

**32 ビット RISC マイクロコントローラー
リファレンスマニュアル**

**シリアルメモリーインターフェース
(SMIF-D)**

Revision 1.0

2026-01

東芝デバイス&ストレージ株式会社

目次

序章	5
関連するドキュメント	5
表記規約	6
用語・略語	8
1. 概要	9
2. 構成	10
3. 機能説明・動作説明	11
3.1. クロック供給	11
3.2. 通信モード	11
3.3. メモリーマッピング	12
3.4. アクセスモード	12
3.4.1. ダイレクトアクセス	12
3.4.2. インダイレクトアクセス	18
3.4.3. インダイレクトアクセスの設定手順	18
3.5. 転送クロック	19
3.6. データ入出力タイミング	20
3.7. 割り込み	20
4. レジスター説明	21
4.1. レジスター一覧	21
4.2. レジスター詳細	22
4.2.1. [SMIxMAP0] (アドレスマップコントロールレジスター0)	22
4.2.2. [SMIxMAP1] (アドレスマップコントロールレジスター1)	23
4.2.3. [SMIxDACRn] (ダイレクトアクセスコントロールレジスターn) (n = 0、1)	24
4.2.4. [SMIxDRCRn] (ダイレクトリードコントロールレジスターn) (n = 0、1)	25
4.2.5. [SMIxDWCRn] (ダイレクトライトコントロールレジスターn) (n = 0、1)	26
4.2.6. [SMIxRACR0] (インダイレクトアクセスコントロールレジスター0)	27
4.2.7. [SMIxRACR1] (インダイレクトアクセスコントロールレジスター1)	28
4.2.8. [SMIxIOCR] (インダイレクトアクセス I/O コントロールレジスター)	29
4.2.9. [SMIxOECR] (インダイレクトアクセス Output イネーブルコントロールレジスター)	29
4.2.10. [SMIxINT] (割り込みコントロールレジスター)	30
4.2.11. [SMIxSTAT] (ステータスレジスター)	31
4.2.12. [SMIxSWR] (ソフトウェアリセットレジスター)	32
4.2.13. [SMIxACKR] (アディショナルクロック制御レジスター)	32
4.2.14. [SMIxCCOR] (転送クロック/CS 端子出力制御レジスター)	33
4.2.15. [SMIxSTPR] (強制停止制御レジスター)	34
4.2.16. [SMIxPBUFm] (インダイレクトアクセスプライマリーバッファレジスターn) (n = 0 ~ 7)	34
4.2.17. [SMIxSBUFm] (インダイレクトアクセスセカンダリーバッファレジスターm) (m = 0 ~ 63)	34
5. 使用方法の例	35

5.1. 初期設定	35
5.1.1. ダイレクトアクセスとインダイレクトアクセスの共通設定	35
5.1.2. ダイレクトアクセスシリアルメモリー0 設定	36
5.1.3. ダイレクトアクセスシリアルメモリー1 設定	36
5.1.4. インダイレクトアクセス設定	37
5.1.5. 転送クロック	37
5.2. ダイレクトアクセスに関するプログラミング方法	37
5.3. インダイレクトアクセスに関するプログラミング方法	38
5.3.1. 基本手順	38
5.3.2. プログラミング例: PAGE PROGRAM、WRITE	40
5.3.3. プログラミング例: Quad I/O PAGE PROGRAM、Quad I/O WRITE	42
5.3.4. プログラミング例: QPI PAGE PROGRAM、QPI WRITE	44
5.3.5. プログラミング例: Octal I/O PAGE PROGRAM、Octal I/O WRITE	46
5.3.6. プログラミング例: OPI PAGE PROGRAM、OPI WRITE	48
5.3.7. プログラミング例: SECTOR ERASE	50
5.3.8. プログラミング例: CHIP ERASE	51
5.3.9. プログラミング例: READ STATUS REGISTER	52
5.3.10. プログラミング例: FAST READ	53
5.3.11. プログラミング例: Quad I/O FAST READ	55
5.3.12. プログラミング例: QPI FAST READ	57
5.3.13. プログラミング例: Octal I/O FAST READ	59
5.3.14. プログラミング例: OPI FAST READ	61
5.4. その他	63
5.4.1. 強制停止	63
5.4.2. ソフトウェアリセット	63
5.5. シリアルメモリーとの接続例	64
5.5.1. Standard SPI での接続例	65
5.5.2. Dual/DPI での接続例	67
5.5.3. Quad/QPI での接続例	69
5.5.4. Octal/OPI での接続例	71
6. 使用上のご注意およびお願い事項	73
7. 改訂履歴	74
製品取り扱い上のお願い	75

図目次

図 2.1	SMIF のブロック図.....	10
図 3.1	メモリーマッピングの例	12
図 3.2	STR-SPI: FAST READ	13
図 3.3	STR-Dual: FAST READ Dual Output	13
図 3.4	STR-Dual: FAST READ Dual I/O	13
図 3.5	STR-DPI: FAST READ	14
図 3.6	STR-Quad: FAST READ Quad Output	14
図 3.7	STR-Quad: FAST READ Quad I/O.....	14
図 3.8	STR-QPI: FAST READ	15
図 3.9	STR-Octal: FAST READ Octal Output	15
図 3.10	STR-Octal: FAST READ Octal I/O	16
図 3.11	STR-OPI: FAST READ	16
図 3.12	インダイレクトアクセス	18
図 3.13	データ入出力タイミング	20
図 5.1	PAGE PROGRAM、WRITE	41
図 5.2	Quad I/O PAGE PROGRAM、Quad I/O WRITE	43
図 5.3	QPI PAGE PROGRAM、QPI WRITE	45
図 5.4	Octal I/O PAGE PROGRAM、Octal I/O WRITE	47
図 5.5	OPI PAGE PROGRAM、OPI WRITE	49
図 5.6	FAST READ	54
図 5.7	Quad I/O FAST READ	56
図 5.8	QPI FAST READ	58
図 5.9	Octal FAST READ	60
図 5.10	OPI FAST READ	62
図 5.11	Standard SPI の接続例 (シリアルメモリー0).....	65
図 5.12	Standard SPI の接続例 (シリアルメモリー0/1).....	66
図 5.13	Dual/DPI の接続例 (シリアルメモリー0).....	67
図 5.14	Dual/DPI の接続例 (シリアルメモリー0/1).....	68
図 5.15	Quad/QPI の接続例 (シリアルメモリー0).....	69
図 5.16	Quad/QPI の接続例 (シリアルメモリー0/1).....	70
図 5.17	Octal/OPI の接続例 (シリアルメモリー0)	71
図 5.18	Octal/OPI の接続例 (シリアルメモリー0/1)	72

表目次

表 2.1	接続仕様.....	10
表 3.1	転送クロックの周波数.....	19
表 7.1	改訂履歴.....	74

序章

関連するドキュメント

文書名
各製品のデータシート
クロック制御と動作モード
例外
入出力ポート

表記規約

- 数値表記は以下の規則に従います。
16 進数表記: 0xABC
10 進数表記: 123 または 0d123 (10 進表記であることを示す必要のある場合だけ使用)
2 進数表記: 0b111 (ビット数が本文中に明記されている場合は「0b」を省略可)
- ローアクティブの信号は信号名の末尾に「_N」で表記します。
- 信号がアクティブレベルに移ることを「アサート (assert)」アクティブでないレベルに移ることを「デアサート (deassert)」と呼びます。
- 複数の信号名は[m: n]とまとめて表記する場合があります。
例: S[3:0] は S3、S2、S1、S0 の 4 つの信号名をまとめて表記しています。
- 本文中[]で囲まれたものはレジスターを定義しています。
例: [ABCD]
- 同種で複数のレジスター、フィールド、ビット名は「n」で一括表記する場合があります。
例: [XYZ1]、[XYZ2]、[XYZ3] → [XYZn]
- 「レジスター一覧」中のレジスター名でユニットまたはチャンネルは「x」で一括表記しています。
ユニットの場合、「x」は A、B、C...を表します。
例: [ADACR0]、[ADBCR0]、[ADCCR0] → [ADxCR0]
チャンネルの場合、「x」は 0、1、2、..を表します。
例: [T32A0RUNA]、[T32A1RUNA]、[T32A2RUNA] → [T32AxRUNA]
- レジスターのビット範囲は[m: n]と表記します。
例: [3:0] はビット 3 から 0 の範囲を表します。
レジスターの設定値は 16 進数または 2 進数のどちらかで表記されています。
例: [ABCD]<EFG> = 0x01 (16 進数)、[XYZn]<VW> = 1 (2 進数)
- ワード、バイトは以下のビット長を表します。
バイト: 8 ビット
ハーフワード: 16 ビット
ワード: 32 ビット
ダブルワード: 64 ビット
- レジスター内の各ビットの属性は以下の表記を使用しています。
R: リードオンリー
W: ライトオンリー
R/W: リード/ライト
- 断りのない限り、レジスターアクセスはワードアクセスだけをサポートします。
- 本文中の予約領域「Reserved」として定義されたレジスターは書き換えを行わないでください。
また、読み出した値を使用しないでください。
- Default 値が「-」となっているビットから読み出した値は不定です。
- 書き込み可能なビットフィールドと、リードオンリー「R」のビットフィールドが共存するレジスターに書き込みを行う場合、リードオンリー「R」のビットフィールドには Default 値を書き込んでください。
Default 値が「-」となっている場合は、個々のレジスターの定義に従ってください。
- ライトオンリーのレジスターの Reserved ビットフィールドには Default 値を書き込んでください。
Default 値が「-」となっている場合は、個々のレジスターの定義に従ってください。
- 書き込みと読み出しで異なる定義のレジスターへのリードモディファイライト処理は行わないでください。

本資料に記載されている社名・商品名・サービス名などは、それぞれ各社が商標として使用している場合があります。

用語・略語

この仕様書で使用されている用語・略語の一部を記載します。

DPI	Dual Peripheral Interface
OPI	Octal Peripheral Interface
SI	Serial Input
SMIF	Serial Memory Interface
SO	Serial Output
SPI	Serial Peripheral Interface
STR	Single Transfer Rate
QPI	Quad Peripheral Interface

1. 概要

シリアルメモリーインターフェース(SMIF)は、シリアルメモリーなどのシリアルまたはマルチ I/O を通信インターフェースとして備えた外部デバイスとの接続を行うためのインターフェースです。SMIF が転送クロックとチップセレクトを出力します。

このリファレンスマニュアルでは、それらを受けるデバイスとして主にシリアルメモリーを対象として記述されています。

以下に、SMIF の機能一覧を示します。

なお、使用できる通信モードは端子仕様によります。端子仕様については製品のデータシートを参照してください。

機能分類	機能	内容
シリアルメモリーとの接続	接続数	● 最大 2 つのシリアルメモリーを接続可能
	接続容量	● 64K バイト ~ 128M バイト
	通信モード	● 通信モード STR-SPI (Standard SPI コンパチブル) STR-Dual (ダイレクトアクセスのみ) STR-DPI (ダイレクトアクセスのみ) STR-Quad STR-QPI STR-Octal STR-OPI ● MSB ファースト ● SPI Mode 0 サポート ● アドレスありコマンドでのアドレスバイト数を 2、3、4 バイトから指定可能 注) データストローブやデータマスク信号を用いた通信は非サポート。
	メモリーマッピング	● アドレス"0xA0000000" ~ "0xA7FFFFFF"の任意領域へマッピング可能
	アクセスモード	● ダイレクトアクセス ● インダイレクトアクセス
	コマンド転送数	● レジスターを介して最大 288 バイト転送可能
	チップセレクト	● シリアルメモリー0、シリアルメモリー1 を選択 ● SMiXCS0_N、SMiXCS1_N のデアサート期間を設定可能
	その他機能	● ダイレクトアクセスの前に SPI Flash のステータスレジスターの WIP を自動的にポーリングし、Idle 状態までウェイトする機能 ● ダミーバイト数を 0 ~ 31 バイトで調整可能

2. 構成

SMIF のブロック図と信号一覧を示します。

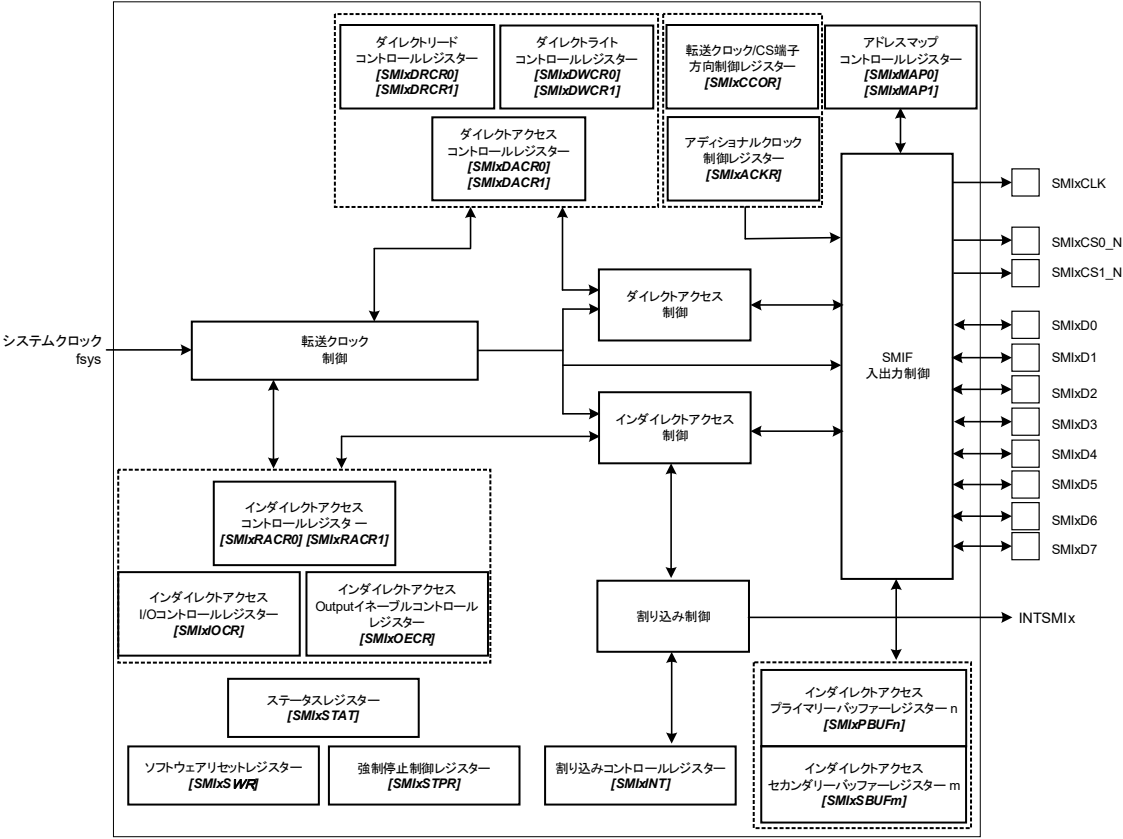


図 2.1 SMIFのブロック図

表 2.1 接続仕様

No.	信号名	信号名称	I/O	参照リファレンスマニュアル
1	fsys	システムクロック	入力	クロック制御と動作モード
2	SMIXCLK	転送クロック	出力	データシート、入出力ポート
3	SMIXCS0_N	チップセレクト 0	出力	データシート、入出力ポート
4	SMIXCS1_N	チップセレクト 1	出力	データシート、入出力ポート
5	SMIXD0	データ入出力 0	入出力	データシート、入出力ポート
6	SMIXD1	データ入出力 1	入出力	データシート、入出力ポート
7	SMIXD2	データ入出力 2	入出力	データシート、入出力ポート
8	SMIXD3	データ入出力 3	入出力	データシート、入出力ポート
9	SMIXD4	データ入出力 4	入出力	データシート、入出力ポート
10	SMIXD5	データ入出力 5	入出力	データシート、入出力ポート
11	SMIXD6	データ入出力 6	入出力	データシート、入出力ポート
12	SMIXD7	データ入出力 7	入出力	データシート、入出力ポート
13	INTSMIX	割り込み	出力	例外

3. 機能説明・動作説明

SMIF は、容量 64K バイトから 128M バイトまでのシリアルメモリーを最大 2 つ接続可能です。

アドレスを指定してデータをリード/ライトする「ダイレクトアクセス」とプログラムレジスターを操作してコマンドを発行する「インダイレクトアクセス」によりアクセスを行います。

3.1. クロック供給

SMIF を使用する場合は、fsys 供給停止レジスターA (*JCGFSYSENA*)/(*JCGFSYSMENA*)、fsys 供給停止レジスターB (*JCGFSYSENB*)/(*JCGFSYSMENB*)、fsys 供給停止レジスターC (*JCGFSYSMENC*)、fc 供給停止レジスター (*JCGFCEN*)で該当するクロックイネーブルビットを"1" (クロック供給)に設定してください。

該当レジスター、ビット位置は製品によって異なります。そのため、製品によってレジスターが存在しない場合があります。詳細はリファレンスマニュアル「クロック制御と動作モード」を参照してください。

クロック供給を停止する場合や STOP1/STOP2 モードに遷移する際は、SMIF が停止していることを確認してください。

3.2. 通信モード

SMIF とシリアルメモリーの通信は、STR-SPI (Standard SPI コンパチブル)、STR-Dual (ダイレクトアクセスのみ)、STR-DPI (ダイレクトアクセスのみ)、STR-Quad、STR-QPI、STR-Octal、STR-OPI に関してリードライトをサポートしています。

接続するシリアルメモリーは、以下の条件を満たしている必要があります。

- 容量: 64K バイト ~ 128M バイト
- SPI Mode 0 をサポート
- アドレスの使用しない上位ビットは don't care
- ステータスレジスターのビット 0 は Write In Progress (WIP)を表す (シリアルメモリーが SPI Flash の場合)
- MSB ファースト

3.3. メモリーマッピング

シリアルメモリーへアクセスするためのアドレスは、"0xA0000000" ~ "0xA7FFFFFF" (128M バイト) の任意領域へマッピング可能です。メモリーマッピングされていない領域をリードした場合は不定値が読み出されます。リセット後、シリアルメモリー0 はメモリーマッピングされていますが、シリアルメモリー1 はメモリーマッピングされていません。シリアルメモリー1 を使用する際は、メモリーマッピングを設定する必要があります。

設定したマッピング領域よりも小さい容量のシリアルメモリーを実装した場合、未実装領域をアクセスすると実装されているシリアルメモリーのミラーが見えます。

「図 3.1 メモリーマッピングの例」に、シリアルメモリー0 の領域として"0xA0000000" ~ "0xA0BFFFFFF"の 12M バイト、シリアルメモリー1 の領域として"0xA0C00000" ~ "0xA0FFFFFF"の 4M バイトをマッピングし、それぞれに 4M バイトのシリアルメモリーを実装した場合の例を示します。

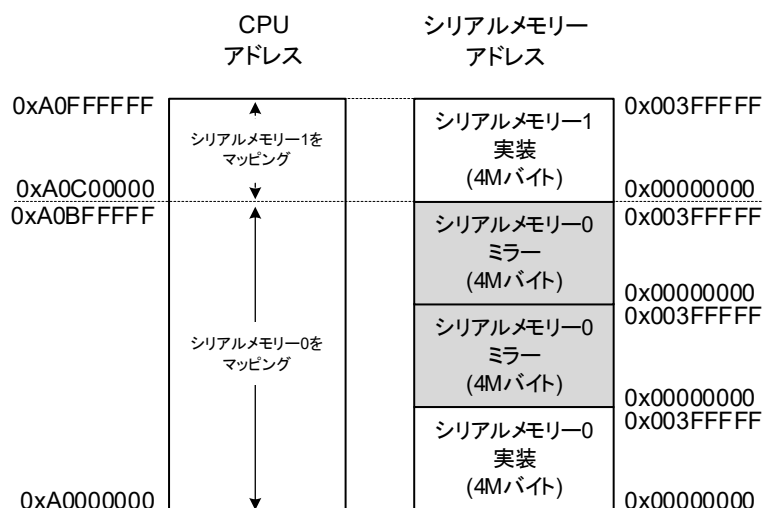


図 3.1 メモリーマッピングの例

3.4. アクセスモード

シリアルメモリーのアクセスは、アドレス指定でデータをリード/ライトする「ダイレクトアクセス」とプログラムレジスターを操作してコマンドを発行する「インダイレクトアクセス」による方法があります。

3.4.1. ダイレクトアクセス

シリアルメモリーに対するリードライトは、アドレス"0xA0000000" ~ "0xA7FFFFFF"から直接行えます。このアクセスをダイレクトアクセスと呼びます。ダイレクトアクセスによるリードライトは、アドレス"0xA0000000" ~ "0xA7FFFFFF"へのリードライトが検出されるとシリアルメモリーに対して Read/Write コマンドが発行されることで行われます。

3.4.1.1. シリアルメモリーコマンド

ダイレクトアクセスで発行されるコマンドは、FAST READ (オペコード = 0x0B)が初期値となりますが、以下のマルチ I/O コマンドに変更可能です。ただし、各コマンドの使用可否は、接続するシリアルメモリーにより異なります。

以下に、主な転送モードとそのタイミングチャートを示します。

(1) STR-SPI

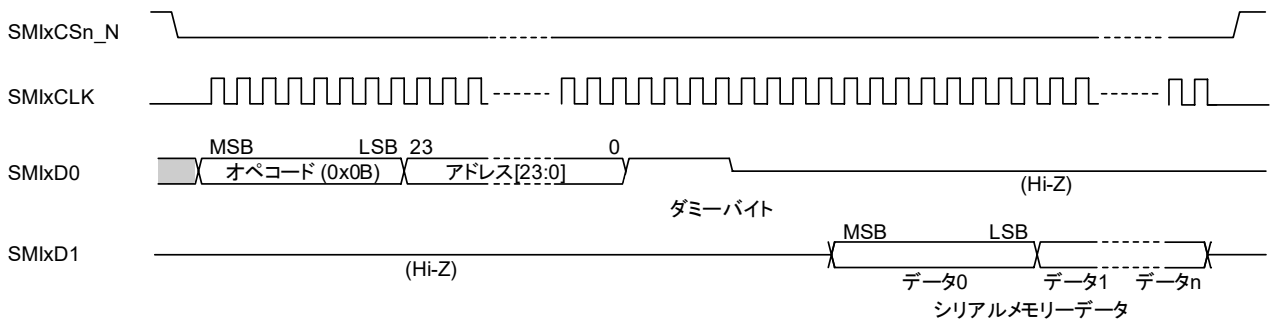


図 3.2 STR-SPI: FAST READ

(2) STR-Dual (ダイレクトアクセスのみ)

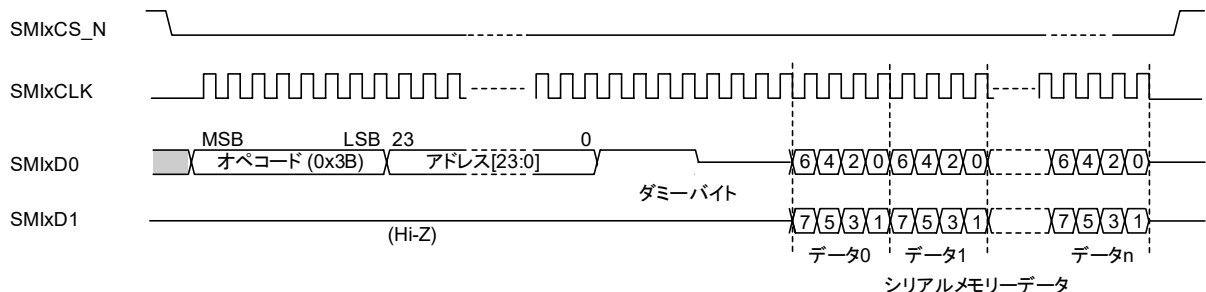


図 3.3 STR-Dual: FAST READ Dual Output

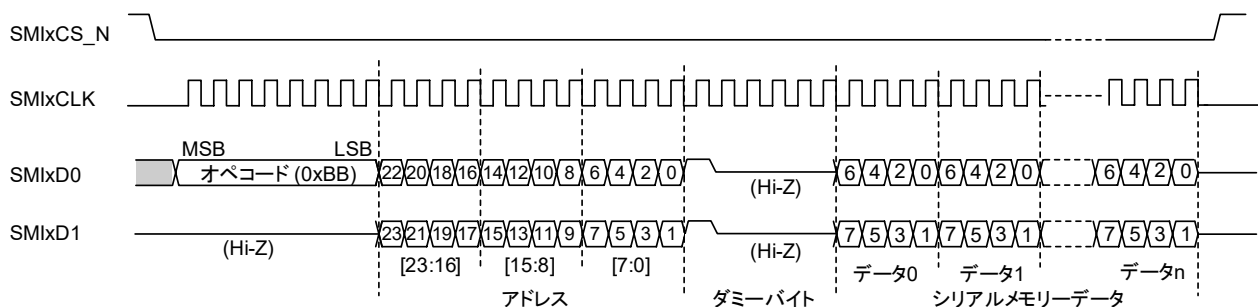


図 3.4 STR-Dual: FAST READ Dual I/O

(3) STR-DPI (ダイレクトアクセスのみ)

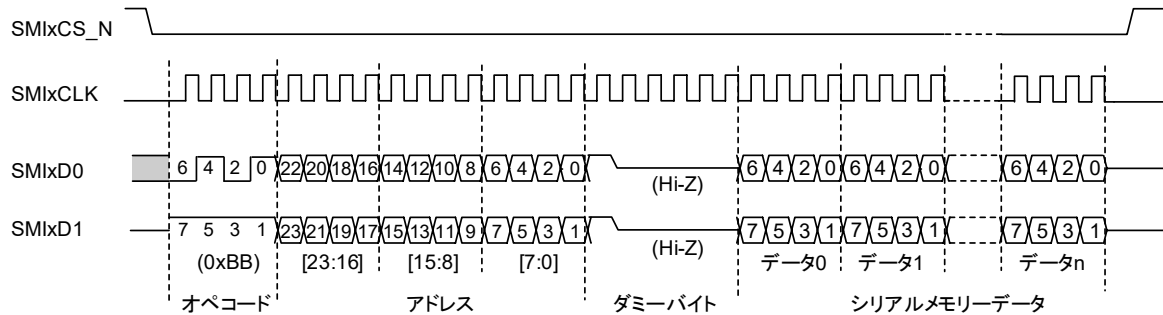


図 3.5 STR-DPI: FAST READ

(4) STR-Quad

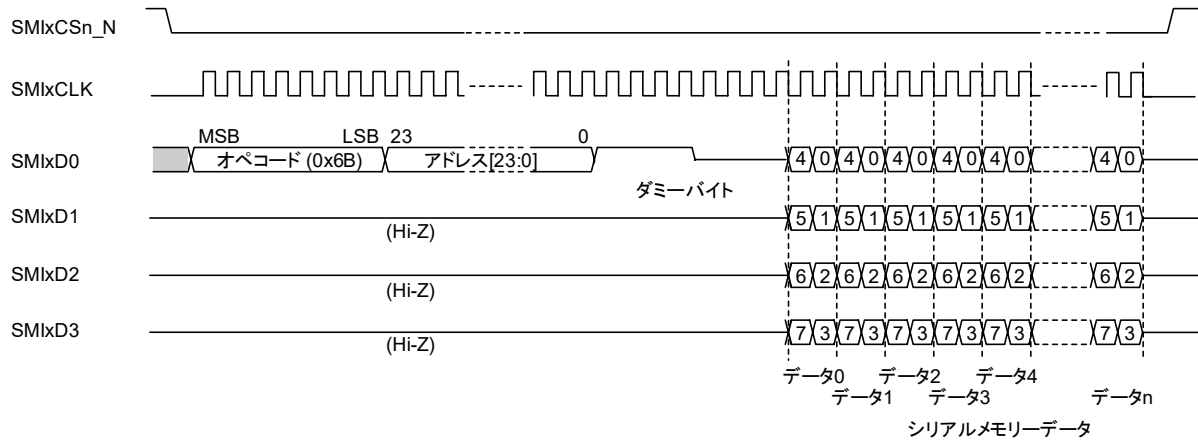


図 3.6 STR-Quad: FAST READ Quad Output

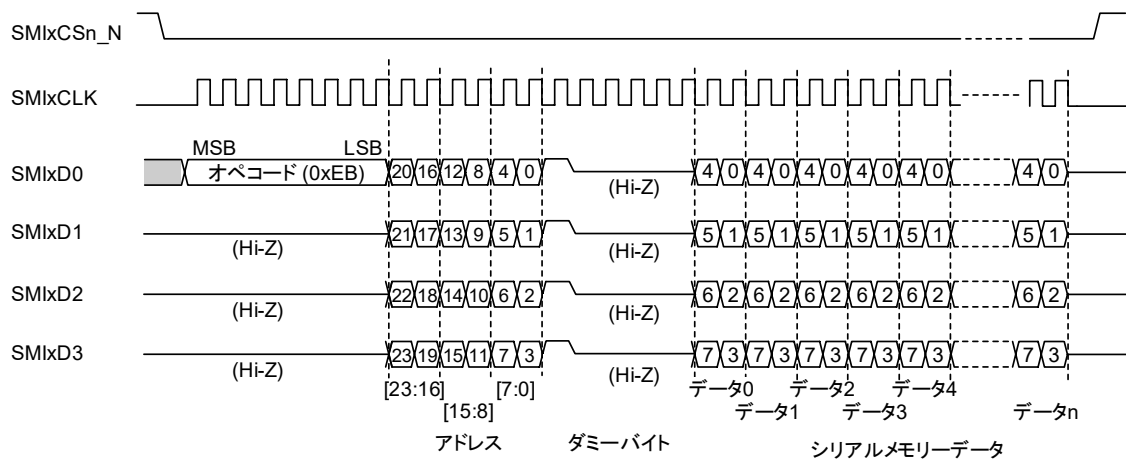


図 3.7 STR-Quad: FAST READ Quad I/O

(5) STR-QPI

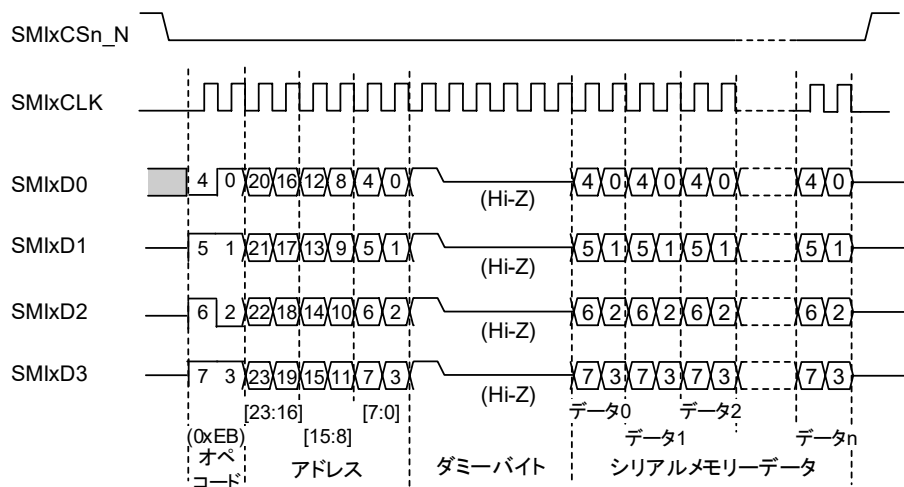


図 3.8 STR-QPI: FAST READ

(6) STR-Octal

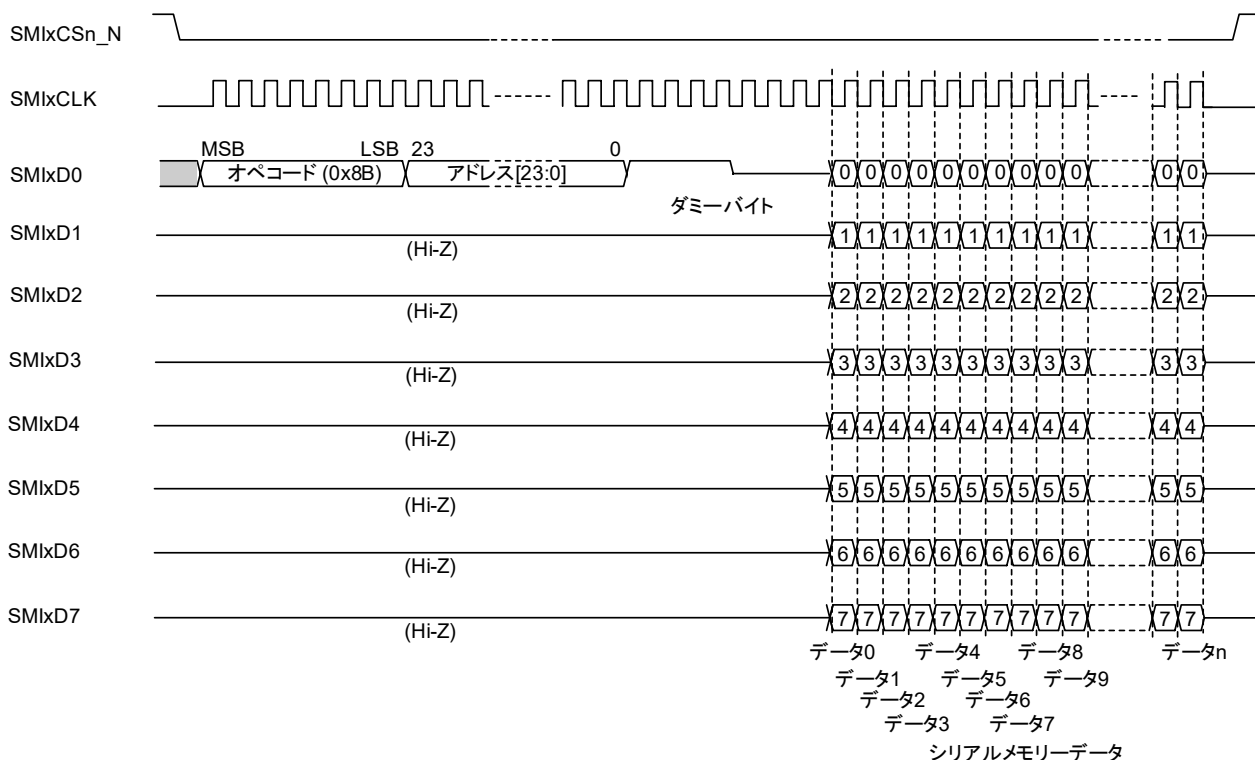


図 3.9 STR-Octal: FAST READ Octal Output

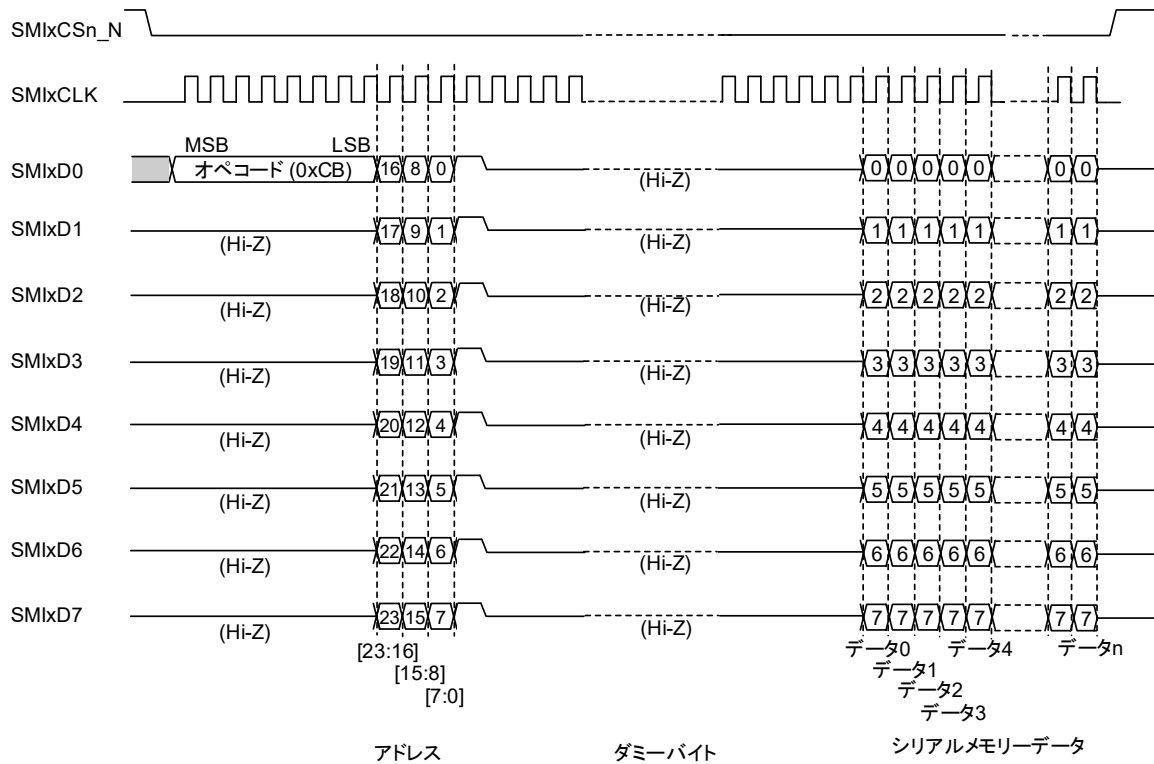


図 3.10 STR-Octal: FAST READ Octal I/O

(7) STR-OPI

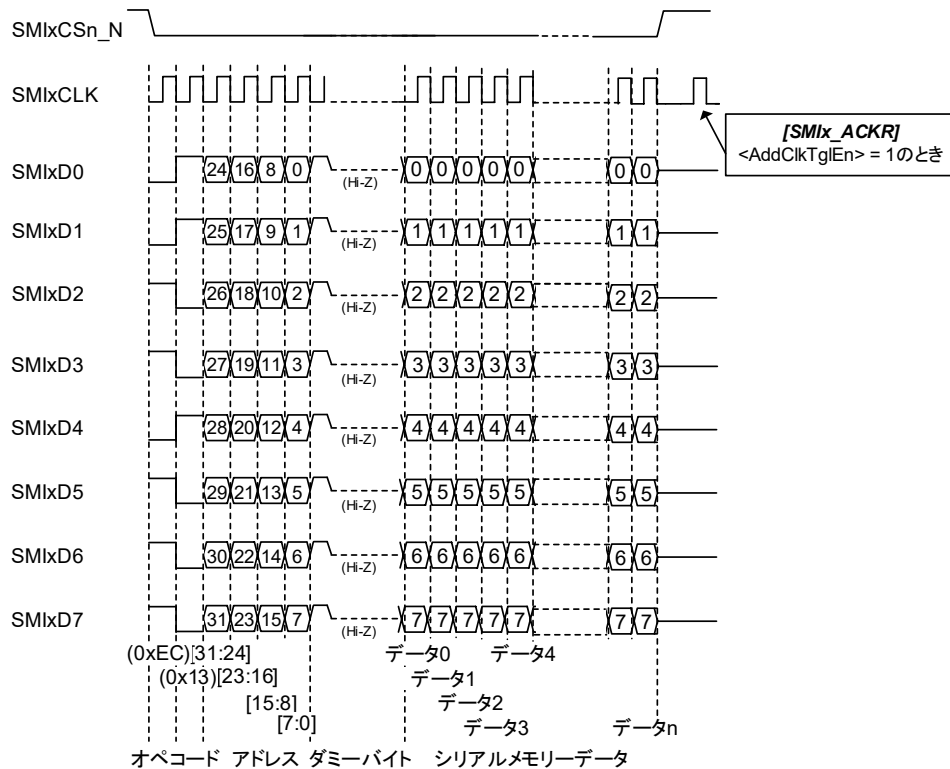


図 3.11 STR-OPI: FAST READ

3.4.1.2. WIP ビットをポーリングする機能

本機能は、主にデバッグ用途のものです。

一般的に SPI Flash は Write、Erase 中に、Read コマンドを受け付けません。もし SPI Flash が Write、Erase 中に SMIF から Read コマンドを発行した場合、SPI Flash は応答しません。結果として無効なデータが返されるようにみえます。基本的にはこのようなことが起こらないようにソフトウェアで制御する必要があります。

この機能は万が一、SPI Flash が Write、Erase 中にリードを行っても正しいデータを返すことを可能にします。この機能が許可時にリードが検出されると SMIF は、SPI Flash の Status レジスタの WIP ビットが"0"になるまでポーリングを行い、その後、Read コマンドを発行します。これで正しいデータを受け取ることを可能にしています。この動作を許可するには *[SMIxDACRn]<PollWIP>* を"1"にセットしてください。

なお、リセット直後の動作開始時には必ず SPI Flash の Status レジスタの WIP ビットをポーリングし、Write、Erase 中でないことを確認してください。その後は *[SMIxDACRn]<PollWIP>* ビットにより制御されます。

この機能を許可するとリードごとに WIP ビットをポーリングするため、リードのオーバーヘッドが大きくなります。

シリアルメモリーが接続されていない場合や、SPI RAM などの WIP ビットを持たないシリアルメモリーを接続している場合、本機能を有効にしないでください。

3.4.1.3. ダイレクトアクセスの設定手順

- (1) 実装するシリアルメモリーに合わせて、マッピングのベースアドレスと容量を *[SMIxMAP0]*、*[SMIxMAP1]* で設定します。
- (2) 転送クロック、CS デアサート時間、WIP ポーリング動作を *[SMIxDACR0]*、*[SMIxDACR1]* で設定します。
- (3) コマンドオペコード、ダミーバイト数、入出力制御を *[SMIxDRCR0]*、*[SMIxDRCR1]*、*[SMIxDWCR0]*、*[SMIxDWCR1]* で設定します。
- (4) シリアルメモリーがマッピングされているアドレスをリードライトします。

注) リセット直後、シリアルメモリー0 が、アドレス"0xA0000000" ~ "0xA0FFFFFF"の 16M バイト空間にマッピングされています。

3.4.2. インダイレクトアクセス

インダイレクトアクセスは、SPI Flash へ PAGE PROGRAM、ERASE、CHIP ERASE、READ STATUS REGISTER、READ などのコマンドをレジスターを介して発行しアクセスする方法です。SPI Flash へのコマンドはプライマリーバッファの 32 バイトとセカンダリーバッファの 256 バイトを使用し、最大 288 バイト分のコマンド発行が可能です。インダイレクトアクセスの設定は、シリアルメモリー0、シリアルメモリー1 共通で $[SMIxRACR0]$ 、 $[SMIxRACR1]$ で行います。

インダイレクトアクセスによるデータ転送シーケンス (SMIFからシリアルメモリー)

- (A) バッファからシフトレジスターへのデータ書き込み
- (B) SMIFとシリアルメモリー間でデータをシフト転送
- (C) シフトレジスターからバッファへのデータ書き込み
- (D) バッファへのアドレスをインクリメント (プライマリーバッファが終わったら、セカンダリーバッファへ)

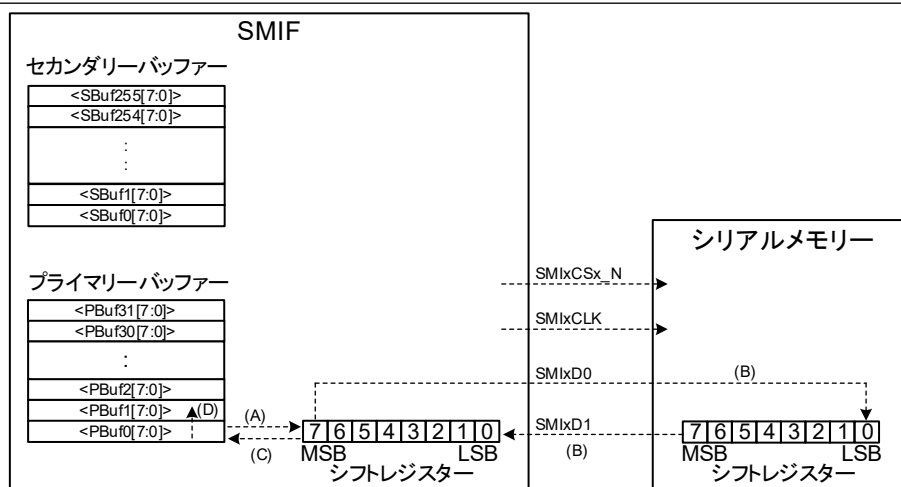


図 3.12 インダイレクトアクセス

3.4.3. インダイレクトアクセスの設定手順

- (1) $[SMIxSTAT]<CycProg>$ で、実行中のインダイレクトアクセスがないか確認します。
- (2) アクセスするシリアルメモリー、転送バイト数などを $[SMIxRACR0]$ 、 $[SMIxRACR1]$ 、 $[SMIxINT]$ に設定します。
- (3) バッファ $[SMIxPBUF_n]$ 、 $[SMIxSBUF_m]$ ($n = 0 \sim 7$ 、 $m = 0 \sim 63$) にコマンドデータを設定します。
- (4) $[SMIxRACR1]<CycGo>$ を "1" にすると順次コマンドが発行されます。

3.5. 転送クロック

転送クロック (SMIxCCLK) の周波数は、 $[SMIxDACR0]/SPR[4:0]$ 、 $[SMIxDACR1]/SPR[4:0]$ 、 $[SMIxDACR2]/SPR[4:0]$ の分周値設定で決まります。

転送クロックの周波数 = fsys 周波数 / 分周値

以下に分周値設定による転送クロックの周波数の例を示します。

表 3.1 転送クロックの周波数

<SPR[4:0]>	0	1	2	3	4	5	6	7	8	9	10	11	12	...	28	29	30	31
分周値 fsys [MHz]	2	2	3	4	5	6	7	8	9	10	11	12	13	...	29	30	31	32
200	100	100	66.6	50	40	33.3	28.5	25	22.2	20	18.1	16.6	15.3	...	6.89	6.66	6.45	6.25
160	80	80	53.3	40.0	32.0	26.7	22.9	20.0	17.8	16.0	14.5	13.3	12.3		5.52	5.33	5.16	5.00
140	70	70	46.7	35.0	28.0	23.3	20.0	17.5	15.6	14.0	12.7	11.7	10.8		4.83	4.67	4.52	4.38
120	60	60	40.0	30.0	24.0	20.0	17.1	15.0	13.3	12.0	10.9	10.0	9.2		4.14	4.00	3.87	3.75
100	50	50	33.3	25.0	20.0	16.7	14.3	12.5	11.1	10.0	9.1	8.3	7.7		3.45	3.33	3.23	3.13
80	40	40	26.7	20.0	16.0	13.3	11.4	10.0	8.9	8.0	7.3	6.7	6.2		2.76	2.67	2.58	2.50
60	30	30	20.0	15.0	12.0	10.0	8.6	7.5	6.7	6.0	5.5	5.0	4.6		2.07	2.00	1.94	1.88
40	20	20	13.3	10.0	8.0	6.7	5.7	5.0	4.4	4.0	3.6	3.3	3.1		1.38	1.33	1.29	1.25
20	10	10	6.7	5.0	4.0	3.3	2.9	2.5	2.2	2.0	1.8	1.7	1.5		0.69	0.67	0.65	0.63

注 1) 分周値が奇数の場合、転送クロックの Duty は 50%ではなく、"Low"幅 = 分周値/2 + 0.5、"High"幅 = 分周値/2 - 0.5 の比となります。

注 2) 使用できる転送クロックの周波数については、データシートの「電気的特性」を参照してください。

3.6. データ入出力タイミング

SMIxCLK の立ち下がりエッジでデータの入出力が行われます。

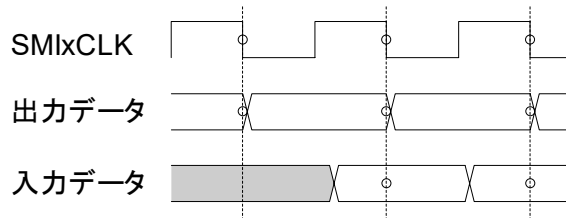


図 3.13 データ入出力タイミング

3.7. 割り込み

インダイレクトアクセスによる SPI サイクルが完了すると $[SMIxSTAT]<CycDone>$ が "1" になるとともに割り込み (INTSMIx) を発生させることができます。 $[SMIxINT]<SCDIntEn> = 1$ で割り込み発生が許可されます。発生した割り込みをクリアするには、 $[SMIxSTAT]<CycDone>$ に "0" を書き込んでください。

強制停止処理による SPI サイクルが完了すると $[SMIxSTAT]<StpProgDone>$ が "1" になるとともに割り込み (INTSMIx) を発生させることができます。 $[SMIxINT]<SDIntEn> = 1$ で割り込み発生が許可されます。発生した割り込みをクリアするには、 $[SMIxSTAT]<StpProgDone>$ に "0" を書き込んでください。

4. レジスター説明

4.1. レジスター一覧

制御レジスターとアドレスは以下のとおりです。

周辺機能		チャンネル/ユニット	ベースアドレス	
			TYPE1	TYPE2
シリアルメモリーインターフェース	SMIF	ch 0	0x4000C000	0x4003C000

注) 製品によって搭載されるベースアドレスは異なります。詳細はリファレンスマニュアル「製品個別情報」を参照してください。

レジスター名		アドレス (Base +)
アドレスマップコントロールレジスター0	[SMIXMAP0]	0x0000
アドレスマップコントロールレジスター1	[SMIXMAP1]	0x0004
ダイレクトアクセスコントロールレジスター0	[SMIXDACR0]	0x0008
ダイレクトアクセスコントロールレジスター1	[SMIXDACR1]	0x000C
ダイレクトリードコントロールレジスター0	[SMIXDRCR0]	0x0010
ダイレクトリードコントロールレジスター1	[SMIXDRCR1]	0x0014
ダイレクトライトコントロールレジスター0	[SMIXDWCR0]	0x0018
ダイレクトライトコントロールレジスター1	[SMIXDWCR1]	0x001C
インダイレクトアクセスコントロールレジスター0	[SMIXRACR0]	0x0400
インダイレクトアクセスコントロールレジスター1	[SMIXRACR1]	0x0404
インダイレクトアクセス I/O コントロールレジスター	[SMIXIOCR]	0x0408
インダイレクトアクセス Output イネーブルコントロールレジスター	[SMIXOECR]	0x040C
割り込みコントロールレジスター	[SMIXINT]	0x0440
ステータスレジスター	[SMIXSTAT]	0x0444
ソフトウェアリセットレジスター	[SMIXSWR]	0x0480
アディショナルクロック制御レジスター	[SMIXACKR]	0x0484
転送クロック/CS 端子出力制御レジスター	[SMIXCCOR]	0x0488
強制停止制御レジスター	[SMIXSTPR]	0x048C
インダイレクトアクセスプライマリーバッファレジスターn	[SMIXPBUF _n] (n = 0 ~ 7)	0x0500 ~ 0x051C
インダイレクトアクセスセカンダリーバッファレジスターm	[SMIXSBUF _m] (m = 0 ~ 63)	0x0600 ~ 0x06FC

4.2. レジスター詳細

4.2.1. [SMIxMAP0] (アドレスマップコントロールレジスター0)

Bit	Bit symbol	リセット後	Type	機能
31: 28	-	0	R	リードすると"0"が読めます。
27: 16	FBA[11: 0]	0x000	R/W	シリアルメモリー0 のマッピングベースアドレス (注 1) (注 2) 0x000 ~ 0x7FF: ベースアドレス指定 <FBA[11: 0]>がシリアルメモリー0 のベースアドレス上位桁の指定となり、下位桁は"0x0000"となります。 ("0xA0000000" ~ "0xA7FF0000")
15: 6	-	0	R	リードすると"0"が読めます。
5: 2	FDEN[3:0]	1000	R/W	シリアルメモリー0 の容量 (注 2) 0000: 64KB 0110: 4MB 0001: 128KB 0111: 8MB 0010: 256KB 1000: 16MB 0011: 512KB 1001: 32MB 0100: 1MB 1010: 64MB 0101: 2MB 1011: 128MB 上記以外: Reserved
1	WE	0	R/W	ライト制御 0: 禁止 1: 許可
0	RE	1	R/W	リード制御 0: 禁止 1: 許可

注 1) <FBA[11:0]>の値は、<FDEN[3:0]>で指定される値でアライメントされている必要があります。

注 2) シリアルメモリー0 とシリアルメモリー1 の領域はオーバーラップしないでください。

4.2.2. [SM1xMAP1] (アドレスマップコントロールレジスター1)

Bit	Bit symbol	リセット後	Type	機能
31: 28	-	0	R	リードすると"0"が読めます。
27: 16	FBA[11: 0]	0x000	R/W	シリアルメモリー1 のマッピングベースアドレス (注 1) (注 2) 0x000 ~ 0x7FF: ベースアドレス指定 <FBA[11: 0]>がシリアルメモリー1 のベースアドレス上位桁 の指定となり、下位桁は"0x0000"となります。 ("0xA0000000" ~ "0xA7FF0000")
15: 6	-	0	R	リードすると"0"が読めます。
5: 2	FDEN[3:0]	1000	R/W	シリアルメモリー1 の容量 (注 2) 0000: 64KB 0110: 4MB 0001: 128KB 0111: 8MB 0010: 256KB 1000: 16MB 0011: 512KB 1001: 32MB 0100: 1MB 1010: 64MB 0101: 2MB 1011: 128MB 上記以外: Reserved
1	WE	0	R/W	ライト制御 0: 禁止 1: 許可
0	RE	0	R/W	リード制御 0: 禁止 1: 許可

注 1) <FBA[11: 0]>の値は、<FDEN[3:0]>で指定される値でアライメントされている必要があります。

注 2) シリアルメモリー0 とシリアルメモリー1 の領域はオーバーラップしないください。

4.2.3. [SMIxDACRn] (ダイレクトアクセスコントロールレジスターn) (n = 0、1)

Bit	Bit symbol	リセット後	Type	機能
31: 21	—	0	R	リードすると"0"が読めます。
20: 16	SPR[4:0]	11111	R/W	シリアルメモリーn の転送クロック分周値 00000: 2 分周 00001: 2 分周 00010: 3 分周 00011: 4 分周 : 11110: 31 分周 11111: 32 分周
15: 8	SCSD[7:0]	0x00	R/W	シリアルメモリーn の SMIxCsn_N デアサート時間 0x00 ~ 0xFF: デアサート時間 デアサート時間 = fsys 周期 × <SCSD[7:0]>
7	—	0	R	リードすると"0"が読めます。
6	PollWIP	0	R/W	SPI Flash ヘリードアクセスを発行する際、直前で SPI Flash 内 Status レジスターの WIP が"0"になるまでポーリングします。 0: 禁止 1: 許可
5: 4	SDCE[1:0]	00	R/W	"01"をライトしてください。(注)
3	-	0	R	リードすると"0"が読めます。
2	-	0	R/W	"0"をライトしてください。
1: 0	-	0	R	リードすると"0"が読めます。

注) リセット後、<SDCE[1:0]>は"00"となりますが初期設定で"01"をライトしてください。

4.2.4. [SMIxDRCRn] (ダイレクトリードコントロールレジスターn) (n = 0, 1)

Bit	Bit symbol	リセット後	Type	機能
31: 24	CmdOp1[7:0]	0x00	R/W	シリアルメモリーn の SPI コマンドオペコード 1 SPI コマンドのオペコードを設定してください。
23: 16	CmdOp0[7:0]	0x0B	R/W	シリアルメモリーn の SPI コマンドオペコード 0 SPI コマンドのオペコードを設定してください。
15: 11	DmyBc[4:0]	00001	R/W	シリアルメモリーn の SPI ダミーバイト数 0 ~ 31 を指定可能です。
10: 9	AddrBc[1:0]	00	R/W	SPI アドレスバイトカウント 00: 3 バイト 01: 2 バイト 10: 4 バイト 11: Reserved
8	CmdBc	0	R/W	SPI コマンドバイトカウント 0: 1 バイト 1: 2 バイト
7: 6	DatIO[1:0]	00	R/W	シリアルメモリーn の SPI データ入出力制御 00: Single 10: Quad 01: Dual 11: Octal
5: 4	DmyIO[1:0]	00	R/W	シリアルメモリーn の SPI ダミー入出力制御 00: Single 10: Quad 01: Dual 11: Octal
3: 2	AdrIO[1:0]	00	R/W	シリアルメモリーn の SPI アドレス入出力制御 00: Single 10: Quad 01: Dual 11: Octal
1: 0	CmdIO[1:0]	00	R/W	シリアルメモリーn の SPI コマンド入出力制御 00: Single 10: Quad 01: Dual 11: Octal

4.2.5. [SMIxDWCRn] (ダイレクトライトコントロールレジスターn) (n = 0、1)

Bit	Bit symbol	リセット後	Type	機能
31: 24	CmdOp1[7:0]	0x00	R/W	シリアルメモリーn の SPI コマンドオペコード 1 SPI コマンドのオペコードを設定してください。
23: 16	CmdOp0[7:0]	0x02	RW	シリアルメモリーn の SPI コマンドオペコード 0 SPI コマンドのオペコードを設定してください。
15: 11	DmyBc[4:0]	00000	R/W	シリアルメモリーn の SPI ダミーバイト数 00000 を設定してください。
10: 9	AddrBc[1:0]	00	R/W	SPI アドレスバイトカウント 00: 3 バイト 01: 2 バイト 10: 4 バイト 11: Reserved
8	CmdBc	0	R/W	SPI コマンドバイトカウント 0: 1 バイト 1: 2 バイト
7: 6	DatIO[1:0]	00	R/W	シリアルメモリーn の SPI データ入出力制御 00: Single 10: Quad 01: Dual 11: Octal
5: 4	DmyIO[1:0]	00	R/W	シリアルメモリーn の SPI ダミー入出力制御 00: Single 10: Quad 01: Dual 11: Octal
3: 2	AdrIO[1:0]	00	R/W	シリアルメモリーn の SPI アドレス入出力制御 00: Single 10: Quad 01: Dual 11: Octal
1: 0	CmdIO[1:0]	00	R/W	シリアルメモリーn の SPI コマンド入出力制御 00: Single 10: Quad 01: Dual 11: Octal

4.2.6. [SMIxRACR0] (インダイレクトアクセスコントロールレジスター0)

Bit	Bit symbol	リセット後	Type	機能
31: 21	-	0	R	リードすると"0"が読めます。
20: 16	SPR[4:0]	11111	R/W	シリアルメモリーの転送クロック分周値 00000: 2 分周 00001: 2 分周 00010: 3 分周 00011: 4 分周 : 11110: 31 分周 11111: 32 分周
15: 8	SCSD[7:0]	0xFF	R/W	SMIxCSn_N デアサート時間 0x00 ~ 0xFF: デアサート時間 デアサート時間 = fsys 周期 × <SCSD[7:0]>
7: 6	-	0	R	リードすると"0"が読めます。
5: 4	SDCE[1:0]	00	R/W	"01"をライトしてください。(注)
3	-	0	R	リードすると"0"が読めます。
2	-	0	R/W	"0"をライトしてください。
1: 0	-	0	R	リードすると"0"が読めます。

注) リセット後、<SDCE[1:0]>は"00"となりますが、初期設定で"01"をライトしてください。

4.2.7. [SMIxRACR1] (インダイレクトアクセスコントロールレジスター1)

Bit	Bit symbol	リセット後	Type	機能
31: 24	SBufBc[7:0]	0x00	R/W	セカンダリーバッファの転送バイト数 0 ~ 255 を指定可能です。 指定した値プラス 1 のバイト数の転送をセカンダリーバッファデータから行います。
23: 22	-	0	R	リードすると"0"が読めます。
21	-	0	R/W	"0"をライトしてください。
20: 16	PBufBc[4:0]	00000	R/W	プライマリーバッファの転送バイト数 (注 2) 0 ~ 31 を指定可能です。 指定した値プラス 1 のバイト数の転送をプライマリーバッファデータから行います。
15: 6	-	0	R	リードすると"0"が読めます。
5	SBufEn	0	R/W	セカンダリーバッファのイネーブル制御 (注 3) 0: セカンダリーバッファを使用しない 1: セカンダリーバッファを使用する
4	PBufEn	0	R/W	プライマリーバッファのイネーブル制御 0: プライマリーバッファを使用しない 1: プライマリーバッファを使用する
3: 2	-	0	R	リードすると"0"が読めます。
1	CSNum	0	R/W	アサートする CS 0: SMiXCS0_N 1: SMiXCS1_N
0	CycGo	0	R/W	インダイレクトアクセス制御 0: don't care 1: インダイレクトアクセスを開始 (注 1) 終了後に"0"が書きこまれます。このビットは読み出すと常に"0"が読めます。

注 1) [SMIxSTAT]<CycProg> = 1 (SPI サイクル中)の間は、<CycGo>に"1"を書き込まないでください。

注 2) セカンダリーバッファを併用して Quad または Octal で転送を行う場合、以下の設定としてください。

Quad の場合: PBufBc[4:0] ≥ 1

Octal の場合: PBufBc[4:0] ≥ 2

注 3) セカンダリーバッファのみの使用はできません。プライマリーバッファと併せて使用してください。

4.2.8. [SMI_{IO}OCR] (インダイレクトアクセス I/O コントロールレジスター)

Bit	Bit symbol	リセット後	Type	機能
31: 21	-	0	R	リードすると"0"が読めます。
20: 16	PBufIOCtrl[4:0]	00000	R/W	プライマリーバッファ出力データ幅制御 0 ~ 6 を指定可能です。 指定したバイト数の転送位置からプライマリーバッファデータを<PBufIOCtrlEn[1:0]>のデータ幅で転送します。 例えば、構成されるコマンドの全てがマルチ I/O 通信となる QPI、OPI では、"00000"を指定します。
15: 4	-	0	R	リードすると"0"が読めます。
3: 2	SBufIOCtrlEn[1:0]	00	R/W	セカンダリーバッファ I/O コントロール制御 (注) 00: Single 01: Reserved 10: Quad 11: Octal
1: 0	PBufIOCtrlEn[1:0]	00	R/W	プライマリーバッファ I/O コントロール制御 00: Single 01: Reserved 10: Quad 11: Octal

注) セカンダリーバッファを使用しない場合、<PBufIOCtrlEn[1:0]>と同じ設定にしてください。

4.2.9. [SMI_{IO}ECR] (インダイレクトアクセス Output イネーブルコントロールレジスター)

Bit	Bit symbol	リセット後	Type	機能
31: 21	-	0	R	リードすると"0"が読めます。
20: 16	PBufOEnC[4:0]	00000	R/W	プライマリーバッファ出力方向制御 (注 1) 0 ~ 6 を指定可能です。 <PBufOEnC> = 1 の場合、指定したバイト数の転送位置からプライマリーバッファデータの送受信方向を切り替えます。 例えば、コマンドオペコードバイト数が 1、アドレスバイト数が 3 の FAST READ の場合は、4 を指定します。こうすることで、I/O 方向をダミー通信期間に入力方向に切り替え、ターンアラウンドを実現します。
15: 1	-	0	R	リードすると"0"が読めます。
0	PBufOEn	0	R/W	プライマリーバッファ出力制御 (注 2) 0: 禁止 1: 許可 I/O の方向を出力から入力に切り替える機能の制御をします。

注 1) [SMI_{IO}OCR]<PBufIOCtrl[4:0]>以上の値を設定してください。

注 2) Single で使用する場合、"0"を設定してください。

4.2.10. [SMIxINT] (割り込みコントロールレジスター)

Bit	Bit symbol	リセット後	Type	機能
31: 5	-	0	R	リードすると"0"が読めます。
4	SDIntEn	0	R/W	強制停止処理完了割り込み許可 0: 禁止 1: 許可 強制停止処理の完了割り込み通知を制御します。
3: 1	-	0	R	リードすると"0"が読めます。
0	SCDIntEn	0	R/W	インダイレクトアクセス完了割り込み許可 0: 禁止 1: 許可 インダイレクトアクセスの完了割り込み通知を制御します。

4.2.11. [SMIxSTAT] (ステータスレジスター)

Bit	Bit symbol	リセット後	Type	機能
31: 7	-	0	R	リードすると"0"が読めます。
6	DiAccInProg	0	R	ダイレクトアクセス状態 0: ダイレクトアクセスは発生していない 1: ダイレクトアクセスは発生している ダイレクトアクセスが発生しているかを示します。
5		0	R	リードすると"0"が読めます。
4	StpProgDone	0	R	強制停止状態 0: 強制停止処理は完了していない、あるいは、発生していない 1: 強制停止処理は完了している 強制停止処理の完了状態を示します。 ハードウェアがこのビットを"1"にセットします。ソフトウェアでクリアされるまで"1"のままです。このビットがセットされ、 [SMIxINT]<SDIntEn> が"1"の場合、割り込みがアサートされます。
			W	強制停止割り込みをクリア 0: クリア 1: don't care
3: 2	-	0	R	リードすると"0"が読めます。
1	CycProg	0	R	SPI サイクル状態 0: インダイレクトアクセスは発生していない 1: インダイレクトアクセスは発生している このビットは<CycGo>をセット後に"1"にセットされます。 SPI サイクル完了時に"0"にリセットされます。
0	CycDone	0	R	SPI サイクル完了状態 0: インダイレクトアクセスは完了していない、あるいは発生していない 1: インダイレクトアクセスは完了している SPI サイクル完了後、ハードウェアがこのビットを"1"にセットします。 ソフトウェアでクリアされるまで"1"のままです。このビットがセットされ [SMIxINT]<SCDIntEn> が"1"の場合、割り込みがアサートされます。
			W	SPI サイクル完了割り込みをクリア 0: クリア 1: don't care

4.2.12. [SMIxSWR] (ソフトウェアリセットレジスター)

Bit	Bit symbol	リセット後	Type	機能
31: 1	-	0	R	リードすると"0"が読めます。
0	RstEn	0	RW	リセットイネーブル 0: ソフトウェアリセット完了した、または未発行 1: ソフトウェアリセット実施中 SMIF へのソフトウェアリセットを制御します。 本フィールドに"1"をライトすると、SMIF は直ちにリセット処理を実施します。その後、リセットが完了すると本フィールドは"0"にクリアされます (注)。

注) ソフトウェアリセット実施後は、<RstEn>"0"が読み出せるまでポーリングしてください。

4.2.13. [SMIxACKR] (アディショナルクロック制御レジスター)

Bit	Bit symbol	リセット後	Type	機能
31: 9	-	0	R	リードすると"0"が読めます。
8	AddClkTglEn	0	RW	アディショナルクロックトグルイネーブル 0: 出力しない 1: 出力する SPI 転送が完了し SMiXCsn_N が"High"になる際、転送クロックを追加で 1 パルス出力させるかを制御します。 <FrcClkOutEn>の設定が優先されます。
7: 1	-	0	R	リードすると"0"が読めます。
0	FrcClkOutEn	0	RW	SMiXCLK アウトプットイネーブル 0: SMiXCLK は、SPI 通信発生時のみ出力される 1: SMiXCLK は、SPI 転送がなくても出力される [SMiXCCOR]<ClkOE> = 1 の場合、SMiXCLK の出力を実施します。

4.2.14. [SMIxCCOR] (転送クロック/CS 端子出力制御レジスター)

Bit	Bit symbol	リセット後	Type	機能
31: 6	-	0	R	リードすると"0"が読めます。
5	Cs1OE	0	RW	チップセレクト 1 出力イネーブル 0: ディセーブル (Hi-Z) 1: イネーブル SMIxCS1_N の出力を制御します。
4	Cs0OE	0	RW	チップセレクト 0 出力イネーブル 0: ディセーブル (Hi-Z) 1: イネーブル SMIxCS0_N の出力を制御します。
3: 1	-	0	R	リードすると"0"が読めます。
0	ClkOE	0	RW	クロック出力イネーブル 0: ディセーブル (Hi-Z) 1: イネーブル SMIxCLK の出力を制御します。

注) ポートの設定も行ってください。詳細は、リファレンスマニュアル「入出力ポート」を参照してください。

4.2.15. [SMIxSTPR] (強制停止制御レジスター)

Bit	Bit symbol	リセット後	Type	機能
31: 1	-	0	R	リードすると"0"が読めます。
0	StopEn	0	RW	強制停止イネーブル 0: リクエストは未発行 1: リクエストは発行済 SMIF の動作を強制停止 (注)するリクエストを発行します。 本フィールドに"1"をライトした場合、SMIF は直ちに停止処理を実施します。復帰するには、ソフトウェアリセットする必要があります。 "0"ライトは無視されます。

注) シリアルメモリーとの通信が発生中の場合、SMIF はチップセレクト信号を"High"にし通信を中断します。強制停止を行った場合、データの正当性は保証されません。

4.2.16. [SMIxPBUFn] (インダイレクトアクセスプライマリーバッファレジスターn) (n = 0 ~ 7)

Bit	Bit symbol	リセット後	Type	機能
31: 24	PBuf (4n + 3)[7:0]	不定	R/W	プライマリーバッファ
23: 16	PBuf (4n + 2)[7:0]	不定	R/W	プライマリーバッファ
15: 8	PBuf (4n + 1)[7:0]	不定	R/W	プライマリーバッファ
7: 0	PBuf (4n + 0)[7:0]	不定	R/W	プライマリーバッファ

注) 製品によって、プライマリーバッファの容量は異なります。詳細はリファレンスマニュアル「製品個別情報」を参照してください。

4.2.17. [SMIxSBUFm] (インダイレクトアクセスセカンダリーバッファレジスターm) (m = 0 ~ 63)

Bit	Bit symbol	リセット後	Type	機能
31: 24	SBuf (4m + 3)[7:0]	不定	R/W	セカンダリーバッファ
23: 16	SBuf (4m + 2)[7:0]	不定	R/W	セカンダリーバッファ
15: 8	SBuf (4m + 1)[7:0]	不定	R/W	セカンダリーバッファ
7: 0	SBuf (4m + 0)[7:0]	不定	R/W	セカンダリーバッファ

注) 製品によって、セカンダリーバッファの容量は異なります。詳細はリファレンスマニュアル「製品個別情報」を参照してください。

5. 使用方法の例

5.1. 初期設定

ソフトウェアは、SMIFのリセットが解除された後に、SMIFに初期設定をする必要があります。ここでは、以下について説明します。

SMIFを使用するためには、SMIFにクロックを供給しなくてはなりません。その手順は、リファレンスマニュアル「クロック制御と動作モード」を参照してください。次に、SMIFが製品のI/O端子を使用できるように、I/O端子機能の割り当てをする必要があります。その手順は、リファレンスマニュアル「入出力ポート」を参照してください。

以上の設定が終わった後は、SMIFのレジスターを使用して初期設定を行います。初期設定の大枠は、以下のとおりです。

- ダイレクトアクセスとインダイレクトアクセスの共通設定
- ダイレクトアクセスシリアルメモリー0設定
- ダイレクトアクセスシリアルメモリー1設定
- インダイレクトアクセス設定

5.1.1. ダイレクトアクセスとインダイレクトアクセスの共通設定

以下のレジスターは、ダイレクトアクセスとインダイレクトアクセスの両方に関係するレジスターです。

- **[SMIxINT]**
 - － 強制停止やインダイレクトアクセス完了割り込みの発行を制御します。
 - － SMIFの割り込みは、極性"High"のレベル出力です。したがって、割り込みを使用する場合は、そのクリアはソフトウェアで実行する必要があります。
- **[SMIxACKR]**
 - － <AddClkTglEn>
シリアルメモリーが、各SPI通信の完了を確定するためにチップセレクトのデアサート後にSMIxCLKの出力が必要な場合は"1"を設定します。通常このフィールドは"0"に設定すればよいですが、一部の特殊なデバイスでは"1"に設定する必要があります。シリアルメモリーの仕様に合わせて、本フィールドを設定してください。
 - － <FrcClkOutEn>
シリアルメモリーのリセット解除の確定に、SMIxCLKの出力が必要な場合に使用します。シリアルメモリーのリセット解除シーケンスに合わせて、このbitを"1"に設定し、ウェイト期間後、再度"0"に設定してください。ここでウェイト期間は、シリアルメモリーの仕様に依存します。なお、このbitを"1"にしたままSPI通信を行ってはいけません。
- **[SMIxCCOR]**
 - － SMIxCLK、SMIxCS0_N、SMIxCS1_NのI/O方向を制御します。使用する端子は、出力イネーブルに設定してください。ポート部の制御とは異なります。両方の制御が必要なことに注意してください。

5.1.2. ダイレクトアクセスシリアルメモリー0 設定

ダイレクトアクセスシリアルメモリー0に関する初期状態を示します。

- メモリーマップ
リファレンスマニュアル「クロック制御と動作モード」のメモリーマップを参照してください。SMIF
としては、128MB 領域を備えます
- リードコマンド発行機能
イネーブル
- ライトコマンド発行機能
ディセーブル
- リードコマンド
オペコードバイト数: 1、オペコード: 0x0B、アドレスバイト数: 3、ダミーバイト数: 1
- ライトコマンド
オペコードバイト数: 1、オペコード: 0x02、アドレスバイト数: 3、ダミーバイト数: 0
- チップセレクトデアサートタイム
0 ns
- 転送クロック
fsys の 32 分周

SMIxCS0_Nに接続されたシリアルメモリーに応じて、設定の変更をします。以下の手順を実施してください。

- (1) メモリーマップをシリアルメモリーの容量に合わせて変更します。*[SMIxMAP0]*<FBA[11:0]>、<FDEN[3:0]>で設定します。
- (2) シリアルメモリーの AC 特性に合わせて、転送クロック、チップセレクトデアサートタイムを変更します。*[SMIxDACR0]*<SPR[4:0]>、<SCSD[7:0]>で設定します。
- (3) 必要に応じて、シリアルメモリーに発行するコマンド構成を変更します (例: 4×I/O や 8×I/O のマルチ I/O を利用する場合)。*[SMIxDRCR0]*で設定します。
- (4) シリアルメモリーにライトコマンドを発行したい場合、発行するコマンド構成を*[SMIxDWCR0]*で設定します。その後、*[SMIxMAP0]*<WE>を"1"にセットします。

5.1.3. ダイレクトアクセスシリアルメモリー1 設定

ダイレクトアクセスシリアルメモリー1の機能は、リセット直後はディセーブルです。

使用する場合、シリアルメモリーの仕様に合わせて、*[SMIxMAP1]*<FBA[11:0]>、<FDEN[3:0]>、*[SMIxDACR1]*<SPR[4:0]>、<SCSD[7:0]>、<SDCE[1:0]>、*[SMIxDRCR1]*、*[SMIxDWCR1]*を設定する必要があります。これらの設定を終えた後に、*[SMIxMAP1]*<RE>、<WE>に"1"をセットして、リードやライトコマンドの発行を有効にしてください。

5.1.4. インダイレクトアクセス設定

インダイレクトアクセス設定には、コマンドを発行するチップセレクトの選択も含まれます。インダイレクトアクセスでは、転送を実施するたびに異なる設定をすることが可能です。

転送クロックの分周値、チップセレクトデアサートタイムは、シリアルメモリーに応じた設定が必要です。

それぞれ、 $[SMIxRACR0]<SPR[4:0]>$ 、 $<SCSD[7:0]>$ 、 $<SDCE[1:0]>$ で設定します。インダイレクトアクセス完了時に割り込みを発生するかどうかを設定するには、 $[SMIxINT]<SCDIntEn>$ を使用します。

5.1.5. 転送クロック

転送クロック (SMiCLK) の周波数は、 $[SMIxDACR0]<SPR[4:0]>$ 、 $[SMIxDACR1]<SPR[4:0]>$ 、 $[SMIxRACR0]<SPR[4:0]>$ の分周値設定で決まります。

転送クロックの周波数 = fsys 周波数 / 分周値

注 1) SMiCLK 周波数は、データシートの「電気的特性」を満たす範囲で設定してください。

注 2) 分周値が奇数の場合、転送クロックの Duty は 50%ではなく、"Low"幅 = 分周値/2 + 0.5、"High"幅 = 分周値/2-0.5 の比となります。

5.2. ダイレクトアクセスに関するプログラミング方法

ソフトウェアは、シリアルメモリーと通信し続ける限り、ダイレクトアクセスの設定を変更する必要はありません。

SMIF は、ダイレクトアクセス領域にリード、またはライトアクセスを検知すると、以下のレジスター設定に基づき、シリアルメモリーにコマンドを発行します。設定の詳細は、「5.1.2. ダイレクトアクセスシリアルメモリー0 設定」と「5.1.3. ダイレクトアクセスシリアルメモリー1 設定」参照してください。

- $[SMIxMAP0]$
- $[SMIxMAP1]$
- $[SMIxDACR0]$
- $[SMIxDACR1]$
- $[SMIxDRCR0]$
- $[SMIxDRCR1]$
- $[SMIxDWCR0]$
- $[SMIxDWCR1]$

$[SMIxDACR0]<SPR[4:0]>$ 、 $[SMIxDACR1]<SPR[4:0]>$ を変更する場合は、設定変更後に、対応するダイレクトアクセス領域をダミーリード (リードの返り値を使用しないリード)をしてください。このダミーリードは、初期設定時には不要です。

注) ダイレクトアクセスの設定を変更する場合は、対象となるダイレクトアクセス領域での命令実行はできません。

5.3. インダイレクトアクセスに関するプログラミング方法

5.3.1. 基本手順

インダイレクトアクセスのプログラムは以下の手順で行います。

- (1) インダイレクトアクセスの完了を割り込みで待つ場合は、**[SMIxINT]/<SCDIntEn>**に"1"をライトします。
- (2) 実行中のインダイレクトアクセスがないことを**[SMIxSTAT]/<CycPrgrs>**でチェックします。
- (3) 必要に応じて、転送クロックの分周値、チップセレクトデアサートタイムの設定を**[SMIxRACR0]/<SPR[4:0]>**、**<SCSD[7:0]>**で変更します。
- (4) シリアルメモリーに発行するコマンド構成を変更します。
- (5) **[SMIxRACR1]/<CycGo>**に"1"をセットし、インダイレクトアクセスを開始します (注 1)。
- (6) インダイレクトアクセスの完了を待ちます (インダイレクトアクセス完了の割り込みを待つか、**[SMIxSTAT]/<CycDone>**に"1"がセットされるのを待ちます (注 2))。
- (7) **[SMIxSTAT]/<CycDone>**に"0"をライトします。

注 1) **[SMIxRACR1]/<CycGo>**への"1"ライトと、他のフィールドの設定値を同時に変更してもかまいません。

注 2) **[SMIxRACR1]/<CycGo>**に"1"をライトしてから、**[SMIxSTAT]/<CycDone>**に"1"がセットされるまでは、以下のレジスターに対するライトは禁止です。

- **[SMIxRACR0]**
- **[SMIxRACR1]**
- **[SMIxIOCR]**
- **[SMIxOECR]**
- **[SMIxPBUFn]**
- **[SMIxSBUFm]**

対応デバイスに対するコマンド構成は、次のように設定します。

シリアルメモリーとの通信に使用する I/O のデータ幅や転送方向は、*[SMLxIOCR]*、*[SMLxOECR]*を用いて、以下のように設定します。

(1) *[SMLxIOCR]*

- (a) <PBufIOCtrlEn[1:0]>
SPI 通信ならば"00"、マルチ I/O 通信ならば"10" (4 × I/O) or "11" (8 × I/O)を設定します。
- (b) <PBufIOCtrl[4:0]>
<PBufIOCtrlEn>が SPI 通信設定の時は、本フィールドの設定は無効です。マルチ I/O 通信設定時は、I/O 幅が切り替わるバイト番目を指定します。例えば、コマンド全体がマルチ I/O 通信となる QPI、OPI 通信では、本フィールドに"00000"を設定します。

(2) *[SMLxOECR]*

- (a) <PBufOEn>
送信ならば"0"、受信で SPI 通信ならば"0"、受信でマルチ I/O 通信ならば"1"をセットします。
- (b) <PBufOEnC[4:0]>
送信、または<PBufOEn>が SPI 通信設定の時は、本フィールドの設定は無効です。受信のマルチ I/O 通信ならば、I/O 方向が OUT から IN に切替わるバイト番目を指定します。例えば、コマンドオペコードバイト数が"1"、アドレスバイト数が"3"の FAST READ の場合は、"00100"を指定します。
- － 転送に使用するコマンドオペコード、アドレスを*[SMLxPBUFm]*にセットします。
 - － 転送方向が送信の場合、*[SMLxSBUFm]*に送信データをセットします。
 - － 転送に使用するプライマリー、セカンダリーバッファのバイト数を*[SMLxRACR1]*<SBufBc[7:0]>、<PBufBc[4:0]>、<SBufEn>、<PBufEn>で設定します (注)。
 - － 通信先のデバイスが接続されているチップセレクト番号を*[SMLxRACR1]*<CSNum>に設定します (注)。

注) *[SMLxRACR1]*<CycGo>への"1"ライトと、他のフィールドの設定値を同時に変更してもかまいません。

5.3.2. プログラミング例: PAGE PROGRAM、WRITE

256 バイトのデータ(Data[2047: 0])を SPI Flash の 3 バイト物理アドレス(アドレス[23: 0])に送信する手順を示します。

注) SPI Flash は、本コマンドを実行後に BUSY 状態になります。BUSY 状態のデバイスは、ステータスレジスタリードなどの一部コマンドにしか正常な通信をしません。SPI Flash に命令を置いている場合などは、BUSY 状態が解除されるまで、SPI Flash に対する命令フェッチが起こらないようにしてください。BUSY 中にシリアルメモリーへメモリーリードアクセスをした場合、SPI Flash からは正常なデータが読みだせません。

コマンドの構成は、「コマンドオペコードバイト数: 1、アドレスバイト数: 3、ダミーバイト数: 0、ペイロードデータバイト数: 256」とします。コマンドの I/O 幅は、全て SPI とします。コマンドオペコードは、"0x02"とします。

(1) プライマリバッファに以下の値を設定します。

- <PBuf0[7:0]> = オペコード (0x02)
- <PBuf1[7:0]> = アドレス[23: 16] (Addr[23:16])
- <PBuf2[7:0]> = アドレス[15: 8] (Addr[15:8])
- <PBuf3[7:0]> = アドレス[7:0] (Addr[7:0])

(2) セカンダリバッファに以下の値を設定します。

- <SBuf0> = データ 0 (Data[7:0]、アドレス[23: 0] + 0x00 に書かれるデータ)
- <SBuf1> = データ 1 (Data[15: 8]、アドレス[23: 0] + 0x01 に書かれるデータ)
- :
- <SBuf254> = データ 254 (Data[2039: 2032]、アドレス[23: 0] + 0xFE に書かれるデータ)
- <SBuf255> = データ 255 (Data[2047: 2040]、アドレス[23: 0] + 0xFF に書かれるデータ)

(3) *[SMIxRACR1]* の設定

- <PBufEn> = 1、<PBufBc[4:0]> = 00011
- <SBufEn> = 1、<SBufBc[7:0]> = 0xFF
- <CSNum> = SPI Flash の接続されているチップセレクト端子番号

(4) *[SMIxIOCR]* の設定

- <PBufIOCtrlEn[1:0]> = 00 (SPI 通信)
- <SBufIOCtrlEn[1:0]> = 00 (SPI 通信)
- <PBufIOCtrl[4:0]> = don't care

(5) *[SMIxOECR]* の設定

- <PBufOCEn> = 0 (送信)
- <PBufOEnC[4:0]> = don't care

(6) *[SMIxRACR1]*<CycGo>に"1"をセットし、インダイレクトアクセスを開始します。

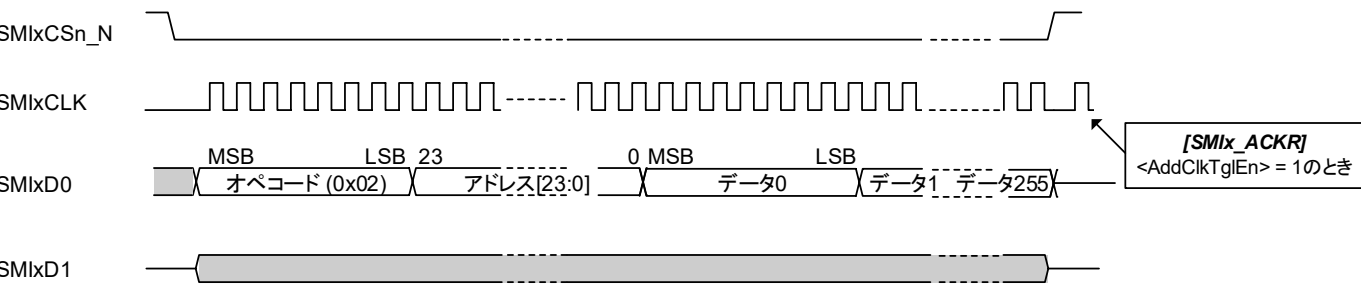


図 5.1 PAGE PROGRAM、WRITE

5.3.3. プログラミング例: Quad I/O PAGE PROGRAM、Quad I/O WRITE

256 バイトのデータ(Data[2047: 0])を SPI Flash の 3 バイト物理アドレス(Addr[23: 0])に 4 × I/O で送信する手順を示します。

注) SPI Flash は、本コマンドを実行後に BUSY 状態になります。BUSY 状態のデバイスは、ステータスレジスタリードなどの一部コマンドにしか正常な通信をしません。SPI Flash に命令を置いている場合などは、BUSY 状態が解除されるまで、SPI Flash に対する命令フェッチが起こらないようにしてください。BUSY 中に SPI Flash へメモリーリードアクセスをした場合、SPI Flash からは正常なデータが読みだせません。

コマンドの構成は、「コマンドオペコードバイト数: 1、アドレスバイト数: 3、ダミーバイト数: 0、ペイロードデータバイト数: 256」とします。コマンドの I/O 幅は、コマンドオペコードが SPI、アドレス以降は 4 × I/O とします。コマンドオペコードは、"0x38"とします。

(1) プライマリバッファに以下の値を設定します。

- <PBuf0[7:0]> = オペコード (0x38)
- <PBuf1[7:0]> = アドレス[23: 16] (Addr[23: 16])
- <PBuf2[7:0]> = アドレス[15: 8] (Addr[15: 8])
- <PBuf3[7:0]> = アドレス[7: 0] (Addr[7:0])

(2) セカンダリバッファに以下の値を設定します。

- <SBuf0[7:0]> = データ 0 (Data[7:0]、アドレス[23: 0] + 0x00 に書かれるデータ)
- <SBuf1[7:0]> = データ 1 (Data[15: 8]、アドレス[23: 0] + 0x01 に書かれるデータ)
- :
- <SBuf254[7:0]> = データ 254 (Data[2039: 2032]、アドレス[23: 0] + 0xFE に書かれるデータ)
- <SBuf255[7:0]> = データ 255 (Data[2047: 2040]、アドレス[23: 0] + 0xFF に書かれるデータ)

(3) *[SMIxRACR1]*の設定

- <PBufEn> = 1、<PBufBc[4:0]> = 00011
- <SBufEn> = 1、<SBufBc[7:0]> = 0xFF
- <CSNum> = SPI Flash の接続されているチップセレクト番号

(4) *[SMIxIOCR]*の設定

- <PBufIOCtrlEn[1:0]> = 10 (4 × I/O 通信)
- <SBufIOCtrlEn[1:0]> = 10 (4 × I/O 通信)
- <PBufIOCtrl[4:0]> = 00001 (コマンドオペコードはシングル、アドレスからマルチ I/O 通信)

(5) *[SMIxOECR]*の設定

- <PBufOCEn> = 0 (送信)
- <PBufOEnC[4:0]> = don't care

(6) *[SMIxRACR1]*<CycGo>に"1"をセットし、インダイレクトアクセスを開始します。

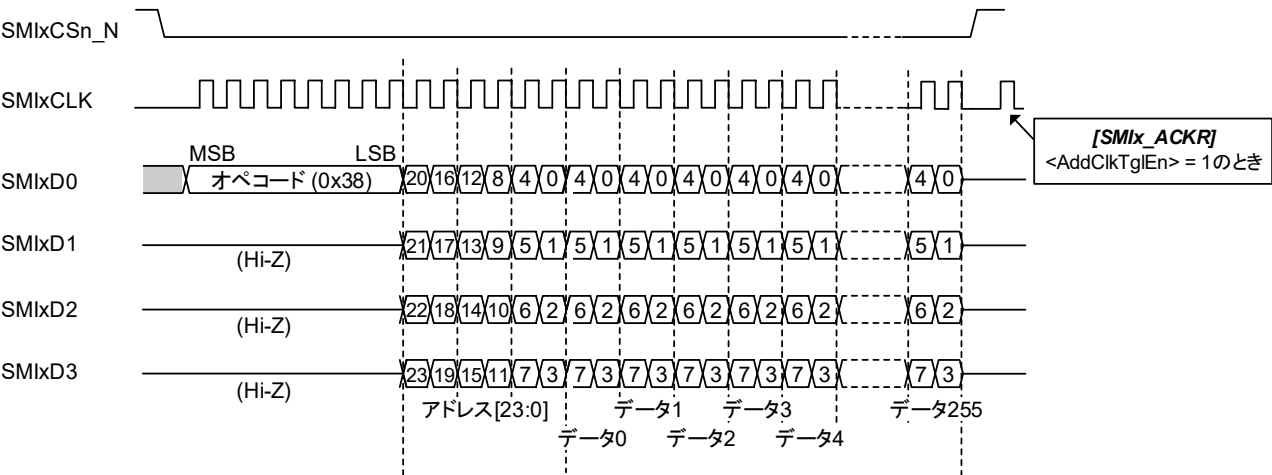


図 5.2 Quad I/O PAGE PROGRAM、Quad I/O WRITE

5.3.4. プログラミング例: QPI PAGE PROGRAM、QPI WRITE

256 バイトのデータ(Data[2047: 0])を SPI Flash の 3 バイト物理アドレス(Addr[23: 0])に QPI で送信する手順を示します。

注) SPI Flash は、本コマンドを実行後に BUSY 状態になります。BUSY 状態のデバイスは、ステータスレジスタリードなどの一部コマンドにしか正常な通信をしません。SPI Flash に命令を置いている場合などは、BUSY 状態が解除されるまで、SPI Flash に対する命令フェッチが起こらないようにしてください。BUSY 中に SPI Flash へメモリーリードアクセスをした場合、SPI Flash からは正常なデータが読みだせません。

コマンドの構成は、「コマンドオペコードバイト数: 1、アドレスバイト数: 3、ダミーバイト数: 0、ペイロードデータバイト数: 256」とします。コマンドの I/O 幅は、全て 4 × I/O とします。コマンドオペコードは、"0x02"とします。

(1) プライマリバッファに以下の値を設定します。

- <PBuf0[7:0]> = オペコード (0x02)
- <PBuf1[7:0]> = アドレス[23: 16] (Addr[23:16])
- <PBuf2[7:0]> = アドレス[15: 8] (Addr[15:8])
- <PBuf3[7:0]> = アドレス[7:0] (Addr[7:0])

(2) セカンダリバッファに以下の値を設定します。

- <SBuf0[7:0]> = データ 0 (Data[7:0]、アドレス[23: 0] + 0x00 に書かれるデータ)
- <SBuf1[7:0]> = データ 1 (Data[15: 8]、アドレス[23: 0] + 0x01 に書かれるデータ)
- :
- <SBuf254[7:0]> = データ 254 (Data[2039: 2032]、アドレス[23: 0] + 0xFE に書かれるデータ)
- <SBuf255[7:0]> = データ 255 (Data[2047: 2040]、アドレス[23: 0] + 0xFF に書かれるデータ)

(3) *[SMI_xRACR1]*の設定

- <PBufEn> = 1、<PBufBc[4:0]> = 00011
- <SBufEn> = 1、<SBufBc[7:0]> = 0xFF
- <CSNum> = SPI Flash の接続されているチップセレクト番号

(4) *[SMI_xIOCR]*の設定

- <PBufIOCtrlEn[1:0]> = 10 (4 × I/O 通信)
- <SBufIOCtrlEn[1:0]> = 10 (4 × I/O 通信)
- <PBufIOCtrl[4:0]> = 00000 (マルチ I/O 通信)

(5) *[SMI_xOECR]*の設定

- <PBufOCEn> = 0 (送信)
- <PBufOEnC[4:0]> = don't care

(6) *[SMI_xRACR1]*<CycGo>に"1"をセットし、インダイレクトアクセスを開始します。

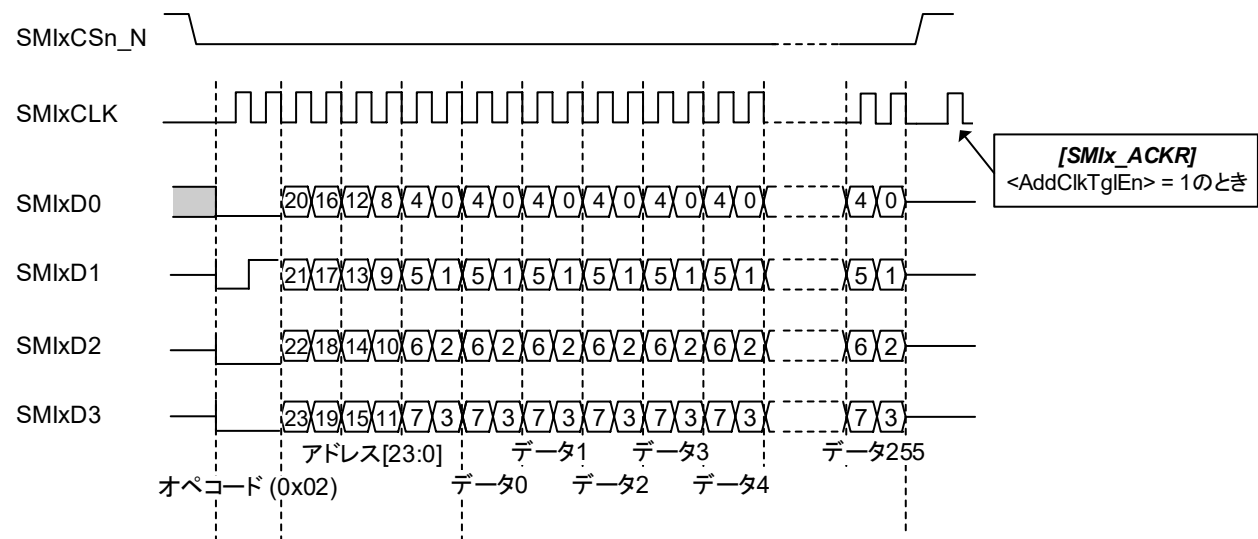


図 5.3 QPI PAGE PROGRAM、QPI WRITE

5.3.5. プログラミング例: Octal I/O PAGE PROGRAM、Octal I/O WRITE

256 バイトデータ(Data[2047: 0])を SPI Flash の 3 バイト物理アドレス(Addr[23: 0])に 8 × I/O で送信する手順を示します。

注) SPI Flash は、本コマンドを実行後に BUSY 状態になります。BUSY 状態のデバイスは、ステータスレジスターリードなどの一部コマンドにしか正常な通信をしません。SPI Flash に命令を置いている場合などは、BUSY 状態が解除されるまで、SPI Flash に対する命令フェッチが起こらないようにしてください。BUSY 中に SPI Flash へメモリーリードアクセスをした場合、SPI Flash からは正常なデータが読みだせません。

コマンドの構成は、「コマンドオペコードバイト数: 1、アドレスバイト数: 3、ダミーバイト数: 0、ペイロードデータバイト数: 256」とします。コマンドの I/O 幅は、コマンドオペコードが SPI、アドレス以降は 8 × I/O とします。コマンドオペコードは、"0xC2"とします。

(1) プライマリバッファに以下の値を設定します。

- <PBuf0[7:0]> = オペコード (0xC2)
- <PBuf1[7:0]> = アドレス[23: 16] (Addr[23:16])
- <PBuf2[7:0]> = アドレス[15: 8] (Addr[15:8])
- <PBuf3[7:0]> = アドレス[7:0] (Addr[7:0])

(2) セカンダリバッファに以下の値を設定します。

- <SBuf0[7:0]> = データ 0 (Data[7:0]、アドレス[23: 0] + 0x00 に書かれるデータ)
- <SBuf1[7:0]> = データ 1 (Data[15: 8]、アドレス[23: 0] + 0x01 に書かれるデータ)
- :
- <SBuf254[7:0]> = データ 254 (Data[2039: 2032]、アドレス[23: 0] + 0xFE に書かれるデータ)
- <SBuf255[7:0]> = データ 255 (Data[2047: 2040]、アドレス[23: 0] + 0xFF に書かれるデータ)

(3) *[SMIxRACR1]*の設定

- <PBufEn> = 1、<PBufBc[4:0]> = 00011
- <SBufEn> = 1、<SBufBc[7:0]> = 0xFF
- <CSNum> = SPI Flash の接続されているチップセレクト番号

(4) *[SMIxIOCR]*の設定

- <PBufIOCtrlEn[1:0]> = 11 (8 × I/O 通信)
- <SBufIOCtrlEn[1:0]> = 11 (8 × I/O 通信)
- <PBufIOCtrl[4:0]> = 00001 (コマンドオペコードはシングル、アドレスからマルチ I/O 通信)

(5) *[SMIxOECR]*の設定

- <PBufOCEn> = 0 (送信)
- <PBufOEnC[4:0]> = don't care

(6) *[SMIxRACR1]*<CycGo>に"1"をセットし、インダイレクトアクセスを開始します。

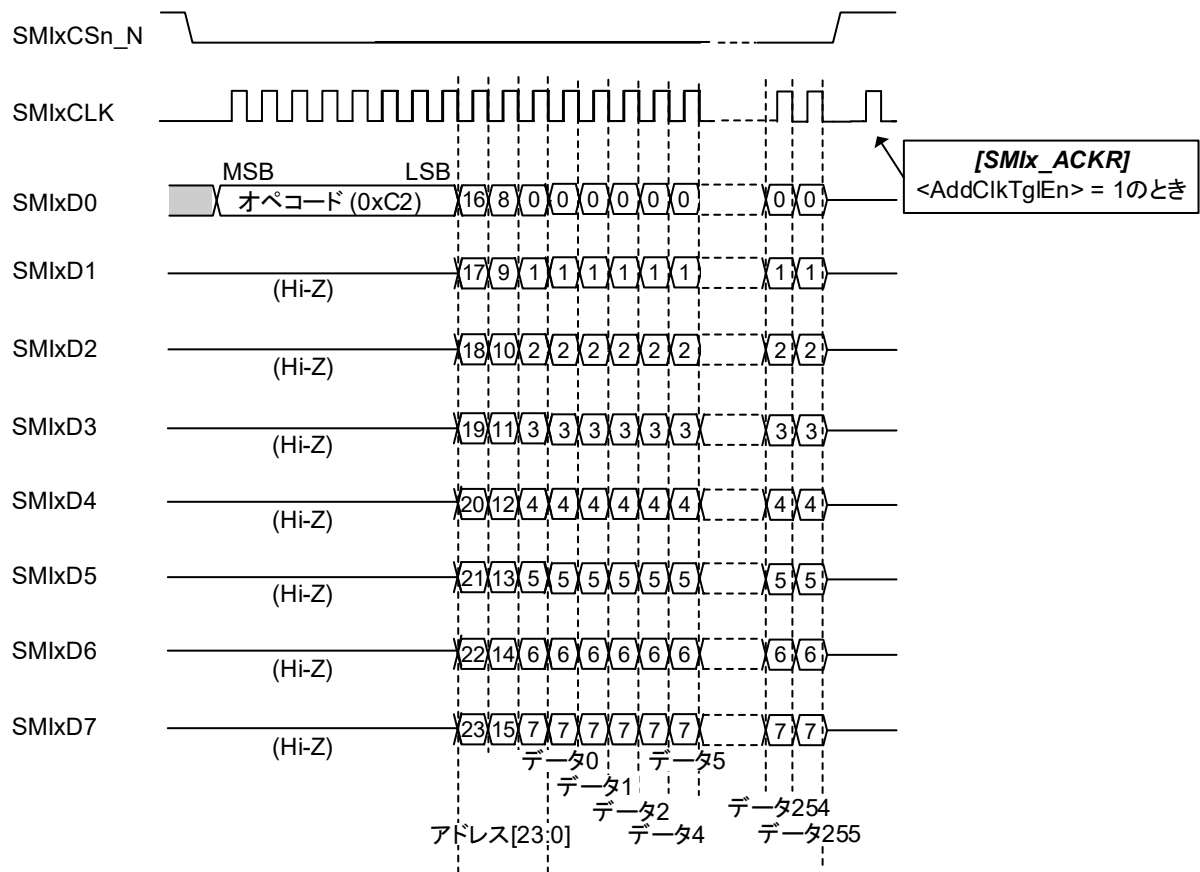


図 5.4 Octal I/O PAGE PROGRAM、Octal I/O WRITE

5.3.6. プログラミング例: OPI PAGE PROGRAM、OPI WRITE

256 バイトのデータ(Data[2047: 0])を SPI Flash の 4 バイト物理アドレス(Addr[31: 0])に OPI で送信する手順を示します。

注) SPI Flash は、本コマンドを実行後に BUSY 状態になります。BUSY 状態のデバイスは、ステータスレジスタリードなどの一部コマンドにしか正常な通信をしません。SPI Flash に命令を置いている場合などは、BUSY 状態が解除されるまで、SPI Flash に対する命令フェッチが起こらないようにしてください。BUSY 中に SPI Flash へメモリーリードアクセスをした場合、SPI Flash からは正常なデータが読みだせません。

コマンドの構成は、「コマンドオペコードバイト数: 2、アドレスバイト数: 4、ダミーバイト数: 0、ペイロードデータバイト数: 256」とします。コマンドの I/O 幅は、全て 8 × I/O とします。コマンドオペコードは、"0x12ED"とします。

(1) プライマリバッファに以下の値を設定します。

- <PBuf0[7:0]> = オペコード (0x12)
- <PBuf1[7:0]> = オペコード (0xED)
- <PBuf2[7:0]> = アドレス[31: 24] (Addr[31: 24])
- <PBuf3[7:0]> = アドレス[23: 16] (Addr[23: 16])
- <PBuf4[7:0]> = アドレス[15: 8] (Addr[15: 8])
- <PBuf5[7:0]> = アドレス[7: 0] (Addr[7: 0])

(2) セカンダリバッファに以下の値を設定します。

- <SBuf0[7:0]> = データ 0 (Data[7:0]、アドレス[23: 0] + 0x00 に書かれるデータ)
- <SBuf1[7:0]> = データ 1 (Data[15: 8]、アドレス[23: 0] + 0x01 に書かれるデータ)
- :
- <SBuf254[7:0]> = データ 254 (Data[2039: 2032]、アドレス[23: 0] + 0xFE に書かれるデータ)
- <SBuf255[7:0]> = データ 255 (Data[2047: 2040]、アドレス[23: 0] + 0xFF に書かれるデータ)

(3) *[SMIxRACR]* の設定

- <PBufEn> = 1、<PBufBc[4:0]> = 00101
- <SBufEn> = 1、<SBufBc[7:0]> = 0xFF
- <CSNum> = SPI Flash の接続されているチップセレクト番号

(4) *[SMIxIOCR]* の設定

- <PBufIOCtrlEn[1:0]> = 11 (8 × I/O 通信)
- <SBufIOCtrlEn[1:0]> = 11 (8 × I/O 通信)
- <PBufIOCtrl[4:0]> = 00000 (マルチ I/O 通信)

(5) *[SMIxOECR]* の設定

- <PBufOCEn> = 0 (送信)
- <PBufOEnC[4:0]> = don't care

(6) *[SMIxRACR]* <CycGo>に"1"をセットし、インダイレクトアクセスを開始します。

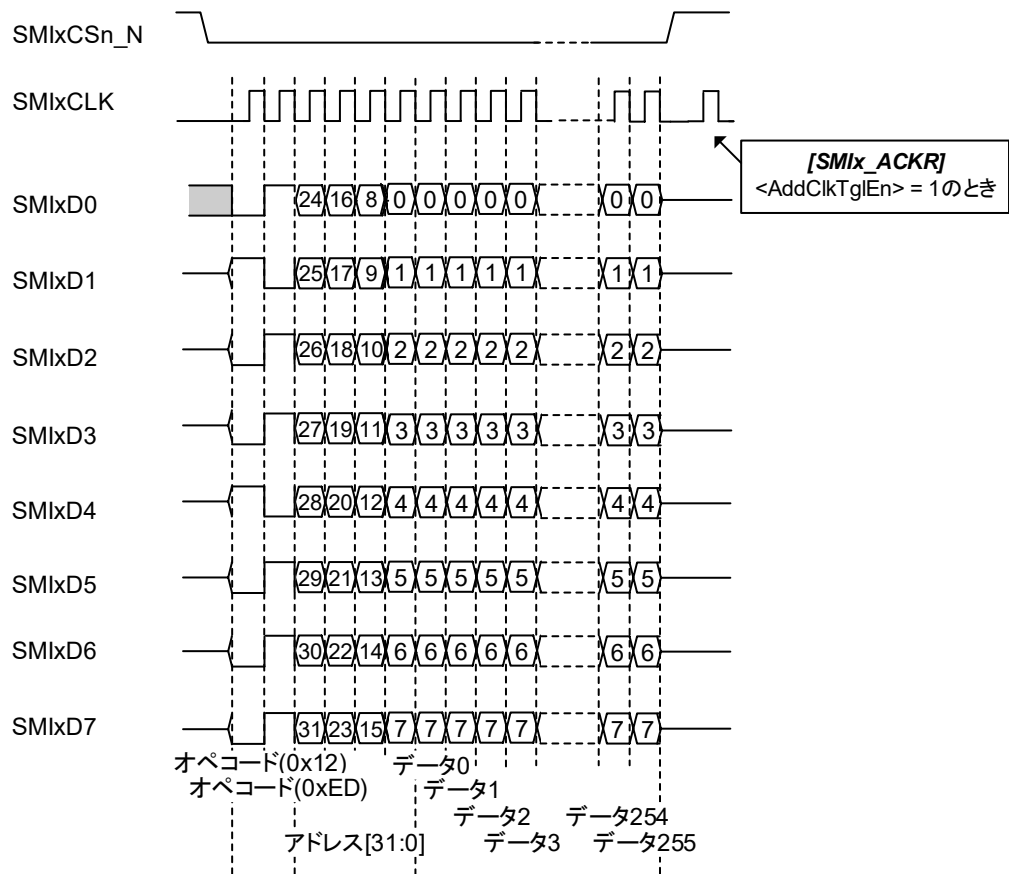


図 5.5 OPI PAGE PROGRAM、OPI WRITE

5.3.7. プログラミング例: SECTOR ERASE

SPI Flash の 3 バイト物理アドレス Addr[23: 0]に SECTOR ERASE(コマンドオペコード: 0x20)を発行する手順を示します。

注) SPI Flash は、本コマンドを実行後に BUSY 状態になります。BUSY 状態のデバイスは、ステータスレジスターリードなどの一部コマンドにしか正常な通信をしません。SPI Flash に命令を置いている場合などは、BUSY 状態が解除されるまで、SPI Flash に対する命令フェッチが起こらないようにしてください。BUSY 中に SPI Flash へメモリーリードアクセスをした場合、SPI Flash からは正常なデータが読みだせません。

(1) プライマリーバッファに以下の値を設定します。

- <PBuf0[7:0]> = オペコード (0x20)
- <PBuf1[7:0]> = アドレス[23: 16] (Addr[23: 16])
- <PBuf2[7:0]> = アドレス[15: 8] (Addr[15: 8])
- <PBuf3[7:0]> = アドレス[7: 0] (Addr[7:0])

(2) セカンダリーバッファは使用しません。

(3) *[SMIxRACR1]*の設定

- <PBufEn> = 1、<PBufBc[4:0]> = 00011
- <SBufEn> = 0、<SBufBc> = don't care
- <CSNum> = SPI Flash の接続されているチップセレクト番号

(4) *[SMIxIOCR]*の設定

- <PBufIOCtrlEn[1:0]> = 00 (SPI 通信)
- <SBufIOCtrlEn[1:0]> = 00 (SPI 通信)
- <PBufIOCtrl[4:0]> = don't care

(5) *[SMIxOECR]*の設定

- <PBufOCEn> = 0 (送信)
- <PBufOEnC[4:0]> = don't care

(6) *[SMIxRACR1]*<CycGo>に"1"をセットし、インダイレクトアクセスを開始します。

5.3.8. プログラミング例: CHIP ERASE

SPI Flash に CHIP ERASE(コマンドオペコード: 0xC7)を発行する手順を示します。

注) SPI Flash は、本コマンドを実行後に BUSY 状態になります。BUSY 状態のデバイスは、ステータスレジスターリードなどの一部コマンドにしか正常な通信をしません。SPI Flash に命令を置いている場合などは、BUSY 状態が解除されるまで、SPI Flash に対する命令フェッチが起こらないようにしてください。BUSY 中に SPI Flash へメモリーリードアクセスをした場合、SPI Flash からは正常なデータが読みだせません。

(1) プライマリーバッファに以下の値を設定します。

- <PBuf0[7:0]> = オペコード (0xC7)

(2) セカンダリーバッファは使用しません。

(3) *[SMIxRACR1]* の設定

- <PBufEn> = 1、<PBufBc[4:0]> = 00000
- <SBufEn> = 0、<SBufBc[7:0]> = don't care
- <CSNum> = シリアルメモリーの接続されているチップセレクト番号

(4) *[SMIxIOCR]* の設定

- <PBufIOCtrlEn[1:0]> = 00 (SPI 通信)
- <SBufIOCtrlEn[1:0]> = 00 (SPI 通信)
- <PBufIOCtrl[4:0]> = don't care

(5) *[SMIxOECR]* の設定

- <PBufOCEn> = 0 (送信)
- <PBufOEnC[4:0]> = don't care

(6) *[SMIxRACR1]* <CycGo> に "1" をセットし、インダイレクトアクセスを開始します。

5.3.9. プログラミング例: READ STATUS REGISTER

シリアルメモリーから、READ STATUS REGISTER (オペコード: 0x05)により、1 バイトデータを読み出す手順を示します。

- (1) プライマリバッファーに以下の値を設定します。

-<PBuf0[7:0]> = オペコード (0x05)

- (2) セカンダリバッファーには、インダイレクトアクセス完了後、デバイスが返すステータスデータが格納されます。

-<SBuf0[7:0]> = Data[7:0] (デバイスのステータスレジスタービット[7:0]が格納される)

- (3) *[SMIxRACR1]*の設定

-<PBufEn> = 1、<PBufBc[4:0]> = 00000

-<SBufEn> = 1、<SBufBc[7:0]> = 0x00

-<CSNum> = シリアルメモリーの接続されているチップセレクト番号

- (4) *[SMIxIOCR]*の設定

-<PBufIOCtrlEn[1:0]> = 00 (SPI 通信)

-<SBufIOCtrlEn[1:0]> = 00 (SPI 通信)

-<PBufIOCtrl[4:0]> = don't care

- (5) *[SMIxOECR]*の設定

-<PBufOCEn> = 0 (受信)

-<PBufOEnC[4:0]> = don't care

- (6) *[SMIxRACR1]*<CycGo>に"1"をセットし、インダイレクトアクセスを開始します。

インダイレクトアクセスが完了すると、<SBuf0[7:0]>にシリアルメモリーのステータスレジスターの値が格納されます。

5.3.10. プログラミング例: FAST READ

シリアルメモリーの3バイト物理アドレス(Addr[23: 0])から、256バイトデータ(Data[2047: 0])のリードを受信する手順を示します。

コマンドの構成は、「コマンドオペコードバイト数: 1、アドレスバイト数: 3、ダミーバイト数: 1、ペイロードデータバイト数: 256」とします。コマンドの I/O 幅は、全て SPI とします。コマンドオペコードは、"0x0B"とします。

(1) プライマリバッファに以下の値を設定します。

- <PBuf0[7:0]> = オペコード (0x0B)
- <PBuf1[7:0]> = アドレス[23: 16] (Addr[23: 16])
- <PBuf2[7:0]> = アドレス[15: 8] (Addr[15: 8])
- <PBuf3[7:0]> = アドレス[7: 0] (Addr[7:0])

(2) セカンダリバッファには、インダイレクトアクセス完了後、シリアルメモリーからのリードデータが格納されます。

- <SBuf0[7:0]> = データ 0 (Data[7:0]、アドレス[23: 0] + 0x00 から読み出されるデータ)
- <SBuf1[7:0]> = データ 1 (Data[15: 8]、アドレス[23: 0] + 0x01 から読み出されるデータ)
- :
- <SBuf254[7:0]> = データ 254 (Data[2039: 2032]、アドレス[23: 0] + 0xFE から読み出されるデータ)
- <SBuf255[7:0]> = データ 255 (Data[2047: 2040]、アドレス[23: 0] + 0xFF から読み出されるデータ)

(3) *[SMIxRACR]*の設定

- <PBufEn> = 1、<PBufBc[4:0]> = 00100
- <SBufEn> = 1、<SBufBc[7:0]> = 0xFF
- <CSNum> = シリアルメモリーの接続されているチップセレクト番号

(4) *[SMIxIOCR]*の設定

- <PBufIOCtrlEn[1:0]> = 00 (SPI 通信)
- <SBufIOCtrlEn[1:0]> = 00 (SPI 通信)
- <PBufIOCtrl[4:0]> = don't care

(5) *[SMIxOECR]*の設定

- <PBufOCEn> = 0 (受信)
- <PBufOEnC[4:0]> = don't care

(6) *[SMIxRACR]*/*<CycGo>*に"1"をセットし、インダイレクトアクセスを開始します。

インダイレクトアクセスが完了すると、<SBuf0[7:0]>から<SBuf255[7:0]>にシリアルメモリーから読み出したデータが格納されます。

注) 実際に設定すべきダミーバイト数は、シリアルメモリーの型番や、そのコンフィグレーションレジスターの設定に依存します。

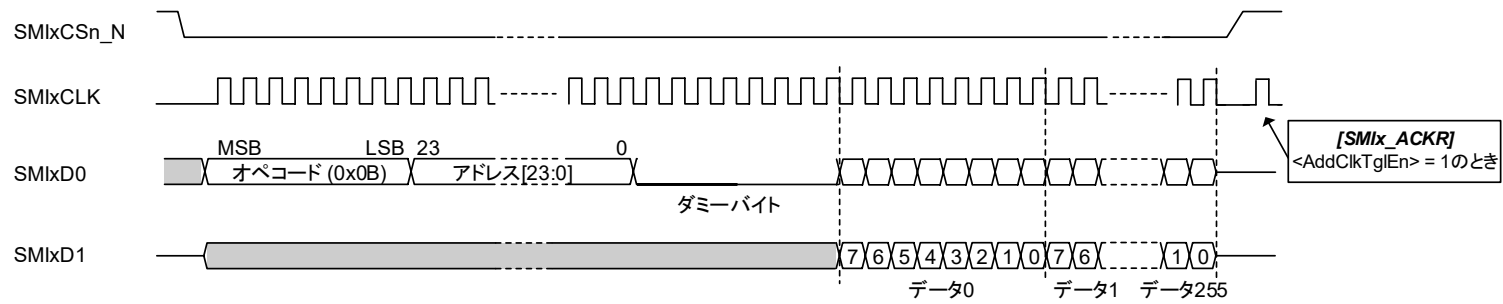


図 5.6 FAST READ

5.3.11. プログラミング例: Quad I/O FAST READ

シリアルメモリーの3バイト物理アドレス (Addr[23: 0])から、256バイトデータ (Data[2047: 0])のリードを4×I/Oで受信する手順を示します。

コマンドの構成は、「コマンドオペコードバイト数: 1、アドレスバイト数: 3、ダミーバイト数: 3、ペイロードデータバイト数: 256」とします。コマンドのI/O幅は、コマンドオペコードがSPI、アドレス以降は4×I/Oとします。コマンドオペコードは、"0xEB"とします。

(1) プライマリバッファに以下の値を設定します。

- <PBuf0[7:0]> = オペコード (0xEB)
- <PBuf1[7:0]> = アドレス[23: 16] (Addr[23:16])
- <PBuf2[7:0]> = アドレス[15: 8] (Addr[15:8])
- <PBuf3[7:0]> = アドレス[7:0] (Addr[7:0])

(2) セカンダリバッファには、インダイレクトアクセス完了後、シリアルメモリーからのリードデータが格納されます。

- <SBuf0[7:0]> = データ 0 (Data[7:0]、アドレス[23: 0] + 0x00 から読み出されるデータ)
- <SBuf1[7:0]> = データ 1 (Data[15: 8]、アドレス[23: 0] + 0x01 から読み出されるデータ)
- :
- <SBuf254[7:0]> = データ 254 (Data[2039: 2032]、アドレス[23: 0] + 0xFE から読み出されるデータ)
- <SBuf255[7:0]> = データ 255 (Data[2047: 2040]、アドレス[23: 0] + 0xFF から読み出されるデータ)

(3) *[SMIxRACR]*の設定

- <PBufEn> = 1、<PBufBc[4:0]> = 00110
- <SBufEn> = 1、<SBufBc[7:0]> = 0xFF
- <CSNum> = シリアルメモリーの接続されているチップセレクト番号

(4) *[SMIxIOCR]*の設定

- <PBufIOCtrlEn[1:0]> = 10 (4×I/O 通信)
- <SBufIOCtrlEn[1:0]> = 10 (4×I/O 通信)
- <PBufIOCtrl[4:0]> = 00001 (コマンドオペコードはシングル、アドレスからマルチ I/O 通信)

(5) *[SMIxOECR]*の設定

- <PBufOCEn> = 1
- <PBufOEnC[4:0]> = 00100 (ダミーバイトサイクルから入力に切り替え)

(6) *[SMIxRACR]*<CycGo>に"1"をセットし、インダイレクトアクセスを開始します。

インダイレクトアクセスが完了すると、<SBuf0[7:0]>から<SBuf255[7:0]>にシリアルメモリーから読み出したデータが格納されます。

注) 実際に設定すべきダミーバイト数は、シリアルメモリーの型番や、そのコンフィグレーションレジスターの設定に依存します。

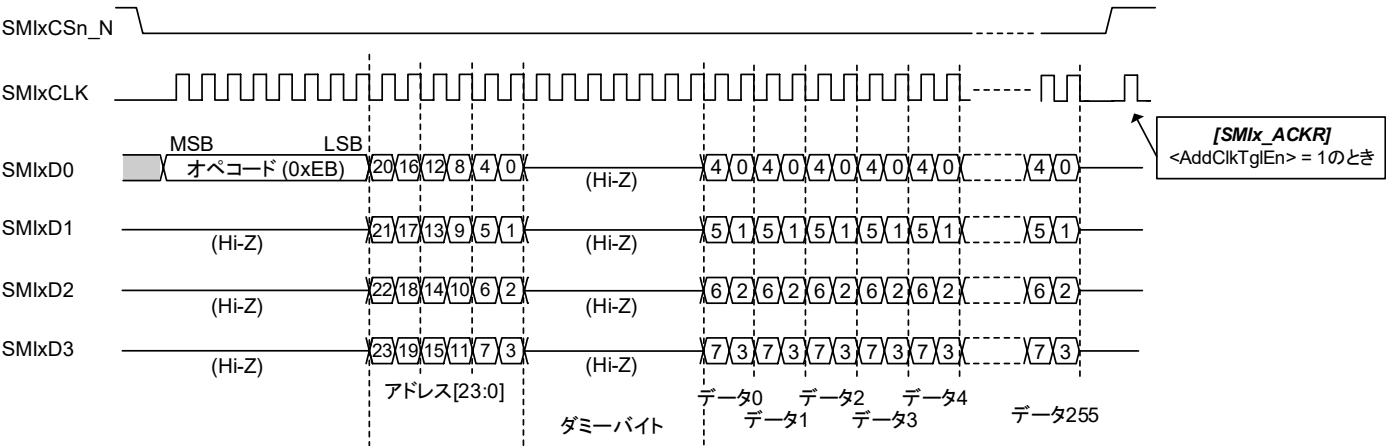


図 5.7 Quad I/O FAST READ

5.3.12. プログラミング例: QPI FAST READ

シリアルメモリーの3バイト物理アドレス(Addr[23: 0])から、256バイトデータ(Data[2047: 0])のリードを4×I/Oで受信する手順を示します。

コマンドの構成は、「コマンドオペコードバイト数: 1、アドレスバイト数: 3、ダミーバイト数: 3、ペイロードデータバイト数: 256」とします。コマンドのI/O幅は、全て4×I/Oとします。コマンドオペコードは、"0xEB"とします。

(1) プライマリバッファに以下の値を設定します。

- <PBuf0[7:0]> = オペコード (0xEB)
- <PBuf1[7:0]> = アドレス[23: 16] (Addr[23: 16])
- <PBuf2[7:0]> = アドレス[15: 8] (Addr[15: 8])
- <PBuf3[7:0]> = アドレス[7: 0] (Addr[7:0])

(2) セカンダリバッファには、インダイレクトアクセス完了後、シリアルメモリーからのリードデータが格納されます。

- <SBuf0[7:0]> = データ 0 (Data[7:0]、アドレス[23: 0] + 0x00 から読み出されるデータ)
- <SBuf1[7:0]> = データ 1 (Data[15: 8]、アドレス[23: 0] + 0x01 から読み出されるデータ)
- :
- <SBuf254[7:0]> = データ 254 (Data[2039: 2032]、アドレス[23: 0] + 0xFE から読み出されるデータ)
- <SBuf255[7:0]> = データ 255 (Data[2047: 2040]、アドレス[23: 0] + 0xFF から読み出されるデータ)

(3) *[SMIxRACR]*の設定

- <PBufEn> = 1、<PBufBc[4:0]> = 00110
- <SBufEn> = 1、<SBufBc[7:0]> = 0xFF
- <CSNum> = シリアルメモリーの接続されているチップセレクト番号

(4) *[SMIxIOCR]*の設定

- <PBufIOCtrlEn[1:0]> = 10 (4×I/O 通信)
- <SBufIOCtrlEn[1:0]> = 10 (4×I/O 通信)
- <PBufIOCtrl[4:0]> = 00000 (マルチ I/O 通信)

(5) *[SMIxOECR]*の設定

- <PBufOCEn> = 1
- <PBufOCEnC[4:0]> = 00100 (ダミーバイトサイクルから入力に切り替え)

(6) *[SMIxRACR]*<CycGo>に"1"をセットし、インダイレクトアクセスを開始します。

インダイレクトアクセスが完了すると、<SBuf0[7:0]>から<SBuf255[7:0]>にシリアルメモリーから読み出したデータが格納されます。

注) 実際に設定すべきダミーバイト数は、シリアルメモリーの型番や、そのコンフィグレーションレジスターの設定に依存します。

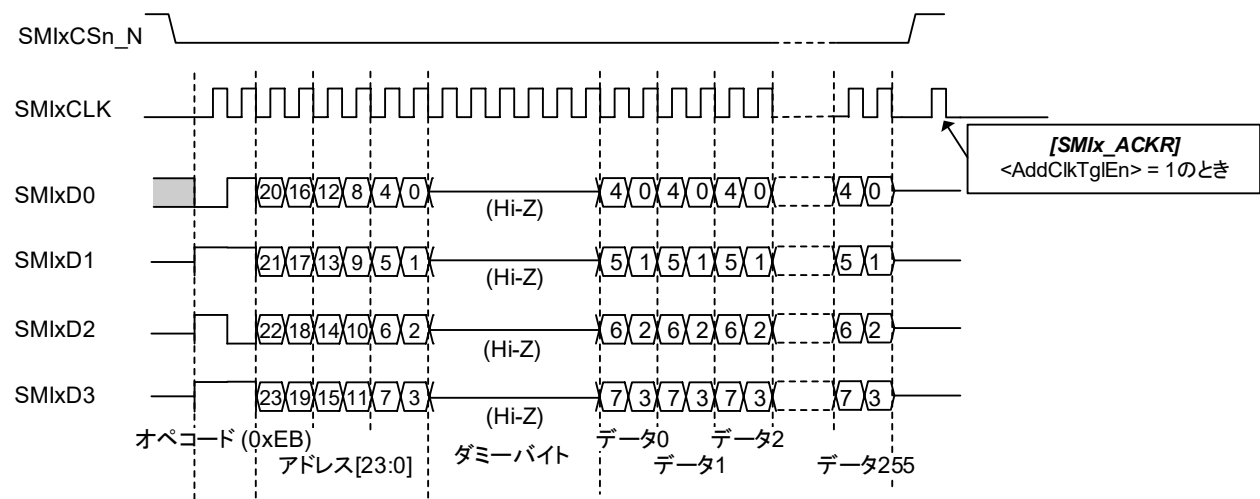


図 5.8 QPI FAST READ

5.3.13. プログラミング例: Octal I/O FAST READ

シリアルメモリーの3バイト物理アドレス(Addr[23: 0])から、256バイトデータ(Data[2047: 0])のリードを8×I/Oで受信する手順を示します。

コマンドの構成は、「コマンドオペコードバイト数: 1、アドレスバイト数: 3、ダミーバイト数: 16、ペイロードデータバイト数: 256」とします。コマンドのI/O幅は、コマンドオペコードがSPI、アドレス以降が8×I/Oとします。コマンドオペコードは、"0xCB"とします。

(1) プライマリバッファに以下の値を設定します。

- <PBuf0[7:0]> = オペコード (0xCB)
- <PBuf1[7:0]> = アドレス[23: 16] (Addr[23: 16])
- <PBuf2[7:0]> = アドレス[15: 8] (Addr[15: 8])
- <PBuf3[7:0]> = アドレス[7: 0] (Addr[7:0])

(2) セカンダリバッファには、インダイレクトアクセス完了後、シリアルメモリーからのリードデータが格納されます。

- <SBuf0[7:0]> = データ 0 (Data[7:0]、アドレス[23: 0] + 0x00 から読み出されるデータ)
- <SBuf1[7:0]> = データ 1 (Data[15: 8]、アドレス[23: 0] + 0x01 から読み出されるデータ)
- :
- <SBuf254[7:0]> = データ 254 (Data[2039: 2032]、アドレス[23: 0] + 0xFE から読み出されるデータ)
- <SBuf255[7:0]> = データ 255 (Data[2047: 2040]、アドレス[23: 0] + 0xFF から読み出されるデータ)

(3) *[SMIxRACR]*の設定

- <PBufEn> = 1、<PBufBc[4:0]> = 10011
- <SBufEn> = 1、<SBufBc[7:0]> = 0xFF
- <CsNum> = シリアルメモリーの接続されているチップセレクト番号

(4) *[SMIxIOCR]*の設定

- <PBufIOCtrlEn[1:0]> = 11 (8×I/O 通信)
- <SBufIOCtrlEn[1:0]> = 11 (8×I/O 通信)
- <PBufIOCtrl[4:0]> = 00001 (コマンドオペコードはシングル、アドレスからマルチ I/O 通信)

(5) *[SMIxOECR]*の設定

- <PBufOCEn> = 1
- <PBufOEnC[4:0]> = 00100 (ダミーバイトサイクルから入力に切り替え)

(6) *[SMIxRACR]*<CycGo>に"1"をセットし、インダイレクトアクセスを開始します。

インダイレクトアクセスが完了すると、<SBuf0[7:0]>から<SBuf255[7:0]>にシリアルメモリーから読み出したデータが格納されます。

注) 実際に設定すべきダミーバイト数は、シリアルメモリーの型番や、そのコンフィグレーションレジスターの設定に依存します。

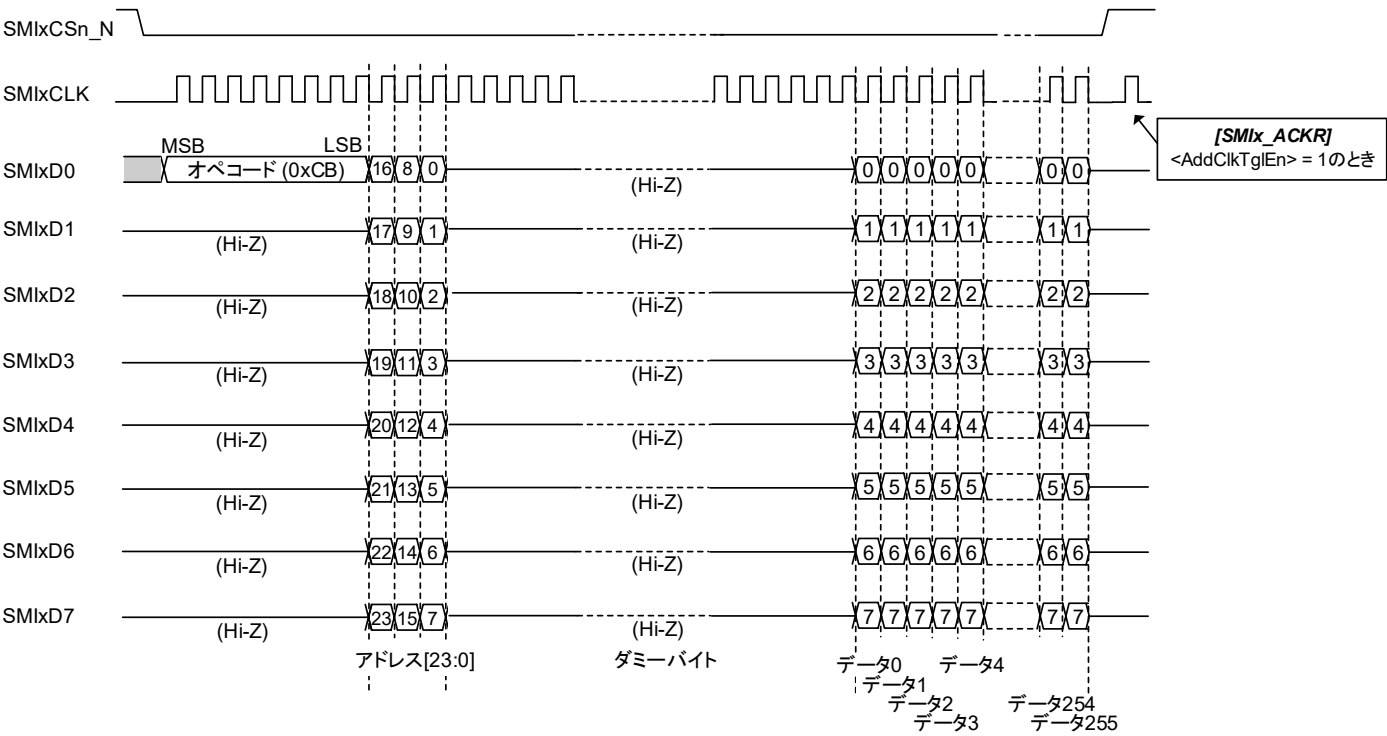


図 5.9 Octal FAST READ

5.3.14. プログラミング例: OPI FAST READ

シリアルメモリーの4バイト物理アドレス(Addr[31: 0])から、256バイトデータ(Data[2047: 0])のリードを8×I/Oで受信する手順を示します。

コマンドの構成は、「コマンドオペコードバイト数: 2、アドレスバイト数: 4、ダミーバイト数: 20、ペイロードデータバイト数: 256」とします。コマンドのI/O幅は、全て8×I/Oとします。コマンドオペコードは、「0xEC13」とします。

(1) プライマリバッファに以下の値を設定します。

- <PBuf0[7:0]> = オペコード (0xEC)
- <PBuf1[7:0]> = オペコード (0x13)
- <PBuf2[7:0]> = アドレス[31: 24] (Addr[31: 24])
- <PBuf3[7:0]> = アドレス[23: 16] (Addr[23: 16])
- <PBuf4[7:0]> = アドレス[15: 8] (Addr[15: 8])
- <PBuf5[7:0]> = アドレス[7: 0] (Addr[7: 0])

(2) セカンダリバッファには、インダイレクトアクセス完了後、シリアルメモリーからのリードデータが格納されます。

- <SBuf0[7:0]> = データ 0 (Data[7:0]、アドレス[23: 0] + 0x00 から読み出されるデータ)
- <SBuf1[7:0]> = データ 1 (Data[15: 8]、アドレス[23: 0] + 0x01 から読み出されるデータ)
- ⋮
- <SBuf254[7:0]> = データ 254 (Data[2039: 2032]、アドレス[23: 0] + 0xFE から読み出されるデータ)
- <SBuf255[7:0]> = データ 255 (Data[2047: 2040]、アドレス[23: 0] + 0xFF から読み出されるデータ)

(3) *[SMIxRACR1]*の設定

- <PBufEn> = 1、<PBufBc[4:0]> = 11001
- <SBufEn> = 1、<SBufBc[7:0]> = 0xFF
- <CSNum> = シリアルメモリーの接続されているチップセレクト番号

(4) *[SMIxIOCR]*の設定

- <PBufIOCtrlEn[1:0]> = 11 (8×I/O 通信)
- <SBufIOCtrlEn[1:0]> = 11 (8×I/O 通信)
- <PBufIOCtrl[4:0]> = 00000 (全てマルチ I/O 通信)

(5) *[SMIxOECR]*の設定

- <PBufOCEn> = 1.
- <PBufOEnC[4:0]> = 00110 (ダミーバイトサイクルから入力に切り替え)

(6) *[SMIxRACR1]*/*<CycGo>*に"1"をセットし、インダイレクトアクセスを開始します。

インダイレクトアクセスが完了すると、<SBuf0[7:0]>から<SBuf255[7:0]>にシリアルメモリーから読み出したデータが格納されます。

注) 実際に設定すべきダミーバイト数は、シリアルメモリーの型番や、そのコンフィグレーションレジスターの設定に依存します。



5.4. その他

5.4.1. 強制停止

強制停止は、SMIF の動作を緊急停止する処理です。シリアルメモリーと通信中の場合は、通信が途中で打ち切れ、チップセレクト信号が"High"になります。この際、通信データは保証されません。

強制停止をした後でも、SMIF の内蔵レジスターにはアクセス可能です。ただし、シリアルメモリーとの通信はできません。再度通信をする場合は、ソフトウェアリセットを実施と、通信設定を行う必要があります。

強制停止の実行手順を以下に示します。

- (1) 割り込みを利用する場合は、**[SMIxINT]/<SDIntEn>**に"1"をライトします。
- (2) **[SMIxSTPR]/<StopEn>**に"1"をライトします。
- (3) 割り込みの発生、または**[SMIxSTAT]/<StpProgDone>**に"1"がセットされるのを待ちます。
- (4) **[SMIxSTAT]/<StpProgDone>**に"0"をライトします
- (5) 5.4.2 の手順に従い、ソフトウェアリセットを発行します。

5.4.2. ソフトウェアリセット

ソフトウェアリセットの実行手順を、以下に示します。

- (1) **[SMIxSTAT]/<DirAccInPrgrs>**、**<CycPrgrs>**の両方が"0"であることを確認します。
- (2) **[SMIxSWR]/<RstEn>**に"1"をライトします。
- (3) **[SMIxSWR]/<RstEn>**に"0"が読み出されるまで待ちます。

注 1) 強制停止を行った場合、手順 1 は省略できます。

注 2) 本手順の実行中に、新規でダイレクトアクセスやインダイレクトアクセス要求を SMIF に発行してはいけません。

5.5. シリアルメモリーとの接続例

以下にシリアルメモリーとの接続例を示します。

各端子は、下記のように外部で処理を行うことを想定しています。実際に接続するシリアルメモリーや外部回路により各端子の処理を判断してください。

- SMiXCS0_N、SMiXCS1_N: Pull-up
- SMiXCLK: Pull-down
- SMiXD0 ~ 7: Pull-up (注)

注) シリアルメモリーとの接続に使用しない端子は、入出力ポートとして使用できます。

5.5.1. Standard SPI での接続例

(1) シリアルメモリー0

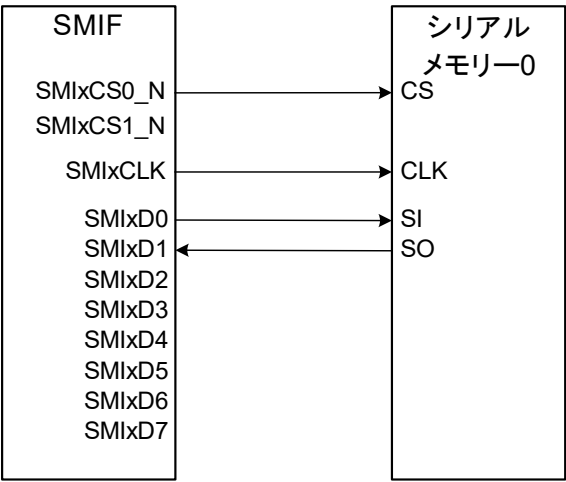


図 5.11 Standard SPIの接続例 (シリアルメモリー0)

(2) シリアルメモリー0、シリアルメモリー1

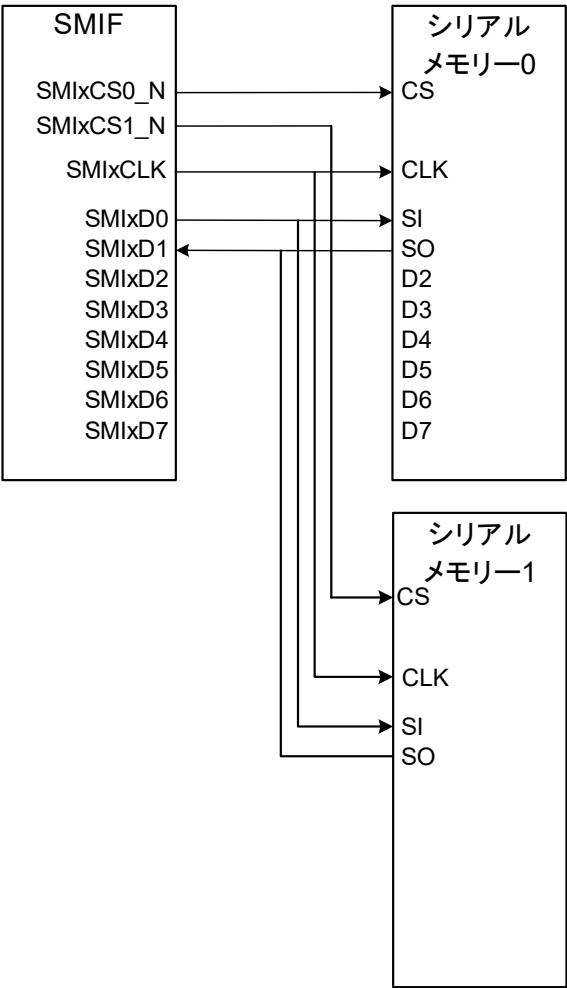


図 5.12 Standard SPIの接続例 (シリアルメモリー0/1)

5.5.2. Dual/DPI での接続例

(1) シリアルメモリー0

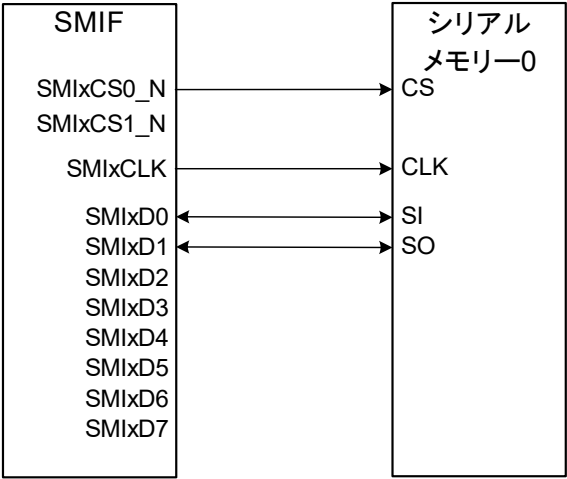


図 5.13 Dual/DPIの接続例 (シリアルメモリー0)

(2) シリアルメモリー0、シリアルメモリー1

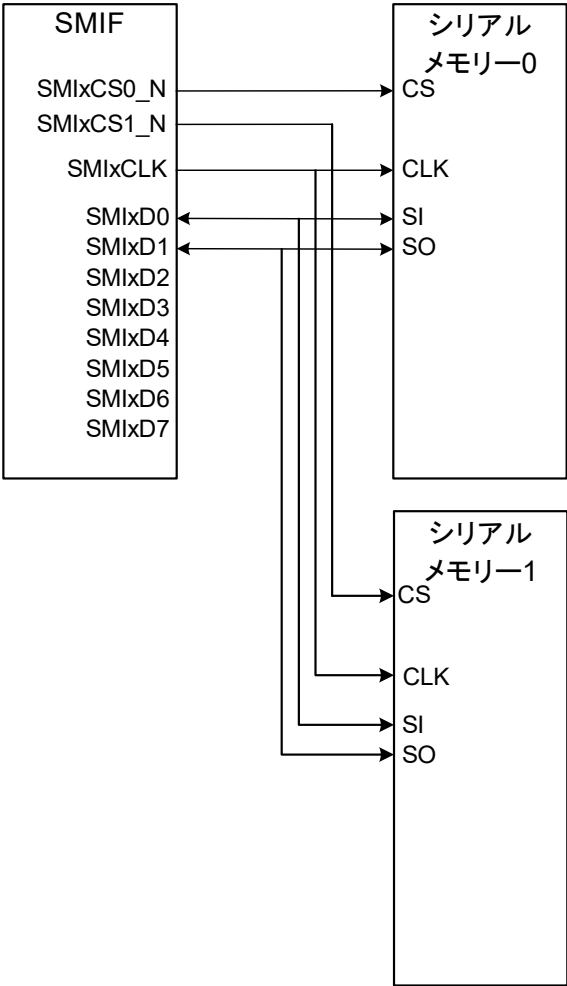


図 5.14 Dual/DPIの接続例 (シリアルメモリー0/1)

5.5.3. Quad/QPI での接続例

(1) シリアルメモリー0

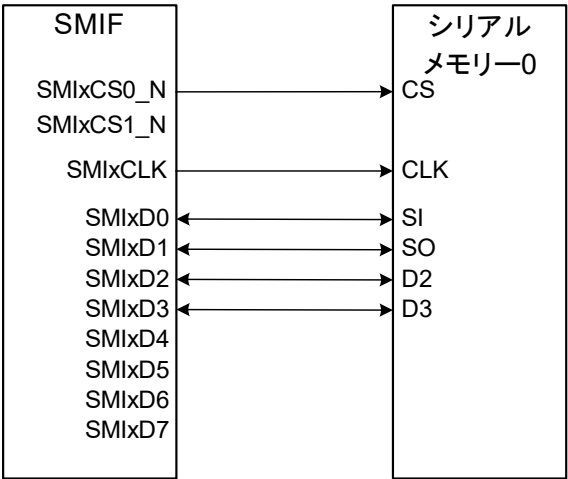


図 5.15 Quad/QPIの接続例 (シリアルメモリー0)

(2) シリアルメモリー0、シリアルメモリー1

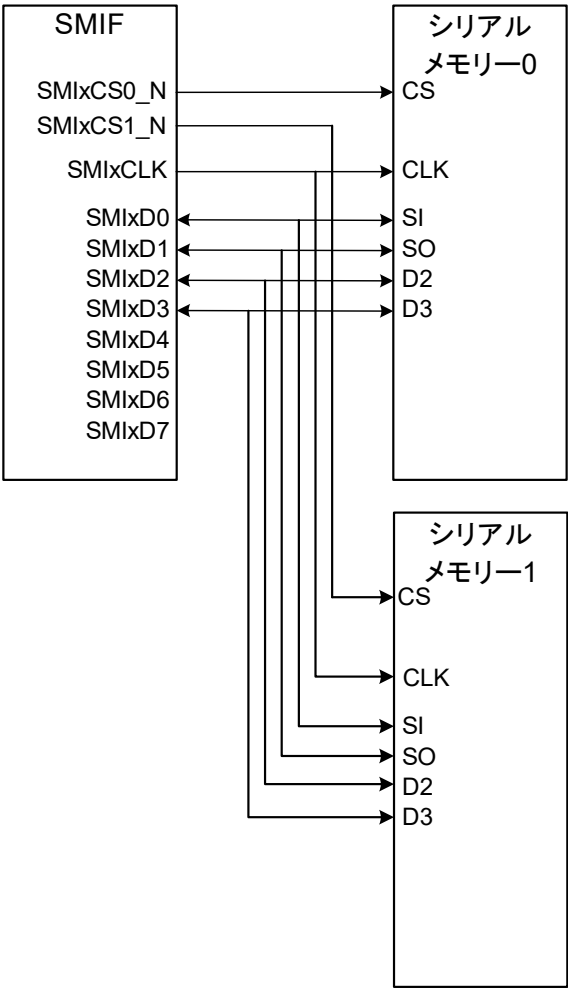


図 5.16 Quad/QPIの接続例 (シリアルメモリー0/1)

5.5.4. Octal/OPI での接続例

(1) シリアルメモリー0

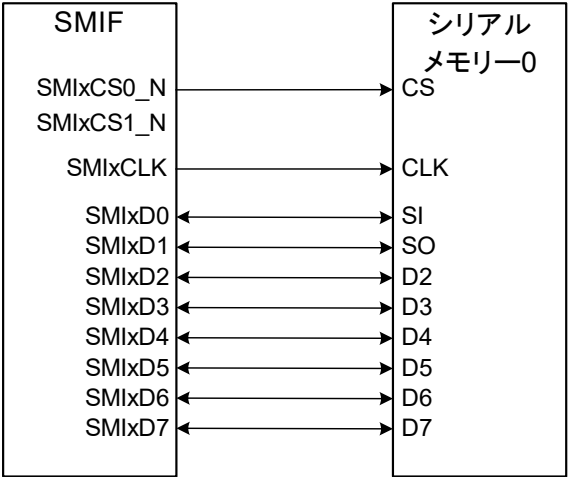


図 5.17 Octal/OPIの接続例 (シリアルメモリー0)

(2) シリアルメモリー0、シリアルメモリー1

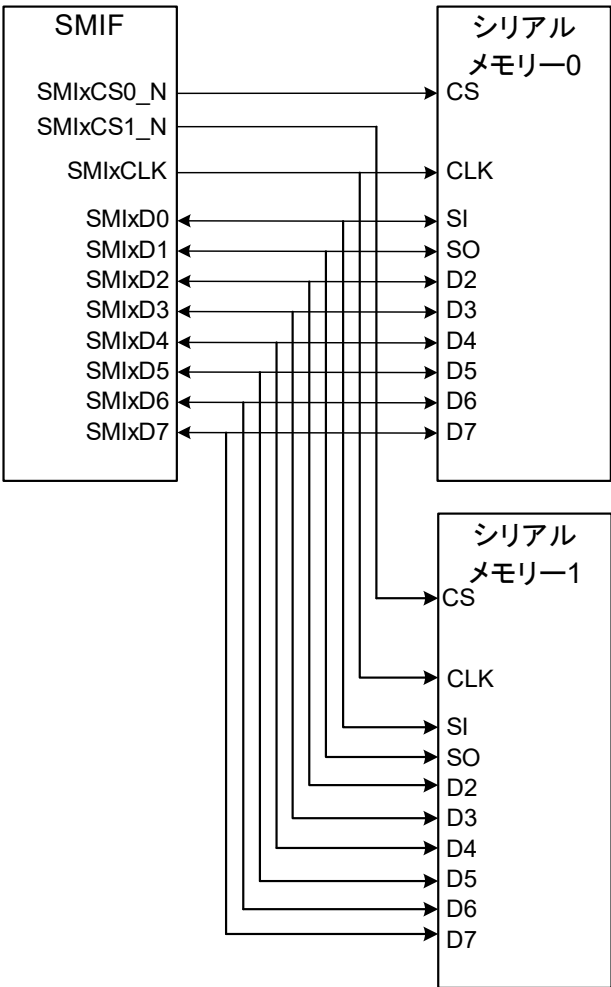


図 5.18 Octal/OPIの接続例 (シリアルメモリー0/1)

6. 使用上のご注意およびお願い事項

レジスターがアサインされていないアドレスはアクセスしないでください。

7. 改訂履歴

表 7.1 改訂履歴

Revision	Date	Description
1.0	2026-01-30	・新規作成

製品取り扱い上のお願い

株式会社東芝およびその子会社ならびに関係会社を以下「当社」といいます。

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を以下「本製品」といいます。

- 本製品に関する情報等、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・ストレージ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、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(ヘルスケア除く)、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、発電関連機器などが含まれますが、本資料に個別に記載する用途は除きます。特定用途に使用された場合には、当社は一切の責任を負いません。なお、詳細は当社営業窓口まで、または当社 Web サイトのお問い合わせフォームからお問い合わせください。
- 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当社営業窓口までお問い合わせください。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。