードになると、テストプログラムの指示に従ってスキャンセルからシフトレジスタパスに沿ってデータが送られ、各種の診断テストが実行されます。テストのときには TCK, TMS, TDI, TDO, RTCK および TRSTn という 6 種類の信号が使われます。

JTAG バウンダリスキャンメカニズム (本章では「JTAG メカニズム」と呼びます) により、プロセッサ、プロセッサが接続されているプリント回路基板、および回路基板上の他のコンポーネントの間の接続をテストできます。JTAG メカニズムそのものには、プロセッサ自体をテストする機能はありません。

3.2.4 JTAG コントローラとレジスタ

プロセッサには、次の JTAG コントローラとレジスタが内蔵されています。

- 命令レジスタ
- バウンダリスキャンレジスタ
- バイパスレジスタ
- デバイス識別レジスタ
- テストアクセスポート (TAP) コントローラ

JTAGの基本動作は、TAPコントローラステートマシンがTMS入力信号をモニタすることです。実行が開始されると、TAPコントローラは実行されるテスト機能を決定します。これは、表 3.2.1に示すように、JTAG命令レジスタ (IR) のローディングとデータレジスタ (DR) を介してのシリアルデータスキャンの開始からなります。データがスキャンされる時、TMSピンの状態はそれぞれの新しいデータワードを示し、データの流れの最後を示します。選択されたデータレジスタは命令レジスタの内容で決められます。

3.2.5 命令レジスタ

JTAGの命令レジスタには、シフトレジスタを基本とする 4 個のセルが含まれています。このレジスタは、実施対象のテストとアクセスされるテストデータレジスタの両方またはその一方を選択するために使います。表 3.2.1の組み合わせに従って、バウンダリスキャンレジスタかバイパスレジスタが選択されます。

表 3.2.1 JTAG の命令レジスタのビット構成

命令コード 最上位→最下位	命令	選択されるデータレジスタ
0000	EXTEST	バウンダリスキャンレジスタ
0001	SAMPLE/PRELOAD	バウンダリスキャンレジスタ
0100 ~1110	予約	予約
0010	HIGHZ	バイパスレジスタ
0011	CLAMP	バイパスレジスタ
1111	BYPASS	バイパスレジスタ

命令レジスタのフォーマットは図 3.2.2のとおりです。



図 3.2.2 命令レジスタ

命令コードは、最下位ビットから命令レジスタにシフトされます。

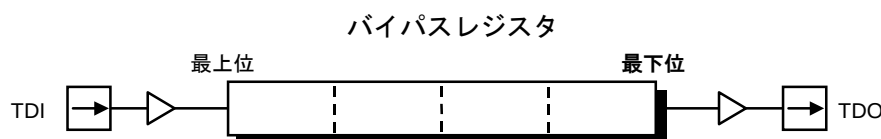


図 3.2.3 命令レジスタのシフト方向

バイパスレジスタは1ビット幅です。TAPコントローラがShift-DR (バイパス) 状態のとき、TDI ピンのデータはバイパスレジスタにシフトインされ、バイパスレジスタの出力は TDO 出力ピンにシフトアウトされます。

バイパスレジスタとは、簡単に言えば、特定のテストに必要でない基板レベルの直列バウンダリスキャンチェーン内のデバイスをバイパスできるようにする、迂回のための回路です。バウンダリスキャンチェーン内のバイパスレジスタの論理的な位置は図 3.2.4のとおりです。

バイパスレジスタを使用すれば、基板レベルテストのデータパス内でアクティブのままである IC 内のバウンダリスキャンレジスタへのアクセスが速くなります。

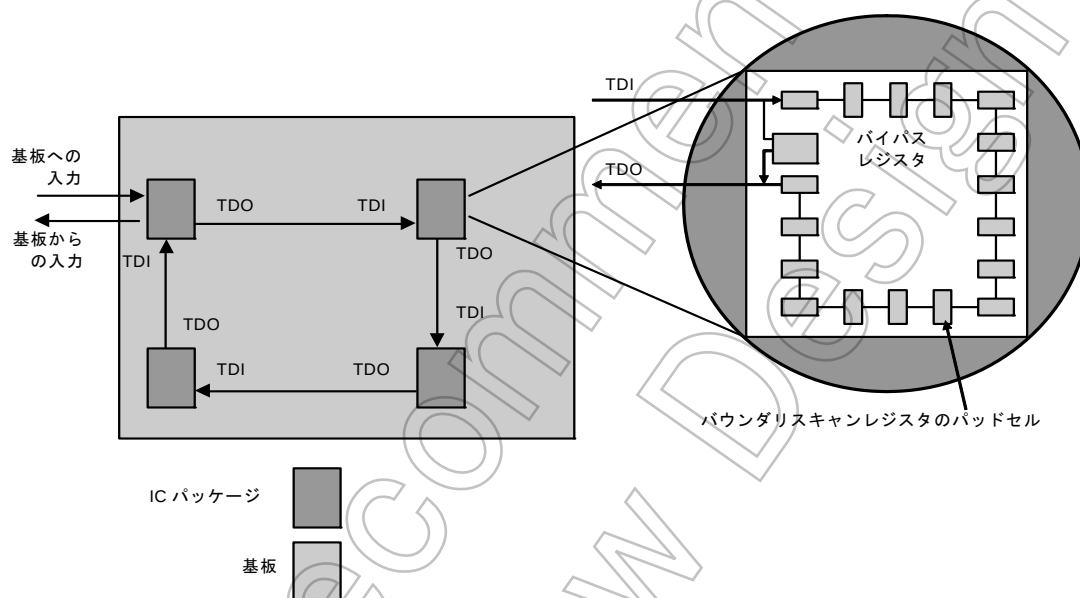


図 3.2.4 バイパスレジスタの機能

3.2.6 バウンダリスキャンレジスタ

バウンダリスキャンレジスタには、何本かのアナログ出力信号、制御信号を除くすべての TMPA901CM の入出力があります。TMPA901CM のピンは Shift-DR 状態からバウンダリスキャンレジスタの中へスキャンすることによって任意のパターンをドライブすることができます。プロセッサに入るデータは、バウンダリスキャンレジスタを許可して Capture-DR 状態のときにシフトすることにより検査されます。

バウンダリスキャンレジスタは、単一の 231 ビット幅のシフトレジスタを基本とするパスです。このパス内のセルは、TMPA901CM の入力パッドと出力パッドに接続されています。

TDI 入力はバウンダリスキャンレジスタの最下位ビット (LSB) に取り込まれ、バウンダリスキャンレジスタの最上位ビット (MSB) は TDO 出力から取り出されます。

3.2.7 テストアクセスポート (TAP)

テストアクセスポート (TAP) は、5 個の信号ピン TRSTn、TDI、TDO、TMS、および TCK で構成されます。直列のテストデータ、命令、および実施するテストの制御は、この 5 個の信号ピンを通じて送受信されます。

図 3.2.5 のように、データは 3 本のレジスタ (命令レジスタ、バイパスレジスタ、バウンダリスキャンレジスタ) のうちの 1 本に TDI ピンから直列にスキャンインされるか、またはその 3 本のレジスタの 1 本から TDO ピンにスキャンアウトされます。

TMS 入力は、主 TAP コントローラステートマシンの状態遷移を制御するものです。TCK 入力は直列 JTAG データが同期してシフトされるようにする専用のテストクロックであり、チップ固有クロックやシステムクロックには依存していません。

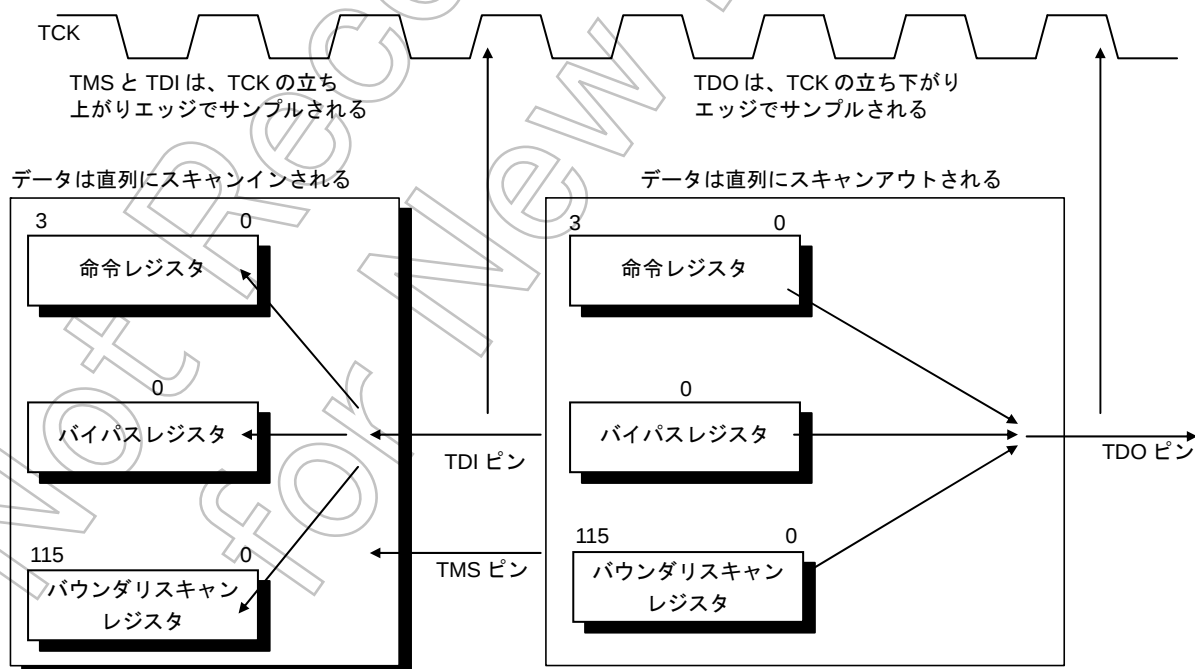


図3.2.5 JTAG テストアクセスポート

TDI ピンと TMS ピンのデータは、入力クロック信号 TCK の立ち上がりエッジでサンプルされます。TDO ピンのデータは、クロック信号 TCK の立ち下がりエッジで変化します。

3.2.8 TAPコントローラ

プロセッサには、IEEE JTAG 規格に規定されている 16 ステートの TAP コントローラが実現されています。

3.2.9 TAPコントローラのリセット

TAP コントローラのステートマシンは、次の方法によりリセット状態になります。

TRSTn 信号入力のアサート (Low) により、TAP コントローラはリセットされる。

プロセッサのリセット解除後 TCK 入力の立ち上がりエッジを連続 5 個使用して入力信号 TMS をアサートし続けます。TMS をアサート状態に保てば、リセット状態が保たれます。

3.2.10 コントローラの状態

図 3.2.6にTAPコントローラの状態遷移図を示します。TCKの立ち上がりエッジで、TMSが 0 か 1 のどちらの値を取るかによってTAPコントローラの状態が変化します。状態の遷移を示す矢印のわきにTMSの取る値を示します。

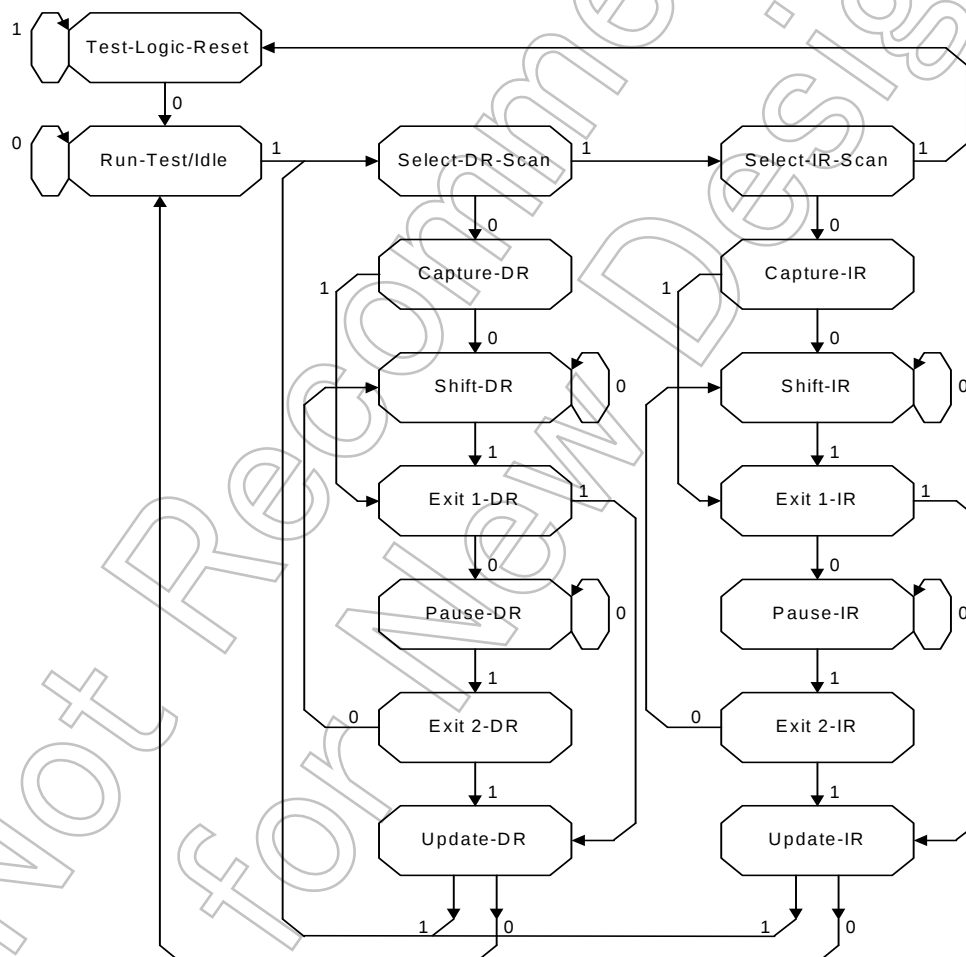


図 3.2.6 TAP コントローラの状態遷移図

以下コントローラの各状態について説明します。図 3.2.6の左側のカラムはデータカラムで右側のカラムは命令カラムです。データカラムと命令カラムはそれぞれデータレジスタ (DR)、命令レジスタ (IR) を表します。

- **Test-Logic-Reset**

TAP コントローラが **Reset** 状態の場合は、デフォルトとしてデバイス識別レジスタが選択されます。バウンダリスキャンレジスタの最上位 1 ビットは 0 にクリアされ、出力はディセーブルされます。

TMS が 1 の期間は、TAP コントローラは **Test-Logic-Reset** 状態を保持します。**Test-Logic-Reset** 状態のとき TMS に 0 を入力すると、TAP コントローラは **Run-Test/Idle** 状態に遷移します。

- **Run-Test/Idle**

Run-Test/Idle 状態では、組み込み自己テスト (BIST) 命令など特定の命令が設定されているときのみ、IC はテストモードになります。**Run-Test/Idle** 状態で実行されない命令に対しては、現在の命令により選択されているテストデータレジスタは前の状態を保持します。

TMS が「0」の期間は、TAP コントローラは **Run-Test/Idle** 状態を保持します。TMS に「1」を入力すると、TAP コントローラは **Select-DR-Scan** 状態に遷移します。

- **Select-DR-Scan**

Select-DR-Scan 状態は TAP コントローラの一時的な状態です。ここでは、IC が特別な動作をすることはありません。

TAP コントローラが **Select-DR-Scan** 状態のとき TMS に 0 を入力すると **Capture-DR** 状態に遷移します。TMS に 1 を入力すると命令カラムの **Select-IR-Scan** 状態に遷移します。

- **Select-IR-Scan**

Select-IR-Scan 状態は TAP コントローラの一時的な状態です。ここでは IC が特別な動作をすることはありません。

TAP コントローラが **Select-IR-Scan** 状態のとき、TMS に 0 を入力すると **Capture-IR** 状態に遷移します。TMS に 1 を入力すると TAP コントローラは **Test-Logic-Reset** 状態に戻ります。

- **Capture-DR**

TAP コントローラが **Capture-DR** 状態のとき、命令レジスタによって選択されたデータレジスタがパラレル入力をもっている場合、データがデータレジスタにパラレルにロードされます。データレジスタにパラレル入力がない場合、あるいは選択されたテストデータレジスタにデータをロードする必要がない場合は、データレジスタは前の状態を保持します。

TAP コントローラが **Capture-DR** 状態のとき TMS に 0 を入力すると、**Shift-DR** 状態に遷移します。TMS に 1 を入力すると TAP コントローラは **Exit 1-DR** 状態に遷移します。

- Shift-DR

TAP コントローラが Shift-DR 状態のとき、TDI-TDO 間に接続されたデータレジスタはシリアルにデータをシフトアウトします。

TAP コントローラが Shift-DR 状態のとき、TMS が 0 のあいだ Shift-DR 状態を保持します。TMS に 1 を入力すると TAP コントローラは Exit 1-DR 状態に遷移します。

- Exit 1-DR

Exit 1-DR 状態は TAP コントローラの一時的な状態です。

TAP コントローラが Exit 1-DR 状態のとき、TMS に 0 を入力すると Pause-DR 状態に遷移します。TMS に 1 を入力すると Update-DR 状態に遷移します。

- Pause-DR

Pause-DR 状態は命令レジスタによって選択されたデータレジスタのシフト動作を一時的に停止します。命令レジスタ、データレジスタは現在の状態を保持します。

TAP コントローラが Pause-DR 状態のとき、TMS が 0 のあいだ、この状態を保持します。TMS に 1 を入力すると Exit 2-DR 状態に遷移します。

- Exit 2-DR

Exit 2-DR 状態は TAP コントローラの一時的な状態です。

TAP コントローラが Exit 2-DR 状態のとき、TMS に 0 を入力すると、Shift-DR 状態に戻ります。TMS に 1 を入力すると Update-DR 状態に遷移します。

- Update-DR

Update-DR 状態では、TCK の立ち上がりエッジに同期してパラレル出力をもっているレジスタからデータをパラレルに出力します。パラレル出力ラッチをもっているデータレジスタはシフト中にデータを出力することではなく、この状態でのみデータを出力します。

TAP コントローラが Update-DR 状態のとき TMS に 0 を入力すると Run-Test/Idle 状態に遷移します。TMS に 1 を入力すると Select-DR-Scan 状態に遷移します。

- Capture-IR

Capture-IR 状態ではデータは命令レジスタにパラレルにロードされます。ロードされるデータは 0y0001 です。Capture-IR 状態は命令レジスタのテストに使用します。命令レジスタの故障はロードされたデータをシフトアウトすることにより検出できます。

TAP コントローラが Capture-IR 状態のとき TMS に 0 を入力すると Shift-IR 状態に遷移します。TMS に 1 を入力すると Exit 1-IR 状態に遷移します。

- Shift-IR

Shift-IR 状態では、命令レジスタが TDI-TDO 間に接続され、TCK の立ち上がりエッジに同期してロードされたデータをシリアルにシフトアウトします。

TAP コントローラが Shift-IR 状態のとき TMS が 0 のあいだ、この状態を保持します。TMS に 1 を入力すると、Exit 1-IR 状態に遷移します。

- Exit 1-IR

Exit 1-IR 状態は TAP コントローラの一時的な状態です。

TAP コントローラが Exit 1-IR 状態のとき TMS に 0 を入力すると、Pause-IR 状態に遷移します。TMS に 1 を入力すると Update-IR 状態に遷移します。

- Pause-IR

Pause-IR 状態は命令レジスタのシフト動作を一時的に停止する状態です。命令レジスタとデータレジスタはそのままの状態を保持します。

TAP コントローラが Pause-IR 状態のとき、TMS が 0 のあいだ、この状態を保持します。TMS に 1 を入力すると Exit 2-IR 状態に遷移します。

- Exit 2-IR

Exit 2-IR 状態は TAP コントローラの一時的な状態です。

TAP コントローラが Exit 2-IR 状態のとき、TMS に 0 を入力すると、Shift-IR 状態に遷移します。TMS に 1 を入力すると Update-IR 状態に遷移します。

- Update-IR

Update-IR 状態は命令レジスタにシフトされた命令を TCK の立ち上がりエッジに同期してパラレルに出力し、命令を更新します。

TAP コントローラが Update-IR 状態のとき、TMS に 0 を入力すると、Run-Test/Idle 状態に遷移します。TMS に 1 を入力すると、Select-DR-Scan 状態に遷移します。

3.2.11 バウンダリスキャン順序

プロセッサ信号に対するバウンダリスキャン順序は、表 3.2.2 のとおりです。

TDI → 1 (PC6) → 2(PC7) → ... → 180(PC4) → 181(PC2) → TDO

表 3.2.2 TMPA901CM プロセッサのピンに対する JTAG スキャン順序

番号	端子名	番号	端子名	番号	端子名	番号	端子名	番号	端子名
	TDI								
1	PC6	41	PT2	81	SH2	121	SK1	161	PV4
2	PC7	42	PT5	82	SJ1	122	TSBRVS66	162	PU1
3	TSBRVS00	43	PB3	83	TSBRVS50	123	SL0	163	TSBRVS71
4	TSBRVS01	44	SM4	84	SB0	124	SL1	164	PU5
5	TSBRVS02	45	PA2	85	SJ6	125	TSBRVS67	165	TSBRVS72
6	TSBRVS03	46	PT3	86	TSBRVS51	126	SL6	166	PV1
7	TSBRVS04	47	PA0	87	SA0	127	SL5	167	PU2
8	TSBRVS05	48	PT6	88	SB1	128	SK5	168	PV5
9	TSBRVS06	49	PT0	89	TSBRVS52	129	TSBRVS68	169	PU6
10	TSBRVS07	50	PA1	90	SA1	130	SL2	170	PV2
11	TSBRVS08	51	PT4	91	SH3	131	SL4	171	PU3
12	TSBRVS09	52	PT1	92	TSBRVS53	132	SE6	172	TSBRVS73
13	TSBRVS10	53	PB1	93	SB2	133	SF4	173	PU7
14	TSBRVS11	54	PA3	94	SA2	134	SE0	174	PV6
15	TSBRVS12	55	PB0	95	TSBRVS54	135	SE7	175	PV3
16	TSBRVS13	56	PN0	96	TSBRVS55	136	SG1	176	TSBRVS74
17	TSBRVS14	57	PB2	97	TSBRVS56	137	SE1	177	PC3
18	TSBRVS15	58	TSBRVS38	98	SB3	138	SF5	178	TSBRVS75
19	TSBRVS16	59	PN1	99	SA3	139	SG5	179	PV7
20	TSBRVS17	60	TSBRVS39	100	SH4	140	SF0	180	PC4
21	TSBRVS18	61	TSBRVS40	101	TSBRVS57	141	SG2	181	PC2
22	TSBRVS19	62	SM6	102	TSBRVS58	142	SE2		TDO
23	TSBRVS20	63	SM7	103	TSBRVS59	143	SF6		
24	TSBRVS21	64	PT7	104	SB4	144	TSBRVS69		
25	TSBRVS22	65	SN0	105	SA4	145	SF1		
26	TSBRVS23	66	SN1	106	SA5	146	SG6		
27	TSBRVS24	67	TSBRVS41	107	SB5	147	SE3		
28	TSBRVS25	68	TSBRVS42	108	TSBRVS60	148	SG3		
29	TSBRVS26	69	TSBRVS43	109	TSBRVS61	149	SF7		
30	TSBRVS27	70	TSBRVS44	110	TSBRVS62	150	SF2		
31	TSBRVS28	71	TSBRVS45	111	SA6	151	SE4		
32	TSBRVS29	72	TSBRVS46	112	SB6	152	TSBRVS70		
33	TSBRVS30	73	TSBRVS47	113	SH7	153	SG7		
34	TSBRVS31	74	SJ2	114	TSBRVS63	154	SG4		
35	TSBRVS32	75	SJ4	115	SA7	155	SG0		
36	TSBRVS33	76	TSBRVS48	116	TSBRVS64	156	SF3		
37	TSBRVS34	77	SJ0	117	SB7	157	SE5		
38	TSBRVS35	78	SJ5	118	SK0	158	PU0		
39	TSBRVS36	79	SJ3	119	TSBRVS65	159	PU4		
40	TSBRVS37	80	TSBRVS49	120	SK4	160	PV0		

注) バウンダリスキャン順序の TSBRVS[00:75]は Reserved 端子となります。

3.2.12 JTAGコントローラセルでサポートしている命令

この項では、TMPA901CM の JTAG コントローラセルでサポートしている命令について説明します。

(1) EXTEST 命令

EXTEST 命令は外部接続テストに使用します。EXTEST 命令では、出力端子の BSR セルは Update-DR 時にテストパターンを出力し、入力端子の BSR セルは Capture-DR 時にテスト結果を取り込みます。

通常、EXTEST 命令を選択するまえに SAMPLE/PRELOAD 命令を使ってバウンダリスキャンレジスタを初期化します。バウンダリスキャンレジスタを初期化しておかないと、Update-DR 状態において不確定なデータが伝送され、IC 間でバスのコンフリクトが起こる可能性があります。EXTEST 命令が選択されているあいだのデータの流れを図 3.2.7 に示します。

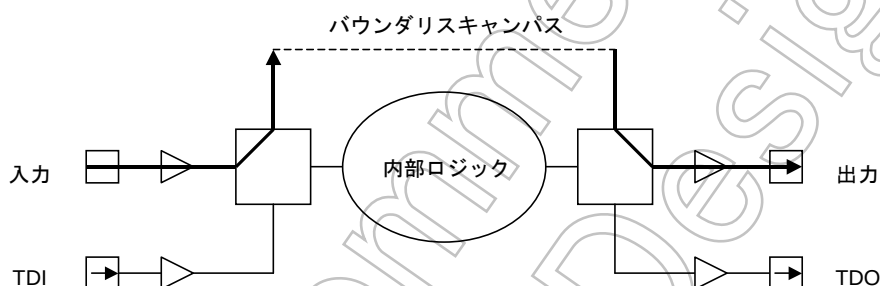


図 3.2.7 EXTEST 命令が選択されているときのテストデータの流れ

次に外部接続テストの基本的なテスト手順を示します。

1. TAP コントローラを初期化して、Test-Logic-Reset 状態にします。
2. 命令レジスタに SAMPLE/PRELOAD 命令をロードします。これによりバウンダリスキャンレジスタが TDI-TDO 間に接続されます。
3. 確定したデータをシフトインすることにより、バウンダリスキャンレジスタを初期化します。
4. 最初のテストデータをバウンダリスキャンレジスタにロードします。
5. 命令レジスタに EXTEST 命令をロードします。
6. 入力端子に印加されているデータを入力用バウンダリスキャンレジスタに取り込みます。
7. 取り込んだデータをシフトアウトすると同時に、次のテストパターンをシフトインします。
8. 出力用バウンダリスキャンレジスタにシフトインされたテストパターンを出力端子に出力します。

6 から 8 をテストパターンごとに繰り返します。

(2) SAMPLE/PRELOAD 命令

この命令は TDI-TDO 間をバウンダリスキャンレジスタで接続します。名前が示すとおり、SAMPLE/PRELOAD 命令には次の 2 つの機能があります。

SAMPLEはICのI/Oパッドを観測するのに使います。SAMPLEがI/Oパッドを観測しているあいだ、内部ロジックはICのI/O端子から切り離されません。SAMPLEはCapture-DR状態で実行します。通常動作中、TCKの立ち上がりエッジにおいてICのI/O端子の値を読み取ることがSAMPLEの主な用途です。図 3.2.8にSAMPLE/PRELOAD命令のSAMPLEを実行しているあいだのデータの流れを示します。

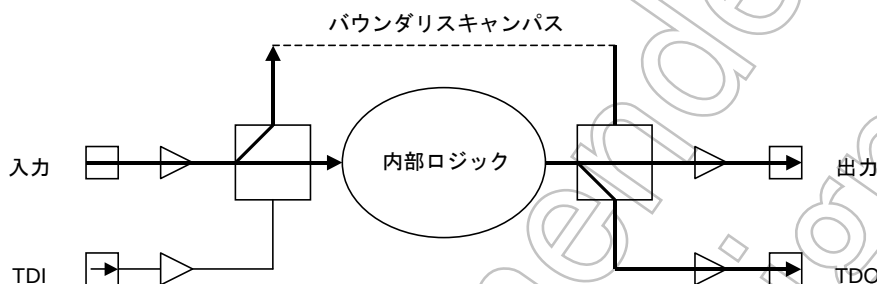


図 3.2.8 SAMPLE が選択されているときのテストデータの流れ

PRELOADは他の命令を選択するまえに、バウンダリスキャンレジスタを初期化するのに使います。例えば、前に述べたようにEXTEST命令を選択するまえにPRELOADを用いてバウンダリスキャンレジスタを初期化します。PRELOADはシステムロジックの通常動作に影響を与えずに、バウンダリスキャンレジスタにデータをシフトします。図 3.2.9にSAMPLE/PRELOAD命令のPRELOADを実行しているあいだのデータの流れを示します。

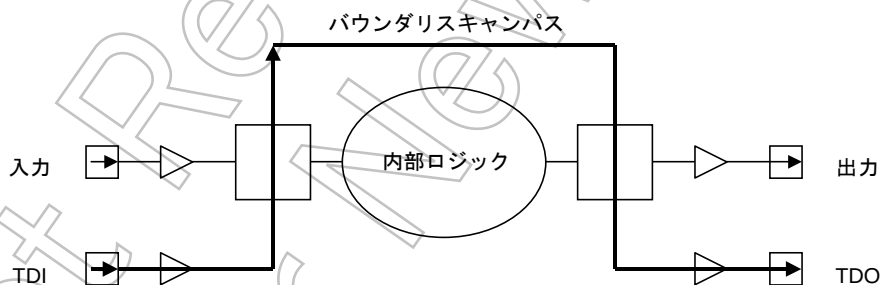


図 3.2.9 PRELOAD が選択されているときのテストデータの流れ

(3) BYPASS 命令

BYPASS命令はICを制御、観測する必要がないテストの場合に、バイパスレジスタをJTDI-JTDO間に接続することによりICをバイパスする最短のシリアルパスを構成します。BYPASS命令はチップ上のシステムロジックの通常動作には影響を与えません。図 3.2.10 に示すようにBYPASS命令が選択されているあいだ、データはバイパスレジスタを通ります。

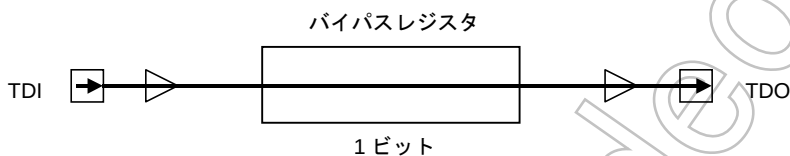


図 3.2.10 バイパスレジスタが選択されているときのテストデータの流れ

(4) CLAMP 命令

CLAMPはPreload命令によって設定されたバウンダリスキャンレジスタの値を出力し、かつバイパス動作を行います。CLAMP命令はTDIとTDO間にBypassレジスタを選択します。

(5) HIGHZ 命令

HIGHZ命令は内部論理回路からの出力をDisable状態にします。HIGHZ命令が実行されると、3ステート出力をハイ・インピーダンス状態にします。

HIGHZ命令もTDIとTDO間にBypassレジスタを選択します。

- 注意事項

本節では、当プロセッサで採用しているJTAGバウンダリスキャン処理の注意点について説明します。

1) PR2端子は、入出力端子ですが、出力用のBSRが接続されているため、SAMPLE/PRELOAD命令にて端子の状態をキャプチャすることはできません。

2) JTAG回路のリセット解除は下記の2種類のどちらかの手順を選択してください。

TRST_nをアサートしてJTAG回路を初期化後デアサート。

TMS端子=1の状態、TCKに5クロック以上供給

3.3 メモリマップ

TMPA901CM のメモリマップを示します。

表 3.3.1 各内蔵エリアへアクセスの概要

項目	アクセス概要	
CPU アドレスバス幅	32 ビット	
CPU データバス幅	32 ビット	
内部動作周波数	Max 200MHz @ 0 to 70°C Max 150MHz @ -20 to 85°C	
最小バスサイクル	1- f_{CLK} クロックアクセス(5ns@200MHz)	
内蔵 RAM	32-bit 1-HCLK クロック アクセス	
内蔵 Boot ROM	32-bit 1-HCLK クロック アクセス	
内蔵 I/O	32-bit,1-HCLK クロック アクセス	LCDC, LCDDA, INTC, DMAC, USB Device, USB Host, I ² S, NANDFC, SSP, MPMC
	32-bit,2-PCLK クロック アクセス	A/D C, TSI, Timer/PWM, PMC, I ² C, UART, RTC, WDT, OFD, System C, PLL CG, GPIO

Start Address	内部ブート ROM 起動		外部メモリ起動	
0x0000_0000	内蔵 ROM : 8KB+ 8KB	Remap エリア(8KB)	SMCCS0n	
0x0000_2000				
0x0000_4000	SMCCS0n	外部エリア(15.8MB)		
0x0100_0000	未使用エリア	外部エリア(512MB)	未使用エリア	
0x2000_0000				
0x2100_0000	SMCCS0n	外部エリア(496MB)	SMCCS0n	
0x4000_0000	DMCCSn	外部エリア(512MB)	DMCCSn	
0x6000_0000	SMCCS1n	外部エリア(512MB)	SMCCS1n	
0x8000_0000	未使用エリア	外部エリア(512MB)	未使用エリア	
0xA000_0000	未使用エリア	外部エリア(512MB)	未使用エリア	
0xC000_0000	未使用エリア	外部エリア(512MB)	未使用エリア	
0xE000_0000	未使用エリア	外部エリア(256MB)	未使用エリア	
0xF000_0000	内蔵 IO-0 (APB) : 1MB	内蔵 I/O エリア (128MB)	内蔵 IO-0 (APB) : 1MB	
0xF010_0000	未使用エリア		未使用エリア	
0xF080_0000	内蔵 IO-1 (APB Port1/2) : 1MB		内蔵 IO-1 (APB Port1/2) : 1MB	
0xF090_0000	内蔵 IO-2 (APB Port2/2) : 1MB		内蔵 IO-2 (APB Port2/2) : 1MB	
0xF0A0_0000	未使用エリア		未使用エリア	
0xF200_0000	内蔵 IO-3 (AHB+APB) : 16MB		内蔵 IO-3 (AHB+APB) : 16MB	
0xF300_0000	未使用エリア		未使用エリア	
0xF400_0000	内蔵 IO-4 (AHB) : 16MB		内蔵 IO-4 (AHB) : 16MB	
0xF600_0000	未使用エリア		未使用エリア	
0xF800_0000	未使用エリア		未使用エリア	
0xF800_2000	内蔵 RAM-3 : 8KB(Remap)		内蔵 RAM-3 : 8KB(Remap)	
0xF800_4000	内蔵 RAM-0 : 16KB		内蔵 RAM-0 : 16KB	
0xF800_8000	内蔵 RAM-1 : 8KB		内蔵 RAM-1 : 8KB	
0xF800_A000	未使用エリア		内蔵メモリエリア (128MB)	未使用エリア

0xFFFF_FFFF

注 1) 0x0000_0000~0x0000_1FFF(8KB)の空間は Remap エリアとなっており、Remap_ON 設定で内蔵 RAM3 エリアへのアクセスに切り替わります(F8000_2000 へのアクセス時も RAM3 エリアを示します)。

注 2) 未使用エリアのアクセスは禁止です。

図 3.3.1 メモリマップ(起動モード・外部エリア・内蔵エリア詳細)

Start Address	内部ブート ROM 起動		Bus Master と Slave 接続関係 ○ : アクセス可, × : アクセス不可, - : アクセス禁止						
			CPU(D)	CPU(I)	LCDC	LCDDA	DMA1	DMA2	USB
			M1	M2	M3	M4	M5	M6	M7
0x0000_0000	内蔵 ROM : 8KB+ 8KB	Remap エリア (8KB)	○	○	×	×	○	○	×
0x0000_2000									
0x0000_4000	SMCCS0n	外部エリア (15.8MB)	○	○	○	○	○	○	○
0x0100_0000	未使用エリア	外部エリア (512MB)	-						
0x2000_0000									
0x2100_0000	SMCCS0n	外部エリア (496MB)	○	○	○	○	○	○	○
0x4000_0000	DMCCS0n	外部エリア (512MB)	○	○	○	○	○	○	○
0x6000_0000	SMCCS1n	外部エリア (512MB)	○	○	○	○	○	○	○
0x8000_0000	未使用エリア	外部エリア (1792MB)	-						
0xF000_0000	内蔵 IO-0 (APB) : 1MB	内蔵 I/O エリア (128MB)	アクセスについて次ページを参照してください						
0xF010_0000	未使用エリア								
0xF080_0000	内蔵 IO-1 (APB Port1/2) : 1MB								
0xF090_0000	内蔵 IO-2 (APB Port2/2) : 1MB								
0xF0A0_0000	未使用エリア								
0xF200_0000	内蔵 IO-3 (AHB+APB) : 16MB								
0xF300_0000	未使用エリア								
0xF400_0000	内蔵 IO-4 (AHB) : 16MB								
0xF600_0000	未使用エリア								
0xF800_0000	未使用エリア								
0xF800_2000	内蔵 RAM-3 : 8KB (Remap)	内蔵メモリエリア (128MB)	○	○	○	○	○	○	○
0xF800_4000	内蔵 RAM-0 : 16KB LCDDA 兼用 Dual port RAM		○	○	○	○	○	○	○
0xF800_8000	内蔵 RAM-1 : 8KB (USB Host 使用エリア)		○	○	○	○	○	○	○
0xF800_A000	未使用エリア		-						

0xFFFF_FFFF

注 : USB Host は 0xF800_8000 ~ 0xF800_9FFF のエリアのみをアクセスできます。

図 3.3.2 メモリマップ(内蔵エリア詳細・Bus Master と Slave 接続関係)

スタートアドレス	エンドアドレス	内蔵 IO 詳細		アクセス可能マスタ	
0xF000_0000	0xF000_0FFF	内蔵 IO(APB) 1MB	SysCtrl	M1(CPU Data)	内蔵 IO エリア
0xF001_0000	0xF001_0FFF		WDT		
0xF002_0000	0xF002_0FFF		PMC		
0xF003_0000	0xF003_0FFF		RTC		
0xF004_0000	0xF004_0FFF		Timer01/PWM		
0xF004_1000	0xF004_1FFF		Timer23/PWM		
0xF004_2000	0xF004_2FFF		Timer45		
0xF005_0000	0xF005_0FFF		PLLCG		
0xF006_0000	0xF006_0FFF		TSI		
0xF007_0000	0xF007_0FFF		I ² C0		
0xF007_1000	0xF007_1FFF		Reserved		
0xF008_0000	0xF008_0FFF		ADC		
0xF009_0000	0xF009_0FFF		OFD		
0xF00A_0000	0xF00A_0FFF		EBI		
0xF00B_0000	0xF00B_0FFF		LCDOP		
0xF080_0000	0xF080_FFFF	内蔵 IO(APB) 1MB	PORT	M1(CPU Data)	
0xF200_0000	0xF200_1FFF	内蔵 IO(AHB+APB) 16MB	UART0,1 注2)	M1(CPU Data) M5(DMAC1) M6(DMAC2)	
0xF200_2000	0xF200_3FFF		SSP		
0xF200_4000	0xF200_4FFF		Reserved		
0xF201_0000	0xF201_0FFF		NANDFC		
0xF202_0000	0xF202_0FFF		Reserved		
0xF203_0000	0xF203_0FFF		Reserved		
0xF204_0000	0xF204_0FFF		I ² S		
0xF205_0000	0xF205_0FFF	LCDDA	M1(CPU Data)		
0xF400_0000	0xF400_0FFF	INTC			
0xF410_0000	0xF410_0FFF	DMAC			
0xF420_0000	0xF420_0FFF	LCDC			
0xF430_0000	0xF430_0FFF	MPMC0			
0xF431_0000	0xF431_0FFF	MPMC1			
0xF440_0000	0xF440_0FFF	USB Device			
0xF450_0000	0xF450_F000	USB Host			

注 1) 上記に割りつけられたアドレス以外は Reserved エリアです。アクセスしないでください。

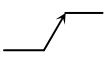
注 2) UART1 は DMAC 転送をサポートしていません。

図 3.3.3 メモリマップ(内部レジスタ詳細)

3.3.1 起動モード

本マイコンでは、外部の端子設定にて、いくつかの起動モードを選択することが出来ます。

1. 起動メモリ設定

モード設定端子			動作モード
RESETn	AM1	AM0	
	0	1	外部 16bit NOR Flash メモリから起動(内蔵ブート ROM は見えません)
	1	0	外部 32bit NOR Flash メモリから起動(内蔵ブート ROM は見えません)
	1	1	BOOT (内蔵ブート ROM から起動)
	0	0	TEST (設定禁止)

2. 外部メモリ電圧設定(NANDF を除く)

モード設定端子	動作モード
SELDVCCM	
0	メモリ系の制御端子は $1.8 \pm 0.1V$ (DVCCM) 動作
1	メモリ系の制御端子は $3.3 \pm 0.3V$ (DVCCM) 動作

3. 外部メモリコントローラ設定

モード設定端子	動作モード
SELMEMC	
0	利用できる SDRAM は SDR(Single Data Rate)および、Mobile SDR タイプのみ
1	利用できる SDRAM は LVCMOS DDR(LVCMOS Double Data Rate)タイプのみ

4. JTAG 端子設定

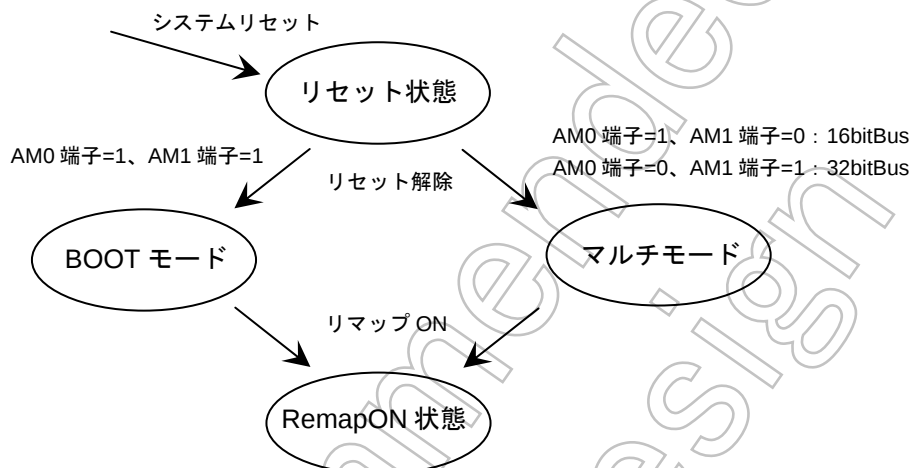
モード設定端子	動作モード
SELJTAG	
0	Boundary Scan Mode 以外の時は"0"にしてください。 通常の Debug Mode として使用できます。 注) ただし、AM1 = "1", AM0 = "1" の内部 BOOT 起動時は Debug 出来ません。
1	Boundary Scan Mode として使用できます。

3.4 システムコントローラ

3.4.1 リマップ機能

本 LSI では、リマップ機能により、内蔵 RAM8K バイトの領域を 2 箇所のメモリ領域 (0x0000_0000~0x0000_1FFF および 0xF800_2000~0xF800_3FFF) からアクセスすることが可能となります。

リマップ機能は Remap<REMAP>をライトすることによって ON になります。



注) Remap ON 状態へは、レジスタ設定で移行しますが、Remap OFF 状態へはシステムリセットか、PCM 状態からの解除でしか移行することは出来ません。

図 3.4.1 メモリ空間状態遷移図

ブートモード

Remap_ON

Multi Mode

0x0000_0000	内蔵 ROM 16KB	内蔵 RAM-3 8KB(Remap)	外部エリア SMCCS0n
0x0000_2000		使用禁止	
0x0000_4000	未使用領域	未使用領域	未使用領域
0x2100_0000	外部エリア	外部エリア	外部エリア
0xF000_0000	内蔵 IO エリア	内蔵 IO エリア	内蔵 IO エリア
0xF800_0000			
0xF800_2000	内蔵 RAM-3 : 8KB(Remap)	内蔵 RAM-3 : 8KB(Remap)	内蔵 RAM-3 : 8KB(Remap)
0xF800_4000	内蔵 RAM-0 : 16KB	内蔵 RAM-0 : 16KB	内蔵 RAM-0 : 16KB
0xF800_8000	内蔵 RAM-1 : 8KB	内蔵 RAM-1 : 8KB	内蔵 RAM-1 : 8KB
0xF800_A000	未使用エリア	未使用エリア	未使用エリア
0xF801_0000	未使用エリア	未使用エリア	未使用エリア
0xFFFF_FFFF			

注) 0x0000_0000~0x0000_1FFF(8KB)の空間は Remap エリアとなっており、Remap_ON 設定で内蔵 RAM3 エリアへのアクセスに切り替わります(0xF800_2000 へのアクセス時も RAM3 エリアを示します)。

図 3.4.2 メモリマップ(起動モード・外部エリア詳細)

3.4.2 レジスタの説明

SFR のリストと機能を以下に示します。

base アドレス= 0F000_0000

Register Name	Address (base+)	Description
Remap	0x0004	Reset memory map (REMAP)

1. Remap Register

Address = (0xF000_0000) + 0x0004

Bit	Bit Symbol	Type	Reset Value	Description
[31:1]	—	—	Undefined	Read undefined. Write as zero.
[0]	REMAP	RW	0y0	REMAP 設定

(個別説明)

a. <REMAP>

REMAP 機能を有効にするレジスタです。

任意のデータを書き込むことで、内蔵 RAM3 をメモリマップの先頭からもアクセスできるようになります。レジスタ設定で、リマップ状態を OFF (初期状態) することは出来ません。

3.5 クロックコントローラ

3.5.1 概要

クロックコントローラはMCU全体のクロック制御回路です。以下の特長があります。

- クロック逡倍回路(PLL)を使用し、最大 200MHz のクロックを CPU に供給。逡倍数は 1 倍、6 倍、8 倍をダイナミックに選択可能。
- クロックギアによる 消費電流削減機能
- クロックコントローラ内のレジスタへのライト禁止機能

以下に、クロックの動作モードの状態遷移図を示します。

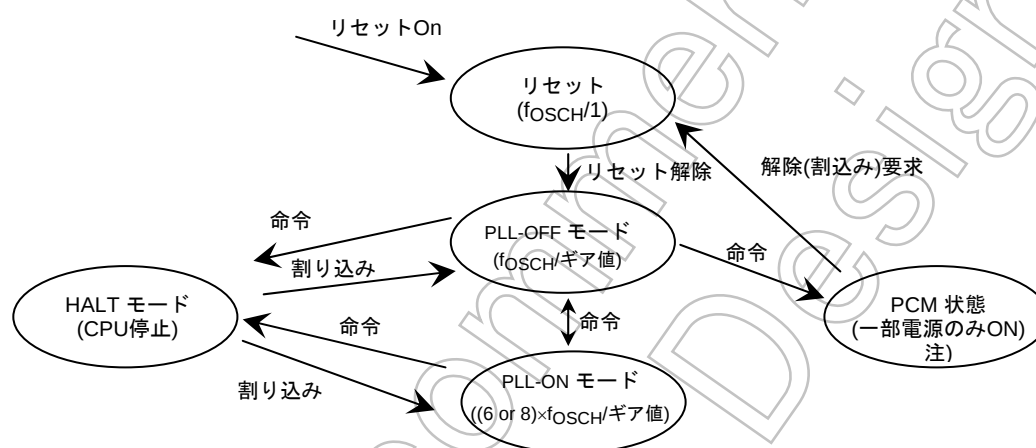
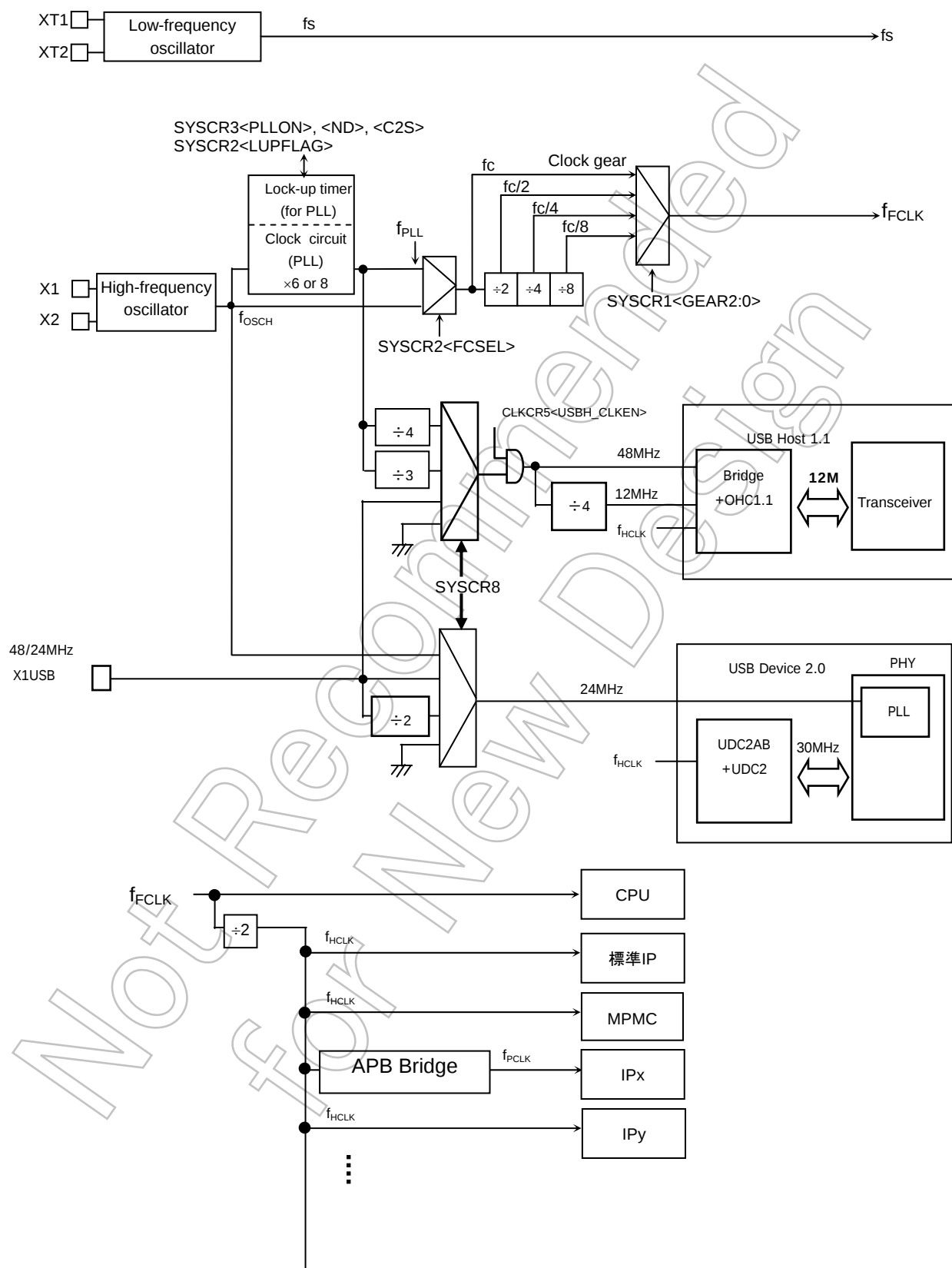


図 3.5.1 クロックモード状態遷移図

注) PCMモードに関しては第26章（電源管理回路）を参照願います。

3.5.2 ブロック図



X1,X2 端子より入力されるクロック周波数を f_{OSCH} 、XT1,XT2 端子より入力されるクロック周波数を f_S 、SYSCR1<GEAR2:0>で選択したクロックが CPU コア用のクロック f_{CLK} と定義します。AHB バスに接続する周辺 IP は f_{CLK} を 2 分周したクロックで f_{HCLK} と定義します (信号名 : HCLK)。APB バスに接続する周辺 IP は f_{CLK} を 2 分周したクロックで f_{PCLK} と定義します (信号名 : PCLK)。

また、メモリコントローラには、DRAM 用、SRAM/NORF 用の 2 系統のクロックが入力されており、SRAM/NORF 用のクロックは、 f_{HCLK} と、 f_{HCLK} を 2 分周したクロックを選択可能です (詳細は MPMC の章を参照ください)。

各クロックについての使用上の制約を以下に定義します。その制約を満たした上で用途にあわせて使用クロックを決定してください。

表 3.5.1 各クロックについての制約 @ $T_a = 0 \text{ to } 70^\circ\text{C}$

	最低周波数	最高周波数	注意事項
(a) f_{OSCH} (高速発振器周波数)	10MHz	27MHz	
(b) f_{PLL} (PLL 出力周波数)	60MHz	200MHz	
(c) f_{CLK} (CPU 用周波数)	1.25MHz	200MHz	
(d) f_{USB} (USB Device 用周波数)	24MHz	24MHz	24MHz $\pm$ 100 ppm の精度が必要です。
(e) f_{USB} (USB Host 用周波数)	48MHz	48MHz	48MHz $\pm$ 100 ppm の精度が必要です。
(f) f_S (低速発振器周波数)	30kHz	34kHz	

上記を満たした推奨の使用例を以下の表に定義します。

Table 3.5.2 各クロックについての制約 @ $-20 \text{ to } 85^\circ\text{C}$

	Lowest frequency	Highest frequency	Notes
(a) f_{OSCH} (高速発振器周波数)	10 MHz	27 MHz	
(b) f_{PLL} (PLL 出力周波数)	60 MHz	150MHz	
(c) f_{CLK} (CPU 用周波数)	1.25 MHz	150 MHz	
(d) f_{USB} (USB 用周波数)	24 MHz	24 MHz	24MHz $\pm$ 100 ppm の精度が必要です。
(e) f_{USB} (USB Host 用周波数)	48MHz	48MHz	48MHz $\pm$ 100 ppm の精度が必要です。
(f) f_S (低速発振器周波数)	30 kHz	34 kHz	

表 3.5.2 推奨使用例 @ Ta = 0 to 70°C

	高速発振: f_{OSCH}	PLL 出力 クロック: f_{PLL}	CPU 用 クロック: f_{FCLK}	USB 用 クロック: f_{USB}
(a) USB 使用, CPU 最大 192MHz	24MHz	最大 192MHz	最大 192MHz	24MHz / 48MHz
(b) USB 使用, CPU 最大 200MHz	25MHz	最大 200MHz	最大 200MHz	24MHz / 48MHz (X1USB 端子より 入力要)
(c) USB 不要 CPU 最大 200MHz	25MHz	最大 200MHz	最大 200MHz	—

表 3.5.3 推奨使用例 @ Ta = -20 to 85°C

	高速発振: f_{OSCH}	PLL 出力 クロック: f_{PLL}	CPU 用 クロック: f_{FCLK}	USB 用 クロック: f_{USB}
(1) USB 使用, CPU 最大: 144 MHz	24 MHz	Maximum of 144 MHz	Maximum of 144 MHz	24MHz / 48MHz
(2) USB 使用 CPU 最大: 150 MHz	25 MHz	Maximum of 150 MHz	Maximum of 150 MHz	24MHz / 48MHz (X1USB 端子より 入力要)
(3) USB 不要 CPU 最大: 150 MHz	25 MHz	Maximum of 150 MHz	Maximum of 150 MHz	—

3.5.3 動作説明

3.5.3.1 レジスタの説明

SFR のリストと機能を以下に示します。

base アドレス= 0xF005_0000

Register Name	Address (base+)	Description
Reserved	0x000	Reserved
SYSCR1	0x004	System Control Register 1
SYSCR2	0x008	System Control Register 2
SYSCR3	0x00C	System Control Register 3
SYSCR4	0x010	System Control Register 4
SYSCR5	0x014	System Control Register 5
SYSCR6	0x018	System Control Register 6
SYSCR7	0x01C	System Control Register 7
SYSCR8	0x020	System Control Register 8
Reserved	0x040	Reserved
Reserved	0x044	Reserved
Reserved	0x048	Reserved
Reserved	0x04C	Reserved
Reserved	0x050	Reserved
CLKCR5	0x054	Clock Control Register 5

1. SYSCR1 (System Control Register 1)

Address = (0xF005_0000) + (0x0004)

Bit	Bit Symbol	Type	Reset Value	Description
[31:3]	–	–	Undefined	Read as undefined. Write as zero.
[2:0]	GEAR	R/W	0y000	クロックギア選択 (fc) 0y000: fc 0y001: fc/2 0y010: fc/4 0y011: fc/8 0y1xx: Reserved

(個別説明)

a. <GEAR>

クロックギアを選択します。

0y000: fc

0y001: fc/2

0y010: fc/4

0y011: fc/8

0y1xx: Reserved

2. SYSCR2 (System Control Register-2)

Address = (0xF005_0000) + (0x0008)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7]	Reserved	R/W	0y0	Read as undefined. Write as zero.
[6:2]	–	–	Undefined	Read as undefined. Write as zero.
[1]	FCSEL	R/W	0y0	PLL 出力クロック使用選択 0y0: f _{OSCH} 0y1: f _{PLL}
[0]	LUPFLAG	RO	0y0	PLL 用ロックアップカウンタ終了フラグ リード時: 0y0: Not End 0y1: End ライト時: 無効

(個別説明)

a. <FCSEL>

PLL 出力クロックの選択をします。

0y0: f_{OSCH}0y1: f_{PLL}

b. <LUPFLAG>

PLL 用ロックアップカウンタ終了フラグです。

0y0: Not End

0y1: End

3. SYSCR3 (System Control Register 3)

Address = (0xF005_0000) + (0x000C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7]	PLLON	R/W	0y0	PLL 動作制御 0y0: OFF 0y1: ON
[6]	–	–	Undefined	Read as undefined. Write as zero.
[5]	C2S	R/W	0y1	PLL 定数値設定 1 常に 0y0 に設定してください
[4:0]	ND	R/W	0y00111	PLL 定数値設定 2 6 倍時=0y00101, 8 倍時=0y00111

(個別説明)

a. <PLLON>

PLL 動作を制御します。

0y0: OFF

0y1: ON

b. <C2S>

PLL 定数値設定 1 です。

初期値が 0y1 となっていますが、使用前に必ず 0y0 に設定してください。

c. <ND>

PLL 定数値設定 2 です。

6 倍時=0y0_0101, 8 倍時=0y0_0111 に設定してください。

4. SYSCR4 (System Control Register 4)

Address = (0xF005_0000) + (0x010)

Bit	Bit Symbol	Type	Reset Value	Description			
[31:8]	–	–	Undefined	Read as undefined. Write as zero.			
[7:4]	RS	R/W	0y0111	PLL 定数値設定 3			
				8 通倍	6 通倍		
				140MHz 以上	140MHz 未満	140MHz 以上	140MHz 未満
				0y0110	0y1001	0y0110	0y0111
[3:2]	IS	R/W	0y10	PLL 定数値設定 4 常に 0y01 に設定してください			
[1:0]	FS	R/W	0y01	PLL 定数値設定 5			
				8 通倍	6 通倍		
				140MHz 以上	140MHz 未満	140MHz 以上	140MHz 未満
				0y01	0y10	0y01	0y10

(個別説明)

a. <RS>

PLL 定数値設定 3 です。

PLL の通倍数と、通倍する周波数に合わせて以下の値を設定してください。

8 通倍の場合：

通倍された周波数が 140MHz または 140MHz を超える場合: 0y0110

通倍された周波数が 140MHz 未満の場合: 0y1001

6 通倍の場合：

通倍された周波数が 140MHz または 140MHz を超える場合: 0y0110

通倍された周波数が 140MHz 未満の場合: 0y0111

b. <IS>

PLL 定数値設定 4 です。

初期値が 0y10 となっていますが、使用前に必ず 0y01 と設定してください。

c. <FS>

PLL 定数値設定 5 です。

PLL の通倍数と、通倍する周波数に合わせて以下の値を設定してください。

8 通倍の場合：

通倍された周波数が 140MHz または 140MHz を超える場合: 0y01

通倍された周波数が 140MHz 未満の場合: 0y10

6 通倍の場合：

通倍された周波数が 140MHz または 140MHz を超える場合: 0y01

通倍された周波数が 140MHz 未満の場合: 0y10

5. SYSCR5 (System Control Register 5)

Address = (0xF005_0000) + (0x0014)

Bit	Bit Symbol	Type	Reset Value	Description
[31:1]	–	–	Undefined	Read as undefined.
[0]	PROTECT	RO	0y0	Protect Flag 0y0: プロテクト OFF 状態 0y1: プロテクト ON 状態

(個別説明)

SYSCR6 と SYSCR7 レジスタに 2 重の鍵を設定する事によりプロテクト(Clock Controller 内の特定の SFR へのライト動作)の実行、解除が可能となります。

(2 重の鍵)

1st-KEY: SYSCR6 に 0x5A、SYSCR7 に 0xA5 を連続ライト

2nd-KEY: SYSCR6 に 0xA5、SYSCR7 に 0x5A を連続ライト

プロテクトの状態は、SYSCR5<PROTECT>をリードすることにより確認できます。

リセットにより、プロテクト OFF 状態となります。また、プロテクト ON 状態にて特定の SFR へのライト動作が実行された場合、ライトされたデータは無効となります。

特定のレジスタは以下です。

SYSCR1, SYSCR2, SYSCR3, SYSCR4, SYSCR5, SYSCR8

CLKCR5

6. SYSCR6 (System Control Register 6)

Address = (0xF005_0000) + (0x0018)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:0]	P-CODE0	WO	0X00	Protect コード設定-0

(個別説明)

a. <P-CODE0>

Protect コード 0 を設定します。

7. SYSCR7 (System Control Register 7)

Address = (0xF005_0000) + (0x001C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:0]	P-CODE1	WO	0X00	Protect コード設定-1

(個別説明)

a. <P-CODE1>

Protect コード 1 を設定します。

8. SYSCR8 (System Control Register 8)

Address = (0xF005_0000) + (0x0020)

Bit	Bit Symbol	Type	Reset value	Description
[31:8]	-	-	Undefined	Read as undefined. Write as zero.
[7:6]	-	-	Undefined	Read as undefined. Write as zero.
[5:4]	USBD_CLKSEL	R/W	0y00	USB Device 用クロック選択 00 : "0"固定 01 : X1USB 端子クロックの 2 分周クロック(1/2) 10 : X1USB 端子クロック 11 : X1 端子クロック
[3]	-	-	Undefined	Read as undefined. Write as zero.
[2:0]	USBH_CLKSEL	R/W	0y000	USB host 用クロック選択 000 : "0"固定 001 : X1USB 端子クロック 010: PLL 出力クロックの 3 分周クロック(1/3) 011 : "0"固定 100 : PLL 出力クロックの 4 分周クロック(1/4) 101 : X1 端子クロック 110 : "0"固定 111 : "0"固定

9. CLKCR5 (Clock Control Register-5)

Address = (0xF005_0000) + (0x0054)

Bit	Bit Symbol	Type	Reset Value	Description
[31:7]	—	—	Undefined	Read as undefined. Write as zero.
[6]	Reserved	R/W	0y1	Read as undefined. Write as one.
[5]	—	—	Undefined	Read as undefined. Write as zero.
[4]	USBH_CLKEN	R/W	0y1	USB_HOST 用クロック供給 enable/disable 選択 0y0 : disable 0y1 : enable
[3]	Reserved	R/W	0y1	Write as one.
[2]	SEL_TIM45	R/W	0y1	Timer45 用プリスケラクロック選択 0y0: fs (32.768kHz)クロック 0y1: f _{PCLK} /2
[1]	SEL_TIM23	R/W	0y1	Timer23 用プリスケラクロック選択 0y0: fs (32.768kHz)クロック 0y1: f _{PCLK} /2
[0]	SEL_TIM01	R/W	0y1	Timer01 用プリスケラクロック選択 0y0: fs (32.768kHz)クロック 0y1: f _{PCLK} /2

(個別説明)

a. <USBH_CLKEN>

USB Host クロック出力 Enable

0y0: Disable

0y1: Enable

USB Host 用クロックを変更したい時、USB Host 用クロックの出力を Disable して、切り替えを実施してください。

b. <SEL_TIM45>

Timer45 用プリスケラクロックを選択します。

0y0: fs (32.768kHz)クロック

0y1: f_{PCLK}/2

c. <SEL_TIM23>

Timer23 用プリスケラクロックを選択します。

0y0: fs (32.768kHz)クロック

0y1: f_{PCLK}/2

d. <SEL_TIM01>

Timer01 用プリスケラクロックを選択します。

0y0: fs (32.768kHz)クロック

0y1: f_{PCLK}/2

3.5.4 システムクロック制御部

システムクロック制御部は、CPU コア(f_{FCLK})および その他の内蔵 I/O へ供給されるクロック (f_{HCLK}) を生成する回路です。 f_{OSCH} または f_{PLL} クロックを入力として $SYSCR1<GEAR[2:0]>$ で高速クロックのギアを 1、2、4、8 段 (f_c 、 $f_c/2$ 、 $f_c/4$ 、 $f_c/8$) に切り替え、消費電力の低減を図ることができます。

リセットにより、PLL-OFF モードになり $<GEAR2:0> = "000"$ に初期化されますので CPU 用クロック f_{FCLK} は f_{OSCH} と同一の周波数になります。例えば、X1, X2 端子に 24MHz の発振子を接続していると、リセットにより f_{FCLK} は 24MHz となります。

(1) クロックギア

クロックギア選択レジスタ $SYSCR1<GEAR2:0>$ により f_c 、 $f_c/2$ 、 $f_c/4$ 、 $f_c/8$ のいずれかに設定できます。

クロックギアを使用して f_{FCLK} を切り替えることにより、消費電力の低減が図れます。下記に、クロックギアの切り替え例を示します。

(設定例)

```

;
(SYSCR1)          ←      0x0000_0011      ; fFCLK を 1/8 へ切り替え

```

3.5.5 クロック通倍回路(PLL)

PLL は f_{OSCH} の 6 または 8 倍となる、 f_{PLL} クロック信号を出力します。これにより発振器の周波数は低く、内部クロックは高速にすることが可能です。

リセットにより PLL は停止状態に初期化されますので PLL を使用する場合は $SYSCR2$ 、 $SYSCR3$ 、 $SYSCR4$ レジスタへの設定が必要です。

この回路は発振器のように動作許可後に安定させる時間を必要とし、それをロックアップタイムと呼びます。

この時間を確認するために 12 段のバイナリカウンタがあります。ロックアップタイムは $f_{OSCH} = 25\text{MHz}$ の場合で約 $164\mu\text{s}$ です。

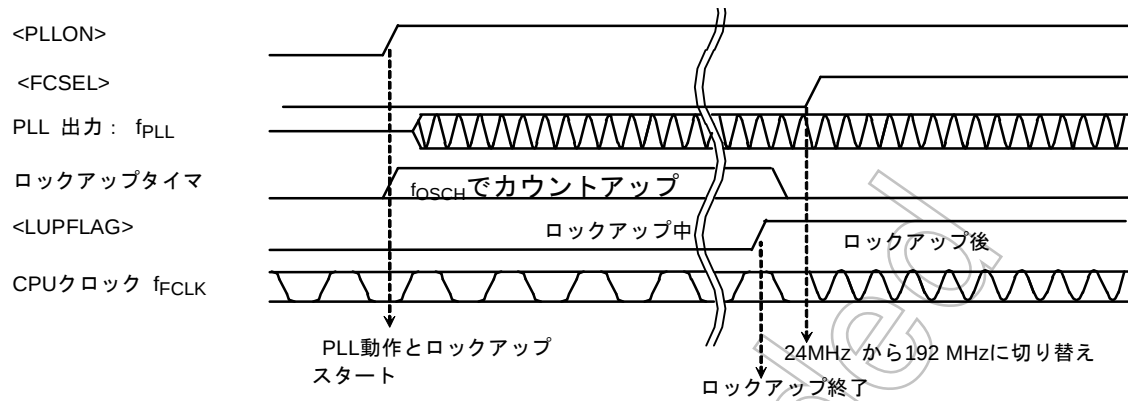
以下は PLL スタートと PLL ストップの設定例です。

(設定例-1) PLL スタート

```

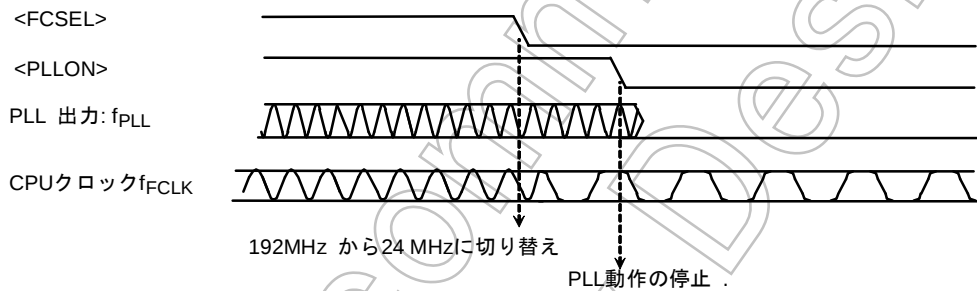
SYSCR4          ←      0x00000065      ; PLL8 通倍の定数を設定
SYSCR3          ←      0x00000087      ; PLL8 通倍で動作 ON
LOCKUP:
SYSCR2          →      r0              ; <LUPFLAG>==1?
LDR r1,=0x01
AND r0,r0,r1
LDR r1,=0x01
CMP r0,r1
BNE LOCKUP      ; r0 ≠ r1 , jump to LOCKUP
(SYSCR2)        ←      0x00000002      ; <FCSEL>=1 (24MHz から 192MHz へ切換え)

```



(設定例-2) PLL ストップ

SYSCR2 ← 0x0000_0000 ; <FCSEL>=0 (192MHz から 24MHz へ切換え)
 Dummy 命令実行 (注)
 SYSCR3 ← 0x0000_0007 ; <PLLON>=0



注) <FCSEL>を"1"から"0"へ切り替える場合、レジスタライト後に f_{CLK} が f_{OSCH} へ切り替わるまでに数クロックのクロックを必要とします。よって、その時間分待ってから次の命令の実行が必要です。具体的には NOP 命令を 10 命令実行してください。

3.6 ブートROM

ユーザプログラムを、内蔵 RAM に LOAD するブート ROM を内蔵しています。
以下のロード方法に対応しています。

3.6.1 動作モード

本製品には外部メモリ起動、内蔵ブート ROM 起動 2 つの動作モードがあります。各モードは RESETn がアサートされたときの AM1、AM0 端子の状態により設定されます。

- (1) 外部メモリモード:リセット後、CPU は外部メモリから命令をフェッチし、実行します。
- (2) ブートモード:リセット後、CPUは内蔵ブートROMからの命令をフェッチし、実行します。内蔵ブートROMは、USB通信によりユーザプログラムを内蔵RAMに転送した後、内蔵RAMに分岐します。
この動作によって、ユーザプログラムはブートを開始します。
表 3.6.2 はブート動作の概要を示しています。

表 3.6.1 動作モード

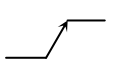
モード設定端子			動作モード
RESETn	AM1	AM0	
	0	1	外部バスメモリから開始(16bit バス起動)
	1	0	外部バスメモリから開始(32bit バス起動)
	1	1	ブート (内蔵ブート ROM から開始)
	0	0	TEST (設定禁止)

表 3.6.2 ブート動作の概要

認識順	ロード			ロード後の動作
	転送元	I/F	転送先	
1	PC 等 USB HOST	USB	内蔵 RAM	内蔵 8KB_RAM に分岐 0x0000_0000

3.6.2 内蔵ブートROMのハードウェア仕様

(1) メモリマップ

図 3.6.1はブートモードのメモリマップを示します。

内蔵ブート ROM は、16KB ROM で構成され、0x0000_0000~0x0000_3FFF アドレスにアサインされています。

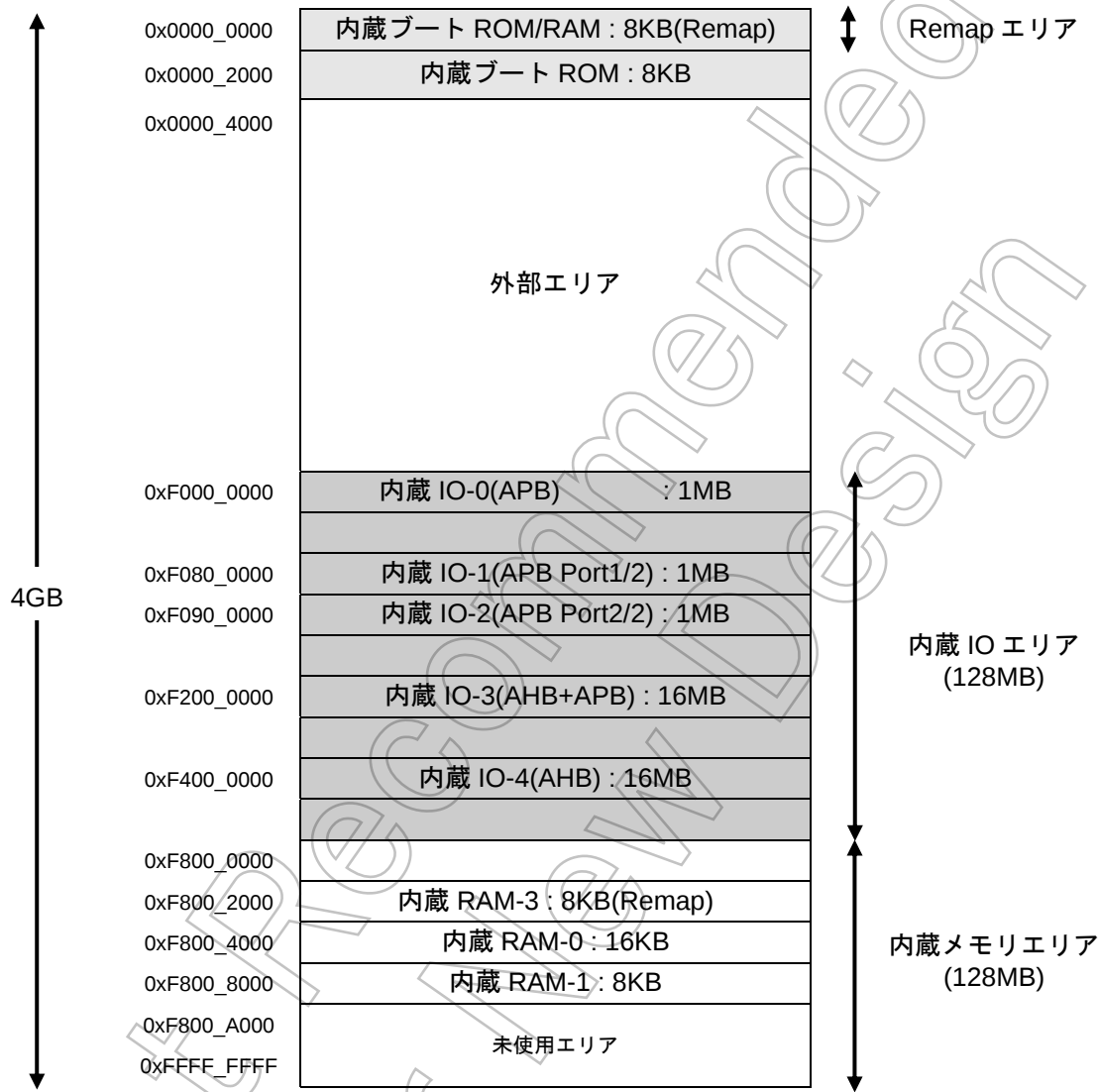
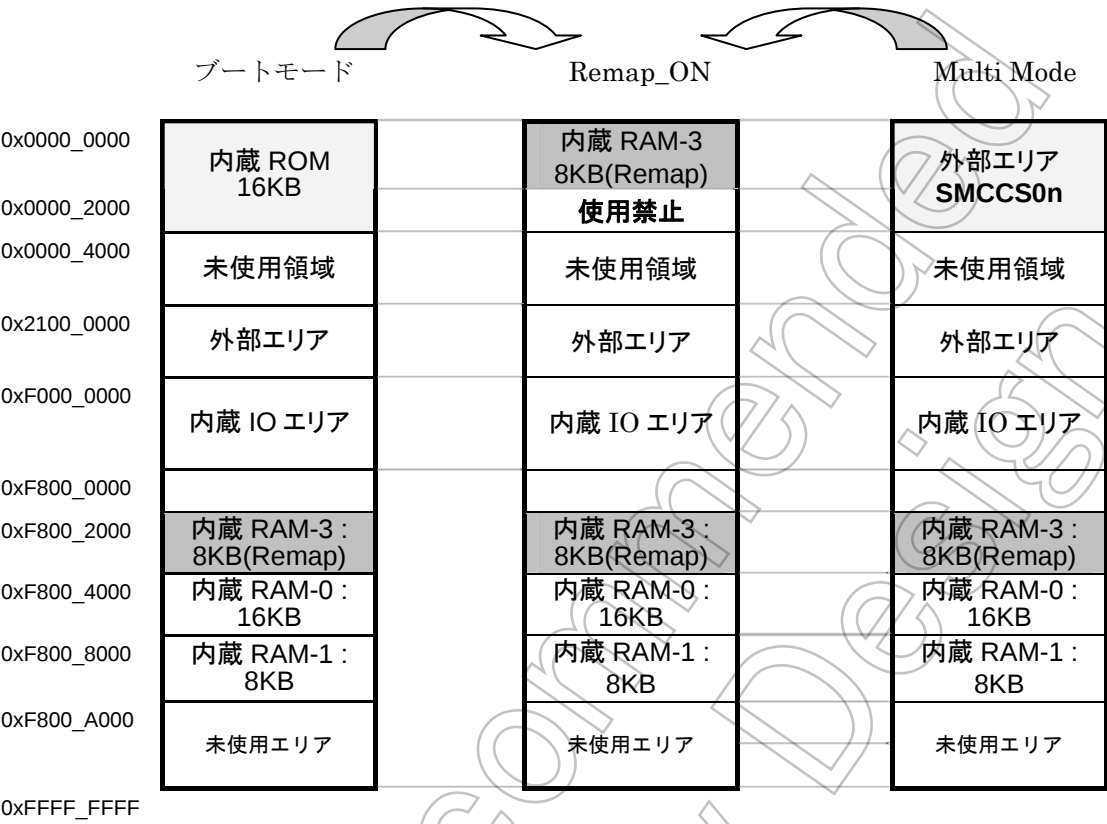


図 3.6.1 ブートモードのメモリマップ

(2) ブート ROM の排除機能

ブートモードでブートシーケンス実行後、Remap し、内蔵ブート ROM 領域が RAM に切り替わります。



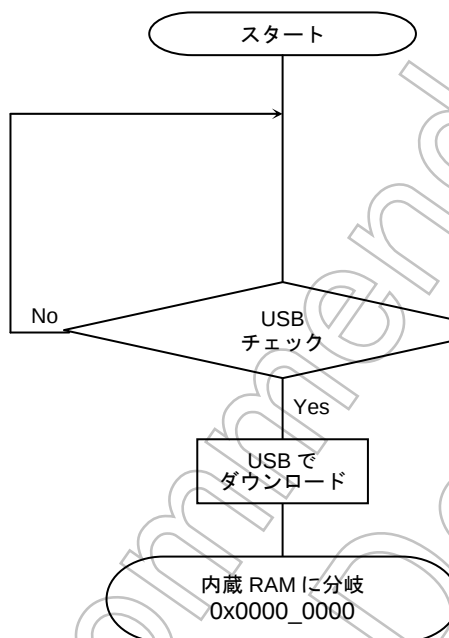
注) 0x0000_0000~0x0000_1FFF(8KB)の空間は Remap エリアとなっており、Remap_ON 設定で内蔵 RAM3 エリアへのアクセスに切り替わります (F8000_2000 へのアクセス時も RAM3 エリアを示します)。

図 3.6.2 メモリマップ(起動モード・外部エリア詳細)

3.6.3 ブート動作の概要

ブート動作の転送元として USB から選択可能です。

リセット後、内蔵ブートROM上のブートプログラムは、図 3.6.3 のようなフローチャートで動作します。いずれも転送元から内蔵RAMへユーザプログラムを転送後、内蔵RAMへ分岐します。また、内蔵RAMの使用方法は転送元に関係なく共通であり図 3.6.4に示します。



注) USB でダウンロードする場合、PC 上に専用の USB デバイスドライバ、アプリケーションソフトが必要となります。

図 3.6.3 内蔵ブート ROM のフローチャート概要

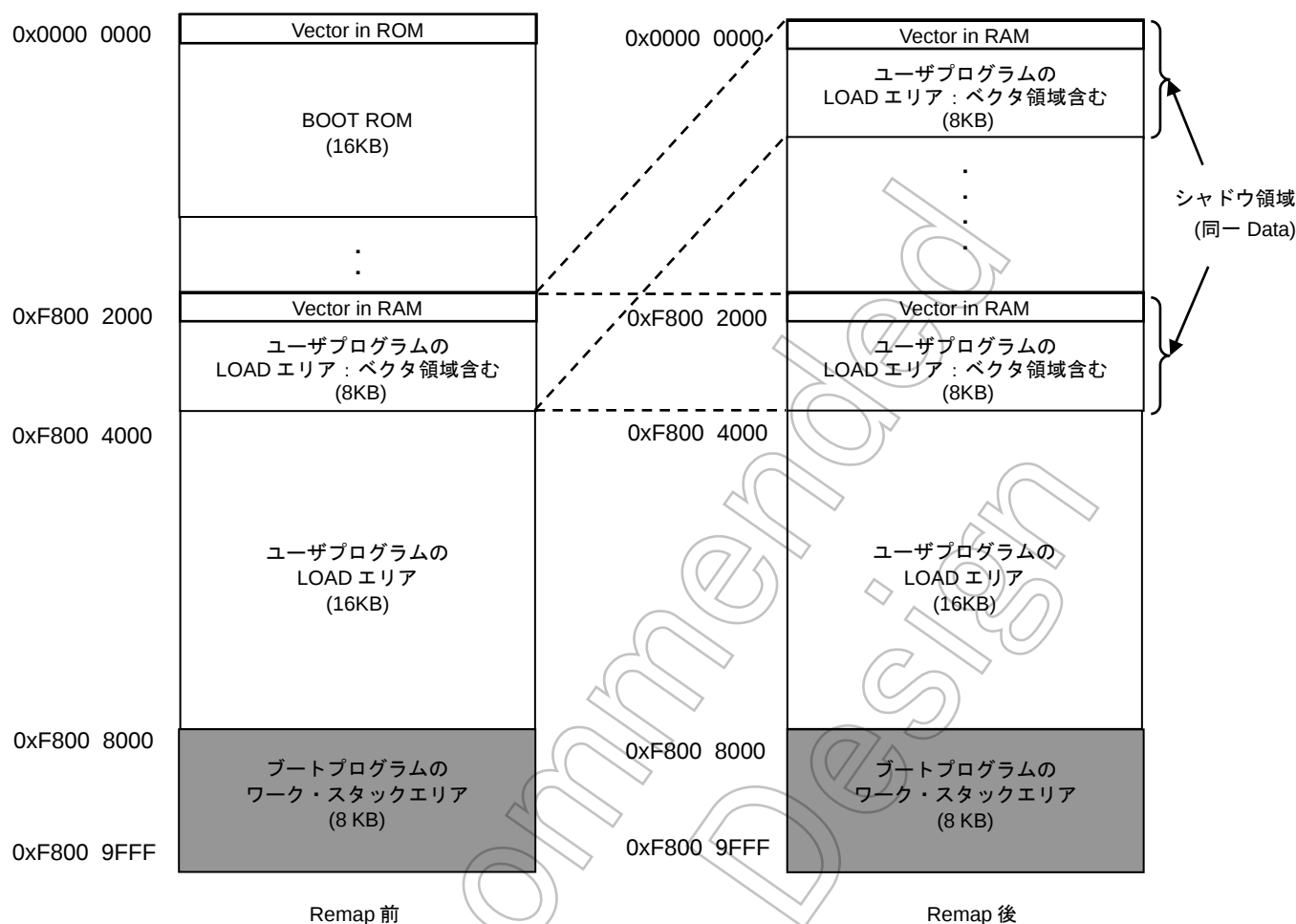


図 3.6.4 ブートプログラムの内蔵 RAM 使用方法

内蔵 RAM の内、0xF800_8000~0xF800_9FFF の領域はブートプログラムを実行する際のワーク領域とスタック領域として使用されています。このため、内蔵 RAM に LOAD するユーザプログラムの最大容量は、24KB となります。

ユーザプログラム領域の 0xF800_2000~0xF800_7FFF の 24KB の内、0xF800_2000~0xF800_3FFF の 8KB の空間は、ベクタ+プログラムを書き込む領域となります。

ブートプログラムは、ユーザプログラムを内蔵 RAM のユーザプログラムエリアに LOAD した後、ブートプログラムのワーク領域 (0xF800_8000~0xF800_9FFF) に JUMP し、ワーク領域内で Remap 機能を ON します。

Remap 機能を ON することで、0xF800_2000~0xF800_3FFF の 8KB の空間を、0x0000_0000~0x0000_1FFF から、アクセスできるようになります。

Remap 機能の詳細は、「システムコントローラ」の章を参照ください。

ブートプログラムは、最後に Remap された RAM 領域の 0x0000_0000(RESET ベクタ)に分岐します。

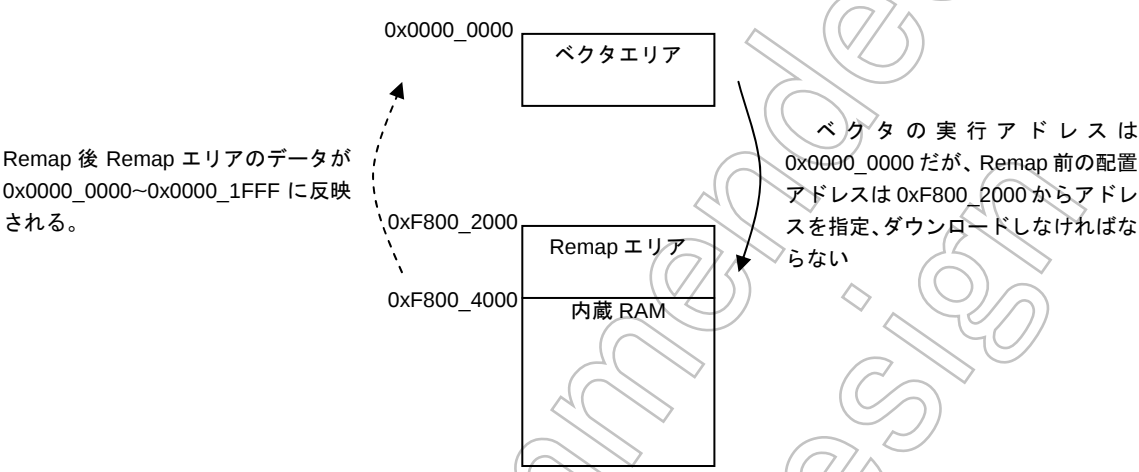
上図のように、リマップ前 RAM 領域の 0xF800_2000 番地は、リマップ後 0x0000_0000 番地(RESET ベクタ)に、リマップ前 RAM 領域の 0xF800_2018 番地は、リマップ後 0x0000_0018 番地(IRQ ベクタ)という具合に、アドレス変換されます。

よって、ブート ROM 起動後のベクタ設定を、0xF800_2000 番地以降にあらかじめ書き込んでおく必要があります。

3.6.3.1 USBブート例

USB からのブートでは Remap エリア(0xF800_2000~0xF800_3FFF)の 8KB にユーザプログラムのベクタ、内蔵 RAM エリア(0xF800_4000~0xF800_7FFF)の 16KB にプログラム本体をダウンロードします。

その後ブートプログラムにより Remap が行われ、Remap エリアのデータがベクタエリア(0x0000_0000~0x0000_1FFF)に反映され、0x0000_0000 番地にジャンプすることによりユーザプログラムが開始されます。



(1) CPU 状態、ポート設定

ARM926EJ™-S はリセット後、スーパーバイザーモードからスタートしますが、ブートプログラムでは、モード変更を行わず、スーパーバイザーモードのまま、すべてのプログラム処理を実行します。

表 3.6.3 ブートプログラムでのポート設定

BOOT	PORT 名	I/O	ブートプログラムによるポート設定
USB	DDP	入出力	専用端子のため、特に設定しません。
	DDM	入出力	

(2) ブートプログラムの制御レジスタ設定

ブートプログラムにて設定している内部回路の制御レジスタを表 3.6.4に示します。

ブートシーケンス実行後これらの設定値を注意の上プログラムを作成してください。また、スタックポインタ、0xF800_8000~0xF800_9FFF を含む内蔵 RAM もブートプログラム実行後の状態となっていますので必要に応じて一度初期化して使用するよう注意してください。

表 3.6.4 SFR 一覧

レジスタ 名称	設定値	設定内容
SYSCR1	0x0002	クロックギアは 1/4 状態です。
SYSCR2	0x0002	PLL クロックを使用する(×8)状態です。
SYSCR3	0x0087	
SYSCR4	0x0065	
SYSCR8	0x0030	USB Device 用クロック
REMAP	0x0001	Remap ON

【 注意・重要 】

ブートシーケンス内では、Timer0 が使用されています。(Timer0control<TIM0EN> = 1 となり、動作が Enable 状態になっています。)

ユーザプログラム実行時に、Timer0 の割り込みが発生する可能性がありますので、利用前に、一旦 Timer0 の割り込みをクリアしてください。

Timer0IntClr<TIM0INTCLR>に任意の値を書き込み、Timer0 割り込みをクリアする

注) ここでは USB、INTC、DMAC の各 SFR に対する設定値は省略しています。ユーザプログラム内でこれらの機能を使用する場合は、各 SFR を再設定してください。

3.6.4 USBでのダウンロード

(1) 接続例

図 3.6.5 はUSB (NORフラッシュはプログラムメモリを想定) の接続例を示しています。

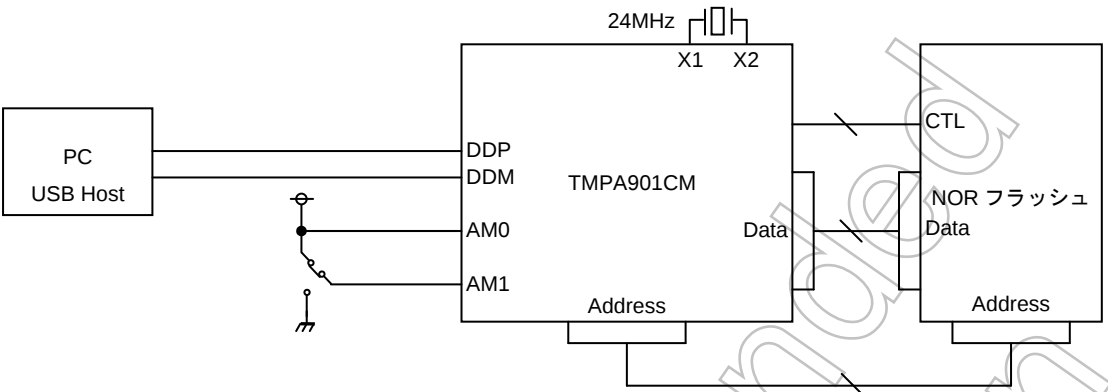


図 3.6.5 USB の接続例

(2) USB インタフェース仕様概要

USB でブートする場合、X1,X2 端子に接続の発振周波数は 24.00 MHz(±100ppm)にしてください。

本マイコンの USB Device は High Speed 対応ですが、相手側の USB_HOST が USB_High Speed に対応していない(USB1.1 以前)場合は、Full Speed での通信となります。

(USB 用の専用クロック(X1USB)端子からクロックを供給する方式は、ブート ROM 機能ではサポートしていません)

USB Device 使用時は USB Device の章を参照ください。

USB 仕様の 4 つの転送タイプ中、下記 2 つの転送タイプを使用してブート機能を実現しています。

表 3.6.5 ブートプログラムが使用する転送タイプ

転送タイプ	用 途
コントロール転送	スタンダードリクエスト、ベンダーリクエストの送信を行います。
バルク転送	ベンダーリクエストでの要求物の返信、ユーザ プログラムの送信を行います。

全体フロー概要を以下に記述します。

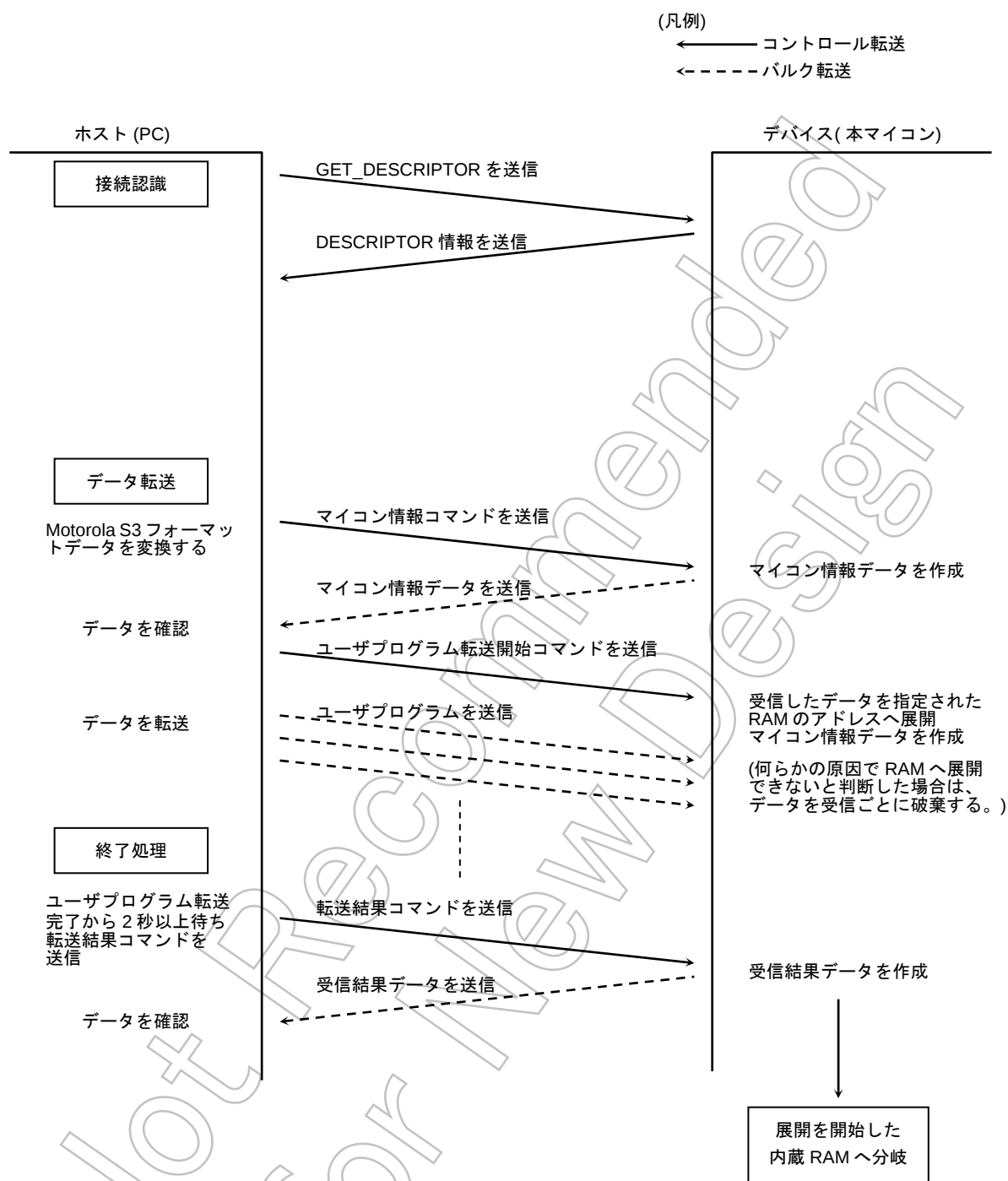


図 3.6.6 全体フロー概要

ベンダークラスでの通信は以下のとおりとなります。

セットアップコマンドのデータ構造を以下に記述します。

表 3.6.6 セットアップコマンドのデータ構造

フィールド名	値	意味
bmRequestType	0x40	D7 0y0: ホスト-デバイス D6-D5 0y10: ベンダ D4-D0 0y00000: デバイス
bRequest	0x00, 0x02, 0x04	0x00: マイコン情報 0x02: ユーザプログラム転送開始 0x04: ユーザプログラム転送結果
wValue	0x00~0xFFFF	固有データ番号 (マイコン側では未使用)
wIndex	0x00~0xFFFF	ライトする長さ ユーザプログラム転送開始時使用 (ユーザプログラム転送サイズ)
wLength	0x0000	固定

ベンダーリクエストコマンド表を以下に記述します。

表 3.6.7ベンダーリクエストコマンド表

コマンド名称	ベンダー リクエスト の数値	動作概要	備考
マイコン情報コマンド	0x00	デバイスがマイコン 情報を送信する。	マイコン情報データの送信はセット アップステージ終了後、バルク IN 転 送で行います。
ユーザプログラム転送 開始コマンド	0x02	デバイスがユーザ プログラムの受信を 開始する。	wIndex にユーザプログラムの転送サ イズを入れてください。 ユーザプログラムの受信はセットア ップステージ終了後、バルク OUT 転 送で行ってください。
ユーザプログラム 転送結果コマンド	0x04	デバイスが転送結果 を送信する。	転送結果データの送信はセットアッ プステージ終了後、バルク IN 転送で 行います。

スタンダードリクエストコマンド表を以下に記述します。

表 3.6.8 スタンダードリクエストコマンド表

スタンダードリクエスト	応答方法
GET_STATUS	未対応
CLEAR_FEATURE	未対応
SET_FEATURE	未対応
SET_ADDRESS	対応
GET_DESCRIPTOR	対応
SET_DESCRIPTOR	未対応
GET_CONFIGURATION	未対応
SET_CONFIGURATION	対応
GET_INTERFACE	未対応
SET_INTERFACE	未対応
SYNCH_FRAME	無視

GET_DESCRIPTOR で返信する情報を以下に記述します。

表 3.6.9 GET_DESCRIPTOR で返信する情報

DeviceDescriptor

フィールド名	値	意味
Blength	0x12	18 バイト
BdescriptorType	0x01	デバイスディスクリプタ
BcdUSB	0x0200	USB Version 2.0
BdeviceClass	0x00	デバイスクラス未使用
BdeviceSubClass	0x00	サブコマンド未使用
BdeviceProtocol	0x00	プロトコル未使用
BmaxPacketSize0	0x40	EP0 最大パケットサイズ 64 バイト
IdVendor	0x0930	ベンダ ID
IdProduct	0x6504	プロダクト ID (0)
BcdDevice	0x0001	デバイスバージョン (v0.1)
Imanufacturer	0x00	製造者名を示すストリングディスクリプタのインデックス値
lproduct	0x00	製品名を示すストリングディスクリプタのインデックス値
lserialNumber	0x00	製造番号を示すストリングディスクリプタのインデックス値
BnumConfigurations	0x01	構成は 1 つ

注) 上記情報は、ブート動作時に USB_HOST に返信する情報です。使用するアプリケーションに応じて、DESCRIPTOR 情報を変更してご使用ください。

ConfigurationDescriptor

フィールド名	値	意味
bLength	0x09	9 バイト
bDescriptorType	0x02	コンフィグレーションディスクリプタ
wTotalLength	0x0020	コンフィグレーション++エンドポイントの 各ディスクリプタを合わせた長さ (32 バイト)
bNumInterfaces	0x01	インタフェースは 1 つ
bConfigurationValue	0x01	構成番号 1
iConfiguration	0x00	このコンフィグレーション名を示すストリングディス クリプタのインデックス値 (未使用)
bmAttributes	0x80	バス電源
MaxPower	0x31	最大消費電力 (49 mA)

InterfaceDescriptor

フィールド名	値	意味
bLength	0x09	9 バイト
bDescriptorType	0x04	インタフェースディスクリプタ
bInterfaceNumber	0x00	インタフェース番号 0
bAlternateSetting	0x00	代替設定番号 0
bNumEndpoints	0x02	エンドポイントは 2 つ
bInterfaceClass	0xFF	固有のデバイス
bInterfaceSubClass	0x00	
bInterfaceProtocol	0x50	BulkOnly プロトコル
iInterface	0x00	このインタフェース名を示すストリングディスクリ プタのインデックス値 (未使用)

注) 上記情報は、ブート動作時に USB_HOST に返信する情報です。使用するアプリケーションに応じて、
DESCRIPTOR 情報を変更してご使用ください。

EndpointDescriptor(USB_HOST が USB2.0 対応の場合)

フィールド名	値	意味
<Endpoint1>		
bLength	0x07	7 バイト
bDescriptorType	0x05	エンドポイントディスクリプタ
bEndpointAddress	0x81	EP1 は IN
bmAttributes	0x02	バルク転送
wMaxPacketSize	0x0200	ペイロード 512 バイト
bInterval	0x00	(バルク転送のため、無視される)
<Endpoint2>		
bLength	0x07	7 バイト
bDescriptor	0x05	エンドポイントディスクリプタ
bEndpointAddress	0x02	EP2 は OUT
bmAttributes	0x02	バルク転送
wMaxPacketSize	0x0200	ペイロード 512 バイト
bInterval	0x00	(バルク転送のため、無視される)

EndpointDescriptor(USB_HOST が USB1.1 対応の場合)

フィールド名	値	意味
<Endpoint1>		
bLength	0x07	7 バイト
bDescriptorType	0x05	エンドポイントディスクリプタ
bEndpointAddress	0x81	EP1 は IN
bmAttributes	0x02	バルク転送
wMaxPacketSize	0x0040	ペイロード 64 バイト
bInterval	0x00	(バルク転送のため、無視される)
<Endpoint2>		
bLength	0x07	7 バイト
bDescriptor	0x05	エンドポイントディスクリプタ
bEndpointAddress	0x02	EP2 は OUT
bmAttributes	0x02	バルク転送
wMaxPacketSize	0x0040	ペイロード 64 バイト
bInterval	0x00	(バルク転送のため、無視される)

注) 上記情報は、ブート動作時に USB_HOST に返信する情報です。使用するアプリケーションに応じて、DESCRIPTOR 情報を変更してご使用ください。

マイコン情報コマンドで返信する情報を以下に記述します。

表 3.6.10 マイコン情報コマンドで返信する情報

マイコン情報	ASCII コード
TMPA900CM	0x54,0x4D,0x50,0x41,0x39,0x30,0x30,0x43,0x4D,0x20,0x20,0x20,0x20,0x20,0x20

注 1) マイコン情報の製品名の後にはスペースが 6 個含まれています。

注 2) マイコン情報について、TMPA900CMXBG および TMPA901CMXBG はシリーズ共通の情報 TMPA900CM を使用しています。

転送結果コマンドで返信する情報を以下に記述します。

表 3.6.11 転送結果コマンドで返信する情報

転送結果	値	エラー条件
正常終了	0x00	
ユーザ プログラム未受信	0x02	ユーザ プログラム転送開始が抜けて、ユーザ プログラム転送結果を受信した場合。
Motorola S3 形式以外のファイル受信	0x04	ユーザ プログラムの最初のデータが 'S' (0x53) 以外だった場合。
規定以上のユーザプログラム容量受信	0x06	ユーザ プログラム転送開始コマンド受信時の wIndex 以上のデータを受信した場合。
規定外アドレス受信	0x08	ユーザ プログラムダウンロード領域以外のアドレスにダウンロード要求された場合。
プロトコルエラーまたは上記以外のエラー	0x0A	ユーザ プログラム転送開始、ユーザ プログラム転送結果を最初に受信した場合。 Motorola S3 ファイル内のチェック SUM 異常を検出した場合。 Motorola S3 ファイル内のレコードタイプ異常を検出した場合。 DMA 転送にて異常を検出した場合。

(3) USB ブートプログラム動作説明

PCからMotorola S3 フォーマットで送られてきたデータを内蔵RAMへ転送します。転送が正常に終了するとユーザプログラムの実行を開始します。実行開始アドレスは、0x0000_0000 番地です。詳細は 3.6.3 章を参照してください。

この機能により、ユーザ独自のオンボードプログラミング制御を行うことができます。

a. 動作手順

1. USB ケーブルを接続してください。
2. AM0、AM1 端子を共に “1” に設定し、マイコンをリセットします。
3. PC は USB の接続を認識したら、GET_DESCRIPTOR コマンドで接続先の情報を確認してください。
4. PC はマイコン情報コマンドをコマンド転送（ベンダーリクエスト）で送信します。
5. ブートプログラムはマイコン情報コマンドを受信すると、マイコン情報を ASCII コードで送信準備します。
6. PC はマイコン情報データを確認してください。
7. PC はユーザプログラム転送開始コマンドをコマンド転送（ベンダーリクエスト）で送信し、セットアップステージ終了後、バルク OUT 転送でユーザプログラムを転送してください。
8. PC はユーザプログラム転送完了後、2 秒以上待つユーザプログラム転送結果コマンドをコマンド転送（ベンダーリクエスト）で送信します。
9. ブートプログラムは、ユーザ プログラム転送結果コマンドを受信すると、転送結果の値を送信準備します。
10. PC は転送結果を確認してください。
11. 転送結果が正常終了以外の場合、ブートプログラムは異常処理に入り自然復旧しません。PC 上のデバイスドライバを終了させ、2. からやり直してください。

b. ユーザプログラムフォーマット（バイナリ）の注意点

1. レコードのチェックサム受信後は、次のレコードのスタートマーク (0x53, “S”) 待ち状態になりますので、レコード間に 0x53 以外のデータを送信しても、そのデータは無視します。

注) USB 転送においては、ライトサイズを wIndex で 0x0000-0xFFFF の範囲で設定するため、転送する Object サイズの最大値は 64KB になりますので注意してください。

3.6.5 注意事項

ブート ROM を利用する上での注意点をまとめています。

1. **TIMER0** の利用について

ブートシーケンス内では、**Timer0** が使用されています。(Timer0control<TIM0EN> = 0y1 となり、動作が **Enable** 状態になっています。)

ユーザプログラム実行時に **Timer0** の割り込みが発生する可能性がありますので、利用前に、一旦 **Timer0** の割り込みをクリアしてください。

Timer0IntClr<TIM0INTCLR>に任意の値を書き込み、**Timer0** 割り込みをクリアする

2. **USB** コネクタについて

USB でのブート中に **USB** コネクタの抜き差しを行わないでください。

3. **PC** 上のソフトウェアについて

USB でのブート中に **PC** 上に専用の **USB** デバイスドライバ、アプリケーションソフトが必要となります。

3.7 割り込み

3.7.1 機能概要

- 21 種類の割り込みソースに対応
- ハードウェア(H/W)で、32 段階の Default Priority に固定(ソフトウェアのレベルが同一の場合に有効)
- ソフトウェア(S/W)で、16 段階(0~15)の割り込みレベルの設定が可能
- ハードウェアおよびソフトウェア Priority レベルの MASK 設定が可能
- 通常割り込み要求(IRQ)と高速割り込み要求(FIQ)の設定が可能
- ソフトウェア割り込み発生可能

3.7.2 ブロック図

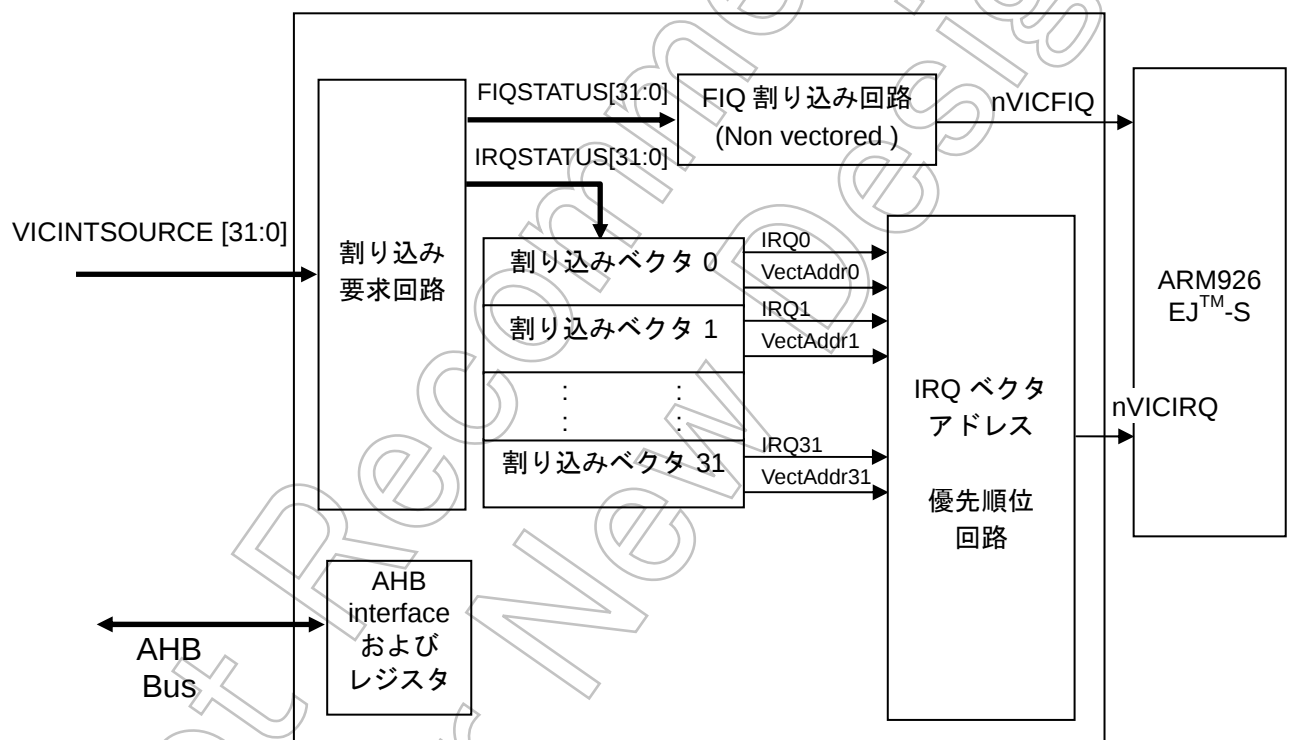


図 3.7.1 ブロック図

- 割り込み要求のロジック回路図

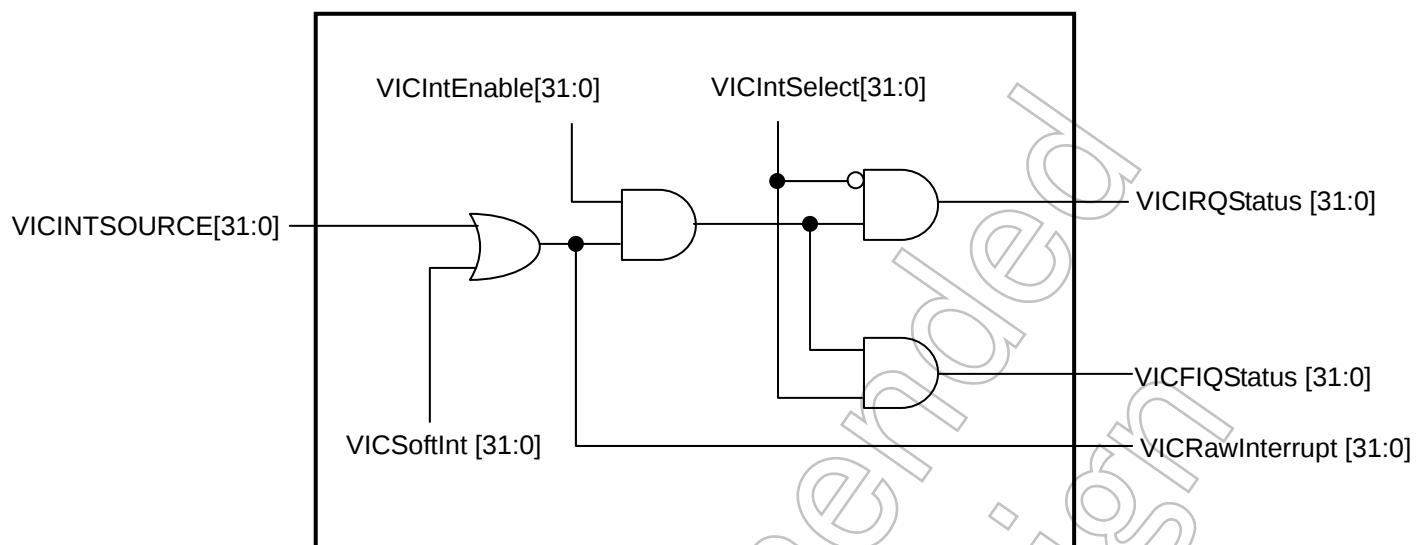


図 3.7.2 ステータスフラグ図

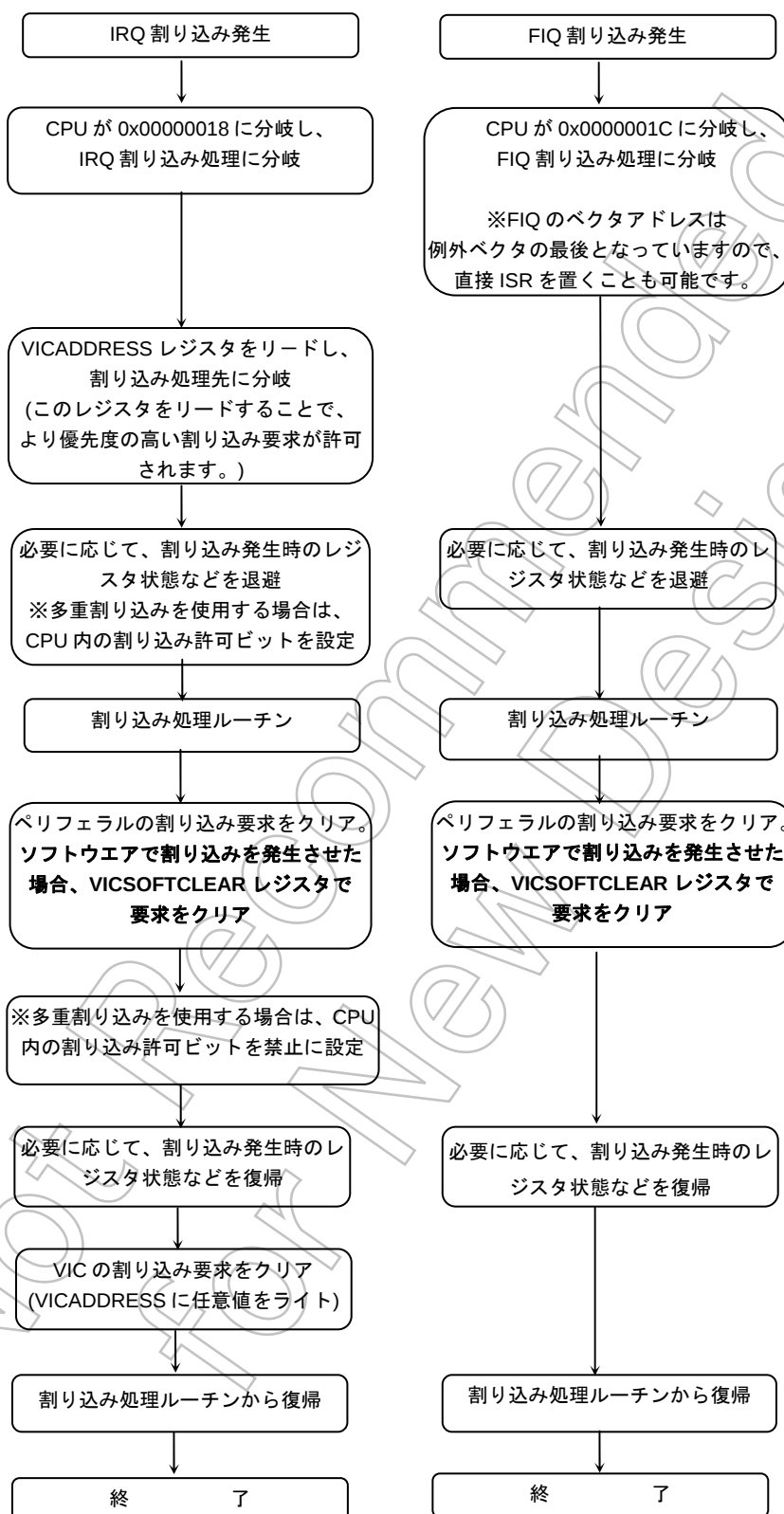
3.7.3 動作説明

割り込みコントローラ(VIC)では、FIQ(高速割り込み)およびIRQ(通常割り込み)を設定できます。

FIQ は一回で一つの FIQ ソースしか使用できません、FIQ は低レイテンシの割り込み、割り込みソースを確認しなくても割り込みサービスルーチンを実行できます。また、FIQ は最高プライオリティを持っています。

Not Recommended
for New Design

- 割り込み処理フロー



3.7.4 割り込み要因一覧

表 3.7.1 割り込み要因一覧表

割り込み要因 Num (注)	割り込み要因(ソース)	対応する Vector Address
0	WDT	Vector Address 0
1	RTC	Vector Address 1
2	Timer01	Vector Address 2
3	Timer23	Vector Address 3
4	Timer45	Vector Address 4
5	GPIOD:INTA (TSI), INTB	Vector Address 5
6	I2C ch0	Vector Address 6
7	Reserved	Vector Address 7
8	ADC	Vector Address 8
9	Reserved	Vector Address 9
10	UART ch0	Vector Address 10
11	UART ch1	Vector Address 11
12	SSP ch0	Vector Address 12
13	Reserved	Vector Address 13
14	NDFC	Vector Address 14
15	Reserved	Vector Address 15
16	DMA 転送エラー	Vector Address 16
17	DMA 転送終了	Vector Address 17
18	LCDC	Vector Address 18
19	Reserved	Vector Address 19
20	LCDDA	Vector Address 20
21	USB Device	Vector Address 21
22	Reserved	Vector Address 22
23	I ² S	Vector Address 23
24	Reserved	Vector Address 24
25	Reserved	Vector Address 25
26	Reserved	Vector Address 26
27	USB Host	Vector Address 27
28	Reserved	Vector Address 28
29	Reserved	Vector Address 29
30	GPIOC (INT9)	Vector Address 30
31	GPIOA (KI0 ~ KI3)	Vector Address 31

注) INTS[Num]は各割り込み要因の信号名、例：INTS[1]は RTC の割り込み要因信号。

3.7.5 SFR

SFR のリストを以下に示します。

表 3.7.2 SFR 一覧表 (1/2)

base アドレス= 0xF400_0000

Register Name	Address (base+)	Description
VICIRQSTATUS	0x0000	IRQ Status Register
VICFIQSTATUS	0x0004	FIQ Status Register
VICRAWINTR	0x0008	Raw Interrupt Status Register
VICINTSELECT	0x000C	Interrupt Select Register
VICINTENABLE	0x0010	Interrupt Enable Register
VICINTENCLEAR	0x0014	Interrupt Enable Clear Register
VICSOFTINT	0x0018	Software Interrupt Register
VICSOFTINTCLEAR	0x001C	Software Interrupt Clear Register
VICPROTECTION	0x0020	Protection Enable Register
VICSWPRIORITYMASK	0x0024	Software Priority Mask Register
–	0x0028	Reserved
VICVECTADDR0	0x0100	Vector Address 0 Register
VICVECTADDR1	0x0104	Vector Address 1 Register
VICVECTADDR2	0x0108	Vector Address 2 Register
VICVECTADDR3	0x010C	Vector Address 3 Register
VICVECTADDR4	0x0110	Vector Address 4 Register
VICVECTADDR5	0x0114	Vector Address 5 Register
VICVECTADDR6	0x0118	Vector Address 6 Register
–	0x011C	Reserved
VICVECTADDR8	0x0120	Vector Address 8 Register
–	0x0124	Reserved
VICVECTADDR10	0x0128	Vector Address 10 Register
VICVECTADDR11	0x012C	Vector Address 11 Register
VICVECTADDR12	0x0130	Vector Address 12 Register
–	0x0134	Reserved
VICVECTADDR14	0x0138	Vector Address 14 Register
–	0x013C	Reserved
VICVECTADDR16	0x0140	Vector Address 16 Register
VICVECTADDR17	0x0144	Vector Address 17 Register
VICVECTADDR18	0x0148	Vector Address 18 Register
–	0x014C	Reserved
VICVECTADDR20	0x0150	Vector Address 20 Register
VICVECTADDR21	0x0154	Vector Address 21 Register
–	0x0158	Reserved
VICVECTADDR23	0x015C	Vector Address 23 Register
–	0x0160	Reserved
–	0x0164	Reserved
–	0x0168	Reserved
VICVECTADDR27	0x016C	Vector Address 27 Register
–	0x0170	Reserved
–	0x0174	Reserved
VICVECTADDR30	0x0178	Vector Address 30 Register
VICVECTADDR31	0x017C	Vector Address 31 Register

表 3.7.2 SFR 一覧表 (2/2)

Register Name	Address (base+)	Description
VICVECTPRIORITY0	0x0200	Vector Priority 0 Register
VICVECTPRIORITY1	0x0204	Vector Priority 1 Register
VICVECTPRIORITY2	0x0208	Vector Priority 2 Register
VICVECTPRIORITY3	0x020C	Vector Priority 3 Register
VICVECTPRIORITY4	0x0210	Vector Priority 4 Register
VICVECTPRIORITY5	0x0214	Vector Priority 5 Register
VICVECTPRIORITY6	0x0218	Vector Priority 6 Register
–	0x021C	Reserved
VICVECTPRIORITY8	0x0220	Vector Priority 8 Register
–	0x0224	Reserved
VICVECTPRIORITY10	0x0228	Vector Priority 10 Register
VICVECTPRIORITY11	0x022C	Vector Priority 11 Register
VICVECTPRIORITY12	0x0230	Vector Priority 12 Register
–	0x0234	Reserved
VICVECTPRIORITY14	0x0238	Vector Priority 14 Register
–	0x023C	Reserved
VICVECTPRIORITY16	0x0240	Vector Priority 16 Register
VICVECTPRIORITY17	0x0244	Vector Priority 17 Register
VICVECTPRIORITY18	0x0248	Vector Priority 18 Register
–	0x024C	Reserved
VICVECTPRIORITY20	0x0250	Vector Priority 20 Register
VICVECTPRIORITY21	0x0254	Vector Priority 21 Register
–	0x0258	Reserved
VICVECTPRIORITY23	0x025C	Vector Priority 23 Register
–	0x0260	Reserved
–	0x0264	Reserved
–	0x0268	Reserved
VICVECTPRIORITY27	0x026C	Vector Priority 27 Register
–	0x0270	Reserved
–	0x0274	Reserved
VICVECTPRIORITY30	0x0278	Vector Priority 30 Register
VICVECTPRIORITY31	0x027C	Vector Priority 31 Register
VICADDRESS	0x0F00	Vector Address Register

1. VICIRQSTATUS (IRQ Status Register)

Address = (0xF400_0000) + 0x0000

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	IRQStatus	RO	0x00000000	マスク後の IRQ 割り込みステータス: (各ビット) 0y0: 割り込みなし 0y1: 割り込み有り

(個別説明)

a. <IRQStatus>

マスク後の IRQ 割り込みのステータスレジスタです。

IRQStatus[31:0]の各ビットは割り込み要因 Num(31~0)に対応します。

各割り込み要因については表 3.7.1 割り込み要因一覧表を参照ください。

例: 本レジスタ bit0 = 1, 割り込み要因 Num= 0, 割り込み要因は WDT。

2. VICFIQSTATUS (FIQ Status Register)

Address = (0xF400_0000) + 0x0004

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	FIQStatus	RO	0x00000000	マスク後の FIQ 割り込みステータス: (各ビット) 0y0: 割り込みなし 0y1: 割り込み有り

(個別説明)

a. <FIQStatus>

マスク後の FIQ 割り込みのステータスレジスタです。

FIQStatus[31:0]の各ビットは割り込み要因 Num(31~0)に対応します。

各割り込み要因については表 3.7.1 割り込み要因一覧表を参照ください。

例: 本レジスタ bit0 = 1, 割り込み要因 Num= 0, 割り込み要因は WDT。

3. VICRAWINTR (Raw Interrupt Status Register)

Address = (0xF400_0000) + 0x0008

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	RawInterrupt	RO	Undefined	マスク前の IRQ 割り込みステータス: (各ビット) 0y0: 割り込みなし 0y1: 割り込み有り

(個別説明)

a. <RawInterrupt>

マスク前の IRQ 割り込みのステータスレジスタです。

RawInterrupt [31:0]の各ビットは割り込み要因 Num(31~0)に対応します。

各割り込み要因については表 3.7.1 割り込み要因一覧表を参照ください。

例: 本レジスタ bit0 = 1, 割り込み要因 Num= 0, 割り込み要因は WDT。

4. VICINTSELECT (Interrupt Select Register)

Address = (0xF400_0000) + 0x000C

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	IntSelect	R/W	0x00000000	割り込みタイプ選択 (各ビット) 0y0: IRQ 0y1: FIQ

(個別説明)

a. <IntSelect>

割り込みタイプ選択レジスタです。

割り込み発生前にあらかじめ各割り込みのタイプを設定しておく必要があります。

IntSelect [31:0]の各ビットは割り込み要因 Num(31~0)に対応します。

各割り込み要因については表 3.7.1 割り込み要因一覧表を参照ください。

例: 本レジスタ bit0 = 1, 割り込み要因 Num= 0, 割り込み要因 WDT の割り込みタイプは FIQ。

注) 本 LSI では FIQ ソースを 1 本しかサポートしていないため、このレジスタには 1 ビットのみを“1”に設定してください。このレジスタの設定を変更したい場合は、必ず関係している割り込みを Disable した後変更してください。割り込みが動作中、かつ Enable の状態ではこのレジスタの設定を変更しないでください。

5. VICINTENABLE (Interrupt Enable Register)

Address = (0xF400_0000) + 0x0010

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	IntEnable	RO	0x00000000	割り込み Enable (各ビット) 0y0: Disable 0y1: Enable

Address = (0xF400_0000) + 0x0010

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	IntEnable	WO	0x00000000	割り込み許可(各ビット) 0y0: 無効 0y1: 許可

(個別説明)

a. <IntEnable>

- ・リード時：割り込みの Enable/Disable リードレジスタです。
- ・ライト時：割り込みの許可設定レジスタです。

本レジスタは割り込み許可に設定できますが、禁止設定はできません。割り込みの禁止設定は VICINTENCLEAR レジスタで設定してください。

IntEnable [31:0]の各ビットは割り込み要因 Num(31~0)に対応します。

各割り込み要因については表 3.7.1 割り込み要因一覧表を参照ください。

例: 本レジスタ bit0 = 1, 割り込み要因 Num= 0, 割り込み要因 WDT が Enable。

6. VICINTENCLEAR (Interrupt Enable Clear Register)

Address = (0xF400_0000) + 0x0014

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	IntEnable Clear	WO	Undefined	割り込み禁止: (各ビット) 0y0: 無効 0y1: 禁止

(個別説明)

a. <IntEnable Clear>

割り込みの禁止設定レジスタです。

VICINTENABLE レジスタの割り込み許可をクリアし、割り込みは禁止になります。

IntEnable Clear [31:0]の各ビットは割り込み要因 Num(31~0)に対応します。

各割り込み要因については表 3.7.1 割り込み要因一覧表を参照ください。

7. VICSOFTINT (Software Interrupt Register)

Address = (0xF400_0000) + 0x0018

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	SoftInt	WO	0x00000000	ソフトウェア割り込み: (各ビット) 0y0: 無効 0y1: ソフトウェア割り込み発生

Address = (0xF400_0000) + 0x0018

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	SoftInt	RO	0x00000000	ソフトウェア割り込み: (各ビット) 0y0: No Active 0y1: Active

(個別説明)

a. <SoftInt>

- ・リード時：ソフトウェア割り込みの Active/Inactive リードレジスタです。

- ・ライト時：ソフトウェア割り込み発生設定レジスタです。

各ビットに 1 を設定することにより、該当する割り込み要因のソフトウェア割り込みが発生します。

SoftInt[31:0]の各ビットは割り込み要因 Num(31~0)に対応します。

各割り込み要因については表 3.7.1 割り込み要因一覧表を参照ください。

8. VICSOFTINTCLEAR (Software Interrupt Clear Register)

Address = (0xF400_0000) + 0x001C

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	SoftIntClear	WO	Undefined	ソフトウェア割り込み禁止 (各ビット) 0y0: 無効 0y1: 禁止

(個別説明)

a. <SoftIntClear>

ソフトウェア割り込み禁止設定レジスタです。

VICSOFTINT レジスタのソフトウェア割り込みを禁止します。

SoftIntClear [31:0]の各ビットは割り込み要因 Num(31~0)に対応します。

各割り込み要因については表 3.7.1 割り込み要因一覧表を参照ください。

9. VICPROTECTION (Protection Enable Register)

Address = (0xF400_0000) + 0x0020

Bit	Bit Symbol	Type	Reset Value	Description
[31:1]	–	–	Undefined	Read as undefined. Write as zero.
[0]	Protection	R/W	0y0	プロテクトモード Enable : 0y0: Disable 0y1: Enable

(個別説明)

a. <Protection>

プロテクトモードの Enable レジスタです。

Enable に設定した場合は、特権モードでのみ割り込みコントローラのレジスタをアクセス可能になります。

このレジスタは特権モードでしか R/W できません。

10. VICSWPRIORITYMASK (Software Priority Mask Register)

Address = (0xF400_0000) + 0x0024

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined. Write as zero.
[15:0]	SWPriorityMask	R/W	0xFFFF	ソフトウェアプライオリティレベルをマスク: 0y0: マスク 0y1: マスクしない

(個別説明)

a. <SWPriorityMask>

ソフトウェアプライオリティレベルのマスク設定レジスタです。

SWPriorityMask[15:0]の各ビットは 15~0 のプライオリティレベルに対応します。

例: SWPriorityMask[15:0] = 0xFF7F, プライオリティレベル = 7 の割り込みはマスクされています。

11. VICVECTADDR0 (Vector Address 0 Register)

Address = (0xF400_0000) + 0x0100

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	VectorAddr 0	R/W	0x00000000	割り込み要因 0 の ISR のアドレス:

(個別説明)

a. <VectorAddr 0>

割り込み要因の割り込みサービスルーチン(ISR)のアドレス設定レジスタです。

このレジスタの再設定は、必ず関係する割り込みを Disable している状態で実施してください。

- VICVECTADDRn (Vector Address n Register) (n = 0~6, 8, 10~12, 14, 16~18, 20, 21, 23, 27, 30, 31)

上記のレジスタ群について、構造および説明はVICVECTADDR0 と同様のため、VICVECTADDR0 の説明を参照してください。またレジスタ名およびアドレスは表 3.7.2 SFR 一覧表を参照してください。

12. VICVECTPRIORITY0 (Vector Priority 0 Register)

Address = (0xF400_0000) + 0x0200

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	–	–	Undefined	Read as undefined. Write as zero.
[3:0]	VectPriority	R/W	0y1111	割り込み要因 0 の S/W プライオリティレベル設定: 0y0000 ~ 0y1111

(個別説明)

a. <VectPriority>

IRQ のソフトウェア(S/W)プライオリティレベルを設定します。

0y0000 が最高レベルとなり、0y1111 まで 16 レベルの設定が可能です。

なお、複数の割り込み要因に同レベルの S/W プライオリティレベルを設定した時にその割り込みが同時に発生した場合は、ハードウェア(H/W)のプライオリティが有効になります。割り込み要因の Num(0~31)が H/W のプライオリティを示し、Num 0 は最高レベルのプライオリティ、Num 31 は最低レベルのプライオリティとなります。

各割り込み要因については表 3.7.1 割り込み要因一覧表を参照ください。

- VICVECTPRIORITYn (Vector Priority n Register)(n = 0~6, 8, 10~12, 14, 16~18, 20, 21, 23, 27, 30, 31)

上記のレジスタ群について、構造および説明はVICVECTPRIORITY0 と同様のため、VICVECTPRIORITY0 の説明を参照してください。またレジスタ名およびアドレスは表 3.7.2 SFR 一覧表を参照してください。

13. VICADDRESS (Vector Address Register)

Address = (0xF400_0000) + 0x0F00

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	VectAddr	R/W	0x00000000	割り込みサービスルーチン(ISR)のアドレス値:

(個別説明)

a. <VectAddr>

割り込みサービスルーチン(ISR)のアドレス設定レジスタです。

- ・リード時: 現在 **Active** な割り込みの **ISR** アドレスが反映されています。
- ・ライト時: 任意値をライトすると、現状の割り込みをクリアします。

注) このレジスタは割り込みが発生したことを確認した上でをリードしてください。また、割り込みサービスルーチン処理終了後、このレジスタをライトしてクリアしてください。

3.8 DMAC (DMA Controller)

3.8.1 機能概要

主な機能を以下に説明します。

表 3.8.1 DMA コントローラの機能

項目	機能		概要
チャンネル数	8ch		
DMA 起動	ハードウェアスタート		16 種類の周辺 IP の DMA 要求に対応 表 3.8.2を参照ください。
	ソフトウェアスタート		DMACSoftBReq レジスタへのライトで起動
バスマスタ	32bit×2 (AHB)		DMA1, DMA2
プライオリティ	DMA チャンネル 0 (高) ~ DMA チャンネル 7(低)		ハードウェア固定
FIFO	4word × 8ch		
バス幅	8/16/32bit		ソース側、ディスティネーション側で別々に 設定可能
バーストサイズ	1/4/8/16/32/64/128/256		
転送回数	~4095 回		
アドレス	ソースアドレス	incr / no-incr	アドレス wrapping はサポートしていません。
	ディスティネーション アドレス	incr / no-incr	
エンディアン	リトルエンディアンのみをサポートします。		
転送タイプ	周辺回路(レジスタ) → 周辺回路(レジスタ) 周辺回路(レジスタ) → メモリ メモリ → 周辺回路(レジスタ) メモリ → メモリ		メモリ → メモリを選択した場合、DMA 起動 のハードウェアスタートはサポートして いません。詳細は DMACCxConfiguration レジ スタを参照してください。
割り込み機能	転送終了割り込み エラー割り込み		
特殊機能	Scatter/gather 機能		

- DMA 転送タイプの詳細

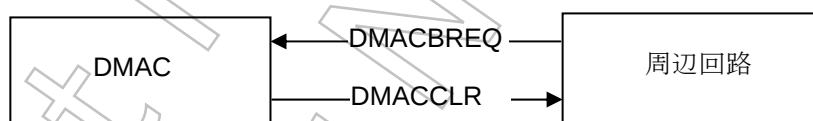
	DMA 転送方向	DMA 要求を出す回路	使用可能の DMA 要求	説明		
1	メモリ → 周辺回路	周辺回路	バースト要求	1) 全てバースト要求を使用 2) 転送はシングル要求時、DMAC のバーストを"1"に設定してください		
2	周辺回路 → メモリ	周辺回路	バースト要求/ シングル要求 注 1)	データ転送量はバーストサイズの整数倍ではない時、バーストおよびシングル両方を使います。 残る転送データの量 $\geq$ バーストサイズ: バースト転送を使用 残る転送データの量 $<$ バーストサイズ: シングル転送を使用		
3	メモリ → メモリ 注 2)	DMAC	None	開始条件: DMAC の要求は必要なく、DMAC チャンネルを Enable にするとデータ転送を開始します。 停止条件: ・ 全て転送データが転送完了 ・ DMAC チャンネルを Disable		
4	周辺回路 → 周辺回路	ソース周辺回路	バースト要求/ シングル要求 注 1)	転送サイズ	ソース側	ディスティネーション側
				1) バーストの整数倍	バースト要求	バースト要求
				2) シングル転送	シングル要求	
		ディスティネーション周辺回路	バースト要求	3) バーストの整数倍ではない	バースト要求 シングル要求	

注 1) シングル要求が対応する周辺回路 : UART および LCDDA。

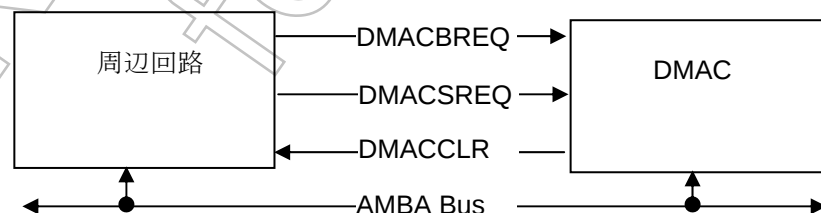
注 2) メモリ → メモリを使用して、(大量)データを転送する場合、低い Priority のチャンネルを推奨します。高い Priority のチャンネル使用を使用した場合、低い Priority チャンネルの DMA 転送が、メモリ → メモリのデータ転送終了まで待たされる可能性があるためです。

- 使用可能の DMA 要求 :

1. メモリ → 周辺回路



2. 周辺回路 → メモリ

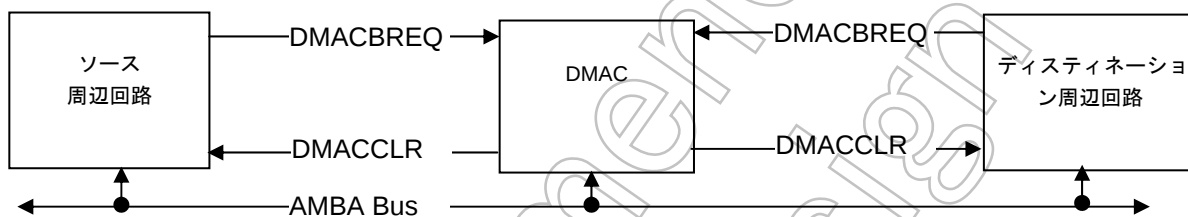


3. メモリ → メモリ

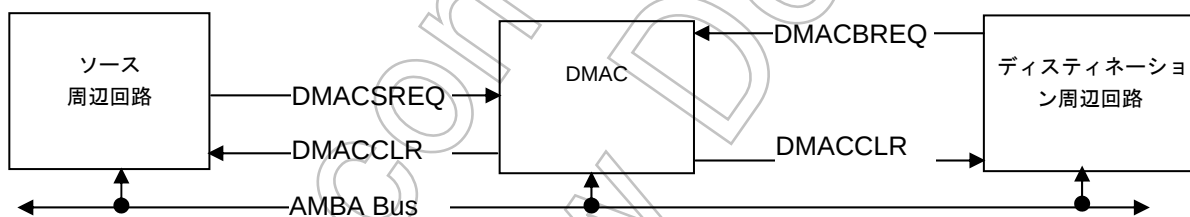


4. 周辺回路 → 周辺回路

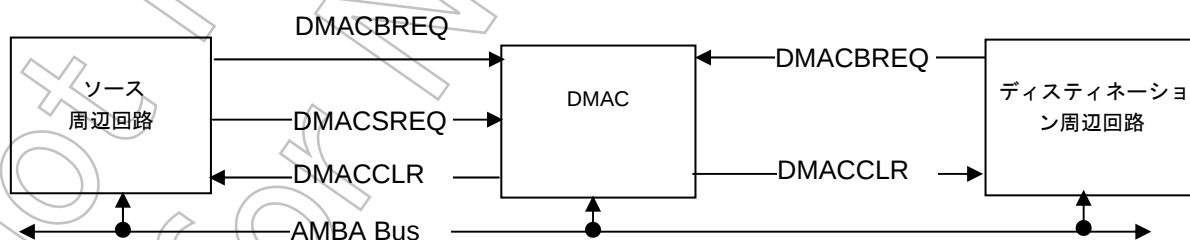
1) バーストの整数倍



2) シングル転送



3) バーストの整数倍ではない



3.8.2 ブロック図

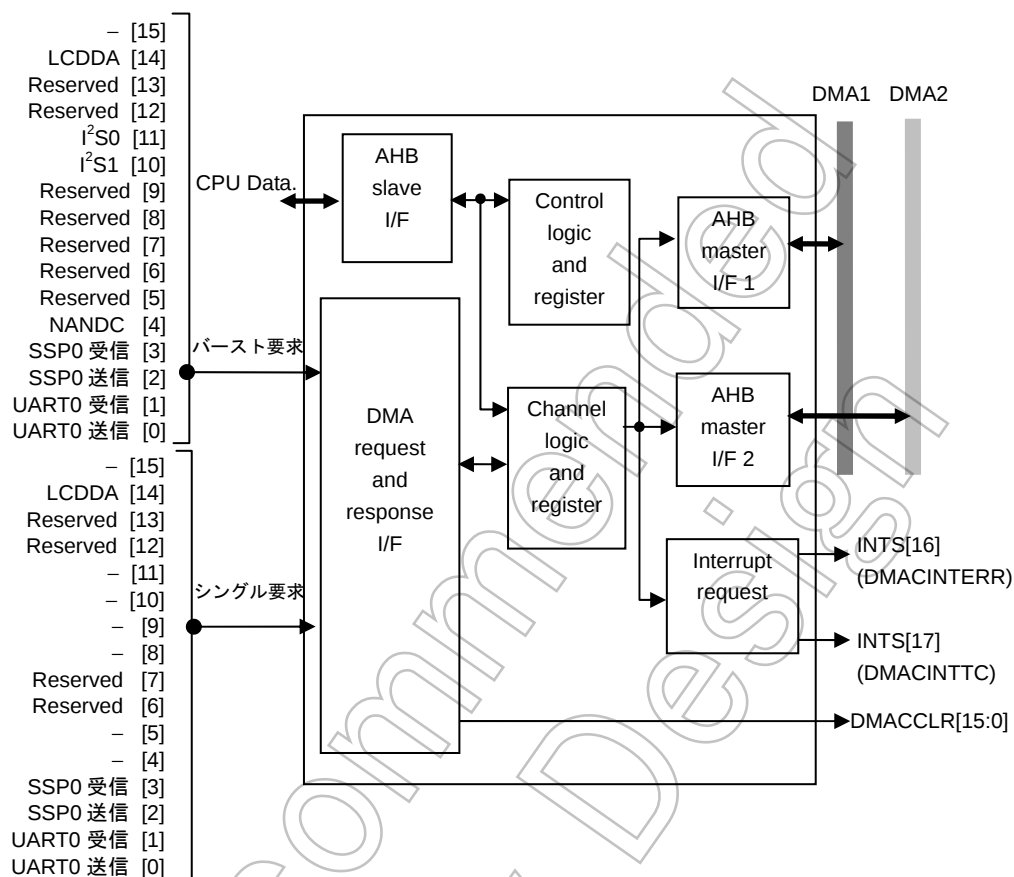


表 3.8.2 DMA リクエスト番号表

DMA リクエスト番号	対応するペリフェラル	
	バースト	シングル
0	UART0 送信	UART0 送信
1	UART0 受信	UART0 受信
2	SSP0 送信	SSP0 送信
3	SSP0 受信	SSP0 受信
4	NANDC	—
5	Reserved	—
6	Reserved	Reserved
7	Reserved	Reserved
8	Reserved	—
9	Reserved	—
10	I ² S1	—
11	I ² S0	—
12	Reserved	Reserved
13	Reserved	Reserved
14	LCDDA	LCDDA
15	—	—

3.8.3 レジスタの説明

SFR のリストと機能を以下に示します。

表 3.8.3 SFR 一覧表

base アドレス= 0xF410_0000

RegisterName	Address (base+)	Description
DMACIntStaus	0x0000	DMAC Interrupt Status Register
DMACIntTCStatus	0x0004	DMAC Interrupt Terminal Count Status Register
DMACIntTCClear	0x0008	DMAC Interrupt Terminal Count Clear Register
DMACIntErrorStatus	0x000C	DMAC Interrupt Error Status Register
DMACIntErrClr	0x0010	DMAC Interrupt Error Clear Register
DMACRawIntTCStatus	0x0014	DMAC Raw Interrupt Terminal Count Status Register
DMACRawIntErrorStatus	0x0018	DMAC Raw Error Interrupt Status Register
DMACENbldChns	0x001C	DMAC Enabled Channel Register
DMACSoftBReq	0x0020	DMAC Software Burst Request Register
DMACSoftSReq	0x0024	DMAC Software Single Request Register
–	0x0028	Reserved
–	0x002C	Reserved
DMACConfiguration	0x0030	DMAC Configuration Register
–	0x0034	Reserved
DMACC0SrcAddr	0x0100	DMAC Channel0 Source Address Register
DMACC0DestAddr	0x0104	DMAC Channel0 Destination Address Register
DMACC0LLI	0x0108	DMAC Channel0 Linked List Item Register
DMACC0Control	0x010C	DMAC Channel0 Control Register
DMACC0Configuration	0x0110	DMAC Channel0 Configuration Register
DMACC1SrcAddr	0x0120	DMAC Channel1 Source Address Register
DMACC1DestAddr	0x0124	DMAC Channel1 Destination Address Register
DMACC1LLI	0x0128	DMAC Channel1 Linked List Item Register
DMACC1Control	0x012C	DMAC Channel1 Control Register
DMACC1Configuration	0x0130	DMAC Channel1 Configuration Register
DMACC2SrcAddr	0x0140	DMAC Channel2 Source Address Register
DMACC2DestAddr	0x0144	DMAC Channel2 Destination Address Register
DMACC2LLI	0x0148	DMAC Channel2 Linked List Item Register
DMACC2Control	0x014C	DMAC Channel2 Control Register
DMACC2Configuration	0x0150	DMAC Channel2 Configuration Register
DMACC3SrcAddr	0x0160	DMAC Channel3 Source Address Register
DMACC3DestAddr	0x0164	DMAC Channel3 Destination Address Register
DMACC3LLI	0x0168	DMAC Channel3 Linked List Item Register
DMACC3Control	0x016C	DMAC Channel3 Control Register
DMACC3Configuration	0x0170	DMAC Channel3 Configuration Register
DMACC4SrcAddr	0x0180	DMAC Channel4 Source Address Register
DMACC4DestAddr	0x0184	DMAC Channel4 Destination Address Register
DMACC4LLI	0x0188	DMAC Channel4 Linked List Item Register
DMACC4Control	0x018C	DMAC Channel4 Control Register
DMACC4Configuration	0x0190	DMAC Channel4 Configuration Register
DMACC5SrcAddr	0x01A0	DMAC Channel5 Source Address Register
DMACC5DestAddr	0x01A4	DMAC Channel5 Destination Address Register
DMACC5LLI	0x01A8	DMAC Channel5 Linked List Item Register
DMACC5Control	0x01AC	DMAC Channel5 Control Register
DMACC5Configuration	0x01B0	DMAC Channel5 Configuration Register

Register Name	Address (base+)	Description
DMACC6SrcAddr	0x1C0	DMAC Channel6 Source Address Register
DMACC6DestAddr	0x1C4	DMAC Channel6 Destination Address Register
DMACC6LLI	0x1C8	DMAC Channel6 Linked List Item Register
DMACC6Control	0x1CC	DMAC Channel6 Control Register
DMACC6Configuration	0x1D0	DMAC Channel6 Configuration Register
DMACC7SrcAddr	0x1E0	DMAC Channel7 Source Address Register
DMACC7DestAddr	0x1E4	DMAC Channel7 Destination Address Register
DMACC7LLI	0x1E8	DMAC Channel7 Linked List Item Register
DMACC7Control	0x1EC	DMAC Channel7 Control Register
DMACC7Configuration	0x1F0	DMAC Channel7 Configuration Register
–	0xFE0	Reserved
–	0xFE4	Reserved
–	0xFE8	Reserved
–	0xFEC	Reserved
–	0xFF0	Reserved
–	0xFF4	Reserved
–	0xFF8	Reserved
–	0xFFC	Reserved
–	0x500	Reserved
–	0x504	Reserved
–	0x508	Reserved
–	0x50C	Reserved

注) レジスタのアクセスはワードで実施してください。

1. DMACIntStatus (DMAC Interrupt Status Register)

Address = (0xF410_0000) + 0x0000

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	—	—	Undefined	Read as undefined.
[7]	IntStatus7	RO	0y0	DMAC チャンネル 7 の割り込み発生状態 0y1: 割り込み要求有り 0y0: 割り込み要求無し
[6]	IntStatus6	RO	0y0	DMAC チャンネル 6 の割り込み発生状態 0y1: 割り込み要求有り 0y0: 割り込み要求無し
[5]	IntStatus5	RO	0y0	DMAC チャンネル 5 の割り込み発生状態 0y1: 割り込み要求有り 0y0: 割り込み要求無し
[4]	IntStatus4	RO	0y0	DMAC チャンネル 4 の割り込み発生状態 0y1: 割り込み要求有り 0y0: 割り込み要求無し
[3]	IntStatus3	RO	0y0	DMAC チャンネル 3 の割り込み発生状態 0y1: 割り込み要求有り 0y0: 割り込み要求無し
[2]	IntStatus2	RO	0y0	DMAC チャンネル 2 の割り込み発生状態 0y1: 割り込み要求有り 0y0: 割り込み要求無し
[1]	IntStatus1	RO	0y0	DMAC チャンネル 1 の割り込み発生状態 0y1: 割り込み要求有り 0y0: 割り込み要求無し
[0]	IntStatus0	RO	0y0	DMAC チャンネル 0 の割り込み発生状態 0y1: 割り込み要求有り 0y0: 割り込み要求無し

(個別説明)

a. <IntStatus[7:0]>

転送終了割り込み許可レジスタおよびエラー割り込み許可レジスタを経由した後の DMAC 割り込み発生状態。転送エラーか、カウンタ終了か、どちらでも割り込み要求が発生します。

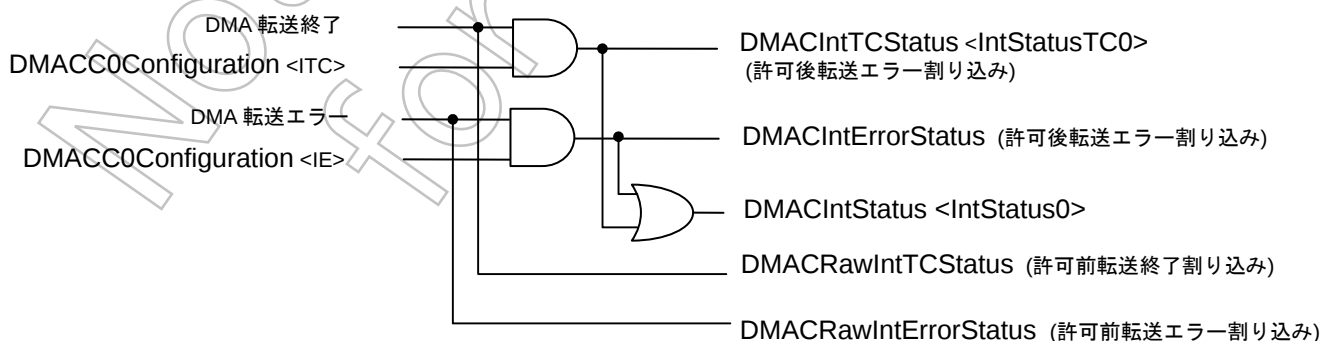


図 3.8.1 割り込み関連ブロック図

2. DMACIntTCStatus (DMAC Interrupt Terminal Count Status Register)

Address = (0xF410_0000) + 0x0004

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	—	—	Undefined	Read as undefined.
[7]	IntStatusTC7	RO	0y0	DMAC チャネル 7 の転送終了割り込み発生状態 0y1: 割り込み要求有り 0y0: 割り込み要求無し
[6]	IntStatusTC6	RO	0y0	DMAC チャネル 6 の転送終了割り込み発生状態 0y1: 割り込み要求有り 0y0: 割り込み要求無し
[5]	IntStatusTC5	RO	0y0	DMAC チャネル 5 の転送終了割り込み発生状態 0y1: 割り込み要求有り 0y0: 割り込み要求無し
[4]	IntStatusTC4	RO	0y0	DMAC チャネル 4 の転送終了割り込み発生状態 0y1: 割り込み要求有り 0y0: 割り込み要求無し
[3]	IntStatusTC3	RO	0y0	DMAC チャネル 3 の転送終了割り込み発生状態 0y1: 割り込み要求有り 0y0: 割り込み要求無し
[2]	IntStatusTC2	RO	0y0	DMAC チャネル 2 の転送終了割り込み発生状態 0y1: 割り込み要求有り 0y0: 割り込み要求無し
[1]	IntStatusTC1	RO	0y0	DMAC チャネル 1 の転送終了割り込み発生状態 0y1: 割り込み要求有り 0y0: 割り込み要求無し
[0]	IntStatusTC0	RO	0y0	DMAC チャネル 0 の転送終了割り込み発生状態 0y1: 割り込み要求有り 0y0: 割り込み要求無し

(個別説明)

a. <IntStatusTC[7:0]>

許可後の転送終了割り込み発生状態です。

3. DMACIntTCClear (DMAC Interrupt Terminal Count Clear Register)

Address = (0xF410_0000) + 0x0008

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	—	—	Undefined	Read as undefined. Write as zero.
[7]	IntTCClear7	WO	0y0	DMAC チャンネル 7 の転送終了割り込みクリア 0y1: クリア 0y0: 無効
[6]	IntTCClear6	WO	0y0	DMAC チャンネル 6 の転送終了割り込みクリア 0y1: クリア 0y0: 無効
[5]	IntTCClear5	WO	0y0	DMAC チャンネル 5 の転送終了割り込みクリア 0y1: クリア 0y0: 無効
[4]	IntTCClear4	WO	0y0	DMAC チャンネル 4 の転送終了割り込みクリア 0y1: クリア 0y0: 無効
[3]	IntTCClear3	WO	0y0	DMAC チャンネル 3 の転送終了割り込みクリア 0y1: クリア 0y0: 無効
[2]	IntTCClear2	WO	0y0	DMAC チャンネル 2 の転送終了割り込みクリア 0y1: クリア 0y0: 無効
[1]	IntTCClear1	WO	0y0	DMAC チャンネル 1 の転送終了割り込みクリア 0y1: クリア 0y0: 無効
[0]	IntTCClear0	WO	0y0	DMAC チャンネル 0 の転送終了割り込みクリア 0y1: クリア 0y0: 無効

(個別説明)

a. <IntTCClear[7:0]>

各チャンネルに対応したビットに1をライトすると、DMACIntTCStatus レジスタの対応するビットがクリアされます。

4. DMACIntErrorStatus (DMAC Interrupt Error Status Register)

Address = (0xF410_0000) + 0x000C

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	—	—	Undefined	Read as undefined.
[7]	IntErrStatus7	RO	0y0	DMAC チャンネル 7 のエラー割り込み発生状態 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[6]	IntErrStatus6	RO	0y0	DMAC チャンネル 6 のエラー割り込み発生状態 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[5]	IntErrStatus5	RO	0y0	DMAC チャンネル 5 のエラー割り込み発生状態 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[4]	IntErrStatus4	RO	0y0	DMAC チャンネル 4 のエラー割り込み発生状態 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[3]	IntErrStatus3	RO	0y0	DMAC チャンネル 3 のエラー割り込み発生状態 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[2]	IntErrStatus2	RO	0y0	DMAC チャンネル 2 のエラー割り込み発生状態 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[1]	IntErrStatus1	RO	0y0	DMAC チャンネル 1 のエラー割り込み発生状態 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[0]	IntErrStatus0	RO	0y0	DMAC チャンネル 0 のエラー割り込み発生状態 0y0: 割り込み要求無し 0y1: 割り込み要求有り

(個別説明)

a. <IntErrStatus[7:0]>

許可後のエラー割り込み状態です。

5. DMACIntErrClr (DMAC Interrupt Error Clear Register)

Address = (0xF410_0000) + 0x0010

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	—	—	Undefined	Read as undefined. Write as zero.
[7]	IntErrClr7	WO	0y0	DMAC チャンネル 7 のエラー割り込みクリア 0y0: 無効 0y1: クリア
[6]	IntErrClr6	WO	0y0	DMAC チャンネル 6 のエラー割り込みクリア 0y0: 無効 0y1: クリア
[5]	IntErrClr5	WO	0y0	DMAC チャンネル 5 のエラー割り込みクリア 0y0: 無効 0y1: クリア
[4]	IntErrClr4	WO	0y0	DMAC チャンネル 4 のエラー割り込みクリア 0y0: 無効 0y1: クリア
[3]	IntErrClr3	WO	0y0	DMAC チャンネル 3 のエラー割り込みクリア 0y0: 無効 0y1: クリア
[2]	IntErrClr2	WO	0y0	DMAC チャンネル 2 のエラー割り込みクリア 0y0: 無効 0y1: クリア
[1]	IntErrClr1	WO	0y0	DMAC チャンネル 1 のエラー割り込みクリア 0y0: 無効 0y1: クリア
[0]	IntErrClr0	WO	0y0	DMAC チャンネル 0 のエラー割り込みクリア 0y0: 無効 0y1: クリア

(個別説明)

a. <IntErrClr[7:0]>

0y1: エラー割り込み要求をクリアします。

6. DMACRawIntTCStatus (DMAC Raw Interrupt Terminal Count Status Register)

Address = (0xF410_0000) + 0x0014

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	—	—	Undefined	Read as undefined.
[7]	RawIntTCS7	RO	0y0	DMAC チャンネル 7 の許可前転送終了割り込み発生状態 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[6]	RawIntTCS6	RO	0y0	DMAC チャンネル 6 の許可前転送終了割り込み発生状態 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[5]	RawIntTCS5	RO	0y0	DMAC チャンネル 5 の許可前転送終了割り込み発生状態 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[4]	RawIntTCS4	RO	0y0	DMAC チャンネル 4 の許可前転送終了割り込み発生状態 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[3]	RawIntTCS3	RO	0y0	DMAC チャンネル 3 の許可前転送終了割り込み発生状態 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[2]	RawIntTCS2	RO	0y0	DMAC チャンネル 2 の許可前転送終了割り込み発生状態 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[1]	RawIntTCS1	RO	0y0	DMAC チャンネル 1 の許可前転送終了割り込み発生状態 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[0]	RawIntTCS0	RO	0y0	DMAC チャンネル 0 の許可前転送終了割り込み発生状態 0y0: 割り込み要求無し 0y1: 割り込み要求有り

(個別説明)

a. <RawIntTCS[7:0]>

許可前の転送終了割り込み発生状態です。

7. DMACRawIntErrorStatus (DMAC Raw Error Interrupt Status Register)

Address = (0xF410_0000) + 0x0018

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	—	—	Undefined	Read as undefined.
[7]	RawIntErrS7	RO	0y0	DMAC チャンネル 7 の許可前エラー割り込み発生状態 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[6]	RawIntErrS6	RO	0y0	DMAC チャンネル 6 の許可前エラー割り込み発生状態 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[5]	RawIntErrS5	RO	0y0	DMAC チャンネル 5 の許可前エラー割り込み発生状態 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[4]	RawIntErrS4	RO	0y0	DMAC チャンネル 4 の許可前エラー割り込み発生状態 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[3]	RawIntErrS3	RO	0y0	DMAC チャンネル 3 の許可前エラー割り込み発生状態 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[2]	RawIntErrS2	RO	0y0	DMAC チャンネル 2 の許可前エラー割り込み発生状態 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[1]	RawIntErrS1	RO	0y0	DMAC チャンネル 1 の許可前エラー割り込み発生状態 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[0]	RawIntErrS0	RO	0y0	DMAC チャンネル 0 の許可前エラー割り込み発生状態 0y0: 割り込み要求無し 0y1: 割り込み要求有り

(個別説明)

a. <RawIntErrS[7:0]>

許可前のエラー割り込み発生状態です。

8. DMACEnbldChns (DMAC Enabled Channel Register)

Address = (0xF410_0000) + 0x001C

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	—	—	Undefined	Read as undefined.
[7]	EnabledCH7	RO	0y0	DMA チャンネル 7 のイネーブル状態 0y0: ディゼーブル 0y1: イネーブル
[6]	EnabledCH6	RO	0y0	DMA チャンネル 6 のイネーブル状態 0y0: ディゼーブル 0y1: イネーブル
[5]	EnabledCH5	RO	0y0	DMA チャンネル 5 のイネーブル状態 0y0: ディゼーブル 0y1: イネーブル
[4]	EnabledCH4	RO	0y0	DMA チャンネル 4 のイネーブル状態 0y0: ディゼーブル 0y1: イネーブル
[3]	EnabledCH3	RO	0y0	DMA チャンネル 3 のイネーブル状態 0y0: ディゼーブル 0y1: イネーブル
[2]	EnabledCH2	RO	0y0	DMA チャンネル 2 のイネーブル状態 0y0: ディゼーブル 0y1: イネーブル
[1]	EnabledCH1	RO	0y0	DMA チャンネル 1 のイネーブル状態 0y0: ディゼーブル 0y1: イネーブル
[0]	EnabledCH0	RO	0y0	DMA チャンネル 0 のイネーブル状態 0y0: ディゼーブル 0y1: イネーブル

(個別説明)

a. <EnabledCH[7:0]>

0y0: DMA 転送終了時、該当チャンネルのビットがクリアされています。

0y1: 該当チャンネル DMA が Enable 状態です。

9. DMACSoftBReq (DMAC Software Burst Request Register)

Address = (0xF410_0000) + 0x0020

Bit	Bit Symbol	Type	Reset Value	Description
[31:15]	–	–	Undefined	Read as undefined. Write as zero.
[14]	SoftBReq14	R/W	0y0	ソフトウェアによる LCDDA の DMA バースト要求の発生 0y0: 無効(WR) 0y1: DMA バースト要求の発生
[13:10]	Reserved	R/W	Undefined	Read as undefined. Write as zero.
[11]	SoftBReq11	R/W	0y0	ソフトウェアによる I ² S0 の DMA バースト要求の発生 0y0: 無効(WR) 0y1: DMA バースト要求の発生
[10]	SoftBReq10	R/W	0y0	ソフトウェアによる I ² S1 の DMA バースト要求の発生 0y0: 無効(WR) 0y1: DMA バースト要求の発生
[9:5]	Reserved	R/W	Undefined	Read as undefined. Write as zero.
[4]	SoftBReq4	R/W	0y0	ソフトウェアによる NANDC0 の DMA バースト要求の発生 0y0: 無効(WR) 0y1: DMA バースト要求の発生
[3]	SoftBReq3	R/W	0y0	ソフトウェアによる SSP0 受信の DMA バースト要求の発生 0y0: 無効(WR) 0y1: DMA バースト要求の発生
[2]	SoftBReq2	R/W	0y0	ソフトウェアによる SSP0 送信の DMA バースト要求の発生 0y0: 無効(WR) 0y1: DMA バースト要求の発生
[1]	SoftBReq1	R/W	0y0	ソフトウェアによる UART0 受信の DMA バースト要求の発生 0y0: 無効(WR) 0y1: DMA バースト要求の発生
[0]	SoftBReq0	R/W	0y0	ソフトウェアによる UART0 送信の DMA バースト要求の発生 0y0: 無効(WR) 0y1: DMA バースト要求の発生

(個別説明)

a. <SoftBReq[14:0]>

ソフトウェアによる DMA バースト転送要求を設定します。ソフトウェアによる DMA バースト転送が終了すると SoftBReq[14:0]の該当ビットがクリアされます。

リードした場合はバースト要求の状態がリードされます。(周辺回路からの要求も含まれます)。

注) 同時にソフトウェアとハードウェアペリフェラルによる DMA 要求を実施しないでください。

10. DMACSoftSReq (DMAC Software Single Request Register)

Address = (0xF410_0000) + 0x0024

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	—	—	Undefined	Read as undefined. Write as zero.
[15]	Reserved	R/W	Undefined	Read as undefined. Write as zero.
[14]	SoftSReq14	R/W	0y0	ソフトウェアによる LCDDA の DMA シングル要求の発生 0y0: 無効(WR) 0y1: DMA シングル要求の発生
[13:4]	Reserved	R/W	Undefined	Read as undefined. Write as zero.
[3]	SoftSReq3	R/W	0y0	ソフトウェアによる SSP0 受信の DMA シングル要求の発生 0y0: 無効(WR) 0y1: DMA シングル要求の発生
[2]	SoftSReq2	R/W	0y0	ソフトウェアによる SSP0 送信の DMA シングル要求の発生 0y0: 無効(WR) 0y1: DMA シングル要求の発生
[1]	SoftSReq1	R/W	0y0	ソフトウェアによる UART0 受信の DMA シングル要求の発生 0y0: 無効(WR) 0y1: DMA シングル要求の発生
[0]	SoftSReq0	R/W	0y0	ソフトウェアによる UART0 送信の DMA シングル要求の発生 0y0: 無効(WR) 0y1: DMA シングル要求の発生

(個別説明)

a. <SoftSReq[14:0]>

ソフトウェアによる DMA シングル転送要求を設定します。ソフトウェアによる DMA シングル転送が終了すると SoftSReq[14:0]の該当ビットがクリアされます。

リードした場合はシングル要求の状態がリードされます。(周辺回路からの要求も含まれます)。

注) ハードウェアペリフェラルによる DMA 要求が発生しているときに、同時にソフトウェアの DMA 要求を実施しないでください。

11. DMACConfiguration (DMAC Configuration Register)

Address = (0xF410_0000) + 0x0030

Bit	Bit Symbol	Type	Reset Value	Description
[31:3]	—	—	Undefined	Read as undefined. Write as zero.
[2]	M2	R/W	0y0	DMA2 エンディアン区別: 0y0: リトルエンディアン 0y1: Reserved
[1]	M1	R/W	0y0	DMA1 エンディアン区別: 0y0: リトルエンディアン 0y1: Reserved
[0]	E	R/W	0y0	DMA 回路制御: 0y0: 停止 0y1: 動作

(個別説明)

a. <E>

DMA 回路を動作状態にしないと、DMA 回路のレジスタに書き込み、読み出しが出来ません。
DMA を動作させる場合には常に動作状態にしてください。

12. DMACC0SrcAddr (DMAC Channel0 Source Address Register)

Address = (0xF410_0000) + 0x0100

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	SrcAddr	R/W	0x00000000	DMA 転送元アドレスの設定

(個別説明)

a. <SrcAddr>

チャンネルを **Enable** するとレジスタに記述された内容が更新されますので、チャンネルを **Enable** する前に **DMACCxSrcAddr** を設定してください。

DMA が動作中の場合、**DMACCxSrcAddr** レジスタの値は逐次変化するため、リード値は固定ではありません。

転送中 **DMACC0SrcAddr** をアップデートしないでください。変更する場合は必ず **DMACCxConfiguration** レジスタ設定でチャンネルを **Disable** した後、変更してください。

- **DMACCxSrcAddr (DMAC Channel x Source Address Register) (x = 0~7)**

上記のレジスタ群について、構造および説明は **DMACC0SrcAddr** と同様のため、**DMACC0SrcAddr** の説明を参照してください。またレジスタ名およびアドレスは表 3.8.3 SFR 一覧表を参照してください。

13. DMACC0DestAddr (DMAC Channel0 Destination Address Register)

Address = (0xF410_0000) + 0x0104

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	DestAddr	R/W	0x00000000	DMA 転送先アドレスの設定

(個別説明)

a. <DestAddr>

転送中 DMACC0DestAddr をアップデートしないでください。変更する場合は必ず DMACCxConfiguration レジスタ設定で、チャンネルを Disable して、変更してください。

• DMACCxDestAddr (DMAC Channel x Destination Address Register) (x = 0~7)

上記のレジスタ群について、構造および説明は DMACC0DestAddr と同様のため、DMACC0DestAddr の説明を参照してください。またレジスタ名およびアドレスは 表 3.8.3 SFR 一覧表を参照してください。

14. DMACC0LLI (DMAC Channel0 Linked List Item Register)

Address = (0xF410_0000) + 0x0108

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	LLI	R/W	0x00000000	次の転送情報の先頭アドレスを設定。
[1]	—	—	Undefined	Read as undefined. Write as zero.
[0]	LM	R/W	0y0	LLI 格納先 AHB Master 選択: 0y0: DMA1 0y1: DMA2

(個別説明)

a. <LLI>

<LLI>の設定値は 0xFFFF_FFF0 以内に設定してください。

<LLI> = 0 時、現状 LLI は最後チェーンであり、最後 DMA 転送が終了後、DMA チャンネルは Disable になります。

• DMACCxLLI (DMAC Channel x Linked List Item Register) (x = 0~7)

上記のレジスタ群について、構造および説明はDMACC0LLIと同様のため、DMACC0LLIの説明を参照してください。またレジスタ名およびアドレスは表 3.8.3 SFR一覧表を参照してください。

15. DMACC0Control (DMAC Channel0 Control Register)

Address = (0xF410_0000) + 0x010C

Bit	Bit Symbol	Type	Reset Value	Description
[31]	I	R/W	0y0	Scatter/gather 機能使用時の転送終了割り込み許可レジスタ 0y0: Disable 0y1: Enable
[30]	Prot[3]	R/W	0y0	キャッシュ可否 HPROT[3]を制御 0y0: 不可 0y1: キャッシュ可能
[29]	Prot[2]	R/W	0y0	バッファ可否 HPROT[2]を制御 0y0: 不可 0y1: バッファ可能
[28]	Prot[1]	R/W	0y0	特権モード HPROT[1]を制御 0y0: ユーザー 0y1: 特権
[27]	DI	R/W	0y0	転送先アドレスインクリメント 0y0: アドレス固定 0y1: インクリメント
[26]	SI	R/W	0y0	転送元アドレスインクリメント 0y0: アドレス固定 0y1: インクリメント
[25]	D	R/W	0y0	転送先 AHB Master 0y0: DMA1 0y1: DMA2
[24]	S	R/W	0y0	転送元 AHB Master 0y0: DMA1 0y1: DMA2
[23:21]	Dwidth[2:0]	R/W	0y000	転送先ビット幅 0y000: バイト(8 ビット) 0y001: ハーフワード(16 ビット) 0y010: ワード(32 ビット) other: Reserved
[20:18]	Swidth[2:0]	R/W	0y000	転送元ビット幅 0y000: バイト(8 ビット) 0y001: ハーフワード(16 ビット) 0y010: ワード(32 ビット) other: Reserved
[17:15]	DBSize[2:0]	R/W	0y000	転送先バーストサイズ : 0y000: 1 ビート 0y001: 4 ビート 0y010: 8 ビート 0y011: 16 ビート 0y100: 32 ビート 0y101: 64 ビート 0y110: 128 ビート 0y111: 256 ビート
[14:12]	SBSIZE[2:0]	R/W	0y000	転送元バーストサイズ : 0y000: 1 ビート 0y001: 4 ビート 0y010: 8 ビート 0y011: 16 ビート 0y100: 32 ビート 0y101: 64 ビート 0y110: 128 ビート 0y111: 256 ビート
[11:0]	TransferSize	R/W	0x000	総転送回数の設定

(個別説明) 他のチャネルは下記の説明と同様です。

a. <I>

転送終了割り込みの許可レジスタです。

<I>=1、且つ、DMACCxConfiguration Register<ITC>=1 の設定で、転送終了割り込みが発生します。Scatter/gather 機能使用時に、最終転送の DMAC 設定フロー内で、本 bit を Enable にすることで、最終転送時にのみ転送終了割り込みを発生することが可能になります。

通常転送時に割り込みを発生させたいときには、本 bit も”1”に設定し Enable 状態にする必要があります。

b. <Swidth[2:0]>

転送サイズは転送先ビット幅の整数倍になるように、転送先のビット幅を設定してください。

c. <DBSize[2:0]>

注) DBSize で設定するバーストサイズは AHB バスの HBURST とは関係ありません。

d. <SBSize[2:0]>

注) SBSize で設定するバーストサイズは AHB バスの HBURST とは関係ありません。

e. <TransferSize>

DMAC が Flow コントローラ時、総転送回数を設定。

この値は DMAC 転送を実施に伴い、“0”までにデクリメントします。リードすると未転送回数が読み出されます。

総転送回数は、転送元ビット幅の単位となる。

ex:Swidth=8bit の場合、転送回数は byte 単位

ex:Swidth=16bit の場合、転送回数は half word 単位

ex:Swidth=32bit の場合、転送回数は word 単位

注) 転送元ビット幅が転送先ビット幅よりも小さい場合は、総転送回数を設定する場合に注意が必要です。
以下の計算式を満たすように設定してください。

転送元ビット幅 × 総転送回数 = 転送先ビット幅 × N
N: 整数

- DMACCxControl (DMAC Channel x Control Register) (x = 0~7)

上記のレジスタ群について、構造および説明はDMACC0Controlと同様のため、DMACC0Controlの説明を参照してください。またレジスタ名およびアドレスは表 3.8.3 SFR 一覧表を参照してください。

16. DMACC0Configuration (DMAC Channel0 Configuration Register)

Address = (0xF410_0000) + 0x0110

Bit	Bit Symbol	Type	Reset Value	Description										
[31:19]	–	–	Undefined	Read as undefined. Write as zero.										
[18]	Halt	R/W	0y0	0y0: DMA 要求 受付 0y1: DMA 要求 無視										
[17]	Active	RO	0y0	リード時: 0y0: FIFO 内にデータ無 0y1: FIFO 内にデータ有 ライト時: 無効										
[16]	Lock	R/W	0y0	0y0: ロック転送 禁止 0y1: ロック転送 許可										
[15]	ITC	R/W	0y0	転送終了割り込み許可レジスタ 0y0: 割り込み禁止 0y1: 割り込み許可										
[14]	IE	R/W	0y0	エラー割り込み許可レジスタ 0y0: 割り込み禁止 0y1: 割り込み許可										
[13:11]	FlowCntrl	R/W	0y000	<table><tr><th>FlowCntrl 設定値</th><th>転送方式</th></tr><tr><td>0y000</td><td>Memory to Memory</td></tr><tr><td>0y001</td><td>Memory to Peripheral</td></tr><tr><td>0y010</td><td>Peripheral to Memory</td></tr><tr><td>0y011</td><td>Peripheral to Peripheral</td></tr></table> 0y100~0y111: Reserved	FlowCntrl 設定値	転送方式	0y000	Memory to Memory	0y001	Memory to Peripheral	0y010	Peripheral to Memory	0y011	Peripheral to Peripheral
FlowCntrl 設定値	転送方式													
0y000	Memory to Memory													
0y001	Memory to Peripheral													
0y010	Peripheral to Memory													
0y011	Peripheral to Peripheral													
[10]	–	–	Undefined	Read as undefined. Write as zero.										
[9:6]	DestPeripheral	R/W	0y000	転送先ペリフェラル (注) 0y000~0y1111										
[5]	–	–	Undefined	Read as undefined. Write as zero.										
[4:1]	SrcPeripheral	R/W	0y000	転送元ペリフェラル (注) 0y000~0y1111										
[0]	E	R/W	0y0	チャンネルイネーブル 0y0: Disable 0y1: Enable										

(注) DMA リクエスト番号表を参照してください。

(個別説明)

a. <ITC>

転送終了割り込みの許可レジスタです。

<ITC>=1 且つ、DMACCxControl Register<I>=1、の設定で、転送終了割り込みが発生します。

b. <FlowCtrl>

転送方式を設定するビットです。

0y000: Memory to Memory

0y001: Memory to Peripheral

0y010: Peripheral to Memory

0y011: Peripheral to Peripheral

0y100~0y111: Reserved

注) Memory to Memory を選択した場合、DMA 起動のハードウェアスタートはサポートしていません。<E>= 0y1 をライトすることで転送を開始します。

c. <DestPeripheral>

DMA リクエストのペリフェラル番号、2 進数で表現。

転送先が Memory の場合はこの設定は無視されます。

d. <SrcPeripheral>

DMA リクエストのペリフェラル番号、2 進数で表現。

転送元が Memory の場合はこの設定は無視されます。

e. <E>

このビットでチャネルを Enable/Disable できます。転送中で Disable を実行するとチャネル FIFO のデータが消失してしまいますので、再スタートする場合はチャネルをすべて初期化して、スタートしてください。

もし、一時的に停止したい場合は、<HALT>ビットで DMA 要求を停止して、<Active>ビットを“0”になるまでポーリングした後、<E>ビットでチャネルを Disable してください。

- DMACCxConfiguration (DMAC Channel x Configuration Register) (x = 0~7)

上記のレジスタ群について、構造および説明はDMACC0Configurationと同様のため、DMACC0Configurationの説明を参照してください。またレジスタ名およびアドレスは 表 3.8.3 SFR一覧表を参照してください。

- DMAC の設定フロー

(DMAC ch1 を使用してメモリから I²S 内蔵 FIFO への転送設定例)

総転送データ量 32word, Swidth=Word 単位、総転送回数=32 回

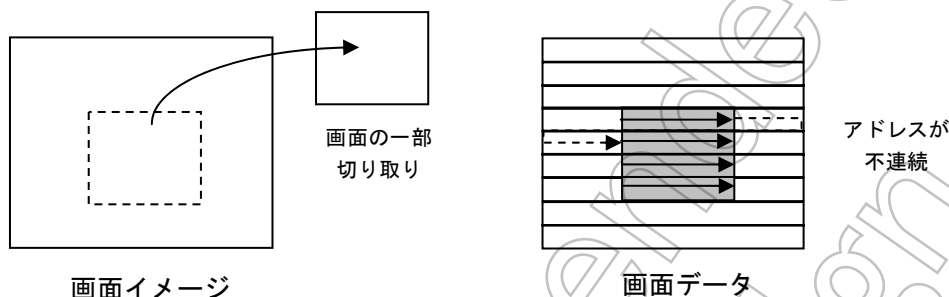
DMACConfiguration	←	0x00000001	; Set DMAC Active
DMACC1SrcAddr	←	メモリアドレス	; Source address (DMAC ch1)
DMACC1DestAddr	←	I2STDAT	; Destination address
DMACC1Control	←	0x04492020	; Destination address fixed
			; Source address increment
			; Swidth = word, Dswidth = word
			; DBSize= 8 burst, SBSIZE= 8 burst(注)
			; TransferSize = 32 回
DMACC1Configuration	←	0x00000a81	; channel1 Enable,
			; Memory to Peripheral (I2S1)
.		.	
.		.	; I ² S の設定、準備
.		.	
I2STDMA1	←	0x00000001	; I ² S DMA Ready, DMA 転送を要求

(注) Peripheral の FIFO のサイズに合わせてバーストサイズを設定してください。

3.8.4 特殊機能

1) Scatter/gather 機能

画像データの一部を切り取ってデータを転送するような場合、画像データはすべて連続データとしては扱えず、特定の規則に従ってアドレスが大きく変化します。そのため、常に連続のアドレスでしか転送出来ない DMA では、アドレスが変化する箇所、その都度再設定が必要になります。



Scatter/gather 機能とは、あらかじめ設定された”Linked list”を通じて、CPU がその動作の制御を行う必要なく、DMA の各種設定（ソースアドレス、デスティネーションアドレス、転送回数、転送バス幅）を、指定された DMA 回数を終了毎に再ロードして、連続動作することが出来る機能です。

DMACCxLLI レジスタに”1”をセットすることで動作の許可/停止を制御します。

Linked List で設定出来る項目は、以下の 4word で構成されています。

- 1) DMACCxSrcAddr
- 2) DMACCxDestAddr
- 3) DMACCxLLI
- 4) DMACCxControl

割り込み動作との併用も可能です。

DMACCxControl Register<I>=1、且つ、DMACCxConfiguration Register<ITC>=1 の設定で、DMA 転送終了割り込みが発生します。Scatter/gather 機能使用時して、DMA 最終転送の時のみ、終了割り込みを発生させたい場合は、DMACCxControl Register<I>=0、且つ、DMACCxConfiguration Register<ITC>=1 にて転送を開始し、最終回の DMA 転送設定フロー内で、<I>=1 にすることで、最終転送でのみ転送終了割り込みを発生することが可能になります。

このビットを利用することで、LLI を使った転送途中でも、条件を追加し分岐処理などの動作が可能です。割り込みをクリアするためには、DMACIntTCClear レジスタの対応 bit を制御します。

2) Liked list 動作

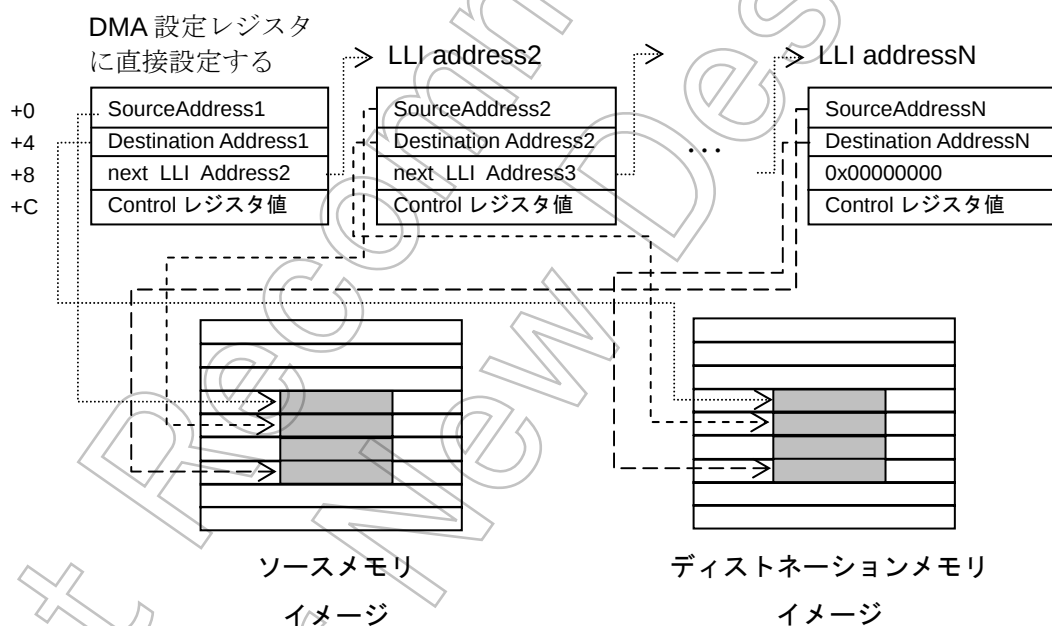
Scatter/gather 機能を動作させるには、まず一連の Linked List を作成し、ソースとディストネーションデータエリアを定義する必要があります。

各々の設定を LLI (LinkedList) と呼びます。

LLI は、1 ブロック分のデータ転送を制御しています。1 回の LLI は通常の DMA 設定を示し、連続データの転送の制御を行っています。1 回の、DMA 転送が終了するたびに、次の LLI 設定をロードし、DMA 動作の継続(Daisy Chain)をすることが出来ます。

以下に、設定例を示します。

1. 一番最初の DMA 転送設定は、DMA のレジスタに直接設定します
2. 二番目の DMA 転送以降は、“next LLI AddressX” に設定されたメモリのアドレスに書き込みます。
3. N 番目の DMA 転送で終了させる場合は、“next LLI AddressX”を 0x00000000 と設定する

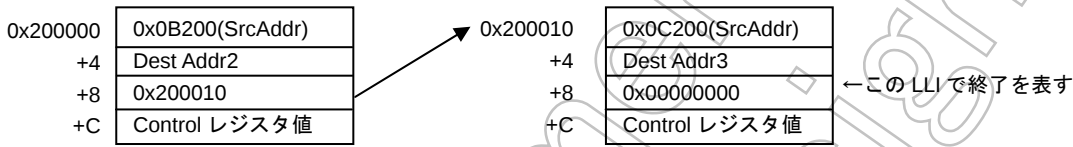


例) ソースメモリイメージの四角で囲まれたエリアを転送する場合の設定例。

	0x00200	0x00E00
0x0A000		
0x0B000		
0x0C000		

DMACCxSrcAddr : 0x0A200
DMACCxDestAddr : Destination アドレス 1
DMACCxLLI: 0x200000
DMACCxControl: バースト転送回数,転送回数などを設定

Linked List



3.9 ポート機能

各ポート端子の機能と、入出力ポート設定一覧表に各端子の設定方法を示します。
また、各々の外部端子の電源が分かれているため、あわせて電源の情報を記述しています。

表 3.9.1 TMPA901CM Pin Assign (専用端子)

電源	接続先	代表名	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	端子数	外部 INT 端子数	内部 INT ベクタ数	入出力ポート	Open-Drain 設定ポート
DVCCM	Memory	SA	D[7:0]								8				
		SB	D[15:8]								8				
		-													
		-													
		SE	A[7:0]								8				
		SF	A[15:8]								8				
		SG	A[23:16]								8				
		SH	DMCCSn	-	SMCCS1n	SMCCS0n	SMCBE0n		-	-	4				
		SJ	-	DMCCKE	DMCBA1	DMCBA0	DMCCASn	DMCRASh	DMCWEn	SMCOEn	7				
		SK	-	-	SMCBE1n	SMCWEn	-	-	DMCSDQM1	DMCSDQM0	4				
DVCC3IO DVCC1C,1B	Clock Mode	SL	-	DMCCLKIN	DMCDDQS1	DMCDDQS0	-	DMCAP	DMCDDLKIN	DMCCLKP DMCCLK	6				
		SM	AM1	AM0	-	RESETn	XT2	XT1	X2	X1	7				
		SN	HDM	HDP	-	-	-	SELJTAG	SELVCCM	SELMEMC	5				
		SP	-	-	TDO	RTCK	TRSTn	TDI	TMS	TCK	6				
		SR	-	-	-	VSENS	REXT	-	DDM	DDP	4				
DVCC3IO, AVCC3H	USB Host Mode														
DVCC3IO	JTAG														
AVDD3T/3C	USB2														

注 1) 専用ピン(ポート機能持ちません)。

注 2) 表内の代表名称“Sx”は単なるシンボルであり、汎用ポートの機能はありません。

表 3.9.2 TMPA901CM Pin Assign (ポート兼用端子)

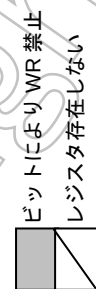
電源	接続先	代表名	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	端子数	外部 INT 端子数	内部 INT ベクタ数	入出力ポート	Open-Drain 設定ポート
DVCC3IO	Key I2C0 INT Other	PA	-	-	-	-	KI3	KI2	KI1	KI0	4	4	1	4(I)	
		PB	-	-	-	-	KO3 LCLLP	KO2 LCLFP	KO1 LCLAC	KO0 LCLCP	4			4(O)	4
		PC	I2C0DA INT9	I2C0CL	-	FSOUT PWM2OUT	MLDALM PWM0OUT	PWE	-	-	5	1	1	3(O) 2(I/O)	2
AVCC3AD	ADC TSLINT	PD	AN7 / PY INTB	AN6 / PX INTA(INTTSJ)	AN5 MY	AN4 MX	-	-	-	-	4	2	1	4(I)	
DVCC3IO	UART0	PN	-	-	-	-	-	-	U0RXD SIR0IN	U0TXD SIR0OUT	2	0	0	2(I/O)	
DVCC3IO	SSP0, USB H/D UART1	PT	X1USB	U1CTS _n I2S1DATO	U1RXD USBOC _n	U1TXD USBPON	SP0DI I2S0MCLK	SP0DO I2S0DATI	SP0CLK I2S0CLK	SP0FSS I2S0WS	8			8(I/O)	
DVCC3IO	NAND LCDC	PU	NDD7 LD7	NDD6 LD6	NDD5 LD5	NDD4 LD4	NDD3 LD3	NDD2 LD2	NDD1 LD1	NDD0 LD0	8			8(I/O)	
DVCC3IO	NAND LCDC	PV	LD15	NDRB LD14	NDCE1 _n LD13	NDCE0 _n LD12	NDCLE LD11	NDALD LD10	NDWEN LD9	NDREN LD8	8			8(I/O)	

注 1) 兼用ピン(ポート機能との兼用ピンです)。

注 2) 表内代表名称 "Px" は、汎用ポートの機能を示しています。

表 3.9.3 TMPA901CM アドレス、初期値一覧表

レジスタ名	R/W	アドレス	レジスタ説明	意味		PortA	PortB	PortC	PortD	PortN	PortT	PortU	PortV
				0	1								
GPIONDATA	R/W	0x000 -0x3FC	データレジスタ	—	—	0x0	0x0	0xEF	0xFF	0xFF	0xFF	0xFF	0xFF
GPIONDIR	R/W	0x400	データ方向レジスタ	入力ポート	出力ポート	注)	注)	0x1F	注)	0x00	0x00	0x00	0x00
GPIONFR1	R/W	0x424	ファンクションレジスタ 1	GPIO	機能 1 入力 or 出力許可	注)	0x0	0x00	0xFF	0x00	0x00	0x00	0x00
GPIONFR2	R/W	0x428	ファンクションレジスタ 2	GPIO	機能 2 入力 or 出力許可	注)	0x0	0x00	0x00	0x00	0x00	0x00	0x00
GPIONIS	R/W	0x804	割込み感知レジスタ	エッジ	レベル	0x0		0x00	0x00				
GPIONIBE	R/W	0x808	割込み面エッジレジスタ	両エッジ禁止	両エッジ許可	0x0		0x00	0x00				
GPIONIEV	R/W	0x80C	割込みイベントレジスタ	立ち下がり or L レベル	立ち上がり or H レベル	0x0		0x00	0x00				
GPIONIE	R/W	0x810	割込み許可レジスタ	禁止	許可	0x0		0x00	0x00				
GPIONRIS	RO	0x814	割込み許可前 ステータスレジスタ	要求なし	要求あり	0x0		0x00	0x00				
GPIONMIS	RO	0x818	割込み許可後 ステータスレジスタ	要求なし	要求あり	0x0		0x00	0x00				
GPIONIC	WO	0x81C	割込みクリアレジスタ	—	要求をクリア	0x0		0x00	0x00				
GPIONODE	R/W	0xC00	オーブンドレイン 出力許可レジスタ	3 ステート出力	オーブンドレイン出力		0x0	0x00	0x00				



注)Reserved: このレジスタをアクセスしないでください。

3.9.1 データレジスタ

【データレジスタについての注意事項】

全データレジスタには、全 8 ビットのデータを同時にリード/ライトする以外に、特定のビットをマスクしてリード/ライトするビットマスクの機能があります。

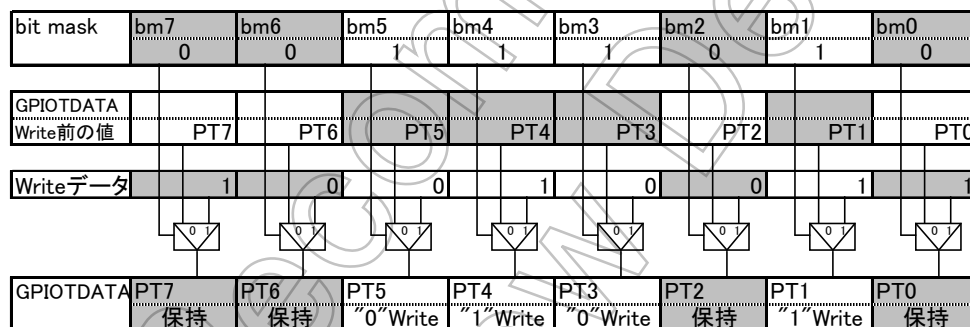
まず、データレジスタへのアクセスアドレスは、0x0000~0x03FC の 256 アドレスの空間を持っています。(アドレスが 2 ビット上位側シフトされたイメージです。下位 2 ビットのアドレスは意味を持ちません。0x0000、0x0004 といった具合に 4 アドレスごとに有効なアドレスとなります。)

この 256 アドレス空間へのアクセスはすべて、同一のデータレジスタへアクセスされますが、アクセスするアドレスに応じて、有効ビットが異なります。

アクセスするアドレスの[9:2]ビットがデータレジスタの[7:0]ビットに対応して、データをマスクします。アドレス“1”のビットがデータレジスタへアクセスされ、アドレス“0”のビットがマスクされたビットとなります。

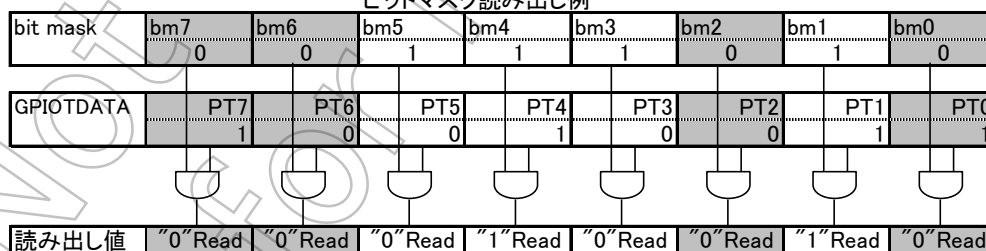
Address[9:2]	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2
Bit mask	bm7	bm6	bm5	bm4	bm3	bm2	bm1	bm0

- ビットマスク WRITE(ポート T の 0x00E8 アドレスへ、0x93 の WRITE 例)



- ビットマスク READ 例(ポート T の 0x00E8 アドレスからの 0x12 の READ 例)

ビットマスク読み出し例



注) 0x03FC へのアクセスは全ビットが有効となり、0x0000 へのアクセスでは有効なビットはありません。

3.9.2 各ポートの機能設定

本章では、汎用ポートとの切り替えが可能なポート A~ポート V の設定を、ポートごとに説明します。なお、各ポートへのアクセスはすべて **WORD (32bit)** アクセスが基本です。

3.9.2.1 ポートA

ポート A は、プルアップ付きの汎用入力端子として使用する以外に、キー入力端子としても使用できます。

割り込み許可を **Enable** とすることで、キー入力(KI3~KI0)として使用します。

プルアップ機能は使用しないことも可能です。詳細は **PMC** 章を参照ください。

汎用入力 設定

用途	データ値	割り込み許可
汎用入力	GPIOADATA	GPIOAIE
	*	0

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
				入力	入力	入力	入力

キー入力設定

用途	割り込み許可
キー入力	GPIOAIE
	1

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
				KI3	KI2	KI1	KI0

注) 割り込みは、全ビット対応、また全ビットの初期状態で Pull up 付きです。

base アドレス= 0xF080_0000

Register Name	Address (base+)	Description
GPIOADATA	0x03FC	Port A Data Register
—	0x0400	Reserved
—	0x0424	Reserved
—	0x0428	Reserved
GPIOAIS	0x0804	PortA Interrupt Selection Register (Level and Edge)
GPIOAIBE	0x0808	PortA Interrupt Selection Register (Fellow edge and Both edge)
GPIOAIEV	0x080C	PortA Interrupt Selection Register (Fall down edge/Low level and Rising up edge/High level)
GPIOAIE	0x0810	PortA Interrupt Enable Register
GPIOARIS	0x0814	PortA Interrupt Status Register (Raw)
GPIOAMIS	0x0818	PortA Interrupt Status Register (Masked)
GPIOAIC	0x081C	PortA Interrupt Clear Register
—	0x0C00	Reserved

1. GPIOADATA (Port A Data Register)

Address = (0xF080_0000) + (0x03FC)

Bit	Bit Symbol	Type	Reset Value	Bit mask	Description
[31:4]	–	–	Undefined	–	Read as undefined. Write as zero.
[3:0]	PA[3:0]	RO	0xF	Bm3:0	ポート A データレジスタ

(個別説明)

a. <PA[3:0]>

データレジスタ：データを保持するレジスタです。

ビットマスク機能については、データレジスタについての注意事項を参照してください。

2. GPIOAIS (Port A Interrupt Selection Register (Level and Edge))

Address = (0xF080_0000) + (0x0804)

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	–	–	Undefined	Read as undefined. Write as zero.
[3:0]	PA3IS~PA0IS	R/W	0x0	ポート A 割り込み感知レジスタ(各ビット) 0y0: エッジ検出 0y1: レベル検出

(個別説明)

b. <PA3IS~PA0IS>

割り込み感知レジスタ：エッジ検出とレベル検出を選択するレジスタです。

0y0: エッジ検出

0y1: レベル検出

3. GPIOAIBE (Port A Interrupt Selection Register (Fellow edge and Both edge))

Address = (0xF080_0000) + (0x0808)

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	–	–	Undefined	Read as undefined. Write as zero.
[3:0]	PA3IBE~PA0IBE	R/W	0x0	ポート A 割り込み両エッジレジスタ (各ビット) 0y0: 片エッジ 0y1: 両エッジ

(個別説明)

a. <PA3IBE~PA0IBE>

割り込み両エッジレジスタ：片エッジか両エッジか選択するレジスタです。

0y0: 片エッジ

0y1: 両エッジ

4. GPIOAIEV (Port A Interrupt Select Register (“Fall down edge/Low level” and “Rising up edge/High level”))

Address = (0xF080_0000) + (0x080C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	–	–	Undefined	Read as undefined. Write as zero.
[3:0]	PA3IEV~PA0IEV	R/W	0x0	ポート A 割り込みイベントレジスタ (各ビット) 0y0: 立ち下がりエッジ/Low レベル 0y1: 立ち上がりエッジ/High レベル

(個別説明)

a. <PA3IEV~PA0IEV>

割り込みイベントレジスタ：エッジ検出の場合、立ち下がり or 立ち上がりを制御し、レベル検出の場合、Low レベル or High レベルを選択するレジスタです。

0y0: 立ち下がりエッジ/Low レベル

0y1: 立ち上がりエッジ/High レベル

5. GPIOAIE (Port A Interrupt Enable Register)

Address = (0xF080_0000) + (0x0810)

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	–	–	Undefined	Read as undefined. Write as zero.
[3:0]	PA3IE~PA0IE	R/W	0x0	ポート A 割り込み許可レジスタ(各ビット) 0y0: 禁止 0y1: 許可

(個別説明)

a. <PA3IE~PA0IE>

割り込み許可レジスタ：割り込みの許可・禁止を制御するレジスタです。

0y0: 禁止

0y1: 許可

6. GPIOARIS (Port A Interrupt Status Register (Raw))

Address = (0xF080_0000) + (0x0814)

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	–	–	Undefined	Read as undefined.
[3:0]	PA3RIS~PA0RIS	RO	0x0	ポート A 割り込み許可前ステータスレジスタ (各ビット) 0y0: 要求なし 0y1: 要求あり

(個別説明)

a. <PA3RIS~PA0RIS>

割り込み許可前ステータスレジスタ：許可レジスタでマスクされる前の状態をモニタするレジスタです。

0y0: 要求なし

0y1: 要求あり

7. GPIOAMIS (Port A Interrupt Status Register (Masked))

Address = (0xF080_0000) + (0x0818)

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	–	–	Undefined	Read as undefined.
[3:0]	PA3MIS~PA0MIS	RO	0x0	ポート A 割り込み許可後ステータスレジスタ (各ビット) 0y0: 要求なし 0y1: 要求あり

(個別説明)

a. <PA3MIS~PA0MIS>

割り込み許可後ステータスレジスタ：許可レジスタでマスクされた後の状態をモニタするレジスタです。

0y0: 要求なし

0y1: 要求あり

8. GPIOAIC (Port A Interrupt Clear Register)

Address = (0xF080_0000) + (0x081C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	–	–	Undefined	Read as undefined. Write as zero.
[3:0]	PA3IC~PA0IC	WO	0x0	ポート A 割り込みクリアレジスタ(各ビット) 0y0: 無効 0y1: 要求クリア

(個別説明)

a. <PA3C~PA0IC>

割り込みクリアレジスタ：エッジ割り込みをクリアするレジスタです。

0y0: 無効

0y1: 要求クリア

3.9.2.2 ポートB

ポート B は汎用出力端子として使用する以外に、キー出力端子としても使用できます。
 オープンドレインを Enable とすることで、キー出力(KO3~KO0)として使用します。
 また、LCDC 制御信号として使用します。

汎用出力機能設定

用途	データ値	オープン ドレイン許可	機能切り替え 2
汎用出力	GPIOBDATA	GPIOBODE	GPIOBFR2
	*	0	0

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
				出力	出力	出力	出力

キー出力機能設定

用途	データ値	オープン ドレイン許可
キー出力	GPIOBDATA	GPIOBODE
	*	1

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
				KO3	KO2	KO1	KO0

注) オープンドレインは、ビット 3~0 対応

LCDC 制御の出力機能設定

用途	データ値	オープン ドレイン許可	機能切り替え 2
LCDC	GPIOBDATA	GPIOBODE	GPIOBFR2
	*	0	1

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
				LCLLP	LCLFP	LCLAC	LCLCP

base アドレス= 0xF080_1000

Register Name	Address (base+)	Description
GPIOBDATA	0x03FC	Port B Data Register
—	0x0400	Reserved
—	0x0424	Reserved
GPIOBFR2	0x0428	PortB Function Register2
—	0x0804	Reserved
—	0x0808	Reserved
—	0x080C	Reserved
—	0x0810	Reserved
—	0x0814	Reserved
—	0x0818	Reserved
—	0x081C	Reserved
GPIOBODE	0x0C00	Port B Open-drain Output Enable Register

1. GPIOBDATA (Port B Data Register)

Address = (0xF080_1000) + (0x03FC)

Bit	Bit Symbol	Type	Reset Value	Bit mask	Description
[31:4]	–	–	Undefined	–	Read as undefined. Write as zero.
[3:0]	PB[3:0]	R/W	0x0	Bm3:0	ポート B データレジスタ

(個別説明)

a. <PB[3:0]>

データレジスタ：データを保持するレジスタです。

ビットマスク機能については、データレジスタについての注意事項を参照してください。

2. GPIOBFR2 (Port B Function Register2)

Address = (0xF080_1000) + (0x0428)

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	–	–	Undefined	Read as undefined. Write as zero.
[3:0]	PB3F2~PB0F2	R/W	0x0	ポート B ファンクションレジスタ 2

(個別説明)

a. <PB3F2~PB0F2>

ファンクションレジスタ 2：機能切り替えの制御レジスタです。

3. GPIOBODE (Port B Open-drain Output Enable Register)

Address = (0xF080_1000) + (0x0C00)

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	–	–	Undefined	Read as undefined. Write as zero.
[3:0]	PB3DE~PB0ODE	R/W	0x0	ポート B OPD 出力許可レジスタ(各ビット) 0y0: プッシュプル出力 0y1: オープンドレイン(Pch ディセーブル) 出力

(個別説明)

a. <PB3ODE~PB0ODE>

OPD 出力許可レジスタ：プッシュプル出力か OPD 出力かを選択するレジスタです。

0y0: プッシュプル出力

0y1: オープンドレイン(Pch ディセーブル) 出力

3.9.2.3 ポートC

ポート C は、Bit[7:6]は汎用入出力端子として、Bit[4:2]は汎用出力端子として使用することが出来ます。

また、これ以外に、割り込み端子(INT9)、I²C 機能(I2C0DA,I2C0CL)、低周波クロック出力(FSOUT)、メロディ出力(MLDALM)、外部電源制御機能(PWE)、PWM 出力機能(PWM0OUT, PWM2OUT),に切り替えて使用できます。また、PC2 端子を PWE 機能として使用する際は後述の注意点を参照してください。

汎用入力・割り込み設定

用途	データ値	入出力切り替え	機能切り替え 1	機能切り替え 2	割り込み許可	オープン ドレイン許可
汎用入力	GPIOCDATA	GPIOCDIR	GPIOCFR1	GPIOCFR2	GPIOCIE	GPIOCODE
割り込み	*	0	0	0	0/1	*

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
入力/INT9	入力						

注) 割り込みは、ビット 7 み対応

汎用出力設定

用途	データ値	入出力切り替え	機能切り替え 1	機能切り替え 2	割り込み許可	オープン ドレイン許可
汎用出力	GPIOCDATA	GPIOCDIR	GPIOCFR1	GPIOCFR2	GPIOCIE	GPIOCODE
	*	1	0	0	0	0/1

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
出力	出力		出力	出力	出力		

注) オープンドレインは、ビット 7~ビット 6 対応

PWE 設定

用途	データ値	入出力切り替え	機能切り替え 1	機能切り替え 2	割り込み許可	オープン ドレイン許可	PMC レジスタ
PWE	GPIOCDATA	GPIOC DIR	GPIOCFR1	GPIOCFR2	GPIOCIE	GPIOCODE	PMCCTL <PMCPWE>
	*	*	*	*	*	*	1

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
					PWE		

I²C 設定

用途	データ値	入出力切り替え	機能切り替え 1	機能切り替え 2	割り込み許可	オープン ドレイン許可
I ² C	GPIOCDATA	GPIOC DIR	GPIOCFR1	GPIOCFR2	GPIOCIE	GPIOCODE
	*	*	1	0	0	1

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2C0DA	I2C0CL						

MLDALM, FSOUT 出力設定

用途	データ値	入出力切り替え	機能切り替え 1	機能切り替え 2	割り込み許可	オープン ドレイン許可
MLDALM	GPIOCDATA	GPIOC DIR	GPIOCFR1	GPIOCFR2	GPIOCIE	GPIOCODE
	*	*	1	0	0	0

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
			FSOUT	MLDALM			

PWM 出力設定

用途	データ値	入出力切り替え	機能切り替え 1	機能切り替え 2	割り込み許可	オープン ドレイン許可
PWM 出力	GPIOCDATA	GPIOC DIR	GPIOCFR1	GPIOCFR2	GPIOCIE	GPIOCODE
	*	*	0	1	0	0

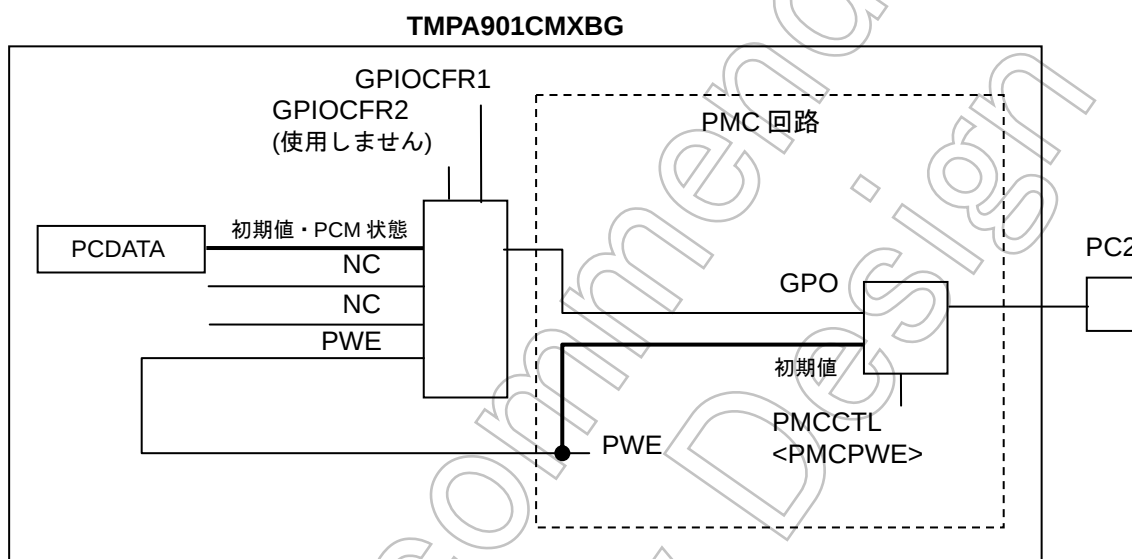
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
-	-		PWM2OUT	PWM0OUT			

- 注意点

本マイコンには電源管理回路が内蔵されており、特定の回路や外部 IO 端子以外の回路の電源をカットする機能があります(詳細は電源管理回路の章を参照ください)。

内部回路の電源回路をカットした場合でも、外部 IO の状態を保持する機能があります。その為、ポート制御をする場合、注意が必要な場合があります。中でも PC2 端子は特殊な回路構成になっていますので、制御には注意してください。

下記に内部回路の接続イメージを記載いたします。



- PC2 端子は、PCM モード(PowerCut 状態)では、汎用ポート機能は使用できません。
- PC2 端子を汎用ポートとして使用する場合、GPIOCFR1<PC2F1>=0y0、GPIOCFR2<PC2F2>=0y0 に設定し、PMU 回路内の PMUCTL<PMCPWE>=0y0、PMCWV1<PMUCTLV>=0y1 を設定後、PMCWV1<PMUCTLV>のリード値に”1”が反映(約 2.5XT1 サイクル:77μs)された場合 PMUCTL<PMCPWE>の”0”が有効となります。

設定 :	GPIOCFR1<PC2F1>=0y0
設定 :	GPIOCFR2<PC2F2>=0y0
設定 :	PMUCTL<PMCPWE>=0y0
設定 :	PMCWV1<PMUCTLV>=0y1
	約 2.5 XT1 サイクル
Read 値 :	PMCWV1<PMUCTLV>=0y1
有効 :	PMUCTL<PMCPWE>=0y0

注) 低速クロック(32KHz)を使用しない場合、PC2 端子は汎用出力として使用できません。

base アドレス = 0xF080_2000

Register Name	Address (base+)	Description
GPIOCDATA	0x03FC	PortC Data Register
GPIODIR	0x0400	PortC Data Direction Register
GPIOCFR1	0x0424	PortC Function Register1
GPIOCFR2	0x0428	PortC Function Register2
GPIOCIS	0x0804	PortC Interrupt Selection Register (Level and Edge)
GPIOCIBE	0x0808	PortC Interrupt Selection Register (Fellow edge and Both edge)
GPIOCIEV	0x080C	PortC Interrupt Selection Register (Fall down edge/Low level and Rising up edge/High level)
GPIOCIE	0x0810	PortC Interrupt Enable Register
GPIOCRIS	0x0814	PortC Interrupt Status Register (Raw)
GPIOCMIS	0x0818	PortC Interrupt Status Register (Masked)
GPIOCIC	0x081C	PortC Interrupt Clear Register
GPIOCODE	0x0C00	PortC Open-drain Output Enable Register

1. GPIOCDATA (Port C Data Register)

Address = (0xF080_2000) + (0x03FC)

Bit	Bit Symbol	Type	Reset Value	Bit mask	Description
[31:8]	–	–	Undefined	–	Read as undefined. Write as zero.
[7:6]	PC[7:6]	R/W	0y11	Bm7:6	ポート C データレジスタ
[5]	–	–	Undefined	–	Read as undefined. Write as zero.
[4:2]	PC[4:2]	R/W	0y011	Bm4:2	ポート C データレジスタ
[1:0]	–	–	Undefined	–	Read as undefined. Write as zero.

(個別説明)

a. <PC7,PC6,PC4,PC3,PC2>

データレジスタ : データを保持するレジスタです。

ビットマスク機能については、データレジスタについての注意事項を参照してください。

2. GPIOCDIR (Port C Data Direction Register)

Address = (0xF080_2000) + (0x0400)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:6]	PC7C~PC6C	R/W	0y00	ポート C データ方向レジスタ(各ビット) 0y0: 入力 0y1: 出力
[5]	–	–	Undefined	Read as undefined. Write as zero.
[4:2]	PC4C~PC0C	–	0y111	“1”をライトしてください。 リードすると“1”になります。
[1:0]	–	–	Undefined	Read as undefined. Write as zero.

(個別説明)

a. <PC7C~PC6C>

データ方向レジスタ：汎用ポートとして使用するときの各ピンの入出力を制御するレジスタです。

0y0: 入力, 0y1: 出力

3. GPIOCFR1(Port C Function Register1)

Address = (0xF080_2000) + (0x0424)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:6]	PC7F1~PC6F1	R/W	0y00	ポート C ファンクションレジスタ 1
[5]	Reserved	R/W	0y0	常に“0”をライトしてください。 リードすると“0”になります。
[4:2]	PC4F1~PC2F1	R/W	0y000	ポート C ファンクションレジスタ 1
[1:0]	Reserved	R/W	0y00	常に“0”をライトしてください。 リードすると“0”になります。

(個別説明)

a. <PC7F1~PC6F1, PC4F1~PC2F1>

ファンクションレジスタ 1：機能切り替えの制御レジスタです。

4. GPIOCFR2 (Port C Function Register2)

Address = (0xF080_2000) +(0x0428)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:5]	Reserved	R/W	0y000	常に“0”をライトしてください。 リードすると“0”になります。
[4:3]	PC4F2~PC3F2	R/W	0y00	ポート C ファンクションレジスタ 2
[2:0]	Reserved	R/W	0y000	常に“0”をライトしてください。 リードすると“0”になります。

(個別説明)

a. <PC4F2~PC3F2>

ファンクションレジスタ 2 : 機能切り替えの制御レジスタです。

(注意) ファンクションレジスタ 1 とファンクションレジスタ 2 は、排他的にしか設定することは出来ません。瞬時でも、同時に“1”を書き込まないようにしてください。

表 3.9.4 ファンクションレジスタ機能表

モード	GPIOCFR1	GPIOCFR2
汎用	0	0
機能 1	1	0
機能 2	0	1
禁止	1	1

5. GPIOCIS (Port C Interrupt Selection Register (Level and Edge))

Address = (0xF080_2000) +(0x0804)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7]	PC7IS	R/W	0y0	ポート C 割り込み感知レジスタ 0y0: エッジ検出 0y1: レベル検出
[6:0]	–	–	Undefined	Read as undefined. Write as zero.

(個別説明)

a. <PC7IS>

割り込み感知レジスタ : エッジ検出とレベル検出を選択するレジスタです。

0y0: エッジ検出

0y1: レベル検出

6. GPIOCIBE (Port C Interrupt Selection Register (Fellow edge and Both edge))

Address = (0xF080_2000) + (0x0808)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7]	PC7IBE	R/W	0y0	ポート C 割り込み両エッジレジスタ 0y0: 片エッジ 0y1: 両エッジ
[6:0]	–	–	Undefined	Read as undefined. Write as zero.

(個別説明)

a. <PC7IBE>

割り込み両エッジレジスタ：片エッジか両エッジが選択するレジスタです。

0y0: 片エッジ

0y1: 両エッジ

7. GPIOCIEV (Port C Interrupt Select Register (“Fall down edge/Low level” and “Rising up edge/High level”))

Address = (0xF080_2000) + (0x080C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7]	PC7IEV	R/W	0y0	ポート C 割り込みイベントレジスタ 0y0: 立ち下がりエッジ/Low レベル 0y1: 立ち上がりエッジ/High レベル
[6:0]	–	–	Undefined	Read as undefined. Write as zero.

(個別説明)

a. <PC7IEV>

割り込みイベントレジスタ：エッジ検出の場合、立ち下がり or 立ち上がりを制御し、レベル検出の場合、Low レベル or High レベルを選択するレジスタです。

0y0: 立ち下がりエッジ/Low レベル

0y1: 立ち上がりエッジ/High レベル

8. GPIOCIE (Port C Interrupt Enable Register)

Address = (0xF080_2000) + (0x0810)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7]	PC7IE	R/W	0y0	ポート C 割り込み許可レジスタ 0y0: 禁止 0y1: 許可
[6:0]	Reserved	R/W	0y0000000	常に“0”をライトしてください。 リードすると“0”になります。

(個別説明)

a. <PC7IE >

割り込み許可レジスタ：割り込みの許可・禁止を制御するレジスタです。

0y0: 禁止

0y1: 許可

9. GPIOCRIS (Port C Interrupt Status Register (Raw))

Address = (0xF080_2000) + (0x0814)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined.
[7]	PC7RIS	RO	0y0	ポート C 割り込み許可前ステータスレジスタ 0y0: 要求なし 0y1: 要求あり
[6:0]	–	–	Undefined	Read as undefined.

(個別説明)

a. <PC7RIS >

割り込み許可前ステータスレジスタ：許可レジスタでマスクされる前の状態をモニタするレジスタです。

0y0: 要求なし

0y1: 要求あり

10. GPIOCMIS (Port C Interrupt Status Register (Masked))

Address = (0xF080_2000) +(0x0818)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined.
[7]	PC7MIS	RO	0y0	ポート C 割り込み許可後ステータスレジスタ 0y0: 要求なし 0y1: 要求あり
[6:0]	–	–	Undefined	Read as undefined.

(個別説明)

a. <PC7MIS >

割り込み許可後ステータスレジスタ：許可レジスタでマスクされた後の状態をモニタするレジスタです。

0y0: 要求なし

0y1: 要求あり

外部割り込みレジスタの設定表です。
各レジスタの同ビットの設定の組み合わせを下表に示します。

表 3.9.5 割り込みレジスタ関係設定例

(GPIOnIS, GPIOnIBE, GPIOnIEV, GPIOnIE, GPIOnRIS, GPIOnMIS: n=A, C, D)

レジスタ設定				検出方法	出力		
GPIOnIS (割り込み感知レジスタ)	GPIOnIBE (割り込み面エッジレジスタ)	GPIOnIEV (割り込みイベントレジスタ)	GPIOnIE (割り込み許可レジスタ)		GPIOnRIS (割り込み許可ステータスレジスタ)	GPIOnMIS (割り込み許可後ステータスレジスタ)	INTS[Num]
0	0	0	0	立ち下がりエッジ検出	検出可	検出不可 (0x00)	検出不可
		1		立ち上がりエッジ検出			
	1	0	0	両エッジ検出	検出可	検出不可 (0x00)	検出不可
		1					
	0	0	1	立ち下がりエッジ検出	検出可	検出可	検出可
		1		立ち上がりエッジ検出			
	1	0	1	両エッジ検出	検出可	検出可	検出可
		1					
1	0	0	0	Low レベル検出	検出可	検出不可 (0x00)	検出不可
		1		High レベル検出			
	1	0	0	Low レベル検出	検出可	検出不可 (0x00)	検出不可
		1		High レベル検出			
	0	0	1	Low レベル検出	検出可	検出可	検出可
		1		High レベル検出			
	1	0	1	Low レベル検出	検出可	検出可	検出可
		1		High レベル検出			

11. GPIOCIC (Port C Interrupt Clear Register)

Address = (0xF080_2000) + (0x081C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7]	PC7IC	WO	0y0	ポート C 割り込みクリアレジスタ 0y0: 無効 0y1: 要求クリア
[6:0]	–	–	Undefined	Read as undefined. Write as zero.

(個別説明)

a. <PC7IC >

割り込みクリアレジスタ：エッジ割り込みをクリアするレジスタです。

0y0: 無効

0y1: 要求クリア

12. GPIOCODE (Port C Open-drain Output Enable Register)

Address = (0xF080_2000) + (0x0C00)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:6]	PC7ODE~PC6ODE	R/W	0x00	ポート C OPD 出力許可レジスタ(各ビット) 0y0: プッシュプル出力 0y1: オープンドレイン(Pch ディセーブル) 出力
[5]	–	–	Undefined	Read as undefined. Write as zero.
[4:2]	Reserved	R/W	0y000	常に“0”をライトしてください。 リードすると“0”になります。
[1:0]	–	–	Undefined	Read as undefined. Write as zero.

(個別説明)

a. <PC7ODE, PC6ODE >

OPD 出力許可レジスタ：プッシュプル出力か OPD 出力かを選択するレジスタです。

0y0: プッシュプル出力

0y1: オープンドレイン(Pch ディセーブル) 出力

3.9.2.4 ポートD

ポート D は、汎用入力端子として使用することができます。

また、これ以外に、割り込み端子(INTB,INTA)、ADC 機能(AN7~AN4)、タッチスクリーン切り替え機能(PX,PY,MX,MY)に切り替えて使用できます。

汎用入力・割り込み設定

用途	データ値	機能切り替え 1	機能切り替え 2	割り込み許可
汎用入力 割り込み	GPIODDATA	GPIODFR1	GPIODFR2	GPIODIE
	*	0	0	0/1

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
入力/INTB	入力/INTA	入力	入力	-	-	-	-

注) 割り込みは、ビット 7、ビット 6 のみ対応

ADC 設定

用途	データ値	機能切り替え 1	機能切り替え 2	割り込み許可
ADC	GPIODDATA	GPIODFR1	GPIODFR2	GPIODIE
	*	1	0	0

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
AN7	AN6	AN5	AN4	-	-	-	-

TSI 設定

用途	データ値	機能切り替え 1	機能切り替え 2	割り込み許可
TSI	GPIODDATA	GPIODFR1	GPIODFR2	GPIODIE
	*	0	1	0

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PY	PX//INTA(INTTSI)	MY	MX	-	-	-	-

base アドレス = 0xF080_3000

Register Name	Address (base+)	Description
GPIODDATA	0x03FC	PortD Data Register
–	0x0400	Reserved
GPIODFR1	0x0424	PortD Function Register1
GPIODFR2	0x0428	PortD Function Register2
GPIODIS	0x0804	PortD Interrupt Selection Register (Level and Edge)
GPIODIBE	0x0808	PortD Interrupt Selection Register (Fellow edge and Both edge)
GPIODIEV	0x080C	PortD Interrupt Selection Register (Fall down edge/Low level and Rising up edge/High level)
GPIODIE	0x0810	PortD Interrupt Enable Register
GPIDRIS	0x0814	PortD Interrupt Status Register (Raw)
GPIDMIS	0x0818	PortD Interrupt Status Register (Masked)
GPIDIC	0x081C	PortD Interrupt Clear Register
–	0x0C00	Reserved

1. GPIODDATA (Port D Data Register)

Address = (0xF080_3000) + (0x03FC)

Bit	Bit Symbol	Type	Reset Value	Bit mask	Description
[31:4]	–	–	Undefined	–	Read as undefined.
[7:4]	PD[7:4]	RO	0xF	Bm7:4	ポート D データレジスタ
[3:0]	–	–	Undefined	–	Read as undefined.

(個別説明)

a. <PD[7:4]>

データレジスタ : データを保持するレジスタです。

ビットマスク機能については、データレジスタについての注意事項を参照してください。

2. GPIODFR1(Port D Function Register1)

Address = (0xF080_3000) + (0x0424)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:4]	PD7F1~PD4F1	R/W	0xF	ポート D ファンクションレジスタ 1
[3:0]	–	–	Undefined	Read as undefined. Write as zero.

(個別説明)

a. <PD7F1~PD4F1>

ファンクションレジスタ 1 : 機能切り替えの制御レジスタです。

3. GPIODFR2 (Port D Function Register2)

Address = (0xF080_3000) + (0x0428)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:4]	PD7F2~PD4F2	R/W	0y0000	ポート D ファンクションレジスタ 2
[3:0]	Reserved	R/W	0y0000	“0”をライトしてください。 リードすると“0”になります。

(個別説明)

a. <PD7F2~PD4F2>

ファンクションレジスタ 2：機能切り替えの制御レジスタです。

(注意) ファンクションレジスタ 1 とファンクションレジスタ 2 は、排他的にしか設定することは出来ません。瞬時でも、同時に“1”を書き込まないようにしてください。

表 3.9.6 ファンクションレジスタ機能表

モード	GPIODFR1	GPIODFR2
汎用	0	0
機能 1	1	0
機能 2	0	1
禁止	1	1

4. GPIODIS (Port D Interrupt Selection Register (Level and Edge))

Address = (0xF080_3000) + (0x0804)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:6]	PD7IS~PD6IS	R/W	0y00	ポート D 割り込み感知レジスタ(各ビット) 0y0: エッジ検出 0y1: レベル検出
[5:0]	Reserved	R/W	0y000000	“0”をライトしてください。 リードすると“0”になります。

(個別説明)

a. <PD7IS~PD6IS>

割り込み感知レジスタ：エッジ検出とレベル検出を選択するレジスタです。

0y0: エッジ検出

0y1: レベル検出

5. GPIODIBE (Port D Interrupt Select Register (Fellow edge and Both edge))

Address = (0xF080_3000) + (0x0808)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:6]	PD7IBE~PD6IBE	R/W	0y00	ポート D 割り込み両エッジレジスタ(各ビット) 0y0: 片エッジ 0y1: 両エッジ
[5:0]	Reserved	R/W	0y000000	“0”をライトしてください。 リードすると“0”になります。

(個別説明)

a. <PD7IBE~PD6IBE>

割り込み両エッジレジスタ：片エッジか両エッジか選択するレジスタです。

0y0: 片エッジ

0y1: 両エッジ

6. GPIODIEV (Port D Interrupt Select Register (“Fall down edge/Low level” and “Rising up edge/High level”))

Address = (0xF080_3000) + (0x080C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:6]	PD7IEV~PD6IEV	R/W	0y00	ポート D 割り込みイベントレジスタ(各ビット) 0y0: 立ち下がりエッジ/Low レベル 0y1: 立ち上がりエッジ/High レベル
[5:0]	Reserved	R/W	0y000000	“0”をライトしてください。 リードすると“0”になります。

(個別説明)

a. <PD7IEV~PD6IEV>

割り込みイベントレジスタ：エッジ検出の場合、立ち下がり or 立ち上りを制御し、レベル検出の場合、Low レベル or High レベルを選択するレジスタです。

0y0: 立ち下がりエッジ/Low レベル

0y1: 立ち上がりエッジ/High レベル

7. GPIODIE (Port D Interrupt Enable Register)

Address = (0xF080_3000) + (0x0810)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:6]	PD7IE~PD6IE	R/W	0y00	ポート D 割り込み許可レジスタ(各ビット) 0y0: 禁止 0y1: 許可
[5:0]	Reserved	R/W	0y000000	“0”をライトしてください。 リードすると“0”になります。

(個別説明)

a. <PD7IE~PD6IE>

割り込み許可レジスタ：割り込みの許可・禁止を制御するレジスタです。

0y0: 禁止

0y1: 許可

8. GPIODRIS (Port D Interrupt Status Register (Raw))

Address = (0xF080_3000) + (0x0814)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined.
[7:6]	PD7RIS~PD6RIS	RO	0y00	ポート D 割り込み許可前ステータスレジスタ (各ビット) 0y0: 要求なし 0y1: 要求あり
[5:0]	–	–	Undefined	Read as undefined.

(個別説明)

a. <PD7RIS~PD6RIS>

割り込み許可前ステータスレジスタ：許可レジスタでマスクされる前の状態をモニタするレジスタです。

0y0: 要求なし

0y1: 要求あり

9. GPIODMIS (Port D Interrupt Status Register (Masked))

Address = (0xF080_3000) + (0x0818)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined.
[7:6]	PD7MIS~PD6MIS	RO	0y00	ポート D 割り込み許可後ステータスレジスタ (各ビット) 0y0: 要求なし 0y1: 要求あり
[5:0]	–	–	Undefined	Read as undefined.

(個別説明)

a. <PD7MIS~PD6MIS>

割り込み許可後ステータスレジスタ：許可レジスタでマスクされた後の状態をモニタするレジスタです。

0y0: 要求なし

0y1: 要求あり

注) 各外部割り込みレジスタの設定表は表 3.9.5を参照してください。

10. GPIODIC (Port D Interrupt Clear Register)

Address = (0xF080_3000) + (0x081C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:6]	PD7IC~PD6IC	WO	0y00	ポート D 割り込みクリアレジスタ (各ビット) 0y0: 無効 0y1: 要求クリア
[5:0]	–	–	Undefined	Read as undefined. Write as zero.

(個別説明)

a. <PD7IC~PD6IC>

割り込みクリアレジスタ：エッジ割り込みをクリアするレジスタです。

0y0: 無効

0y1: 要求クリア

3.9.2.5 ポートN

ポート N は、汎用入出力端子として使用することが出来ます。

また、これ以外に、UART/IrDA 機能(U0RXD, U0TXD, SIR0IN, SIR0OUT) に切り替えて使用できます。

汎用入力設定

用途	データ値	入出力切り替え	機能切り替え 1	機能切り替え 2
汎用入力	GPIONDATA	GPIONDIR	GPIONFR1	GPIONFR2
	*	0	0	0

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
-	-	-	-	-	-	入力	入力

汎用出力設定

用途	データ値	入出力切り替え	機能切り替え 1	機能切り替え 2
汎用出力	GPIONDATA	GPIONDIR	GPIONFR1	GPIONFR2
	*	1	0	0

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
-	-	-	-	-	-	出力	出力

UART(ch0)設定

用途	データ値	入出力切り替え	機能切り替え 1	機能切り替え 2
UART(ch0)	GPIONDATA	GPIONDIR	GPIONFR1	GPIONFR2
	*	*	1	0

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
-	-	-	-	-	-	-	U0TXD

UART(ch0/IrDA)設定

用途	データ値	入出力切り替え	機能切り替え 1	機能切り替え 2
UART (ch0/IrDA)	GPIONDATA	GPIONDIR	GPIONFR1	GPIONFR2
	*	*	0	1

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
-	-	-	-	-	-	SIR0IN /U0RXD	SIR0OUT

base アドレス = 0xF080_C000

Register Name	Address (base+)	Description
GPIONDATA	0x03FC	PortN Data Register
GPIONDIR	0x0400	PortN Data Direction Register
GPIONFR1	0x0424	PortN Function Register1
GPIONFR2	0x0428	PortN Function Register2
Reserved	0x0804	
Reserved	0x0808	
Reserved	0x080C	
Reserved	0x0810	
Reserved	0x0814	
Reserved	0x0818	
Reserved	0x081C	
Reserved	0x0C00	

1. GPIONDATA (Port N Data Register)

Address = (0xF080_C000) + (0x03FC)

Bit	Bit Symbol	Type	Reset Value	Bit mask	Description
[31:2]	–	–	Undefined	–	Read as undefined. Write as zero.
[1:0]	PN[1:0]	R/W	0y11	Bm1:0	ポート N データレジスタ

(個別説明)

a. <PN[1:0]>

データレジスタ：データを保持するレジスタです。

ビットマスク機能については、データレジスタについての注意事項を参照してください。

2. GPIONDIR (Port N Data Direction Register)

Address = (0xF080_C000) + (0x0400)

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	–	–	Undefined	Read as undefined. Write as zero.
[1:0]	PN1C~PN0C	R/W	0y00	ポート N データ方向レジスタ(各ビット) 0y0: 入力 0y1: 出力

(個別説明)

a. <PN1C~PN0C>

データ方向レジスタ：汎用ポートとして使用するときの各ピンの入出力を制御するレジスタです。

0y0: 入力

0y1: 出力

3. GPIONFR1 (Port N Function Register1)

Address = (0xF080_C000) + (0x0424)

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	–	–	Undefined	Read as undefined. Write as zero.
[1]	Reserved	–	0y0	0をライトしてください。 リードすると0になります。
[0]	PN0F1	R/W	0y0	ポートNファンクションレジスタ1

(個別説明)

a. <PN0F1>

ファンクションレジスタ1：機能切り替えの制御レジスタです。

4. GPIONFR2 (Port N Function Register2)

Address = (0xF080_C000) + (0x0428)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:2]	Reserved	–	0y000000	"0"をライトしてください。 リードすると"0"になります。
[1:0]	PN1F2~PN0F2	R/W	0y00	ポートNファンクションレジスタ2

(個別説明)

a. <PN1F2~PN0F2>

ファンクションレジスタ2：機能切り替えの制御レジスタです。

(注意) ファンクションレジスタ1とファンクションレジスタ2は、排他的にしか設定することは出来ません。瞬時でも、同時に"1"を書き込まないようにしてください。

表 3.9.7 ファンクションレジスタ機能表

モード	GPIONFR1	GPIONFR2
汎用	0	0
機能1	1	0
機能2	0	1
禁止	1	1

3.9.2.6 ポートT

ポートTは、汎用入出力端子として使用することが出来ます。

また、これ以外に、USB の外部クロック入力(X1USB)、UART 機能(U1CTS_n, U1RXD, U1TXD)、および SPI 機能(SP0DI, SP0DO, SP0CLK, SP0FSS)、I2S 制御機能および USB0C_n, USBPON に切り替えて使用できます。

汎用入力設定

用途	データ値	入出力切り替え	機能切り替え 1	機能切り替え 2
汎用入力	GPIOTDATA	GPIOTDIR	GPIOTFR1	GPIOTFR2
	*	0	0	0

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
入力	入力	入力	入力	入力	入力	入力	入力

汎用出力設定

用途	データ値	入出力切り替え	機能切り替え 1	機能切り替え 2
汎用出力	GPIOTDATA	GPIOTDIR	GPIOTFR1	GPIOTFR2
	*	1	0	0

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
出力	出力	出力	出力	出力	出力	出力	出力

UART・SPI 設定

用途	データ値	入出力切り替え	機能切り替え 1	機能切り替え 2
UART SPI	GPIOTDATA	GPIOTDIR	GPIOTFR1	GPIOTFR2
	*	*	1	0

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
X1USB	U1CTS _n	U1RXD	U1TXD	SP0DI	SP0DO	SP0CLK	SP0FSS

USB Host 電源制御・I2S 設定

用途	データ値	入出力切り替え	機能切り替え 1	機能切り替え 2
USB Host I2S	GPIOTDATA	GPIOTDIR	GPIOTFR1	GPIOTFR2
	*	*	0	1

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	I2S1DATO	USBOC _n	USBPON	I2S0MCLK	I2S0DATI	I2S0CLK	I2S0WS

base アドレス = 0xF080_F000

Register Name	Address (base+)	Description
GPIOTDATA	0x03FC	PortT Data Register
GPIOTDIR	0x0400	PortT Data Direction Register
GPIOTFR1	0x0424	PortT Function Register1
GPIOTFR2	0x0428	PortT Function Register2
–	0x0804	Reserved
–	0x0808	Reserved
–	0x080C	Reserved
–	0x0810	Reserved
–	0x0814	Reserved
–	0x0818	Reserved
–	0x081C	Reserved
–	0x0C00	Reserved

1. GPIOTDATA (Port T Data Register)

Address = (0xF080_F000) + (0x03FC)

Bit	Bit Symbol	Type	Reset Value	Bit mask	Description
[31:8]	–	–	Undefined	–	Read as undefined. Write as zero.
[7:0]	PT7~PT0	R/W	0xFF	Bm7:0	ポートTデータレジスタ

(個別説明)

a. <PT7~PT0>

データレジスタ：データを保持するレジスタです。

ビットマスク機能については、データレジスタについての注意事項を参照してください。

2. GPIOTDIR (Port T Data Direction Register)

Address = (0xF080_F000) + (0x0400)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:0]	PT7C~PT0C	R/W	0x00	ポートTデータ方向レジスタ(各ビット) 0y0: 入力 0y1: 出力

(個別説明)

a. <PT7C~PT0C>

データ方向レジスタ：汎用ポートとして使用するときの各ピンの入出力を制御するレジスタです。

0y0: 入力

0y1: 出力

3. GPIOTFR1 (Port T Function Register1)

Address = (0xF080_F000) + (0x0424)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:0]	PT7F1~PT0F1	R/W	0x00	ポート T ファンクションレジスタ 1

(個別説明)

a. <PT7F1~PT0F1>

ファンクションレジスタ 1: 機能切り替えの制御レジスタです。

4. GPIOTFR2 (Port T Function Register2)

Address = (0xF080_F000) + (0x0428)

Bit	Bit Symbol	Type	Reset Value	Description
[31:7]	–	–	Undefined	Read as undefined. Write as zero.
[6:0]	PT6F2~PT0F2	R/W	0y0000000	ポート T ファンクションレジスタ 2

(個別説明)

a. <PT6F2~PT0F2>

ファンクションレジスタ 2: 機能切り替えの制御レジスタです。

(注意) ファンクションレジスタ 1 とファンクションレジスタ 2 は、排他的にしか設定することは出来ません。瞬時でも、同時に“1”を書き込まないようにしてください。

表 3.9.8 ファンクションレジスタ機能表

モード	GPIOTFR1	GPIOTFR2
汎用	0	0
機能 1	1	0
機能 2	0	1
禁止	1	1

3.9.2.7 ポートU

ポート U は、汎用入出力端子として使用することが出来ます。

また、これ以外に、NAND コントローラ機能(NDD7~NDD0),LCD コントローラ機能(LD7~LD0)に切り替えて使用できます。

汎用入力設定

用途	データ値	入出力切り替え	機能切り替え 1	機能切り替え 2
汎用入力	GPIODATA	GPIODIR	GPIOUFR1	GPIOUFR2
	*	0	0	0

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
入力	入力	入力	入力	入力	入力	入力	入力

汎用出力設定

用途	データ値	入出力切り替え	機能切り替え 1	機能切り替え 2
汎用出力	GPIODATA	GPIODIR	GPIOUFR1	GPIOUFR2
	*	1	0	0

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
出力	出力	出力	出力	出力	出力	出力	出力

NAND 機能設定

用途	データ値	入出力切り替え	機能切り替え 1	機能切り替え 2
NAND	GPIODATA	GPIODIR	GPIOUFR1	GPIOUFR2
	*	*	1	0

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
NDD7	NDD6	NDD5	NDD4	NDD3	NDD2	NDD1	NDD0

LCDC 機能設定

用途	データ値	入出力切り替え	機能切り替え 1	機能切り替え 2
LCDC	GPIODATA	GPIODIR	GPIOUFR1	GPIOUFR2
	*	*	0	1

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LD7	LD6	LD5	LD4	LD3	LD2	LD1	LD0

base アドレス = 0xF080_4000

Register Name	Address (base+)	Description
GPIODATA	0x03FC	PortU Data Register
GPIODIR	0x0400	PortU Data Direction Register
GPIOUFR1	0x0424	PortU Function Register1
GPIOUFR2	0x0428	PortU Function Register2
–	0x0804	Reserved
–	0x0808	Reserved
–	0x080C	Reserved
–	0x0810	Reserved
–	0x0814	Reserved
–	0x0818	Reserved
–	0x081C	Reserved
–	0x0C00	Reserved

1. GPIODATA (Port U Data Register)

Address = (0xF080_4000) + (0x03FC)

Bit	Bit Symbol	Type	Reset Value	Bit mask	Description
[31:8]	–	–	Undefined	–	Read as undefined. Write as zero.
[7:0]	PU[7:0]	R/W	0xFF	Bm7:0	ポート U データレジスタ

(個別説明)

a. <PU[7:0]>

データレジスタ : データを保持するレジスタです。

ビットマスク機能については、データレジスタについての注意事項を参照してください。

2. GPIODIR (Port U Data Direction Register)

Address = (0xF080_4000) + (0x0400)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:0]	PU7C~PU0C	R/W	0x00	ポート U データ方向レジスタ(各ビット) 0y0: 入力 0y1: 出力

(個別説明)

a. <PU7C~PU0C>

データ方向レジスタ : 汎用ポートとして使用するときの各ピンの入出力を制御するレジスタです。

0y0: 入力

0y1: 出力

3. GPIOUFR1 (Port U Function Register1)

Address = (0xF080_4000) + (0x0424)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:0]	PU7F1~PU0F1	R/W	0x00	ポート U ファンクションレジスタ 1

(個別説明)

a. <PU7F1~PU0F1>

ファンクションレジスタ 1 : 機能切り替えの制御レジスタです。

4. GPIOUFR2 (Port U Function Register2)

Address = (0xF080_4000) + (0x0428)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:0]	PU7F2~PU0F2	R/W	0x00	ポート U ファンクションレジスタ 2

(個別説明)

a. <PU7F2~PU0F2>

ファンクションレジスタ 2 : 機能切り替えの制御レジスタです。

(注意) ファンクションレジスタ 1 とファンクションレジスタ 2 は、排他的にしか設定することは出来ません。瞬時でも、同時に“1”を書き込まないようにしてください。

表 3.9.9 ファンクションレジスタ機能表

モード	GPIOUFR1	GPIOUFR2
汎用	0	0
機能 1	1	0
機能 2	0	1
禁止	1	1

3.9.2.8 ポートV

ポート V は、汎用入出力端子として使用することが出来ます。

また、これ以外に、NAND コントローラ機能(NDRBn,NDCE1n,NDCE0n, NDCLE,NDALE, NDWEEn,NDREn),LCD コントローラ機能(LD15~LD8)に切り替えて使用できます。

汎用入力設定

用途	データ値	入出力切り替え	機能切り替え 1	機能切り替え 2
汎用入力	GPIOVDATA	GPIOVDIR	GPIOVFR1	GPIOVFR2
	*	0	0	0

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
入力	入力	入力	入力	入力	入力	入力	入力

汎用出力設定

用途	データ値	入出力切り替え	機能切り替え 1	機能切り替え 2
汎用出力	GPIOVDATA	GPIOVDIR	GPIOVFR1	GPIOVFR2
	*	1	0	0

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
出力	出力	出力	出力	出力	出力	出力	出力

NAND 機能設定

用途	データ値	入出力切り替え	機能切り替え 1	機能切り替え 2
NAND	GPIOVDATA	GPIOVDIR	GPIOVFR1	GPIOVFR2
	*	*	1	0

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
—	NDRB	NDCE1n	NDCE0n	NDCLE	NDALE	NDWEEn	NDREn

LCDC 機能設定

用途	データ値	入出力切り替え	機能切り替え 1	機能切り替え 2
LCDC	GPIOVDATA	GPIOVDIR	GPIOVFR1	GPIOVFR2
	*	*	0	1

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LD15	LD14	LD13	LD12	LD11	LD10	LD9	LD8

base アドレス = 0xF080_7000

Register Name	Address (base+)	Description
GPIOV DATA	0x03FC	PortV Data Register
GPIOV DIR	0x0400	PortV Data Direction Register
GPIOV FR1	0x0424	PortV Function Register1
GPIOV FR2	0x0428	PortV Function Register2
–	0x0804	Reserved
–	0x0808	Reserved
–	0x080C	Reserved
–	0x0810	Reserved
–	0x0814	Reserved
–	0x0818	Reserved
–	0x081C	Reserved
–	0x0C00	Reserved

1. GPIOV DATA (Port V Data Register)

Address = (0xF080_7000) + (0x03FC)

Bit	Bit Symbol	Type	Reset Value	Bit mask	Description
[31:8]	–	–	Undefined	–	Read as undefined. Write as zero.
[7:0]	PV[7:0]	R/W	0xFF	Bm7:0	ポート V データレジスタ

(個別説明)

a. <PV[7:0]>

データレジスタ：データを保持するレジスタです。

ビットマスク機能については、データレジスタについての注意事項を参照してください。

2. GPIOV DIR (Port V Data Direction Register)

Address = (0xF080_7000) + (0x0400)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:0]	PV7C~PV0C	R/W	0x00	ポート V データ方向レジスタ(各ビット) 0y0: 入力 0y1: 出力

(個別説明)

a. <PV7C~PV0C>

データ方向レジスタ：汎用ポートとして使用するときの各ピンの入出力を制御するレジスタです。

0y0: 入力

0y1: 出力

3. GPIOVFR1 (Port V Function Register1)

Address = (0xF080_7000) + (0x0424)

Bit	Bit Symbol	Type	Reset Value	Description
[31:7]	–	–	Undefined	Read as undefined. Write as zero.
[6:0]	PV6F1~PV0F1	R/W	0y0000000	ポート V ファンクションレジスタ 1

(個別説明)

b. <PV6F1~PV0F1>

ファンクションレジスタ 1 : 機能切り替えの制御レジスタです。

4. GPIOVFR2 (Port V Function Register2)

Address = (0xF080_7000) + (0x0428)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:0]	PV7F2~PV0F2	R/W	0x00	ポート V ファンクションレジスタ 2

(個別説明)

c. <PV7F2~PV0F2>

ファンクションレジスタ 2 : 機能切り替えの制御レジスタです。

(注意) ファンクションレジスタ 1 とファンクションレジスタ 2 は、排他的にしか設定することは出来ません。瞬時でも、同時に“1”を書き込まないようにしてください。

表 3.9.10 ファンクションレジスタ機能表

モード	GPIOVFR1	GPIOVFR2
汎用	0	0
機能 1	1	0
機能 2	0	1
禁止	1	1

3.9.3 注意事項

- 割り込み機能使用手順

割り込みは、エッジやレベルの様々なモードで検出が可能です。

割り込み検出機能を使用する場合(GPIONIE の該当 bit を 1 にする)やモードを切り替える場合(GPIONDIR,GPIONIS,GPIONIBE,GPONIEV の該当 bit を変更する)は、下記の手順に従ってください。

- | | |
|----------------------|----------------------------|
| 1. 該当 bit の割り込み方向を設定 | GPIONDIR を設定 |
| 2. 該当 bit を割り込み禁止 | GPONIE=0 |
| 3. 該当 bit のモード切り替え | GPONIS,GPONIBE,GPONIEV を設定 |
| 4. 該当 bit の割り込みクリア | GPONIC=1 |
| 5. 該当 bit の割り込み許可 | GPONIE=1 |

3.10 MPMC

本 LSI には、異なる仕様の 2 種類のメモリコントローラが内蔵されています。

接続する外部メモリに応じて、以下の選択を行います。

外部端子 SELMEMC(ポート SN0)の設定で、2 種類のメモリコントローラ(MPMC0/MPMC1)を選択することができます。

また、外部端子 SELDVCCM(ポート SN1)の設定と内部レジスタ(PMCDRV)の設定を変更することで、使用するメモリのインターフェース電源 DVCCM の電圧(1.8V 系/3.3V 系)に対応しています。

更に SDRAM を使用する場合には、特殊な端子接続や、レジスタ設定が必要です。以下に一覧表を示します。

表 3.10.1 メモリコントローラと対応電圧設定一覧

外部メモリ電源電圧			1.8V ± 0.1V	3.3V ± 0.3V
メモリコントローラ及び設定				
MPMC0	端子設定	SELMEMC (注 1)	0 入力	
		SELDVCCM (注 1)	0 入力	1 入力
		DMCCLKIN	0 入力	
	レジスタ設定	PMCDRV<DRV_MEM1:0>	0y11	0y01
SDRAM を使用する場合		16bit_bus dmc_user_config_3	0x00000000	
MPMC1	端子設定	SELMEMC (注 1)	1 入力	N/A
		SELDVCCM (注 1)	0 入力	
		DMCCLKIN	DMCDCLKP 信号と外部で接続 (注 2)	
	レジスタ設定	PMCDRV<DRV_MEM1:0>	0y11	
SDRAM を使用する場合		16bit_bus dmc_user_config_5	0x00000058	

注 1) SELMEMC, SELDVCCM 端子の電源は DVCC3IO です。よって、0 入力は 0V、1 入力は 3.3V を入力してください。

注 2) MPMC1 を使用して、DDR タイプの SDRAM を使用する場合は、内部のデータラッチ用のフィードバッククロックを入力する必要があります。基板設計時には、必ず DMCCLKP 信号を、出来るだけ最短の距離で DMCCLKIN 入力端子に接続してください。MPMC0 を使用して、SDR タイプの SDRAM を使用する場合や、SDRAM を使用しない場合は、GND に接地するなど、貫通電流の対策をしてください。

以下に、MPMC0 と MPMC1 の、サポートされるメモリの違いを示します。
使用する SDRAM の種類に応じて、MPMC0 か MPMC1 を選択してください。

- MPMC0 : 16-bit Standard type SDR SDRAM
 16-bit Mobile type SDR SDRAM
 16-bit NORF(非同期、セパレートバスのみ)
 16-bit SRAM(非同期、セパレートバスのみ)
- MPMC1 : 16-bit LVCMOS type DDR SDRAM
 16-bit NORF(非同期、セパレートバスのみ)
 16-bit SRAM(非同期、セパレートバスのみ)

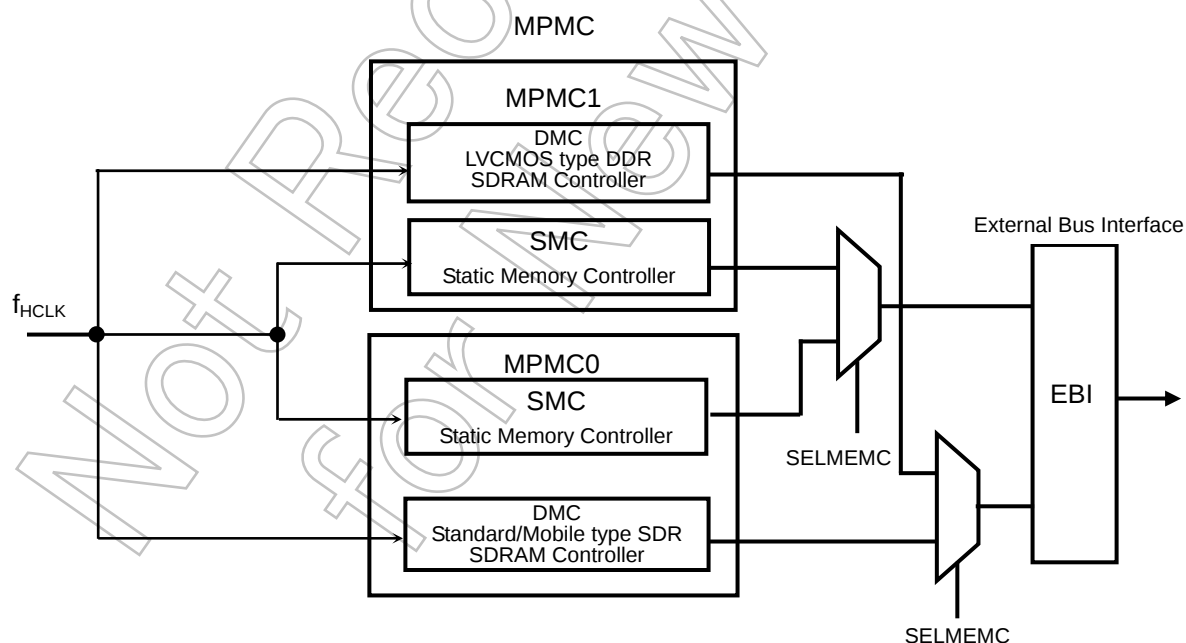
モード設定端子	動作モード
SELMEMC	
0	MPMC0 を使用
1	MPMC1 を使用

注 1) SDR タイプの SDRAM と DDR タイプの SDRAM の同時使用は出来ません。

注 2) 2 つのメモリコントローラは、ダイナミックに切り替えての使用は出来ません。どちらかに固定してください。

詳細は、各々の回路説明の章を参照してください。

以下に回路ブロックのイメージ図を載せます。



以下に、対応電源の違いによる切換え方法を示します。

使用するメモリの電圧に応じて、設定してください。

注)2つのメモリコントローラは、ダイナミックに切り替えての使用は出来ません。どちらかに固定してください。

モード設定端子	動作モード
SELDVCCM	
0	(NANDF を除く)メモリ系の制御端子は $1.8 \pm 0.1V$ (DVCCM)動作
1	メモリ系の制御端子は $3.3 \pm 0.3V$ (DVCCM)動作

また、使用する電源電圧に応じて、外部端子の電流駆動能力を変更する必要があります。

さらに SDRAM を使用する場合には、外部端子の接続及び、常数設定レジスタを変更する必要があります。以下の設定にしてください。

駆動能力設定レジスタ	動作モード
PMCDRV<DRV_MEM1:0>	
0y11	(NANDF を除く)メモリ系の制御端子は $1.8 \pm 0.1V$ (DVCCM)動作
0y01	メモリ系の制御端子は $3.3 \pm 0.3V$ (DVCCM)動作

注) PMCDRV レジスタはリセット解除後の低速動作(PLL が OFF 時)中に設定してください。

【SDR タイプの SDRAM を使用する場合】

バス幅設定レジスタ	動作モード
dmc_user_config_3	
0x00000000	SDR タイプの SDRAM(MPMC0 使用が前提)で 16bit バス

注) dmc_user_config_3 レジスタはリセット解除後、SDRAM 初期設定前に設定してください。PMC による HOT_RESET 解除後も同様に設定してください。

端子処理	動作モード
DMCCLKIN	
本端子は使用しません。 入力端子ですので、GND に接続する などの端子処理をしてください	SDR タイプの SDRAM(MPMC0 使用が前提)で 16bit バス

【DDR タイプの SDRAM を使用する場合】

バス幅設定レジスタ	動作モード
dmc_user_config_5	
0x00000058	DDR タイプの SDRAM(MPMC1 使用が前提)で 16bit バス

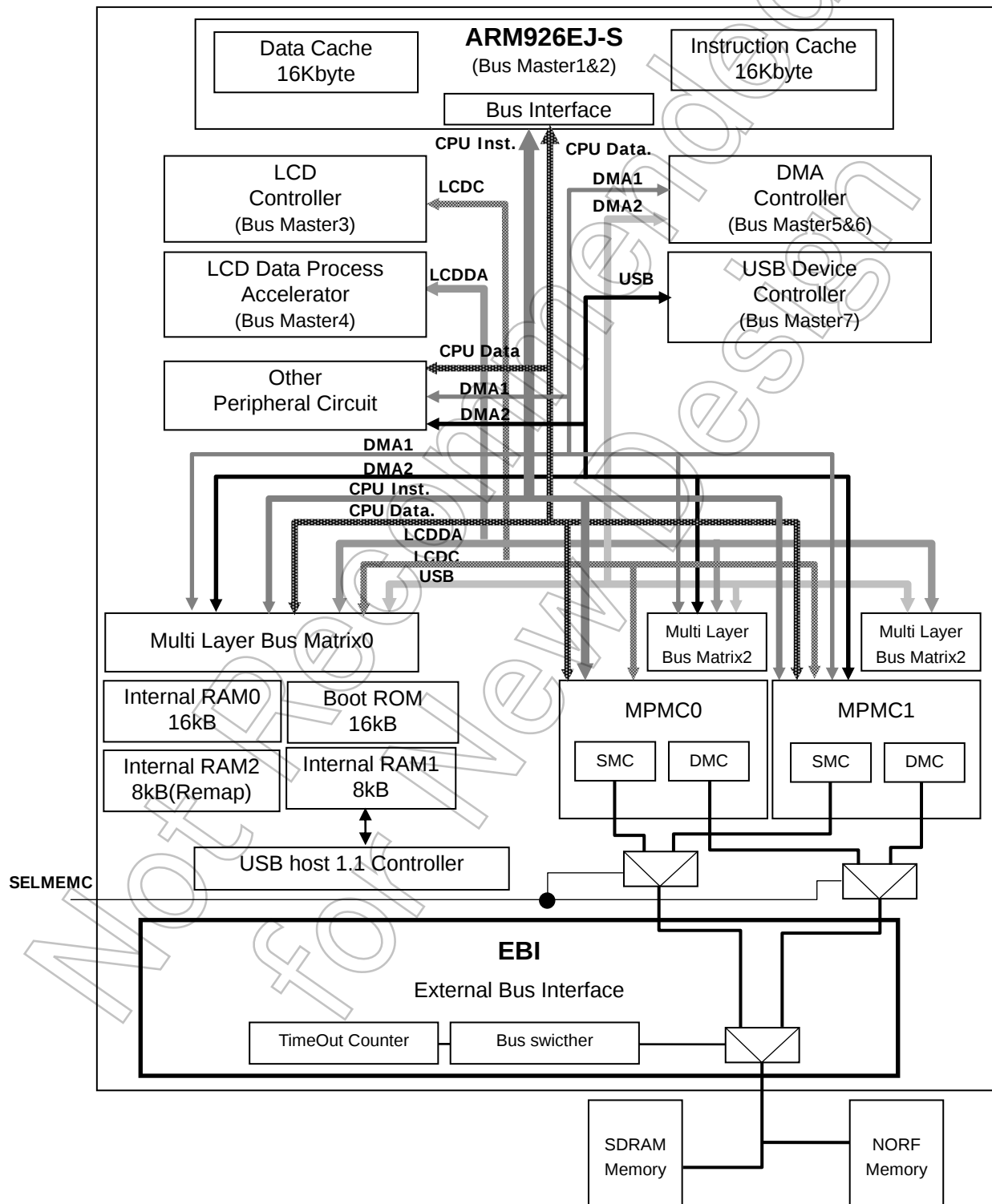
注) dmc_user_config_5 レジスタはリセット解除後、SDRAM 初期設定前に設定してください。PMC による HOT_RESET 解除後も同様に設定してください。

端子処理	動作モード
DMCCLKIN	
本端子と DMCCLKP 端子を接続してください	DDR タイプの SDRAM(MPMC1 使用が前提)で 16bit バス

3.10.1 EBI (External Bus Interface) について

各メモリコントローラ(MPMC0 および MPMC1)には、各々 SMC (Static Memory Controller)回路、および DMC (Dynamic Memory Controller)回路が内蔵されています。

本マイコンでは、SMC と DMC の外部バスを兼用しておりますが、メモリコントローラでは SMC と DMC は独立して動作するため、DMC と SMC の切換えは EBI (Exteranal Bus Interface)が制御しています。



EBI は、メモリコントローラ (DMC および SMC)からのアクセス要求に合わせて、バスの切換えを行っています。DMC および SMC のアクセス要求が重なった場合、一方のコントローラのアクセス実行中、もう一方のコントローラの要求を待たせます。

一方のメモリの連続要求が発生した場合に、もう一方のアクセスが長時間待たされることを避けるため、アクセス要求が重なった時間を管理し、強制的にバスをリリースする「タイムアウトカウンタ」が設定されています。

本マイコンでは、高速アクセスで要求頻度が高い DMC の要求を優先しています。

そのため、SMC 側のタイムアウトサイクルをレジスタで設定し、DMC の要求を優先的に処理する機能を準備しています。

表 3.10.2 EBI の Timeout

DMC タイムアウトサイクル	SMC タイムアウトサイクル
1024 クロック (固定)	~1024 クロック (レジスタ設定可能)

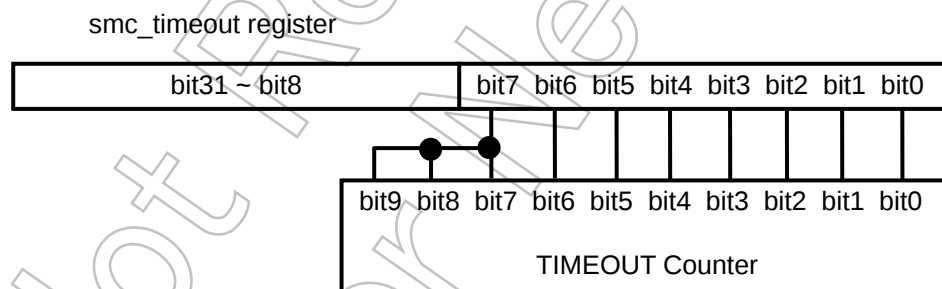
SMC タイムアウトサイクル設定レジスタ

base アドレス= 0xF00A_0000

Register Name	Address (base+)	Type	Reset Value	Description
smc_timeout	0x0050	R/W	0x000000FF	SMC TimeOut Register

注) 0x00000000 の設定は出来ません。0x00000001~0x000000FF のみ有効です。

smc_timeout サイクルは、10bit のカウンタで管理されていますが、制御レジスタの有効 bit は、下位 8bit のみとなります。有効 bit の最上位 (bit7)bit が、10bit のカウンタの上位 3bit の値を同時制御しています。



注) DMC メモリ (SDRAM) を LCDC の VRAM に設定する場合は、LCDC の UnderFlow を防ぐ為に、本レジスタの値を 0y01 に設定することをお勧めいたします。DMC 回路内に説明している QOS 機能と併用して使用してください。

3.10.2 MPMC0 の概要

MPMC0 には、SDRAM を制御する、DMC(Dynamic Memory Controller)と、NOR Flash、SRAM を制御する SMC(Static Memory Controller)が準備されています。

DMC(Dynamic Memory Controller)の特長

- (a) 16-bit SDR SDRAM に対応
- (b) 1 チャンネルのチップセレクトをサポート
- (c) SDRAM の各タイミングに対応するために、クロック単位での調整機能をサポート

SMC(Static Memory Controller)の特長

- (a) 非同期、16-bit の SRAM , NOR Flash(セパレートバスのみ、マルチプレクスバスはサポートしません) メモリをサポート
- (b) 2 チャンネルチップセレクト信号をサポート
- (c) チップセレクト毎に、サイクルの調整およびメモリデータバス幅をプログラムで設定可能

3.10.3 機能概要(MPMC0)

図 3.10.1は、MPMC0 回路の簡単なブロック図を示しています。

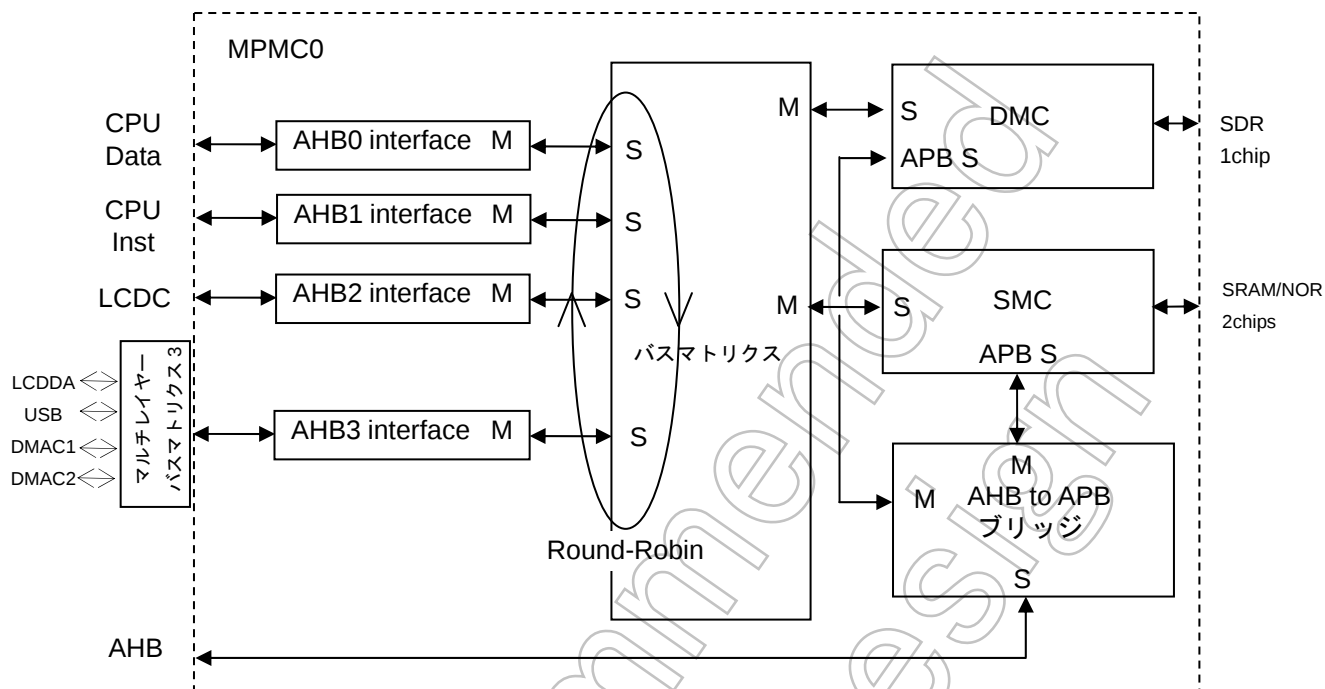
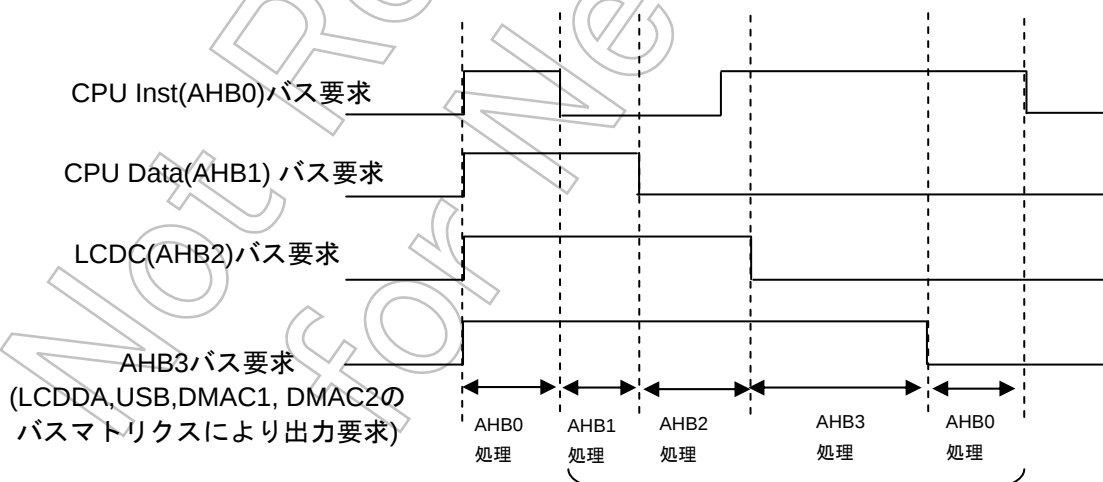


図 3.10.1 MPMC0 ブロック図

(a) バスマトリクス

1. AHB0, AHB1, AHB2, AHB3 のバスマトリクスは、Round-Robin(ラウンドロビン: 巡回的並列処理)アービタ方式をサポートします。下記の図にバス要求優先順位例を示します。



点線のポイントは処理完了し、バスを開放していることを示しています

処理優先順位は①→②→③→④→⑤

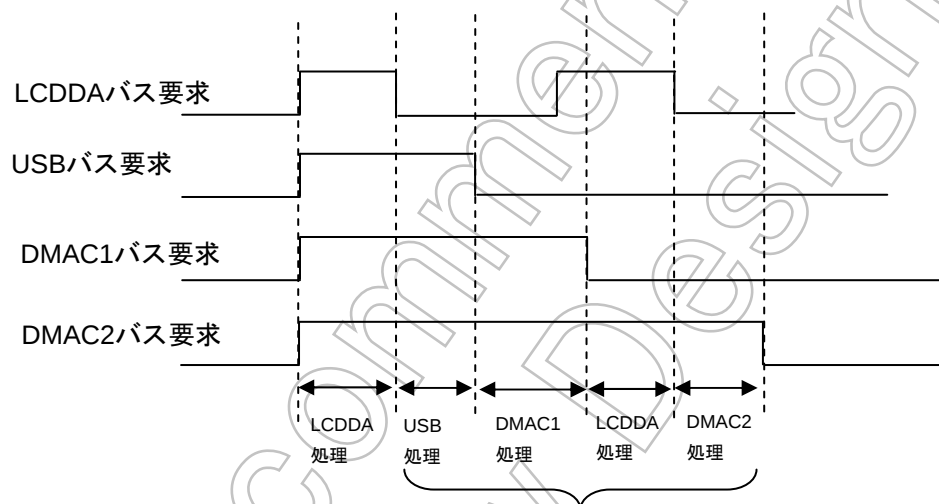
2. LCDDA, DMAC1, DMAC2, USB のバスマトリクス 2 の優先順位については、バス要求が早い方を優先的に処理します。同時のタイミングで複数のバスマスタ要求を受け付けた場合には、以下の順序で優先順位が固定されています。

同時要求時の優先順位は

LCDDA > USB > DMAC1 > DMAC2

となります。

下記の図にバス要求優先順位例を示します。



点線のポイントは処理完了し、バスを開放していることを示しています

処理優先順位は①→②→③→⑤→④

(b) クロック種類

クロックギア回路にてクロックを制御しています。

1. Dynamic メモリクロック : HCLK クロックを接続
2. Static メモリクロック : HCLK クロックまたは 1/2_HCLK クロックを接続(CLKCR5<SEL_SMC_MCLK>で選択)

3.10.3.1 DMC (Dynamic Memory Controller)

(1) DMC 機能概要

表 3.10.3 に、DMCの特長を示します。

表 3.10.3 DMC の特長

	特長
サポートメモリ	SDR SDRAM セパレートバスのみサポート
データバス幅	16bit データバス幅をサポート
アクセス空間	最大 512MB 空間のアクセス空間をサポート チップセレクト: DMCCSn のみをサポート
タイミング調整	レジスタによる AC タイミングの調整が可能
コマンド	Mode Register 設定、Auto refresh, Self Refresh, アクティブ、プリチャージ、リード/ライト コマンド、Powerdown などコマンドをサポート
クロック	DMCCLK の周波数 = f_{HCLK} 入力クロック端子 DMCCCLKIN を使用しないため、GND に固定
外部制御端子	D15~D0, A23~A0, DMCSDQM1, DMCSDQM0, DMCCSn, DMCWEn, DMCRASn, DMCCASn, DMCBA0, DMCBA1, DMCCKE, DMCCLK, DMCDCLKN, DMCAP

(2) DMC ブロック図

以下に、DMC 回路のブロック図を示します。

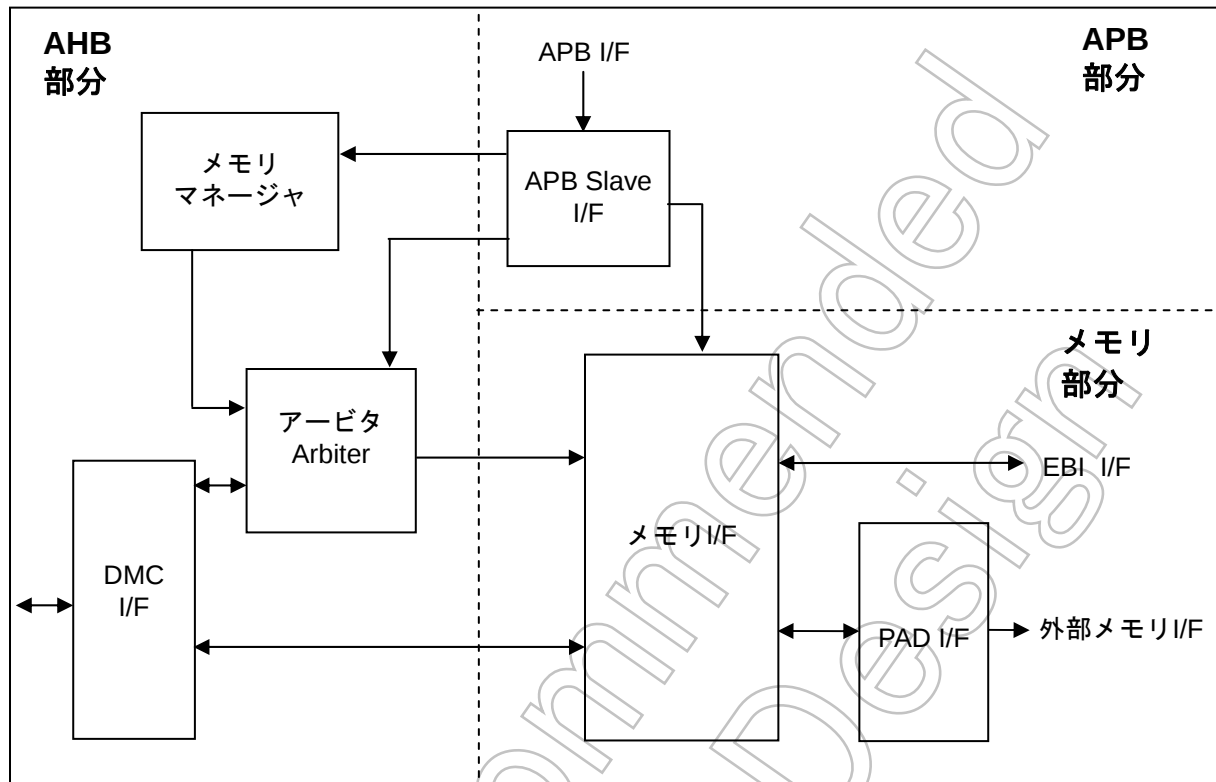


図 3.10.2 DMC ブロック図

(a) アービタ

アービタは DMC I/F とメモリ マネージャからアクセスコマンドを受けて、アクセスの調停後、優先度が一番高いコマンドをメモリ I/F に渡します。データはメモリ I/F から DMC I/F に直接接続されています。

(b) メモリ マネージャ

メモリ マネージャは、DMC 回路の状態を監視し、制御を行います。

(3) DMC Function 動作

(a) アービタ動作

1. リード/ライトアクセスの調停
2. リードアクセスでは、QoS 動作(Quality of service)をサポート
3. ハザード処理

同一のバスマスタ単体が、外部メモリをアクセスする場合、実際のメモリアクセス順序は命令の順序通りに実行されます。

ただし、複数のバスマスタが、外部メモリアクセスする場合、READ と WRITE は各々別々の Buffer に一旦蓄積され、優先順位回路によって実行される為、READ と WRITE の実行順序が入れ替わることがあります。

十分な時間を取るか、先に実行したアクセスが完了したことを確認するか、または、共用するメモリは内蔵 RAM を利用するなどの管理をしてください。

4. ステートマシンを監視し、適正なパイプラインのエントリを選択

(b) メモリマネージャ動作

1. DMC 回路の監視と制御
2. ダイレクトコマンドの発行：
 - NOP
 - Prechargeall
 - Autorefresh
 - Modereg
 - 拡張 modereg
3. Auto Refresh 機能のサポート

15 ビットのカウンタでオートリフレッシュのサイクルを設定できます。

(c) メモリ I/F 動作

用途に応じた下記 3 種類の FIFO を内蔵しています。

1. コマンド FIFO : 2word
2. データリード FIFO : 10word
3. データライト FIFO : 10word

※リードおよびライトの FIFO サイズが 10word のため、最大 8word の連続転送しかサポートしていません。(1word=32bit data)

(d) Low-power 機能

DMC は下記 2 種類の Low-power をサポートしています。

1. dmc_memc_cmd_3 レジスタの設定により、Low-power(Self-refresh Mode)を実現できます。
2. dmc_memory_cfg_3 レジスタの設定により、メモリクロックを停止する(Clock Suspend Mode)機能とメモリアクセスがないときに CKE 端子を自動的に無効化する(Power Down Mode)機能をサポートしています。

注) Clock Suspend Mode 機能と Power Down Mode 機能は併用できません。

(e) QoS 機能

QoS はリードアクセスでしか使用できません。

QoS 機能とは、MPMC 内のバスマトリクスが通常管理する順序回路(RoundRobin)において、例外処理を行う為のサービス機能です。

dmc_id_x_cfg_3<qos_max>には要求されているリードの最大レイテンシーを設定します。QoS_max の Timeout が発生する時、DMC に内蔵するアービタは、該当する転送マスターのプライオリティを上げます。

また、dmc_id_x_cfg_3<qos_min>を enable に設定すると、アクセスを最優先させることが出来ます。しかし、この場合 DMC のスケジューリングアルゴリズムが制限されて、全体のメモリバンド幅が影響されることがあります。

dmc_id_x_cfg_3<qos_enable>を Enable に設定すると、qos_max に設定した内部カウンタは、ゼロになるまでデクリメントを開始します。もし、内部カウンタの値がゼロになっても(Time-out 状態)、待ち状態が続く場合は、該当するバスマスタ要求のプライオリティを、RoundRobin でスケジューリングされた要求順位から、最上位に上げます。

qos_min を enable に設定すると、qos_max に設定された値は無視され、常に最優先での処理を行います。

表 3.10.4 SDR メモリのセットアップ例

base アドレス= 0xF430_0000

Register address	Write data	Description
0x0014	0x00000006	Set cas_Latency to 3
0x0018	0x00000000	Set t_dqss to 0
0x001C	0x00000002	Set t_mrd to 2
0x0020	0x00000007	Set t_ras to 7
0x0024	0x0000000B	Set t_rc to 11
0x0028	0x00000015	Set t_rcd to 5 and schedule_rcd to 2
0x002C	0x000001F2	Set t_rfc to 18 and schedule_rfc to 15
0x0030	0x00000015	Set t_rp to 5 and schedule_rp to 2
0x0034	0x00000002	Set t_rrd to 2
0x0038	0x00000003	Set t_wr to 3
0x003C	0x00000002	Set t_wtr to 2
0x0040	0x00000001	Set t_xp to 1
0x0044	0x0000000A	Set t_xsr to 10
0x0048	0x00000014	Set t_esr to 20
0x000C	0x00010020	Set memory configuration
0x0010	0x00000A60	Set auto refresh period to be every 2656 DMCSCLK periods
0x0200	0x000000FF	Set chip select for chip 0 to be 0x00XXXXXX, rbc configuration
0x0008	0x000C0000	Carry out chip 0 Nop command
0x0008	0x00000000	Carry out chip 0 Prechargeall command
0x0008	0x00040000	Carry out chip 0 Autorefresh command
0x0008	0x00040000	Carry out chip 0 Autorefresh command
0x0008	0x00080032	Carry out chip 0 Mode Reg command 0x32 mapped to low add bits
0x0004	0x00000000	Change DMC state to Ready

(4) MPMC0 の DMC レジスタの説明

表 3.10.5 MPMC0 の DMC SFR 一覧

base アドレス= 0xF430_0000

Register Name	Address (base+)	Type	Reset Value	Description
dmc_memc_status_3	0x0000	RO	0x00000380	DMC Memory Controller Status Register
dmc_memc_cmd_3	0x0004	WO	–	DMC Memory Controller Command Register
dmc_direct_cmd_3	0x0008	WO	–	DMC Direct Command Register
dmc_memory_cfg_3	0x000C	R/W	0x00010020	DMC Memory Configuration Register
dmc_refresh_prd_3	0x0010	R/W	0x00000A60	DMC Refresh Period Register
dmc_cas_latency_3	0x0014	R/W	0x00000006	DMC CAS Latency Register
dmc_t_dqss_3	0x0018	R/W	0x00000001	DMC t_dqss Register
dmc_t_mrd_3	0x001C	R/W	0x00000002	DMC t_mrd Register
dmc_t_ras_3	0x0020	R/W	0x00000007	DMC t_ras Register
dmc_t_rc_3	0x0024	R/W	0x0000000B	DMC t_rc Register
dmc_t_rcd_3	0x0028	R/W	0x0000001D	DMC t_rcd Register
dmc_t_rfc_3	0x002C	R/W	0x00000212	DMC t_rfc Register
dmc_t_rp_3	0x0030	R/W	0x0000001D	DMC t_rp Register
dmc_t_rrd_3	0x0034	R/W	0x00000002	DMC t_rrd Register
dmc_t_wr_3	0x0038	R/W	0x00000003	DMC t_wr Register
dmc_t_wtr_3	0x003C	R/W	0x00000002	DMC t_wtr Register
dmc_t_xp_3	0x0040	R/W	0x00000001	DMC t_xp Register
dmc_t_xsr_3	0x0044	R/W	0x0000000A	DMC t_xsr Register
dmc_t_esr_3	0x0048	R/W	0x00000014	DMC t_esr Register
dmc_id_0_cfg_3 dmc_id_1_cfg_3 dmc_id_2_cfg_3 dmc_id_3_cfg_3	0x0100 0x0104 0x0108 0x010C	R/W	0x00000000	DMC id_<0-3>_cfg Registers
dmc_chip_0_cfg_3	0x0200	R/W	0x0000FF00	DMC chip_0_cfg Registers
Reserved	0x0204	–	Undefined	Read as undefined. Write as zero.
Reserved	0x0208	–	Undefined	Read as undefined. Write as zero.
Reserved	0x020C	–	Undefined	Read as undefined. Write as zero.
Reserved	0x0300	–	Undefined	Read as undefined. Write as zero.
dmc_user_config_3	0x0304	WO	Undefined	DMC user_config Register
Reserved	0x0E00	–	Undefined	Read as undefined. Write as zero.
Reserved	0x0E04	–	Undefined	Read as undefined. Write as zero.
Reserved	0x0E08	–	Undefined	Read as undefined. Write as zero.
Reserved	0x0FE0-0x0FEC	–	Undefined	Read as undefined. Write as zero.
Reserved	0x0FF0-0x0FFC	–	Undefined	Read as undefined. Write as zero.

注) APB は single-word 32-bit アクセスのみをサポート、レジスタのリード/ライトは single-word 32-bit で実行してください。

MPMC0

レジスタ Read/Write できる dmc_memc_status_3 状態一覧 ○ : 許可 × : 禁止

Register Name	Type	Read				Write			
		dmc_memc_status_3				dmc_memc_status_3			
		config	Ready	Paused	Low_power	config	Ready	Paused	Low_power
dmc_memc_status_3	RO	○	○	○	○	—	—	—	—
dmc_memc_cmd_3	WO	—	—	—	—	○	○	○	○
dmc_direct_cmd_3	WO	—	—	—	—	○	×	×	×
dmc_memory_cfg_3	R/W	○	○	○	○	○	×	×	×
dmc_refresh_prd_3	R/W	○	○	○	○	○	×	×	×
dmc_cas_latency_3	R/W	○	○	○	○	○	×	×	×
dmc_t_dqss_3	R/W	○	○	○	○	○	×	×	×
dmc_t_mrd_3	R/W	○	○	○	○	○	×	×	×
dmc_t_ras_3	R/W	○	○	○	○	○	×	×	×
dmc_t_rc_3	R/W	○	○	○	○	○	×	×	×
dmc_t_rcd_3	R/W	○	○	○	○	○	×	×	×
dmc_t_rfc_3	R/W	○	○	○	○	○	×	×	×
dmc_t_rp_3	R/W	○	○	○	○	○	×	×	×
dmc_t_rrd_3	R/W	○	○	○	○	○	×	×	×
dmc_t_wr_3	R/W	○	○	○	○	○	×	×	×
dmc_t_wtr_3	R/W	○	○	○	○	○	×	×	×
dmc_t_xp_3	R/W	○	○	○	○	○	×	×	×
dmc_t_xsr_3	R/W	○	○	○	○	○	×	×	×
dmc_t_esr_3	R/W	○	○	○	○	○	×	×	×
dmc_id_0_cfg_3 dmc_id_1_cfg_3 dmc_id_2_cfg_3 dmc_id_3_cfg_3	R/W	○	○	○	○	○	×	×	○
dmc_chip_0_cfg_3	R/W	○	○	○	○	○	×	×	○
dmc_user_config_3	WO	—	—	—	—	○	○	○	○

1. dmc_memc_status_3 (DMC Memory Controller Status Register)

Address = (0xF430_0000) + (0x0000)

Bit	Bit Symbol	Type	Reset Value	Description
[31:10]	–	–	Undefined	Read as undefined.
[9]	memory_banks	RO	0y1	DMC がサポートする最大バンク数 の設定値 : (4 バンクに固定)
[8:7]	Reserved	–	Undefined	Read as undefined.
[6:4]	memory_ddr	RO	0y000	DMC がサポートする SDRAM 種類 : 0y000 = SDR SDRAM 0y001 = Reserved 0y011 = Reserved 0y010 = Reserved 0y1xx = Reserved
[3:2]	memory_width	RO	0y01	外部メモリバス幅: 0y00 = 16-bit 0y01 = Reserved 0y10 = Reserved 0y11 = Reserved
[1:0]	memc_status	RO	0y00	メモリコントローラの状態: 0y00 = Config 0y01 = Ready 0y10 = Paused 0y11 = Low-power

(個別説明)

a. <memory_banks>

DMC がサポートする最大バンク数 を示します。
4 バンクに固定されています。

b. <memory_ddr>

DMC がサポートする SDRAM の種類を示します。
0y000 (SDR SDRAM)に固定されています。

c. <memory_width>

外部メモリバス幅を示します。

0y00 = 16-bit

0y01 = Reserved

0y10 = Reserved

0y11 = Reserved

d. <memc_status>

メモリコントローラの状態を示します。

0y00 = Config

0y01 = Ready

0y10 = Paused

0y11 = Low-power

2. dmc_memc_cmd_3 (DMC Memory Controller Command Register)

Address = (0xF430_0000) + (0x0004)

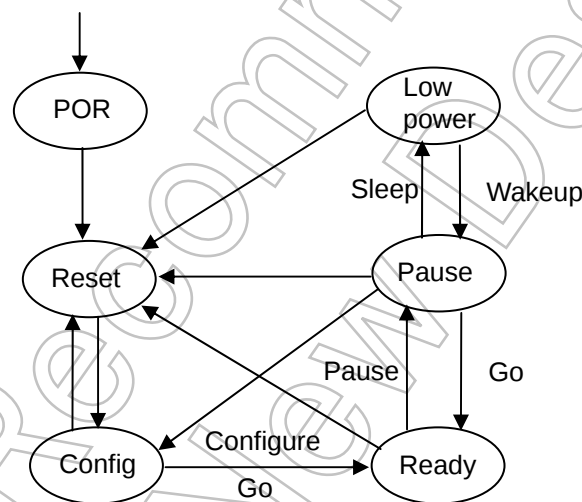
Bit	Bit Symbol	Type	Reset Value	Description
[31:3]	—	—	Undefined	Read as undefined. Write as zero.
[2:0]	memc_cmd	WO	—	メモリコントローラ状態を変更: 0y000 = Go 0y001 = Sleep 0y010 = Wakeup 0y011 = Pause 0y100 = Configure

(個別説明)

a. <memc_cmd>

このレジスタの設定により、DMC のステートマシンを変更できます。前の状態変更のコマンド実行中は、完了までを待って新規コマンドを発行します。

Low-power 実施フローは下記の図に示します。



DMC 状態遷移

Reset 状態から抜けると自動的に Config 状態となります。また、Pause 状態からは Config コマンドで Config 状態に遷移します。各レジスタの設定は Config 状態で行う必要があります。

Ready 状態に遷移すると、SDRAM への Read/Write が可能となります。また、Read/Write を実行することで SDRAM は IDLE 状態から ACTIVE 状態へ移行します。

Ready 状態から Pause コマンドにて Pause 状態へ遷移します。この時の SDRAM の状態は、SDRAM への前コマンドによって異なります。Read/Write 直後であれば ACTIVE 状態、AutoRefresh 直後であれば IDLE 状態となります^{注)}。

Pause 状態から Sleep コマンドにて Low power 状態に遷移すると、All Bank Precharge を実施後、CKE="L"にして SDRAM が自動的に Self-refresh 状態になります。

Low power 状態から Wakeup コマンドにて Pause 状態に遷移すると、self_refresh Exit コマンドを発行します。SDRAM は自動的に self_refersch 状態から抜け IDLE 状態に移行します。

注) SDRAM を ACTIVE 状態から IDLE 状態へ移行させたい場合は、2 種類の方法があります。dmc_direct_cmd_3<memory_cmd>0y00 = Prechargeall, 0y01 = Autorefresh のいずれかを設定してください。

3. dmc_direct_cmd_3 (DMC Direct Command Register)

外部メモリに対する各コマンド発行、および外部メモリのモードレジスタを設定します。

このレジスタで外部メモリの初期設定を行います。

Address = (0xF430_0000) + (0x0008)

Bit	Bit Symbol	Type	Reset Value	Description
[31:22]	–	–	Undefined	Read as undefined. Write as zero.
[21:20]	chip_nمبر	WO	–	常に 0y00 を設定してください。
[19:18]	memory_cmd	WO	–	要求コマンド : 0y00 = Prechargeall 0y01 = Autorefresh 0y10 = Modereg or 拡張 Modereg 0y11 = NOP
[17:16]	bank_addr	WO	–	Modereg をアクセスする時の外部メモリバンクアドレスビット設定 0y00 = bank0 0y01 = bank1 0y10 = bank2 0y11 = bank3
[15:14]	–	–	Undefined	Read as undefined. Write as zero.
[13:0]	addr_13_to_0	WO	–	Modereg をアクセスする時のコマンドのデータ設定 (外部メモリアドレスビットに対応)

(個別説明)

a. <memory_cmd>

要求コマンドを選択します。

0y00 = Prechargeall

0y01 = Autorefresh

0y10 = Modereg or 拡張 modereg

0y11 = NOP

b. <bank_addr>

Modereg をアクセスする時の外部メモリバンクアドレスビットを設定します。

0y00 = bank0

0y01 = bank1

0y10 = bank2

0y11 = bank3

c. <addr_13_to_0>

Modereg をアクセス時のコマンドのデータを設定します(外部メモリアドレスビットに対応)。

4. dmc_memory_cfg_3 (DMC Memory Configuration Register)

Address = (0xF430_0000) + (0x000C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:23]	—	—	Undefined	Read as undefined. Write as zero.
[22:21]	active_chips	R/W	0y00	常に 0y00 を設定してください。
[20:18]	—	—	Undefined	Read as undefined. Write as zero.
[17:15]	memory_burst	R/W	0y010	SDRAM をリード/ライトするときのバースト長設定 0y000 = Burst 1 0y001 = Burst 2 0y010 = Burst 4 0y011 = Burst 8 0y100 = Burst 16 ^{注)} Other = Reserved
[14]	stop_mem_clock	R/W	0y0	メモリクロック停止: 0y0 = Disable 0y1 = Enable
[13]	auto_power_down	R/W	0y0	SDRAM 自動 Powerdown 設定: 0y0 = Disable 0y1 = Enable
[12:7]	power_down_prd	R/W	0y000000	SDRAM 自動 Powerdown メモリクロック数設定: (Min 値 = 1) 0y000001 ~ 0y111111
[6]	ap_bit	R/W	0y0	メモリアドレスで auto-precharge ビット位置: 0y0 = address bit 10 0y1 = address bit 8
[5:3]	row_bits	R/W	0y100	row アドレスビット数: 0y000 = 11 bits 0y001 = 12 bits 0y010 = 13 bits 0y011 = 14 bits 0y100 = 15 bits 0y101 = 16 bits Other = Reserved
[2:0]	column_bits	R/W	0y000	column アドレスビット数: 0y000 = 8 bits 0y001 = 9 bits 0y010 = 10 bits 0y011 = 11 bits 0y100 = 12 bits Other = Reserved

(個別説明)

a. <memory_burst>

メモリをアクセスするコントローラのバースト長を設定します。

また、dmc_direct_cmd_3 で設定したメモリのバースト長と一致させてください。

b. <stop_mem_clock>

SDRAM にアクセスしていない時、クロック供給を停止します。再度 SDRAM へのアクセス要求があると、自動的にクロックを再スタートします。

注) 利用する SDRAM によっては、アクセスしていない期間中のクロックの停止を許可していない製品が存在いたします。本機能をご使用の際は、使用する SDRAM の SPEC を十分に確認してください。

注) メモリクロック停止機能と SDRAM 自動 Powerdown 機能の併用はできません。どちらか一方のみの利用が前提となります。

c. <auto_power_down>

SDRAM へのアクセス要求が無くなり、メモリコントローラのコマンド FIFO が空き状態になると、自動的に power_down_prd で設定されたクロック後に、CKE を Disable 状態に変化させ、SDRAM を Powerdown 状態に Entry することができます。再度 SDRAM へのアクセス要求があると、自動的に CKE を Enable 状態にして Powerdown から Exit します。

注) メモリクロック停止機能と SDRAM 自動 Powerdown 機能の併用はできません。どちらか一方のみの利用が前提となります。

d. <row_bits>,<column_bits>

row および column アドレスの設定をします。

接続可能な最大メモリサイズは、Column アドレスと row アドレスの総和で制限されます。

16bit バスの場合 : $R+C=26$ ビット (128Mbyte) までで、その 4Bank 分 (512Mbyte)

5. dmc_refresh_prd_3 (DMC Refresh Period Register)

Address = (0xF430_0000) + (0x0010)

Bit	Bit Symbol	Type	Reset Value	Description
[31:15]	—	—	Undefined	Read as undefined. Write as zero.
[14:0]	refresh_prd	R/W	0x0A60	Auto-refresh の周期(メモリクロック数): 0x0000 ~ 0x7FFF

(個別説明)

a. <refresh_prd>

Refresh カウンタは dmc_refresh_prd_3 レジスタで設定した値(メモリクロック数)からデクリメントして、0 になると Auto-refresh コマンドを外部メモリに発行します。

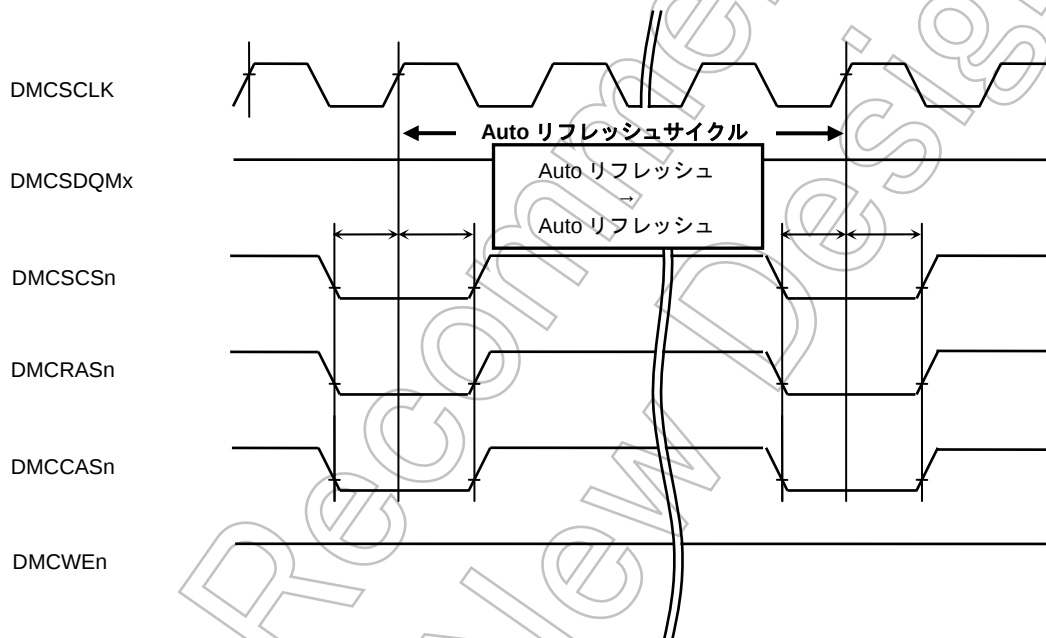


図 3.10.3 Auto リフレッシュサイクル動作例

6. dmc_cas_latency_3 (DMC CAS Latency Register)

Address = (0xF430_0000) + (0x0014)

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	—	—	Undefined	Read as undefined. Write as zero.
[3:1]	cas_latency	R/W	0y011	CAS レイテンシーの設定 (メモリクロック数) 0y000 ~ 0y111
[0]	—	—	Undefined	Read as undefined. Write as zero.

(個別説明)

a. <cas_latency>

CAS レイテンシーを設定します。(メモリクロック数)

0y000 ~ 0y111

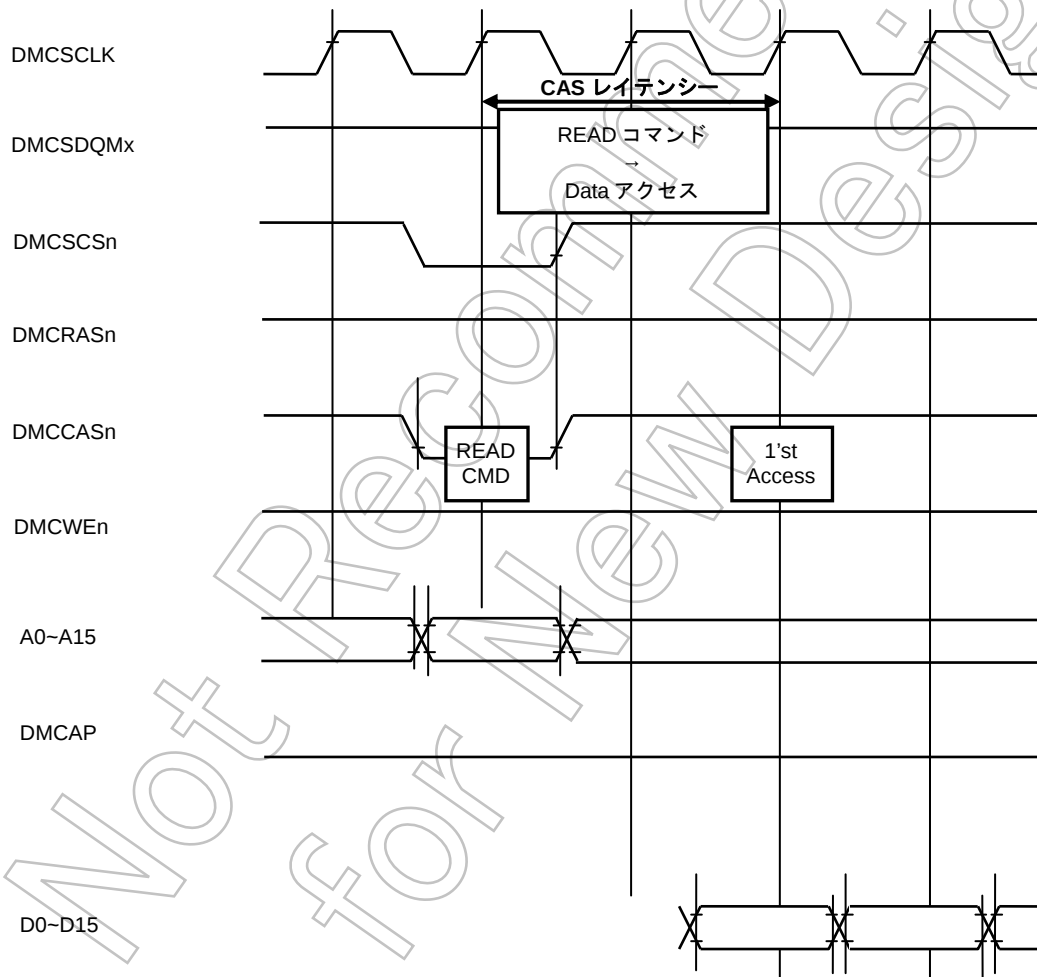


図 3.10.4 CAS レイテンシー時間例(CL=2)

7. dmc_t_dqss_3 (DMC t_dqss Register)

Address = (0xF430_0000) + (0x0018)

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	—	—	Undefined	Read as undefined. Write as zero.
[1:0]	t_dqss	R/W	0y01	DQS の設定 (メモリクロック数) 常に動作前に 0y00 とライトしてください。

(個別説明)

MPMC0 では、DQS 信号は存在しません。必ず初期設定時に 0y00 をライトしてください。

8. dmc_t_mrd_3 (DMC t_mrd Register)

Address = (0xF430_0000) + (0x001C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:7]	—	—	Undefined	Read as undefined. Write as zero.
[6:0]	t_mrd	R/W	0y0000010	モードレジスタコマンド時間 (メモリクロック数) 0x00 ~ 0x7F

(個別説明)

a. <t_mrd>

ダイレクトコマンドレジスタ(dmc_direct_cmd_3<addr_13_to_0>)にて設定するモードレジスタコマンド時間から、他のすべてのコマンドまでの時間を設定します。

(メモリクロック数)

0x00 ~ 0x7F

他の AC 設定または他の動作との兼ね合いにより、設定されたサイクル数よりも長いサイクルで動作する場合があります。よって、本設定は最低クロック数を設定します。

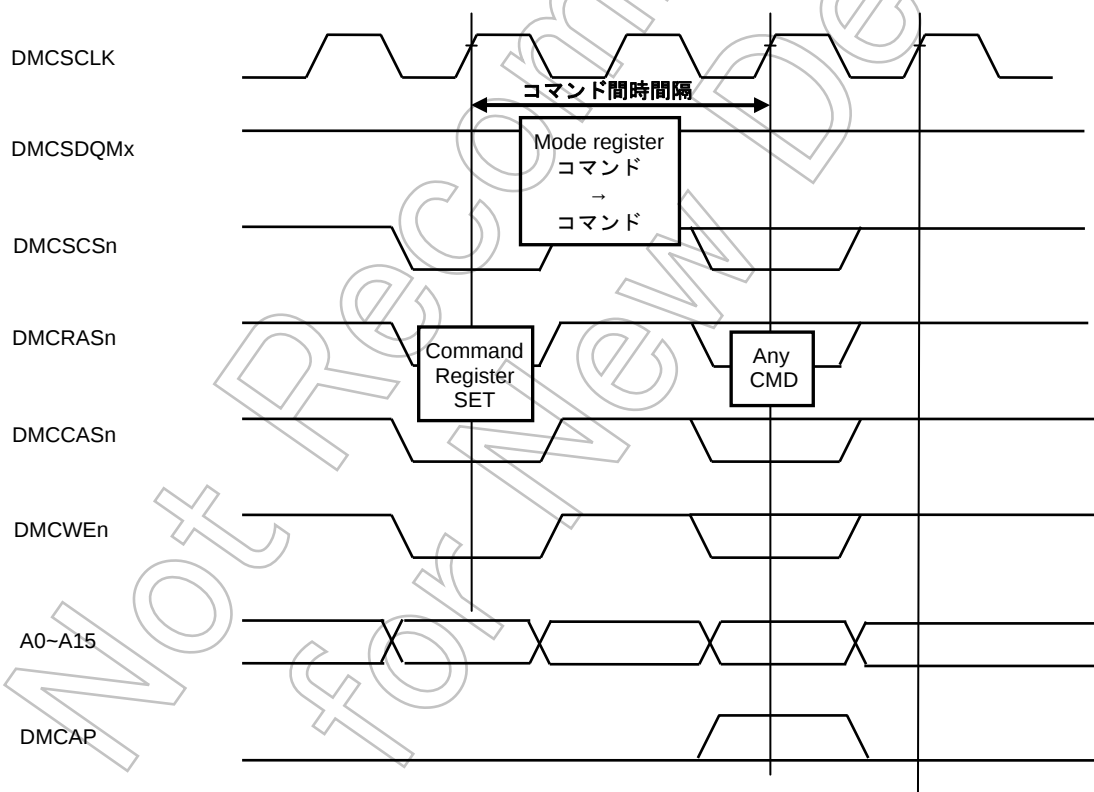


図 3.10.5 モードレジスタライトから他コマンドまでの時間

9. dmc_t_ras_3 (DMC t_ras Register)

Address = (0xF430_0000) + (0x0020)

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	—	—	Undefined	Read as undefined. Write as zero.
[3:0]	t_ras	R/W	0x7	RAS から Precharge の時間 (メモリクロック数) 0x0 ~ 0xF

(個別説明)

a. <t_ras>

RAS から Precharge の時間を設定します。(メモリクロック数)

0x0 ~ 0xF

他の AC 設定または他の動作との兼ね合いにより、設定されたサイクル数よりも長いサイクルで動作する場合があります。よって、本設定は最低クロック数を設定します。

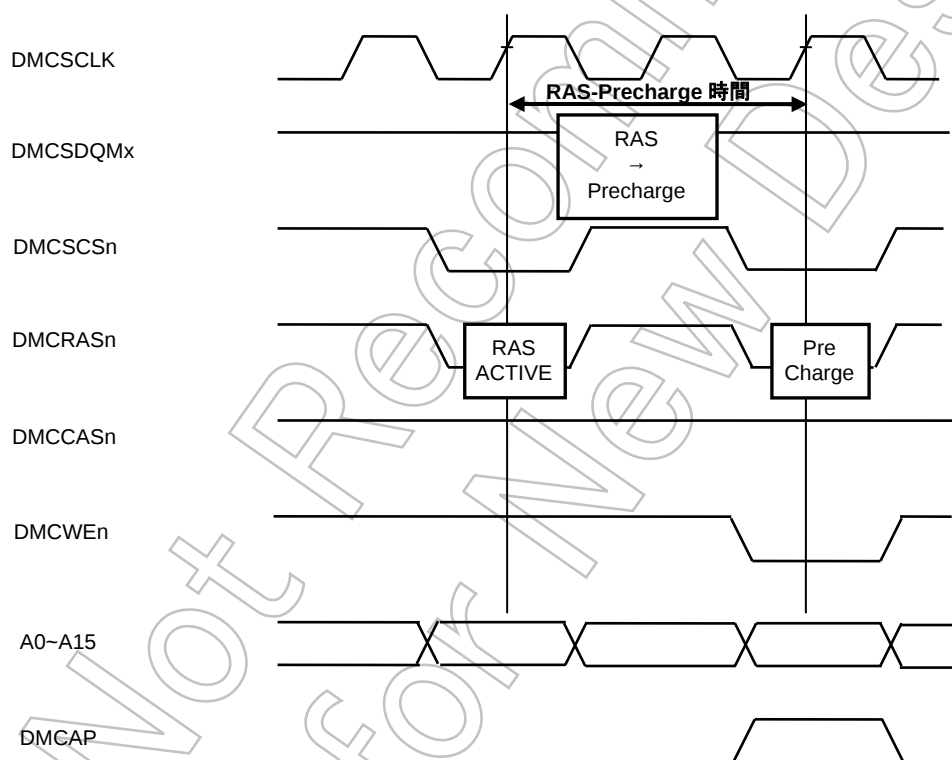


図 3.10.6 Active から Precharge までの時間例

10. dmc_t_rc_3 (DMC t_rc Register)

Address = (0xF430_0000) + (0x0024)

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	—	—	Undefined	Read as undefined. Write as zero.
[3:0]	t_rc	R/W	0y1011	Active バンク A から Active バンク A の Delay 時間 (メモリクロック数) 0x0 ~ 0xF

(個別説明)

a. <t_rc>

同一 BANK 上での、Active バンクコマンドから Active バンクコマンドの Delay 時間を設定します。(メモリクロック数)

0x0 ~ 0xF

他の AC 設定または他の動作との兼ね合いにより、設定されたサイクル数よりも長いサイクルで動作する場合があります。よって、本設定は最低クロック数を設定します。

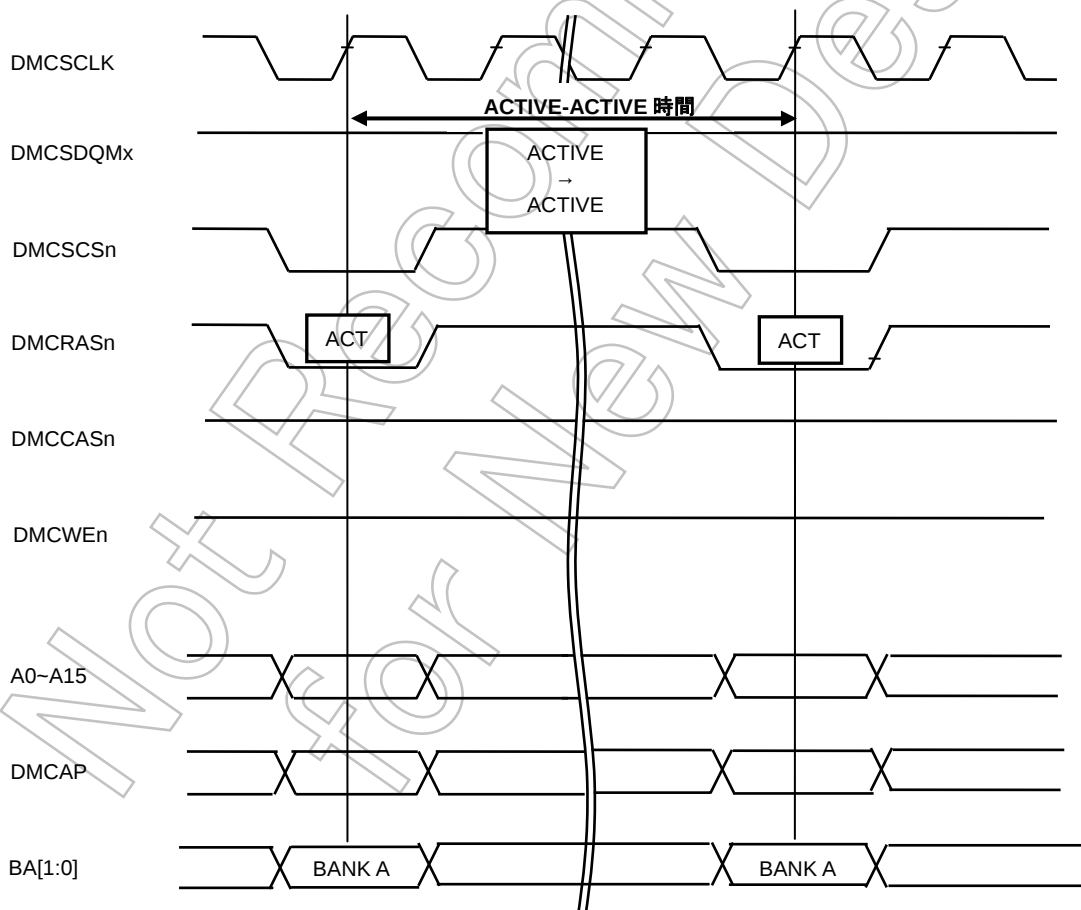


図 3.10.7 Active バンク A から Active バンク A 例

11. dmc_t_rcd_3 (DMC t_rcd Register)

$$\text{Address} = (0xF430_0000) + (0x0028)$$

Bit	Bit Symbol	Type	Reset Value	Description
[31:6]	–	–	Undefined	Read as undefined. Write as zero.
[5:3]	schedule_rcd	R/W	0y011	RAS から CAS まで Min delay 設定 (t_rcd の設定値 -3)で設定してください。
[2:0]	t_rcd	R/W	0y101	RAS から CAS まで Min delay 設定 (メモリクロック数) 0y000 ~ 0y111

(個別説明)

a. <schedule_rcd>

RAS から CAS まで Min delay を設定します。(メモリクロック数)
(t_rcd の設定値 -3)で設定してください。

b. $\langle t_{\text{rcd}} \rangle$

RAS から CAS まで Min delay を設定します。(メモリクロック数)
0y000 ~ 0y111

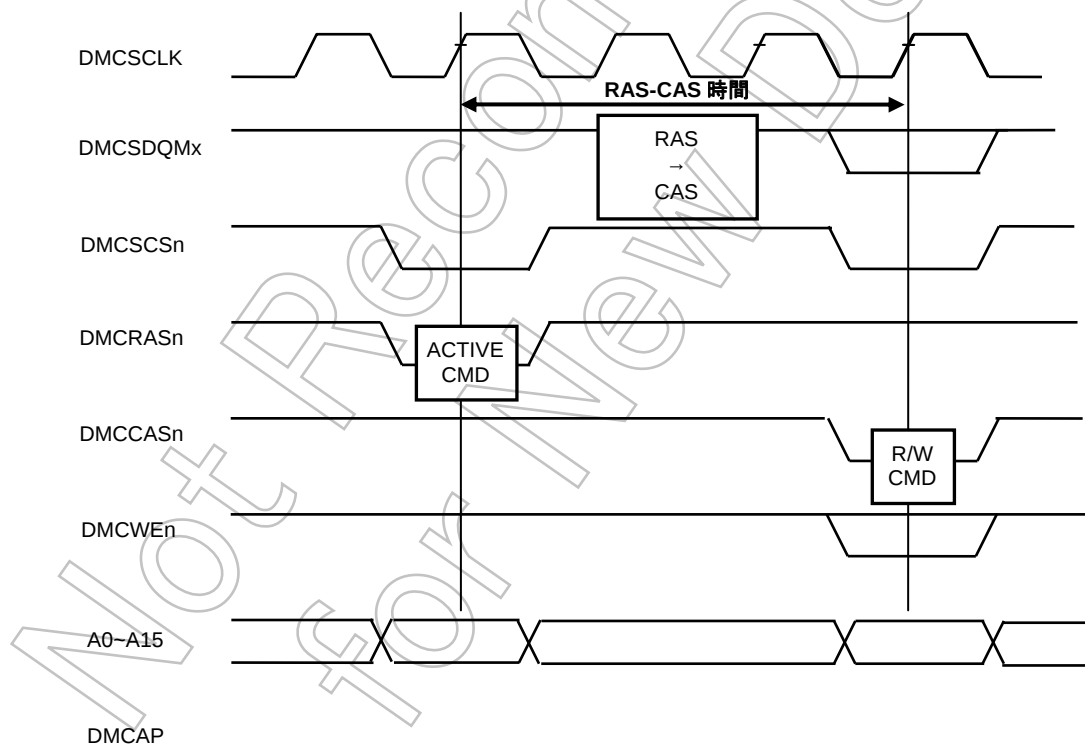


図 3.10.8 Active からリードコマンドのタイミング例

12. dmc_t_rfc_3 (DMC t_rfc Register)

Address = (0xF430_0000) + (0x002C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:10]	—	—	Undefined	Read as undefined. Write as zero.
[9:5]	schedule_rfc	R/W	0y10000	Autorefresh コマンド時間設定 (t_rfc の設定値 -3)で設定してください。
[4:0]	t_rfc	R/W	0y10010	Autorefresh コマンド時間設定 (メモリクロック数) 0y00000 ~ 0y11111

(個別説明)

a. <schedule_rfc>

Autorefresh コマンド時間を設定します。
(t_rfc の設定値 -3)で設定してください。

b. <t_rfc>

Autorefresh コマンド時間を設定します。(メモリクロック数)
0y00000 ~ 0y11111

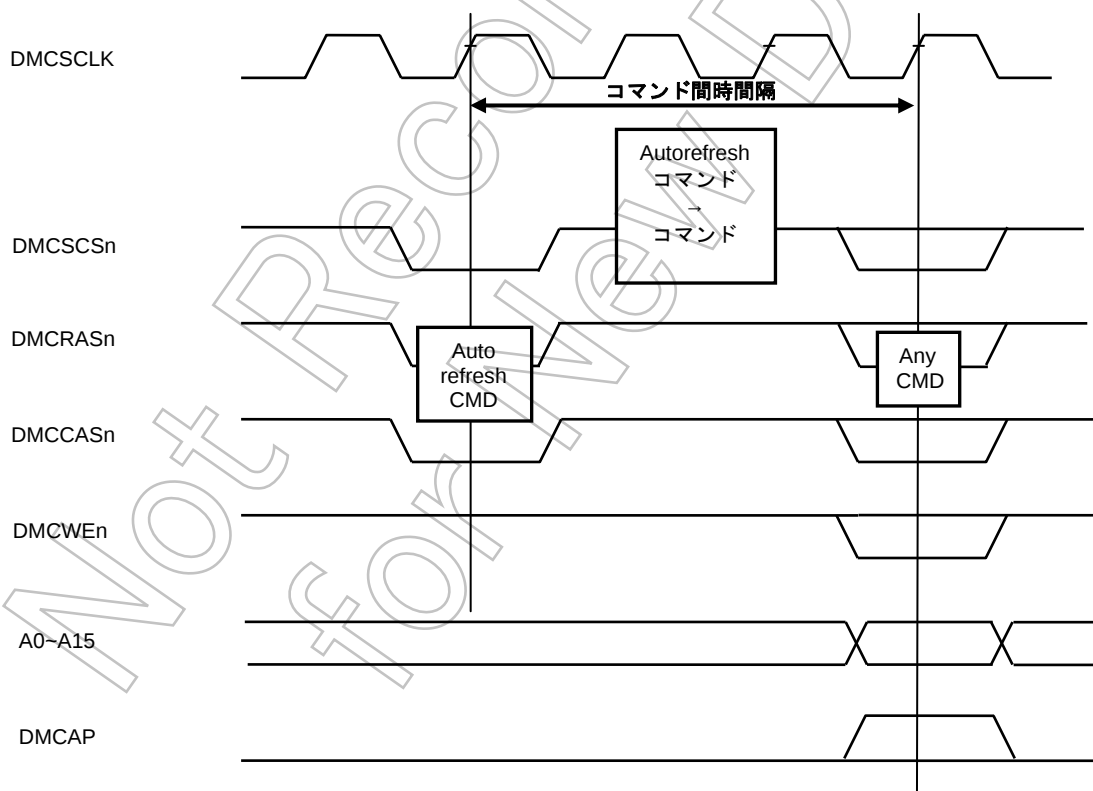


図 3.10.9 Autorefresh コマンドから他コマンドのタイミング例

13. dmc_t_rp_3 (DMC t_rp Register)

Address = (0xF430_0000) + (0x0030)

Bit	Bit Symbol	Type	Reset Value	Description
[31:6]	—	—	Undefined	Read as undefined. Write as zero.
[5:3]	schedule_rp	R/W	0y011	RAS に Precharge Delay 設定 (t_rp の設定値 -3) で設定してください。
[2:0]	t_rp	R/W	0y101	Precharge から RAS までの時間を設定(メモリクロック数) 0y000 ~ 0y111

(個別説明)

a. <schedule_rp>

Precharge から RAS までの 時間を設定します。

(t_rp の設定値 -3) で設定してください。

c. <t_rp>

Precharge から RAS までの 時間を設定します。(メモリクロック数)

0y000 ~ 0y111

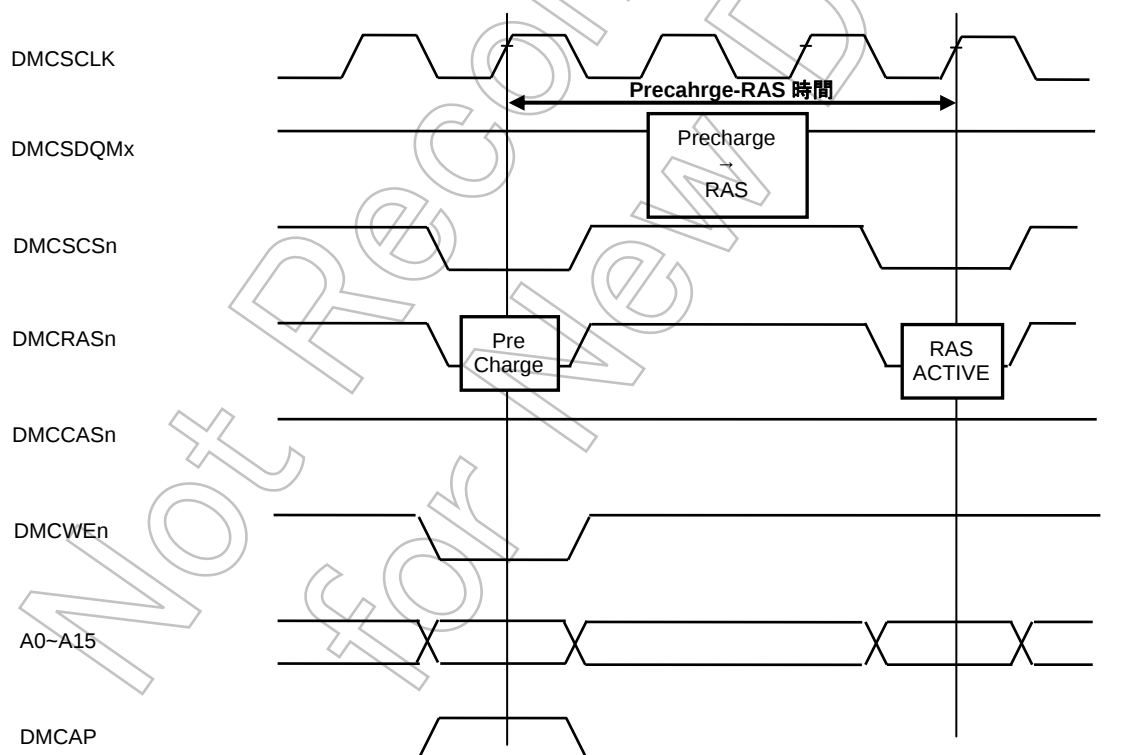


図 3.10.10 Precharge から他のコマンド(Autorefresh コマンドを含む)までの時間

14. dmc_t_rrd_3 (DMC t_rrd Register)

Address = (0xF430_0000) + (0x0034)

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	—	—	Undefined	Read as undefined. Write as zero.
[3:0]	t_rrd	R/W	0y0010	Active バンク A から Active バンク B の Delay 時間 (メモリクロック数) 0x0 ~ 0xF

(個別説明)

a. <t_rrd>

Active バンク A から Active バンク B の Delay 時間を設定します。(メモリクロック数)

0x0 ~ 0xF

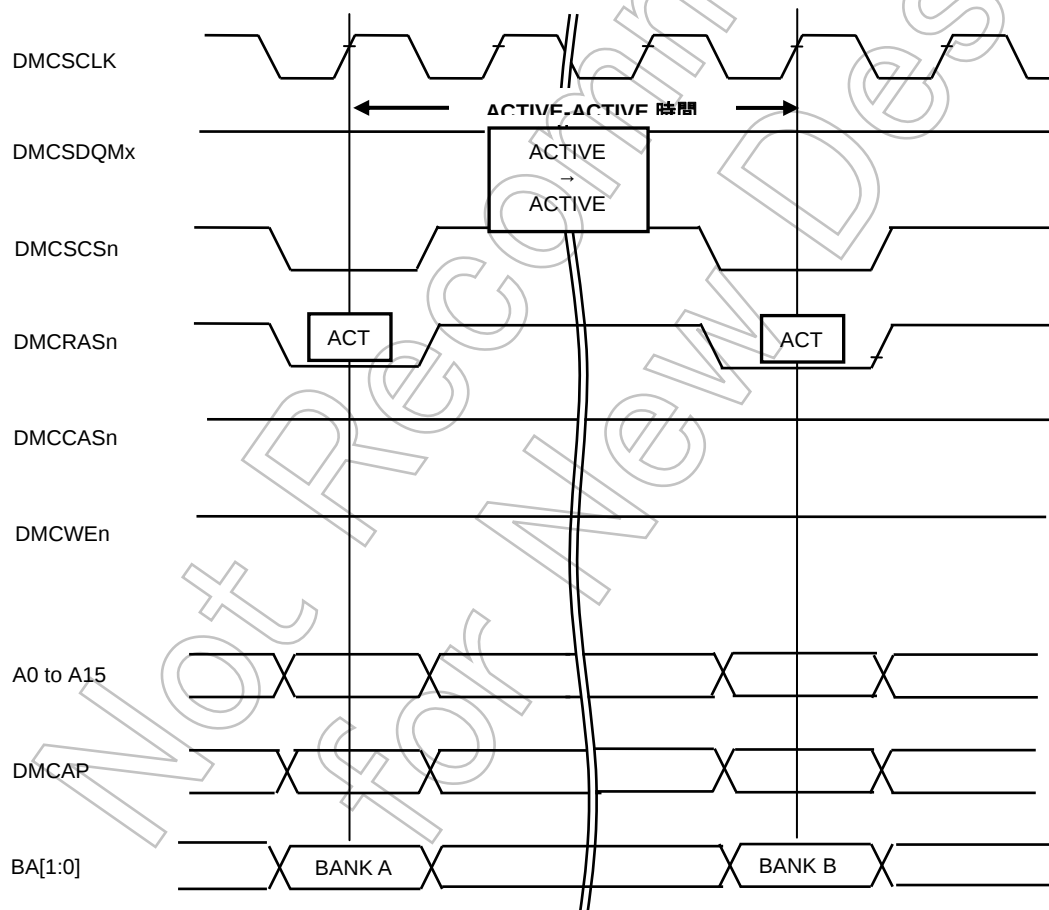


図 3.10.11 Active バンク A から他の Active バンク B の時間

15. dmc_t_wr_3 (DMC t_wr Register)

Address = (0xF430_0000) + (0x0038)

Bit	Bit Symbol	Type	Reset Value	Description
[31:3]	—	—	Undefined	Read as undefined. Write as zero.
[2:0]	t_wr	R/W	0y011	ライトの最終データから Precharge までの 設定値 0y000 ~ 0y111

(個別説明)

a. <t_wr>

ライトの最終データから Precharge までの 期間を設定します (メモリクロック数)。

実際の期間(メモリクロック数) : t_wr の設定値 + 1、ただし 0y000 の設定した場合は 9 クロックになります。

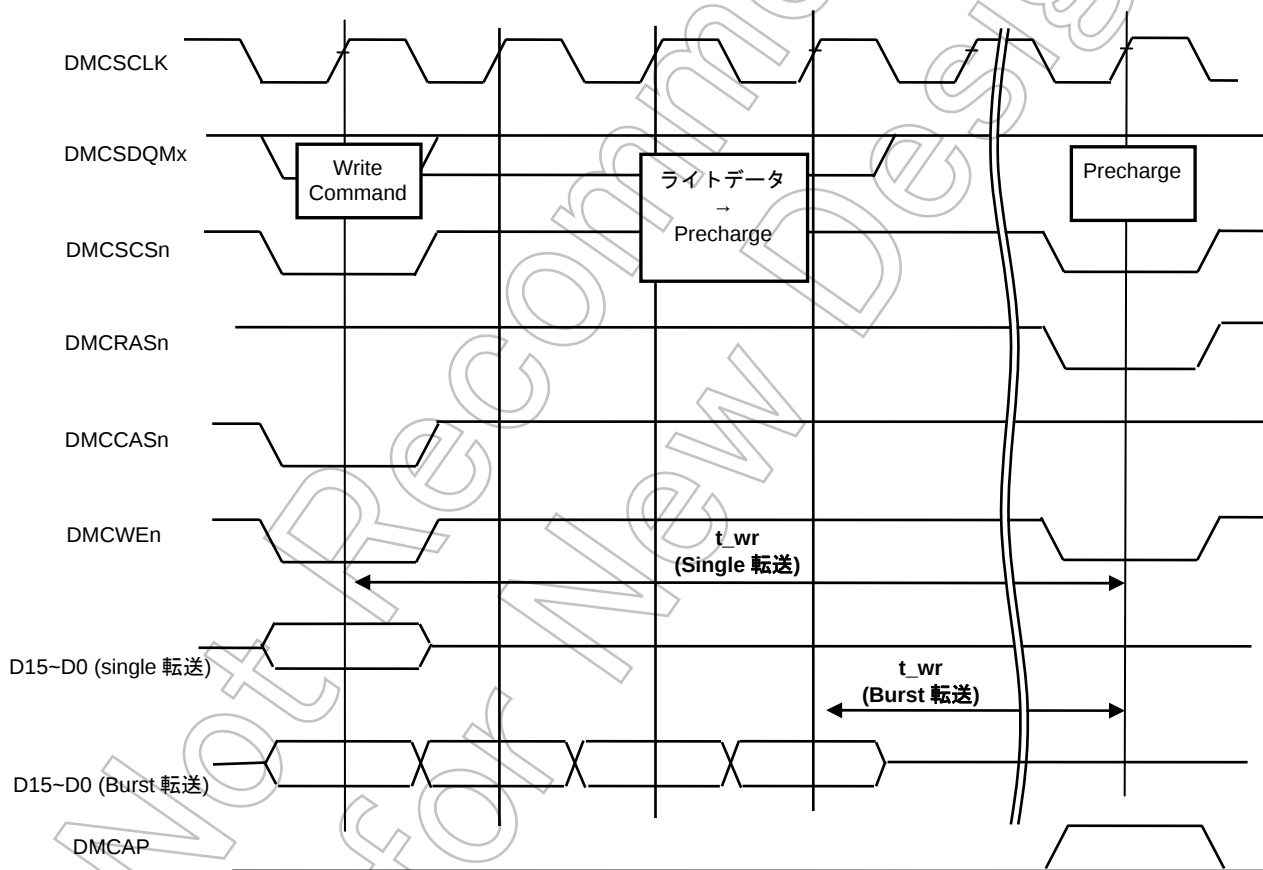


図 3.10.12 ライトの最終データから Precharge までの時間

16. dmc_t_wtr_3 (DMC t_wtr Register)

Address = (0xF430_0000) + (0x003C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:3]	—	—	Undefined	Read as undefined. Write as zero.
[2:0]	t_wtr	R/W	0y010	ライトの最終データからリードコマンドまでの設定値 (メモリクロック数) 0y000 ~ 0y111

(個別説明)

a. <t_wtr>

ライトの最終データからリードコマンドまでの期間を設定します (メモリクロック数)

注) 0y000 時、8 メモリクロックになります。

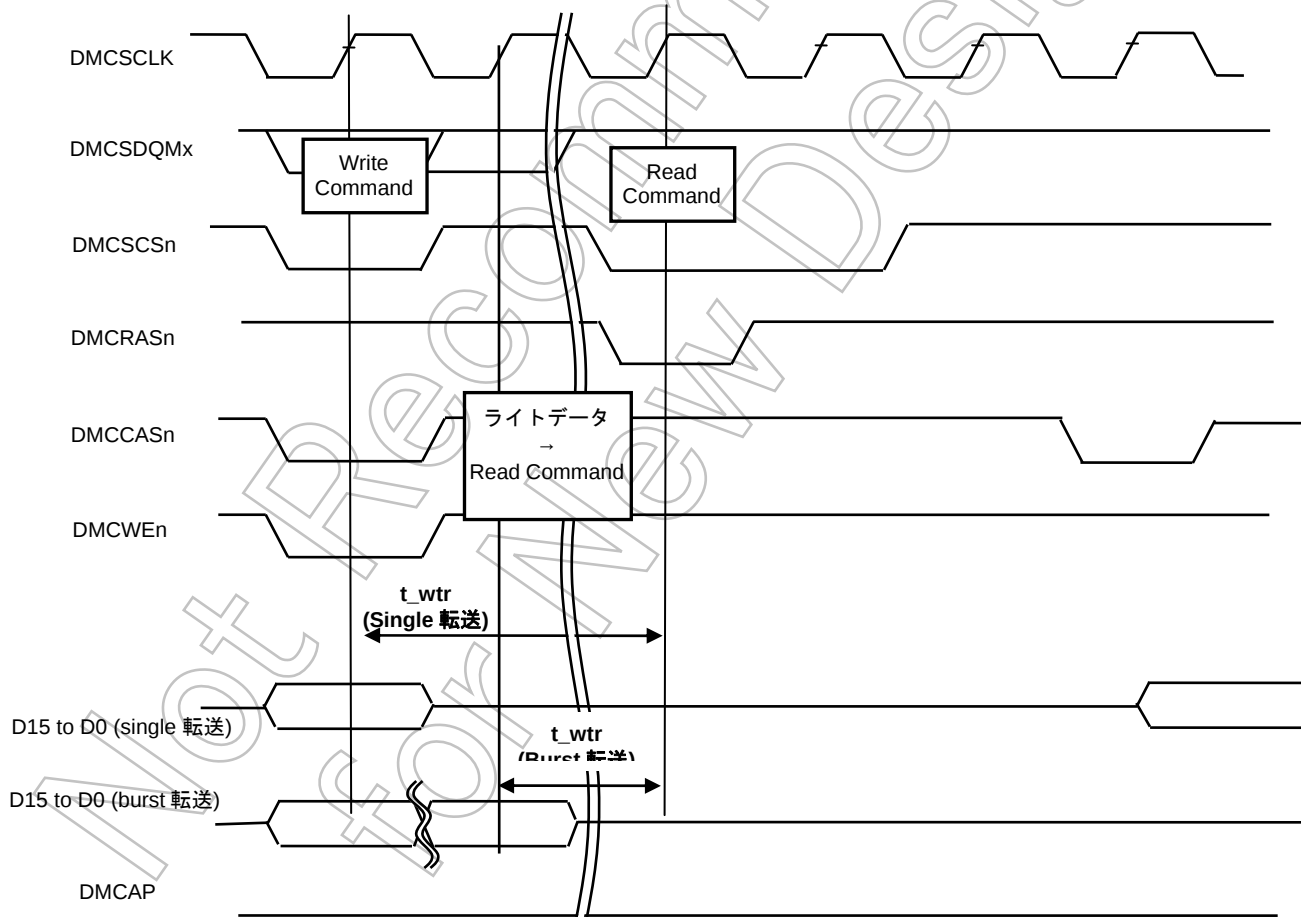


図 3.10.13 ライトの最終データからリードコマンドまでの時間

17. dmc_t_xp_3 (DMC t_xp Register)

Address = (0xF430_0000) + (0x0040)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	—	—	Undefined	Read as undefined. Write as zero.
[7:0]	t_xp	R/W	0x01	Powerdown から Exit 時間の設定値(メモリクロック数) 0x00 ~ 0xFF

(個別説明)

a. <t_xp>

Powerdown Exit から他コマンドまでの時間を設定します。(メモリクロック数)

実際の期間(メモリクロック数) : t_xp の設定値 + 1

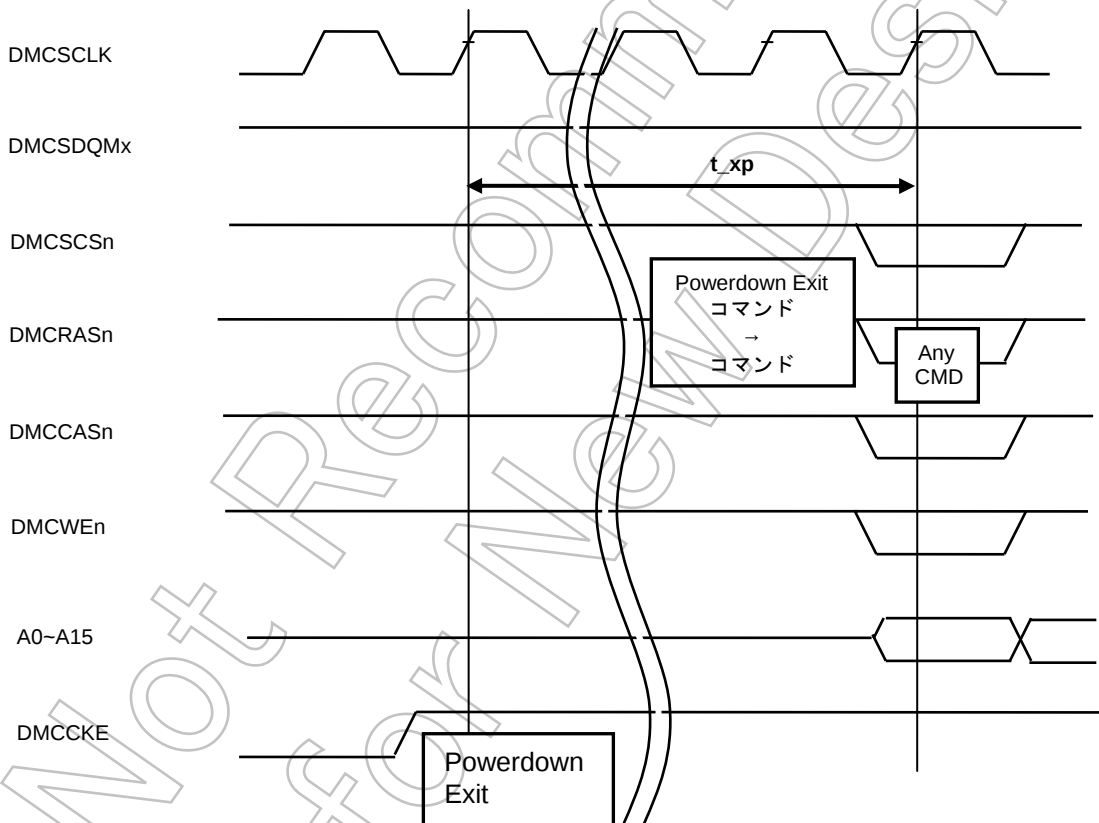


図 3.10.14 Powerdown Exit から他のコマンドまでの時間

18. dmc_t_xsr_3 (DMC t_xsr Register)

Address = (0xF430_0000) + (0x0044)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	—	—	Undefined	Read as undefined. Write as zero.
[7:0]	t_xsr	R/W	0x0A	Self-refresh Exit コマンド時間設定 : (メモリクロック数) 0x00 ~ 0xFF

(個別説明)

a. <t_xsr>

Self-refresh Exit コマンドから他のコマンドまでの時間を設定します。(メモリクロック数)

0x00 ~ 0xFF

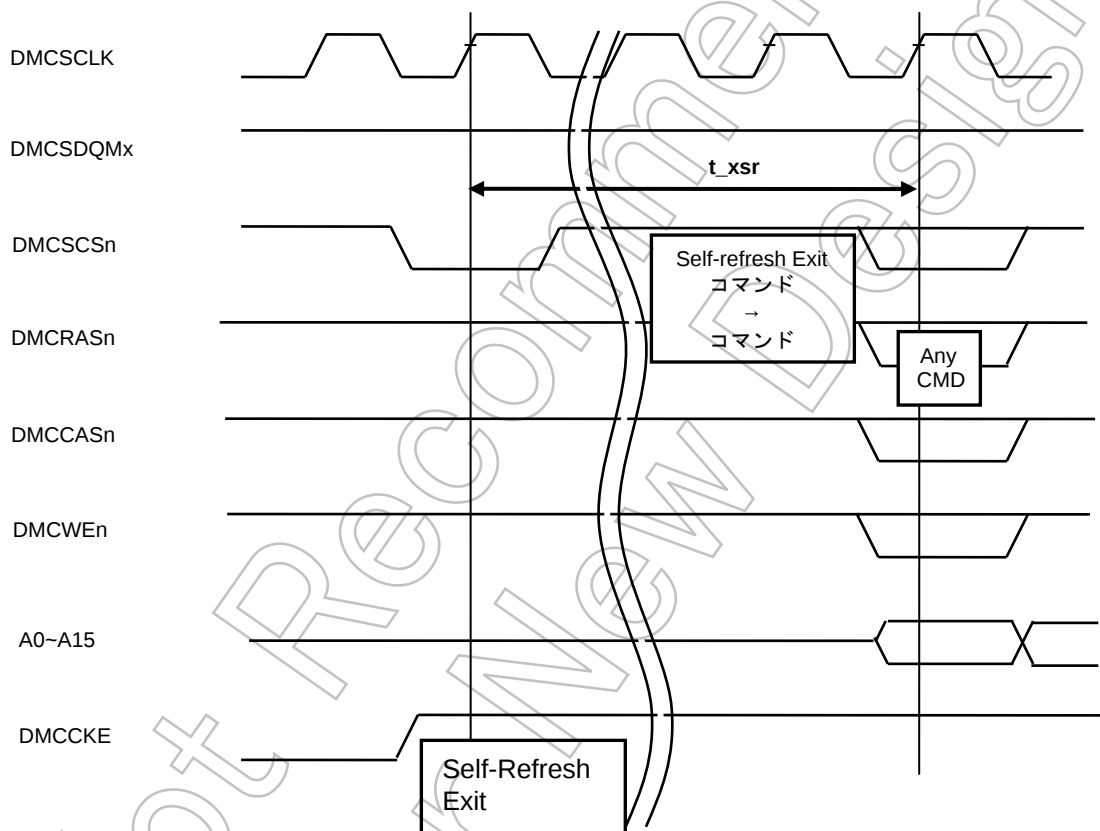


図 3.10.15 Self-refresh Exit コマンドから他のコマンドまでの時間

19. dmc_t_esr_3 (DMC t_esr Register)

Address = (0xF430_0000) + (0x0048)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	—	—	Undefined	Read as undefined. Write as zero.
[7:0]	t_esr	R/W	0x14	Self-refresh Entry から Exit まで実行最低限時間の設定 : (メモリクロック数) 0x00 ~ 0xFF

注) Self-refersh は Wakeup の実行により Exit します。

このレジスタは Self-refresh Entry から Exit まで最低限時間を設定します。

(個別説明)

a. <t_esr>

Self-refresh 実行時間を設定します。(メモリクロック数)

0x00 ~ 0xFF

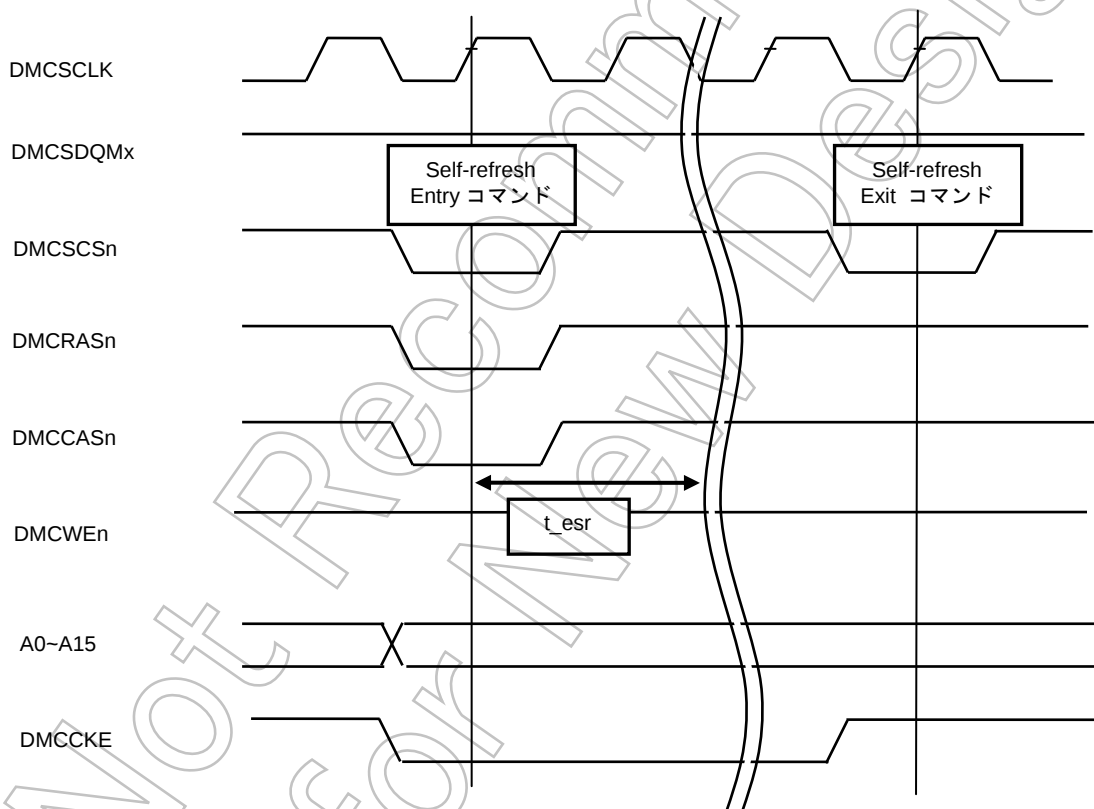


図 3.10.16 Self-Refresh Entry から Exit までの最低限時間実行時間

20. dmc_id_<0-3>_cfg_3 Registers

dmc_id_<0-3>_cfg_3 は 4 個のレジスタ QoS を設定します。

Address = (0xF430_0000) + (0x0100)

Address = (0xF430_0000) + (0x0104)

Address = (0xF430_0000) + (0x0108)

Address = (0xF430_0000) + (0x010C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:10]	–	–	Undefined	Read as undefined. Write as zero.
[9:2]	qos_max	R/W	0x00	maximum QoS : 0x00 ~ 0xFF
[1]	qos_min	R/W	0y0	minimum QoS 選択 : 0y0 = qos_max mode 0y1 = qos_min mode
[0]	qos_enable	R/W	0y0	QoS の Enable: 0y0 = Disable 0y1 = Enable

(個別説明)

QoS 設定レジスタ一覧

レジスタ	アドレス	対応 AHB バス
dmc_id_0_cfg_3	(0xF430_0000) + (0x0100)	AHB0 : CPU Data
dmc_id_1_cfg_3	(0xF430_0000) + (0x0104)	AHB1 : CPU Inst
dmc_id_2_cfg_3	(0xF430_0000) + (0x0108)	AHB2 : LCDC
dmc_id_3_cfg_3	(0xF430_0000) + (0x010C)	AHB3 : マルチレイヤーバスマトリクス 2 (LCDDA,USB,DMAC1,DMAC2)

a. <qos_max>

maximum QoS を設定します。

0x00 ~ 0xFF

b. <qos_min>

minimum QoS 選択:

0y0 = qos_max mode

0y1 = qos_min mode

QoS min は QoS max より高い優先度を持ちます。

c. <qos_enable>

QoS の Enable を設定します。

0y0 = Disable

0y1 = Enable

21. dmc_chip_0_cfg_3 (DMC chip_0_cfg Registers)

Address = (0xF430_0000) + (0x0200)

Bit	Bit Symbol	Type	Reset Value	Description
[31:17]	–	–	Undefined	Read as undefined. Write as zero.
[16]	brc_n_rbc	R/W	0y0	SDRAM アドレス構成 : 0y0 = row, bank, column 0y1 = bank, row, column
[15:8]	address_match	R/W	0xFF	スタートアドレス[31:24]を設定 : 0x00 ~ 0xFF
[7:0]	address_mask	R/W	0x00	スタートアドレス[31:24]のマスク値を設定 : 値 1 のビットはアドレス比較対象ビット 0x00 ~ 0xFF

(個別説明)

a. <brc_n_rbc>

SDRAM アドレス構成を選択します。

0y0 = row, bank, column

0y1 = bank, row, column

b. <address_match>

スタートアドレス[31:24]を設定するレジスタです。

512MB 未満のメモリを接続した場合は、設定した CS エリア以外の DMC エリア(未使用)をアクセスしないでください。

注) スタートアドレスを設定する場合は 3.3 章メモリマップを参照し、有効領域を確認した上で設定してください。

c. <address_mask>

このレジスタで CS 領域を設定します。

スタートアドレス[31:24]に対してどのビットの値を比較するか、比較しないかを設定します。

0y0 = 比較しない

0y1 = 比較する

設定例

16MB CS 領域 : <address_mask [7:0]> = 0y11111111

32MB CS 領域 : <address_mask [7:0]> = 0y11111110

|

512MB CS 領域 : <address_mask [7:0]> = 0y11100000

4GB CS 領域 : <address_mask [7:0]> = 0y00000000

22. dmc_user_config_3 (DMC user_config Register)

Address = (0xF430_0000) + (0x0304)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7]	Reserved	–	Undefined	Read as undefined. Write as zero.
[6:4]	dmclk_out1	WO	0y000	SDR SDRAM 常数設定 : 必ず 0y000 に設定してください
[3:1]	Reserved	–	Undefined	Read as undefined. Write as zero.
[0]	sdr_width	WO	0y0	対応する外部 SDR メモリのデータバス幅の設定 0y0: 16bit 0y1: Reserved

(個別説明)

a. <sdr_width>

対応する外部 SDR メモリのデータバス幅を設定します。

0y0 = 16bit

0y1 = Reserved

3.10.3.2 SMC (Static Memory Controller)

外部メモリ (NOR フラッシュメモリ、Mask ROM、SRAM 等) を制御する SMC (Static Memory Controller) を内蔵しています。

(1) SMC 機能概要

表 3.10.6 に、SMC の特長を示します。

表 3.10.6 SMC の特長

	特長
サポートメモリ	外部非同期 Static メモリ (NOR フラッシュメモリ、SRAM 等) セパレートバスのみサポート
データバス幅	16bit データバス幅をサポート
アクセス空間	チップセレクトによる 2 つの空間をサポート それぞれの最大アクセス空間 : SMCCS0n : 512MB SMCCS1n : 512MB
タイミング調整	レジスタによる AC タイミングの調整が可能 同期モード選択時のみ外部からのウェイト要求に対応
クロック	外部制御端子を生成するクロックをクロックコントローラの CLKCR5<SEL_SMC_MCLK> で $f_{HCLK} / f_{HCLK} / 2$ を選択可能
外部制御端子	D15~D0, A23~A0, SMCBE0n, SMCBE1n, SMCCS0n, SMCCS1n, SMCOEn, SMCWEEn

3.10.3.3

(2) SMC ブロック図

以下に、SMC 回路のブロック図を示します。

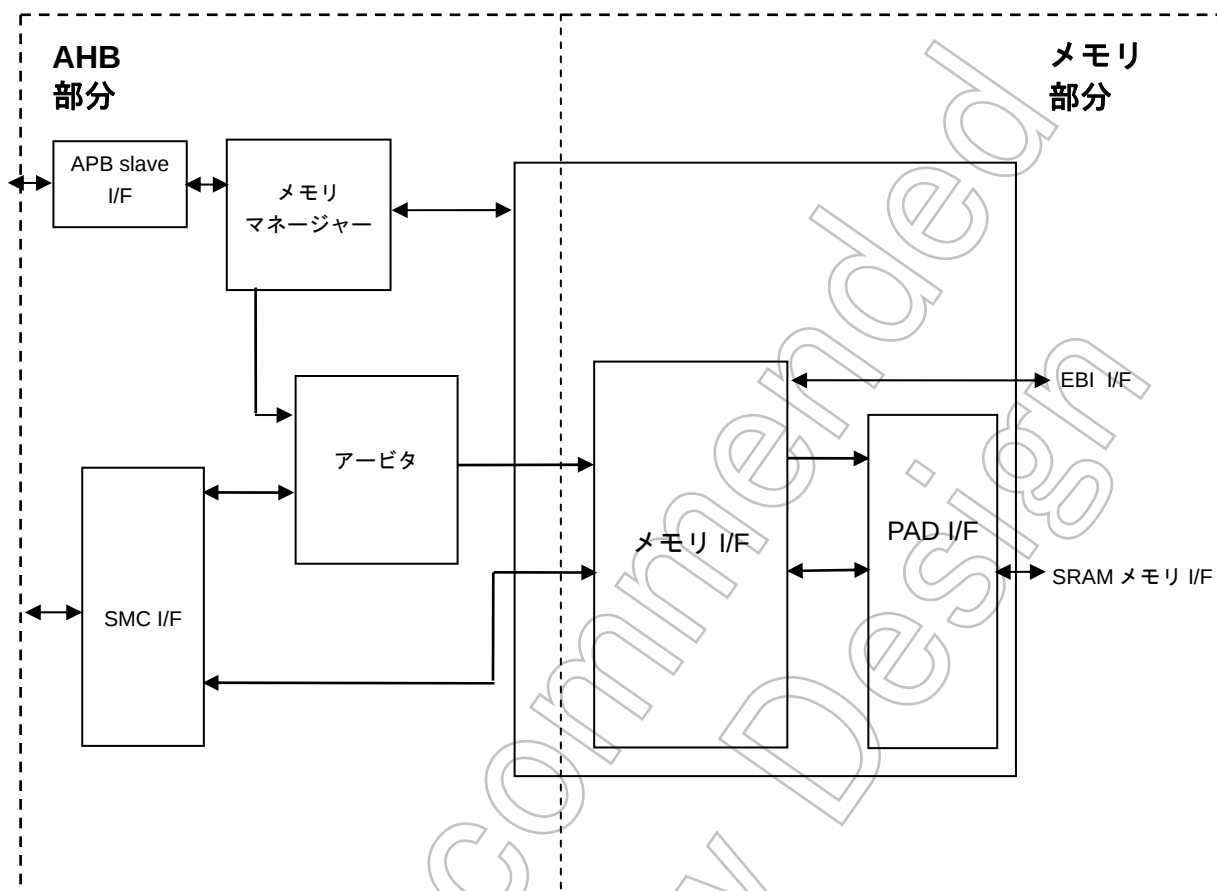


図 3.10.17 SMC ブロック図

(a) アービタ

アービタは SMC I/F とメモリマネージャからアクセスコマンドを受けて、基本的にラウンドロビン方式で、Read/Write 要求を調停します。マネージャからの要求が最優先で処理されます。

(b) メモリマネージャ

タイミングレジスタの更新、メモリに対してのコマンドを制御しています。

(3) SMC Function 動作

(a) APB slave I/F

APB slave I/F はリードとライト 1state wait を追加

下記の場合は 1state 以上の wait が生成される：

例外(outstanding) direct コマンド

メモリコマンド要求があつて、ただし、前のメモリコマンド実行は未完了。

(b) フォーマット

1. ハザード処理

同一のバスマスタ単体が、外部メモリをアクセスする場合、実際のメモリアクセス順序は命令の順序通りに実行されます。

ただし、複数のバスマスタが、外部メモリアクセスする場合、READ と WRITE は各々別々の Buffer に一旦蓄積され、優先順位回路によって実行される為、READ と WRITE の実行順序が入れ替わることがあります。

十分な時間を取るか、先に実行したアクセスが完了したことを確認するか、または、共用するメモリは内蔵 RAM を利用するなどの管理をしてください。

2. SRAM メモリをアクセス

- Standard SRAM access
- Memory address shifting
- Memory burst alignment

非同期ページモードメモリに対応するためには、burst align 設定が必要です。

3.10.3.2章 (4) 3. smc_set_opmode_3 レジスタ説明を参照してください。

注) Page 概念のないデバイスを使用する場合は、burst align 設定の必要がありません。

Memory burst length：設定可能なメモリバースト転送の length は 4 です。

(c) メモリマネージャ動作

SMC 状態の制御およびチップ設定レジスタの更新を管理しています。

(d) メモリ I/F 動作

メモリ I/F はコマンドを発行、これらコマンドタイミングを制御します。

表 3.10.7 Static メモリのセットアップ例

base アドレス= 0xF430_1000

Register address	Write data	Description
0x0014	0x00029266	smc_set_cycles_3
0x0018	0x00000809	smc_set_opmode_3
0x0010	0x00400000	smc_direct_cmd_3

Not Recommended
for New Design

(4) MPMC0 の SMC レジスタの説明

表 3.10.8 MPMC0 の SMC SFR 一覧

base アドレス= 0xF430_1000

Register Name	Address (base+)	Type	Reset Value	Description
Reserved	0x0000	–	Undefined	Read as undefined. Write as zero.
Reserved	0x0004	–	Undefined	Read as undefined. Write as zero.
Reserved	0x0008	–	–	Write prohibited
Reserved	0x000C	–	–	Write prohibited
smc_direct_cmd_3	0x0010	WO	–	SMC Direct Command Register
smc_set_cycles_3	0x0014	WO	–	SMC Set Cycles Register
smc_set_opmode_3	0x0018	WO	–	SMC Set Opmode Register
Reserved	0x0020	–	Undefined	Read as undefined. Write as zero.
smc_sram_cycles0_0_3	0x0100	RO	0x0002B3CC	SMC SRAM Cycles Registers <0-1>
smc_sram_cycles0_1_3	0x0120			
Reserved	0x0140	RO	Undefined	Read as undefined.
Reserved	0x0160	RO	Undefined	Read as undefined.
smc_opmode0_0_3	0x0104	RO	0x20E00802	SMC Opmode Registers <0-1>
smc_opmode0_1_3	0x0124		0x60E00802	
Reserved	0x0144	RO	Undefined	Read as undefined.
Reserved	0x0164	RO	Undefined	Read as undefined.
Reserved	0x0200	–	Undefined	Read as undefined. Write as zero.
Reserved	0x0204	–	Undefined	Read as undefined. Write as zero.
Reserved	0x0E00	–	Undefined	Read as undefined. Write as zero.
Reserved	0x0E04	–	Undefined	Read as undefined. Write as zero.
Reserved	0x0E08	–	Undefined	Read as undefined. Write as zero.
Reserved	0x0FE0-0x0FEC	–	Undefined	Read as undefined. Write as zero.
Reserved	0x0FF0-0x0FFC	–	Undefined	Read as undefined. Write as zero.

注) APB は single-word 32-bit アクセスのみをサポート、レジスタのリード/ライトは single-word 32-bit で実行してください。

1. smc_direct_cmd_3 (SMC Direct Command Register)

Address = (0xF430_1000) + (0x0010)

Bit	Bit Symbol	Type	Reset Value	Description
[31:26]	–	–	Undefined	Read as undefined. Write as zero.
[25:23]	chip_select	WO	–	CS 選択 : 0y000 = CS0 0y001 = CS1 0y010-0y111 = reserved
[22:21]	cmd_type	WO	–	カレントコマンド: 0y00 = Reserved 0y01 = Reserved 0y10 = UpdateRegs 0y11 = Reserved
[20:0]	–	–	Undefined	Reserved

(個別説明)

a. <chip_select>

CS 選択を選択します。

0y000 = CS0

0y001 = CS1

0y010-0y111 = Reserved

b. <cmd_type>

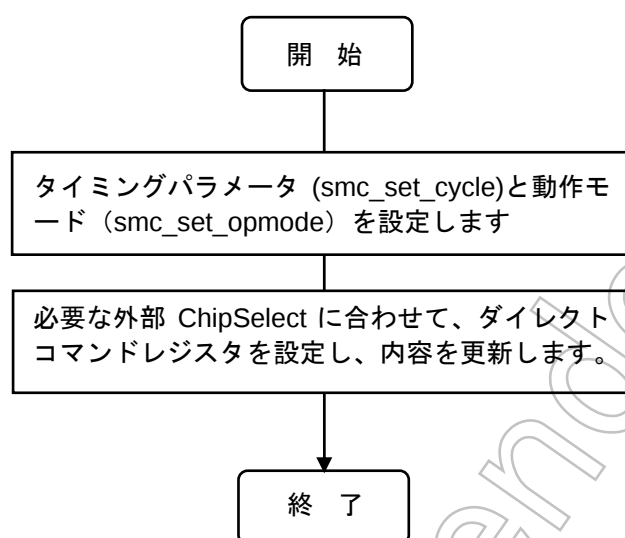
カレントコマンドを選択します。

0y00 = Reserved

0y01 = Reserved

0y10 = UpdateRegs

0y11 = Reserved



2. smc_set_cycles_3 (SMC Set Cycles Register)

Address = (0xF430_1000) + (0x0014)

Bit	Bit Symbol	Type	Reset Value	Description
[31:23]	–	–	Undefined	Read as undefined. Write as zero.
[22:20]	Reserved	–	Undefined	Read as undefined. Write as zero.
[19:17]	Set_t5	WO	–	t _{TR} の設定値(保持レジスタ) 0y000 ~ 0y111
[16:14]	Set_t4	WO	–	t _{PC} の設定値(保持レジスタ) 0y000 ~ 0y111
[13:11]	Set_t3	WO	–	t _{WP} の設定値(保持レジスタ) 0y000 ~ 0y111
[10:8]	Set_t2	WO	–	t _{CEOE} の設定値(保持レジスタ) 0y000 ~ 0y111
[7:4]	Set_t1	WO	–	t _{WC} の設定値(保持レジスタ) 0y0000 ~ 0y1111
[3:0]	Set_t0	WO	–	t _{RC} の設定値(保持レジスタ) 0y0000 ~ 0y1111

このレジスタは、StaticMemory のアクセスサイクルを調整するレジスタです。メモリの要求する A.C.に合わせて設定してください。尚、外部端子による Wait 信号との併用の場合、本設定と外部 Wait 信号を両方を満足する形でサイクルが決定します。

また、外部 Wait 信号は、同期モードでのみ有効となり、非同期モードでは有効となりませんので注意してください。

このレジスタは保持レジスタであり、設定値を有効にするため、下記の動作によりこのレジスタの設定値がメモリマネージャの configuration レジスタに update されて、有効になります。

- smc_direct_cmd レジスタでは UpdateRegs を実行します。

(個別説明)

- <Set_t5>
t_{TR} の設定値(保持レジスタ)を設定します。
0y000 ~ 0y111
- <Set_t4>
t_{PC} の設定値(保持レジスタ)を設定します。
0y000 ~ 0y111
- <Set_t3>
t_{WP} の設定値(保持レジスタ)を設定します。
0y000 ~ 0y111
- <Set_t2>
t_{CEOE} の設定値(保持レジスタ)を設定します。
0y000 ~ 0y111

- e. <Set_t1>
t_{wc} の設定値(保持レジスタ)を設定します。
0y000 ~ 0y111
- f. <Set_t0>
t_{rc} の設定値(保持レジスタ)を設定します。
0y000 ~ 0y111

タイミング設定例

レジスタ設定例: SMC Set Cycles Register = 0x0002B1C3

レジスタ設定値	t _{TR}	t _{PC}	t _{WP}	t _{CEOE}	T _{wc}	t _{RC}
0x0002B1C3	—	—	—	1	—	3

— : don't care

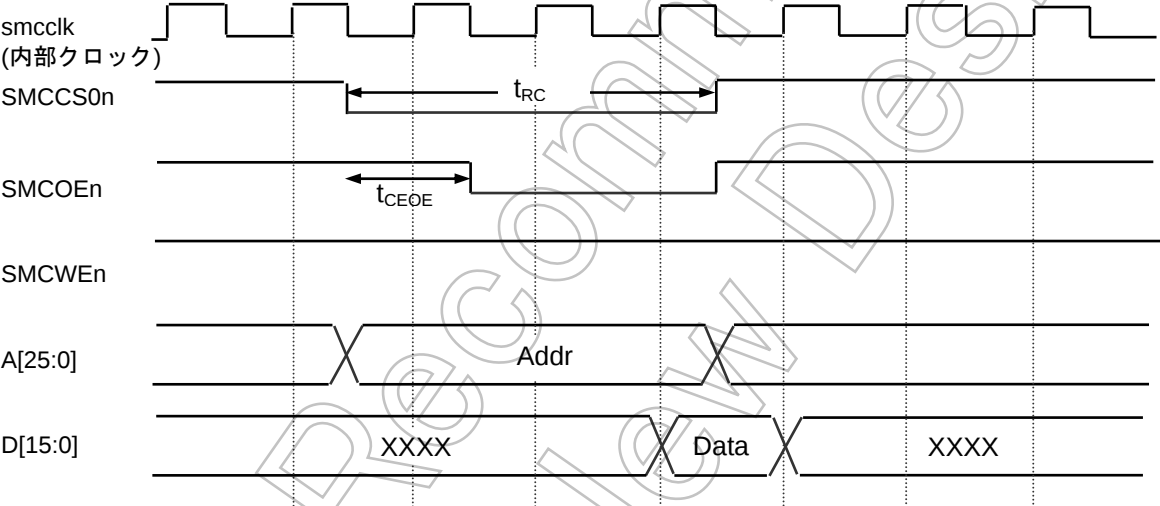
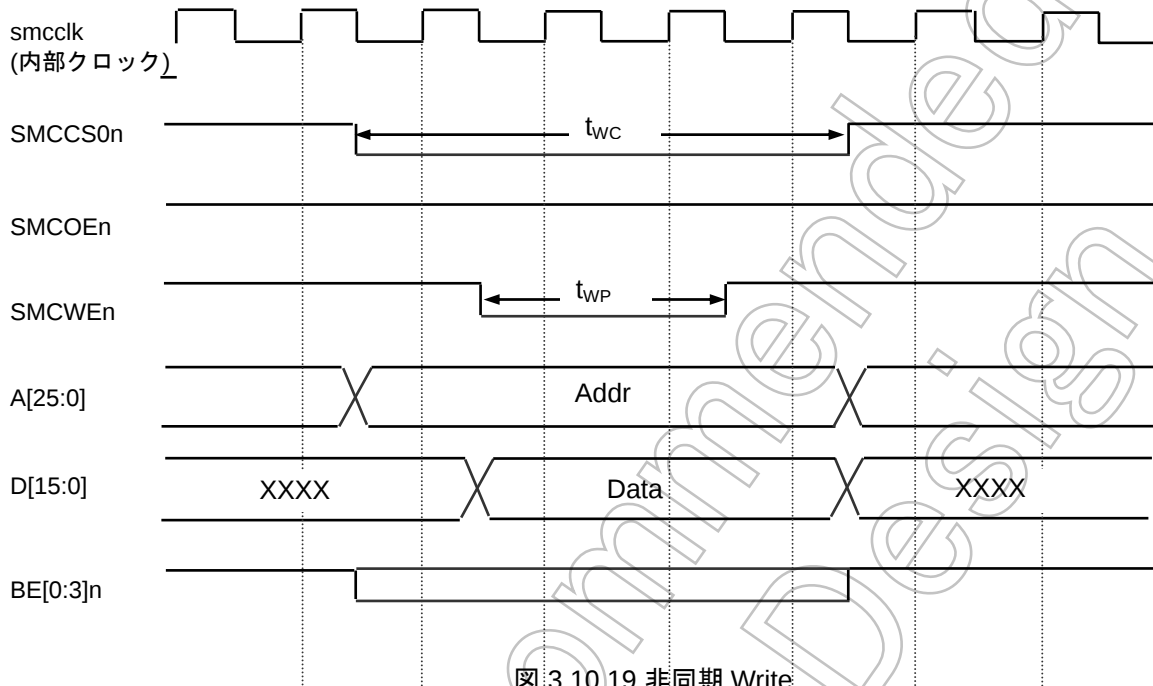


図 3.10.18 非同期 Read

レジスタ設定例: SMC Set Cycles Register = 0x0002934C

レジスタ設定値	t_{TR}	t_{PC}	t_{WP}	t_{CEOE}	T_{WC}	t_{RC}
0x0002934C	—	—	2	—	4	—

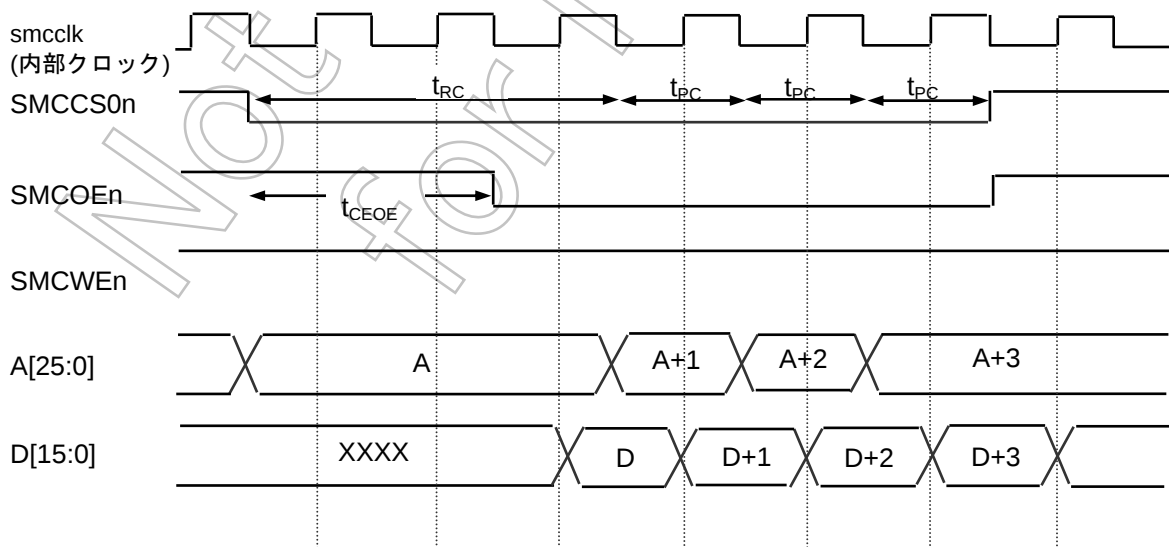
— : don't care



レジスタ設定例: SMC Set Cycles Register = 0x000272C3

レジスタ設定値	t_{TR}	t_{PC}	t_{WP}	t_{CEOE}	T_{WC}	t_{RC}
0x000272C3	—	1	—	2	—	3

— : don't care



レジスタ設定例: SMC Set Cycles Register = 0x00029143

レジスタ設定値	t_{TR}	t_{PC}	t_{WP}	t_{CEOE}	T_{WC}	t_{RC}
0x00029143	1	—	2	1	4	3

— : don't care

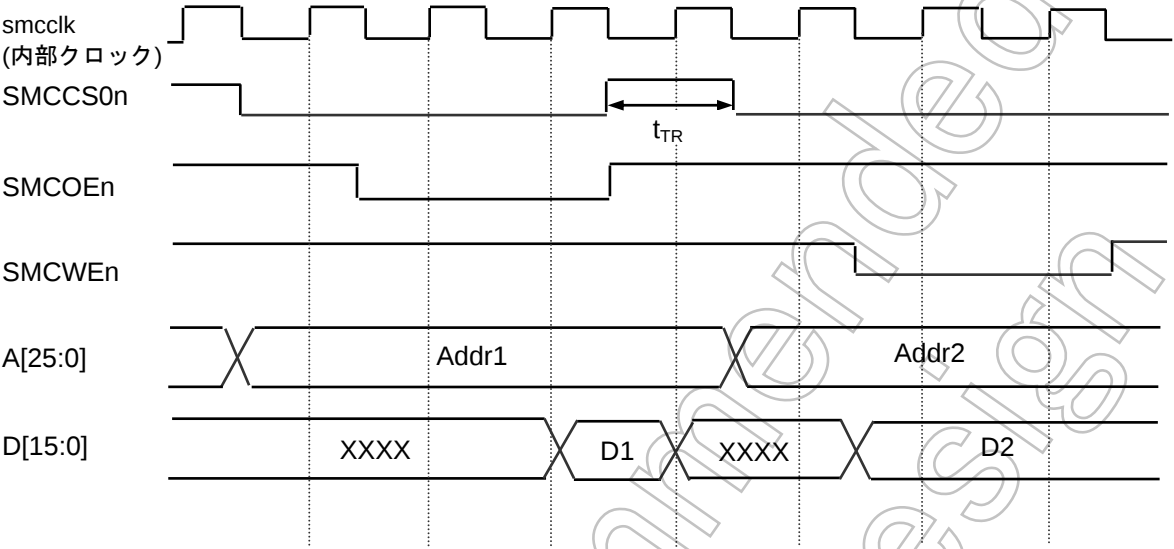


図 3.10.21 非同期 Read 後に非同期 Write

3. smc_set_opmode_3 (SMC Set Opmode Register)

Address = (0xF430_1000) + (0x0018)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined. Write as zero.
[15:13]	set_burst_align	WO	–	メモリバースト境界の分割設定: 0y000 = 任意アドレスの境界を越えられる。 0y001 = 32 ビートバースト境界で分割 0y010 = 64 ビートバースト境界で分割 0y011 = 128 ビートバースト境界で分割 0y100 = 256 ビートバースト境界で分割 other = Reserved
[12]	set_bls	WO	–	Byte Enable(SMCBE0-3)タイミング設定: 0y0 = SMCCSn タイミング 0y1 = SMCWEn タイミング
[11]	Reserved	WO	–	Write as zero.
[10]	-	–	Undefined	Read as undefined. Write as zero.
[9:7]	set_wr_bl	WO	–	ライトのバースト長: 0y000 = 1 ビート 0y001 = 4 ビート other = Reserved
[6]	set_wr_sync	WO	–	ライト同期方式の設定: 0y0 = 非同期ライトモード 0y1 = Reserved
[5:3]	set_rd_bl	WO	–	リードのバースト長: 0y000 = 1 ビート 0y001 = 4 ビート other = Reserved
[2]	set_rd_sync	WO	–	リード同期方式設定: 0y0 = 非同期リードモード 0y1 = Reserved
[1:0]	set_mw	WO	–	メモリデータバス幅設定値: 0y00 = Reserved 0y01 = 16bits 0y10 = Reserved 0y11 = Reserved

このレジスタは保持レジスタであり、設定値を有効にするため、下記の動作によりこのレジスタの設定値がメモリマネージャの configuration レジスタに update されて、有効になります。

- smc_direct_cmd レジスタで UpdateRegs を実行します。

(個別説明)

- a. < set_burst_align >
メモリバースト境界の分割設定:
0y000 = 任意アドレスの境界を越えられる。
0y001 = 32 ビートバースト境界で分割
0y010 = 64 ビートバースト境界で分割
0y011 = 128 ビートバースト境界で分割
0y100 = 256 ビートバースト境界で分割
other = Reserved
- b. < set_bls >
Byte Enable(SMCBE0-3)タイミング設定:
0y0 = SMCCSn タイミング
0y1 = SMCWEn タイミング
- c. < set_wr_bl >
ライトのバースト長:
0y000 = 1 ビート
0y001 = 4 ビート
other = Reserved
- d. < set_wr_sync >
ライト同期方式の設定:
0y0 = 非同期ライトモード
0y1 = Reserved
- e. < set_rd_bl >
リードのバースト長:
0y000 = 1 ビート
0y001 = 4 ビート
other = Reserved
- f. < set_rd_sync >
リード同期方式設定:
0y0 = 非同期リードモード
0y1 = Reserved
- g. < set_mw >
メモリデータバス幅設定値:
0y00 = Reserved
0y01 = 16bits
0y10 = Reserved
0y11 = Reserved

4. smc_sram_cycles0_0_3 (SMC SRAM Cycles Registers 0 <0>)

Address = (0xF430_1000) + (0x0100)

Bit	Bit Symbol	Type	Reset Value	Description
[31:20]	–	–	Undefined	Read as undefined.
[19:17]	t_tr	RO	0y001	SRAM chip 設定のターンアラウンド時間 0y000 ~ 0y111
[16:14]	t_pc	RO	0y010	ページサイクル時間 : 0y000 ~ 0y111
[13:11]	t_wp	RO	0y110	smc_we_n_0 の delay 時間 : 0y000 ~ 0y111
[10:8]	t_ceoe	RO	0y011	smc_oe_n_0 の delay 時間 : 0y000 ~ 0y111
[7:4]	t_wc	RO	0y1100	ライトサイクル時間 : 0y0000 ~ 0y1111
[3:0]	t_rc	RO	0y1100	リードサイクル時間 : 0y0000 ~ 0y1111

(個別説明)

- a. <t_tr>
SRAM chip 設定のターンアラウンド時間です。
0y000 ~ 0y111
- b. <t_pc>
ページサイクル時間です。
0y000 ~ 0y111
- c. <t_wp>
smc_we_n_0 の delay 時間です。
0y000 ~ 0y111
- d. <t_ceoe>
smc_oe_n_0 の delay 時間です。
0y000 ~ 0y111
- e. <t_wc>
ライトサイクル時間です。
0y0000 ~ 0y1111
- f. <t_rc>
リードサイクル時間です。
0y0000 ~ 0y1111

- smc_sram_cycles0_x_3 (SMC SRAM Cycles Registers 0 <x>) (x = 0~1)

上記のレジスタ群について、構造および説明はsmc_sram_cycles0_0_3 と同様のため、smc_sram_cycles0_0_3 の説明を参照してください。またレジスタ名およびアドレスは表 3.10.8 MPMC0 のSMC SFR一覧を参照してください。

5. smc_opmode0_0_3 (SMC Opmode Registers 0 <0>)

Address = (0xF430_1000) + (0x0104)

Bit	Bit Symbol	Type	Reset Value	Description
[31:24]	Reserved	RO	0x20	0x20 がリードされます。
[23:16]	Reserved	RO	0xE0	0xE0 がリードされます。
[15:13]	burst_align	RO	0y000	メモリバースト境界で分割の設定値： 0y000 = 任意アドレスの境界を越えられる。 0y001 = 32 ビートバースト境界で分割 0y010 = 64 ビートバースト境界で分割 0y011 = 128 ビートバースト境界で分割 0y100 = 256 ビートバースト境界で分割 other = Reserved
[12]	bls	RO	0y0	bls タイミング設定： 0y0 = chip select 0y1 = Reserved
[11]	Reserved	RO	0y1	
[10]	–	–	Undefined	Read as undefined.
[9:7]	wr_bl	RO	0y000	ライト用メモリのバースト長： 0y000 = 1 ビート 0y001 = 4 ビート other = Reserved
[6]	wr_sync	RO	0y0	メモリ動作モード： 0y0 = ライト非同期動作 0y1 = Reserved
[5:3]	rd_bl	RO	0y000	リード用メモリのバースト長： 0y000 = 1 ビート 0y001 = 4 ビート other = Reserved
[2]	rd_sync	RO	0y0	メモリ動作モード： 0y0 = リード非同期動作 0y1 = Reserved
[1:0]	mw	RO	0y10	メモリデータバス幅設定値： 0y00 = Reserved 0y01 = 16bits 0y10 = Reserved 0y11 = Reserved

6. smc_opmode0_1_3 (SMC Opmode Registers 0 <1>)

Address = (0xF430_1000) + (0x0124)

Bit	Bit Symbol	Type	Reset Value	Description
[31:24]	Reserved	RO	0x60	0x60 がリードされます。
[23:16]	Reserved	RO	0xE0	0xE0 がリードされます。
[15:13]	burst_align	RO	0y000	メモリバースト境界で分割の設定値： 0y000 = 任意アドレスの境界を越えられる。 0y001 = 32 ビートバースト境界で分割 0y010 = 64 ビートバースト境界で分割 0y011 = 128 ビートバースト境界で分割 0y100 = 256 ビートバースト境界で分割 other = Reserved
[12]	bls	RO	0y0	bls タイミング設定： 0y0 = chip select 0y1 = Reserved
[11]	Reserved	RO	0y1	
[10]	–	–	Undefined	Read as undefined.
[9:7]	wr_bl	RO	0y000	ライト用メモリのバースト長： 0y000 = 1 ビート 0y001 = 4 ビート other = Reserved
[6]	wr_sync	RO	0y0	メモリ動作モード： 0y0 = ライト非同期動作 0y1 = Reserved
[5:3]	rd_bl	RO	0y000	リード用メモリのバースト長： 0y000 = 1 ビート 0y001 = 4 ビート other = Reserved
[2]	rd_sync	RO	0y0	メモリ動作モード： 0y0 = リード非同期動作 0y1 = Reserved
[1:0]	mw	RO	0y10	メモリデータバス幅設定値： 0y00 = Reserved 0y01 = 16bits 0y10 = Reserved 0y11 = Reserved

(個別説明)

a. <burst_align>

メモリバースト境界で分割の設定値：

0y000 = 任意アドレスの境界を越えられる。

0y001 = 32 ビートバースト境界で分割

0y010 = 64 ビートバースト境界で分割

0y011 = 128 ビートバースト境界で分割

0y100 = 256 ビートバースト境界で分割

Other = Reserved

- b. <bls>
bls タイミング設定 :
0y0 = chip select
0y1 = Reserved
- c. <wr_bl>
ライト用メモリのバースト長 :
0y000 = 1 ビート
0y001 = 4 ビート
Other = Reserved
- d. <wr_sync>
メモリ動作モード :
0y0 = ライト非同期動作
0y1 = Reserved
- e. <rd_bl>
リード用メモリのバースト長 :
0y000 = 1 ビート
0y001 = 4 ビート
Other = Reserved
- f. <rd_sync>
メモリ動作モード :
0y0 = リード非同期動作
0y1 = Reserved
- g. <mw>
レジスタのリセット値は設定条件に依存します。ブートのため、CS0 のメモリデータバスの幅を設定できます。

3.10.4 MPMC1 の概要

MPMC1 には、SDRAM を制御する、DMC(Dynamic Memory Controller)と、NOR Flash、SRAM を制御する SMC(Static Memory Controller)が準備されています。

DMC(Dynamic Memory Controller)の特長

- (a) 16-bit DDR SDRAM に対応(メモリの I/O 電源電圧が LVCMOS タイプのみ対応)
- (b) 1 チャンネルのチップセレクトをサポート
- (c) SDRAM の各タイミングに対応するために、クロック単位での調整機能をサポート

SMC(Static Memory Controller)の特長

- (a) 非同期、16-bit の SRAM , NOR flash(セパレートバスのみ、マルチプレクスバスはサポートしません) メモリをサポート
- (b) 2 チャンネルチップセレクト信号をサポート
- (c) チップセレクト毎に、サイクルの調整およびメモリデータバス幅をプログラムで設定可能

3.10.5 機能概要(MPMC 1)

図 3.10.22は、MPMC1 の簡単なブロック図を示しています。

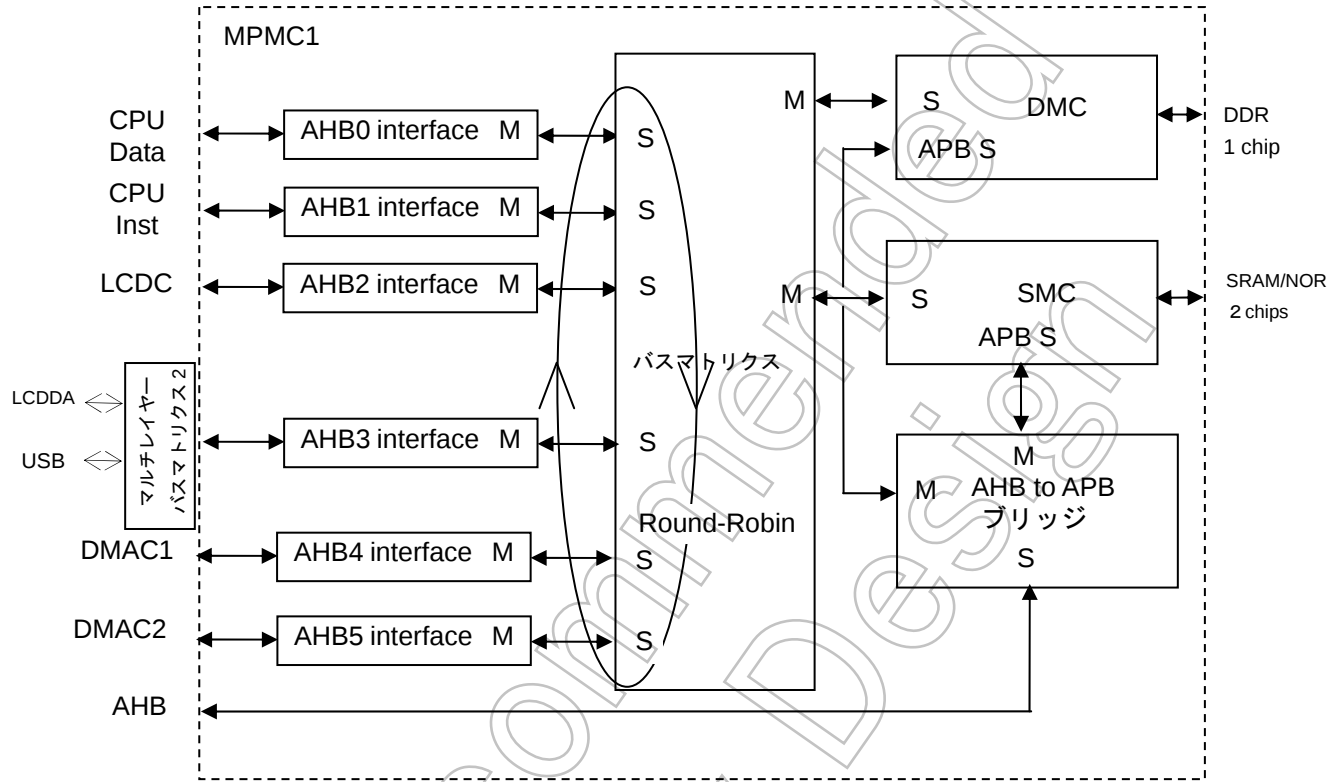
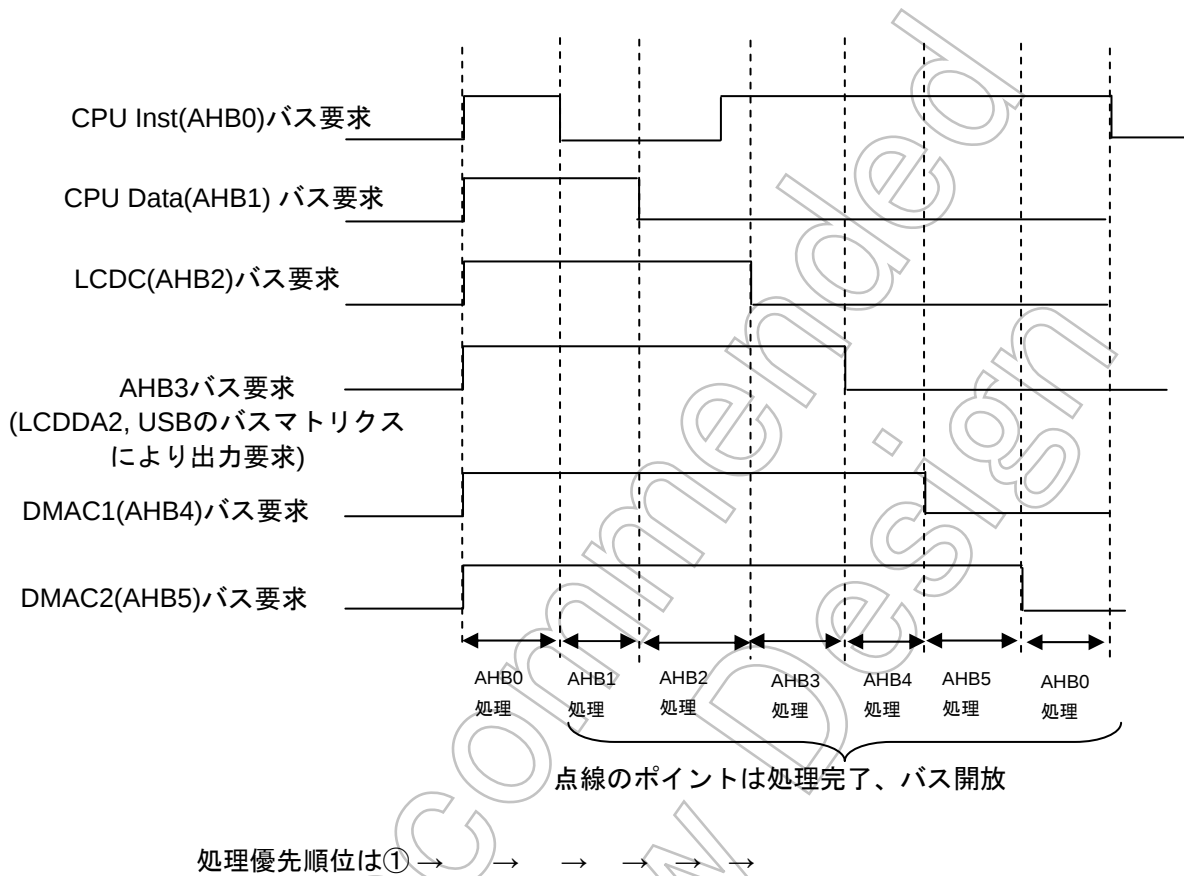


図 3.10.22 MPMC1 ブロック図

(a) バスマトリクス

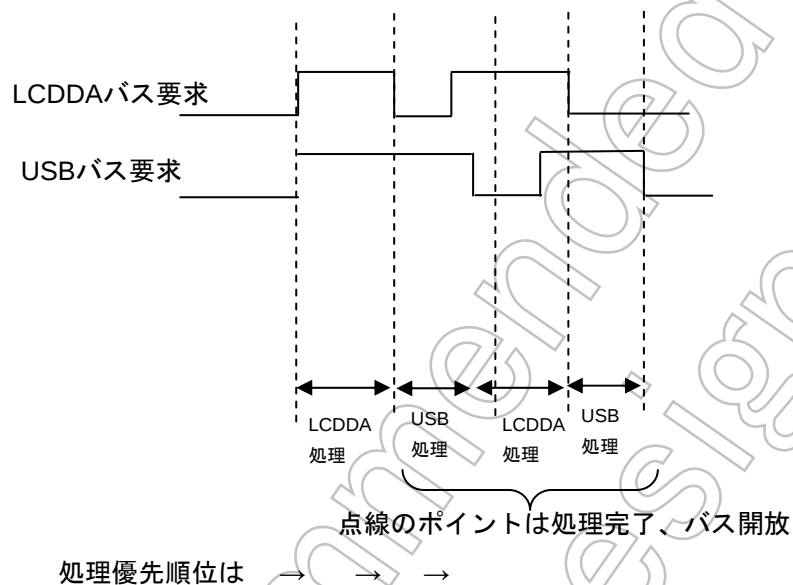
1. AHB0, AHB1, AHB2, AHB3, AHB4, AHB5 のバスマトリクスは、Round-Robin(ラウンドロビン: 巡回的並列処理)アービタ方式をサポートします。下記の図にバス要求優先順位例を示します。



2. LCDDA,USB のバスマトリクス 2 の優先順位について、バス要求が早い方が優先的に処理します。バス要求が同じタイミング時、ハードウェアの優先順位で処理を決めます。

ハードウェアの優先順位は LCDDA(高い)→ USB(低い)

下記の図にバス要求優先順位例を示します。



(b) クロック種類

クロックギア回路にてクロックを制御しています。

1. Dynamic メモリクロック : HCLK クロックを接続
2. Static メモリクロック : HCLK クロックまたは 1/2_HCLK クロックを選択

3.10.5.1 DMC (Dynamic Memory Controller)

(1) DMC ブロック図

以下に、DMC 回路のブロック図を示します。

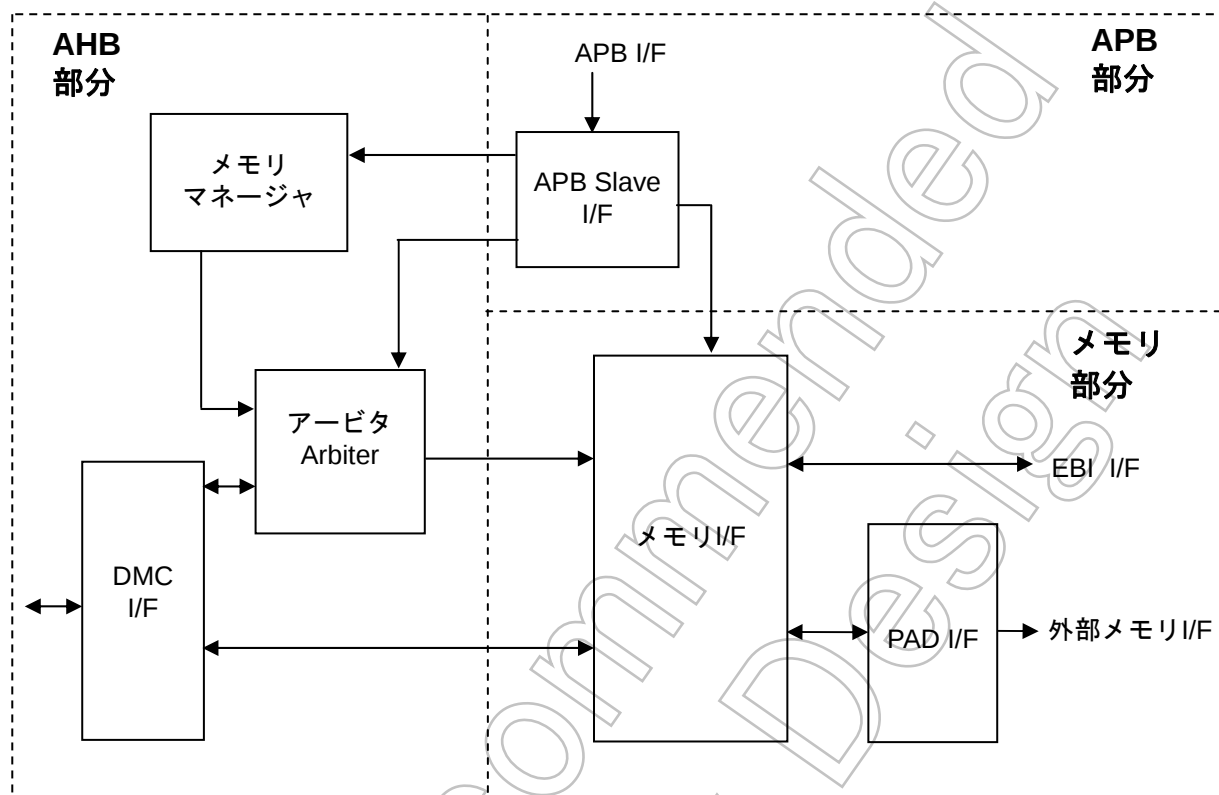


図 3.10.23 DMC ブロック図

(f) アービタ

アービタは DMC I/F とメモリマネージャからアクセスコマンドを受けて、アクセスの調停後、優先度が一番高いコマンドをメモリ I/F に渡します。データはメモリ I/F から DMC I/F に直接接続されています。

(g) メモリマネージャ

メモリマネージャは、DMC 回路の状態を監視し、制御を行います。

(2) DMC Function 動作

(a) アービタ動作

1. リードとライトアクセスを調停
2. リードアクセスでは、QoS 動作(Quality of service)をサポート
3. ハザード処理

同一のバスマスタ単体が、外部メモリをアクセスする場合、実際のメモリアクセス順序は命令の順序通りに実行されます。

ただし、複数のバスマスタが、外部メモリアクセスする場合、READ と WRITE は各々別々の Buffer に一旦蓄積され、優先順位回路によって実行される為、READ と WRITE の実行順序が入れ替わることがあります。

十分な時間を取るか、先に実行したアクセスが完了したことを確認するか、または、共用するメモリは内蔵 RAM を利用するなどの管理をしてください。

4. ステートマシンを監視し、適正なパイプラインのエントリを選択

(b) メモリマネージャ動作

1. DMC 回路の監視と制御
2. ダイレクトコマンドの発行:
 - NOP
 - Prechargeall
 - Autorefresh
 - ModeReg
 - 拡張 modereg.

3. Auto-refresh 機能のサポート

15ビットのカウンタでオートリフレッシュのサイクルを設定できます。

(c) メモリ I/F 動作

用途に応じた下記 3 種類の FIFO を内蔵しています。

1. コマンド FIFO : 2word
2. データリード FIFO : 10word
3. データライト FIFO : 10word

※リードおよびライトの FIFO サイズが 10word のため、最大 8word の連続転送しかサポートしていません。(1word=32bit data)

(d) Low-power 機能

DMC は下記の 2 種類 Low-power をサポートしています。

1. dmc_memc_cmd_5 レジスタの設定により、Low-power(Self-refresh Mode)を実現できます。
2. dmc_memory_cfg_5 レジスタの設定により、メモリクロックを停止する(Clock Suspend Mode)とメモリアクセスがないときに CKE 端子を自動的に無効化する(Power Down Mode)機能をサポートしています。

注) Clock Suspend Mode 機能と Power Down Mode 機能は併用できません。

(e) QoS 機能

QoS はリードアクセスでしか使用できません。

QoS 機能とは、MPMC 内のバスマトリクスが通常管理する順序回路(RoundRobin)において、例外処理を行う為のサービス機能です。

dmc_id_x_cfg_5<qos_max>には要求されているリードの最大レイテンシーを設定します。QoS_max の Timeout が発生する時、DMC に内蔵するアービタは、該当する転送マスタのプライオリティを上げます。

また、dmc_id_x_cfg_5<qos_min>を enable に設定すると、アクセスを最優先させることが出来ます。しかし、この場合 DMC のスケジューリングアルゴリズムが制限されて、全体のメモリバンド幅が影響されることがあります。

Not Recommended
for New Design

表 3.10.9 DDR メモリのセットアップ例

base アドレス= 0xF431_0000

Register address	Write data	Description
0x0014	0x00000004	Set cas_latency to 2
0x0018	0x00000001	Set t_dqss to 1
0x001C	0x00000002	Sett_mrd to 2
0x0020	0x00000007	Sett_ras to 7
0x0024	0x0000000B	Set t_rc to 11
0x0028	0x00000015	Set t_rcd to 5 and schedule_rcd to 2
0x002C	0x000001F2	Set t_rfc to 18 and schedule_rfc to 15
0x0030	0x00000015	Set t_rp to 5 and schedule_rp to 2
0x0034	0x00000002	Set t_rrd to 2
0x0038	0x00000003	Set t_wr to 3
0x003C	0x00000002	Set t_wtr to 2
0x0040	0x00000001	Set t_xp to 1
0x0044	0x0000000A	Set t_xsr to 10
0x0048	0x00000014	Set t_esr to 20
0x000C	0x00010009	Set memory configuration
0x0010	0x00000640	Set auto refresh period to be every 1600 DMCSCLK periods
0x0200	0x000000FF	Set chip select for chip 0 to be 0x00XXXXXX, rbc configuration
0x0008	0x000C0000	Carry out chip 0 Nop command
0x0008	0x00000000	Carry out chip 0 Prechargeall command
0x0008	0x00090000	Extended mode register setup
0x0008	0x00080122	Mode register setup
0x0008	0x00000000	Precharge all
0x0008	0x00040000	Carry out chip 0 Autorefresh command
0x0008	0x00040000	Carry out chip 0 Autorefresh command
0x0008	0x00080032	Carry out chip 0 Mode Reg command 0x32 mapped to low add bits
0x0004	0x00000000	Change DMC state to ready

(3) MPMC1 の DMC レジスタ説明

表 3.10.10 SFR 一覧

base アドレス= 0xF431_0000

Register Name	Address (base+)	Type	Reset Value	Description
dmc_memc_status_5	0x0000	RO	0x00000390	DMC Memory Controller Status Register
dmc_memc_cmd_5	0x0004	WO	–	DMC Memory Controller Command Register
dmc_direct_cmd_5	0x0008	WO	–	DMC Direct Command Register
dmc_memory_cfg_5	0x000C	R/W	0x00010020	DMC Memory Configuration Register
dmc_refresh_prd_5	0x0010	R/W	0x00000A60	DMC Refresh Period Register
dmc_cas_latency_5	0x0014	R/W	0x00000006	DMC CAS Latency Register
dmc_t_dqss_5	0x0018	R/W	0x00000001	DMC t_dqss Register
dmc_t_mrd_5	0x001C	R/W	0x00000002	DMC t_mrd Register
dmc_t_ras_5	0x0020	R/W	0x00000007	DMC t_ras Register
dmc_t_rc_5	0x0024	R/W	0x0000000B	DMC t_rc Register
dmc_t_rcd_5	0x0028	R/W	0x0000001D	DMC t_rcd Register
dmc_t_rfc_5	0x002C	R/W	0x00000212	DMC t_rfc Register
dmc_t_rp_5	0x0030	R/W	0x0000001D	DMC t_rp Register
dmc_t_rrd_5	0x0034	R/W	0x00000002	DMC t_rrd Register
dmc_t_wr_5	0x0038	R/W	0x00000003	DMC t_wr Register
dmc_t_wtr_5	0x003C	R/W	0x00000002	DMC t_wtr Register
dmc_t_xp_5	0x0040	R/W	0x00000001	DMC t_xp Register
dmc_t_xsr_5	0x0044	R/W	0x0000000A	DMC t_xsr Register
dmc_t_esr_5	0x0048	R/W	0x00000014	DMC t_esr Register
dmc_id_0_cfg_5	0x0100	R/W	0x00000000	DMC id_<0-5>_cfg Registers
dmc_id_1_cfg_5	0x0104			
dmc_id_2_cfg_5	0x0108			
dmc_id_3_cfg_5	0x010C			
dmc_id_4_cfg_5	0x0110			
dmc_id_5_cfg_5	0x0114			
dmc_chip_0_cfg_5	0x0200	R/W	0x0000FF00	DMC chip_0_cfg Registers
Reserved	0x0204	–	Undefined	Read as undefined. Write as zero.
Reserved	0x0208	–	Undefined	Read as undefined. Write as zero.
Reserved	0x020C	–	Undefined	Read as undefined. Write as zero.
Reserved	0x0300	–	Undefined	Read as undefined. Write as zero.
dmc_user_config_5	0x0304	WO	Undefined	DMC user_config Register
Reserved	0x0E00	–	Undefined	Read as undefined. Write as zero.
Reserved	0x0E04	–	Undefined	Read as undefined. Write as zero.
Reserved	0x0E08	–	Undefined	Read as undefined. Write as zero.
Reserved	0x0FE0-0x0FEC	–	Undefined	Read as undefined. Write as zero.
Reserved	0x0FF0-0x0FFC	–	Undefined	Read as undefined. Write as zero.

注) APB は single-word 32-bit アクセスのみをサポート、レジスタのリード/ライトは single-word 32-bit で実行してください。

MPMC1

レジスタ Read/Write できる dmc_memc_status_5 状態一覧 ○ : 許可 × : 禁止

Register Name	Type	Read				Write			
		dmc_memc_status_5				dmc_memc_status_5			
		Config	Ready	Paused	Low_power	Config	Ready	Paused	Low_power
dmc_memc_status_5	RO	○	○	○	○	—	—	—	—
dmc_memc_cmd_5	WO	—	—	—	—	○	○	○	○
dmc_direct_cmd_5	WO	—	—	—	—	○	×	×	×
dmc_memory_cfg_5	R/W	○	○	○	○	○	×	×	×
dmc_refresh_prd_5	R/W	○	○	○	○	○	×	×	×
dmc_cas_latency_5	R/W	○	○	○	○	○	×	×	×
dmc_t_dqss_5	R/W	○	○	○	○	○	×	×	×
dmc_t_mrd_5	R/W	○	○	○	○	○	×	×	×
dmc_t_ras_5	R/W	○	○	○	○	○	×	×	×
dmc_t_rc_5	R/W	○	○	○	○	○	×	×	×
dmc_t_rcd_5	R/W	○	○	○	○	○	×	×	×
dmc_t_rfc_5	R/W	○	○	○	○	○	×	×	×
dmc_t_rp_5	R/W	○	○	○	○	○	×	×	×
dmc_t_rrd_5	R/W	○	○	○	○	○	×	×	×
dmc_t_wr_5	R/W	○	○	○	○	○	×	×	×
dmc_t_wtr_5	R/W	○	○	○	○	○	×	×	×
dmc_t_xp_5	R/W	○	○	○	○	○	×	×	×
dmc_t_xsr_5	R/W	○	○	○	○	○	×	×	×
dmc_t_esr_5	R/W	○	○	○	○	○	×	×	×
dmc_id_0_cfg_5 dmc_id_1_cfg_5 dmc_id_2_cfg_5 dmc_id_3_cfg_5 dmc_id_4_cfg_5 dmc_id_5_cfg_5	R/W	○	○	○	○	○	×	×	○
dmc_chip_0_cfg_5	R/W	○	○	○	○	○	×	×	○
dmc_user_config_5	WO	—	—	—	—	○	○	○	○

MPMC1のレジスタについて、リセットおよびPOR状態でリード/ライトできません。

1. dmc_memc_status_5 (DMC Memory Controller Status Register)

Address = (0xF431_0000) + (0x0000)

Bit	Bit Symbol	Type	Reset Value	Description
[31:10]	–	–	Undefined	Read as undefined. Write as zero.
[9]	memory_banks	RO	0y1	DMC がサポートする最大バンク数 の設定値 : (4 バンクに固定)
[8:7]	–	–	Undefined	Read as undefined.
[6:4]	memory_ddr	RO	0y001	DMC がサポートする SDRAM 種類 : 0y000 = Reserved 0y001 = DDR SDRAM 0y011 = Reserved 0y010 = Reserved 0y1xx = Reserved
[3:2]	memory_width	RO	0y00	外部メモリバス幅: 0y00 = 16-bit 0y01 = Reserved 0y10 = Reserved 0y11 = Reserved
[1:0]	memc_status	RO	0y00	メモリコントローラの状態: 0y00 = Config 0y01 = Ready 0y10 = Paused 0y11 = Low-power

(個別説明)

a. <memory_banks>

DMC がサポートする最大バンク数 の設定値。
4 バンクに固定されています。

b. <memory_ddr>

DMC がサポートする SDRAM の種類を示します。
0y001 (DDR SDRAM)に固定されています。

c. <memory_width>

外部メモリバス幅を示します。

0y00 = 16-bit

0y01 = Reserved

0y10 = Reserved

0y11 = Reserved

d. <memc_status>

メモリコントローラの状態を示します。

0y00 = Config

0y01 = Ready

0y10 = Paused

0y11 = Low-power

2. dmc_memc_cmd_5 (DMC Memory Controller Command Register)

Address = (0xF431_0000) + (0x0004)

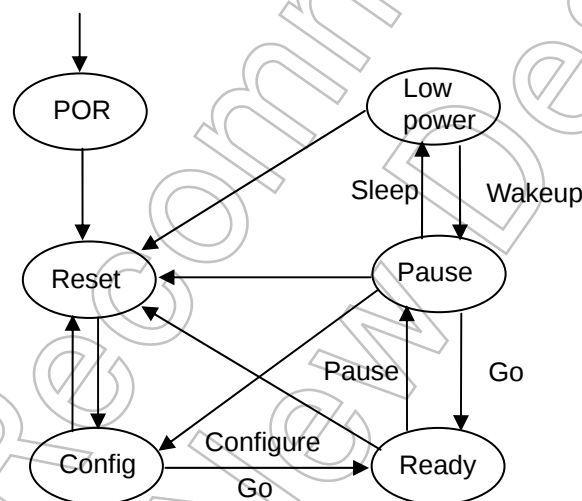
Bit	Bit Symbol	Type	Reset Value	Description
[31:3]	—	—	Undefined	Read as undefined. Write as zero.
[2:0]	memc_cmd	WO	—	メモリコントローラ状態を変更: 0y000 = Go 0y001 = Sleep 0y010 = Wakeup 0y011 = Pause 0y100 = Configure

(個別説明)

a. <memc_cmd>

このレジスタの設定により、DMC のステートマシンを変更できます。前の状態変更のコマンド実行中は、完了までを待って新規コマンドを発行します。

Low-power 実施フローは下記の図に示します。



Reset 状態から抜けると自動的に Config 状態となります。また、Pause 状態からは Config コマンドで Config 状態に遷移します。各レジスタの設定は Config 状態で行う必要があります。

Ready 状態に遷移すると、SDRAM への Read/Write が可能となります。また、Read/Write を実行することで SDRAM は IDLE 状態から ACTIVE 状態へ移行します。

Ready 状態から Pause コマンドにて Pause 状態へ遷移します。この時の SDRAM の状態は、SDRAM への前コマンドによって異なります。Read/Write 直後であれば ACTIVE 状態、AutoRefresh 直後であれば IDLE 状態となります^{注)}。

Pause 状態から Sleep コマンドにて Low power 状態に遷移すると、All Bank Precharge を実施後、CKE="L"にして SDRAM が自動的に Self-refresh 状態になります。

Low power 状態から Wakeup コマンドにて Pause 状態に遷移すると、self_refresh Exit コマンドを発行します。SDRAM は自動的に self_refersh 状態から抜け IDLE 状態に移行します。

注) SDRAM を ACTIVE 状態から IDLE 状態へ移行させたい場合は、2 種類の方法があります。dmc_direct_cmd_5<memory_cmd>0y00 = Prechargeall, 0y01 = Autorefresh のいずれかを設定してください。

3. dmc_direct_cmd_5 (DMC Direct Command Register)

外部メモリに対する各コマンド発行、および外部メモリのモードレジスタを設定します。
このレジスタで外部メモリの初期設定を行います。

$$\text{Address} = (0xF431_0000) + (0x0008)$$

Bit	Bit Symbol	Type	Reset Value	Description
[31:22]	–	–	Undefined	Read as undefined. Write as zero.
[21:20]	chip_nمبر	WO	–	常に 0y00 を設定してください。
[19:18]	memory_cmd	WO	–	要求コマンド : 0y00 = Prechargeall 0y01 = Autorefresh 0y10 = Modereg or 拡張 modereg 0y11 = NOP
[17:16]	bank_addr	WO	–	Modereg をアクセスする時、外部メモリバンクアドレスビット設定 0y00 = bank0 0y01 = bank1 0y10 = bank2 0y11 = bank3
[15:14]	–	–	Undefined	Read as undefined. Write as zero.
[13:0]	addr_13_to_0	WO	–	Modereg をアクセス時、コマンドのデータ設定 (外部メモリアドレスビットに対応)

注意：

dmc_direct_cmd_5 により DDR_SDRAM メモリの CAS レイテンシーの設定値について、CAS レイテンシー(CL)の数が SDR_SDRAM と異なります。メモリコントローラには、DDR_SDRAM メモリの CAS レイテンシー(CL)より 1 小さい値を設定してください。

例：

dmc_cas_latency_5 ← 0x00000004 (メモリコントローラの CL=2 を設定)

dmc_direct_cmd_5 ← 0x00080033 (DDR_SDRAM メモリの CL=3 を設定)

(個別説明)

a. <memory_cmd>

要求コマンドを選択します。

0y00 = Prechargeall

0y01 = Autorefresh

0y10 = Modereg or 拡張 modereg

0y11 = NOP

b. <bank_addr>

Modereg をアクセスする時の外部メモリバンクアドレスビットを設定します。

0y00 = bank0

0y01 = bank1

0y10 = bank2

0y11 = bank3

c. <addr_13_to_0>

Modereg をアクセス時のコマンドのデータを設定します。

(外部メモリアドレスビットに対応)

Not Recommended
for New Design

4. dmc_memory_cfg_5 (DMC Memory Configuration Register)

Address = (0xF431_0000) + (0x000C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:23]	–	–	Undefined	Read as undefined. Write as zero.
[22:21]	active_chips	R/W	0y00	常に 0y00 を設定してください。
[20:18]	–	–	Undefined	Read as undefined. Write as zero.
[17:15]	memory_burst	R/W	0y010	SDRAM をリード/ライトするときのバースト長設定 0y000 = Reserved 0y001 = Burst 2 0y010 = Burst 4 0y011 = Burst 8 0y100 = Burst 16 other = Reserved
[14]	stop_mem_clock	R/W	0y0	メモリクロック停止: 0y0 = Disable 0y1 = Enable
[13]	auto_power_down	R/W	0y0	SDRAM 自動 Powerdown 設定: 0y0 = Disable 0y1 = Enable
[12:7]	power_down_prd	R/W	0y000000	SDRAM 自動 Powerdown メモリクロック数設定: (Min 値 = 1) 0y000001 ~ 0y111111
[6]	ap_bit	R/W	0y0	メモリアドレスで auto-precharge ビット位置: 0y0 = address bit 10 0y1 = address bit 8
[5:3]	row_bits	R/W	0y100	row アドレスビット数: 0y000 = 11 bits 0y001 = 12 bits 0y010 = 13 bits 0y011 = 14 bits 0y100 = 15 bits 0y101 = 16 bits other = Reserved
[2:0]	column_bits	R/W	0y000	column アドレスビット数: 0y000 = 8 bits 0y001 = 9 bits 0y010 = 10 bits 0y011 = 11 bits 0y100 = 12 bits other = Reserved

(個別説明)

a. <memory_burst>

メモリをアクセスするコントローラのバースト長を設定します。

また、dmc_direct_cmd_5 で設定したメモリのバースト長と一致させてください。

b. <stop_mem_clock>

SDRAM にアクセスしていない時、クロック供給を停止します。再度 SDRAM へのアクセス要求があると、自動的にクロックを再スタートします。

注 1) 利用する SDRAM によっては、アクセスしていない期間中のクロックの停止を許可していない製品が存在いたします。本機能をご使用の際は、使用する SDRAM の SPEC を十分に確認してください。

注 2) メモリクロック停止機能と SDRAM 自動 Powerdown 機能の併用はできません。どちらか一方のみの利用が前提となります。

c. <auto_power_down>

SDRAM へのアクセス要求が無くなり、メモリコントローラのコマンド FIFO が空き状態になると、自動的に power_down_prd で設定されたクロック後に、CKE を Disable 状態に変化させ、SDRAM を Powerdown 状態に Entry することができます。再度 SDRAM へのアクセス要求があると、自動的に CKE を Enable 状態にして Powerdown から Exit します。

注) メモリクロック停止機能と SDRAM 自動 Powerdown 機能の併用はできません。どちらか一方のみの利用が前提となります。

d. <row_bits><column_bits>

row および column アドレスの設定をします。

接続可能な最大メモリサイズは、Column アドレスと row アドレスの総和で制限されます。

32bit バスの場合 : $R+C=25$ ビット (128Mbyte) までで、その 4Bank 分 (512Mbyte)

16bit バスの場合 : $R+C=26$ ビット (128Mbyte) までで、その 4Bank 分 (512Mbyte)

5. dmc_refresh_prd_5 (DMC Refresh Period Register)

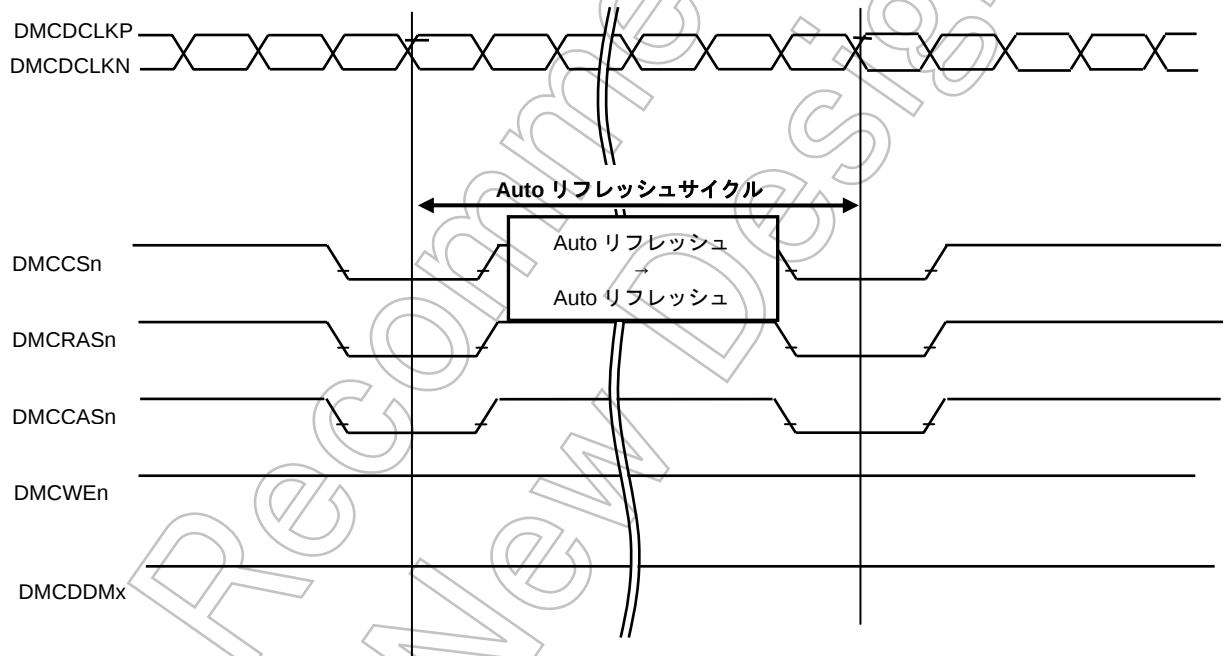
Address = (0xF431_0000) + (0x0010)

Bit	Bit Symbol	Type	Reset Value	Description
[31:15]	—	—	Undefined	Read as undefined. Write as zero.
[14:0]	refresh_prd	R/W	0x0A60	Auto refresh の周期(メモリクロック数): 0x0000 ~ 0x7FFF

(個別説明)

a. <refresh_prd>

Refresh カウンタは dmc_refresh_prd_5 レジスタで設定した値(メモリクロック数)からデクリメントして、0 になると Autorefresh コマンドを外部メモリに発行します。



6. dmc_cas_latency_5 (DMC CAS Latency Register)

Address = (0xF431_0000) + (0x0014)

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	—	—	Undefined	Read as undefined. Write as zero.
[3:1]	cas_latency	R/W	0y011	CAS レイテンシーの設定 (メモリクロック数) 0y000 ~ 0y111
[0]	cas_half_cycle	R/W	0y0	CAS レイテンシーの offset 設定 0y0 = 0 offset 0y1 = Half cycle offset

注意：

dmc_cas_latency_5により DDR_SDRAM メモリコントローラの CAS レイテンシーの設定値について、CAS レイテンシー(CL)の数が SDR_SDRAM と異なります。メモリコントローラには、DDR_SDRAM メモリの CAS レイテンシー(CL)より 1 小さい値を設定してください。

例：

dmc_cas_latency_5 ← 0x00000004 (メモリコントローラの CL=2 を設定)

dmc_direct_cmd_5 ← 0x00080033 (DDR SDRAM メモリの CL=3 を設定)

(個別説明)

a. <cas_latency>

CAS レイテンシーを設定します。(メモリクロック数)

0y000 ~ 0y111

b. <cas_half_cycle>

CAS レイテンシの offset 設定します。

0y0 = 0 offset

0y1 = Half cycle offset

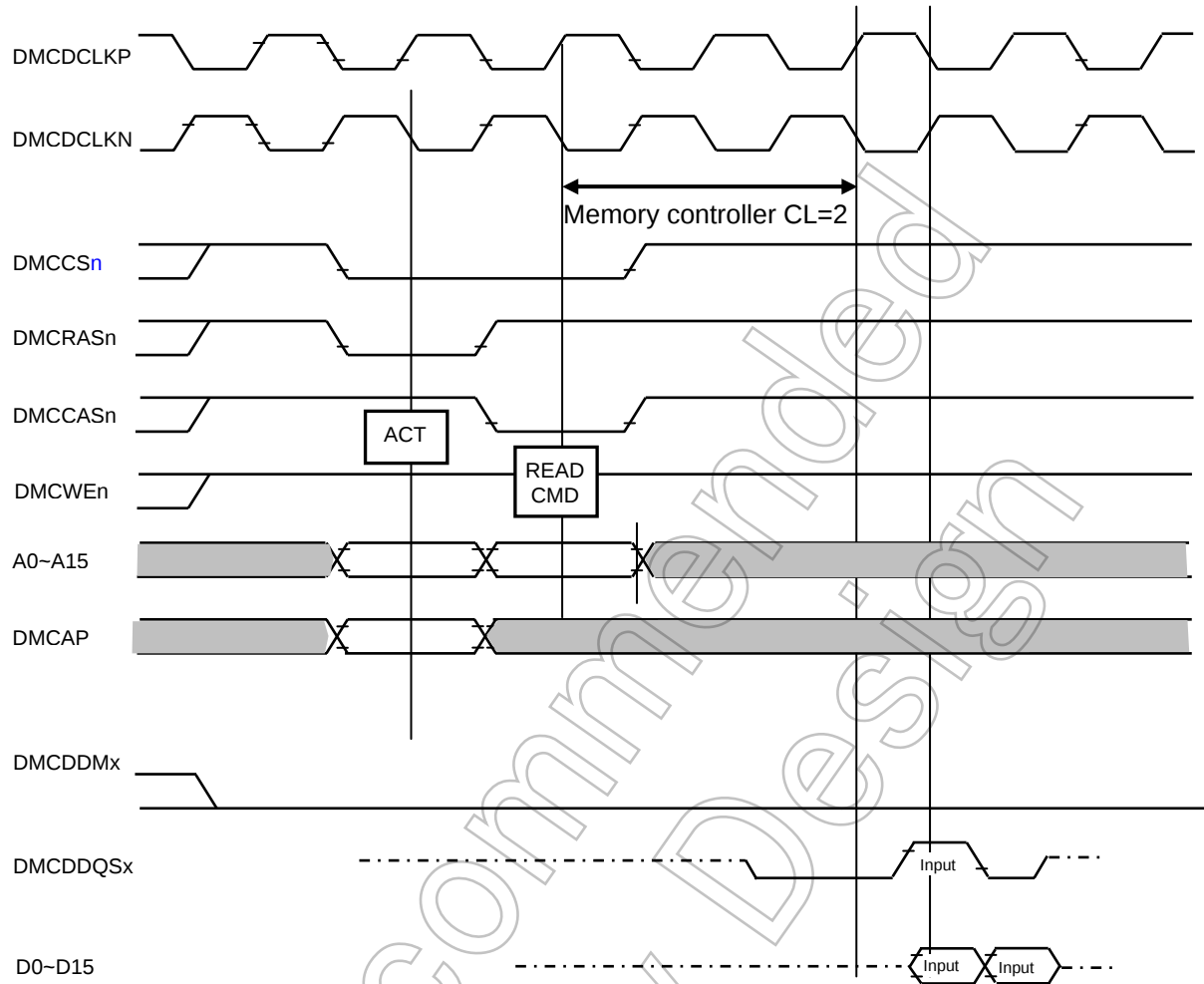


図 3.10.24 CAS レイテンシー設定例 (CL=2)

7. dmc_t_dqss_5 (DMC t_dqss Register)

Address = (0xF431_0000) + (0x0018)

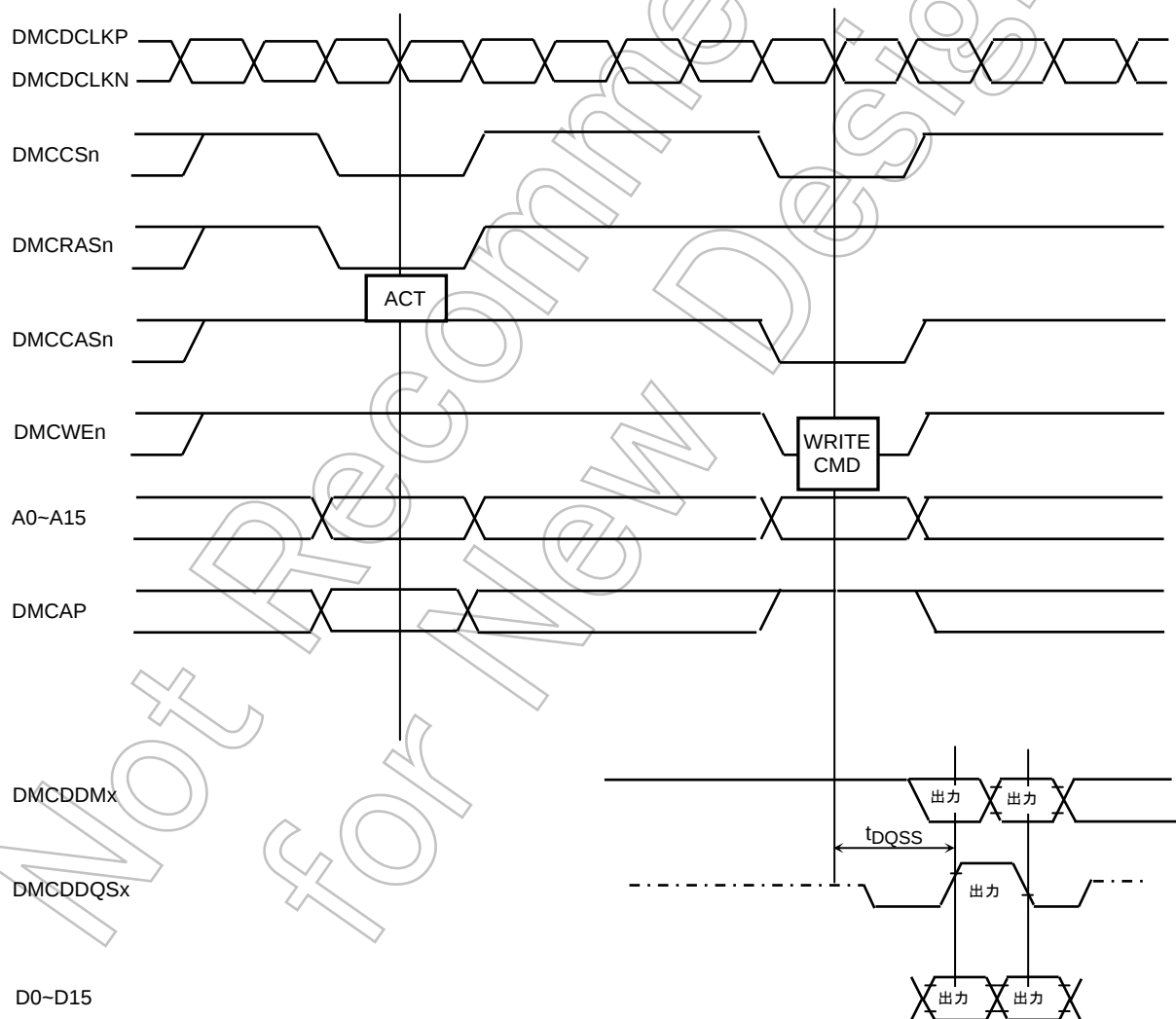
Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	—	—	Undefined	Read as undefined. Write as zero.
[1:0]	t_dqss	R/W	0y01	DQS の設定 (メモリクロック数) 0y00 ~ 0y11

(個別説明)

a. <t_dqss>

DQS を設定します。(メモリクロック数)

0y00 ~ 0y11



8. dmc_t_mrd_5 (DMC t_mrd Register)

Address = (0xF431_0000) + (0x001C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:7]	—	—	Undefined	Read as undefined. Write as zero.
[6:0]	t_mrd	R/W	0y0000010	モードレジスタコマンド時間 (メモリクロック数) 0x00 ~ 0x7F

(個別説明)

a. <t_mrd>

ダイレクトコマンドレジスタ(dmc_direct_cmd_5<addr_13_to_0>)にて設定するモードレジスタコマンド時間から、他のすべてのコマンドまでの時間を設定します。

(メモリクロック数)

0x00 ~ 0x7F

他の AC 設定または他の動作との兼ね合いにより、設定されたサイクル数よりも長いサイクルで動作する場合があります。よって、本設定は最低クロック数を設定します。

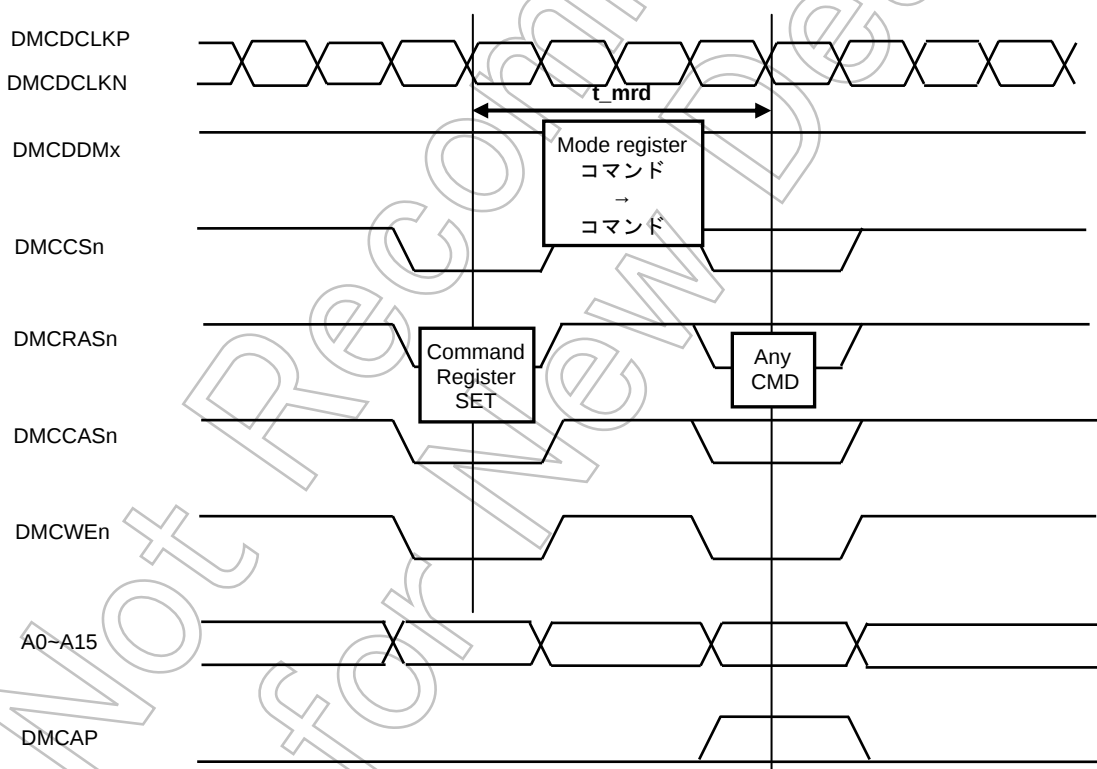


図 3.10.25 モードレジスタライトから他コマンドまでの時間

9. dmc_t_ras_5 (DMC t_ras Register)

Address = (0xF431_0000) + (0x0020)

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	—	—	Undefined	Read as undefined. Write as zero.
[3:0]	t_ras	R/W	0x7	RAS から Precharge の時間 (メモリクロック数) 0x0 ~ 0xF

(個別説明)

a. <t_ras>

RAS から Precharge の時間を設定します。(メモリクロック数)

0x0 ~ 0xF

他の AC 設定または他の動作との兼ね合いにより、設定されたサイクル数よりも長いサイクルで動作する場合があります。よって、本設定は最低クロック数を設定します。

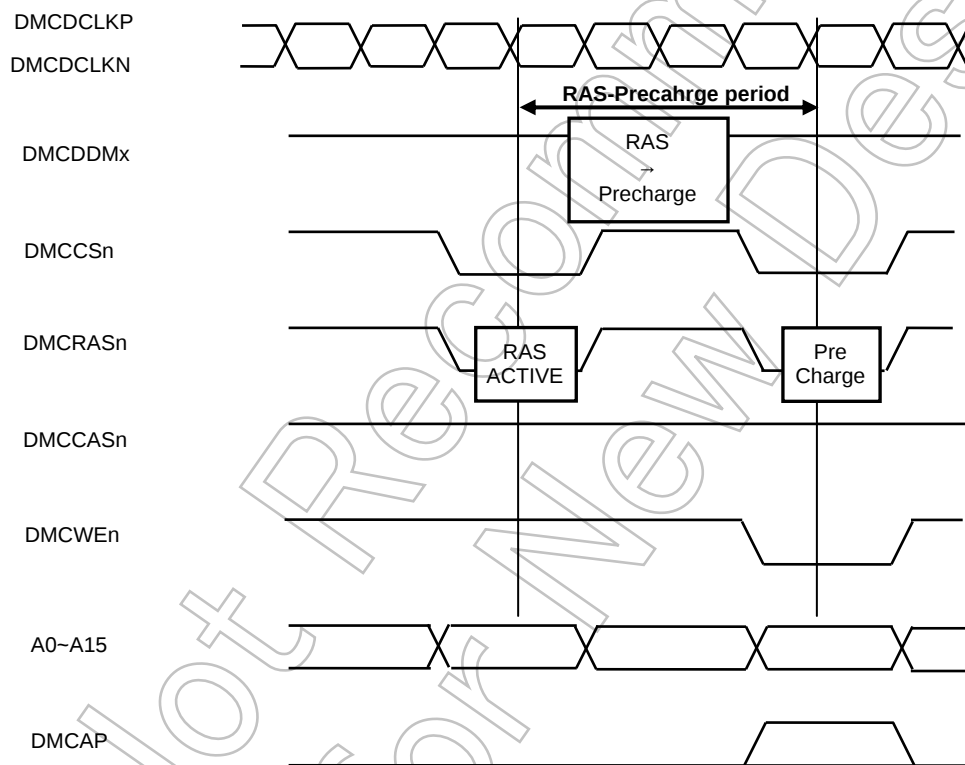


図 3.10.26 Active から Precharge までの時間

10. dmc_t_rc_5 (DMC t_rc Register)

Address = (0xF431_0000) + (0x0024)

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	—	—	Undefined	Read as undefined. Write as zero.
[3:0]	t_rc	R/W	0y1011	Active バンク A から Active バンク A の Delay 時間 (メモリクロック数) 0x0 ~ 0xF

(個別説明)

a. <t_rc>

同一 BANK 上での、Active バンクコマンドから Active バンクコマンドの Delay 時間を設定します。(メモリクロック数)

0x0 ~ 0xF

他の AC 設定または他の動作との兼ね合いにより、設定されたサイクル数よりも長いサイクルで動作する場合があります。よって、本設定は最低クロック数を設定します。

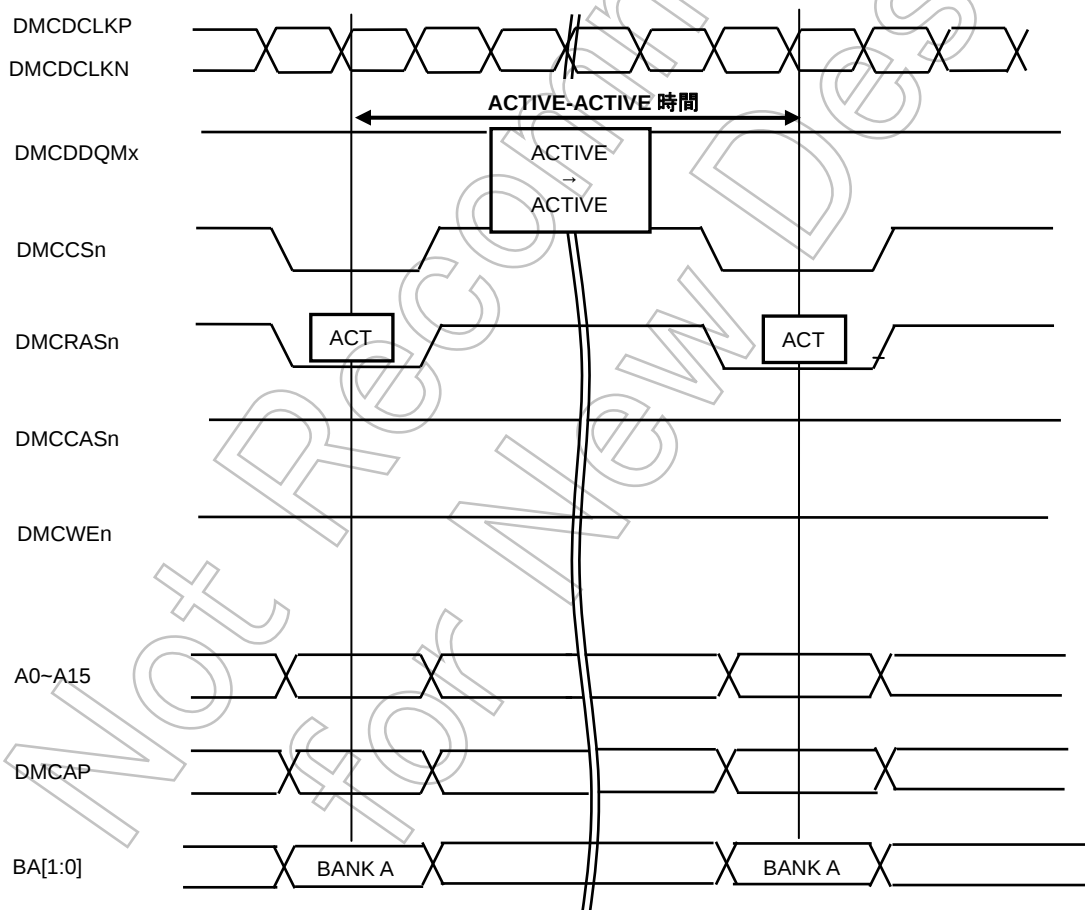


図 3.10.27 Active バンク A から Active バンク A 例

11. dmc_t_rcd_5 (DMC t_rcd Register)

Address = (0xF431_0000) + (0x0028)

Bit	Bit Symbol	Type	Reset Value	Description
[31:6]	—	—	Undefined	Read as undefined. Write as zero.
[5:3]	schedule_rcd	R/W	0y011	RAS から CAS まで Min delay 設定 (t_rcd の設定値 -3)で設定してください。
[2:0]	t_rcd	R/W	0y101	RAS から CAS まで Min delay 設定 (メモリクロック数) 0y000 ~ 0y111

(個別説明)

a. <schedule_rcd>

RAS から CAS まで Min delay を設定します。(メモリクロック数)
(t_rcd の設定値 -3)で設定してください。

b. <t_rcd>

RAS から CAS まで Min delay を設定します。(メモリクロック数)
0y000 ~ 0y111

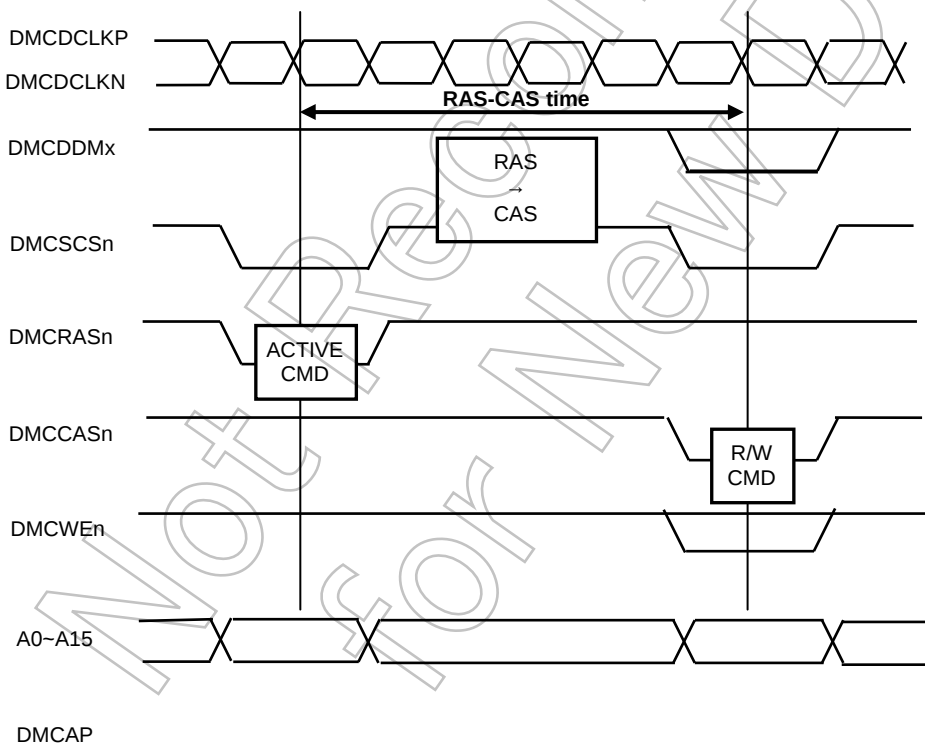


図 3.10.28 Active からリードコマンドのタイミング

12. dmc_t_rfc_5 (DMC t_rfc Register)

Address = (0xF431_0000) + (0x002C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:10]	—	—	Undefined	Read as undefined. Write as zero.
[9:5]	schedule_rfc	R/W	0y10000	autorefresh コマンド時間設定 (t_rfc の設定値 -3)で設定してください。
[4:0]	t_rfc	R/W	0y10010	Autorefresh コマンド時間設定 (メモリクロック数) 0y00000 ~ 0y11111

(個別説明)

a. <schedule_rfc>

Autorefresh コマンド時間を設定します。

(t_rfc の設定値 -3)で設定してください。

b. <t_rfc>

autorefresh コマンド時間を設定します。(メモリクロック数)

0y00000 ~ 0y11111

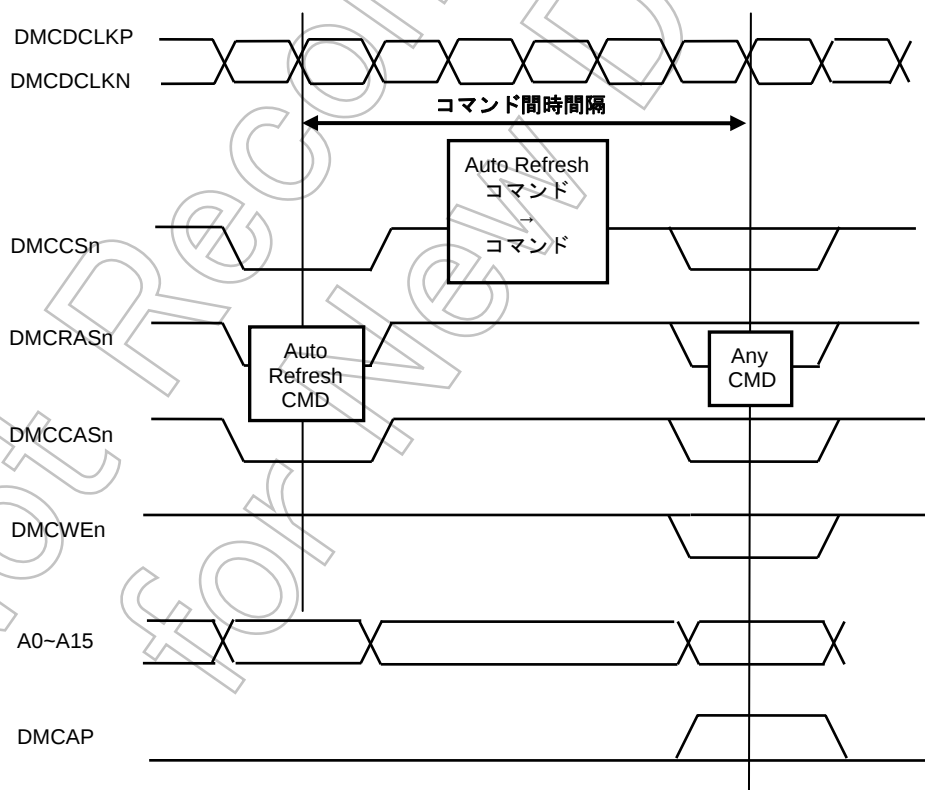


図 3.10.29 Autorefresh コマンドから他コマンドのタイミング例

13. dmc_t_rp_5 (DMC t_rp Register)

Address = (0xF431_0000) + (0x0030)

Bit	Bit Symbol	Type	Reset Value	Description
[31:6]	—	—	Undefined	Read as undefined. Write as zero.
[5:3]	schedule_rp	R/W	0y011	RAS に precharge Delay 設定 (t_rp の設定値 -3)で設定してください。
[2:0]	t_rp	R/W	0y101	precharge から RAS まで時間を設定(メモリクロック数) 0y000 ~ 0y111

(個別説明)

a. <schedule_rp>

precharge から RAS まで時間をを設定します。

(t_rp の設定値 -3)で設定してください。

b. <t_rp>

precharge から RAS まで時間をを設定します。(メモリクロック数)

0y000 ~ 0y111

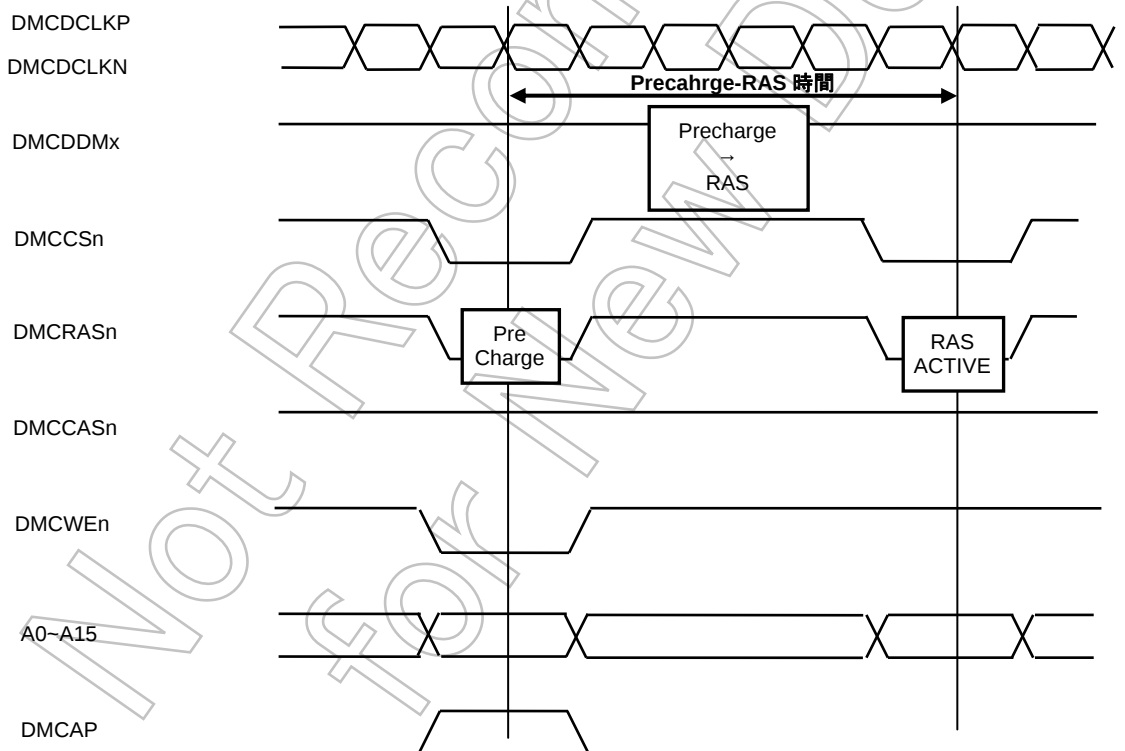


図 3.10.30 Precharge から他のコマンド(Autorefresh コマンドを含む)までの時間

14. dmc_t_rrd_5 (DMC t_rrd Register)

Address = (0xF431_0000) + (0x0034)

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	—	—	Undefined	Read as undefined. Write as zero.
[3:0]	t_rrd	R/W	0y0010	Active バンク A から Active バンク B の Delay 時間 (メモリクロック数) 0x0 ~ 0xF

(個別説明)

a. <t_rrd>

Active バンク A から Active バンク B の Delay 時間を設定します。(メモリクロック数)

0x0 ~ 0xF

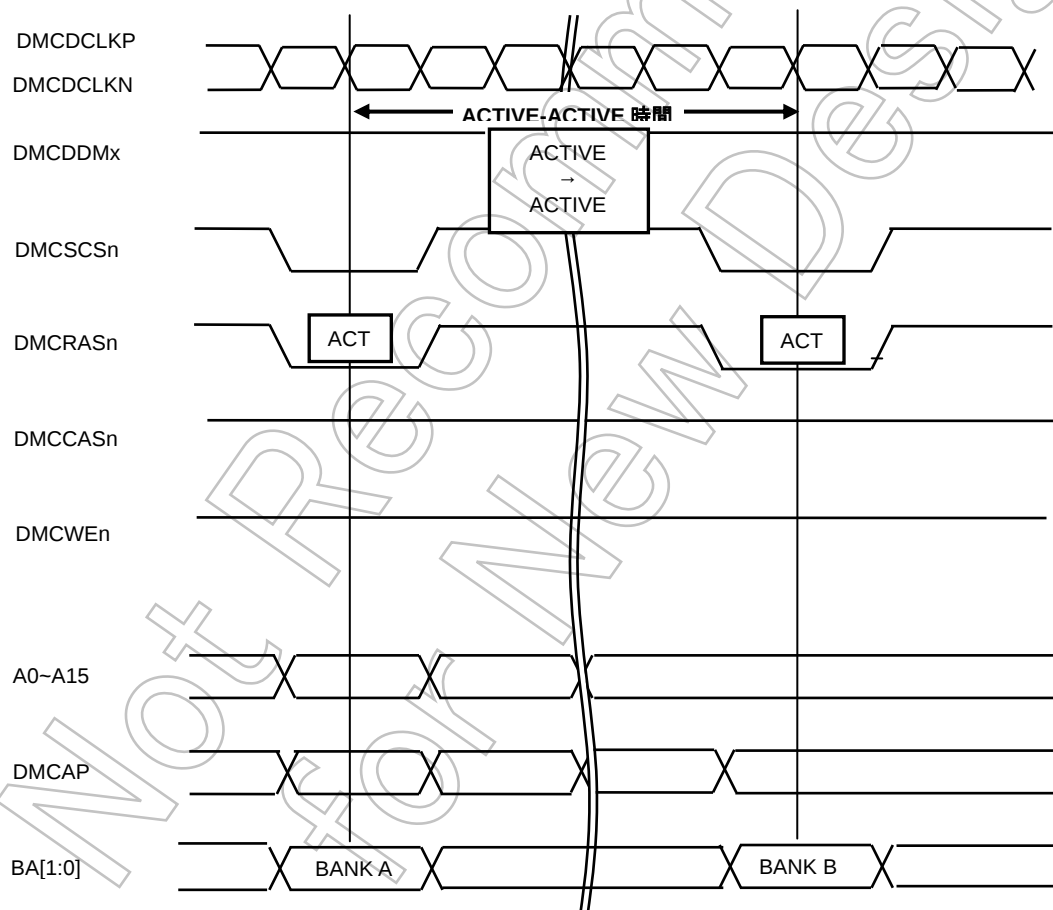


図 3.10.31 Active バンク A から他の Active バンク B の時間

15. dmc_t_wr_5 (DMC t_wr Register)

Address = (0xF431_0000) + (0x0038)

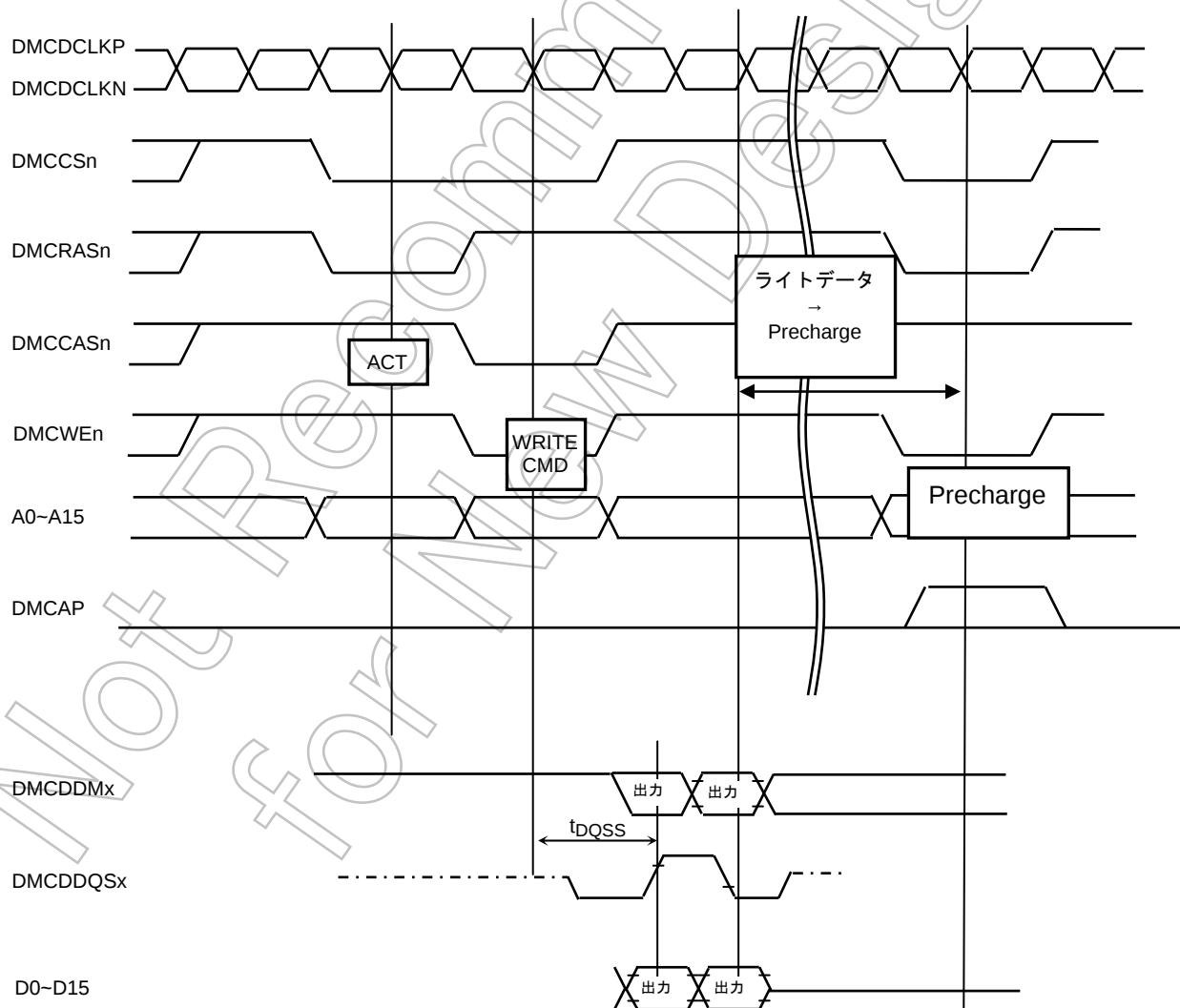
Bit	Bit Symbol	Type	Reset Value	Description
[31:3]	—	—	Undefined	Read as undefined. Write as zero.
[2:0]	t_wr	R/W	0y011	ライトの最終データから Precharge までの 設定値 0y000 ~ 0y111

(個別説明)

a. <t_wr>

ライトの最終データから Precharge までの 期間を設定します (メモリクロック数)。

実際の期間(メモリクロック数) : t_wr の設定値 + 1、ただし 0y000 の設定した場合は 9 クロックになります。



16. dmc_t_wtr_5 (DMC t_wtr Register)

Address = (0xF431_0000) + (0x003C)

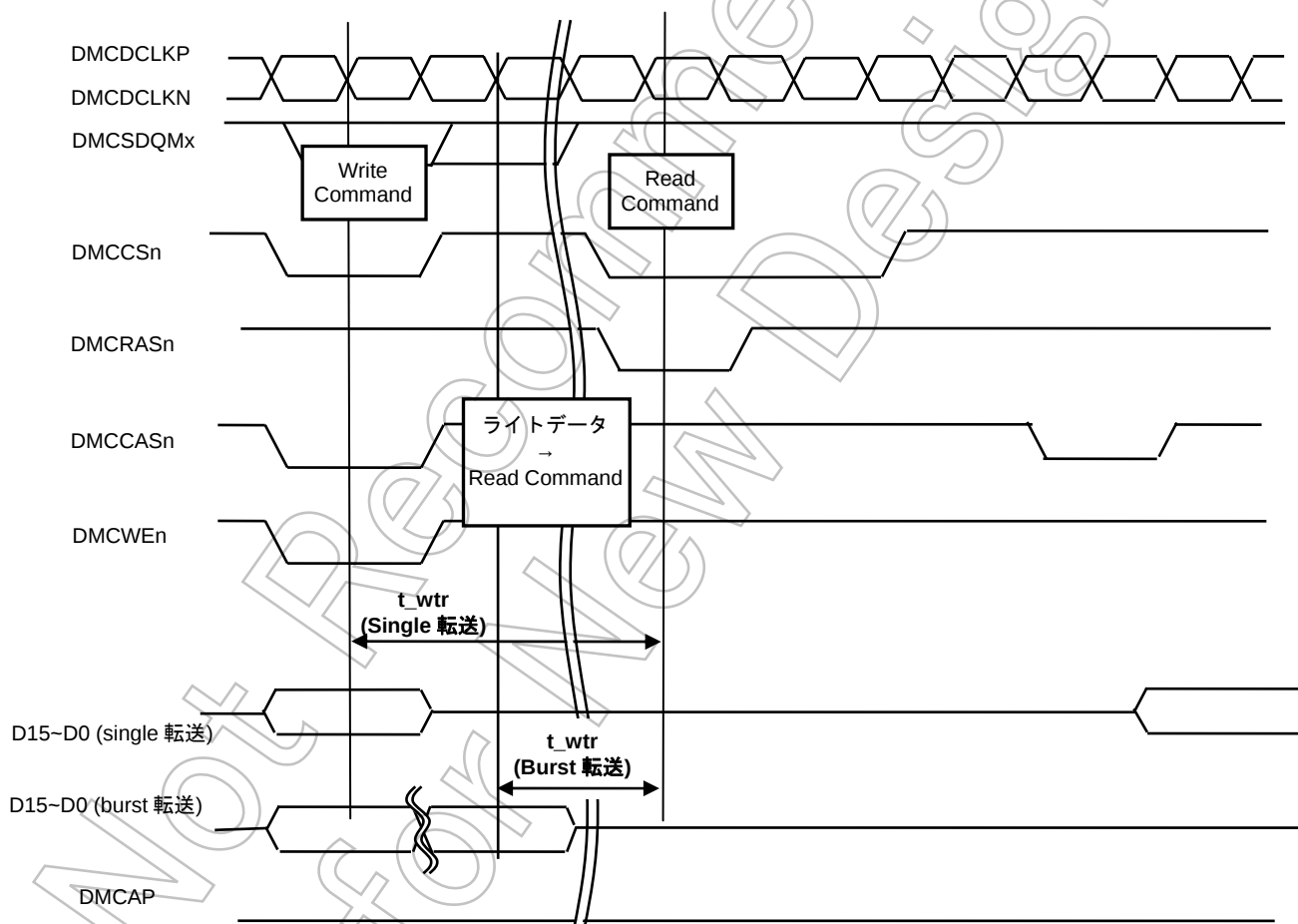
Bit	Bit Symbol	Type	Reset Value	Description
[31:3]	—	—	Undefined	Read as undefined. Write as zero.
[2:0]	t_wtr	R/W	0y010	ライトの最終データからリードコマンドまでの設定値 (メモリクロック数) 0y000 ~ 0y111

(個別説明)

a. <t_wtr>

ライトの最終データからリードコマンドまでの期間を設定します(メモリクロック数)

注) 0y000 時、8 メモリクロックになります。



17. dmc_t_xp_5 (DMC t_xp Register)

$$\text{Address} = (0xF431_0000) + (0x0040)$$

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:0]	t_xp	R/W	0x01	Powerdown から Exit 時間の設定値(メモリクロック数) 0x00 ~ 0xFF

(個別説明)

a. $\langle t_{xp} \rangle$

Powerdown Exit から他コマンドまでの時間を設定します。(メモリクロック数)

実際の期間(メモリクロック数) : t_{xp} の設定値 + 1

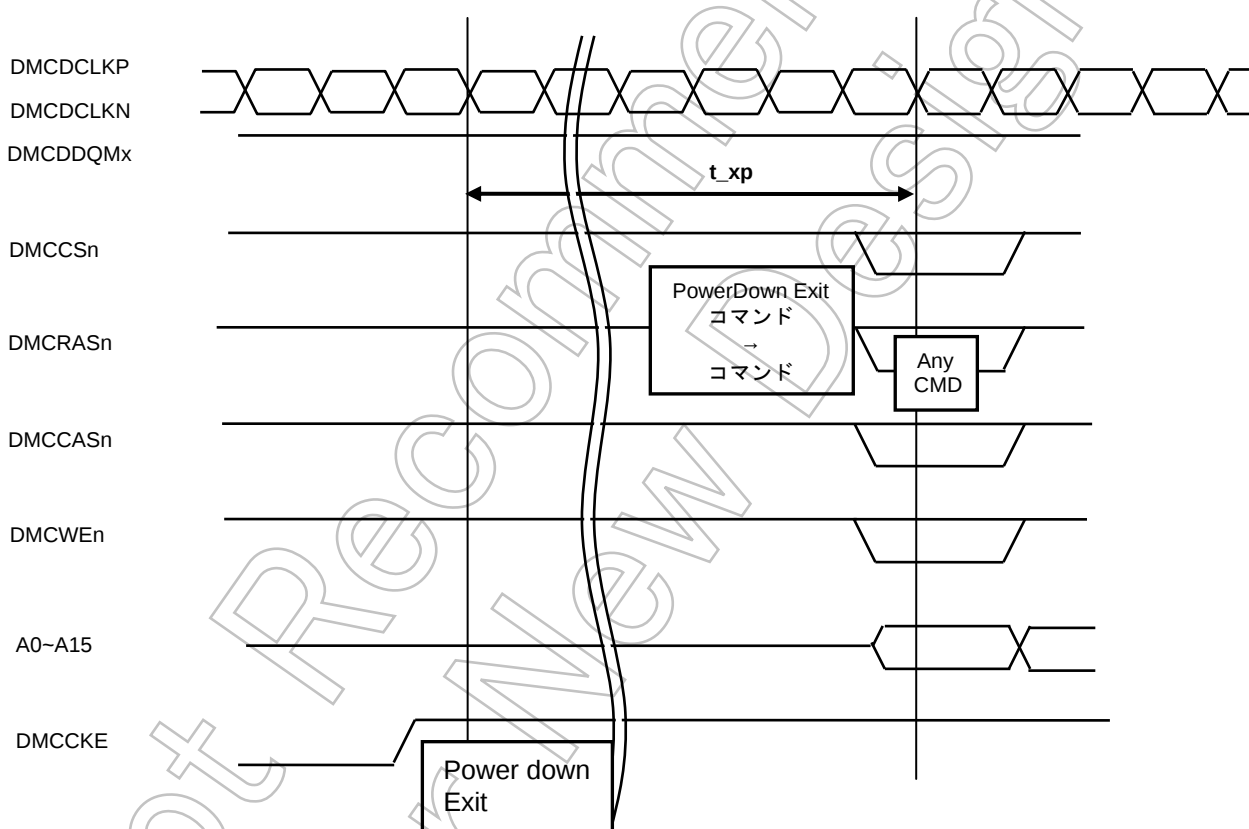


図 3.10.32 PowerDown エントリと Exit

18. dmc_t_xsr_5 (DMC t_xsr Register)

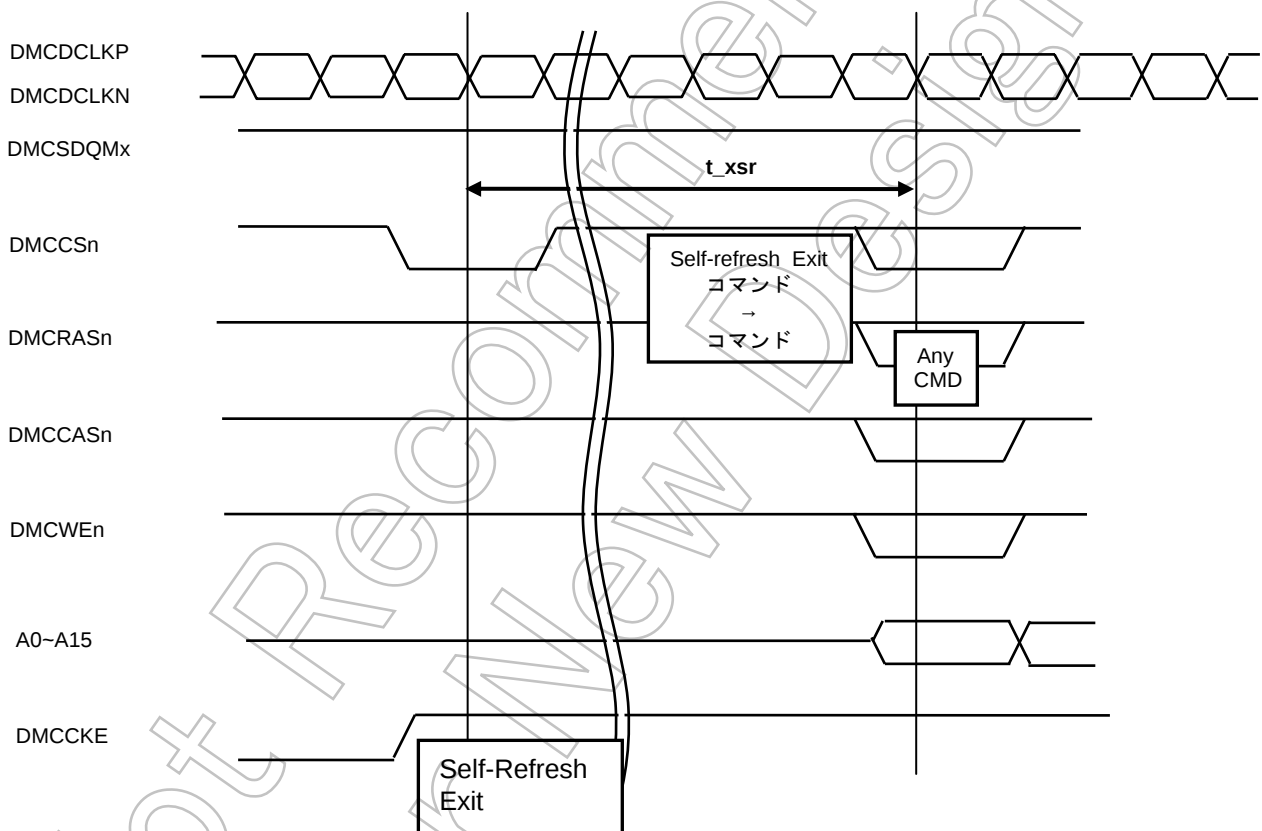
Address = (0xF431_0000) + (0x0044)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	—	—	Undefined	Read as undefined. Write as zero.
[7:0]	t_xsr	R/W	0x0A	Self-refresh Exit コマンド時間設定 : (メモリクロック数) 0x00 ~ 0xFF

(個別説明)

a. <t_xsr>

Self-refresh Exit コマンドから他のコマンドまでの時間を設定します。(メモリクロック数)
0x00 ~ 0xFF



19. dmc_t_esr_5 (DMC t_esr Register)

Address = (0xF431_0000) + (0x0048)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	—	—	Undefined	Read as undefined. Write as zero.
[7:0]	t_esr	R/W	0x14	Self-refresh Entry から Exit までの実行最低限時間の設定 : (メモリクロック数) 0x00 ~ 0xFF

注) Self-refresh は Wakeup の実行により Exit します。

このレジスタは Self-refresh Entry から Exit まで最低限時間を設定します。

(個別説明)

a. <t_esr>

Self-refresh 実行時間を設定します。(メモリクロック数)

0x00 ~ 0xFF

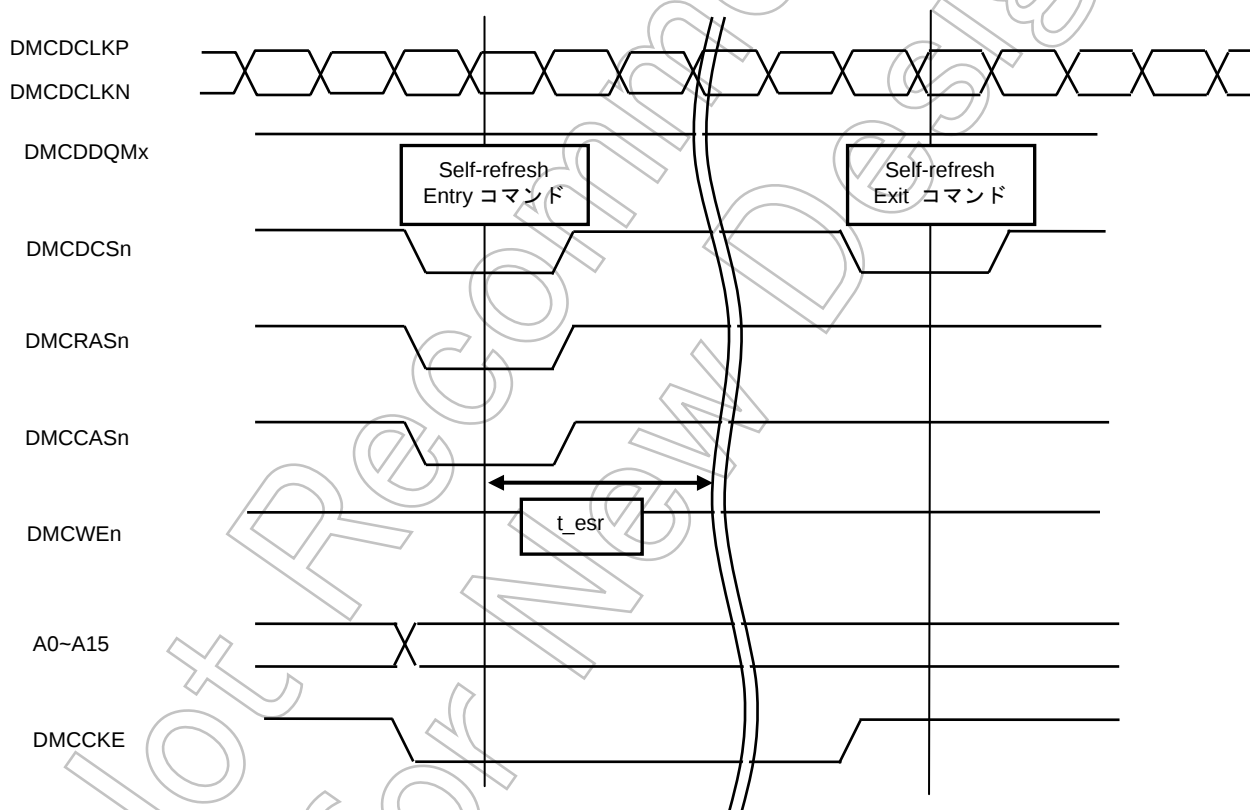


図 3.10.33 Self-Refresh エントリと Exit タイミング

20. dmc_id_<0-5>_cfg_5 (DMC id_<0-5>_cfg Registers)

Address = (0xF431_0000) + (0x0100)

Address = (0xF431_0000) + (0x0104)

Address = (0xF431_0000) + (0x0108)

Address = (0xF431_0000) + (0x010C)

Address = (0xF431_0000) + (0x0110)

Address = (0xF431_0000) + (0x0114)

Bit	Bit Symbol	Type	Reset Value	Description
[31:10]	–	–	Undefined	Read as undefined. Write as zero.
[9:2]	qos_max	R/W	0x00	maximum QoS : 0x00 ~ 0xFF
[1]	qos_min	R/W	0y0	minimum QoS 選択: 0y0: qos_max mode 0y1: qos_min mode
[0]	qos_enable	R/W	0y0	QoS の Enable: 0y0 = Disable 0y1 = Enable

QoS 設定レジスタ一覧

レジスタ	アドレス	対応 AHB バス
dmc_id_0_cfg_5	(0xF431_0000) + (0x0100)	AHB0 : CPU Data
dmc_id_1_cfg_5	(0xF431_0000) + (0x0104)	AHB1 : CPU Inst
dmc_id_2_cfg_5	(0xF431_0000) + (0x0108)	AHB2 : LCD C
dmc_id_3_cfg_5	(0xF431_0000) + (0x010C)	AHB3 : マルチレイヤーバスマトリクス 2 (LCDDA, USB)
dmc_id_4_cfg_5	(0xF431_0000) + (0x0110)	AHB4 : DMA1
dmc_id_5_cfg_5	(0xF431_0000) + (0x0114)	AHB5 : DMA2

(個別説明)

a. <qos_max>

maximum QoS を設定します。

0x00 ~ 0xFF

b. <qos_min>

minimum QoS 選択:

0y0 = qos_max mode

0y1 = qos_min mode

QoS min は QoS max より高い優先度を持ちます。

c. <qos_enable>

QoS の Enable を設定します。

0y0 = Disable

0y1 = Enable

21. dmc_chip_0_cfg_5 (DMC chip_0_cfg Registers)

Address = (0xF431_0000) + (0x0200)

Bit	Bit Symbol	Type	Reset Value	Description
[31:17]	–	–	Undefined	Read as undefined. Write as zero.
[16]	brc_n_rbc	R/W	0y0	SDRAM アドレス構成 : 0y0 = row, bank, column 0y1 = bank, row, column
[15:8]	address_match	R/W	0xFF	スタートアドレス[31:24]を設定 : 0x00 ~ 0xFF
[7:0]	address_mask	R/W	0x00	スタートアドレス[31:24]のマスク値を設定 : 値 1 のビットはアドレス比較対象ビット 0x00 ~ 0xFF

(個別説明)

a. <brc_n_rbc>

SDRAM アドレス構成を選択します。

0y0 = row, bank, column

0y1 = bank, row, column

b. <address_match>

スタートアドレス[31:24]を設定するレジスタです。

512MB 未満のメモリを接続した場合は、設定した CS エリア以外の DMC エリア(未使用)をアクセスしないでください。

注) スタートアドレスを設定する場合は 3.3 章メモリマップを参照し、有効領域を確認した上で設定してください。

c. <address_mask>

このレジスタで CS 領域を設定します。

スタートアドレス[31:24]に対してどのビットの値を比較するか、比較しないかを設定します。

0y0 = 比較しない

0y1 = 比較する

22. dmc_user_config_5 (DMC user_config Register)

Address = (0xF431_0000) + (0x0304)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7]	Reserved	–	Undefined	Read as undefined. Write as zero.
[6:4]	dqs_in	WO	0y000	DDR SDRAM 常数設定 : 必ず 0y101 に設定してください
[3:1]	dmc_clk_in	WO	0y000	DDR SDRAM 常数設定 : 必ず 0y100 に設定してください
[0]	sdr_width	WO	0y0	対応する外部 DDR SDRAM のデータバス幅の設定 0y0 : 16bit 0y1 : Reserved

(個別説明)

a. <sdr_width>

対応する外部 SDR メモリのデータバス幅を設定します。

0y0 = 16bit

0y1 = Reserved

3.10.5.2 SMC (Static Memory Controller)

外部メモリ (NOR フラッシュメモリ、Mask ROM、SRAM 等) を制御する SMC (Static Memory Controller) を内蔵しています。

(1) SMC 機能概要

表 3.10.11 に、SMC の特長を示します。

表 3.10.11 SMC の特長

特長	チップセレクト 0/1
サポートメモリ	外部 Static メモリ (NOR フラッシュメモリ、SRAM 等) セパレートバスのみサポート
データバス幅	16bit データバス幅をサポート
アクセス空間	チップセレクトによる 2 つの空間をサポート それぞれの最大アクセス空間： SMCCS0n : 512 MB SMCCS1n : 512 MB
タイミング調整	レジスタによる AC タイミングの調整が可能
クロック	外部制御端子を生成するクロックをクロックコントローラの CLKCR5<SEL_SMC_MCLK> で f_{HCLK} 、 $f_{HCLK}/2$ を選択可能
外部制御端子	D15~D0, A23~A0, SMCBE0n, SMCBE1n, SMCCS0n, SMCCS1n, SMCOEn, SMCWEn,

(2) SMC ブロック図

以下に、SMC 回路のブロック図を示します。

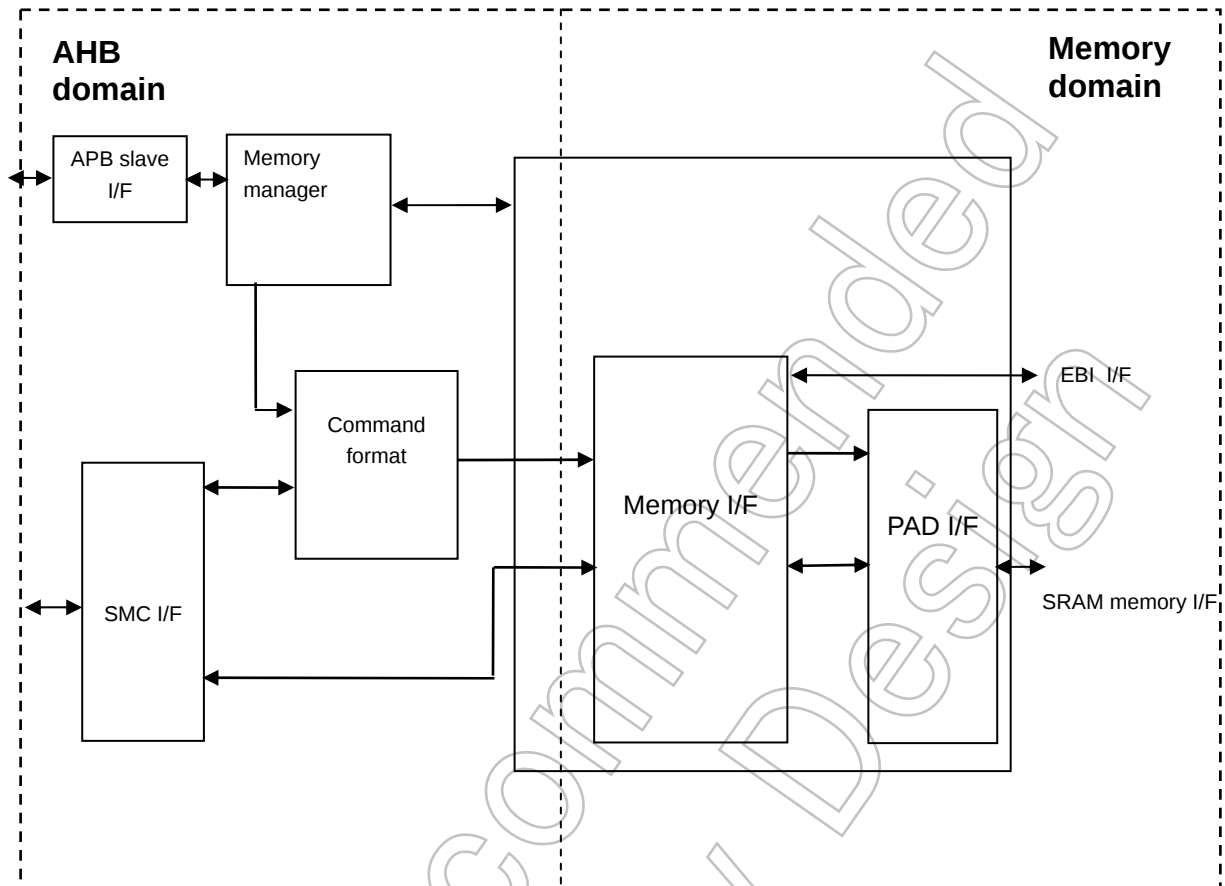


図 3.10.34 SMC ブロック図

(a) アービタ

アービタは **SMC I/F** とメモリマネージャからアクセスコマンドを受けて、アクセスの調停後、優先度が一番高いコマンドをメモリ I/F に渡します。データはメモリ I/F から **SMC I/F** に直接接続されています。

(b) メモリマネージャ

タイミングレジスタの更新、メモリに対してのコマンドを制御しています。

(3) SMC Function 動作

(a) APB slave I/F

APB slave I/F はリードとライト 1state wait を追加

下記の場合は 1state 以上の wait が生成される：

- ・ 例外(outstanding) direct コマンド
- ・ メモリコマンド要求があつて、ただし、前のメモリコマンド実行は未完了。

(b) フォーマット

1. ハザード処理

同一のバスマスタ単体が、外部メモリをアクセスする場合、実際のメモリアクセス順序は命令の順序通りに実行されます。

ただし、複数のバスマスタが、外部メモリアクセスする場合、READ と WRITE は各々別々の Buffer に一旦蓄積され、優先順位回路によって実行される為、READ と WRITE の実行順序が入れ替わることがあります。

十分な時間を取るか、先に実行したアクセスが完了したことを確認するか、または、共用するメモリは内蔵 RAM を利用するなどの管理をしてください。

2. SRAM メモリをアクセス

- Standard SRAM access
- Memory address shifting
- Memory burst alignment

非同期ページモードメモリに対応するためには、burst align 設定が必要です。

3.10.5.2章 (4)d. smc_set_opmode_5 レジスタ説明を参照してください。

注) Page 概念のないデバイスを使用する場合は、burst align 設定の必要がありません。

Memory burst length：設定可能なメモリバースト転送の length は 4 です。

(c) メモリマネージャ動作

SMC 状態の制御およびチップ設定レジスタの更新を管理しています。

(d) メモリ I/F 動作

メモリ I/F はコマンドを発行、これらコマンドタイミングを制御します。

表 3.10.12 Static メモリのセットアップ例

base アドレス= 0xF431_1000

Register address	Write data	Description
0x0014	0x00029266	smc_set_cycles 5
0x0018	0x00000809	smc_set_opmode 5
0x0010	0x00400000	smc_direct cmd 5

Not Recommended
for New Design

(4) MPMC1 の SMC レジスタの説明

表 3.10.13 SFR 一覧

base アドレス= 0xF431_1000

Register Name	Address (base+)	Type	Reset Value	Description
Reserved	0x0000	—	Undefined	Read as undefined. Write as zero.
Reserved	0x0004	—	Undefined	Read as undefined. Write as zero.
Reserved	0x0008	—	—	Writing prohibited
Reserved	0x000C	—	—	Writing prohibited
smc_direct_cmd_5	0x0010	WO	—	SMC Direct Command Register
smc_set_cycles_5	0x0014	WO	—	SMC Set Cycles Register
smc_set_opmode_5	0x0018	WO	—	SMC Set Opmode Register
Reserved	0x0020	—	Undefined	Read as undefined. Write as zero.
smc_sram_cycles0_0_5	0x0100	RO	0x0002B3CC	SMC SRAM Cycles Registers <0-1>
smc_sram_cycles0_1_5	0x0120			
Reserved	0x0140	RO	Undefined	Read as undefined
Reserved	0x0160	RO	Undefined	Read as undefined
smc_opmode0_0_5	0x0104	RO	0x20E00802	SMC Opmode Registers <0-1>
smc_opmode0_1_5	0x0124		0x60E00802	
Reserved	0x0144	RO	Undefined	Read as undefined
Reserved	0x0164	RO	Undefined	Read as undefined
Reserved	0x0200	—	Undefined	Read as undefined. Write as zero.
Reserved	0x0204	—	Undefined	Read as undefined. Write as zero.
Reserved	0x0E00	—	Undefined	Read as undefined. Write as zero.
Reserved	0x0E04	—	Undefined	Read as undefined. Write as zero.
Reserved	0x0E08	—	Undefined	Read as undefined. Write as zero.
Reserve	0x0FE0-0x0FEC	—	Undefined	Read as undefined. Write as zero.
Reserve	0x0FF0-0x0FFC	—	Undefined	Read as undefined. Write as zero.

注) APB は single-word 32-bit アクセスのみをサポート、レジスタのリード/ライトは single-word 32-bit で実行してください。

b. smc_direct_cmd_5 (SMC Direct Command Register)

Address = (0xF431_1000) + (0x0010)

Bit	Bit Symbol	Type	Reset Value	Description
[31:26]	—	—	Undefined	Read as undefined. Write as zero.
[25:23]	chip_select	WO	—	CS 選択 : 0y000 = CS0 0y001 = CS1 0y010-0y111 = Reserved
[22:21]	cmd_type	WO	—	カレントコマンド: 0y00 = Reserved 0y01 = Reserved 0y10 = UpdateRegs 0y11 = Reserved
[20:0]	—	—	Undefined	Read as undefined. Write as zero.

(個別説明)

a. <chip_select>

CS 選択を選択します。

0y000 = CS0

0y001 = CS1

0y010-0y111 = Reserved

b. <cmd_type>

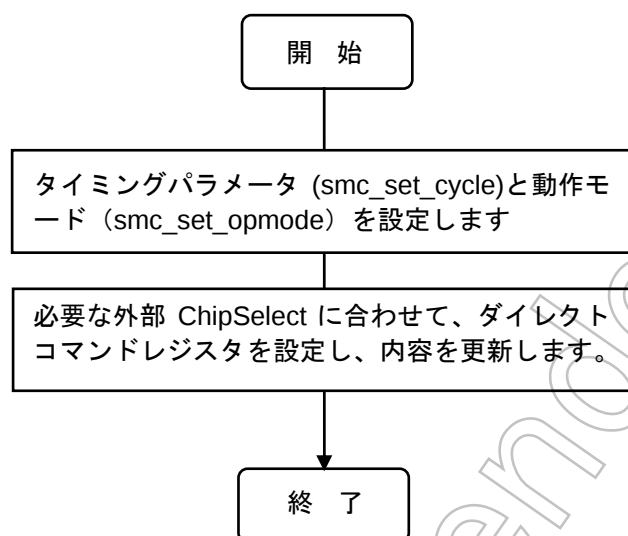
カレントコマンドを選択します。

0y00 = Reserved

0y01 = Reserved

0y10 = UpdateRegs

0y11 = Reserved



c. smc_set_cycles_5 (SMC Set Cycles Register)

Address = (0xF431_1000) + (0x0014)

Bit	Bit Symbol	Type	Reset Value	Description
[31:23]	–	–	Undefined	Read as undefined. Write as zero.
[22:20]	Reserved	–	Undefined	Read as undefined. Write as zero.
[19:17]	Set_t5	WO	–	t _{TR} の設定値(保持レジスタ) 0y000 ~ 0y111
[16:14]	Set_t4	WO	–	t _{PC} の設定値(保持レジスタ) 0y000 ~ 0y111
[13:11]	Set_t3	WO	–	t _{WP} の設定値(保持レジスタ) 0y000 ~ 0y111
[10:8]	Set_t2	WO	–	t _{CEOE} の設定値(保持レジスタ) 0y000 ~ 0y111
[7:4]	Set_t1	WO	–	t _{WC} の設定値(保持レジスタ) 0y0000 ~ 0y1111
[3:0]	Set_t0	WO	–	t _{RC} の設定値(保持レジスタ) 0y0000 ~ 0y1111

このレジスタは、StaticMemory のアクセスサイクルを調整するレジスタです。メモリの要求する A.C.に合わせて設定してください。尚、外部端子による Wait 信号との併用の場合、本設定と外部 Wait 信号を両方を満足する形でサイクルが決定します。

また、外部 Wait 信号は、同期モードでのみ有効となり、非同期モードでは有効となりませんので注意してください。

このレジスタは保持レジスタであり、設定値を有効にするため、下記の動作によりこのレジスタの設定値がメモリマネージャの configuration レジスタに update されて、有効になります。

- smc_direct_cmd レジスタでは UpdateRegs を実行します。

(個別説明)

- <Set_t5>
t_{TR} の設定値(保持レジスタ)を設定します。
0y000 ~ 0y111
- <Set_t4>
t_{PC} の設定値(保持レジスタ)を設定します。
0y000 ~ 0y111
- <Set_t3>
t_{WP} の設定値(保持レジスタ)を設定します。
0y000 ~ 0y111

- d. <Set_t2>
t_{CEOE} の設定値(保持レジスタ)を設定します。
0y000 ~ 0y111
- e. <Set_t1>
t_{wc} の設定値(保持レジスタ)を設定します。
0y000 ~ 0y111
- f. <Set_t0>
t_{RC} の設定値(保持レジスタ)を設定します。
0y000 ~ 0y111

タイミング設定例

レジスタ設定例: SMC Set Cycles Register = 0x0002B1C3

レジスタ設定値	t _{TR}	t _{PC}	t _{WP}	t _{CEOE}	T _{wc}	t _{RC}
0x0002B1C3	—	—	—	1	—	3

— : don't care

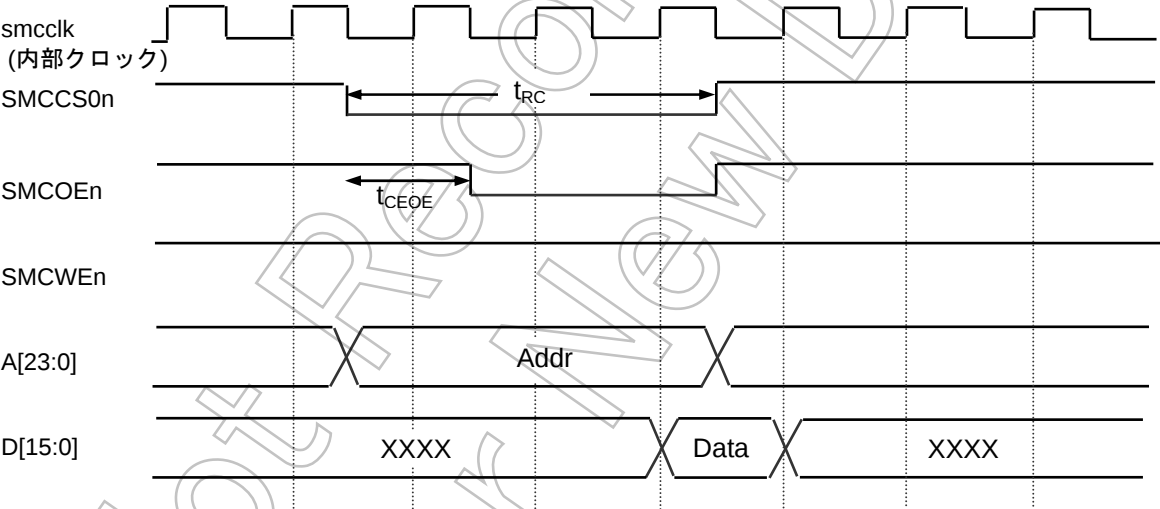


図 3.10.35 非同期 Read

レジスタ設定例: SMC Set Cycles Register = 0x0002934C

レジスタ設定値	t_{TR}	t_{PC}	t_{WP}	t_{CEOE}	T_{WC}	t_{RC}
0x0002934C	—	—	2	—	4	—

— : don't care

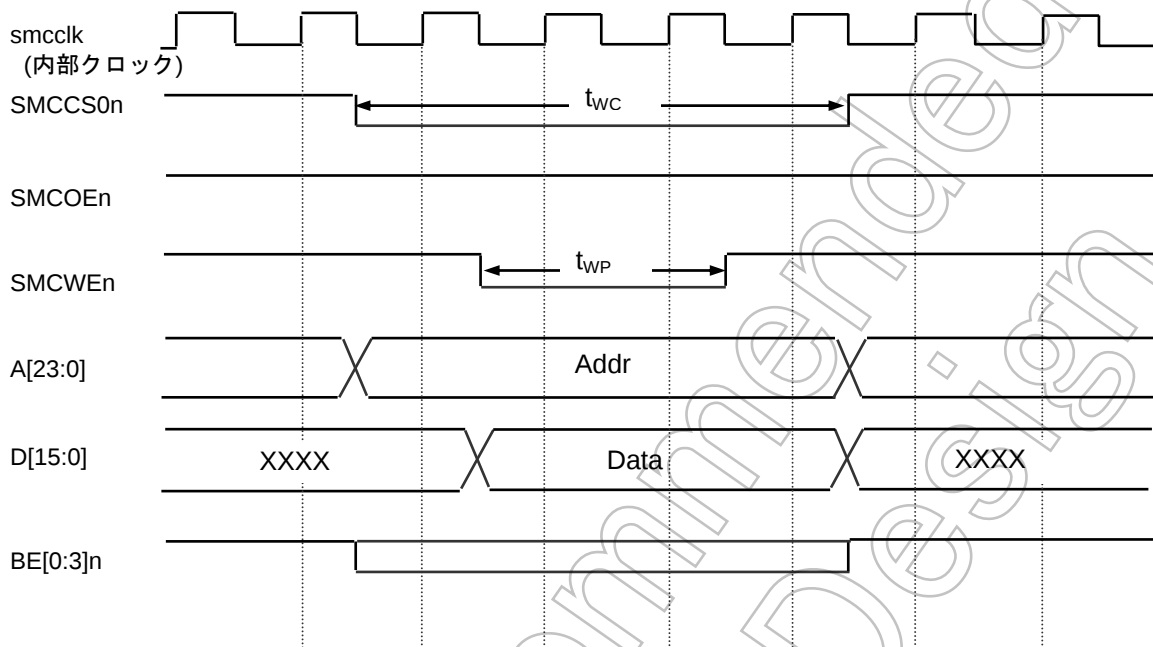


図 3.10.36 非同期 Write

レジスタ設定例: SMC Set Cycles Register = 0x000272C3

レジスタ設定値	t_{TR}	t_{PC}	t_{WP}	t_{CEOE}	T_{WC}	t_{RC}
0x000272C3	—	1	—	2	—	3

— : don't care

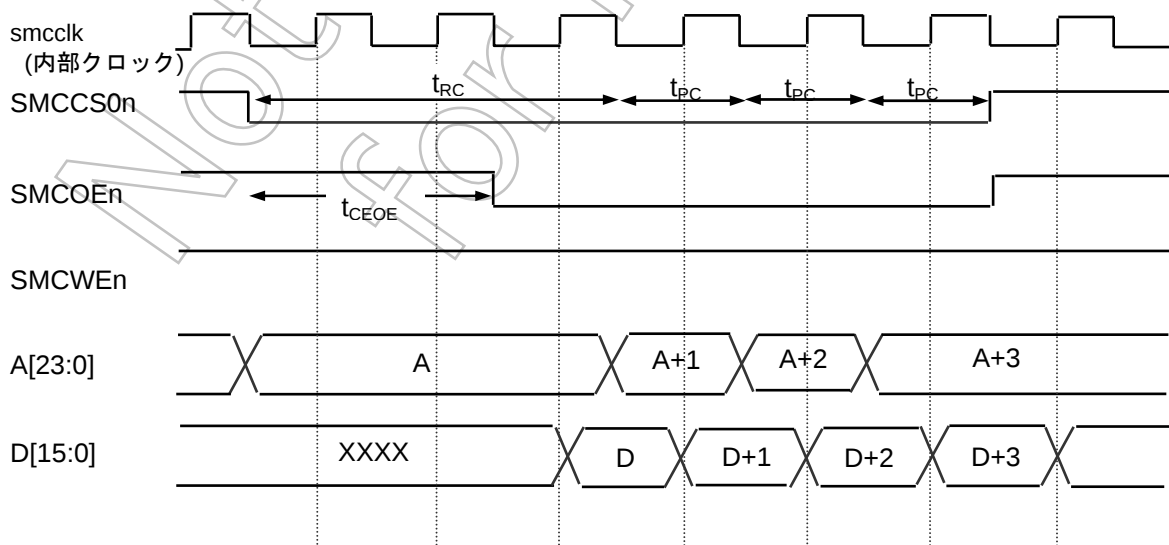


図 3.10.37 非同期 Page Read

レジスタ設定例: SMC Set Cycles Register = 0x00029143

レジスタ設定値	t_{TR}	t_{PC}	t_{WP}	t_{CEOE}	T_{WC}	t_{RC}
0x00029143	1	—	2	1	4	3

— : don't care

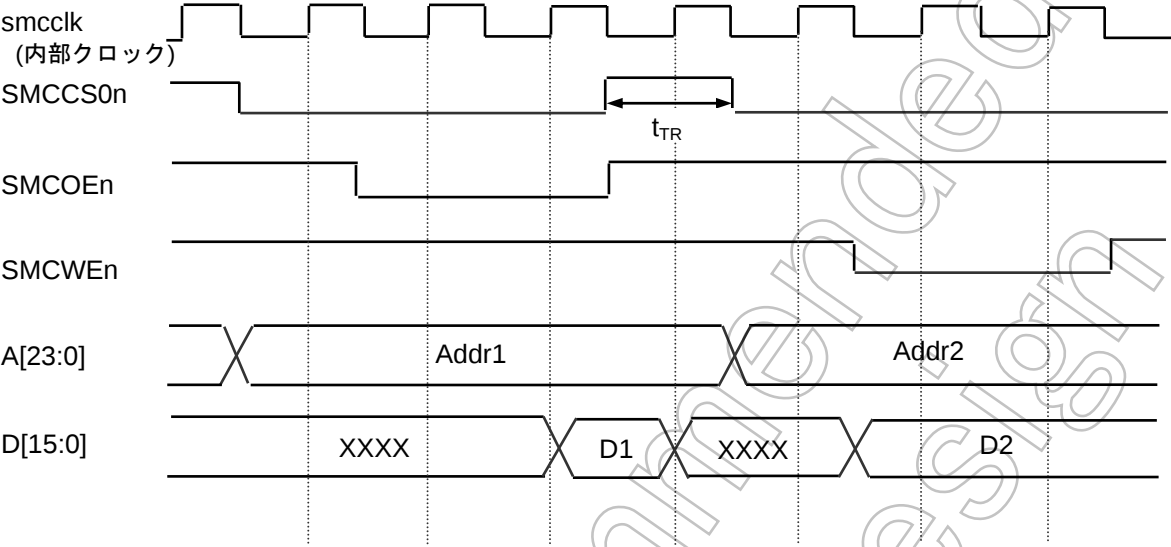


図 3.10.38 非同期 Read 後に非同期 Write

d. smc_set_opmode_5 (SMC Set Opmode Register)

Address = (0xF431_1000) + (0x0018)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined. Write as zero.
[15:13]	set_burst_align	WO	–	メモリバースト境界の分割設定: (保持レジスタ) 0y000 = 任意アドレスの境界を越えられる。 0y001 = 32 ビートバースト境界で分割 0y010 = 64 ビートバースト境界で分割 0y011 = 128 ビートバースト境界で分割 0y100 = 256 ビートバースト境界で分割 other = Reserved
[12]	set_bls	WO	–	Byte Enable(SMCBE0-1)タイミング設定: 0y0 = SMCCSn タイミング 0y1 = SMCWEn タイミング
[11]	Reserved	WO	–	Read as undefined. Write as zero.
[10]	–	–	Undefined	Read as undefined. Write as zero.
[9:7]	set_wr_bl	WO	–	ライトのバースト長: (保持レジスタ) 0y000 = 1 ビート 0y001 = 4 ビート other = Reserved
[6]	set_wr_sync	WO	–	wr_sync フィード設定値: (保持レジスタ) 0y0 = 非同期ライトモード 0y1 = Reserved
[5:3]	set_rd_bl	WO	–	リードのバースト長: (保持レジスタ) 0y000 = 1 ビート 0y001 = 4 ビート other = Reserved
[2]	set_rd_sync	WO	–	rd_sync フィード設定値: (保持レジスタ) 0y0 = 非同期リードモード 0y1 = Reserved
[1:0]	set_mw	WO	–	メモリデータバス幅設定値: (保持レジスタ) 0y00 = Reserved 0y01 = 16bits 0y10 = Reserved 0y11 = Reserved

このレジスタは保持レジスタであり、設定値を有効にするため、下記の動作により、このレジスタの設定値がメモリマネージャの **configuration** レジスタに **update** されて、有効になります。

- smc_direct_cmd レジスタでは UpdateRegs を実行します。

(個別説明)

a. <set_burst_align>

非同期転送の場合は

set_rd_sync = 0 時、MPMC0 は常にメモリバースト境界にリードバーストを調整します。

set_wr_sync = 0 時、MPMC0 は常にメモリバースト境界にライトバーストを調整します。

- b. <set_bls>
Byte Enable(SMCBE0-1)タイミング設定:
0y0 = SMCCSn タイミング
0y1 = SMCWEn タイミング
- c. <set_wr_bl>
ライトのバースト長:
0y000 = 1 ビート
0y001 = 4 ビート
other = Reserved
- d. <set_wr_sync>
wr_sync フィールド設定値: (保持レジスタ)
0y0 = 非同期ライトモード
0y1 = Reserved
- e. <set_rd_bl>
リードのバースト長: (保持レジスタ)
0y000 = 1 ビート
0y001 = 4 ビート
other = Reserved
- f. <set_rd_sync>
rd_sync フィールド設定値: (保持レジスタ)
0y0 = 非同期リードモード
0y1 = Reserved
- g. <set_mw >
メモリデータバス幅設定値: (保持レジスタ)
0y00 = Reserved
0y01 = 16bits
0y10 = Reserved
0y11 = Reserved

b. smc_sram_cycles0_0_5 (SMC SRAM Cycles Registers0 <0>)

Address = (0xF431_1000) + (0x0100)

Bit	Bit Symbol	Type	Reset Value	Description
[31:20]	–	–	Undefined	Read as undefined.
[19:17]	t_tr	RO	0y001	SRAM chip 設定のターンアラウンド時間 0y000 ~ 0y111
[16:14]	t_pc	RO	0y010	ページサイクル時間 : 0y000 ~ 0y111
[13:11]	t_wp	RO	0y110	smc_we_n_0 の delay 時間 : 0y000 ~ 0y111
[10:8]	t_ceoe	RO	0y011	smc_oe_n_0 の delay 時間 : 0y000 ~ 0y111
[7:4]	t_wc	RO	0y1100	ライトサイクル時間 : 0y0000 ~ 0y1111
[3:0]	t_rc	RO	0y1100	リードサイクル時間 : 0y0000 ~ 0y1111

(個別説明)

a. <t_tr>

SRAM chip 設定のターンアラウンド時間を設定します。

0y000 ~ 0y111

b. <t_pc>

ページサイクル時間を設定します。

0y000 ~ 0y111

c. <t_wp>

smc_we_n_0 の delay 時間を設定します。

0y000 ~ 0y111

d. <t_ceoe>

smc_oe_n_0 の delay 時間を設定します。

0y000 ~ 0y111

e. <t_wc>

ライトサイクル時間を設定します。

0y0000 ~ 0y1111

f. <t_rc>

リードサイクル時間を設定します。

0y0000 ~ 0y1111

• smc_sram_cycles0_x_5 (SMC SRAM Cycles Registers 0 <x>) (x = 0~1)

上記のレジスタ群について、構造および説明はsmc_sram_cycles0_0_5 と同様のため、smc_sram_cycles0_0_5 の説明を参照してください。またレジスタ名およびアドレスは表 3.10.13 SFR一覧を参照してください。

c. smc_opmode0_0_5 (SMC Opmode Registers 0 <0>)

Address = (0xF431_1000) + (0x0104)

Bit	Bit Symbol	Type	Reset Value	Description
[31:24]	Reserved	RO	0x20	0x20 がリードされます。
[23:16]	Reserved	RO	0xE0	0xE0 がリードされます。
[15:13]	burst_align	RO	0y000	メモリバースト境界で分割の設定値： 0y000 = 任意アドレスの境界を越えられる。 0y001 = 32 ビートバースト境界で分割 0y010 = 64 ビートバースト境界で分割 0y011 = 128 ビートバースト境界で分割 0y100 = 256 ビートバースト境界で分割 other = Reserved
[12]	bls	RO	0y0	bls タイミング設定： 0y0 = chip select 0y1 = Reserved
[11]	Reserved	RO	0y1	
[10]	–	–	Undefined	Read as undefined.
[9:7]	wr_bl	RO	0y000	ライト用メモリのバースト長： 0y000 = 1 ビート 0y001 = 4 ビート other = Reserved
[6]	wr_sync	RO	0y0	メモリ動作モード： 0y0 = ライト非同期動作 0y1 = Reserved
[5:3]	rd_bl	RO	0y000	リード用メモリのバースト長： 0y000 = 1 ビート 0y001 = 4 ビート other = Reserved
[2]	rd_sync	RO	0y0	メモリ動作モード： 0y0 = リード非同期動作 0y1 = Reserved
[1:0]	mw	RO	0y10	メモリデータバス幅設定値： 0y00 = Reserved 0y01 = 16bits 0y10 = Reserved 0y11 = Reserved

d. smc_opmode0_1_5 (SMC Opmode Registers 0 <1>)

Address = (0xF431_1000) + (0x0124)

Bit	Bit Symbol	Type	Reset Value	Description
[31:24]	Reserved	RO	0x60	0x60 がリードされます。
[23:16]	Reserved	RO	0xE0	0xE0 がリードされます。
[15:13]	burst_align	RO	0y000	メモリバースト境界で分割の設定値： 0y000 = 任意アドレスの境界を越えられる。 0y001 = 32 ビートバースト境界で分割 0y010 = 64 ビートバースト境界で分割 0y011 = 128 ビートバースト境界で分割 0y100 = 256 ビートバースト境界で分割 other = Reserved
[12]	bls	RO	0y0	bls タイミング設定： 0y0 = chip select 0y1 = Reserved
[11]	Reserved	RO	0y1	
[10]	–	–	Undefined	Read as undefined.
[9:7]	wr_bl	RO	0y000	ライト用メモリのバースト長： 0y000 = 1 ビート 0y001 = 4 ビート other = Reserved
[6]	wr_sync	RO	0y0	メモリ動作モード： 0y0 = ライト非同期動作 0y1 = Reserved
[5:3]	rd_bl	RO	0y000	リード用メモリのバースト長： 0y000 = 1 ビート 0y001 = 4 ビート other = Reserved
[2]	rd_sync	RO	0y0	メモリ動作モード： 0y0 = リード非同期動作 0y1 = Reserved
[1:0]	mw	RO	0y10	メモリデータバス幅設定値： 0y00 = Reserved 0y01 = 16bits 0y10 = Reserved 0y11 = Reserved

(個別説明)

a. <burst_align>

メモリバースト境界で分割の設定値：

0y000 = 任意アドレスの境界を越えられる。

0y001 = 32 ビートバースト境界で分割

0y010 = 64 ビートバースト境界で分割

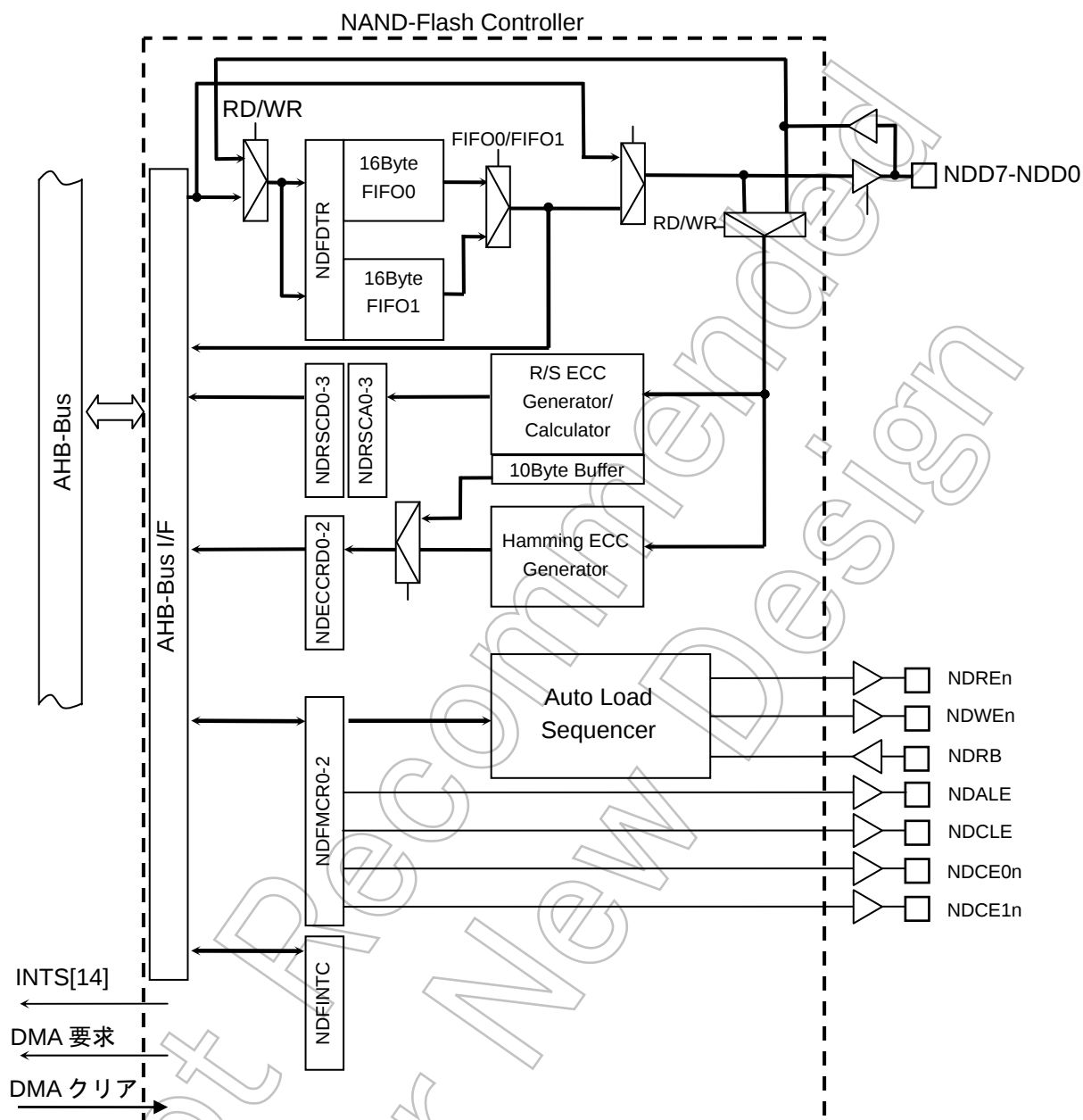
0y011 = 128 ビートバースト境界で分割

0y100 = 256 ビートバースト境界で分割

other = Reserved

- b. <bls>
bls タイミング設定 :
0y0 = chip select
0y1 = Reserved
- c. <wr_bl>
ライト用メモリのバースト長 :
0y000 = 1 ビート
0y001 = 4 ビート
other = Reserved
- d. <wr_sync>
メモリ動作モード :
0y0 = ライト非同期動作
0y1 = Reserved
- e. <rd_bl>
リード用メモリのバースト長 :
0y000 = 1 ビート
0y001 = 4 ビート
other = Reserved
- f. <rd_sync>
メモリ動作モード :
0y0 = リード非同期動作
0y1 = Reserved
- g. <mw>
レジスタのリセット値は設定条件に依存します。ブートのため、CS0 のメモリデータバスの幅を設定できます。

3.11.2 ブロック図



3.11.3 動作説明

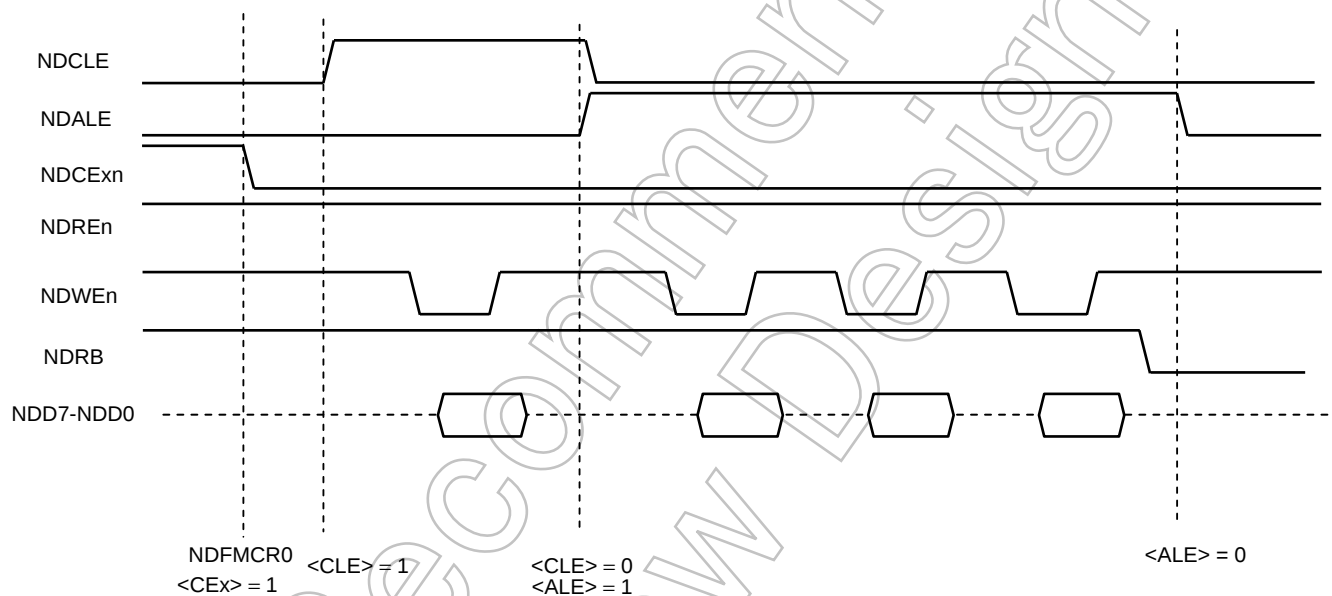
a. NAND-Flash メモリへのコマンド,アドレス設定

NAND-Flash メモリに対して ページリード,ページライトなどを実行するためのコマンド,アドレスをソフトウェアで設定します。

NDCExn, NDCLE, NDALE 端子については NDFMCR0<CEx,CLE,ALE>レジスタを設定する事で対応した外部端子が変化します。

NAND-Flash メモリへのリードライトは、NDFDTR レジスタのリードライトで実行されます。

A.C タイミングの設定は、NDFMCR2<SPLW[2:0],SPHW[2:0],SPLR[2:0],SPHR[2:0]>レジスタで調整が可能です。



b. NAND-Flash メモリからのページ単位でのデータリード, 内蔵 RAM へのデータライト

ここでは NDFC に内蔵する 2 つの 32bit×4word の FIFO, オートロード機能に加えて, 製品として内蔵している DMA コントローラを併用する事で CPU の負担を削減した高速のデータリード機能を実現します。

注) オートロード機能を使用したデータリードには必ず DMA 機能を使用してください。

なお、データリード時のオートロード機能は NDFMCR1<ALS>=1 の状態で NDRB 端子の立ち上がりエッジ検出後に自動的に開始するため、NAND-Flash へのコマンド設定後で アドレス設定前に 以下の(1),(2)を設定してください。

(1) DMA コントローラの任意のチャンネルに NDFC を割り当て、関係レジスタを設定

例として DMAC チャンネル 0 に割り当てた場合

DMACC0SrcAddr ← NDFDTR アドレス

DMACC0DestAddr ← 内蔵 RAM アドレス

DMACC0Control ← <Swidth[2:0]>= 0y010(32bit),
<Dwidth[2:0]>= 0y010(32bit),
<SBSIZE[2:0]>= 0y001(4 ビート),
<TransferSize[11:0]>= 0x80(512Byte/4Byte)

DMACC0Configuration ← <FlowCtrl[13:11]>= 0y010 (Peripheral to Memory),
<ITC>=1(DMA 終了割込みを許可)

(2) NDFMCR1<SELAL>に 0, <ALS>レジスタに 1 をライト

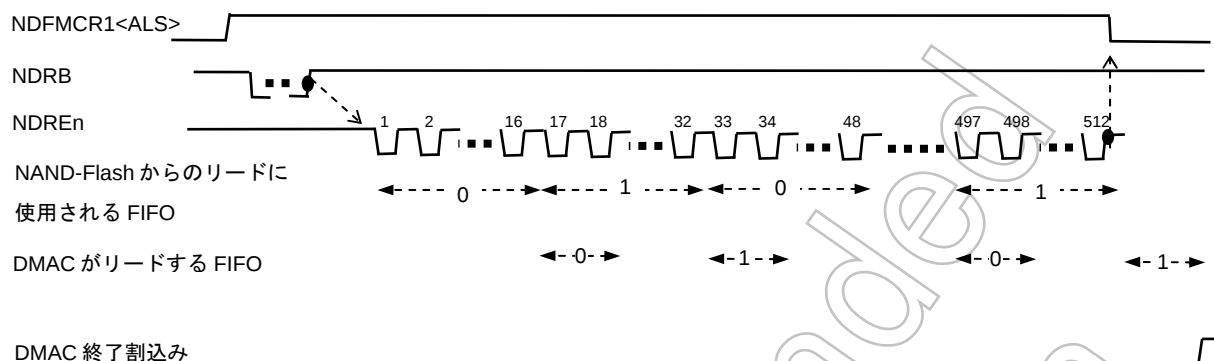
(2)により、NDFC は まず R/B 端子の立ち上がりエッジ検出を開始し、検出されたら 1 バイトのデータリードサイクルを起動します。リード毎に 16 バイトの FIFO0 にデータを格納すると共に NDFMCR1<ECCS>レジスタの設定に応じて ハミングコード, リードソロモンのどちらかのエラー演算器にデータを入力し ECC を生成します。

FIFO0 がフルになったら格納先を FIFO1 にスイッチし 引続きデータリードを続けます。また、FIFO0 がフルになった事で NDFC は DMAC に対して DMA 転送要求をアサートし FIFO0 のデータを内蔵 RAM へ転送する事を要求します。

この様に 2 つの 16 バイト FIFO を交互にスイッチし使用する事でデータリードを効率よく高速に実現できます。

合計 512 バイトのデータリードが終了すると DMAC が DMA 終了割込みをアサートしますのでその割込みを利用し CPU は次の処理を開始します。

以下に、DMA によるデータリードのタイミングイメージ図を示します。



なお、FIFO0,FIFO1 がともにフルの状態、DMAC が NDFC の FIFO からデータをリードできないとオートロード機能はその間停止します。

c. 内蔵 RAM から NAND-Flash メモリへのページ単位でのデータライト

NAND-Flash からのデータリードと同様にオートロード機能を使用したデータライトについて以下に説明します。実行方法は、以下の(1),(2)を設定してください。

注) オートロード機能を使用したデータライトには必ず DMA 機能を使用してください。

(1) DMA コントローラの任意のチャンネルに NDFC を割り当て、関係レジスタを設定

例として DMAC チャンネル 0 に割り当てた場合

DMACC0SrcAddr ← 内蔵 RAM アドレス

DMACC0DestAddr ← NDFDTR アドレス

DMACC0Control ← <Swidth[2:0]>= 0y010(32bit),
<Dwidth[2:0]>= 0y010(32bit),
<SBSIZE[2:0]>= 0y001(4 ビート),
<TransferSize[11:0]>= 0x80(512Byte/4Byte)

DMACC0Configuration ← <FlowCtrl>= 0y001(Memory to Peripheral),
<ITC>= 1(DMA 終了割込みを許可)

(2) NDFMCR1<SELAL>に 1,<ALS>レジスタに 1 をライト

(2)により、NDFC の FIFO0,FIFO1 は共に空のため DMA 要求をアサートし DMA コントローラが内蔵 RAM から FIFO0,FIFO1 への転送を実行します。

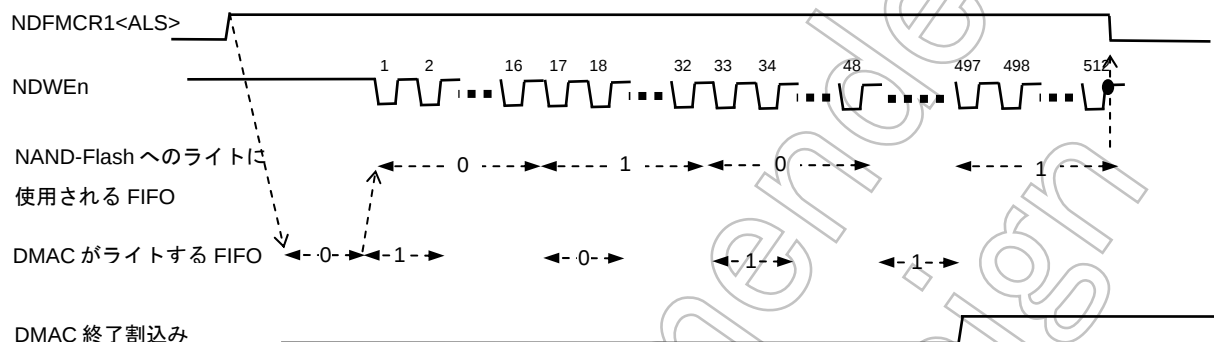
DMA コントローラから FIFO0 へのデータ転送が終了すると NDFC は FIFO0 のデータを使用し NAND-Flash メモリに対しデータライトサイクルを起動します。ライト毎に NDFMCR1<ECCS>レジスタの設定に応じて ハミングコード、リードソロモンのどちらかのエラー演算器にデータを入力し ECC を生成します。FIFO0 が空になったら取り出し先を FIFO1 にスイッチし引き続きデータライトを継続します。

また、FIFO0 が空になった事で NDFC は DMAC に対して DMA 転送要求をアサートし内蔵 RAM から FIFO0 へのデータ転送を要求します。

この様に二つの 16 バイト FIFO を交互にスイッチし使用する事でデータライトを効率よく高速に実現できます。

合計 512 バイトのデータライトが終了すると DMAC が DMA 終了割込みをアサートしますのでその割込みを利用し CPU は次の処理を開始します。

以下に、DMA によるデータライトのタイミングイメージ図を示します。



注) DMAC 終了割込みがアサートされた時点では NDFC のオートロード機能による NAND-Flash へのライトは終了していません。DMAC 終了割込み処理の中で<ALS>=0 であることを確認後、次の処理(ECC 処理)を実行してください。

なお、FIFO0, FIFO1 がともに空の状態、DMAC が NDFC の FIFO へデータをライトできないとオートロード機能はその間停止します。

d. 冗長エリアからの ECC リードまたはライト

オートロード機能は使用できません。コマンド、アドレス設定時同様ソフトウェアで NAND-Flash からのリードまたはライトを実行してください。

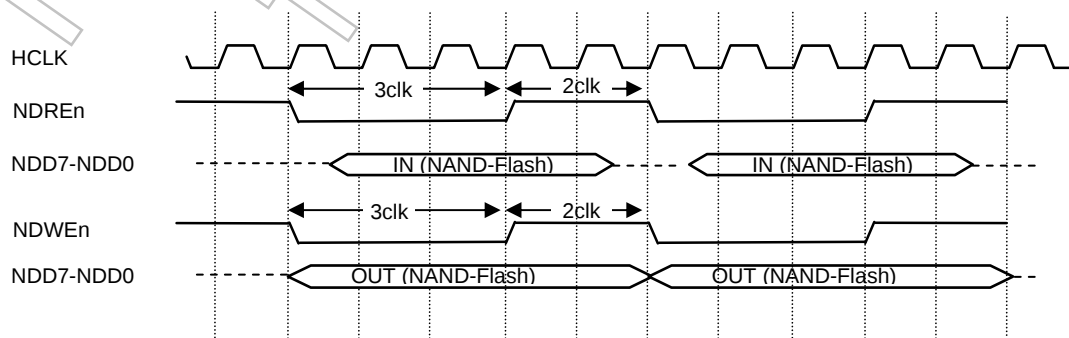
e. NDREn, NDWEn の波形調整機能

NDFDTR レジスタに対して、コマンド、アドレス設定、データのリード/ライトの動作を実行時に NDREn、NDWEn 端子の波形は NDFC が生成します。

この時、NDREn、NDWEn 端子の Low 幅と High 幅の調整が可能です。NDFC の動作クロック HCLK(最大 100MHz)と NAND-Flash のアクセスタイムなどの A.C スペックにあわせて調整してください。(詳細は電気的特性を参照ください)。

下記の例は、NDFMCR2<SPLW[2:0]>=0y011, <SPLR[2:0]>= 0y011、

NDFMCR2<SPHW[2:0]>= 0y010, <SPHR[2:0]>= 0y010 に設定し連続アクセスした場合のタイミング図です(ライト時にはデータのドライブ時間も延びます)。



3.11.4 ECC 制御

ここでは、ECC の制御に関する説明をします。NAND-Flash メモリでは、その特性上メモリ内にエラービットが含まれる可能性があります。そこで、ECC(Error Correction Code)を利用して、エラー訂正処理が必要です。

下記のフローは、ECC 制御の基本的な流れを示すフローチャートです。

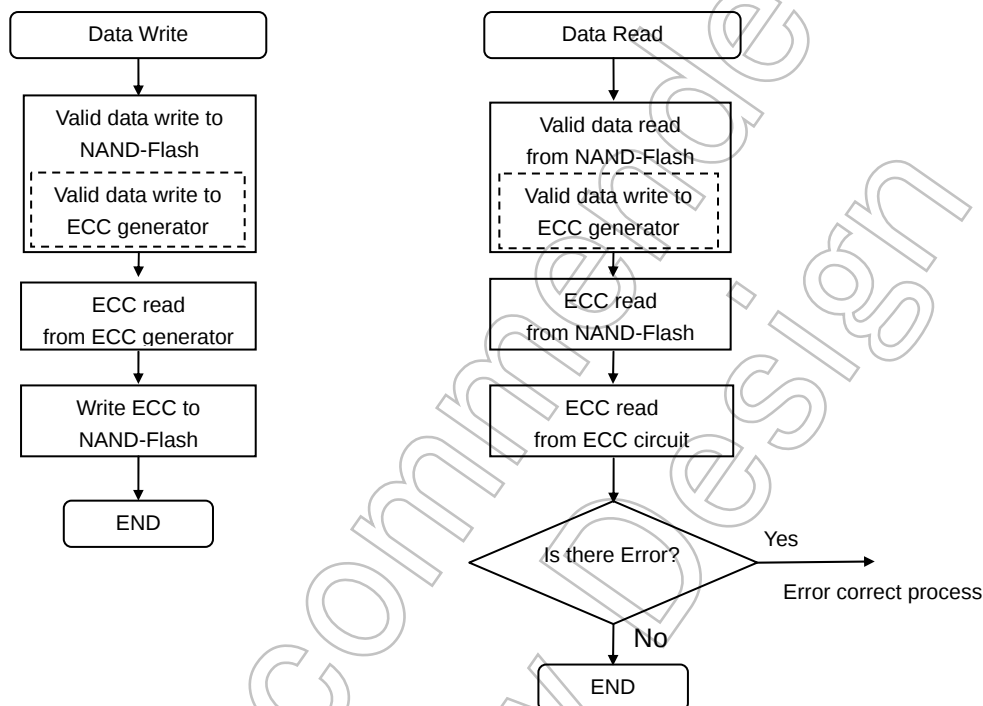


図 3.11.1 ECC 制御基本フロー

ライト時：

1. NAND-Flash メモリにデータライトする時は、実際の NAND-Flash メモリへライトするのと同時に、本回路内の ECC generator がライトしたデータの ECC を生成します。
2. 有効データとは別に、ECC を NAND-Flash メモリの冗長エリアにライトします。

リード時：

1. NAND-Flash メモリからリードする時は、ライト時と同様に、データのリードと同時に本回路内の ECC generator が ECC を発生させます。
2. NAND-Flash メモリから冗長エリアにライトしておいた ECC をリードし、ライト時とリード時の ECC を使って、エラービットを算出し、訂正を行います。

3.11.4.1 ハミングとリードソロモンの相違点

本回路では、2LC および 4LC に対応した、ECC 発生回路(ECC Generator)を内蔵しています。

2LC に対応したハミング ECC 計算では、有効データ 256Byte に対し、22bit の ECC を発生させ、1-bit/256byte のエラーを検知・訂正することが出来ます。エラービットの検出計算、訂正はソフトウェアにて対応する必要があります。

SmartMedia を使用する場合ハミングを使用します。

4LC に対応したリードソロモン ECC 計算では、有効データ 1Byte~512Byte に対し、80bit の ECC を発生させ、4-symbol/512byte のエラービットを検知・訂正することが出来ます。リードソロモン ECC も、ハミングと同様にエラービットの訂正はソフトウェアでの対応が必要ですが、エラービットの検出計算はハードウェアがサポートします。

下記にハミングとリードソロモンの違いを表にまとめています。

表 3.11.1 ハミングとリードソロモンの相違点

	Hamming	Reed Solomon
訂正可能な最大エラー数	1-bit	4-symbol (1-symbol = 8bit は全て訂正可能)
ECC のビット数	22 bit / 256 byte	80bit / ~512 byte
エラービットの検出方法	ソフトウェアにて対応します。	ハードウェアが検出します。
エラービットの訂正方法	ソフトウェアにて対応します。	ソフトウェアにて対応します。
エラービットの検出時間	ソフトウェアにて対応しますので、ソフトウェアの作成方法に依存します。	下記の表を参照ください。
その他	SmartMedia 対応	—

エラービット数	リードソロモン・ エラービット 検出時間 HCLK	備 考
4	813 (max)	検出までにかかるトータルのクロック数を表していますが、CPUがレジスタに対してリード/ライトする時間は含んでいません。
3	648 (max)	
2	358 (max)	
1	219 (max)	
0	1	

3.11.4.2 エラー訂正方法

Hamming ECC

- 1 ページが 512byte の有効データの場合 44bit の ECC が発生します。エラーの訂正処理は 256byte ずつ、(ECC : 22bit 分)行う必要があります。ここでは、22bit の ECC を使って、256byte の訂正処理の方法を説明しています。
- 大容量ページ(2048byte 等)の NAND-Flash メモリの場合は、複数回にわけて対応することが必要です。

1. 計算 ECC と冗長部の ECC^{注1)}を、それぞれ下位 2 バイトがラインパリティ(LPR15:0)、上位 1 バイトがカラムパリティ(CPR7:2(カラムパリティの有効データは上位 6 ビット))になるように、並び替えます。
2. 各々の ECC の、排他的論理和をとり XOR データを生成します。
3. XOR データが 0 なら ECC の一致が取れているので正常終了(エラーなし)。0 以外の場合、データ補正可能なエラーなのかをチェックします。
4. XOR データ内に ON ビットが 2 ビット以上無ければ、ECC 自体の 1 ビット異常と認識し、終了します(訂正不可)。
5. XOR データ内の有効データ、bit0~bit15, 18~23 までの 2 ビットずつのデータが 0y01 または 0y10 のいずれかになっていればデータ補正可能なエラーと判定し、データ補正に移行。0y00 または 0y11 が存在するならデータ補正不可能なエラーと判定し、異常終了します。

	訂正可能な XOR データ例	訂正不可能な XOR データ例
2 進数	10 01 10 00 カラムパリティ 10 10 01 10 ラインパリティ 01 01 10 10	10 11 10 00 カラムパリティ 10 10 01 10 ラインパリティ 01 01 10 10

6. データ補正は、XOR データのラインパリティからエラーのあった行情報、カラムパリティからビット情報を作成し、その 1 ビットを反転して終了します。

例) XOR データが 0y10_01_10_00_10_10_01_10_01_01_10_10 の場合

2 バイトのラインパリティを 1 バイトに変換する。(10→1、01→0)

6 ビットのカラムパリティを 3 ビットに変換する。(10→1、01→0)

ラインパリティ : 10 10 01 10 01 01 10 10
 ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
 1 1 0 1 0 0 1 1 = 0xD3

この場合、0xD3 番地にエラーがあります。ただし、このアドレスは 256 バイト中のアドレスであり絶対アドレスではありませんので訂正の際はご注意ください。

カラムパリティ : 10 01 10
 ↓ ↓ ↓
 1 0 1 = 5 →bit5 にエラー

0xD3 番地のビット 5 のデータを反転し、訂正終了。

Reed-Solomon ECC

- 本回路では、512byte までの有効データに対し 80bit の ECC を発生します。大容量ページ(2048byte 等)の NAND-Flash メモリの場合は複数回にわけて対応することが必要です。
- 基本的にエラー訂正のための計算作業は必要ありません。エラー検出が正常にされていれば、エラーアドレスとエラービットを参照するのみです。ただし、エラーアドレスの変換を必要とする場合があります。下記に例を用いて説明します。

1. NDRSCAn レジスタの示すエラーアドレスが 0x000~0x007 までの場合は、ECC 部にエラーが存在しますが、訂正の必要はありません。(ECC 部のエラー箇所の訂正は出来ません。しかし、ECC 部にエラーが存在した場合は ECC 部エラーを含め、トータル 4symbol のエラー訂正が本製品の能力になりますので、注意が必要です)
2. NDRSCAn レジスタの示すエラーアドレスが 0x008~0x207 の場合は、0x207 との差がエラーアドレスとなります。

例 1) NDRSCAn = 0x005, NDRSCDn = 0x04 = 0y0000_0100 の場合

エラーアドレスが 0x000~0x007 の範囲であるため、訂正不要。

(エラービットはビット 2 に存在しますが、訂正不要)

例 2) NDRSCAn = 0x083, NDRSCDn = 0x81 = 0y1000_0001 の場合

$0x207 - 0x083 = 0x184$ 番地のビット 7 とビット 0 のデータを反転し、訂正終了。

注) エラーアドレス(変換後のエラーアドレス) が、0x000~0x007 と計算された場合は、冗長エリア(ECC)内にエラービットが存在することを示していますが、この場合エラー訂正は必要ありません。ReedSolomon では、冗長エリアも含め 4symbol 以内のエラー数であれば、たとえエラービットが冗長エリアの ECC であっても、正確にエラービットの算出をします。

3.11.5 レジスタの説明

SFR のリストを以下に示します。

3.11.2 SFR 一覧表

base アドレス= 0xF201_0000

Register Name	Address (base+)	Description
NDFMCR0	0x0000	NAND-Flash Control Register 0
NDFMCR1	0x0004	NAND-Flash Control Register 1
NDFMCR2	0x0008	NAND-Flash Control Register 2
NDFINTC	0x000C	NAND-Flash Interrupt Control Register
NDFDTR	0x0010	NAND-Flash Data Register
NDECCRD0	0x0020	NAND-Flash ECC Read Register 0
NDECCRD1	0x0024	NAND-Flash ECC Read Register 1
NDECCRD2	0x0028	NAND-Flash ECC Read Register 2
NDRSCA0	0x0030	NAND-Flash Reed-Solomon Calculation Result Address Register 0
NDRSCD0	0x0034	NAND-Flash Reed-Solomon Calculation Result Data Register 0
NDRSCA1	0x0038	NAND-Flash Reed-Solomon Calculation Result Address Register 1
NDRSCD1	0x003C	NAND-Flash Reed-Solomon Calculation Result Data Register 1
NDRSCA2	0x0040	NAND-Flash Reed-Solomon Calculation Result Address Register 2
NDRSCD2	0x0044	NAND-Flash Reed-Solomon Calculation Result Data Register 2
NDRSCA3	0x0048	NAND-Flash Reed-Solomon Calculation Result Address Register 3
NDRSCD3	0x004C	NAND-Flash Reed-Solomon Calculation Result Data Register 3

1. NDFMCR0 (NAND-Flash Control Register 0)

Address = (0xF201_0000) + (0x0000)

Bit	Bit Symbol	Type	Reset Value	Description
[31:12]	–	–	Undefined	Read as undefined. Write as zero.
[11]	RSECCL	R/W	0y0	Reed Solomon ECC-Latch 0y0: Disable (80bit の F/F の更新を許可します) 0y1: Enable (80bit の F/F の更新を禁止します(保持します))
[10]	RSEDN	R/W	0y0	Reed-Solomon 動作選択 0y0: リード時 0y1: ライト時
[9]	RSESTA	WO	0y0	Reed-Solomon エラー計算開始 0y0: – 0y1: Start
[8]	RSECGW	R/W	0y0	Reed-Solomon ECC-Generator Write Enable 0y0: Disable 0y1: Enable
[7]	WE	R/W	0y0	Write-operation Enable 0y0: Disable 0y1: Enable
[6]	ALE	R/W	0y0	NDALE 端子制御 0y0: 0 出力 0y1: 1 出力
[5]	CLE	R/W	0y0	NDCLE 端子制御 0y0: 0 出力 0y1: 1 出力
[4]	CE0	R/W	0y0	NDCE0n 端子制御 0y0: 1 出力 0y1: 0 出力
[3]	CE1	R/W	0y0	NDCE1n 端子制御 0y0: 1 出力 0y1: 0 出力
[2]	ECCE	R/W	0y0	ECC 回路 Enable 0y0: Disable 0y1: Enable
[1]	BUSY	RO	0y0	リード時: NAND-Flash Status 0y0: Ready 0y1: Busy ライト時: 無効
[0]	ECCRST	WO	0y0	ECC 回路リセット 0y0: – 0y1: リセット

(個別説明)

a. <RSECCL>

リードソロモンの場合のみ、使用するビットです。ハミング使用時には 0 に設定してください。

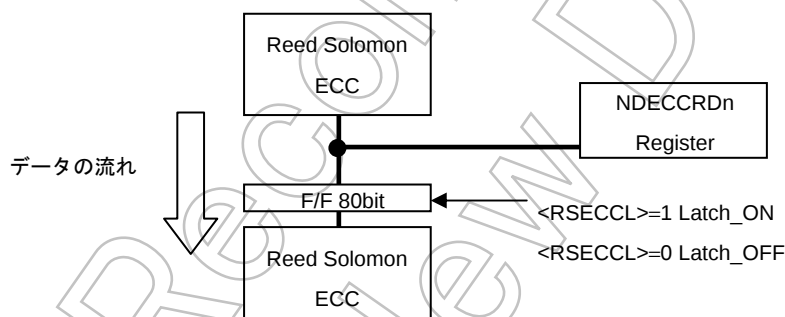
リードソロモンの回路は、2 種類の回路からなっており、ECC を発生する回路と、ECC から、エラーアドレスとエラービットを算出する回路から構成されています。

ECC の発生と、エラー算出がシリアルに行われる場合は特に問題ありませんが、ECC 発生とエラー計算を並列に行いたい場合は、エラー算出中には、エラー算出に使用する中間コードが変化しないように固定(ラッチ)しておく必要があります。

ライト時の ECC とリード時の ECC から生成された中間コードを使って、エラーアドレスとエラービットを計算しますが、その中間コードに、回路内部でラッチをかけるためのビットです。

<RSECCL>に 1 をライトすることで、中間コードにラッチがかかり、ECC 発生回路が ECC を更新させても、エラー算出回路には伝達されなくなり、エラーアドレス・エラービットを計算中も並行して、別ページの、ECC を発生させることが可能となります。この時の ECC 発生はライト時、リード時共に対応可能です。

<RSECCL>に 0 をライトすると、ラッチが外れて ECC 発生回路のデータがエラー算出回路に逐次更新されます。



b. <RSEDN>

リードソロモンの場合のみ、使用するビットです。ハミング使用時には 0 に設定してください。

ライト時には ECC を発生させるために、1: ライトを選択し、NDECCRDn からリードした ECC を NAND-Flash メモリの冗長エリアにライトします。またリード時には 0: リードを選択し、NAND-Flash メモリから有効データをリードし、さらに NAND-Flash の冗長エリアにライトされた ECC をリードすることで、エラーアドレスとエラービットの箇所を計算させるための中間コードを発生します。

c. <RSESTA>

リードソロモンの場合のみ、使用するビットです。

ライト時の ECC とリード時の ECC から生成された中間コードを使って、エラーアドレスとエラービットの箇所を計算します。1 をライトすると、計算を開始します。

d. <RSECGW>

リードソロモンの場合のみ、使用するビットです。ハミング使用時には 0 に設定してください。

有効データ部と ECC では、本回路内部でその処理方法が異なるため、ソフトウェアで有効データ部をリードする時と、ECC をリードする時を分けて管理する必要があります。

NAND-Flash メモリから有効データをリードする時には、0: Disable とし、NAND-Flash の冗長エリアにライトされた ECC をリードする時には、1: Enable とします。

注 1) DMAC を利用した有効データ部と ECC の連続リードは出来ません。有効データのリードが終了した後、一旦転送を停止させ、本ビットの設定を 0→1 に変更した後に ECC をリードしてください。

注 2) NAND-Flash メモリから ECC をリードした直後は、内部処理のために、システムクロック 20 クロックの期間は、NAND-Flash メモリへのアクセス(リード/ライト)とエラービットの算出計算が出来ません。ソフトウェアでのウェイト処理などが必要です。

e. <WE>

ハミング、リードソロモンともに、使用するビットです。NDWEn 端子の許可を制御します。不用意に NAND-Flash メモリに対して NDWEn 端子をアクティブにしないための保護レジスタです。

f. <ALE>, <CLE>, <CE0>, <CE1>

ハミング、リードソロモンともに、使用するビットです。NAND-Flash メモリの各端子を制御するビットです。

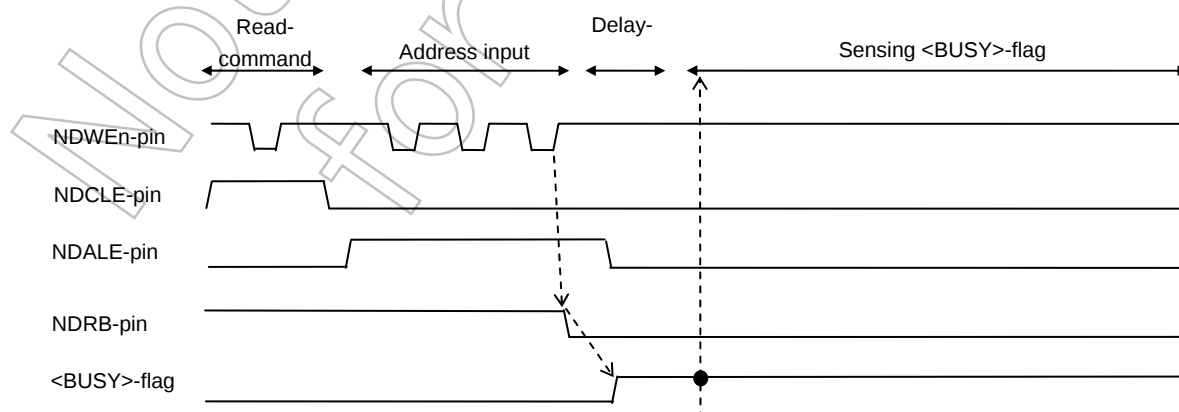
g. <ECCE>

ハミング、リードソロモンともに、使用するビットです。

ECC 回路の制御を行うビットです。ECC をリセットする際(<ECCRST>に 1 をライトする時)には、イネーブル状態(1)である必要があります。

h. <BUSY>

ハミング、リードソロモンともに、使用するビットです。NAND-Flash メモリの状態(NDRB 端子状態)を確認するビットです。BUSY 状態の時は 1 となり、RDY 状態の時は 0 となります。数クロックのノイズフィルタを NDFC に内蔵しているため、NDRB 端子の状態が変化した場合、同時に<BUSY>フラグは更新されません。



i. <ECCRST>

ハミング、リードソロモンともに、使用するビットです。

ハミング ECC のリセットの場合 NDFMCR1<ECCS> = 0、リードソロモン ECC をリセットする場合 NDFMCR1<ECCS> = 1 と設定し、一度 1 をライトすることで、回路内の ECC をリセットします(自動的に解除されます)。また、同時に NDECCRD_n レジスタの内容もクリアされます。

ECC をリセットする場合は、<ECCE>= 1 に設定してください。

Not Recommended
for New Design

2. NDFMCR1 (NAND-Flash Control Register 1)

Address = (0xF201_0000) + (0x0004)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined. Write as zero.
[15:12]	STATE[3:0]	RO	0y0000	リード時: Reed Solomon ECC-Calculator のステータス (計算処理を開始後、有効) 0y0000: 計算終了し Error は無し 0y0001: 計算終了し Error は 5-symbol 以上あり(訂正不可) 0y0010, 0y0011: 計算終了し Error は 4-symbol 以内(訂正可) 0y0100-0y1111: 計算中 ライト時: 無効
[11:10]	SERR[1:0]	RO	Undefined	リード時: Reed Solomon ECC-Calculator 結果の Error 個数 (計算処理を終了後、有効) 0y00: 1-address Err 0y01: 2-address Err 0y10: 3-address Err 0y11: 4-address Err ライト時: 無効
[9]	SELAL	R/W	0y0	Auto Load 機能選択 0y0: NAND-Flash からのデータリード 0y1: NAND-Flash へのデータライト
[8]	ALS	R/W	0y0	Auto Load 開始(ライト時) 0y0: – 0y1: Start Auto Load Status(リード時) 0y0: 実行前 または 実行後 0y1: 実行中
[7:2]	–	–	Undefined	Read as undefined. Write as zero.
[1]	ECCS	R/W	0y0	ECC 回路選択 0y0: Hamming 0y1: Reed-Solomon
[0]	–	–	Undefined	Read as undefined. Write as zero.

(個別説明)

a. <STATE[3:0]>,<SERR[1:0]>

リードソロモンの場合のみ、使用するビットです。ハミング使用時には、意味を持ちません。
エラーアドレスとエラービットの計算結果の状態を示すフラグです。

b. <SELAL>

ハミング、リードソロモンともに、使用するビットです。

オートロード機能を実行する際に NAND-Flash に対してのデータリード、ライトを選択します。

c. <ALS>

ハミング、リードソロモンともに、使用するビットです。

NAND-Flash に対してのデータリード、ライトを DMAC を併用し高速に転送する機能の制御レジスタです。1 をライトすると 16 バイト FIFO0/FIFO1 がイネーブルになります。

また、リードする事で オートロード機能のステータスを知る事ができます。(オートロード機能にて 512 バイトのリードまたはライトを実行すると 0 にクリアされるようになっています。)

d. <ECCS>

ハミング/リードソロモンの切り替えビットです。0 でハミング、1 でリードソロモンの設定となります。ECC をリセットする場合にも選択する必要があります。

3. NDFMCR2 (NAND-Flash Control Register 2)

Address = (0xf201_0000) + (0x0008)

Bit	Bit Symbol	Type	Reset Value	Description
[31:15]	–	–	Undefined	Read as undefined. Write as zero.
[14:12]	SPLW[2:0]	R/W	0y000	NDWEn Low-pulse width setting 0y000: Reserved 0y001: 1-cycle of HCLK 0y010: 2-cycle of HCLK 0y011: 3-cycle of HCLK 0y100: 4-cycle of HCLK 0y101: 5-cycle of HCLK 0y110-0y111: Reserved
[11]	–	–	Undefined	Read as undefined. Write as zero.
[10:8]	SPHW[2:0]	R/W	0y000	NDWEn High-pulse width setting 0y000: Reserved 0y001: 1-cycle of HCLK 0y010: 2-cycle of HCLK 0y011: 3-cycle of HCLK 0y100: 4-cycle of HCLK 0y101: 5-cycle of HCLK 0y110-0y111: Reserved
[7]	–	–	Undefined	Read as undefined. Write as zero.
[6:4]	SPLR[2:0]	R/W	0y000	NDREn Low-pulse width setting 0y000: Reserved 0y001: 1-cycle of HCLK 0y010: 2-cycle of HCLK 0y011: 3-cycle of HCLK 0y100: 4-cycle of HCLK 0y101: 5-cycle of HCLK 0y110-0y111: Reserved
[3]	–	–	Undefined	Read as undefined. Write as zero.
[2:0]	SPHR[2:0]	R/W	0y000	NDREn High-pulse width setting 0y000: Reserved 0y001: 1-cycle of HCLK 0y010: 2-cycle of HCLK 0y011: 3-cycle of HCLK 0y100: 4-cycle of HCLK 0y101: 5-cycle of HCLK 0y110-0y111: Reserved

(個別説明)

a. <SPLW[2:0]>, <SPHW[2:0]>, <SPLR[2:0]>, <SPHR[2:0]>

NDREn, NDWEn 端子の Low パルス, High パルスの幅を設定するレジスタです。

設定値 × HCLK の周期分のパルス幅になります。0y000, 0y110, 0y111 は設定禁止です。

4. NDFINTC (NAND-Flash Interrupt Control Register)

Address = (0xF201_0000) + (0x000C)

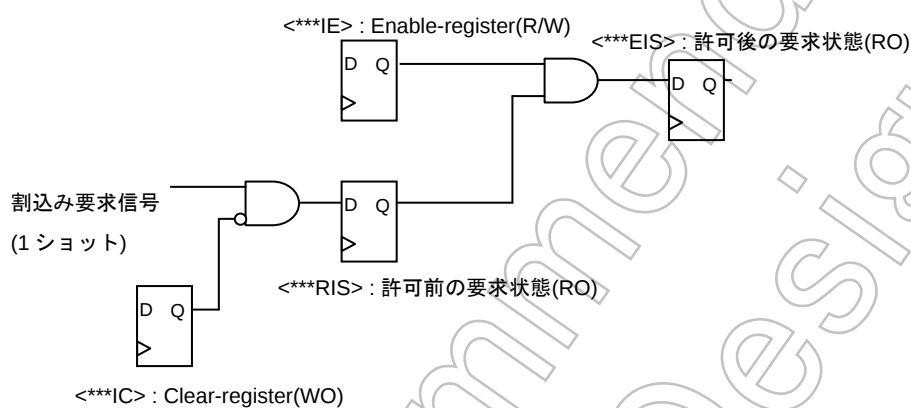
Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	—	—	Undefined	Read as undefined. Write as zero.
[7]	RSEIC	WO	0y0	(R/S Calculator End Interrupt) Clear register 0y0: — 0y1: クリア
[6]	RSEEIS	RO	0y0	リード時: (R/S Calculator End Interrupt) 許可後の割込み要求状態 0y0: 割込み要求なし 0y1: 割込み要求あり ライト時: 無効
[5]	RSERIS	RO	0y0	リード時: (R/S Calculator End Interrupt) 許可前の割込み要求状態 0y0: 割込み要求なし 0y1: 割込み要求あり ライト時: 無効
[4]	RSEIE	R/W	0y0	(R/S Calculator End Interrupt) 許可 register 0y0: 割込み要求禁止 0y1: 割込み要求許可
[3]	RDYIC	WO	0y0	(NAND-Flash Ready Interrupt) Clear register 0y0: — 0y1: クリア
[2]	RDYEIS	RO	0y0	リード時: (NAND-Flash Ready Interrupt) 許可後の割込み要求状態 0y0: 割込み要求なし 0y1: 割込み要求あり ライト時: 無効
[1]	RDYRIS	RO	0y0	リード時: (NAND-Flash Ready Interrupt) 許可前の割込み要求状態 0y0: 割込み要求なし 0y1: 割込み要求あり ライト時: 無効
[0]	RDYIE	R/W	0y0	(NAND-Flash Ready Interrupt) 許可 register 0y0: 割込み要求禁止 0y1: 割込み要求許可

(個別説明)

a. <xxxIC>, <xxxEIS>, <xxxRIS>, <xxxIE>

NDRB 端子の状態をモニタし Busy 状態から Ready 状態へ遷移した際に発生する READY 割込み, リードソロモンアドレス, データ計算が終了時に発生するリードソロモン計算終了割込みの 2 つの割込みに対応する各 4 ビットのレジスタです。NDFC はこれらの 2 つの割込み要求を論理 OR した 1 つの割込み要求を割込みコントローラへアサートしますので割込み処理の中でこれらのレジスタを確認し 各々の割込みソースに対応した処理が必要になります。

各レジスタの関係を以下の図で表します。



5. NDFDTR (NAND-Flash Data Register)

Address = (0xF201_0000) + (0x0010)

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	DATA[31:0]	R/W	Undefined	データレジスタ

(個別説明)

a. <DATA[31:0]>

NAND-Flash メモリヘデータをリード,ライトする場合やコマンド,アドレスを設定する際に NDFDTR レジスタをアクセスします。

このレジスタにデータをライトすると、NAND-Flash メモリヘデータがライトされ、このレジスタをリードすると NAND-Flash からデータをリードします。DMA 動作で、1 ワード転送を利用できるようになっています。

注) このレジスタは R/W 可能なレジスタですが、内部に F/F は存在しません。ライト時とリード時の動作が異なるため、ライト後、リードしてもライトデータは保持されていません。

6. NDECCRD0 (NAND-Flash ECC Read Register 0)

Address = (0xF201_0000) + (0x0020)

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	CODE0[31:0]	RO	0x00000000	ECC 格納レジスタ

7. NDECCRD1 (NAND-Flash ECC Read Register 1)

Address = (0xF201_0000) + (0x0024)

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	CODE1[31:0]	RO	0x00000000	ECC 格納レジスタ

8. NDECCRD2 (NAND-Flash ECC Read Register 2)

Address = (0xF201_0000) + (0x0028)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined.
[15:0]	CODE2[15:0]	RO	0x0000	ECC 格納レジスタ

(個別説明)

a. <CODE0[31:0]>, <CODE1[31:0]>, <CODE2[15:0]>

本回路内で計算した、ECC をリードするためのレジスタです。

有効データのリード/ライトが終了し、NDFMCR0<ECCE>に 0 をライトした時点で、本レジスタには ECC が準備されています(NDFMCR0<ECCE>が 1 から 0 に変化したときに、最新の ECC に更新されます)。

ハミングの場合、22bit/256Byte、リードシロモンの場合 80bit/512Byte となります。

80 ビット分のレジスタが用意されており、32 ビット幅のレジスタが 3 つの構成となっています。

格納フォーマットは、以下の表を参照願います。

注) ECC をリードする場合、NDFMCR0<ECCE>を 0 にした後に ECC データをリードしてください。

NDFMCR0<ECCE>が 1 から 0 に変化したときに最新の ECC に更新されます。また、NDFMCR0<ECCRST>にて、ECC をリセットしても、ECC Generator 内部の ECC がクリアされるだけで、本レジスタの内容はリセットされません。

レジスタ名	ハミング	リードソロモン
NDECCRD0<15:0>	[15:0] ラインパリティ (前半 256 バイト分)	[15:0] R/S ECC 79:64
NDECCRD0<31:16>	[23:18] カラムパリティ (前半 256 バイト分)	[31:16] R/S ECC 63:48
NDECCRD1<15:0>	[15:0] ラインパリティ (後半 256 バイト分)	[15:0] R/S ECC 47:32
NDECCRD1<31:16>	[23:18] カラムパリティ (後半 256 バイト分)	[31:16] R/S ECC 31:16
NDECCRD2<15:0>	未使用	[15:0] R/S ECC 15:0

ECC を NAND-Flash メモリの冗長エリアにライトする方法を、下記の表に示します。

ハミングの場合で、SmartMedia を利用する際は、SmartMedia の物理フォーマット仕様で冗長エリアのアドレスが規定されています。詳細は SmartMedia の物理フォーマット仕様を参照ください。

レジスタ名	リードソロモン	NAND-Flash アドレス
NDECCRD0	[31:0] R/S ECC 79:48	上位 8bit [79:72] → 518 番地 8bit [71:64] → 519 番地 8bit [63:56] → 520 番地 下位 8bit [55:48] → 521 番地
NDECCRD1	[31:0] R/S ECC 47:16	上位 8bit [47:40] → 522 番地 8bit [39:32] → 523 番地 8bit [31:24] → 524 番地 下位 8bit [23:16] → 525 番地
NDECCRD2	[15:0] R/S ECC 15:0	上位 8bit [15:8] → 526 番地 下位 8bit [7:0] → 527 番地

9. NDRSCA0 (NAND-Flash Reed-Solomon Calculation Result Address Register 0)

Address = (0xF201_0000) + (0x0030)

Bit	Bit Symbol	Type	Reset Value	Description
[31:10]	–	–	Undefined	Read as undefined.
[9:0]	AL	RO	0x000	R/S 用エラー0 アドレス格納レジスタ

10. NDRSCD0 (NAND-Flash Reed-Solomon Calculation Result Data Register 0)

Address = (0xF201_0000) + (0x0034)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined.
[7:0]	DATA	RO	0x00	R/S 用エラー0 データ格納レジスタ

(個別説明)

a. <AL>,<DATA>

エラーが 1-address のみの場合 NDRSCA0 レジスタにエラーアドレス、NDRSCD0 レジスタにエラーデータが格納され、エラーが 2-address の場合、NDRSCA0、NDRSCA1 レジスタにエラーアドレス、NDRSCD0、NDRSCD1 レジスタにエラーデータが格納されます。このように、エラービットが 4-address 以内の場合に有効なエラーアドレスが格納されます。

エラーアドレスの数は、NDFMCR1<SEER[1:0]>を確認ください。

- NDRSCAx (NAND-Flash Reed-Solomon Calculation Result Address Register x) (x = 0~3)
- NDRSCDx (NAND-Flash Reed-Solomon Calculation Result Data Register x) (x = 0~3)

上記のレジスタ群について、構造および説明はそれぞれNDRSCA0、NDRSCD0 と同様のため、NDRSCA0、NDRSCD0 の説明を参照してください。またレジスタ名およびアドレスは 3.11.2 SFR一覧表を参照してください。

3.11.6 NAND Flashへのアクセス例

下記に NAND Flash メモリのアクセス例を示します。本例は設定の手順を示したものであり、その動作を保証するものではありません。プログラムの作成時の目安としてご使用ください。

(1) ページライト (2LC タイプ)

——— メインプログラム ———

```

;
; ***** Initialize for NDFC *****
;         condition: 8bit-bus, CE0, SLC, 512Byte/Page, Hamming
;
;
NDFMCR0      ←          0x0000_0010          ; NDCE0n 端子=0, ECC Disable
NDFMCR1      ←          0x0000_0000          ; ECC=Hamming
NDFMCR2      ←          0x0000_3343          ; NDWEn  L=3clk,H=3clk,
;                                     ; NDREn  L=4clk,H=3clk
NDFINTC      ←          0x0000_0000          ; ALL Interrupt Disable
;
; ***** Setting Command, Address to NAND-Flash *****
;
;
NDFMCR0      ←          0x0000_00b0          ; NDCE0n 端子=0, NDCLE=1, NDALE=0
NDFDTR      ←          0x80                  ; Write Command(1st cycle of Page-Program)
NDFMCR0      ←          0x0000_00d0          ; NDCE0n 端子=0, NDCLE=0, NDALE=1
NDFDTR      ←          0x??                  ; Write Address (n-times)
NDFMCR0      ←          0x0000_0095          ; NDCE0n 端子=0, NDCLE=0, NDALE=0
;                                     ; ECC Enable and Reset
;
; ***** Writing 512Byte Valid data *****
;
;
DMAC, INTC へ ライトデータ 512 バイトのオートロード機能対応を設定(詳細は省略)
;                                     (INTTC 割込み許可含む)
NDFMCR1      ←          0x0000_0300          ; <SELALS>=1, Start Autoload
;

```

—— INTTC 割込み処理プログラム ——

NDFMCR1 → リード、チェック ; <ALS>=0(終了)である事を確認。終了してなければポーリング

NDFMCR0 ← 0x0000_0090 ; ECC Disable

メインプログラムへの復帰

——— メインプログラム ———

```

; ***** Reading ECC from NDFC *****
NDECCRD0      →      リード      ; 前半 256 バイト用 ECC
NDECCRD1      →      リード      ; 後半 256 バイト用 ECC

; ***** Writing Dummy data & ECC *****
;
;
NDFDTR        ←      0x??        ; Write Dummy data(1 バイトを 8times)
NDFDTR        ←      ECC         ; Write ECC (1 バイトを 3times)
;
;                               Write to D520: LPR7:0      For last half 256byte
;                               Write to D521: LPR15:8     For last half 256byte
;                               Write to D522: CPR5:0+11b  For last half 256byte
;
;
NDFDTR        ←      0x??        ; Write Dummy data(1 バイトを 2times)
NDFDTR        ←      ECC         ; Write ECC (1 バイトを 3times)
;
;                               Write to D525: LPR7:0      For first half 256byte
;                               Write to D526: LPR15:8     For first half 256byte
;                               Write to D527: CPR5:0+11b  For first half 256byte

```

```
; ***** Set Page program command*****
```

```
;
```

```
NDFMCR0 ← 0x0000_00b0 ; NDCEn 端子=0, NDCLE=1, NDALE=0
```

```
NDFDTR ← 0x10 ; Write Command (2nd cycle of Page-Program)
```

```
NDFMCR0 ← 0x0000_0010 ; NDCEn 端子=0, NDCLE=0, NDALE=0
```

```
; ***** Wait till Page-Program End *****
```

```
;
```

```
; ページプログラムの終了を待ちます。確認方法としては、リードステータスコマンドをライトしてステータスを  
; NDD7-0 端子からリードするポーリング方法と、NDRB 端子の立ち上がりエッジ検出による Ready 割込みを  
; 使用する方法があります。ここでは Ready 割込みを使用する場合を記述します。
```

```
;
```

```
NDFINTC ← 0x0000_0009 ; RDY 割込みのクリア許可
```

INTC へ NDFC 割込みの許可設定(詳細は省略)

—— INTNDFC 割込み処理プログラム ——

終了処理

メインプログラムへの復帰

(2) ページリード(2LC タイプ)

—— メインプログラム ——

```

;
; ***** Initialize for NDFC *****
; condition: 8bit-bus, CE0, SLC, 512Byte/Page, Hamming
;
;
NDFMCR0      ←    0x0000_0010      ; NDCEn 端子=0, ECC Disable
NDFMCR1      ←    0x0000_0000      ; ECC=Hamming
NDFMCR2      ←    0x0000_3343      ; NDWEn L=3clk,H=3clk,
; NDREn L=4clk,H=3clk
NDFINTC      ←    0x0000_0000      ; ALL Interrupt Disable

; ***** Setting Command, Address to NAND-Flash *****
;
;
NDFMCR0      ←    0x0000_00b0      ; NDCEn 端子=0, NDCLE=1, NDALE=0
NDFDTR       ←    0x00              ; Write Command (1st cycle of Page-Read)

; ***** Reading 512Byte Valid data *****
DMAC,INTC へ リードデータ 512 バイトのオートロード機能対応を設定(詳細は省略)
; (INTTC 割込み許可含む)
NDFMCR1      ←    0x0000_0100      ; <SELALS>=0, Start Autoload

NDFMCR0      ←    0x0000_00d0      ; NDCEn 端子=0, NDCLE=0, NDALE=1
NDFDTR       ←    0x??              ; Write Address (n-times)
NDFMCR0      ←    0x0000_00b0      ; NDCEn 端子=0, NDCLE=1, NDALE=0
NDFDTR       ←    0x030             ; Read Command (2nd cycle of Page-Read)
NDFMCR0      ←    0x0000_0015      ; NDCEn 端子=0, NDCLE=0, NDALE=0
; ECC Enable and Reset

```

—— INTTC 割込み処理プログラム ——

```

NDFMCR0      ←    0x0000_0010      ; ECC Disable

; ***** Reading Dummy data & ECC from NAND-Flash *****
;
;
NDFDTR       →    リード            ; Read Dummy data (1 バイトを 8times)
NDFDTR       →    リード            ; Read ECC (1 バイトを 3times)
NDFDTR       →    リード            ; Read Dummy data (1 バイトを 2times)
NDFDTR       →    リード            ; Read ECC (1 バイトを 3times)
;
; ***** Reading ECC from NDFC *****
;
;
NDECCRD0     →    リード            ; 前半 256 バイト用 ECC
NDECCRD1     →    リード            ; 後半 256 バイト用 ECC

```

ソフトウェア処理

リード時に発生したECCと、メモリからリードしたECCを比較し、エラーが発生した場合には、エラールーチン処理を行い、データ補正を行います。詳細は 3.11.4.2 の『エラー訂正方法』を参照ください。

メインプログラムへの復帰

(3) ページライト (4LC タイプ)

—— メインプログラム ——

```

;
; ***** Initialize for NDFC *****
;
;           condition: 8bit-bus, CE0, MLC, 512Byte/Page, Reed Solomon
;
;
NDFMCR0    ←    0x0000_0410    ; NDCEn 端子=0, ECC Disable
NDFMCR1    ←    0x0000_0002    ; ECC=Reed Solomon
NDFMCR2    ←    0x0000_3343    ; NDWEn L=3clk,H=3clk,
; NDWEn L=4clk,H=3clk
NDFINTC    ←    0x0000_0000    ; ALL Interrupt Disable

; ***** Setting Command, Address to NAND-Flash *****
;
;
NDFMCR0    ←    0x0000_04b0    ; NDCEn 端子=0, NDCLE=1, NDALE=0
NDFDTR     ←    0x80           ; Write Command (1st cycle of Page-Program)
NDFMCR0    ←    0x0000_04d0    ; NDCEn 端子=0, NDCLE=0, NDALE=1
NDFDTR     ←    0x??          ; Write Address (n-times)
NDFMCR0    ←    0x0000_0495    ; NDCEn 端子=0, NDCLE=0, NDALE=0
; ECC Enable and Reset

```

```

; ***** Writing 512Byte Valid data *****

```

DMAC,INTC ヘライトデータ 512 バイトのオートロード機能対応を設定(詳細は省略)
(INTTC 割込み許可含む)

```

NDFMCR1    ←    0x0000_0302    ; <SELALS>=1, Start Auto-Load
;

```

—— INTTC 割込み処理プログラム ——

```

NDFMCR1    →    リード,チェック    ; <ALS>=0(終了)である事を確認。終了してなければポーリング
NDFMCR0    ←    0x0000_0490    ; ECC Disable

```

メインプログラムへの復帰

—— メインプログラム ——

```

; ***** Reading ECC from NDFC *****

```

```

NDECCRD0   →    リード            ; ECC(1/3)
NDECCRD1   →    リード            ; ECC(2/3)
NDECCRD2   →    リード            ; ECC(3/3)

```

```

; ***** Writing Dummy data & ECC *****
;

```

```

NDFDTR     ←    ECC              ; Write ECC (1 バイトを 10times)
NDFDTR     ←    0x??            ; Write Dummy data (1 バイトを 6times)

```

```

; ***** Set Page program command*****
;

```

```

NDFMCR0    ←    0x0000_04b0    ; NDCEn 端子=0, NDCLE=1, NDALE=0
NDFDTR     ←    0x10           ; Write Command (2nd cycle of Page-Program)
NDFMCR0    ←    0x0000_0410    ; NDCEn 端子=0, NDCLE=0, NDALE=0

```

```

; ***** Wait till Page-Program End *****
;

```

ページプログラムの終了を待ちます。確認方法としてはリードステータスコマンドをライトしてステータスを
; NDD7-0 端子からリードするポーリング方法と、NDRB 端子の立ち上がりエッジ検出による Ready 割込みを
; 使用する方法があります。ここでは Ready 割込みを使用する場合を記述します。

```

NDFINTC    ←    0x0000_0009    ; RDY 割込みのクリア,許可

```

INTC へ NDFC 割込みの許可設定(詳細は省略)

—— INTNDFC 割込み処理プログラム ——

終了処理

メインプログラムへの復帰

(4) ページリード(4LC タイプ)

—— メインプログラム ——

```

;
; ***** Initialize for NDFC *****
;
;           condition: 8bit-bus, CE0, MLC, 512Byte/Page, Reed Solomon
;
;
NDFMCR0      ←    0x0000_0010      ; NDCEn 端子=0, ECC Disable
NDFMCR1      ←    0x0000_0002      ; ECC= Reed Solomon
NDFMCR2      ←    0x0000_3343      ; NEWEn L=3clk,H=3clk,
; NDWEn L=4clk,H=3clk
NDFINTC      ←    0x0000_0000      ; ALL Interrupt Disable

; ***** Setting Command, Address to NAND-Flash *****
;
;
NDFMCR0      ←    0x0000_00b0      ; NDCEn 端子=0, NDCLE=1, NDALE=0
NDFDTR       ←    0x00             ; Write Command (1st cycle of Page-Read)

; ***** Reading 512Byte Valid data *****
DMAC,INTC へ リードデータ 512 バイトのオートロード機能対応を設定(詳細は省略)
              (INTC 割込み許可含む)
NDFMCR1      ←    0x0000_0102      ; <SELALS>=0, Start Autoload

NDFMCR0      ←    0x0000_00d0      ; NDCEn 端子=0, NDCLE=0, NDALE=1
NDFDTR       ←    0x??             ; Write Address (n-times)
NDFMCR0      ←    0x0000_00b0      ; NDCEn 端子=0, NDCLE=1, NDALE=0
NDFDTR       ←    0x030            ; Read Command (2nd cycle of Page-Read)
NDFMCR0      ←    0x0000_0015      ; NDCEn 端子=0, NDCLE=0, NDALE=0
; ECC Enable and Reset, <RSECGW>=0

```

—— INTTC 割込み処理プログラム ——

```

NDFMCR0      ←    0x0000_0114      ; ECC-Enable, <RSECGW>=1
メインプログラムへの復帰

```

—— メインプログラム ——

```

; ***** Reading Dummy data & ECC from NAND-Flash *****
;
;
NDFDTR       →    リード            ; Read ECC (1 バイトを 10times)
;
; ***** Calculation Error Address and Data *****
;
;
NDFINTC      ←    0x0000_0090      ; R/S Calculation End 割込みのクリア,許可
INTC へ NDFC 割込みの許可設定(詳細は省略)
NDFMCR0      ←    0x0000_0310      ; ECC-Disable, <RSECGW>=1, <RSESTA>=1

```

—— INTNDFC 割込み処理プログラム ——

```

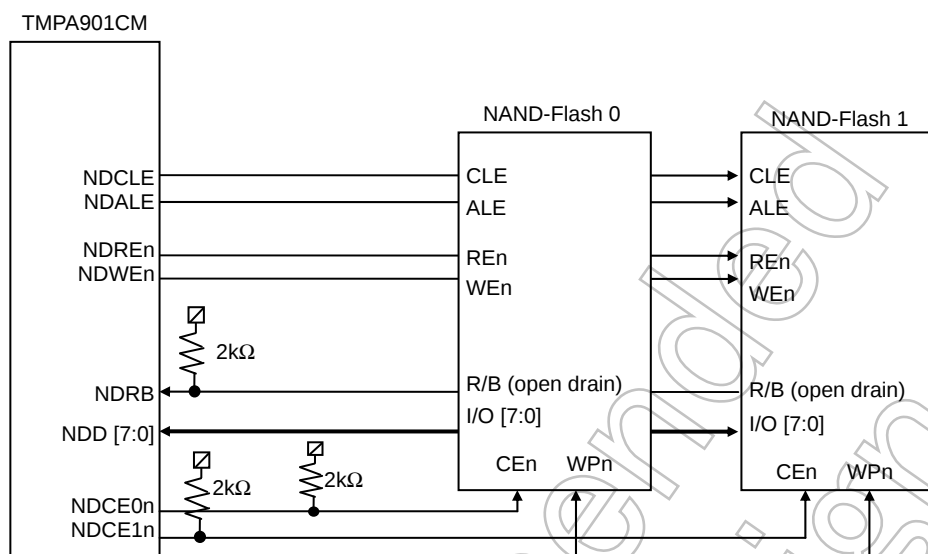
NDFMCR1      →    リード, チェック      ; <STATE>,<SERR>フラグをチェック
ソフトウェア処理

```

エラーが発生した場合には、エラールーチン処理を行い、データ補正を行います。
 詳細は 3.11.4.2 の『エラー訂正方法』を参照ください。

メインプログラムへの復帰

3.11.7 NAND-Flash接続例



注 1) NDRB、NDCE[1:0]n 端子のプルアップ抵抗値については、使用する NAND-Flash メモリと、基板容量等によって適切な値を設定する必要があります。(標準設定例: 2kΩ)

注 2) NAND-Flash の WPn (Write Protect) 端子については、サポートしていません。必要な場合、外部で準備して下さい。

図 3.11.2 NAND-Flash 接続例

3.12 16 ビットタイマ／PWM

3.12.1 機能概要

多機能 16 ビットタイマを 6 チャンネル内蔵しています。次の 2 つの動作モードを持っています。

- 1) Free-Running モード
- 2) Periodic タイマモード

PWM 機能対応

また、回路は 2 チャンネルを 1 ブロックで、3 ブロックの構成となっており、Block1 と Block2 は PWM(Pulse Width Modulation)出力に対応しています。

	Block1		Block2		Block3	
	Timer 0	Timer 1	Timer 2	Timer 3	Timer 4	Timer 5
Free-Running モード	○	○	○	○	○	○
Periodic タイマモード	○	○	○	○	○	○
PWM 機能	○	N/A	○	N/A	N/A	
	PWM0OUT (PC3)	×	PWM2OUT (PC4)	×	×	×
割り込み要因信号	INTS[2]		INTS[3]		INTS[4]	

本仕様書では、すべて同一(PWM 機能、割り込み要因を除く)の仕様となっているため、ブロック 1 の回路のみを説明しています。

3.12.2 ブロック図

2 チャンネルのタイマ回路を内蔵するタイマブロックはプログラム可能な 16bitフリーラン・デクリメントカウンタによって構成されています。TIMCLK入力はカウンタ動作に使用されます。このクロックは、内部システムクロックの 2 分周のクロック ($f_{PCLK}/2$)、または $f_s(32.768kHz)$ を選択することが可能です。タイマのブロック図を図 3.12.1に示します。

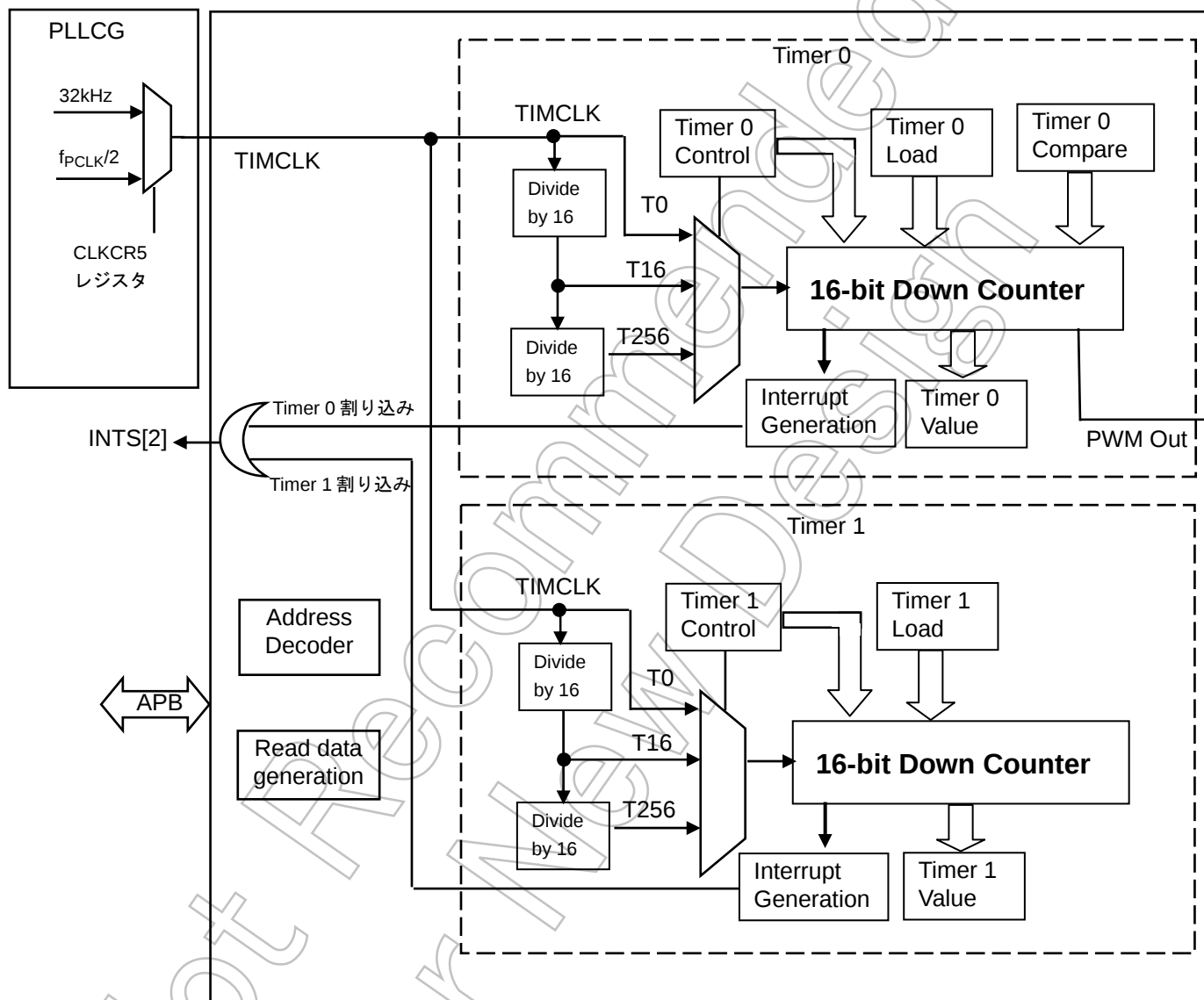


図 3.12.1 タイマブロック図 (Timer 0 および Timer 1)

タイマのクロック (TIMCLK) はプリスケールユニットで生成されます。

T0: $f_{PCLK}/2$

T16: $f_{PCLK}/2$ の 16 分周。4bit の prescale によって生成されます。

T256: $f_{PCLK}/2$ の 256 分周。8bit の prescale によって生成されます。

3.12.3 動作説明

下記の説明は Timer 0 の設定例です。他のチャンネルに関しても動作は Timer 0 と同様です。

1) Free-Running モード

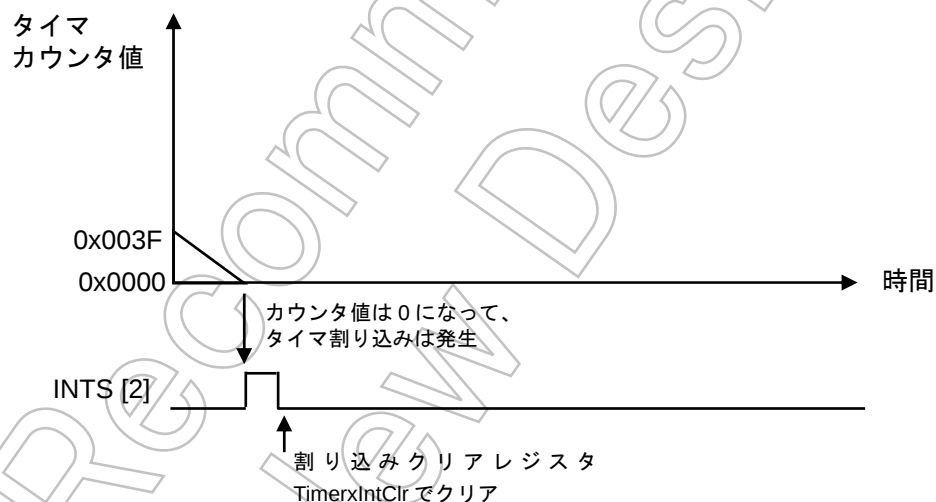
タイマをスタートすると設定されたカウンタ値からデクリメントを開始します。カウンタの値が 0 になると割り込みが発生します。

One-shot-operation (Timer0Control<TIM0OSCTL> = 1) の場合は、一度だけ割り込みを発生します。

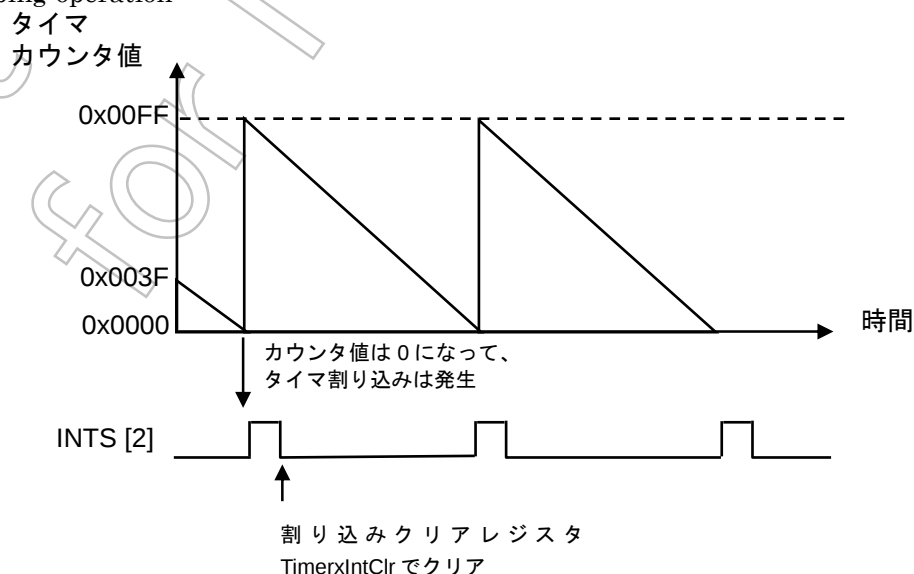
Wrapping-operation (Timer0Control<TIM0OSCTL> = 0) の場合は、カウンタの最大値をリロードし、デクリメントを継続します。8 ビットカウンタの場合は最大 0x000000FF, 16 ビットカウンタの場合は最大 0x0000FFFF になります。

下の動作概略図は 8 ビットカウンタ、タイマの値を 0x0000003F に設定した例を示しています。

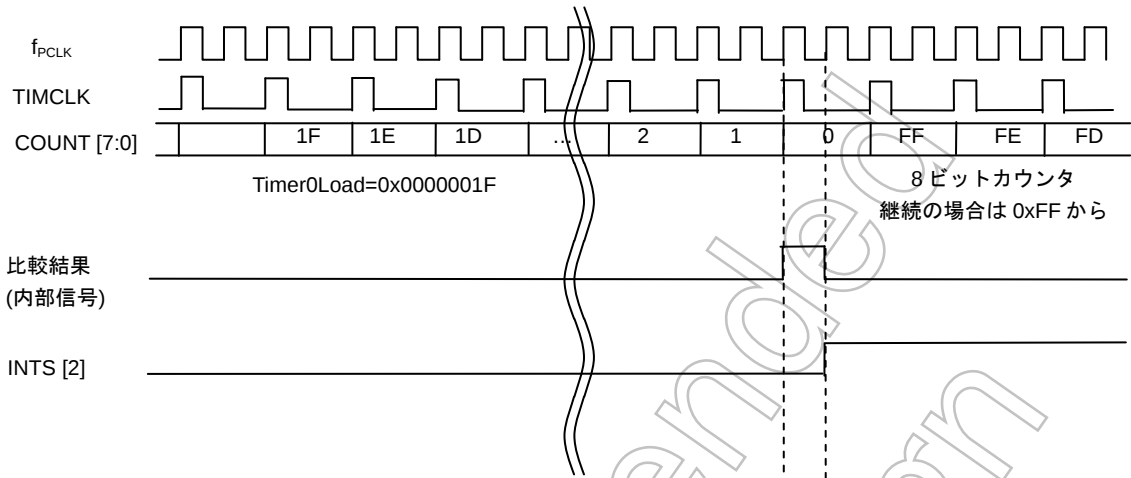
(1) One-shot-operation



(2) Wrapping-operation



下図はタイマの値を 0x0000001F に設定した例を示しています。



Free-Running モード設定例 (Wrapping-operation)

レジスタ	Bits										Function
	MSB									LSB	
	[31:8]	7	6	5	4	3	2	1	0		
Timer0Control	0x000000	0	×	×	0	×	×	×	×		[7]: タイマ 0 を停止します。
Timer0Load	0x000000	0	0	0	1	1	1	1	1		[15:0]: タイマ 0 周期を 0x0000001F に設定します。
Timer0Control	0x000000	1	0	1	0	0	0	0	0		[7]: タイマ 0 を許可します。(カウントを開始する) [6]: Free-Runnig モードを選択します。 [5]: タイマ割り込みを許可します。 [3:2]: 入力クロック T0 を選択します。 [1]: 8-bit カウンタを選択します。 [0]: Wrapping-operation を選択します。

×: Don't care

2) Periodic タイマモード

タイマをスタートすると設定されたカウンタ値からデクリメントを開始します。カウンタの値が 0 になると割り込みが発生します。

One-shot-operation(Timer0Control<TIM0OSCTL> = 1)の場合は、一度だけ割り込みを発生します。

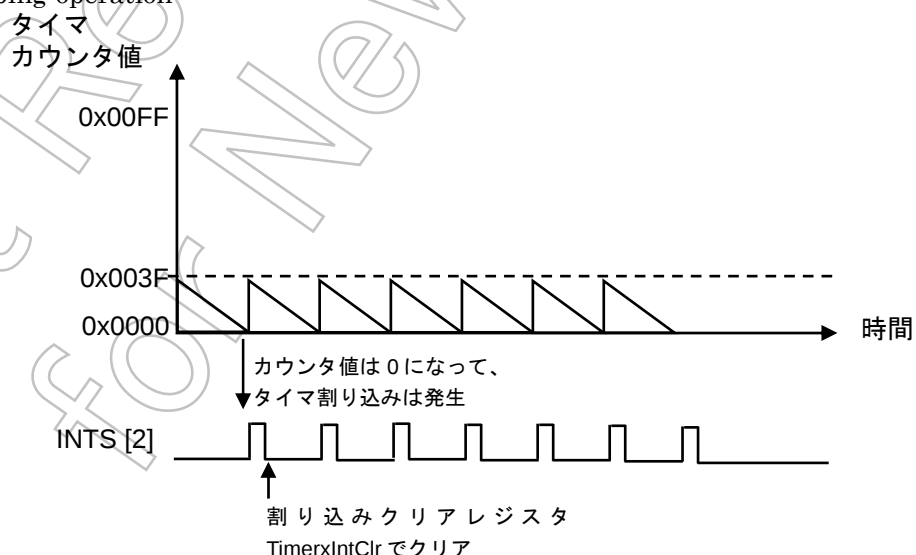
Wrapping-operation(Timer0Control<TIM0OSCTL>=0)の場合は、カウンタに設定された値リロードし、デクリメントを継続します。そのため、割り込みは固定周期で発生します。

下の動作概略図は 8 ビットカウンタ、タイマの値を 0x0000003F に設定した例を示しています。

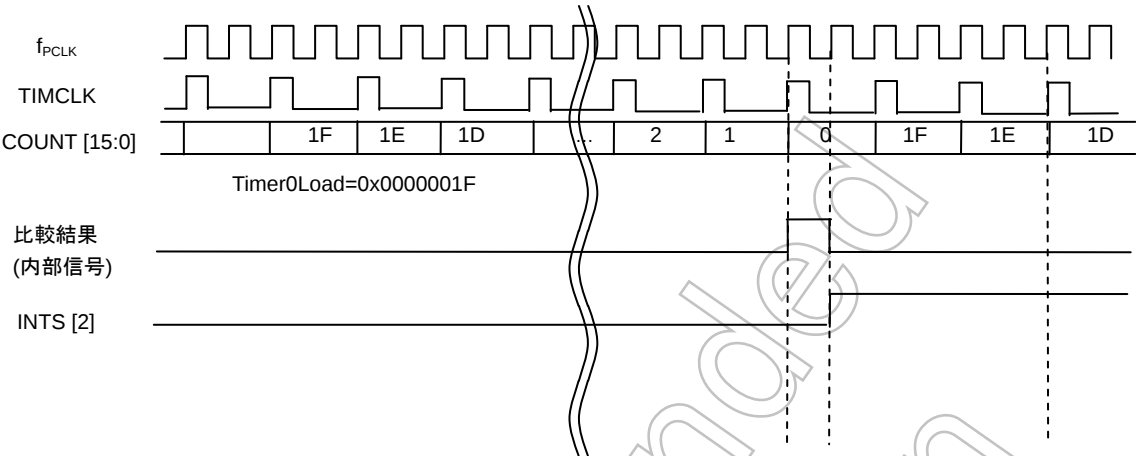
(1) One-shot-operation



(2) Wrapping-operation



下記図はタイマの値を 0x0000001F に設定した例を示しています。



Periodic タイマモード設定例 (Wrapping-operation)

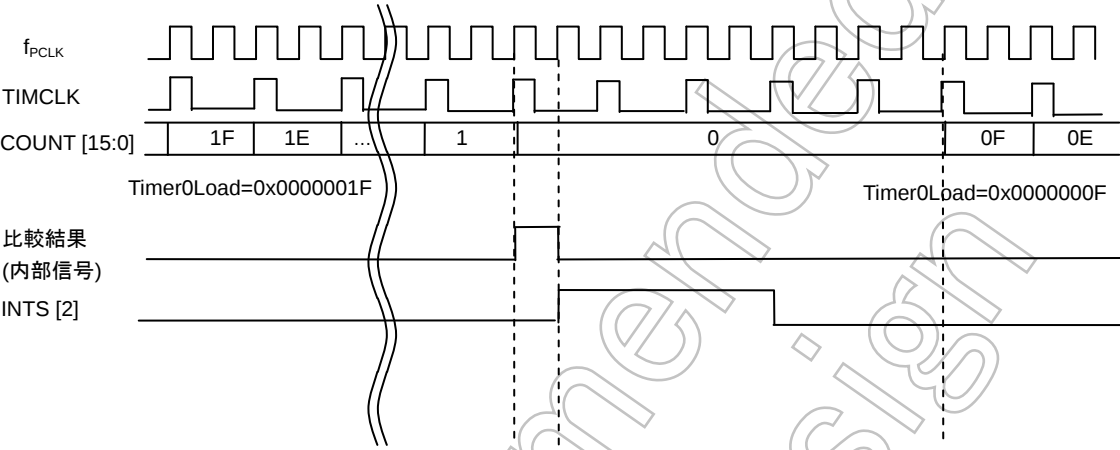
レジスタ	Bits									Function
	MSB								LSB	
	[31:8]	7	6	5	4	3	2	1	0	
Timer0Control	0x000000	0	×	×	0	×	×	×	×	[7]: タイマ 0 を停止します。
Timer0Load	0x000000	0	0	0	1	1	1	1	1	[15:0]: タイマ 0 周期を 0x0000001F に設定します。
Timer0Control	0x000000	1	1	1	0	0	0	1	0	[7]: タイマ 0 を許可します。(カウントを開始する) [6]: Periodic タイマモードを選択します。 [5]: タイマ割り込みを許可します。 [3:2]: 入力クロック T0 を選択します。 [1]: 16-bit カウンタを選択します。 [0]: Wrapping-operation を選択します。

×: Don't care

・ One-shot operation の注意

One-shot-operation で、タイマを再スタートしたい場合には、Timer0Load レジスタ値を再設定してください。

Timer0Load レジスタ値を再設定しないまま、Timer0Control<TIM0EN>に 1 をライトしてもタイマを再スタートできません。



Free-Running モード設定例 (One-shot-operation)

レジスタ	Bits										Function
	MSB									LSB	
	[31:8]	7	6	5	4	3	2	1	0		
Timer0Control	0x000000	0	x	x	0	x	x	x	x		[7]: タイマ 0 を停止します。
Timer0Load	0x000000	0	0	0	1	1	1	1	1		[15:0]: タイマ 0 周期を 0x0000001F に設定します
Timer0Control	0x000000	1	0	1	0	0	0	0	1		[7]: タイマ 0 を許可します。(カウントを開始する) [6]: Free-Running モードを選択します。 [5]: タイマ割り込みを許可します。 [3:2]: 入力クロックを T0 にします。 [1]: 8-bit カウンタを選択します。 [0]: One-shot-operation を選択します。
Timer0IntClr	x	x	x	x	x	x	x	x	x		[32:0]: 任意値をライトすると割り込みをクリアできます。

x: Don't care

・ PWM 機能のサポート

Block 1 と Block 2 に 2 チャンネルの 16 ビット PWM 機能があります。2 チャンネルの PWM 出力は PWM0OUT 端子(PC3)と PWM2OUT 端子 (PC4) へ出力されます。

PWM0OUT は、デクリメントカウンタがコンペアレジスタ Timer0Compare1 の設定値と一致したとき、または Timer0Mode<PWM Period>で指定されたカウンタがデクリメントして 0 になった時、出力を反転します。

Timer0Compare1 への設定は Duty 0%から 100%まで可能です。また、デクリメントカウンタのカウンタ値は 0 になると $2^n - 2$ からカウンタダウンを再開します。

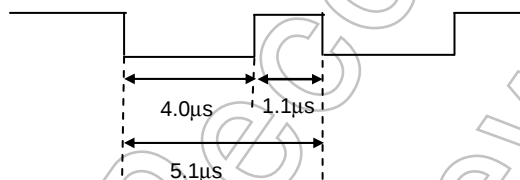
また、2 チャンネルは同様の仕様となっていますので、タイマ 2 の場合も同様です。

注 1) PWM 機能を使用する場合は、必ず「Periodic タイマモード」、「16 ビットカウンタ」、「Wrapping-operation」の設定にしてください。

注 2) Timer0Control< TIM0EN >に 0 を設定しタイマを停止することで PWM 機能を停止させた場合、PWM 出力の状態は保持されます。この状態で Timer0Control< TIM0EN >に 1 を設定しタイマを再動作させた場合、タイマを停止した時の状態から PWM の出力が開始されます。したがって、タイマを停止した時の PWM の出力が H レベルの場合、H レベルから出力されるため、位相が反転します。

PWM 機能を Disable にしていただくことで、出力のデータは初期化されますので、PWM 機能を再動作する際、初期化したデータを出力させる場合、必ず、Timer0Mode<PWM Mode>を Disable に設定して下さい。

例 : $f_{PCLK}=100\text{MHz}$, $TIMCLK=50\text{MHz}$ 時、タイマ 0 を使用して下記の PWM 波形を PWM0OUT 端子へ出力する場合



(1) PWM 周期 $5.1\mu\text{s}$ を $T0=0.02\mu\text{s}$ で実現する場合、

$5.1\mu\text{s} \div 0.02\mu\text{s} = 255 = 2^n - 1$ 、したがって $n=8$ に設定します。

(2) “L”レベルの期間は $4.0\mu\text{s}$ なので、 $T0=0.02\mu\text{s}$ では、

$(5.1\mu\text{s} - 4.0\mu\text{s}) / 0.02\mu\text{s} = 55 = 0x37$ を Timer0Compare1 に設定します。

レジスタ	Bits [31:8]	MSB				LSB				Function
		7	6	5	4	3	2	1	0	
Timer0Control	0x000000	0	x	x	0	x	x	x	x	[7]: タイマ 0 を停止します。
Timer0Mode	0x000000	0	1	0	0	0	0	0	0	[6]: PWM モードを選択します。 [5:4]: PWM 周期を $2^8 - 1$ にします。
Timer0Compare1	0x000000	0	0	1	1	0	1	1	1	[7:0]: コンペア値 0x37 を設定します。
Timer0CmpEn	0x000000	0	0	0	0	0	0	0	1	[0]: コンペアを許可します。
Timer0Control	0x000000	1	1	1	0	0	0	1	0	[7]: タイマ 0 を許可します。(カウントを開始する) [6]: Periodic タイマモードを選択します。 [5]: タイマ割り込みを許可します。 [3:2]: 入力クロックを T0 にします。 [1]: 16-bit カウンタを選択します。(必ず 16bit にする) [0]: Wrapping-operation を選択します。

x: Don't care

PWM 最小分解能とデューティについて説明します。

表 3.12.1 PWM 最小分解能 (TIMCLK=50MHz)

PWM 周期 プリスケアラ	2^8-1	2^9-1	$2^{10}-1$	$2^{16}-1$
T0	5.1 μ s	10.22 μ s	20.46 μ s	1.31ms
T16	81.6 μ s	163.52 μ s	327.36 μ s	20.97ms
T256	1.305ms	2.62ms	5.24ms	335.54ms



図 3.12.2 PWM 出力波形例

例：周期 2^8-1 (255 カウント) の場合の Duty について

PWM 出力の初期値は常に“L”出力です。また、Duty 0%は常に“L”出力を示し、Duty 100%は常に“H”出力を示します。

Timer0Compare1 0x00 セットで Duty は $0/255 \times 100 = 0\%$

Timer0Compare1 0x01 セットで Duty は $1/255 \times 100 = 0.39\%$

：

Timer0Compare1 0xFE セットで Duty は $254/255 \times 100 = 99.6\%$

Timer0Compare1 0xFF セットで Duty は $255/255 \times 100 = 100\%$

- 2^n-1 モード時、Timer0Compare1 値に 2^n-1 を設定すると PWM 出力の F/F は“H”にセットされてしまうのでこの後に PWM 周期だけを変更してスタートする場合は、一度 PWM モードを disable してから設定し直してください。
- PWM モードを使用する場合は次の条件を満たしてください。
 $0 \leq (\text{TimerxCompare1 の設定値}) \leq 2^n-1$

3.12.4 レジスタの説明

SFR のリストと機能を以下に示します。

表 3.12.2 SFR 一覧表 (1/3)

base アドレス= 0xF004_0000

Register Name	Address (base+)	Description
Timer0Load	0x0000	Timer 0 Load value
Timer0Value	0x0004	The current value for Timer 0
Timer0Control	0x0008	Timer 0 control register
Timer0IntClr	0x000C	Timer 0 interrupt clear
Timer0RIS	0x0010	Timer 0 raw interrupt status
Timer0MIS	0x0014	Timer 0 masked interrupt status
Timer0BGLoad	0x0018	Background load value for Timer 0
Timer0Mode	0x001C	Timer 0 mode register
–	0x0020	Reserved
–	0x0040	Reserved
–	0x0060	Reserved
–	0x0064	Reserved
–	0x0068	Reserved
Timer0Compare1	0x00A0	Timer 0 Compare value
Timer0CmplIntClr1	0x00C0	Timer 0 Compare Interrupt clear
Timer0CmpEn	0x00E0	Timer 0 Compare Enable
Timer0CmpRIS	0x00E4	Timer 0 Compare raw interrupt status
Timer0CmpMIS	0x00E8	Timer 0 Compare masked int status
Timer0BGCmp	0x00EC	Background compare value for Timer 0
–	0x00F0	Reserved
Timer 1Load	0x0100	Timer 1 Load value
Timer 1Value	0x0104	The current value for Timer 1
Timer 1Control	0x0108	Timer 1 control register
Timer 1IntClr	0x010C	Timer 1 interrupt clear
Timer 1RIS	0x0110	Timer 1 raw interrupt status
Timer 1MIS	0x0114	Timer 1 masked interrupt status
Timer 1BGLoad	0x0118	Background load value for Timer 1
–	0x0120	Reserved
–	0x0140	Reserved
–	0x0160	Reserved
–	0x0164	Reserved
–	0x0168	Reserved
–	0x01A0	Reserved
–	0x01C0	Reserved
–	0x01E0	Reserved
–	0x01E4	Reserved
–	0x01E8	Reserved

表 3.12.2 SFR 一覧表 (2/3)

base アドレス= 0xF004_1000

Register Name	Address (base+)	Description
Timer2Load	0x0000	Timer 2 Load value
Timer2Value	0x0004	The current value for Timer 2
Timer2Control	0x0008	Timer 2 control register
Timer2IntClr	0x000C	Timer 2 interrupt clear
Timer2RIS	0x0010	Timer 2 raw interrupt status
Timer2MIS	0x0014	Timer 2 masked interrupt status
Timer2BGLoad	0x0018	Background load value for Timer 2
Timer2Mode	0x001C	Timer 2 mode register
–	0x0020	Reserved
–	0x0040	Reserved
–	0x0060	Reserved
–	0x0064	Reserved
–	0x0068	Reserved
Timer2Compare1	0x00A0	Timer 2 Compare value
Timer2CmplntClr1	0x00C0	Timer 2 Compare Interrupt clear
Timer2CmpEn	0x00E0	Timer 2 Compare Enable
Timer2CmpRIS	0x00E4	Timer 2 Compare raw interrupt status
Timer2CmpMIS	0x00E8	Timer 2 Compare masked int status
Timer2BGCmp	0x00EC	Background compare value for Timer 2
:	:	:
Timer3Load	0x0100	Timer 3 Load value
Timer3Value	0x0104	The current value for Timer 3
Timer3Control	0x0108	Timer 3 control register
Timer3IntClr	0x010C	Timer 3 interrupt clear
Timer3RIS	0x0110	Timer 3 raw interrupt status
Timer3MIS	0x0114	Timer 3 masked interrupt status
Timer3BGLoad	0x0118	Background load value for Timer 3
–	0x0120	Reserved
–	0x0140	Reserved
–	0x0160	Reserved
–	0x0164	Reserved
–	0x0168	Reserved
–	0x01A0	Reserved
–	0x01C0	Reserved
–	0x01E0	Reserved
–	0x01E4	Reserved
–	0x01E8	Reserved

表 3.12.2 SFR 一覧表 (3/3)

base アドレス= 0xF004_2000

Register Name	Address (base+)	Description
Timer4Load	0x0000	Timer 4 Load value
Timer4Value	0x0004	The current value for Timer 4
Timer4Control	0x0008	Timer 4 control register
Timer4IntClr	0x000C	Timer 4 interrupt clear
Timer4RIS	0x0010	Timer 4 raw interrupt status
Timer4MIS	0x0014	Timer 4 masked interrupt status
Timer4BGLoad	0x0018	Background load value for Timer 4
–	0x001C	Reserved
–	0x0020	Reserved
–	0x0040	Reserved
–	0x0060	Reserved
–	0x0064	Reserved
–	0x0068	Reserved
–	0x00A0	Reserved
–	0x00C0	Reserved
–	0x00E0	Reserved
–	0x00E4	Reserved
–	0x00E8	Reserved
–	0x00EC	Reserved
:	:	:
Timer5Load	0x0100	Timer 5 Load value
Timer5Value	0x0104	The current value for Timer 5
Timer5Control	0x0108	Timer 5 control register
Timer5IntClr	0x010C	Timer 5 interrupt clear
Timer5RIS	0x0110	Timer 5 raw interrupt status
Timer5MIS	0x0114	Timer 5 masked interrupt status
Timer5BGLoad	0x0118	Background load value for Timer 5
–	0x0120	Reserved
–	0x0140	Reserved
–	0x0160	Reserved
–	0x0164	Reserved
–	0x0168	Reserved
–	0x01A0	Reserved
–	0x01C0	Reserved
–	0x01E0	Reserved
–	0x01E4	Reserved
–	0x01E8	Reserved

1. Timer0Load Register

Address = (0xF004_0000) + (0x0000)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	—	—	Undefined	Read as undefined. Write as zero.
[15:0]	TIM0SD[15:0]	R/W	0x0000	タイマ 0 のインターバルの値を設定

(個別説明)

a. <TIMxSD[15:0]>

タイマの周期をセットするレジスタです。

カウンタはデクリメントカウンタとなっており、0x0001~0xFFFF の設定が可能です(設定したいタイマの周期-1 の値を設定してください)。

注) 0x0000 の設定は禁止です。0x0000 を設定した場合、0xFFFF からデクリメントを開始します。

8 ビットカウンタを使用する場合は、上位 8 ビットの値は無視されます。

Periodic タイマモードで、かつ Wrapping-operation を使用する場合は、リロードする値にも使用されます。

また、本レジスタにかかれた値はレジスタにライトされた後、すぐに更新されます。

デクリメントカウンタの値が 0x0000 状態になった時に、更新したい場合には、後述の Timer0BGLoad レジスタにライトすることで対応出来ます。

- TimerxLoad (Timer x Load value register) (x = 0 ~ 5)

上記のレジスタ群について、構造および説明は Timer0Load と同様のため、Timer0Load の説明を参照してください。またレジスタ名およびアドレスは表 3.12.2 SFR 一覧表を参照してください。

2. Timer0Value Register

Address = (0xF004_0000) + (0x0004)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined.
[15:0]	TIM0CD[15:0]	RO	0xFFFF	タイマ 0 の現在のカウンタ値

(個別説明)

a. <TIMxCD[15:0]>

現在のタイマの値を読み出すためのレジスタです。

このレジスタはデクリメントカウンタの現在の値を示しています。

- TimerxValue (Timer x value register) (x = 0 ~ 5)

上記のレジスタ群について、構造および説明はTimer0Value と同様のため、Timer0Value の説明を参照してください。またレジスタ名およびアドレスは表 3.12.2 SFR一覧表を参照してください。

3. Timer0Control Register

Address = (0xF004_0000) + (0x0008)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7]	TIM0EN	R/W	0y0	タイマ 0 のイネーブルビット 0y0: 停止 0y1: 動作
[6]	TIM0MOD	R/W	0y0	タイマ 0 のモード設定 0y0: Free-Running モード 0y1: Periodic タイマモード
[5]	TIM0INTE	R/W	0y1	タイマ 0 割り込み制御 0y0: 割り込み Disable 0y1: 割り込み Enable
[4]	–	–	Undefined	Read as undefined. Write as zero.
[3:2]	TIM0PRS	R/W	0y00	タイマ 0 プリスケアラ設定 0y00: 等倍 0y01: 16 分周 0y10: 256 分周 0y11: 設定禁止
[1]	TIM0SIZE	R/W	0y0	タイマ 0 の 8/16 カウンタ切り替え 0y0: 8bit カウンタ 0y1: 16bit カウンタ
[0]	TIM0OSCTL	R/W	0y0	タイマ 0 の One-shot/Wrapping カウンタ切り替え 0y0: Wrapping operation 0y1: One-shot operation

(個別説明)

a. <TIMxEN>

タイマの動作の動作/停止を制御するビットです。

0y0: 停止

0y1: 動作

タイマのカウンタ途中でタイマを停止して、動作の再開について

1) 再動作(Enable)をする場合、停止したカウンタ値からデクリメントします。

2) TimerxLoad を設定し、再動作(Enable)をする場合、TimerxLoad の設定値からデクリメントします。

b. <TIMxMOD>

タイマ動作のモードを切り替えるビットです。

c. <TIMxINTE>

タイマ割り込みのマスク制御を行います。

d. <TIMxPRS>

タイマのソースクロックを分周するプリスケアラを設定します。

e. <TIMxSIZE>

8/16 ビットのカウンタを選択します。

f. <TIMxOSCTL>

One-shot/Wrapping カウンタを選択します。

- TimerxControl (Timer x Control register) (x = 0 ~ 5)

上記のレジスタ群について、構造および説明はTimer0Control と同様のため、Timer0Controlの説明を参照してください。またレジスタ名およびアドレスは表 3.12.2 SFR 一覧表を参照してください。

Not Recommended
for New Design

4. Timer0IntClr Register

Address = (0xF004_0000) + (0x000C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	TIM0INTCLR	WO	Undefined	タイマ 0 の割り込みクリア

(個別説明)

a. <TIMxINTCLR>

タイマの割り込みをクリアするレジスタです。

書き込むデータは任意で、本レジスタにライトすることで対象の割り込みがクリアされます。

(バス幅も任意：8/16/32 のバス幅に対応しています)

• TimerxIntClr (Timer x Interrupt Clear register) (x = 0 ~ 5)

上記のレジスタ群について、構造および説明はTimer0IntClrと同様のため、Timer0IntClrの説明を参照してください。またレジスタ名およびアドレスは表 3.12.2 SFR一覧表を参照してください。

5. Timer0RIS Register

Address = (0xF004_0000) + (0x0010)

Bit	Bit Symbol	Type	Reset Value	Description
[31:1]	–	–	Undefined	Read as undefined.
[0]	TIM0RIF	RO	0y0	タイマ 0 の割り込みフラグ 0y0: 割り込みなし 0y1: 割り込みあり

(個別説明)

a. <TIMxRIF>

このレジスタは、内部カウンタの割込ステータスを示します。TIMxCR<TIMxINTE>で設定された割り込み制御の状態にかかわらず、内部の状態を示します。

- TimerxRIS (Timer x Interrupt Raw Flag register) (x = 0 ~ 5)

上記のレジスタ群について、構造および説明はTimer0RISと同様のため、Timer0RISの説明を参照してください。またレジスタ名およびアドレスは表 3.12.2 SFR一覧表を参照してください。

6. Timer0MIS Register

Address = (0xF004_0000) + (0x0014)

Bit	Bit Symbol	Type	Reset Value	Description
[31:1]	–	–	Undefined	Read as undefined.
[0]	TIM0MIF	RO	0y0	タイマ 0 の割り込みフラグ 0y0: 割り込みなし 0y1: 割り込みあり

(個別説明)

a. <TIMxMIF>

このレジスタは、割り込みマスク回路を経由した、割込ステータスを示します。

TIMxCR<TIMxINTE>で設定された割り込み制御の状態によって、Flag の状態は変化します。

(TIMxCR<TIMxINTE> = 0 の場合は常に 0 を示します)

- TimerxMIS (Timer x Interrupt Masked Flag register) (x = 0 ~ 5)

上記のレジスタ群について、構造および説明はTimer0MISと同様のため、Timer0MISの説明を参照してください。またレジスタ名およびアドレスは表 3.12.2 SFR一覧表を参照してください。

7. Timer0BGLoad Register

Address = (0xF004_0000) + (0x0018)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined. Write as zero.
[15:0]	TIM0BSD[15:0]	R/W	0x00	タイマ 0、BG カウンタのインターバルの値を設定

(個別説明)

a. <TIMxBSD[15:0]>

このレジスタは、TimerxLoad レジスタのバックグラウンド用カウンタ値を設定します。

Periodic タイマモードで、かつ Wrapping-operation を使用する場合、カウンタをリロードするために使われます。

TimerxLoad レジスタと違い、TimerxBGLoad レジスタへ書き込んでもすぐにリロードされず、デクリメントカウンタの値が 0x0000 状態になった時にリロードされます。

また、TimerxLoad レジスタと TimerxBGLoad レジスタのリード時/ライト時の関係は下記のようになっています。

- ライト時

TimerxLoad レジスタにライトした場合、同じデータが TimerxLoad レジスタと TimerxBGLoad レジスタの両方にライトされます。また、TimerxBGLoad レジスタにライトした場合は TimerxBGLoad レジスタのみにライトされます。

- リード時

TimerxLoad レジスタ、TimerxBGLoad レジスタのどちらをリードした場合も TimerxBGLoad レジスタのデータがリードされ、最後に設定されたタイマの周期をリードすることができます。

- TimerxBGLoad (Timer x Back Ground Counter Data register) (x = 0 ~ 5)

上記のレジスタ群について、構造および説明は Timer0BGLoad と同様のため、Timer0BGLoad の説明を参照してください。またレジスタ名およびアドレスは表 3.12.2 SFR 一覧表を参照してください。

8. Timer0Mode Register

Address = (0xF004_0000) + (0x001C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:7]	–	–	Undefined	Read as undefined. Write as zero.
[6]	PWM Mode	R/W	0y0	PWM モード選択 : 0y0: PWM Disable 0y1: PWM Enable
[5:4]	PWM Period	R/W	0y00	PWM モード時の周期選択: 0y00: $2^8 - 1$ 0y01: $2^9 - 1$ 0y10: $2^{10} - 1$ 0y11: $2^{16} - 1$
[3:0]	–	–	Undefined	Read as undefined. Write as zero.

(個別説明)

- a. <PWM Mode>
PWM モードの Enable を選択

- b. <PWM Period>
PWM モード時の周期選択

- TimerxMode (Timer x mode register) (x = 0, 2)

上記のレジスタ群について、構造および説明はTimer0Modeと同様のため、Timer0Modeの説明を参照してください。またレジスタ名およびアドレスは表 3.12.2 SFR一覧表を参照してください。

9. Timer0Compare1 Register

Address = (0xF004_0000) + (0x00A0)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined. Write as zero.
[15:0]	TIM0CPD	R/W	0x00	タイマ 0、カウンタの値と比較する値を設定: 0x0001 ~ 0xFFFF

(個別説明)

a. <TIMxCPD>

タイマカウンタの値と比較する値をセットするレジスタです。

- TimerxCompare1 (Timer x Compare Value register) (x = 0 ~ 5)

上記のレジスタ群について、構造および説明はTimer0Compare1 と同様のため、Timer0Compare1 の説明を参照してください。またレジスタ名およびアドレスは 表 3.12.2 SFR一覧表を参照してください。

10. Timer0CmpIntClr1 Register

Address = (0xF004_0000) + (0x00C0)

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	TIM0CMINTCLR	WO	Undefined	タイマ 0 のコンペア割り込みクリア

(個別説明)

a. <TIMxCMINTCLR>

タイマコンペア割り込みをクリアするレジスタです。

書き込むデータは任意で、本レジスタにライトすることで対象の割り込みがクリアされます。

(バス幅も任意: 8/16/32 のバス幅に対応しています)

- TimerxCmpIntClr1 (Timer x Compare Interrupt Clear register) (x = 0 ~ 5)

上記のレジスタ群について、構造および説明はTimer0CmpIntClr1 と同様のため、Timer0CmpIntClr1 の説明を参照してください。またレジスタ名およびアドレスは表 3.12.2 SFR一覧表を参照してください。

11. Timer0CmpEn Register

Address = (0xF004_0000) + (0x00E0)

Bit	Bit Symbol	Type	Reset Value	Description
[31:1]	–	–	Undefined	Read as undefined. Write as zero.
[0]	TIM0CPE	RO	0y0	タイマ 0 のコンペア動作の Enable 0y0: Disable 0y1: Enable

(個別説明)

a. <TIMxCPE>

タイマのコンペア動作を許可するためのレジスタです。

また、割り込みのマスクも兼ねています。

• TimerxCmpEn (Timer x Compare Enable register) (x = 0 ~ 5)

上記のレジスタ群について、構造および説明はTimer0CmpEnと同様のため、Timer0CmpEnの説明を参照してください。またレジスタ名およびアドレスは表 3.12.2 SFR一覧表を参照してください。

12. Timer0CmpRIS Register

Address = (0xF004_0000) + (0x00E4)

Bit	Bit Symbol	Type	Reset Value	Description
[31:1]	–	–	Undefined	Read as undefined.
[0]	TIM0CRIF	RO	0y0	タイマ 0 コンペアの許可前割り込みフラグ 0y0: 割り込みなし 0y1: 割り込みあり

(個別説明)

a. <TIMxCRIF>

このレジスタはコンペアにより発生する許可前の (マスクされていない) 割り込みステータスを示します。TIMxCPMIS で設定された割り込み制御の状態にかかわらず、内部の状態を示します。

- TimerxCmpRIS (Timer x Compare raw interrupt status register) (x = 0 ~ 5)

上記のレジスタ群について、構造および説明はTimer0CmpRISと同様のため、Timer0CmpRISの説明を参照してください。またレジスタ名およびアドレスは表 3.12.2 SFR 一覧表を参照してください。

13. Timer0CmpMIS Register

Address = (0xF004_0000) + (0x00E8)

Bit	Bit Symbol	Type	Reset Value	Description
[31:1]	–	–	Undefined	Read as undefined.
[0]	TIM0CMIF	RO	0y0	タイマ 0 のコンペアの割り込みフラグ 0y0: 割り込みなし 0y1: 割り込みあり

(個別説明)

a. <TIMxCMIF>

このレジスタはマスクされたコンペア割り込みのステータスを示します。

- TimerxCmpMIS (Timer x Compare Masked interrupt status register) (x = 0 ~ 5)

上記のレジスタ群について、構造および説明はTimer0CmpMISと同様のため、Timer0CmpMISの説明を参照してください。またレジスタ名およびアドレスは表 3.12.2 SFR 一覧表を参照してください。

14. Timer0BGComp Register

Address = (0xF004_0000) + (0x00EC)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined. Write as zero.
[15:0]	TIM0BGCPD	R/W	0x0000	タイマ 0、カウンタの値と比較する BackGround 値を設定: 0x0001 ~ 0xFFFF

Periodic モードのタイマ動作中にリロードしたいコンペア値を **TimerxBGComp** レジスタに書き込んだ場合、現在の値のままカウントは継続しカウント値が 0 に達すると **TimerxBGComp** レジスタの値がコンペアレジスタ **TimerxCompare1** レジスタへシフトインされます。

なお、PWM モードを使用する場合は次の条件を満たしてください。

$$0 \leq (\text{TIMxBGCPD の設定値}) \leq 2^n - 1$$

また、**TimerxCompare1** レジスタと **TimerxBGComp** レジスタのリード時/ライト時の関係は下記のようになっています。

- ライト時

TimerxCompare1 レジスタにライトした場合、同じデータが **TimerxCompare1** レジスタと **TimerxBGComp** レジスタの両方にライトされます。また、**TimerxBGComp** レジスタにライトした場合は **TimerxBGComp** レジスタのみにライトされます。

- リード時

TimerxCompare1 レジスタ、**TimerxBGComp** レジスタのどちらをリードした場合も **TimerxBGComp** レジスタのデータがリードされ、最後に設定されたタイマのコンペア値をリードすることができます。

TimerxBGComp (Timer x Back Ground Compare register) (x = 0, 2, 4)

上記のレジスタ群について、構造および説明は **Timer0BGComp** と同様のため、**Timer0BGComp** の説明を参照してください。またレジスタ名およびアドレスは表 3.12.2 SFR 一覧表を参照してください。

3.13 UART

本 LSI は UART: 2 チャンネルを内蔵しています。各チャンネル特長は下記の表に示します。

	チャンネル 0 (UART0)	チャンネル 1 (UART1)
送信 FIFO	幅 8-bit / 深さ 16 段、	
受信 FIFO	幅 12-bit / 深さ 16 段	
送受信データフォーマット	DATA: 5, 6, 7, 8bit を設定可能 PARITY: 有り / 無し STOP: 1bit / 2bit	
FIFO ON/OFF	ON(FIFO モード) / OFF(キャラクタモード)	
割り込み	(1) 各要因の結合割り込みを割り込みコントローラへ出力 (2) 各要因ごとに許可 / 不許可が設定可能。	
ボーレートジェネレータ	送受信共通の UART 内部クロックを生成するボーレートジェネレータ 最高 6.15Mbps(PCLK=100MHz 時)までのボーレート	
DMA	サポート	N/A
IrDA 1.0 機能	(1) 最大 115.2kbps(半二重) のデータレート (2) 低電力モードをサポート	N/A
制御端子	U0RXD U0TXD	U1RXD U1TXD U1CTS _n
ハードウェアフロー制御	RTS サポート CTS サポート	CTS サポート

(1) UART 送受信データフォーマット

送/受信データフォーマット			
START	DATA (LSB → MSB)	PARITY	STOP

(2) 受信 FIFO データフォーマット

Bit Number	受信 DATA (LSB → MSB)								フレーミング エラーフラグ	パリティ エラーフラグ	ブレーク エラーフラグ	オーバラン エラーフラグ
	0	1	2	3	4	5	6	7				
8 ビットデータ受信	1	1	1	1	1	1	1	1				
7 ビットデータ受信	1	1	1	1	1	1	1	0				
6 ビットデータ受信	1	1	1	1	1	1	0	0				
5 ビットデータ受信	1	1	1	1	1	0	0	0				

3.13.1 ブロック図

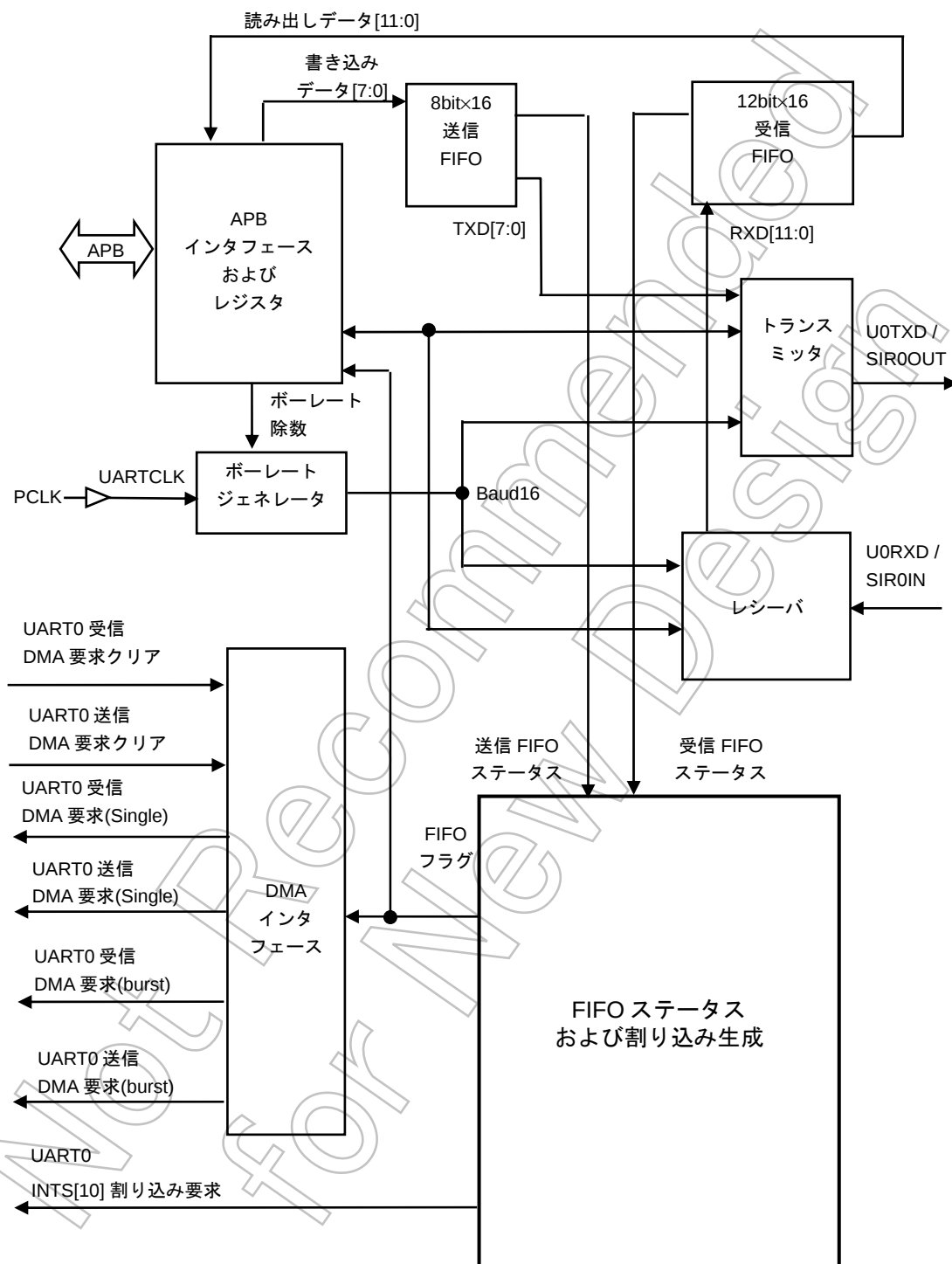


図 3.13.1 UART チャンネル 0 (UART0) ブロック図

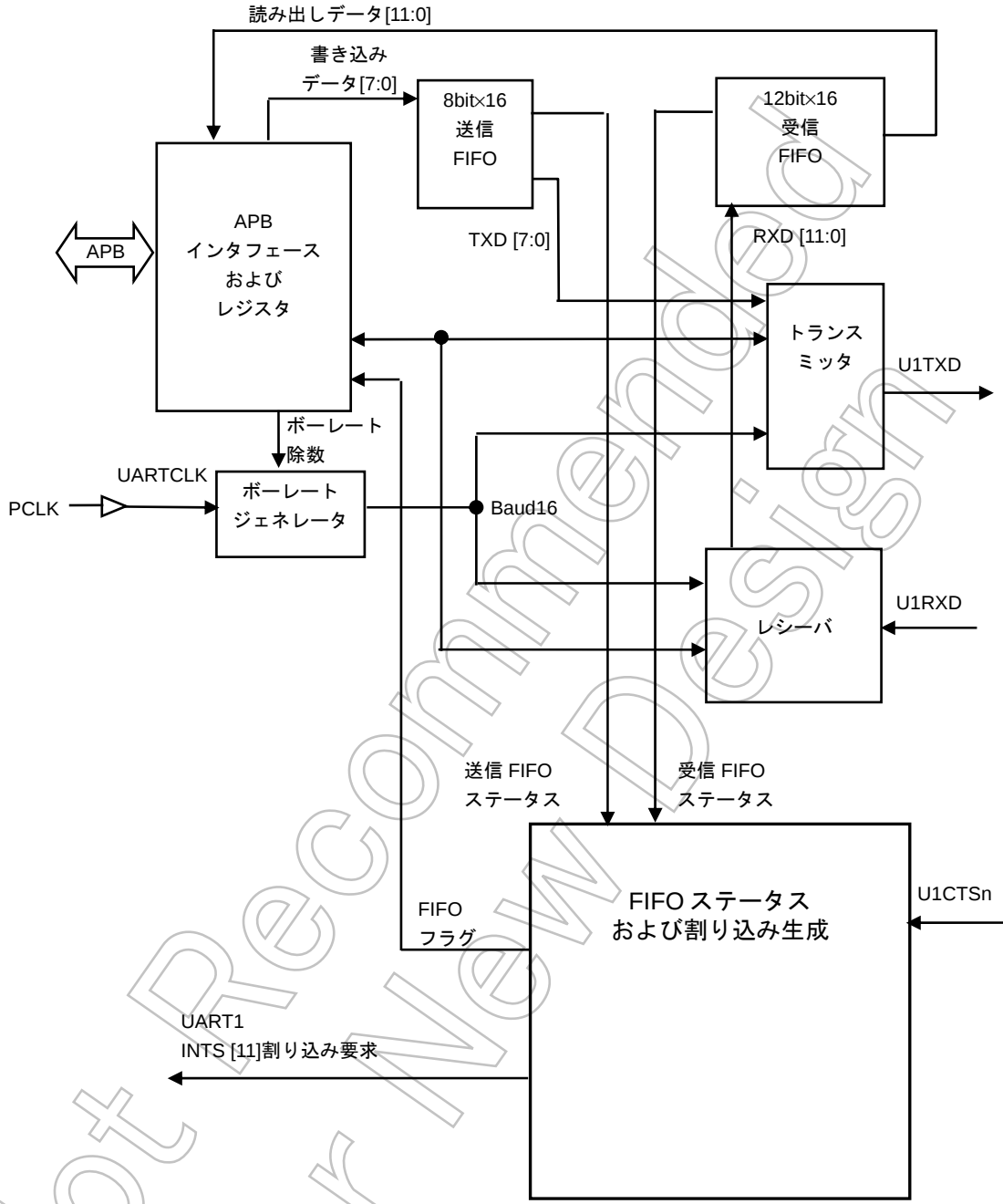


図 3.13.2 UART チャンネル 1 (UART1) ブロック図

3.13.1.1 動作説明

(1) ボーレートジェネレータ

ボーレートジェネレータは、UART 送信/受信制御のタイミングを生成する内部クロック(Baud16)と、低電力モード時に IrDA エンコード送信ビットストリームのパルス幅を生成する内部クロック(IrLPBaud16)で構成されています。

(2) 送信 FIFO

送信 FIFO は、8 ビットの幅と 16 段の深さを持つ FIFO メモリバッファです。APB インタフェース経由で書き込まれた CPU データは、送信ロジックによって読み出されるまで、この FIFO にストアされます。送信 FIFO はディセーブルすることによって 1 バイト保持レジスタのように動作させることができます。

(3) 受信 FIFO

受信 FIFO は、12 ビットの幅と 16 段の深さを持つ FIFO メモリバッファです。受信データと対応するエラービットは、APB インタフェース経由で CPU によって読み出されるまで、受信ロジックによって受信 FIFO にストアされます。受信 FIFO はディセーブルすることによって 1 バイト保持レジスタのように動作させることができます。

(4) 送信ロジック

送信ロジックは送信 FIFO から読み出されたデータの平行/シリアル変換を行います。制御ロジックは、制御レジスタ内にプログラムされたコンフィグレーションに基づき、スタートビット、最下位ビット(LSB)で始まるデータビット、その後パリティビット、ストップビットと続くシリアルビットストリームを出力します。

(5) 受信ロジック

受信ロジックは、スタートビット検出後に受信されたビットストリームのシリアル/平行変換を行います。オーバラン、パリティ、フレームの各エラーチェックとラインブレイクの検出も行われ、オーバラン、パリティ、フレーミング、ブレイクのエラービットに関連するデータが受信FIFO に書き込まれます。

(6) 割り込み生成ロジック

UART は、割り込み要因別にマスク可能な結合割り込みを出力します。

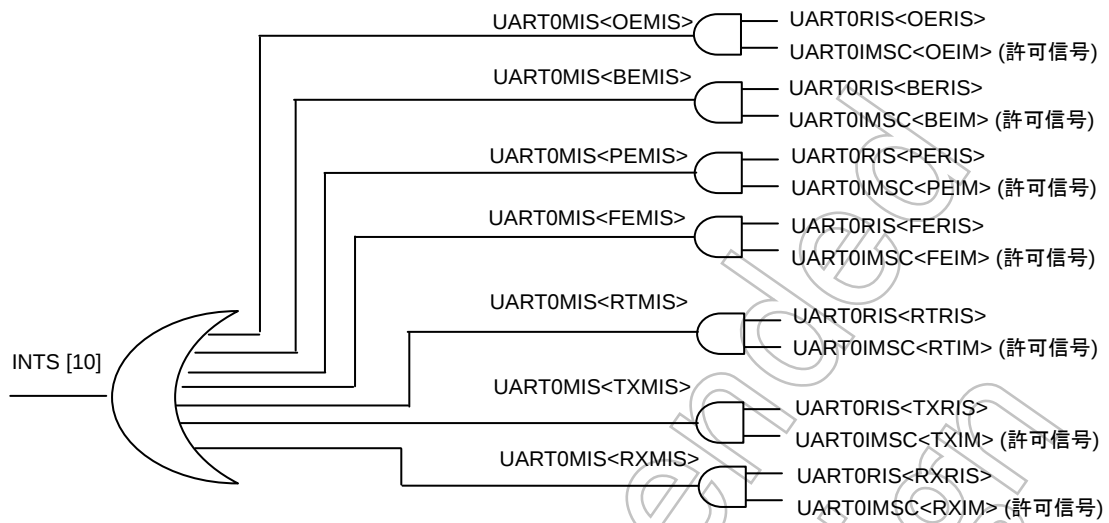
(7) 割り込み発生タイミング

割り込み種類	割り込み発生タイミング
オーバランエラー発生	過剰データの STOP ビット受信後
ブレークエラー割り込み	STOP ビット受信後
パリティ エラー発生	パリティデータ受信後
フレーミングエラー発生	フレームオーバーとなる Bit データを受信した後
受信タイムアウト割り込み	受信 FIFO にデータを取り込んでから Baud16 の 511 クロック後
送信割り込み	最終 DATA (MSB データ)を送信した後
受信割り込み	STOP ビット受信後

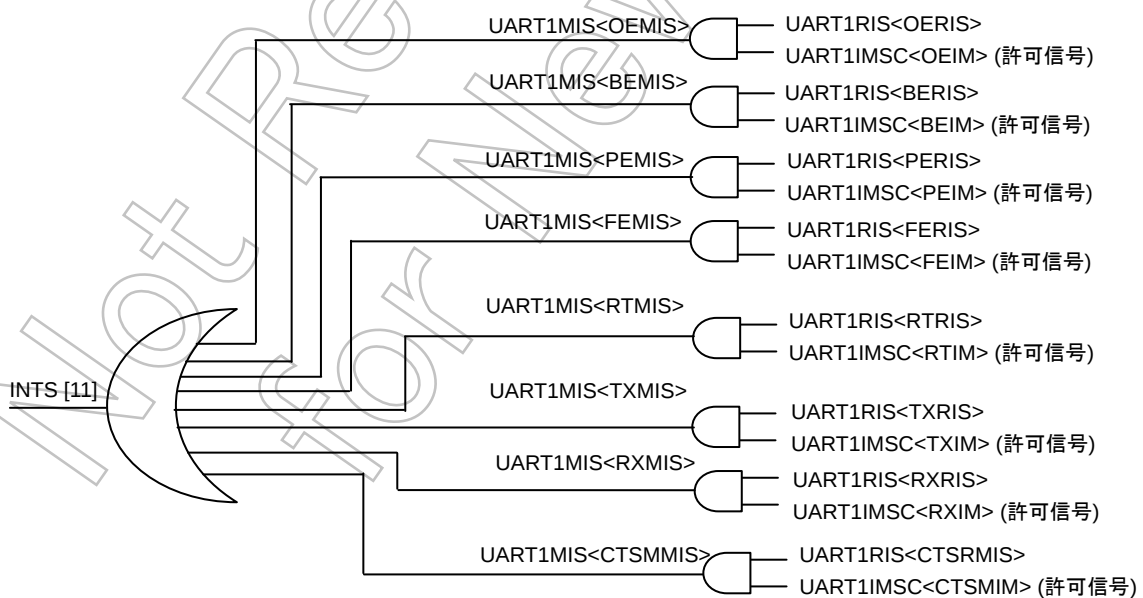
注) STOP ビットは最終 STOP ビットのことを意味します。(UARTxLCR_H<STP2>で STOP ビット選択 1/2 ビット可能)

(8) UART 割り込みブロック図

1) UART0 割り込みブロック図



2) UART1 割り込みブロック図



(9) DMA インタフェース

チャンネル 0 は DMA をサポートします。(チャンネル 1 はサポートしていません。)

(10) IrDA 回路解説

IrDA は以下から構成されています。

- ・ IrDA SIR 送信エンコーダ
- ・ IrDA SIR 受信デコーダ

注)送信エンコーダの出力SROUTは、受信デコーダの入力SIRINと逆の極性を持ちます。図 3.13.4を参照してください。

図 3.13.3は、IrDA回路のブロックダイアグラムを示しています。

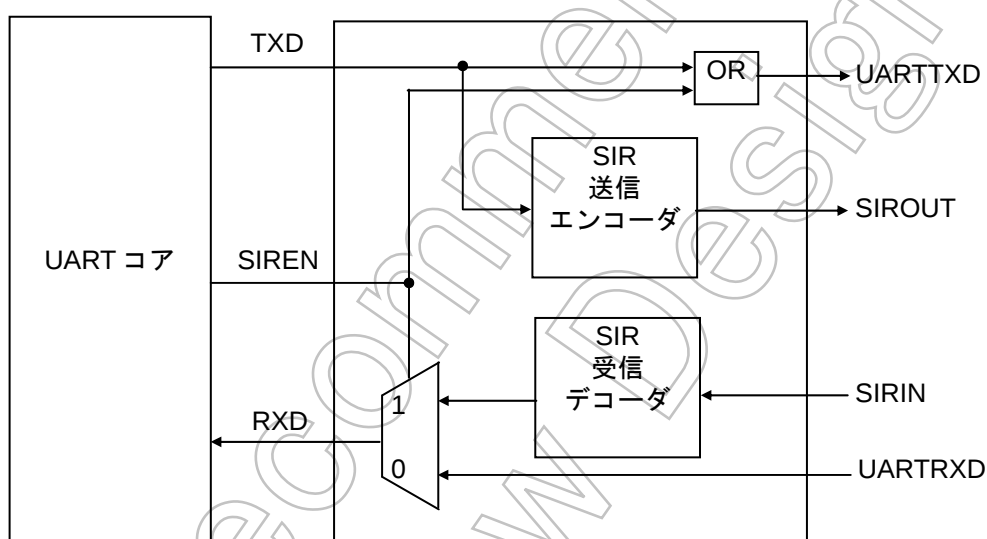


図 3.13.3 IrDA 回路のブロックダイアグラム

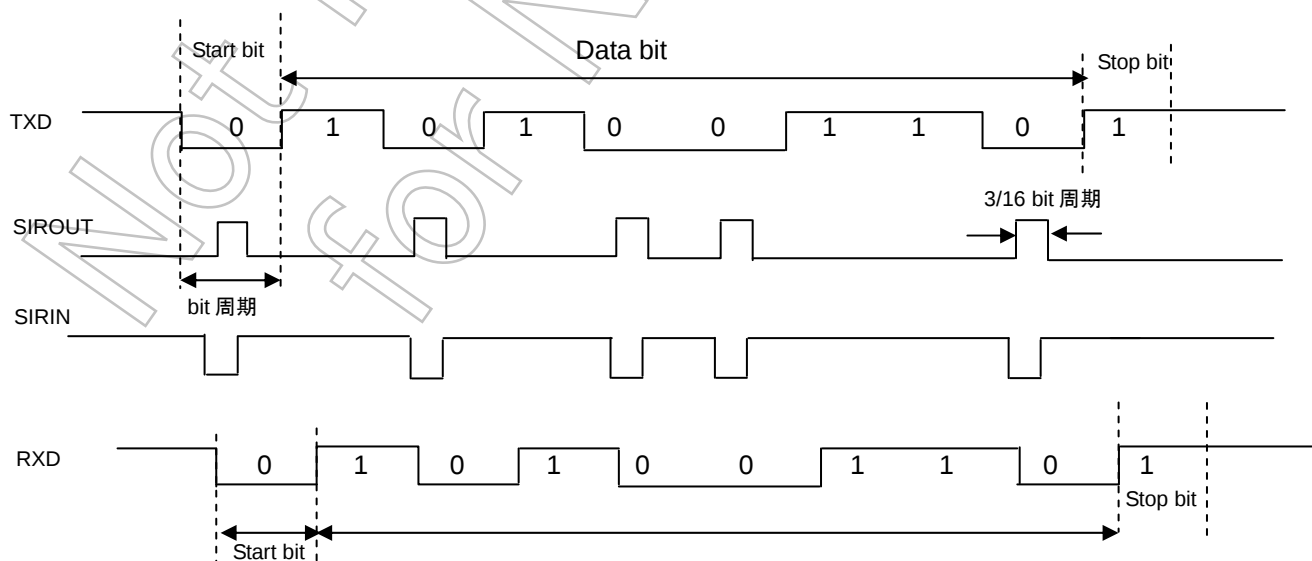


図 3.13.4 IrDA データ変調の波形

(11) ハードウェアフロー制御

ハードウェアフロー制御機能は選択可能であり、UxRTSn 出力シグナルと UxCTSn 入力シグナルを用いてシリアルデータフローを制御することができます。

図 3.13.5は、2つのデバイスがハードウェアフロー制御を用いてどのように通信するかを示しています。

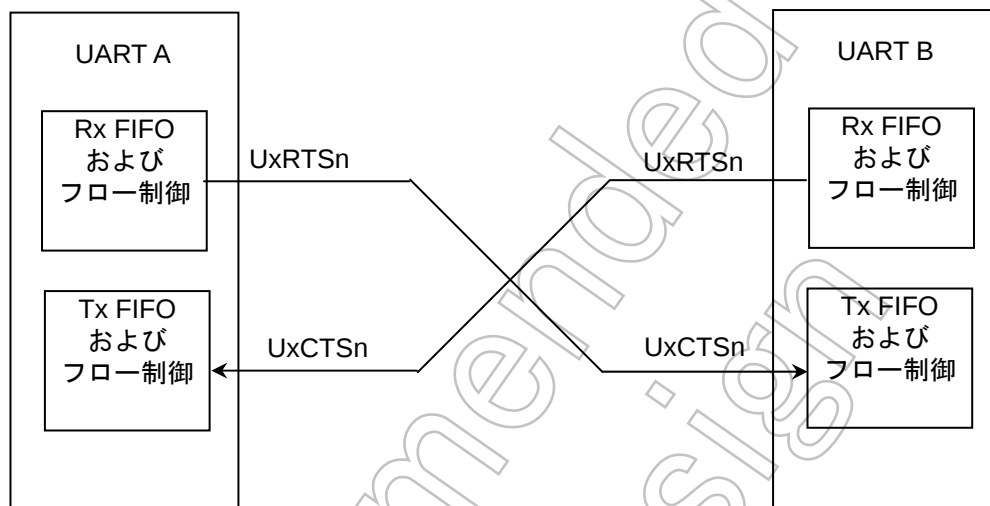


図 3.13.5 ハードウェアフロー制御

RTSフロー制御

RTS フロー制御ロジックは、プログラム可能な受信 FIFO ウォーターマークレベルにリンクしています。RTS フロー制御がイネーブルされている場合は、受信 FIFO がウォーターマークレベル未満の場合 UxRTSn がアサートされます。受信 FIFO がウォーターマークレベル以上になると、UxRTSn がアサート解除され、データを受信するための空き場所がないことを示します。

受信 FIFO からデータが読み出され、ウォーターマークレベル未満になると、UxRTSn シグナルが再度アサートされます。

RTS フロー制御がディセーブル状態でも通信は可能です。

CTSフロー制御

CTS フロー制御がイネーブルされている場合、送信する前に UxCTSn をチェックします。UxCTSn がアサートされていれば送信しますが、アサートされていなければ送信は発生しません。

UxCTSn がアサートされ、かつ送信 FIFO が空でない間はデータが送信され続けます。送信 FIFO が空であれば、UxCTSn がアサートされていてもデータは送信されません。

CTS フロー制御がイネーブルされている時に UxCTSn がアサート解除された場合には、現在送信中のデータが完了してから停止します。

CTS フロー制御がディセーブル状態でも通信は可能です。

表 3.13.1 ハードウェアフロー制御のイネーブル/ ディセーブルに用いる制御ビット

UARTxCR		nUARTRTS	説明
CTSEn	RTSEn		
1	1	0 ^{注)}	RTS および CTS の両方のフロー制御がイネーブルされます。
1	0	1	CTS フロー制御のみがイネーブルされます。
0	1	0 ^{注)}	RTS フロー制御のみがイネーブルされます。
0	0	1	RTS および CTS の両方のフロー制御がディセーブルされます。

注) RTSEn =1(Enable)の時は受信 FIFO がウォーターマークレベルに達するまで nUARTRTS= 0(Enable)となります。

3.13.2 レジスタの説明

SFR のリストと機能を以下に示します。

•UART0

base アドレス= 0xF200_0000

Register Name	Address (base+)	Description
UART0DR	0x000	UART0 Data register
UART0SR/ UART0ECR	0x004	UART0 Receive status register/ UART0 error clear register
—	0x008-0x014	Reserved
UART0FR	0x018	UART0 Flag register
—	0x01C	Reserved
UART0ILPR	0x020	UART0 IrDA low-power counter register
UART0IBRD	0x024	UART0 Integer baud rate register
UART0FBRD	0x028	UART0 Fractional baud rate register
UART0LCR_H	0x02C	UART0 Line control register
UART0CR	0x030	UART0 Control register
UART0IFLS	0x034	UART0 Interrupt FIFO level select register
UART0IMSC	0x038	UART0 Interrupt mask set/clear register
UART0RIS	0x03C	UART0 Raw interrupt status register
UART0MIS	0x040	UART0 Masked interrupt status register
UART0ICR	0x044	UART0 Interrupt clear register
UART0DMACR	0x048	UART0 DMA control register
—	0x04C-0x07C	Reserved
—	0x080-0x08C	Reserved
—	0x090-0xFCC	Reserved
—	0xFD0-0xFDC	Reserved
—	0xFE0	Reserved
—	0xFE4	Reserved
—	0xFE8	Reserved
—	0xFEC	Reserved
—	0xFF0	Reserved
—	0xFF4	Reserved
—	0xFF8	Reserved
—	0xFFC	Reserved

注) どの制御レジスタを再プログラムする場合は、事前に必ず UART をディセーブルして下さい。また、送信または受信の途中でディセーブルされると、現在の送受信を完了してから停止します。

•UART1

base アドレス = 0xF200_1000

Register Name	Address (base+)	Description
UART1DR	0x0000	UART1 Data register
UART1SR/ UART1ECR	0x0004	UART1 Receive status register/ UART1 error clear register
—	0x0008-0x0014	Reserved
UART1FR	0x0018	UART1 Flag register
—	0x001C	Reserved
—	0x0020	Reserved
UART1IBRD	0x0024	UART1 Integer baud rate register
UART1FBRD	0x0028	UART1 Fractional baud rate register
UART1LCR_H	0x002C	UART1 Line control register
UART1CR	0x0030	UART1 Control register
UART1IFLS	0x0034	UART1 Interrupt FIFO level select register
UART1IMSC	0x0038	UART1 Interrupt mask set/clear register
UART1RIS	0x003C	UART1 Raw interrupt status register
UART1MIS	0x0040	UART1 Masked interrupt status register
UART1ICR	0x0044	UART1 Interrupt clear register
—	0x0048	Reserved
—	0x004C-0x007C	Reserved
—	0x0080-0x008C	Reserved
—	0x0090-0x0FCC	Reserved
—	0x0FD0-0x0FDC	Reserved
—	0x0FE0	Reserved
—	0x0FE4	Reserved
—	0x0FE8	Reserved
—	0x0FEC	Reserved
—	0x0FF0	Reserved
—	0x0FF4	Reserved
—	0x0FF8	Reserved
—	0x0FFC	Reserved

注) どの制御レジスタを再プログラムする場合は、事前に必ず UART をディセーブルして下さい。また、送信または受信の途中でディセーブルされると、現在の送受信を完了してから停止します。

1. UART0DR (UART0 Data Register)

Address = (0xF200_0000) + (0x0000)

Bit	Bit Symbol	Type	Reset Value	Description
[31:12]	–	–	Undefined	Read as undefined. Write as zero.
[11]	OE	RO	Undefined	オーバーランエラー: リード: 0y0: FIFO に空き有り 0y1: オーバーランエラーフラグ ライト: 無効
[10]	BE	RO	Undefined	ブレークエラー リード: 0y0: エラー無し 0y1: エラー有り ライト: 無効
[9]	PE	RO	Undefined	パリティエラー リード: 0y0: エラー無し 0y1: エラー有り ライト: 無効
[8]	FE	RO	Undefined	フレーミングエラー リード: 0y0: エラー無し 0y1: エラー有り ライト: 無効
[7:0]	DATA	R/W	Undefined	リード: 受信データ ライト: 送信データ

2. UART1DR (UART1 Data Register)

Address = (0xF200_1000) + (0x0000)

Bit	Bit Symbol	Type	Reset Value	Description
[31:12]	–	–	Undefined	Read as undefined. Write as zero.
[11]	OE	RO	Undefined	オーバーランエラー: リード: 0y0: FIFO に空き有り 0y1: オーバーランエラーフラグ ライト: 無効
[10]	BE	RO	Undefined	ブレークエラー リード: 0y0: エラー無し 0y1: エラー有り ライト: 無効
[9]	PE	RO	Undefined	パリティエラー リード: 0y0: エラー無し 0y1: エラー有り ライト: 無効
[8]	FE	RO	Undefined	フレーミングエラー リード: 0y0: エラー無し 0y1: エラー有り ライト: 無効
[7:0]	DATA	R/W	Undefined	リード: 受信データ ライト: 送信データ

(個別説明)

a. <OE>

FIFO が既に Full 状態でデータを受信したとき、このビットに 1 がセットされます。FIFO が Full の状態で受信されたデータは FIFO 内に更新されずに破棄されます。

FIFO に空き空間が生じ、新しいデータを書き込めるようになると、このビットは 0 にクリアされます。

b. <BE>

ブレーク条件が検出されるとこのビットに 1 がセットされ、受信データ入力(スタートビット、データビット、パリティビット、ストップビットとして定義された)フルワード送信時間よりも長く Low で保持されたことを示します。

c. <PE>

このビットに 1 がセットされた場合は、受信されたデータのパリティが、UARTxLCR_H レジスタのビット 2 および 7 によって定義されたパリティと一致しないことを示しています。

d. <FE>

このビットに 1 がセットされた場合は、受信したデータに有効ストップビット(有効ストップビットは 1)が含まれていなかったことを示しています。

3. UART0SR/UART0ECR (UART0 Receive status register/ UART0 error clear register)

UART0SR と UART0ECR は同一のアドレスにマッピングされています。

リード時とライト時で機能が異なります。

Address = (0xF200_0000) + (0x0004)

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	—	—	—	Read as undefined.
[3]	OE	RO	0y0	オーバランエラー: 0y0: FIFO に空き有り 0y1: オーバランエラーフラグ
[2]	BE	RO	0y0	ブレークエラー 0y0: エラー無し 0y1: エラー有り
[1]	PE	RO	0y0	パリティエラー 0y0: エラー無し 0y1: エラー有り
[0]	FE	RO	0y0	フレーミングエラー 0y0: エラー無し 0y1: エラー有り

Address = (0xF200_0000) + (0x0004)

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	—	WO	—	このレジスタへの書き込みが行われると、フレーミングエラー、パリティエラー、ブレークエラー、ならびにオーバランエラーがクリアされます。データの値は関係ありません。 レジスタのアドレスは UART0SR レジスタと同じ。

4. UART1SR/ UART1ECR (UART1 Receive status register/ UART1 error clear register)

Address = (0xF200_1000) + (0x0004)

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	—	—	—	Read as undefined.
[3]	OE	RO	0y0	オーバーランエラー: 0y0: FIFO に空き有り 0y1: オーバーランエラーフラグ
[2]	BE	RO	0y0	ブレークエラー: 0y0: エラー無し 0y1: エラー有り
[1]	PE	RO	0y0	パリティエラー: 0y0: エラー無し 0y1: エラー有り
[0]	FE	RO	0y0	フレーミングエラー: 0y0: エラー無し 0y1: エラー有り

Address = (0xF200_1000) + (0x0004)

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	—	WO	—	このレジスタへの書き込みが行われると、フレーミングエラー、パリティエラー、ブレークエラー、ならびにオーバーランエラーがクリアされます。データの値は関係ありません。 レジスタのアドレスは UART1SR レジスタと同じ。

注 1) UARTxSR/UARTxECR レジスタは、受信ステータスレジスタ/エラークリアレジスタです。受信ステータスは、UARTxSR から読み出すこともできます。ステータスがこのレジスタから読み出される場合は、ブレーク、フレーミングならびにパリティに関するステータス情報は、UARTxSR の読み出し前に UARTxDR から読み出されたデータに対応します。オーバーランに関するステータス情報は、オーバーラン条件が発生するとすぐにセットされます。UARTxECR への書き込みが行われると、フレーミング、パリティ、ブレーク、オーバーランの各エラーがクリアされます。リセット時にはすべてのビットが 0 にクリアされます。

注 2) 受信データは、UARTxSR からそのデータに対応するエラーステータスを読み出す前に、先に UARTxDR から読み出す必要があります。ステータスレジスタ UARTxSR は、データレジスタ UARTxDR からの読み出しが発生した場合にのみ更新されるため、この読み出しシーケンスを逆にすることはできません。ステータス情報は UARTxDR レジスタから直接読み出すこともできます。

(個別説明)

a. <OE>

データ受信時に FIFO が既に Full 状態の場合には、このビットに 1 がセットされます。FIFO が Full の状態で受信されたデータは FIFO 内に更新されずに破棄されます。

FIFO に空き空間が生じ、新しいデータを書き込めるようになると、このビットは 0 にクリアされます。

b. <BE>

ブ레이크条件が検出されるとこのビットに 1 がセットされ、受信データ入力が(スタートビット、データビット、パリティビット、ストップビットとして定義された)フルワード送信時間よりも長く LOW で保持されたことを示します。

c. <PE>

このビットに 1 がセットされた場合は、受信されたデータデータのパリティが、UARTxLCR_H レジスタのビット 2 および 7 によって定義されたパリティと一致しないことを示しています。

d. <FE>

このビットに 1 がセットされた場合は、受信したデータに有効ストップビット(有効ストップビットは 1)が含まれていなかったことを示しています。

5. UART0FR (UART0 Flag register)

下記レジスタのビット <TXFE>、<RXFF>、<TXFF>、<RXFE>は UART0LCR_H<FEN>の状態に依存します。

(1) 送信 FIFO

送信 FIFO は、8 ビットの幅と 16 段の深さを持つ FIFO メモリバッファです。APB インタフェース経由で書き込まれた CPU データは、送信ロジックによって読み出されるまで、この FIFO にストアされます。送信 FIFO はディセーブルすることによって 1 バイト保持レジスタのように動作させることができます。

(2) 受信 FIFO

受信 FIFO は、12 ビットの幅と 16 段の深さを持つ FIFO メモリバッファです。受信データと対応するエラービットは、APB インタフェース経由で CPU によって読み出されるまで、受信ロジックによって受信 FIFO にストアされます。受信 FIFO はディセーブルすることによって 1 バイト保持レジスタのように動作させることができます。

$$\text{Address} = (0xF200_0000) + (0x0018)$$

Bit	Bit Symbol	Type	Reset Value	Description	
				FIFO mode (FEN = 1)	Character mode (FEN = 0)
[31:9]	–	–	Undefined	Read as undefined.	Read as undefined.
[8]	Reserved	RO	Undefined	Read as undefined.	Read as undefined.
[7]	TXFE	RO	0y1	送信 FIFO 空フラグ: 0y0: 空きではない 0y1: 空き	送信保持レジスタ空きフラグ 0y0: 空きではない 0y1: 空き
[6]	RXFF	RO	0y0	受信 FIFO がフルフラグ: 0y0: フルではない 0y1: フル	受信保持レジスタがフルフラグ 0y0: フルではない 0y1: フル
[5]	TXFF	RO	0y0	送信 FIFO がフルフラグ: 0y0: フルではない 0y1: フル	送信保持レジスタがフルフラグ 0y0: フルではない 0y1: フル
[4]	RXFE	RO	0y1	受信 FIFO 空フラグ: 0y0: 空きではない 0y1: 空き	受信保持レジスタが空きフラグ 0y0: 空きではない 0y1: 空き
[3]	BUSY	RO	0y0	BUSY フラグ: 0y0: 停止 0y1: データ送信中(BUSY)	BUSY フラグ: 0y0: 停止 0y1: データ送信中(BUSY)
[2]	Reserved	RO	Undefined	Read as undefined.	Read as undefined.
[1]	Reserved	RO	Undefined	Read as undefined.	Read as undefined.
[0]	Reserved	RO	Undefined	Read as undefined.	Read as undefined.

6. UART1FR (UART1 Flag register)

Address = (0xF200_1000) + (0x0018)

Bit	Bit Symbol	Type	Reset Value	Description	
				FIFO mode (FEN = 1)	Character mode (FEN = 0)
[31:8]	–	–	Undefined	Read as undefined.	Read as undefined.
[7]	TXFE	RO	0y1	送信 FIFO 空フラグ: 0y0: 空きではない 0y1: 空き	送信保持レジスタ空きフラグ 0y0: 空きではない 0y1: 空き
[6]	RXFF	RO	0y0	受信 FIFO がフルフラグ: 0y0: フルではない 0y1: フル	受信保持レジスタがフルフラグ 0y0: フルではない 0y1: フル
[5]	TXFF	RO	0y0	送信 FIFO がフルフラグ: 0y0: フルではない 0y1: フル	送信保持レジスタがフルフラグ 0y1: フル 0y0: フルではない
[4]	RXFE	RO	0y1	受信 FIFO 空フラグ: 0y0: 空きではない 0y1: 空き	受信保持レジスタが空きフラグ 0y0: 空きではない 0y1: 空き
[3]	BUSY	RO	0y0	BUSY フラグ: 0y0: 停止 0y1: データ送信中 (BUSY)	BUSY フラグ: 0y0: 停止 0y1: データ送信中 (BUSY)
[2:1]	–	–	Undefined	Read as undefined.	Read as undefined.
[0]	CTS	RO	Undefined	送信可能なクリア(CTS)フラグ 0y1: ステータス入力が 0 時	送信可能なクリア(CTS)フラグ 0y1: ステータス入力が 0 時

(個別説明)

a. <RI>

リングインジケータ(nUART1RD):モデムステータス入力が 0 の時はこのビットに 1 がセットされます。

b. <BUSY>

データ送信中に 1 をセットするビットです。シフトレジスタからストップビットを含めたデータ全てが送信されるまで、このビットはセットされたままです。

c. <DCD>

データキャリア検出(U0DCDn):モデムステータス入力が 0 の時は、このビットに 1 がセットされます。

d. <DSR>

UART データセット準備完了(U0DSRn):モデムステータス入力が 0 の時は、このビットに 1 がセットされます。

e. <CTS>

送信可能なクリア(U0CTS_n):モデムステータス入力が 0 の時は、このビットに 1 がセットされます。

7. UART0ILPR (UART0 IrDA low-power counter register)

Address = (0xF200_0000) + (0x0020)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:0]	ILPDVSR	R/W	0x00	IrDA 低電力除数: 0x01 ~ 0xFF

(個別説明)

a. <ILPDVSR>

低電力除数(ILPDVSR)=(f_{UARTCLK} / f_{IrLPBaud16})

UART0ILPR レジスタは IrDA 低電力カウンタレジスタです。この 8 ビット読み出し/ 書き込みレジスタは、UARTCLK の除算による IrLPBaud16 シグナルの生成に用いられる低電力カウンタ除数値をストアします。リセット時には全てのビットが 0 にクリアされます。上記レジスタ表は、UARTILPR レジスタのビット割り当てを示しています。

注 1) UART0CR <SIRLP>=1 に設定する前にこのレジスタを設定してください。

注 2) 0x00 は設定禁止です。0x00 をプログラムすると、IrLPBaud16 パルスは生成されません。

8. UART0IBRD (UART0 Integer baud rate register)

Address = (0xF200_0000) + (0x0024)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined. Write as zero.
[15:0]	BAUD DIVINT	R/W	0x0000	整数ボーレート除数: 0x0001 ~ 0xFFFF

9. UART1IBRD(UART1 Integer baud rate register)

Address = (0xF200_1000) + 0x0024

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined. Write as zero.
[15:0]	BAUD DIVINT	R/W	0x0000	整数ボーレート除数: 0x0001 ~ 0xFFFF

(個別説明)

a. <BAUD DIVINT>

次に説明する小数ボーレート除数と合わせて、ボーレート除数 BAUDDIV になります。

注 1) UARTxIBRD の内容を内部で更新するには、UARTxLCR_H の書き込みを常に最後に実行する必要があります。詳細は UARTxLCR_H の説明を参照してください。

注 2) UARTxCR <UARTEN>=1 に設定する前にこのレジスタを設定してください。

注 3) 0x0000 は設定禁止です。

10. UART0FBRD (UART0 Fractional baud rate register)

Address = (0xF200_0000) + (0x0028)

Bit	Bit Symbol	Type	Reset Value	Description
[31:6]	–	–	Undefined	Read as undefined. Write as zero.
[5:0]	BAUD DIVFRAC	R/W	0x00	小数ボーレート除数: 0x01 ~ 0x3F

11. UART1FBRD (UART1 Fractional baud rate register)

Address = (0xF200_1000) + (0x0028)

Bit	Bit Symbol	Type	Reset Value	Description
[31:6]	–	–	Undefined	Read as undefined. Write as zero.
[5:0]	BAUD DIVFRAC	R/W	0x00	小数ボーレート除数: 0x01 ~ 0x3F

(個別説明)

a. <BAUDDIVFRAC>

ボーレート除数は以下のように求めることができます。

ボーレート除数 $BAUDDIV = (f_{UARTCLK}) / (16 \times \text{ボーレート})$ $f_{UARTCLK}$ は、 $UARTCLK$ の周波数です。

BAUDDIV は整数値(BAUD DIVINT)と小数値(BAUD DIVFRAC)から構成されます。

注) $UARTxFBRD$ の内容を内部で更新するには、 $UARTxLCR_H$ の書き込みを常に最後に実行する必要があります。詳細は $UARTxLCR_H$ の説明を参照してください。注 2) $UARTxCR \langle UARTEN \rangle = 1$ に設定する前にこのレジスタを設定してください。

注 3) 整数ボーレートの除数設定は、0 設定は禁止ですので、少数ボーレート除数だけのボーレート設定は出来ません。また、最小ボーレートの除数設定は、整数#1、少数"1" (整数ボーレートに 1 を設定した場合は"0"の設定は禁止) となりますのでご注意ください。

例) 除数値の計算

要求されるボーレートが **230400** で $f_{UARTCLK} = 4\text{MHz}$ の場合:ボーレート除数 $= (4 \times 10^6) / (16 \times 230400) = 1.085$ したがって、 $BRDI = 1$ 、 $BRDF = 0.085$ 上記から小数部: $((0.085 \times 64) + 0.5) = 5.94$,

小数ボーレート除数の設定値は 5.94 の整数部分 0x 5 をしてください。

生成されるボーレート除数 $= 1 + 5/64 = 1.078$ 生成されるボーレート $= (4 \times 10^6) / (16 \times 1.078) = 231911$ 誤差 $= (231911 - 230400) / 230400 \times 100 = 0.656\%$ 6 ビット $UARTxFBRD$ レジスタを使用した時の最大誤差 $= 1/64 \times 100 = 1.56\%$ この誤差は $m = 1$ の時に発生し、64 クロック間の累積誤差です。

標準ボーレート設定例

 $f_{UARTCLK} = 100\text{MHz}$ 動作時

設定する整数値 (除数)	設定する小数値 (除数)	要求される ビットレート(bps)	生成される ビットレート(bps)	誤差(%)
0x1	0x1	—	6153846(最高速)	—
0xD	0x24	460800	460829.493	0.0064
0x1B	0x8	230400	230414.747	0.0064
0x36	0x10	115200	115207.373	0.0064
0x51	0x18	76800	76804.916	0.0064
0x6C	0x20	57600	57603.687	0.0064
0xA2	0x31	38400	38398.771	-0.0032
0x145	0x21	19200	19200.307	0.0016
0x1B2	0x2	14400	14399.885	-0.0008
0x28B	0x3	9600	9599.923	-0.0008
0xA2C	0xB	2400	2399.995	-0.0002
0x1458	0x15	1200	1200.001	0.0001
0xDDF2	0xC	110	109.99999	-1.00E-05

 $f_{UARTCLK} = 96\text{MHz}$ 動作時

設定する整数値 (除数)	設定する小数値 (除数)	要求される ビットレート(bps)	生成される ビットレート(bps)	誤差(%)
0x1	0x1	—	5907692(最高速)	—
0xD	0x1	460800	460984.394	0.0400
0x1A	0x3	230400	230353.929	-0.0200
0x34	0x5	115200	115211.521	0.0100
0x4E	0x8	76800	76800.000	0
0x68	0xB	57600	57597.120	-0.0050
0x9C	0x10	38400	38400.000	0
0x138	0x20	19200	19200.000	0
0x1A0	0x2B	14400	14399.820	-0.0012
0x271	0x1	9600	9599.760	-0.0025
0x9C4	0x1	2400	2399.985	-0.0006
0x1388	0x1	1200	1199.996	-0.0003
0xD511	0x1D	110	110.000	2.60E-06

 $f_{UARTCLK} = 25\text{MHz}$ 動作時

設定する整数値 (除数)	設定する小数値 (除数)	要求される ビットレート(bps)	生成される ビットレート(bps)	誤差(%)
0x1	0x1	—	1538461(最高速)	—
0x3	0x19	460800	460829.493	0.0064
0x6	0x32	230400	230414.747	0.0064
0xD	0x24	115200	115207.373	0.0064
0x14	0x16	76800	76804.916	0.0064
0x1B	0x8	57600	57603.687	0.0064
0x28	0x2C	38400	38402.458	0.0064
0x51	0x18	19200	19201.229	0.0064
0x6C	0x20	14400	14400.922	0.0064
0xA2	0x31	9600	9599.693	-0.0032
0x28B	0x3	2400	2399.981	-0.0008
0x516	0x5	1200	1200.005	0.0004
0x377C	0x23	110	110.000	-1.00E-05

12. UART0LCR_H (UART0 Line control register)

Address = (0xF200_0000) + (0x002C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7]	SPS	R/W	0y0	スティックパリティ選択: 表 3.13.2 真理値表参照
[6:5]	WLEN	R/W	0y00	ワード長: 0y00: 5 ビット 0y01: 6 ビット 0y10: 7 ビット 0y11: 8 ビット
[4]	FEN	R/W	0y0	FIFO 制御 0y0: キャラクタモード(FIFO 無効) 0y1: FIFO モード
[3]	STP2	R/W	0y0	ストップビット選択 0y0: 1 ビット 0y1: 2 ビット
[2]	EPS	R/W	0y0	偶数パリティ選択(表 3.13.2 真理値表参照) 0y0: 奇数 0y1: 偶数
[1]	PEN	R/W	0y0	パリティ制御(表 3.13.2 真理値表に参照) 0y0: Disable 0y1: Enable
[0]	BRK	R/W	0y0	送信ブレーク 0y0: 無効 0y1: 送信ブレーク

13. UART1LCR_H (UART1 Line control register)

Address = (0xF200_1000) + (0x002C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7]	SPS	R/W	0y0	スティックパリティ選択: 表 3.13.2 真理値表に参照
[6:5]	WLEN	R/W	0y00	ワード長: 0y00: 5 ビット 0y01: 6 ビット 0y10: 7 ビット 0y11: 8 ビット
[4]	FEN	R/W	0y0	FIFO 制御 0y0: キャラクタモード(FIFO 無効) 0y1: FIFO モード
[3]	STP2	R/W	0y0	ストップビット選択 0y0: 1 ビット 0y1: 2 ビット
[2]	EPS	R/W	0y0	偶数パリティ選択(表 3.13.2 真理値表に参照) 0y0: 奇数 0y1: 偶数
[1]	PEN	R/W	0y0	パリティ制御(表 3.13.2 真理値表に参照) 0y0: Disable 0y1: Enable
[0]	BRK	R/W	0y0	送信ブレーク 0y0: 無効 0y1: 送信ブレーク

(個別説明)

a. <SPS>

UARTxLCR_H レジスタのビット 1、2、7 がセットされている場合は、パリティビットが送信され、0 としてチェックされます。ビット 1 と 7 がセットされ、ビット 2 が 0 の場合には、パリティビットが送信され、1 としてチェックされます。このビットがクリアされると、ステイックパリティがディセーブルされます。SPS、EPS ならびにPENビットの真理値表については、表 3.13.2を参照して下さい。

b. <WLEN>

フレームで送信または受信されたデータビットの数を示します。

c. <FEN>

このビットに 1 がセットされると、送信および受信 FIFO バッファがイネーブルされます (FIFO モード)。

このビットが 0 にクリアされると、これらの FIFO はディセーブルされ (キャラクタモード)、1 段の深さを持つ保持レジスタとなります。

d. <STP2>

このビットに 1 がセットされると、フレームの終わりに 2 ビットのストップビットが付加されて送信されます。受信ロジックは、受信中の 2 ビット (目) のストップビットをチェックしません。

e. <EPS>

このビットに 1 がセットされると、送信中または受信中に偶数パリティの生成とチェックが実行されます。このチェックでは、データビットとパリティビットに含まれる 1 の数が偶数個かどうかチェックされます。このビットが 0 にクリアされると、1 の数が奇数個かどうかをチェックする奇数パリティチェックが実行されます。パリティイネーブルビット <PEN>=0 の場合には、このビットは無効になります。表 3.13.2を参照してください。

f. <PEN>

このビットに 1 がセットされている場合はパリティのチェックおよび生成がイネーブルされます。それ以外の場合にはパリティがディセーブルされ、データフレームにパリティビットが追加されません。SPS、EPS ならびにPEN ビットの真理値表については、表 3.13.2を参照して下さい。

g. <BRK>

このビットに 1 がセットされている場合は、現在のキャラクタの送信完了後、UxTXD 出力に LOW レベルになります。送信ブレークを生成するためには、少なくとも 1 フレーム以上の送信時間このビットをアサートする必要があります。ブレーク条件が生成されても、送信 FIFO の内容は影響を受けません。

注) UARTxLCR_H、UARTxIBRD、UARTxFBRD のレジスタを設定する場合は、最後に UARTxLCR_H を設定する必要があります。

また、UARTxIBRD または UARTxFBRD だけを更新する場合でも、再度 UARTxLCR_H を設定してください。

下記の表は UARTxLCR_H<SPS>、<EPS> ならびに<PEN>の真理値表を示しています。

表 3.13.2 UARTxLCR_H <SPS>、<EPS> ならびに<PEN>の真理値

パリティ イネーブル (PEN)	偶数パリティ 選択(EPS)	スティック パリティ選択 (SPS)	パリティビット(送信またはチェック)
0	x	x	送信およびチェックなし
1	1	0	偶数パリティ
1	0	0	奇数パリティ
1	0	1	1
1	1	1	0

14. UART0CR (UART0 Control register)

Address = (0xF200_0000) + (0x0030)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined. Write as zero.
[15]	Reserved	R/W	0y0	Read as undefined. Write as zero.
[14]	Reserved	R/W	0y0	Read as undefined. Write as zero.
[13:12]	–	–	Undefined	Read as undefined. Write as zero.
[11]	Reserved	R/W	0y0	Read as undefined. Write as zero.
[10]	Reserved	R/W	0y0	Read as undefined. Write as zero.
[9]	RXE	R/W	0y1	UART の受信イネーブル 0y0: Disable 0y1: Enable
[8]	TXE	R/W	0y1	UART の送信イネーブル 0y0: Disable 0y1: Enable
[7]	Reserved	R/W	0y0	0 をライトしてください。
[6:3]	Reserved	–	Undefined	Read as undefined. Write as zero.
[2]	SIRLP	R/W	0y0	IrDA 送信データが 0 ビットのエンコーディングモードの選択 0y0: ビット周期の 3/16 の幅を持つ HIGH アクティブパルスとして送信されます。 0y1: IrLPBaud16 入力 シグナル周期の 3 倍のパルス幅
[1]	SIREN	R/W	0y0	SIR イネーブル 0y0: Disable 0y1: Enable
[0]	UARTEN	R/W	0y0	UART イネーブル 0y0: Disable 0y1: Enable

15. UART1CR (UART1 Control register)

Address = (0xF200_1000) + (0x0030)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined. Write as zero.
[15]	CTSEN	R/W	0y0	CTS ハードウェアフロー制御イネーブル 0y0 : Disable 0y1 : Enable
[14:10]	–	–	Undefined	Read as undefined. Write as zero.
[9]	RXE	R/W	0y1	UART の受信イネーブル 0y0: Disable 0y1: Enable
[8]	TXE	R/W	0y1	UART の送信イネーブル 0y0: Disable 0y1: Enable
[7]	Reserved	R/W	0y0	0 をライトしてください。
[6:1]	Reserved	–	Undefined	Read as undefined. Write as zero.
[0]	UARTEN	R/W	0y0	UART イネーブル 0y0: Disable 0y1: Enable

(個別説明)

a. <CTSEN>

このビットに 1 がセットされると、CTS ハードウェアフロー制御がイネーブルされます。データは、UxCTS_n シグナルがアサートされて初めて送信されます。

b. <RTSEN>

このビットに 1 がセットされると、RTS ハードウェアフロー制御がイネーブルされます。受信 FIFO に空きがある場合にのみ要求されます。

c. <RTS>

このビットは、UART 送信要求(UxRTS_n)モデムステータス出力信号です。このビットに 1 が設定されていると、0 を出力します。

d. <DTR>

このビットは、UART データ送信準備完了(UxDTR_n)モデムステータス出力信号です。このビットに 1 が設定されていると、0 を出力します。

e. <RXE>

このビットに 1 を設定すると、UART の受信回路がイネーブルされます。<SIREN>に基づいて UART 機能または SIR 機能のどちらかでデータを受信します。受信の途中で UART がディセーブルされると、現在の受信を完了してから停止します。

f. <TXE>

このビットに 1 を設定すると、UART の送信回路がイネーブルされます。<SIREN>に基づいて、UART 機能または SIR 機能のどちらかでデータを送信します。送信の途中で UART がディセーブルされると、現在の送信を完了してから停止します。

g. <SIRLP>

<SIRLP>は IrDA エンコーディングモードを選択します。<SIRLP>に 0 を設定すると、

IrDA 送信データが 0 のビットはビット周期の 3/16 の幅を持つ HIGH アクティブパルス (SIR0OUT)として送信されます。

<SIRLP>に 1 を設定すると、IrDA 送信データが 0 のビットは IrLPBaud16 入力シグナル周期の 3 倍のパルス幅を使用して送信されます。このビットを設定すると消費電力を軽減できますが、送信距離が短くなる可能性があります。

h. <SIREN>

このビットに 1 を設定すると、IrDA 回路がイネーブルになります。UART 回路を動作させるためには<UARTEN>を 1 に設定する必要があります。IrDA 回路がイネーブルの場合、SIR0OUT および SIR0IN 端子が有効になります。U0TXD 端子は 1 をセットした状態になります。U0RXD 端子 とモデムステータス入力におけるシグナル遷移は無効です。IrDA 回路がディセーブルされると、SIR0OUT は 0 になり (光パルスが生成されない)、SIR0IN 端子は無効となります。

i. <UARTEN>

このビットに 1 を設定すると UART 回路 がイネーブルされます。データの送信および受信は、<SIREN>に基づいて、UART 機能または SIR 機能のどちらかで動作します。UART は送信または受信の途中でディセーブルされると、現在の送受信が完了してから停止します。

16. UART0IFLS (UART0 Interrupt FIFO level select register)

Address = (0xF200_0000) + (0x0034)

Bit	Bit Symbol	Type	Reset Value	Description
[31:6]	–	–	Undefined	Read as undefined. Write as zero.
[5:3]	RXIFLSEL	R/W	0y010	受信割り込み FIFO レベル選択(1 word = 12 bit): 0y000: 2word 目が受信 FIFO に格納されたとき 0y001: 4 word 目が受信 FIFO に格納されたとき 0y010: 8 word 目が受信 FIFO に格納されたとき 0y011: 12 word 目が受信 FIFO に格納されたとき 0y100: 14 word 目が受信 FIFO に格納されたとき 0y101~0y111: Reserved
[2:0]	TXIFLSEL	R/W	0y010	送信割り込み FIFO レベル選択(1 word = 8 bit): 0y000: 送信 FIFO が残り 2 word になったとき 0y001: 送信 FIFO が残り 4 word になったとき 0y010: 送信 FIFO が残り 8 word になったとき 0y011: 送信 FIFO が残り 12 word になったとき 0y100: 送信 FIFO が残り 14word になったとき 0y101~0y111: Reserved

17. UART1IFLS (UART1 Interrupt FIFO level select register)

Address = (0xF200_1000) + (0x0034)

Bit	Bit Symbol	Type	Reset Value	Description
[31:6]	–	–	Undefined	Read as undefined. Write as zero.
[5:3]	RXIFLSEL	R/W	0y010	受信割り込み FIFO レベル選択(1 word = 12 bit): 0y000: 2 word 目が受信 FIFO に格納されたとき 0y001: 4 word 目が受信 FIFO に格納されたとき 0y010: 8 word 目が受信 FIFO に格納されたとき 0y011: 12 word 目が受信 FIFO に格納されたとき 0y100: 14 word 目が受信 FIFO に格納されたとき 0y101~0y111: Reserved
[2:0]	TXIFLSEL	R/W	0y010	送信割り込み FIFO レベル選択(1 word = 8 bit): 0y000: 送信 FIFO が残り 2 word になったとき 0y001: 送信 FIFO が残り 4 word になったとき 0y010: 送信 FIFO が残り 8 word になったとき 0y011: 送信 FIFO が残り 12 word になったとき 0y100: 送信 FIFO が残り 14 word になったとき 0y101~0y111: Reserved

(個別説明)

UARTxIFLS レジスタは割り込み FIFO レベル選択レジスタです。この UARTxIFLS レジスタを使用して、UARTTXINTR および UARTRXINTR が発生する FIFO レベルを定義しています。

割り込みは、FIFO レベルに基づくのではなく、そのレベルを経由する遷移に基づいて生成されます。例えば、2word のデータ受信後、3word 目のデータが受信 FIFO に格納されたときに割り込みが発生されます。

18. UART0IMSC (UART0 Interrupt mask set/clear register)

Address = (0xF200_0000) + (0x0038)

Bit	Bit Symbol	Type	Reset Value	Description
[31:11]	–	–	Undefined	Read as undefined. Write as zero.
[10]	OEIM	R/W	0y0	オーバーランエラー割り込み許可 0y0: 許可しない 0y1: 許可
[9]	BEIM	R/W	0y0	ブレークエラー割り込み許可 0y0: 許可しない 0y1: 許可
[8]	PEIM	R/W	0y0	パリティエラー割り込み許可 0y0: 許可しない 0y1: 許可
[7]	FEIM	R/W	0y0	フレーミングエラー割り込み許可 0y0: 許可しない 0y1: 許可
[6]	RTIM	R/W	0y0	受信タイムアウト割り込み許可 0y0: 許可しない 0y1: 許可
[5]	TXIM	R/W	0y0	送信 FIFO 割り込み許可 0y0: 許可しない 0y1: 許可
[4]	RXIM	R/W	0y0	受信 FIFO 割り込み許可 0y0: 許可しない 0y1: 許可
[3]	Reserved	R/W	0y0	Read as undefined. Write as zero.
[2]	Reserved	R/W	0y0	Read as undefined. Write as zero.
[1]	Reserved	R/W	0y0	Read as undefined. Write as zero.
[0]	Reserved	R/W	0y0	Read as undefined. Write as zero.

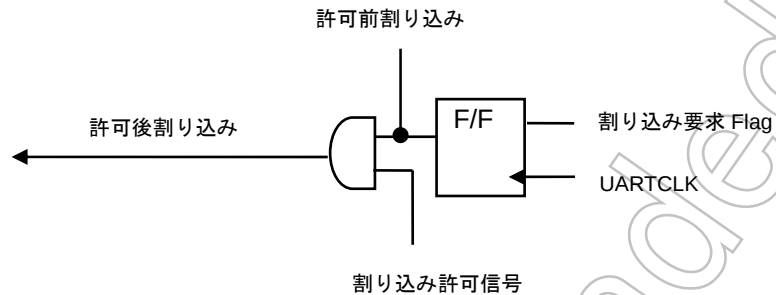
19. UART1IMSC (UART1 Interrupt mask set/clear register)

Address = (0xF200_1000) + (0x0038)

Bit	Bit Symbol	Type	Reset Value	Description
[31:11]	–	–	Undefined	Read as undefined. Write as zero.
[10]	OEIM	R/W	0y0	オーバーランエラー割り込み許可 0y0: 許可しない 0y1: 許可
[9]	BEIM	R/W	0y0	ブレークエラー割り込み許可 0y0: 許可しない 0y1: 許可
[8]	PEIM	R/W	0y0	パリティエラー割り込み許可 0y0: 許可しない 0y1: 許可
[7]	FEIM	R/W	0y0	フレーミングエラー割り込み許可 0y0: 許可をクリア 0y1: 許可をセット
[6]	RTIM	R/W	0y0	受信タイムアウト割り込み許可 0y0: 許可しない 0y1: 許可
[5]	TXIM	R/W	0y0	送信割り込み許可 0y0: 許可しない 0y1: 許可
[4]	RXIM	R/W	0y0	受信割り込み許可 0y0: 許可しない 0y1: 許可
[3:2]	–	–	Undefined	Read as undefined. Write as zero.
[1]	CTSMIM	R/W	0y0	U0CTS _n 割り込み許可 0y0: 許可しない 0y1: 許可
[0]	–	–	Undefined	Read as undefined. Write as zero.

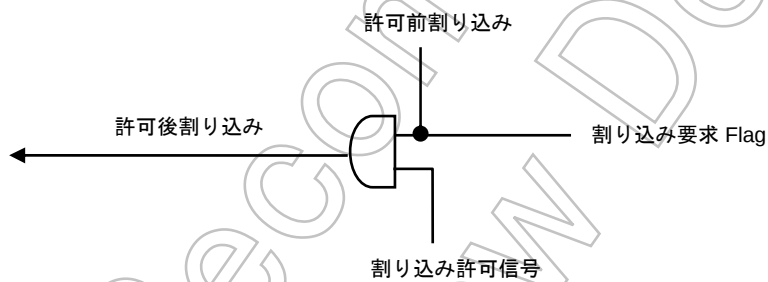
- UART 割り込み発生ブロック図

(1) エラーフラグ(BE, PE, FE) ブロック図



- 割り込み要求 Flag はリアルタイムに変化し F/F で保持します。割り込み要求をクリアする場合、割り込みクリアレジスタの該当 bit でクリアします。

(2) エラーフラグ(OE) ブロック図



- 割り込み要求 Flag の OE オーバーランフラグはリアルタイムで変化しますが、割り込みは保存されません。割り込み要求クリアは受信 FIFO のリードによってクリアされます。

20. UART0RIS (UART0 Raw interrupt status register)

Address = (0xF200_0000) + (0x003C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:11]	–	–	Undefined	Read as undefined.
[10]	OERIS	RO	0y0	オーバランエラー許可前割り込みステータス 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[9]	BERIS	RO	0y0	ブレークエラー許可前割り込みステータス 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[8]	PERIS	RO	0y0	パリティエラー許可前割り込みステータス 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[7]	FERIS	RO	0y0	フレーミングエラー許可前割り込みステータス 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[6]	RTRIS	RO	0y0	受信タイムアウト許可前割り込みステータス 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[5]	TXRIS	RO	0y0	送信許可前割り込みステータス 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[4]	RXRIS	RO	0y0	受信許可前割り込みステータス 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[3]	Reserved	RO	Undefined	Read as undefined.
[2]	Reserved	RO	Undefined	Read as undefined.
[1]	Reserved	RO	Undefined	Read as undefined.
[0]	Reserved	RO	Undefined	Read as undefined.

注) リセット時には、モデム許可前割り込みステータスビット(ビット3 ~ 0)を除いた全てのビットが0にクリアされます。リセット後のモデム許可前割り込みステータスビットは定義されていません。

21. UART1RIS (UART1 Raw interrupt status register)

Address = (0xF200_1000) + (0x003C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:11]	–	–	Undefined	Read as undefined.
[10]	OERIS	RO	0y0	オーバーランエラー許可前割り込みステータス 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[9]	BERIS	RO	0y0	ブレークエラー許可前割り込みステータス 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[8]	PERIS	RO	0y0	パリティエラー許可前割り込みステータス 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[7]	FERIS	RO	0y0	フレーミングエラー許可前割り込みステータス 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[6]	RTRIS	RO	0y0	受信タイムアウト許可前割り込みステータス 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[5]	TXRIS	RO	0y0	送信許可前割り込みステータス 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[4]	RXRIS	RO	0y0	受信許可前割り込みステータス 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[3:2]	–	–	Undefined	Read as undefined.
[1]	CTSRMIS	RO	Undefined	U0CTS _n モデム許可前割り込みステータス 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[0]	–	–	Undefined	Read as undefined.

注) リセット時には、モデム許可前割り込みステータスビット(ビット1)を除いた全てのビットが0にクリアされます。リセット後のモデム許可前割り込みステータスビットは定義されていません。

22. UART0MIS (UART0 Masked interrupt status register)

Address = (0xF200_0000) + (0x0040)

Bit	Bit Symbol	Type	Reset Value	Description
[31:11]	–	–	Undefined	Read as undefined.
[10]	OEMIS	RO	0y0	オーバランエラー許可後割り込みステータス 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[9]	BEMIS	RO	0y0	ブレークエラー許可後割り込みステータス 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[8]	PEMIS	RO	0y0	パリティエラー許可後割り込みステータス 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[7]	FEMIS	RO	0y0	フレーミングエラー許可後割り込みステータス 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[6]	RTMIS	RO	0y0	受信タイムアウト許可後割り込みステータス 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[5]	TXMIS	RO	0y0	送信許可後割り込みステータス 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[4]	RXMIS	RO	0y0	受信許可後割り込みステータス 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[3]	Reserved	RO	Undefined	Read as undefined.
[2]	Reserved	RO	Undefined	Read as undefined.
[1]	Reserved	RO	Undefined	Read as undefined.
[0]	Reserved	RO	Undefined	Read as undefined.

注) リセット時には、モデム許可後割り込みステータスビット(ビット3～0)を除いた全てのビットが0にクリアされます。リセット後のモデム許可後割り込みステータスビットは定義されていません。

23. UART1MIS (UART1 Masked interrupt status register)

Address = (0xF200_1000) + (0x0040)

Bit	Bit Symbol	Type	Reset Value	Description
[31:11]	–	–	Undefined	Read as undefined.
[10]	OEMIS	RO	0y0	オーバーランエラー許可後割り込みステータス 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[9]	BEMIS	RO	0y0	ブレークエラー許可後割り込みステータス 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[8]	PEMIS	RO	0y0	パリティエラー許可後割り込みステータス 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[7]	FEMIS	RO	0y0	フレーミングエラー許可後割り込みステータス 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[6]	RTMIS	RO	0y0	受信タイムアウト許可後割り込みステータス 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[5]	TXMIS	RO	0y0	送信許可後割り込みステータス 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[4]	RXMIS	RO	0y0	受信許可後割り込みステータス 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[3:2]	–	–	Undefined	Read as undefined.
[1]	CTSMMIS	RO	Undefined	U0CTS _n 許可後割り込みステータス 0y0: 割り込み要求無し 0y1: 割り込み要求有り
[0]	–	–	Undefined	Read as undefined.

24. UART0ICR (UART0 Interrupt clear register)

Address = (0xF200_0000) + (0x0044)

Bit	Bit Symbol	Type	Reset Value	Description
[31:11]	–	–	Undefined	Write as zero.
[10]	OEIC	WO	Undefined	オーバランエラー割り込みをクリア 0y0: 無効 0y1: 割り込みをクリア
[9]	BEIC	WO	Undefined	ブレークエラー割り込みをクリア 0y0: 無効 0y1: 割り込みをクリア
[8]	PEIC	WO	Undefined	パリティエラー割り込みをクリア 0y0: 無効 0y1: 割り込みをクリア
[7]	FEIC	WO	Undefined	フレーミングエラー割り込みをクリア 0y0: 無効 0y1: 割り込みをクリア
[6]	RTIC	WO	Undefined	受信タイムアウト割り込みをクリア 0y0: 無効 0y1: 割り込みをクリア
[5]	TXIC	WO	Undefined	送信割り込みをクリア 0y0: 無効 0y1: 割り込みをクリア
[4]	RXIC	WO	Undefined	受信割り込みをクリア 0y0: 無効 0y1: 割り込みをクリア
[3]	Reserved	WO	Undefined	Write as zero.
[2]	Reserved	WO	Undefined	Write as zero.
[1]	Reserved	WO	Undefined	Write as zero.
[0]	Reserved	WO	Undefined	Write as zero.

注) UART0ICR レジスタは、書き込み専用の割り込みクリアレジスタです。このレジスタに 1 を書き込むと、対応する割り込みがクリアされます。0 の書き込みは無効です。

25. UART1ICR (UART1 Interrupt clear register)

Address = (0xF200_1000) + (0x0044)

Bit	Bit Symbol	Type	Reset Value	Description
[31:11]	–	–	Undefined	Read as undefined. Write as zero.
[10]	OEIC	WO	Undefined	オーバランエラー割り込みをクリア 0y0: 無効 0y1: 割り込みをクリア
[9]	BEIC	WO	Undefined	ブレークエラー割り込みをクリア 0y0: 無効 0y1: 割り込みをクリア
[8]	PEIC	WO	Undefined	パリティエラー割り込みをクリア 0y0: 無効 0y1: 割り込みをクリア
[7]	FEIC	WO	Undefined	フレーミングエラー割り込みをクリア 0y0: 無効 0y1: 割り込みをクリア
[6]	RTIC	WO	Undefined	受信タイムアウト割り込みをクリア 0y0: 無効 0y1: 割り込みをクリア
[5]	TXIC	WO	Undefined	送信割り込みをクリア 0y0: 無効 0y1: 割り込みをクリア
[4]	RXIC	WO	Undefined	受信割り込みをクリア 0y0: 無効 0y1: 割り込みをクリア
[3:2]	–	–	Undefined	Read as undefined. Write as zero.
[1]	CTSMIC	WO	Undefined	U0CTS _n 割り込みをクリア 0y0: 無効 0y1: 割り込みをクリア
[0]	–	–	Undefined	Read as undefined. Write as zero.

注) UART1ICR レジスタは、書き込み専用の割り込みクリアレジスタです。このレジスタに 1 を書き込むと、対応する割り込みがクリアされます。0 の書き込みは無効です。

26. UART0DMACR (UART0 DMA control register)

Address = (0xF200_0000) + (0x0048)

Bit	Bit Symbol	Type	Reset Value	Description
[31:3]	—	—	—	Read as undefined. Write as zero.
[2]	DMAONERR	R/W	0y0	DMA オンエラー 0y1: 有り 0y0: 無し
[1]	TXDMAE	R/W	0y0	送信 FIFO の DMA イネーブル 0y0: Disable 0y1: Enable
[0]	RXDMAE	R/W	0y0	受信 FIFO の DMA イネーブル 0y0: Disable 0y1: Enable

注 1) 例えば、19 キャラクタを受信する必要がある時に、ウォーターマークレベルが 4 になるようにプログラムされている場合、DMA コントローラは 4 キャラクタのバーストを 4 回転送し、3 回のシングル転送を行ってストリームを完了させます。

注 2) DMAC を使って、送信 / 受信 FIFO のデータ転送する場合は、バス幅は 8bit に設定してください。

(個別説明)

a. <DMAONERR>

このビットに 1 がセットされると、UART エラー割り込みがアサートされた時に DMA 受信要求出力 UARTxRXDMASREQ または UARTxRXDMABREQ がディセーブルされます。

3.14 I²C

3.14.1 概要

本モジュールは、I²C バスモードで動作し、標準的な I²C バス規格(PHILIPS 仕様)に準拠します(注 1)。

主な機能は以下のとおりです。

- Ch0 の 1 チャンネル内蔵
- マスタ/スレーブ送受信選択可能
- 送信/受信選択可能
- マルチマスタ対応(アービトレーション有り、クロック同期認識)
- 通信速度(STANDARD モード/FAST モード対応 ...

マスタ時の最高通信ボーレート:89.91kHz/357.14kHz@f_{PCLK}=100MHz)

- アドレッシングフォーマット 7bit のみサポート
- 転送データサイズ 1~8bit
- 転送(送信 or 受信)完了割込み(レベル)1 本
- 割込みの許可/禁止設定可能(I²C Ch0 の割込み要因 INTS[6])

また独自フォーマットのフリーデータフォーマットに対応しています。

注 1) FAST MODE に準拠。ただし、以下の項目を除きます。

注 2) 本モジュールは I²C バス規格に一部非対応です。

I ² C バス規格 項目	I ² C 規格	本 IP
STANDARD モード対応 (~100kHz)	必要	対応
FAST モード対応 (~400kHz)	必要	対応
Hs (High speed) モード対応 (~3.4Mbps)	必要	非対応
7 ビットアドレッシング対応	必要	対応
10 ビットアドレッシング対応	必要	非対応
START バイト	必要	非対応
ノイズキャンセラ対応	必要	対応(デジタル)
スロープ制御対応	必要	非対応
電源 OFF 時の I/O 対応	必要	非対応
シュミット (VIH/VIL)	VDDx0.3 / VDDx0.7	対応
出力電流@V _{OL} =0.4V, VDD>2V	3mA	対応

3.14.1.1 I²Cバスモード

I²C バスは I2C0DA と I2C0CL を通して、デバイスがバスに接続されるバスで、複数のデバイスと通信が可能です。

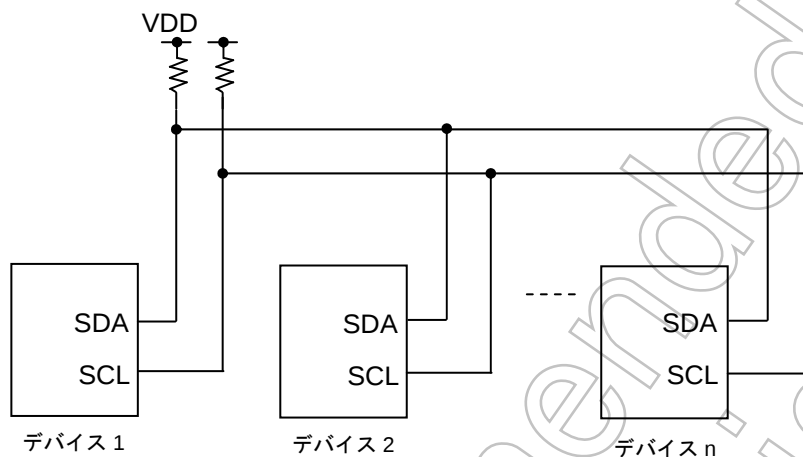


図 3.14.1 デバイスの接続

I²C バス上のマスタ/スレーブデバイスとして動作します。マスタデバイスは、バス上のシリアルクロックライン(SCL)のドライブ、8bit アドレス送信、1~8bit のデータ送信/受信を行います。スレーブデバイスは、バス上のシリアルクロックに同期して、8bit アドレスの受信、1~8bit のシリアルデータの送信/受信を行います。

マスタ/スレーブに関わらず、受信動作をしたデバイスはシリアルデータ受信後アクノリッジ信号を出力することができ、送信動作をしたデバイスはそのアクノリッジを受けることができます。マスタはそのアクノリッジのためのクロックを出力することができます。

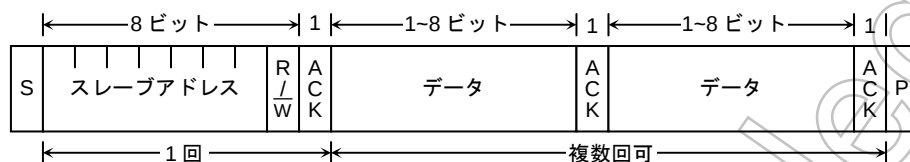
また、同一バス上に複数のマスタのあるマルチマスタ時、シリアルクロック同期化、およびシリアルデータの整合性を保つためのアービトレーションロストをサポートしています。

3.14.2 I²Cバスモード時のデータフォーマット

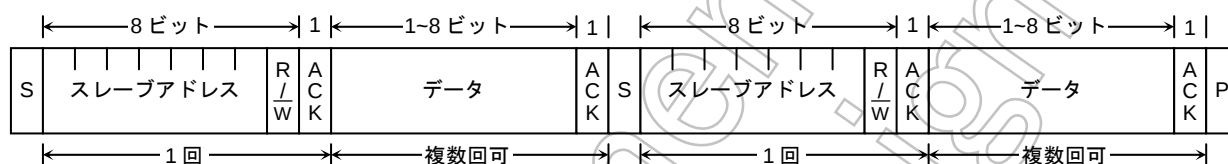
I²Cバスモード時のデータフォーマットを下図に示します。

3.14.2.1 アドレッシングフォーマット

(a) アドレッシングフォーマット



(b) アドレッシングフォーマット (再スタートあり)



S: スタートコンディション
R/W: 方向ビット
ACK: アクノリッジビット
P: ストップコンディション

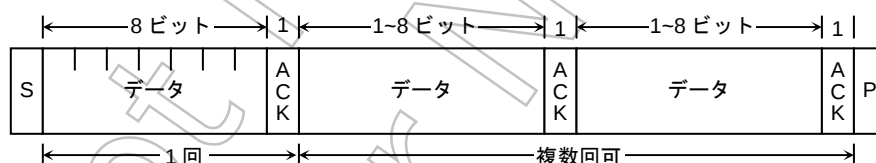
図 3.14.2 I²Cバスモード時のデータフォーマット

3.14.2.2 フリーデータフォーマット

フリーデータフォーマットは、1つのマスタと1つのスレーブ同士の通信です。

フリーデータフォーマットでは、スレーブアドレス、方向ビットはデータとして扱われます。

(a) フリーデータフォーマット (マスタデバイスからスレーブデバイスへデータを転送する転送フォーマット)



S: スタートコンディション
R/W: 方向ビット
ACK: アクノリッジビット
P: ストップコンディション

図 3.14.3 I²Cバスモード時のフリーデータフォーマット

3.14.3 ブロック図

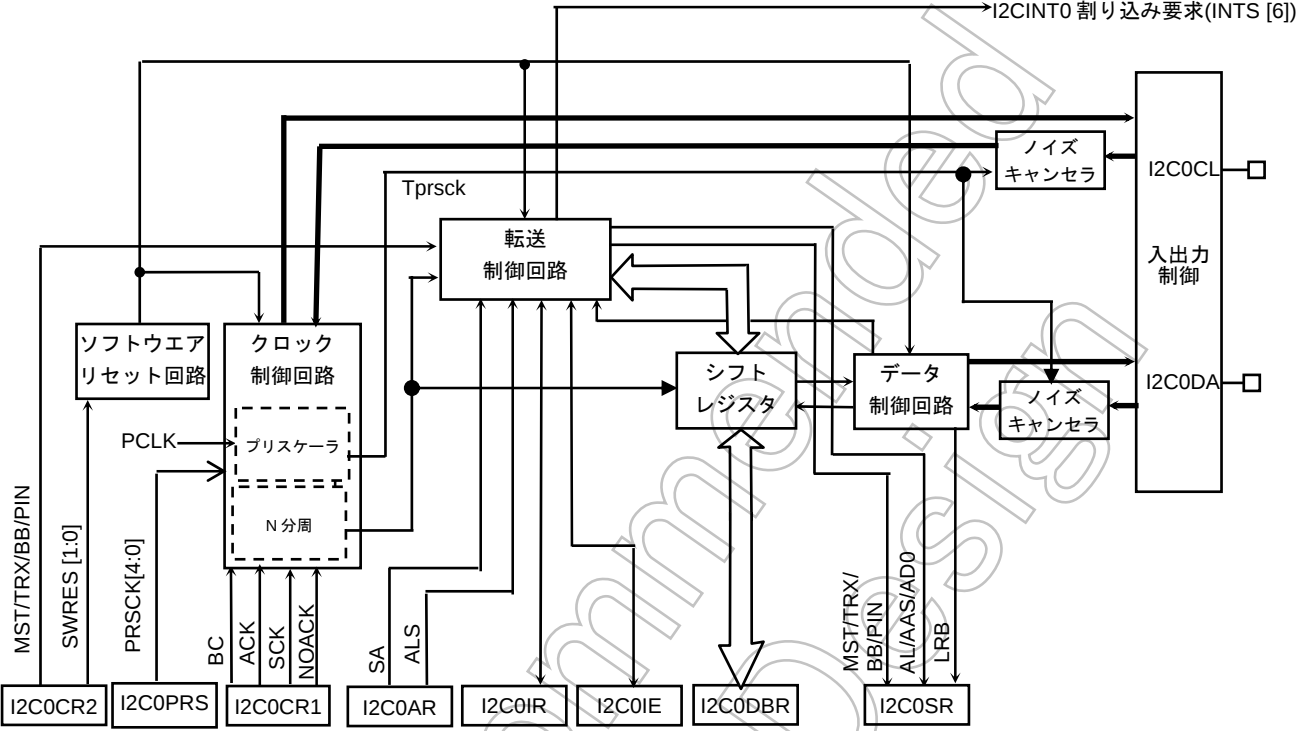


図 3.14.4 I²C Channel 0

3.14.4 動作説明

3.14.4.1 I²C バスモード時のデータ転送手順

1. デバイスの初期化

I2C0DA と I2C0CL 端子の状態が High(バスフリー)になっていることを確認し、I2C0CR2<I2CM>に 1 を設定し、I²C をイネーブルに設定します。

次に、I2C0CR1<ACK>に 1 を、I2C0CR1<NOACK>に 0 を、I2C0CR1<BC>に 0y000 をそれぞれ書き込み、アクノリッジのためのクロック数を“カウントする”、スレーブアドレス一致検出およびゼネラルコール検出を”許可”に、データ長を“8 ビット”にそれぞれ設定します。また、I2C0CR1<SCK>で t_{HIGH}、t_{LOW} を設定します。

次に I2C0AR<SA>にスレーブアドレスを設定し、I2C0AR<ALS>を 0 に設定してアドレッシングフォーマットを設定します。

最後に、I2C0CR2<MST>、I2C0CR2<TRX>、I2C0CR2<BB>に 0 を、I2C0CR2<PIN>に 1 を、I2C0CR2<SWRES[1:0]>に 0y00 を設定し、初期状態をスレーブレシーバモードにします。

注) I²C の初期化は、バスに接続されているすべてのデバイスが初期化された後に、どのデバイスも一定期間スタートコンディションを発生しない期間内に終了するようにしてください。この制約が守られない場合、I²C の初期化が終了する前に、ほかのデバイスが転送を開始することがあり、正常にデータを受信することができません。

(プログラム例)デバイスの初期化

```
CHK_PORT: r1      ← (GPIOCDATA) ; 外部端子状態が“H” を確認
          CMP      r1, #0xC0
          BNE      CHK_PORT
          (I2C0CR2) ← 0x18      ; I2C を許可
          (I2C0CR1) ← 0x16      ; アクノリジメントモード、I2C0CR1<SCK>= 0y110 に設定
          (I2C0AR)  ← 0xA0      ; スレーブアドレスを 1010000 に、アドレッシングフォーマットに設定
          (I2C0CR2) ← 0x18      ; スレーブレシーバモードに設定
```

2. スタートコンディション、スレーブアドレスの発生

バスフリー (I2C0SR<BB> = 0) 状態を確認します。

I2C0CR1<ACK>を 1 にセットし、I2C0DBR に送信スレーブアドレスと方向ビットのデータを書き込みます。I2C0CR2<MST>, I2C0CR2<TRX>, I2C0CR2<BB>, I2C0CR2<PIN>に 1 を書き込むと、バス上にスタートコンディションと、スレーブアドレスおよび方向ビットが出力されます。なお、スタートコンディション出力後、I2C0CL 端子が立ち下がるまで、 t_{HIGH} の時間がかかります。

この後、I2C0CL の 9 クロック目の立ち下がり、I2CINT0 割り込み要求が発生し、I2C0SR<PIN>が 0 にクリアされます。I2C0SR<PIN>が 0 の間、I2C0CL を “L” レベルに引きます。スレーブデバイスからアクノリッジ信号が返ってきたときのみハードウェアにより、I2CINT0 割り込み要求発生タイミングで I2C0SR<TRX>が方向ビットに合わせて変化します。

注 1) スレーブアドレスを出力するために I2C0DBR を書き込むときは、事前に、ソフトウェアによってバスフリーを検出してから行ってください。

注 2) スレーブアドレスを書き込んでからスタートコンディションを発生させるまでの間に、ほかのマスタによる転送が行われる場合があります。そのため、出力するスレーブアドレスの書き込みから 98.0 μ s (STANDARD モード I²C バス規格による最短の転送時間)、または 23.7 μ s (FAST モード I²C バス規格による最短の転送時間) 以内に、再度ソフトウェアによるバスフリーの確認を行い、バスフリーが確認できたときのみスタートコンディションを発生させてください。

(プログラム例) スタートコンディションの発生

```
CHK_BB:  r1      ← (I2C0SR)      ; バスフリーを確認
          AND     r1, #0x20
          CMP     r1, #0x00
          BNE     CHK_BB
          (I2C0DBR) ← 0xCB        ; 送信スレーブアドレス 0x65、方向ビット 1
          (I2C0CR2) ← 0xF8        ; I2C0CR2<MST>,<TRX>,<BB>,<PIN>に 1 を設定
```

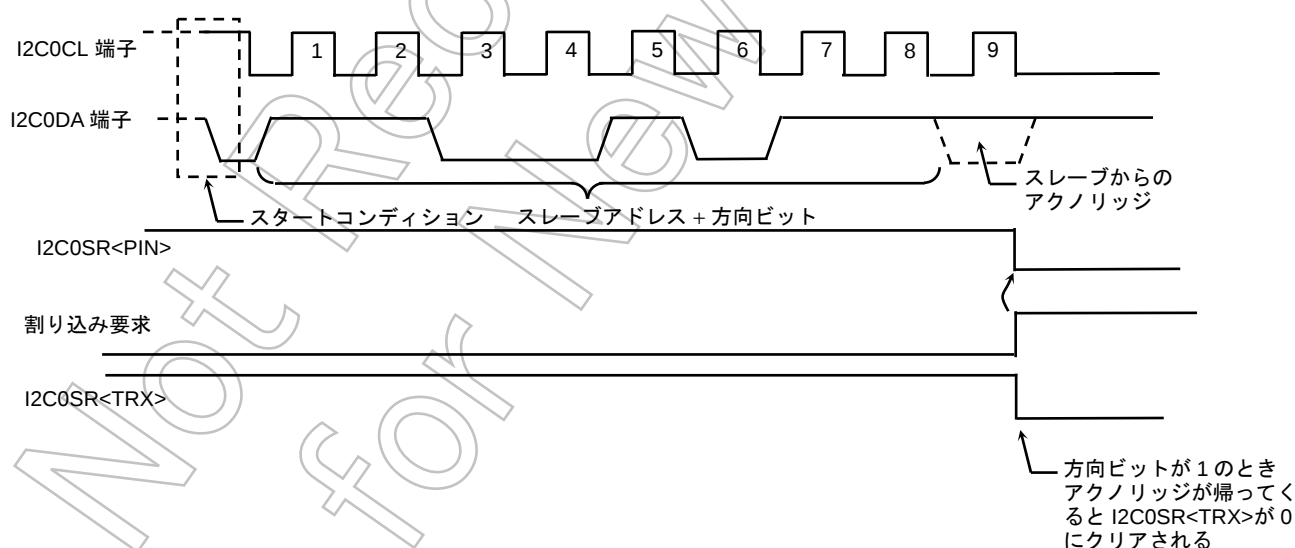


図 3.14.5 スタートコンディションとスレーブアドレスの発生

3. 1 ワードのデータ転送

1 ワード転送終了の割り込みの処理の中で、I2C0SR<MST>の状態を確認し、マスタモード / スレーブモードの判断をします。

(1) I2C0SR<MST>が 1 のとき (マスタモード)

I2C0SR<TRX>の状態を確認し、トランスミッタ/レシーバの判断をします。

a. I2C0SR<TRX>が 1 のとき (トランスミッタモード)

I2C0SR<LRB> フラグにてレシーバからのアクノリッジ状態を確認します。I2C0SR<LRB>が 0 のとき、レシーバが次のデータを要求しているので、送信データを I2C0DBR に書き込みます。

ただし、転送ビット数を変更したい場合には、I2C0CR1<BC>を再設定し、I2C0CR1<ACK>を 1 にセットした後、送信データを I2C0DBR に書き込んでください。

データを書き込むと I2C0SR<PIN>が 1 になり、I2C0CL からシリアルクロックが発生し、I2C0DA からデータが送信されます。

送信終了後、I2CINT0 割り込み要求が発生し、I2C0SR<PIN>が 0 になり、I2C0CL を“L”レベルに引きます。複数ワードの転送が必要な場合は、上記 I2C0SR<LRB>フラグの状態確認から繰り返します。

I2C0SR<LRB>が 1 のとき、レシーバはデータを要求していないので、ストップコンディションが発生する処理 (後述参照) を行ってデータ転送を終了します。

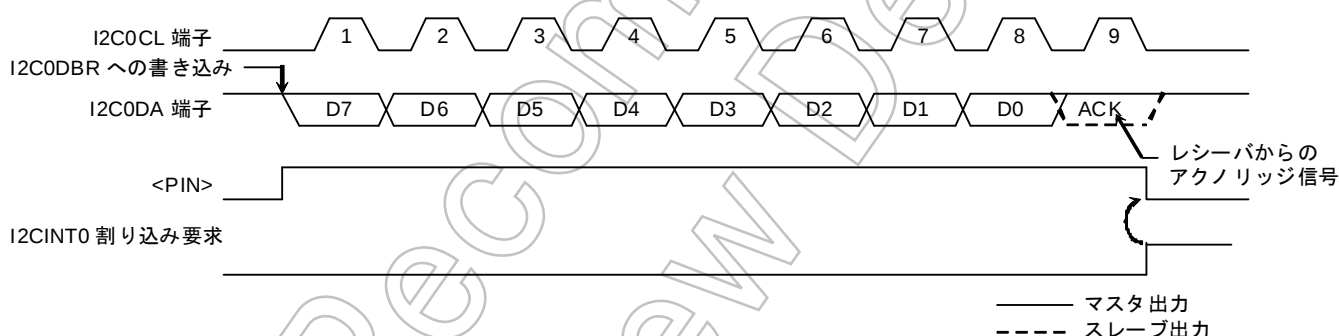


図 3.14.6 I2C0CR1<BC> = 0y000, I2C0CR1<ACK> = 1 の場合

b. I2C0SR<TRX>が 0 のとき (レシーバモード)

I2C0DBR にダミーデータ (0x00) を書き込むか、I2C0CR2<PIN>を 1 にセットすると、1 ワードの転送クロックとアクノリッジを出力します。

受信完了を示す I2CINT0 割り込み要求が発生した後、I2C0DBR から受信データを読み出します。

ただし、受信データビット数を変更したい場合には、I2C0CR1<BC>を再設定し、I2C0CR1<ACK>に 1 をセットした後、ダミーデータ (0x00) を書き込むか、I2C0CR2<PIN>を 1 にセットしてください。

(スレーブアドレス送信直後のリードデータは不定です)。

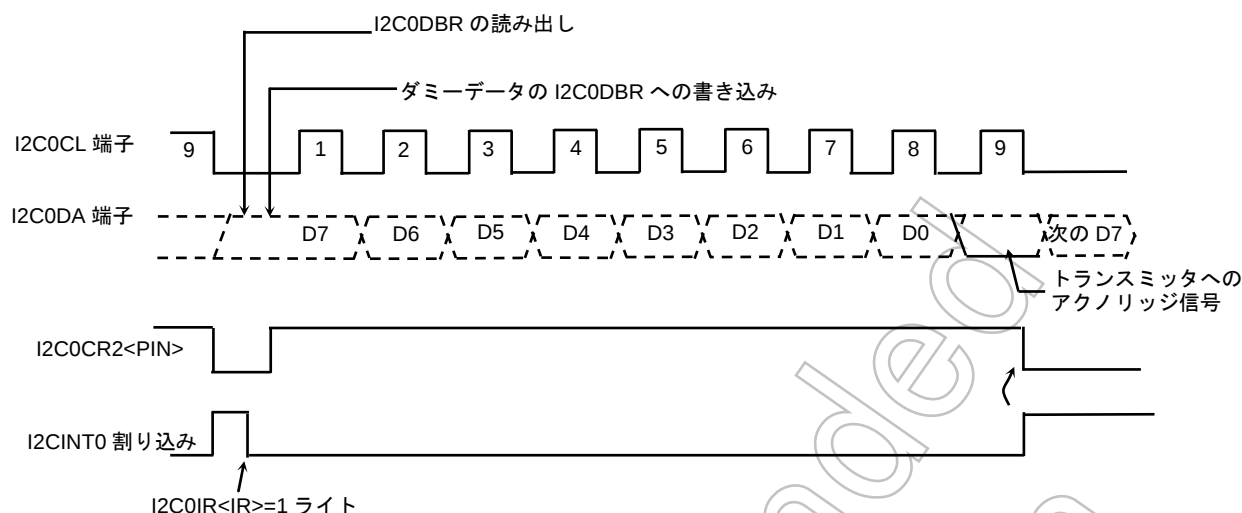


図3.14.7 I2C0CR1<BC> = 0y000, I2C0CR1<ACK> = 1 の場合

c. I2C0SR<TRX>が0のとき(最終ワードを受信する場合)

アクノリッジを出さない擬似通信を行って最終ワードの判断をします。

以下にそのフローを説明します。

トランスミッタに対してデータの送信を終了させるときには、最後のデータを受信する前に下記の処理を行います。

1. 受信データを I2C0DBR から読み出す。
2. I2C0CR1<ACK>を0にクリアし、I2C0CR1<BC>を0y000に設定します。
3. I2C0CR2<PIN>を1にセットするために、I2C0DBR にダミーデータ(0x00)を書き込みます。

I2C0CR2<PIN>が1に設定されると、アクノリッジのためのクロックが発生されない1ワードの転送が実施されます。1ワードの転送の後、下記の処理を行います。

1. 受信データを I2C0DBR から読み出す。
2. I2C0CR1<ACK>を0にクリアし、I2C0CR1<BC>を0y001に設定します。(ネガティブアクノリッジの発行)
3. I2C0DBR にダミーデータ(0x00)か I2C0CR2<PIN>を1にセットします。

I2C0CR2<PIN>が1に設定されると、1ビットの転送が実施されます。このときマスタはレシーバなので、バスの SDA ラインは“H”レベルを保ちます。トランスミッタはこの“H”レベルをネガティブアクノリッジ信号として受信するので、レシーバはトランスミッタへ送信終了を知らせることができます。この1ビット転送の受信終了割り込みの処理で、ストップコンディションを発生させ、データ転送を終了させます。

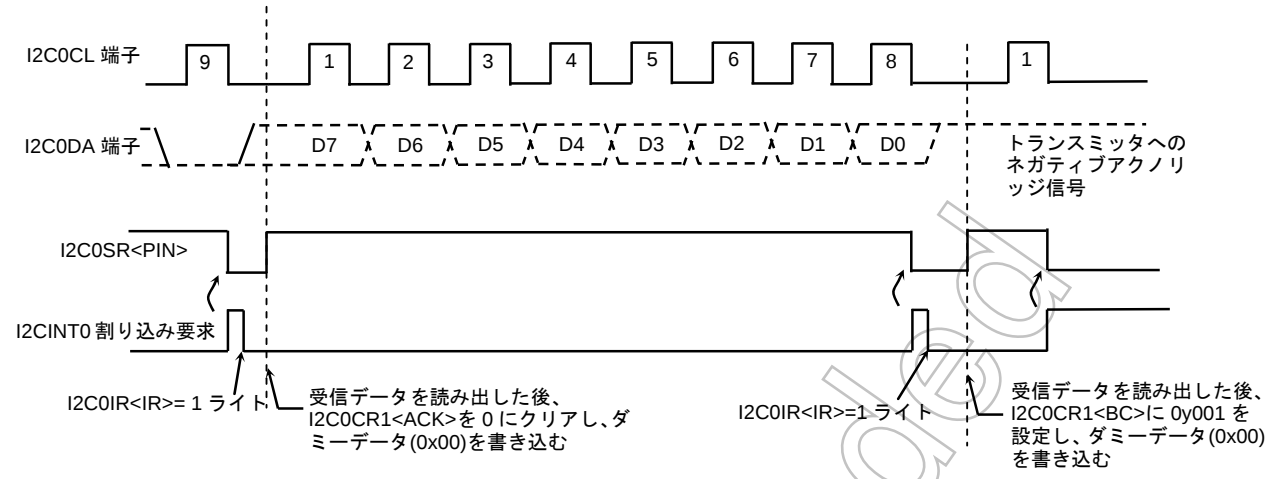


図 3.14.8 マスタレシーバモード時、データの送信を終了させるときの処理

(2) I2C0SR<MST>が 0 のとき(スレーブモード)

スレーブモードのときは、通常のスレーブモードとしての処理または I²C がアービトレーションを失いスレーブモードになったときの処理を行います。

スレーブモードの場合、以下のときに I2CINT0 割り込み要求が発生します。

- I2C0CR1<NOACK>が 0 のとき、受信したスレーブアドレスが、I2C0AR<SA>に設定されたスレーブアドレスと一致したときのアクノリッジ信号出力後
- I2C0CR1<NOACK>が 0 のとき、ゼネラルコールを受信したときのアクノリッジ信号出力後
- スレーブアドレス一致、またはゼネラルコール受信後におけるデータ転送終了時

I²Cがマスタモードのとき、アービトレーションを失うとスレーブモードとして動作し、アービトレーションを失ったワード転送の終了時にI2CINT0 割り込み要求が発生します。アービトレーションを失った後のI2CINT0 割り込み要求発生とI2C0SR<PIN>の動作を表 3.14.1 に示します。

表 3.14.1 アービトレーションロスト時の I2CINT0 割り込み要求と I2C0SR<PIN>の動作

	マスタモード時、スレーブアドレス送信中に アービトレーションを失った場合	マスタトランスミッタモード時、データ送信中に アービトレーションを失った場合
I2CINT0 割り込み要求	ワード転送終了時に I2CINT0 割り込み要求発生	
I2C0SR<PIN>	I2C0SR<PIN>は 0 にクリアされます。	

I2CINT0 割り込み要求が発生すると、I2C0SR<PIN>が 0 にリセットされ、I2C0CL を “L” レベルに引きます。I2C0DBR にデータを書き込むか、I2C0CR2<PIN>を 1 にセットすると I2C0CL が t_{Low} 後に解放されます。

I2C0SR<AL>、I2C0SR<TRX>、I2C0SR<AAS>、I2C0SR<AD0>をテストし、場合分けを行います。表 3.14.2 にスレーブモード時の状態と必要な処理を示します。

表3.14.2 スレーブモード時の処理

I2C0SR <TRX>	I2C0SR <AL>	I2C0SR <AAS>	I2C0SR <AD0>	状態	処理
1	1	1	0	シリアルバスインタフェース回路が、スレーブアドレス送信中にアービトレーションを失い、ほかのマスタが送った方向ビットが 1 のシリアルバスインタフェース回路のスレーブアドレスを受信	1 ワードのビット数を I2C0CR1<BC>にセットし、送信するデータを I2C0DBR に書き込みます。
		1	0	スレーブレシーバモード時、マスタが送った方向ビットが 1 のシリアルバスインタフェース回路のスレーブアドレスを受信	
	0	0	0	スレーブトランスミッタモード時、1 ワードのデータの送信が終了	I2C0SR<LRB>をテストし、1 にセットされていた場合、レシーバが次のデータを要求していないので I2C0CR2<PIN>に 1 をセット、I2C0CR2<TRX>を 0 にリセットしバスを解放します。 I2C0SR<LRB>が 0 にリセットされていた場合、レシーバが次のデータを要求しているため 1 ワードのビット数を I2C0CR1<BC>にセットし、送信するデータを I2C0DBR に書き込みます。
0	1	1	1/0	シリアルバスインタフェース回路が、スレーブアドレス送信中にアービトレーションを失い、ほかのマスタが送った方向ビットが 0 のシリアルバスインタフェース回路のスレーブアドレス、またはゼネラルコールを受信	I2C0SR<PIN>を 1 にセットするために I2C0DBR にダミーデータ (0x00) を書き込みます。または I2C0CR2<PIN>に 1 を書き込みます。
		0	0	シリアルバスインタフェース回路が、スレーブアドレスを送信中またはデータ送信中にアービトレーションを失い、そのワードの転送が終了	シリアルバスインタフェース回路はスレーブモードとなっています。 I2C0SR<AL>を 0 にクリア、I2C0SR<PIN>を 1 にセットするために I2C0DBR にダミーデータ (0x00) を書き込みます。
	0	1	1/0	スレーブレシーバモード時、マスタの送った方向ビットが 0 のシリアルバスインタフェース回路のスレーブアドレス、またはゼネラルコールを受信	I2C0SR<PIN>を 1 にセットするために I2C0DBR にダミーデータ (0x00) を書き込みます。または I2C0CR2<PIN>に 1 を書き込みます。
		0	1/0	スレーブレシーバモード時、1 ワードのデータの受信が終了	1 ワードのビット数を I2C0CR1<BC>にセットし、受信データを I2C0DBR から読み込み、ダミーデータ (0x00) を書き込みます。

注) スレーブモードで I2C0AR<SA>が 0x00 に設定されている場合、I²C バス規格の START バイト (0x01) を受信したときにスレーブアドレスが一致したと判断し、I2C0SR<TRX>が 1 にセットされます。I2C0AR<SA>に 0x00 を設定しないでください。

4. ストップコンディションの発生

I2C0SR<BB>が 1 のときに、I2C0CR2<MST>, I2C0CR2<TRX>, I2C0CR2<PIN>に 1、I2C0CR2<BB>に 0 を書き込むと、バス上にストップコンディションを出力するシーケンスが開始されます。なお、バス上にストップコンディションが発生するまでは、I2C0CR2<MST>, I2C0CR2<TRX>, I2C0CR2<BB>, I2C0CR2<PIN>の内容を書き替えないでください。

また、ストップコンディション発生時に、I2C0CL ラインがほかのデバイスにより“L”レベルに引かれていた場合、I2C0CL ラインが解放された後、ストップコンディションが発生します。

I2C0CL ラインが解放されてからストップコンディションが発生するまで、 t_{HIGH} かかります。

(プログラム例)ストップコンディションの発生

```

I2C0CR2 ← 0xD8 ; I2C0CR2<MST>,<TRX>,<PIN>に 1 を, I2C0CR2<BB>に
                                0 を設定
CHK_BB:  r1 ← (I2C0SR) ; バスフリーを確認
        AND  r1, #0x20
        CMP  r1, #0x00
        BNE  CHK_BB
    
```

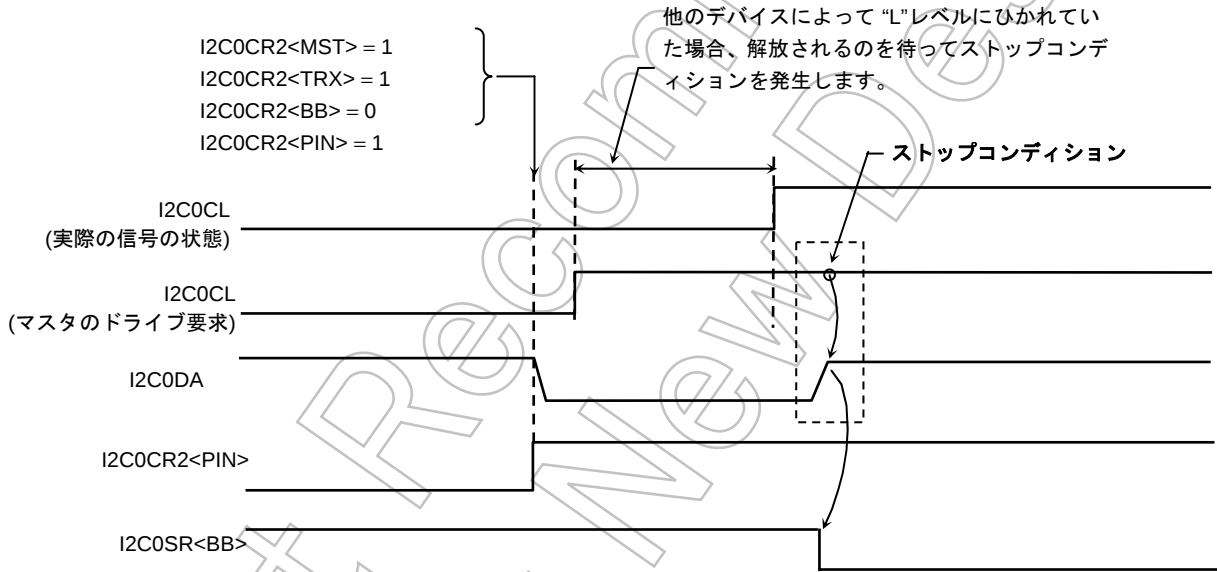


図 3.14.9 ストップコンディションの発生

5. 反復スタートの手順

反復スタートは、マスタデバイスがスレーブデバイスに対してデータ転送を終了させずに、転送の方向を変化させるときに使用します。反復スタートを発生させる場合の手順を以下に示します。

まず、I2C0CR2<MST>, I2C0CR2<TRX>, I2C0CR2<BB>に 0 を、I2C0CR2<PIN>に 1 を書き込むと、I2C0DA は “H” レベルを保ち、I2C0CL が解放されます。

この状態はストップコンディションではないため、ほかのデバイスから見ると、バスはビジー状態のままです。

この後、I2C0SR<BB>の状態を確認して、0 になるまで待ち、I²C の I2C0CL が解放されたことを確認します。

次に I2C0SR<LRB>の状態を確認して 1 になるまで待ち、ほかのデバイスによって、バスの I2C0CL ラインが “L” レベルに引かれていないことを確認します。

以上の手順によってバスが解放状態になっていることを確認した後に、前記「2 スタートコンディション、スレーブアドレスの発生」の手順でスタートコンディションの発生を行います。

なお、反復スタート時のセットアップタイムを満たすために、バスフリーの確認からスタートの発生まで、STANDARD モード I²C バス規格では最低 4.7μs、FAST モード I²C バス規格では最低 0.6μs のソフトウェアによる待ち時間が必要です。

注) マスタデバイスがレシーバの時、反復スタートを発生させる前に、トランスミッタとなっているスレーブデバイスからのデータ送信を終了させる必要があります。データ送信を終了させるために、“H”レベルのネガティブアックノリッジ信号をスレーブデバイスに受信させます。このため、反復スタート発生前の I2C0SR<LRB>は 1 となり、反復スタートの手順で、I2C0SR<LRB>が 1 であることを確認しても、SCL ラインの立ち上がりを確認できません。I2C0CL ラインの状態を確認するには、ポートを読み出してください。

(プログラム例)反復スタートの発生

```

(I2C0CR2) ← 0x18 ; I2C0CR2<MST>,<TRX>,<BB>に 0 を、I2C0CR2<PIN>に
                                1 を設定
CHK_BB:  r1 ← (I2C0SR) ; I2C0SR<BB>が 0 になるのを待つ
        AND r1, #0x20
        CMP r1, #0x00
        BNE CHK_BB
CHK_LRB: r1 ← (I2C0SR) ; I2C0SR<LRB>が 1 になるのを待つ
        AND r1, #0x01
        CMP r1, #0x01
        BNE CHK_LRB
        .
        . ; ソフトウェアによる時間待ち処理
(I2C0CR2) ← 0xF8 ; I2C0CR2<MST>,<TRX>,<BB>,<PIN>に 1 を設定

```

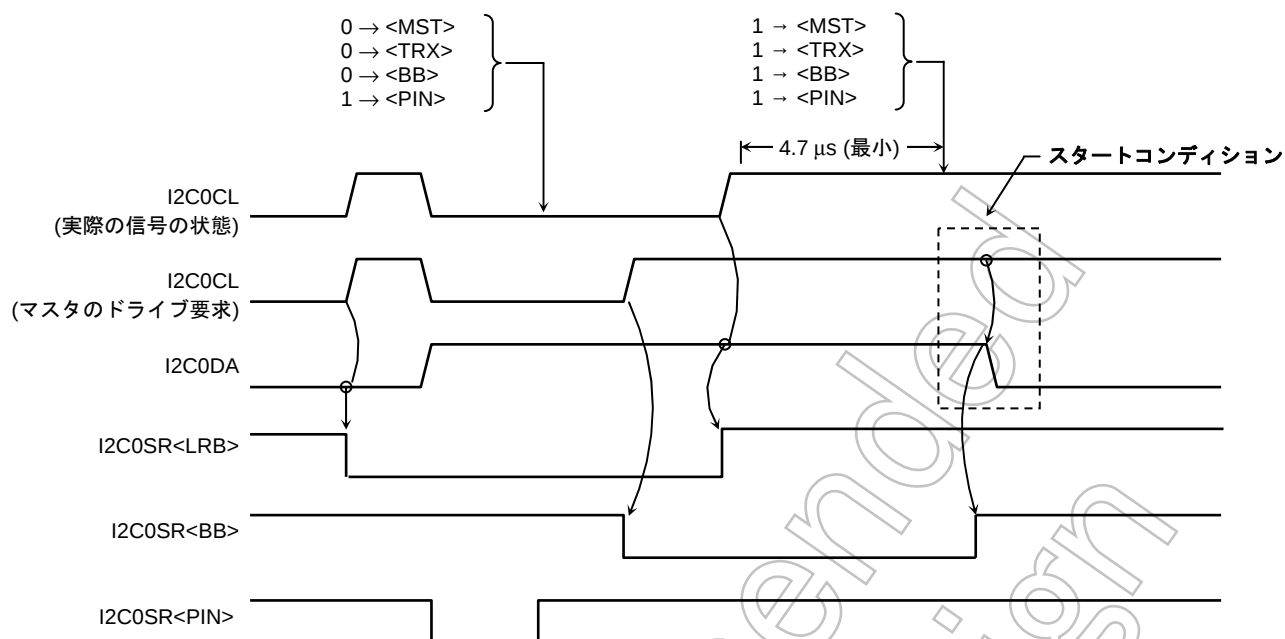


図 3.14.10 反復スタートを発生させる場合のタイミングチャート

注) <MST> = 0 の状態の時に<MST> = 0 をライトしないでください。(再スタートできません)

3.14.5 レジスタの説明

SFR のリストを以下に示します。

base アドレス= 0xF007_0000

Register Name	Address (base+)	Description
I2C0CR1	0x0000	I ² C0 Control Register 1
I2C0DBR	0x0004	I ² C0 Data Buffer Register
I2C0AR	0x0008	I ² C0 (Slave) Address Register
I2C0CR2	0x000C	I ² C0 Control Register 2
I2C0SR		I ² C0 Status Register
I2C0PRS	0x0010	I ² C0 Prescaler Clock Set Register
I2C0IE	0x0014	I ² C0 Interrupt Enable Register
I2C0IR	0x0018	I ² C0 Interrupt Register

1. I2C0CR1 (I²C0 Control Register 1)

Address = (0xF007_0000) + (0x0000)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:5]	BC[2:0]	R/W	0y000	転送ビット数の選択 0y000: 8 ビット 0y100: 4 ビット 0y001: 1 ビット 0y101: 5 ビット 0y010: 2 ビット 0y110: 6 ビット 0y011: 3 ビット 0y111: 7 ビット
[4]	ACK	R/W	0y0	アクノリッジのためのクロック発生／認識選択 0y0: なし 0y1: あり
[3]	NOACK	R/W	0y0	スレーブアドレス一致検出およびゼネラルコール検出 0y0: する 0y1: しない
[2:0]	SCK[2:0]	R/W	0y000	シリアルクロック周波数の選択 0y000: n = 0 0y100: n = 4 0y001: n = 1 0y101: n = 5 0y010: n = 2 0y110: n = 6 0y011: n = 3 0y111: n = 7

(個別説明)

a. <BC[2:0]>

転送ビット数を設定します。

0y000: 8 ビット 0y100: 4 ビット

0y001: 1 ビット 0y101: 5 ビット

0y010: 2 ビット 0y110: 6 ビット

0y011: 3 ビット 0y111: 7 ビット

b. <ACK>

アクノリッジのためのクロック発生／認識選択をします。

0y0: なし

0y1: あり

c. <NOACK>

スレーブ動作時、スレーブアドレス一致およびゼネラルコールを検出する／しないを設定します。

0y0: する

0y1: しない

I2C0AR<ALS>が1のときは、このビットは意味を持ちません。

<NOACK>=0の時、スレーブアドレスの一致検出およびゼネラルコールの検出をおこない、スレーブアドレスの一致検出時またはゼネラルコールの検出時、マスタの出力する9クロック目(アクノリッジクロック)の間 SDA ラインを“L”に引き、アクノリッジを返します。

<NOACK>=1の時、スレーブアドレスの一致検出およびゼネラルコールの検出をおこなわず、スレーブアドレスの一致時、またはゼネラルコール時、マスタの出力するアクノリッジクロック(9クロック目)の間 SDA ラインを解放(High 状態)し、アクノリッジを返しません。

d. <SCK[2:0]>

マスタ時に出力するシリアルクロックの速度を設定します。

シリアルクロック生成の基準クロックには、I2C0PRS<PRSC[4:0]>に従って分周されたプリスケールクロックを使用します。プリスケールクロックは I2C0CR1<SCK[2:0]>に従ってさらに分周し、シリアルクロックとして使用します。プリスケールクロックのデフォルト設定は1分周(= fPCLK)です。

注) I2C0CR1<SCK[2:0]>による分周設定は6.I2C0PRS (I2C0 Prescaler Clock Set Register)、3.14.6.3シリアルクロックを参照してください。

本レジスタに対する書き込みは、スタートコンディション発生前およびストップコンディション発生後、もしくはアドレスまたはデータ転送後の割込み発生から、内部割込み解除までの間に行ってください。アドレス、データ転送中の書き込みはしないでください。

2. I2C0DBR (I²C0 Data Buffer Register)

Address = (0xF007_0000) + 0x0004

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined.
[7:0]	DB[7:0]	RO	0x00	RD 時: 受信データ読み出し (注)

Address = (0xF007_0000) + (0x0004)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:0]	DB[7:0]	WO	0x00	WR 時: 送信データ書き込み (注)

注) 初期化されるのはハードウェアリセット後のみです。ソフトウェアリセット後は初期化されません。(最終データを保持します)

(個別説明)

a. <DB[7:0]>

シリアル転送のデータを格納します。

送信デバイスとして機能している場合、送信したいデータを DB[7:0] に「左詰めで」書き込みます。

受信デバイスとして機能している場合、シリアル転送によって受信したデータが DB[7:0] に「右詰めで」格納されます。

マスタからのアドレス送信時は、転送のターゲットとなるデバイスのアドレスを I2C0DBR<DB[7:1]>に書き込み、I2C0DBR<DB[0]>には、転送の方向ビットとして

0y0: マスタ送信 → スレーブ受信

0y1: マスタ受信 ← スレーブ送信

を書き込みます。また、I2C0DBR レジスタの全てのビットに 0 を書き込むと、バスにゼネラルコールを送信することができます。

送信時／受信時とも、I2C0DBR に対しての書き込み動作によって、転送後の内部割込みが解除され、次の転送が開始されます。

I2C0DBR は書き込み用のバッファと読み出し用のバッファを兼用していますが、送信時は送信専用、受信時は受信専用として使用してください。また、1 回の転送ごとにレジスタをアクセスするようにしてください。

注) 受信設定時、受信データ読み出し前に I2C0DBR にデータを書き込むと受信データが破壊されてしまいますので注意してください。

3. I2C0AR (I²C0 (Slave) Address Register)

Address = (0xF007_0000) + (0x0008)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:1]	SA[6:0]	R/W	0y0000000	スレーブアドレスの設定
[0]	ALS	R/W	0y0	アドレス認識モードの設定 0y0: する(I ² C バスモード) 0y1: しない(フリーデータフォーマット)

(個別説明)

a. <SA[6:0]>

スレーブデバイスとして動作する場合のデバイスのアドレス(7bit)を設定します。

後述の I2C0AR<ALS>によってスレーブアドレスを認識するよう設定されていた場合、マスタがスタートコンディション直後に送信する 7bit のアドレス(+1bit の方向ビット)によって、データ転送時の動作を決定します。

b. <ALS>

アドレス認識モードを設定します。

0y0: する(I²C バスモード)

0y1: しない(フリーデータフォーマット)

スレーブデバイスとして動作する場合、マスタがスタートコンディション直後に送信する 8bit のデータを、7bit のアドレス+1bit 方向ビットとして認識する／しないを設定します。

認識する場合は I²C バスモード動作、認識しない場合はフリーフォーマット動作を行います。

<ALS>= 0 の時、デバイスはマスタの送信する 7bit のアドレスと I2C0AR<SA[6:0]>を比較し、一致していた場合、方向ビットによって送信／受信を決定し、I2C0CR1<NOACK>=0 であれば、マスタの出力するアクノリッジタロック(9 クロック目)の間 SDA ラインを”L”に引きます。

以降、ストップコンディション、または再スタート手順によるスタートコンディションがバス上に現れるまで、スレーブ・送信／受信デバイスとして転送動作を行います。

一致しなかった場合は、SDA ラインを解放(High 状態)したまま、以降、ストップコンディション、または再スタート手順によるスタートコンディションがバス上に現れるまで SDA／SCL ラインを解放(High 状態)し続け、転送に参加しません(I²C バスモード動作、スレーブアドレス一致検出)。

特に、マスタの送信する 7bit のアドレス+1bit の方向ビットが全て 0(ゼネラルコール)であった場合、I2C0CR1<NOACK>=0 であれば、デバイスは I2C0AR<SA[6:0]>に設定されたスレーブアドレスとは関係無くアクノリッジ(Low 状態)を出力し、スレーブ・受信デバイスとして動作します(I²C バスモード動作、ゼネラルコール検出)。

I2C0CR1<NOACK>=1 である場合は、スレーブアドレス一致検出もしくはゼネラルコール検出が起ころても、アクノリッジを出力せず、スレーブデバイスとして動作しません。

<ALS>= 1 の時、マスタデバイスの送信する 7bit のアドレス+1bit の方向ビットをデータとして受信し、マスタの出力するアクノリッジクロック(9 クロック目)の間 SDA ラインを”L”に引きます。

以降、ストップコンディション、または再スタート手順によるスタートコンディションがバス上に現れるまで、スレーブ受信デバイスとして転送動作を行います(フリーフォーマット動作)。この場合、I2C0CR1<NOACK>の値には影響されません。

本レジスタに対する書き込みは、スタートコンディション発生前もしくはストップコンディション発生後に行ってください。転送中の書き込みはできません。

Not Recommended
for New Design

4. I2C0CR2 (I²C0 Control Register 2) (Write Only)

Address = (0xF007_0000) + (0x000C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7]	MST	WO	0y0	マスタ/スレーブ選択 0y0: スレーブ 0y1: マスタ
[6]	TRX	WO	0y0	送信/受信選択 0y0: レシーバ 0y1: トランスミッタ
[5]	BB	WO	0y0	スタート/ストップコンディションの発生 0y0: ストップコンディション発生 0y1: スタートコンディション発生
[4]	PIN	WO	0y1	処理要求解除 0y0: 無効 0y1: 処理要求解除
[3]	I2CM	WO	0y0	I ² C 動作制御 0y0: ディセーブル 0y1: イネーブル
[2]	–	–	Undefined	Read as undefined. Write as zero.
[1:0]	SWRES[1:0]	WO	0y00	ソフトウェアリセット発生 0y10–0y01 の連続ライトで発生

(個別説明)

a. <MST>

マスタ/スレーブの選択をします。

0y0: スレーブ

0y1: マスタ

注) 3.14.6.3シリアルクロックを参照してください。

b. <TRX>

送信/受信の選択をします。

0y0: レシーバ

0y1: トランスミッタ

注) 3.14.6.3シリアルクロックを参照してください。

c. <BB>

スタート/ストップコンディションを発生します。

0y0: ストップコンディション発生

0y1: スタートコンディション発生

注) 3.14.6.3シリアルクロックを参照してください。

d. <PIN>

I²C 通信の、処理要求を解除します。

0y0: 無効

0y1: 処理要求解除

注) 3.14.6.3シリアルクロックを参照してください。

e. <I2CM>

I²C 動作の制御をします。

0y0: ディセーブル

0y1: イネーブル

通信中はディセーブル設定をすることはできません。ステータスレジスタをリードし、転送が完全に停止していることを確認してからディセーブル設定をしてください。

f. <SWRES[1:0]>

この 2 ビットを 0y10→0y01 の順にライトすると、ソフトウェアリセットが発生します。(リセット幅 = f_{CLK} 1 クロック分)

ソフトウェアリセットが発生すると転送中であっても強制的に SCL、SDA ラインを解放 (High 状態) し、転送動作を中断します。また、I2C0CR2<I2CM>以外の全ての設定が初期化されます。(I2C0DBR は初期化されません。)

ソフトウェアリセット書き込みの際は、必ず I2C0CR2[7:4]に 0 をライトして下さい。

5. I2C0SR (I²C0 Status Register) (Read Only)

Address = (0xF007_0000) + (0x000C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7]	MST	RO	0y0	マスタ/スレーブ選択状態モニタ 0y0: スレーブ 0y1: マスタ
[6]	TRX	RO	0y0	送信/受信選択状態モニタ 0y0: レシーバ 0y1: トランスミッタ
[5]	BB	RO	0y0	バス状態モニタ 0y0: バスフリー 0y1: バスビジー
[4]	PIN	RO	0y1	処理要求状態および SCL ライン状態モニタ 0y0: 処理要求中、SCL ライン"1"状態 0y1: 処理要求無し、SCL ラインフリー
[3]	AL	RO	0y0	アービトレーションロスト検出モニタ 0y0: 無効 0y1: 検出
[2]	AAS	RO	0y0	スレーブアドレス一致検出モニタ 0y0: 無効 0y1: 検出
[1]	AD0	RO	0y0	ゼネラルコール検出モニタ 0y0: 無効 0y1: 検出
[0]	LRB	RO	0y0	最終受信ビットモニタ 0y0: 最終受信ビット 0 0y1: 最終受信ビット 1

(個別説明)

a. <MST>

マスタ/スレーブの選択状態をモニタします。

0y0: スレーブ

0y1: マスタ

b. <TRX>

送信/受信の選択状態をモニタします。

0y0: レシーバ

0y1: トランスミッタ

c. <BB>

バスの状態をモニタします。

0y0: バスフリー

0y1: バスビジー

バス上のスタートコンディションの検出後 1 にセットされ、ストップコンディションの検出によって 0 にクリアされます。スレーブデバイス時、マスタの送信するアドレスに一致せず転送動作に関与しない場合においても、このビットは 1 にセットされ、デバイスはストップコンディションの発生を監視します。

また、このビットが 1 にセットされている間はスタートコンディションを発生することができません。

d. <PIN>

処理要求と SCL の状態をモニタします。

0y0: 処理要求中、SCL ライン “L”OUT 中

0y1: 処理要求無し、SCL ラインフリー

e. <AL>

アービトラクションロストの検出をモニタします。

0y0: 無効

0y1: 検出

f. <AAS>

スレーブアドレス一致の検出をモニタします。

0y0: 無効

0y1: 検出

スレーブ動作時、マスタの送信したアドレスと I2C0AR<SA[6:0]>が一致した場合、1 にセットされます。その後の内部割込み解除以降 0 にクリアされ、ストップコンディション、または再スタート手順によるスタートコンディションがバス上に現れ、再度スタートコンディション後のアドレス転送によって一致検出→1 にセットされるまで、0 のままになります。

g. <AD0>

ゼネラルコールの検出をモニタします。

0y0: 無効

0y1: 検出

ゼネラルコール(スタートコンディション後のアドレス転送期間中、SDA ラインが常に”L”)を受信後、ストップコンディション、または再スタート手順によるスタートコンディションがバス上に現れるまで、1 にセットされ続けます。ゼネラルコール受信後、I2C0SR<AAS>も 1 にセットされますが、こちらは前述のように、次のデータ転送時に 0 にクリアされます。

h. <LRB>

最終受信ビットをモニタします。

0y0: 最終受信ビット 0

0y1: 最終受信ビット 1

アクノリッジ有りに設定している場合、転送後の割込み時に読み出すことで、受信デバイスがアクノリッジを出力(Low 状態)したかどうかを確認することができます。送信、受信デバイスに関わらず、このモニタは有効です。

注) 3.14.6.15ソフトウェアリセット後のレジスタ値を参照してください。

Not Recommended
for New Design

6. I2C0PRS (I²C0 Prescaler Clock Set Register)

Address = (0xF007_0000) + (0x0010)

Bit	Bit Symbol	Type	Reset Value	Description
[31:5]	–	–	Undefined	Read as undefined. Write as zero.
[4:0]	PRSCK[4:0]	R/W	0y00001	シリアルクロック生成用プリスケールクロック周波数の選択 0y00000: p = 32 分周 0y00001: p = 1 分周 ⋮ 0y11111: p = 31 分周

(個別説明)

a. <PRSCK[4:0]>

シリアルクロック生成用プリスケールクロック周波数の選択をします。

0y00000: p = 32 分周

0y00001: p = 1 分周

⋮

0y11111: p = 31 分周

注)1.I2C0CR1 (I2C0 Control Register 1)、3.14.6.3シリアルクロックを参照してください。

7. I2C0IE (I²C0 Interrupt Enable Register)

Address = (0xF007_0000) + (0x0014)

Bit	Bit Symbol	Type	Reset Value	Description
[31:1]	–	–	Undefined	Read as undefined. Write as zero.
[0]	IE	R/W	0y0	I ² C 割り込み 0y0: 禁止 0y1: 許可

(個別説明)

a. <IE>

I²C 割り込み出力の許可／禁止を設定します。

0y0: 禁止

0y1: 許可

8. I2C0IR (I²C0 Interrupt Register)

Address = (0xF007_0000) + (0x0018)

Bit	Bit Symbol	Type	Reset Value	Description
[31:1]	–	–	Undefined	Read as undefined. Write as zero.
[0]	IS/IC	R/W	0y0	(RD 時) I ² C(禁止前)割り込みステータス 0y0: 割り込みなし 0y1: 割り込み発生 (WR 時) I ² C 割り込みをクリア 0y0: 無効 0y1: クリア

(個別説明)

a. <IS/IC>

(RD 時)

I2C0IE<IE>によるマスク前の I²C 割り込みステータスです。

0y0: 割り込みなし

0y1: 割り込み発生

(WR 時)

I²C 割り込みをクリアします。

0y0: 無効

0y1: クリア

1 を書き込むと、I²C 割り込み出力(I2CINT0)をクリアします。

0 を書き込んでも何も起こりません。

3.14.6 機能

3.14.6.1 スレーブアドレス一致検出、ゼネラルコール検出の選択

スレーブデバイスがスレーブアドレスの一致検出、ゼネラルコール検出をする際に、以下の設定をします。

I2C0CR1<NOACK>で、スレーブモードのスレーブアドレス一致検出、ゼネラルコール検出の許可/不許可の設定を行います。

I2C0CR1<NOACK>を 0 にクリアすると、スレーブアドレス一致検出、ゼネラルコール検出を許可します。

I2C0CR1<NOACK>を 1 にセットすると、セット後のスレーブアドレス一致検出、ゼネラルコール検出を禁止します。このとき、マスタから送られてくるスレーブアドレス、ゼネラルコールを無視し、アクノリッジを返さず、I2CINT0 割り込み要求も発生しません。

マスタモード時、I2C0CR1<NOACK>は無視され、動作に影響を与えません。

注) スレーブモードでデータを転送中に I2C0CR1<NOACK>を 0 にクリアしても 1 の状態が保持され、データ転送時のアクノリッジを返します。

3.14.6.2 データ転送のクロック数とアクノリッジ有無の選択

(1) データ転送のクロック数

データ転送のクロック数は I2C0CR1<BC>と I2C0CR1<ACK>で設定されます。

I2C0CR1<ACK>が 1 にセットされると、アクノリッジメントモードで動作します。

アクノリッジメントモードの時、マスタデバイスではデータビット数分のクロックを発生した後、アクノリッジのためのクロックを発生し I2CINT0 割り込み要求を発生します。

スレーブデバイスではデータビット分のクロックをカウントした後、アクノリッジのためのクロックカウントをし I2CINT0 割り込み要求を発生します。

I2C0CR1<ACK>が 0 にクリアされると、非アクノリッジメントモードで動作します。

非アクノリッジメントモードの時、マスタデバイスではデータビット数分のクロックを発生した後、I2CINT0 割り込み要求を発生します。

スレーブデバイスではデータビット分のクロックをカウントした後、I2CINT0 割り込み要求を発生します。

受信デバイス時にアクノリッジ有りに設定している場合、マスタの出力するアクノリッジクロックの期間 I2C0DA を“L”にドライブし、送信デバイスに次のワードの転送を要求します。逆に、アクノリッジ無しに設定することによって、マスタの出力するアクノリッジクロックの期間も I2C0DA を解放(High 状態)し、送信デバイスに次のワード転送を要求していないことを伝えることができます。

アドレス送信時(スタートコンディション発生前)は必ずマスタ、スレーブ共に、転送ビット数 8bit/アクノリッジ有りに設定してください。

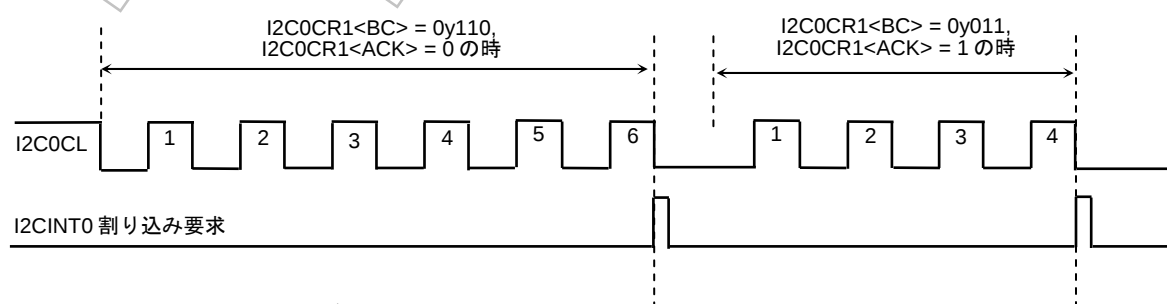


図 3.14.11 データ転送クロック数と I2C0CR1<BC>, I2C0CR1<ACK>

データ転送のクロック数とI2C0CR1<BC>と I2C0CR1<ACK>の関係は 表 3.14.3の通りとなります。

表 3.14.3 データ転送のクロック数

BC[2:0]	アクノリジメント動作 (I2C0CR1<ACK>)			
	0: 無し		1: 有り	
	データ長	クロック数	データ長	クロック数
000	8	8	8	9
001	1	1	1	2
010	2	2	2	3
011	3	3	3	4
100	4	4	4	5
101	5	5	5	6
110	6	6	6	7
111	7	7	7	8

<BC> はスタートコンディションにより 0y000 にクリアされます。

そのため、スレーブアドレス、方向ビットの転送は必ず 8 ビットで行われます。それ以外のときは、<BC> は一度設定された値を保持します。

注) スレーブアドレスの送信、受信は I2C0CR1<ACK>をセットした状態で実施してください。I2C0CR1<ACK>がクリアされた状態ではスレーブアドレスの一致、方向ビットの検出が正常に行われません。

(2) アクノリッジ出力

アクノリッジメントモードの時、アクノリッジのためのクロック期間中、I2C0DA が下記のように変化します。

- マスタモード時

トランスミッタモードのとき、アクノリッジのためのクロック期間中、レシーバからのアクノリッジ信号を受信するために I2C0DA を解放します。

レシーバモードのときは、アクノリッジのためのクロック期間中、I2C0DA を“L”レベルに引きアクノリッジ信号を発生します。

- スレーブモード時

受信したスレーブアドレスと I2C0AR<SA>に設定されたスレーブアドレスが一致したとき、またはゼネラルコールを受信したときに、アクノリッジのためのクロック期間中、I2C0DA を“L”レベルに引きアクノリッジ信号を発生します。

スレーブアドレス一致またはゼネラルコール受信後のデータ転送において、トランスミッタモードのときには、アクノリッジのためのクロック期間中、I2C0DA を解放し、レシーバからのアクノリッジ信号を受信できる状態にします。

レシーバモードのときには、I2C0DA を“L”レベルに引きアクノリッジ信号を発生します。

表 3.14.4にアクノリッジメントモードにおけるI2C0CLとI2C0DAの状態を示します。

注) 非アクノリッジメントモードの時、アクノリッジのためのクロック発生、カウントを行わないため、アクノリッジ出力はされません。

表 3.14.4 アクノリッジメントモードにおける I2C0CL と I2C0DA の状態

モード	端子	条件	トランスミッタ	レシーバ
マスタ	I2C0CL	—	アクノリッジのためのクロックを付加	アクノリッジのためのクロックを付加
	I2C0DA	—	アクノリッジ信号受信のために端子を解放	アクノリッジ信号として端子に“L”を出力
スレーブ	I2C0CL	—	アクノリッジのためのクロックをカウント	アクノリッジのためのクロックをカウント
	I2C0DA	スレーブアドレスが一致したとき、またはゼネラルコールを受信したとき	—	アクノリッジ信号として端子に“L”を出力
		スレーブアドレスが一致したとき、またはゼネラルコール受信後の、転送時	アクノリッジ信号受信のために端子を解放	アクノリッジ信号として端子に“L”を出力

3.14.6.3 シリアルクロック

(1) クロックソース

I2C0CR1<SCK>で、マスタモード時に出力されるシリアルクロックの HIGH 時間、LOW 時間を設定します。

SCK	$t_{\text{HIGH}} (i / T_{\text{prsc}})$	$t_{\text{LOW}} (j / T_{\text{prsc}})$
	i	j
0y000	8	12
0y001	10	14
0y010	14	18
0y011	22	26
0y100	38	42
0y101	70	74
0y110	134	138
0y111	262	266

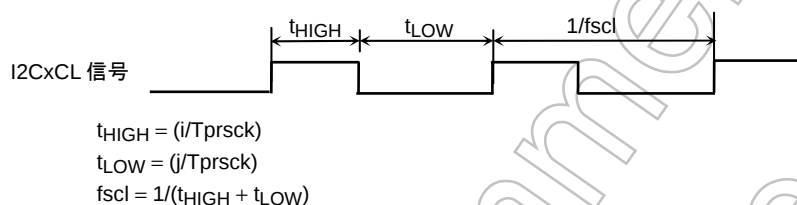


図 3.14.12 I2CxCL 出力

注) バスの負荷容量と Pull-Up 抵抗との組み合わせにより、立ち上がりが鈍り、設定した t_{HIGH} とならないことがあります。また、他のデバイスの出力するシリアルクロックと同期をとる機能（クロック同期化機能）が働いた場合、設定値と異なるクロックとなる場合があります。

マスタ時、スタートコンディション発生時のホールドタイムと、ストップコンディション発生時のセットアップタイムは $t_{\text{HIGH}}[s]$ となります。

スレーブ時の I2C0CR2<PIN>を 1 にセットしたとき、I2C0CL 解放までの時間は $t_{\text{LOW}}[s]$ となります。

なお、マスタモード/ スレーブモードともに I2C0CR1<SCK>と関係なく、外部から入力されるシリアルクロックは、“H” レベルは $4/T_{\text{prsc}}[s]$ 以上、“L” レベルは $5/T_{\text{prsc}}[s]$ 以上必要です。

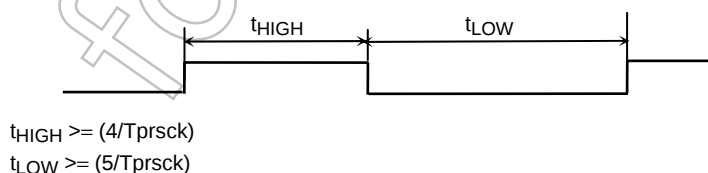


図 3.14.13 SCLK 入力

I2C0CR1<SCK[2:0]>, I2C0PRS<PRSCCK[4:0]>により、マスタ時に出力するシリアルクロックの速度を設定します。シリアルクロック生成の基準クロックには、I2C0PRS<PRSCCK[4:0]>に従って分周されたプリスケラクロックを使用します。

プリスケラクロックは I2C0CR1<SCK[2:0]>に従ってさらに分周し、シリアルクロックとして使用します。プリスケラクロックのデフォルト設定は 1 分周(=PCLK)です。

<シリアル転送レートについて>

シリアルクロックレート(Fscl)は、動作周波数(PCLK)をもとに、プリスケラ設定“p”(I2C0PRS<PRSCCK[4:0]>, 1~32) およびシリアルクロック設定“n”(I2C0CR1<SCK[2:0]>, 0~7) の組み合わせにより、以下のように決定されます。

$$\text{シリアルクロックレート Fscl(kHz)} = \frac{\text{PCLK (MHz)}}{p \times (2^{n+2} + 16)} \times 1000$$

* 注意*

設定範囲は動作周波数(PCLK)により変わります。以下の条件を満たすように、プリスケラ設定“p”(I2C0PRS<PRSCCK[4:0]>)の設定可能範囲を決定してください。

$$50\text{ns} < \text{プリスケラクロック幅 Tprsck (ns)} \leq 150\text{ns}$$

注)マスタ/スレーブに関わらず、この範囲外はプリスケラ設定禁止です。

また、他のデバイスの出力するシリアルクロックと同期をとる機能があるため、シリアルクロックの速度は一定で無くなることもあります。

SCK[2:0]=(n)			PRSCCK[4:0] = (p)		
			0y00001(1 分周)	0y01101(13 分周)	0y00000(32 分周)
			(PCLK との比)		
0	0	0	20	260	640
0	0	1	24	312	768
0	1	0	32	416	1024
0	1	1	48	624	1536
1	0	0	80	1040	2560
1	0	1	144	1872	4608
1	1	0	272	3536	8704
1	1	1	528	6864	16896

このビット(群)に対する書き込みは、ストップコンディション発生前もしくはストップコンディション発生後に行ってください。転送中の書き込みは予期せぬ動作の原因になります。

<プリスケラクロック幅(=ノイズキャンセル幅)について>

プリスケラクロック幅(Tprsck) (=ノイズキャンセル幅)は、動作周波数(PCLK)をもとに、プリスケラ設定“p”(I2C0PRS<PRSCCK[4:0]>, 1~32)により、以下のように決定されます。

$$\begin{aligned} \text{プリスケラクロック幅 Tprsck (ns)} &= \frac{1}{\text{PCLK (MHz)}} \times 1000 \times p \\ &(\text{=ノイズキャンセル幅}) \end{aligned}$$

(2) クロック同期化

I²C バスでは端子の構造上、バスをワイヤードアンドで駆動させるために、クロックラインを最初に“L”レベルに引いたマスタが、“H”レベルを出力しているマスタのクロックを無効にします。このため、“H”レベルを出力しているマスタは、これを検出し対応する必要があります。

I²C はクロック同期化機能をもっており、バス上に複数のマスタが存在する場合でも、正常に転送が行われます。

クロック同期の手順を、バス上に 2 つのマスタが同時に存在した場合を例にあげて以下に示します。

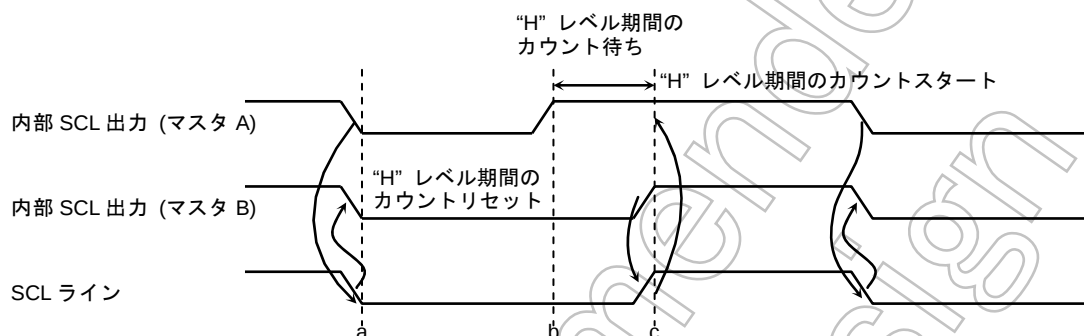


図 3.14.14 クロック同期化の例

a 点でマスタ A が I²C0CL を“L”レベルに引くことで、バスの SCL ラインは“L”レベルになります。マスタ B はこれを検出し、マスタ B の“H”レベル期間のカウントをリセットし、I²C0CL を“L”レベルに引きます。

b 点でマスタ A は“L”レベル期間のカウントを終わり、I²C0CL を“H”レベルにします。しかしマスタ B がバスの SCL ラインを“L”レベルに保持し続けているので、マスタ A は“H”レベル期間のカウントを始めません。マスタ A は、c 点でマスタ B が I²C0CL を“H”レベルにし、バスの SCL ラインが“H”レベルになったことを検出後、“H”レベル期間のカウントを始めます。

その後“H”レベル期間のカウントを終了したマスタ A が I²C0CL を“L”に引くことで、バスの SCL ラインは“L”レベルになります。

以上のようにバス上のクロックは、バスに接続されているマスタの中で最も短い“H”レベル期間をもつマスタと、最も長い“L”レベル期間をもつマスタによって決定されます。

3.14.6.4 マスタ/スレーブの選択

I²C0CR2<MST>を 1 にセットすると、I²C はマスタデバイスとして動作します。

I²C0CR2<MST>を 0 にクリアすると、スレーブデバイスとして動作します。I²C0SR<MST>はバス上のストップコンディションを検出したとき、またはアービトレーションロストを検出したとき、ハードウェアにより 0 にクリアされます。

3.14.6.5 トランスミッタ/レシーバの選択

I2C0CR2<TRX>を1にセットすると、I²Cはトランスミッタとして動作し、I2C0CR2<TRX>を0にクリアするとレシーバとして動作します。

I²C バスモードのデータ転送を行うとき、スレーブモード時は、ハードウェアにより、マスタデバイスから送られてくる方向ビット(R/W)が1の場合、I2C0SR<TRX>は1にセットされ、0の場合、I2C0SR<TRX>は0にクリアされます。

マスタモード時は、スレーブデバイスからアクノリッジが返ってくると、ハードウェアにより送信した方向ビットが1の場合、I2C0SR<TRX>は0にクリアされ、0の場合、I2C0SR<TRX>は1に変化します。アクノリッジが返ってこないときは、以前の状態を保ちます。

I2C0SR<TRX>はバス上のストップコンディションを検出したとき、またはアービトレーションロストが検出されると、ハードウェアにより0にクリアされます。表 3.14.5に各モードでのI2C0SR<TRX>の変化条件と変化後のI2C0SR<TRX>の値を示します。

注) I2C0CR1<NOACK>が1のとき、スレーブアドレス一致の検出、ゼネラルコールの検出が禁止されているため、I2C0SR<TRX>は変化しません。

表 3.14.5 各モードでの I2C0SR<TRX>の動作

モード	方向ビット	変化条件	変化後の TRX
スレーブモード	0	受信したスレーブアドレスが I2C0AR<SA>に設定された値と同じとき	0
	1		1
マスタモード	0	ACK 信号が返ってきたとき	1
	1		0

I²C を、フリーデータフォーマットで使用している場合、スレーブアドレス、方向ビットの認識は行われず、スタートコンディション直後からデータとして扱われます。そのために、I2C0SR<TRX>はハードウェアによって変化することはありません。

3.14.6.6 スタート/ストップコンディションの発生

I2C0SR<BB> が 0 のときに、I2C0CR2<MST>, I2C0CR2<TRX>, I2C0CR2<BB>, I2C0CR2<PIN>に 1 を書き込むと、バス上にスタートコンディション、あらかじめデータバッファレジスタに書き込んだスレーブアドレスと、方向ビットが出力されます。スタートコンディションを発生させる前に、I2C0CR1<ACK>を 1 にセットしておいてください。

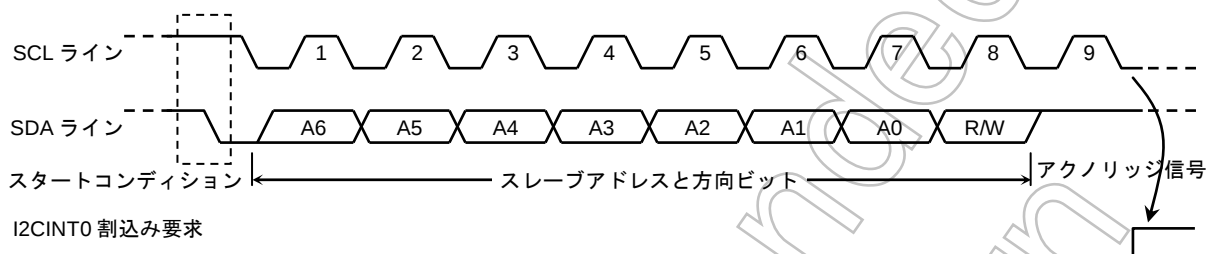


図 3.14.15 スタートコンディションの発生とスレーブアドレスの発生

I2C0SR<BB>が 1 のときに、I2C0CR2<MST>, I2C0CR2<TRX>, I2C0CR2<PIN>に 1 を、I2C0CR2<BB>に 0 を書き込むと、バス上にストップコンディションを出力するシーケンスが開始され、バス上にストップコンディションが発生します。

ストップコンディション発生時に、バスの SCL ラインがほかのデバイスにより“L”レベルに引かれていた場合、SCL ラインが解放された後に、ストップコンディションが発生します。

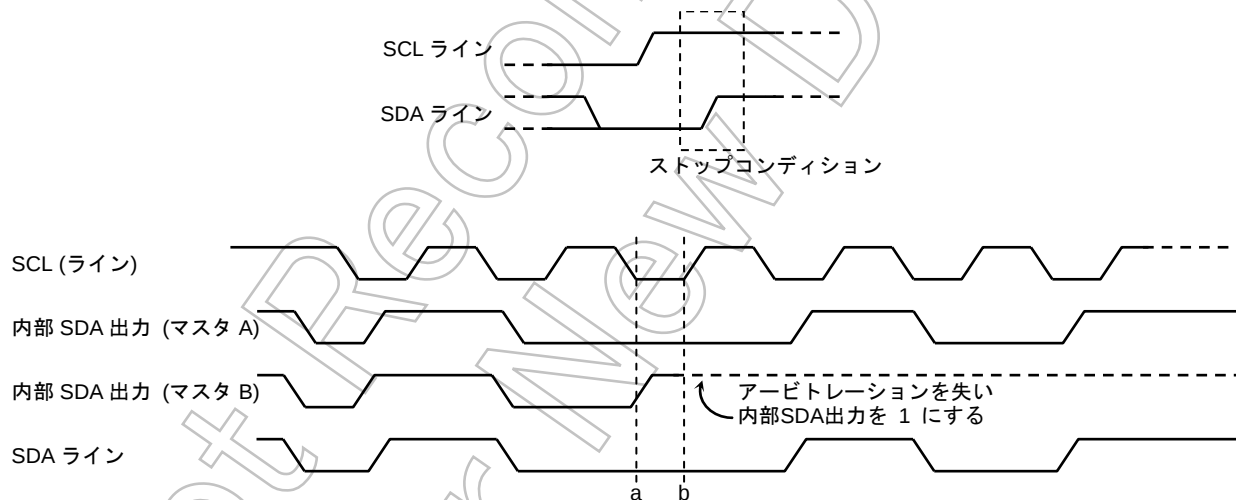


図 3.14.16 ストップコンディションの発生

また、I2C0SR<BB>を読み出すことで、バスの状態を知ることができます。I2C0SR<BB>は、バス上のスタートコンディションを検出すると 1 にセットされ (バスビジー状態)、ストップコンディションを検出すると 0 にクリアされます (バスフリー状態)。

一般的な動作として、I2C0SR の状態と、その際の設定例を示します。

I2C0CR2<MST>,<TRX>,<BB>,<PIN>のビットは本来独立した機能を持っていますが、I2C0SR の状態に応じて次のような定型的な組合せで使用します。

I2C0SR			I2C0CR2				動作
[7]MST	[5]BB	[4]PIN	[7]MST	[6]TRX	[5]BB	[4]PIN	
0	0	1	0	0	0	0	スレーブとしてスタートコンディション待ち
			1	1	1	1	スタートコンディション発生
1	1	0	1	1	0	1	ストップコンディション発生
			0	0	0	1	再スタートのため内部割込み解除

このビット(群)に対する書き込みの際に、誤って I2C0CR2<I2CM>を変更しないようにして下さい。

3.14.6.7 割り込みサービス要求と解除

マスタモードの場合、I2C0CR1<BC>と I2C0CR1<ACK>によって設定されたデータ転送のクロック数の転送が終了すると I2CINT0 割り込み要求が発生します。

スレーブモードの場合、上記に加え以下の条件が成立したとき I2CINT0 割り込み要求が発生します。

- I2C0CR1<NOACK>が 0 のとき、受信したスレーブアドレスが、I2C0AR<SA>に設定されたスレーブアドレスと一致したときのアクノリッジ信号出力後
 - I2C0CR1<NOACK>が 0 のとき、ゼネラルコールを受信したときのアクノリッジ信号出力後
 - スレーブアドレス一致、またはゼネラルコール受信後におけるデータ転送終了時
- I2CINT0 割り込み要求が発生すると、I2C0SR<PIN>が 0 にクリアされます。
I2C0SR<PIN>が 0 の間、I2C0CL が “L” レベルに引かれます。

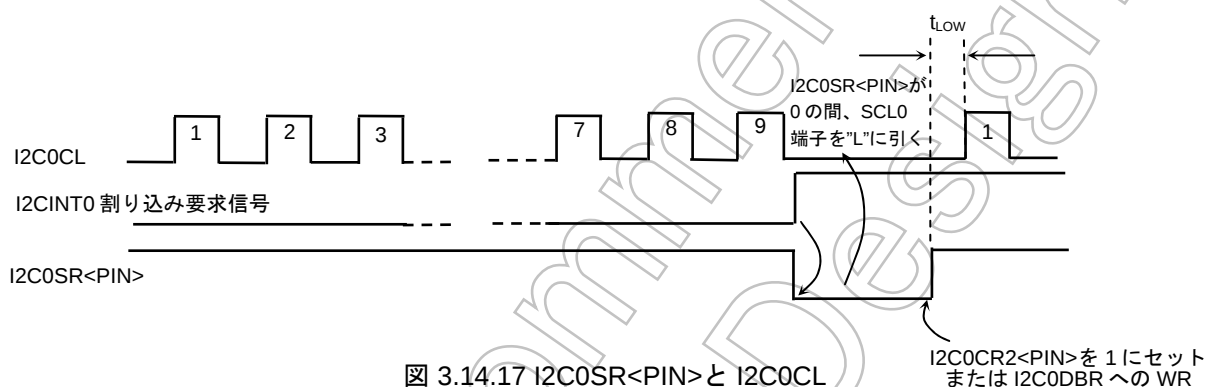


図 3.14.17 I2C0SR<PIN>と I2C0CL

I2C0DBR にデータを書き込むと I2C0SR<PIN>が 1 にセットされます。

なお、I2C0SR<PIN>が 1 にセットされてから I2C0CL が解放されるまで、 t_{LOW} の時間がかかります。プログラムで I2C0CR2<PIN>に 1 を書き込むと 1 にセットされますが、0 を書き込んでも 0 にクリアされません。

3.14.6.8 I²Cバスモード

I2C0CR2<I2CM>を1にセットするとI²Cバスモードになります。

I²Cバスモードで使用するときは、端子の状態が“H”になっていることを確認後、I2C0CR2<I2CM>を1にしてください。また、初期状態への切り替えはバスフリーを確認後、I2C0CR2<I2CM>を0にしてください。

注) I2C0CR2<I2CM>が0のとき、I2C0CR2<I2CM>以外のI2C0CR2に値を書き込むことはできません。

I2C0CR2に値を設定する前にI2C0CR2<I2CM>に1を書き込んでI²Cバスモードにしてください。

3.14.6.9 ソフトウェアリセット

I²Cは、I²Cを初期化するソフトウェアリセット機能を持っています。ノイズなどによりI²Cがロックしたとき、この機能を使うことでI²Cを初期化することができます。

I2C0CR2<SWRES[1:0]>に0y10、0y01の順に書き込みを行うとソフトウェアリセットが発生します。

ソフトウェアリセット発生後、I²Cが初期化されますが、I2C0CR2<I2CM>とI2C0DBRレジスタは初期化されません。

3.14.6.10 アービトレーションロスト検出モニタ

I²C バスではマルチマスタ (1 つのバス上で同時に 2 つ以上のマスタが存在する) が可能なため、転送されるデータの内容を保証するために、バスのアービトレーション手段が必要となります。

I²C バスでは、バスのアービトレーションに SDA ラインのデータを使用します。

アービトレーションの手順を、バス上に 2 つのマスタが同時に存在した場合を例にあげて以下に示します。

a 点のビットまでマスタ A、マスタ B とともに同じデータを出力し、a 点でマスタ B がデータ 1 を出力、マスタ A がデータ 0 を出力すると、バスの SDA ラインはワイヤードアンドで駆動されるために、SDA ラインはマスタ A によって “L” レベルに引かれます。

b 点でバスの SCL ラインが立ち上がると、スレーブデバイスは SDA ラインデータ、すなわちマスタ A のデータを取り込みます。

このときマスタ B の出力したデータは無効になります。マスタ B のこの状態を“アービトレーションロスト”と呼び、アービトレーションを失ったマスタ B は、I²C0DA、I²C0CL を解放し、アービトレーションを失っていない、マスタ A の出力するデータに影響を及ぼさないようにします。また、複数のマスタが、1 ワード目でまったく等しいデータを送信した場合、アービトレーションの手段は 2 ワード目以降も継続されます。

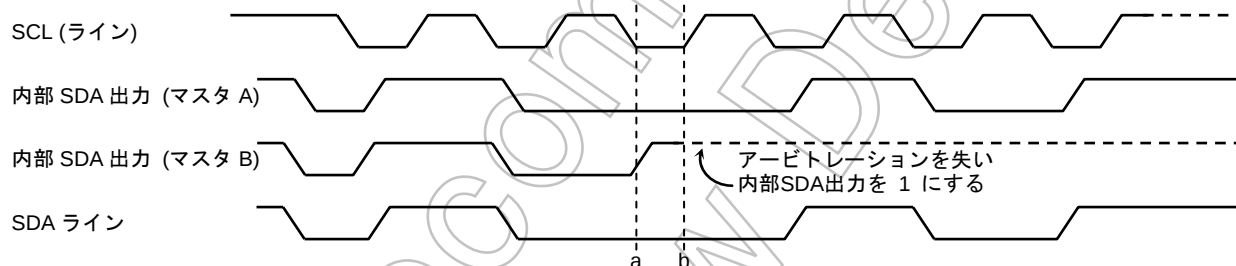


図 3.14.18 アービトレーションロスト

マスタ B は、バスの SDA ラインのレベルと I²C0DA のレベルの比較を、SCL ラインの立ち上がりで行います。このとき、不一致を検出するとアービトレーションを失い、I²C0SR<AL> が 1 にセットされます。

I²C0SR<AL> が 1 にセットされると I²C0SR<MST>, I²C0SR<TRX> は 0 にリセットされ、スレーブレシーバモードになります。そのため、I²C0SR<AL> が 1 にセットされた後のデータの転送では、マスタ B はクロックの出力を停止します。データ転送が終了すると I²C0SR<PIN> が 0 にクリアされ、I²C0CL が “L” に引かれます。

I²C0SR<AL> は、I²C0DBR にデータを書き込むか、I²C0DBR からデータを読み込む、または I²C0CR2 にデータを書き込むと 0 にリセットされます。

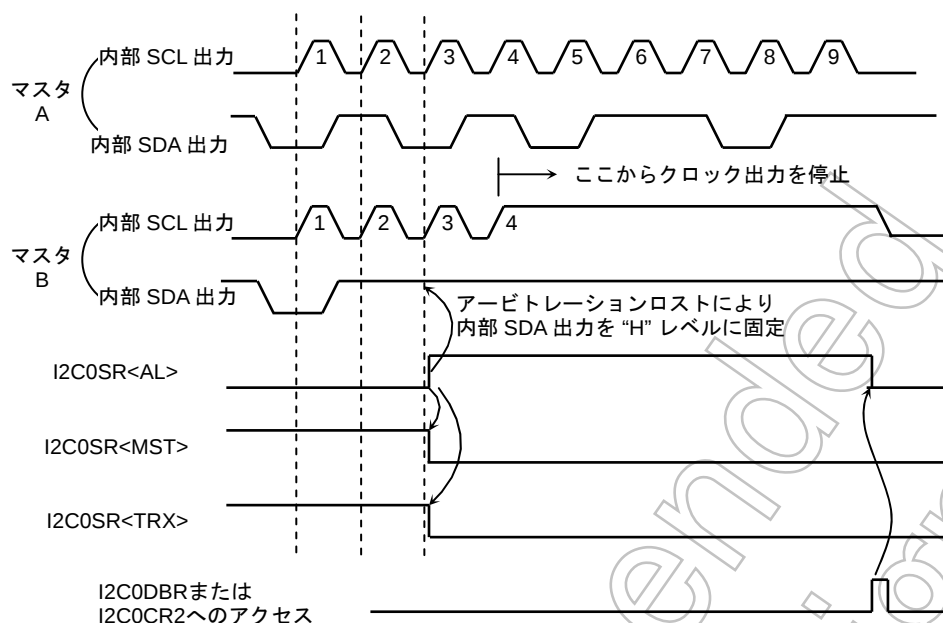


図 3.14.19 アービトレーションロスト動作（上記内部フラグはマスタ B を示す）

3.14.6.11 スレーブアドレス一致検出モニタ

I²C バスモード (I2C0AR<ALS> = 0) で、スレーブモードの場合に、スレーブアドレスの一致検出が出来ます。

I2C0CR1<NOACK>を 0 にクリアすると、アドレス一致検出を許可し、ゼネラルコールまたは、I2C0AR<SA>にセットした値と同じスレーブアドレスを受信することで、I2C0SR<AAS>は、1 にセットされます。

I2C0CR1<NOACK>を 1 にセットすると、アドレス一致検出を禁止し、ゼネラルコールまたは、I2C0AR<SA>にセットした値と同じスレーブアドレスを受信しても、I2C0SR<AAS>は、1 にセットされません。

フリーデータフォーマット (I2C0AR<ALS> = 1) のときは、アドレス一致検出としては機能せず、最初の 1 ワードが受信されると、I2C0SR<AAS>は 1 にセットされます。I2C0SR<AAS>は I2C0DBR にデータを書き込むか、I2C0DBR からデータを読み出すと 0 にクリアされます。



図 3.14.20 スレーブアドレス一致モニタの変化

3.14.6.12 ゼネラルコール検出モニタ

I²C バスモード (I2C0AR<ALS> = 0) で、スレーブモードの場合に、スレーブアドレスの一致検出と共に、ゼネラルコール検出が出来ます。

I2C0CR1<NOACK>が 0 のとき、I2C0SR<AD0>は、ゼネラルコール (スタートコンディション直後に受信した 8 ビットのデータがすべて 0) を受信すると、1 にセットされます。

(この時、同時に I2C0SR<AAS>も、1 にセットされます)

I2C0CR1<NOACK>が 1 のとき、ゼネラルコール検出を禁止します。そのため、ゼネラルコールを受信しても I2C0SR<AD0>は 0 のままです。

(この時、同時に I2C0SR<AAS>も、1 にセットされません)

I2C0SR<AD0>は、バス上のスタートコンディションまたはストップコンディションが検出されると、0 にクリアされます。

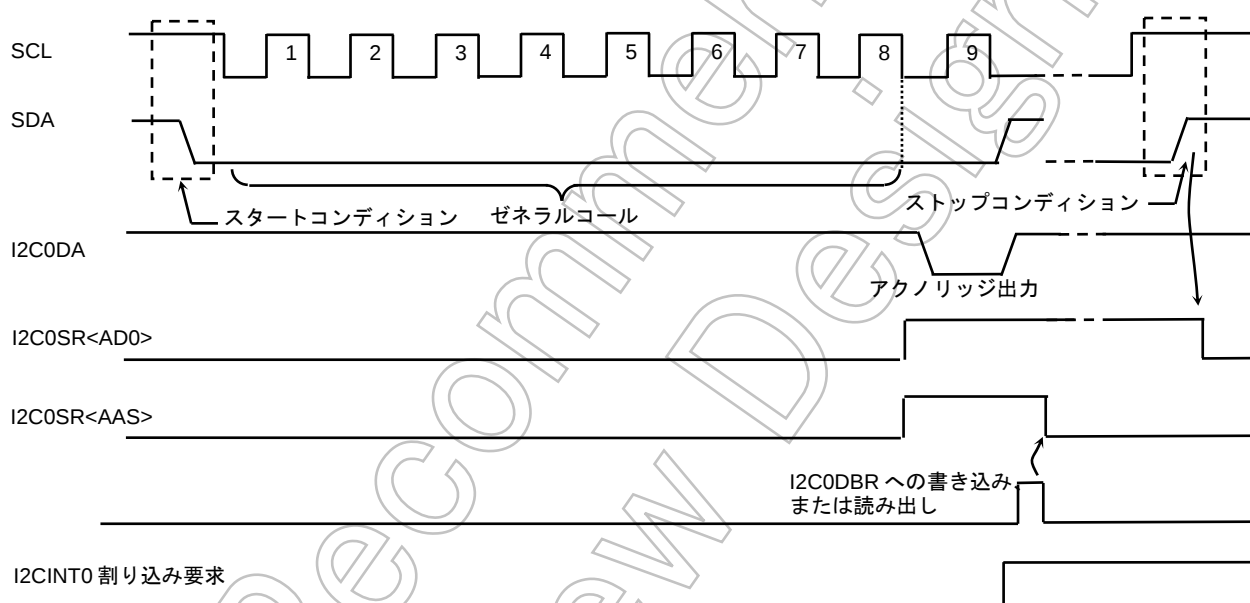


図 3.14.21 ゼネラルコール検出モニタの変化

3.14.6.13 最終受信ビットモニタ

I2C0SR<LRB>には、バス上の SCL ラインの立ち上がりで取り込まれた、SDA ラインの値が、常に更新されセットされます。

その為、アクノリッジメントモードのとき、I2CINT0 割り込み要求発生直後に I2C0SR<LRB>を読み出すと、アクノリッジ信号を読み出すことが出来ます。

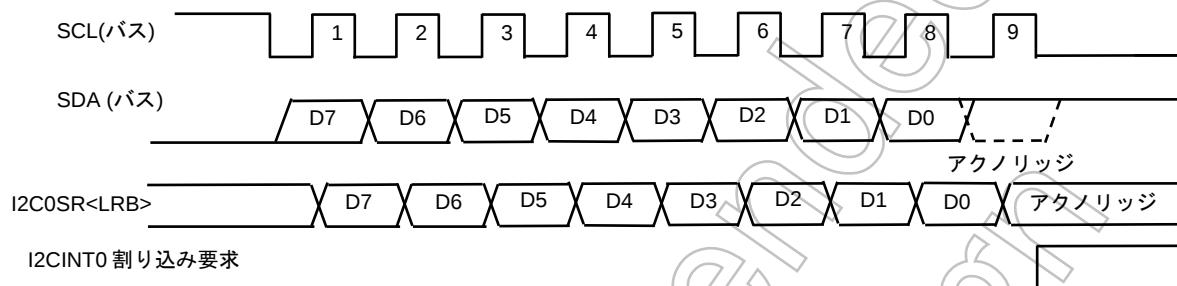


図 3.14.22 最終受信ビットモニタの変化

3.14.6.14 スレーブアドレスとアドレス認識モードの設定

I²C を、I²C バスモードで使用する際には、I2C0AR<ALS>を 0 にクリアし、I2C0AR<SA>にスレーブアドレスを設定します。

スレーブアドレスを認識しないフリーデータフォーマットで使用する際には、I2C0AR<ALS>を 1 にセットします。なお、I²C をフリーデータフォーマットで使した場合、スレーブアドレスと方向ビットの認識は行われず、スタートコンディション直後からデータとして扱われます。

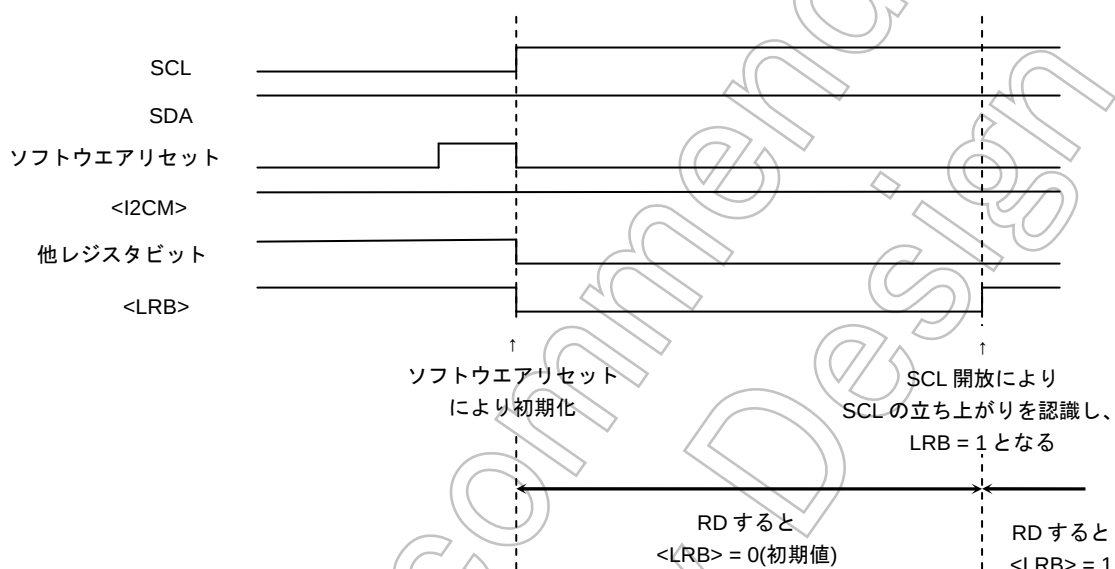
仕様上の注意点

3.14.6.15 ソフトウェアリセット後のレジスタ値

ソフトウェアリセットの実行により、I2C0CR2<I2CM>以外のレジスタと内部回路が初期化され、SCLおよびSDAは開放されます。(「3.14.6.3(2) クロック同期化参照)

ただし、I2C0SR<LRB>については、ソフトウェアリセット後リードするタイミングによって、初期値(0)と異なる値が読み出される場合があります。

<SDA = 1 の状態で、ソフトウェアリセットにより SCL=0→1 に開放する場合>



3.15 SSP (Synchronous Serial Port)

同期式シリアルインタフェース(SSP: Synchronous Serial Port)を 1 チャンネル内蔵しています。
チャンネルの特長を下記の表に示します。

	チャンネル 0
通信プロトコル	SPI を含む 3 種類の同期式シリアル
動作モード	マスタ/スレーブモード
送信 FIFO	幅 16-bit / 深さ 8 段
受信 FIFO	幅 16-bit / 深さ 8 段
送受信データのサイズ	4-16 ビット
割り込みの種類	送信割り込み 受信割り込み 受信オーバーラン割り込み タイムアウト割り込み
通信速度	マスタモード時: $f_{PCLK}/2$ (ただし最大 20Mbps) スレーブモード時: $f_{PCLK}/12$ (最大 8.33Mbps)
DMA	サポート
内部テスト機能	内部ループバックテストモードの使用が可能
制御端子	SP0CLK SP0FSS SP0DO SP0DI

3.15.1 ブロック図

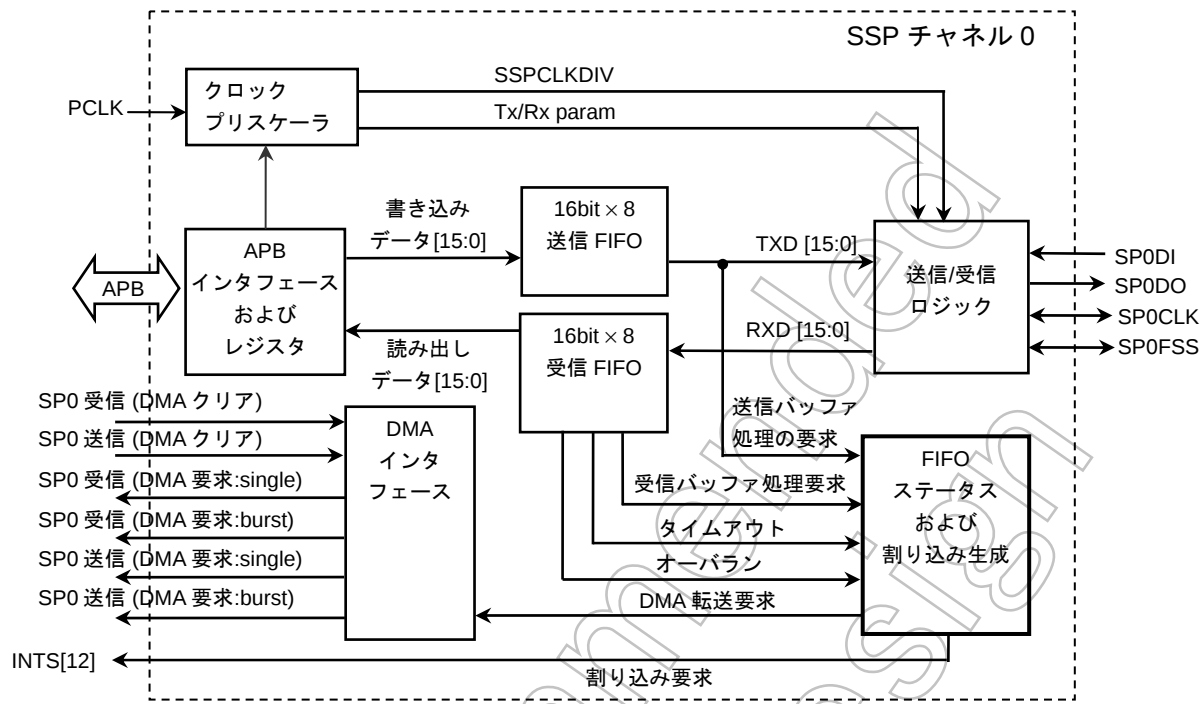


図 3.15.1 SSP ブロック図

3.15.2 SSPの概要

SSP は、3 種類の同期シリアルインタフェース機能を持つ周辺デバイスと、シリアル通信が可能なインタフェースです。

SSP は、周辺デバイスから受信したデータのシリアル・パラレル変換を実行します。送信パスと受信パスは、送信モードと受信モードにおいて 16 ビット幅 8 段のそれぞれ独立した送信 FIFO と受信 FIFO にデータをバッファリングします。シリアルデータは、SP0DO から送信し、SP0DI から受信します。

SSP には、入力クロック PCLK からシリアル出力クロック SP0CLK を生成するために、プログラム可能なプリスケアラが含まれています。SSP の動作モード、フレームフォーマットならびにサイズは、制御レジスタ SSP0CR0 および SSP0CR1 を通じてプログラムします。

(1) クロックプリスケアラ

マスタとして動作する場合、シリアル出力クロック SP0CLK を生成するために、シリアルにリンクした 2 つのフリーランカウンタから構成されるクロックプリスケアラが使用されます。

このクロックプリスケアラは、SSP0CPSR レジスタを介し、2 ~ 254 の偶数ステップで PCLK を除算するようにプログラムすることができます。SSP0CPSR レジスタの最下位ビットを使用しないことにより、奇数ステップによるプログラミングはされなくなります。

プリスケアラの出力は、さらに SSP0CR0 制御レジスタにプログラミングされた値に+1 された 1~256 のステップで除算され、マスタ出力クロック SP0CLK が生成されます。

$$\text{ビットレート} = f_{\text{PCLK}} / (\text{CPSDVSR} \times (1 + \text{SCR}))$$

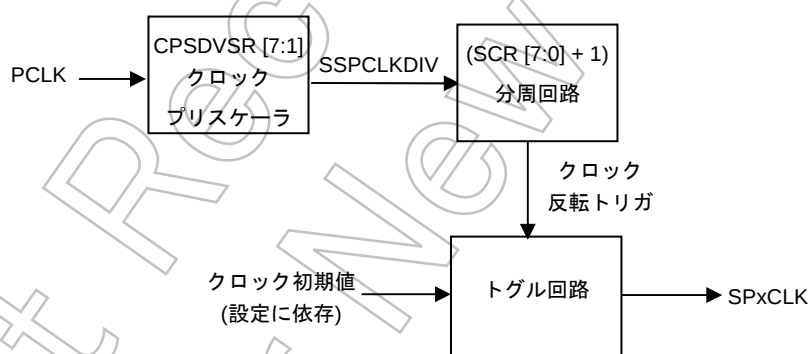


図 3.15.2 クロックプリスケアラブロック図

(2) 送信 FIFO

マスタおよびスレーブモード共有の 16 ビット幅 8 段の送信 FIFO バッファです。

(3) 受信 FIFO

マスタおよびスレーブモード共有の 16 ビット幅 8 段の受信 FIFO バッファです。

(4) 割り込み

個々にマスク可能な 4 種類の割り込みをサポートしています。

また、個々の割り込み要求はひとつの結合割り込みとして出力されます。

- 送信割り込み: TxFIFO の空領域が半分以上である条件割り込み
(TxFIFO 内の有効データ数 ≤ 4)
- 受信割り込み: RxFIFO の有効データが半分以上である条件割り込み
(RxFIFO 内の有効データ数 ≥ 4)
- タイムアウト割り込み: タイムアウトまでに読み出されていないデータが RxFIFO に存在することを示す割り込み
- 受信オーバラン割り込み: RxFIFO がフルのときに書き込みが行われたことを示す条件割り込み

個々にマスク可能な 4 つの割り込みは、割り込み許可/禁止レジスタ内の対応ビットをセットすることによってマスクできます。適切なマスクビットを High にセットすると、その割り込みがイネーブルされます。

(a) 送信割り込み

送信割り込みは、送信 FIFO 内の有効データ数が 4 エントリ以下になるとアサートされます。SSP 動作ディセーブル(SSPxCR1<SSE>=0y0)でも送信割り込みが発生します。

初回の送信データは、本割り込みを使用して FIFO に書き込むことができます。

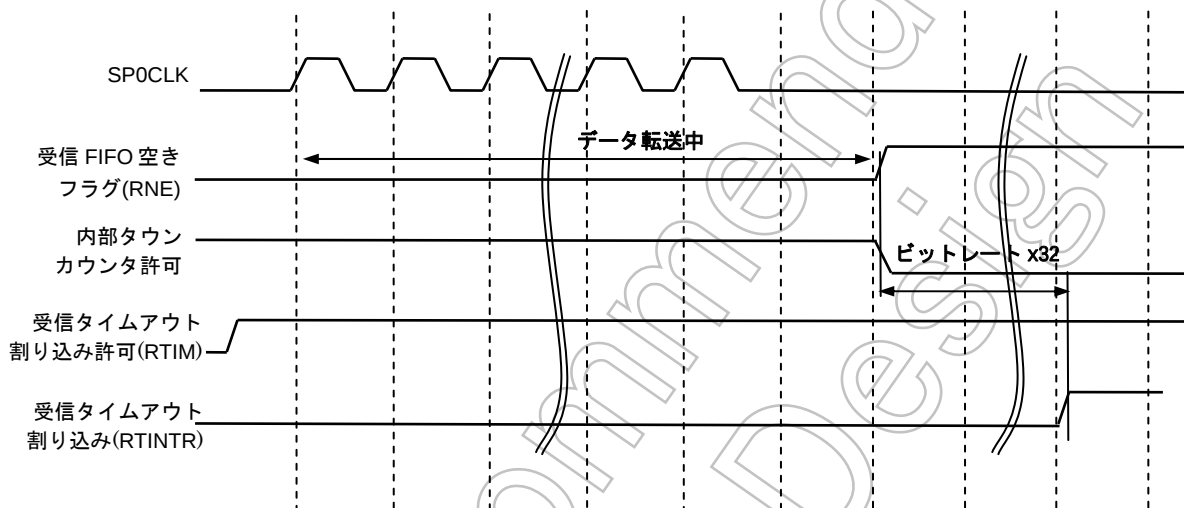
(b) 受信割り込み

受信 FIFO 内の有効データ数が 4 エントリ以上になるとアサートされます。

送受信割り込みのクリアについて、FIFO の有効エントリ数を監視し、ウォーターマークを境に、ダイナミックに割り込み要求、割り込みクリアが発生します。要求クリアのレジスタが存在しません。

(c) タイムアウト割り込み

受信タイムアウト割り込みは、受信 FIFO が空ではないときに SSP が一定の 32 ビット周期(ビットレート)の間アイドル状態を続けるとアサートされます。この機構により、ユーザは受信 FIFO にまだデータが存在し、処理が必要であることを認識することができます。マスタ/スレーブ モード共に発生します。タイムアウト割り込みが発生した場合は、受信 FIFO から全データをリードしてください。全データをリードしない場合でも、受信 FIFO に空きがあり、かつ転送データ数が受信 FIFO の空き以下であれば送受信可能です。尚、転送開始でタイムアウト割り込みはクリアされます。また、受信 FIFO に空きがない状態で送受信を行った場合、タイムアウト割り込みはクリアされオーバーラン割り込みが発生します。



(d) 受信オーバーラン割り込み

受信 FIFO が既にフルのときに次のデータ(9 個目)を受信すると、転送終了直後、受信オーバーラン割り込みが発生します。受信オーバーラン割り込み発生以降の受信データ(9 個目のデータを含む)は無効となり、破棄されます。ただし、9 個目のデータ受信(割り込み発生する前)に受信 FIFO のデータをリードすると、9 個目の受信データは有効データとして受信 FIFO に書き込まれます。受信オーバーラン割り込みが発生した場合、正しい転送を行なうためには、受信オーバーラン割り込みクリアレジスタに“1”を書き込んだ後、受信 FIFO から全データをリードしてください。全データをリードしない場合でも、受信 FIFO に空きがあり、かつ転送データ数が受信 FIFO の空き以下であれば送受信可能です。なお、受信オーバーラン割り込みをクリアした後、一定の 32 ビット周期(ビットレート)の間、受信 FIFO をリードしない場合(受信 FIFO が空ではない時)はタイムアウト割り込みが発生します。

(e) 結合割り込み

上記の 4 つの割り込みは、マスクされた個々のソースを 1 つの割り込みに結合しています。上記割り込みのいずれかがアサートされると、結合割り込み INTS[12]がアサートされます。

(5) DMA インタフェース

DMA コントローラに接続するインタフェースを提供します。

3.15.3 SSPの動作

(1) SSP の初期設定

SSP 通信プロトコルの設定は SSP がディセーブルの状態、設定する必要があります。

制御レジスタ SSP0CR0 および SSP0CR1 を使って、以下のいずれかのプロトコルの下、本 SSP をマスタまたはスレーブを設定してください。また、通信速度に関して、プリスケールレジスタ SSP0CPSR および SSP0CR0<SCR>を設定してください。

本 SSP は以下のフレームフォーマットをサポートします。

- SPI / SSI / Microwire

(2) SSP のイネーブル

送信 FIFO に送信データが書き込まれていて、且つ動作イネーブルした時、もしくは動作イネーブル状態で送信 FIFO に送信データが書き込まれたときに転送が開始されます。

ただし、動作イネーブルにした時に送信 FIFO が 4 エントリ以下の場合は送信割り込みが発生します。この割り込みを使って初回データを書き込むことも可能です。

注) SPI のスレーブモードで且つ FSS 端子未使用の場合、イネーブルする前に必ず 1Byte 以上のデータを送信 FIFO に書き込んでください。送信 FIFO が空の状態ではイネーブルすると、転送データが正常に出力されません。

(3) クロック比

PCLK の周波数設定は以下の条件を満たす必要があります。

[マスタモードの場合]

$$f_{SP0CLK}(\text{最大}) : f_{PCLK} / 2$$

$$f_{SP0CLK}(\text{最小}) : f_{PCLK} / (254 \times 256)$$

[スレーブモードの場合]

$$f_{SP0CLK}(\text{最大}) : f_{PCLK} / 12$$

$$f_{SP0CLK}(\text{最小}) : f_{PCLK} / (254 \times 256)$$

(4) フレーム形式

各フレームフォーマットは、プログラムされたデータのサイズにより、4~16 ビットの幅を持ち、MSB から送信が開始されます。

• シリアルクロック(SP0CLK)

SSP がアイドル状態の間は SSI と Microwire では Low でホールドされ、SPI では非アクティブでホールドされます。また、データ送信中にのみ設定されたビットレートで出力されます。

• シリアルフレーム(SP0FSS)

SPI および Microwire のフレーム形式では、Low アクティブとなり、フレームの送信中は常に Low にアサートされます。

SSI フレーム形式の場合は、各フレームの送信前に 1 ビットレートの間のみアサートされます。このフレーム形式では、SP0CLK の立ち上がりエッジで出力データを送信し、その立下りエッジで入力データを受信します。

- Microwire について

Microwire 形式は半二重モードで動作する特殊なマスタ・スレーブ伝達方式を使用します。このモードでは、フレームの開始時に 8 ビットの制御メッセージがスレーブに送信されます。この送信中、SSP はデータを受信しません。メッセージが送信された後、スレーブがそれをデコードし、8 ビット制御メッセージの末尾ビットの送信から 1 シリアルクロック間待機した後、要求されたデータで応答します。戻されるデータの長さは 4~16 ビットであり、全体のフレーム長は 13~25 ビットとなります。

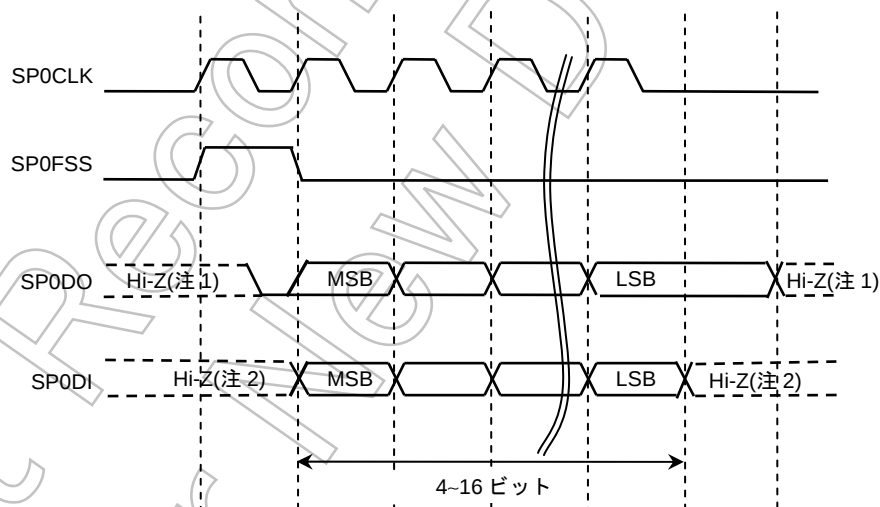
各フレーム フォーマットの詳細は下記を参照してください。

1) SSI のフレームフォーマット

このモードでは、SSP はアイドル状態で SP0CLK および SP0FSS が強制的に Low にセットされ、送信データライン SP0DO が Hi-Z になります。送信 FIFO にデータをライトすると、マスタは SP0FSS ラインに 1 SP0CLK 分の High パルスを出力します。送信データは送信 FIFO から送信のシリアルシフトレジスタに転送されます。SP0CLK の次の立ち上がりエッジで、MSB から 4~16 ビットのデータが SP0DO 端子により出力されます。

同様に、受信データは SP0CLK の立ち下がりエッジで MSB から SP0DI 端子に入力されます。受信されたデータは、その LSB データがラッチされた後、SP0CLK の立ち上がりエッジでシリアルシフトレジスタから受信 FIFO に転送されます。

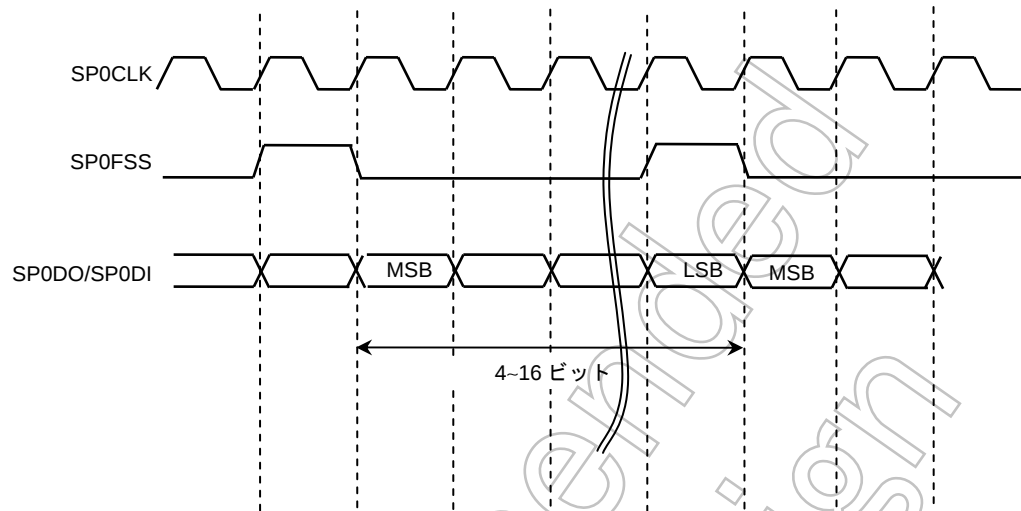
SSI フレームフォーマット(シングル転送送受信)



注1) SP0DO 端子は非送信中に出力 OFF となり Hi-Z 状態となります。そのため、システムに合わせプルアップ/プルダウン抵抗によるレベル固定をしてください。

注2) SP0DI 端子は常時入力となっていますので、送信側が非送信中に出力 OFF となる場合はプルアップ/プルダウン抵抗によるレベル固定をしてください。

SSI フレームフォーマット(連続転送送受信)



注1) SP0DO 端子は非送信中に出力 OFF となり Hi-Z 状態となります。そのため、システムに合わせプルアップ/プルダウン抵抗によるレベル固定をしてください。

注2) SP0DI 端子は常時入力となっていますので、送信側が非送信中に出力 OFF となる場合はプルアップ/プルダウン抵抗によるレベル固定をしてください。

2) SPI

SPI インタフェースは 4 線インタフェースであり、SP0FSS はスレーブ選択として動作します。SPI フォーマットの主な特長は、SP0CLK の動作タイミングを SSP0CR0 制御レジスタ内の<SPO> ビットと<SPH> ビットを使って、設定できます。

SSP0CR0<SPO>

SSP0CR0<SPO> はアイドル状態時の SP0CLK をホールドするレベルを設定します。

<SPO>= 0y1: SP0CLK は High 状態

<SPO>= 0y0: SP0CLK は Low 状態

SSP0CR0<SPH>

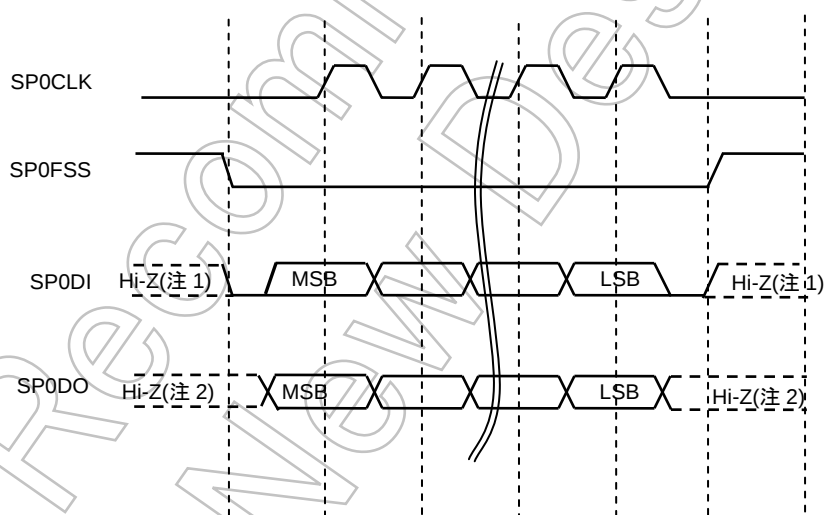
SSP0CR0<SPH>はデータをラッチするクロックエッジを選択します。

SSP0CR0<SPH>= 0y0: 1st クロックエッジでデータを取り込みます。

SSP0CR0<SPH>= 0y1: 2nd クロックエッジでデータを取り込みます。

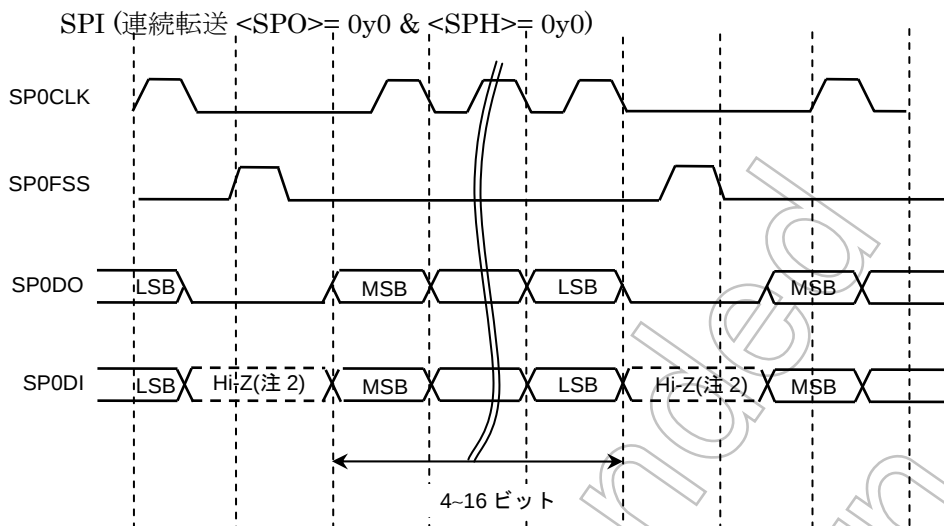
SPI の動作例:

SPI (シングル転送, <SPO>= 0y0 & <SPH>= 0y0)



注1) SP0DO 端子は非送信中に出力 OFF となり Hi-Z 状態となります。そのため、システムに合わせプルアップ/プルダウン抵抗によるレベル固定をしてください。

注2) SP0DI 端子は常時入力となっていますので、送信側が非送信中に出力 OFF となる場合はプルアップ/プルダウン抵抗によるレベル固定をしてください。



注 1) SP0DO 端子は非送信中に出力 OFF となり Hi-Z 状態となります。そのため、システムに合わせプルアップ/プルダウン抵抗によるレベル固定をしてください。

注 2) SP0DI 端子は常時入力となっていますので、送信側が非送信中に出力 OFF となる場合はプルアップ/プルダウン抵抗によるレベル固定をしてください。

この設定におけるアイドル状態では:

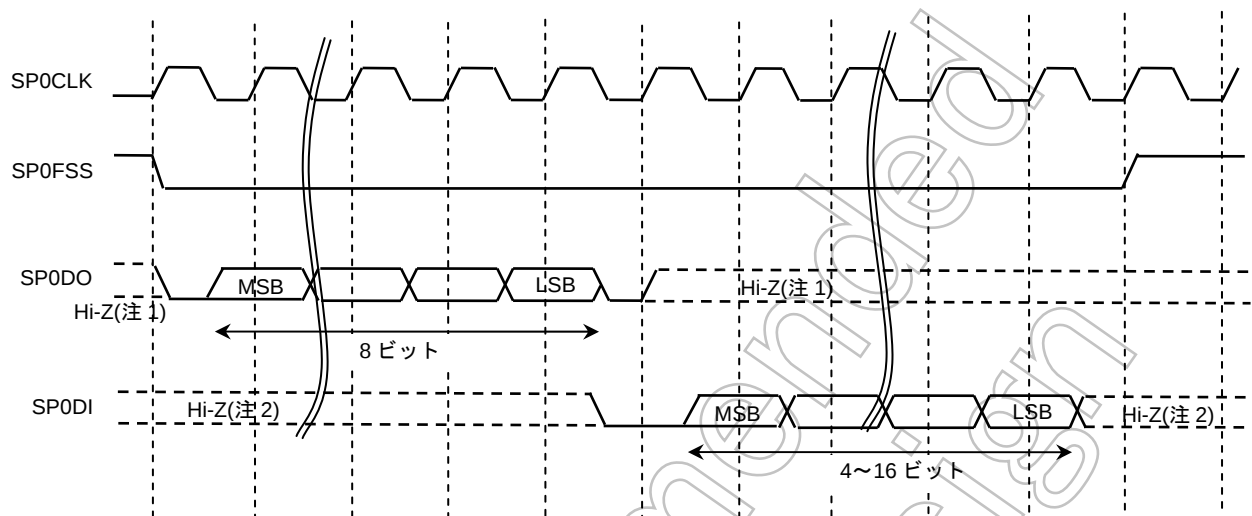
- SP0CLK シグナル が強制的に Low にセットされます。
- SP0FSS が強制的に High にセットされます。
- 送信データライン SP0DO が任意に LOW にセットされます。

SSP がイネーブルされているときに送信 FIFO 内に有効データが存在する場合、送信開始は LOW でドライブされる SP0FSS マスタシグナルによって通知されます。これにより、マスタの SP0DI 入力ラインでスレーブデータがイネーブルされます。

SP0CLK の半周期後、有効マスタデータが SP0DO ピンに転送されます。これで、マスタデータとスレーブデータの両方がセットされ、SP0CLK のさらに半周期後に SP0CLK マスタクロックピンが HIGH になります。その後、データは SP0CLK シグナルの立ち上がりエッジで捕捉され、立下りエッジで伝達されます。シングルワード転送の場合、そのデータワードの全ビットの転送が完了し、その末尾ビットが捕捉されてから SP0CLK 1 周期後に、SP0FSS ラインがアイドル HIGH 状態に戻ります。しかし、連続転送の場合には、各データワード転送間で SP0FSS シグナルを HIGH でパルスする必要があります。これは、スレーブ選択ピンがそのシリアルペリフェラルレジスタにデータをフリーズし、<SPH> ビットが論理 0 の場合は変更を許可しないためです。したがって、シリアルペリフェラルデータの書き込みをイネーブルするには、マスタデバイスが各データ転送間でスレーブデバイスの SP0FSS ピンをドライブする必要があります。連続転送が完了すると、末尾ビットが捕捉されてから SP0CLK 1 周期後に SP0FSS ピンがアイドル状態に戻ります。

3) Microwire フレームフォーマット

Microwire フレームフォーマット(シングル転送)



注 1) SP0DO 端子は非送信中に出力 OFF となり Hi-Z 状態となります。そのため、システムに合わせプルアップ/プルダウン抵抗によるレベル固定をしてください。

注 2) SP0DI 端子は常時入力となっていますので、送信側が非送信中に出力 OFF となる場合はプルアップ/プルダウン抵抗によるレベル固定をしてください。

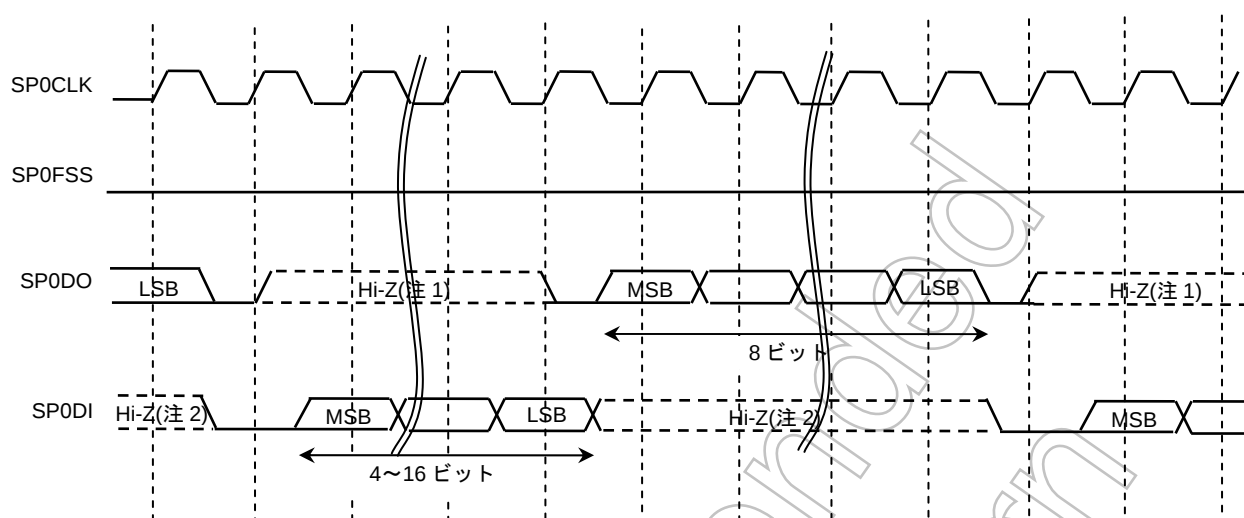
Microwire 形式は SPI 形式と非常に似ていますが、全二重ではなく半二重通信のマスター-スレーブメッセージ伝達方式を使用する点が異なります。各シリアル伝送は、SSP からオフチップスレーブデバイスに送信される 8 ビットの制御ワードで開始されます。この送信中、SSP は入力データを受信しません。このメッセージの送信後、オフチップスレーブがこれをデコードし、8 ビット制御メッセージの末尾ビット送信後から 1 シリアルクロックが経過するまで待機し、要求されたデータで応答します。戻されるデータは $4 \square 16$ ビット長、フレーム全体は $13 \square 25$ ビット長になります。このコンフィギュレーションにおけるアイドル周期では：

- ・ SP0CLK シグナルが強制的に LOW にセットされます。
- ・ SP0FSS が強制的に HIGH にセットされます。
- ・ 送信データライン SP0DO が任意に LOW にセットされます。

伝送は送信 FIFO への制御バイトの書き込みによってトリガされます。SP0FSS の立下りエッジによって送信 FIFO の最下位エンタリにストアされていた値が送信ロジックのシリアルシフトレジスタに転送され、8 ビット制御フレームの MSB が SP0DO ピンにシフトアウトされます。このフレーム伝送の間、SP0FSS は LOW でホールドされ、SP0DI ピンはトライステートでホールドされます。オフチップシリアルスレーブデバイスは、SP0CLK の立ち上がりエッジでそのシリアルシフトに各制御ビットをラッチします。スレーブデバイスによって末尾ビットがラッチされると、1 クロックウェイト状態の間にその制御バイトがデコードされ、スレーブはデータを送信することによって SSP に応答します。各ビットは、SP0CLK の立下りエッジで SP0DI ラインにドライブされます。一方、SSP は SP0CLK の立ち上がりエッジで各ビットをラッチします。シングル転送の場合は、フレームの終わりで末尾ビットが受信シリアルシフトにラッチされてから 1 クロック周期後に SP0FSS シグナルが HIGH にプルされ、これによってデータが受信 FIFO に転送されます。

注) オフチップスレーブデバイスは、受信シフトによって LSB がラッチされた後の SP0CLK の立下りエッジで、または SP0FSS ピンが HIGH になるときのどちらかで、受信ラインをトライステートでホールドすることができます。

Microwire フレームフォーマット(連続転送)



注 1) SP0DO 端子は非送信中に出力 OFF となり Hi-Z 状態となります。そのため、システムに合わせプルアップ/プルダウン抵抗によるレベル固定をしてください。

注 2) SP0DI 端子は常時入力となっていますので、送信側が非送信中に出力 OFF となる場合はプルアップ/プルダウン抵抗によるレベル固定をしてください。

連続転送の場合、データ転送の開始と終了はシングル転送の場合と同じように行われます。しかし、SP0FSS ラインは常にアサート (LOW でホールド) され、データの伝送が次から次に発生します。次のフレームの制御バイトは、現在のフレームから受信されたデータの LSB の直後に続きます。受信された各値は、そのフレームの LSB が SSP にラッチされた後、SP0CLK の立下りエッジで受信シフトから転送されます。

注) (接続例)

SSP は、システム内におけるマスタ/スレーブの dynamics 切り替えをサポートしていません。各 SSP の例は、マスタまたはスレーブのどちらかとして構成/接続されています。

(5) DMA インタフェース

SSP の DMA 操作は DMA 制御レジスタ SSP0DMACR を介して制御されます。

受信 FIFO に格納されるデータがウォーターマークレベル(FIFO の 1/2)以上になると受信 DMA 要求がアサートされます。

送信 FIFO に格納するデータがウォーターマークレベル(FIFO の 1/2)以下になると送信 DMA 要求がアサートされます。

送信/受信 DMA 要求信号をクリアするために、DMA コントローラによってアサートされる送信/受信 DMA 要求クリア信号の入力端子を持っています。

また、DMA のバースト長は 4 ワードに設定してください。

※残りの 3 文字に対し、SSP はバースト要求をアサートしません。

各要求シグナルは、関連する DMA クリアシグナルがアサートされるまでアサートされ続けます。要求クリアシグナルがアサート解除された後、前述の条件に従って要求シグナルを再度アクティブにすることができます。SSP がディセーブルされたり、DMA イネーブルシグナルがクリアされると、全ての要求シグナルがアサート解除されます。

以下の表は、送信 FIFO および受信 FIFO に対する DMABREQ のトリガポイントを示しています。

ウォーターマークレベル	バースト長	
	送信 (空き位置の数)	受信 (充填位置の数)
1/2	4	4

3.15.4 レジスタの説明

SFR のリストを以下に示します。

- SSP0

base アドレス= 0xF200_2000

Register Name	Address (base+)	Description
SSP0CR0	0x0000	SSP0 Control register 0
SSP0CR1	0x0004	SSP0 Control register 1
SSP0DR	0x0008	SSP0 Data register
SSP0SR	0x000C	SSP0 Status register
SSP0CPSR	0x0010	SSP0 Clock prescale register
SSP0IMSC	0x0014	SSP0 Interrupt mask set and clear register
SSP0RIS	0x0018	SSP0 Raw interrupt status register
SSP0MIS	0x001C	SSP0 Masked interrupt status register
SSP0ICR	0x0020	SSP0 Interrupt clear register
SSP0DMACR	0x0024	DMA Control register
–	0x0028 ~ 0xFFC	Reserved

1. SSP0CR0 (SSP0 Control register 0)

Address = (0xF200_2000) + 0x0000

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined. Write as zero.
[15:8]	SCR	R/W	0y0	シリアルクロックレート設定用 パラメータ: (個別説明参照) 0x00 ~ 0xFF
[7]	SPH	R/W	0y0	SPCLK フェーズ 0y0: 1st クロックエッジでデータを取り込み 0y1: 2nd クロックエッジでデータを取り込み (SPI フレームフォーマットのみに適用可能, 2) SPI 参照)。
[6]	SPO	R/W	0y0	SPCLK 極性 0y0: SP0CLK は Low 状態 0y1: SP0CLK は High 状態 (SPI フレームフォーマットのみに適用可能, 2) SPI 参照)。
[5:4]	FRF	R/W	0y00	フレームフォーマット: 0y00: SPI フレームフォーマット 0y01: SSI フレームフォーマット 0y10: Microwire フレームフォーマット 0y11: Reserved、動作未定義
[3:0]	DSS	R/W	0y0000	データサイズ選択: 0y0000: Reserved、動作未定義 0y0001: Reserved、動作未定義 0y0010: Reserved、動作未定義 0y0011: 4 ビットデータ 0y0100: 5 ビットデータ 0y0101: 6 ビットデータ 0y0110: 7 ビットデータ 0y0111: 8 ビットデータ 0y1000: 9 ビットデータ 0y1001: 10 ビットデータ 0y1010: 11 ビットデータ 0y1011: 12 ビットデータ 0y1100: 13 ビットデータ 0y1101: 14 ビットデータ 0y1110: 15 ビットデータ 0y1111: 16 ビットデータ

(個別説明)

a. <SCR>

SSP の送信ビットレートおよび受信ビットレートの生成に使用されます。

このビットレートは以下の式から求められます。

ビットレート = $f_{PCLK} / (<CPSDVSr> \times (1 + <SCR>))$

<CPSDVSr> は、SSPxCPSP レジスタを参照してください。

2. SSP0CR1 (SSP0 Control register 1)

Address = (0xF200_2000) + 0x0004

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	–	–	Undefined	Read as undefined. Write as zero.
[3]	SOD	R/W	0y0	スレーブモード SP0DO 出力 Disable: 0y0: Enable 0y1: Disable
[2]	MS	R/W	0y0	マスタ/スレーブモード選択: 0y0: デバイスがマスタ 0y1: デバイスがスレーブ
[1]	SSE	R/W	0y0	SSP イネーブル: 0y0: Disable 0y1: Enable
[0]	Reserved	R/W	0y0	Write as zero.

(個別説明)

a. <SOD>

スレーブモード出力ディセーブル。このビットは、スレーブモード(<MS>=0y1)の場合にのみ作用します。

b. <MS>

マスタ/スレーブの切り替えビットです。スレーブモードで且つ送信時は、以下の設定順序を必ず守ってください。

1. スレーブモードに設定(<MS>=0y1)
2. 送信データを FIFO に設定
3. SSP をイネーブルに設定(<SSE>=0y1)

3. SSP0DR (SSP0 Data register)

Address = (0xF200_2000) + (0x0008)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined. Write as zero.
[15:0]	DATA	R/W	0x0000	送信/受信 FIFO のデータ: 0x00 ~ 0xFF

(個別説明)

a. <DATA>

リード時: 受信 FIFO

ライト時: 送信 FIFO

16 ビット未満のデータサイズに合わせてプログラムしている場合は、データを右寄せにする必要があります。先頭の未使用ビットは送信ロジックによって無視されます。受信ロジックは自動的に右寄せを行います。

4. SSP0SR (SSP0 Status register)

Address = (0xF200_2000) + (0x000C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:5]	–	–	Undefined	Read as undefined.
[4]	BSY	RO	0y0	ビジーフラグ: 0y0: アイドル 0y1: ビジー
[3]	RFF	RO	0y0	受信 FIFO フルフラグ: 0y0: 受信 FIFO がフルではない 0y1: 受信 FIFO がフル
[2]	RNE	RO	0y0	受信 FIFO 空きフラグ: 0y0: 受信 FIFO が空 0y1: 受信 FIFO が空ではない
[1]	TNF	RO	0y1	送信 FIFO フルフラグ: 0y0: 送信 FIFO がフル 0y1: 送信 FIFO がフルではない
[0]	TFE	RO	0y1	送信 FIFO 空きフラグ: 0y0: 送信 FIFO が空ではない 0y1: 送信 FIFO が空

(個別説明)

a. <BSY>

BSY= “0y1”時、現在フレームの送信中/ 受信中であるか、送信 FIFO が空ではないことを示します。

5. SSP0CPSR (SSP0 Clock prescale register)

Address = (0xF200_2000) + (0x0010)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:0]	CPSDVSR	R/W	0x0000	クロックプリスケール除数: 2 ~254 の偶数値を設定してください。

(個別説明)

a. <CPSDVSR>

クロックプリスケール除数。PCLK の周波数に基づき、2 ~ 254 の偶数値を設定する必要があります。最下位ビットは読み出し時に常に 0y0 を戻します。

6. SSP0IMSC (SSP0 Interrupt mask set and clear register)

Address = (0xF200_2000) + (0x0014)

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	–	–	Undefined	Read as undefined. Write as zero.
[3]	TXIM	R/W	0y0	送信 FIFO 割り込み許可: 0y0: 禁止 0y1: 許可
[2]	RXIM	R/W	0y0	受信 FIFO 割り込み許可: 0y0: 禁止 0y1: 許可
[1]	RTIM	R/W	0y0	受信タイムアウト割り込み許可: 0y0: 禁止 0y1: 許可
[0]	RORIM	R/W	0y0	受信オーバーラン割り込み許可: 0y0: 禁止 0y1: 許可

(個別説明)

a. <TXIM>

TxFIFO に半分以下の空領域がある条件割り込みの許可/禁止

b. <RXIM>

RxFIFO の半分以下がフルである条件割り込みの許可/禁止

c. <RTIM>

タイムアウト期限までに RxFIFO にデータが存在し、読み出されていないことを示す割り込みの許可/禁止。

d. <RORIM>

RxFIFO がフルのときに書き込みが行われたことを示す条件割り込みの許可/禁止。

7. SSP0RIS (SSP0 Raw interrupt status register)

Address = (0xF200_2000) + (0x0018)

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	–	–	Undefined	Read as undefined.
[3]	TXRIS	RO	0y1	許可ゲート前の送信割り込みフラグ: 0y0: 割り込み無し 0y1: 割り込み有り
[2]	RXRIS	RO	0y0	許可ゲート前の受信割り込みフラグ: 0y0: 割り込み無し 0y1: 割り込み有り
[1]	RTRIS	RO	0y0	許可ゲート前の受信タイムアウト割り込みフラグ: 0y0: 割り込み無し 0y1: 割り込み有り
[0]	RORRIS	RO	0y0	許可ゲート前の受信オーバーラン割り込みフラグ: 0y0: 割り込み無し 0y1: 割り込み有り

8. SSP0MIS (SSP0 Masked interrupt status register)

Address = (0xF200_2000) + (0x001C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	–	–	Undefined	Read as undefined.
[3]	TXMIS	RO	0y0	許可ゲート後の送信割り込みフラグ: 0y0: 割り込み無し 0y1: 割り込み有り
[2]	RXMIS	RO	0y0	許可ゲート後の受信割り込みフラグ: 0y0: 割り込み無し 0y1: 割り込み有り
[1]	RTMIS	RO	0y0	許可ゲート後の受信タイムアウト割り込みフラグ: 0y0: 割り込み無し 0y1: 割り込み有り
[0]	RORMIS	RO	0y0	許可ゲート後の受信オーバーラン割り込みフラグ: 0y0: 割り込み無し 0y1: 割り込み有り

9. SSP0ICR (SSP0 Interrupt clear register)

Address = (0xF200_2000) + 0x0020

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	—	—	Undefined	Read as undefined. Write as zero.
[1]	RTIC	WO	Undefined	受信タイムアウト割り込みフラグをクリア: 0y0: 無効 0y1: クリア
[0]	RORIC	WO	Undefined	受信オーバーラン割り込みフラグをクリア: 0y0: 無効 0y1: クリア

10. SSP0DMACR (SSP0DMA 制御レジスタ)

Address = (0x4001_D000) + (0x0024)

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	—	—	Undefined	Read as undefined. Write as zero.
[1]	TXDMAE	R/W	0y0	送信 FIFO の DMA イネーブル : 0y0: Disable 0y1: Enable
[0]	RXDMAE	R/W	0y0	受信 FIFO の DMA イネーブル : 0y0: Disable 0y1: Enable

3.16 USB デバイスコントローラ

3.16.1 システム概要

- 1) Universal Serial Bus Specification Rev.2.0 に準拠
- 2) High-Speed / Full-Speed をサポート (Low-Speed は非対応)
- 3) Chirp 対応
- 4) USB プロトコル処理
- 5) SOF / USB_RESET / SUSPEND / RESUME の検出
- 6) パケット ID の生成およびチェック
- 7) データ同期ビット(DATA0/DATA1/DATA2/MDATA)の生成およびチェック
- 8) CRC5 チェック、CRC16 の生成およびチェック
- 9) PING 対応
- 10) 4 種類(Control / Interrupt / Bulk / Isochronous)の転送モードをサポート
- 11) 4EndPoint のサポート

EndPoint 0:	コントロール	64 バイト×1FIFO
EndPoint 1:	バルク (IN)	512 バイト×2FIFO
EndPoint 2:	バルク (OUT)	512 バイト×2FIFO
EndPoint 3:	インターラプト (IN)	64 バイト×1FIFO
- 12) デュアルパケットモード対応(EndPoint0 は除く)
- 13) 割り込みコントローラへ割り込み要因信号: INTS [21]

3.16.1.1 システム構成

USB デバイスコントローラ は、UDC2 と呼ばれるコア部分と、AHB バスと接続機能を有する UDC2AB のバスブリッジから構成されています。

本章では、まず回路機能の概要を説明し、UDC2AB、UDC2 の順序で説明します。

3.16.2章では、UDC2ABのバスブリッジの構成を 3.16.3章では、UDC2 関連の構成を説明します。

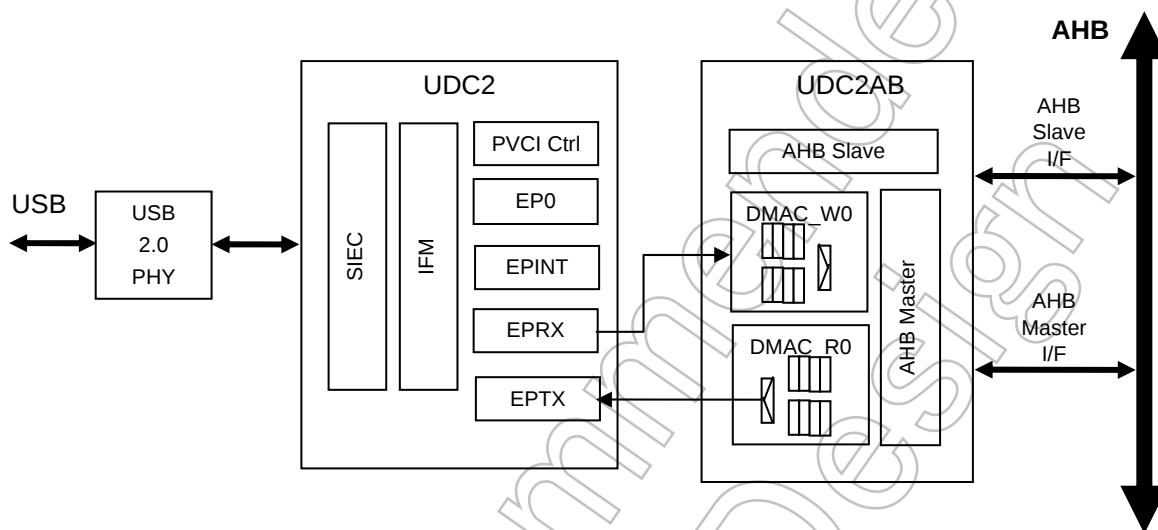
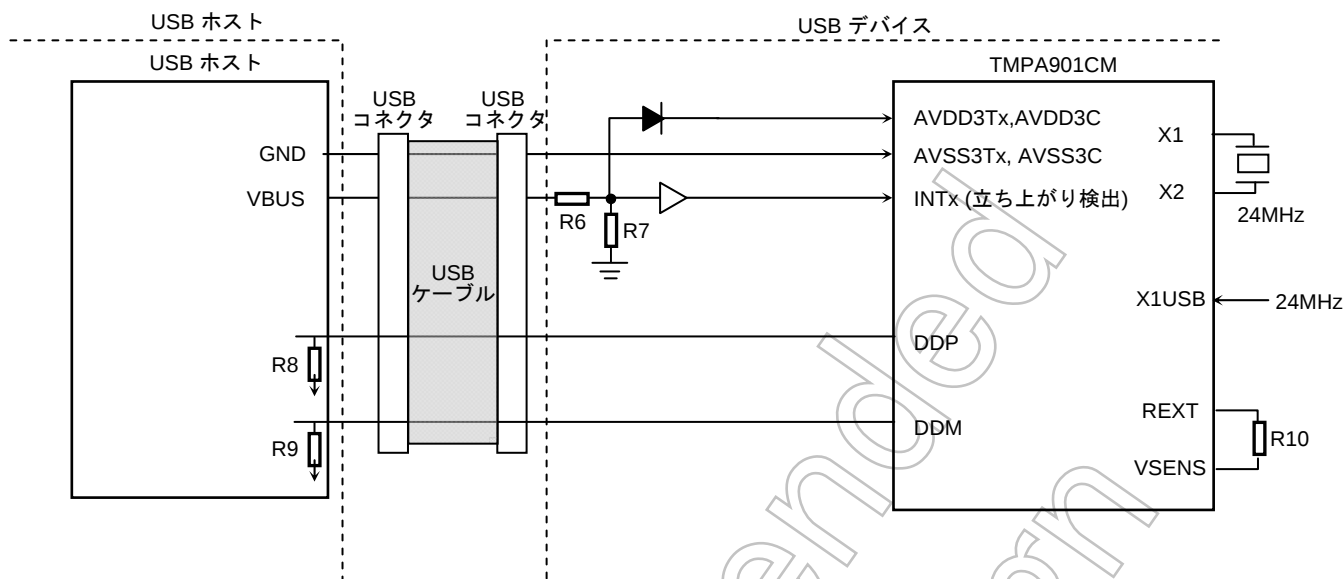


図 3.16.1 USB デバイスコントローラブロック図

3.16.1.2 接続例



TMPA901CM 内蔵の USB デバイスコントローラを使用する場合、以上のような接続が必要です。

(1) DDP 端子のプルアップ処理

USB 規定上、Full Speed の通信では DDP 端子をプルアップすることが必要ですが、プルアップ抵抗が内蔵されています。外付けの回路は不要です。

(2) DDP、DDM 端子の直列抵抗挿入

USB 規定上、DDP、DDM 端子には各々直列抵抗の挿入が必要ですが、直列抵抗が内蔵されています。外付けの回路は不要です。

(3) コネクタ接続検出処理

コネクタ接続の検出を VBUS(5V 電圧)で検出する方法を例として紹介します。

上記接続例のように VBUS 端子に抵抗分割用の R6、R7 を接続し、電源が接続された際に H レベル(3.3V)が割り込み端子へアサートされるようにします。この割り込みをソフトウェアで検出することによりコネクタ接続の検出が可能となります。

注) 波形の立ち上がりが緩慢な場合、波形整形用に適当なバッファリングをすることをお勧めします。

推奨値: R6=60k Ω 、R7=100k Ω (サスペンド時の VBUS 消費電流 < 500 μ A)

(4) 24MHz クロック入力

USB デバイスコントローラは 24MHz のクロックが必要です。そのために、2 通りの入力方法があります。ひとつは X1、X2 端子に 24MHz 発振子の接続、2 つ目は X1USB 端子より 24MHz クロックを入力できます。どちらを使用するかはクロックコントローラの SYSCRO<USBCCLKSEL>で選択します。どちらで使用する場合も、クロックは ± 100 ppm 以下の精度にしてください。

(5) ホスト側プルダウン抵抗

USB 規定上、USB ホスト側で DDP 端子と DDM 端子をプルダウンする必要があります。

推奨値: R8=15k Ω 、R9=15k Ω

(6) USB_PHY 用抵抗

REXT 端子と VSENS 端子間に抵抗を接続する必要があります。R10=12k Ω (±1.0%以内の誤差) を接続して下さい。

注) 上記の接続、および抵抗値などは一例であり、その動作を保証するものではありません。必ず最新の USB 規格の確認と、セット上での動作確認をお願いします。

Not Recommended
for New Design

3.16.2 UDC2AB AHBバスブリッジ

UDC2AB(UDC2 AHB Bridge)は、東芝 USB-Spec2.0 デバイスコントローラ(以下、UDC2)と AHB とのブリッジ回路です。UDC2AB は AHB マスタ転送をサポートする DMA コントローラを有し、AHB 上の指定されたアドレスと UDC2 内部の Endpoint-FIFO(Endpoint I/F)間の転送を制御します。

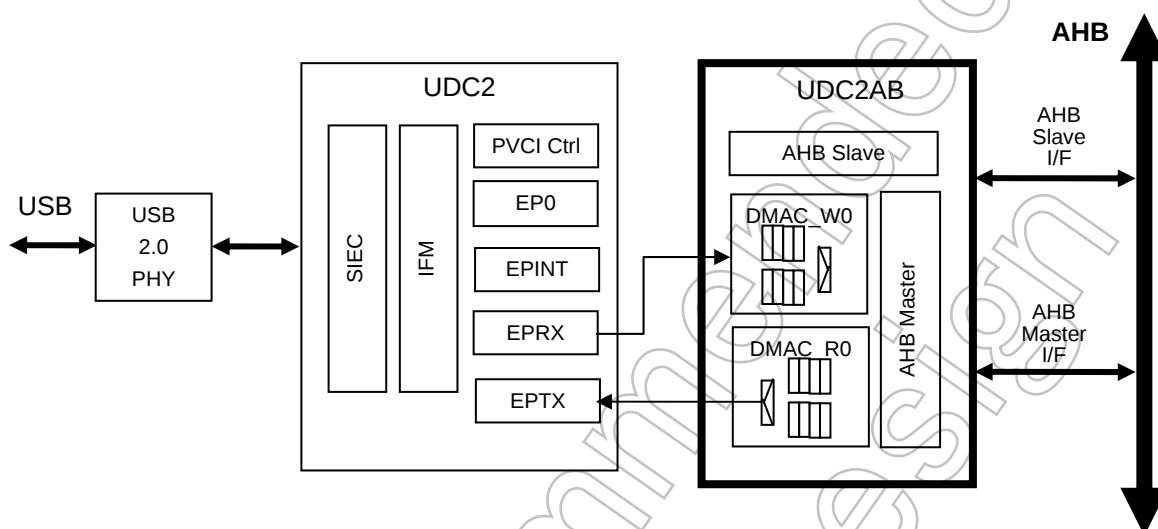


図 3.16.2 UDC2AB ブロック図

3.16.2.1 機能・特長

UDC2AB の機能・特長を以下に記します。

(1) UDC2 との接続

接続する UDC2 の Endpoint 構成に特に制限はありません。ただし、UDC2AB 内の DMA コントローラ(AHB マスタ機能)と接続可能なのは、Rx 用 EP が 1 個、Tx 用 EP が 1 個となります。その他の Endpoint(EP0 を含む)へのアクセスは、UDC2AB の AHB スレーブ機能を用いて、UDC2 の PVIC I/F と行います。なお、DMA コントローラでのマスタ転送中の UDC2 Endpoint の EPx_FIFO レジスタに対して、PVCI I/F 経由でアクセスはできません。

AHB マスタリード機能と接続する Endpoint の Max パケットサイズが奇数となる場合には使用方法に制限があります。詳細は「3.16.2.9(3) マスタリード転送時の Max パケットサイズ設定」を参照して下さい。

(2) AHB 機能

AHB マスタと AHB スレーブ機能を持ちます。

(a) AHB マスタ機能

AHB マスタ機能の仕様は以下となります。

- 2つのDMAチャンネルを持ち、Rx用EP、Tx用EPに対して、それぞれ、1チャンネルが割り当てられます。
- Single、Burst(INCR/INCR8) transactionをサポート
- Split transactionをサポートしません。
- Little Endianをサポートします。
- Protection Controlをサポートしません。
- Early Burst Terminationをサポートします
- アドレス幅32ビット、データ幅32ビット
- BYTE、WORDのtransaction sizeをサポートします。

なお、Endian変換のイメージは下図のようになります。

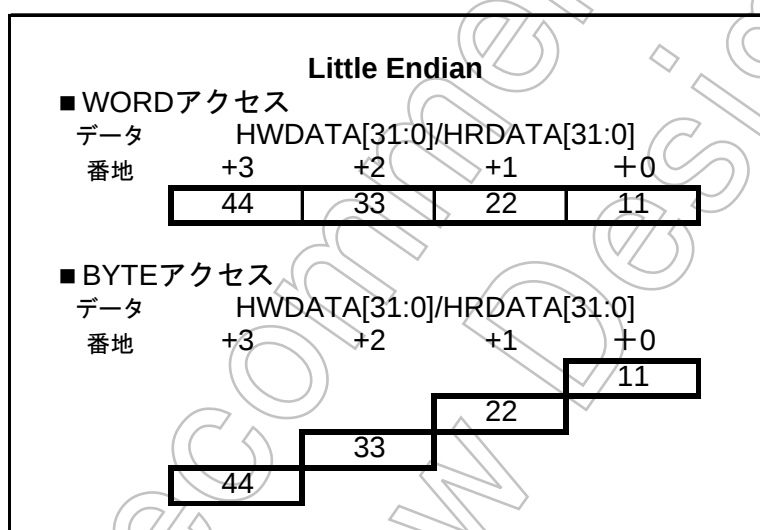


図 3.16.3 AHB マスタ機能 Endian 変換イメージ

(b) AHB スレーブ機能

AHB スレーブ機能の仕様は以下となります。

- 内部レジスタアクセスに使用します。
- Little Endian をサポートします。
- Single transaction のみサポート。
- アドレス幅 32 ビット、データ幅 32 ビット。
- BYTE、WORD の transaction size をサポートします。

なお、Endian 変換のイメージは下図のようになります。

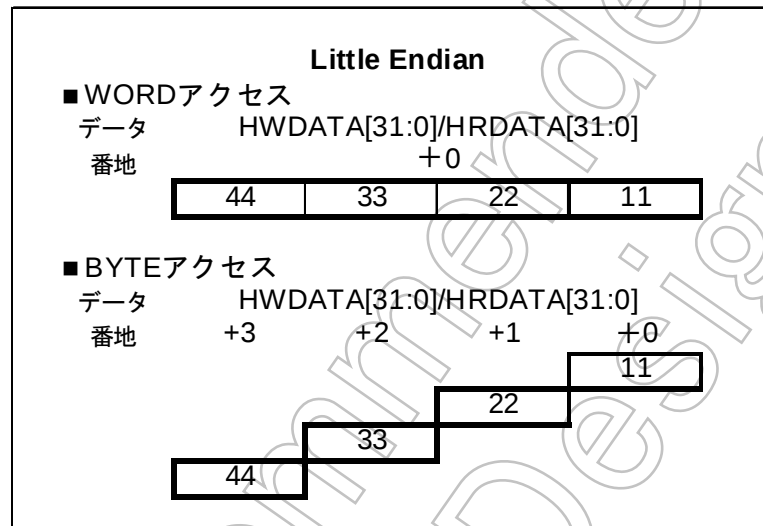


図 3.16.4 AHB スレーブ機能 Endian 変換イメージ

3.16.2.2 全体構成

UDC2AB は主に、UDC2AB 内部レジスタおよび UDC2 レジスタ(UDC2 PPCI I/F)とのアクセスを制御する AHB スレーブ機能と、UDC2 Endpoint I/F との DMA アクセスを制御する AHB マスタ機能から構成されます。

AHB マスタ機能には、マスタリードチャネル(AHB->UDC2)とマスタライトチャネル(UDC2->AHB)の計 2 チャネルが内蔵されており、UDC2 の Rx 用 EP、Tx 用 EP の Endpoint I/F との DMA 転送が可能です。各チャネルは、8Word のバッファを 2 つ(計 4 つ)内蔵しています。

3.16.2.3 クロックドメイン

CLK_U: 30MHz (USB 2.0 PHY から供給されます)

CLK_H: HCLK を表します。

3.16.2.4 用語と表記法

アサート	: 信号が有効な状態を示します。
デアサート	: 信号が無効な状態を示します。
Word(ワード)	: 32bit
Byte(バイト)	: 8bit
UDC2	: UDC2AB に接続される USB2.0 デバイスコントローラを示します。
UDC2AB	: 本 IP: UDC2-AHB-Bridge の略称
Endpoint	: UDC2 が持つ USB ホストとの通信用 FIFO。略称は“EP”
Rx 用 EP	: 受信用 Endpoint。USB 転送の OUT 転送(USB ホスト→USB デバイス)に対応する。
Tx 用 EP	: 送信用 Endpoint。USB 転送の IN 転送(USB ホスト←USB デバイス)に対応する。
Endpoint I/F	: UDC2 が持つ Endpoint 専用 DMA インターフェース。
PVCI I/F	: UDC2 が持つ共通インターフェース。UDC2 内部レジスタアクセスに使用。
マスタ転送	: UDC2AB がバス権を取得して転送することを示します。
ターゲットデバイス	: UDC2AB がマスタ転送でアクセスする相手先デバイス(メモリ等)を示します。
マスタライト転送	: UDC2AB が Rx 用 EP と行う転送を示します。
マスタリード転送	: UDC2AB が Tx 用 EP と行うマスタ転送を示します。
スレーブ転送	: UDC2AB 以外のデバイスが UDC2AB をターゲットとして行う転送を示します。
USB_RESET	: USB ホストから来るバスリセット。USB Spec 上では“Reset Signaling”
NULL パケット	: USB で転送される、0 Length データの事です。
PHY	: USB 2.0 PHY
割り込み	: INTS[21]出力信号の事を示します。本文中に「xx割り込みをアサート」といった表記をしている箇所は、Interrupt Enableレジスタの該当ビットをイネーブルとしている前提となっています。「3.16.2.7 割り込み信号(INTS[21])」を参照して下さい。

3.16.2.5 レジスタ

UDC2AB のレジスタマップは、UDC2AB 設定用のレジスタと、UDC2 設定用のレジスタで構成されます。UDC2 設定用のレジスタにアクセスした場合は、UDC2AB が UDC2 へ PVC I/F 経由で自動的にアクセスします。各レジスタは 32bit 幅です。

(1) レジスタマップ

以下に UDC2AB のレジスタマップを示します。

表 3.16.1 UDC2AB,UDC2 レジスタマップ(1/2)

base アドレス= 0xF440_0000

	Register Name	Address (base+)	Description
UDC2AB Bridge	UDINTSTS	0x0000	Interrupt Status Register
	UDINTENB	0x0004	Interrupt Enable Register
	UDMWTOU	0x0008	Master Write Timeout Register
	UDC2STSET	0x000C	UDC2 Setting Register
	UDMSTSET	0x0010	DMAC Setting Register
	DMACRDREQ	0x0014	DMAC Read Request Register
	DMACRDVL	0x0018	DMAC Read Value Register
	UDC2RDREQ	0x001C	UDC2 Read Request Register
	UDC2RDVL	0x0020	UDC2 Read Value Register
	Reserved	0x0024 ~ 0x0038 *4)	
	ARBTSET	0x003C	Arbiter Setting Register
	UDMWSADR	0x0040	Master Write Start Address Register
	UDMWEADR	0x0044	Master Write End Address Register
	UDMWCADR	0x0048 *1)	Master Write Current Address Register
	UDMWAHBADR	0x004C	Master Write AHB Address Register
	UDMRSADR	0x0050	Master Read Start Address Register
	UDMREADR	0x0054	Master Read End Address Register
	UDMRCADR	0x0058 *1)	Master Read Current Address Register
	UDMRAHBADR	0x005C	Master Read AHB Address Register
	Reserved	0x0060 ~ 0x007C	
	UDPWCTL	0x0080	Power Detect Control Register
	UDMSTSTS	0x0084	Master Status Register
	UDTOUTCNT	0x0088 *1)	Timeout Count Register
	Reserved	0x008C ~ 0x1FC *4)	

UDC2AB,UDC2 レジスタマップ (2/2)

base アドレス= 0xF440_0000

	Register Name	Address (base+)	Description
UDC2 *2), *3)	UD2ADR	0x0200	UDC2 Address-State Register
	UD2FRM	0x0204	UDC2 Frame Register
	UD2TMD	0x0208	UDC2 USB-Testmode Register
	UD2CMD	0x020C	UDC2 Command Register
	UD2BRQ	0x0210	UDC2 bRequest-bmRequestType Register
	UD2WVL	0x0214	UDC2 wValue Register
	UD2WIDX	0x0218	UDC2 wIndex Register
	UD2WLGTH	0x021C	UDC2 wLength Register
	UD2INT	0x0220	UDC2 INT Register
	UD2INTEP	0x0224	UDC2 INT_EP Register
	UD2INTEPMSK	0x0228	UDC2 INT_EP_MASK Register
	UD2INTRX0	0x022C	UDC2 INT_RX_DATA0 Register
	UD2EP0MSZ	0x0230	UDC2 EP0_MaxPacketSize Register
	UD2EP0STS	0x0234	UDC2 EP0_Status Register
	UD2EP0DSZ	0x0238	UDC2 EP0_Datasize Register
	UD2EP0FIFO	0x023C	UDC2 EP0_FIFO Register
	UD2EP1MSZ	0x0240	UDC2 EP1_MaxPacketSize Register
	UD2EP1STS	0x0244	UDC2 EP1_Status Register
	UD2EP1DSZ	0x0248	UDC2 EP1_Datasize Register
	UD2EP1FIFO	0x024C	UDC2 EP1_FIFO Register
	UD2EP2MSZ	0x0250	UDC2 EP2_MaxPacketSize Register
	UD2EP2STS	0x0254	UDC2 EP2_Status Register
	UD2EP2DSZ	0x0258	UDC2 EP2_Datasize Register
	UD2EP2FIFO	0x025C	UDC2 EP2_FIFO Register
	UD2EP3MSZ	0x0260	UDC2 EP3_MaxPacketSize Register
	UD2EP3STS	0x0264	UDC2 EP3_Status Register
	UD2EP3DSZ	0x0268	UDC2 EP3_Datasize Register
	UD2EP3FIFO	0x026C	UDC2 EP3_FIFO Register
	Reserved	0x0270~ 0x032C	
	UD2INTNAK	0x0330	UDC2 INT_NAK Register
	UD2INTNAKMSK	0x0334	UDC2 INT_NAK_MASK Register
	Reserved	0x0338 ~ 0x03FC	

*1) 必ず DMAC Read Request レジスタを経由してリードアクセスを行って下さい。

*2) 必ず UDC2 Read Request レジスタを経由してリードアクセスを行って下さい。

*3) +0x200~+0x3FC には UDC2 のレジスタがアサインされていますが、接続する UDC2 で未サポートとなっている Endpoint のレジスタや、Reserved のレジスタにはアクセスしないで下さい。

*4) 上記で Reserved と記されている領域と 0x400 ~ 0xFFFF の領域はアクセス禁止です。リードもライトもしないでください。

(2) 各レジスタ説明

次項より、UDC2AB 内各レジスタの詳細説明を行います。
各ビットの説明の意味は以下のようになります。

(例)

Address =(0xF440_0000)+ (0xxxxx)

Bit	Bit Symbol (注 1)	Type (注 2)	Reset Value (注 3)	Description
[31:30]	—	—	Undefined	Read as undefined. Write as zero.
[29]	mw_error_en	R/W	0y0,(-)	
[28]	power_detect_en	R/W	0y0,(-)	
[27:26]	—	—	Undefined	Read as undefined. Write as zero.
[25]	dmac_reg_rd_en	R/W	0y0,(-)	
[24]	udc2_reg_rd_en	R/W	0y0,(-)	

注 1) Bit Symbol

そのビットの名称です。

“—”となっているのは Reserved ビットとなり、ライトは出来ません。リード時は 0 が読み出されます。

注 2) レジスタ属性の説明

RO : リードオンリー。ライトは無視されます。

WO : ライトオンリー。リード時は 0 となります。

R/W : リード/ライト。

R/W1C : リード/ライト 1 クリア。これらのビットはリードとライトができます。

1 をライトすると対応するビットが 0 にクリアされ、0 をライトすると無効です。

R/W1S : リード/ライト 1 セット。リードとライトができます。

1 をライトすると対応するビットがセットされます。0 をライトすると無効です。

注 3) 初期値

そのビットのリセット後初期値 (1 or 0) です。ハードウェアリセットとソフトウェアリセット (UDPWCTL<pw_resetb>)で、初期値は同じとなります。

但し、ソフトウェアリセットではリセットされないビットに関しては、“(-)”を記しています。

1. UDINTSTS (Interrupt Status register)

本レジスタは、割り込み要因が発生すると各対応ビットに 1 がセットされます。Bit[29:8]へ 1 を書き込むことによってステータスをクリアすることが出来ます。Bit[7:0]は UDC2 の出力端子に対応しており、リードオンリーとなります。UDC2 の INT レジスタの該当ビットに 1 を書き込むことでクリアされます。

注) 割り込み信号の動作に関しては「3.16.2.7 割り込み信号(INTS[21])」をご覧ください。

Address =(0xF440_0000)+ (0x0000)

Bit	Bit Symbol	Type	Reset Value	Description
[31:30]	—	—	Undefined	Read as undefined. Write as zero.
[29]	int_mw_rerror	R/W1C	0y0	マスタライト Endpoint リードエラー 0y0: 未検出 0y1: マスタライト Endpoint リードエラー発生
[28:26]	—	—	Undefined	Read as undefined. Write as zero.
[25]	int_dmac_reg_rd	R/W1C	0y0	DMAC レジスタアクセス完了 0y0: 未検出 0y1: レジスタリード完了
[24]	int_udc2_reg_rd	R/W1C	0y0	UDC2 レジスタアクセス完了 0y0: 未検出 0y1: レジスタリード/ライト完了
[23]	int_mr_ahberr	R/W1C	0y0	マスタリード転送エラーステータス 0y0: 未検出 0y1: AHB エラー発生
[22]	int_mr_ep_dset	R/W1C	0y0	マスタリード Endpoint データセットステータス 0y0: FIFO ライト不可 0y1: FIFO ライト可
[21]	int_mr_end_add	R/W1C	0y0	マスタリード転送終了ステータス 0y0: 未検出 0y1: マスタリード転送終了
[20]	int_mw_ahberr	R/W1C	0y0	マスタライト転送エラーステータス 0y0: 未検出 0y1: AHB エラー発生
[19]	int_mw_timeout	R/W1C	0y0	マスタライト転送タイムアウトステータス 0y0: 未検出 0y1: マスタライト転送タイムアウト
[18]	int_mw_end_add	R/W1C	0y0	マスタライト転送終了ステータス 0y0: 未検出 0y1: マスタライト転送終了
[17]	int_mw_set_add	R/W1C	0y0	マスタライト転送アドレス要求ステータス 0y0: 未検出 0y1: マスタライト転送アドレス要求
[16:11]	—	—	Undefined	Read as undefined. Write as zero.
[10]	int_usb_reset_end	R/W1C	0y0	USB_RESET END 0y0: このビットがクリアされてから UDC2 は usb_reset 信号をデアサートしていません。 0y1: UDC2 が usb_reset 信号をデアサートしたことを示します。
[9]	int_usb_reset	R/W1C	0y0	USB_RESET 0y0: このビットがクリアされてから UDC2 は usb_reset 信号をアサートしていません。 0y1: UDC2 が usb_reset 信号をアサートしたことを示します。
[8]	int_suspend_resume	R/W1C	0y0	サスペンド/レジャーム割り込みステータス 0y0: 変化無し 0y1: ステータス変化

Bit	Bit Symbol	Type	Reset Value	Description
[7]	int_nak	RO	0y0	UDC2_INT_NAK レジスタ
[6]	int_ep	RO	0y0	UDC2_INT_EP レジスタ
[5]	int_ep0	RO	0y0	UDC2_INT_EP0 レジスタ
[4]	int_sof	RO	0y0	UDC2_INT_SOF レジスタ
[3]	int_rx_zero	RO	0y0	UDC2_INT_RXDATA0 レジスタ
[2]	int_status	RO	0y0	UDC2_INT_STATUS レジスタ
[1]	int_status_nak	RO	0y0	UDC2_INT_STATUS_NAK レジスタ
[0]	int_setup	RO	0y0	UDC2_INT_STATUS レジスタ

(個別説明)

a. <int_mw_error>

Endpoint へのアクセスが共通バスアクセスの設定時(EPx_Status レジスタの bus_sel bit が 0)にマスタライト転送を起動した場合に、1 にセットされます。

0y0: 未検出

0y1: マスタライト Endpoint リードエラー発生

b. <int_dmac_reg_rd>

DMAC Read Requeset レジスタの設定により実行されたレジスタアクセスが完了して、DMAC Read Value レジスタに読み出した値がセットされたときに、1 にセットされます。

0y0: 未検出

0y1: レジスタリード完了

c. <int_udc2_reg_rd>

UDC2 Read Request レジスタの設定により実行された UDC2 へのアクセスが完了して、UDC2 Read Value レジスタに読み出した値がセットされたときに、1 にセットされます。

また、UDC2 内部レジスタへのライトアクセスが完了したときに、1 にセットされます。

0y0: 未検出

0y1: レジスタリード/ライト完了

d. <int_mr_ahberr>

マスタリード転送動作中に、AHB エラーが発生した場合、本ステータスが 1 にセットされます。

この割り込み発生後は、UDMSTSET<mr_reset>によりマスタリード転送ブロックをリセットする必要があります。

0y0: 未検出

0y1: AHB エラー発生

- e. <int_mr_ep_dset>
マスタリード転送で使用する、UDC2 Tx 用 EP の FIFO がライト可能(Full ではない状態)となった時に、1 にセットされます。
0y0: FIFO ライト不可
0y1: FIFO ライト可
- f. <int_mr_end_add>
マスタリード転送が終了した際に、1 にセットされます。
0y0: 未検出
0y1: マスタリード転送終了
- g. <int_mw_ahberr>
マスタライト転送動作中に、AHB エラーが発生した場合、本ステータスが 1 にセットされます。
この割り込み発生後は、UDMSTSET<mw_reset>によりマスタライト転送ブロックをリセットする必要があります。
0y0: 未検出
0y1: AHB エラー発生
- h. <int_mw_timeout>
マスタライト転送動作中に、タイムアウトした場合、本ステータスが 1 にセットされます。
0y0: 未検出
0y1: マスタライト転送タイムアウト
- i. <int_mw_end_add>
マスタライト転送が終了した際に、1 にセットされます。
0y0: 未検出
0y1: マスタライト転送終了
- j. <int_mw_set_add>
マスタライト転送がディセーブル状態で、該当する Rx 用 EP にマスタライト転送されるべきデータがセットされると 1 にセットされます。
0y0: 未検出
0y1: マスタライト転送アドレス要求
- k. <int_usb_reset_end>
UDC2 が usb_reset 信号をデアサートしたかどうかを示します。
UDC2 が UDC2 レジスタを USB_RESET 後初期値に設定するタイミングは usb_reset 信号のデアサート時となります。このタイミングを検知したい場合は、本ビットを使用して下さい。なお、usb_reset 信号の状態は UDPWCTL<usb_reset>にて確認できます。
0y0: このビットがクリアされてから UDC2 は usb_reset 信号をデアサートしていません。
0y1: UDC2 が usb_reset 信号をデアサートしたことを示します。

l. <int_usb_reset>

UDC2 が usb_reset 信号をアサートしたかどうかを示します。なお、usb_reset 信号の状態は UDPWCTL <usb_reset>にて確認できます。

0y0: このビットがクリアされてから UDC2 は usb_reset 信号をアサートしていません。

0y1: UDC2 が usb_reset 信号をアサートしたことを示します。

m. <int_suspend_resume>

UDC2 の suspend_x 信号が変化するたびに 1 をアサートします。UDPWCTL<suspend_x>により状態を確認して下さい。

0y0: 変化無し

0y1: ステータス変化

n. <int_nak>

UDC2 の int_nak 信号が直接読み出せます。クリアするには UDC2 の INT レジスタもしくは INT_NAK レジスタの該当ビットをクリアして下さい。

o. <int_ep>

UDC2 の int_ep 信号が直接読み出せます。クリアするには UDC2 の INT レジスタもしくは INT_EP レジスタの該当ビットをクリアして下さい。

p. <int_ep0>

UDC2 の int_ep0 信号が直接読み出せます。クリアするには UDC2 の INT レジスタの該当ビットをクリアして下さい。

q. <int_sof>

UDC2 の int_sof 信号が直接読み出せます。クリアするには UDC2 の INT レジスタの該当ビットをクリアして下さい。

r. <int_rx_zero>

UDC2 の int_rx_zero 信号が直接読み出せます。クリアするには UDC2 の INT レジスタもしくは INT_RX_ZERO レジスタの該当ビットをクリアして下さい。

s. <int_status>

UDC2 の int_status 信号が直接読み出せます。クリアするには UDC2 の INT レジスタの該当ビットをクリアして下さい。

t. <int_status_nak>

UDC2 の int_status_nak 信号が直接読み出せます。クリアするには UDC2 の INT レジスタの該当ビットをクリアして下さい。

u. <int_setup>

UDC2 の int_setup 信号が直接読み出せます。クリアするには UDC2 の INT レジスタの該当ビットをクリアして下さい。

UDC2 出力信号と本レジスタ bit[9]、bit[7:0]の接続関係を、以下の図に示します。

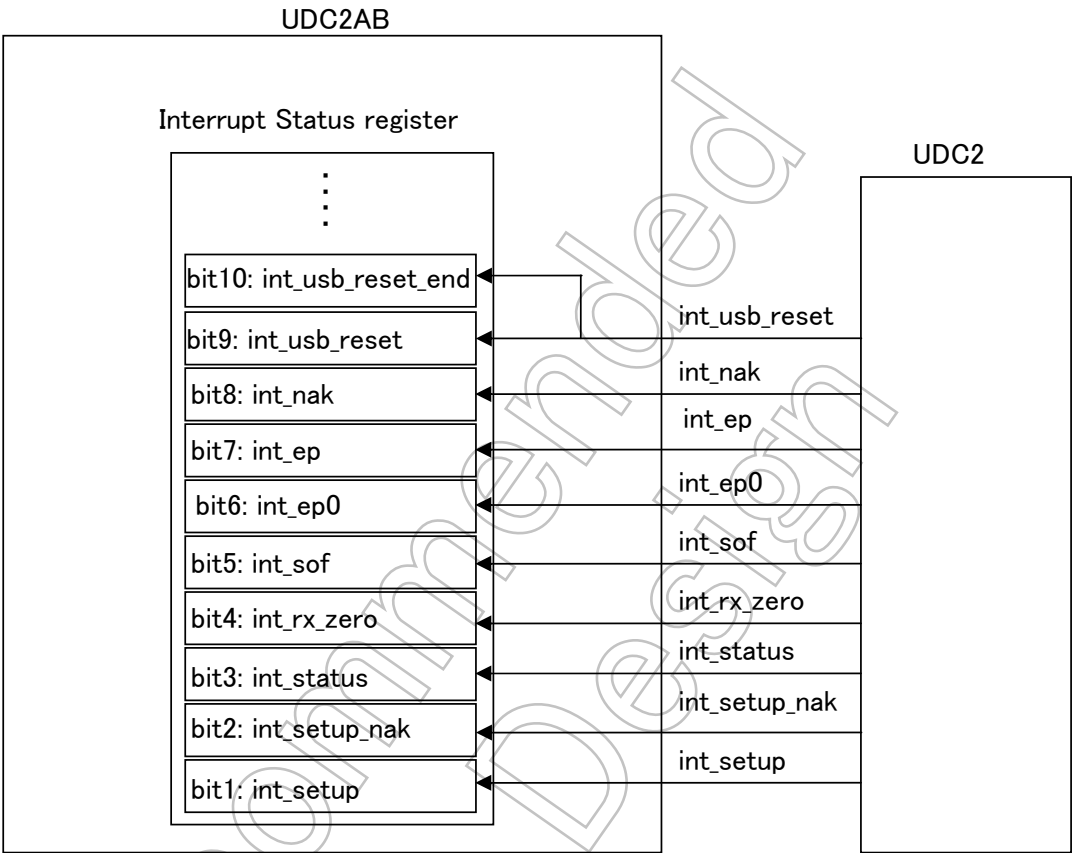


図 3.16.5 フラグ出力信号と割り込みビットの接続関係

2. UDINTENB (Interrupt Enable register)

本レジスタの対応ビットに 0 を書き込むことにより、割り込み信号: INTS[21]出力信号の、対応する割り込み要因をディセーブルすることが出来ます。また、1 を書き込むことにより対応する割り込み要因が有効となります。

各ビットのイネーブル/ディセーブルの状態に関わらず Interrupt Status レジスタの該当ビットはセットされますので、本レジスタのイネーブルと同時に割り込みが発生する可能性があります。その動作を意図しない場合、先に Interrupt Status レジスタの対応するビットをクリアする必要があります。

Interrupt Status レジスタ bit[7:0] に対する割り込み制御用のレジスタは、本レジスタではなく、UDC2 の INT レジスタの bit[15:8] となります。UDC2 の章をご覧ください。

注) 割り込み信号の動作に関しては「3.16.2.7 割り込み信号(INTS[21])」をご覧ください。

Address =(0xF440_0000)+(0x0004)

Bit	Bit Symbol	Type	Reset Value	Description
[31:30]	—	—	Undefined	Read as undefined. Write as zero.
[29]	mw_rerror_en	R/W	0y0,(-)	マスタライト Endpoint リードエラー 0y0: ディセーブル 0y1: イネーブル
[28:26]	—	—	Undefined	Read as undefined. Write as zero.
[25]	dmac_reg_rd_en	R/W	0y0,(-)	DMAC レジスタリード完了 0y0: ディセーブル 0y1: イネーブル
[24]	udc2_reg_rd_en	R/W	0y0,(-)	UDC2 レジスタリードアクセス完了 0y0: ディセーブル 0y1: イネーブル
[23]	mr_ahberr_en	R/W	0y0,(-)	マスタリード転送エラーステータス要因イネーブル 0y0: ディセーブル 0y1: イネーブル
[22]	mr_ep_dset_en	R/W	0y0,(-)	マスタリード Endpoint データセットステータス要因イネーブル 0y0: ディセーブル 0y1: イネーブル
[21]	mr_end_add_en	R/W	0y0,(-)	マスタリード転送終了ステータス要因イネーブル 0y1: イネーブル 0y0: ディセーブル
[20]	mw_ahberr_en	R/W	0y0,(-)	マスタライト転送エラーステータス要因イネーブル 0y0: ディセーブル 0y1: イネーブル
[19]	mw_timeout_en	R/W	0y0,(-)	マスタライト転送タイムアウトステータス要因イネーブル 0y0: ディセーブル 0y1: イネーブル
[18]	mw_end_add_en	R/W	0y0,(-)	マスタライト転送終了ステータス要因イネーブル 0y0: ディセーブル 0y1: イネーブル
[17]	mw_set_add_en	R/W	0y0,(-)	マスタライト転送アドレス要求ステータス要因イネーブル 0y0: ディセーブル 0y1: イネーブル

Bit	Bit Symbol	Type	Reset Value	Description
[16:11]	–	–	Undefined	Read as undefined. Write as zero.
[10]	usb_reset_end_en	R/W	0y0,(-)	USB_RESET END 割り込み要因イネーブル 0y0: ディセーブル 0y1: イネーブル
[9]	usb_reset_en	R/W	0y0,(-)	USB_RESET 割り込み要因イネーブル 0y0: ディセーブル 0y1: イネーブル
[8]	suspend_resume_en	R/W	0y0,(-)	サスペンド/レジューム割り込み要因イネーブル 0y0: ディセーブル 0y1: イネーブル
[7:0]	–	–	Undefined	Read as undefined. Write as zero.

(個別説明)

- a. <mw_rerror_en>
int_mw_rerror 割り込みを制御します。
0y0: ディセーブル
0y1: イネーブル
- b. <dmac_reg_rd_en>
int_dmac_reg_rd 割り込みを制御します。
0y0: ディセーブル
0y1: イネーブル
- c. <udc2_reg_rd_en>
int_udc2_reg_rd 割り込みを制御します。
0y0: ディセーブル
0y1: イネーブル
- d. <mr_ahberr_en>
int_mr_ahberr 割り込みを制御します。
0y0: ディセーブル
0y1: イネーブル
- e. <mr_ep_dset_en>
int_mr_ep_dset 割り込みを制御します。
0y0: ディセーブル
0y1: イネーブル
- f. <mr_end_add_en>
int_mr_end_add 割り込みを制御します。
0y0: ディセーブル
0y1: イネーブル

- g. <mw_ahberr_en>
int_mw_ahberr 割り込みを制御します。
0y0: ディセーブル
0y1: イネーブル
- h. <mw_timeout_en>
int_mw_timeout 割り込みを制御します。
0y0: ディセーブル
0y1: イネーブル
- i. <mw_end_add_en>
int_mw_end_add 割り込みを制御します。
0y0: ディセーブル
0y1: イネーブル
- j. <mw_set_add_en>
int_mw_set_add 割り込みを制御します。
0y0: ディセーブル
0y1: イネーブル
- k. <usb_reset_end_en>
int_usb_reset_end 割り込みを制御します。
0y0: ディセーブル
0y1: イネーブル
- l. <usb_reset_en>
int_usb_reset 割り込みを制御します。
0y0: ディセーブル
0y1: イネーブル
- m. <suspend_resume_en>
int_suspend_resume 割り込みを制御します。
0y0: ディセーブル
0y1: イネーブル

3. UDMWTOUT (Master Write Timeout register)

マスタライト動作中のタイムアウトを制御するためのレジスタです。

Address =(0xF440_0000)+ (0x0008)

Bit	Bit Symbol	Type	Reset Value	Description
[31:1]	timeoutset	R/W	0x7FFFFFFF	マスタライト・タイムアウト用タイマ設定レジスタ
[0]	timeout_en	R/W	0y1	マスタライト・タイムアウト Enable 設定レジスタ 0y0: Disable 0y1: Enable

(個別説明)

a. <timeoutset>

マスタライト・タイムアウト用のタイマ設定レジスタです。

マスタライト転送中に設定を変更しないで下さい。マスタライト(Rx)用 Endpoint のデータが無くなって(eprx_dataset = 0)から、CLK_U を設定した回数カウントするとタイムアウトとなります。

タイムアウト用カウンタは 32bit で構成され、本レジスタ timeoutset[31:1]はそのカウンタの 32bit 中の上位 31bit を設定でき、カウンタの最下位 bit は 1 として設定されます。

CLK_U が 30MHz の場合は、約 33[ns]~143[s]がタイムアウト値として設定可能です。

また、CLK_U が停止中(PHY をサスペンド中など)は、カウンタの動作が止まるため、タイムアウト割込みは発生しません。

b. <timeout_en>

マスタライト・タイムアウトの Enable 設定レジスタです。

マスタライト・タイムアウトをイネーブルする際に使用します。初期状態はイネーブルです。マスタライト転送中に設定を変更しないで下さい。

0y0: Disable

0y1: Enable

4. UDC2STSET (UDC2 Setting register)

UDC2 の転送動作を制御するためのレジスタです。

Address =(0xF440_0000)+ (0x000C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:5]	–	–	Undefined	Read as undefined. Write as zero.
[4]	eopb_enable	R/W	0y1	マスタリード EOP イネーブルレジスタ 0y0: マスタリード EOP ディセーブル 0y1: マスタリード EOP イネーブル
[3:1]	–	–	Undefined	Read as undefined. Write as zero.
[0]	tx0	R/W1S	0y0	NULL パケット送信レジスタ 0y0: ノーオペレーション 0y1: NULL パケットを送信します。

(個別説明)

a. <eopb_enable>

マスタリード EOP をイネーブルする際に使用します。初期状態はイネーブルです。マスタリード転送中に設定を変更しないで下さい。

本ビットが 0 の場合、最終 Word が 1Byte の場合には UDC2 への最終データ転送は行なわれません。最終 Word が 2byte の場合には ep_x_w_eop=0 での UDC2 への最終データ転送が行なわれます。

本ビットが 1 の場合、最終 Word の Byte 数に関わらず ep_x_w_eop=1 での UDC2 への最終データ転送が行なわれます。

注)「3.16.2.9(1) マスタリード転送」を参照して下さい。

0y0: マスタリード EOP ディセーブル

0y1: マスタリード EOP イネーブル

b. <tx0>

マスタリード動作側に接続されている Endpoint で、NULL パケットを送信する際に使用します。UDMSTSTS<mrepempty>が 1 の場合のみ有効で、それ以外では本ビットは無視されます。ライト後に自動的に 0 にクリアされます。

本ビットに 1 をセットすることで UDC2 Endpoint-I/F の ep_x_tx0data 信号がアサートされ、NULL パケット送信実行中は、1 の値を保持しています。本ビットセット後は、0 にクリアされるまで Tx 用 EP への次のデータセットは行わないで下さい。

0y0: ノーオペレーション

0y1: NULL パケットを送信します。

5. UDMSTSET (DMAC Setting register)

内蔵 DMAC の転送を制御するためのレジスタです。

Address =(0xF440_0000)+ (0x0010)

Bit	Bit Symbol	Type	Reset Value	Description
[31:9]	–	–	Undefined	Read as undefined. Write as zero.
[8]	m_burst_type	R/W	0y0,(-)	マスタバーストタイプ選択レジスタ 0y0: INCR8 (HBURST=5h) 0y1: INCR (HBURST=1h)
[7]	–	–	Undefined	Read as undefined. Write as zero.
[6]	mr_reset	R/W1S	0y0	マスタリードリセットレジスタ 0y0: ノーオペレーション 0y1: リセット
[5]	mr_abort	WO	0y0	マスタリードアボートレジスタ 0y0: ノーオペレーション 0y1: アボート
[4]	mr_enable	R/W1S	0y0	マスタリードイネーブルレジスタ 0y0: ディセーブル 0y1: イネーブル
[3]	–	–	Undefined	Read as undefined. Write as zero.
[2]	mw_reset	R/W1S	0y0	マスタライトリセットレジスタ 0y0: ノーオペレーション 0y1: リセット
[1]	mw_abort	WO	0y0	マスタライトアボートレジスタ 0y0: ノーオペレーション 0y1: アボート
[0]	mw_enable	R/W1S	0y0	マスタライトイネーブルレジスタ 0y0: ディセーブル 0y1: イネーブル

(個別説明)

a. <m_burst_type>

マスタライト/リード転送時のバースト転送実行時の HBURST[2:0]のタイプを選択します。UDC2AB が行うバースト転送のタイプは INCR8(8 ビート インクリメント式バースト)となります。従って、通常は初期値である 0 を設定して下さい。但し、システムの AHB 仕様によりバースト転送のタイプとして INCR しか使用できない場合には、このビットに 1 を設定して下さい。この場合、UDC2AB は 8 ビート下の INCR 転送を実行します。なお、バースト転送のビット数を変更することはできません。

このビットの設定は UDC2AB への初期設定にて行って下さい。マスタライト/リード転送を開始してからは変更しないで下さい。

注) UDC2AB はマスタライト/リード転送でバースト転送のみを行うわけではなく、バースト転送とシングル転送を組み合わせで転送します。このビットはあくまでバースト転送実行時にのみ影響します。

0y0: INCR8

0y1: INCR

b. <mr_reset>

UDC2AB のマスタリード転送ブロックを初期化します。ただし Endpoint の FIFO は初期化されませんので、本リセットとは別に UDC2 の Command レジスタへアクセスして、対応する Endpoint の初期化を行う必要があります。

本リセットはマスタ動作を停止させてから使用して下さい。

本ビットを 1 へセット後、自動的に 0 にクリアされます。クリアされるまで次のマスタリード転送を行わないで下さい。

0y0: ノーオペレーション

0y1: リセット

c. <mr_abort>

マスタリード転送を制御します。本ビットに 1 をセットすることによりマスタリード動作を停止させることができます。

転送途中にアボートした場合、マスタリード用バッファの UDC2 への転送を中断し mr_enable ビットがクリアされ、マスタリード転送は停止されます。

本ビットを 1 へセット後、mr_enable ビットが 0 へディセーブルされるとアボート完了となります。

0y0: ノーオペレーション

0y1: アボート

d. <mr_enable>

マスタリード転送を制御します。転送アドレスのセット完了時にイネーブルにして下さい。マスタ転送の終了とともに、自動的にディセーブルされます。本レジスタではマスタリード動作のディセーブルを行うことはできませんので、マスタリード転送を停止させる際は mr_abort ビットを使用して下さい。

0y0: ディセーブル

0y1: イネーブル

e. <mw_reset>

マスタライト転送ブロックを初期化します。ただし Endpoint の FIFO は初期化されませんので、本リセットとは別に UDC2 の Command レジスタへアクセスして、対応する Endpoint の初期化を行う必要があります。

本リセットはマスタ動作を停止させてから使用して下さい。

本ビットを 1 へセット後、自動的に 0 にクリアされます。クリアされるまで次のマスタライト転送を行わないで下さい。

0y0: ノーオペレーション

0y1: リセット

f. <mw_abort>

マスタライト転送を制御します。本ビットに 1 をセットすることによりマスタライト動作を停止させることができます。

転送途中にアボートした場合、UDC2 からマスタライト用バッファへの転送を中断して mw_enable ビットがクリアされ、マスタライト転送は停止されます。

本ビットを 1 へセット後、mw_enable ビットが 0 へディセーブルされるとアボート完了となります。

0y0: ノーオペレーション

0y1: アボート

g. <mw_enable>

マスタライト転送を制御します。転送アドレスのセット完了時にイネーブルにしてください。マスタ転送の終了とともに、自動的にディセーブルされます。本レジスタではマスタライト動作のディセーブルを行うことはできませんのでマスタライト転送を停止させる際は、mw_abort ビットを使用してください。

0y0: ディセーブル

0y1: イネーブル

6. DMACRDREQ (DMAC Read Requet register)

- Master Read Current Address レジスタ
- Timeout Count レジスタ

をリードする場合に、リード要求を発行するためのレジスタです。読み出した値は、DMAC Read Value レジスタに保存されます。

注) UDC2 がサスペンド状態の時に、PHY からのクロック(=CLK_U)供給が停止している場合は、本レジスタアクセスは不可能となりますので、アクセスしないで下さい。なお、Power Detect Control レジスタ・phy_suspend ビットを 1 にセットしている時に、本レジスタにアクセスした場合、AHB エラー応答となります。

Address =(0xF440_0000)+ (0x0014)

Bit	Bit Symbol	Type	Reset Value	Description
[31]	dmardreq	R/W1S	0y0	レジスタリード要求&BUSY レジスタ 0y0: ノーオペレーション 0y1: リード要求発行
[30]	dmardclr	R/W1S	0y0	リード要求クリアレジスタ 0y0: ノーオペレーション 0y1: 強制クリア発行
[29:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:2]	dmardadr	R/W	0y000000	リード要求レジスタアドレス(上位 6bit)選択 0x48: Master Write Current Address レジスタリード時 0x58: Master Read Current Address レジスタリード時 0x88: Timeout Count レジスタリード時
[1:0]	–	–	Undefined	Read as undefined. Write as zero.

(個別説明)

a. <dmardreq>

特定の DMAC 関連のレジスタリードアクセス要求ビットです。本ビットを 1 にセットすることにより、dmardadr に設定されたアドレスにリードアクセスが実行されます。リードアクセスが完了して、DMAC Read Value レジスタに読み出した値がセットされると、本ビットは、自動的にクリアされ、UDINTSTS<int_dmac_reg_rd>が 1 にセットされます。

0y0: ノーオペレーション

0y1: リード要求発行

b. <dmardclr>

DMAC 関連のレジスタリードアクセス要求を強制的にクリアするビットです。本ビットを 1 にセットすることにより、dmardreq によるレジスタリード要求は強制的に終了し、dmardreq の値が 0 となります。強制クリア処理が完了すると、本ビットは、自動的にクリアされます。

0y0: ノーオペレーション

0y1: 強制クリア発行

c. <dmardadr>

リード要求を発行するレジスタのアドレス(上位 6 ビット)をセットします。上記の dmardreq ビットと共にセットして下さい。

以下に示す何れかのアドレスをセットして下さい。

0x48: Master Write Current Address レジスタリード時

0x58: Master Read Current Address レジスタリード時

0x88: Timeout Count レジスタリード時

Not Recommended
for New Design

7. DMACRDVL (DMAC Read Value register)

DMAC Read Request レジスタを通して読み出した値が格納されるレジスタです。

(対応レジスタ)

- Master Write Current Address レジスタ
- Master Read Current Address レジスタ
- Timeout Count レジスタ

$$\text{Address} = (0xF440_0000) + (0x0018)$$

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	dmardata	RO	0x00000000	レジスタリードデータ

(個別説明)

a. <dmardata>

本レジスタは、DMAC Read Request レジスタによって要求されたデータが格納されます。
DMAC Read Request レジスタの dmardreq ビットが 1 の時は、本レジスタにアクセスしないで下さい。

8. UDC2RDREQ (UDC2 Read Request register)

UDC2 のレジスタをリードする場合、リード要求を発行するためのレジスタです。読み出した値は、UDC2 Read Value レジスタに保存されます。

Address =(0xF440_0000)+ (0x001C)

Bit	Bit Symbol	Type	Reset Value	Description
[31]	udc2rdreq	R/W1S	0y0	レジスタリード要求&BUSY 設定レジスタ 0y0: ノーオペレーション 0y1: リード要求発行
[30]	udc2rdclr	R/W1S	0y0	リード要求クリアレジスタ 0y0: ノーオペレーション 0y1: 強制クリア発行
[29:10]	–	–	Undefined	Read as undefined. Write as zero.
[9:2]	udc2rdadr	R/W	0x00	リード要求 UDC2 レジスタアドレス設定レジスタ
[1:0]	–	–	Undefined	Read as undefined. Write as zero.

(個別説明)

a. <udc2rdreq>

UDC2 レジスタへのリードアクセス要求ビットです。本ビットを1にセットすることにより、udc2rdadr ビットにセットされたアドレスにリードアクセスが実行されます。リードアクセスが完了して、UDC2 Read Value レジスタに読み出した値がセットされると、本ビットは、自動的にクリアされ、UDINTSTS<int_udc2_reg_rd>が1にセットされます。

また、UDC2 のレジスタにライトアクセス実行中は、アクセス中を示すステータスビットとして機能して、1の値を示します。本ビットが1の間はUDC2 レジスタへの次のアクセスを実行しないで下さい。

0y0: ノーオペレーション

0y1: リード要求発行

b. <udc2rdclr>

UDC2 レジスタのリード/ライトアクセス要求を強制的にクリアするビットです。本ビットを1にセットすることにより、udc2rdreq によるレジスタリード要求/UDC2 ライトアクセスは強制的に終了し、udc2rdreq の値が0となります。強制クリア処理が完了すると、本ビットは、自動的に0にクリアされます。中断した場合、アクセス中のリード値、ライト値は保証されません。

0y0: ノーオペレーション

0y1: 強制クリア発行

c. <udc2rdadr>

リード要求を発行するUDC2 のレジスタアドレス[9:2]をセットします。UDC2 のレジスタアドレスは「表 3.16.2 レジスタマップ」を参照してください。本レジスタマップのオフセットアドレスである 0x0200~0x0334 が該当します。上記のudc2rdreqと共にセットして下さい。

9. UDC2RDVL (UDC2 Read Value register)

UDC2 Read Request レジスタを通して読み出した値が格納されるレジスタです。

Address =(0xF440_0000)+ (0x0020)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined.
[15:0]	udc2rdata	RO	0x0000	レジスタリードデータ

(個別説明)

a. <udc2rdata>

本レジスタは、UDC2 Read Request レジスタによって要求されたデータが格納されます。
UDC2RDREQ <udc2rdreq>が 1 の時は、本レジスタにアクセスしないで下さい。

10.ARBTSSET (Arbiter Setting register)

内蔵アービターが AHB にアクセスするときの優先度を設定するレジスタです。

本レジスタはマスタ動作を停止させてから設定変更を行って下さい。

アービトレーション方式の設定は必ず以下の手順で行って下さい。(計 3 回のアクセスが必要)

(1) abt_en ビットに 0 をライトしてアービトレーション回路を無効にする。

(2) abtmod ビット、abtpri_*ビットの設定を行なう。

(1) で abt_en ビットに 0 をライトしていない場合は、abtmod ビット、abtpri_*ビットへは設定できない。また、abtmod ビットの値に関わらず、優先順位を設定するレジスタの値は重複させない必要がある。

(3) (2)で設定した abtmod ビット、abtpri_*ビットを保持したまま、abt_en ビットに 1 をライトしてアービトレーション回路を有効にする。

Address =(0xF440_0000)+(0x003C)

Bit	Bit Symbol	Type	Reset Value	Description
[31]	abt_en	R/W	0y1	アービターイネーブルレジスタ 0y0: 無効(DMA アクセス不可) 0y1: 有効
[30:29]	–	–	Undefined	Read as undefined. Write as zero.
[28]	abtmod	R/W	0y0	アービターモードレジスタ 0y0: ラウンドロビン 0y1: 固定優先順位
[27:14]	–	–	Undefined	Read as undefined. Write as zero.
[13:12]	abtpri_w1	R/W	0y11	マスタライト 1 プライオリティ設定レジスタ 0y00 ~ 0y11
[11:10]	–	–	Undefined	Read as undefined. Write as zero.
[9:8]	abtpri_w0	R/W	0y10	マスタライト 0 プライオリティ設定レジスタ 0y00 ~ 0y11
[7:6]	–	–	Undefined	Read as undefined. Write as zero.
[5:4]	abtpri_r1	R/W	0y01	マスタリード 1 プライオリティ設定レジスタ 0y00 ~ 0y11
[3:2]	–	–	Undefined	Read as undefined. Write as zero.
[1:0]	abtpri_r0	R/W	0y00	マスタリード 0 プライオリティ設定レジスタ 0y00 ~ 0y11

(個別説明)

a. <abt_en>

DMAC-AHB 間のアクセスするとき、アービター動作を有効にします。

本ビットは、本レジスタの abtmod, abtpri_*の各ビットをセットする場合には、0 をセットする必要があります。また、abtpri_*の各設定値が重複している場合には、本ビットに 1 をセットすることができませんので、ご注意ください。

必ずこのビットを 1 にセットした状態で、DMA アクセスを開始して下さい。

0y0: 無効 (DMA アクセス不可)

0y1: 有効

b. <abtmod>

アービターの方式を設定します。abt_en ビットが 0 の時のみライトアクセス可能です。

本ビットに 0 を設定した場合は、abtpri_*の各ビット設定値に関わらず、ラウンドロビン方式で AHB バスへのアクセス権が与えられます。

本ビットに 1 を設定した場合は、abtpri_*の各ビット設定値によってアクセス優先順位に従って AHB バスへアクセス権が与えられます。

0y0: ラウンドロビン

0y1: 固定優先順位

c. <abtpri_w1>

固定優先順位のモードを選択したときのマスタライト 1 用 DMA アクセスの優先度を設定します。abt_en ビットが 0 の時のみライトアクセス可能です。

優先順位は、高 [0]– [3] 低。

d. <abtpri_w0>

固定優先順位のモードを選択したときのマスタライト 0 用 DMA アクセスの優先度を設定します。abt_en ビットが 0 の時のみライトアクセス可能です。

優先順位は、高 [0]– [3] 低。

e. <abtpri_r1>

固定優先順位のモードを選択したときのマスタリード 1 用 DMA アクセスの優先度を設定します。abt_en ビットが 0 の時のみライトアクセス可能です。

優先順位は、高 [0]– [3] 低。

f. <abtpri_r0>

固定優先順位のモードを選択したときのマスタリード 0 用 DMA アクセスの優先度を設定します。abt_en ビットが 0 の時のみライトアクセス可能です。

優先順位は、高 [0]– [3] 低。

- 注意点

abtpri_w1、abtpri_w0、abtpri_r1、abtpri_r0 の各ビットには、必ず異なった優先度の値を設定して下さい。もし、同じ優先度の値が設定された場合には、abt_en に、1 をセットすることが出来ません。

<DMAC と Arbiter Setting レジスタのプライオリティ領域との関係>

現 UDC2AB 仕様ではマスタライト用 DMAC(DMAC_W0)が 1 個、マスタリード用 DMAC(DMAC_R0)が 1 個、をサポートしています。各 2 個目にあたるマスタライト用 DMAC(DMAC_W1)、マスタリード用 DMAC(DMAC_R1)はサポートしていません。従って、DMAC_W1、DMAC_R1 への優先度の設定は実質的には意味がありませんが、前述の通り、abtpri_w1、abtpri_w0、abtpri_r1、abtpri_r0 の各ビットには、必ず、異なった優先度の値を設定して下さい。未実装の DMAC の対するレジスタ領域に、値を設定しても問題はありません。Arbiter Setting レジスタのプライオリティ領域は、以下のように DMAC と対応しています。

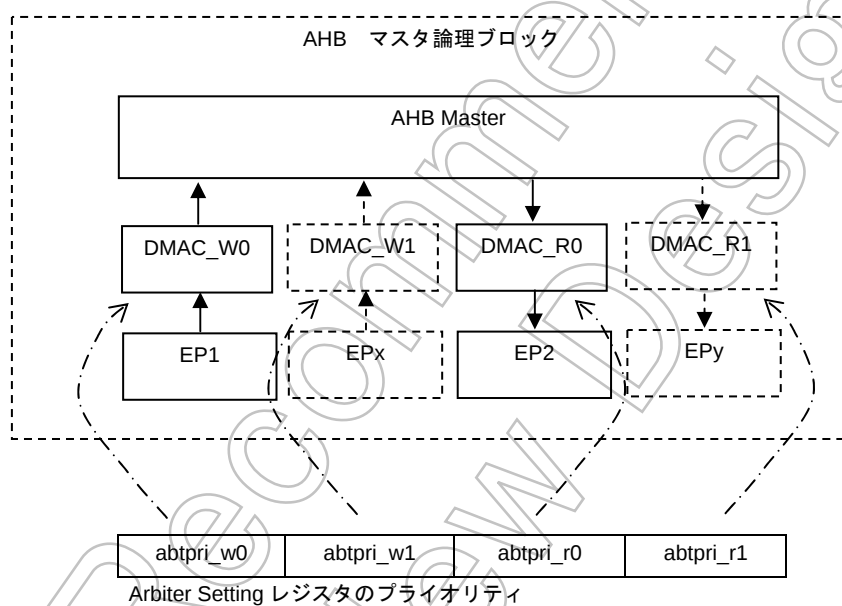


図 3.16.6 DMAC とプライオリティ領域の関係

11.UDMWSADR (Master Write Start Address register)

マスタライト転送(UDC2→AHB)のスタートアドレスを設定します。

Address =(0xF440_0000)+ (0x0040)

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	mwsadr	R/W	0xFFFFFFFF	マスタライトスタートアドレス設定レジスタ

(個別説明)

a. <mwsadr>

マスタライト転送のスタートアドレスを設定して下さい。ただし、本マスタ動作はアドレス増加にのみ対応していますので、Master Write End Address レジスタよりも下位の値を設定して下さい。

12.UDMWEADR (Master Write End Address register)

マスタライト転送(UDC2→AHB)のエンドアドレスを設定します。

Address =(0xF440_0000)+ (0x0044)

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	mweadr	R/W	0xFFFFFFFF	マスタライトエンドアドレス設定レジスタ

(個別説明)

a. <mweadr>

マスタライト転送のエンドアドレスを設定して下さい。ただし、本マスタはアドレス増加にのみ対応していますので、Master Write Start Address レジスタよりも上位の値を設定して下さい。

13.UDMWCADR (Master Write Current Address register)

マスタライト転送(UDC2→AHB)において、Endpoint からマスタライト用バッファへの現在までの、転送完了アドレスを表示します。

本レジスタは、直接アドレスを指定して読み出すことは出来ません。読み出す場合は、DMAC Read Requeset レジスタに値をセットして、次に、DMAC Read Value レジスタより、値を読み出します。

Address =(0xF440_0000)+ (0x0048)

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	mwcadr	RO	0x00000000	マスタライトカレントアドレス設定レジスタ

(個別説明)

a. <mwcadr>

マスタライト転送における Endpoint からマスタライト用バッファへの現在までの転送完了アドレスを表示します。タイムアウト割り込みが発生した場合や、転送途中でエラーが発生した際に使用することができます。

本アドレスは Endpoint からマスタライト用バッファヘデータがセットされた時点でインクリメントされ、マスタライト転送途中では、表示アドレスまでのデータはターゲットデバイスまたはマスタライト用バッファ内に存在することとなります。

14.UDMWAHBADR (Master Write AHB Address register)

マスタライト転送(UDC2→AHB)におけるターゲットデバイスへの転送完了アドレスを表示します。

DMA 転送は、条件によって、バイト単位でアクセスを実行する場合があります。バイトアクセス実行時でも、保存されるアドレスは、ワード境界となりますので注意して下さい。

Address =(0xF440_0000)+ (0x004C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	mwahbadr	RO	0xFFFFFFFF	マスタライト AHB アドレス設定レジスタ

(個別説明)

a. <mwahbadr>

マスタライト転送におけるターゲットデバイスへの転送完了アドレスを表示します。タイムアウト割り込みが発生した場合や、転送途中でエラーが発生した際に使用することができます。本アドレスはターゲットデバイスへデータがセットされた時点でインクリメントされ、マスタライト転送途中では、表示アドレスまでのデータはターゲットデバイスに存在することとなります。

15.UDMRSADR (Master Read Start Address register)

マスタリード転送(AHB→UDC2)のスタートアドレスを設定します。

Address =(0xF440_0000)+ (0x0050)

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	mrsadr	R/W	0xFFFFFFFF	マスタリードスタートアドレス設定レジスタ

(個別説明)

a. <mrsadr>

マスタリード転送のスタートアドレスを設定して下さい。ただし、本マスタはアドレス増加にのみ対応していますので、Master Read End Address レジスタよりも下位の値を設定して下さい。

16.UDMREADR (Master Read End Address register)

マスタリード転送(AHB→UDC2)のエンドアドレスを設定します。

Address =(0xF440_0000)+ (0x0054)

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	Mreadr	R/W	0xFFFFFFFF	マスタリードエンドアドレス設定レジスタ

(個別説明)

a. <Mreadr>

マスタリード転送のエンドアドレスを設定して下さい。ただし、本マスタはアドレス増加にのみ対応していますので、Master Read Start Address よりも上位の値を設定して下さい。

17.UDMRCADR (Master Read Current Address register)

マスタリード転送(AHB→UDC2)におけるターゲットデバイスから Endpoint への転送完了アドレスを示します。

本レジスタは、直接アドレスを指定して読み出すことは出来ません。読み出す場合は、DMAC Read Requeset レジスタに値をセットして、次に、DMAC Read Value レジスタより、値を読み出します。

Address =(0xF440_0000)+ (0x0058)

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	mrcadr	RO	0x00000000	マスタリードカレントアドレス設定レジスタ

(個別説明)

a. <mrcadr>

マスタリード転送におけるターゲットデバイスから Endpoint への現在までの転送完了アドレスを表示します。

本アドレスはマスタリード用バッファから Endpoint へデータがセットされた時点でインクリメントされ、マスタリード転送途中では、表示アドレスまでのデータは Endpoint 用 FIFO 内に存在することとなります。

18.UDMRAHBADR (Master Read AHB Address register)

マスタリード転送(AHB→UDC2)におけるターゲットデバイスから、UDC2AB への転送完了アドレスを表示します。

DMA 転送は、条件によって、バイト単位でアクセスを実行する場合があります。バイトアクセス実行時に保存されるアドレスは、ワード境界となります。

Address =(0xF440_0000)+ (0x005C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	mrahbadr	RO	0xFFFFFFFF	マスタリード AHB アドレス設定レジスタ

(個別説明)

a. <mrahbadr>

マスタリード転送におけるターゲットデバイスから、UDC2AB への転送完了アドレスを表示します。本アドレスはターゲットデバイスからデータがセットされた時点でインクリメントされ、マスタリード転送途中では、表示アドレスまでのデータは、バッファまたは、Endpoint 用 FIFO に存在することとなります。

19. UDPWCTL (Power Detect Control register)

UDC2AB のリセット/サスペンド時の制御を行います。

Address =(0xF440_0000)+ (0x0080)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	—	Undefined	—	Read as undefined. Write as zero.
[7]	wakeup_en	R/W	0y0,(-)	ウェイクアップイネーブル 0y0: WAKEUP_X 信号をアサートしない。 0y1: WAKEUP_X 信号をアサートする。 (注 1)
[6]	phy_remote_wkup	R/W1S	0y0,(-)	リモートウェイクアップ設定レジスタ 0y0: ノーオペレーション 0y1: ウェイクアップ
[5]	phy_resetb	R/W	0y1,(-)	PHY リセットレジスタ 0y0: リセット・アサート 0y1: リセット・デアサート
[4]	suspend_x	RO	0y1	サスペンド検出レジスタ 0y0: サスペンド状態(suspend_x=0) 0y1: 非サスペンド状態(suspend_x=1)
[3]	phy_suspend	R/W	0y0,(-)	PHY サスペンド設定レジスタ 0y0: 非サスペンド状態 0y1: サスペンド状態
[2]	pw_detect	RO	0y0,(-)	USB バス電源検出レジスタ 0y0: USB バス・ディスコネクト(VBUSPOWER=0) 0y1: USB バス・コネクト(VBUSPOWER=1) (注 2)
[1]	pw_resetb	R/W	0y1,(-)	パワーリセット設定レジスタ 0y0: リセット・アサート 0y1: リセット・デアサート
[0]	usb_reset	RO	0y0	USB_RESET 設定レジスタ 0y0: usb_reset=0 0y1: usb_reset=1

注 1) 本来、UDC2AB は Wakeup 信号をアサートする機能を持っていますが、本 LSI ではサポートしていません。

注 2) 本来、UDC2AB は VBUS 検出時に int_powerdetect 割り込みをアサートする機能を持っていますが、本 LSI ではサポートしていません。UDPWCTL<pw_detect>は常に 0 を示します。

(個別説明)

a. <wakeup_en>

USB のサスペンド時に、(AHB 側)System を Sleep させて CLK_H を停止する時には、本ビットを 1 にセットして下さい。本ビットを 1 にセットしているとサスペンドが解除された時(suspend_x=1)、またはディスコネクトされた時(VBUSPOWER=0)に、WAKEUP_X 信号が非同期で 0 にアサートされますので、System の Resume に利用可能です。

本ビットの利用方法に関しては「3.16.2.13(4) サスペンド、レジューム(ディスコネクト)時の信号動作」も参照して下さい。

0y0: WAKEUP_X 信号をアサートしない。

0y1: WAKEUP_X 信号をアサートする。

注) 本来、UDC2AB は Wakeup 信号をアサートする機能を持っていますが、本 LSI ではサポートしていません。

b. <phy_remote_wkup>

USB のリモートウェイクアップ機能を実行するために使用します。本ビットに 1 をセットすることで、udc2_wakeup 出力信号(UDC2 の wakeup 入力端子)を 1 にアサートすることができます。但し、UDC2 がサスペンドを検出していない時(suspend_x=1 の時)に本ビットを 1 にセットした場合は無視されます(1 にセットされません)ので、サスペンド検出時にのみセットして下さい。USB レジューム完了時(suspend_x デアサート時)に自動的に 0 にクリアされます。

本ビットの利用方法に関しては「3.16.2.13(4) サスペンド、レジューム(ディスコネクト)時の信号動作」も参照して下さい。

0y0: ノーオペレーション

0y1: ウェイクアップ

c. <phy_resetb>

このビットに 0 をセットすると、PHYRESET 出力信号が 1 へアサートされます。PHYRESET 信号は PHY のリセットに利用可能です。このビットは自動解除されませんので、必ず PHY のリセット仕様時間経過後に 1 へクリアして下さい。

0y0: リセット・アサート

0y1: リセット・デアサート

d. <suspend_x>

サスペンド信号を検出します(UDC2 からの suspend_x 信号を同期化した値です)。

0y0: サスペンド状態(suspend_x=0)

0y1: 非サスペンド状態(suspend_x=1)

e. <phy_suspend>

本ビットを 1 にセットすることで、PHYSUSPEND_X 出力信号が 0 へアサート(CLK_H 同期)されます。PHY をサスペンドする時の端子として使用可能です。

本ビットを 1 にセットすると、UDC2 レジスタと DMAC Read Request レジスタへのアクセスが禁止となります。

レジューム時(UDC2 の suspend_x デアサート時)に自動的に 0 にクリアされます。

本ビットの利用方法に関しては「3.16.2.13(4) サスペンド、レジューム(ディスコネクト)時の信号動作」も参照して下さい。

0y0: 非サスペンド状態

0y1: サスペンド状態

f. <pw_detect>

VBUSPOWER 入力端子の状態を示します。

0y0: USB バス・ディスコネクト(VBUSPOWER=0)

0y1: USB バス・コネクト(VBUSPOWER=1)

注) 本来、UDC2AB は VBUS 検出時に int_powerdetect 割り込みをアサートする機能を持っていますが、本 LSI ではサポートしていません。UDPWCTL<pw_detect>は常に 0 を示します。

g. <pw_resetb>

UDC2AB用のソフトウェアリセットです。(詳細は「3.16.2.6 リセット」参照)

本ビットを 0 にセットすることで、PW_RESETB 出力端子が 0 にアサートされます。

マスタ動作が停止した状態でリセットを行って下さい。

このビットは自動解除されませんので、必ずクリアして下さい。

0y0: リセット・アサート

0y1: リセット・デアサート

h. <usb_reset>

UDC2 からの usb_reset 信号を同期化した値です。

0y0: usb_reset = 0

0y1: usb_reset = 1

Not Recommended
for New Design

20.UDMSTSTS (Master Status register)

UDC2AB のステータスレジスタです。

Address =(0xF440_0000)+ (0x0084)

Bit	Bit Symbol	Type	Reset Value	Description
[31:5]	—	—	Undefined	Read as undefined.
[4]	mrepempty	RO	0y0,(-)	マスタリード Endpoint エンプティ確認レジスタ 0y0: Endpoint にデータがあることを示します。 0y1: Endpoint が空であることを示します。
[3]	mrbfemp	RO	0y1	マスタリードバッファエンプティ確認レジスタ 0y0: マスタリード DMA 用バッファにデータがあることを示します。 0y1: マスタリード DMA 用バッファが空であることを示します。
[2]	mwbfemp	RO	0y1	マスタライトバッファエンプティ確認レジスタ 0y0: マスタライト DMA 用バッファにデータがあることを示します。 0y1: マスタライト DMA 用バッファが空であることを示します。
[1]	mrepdset	RO	0y0,(-)	マスタリード Endpoint DATASET 設定レジスタ 0y0: Endpoint 内にデータを転送可能です。 0y1: Endpoint 内にデータを転送するスペースがありません。
[0]	mwepdset	RO	0y0,(-)	マスタライト Endpoint DATASET 設定レジスタ 0y0: Endpoint 内にデータはありません。 0y1: Endpoint 内に読み出すべきデータがあります。

(個別説明)

a. <mrepempty>

UDC2Rx 用 EP が空であることを示すレジスタです。UDC2 Setting レジスタの tx0 ビットを使って NULL パケットを送信する場合には、このビットが 1 であることを確認して下さい。(本ビットは eptx_empty 入力信号を CLK_H 同期したものです。)

0y0: Endpoint にデータがあることを示します。

0y1: Endpoint が空であることを示します。

b. <mrbfemp>

UDC2AB 内のマスタリード DMA 用バッファが空であることを示します。

0y0: マスタリード DMA 用バッファにデータがあることを示します。

0y1: マスタリード DMA 用バッファが空であることを示します。

c. <mwbfemp>

UDC2AB 内のマスタライト DMA 用バッファが空であることを示します。

0y0: マスタライト DMA 用バッファにデータがあることを示します。

0y1: マスタライト DMA 用バッファが空であることを示します。

d. <mrepdset>

マスタリード DMA 転送により、UDC2 の Tx 用 EP へ送信データがセットされ、Endpoint に書き込むスペースがなくなると 1 にセットされます。ホストからの IN-Token により UDC2 からデータが転送されると 0 になります。このビットが 0 であるときは Endpoint への DMA 転送が可能です。(本ビットは eptx_dataset 入力信号を CLK_H 同期したものです。)

0y0: Endpoint 内にデータを転送可能です。

0y1: Endpoint 内にデータを転送するスペースがありません。

e. <mwepdset>

UDC2 の Rx 用 EP へ受信データがセットされると 1 にセットされます。全データがマスタライト用 DMA により読み出されると 0 になります。(本ビットは eprx_dataset 入力信号を CLK_H 同期したものです。)

0y0: Endpoint 内にデータはありません。

0y1: Endpoint 内に読み出すべきデータがあります。

21. UDTOUTCNT (Timeout Count register)

タイムアウトカウント値を読み出すレジスタです。(デバッグ用)
本レジスタは、直接アドレスを指定して読み出すことは出来ません。読み出す場合は、DMAC Read Requet レジスタに値をセットして、次に、DMAC Read Value レジスタより、値を読み出します。

Address=(0xF440_0000)+(0x0088)

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	tmoutcnt	RO	0x00000000	タイムアウトカウントレジスタ

(個別説明)

a. <tmoutcnt>

デバッグ用です。Master Write Timeout レジスタの timeout_en ビットをイネーブルにした場合に、タイマの値が読み出せます。マスタライト用 Endpoint(Rx 用 EP)が空になってから CLK_U をカウントするごとにデクリメントされます。

22. UDC2 (UDC2 register) (0x0200~0x03FC)

(0xF440_0000)+0x200-0x3FC にアクセスすることで UDC2 の内部レジスタ(16bit)にアクセス可能です。UDC2AB の AHB データバスは 32bit ありますが、bit15-0 が UDC2 データバスに対応します。bit31-16 は Reserved bit となり、リードオンリー(リード値: 0)となります。ライト/リード共に WORD(32bit)アクセスを実行して下さい。(ただし、EPx_FIFO レジスタへのライトアクセスでは、BYTE(8bit)アクセスを行う場合があります。詳細は後述します。)

ライト/リード共にアクセス完了までに時間がかかります(UDC2 へのアクセス期間)。int_udc2_reg_rd 割り込みを利用して、必ず前の UDC2 レジスタアクセスが完了してから次のアクセスを開始して下さい。(リード時は UDC2RDREQ<udc2rdreq>でもアクセス状況が確認可能です。)

- ライトアクセス

UDC2 レジスタへライトアクセスを実行する場合には、該当するアドレスに直接書き込んで下さい。

- リードアクセス

UDC2 レジスタへリードアクセスを実行する場合には、UDC2 Read Request レジスタと UDC2 Read Value レジスタを使用して下さい。

まず、UDC2 Read Request レジスタにアクセスするアドレスをセットして、次に、読み出し用の UDC2 Read Value レジスタより、データを読み出して下さい。アドレスマップに示されたアドレスから直接読み出すことは出来ません。

- EPx_FIFO レジスタ

EPx_FIFO レジスタにライトアクセスする場合、UDC2 PSCI I/F にて下位 1 バイトアクセスが必要となる場合があります。この時は、UDC2AB に対して、下位 1 バイトへの BYTE アクセスを行って下さい。

リードアクセス時に下位 1 バイトのアクセスが必要な場合は、通常通り、UDC2 Read Request レジスタ経由でアクセスを行い、UDC2 Read Value レジスタからデータをリードして下さい。この時、UDC2 Read Value レジスタへのアクセスは、WORD/BYTE アクセスのどちらでも構いません。

- UDC2 内 Reserved レジスタ

接続する UDC2 で未サポートとなっている Endpoint のレジスタ、Reserved のレジスタにはアクセスしないで下さい。(仮にアクセスした場合は、UDC2AB から UDC2 へのアクセス自体は発生します。ライトアクセスの場合は、UDC2 への Dummy ライトとなります。リードアクセスの場合は、UDC2 からのリードデータ(udc2_rdata)は不定値となり、UDC2 Read Value レジスタに不定値がセットされます。)

- UDC2 サスペンド時のアクセス

UDC2 がサスペンド状態の時に、PHY からのクロック(=CLK_U)供給が停止している場合は、UDC2 へのレジスタアクセスは不可能となります。この時は UDC2 へのレジスタアクセスは実行しないで下さい。なお、Power Detect Control レジスタの phy_suspend ビットを 1 にセットしている時に、UDC2 レジスタにアクセスした場合、AHB エラー応答となります。

Address =(0xF440_0000)+ (0x0200-0x03FC)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	—	—	Undefined	Read as undefined. Write as zero.
[15:0]	udc2_data	—	—	UDC2 データレジスタ データ値の詳細は UDC2 の章を参照して下さい。

UDC2 レジスタへのアクセスフローの図を以下に示します。

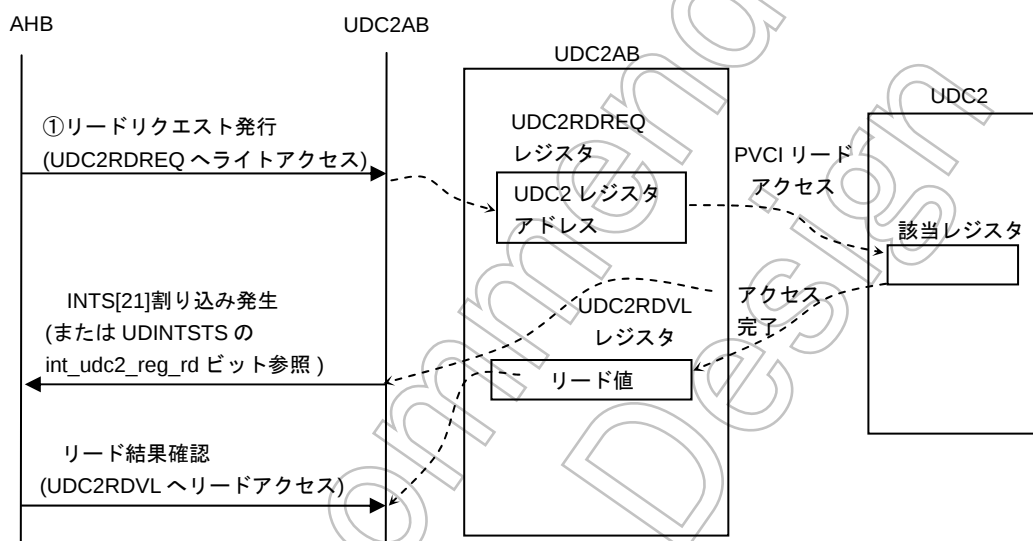


図 3.16.7 UDC2 レジスタ リードアクセスのフロー

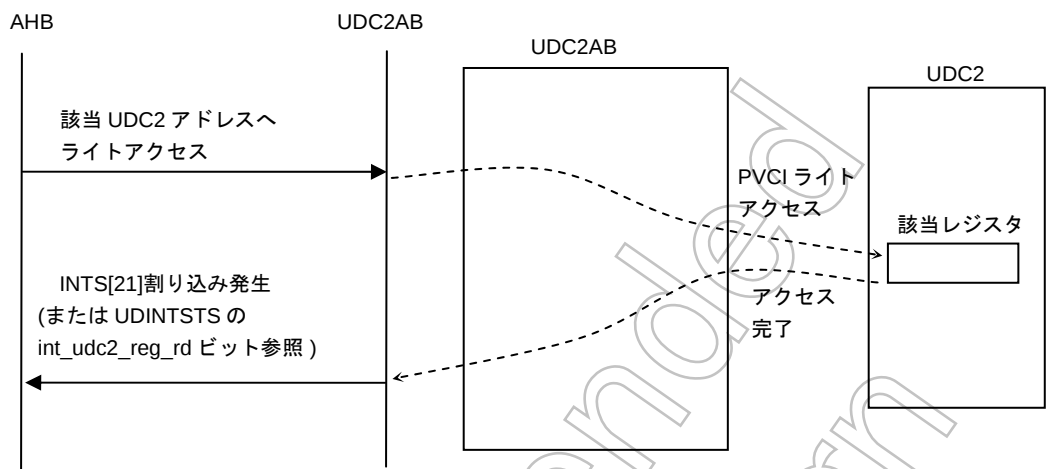


図 3.16.8 UDC2 レジスタ ライトアクセスのフロー

3.16.2.6 リセット

UDC2AB は、UDPWCTL<pw_resetb>によるソフトウェアリセットをサポートしています。

また、DMA マスタ転送用に、マスタチャネルリセット(UDMSTSET<mr_reset/mw_reset>)もサポートしています。

- ソフトウェアリセット(UDPWCTL<pw_resetb>)

各レジスタのビットには、ハードウェアリセットでは初期化されるが、ソフトウェアリセットでは初期化されずに値が保持されるものがあります。詳細は各レジスタの説明に記載していますので、「3.16.2.5 レジスタ」を参照して下さい。

USB バス電源を検出した際には初期化を行う必要がありますので、ソフトウェアリセットを行って下さい。

- マスタチャネルリセット(UDMSTSET<mr_reset/mw_reset>)

マスタライト転送ブロックへはmw_resetビット、マスタリード転送ブロックへはmr_resetビットを用意していますが、該当するマスタブロックの初期化を行うのみでUDC2ABレジスタは初期化されません。各リセットの使用方法については「3.16.2.5 (2) 5. UDMSTSET (DMAC Setting register)」を参照して下さい。

3.16.2.7 割り込み信号(INTS[21])

UDC2AB の割り込み出力信号: INTS[21]は、UDC2 から発生する割り込みと、それ以外から発生する割り込みから構成されます。割り込み条件が成立すると、UDC2AB は内部の Interrupt Status レジスタの対応ビットをセットします。このビットがセットされた時、Interrupt Enable レジスタの該当ビットがイネーブルに設定されていると、INTS[21]がアサートされます。

Interrupt Enable レジスタの該当ビットをディセーブルに設定している時は、割り込み要因が発生すると、対応する Interrupt Status レジスタのビットには1がセットされますが、INTS[21]はアサートされません。Interrupt Status レジスタがセットされている状態で、Interrupt Enable レジスタの該当ビットをイネーブルにセットすると、セット直後に INTS[21]がアサートされます。

Interrupt Enable レジスタの初期値は、全て 0(ディセーブル)です。

上記のイメージ図が下図となります。

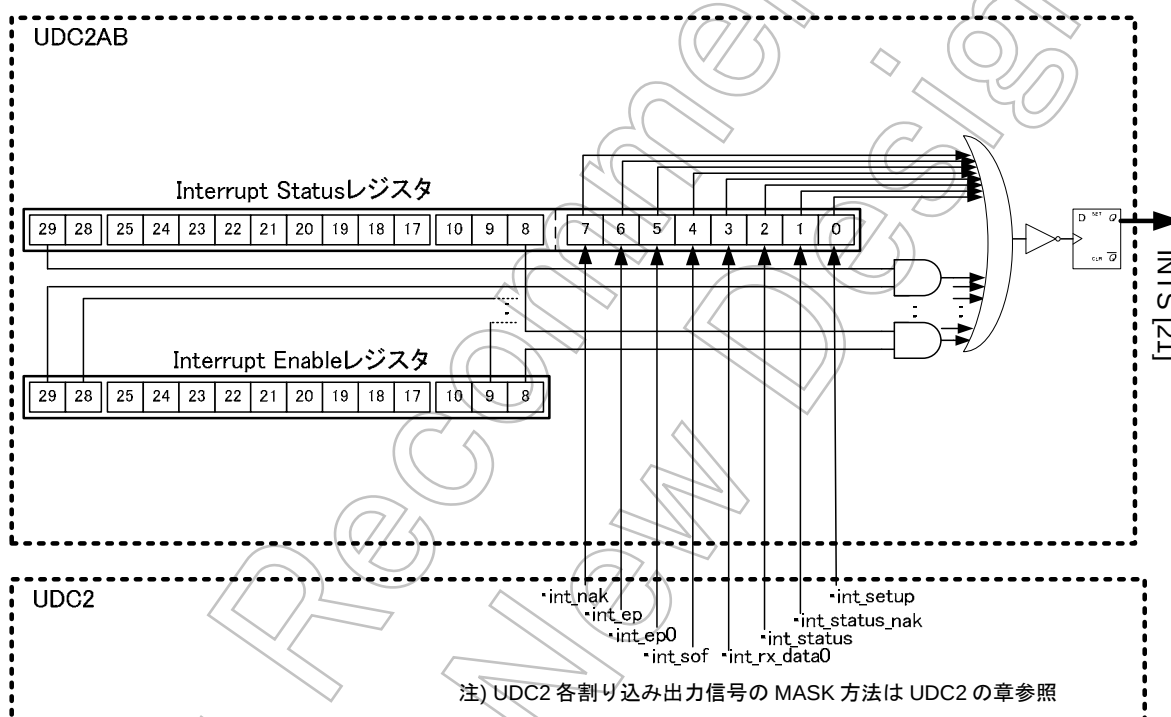


図 3.16.9 INTS[21]とレジスタの関係

3.16.2.8 全体動作シーケンス

UDC2AB の全体動作シーケンスは以下のようになります。

1. ハードウェアリセット

2. 割り込み信号の設定

Interrupt Enable レジスタにて、必要となる割り込み要因のビットをイネーブルにしてください。

詳細は「3.16.2.7割り込み信号(INTS[21])」を参照して下さい。

3. USB バス電源検出(コネク)と初期化

詳細は「3.16.2.11USBバス電源検出シーケンス」を参照して下さい。

注) 本来、UDC2AB はVBUS 検出時に int_powerdetect 割り込みをアサートする機能を持っていますが、本 LSI ではサポートしていません。UDPWCTL<pw_detect>は常に 0 を示します。

4. USB エミュレーション応答

詳細はUDC2 の章「3.16.3.4USBデバイス応答」を参照して下さい。

5a. マスタリード転送

USB ホストからの受信リクエストに対応して、マスタリード転送を行って下さい。

詳細は「3.16.2.9(1) マスタリード転送」を参照して下さい。

5b. マスタライト転送

USB ホストからの送信リクエストに対応して、マスタライト転送を行って下さい。

詳細は「3.16.2.9(4) マスタライト転送」を参照して下さい。

6. USB バス電源切断(ディスコネク)

任意のタイミングで USB バス電源が切断される可能性があります。

詳細は「3.16.2.11USBバス電源検出シーケンス」を参照して下さい。

注) 本来、UDC2AB はVBUS 検出時に int_powerdetect 割り込みをアサートする機能を持っていますが、本 LSI ではサポートしていません。UDPWCTL<pw_detect>は常に 0 を示します。

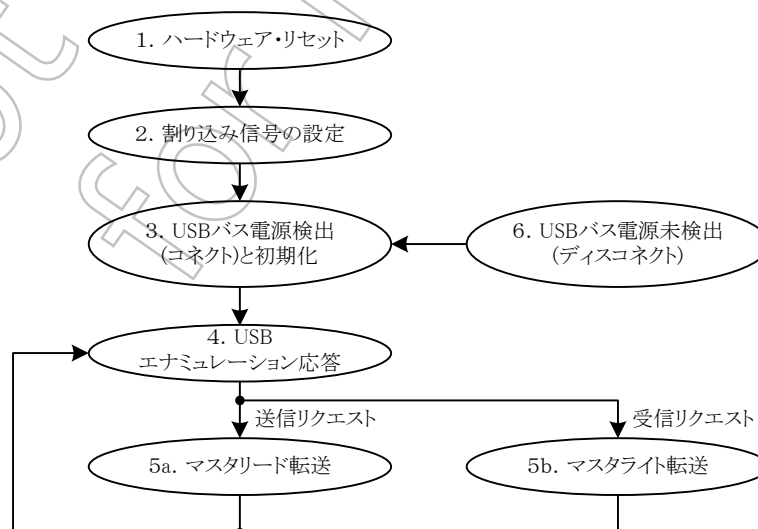


図 3.16.10 全体動作シーケンス

3.16.2.9 マスタ転送動作

UDC2AB のマスタ転送動作について説明します。

マスタ転送を起動する場合は、UDC2 の該当 Endpoint の転送設定(UDC2 EP_x Status レジスタの bit14-bus_sel) を、必ず”直接アクセスモード”に設定して下さい。“共通バスアクセス”に設定されている状態での DMAC の起動は禁止としています。

(1) マスタリード転送

- EOP イネーブルモード

UDC2STSET <eopb_enable>を 1(マスタリード EOP イネーブル)に設定時のマスタリード転送について説明します。マスタリード動作は下記のような動作となります。

1. Master Read Start Address レジスタと Master Read End Address レジスタを設定して下さい。
2. DMAC Setting レジスタのマスタリード動作に関連するビットを設定し、mr_enable ビットに 1 をセットして下さい。
3. UDC2AB は、UDC2 の Endpoint へのデータ転送を開始します。UDC2 は USB ホストからの IN トークンに対してデータを転送します。
4. マスタリード転送がマスタリードエンドアドレスまで到達すると、UDC2AB は int_mr_end_add 割り込みをアサートします。
5. ソフトウェアによる対応が終了したならば、1 へ戻ります。

注 1) ショートパケットについて

転送サイズ(Master Read End Address - Master Read Start Address + 1)が Max パケットサイズの等倍ではない場合は、最後の IN 転送がショートパケットの転送となります。

例: マスタリード転送サイズ: 1035 バイト、Max パケットサイズ: 512 バイト の場合、
1 回目: 512 バイト → 2 回目: 512 バイト → 3 回目: 11 バイト で転送が行われる。

注 2) int_mr_end_add 割り込みについて

int_mr_end_add 割り込みは、UDC2 Endpoint へのデータ転送が終了した時点で発生します。UDC2 から USB ホストに向かっての全てのデータ転送が終了したことを確認するためには、UDMSTSTS<mrepempty>を確認して下さい。

- EOP ディセーブルモード

UDC2STSET <eopb_enable>を 0(マスタリード EOP ディセーブル)に設定時のマスタリード転送について説明します。マスタリード動作は下記のような動作となります。

1. Master Read Start Address レジスタと Master Read End Address レジスタを設定して下さい。
2. DMAC Setting レジスタのマスタリード動作に関連するレジスタを設定し、mr_enable ビットに 1 をセットして下さい。
3. UDC2AB は、UDC2 の Endpoint へのデータ転送を開始します。UDC2 は USB ホストからの IN トークンに対してデータを転送します。
4. マスタリードエンドアドレスまで到達すると、UDC2AB は int_mr_end_add 割り込みをアサートします。マスタリード転送にて Endpoint の FIFO が Max パケットサイズに達している場合、USB ホストからの IN トークンに対してデータが転送されますが、達していない場合 FIFO にデータが残り、次回以降の転送に持ち越されます。
5. ソフトウェアによる対応が終了したならば、1 へ戻ります。

注) UDC2AB を EOP ディセーブルモードで使用する場合、送信すべきデータ列の転送が終了してもショートパケットを送信しません。データ列が Max パケットサイズの等倍の時にのみ EOP ディセーブルモードを使用して下さい。
データ列の合計が Max パケットサイズの等倍であれば構いませんので、例えば、以下のような転送が可能です。

例:

- ・ 1 回目マスタリード転送サイズ: 1000 バイト
- ・ 2 回目マスタリード転送サイズ: 24 バイト (1 回目+2 回目=1024 バイト)
- ・ Max パケットサイズ: 512 バイト

といった構成の時は、IN 転送に対して 512 バイト×2 回の転送が行われる。

(2) マスタリード転送のアボート処理

マスタリード転送中断(アボート)動作は下記のような動作となります。

1. UDC2 Command レジスタを使用して、該当 Endpoint のステータスを Disable 状態 (EP_Disable) にして下さい。(もし、Endpoint を Disable 状態にしないでアボートした場合、意図していないデータを USB ホストに送信する可能性があります。)
2. マスタリード転送を中止するため、UDMSTSET<mr_abort>に 1(アボート)をセットして下さい。
3. アボート完了を確認するため、UDMSTSET<mr_enable>が 0 にディセーブルされるのを確認して下さい。<mr_enable>が 1 の間は、次の動作を実行しないで下さい。
(アボート完了時の転送終了アドレスの情報は、Master Read Current Address レジスタ、Master Read AHB Address レジスタにて確認可能です。)
4. マスタリード転送ブロックを初期化するため、UDMSTSET<mr_reset>を 1(リセット) にして下さい。
5. 該当 Endpoint に対して、Command レジスタ(EP_FIFO_Clear)を使用して FIFO を初期化して下さい。
6. Command レジスタ(EP_Enable)を使用して該当 Endpoint を Enable 状態として下さい。

(3) マスタリード転送時の Max パケットサイズ設定

UDC2ABのマスタリード機能と接続する Endpoint の Max パケットサイズが奇数となる場合、以下のような制限がありますのでご注意ください。

- Endpoint の Max パケットサイズを奇数として扱う場合でも、UDC2 EPx_MaxPacketSize レジスタの max_pkt ビットの設定は偶数として下さい。
注) この設定に関する詳細は「UDC2 の章 3.16.4.2 Appendix(Appendix B)」を参照して下さい。
- UDC2STSET <eopb_enable>を 1(マスタリード EOP イネーブル)にセットして下さい。
- 1 回のマスタリード転送で指定する転送サイズ(Master Read End Address - Master Read Start Address + 1) を奇数である Max パケットサイズ以下として下さい。

(例) 上記を満たす設定は以下のような構成となります。

- Endpoint の Max パケットサイズ(USB ホストへ伝える値)を 63 バイトとする。
- UDC2 EPx_MaxPacketSize レジスタの max_pkt ビットの設定を 64 バイトとする。
- 1 回のマスタリード転送で指定する転送サイズを 63 バイト以下とする。

(4) マスタライト転送

● マスタライト転送シーケンス

マスタライト動作について説明します。マスタライト動作は下記のような動作となります。

1. Master Write Start Address レジスタと Master Write End Address レジスタを設定して下さい。
2. DMAC Setting レジスタのマスタライト動作に関連するビットを設定し、mw_enable ビットに 1 をセットして下さい。
3. UDC2AB は、USB ホストから受信した Endpoint 内のデータに対してマスタライト転送を行います。
4. (タイムアウト処理が発生しない状態で)マスタライトエンドアドレスに到達するまで書き込みが終了した場合、int_mw_end_add 割り込みがアサートされますので、ソフトウェアにより必要な処理を行って下さい。UDC2 が正常なパケットを受信すると 1 へ戻ります。

注) UDC2AB は、UDMSTSET<mw_enable>がディセーブル状態で、USB ホストから正常にパケットを受信すると int_mw_set_add 割り込みをアサートします。

(5) タイムアウト処理

マスタライト転送時、マスタライトエンドアドレスに到達する前に USB ホストからの OUT 転送が停滞してしまった場合、マスタライト転送が終了しません。この場合に備えて、タイムアウト機能を設定することが出来ます。

このタイムアウト機能を利用する場合、タイムアウト時点で UDC2AB 内バッファに格納されているデータは、全て AHB 側に転送されます。

タイムアウト処理は下記のような動作となります。

1. マスタライト転送開始前に Master Write Timeout レジスタへアクセスし、`timeoutset`(タイムアウト時間)を設定し `timeout_en` をイネーブル(1)として下さい。
2. 前項の説明のように、マスタライト転送を開始して下さい。
3. タイムアウトが発生した場合、`int_mw_timeout` 割り込みがアサートされます。
(`int_mw_end_add` 割り込みはアサートされません。)この場合は、マスタライトエンドアドレスまでマスタライト転送が完了していません。UDC2AB は `UDMSTSET<mw_enable>` を 0 にクリアします。
4. Master Write Current Address レジスタで、AHB 側へ転送が完了したアドレスを確認する事ができます。

なお、タイムアウト機能をイネーブルとしてマスタライト転送実行中に、タイムアウト用カウンタが進みますが、USB ホストから該当 Endpoint への OUT 転送を受信した時は、カウンタは設定値に戻り、再カウントを開始します(下図)。つまり、タイムアウトするまでの時間は、「マスタライト転送を開始した時点から設定時間まで」ではなく、「マスタライト転送中に、最後に USB ホストから該当 Endpoint へ転送が発生した時から設定時間まで」となります。

タイムアウト機能を使用しない場合は、必ずマスタライト転送開始前に Master Write Timeout レジスタの `timeout_en` ビットをディセーブル(0)として下さい。この場合は設定されたマスタライトエンドアドレスに到達するまで転送が終了しません。

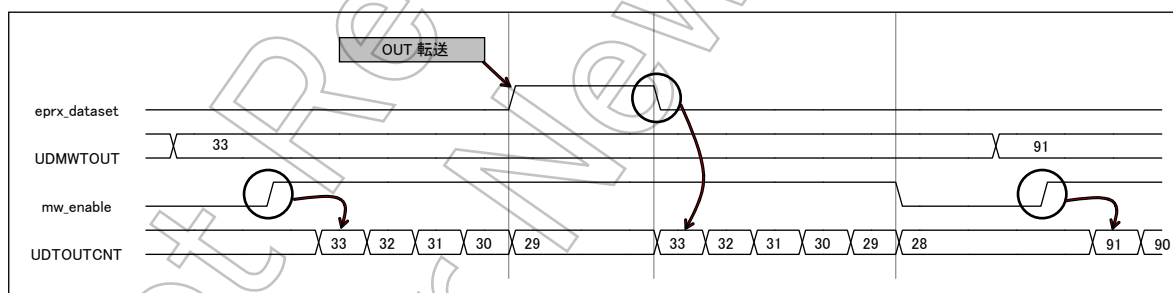


図 3.16.11 MW タイムアウトカウント例

(6) マスタライト転送のアボート処理

マスタライト転送中断(アボート)動作は下記のような動作となります。

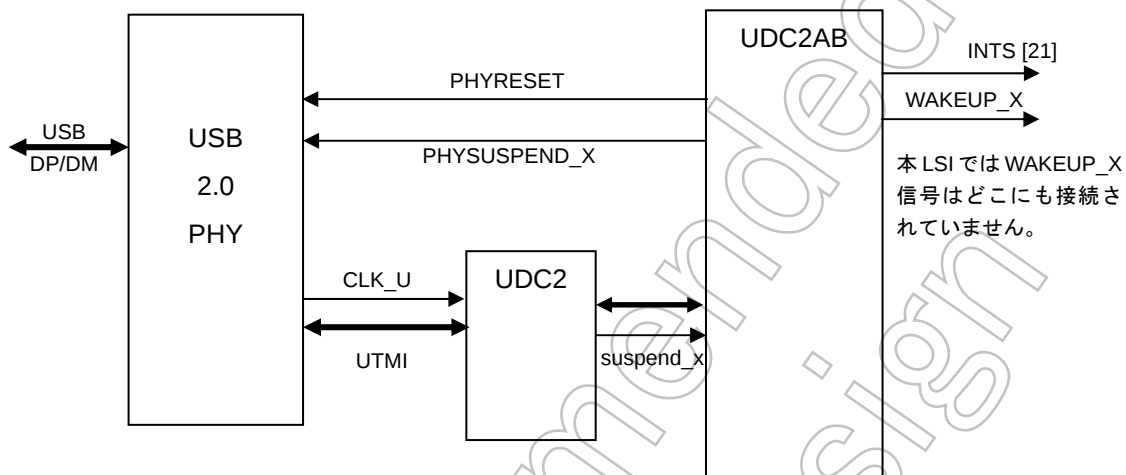
1. UDC2 Command レジスタを使用して、該当 Endpoint のステータスを Disable 状態 (EP_Disable) にして下さい。
2. マスタライト転送を中止するため、UDMSTSET<mw_abort>に 1(アボート)をセットして下さい。
3. アボート完了を確認するため、UDMSTSET<mw_enable>が 0 にディセーブルされるのを確認して下さい。<mw_enable>が 1 の間は、次の動作を実行しないで下さい。
(アボート完了時の転送終了アドレスの情報は、Master Write Current Address レジスタ、Master Write AHB Address レジスタにて確認可能です。)
4. マスタライト転送ブロックを初期化するため、UDMSTSET<mw_reset>を 1(リセット)にして下さい。
5. 該当 Endpoint に対して、Command レジスタ(EP_FIFO_Clear)を使用して FIFO を初期化して下さい。
6. UDC2 Command レジスタを使用して、該当 Endpoint のステータスを Enable 状態 (EP_Enable) にして下さい。

3.16.2.10 USBパワーマネジメント制御

USB では通常のパケット転送の他にも、USB バス電源の検出やサスペンド、レジューム等のパワーマネジメントに関する動作が規定されています。本章ではこれら動作時の制御方法について説明を行います。

下図は、パワーマネジメント制御に関する信号の接続図です。

注) 各動作については必ず USB2.0 Spec をご覧下さい。



※UTMI: USB2.0 Transceiver Macrocell Interface

図 3.16.12 制御信号の接続図

3.16.2.11 USBバス電源検出シーケンス

(1) コネクト

本章では電源検出時のシーケンスを説明します。USB ホストからのバス電源(VBUS)検出後に、以下の手順にて UDC2AB と UDC2 の初期化を行って下さい。

1. UDPWCTL <pw_resetb>にてソフトウェアリセットを行って下さい。(pw_resetb ビットは自動解除されないで、ソフトウェアでクリアして下さい。)
2. UDC2AB レジスタと UDC2 レジスタにアクセスし、必要な初期設定を行って下さい。
3. UDC2 Command レジスタにて USB Ready コマンドを発行して下さい。UDC2 は PHY を通して USB ホストにコネクトを通知します。この状態で UDC2 は USB ホストからの USB_RESET を受け付けることが可能となります。
4. UDC2 は USB ホストからの USB_RESET を検出すると UDC2 内部レジスタの初期化を行い、USB ホストとのエミュレーションが可能な状態となります。なお、USB_RESET 検出時には int_usb_reset/int_usb_reset_end 割り込みが発生します。

注) 本来、UDC2AB は VBUS 検出時に int_powerdetect 割り込みをアサートする機能を持っていますが、本 LSI ではサポートしていません。UDPWCTL<pw_detect>は常に 0 を示します。

(2) ディスコネクト

USB バス電源が切断状態となった場合、UDC2AB は外部割り込みにて通知します。この時、各マスタ転送は自動的に停止しませんので、アボート処理を実施して下さい。その後、UDPWCTL <pw_resetb>によりソフトウェアリセットを行って下さい。

なお、System にて USB サスペンド中に(AHB 側)CLK_H を停止させる制御を行っている場合、CLK_H 停止中にディスコネクト状態となっても割り込みは通知されません。

この場合はWAKEUP_X出力信号を利用してCLK_Hの復帰処理が必要となります。詳細は「3.16.2.13(4) サスペンド、レジューム(ディスコネクト)時の信号動作」を参照して下さい。

3.16.2.12 USB_RESET

USB_RESET は USB ホスト接続時だけではなく、任意のタイミングで受信する可能性があります。

UDC2AB は、UDC2 が USB_RESET を受信すると `int_usb_reset/int_usb_reset_end` 割り込みをアサートし、Default ステートに戻ります。この時、各マスタ転送は自動的に停止することはありません。アボート機能を使用して転送を終了させて下さい。なお、UDC2 のレジスタには USB_RESET によって値が初期化されるものと値を保持するものがあります(詳しくは UDC2 の章を参照して下さい)。

USB_RESET 認識時に UDC2 レジスタの再設定を行う場合は、`int_usb_reset_end` 割り込み発生後に行ってください。これは、UDC2 が `usb_reset` 信号をデアサートするタイミングで UDC2 レジスタを初期化するためです。

Not Recommended
for New Design

3.16.2.13 サスペンド、レジューム

(1) サスペンド状態への移行

UDC2AB は UDC2 のサスペンド状態の検出を、int_suspend_resume 割り込みと、UDPWCTL<suspend_x>により通知します。この時、各マスタ転送は自動的に停止しませんが、停止する必要がある場合には各マスタ転送のアボート機能を使用して強制終了して下さい。ソフトウェアにて必要な処理が終了した後に、PHY をサスペンド(クロック停止)する必要がある場合は、Power Detect Control レジスタの phy_suspend ビットをセットすることにより、UDC2AB が PHYSUSPEND_X をアサートし PHY がサスペンド状態となります。

(2) サスペンド状態からの復帰(レジューム)

UDC2AB は USB ホストからのレジューム状態の検出を、int_suspend_resume 割り込みと、UDPWCTL<suspend_x>により通知します。(CLK_H を停止させている場合、事前に UDPWCTL<wakeup_en>をイネーブルとしている場合には、WAKEUP_X 出力信号により通知します。)

注) 本来、UDC2AB は Wakeup 信号をアサートする機能を持っていますが、本 LSI ではサポートしていません。

なお、レジューム時には PHY へのサスペンド信号(PHYSUSPEND_X)は自動的にデアサートされるので、サスペンド時と異なりソフトウェアで制御する必要はありません。レジュームを認識したら、各マスタ転送を再開するために再設定して下さい。

(3) サスペンドからのリモートウェイクアップ

サスペンド時には、(PHY をサスペンド状態としている場合)PHY から供給される UDC2AB と UDC2 へのクロック (CLK_U) が停止しています。この状態で、UDPWCTL<phy_remote_wkup>を 1 にセットすることにより、UDC2AB は UDC2 への udc2_wakeup をアサートし、PHYSUSPEND_X 信号をデアサートします。一定期間後に PHY からのクロック (CLK_U) 出力が復帰して、クロックが供給されると UDC2 は自動的にレジューム動作を開始します。

(4) サスペンド、レジューム(ディスコネクト)時の信号動作

前項までの内容を踏まえて、サスペンド、レジューム(ディスコネクト)時の信号動作を図に示します。

USBホストからのレジューム(ディスコネクト)時で、CLK_Hを停止する場合は「図 3.16.13 サスペンド、レジューム信号動作(CLK_H停止時)」、「図 3.16.14 サスペンド、ディスコネクト信号動作(CLK_H停止時)」を、停止しない場合は「図 3.16.15 サスペンド、レジューム信号動作(CLK_H動作時)」を参照して下さい。また、UDC2ABからのリモートウェイクアップは「図 3.16.16 サスペンド、リモートウェイクアップ信号動作」を参照して下さい。

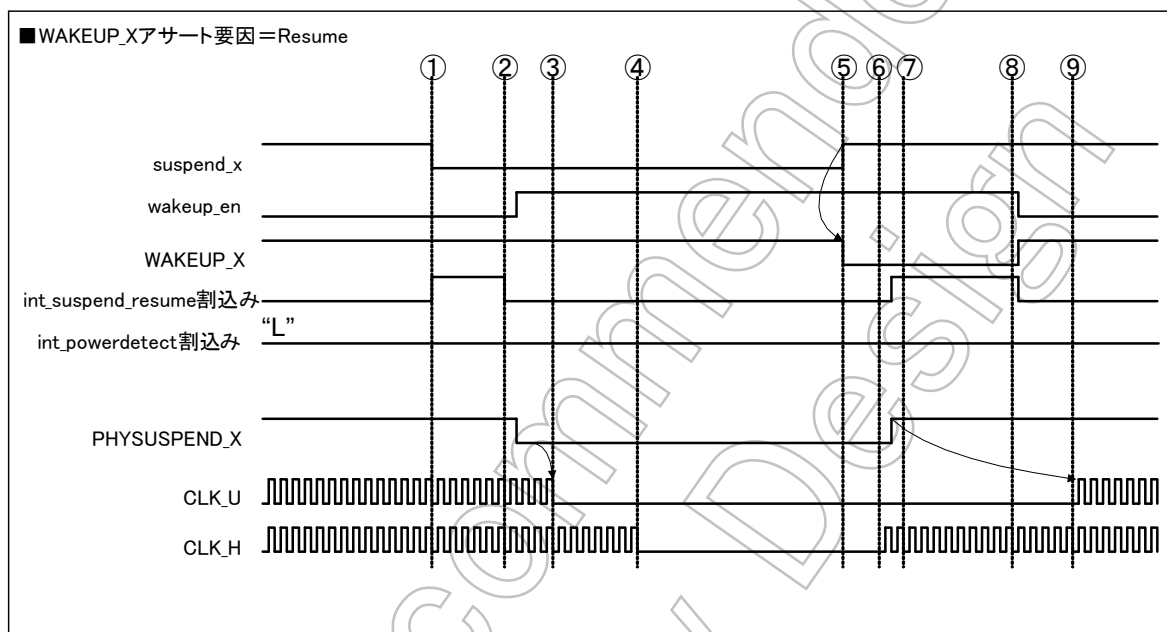


図 3.16.13 サスペンド、レジューム信号動作(CLK_H停止時)

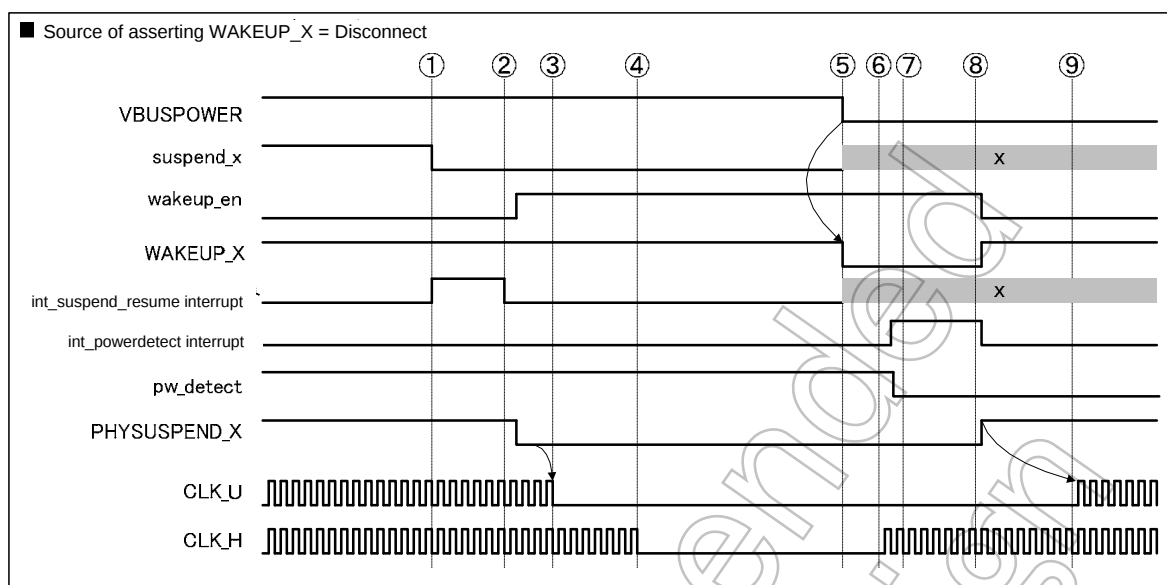


図 3.16.14 サスペンド、ディスコネクト信号動作(CLK_H停止時)

図 3.16.13、図 3.16.14の信号動作について以下に説明します。

USB Bus 上でサスペンド状態を検出することにより、int_suspend_resume 割り込みが発生する。

int_suspend_resume 割り込みによって、ソフトウェアで割り込み要因のクリアと、UDPWCTL<phy_suspend>を 1 にセットする。

phy_suspend ビットがセットされることで、PHYSUSPEND_X 出力信号が 0 にアサートされ、CLK_U の供給が停止する。

ソフトウェアによって UDPWCTL<wakeup_en>を 1 にセット後、CLK_H を停止させることが可能となる。

USB バス上で、Resume を検出するか、またはディスコネクト(VBUS が切断)すること、WAKEUP_X 出力信号が非同期で 0 にアサートされる。

WAKEUP_X 出力信号によって、CLK_H の供給を開始させる。CLK_H が供給されることで、int_suspend_resume 割り込み、または int_powerdetect 割り込みが発生する。

(suspend_x の立ち上がりを検出した場合、PHYSUSPEND_X 出力信号が自動的に 1 にデアサートされる。)

割り込みアサートから 2.5μs 以上経過してから(VBUS 切断時に信号が安定するまでの期間)、UDPWCTL<pw_detect>を確認する。

外部割り込みにより判断 → -a へ …… WAKEUP_X アサート原因は Resume

→ -b へ …… WAKEUP_X アサート原因はディスコネクト

<Resume 時>

- a ソフトウェアは割り込み要因と<wakeup_en>をクリアし、WAKEUP_X 出力信号をデアサートさせる。
- a サスペンドからの復帰

<ディスコネクト時>

- b ソフトウェアにて<phy_suspend>を 0 にクリアして、PHYSUSPEND_X 出力信号をデアサートする。また、割り込み要因と<wakeup_en>をクリアし、WAKEUP_X 出力信号をデアサートさせる。
- b ソフトウェアにて UDPWCTL<pw_reseth>をセットし、UDC2AB を初期化する。

注) 本来、UDC2AB は Wakeup 信号をアサートする機能を持っていますが、本 LSI ではサポートしていません。

Not Recommended
for New Design

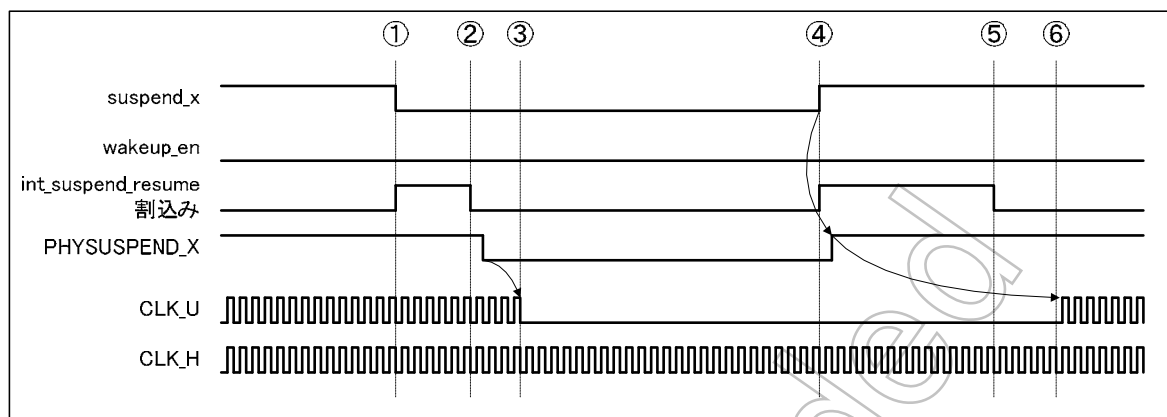


図 3.16.15 サスペンド、レジューム信号動作(CLK_H 動作時)

USB Bus 上でサスペンド状態を検出することにより、int_suspend_resume 割り込みが発生する。

int_suspend_resume 割り込みによって、ソフトウェアは割り込み要因のクリアと、UDPWCTL<phy_suspend>を 1 にセットする。

<phy_suspend>がセットされることで、PHYSUSPEND_X 出力信号がアサートされ、CLK_U の供給が停止する。

USB バス上で、Resume を検出することで、int_suspend_resume 割り込みが発生する。

また、suspend_x の立ち上がりを検出することによって、PHYSUSPEND_X 出力信号が 1 にデアサートされる。

int_suspend_resume 割り込みによって、ソフトウェアは割り込み要因のクリアを行う。

PHYSUSPEND_X 出力信号がデアサートされることで、CLK_U の供給が再開される。

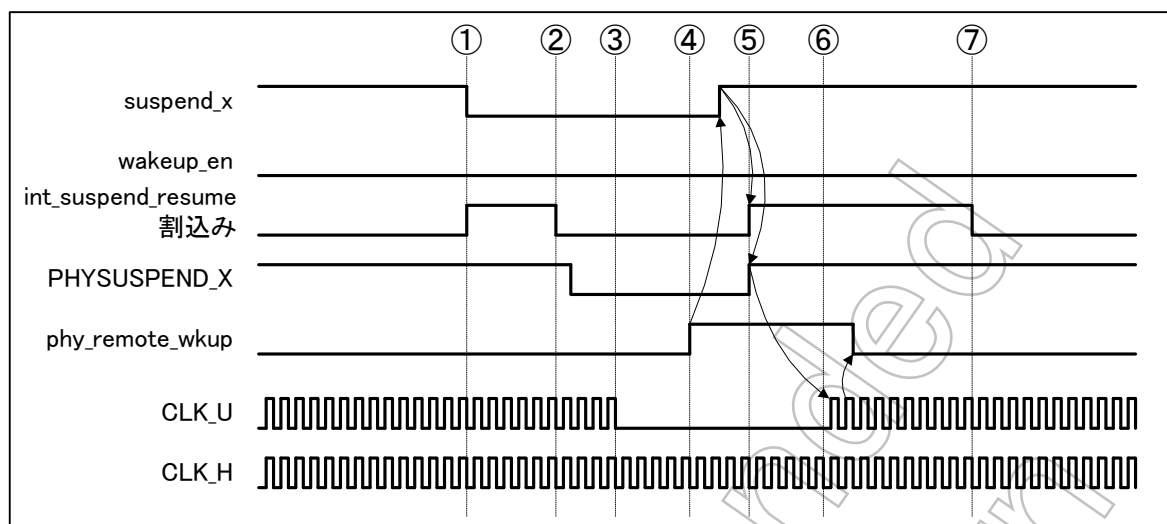


図 3.16.16 サスペンド、リモートウェイクアップ信号動作

USB Bus 上でサスペンド状態を検出することにより、int_suspend_resume 割り込みが発生する。

int_suspend_resume 割り込みによって、ソフトウェアは割り込み要因のクリアと、UDPWCTL<phy_suspend>を1にセットする。

<phy_suspend>がセットされることで、PHYSUSPEND_X 出力信号が0にアサートされ、CLK_U の供給が停止する。

リモートウェイクアップを要求する場合は、UDPWCTL<phy_remote_wkup>を1にセットする。<phy_remote_wkup>をセットすることで、UDC2 から USB バス上にリモートウェイクアップ要求が行われる。また、suspend_x が非同期で1にデアサートされる。

suspend_x がデアサートされることで、int_suspend_resume 割り込みが発生し、PHYSUSPEND_X 出力信号が1にデアサートされる。

PHYSUSPEND_X 出力信号がデアサートされることで、CLK_U が再開される。

<phy_remote_wkup> は自動的にクリアされる。

int_suspend_resume 割り込み要因のクリアを行う。

3.16.3 UDC2 概要

UDC2 は Universal Serial Bus への USB ファンクションの接続をコントロールする機能を持つコアです。UDC2 は USB プロトコルを自動処理し、PHY 側インターフェースは UTMI によりアクセス可能となっています。

コアの主な機能と特長は以下です。

- Universal Serial Bus Specification Rev.2.0 をサポート
- High-Speed(HS)/Full-Speed(FS)をサポート (Low-Speed は非対応)
- Chirp 対応
- USB プロトコル処理
- SOF/USB_RESET/SUSPEND/RESUME の検出
- パケット ID の生成およびチェック
- データ同期ビット(DATA0/DATA1/DATA2/MDATA)の生成およびチェック
- CRC5 チェック、CRC16 の生成およびチェック
- PING 対応
- 4 種類(Control/Interrupt/Bulk/Isochronous)の転送モードをサポート
- 4 EndPoint までサポート
- デュアルパケットモード対応(EndPoint0 は除く)
- EndPoint1~3 については、FIFO へ直接アクセス可能(Endpoint-I/F)
- USB2.0 Transceiver Macrocell Interface (UTMI) 対応(16bit@30MHz)

3.16.3.1 UDC2 内部ブロック構成

UDC2 のブロック構成および各ブロックの概略をを以下に示します。

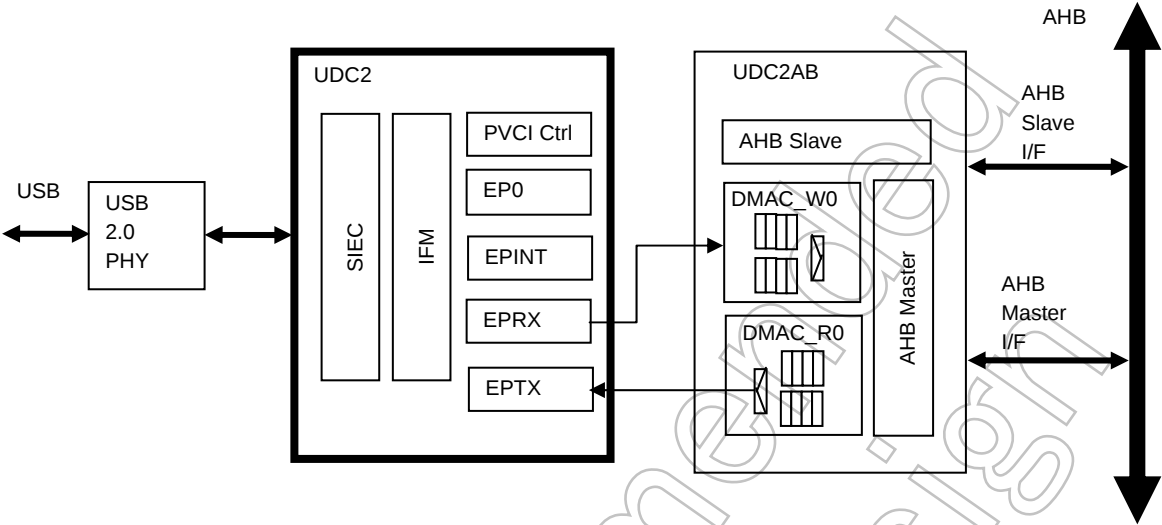


図 3.16.17 UDC2 ブロック図

(1) SIEC(Serial Interface Engine Control)ブロック

このブロックでは、USB におけるプロトコルの管理を行います。主な機能を以下に示します。

- PID のチェック/生成。
- CRC チェック/生成。
- デバイスアドレスのチェック。
- 転送スピード(HS/FS)の管理。
- PHY のコントロール(転送 Speed(HS/FS)、モード、etc.)。
- テストモードの生成。

(2) IFM ブロック

このブロックでは、SIEC と Endpoint の制御を行います。主な機能を以下に示します。

- OUT-Token 受信時に、受信データを該当する Endpoint へライト。
- IN-Token 受信時に、送信データを該当する Endpoint からリード。
- UDC2.0 のステータス制御/管理。

(3) PVCIF ブロック

このブロックでは、IFM と外部レジスタアクセスバス(PVCI)とのリード/ライトを制御します。

PVCI バスは UDC2AB を経由してアクセスします。

(4) EP0 ブロック

このブロックでは、Control 転送時の送受信データをコントロールします。Control 転送の DATA-Stage でデータを送受信する場合、このブロック内の FIFO へ PVCI-I/F よりアクセスして下さい。

(5) EPx ブロック

このブロックでは、EPx(x=1,2,3)の送受信データをコントロールします。Endpoint-I/F により FIFO へ直接アクセスすることが可能です。Endpoint-I/F では、バースト転送が可能です。

なお、Endpoint は送信用 Endpoint と、受信用 Endpoint の 2 種類があります。Endpoint の方向(送信/受信)についてはハードで固定となります。

3.16.3.2 各フラグ仕様

UDC2 コアは、USB 上での各種イベント発生時に、フラグとして出力しています。この章では、各フラグについて説明します。

(1) USB_RESET

USB_RESET 受信期間中、“H”をアサートします。UDC2 は、USB_RESET を受信することにより、Default-State に戻りますので、アプリケーションも Default-State に戻る必要があります。

UDC2 が Full-Speed 動作時は USB バス上の SE0 を 2.5 μ s 以上認識した時にこのフラグをアサートします。High-Speed 動作時は SE0 を 3ms 以上認識した時に、USB_RESET かサスペンド状態かを判断した後にアサートします。その後、UDC2 が Chirp-K を約 1.5ms ドライブした後に、以下の 2 種類の状態のどちらかを認識するとデアサートします。

1. ホストからの Chirp (K-J-K-J-K-J) を認識する。
2. ホストからの Chirp(K-J-K-J-K-J) を認識しない状態で 2ms 以上経過する。

注) ホストが Chirp を開始する時間、Chirp-K、Chirp-J のドライブ時間はホストに依存しますが、USB_RESET フラグのアサート期間は 1.74ms~3.5ms 程度となります。

(2) INT_SETUP

Control 転送において、Setup-Token 受信後、“H”をアサートします。ソフトウェアはこの割込みを認識したら、Setup-Data 格納レジスタ(8 バイト)をリードし、リクエストの判断をして下さい。なお、この割込みは INT レジスタの該当 bit(bit0)に 1 をライトすることによりデアサートされます。割込みを認識した時点で INT レジスタのクリアを行うようにして下さい。

(3) INT_STATUS_NAK

Control 転送において、UDC2 が DATA-Stage を処理中(“Setup_Fin”コマンド発行前)に、ホストが STATUS-Stage へ移行してパケットを送信してくると、UDC2 は“NAK”を返信しこのフラグを“H”にアサートします。ソフトウェアはこの割込みを認識したら、Command レジスタにより“Setup_Fin”コマンドを発行し、UDC2 の STATUS-Stage を終了させる必要があります。なお、この割込みは INT レジスタの該当 bit(bit1)に 1 をライトすることによりデアサートされます。割込みを認識した時点で INT レジスタのクリアを行うようにして下さい。

(4) INT_STATUS

Control 転送において、STATUS-Stage を正常に終了後、“H”をアサートします。なお、この割込みは INT レジスタの該当 bit(bit2)に 1 をライトすることによりデアサートされます。割込みを認識した時点で INT レジスタのクリアを行うようにして下さい。

(5) INT_EP0

Control 転送の DATA-Stage において、“ACK”を送受信した際(正常にトランザクションが終了した際に)“H”をアサートします。なお、この割込みは INT レジスタの該当 bit(bit5)に 1 をライトすることによりデアサートされます。割込みを認識した時点で INT レジスタのクリアを行うようにして下さい。

(6) INT_EP

Endpoint0 以外の Endpoint において、“ACK”を送受信した際(正常にトランザクションが終了した際)に“H”をアサートします。その際、INT_EP レジスタを確認することにより、どの Endpoint への転送かを判断することができます。なお、この割込みは INT レジスタの該当 bit(bit6)に 1 をライトするか、INT_EP レジスタのセットされている全 bit に 1 をライトすることによりデアサートされます。割込みを認識した時点で INT レジスタのクリアを行うようにして下さい。

(7) INT_RX_ZERO

Zero-Length データを受信時に“H”がアサートされます。ただし、Control 転送では DATA-Stage での Zero-Length データ受信時にのみアサートされます。STATUS-Stage での Zero-Length データ受信時にはアサートされません。どの Endpoint に受信したかについては、Command レジスタの bit[11:8]をリードするか、INT_RX_DATA0 レジスタを確認することにより判断できます。なお、この割込みは INT レジスタの該当 bit(bit3)に 1 をライトするか、INT_RX_DATA0 レジスタのセットされている全 bit に 1 をライトすることによりデアサートされます。割込みを認識した時点で INT_RX_DATA0 レジスタのクリアを行うようにして下さい。

(8) INT_SOF

SOF 受信時に“H”をアサートします。なお、この割込みは INT レジスタの該当 bit(bit4)に 1 をライトすることによりデアサートされます。割込みを認識した時点で INT レジスタのクリアを行うようにして下さい。

SOF はフレーム(μフレーム)の開始を示すパケットです。Full-Speed 転送では 1ms ごと、High-Speed 転送では 125μs ごとにホストからデバイスへ送信されます。

(9) INT_NAK

Endpoint0 以外の Endpoint において、NAK を送信するとアサートします。その際、INT_NAK レジスタを確認することにより、どの Endpoint が NAK を送信したか判断することができます。なお、この割り込みは INT レジスタの該当ビット(bit7)に 1 をライトするか、INT_NAK レジスタのセットされている全ビットに 1 をライトすることによりデアサートされます。デフォルトでは NAK を送信してもこのフラグをアサートしないため、このフラグを使用する際には INT_NAK_MASK レジスタの該当 Endpoint に 0 をライトして、マスクを解除して下さい。

3.16.3.3 レジスタ構成

UDC2 には、以下のレジスタが用意されています。

- デバイスステータス

Address-State レジスタ

Frame レジスタ

USB-Testmode レジスタ

Command レジスタ

- Setup-Data 格納

bRequest-bmRequestType レジスタ

wValue レジスタ

wIndex レジスタ

wLength レジスタ

- 割込み制御

INT レジスタ

INT_EP レジスタ

INT_EP_MASK レジスタ

INT_RX_DATA0 レジスタ

INT_NAK レジスタ

INT_NAK_MASK レジスタ

- EP0 制御ステータス

EP0_MaxPacketSize レジスタ

EP0_Status レジスタ

EP0_Datasize レジスタ

EP0_FIFO レジスタ

- EPx 制御ステータス

EPx_MaxPacketSize レジスタ

EPx_Status レジスタ

EPx_Datasize レジスタ

EPx_FIFO レジスタ

レジスタマップを表 3.16.2. に示します。

表 3.16.2 レジスタマップ

Address =(0xF440_0000)

	Register Name	Address (base+)	Description
UDC2 *2), *3)	UD2ADR	0x0200	UDC2 Address-State Register
	UD2FRM	0x0204	UDC2 Frame Register
	UD2TMD	0x0208	UDC2 USB-Testmode Register
	UD2CMD	0x020C	UDC2 Command Register
	UD2BRQ	0x0210	UDC2 bRequest-bmRequestType Register
	UD2WVL	0x0214	UDC2 wValue Register
	UD2WIDX	0x0218	UDC2 wIndex Register
	UD2WLGTH	0x021C	UDC2 wLength Register
	UD2INT	0x0220	UDC2 INT Register
	UD2INTEP	0x0224	UDC2 INT_EP Register
	UD2INTEPSK	0x0228	UDC2 INT_EP_MASK Register
	UD2INTRX0	0x022C	UDC2 INT_RX_DATA0 Register
	UD2EP0MSZ	0x0230	UDC2 EP0_MaxPacketSize Register
	UD2EP0STS	0x0234	UDC2 EP0_Status Register
	UD2EP0DSZ	0x0238	UDC2 EP0_Datasize Register
	UD2EP0FIFO	0x023C	UDC2 EP0_FIFO Register
	UD2EP1MSZ	0x0240	UDC2 EP1_MaxPacketSize Register
	UD2EP1STS	0x0244	UDC2 EP1_Status Register
	UD2EP1DSZ	0x0248	UDC2 EP1_Datasize Register
	UD2EP1FIFO	0x024C	UDC2 EP1_FIFO Register
	UD2EP2MSZ	0x0250	UDC2 EP2_MaxPacketSize Register
	UD2EP2STS	0x0254	UDC2 EP2_Status Register
	UD2EP2DSZ	0x0258	UDC2 EP2_Datasize Register
	UD2EP2FIFO	0x025C	UDC2 EP2_FIFO Register
	Reserved	0x0260~ 0x03FC	
	UD2INTNAK	0x0330	UDC2 INT_NAK Register
	UD2INTNAKMSK	0x0334	UDC2 INT_NAK_MASK Register
	Reserved	0x0338 ~ 0x03FC	

次項より、UDC2 内各レジスタの詳細説明を行います。
各ビットの説明の意味は以下のようになります。

(例)

EPx_Datasize (EPx_Datasize register)

Address =(0xF440_0000)+ (0x0000)

Bit	Bit Symbol (注 1)	Type (注 2)	Reset Value (注 3)	Description
[31:11]	—	—	Undefined	Read as undefined.
[10:0]	size[10:0]	RO	0y000000000000, (-)(-)(-)(-)(-)(-) (-)(-)(-)(-)	EPx_FIFO に格納されている有効データバイト数を示します。なお、Dual パケットモード時には、最初にアクセスするパケットのデータバイト数を示します。

注 1) Bit Symbol

そのビットの名称です。

"—"となっているのは Reserved ビットとなり、ライトは出来ません。リード時は 0 が読み出されます。

注 2) レジスタ属性の説明

RO : リードオンリー。ライトは無視されます。

WO : ライトオンリー。リード時は 0 となります。

R/W : リード/ライト。

注 3) 初期値

"Reset Value"はそのビットのリセット後の初期値(1 or 0)です。また、"USB_RESET"後の初期値は "("内"で表され、"USB_RESET" ではリセットされないビットに関しては、"(-)"を記しています。

次節より、各レジスタ詳細について説明します。

(1) デバイスステータス レジスタ

1. UD2ADR (Address-State register)

Address =(0xF440_0000)+ (0x0200)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	—	—	Undefined	Read as undefined. Write as zero.
[15]	stage_err	RO	0y0	Control 転送が正常に STATUS-Stage まで終了したかを示します。 0y0: 下記条件以外 0y1: DATA-Stage/STATUS-Stage 時に Setup-Token を受信、あるいは“STALL”送信
[14]	ep_bi_mode	R/W	0y0, (-)	Endpoint をドライバとして双方向に使用するかを選択します。(注 2) 0y0: 単方向 0y1: 双方向
[13:12]	cur_speed[1:0]	RO	0y01,(注 1)	現在の USB バス上での転送モードを示します。 0y00: (Reserved) 0y01: Full-Speed、 0y10: High-Speed 0y11: (Reserved)
[11]	Suspend	RO	0y0	UDC2 がサスペンド状態かどうかを示します。 0y0: Normal 0y1: suspend
[10]	Configured	R/W	0y0	現在の UDC2 のデバイスステートを設定します。
[9]	Addressed	R/W	0y0	0y001: default(Default/Address state にいる時に Set_address リクエストにて DeviceAddress=0 を指定された時にセット(USB_RESET を受信時にはハードにてセットされます))
[8]	Default	R/W	0y0,(1)	0y010: addressed(Set_address リクエストが正常終了時、Address/Configured state にいる時に Set_configuration リクエストにて ConfigurationVallue=0 を指定された時にセット) 0y100: configured(Set_config リクエストを受信時にセット)
[7]	—	—	Undefined	Read as undefined. Write as zero.
[6:0]	dev_adr[6:0]	R/W	0y0000000	ホストから割り振られたデバイスアドレスを設定します。

注 1) cur_speed[1:0](bit[13:12]) の USB_RESET 後の初期値は、Chirp シーケンスが成功していれば 0y10 (High-Speed)に、失敗していれば 0y01(Full-Speed)となります。

注 2) TMPA900CM 製品では EP0 : 単方向/双方向。EP1 , EP2 と EP3 : 単方向のみ

(個別説明)

a. <stage_err>

Control 転送が正常に STATUS-Stage まで終了したかを示します。

DATA-Stage/STATUS-Stage 時に Setup-Token を受信、あるいは“STALL”送信時に 1 がセットされます。セットされた場合には、次の Control 転送が正常に終了した場合にクリアされます。

0y0: 下記条件以外

0y1: DATA-Stage/STATUS-Stage 時に Setup-Token を受信、あるいは“STALL”送信

b. <ep_bi_mode>

Endpoint をドライバとして双方向に使用するかを選択します。この bit を 1 にセットすることにより、USB 通信上において一つの Endpoint Number を双方向に扱うことが出来ます。

0y0: 単方向

0y1: 双方向

c. <cur_speed[1:0]>

現在の USB バス上での転送モードを示します。

0y00: Reserved

0y01: Full-Speed

0y10: High-Speed

0y11: Reserved

d. <suspend>

UDC2 がサスペンド状態かどうかを示します。

0y0: Normal

0y1: suspend

e. <configured>,<addressed>,<default>

現在の UDC2 のデバイスステートを設定します。ホストからのリクエスト受信に併せて、セットして下さい。なお、同時に複数の bit に 1 をセットしないように注意願います。

0y001: default(Default/Address state にいる時に Set_address リクエストにて DeviceAddress=0 を指定された時にセット(USB_RESET を受信時にはハードにてセットされます))

0y010: addressed(Set_address リクエストが正常終了時、Address/Configured state にいる時に Set_configuration リクエストにて ConfigurationValue=0 を指定された時にセット)

0y100: configured(Set_config リクエストを受信時にセット)

f. <dev_adr[6:0]>

ホストから割り振られたデバイスアドレスを設定します。Set_address が正常終了後 (STATUS_Stage 正常終了後) にデバイスアドレス値をセットして下さい。

2. UD2FRM (Frame register)

Address =(0xF440_0000)+ (0x0204)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined. Write as zero.
[15]	create_sof	R/W	0y0	ホストからの SOF がバスエラーにより取れない場合、SOF フラグを内部生成するかを設定します。 0y0: 生成しない 0y1: 生成する
[14]	–	–	Undefined	Read as undefined. Write as zero.
[13:12]	f_status[1:0]	RO	0y10,(-)(-)	フレーム番号の状態を示します。 0y00: Before 0y01: Valid 0y10: Lost
[11]	–	–	Undefined	Read as undefined. Write as zero.
[10:0]	frame[10:0]	RO	0y000000000000	sof 受信時のフレーム番号を示します。

(個別説明)

a. <create_sof>

ホストからの SOF がバスエラーにより取れない場合、SOF フラグを内部生成するかを設定します。Isochronous 転送を使用し、sof によりフレーム同期をとりたい場合にセットして下さい。イネーブルにすることにより、内部のフレーム時間カウンタを動作させることにより、SOF-Token を正常に受信出来なかった場合にも SOF フラグを出力します。

0y0: 生成しない

0y1: 生成する

b. <f_status[1:0]>

フレーム番号の状態を示します。

0y00: Before : Create_sof イネーブル時にマイクロ SOF/SOF を受信から 1frame-time(HS:125 μ s/FS:1ms) 経過してもマイクロ SOF/SOF を受信しなかったときにセットされます。フレームレジスタには 1 つ前のマイクロ SOF/SOF で受信したときのフレーム番号がセットされています。

0y01: Valid : マイクロ SOF/SOF を受信するとセットされます。Frame レジスタには有効なフレーム番号がセットされていることを示しています。

0y10: Lost : ホストが管理しているフレーム番号と Frame レジスタの値とが同期が取れていない状態示しています。そのため以下の 2 つの場合にセットされます。

1. システムリセット後あるいはサスペンド時
2. Create_sof イネーブル時に前回マイクロ SOF/SOF を受信してから 2frame-time(HS:125 $\times$ 2 μ s/FS:1 $\times$ 2ms)以上経過しても次のマイクロ SOF/SOF を受信しなかった時

ただし High-Speed の場合マイクロ SOF のフレーム番号は 8 回連続で同じ値が送られるため、Lost 状態でもホストが送るフレーム番号とフレームレジスタの値と同期が取れるように見えることがあります。しかし、フレーム番号とそのフレーム番号が来た回数まで考慮すると同期が取れていませんので注意してください。また、Create_sof がディセーブル時はシステムリセット後かサスペンド時しか Lost には遷移しません。

c. <frame[10:0]>

sof 受信時のフレーム番号を示します。f_status が“valid”の時有効となります。f_status が“before”あるいは“lost”の時は正しい値がセットされていませんので、使用しないで下さい。

Not Recommended
for New Design

3. UD2TMD (USB-Testmode register)

Address =(0xF440_0000)+ (0x0208)

Bit	Bit Symbol	Type	Reset Value	Description
[31:13]	–	–	Undefined	Read as undefined. Write as zero.
[12]	packet	RO	0y0	現在設定されているテストモードを示します。 0y0001: test_j 0y0010: test_k 0y0100: se0_nak 0y1000: test_packet
[11]	se0_nak	RO	0y0	
[10]	test_k	RO	0y0	
[9]	test_j	RO	0y0	
[8]	–	–	Undefined	Read as undefined. Write as zero.
[7:0]	t_sel[7:0]	R/W	0x00	Test-Mode を設定します。

(個別説明)

- a. <packet>, <se0_nak>, <test_k>, <test_j>,

現在設定されているテストモードを示します。

0y0001: test_j

0y0010: test_k

0y0100: se0_nak

0y1000: test_packet

- b. <t_sel[7:0]>

Test-Mode を設定します。Set_Feature で指定された TestModeSelectors の値を設定して下さい。

4. UD2CMD (Command register)

Address =(0xF440_0000)+ (0x020C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined. Write as zero.
[15]	int_toggle	R/W	0y0	Interrupt-IN 転送時に、Handshake 未受信時の DATA-PID をトグルする様に設定します。 0y0: 未受信時はトグルしない 0y1: 未受信時もトグルする
[14:12]	–	–	Undefined	Read as undefined. Write as zero.
[11:8]	rx_nullpkt_ep[3:0]	RO	0y0000, (-)(-)(-)(-)	Zero-Length データ受信時に、受信した Endpoint を示します。
[7:4]	ep[3:0]	R/W	0y0000	発行されるコマンドが有効となる Endpoint を設定します。
[3:0]	com[3:0]	R/W	0y0000	ep[3:0]にて選択した Endpoint に対して発行するコマンドを設定します。 0x0: Reserved 0x1: Setup_Fin 0x2: Set_DATA0 0x4: EP_Stall 0x5: EP_Invalid 0x6: Reserved 0x7: EP_Disable 0x8: EP_Enable 0x9: All_EP_Invalid 0xA: USB_Ready 0xB: Setup_Received 0xC: EP_EOP 0xD: EP_FIFO_Clear 0xE: EP_TX_0DATA 0xF: Reserved

(個別説明)

a. <int_toggle>

Interrupt-IN 転送時に、Handshake 未受信時の DATA-PID をトグルする様に設定します。

0y0: 未受信時はトグルしない

0y1: 未受信時もトグルする

b. <rx_nullpkt_ep[3:0]>

Zero-Length データ受信時に、受信した Endpoint を示します。“int_rx_zero”フラグがアサートされた時、この bit をリードしてどの Endpoint に対するものかを確認してください。このレジスタの値は一度 Zero-Length データを受信し Endpoint 番号を保持すると、次に Zero-Length データを受信するかハードウェアリセット(reset_x=0)まで保持します。また、OUT 方向の Endpoint が複数ある場合には、Zero-Length データを受信毎にこの bit を更新してしまいます。その場合は INT_RX_DATA0 レジスタを使用することにより、どの Endpoint に受信したかを確認できます。

c. <ep[3:0]>

発行されるコマンドが有効となる Endpoint を設定します。(存在しない Endpoint は指定しないで下さい。)

d. <com[3:0]>

ep[3:0]にて選択した Endpoint に対して発行するコマンドを設定します。

0x0: Reserved

指定しないで下さい。

0x1: Setup_Fin

(EP0 にのみ発行して下さい。)

Control 転送の DATA-Stage 終了を設定するコマンドです。UDC2 は、このコマンドが発行されるまで STATUS-Stage に対して“NAK”を返信しつづけますので、DATA-Stage 終了時あるいは INT_STATUS_NAK 受信時にこのコマンドを発行して下さい。

注) Control-WR では DATA-Stage で受信した全データをリード後に“Setup_Fin”コマンドを発行して下さい。

0x2: Set_DATA0

(EP0 を除く EP に対し発行可能。EP0 へは発行しないで下さい。) Endpoint のトグルをクリアするコマンドです。通常の転送時のトグル更新は UDC2 により自動的に行われますが、ソフトからクリアする必要がある場合はこのコマンドを発行して下さい。

0x3: EP_Reset

(すべての EP に対し発行可能。)

Endpoint のデータおよびステータスをクリアするコマンドです。Set_Configuration、Set_Interface の Endpoint 設定時、Clear_Feature による Endpoint のリセット時等、Endpoint をリセットしたい場合にこのコマンドを発行して下さい。なお、このコマンドによりリセットされるのは、

1. EP0/EPx_Status レジスタの bit13-12(toggle)を DATA0 ヘクリア
2. EP0/EPx_Status レジスタの bit11-9(status)を Ready ヘクリア
3. EP0/EPx_MaxPacketSize レジスタの bit12(dset)および EP0/EPx_Datasize レジスタをクリア
4. EP0/EPx_MaxPacketSize レジスタの bit15(tx_0data)をクリア
5. EP0/EPx_MaxPacketSize レジスタの bit15(tx_0data)をクリア

の 5 点です。

UDC2 は全ての転送においてハードによるトグルの制御を行っています。各 Endpoint の転送が行われている時にこのコマンドを発行すると、該当 Endpoint のトグルもクリアされますのでホストとの同期がとれなくなる可能性があります。前述にありますリクエスト受信時のように、ホストとの同期がとれる時にコマンドを発行して下さい。

0x4: EP_Stall

(すべての EP に対し発行可能。)

Endpoint のステータスを“Stall”にセットするコマンドです。Set_Feature による Endpoint の Stall 時等、Endpoint のステータスを“Stall”にしたい場合にこのコマンドを発行して下さい。このコマンドを発行することにより、設定された Endpoint については“STALL”を返信するようになります。ただし EP0 の Stall 状態は Setup-Token 受信時にクリアされます。

Isochronous 転送では Handshake 無しで転送が行われますので、Isochronous 転送を使用中の Endpoint に対してはこのコマンドは発行しないで下さい。(EPx_Status レジスタの bit[3:2]t_type で)Isochronous 転送を設定している Endpoint に対してこのコマンドを発行した場合でも、“STALL”は返信しません。

0x5: EP_Invalid

(EP0 を除く EP に対し発行可能。EP0 へは発行しないで下さい。)

Endpoint のステータスを“Invalid”にセットするコマンドです。Set_Configuration、Set_Interface による設定時、使用しない Endpoint を使用禁止に設定する場合にこのコマンドを発行して下さい。このコマンドを発行することにより、設定された Endpoint については無応答となります。各 Endpoint の転送が行われている時にはこのコマンドを発行しないで下さい。

0x6: Reserved	指定しないで下さい。
0x7: EP_Disable	(EP0 を除く EP に対し発行可能。EP0 へは発行しないで下さい。) Endpoint を Disable にするコマンドです。このコマンドを発行することにより、設定された Endpoint は、“NAK”を返信するようになります。Isochronous 転送では Handshake 無しで転送が行われますので、Isochronous 転送を使用中の Endpoint に対してはこのコマンドは発行しないで下さい。(EPx_Status レジスタの bit[3:2]t_type で)Isochronous 転送を設定している Endpoint に対してこのコマンドを発行した場合でも、“NAK”は返信しません。
0x8: EP_Enable	(EP0 を除く EP に対し発行可能。EP0 へは発行しないで下さい。) Endpoint を Enable にするコマンドです。“EP_Disable”コマンドによる Disable 状態を解除する際に、このコマンドを発行して下さい。
0x9: All_EP_Invalid	(EP の設定は無効。) EP0 以外の全 Endpoint のステータスを“Invalid”にセットするコマンドです。“EP_Invalid”コマンドを全 Endpoint に行いたい場合、このコマンドを発行して下さい。“EP_Invalid”コマンドと同様に Set_Configuration、Set_Interface 処理時に発行して下さい。
0xA: USB_Ready	(EP0 にのみ発行して下さい。) USB ケーブルへの接続をするためのコマンドです。ケーブルに接続されたことを確認後、ホストとの通信が可能になった時点でこのコマンドを発行して下さい。このコマンドが発行されて初めて DDP の Pull-Up をし、ホストへケーブルコネクタ状態を知らせます。 なお、このコマンドを発行すると UDC2 のデバイスステート (Address-state レジスタの bit[10:8])は“Default”に設定されますので、このコマンドを発行する際は注意してください。
0xB: Setup_Received	(EP0 にのみ発行して下さい。) Control 転送の SETUP-Stage を認識したことを UDC2 へ知らせるためのコマンドです。INT_SETUP 割り込みを受け付けて、リクエストコードを認識した後にこのコマンドを発行して下さい。このコマンドが発行されるまで DATA-Stage/STATUS-Stage に対して“NAK”を返信しつづけますので、INT_SETUP 割り込み処理ルーチンの最後にこのコマンドを発行して下さい。
0xC: EP_EOP	(すべての EP に対し発行可能。) 送信データ書き込み終了を UDC2 へ知らせるためのコマンドです。最大転送バイト数(Endpoint の FIFO 容量か MaxPacketSize のうち小さいバイト数)よりも少ないバイト数を送信したい場合、このコマンドを発行して下さい。このコマンドを発行することにより Dataset フラグがセットされ、ホストからの IN-Token に対しデータを返信します。Zero-Length データ、MaxPacketSize のデータをセットする際には使用しないで下さい。
0xD: EP_FIFO_Clear	(すべての EP に対し発行可能。) Endpoint のデータをクリアするコマンドです。同時に EPx_MaxPacketSize レジスタの bit12(dset)、EPx_Datasize レジスタがクリアされます。Interrupt 転送等で、ホストへデータを送信する前に現在 FIFO へ格納されているデータをクリアし、最新のデータをセットしたい場合にこのコマンドを発行して下さい。Endpoint-I/F アクセス中にこのコマンドを発行すると、Endpoint の FIFO が正常にクリアされません。このコマンドを発行するときには Endpoint-I/F の ep_x_val を 0 にした状態で発行して下さい。

0xE: EP_TX_0DATA (すべての EP に対し発行可能。)

Endpoint に Zero-Length データをセットするコマンドです。Zero-Length データを送信したい場合、このコマンドを発行して下さい。Bulk-IN 転送等で転送の最後を示すために Zero-Length データを送る場合には EPx_Datasize レジスタを読み、0 になった(EPx の FIFO にデータが無くなった)ことを確認して、このコマンドをセットしてください。また、Endpoint-I/F からデータを書き込む場合書き込みが終了して、epx_val を 0 の状態にしてこのコマンドをセットして下さい。このコマンドをセットするとセットした Endpoint の EPx_MaxPacketSize レジスタの bit15(tx_0data)がセットされます。この tx_0data が 0 になったら次のデータをセットするようになして下さい。Isochronous-IN 転送では、Endpoint の FIFO にデータがセットされていない状態だと、IN-Token に対して自動で Zero-Length データを送信します。このコマンドは発行しないで下さい。

0xF: Reserved

指定しないで下さい。

以下のコマンドにつきましては、USB 転送実行中に発行された場合設定は保留され、USB 転送終了後に実行されます。なお、コマンドの保留は Endpoint 毎に行われます。

0x2: Set_DATA0

0x3: EP_Reset

0x4: EP_Stall

0x5: EP_Invalid

0x7: EP_Disable

0x8: EP_Enable

0x9: All_EP_Invalid

0xD: EP_FIFO_Clear

0xE: EP_TX_0DATA

このため、USB 転送実行中に同一 Endpoint に連続してコマンドが発行された場合、コマンドは上書きされ最後に発行したコマンドのみ有効となります。同一 Endpoint に連続してコマンドを発行する必要がある場合は、Epx_Status/Epx_Datasize レジスタをポーリングしてコマンドが有効になった事を確認してから発行して下さい。また、EP_Reset/EP_FIFO_Clear コマンドにて FIFO をクリアした直後に Endpoint-I/F アクセスを行う場合には、EPx_Datasize レジスタをポーリングして、コマンドが有効になった事を確認してから Endpoint-I/F アクセスを再開して下さい。

Endpoint0 につきましては、Setup-Token 受信後 Setup_Received コマンドが発行されるまでは、Endpoint0 に対する以下のコマンドは無効となります。

0x1: Setup_Fin

0x2: Set_DATA0

0x3: EP_Reset

0x4: EP_Stall

0xC: EP_EOP

0xD: EP_FIFO_Clear

0xE: EP_TX_0DATA

EPx へ“EP_Stall”コマンドをセットすると、EPx_Status レジスタの bit[11:9](status)に “Stall”がセットされます。また、EP_Disable をセットすると EPx_Status レジスタの bit8(disable)に 1 がセットされます。この EP_Stall と EP_Disable の 2 種類のコマンドを同じ EPx へセットし、status が “Stall”に、disable=1 になった場合、転送では “STALL”が送信されます。

EPx へ“EP_Invalid”コマンドをセットすると、EPx_Status レジスタの status に “Invalid”がセットされます。EP_Invalid と EP_Disable の 2 種類のコマンドを同じ EPx へセットし、status が “Invalid”に、disable= 1 になった場合、転送では無反応になります。

EPx_Status レジスタの disable=1 で、EPx_MaxPacketSize レジスタの bit15(tx_0data)=1 の場合には、転送では Zero-Length データが 1 回送信されます。Zero-Length データの転送が成功した後に、“NAK”が送信されるようになります。

(2) Setup-Data 格納レジスタ

これらのレジスタは、Setup-Token 受信毎に、受信した Setup-Data を上書きします。
INT_SETUP 割込みが発生したら、これらのレジスタをリードし、リクエストの種類を判断して下さい。

1. UD2BRQ (bRequest-bmRequestType register)

Address = (0xF440_0000) + (0x0210)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined.
[15:8]	request[7:0]	RO	0x00	Setup-Token で受信した 2 バイト目のデータ (bRequest フィールド) を示します。
[7]	dir	RO	0y0	Setup-Token で受信した 1 バイト目のデータ (bRequestType フィールド) を示します。 Control 転送の方向 0y0: Control-WR 転送 0y1: Control-RD 転送
[6:5]	req_type[1:0]	RO	0y00	リクエストの種類 0y00: スタンダードリクエスト 0y01: クラスリクエスト 0y10: ベンダーリクエスト 0y11: Reserved
[4:0]	recipient[4:0]	RO	0y00000	リクエストの受け取り先 0y00000: Device 0y00001: Interface 0y00010: Endpoint 0y00011: etc. 0y00100-0y11111: Reserved

(個別説明)

a. <request[7:0]>

Setup-Token で受信した 2 バイト目のデータ (bRequest フィールド) を示します。

b. <dir>

Setup-Token で受信した 1 バイト目のデータ (bRequestType フィールド) を示します。

Control 転送の方向

0y0: Control-WR 転送

0y1: Control-RD 転送

c. <req_type[1:0]>

リクエストの種類を設定します。

0y00: スタンダードリクエスト

0y01: クラスリクエスト

0y10: ベンダーリクエスト

0y11: Reserved

d. <recipient[4:0]>

リクエストの受け取り先を設定します。

0y00000: Device

0y00001: Interface

0y00010: Endpoint

0y00011: etc.

0y00100-0y11111: Reserved

Not Recommended
for New Design

2. UD2WVL (wValue register)

Address =(0xF440_0000)+ (0x0214)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined.
[15:8]	value[15:8]	RO	0x00	Setup-Token で受信した 4 バイト目のデータ (wValue(H)フィールド)を示します。
[7:0]	value[7:0]	RO	0x00	Setup-Token で受信した 3 バイト目のデータ (wValue(L)フィールド)を示します。

(個別説明)

a. <value[15:8]>

Setup-Token で受信した 4 バイト目のデータ (wValue(H)フィールド)を示します。

b. <value[7:0]>

Setup-Token で受信した 3 バイト目のデータ (wValue(L)フィールド)を示します。

3. UD2WIDX (wIndex register)

Address =(0xF440_0000)+ (0x0218)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined.
[15:8]	index[15:8]	RO	0x00	Setup-Token で受信した 6 バイト目のデータ(wIndex(H)フィールド)を示します。
[7:0]	index[7:0]	RO	0x00	Setup-Token で受信した 5 バイト目のデータ(wIndex(L)フィールド)を示します。

(個別説明)

a. <index[15:8]>

Setup-Token で受信した 6 バイト目のデータ(wIndex(H)フィールド)を示します。

b. <index[7:0]>

Setup-Token で受信した 5 バイト目のデータ(wIndex(L)フィールド)を示します。

4. UD2WLGTH (wLength register)

Address =(0xF440_0000)+ (0x021C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined.
[15:8]	length[15:8]	RO	0x00	Setup-Token で受信した 8 バイト目のデータ(wLength(H)フィールド)を示します。
[7:0]	length[7:0]	RO	0x00	Setup-Token で受信した 7 バイト目のデータ(wLength(L)フィールド)を示します。

(個別説明)

a. <length[15:8]>

Setup-Token で受信した 8 バイト目のデータ(wLength(H)フィールド)を示します。

b. <length[7:0]>

Setup-Token で受信した 7 バイト目のデータ(wLength(L)フィールド)を示します。

(3) 割込み制御レジスタ

1. UD2INT (INT register)

Address =(0xF440_0000)+ (0x0220)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined. Write as zero.
[15]	m_nak	R/W	0y0	“i_nak(bit7)”を INT_NAK 端子に出力するかどうかを設定します。 0y0: イネーブル(出力する) 0y1: ディセーブル(出力しない)
[14]	m_ep	R/W	0y0	“i_ep(bit6)”を INT_EP 端子に出力するかどうかを設定します。 0y0: イネーブル(出力する) 0y1: ディセーブル(出力しない)
[13]	m_ep0	R/W	0y0	“i_ep0(bit5)”を INT_EP0 端子に出力するかどうかを設定します。 0y0: イネーブル(出力する) 0y1: ディセーブル(出力しない)
[12]	m_sof	R/W	0y0	“i_sof(bit4)”を INT_SOF 端子に出力するかどうかを設定します。 0y0: イネーブル(出力する) 0y1: ディセーブル(出力しない)
[11]	m_rx_data0	R/W	0y0	“i_rx_data0(bit3)”を INT_RX_ZERO 端子に出力するかどうかを設定します。 0y0: イネーブル(出力する) 0y1: ディセーブル(出力しない)
[10]	m_status	R/W	0y0	“i_status(bit2)”を INT_STATUS 端子に出力するかどうかを設定します。 0y0: イネーブル(出力する) 0y1: ディセーブル(出力しない)
[9]	m_status_nak	R/W	0y0	“i_status_nak(bit1)”を INT_STATUS_NAK 端子に出力するかどうかを設定します。 0y0: イネーブル(出力する) 0y1: ディセーブル(出力しない)
[8]	m_setup	R/W	0y0	“i_setup(bit0)”を INT_SETUP 端子に出力するかどうかを設定します。 0y0: イネーブル(出力する) 0y1: ディセーブル(出力しない)
[7]	i_nak	R/W	0y0	EP0 以外の各 EP が NAK を送信すると、1 にセットされます。
[6]	i_ep	R/W	0y0	EP0 を除く各 EP に対する転送が正常に終了時、1 にセットされます。
[5]	i_ep0	R/W	0y0	EP0 に対する転送が正常に終了時、1 にセットされます。
[4]	i_sof	R/W	0y0	SOF-token 受信時、あるいは create_sof モードで 1frame-time カウント時に 1 にセットされます。
[3]	i_rx_data0	R/W	0y0	Zero-Length データを受信時に 1 にセットされます。
[2]	i_status	R/W	0y0	EP0 における Control 転送において STATUS-Stage 正常終了時、1 にセットされます。
[1]	i_status_nak	R/W	0y0	EP0 における Control-RD 転送において、STATUS-Stage のパケット受信時に“NAK”を返信した場合、1 にセットされます。
[0]	i_setup	R/W	0y0	EP0 における Control 転送において、Setup-token 受信時に 1 にセットされます。

注)下位バイト(bit7-0)は、該当 bit に 1 をライトすることによりクリアされます。

(個別説明)

a. <m_nak>

“i_nak(bit7)”を INT_NAK 端子に出力するかどうかを設定します。

0y0: イネーブル(出力する)

0y1: ディセーブル(出力しない)

b. <m_ep>

“i_ep(bit6)”を INT_EP 端子に出力するかどうかを設定します。

0y0: イネーブル(出力する)

0y1: ディセーブル(出力しない)

- c. <m_ep0>
“i_ep0(bit5)”を INT_EP0 端子に出力するかどうかを設定します。
0y0: イネーブル(出力する)
0y1: ディセーブル(出力しない)
- d. <m_sof>
“i_sof(bit4)”を INT_SOF 端子に出力するかどうかを設定します。
0y0: イネーブル(出力する)
0y1: ディセーブル(出力しない)
- e. <m_rx_data0>
“i_rx_data0(bit3)”を INT_RX_ZERO 端子に出力するかどうかを設定します。
0y0: イネーブル(出力する)
0y1: ディセーブル(出力しない)
- f. <m_status>
“i_status(bit2)”を INT_STATUS 端子に出力するかどうかを設定します。
0y0: イネーブル(出力する)
0y1: ディセーブル(出力しない)
- g. <m_status_nak>
“i_status_nak(bit1)”を INT_STATUS_NAK 端子に出力するかどうかを設定します。
0y0: イネーブル(出力する)
0y1: ディセーブル(出力しない)
- h. <m_setup>
“i_setup(bit0)”を INT_SETUP 端子に出力するかどうかを設定します。
0y0: イネーブル(出力する)
0y1: ディセーブル(出力しない)
- i. <i_nak>
EP0 以外の各 EP が NAK を送信すると、1 にセットされます。(INT_NAK フラグ出力をしたい EP につきましては、INT_NAK_MASK レジスタにて選択可能です)。この bit に 1 をライトすることで INT_NAK レジスタの各 bit も 0 にクリアされます。
- j. <i_ep>
EP0 を除く各 EP に対する転送が正常に終了時、1 にセットされます(フラグ出力をしたい EP につきましては、INT_EP_MASK レジスタにて選択可能です)。この bit に 1 をライトすることで INT_EP レジスタの各 bit も 0 にクリアされます。

k. <i_ep0>

EP0 に対する転送が正常に終了時、1 にセットされます。

l. <i_sof>

SOF-token 受信時、あるいは create_sof モードで 1frame-time カウント時に 1 にセットされます。

m. <i_rx_data0>

Zero-Length データを受信時に 1 にセットされます。(フラグ出力をしたい EP につきましては、INT_EP_MASK レジスタにて選択可能です)。この bit に 1 をライトすることで INT_RX_DATA0 レジスタの各 bit も 0 にクリアされます。また、Control-RD 転送の STATUS-Stage で Zero-Length データを受信した時には、1 にセットされません。

n. <i_status>

EP0 における Control 転送において STATUS-Stage 正常終了時、1 にセットされます。(Control-RD 転送において STATUS-Stage で Zero-Length データを受信し正常終了、Control-WR 転送において STATUS-Stage で Zero-Length データを送信した後に正常終了すると 1 にセットされます。)

o. <i_status_nak>

EP0 における Control-RD 転送において、STATUS-Stage のパケット受信時に“NAK”を返信した場合、1 にセットされます。この bit がセットされた場合、DATA-Stage は終了していますので、Command レジスタにより“Setup-Fin”コマンドをセットし、UDC2 のステージを STATUS-Stage へと移行させて下さい。また、Control-WR 転送の DATA-Stage において MaxPacketSize(64 バイト:High-Speed)の整数倍を受信する際に、DATA-Stage の最後を示すために Zero-Length データを受信することがあります。その後、STATUS-Stage での In-token 受信時に、この i_status_nak により DATA-Stage が終わったことを認識することができますので、UDC2 を STATUS-Stage へと移行させて下さい。

p. <i_setup>

EP0 における Control 転送において、Setup-token 受信時に 1 にセットされます。

2. UD2INTEP (INT_EP register)

Address =(0xF440_0000)+ (0x0224)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined. Write as zero.
[15]	Reserved	R/W	0y0	EP(EP0 を除く)送受信状態フラグ 0y0: 送受信なし 0y1: 送受信有り
[14]	Reserved	R/W	0y0	
[13]	Reserved	R/W	0y0	
[12]	Reserved	R/W	0y0	
[11]	Reserved	R/W	0y0	
[10]	Reserved	R/W	0y0	
[9]	Reserved	R/W	0y0	
[8]	Reserved	R/W	0y0	
[7]	Reserved	R/W	0y0	
[6]	Reserved	R/W	0y0	
[5]	Reserved	R/W	0y0	
[4]	Reserved	R/W	0y0	
[3]	i_ep3	R/W	0y0	
[2]	i_ep2	R/W	0y0	
[1]	i_ep1	R/W	0y0	
[0]	–	–	Undefined	Read as undefined. Write as zero.

注) 該当 bit に 1 をライトすることによりクリアされます。

(個別説明)

a. <i_ep[3:1]>

EP(EP0 を除く)送受信状態フラグ

EP0 を除く各 EP に対する転送が正常に終了時、該当 bit が 1 にセットされます。(int_ep フラグを外部に出力したい EP につきましては、INT_EP_MASK レジスタにて選択可能です。)

0y0: 送受信なし

0y1: 送受信有り

3. UD2INTEPMSK (INT_EP_MASK register)

Address =(0xF440_0000)+ (0x0228)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined. Write as zero.
[15]	Reserved	R/W	0y0	フラグ出力マスク制御 0y0: イネーブル(出力する) 0y1: ディセーブル(出力しない)
[14]	Reserved	R/W	0y0	
[13]	Reserved	R/W	0y0	
[12]	Reserved	R/W	0y0	
[11]	Reserved	R/W	0y0	
[10]	Reserved	R/W	0y0	
[9]	Reserved	R/W	0y0	
[8]	Reserved	R/W	0y0	
[7]	Reserved	R/W	0y0	
[6]	Reserved	R/W	0y0	
[5]	Reserved	R/W	0y0	
[4]	Reserved	R/W	0y0	
[3]	m_ep3	R/W	0y0	
[2]	m_ep2	R/W	0y0	
[1]	m_ep1	R/W	0y0	
[0]	m_ep0	R/W	0y0	

注) 該当 bit に 1 をライトすることによりクリアされます。

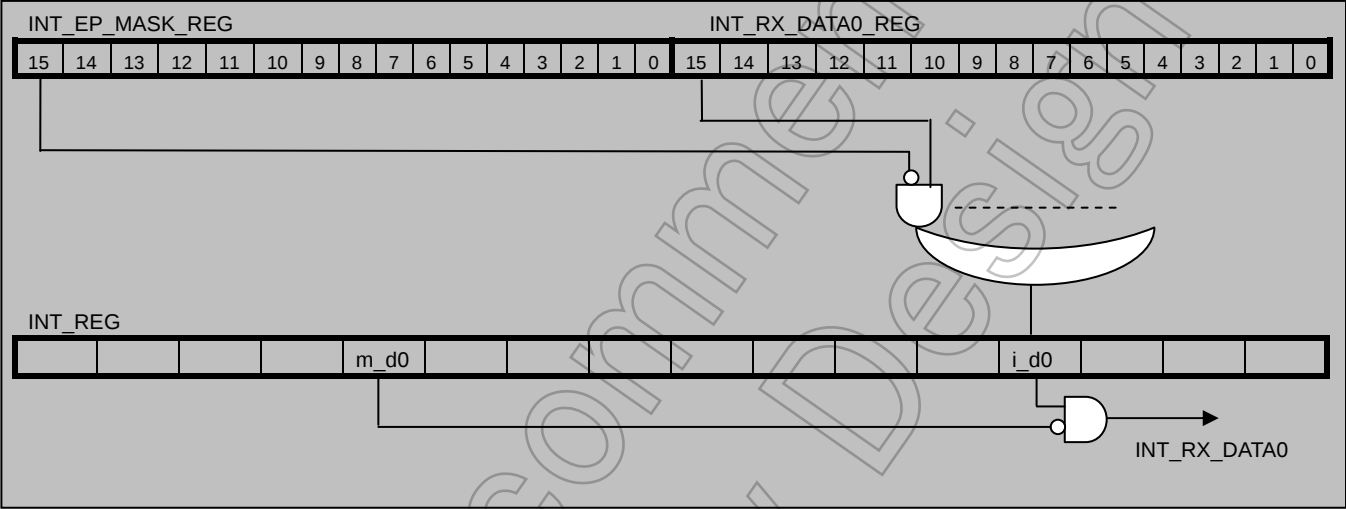
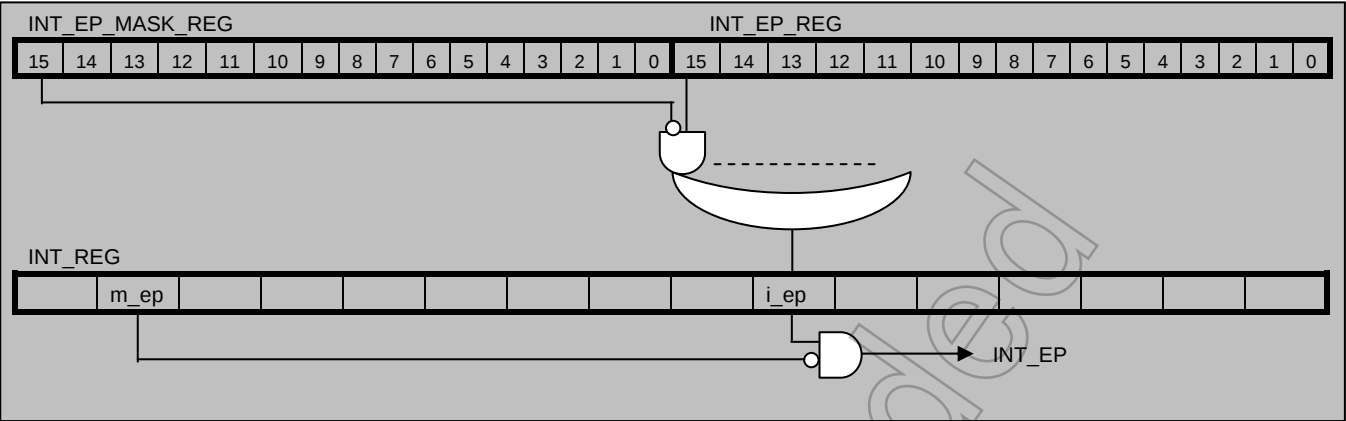
(個別説明)

a. <m_ep[3:0]>

INT_EP レジスタ及び INT_RX_DATA0 レジスタの各フラグを、それぞれ int_ep 端子、int_rx_zero 端子へ出力するかどうかを設定します。各 EP がマスクされた場合、該当 EP の転送正常終了時に INT_EP レジスタの各 bit はセットされますが、int_ep 端子はアサートされません。同様に、各 EP がマスクされた場合、該当 EP の Zero-Length データ受信時に、INT_RX_DATA0 レジスタの各 bit はセットされますが、int_rx_zero 端子はアサートされません。ただし、bit0 は INT_RX_DATA0 レジスタのみに有効です。

0y0: イネーブル(出力する)

0y1: ディセーブル(出力しない)



Endpoint1~3 のケースで i_ep/INT_EP/INT_EP_MASK の使い方の例を示します。

1. Endpoint1/Endpoint2 を DMA(Endpoint-I/F)で使用し、Endpoint3 のみ PSCI-I/F 経由で使用する場合

初期化後、INT_EP_MASK レジスタの bit1,2 は 1 をセットしてマスクしてください。EP3 に対する割り込み応答は、INT_EP レジスタの bit3 を使用しても、INT レジスタの bit6 を使用しても同様の動作となります。INT レジスタのみで使用した方が確認するレジスタが 1 箇所なので効率がよいと思います。割り込み応答は INT レジスタを使用してください。

INT	bit6:	EP3 の割り込み要因として使用します。クリア時もこの bit を使用します。
	bit14:	EP3 の割り込み要因のマスクとして使用します。
INT_EP	bit1:	無視してください。
	bit2:	無視してください。
	bit3:	無視してください。
INT_EP_MASK	bit1:	1 をセットしてマスクしてください。
	bit2:	1 をセットしてマスクしてください。
	bit3:	0 のまま変更しないでください。

2. EP0 以外に PSCI-I/F で制御する EPx が複数ある場合

以下説明のため EP2/EP3 が PSCI-I/F ,EP1 が DMA として説明します。

初期化後、DMA で使用する EP の INT_EP_MASK レジスタに 1 をセットしてマスクしてください。複数の EP に対して割り込み応答を行う場合は、必ず INT_EP レジスタを使用します。INT レジスタの i_ep は無視して、m_ep については必ず 0 のイネーブルとしてください。

INT レジスタの i_ep を使用した要因のクリアは行わないでください。割り込み発生後は、INT レジスタと INT_EP レジスタの 2 つのレジスタを確認して要因を決定する必要があります。割り込み要因のクリアは INT_EP レジスタの各要因 bit を使用してクリアしてください。

INT	bit6:	無視してください。 この bit でクリアしないでください。
	bit14:	0 のまま変更しないでください。
INT_EP	bit1:	無視してください。
	bit2:	EP2 の割り込み要因として使用します。クリア時もこの bit を使用します。
	bit3:	EP3 の割り込み要因として使用します。クリア時もこの bit を使用します。
INT_EP_MASK	bit1:	1 をセットしてマスクしてください。
	bit2:	EP2 の割り込み要因のマスクとして使用します。
	bit3:	EP3 の割り込み要因のマスクとして使用します。

4. UD2INTRX0 (INT_RX_DATA0 register)

Address =(0xF440_0000)+ (0x022C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined. Write as zero.
[15]	Reserved	R/W	0y0	EP Zero-Length データ受信フラグ 0y0: Zero-Length データ受信なし 0y1: Zero-Length データ受信有り
[14]	Reserved	R/W	0y0	
[13]	Reserved	R/W	0y0	
[12]	Reserved	R/W	0y0	
[11]	Reserved	R/W	0y0	
[10]	Reserved	R/W	0y0	
[9]	Reserved	R/W	0y0	
[8]	Reserved	R/W	0y0	
[7]	Reserved	R/W	0y0	
[6]	Reserved	R/W	0y0	
[5]	Reserved	R/W	0y0	
[4]	Reserved	R/W	0y0	
[3]	rx_d0_ep3	R/W	0y0	
[2]	rx_d0_ep2	R/W	0y0	
[1]	rx_d0_ep1	R/W	0y0	
[0]	rx_d0_ep0	R/W	0y0	

注) 該当 bit に 1 をライトすることによりクリアされます。

(個別説明)

a. <rx_d0_ep[3:0]>

EP Zero-Length データ受信フラグ

各 EP が Zero-Length データを受信した時、該当 bit が 1 にセットされます。(int_rx_zero フラグを外部に出力したい EP につきましては、INT_EP_MASK レジスタにて選択可能です。)

なお、bit0(Endpoint0) に関しましては、リクエスト処理中、DATA-Stage での Zero-Length データ受信時にのみ 1 にセットされます。STATUS-Stage での Zero-Length データ受信時にはセットされませんので、int_status フラグを使用して下さい。

0y0: Zero-Length データ受信なし

0y1: Zero-Length データ受信有り

5. UD2INTNAK (INT_NAK register)

Address =(0xF440_0000)+ (0x0330)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined. Write as zero.
[15]	Reserved	R/W	0y0	EP(EP0 を除く)NAK 送信状態フラグ 0y0: NAK 送信なし 0y1: NAK 送信有り
[14]	Reserved	R/W	0y0	
[13]	Reserved	R/W	0y0	
[12]	Reserved	R/W	0y0	
[11]	Reserved	R/W	0y0	
[10]	Reserved	R/W	0y0	
[9]	Reserved	R/W	0y0	
[8]	Reserved	R/W	0y0	
[7]	Reserved	R/W	0y0	
[6]	Reserved	R/W	0y0	
[5]	Reserved	R/W	0y0	
[4]	Reserved	R/W	0y0	
[3]	i_ep3	R/W	0y0	
[2]	i_ep2	R/W	0y0	
[1]	i_ep1	R/W	0y0	
[0]	–	–	Undefined	Read as undefined. Write as zero.

注) 該当 bit に 1 をライトすることによりクリアされます。

(個別説明)

a. <i_ep[3:1]>

EP(EP0 を除く)NAK 送信状態フラグ

EP0 を除く各 EP が NAK を送信すると該当 bit が 1 にセットされます。(該当 EP について INT_NAK フラグを出したいときは INT_EP_MASK レジスタにて選択可能です。)

0y0: NAK 送信なし

0y1: NAK 送信有り

6. UD2INTNAKMSK (INT_NAK_MASK register)

Address =(0xF440_0000)+ (0x0334)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined. Write as zero.
[15]	Reserved	R/W	0y0	フラグ出力マスク制御 0y0: イネーブル(出力する) 0y1: ディセーブル(出力しない)
[14]	Reserved	R/W	0y0	
[13]	Reserved	R/W	0y0	
[12]	Reserved	R/W	0y0	
[11]	Reserved	R/W	0y0	
[10]	Reserved	R/W	0y0	
[9]	Reserved	R/W	0y0	
[8]	Reserved	R/W	0y0	
[7]	Reserved	R/W	0y0	
[6]	Reserved	R/W	0y0	
[5]	Reserved	R/W	0y0	
[4]	Reserved	R/W	0y0	
[3]	m_ep3	R/W	0y0	
[2]	m_ep2	R/W	0y0	
[1]	m_ep1	R/W	0y0	
[0]	–	–	Undefined	Read as undefined. Write as zero.

注) 該当 bit に 1 をライトすることによりクリアされます。

(個別説明)

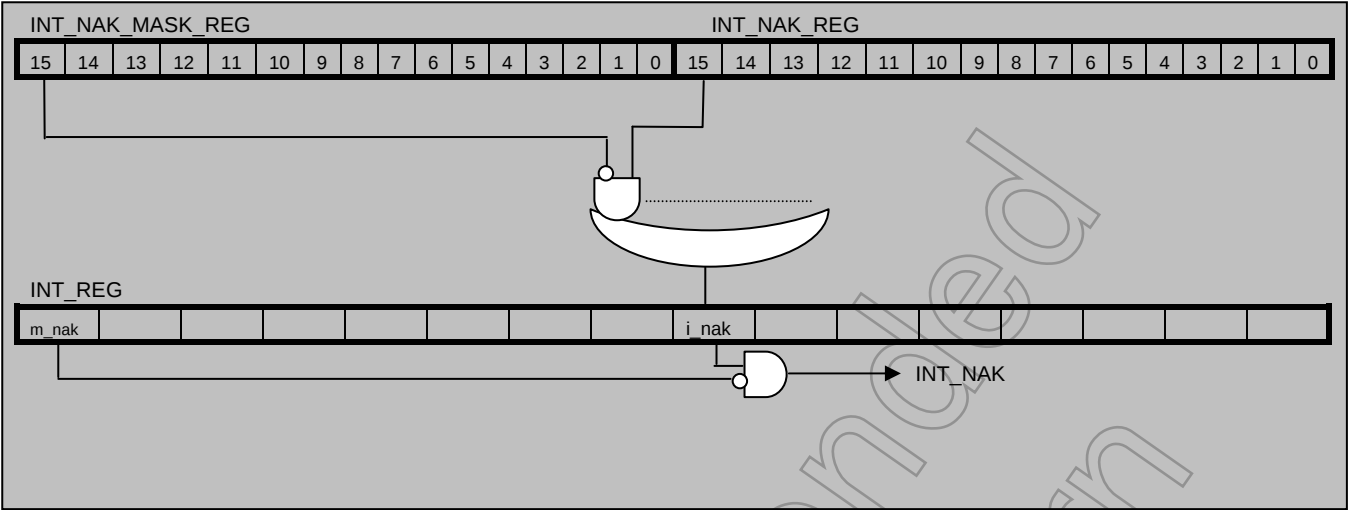
a. <m_ep[3:1]>

フラグ出力マスク制御

INT_NAK レジスタの各フラグを、それぞれ int_nak 端子へ出力するかどうかを設定します。各 EP がマスクされた場合、該当 EP の転送で NAK 送信時に INT_NAK レジスタの各 bit はセットされますが、int_nak 端子はアサートされません。

0y0: イネーブル(出力する)

0y1: ディセーブル(出力しない)



Not Recommended for New Design

(4) EP0 制御ステータスレジスタ

1. UD2EP0MSZ (EP0_MaxPacketSize register)

Address =(0xF440_0000)+ (0x0230)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	—	—	Undefined	Read as undefined. Write as zero.
[15]	tx_0data	RO	0y0	Command レジスタにより EP0 に対して“EP_TX_0DATA”コマンドを発行すると、この bit が 1 にセットされ、Zero-Length データを送信終了後 0 にクリアされます。
[14:13]	—	—	Undefined	Read as undefined. Write as zero.
[12]	dset	R/W	0y0,(-)	EP0_FIFO の状態を示します。Setup-Token 受信時に 0 にクリアされます。 0y0: 有効データ無し 0y1: 有効データ有り
[11:7]	—	—	Undefined	Read as undefined. Write as zero.
[6:0]	max_pkt[6:0]	R/W	0y1000000, (-)(-)(-)(-)(-)(-)	EP0 の MaxPacketSize を設定します。

(個別説明)

a. <tx_0data>

Command レジスタにより EP0 に対して“EP_TX_0DATA”コマンドを発行すると、この bit が 1 にセットされ、Zero-Length データを送信終了後 0 にクリアされます。

b. <dset>

EP0_FIFO の状態を示します。Setup-Token 受信時に 0 にクリアされます。

0y0: 有効データ無し

0y1: 有効データ有り

c. <max_pkt[6:0]>

EP0 の MaxPacketSize を設定します。

2. UD2EP0STS (EP0_ Status register)

Address =(0xF440_0000)+ (0x0234)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined.
[15]	ep0_mask	RO	0y0	0y0: EP0_FIFO 書込み可 0y1: EP0_FIFO への書込み不可
[14]	–	–	Undefined	Read as undefined. Write as zero.
[13:12]	toggle[1:0]	RO	0y00	現在の EP0 のトグル値を示します。 0y00: DATA0 0y01: DATA1 0y10: Reserved 0y11: Reserved
[11:9]	status[2:0]	RO	0y000	現在の EP0 の状態を示します。なお、Setup-Token を受信すると、“Ready”にクリアされます。 0y000: Ready 0y001: Busy 0y010: Error 0y011: Stall 0y100-0y111: (Reserved)
[8:0]	–	–	Undefined	Read as undefined.

(個別説明)

a. <ep0_mask>

Setup-Token 受信後、1 にセットされます。“Setup_Received”コマンドを発行することにより 0 にクリアされます。この bit が 1 の間は、EP0_FIFO への書込みが行われません。

0y0: EP0_FIFO 書込み可

0y1: EP0_FIFO への書込み不可

b. <toggle[1:0]>

現在の EP0 のトグル値を示します。

0y00: DATA0

0y01: DATA1

0y10: Reserved

0y11: Reserved

c. <status[2:0]>

現在の EP0 の状態を示します。なお、Setup-Token を受信すると、“Ready”にクリアされます。

0y000: Ready(通常の状態を示します)

0y001: Busy(STATUS-Stage で“NAK”を返信した際にセットされます)

0y010: Error(受信データが CRC エラーの場合、およびデータ送信後タイムアウトした際にセットされます)

0y011: Stall(Control-RD 転送において Length 以上のデータを要求された場合に“STALL”を返信し、status がセットされます。また、Command レジスタにより“EP0-STALL”を発行した場合もセットされます)

0y100-0y111: Reserved

3. UD2EP0DSZ (EP0_Datasize register)

Address =(0xF440_0000)+ (0x0238)

Bit	Bit Symbol	Type	Reset Value	Description
[31:7]	–	–	Undefined	Read as undefined.
[6:0]	size[6:0]	RO	0y0000000, (-)(-)(-)(-)(-)(-)	EP0_FIFO に格納されている有効データバイト数を示します。 Setup-Token 受信時にクリアされます。

(個別説明)

a. <size[6:0]>

EP0_FIFO に格納されている有効データバイト数を示します。

Setup-Token 受信時にクリアされます。

4. UD2EP0FIFO (EP0_FIFO register)

Address =(0xF440_0000)+ (0x023C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined, Write as zero.
[15:0]	data[15:0]	R/W	Undefined	EP0 への PPCI-I/F からのデータアクセスに使用します。

(個別説明)

a. <data[15:0]>

EP0 への PPCI-I/F からのデータアクセスに使用します。

このレジスタへのアクセス方法については「3.16.3.5(1)Control-RD 転送」、
「3.16.3.5(3)Control-WR転送(DATA-Stage有り)」を参照ください。

このレジスタに格納されているデータは、リクエスト受信時(INT_SETUP 割り込みアサート時)にクリアされます

(5) EP1 制御・ステータスレジスタ

1. UD2EP1MSZ (EP1_MaxPacketSize register)

Address =(0xF440_0000)+ (0x0240)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined. Write as zero.
[15]	tx_0data	RO	0y0	Command レジスタにより EP1 に対して“EP1_TX_0DATA”コマンドを発行するか、Endpoint-I/F で Zero-Length データをセットすると、この bit が 1 にセットされます。Zero-Length データを送信終了後 0 にクリアされます。
[14:13]	–	–	Undefined	Read as undefined. Write as zero.
[12]	dset	RO	(注 1), (注 2)	EP1_FIFO の状態を示します。 0y0: 有効データ無し 0y1: 有効データ有り
[11]	–	–	Undefined	Read as undefined. Write as zero.
[10:0]	max_pkt[10:0]	R/W	0y0000000000 (-)(-)(-)(-)(-)(-) (-)(-)(-)(-)(-)(-)	EP1 の MaxPacketSize を設定します。 注) 詳細は 3.16.4.2 Appendix B MaxPacketSize 奇数バイト設定関連を参照して下さい。

注 1) dset(bit12)の reset_x 後の初期値は、EPx が Tx 用 Endpoint の場合は 1 に、Rx 用 Endpoint の場合は 0 となります。

注 2) dset(bit12)の USB_RESET 後の初期値は、EPx が Tx 用 Endpoint の場合は 1 に、Rx 用 Endpoint の場合は“保持”となります。

注 3) レジスタの構造は EP1~EP3 は全て同じです。そのため、EP1 のみ表現しています。

EP2, EP3 のレジスタのアドレスは、レジスタマップでご確認願います。

注 4) EP3 を Dual で使用する場合、1 パケットの Max サイズが 32 バイトになります。

(個別説明)

a. <tx_0data>

Command レジスタにより EP1 に対して“EP1_TX_0DATA”コマンドを発行するか、Endpoint-I/F で Zero-Length データをセットすると、この bit が 1 にセットされます。Zero-Length データを送信終了後 0 にクリアされます。

b. <dset>

EP1_FIFO の状態を示します。

0y0: 有効データ無し

0y1: 有効データ有り

c. <max_pkt[10:0]>

EP1 の MaxPacketSize を設定します。

Set_Configuration、Set_Interface 受信時に Endpoint の構成を行う時にセットして下さい。

送信用 Endpoint 使用時には偶数をセットして下さい。USB 上、送信用 Endpoint の MaxPacketSize を奇数として動作させる場合には、max_pkt には偶数をセットし、Endpoint への Write アクセスで奇数アクセスを実行して下さい。(例えば、MaxPacketSize を 1023 バイトとする場合、max_pkt には 1024 をセットして下さい。)

注) 詳細は 3.16.4.2 Appendix B MaxPacketSize 奇数バイト設定関連を参照して下さい。

2. UD2EP1STS (EP1_Status register)

Address =(0xF440_0000)+ (0x0244)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined. Write as zero.
[15]	pkt_mode	R/W	0y0	EP1 のパケットモードを選択します。 0y0: Single モード 0y1: Dual モード
[14]	bus_sel	R/W	0y0	EP1 の FIFO へのアクセスをするバスを選択します。 0y0: 共通バスアクセス 0y1: 直接アクセス
[13:12]	toggle[1:0]	RO	0y00	現在の EPx のトグル値を示します。 0y00: DATA0 0y01: DATA1 0y10: DATA2 0y11: MDATA
[11:9]	status[2:0]	RO	0y111	現在の EP1 の状態を示します。command レジスタより EP_Reset を発行することにより status は "Ready" となります。 0y000: Ready 0y001: Reserved 0y010: Error 0y011: Stall 0y100-0y110: Reserved 0y111: Invalid
[8]	disable	RO	0y0	EP1 の転送許可状態を示します。 0y0: 許可 0y1: 禁止
[7]	dir	R/W	0y0	この Endpoint に対する転送方向を設定します。 0y0: OUT(Host-to-device) 0y1: IN(Device-to-host)
[6:4]	–	–	Undefined	Read as undefined. Write as zero.
[3:2]	t_type[1:0]	R/W	0y00	この Endpoint の転送モードを設定します。 0y00: Control 0y01: Isochronous 0y10: Bulk 0y11: Interrupt
[1:0]	num_mf[1:0]	R/W	0y00	Isochronous 転送を選択した場合、μ フレーム中に何回転送をするかを設定します。 0y00: 1-transaction 0y01: 2-transaction 0y10: 3-transaction 0y11: Reserved

注 1) このレジスタへは、Set_Configuration、Set_Interface 受信時に Endpoint の構成を行う時にセットして下さい。

注 2) レジスタの構造は EP1~EP3 は全て同じです。そのため、EP1 のみ表現しています。

EP2,EP3 のレジスタのアドレスは、レジスタマップでご確認願います。

各 EP の設定は、製品仕様に依存いたします。

本製品では EP1 は IN 転送固定の仕様ですので、dir は 1 設定しか出来ません。

EP2 は OUT 転送固定の仕様ですので、0 設定しか出来ません。

EP3 は IN 転送固定の仕様ですので、1 設定しか出来ません。

(個別説明)

a. <pkt_mode>

EP1 のパケットモードを選択します。Dual モードを選択することにより、Epx に対する 2 つのパケットデータを保持することが可能となります。

0y0: Single モード

0y1: Dual モード

b. <bus_sel>

EP1 の FIFO へのアクセスをするバスを選択します。

0y0: 共通バスアクセス

0y1: 直接アクセス

c. <toggle[1:0]>

現在の EPx のトグル値を示します。

0y00: DATA0

0y01: DATA1

0y10: DATA2

0y11: MDATA

d. <status[2:0]>

現在の EP1 の状態を示します。command レジスタより EP_Reset を発行することにより status は“Ready”となります。

0y000: Ready(通常の状態を示します)

0y001: Reserved

0y010: Error(データパケットに受信エラーが発生した時、または送信後タイムアウトが発生した時にセットされます。但し、“Stall”、“Invalid”がセットされている場合にはセットされません)

0y011: Stall (Command レジスタにより“EP_Stall”を発行した場合にセットされます)

0y100-0y110: Reserved

0y111: Invalid (この Endpoint が無効の状態であることを示します)

e. <disable>

EP1 の転送許可状態を示します。“禁止”状態にある場合、この Endpoint に対する Token に対しては“NAK”を返信し続けます。

0y0: 許可

0y1: 禁止

f. <dir>

この Endpoint に対する転送方向を設定します。

0y0: OUT (Host-to-device)

0y1: IN (Device-to-host)

注 1) EP1 は IN 転送固定です。必ず 1 に設定してください。

注 2) EP2 は OUT 転送固定です。必ず 0 に設定してください。

注 3) EP3 は IN 転送固定です。必ず 1 に設定してください。

g. <t_type[1:0]>

この Endpoint の転送モードを設定します。

0y00: Control

0y01: Isochronous

0y10: Bulk

0y11: Interrupt

h. <num_mf[1:0]>

Isochronous 転送を選択した場合、μフレーム中に何回転送をするかを設定します。

0y00: 1-transaction

0y01: 2-transaction

0y10: 3-transaction

0y11: Reserved

3. UD2EP1DSZ (EP1_Datasize register)

Address =(0xF440_0000)+ (0x0248)

Bit	Bit Symbol	Type	Reset Value	Description
[31:11]	–	–	Undefined	Read as undefined.
[10:0]	size[10:0]	RO	0y000000000000, (-)(-)(-)(-)(-)(-)(-)(-)(-)(-)(-)(-)	EP1_FIFO に格納されている有効データバイト数を示します。なお、Dual パケットモード時には、最初にアクセスするパケットのデータバイト数を示します。

注) レジスタの構造は EP1~EP3 は全て同じです。そのため、EP1 のみ表現しています。

EP2,EP3 のレジスタのアドレスは、レジスタマップでご確認ねがいます。

(個別説明)

a. <size[10:0]>

EP1_FIFO に格納されている有効データバイト数を示します。なお、Dual パケットモード時には、最初にアクセスするパケットのデータバイト数を示します。

4. UD2EP1FIFO (EP1_FIFO register)

Address =(0xF440_0000)+ (0x024C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read as undefined. Write as zero.
[15:0]	data[15:0]	–	Undefined	EPx への PPCI-I/F からのデータアクセスに使用します。

注) レジスタの構造は EP1~EP3 は全て同じです。そのため、EP1 のみ表現しています。

EP2,EP3 のレジスタのアドレスは、レジスタマップでご確認ねがいます。

(個別説明)

a. <data[15:0]>

EPx への PPCI-I/F からのデータアクセスに使用します。

3.16.3.4 USBデバイス応答

UDC2はハードウェアリセット検出時、USB_RESET検出時、エナミュレーション応答時にUDC2内部の初期化や各種レジスタの設定を行います。この章では、それぞれの状態でのUDC2の動作および外部からの制御方法を説明します。

(1) ハードウェアリセット検出時

UDC2は電源投入後必ずハードウェアリセットをかけて下さい。ハードウェアリセット後は、UDC2は内部レジスタを初期化し、全てのEndpointは禁止状態(Invalid)になっており、デバイスとしては“Disconnect”状態になっています。

UDC2を“Default”状態にするには、“USB_Ready”コマンドを発行して下さい。このコマンドを発行することにより、UDC2は“Full-Speed”モードとなり、DDPのPull-Up抵抗をイネーブルにし、ホストへ“Connect”を通知します。

この状態では、ホストからの信号はUSB_RESET信号のみ受け付けます。

(2) USB_RESET検出時

UDC2はUSB信号上にバスリセット(USB_RESET)を検出すると、内部レジスタの初期化を行い、デバイスとしては“Default”状態となります。この状態ではEndpoint0のみ“Ready”状態となり、ホストとのエナミュレーションが可能な状態となります。

また、この状態においてUDC2は“HS-Chirp”モードとなり、ホストとのChirp動作に入ります。ホストからのChirpが正常に受信できるとUDC2はHigh-Speed(HS)モードとなり、これ以降のホストとの転送はHSにて行われます。ホストからのChirpが受信出来なかった場合、これ以降のホストとの転送はFull-Speed(FS)にて行われます。

現在の転送スピードは、Address-stateレジスタのbit[13:12]をリードすることにより判断することが出来ます。

(3) “Set_address”リクエスト受信時

“Set_address”リクエスト受信後にAddress-stateレジスタのbit[10:8]に0y010を、bit[6:0]に受信したアドレス値をセットすることにより、UDC2は“Addressed”状態となります。なお、このレジスタへのセットは、Control転送の正常終了後(STATUS-Stage終了後)にセットして下さい。

なお、この状態では未だEndpoint0以外のEndpointへの転送は出来ません。

(4) “Set_configuration”、“Set_interface”リクエスト受信時

“Set_configuration”、“Set_interface”リクエスト受信後に Address-state レジスタの bit[10:8]に 0y100 をセットすることにより、UDC2 は“Configured”状態となります。

“Configured”状態においては、ステータス設定をした Endpoint について転送を行うことが可能となります。

なお、Endpoint を“Ready”とするには、下記設定を行う必要があります。

- EPx_MaxPacketSize レジスタへのマックスパケットサイズのセット
- EPx_Status レジスタへの転送モードのセット
- Command レジスタへの EP_Reset コマンドの発行

この設定を行うことにより、Endpoint は送受信可能となります。

図 3.16.18に、“Device State Diagram”を示します。

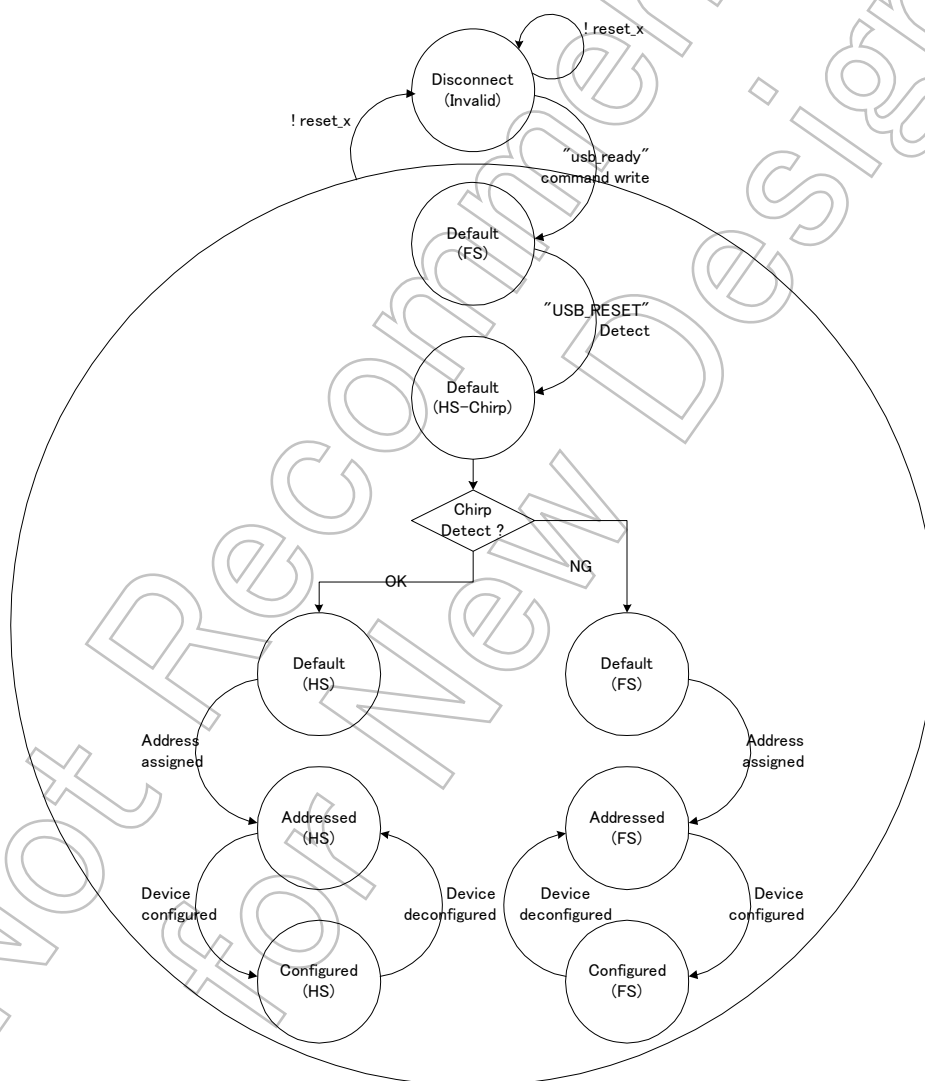


図 3.16.18 Device State Diagram

3.16.3.5 各Endpointの転送における制御フロー

- Endpoint0

Endpoint0 は Control 転送をサポートしており、エミュレーション等デバイスの制御を行うのに使用します。なお、Endpoint0 はシングルパケットモードのみとなります。

Control 転送は次の 3 つのステージで構成されます。

SETUP-Stage

DATA-Stage

STATUS-Stage

また、転送の種類として、大きく以下 3 種類に分類されます。

- Control-RD 転送
- Control-WR 転送(DATA-Stage 無し)
- Control-WR 転送(DATA-Stage 有り)

UDC2 は、ハードウェアで 3 つのステージ管理を行います。以下に各転送における流れを示します。

(1) Control-RD 転送

以下に Control-RD 転送時の流れを示します。

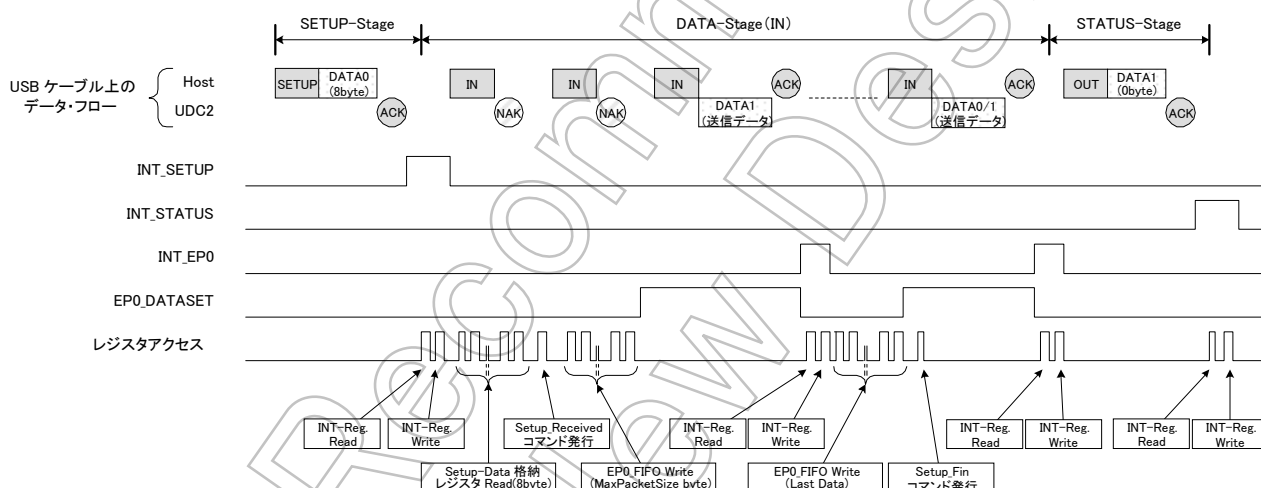


図 3.16.19 Control-RD 転送における制御フロー

以下の説明では EP0_MaxPacketSize レジスタの bit12(dset)を“EP0_DATASET フラグ”としています。

1. SETUP-Stage

UDC2 は、Setup-Token を受信すると、INT_SETUP フラグをアサートします。このフラグをクリアするには、INT レジスタの bit0(i_setup)に 1 をライトして下さい。外部でフラグを束ねている場合は INT レジスタをリードし、どのフラグがアサートされているかを確認し、該当 bit に 1 をライトして下さい。

次に、Setup-Data 格納レジスタ(bRequest-bmRequestType、wValue、wIndex、wLength レジスタ)をリードし、リクエストを判断して下さい。

最後に、“Setup_Received”コマンドを発行し、SETUP-Stage の終了を UDC2 に知らせます。UDC2 はこのコマンドが発行されるまでは、Endpoint0-FIFO へのデータライトを禁止していますので、このコマンドが発行されるまではホストからの IN-Token に対して“NAK”を返信します。

2. DATA-Stage

IN-Token に対して送信すべきデータを Endpoint0-FIFO へライトします。なお、送信したいデータバイト数が MaxPacketSize より大きい場合は、MaxPacketSize 毎に分割してライトして下さい。ライトデータ数が MaxPacketSize になると、EP0_DATASET フラグがアサートされます。

ホストからの IN-Token に対し正常にデータ送信が終了すると、UDC2 は EP0_DATASET フラグをデアサートし、INT_EP0 をアサートします。送信したいデータが残っている場合、Endpoint0-FIFO へのデータライトを行って下さい。

ライトすべきデータが MaxPacketSize より少ない場合は、EP0 へ“EP_EOP”コマンドを発行することにより、UDC2 にショートパケットであることを通知して下さい。これにより、UDC2 はパケットの終了を認識し、ショートパケットデータを送信します。

最後に、“Setup_Fin”コマンドを発行することにより、UDC2 へ DATA-Stage の終了を知らせます。

3. STATUS-Stage

UDC2 は、“Setup_Fin”コマンドが発行されると、STATUS-Stage に対する Handshake を自動で行います。正常に STATUS-Stage が終了すると、INT_STATUS フラグをアサートします。“Setup_Fin”コマンドが発行される前にホストから STATUS-Stage のパケットを受信すると、UDC2 は“NAK”を返信し、INT_STATUS_NAK フラグをアサートしますので、このフラグがアサートされましたら、“Setup_Fin”コマンドを発行するようにして下さい。

(2) Control-WR 転送(DATA-Stage 無し)

以下に Control-WR 転送(DATA-Stage 無し)時の流れを示します。

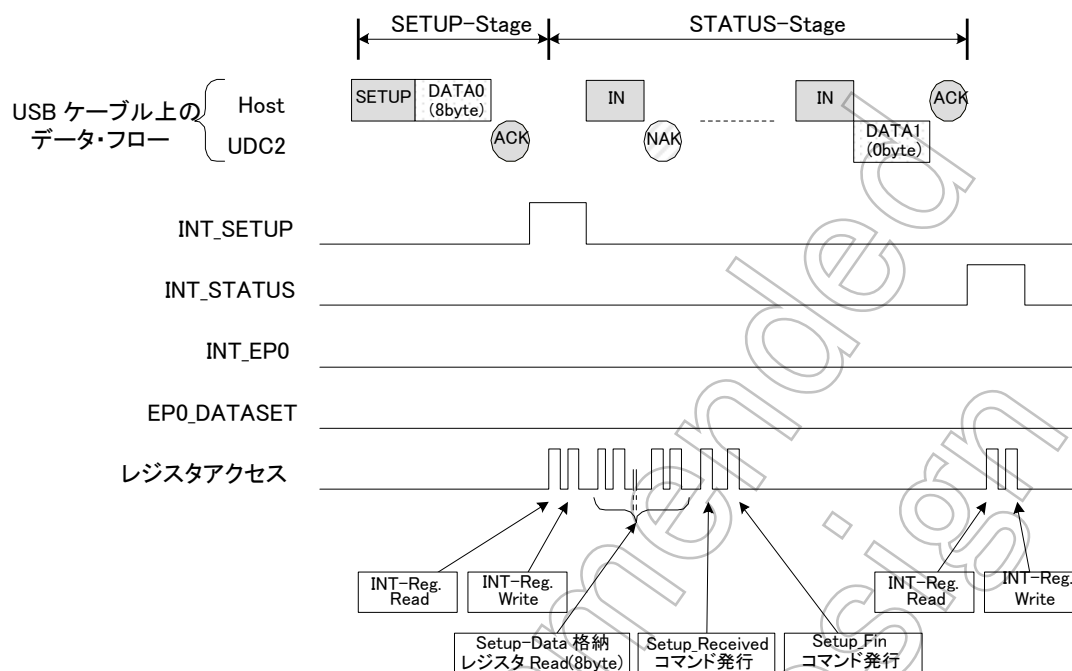


図 3.16.20 Control-WR 転送(DATA-Stage 無し)における制御フロー

1. SETUP-Stage

(1)のSETUP-Stageと同様の処理を行って下さい。

2. STATUS-Stage

“Setup_Received”コマンドを発行した後に、各リクエストに応じてUDC2 へのレジスタアクセスを行って下さい。UDC2 へのレジスタアクセスが全て終了した時に“Setup_Fin”コマンドを発行して下さい。その後の処理は、基本的には (1)のSTATUS-Stageと同様です。UDC2 は“Setup_Fin”コマンドが発行されるまで“NAK”を返信しつづけます。

注) 基本的には、「“Setup_Received”コマンドの発行」 → 「“Setup_Fin”コマンドの発行」の間で各リクエストに必要なレジスタアクセスをUDC2 へ行うのですが、Set AddressリクエストやSet Feature(TEST_MODE)のように、STATUS-Stage終了後にレジスタアクセスが必要な場合があります。各スタンダードリクエストでの処理については (5)にて説明しています。

(3) Control-WR 転送(DATA-Stage 有り)

以下に Control-WR 転送(DATA-Stage 有り)時の流れを示します。

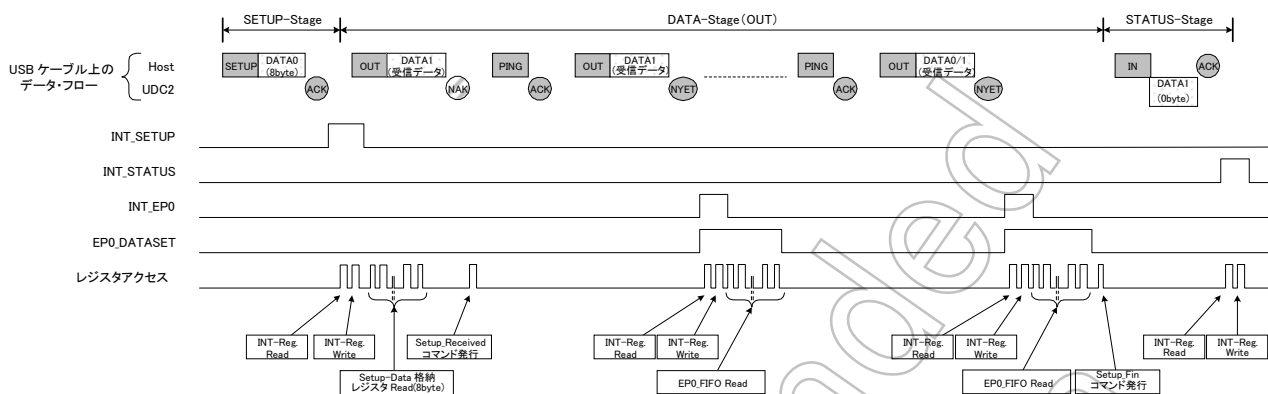


図 3.16.21 Control-WR 転送(DATA-Stage 有り)における制御フロー

1. SETUP-Stage

(1)のSETUP-Stageと同様の処理を行って下さい。

2. DATA-Stage

UDC2 は、ホストからデータを正常に受信すると、EP0_DATASET フラグをアサートし、INT_EP0 フラグをアサートします。このフラグがアサートされたら、受信したデータサイズを EP0_Datasize レジスタで確認した後に EP0_FIFO からデータをリードするか、EP0_DATASET フラグをポーリングしながら EP0_FIFO からデータをリードして下さい。

受信データバイト数をリードすると、UDC2 は EP0_DATASET フラグをデアサートします。

3. STATUS-Stage

(1)のSTATUS-Stageと同様の処理を行って下さい。

注) 図 3.16.21はHigh-Speed転送のフローを表しています。Full-Speed転送の場合には、図中の“PING”パケットは発行されません。また、“NYET”パケットは“ACK”パケットとなります。

(4) INT_STATUS_NAK フラグの使用例

DATA-Stage の無いリクエストの処理では、(特に High-Speed 転送では)INT_SETUP フラグのアサートを受けてからクリアをする前に、ホストからの STATUS-Stage を受信して INT_STATUS_NAK フラグがアサートされる可能性があります。このような多重割り込みの状態をなるべく回避したい場合に、データステージの無いリクエストでは INT_STATUS_NAK フラグをマスクする方法があります。その際は、基本的に INT レジスタの「m_status_nak」に 1 をセットし、DATA-Stage の有るリクエスト受信時にだけ 0 をセットする処理を行って下さい。(以下は Control-RD 時の例です。)

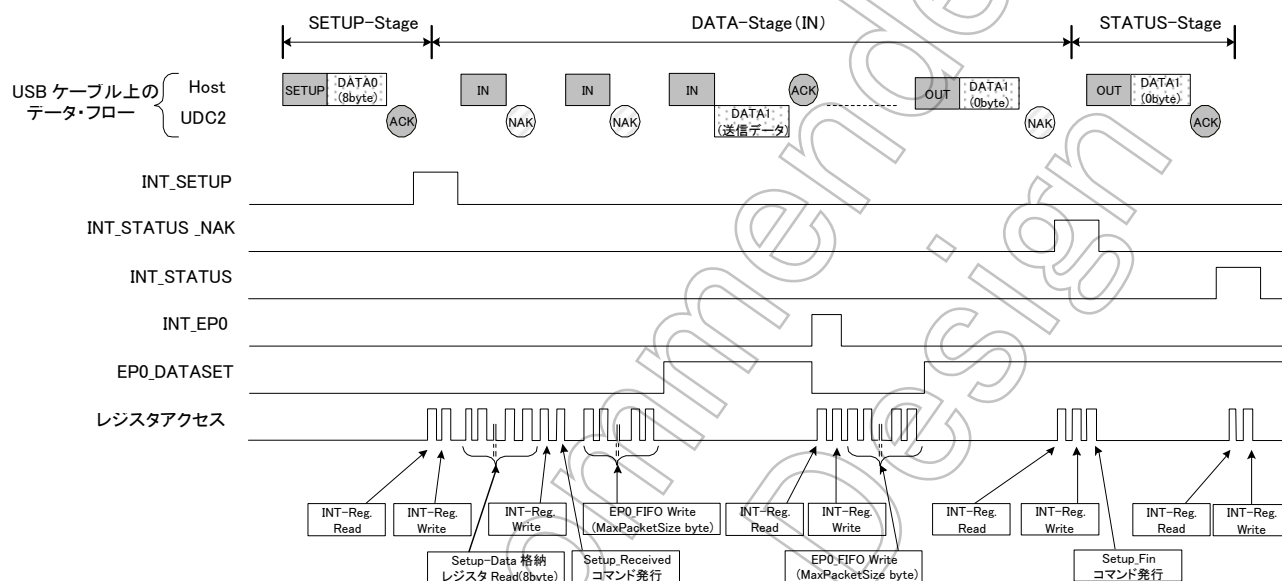


図 3.16.22 Control-RD 転送における INT_STATUS_NAK フラグの使用例

1. SETUP-Stage

INT_SETUP フラグのアサートを受けて、INT レジスタの bit0(i_setup)をクリアして下さい。この時、bit1(i_status_nak)が 1 にセットされている場合にはこの bit もクリアして下さい。

次に、Setup-Data 格納レジスタをリードして DATA-Stage のあるリクエストと判断した場合には、INT レジスタの bit9(m_status_nak)を 0 として下さい。その後、“Setup_Received”コマンドを発行して下さい。

2. DATA-Stage → STATUS-Stage

INT_STATUS_NAK フラグがアサートされると、デバイスも STATUS-Stage に移行する必要があります。INT レジスタの bit1(i_status_nak)をクリアし、その後、“Setup_Fin”コマンドを発行して下さい。また、次からの転送に備えるため再び INT レジスタの bit9(m_status_nak)に 1 をセットして下さい。

(5) スタンダードリクエスト受信時の処理

スタンダードリクエスト受信時の、UDC へのレジスタアクセス例を次項から説明します。各リクエストの説明は、基本的にデバイスのステート(Default,Address,Configured)に分けています。

各リクエスト共通のレジスタアクセスに関しましては、(1)、(2)、(3)に記載してありますのでそちらを参照して下さい。

ただし、次項からの説明では USB2.0 のスタンダードリクエスト仕様の全内容を記載しているわけではありません。ユーザーの使用方法によってレジスタアクセス方法は異なると思いますので、必ず USB2.0 仕様をご覧ください。また、説明中の「Recipient」、「Descriptor Types」、「Standard Feature Selectors」、「Test Mode Selectors」等は USB2.0 仕様に記載されていますので、そちらをご覧ください。

- ・「(1)Control-RD転送」に対応するスタンダードリクエスト
 - ・ Get Status
 - ・ Get Descriptor
 - ・ Get Configuration
 - ・ Get Interface
 - ・ Synch Frame
- ・「(2).Control-WR転送(DATA-Stage無し)」に対応するスタンダードリクエスト
 - ・ Clear Feature
 - ・ Set Feature
 - ・ Set Address
 - ・ Set Configuration
 - ・ Set Interface
- ・「(3).Control-WR転送(DATA-Stage有り)」に対応するスタンダードリクエスト
 - ・ Set Descriptor

注 1) 二重線が引かれている箇所は、UDC2 へのレジスタアクセスを説明しています。

注 2) Command レジスタへのライトアクセスは、文章の簡略化のために以下のように記述しています。

(例 1) Command レジスタの bit7-4(ep)へ 0x0、bit3-0(com)へ 0x4 をライトする場合

→ EP0 へ EP-Stall コマンドを発行して下さい

(例 2) Command レジスタの bit7-4(ep)へ該当 Endpoint、bit3-0(com)へ 0x5 をライトする場合

→ 該当 Endpoint へ EP-Invalid コマンドを発行して下さい

(a) Get Status リクエスト

このリクエストにより、指定された受信側(recipient)のステータスを返信します。

BmRequestType	BRequest	wValue	wIndex	wLength	Data
0y10000000 0y10000001 0y10000010	GET_STATUS	Zero	Zero Interface Endpoint	Two	Device, Interface, or Endpoint Status

- 全ステート共通:

wIndex で指定された Endpoint/Interface が存在しない場合は EP0 へ EP-Stall コマンドを発行して下さい。

- Default state:

USB2.0 仕様ではデバイスの動作に指定はありません。

- Address state:

- Recipient=Device : デバイスの情報(図 3.16.23)を、EP0 FIFOレジスタにライトして下さい。
- Recipient=Interface : EP0 へ EP-Stall コマンドを発行して下さい。
- Recipient=Endpoint : wIndex=0(EP0)の場合は、Endpoint0 の情報(図 3.16.25)をEP0 FIFOレジスタにライトして下さい。
wIndex≠0(EPx)の場合は、EP0 へ EP-Stall コマンドを発行して下さい。

- Configured state:

- Recipient=Device : デバイスの情報(図 3.16.23)をEP0 FIFOレジスタにライトして下さい。
- Recipient=Interface : wIndexで指定されたInterfaceが存在する場合は、インターフェースの情報(図 3.16.24)をEP0 FIFOレジスタにライトして下さい。
- Recipient=Endpoint : wIndexで指定されたEndpointが存在する場合は、該当エンドポイントの情報(図 3.16.25)をEP0_FIFOレジスタにライトして下さい。

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	Remote Wakeup	Self Powered
D15	D14	D13	D12	D11	D10	D9	D8
0	0	0	0	0	0	0	0

図 3.16.23 Get Status リクエストで返信されるデバイスの情報

- SelfPowered(D0) : 0 の場合はバスパワー、1 の場合はセルフパワーを表します。
- RemoteWakeup(D1) : 0 の場合はリモートウェイクアップ機能がディセーブル、1 の場合はイネーブルとなっていることを表します。

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0
D15	D14	D13	D12	D11	D10	D9	D8
0	0	0	0	0	0	0	0

図 3.16.24 Get Status リクエストで返信される Interface の情報

- 全 bit が 0 となっています。

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	Halt
D15	D14	D13	D12	D11	D10	D9	D8
0	0	0	0	0	0	0	0

図 3.16.25 Get Status リクエストで返信される Endpoint の情報

- Halt(D0): この bit が 1 の場合は該当 Endpoint が Halt 状態となっていることを表します。

(b) Clear Feature リクエスト

このリクエストにより、特定機能をクリア/ディセーブルします。

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0y00000000 0y00000001 0y00000010	CLEAR_FEATURE	Feature Selector	Zero Interface Endpoint	Zero	None

- 全ステート共通:

クリア(ディセーブル)できない、または、存在しない Feature Selector(**wValue**)を指定された場合は EP0 へ EP-Stall コマンドを発行して下さい。

wIndex で指定された Endpoint/Interface が存在しない場合は EP0 へ EP-Stall コマンドを発行して下さい。

- Default state:

USB2.0 仕様ではデバイスの動作に指定はありません。

- Address state:

- Recipient=Device : **wValue**=1 の場合はユーザー側で **DEVICE_REMOTE_WAKEUP** 機能をディセーブルして下さい。UDC2 へのレジスタアクセスは必要ありません。

- Recipient=Interface : EP0 へ EP-Stall コマンドを発行して下さい。

- Recipient=Endpoint : **wIndex**≠0(EPx)の場合は、EP0 へ EP-Stall コマンドを発行して下さい。

wValue=0,**wIndex**=0(EP0)の場合は Endpoint0 の Halt 状態をクリアするのですが、UDC2 へのレジスタアクセスは必要ありません

- Configured state:

- Recipient=Device : **wValue**=1 の時はユーザー側で **DEVICE_REMOTE_WAKEUP** 機能をディセーブルして下さい。UDC2 へのレジスタアクセスは必要ありません。

- Recipient=Interface : EP0 へ EP-Stall コマンドを発行して下さい。(注)

- Recipient=Endpoint : **wValue**=0 で,**wIndex**≠0(EPx)の場合は、該当 Endpoint へ EP-Reset コマンドを発行して下さい。
wValue=0 で,**wIndex**=0(EP0)の場合は Endpoint0 の Halt 状態をクリアするのですが、UDC2 へのレジスタアクセスは必要ありません。

注)ここでは、「Interface の Feature Selector は存在しない」という USB2.0 仕様の解釈にて、Endpoint0 を Stall 処理としています。仕様の詳細は USB 仕様をご覧ください。

(c) Set Feature リクエスト

このリクエストにより、特定機能をセット/イネーブルします。

bmRequestType	bRequest	wValue	wIndex		wLength	Data
0y00000000 0y00000001 0y00000010	SET_FEATURE	Feature Selector	Zero Interface Endpoint	Test Selector	Zero	None

- 全ステート共通:

High-Speed 対応デバイスで、Recipient=Device, **wValue=2** を指定された場合は、STATUS-Stage が終了して 3ms 以内に、Test Selector 値(wIndex 上位バイト)を、USB-testmode レジスタの bit7-0(t_sel)へライトして下さい。

ただし、Test Selector 値にて無効な値(test_j, test_k, se0_nak, test_packet 以外)を指定された場合には、EP0 へ EP-Stall コマンドを発行して下さい。

注) 標準ではないベンダー固有の Test Selector を使用する場合はそれに応じて処理して下さい。

- セット(イネーブル)できない、または、存在しない Feature Selector(**wValue**)を指定された場合は EP0 へ EP-Stall コマンドを発行して下さい。
- **wIndex** 下位バイトで指定された Endpoint/Interface が存在しない場合は EP0 へ EP-Stall コマンドを発行して下さい。

- Default state:

上記 TEST_MODE 時以外では、USB2.0 仕様ではデバイスの動作に指定はありません。

- Address state:

- Recipient=Device : wValue=1 の場合はユーザー側で DEVICE_REMOTE_WAKEUP 機能をイネーブルにして下さい。UDC2 へのレジスタアクセスは必要ありません。
- Recipient=Interface : EP0 へ EP-Stall コマンドを発行して下さい。
- Recipient=Endpoint : wIndex 下位バイト≠0(EPx)の場合は、EP0 へ EP-Stall コマンドを発行して下さい。
wValue=0 で、wIndex 下位バイト=0(EP0)の場合は、Endpoint0 を Halt 状態として下さい。(注 2)

- Configured state:

- Recipient=Device : wValue=1 の場合はユーザー側で DEVICE_REMOTE_WAKEUP 機能をイネーブルにして下さい。UDC2 へのレジスタアクセスは必要ありません。
- Recipient=Interface : EP0 へ EP-Stall コマンドを発行して下さい。(注 1)
- Recipient=Endpoint : wValue=0 で、wIndex 下位バイト≠0(EPx)の場合は、該当エンドポイントへ EP-Stall コマンドを発行して下さい。
wValue=0 で、wIndex 下位バイト=0(EP0)の場合は、Endpoint0 を Halt 状態として下さい。(注 2)

注 1)ここでは、「Interface の Feature Selector は存在しない」という USB 仕様の解釈にて、Endpoint0 を Stall 処理としています。仕様の詳細は USB 仕様をご覧ください。

注 2)USB2.0 仕様には、“Endpoint0 に対して Halt 機能が実行される必要はない/推奨しない”のような記述があります。よってこの場合に、UDC2 に対して Stall 状態を設定する必要がないというような解釈が可能です。

実際に Endpoint0 を Halt 状態にするためには、ユーザー側にて「Halt 状態」を管理する必要があります。

そして「Halt 状態」の時にリクエストを受けた場合に、DATA-Stage/STATUS-Stage で EP0 へ EP-Stall コマンド発行を行うといった処理が必要となります。(仮に Endpoint0 を Stall 状態に設定しても、Setup-Token を受信すると UDC2 は Stall 状態を解除して“ACK”を返信します。)

このように Endpoint0 に対して Set Feature/Clear Feature を受信した場合の処理は、ユーザーの使用方法により異なります。

Not Recommended
for New Design

(d) Set Address リクエスト

このリクエストにより、デバイスアドレスをセットします。

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0y00000000	SET_ADDRESS	Device Address	Zero	Zero	None

本リクエストでは、以下に示すレジスタアクセスを STATUS-Stage 終了後 2ms 以内に行ってください。

(Setup_Fin コマンド発行前に、デバイスアドレスを変更しないで下さい。)

- Default state:

- wValue=0 の場合

- デフォルトステートを維持します。UDC2 へのレジスタアクセスは必要ありません。

- wValue≠0 の場合

- Address-State レジスタの bit6-0(dev adr)に wValue、bit10-8(Device State)に 0y010 をセットして下さい。UDC2 はアドレスステートに入ります。

- Address state:

- wValue=0 の場合

- Address-State レジスタの bit6-0(dev adr)に 0x00、bit10-8(Device State)に 0x001 をセットして下さい。UDC2 はデフォルトステートに入ります。

- wValue≠0 の場合

- Address-State レジスタの bit6-0(dev adr)に wValue をセットして下さい。

- UDC2 は新しいデバイスアドレスに設定されます。

- Configured state: USB2.0 仕様ではデバイスの動作に指定はありません。

(e) Get Descriptor リクエスト

このリクエストにより、指定されたディスクリプタを返信します。

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0y10000000	GET_DESCRIPTOR	Descriptor Type and Descriptor Index	Zero or Language ID	Descriptor Length	Descriptor

- 全ステート共通:

- wValueで指定されたディスクリプタ情報を、wLengthにて指定されたバイト数分 EP0 FIFOレジスタにライトして下さい。 ライトするバイト数がEndpoint0 の MaxPacketSizeよりも大きい場合には、複数回に分けてライトする必要があります(詳細は (1)Control-RD転送を参照して下さい)。(ディスクリプタの長さが wLengthより大きい場合は、ディスクリプタの最初からwLengthバイトをライトして下さい。ディスクリプタの長さがwLengthより小さい場合は、そのディスクリプタ全てをライトして下さい。)
 - wValue で指定されたディスクリプタをユーザー側がサポートしない場合は、EP0 へ EP-Stall コマンドを発行して下さい。

(f) Set Descriptor リクエスト

このリクエストにより、ディスクリプタの更新/追加を行います。

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0y00000000	SET_DESCRIPTOR	Descriptor Type and Descriptor Index	Language ID or zero	Descriptor Length	Descriptor

- 全ステート共通:

本リクエストをサポートしない場合には EP0 へ EP-Stall コマンドを発行して下さい。

- Default state:

USB2.0 仕様ではデバイスの動作に指定はありません。

- Address state & Configured state:

UDC2 が受信したディスクリプタ情報を EP0 FIFO レジスタからリードして下さい。

(g) Get Configuration リクエスト

このリクエストにより、現在のデバイスの Configuration 値を返信します。

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0y10000000	GET_CONFIGURATION	Zero	Zero	One	Configuration Value

- Default state:

USB2.0 仕様ではデバイスの動作に指定はありません。

- Address state:

EP0_FIFO レジスタ[7:0]に 0x00 をライトして下さい。 未構成なので 0 を返信する必要があります。

- Configured state:

現在の Configuration 値を EP0_FIFO レジスタ[7:0]にライトして下さい。 構成されている状態なので、0 以外の値を返信する必要があります。

(h) Set Configuration リクエスト

このリクエストにより、デバイス構成を設定します。

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0y00000000	SET_CONFIGURATION	Configuration Value	Zero	Zero	None

- Default state:

USB2.0 仕様ではデバイスの動作に指定はありません。

- Address state:

wValue=0 の場合

- アドレスステートを維持します。UDC2 へのレジスタアクセスは必要ありません。

wValue≠0 で、wValue がディスクリプタに適合する Configuration 値の場合

- Address-State レジスタの bit10-8(Device State)に 0y100 をセットして下さい。
<使用する各 Endpoint に対して>
- EPx MaxPacketSize レジスタの bit10-0(max_pkt)に MaxPacketSize をセットして下さい。
- EPx Status レジスタの bit15(pkt mode)、bit14(bus sel)、bit7(dir)、bit3-2(t type)、bit1-0(num mf)に各値をセットして下さい。
- 該当 Endpoint に EP-Reset コマンドを発行して下さい。

wValue≠0 で、wValue がディスクリプタに適合しない Configuration 値の場合

- EP0 へ EP-Stall コマンドを発行して下さい。

- Configured state:

wValue=0 の場合

- Address-State レジスタの bit10-8(Device State)に 0y010 をセットして下さい。
- All-EP-Invalid コマンドを発行して下さい。

wValue≠0 で、ディスクリプタに適合する Configuration 値の場合

<使用する各 Endpoint に対して>

- EPx MaxPacketSize レジスタの bit10-0(max_pkt)に MaxPacketSize をセットして下さい。
- EPx Status レジスタの bit15(pkt mode)、bit14(bus sel)、bit7(dir)、bit3-2(t type)、bit1-0(num mf)に各値をセットして下さい。
- 該当 Endpoint に EP-Reset コマンドを発行して下さい。

<新たに未使用となる各 Endpoint に対して>

- 該当 Endpoint に EP-Invalid コマンドを発行して下さい。

wValue≠0 で、wValue がディスクリプタに適合しない Configuration 値の場合

- EP0 へ EP-Stall コマンドを発行して下さい。

(i) Get Interface リクエスト

このリクエストにより、指定された Interface で設定されている Alternate Setting 値を返信します。

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0x10000001	GET_INTERFACE	Zero	Interface	One	Alternate Setting

- 全ステート共通:
wIndex で指定された Interface が存在しない場合は EP0 へ EP-Stall コマンドを発行して下さい。
- Default state:
USB2.0 仕様ではデバイスの動作に指定はありません。
- Address state:
EP0 へ EP-Stall コマンドを発行して下さい。
- Configured state:
wIndex で指定された Interface の現在の AlternateSetting 値を、EP0 FIFO レジスタ [7:0] にライトして下さい。

(j) Set Interface リクエスト

このリクエストにより、指定された Interface の Alternate Setting 値を設定します。

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0y00000001	SET_INTERFACE	Alternate Setting	Interface	Zero	None

- 全ステート共通:

wIndex で指定された Interface が存在しない場合、または wValue で指定された Alternate Setting が存在しない場合は、EP0 へ EP-Stall コマンドを発行して下さい。

- Default state:

USB2.0 仕様ではデバイスの動作に指定はありません。

- Address state:

EP0 へ EP-Stall コマンドを発行して下さい。

- Configured state:

<指定された Interface の Alternate Setting で使用する各 Endpoint に対して>

- EPx_MaxPacketSize レジスタの bit10-0(max_pkt)に MaxPacketSize をセットして下さい。

- EPx_Status レジスタの bit15(pkt mode)、bit14(bus_sel)、bit7(dir)、bit3-2(t type)、bit1-0(num_mf)に各値をセットして下さい。

- 該当 Endpoint に EP-Reset コマンドを発行して下さい。

<新たに未使用となる各 Endpoint に対して>

- 該当 Endpoint に EP-Invalid コマンドを発行して下さい。

(k) Synch Frame リクエスト

このリクエストにより、Endpoint の同期フレームを設定し報告します。

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0y10000010	SYNCH_FRAME	Zero	Endpoint	Two	Frame Number

- 全ステート共通:

wIndex で指定された Endpoint で、このリクエストをサポートしない場合は EP0 へ EP-Stall コマンドを発行して下さい。

- Default state:

USB2.0 仕様ではデバイスの動作に指定はありません。

- Address state:

EP0 へ EP-Stall コマンドを発行して下さい。

- Configured state:

wIndex で指定された Endpoint の Frame Number を、EP0_FIFO レジスタにライトして下さい。

- Endpoint0 以外の Endpoint

Endpoint0 以外の Endpoint は、Bulk(送信/受信)転送、Interrupt(送信/受信)転送、Isochronous(送信/受信)転送をサポートしており、データの送受信を行うのに使用します。また、デュアルパケットモードをサポートしており、高速なデータ通信が可能となっています。

3.16.3.6 サスペンド / レジューム状態

UDC2 はホストからの信号状態によりサスペンド状態へと移行します。また、ホストからのレジュームあるいは UDC2 からのレジュームにより、サスペンド状態からの復帰をします。

以下にそれぞれの状態への移行について説明します。

(1) サスペンド状態への移行

通常の状態において、ホストは一定時間ごと(HS: 125 μ s、FS: 1ms)に SOF を発行しますが、ホストがデバイスをサスペンド状態にしようとした際には、この SOF をデバイスへ発行しなくなり、USB 信号線上のデータはアイドル状態のまま無変化の状態となります。UDC2 は、PHY からの“line_state”を常にモニタしており、アイドル状態を 3ms 以上連続で検出するとサスペンド状態か USB_RESET かの判断をし、サスペンド状態であれば“suspend_x”を“L”にアサートして、サスペンド状態へ入ります。

なお、サスペンド状態では PHY からの CLK が供給されなくなりますので、レジスタへのアクセス等は不可能となります。

(2) サスペンド状態からの復帰

サスペンド状態からの復帰には、ホストからのレジューム状態出力による復帰と、UDC2 からのリモートウェイクアップ(レジューム状態出力)による復帰の 2 種類があります。

以下にそれぞれの場合について説明します。

(3) ホストからのレジュームによる復帰

ホストからのレジューム状態出力により UDC2 は“suspend_x”を“H”にデアサートして、サスペンドからの復帰を知らせます。

(4) UDC2 からのリモートウェイクアップによる復帰

リモートウェイクアップはアプリケーションによってはサポートされない場合があります、また、バスエナミュレーション時に USB ホストから許可されている必要があります。システムで許可されていない場合は、“wakeup”をアサートしないで下さい。

システムで許可されている場合、“wakeup”端子をアサートすることにより UDC2 はホストに対しレジューム状態を出力し、レジュームを促します。なお、サスペンド時は PHY からのクロックが停止していますので、クロックが再開されるまでアサートし続けて下さい。また suspend_x が“L”にアサートされてから 2ms 以上経ってから、リモートウェイクアップするようにしてください。

3.16.4 USB-Spec2.0 デバイスコントローラAppendix

3.16.4.1 Appendix A システム・パワー・マネージメント関連

USB では Bulk 転送などの基本転送以外にも、ホストからのリセットやサスペンドなど、エnumレーション/パワーコントロールに関係する信号(DDP/DDM)動作が存在します。これらの処理には UDC2 動作の他にも、接続する USB2.0-PHY の仕様やシステムとしての CLK 制御等が関係してきますので、この Appendix で処理の概略を説明します。ただし、各処理の詳細に关しましては必ず Universal Serial Bus Specification(Revision2.0)(USB-Spec2.0)、PHY 仕様、UTMI Specification

(Version 1.05)を確認して下さい。

- *1) リセット … USB ホスト(以下、ホスト)から USB デバイス(以下、デバイス)を初期化するために行われる DDP/DDM 信号動作です。このリセット終了後にエnumレーションが行われ、その後、Bulk 転送などの通常転送が始まります。コネクト後には必ずリセットが行われますが、デバイスとしてはその他の任意のタイミングでのリセットにも対応する必要があります。なお、リセット中に Chirp 動作も行われ、その結果により HS か FS が選択されます。
- *2) サスペンド … ホストから 3ms 以上、SOF も含めた DDP/DDM 上の全てのバス・アクティビティがない場合には、デバイスはパワー制御のためサスペンド・モードに入る必要があります。デバイスとしては CLK を停止するなどの処理が必要となります。
- *3) レジューム … サスペンド動作から復帰をする場合に行われる DDP/DDM 信号動作です。ホストとデバイスのいずれかから復帰を促すことが可能です。デバイスからのレジュームは“リモート・ウェイクアップ”と呼ばれます。

1. コネクト/ディスコネクト動作

(1) コネクト動作

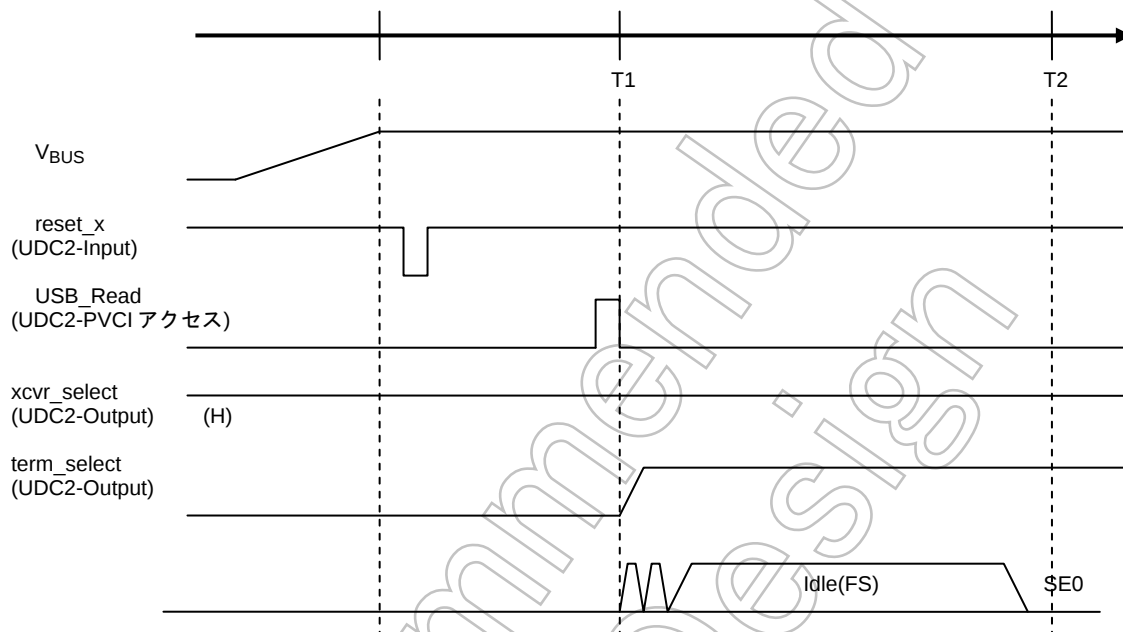


図 3.16.26 コネクト動作タイミング

- T0 … Vbus 検知**
 Vbus を検知したら、UDC2 へのシステム・リセット(reset_x 入力)を行って下さい。
 xcvr_sel は“H”、term_select は“L”となります。
- T1 … デバイス・コネクト (T0 から、100ms 以内)*4) *4)…0内の数値は USB-Spec2.0 の値**
 デバイスは Vbus 検知(T0)から 100ms 以内に、ホストにコネクト状態を知らせるため DDP のプルアップをイネーブルとする必要があります。よって、システムは Vbus を検知した後に、ホストとの通信が可能となった時点で、UDC2 の Command レジスタへアクセスし USB_Ready を設定して下さい。UDC2 は USB_Ready を受け付けると、term_select を“H”にします。これを受けて USB2.0-PHY は DDP のプルアップをイネーブルにします。
- T2 … USB リセット開始 (T1 から、100ms 以上)**

(2) ディスコネクト動作

ディスコネクトを検知した場合には、UDC2 ヘシステムリセットを行うことを推奨します。

2. リセット動作 *1

※ここでの“リセット”は USB2.0-Spec で定義されている“Reset Signaling”を表しています。

UDC2 へのシステムリセット(reset_x)とは異なります。

(1) リセット後、HS で動作する場合

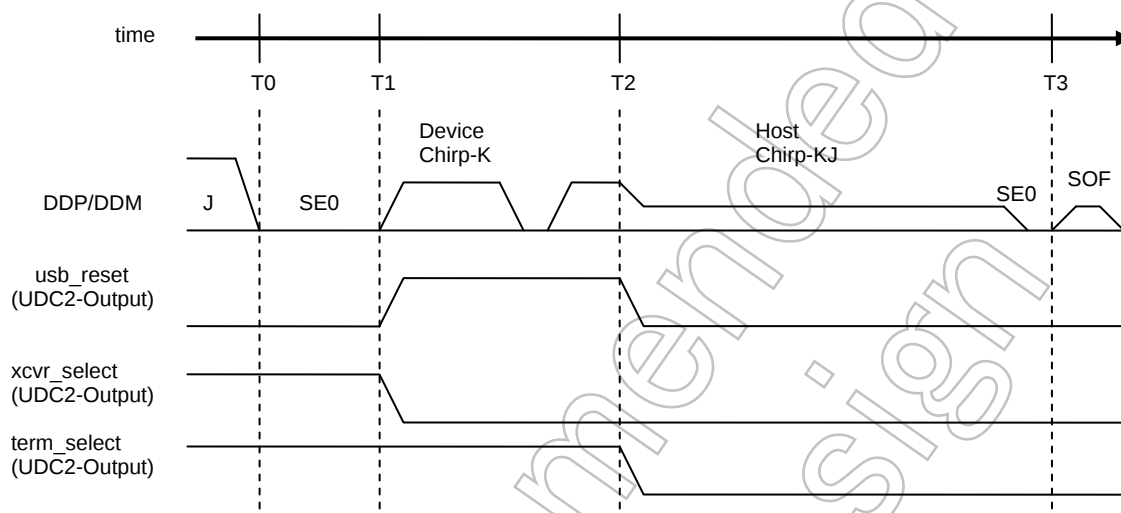


図 3.16.27 リセット(Chirp 後 HS)動作タイミング図

- T0 …… リセット開始

UDC2 はホストからの“SE0”を認識した時点で、リセットを認識するためカウントを始めます。

- T1 …… リセット認識 (T0 から、2.5 μ s 以上)

UDC2 は T0 から約 68 μ s 以上の“SE0”を検知すると、ホストからのリセットを認識して usb_reset を“H”にします。また、同時に UTMI 側ではデバイス Chirp-K 動作を開始します。

- T2 …… HS 動作開始 (T1 から、約 1.74ms~2ms)

ホストが HS 対応だった場合、UDC2 はホストからの Chirp-KJ を検知し、T1 から約 1.74ms~2ms の間で usb_reset を“L”にします(usb_reset が“H”となる期間はホストに依存します)。また、この時点から UDC2 は HS として動作します。

- T3 …… リセット終了 (T0 から、10ms 以上)

ホストからの Chirp-KJ 終了後、SE0 を経てパケット(SOF)が送信されてきます。この時点でリセットが終了となります。ホストからのリセット期間は最小で 10ms となります。

(2) リセット後、FS で動作する場合

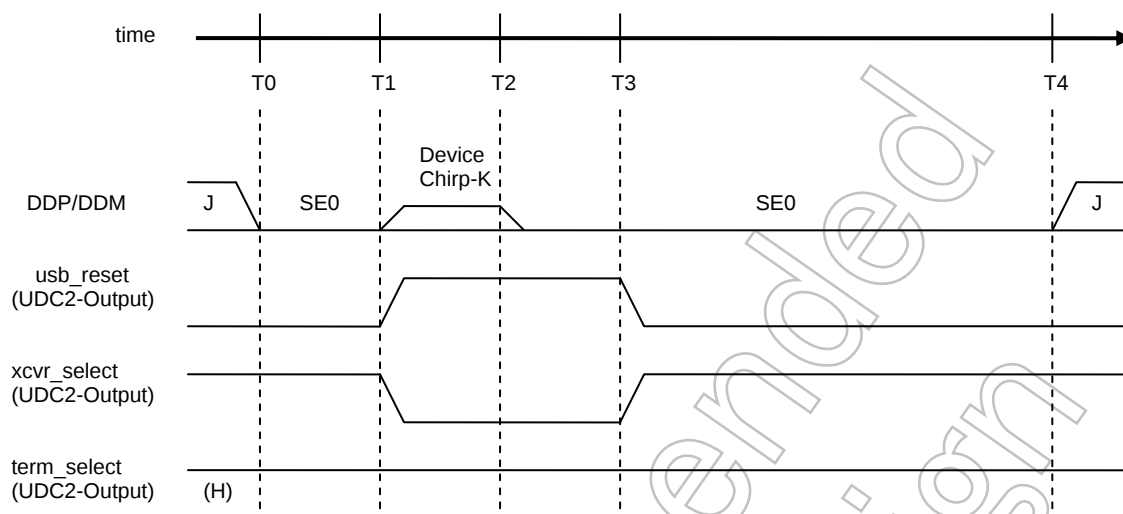


図 3.16.28 リセット(Chirp 後 FS)動作タイミング図

- T0 …… リセット開始**
 UDC2 はホストからの“SE0”を認識した時点で、リセットを認識するためカウントを始めます。
- T1 …… リセット認識 (T0 から、2.5 μ s 以上)**
 UDC2 は T0 から約 68 μ s 以上の“SE0”を検知すると、ホストからのリセットを認識して usb_reset を“H”にします。また、同時に UTMI 側ではデバイス Chirp-K 動作を開始します。
- T2 …… デバイス Chirp-K 終了 (T1 から、1.0ms 以上)**
 UDC2 は T1 から約 1.5ms 後にデバイス Chirp-K 動作を終了します。
- T3 …… FS 動作開始 (T2 から、1.0ms~2.5ms)**
 ホストが FS 対応だった場合、ホスト Chirp-KJ 動作はありません。UDC2 は T2 から約 2ms の間までにホスト Chirp-KJ を検知しない場合に FS に入ります。この時点で usb_reset を“L”にします。usb_reset が“H”となる期間は約 3.5ms となります。
- T4 …… リセット終了 (T0 から、10ms 以上)**
 ホストからの“SE0”が終了し idle に入った時点でリセットが終了となります。ホストからのリセット期間は最小で 10ms となります。

(3) リセット時の注意点

- リセット後のレジスタ初期化について

ホストからのリセットが終了した時点(`usb_reset`が“H”から“L”へ変化した時点)で、UDC2 内部レジスタは全て初期化されます。(各レジスタの初期値はUDC2「3.16.3.3.レジスタ構成」を参照して下さい。)

`usb_reset` が“H”の間に設定されたレジスタも初期化されますので、リセット後の各レジスタの再設定はリセット終了後に行ってください。

- リセット後の DMA 転送(Endpoint-I/F アクセス)について

DMA 転送中に、ホストからのリセットを受信した場合には、`EPx_Status` レジスタの初期化に伴いバスアクセス選択が「共通バスアクセス」となります。従って、正常に DMA 転送を続けることはできなくなります。リセット受信時には DMA コントローラの初期化も行ってください。

リセット後にはエニュメレーションが発生しますので、その際に各 Endpoint の設定を行ったうえ Command レジスタの `EP_Reset` コマンドを用いて Endpoint の初期化を行ってください。

3. サスペンド動作 *2

(1) HS 時のサスペンド動作

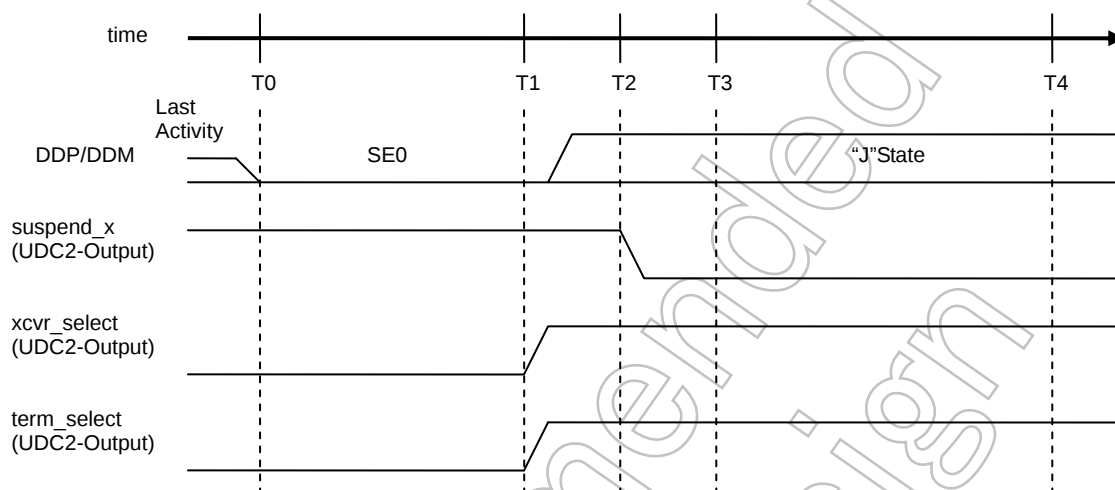


図 3.16.29 HS 時のサスペンド動作タイミング

- T0 …… バスアクティビティの終了**
 ホストからの最後のバスアクティビティ(パケットの終り)を検知した時点で、UDC2 はサスペンド/リセットを認識するためにカウントを始めます。
- T1 …… FS への移行 (T0 から、3.0ms~3.125ms)**
 T0 から、3ms 以上 “SE0”を検知した時、UDC2 は FS へ移行し xcvr_select と term_select を“H”にします。(これを受けて USB2.0-PHY は DDP のプルアップをイネーブルにします。) この時点では UDC2 は、ホストからサスペンド/リセットのどちらが来ているかは判断できません。
- T2 …… サスペンド認識 (T1 から、100μs~875μs)**
 UDC2 は、T1 から約 110μs の間に line state= “J”を検知した場合、サスペンドを認識して suspend_x を “L”とします。line state が “J”に入らずに “SE0”を保持した場合、UDC2 はサスペンドではなくリセットに入る準備をします。この場合は「2.リセット動作」を参照して下さい。
- T3 …… リモートウェイクアップ開始可能 (T0 から、5ms)**
 T0 から 5ms が経過した時点で、デバイスからのレジューム(=リモートウェイクアップ)が可能となります。「4.(2) デバイスレジューム(リモートウェイクアップ)動作」を参照して下さい。
- T4 …… サスペンドへの移行 (T0 から、10ms)**
 T0 から 10ms 以内にデバイスとしてサスペンドに入る必要があります。USB2.0-PHY から CLK を停止させる場合など、デバイスシステムとしてサスペンドに必要な処理を行う場合には、この期間内に行って下さい。

(2) FS 時のサスペンド動作

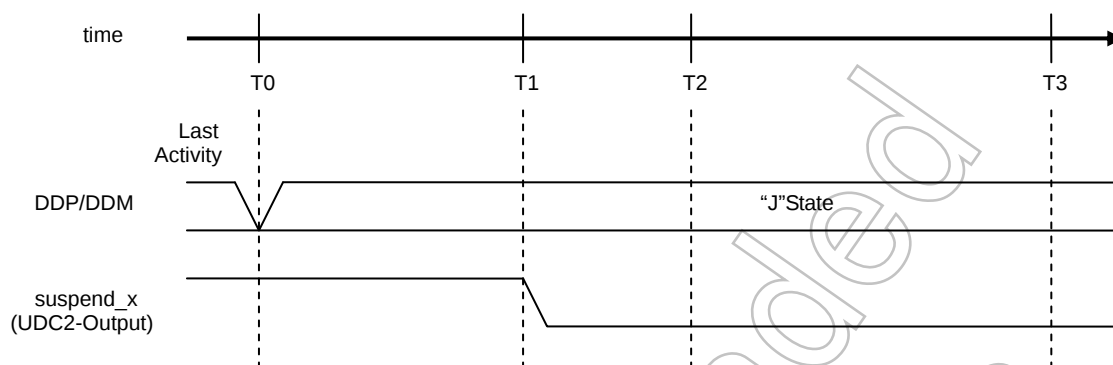


図 3.16.30 FS 時のサスペンド動作タイミング

- T0 … バスアクティビティの終了**
 ホストからの最後のバスアクティビティ(パケットの終り)を検知した時点で、UDC2 はサスペンドを認識するためにカウントを始めます。
- T1 … サスペンド認識 (T0 から、3ms)**
 T0 から 3ms 以上の“FS-J”を検知した時、UDC2 はサスペンドを認識して `suspend_x` を“L”とします。
- T2 … リモートウェイクアップ開始可能 (T0 から、5ms)**
 T0 から 5ms が経過した時点で、デバイスからのレジューム(=リモートウェイクアップ)が可能となります。「4. (2) デバイス・レジューム(リモート・ウェイクアップ)動作」を参照して下さい。
- T3 … サスペンドへの移行 (T0 から、10ms)**
 T0 から 10ms 以内にデバイスとしてサスペンドに入る必要があります。USB2.0-PHY からの CLK を停止させる場合など、デバイスシステムとしてサスペンドに必要な処理を行う場合には、この期間内に行って下さい。

(3) サスペンド時の注意点

● サスペンド時の CLK の扱いについて

通常、サスペンド時に USB2.0-PHY への SuspendM 入力(UTMI)をイネーブルとした場合、PHY から UDC2 へ供給される CLK が停止されます。UDC2 が suspend_x を “L”にした後も PHY から供給される CLK を使用したい場合、suspend_x を直接 PHY へは接続せず、システムで PHY からの CLK を停止しても良いと判断した時点で PHY への SuspendM 入力をイネーブルとする必要があります。

(なお、UDC2 への CLK 入力(30MHz)が停止された場合、PVC1-I/F、Endpoint-I/F を通じての UDC2 内部レジスタアクセスが不可能となります。)

● USB2.0-PHY の CLK 制御について

USB2.0-PHY の SuspendM 入力(UTMI)へは UDC2 の suspend_x 出力を直に接続するのではなく、システムで SuspendM を制御可能として下さい。前項までの説明のように、UDC2 の suspend_x は USB ホストとの通信によって動作します。従って、例えば USB 未使用時(USB ホストと未接続時)には、suspend_x は HW リセット後の値である “H”を保持することになります。この時に suspend_x と SuspendM を直に接続していると、USB2.0-PHY の CLK は停止しないためシステムの消費電力に影響します。

● サスペンド時の内部レジスタについて

サスペンド中、UDC2 は内部レジスタの値や FIFO の中身、各フラグの状態を保持します。レジュームによりサスペンドから復帰した後も同様にサスペンド前の状態を保持しています。

4. レジューム動作 *3

(1) ホストレジューム動作

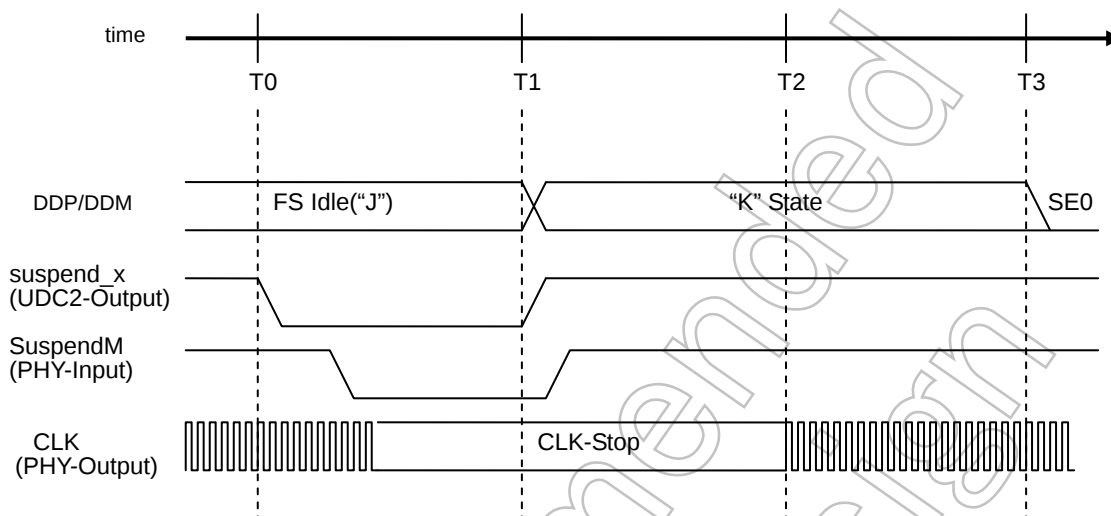


図 3.16.31 ホストレジューム動作タイミング

- T0 … UDC2 の suspend_x 出力が“L”
- T1 … ホストレジューム開始 (タイミング規定なし)
- T2 … USB2.0-PHY からの CLK 再開 (PHY 仕様に依存)
- T3 … ホストレジューム終了 (T1 から 20ms 以上)

ホストはサスペンドから復帰させるため、任意のタイミングでレジューム(“FS-K”)を開始します。UDC2 はこの時点で suspend_x を“H”にします。(UDC2 への CLK 入力停止している状態でも、suspend_x は“H”になります。) suspend_x=“H”を確認したら、USB2.0-PHY からの CLK 出力を再開させるために、PHY への SuspendM(UTMI)入力をディセーブルとして下さい。

ホストからのレジューム(“FS-K”)は 20ms 以上続きます。その後、“SE0”を経てレジューム動作が終了します。なおレジューム終了後、UDC2 はサスペンドに入る前の動作スピード(HS/FS)で復帰します。

(2) デバイス・レジューム(リモート・ウェイクアップ)動作

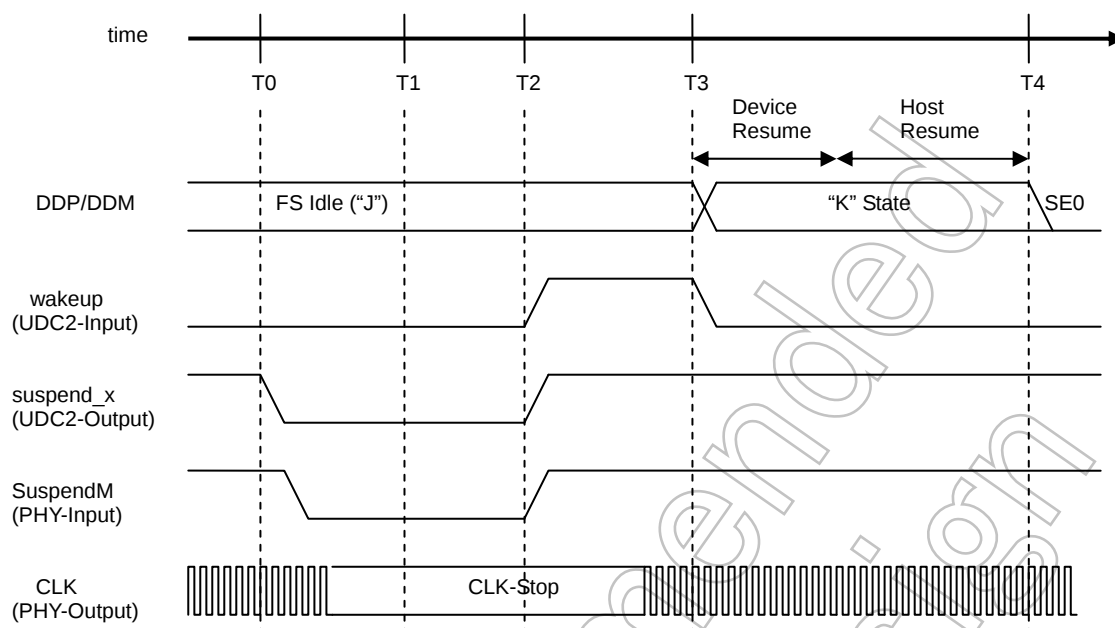


図 3.16.32 リモートウェイクアップ動作タイミング

- T0 … UDC2 の suspend_x 出力が“L”
- T1 … リモートウェイクアップ開始可能 (T0 から、2ms 以上)

デバイスからサスペンド復帰をする場合に、UDC2 の wakeup 入力を使用することで、リモートウェイクアップが可能です。ただし、USB-Spec 上、サスペンド開始から 5ms はリモートウェイクアップが禁止されています。サスペンド開始から T0 までに 3ms は経過しているため、T0 から 2ms 以上経ってから wakeup を“H”にしてください。

- T2 … UDC2 への wakeup 入力を“H” (T1 以降)

任意のタイミングで wakeup を“H”にしてください。UDC2 はこの時点で suspend_x を“H”にします。(UDC2 への CLK 入力が停止している状態でも、suspend_x は“H”になります。) また、UDC2 が実際にレジューム(“FS-K”)を開始するためには、UDC2 へ CLK が入力されている必要がありますので、USB2.0-PHY への SuspendM(UTMI)入力をディセーブルとしてください。その後、CLK が再開されるまで、wakeup を“H”にし続けてください。

- T3 … デバイスレジューム開始 (PHY-Spec に依存)

PHY から UDC2 への CLK 入力が再開されると、UDC2 はデバイス・レジューム(“FS-K”)を開始します。UDC2 のデバイスレジューム期間は約 2ms となっています。ホストはデバイスレジュームを確認するとホスト・レジュームを開始します。

- T4 … ホストレジューム終了 (T3 から 20ms 以上)

ホストからのレジューム(“FS-K”)は 20ms 以上続きます。その後、“SE0”を経てレジューム動作が終了します。なおレジューム終了後、UDC2 はサスペンドに入る前の動作スピード(HS/FS)で復帰します。

(3) レジューム時の注意点

• リモートウェイクアップの使用制限について

リモートウェイクアップをデバイスシステムとしてサポートする場合には、**Configuration** ディスクリプタの中で、デバイスとしてリモートウェイクアップ機能がイネーブルであることをホストに伝えなければいけません。さらに、サポートする場合でも、リモートウェイクアップの使用許可はデフォルトではディセーブルとなっています。ホストからのリクエストによりイネーブルとされるまではこの機能は使用してはいけません。これらの制約を満たしている場合に限り **wakeup** 入力を使用してのリモートウェイクアップを行って下さい。

この仕様の詳細はUSB-Spec2.0 の主に 3.16.3.6に記載してありますので、使用する場合は必ず参照して下さい。

Not Recommended
for New Design

3.16.4.2 Appendix B MaxPacketSize奇数バイト設定関連

1. EPx_MaxPacketSize レジスタでの奇数設定について

USB-Spec 上、Isochronous/Interrupt 転送では Endpoint の MaxPacketSize(以下、MPS)を偶数/奇数バイトのどちらかに設定可能です。(Control/Bulk 転送では偶数のみを設定可能です。)

UDC2 では MPS の設定は、EPx_MaxPacketSize レジスタの max_pkt(bit10-0)で行います。UDC2 で実装する Endpoint の FIFO 構成としては偶数バイトしかサポートしていないので、通常は MPS も偶数バイトを設定することを推奨します。

もしMPSを奇数バイトとして使用したい場合には、max_pktに奇数バイトに設定することも可能ですが、Endpointのバスアクセス方法によって、max_pkt設定値に以下の表 3.16.3のような制限があります。つまり、送信用Endpointでの直接アクセスではmax_pktに奇数バイトを設定できません。この場合には、max_pktへは偶数バイトを設定し、Endpoint-FIFOへのライトアクセスで最大ライトバイト数をMPS(奇数)に制御する必要があります。(例えば、MPSを 1023 バイトにする場合は、max_pktに 1024 バイトを設定する。)

表 3.16.3 max_pkt 設定制限

	受信用 Endpoint	送信用 Endpoint
共通バスアクセス(PVCI-I/F)	偶数/奇数を設定可能	偶数/奇数を設定可能
Endpoint 直接アクセス(Endpoint-I/F)	偶数/奇数を設定可能	偶数のみを設定可能

上記を踏まえて、次項より、各バスアクセス方法に応じた奇数バイトの MPS 設定方法について説明します。

2. 受信用 Endpoint & 共通バスアクセス

max_pkt に奇数/偶数バイトの両方を設定可能です。アクセス方法も奇数/偶数バイトで同じです。

3. 送信用 Endpoint & 共通バスアクセス

max_pkt に奇数/偶数バイトの両方を設定可能です。但し、max_pkt に奇数バイトを設定して、MPS のライトを行う場合には、以下の点に気をつけて共通バスアクセスを行って下さい。

以下は max_pkt=5 を設定し、MPS である 5 バイトをライトする場合のアクセス例です。

- 最後の 5 バイト目のアクセスでは、必ず udc_be= 0y01 として下さい。
- MPS のアクセスなので、Command レジスタでの EP_EOP コマンドは発行しないで下さい。

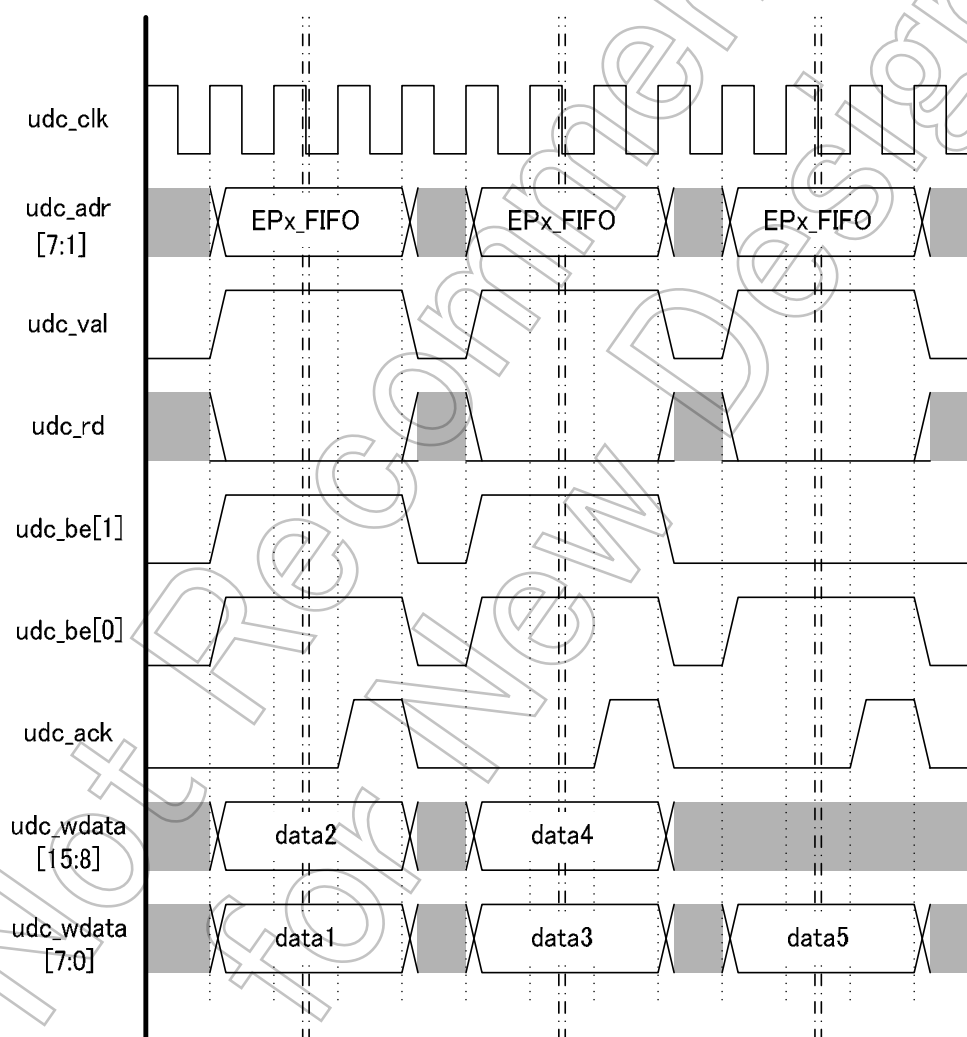


図 3.16.33 max_pkt=奇数で MPS ライト時のアクセス（共通バスアクセス）

4. 受信用 Endpoint & Endpoint 直接アクセス

max_pkt に奇数/偶数バイトの両方を設定可能です。アクセス方法も奇数/偶数バイトで同じです。

5. 送信用 Endpoint & Endpoint 直接アクセス

max_pkt に偶数バイトのみを設定可能です。

USB 上、Endpoint の MPS を奇数バイトとして動作させる場合は以下のように設定して下さい。

(MPS=1023 で使用する例)

- max_pkt=1024 を設定して下さい。
- Endpoint への最大ライトアクセスは 1023 バイトとして下さい。
- (MPS を超える 1024 バイトはライトしないで下さい。)
- ファームウェアで管理する必要がある Endpoint ディスクリプタの “wMaxPacketSize” は、1023 を設定して下さい。(この値は Get Descriptor リクエストで USB-HOST へ伝える情報です。)

以下は max_pkt=1024 を設定し、MPS である 1023 バイトをライトする場合のアクセス例です。

- 最後の 1023 バイト目のアクセスでは、必ず epx_w_be=0y01 として下さい。

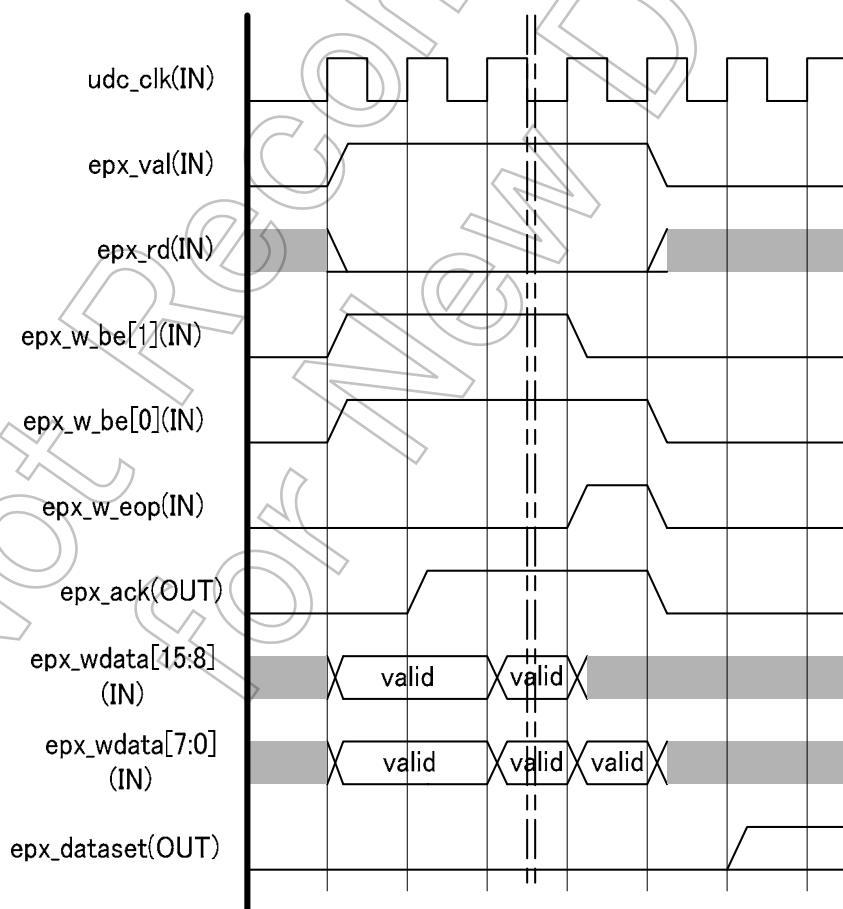


図 3.16.34 max_pkt=偶数で MPS(奇数)ライト時のアクセス (Endpoint 直接アクセス)

3.17 I²S (Inter-IC Sound)

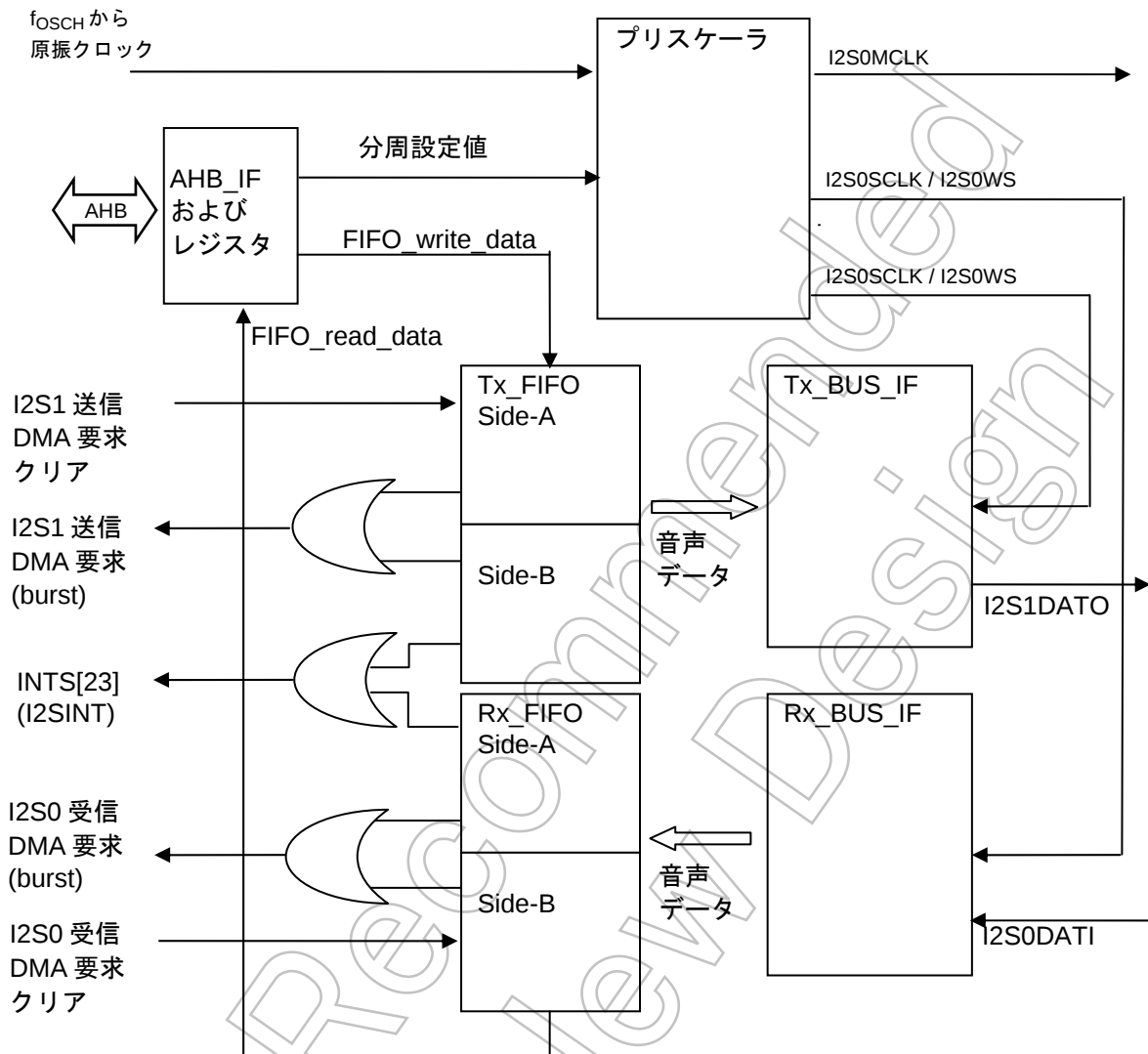
I²S フォーマット互換のシリアル入出力回路を内蔵しています。

外部に AD コンバータまたは、DA コンバータなどの音声用 LSI を接続することによりデジタルオーディオシステムなどに使用できます。下記に特長を記述します。

表 3.17.1 動作の特長

	全二重 MASTER モード 全二重 SLAVE モード クロックスルーモード	
チャンネル	チャンネル 0	チャンネル 1
送受信	受信専用	送信専用
データフォーマット	(1) I ² S フォーマット互換 (2) ステレオ/モノラル対応 (3) MSB first/LSB first の選択が可能 (4) 左寄せ対応可(WS と同期、遅延 OFF)	
使用端子	(1) I2S0SCLK (クロック入出力) (2) I2S0DATI (データ入力) (3) I2S0WS (ワード選択入出力) (4) I2S0MCLK (マスタクロック出力)	(1) I2S1DATO (データ出力)
クロック関係	(1) Master クロックと I2S0WS を 256/384/512 倍の関係に設定可能 (2) ソースクロックとして、内部クロック(X1)のみ (3) ソースクロックを 1, 1/2, 1/4 の 3 種類に分周して Master クロックを生成可能	
FIFO バッファ	2×8 ワード	2×8 ワード
データ長	16bit のみ	16bit のみ
割り込み	FIFO の Over Flow と Under Flow 割り込み	FIFO の Over Flow と Under Flow 割り込み

3.17.1 ブロック図



3.17.2 各モードの説明

I²S 回路は、全二重 MASTER モード,全二重 SLAVE モードを使用して、同時に送受信を実現可能です。

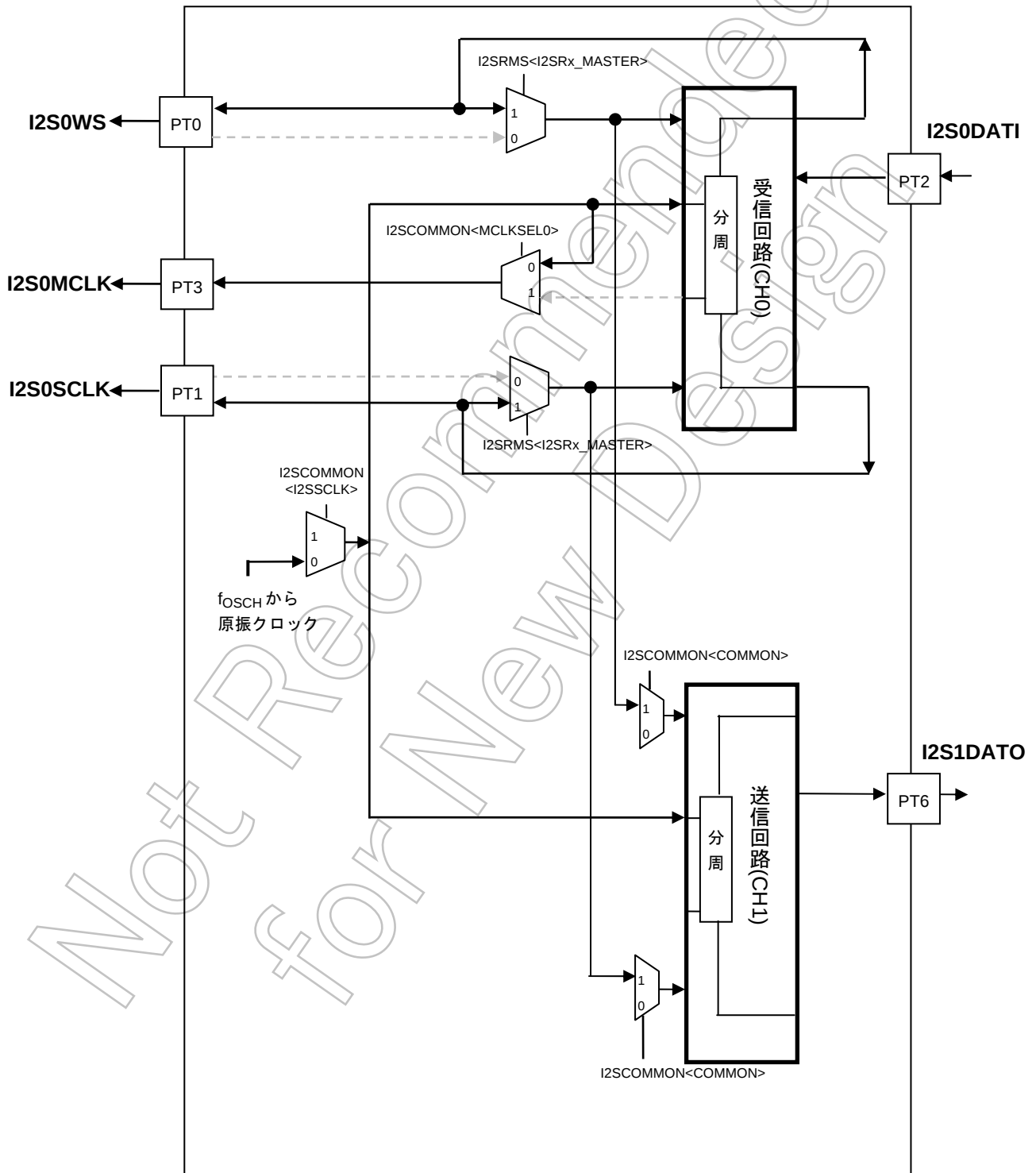
次ページ以降に、各々のモードに関する説明をいたします。

Not Recommended
for New Design

3.17.2.1 モードの使用例 1

(全二重 Master, <I2SSCLK> = 0, <MCLKSEL0> = 0, <COMMON> = 1, <I2SRx_MASTER> = 1)

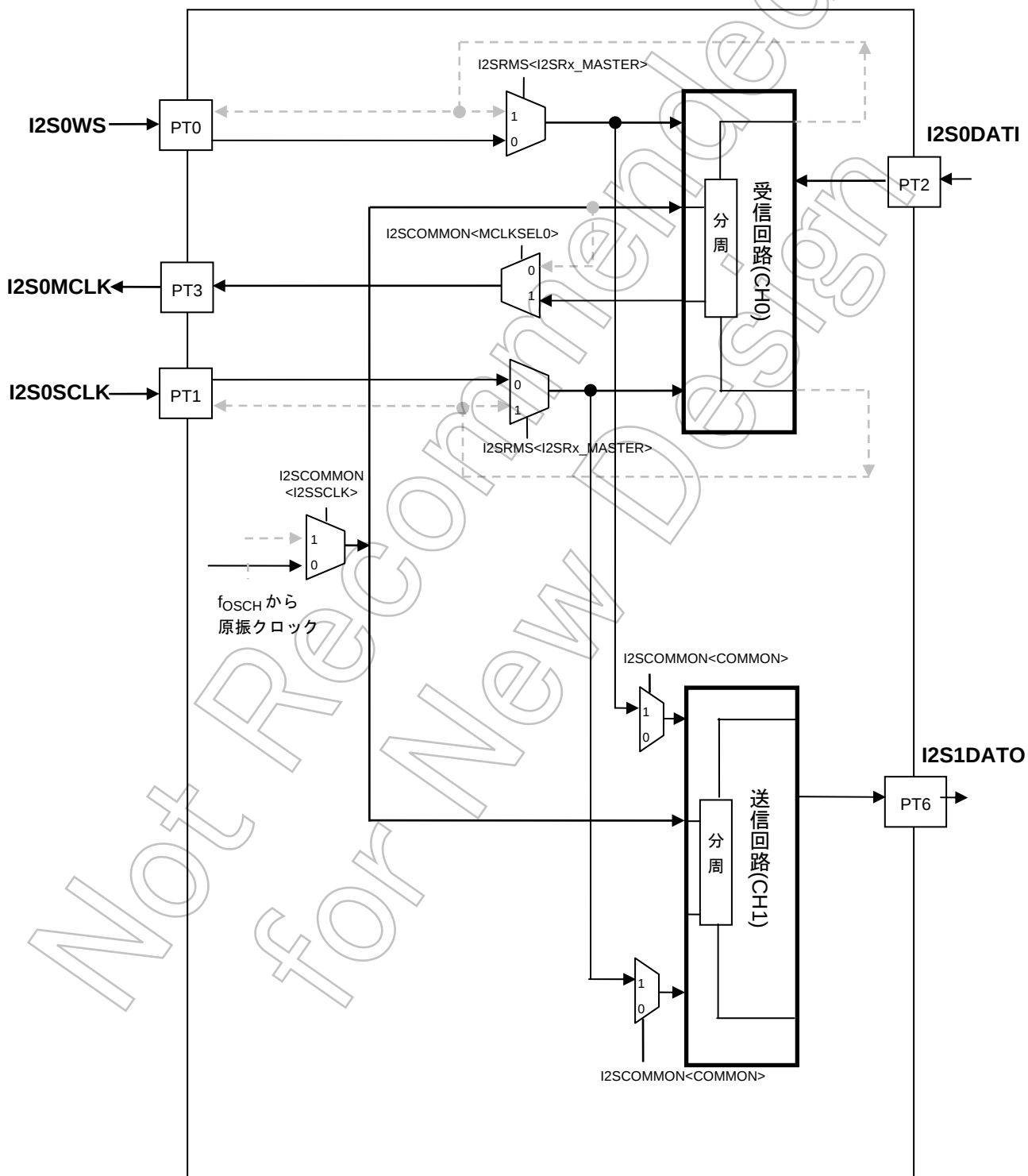
受信回路(Ch0)を Master に設定し、受信回路が出力する I2S0WS および I2S0SCLK に、送信回路(Ch1)が同期した形で、送受信を行うモードです。



3.17.2.2 モードの使用例 2

(全二重 Slave, <I2SSCLK> = 0, <MCLKSEL0> = 1, <COMMON> = 1,
<I2SRx_MASTER> = 0)

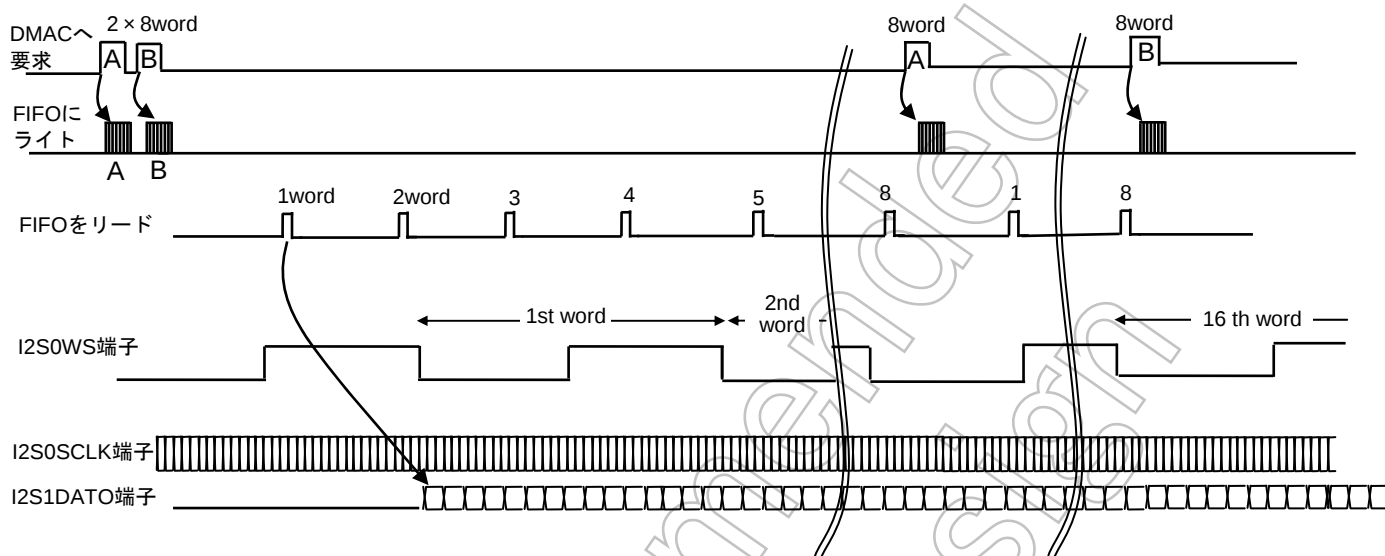
受信回路(Ch0)を Slave に設定し、他の Master が出力する I2S0WS および I2S0SCLK に、送信回路 (Ch1)が同期した形で、送受信を行うモードです。



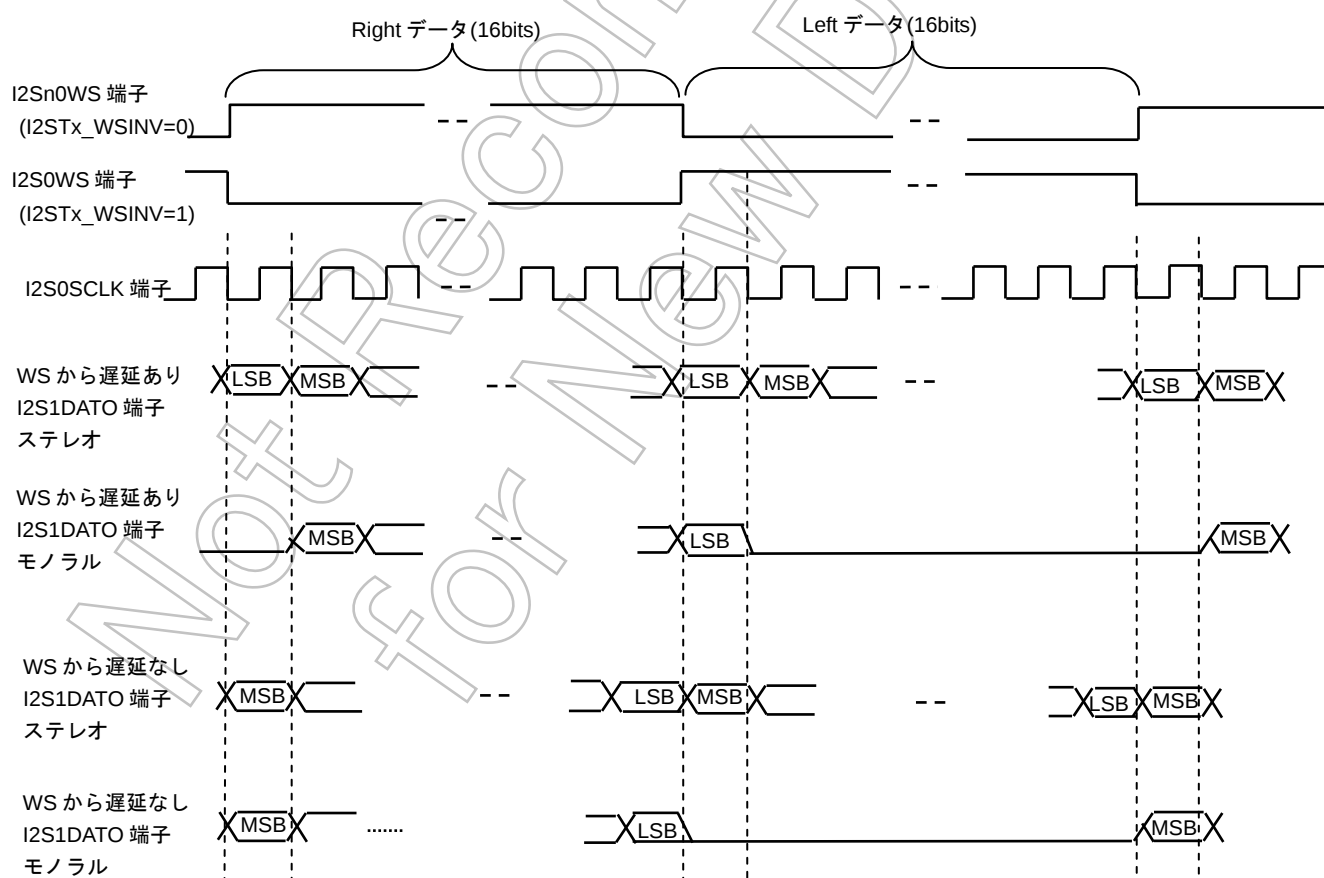
3.17.3 動作説明

3.17.3.1 I²S出力フォーマット

(I2STCON<I2STx_DELAYOFF>= 0, WS より遅延あり)



全体のタイミング図



詳細なタイミング図

3.17.4 レジスタの説明

SFR のリストを以下に示します。

base アドレス = 0xF204_0000

Register Name	Address (base+)	Description
I2STCON	0x0000	Tx Control Register
I2STSLVON	0x0004	Tx I ² S Slave Control Register
I2STFCLR	0x0008	Tx FIFO Clear Register
–	0x000C	Reserved
–	0x0010	Reserved
–	0x0014	Reserved
I2STDMA1	0x0018	Tx DMA Ready Register
–	0x001C	Reserved
I2SRCON	0x0020	Rx Control Register
I2SRSLVON	0x0024	Rx I ² S Slave Control Register
I2SRFCLR	0x0028	Rx FIFO Clear Register
I2SRMS	0x002C	Rx Master/Slave Select Register
I2SRMCON	0x0030	Rx Master I2S0WS/I2S0SCLK Period Register
I2SRMSTP	0x0034	Rx Master Stop Register
I2SRDMA1	0x0038	Rx DMA Ready Register
–	0x003C	Reserved
I2SCOMMON	0x0044	Common WS/SCK and Loop Setting Register
I2STST	0x0048	I ² S Tx Status Register
I2SRST	0x004C	Rx Status Register
I2SINT	0x0050	I ² S Interrupt Register
I2SINTMSK	0x0054	I ² S Interrupt Mask Register
I2STDAT	0x1000~0x1FFF	Transmit FIFO Window DMA Target
I2SRDAT	0x2000~0x2FFF	Receive FIFO Window DMA Target

1. I2STCON (Tx Control Register)

Address = (0xF204_0000) + (0x0000)

Bit	Bit Symbol	Type	Reset Value	Description
[31:14]	–	–	Undefined	Read as undefined. Write as zero.
[13:12]	I2STx_RLCH_CUT	R/W	0y00	ステレオ/モノラル出力設定 0y00:ステレオ設定(両 CH から出力) 0y01:モノラル設定(右片側 CH のみ出力) 0y10:モノラル設定(左片側 CH のみ出力) 0y11: 設定しないでください。
[11:9]	–	–	Undefined	Read as undefined. Write as zero.
[8]	I2STx_BITCNV	R/W	0y0	MSB(符号ビット)の反転: 0y0: 反転しない 0y1: 反転
[7:4]	–	–	Undefined	Read as undefined. Write as zero.
[3]	I2STx_UNDERFLOW	R/W	0y0	FIFO UnderFlow 時のデータ出力 0y0: 0 出力 0y1: 直前データ保持
[2]	I2STx_MSBINV	R/W	0y0	LSB/MSB ファースト 0y0: MSB ファースト 0y1: LSB ファースト
[1]	I2STx_WSINV	R/W	0y0	WS の CH 定義反転 0y0: WS=1(RCH), WS=0(LCH) 0y1: WS の CH 定義反転 WS=0(RCH), WS=1(LCH)
[0]	I2STx_DELAYOFF	R/W	0y0	データ出力の開始タイミングと WS との関係 0y0: WS から 1CLOCK 遅延あり 0y1: WS から遅延なし

(個別説明)

a. <I2STx_RLCH_CUT>

ステレオ、モノラル(右片側 CH 出力、左片側 CH 出力)出力の設定をします。

0y00: ステレオ設定(両 CH から出力)

0y01: モノラル設定(右片側 CH のみ出力)

0y10: モノラル設定(左片側 CH のみ出力)

0y11: 設定しないでください。

b. <I2STx_BITCNV>

MSB(符号ビット)の反転の設定をします。Right と Left 両方とも反転されます。

0y0: 反転しない

0y1: 反転

c. <I2STx_UNDERFLOW>

出力 FIFO 内部の有効データが空になっても出力は継続されます。その際に出力されるデータを定義しています。(FIFO UnderFlow 時の出力データ)

0y0: 0 出力

0y1: 直前データを保持

d. <I2STx_MSBINV>

LSB/MSB ファーストの選択をします。

0y0: MSB ファースト

0y1: LSB ファースト

e. <I2STx_WSINV>

WS の CH 定義反転の設定をします。

0y0: WS 信号が High 出力にて Right channel と定義

WS 信号が Low 出力にて、Left Channel と定義

0y1: WS の CH 定義反転

WS 信号が Low 出力にて Right channel と定義

WS 信号が High 出力にて、Left Channel と定義

f. <I2STx_DELAYOFF>

データ出力の開始タイミングと WS との関係を設定します。

0y0: WS から 1CLOCK 遅延あり

0y1: WS から遅延なし

2. I2SRCON (Rx Control Register)

Address = (0xF204_0000) + (0x0020)

Bit	Bit Symbol	Type	Reset Value	Description
[31:14]	–	–	Undefined	Read as undefined. Write as zero.
[13:12]	I2SRx_RLCH_CUT	R/W	0y00	ステレオ/モノラル出力設定 0y00:ステレオ設定(両 CH から出力) 0y01:モノラル設定(右片側 CH のみ出力) 0y10:モノラル設定(左片側 CH のみ出力) 0y11: 設定しないでください。
[11:9]	–	–	Undefined	Read as undefined. Write as zero.
[8]	I2SRx_BITCNV	R/W	0y0	MSB(符号ビット)の反転: 0y0: 反転しない 0y1: 反転
[7:3]	–	–	Undefined	Read as undefined. Write as zero.
[2]	I2SRx_MSBINV	R/W	0y0	LSB/MSB ファースト 0y0: MSB ファースト 0y1: LSB ファースト
[1]	I2SRx_WSINV	R/W	0y0	WS の CH 定義反転 0y0: WS= 1(RCH), WS= 0(LCH) 0y1: WS の CH 定義反転 WS= 0(RCH), WS= 1(LCH)
[0]	I2SRx_DELAYOFF	R/W	0y0	データ出力の開始タイミングと WS との関係 0y0: WS から 1CLOCK 遅延あり 0y1: WS から遅延なし

(個別説明)

a. <I2SRx_RLCH_CUT>

ステレオ、モノラル(右片側 CH 出力、左片側 CH 出力)出力の設定をします。

0y00:ステレオ設定(両 CH から出力)

0y01:モノラル設定(右片側 CH のみ出力)

0y10:モノラル設定(左片側 CH のみ出力)

0y11: 設定しないでください。

b. <I2SRx_BITCNV>

MSB(符号ビット)の反転の設定をします。

0y0: 反転しない

0y1: 反転

- c. <I2SRx_UNDERFLOW>
FIFO UnderFlow 時の出力データの選択をします。
0y0: 0 出力
0y1: 直前データを保持
- d. <I2SRx_MSBINV>
LSB/MSB ファーストの選択をします。
0y0: MSB ファースト
0y1: LSB ファースト
- e. <I2SRx_WSINV>
WS の CH 定義反転の設定をします。
0y0: WS = 1 (RCH), WS = 0 (LCH)
0y1: WS の CH 定義反転
WS = 0 (RCH), WS = 1 (LCH)
- f. <I2SRx_DELAYOFF>
データ出力の開始タイミングと WS との関係を設定します。
0y0: WS から 1CLOCK 遅延あり
0y1: WS から遅延なし

3. I2STSLVON (Tx I²S Slave Control Register)

Address = (0xF204_0000) + (0x0004)

Bit	Bit Symbol	Type	Reset Value	Description
[31:1]	—	—	Undefined	Read as undefined. Write as zero.
[0]	I2STx_SLAVE	R/W	0y0	送信 FIFO イネーブル 0y0: OFF 0y1: ON (FIFO 読み出しが可能)

(個別説明)

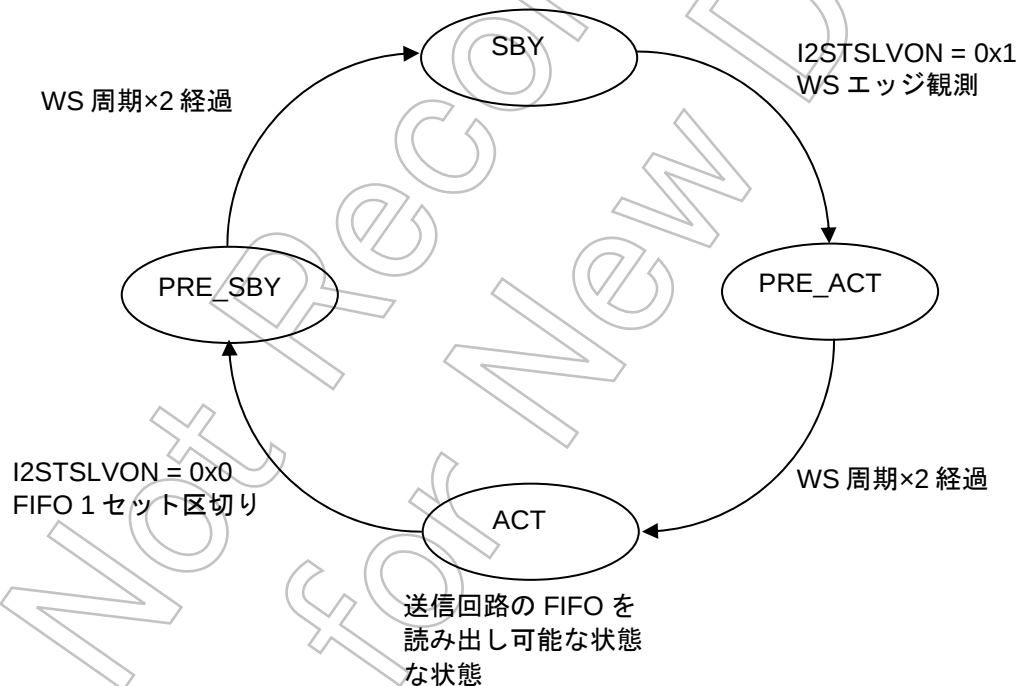
a. <I2STx_SLAVE>

0→1 とする事で内部状態(I2STST<I2STxSTATUS>)が SBY(スタンバイ)→PRE_ACT→ACT と遷移します。

ACT 状態になると FIFO に格納されたデータの出力を開始します。

1→0 とする事で内部状態(I2STST<I2STxSTATUS>)が ACT→PRE_SBY→SBY と遷移します。SBY になるとデータが格納されていても FIFO 内のデータは出力されません。

送信回路ステートマシンフロー



注) 現在の内部動作状態は I2STST<I2STxSTATUS>をリードすることで確認してください。

4. I2SRSLVON (Rx I²S Slave Control Register)

Address = (0xF204_0000) + (0x0024)

Bit	Bit Symbol	Type	Reset Value	Description
[31:1]	—	—	Undefined	Read as undefined. Write as zero.
[0]	I2SRx_SLAVE	R/W	0y0	受信 FIFO イネーブル 0y0: OFF 0y1: ON (FIFO 書き込み動作が可能)

(個別説明)

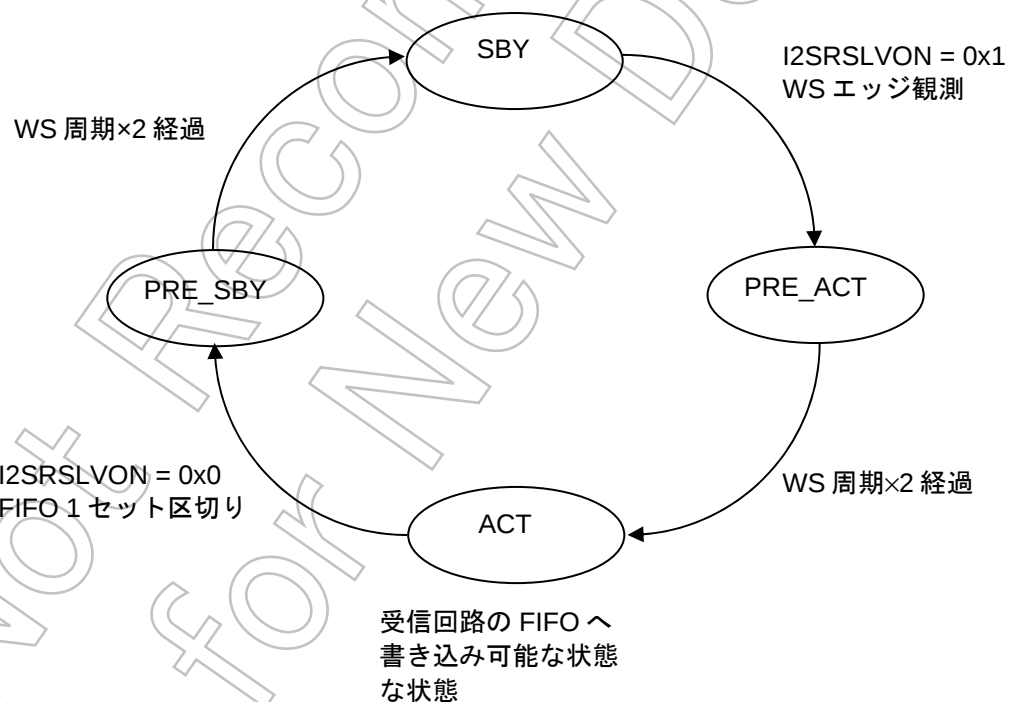
a. <I2SRx_SLAVE>

0→ 1 とする事で内部状態 (I2SRST<I2SRxSTATUS>) が SBY(スタンバイ)→PRE_ACT→ACT と遷移します。

ACT 状態になると FIFO への取り込みを開始します。

1→ 0 とする事で内部状態 (I2SRST<I2SRxSTATUS>) が ACT→PRE_SBY→SBY と遷移します。SBY になるとデータが入力されていても FIFO への取り込みを行いません。

受信回路ステートマシンフロー



注) 現在の内部動作状態は I2SRST<I2SRxSTATUS>をリードすることで確認してください。

5. I2STFCLR (Tx FIFO Clear Register)

Address = (0xF204_0000) + (0x0008)

Bit	Bit Symbol	Type	Reset Value	Description
[31:1]	—	—	Undefined	Read as undefined. Write as zero.
[0]	I2STx_FIFOCCLR	R/W	0y0	FIFO ポインタのクリア 0y0: 無効 0y1: FIFO ポインタクリア

(個別説明)

a. <I2STx_FIFOCCLR>

送信データが破壊される可能性がありますので、DMA 動作中には FIFO クリアは実行しないでください。

リードすると常に 0 になります。

6. I2SFRFCLR (Rx FIFO Clear Register)

Address = (0xF204_0000) + (0x0028)

Bit	Bit Symbol	Type	Reset Value	Description
[31:1]	—	—	Undefined	Read as undefined. Write as zero.
[0]	I2SRx_FIFOCCLR	R/W	0y0	FIFO ポインタのクリア 0y0: 無効 0y1: FIFO ポインタクリア

(個別説明)

a. <I2SRx_FIFOCCLR>

受信データが破壊される可能性がありますので、DMA 動作中には FIFO クリアは実行しないでください。

リードすると常に 0 になります。

7. I2SRMS (Rx Master/Slave Select Register)

Address = (0xF204_0000) + (0x002C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:1]	—	—	Undefined	Read as undefined. Write as zero.
[0]	I2SRx_MASTER	R/W	0y0	Master/Slave 選択 0y0: Slave 0y1: Master(内部生成 I2S0WS/I2S0SCLK を外部へ出力)

(個別説明)

a. <I2SRx_MASTER>

受信マスタ/受信スレーブの切換えを行います。

I2SCOMMON<COMMON>= 1 に設定します。全 2 重マスタ/全 2 重スレーブの切換えとなります。

8. I2SRMCON (Rx Master I2S0WS/I2S0SCLK Period Register)

Address = (0xF204_0000) + (0x0030)

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	—	—	Undefined	Read as undefined. Write as zero.
[3:2]	I2SRx_WS_DIV[1:0]	R/W	0y0	ソースクロックと I2S0MCLK の比 0y00: 1/1 0y01: 1/2 0y10: 1/4 0y11: 禁止
[1:0]	I2SRx_SCLK_DIV[1:0]	R/W	0y0	I2S0MCLK と I2S0WS の比 0y00: 1/8 0y01: 1/12 0y10: 1/16 0y11: 禁止

(個別説明)

- a. <I2SRx_WS_DIV[1:0]>, <I2SRx_SCLK_DIV[1:0]>

I2SCOMMON<COMMON>= 0y1 に設定します。全 2 重モードになります。

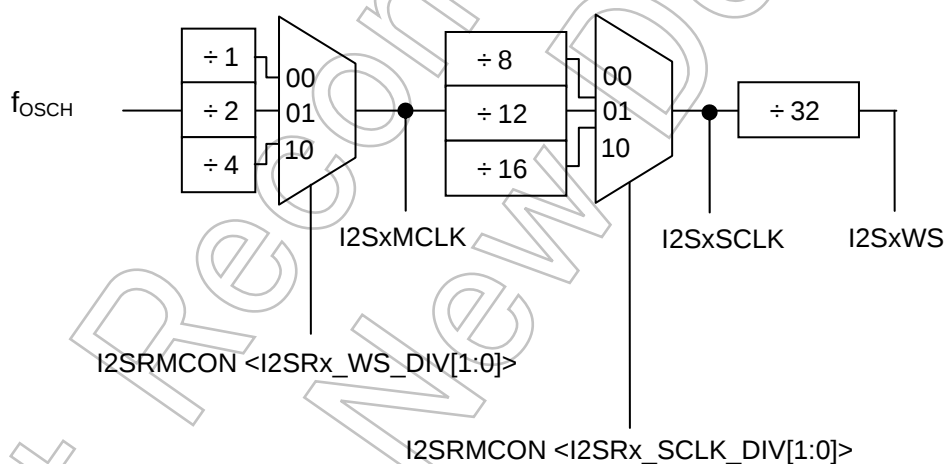


表 3.17.2 クロック関係図

I2SRMCON I2SRx_WS_DIV[1:0]	クロックソースと I2S0MCLK 比率	クロックソースと I2S0SCLK 比率	I2SRMCON I2SRx_SCLK_DIV[1:0]	クロックソースと I2S0WS 比率
0y00: (1/1 設定)	1/1	1/8	0y00: (1/256 設定)	1/256
0y01: (1/2 設定)	1/2	1/16	0y00: (1/256 設定)	1/512
0y10: (1/4 設定)	1/4	1/32	0y00: (1/256 設定)	1/1024
0y00: (1/1 設定)	1/1	1/12	0y01: (1/384 設定)	1/384
0y01: (1/2 設定)	1/2	1/24	0y01: (1/384 設定)	1/768
0y10: (1/4 設定)	1/4	1/48	0y01: (1/384 設定)	1/1536
0y00: (1/1 設定)	1/1	1/16	0y10: (1/512 設定)	1/512
0y01: (1/2 設定)	1/2	1/32	0y10: (1/512 設定)	1/1024
0y10: (1/4 設定)	1/4	1/64	0y10: (1/512 設定)	1/2048

表 3.17.3 48kHz 基準 Audio サンプリング設定例

f_{OSCH}	I2SRMCON I2SRx_WS_DIV[1:0]	I2S0MCLK 周波数 (クロックソース倍率)	I2S0SCLK 周波数 (クロックソース倍率)	I2SRMCON I2SRx_SCLK_DIV[1:0]	I2S0WS 周波数 (クロックソース倍率)
12.288MHz	0y00: (1/1 設定)	12.288MHz (1/1)	1536kHz (1/8)	0y00: (1/256 設定)	48kHz (1/256)
	0y01: (1/2 設定)	6.144MHz (1/2)	768kHz (1/16)	0y00: (1/256 設定)	24kHz (1/512)
	0y10: (1/4 設定)	3.072MHz (1/4)	384kHz (1/32)	0y00: (1/256 設定)	12kHz (1/1024)
18.432MHz	0y00: (1/1 設定)	18.432MHz (1/1)	1536kHz (1/12)	0y01: (1/384 設定)	48kHz (1/384)
	0y01: (1/2 設定)	9.216MHz (1/2)	768kHz (1/24)	0y01: (1/384 設定)	24kHz (1/768)
	0y10: (1/4 設定)	4.608MHz (1/4)	384kHz (1/48)	0y01: (1/384 設定)	12kHz (1/1536)
24.576MHz	0y00: (1/1 設定)	24.576MHz (1/1)	1536kHz (1/16)	0y10: (1/512 設定)	48kHz (1/512)
	0y01: (1/2 設定)	12.288MHz (1/2)	768kHz (1/32)	0y10: (1/512 設定)	24kHz (1/1024)
	0y10: (1/4 設定)	6.144MHz (1/4)	384kHz (1/64)	0y10: (1/512 設定)	12kHz (1/2048)

9. I2SRMSTP (Rx Master Stop Register)

Address = (0xF204_0000) + (0x0034)

Bit	Bit Symbol	Type	Reset Value	Description
[31:1]	—	—	Undefined	Read as undefined. Write as zero.
[0]	I2SRx_MSTOP	R/W	0y0	I2SRx Master STOP 0y0: I2S0WS/I2S0SCLK 生成 0y1: I2S0WS/I2S0SCLK を STOP

(個別説明)

a. <I2SRx_MSTP>

マスタの I2S0WS/I2S0SCLK を STOP(= Low レベル固定)させたい時に用います。通常利用しません。

I2SRx の内部ステータス「SBY」状態(I2SRST<I2SRx_STATUS[1:0]>=0y00)を確認して設定して下さい。それ以外は動作を保証しません。

この設定は Default で解除状態なのでマスタ関連設定を実行すると即 I2S0WS/I2S0SCLK が出力されます。

10. I2STDMA1 (Tx DMA Ready Register)

Address = (0xF204_0000) + (0x0018)

Bit	Bit Symbol	Type	Reset Value	Description
[31:1]	—	—	Undefined	Read as undefined. Write as zero.
[0]	I2STx_DMAREADY1	R/W	0y0	I2STx DMA 起動 0y0: Disable 0y1: Enable

(個別説明)

a. <I2STx_DMAREADY1>

DMA 起動のための設定完了した状態を、ハードへ示すためのレジスタです。

DMA 起動の設定および I²S の設定が完了した後にソフトウェアで 1 に設定することで起動準備ができたことをハードへ示します。この時、ハードウェアは FIFO を監視して DMA を開始します。

注) このレジスタをクリアする場合は、DMA 終了を DMA の SFR にて確認後、I2STSLVON を 0x0 に設定し、I2STST にて SBY 状態であることを確認してから 0x0 を設定してください。

11. I2SRDMA1 (Rx DMA Ready Register)

Address = (0xF204_0000) + (0x0038)

Bit	Bit Symbol	Type	Reset Value	Description
[31:1]	–	–	Undefined	Read as undefined. Write as zero.
[0]	I2SRx_DMAREADY1	R/W	0y0	I2SRx DMA 起動 0y0: Disable 0y1: Enable

(個別説明)

a. <I2SRx_DMAREADY1>

DMA 起動のための設定完了した状態を、ハードへ示すためのレジスタです。

DMA 起動の設定および I²S の設定が完了した後にソフトウェアで 1 に設定することで起動準備ができたことをハードへ示します。この時、ハードウェアは FIFO を監視して DMA を開始します。

注) このレジスタをクリアする場合は、DMA 終了を DMA の SFR にて確認後、I2SRSLVON を 0x0 に設定し、I2SRST にて SBY 状態であることを確認してから 0x0 を設定してください。

12. I2SCOMMON (Common WS/SCK and Loop Setting Register)

Address = (0xF204_0000) + (0x0044)

Bit	Bit Symbol	Type	Reset Value	Description
[31:6]	–	–	Undefined	Read as undefined. Write as zero.
[5]	Reserved	WO	0y0	Read as undefined. Write as zero.
[4]	MCLKSEL0	WO	0y0	受信回路出力のマスタクロック選択 0y0: 音声用ソースクロック 0y1: 音声用ソースクロックを分周したクロックを出力
[3]	MCLKSEL1	WO	0y0	送信回路出力のマスタクロック選択 0y0: 音声用ソースクロック 0y1: 音声用ソースクロックを分周したクロックを出力
[2]	I2SSCLK	WO	0y0	音声ソースクロック: 0y0: f_{OSCH} 0y1: Reserved
[1]	LOOP	R/W	0y0	LOOP 設定 0y0: LOOP をしない 0y1: LOOP
[0]	COMMON	R/W	0y0	全二重モード設定 0y0: Reserved 0y1: 全二重、必ず 0y1 を設定して下さい。

(個別説明)

a. <MCLKSEL0>

受信回路出力のマスタクロックを選択します。

0y0: 音声用ソースクロック

0y1: 音声用ソースクロックを分周したクロックを出力

b. <MCLKSEL1>

送信回路出力のマスタクロックを選択します。

0y0: 音声用ソースクロック

0y1: 音声用ソースクロックを分周したクロックを出力

c. <I2SSCLK>

音声ソースクロックを選択します。

0y0: f_{OSCH}

0y1: Reserved

d. <LOOP>

受信端子から入力されたシリアルデータが送信回路を経て送信端子から出力される、LOOP 機能を設定します。

0y0: LOOP をしない

0y1: LOOP

a. <COMMON>

<COMMON>= 1 にすると、Rx 側の SCK/WS 入力信号が、I2STx 側でも用いられるようになります。これに伴い、マスタ送信側の設定は無効になります。

0y0: Reserved

0y1: 全二重、必ず 0y1 を設定して下さい

Not Recommended
for New Design

13. I2STST (I²S Tx Status Register)

Address = (0xF204_0000) + (0x0048)

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	—	—	Undefined	Read as undefined.
[3:2]	I2STx_STATUS[1:0]	RO	0y00	FIFO のステータス: 0y00: SBY 0y01: PreACT 0y10: PreSBY 0y11: ACT
[1]	I2STx_FIFOFULL	RO	0y0	FIFO の FULL 状態 0y0: FULL ではない 0y1: FULL
[0]	I2STx_FIFOEMPTY	RO	0y1	FIFO の空き状態: 0y0: 空きではない 0y1: 空き

(個別説明)

- a. <I2STx_STATUS[1:0]>
FIFO のステータスを示します。
0y00: SBY
0y01: PreACT
0y10: PreSBY
0y11: ACT
- b. <I2STx_FIFOFULL>
FIFO の FULL 状態を示します。
0y0: FULL ではない
0y1: FULL
- c. <I2STx_FIFOEMPTY>
FIFO の空き状態を示します。
0y0: 空きではない
0y1: 空き

14. I2SRST (Rx Status Register)

Address = (0xF204_0000) + (0x004C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	—	—	Undefined	Read as undefined.
[3:2]	I2SRx_STATUS[1:0]	RO	0y00	FIFO 書込みのステータス: 0y00: SBY 0y01: PreACT 0y10: PreSBY 0y11: ACT
[1]	I2SRx_FIFOFULL	RO	0y0	FIFO の FULL 状態 0y0: FULL ではない 0y1: FULL
[0]	I2SRx_FIFOEMPTY	RO	0y1	FIFO の空き状態: 0y0: 空きではない 0y1: 空き

(個別説明)

a. <I2SRx_STATUS [1:0]>

FIFO 書込みのステータス を示します。

0y00: SBY

0y01: PreACT

0y10: PreSBY

0y11: ACT

b. <I2SRx_FIFOFULL>

FIFO の FULL 状態を示します。

0y0: FULL ではない

0y1: FULL

c. <I2SRx_FIFOEMPTY>

FIFO の空き状態を示します。

0y0: 空きではない

0y1: 空き

15. I2SINT (I²S Interrupt Register)

Address = (0xF204_0000) + (0x0050)

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	—	—	Undefined	Read as undefined. Write as zero.
[3]	I2SRx_OVERFLOW_INT	R/W	0y0	Rx FIFO Over Flow 割り込み: リード時 0y0: 割り込みなし 0y1: 割り込み有り ライト時 0y0: 無効 0y1: クリア
[2]	I2SRx_UNDERFLOW_INT	R/W	0y0	Rx FIFO Under Flow 割り込み: リード時 0y0: 割り込みなし 0y1: 割り込み有り ライト時 0y0: 無効 0y1: クリア
[1]	I2STx_OVERFLOW_INT	R/W	0y0	Tx FIFO Over Flow 割り込み: リード時 0y0: 割り込みなし 0y1: 割り込み有り ライト時 0y0: 無効 0y1: クリア
[0]	I2STx_UNDERFLOW_INT	R/W	0y0	Tx FIFO Under Flow 割り込み: リード時 0y0: 割り込みなし 0y1: 割り込み有り ライト時 0y0: 無効 0y1: クリア

(個別説明)

- a. <I2SRx_OVERFLOW_INT>, <I2SRx_UNDERFLOW_INT>, <I2STx_OVERFLOW_INT>, <I2STx_UNDERFLOW_INT>

割り込み要因レジスタです。

FIFO 割り込み状態をモニタする場合は割り込み許可レジスタ(I2SINTMSK)の対象ビットを 0 に設定してください。

これら割り込み要因のいずれかが発生すると、割り込みコントローラに I2SINT が発生します。また、I2SINT レジスタの各割り込み要因ビットのモニターにより、I2SINT の割り込み要因を確認できます。

それぞれのビットに 1 をライトする事によって、それぞれの現状の割り込みステータスがクリアされます。

注) 割り込み許可レジスタ I2SINTMSK の対象ビットを 0 に設定した場合(disable)もこのレジスタにライトされた値が反映されます。

16. I2SINTMSK (I²S Interrupt Mask Register)

Address = (0xF204_0000) + (0x0054)

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	—	—	Undefined	Read as undefined. Write as zero.
[3]	I2SRx_OVERFLOW_INTMS	R/W	0y1	Rx FIFO Over Flow 許可設定: 0y0: 許可 0y1: 禁止
[2]	I2SRx_UNDERFLOW_INTM	R/W	0y1	Rx FIFO Under Flow 許可設定: 0y0: 許可 0y1: 禁止
[1]	I2STx_OVERFLOW_INTMS	R/W	0y1	Tx FIFO Over Flow 許可設定: 0y0: 許可 0y1: 禁止
[0]	I2STx_UNDERFLOW_INTM	R/W	0y1	Tx FIFO Under Flow 許可設定: 0y0: 許可 0y1: 禁止

(個別説明)

- a. <I2SRx_OVERFLOW_INTMS>
Rx FIFO Over Flow の許可設定をします。
0y0: 許可
0y1: 禁止
- b. <I2SRx_FIFOFULL>
Rx FIFO Under Flow の許可設定をします。
0y0: 許可
0y1: 禁止
- c. <I2SRx_FIFOEMPTY>
Tx FIFO Over Flow の許可設定をします。
0y0: 許可
0y1: 禁止
- d. <I2SRx_FIFOFULL>
Tx FIFO Under Flow の許可設定をします。
0y0: 許可
0y1: 禁止

17. I2STDAT (Transmit FIFO Window DMA Target Register)

Address = (0xF204_1000) + (0x0000)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	Left[15:0]	WO	0x0000	I2STx LEFT 音声データ [15:0]
[15:0]	Right[15:0]	WO	0x0000	I2STx Right 音声データ [15:0]

(個別説明)

a. <Left[15:0]>, <Right[15:0]>

本レジスタに設定したデータは送信 FIFO へライトされます。^{注2)}

Upper を Left Channel Data、Lower を Right Channel Data とステレオ音声時は同時に設定して下さい。0xF2041000~0xF2041FFF のアドレス空間であれば、どのアドレスにライトされても、全てライトデータとみなし、ライトされた順番に FIFO へ格納していきます。

注1) このレジスタは リードは対応していません。

注2) CPU/DMAC とともにマスタとなって本レジスタにアクセスできます。

18. I2SRDAT (Receive FIFO Window DMA Target Register)

Address = (0xF204_2000) + (0x0000)

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	Left[15:0]	RO	0x0000	I2SRx LEFT 音声データ [15:0]
[15:0]	Right[15:0]	RO	0x0000	I2SRx Right 音声データ [15:0]

(個別説明)

a. <Left[15:0]>, <Right[15:0]>

本レジスタをリードすると受信 FIFO のデータが順次リードされます。^{注1)}

Upper を Left Channel Data、Lower を Right Channel Data とし、ステレオ音声時は同時に入力されます。0xF2042000~0xF2042FFF のアドレス空間であれば、どのアドレスからリードしても、全てリードデータとみなし、FIFO から順に読み出して行きます。。

注1) このレジスタはライトは禁止です。

注2) CPU/DMAC とともにマスタとなって本レジスタにアクセスできます。

3.17.5 設定例

- 送信マスタ、受信スレーブの設定例

```

I2SCOMMON      ← 0x00000001 ; write 0x00000001 to Register
I2STCON        ← 0x00000000
I2SRCON        ← 0x00000000
I2SRMS         ← 0x00000000 ; Rx is slave

GPIOMFR1       ← 0x000000FF
GPIOLFR1       ← 0x000000FF
0xf8004000      ← 0x0000FFFF ; Set transfer data
. . .
0xf800403c     ← 0xFFFF0000 ; End of set data
. . .
. . .           ; Use DMA scatter gather link
0xf8004040     ← 0xf8004020 ; source address
0xf8004044     ← I2STDAT    ; destination address
0xf8004048     ← 0xf8004050 ; next address
0xf800404c     ← 0x04492008 ; Set DMAC control register
. . .
DMACConfiguration ← 0x00000001 ; Set Rx DMAC
DMACC0SrcAddr     ← I2SRDAT
DMACC0DestAddr    ← 0xf8008000
DMACC0Control     ← 0x08492008
DMACC0Configuration ← 0x00001017
I2SRSLVON         ← 0x00000001 ; I2S internal clock on
. . .             ; label i2sr_act
i2sr_act         I2SRST      → r0 ; read I2SRST register data to r0
LDR r1,=0xc
AND r0,r0,r1
LDR r2,=0xc
CMP r0,r2 ; check I2S Active
BNE i2sr_act ; r0 ≠ r2 , jump to i2sr_act
I2SRDMA1         ← 0x00000001 ; I2S DMA Ready

DMACConfiguration ← 0x00000001 ; Set Tx DMAC
DMACC1SrcAddr     ← 0xf8004000
DMACC1DestAddr    ← I2STDAT
DMACC1Control     ← 0x04492008
DMACC1Configuration ← 0x00000a81
I2STSLVON         ← 0x00000001 ; I2S internal clock on
. . .             ; label i2st_act
i2st_act         I2SRST      → r0 ; read I2SRST register data to r0
LDR r1,=0xc
AND r0,r0,r1
LDR r2,=0xc
CMP r0,r2 ; check I2S Active
BNE i2st_act ; r0 ≠ r2 , jump to i2st_act
I2STDMA1         ← 0x00000001 ; I2S DMA Ready
. . .             ; label
finish_DMA       DMACC0Control → r0 ; read DMACC0Control data to r0
CMP r0,#0x0 ; check the End of Rx DMAC
BNE finish_DMA ; r0 ≠ 0x0 , jump to finish_DMA
I2STDMA1         ← 0x00000000 ; DMA Clear
I2SRDMA1         ← 0x00000000
I2STSLVON        ← 0x00000000 ; internal clock off
I2SRSLVON        ← 0x00000000
. . .             ; label
i2s_stop_t       I2STST      → r0 ; read I2STST register data to r0
LDR r1,=0xc
AND r0,r0,r1
LDR r1,=0x0
CMP r0,r1
BNE i2s_stop_t ; check I2S Tx standby
. . .             ; label
i2s_stop_r       I2SRST      → r0 ; read I2SRST register data to r0
LDR r1,=0xc
AND r0,r0,r1
LDR r1,=0x0
CMP r0,r1 ; check I2S Rx standby
BNE i2s_stop_r ; r0 ≠ r1 , jump to i2s_stop_r
I2STFCLR         ← 0x00000001 ; clear Tx FIFO
I2SFRFCLR        ← 0x00000001 ; clear Rx FIFO

```

3.18 LCDコントローラ(LCDC)

3.18.1 概要

本 LSI は、カラー対応の液晶表示コントローラ(LCDC)を内蔵しております。
LCDC には、以下のような特長があります。

表 3.18-1 LCDC の特長

LCD パネルのタイプ		TFT	STN (デュアル/ シングル)
同時発色可能な表示色 (パレット色変換あり)		~256 色	2,4,16,256 色
可能表示色 (パレット色変換なし)		~65536 色	3375 色
Bit per Pixel (画素あたりのデータ量)		1/2/4/8/16 bit	1/2/4/8/16 bit
対応水平ピクセル数		16×(PPL + 1)dot : PPL の値は 0 ~63 の整数のみ 注)	
対応垂直行数		4 ~1024(整数) 注)	
転送先データバス幅 (LCD ドライバ)		最大 16 bit	4/8 bit
表示データの FIFO バッファ		32 bit × 16 word × 2	
タイミング調整機能		垂直・水平同期信号のフロント・バックポーチのタイミングをプログラム可能	
表示パレット		256 エントリ、16 ビットパレット RAM	
データ形式		リトルエンディアンのサポート	
接 続 端 子	LD15-LD0 端子	LCD ドライバ専用データバス	
	LCLAC 端子	有効データ Enable 信号	AC バイアス信号(フレーム信号)
	LCLLP 端子	水平同期信号	水平同期パルス
	LCLFP 端子	垂直同期信号	垂直同期パルス
	LCLCP 端子	LCDD のデータラッチ用クロック(パネルクロック)	
	LCLLE 端子	行終了信号	基本的に使用しません
	LCLPOWER 端子	LCD パネルパワー制御信号 注)本 LSI ではサポートしていません	

注) 表示サイズは、表示色と動作クロックによっては制限が生じます。以下を参考にしてください。

LCD タイプ	表示可能な最大 dot 数
TFT 16bit Color	約 350000dot (640 × 480 程度)
TFT 8bit Color	約 500000dot (800 × 600 程度)
TFT 4bit Color	特に制約なし
TFT 2bit Color	特に制約なし
STN 15bit Color	約 350000dot (640 × 480 程度)
STN 8bit Color	約 700000dot (960 × 640 程度)
STN 4bit Color	特に制約なし
STN 2bit Color	特に制約なし
STN 1bit Color(Monochrome)	特に制約なし

3.18.2 機能

図 3.18.1 は、LCDC の簡単なブロック図を示しています。

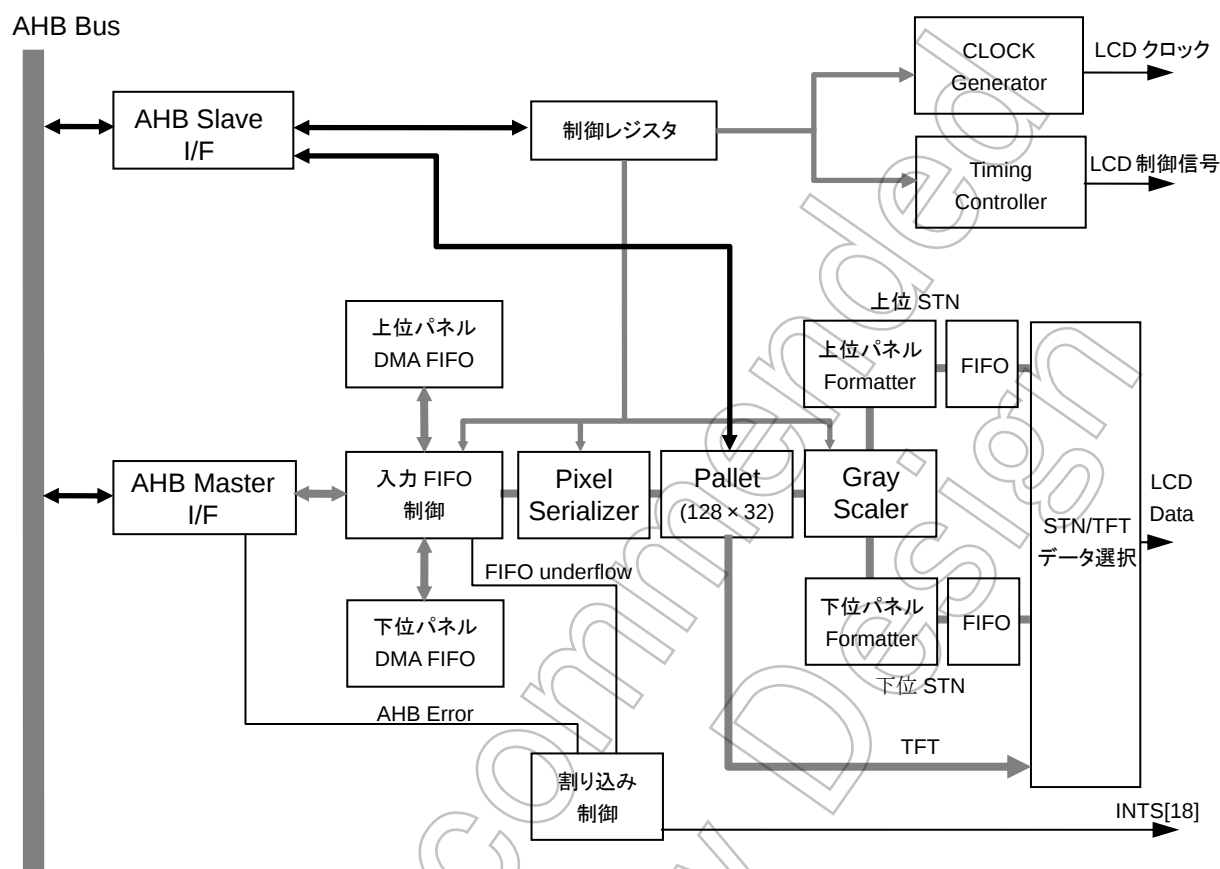


図 3.18.1 LCDC ブロック図

次ページから、各ブロックの説明を致します。

3.18.2.1 デュアル DMA FIFO および関連制御ロジック

FIFO の入力ポートは AMBA AHB インタフェースに接続し、出力ポートはピクセルシリアルライザに接続されています。

表示データは、表示 RAM から READ されると、シングル/デュアルパネル LCD タイプに適合するように、個別に制御可能な 2 つの DMA FIFO にバッファリングされます。

32Word 分の FIFO を使用できます。FIFO は、WATERMARK レジスタ設定によって、4 word 以上の空き、または、8Word 以上の空きが発生した時点で、データを要求します。

また、FIFO が空の状態では LCD データを出力すると、アンダーフロー条件が発生し、割り込み信号がアサートされます。

Not Recommended
for New Design

3.18.2.2 ピクセルシリアライザ

このブロックは、DMA FIFO の出力ポートから 32 ビット幅の LCD データを読み出し、動作モードに応じて 24、16、8、4、2、または 1 ビットのデータに変換します。

デュアルパネルモードでは、上位 DMA FIFO(16word) と下位 DMA FIFO(16word)に分割され、交互に読み出されます。

適切なサイズに変換されたデータは、パレット RAM 内のカラー/グレイ階調値として、使用されるか、または、パレットを経由せず、直接出力されます。

表 3.18-2 LBLP : DMA FIFO 出力ビット 31 ~ 16

bpp	DMA FIFO 出力ビット															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
1	P31	P30	P29	P28	P27	P26	P25	P24	P23	P22	P21	P20	P19	P18	P17	P16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	P15		P14		P13		P12		P11		P10		P9		P8	
	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
4	P7				P6				P5				P4			
	3	2	1	0	3	2	1	0	3	2	1	0	3	2	1	0
8	P3								P2							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
16	P1															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
24	P0															
	-	-	-	-	-	-	-	-	23	22	21	20	19	18	17	16

表 3.18-3 LBLP : DMA FIFO 出力ビット 15 ~ 0

bpp	DMA FIFO 出力ビット															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	P7		P6		P5		P4		P3		P2		P1		P0	
	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
4	P3				P2				P1				P0			
	3	2	1	0	3	2	1	0	3	2	1	0	3	2	1	0
8	P1								P0							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
16	P0															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
24	P0															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

3.18.2.3 RAM パレット

16 ビット×256 エントリのパレットを内蔵しています。

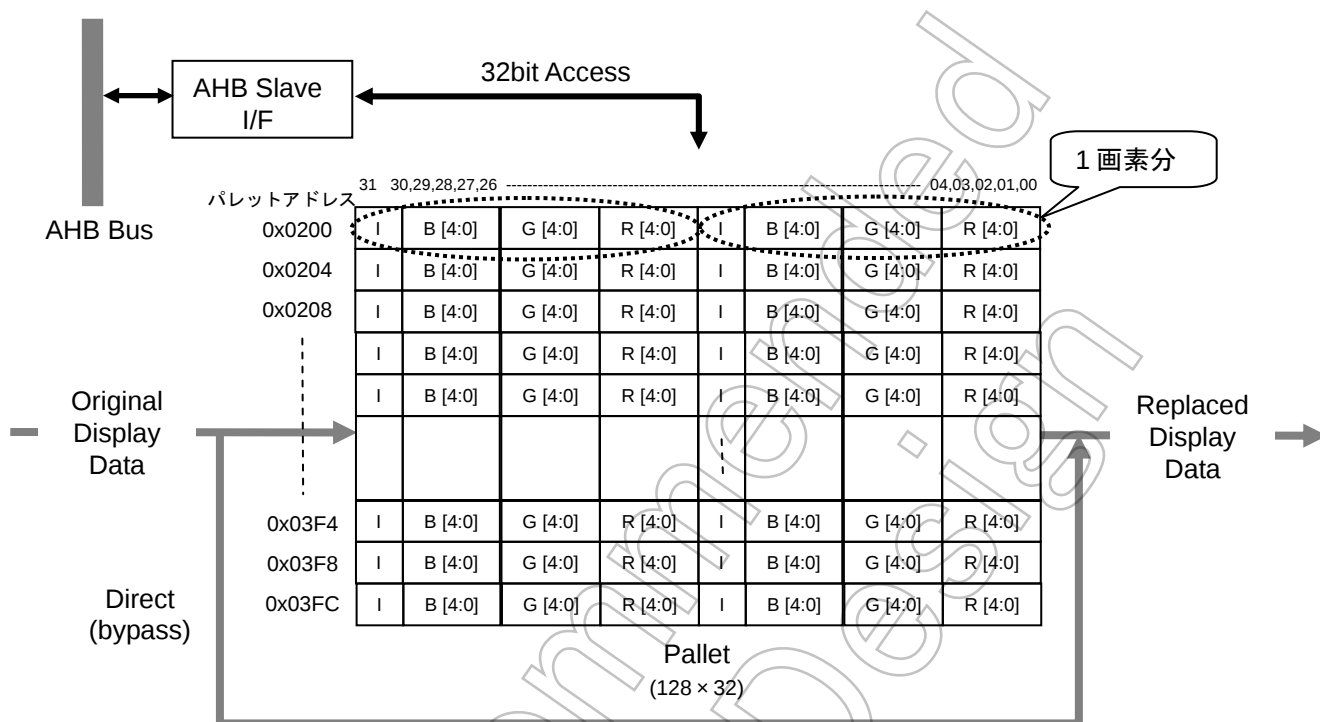


図 3.18.2 パレット RAM

ピクセルデータは、パレット内 16 ビットのデータに置換され、出力されます。

(Original Display Data はパレットアドレスとして扱われ、そのアドレス内のデータが Replaced Display data として出力されます。また、16bpp の場合は、Original Display Data が Replaced Display data として出力されます。)

1word(32bit)のパレット RAM 上のデータは、2 画素分を示しているため、ピクセルデータの最下位ビットは、パレット RAM の上位 16 ビットまたは下位 16 ビットのどちらかを選択するために使用されます。

例) 8bpp で 0x00 のピクセルデータが入力された場合は、0xF420_0200 番地の下位 16bit のデータに置換される。

8bpp で 0xFF のピクセルデータが入力された場合は、0xF420_03FC 番地の上位 16bit のデータに置換される。

R:5 bit、G:5bit、B:5bit で構成されたパレットは、128×32 ビットのデュアルポート RAM で構成されているため、2 画素分のピクセルデータを 1 ワードで書き込むことができます。

LCD Palette

Address = (0xF420_0000) + ((0x0200) ~ (0x03FC))

Bit	Bit Symbol	Type	Reset Value	Description
[31]	I	R/W	0y0	輝度/ 未使用。
[30:26]	B[4:0]	R/W	0y00000	青パレットデータ設定
[25:21]	G[4:0]	R/W	0y00000	緑パレットデータ設定
[20:16]	R[4:0]	–	0y00000	赤パレットデータ
[15]	I	R/W	0y0	輝度/ 未使用。
[14:10]	B[4:0]	R/W	0y00000	青パレットデータ設定
[9:5]	G[4:0]	R/W	0y00000	緑パレットデータ設定
[4:0]	R[4:0]	R/W	0y00000	赤パレットデータ

(グレー表示を含む) モノクロ STN モードでは、赤パレット R[4:1]のみが使用されます (bit0 は使用しない)。

STN カラーモードでは、赤、緑と青のビット [4:1] が使用されます (bit0 は使用しない)。BGR データ形式をサポートするため、この赤と青のピクセルデータは制御レジスタビットを使用してスワップすることができます。

16 / 24 bpp TFT モードではパレットがバイパスされ、ピクセルシリアライザの出力が直接 TFT パネルデータとして使用されます。

RAM パレットは、256 エントリ×16bit が最大です。よって、TFT およびカラーSTN でのパレットは、8bpp データまで対応できます。

注) 本マイコンにおける LCDDA 機能は 64k カラー(16bpp)のみ対応していますので、パレットとの併用はできません。よってスワップ機能も使用できませんのでご注意ください。

3.18.2.4 グレースケール

グレースケールアルゴリズムは、モノクロディスプレイ用の 15 グレー階調をサポートしています。

STN カラーディスプレイの場合は、3 色のカラーコンポーネント(赤、緑、青)が同時にグレースケール階調化され、3375(15 × 15 × 15)色が使用可能になります。

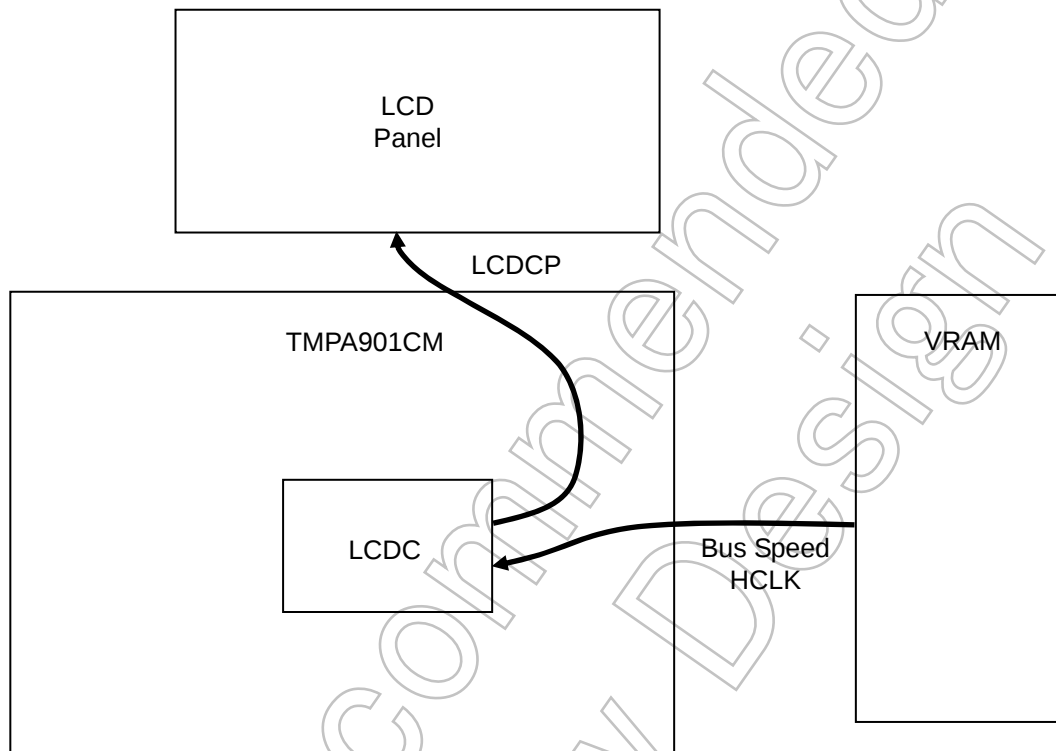
3.18.2.5 上位 / 下位パネルフォーマット

Dual Panel を使用する際の上下画素のデータ分割を行います。

RGB のピクセルデータは、ビット毎、同時にそれぞれの固有のレジスタにシフトインされ、正しいビット位置に構成されます。

3.18.2.6 パネルクロックジェネレータ

内部クロック(HCLK)と、LCDC で使用されるデータ転送クロック(LCLCP 信号)の分周比を設定することが出来ます。LCLCP 信号は、LCD パネルのデータレートに合わせ、 $HCLK/2 \sim HCLK/1025$ の範囲でプログラムすることができます。



3.18.2.7 タイミングコントローラ

タイミングコントローラブロックの主な機能は、水平/垂直タイミングの調整を行います。

3.18.2.8 割り込みの生成

LCDC は、個々にマスク可能な 4 種類の割り込みと、1 種類の結合割り込みを生成します。

3.18.3 レジスタの説明

SFR のリストを以下に示します。

表 3.18-4 レジスター一覧

base アドレス= 0xF420_0000

Register Name	Address (base+)	Description
LCDTiming0	0x000	Horizontal Axis Panel Control Register
LCDTiming1	0x004	Vertical Axis Panel Control Register
LCDTiming2	0x008	Clock and Signal Polarity Control Register
LCDTiming3	0x00C	Line End Control Register
LCDUPBASE	0x010	Upper Panel Frame Base Address Register
LCDLPBASE	0x014	Lower Panel Frame Base Address Register
LCDIMSC	0x018	Interrupt Mask Set/Clear Register
LCDCControl	0x01C	LCDC Control Register
LCDRIS	0x020	Raw Interrupt Status Register
LCDMIS	0x024	Masked Interrupt Status Register
LCDICR	0x028	Interrupt Clear Register
LCDUPCURR	0x02C	Upper Panel Current Address Value Registers
LCDLPCURR	0x030	Lower Panel Current Address Value Registers
LCDPalette	0x200-0x3FC	Color Palette Register

base アドレス= 0xF00B_0000

Register Name	Address (base+)	Description
STN64CR	0x0000	STN 64 階調のコントロールレジスタ

1. LCDTiming0 (Horizontal Axis Panel Control Register)

LCDTiming0 は、以下を制御するレジスタです。

- ・ 水平同期パルス幅(HSW)
- ・ 水平フロントポーチ(HFP)周期
- ・ 水平バックポーチ(HBP)周期
- ・ 1行当たりのピクセル数(PPL)

Address = (0xF420_0000) + (0x0000)

Bit	Bit Symbol	Type	Reset Value	Description
[31:24]	HBP	R/W	0x00	水平バックポーチ幅 (設定値+1) 0x00~0xFF
[23:16]	HFP	R/W	0x00	水平フロントポーチ幅 (設定値+1) 0x00~0xFF
[15:8]	HSW	R/W	0x00	水平同期パルス幅 (設定値+1) 0x00~0xFF
[7:2]	PPL	R/W	0y000000	行当たりのピクセル数 ((設定値+1) × 16) 0y000000 ~ 0y111111
[1:0]	Reserved	–	Undefined	Read as undefined. Write as zero.

(個別説明)

a. <HBP>

水平バックポーチとは、LCLLP の立上り(または立下り)エッジからアクティブデータの開始までの LCLCP サイクル数です。

設定値+1 をカウントするので、(1 ~ 256) × LCLCP サイクル遅延できます。

b. <HFP>

水平フロントポーチとは、アクティブデータの終わってから LCLLP の立下がり(または立上り)エッジまでの LCLCP サイクル数です。

設定値+1 をカウントするので、(1 ~ 256) × LCLCP サイクル遅延できます。

c. <HSW>

水平同期パルス幅とは、LCLLP のアクティブパルス幅を意味します。

設定値+1 をカウントするので、(1 ~ 256) × LCLCP サイクル遅延できます。

d. <PPL>

PPL とは、1行のピクセル数を指定します。

((設定値+1) × 16) で計算されますので、16 ~ 1024 ピクセルを指定できます。

- 水平タイミングの制約

使用する動作モードによって設定上の制約があります。

この最小値は、 $HSW = 2$ 、 $HBP = 2$ です。

STN シングルパネルモードの場合：

- $HSW = 3$
- $HBP = 5$
- $HFP = 5$
- パネルクロック除数(PCD)= $1(HCLK/3)$

STN デュアルパネルモード：

- $HSW = 3$
- $HBP = 5$
- $HFP = 5$
- $PCD = 5(HCLK/7)$

ただし、使用する条件によっては上記行の開始点に十分な時間を設定しておく(例： $HSW = 6$ 、 $HBP = 10$)、 $PCD = 4$ (最小値)の場合でもデータが破損することはありません。

下記の図では動作モードの設定 ($LCDTiming2<IHS>=1$ 、 $LCDTiming2<IPC>=0$ 、 $LCDTiming2<IOE>=0$)を例として、下記図に示す。

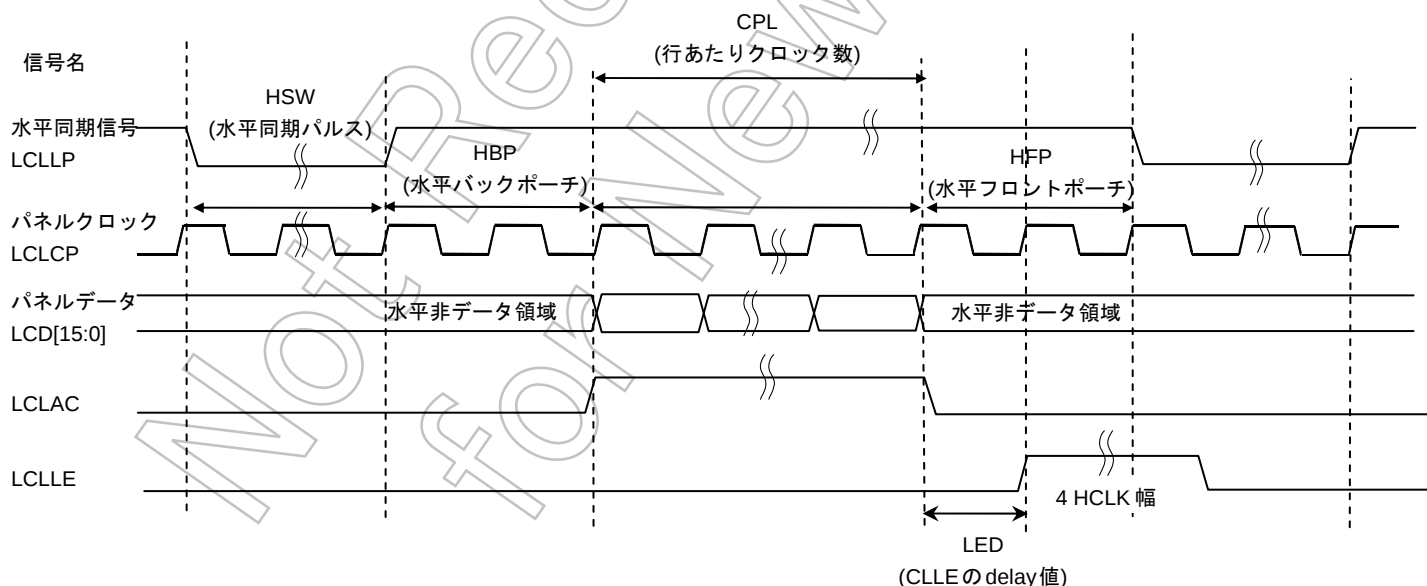


図 3.18.3 水平制御の基本動作

注) CPL は、PPL を $1(TFT)$ 、4 または 8(モノクロ STN)、 $(2 + 2/3)$ (カラーSTN)で除算して、商の値を設定してください。

2. LCDTiming1 (Vertical Axis Panel Control Register)

LCDTiming1 は、以下を制御するレジスタです。

- パネル当たりの行数(LPP)
- 垂直同期パルス幅(VSW)
- 垂直フロントポーチ(VFP)周期
- 垂直バックポーチ(VBP)周期

Address = (0xF420_0000) + (0x0004)

Bit	Bit Symbol	Type	Reset Value	Description
[31:24]	VBP	R/W	0x00	垂直バックポーチの行数設定 0x00 ~0xFF
[23:16]	VFP	R/W	0x00	垂直フロントポーチの行数設定 0x00 ~0xFF
[15:10]	VSW	R/W	0y000000	垂直同期パルスの行数設定(設定値+1) 0y000000 ~0y111111
[9:0]	LPP	R/W	0y0000000000	パネルあたり行数の設定 (設定値+1) 0y0000000000 ~0y1111111111

(個別説明)

a. <VBP>

垂直バックポーチは、垂直同期信号後の、フレーム開始時における非アクティブ行の数を意味します。この 8 ビットの VBP フィールドは、各フレームの開始時に挿入される行クロック数の指定に使用されます。VBP は 0 ~ 255 の行クロックサイクルを生成します。

b. <VFP>

垂直フロントポーチは、垂直同期信号の前のフレーム終了時における非アクティブ行の数を意味します。STN ディスプレイでは 0 をプログラムしないと、鮮明度が低くなります。VFP は 0 ~ 255 の行クロックサイクルを生成します。

c. <VSW>

垂直同期パルス幅は、水平同期行数を意味します。STN ディスプレイ では小さな値(0 をプログラムする等)をセットする必要があります。値が大きくなるほど、鮮明度(コントラスト)が低くなります。

d. <LPP>

LCD パネルのアクティブ行の数を指定します。

設定値+1 をカウントするので、(1 ~ 1024)行が指定できます。

デュアルパネルディスプレイの場合、このレジスタには上位パネルと下位パネルそれぞれの行数をプログラムします。

下記の図では動作モードの設定(LCDTiming2<IVS>=1、LCDTiming2<IHS>=0)を例として、下記図に示します。

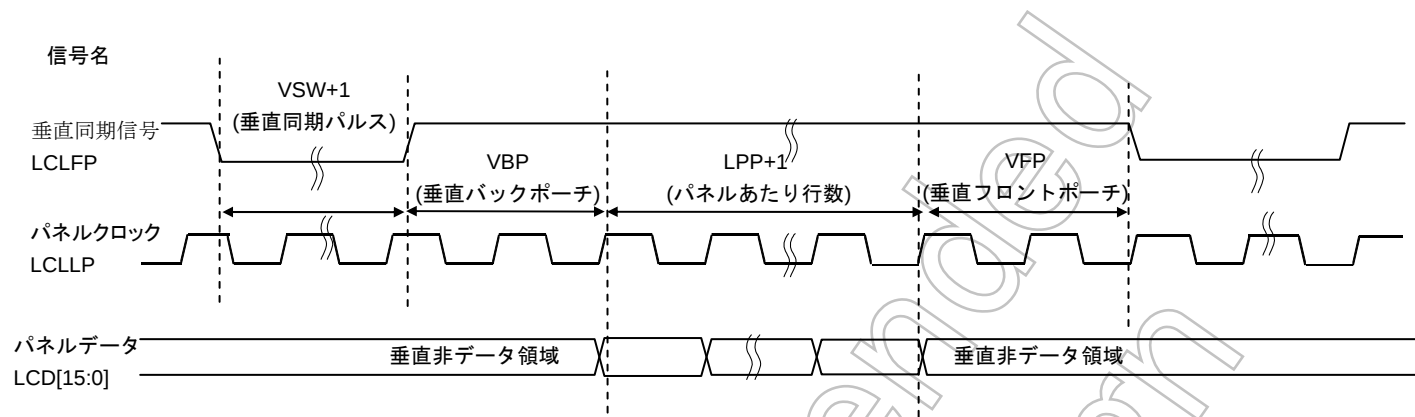


図 3.18.4 垂直制御の基本動作

3. LCDTiming2 (Clock and Signal Polarity Control Register)

LCDTiming2 は、LCDC タイミングを制御する読み出し/書き込みレジスタです。

Address = (0xF420_0000) + (0x0008)

Bit	Bit Symbol	Type	Reset Value	Description
[31:27]	PCD_HI	R/W	0y00000	パネルクロック分周の上位 5 ビット設定値 0y00000 ~ 0y11111
[26]	Reserved	R/W	0y0	Read as undefined. Write as zero.
[25:16]	CPL	R/W	0y0000000000	行当たりクロック数 0y0000000000 ~ 0y1111111111
[15]	—	—	Undefined	Read as undefined. Write as zero.
[14]	IOE	R/W	0y0	データイネーブル信号極性設定 注 1) 0y0: LCLAC 出力が TFT モードで HIGH アクティブ 0y1: LCLAC 出力が TFT モードで LOW アクティブ
[13]	IPC	R/W	0y0	パネルクロック信号エッジ設定 0y0: LCLCP 立ち上がりエッジ 0y1: LCLCP 立ち下がりエッジ
[12]	IHS	R/W	0y0	水平同期信号極性設定 0y0: LCLLP ピンが HIGH アクティブ 0y1: LCLLP ピンが LOW アクティブ
[11]	IVS	R/W	0y0	垂直同期信号極性設定 0y0: LCLFP ピンが HIGH アクティブ 0y1: LCLFP ピンが LOW アクティブ
[10:6]	ACB	R/W	0y00000	バイアス反転周期(設定値 +1) 注 2) 0y00000 ~ 0y11111
[5]	Reserved	—	Undefined	Read as undefined. Write as zero.
[4:0]	PCD_LO	R/W	0y00000	パネルクロック分周の下位 5 ビット設定値 0y00000 ~ 0y11111

注 1) TFT モードのみ有効です。

注 2) STN モードのみ有効です。

(個別説明)

a. <PCD_HI>

LCD パネルクロック周波数を HCLK 周波数から分周して生成します。

PCD_HI(上位 5 ビット)と後述の PCD_LO(下位 5 ビット)と合わせて、10 ビットの分周設定が可能です。LCLCP = HCLK/(PCD+2)。

b. <CPL>

このビットは、各行における LCD パネルへの実際の LCLCP クロック数を指定します。この値は、1 行あたりのピクセル数を 1、4、8、または 8/3 で除算し、そこから 1 を引いた値です。LCD コントローラを正しく機能させるには、PPL に加え、このフィールドを正しくプログラムする必要があります。

パネルタイプ		バス幅	CPL 計算式
TFT		—	$CPL = \left\{ \frac{(1 \text{ 行あたりのピクセル数})}{1} \right\} - 1$
STN	モノクロ	4	$CPL = \left\{ \frac{(1 \text{ 行あたりのピクセル数})}{4} \right\} - 1$
		8	$CPL = \left\{ \frac{(1 \text{ 行あたりのピクセル数})}{8} \right\} - 1$
	カラー	8	$CPL = \left\{ \frac{(1 \text{ 行あたりのピクセル数})}{\frac{8}{3}} \right\} - 1$ 注) 小数点以下切り上げ

c. <IOE>

データイネーブル信号の極性を設定します。

有効表示データが使用可能であるかを LCD パネルに通知するイネーブル信号として LCLAC 端子から出力されます。TFT モードのみ使用可能です。

d. <IPC>

パネルクロック信号のエッジを設定します。

e. <IHS>

水平同期信号の極性を設定します。

f. <IVS>

垂直同期信号の極性を設定します。

g. <ACB>

バイアス反転の周期を設定します。STN ディスプレイでは、DC 電荷の蓄積による液晶の劣化を防ぐため、バイアスの極性を定期的に反転させる必要があります。バイアス反転の周期は行数単位で設定します。設定値+1 の周期でトグルするため、1~32 行単位での反転設定が可能です。STN ディスプレイのみ使用可能です。

h. <PCD_LO>

パネルクロックの分周設定値(10 ビット)の下位 5 ビット。

注) STN モードにおけるパネルクロックディバイダに使用可能な最小値には制約があります。

シングルパネルカラーモード：

$PCD = 1 (LCLCP = HCLK/3)$

デュアルパネルカラーモード：

$PCD = 4 (LCLCP = HCLK/6)$

シングルパネルモノクロ 4 ビットインタフェースモード：

$PCD = 2 (LCLCP = HCLK/4)$

デュアルパネルモノクロ 4 ビットインタフェースモード：

$PCD = 6 (LCLCP = HCLK/8)$

シングルパネルモノクロ 8 ビットインタフェースモード：

$PCD = 6 (LCLCP = HCLK/8)$

デュアルパネルモノクロ 8 ビットインタフェースモード：

$PCD = 14 (LCLCP = HCLK/16)$

4. LCDTiming3 (Line End Control Register)

Address = (0xF420_0000) + (0x000C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:17]	–	–	Undefined	Read as undefined. Write as zero.
[16]	LEE	R/W	0y0	行終端シグナル CLLE のイネーブル: 0y0: Disable (LOW 固定) 0y1: Enable
[15:7]	–	–	Undefined	Read as undefined. Write as zero.
[6:0]	LED	R/W	0y0000000	CLLE 出力の Delay 値 (設定値 +1) 0y0000000 ~ 0y1111111

(個別説明)

a. <LEE>

このシグナルがイネーブルされると、各行の終了後、CLLE は 4 HCLK 周期の正パルスを出します。

行終端シグナルがディセーブルされている場合、このシグナルは常に LOW でホールドされます。

b. <LED>

CLLE 出力の Delay 値を設定します。

5. LCDUPBASE (Upper Panel Frame Base Address Register)

カラーLCD DMA ベースアドレスレジスタです。

$$\text{Address} = (0xF420_0000) + (0x0010)$$

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	LCDUPBASE	R/W	0x00000000	カラーLCD DMA ベースアドレスの設定レジスタ 0x00000000 ~ 0x3FFFFFFF
[1:0]	–	–	Undefined	Read as undefined. Write as zero.

6. LCDLPBASE (Lower Panel Frame Base Address Register)

$$\text{Address} = (0xF420_0000) + (0x0014)$$

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	LCDLPBASE	R/W	0x00000000	カラーLCD DMA ベースアドレスの設定レジスタ 0x00000000 ~ 0x3FFFFFFF
[1:0]	–	–	Undefined	Read as undefined. Write as zero.

LCDUPBASE と LCDLPBASE は、表示 RAM の先頭番地を設定します。

LCDUPBASE は以下に使用します。

- TFT ディスプレイ
- シングルパネル STN ディスプレイ
- デュアルパネル STN ディスプレイの上位パネル

LCDLPBASE は、デュアルパネル STN ディスプレイの下位パネルに使用します。

プログラマは、LCDC をイネーブルする前に、LCDUPBASE(およびデュアルパネルの LCDLPBASE)を設定してください。

各アドレスの設定は[31:0]の 32 ビットフルアドレスを設定します。ただし、下位 2 ビットの設定は無効となり、word(4バイト)単位の設定となります。

7. LCDIMSC (Interrupt Mask Set/Clear Register)

Address = (0xF420_0000) + (0x0018)

Bit	Bit Symbol	Type	Reset Value	Description
[31:5]	–	–	Undefined	Read as undefined. Write as zero.
[4]	MBERRINTRENB	R/W	0y0	AHB マスタエラー割り込みイネーブル 0y0: Disable 0y1: Enable
[3]	VCOMPINTRENB	R/W	0y0	垂直同期割り込みイネーブル 0y0: Disable 0y1: Enable
[2]	LNBUINTRENB	R/W	0y0	ベースアドレス更新割り込みイネーブル 0y0: Disable 0y1: Enable
[1]	FUFINTRENB	R/W	0y0	FIFO アンダーフロー割り込みイネーブル 0y0: Disable 0y1: Enable
[0]	–	–	Undefined	Read as undefined. Write as zero.

LCDIMSC は、割り込みイネーブルレジスタです。このレジスタ内のビットをセットすると、対応する原割り込み LCDRIS ビットの値が LCDMIS レジスタに渡されます。

8. LCDControl (LCDC Control Register)

Address = (0xF420_0000) + (0x001C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:17]	–	–	Undefined	Read as undefined. Write as zero.
[16]	WATERMARK	R/W	0y0	LCD DMA FIFO ウォーターマークレベル 0y0: 2 つの FIFO のどちらかに 4 word 以上の空きが発生すると、DMA を要求します。 0y1: 2 つの FIFO のどちらかに 8word 以上の空きが発生すると、DMA を要求します。
[15:14]	–	–	Undefined	Read as undefined. Write as zero.
[13:12]	LcdVComp	R/W	0y00	割り込み発生タイミング 0y00: 垂直同期の開始時 0y01: バックボーチの開始時 0y10: 表示データの開始時 0y11: フロントボーチの開始時
[11]	Reserved	R/W	0y0	常に"1"をライトしてください。
[10]	Reserved	R/W	0y0	常に"0"をライトしてください。
[9]	Reserved	R/W	0y0	常に"0"をライトしてください。
[8]	BGR	R/W	0y0	BGR 形式選択の RGB 0y0: RGB 通常出力 0y1: BGR 赤/青スワップ
[7]	LcdDual	R/W	0y0	STN パネル選択 0y0: シングルパネル LCD 0y1: デュアルパネル LCD
[6]	LcdMono8	R/W	0y0	モノクロ STN LCD パラレルビット選択 0y0: モノクロ LCD が 4 ビットインタフェース 0y1: モノクロ LCD が 8 ビットインタフェース
[5]	LcdTFT	R/W	0y0	LCD 使用パネル選択 0y0: STN パネル 0y1: TFT パネル
[4]	LcdBW	R/W	0y0	STN LCD がモノクロとカラー選択 0y0: カラー 0y1: モノクロ
[3:1]	LcdBpp	R/W	0y000	ピクセル当たりの LCD ビット数 : 0y000 = 1 bpp 0y001 = 2 bpp 0y010 = 4 bpp 0y011 = 8 bpp 0y100 = 16 bpp 0y101 = Reserved 0y110 = Reserved 0y111 = Reserved
[0]	LcdEn	R/W	0y0	LCD コントローラインープル 0y0: 停止 0y1: 動作

(個別説明)

a. <WATERMARK>

LCD DMA FIFO ウォーターマークレベル

0y0: 2つのFIFOのどちらかに4word以上の空きが発生すると、DMAを要求します。

0y1: 2つのFIFOのどちらかに8word以上の空きが発生すると、DMAを要求します。

b. <LcdVComp>

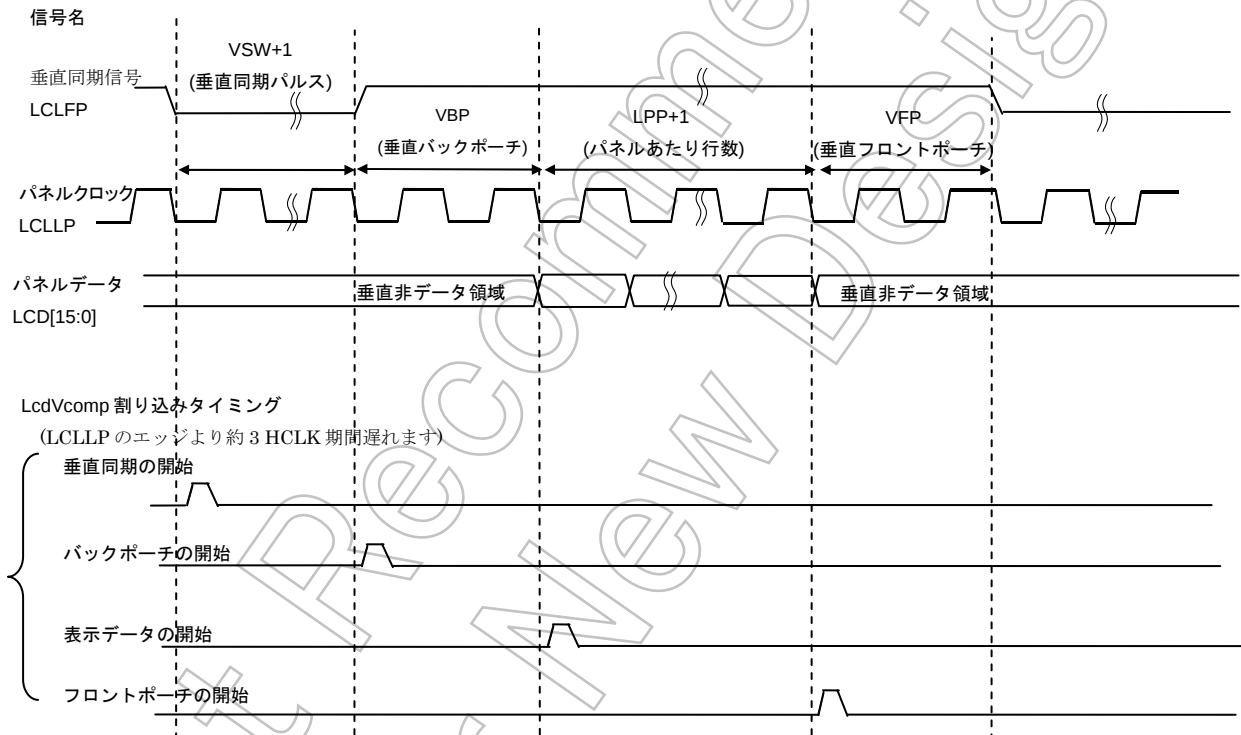
0y00: 垂直同期の開始時

0y01: バックポーチの開始時

0y10: 表示データの開始時

0y11: フロントポーチの開始時

割り込み発生タイミング、下記のタイミングチャートに示します。



c. <BGR>

BGR 形式選択の RGB

0y0: RGB 通常出力

0y1: BGR 赤/青スワップ

スワップ機能とは、以下表に示すように青色と赤色のデータを入れ替える機能です。

対象ビット	<BGR>=0	<BGR>=1
[15]	輝度ビット	輝度ビット
[14:10]	B[4:0]	R[4:0]
[9:5]	G[4:0]	G[4:0]
[4:0]	R[4:0]	B[4:0]

ただし、本マイコンでは LCDDA 機能を有しており、本機能を利用する場合に必要なデータの並びは以下で定義されています。

対象ビット	データ内容
[15:11]	B[4:0]
[10:5]	G[5:0]
[4:0]	R[4:0]

その為、LCDDA 機能をご利用の場合には、スワップ機能は使用できませんのでご注意ください。

d. <LcdDual>

STN パネル選択

0y0: シングルパネル LCD

0y1: デュアルパネル LCD

e. <LcdMono8>

モノクロ LCD が 8 ビットインタフェースを使用することを示します。このビットは、モノクロ STN LCD が 4 ビットまたは 8 ビットパラレルインタフェースのどちらを使用するかを制御します。他のモードでは 0 を設定してください。

f. <Lcd TFT>

0y0 = LCD が STN ディスプレイであり、グレースケールを使用することを示します。

0y1 = LCD が TFT であり、グレースケールを使用しないことを示します。

g. <LcdBW>

STN LCD がモノクロ(白黒)であることを示します。

0y0 = STN LCD がカラーであることを示します。

0y1 = STN LCD がモノクロであることを示します。

TFT モードの場合、このビットは無効です。

h. <Lcd Bpp>

ピクセル当たりの LCD ビット数を設定します。

0y000 = 1 bpp

0y001 = 2 bpp

0y010 = 4 bpp

0y011 = 8 bpp

0y100 = 16 bpp

0y101 = Reserved

0y110 = Reserved

0y111 = Reserved

i. <LcdEn>

LCD コントローラの停止/動作を制御します。

0y0: 停止

LCD シグナル LCLLP、LCLCP、LCLFP、LCLAC、LCLLE がディセーブルされます (LOW 固定)

0y1: 動作

LCD シグナル LCLLP、LCLCP、LCLFP、LCLAC、LCLLE がイネーブルされます (アクティブ)。

注 1) LCDC の各レジスタ設定が完了後に<LcdEn>を"0y1"に設定してください。

注 2) 停止状態にすると LCD シグナル LCLLP、LCLCP、LCLFP、LCLAC、LCLLE は常に LOW に固定されます。

負論理で設定された信号には注意が必要です。

9. LCDRIS (Raw Interrupt Status Register)

Address = (0xF420_0000) + (0x0020)

Bit	Bit Symbol	Type	Reset Value	Description
[31:5]	–	–	Undefined	Read as undefined. Write as zero.
[4]	MBERROR	RO	0y0	AMBA AHB マスタバスエラー割り込み要求 0y0: なし 0y1: あり
[3]	Vcomp	RO	0y0	垂直同期割り込み要求 0y0: なし 0y1: あり
[2]	LNBU	RO	0y0	LCD 次アドレスベース更新の割り込み要求 0y0: なし 0y1: あり
[1]	FUF	RO	0y0	FIFO アンダーフロー割り込み要求 0y0: なし 0y1: あり
[0]	–	–	Undefined	Read as undefined.

(個別説明)

a. <MBERROR>

AMBA AHB マスタバスエラーステータス。AMBA AHB マスタが、スレーブからのバスエラー応答を検出するとセットされます。

b. <Vcomp>

垂直同期。LCDControl[13:12] レジスタから選択された 4 つの垂直領域のいずれか 1 つタイミングに到達すると、セットされます。

c. <LNBU>

LCD 次アドレスベース更新。モードに依存し、カレントベースアドレスレジスタが次のアドレスレジスタによって正しく更新された場合にセットされます。

d. <FUF>

FIFO アンダーフロー。上位または下位どちらかの DMA FIFO が、アンダーフロー条件を発生させる空のときに読み出しアクセスされるとセットされます。

10. LCDMIS (Masked Interrupt Status Register)

LCDMIS は読み出し専用レジスタです。このレジスタは、LCDRIS レジスタと LCDIMSC(Enable)レジスタのビットごとの論理 AND です。全ての割り込みの論理 OR がシステム割り込みコントローラに与えられます。

Address = (0xF420_0000) + (0x0024)

Bit	Bit Symbol	Type	Reset Value	Description
[31:5]	–	–	Undefined	Read as undefined. Write as zero.
[4]	MBERRORINTR	RO	0y0	AMBA AHB マスタバスエラーステータス ビット 0y0: clear 0y1: 割り込み要求あり
[3]	VCOMPINTR	RO	0y0	垂直同期割り込みステータス ビット 0y0: clear 0y1: 割り込み要求あり
[2]	LNBUINTR	RO	0y0	LCD 次アドレスベース更新ステータス ビット 0y0: clear 0y1: 割り込み要求
[1]	FUFINTR	RO	0y0	FIFO アンダーフローステータス ビット 0y0: clear 0y1: 割り込み要求あり
[0]	–	–	Undefined	Read as undefined.

11. LCDICR (Interrupt Clear Register)

Address = (0xF420_0000) + (0x0028)

Bit	Bit Symbol	Type	Reset Value	Description
[31:5]	–	–	Undefined	Read as undefined. Write as zero.
[4]	Clear MBERROR	WO	0y0	AMBA AHB マスタバスエラー割り込み要求フラグのクリア 0y0: 変化しない 0y1: クリア
[3]	Clear Vcomp	WO	0y0	垂直同期割り込み要求フラグのクリア 0y0: 変化しない 0y1: クリア
[2]	Clear LNBU	WO	0y0	LCD 次アドレスベース更新割り込み要求フラグのクリア 0y0: 変化しない 0y1: クリア
[1]	Clear FUF	WO	0y0	FIFO アンダーフロー割り込み要求フラグのクリア 0y0: 変化しない 0y1: クリア
[0]	–	–	Undefined	Read as undefined. Write as zero.

12. LCDUPCURR (Upper Panel Current Address Value Registers)

Address = (0xF420_0000) + (0x002C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	LCDUPCURR	RO	0x00000000	上位パネルデータ DMA アドレスの近似値

13. LCDLPCURR (Lower Panel Current Address Value Registers)

Address = (0xF420_0000) + (0x0030)

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	LCDLPCURR	RO	0x00000000	下位パネルデータ DMA アドレスの近似値

LCDUPCURR レジスタと LCDLPCURR レジスタは、読み出し時に上位および下位パネルデータ DMA アドレスの近似値を保持しています。これらのレジスタは常に変化し得るため、粗いモニタしかできないので、使用時、注意してください。

14. LCDPalette (Color Palette Register)

1word(32bit)のパレット RAM 上のデータは、2 画素分を示している為、ピクセルデータの最下位ビットは、パレット RAM の上位 16 ビットまたは下位 16 ビットのどちらかを選択するために使用されます。

R:5 bit、G:5bit、B:5bit、及び輝度ビットで構成されたパレットは、128 × 32 ビットのデュアルポート RAM で構成されているため、パレットへの 2 画素分のエントリを 1 ワードで書き込むことができます。

TFT モードでは、すべてのパレットデータを利用しますが、モノクロ STN モードでは、赤パレット R[4:1]のみが使用され(bit0 は使用しない)、STN カラーモードでは、赤、緑と青のビット [4:1] が使用されます(bit0 は使用しない)。

BGR データ形式をサポートするため、この赤と青のピクセルデータは制御レジスタビットを使用してスワップすることができます。

16 bpp TFT モードではパレットがバイパスされ、ピクセルシリアルライザの出力が直接 TFT パネルデータとして使用されます。

RAM パレットは、256 エントリ×16bit が最大です。よって、TFT およびカラーSTN でのパレットは、8bpp データまで対応できます。

$$\text{Address} = (0xF420_0000) + ((0x0200) \sim (0x03FC))$$

Bit	Bit Symbol	Type	Reset Value	Description
[31]	I	R/W	0y0	輝度/ 未使用。
[30:26]	B[4:0]	R/W	0y00000	青パレットデータ設定
[25:21]	G[4:0]	R/W	0y00000	緑パレットデータ設定
[20:16]	R[4:0]	R/W	0y00000	赤パレットデータ
[15]	I	R/W	0y0	輝度/ 未使用。
[14:10]	B[4:0]	R/W	0y00000	青パレットデータ設定
[9:5]	G[4:0]	R/W	0y00000	緑パレットデータ設定
[4:0]	R[4:0]	R/W	0y00000	赤パレットデータ

(個別説明)

a. <I>

輝度ビット。6:6:6 TFT ディスプレイへの R、G ならびに B 入力の LSB として使用し、各色に 2 通りの輝度を設定することができます。色数を 2 倍にして、64K になります。

b.

青パレットデータ

c. <G>

緑パレットデータ

d. <R>

STN ディスプレイの場合、4 つの MSB(ビット 4:1)だけが使用されます。モノクロディスプレイの場合は、赤パレットデータだけが使用されます。全てのパレットレジスタに同じビットを配置しています。

3.18.3.1 LCD パネルシグナルの多重化

LCLLP、LCLAC、LCLFP、LCLCP、LCLLE は共通シグナルですが、LCLD[15:0] バスには以下に対応する 8 つの動作モードがあります。

- TFT 16 ビットインタフェース
- カラーSTN シングルパネル
- カラーSTN デュアルパネル
- 4 ビットモノクロ STN シングルパネル
- 4 ビットモノクロ STN デュアルパネル
- 8 ビットモノクロ STN シングルパネル
- 8 ビットモノクロ STN デュアルパネル

注)

CUSTN = カラーSTN デュアルの上位パネルのデータ信号/カラーSTN シングルパネルのデータ信号

CLSTN = カラーSTN デュアルの下位パネルのデータ信号

MUSTN = モノクロ STN デュアルの上位パネルのデータ信号/モノクロ STN シングルパネルのデータ信号

MLSTN = モノクロ STN デュアルの下位パネルのデータ信号

表 3.18-5 LCD TFT panel signal multiplexing [TFT 16bit Interface]

External pin	Color bit allocation	Pallet & RGB-BGR conversion	VRAM bit allocation			
			32bit bus RAM		16bit bus RAM	
			Address	Data bit	Address	Data bit
CLD[0]	BLUE[4]	Color Pallet Bypass	n	D31	n+2	D15
CLD[17]	BLUE[3]			D30		D14
CLD[16]	BLUE[2]			D29		D13
CLD[15]	BLUE[1]			D28		D12
CLD[14]	BLUE[0]			D27		D11
CLD[13]	GREEN[5]			D26		D10
CLD[11]	GREEN[4]			D25		D9
CLD[10]	GREEN[3]			D24		D8
CLD[9]	GREEN[2]			D23		D7
CLD[8]	GREEN[1]			D22		D6
CLD[7]	GREEN[0]			D21		D5
CLD[5]	RED[4]			D20		D4
CLD[4]	RED[3]			D19		D3
CLD[3]	RED[2]			D18		D2
CLD[2]	RED[1]			D17		D1
CLD[1]	RED[0]			D16		D0
CLD[0]	BLUE[4]	RGB-GBR Not Support	n	D15	n	D15
CLD[17]	BLUE[3]			D14		D14
CLD[16]	BLUE[2]			D13		D13
CLD[15]	BLUE[1]			D12		D12
CLD[14]	BLUE[0]			D11		D11
CLD[13]	GREEN[5]			D10		D10
CLD[11]	GREEN[4]			D9		D9
CLD[10]	GREEN[3]			D8		D8
CLD[9]	GREEN[2]			D7		D7
CLD[8]	GREEN[1]			D6		D6
CLD[7]	GREEN[0]			D5		D5
CLD[5]	RED[4]			D4		D4
CLD[4]	RED[3]			D3		D3
CLD[3]	RED[2]			D2		D2
CLD[2]	RED[1]			D1		D1
CLD[1]	RED[0]			D0		D0

注) 16bitTFT を使用する場合は、Intensity ビットを使用しません。また、RGB、BGR のスワップ機能も使用できません。

表 3.18-6 LCD STN panel signal multiplexing

External pin	Color STN single panel	Color STN dual panel	4-bit mono STN single panel	4-bit mono STN dual panel	8-bit mono STN single panel	8-bit mono STN dual panel
CLD[15]	Reserved	CLSTN[7]	Reserved	Reserved	Reserved	MLSTN[7]
CLD[14]	Reserved	CLSTN[6]	Reserved	Reserved	Reserved	MLSTN[6]
CLD[13]	Reserved	CLSTN[5]	Reserved	Reserved	Reserved	MLSTN[5]
CLD[12]	Reserved	CLSTN[4]	Reserved	Reserved	Reserved	MLSTN[4]
CLD[11]	Reserved	CLSTN[3]	Reserved	MLSTN[0]	Reserved	MLSTN[3]
CLD[10]	Reserved	CLSTN[2]	Reserved	MLSTN[1]	Reserved	MLSTN[2]
CLD[9]	Reserved	CLSTN[1]	Reserved	MLSTN[2]	Reserved	MLSTN[1]
CLD[8]	Reserved	CLSTN[0]	Reserved	MLSTN[3]	Reserved	MLSTN[0]
CLD[7]	CUSTN[7]	CUSTN[7]	Reserved	Reserved	MUSTN[7]	MUSTN[7]
CLD[6]	CUSTN[6]	CUSTN[6]	Reserved	Reserved	MUSTN[6]	MUSTN[6]
CLD[5]	CUSTN[5]	CUSTN[5]	Reserved	Reserved	MUSTN[5]	MUSTN[5]
CLD[4]	CUSTN[4]	CUSTN[4]	Reserved	Reserved	MUSTN[4]	MUSTN[4]
CLD[3]	CUSTN[3]	CUSTN[3]	MUSTN[3]	MUSTN[3]	MUSTN[3]	MUSTN[3]
CLD[2]	CUSTN[2]	CUSTN[2]	MUSTN[2]	MUSTN[2]	MUSTN[2]	MUSTN[2]
CLD[1]	CUSTN[1]	CUSTN[1]	MUSTN[1]	MUSTN[1]	MUSTN[1]	MUSTN[1]
CLD[0]	CUSTN[0]	CUSTN[0]	MUSTN[0]	MUSTN[0]	MUSTN[0]	MUSTN[0]

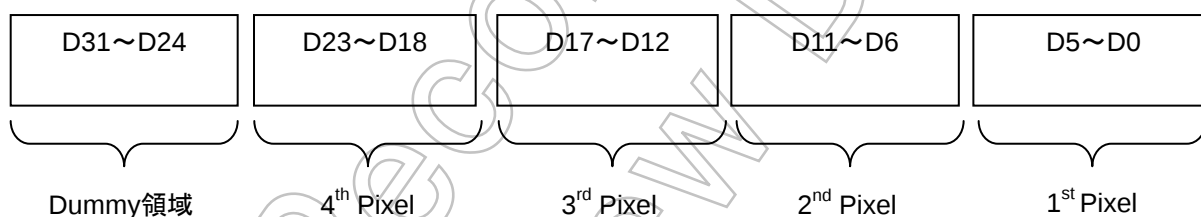
3.18.4 LCDコントローラオプション機能 (LCDCOP)

本 LSI には、LCD コントローラを内蔵しておりますが、オプション機能として 64 階調の LCD 表示機能を追加しております。

表 3.18-7 オプション機能表

LCD パネルのタイプ		STN (デュアル/ シングル)
同時発色可能な表示色		6 4 階調
対応水平ピクセル数		Max : 640
対応垂直行数		Max : 480
転送先データバス幅 (LCD ドライバ)		4/8 bit
入力データ		LCDC の TFT16bit モードの出力データ
データ形式		リトルエンディアンのサポート
接 続 端 子	LD[7:0] / LD[3:0]端子	LCD ドライバ専用データバス
	LCLAC 端子	AC バイアス信号(フレーム信号)
	LCLLP 端子	水平同期パルス
	LCLFP 端子	垂直同期パルス
	LCLCP 端子	LCDD のデータラッチ用クロック(パネルクロック)
	LCLLE 端子	基本的に使用しません

STN64 対応時の VRAM は以下 Format にて対応します。また、外部 Data Bus は 4bit/8bit の LD バスをサポートします。



3.18.4.1 ブロック図

LCDC、LCDCOP のブロック図を示します。

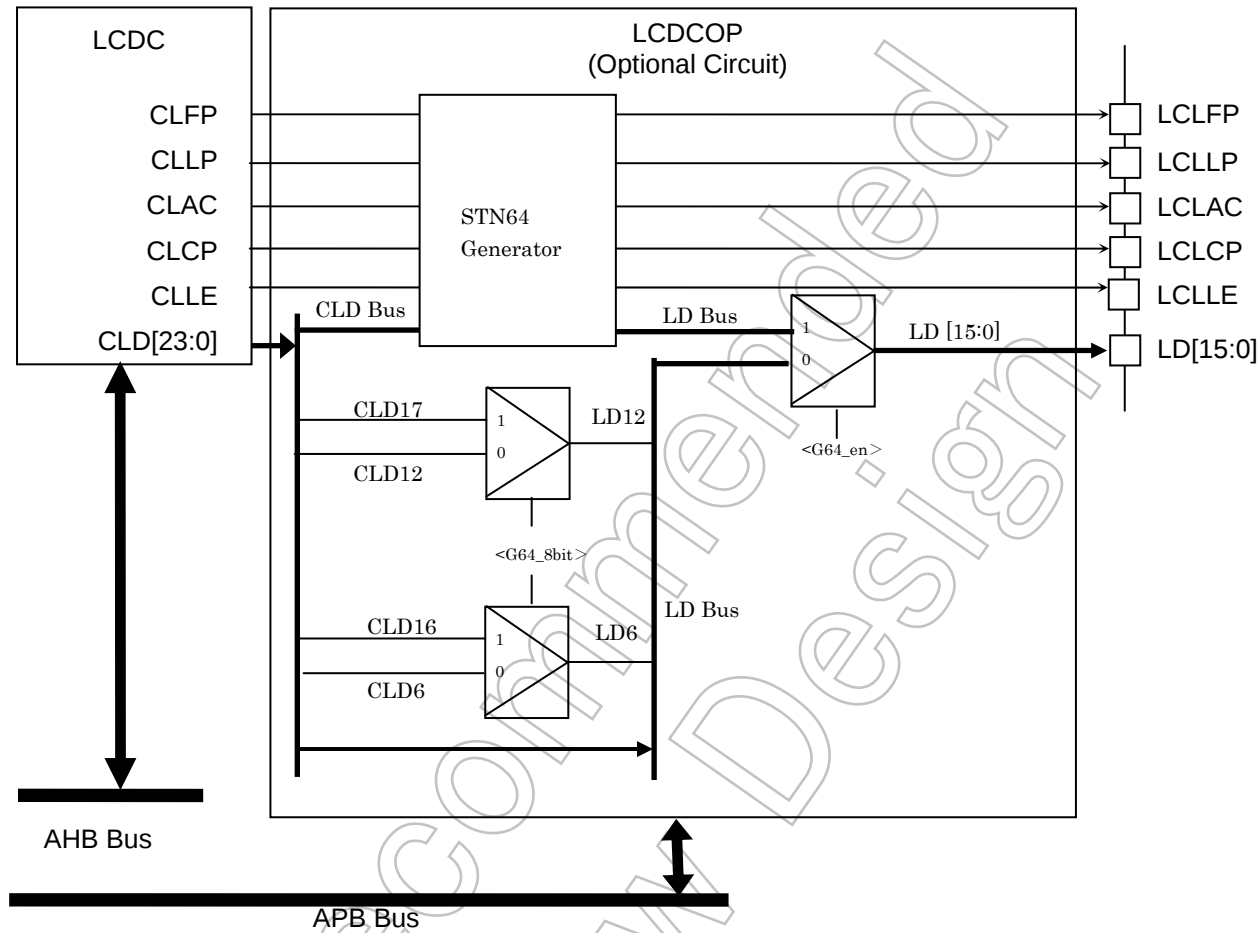


図 3.18.5 LCDCOP ブロック図

3.18.4.2 レジスタの説明

レジスタのリストを以下に示します。

base アドレス= 0xF00B_0000

Register Name	Address (base+)	Description
STN64CR	0x0000	STN 64 階調のコントロールレジスタ

15. STN64CR (LCDC Option Control Register)

Address = (0xF00B_0000) + (0x0000)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7]	NoSpikeMode	R/W	0y0	LCDC 本体の CLCP のノイズ除去回路 0y0 : 無効 0y1 : 有効
[6]	Reserved	R/W	0y0	Read as undefined. Write as zero.
[5]	CLFP_Inv	R/W	0y0	Invert vertical synchronization (=LCDC 本体の VIS) 0y0 : LCLFP pin HIGH active 0y1 : LCLFP pin LOW active
[4]	CLLP_Inv	R/W	0y0	Invert horizontal synchronization (=LCDC 本体の HIS) 0y0 : LCLLP pin HIGH active 0y1 : LCLLP pin LOW active
[3]	CLAC_Inv	R/W	0y0	Invert output enable (=LCDC 本体の IOE) 0y0 : LCLAC output HIGH active in TFT mode 0y1 : LCLAC output LOW active in TFT mode
[2]	LCP_Inv	R/W	0y0	Invert panel clock (=LCDC 本体の IPC) 0y0 : LCLCP rising edge 0y1 : LCLCP falling edge
[1]	G64_8bit	R/W	0y0	下記の表 3.18-8 STNおよびTFT16 bitの信号設定を参照してください。
[0]	G64_en	R/W	0y0	

注： STN64 階調を使用時、LCLFP,LCLLP,LCLAC,LCLCP は必ず LCDC 本体の設定と同様に設定してください。

3.18-8 STN および TFT16 bit の信号設定

モード	レジスタ設定		備考
	<G64_en>	<G64_8bit>	
STN64 階調	0y1 : STN 64 階調回路を使用	0y0 : 外部 4bit LD Bus 0y1 : 外部 8bit LD Bus	
TFT16bit	0y0 : STN64 階調回路を使用 しません。	0y1 : 外部 LD12=CLD17 外部 LD6=CLD16	PU,PV の LD[15:0]機能を使用時、 内部 CLD17, CLD16 を出力するため、 <G64_8bit>=0y1 の設定が必要です。
その他	0y0 : STN64 階調回路を使用 しません。	0y0 : 外部 LD12=CLD12 外部 LD6=CLD6	

注 1) LD Bus の切り替えについて、図 3.18.6 LCDCOP ブロック図を参照してください。

注 2) 外部 16bit TFT の信号について、表 3.18-5 LCD TFT panel signal multiplexing [TFT 16bit Interface]を参照してください。

3.19 LCDデータプロセスアクセラレータ(LCDDA)

本マイコンでは、表示に関する補助機能として、LCD Data Process Accelerator 機能 (LCDDA) を内蔵しています。

フィルタ(Bi-Cubic: バイキュービック法) 処理を含む、表示データ拡大・縮小するスケーラ機能、表示データを回転・ミラー反転する画像回転機能、2枚の画面の重ね合わせ機能(α Blend、Picture in Picture、文字重ね)をサポートしています。

下記にその機能の一覧を示します。

表 3.19.1 LCDDA の機能

機能	概要
スケーラ機能	拡大: $256/n$: ($n=1\sim 255$)倍の拡大が可能 水平方向/垂直方向の独立した拡大が可能 拡大画像は Bi-Cubic 法による Filtering が可能
	縮小: $256/(n\times m)$: ($n=1\sim 255$, $m=1\sim 16$)倍までの縮小が可能 水平方向/垂直方向の独立した縮小が可能 縮小画像は Bi-Cubic 法による Filtering が可能
画像回転機能	90度/180度/270度/水平ミラー反転/垂直ミラー反転可能
画像 Blend 機能	2枚の画像の重ね合わせ(Picture in Picture)機能
	2枚の画像の濃度を調整した重ね合わせ(α -Blend)可能
	2値(モノクロ)で表現された Font Data の Font Draw 機能

本回路は、LCD コントローラとは完全に別の回路として動作しますので、すべて Copy Back(書き戻し)の方式となります。画像データを加工し、LCDC の表示 RAM にそのデータをライトし、LCDC はその加工済みのデータを表示します。

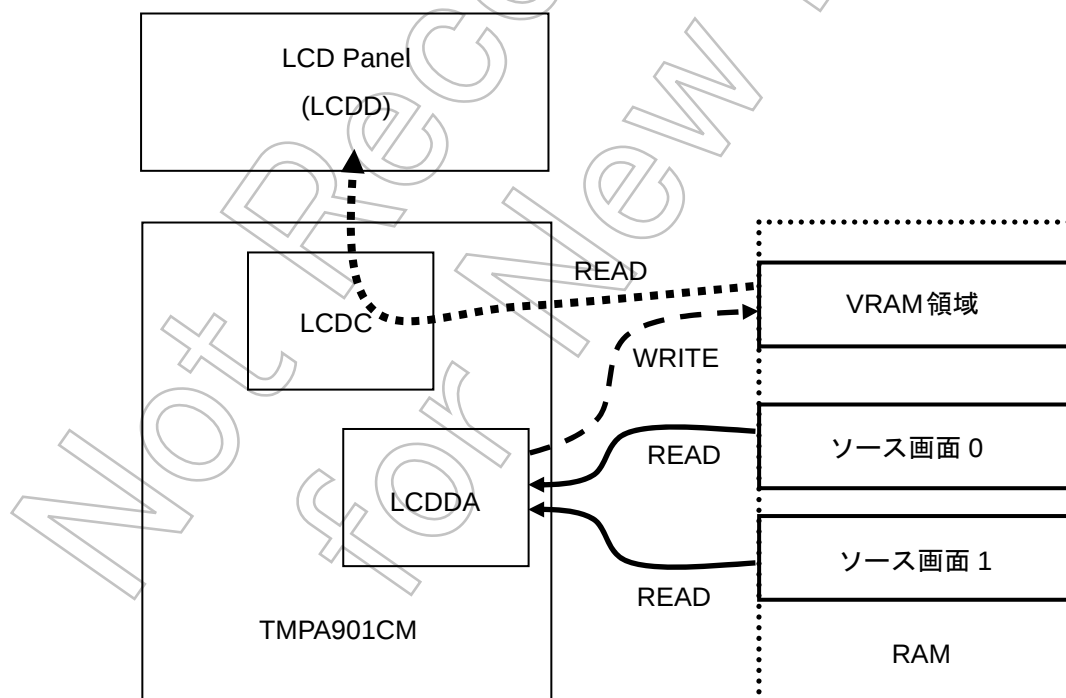


図 3.19.1 LCDDA 動作イメージ(Blend 機能の例)

3.19.1 ブロック図

LCDDA のブロック図を示します。

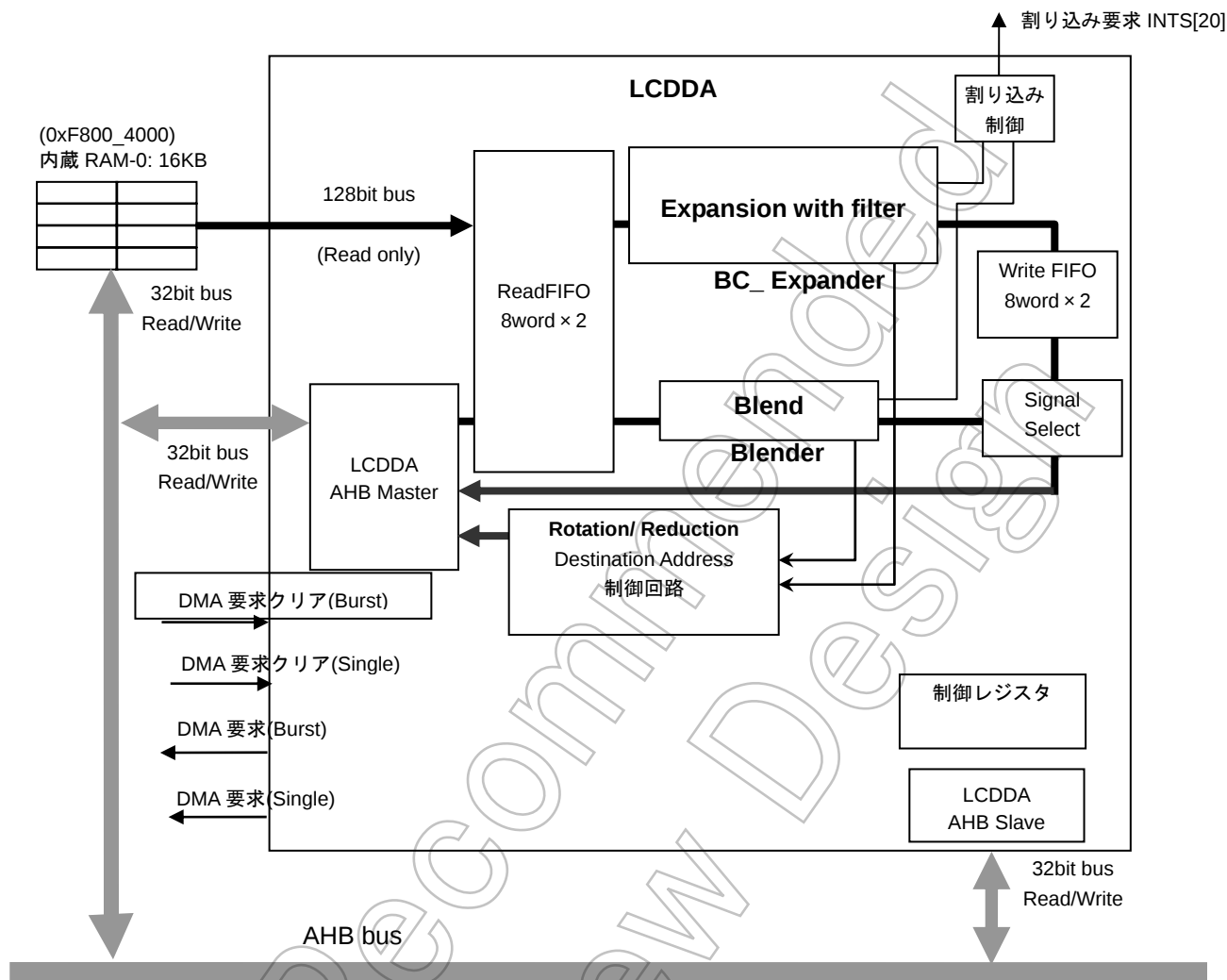


図 3.19.2 LCDDA ブロック図

LCDDAは、主に8つのブロックに分かれています。

- Bi-Cubic 方式を使った拡大・縮小を行う「BC_Expander」
- Blend 処理を行う「Blender」
- ソースデータを蓄積する「リード FIFO バッファ」
- ディストネーションデータを蓄積する「ライト FIFO バッファ」
- 回転・単純縮小処理を行う「転送アドレス制御回路」
- AHB バスから、制御レジスタへのアクセスを制御する「AHB スレーブブロック」
- AHB バスとのデータアクセスを制御する「AHB マスタブロック」
- 処理の終了やエラーを監視して割り込みを発生する「割り込み制御ブロック」

3.19.2 動作説明

ここでは、LCDDA の持つ機能を説明します。

3.19.2.1 スケーラ機能

表 3.19.2 スケーラ機能

機能	機能詳細	説明・規格
拡大機能	拡大率・補間データ数	256/n : (n=1~255)倍の拡大が可能(0 設定は禁止) 水平方向／垂直方向の独立した拡大が可能
	対応データフォーマット	Digital RGB 注) YUV フォーマットのデータはサポートしていません。
	対応画面色数	64K カラー(16bpp) 注)モノクロ、モノクロ諧調およびその他の色数のカラーはサポートしていません。
	対応画面サイズ	ORG 画面 水平： 最大 510 ピクセル 垂直： 特に制限なし 拡大画面 水平： 最大 1024 ピクセル 垂直： 特に制限なし 注) 本マイコンの LCDC がサポートする最大表示サイズは表示パネルの条件により異なります。詳細は 3.19 LCD コントローラを参照してください。
	補正機能	周期ポイント補正機能、終端ポイント補正機能あり
縮小機能	縮小率・補間データ数	オリジナル画素間に 255 ポイントのデータを補間可能 (水平方向／垂直方向の独立設定可能) オリジナル画素間の 255 ポイントのデータを出力可能 (255/N/M 倍設定可能 : N=1~255、M=1~16)
	対応データフォーマット	Digital RGB 注) YUV フォーマットのデータはサポートしていません。
	対応画面色数	64K カラー(16bpp) モノクロ、モノクロ諧調およびその他の色数のカラーはサポートしていません。
	対応画面サイズ	ORG 画面 水平： 最大 511 ピクセル 垂直： 特に制限なし 拡大画面 水平： 最大 1024 ピクセル 垂直： 特に制限なし 注)本マイコンの LCDC がサポートする最大表示サイズは表示パネルの条件により異なります。詳細は 3.19 LCD コントローラを参照してください。
	補正機能	周期ポイント補正機能、終端ポイント補正機能あり

3.19.2.2 スケーラ処理メカニズム

1) 基本構造

LCDDA のスケーラ機能では、Bi-Cubic (バイキュービック) 法を使用して、オリジナル画素間に最大 255 ポイントの補間データを挿入することが出来ます。

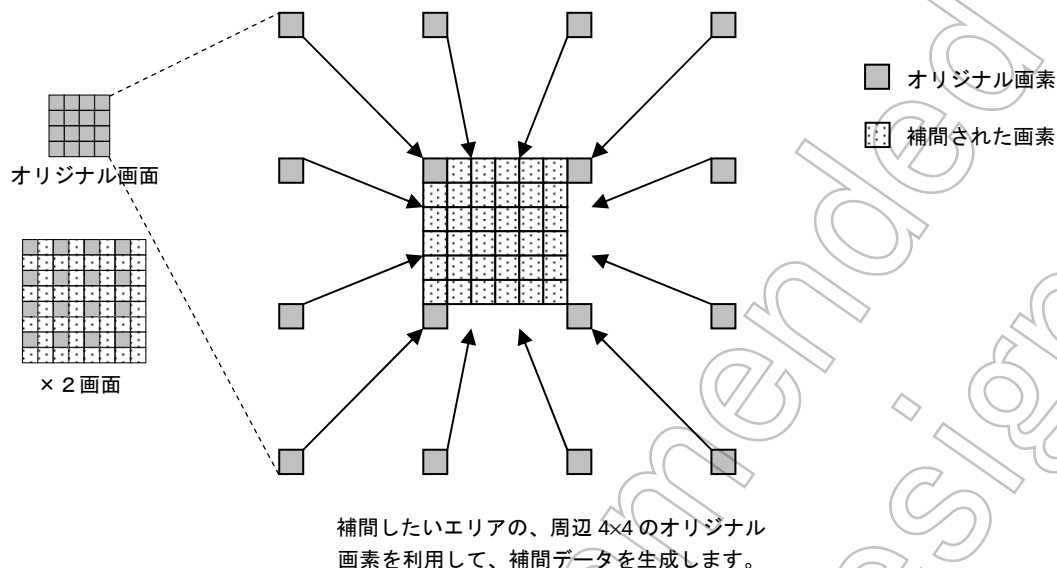


図 3.19.3 データ補間

本手法では、補間するエリア周辺 4×4 ポイントの画素の画像データから、最大 256 倍(8bit)の画像データを H/W が自動計算します。

オリジナルの画素から生成された 256 ポイント (オリジナル画素を含む) の画素を、n_Step ごとに出力することが可能です。

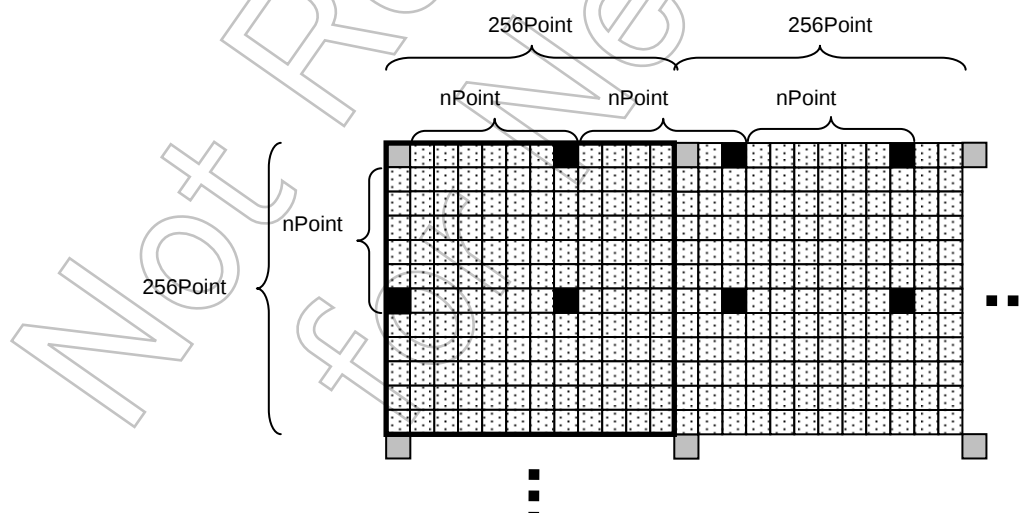


図 3.19.4 スケーリング方法

(Step 数が 1~255 ならば拡大、257 以上であれば縮小となります)

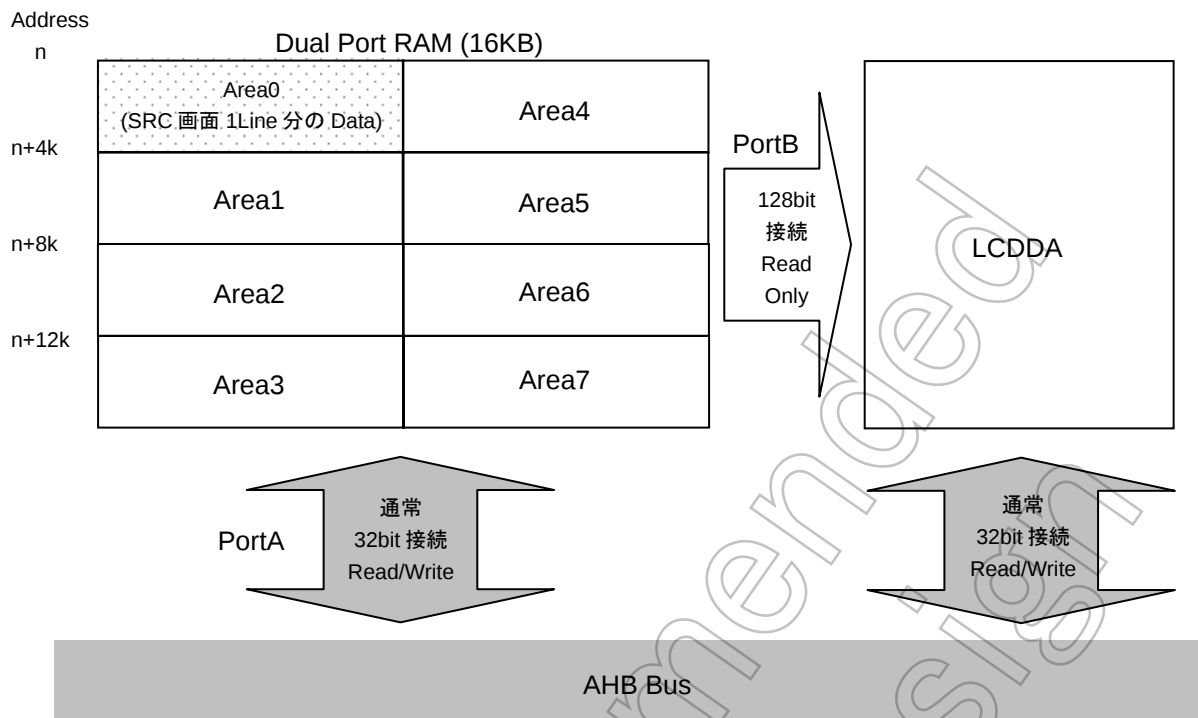


図 3.19.5 メモリとの接続と基本動作

スケーラ機能を使用する場合、オリジナルの画面データ(RGB)は専用の DualPortRAM に書き込んでおく必要があります。

先に説明した様に、補間データを生成するには、 4×4 のオリジナル画素が必要です。その為、BC_Expander 回路では、4 ライン分のオリジナル画面データが揃った時点で、補間データ生成を開始することが出来ます。

LCDDA の BC_Expander 回路は 16kbyte の Dual Port RAM と 128bit 幅のバス (PortB) で接続されています。

この 16Kbyte の Dual Port RAM は、通常の RAM としても使用可能であり、AHB バスに 32bit 幅のバス (PortA) で接続されています。

また、この 16 Kbyte の Dual Port RAM は、2Kbyte $\times$ 8 のエリアに分けられており、各エリアにはオリジナルの画面の 1Line 分のデータを準備します(最大 510 ピクセル:Dummy+510+Dummy)。

BC_Expander 回路は 4 ライン分のデータ (例えばエリア 0~エリア 3) が準備された時点で、計算を開始することが出来ます。その後、1 ライン分のデータが追加 (エリア 4) される度に、再び 4 ライン分のデータが準備され (エリア 1~エリア 4)、次の計算を開始します。

この要領で、エリアがループし、計算を行っていきます。

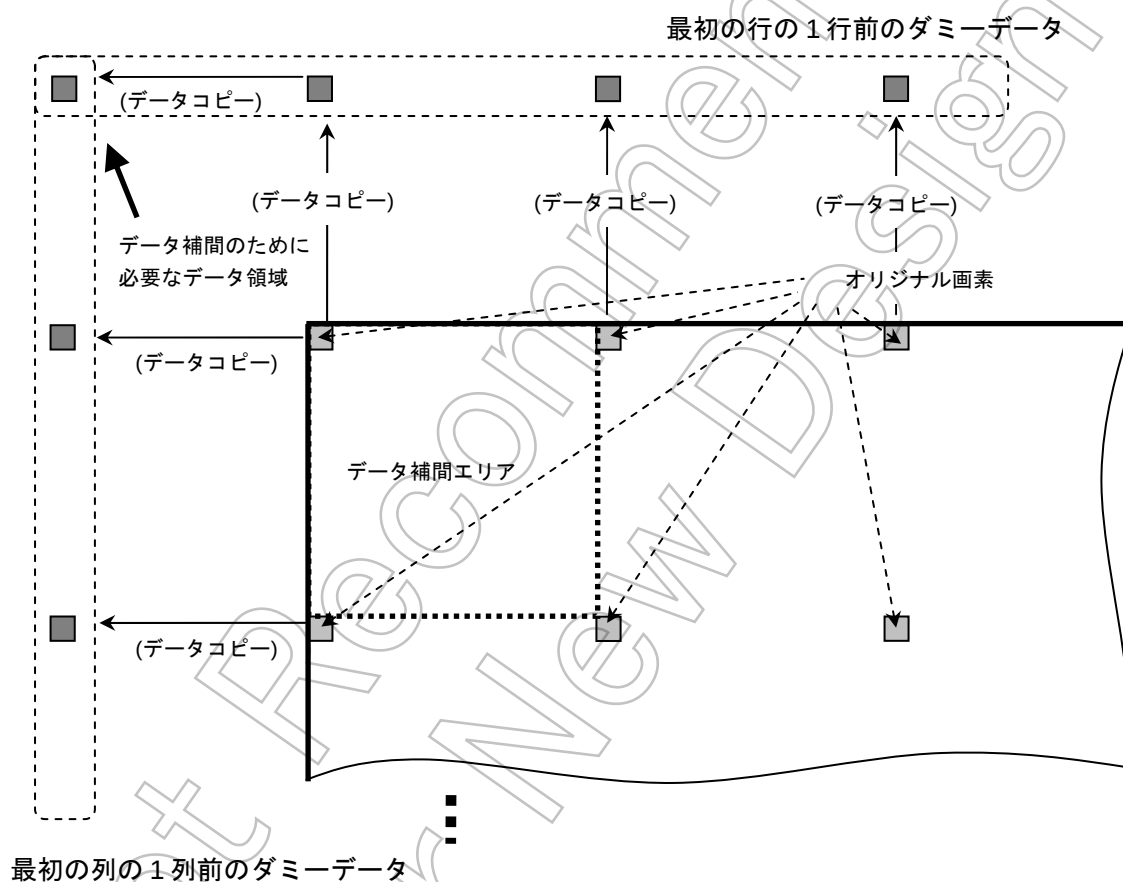
3.19.2.3 補正処理

スケーラ機能にて、拡大・縮小処理を行う場合に、サンプリングポイントの補正機能をサポートしています。この機能を使用することによって、より自然な画面を表現することが出来ます。

補正機能には、大きく「エッジデータ自動追加」機能と、「サンプリング補正」機能の2つの機能があります。

1) エッジデータ自動追加機能

前項で説明したように、スケーラ機能では、補間する周辺の4×4のオリジナル画素が必要です。そのため、画素のエッジの領域ではオリジナル画素が準備できません。最初の行の1行前、最後の行の1行後、または最初の列の1列前、最後の列の1列後のダミーデータを準備する必要があります。



本機能は、このダミーデータを自動的に準備することが出来ます。

LCDDA に接続されているデュアルポート RAM にオリジナル画像を設定する例を以下に示します。デュアルポート RAM 内にはエッジ処理のためのダミーデータの領域も準備しておく必要があります。

ここでは、エッジデータ自動追加機能を使用して、1 行が 510 ピクセルで 16bit カラーの設定例を示します。

- 先頭の 1 行はダミーデータ領域となります。そのため、0xF800_4000~0xF800_47FF 番地には画像データを設定する必要はありません。
- 2 行目以降の先頭画素と最終画素もダミーデータ領域となります。

オリジナルデータ: 510 ピクセル、16bit カラーの場合

;内蔵 RAM エリア 0

0xF800_4800 ; Dummy データ(何も設定する必要はありません)

0xF800_4802 ; 有効データ(最初のピクセルデータをセット)

⋮

⋮

0xF800_4BFC ; 有効データ(最後のピクセルデータをセット)

0xF800_4BFE ; Dummy データ(何も設定する必要はありません)

~0xF800_4FFC

注) 16bit カラー、一行の最大画素数は 510 ピクセルです。

510 ピクセル以上の画像を拡大・縮小する際は、画像の分割が必要です。

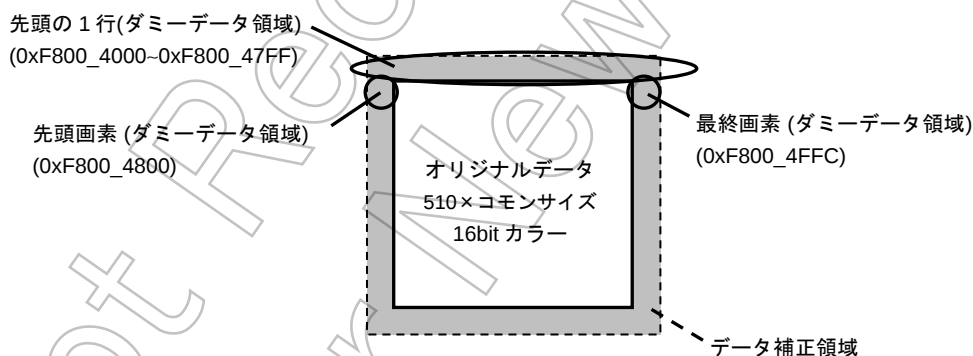


図 3.19.6 オリジナルデータのイメージ図 (510×コモンサイズ、16bit カラー)

2) サンプリング補正機能

スケーラ機能では、オリジナルの画素数に対して、 $256/(n \times m)$ 倍の拡大・縮小が可能です ($n=1\sim 255$ 、 $m=1\sim 16$)。

拡大方式は、上記計算式で設定される為小数点以下の端数が生じますが、実際のサンプリングポイントは整数で設定する為、誤差が生じます。

そのため、画面全体の累積誤差を適切なポイントで補正する必要があります。

本回路では、X方向の最初のサンプリングポイントに、オフセットを追加する「オフセット機能」、設定されたサンプリングポイントが、あるポイントになった場合に、そのポイントをオリジナル画面のポイントに補正する「周期補正機能」の二つの補正機能をサポートしています。

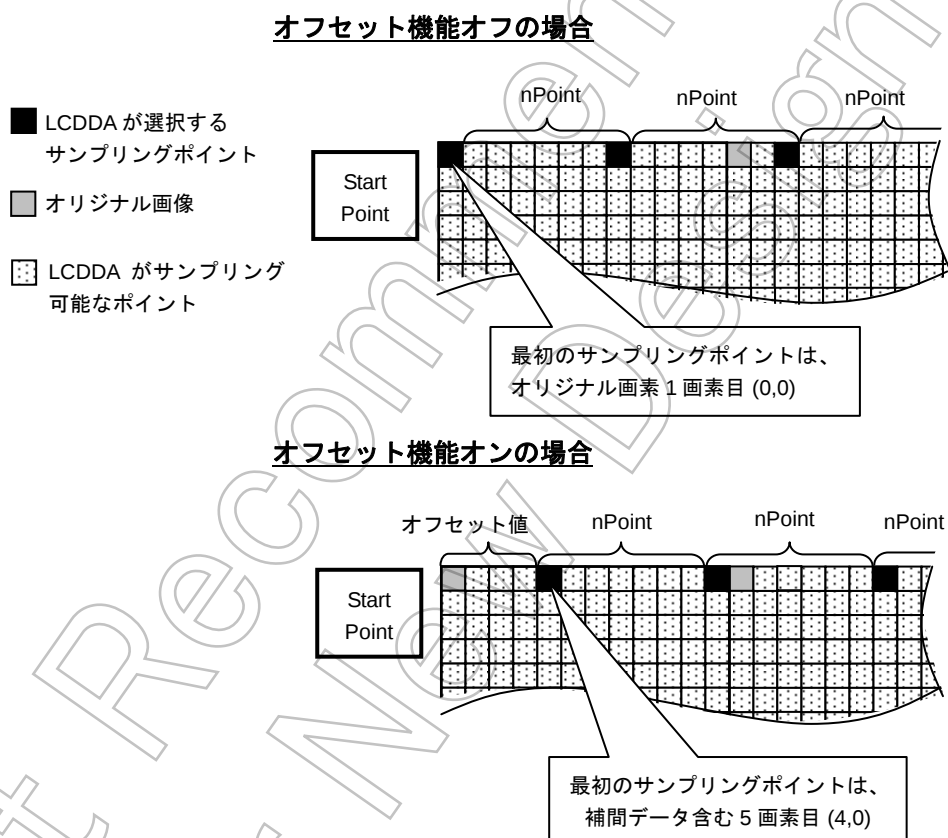


図 3.19.7 オフセット機能

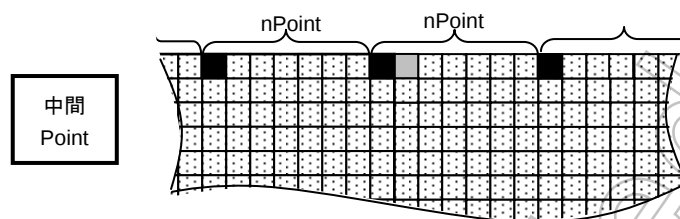
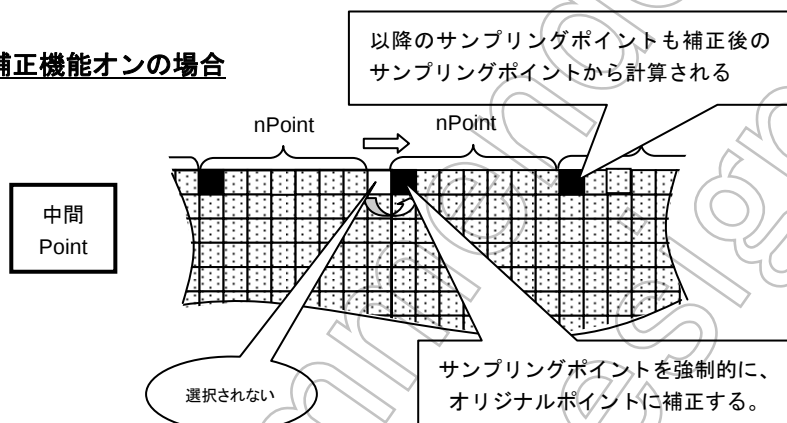
周期補正機能オフの場合**周期補正機能オンの場合**

図 3.19.8 周期補正機能

拡大処理例

補正を含めた、拡大処理を行う際の設定方法例を以下に説明します。

- ・ X 方向のオリジナル画素数：OX ピクセル
- ・ X 方向の拡大後の画素数：GX ピクセル

$$\begin{aligned} (\text{サンプリング間隔}) &= (\text{補間可能な最大画素数}) \div (\text{拡大後の画素数}) \\ &= \{(OX-1) \times 256\} \div GX \end{aligned}$$

例 1) 128 ピクセルの画素を、256 ピクセルに拡大する場合

$$\begin{aligned} & (127 \times 256) \div 256 \\ = & (32512) \div 256 \\ = & 127 \end{aligned}$$

例 2) 128 ピクセルの画素を、255 ピクセルに拡大する場合

$$\begin{aligned} & (127 \times 256) \div 255 \\ = & (32512) \div 255 \\ = & 127.49... \end{aligned}$$

整数設定で、小数点以下は切り捨てが必要ですので、例 1、2 共に、同一のサンプリング間隔(127)となります。

127=0x7F と設定します。127 ピクセル毎にデータをサンプリングした場合、各々のサンプリングポイントは以下のようになります。

例 1) のサンプリングポイント

最初のポイント：	0
2 番目のポイント：	127
3 番目のポイント：	254
↓	↓
254 番目のポイント：	32258
255 番目のポイント：	32385
256 番目のポイント：	32512

例 1) では、256 番目のポイントは 32512 となり、最終データを利用します。

また、例 2) の場合でも、同一のサンプリング間隔になりますので、255 番目のポイントは、32385 となり、残りのポイントからは画素が使用されず、捨てられてしまいます。

例 2) のサンプリングポイント

最初のポイント：	0
2 番目のポイント：	127
3 番目のポイント：	254
↓	↓
254 番目のポイント：	32258
255 番目のポイント：	32385

32386~32512 のデータは利用されない

これでは、若干左側に片寄った拡大画像になってしまいます。

そこで、より自然な拡大を行うために、補正機能を使用します。

【オフセット補正例】

最初のサンプリングポイントにオフセットを追加して利用されないデータを左右に均等に分割する機能です。

移動して、全体のサンプリングポイントを中央に移動します。

最終サンプリングポイント(255番目のポイント)と、生成可能な最終サンプリングポイントの差の半分をオフセットします。

単純オフセット補正の例（前述、例2の場合）

$$\begin{aligned} \text{(オフセット補正值)} &= \{ (\text{補間可能な最大画素数}) - (\text{整数化されたサンプリング間隔}) \times GX \} \div 2 \\ &= (32512 - 32385) \div 2 = 63.5 \end{aligned}$$

X方向のオフセット値は、整数しか設定出来ませんので、63(oct)=3F(hex)と設定します。

同様にY方向のオフセット値も、整数しか設定出来ませんので、63(oct)=3F(hex)と設定します。

3.19.2.4 Blend処理機能

表 3.19.3 Blend 機能

機能	機能詳細	説明・規格
BLEND 機能	α BLEND	各々の画面を RGB 別に、256 段階(0~255)設定可能
	カラー+モノクロ BLEND	2 値(モノクロ)データを、(1)データ、(0)データをそれぞれカラーに、パレットにて変換可能
	フォント	画像上にモノクロ文字の上書き
	対応データフォーマット	Digital RGB 注) YUV フォーマットのデータはサポートしていません。
	対応画面色数	64k カラー(16bpp) 注)モノクロ、モノクロ諧調およびその他の色数のカラーはサポートしていません。
	対応画面サイズ	水平： 最大 1024 ピクセル 垂直： 最大 1024 ピクセル 注)本マイコンの LCDC がサポートする最大表示サイズは表示パネルの条件によります。詳細は 3.18 LCD コントローラを参照してください。1024×1024 です。

3.19.2.5 ブレンド処理メカニズム

LCDDA のブレンド処理は、まず、2 枚の画面データを各々ピクセル単位に分解し、さらに RGB 単位（各 8 ビット：計 24 ビット）にまで、分解します。

この分解された RGB データに、各々 256 段階の重み(0x00/0x100~0xFF/0x100)をつけて、加算結果のデータを出力します。

ソース 0 の画素 : R_{S0}, G_{S0}, B_{S0} :

ソース 1 の画素 : R_{S1}, G_{S1}, B_{S1} :

ソース 0 の重み, :

$LDADDRSRC0<RDRSRC0[7:0]> : R_{S0RATIO},$

$LDADDRSRC0<GDRSRC0[7:0]> : G_{S0RATIO},$

$LDADDRSRC0<BDRSRC0[7:0]> : B_{S0RATIO}$

ソース 1 の重み :

$LDADDRSRC1<RDRSRC1[7:0]> : R_{S1RATIO},$

$LDADDRSRC1<GDRSRC1[7:0]> : G_{S1RATIO},$

$LDADDRSRC1<BDRSRC1[7:0]> : B_{S1RATIO}$

計算手法 :

$$R_{DST} = (R_{S0} \times R_{S0RATIO}) + (R_{S1} \times R_{S1RATIO})$$

$$G_{DST} = (G_{S0} \times G_{S0RATIO}) + (G_{S1} \times G_{S1RATIO})$$

$$B_{DST} = (B_{S0} \times B_{S0RATIO}) + (B_{S1} \times B_{S1RATIO})$$

よって、2 枚の画面の重みの和 (R の場合 : $<RDRSRC0[7:0]> + <RDRSRC1[7:0]>$) は 0x100 を超えないように設定することが必要です。

注) 加算した RGB データが 0x100 を超えるような場合、正常な BLEND は出来ません。

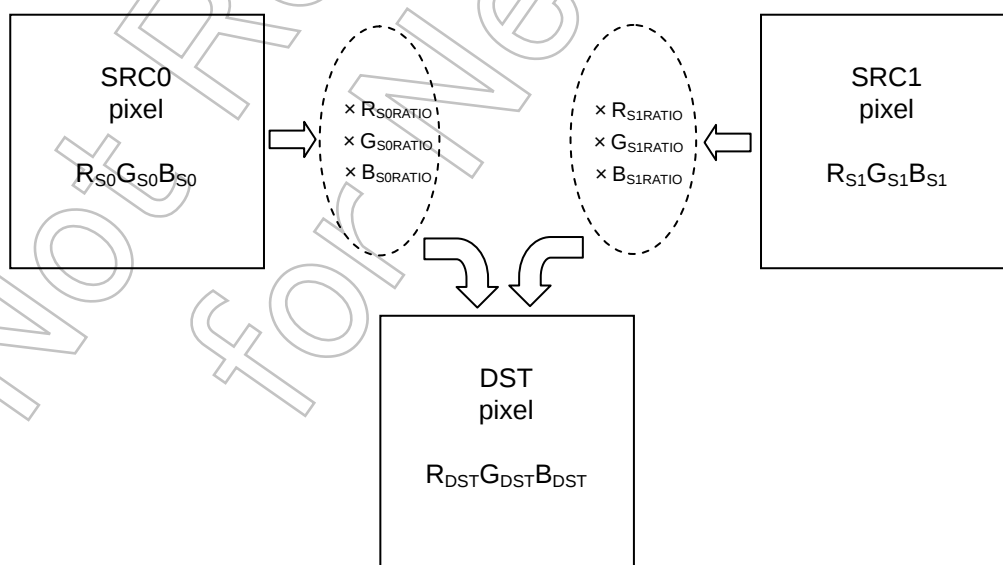
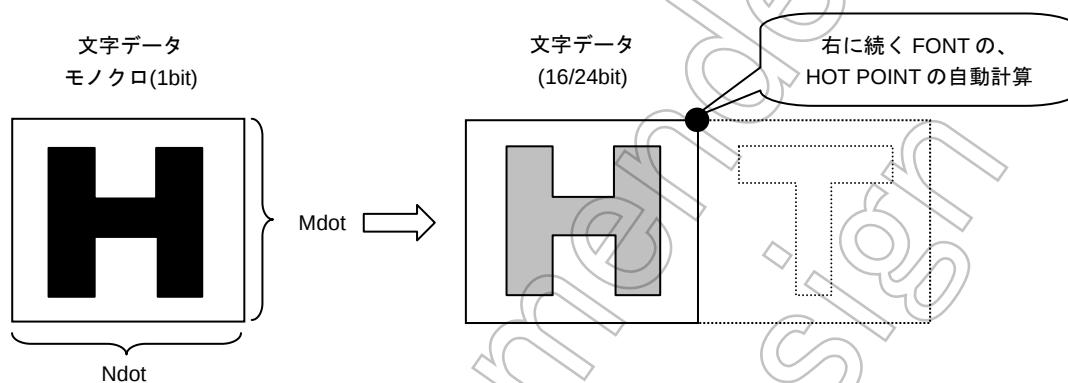


図 3.19.9 BLEND 処理

3.19.2.6 モノクロブレンド機能

モノクロブレンド機能は、モノクロ（2 値）で定義されたデータを、カラーデータに変換してカラー画面上に上書きする機能です。矩形領域のブレンドと文字の上書き（FONT DRAW）が可能です。また、文字データの連続書き込みに優れた自動アドレス計算もサポートしています。

- 連続したアドレスで定義された $N \times M$ の FONT を、左→右の方向に連続描画が可能(次の FONT の HOT_POINT(左上の頂点のアドレス)を自動計算する)。



3.19.2.7 回転機能

表 3.19.4 回転機能

機能	機能詳細	説明・規格
回転機能	回転角度	90 度、180 度、270 度、水平方向ミラー反転、垂直方向ミラー反転
	対応データフォーマット	Digital RGB 注) YUV フォーマットのデータはサポートしていません。
	対応画面色数	64K カラー(16bpp) 注)モノクロ、モノクロ諧調およびその他の色数のカラーはサポートしていません。
	対応画面サイズ	水平： 最大 1024 ピクセル 垂直： 最大 1024 ピクセル 注)本マイコンの LCDC がサポートする最大表示サイズは表示パネルの条件によります。詳細は 3.19 LCD コントローラを参照してください。

3.19.2.8 回転処理

回転処理は、LCDDA がオリジナル画面をリードし、コピーバックする際に、アドレスを計算して回転を実現します。転送先のアドレスのスタートポイントと、X 方向、Y 方向を各々、加算 (INCREMENT) するの、減算 (DECREMENT) するの、で、回転の形を制御します。

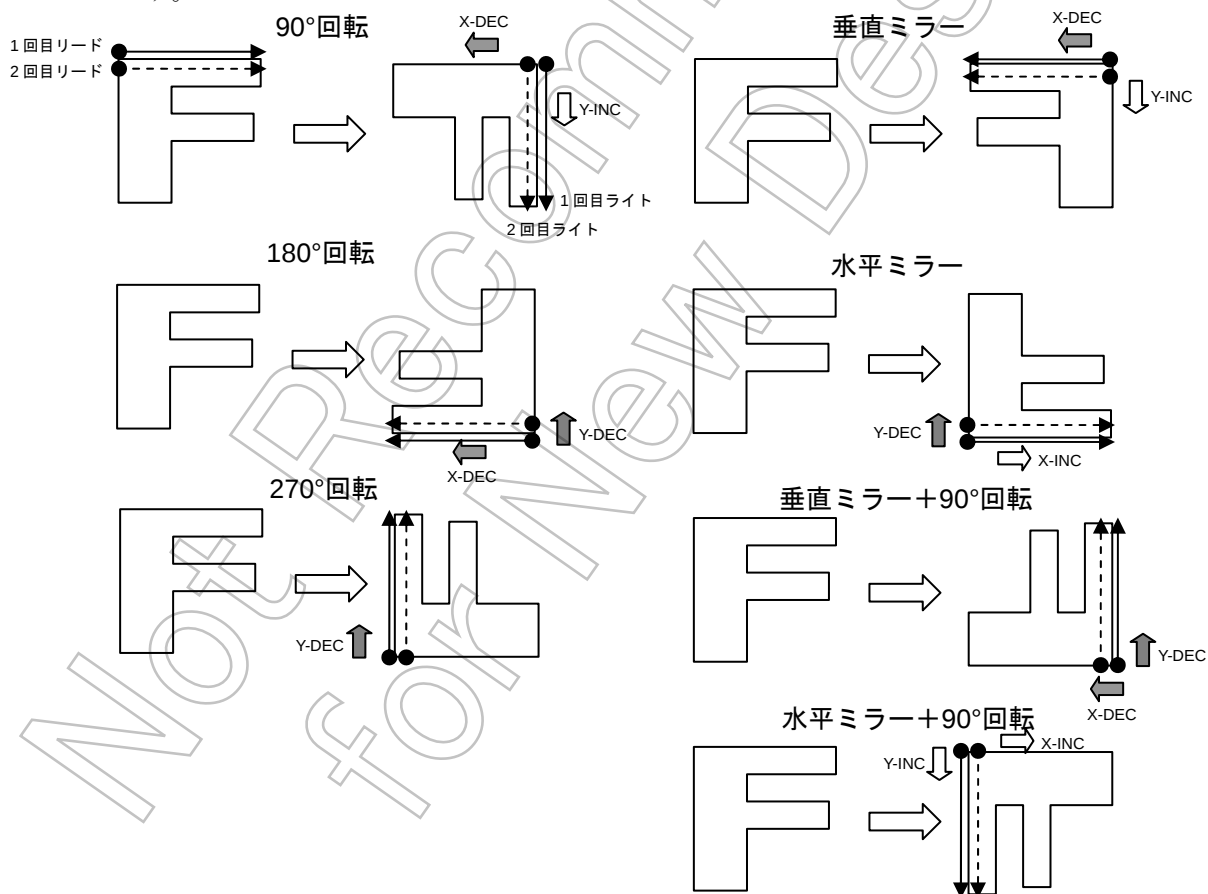


図 3.19.10 回転処理

注) 回転機能は、指定された矩形領域をすべて回転しますので、フォント重ね機能と回転機能を併用した場合フォントと背景がすべて回転されてしまいます。使用時には注意してください。

3.19.3 各モードの動作切り替え

画像合成の各動作モードについて説明します。

LCDDA は以下の回路と使用画像データの組み合わせで各モードを実現しています。

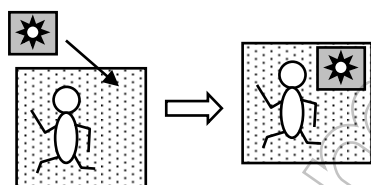
Mode	元画像データ設定		使用回路				
	Source0	Source1	BC Expander	Blender			転送アドレス 制御回路
				色変換	エリア設定	重ね	
機能説明	重ねあわせ時のみ使用する画像データです。 注 2)	重ねあわせ時および単純転送時に使用する画像データです。注 2)	拡大および縮小を実行する回路です。	色変換機能を実行する回路です。	重ね合わせるエリアをフィルタリングする回路です。	画像の重ね合わせを実行する回路です。	回転機能と単純縮小を実行する回路です。

注 1) BC Expander と Blender 回路の同時使用はできません

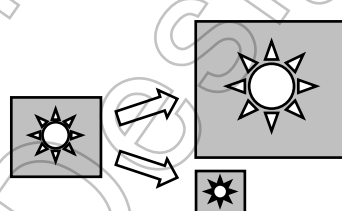
注 2) スケーラモード時には Source0、Source1 は使用しません。

LDACR1<OPMODE[4:0]>で選択し、以下表内の機能をサポートしています。

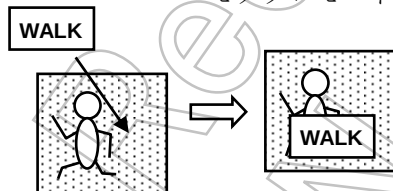
画像に、画像の上書き
ノーマル モード



画像の拡大／縮小
スケーラモード

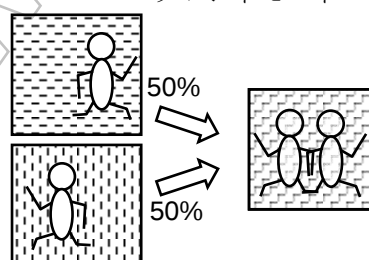


画像にモノクロデータの上書き
モノクロモード

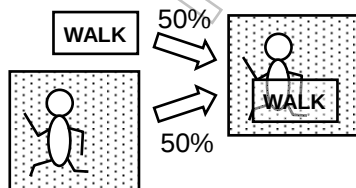


モノクロ反転のデータも対応

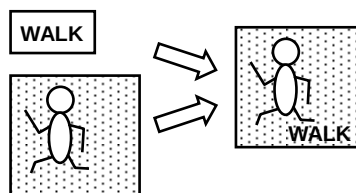
画像の合成
ブレンドモード



画像にモノクロデータの合成
モノクロブレンドモード



モノクロ反転のデータも対応



各動作モード	説明
ノーマルモード	色変換、ブレンド等を行わない、画像データの単純転送モードです。切り出し画面（矩形領域）を、背景画に上書きします。
スケーラモード	画像の拡大縮小モードです。
モノクロモード	モノクロの切り出し画面（矩形領域）を、カラーデータに拡張して、背景画に上書きします。 モノクロ(1bit)のデータを 16bit のカラーデータに拡張して転送するモードです。
モノクロインバートモード	モノクロの切り出し画面（矩形領域）の色を反転した後、カラーデータに拡張して、背景画に上書きします。 モノクロ(1bit)のデータを反転し、16bit のカラーデータに拡張して転送するモードです。
ブレンドモード	2 枚のカラー(16bit)画面を α ブレンドするモードです。
モノクロブレンドモード	モノクロデータをソースとカラーの 2 枚の画面を α ブレンドするモードです。モノクロで指定された画像をカラー変換し、カラー画像に α BLEND する際に使用します。
モノクロインバートブレンドモード	モノクロデータを反転して、カラーの 2 枚の画面を α ブレンドするモードです。モノクロ BLEND モードと、同様ですが、モノクロ→カラー変換時に、モノクロデータを反転してから、処理します。
フォントドローモード	モノクロソースとカラーの 2 枚の画面のモノクロデータの"1"データ部分のみを、諧調表示に変換してカラー画面上にブレンドするモードです。スーパーインポーズ等の処理に便利です。

次ページより各々の動作について説明いたします。

1. ノーマルモード

単純なデータ転送モードです。

このモードは、

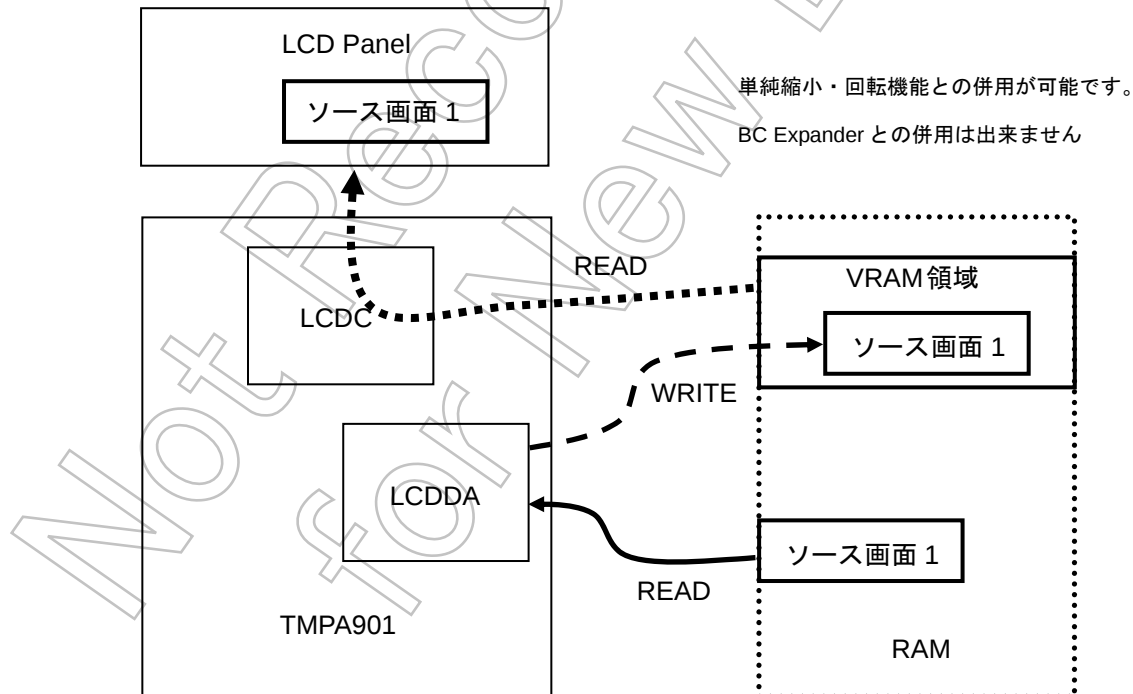
- ・ソース画像 1(S1)のカラー原画から、単純画像転送
- ・S1 のカラー原画から、単純間引きによる縮小転送
- ・S1 のカラー原画から、画像データの回転転送

等に使用します。

Normal	元画像データ設定		使用回路				
	Source0	Source1	BC Expander	Blender			転送アドレス 制御回路
				色変換	エリア設定	重ね	
機能説明	重ねあわせ時のみ使用する画像データです。	重ねあわせ時および単純転送時に使用する画像データです。	拡大および縮小を実行する回路です。	色変換機能を実行する回路です。	重ね合わせるエリアをフィルタリングする回路です。	画像の重ね合わせを実行する回路です。	回転機能と単純縮小を実行する回路です。
使用/未使用	未使用	使用 内蔵 RAM、外部 RAM 共に設定可能	未使用 使用出来ません	未使用	未使用	未使用	使用

応用例：

カラー画像の中に、もう一枚のカラー画像を上書き（Picture In Picture）するような用途に使用します。また、単純な DMA 転送としても使用できます。



2. スケーラモード

BC-Expander 回路の動作を制御し、拡大縮小をするモードです。

このモードでは、

- ・S1 のカラー原画から、バイキュービックを使用した拡大画像生成及び転送
 - ・S1 のカラー原画から、バイキュービックを使用した縮小画像生成及び転送
 - ・S1 のカラー原画から、バイキュービックを使用した拡大画像生成及び回転転送
 - ・S1 のカラー原画から、バイキュービックを使用した縮小画像生成及び回転転送
- 等に使用します。

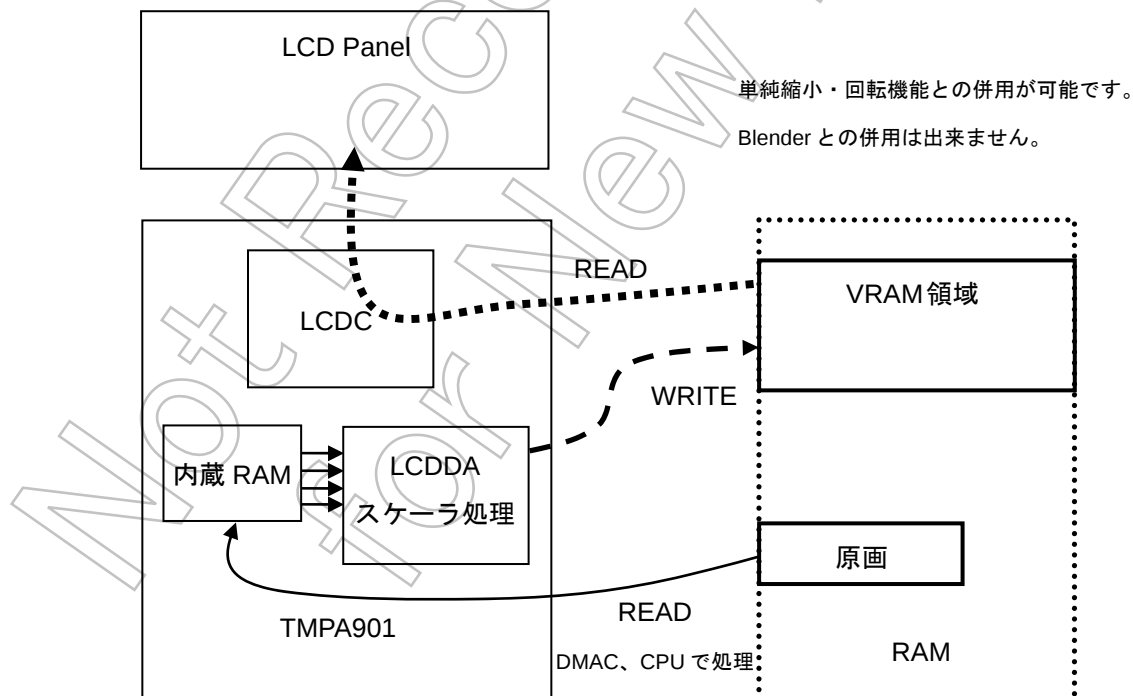
スケーラ モード	元画像データ設定		使用回路				
	Source0	Source1	BC Expander	Blender			転送アドレス 制御回路
				色変換	エリア設定	重ね	
機能説明	重ねあわせ時のみ使用する画像データです。注)	重ねあわせ時および単純転送時に使用する画像データです。注)	拡大および縮小を実行する回路です。	色変換機能を実行する回路です。	重ね合わせるエリアをフィルタリングする回路です。	画像の重ね合わせを実行する回路です。	回転機能と単純縮小を実行する回路です。
使用/未使用	未使用	未使用	使用	未使用 使用出来ません	未使用 使用出来ません	未使用 使用出来ません	使用

注) スケーラモード時には Source0、Source1 は使用しません。オリジナル画像がデュアルポート RAM 内 ((0xF800 4000)内蔵 RAM-0: 16KB) に格納されている前提です。

応用例：

小さな画像を拡大したり、大きな画像を縮小したりする際に使用します。

カラー画像の中に、もう一枚の画像を上書き（Picture In Picture）も併用できます。



3. モノクロモード

モノクロのソースを転送するモードです。

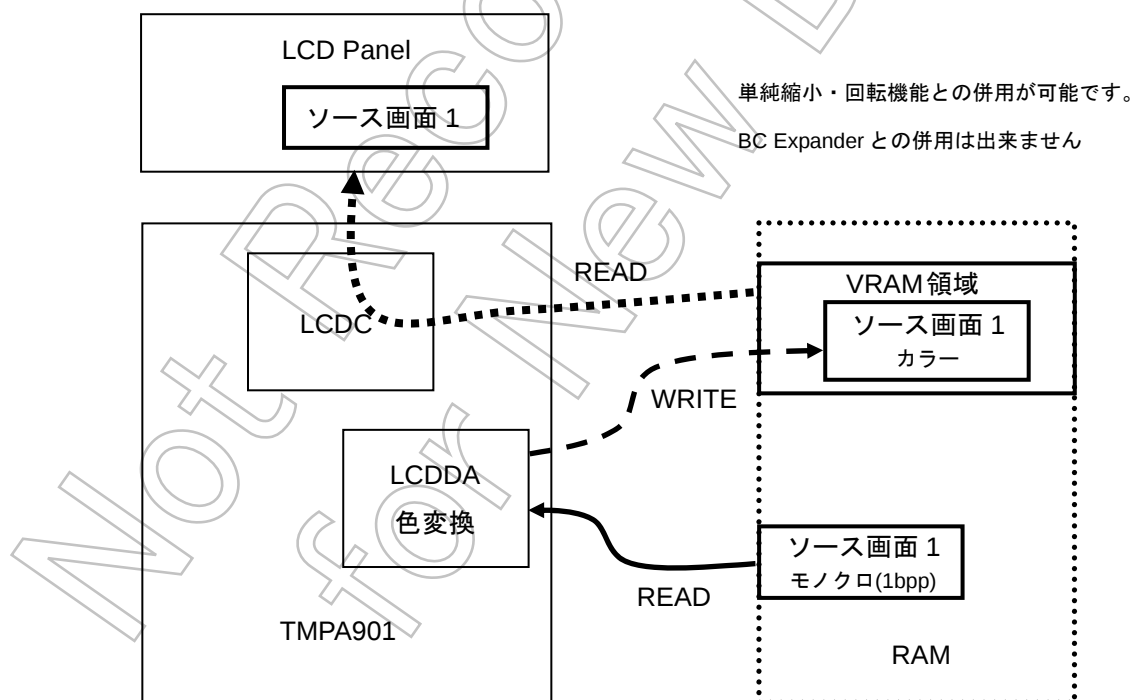
このモードは、

- ・ S1 のモノクロデータをカラーに変換後、画像転送
 - ・ S1 のモノクロデータをカラーに変換後、単純間引きによる縮小転送
 - ・ S1 のモノクロデータをカラーに変換後、画像データの回転転送等
- 等に使用します。

モノクロ	元画像データ		使用回路				
	Source0	Source1	BC Expander	Blender			転送アドレス 制御回路
				色変換	エリア設定	重ね	
機能説明	重ねあわせ時のみ使用する画像データです。	重ねあわせ時および単純転送時に使用する画像データです。	拡大および縮小を実行する回路です。	色変換機能を実行する回路です。	重ね合わせるエリアをフィルタリングする回路です。	画像の重ね合わせを実行する回路です。	回転機能と単純縮小を実行する回路です。
使用/未使用	未使用	使用 内蔵 RAM、外部 RAM 共に設定可能	未使用 使用出来ません	使用	未使用	未使用	使用

応用例：

文字などのモノクロデータを、カラー画像の中に上書き（Picture In Picture）に使用します。



4. モノクロインバートモード

モノクロのソースのデータを反転して転送するモードです。

モノクロモードと、ほぼ同様ですが、モノクロデータを反転→カラー変換して転送します。

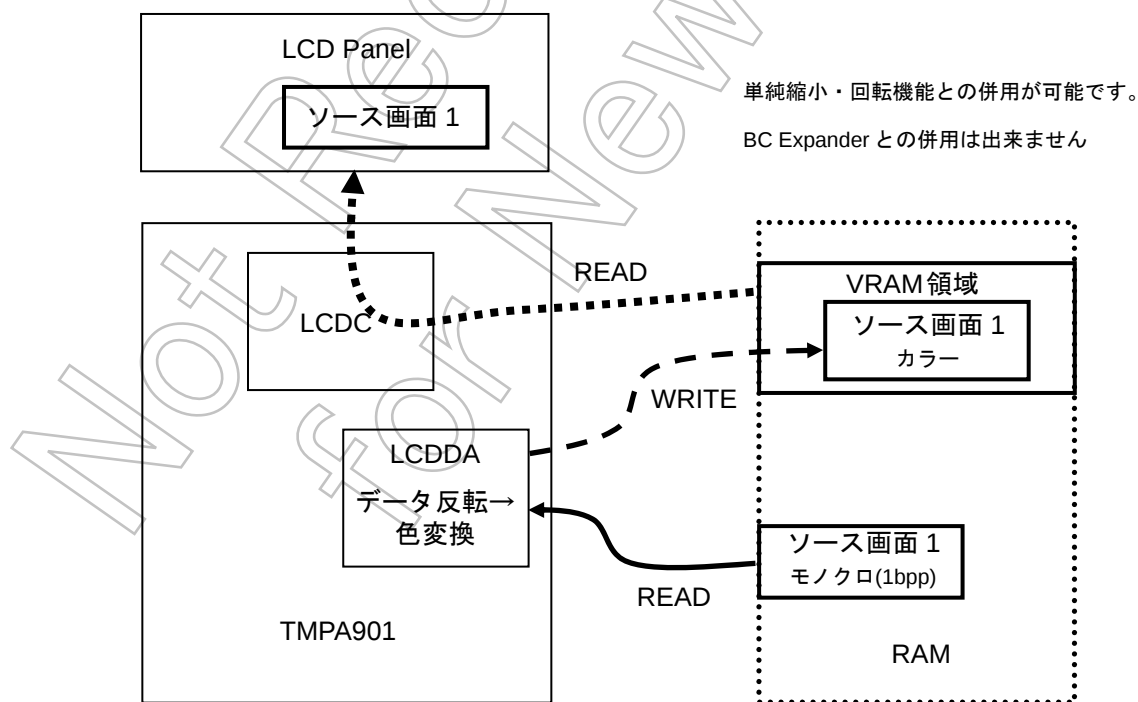
このモードは、

- ・S1のモノクロデータをカラーに変換後、画像転送
- ・S1のモノクロデータをカラーに変換後、単純間引きによる縮小転送
- ・S1のモノクロデータをカラーに変換後、画像データの回転転送等に使用します。

モノクロ INV モード	元画像データ		使用回路				転送アドレス 制御回路
	Source0	Source1	BC Expander	色変換	エリア設定	重ね	
機能説明	重ねあわせ時のみ使用する画像データです。	重ねあわせ時および単純転送時に使用する画像データです。	拡大および縮小を実行する回路です。	色変換機能を実行する回路です。	重ね合わせるエリアをフィルタリングする回路です。	画像の重ね合わせを実行する回路です。	回転機能と単純縮小を実行する回路です。
使用/未使用	未使用	使用 内蔵 RAM、外部 RAM 共に設定可能	未使用 使用出来ません	使用	未使用	未使用	使用

応用例：

文字などのモノクロデータを、インバート（反転）した後に、カラー画像の中に上書き（Picture In Picture）に使用します。



5. ブレンドモード

2枚のカラー（16bpp）画面をブレンドするモードです。

このモードは、

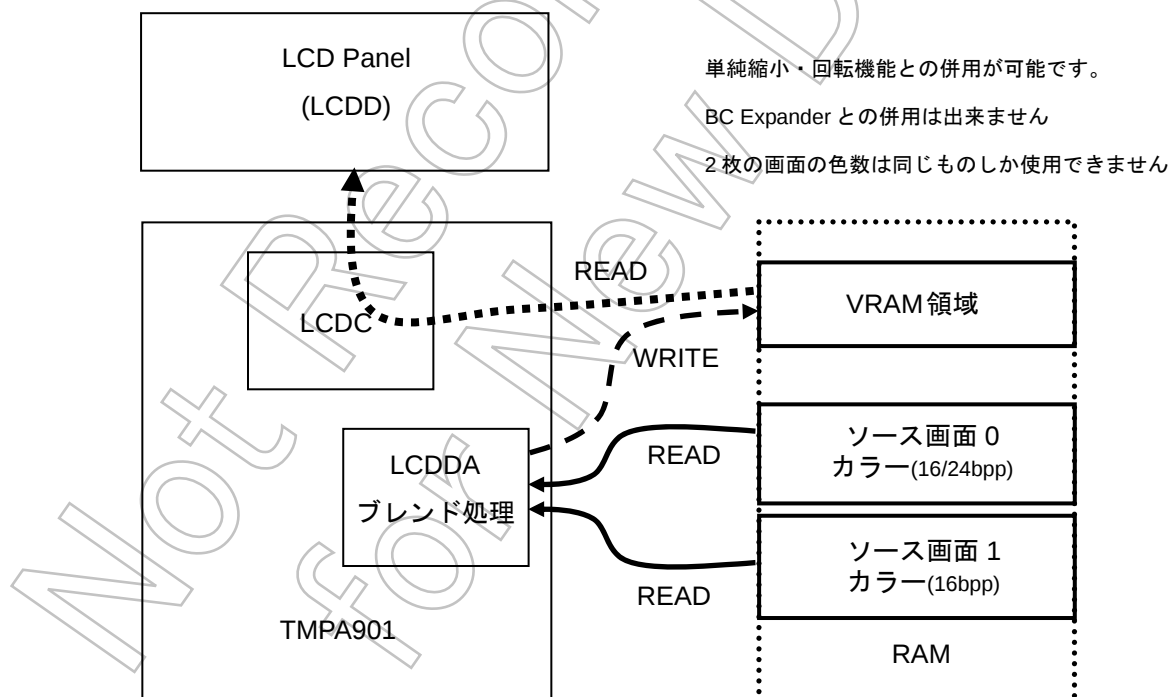
- ・S0とS1の2枚のカラーデータをブレンド後、画像転送
- ・S0とS1の2枚のカラーデータをブレンド後、単純間引きによる縮小転送
- ・S0とS1の2枚のカラーデータをブレンド後、画像データの回転転送

等に使用します。

BLEND	元画像データ		使用回路				
	Source0	Source1	BC Expander	Blender			転送アドレス 制御回路
				色変換	エリア設定	重ね	
機能説明	重ねあわせ時のみ使用する画像データです。	重ねあわせ時および単純転送時に使用する画像データです。	拡大および縮小を実行する回路です。	色変換機能を実行する回路です。	重ね合わせるエリアをフィルタリングする回路です。	画像の重ね合わせを実行する回路です。	回転機能と単純縮小を実行する回路です。
使用/未使用	使用 内蔵 RAM、外部 RAM 共に設定可能	使用 内蔵 RAM、外部 RAM 共に設定可能	未使用 使用出来ません	未使用	未使用	使用	使用

応用例：

2枚のカラー画像を切り替える際に、自然な切換え（ブレンド比を徐々に変更）が可能です。



6. モノクロ ブレンドモード

モノクロソースとカラー（16bpp）の2枚の画面をブレンドするモードです。

このモードは、

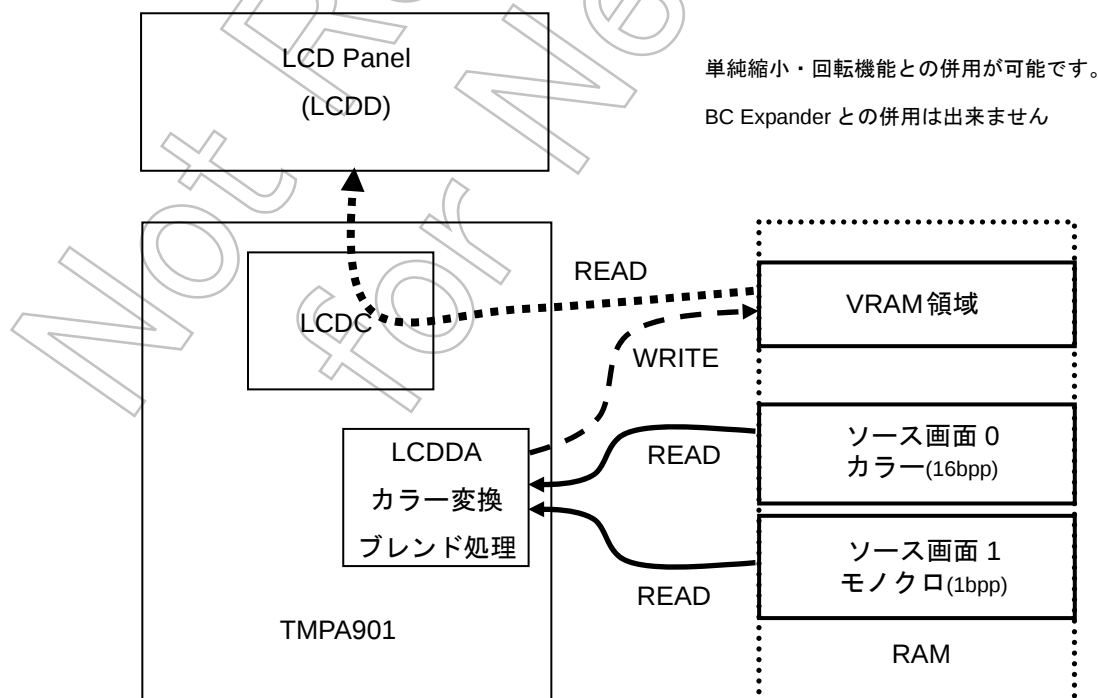
- ・S0のカラーデータと、S1のモノクロデータをカラー化したデータをブレンド後、画像転送
 - ・S0のカラーデータと、S1のモノクロデータをカラー化したデータをブレンド後、単純間引きによる縮小転送
 - ・S0のカラーデータと、S1のモノクロデータをカラー化したデータをブレンド後、回転転送。
- 等に使用します。

モノクロ BLEND	元画像データ		使用回路				
	Source0	Source1	BC Expander	Blender			転送アドレス 制御回路
				色変換	エリア設定	重ね	
機能説明	重ねあわせ時のみ使用する画像データです。	重ねあわせ時および単純転送時に使用する画像データです。	拡大および縮小を実行する回路です。	色変換機能を実行する回路です。	重ね合わせるエリアをフィルタリングする回路です。	画像の重ね合わせを実行する回路です。	回転機能と単純縮小を実行する回路です。
使用/未使用	使用 内蔵 RAM、外部 RAM 共に設定可能	使用 内蔵 RAM、外部 RAM 共に設定可能	未使用 使用出来ません	使用	未使用	使用	使用

注) モノクロデータは必ず Source1 を使用してください。

応用例：

カラー画像とモノクロ画面を切り替える際に、自然な切換え（ブレンド比を徐々に変更）が可能です。モノクロデータの内、“1”のデータの色を R:0xFF、G:0xFF、B:0xFF（白色）に設定し、変換時に 100%のブレンド比、“0”データの色変換時に 0%のブレンド比にすれば、白色限定の FONT ですが、重ね書きにも対応可能です。
（背景色が白の場合、対応できません）



7. モノクロインポートブレンドモード

モノクロソースの反転データと、カラー（16bpp）の2枚の画面をブレンドするモードです。モノクロ BLEND モードと、ほぼ同様ですが、モノクロ→カラー変換時に、モノクロデータを反転してから、処理します。

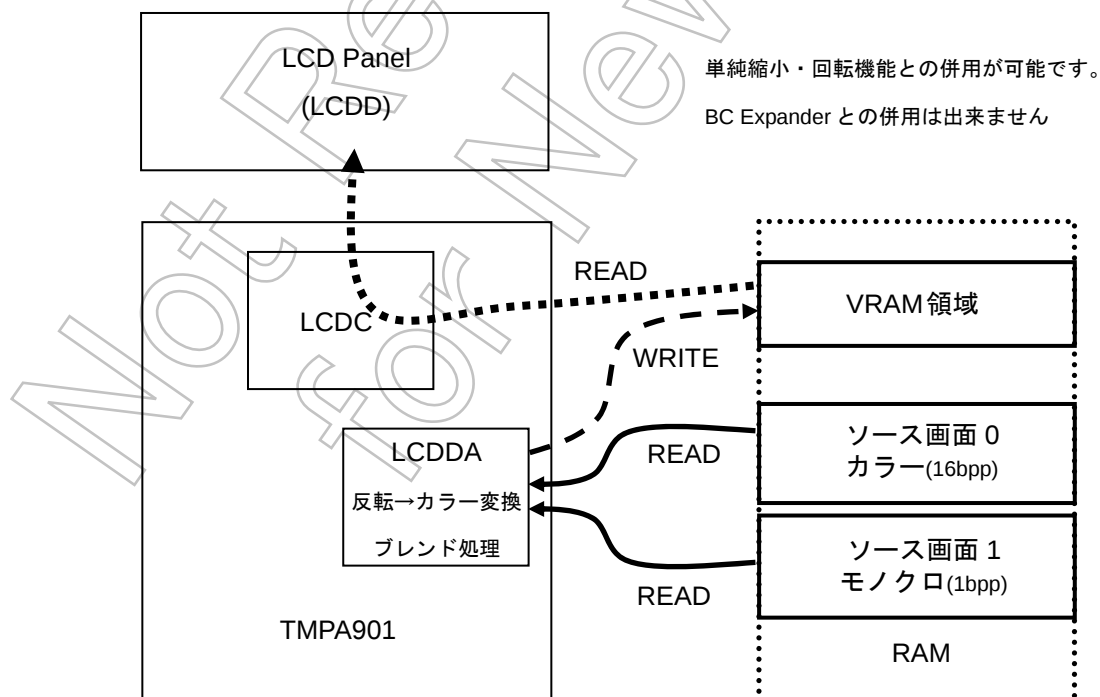
このモードは、

- ・S0 のカラーデータと、S1 のモノクロデータをカラー化したデータをブレンド後、画像転送
- ・S0 のカラーデータと、S1 のモノクロデータをカラー化したデータをブレンド後、単純間引きによる縮小転送
- ・S0 のカラーデータと、S1 のモノクロデータをカラー化したデータをブレンド後、回転転送等に使用します。

モノクロ INV	元画像データ		使用回路				
	Source0	Source1	BC Expander	Blender			転送アドレス 制御回路
				色変換	エリア設定	重ね	
機能説明	重ねあわせ時のみ使用する画像データです。	重ねあわせ時および単純転送時に使用する画像データです。	拡大および縮小を実行する回路です。	色変換機能を実行する回路です。	重ね合わせるエリアをフィルタリングする回路です。	画像の重ね合わせを実行する回路です。	回転機能と単純縮小を実行する回路です。
使用/未使用	使用 内蔵 RAM、外部 RAM 共に設定可能	使用 内蔵 RAM、外部 RAM 共に設定可能	未使用 使用出来ません	使用	未使用	使用	使用

応用例：

カラー画像とモノクロ画面を切り替える際に、自然な切換え（ブレンド比を徐々に変更）が可能です。



8. フォントドローモード

モノクロのソースとカラー(16bpp)の2枚の画面をブレンドするモードです。
モノクロで指定された(S1)画像の、“1”データの箇所のみをビット拡張し、ブレンドします。

このモードは、

- ・S0のカラーデータと、S1のモノクロデータの“1”データ部分のみをビット拡張し、ブレンド後、画像転送
- ・S0のカラーデータと、S1のモノクロデータの“1”データ部分のみをビット拡張し、ブレンド後、単純間引きによる縮小転送
- ・S0のカラーデータと、S1のモノクロデータの“1”データ部分のみをビット拡張し、ブレンド後、回転転送に使用します。

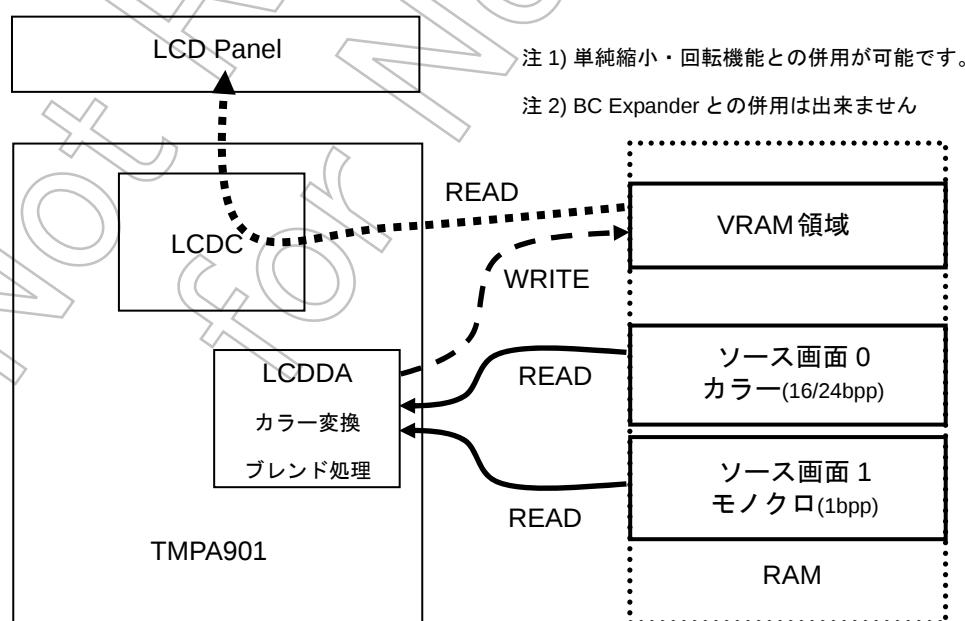
注) ビット拡張をしたフォント色は、LDAFCPSRC1<RFONT[7:0]>、<GFONT[7:0]>、<BFONT[7:0]>にて設定しますが、<RFONT[7:0]>=<BFONT[7:0]>の条件付指定となります。

フォント ドロー	元画像データ		使用回路				
	Source0	Source1	BC Expander	Blender			転送アドレス 制御回路
				色変換	エリア設定	重ね	
機能説明	重ねあわせ時のみ使用する画像データです。	重ねあわせ時および単純転送時に使用する画像データです。	拡大および縮小を実行する回路です。	色変換機能を実行する回路です。	重ね合わせるエリアをフィルタリングする回路です。	画像の重ね合わせを実行する回路です。	回転機能と単純縮小を実行する回路です。
使用/未使用	使用 内蔵 RAM、外部 RAM 共に設定可能	使用 内蔵 RAM、外部 RAM 共に設定可能	未使用 使用出来ません	使用	未使用	使用	使用

応用例：

文字のみを、カラー画像の中に上書きします。

<注意> ブレンド後に転送実行される為、回転等の処理を行った場合背景文字だけでなく背景画面も回転しますのでご注意ください。



3.19.4 SFR

SFR のリストを以下に示します。

base アドレス= 0xF205_0000

Register Name	Address (base+)	Description
LDACR0	0x0000	LCDDA Control Register 0
LDADRSRC1	0x0004	LCDDA Density Ratio of Source 1 Picture
LDADRSRC0	0x0008	LCDDA Density Ratio of Source 0 Picture
LDAFCPSRC1	0x000C	LCDDA Replaced Font Area Color pallet of Source1
LDAEFCPSRC1	0x0010	LCDDA Replaced Except Font Area Color pallet of Source1
LDADVSR1	0x0014	LCDDA Delta Value (Read Step) address Register of Source 1
LDACR2	0x0018	LCDDA Control Register 2
LDADXST	0x001C	LCDDA X-Delta Value (Write Step) address Register of Destination
LDADYDST	0x0020	LCDDA Y-Delta Value (Write Step) address Register of Destination
LDASSIZE	0x0024	LCDDA Source Picture Size
LDADSIZE	0x0028	LCDDA Destination Picture Size
LDAS0AD	0x002C	LCDDA Source 0 Start Address
LDADAD	0x0030	LCDDA Destination Start Address
LDACR1	0x0034	LCDDA Control Register1
LDADVSR0	0x0038	LCDDA Delta Value (Read Step) address Register of Source 0

LCDDA は 14 種類のレジスタを持っています。これらは 32 ビットバスで CPU に接続されています。

1. LDACR0 (LCDDA Control Register 0)

Address = (0xF205_0000) + (0x0000)

Bit	Bit Symbol	Type	Reset Value	Description
[31:22]	—	—	Undefined	Read as undefined. Write as zero.
[21]	ERRINTF	RW	0y0	LCDDA 処理エラーフラグ READ 時 Write 時 0y0: 割り込み無 0y0: フラグクリア 0y1: 割り込み有 0y1: 無効
[20]	EINTF	RW	0y0	LCDDA 処理終了割り込み許可 (スケーラ/Rotation 機能で有効) READ 時 Write 時 0y0: 割り込み無 0y0: フラグクリア 0y1: 割り込み有 0y1: 無効
[19:18]	—	—	Undefined	Read as undefined. Write as zero.
[17]	ERRINTM	RW	0y0	LCDDA 処理エラー割り込み MASK 0y0: 割り込みマスク 0y1: 割り込み有効
[16]	EINTM	RW	0y0	LCDDA 処理終了割り込み MASK (スケーラ/Rotation 機能で有効) 0y0: 割り込みマスク 0y1: 割り込み有効
[15]	BCENYB	RW	0y0	Y 方向最終 LINE データ補正(最終ダミー行追加) 0y0: 停止 0y1: 動作
[14]	AUTOHP	RW	0y0	HOT ポイントの自動計算 0y0: 停止 0y1: 動作
[13]	DMAMD	RW	0y0	DMA 選択 0y0: Single 転送 0y1: Burst 転送
[12]	DMAEN	RW	0y0	DMA 許可 0y0: 停止 0y1: 許可
[11]	BCENYT	RW	0y0	Y 方向先頭 LINE データ補正(先頭ダミー行追加) 0y0: 停止 0y1: 動作
[10]	DTFMT	RW	0y0	表示色選択 0y0: Reserved 0y1: 64K 色 (16bit)
[9]	BCENX	RW	0y0	X 方向エッジデータ補正(左右ダミー列追加) 0y0: 停止 0y1: 動作
[8]	PCEN	RW	0y1	周期補正 0y0: 停止 0y1: 動作
[7:0]	S1ADR[31:24]	RW	0x00	SRC1 画面の先頭アドレス(32bit 中、上位 8bit)

(個別説明)

a. <ERRINTF>

LCD DA 回路の、処理エラー発生割り込みの状態を示します。

READ 時と WRITE 時でその意味が異なりますので御注意ください。

READ 時

WRITE 時

0y0: 割り込み無

0y0: フラグクリア

0y1: 割り込み有

0y1: 無効 (状態は変化しません)

b. <EINTF>

LCD DA の処理終了を示す割り込みの状態を示します。

この割り込みは、LCD DA のスケーラ機能と、BLEND、Rotation 機能の内部処理終了を示し、ライトバッファに蓄積されたデータ転送の終了は示していませんので注意が必要です。

スケーラ処理の場合は、オリジナル画面のラインごとに処理が実行される為、1 画面の拡大／縮小が終了するまでには、複数回の割り込みが発生します。

スケーラ処理以外は、1 回の要求処理を終了した時に割り込みが発生します。

また、READ 時と WRITE 時でその意味が異なりますので御注意ください。

READ 時

WRITE 時

0y0: 割り込み無

0y0: フラグクリア

0y1: 割り込み有

0y1: 無効 (状態は変化しません)

c. <ERRINTM>

LCD DA 回路の、処理エラー発生割り込みマスクを設定します。

0y0: エラー割り込みマスク

0y1: エラー割り込み有効

d. <EINTM>

LCD DA の処理終了を示す割り込みのマスクを設定します。

0y0: 処理割り込みマスク

0y1: 処理割り込み有効

e. <BCENYB>

補間スケーラ機能を利用する際の、Y 方向最終 LINE のダミーサンプリングポイントを自動追加する機能を制御します。

0y0: 停止

0y1: 動作

f. <AUTOHP>

HOT ポイントの自動計算動作を設定します。

0y0: 停止

0y1: 動作

g. <DMAMD>

LCDDA の DMA 転送モードを設定します。

0y0: Single 転送

0y1: Burst 転送

h. <DMAEN>

LCDDA の 1 画面分の処理終了を示す信号で、DMA 転送の許可を制御します。

0y0: DMA 停止

0y1: DMA 許可

i. <BCENYT>

補間スケーラ機能を利用する際の、Y 方向先頭 LINE のダミーサンプリングポイントを自動追加する機能を制御します。

0y0: 停止

0y1: 動作

j. <DTFMT>

LCDDA が取り扱う RGB データのフォーマットを定義します。

0y0: Reserved

0y1: 64K 色 (16bit データ)

k. <BCENX>

補間スケーラ機能を利用する際の、X 方向左右列のダミーサンプリングポイントを自動追加する機能を制御します。

0y0: 停止

0y1: 動作

l. <PCEN>

スケーラ機能を利用する際の、周期的なサンプリングポイントを補正する機能を制御します。

0y0: 停止

0y1: 動作

m. <S1ADR [31:24]>

LCDDA が処理をする元画面 (ソース画面 1) が格納されているメモリの、先頭番地を示します。32bit のアドレス空間の内、上位 8bit を設定します。

2. LDADRSRC1 (LCDDA Density Ratio of Source 1 Picture)

Address = (0xF205_0000) + (0x0004)

Bit	Bit Symbol	Type	Reset Value	Description
[31:24]	—	—	Undefined	Read as undefined. Write as zero.
[23:16]	BDRSRC1[7:0]	R/W	0x00	SRC1 画面の Blue データ濃度調整(256 段階) 0x00: 薄い ~ 0xFF: 濃い
[15:8]	GDRSRC1[7:0]	R/W	0x00	SRC1 画面の Green データ濃度調整(256 段階) 0x00: 薄い ~ 0xFF: 濃い
[7:0]	RDRSRC1[7:0]	R/W	0x00	SRC1 画面の Red データ濃度調整(256 段階) 0x00: 薄い ~ 0xFF: 濃い

ソース 1(フォアグラウンド)画面の濃度を、R、G、B ごとに調整します。

3. LDADRSRC0 (LCDDA Density Ratio of Source 0 Picture)

Address = (0xF205_0000) + (0x0008)

Bit	Bit Symbol	Type	Reset Value	Description
[31:24]	—	—	Undefined	Read as undefined. Write as zero.
[23:16]	BDRSRC0[7:0]	R/W	0x00	SRC0 画面の Blue データ濃度調整(256 段階) 0x00: 薄い ~ 0xFF: 濃い
[15:8]	GDRSRC0[7:0]	R/W	0x00	SRC0 画面の Green データ濃度調整(256 段階) 0x00: 薄い ~ 0xFF: 濃い
[7:0]	RDRSRC0[7:0]	R/W	0x00	SRC0 画面の Red データ濃度調整(256 段階) 0x00: 薄い ~ 0xFF: 濃い

ソース 0(バックグラウンド)画面の濃度を、R、G、B ごとに調整します。

4. LDAFCPSRC1 (LCDDA Replaced Font Area Color pallet of Source1)

Address= (0xF205_0000) + (0x000C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:24]	—	—	Undefined	Read as undefined. Write as zero.
[23:16]	BFONT[7:0]	R/W	0x00	SRC1 画面の FONT エリア色 (Blue データ)
[15:8]	GFONT[7:0]	R/W	0x00	SRC1 画面の FONT エリア色 (Green データ)
[7:0]	RFONT[7:0]	R/W	0x00	SRC1 画面の FONT エリア色 (Red データ)

ソース 1(フォアグラウンド)の FONT 色を、R、G、B ごとに設定します。

フォントドローモード使用時には、<RFONT[7:0]>=<BFONT[7:0]>に設定してください

5. LDAEFCPSRC1 (LCDDA Replaced Except Font Area Color pallet of Source1)

Address= (0xF205_0000) + (0x0010)

Bit	Bit Symbol	Type	Reset Value	Description
[31:24]	—	—	Undefined	Read as undefined. Write as zero.
[23:16]	BEFONT[7:0]	R/W	0x00	SRC1 画面の FONT 以外のエリア色 (Blue データ)
[15:8]	GEFONT[7:0]	R/W	0x00	SRC1 画面の FONT 以外のエリア色 (Green データ)
[7:0]	REFONT[7:0]	R/W	0x00	SRC1 画面の FONT 以外のエリア色 (Red データ)

ソース 1(フォアグラウンド)の FONT 以外の場所の色を、R、G、B ごとに設定します。

6. LDADVSR0 (LCDDA Delta Value (Read Step) address Register of Source 0)

Address = (0xF205_0000) + (0x0038)

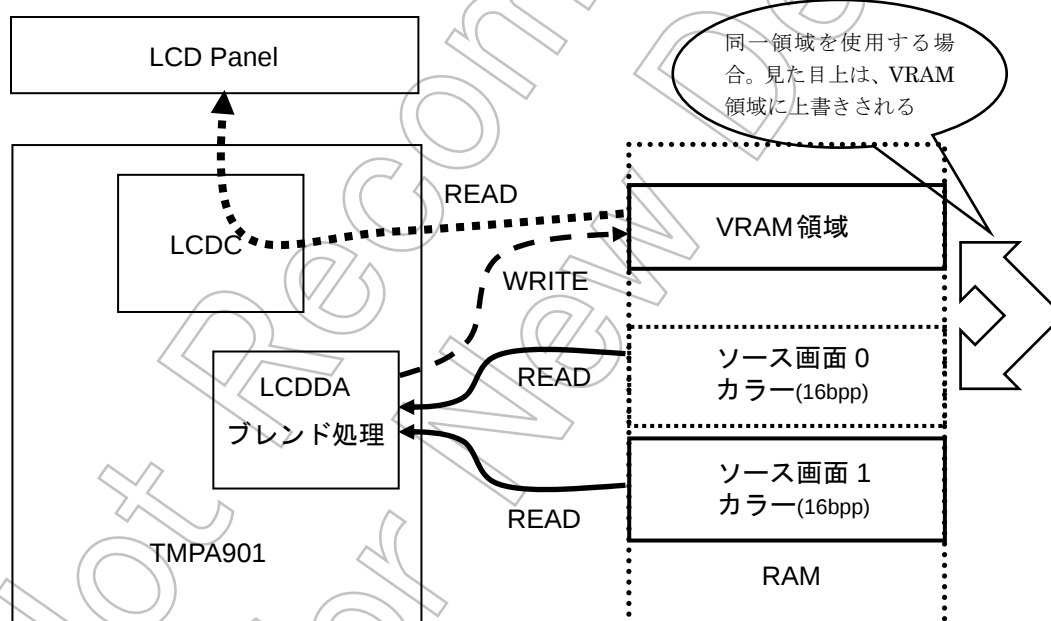
Bit	Bit Symbol	Type	Reset Value	Description
[31]	OVWEN	RW	0y0	0y0: SRC0 のスタートアドレス≠ DST のスタートアドレス 0y1: SRC0 のスタートアドレス = DST のスタートアドレス
[30]	INDSAEN	RW	0y0	0y0: SRC0 の DX,DY= SRC1 の DX,DY 0y1: SRC0 の DX,DY≠SRC1 の DX,DY
[29:19]	–	–	Undefined	Read as undefined. Write as zero.
[18:6]	DYS0[12:0]	R/W	0x000	SRC0 データの次ラインまでの Read Step アドレス
[5:3]	–	–	Undefined	Read as undefined. Write as zero.
[2:0]	DXS0[2:0]	R/W	0y000	SRC0 データの水平方向の Read Step アドレス

(個別説明)

a. <OVWEN>

LCDDA で処理する画像において、ソース 0 画面とディストネーション画面が同一の場合に使用するビットです。“1”を設定することで、ディストネーション画面の先頭アドレスを設定が、ソース 0 画面の先頭アドレスとしても使用されます。

回路上是ブレンドが実行されますが、現在表示中の画面の一部をブレンドして表示を変更する場合に使用します。



b. <INDSAEN>

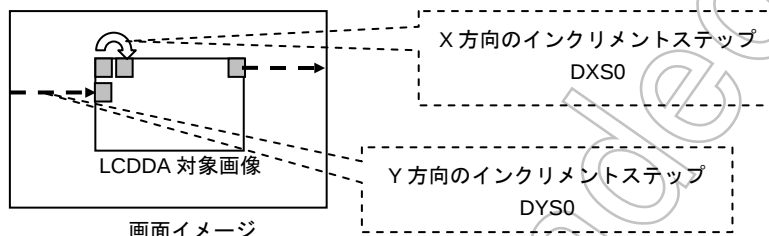
LCDDA で BLEND 処理する画像において、ソース 0 画面とソース 1 画面のインクリメントステップが異なる場合に使用するビットです。“1”を設定することで、ソース 0 画面とソース 1 画面のステップ数を各々において個別に設定することが出来ます。

“0”設定の場合には、ソース 1 画面用に設定されたインクリメントステップが、ソース 0 画面のインクリメントステップに使用されます。

c. <DYS0>

ソース 0 画面のインクリメントステップをソース 1 画面と異なる設定にする際に使用します。INDSAEN ビットに"1"を設定しない場合には、本設定は無効となり、ソース 1 画面のインクリメントステップの値が、ソース 0 画面にも適用されます。

LCDDA が処理をする元画面(ソース画面 0)からデータをリードする際の、垂直方向のインクリメントアドレス(改行時)の Step を設定します。



使用する表示色によって、アドレスのステップ値を計算して設定します。

d. <DXS0[2:0]>

ソース 0 画面のインクリメントステップをソース 1 画面と異なる設定にする際に使用します。INDSAEN ビットに"1"を設定しない場合には、本設定は無効となり、ソース 1 画面のインクリメントステップの値が、ソース 0 画面にも適用されます。

LCDDA が処理をする元画面(ソース画面 0)からデータをリードする際の、水平方向のインクリメントアドレスの Step を設定します。16bpp(64K 色)データの場合は 2 バイトステップですので、"010"と設定し、32bpp(16M 色)データの場合は、"100"と設定します。

7. LDADVSR1 (LCDDA Delta Value (Read Step) address Register of Source 1)

Address = (0xF205_0000) + (0x0014)

Bit	Bit Symbol	Type	Reset Value	Description
[31:24]	OFSETX[7:0]	R/W	0x0000	スケーラ使用時の水平方向サンプリングポイントのオフセット値
[23:18]	–	–	Undefined	Read as undefined. Write as zero.
[17:6]	DYS1[11:0]	R/W	0x000	SRC1 データの次ラインまでの Read Step アドレス
[5:3]	–	–	Undefined	Read as undefined. Write as zero.
[2:0]	DXS1[2:0]	R/W	0y000	SRC1 データの水平方向の Read Step アドレス

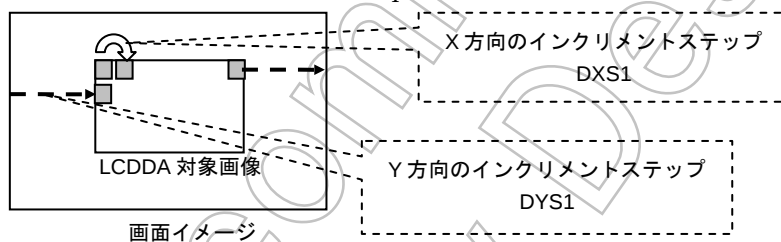
(個別説明)

a. <OFSETX[7:0]>

スケーラ回路における、水平方向画素サンプリングステップのオフセット値を設定します。0x01~0xFF の値を設定し、設定の画素ポイントを最初にサンプリングします。

b. <DYS1[11:0]>

LCDDA が処理をする元画面（ソース画面 1）からデータをリードする際の、垂直方向のインクリメントアドレス（改行時）の Step を設定します。



c. <DXS1[2:0]>

LCDDA が処理をする元画面（ソース画面 1）からデータをリードする際の、水平方向のインクリメントアドレスの Step を設定します。16bpp（64K 色）データの場合は 2 バイトステップですので、“0y010”と設定します。

8. LDACR2 (LCDDA Control Register 2)

Address = (0xF205_0000) + (0x0018)

Bit	Bit Symbol	Type	Reset Value	Description
[31:24]	OFSEY[7:0]	R/W	0x0000	スケーラ使用時の垂直方向サンプリングポイントのオフセット値
[23:16]	–	–	Undefined	Read as undefined. Write as zero.
[15:8]	HCRCT[7:0]	R/W	0x00	水平方向の周期補正值
[7:0]	VCRCT[7:0]	R/W	0x00	垂直方向の周期補正值

(個別説明)

a. <OFSEY[7:0]>

スケーラ回路における、垂直方向画素サンプリングステップのオフセット値を設定します。
0x01~0xFF の値を設定し、設定の画素ポイントを最初にサンプリングします。

b. <HCRCT[7:0]>

スケーラ回路における、水平方向の周期補正の値を設定します。
LDACR0<PCEN>ビットを”1”に設定すると有効になります。0x01~0xFF の値を設定し、
設定の画素ポイントをオリジナル画素 1 列右のポイントに補正します。

c. <VCRCT[7:0]>

スケーラ回路における、垂直方向の周期補正の値を設定します。
LDACR0<PCEN>ビットを”1”に設定すると有効になります。0x01~0xFF の値を設定し、
設定の画素ポイントをオリジナル画素 1 行下のポイントに補正します。

9. LDADX DST (LCDDA X-Delta Value (Write Step) address Register of Destination)

Address = (0xF205_0000) + (0x001C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:28]	XRRATE[3:0]	R/W	0y0000	水平方向縮小率
[27:25]	–	–	Undefined	Read as undefined. Write as zero.
[24]	DXDSIGN	R/W	0y0	DST データの水平方向における進行方向
[23:0]	DXDST[23:0]	R/W	0x000000	DST データの水平方向におけるステップ数

(個別説明)

a. <XRDRATE[3:0]>

水平方向の縮小値を設定します。

0x0~0xfを設定し、設定値+1の値を分母とした縮小を行います。

例)0x2 と設定した場合は、 $1/(2+1)=1/3$ となり水平方向を 1/3 に縮小

b. <DXDSIGN>

水平方向のディストネーションステップの方向を設定します。

Rotation 機能の回転角度、水平／垂直ミラーなどに合わせて、アドレスの進む方向を決めて設定します。

0y0: プラス (インクリメント)

0y1: マイナス (デクリメント)

c. <DXDST[23:0]>

水平方向のディストネーションステップアドレスを設定します。

アドレスを管理し、Rotation 機能を実現するため、ステップアドレスは 24 ビット準備されています。Rotation 機能の回転角度、水平／垂直ミラーなどに合わせて、ステップアドレスを設定します。

10. LDADYDST (LCDDA Y-Delta Value (Write Step) address Register of Destination)

Address = (0xF205_0000) + (0x0020)

Bit	Bit Symbol	Type	Reset Value	Description
[31:28]	YRDRATE[3:0]	R/W	0y0000	垂直方向縮小率
[27:25]	–	–	Undefined	Read as undefined. Write as zero.
[24]	DYDSIGN	R/W	0y0	DST データの垂直方向における Write 方向
[23:0]	DYDST[23:0]	R/W	0x000000	DST データの垂直方向におけるステップ数

(個別説明)

a. <YRRATE[3:0]>

垂直方向の縮小率を設定します。

0x0~0xf を設定し、設定値+1 の値を分母とした縮小を行います。

例)0xf と設定した場合は、 $1/(15+1)=1/16$ となり垂直方向に 1/16 に縮小

b. <DYDSIGN>

垂直方向のディストネーションステップの方向を設定します。

Rotation 機能の回転角度、水平／垂直ミラーなどに合わせて、アドレスの進む方向を決めて設定します。

0y0: プラス (インクリメント)

0y1: マイナス (デクリメント)

c. <DYDST[23:0]>

垂直方向のディストネーションステップアドレスを設定します。

アドレスを管理し、Rotation 機能を実現する為、ステップアドレスは 24 ビット準備されています。Rotation 機能の回転角度、水平／垂直ミラーなどに合わせて、ステップアドレスを設定します。

11. LDASSIZE (LCDDA Source Picture Size)

Address = (0xF205_0000) + (0x0024)

Bit	Bit Symbol	Type	Reset Value	Description
[31:24]	XEXRATE[7:0]	R/W	0x00	スケーラ使用時の水平方向拡大率
[23:22]	–	–	Undefined	Read as undefined. Write as zero.
[21:12]	SYSIZE[9:0]	R/W	0x000	垂直方向の SRC 画像サイズ(Dot 単位設定)
[11:10]	–	–	Undefined	Read as undefined. Write as zero.
[9:0]	SXSIZE[9:0]	R/W	0x000	水平方向の SRC 画像サイズ(Dot 単位設定)

注 1) 全機能において、設定が必要です。

注 2) スケーラ機能で使用する場合の水平方向の画像サイズは最大 511dot です。

(個別説明)

a. <XEXRATE[7:0]>

水平方向の拡大率を設定します。

0x01~0xff を設定し(0x00 設定は禁止です)拡大を行います。

以下の計算式にあわせて値を入力してください。

- X 方向のオリジナル画素数: m ピクセル
- X 方向の拡大後の画素数: n ピクセル

$$\begin{aligned}
 & (\text{補間可能な最大画素数}) \div (\text{拡大後の画素数}) \\
 & = \{(m-1) \times 256\} \div n
 \end{aligned}$$

b. <SYSIZE[9:0]>

垂直方向のソース画像サイズの設定をします。

画像サイズを dot 単位で設定します。入力したサイズ+1 の Dot がサイズ指定されます。

(例):200dot の場合、199(oct)=C7(hex)と設定する。

c. <SXSIZE[9:0]>

水平方向のソース画像サイズの設定をします。

画像サイズを dot 単位で設定します。入力したサイズ+1 の Dot がサイズ指定されます。

(例):200dot の場合、199(oct)=C7(hex)と設定する。

12. LDADSIZE (LCDDA Destination Picture Size)

Address = (0xF205_0000) + (0x0028)

Bit	Bit Symbol	Type	Reset Value	Description
[31:24]	YEXRATE[7:0]	R/W	0x00	スケーラ使用時の垂直方向拡大率
[23:10]	–	–	Undefined	Read as undefined. Write as zero.
[9:0]	DXSIZE[9:0]	R/W	0x000	水平方向の DST 画像サイズ(Dot 単位設定)

注) スケーラ機能で使用する場合、垂直方向の画像サイズの設定は無効です。スケーラ機能は 1 ラインずつ処理されるため、処理の回数がそのまま垂直方向の画像サイズになります。

(個別説明)

a. <YEXRATE[7:0]>

垂直方向の拡大率を設定します。

0x01~0xff を設定し(0x00 設定は禁止です)、拡大を行います。

以下の計算式にあわせて値を入力してください。

- Y 方向のオリジナル画素数: k ピクセル
- Y 方向の拡大後の画素数: 1 ピクセル

$$(\text{補間可能な最大画素数}) \div (\text{拡大後の画素数}) \\ = \{(k - 1) \times 256\} \div 1$$

b. <DXSIZE[9:0]>

水平方向のディストーション画像サイズの設定をします。

画像サイズを dot 単位で設定します。入力したサイズ+1 の Dot がサイズ指定されます。

(例):200dot の場合、199(oct)=C7(hex)と設定する。

13. LDAS0AD (LCDDA Source 0 Start Address)

Address = (0xF205_0000) + (0x002C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	S0ADR[31:0]	R/W	0x00000000	SRC0 画面のスタートアドレス

注) 32bit のアドレスを設定しますが、上位 8bit のアドレスは内部カウンタを持っていません。SRC 画面データ領域で上位 8bit のアドレスが変化するような、スタートアドレスの設定は出来ませんので御注意ください。
(例: 0x00FFFFFF と設定した場合、LCDDA は 0x00FFFFFF→0x00000000→0x00000001 の順序でアクセスします: 上位 8bit のアドレスはインクリメント出来ない)

14. LDADAD (LCDDA Destination Start Address)

Address = (0xF205_0000) + (0x0030)

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	DSADR[31:0]	R/W	0x00000000	DST 画面のスタートアドレス

注) 32bit のアドレスを設定しますが、上位 8bit のアドレスは内部カウンタを持っていません。DST 画面データ領域で上位 8bit のアドレスが変化するような、スタートアドレスの設定は出来ませんので御注意ください。
(例: 0x00FFFFFF と設定した場合、LCDDA は 0x00FFFFFF→0x00000000→0x00000001 の順序でアクセスします: 上位 8bit のアドレスはインクリメント出来ない)

15. LDACR1 (LCDDA Control Register1)

Address = (0xF205_0000) + (0x0034)

Bit	Bit Symbol	Type	Reset Value	Description
[31]	SYNRST	WO	0y0	S/W リセット制御
[30]	LDASTART	WO	0y0	LCDDA の START 制御
[29]	–	–	Undefined	Read as undefined. Write as zero.
[28:24]	OPMODE[4:0]	R/W	0y00000	LCDDA モード設定 0y00000: ノーマルモード 0y00001: スケーラモード 0y00010: モノクロモード 0y00110: モノクロインバートモード 0y10000: ブレンドモード 0y10010: モノクロブレンドモード 0y10110: モノクロインバートブレンドモード 注) 上記以外のビット(機能)の組み合わせの設定は出来ません。
[23:0]	S1ADR[23:0]	R/W	0x000000	SRC1 画面の先頭アドレス(32bit 中、下位 24bit)

(個別説明)

a. <SYNRST>

S/W リセットを制御します。

0y1: リセット

0y0: 無効

b. <LDASTART>

LCDDA の START 制御

0y1: LCDDA スタート

0y0: 無効

c. <OPMODE[4:0]>

LCDDA の動作モードを選択します。

0y00000: ノーマルモード

単純なデータ転送モードです。ソースデータは、ソース 1(S1)のみが選択され (S0 の設定は無効)、矩形の画像を転送 (回転/縮小) する際に使用します。

0y00001: スケーラモード

BC-Expander 回路の動作を制御し、スケーラモードになります。
BLEND 機能、FONT 機能との同時使用は出来ません。

0y00010: モノクロモード

モノクロのソースを転送するモードです。ソースデータは、ソース 1(S1)のみが選択され (S0 の設定は無効)、モノクロデータをカラーに変換して画像を転送 (回転/縮小) する際に使用します。

0y00110: モノクロインバートモード

モノクロのソースのデータを反転して転送するモードです。ソースデータは、ソース 1(S1)のみが選択され(S0 の設定は無効)、モノクロデータを反転し、カラーに変換して画像を転送 (回転/縮小) する際に使用します。

0y10000: ブレンドモード

カラー (16bpp) の 2 枚の画面をブレンドするモードです。ソースデータは、ソース 0(S0)とソース 1(S1)が選択され、画像をブレンドした後、転送 (回転/縮小) する際に使用します。

- 0y10010: モノクロ ブレンドモード
モノクロのソースとカラー(16bpp)の 2 枚の画面をブレンドするモードです。モノクロで指定されたソース 1(S1)画像をカラー変換し、カラーで指定されたソース 0(S0)の 2 枚の画面をブレンドした後転送 (回転/縮小) する際に使用します。
- 0y10110: モノクロインバート ブレンドモード
モノクロのソースを反転データと、カラー(16bpp)の 2 枚の画面をブレンドするモードです。モノクロで指定されたソース 1(S1)画像をカラー変換し、カラーで指定されたソース 0(S0)の 2 枚の画面をブレンドした後転送 (回転/縮小) する際に使用します。
- 0y11010: フォントドローモード
モノクロのソースとカラー(16bpp)の 2 枚の画面をブレンドするモードです。モノクロで指定されたソース 1(S1)画像をカラー変換し、カラーで指定されたソース 0(S0)の 2 枚の画面をブレンドします。モノクロデータは"1"データのみが利用され、拡張して画像を転送(回転/縮小)する際に使用します。

d. <S1ADR[23:0]>

ソース 1 画面のスタートアドレスの下位 24 ビットの設定をします。

ソース 1 画面のアドレスは 32 ビット設定が必要ですが、上位 8 ビットのアドレスは LDACR0<S1ADR[31:24]>で設定します。

3.20 タッチスクリーンインタフェース (TSI)

4 端子型抵抗網タッチスクリーンインタフェースを内蔵しています。TSI は、タッチ検出および X/Y 位置測定 の 2 つの動作を容易に実現できます。TSI 制御レジスタ(TSICR0,TSICR1)および内蔵 AD コンバータを使用して実行します。

3.20.1 TSIの外部接続図、内部ブロック図

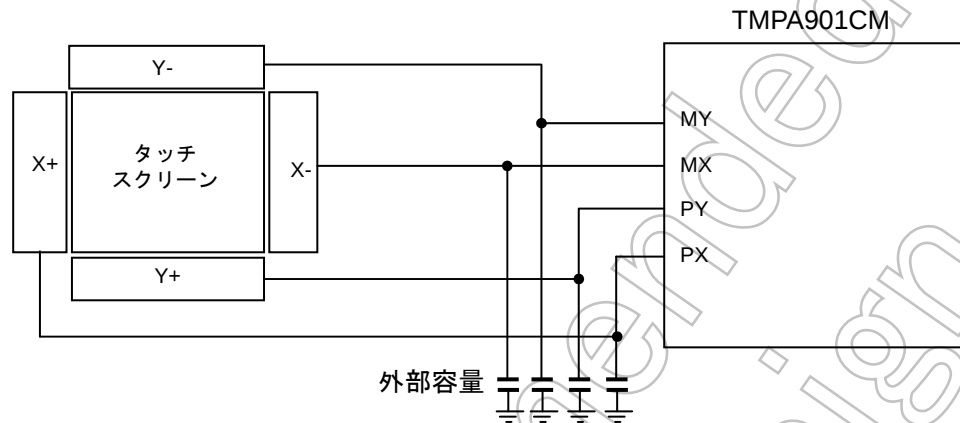


図 3.20.1 TSIの外部接続図

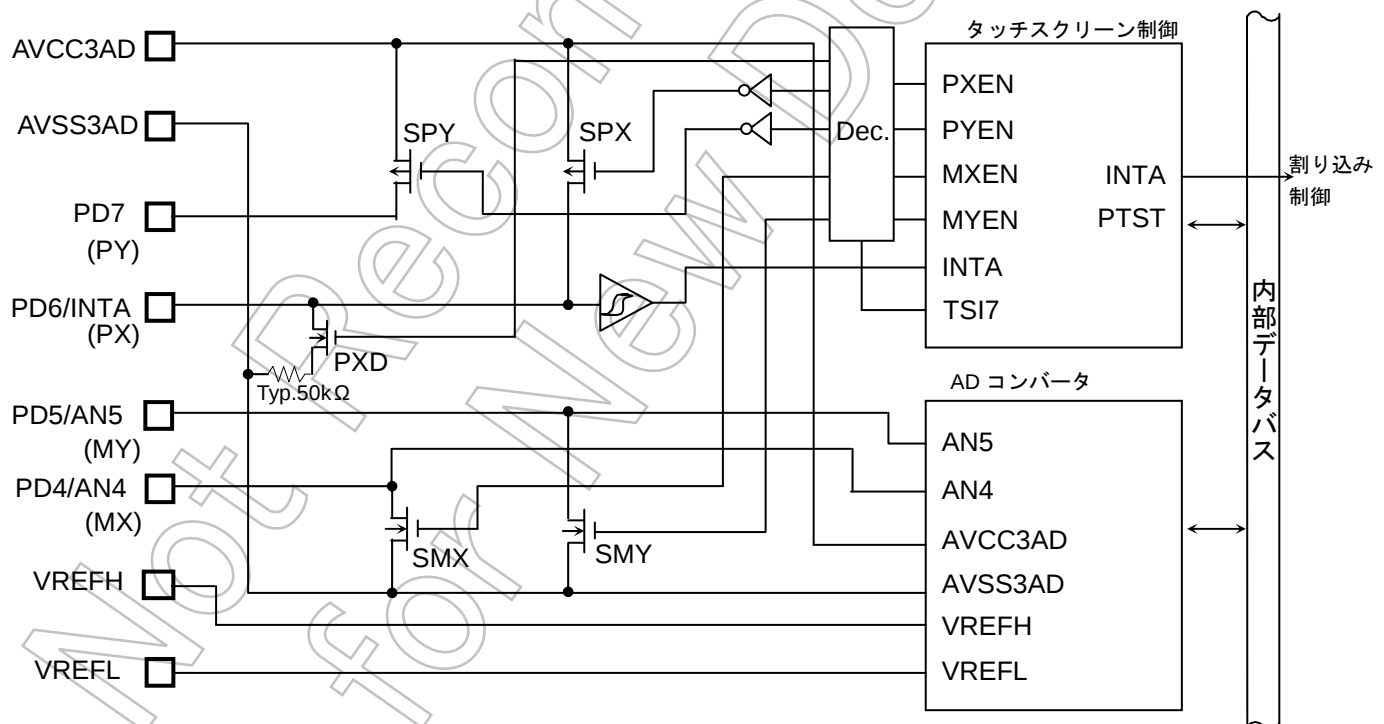


図 3.20.2 TSIの内部ブロック図

3.20.2 SFR

SFR のリストを以下に示します。

base アドレス= 0xF006_0000

Register Name	Address (base+)	Description
TSICR0	0x01F0	TSI Control Register0
TSICR1	0x01F4	TSI Control Register1

Not Recommended
for New Design

1. TSICR0 (TSI Control Register0)

Address = (0xF006_0000) + (0x01F0)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	—	—	Undefined	Read as undefined. Write as zero.
[7]	TSI7	R/W	0y0	Pull-down 抵抗設定(個別説明を参照) : 0y0: 禁止 0y1: 許可
[6]	INGE	R/W	0y0	ポート PD6、PD7 の入力ゲート制御 0y0: 許可 0y1: 禁止
[5]	PTST	RO	0y0	検出状態 リード時: 0y0: 検出無 0y1: 検出中 ライト時: 無効
[4]	TWIEN	R/W	0y0	INTA 割り込み制御 0y0: 禁止 0y1: 許可
[3]	PYEN	R/W	0y0	SPY 0y0: OFF 0y1: ON
[2]	PXEN	R/W	0y0	SPX 0y0: OFF 0y1: ON
[1]	MYEN	R/W	0y0	SMY 0y0: OFF 0y1: ON
[0]	MXEN	R/W	0y0	SMX 0y0: OFF 0y1: ON

注) AD コンバータでアナログ入力データを変換中に通常の C-MOS 入力ゲートに貫通電流が流れることを防ぐために、TSICR0<INGE>の制御をすることができます。中間電圧が入力される場合には本 bit で PD6、PD7 の C-MOS ロジックへの入力信号を遮断してください。TSICR0<PTST>は初期のペンタッチを確認するものです。TSICR0<INGE>にて C-MOS ロジックへの入力が遮断すると、本 bit は常に 1 になりますので注意してください。

(個別説明)

a. <TSI7>

PXD(内部プルダウン抵抗)ON/OFF 設定

<PXEN> <TSI7>	0	1
0	OFF	OFF
1	ON	OFF

2. TSICR1 (TSI Control Register1)

Address = (0xF006_0000) + (0x01F4)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7]	DBC7	R/W	0y0	0y0: 禁止 0y1: 許可
[6]	DB1024	R/W	0y0	デバウンス時間は“(N×64-16)/f _{PCLK} ”の式により設定されます。 “N”はビット 6 からビット 0 に 1 を設定した数の総計を表します。(注 2)
[5]	DB256	R/W	0y0	
[4]	DB64	R/W	0y0	
[3]	DB8	R/W	0y0	
[2]	DB4	R/W	0y0	
[1]	DB2	R/W	0y0	
[0]	DB1	R/W	0y0	1

注1) 実際には PD6/INTA 信号は、デバウンス時間をカウントするカウンタ回路への入力する前に、数発の f_{PCLK} で同期化されているため、上記時間 + α の時間の必要になります。

注2) 例えば、(TSICR1)=0x95 に設定した場合、N = 64 + 4 + 1 = 69 となり、f_{PCLK} = 100MHz の場合デバウンス時間は 44μs + α になります。

3.20.3 タッチ検出手順

タッチ検出手順は、タッチスクリーンにペンがタッチされ、検出されるまでの手順です。

タッチされると割込みINTAを発生して本手順は終了します。INTA割り込みルーチンでX/Y位置測定し、X/Y位置測定手順終了後、再び本手順に戻し次のタッチ待ち状態として下さい。

非接触のタッチ待ち状態時は、SPY スイッチだけを ON させ、他 3 つのスイッチ:SMY, SPX,SMX はすべて OFF させて下さい。またこのとき、PD6/INTA/PX 端子に内蔵するプルダウン抵抗は ON させて下さい。

この状態では タッチスクリーン内の X 方向と Y 方向の内部抵抗は接続されていないので PD6/INTA/PX 端子は、内部プルダウン抵抗(PXD)により Low 状態となり、INTA 割込みは発生しません。

次に、ペンがタッチされるとタッチスクリーン内の X 方向と Y 方向の内部抵抗が接続され PD6/INTA/PX 端子は High 状態となり、INTA 割込みを発生します。

1 回のペンタッチにより複数回の INTA 発生を防止するため、下記図のようなデバウンス回路があります。TSICR1 レジスタにデバウンス時間を設定することによりその時間以下のパルスを無視します。

デバウンス回路は、信号の立ち上がりを検出し、設定されたデバウンスカウンタ時間をカウントアップし、カウント後内部に信号を取り込みます。カウント中に信号が“L”になるとカウンタをクリアし、再度立ち上がりエッジ待ち状態になります。

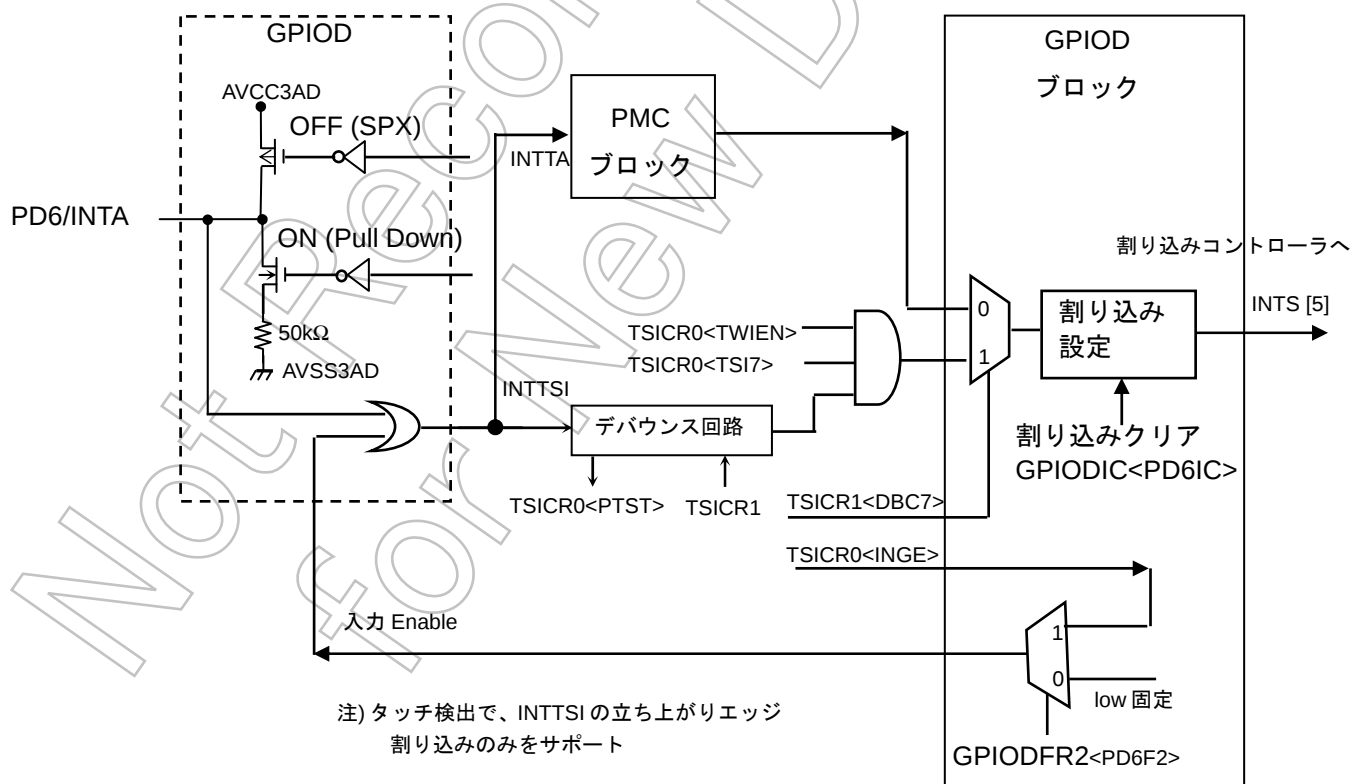


図 3.20.3 タッチ検出の概略図

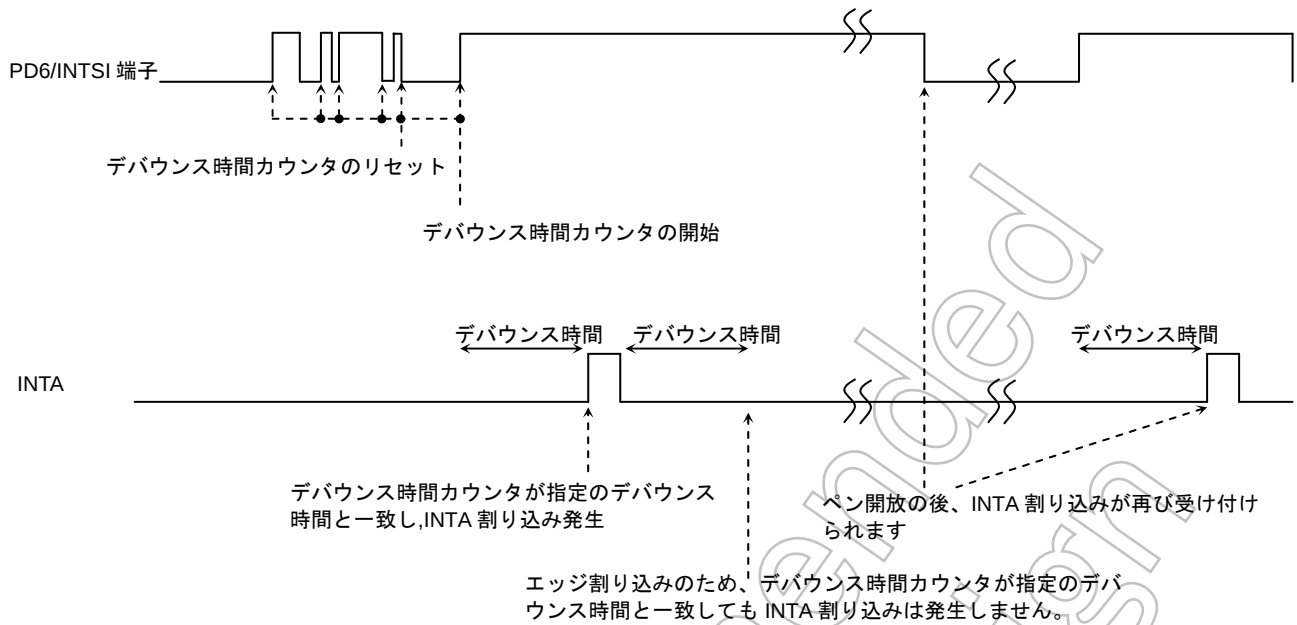


図 3.20.4 デバウンス回路のタイミング図

3.20.4 X/Y 位置測定手順

ペンがタッチされ INTA 割込み発生により、ペンの位置の測定を下記手順で実行してください。

< X 位置座標測定 >

まず SPX と SMX スイッチを ON、SPY と SMY スイッチを OFF させます。これにより、PD5/MY/AN5 端子に X 位置を示すアナログ電圧が入力されます。この電圧を AD コンバータでデジタルコードに変換させることにより X 位置座標を測定できます。

< Y 位置座標測定 >

まず SPY と SMY スイッチを ON、SPX と SMX スイッチを OFF させます。これにより、PD4/MX/AN4 端子に Y 位置を示すアナログ電圧が入力されます。この電圧を AD コンバータでデジタルコードに変換させることにより Y 位置座標を測定できます。

上記X,Y位置測定時のAN5,AN4 端子へ入力されるアナログ電圧は図 3.20.5 に示すような MCU 内部のスイッチの ON 抵抗値とタッチスクリーン内部の抵抗の比で求められます。

したがって、タッチスクリーンの端をタッチした場合でもアナログ入力電圧は 3.3V もしくは 0V にはなりません。

また、各々の抵抗値はばらつきがありますので、これらの点を考慮の上設計してください。なお、AD 変換は必要に応じて数回実行して平均値を最終値とするなどして下さい。

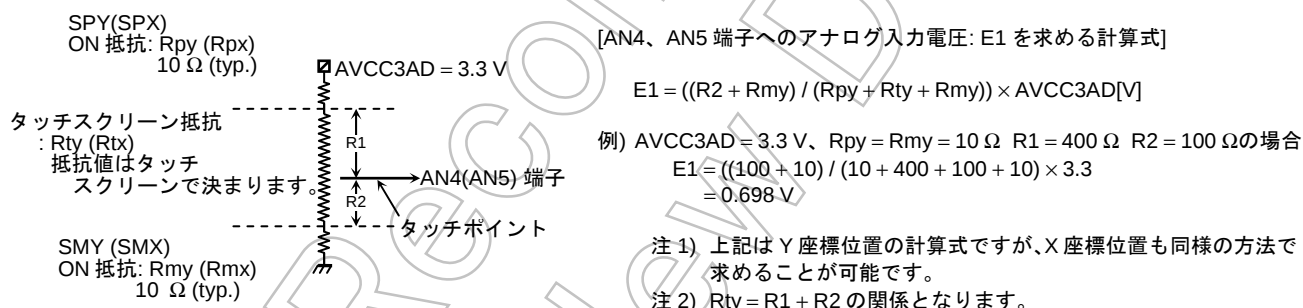


図 3.20.5 アナログ入力電圧算出値

3.20.5 タッチスクリーンインタフェース(TSI)のフローチャート

(1) タッチ検出手順

(2) X/Y 位置測定手順

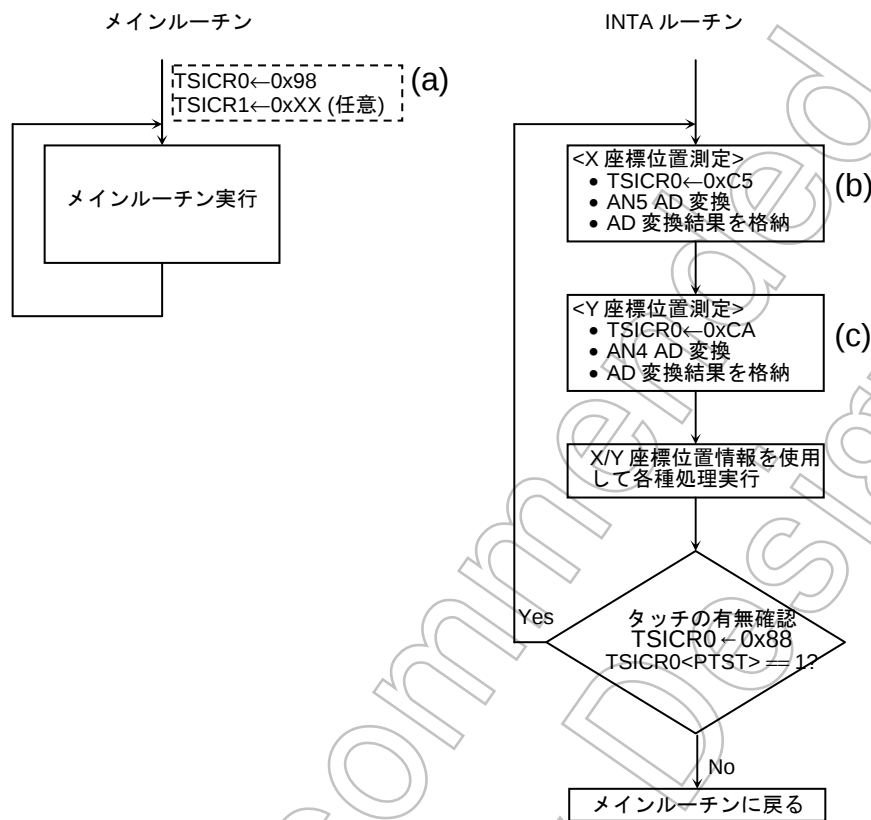
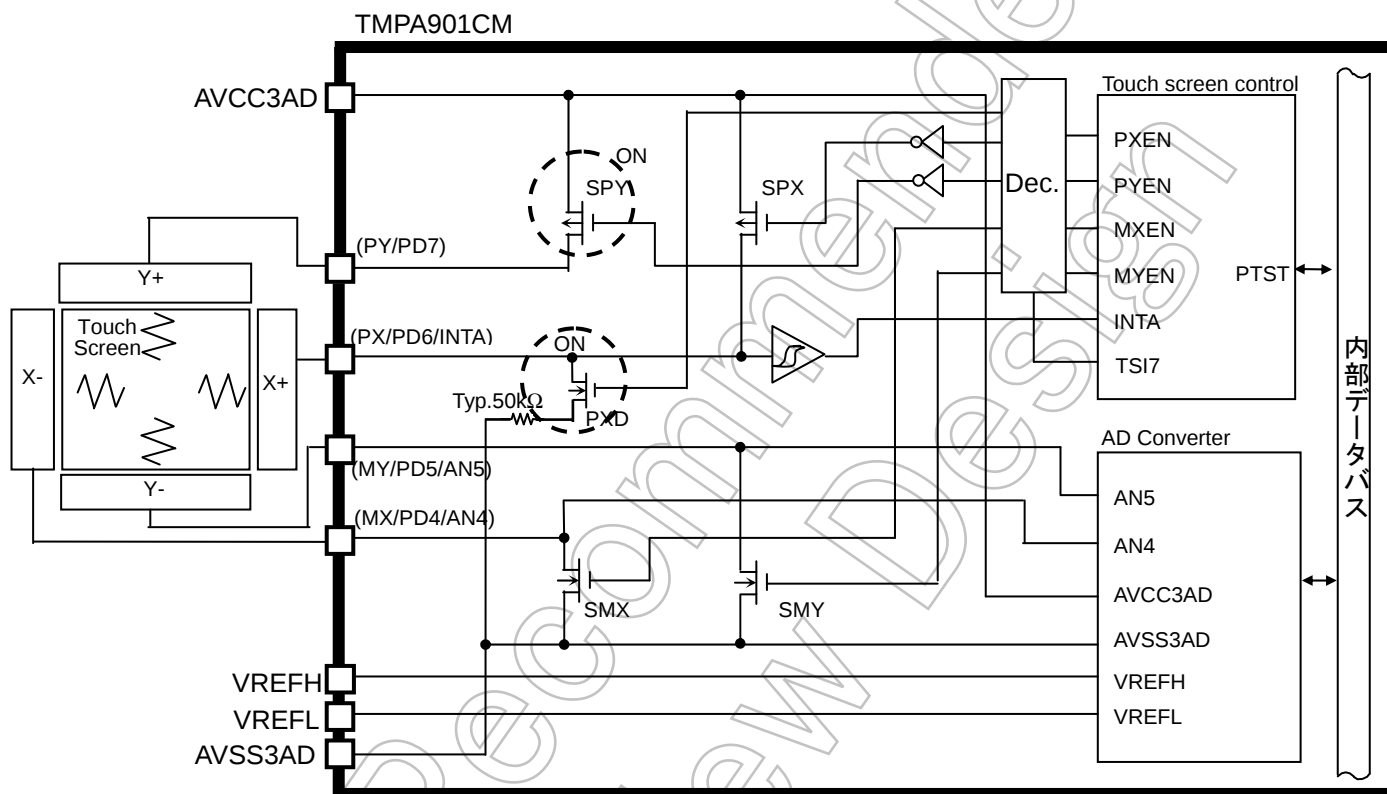


図 3.20.6 TSI 関係のフロー例

次ページに、フロー内の(a)(b)(c)それぞれの回路状態を説明します。

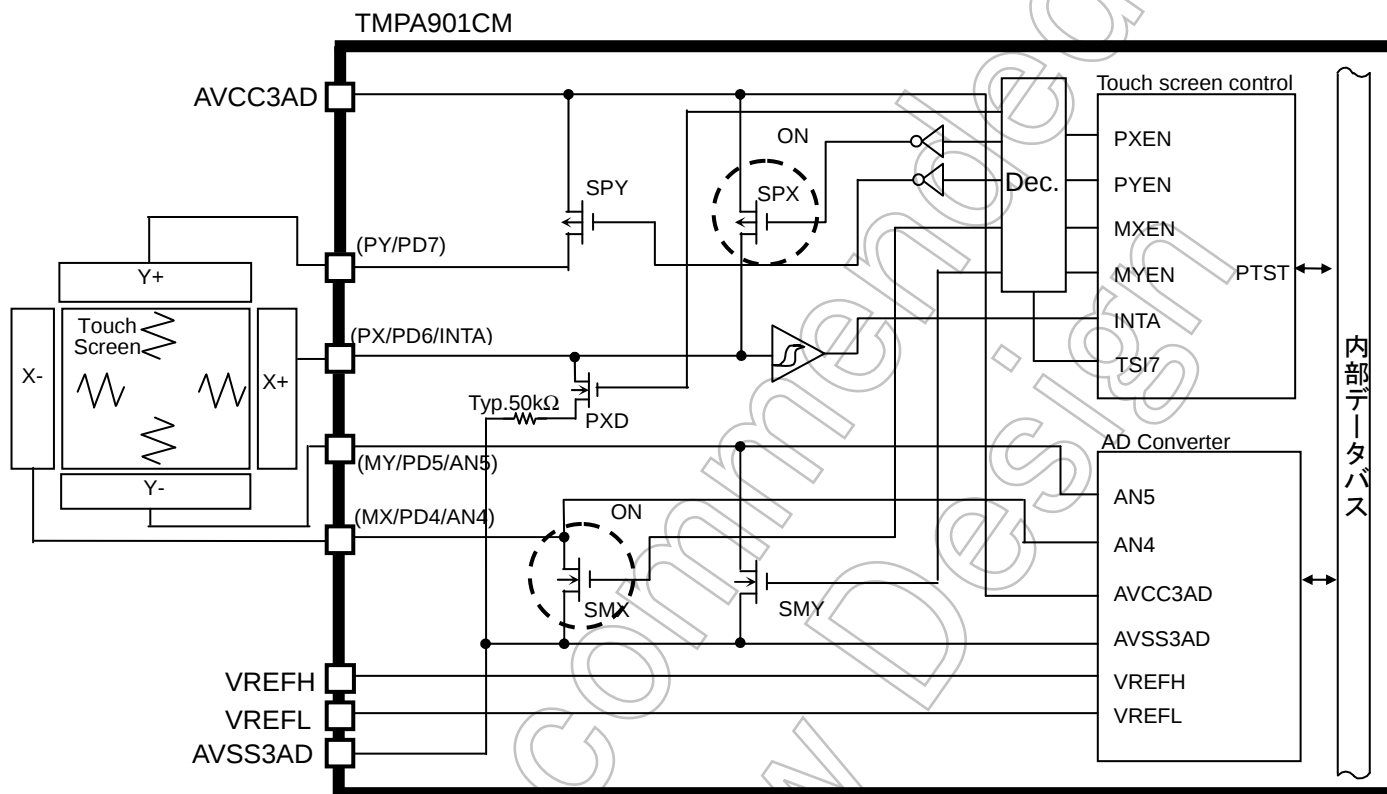
(a) メインルーチン: INTA 割り込み待ち状態

GPIODFR1	←	0x00000030	;PD[6] INTA,PD[5]AN5 , PD[4]AN4
GPIODFR2	←	0x000000C0	;PD[7]PY , PD[6] PX
GPIODIS	←	0x00000000	;set INTA edge interrupt
GPIODIBE	←	0x00000000	;set INTA single edge
GPIODIE	←	0x00000040	;enable INTA
GPIODIEV	←	0x00000040	;set INTA positive edge
TSICR0	←	0x00000098	;[7] enable TSI / PXD ON
			;[4] INTA enable
			;[3] SPY:ON



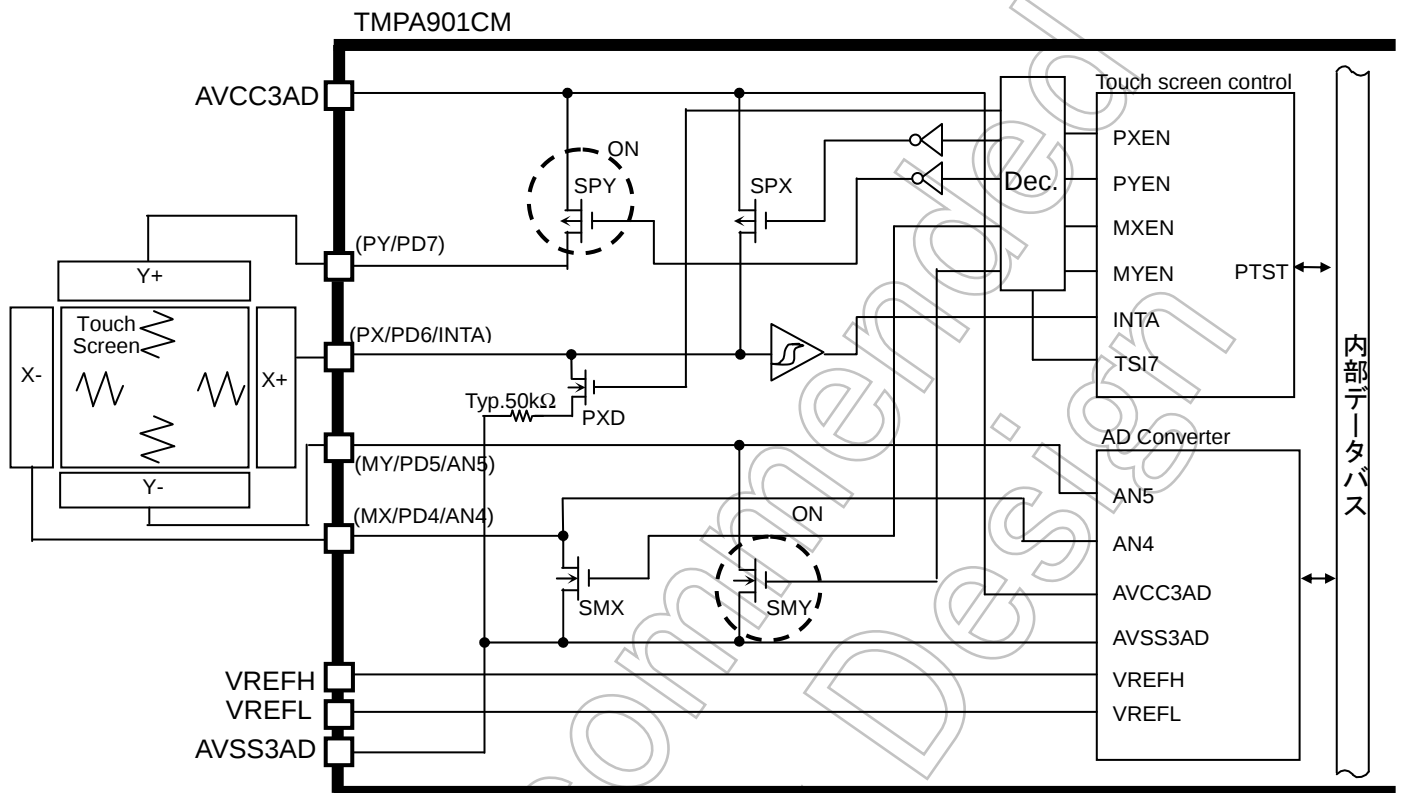
(b) INTA ルーチン: "X" 位置座標測定(AD 変換スタート)

GPIODIC	← 0x00000040	;INT request clear
TSICR0	← 0x00000085	;Disable INTA
TSICR0	← 0x000000C5	;[7] enable TSI
		;[6] PD6/PD7 : disable input
		;[2] SPX:ON , [0] SMX:ON
ADMOD1	← 0x00000085	;Set AN5
ADMOD0	← 0x00000001	;Start AD conversion



(c) INT4 ルーチン: "Y" 位置座標測定 (AD 変換スタート)

TSICR0	← 0x000000CA	;[7] enable TSI ;[6] PD6/PD7 : disable input ;[3] SPY:ON , [1] SMY:ON
ADMOD1	← 0x00000084	;set AN4
ADMOD0	← 0x00000001	;start AD conversion



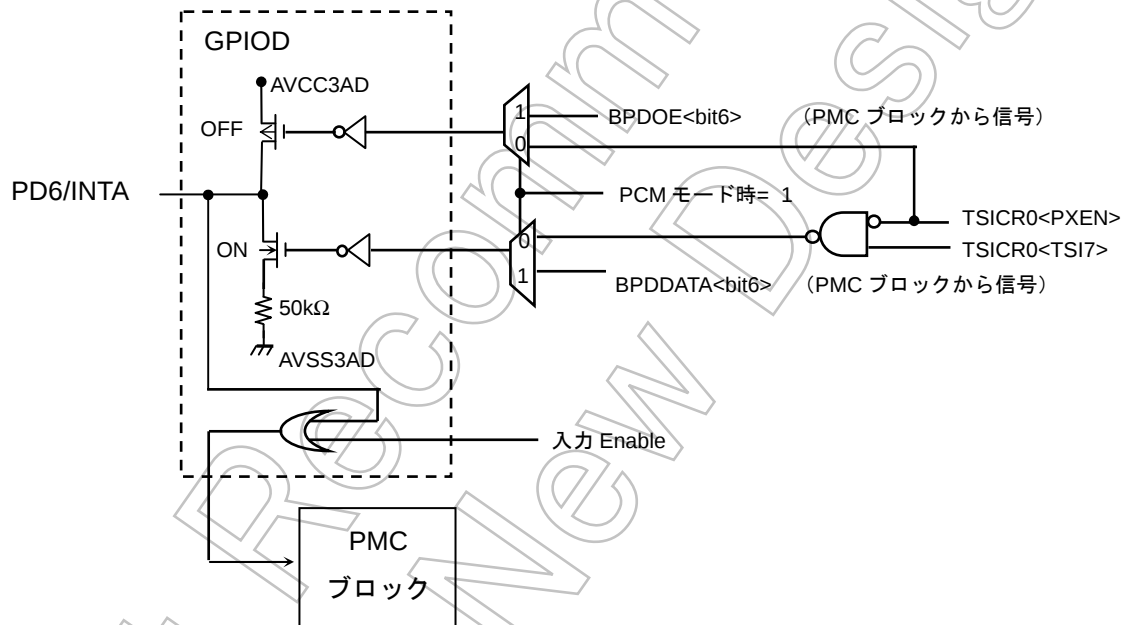
3.20.6 使用上の注意

1. PCM 状態からの復帰について

PCM 状態では TSI 回路は電源が OFF されますが、PD6/INTA 端子からの入力 が直接 PMC へ入力されているため復帰可能です。

設定例：

BPDRINT	←	0x00000000	; write 0x00000000 to Register ; INT status initial
BPARINT	←	0x00000000	; INT status initial
BPPRINT	←	0x00000000	; INT status initial
BSMRINT	←	0x00000000	; INT status initial
BRTRINT	←	0x00000000	; INT status initial
BPDEEDGE	←	0x00000000	; Rising edge
BPDOE	←	0x000000B0	; PD7(SPY):ON ,PD6(SPX):OFF ; PD5(SMY):OFF , PD4(SMX):OFF
BPDDATA	←	0x00000000	; PD6(PXD):ON (pull down)
BPDRELE	←	0x00000040	; PCM release Enable by INTA
...	←	...	; and other setting before enter PCM ; refer to PMC chapter



注) INTA で PCM 状態を wake up させる場合、立ち上がり/立ち下がり両エッジをサポートしていますが、TSI を併用する場合は立ち上がりエッジを使用することをお勧め致します。

2. ポート処理

0V~AVCC3AD の中間電圧を A/D コンバータで変換中に、回路構成上、通常の C-MOS 入力ゲートにも中間電圧が印加されてしまいます。

TSICR0<INGE>を利用して PD6、PD7 の貫通電流対策をしてください。また、この時 (TSICR0<INGE>= 1)、C-MOS ロジックへの入力遮断すると、初期のペンタッチの確認を行う TSICR0<PTST>は、常に 1 になりますので注意してください。

3.21 リアルタイムクロック・メロディアラームジェネレータ

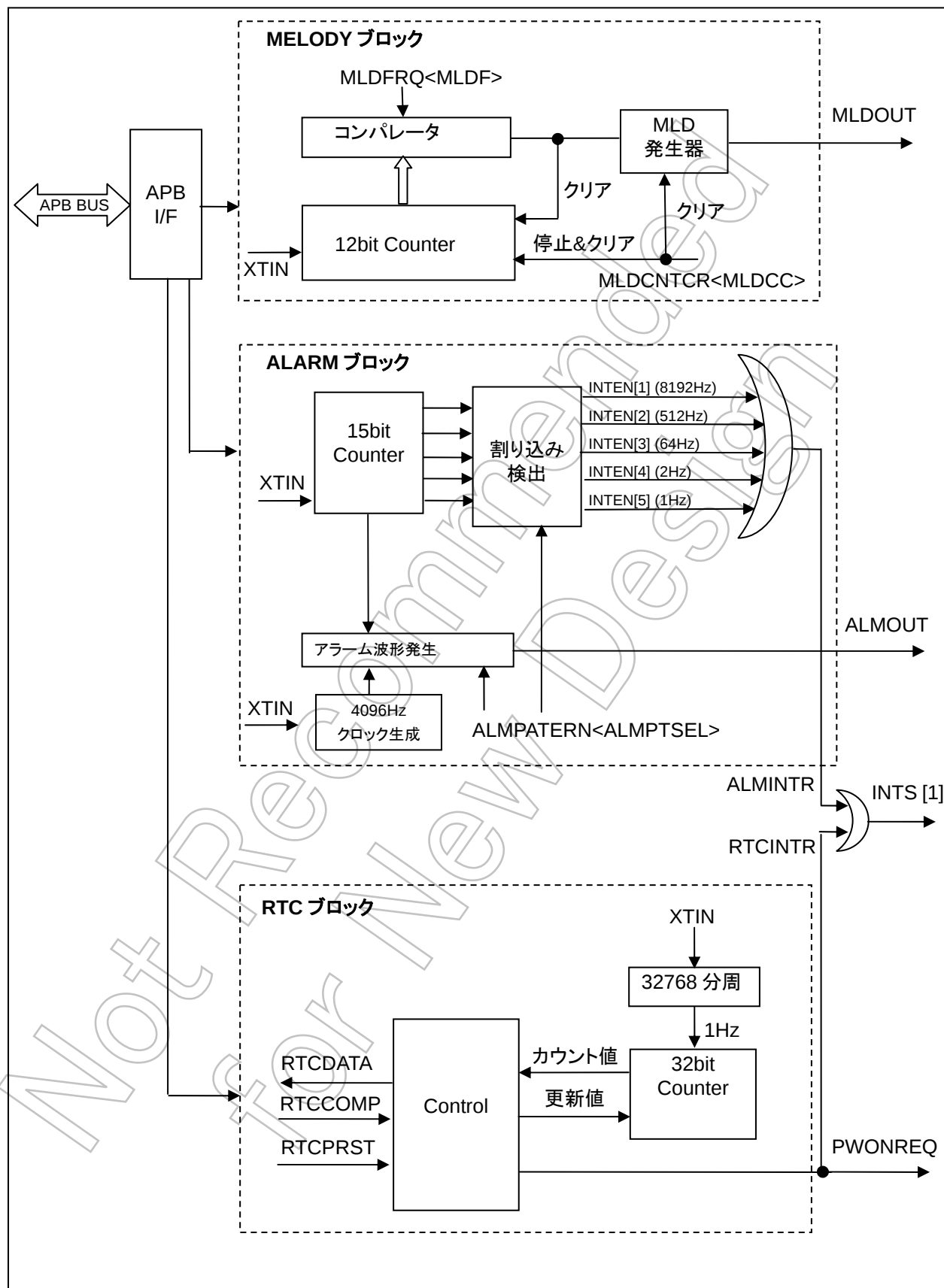
3.21.1 機能概略

32kHz の低周波クロックをベースに、時計機能を実現する回路です。
同一回路内に、メロディを作成するメロディ生成回路および、アラーム生成回路が構成されています。

以下に各々の回路の概要機能を示します。

- 1) メロディ:
 - ・ 4Hz ~ 5461Hz 任意周波数のメロディ波形を生成
- 2) アラーム:
 - ・ 8 種類アラームパターンの生成
 - ・ 5 種類(1/2/64/512/8192 Hz)の周期割り込みを発生
- 3) RTC :
 - ・ 一秒毎カウンタの 32bit カウンタ
 - ・ 32bit カウンタ比較一致の割り込みの要求
(電源管理回路 (PMC)に対する電源復帰の要求に対応)

3.21.2 ブロック図



3.21.3 動作説明

3.21.3.1 メロディジェネレータ

(1) メロディ動作概要

低速クロック 32.768kHz を元に、4Hz~5461Hz のクロック波形を生成し、MLDALM 端子から出力します。

外部に圧電ブザー等を接続することにより容易にメロディ音を鳴らすことが可能です。
メロディ出力波形の周波数は、以下のように計算します。

$$XTIN = 32.768 \text{ [kHz]}$$

$$\text{メロディ出力波形 } f_{\text{MLD}}[\text{Hz}] = 32768 / (2N + 4)$$

$$\text{メロディ設定値 } N = (16384 / f_{\text{MLD}}) - 2$$

(注: レジスタ MLDFRQ<MLDF>の設定値 $N = 1 \sim 4095$ (0x001~0xFFFF)。

"0"は設定禁止

上記式については下記の波形図参照。

(参考: 基本音階別設定値表)

音階	周波数 [Hz]	MLDFRQ<MLDF> レジスタ値: N
C	264	0x03C
D	297	0x035
E	330	0x030
F	352	0x02D
G	396	0x027
A	440	0x023
B	495	0x01F
C	528	0x01D

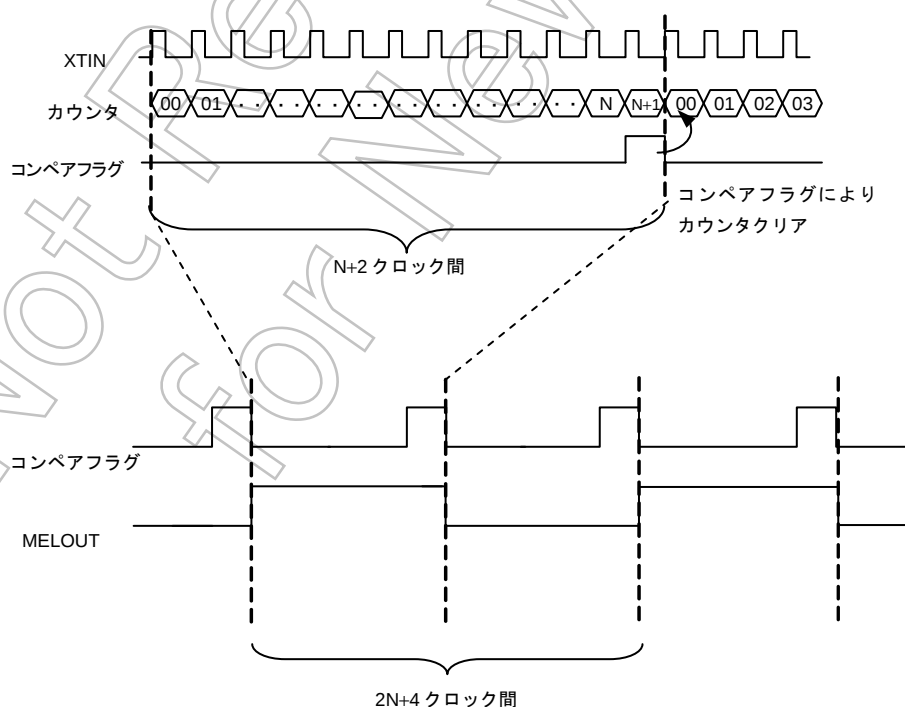
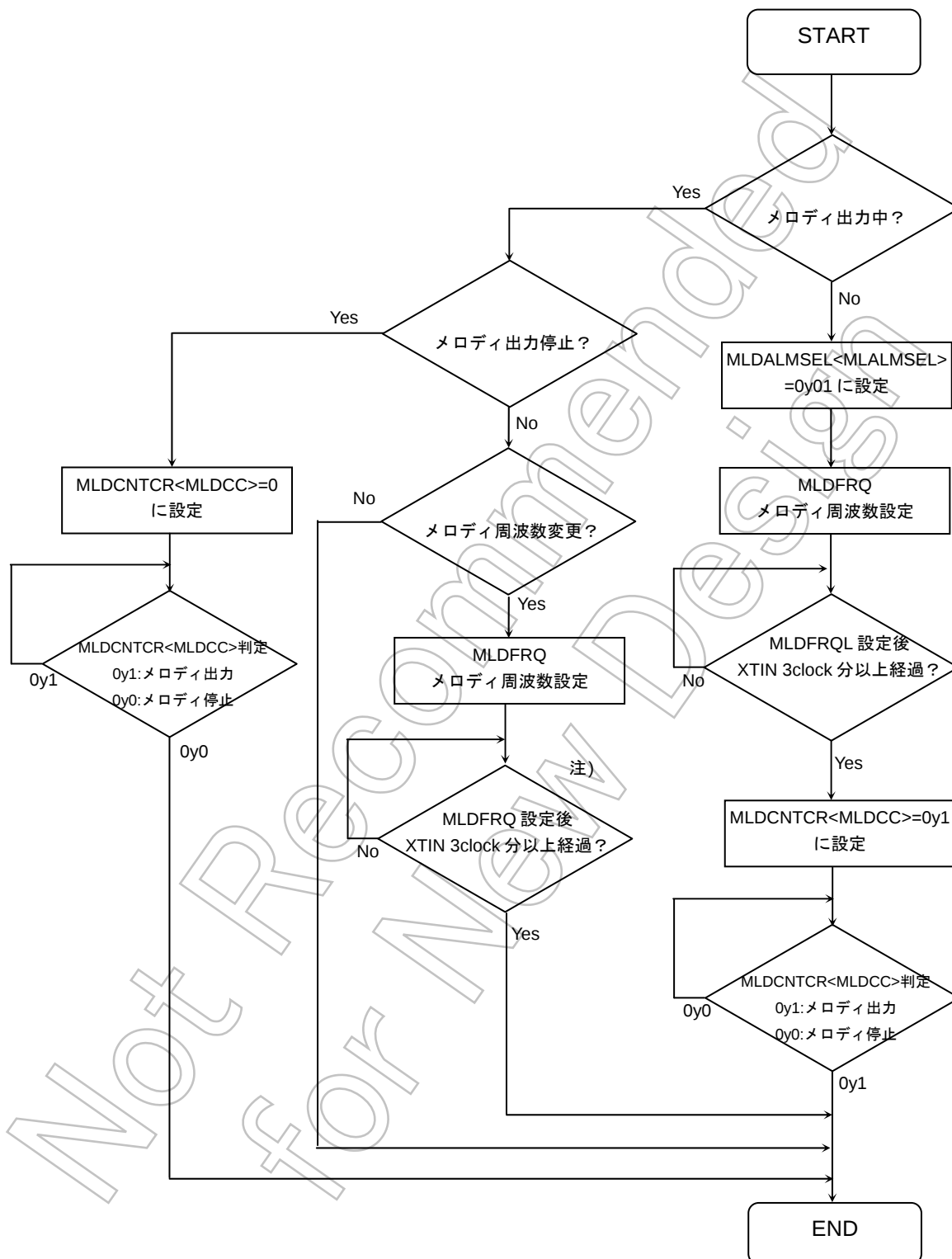


図 3.21.1 メロディ波形

(2) メロディ設定フロー例



注) 上記フローチャート中の MLDFRQ はリード出来ないため、データライト後 32kHz の 3 クロック (約 93μs) 分の時間が経過後、次の処理へ移行して下さい。それ以外のレジスタは、ライト動作後、データをポーリングして更新を確認してから次の処理に移行して下さい。

3.21.3.2 アラームジェネレータ

(1) アラームジェネレータ動作概要

低速クロック 32.768 kHz を元に分周された周波数 4096 Hz から、8 種類のアラーム波形を生成し MLDALM 端子より出力します。

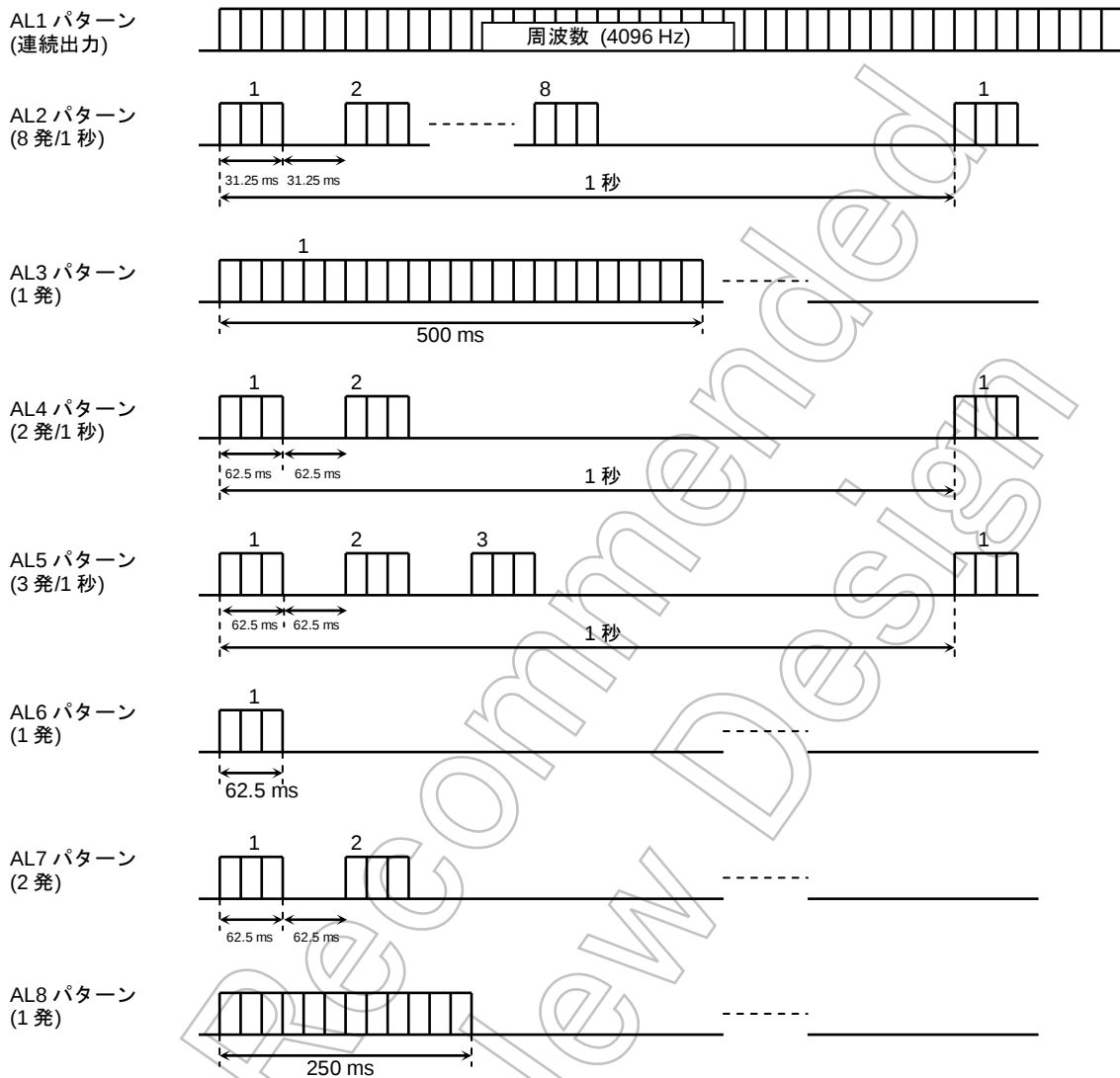
外部に圧電ブザー等を接続することにより容易に電子アラーム音を鳴らすことが可能です。

また、アラームジェネレータに使用されるフリーランカウンタを使用し 5 種類 (1 Hz, 2 Hz, 64 Hz, 512 Hz, 8192Hz) の周期割り込みを発生させることも可能です。

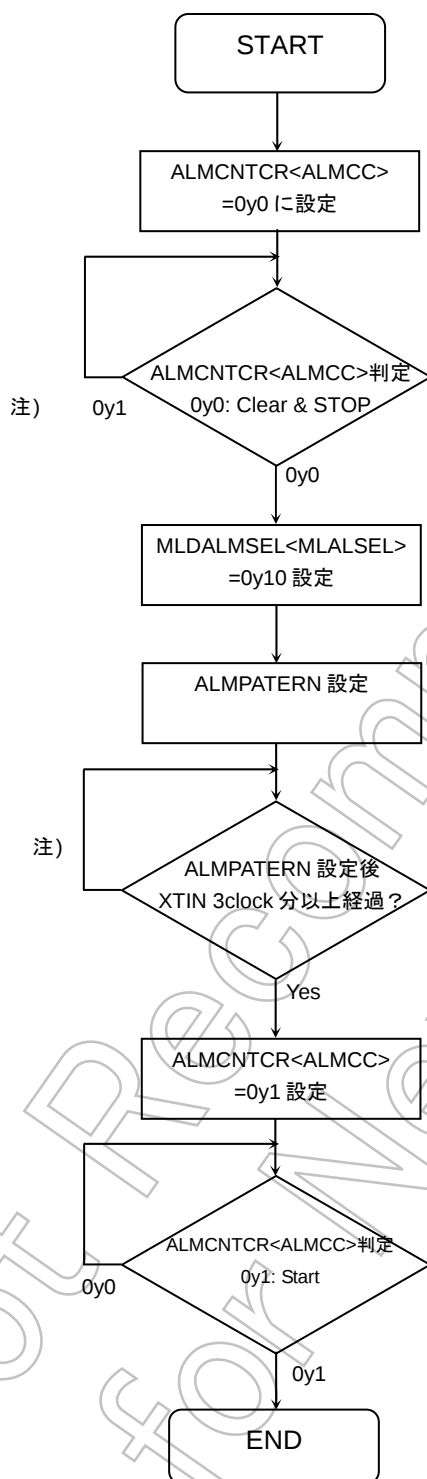
(アラームパターンの設定値)

ALM レジスタ の設定値	アラーム波形
0x00	"0" 固定
0x01	AL1 パターン
0x02	AL2 パターン
0x04	AL3 パターン
0x08	AL4 パターン
0x10	AL5 パターン
0x20	AL6 パターン
0x40	AL7 パターン
0x80	AL8 パターン
その他	未定義 (設定しないでください。)

例: 各設定値のアラームパターンの波形:



(2) アラームジェネレータ 設定フロー例



注) 上記フローチャート中のレジスタ ALMPATTERN は、リード出来ないため、データライト後 32kHz の 3 クロック (約 93μs) 分の時間が経過後、次の処理へ移行して下さい。それ以外のレジスタはライト動作後、データをポーリングして更新を確認してから次の処理に移行して下さい。

3.21.4 Real Time Clock

(1) 動作概要

低速クロック 32.768kHz を元に分周された 1Hz のソースクロックで動作する、32bit カウンタです。

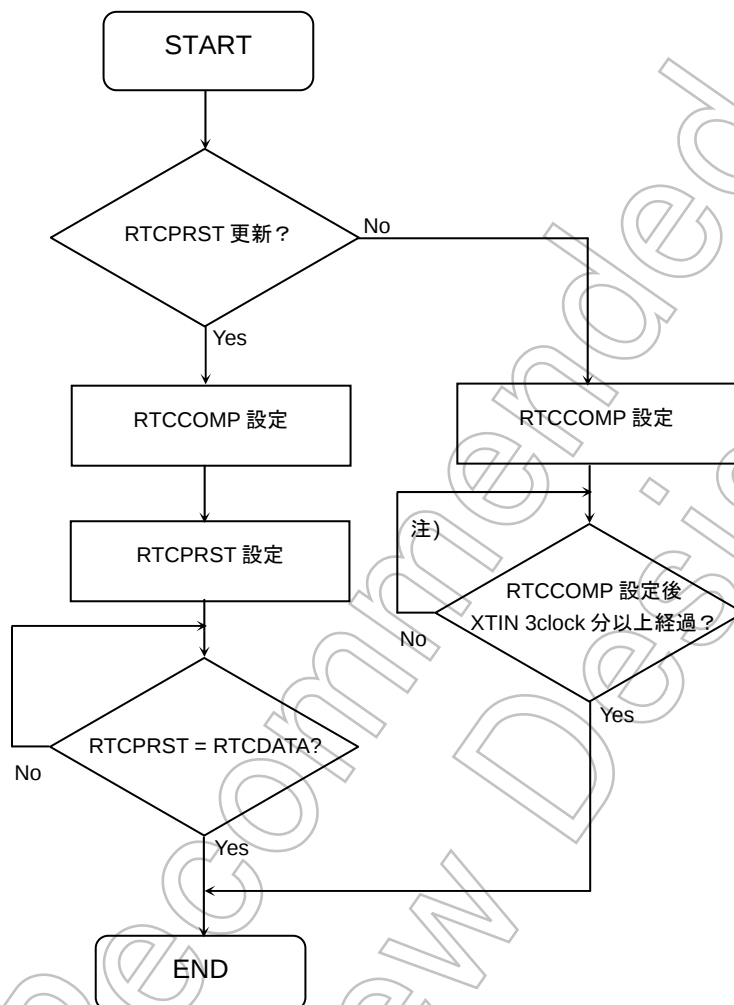
現在のカウンタ値を RTCDATA レジスタからリードすることで、時計機能を容易に実現することができます。

RTCCOMP レジスタに設定された値とカウンタ値を比較し、割り込み信号を発生します (PCM(Power Cut Mode)状態からの復帰要求信号として利用可能です)。

カウンタの値は RTCPRST レジスタに値を設定することで、変更が可能です。

本回路は、PCM 状態でも常に電源が供給される DVCC1B 系の電源で動作しており、常に時間をカウントし続けます。そのため、外部に PCM 状態から復帰させる要因が無い場合でも、本マイコン単体で電源復帰要求を出し、PCM 状態から復帰することが可能です。

(2) Real Time Clock 設定フロー例



注) データライト後 32kHz の 3 クロック (約 93 μ s) 分の時間が経過後、次の処理へ移行して下さい。

3.21.5 SFR

SFR のリストを以下に示します。

base アドレス= 0xF003_0000

Register Name	Address (base+)	Description
RTCDATA	0x0000	RTC Data Register
RTCCOMP	0x0004	RTC Compare Register
RTCPRST	0x0008	RTC Preset Register
MLDALMINV	0x0100	Melody Alarm Invert Register
MLDALMSEL	0x0104	Melody Alarm signal Select Register
ALMCNTCR	0x0108	Alarm Counter Control Register
ALMPATTERN	0x010C	Alarm Pattern Register
MLDCNTCR	0x0110	Melody Counter Control Register
MLDFRQ	0x0114	Melody Frequency Register
RTCALMINTCTR	0x0200	RTC ALM Interrupt Control Register
RTCALMMIS	0x0204	RTC ALM Interrupt Status Register

1. RTCDATA (RTC Data Register)

Address = (0xF003_0000) + 0x0000

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	RTCCOUNT	RO	Undefined	32 ビットカウンタ値

(個別説明)

a. <RTCCOUNT>

読み出すと、RTC(32 ビット)カウンタ値が返ってきます。

2. RTCCOMP (RTC Compare Register)

Address = (0xF003_0000) + 0x0004

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	RTCCP	WO	Undefined	RTC カウンタとの比較値の設定

(個別説明)

a. <RTCCP>

このレジスタに書き込まれた値と、カウンタの値が一致することで、割り込み（電源復帰要求）が発生します。

3. RTCPRST (RTC Preset Register)

Address = (0xF003_0000) + 0x0008

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	RTCPR	WO	Undefined	RTC カウンタのプリセット値

(個別説明)

a. <RTCPR>

RTC カウンタのプリセット値を設定します。

4. MLDALMINV (Melody Alarm signal Invert Register)

Address = (0xF003_0000) + 0x0100

Bit	Bit Symbol	Type	Reset Value	Description
[31:1]	–	–	Undefined	Read as undefined. Write as zero.
[0]	MLALINV	WO	0y0	MLDALM 出力信号反転 0y0: 反転しない 0y1: 反転

(個別説明)

a. <MLALINV>

メロディ信号、またはアラーム信号の出力を反転します。

5. MLDALMSEL (Melody Alarm Select Register)

Address = (0xF003_0000) + 0x0104

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	–	–	Undefined	Read as undefined. Write as zero.
[1:0]	MLALSEL	WO	0y00	出力信号選択 0y00: 出力停止 0y01: メロディ 0y10: アラーム 0y11: 設定禁止

(個別説明)

a. <MLALSEL>

MLDALM 端子から、メロディ信号を出力するか、アラーム信号を出力するかを選択します。

6. ALMCNTCR (Alarm Counter Control Register)

Address = (0xF003_0000) + 0x0108

Bit	Bit Symbol	Type	Reset Value	Description
[31:1]	–	–	Undefined	Read as undefined. Write as zero.
[0]	ALMCC	R/W	0y0	フリーランカウンタコントロール 0y0: Clear & Stop 0y1: Start

(個別説明)

a. <ALMCC>

アラーム用 15 ビットカウンタを制御します。

7. ALMPATTERN (Alarm Pattern Register)

Address = (0xF003_0000) + 0x010C

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:0]	ALMPTSEL	WO	0x00	アラームパターンの設定 : 個別説明を参照してください。

(個別説明)

a. <ALMPTSEL>

アラームパターンを選択します。

(アラームパターンの設定値)

ALMPTSEL の設定値	アラーム波形
0x00	“0” 固定
0x01	AL1 パターン
0x02	AL2 パターン
0x04	AL3 パターン
0x08	AL4 パターン
0x10	AL5 パターン
0x20	AL6 パターン
0x40	AL7 パターン
0x80	AL8 パターン
その他	未定義 (設定しないでください。)

8. MLDCNTR (Melody Counter Control Register)

Address = (0xF003_0000) + 0x0110

Bit	Bit Symbol	Type	Reset Value	Description
[31:1]	–	–	Undefined	Read as undefined. Write as zero.
[0]	MLDCC	R/W	0y0	フリーランカウンタコントロール 0y0: Clear & Stop 0y1: Start

(個別説明)

a. <MLDCC>

メロディ用 12bit カウンタを制御します。

9. MLDFRQ (Melody Frequency Register)

Address = (0xF003_0000) + 0x0114

Bit	Bit Symbol	Type	Reset Value	Description
[31:12]	–	–	Undefined	Read as undefined. Write as zero.
[11:0]	MLDF	WO	0x000	メロディ出力周波数の設定値 N : 0x001 ~ 0xFFFF

(個別説明)

a. <MLDF>

メロディ出力の周波数を決めるカウンタ値 (N) を設定します。

計算式 : $f_{MLD[Hz]} = 32768 / (2N+4)$

10. RTCALMINTCTR (RTC ALM Interrupt Control Register)

Address = (0xF003_0000) + 0x0200

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7]	ALMINTCLR	WO	0y0	アラーム割り込みフラグをクリア 0y0 : 無効 0y1 : クリア
[6]	RTCINTCLR	WO	0y0	RTC 割り込みフラグをクリア 0y0 : 無効 0y1 : クリア
[5]	AINTEN1	R/W	0y0	アラーム(1Hz)割り込みの許可 0y0 : 禁止 0y1 : 許可
[4]	AINTEN2	R/W	0y0	アラーム(2Hz)割り込みの許可 0y0 : 禁止 0y1 : 許可
[3]	AINTEN64	R/W	0y0	アラーム(64Hz)割り込みの許可 0y0 : 禁止 0y1 : 許可
[2]	AINTEN512	R/W	0y0	アラーム(512Hz)割り込みの許可 0y0 : 禁止 0y1 : 許可
[1]	AINTEN8192	R/W	0y0	アラーム(8192Hz)割り込みの許可 0y0 : 禁止 0y1 : 許可
[0]	RTCINTEN	R/W	0y0	RTC 割り込みの許可 0y0 : 禁止 0y1 : 許可

(個別説明)

- a. <ALMINTCLR>
許可後の アラーム割り込みフラグをクリア
- b. <RTCINTCLR>
許可後の RTC 割り込みのフラグをクリア
- c. <AINTEN1>
アラーム(1Hz)割り込みの許可
- d. <AINTEN2>
アラーム(2Hz)割り込みの許可
- e. <AINTEN64>
アラーム(64Hz)割り込みの許可
- f. <AINTEN512>
アラーム(512Hz)割り込みの許可
- g. <AINTEN8192>
アラーム(8192Hz)割り込みの許可
- h. <RTCINTEN>
RTC 割り込み許可

11. RTCALMMIS (RTC ALM Masked Interrupt Status Register)

Address = (0xF003_0000) + 0x0204

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	–	–	Undefined	Read as undefined.
[1]	ALMINT	RO	0y0	割り込み許可後のアラーム割り込み 0y0 : 要求無し 0y1 : 要求有り
[0]	RTCINT	RO	0y0	割り込み許可後の RTC 割り込み 0y0 : 要求無し 0y1 : 要求有り



- 使用上の注意事項：

1. RTC の 32bit カウンタは停止することが出来ません。常にカウントアップしています。
2. 電源投入後、RTCCOMP および RTCPRST レジスタは不定状態です。必ず、RTCCOMP および RTCPRST レジスタを設定後、RTCDATA レジスタから正しい値がリードできる事を確認してから、割り込みをイネーブルにしてください。
同様に、RTCCOMP および RTCPRST レジスタなど、RTC のレジスタが不定状態のまま PCM 状態にしないでください (予期せぬ PCM からの復帰の回避のため)。
3. RTC のカウンタ値は、PCM 状態から復帰した後、最大 1 秒後まで、更新されません。
時計表示を行う場合、電源復帰の際は一秒以上経過するまでは、RTC カウンタの値を利用しないで下さい。
4. メロディとアラームの回路電源は DVCC1A 系で供給しています(RTC 回路は DVCC1B 系)。

Not Recommended
for New Design

3.22 アナログ/デジタルコンバータ

4 チャンネルのアナログ入力を持つ、10 ビット逐次変換方式アナログ/デジタルコンバータ (AD コンバータ) を内蔵しています。

図 3.22.1に、ADコンバータのブロック図を示します。4チャンネルのアナログ入力端子(AN4~AN7)は、入力専用ポートD(PD4~PD7)と兼用です。

注 1) PCM モードにより電源電流を低減させる場合は、タイミングにより内部コンパレータがイネーブル状態のままスタンバイに入ることがありますので、AD コンパータの動作が停止していることを確認してから モード移行を実行してください。

注 2) AD コンバータを停止している状態で $ADMOD1<DACON> = 0$ にすると、消費電流の低減が図れます。

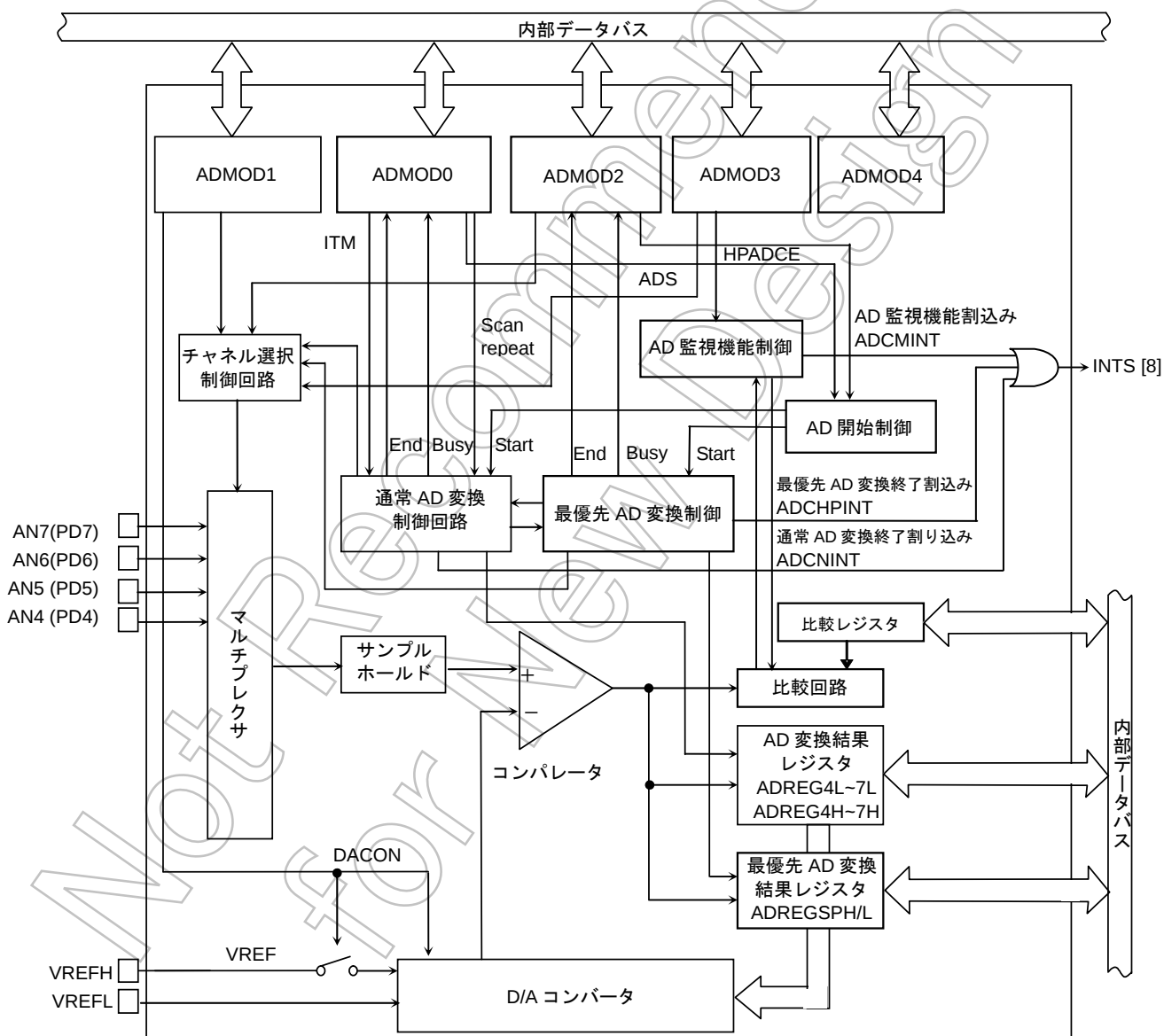


図 3.22.1 AD コンバータのブロック図

3.22.1 レジスタの説明

AD コンバータ関係のレジスタの一覧を以下に示します。ベースアドレスを加算すると本製品のアドレスとなります。

base アドレス= 0xF008_0000

Register Name	Address (base+)	Description
—	0x0000	Reserved
—	0x0004	Reserved
—	0x0008	Reserved
—	0x000C	Reserved
—	0x0010	Reserved
—	0x0014	Reserved
—	0x0018	Reserved
—	0x001C	Reserved
ADREG4L	0x0020	A/D 変換結果下位レジスタ 4
ADREG4H	0x0024	A/D 変換結果上位レジスタ 4
ADREG5L	0x0028	A/D 変換結果下位レジスタ 5
ADREG5H	0x002C	A/D 変換結果上位レジスタ 5
ADREG6L	0x0030	A/D 変換結果下位レジスタ 6
ADREG6H	0x0034	A/D 変換結果上位レジスタ 6
ADREG7L	0x0038	A/D 変換結果下位レジスタ 7
ADREG7H	0x003C	A/D 変換結果上位レジスタ 7
ADREGSPL	0x0040	最優先 A/D 変換結果下位レジスタ
ADREGSPH	0x0044	最優先 A/D 変換結果上位レジスタ
ADCOMREGL	0x0048	A/D 変換結果比較下位レジスタ
ADCOMREGH	0x004C	A/D 変換結果比較下位レジスタ
ADMOD0	0x0050	A/D モードコントロールレジスタ 0
ADMOD1	0x0054	A/D モードコントロールレジスタ 1
ADMOD2	0x0058	A/D モードコントロールレジスタ 2
ADMOD3	0x005C	A/D モードコントロールレジスタ 3
ADMOD4	0x0060	A/D モードコントロールレジスタ 4
—	0x0064	Reserved
—	0x0068	Reserved
—	0x006C	Reserved
ADCLK	0x0070	A/D 変換クロック設定レジスタ
ADIE	0x0074	A/D 割込みイネーブルレジスタ
ADIS	0x0078	A/D 割込みステータスレジスタ
ADIC	0x007C	A/D 割込みクリアレジスタ
—	0x0080	Reserved
—	:	Reserved
—	0x0FFF	Reserved

注) レジスタの説明において、
 R/W: リード/ライト可能
 RO: リード可能/ライトしても反映されません。
 WO: ライト可能/リードした場合は 0 になります。

3.22.1.1 コントロールレジスタ

A/D コンバータは、A/D モードコントロールレジスタ(ADMOD0、ADMOD1、ADMOD2、ADMOD3、ADMOD4)により制御されています。また、A/D 変換結果は、A/D 変換結果上位/下位レジスタ ADREG4H/L~ADREG7H/L の 4 個のレジスタに格納されます。また、最優先変換結果は ADREGSPH/L に格納されます。

- ADMOD レジスタ

1. ADMOD0 (AD mode control register 0)

$$\text{Address} = (0xF008_0000) + (0x0050)$$

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7]	EOCFN (注)	RO	0y0	通常 AD 変換終了フラグ リード時: 0y0: 変換前または変換中 0y1: 終了 ライト時: 無効
[6]	ADBFN	RO	0y0	通常 AD 変換 BUSY フラグ リード時: 0y0: 変換停止 0y1: 変換中 ライト時: 無効
[5]	–	RO	0y0	リードすると常に 0 になります。
[4]	–	R/W	0y0	常に 0 をライトしてください。
[3]	ITM	R/W	0y0	チャンネル固定リピート変換モード時の A/D 変換割込み 0y0: 1 回変換するごとに割込み発生 0y1: Reserved
[2]	REPEAT	R/W	0y0	リピートモード指定 0y0: シングル変換モード 0y1: リピート変換モード
[1]	SCAN	R/W	0y0	スキャンモード指定 0y0: チャンネル固定モード 0y1: チャンネルスキャンモード
[0]	ADS	R/W	0y0	A/D 変換スタート 0y0: Don't care 0y1: 変換開始 リードすると常に 0 になります。

R/W: Read/Write RO: Read Only WO: Write Only

注) <EOCFN>は、読み出すことにより 0 にクリアされます。

(個別説明)

a. <EOCFN>

通常 AD 変換終了フラグの状態を示します。

0y0: 変換前または変換中

0y1: 終了

b. <ADBFN>

通常 AD 変換 BUSY フラグの状態を示します。

0y0: 変換停止

0y1: 変換中

c. <ITM>

チャンネル固定リピート変換モード時の A/D 変換割込み。

0y0: 1 回変換するごとに割込み発生

0y1: Reserved

d. <REPEAT >

リピートモードを選択します。

0y0: シングル変換モード

0y1: リピート変換モード

e. <SCAN >

スキャンモードを選択します。

0y0: チャンネル固定モード

0y1: チャンネルスキャンモード

f. <ADS>

A/D 変換スタートモードを選択します。

0y0: Don't care

0y1: 変換開始

注) リードすると常に 0 になります。

2. ADMOD1 (AD mode control register 1)

Address = (0xF008_0000) + (0x0054)

Bit	Bit Symbol	Type	Reset Value	Description																																											
[31:8]	–	–	Undefined	Read as undefined. Write as zero.																																											
[7]	DACON	R/W	0y0	VREF 印加制御 0y0: OFF 0y1: ON																																											
[6]	–	RO	0y0	リードすると常に 0 になります。																																											
[5]	ADSCN	R/W	0y0	チャンネルスキャン時の動作モード設定 0y0: 4ch スキャン 0y1: Reserved																																											
[4:3]	–	R/W	0y00	常に 0 をライトして下さい。																																											
[2:0]	ADCH[2:0]	R/W	0y000	アナログ入力チャンネル選択 <table><tr><th colspan="2">SCAN</th><th>0</th><th colspan="2">1</th></tr><tr><th colspan="2">ADSCN</th><th>0/1</th><th>0</th><th>1</th></tr><tr><td rowspan="8">ADCH [2:0]</td><td>0y000</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr><tr><td>0y001</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr><tr><td>0y010</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr><tr><td>0y011</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr><tr><td>0y100</td><td>AN4</td><td>AN4</td><td>Reserved</td></tr><tr><td>0y101</td><td>AN5</td><td>AN4-5</td><td>Reserved</td></tr><tr><td>0y110</td><td>AN6</td><td>AN4-6</td><td>Reserved</td></tr><tr><td>0y111</td><td>AN7</td><td>AN4-7</td><td>Reserved</td></tr></table>	SCAN		0	1		ADSCN		0/1	0	1	ADCH [2:0]	0y000	Reserved	Reserved	Reserved	0y001	Reserved	Reserved	Reserved	0y010	Reserved	Reserved	Reserved	0y011	Reserved	Reserved	Reserved	0y100	AN4	AN4	Reserved	0y101	AN5	AN4-5	Reserved	0y110	AN6	AN4-6	Reserved	0y111	AN7	AN4-7	Reserved
SCAN		0	1																																												
ADSCN		0/1	0	1																																											
ADCH [2:0]	0y000	Reserved	Reserved	Reserved																																											
	0y001	Reserved	Reserved	Reserved																																											
	0y010	Reserved	Reserved	Reserved																																											
	0y011	Reserved	Reserved	Reserved																																											
	0y100	AN4	AN4	Reserved																																											
	0y101	AN5	AN4-5	Reserved																																											
	0y110	AN6	AN4-6	Reserved																																											
	0y111	AN7	AN4-7	Reserved																																											

R/W: Read/Write RO: Read Only WO: Write Only

注 1) AD 変換をスタートさせる場合は、かならず ADMOD1<DACON>に 1 をライトした後、内部基準電圧が安定するまでの 3μs 待ってから、ADMOD0<ADS>に 1 をライトしてください。

注 2) AD 変換終了後にスタンバイモードへ移行する場合は、ADMOD1<DACON>を 0 に設定して下さい。

(個別説明)

a. <DACON>

VREF 印加を制御します。

0y0: OFF

0y1: ON

b. <ADSCN>

チャンネルスキャン時の動作モードを設定します。

0y0: 4ch スキャン

0y1: Reserved

- c. <ADCH[2:0]>
アナログ入力チャネルを選択します。

SCAN		0	1	
ADSCN		0/1	0	1
ADCH [2:0]	0y000	Reserved	Reserved	Reserved
	0y001	Reserved	Reserved	Reserved
	0y010	Reserved	Reserved	Reserved
	0y011	Reserved	Reserved	Reserved
	0y100	AN4	AN4	Reserved
	0y101	AN5	AN4~5	Reserved
	0y110	AN6	AN4~6	Reserved
	0y111	AN7	AN4~7	Reserved

Not Recommended for New Design

3. ADMOD2 (AD mode control register 2)

Address = (0xF008_0000) + (0x0058)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7]	EOCFHP (注)	RO	0y0	最優先 AD 変換終了フラグ リード時: 0y0: 変換前または変換中 0y1: 終了 ライト時: 無効
[6]	ADBFHP	RO	0y0	最優先 AD 変換 BUSY フラグ リード時: 0y0: 変換停止 0y1: 変換中 ライト時: 無効
[5]	HPADCE	R/W	0y0	最優先 AD 変換起動 0y0: Don't care 0y1: 変換開始 リードすると常に 0 になります。
[4:3]	–	R/W	0y00	常に 0 をライトして下さい。
[2:0]	HPADCH[2:0]	R/W	0y000	最優先変換時のアナログ入力チャネル選択 0y100: AN4 0y110: AN6 0y101: AN5 0y111: AN7 Other: Reserved

R/W: Read/Write RO: Read Only WO: Write Only

注) <EOCFHP>は、読み出すことにより 0 にクリアされます。

(個別説明)

a. <EOCFHP>

最優先 AD 変換終了フラグです。

0y0: 変換前または変換中

0y1: 終了

b. <ADBFHP>

最優先 AD 変換 BUSY フラグです。

0y0: 変換停止

0y1: 変換中

c. <HPADCE>

最優先 AD 変換起動を制御します。

0y0: Don't care

0y1: 変換開始

注) リードすると常に 0 になります。

d. <HPADCH[2:0]>

最優先変換時のアナログ入力チャネル選択

0y100: AN4 0y110: AN6

0y101: AN5 0y111: AN7

Other: Reserved

4. ADMOD3 (AD mode control register 3)

Address = (0xF008_0000) + (0x005C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7]	–	R/W	0y0	常に 0 をライトして下さい。
[6]	–	RO	0y0	リードすると常に 0 になります。
[5]	ADOBIC	R/W	0y0	AD 監視機能割込みの設定 0y0: 比較レジスタ値より小さい 0y1: 比較レジスタ値より大きい
[4:1]	REGS[3:0]	R/W	0y0000	AD 監視機能が Enable 時に、比較レジスタの内容と比較される AD 変換結果格納レジスタの選択ビット REGS[3:0] 0y0000: Reserved 0y0001: Reserved 0y0010: Reserved 0y0011: Reserved 0y0100: ADREG4 0y0101: ADREG5 0y0110: ADREG6 0y0111: ADREG7 0y1xxx: ADREGSP
[0]	ADOBSV	R/W	0y0	AD 監視機能 0y0: Disable 0y1: Enable

R/W: Read/Write RO: Read Only WO: Write Only
(個別説明)

a. <ADOBIC>

AD 監視機能割込みを設定します。

0y0: 比較レジスタ値より小さい

0y1: 比較レジスタ値より大きい

b. <REGS[3:0]>

AD 監視機能が Enable 時に、比較レジスタの内容と比較される AD 変換結果格納レジスタを選択します。

0y0000: Reserved 0y0001: Reserved

0y0010: Reserved 0y0011: Reserved

0y0100: ADREG4 0y0101: ADREG5

0y0110: ADREG6 0y0111: ADREG7

0y1xxx: ADREGSP

c. <ADOBSV>

AD 監視機能を制御します。

0y0: Disable

0y1: Enable

5. ADMOD4 (AD mode control register 4)

Address = (0xF008_0000) + (0x0060)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7]	–	R/W	0y0	常に 0 をライトして下さい。
[6]	–	R/W	0y0	常に 0 をライトして下さい。
[5]	–	R/W	0y0	常に 0 をライトして下さい。
[4]	–	R/W	0y0	常に 0 をライトして下さい。
[3:2]	–	RO	0y00	リードすると常に 0 になります。
[1:0]	ADRST[1:0]	R/W	0y00	0y10→0y01 のライトで ADC をソフトウェアリセットします。(ADCLK)レジスタ以外は全て初期化されます。

R/W: Read/Write RO: Read Only WO: Write Only

(個別説明)

a. <ADRST[1:0]>

0y10→0y01 のライトで ADC をソフトウェアリセットします。ADCLK レジスタ以外は全て初期化されます。

- AD 変換結果レジスタ

A/D 変換結果は、A/D 変換結果上位/下位レジスタ ADREG4H/L~ADREG7H/L の 4 個のレジスタに格納されます。また、最優先変換結果は ADREGSPH/L に格納されます。これらのレジスタは同じ構造であるため、4 チャンネルの変換結果格納レジスタである ADREG4H/L を示します。

1. ADREG4L (AD 変換結果下位レジスタ 4)

Address = (0xF008_0000) + (0x0020)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined.
[7]	ADR41	RO	0y0	AD 変換結果下位ビット 1
[6]	ADR40	RO	0y0	AD 変換結果下位ビット 0
[5:2]	–	RO	0y0000	リードすると常に 0 になります。
[1]	OVR4	RO	0y0	オーバランフラグ 0y0: 発生なし 0y1: 発生あり
[0]	ADR4RF	RO	0y0	AD 変換結果格納フラグ 0y0: 変更結果なし 0y1: 変換結果あり

(個別説明)

a. <ADR4[1:0]>

AD 変換結果下位ビット 1~0 です。

b. <OVR4>

オーバランフラグです。

0y0: 発生なし

0y1: 発生あり

c. <ADR4RF>

AD 変換結果格納フラグです。

0y0: 変更結果なし

0y1: 変換結果あり

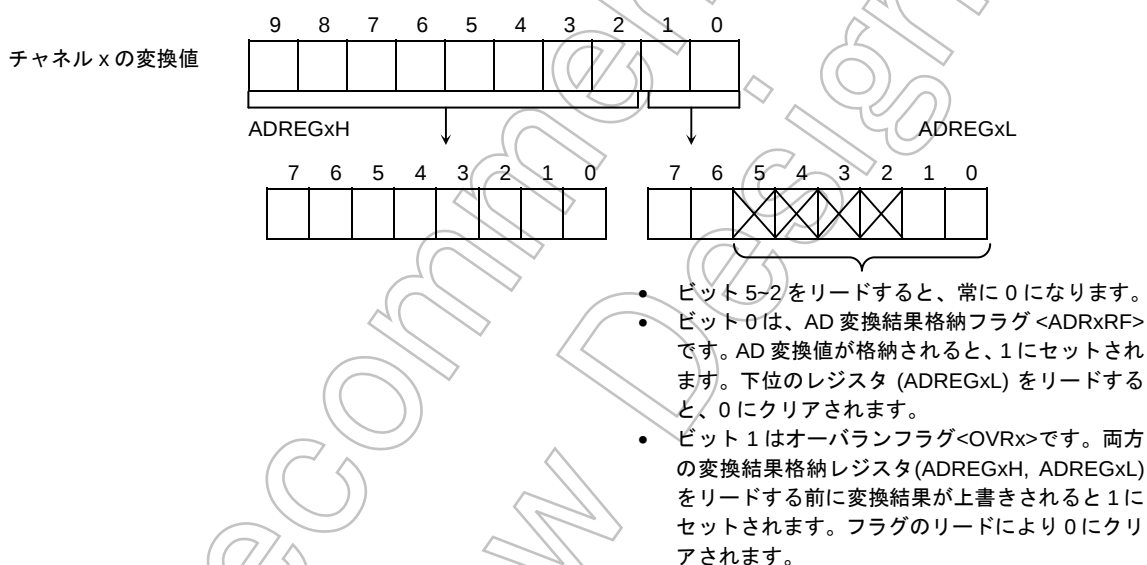
注) レジスタの構造は ADREG4L~ADREG7L および ADREGSPL は全て同じです。そのため、ADREG4L のみ表現しています。

ADREG4L~ADREG7L および ADREGSPL のレジスタのアドレスは、レジスタマップでご確認願います。

2. ADREG4H (AD 変換結果上位レジスタ 4)

Address = (0xF008_0000) + (0x0024)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined.
[7]	ADR49	RO	0y0	AD 変換結果上位ビット 9
[6]	ADR48	RO	0y0	AD 変換結果上位ビット 8
[5]	ADR47	RO	0y0	AD 変換結果上位ビット 7
[4]	ADR46	RO	0y0	AD 変換結果上位ビット 6
[3]	ADR45	RO	0y0	AD 変換結果上位ビット 5
[2]	ADR44	RO	0y0	AD 変換結果上位ビット 4
[1]	ADR43	RO	0y0	AD 変換結果上位ビット 3
[0]	ADR42	RO	0y0	AD 変換結果上位ビット 2



(個別説明)

a. <ADR4[9:2]>

AD 変換結果上位ビット 9~2 です。

注) レジスタの構造は ADREG4H~ADREG7H および ADREGSPH は全て同じです。そのため、ADREG4H のみ表現しています。

ADREG5H~ADREG7H および ADREGSPH のレジスタのアドレスは、レジスタマップでご確認願います。

- AD 変換結果比較レジスタ

1. ADCOMREGL (AD 変換結果比較下位レジスタ)

Address = (0xF008_0000) + (0x0048)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:6]	ADRCOM[1:0]	R/W	0y00	AD 変換結果比較下位ビット 1~0
[5:0]	–	RO	0y000000	リードすると常に 0 になります。

(個別説明)

a. <ADRCOM[1:0]>

AD 変換結果比較下位ビット 1~0 です。

2. ADCOMREGH (AD 変換結果比較上位レジスタ)

Address = (0xF008_0000) + (0x004C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7]	ADRCOM9	R/W	0y0	AD 変換結果比較上位ビット 9
[6]	ADRCOM8	R/W	0y0	AD 変換結果比較上位ビット 8
[5]	ADRCOM7	R/W	0y0	AD 変換結果比較上位ビット 7
[4]	ADRCOM6	R/W	0y0	AD 変換結果比較上位ビット 6
[3]	ADRCOM5	R/W	0y0	AD 変換結果比較上位ビット 5
[2]	ADRCOM4	R/W	0y0	AD 変換結果比較上位ビット 4
[1]	ADRCOM3	R/W	0y0	AD 変換結果比較上位ビット 3
[0]	ADRCOM2	R/W	0y0	AD 変換結果比較上位ビット 2

注) このレジスタへ値を設定する時、または値を変更する時は、AD 監視機能を禁止(ADMOD3<ADOBSV> = 0)した状態で行って下さい。

(個別説明)

a. <ADRCOM[9:2]>

AD 変換結果比較上位ビット 9~2 です。

- AD 変換クロック設定レジスタ

1. ADCLK (AD 変換クロック設定レジスタ)

Address = (0xF008_0000) + (0x0070)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7]	–	R/W	0y1	常に 1 をライトして下さい。
[6:4]	–	R/W	0y000	常に 0 をライトして下さい。
[3]	–	RO	0y0	リードすると常に 0 になります。
[2:0]	ADCLK[2:0]	R/W	0y000	AD プリスケアラ出力選択 AD 変換 1 クロック周期= 0y000: PCLK 0y001: PCLK/2 0y010: PCLK/4 0y011: PCLK/8 0y1xx: PCLK/16

注 1) AD 変換は上記レジスタで選択されたクロックで実行されますが、保証精度を満足するためには AD 変換クロックを 33MHz 以下になるように変換クロックを選択する必要があります。

注 2) AD 変換中は変換クロックを切り替えないでください。

(個別説明)

a. <ADCLK[2:0]>

AD プリスケアラ出力を選択します。

AD 変換 1 クロック周期=

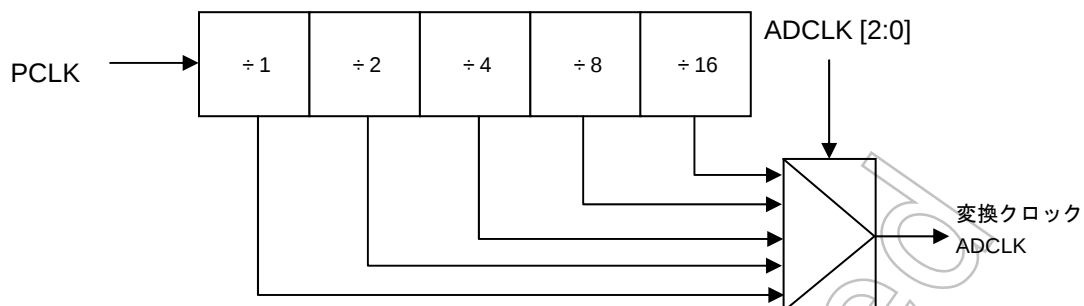
0y000: PCLK

0y001: PCLK/2

0y010: PCLK/4

0y011: PCLK/8

0y1xx: PCLK/16



PCLK	<ADCLK2:0>	ADCLK	AD 変換速度
100MHz	00x	—	設定禁止
	010(PCLK/4)	25MHz	1.84 μsec
96MHz	00x	—	設定禁止
	010(PCLK/4)	24MHz	1.92 μsec

AD 変換速度は下記計算式で求められます。

$$\text{変換速度} = 46 \times (1/\text{ADCLK})$$

注) AD 変換動作中に、ADCLK の値を変更しないでください。

- 割込みレジスタ

1. ADIE (AD 割込みイネーブルレジスタ)

Address = (0xF008_0000) + (0x0074)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	—	—	Undefined	Read as undefined. Write as zero.
[7:3]	—	RO	0y00000	リードすると常に 0 になります。
[2]	MIE	R/W	0y0	AD 監視割込みイネーブル 0y0: 禁止 0y1: 許可
[1]	HPIE	R/W	0y0	最優先 AD 変換割込みイネーブル 0y0: 禁止 0y1: 許可
[0]	NIE	R/W	0y0	通常 AD 変換割込みイネーブル 0y0: 禁止 0y1: 許可

注) 最優先 AD 変換割込みと通常 AD 変換割込みイネーブルを同時に許可しないで下さい。

(個別説明)

a. <MIE>

AD 監視割込みを制御します。

0y0: 禁止

0y1: 許可

b. <HPIE>

最優先 AD 変換割込みを制御します。

0y0: 禁止

0y1: 許可

c. <NIE>

通常 AD 変換割込みを制御します。

0y0: 禁止

0y1: 許可

2. ADIS (AD 割込みステータスレジスタ)

Address = (0xF008_0000) + (0x0078)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined.
[7:3]	–	RO	0y00000	リードすると常に 0 になります。
[2]	MIS	RO	0y0	AD 監視割込みのマスク前ステータス 0y0: なし 0y1: 割込みあり
[1]	HPIS	RO	0y0	最優先 AD 変換割込みのマスク前ステータス 0y0: なし 0y1: 割込みあり
[0]	NIS	RO	0y0	通常 AD 変換割込みのマスク前ステータス 0y0: なし 0y1: 割込みあり

(個別説明)

a. <MIS>

AD 監視割込みのマスク前ステータスの状態を示します。

0y0: なし

0y1: 割込みあり

b. <HPIS>

最優先 AD 変換割込みのマスク前ステータスの状態を示します。

0y0: なし

0y1: 割込みあり

c. <NIS>

通常 AD 変換割込みのマスク前ステータスの状態を示します。

0y0: なし

0y1: 割込みあり

3. ADIC (AD 割込みクリアレジスタ)

Address = (0xF008_0000) + (0x007C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:3]	–	RO	0y00000	リードすると常に 0 になります。
[2]	MIC	WO	0y0	AD 監視割込みクリア 0y0: – 0y1: クリア
[1]	HPIC	WO	0y0	最優先 AD 変換割込みクリア 0y0: – 0y1: クリア
[0]	NIC	WO	0y0	通常 AD 変換割込みクリア 0y0: – 0y1: クリア

(個別説明)

a. <MIC>

AD 監視機能割込みを制御します。

0y0: –

0y1: クリア

b. <HPIC>

最優先 AD 変換割込みを制御します。

0y0: –

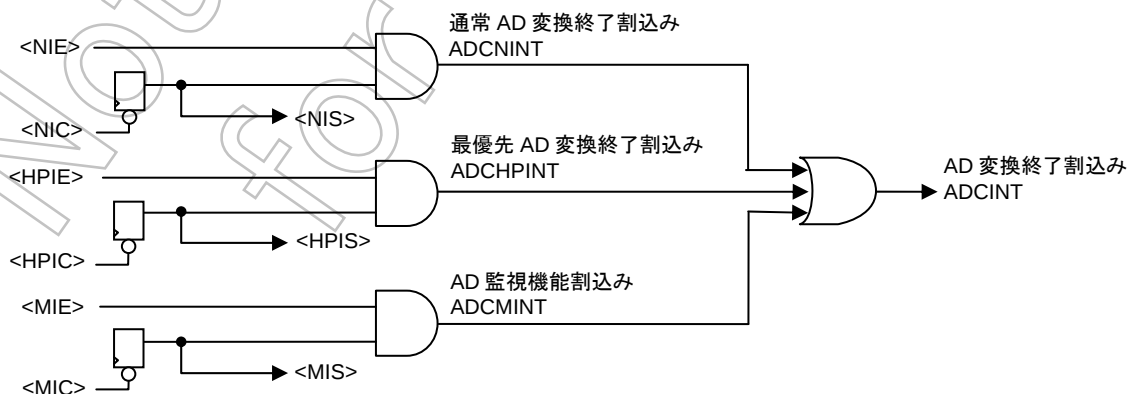
0y1: クリア

c. <NIC>

通常 AD 変換割込みを制御します。

0y0: –

0y1: クリア



3.22.2 動作説明

3.22.2.1 アナログ基準電圧

アナログ基準電圧の“H”レベル側を VREFH 端子に、“L”レベル側を VREFL 端子に印加します。ADMOD1<DACON>に 0 をライトすることにより、VREFH-VREFL 間のスイッチを OFF できます。AD 変換をスタートさせる場合は、かならず<DACON>に 1 をライトした後、内部基準電圧が安定するまでの 3 μ s を待ってから、ADMOD0<ADS>に 1 をライトしてください。

3.22.2.2 アナログ入力チャネルの選択

アナログ入力チャネルの選択は、AD コンバータの動作モードによって異なります。

(1) 通常 AD 変換時

アナログ入力チャネルを固定で使用する場合 (ADMOD0<SCAN>=0)

ADMOD1<ADCH[2:0]>の設定により、アナログ入力 AN4 ~ AN7 端子の中から 1 チャネルを選択します。

アナログ入力チャネルをスキャンで使用する場合 (ADMOD0<SCAN>=1)

ADMOD1<ADCH[2:0]>の設定により、4 種類のスキャンモードの中から 1 つのスキャンモードを選択します。

(2) 最優先 AD 変換時

ADMOD2<HPADCH[2:0]>の設定により、アナログ入力 AN4~AN7 端子の中から 1 チャネルを選択します。

リセット後は ADMOD0<SCAN> は 0 に ADMOD1<ADCH[2:0]> は 0y000 に初期化され、これにより選択が行なわれますので、AN0 端子のチャネル固定入力を選択されます。なお、アナログ入力チャネルとして使用しない端子は、通常のポートとして使用できます。

3.22.2.3 AD変換開始

AD 変換には、通常 AD 変換と最優先 AD 変換の 2 種類があります。

通常 AD 変換は ADMOD0<ADS> に 1 を設定することによりソフトで起動が掛かります。また、最優先 AD 変換は ADMOD2<HPADCE> に 1 を設定することによりソフトで起動が掛かります。

通常 AD 変換は ADMOD0<REPEAT,SCAN>で指定される 4 種類の動作モードから 1 つの動作モードが選択されます。最優先 AD 変換の動作モードはチャンネル固定のシングル変換のみです。

通常 AD 変換が開始されると、AD 変換中を示す AD 変換 Busy フラグ (ADMOD0<ADBFN>) が 1 にセットされます。また、最優先 AD 変換が開始されると、AD 変換中を示す AD 変換 Busy フラグ (ADMOD2<ADBFHP>) が 1 にセットされます。

通常 AD 変換中に最優先変換開始された場合、ADMOD0<ADBFN>は 1 に保持されます。

<EOCFHP>, <EOCFN>は変換が終了すると 1 になります。このフラグはリードした時のみ 0 にクリアされます。

AD 変換には、通常 AD 変換と最優先 AD 変換の 2 種類があり、通常変換中に最優先で AD 変換を開始することが出来ます。

通常 AD 変換中に ADMOD2<HPADCE>に 1 を設定し、最優先 AD 変換を開始すると、現在変換中の AD 変換を直ちに中止して、最優先 AD 変換が始まり ADMOD2<HPADCH[2:0]>で指定されるチャンネルの AD 変換(チャンネル固定のシングル変換)が開始されます。この結果が ADREGSPH/L へ格納されると、中止したチャンネルから通常 AD 変換を再開します。

ただし、最優先変換中に、再度最優先変換を開始することは出来ません。

再度、最優先変換を開始する場合は、現在の最優先変換が終了していることを(AD 変換終了フラグ : ADMOD<ADBFHP>)確認した上で起動するようにしてください。

3.22.2.4 AD変換モードとAD変換終了割り込み

AD 変換には、次の 4 つの動作モードが用意されています。通常 AD 変換の場合は ADMOD0<REPEAT, SCAN>の設定により選択ができます。最優先 AD 変換の場合はチャンネル固定のシングル変換モードのみです。

- a. チャンネル固定シングル変換モード
- b. チャンネルスキャンシングル変換モード
- c. チャンネル固定リピート変換モード
- d. チャンネルスキャンリピート変換モード

(1) 通常 AD 変換

動作モードの選択は、ADMOD0<REPEAT, SCAN>で行います。AD 変換が開始されると ADMOD0<ADBFN>が 1 にセットされます。指定された AD 変換が終了すると、通常 AD 変換終了割り込みが発生し、AD 変換シーケンス終了を示す ADMOD0<EOCFN> が 1 にセットされます。

a. チャンネル固定シングル変換モード

ADMOD0 <REPEAT, SCAN> に 0y00 を設定すると、チャンネル固定シングル変換モードになります。

このモードでは、選択した 1 チャンネルの変換を 1 回だけ行います。変換が終了した後、ADMOD0<EOCFN>が 1 にセットにされ、通常 AD 変換終了割り込み要求が発生します。<EOCFN>はリードする事でのみ 0 にクリアされます。

b. チャンネルスキャンシングル変換モード

ADMOD0 <REPEAT,SCAN> に 0y01 を設定すると、チャンネルスキャンシングル変換モードになります。

このモードでは、選択したスキャンチャンネルの変換をそれぞれ 1 回だけ行います。スキャン変換が終了した後、ADMOD0<EOCFN>が 1 にセットにされ、通常 AD 変換終了割り込み要求が発生します。<EOCFN>はリードする事でのみ 0 にクリアされます。

c. チャンネル固定リピート変換モード

ADMOD0<REPEAT,SCAN>に 0y10 を設定するとチャンネル固定リピート変換モードになります。

このモードでは、選択した 1 チャンネルの変換を繰り返し行います。変換が終了した後、ADMOD0<EOCFN>が 1 にセットされます。通常 AD 変換終了割り込み要求発生タイミングは ADMOD0 <ITM> の設定により選択できます。<EOCFN>がセットされるタイミングも割り込みのタイミングに連動します。

ADMOD0<EOCFN>はリードする事でのみ 0 にクリアされます。

<ITM> を 0 に設定すると AD 変換が 1 回終了するごとに割り込み要求が発生します。この場合変換結果は常に格納レジスタの ADREGxH/L に格納されます。格納時点で<EOCFN>は 1 になります。

d. チャネルスキャンリピート変換モード

ADMOD0 <REPEAT, SCAN> に 0y11 を設定するとチャネルスキャンリピート変換モードになります。

このモードでは、選択したスキャンチャンネルの変換を繰り返し行います。最終チャンネルの変換が終了するごとに ADMOD0<EOCFN> が 1 にセットされ、通常 AD 変換終了割り込み要求が発生します。ADMOD0<EOCFN> はリードする事でのみ 0 にクリアされます。

リピート変換モード (c, d のモード) の動作を停止する場合は、ADMOD1 <REPEAT> に 0 をライトしてください。実行中のスキャン変換を終了した時点で、リピート変換モードは終了します。

(2) 最優先 AD 変換

動作モードはチャンネル固定のシングル変換のみです。ADMOD0<REPEAT, SCAN> の設定は関係ありません。起動条件が成立すると、ADMOD2<HPADCH[2:0]> で指定されるチャンネルの変換を一回だけ行います。変換が終了すると、最優先 AD 変換終了割り込みが発生して、ADMOD2<EOCFHP> は 1 にセットされます。EOCFHP フラグはリードする事でのみ 0 にクリアされます。

表 3.22.1 AD 変換モードと割り込み発生タイミング、フラグ動作の関係

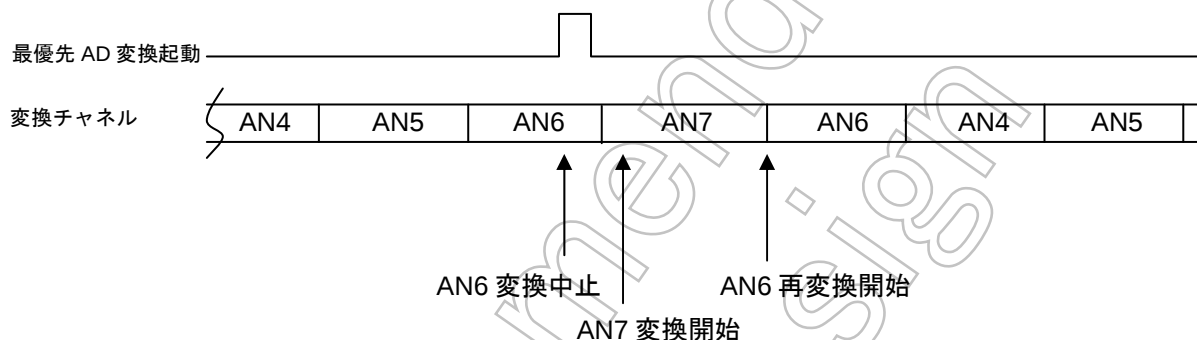
変換モード	割り込み発生 タイミング	EOCFN セット タイミング (注)	ADMOD0		
			ITM	REPEAT	SCAN
チャンネル固定 シングル変換	変換終了後	変換終了後	–	0	0
チャンネル固定 リピート変換	1 回変換ごと	変換が 1 回 終了ごと	0	1	0
チャンネルスキャン シングル変換	スキャン変換 終了後	スキャン変換 終了後	–	0	1
チャンネルスキャン リピート変換	1 回のスキャン 変換終了ごと	1 回のスキャン 変換終了ごと	–	1	1

注) EOCFN はリードする事でのみ 0 にクリアされます。

3.22.2.5 最優先変換モード

通常 AD 変換に割り込んで、最優先 AD 変換を行う事ができます。最優先 AD 変換は $ADMOD2<HPADCE>$ に 1 を設定することによりソフトで起動が掛かります。通常 AD 変換中に最優先 AD 変換が起動されると、現在変換中の AD 変換を直ちに中止して $ADMOD2<HPADCH[2:0]>$ で指定されるチャンネルのシングル変換を行います。変換結果は $ADREGSPH/L$ へ格納され、最優先 AD 変換割込みが発生します。その後に通常 AD 変換が中止したチャンネルから再開されます。また、最優先 AD 変換中に最優先 AD 変換を起動しても無視されます。

例) $ADMOD0<REPEAT,SCAN> = 0y11$ 、 $ADMOD1<ADCH2:0> = 0y110$ でチャンネル AN4~AN6 までのリピートスキャン変換中に $ADMOD2<HPADCH[2:0]> = 0y111$ で AN7 の最優先 AD 変換が起動された場合



3.22.2.6 AD監視機能

$ADMOD3<ADOBSV>$ に 1 を設定すると、AD 監視機能が有効になり、 $ADMOD3<REGS[3:0]>$ で指定されたアナログチャンネル変換終了後に結果格納レジスタの内容が AD 変換結果比較レジスタ H/L の値より大または小 ($ADMOD3<ADOIBC>$ で大か小は指定) になると $ADIE<MIE>$ の許可状態に応じて AD 監視機能割り込みが発生します。また、この比較動作は該当変換結果格納レジスタへ結果が格納されるごとに行われ、条件が成立すると割り込みが発生します。また、AD 監視機能に割り当てている格納レジスタは通常ではソフトでリードは行われませんので、オーバーランフラグ $<OVRx>$ は常にセットされていることになり、変換結果格納フラグ $<ADR_xRF>$ もセットされているので注意してください。

3.22.2.7 AD変換時間

1 回当たりの AD 変換は、サンプリングクロックを含む 46 クロックです。AD 変換クロックは $<ADCLK[2:0]>$ によって、PCLK の 1/1, 1/2, 1/4, 1/8, 1/16 の中から選択されます。保証精度を満足するためには AD 変換クロックを 0.625MHz 以上 33MHz 以下、すなわち AD 変換時間を 1.39μs 以上 73.6μs 以下にする必要があります。

3.22.2.8 AD変換結果の格納とリード

AD変換結果は、通常AD変換のAD変換結果上位/下位レジスタ (ADREG4H/L~ADREG7H/L) に格納されます(ADREG4H/L~ADREG7H/Lは、リード専用のレジスタです)。

AD変換結果は、チャンネルAN4、AN5、AN6、AN7の変換結果がそれぞれADREG4H/L、ADREG5H/L、ADREG6H/L、ADREG7H/Lに格納されます。

表 3.22.2にアナログ入力チャンネルとAD変換結果レジスタの対応を示します。

表 3.22.2アナログ入力チャンネルと AD 変換結果レジスタの対応

アナログ入力 チャンネル (ポート D)	AD 変換結果レジスタ
AN4	ADREG4H/L
AN5	ADREG5H/L
AN6	ADREG6H/L
AN7	ADREG7H/L

注) オーバーランを漏れなく検出するために、変換結果格納レジスタの上位を先にリードし、次に下位をリードしてください。この結果、下位に存在する $OVRn = 0$ 、 $ADRnRF = 1$ であれば、正しい変換結果を得た事になります。

3.22.2.9 データポーリング

割り込みを使用せずにデータポーリングでAD変換結果を処理する場合は $ADM0D0 < EOCFN >$ のポーリングをしてください。 $ADM0D0 < EOCFN >$ が 1 にセットされたことを確認後にAD変換格納レジスタをリードしてください。

3.23 ウォッチドッグタイマ (暴走検出用タイマWDT)

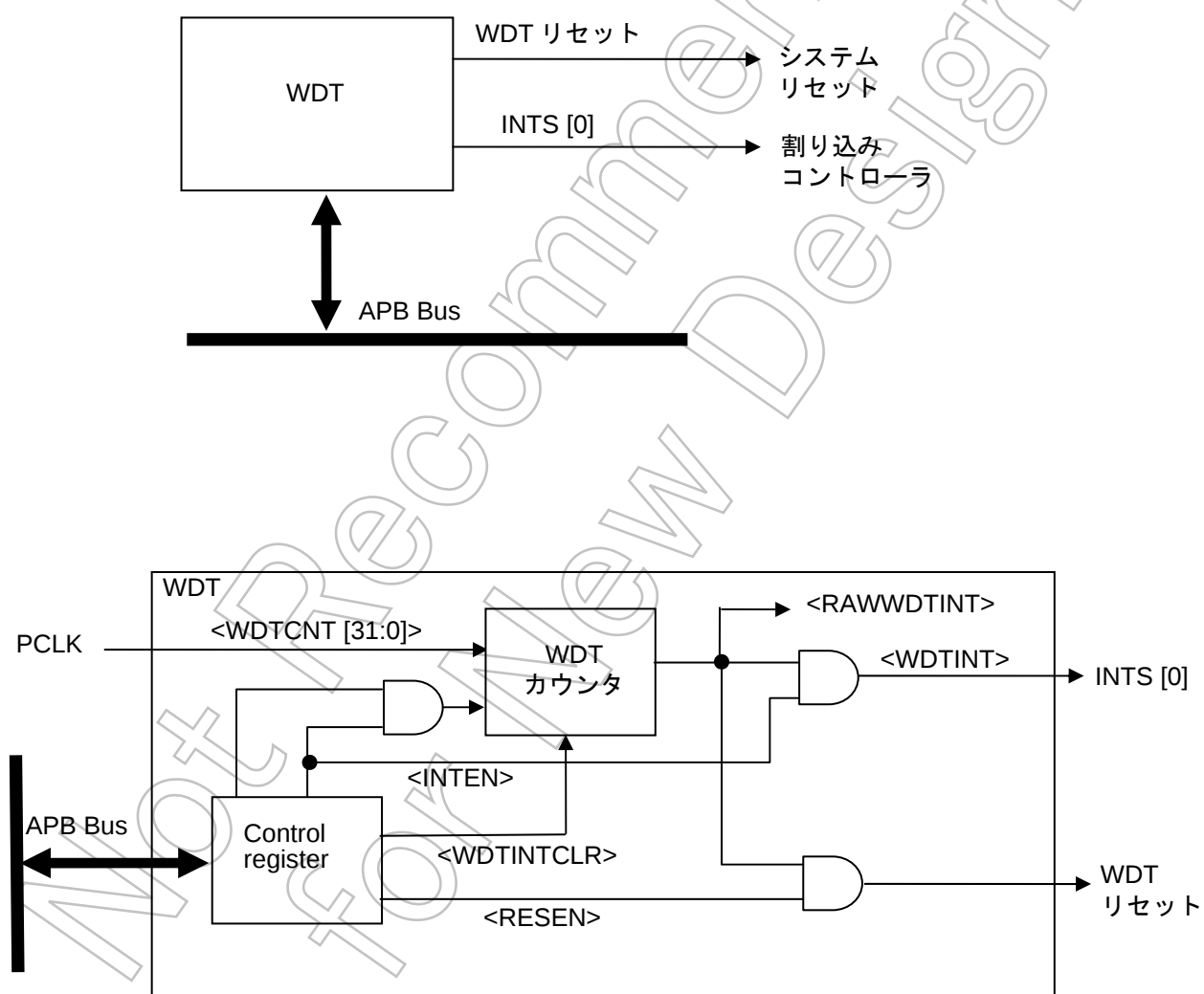
暴走検出用のウォッチドッグタイマを内蔵しています。

ウォッチドッグタイマ (WDT) は、ノイズなどの原因により CPU が誤動作 (暴走) を始めた場合これを検出し、正常な状態に戻すことを目的としています。暴走を検出すると割り込みを発生し、割り込みコントローラおよび CPU に知らせます(割り込みコントローラへ割り込み要因信号は INTS [0])。

また、このウォッチドッグタイマアウトを内部リセットへ接続することにより、強制的にリセット動作を行うことができます。

注) デバッグ中は WDT を停止してください。

3.23.1 ブロック図



3.23.2 レジスタ機能

内蔵レジスタリストと機能を以下に示します。

base アドレス= 0xF001_0000

Register Name	Address (base+)	Description
WdogLoad	0x0000	Watchdog load register
WdogValue	0x0004	The current value for the watchdog counter
WdogControl	0x0008	Watchdog control register
WdogIntClr	0x000C	Clears the watchdog interrupt
WdogRIS	0x0010	Watchdog raw interrupt status
WdogMIS	0x0014	Watchdog masked interrupt status
WdogLock	0x0C00	Watchdog Lock register

1. WdogLoad (Watchdog load register)

Address = (0xF001_0000) + (0x0000)

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	WDTCNT	R/W	0xFFFFFFFF	WDT の検出カウンタの設定値 0x00000001 ~ 0xFFFFFFFF

(個別説明)

a. <WDTCNT>

WDT の 32 ビットのカウント値を設定します(WDT 検出カウンタのクロックは PCLK)。
 後述の WdogControl<INTEN> を許可後、内部デクリメントカウンタに
 WdogLoad<WDTCNT>設定値をロードします。
 カウンタに設定できる値は 0x00000001 ~ 0xFFFFFFFF です。(0 は設定できません)
 また、リードした場合、設定値が読み出されます。

2. WdogValue (The current value for the watchdog counter)

Address = (0xF001_0000) + (0x0004)

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	CWDTCNT	RO	0xFFFFFFFF	現在 WDT 検出カウンタ値

(個別説明)

a. <CWDTCNT>

現在のウォッチドッグカウンタの値を読み出すことができます。

3. WdogControl (Watchdog control register)

Address = (0xF001_0000) + (0x0008)

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	—	—	Undefined	Read as undefined. Write as zero.
[1]	RESEN	R/W	0y0	WDT リセット出力許可 0y0 : 禁止 0y1 : 許可
[0]	INTEN	R/W	0y0	WDT カウンタおよび割り込み許可 0y0 : 禁止 0y1 : 許可

(個別説明)

a. <RESEN>

WDT リセットの出力許可を制御するビットです。

また、リセット解除までの時間は PCLK 5 クロック後となります。

b. <INTEN>

0y1: WDT のカウンタを Enable、WdogLoad レジスタからカウンタ設定値をリロードして、カウンタはデクリメント開始し、割り込みを許可します。

4. WdogIntClr (Clears the watchdog interrupt)

Address = (0xF001_0000) + (0x000C)

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	WDTINTCLR	WO	Undefined	WDT の割り込みをクリア (値が任意)

(個別説明)

a. <WDTINTCLR>

このレジスタに任意の値をライトすると、WDT の割り込みをクリアします。また、カウンタに WdogLoad レジスタの設定値をロードします。

5. WdogRIS (Watchdog raw interrupt status)

Address = (0xF001_0000) + (0x0010)

Bit	Bit Symbol	Type	Reset Value	Description
[31:1]	–	–	Undefined	Read as undefined.
[0]	RAWWDTINT	RO	0y0	許可ゲート前の割り込みステータス 0y0 : 割り込みなし 0y1 : 割り込み有り

(個別説明)

a. <RAWWDTINT>

WDT カウンタからの割り込みステータスです。この値と割り込み許可信号(WdogControl <INTEN>)との AND で許可後の割り込み(WdogMIS< WDTINT >)を生成します。

6. WdogMIS (Watchdog masked interrupt status)

Address = (0xF001_0000) + (0x0014)

Bit	Bit Symbol	Type	Reset Value	Description
[31:1]	–	–	Undefined	Read as undefined.
[0]	WDTINT	RO	0y0	許可ゲート後の割り込みステータス 0y0 : 割り込みなし 0y1 : 割り込み有り

(個別説明)

a. <WDTINT>

WDT カウンタからの割り込みステータスです。WdogRIS <RAWWDTINT>と WdogControl <INTEN>の AND 値が読み出されます。

7. WdogLock (Watchdog Lock register)

Address = (0xF001_0000) + (0x0C00)

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	REGWEN	WO	undefined	他の WDT レジスタへライト Enable 0x1ACCE551: Enable Others: Disable (初期状態は Enable)

Address = (0xF001_0000) + (0x0C00)

Bit	Bit Symbol	Type	Reset Value	Description
[31:1]	Reserved	–	undefined	Read as undefined.
[0]	REGWENST	RO	0y0	他の WDT レジスタへライト Disable のステータス: 0y0: Enable (ロックしていない) 0y1: Disable (ロック)

(個別説明)

a. <REGWEN>

プログラムの暴走等により WDT レジスタが書き換わることを防ぐために、本レジスタ以外の WDT レジスタへライトを禁止することができます。

0x1ACCE551 以外のデータをライトすると、本レジスタ以外の WDT レジスタへライトを禁止します。

0x1ACCE551 をライトすると、本レジスタ以外の WDT レジスタへライトを許可します。

b. <REGWENST>

他の WDT レジスタへライト Disable、ロックするステータスを示します。

3.24 電源管理回路PMC (Power Management Circuit)

微細プロセス製品のリーク(漏れ)電流対策のために、本製品ではスタンバイ時の電流を管理する電源管理回路を内蔵しています。電源供給は下記の 8 系統があります。

- 3.3V 系 A/D コンバータ用電源 (A/D コンバータ用 : AVCC3AD & AVSS3AD)
- 3.3V 系デジタル電源 (一般端子用 : DVCC3IO & DVSSCOM (注))
- 3.3V 及び 1.8V 系メモリ用電源 (メモリ制御用 : DVCCM & DVSSCOM)
- 3.3V 系 USB Device 用電源 (USB Device 制御用 : AVDD3T/C & AVSS3T/C)
- 3.3V 系 USB Host 用電源 (USB Host 制御用 : AVCC3H)
- 1.5V-A 系内部電源 (一般回路用 : DVCC1A & DVSSCOM)
- 1.5V-B 系内部電源 (RTC、PMC 用 : DVCC1B & DVSSCOM)
- 1.5V-C 系発振電源 (高周波発振器、PLL 用 : DVCC1C & DVSS1C)

注) 低周波発振回路(XT1, XT2)は 3.3V 系デジタル電源(DVCC3IO)から電源供給されます。

各電源供給は独立しています(VSS は一部共通)。

PCM 状態では、外部回路で大半の回路へ電源を遮断することによりスタンバイ状態におけるリーク電流を低減します。

また、そのときの外部端子の状態は“H 出力”、“L 出力”、“Hi-z”、および入力状態に固定することが出来ます (バックアップレジスタ出力データレジスタ一覧参照)。

8 系統の電源の内、PCM 状態時に供給すべき電源は、DVCC3IO, DVCCM, AVCC3AD, DVCC1B です。

遮断すべき電源は、DVCC1A, DVCC1C, AVDD3T/AVDD3C および AVCC3H です。これらは PCM 状態に移行した後電源を遮断しても LSI 内に貫通電流等が発生しないようになっています。

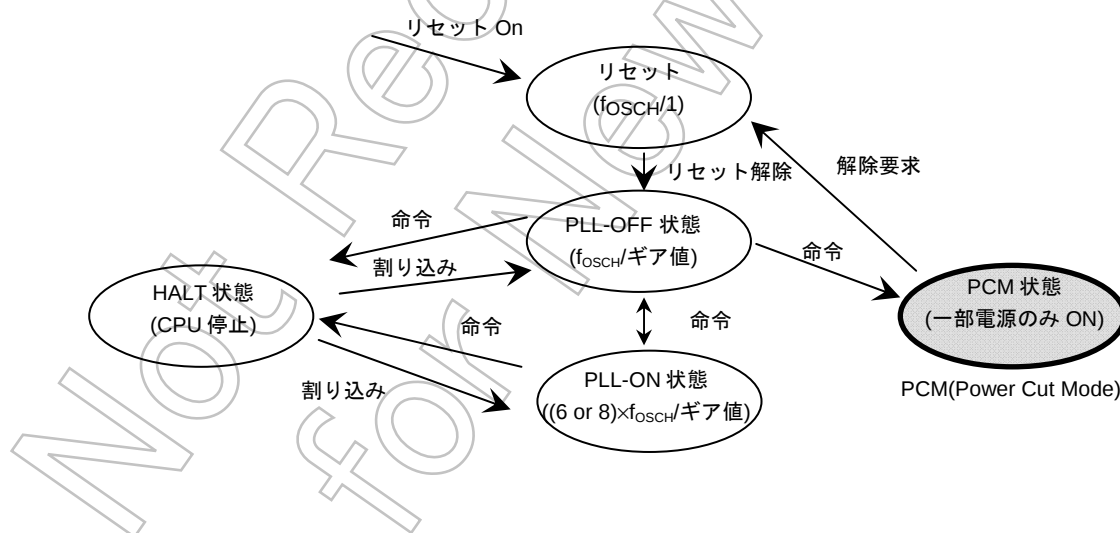


図 3.24.1 状態遷移図

3.24.1 電源供給システム

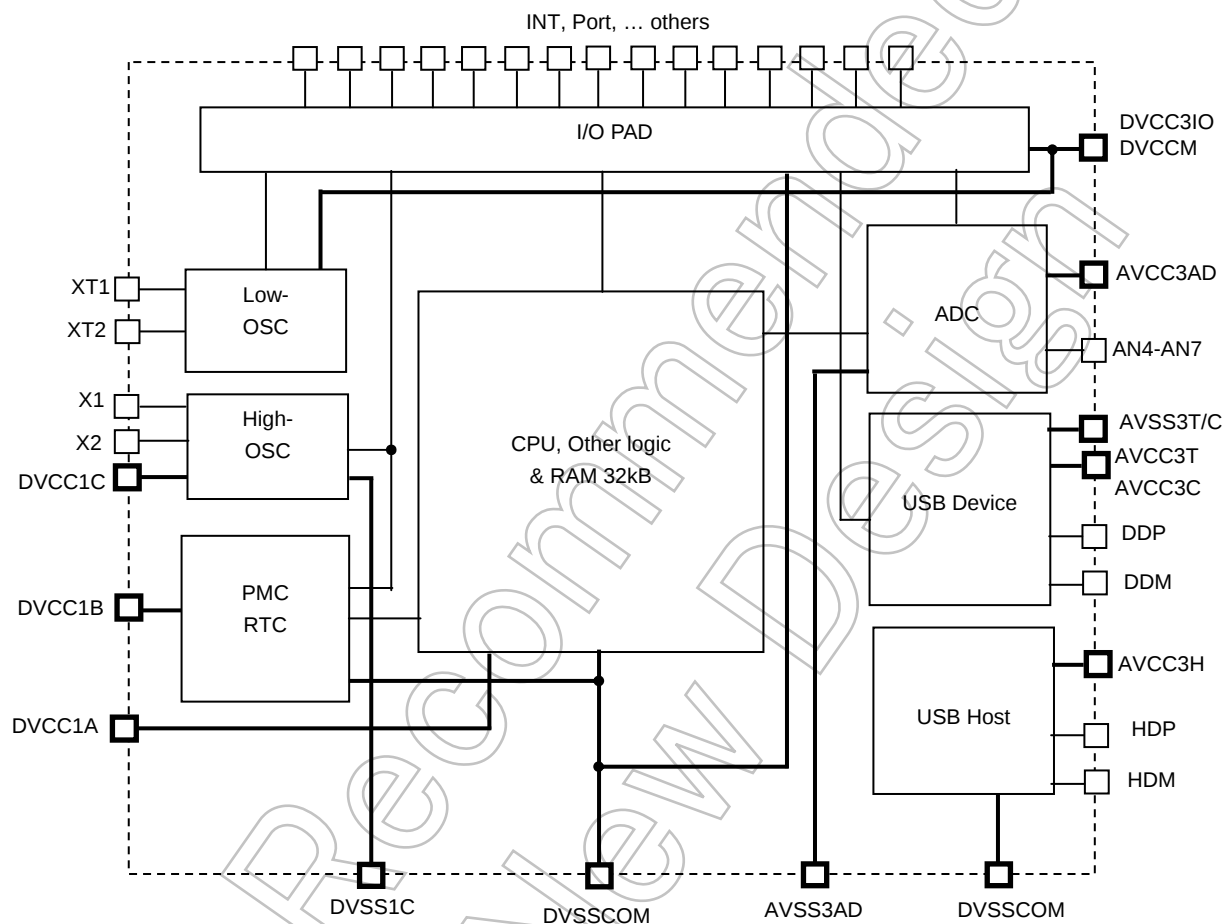


図 3.24.2 電源供給システム

3.24.2 接続・システム応用例

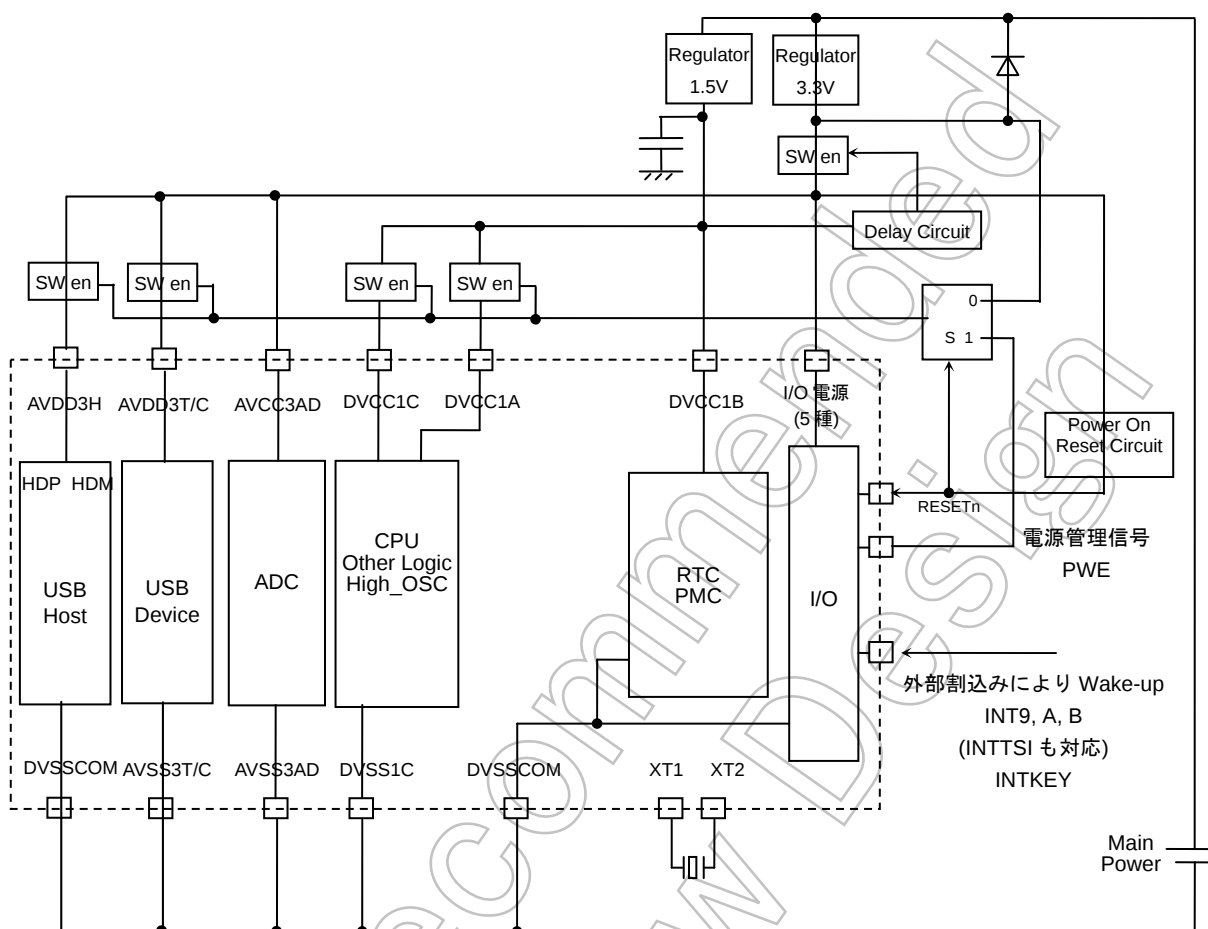


図 3.24.3 電源供給システム用 外部回路例

図 3.24.3は本システムを利用した外部回路例です。

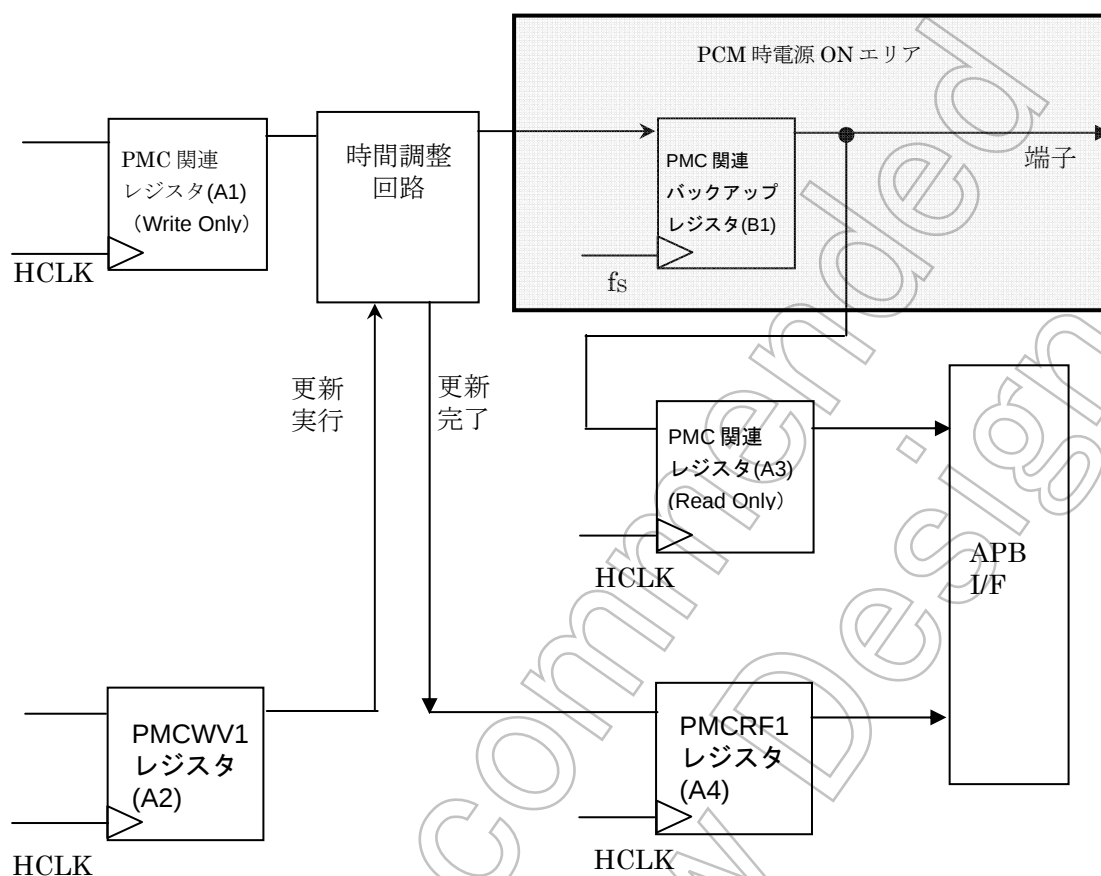
電源管理端子(PWE)は、電源をカットする回路や端子の電源回路を制御します。システムリセット状態を含む通常時には“High”を出力し、すべてのブロックに電源を供給します。

また、Power Cut モード時には“Low”を出力し、CPU を含む大半の回路、高速発振器、USB の電源を OFF し、消費電流を低減します。

Power Cut モードは、Wake-Up 要求により解除されます。PWE 端子は再び“High”を出力し、内部回路に電源を供給します。

3.24.3 PMCレジスタの構成

Power Cut Mode 時、動作電流を削減するため、PMC 関連レジスタの構成を下記図に示します。
対象レジスタは表 3. 24.3(Address(0xF002_0200~0xF002_041C))を参照してください。



電源遮断対応のために PMC の設定レジスタは電源遮断領域部分(A1, A2, A3, A4)と電源常時通電領域部分(B1)にそれぞれレジスタを持っています。そのため、バックアップレジスタに設定値を更新するためには更新レジスタ(PMCWV1)をライトする必要があります。

3.24.4 動作説明

PCM 状態への状態遷移フローを以下に説明します。

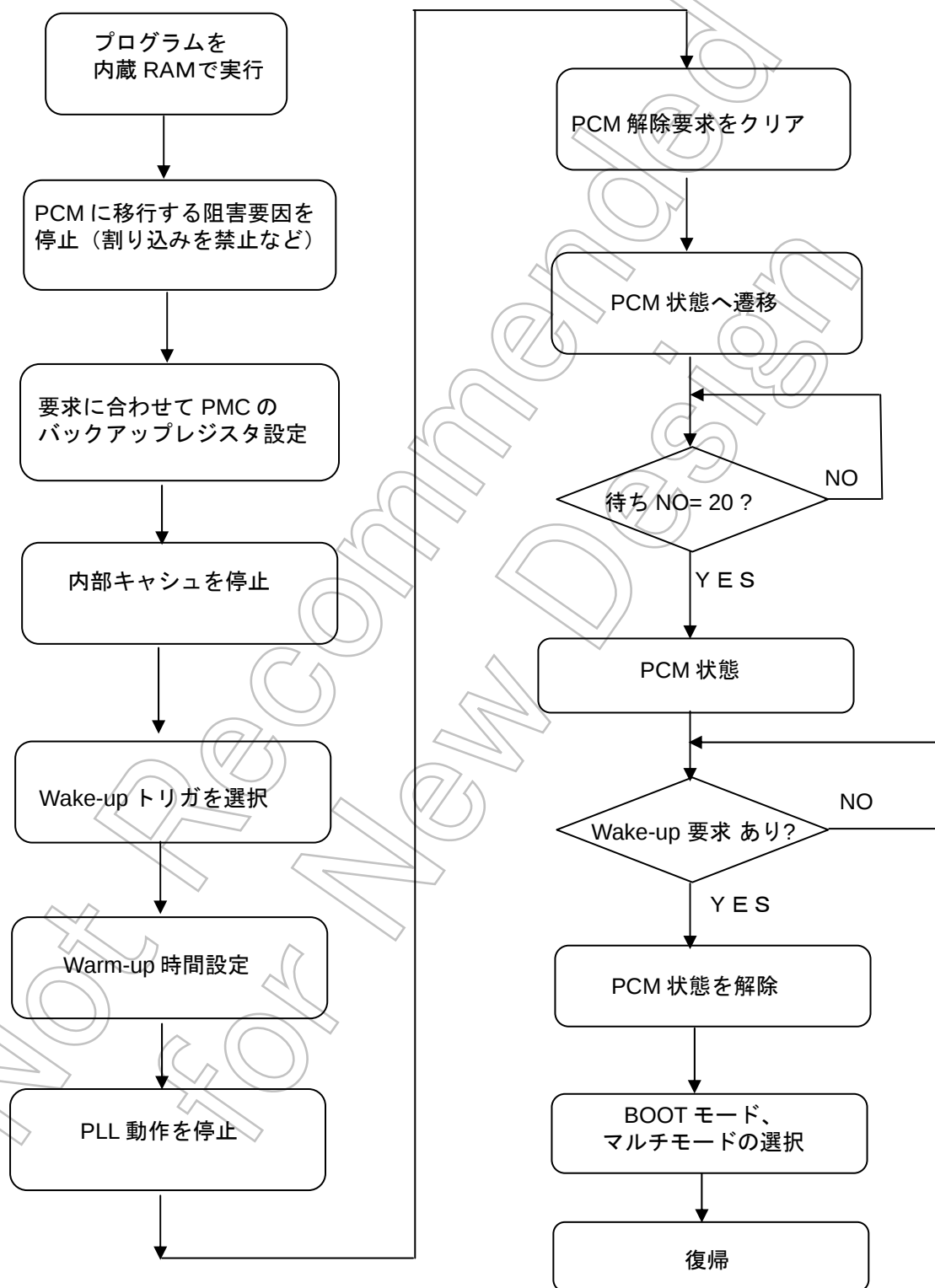


図 3.24.4 PCM 状態へ遷移のフローチャート

3.24.4.1 PCM状態への移行

PCM 状態へ移行させる場合には、CPU を含む内部回路の電源をカットするため、PCM 状態から起動した後の準備、PCM 状態中の外部端子の定義および、正常なモード切り替えを行うために以下のフローに従ってください。

1. 実行順序

(1) プログラムの実行エリア

必ず内蔵 RAM からプログラムを実行してください。

(2) 障害要因の排除

PCM モードへ移行する際に、その動作を障害する可能性のある要因をすべて停止します。

a. 割り込みの禁止

b. ウォッチドッグタイマの停止(初期値は停止状態です)

c. AD コンバータ動作の停止

d. システムの DMA 動作の停止

• LCDC の停止

• SDRAM の Auto リフレッシュの停止(セルフリフレッシュモードへの移行)

• DMA 転送の停止

(3) 端子の設定

バックアップレジスタで端子の状態を固定してください。

PCM 状態では、ポートの状態を管理するレジスタが、PMC 回路内のバックアップレジスタに切り替わります。

PC2 は PWE に機能設定してください。

SDRAM制御のCKE端子のみ、システムリセット後の端子状態とPCM状態解除時の端子状態を変えることが可能です。詳細は後述 0を参照願います。

(4) Wake-up の設定

PCM 状態を解除(Wake-up)する外部端子を設定します。

各々の端子ごとに、許可レジスタ、エッジ選択レジスタおよび、解除割り込みフラグレジスタが設定可能です。

内部 RTC、外部 Key および、外部割り込みを選択可能です。(不要な割り込みのマスク処理が可能です)。

PD6 を TSI の割り込み INTA として利用する場合は、デバウンス回路は Disable 状態に設定することが必要です。詳細は TSI の章を参照願います。

(5) OFD 動作の停止

OFD 機能を停止させてください。

(6) PLL 動作の停止

高周波クロックを、f_{osch}に設定し、PLL 動作を停止させてください。

(7) Warm-up 時間の設定 : PMCCTL<WUTM1:0>

Wake-up 要求後、約 1.5 XT1(48μs)後に外部 PWE 信号が“0”から“1”に変化します。この後 Warm-up タイマで設定された時間をカウントアップし、さらに約 1 XT1(32μs)後に、内部のリセット信号が解除される仕組みとなっています。使用する電源のレスポンスや、セット上の条件によって、電源が安定する時間は異なりますので、電源が安定する時間を考慮して Warm-up 時間を決定してください。(15.625ms~125ms から選択可能です。)

(8) 内部キャッシュメモリの Disable 設定

(9) PMC 回路内の解除要求信号のクリア

PCM 状態に移行する前に、PMC 回路内部の PCM 解除要求信号をクリアしておく必要があります。

(10) PMC 関連ライトレジスタを設定し、ライト更新レジスタ(PMCWV1)で関連ビットを有効にしてください。

(11) PMCWV1 レジスタで設定した Write レジスタの設定値は PMCRF1 レジスタをリードすることにより、バックアップレジスタに更新されたかを確認できます。

(12) PCM 状態への移行(PMCCTL<PCM_ON>= “1”)

PMCCTL レジスタの PCM_ON 以外のビットの設定を変更<PMCPWE、WMTM1、WMTM0>と、<PCM_ON>の 0y0→0y1 ライトを同時に行うことは出来ません。<PMCPWE、WMTM1、WMTM0>の設定変更は、<PCM_ON>ビットが“0y0”の状態に変更し、<PCM_ON>に“0y1”をライトする際には、<PMCPWE、WMTM1、WMTM0>は直前の状態と同じ値をライトしてください。

(13) PCM 状態への移行へは約 7XT1(224μs)かかります。その期間はダミー命令の挿入が必要となります。

注) Warm-up 後に起動するプログラム

Wake-up 後は、システムリセットがアサートされた場合と同様に、外部 AM0/1 端子の設定に応じて、内蔵ブート ROM または、外部メモリ(SMCCS0n)のどちらかが選択されて、プログラムが起動します。

起動プログラムの初期ルーチン内で、PMC 回路内の、PMCCTL<PCM_ON>のフラグを確認し、システムリセット起動か、PCM 状態からの Wake-up かを判断することが出来ます。

- ・ PCM 状態からの Wake-up : PMCCTL<PCM_ON>=0y1
- ・ システムリセット起動 : PMCCTL<PCM_ON>=0y0

起動プログラムの中で、あらかじめ上記のフラグを確認し、プログラムを分岐する仕組みを準備しておく必要があります。

3.24.4.2 PCM状態からのWake-up

PCM 状態からの Wake-up は、外部割り込みにて実行されます。

起動プログラムの初期ルーチン内で、PMC 回路内の、PMCCTL<PCM_ON>のフラグを確認し、システムリセット起動か、PCM 状態からの Wake-up かを判断することが出来ます。

PCM 状態から Wake-up 後は、外部 AM0/1 端子の設定に応じて、内蔵ブート ROM または、外部メモリ (SMCCS0n)のどちらかが選択されて、プログラムが起動します。

(DVCC1A が遮断された状態でリセットからの復帰は禁止です。事前に DVCC1A への電源を供給し、十分に電圧が安定してからリセットしてください)。対象の割り込みは、RTC 割り込み、INT9, INTB, INTA(INTTSI)および KI0~KI3 割り込みです。詳細は後述の「PCM 状態解除端子」を確認してください。

Wake-up 要求を受付けると PWE 端子を "0" → "1"とし、電源 OFF してあった各ブロックに電源を供給します。

PMCCTL<WUTM1:0>で設定された時間 Warm-up を行った後、自動的に HOT_RESET を解除します。

PCM 状態時、外部端子は PMC 回路内に設定された状態が保持されていますが、内部 HOT_RESET の解除とほぼ同時に、「DMCCKE」端子以外⁴⁾は、外部システムリセットと同様の状態になります。

起動プログラムの初期ルーチン内で、PMC 回路内の<PCM_ON>を確認し、システムリセット起動か、PCM 状態からの Wake-up かを判断することが出来ます。

注1) PMC 回路内の PMCCTL<PCM_ON>は、PCM 状態からの Wake-up では初期化されません。Wake-up 後、レジスタ PMCRES<RES_PCMON>を使用してクリアしてください。

PCM 状態を Wake-up した要因を確認するために、PMC 回路内には確認用フラグ(BxxRINT)が準備されています。どの要因で PCM 状態を Wake-up したかを確認することが可能です。

注) システムリセット後 DMCCKE 端子は必ず"H"状態から開始されます。よって PCM 状態に移行する前に SDRAM をセルフリフレッシュ状態に設定した場合、PCM 状態解除によって DMCCKE 端子が"H"状態になるとセルフリフレッシュが解除されてしまいます。そこで、SDRAM 制御の DMCCKE 端子のみシステムリセット後の端子状態と PCM 状態解除時の端子状態を変えることが可能です。PMC 回路内のレジスタに端子状態を設定します。

	DMCCKE 端子の状態
システムリセット後	"H"状態
PCM 解除後	"L"状態にする場合 : DMCCKECTL<DMCCKEHL>= 0y1 BSJOE<BSJOE6>= 0y1 BSJDATA<BSJDATA6>= 0y0

3.24.4.3 Power Cut Mode移行およびWake-upの状態遷移

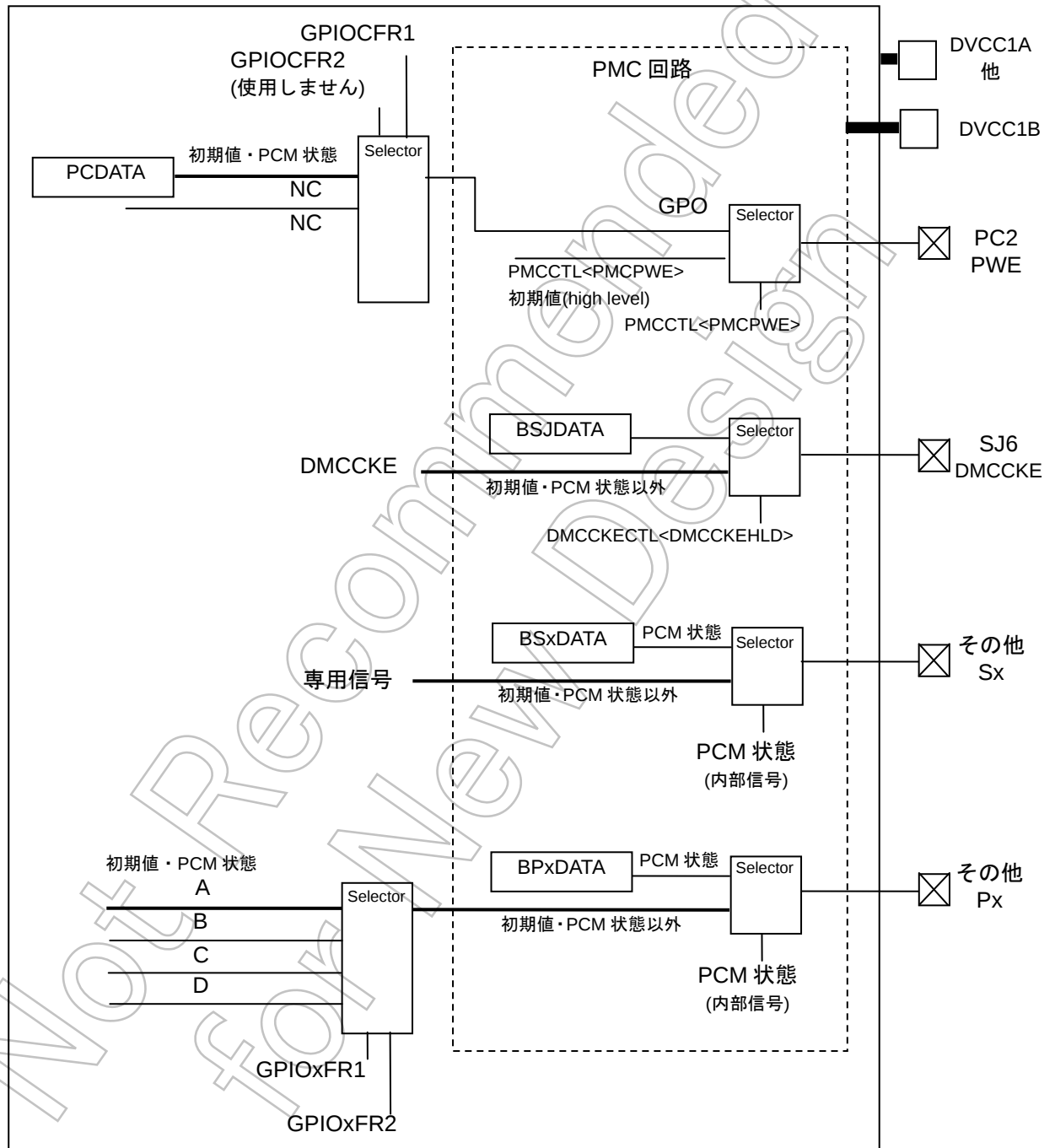
Power Cut Mode 移行および Wake-up の状態遷移

	移行中	PCM 状態 Power Cut	解除中	通常状態
DVCC1A	電源 ON	電源 OFF	電源 ON	電源 ON
CPU・RAM 等	RESET	停止	RESET→再起動	動作
Melody / Alarm 回路	RESET	停止	RESET→再起動	動作
DVCC1B	電源 ON	電源 ON	電源 ON	電源 ON
PMC 回路	動作	動作	動作	動作
RTC 回路	動作	動作	動作	動作
DVCC1C	電源 ON	電源 OFF	電源 ON	電源 ON
高周波発振	動作	停止	再起動	動作
DVCC3IO	電源 ON	電源 ON	電源 ON	電源 ON
PWE 端子	“H” 出力	“L” 出力	“L→H” 出力	“H” 出力
低周波発振器	動作	動作	動作	動作
リセット、AM 端子など 基本端子	動作	動作	動作	動作
JTAG 端子	端子固定(ポート)	端子固定(PMC)	RESET→再起動	動作
NANDF 端子	端子固定(専用機能)	端子固定(PMC)	RESET→再起動	動作
KEYOUT 端子	端子固定(ポート)	端子固定(PMC)	RESET→再起動	動作
KEYIN 端子	入力端子	入力端子	入力端子	入力端子
I ² C0 端子	端子固定(ポート)	端子固定(PMC)	RESET→再起動	動作
INT 端子	端子固定(ポート)	端子固定(PMC)	RESET→再起動	動作
UART0/1 端子	端子固定(ポート)	端子固定(PMC)	RESET→再起動	動作
SPIC0 端子	端子固定(ポート)	端子固定(PMC)	RESET→再起動	動作
I ² S0/1 端子	端子固定(ポート)	端子固定(PMC)	RESET→再起動	動作
LCDC 端子	端子固定(専用機能)	端子固定(PMC)	RESET→再起動	動作
DVCCM	電源 ON	電源 ON	電源 ON	電源 ON
メモリ制御関連端子	端子固定(専用機能)	端子固定(PMC)	RESET→再起動	動作
DMCKKE 端子	端子固定(専用機能)	端子固定(PMC)	端子固定(PMC)	ソフトウェア処理→動作
AVCC3AD	電源 ON	電源 ON	電源 ON	電源 ON
AN4, AN5, AN7 端子	入力端子	入力端子	入力端子	入力端子
AN6 端子	入力端子	Pull-down (PMC)	RESET→再起動	動作
TSI 端子	端子固定(ポート)	端子固定(PMC)	RESET→再起動	動作
DVCC3T/C	電源 ON	電源 ON	電源 ON	電源 ON
USB Device 端子	動作	停止	RESET→再起動	動作
AVDD3T/C	電源 ON	電源 OFF	電源 ON	電源 ON
USB Host 端子	動作	停止	RESET→再起動	動作

注 1) PCM 状態を Wake-up した要因を確認するために、PMC 回路内には確認用フラグ (BxxRINT) が準備されています。どの要因で PCM 状態を Wake-up したかを確認することが可能です。

注 2) PCM 状態から Wake-up 後は、PMCCTL<PCM_ON>は“1”のまま保持されています。再度 Power Cut 状態にする場合は、レジスタ PMCRESC<RES_PCMON>を使用してクリアしてください。

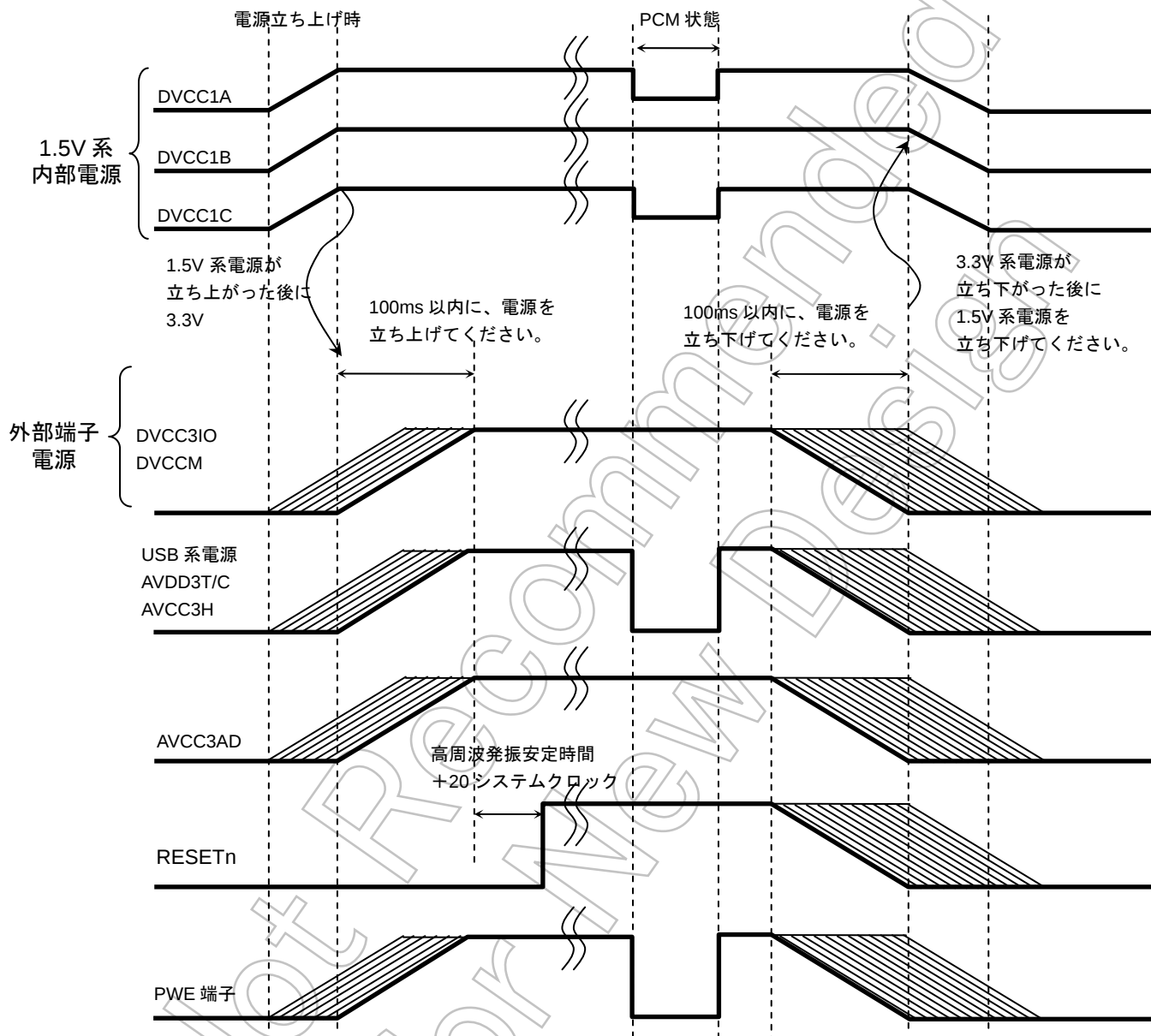
TMPA901CMXBG



3.24.5 動作上の注意点

電源供給/停止に関する順序(初期電源投入/完全電源停止)

電源の初期電源の供給/完全電源の停止は以下に示すように、電源投入時には内部電源が先に電源投入され、電源停止時は内部電源が後に電源停止されることが必要です。



注 1) 内部 1.5V 系と外部端子電源を、同時に立ち上げ/立ち下げることも可能ですが、その場合、瞬間的に外部端子が不安定の状態になる可能性がありますので、周辺の LSI に接続の機器に影響を与える可能性がある場合には、上図の太線で示すように、内部 1.5V 系電源が確定・安定している間に、外部電源の立ち上げ/立ち下げを行ってください。

注 2) 立ち上げ時は 1.5V 系の電源より早く 3.3V 系電源を立ち上げないようにしてください。また、立ち下げ時は 1.5V 系の電源より後に 3.3V 系電源を立ち下げないようにしてください。

3.24.6 PCM状態解除端子

Power Cut Mode を、割り込み要求で解除します。
以下に、解除可能な外部端子表を載せます。
注) 下記以外に、内蔵 RTC からの割り込みでも Power Cut Mode を解除できます。

PORT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PA					KI3	KI2	KI1	KI0
PC	I2C0DA INT9							
PD	PY INTB	PX INTA(INTTSI)						

Not Recommended for New Design

3.24.7 レジスタの説明

レジスタのリストを以下に示します。

表 3.24.1 レジスタ一覧 (1 / 3)

base アドレス = 0xF002_0000

Register Name	Address (base+)	Description
BPADATA	0x0900	PortA Data Set Register when Power Cut Mode
BPBDATA	0x0904	PortB Data Set Register when Power Cut Mode
BPCDATA	0x0908	PortC Data Set Register when Power Cut Mode
BPDDATA	0x090C	PortD Data Set Register when Power Cut Mode
–	0x0914	Reserved
–	0x0918	Reserved
–	0x0924	Reserved
–	0x0928	Reserved
–	0x092C	Reserved
–	0x0930	Reserved
BPNDATA	0x0934	PortN Data Set Register when Power Cut Mode
–	0x0944	Reserved
BPTDATA	0x094C	PortT Data Set Register when Power Cut Mode
BPUDATA	0x0950	PortU Data Set Register when Power Cut Mode
BPVDATA	0x0954	PortV Data Set Register when Power Cut Mode
–	0x0B80	Reserved
BPBOE	0x0B84	PortB Data Out Enable Control when Power Cut Mode
BPCOE	0x0B88	PortC Data Out Enable Control when Power Cut Mode
BPDOE	0x0B8C	PortD Data Out Enable Control when Power Cut Mode
–	0x0B90	Reserved
–	0x0B94	Reserved
–	0x0B98	Reserved
–	0x0BA4	Reserved
–	0x0BA8	Reserved
–	0x0BAC	Reserved
–	0x0BB0	Reserved
BPNOE	0x0BB4	PortN Data Out Enable Control when Power Cut Mode
–	0x0BC4	Reserved
BPTOE	0x0BCC	PortT Data Out Enable Control when Power Cut Mode
BPUOE	0x0BD0	PortU Data Out Enable Control when Power Cut Mode
BPVOE	0x0BD4	PortV Data Out Enable Control when Power Cut Mode

表 3.24.2 レジスタ一覧 (2 / 3)

base アドレス= 0xF002_0000

Register Name	Address (base+)	Description
BSADATA	0x0800	SA Data Set Register when Power Cut Mode
BSBDATA	0x0804	SB Data Set Register when Power Cut Mode
–	0x0808	Reserved
–	0x080C	Reserved
BSADATA	0x0810	SE Data Set Register when Power Cut Mode
BSFDATA	0x0814	SF Data Set Register when Power Cut Mode
BSGDATA	0x0818	SG Data Set Register when Power Cut Mode
BSHDATA	0x081C	SH Data Set Register when Power Cut Mode
BSJDATA	0x0824	SJ Data Set Register when Power Cut Mode
BSKDATA	0x0828	SK Data Set Register when Power Cut Mode
BSLDATA	0x082C	SL Data Set Register when Power Cut Mode
–	0x084C	Reserved
–	0x0850	Reserved
BSAOE	0x0A80	SA Data Out Enable Control when Power Cut Mode
BSBOE	0x0A84	SB Data Out Enable Control when Power Cut Mode
–	0x0A88	Reserved
–	0x0A8C	Reserved
BSEOE	0x0A90	SE Data Out Enable Control when Power Cut Mode
BSFOE	0x0A94	SF Data Out Enable Control when Power Cut Mode
BSGOE	0x0A98	SG Data Out Enable Control when Power Cut Mode
BSHOE	0x0A9C	SH Data Out Enable Control when Power Cut Mode
BSJOE	0x0AA4	SJ Data Out Enable Control when Power Cut Mode
BSKOE	0x0AA8	SK Data Out Enable Control when Power Cut Mode
BSLOE	0x0AAC	SL Data Out Enable Control when Power Cut Mode
–	0x0ACC	Reserved
–	0x0AD0	Reserved
BPAIE	0x0D80	PORT A WakeUp 入力 Enable
BPCIE	0x0D88	PORT C WakeUp 入力 Enable
BPDIE	0x0D8C	PORT D WakeUp 入力 Enable
–	0x0D94	Reserved
–	0x0DB4	Reserved
–	0x0DC4	Reserved

表 3.24.3 レジスター一覧 (3 /3)

base アドレス= 0xF002_0000

Register Name	Address (base+)	Description
BPARELE	0x0200	PortA Enable Register of Wake-up trigger from Power Cut Mode
BPDRELE	0x0204	PortD Enable Register of Wake-up trigger from Power Cut Mode
BRTRELE	0x0208	RTC Request Enable Register of Wake-up trigger from Power Cut Mode
BPXRELE	0x020C	Others Port Enable Register of Wake-up trigger from Power Cut Mode
BPAEDGE	0x0220	PortA Selection Register of Wake-up trigger Edge from Power Cut Mode
BPDEEDGE	0x0224	PortD Selection Register of Wake-up trigger Edge from Power Cut Mode
BPXEDGE	0x022C	Others Ports Selection Register of Wake-up trigger Edge from Power Cut Mode
BADRINT	0x0240	PortA Wake-up Interrupt status Register
BPDRINT	0x0244	PortD Wake-up Interrupt status Register
BRTRINT	0x0248	RTC Wake-up Interrupt status Register
BPXRINT	0x024C	Others Ports Wake-up Interrupt status Register
PMCDRV	0x0260	External Port Driverbility control Register
DMCCKECTL	0x0280	DMCCKE pin setting Register (PCM mode)
PMCCTL	0x0300	Power Management Circuit Control Register
PMCWV1	0x0400	Renew Control for PMC Registers
-	0x0408	Reserved
PMCRES	0x041C	<PCM_ON> Flag Clear Register for PMCCTL_R<PMC_ON>

表 3.24.4 バックアップレジスタ出力データレジスター一覧 1

レジスタ Address	対応 ポート	レジスタ 名称	type	bit7	bit6	bit5	bit4	bit3	bit2	bit1	0bit
0xF002_0900	PA	BPADATA 注 1)	W	—	—	—	—	BPADATA3	BPADATA2	BPADATA1	BPADATA0
0xF002_0904	PB	BPBDATA	W	—	—	—	—	BPBDATA3	BPBDATA2	BPBDATA1	BPBDATA0
0xF002_0908	PC	BPCDATA	W	BPCDATA7	BPCDATA6	—	BPCDATA4	BPCDATA3	—	—	—
0xF002_090C	PD	BPDDATA	W	Read as undefined. Write as zero.							
0xF002_0914	-	Reserved	W	—	—	—	—	—	—	—	—
0xF002_0918	-	Reserved	W	—	—	—	—	—	—	—	—
0xF002_0924	-	Reserved	W	—	—	—	—	—	—	—	—
0xF002_0928	-	Reserved	W	—	—	—	—	—	—	—	—
0xF002_092C	-	Reserved	W	—	—	—	—	—	—	—	—
0xF002_0930	-	Reserved	W	—	—	—	—	—	—	—	—
0xF002_0934	PN	BPNDATA	W	—	—	—	—	—	—	BPNDATA1	BPNDATA0
0xF002_0944	-	Reserved	W	—	—	—	—	—	—	—	—
0xF002_094C	PT	BPTDATA	W	BPTDATA7	BPTDATA6	BPTDATA5	BPTDATA4	BPTDATA3	BPTDATA2	BPTDATA1	BPTDATA0
0xF002_0950	PU	BPUDATA	W	BPUDATA7	BPUDATA6	BPUDATA5	BPUDATA4	BPUDATA3	BPUDATA2	BPUDATA1	BPUDATA0
0xF002_0954	PV	BPVDATA	W	BPVDATA7	BPVDATA6	BPVDATA5	BPVDATA4	BPVDATA3	BPVDATA2	BPVDATA1	BPVDATA0

PCM 状態における出力データを決めるレジスタです。リセット解除後は"0"、ホットリセット時はデータ(レジスタの設定値)を保持します。

レジスタの各ビット=0y0: "L"出力

レジスタの各ビット=0y1: "H"出力

注 1) PCM 状態では BPADATA 初期値=0x00 となり,ポート A の内部 PULL-UP 回路が OFF されます。ポート A の PULL-UP 状態を維持するには PCM 状態になる前に BPADATA に 0xFF を設定してください。

表 3.24.5 バックアップレジスタの出カイネーブルレジスタ一覧 1

レジスタ Address	対応 ポート	レジスタ 名称	type	bit7	bit6	bit5	bit4	bit3	bit2	bit1	0bit
0xF002_0B80	PA	Reserved	—	—				Read as undefined. Write as zero.			
0xF002_0B84	PB	BPBOE	W	—				BPBOE3	BPBOE2	BPBOE1	BPBOE0
0xF002_0B88	PC	BPCOE	W	BPCOE7	BPCOE6	—	BPCOE4	BPCOE3	—	—	—
0xF002_0B8C	PD	BPDOE	W	BPDOE7	BPDOE6	BPDOE5	BPDOE4	—			
0xF002_0B94	-	Reserved	W	—				—			
0xF002_0B98	-	Reserved	W	—				—			
0xF002_0BA4	-	Reserved	W	—				—			
0xF002_0BA8	-	Reserved	W	—				—			
0xF002_0BAC	-	Reserved	W	—				—			
0xF002_0BB0	-	Reserved	W	—				—			
0xF002_0BB4	PN	BPNOE	W	—				—		BPNOE1	BPNOE0
0xF002_0BC4	-	Reserved	W	—				—			
0xF002_0BCC	PT	BPTOE	W	BPTOE7	BPTOE6	BPTOE5	BPTOE4	BPTOE3	BPTOE2	BPTOE1	BPTOE0
0xF002_0BD0	PU	BPUOE	W	BPUOE7	BPUOE6	BPUOE5	BPUOE4	BPUOE3	BPUOE2	BPUOE1	BPUOE0
0xF002_0BD4	PV	BPVOE	W	BPVOE7	BPVOE6	BPVOE5	BPVOE4	BPVOE3	BPVOE2	BPVOE1	BPVOE0

PCM 状態におけるデータ出力のイネーブルレジスタです。リセット解除後は"0"、ホットリセット時はデータ(レジスタの設定値)を保持します。

レジスタの各ビット=0y0: 出力 Disable

レジスタの各ビット=0y1: “出力 Enable

注) PORT D をタッチスクリーン機能として使用した場合、Power Cut Mode モード時の BPDOE および BPDDATA の設定は以下となります。

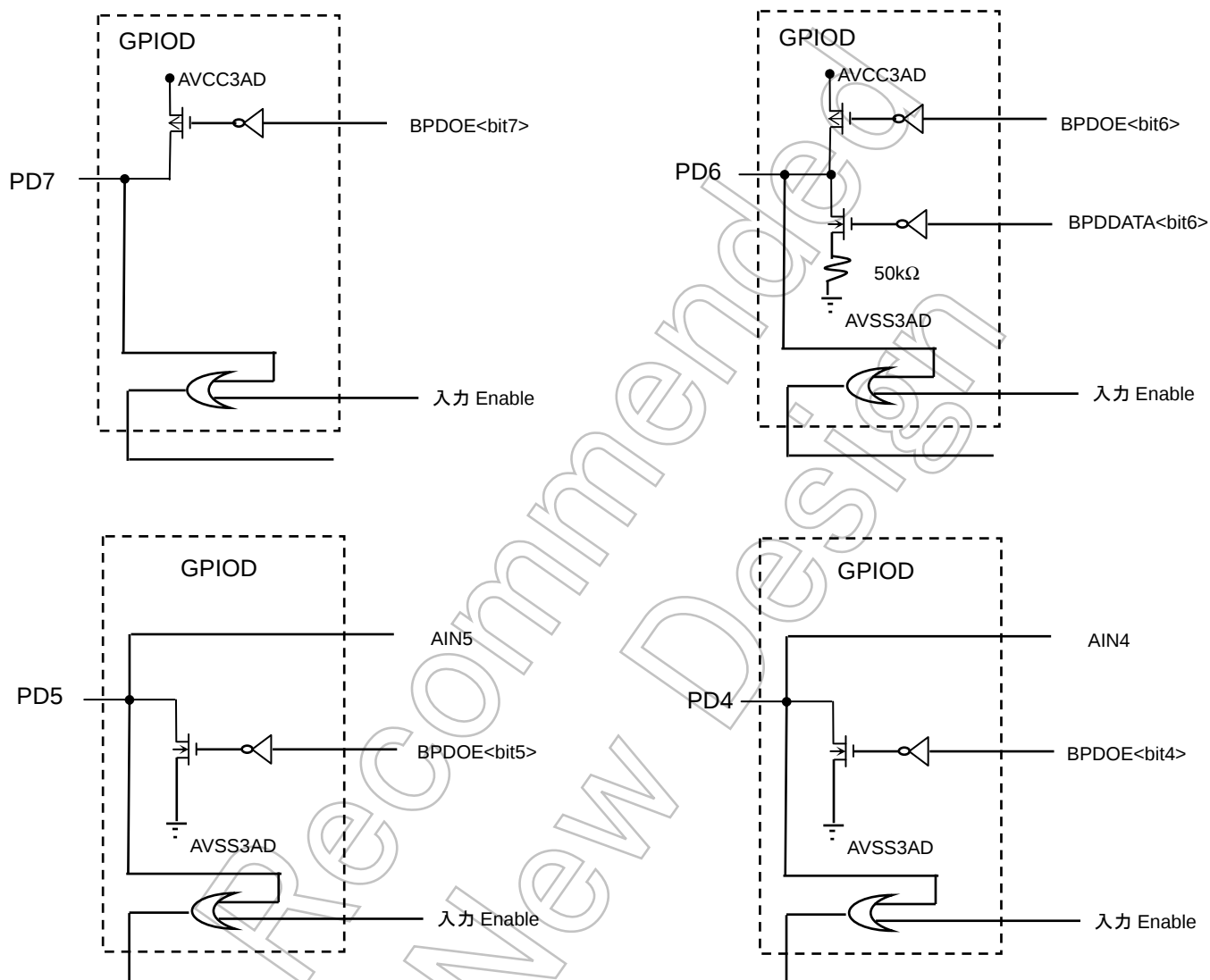


表 3.24.6 バックアップ時の出力データ設定レジスタ一覧 2

レジスタ Address	対応 ポート	レジスタ 名称	type	bit7	bit6	bit5	bit4	bit3	bit2	bit1	Obit
0xF002_0800	SA	BSADATA	W	BSADATA7	BSADATA6	BSADATA5	BSADATA4	BSADATA3	BSADATA2	BSADATA1	BSADATA0
0xF002_0804	SB	BSBDATA	W	BSBDATA7	BSBDATA6	BSBDATA5	BSBDATA4	BSBDATA3	BSBDATA2	BSBDATA1	BSBDATA0
0xF002_0808	-	Reserved	W								
0xF002_080C	-	Reserved	W								
0xF002_0810	SE	BSEDATA	W	BSEDATA7	BSEDATA6	BSEDATA5	BSEDATA4	BSEDATA3	BSEDATA2	BSEDATA1	BSEDATA0
0xF002_0814	SF	BSFDATA	W	BSFDATA7	BSFDATA6	BSFDATA5	BSFDATA4	BSFDATA3	BSFDATA2	BSFDATA1	BSFDATA0
0xF002_0818	SG	BSGDATA	W	BSGDATA7	BSGDATA6	BSGDATA5	BSGDATA4	BSGDATA3	BSGDATA2	BSGDATA1	BSGDATA0
0xF002_081C	SH	BSHDATA	W	BSHDATA7	—	—	BSHDATA4	BSHDATA3	BSHDATA2	—	—
0xF002_0824	SJ	BSJDATA	W	—	BSJDATA6	BSJDATA5	BSJDATA4	BSJDATA3	BSJDATA2	BSJDATA1	BSJDATA0
0xF002_0828	SK	BSKDATA	W	—	—	BSKDATA5	BSKDATA4	—	—	BSKDATA1	BSKDATA0
0xF002_082C	SL	BSLDATA	W	—	注	BSLDATA5	BSLDATA4	—	BSLDATA2	BSLDATA1	BSLDATA0
0xF002_084C	-	Reserved	W								
0xF002_0850	-	Reserved	W								

PCM 状態におけるデータ出力データ決めるレジスタです。リセット解除後は"0"、ホットリセット時はデータ(レジスタの設定値)を保持します。

レジスタの各ビット=0y0: "L"出力

レジスタの各ビット=0y1: "H"出力

注) Read as undefined.

表 3.24.7 バックアップレジスタの出力イネーブルレジスタ一覧 2

レジスタ Address	対応 ポート	レジスタ 名称	type	bit7	bit6	bit5	bit4	bit3	bit2	bit1	0bit
0xF002_0A80	SA	BSAOE	W	BSAOE7	BSAOE6	BSAOE5	BSAOE4	BSAOE3	BSAOE2	BSAOE1	BSAOE0
0xF002_0A84	SB	BSBOE	W	BSBOE7	BSBOE6	BSBOE5	BSBOE4	BSBOE3	BSBOE2	BSBOE1	BSBOE0
0xF002_0A88	-	Reserved	W								
0xF002_0A8C	-	Reserved	W								
0xF002_0A90	SE	BSEOE	W	BSEOE7	BSEOE6	BSEOE5	BSEOE4	BSEOE3	BSEOE2	BSEOE1	BSEOE0
0xF002_0A94	SF	BSFOE	W	BSFOE7	BSFOE6	BSFOE5	BSFOE4	BSFOE3	BSFOE2	BSFOE1	BSFOE0
0xF002_0A98	SG	BSGOE	W	BSGOE7	BSGOE6	BSGOE5	BSGOE4	BSGOE3	BSGOE2	BSGOE1	BSGOE0
0xF002_0A9C	SH	BSHOE	W	BSHOE7	—	—	BSHOE4	BSHOE3	BSHOE2	—	—
0xF002_0AA4	SJ	BSJOE	W	—	BSJOE6	BSJOE5	BSJOE4	BSJOE3	BSJOE2	BSJOE1	BSJOE0
0xF002_0AA8	SK	BSKOE	W			BSKOE5	BSKOE4			BSKOE1	BSKOE0
0xF002_0AAC	SL	BSLOE	W	—	注	BSLOE5	BSLOE4	—	BSLOE2	BSLOE1	BSLOE0
0xF002_0ACC	-	Reserved	W								
0xF002_0AD0	-	Reserved	W								

PCM 状態におけるデータ出力のイネーブルレジスタです。リセット解除後は"0"、ホットリセット時はデータ(レジスタの設定値)を保持します。

レジスタの各ビット=0y0: 出力 Disable

レジスタの各ビット=0y1: “出力 Enable

注)Read as undefined.

1. BPARELE register

Address = (0xF002_0000) + 0x0200

Bit	Bit Symbol	Type	Reset Value	Hot Reset Value	Description
[31:4]	–	–	Undefined	Undefined	Read as undefined. Write as zero.
[3]	BPARELE3	R/W	0y0	データ保持	PCM 解除要求許可 0y0: disable 0y1: enable
[2]	BPARELE2	R/W	0y0	データ保持	PCM 解除要求許可 0y0: disable 0y1: enable
[1]	BPARELE1	R/W	0y0	データ保持	PCM 解除要求許可 0y0: disable 0y1: enable
[0]	BPARELE0	R/W	0y0	データ保持	PCM 解除要求許可 0y0: disable 0y1: enable

(個別説明)

a. <BPARELE[3:0]>

キー入力 KI[3:0]により、PCM 解除要求許可するレジスタです。

注) Enable を設定する場合、BPAIE[3:0]も同時に設定が必要です。

2. BPDRELE register

Address = (0xF002_0000) + 0x0204

Bit	Bit Symbol	Type	Reset Value	Hot Reset Value	Description
[31:8]	–	–	Undefined	Undefined	Read as undefined. Write as zero.
[7]	–	–	Undefined	Undefined	Read as undefined. Write as zero.
[6]	BPDRELE6	R/W	0y0	データ保持	PCM 解除要求許可 0y0: disable 0y1: enable
[5:0]	–	–	Undefined	Undefined	Read as undefined. Write as zero.

(個別説明)

a. <BPDRELE6>

INTA(INTTSD)により、PCM 解除要求許可するレジスタです。

注) BPDIE[6]と同時に設定が必要です。

3. BRTRELE register

Address = (0xF002_0000) + 0x0208

Bit	Bit Symbol	Type	Reset Value	Hot Reset Value	Description
[31:1]	–	–	Undefined	Undefined	Read as undefined. Write as zero.
[0]	BRTRELE0	R/W	0y0	データ保持	PCM 解除要求許可 0y0: disable 0y1: enable

(個別説明)

a. <BRTRELE0>

RTC により、PCM 解除要求許可するレジスタです。

4. BPXRELE register

Address = (0xF002_0000) + 0x020C

Bit	Bit Symbol	Type	Reset Value	Hot Reset Value	Description
[31:8]	–	–	Undefined	Undefined	Read as undefined. Write as zero.
[7:2]	Reserved	R/W	0y0	データ保持	Read as undefined. Write as zero.
[1]	PD7INTB	R/W	0y0	データ保持	PCM 解除要求許可 0y0: disable 0y1: enable
[0]	PC7INT9	R/W	0y0	データ保持	PCM 解除要求許可 0y0: disable 0y1: enable

(個別説明)

a. <PD7INTB>

PD7(INTB)により、PCM 解除要求許可するレジスタです。

注) BPDIE[7]と同時に設定が必要です。

b. <PC7INT9>

PC7(INT9)により、PCM 解除要求許可するレジスタです。

注) BPCIE[7]と同時に設定が必要です。

5. BPAEDGE register

Address = (0xF002_0000) + 0x0220

Bit	Bit Symbol	Type	Reset Value	Hot Reset Value	Description
[31:4]	–	–	Undefined	Undefined	Read as undefined. Write as zero.
[3]	BPAEDGE3	R/W	0y0	データ保持	PCM 解除要求受けエッジ選択 0y0: 立ち上がりエッジ選択 0y1: 立ち下がりエッジ選択
[2]	BPAEDGE2	R/W	0y0	データ保持	PCM 解除要求受けエッジ選択 0y0: 立ち上がりエッジ選択 0y1: 立ち下がりエッジ選択
[1]	BPAEDGE1	R/W	0y0	データ保持	PCM 解除要求受けエッジ選択 0y0: 立ち上がりエッジ選択 0y1: 立ち下がりエッジ選択
[0]	BPAEDGE0	R/W	0y0	データ保持	PCM 解除要求受けエッジ選択 0y0: 立ち上がりエッジ選択 0y1: 立ち下がりエッジ選択

(個別説明)

a. <BPAEDGE[3:0]>

キー入力 KI[3:0]により、PCM 解除要求受けエッジを選択するレジスタです。

6. BPDEEDGE register

Address = (0xF002_0000) + 0x0224

Bit	Bit Symbol	Type	Reset Value	Hot Reset Value	Description
[31:8]	–	–	Undefined	Undefined	Read as undefined. Write as zero.
[7]	–	–	Undefined	Undefined	Read as undefined. Write as zero.
[6]	BPDEEDGE6	R/W	0y0	データ保持	PCM 解除要求受けエッジ選択 0y0: 立ち上がりエッジ選択 0y1: 立ち下がりエッジ選択
[5:0]	–	–	Undefined	Undefined	Read as undefined. Write as zero.

(個別説明)

a. <BPDEEDGE6>

INTA(INTTSD)により、PCM 解除要求受けエッジを選択するレジスタです。

7. BPXEDGE register

Address = (0xF002_0000) + (0x022C)

Bit	Bit Symbol	Type	Reset Value	Hot Reset Value	Description
[31:8]	–	–	Undefined	Undefined	Read as undefined. Write as zero.
[7:2]	Reserved	–	Undefined	Undefined	Read as undefined. Write as zero.
[1]	BPXEDGE8	R/W	0y0	データ保持	PCM 解除要求受けエッジ選択
[0]	BPXEDGE9				0y0: 立ち上がりエッジ選択 0y1: 立ち下がりエッジ選択

a. <BPXEDGE8>

INTB により、PCM 解除要求受けエッジを選択するレジスタです。

b. <BPXEDGE9>

INT9 により、PCM 解除要求受けエッジを選択するレジスタです。

8. BPARINT register

Address = (0xF002_0000) + 0x0240

Bit	Bit Symbol	Type	Reset Value	Hot Reset Value	Description
[31:4]	–	–	Undefined	Undefined	Read as undefined. Write as zero.
[3]	BPARINT3	R/W	Undefined	データ保持	PCM 解除割り込みステータス Write: 0y0: クリア 0y1: 禁止 Read: 0y0: 割り込み無し 0y1: 割り込み有り
[2]	BPARINT2	R/W	Undefined	データ保持	PCM 解除割り込みステータス Write: 0y0: クリア 0y1: 無効 Read: 0y0: 割り込み無し 0y1: 割り込み有り
[1]	BPARINT1	R/W	Undefined	データ保持	PCM 解除割り込みステータス Write: 0y0: クリア 0y1: 無効 Read: 0y0: 割り込み無し 0y1: 割り込み有り
[0]	BPARINT0	R/W	Undefined	データ保持	PCM 解除割り込みステータス Write: 0y0: クリア 0y1: 無効 Read: 0y0: 割り込み無し 0y1: 割り込み有り

(個別説明)

a. < BPARINT[3:0]>

KI[3:0]による PCM 解除のステータスフラグです。

Write:

0y0: クリア

0y1: Reserved

Read:

0y0: 割り込み無し

0y1: 割り込み有り

PCM 状態をその要因で解除したかを確認する場合は、PCM の設定前にレジスタに"0"を書き込んで、値をクリアしてください。

9. BPDRINT register

Address = (0xF002_0000) + 0x0244

Bit	Bit Symbol	Type	Reset Value	Hot Reset Value	Description
[31:7]	–	–	Undefined	Undefined	Read as undefined. Write as zero.
[6]	BPDRINT6	R/W	Undefined	データ保持	PCM 解除割り込みステータス Write: 0y0: クリア 0y1: 無効 Read: 0y0: 割り込み無し 0y1: 割り込み有り
[5:0]	–	–	Undefined	Undefined	Read as undefined. Write as zero.

(個別説明)

a. <BPDRINT6>

INTA(INTTSD)による PCM 解除ステータスレジスタです。

Write:

0y0: クリア

0y1: Reserved

Read:

0y0: 割り込み無し

0y1: 割り込み有り

PCM 状態をその要因で解除したかを確認する場合は、PCM の設定前にレジスタに"0"を書き込んで、値をクリアしてください。

10. BRTRINT register

Address = (0xF002_0000) + 0x0248

Bit	Bit Symbol	Type	Reset Value	Hot Reset Value	Description
[31:1]	–	–	Undefined	Undefined	Read as undefined. Write as zero.
[0]	BRTRINT0	R/W	Undefined	データ保持	PCM 解除割り込みステータス Write: 0y0: クリア 0y1: 無効 Read: 0y0: 割り込み無し 0y1: 割り込み有り

(個別説明)

a. <BRTRINT0>

RTC による PCM 解除ステータスレジスタです。

Write:

0y0: クリア

0y1: Reserved

Read:

0y0: 割り込み無し

0y1: 割り込み有り

PCM 状態をその要因で解除したかを確認する場合は、PCM の設定前にレジスタに 0 を書き込んで、値をクリアしてください。

11. BPXRINT register

Address = (0xF002_0000) + 0x024C

Bit	Bit Symbol	Type	Reset Value	Hot Reset Value	Description
[31:8]	–	–	Undefined	Undefined	Read as undefined. Write as zero.
[7:2]	Reserved	–	Undefined	Undefined	Read as undefined. Write as zero.
[1]	PD7INTB	R/W	Undefined	データ保持	PCM 解除割り込みステータス Write: 0y0: クリア 0y1: 無効 Read: 0y0: 割り込み無し 0y1: 割り込み有り
[0]	PC7INT9	R/W	Undefined	データ保持	PCM 解除割り込みステータス Write: 0y0: クリア 0y1: 無効 Read: 0y0: 割り込み無し 0y1: 割り込み有り

(個別説明)

a. < PD7INTB>

INTB による PCM 解除ステータスレジスタです。

b. < PC7INT9>

INT9 による PCM 解除ステータスレジスタです。

Write:

0y0: クリア

0y1: Reserved

Read:

0y0: 割り込み無し

0y1: 割り込み有り

PCM 状態をその要因で解除したかを確認する場合は、PCM の設定前にレジスタに"0"を書き込んで、値をクリアしてください。

12. PMCDRV register

Address = (0xF002_0000) + 0x0260

Bit	Bit Symbol	Type	Reset Value	Hot Reset Value	Description
[31:5]	–	–	Undefined	Undefined	Read as undefined. Write as zero.
[4]	DRV_LCD	R/W	0y1	データ保持	LDCD 関連ポートのドライブ能力設定 0y0: 12mA (1.8V~3.0V) 0y1: 6mA (3.0V~3.6V)
[3:2]	–	–	Undefined	Undefined	Read as undefined. Write as zero.
[1]	DRV_MEM1	R/W	0y0	データ保持	memory 関連ポートドライブ能力設定
[0]	DRV_MEM0	R/W	0y1	データ保持	0y00: Reserved 0y01: 1/2 (3.3V ± 0.3V) 0y10: Reserved 0y11: 1/1 (1.8V ± 0.1V)

(個別説明)

a. <DRV_LCD、DRV_MEM[1:0]>

LCD および memory 関連のポートのドライブ能力を変更することが可能です。使用する電圧範囲にあわせて変更してください。

13. DMCCCKCTL register

Address = (0xF002_0000) + 0x0280

Bit	Bit Symbol	Type	Reset Value	Hot Reset Value	Description
[31:1]	–	–	Undefined	Undefined	Read as undefined. Write as zero.
[0]	DMCCKEHLD	R/W	0y0	データ保持	ポート SJ6 の出力選択 0y0: DMCCKE 0y1: PMC のレジスタ設定値

(個別説明)

a. <DMCCKEHLD>

システムリセット後 DMCCKE 端子は必ず”H”状態から開始されます。よって PCM 状態に移行する前に SDRAM をセルフリフレッシュ状態に設定した場合、PCM 状態解除によって DMCCKE 端子が”H”状態になるとセルフリフレッシュが解除されてしまいます。そこで、SDRAM 制御の DMCCKE 端子のみシステムリセット後の端子状態と PCM 状態解除時の端子状態を変えることが可能です。PMC 回路内のレジスタに端子状態を設定します。

	DMCCKE 端子の状態
システムリセット後	”H”状態
PCM 解除後	”L”状態にする場合 : DMCCCKCTL<DMCCKEHLD>= 0y1 BSJOE<BSJOE6>= 0y1 BSJDATA<BSJDATA6>= 0y0

14. PMCCTL register

Address = (0xF002_0000) + 0x0300

Bit	Bit Symbol	Type	Reset Value	Hot Reset Value	Description
[31:8]	–	–	Undefined	Undefined	Read as undefined. Write as zero.
[7]	PCM_ON 注 1)	R/W	0y0	データ保持	Power Cut 機能 0y0: 無効 0y1: Enable
[6]	PMCPWE	R/W	0y1	データ保持	PWE 出力選択 0y0: Port 機能(PC2) 0y1: PMC レジスタ(PWE)出力
[5:3]	–	–	Undefined	Undefined	Read as undefined. Write as zero.
[2]	Reserved	–	Undefined	Undefined	Read as undefined. Write as zero.
[1]	WUTM1	R/W	0y0	データ保持	Warm-up 時間設定
[0]	WUTM0	R/W	0y0	データ保持	0y00: 2^9 (15.625ms) 0y01: 2^{10} (31.25ms) 0y10: 2^{11} (62.5ms) 0y11: 2^{12} (125ms)

(個別説明)

a. <PCM_ON>

<PCM_ON>に“1”をライトすることで、PCM 状態に移行します。

注2) Power Cut モードへの移行には、<PCM_ON>ビットに“1”をライトしますが、この時同一レジスタ内の他のビット、<PMCPWE>,<WMTM1:0>の設定変更を同時に行うことができません。<PMCPWE>,<WMTM1:0>の設定変更は、<PCM_ON>ビットが“0”の状態で行い、<PCM_ON>ビットに“1”をライトする際は、<PMCPWE>,<WMTM1:0>の値は直前にライトされたデータを設定する様にしてください。

注3) 本レジスタ<PCM_ON>に“0”をライトしてもクリアできません。クリアするためには、レジスタ PMCRES<RES_PCMON>を使用してください。

注 3) Wake-up 要求の割り込み後、約 $1.5 \times T1$ (48μs)後に外部 PWE 信号が“0”から“1”に変化します。この後 Warm-up タイマで設定された時間をカウントアップし、さらに約 $1 \times T1$ (32μs)後に、内部のリセット信号が解除される仕組みとなっています。使用する電源のレスポンスや、セット上の条件によって、電源が安定する時間は異なりますので、電源が安定する時間を考慮して、Warm-up 時間を決定してください。

15. PMCWW1 register

Address = (0xF002_0000) + 0x0400

Bit	Bit Symbol	Type	Reset Value	Hot Reset Value	Description
[31:5]	–	–	Undefined	Undefined	Read as undefined. Write as zero.
[6]	PMCBDTV	W	0y0	データ保持	バックアップレジスタ設定値の更新 0y0: 保持 0y1: 更新
[5]	PMCCTLV	R/W	0y0	データ保持	PMCCTL レジスタ設定値の更新 0y0: 保持 0y1: 更新
[4]	DMCCKECTLV	R/W	0y0	データ保持	DMCCKECTLV レジスタ設定値の更新 0y0: 保持 0y1: 更新
[3]	PMCDRVV	R/W	0y0	データ保持	PMCDRVV レジスタ設定値の更新 0y0: 保持 0y1: 更新
[2]	BPARINTV	R/W	0y0	データ保持	BPARINTV レジスタ設定値の更新 0y0: 保持 0y1: 更新
[1]	BPAEDGEV	R/W	0y0	データ保持	BPAEDGEV レジスタ設定値の更新 0y0: 保持 0y1: 更新
[0]	BPARELEV	R/W	0y0	データ保持	BPARELEV レジスタ設定値の更新 0y0: 保持 0y1: 更新

(個別説明)

電源遮断対応のために PMC の設定レジスタは電源遮断領域部分と電源常時通電領域部分にそれぞれレジスタを持っています。そのため、バックアップレジスタに設定値を更新するためには更新レジスタ(PMCWW1)をライトする必要があります。

PMCレジスタの制御は 3.24.3PMCレジスタの構成参照ください。

a. <PMCCTLV> <DMCCKECTLV> <PMCDRVV> . . .

0y0: 保持

0y0 に設定した場合、各設定レジスタの設定値を変更しても DVCC1B 回路のバックアップレジスタに更新されません。

0y1: 更新

各設定レジスタの設定値を DVCC1B 回路のバックアップレジスタに更新するため、0y1 を本ビットに書き込みしてください。

注) PMCCTL<PCM_ON>は特殊ビットであり、PMCWW1 レジスタで更新できません。

PMCREG レジスタで更新してください。

16. PMCRES register

Address = (0xF002_0000) + 0x041C

Bit	Bit Symbol	Type	Reset Value	Hot Reset Value	Description
[31:8]	–	–	Undefined	Undefined	Read as undefined. Write as zero.
[7]	RES_PCMON	R/W	0y0	0y0	PMCCTL<PCM_ON> Flag クリア : 0y0: 無効 0y1: クリアします
[6:0]	–	–	Undefined	Undefined	Read as undefined. Write as zero.

(個別説明)

a. <RES_PCMON>

<RES_PCMON>に“1”をライトすることで、レジスタ PMCCTL<PCM_ON>がクリアされます。
Power Cut Mode から復帰機能として使用出来ません。

0y0: 無効

0y1: <PCM_ON>をクリアします

17. BPAIE register

Address = (0xF002_0000) + (0x0D80)

Bit	Bit Symbol	Type	Reset Value	Hot Reset Value	Description
[31:4]	–	–	Undefined	Undefined	Read as undefined. Write as zero.
[3]	BPAIE3	W	0y0	データ保持	PCM 解除要求許可 0y0: disable 0y1: enable
[2]	BPAIE2	W	0y0	データ保持	PCM 解除要求許可 0y0: disable 0y1: enable
[1]	BPAIE1	W	0y0	データ保持	PCM 解除要求許可 0y0: disable 0y1: enable
[0]	BPAIE0	W	0y0	データ保持	PCM 解除要求許可 0y0: disable 0y1: enable

(個別説明)

a. <BPAIE[3:0]>

キー入力 KI[3:0]により、PCM 解除要求許可するレジスタです。

注) BPARELE[3:0]と同時に設定が必要です。

18. BPCIE register

Address = (0xF002_0000) + (0x0D88)

Bit	Bit Symbol	Type	Reset Value	Hot Reset Value	Description
[31:8]	–	–	Undefined	Undefined	Read as undefined. Write as zero.
[7]	BPAIE7	W	0y0	データ保持	PCM 解除要求許可 0y0: disable 0y1: enable
[6:0]	–	–	Undefined	Undefined	Read as undefined. Write as zero.

(個別説明)

a. <BPCIE7>

INT9 により、PCM 解除要求許可するレジスタです。

注) BPXRELE[0]と同時に設定が必要です。

19. BPDIE register

Address = (0xF002_0000) + (0x0D8C)

Bit	Bit Symbol	Type	Reset Value	Hot Reset Value	Description
[31:8]	–	–	Undefined	Undefined	Read as undefined. Write as zero.
[7]	BPDIE7	W	0y0	データ保持	PCM 解除要求許可 0y0: disable 0y1: enable
[6]	BPDIE6	W	0y0	データ保持	PCM 解除要求許可 0y0: disable 0y1: enable
[5:0]	–	–	Undefined	Undefined	Read as undefined. Write as zero.

(個別説明)

a. <BPDIE[7:6]>

PD7(INTB)、PD6(INTA)により、PCM 解除要求許可するレジスタです。

注) BPDRELE[6]と同時に設定が必要です。

注) BPXRELE[1]と同時に設定が必要です。

3.24.8 使用設定例

下記プログラム例は、内蔵 RAM からのプログラム実行が前提です。

また、PCM モードへ移行する際に、その動作を阻害する可能性のある要因をすべて停止と端子処理を行ってください。

- ウォッチドッグタイマの停止(初期値は停止状態です)
- AD コンバータ動作の停止
- システムの DMA 動作の停止
- LCDC の停止
- SDRAM の Auto リフレッシュの停止(セルフリフレッシュモードへの移行)
- DMA 転送の停止
- 端子状態の固定

端子の状態を固定すると同時に、PCM 状態の端子の状態も設定します。また、Wake-up 可能な外部割り込みは、エッジ選択可能です。エッジの選択設定をしてください。PD6 を INTTSI の割り込み INTA として利用する場合は、デバウンス回路は Disable 状態に設定することが必要です。

- 割り込みの禁止
- OFD 動作の停止
- PLL 動作の停止
- 内部キャッシュメモリの Disable 設定

; Port A[0]で Wakeup するプログラム設定例

;----- Back up Data set Register in Power Cut Mode

LDR	r0,=BPADATA	; Port A Pull-up enable for PMC mode
MOV	r1,#0x000000FF	
STR	r1,[r0]	
LDR	r0,=BPBDATA	; PB Data set in Power Cut Mode
MOV	r1,#0x000000FF	
STR	r1,[r0]	
LDR	r0,=BPCDATA	; PC Data set in Power Cut Mode
MOV	r1,#0x000000FF	
STR	r1,[r0]	
LDR	r0,=BPDDATA	; PD Data set in Power Cut Mode
MOV	r1,#0x00000040	
STR	r1,[r0]	
LDR	r0,=BPNDATA	; PN Data set in Power Cut Mode
MOV	r1,#0x000000FF	
STR	r1,[r0]	
LDR	r0,=BPTDATA	; PT Data set in Power Cut Mode
MOV	r1,#0x000000FF	
STR	r1,[r0]	
LDR	r0,=BPUDATA	; PU Data set in Power Cut Mode
MOV	r1,#0x000000FF	
STR	r1,[r0]	
LDR	r0,=BPVDATA	; PV Data set in Power Cut Mode
MOV	r1,#0x000000FF	
STR	r1,[r0]	

;----- Back Up Data Output Enable Register

LDR	r0,=BPBOE	; PB Output enable
MOV	r1,#0x000000FF	
STR	r1,[r0]	
LDR	r0,=BPCOE	; PC Output enable
MOV	r1,#0x000000FF	
STR	r1,[r0]	
LDR	r0,=BPDOE	; PD Output enable
MOV	r1,#0x000000F0	
STR	r1,[r0]	
LDR	r0,=BPNOE	; PN Output enable
MOV	r1,#0x000000FF	
STR	r1,[r0]	
LDR	r0,=BPTOE	; PT Output enable
MOV	r1,#0x000000FF	
STR	r1,[r0]	
LDR	r0,=BPUOE	; PU Output enable
MOV	r1,#0x000000FF	
STR	r1,[r0]	
LDR	r0,=BPVOE	; PV Output enable
MOV	r1,#0x000000FF	
STR	r1,[r0]	

;----- Back up Data set Register in Power Cut Mode

LDR	r0,=BSADATA	; SA Data set in Power Cut Mode
MOV	r1,#0x000000FF	
STR	r1,[r0]	
LDR	r0,=BSBDATA	; SB Data set in Power Cut Mode
MOV	r1,#0x000000FF	
STR	r1,[r0]	
LDR	r0,=BSEDATA	; SE Data set in Power Cut Mode
MOV	r1,#0x000000FF	
STR	r1,[r0]	
LDR	r0,=BSFDATA	; SF Data set in Power Cut Mode
MOV	r1,#0x000000FF	
STR	r1,[r0]	
LDR	r0,=BSGDATA	; SG Data set in Power Cut Mode
MOV	r1,#0x000000FF	
STR	r1,[r0]	
LDR	r0,=BSHDATA	; SH Data set in Power Cut Mode
MOV	r1,#0x000000FF	
STR	r1,[r0]	
LDR	r0,=BSJDATA	; SJ Data set in Power Cut Mode
MOV	r1,#0x000000FF	
STR	r1,[r0]	
LDR	r0,=BSKDATA	; SK Data set in Power Cut Mode
MOV	r1,#0x000000FF	
STR	r1,[r0]	
LDR	r0,=BSLDATA	; SL Data set in Power Cut Mode
MOV	r1,#0x000000FF	
STR	r1,[r0]	

;----- Back Up Data Output Enable Register

LDR	r0,=BSAOE	; SA Output enable
MOV	r1,#0x000000FF	
STR	r1,[r0]	
LDR	r0,=BSBOE	; SB Output enable
MOV	r1,#0x000000FF	
STR	r1,[r0]	
LDR	r0,=BSEOE	; SE Output enable
MOV	r1,#0x000000FF	
STR	r1,[r0]	
LDR	r0,=BSFOE	; SF Output enable
MOV	r1,#0x000000FF	
STR	r1,[r0]	
LDR	r0,=BSGOE	; SG Output enable
MOV	r1,#0x000000FF	
STR	r1,[r0]	
LDR	r0,=BSHOE	; SH Output enable
MOV	r1,#0x000000FF	
STR	r1,[r0]	
LDR	r0,=BSJOE	; SJ Output enable
MOV	r1,#0x000000FF	
STR	r1,[r0]	
LDR	r0,=BSKOE	; SK Output enable
MOV	r1,#0x000000FF	
STR	r1,[r0]	
LDR	r0,=BSLOE	; SL Output enable
MOV	r1,#0x000000FF	
STR	r1,[r0]	

----- Wake Up Enable Register

```

LDR      r0,=BPAIE          ; PA0, set enable
MOV      r1,#0x00000001
STR      r1,[r0]

LDR      r0,=BPCIE          ; Disable
MOV      r1,#0x00000000
STR      r1,[r0]

LDR      r0,=BPDIE          ; Disable
MOV      r1,#0x00000000
STR      r1,[r0]

```

----- Wake Up Enable Register

```

LDR      r0,=BPARELE        ; PA0, set enable
MOV      r1,#0x00000001
STR      r1,[r0]

LDR      r0,=BPDRELE        ; Disable
MOV      r1,#0x00000000
STR      r1,[r0]

LDR      r0,=BRTRELE        ; Disable
MOV      r1,#0x00000000
STR      r1,[r0]

LDR      r0,=BPXRELE        ; Disable
MOV      r1,#0x00000000
STR      r1,[r0]

```

----- Wake Up Source Edge Select Register

```

LDR      r0,=BPAEDGE        ; PA0, set rising edge
MOV      r1,#0x00000000
STR      r1,[r0]

LDR      r0,=BPDEGE         ;
MOV      r1,#0x00000000
STR      r1,[r0]

LDR      r0,=BPXEDGE        ;
MOV      r1,#0x00000000
STR      r1,[r0]

```

----- Wake Up Source Initial Register

```

    LDR    r0,=BPARINT          ; Clear the status of wakeup request
    MOV    r1,#0x00000000
    STR    r1,[r0]

    LDR    r0,=BPDRINT          ; Clear the status of wakeup request
    MOV    r1,#0x00000000
    STR    r1,[r0]

    LDR    r0,=BRTRINT          ; Clear the status of wakeup request
    MOV    r1,#0x00000000
    STR    r1,[r0]

    LDR    r0,=BPRINT           ; clear the status of wakeup request
    MOV    r1,#0x00000000
    STR    r1,[r0]

```

----- Pre PMCCTL Register set

```

    LDR    r0,=PMCCTL           ;PMCCTL pre set
    MOV    r1,#0x00000043
    STR    r1,[r0]

```

----- PMC Write Valid Register

```

    LDR    r0,=PMCWW1           ;PMCWW1
    MOV    r1,#0x0000007F
    STR    r1,[r0]

    NOP
    NOP
    NOP
    NOP
    NOP
    NOP

```

----- Write Valid Flag Check

WAIT_PMCWW1

```

    LDR    r0,=PMCWW1
    LDR    r0,[r0]
    AND    r0,#0x7F
    CMP    r0,#0x7F
    BNE    WAIT_PMCWW1

```

```

    NOP
    NOP

```

```

    LDR    r0,=PMCCTL           ;PMCCTL :PMC MODE
    MOV    r1,#0xc3
    STR    r1,[r0]

```

NOP_LOOP

```

    NOP
    NOP
    NOP
    NOP
    B      NOP_LOOP

```

```

    HALT

```

;

3.25 USBホストコントローラ

USB ホストコントローラ(USBHC)は、USB Specification Rev 2.0、および Open HCI Specification Release 1.0a に準拠しており、12Mbps(フルスピード)の USB 転送が可能です。USBHC はバスブリッジ回路を通して CPU に接続されます。

なお、USBホストコントローラには幾つかの使用上の注意、制約事項があります。詳しくは 3.25.8 項をご確認下さい。

3.25.1 システム概要

USBHC の主な機能は次のとおりです。

(1)12Mbps(フルスピード)の USB デバイスを接続できます。

Low-Speed(1.5Mbps)はサポートしていません。

(2)コントロール、バルク、インタラプト、アイソクロナス転送をサポートしています。

(3)CPU とのバスブリッジ回路内に IN、OUT それぞれ16バイトの FIFO バッファを内蔵しており、最大 16 バイトのバースト転送が可能です。

(4)バスブリッジ回路内の FIFO バッファと内蔵の SRAM 間のデータ転送を行います。

3.25.2 システム構成

USBHC は以下のブロック構成となっています。

- (1)USBHC コア(OHCI)
- (2)USB トランシーバ
- (3)CPU バスブリッジ回路

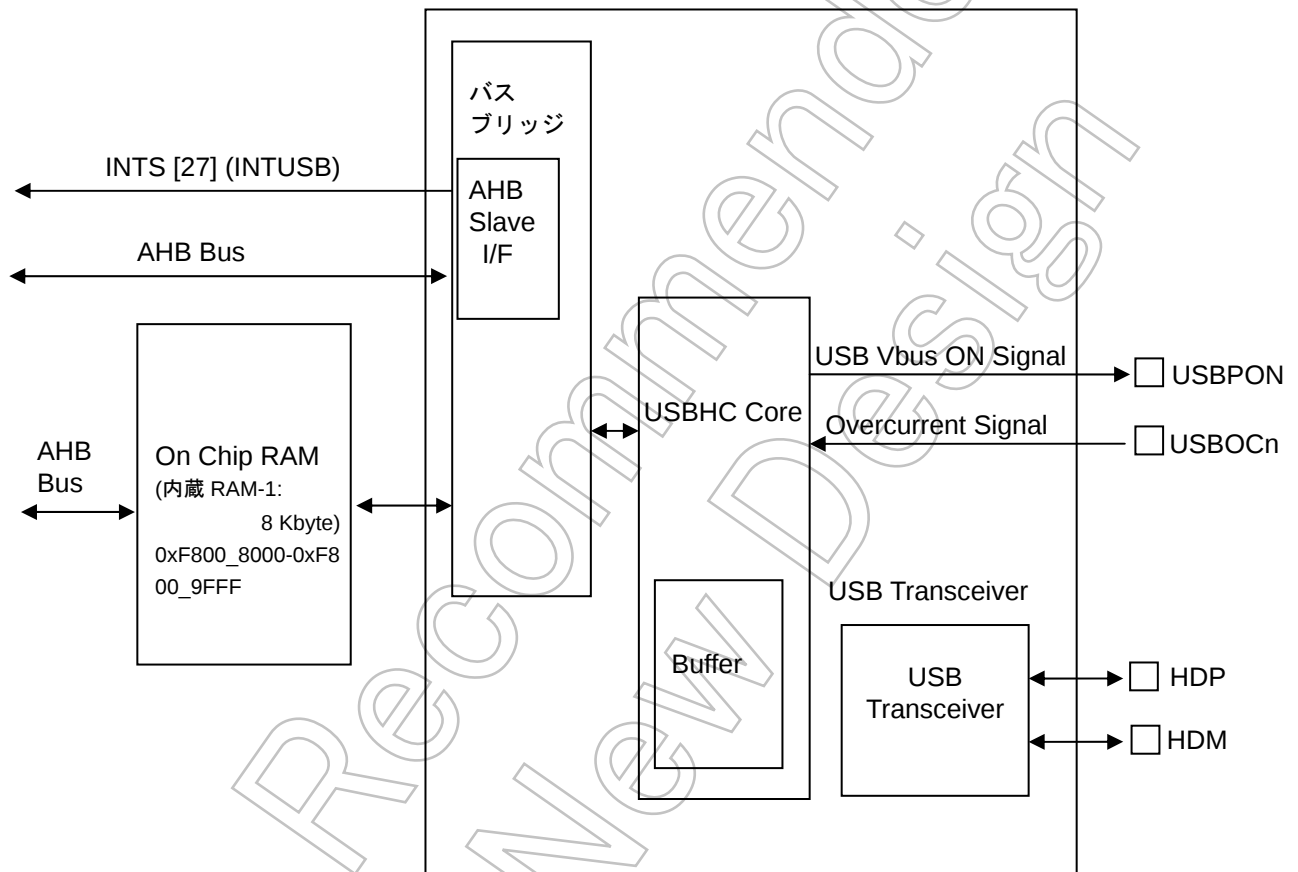


図 3.25.1 USB Host Controller

3.25.3 割り込み

USBHC は以下の割り込みを発生します。

- Scheduling Overrun
- HcDoneHead Write back
- Start of Frame
- Resume Detect
- Unrecoverable Error
- Frame Number Overflow
- Root Hub Status Change
- Ownership Change

割り込み条件が成立すると、USBHC は内部の **HcInterruptStatus** レジスタの対応するビットをセットします。このビットがセットされたとき、**HcInterruptEnable** レジスタの **MasterInterruptEnable (MIE)** ビットが許可状態で、かつ **HcInterruptEnable** レジスタで当該ビットが許可されていると、USB 割り込み(**INTUSB**)が発生します。

USBHC のドライバソフトウェアは、**HcInterruptStatus** レジスタの当該ビットに 1 を書き込むことでビットをクリアできます。(ドライバソフトウェアは、これらのビットをセットできません。また USBHC はこれらのビットをクリアできません。)

3.25.4 リセット

USBHC はハードウェアリセット、もしくはソフトウェアリセットにより初期化されます。

3.25.4.1 ハードウェアリセット

外部リセット端子、内部リセット（WDT リセット、OFD リセット、PCM 復帰）が発生します。

- 全てのレジスタは初期化されます。
- USBHC は USB バス上にリセット信号を出力します。
- USB の状態は、USBRESET ステートへ遷移します。
- USBHC のリスト処理と SOF トークンの生成は禁止されます。
- HcFmNumber レジスタの FrameNumber は増加しません。

3.25.4.2 ソフトウェアリセット

ソフトウェアリセットは、HcCommandStatus レジスタ内の HostControllerReset ビットに 1 を書き込むと発生します。

- OHCI の全てのレジスタは初期化されます。
但し、HcControl レジスタの RemoteWakeupConnected ビット、及び InterruptRouting ビットは変更しません。
また、HcBCR0 レジスタは初期化されません。
- USBHC は USB バス上にリセット信号を出力します。
- USB の状態は USBSUSPEND へ遷移します。
(HcControl レジスタの FunctionalState ビットが 0x03 にセットされ USBSUSPEND ステートへ遷移します。)

3.25.5 バスパワー制御

USBHC は外部の Vbus 用電源 IC の制御信号を備えており、USBPON 端子(PT4)で制御します。

まず端子の機能設定をポート T の制御レジスタ(PTFC)にて行います。その上で、OHCI レジスタの HcRhStatus レジスタの LPSC を 1 にセットすると、USBPON 端子から High を出力します。

一方、過電流の検出は、USBOCn(PT5)端子で行います。USBHC は、この端子から Low を検出すると、OHCI レジスタの HcRhStatus の OCI ビットを 1 にします。(USBOCn の端子として使用する場合は、端子機能の設定をポート T の制御レジスタ(PTFC)にて行って下さい。)

3.25.6 レジスタ

USBHC の制御用に OHCI に準拠したレジスタをメモリ空間に実装しています。また、CPU とのバスブリッジ回路にも制御レジスタがあります。

これらのレジスタは、32 ビットにて CPU から直接アクセスすることができます。

表 3.25.1 USB OpenHCI 準拠のレジスタ一覧

base アドレス= 0xF450_0000

[illegible]

注 1) 表 3.25.1に記載しているアドレスは、CPUレジスタ上にマッピングされたアドレスを表記しています。

注 2) Open HCI Specification Release 1.0a では、HcFmRemaining レジスタの FrameRemaining(FR)ビットと FrameRemainingToggle(FRT)ビット、HcFmNumber レジスタの FrameNumber(FN)ビットはホストコントロールドライバ(HCD)ではリードオンリーです。しかし、USB 1.1 OHCI ホストコントロールコアは、デバッグ目的で HCD によるこれらのレジスタへの書き込みを許可しています。HCD がこれらのレジスタに書込むと、未定義の結果となりますので、HCD でこれらのビットに書き込まないで下さい。

注 3) 次項以降の 1Hc Revision レジスタ～22HcRhPortStatus レジスタは、参考と訳です。OHCI に準拠した各レジスタは、Open HCI Specification Release 1.0a の仕様書をご参照下さい。

1. HcRevision レジスタ

Address = (0xF450_0000) + (0x0000)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
bit Symbol	Reserved															
Read/Write (HCD)																
Read/Write (HC)																
リセット後																
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bit Symbol	Reserved								Rev							
Read/Write (HCD)									R							
Read/Write (HC)									R							
リセット後									0	0	0	1	0	0	0	0

ビット	ニモニック	フィールド名	機能
[31:8]	–	Reserved	
[7:0]	REV	Revision	このリードオンリーフィールドには、HC で実行される HCI 仕様バージョンの BCD 表記を設定します。 例えば、値 0x11 はバージョン 1.1 を示します。この仕様に準拠する全ての HC 実行は 0x10 となります。

2. HcControl レジスタ

HcControl レジスタは、ホストコントローラの動作モードを定義します。このレジスタで HostControllerFunctionState と RemoteWakeupConnected 以外の殆どのフィールドはホストドライバだけで変更できます。

$$\text{Address} = (0xF450_0000) + (0x0004)$$

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
bit Symbol	Reserved															
Read/Write (HCD)																
Read/Write (HC)																
リセット後																
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bit Symbol	Reserved					RWE	RWC	IR	HCFS		BLE	CLE	PLE	CBSR		
Read/Write (HCD)						R/W										
Read/Write (HC)						R	R/W	R	R/W		R	R	R	R	R	
リセット後						0	0	0	0	0	0	0	0	0	0	0

ビット	ニモニック	フィールド名	機能
[31:11]		Reserved	
[10]	RWE	RemoteWakeup Enable	このビットはHCDによって使用され、上流の Resume 信号検出のときリモートウェイクアップ機能をイネーブルまたはディセーブルします。このビットがセットされ、HcInterruptStatus レジスタの ResumeDetect ビットがセットされると、リモートウェイクアップがホストシステムに通知されます。このビットをセットしても、ハードウェア割り込みの生成に影響されません。
[9]	RWC	RemoteWakeup Connected	このビットは、HC がリモートウェイクアップ機能をサポートするかどうかを示します。リモートウェイクアップがサポートされていて、システムで使用される場合、POST(Power on Self Test)期間中にシステムファームウェアがこのビットをセットします。HC はハードウェアリセット時にはこのビットをクリアします。ソフトウェアリセットでは変更されません。
[8]	IR	Interrupt Routing	このビットはHcInterruptStatus レジスタに登録されているイベントによって生成される割り込みルーチンを決めます。クリアされると全ての割り込みは、通常のホストバス割り込みのルーチンになります。セットされると割り込みがシステムマネージメント割り込みにルーチンされます。HCD はハードウェアのリセット時にこのビットをクリアしますが、ソフトウェアリセット時にはこのビットを変更しません。HCD は HC のオーナーシップであることを示すタグとしてこのビットを使用します。

ビット	ニモニック	フィールド名	機能
[7:6]	HCFS	HostController FunctionalState ForUSB	0y00:USBRESET 0y01:USBRESUME 0y10:USBOPERATIONAL 0y11:USBSUSPEND あるステートから USBOPERATIONAL への遷移により、1ms 後に SOF パケットが生成されます。HCD は HcInterrupt レジスタの StartofFrame フィールドを読むことにより、HC が SOF を送り始めたかを判断します。 このフィールドは USBSUSPEND ステートのときのみ HC によって変更できます。 HC は下流のポートからレジャーム信号を検出したあと、USBSUSPEND ステートから USBRESUME ステートへ遷移します。 HC はソフトウェアリセット後、USBSUSPEND ステートに入り、ハードウェアリセット後には USBRESET ステートに入ります。ハードウェアリセット後には Root Hub もリセットし、下流のポートへのリセット信号をアサートします。
[5]	BLE	BulkListEnable	このビットをセットすると、次のフレームの Bulk リストの処理をイネーブルします。HCD によりクリアされると、Bulk リストの処理は次の SOF の後に実行されません。 HC はリストの処理を決めるとき必ずこのビットをチェックします。ディセーブル時には、HCD はこのリストを変更できます。 HCBulkCurrentED レジスタが削除する ED を示している場合、HCD はリストの処理を再度イネーブルする前に HCBulkCurrentED を更新してポインタを進めなければなりません。
[4]	CLE	ControlList Enable	このビットをセットすると、次のフレームの Control リストの処理をイネーブルします。HCD によりクリアされると、Control リストの処理は次の SOF の後に実行されません。HC はリストの処理を決めるときは必ずこのビットをチェックします。ディセーブル時には、HCD はこのリストを変更できます。HcControlCurrentED レジスタが削除する ED を示している場合、HCD はリストの処理を再度イネーブルする前に HcControlCurrentED を更新してポインタを進めなければなりません。
[3]	IE	Isochronous Enable	このビットは HCD がアイソクロナス ED の処理をイネーブル、またはディセーブルするのに使用します。フレームで周期転送リストを処理しているときに、HC はアイソクロナス ED (F=1) を検出するとこのビットをチェックします。セット（イネーブル）の場合は、ED の処理を継続します。クリア（ディセーブル）の場合、HC は周期転送リストの処理を中止して Bulk リストと Control リストの処理を開始します。このビットをセットすると次のフレームでも有効です。 ※本製品はアイソクロナス転送時に制約事項があります。
[2]	PLE	PeriodicList Enable	このビットをセットすると、次のフレームでの周期転送リストの処理をイネーブルします。HCD でクリアされると、周期転送リストの処理は次の SOF の後に実行されません。HC は、リストの処理を始める前に、このビットをチェックしなければなりません。
[1:0]	CBSR	ControlBulk ServiceRatio	このビットは Control ED と Bulk ED の処理比率を指定します。非周期転送リストを処理する前に、HC は指定された比率と空でない Control ED が幾つ処理されたかについての内部カウンタを比較し、次の Control ED の処理を継続するか、または Bulk ED の処理に変更するかを決めます。内部カウンタはフレームの境界を跨いでも保持されます。リセットの場合、HCD はこの値を再度保持します。 CBSR No. of Control EDs over Bulk EDs served 00 1:1 01 2:1 10 3:1 11 4:1

3. HcCommandStatus レジスタ

HcCommandStatus レジスタは、ホストコントローラの現在のステータスを反映するだけでなく、ホストコントローラドライバで発行されるコマンドを受け取るためにホストコントローラによって使用されます。ホストコントローラドライバに対して、HcCommandStatus レジスタは、「セットするために書き込む」レジスタに見えます。ホストコントローラは、1 が書き込まれたビットはレジスタ内でセットされ、0 が書き込まれたビットはレジスタ内で変化しないことを確認します。ホストコントローラドライバは、以前に発行されたコマンドが無効になることに無関係に、ホストコントローラに対して複数の異なったコマンドを発行します。ホストコントローラドライバは、全てのビットに通常のリードアクセスをします。

SchdulingOverrunCount フィールドは、ホストコントローラがスケジューリングオーバーランエラーを検出したフレームの数です。このエラーは周期転送リストが EOF 前に完了しないときに発生します。スケジューリングオーバーランエラーが検出されると、ホストコントローラはカウンタをインクリメントして HcInterruptStatus レジスタの SchedulingOverrun フィールドをセットします。

$$\text{Address} = (\text{0xF450_0000}) + (\text{0x0008})$$

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
bit Symbol	Reserved														SOC	
Read/Write (HCD)															R	
Read/Write (HC)															R/W	
リセット後															0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bit Symbol	Reserved												OCR	BLF	CLF	HCR
Read/Write (HCD)													R/W			
Read/Write (HC)													R/W			
リセット後													0	0	0	0

ビット	ニモニク	フィールド名	機能
[31:18]	—	Reserved	
[17:16]	SOC	Scheduling OverrunCount	これらのビットは、スケジューリングオーバーランエラーが発生するたびにインクリメントされます。00bに初期化され、11bでラップアラウンドします。HcInterruptStatus レジスタの SchedulingOverrun フィールドがセットされているときでさえも、スケジューリングオーバーランエラーが検出されると、これらのビットはインクリメントされます。なかなか解消しないスケジューリングの問題をモニタするのに HCD で使用されます。
[15:4]	—	Reserved	
[3]	OCR	Ownership ChangeRequest	このビットは HC の制御変更をリクエストするために、OS HCD によりセットされます。このビットがセットされると、HC は HcInterruptStatus レジスタの OwnershipChange フィールドをセットします。変更後このビットはクリアされ、OS HCD からの次のリクエストが出されるまでクリアされたままです。

ビット	ニモニック	フィールド名	機能
[2]	BLF	BulkListFilled	このビットは、Bulk リストに TD が存在するかどうかを示すのに使用されます。HCD は TD を Bulk リストの ED に追加するたびにこのビットをセットします。 HC は Bulk リストのヘッド処理を開始するとき BLF をチェックします。BulkListFilled が 0 の間、HC は Bulk リストの処理を開始しません。1 のとき、HC は Bulk リストの処理を開始し、BLF を 0 にクリアします。HC がリストに TD を見つけると、Bulk リスト処理を継続させる 1 を BulkListFilled にセットします。Bulk リストに TD が見つからず、HCD が BulkListFilled をセットしていない場合には、HC が Bulk リストの処理を完了したとき BulkListFilled は 0 のままで Bulk リスト処理は停止します。
[1]	CLF	ControllistFilled	このビットは、Control リストに TD が存在するかどうかを示すのに使用されます。HCD は TD を Control リストの ED に追加するたびにこのビットをセットします。 HC は Control リストのヘッド処理を開始するとき CLF をチェックします。ControllistFilled が 0 の間、HC は Control リストの処理を開始しません。1 のとき、HC は Control リストの処理を開始し、CLF を 0 にクリアします。HC がリストに TD を見つけると、Control リスト処理を継続させる 1 を ControllistFilled にセットします。Control リストに TD が見つからず、HCD が ControllistFilled をセットしていない場合には、HC が Control リストの処理を完了したとき ControllistFilled は 0 のままで Control リスト処理は停止します。
[0]	HCR	HostController Reset	このビットは、HC のソフトウェアリセットを起動するために HCD によりセットされます。HC の機能ステートと無関係に、HC は USB SUSPEND ステートに入り、例えば HcControl レジスタの InterruptRouting フィールドのように特別に記述されているもの以外の殆どの動作レジスタはリセットされ、ホストバスアクセスは許可されません。このビットはリセット動作の完了時にクリアされます。リセット動作は 10us 以内に完了しなければなりません。このビットをセットすることにより、Root Hub がリセットされることはなく、また下流ポートでリセット信号がアサートすることはありません。

4. HcInterruptStatus レジスタ

HcInterruptStatus レジスタは、ハードウェア割り込みの原因となる様々なイベントのステータスを提供します。イベントが発生するとき、ホストコントローラはこのレジスタで対応するビットをセットします。ビットがセットされると、HcInterruptEnable レジスタで割り込みがイネーブルされていて、また MasterInterruptEnable ビットがセットされている場合、ハードウェア割り込みが発生します。ホストコントローラドライバは、クリアするビットに1を書き込むことでこのレジスタの特定のビットをクリアできます。ホストコントローラドライバは、これらのビットをセットできません。ホストコントローラはこのビットをクリアできません。

$$\text{Address} = (0xF450_0000) + (0x000C)$$

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
bit Symbol	Reserved	OC	Reserved													
Read/Write (HCD)		R/W														
Read/Write (HC)		R/W														
リセット後		0														
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bit Symbol	Reserved									RHSC	FNO	UE	RD	SF	WDH	SO
Read/Write (HCD)										R/W						
Read/Write (HC)										R/W						
リセット後										0	0	0	0	0	0	0

ビット	ニモニク	フィールド名	機能
[31]	—	Reserved	
[30]	OC	Ownership Change	このビットは HCD が HcCommandStatus レジスタの OwnershipChangeRequest フィールドをセットするとき HC によってセットされます。マスクされていない場合、このイベントは直ちにシステムマネージメント割り込み(SMI)を発生します。SMI ビンが実行されないとき、このビットは0になります。
[29:7]	—	Reserved	
[6]	RHSC	RootHubStatus Change	このビットは HcRhStatus レジスタの内容、または HcRhPortStatus [NumberOfDownstreamPort] レジスタの内容が変更されたとき、セットされます。
[5]	FNO	FrameNumber Overflow	このビットは HcFmNumber レジスタの MSB(ビット 15)の値が0から1へ、または1から0へ変化し、HccaFrameNumber が更新されたあとでセットされます。
[4]	UE	Unrecoverable Error	このビットは HC が USB に関係の無いシステムエラーを検出するとセットされます。HC はシステムエラーの修正が終了する前に処理を進めたり、信号を送ることはありません。HCD は HC がリセットされた後、このビットをクリアします。

ビット	ニモニック	フィールド名	機能
[3]	RD	ResumeDetected	USB デバイスがレジューム信号をアサートしているのを HC が検出すると、このビットはセットされます。これは、レジューム信号でない信号からこのビットがセットされる原因となるレジューム信号への遷移です。HCD が USBRESUME ステートをセットするとき、このビットはセットされません。
[2]	SF	StartofFrame	このビットはフレームの開始毎に、また HccaFrameNumber の更新後に HC によりセットされます。HC は同時に SOF トークンを生成します。
[1]	WDH	WritebackDone Head	このビットは HC が HcDoneDead を HccaDoneHead に書き込んだ直後にセットされます。さらに HccaDoneHead の更新はこのビットがクリアされてから初めて実行されます。HCD は HccaDoneHead の内容を保存した後、このビットをクリアするだけです。
[0]	SO	Scheduling Overrun	現在のフレーム用 USB スケジュールがオーバーランしたときと、HccaFrameNumber の更新後にこのビットはセットされます。スケジュールのオーバーランは、HcCommandStatus レジスタの SchedulingOverrunCount フィールドをインクリメントさせます。

5. HcInterruptEnable レジスタ

HcInterruptEnable レジスタの各イネーブルビットは、HcInterruptStatus レジスタの関連する割り込みビットに対応しています。HcInterruptEnable レジスタは、どのイベントがハードウェア割り込みを発生させるかを制御するのに使用されます。ビットが HcInterruptStatus レジスタにセットされ、HcInterruptEnable レジスタの対応するビットがセットされ、MasterInterruptEnable ビットがセットされると初めてハードウェア割り込みがホストバス上でリクエストされます。

このレジスタのビットに 1 を書き込むと、対応するビットをセットします。0 を書き込むと対応するビットの値は変化しません。読み出し時に、このレジスタの現在値が返されます。

$$\text{Address} = (0xF450_0000) + (0x0010)$$

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
bit Symbol	MIE	OC	Reserved													
Read/Write (HCD)	R/W															
Read/Write (HC)	R															
リセット後	0	0														
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bit Symbol	Reserved									RHSC	FNO	UE	RD	SF	WDH	SO
Read/Write (HCD)										R/W						
Read/Write (HC)										R						
リセット後										0	0	0	0	0	0	0

ビット	ニモニック	フィールド名	機能
[31]	MIE	MasterInterrupt Enable	このフィールドに書き込まれた 0 は、HC で無視されます。このフィールドに書き込まれた 1 はこのレジスタの他のビットで指定されているイベントにより、割り込みの生成をイネーブルします。これは、マスタ割り込みイネーブルとして HCD により使用されます。
[30]	OC	Ownership Change	0y0: 無視 0y1: OwnershipChange による割り込みの生成をイネーブルします。
[29:7]	—	Reserved	
[6]	RHSC	RootHubStatus Change	0y0: 無視 0y1: RootHubStatusChange による割り込みの生成をイネーブルします。
[5]	FNO	FrameNumber Overflow	0y0: 無視 0y1: FrameNumberOverflow による割り込みの生成をイネーブルします。
[4]	UE	Unrecoverable Error	0y0: 無視 0y1: UnrecoverableError による割り込みの生成をイネーブルします。
[3]	RD	ResumeDetected	0y0: 無視 0y1: ResumeDetected による割り込みの生成をイネーブルします。
[2]	SF	StartofFrame	0y0: 無視 0y1: StartofFrame による割り込みの生成をイネーブルします。
[1]	WDH	WritebackDone Head	0y0: 無視 0y1: HcDoneHeadWriteback による割り込みの生成をイネーブルします。
[0]	SO	Scheduling Overrun	0y0: 無視 0y1: SchedulingOverrun による割り込みの生成をイネーブルします。

6. HcInterruptDisable レジスタ

HcInterruptDisable レジスタの各ディセーブルビットは、HcInterruptStatus レジスタの関連する割り込みビットに対応しています。HcInterruptDisable レジスタは、HcInterruptEnable レジスタと連結しています。HcInterruptDisable レジスタに 1 を書くと HcInterruptEnable レジスタの対応するビットをクリアし、0 を書くと HcInterruptEnable レジスタの対応するビットを変更しません。読み出し時、HcInterruptEnable レジスタの現在値が返されます。

$$\text{Address} = (0xF450_0000) + (0x0014)$$

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
bit Symbol	MIE	OC	Reserved													
Read/Write (HCD)	R/W															
Read/Write (HC)	R															
リセット後	0	0														
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bit Symbol	Reserved									RHSC	FNO	UE	RD	SF	WDH	SO
Read/Write (HCD)										R/W						
Read/Write (HC)										R						
リセット後										0	0	0	0	0	0	0

ビット	ニモニック	フィールド名	機能
[31]	MIE	MasterInterrupt Enable	このフィールドに書き込まれた 0 は、HC で無視されます。このフィールドに書き込まれた 1 はこのレジスタの他のビットで指定されているイベントにより、割り込みの生成をディセーブルします。このフィールドはハードウェアリセット、またはソフトウェアリセット後にセットされます。
[30]	OC	Ownership Change	0y0: 無視 0y1: OwnershipChange による割り込みの生成をディセーブルします。
[29:7]		Reserved	
[6]	RHSC	RootHubStatus Change	0y0: 無視 0y1: RootHubStatusChange による割り込みの生成をディセーブルします。
[5]	FNO	FrameNumber Overflow	0y0: 無視 0y1: FrameNumberOverflow による割り込みの生成をディセーブルします。
[4]	UE	Unrecoverable Error	0y0: 無視 0y1: UnrecoverableError による割り込みの生成をディセーブルします。
[3]	RD	ResumeDetected	0y0: 無視 0y1: ResumeDetected による割り込みの生成をディセーブルします。
[2]	SF	StartoffFrame	0y0: 無視 0y1: StartoffFrame による割り込みの生成をディセーブルします。
[1]	WDH	WritebackDone Head	0y0: 無視 0y1: HcDoneHeadWriteback による割り込みの生成をディセーブルします。
[0]	SO	Scheduling Overrun	0y0: 無視 0y1: SchedulingOverrun による割り込みの生成をディセーブルします。

7. HcHCCA レジスタ

HcHCCA レジスタは、ホストコントローラコミュニケーション領域の物理アドレスを設定します。ホストコントローラドライバは、HcHCCA レジスタに 1 を書き込み、HcHCCA レジスタの内容を読み出すことによりアライメント制約を決めます。下位のビットでゼロの数を調べることでアライメントを評価します。最小アライメントは 256 バイトです。そのため 0 から 7 までのビットは、読み出し時常に 0 を返します。この領域は、ホストとコントローラとホストコントローラドライバの両方によりアクセスされるコントロール構成と割り込みテーブルを置くのに使用されます。

$$\text{Address} = (0xF450_0000) + (0x0018)$$

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
bit Symbol	HCCA															
Read/Write (HCD)	R/W															
Read/Write (HC)	R															
リセット後	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bit Symbol	HCCA								Reserved							
Read/Write (HCD)	R/W															
Read/Write (HC)	R															
リセット後	0	0	0	0	0	0	0	0								

ビット	ニモニック	フィールド名	機能
[31:8]	HCCA	HostController Communication Area	ホストコントローラコミュニケーション領域のページアドレスです。
[7:0]	-	Reserved	

8. HcPeriodCurrentED レジスタ

HcPeriodCurrentED レジスタは、現在のアイソクロナス転送、または割り込みエンドポイントディスクリプタの物理アドレスを設定します。

$$\text{Address} = (0xF450_0000) + (0x001C)$$

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
bit Symbol	PCED															
Read/Write (HCD)	R															
Read/Write (HC)	R/W															
リセット後	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bit Symbol	PCED												Reserved			
Read/Write (HCD)	R															
Read/Write (HC)	R/W															
リセット後	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ビット	ニモニック	フィールド名	機能
[31:4]	PCED	PeriodCurrent ED	このフィールドは、現在のフレームで処理される周期転送リストの先頭を示すために HC で使用されます。このレジスタの内容は、周期転送 ED が処理された後で HC により更新されます。HCD は読み出しの際に、どの ED が現在処理されているか判断しているときに内容を読み出すかもしれません。
[3:0]	–	Reserved	

9. HcControlHeadED レジスタ

HcControlHeadED レジスタは、Control リスト中の最初のエンドポイントディスクリプタの物理アドレスを設定します。

$$\text{Address} = (0xF450_0000) + (0x0020)$$

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
bit Symbol	CHED															
Read/Write (HCD)	R/W															
Read/Write (HC)	R															
リセット後	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bit Symbol	CHED												Reserved			
Read/Write (HCD)	R/W															
Read/Write (HC)	R															
リセット後	0	0	0	0	0	0	0	0	0	0	0	0				

ビット	ニモニック	フィールド名	機能
[31:4]	CHED	ControlHeadED	HC は、HcControlHeadED ポインタで始まる Control リストをトラバースします。内容は、HC の初期化中に HCCA からロードされます。
[3:0]	–	Reserved	

10. HcControlCurrentED レジスタ

HcControlCurrentED レジスタは、Control リスト中の現在のエンドポイントディスクリプタの物理アドレスを設定します。

$$\text{Address} = (0xF450_0000) + (0x0024)$$

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
bit Symbol	CCED															
Read/Write (HCD)	R/W															
Read/Write (HC)	R/W															
リセット後	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bit Symbol	CCED												Reserved			
Read/Write (HCD)	R/W															
Read/Write (HC)	R/W															
リセット後	0	0	0	0	0	0	0	0	0	0	0	0				

ビット	ニモニック	フィールド名	機能
[31:4]	CCED	ControlCurrentED	現在の ED の処理後、ポインタは次の ED に進みます。HC は最後のフレームでリストの処理を中止したところから処理を継続します。Control リストの終端に到達すると、HC は HcCommandStatus レジスタの ControlListFilled フィールドをチェックします。セットされていると HC は HcControlHeadED レジスタの内容を HcControlCurrentED レジスタにコピーし、このビットをクリアします。セットされていない場合、HC は何も実行しません。HCD は HcControl レジスタの ControlListEnable フィールドがクリアされている時のみこのレジスタの変更が許可されます。セットされていると、HCD はこのレジスタの瞬時値を読み出すだけです。このフィールドには、最初 Control リスト終値を示すゼロがセットされています。
[3:0]	–	Reserved	

11. HcBulkHeadED レジスタ

HcBulkHeadED レジスタは、Bulk リスト中の最初のエンドポイントディスクリプタの物理アドレスを設定します。

$$\text{Address} = (0xF450_0000) + (0x0028)$$

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
bit Symbol	BHED															
Read/Write (HCD)	R/W															
Read/Write (HC)	R															
リセット後	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bit Symbol	BHED												Reserved			
Read/Write (HCD)	R/W															
Read/Write (HC)	R															
リセット後	0	0	0	0	0	0	0	0	0	0	0	0				

ビット	ニモニック	フィールド名	機能
[31:4]	BHED	BulkHeadED	HC は、HCBulkHeadED ポインタで始まる Bulk リストをトラバースします。内容は、HC の初期化中に HCCA からロードされます。
[3:0]	–	Reserved	

12. HcBulkCurrentED レジスタ

HcBulkCurrentED レジスタは、Bulk リスト中の現在のエンドポイントの物理アドレスを設定します。Bulk リストは順繰りに処理されるので、エンドポイントはリストへの挿入に従い順序づけられます。

$$\text{Address} = (0xF450_0000) + (0x002C)$$

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
bit Symbol	BCED															
Read/Write (HCD)	R/W															
Read/Write (HC)	R/W															
リセット後	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bit Symbol	BCED												Reserved			
Read/Write (HCD)	R/W															
Read/Write (HC)	R/W															
リセット後	0	0	0	0	0	0	0	0	0	0	0	0				

ビット	ニモニック	フィールド名	機能
[31:4]	BCED	BulkCurrentED	HC が現在の ED の処理後、ポインタは次の ED に進みます。HC は最後のフレームでリストの処理を中止したところから処理を継続します。Bulk リストの終端に到達すると、HC は HcCommandStatus レジスタの BulkListFilled フィールドをチェックします。セットされていると HC は HcBulkHeadED レジスタの内容を HcBulkCurrentED レジスタにコピーし、このビットをクリアします。セットされていない場合、HC は何も実行しません。HCD は HcControl レジスタの BulkListEnable フィールドがクリアされている時のみこのレジスタの変更が許可されます。セットされていると、HCD はこのレジスタの瞬時値を読み出すだけです。このフィールドには、最初 Bulk リスト終値を示すゼロがセットされています。
[3:0]	–	Reserved	

13. HcDoneHead レジスタ

HcDoneHead レジスタは、Done キューに追加されている最後に実行した転送ディスクリプタの物理アドレスを設定します。通常動作では、このレジスタの内容は定期的に HCCA に書き込まれるので、ホストコントローラドライバはこのレジスタを読み出す必要はありません。

Address = (0xF450_0000) + (0x0030)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
bit Symbol	DH															
Read/Write (HCD)	R															
Read/Write (HC)	R/W															
リセット後	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bit Symbol	DH												Reserved			
Read/Write (HCD)	R															
Read/Write (HC)	R/W															
リセット後	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ビット	ニモニク	フィールド名	機能
[31:4]	DH	DoneHead	TD が完了すると、HC は HcDoneHead の内容を TD の NextTD フィールドに書き込みます。HC は、次に HcDoneHead の内容をこの TD のアドレスで上書きします。HC がこのレジスタの内容を HCCA に書き込む度に、このフィールドはゼロにクリアされ、HcInterruptStatus レジスタの WritebackDoneHead フィールドがセットされます。
[3:0]	–	Reserved	

14. HcFmInterval レジスタ

HcFmInterval レジスタは、1つのフレーム内でのビットタイムの間隔(例えば、連続した2つの SOF 間)を示す 14 ビット値と、ホストコントローラがスケジューリングオーバーランなしに送信、または受信できるフルスピードの最大パケットサイズを示す 15 ビット値を設定します。ホストコントローラドライバは、SOF ごとに新しい値を現在値に上書きすることにより、FrameInterval の微調整を実行します。これにより、ホストコントローラは、外部のクロックリソースと同期を取り、不定のローカルクロックのオフセットを調整できます。

$$\text{Address} = (0xF450_0000) + (0x0034)$$

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
bit Symbol	FIT		FSMPS													
Read/Write (HCD)	R/W		R/W													
Read/Write (HC)	R		R													
リセット後	0		TBD													
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bit Symbol	Reserved		FI													
Read/Write (HCD)			R/W													
Read/Write (HC)			R													
リセット後			1	0	1	1	1	0	1	1	0	1	1	1	1	1

ビット	ニモニク	フィールド名	機能
[31]	FIT	FrameInterval Toggle	HCD は、新しい値を FrameInterval にロードするたびにこのビットをトグルします。
[30:16]	FSMPS	FSLargestData Packet	このフィールドは、各フレーム初めに最大データパケットカウンタにロードする値を指定します。カウンタ値は与えられた時刻に HC がスケジューリングオーバーランなしに 1 回のトランザクションで受信、または送信できる最大データをビットで示します。フィールド値は HCD で計算されます。
[15:14]	–	Reserved	
[13:0]	FI	FrameInterval	このフィールドは、連続した 2 つの SOF の間隔をビットタイムで指定します。標準値は 11,999 にセットされています。HCD は、HC をリセットする前にこのフィールドの現在値を保存しなければなりません。HcCommandStatus レジスタの HostControlReset フィールドをセットすることにより、HC はこのフィールドを標準値にリセットします。HCD は、Reset シーケンスの終了時に保存した値をリストアすることを選択できます。

15. HcFmRemaining レジスタ

HcFmRemaining レジスタは、現在のフレームに残っているビットタイムを示す 14 ビットのダウンカウンタです。

$$\text{Address} = (0xF450_0000) + (0x0038)$$

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
bit Symbol	FRT	Reserved														
Read/Write (HCD)	R															
Read/Write (HC)	R/W															
リセット後	0															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bit Symbol	Reserved		FR													
Read/Write (HCD)			R													
Read/Write (HC)			R/W													
リセット後			0	0	0	0	0	0	0	0	0	0	0	0	0	0

ビット	ニモニク	フィールド名	機能
[31]	FRT	FrameRemaining Toggle	このビットは、FrameRemaining が 0 になる度に HcFmInterval レジスタの FrameIntervalToggle フィールドからロードされます。このビットは、FmInterval フィールドと FmRemaining フィールド間の同期をとるために HCD により使用されます。
[30:14]	–	Reserved	
[13:0]	FR	FrameRemaining	このカウンタは各ビットタイムでデクリメントされます。ゼロに到達すると、次のビットタイム境界で HcFmInterval レジスタで指定される FrameInterval フィールド値をロードすることによってリセットされます。USBOPERATIONAL ステートに入る時、HC は HcFmInterval レジスタの FrameInterval フィールドで内容を再ロードし、次の SOF から更新された値を使用します。

16. HcFmNumber レジスタ

HcFmNumber レジスタは、16 ビットのカウンタです。ホストコントローラとホストコントローラドライバで発生するイベント間のタイミングリファレンスを提供します。ホストコントローラドライバは、このレジスタで指定されている 16 ビット値を使用し、レジスタに頻繁にアクセスすることなく 32 ビットのフレーム番号を生成できます。

$$\text{Address} = (0xF450_0000) + (0x003C)$$

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
bit Symbol	Reserved															
Read/Write (HCD)																
Read/Write (HC)																
リセット後																
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bit Symbol	FN															
Read/Write (HCD)	R															
Read/Write (HC)	R/W															
リセット後	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ビット	ニモニック	フィールド名	機能
[31:16]	–	Reserved	
[15:0]	FN	FrameNumber	このフィールドは、HcFmRemaining レジスタが再ロードされる時、インクリメントされます。0xFFFF のあと、ロールオーバーして 0x0000 になります。USBOPERATIONAL ステートに入ると、このフィールドは自動的にインクリメントされます。HC が各フレーム境界で FrameNumber をインクリメントし SOF を送付したあと、また HC がそのフレームの最初の ED を読む前に、内容は HCCA に書き込まれます。HCCA に書き込み後、HC は HcInterruptStatus レジスタの StartofFrame フィールドをセットします。

17. HcPeriodicStart レジスタ

HcPeriodicStart レジスタは、HC が周期転送リストの処理を開始する最も早い時刻を決める 14 ビットのプログラマブルな値を設定します。

$$\text{Address} = (0xF450_0000) + (0x0040)$$

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
bit Symbol	Reserved															
Read/Write (HCD)																
Read/Write (HC)																
リセット後																
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bit Symbol	Reserved		PS													
Read/Write (HCD)			R/W													
Read/Write (HC)			R													
リセット後			0	0	0	0	0	0	0	0	0	0	0	0	0	0

ビット	ニモニク	フィールド名	機能
[31:14]	–	Reserved	
[13:0]	PS	PeriodicStart	ハードウェアリセット後、このフィールドはクリアされます。HC 初期化の間、このフィールドは HCD によって設定されます。値は概算で HcFmInterval レジスタの値から 10%削減して計算されます。代表値は 0x3E67 です。HcFmRemaining レジスタが指定された値に到達すると、周期転送リストの処理は Control リストと Bulk リストの処理より優先されます。その為、HC は処理中の現在の Control または Bulk トランザクションの終了後に、Interrupt リストの処理を開始します。

18. HcLSThreshold レジスタ

HcLSThreshold レジスタは、ホストコントローラによって使用され、EOF 前に最大 8 バイトの LS パケット転送を実行するかどうかを指定する 11 ビットの値を含みます。ホストコントローラもホストコントローラドライバもこの値は変更できません。

$$\text{Address} = (0xF450_0000) + (0x0044)$$

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
bit Symbol	Reserved															
Read/Write (HCD)																
Read/Write (HC)																
リセット後																
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bit Symbol	Reserved				LST											
Read/Write (HCD)					R/W											
Read/Write (HC)					R											
リセット後					0	1	1	0	0	0	1	0	1	0	0	0

ビット	ニモニック	フィールド名	機能
[31:12]	–	Reserved	
[11:0]	LST	LSThreshold	このフィールドは、ロースピードトランザクションを起動する前に FrameRemaining フィールドと比較される値を設定します。FrameRemaining フィールドの値が、このフィールドの値以上の時のみトランザクションは開始されます。値はトランスミッションとセットアップオーバーヘッドを考慮して HCD によって計算されます。

19. HcRhDescriptorA レジスタ

HcRhDescriptorA レジスタはルートハブの特長を記述する 2 つのレジスタの 1 つです。リセット値はインプリメントに固有です。ハブクラスディスクリプタの descriptor length、descriptor type、hub descriptor current フィールドは HCD によってエミュレートされます。全ての他のフィールドは、HcRhDescriptorA レジスタと HcRhDescriptorB レジスタに割り当てられます。

Address = (0xF450_0000) + (0x0048)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
bit Symbol	POTPGT								Reserved							
Read/Write (HCD)	R/W															
Read/Write (HC)	R															
リセット後	0	0	0	0	0	0	1	0								
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bit Symbol	Reserved			NOCP	OCPM	DT	NPS	PSM	NDP							
Read/Write (HCD)				R/W	R/W	R	R/W	R/W	R							
Read/Write (HC)				R	R	R	R	R	R							
リセット後				0	0	0	0	0	0	0	0	0	0	0	0	1

ビット	ニモニク	フィールド名	機能
[31:24]	POTPGT	PowerOnTo PowerGoodTime	このバイトは、ルートハブのパワーオン状態のポートをアクセスする前に、HCD がウェイトしなければならない期間をしています。インプリメントに固有です。時間の単位は 2ms です。期間は POTPGTx2ms として計算されます。
[23:13]		Reserved	
[12]	NOCP	NoOverCurrent Protection	このビットはルートハブポート用の過電流状態をどの様にレポートするかを記述します。このビットがクリアされる時、OverCurrentProtectionMode フィールドは全体、またはポートごとのレポートを指定します。 0y0: 過電流状態は、全ての下流ポートが纏めてレポートされます。 0y1: 過電流保護はサポートされません。
[11]	OCPM	OverCurrent ProtectionMode	このビットはルートハブポートの過電流状態をどの様にレポートするかを記述します。リセット時、このフィールドは PowerSwitchingMode と同じモードを反映しなければなりません。このフィールドは NoOverCurrent Protection フィールドがクリアされている時のみ有効です。 0y0: 過電流状態は、全ての下流ポートが纏めてレポートされます。 0y1: ポート毎に過電流状態はレポートされます。
[10]	DT	DeviceType	このビットは、ルートハブはコンパウンドデバイスではないことを指定します。ルートハブはコンパウンドデバイスであることを許されません。このフィールドは常に読み書き共に 0 です。
[9]	NPS	NoPower Switching	これらのビットはパワースイッチングがサポートされているか、またはポートが常にパワーオン状態かを指定します。インプリメントに固有です。このビットがクリアされると PowerSwitchingMode フィールドは全体の、またはポート毎の切り替えを指定します。 0y0: ポートがパワースイッチされます。 0y1: HC がパワーオン状態のとき、ポートは常にパワーオン状態です。
[8]	PSM	PowerSwitching Mode	このビットはルートハブポートのパワースイッチングをどの様に制御するかを指定するのに使用されます。インプリメントに固有です。このフィールドは NoPower Switching フィールドがクリアされる場合のみに有効です。 0y0: 全てのポートは同時にパワーオンされます。 0y1: 各ポートは個別にパワーオンされます。このモードにより、ポートのパワーオンは全体のスイッチ、またはポート毎のスイッチにより制御されます。HcRhDescriptorB レジスタの PortPowerCurrentMask ビットがセットされている場合、ポートはポートパワーコマンド(Set/Clear PortPower)だけに応答します。ポートマスクがクリアされている場合、ポートは全体のパワースイッチ(Set/ClearGlobalPower)によってのみ制御されます。
[7:0]	NDP	Number DownstreamPorts	これらのビットは、ルートハブだけにサポートされている下流のポート数を指定します。インプリメントに固有です。このモジュールのポート数は 1 なので、0x01 が読み出されます。

20. HcRhDescriptorB レジスタ

HcRhDescriptorB レジスタは、ルートハブの特長を記述する 2 つのレジスタの 1 つです。これらのフィールドは、システム動作に対応して初期化中に書き込まれます。リセット値はインプリメントに固有です。

$$\text{Address} = (0xF450_0000) + (0x004C)$$

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
bit Symbol	PPCM															
Read/Write (HCD)	R/W															
Read/Write (HC)	R															
リセット後	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bit Symbol	DR															
Read/Write (HCD)	R/W															
Read/Write (HC)	R															
リセット後	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ビット	ニモニック	フィールド名	機能
[31:16]	PPCM	PortPower ControlMask	各ビットは HcRhDescriptorA レジスタの PowerSwitchingMode がセットされている時、ポートがグローバルパワーコントロールコマンドで制御されるかどうかを示します。セットされている時、ポートのパワー状態はポート毎のパワーコントロール(Set/ClearPortPower)によって制御されます。クリアされている時、ポートはグローバルパワースイッチ(Set/ClearGlobalPower)によって制御されます。デバイスがグローバルスイッチングモード(PowerSwitchingMode=0)で構成されている場合、このフィールドは無効です。 bit0: Reserved bit1: Ganged-power mask on Port#1
[15:0]	DR	Device Removable	各ビットは、ルートハブのポートを示しています。 0 の時接続されたデバイスは括線挿抜が可能です。 1 の時接続されたデバイスは括線挿抜ができません。 bit0: Reserved bit1: Device attached to Port#1

21. HcRhStatus レジスタ

HcRhStatus レジスタは、2 つのフィールドに区分されます。Dword の下位のワードは、Hub Status フィールドを示し、上位のワードは Hub Status Change フィールドを示します。予約ビットは常に 0 でなければなりません。

$$\text{Address} = (0xF450_0000) + (0x0050)$$

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
bit Symbol	CRWE	Reserved													OCIC	LPSC
Read/Write (HCD)	W														R/W	R/W
Read/Write (HC)	R														R/W	R
リセット後	—														0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bit Symbol	DRWE	Reserved													OCI	LPS
Read/Write (HCD)	R/W														R	R/W
Read/Write (HC)	R														R/W	R
リセット後	0														0	0

ビット	ニモニック	フィールド名	機能
[31]	CRWE	ClearRemote WakeupEnable	1 を書き込むと DeviceRemoteWakeUpEbable をクリアします。0 を書き込んでも何の影響も与えません。
[30:18]	–	Reserved	
[17]	OCIC	OverCurrent Indicator Change	このビットは、このレジスタの OCI フィールドで変更が発生したとき、ハードウェアによってセットされます。HCD は 1 を書き込むことにより、このビットをクリアします。0 を書き込んでも何の影響も与えません。
[16]	LPSC	LocalPower StatusChange	(read)LocalPowerStatusChange ルートハブは LocalPowerStatusChange をサポートしません。そのためこのビットは常に 0 として読み込まれます。 (write)SetGlobalPower グローバルパワーモード(PowerSwitchingMode=0)では、このビットは全てのポートをパワーオンするため(PortPowerStatus をクリア)1 が書き込まれます。ポート毎のパワーモードでは、PortPowerControlMask ビットがセットされていないポートだけに PortPowerStatus をセットします。0 を書き込んでも何の影響も与えません。
[15]	DRWE	DeviceRemote WakeupEnable	(read)DeviceRemoteWakeUpEnable このビットは ConnectStatusChange ビットをレジュームイベントとしてイネーブルし、USBSUSPEND ステートを USBRESUME ステートに遷移させ、ResumeDetected 割り込みをセットします。 0y0: ConnectStatusChange はリモートウェイクアップイベントではありません。 0y1: ConnectStatusChange はリモートウェイクアップイベントです。 (write) 1 を書き込むと、DeviceRemoteWakeUpEnable をセットします。0 を書き込んでも何の影響も与えません。
[14:2]	–	Reserved	
[1]	OCI	OverCurrent Indicator	このビットは、グローバルレポートが実行されるとき電流をレポートします。セット時、過電流状態が存在します。クリア時は定電流状態です。ポート毎の過電流保護を実行する場合は、このビットは常に 0 にします。
[0]	LPS	LocalPower Status	(read)LocalPowerStatus ルートハブは、LocalPowerStatus をサポートしません。そのためこのビットは常に 0 として読み込まれます。 (write)ClearGlobalPower グローバルパワーモード(PowerSwitchMode=0)では、このビットは 1 が書き込まれ、全てのポートをパワーオフします(PortPowerStatus をクリア)。ポート毎のパワーモードでは、PortPowerControlMask ビットがセットされていないポートだけで PortPowerStatus をクリアします。 0 を書き込んでも何の影響も与えません

22. HcRhPortStatus レジスタ

HcRhPortStatus レジスタは、ポート毎のポートイベントを制御し、レポートするのに使用されます。HcRhDescriptorA レジスタの NumberDownstreamPorts は、ハードウェアで実行される HcRhPortStatus レジスタ数を示します。下位のワードはポートステータスを表すのに使用され、上位のワードはステータスチェンジビットを表します。幾つかのステータスビットは以下に示す特別な書き込み動作により実行されます。ポートステータスを変更する書き込みが発生したとき、トランザクション(ハンドシェイクによるトークン)が実行中の場合は、発生したポートステータス変更はトランザクション完了後に実行されます。予約ビットは常に 0 が書き込まれます。

$$\text{Address} = (0xF450_0000) + (0x0054)$$

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
bit Symbol	Reserved											PRSC	OCIC	PSSC	PESC	CSC
Read/Write (HCD)												R/W	R/W	R/W	R/W	R/W
Read/Write (HC)												R/W	R/W	R/W	R/W	R/W
リセット後												0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bit Symbol	Reserved						LSDA	PPS	Reserved			PRS	POCI	PSS	PES	CCS
Read/Write (HCD)							R/W	R/W				R/W	R/W	R/W	R/W	R/W
Read/Write (HC)							R/W	R/W				R/W	R/W	R/W	R/W	R/W
リセット後							X	0				0	0	0	0	0

ビット	ニモニック	フィールド名	機能
[31:21]	–	Reserved	
[20]	PRSC	PortResetStatus Change	このビットは、10ms ポートリセット信号の終了時にセットされます。HCD はこのビットをクリアするのに 1 を書き込みます。0 を書き込んでも何の影響も与えません。 0y0: ポートリセットが完了していません。 0y1: ポートリセットが完了しました。
[19]	OCIC	PortOverCurrent IndicatorChange	このビットは過電流状態がポート毎にレポートされるとき有効です。ルートハブが PortOverCurrentIndicator ビットを変更するとき、このビットがセットされます。HCD は 1 を書き込んでこのビットをクリアします。0 を書き込んでも何の影響も与えません。 0y0: PortOverCurrentIndicator に変更はありません。 0y1: PortOverCurrentIndicator が変更されました。
[18]	PSSC	PortSuspend StatusChange	フルレジュームシーケンスが完了したとき、このビットがセットされます。このシーケンスは、20s レジュームパルス、LS EOP、3ms 再同期遅延で構成されます。HCD は 1 を書き込んでこのビットをクリアします。0 を書き込んでも何の影響も与えません。ResetStatusChange がセットされるとき、このビットもクリアされます。 0y0: レジュームが完了していません。 0y1: レジュームが完了しました。

ビット	ニモニック	フィールド名	機能
[17]	PESC	PortEnable StatusChange	ハードウェアイベントが PortEnableStatus ビットをクリアさせる原因となるとき、このビットがセットされます。HCD 書込みからの変更は、このビットをセットしません。HCD は 1 を書き込んでこのビットをクリアします。0 を書き込んで何の影響も与えません。 0y0: PortEnableStatus に変更はありません。 0y1: PortEnableStatus が変更されました。
[16]	CSC	ConnectStatus Change	コネクトまたはディスコネクトイベントが発生するたびにこのビットがセットされます。HCD は 1 を書き込んでこのビットをクリアします。0 を書き込んで何の影響も与えません。SetPortReset、SetPortEnable、または SetPortSuspend 書込みが発生したときに、CurrentConnectStatus がクリアされていると、ポートが未接続の場合これらの書込みは発生しないため、この bit がセットされ強制的にドライバに接続状態を再確認させます。 0y0: CurrentConnectStatus に変更はありません。 0y1: CurrentControlStatus が変更されました。 (注) DeviceRemovable[NDP]ビットがセットされている場合、このビットはルートハブリセットのみセットされ、システムにデバイスがアタッチされていることを知らせます。
[15:10]		Reserved	
[9]	LSDA	LowSpeed DeviceAttached	(read)LowSpeedDeviceAttached このビットはこのポートにアタッチされているデバイスのスピードを示します。セット時、Low スピードデバイスがこのポートにアタッチされます。クリア時、Full スピードデバイスがアタッチされます。このフィールドは、CurrentConnectStatus がセットされているときのみ有効です。 0y0: Full スピードデバイスがアタッチされています。 0y1: Low スピードデバイスがアタッチされています。 (write)ClearPortPower HCD はこのビットに 1 を書き込んで PortPowerStatus ビットをクリアします。0 を書き込むと何の影響も与えません。

ビット	ニモニック	フィールド名	機能
[8]	PPS	PortPower Status	(read)PortPowerStatus このビットは、実行されるパワースイッチングタイプに関係なく、ポートのパワーステータスを示します。過電流状態が検出されると、このビットはクリアされます。HCD は、SetPortPower または SetGlobalPower を書き込むことによりこのビットをセットします。HCD は ClearPortPower または ClearGlobalPower を書き込むことにより、このビットをクリアします。どのパワーコントロールスイッチがイネーブルされるかは、PortSwitchingMode と PortPowerControlMask[NDP]により決定されます。グローバルスイッチングモード (PowerSwitchingMode=0) では、Set/ClearGlobalPower だけがこのビットを制御します。ポート毎のパワースイッチング (PowerSwitchingMode=1) では、ポート用の PortPowerControlMask[NDP]ビットがセットされている場合、Set/ClearPortPower コマンドのみイネーブルされます。マスクがセットされていない場合、Set/ClearGlobalPower コマンドのみがイネーブルされます。ポートパワーがディセーブルされるとき、CurrentConnectStatus、PortEnableStatus、PortSuspendStatus、PortResetStatus がリセットされなければなりません。 0y0: ポートパワーオフ 0y1: ポートパワーオン (write)SetPortPower HCD は 1 を書き込んで PortPowerStatus ビットをセットします。0 を書き込んでも何の影響も与えません。 (注) パワースイッチングがサポートされていない場合、このビットは常に 1 を読み込みます。
[7:5]	–	Reserved	
[4]	PRS	PortReset Status	(read)PortResetStatus このビットが SetPortReset への書込みによりリセットされるとき、ポートリセット信号がアサートされます。リセットが完了すると、PortResetStatusChange がセットされている時このビットはクリアされます。CurrentConnectStatus がクリアされている場合、このビットをセットできません。 0y0: ポートリセット信号がインアクティブです 0y1: ポートリセット信号がアクティブです。 (write)SetPortReset HCD はこのビットに 1 を書き込んでポートリセット信号をセットします。0 を書き込んでも何の影響も与えません。CurrentConnectStatus がクリアされている場合、この書込みは PortResetStatus をセットせず、代わりに CurrentConnectStatusChange をセットします。これにより、未接続のポートをリセットしようとしたことをドライバに報告します。

ビット	ニモニック	フィールド名	機能
[3]	POCI	PortOverCurrent Indicator	(read)PortOverCurrentIndicator このビットは過電流状態がポート毎にレポートされる方法でルートハブが構成されているときのみ有効です。ポート毎に過電流レポートがサポートされていない場合、このビットは0にクリアされます。クリアされた場合、このポートは定電流状態です。セットされた場合、過電流状態がこのポートに存在します。このビットは、常に過電流入力信号を示します。 0y0: 過電流状態ではありません。 0y1: 過電流状態が検出されました。 (write)ClearSuspendStatus HCDは1を書込みレジューム起動します。0を書き込んでも何の影響も与えません。PortSuspendStatusがセットされている時のみレジューム起動します。
[2]	PSS	PortSuspend Status	(read)PortSuspendStatus このビットはポートがサスペンド状態か、またはレジュームシーケンス中にあることを示します。SetSuspendState 書込みでセットされレジューム終了時点で PortSuspendStatusChange がセットされるときクリアされます。このビットは CurrentConnectStatus がセットされるときクリアされます。このビットはポートリセットの終了時点で PortResetStatusChange がセットされているときまたは HC が USBRESUME 状態にあるときクリアされます。 0y0: ポートがサスペンド状態ではありません。 0y1: ポートがサスペンド状態です。 (write)SetPortSuspend HCDはこのビットに1を書き込んで PortSuspendStatus ビットをセットします。0を書き込んでも何の影響も与えません。 CurrentConnectStatus がクリアされている場合、この書込みは PortSuspendStatus をセットしないで代わりに ConnectStatusChange をセットし、未接続のポートをサスペンドしようとしたことをドライバに報告します。

ビット	ニモニック	フィールド名	機能
[1]	PES	PortEnable Status	(read)PortEnableStatus このビットは、ポートがイネーブル状態であるか、またはディセーブル状態であるかを示します。ルートハブは、過電流状態、切断イベント、パワースイッチオフ状態、Babble などのオペレーションエラーが検出されると、このビットをクリアします。このビットがクリアされることにより、PortEnabledStatusChange がセットされます。HCD は SetPortEnable に書き込んでこのビットをセットし、ClearPortEnable に書き込んでクリアします。CurrentConnectStatus がクリアされているとき、このビットはセットできません。ResetStatusChange がセットされているときはポートリセットの終了時、または SuspendStatusChange がセットされているときはポートサスペンドの終了時にこのビットはセットされます。 0y0: ポートがディセーブル状態です。 0y1: ポートがイネーブル状態です。 (write)SetPortEnable HCD は 1 を書き込むことにより PortEnableStatus をセットします。0 を書き込むと何の影響も与えません。CurrentConnectStatus がクリアされている場合、この書き込みは PortEnableStatus をセットしないで、代わりに ConnectStatusChange をセットし、未接続のポートをイネーブルしようとした事をドライバに報告します。
[0]	CCS	CurrentConnect Status	(read)CurrentConnectStatus このビットは下流ポートの現在のステータスを反映します。 0y0: デバイスが未接続状態です。 0y1: デバイスが接続状態です。 (write)ClearPortEnable HCD はこのビットに 1 を書き込んで PortEnableStatus ビットをクリアします。0 を書き込んで何の影響も与えません。CurrentConectStatus はどのような書き込みによっても影響を受けません。 (注) アタッチされたデバイスが外せないとき(Device Removeable[NDP])、このビットは常に 1 を読み出します。

23. HcBCR0 レジスタ

HcBCR0 レジスタは、USBブリッジ回路から USB ホストコアへのクロック供給制御と、USB トランシーバの SUSPEND 制御を行うためのレジスタです。また USB トランシーバを SUSPEND 状態にしてパワーカットモードに移行する場合は、TRNS_SUSP ビットに 1 をライトします。

$$\text{Address} = (0xF450_0000) + (0x0080)$$

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
bit Symbol	Reserved	TRNS_SUSP	OVCE	Reserved												
Read/Write	R	R/W	R/W	R												
リセット後	0	1	0	0												
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bit Symbol	Reserved														Reserved	Reserved
Read/Write	R														R/W	R/W
リセット後	0														0	0

ビット	ニモニック	フィールド名	機能
[31]	–	Reserved	
[30]	TRNS_SUSP	Transceiver Suspend	USB トランシーバの SUSPEND 制御ビットです。 USB トランシーバを SUSPEND 状態にして STOP モード、もしくはパワーカットモードに移行する場合は、1 にセットします。 0y0: -(USB ホストコントローラが制御します) 0y1: Suspend
[29]	OVCE	USB Host Over Current Input Enable	USB Host Over Current input Enable 0y0: Enable 0y1: Disable
[28:2]	–	Reserved	
[1:0]	–	Reserved	Write as zero

注) USB ホストコントローラの過電流検知機能を使用しない場合、USBOCn 端子は出力ポートとしてご使用頂くか、出力固定にしてください。

3.25.7 使用上の注意点

3.25.7.1 USBクロック設定時の注意点

USB Host をアクセスする前に、クロックコントローラのレジスタ CLKCR5<USBH_CLKEN>=0y1 に設定してください。

USB クロックとして、X1USB、f_{PLL}の4分周または3分周のいずれかを選択できます。
選択方法は以下の順番にて行って下さい。

(1) X1USB を使用する場合

```
SYSCR8 ← 0yXXXX_X001      ; USBH_CLKSEL=0y001
                                   ; X1USB 端子クロック選択
CLKCR5 ← 0yXXX1_XXXX        ; USBH_CLKEN=1
                                   ; USB HOST 用クロックイネーブル
```

注1) X1USB が安定している状態で設定を行って下さい。

注2) パワーカットモード(PCM)へ移行する際は、CLKCR5<USBH_CLKEN>=0y0 に設定してから、モードへ移行して下さい。またパワーカットモード解除後は、X1USB が安定している状態で再度 X1USB に設定を行って下さい。

注3) USB を停止する場合は、CLKCR5<USBH_CLKEN>=0y0 に設定して下さい。

(2) リセット解除後(PLL 使用)、f_{PLL}の4分周を使用する場合

X1=24MHz、8 通倍の場合

```
SYSCR8 ← 0yXXXX_X100      ; USBH_CLKSEL=0y100
                                   ; PLL 出力クロックの4分周クロック選択
CLKCR5 ← 0yXXX1_XXXX        ; USBH_CLKEN=1
                                   ; USB HOST 用クロックイネーブル
```

注) f_{PLL} 使用時、PLL クロックが安定している状態で設定を行って下さい。

(3) リセット解除後(PLL 使用)、f_{PLL}の3分周を使用する場合

X1=24MHz、6 通倍の場合

```
SYSCR8 ← 0yXXXX_X010      ; USBH_CLKSEL=0y010
                                   ; PLL 出力クロックの3分周クロック選択
CLKCR5 ← 0yXXX1_XXXX        ; USBH_CLKEN=1
                                   ; USB HOST 用クロックイネーブル
```

注) f_{PLL} 使用時、PLL クロックが安定している状態で設定を行って下さい。

3.25.7.2 発振子の注意点

本製品に内蔵される USB ホストコントローラを利用する場合は、USB の規程から $24\text{MHz} \pm 100\text{ppm}$ 以下(クリスタル発振子)の発振子を推奨いたします。

なお、内蔵 PLL により USB クロックを生成する場合、実装環境、条件、ばらつきによっては、USB で規定されているスペックを満たせない場合があります。

USB コンプライアンス(USB ロゴ)の認証を受ける場合は、必ず X1USB 端子より $\pm 100\text{ppm}$ 以下の精度で 48MHz のクロックを入力する必要があります。

3.25.7.3 PMC遷移時の注意点

PMC 状態へ遷移する前にサスペンド状態へ遷移させてください。

PMC 状態への遷移により、USB ホストコントローラの電源である DVCC1A, AVCC3H は電源を外部回路にて遮断してください。

3.25.8 USBホストコントローラ制約事項

1. アイソクロナス転送は、ディスクリプタ (ITD) 内において転送を実行すべき Frame No が定義されています。しかし、Frame No が Host とソフトウェア間で同期がとれておらず、過去の Frame で実行されるべきディスクリプタが、現在以降にスケジューリングされた場合、Host は Time Error が発生したと判断し、DATAOVERRUN を ITD の CC へ Write Back します。しかし、以下で述べる発生条件が成立する場合、不適切なステータス (NOERROR) を Write Back してしまいます。

＜発生条件＞

以下の関係が全て成立するようにスケジューリングされている場合。

1. $ITD.FC[2:0] = R[2:0]$
2. $ITD.FC[2:0] < R[15:0]$

ここで、ITD.FCとはITDの実行回数を示します。

また、Rは $R = HcFmNumber$ (現在のFrame No.) $- ITD.SF$ (転送開始Frame No.) で表されます。

現在のFrame No.と同期がとれていないかこのITDをリンクする様なソフトウェア構成にならない様に設計して下さい。

2. Host が Low Speed IN 転送を実行する場合、Host の Data packet 受信から Handshake packet 送信までの時間(Inter-packet Delay)として USB 2.0 仕様上では 7.5 bit times 未満と定義されていますが、本製品ではワーストで約 9.2 bit times となります。
3. USB システム上で致命的なエラーが発生し、Host がそれを検出する場合 (例: PCI Bus 上での Master Abort、Target Abort のような fatal error)、OHCI コアは HcInterruptStatus.

UnrecoverableError (UE)ビットをセットします。

Unrecoverable Error (UE) ビットがセットされ、かつ HcInterruptEnable.UE ビットがセットされていた場合、ハードウェア割りこみが発生します。

この割りこみを検出後、ソフトウェアリセット (HcCommandStatus.HCR = 1'b1) を行うことによって UE から復帰し、ホストは Suspend ステートに遷移します。

ソフトウェアリセット後、OHCI レジスタが初期化されますが、初期化後デバイスからの Remote Wake-up が発生した場合は、Suspend ステート内にとどまってしまう。

Remote Wake Up 機能をご使用になる場合は、ソフトウェアにより、Suspend からの復帰プログラムを実装して下さい。

ソフトウェア実施例

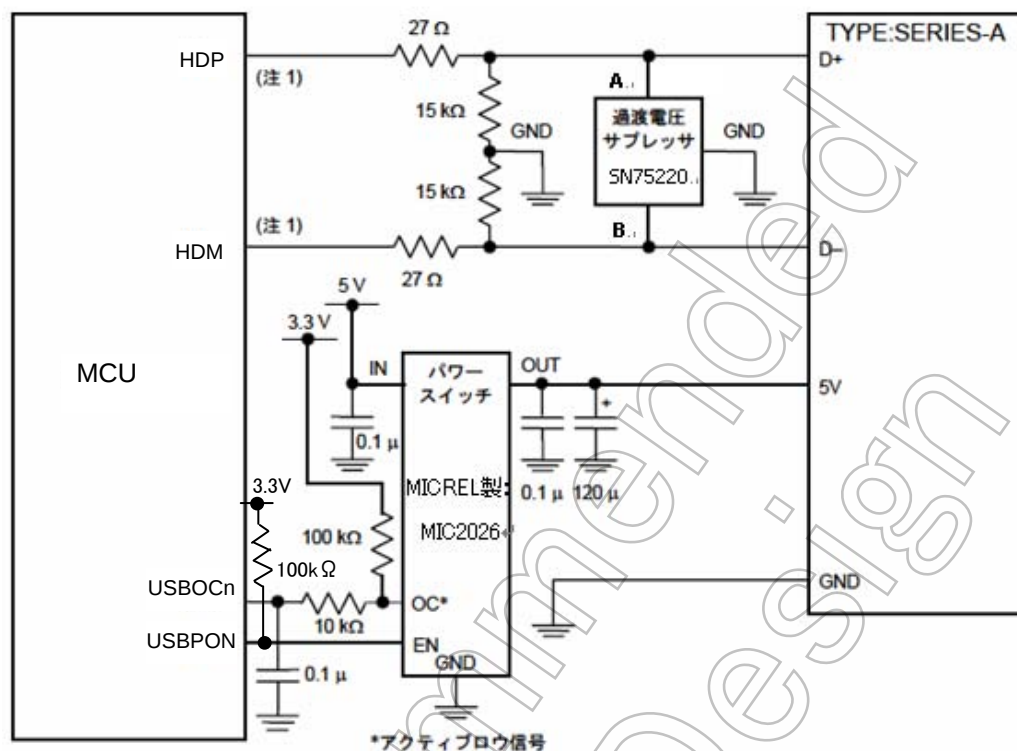
- 1)ソフトウェアリセットを実施しOHCIレジスタの初期化後、HcControl.HCFSを
USBSUSPEND (2'b11) 以外にセットする。

- 2)Remote Wake-up が検出されると HcInterruptStatus.RD = 1'b1 となります。

この割り込みを検出後、HcControl.HCFS を USBSUSPEND (2'b11) 以外にセットする。

4. Over Current 発生時、HcRhDescriptorA.NPS[9]に 1'b1 がセットされていると、HcRhPortStatus レジスタの PortResetStatus.PRS[4]、PortSuspendStatus.PSS[2]がクリアされません。そのため、HcRhDescriptorA.NPS[9] = 1'b1 および HcRhDescriptorB.DR[PortNo] = 1'b1 に設定しないようにしてください。
5. Full-Speed/Low-Speed デバイス 接続の際、HcRhStatus.DRWE[15]をセットしている場合、Remote Wake-up イベントが発生しても、Status が USBSUSPEND から USBRESUME へ遷移いたしません。
Remote Wake-up イベントを使用する場合は、本制約により Status が USBSUSPEND から USBRESUME へ遷移されませんが、HcInterruptStatus.RD[3]はセットされるため、Driver にてこのビットをみて Status を遷移させるようにしてください。
Remote Wake-up イベントを使用しない場合は、HcRhStatus.DRWE[15]に 1'b1 をセットしないようにしてください。
6. OverCurrent をサポートするシステムの場合、HcRhDescriptorA レジスタの NoOverCurrentProtection.NOCP[12]を 1'b0 に設定するようにしてください。
7. USB ホストコントローラの過電流検知機能を使用しない場合、PT5 (USBOC) 端子は出力ポートとしてご使用頂くか、出力固定にしてください。
8. Isochronous 転送を使用する場合はスケジュールオーバーランを発生させないように使用してください。
9. ポートリセット実施後にポートディセーブル状態になった場合、再度ポートリセットを実施してください。

3.25.9 回路接続例



バスパワースイッチデバイス例：Texas Instruments 社 TPS2052

MICREL 社 MIC2526-1BM

MICREL 社 MIC2536-1BN

過渡電圧サプレッサデバイス例：Texas Instruments 社 SN75240

抵抗精度：5%

抵抗率 : 27Ω に対して $1/2W$ を推奨します。

キャパシタ：120uF キャパシタに対して、低 ESR タイプ(OS-CON など)を推奨します。

注 1) 絶対最大値を超えた電圧を適用してはいけません。

注2) HDP、HDM 端子から、それぞれ USB A レセプタクルまでの配線長が同じになる様に基板設計を行って下さい。

注3) サプレッサデバイスは、USB 仕様上は必要ありません。

注 4) リセット解除後の USBPON、USBOn 端子初期状態はポート入力モードです。外部回路にて端子処理を行なって下さい。

3.26 周波数検知回路 (OFD: Oscillation Frequency Detector)

3.26.1 概要

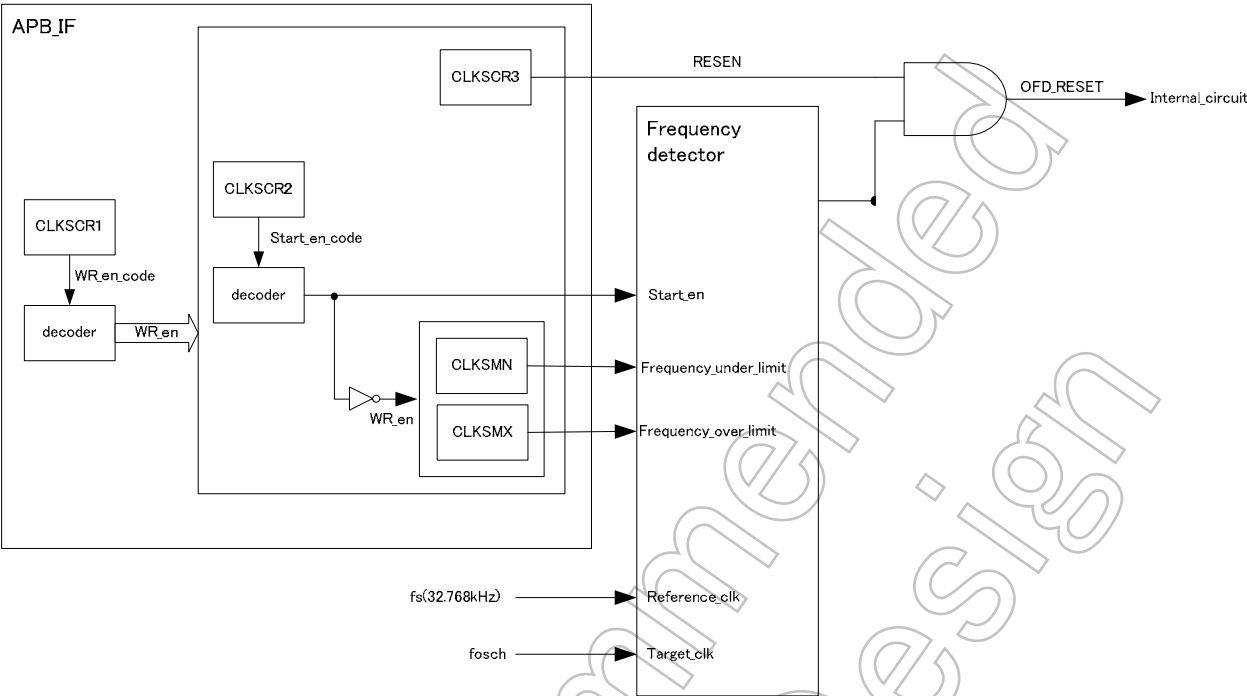
周波数検知回路(Oscillation Frequency Detector)は、高速発振の周波数(f_{OSCH})が検知周波数設定レジスタによって設定された周波数範囲を超えた場合に内部リセットを発生する回路です。

また、高速発振器の周波数(f_{OSCH})は検出クロック、32kHz 低速クロック (f_L)は OFD 回路の基準クロックとなっています。また、32KHz の基準クロックが停止した場合も異常として認識され、内部回路をリセット状態にします。

注) PCM モードでは OFD 回路は電源供給されておらず、動作しません。また、PCM を実行する前に OFD 機能を停止してください。

Not Recommended
for New Design

3.26.2 OFD Block



3.26.3 動作説明

1) リセットの発生と解除

OFD 回路は高周波クロック(f_{osch})が異常で、周期比が CLKSMN と CLKSMX に設定されている範囲外になった場合(停止も含む)、低周波クロック(f_s)の 2~3 回目の立ち上がりエッジでリセットが発生します。クロックが正常な周波数帯域に戻るまでリセットは継続してアサートされます。

低周波クロックの場合は、クロックが停止するとリセットが発生します。低周波クロックが発振を再開するまでリセットが解除されません。また、 f_s , f_{osch} 共に周波数が不安定な場合、動作を保障することはできません。

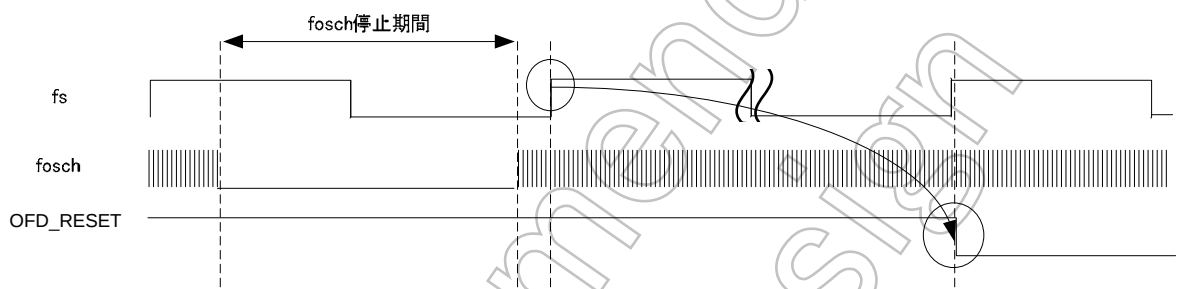


図 3.26.1 リセット発生タイミング 1 (f_{osch} が停止した場合)

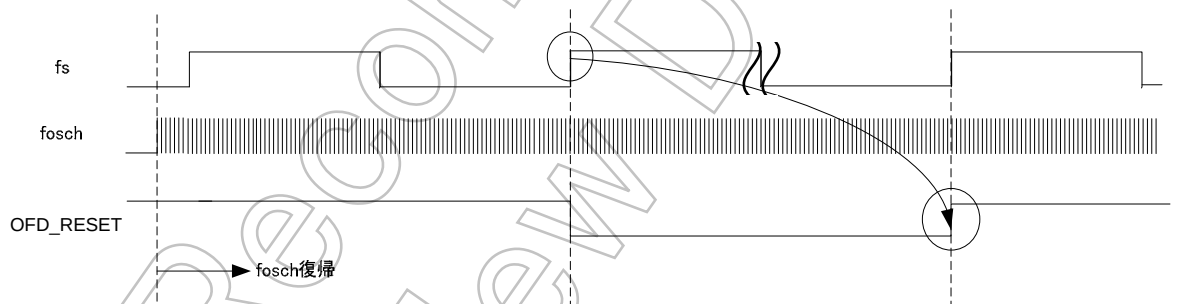


図 3.26.2 リセット解除タイミング 1 (f_{osch} が停止した場合)

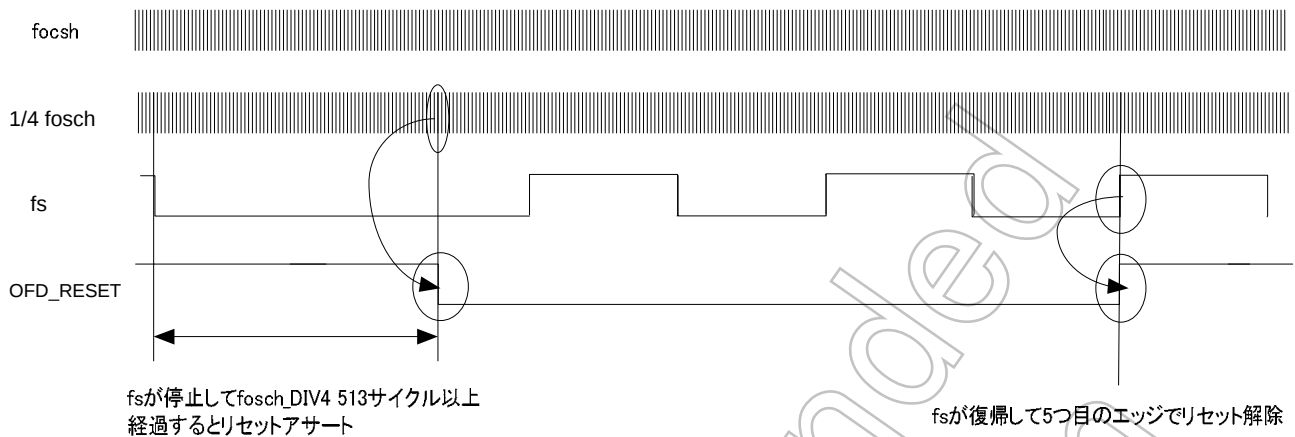


図 3.26.3 リセット発生/解除タイミング2 (fs が停止した場合)

注) 高周波または低周波のどちらか一方のクロック周波数がノイズなどの影響で一時的に周期比の設定範囲外になった場合もクロック停止検出リセットが発生します。

2) OFD 回路の初期化

- OFD 回路(OFD のレジスタを含む)の初期化について
外部リセット(RESETn 端子への“L”入力)により初期化されます。
PCM モードから復帰する時、初期化されます。

注) WDT リセット及び OFD リセットでは初期化されません。

3) クロック周波数の異常検知確認方法

OFD 検知フラグ(CLKSCR3<CLKSF>=1)により OFD 回路が異常を検出したことを確認できます。

4) クロック検出の最小値/最大値の設定方法

設定周波数 = {(使用する発振子の周波数) + (使用する発振子の誤差) + (検知回路誤差) + α } を計算し、その値から実際にレジスタ (CLKSMN および CLKSMX) に設定する値を計算します。

以下に $f_{OSCH}=24\text{MHz}$ の場合、計算式と設定例を示します。

- 設定周波数の計算例

例えば、発振子の誤差、検知回路の誤差 (約 $\pm 7\%$) などを十分考慮して f_{OSCH} の総合誤差を 10% とした場合

設定周波数 (Low) : $24\text{MHz} \times (1-10\%) = 21.6\text{MHz}$ 、

設定周波数 (High) : $24\text{MHz} \times (1+10\%) = 26.4\text{MHz}$

- CLKSMN および CLKSMX の設定例

設定周波数 (Low) : $24\text{MHz} \times (1-10\%) = 21.6\text{MHz}$ 、

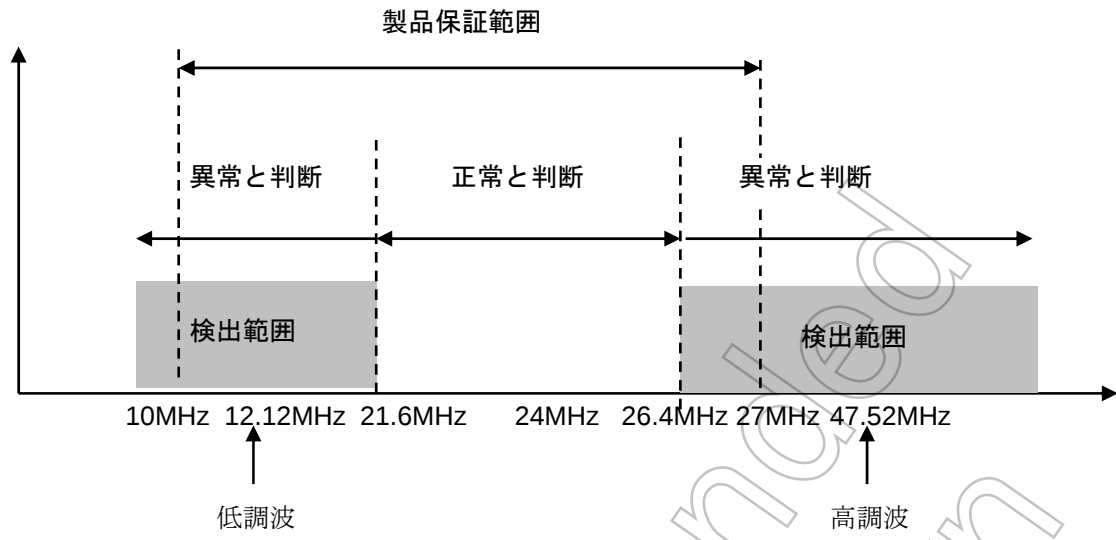
設定周波数 (High) : $24\text{MHz} \times (1+10\%) = 26.4\text{MHz}$

発振クロックの周波数は検知クロックの設定範囲外になると、OFD リセットがアサートされます。

$$\text{CLKSMN 設定値} = \frac{\text{設定周波数(Low)}}{f_s \times 4} = \frac{21.6}{32.768 \times 10^{-3} \times 4} = 165 = 0xA5 \quad (\text{端数は切り上げ})$$

$$\text{CLKSMX 設定値} = \frac{\text{設定周波数(High)}}{f_s \times 4} = \frac{26.4}{32.768 \times 10^{-3} \times 4} = 201 = 0xC9 \quad (\text{端数は切り捨て})$$

注) 本周波数検知回路は発振周波数の高周波、低周波発振を検知するための回路です。発振子および発振器の発振精度誤差を測定する回路ではありません。



検知する周波数：f_{OSCH}の低調波および高調波の計算例（発振子の周波数保証誤差±1%の場合）

低調波f_c（高速発振子誤差のMaxを考慮）＝24MHz × (1+0.01) ÷ 2 ＝ 12.12MHz

高調波f_c（高速発振子誤差のMinを考慮）＝24MHz × (1-0.01) × 2 ＝ 47.52MHz

下記に 24MHz、10MHz の発振子を使用した場合の設定推奨例を示します。推奨例は発振子の誤差及び、検知回路の総合誤差を 10%として計算しています。

f _{OSCH}	上段：低調波	上段：推奨 CLKSMN 設定値	上段：検知周波数範囲(Low 側)
	下段：高調波	下段：推奨 CLKSMX 設定値	下段：検知周波数範囲(High 側)
24MHz	12.12 MHz	0xA5	4.0 MHz ～ 21.6MHz および停止
	47.52 MHz	0xC9	26.4 MHz ～ 60MHz (MAX)
10MHz	5.05 MHz	0x45	4.0 MHz ～ 9.0MHz および停止
	19.8 MHz	0x54	11.0 MHz ～ 60MHz (MAX)

注) 高速発振器の動作保証範囲については 4 章 電気的特性を参照してください。

3.26.4 SFR

Base address = 0xF009_0000

Register Name	Address (base+)	Description
CLKSCR1	0x0000	周波数検知回路コントロールレジスタ 1
CLKSCR2	0x0004	周波数検知回路コントロールレジスタ 2
CLRSCR3	0x0008	周波数検知回路コントロールレジスタ 3
CLKSMN	0x0010	検知周波数設定レジスタ (Low 側)
CLKSMX	0x0020	検知周波数設定レジスタ (High 側)

Not Recommended
for New Design

1. CLKSCR1 (周波数検知回路コントロールレジスタ 1)

Address = (0xF009_0000)+(0x0000)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:0]	CLKWEN	R/W	0x06	OFD register Write Enable Code: 0x06: Disabling of writing to CLKSCR2/CLKSCR3/CLKSMN/CLKSMX 0xF9: Enabling of writing to CLKSCR2/CLKSCR3/CLKSMN/CLKSMX Others: Reserved (Note 1)

(個別説明)

a. <CLKWEN>

0x06: CLKSCR2/ CLKSCR3/CLKSMN/CLKSMXレジスタ書き込み禁止

0xF9: CLKSCR2/ CLKSCR3/CLKSMN/CLKSMXレジスタ書き込み許可

注) CLKSCR1 への書き込みデータは"0x06"と"0xF9"のみ有効です。

"0xF9"以外のデータを CLKSCR1 にライトした場合、Disable になり、リード値は"0x06"に固定されます。

2. CLKSCR2 (周波数検知回路コントロールレジスタ 2)

Address = (0xF009_0000)+(0x0004)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:0]	CLKSEN	R/W	0x00	OFD circuits operation Enable Code: 0x00: Disabling 0xE4: Enabling Others: Invalid(Note 1)

(個別説明)

a. <OFDEN>

0x00: 周波数検知 回路停止

0xE4: 周波数検知回路スタート

注 1) CLKSCR2 への書き込みデータは "0x00"と"0xE4"のみ有効です。"0x00"と"0xE4"以外のデータをライトした場合は無効となります(レジスタ値は変化しません)。

注 2) CLKSCR1 に書き込み禁止コード"0x06"を書き込むと、CLKSCR2 への書き込みは禁止となり、CLKSCR2 に書いた全てのデータは無視されます。なお、書き込み禁止の場合でも CLKSCR2 の読み出しは可能です。

注 3) 本レジスタに WR してから検知機能が ON/OFF するまで最大で fs 2 サイクルかかります。したがって本レジスタに一度 WR したら fs 2 サイクル以上経過してから以降のライトをして下さい。検知機能の ON/OFF はレジスタ上に反映されないため、RD 値と動作状態が上記タイムラグ中は一致しません。

3. CLKSCR3 (周波数検知回路コントロールレジスタ 3)

Address = (0xF009_0000)+(0x0008)

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	—	—	Undefined	Read as undefined. Write as zero.
[1]	RESEN	R/W	0y0	OFD Reset Enable: 0y0 : Disable 0y1 : Enable
[0]	CLKSF	R/W	0y0	高周波発振器の周波数検出 Flag : Read 0y0 : OSC 正常 0y1 : OSC 異常 Write: 0y0 : 無効 0y1 : フラグクリア

(個別説明)

a. <RESEN>

0y0: OFD リセットは禁止になります。

0y1: OFD リセットは許可になります。

周波数異常を検出した場合、内部リセット動作を行ないます。

注) RESEN ビットは WR してからリセット出力が Enable/Disable するまで最大で fs 2 サイクルかかります。ポーリング可能なレジスタであるため、値が確定してから以降の WR を行って下さい。

b. <CLKSF>

Read の場合

0y0: 高速発振クロックの周波数は設定範囲内です。

0y1: 高速発振クロックの周波数は設定範囲外です。

<CLKSF>=0y1 の時、履歴を確認するためフラグクリアされるまで値は保持されます。
高速発振クロックの周波数が設定範囲に戻った場合でもフラグは<CLKSF>=0y1 を保持します。

Write の場合

0y0: 無効

0y1: Flag を”0”クリアします。

4. CLKSMN (検知周波数設定レジスタ (Low 側))

Address = (0xF009_0000)+(0x0010)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:0]	LDFS	R/W	0xA5	Lower detection frequency setting :

5. CLKSMX (検知周波数設定レジスタ (High 側))

Address = (0xF009_0000)+(0x0020)

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read as undefined. Write as zero.
[7:0]	HDFS	R/W	0xC9	High detection frequency setting :

(個別説明)

a. <LDFS>, <HDFS>

検知周波数設定ビットです。

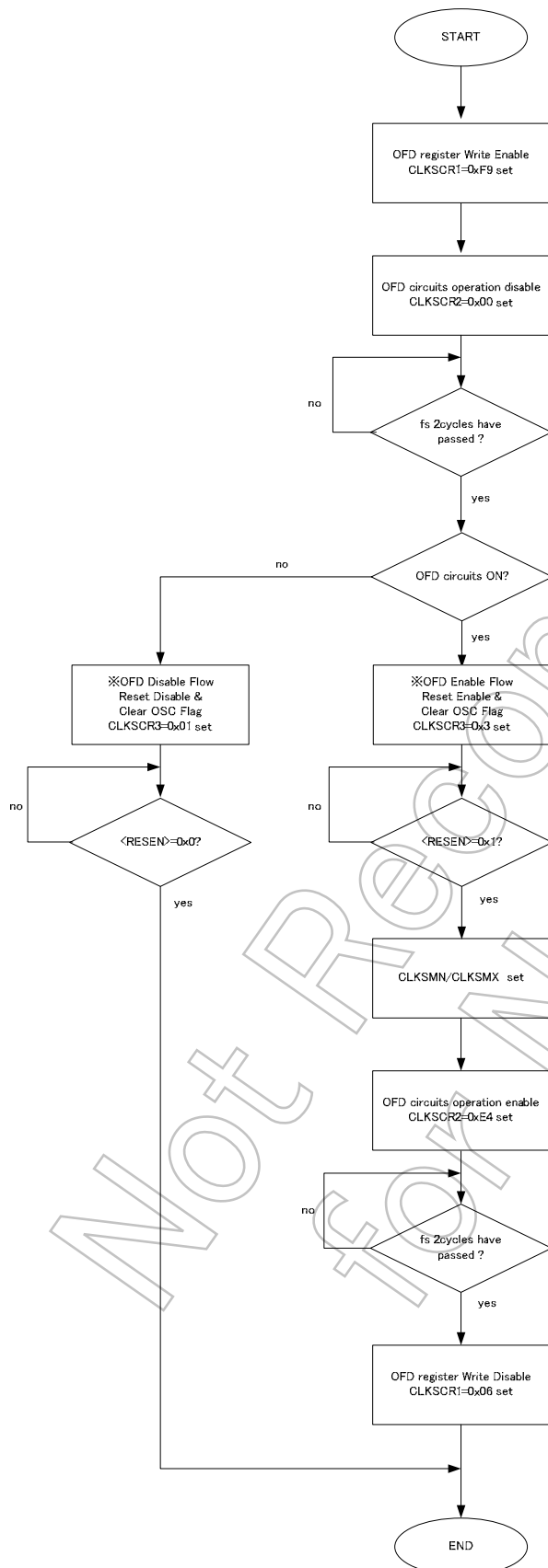
注 1) CLKSMN と CLKSMX レジスタは周波数検知動作が許可状態(CLKSCR2="0xE4") またはレジスタへの書き込み禁止状態(CLKSCR1="0x06")のときは書き換えできません。

注 2) CLKSMN と CLKSMX レジスタへの書き込みは CLKSCR1 への"0x06"書き込みでプロテクトされますが、読み出しについては CLKSCR1 の設定にかかわらず可能です。

注 3) CLKSMN と CLKSMX は CLKSMN < CLKSMX の条件下で、使用するクロック周波数に合わせて適切な値を設定してください。設定値の算出方法は、3.26.3 動作説明を参照してください。

注 4) CLKSMN および CLKSMX の設定は誤差を含め十分に余裕を持った設定を行なって下さい。CPU に対し内部リセットが掛かったままの状態になります。(デッドロック状態)

3.26.5 プログラム設定例



1) OFDのON/OFFプログラム設定例

< ON設定例 >

CLKSCR1=0xF9 ; OFDレジスタWrite Enable
↓
CLKSCR2=0x00 ; OFD circuits operation disable
↓
fs 2cycle 経過
↓
CLKSCR3=0x03 ; RESEN Enable & Clear OSC Flag
↓
RESEN=1 になるまでポーリング ; RESEN Bit 状態確認
↓
CLKSMN,CLKSMX レジスタの設定
↓
CLKSCR2=0xE4 ; OFD circuits operation enable
↓;
fs 2cycle 経過
↓
CLKSCR1=0x06 ; OFDレジスタWrite Disable

< OFF設定例 >

CLKSCR1=0xF9 ; OFDレジスタWrite Enable
↓
CLKSCR2=0x00 ; OFD circuits operation disable
↓
fs 2cycle 経過
↓
CLKSCR3=0x01 ; RESEN Disable
↓
RESEN=0 になるまでポーリング ; RESEN Bit 状態確認
↓
CLKSCR1=0x06 ; OFDレジスタWrite Disable

注) PCM モードでは OFD 回路は電源供給されておらず、動作しません。また、PCM を実行する前に OFD 機能を停止してください。

4. 電気的特性

4.1 絶対最大定格

記号	項目	定格	単位
DVCC3IO DVCCM	電源電圧	-0.3 ~ 3.8	V
DVCC1A DVCC1B DVCC1C		-0.3 ~ 2.0	
AVCC3AD AVDD3T AVDD3C AVCC3H		-0.3 ~ 3.8	
V_{IN}	入力電圧	-0.3 ~ DVCC3IO+0.3 (注 1) -0.3 ~ DVCCM+0.3 (注 2) -0.3 ~ AVCC3AD+0.3 (注 3) -0.3 ~ AVDD3T+0.3 (注 4) -0.3 ~ AVDD3C+0.3 (注 4) -0.3 ~ AVDD3H+0.3	V
IOL	出力電流(1 端子当り)	5	mA
IOH	出力電流(1 端子当り)	-5	mA
ΣI_{OL}	出力電流(合計)	80	mA
ΣI_{OH}	出力電流(合計)	-80	mA
P_D	消費電力($T_a=85^\circ\text{C}$)	800	mW
T_{SOLDER}	はんだ付け温度(10s)	260	$^\circ\text{C}$
T_{STG}	保存温度	-65~150	$^\circ\text{C}$
T_{OPR}	動作温度	-20~+85	$^\circ\text{C}$

注1) DVCC3IO (SM2-4, SM6, SM7, SN0-2, SP0-5, PA0-3, PB0-3, PC2-4, PC6, PC7, PN0, PN1, PT0-7, PU0-7, PV0-7) の絶対最大定格を超えないようにしてください。

注2) DVCCM (SA0-7, SB0-7, SE0-7, SF0-7, SG0-7, SH2, SH3, SH4, SH7, SJ0-6, SK0, SK1, SK4, SK5, SL0-2, SL4-SL6) の絶対最大定格を超えないようにしてください。

注3) PD4-7, VREFH, VREFL は AVCC3AD の絶対最大定格が適用されます。

注4) USB 端子、DDM、DDP は AVCC3T/3C の絶対最大定格が適用されます。

注5) 絶対最大定格とは瞬時たりとも超えてはならない規格であり、どの 1 つの項目も超える事ができない規格です。絶対最大定格を超えると、破壊や劣化の原因となり、破裂・燃焼による傷害を負うことがあります。従って、必ず絶対最大定格を超えないように、応用機器の設計を行ってください。

はんだ濡れ性についての注意事項

試験項目	試験条件	備考
はんだ付け性	230 $^\circ\text{C}$ 5 秒間 1 回 R タイプフラックス使用 (Sn-37Pb 鉛はんだ使用時) 245 $^\circ\text{C}$ 5 秒間 1 回 R タイプフラックス使用 (Sn-3.0Ag-0.5Cu はんだ使用時)	フォーミングまでの半田付着率 95%を良品とする

4.2 DC電気的特性

動作電圧

記号	項目	最小	標準	最大	単位	条件	
DVCC3IO	General I/O Power Supply Voltage (DVCC3IO) (DVSSCOMx=AVSS=0V)	3.0	3.3	3.6	V	X1=10 ~ 27MHz CPU CLK (~200MHz)	XT1=30 ~ 34kHz
DVCCM_1	Memory I/O Power	3.0	3.3	3.6			
DVCCM_2	Memory I/O Power	1.7	1.8	1.9			
AVCC3AD	ADC Power	3.0	3.3	3.6			
AVDD3T/3C	USB Device Power	3.15	3.3	3.45			
AVCC3H	USB Host Power	3.0	3.3	3.6			
DVCC1A	Internal Power A	1.4	1.5	1.6			
DVCC1B	Internal Power B						
DVCC1C	High CLK oscillator and PLL Power						

尚、各同一系の電源端子は、複数端子存在していますが、すべての同一系電源端子が、外部で電氣的に接続され、全て等しい電圧が供給されていることが前提となります。

注) I²S 機能は DVCC3IO から供給されます。

入力電圧 (1)

記号	項目	最小	標準	最大	単位	条件
VIL0	Input Low Voltage for SM2, SM6-7, SN0-2, SP0-3 PA0-3, PT0-7	-0.3	—	0.3×DVCC3IO	V	3.0≦DVCC3IO≦3.6V
VIL1	Input Low Voltage for SA0-7, SB0-7, SL4-6		—	0.3×DVCCM		3.0≦DVCCM≦3.6V SELDVCCM=1
VIL2			—	0.3×DVCCM		1.7≦DVCCM≦1.9V SELDVCCM=0
VIL5(注)	Input Low Voltage for PD4-7		—	0.3×AVCC3AD		3.0≦AVCC3AD≦3.6V
VIL6	Input Low Voltage for SM4, PC6, PC7, PN0, PN1		—	0.25×DVCC3IO		3.0≦DVCC3IO≦3.6V

注) PD4-7 ポートを汎用入力として使用した場合です

入力電圧 (2)

記号	項目	最小	標準	最大	単位	条件
VIH0	Input Low Voltage for SM2, SM6-7, SN0-2, SP0-3 PA0-3, PT0-7	$0.7 \times DVCC3IO$	—	$DVCC3IO + 0.3$	V	$3.0 \leq DVCC3IO \leq 3.6V$
VIH1	Input High Voltage for SA0-7, SB0-7, SL4-6	$0.7 \times DVCCM$	—	$DVCCM + 0.3$		$3.0 \leq DVCCM \leq 3.6V$
VIH2		$0.7 \times DVCCM$	—	$DVCCM + 0.3$		$1.7 \leq DVCCM \leq 1.9V$
VIH5(注)	Input High Voltage for PD4-7	$0.7 \times ADCC3AD$	—	$AVCC3AD + 0.3$		$3.0 \leq AVCC3AD \leq 3.6V$
VIH6	Input High Voltage for SM4, PC6, PC7, PN0, PN1	$0.75 \times DVCC3IO$	—	$DVCC3IO + 0.3$		$3.0 \leq DVCC3IO \leq 3.6V$

注) PD4-7 ポートを汎用入力として使用した場合です

出力電圧 (1)

記号	項目	最小	標準	最大	単位	条件
VOL0	Output Low Voltage for SM3, SP4, SP5, PB0-3, PC2-4, PC6, PC7, PN0, PN1, PT0-7	-	-	0.4	V	IOL=2.0 mA $3.0 \leq DVCC3IO \leq 3.6V$
VOL1	Output Low Voltage for SA0-7, SB0-7, SE0-7, SF0-7,					IOL=2.0 mA $3.0 \leq DVCCM \leq 3.6V$
VOL2	SG0-7, SH2-4, SH7, SJ0-6, SK0-1, SK4-5, SL0-2, SL4-5					IOL=2.0 mA $1.8 \leq DVCCM \leq 1.9V$
VOL5	Output Low Voltage for PD4-7					IOL=2.0 mA $3.0 \leq AVCC3AD \leq 3.6V$

出力電圧 (2)

記号	項目	最小	標準	最大	単位	条件
VOH0	Output Low Voltage for SM3, SP4, SP5, PB0-3, PC2-4, PC6, PC7, PN0, PN1, PT0-7	DVCC3IO-0.4	-	-	V	IOH=-1.0 mA $3.0 \leq DVCC3IO \leq 3.6V$
VOH1	Output Low Voltage for SA0-7, SB0-7, SE0-7, SF0-7,	DVCCM-0.4				IOH=-1.0 mA $3.0 \leq DVCCM \leq 3.6V$
VOH2	SG0-7, SH2-4, SH7, SJ0-6, SK0-1, SK4-5, SL0-2, SL4-5	DVCCM-0.4				IOH=-1.0 mA $1.8 \leq DVCCM \leq 1.9V$
VOH5	Output Low Voltage for PD4-7	AVCC3AD-0.4				IOH=-1.0 mA $3.0 \leq AVCC3AD \leq 3.6V$

その他

記号	項目	最小	標準	最大	単位	条件	
IMon	Internal resistor (ON) MX, MY pins	—	—	30	Ω	VOL =0.2V	3.0≦AVCC3AD≦3.6V
IMon	Internal resistor (ON) PX, PY pins	—	—	30		VOH = AVCC3AD−0.2V	
ILI	Input Leakage Current	—	0.02	±5	μA		
ILO	Output Leakage Current	—	0.05	±10	μA		
R	Pull Up/Down Resistor for RESETn, PA0-3, PD6	30	50	70	kΩ		
CIO	Pin Capacitance	—	1.0	—	pF	fc=1MHz	
VTH	Schmitt Width for SM4, PA0-3, PD6, PD7, PC6, PC7, PN0-1, PT4-6	—	0.6	—	V	3.0≦DVCC3IO≦3.6V	

注 1) Typ 値は特に指定のない限り Ta = 25°C、DVCC3IO = 3.3 V,
DVCCM=3.3V または DVCCM=1.8V、DVCC1A,1B,1C = 1.5V の値です。

注 2) 上記データはデバッグモード以外のデータを示します。

記号	項目	最小	標準	最大	単位	条件
ICC	NORMAL (注 2)	—	1.8	3.0	mA	DVCC3IO = 3.6V
		—	10	12		DVCC3LCD = 3.6V
		—	5	7		AVCC3AD = 3.6V
		—	50	—		AVDD3T = 3.3V AVDD3C = 3.3V
		—	0.7	2.0		AVCC3H = 3.3V
		—	28	36		上段は SDRAM アクセスのみ 下段は NORF アクセスのみ DVCCM = 3.6V
			13	17		
		—	15	19		上段は SDRAM アクセスのみ 下段は NORF アクセスのみ DVCCM = 1.9V
			7	9		
		—	136	206		DVCC1A = 1.6V DVCC1B = 1.6V DVCC1C = 1.6V
	CPU HALT	—	105	205	mA	PLL_ON $f_{FCLK} = 150\text{MHz}$ $T_a \leq 85^\circ\text{C}$ DVCC1A = 1.6V DVCC1B = 1.6V DVCC1C = 1.6V
		—	6.1	10		DVCC3IO = 3.6V DVCCM = 3.6V DVCC3LCD = 3.6V AVCC3AD = 3.6V AVDD3T = 3.6V AVDD3C = 3.6V AVCC3H = 3.6V
		—	16	116		DVCC1A = 1.6V DVCC1B = 1.6V DVCC1C = 1.6V
		—	—	—		PLL_OFF $f_{FCLK} = 25\text{MHz}$

動作状態: NORMAL

CPU は Drystone2.1 をベースに若干修正したプログラムを実行中

命令キャッシュ/データキャッシュ共に ON

プログラムメモリ: 内蔵 RAM データメモリ: 内蔵 RAM スタックメモリ: 内蔵 RAM

USB 初期状態

LCDC HVGA_16bpp 表示中

A/D 5us リピート変換中

I²S, CMS 動作停止

UART 480kbps 送信

SSP 100kbps 送信

PWM 100kHz 出力

SDRAM 100MHz CL=2 BL=8 32bit バス 読み書き (約 17% 外部バス起動)

または NORF 非同期 NORF 160ns アクセス 16bit バス 連続読み出し

動作状態: CPU HALT

CPU は HALT 状態、周辺回路は TSB オリジナルフローにて実行中

USB Suspend 状態

LCDC 簡易動作 (内蔵 RAM を VRAM として使用)

A/D、I²S、CMS、UART、SSP、PWM、外部メモリアクセス

動作停止

注 1) Typ 値は特に指定のない限り $T_a = 25^\circ\text{C}$ 、DVCC3IO=AVCC3H=3.3 V、

DVCCM=3.3V または DVCCM=1.8V、DVCC1A,1B,1C = 1.5V の値です。

注 2) ICC の測定条件: バス端子の CL=25pF、バス以外の出力端子は開放、入力端子はレベル固定

注 3) 上記データはデバッグモード以外のデータを示します。

記号	項目	最小	標準	最大	単位	条件	
ICC	Power Cut Mode (With PMC function)	—	4.5	100	μA	Ta ≤ 85°C	DVCC3IO = 3.6V DVCCM = 3.6V AVCC3AD = 3.6V AVDD3T = OPEN AVDD3C = OPEN AVCC3H = OPEN
				25		Ta ≤ 70°C	
				15		Ta ≤ 50°C	
			1.5	230		Ta ≤ 85°C	DVCC1A=0V DVCC1B=1.6V, DVCC1C=0V XT=32kHz X1,X2=OFF
				135		Ta ≤ 70°C	
				70		Ta ≤ 50°C	

注 1) Typ 値は特に指定のない限り Ta = 25°C、DVCC3IO=3.3 V、
DVCCM=3.3V または DVCCM=1.8V、DVCC1A,1B,1C = 1.5V の値です。

注 2) ICC の測定条件: バス端子の CL=50pF、バス以外の出力端子は開放、入力端子はレベル固定
(外部メモリに 8-wait アクセスで動作している状態)

注 3) 上記データはデバッグモード以外のデータを示します。

4.3 AC電気的特性

以下に示す、AC 規定は、特に指定の無い限り下記の条件下での測定結果となります。

AC 測定条件

- 表中の計算式に使われる" T "は内部バス周波数(f_{HCLK})の周期を示し、CPU のクロック f_{FCLK} の 1/2 の時間を示しています。
- 出力レベル: High = $0.7 \times DVCCM$, Low = $0.3 \times DVCCM$
- 入力レベル: High = $0.9 \times DVCCM$, Low = $0.1 \times DVCCM$

注) 表中の"計算式"は $DVCCM=1.7\sim 1.9V$ または $DVCCM=3.0\sim 3.6V$ 、 $DVCC1A=DVCC1B=DVCC1C=1.4\sim 1.6V$ の範囲での規定を示します。

4.3.1 基本バスサイクル

リードサイクル(非同期モード)

No.	項目	記号	計算式		100 MHz N=10 M=3 K=10 L=6	96 MHz N=10 M=3 K=10 L=6	48 MHz N=5 M=1 K=5 L=3	単位
			Min	Max				
1	内部バス周期(=T) 注1)	t_{CYC}	10	800	10.0	10.4	20.8	ns
2	A0 ~ A23 有効 → D0 ~ D15 入力	t_{AD}	(N)T - 15.0		85.0	89.2	89.2	
3	SMCOEn 立ち下がり → D0 ~ D15 入力	t_{OED}	(N-M)T - 10.0		60.0	62.8	73.3	
4	SMCOEn 低レベルパルス幅	$t_{OE\overline{W}}$	(N-M)T - 8.0		62.0	64.9	75.3	
5	A0 ~ A23 有効 → SMCOEn 立ち下がり	t_{AOE}	MT - 5.0		25	26.3	15.8	
6	SMCOEn 立ち上がり → D0 ~ D15 保持	t_{HR}	0		0	0	0	
7	SMCOEn 高レベルパルス幅	t_{OEHW}	MT - 8.0		22	23.3	12.8	

ライトサイクル(非同期モード)

8	D0 ~ D15 有効 → SMCWEn 立ち上がり	t_{DW}	(L+1)T - 10.0		60.0	62.9	73.3	ns
9	D0 ~ D15 有効 → SMCBEn 立ち上がり(bls=1)	t_{SDS}	(L+1)T - 10.0		60.0	62.9	73.3	
10	SMCWEn 低レベルパルス幅	t_{WW}	LT - 8.0		52.0	54.5	54	
11	A0 ~ A23 有効 → SMCWEn 立ち下がり	t_{AW}	T - 5.0		5.0	5.4	15.8	
12	SMCWEn 立ち上がり → A0 ~ A23 保持	t_{WA}	(K-L-1)T - 5.0		25.0	26.3	15.8	
13	SMCWEn 立ち上がり → D0 ~ D15 保持	t_{WD}	(K-L-1)T - 5.0		25.0	26.3	15.8	
14	SMCOEn 立ち上がり → D0 ~ D15 出力	t_{OEO}	2		2.0	2.0	2.0	
15	データバイト制御 ~ ライト終了時間	t_{SBW}	LT - 8.0		52.0	54.5	54.5	

- 表中の計算式に使われる変数は以下で定義されます。
 $N = t_{RC}$ サイクル数 $M = t_{CEOE}$ サイクル数
 $K = t_{WC}$ サイクル数 $L = t_{WP}$ サイクル数

測定条件

接続設定

1. $DVCC3IO \times 0.7 \leq SELDVCCM \leq DVCC3IO$

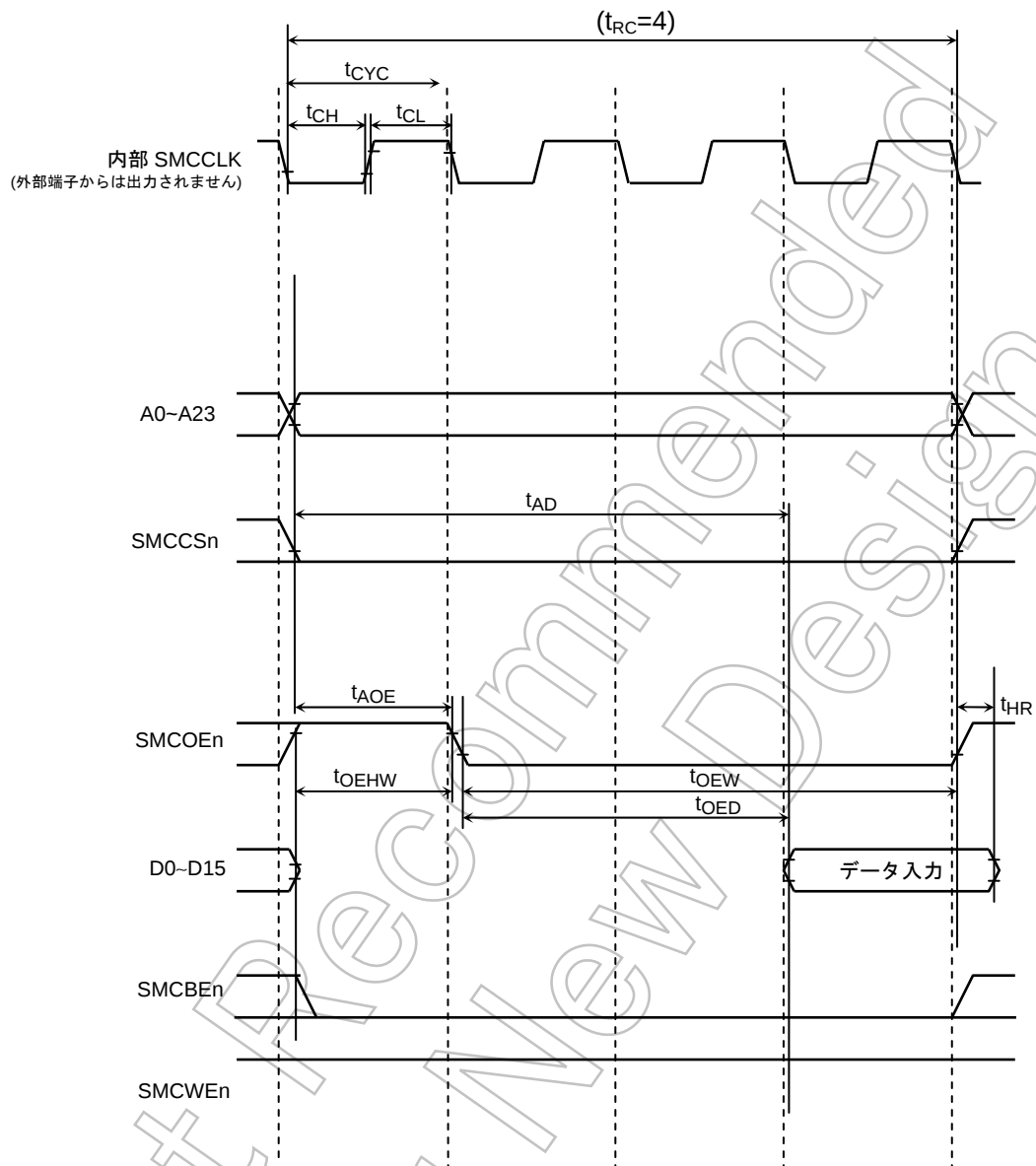
ソフトウェア設定

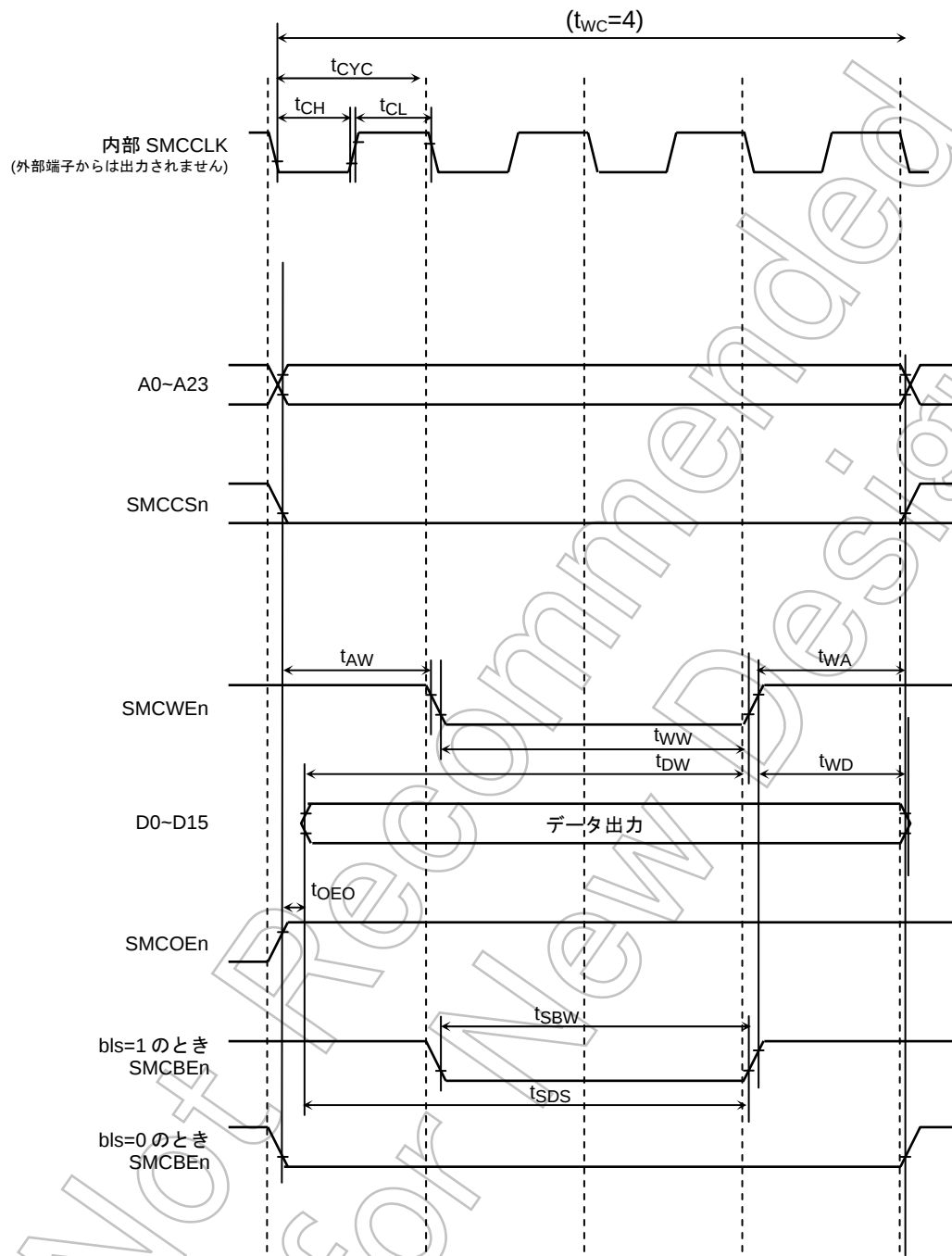
1. $PMCDRV < DRV_MEM1:0 > = 0y11$ (Full 設定 @ $1.8 \pm 0.1V$)
2. $PMCDRV < DRV_MEM1:0 > = 0y01$ (Half 設定 @ $3.3 \pm 0.3V$)

負荷測定条件

負荷容量 CL = 25 pF

注 1) 上記表の計算で保証温度は $0\sim 70^{\circ}C$ 、内部バス周期の Min 値 T = 10ns です。保証温度は $-20\sim 80^{\circ}C$ 、内部バス周期の保証 Min 値 T = 13.3ns です。

(1) 非同期式メモリ・リードサイクル ($t_{RC}=4, t_{CEOE}=1$ 設定例)

(2) 非同期式メモリ・ライトサイクル ($t_{WC}=4$, $t_{WP}=2$ 設定例)

4.3.2 DDR SDRAMコントローラAC電気的特性

AC 測定条件

- 表中の計算式に使われる" T "は内部バス周波数(f_{HCLK})の周期を示し、CPU のクロック f_{FCLK} の 1/2 の時間を示しています。
- 出力レベル: High = $0.7 \times DVCCM$, Low = $0.3 \times DVCCM$
- 入力レベル: High = $0.9 \times DVCCM$, Low = $0.1 \times DVCCM$
- クロック出力 Differential レベル(VOD): $VOD = 0.6 \times DVCCM$
- クロック出力 Differential Crosspoint レベル(VOX): High = $0.6 \times DVCCM$, Low = $0.4 \times DVCCM$

注) LVCMOS タイプの DDR_SDRAM のみ対応です。SSTIL(2.5V)タイプの DDR_SDRAM には対応出来ません。

注) 表中の "計算式" は $DVCCM=1.7\sim1.9V$ 、 $DVCC1A=DVCC1B=DVCC1C=1.4\sim1.6V$ の範囲での規定を示します。

記号	項目	最小	標準	最大	単位	条件
VIX	AC Differential Cross point Voltage	$0.4 \times DVCCM$		$0.6 \times DVCCM$		$1.7 \leq DVCCM \leq 1.9V$

No.	項目	記号	計算式		100 MHz	96 MHz	単位
			Min	Max			
1	DMCDCLKP/ DMCDCLKN cycle time 注 1)	t_{CK}	T		10	10.4	ns
2	DMCDCLKP&DMCDCLKN Clock Skew time		-0.35	0.35			
3	CLK Differential Crosspoint cycle	t_{CH}	$0.5T - 0.5$		4.5	4.7	
4	CLK Differential Crosspoint cycle	t_{CL}	$0.5T - 0.5$		4.5	4.7	
5	DMCDDQSx Access time from CLK($CL^*=3$)	t_{AC1}		2T-13.5	6.5	7.3	
6	Data Access time from CLK($CL^*=3$)	t_{AC2}		2T-13.5	6.5	7.3	
7	DQS to Data Skew time	t_{DQSQ}	0	0.7	0.7	0.7	
8	Address set-up time	t_{AS}	$0.5T - 3.0$		2.0	2.2	
9	Address hold time	t_{AH}	$0.5T - 3.0$		2.0	2.2	
10	CKE set-up time	t_{CKS}	$0.5T - 3.0$		2.0	2.2	
11	Command set-up time	t_{CMS}	$0.5T - 3.0$		2.0	2.2	
12	Command hold time	t_{CMH}	$0.5T - 3.0$		2.0	2.2	
13	Data Setup Time	t_{PS}	$0.25T - 1.5$		1.0	1.1	
14	Data Hold Time	t_{PH}	$0.25T - 1.5$		1.0	1.1	
15	DMCDDM Setup Time	t_{MS}	$0.25T - 1.5$		1.0	1.1	
16	DMCDDM Hold Time	t_{MH}	$0.25T - 1.5$		1.0	1.1	
17	Write command to 1'st DQS Latching Transition	t_{POSS}	$0.75T$	$1.25T$	7.50~12.5	7.80~13.0	

*CL は CAS レイテンシを示していますが、DDR_SDRAM の場合、CAS レイテンシの数え方が SDR_SDRAM と異なります(メモリコントローラには、DDR_SDRAM の CL より"1"小さい値を設定する)。本仕様書ではメモリコントローラへの設定値をベースに記述されていますので、御注意ください。

注 1) 上記表の計算で保証温度は $0\sim70^{\circ}C$ 、内部バス周期の Min 値 $T = 10ns$ です。保証温度は $-20\sim80^{\circ}C$ 、内部バス周期の保証 Min 値 $T = 13.3ns$ です。

測定条件

接続設定

1. $DVCC3IO \times 0.7 \leq SELDVCCM \leq DVCC3IO$
2. $DVCC3IO \times 0.7 \leq SELMEMC \leq DVCC3IO$
3. DMCCLKIN 端子に F/B 用のクロック DMCDCLKP 信号をマイコン外部で接続してください。

ソフトウェア設定

1. $PMCDRV < DRV_MEM1:0 > = 0y11$ (Full 設定)
2. $dmc_user_config_5 = 0x0000_0058$

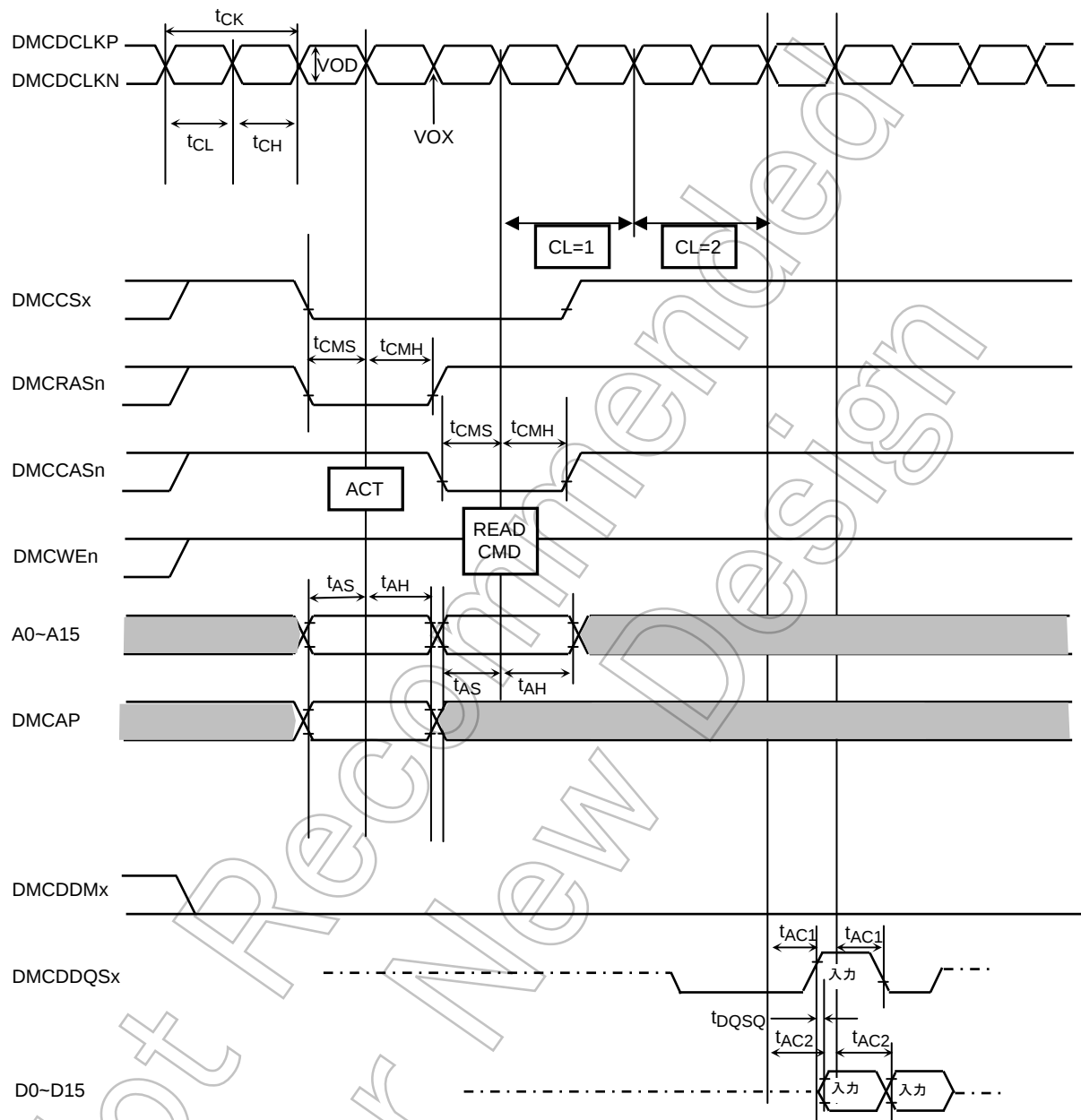
負荷測定条件

DMCDCLKP 端子、DMCDCLKN 端子の付加容量 $CL = 15pF$ 、その他の端子の付加容量 $CL = 25pF$

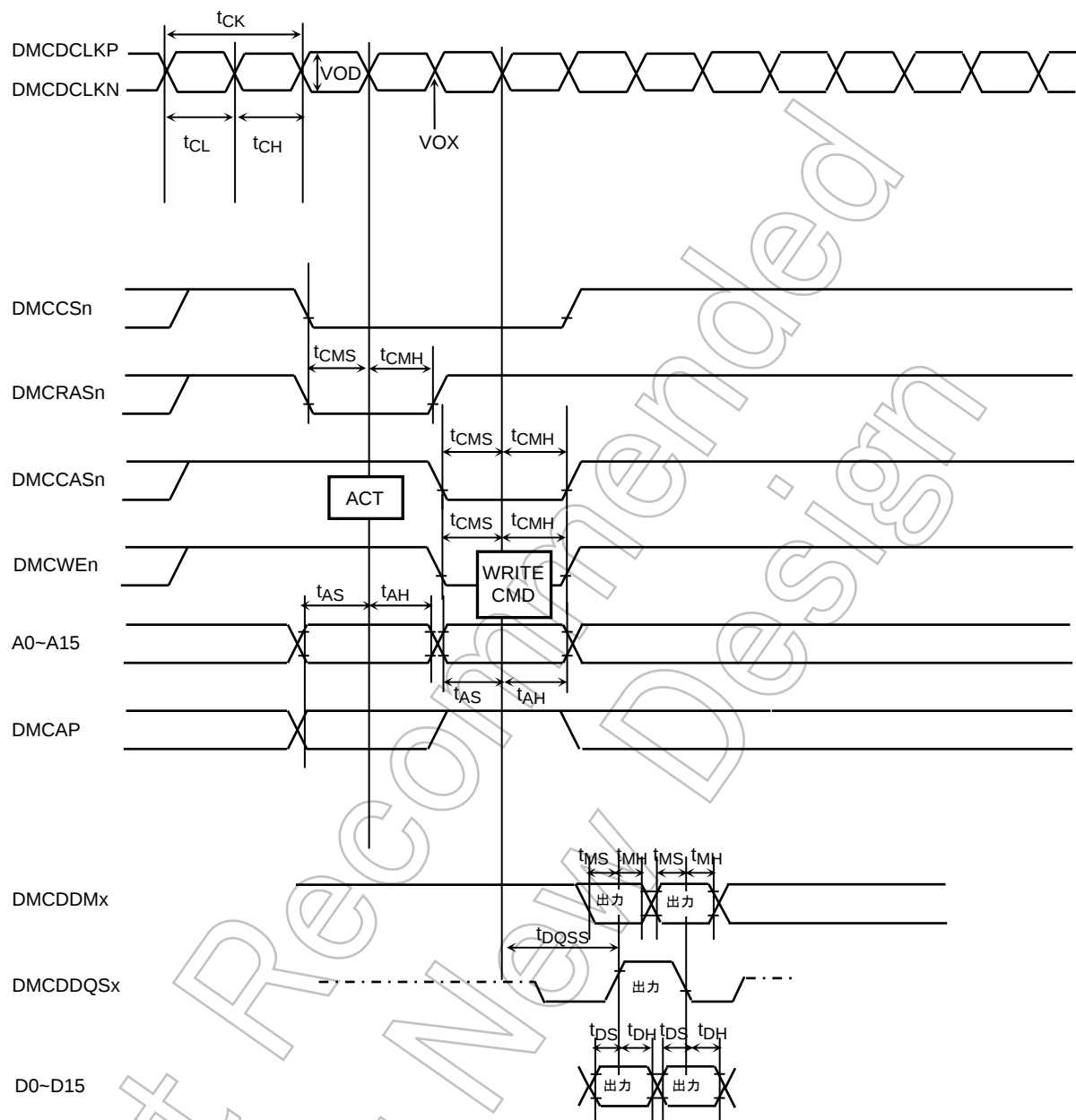
(1) DDR_SDRAM リードタイミング

(2Word 長リードモード: Memory の CAS レイテンシ=3

Memory Controller の CAS レイテンシ=2)



(2) DDR_SDRAM ライトタイミング (2Word ライトモード)



4.3.3 Mobile SDR SDRAMコントローラAC電气的特性

AC 測定条件

- 表中の計算式に使われる" T "は内部バス周波数(f_{HCLK})の周期を示し、CPUのクロック f_{FCLK} の1/2の時間を示しています。
- 出力レベル: High = $0.7 \times DVCCM$, Low = $0.3 \times DVCCM$
- 入力レベル: High = $0.9 \times DVCCM$, Low = $0.1 \times DVCCM$

注) 表中の“計算式”は $DVCCM=1.7\sim1.9V$ 、 $DVCC1A=DVCC1B=DVCC1C=1.4\sim1.6V$ の範囲での規定を示します。

No.	項目	記号	計算式		100 MHz	96MHz	単位
			Min	Max			
1	CLK cycle time 注 1)	t_{CK}	T		10	10.4	ns
2	DMCCLK high level width	t_{CH}	$0.5T - 1.5$		3.5	3.7	
3	DMCCLK low level width	t_{CL}	$0.5T - 1.5$		3.5	3.7	
4	Access time from CLK($CL^*=2$)	t_{AC}		$T - 4.0$	6.0	6.4	
5	Data hold time from internal read	t_{HR}	2.0		2.0	2.0	
6	Data set-up time	t_{DS}	$0.5T - 3.0$		2.0	2.2	
7	Data hold time	t_{DH}	$0.5T - 4.0$		1.0	1.2	
8	Address set-up time	t_{AS}	$0.5T - 3.0$		2.0	2.2	
9	Address hold time	t_{AH}	$0.5T - 4.0$		1.0	1.2	
10	CKE set-up time	t_{CKS}	$0.5T - 3.0$		2.0	2.2	
11	Command set-up time	t_{CMS}	$0.5T - 3.0$		2.0	2.2	
12	Command hold time	t_{CMH}	$0.5T - 4.0$		1.0	1.2	

*CL は CAS レイテンシを示しています。

測定条件

接続設定

1. $DVSSCOMx \leq SELDVCCM \leq DVCC3IO \times 0.3$
2. $DVSSCOMx \leq SELMEMC \leq DVCC3IO \times 0.3$
3. DMCCLKIN 端子に F/B 用のクロック接続は不要です。DVSSCOMx へ接続するなど、貫通電流対策用にレベル固定してください。

ソフトウェア設定

1. $PMCDRV < DRV_MEM1:0 > = 11$ (Full 設定)
2. $dmc_user_config_3 = 0x0000_0000$ (16bit バスメモリ)

負荷測定条件

DMCCLK 端子の付加容量 CL = 15pF、その他の端子の付加容量 CL = 25 pF

注 1) 上記表の計算で保証温度は $0\sim70^{\circ}C$ 、内部バス周期の Min 値 $T = 10ns$ です。保証温度は $-20\sim80^{\circ}C$ 、内部バス周期の保証 Min 値 $T = 13.3ns$ です。

4.3.4 SDR SDRAMコントローラAC電气的特性

AC 測定条件

- 表中の計算式に使われる" T "は内部バス周波数(f_{HCLK})の周期を示し、CPU のクロック f_{FCLK} の 1/2 の時間を示しています。
- 出力レベル: High = $0.7 \times DVCCM$, Low = $0.3 \times DVCCM$
- 入力レベル: High = $0.9 \times DVCCM$, Low = $0.1 \times DVCCM$

注) 表中の "計算式" は $DVCCM=3.0\sim 3.6V$ 、 $DVCC1A=DVCC1B=DVCC1C=1.4\sim 1.6V$ の範囲での規定を示します。

No.	項目	記号	計算式		100 MHz	96MHz	単位
			Min	Max			
1	CLK cycle time 注 1)	t_{CK}	T		10	10.4	ns
2	DMCCLK high level width	t_{CH}	$0.5T - 1.5$		3.5	3.7	
3	DMCCLK low level width	t_{CL}	$0.5T - 1.5$		3.5	3.7	
4	Access time from CLK($CL^*=2$)	t_{AC}		$T - 4.0$	6.0	6.4	
5	Data hold time from internal read	t_{HR}	2.0		2.0	2.0	
6	Data set-up time	t_{DS}	$0.5T - 3$		2.0	2.2	
7	Data hold time	t_{DH}	$0.5T - 4.0$		1.0	1.2	
8	Address set-up time	t_{AS}	$0.5T - 3.0$		2.0	2.2	
9	Address hold time	t_{AH}	$0.5T - 4.0$		1.0	1.2	
10	CKE set-up time	t_{CKS}	$0.5T - 3.0$		2.0	2.2	
11	Command set-up time	t_{CMS}	$0.5T - 3.0$		2.0	2.2	
12	Command hold time	t_{CMH}	$0.5T - 4.0$		1.0	1.2	

*CL は CAS レイテンシを示しています。

測定条件

接続設定

1. $DVCC3IO \times 0.7 \leq SELDVCCM \leq DVCC3IO$
2. $DVSSCOMx \leq SELMEMC \leq DVCC3IO \times 0.3$
3. DMCCLKIN 端子に F/B 用のクロック接続は不要です。DVSSCOMx へ接続するなど、貫通電流対策用にレベル固定してください。

ソフトウェア設定

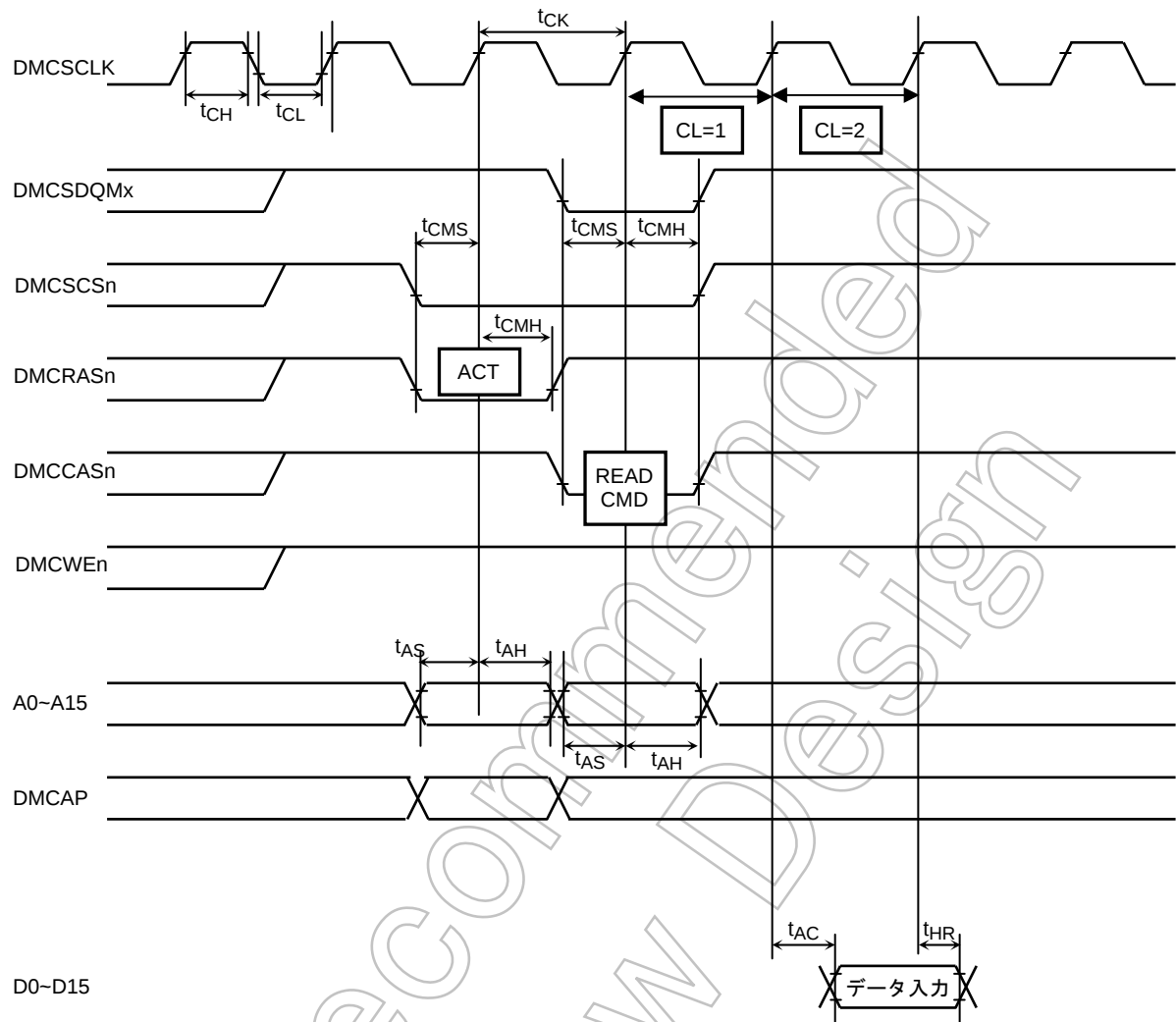
1. $PMCDRV < DRV_MEM1:0 > = 11$ (Full 設定)
2. $dmc_user_config_3 = 0x0000_0000$ (16bit バスメモリ)

負荷測定条件

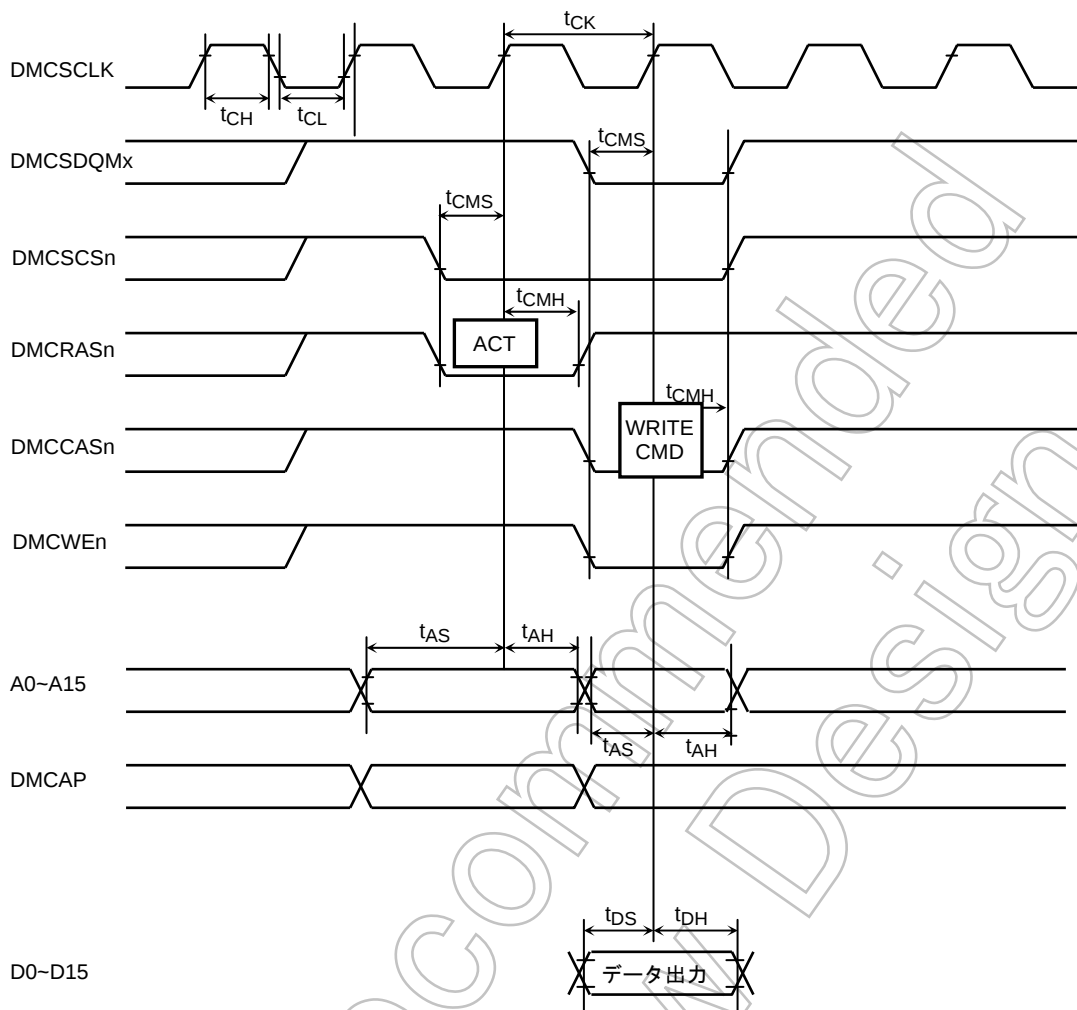
DMCCLK 端子の付加容量 CL = 15pF、その他の端子の付加容量 CL = 25 pF

注 1) 上記表の計算で保証温度は $0\sim 70^{\circ}C$ 、内部バス周期の Min 値 $T = 10ns$ です。保証温度は $-20\sim 80^{\circ}C$ 、内部バス周期の保証 Min 値 $T = 13.3ns$ です。

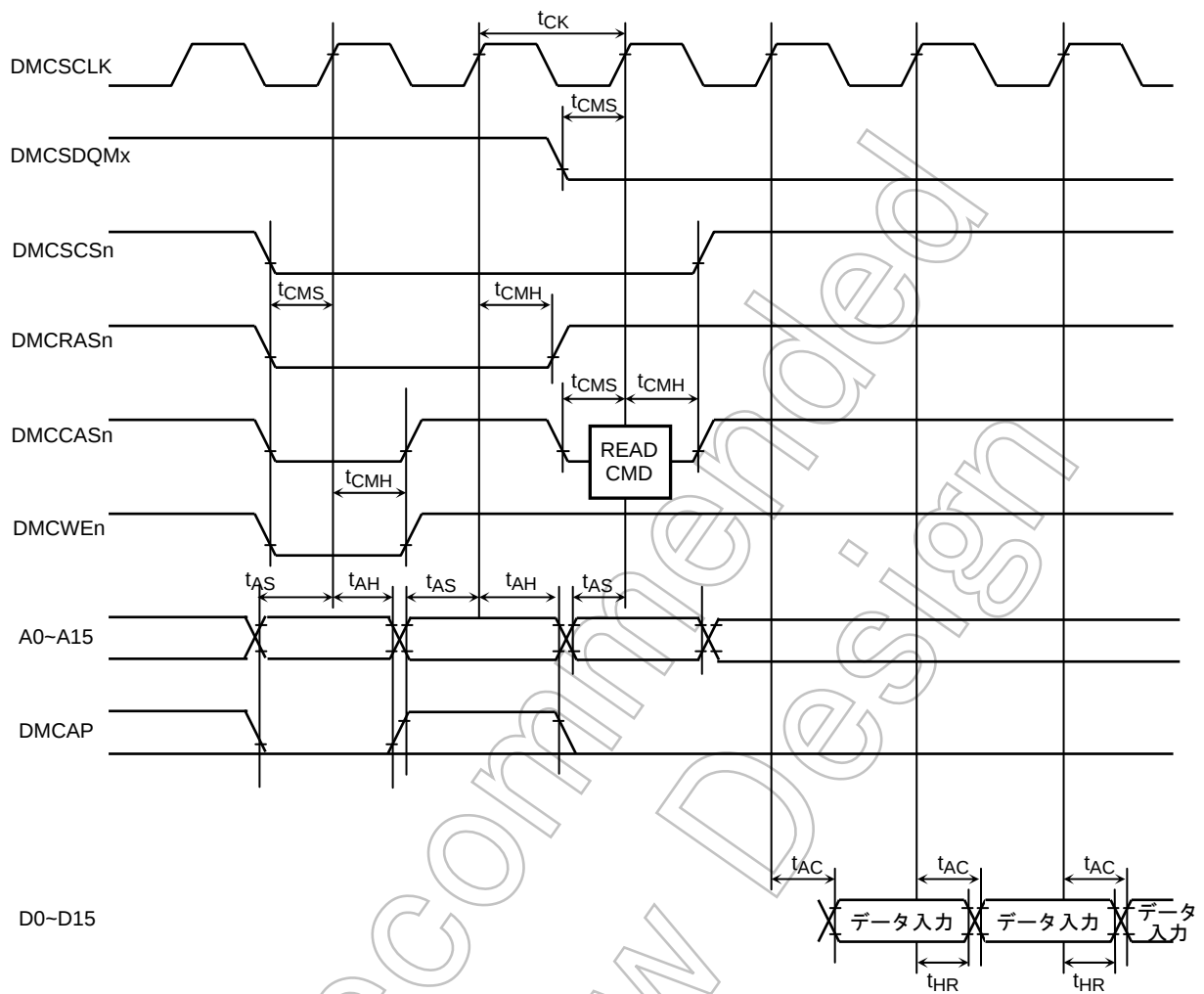
(1) SDRAM リードタイミング (CAS レイテンシ=2)



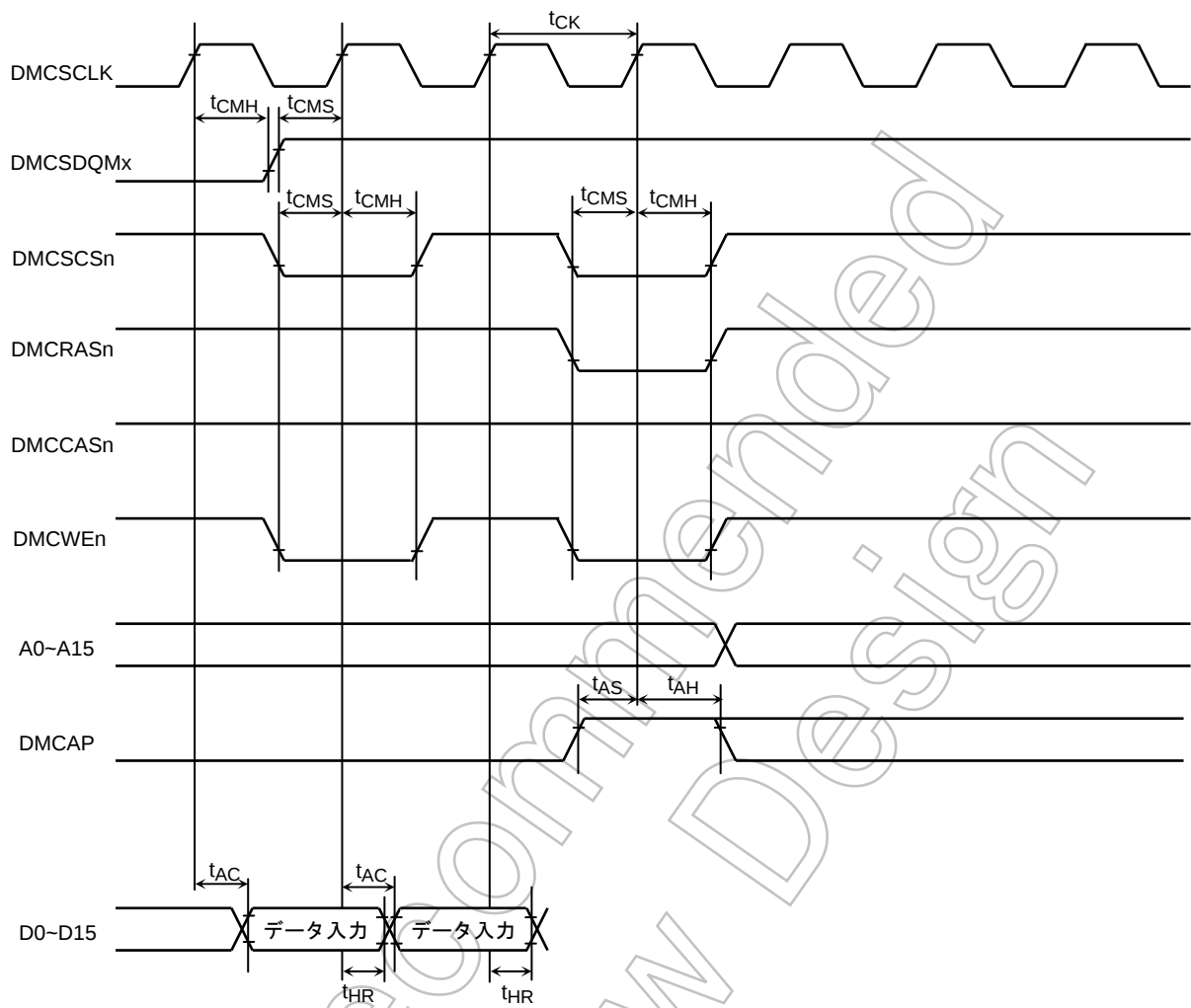
(2) SDRAM ライトタイミング



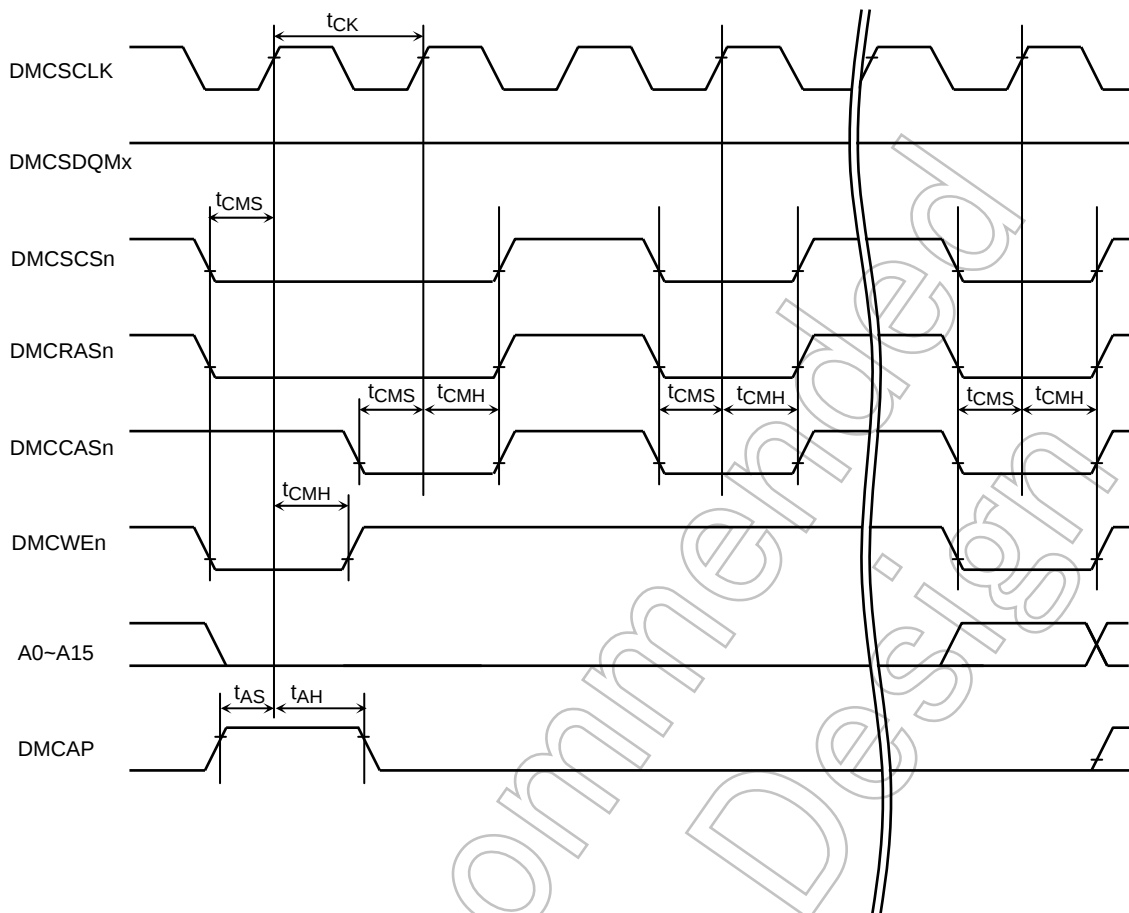
(3) SDRAM バーストリードタイミング (バーストサイクル開始、CAS レイテンシ=2)



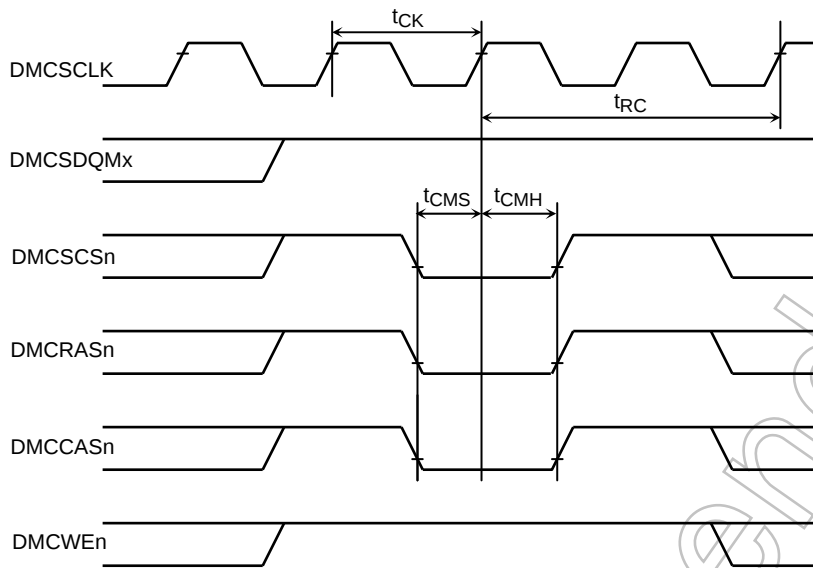
(4) SDRAM バーストリードタイミング (バーストタイミング終了)



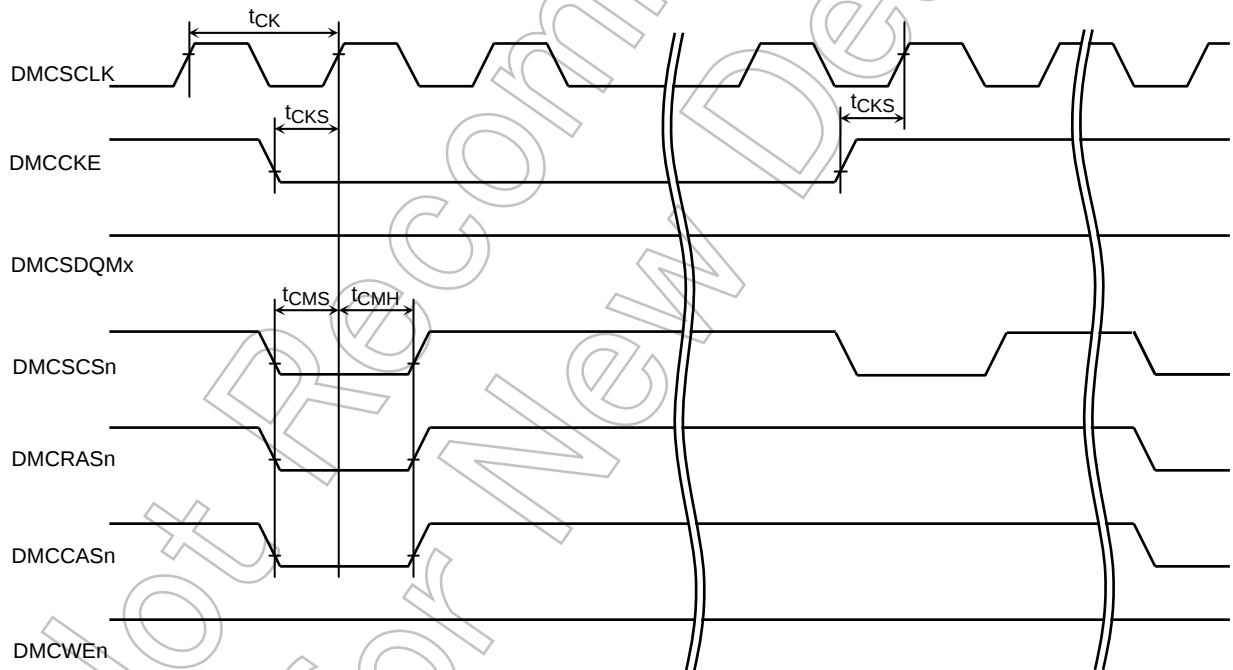
(5) SDRAM 初期化タイミング



(6) SDRAM リフレッシュタイミング



(7) SDRAM セルフリフレッシュタイミング



4.3.5 NANDフラッシュコントローラAC電气的特性

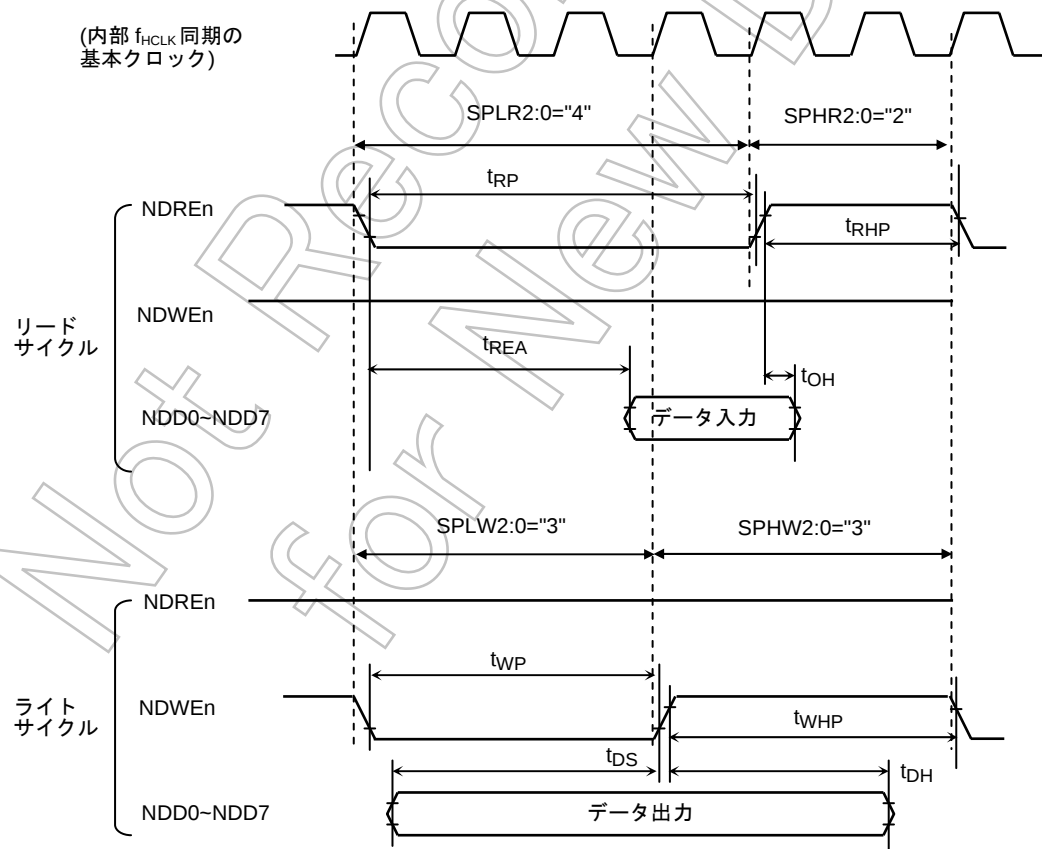
AC 測定条件

- 表中の計算式に使われる" T "は内部バス周波数(f_{HCLK})の周期を示し、CPUのクロック f_{FCLK} の1/2の時間を示しています。
- 出力レベル: High = $0.7 \times DVCC3IO$, Low = $0.3 \times DVCC3IO$
- 入力レベル: High = $0.9 \times DVCC3IO$, Low = $0.1 \times DVCC3IO$

注) 表中の“計算式”は $DVCC3IO=3.0\sim3.6V$ 、 $DVCC1A=DVCC1B=DVCC1C=1.4\sim1.6V$ の範囲での規定を示します。

注) 計算式中の n は NDFMCR2<SPLR2:0>に設定された値を示し、m は NDFMCR2<SPHR2:0>に設定された値を示し、k は NDFMCR2<SPLW2:0>に設定された値を示し、l は NDFMCR2<SPHW2:0>に設定された値を示します。計算式中で、結果がマイナスになる設定は使用できませんので注意してください。

No.	記号	項目	計算式		100 MHz (n=3) (m=3) (k=3) (l=3)	96 MHz (n=3) (m=3) (k=3) (l=3)	単位
			Min	Max			
1-1	t_{RC}	リードアクセスサイクル	$(n+m)T$		60.0	62.5	ns
1-2	t_{WC}	ライトアクセスサイクル	$(k+l)T$		60.0	62.5	
2	t_{RP}	NDREn 低レベルパルス幅	$(n)T - 10.0$		20.0	21.2	
3	t_{RHP}	NDREn 高レベルパルス幅	$(m)T - 10.0$		20.0	21.2	
4	t_{REA}	NDREn データアクセス時間		$(n)T - 14$	16.0	17.2	
5	t_{OH}	リードデータ保持時間	0		0	0	
6	t_{WP}	NDWEn 低レベルパルス幅	$(k)T - 10.0$		20.0	21.2	
7	t_{WHP}	NDWEn 高レベルパルス幅	$(l)T - 10.0$		20.0	21.2	
8	t_{DS}	ライトデータセットアップ時間	$(k)T - 10.0$		20.0	21.2	
9	t_{DH}	ライトデータ保持時間	$(l)T - 10.0$		20.0	21.2	



AC 測定条件

- CL = 40 pF

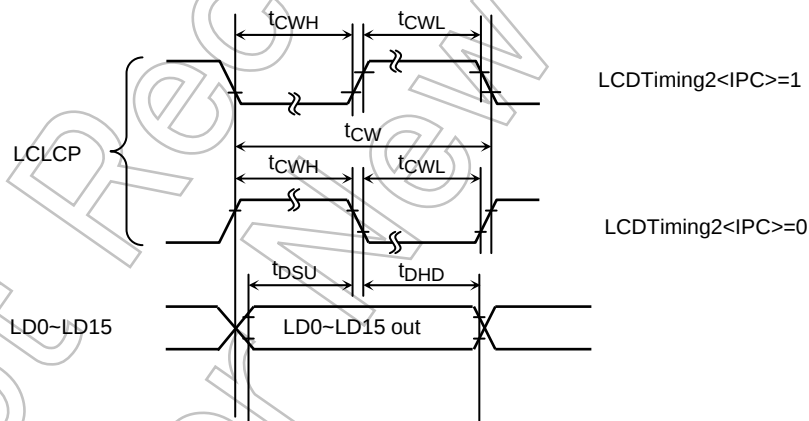
4.3.6 LCDコントローラ

AC 測定条件

- 表中の計算式に使われる" T "は内部バス周波数(f_{HCLK})の周期を示し、CPU のクロック f_{FCLK} の 1/2 の時間を示しています。
- 出力レベル: High = $0.7 \times DVCC3LCD$, Low = $0.3 \times DVCC3IO$

注) 表中の“計算式”は $DVCC3LCD=1.8\sim 3.6V$ 、 $DVCC1A=DVCC1B=DVCC1C=1.4\sim 1.6V$ の範囲での規定を示します。

項目	記号	計算式		100 MHz (n=3)	96 MHz (n=3)	48 MHz (n=2)	単位
		Min	Max				
LCLCP クロック周期 (nT)	t_{CW}	30		30.0	31.25	41.6	ns
LCLCP 高レベルパルス幅 (位相反転時も含む)	t_{CWH}	n が偶数のとき $(nT/2) - 5.0$ n が奇数のとき $((n-1)T/2) - 5.0$		5.0	5.4	15.8	
LCLCP 低レベルパルス幅 (位相反転時も含む)	t_{CWL}	n が偶数のとき $(nT/2) - 5.0$ n が奇数のとき $((n+1)T/2) - 5.0$		15.0	15.8	15.8	
データ有効 → LCLCP 立ち下がり (位相反転時も含む)	t_{DSU}	n が偶数のとき $(nT/2) - 6.0$ n が奇数のとき $((n-1)T/2) - 6.0$		4.0	4.4	14.8	
LCLCP 立ち下がり → データ保持 (位相反転時も含む)	t_{DHD}	n が偶数のとき $(nT/2) - 6.0$ n が奇数のとき $((n+1)T/2) - 6.0$		14.0	14.8	14.8	



AC 測定条件

- CL = 20 pF

注) 計算式中の n は LCDTiming2<PCD_HI>および<PCD_LO>に設定された値+2 を示し、n の値は動作周波数によって制限されます。

例 1) $f_{HCLK} = 100\text{MHz}$ のとき、LCDTiming2<PCD_HI>=0y00000、<PCD_LO>=0y00001、(n=3)

例 2) $f_{HCLK} = 48\text{MHz}$ のとき、LCDTiming2<PCD_HI>=0y00000、<PCD_LO>=0y00000、(n=2)

4.3.7 SSPコントローラ

AC 測定条件

表中の計算式に使われる" T "は内部バス PCLK 周期を示し、CPU のクロック f_{CLK} 周期の 1/2 を示しています。保証温度は 0~70°C、内部バス周期の Min 値 $T = 10\text{ns}$ です。

保証温度は-20~80°C、内部バス周期の保証 Min 値 $T = 13.3\text{ns}$ です。

- 出力レベル: $\text{High} = 0.7 \times \text{DVCC3IO}$, $\text{Low} = 0.3 \times \text{DVCC3IO}$
- 入力レベル: $\text{High} = 0.9 \times \text{DVCC3IO}$, $\text{Low} = 0.1 \times \text{DVCC3IO}$

注) 表中の "計算式" は $\text{DVCC3IO} = 3.0 \sim 3.6\text{V}$ 、 $\text{DVCC1A} = \text{DVCC1B} = \text{DVCC1C} = 1.4 \sim 1.6\text{V}$ の範囲での規定を示します。

項目	記号	計算式		PCLK 100MHz (m=6 n=12)	PCLK 96MHz (m=6 n=12)	単位
		Min	Max			
SPxCLK 周期 (マスタ)	T_m	$(m)T$ ただし、50ns 以上		60.0	62.5	
SPxCLK 周期 (スレーブ)	T_s	$(n)T$		120.0	125.0	
SPxCLK 立ち上がり時間	t_r		10.0	10.0	10.0	
SPxCLK 立ち下がり時間	t_f		10.0	10.0	10.0	
マスタモード時 SPxCLK 低レベルパルス幅	t_{WLM}	$(m)T / 2 - 7.0$		23.0	24.3	
マスタモード時 SPxCLK 高レベルパルス幅	t_{WHM}	$(m)T / 2 - 7.0$		23.0	24.3	
スレーブモード時 SPxCLK 低レベルパルス幅	t_{WLS}	$(n)T / 2 - 7.0$		53.0	55.5	
スレーブモード時 SPxCLK 高レベルパルス幅	t_{WHS}	$(n)T / 2 - 7.0$		53.0	55.5	
マスタモード時 SPxCLK 立ち上がり/立ち下がり→出力データ有効	t_{ODSM}		15.0	15.0	15.0	
マスタモード時 SPxCLK 立ち上がり/立ち下がり→出力データ保持	t_{ODHM}	$(m)T / 2 - 10$		20.0	21.3	
マスタモード時 SPxCLK 立ち上がり/立ち下がり →入力データ有効 遅延時間	t_{IDSM}		$(m)T / 2 - 20$	10.0	11.2	
マスタモード時 SPxCLK 立ち上がり/立ち下がり→入力データ保持	t_{IDHM}	5.0		5.0	5.0	
マスタモード時 SPxFS 有効→SPxCLK 立ち上がり/立ち下がり	t_{OFSM}	$(m)T - 10$	$(m)T + 10$	50~70	52.5~72.5	
スレーブモード時 SPxCLK 立ち上がり/立ち下がり →出力データ有効遅延時間	t_{ODSS}		$(3T) + 25$	55.0	56.3	
スレーブモード時 SPxCLK 立ち上がり/立ち下がり→出力データ保持	t_{ODHS} 注1)	$(n)T / 2 + (2T)$		80.0	83.3	
スレーブモード時 SPxCLK 立ち上がり/立ち下がり →入力データ有効 遅延時間	t_{IDSS}		$(n)T / 2 + (3T) - 10.0$	80.0	83.8	
スレーブモード時 SPxCLK 立ち上がり/立ち下がり→入力データ保持	t_{IDHS}	$(3T) + 10$		40.0	41.3	
スレーブモード時 SPxFS 有効→SPxCLK 立ち上がり/立ち下がり	t_{OFSS}	$(n)T$		120	125	

注 1) 通信ポーレートクロックは以下の条件範囲で設定する必要があります。

マスタモード時

$$m = (\text{CPSDVSR} \times (1 + \text{SCR})) = f_{PCLK} / \text{SPxCLK}$$

<CPSDVSR>は偶数のみが設定可能です。また m の範囲は $65204 \geq m \geq 2$ となります。

スレーブモード時

$$n = f_{PCLK} / \text{SPxCLK} \quad (65204 \geq n \geq 12)$$

AC 測定条件

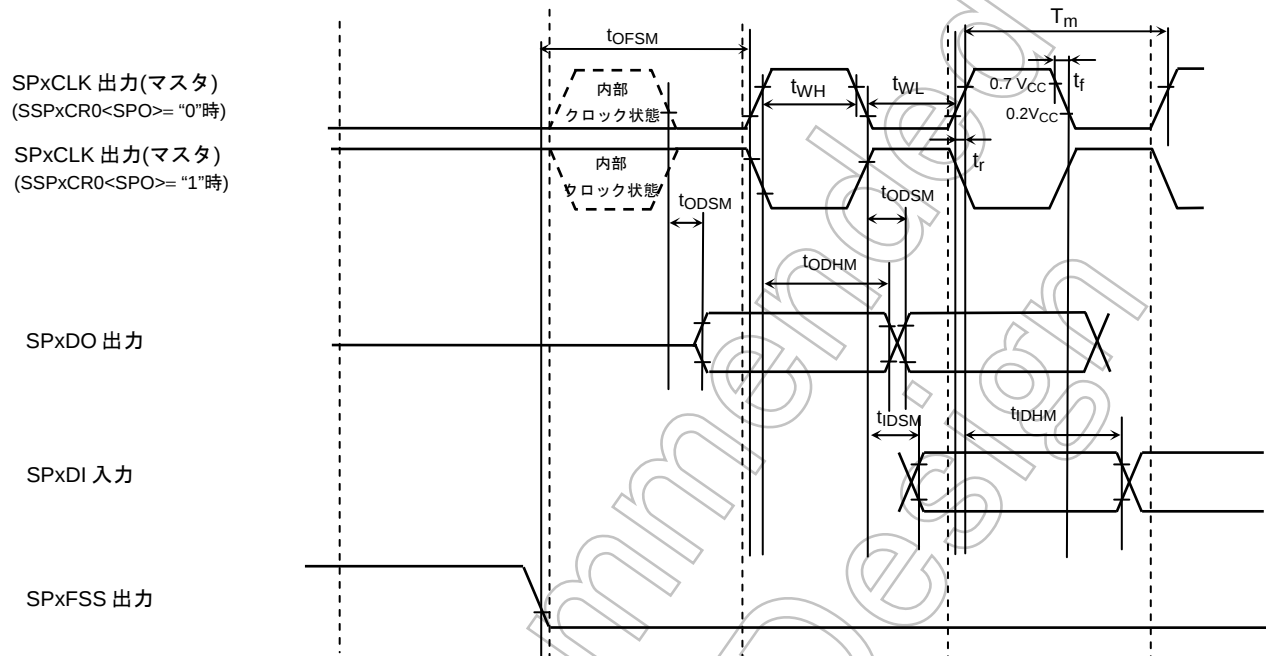
負荷容量 $CL = 25\text{pF}$

- SSP の SPI モード(マスタ)

$$f_{PCLK} \geq 2 \times SPxCLK(\text{最高})$$

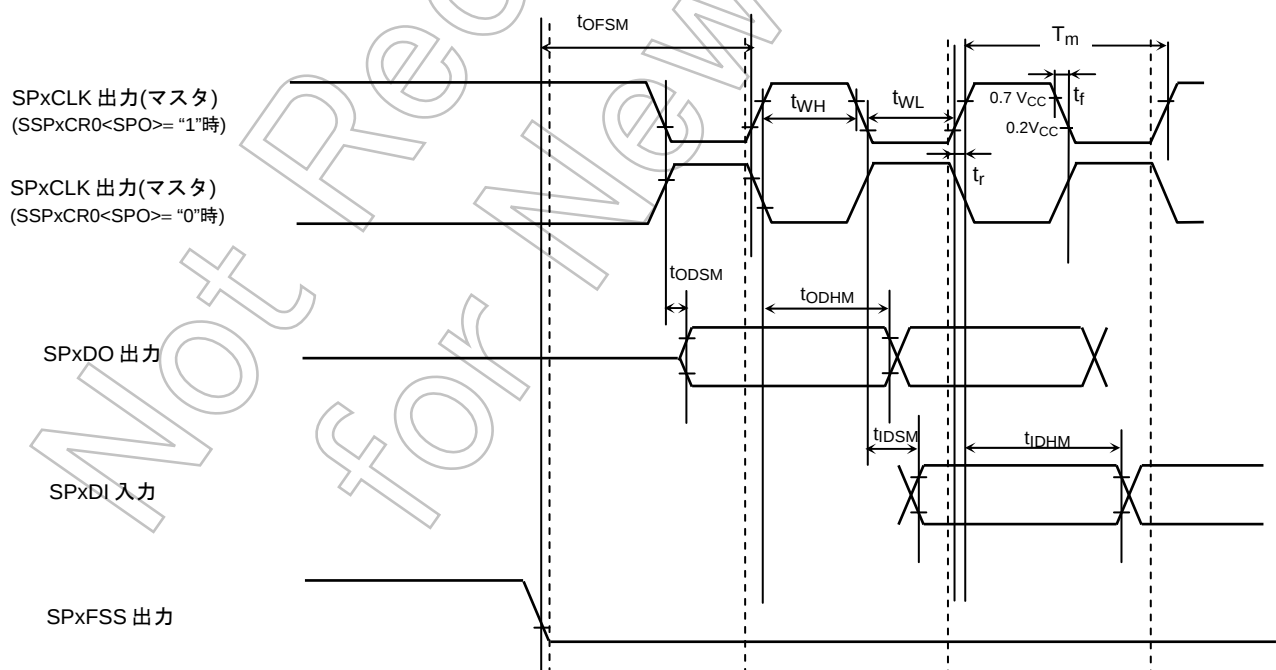
$$f_{PCLK} \geq 65204 \times SPxCLK(\text{最低})$$

(1) マスタ SSPxCR0<SPH>= "0"(1st エッジでデータをラッチ)



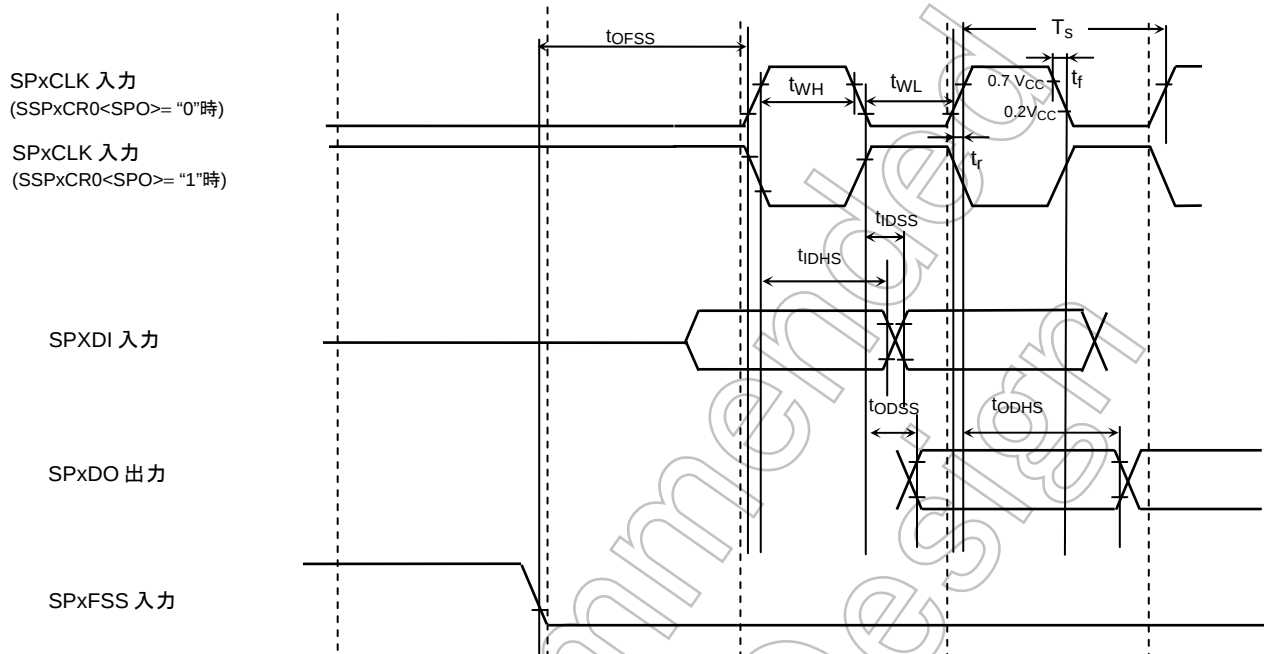
- SSP の SPI モード(マスタ)

(2) マスタ SSPxCR0<SPH>= "1" (2nd エッジでデータをラッチ)

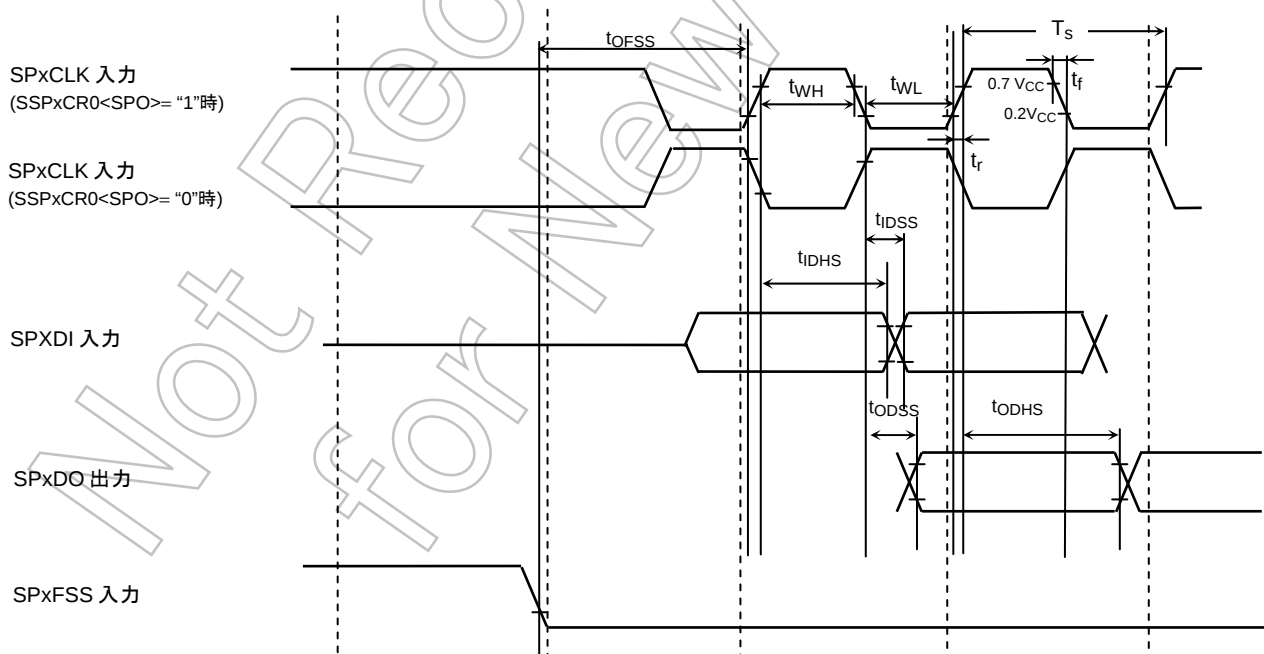


- SSP の SPI モード(スレーブ)
 - $f_{\text{PCLK}} \geq 12 \times \text{SPxCLK}$ (最高)
 - $f_{\text{PCLK}} \geq 65204 \times \text{SPxCLK}$ (最低)

(3) スレーブ SSPxCR0<SPH>= "0"(1st エッジでデータをラッチ)



- SSP の SPI モード(スレーブ)
 - (4) スレーブ SSPxCR0<SPH>= "1" (2nd エッジでデータをラッチ)



4.3.8 I²S

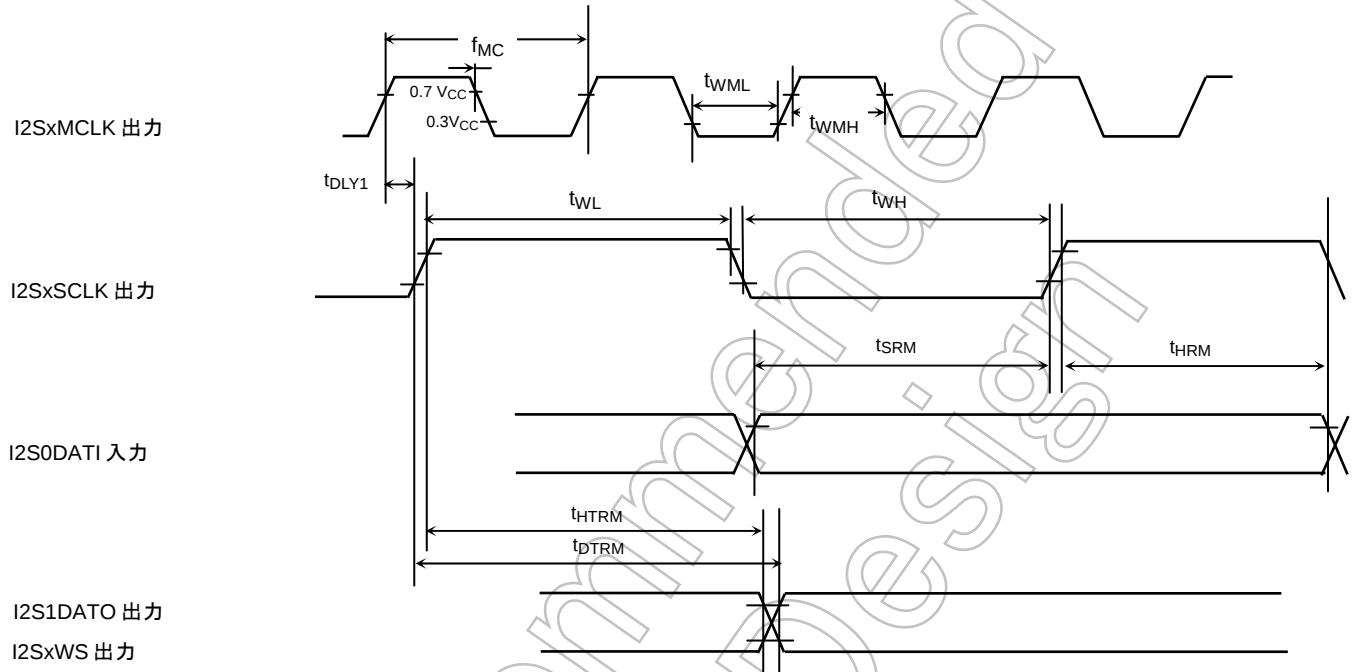
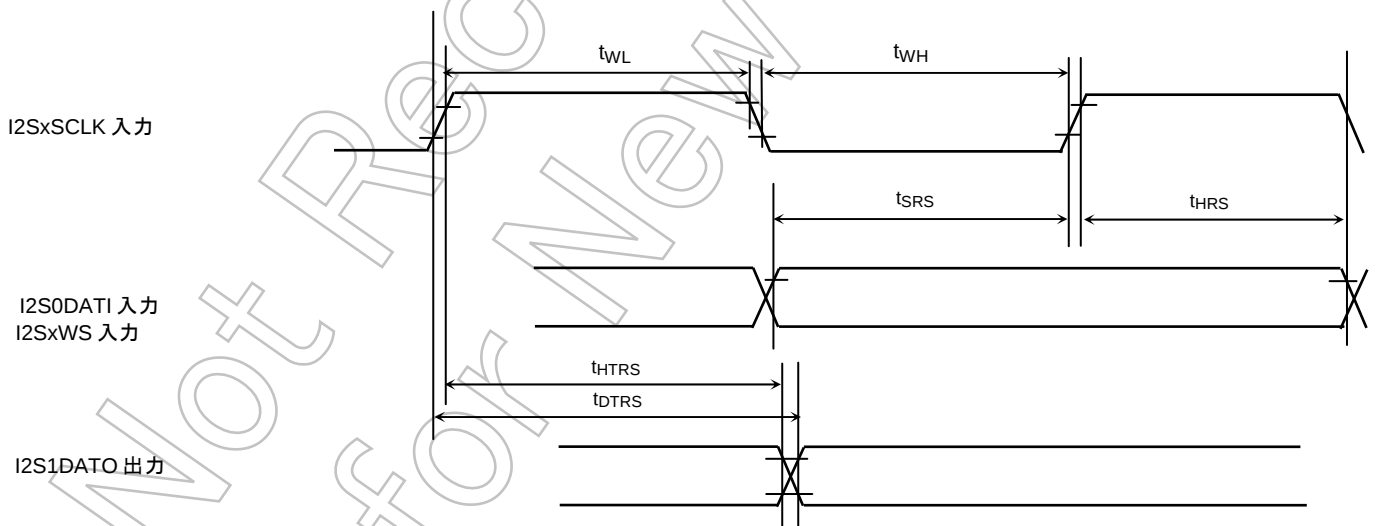
AC 測定条件

- 表中の計算式に使われる" T "は内部バス PCLK 周期を示し、CPU のクロック f_{CLK} 周期の 1/2 を示しています。 $f_{\text{PCLK}} = 16 \times I2Sx\text{SCLK}$
- 表中の計算式に使われる" t "は f_{OSCH} 周期の を示しています。
- 出力レベル: High = $0.7 \times \text{DVCC3IO}$, Low = $0.3 \times \text{DVCC3IO}$
- 入力レベル: High = $0.9 \times \text{DVCC3IO}$, Low = $0.1 \times \text{DVCC3IO}$

注) 表中の "計算式" は $\text{DVCC3IO}=1.8\sim3.6\text{V}$ 、 $\text{DVCC1A}=\text{DVCC1B}=\text{DVCC1C}=1.4\sim1.6\text{V}$ の範囲での規定を示します。
Slave Mode 時、I2SxCLK の入力から安定時間が必要です。

項目	記号	計算式		100MHz	96 MHz	単位
		Min	Max			
I2SxMCLK 出力 f_{OSCH}	f_{MC}	$t/4$	$t/1$	25	24	MHz
I2SxMCLK 出力高レベルパルス幅	t_{WMH}	10	—	10	10	ns
I2SxMCLK 出力低レベルパルス幅	t_{WML}	10	—	10	10	
I2SxCLK 出カクロック周期	f_{SCKM}	333	—	333	333	
I2SxCLK 入カクロック周期	f_{SCKS}	160	—	160	160	
I2SxCLK 出力高レベルパルス幅	t_{WH}	$0.45 f_{\text{SCKM}}$	—	149	149	
I2SxCLK 出力低レベルパルス幅	t_{WL}	$0.45 f_{\text{SCKM}}$	—	149	149	
マスタモード I2S1DATO, I2SxWS Hold 時間	t_{HTRM}	$0.5 f_{\text{SCKM}} + 2T$	—	186	187	
マスタモード I2S1DATO, I2SxWS 遅延時間	t_{DTRM}	—	$0.5 f_{\text{SCKM}} + 3T + 10$	206	207	
マスタモード I2S0DATI セットアップ時間	t_{SRM}	10.0	—	10	10	
マスタモード I2S0DATI Hold 時間	t_{HRM}	$0.2 f_{\text{SCKM}}$	—	66	66	
マスタモード I2SxMCLK, I2SxSCLK 遅延時間	t_{DLY1}	—	10.0	10	10	
スレーブモード I2S1DATO, I2SxWS Hold 時間	t_{HTRS}	$0.5 f_{\text{SCKS}} + 2T$	—	100	100	
スレーブモード I2S1DATO, I2SxWS 遅延時間	t_{DTRS}	—	$0.8 f_{\text{SCKS}} + 3T + 20$	178	179	
スレーブモード I2S0DATI セットアップ時間	t_{SRS}	10.0	—	10	10	
スレーブモード I2S0DATI Hold 時間	t_{HRS}	$0.2 f_{\text{SCKS}}$	—	32	32	

AC 測定条件 : 付加容量 $\text{CL} = 25 \text{ pF}$

(1) I²S マスタモード(2) I²S スレーブモード

4.4 AD変換特性

注) 表中の“計算式”は $AVCC3AD = 3.0 \sim 3.6V$ 、 $DVCC1A = DVCC1B = DVCC1C = 1.4 \sim 1.6V$ の範囲での規定を示します。

項目	記号	条件	Min	Typ.	Max	単位
アナログ基準電圧(+)	VREFH		AVCC3AD	AVCC3AD	AVCC3AD	V
アナログ基準電圧(-)	VREFL		DVSSCOMn	DVSSCOMn	DVSSCOMn	
ADコンバータ電源供給電圧	AVCC3AD		3.0	3.3	3.6	
ADコンバータ GND	AVSS		DVSSCOMn	DVSSCOMn	DVSSCOMn	
アナログ入力電圧	AVIN		VREFL		VREFH	
アナログ基準電圧 電源電流	IREFON	$\langle VREFON \rangle = 1$		2.1	3.5	mA
	IREFOFF	$\langle VREFON \rangle = 0$		0.1	10	μA
フルスケール誤差	EFULL			+1	-1~+4	LSB
オフセット誤差	EOFF			-3	-4~+1	LSB
微分非直線性	EDNL			-1~+2	± 2	LSB
積分非直線性	EINL			-2~+3	± 3	LSB

注 1) 記の誤差は、誤差 = (変換結果 - 理論値) であり、また $1LSB = (VREFH - VREFL) \div 1024$ で定義しています。

注 2) 量子化誤差は含んでいません。

注 3) ADコンバータの最大動作クロック (ADCLK) は 33MHz で、最低動作クロックは 3MHz です。最小変換時間は 33MHz 時の $1.39\mu s$ で最大変換時間は 3MHz 時の $15.3\mu s$ です。

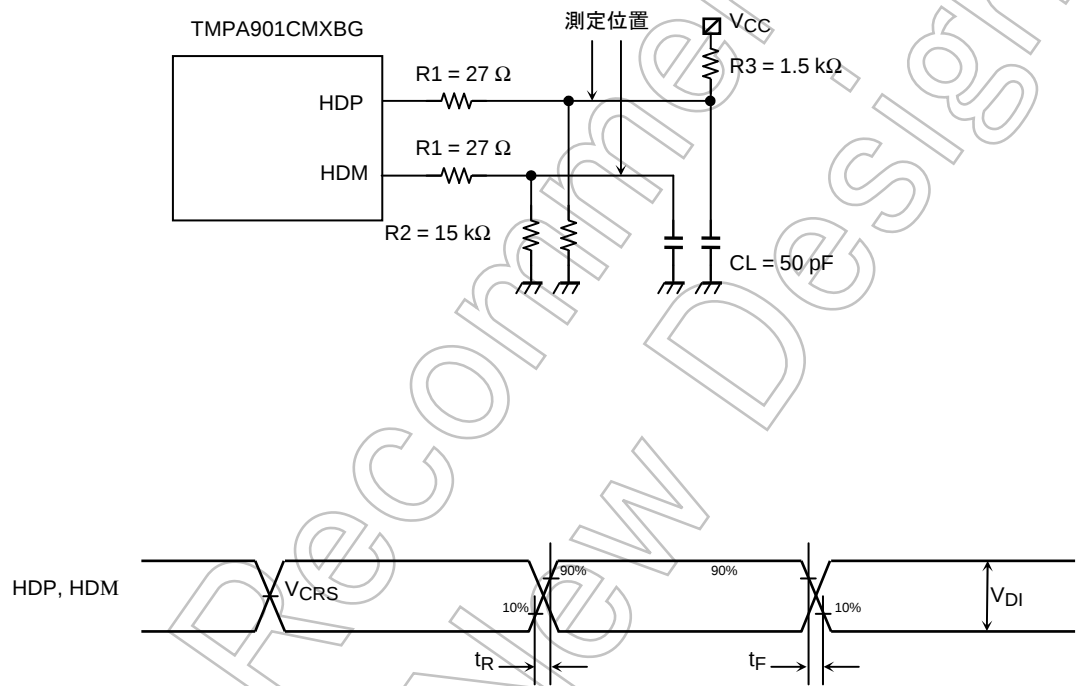
4.5 USBホストコントローラ

$AVCC3H = 3.3 \pm 0.3 \text{ V} / f_{\text{USB}} = 48 \text{ MHz}$

項目	記号	Min	Max	単位	備考
HDP, HDM 立ち上がり時間	t_R	4	20	nS	Full Speed
HDP, HDM 立ち下がり時間	t_F	4	20		Full Speed
差動共通モード範囲	VDI	0.2		V	$ (\text{HDP}) - (\text{HDM}) $
出力信号交差電圧	V_{CRS}	1.3	2.0		V

AC 測定条件

<Full Speed>

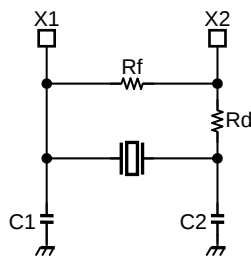


4.6 推奨発振回路

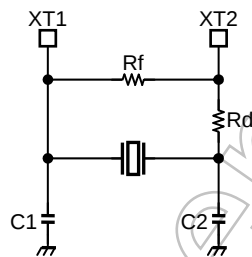
本製品は、下記の発振子メーカーにて評価して頂きました。発振子の選択時に活用願います。

注) 発振端子のトータル負荷容量は接続する外付け(または内蔵) 負荷容量 C1、C2 と、実装基板上の浮遊容量の和になります。C1、C2 の定数を使用した場合でも実装基板によりトータル負荷容量が異なり、誤動作する可能性があります。基板設計の際は発振回路周辺のパターンが最短距離になるようにしてください。また、実際に使用される実装基板での発振評価を行うことを推奨いたします。

(1) 接続例



高周波発振器の接続図



低周波発振器の接続図

(2) 推奨セラミック発振子: 京セラキンセキ株式会社

下表に高周波発振回路の推奨回路定数を示します。

MCU	発振 周期数 [MHz]	タイプ	推奨発振子	当社推奨定数				動作推奨動作条件	
				C1 [pF]	C2 [pF]	Rd [Ω]	Rf [Ω]	電源電圧範囲 [V]	温度範囲 [°C]
TMPA901CMXBG	27.000	SMD	CX2520SB27000B0HLQZ1	6	6	0	Open	1.4 ~ 1.6	-20 ~ 85
	24.000	SMD	CX2520SB24000C0HLQZ1	7	7				

注 1) 定数 C1, C2 の () は、負荷容量内蔵タイプを表しています。

品名末尾の「-」以降は梱包仕様を示します。

注 2) 京セラキンセキ製発振子は、型番・仕様の切り替えが随時行われております。

詳細につきましては、下記アドレスの同社ホームページをご参照ください。

<http://www.kinseki.co.jp/>

注 3) 高周波発振回路でセラミック発振子を使用する場合は、USB デバイスおよびホストコントローラは高精度のクロックが必要ですので、X1USB 端子より 24MHz ± 100ppm 以下の精度クロックを使用してください。

(3) 推奨低周波数水晶発振子: 京セラキンセキ株式会社

下表に低周波発振回路の推奨回路定数を示します。

MCU	発振 周期数 [kHz]	推奨発振子	当社推奨定数				動作推奨動作条件	
			C1 [pF]	C2 [pF]	Rd [kΩ]	Rf [Ω]	電源電圧範囲 [V]	温度範囲[°C]
TMPA901CMXBG	32.768	ST3215SB	7	7	820	内蔵	3.0 ~ 3.6	-20 ~ 85

注) 京セラキンセキ製発振子は、型番・仕様の切り替えが随時行われております。

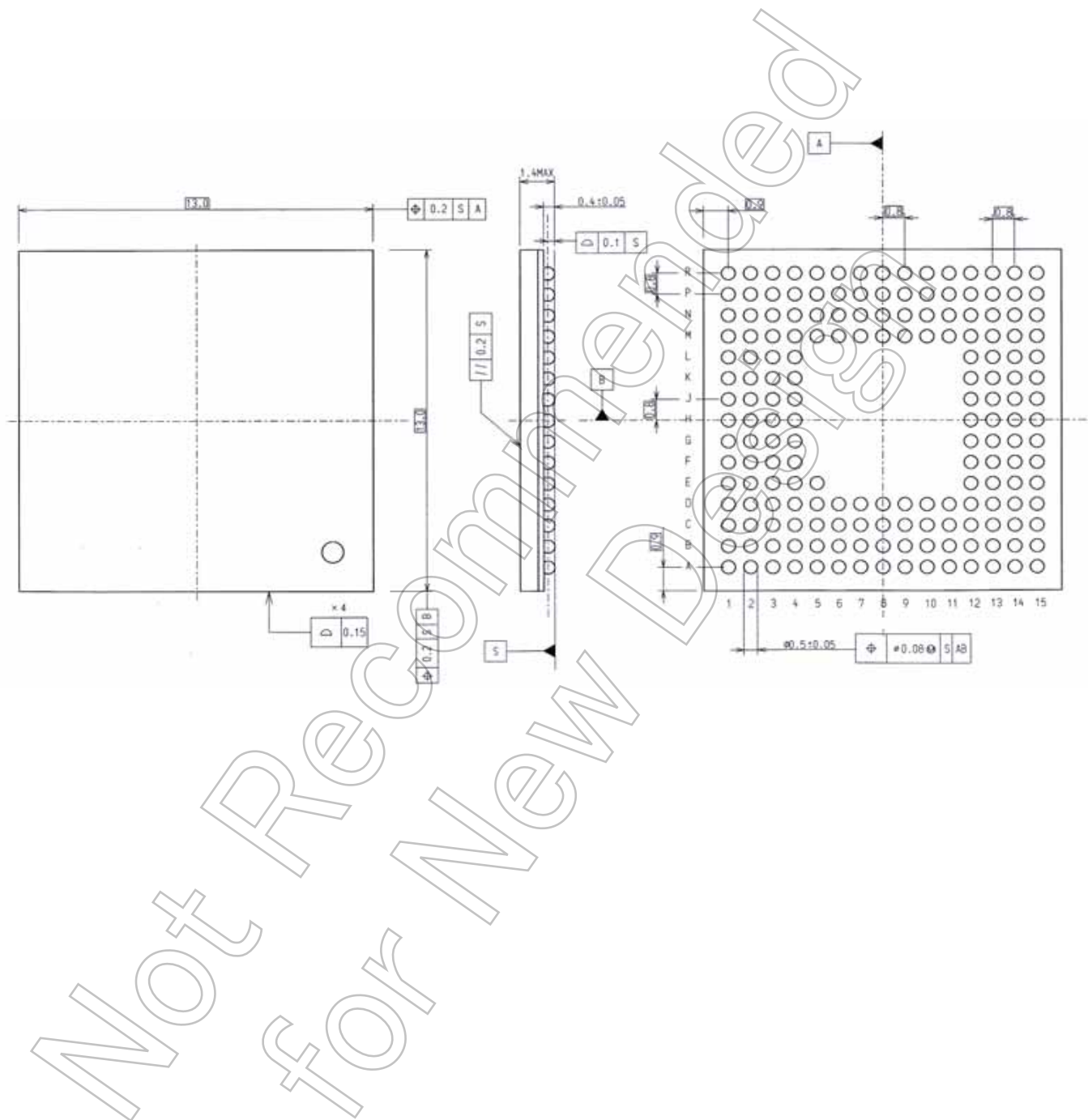
詳細につきましては、下記アドレスの同社ホームページをご参照ください。

<http://www.kinseki.co.jp/>

5. 外形寸法図

Package name: P-FBGA177-1313-0.8C4

Unit: mm



製品取り扱い上のお願い

- 本資料に掲載されているハードウェア、ソフトウェアおよびシステム（以下、本製品という）に関する情報等、本資料の掲載内容は、技術の進歩などにより予告なしに変更されることがあります。
- 文書による当社の事前の承諾なしに本資料の転載複製を禁じます。また、文書による当社の事前の承諾を得て本資料を転載複製する場合でも、記載内容に一切変更を加えたり、削除したりしないでください。
- 当社は品質、信頼性の向上に努めていますが、半導体製品は一般に誤作動または故障する場合があります。本製品をご使用頂く場合は、本製品の誤作動や故障により生命・身体・財産が侵害されることのないように、お客様の責任において、お客様のハードウェア・ソフトウェア・システムに必要な安全設計を行うことをお願いします。なお、設計および使用に際しては、本製品に関する最新の情報（本資料、仕様書、データシート、アプリケーションノート、半導体信頼性ハンドブックなど）および本製品が使用される機器の取扱説明書、操作説明書などをご確認の上、これに従ってください。また、上記資料などに記載の製品データ、図、表などに示す技術的な内容、プログラム、アルゴリズムその他応用回路例などの情報を使用する場合は、お客様の製品単独およびシステム全体で十分に評価し、お客様の責任において適用可否を判断してください。
- 本製品は、一般的電子機器（コンピュータ、パーソナル機器、事務機器、計測機器、産業用ロボット、家電機器など）または本資料に個別に記載されている用途に使用されることが意図されています。本製品は、特別に高い品質・信頼性が要求され、またはその故障や誤作動が生命・身体に危害を及ぼす恐れ、膨大な財産損害を引き起こす恐れ、もしくは社会に深刻な影響を及ぼす恐れのある機器（以下“特定用途”という）に使用されることは意図されていませんし、保証もされていません。特定用途には原子力関連機器、航空・宇宙機器、医療機器、車載・輸送機器、列車・船舶機器、交通信号機器、燃焼・爆発制御機器、各種安全関連機器、昇降機器、電力機器、金融関連機器などが含まれます。本資料に個別に記載されている場合を除き、本製品を特定用途に使用しないでください。
- 本製品を分解、解析、リバースエンジニアリング、改造、改変、翻案、複製等しないでください。
- 本製品を、国内外の法令、規則及び命令により、製造、使用、販売を禁止されている製品に使用することはできません。
- 本資料に掲載してある技術情報は、製品の代表的動作・応用を説明するためのもので、その使用に際して当社及び第三者の知的財産権その他の権利に対する保証または実施権の許諾を行うものではありません。
- 別途、書面による契約またはお客様と当社が合意した仕様書がない限り、当社は、本製品および技術情報に関して、明示的にも黙示的にも一切の保証（機能動作の保証、商品性の保証、特定目的への合致の保証、情報の正確性の保証、第三者の権利の非侵害保証を含むがこれに限らない。）をしておりません。
- 本製品、または本資料に掲載されている技術情報を、大量破壊兵器の開発等の目的、軍事利用の目的、あるいはその他軍事用途の目的で使用しないでください。また、輸出に際しては、「外国為替及び外国貿易法」、「米国輸出管理規則」等、適用ある輸出関連法令を遵守し、それらの定めるところにより必要な手続を行ってください。
- 本製品の RoHS 適合性など、詳細につきましては製品個別に必ず弊社営業窓口までお問合せください。本製品のご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用ある環境関連法令を十分調査の上、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は一切の責任を負いかねます。