32-bit RISC Microcontroller

TMPM4L Group(1)

Reference Manual
Clock Control and Operation Mode
(CG-M4L(1)-A)

Revision 1.0

2018-09

TOSHIBA ELECTRONIC DEVICES & STORAGE CORPORATION

Contents

List of Figures

List of Tables

Preface

Related document

Document name
Exception

Conventions

- Numeric formats follow the rules as shown below:
  Hexadecimal: 0xABC
  Decimal: 123 or 0d123 – Only when it needs to be explicitly shown that they are decimal numbers.
  Binary: 0b111 – It is possible to omit the "0b" when the number of bit can be distinctly understood from a sentence.
- "_N" is added to the end of signal names to indicate low active signals.
- It is called "assert" that a signal moves to its active level, "deassert" to its inactive level.
- When two or more signal names are referred, they are described like as [m: n].
  Example: S[3: 0] shows four signal names S3, S2, S1 and S0 together.
- The characters surrounded by ***[ ]*** defines the register.
  Example: ***[ABCD]***
- "n" substitutes suffix number of two or more same kind of registers, fields, and bit names.
  Example: ***[XYZ1]***, ***[XYZ2]***, ***[XYZ3]*** → ***[XYZn]***
- "x" substitutes suffix number or character of units and channels in the Register List.
  In case of unit, "x" means A, B, and C ...
  Example: ***[ADACR0]***, ***[ADBCR0]***, ***[ADCCR0]*** → ***[ADxCR0]***
  In case of channel, "x" means 0, 1, and 2 ...
  Example: ***[T32A0RUNA]***, ***[T32A1RUNA]***, ***[T32A2RUNA]*** → ***[T32AxRUNA]***
- The bit range of a register is written like as [m: n].
  Example: Bit[3: 0] expresses the range of bit 3 to 0.
- The configuration value of a register is expressed by either the hexadecimal number or the binary number.
  Example: ***[ABCD]***<EFG> =0x01 (hexadecimal), ***[XYZn]***<VW> =1 (binary)
- Word and Byte represent the following bit length.
  Byte: 8 bits
  Half word: 16 bits
  Word: 32 bits
  Double word: 64 bits
- Properties of each bit in a register are expressed as follows:
  R: Read only
  W: Write only
  R/W: Read and Write are possible
- Unless otherwise specified, register access supports only word access.
- The register defined as reserved must not be rewritten. Moreover, do not use the read value.
- The value read from the bit having default value of "—" is unknown.
- When a register containing both of writable bits and read-only bits is written, read-only bits should be written with their default value, In the cases that default is "—", follow the definition of each register.
- Reserved bits of the Write-only register should be written with their default value.
  In the cases that default is "—", follow the definition of each register.
- Do not use read-modified-write processing to the register of a definition which is different by writing and read out.

The flash memory uses the Super Flash® technology under license from Silicon Storage Technology, Inc.
Super Flash® is registered trademark of SST, Inc.

All other company names, product names, and service names mentioned herein may be trademarks of their respective companies.

Terms and Abbreviations

Some of abbreviations used in this document are as follows:

ADC	Analog to Digital Converter
A-ENC32	Advanced Encoder input Circuit (32-bit)
A-VE	Advanced Vector Engine
CG	Clock Control and Operation Mode
CRC	Cyclic Redundancy Check
DNF	Digital Noise Filter
EHOSC	External High speed Oscillator
fsys	frequency of SYSTEM Clock
IHOSC	Internal High speed Oscillator
INT	Interrupt
LVD	Voltage Detection Circuit
NMI	Non-Maskable Interrupt
OFD	Oscillation Frequency Detector
PMD+	Programmable Motor Control Plus Circuit
POR	Power On Reset Circuit
RAMP	RAM Parity Circuit
SIWDT	Clock Selective Watchdog Timer
TRGSEL	Trigger Selection circuit
TRM	Trimming Circuit
TSPI	Toshiba Serial Peripheral Interface
T32A	32-bit Timer Event Counter
UART	Universal Asynchronous Receiver Transmitter

1. Outlines

The clock mode control block can select a clock gear or prescaler clock and set the warm up of oscillator.
Furthermore, it has NORMAL mode and a Low power consumption mode in order to reduce power consumption using mode transition.

There is the following as a function relevant to a clock.

- System clock control
- Prescaler clock control

2. Clock Control

2.1. Clock Type

This section shows a list of clocks:

EHCLKIN: The clock input from the external
f_{OSC} : A clock generated in the internal oscillation circuit or input from the X1 and X2 pins
f_{PLL} : A clock multiplied by PLL
fc : A clock selected by ***[CGOSCCR]***<OSCSEL> (high speed clock)
fsys : A system clock selected by ***[CGSYSCR]***<GEAR[2:0]>
ΦT0 : A clock selected by ***[CGSYSCR]***<PRCK[3:0]> (prescaler clock)
f_{IHOSC1} : A clock generated with the internal high speed oscillator 1
f_{IHOSC2} : A clock generated with the internal high speed oscillator 2
ADCLK : A conversion clock for AD converter

2.2. The Initial Value by a reset action

A clock setup is initialized by the following states by a reset action.

External high speed oscillator : Stop
Internal high speed oscillator 1 : Oscillation
Internal high speed oscillator 2 : Stop
PLL (Clock multiplying circuit) : Stop
Gear clock : fc(no frequency dividing)

2.3. Clock System Diagram

The figure below shows a clock system diagram.

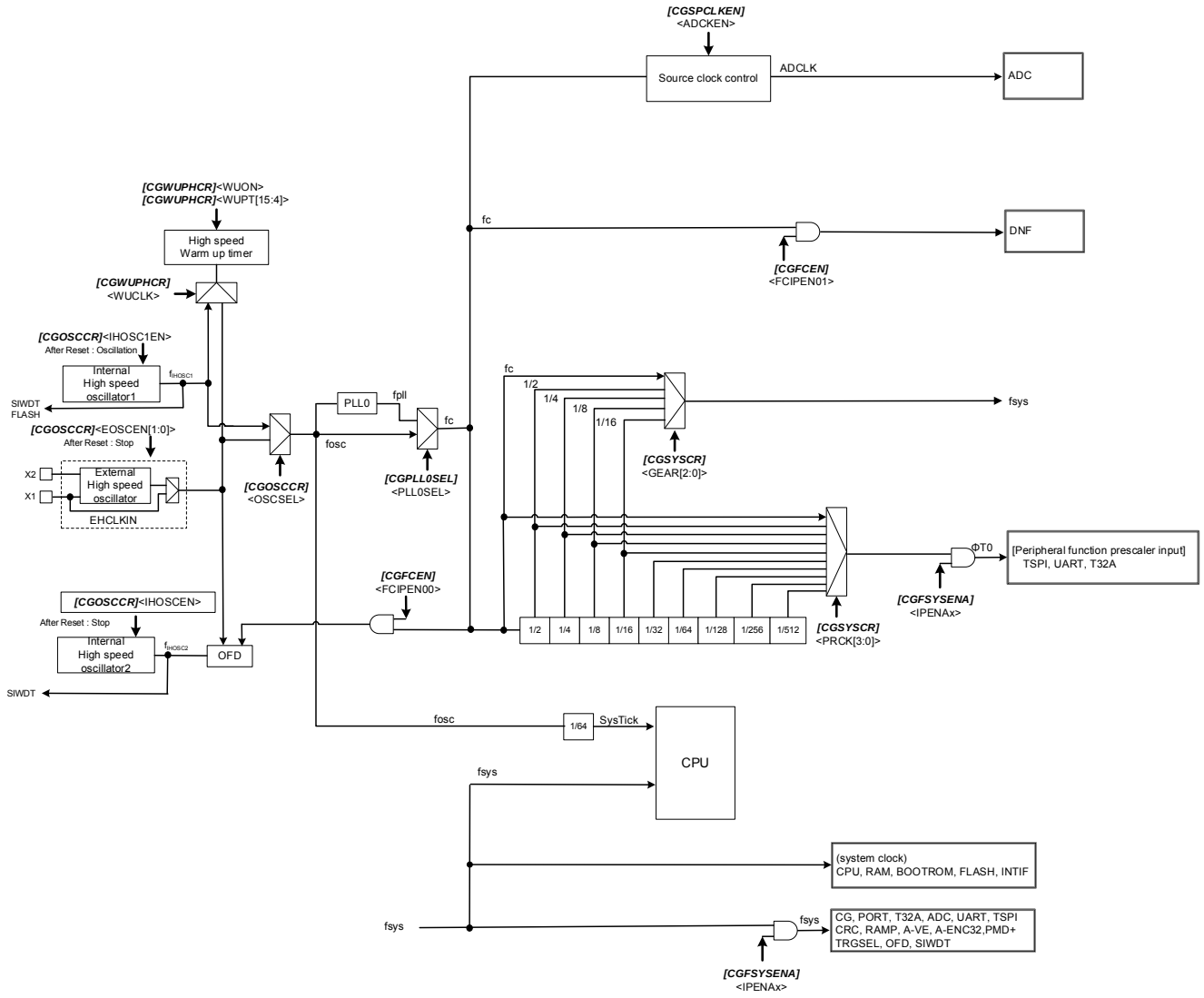

Figure 2.1 Clock System Diagram

2.4. Warming up function

A function for a warming up function to secure the oscillation stable time at the time of the STOP1 mode release which starts the warming up counter for high speed clock automatically,

It is available also as a count up timer which uses the exclusive warming up counter of high speed clock for the waiting for the stability of an external oscillator or an internal oscillator.

This chapter explains the setting method to the register for warming up timers, and the case where it is used as a count up timer. The detailed explanation at the time of STOP1 mode release, refer to "3.3.2 Warming up at the release of Low power consumption Mode".

2.4.1. The warming up counter for a high speed oscillation

A 16-bit up counter is built in as a warming up counter only for a high speed oscillation. Also when setting before changing to the STOP1 mode, it computes in the following formula, 4 bits of low ranks are omitted, and it sets to top 12 bits. A register will be set as ***[CGWUPHCR]***<WUPT[15:4]>. 16 is subtracted in order to perform the count for 4 bits of low ranks, even when a set point is 0.

<Formula>

Warming up counter value (16 bits) = (Warming up time (s) / Clock period (s)) - 16

(Example) When 5 ms of warming time is set up with 10 MHz (100 ns of clock periods) of oscillators

Warming up counter value (16 bits) = (5ms / 100ns) - 16
= 50000 - 16
= 49984
= 0xC340

Since top 12 bits is set up, it sets to a register as follows.
[CGWUPHCR]<WUPT[15:4]> = 0xC34

In the case of 10 MHz, the setting range is 0 ≤ <WUPT[15:4]> ≤ 0xFFF, Warming up time is set from 1.6μs to 6.5536ms.

2.4.2. The directions for a warming up timer

The directions for a warming up function are explained.

(1) Selection of a clock

In a high speed oscillation, the clock classification (an internal oscillation / external oscillation) counted at a warming up counter is chosen by ***[CGWUPHCR]***<WUCLK>.

(2) Calculation of a warming up counter set value

The warming up time can set any value to the counter a high speed oscillation. Please compute and set up from the formula.

(3) The start of warming up, and a termination Confirmation

When software (command) performs the start of warming up, and a termination Confirmation, a warming up count start is carried out by setting "1" to ***[CGWUPHCR]***<WUON>. Termination is ***[CGWUPHCR]*** <WUEF>. It distinguishes by becoming "1" to "0". "1" shows the inside of warming up and "0" shows termination. After a counting end, a counter is reset and returns to an initial state.

It does not become forced termination although "0" is written in during counter operation to ***[CGWUPHCR]*** <WUON>. "0" writing is disregarded.

Note: Since it is operating with the oscillating clock, a warming up timer includes an error, when Oscillation frequency has fluctuation. Therefore, It serves as time of an outline.

2.5. Clock Multiplying Circuit (PLL)

The clock multiplying circuit outputs the f_{PLL} clock multiplied by the optimum condition for the frequency (6 MHz to 12 MHz) of the output clock f_{OSC} of the high speed oscillator.

So, it is possible to make input frequency to an oscillator low and to make an internal clock high speed by this circuit.

TMPM4L Group(1) products is implemented a PLLs for fsys (PLL0: a maximum of 80 MHz).

2.5.1. A PLL setup after reset release

The PLL is disabled after reset release.

In order to use the PLL, set a multiplication value to ***[CGPLL0SEL]***<PLL0SET> while ***[CGPLL0SEL]*** <PLL0ON> is "0". Then wait until approximately 100µs has elapsed as a PLL initial stabilization time, and set "1" to <PLL0ON> to start PLL operation. After that, to use f_{PLL} clock which is multiplied f_{OSC}, wait until approximately 400µs has elapsed as a lock up time. Then set "1" to ***[CGPLL0SEL]***<PLL0SEL>.

Note that a warm up time is required until PLL operation becomes stable using the warm up function, etc.

2.5.2. The formula and the example of a setting of a PLL multiplication value

The details of the items of ***[CGPLL0SEL]***<PLL0SET[23:0]> which set up a PLL multiplication value are shown below.

Table 2.1 Details of *[CGPLL0SEL]*<PLL0SET[23:0]> setup

The items of PLL0SET	Function	
[23:17]	Correction value setup	The quotient of f_{OSC}/450000 (integers). For detail, refer to "Table 2.2".
[16:14]	f_{OSC} setup	111: 20 < f_{OSC} ≤ 24 (unit: MHz) 011: 10 < f_{OSC} ≤ 20 010: Reserved 001: 6 ≤ f_{OSC} ≤ 10 000: Reserved
[13:12]	Dividing setup	00: Reserved 01: 2 dividing (×1/2) 10: 4 dividing (×1/4) 11: 8 dividing (×1/8)
[11:8]	Fraction part multiplication setup	0000: 0.0000 0001: 0.0625 0010: 0.1250 0011: 0.1875 0100: 0.2500 0101: 0.3125 0110: 0.3750 0111: 0.4375 1000: 0.5000 1001: 0.5625 1010: 0.6250 1011: 0.6875 1100: 0.7500 1101: 0.8125 1110: 0.8750 1111: 0.9375
[7:0]	Integer part multiplication setup	0x00: 0 0x01: 1 0x02: 2 : 0xFD: 253 0xFE: 254 0xFF: 255

Note: A multiplication value is the total of <PLL0SET[7:0]> (integer part) and <PLL0SET[11:8]> (fraction part).

f_{PLL} is denoted by the following formulas.

$$f_{PLL} = f_{OSC} \times (\textbf{\textit{[CGPLL0SEL]}}\text{<PLL0SET[7:0]>} + \textbf{\textit{[CGPLL0SEL]}}\text{<PLL0SET[11:8]>}) \times (\textbf{\textit{[CGPLL0SEL]}}\text{<PLL0SET[13:12]>})$$

Note 1: The absolute value of frequency accuracy is not guaranteed.

Note 2: There is no Linearity in the frequency by the Fraction part Multiplication setup.

Note 3: $f_{PLL} \leq$ (Maximum Operating Frequency)

Table 2.2 PLL correction (example)

f_{OSC}(MHz)	<PLL0SET[23:17]> (An integer value in decimal)
6.00	14
8.00	18
10.00	23
12.00	27

A PLL correction can be calculated below.

f_{OSC}=6.0MHz, 6.0/0.45=13.33 → 14; A decimal fraction rounds up

The main examples of a setting of ***[CGPLL0SEL]***<PLL0SET[23:0]> for fsys are shown below.

It multiplies by PLL, and dividing is carried out and the target Clock frequency (f_{PLL}) is generated for input frequency (f_{OSC}).

A dividing value is chosen from 1/2, 1/4, and 1/8.

Moreover, set up the frequency after multiplication in the following ranges.

200MHz ≤ (f_{OSC} × Multiplication value) ≤ 400MHz

Table 2.3 PLL0SET set point (example)

f_{OSC}(MHz)	Multiplication value	Dividing value	f_{PLL}(MHz)	<PLL0SET[23:0]>
6.00	53.3125	1/4	79.97	0x1C6535
8.00	40.0000	1/4	80.00	0x246028
10.00	32.0000	1/4	80.00	0x2E6020
12.00	26.6250	1/4	79.88	0x36EA1A

2.5.3. Change of the PLL multiplication value under operation

It changes to a setup which sets "0" to ***[CGPLL0SEL]***<PLL0SEL> first, and does not use a PLL multiplication clock during PLL multiplication clock operation when changing a multiplication value. And ***[CGPLL0SEL]*** <PLL0ST> =0 is read, after checking having changed to a setup which does not use a multiplication clock, ***[CGPLL0SEL]***<PLL0ON> is set to "0", and PLL is stopped.

Then, the multiplication value of ***[CGPLL0SEL]***<PLL0SET> is changed, as reset time of PLL, after about 100μs progress, ***[CGPLL0SEL]***<PLL0ON> is set as "1", and operation of PLL is started.

Then, ***[CGPLL0SEL]***<PLL0SEL> is set as "1" after lock up time and about 400μs progress.

Finally, ***[CGPLL0SEL]***<PLL0ST> is read and it checks having changed.

2.5.4. PLL operation start / stop / switching procedure

2.5.4.1. fc setup (PLL stop >>> PLL start)

As an fc setup, the example of switching procedure from the PLL stop state to the PLL operation state is as follows.

<< The state before switching >>	
[CGPLL0SEL]<PLL0ON> =0	Stops the PLL operation for fsys.
[CGPLL0SEL]<PLL0SEL> =0	Selects the setting of the PLL for fsys to "PLL is unused (f_{OSC})".
[CGPLL0SEL]<PLL0ST> =0	Selects the status of the PLL for fsys to "PLL is unused (f_{OSC})".

<< The example of switching procedure >>		
1	***[CGPLL0SEL]***<PLL0SET> =0xX	A PLL multiplication value (0xX) setup is chosen.
2	Wait 100μs or more.	Latency time after a multiplication setup.
3	***[CGPLL0SEL]***<PLL0ON> =1	PLL operation for fsys is carried out to an oscillation.
4	Wait 400μs or more.	PLL output clock stable latency time. (Lock up time)
5	***[CGPLL0SEL]***<PLL0SEL> =1	PLL selection for fsys is carried out to PLL use (f_{PLL}).
6	***[CGPLL0SEL]***<PLL0ST> is read.	It waits until the PLL selection status for fsys becomes PLL use (f_{PLL}) (=1).

Note: 1 to 4 is unnecessary when the state before switching is ***[CGPLL0SEL]***<PLL0ON> =1. When changing from the state where the PLL Output clock was stabilized, it can change to the conduct PLL state by execution of only 5 and 6.

2.5.4.2. fc setup (PLL operating >>> PLL stop)

As an fc setup, the example of switching procedure from the PLL operation state to a PLL stop state is as follows.

<< The state before switching >>	
[CGPLL0SEL]<PLL0ON> =1	Sets the PLL oscillation for fsys.
[CGPLL0SEL]<PLL0SEL> =1	Select the PLL for fsys to "PLL is used (f_{PLL})".
[CGPLL0SEL]<PLL0ST> =1	Select the status of the PLL for fsys to "PLL is used (f_{PLL})".

<< The example of switching procedure >>		
1	***[CGPLL0SEL]***<PLL0SEL> =0	Select the PLL for fsys to "PLL is unused (f_{OSC})".
2	***[CGPLL0SEL]***<PLL0ST> is read.	Waits until the status of the PLL for fsys becomes "PLL is unused (f_{OSC}) (=0)".
3	***[CGPLL0SEL]***<PLL0ON> =0	Sets the PLL oscillation for fsys to stop.

2.6. System Clock

An internal high speed oscillation clock and external high speed oscillation clock (connected oscillator or clock input) can be used as a source of system clock.

Dividing is possible for a system clock at ***[CGSYSCR]***<GEAR[2:0]> (clock gear). Although a setup can be changed during operation, after register writing before a clock actually changes, a maximum of 16-clock time is required of fc. Check completion of a clock change by ***[CGSYSCR]***<GEARST[2:0]>.

Note: Do not change a clock gear during operation of peripheral functions, such as a timer counter.

It is the following about the example of operation frequency by the clock gear ratio (1/1 to 1/16) to the frequency fc set up with oscillation frequency, a PLL multiplication value, etc. .

Table 2.4 The example of operation frequency (unit: MHz)

External Oscillation (MHz)	External Clock input (MHz)	Internal Oscillation IHOSC1 (MHz)	PLL Multiplication value (After dividing)	Maximum Frequency (fc)(MHz)	Clock gear PLL=ON					Clock gear PLL=OFF				
					1/1	1/2	1/4	1/8	1/16	1/1	1/2	1/4	1/8	1/16
6	6	—	13.329	79.97	79.97	39.99	20	10	5	6	3	1.5	—	—
8	8	—	10	80	80	40	20	10	5	8	4	2	1	—
10	10	10	8	80	80	40	20	10	5	10	5	2.5	1.25	—
12	12	—	6.657	79.88	79.88	39.95	19.98	9.99	4.99	12	6	3	1.5	—

2.6.1. The setting method of a system clock

2.6.1.1. f_{OSC} setup (internal oscillation >>> external oscillation)

As a f_{OSC} setup, the example of switching procedure to the external high speed oscillator (EHOSC) from an internal high speed oscillator 1 (IHOSC1) is shown below.

<< The state before switching >>	
[CGOSCCR]<IHOSC1EN> =1	An internal high speed oscillator 1 oscillates.
[CGOSCCR]<OSCSEL> =0	The high speed oscillation selection for f_{OSC} is an internal high speed oscillator 1(IHOSC1).
[CGOSCCR]<OSCF> =0	The high speed oscillation selection status for f_{OSC} is an internal high speed oscillator 1(IHOSC1).
An oscillator is connected to X1 / X2 pin.(Note)	

Note: Do not connect except an oscillator

<< The example of switching procedure >>		
1	***[PJPDN]***<bit[1:0]> =00 ***[PJIE]***<bit[1:0]> =00	Disable the pull-down of X1/X2 pin. Disable Input control of X1/X2 pin.
2	***[CGOSCCR]***<EOSCEN[1:0]> =01	It is an external high speed oscillator (EHOSC) about selection of an external high speed oscillation of operation.
3	***[CGWUPHCR]***<WUCLK> =1 ***[CGWUPHCR]***<WUPT[15:4]> = "arbitrary value"	It is the external high speed oscillator (EHOSC) about high speed oscillation warming up clock selection. Oscillator stable time is set to a warming up counter set value.
4	***[CGWUPHCR]***<WUON> =1	High speed oscillation warming up is started.
5	***[CGWUPHCR]***<WUEF> is read	It waits until it becomes the termination of high speed oscillation warming up (=0).
6	***[CGOSCCR]***<OSCSEL> =1	It is high speed oscillation selection for f_{OSC} to the external high speed oscillator (EHOSC).

<< The example of switching procedure >>		
7	***[CGOSCCR]***<OSCF> is read	It waits until the high speed oscillation selection status for f_{OSC} becomes external high speed oscillator (=1).
8	***[CGOSCCR]***<IHOSC1EN> =0	An internal high speed oscillator 1 is suspended.

2.6.1.2. f_{OSC} setup (internal oscillation >>> external clock input)

As a f_{OSC} setup, the example of switching procedure to the external clock input (EHCLKIN) from an internal high speed oscillator 1(IHOSC1) is shown below.

<< The state before switching >>	
[CGOSCCR]<IHOSC1EN> =1	An internal high speed oscillator 1 oscillates.
[CGOSCCR]<OSCSEL> =0	The high speed oscillation selection for f_{OSC} is an internal high speed oscillator 1(IHOSC1).
[CGOSCCR]<OSCF> =0	The high speed oscillation selection status for f_{OSC} is an internal high speed oscillator 1(IHOSC1).
Clock into to EHCLKIN	Input in the proper voltage range.

<< The example of switching procedure >>		
1	***[PJPDN]***<bit[0]> =0 ***[PJIE]***<bit[0]> =1	Disable the pull-down of X1 pin. Enable the input control of an X1/EHCLKIN pin.
2	***[CGOSCCR]***<EOSCEN[1:0]> =10	Selection of an external high speed oscillation of operation is carried out to an external clock input (EHCLKIN).
3	***[CGOSCCR]***<OSCSEL> =1	It is high speed oscillation selection for f_{OSC} to an external clock.
4	***[CGOSCCR]***<OSCF> is read.	It waits until the high speed oscillation selection status for f_{OSC} becomes external high speed oscillator (=1).
5	***[CGOSCCR]***<IHOSC1EN> =0	An internal high speed oscillator 1 is suspended.

2.6.1.3. f_{OSC} setup (an external oscillation / external clock input >>> internal oscillation)

As a f_{OSC} setup, the example of switching procedure to the internal high speed oscillator 1(IHOSC1) from an external high speed oscillation (EHOSC) Operation State or an external clock input (EHCLKIN) Operation State is shown below.

<< The state before switching >>	
[CGOSCCR]<EOSCEN[1:0]> = 01, 10	Selection of an external high speed oscillator of operation is an external high speed oscillator (EHOSC) or external clock input.
[CGOSCCR]<OSCSEL> =1	The high speed oscillation selection for f_{OSC} is the external high speed (EHOSC).
[CGOSCCR]<OSCF> =1	The high speed oscillation selection status for f_{OSC} is the external high speed oscillator (EHOSC).

<< The example of switching procedure >>		
1	***[CGOSCCR]***<IHOSC1EN> =1	An internal high speed oscillator 1 is oscillated.
2	***[CGOSCCR]***<IHOSC1F> is read	It waits until an internal high speed oscillation stable flag for IHOSC1 becomes oscillation stability (=1).
3	***[CGOSCCR]***<OSCSEL> =0	It is high speed oscillation selection for f_{OSC} to an internal high speed oscillator 1(IHOSC1).
4	***[CGOSCCR]***<OSCF> is read.	It waits until the high speed oscillation selection status for f_{OSC} becomes an internal high speed oscillator (=0).
5	***[CGOSCCR]***<EOSCEN[1:0]> =00	Set the selection of an external high speed oscillator operation to unused.

2.7. Clock Supply Setting function

This CPU has the clock on/off function for the peripheral circuits. To reduce the power consumption, this CPU can stop supplying the clock to the peripheral functions that are not used.

Except some peripheral functions, clocks are not supplied after reset.

In order to supply the clock of the function to be used, set the bit of relevance of ***[CGFSYSENA]***, ***[CGFCEN]***, and ***[CGSPCLKEN]*** to "1".

For details, refer to "4. Registers".

2.8. Prescaler clock

Peripheral function each have a Prescaler circuit to divide the ΦT0 clock.

The ΦT0 clock inputted into the prescaler circuit can be divided by the ***[CGSYSCR]***<PRCK[3:0]>.

As for ΦT0 clock after reset, fc is chosen.

After register writing before a clock actually changes, a maximum of 512-clock time is required of fc.

To confirm the completion of clock changed, check the status of ***[CGSYSCR]***<PRCKST[3:0]>.

Note: Do not change a prescaler clock during operation of peripheral functions, such as a timer counter.

3. Operation mode

There are NORMAL mode and a Low power consumption mode (IDLE, STOP1) in this product as an Operation mode, and it can reduce power consumption by performing mode changes according to directions for use.

3.1. Details of an Operation Mode

3.1.1. The feature in each mode

The feature in NORMAL, Low power consumption mode is as follows.

- NORMAL Mode

  They are a CPU core and the mode which operates peripheral circuits with high speed clock. After reset release serves as NORMAL mode.

- Low power consumption Mode

  The Low power consumption modes are the following 2 modes.

  – IDLE Mode

  It is the mode which CPU stops.
  The peripheral function should perform operation/stop by the register of each peripheral function, a clock supply setting function, etc.

  Note: In IDLE mode the CPU cannot perform the clearance of the watchdog timer, it is careful of it.

  – STOP1 Mode

  It is the mode which all the internal circuits also including an internal high speed oscillator stop.
  If the STOP1 mode is canceled, an internal high speed oscillator 1(IHOSC1) will start an oscillation, and will return to NORMAL mode.
  Please disable interrupt which is not used for STOP1 release before shifting to the STOP1 mode.

3.1.2. Transition to and Return from Low power consumption Mode

In order to shift to each Low power consumption mode, the IDLE/STOP1 mode is chosen by standby control register ***[CGSTBYCR]***<STBY[1:0] >, and a WFI command is executed. When it shifts to a Low power consumption mode by WFI command, the restart operation from a Low power consumption mode is performed by reset or interrupt generating. To return by interrupt, it is necessary to set up. Please refer to "Interrupts" chapter of the "Exception" of a reference manual for details.

Note 1: This product does not support a return by events; therefore, do not make a transition to Low power consumption mode triggered by WFE (Wait For Event).

Note 2: This product does not support Low power consumption mode by SLEEPDEEP of the Arm® Cortex®-M4 processor with FPU.
Do not use the <SLEEPDEEP> bit of the system control register.

3.1.3. Selection of a Low power consumption mode

Low power consumption mode selection is chosen by setup of ***[CGSTBYCR]***<STBY[1:0]>.
The following table shows the mode chosen from a setup of <STBY[1:0]>.

Table 3.1 Low power consumption mode selection

Mode	*[CGSTBYCR]* <STBY[1:0]>
IDLE	00
STOP1	01

Note: Do not use the settings other than the above.

3.1.4. The peripheral function state in a Low power consumption Mode

The following Table 3.2 shows the operation state of the peripheral function (block) in each mode.
In addition, after reset release it will be in the state where a clock is not supplied except for a part of blocks.
If needed, set up ***[CGFSYSENA]***, ***[CGFCEN]***, and ***[CGSPCLKEN]*** and enable clock supply.

Table 3.2 Block operation status in each Low power consumption mode

Block		NORMAL	IDLE	STOP1
Processor core		✓	—	—
I/O port	Pin status	✓	✓	✓
	Register	✓	✓	—
ADC		✓	✓	—
UART		✓	✓	—
TSPI		✓	✓	—
PMD+		✓	✓	—
A-ENC32		✓	✓	—
A-VE		✓	✓	—
T32A		✓	✓	—
TRGSEL		✓	✓	—
CRC		✓	✓	—
SIWDT		✓	✓ (Note)	—
LVD		✓	✓	✓
OFD		✓	✓	—
TRM		✓	Unavailable	—
CG		✓	✓	✓
PLL		✓	✓	—
RAM Parity		✓	✓	—
External high speed oscillator (EHOSC)		✓	✓	—
Internal high speed oscillator 1 (IHOSC1)		✓	✓	—
Internal high speed oscillator 2 (IHOSC2)		✓	✓	—
Code Flash		Access Possible	Data hold	Data hold
RAM				

✓: Operation is possible.
—: If it shifts to the object mode, the clock to a peripheral circuit will stop automatically.

Note: It's in the protected mode A only. In other case, Stop SIWDT before shift to IDLE mode.

3.2. Mode State Transition

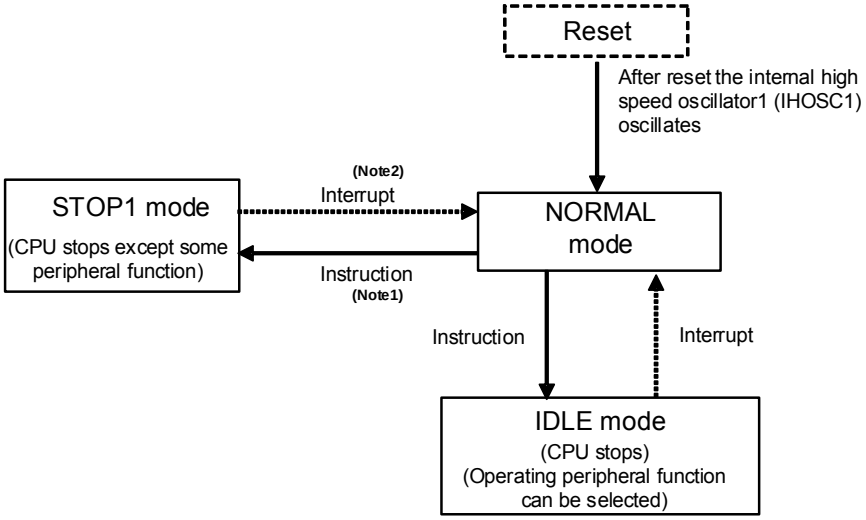

Figure 3.1 Change state

Note 1: Warm up is required at returning. A warm up time must be set in the previous mode (NORMAL mode) before entering to STOP1 mode.

Note 2: When the CPU returns from STOP1 mode, the CPU branches to the interrupt service routine triggered by interrupt events.

3.2.1. IDLE mode Transition flow

Set up the following procedure at switching to IDLE mode.

Because IDLE mode is released by an interrupt, set the interrupt before switching to IDLE mode. For the interrupts that can be used to release the IDLE mode, refer to "3.3.1.The release source of a Low power consumption Mode". Disable interrupts not used for release and interrupts that cannot be used.

Switching procedure		
1	***[SIWDxEN]***<WDTE> =0	Disable SIWDT.
2	***[SIWDxCR]***<WDCR[7:0]> =0xB1	Disable SIWDT.
3	***[FCSR0]***<RDYBSY> is read.	It waits until Flash will be in a Ready state (=1).
4	***[CGSTBYCR]***<STBY[1:0]> =00	Low power consumption mode selection is set to IDLE.
5	***[CGSTBYCR]***<STBY[1:0]> is read.	Check the 4th line register writing (=00).
6	WFI command execution	Switch to IDLE.

3.2.2. STOP1 mode Transition flow

Set up the following procedure at switching to STOP1.

Because STOP1 mode is released by an interrupt, set the interrupt before switching to STOP1 mode. For the interrupts that can be used to release the STOP1 mode, refer to "3.3.1.The release source of a Low power consumption Mode". Disable interrupts not used for release and interrupts that cannot be used.

Switching procedure (from NORMAL mode)		
1	***[SIWDxEN]***<WDTE> =0	Disable SIWDT
2	***[SIWDxCR]***<WDCR[7:0]> =0xB1	Disable SIWDT
3	***[FCSR0]***<RDYBSY> is read.	It waits until Flash will be in a Ready state (=1).
4	***[CGWUPHCR]***<WUEF> is read.	It waits until it becomes the termination of high speed oscillation warming up (=0)
5	***[CGWUPHCR]***<WUCLK> =0	High speed oscillation warming up clock selection is made into an internal high speed oscillator 1(IHOSC1).
	[CGWUPHCR]<WUPT[15:4]> = "arbitrary value"	High speed oscillation warming up counter set value is set as time required for STOP1 restart operation.
6	***[CGSTBYCR]***<STBY[1:0]> =01	Low power consumption mode selection is set to STOP1.
7	***[CGPLL0SEL]***<PLL0SEL> =0	Set PLL of fsys to f_{OSC}(= PLL no USE)
8	***[CGPLL0SEL]***<PLL0ST> is read.	Wait for PLL status of fsys until off state (=0).
9	***[CGPLL0SEL]***<PLL0ON> =0	Stop PLL for fsys
10	***[CGOSCCR]***<IHOSC1EN> =1	Enable the internal high speed oscillator 1.
11	***[CGOSCCR]***<OSCSEL> =0	High speed oscillation selection for f_{OSC} is made into an internal high speed oscillator 1(IHOSC1).
12	***[CGOSCCR]***<OSCF> is read.	It waits until the high speed oscillation selection status for f_{OSC} becomes an inside (IHOSC1) (=0).
13	***[CGOSCCR]***<EOSCEN[1:0]> =00	Selection of an external oscillation of operation is unused.
14	***[CGOSCCR]***<IHOSC2EN> =0	The internal high speed oscillator 2 (IHOSC2) is stopped.
15	***[CGOSCCR]***<EOSCEN[1:0]> is read.	The register writing of above 13th is checked (=00).
16	***[CGOSCCR]***<IHOSC2F> is read.	Wait for flag of IHOSC2 until off "0"
17	WFI command execution	Switch to STOP1.

Note: When using the A mode of SIWDT, 1,2,14 and 16 step is not required.

3.3. The return operation from a Low power consumption mode

3.3.1. The release source of a Low power consumption Mode

Interruption, Non-Maskable Interrupt, and reset can perform release from a Low power consumption mode. The standby release source which can be used is decided by a Low power consumption mode.

It shows the following table about details.

Table 3.3 Release source list

Low power consumption mode			IDLE	STOP1
Release source	Interruption	INT00 to INT07(Note 1)	✓	✓
		INTVCN0, INTVCT0	✓	×
		INTEMG0, INTOVV0, INTPMD0	✓	×
		INTENC00, INTENC01	✓	×
		INTADAPDA, INTADAPDB INTADACP0, INTADACP1, INTADATRG INTADASGL, INTADACNT	✓	×
		INTTxRX, INTTxTX, INTTxERR	✓	×
		INTUARTxRX, INTUARTxTX, INTUARTxERR	✓	×
		INTT32AxA,INTT32AxACAP0,INTT32AxACAP1 INTT32AxB,INTT32AxBCAP0,INTT32BxBCAP1 INTT32AxC,INTT32AxCCAP0,INTT32CxCCAP1	✓	×
		INTPARI	✓	×
		INTFLCRDY	✓	×
	SysTick interrupt		✓	×
	Non-maskable interrupt (INTWDT)		✓ (Note2)	×
	Non-maskable interrupt (INTLVD)		✓	✓
	Reset (SIWDT)		✓ (Note2)	×
	Reset (LVD)		✓	✓
	Reset (OFD)		✓	×
	Reset (RESET_N pin)		✓	✓

✓: After release an interrupt processing will start.

×: It cannot be used for release.

Note 1: INT00 to INT07(External Interrupt 00 to 07) can select one of falling edge, rising edge and level. For details, please refer to "Exception" of reference manual.

Note2: It's in the protect mode A only. In other case, stop SIWDT before shift to IDLE mode.

- Releasing by interrupt request

  When interrupt cancels a Low power consumption mode, it is necessary to prepare so that interrupt may be detected by CPU. The interrupt used for release in the STOP1 mode need to interrupt by INTIF other than setup of CPU, and needs to set up detection.

- Released by Non-Maskable Interrupt (NMI)

  The factor of NMIs are WDT interrupt (INTWDT, protected mode A only) and LVD interrupt (INTLVD).

- Released by reset

  The reset can perform release from all the Low power consumption modes.
  When released by reset, all the registers will be initialized in NORMAL mode after release.

- Released by SysTick interrupt

  SysTick interrupt is available only in IDLE mode.
  Refer to "Interrupts" chapter of a reference manual of "Exception" about the details of interrupt.

3.3.2. Warming up at the release of Low power consumption Mode

Warming up may be required because of stability of an internal oscillator at the time of mode transition.
When moving from STOP1 mode to a NORMAL mode, an internal oscillation is chosen automatically and the warming up counter is started. The Output of a system clock is started after warming up time progress.

For this reason, before executing the command which transition to the STOP1 mode, set up warming up time by ***[CGWUPHCR]***<WUPT[15:4]>. For the setting method, refer to "2.4.1 The warming up counter for a high speed oscillation".

The following table shows the existence of a warming up setup at the time of each operation mode transition.

Table 3.4 Warming up

Operation mode transition	Warming up setup
NORMAL → IDLE	Not required
NORMAL → STOP1	Not required
IDLE → NORMAL	Not required
STOP1 → NORMAL	Required

3.4. Clock Operation in Mode Transition

The clock operation in case of mode transition is shown below.

3.4.1. NORMAL >>> IDLE >>> NORMAL Operation mode transition

CPU stops at IDLE mode. The clock supply to a peripheral function holds a setting state. Please perform operation/stop by the register of each peripheral function, a clock supply setting function, etc. if needed. Execution of Warming up operation is not performed at the time of the restart operation in NORMAL mode from IDLE state.

After the command (WFI) execution which switch to IDLE mode, a program counter will show the next point and will be in a CPU idle state. With a release source, it becomes a CPU reboot and, in the case of an enable interrupt state, the shift to next point by transition command (WFI) will be done, after the interrupt processing by release source.

3.4.2. NORMAL >>> STOP1 >>> NORMAL Operation mode transition

When returning to NORMAL mode from the STOP1 mode, warming up is started automatically.

Please set warming up time (24µs, Min.) to ***[CGWUPHCR]***<WUPT[15:4]> before moving to the STOP1 mode.

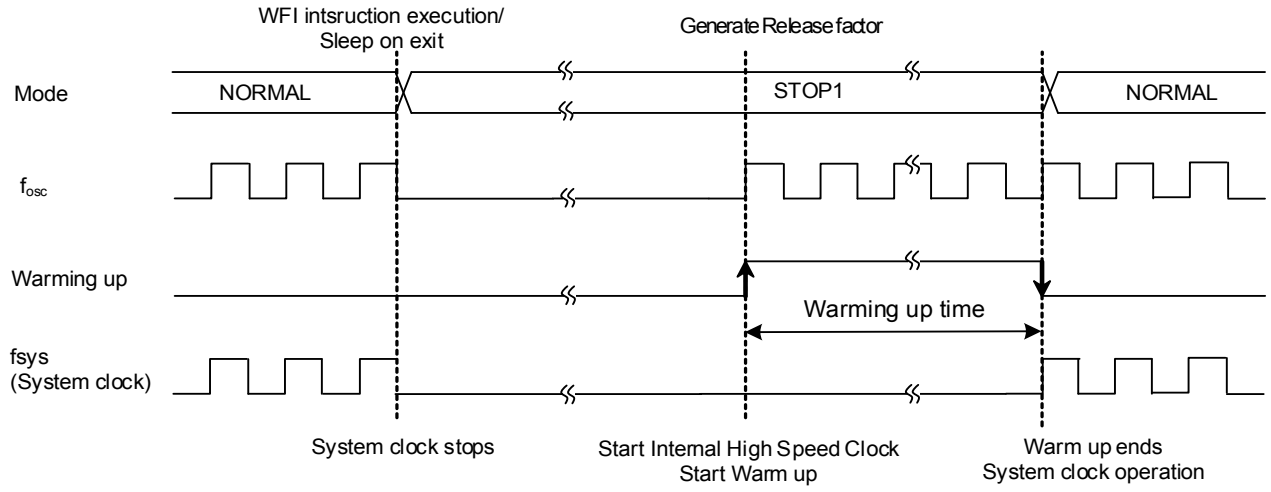

Figure 3.2 NORMAL >>> STOP1 >>> NORMAL Operation mode transition

4. Registers

4.1. Register List

The register related to CG and its address information are shown below.

Function		Channel/Unit	Base Address
Clock Control and Operation Mode	CG	—	0x40083000

Register name		Address (Base+)
CG write protection register	***[CGPROTECT]***	0x0000
Oscillation control register	***[CGOSCCR]***	0x0004
System clock control register	***[CGSYSCR]***	0x0008
Standby control register	***[CGSTBYCR]***	0x000C
PLL selection register for fsys	***[CGPLL0SEL]***	0x0020
High speed oscillation warming up register	***[CGWUPHCR]***	0x0030
Clock supply and stop register A for fsys	***[CGFSYSENA]***	0x0050
Clock supply and stop register for fc	***[CGFCEN]***	0x0058
Clock supply and stop register for ADC	***[CGSPCLKEN]***	0x005C

4.2. Details of Registers

4.2.1. *[CGPROTECT]* (CG write protection register)

Bit	Bit Symbol	After reset	Type	Functions
31:8	—	0	R	Read as "0"
7:0	PROTECT[7:0]	0xC1	R/W	Control write-protection for the CG register (all registers included except this register) 0xC1: CG Registers are write-enabled.(No Protection) Other than 0xC1: Sets write protection.(Protection enable)

4.2.2. *[CGOSCCR]* (Oscillation control register)

Bit	Bit Symbol	After reset	Type	Functions
31:20	—	0	R	Read as "0"
19	IHOSC2F	0	R	Indicates the stability flag of internal oscillation for IHOSC2. 0: Stopping or being in warm up 1: Stable oscillation
18:17	—	0	R	Read as "0"
16	IHOSC1F	1	R	Indicates the stability flag of internal oscillation for IHOSC1. 0: Stopping or being in warm up 1: Stable oscillation
15:13	—	0	R	Read as "0"
12	—	0	R/W	Write as "0"
11:10	—	0	R	Read as "0"
9	OSCF	0	R	Indicates high speed oscillator for f_{OSC} selection status. 0: Internal high speed oscillator 1(IHOSC1) 1: External high speed oscillator (EHOSC)
8	OSCSEL	0	R/W	Selects a high speed oscillation for f_{OSC}. (Note 1) 0: Internal high speed oscillator 1(IHOSC1) 1: External high speed oscillator (EHOSC)
7:4	—	0	R	Read as "0"
3	IHOSC2EN	0	R/W	Enables the internal high speed oscillator 2. (IHOSC2)(Note 2) 0: Stop 1: Oscillation
2:1	EOSCEN[1:0]	00	R/W	Selects the operation of the external high speed oscillator. (EHOSC)(Note 3) 00: External oscillator is not used. 01: Uses the external high speed oscillator. (EHOSC) 10: Uses the external clock. (EHCLKIN) 11: Reserved
0	IHOSC1EN	1	R/W	Internal high speed oscillator 1 (IHOSC1) 0: Stop 1: Oscillation

Note 1: When the setting is modified, confirm whether the written value has been reflected to the ***[CGOSCCR]*** <OSCF> bit before executing the next operation.

Note 2: Setting cannot be changed, when it is ***[SIWDxOSCCR]***<OSCPRO> =1 (Write protect of SIWDT is effective)

Note 3: When an external high speed clock (oscillator connection) is used, set "01" to this bit.

4.2.3. *[CGSYSCR]* (System clock control register)

Bit	Bit Symbol	After reset	Type	Functions
31:28	—	0	R	Read as "0"
27:24	PRCKST[3:0]	0000	R	Indicates a prescaler clock (ΦT0) selection. 0000: fc 0100: fc/16 1000: fc/256 0001: fc/2 0101: fc/32 1001: fc/512 0010: fc/4 0110: fc/64 1010 to 1111: Reserved 0011: fc/8 0111: fc/128
23:19	—	0	R	Read as "0"
18:16	GEARST[2:0]	000	R	Indicates selection status of the gear ratio of the system clock (fsys). 000: fc 100: fc/16 001: fc/2 101 to 111: Reserved 010: fc/4 011: fc/8
15:12	—	0	R	Read as "0"
11:8	PRCK[3:0]	0000	R/W	Selects a prescaler clock (ΦT0). 0000: fc 0100: fc/16 1000: fc/256 0001: fc/2 0101: fc/32 1001: fc/512 0010: fc/4 0110: fc/64 1010 to 1111: Reserved 0011: fc/8 0111: fc/128 Selects a prescaler clock for the peripheral functions.
7:3	—	0	R	Read as "0"
2:0	GEAR[2:0]	000	R/W	Selects a gear ratio of the system clock (fsys). 000: fc 100: fc/16 001: fc/2 101 to 111: Reserved 010: fc/4 011: fc/8

4.2.4. *[CGSTBYCR]* (Standby control register)

Bit	Bit Symbol	After reset	Type	Functions
31:2	—	0	R	Read as "0"
1:0	STBY[1:0]	00	R/W	Selects a Low power consumption mode. 00: IDLE 01: STOP1 10: Reserved 11: Reserved

4.2.5. *[CGPLL0SEL]* (PLL Selection register for fsys)

Bit	Bit Symbol	After reset	Type	Functions
31:8	PLL0SET[23:0]	0x000000	R/W	PLL0 multiplication setup. About a multiplication setup, refer to "2.5.2 The formula and the example of a setting of a PLL multiplication value
7:3	—	0	R	Read as "0"
2	PLL0ST	0	R	Indicates PLL selection status for fsys. 0: f_{OSC} 1: f_{PLL}
1	PLL0SEL	0	R/W	Indicates clock selection for fsys. 0: f_{OSC} 1: f_{PLL}
0	PLL0ON	0	R/W	Indicates PLL operation for fsys. 0: Stop 1: Oscillation

4.2.6. *[CGWUPHCR]* (High speed oscillation warming up register)

Bit	Bit Symbol	After reset	Type	Functions
31:20	WUPT[15:4]	0x800	R/W	Sets the upper 12 bits of the 16 bits of calculation values of the warm up timer. About a setup of a warming up timer, refer to "2.4.1 The warming up counter for a high speed oscillation".
19:16	WUPT[3:0]	0x0	R	Sets the lower 4 bits of the 16 bits of calculation values of the warm up timer. it is fixed by 0x0.
15:9	—	0	R	Read as "0"
8	WUCLK	0	R/W	Warming up clock selection. (Note 1) 0: Internal high speed oscillator 1(IHOSC1) 1: External high speed oscillator (EHOSC)
7:2	—	0	R	Read as "0"
1	WUEF	0	R	Indicates status of the warming up timer. (Note 2) 0: The end of warming up 1: In warming up operation
0	WUON	0	W	Control the warming up timer. 0: Don’t care 1: Warming up operation start.

Note 1: Use the internal oscillator for warm up when the CPU returns from STOP1 mode.
Do not use an external oscillator when the CPU returns from STOP1 mode.

Note 2: Do not modify the registers during the warm up (<WUEF> =1). Set the registers when <WUEF> =0.

4.2.7. *[CGFSYSENA]* (Clock supply and stop register A for fsys)

Bit	Bit Symbol	After reset	Type	Functions
31	IPENA31	1	R/W	Enables the clock of SIWDT 0: Clock stop 1: Clock supply
30	IPENA30	1	R/W	Write as "1
29	IPENA29	0	R/W	Write as "0"
28	IPENA28	0	R/W	Enables the clock of OFD 0: Clock stop 1: Clock supply
27	IPENA27	0	R/W	Enables the clock of TRM 0: Clock stop 1: Clock supply
26	IPENA26	0	R/W	Enables the clock of TRGSEL 0: Clock stop 1: Clock supply
25	IPENA25	0	R/W	Enables the clock of PMD+ 0: Clock stop 1: Clock supply
24	IPENA24	0	R/W	Enables the clock of A-ENC32 0: Clock stop 1: Clock supply
23	IPENA23	0	R/W	Enables the clock of A-VE 0: Clock stop 1: Clock supply
22	IPENA22	0	R/W	Enables the clock of RAMP 0: Clock stop 1: Clock supply
21	IPENA21	0	R/W	Enables the clock of CRC 0: Clock stop 1: Clock supply
20	IPENA20	0	R/W	Enables the clock of TSPI ch2 0: Clock stop 1: Clock supply
19	IPENA19	0	R/W	Enables the clock of TSPI ch1 0: Clock stop 1: Clock supply
18	IPENA18	0	R/W	Enables the clock of TSPI ch0 0: Clock stop 1: Clock supply
17	IPENA17	0	R/W	Enables the clock of UART ch2 0: Clock stop 1: Clock supply
16	IPENA16	0	R/W	Enables the clock of UART ch1 0: Clock stop 1: Clock supply
15	IPENA15	1	R/W	Enables the clock of UART ch0 0: Clock stop 1: Clock supply
14	IPENA14	0	R/W	Enables the clock of ADC 0: Clock stop 1: Clock supply
13	IPENA13	0	R/W	Enables the clock of T32A ch03 0: Clock stop 1: Clock supply

Bit	Bit Symbol	After reset	Type	Functions
12	IPENA12	0	R/W	Enables the clock of T32A ch02 0: Clock stop 1: Clock supply
11	IPENA11	0	R/W	Enables the clock of T32A ch01 0: Clock stop 1: Clock supply
10	IPENA10	1	R/W	Enables the clock of T32A ch00 0: Clock stop 1: Clock supply
9	IPENA09	0	R/W	Enables the clock of PORT K 0: Clock stop 1: Clock supply
8	IPENA08	0	R/W	Enables the clock of PORT J 0: Clock stop 1: Clock supply
7	IPENA07	0	R/W	Enables the clock of PORT H 0: Clock stop 1: Clock supply
6	IPENA06	0	R/W	Enables the clock of PORT G 0: Clock stop 1: Clock supply
5	IPENA05	0	R/W	Enables the clock of PORT F 0: Clock stop 1: Clock supply
4	IPENA04	0	R/W	Enables the clock of PORT E 0: Clock stop 1: Clock supply
3	IPENA03	0	R/W	Enables the clock of PORT D 0: Clock stop 1: Clock supply
2	IPENA02	1	R/W	Enables the clock of PORT C 0: Clock stop 1: Clock supply
1	IPENA01	0	R/W	Enables the clock of PORT B 0: Clock stop 1: Clock supply
0	IPENA00	0	R/W	Enables the clock of PORT A 0: Clock stop 1: Clock supply

Note: Even if the initial value of a register is set to stop of the clock, the clock is supplied to the register during the reset.

4.2.8. *[CGFCEN]* (Clock supply and stop register for fc)

Bit	Bit Symbol	After reset	Type	Functions
31:2	—	0	R	Read as "0"
1	FCIPEN01	0	R/W	Enables the clock of DNF UnitA 0: Clock stop 1: Clock supply
0	FCIPEN00	0	R/W	Enables the clock of OFD(Note) 0: Clock stop 1: Clock supply

Note: When use the monitor clock of fc,***[CGFSYSENA]***<IPENA28> and ***[CGFCEN]***<FCIPEN00> should be enabled.

4.2.9. *[CGSPCLKEN]* (Clock supply and stop register for ADC)

Bit	Bit Symbol	After reset	Type	Functions
31:17	—	0	R	Read as "0"
16	ADCKEN	0	R/W	Enable the clock for ADC. 0: Clock stop 1: Clock supply
15:1	—	0	R	Read as "0"
0	—	0	R/W	Write as "0"

5. Information according to product

The information about ***[CGFSYSENA]*** and ***[CGFCEN]*** which is different according to each product is shown below.

5.1. *[CGFSYSENA]*

Table 5.1 *[CGFSYSENA]* per product allocation

Bit	Bit Symbol	Internal connection peripheral circuit	Channel No. / Unit name Port name	M4L2	M4L1
31	IPENA31	SIWDT	—	✓	✓
30	IPENA30	—	—	—	—
29	IPENA29	—	—	—	—
28	IPENA28	OFD	—	✓	✓
27	IPENA27	TRM	—	✓	✓
26	IPENA26	TRGSEL	0	✓	✓
25	IPENA25	PMD+	0	✓	✓
24	IPENA24	A-ENC32	0	✓	✓
23	IPENA23	A-VE	0	✓	✓
22	IPENA22	RAMP	0	✓	✓
21	IPENA21	CRC	—	✓	✓
20	IPENA20	TSPI	2	✓	✓
19	IPENA19		1	✓	✓
18	IPENA18		0	✓	✓
17	IPENA17	UART	2	✓	✓
16	IPENA16		1	✓	✓
15	IPENA15		0	✓	✓
14	IPENA14	ADC	A	✓	✓
13	IPENA13	T32A	3	✓	✓
12	IPENA12		2	✓	✓
11	IPENA11		1	✓	✓
10	IPENA10		0	✓	✓
9	IPENA09	PORT	K	✓	—
8	IPENA08		J	✓	✓
7	IPENA07		H	✓	✓
6	IPENA06		G	✓	✓
5	IPENA05		F	✓	✓
4	IPENA04		E	✓	✓
3	IPENA03		D	✓	✓
2	IPENA02		C	✓	✓
1	IPENA01		B	✓	✓
0	IPENA00		A	✓	✓

Note: ✓: Available, — : N/A

5.2. *[CGFCENA]*

Table 5.2 *[CGFCENA]* per product allocation

Bit	Bit Symbol	Internal connection peripheral circuit	Channel No. / Unit name Port name	M4L2	M4L1
1	FCIPENA01	DNF	A	✓	✓
0	FCIPENA00	OFD	—	✓	✓

Note: ✓: Available, — : N/A

6. Revision History

Table 6.1 Revision History

Revision	Date	Description
1.0	2018-09-07	・First release

RESTRICTIONS ON PRODUCT USE

Toshiba Corporation and its subsidiaries and affiliates are collectively referred to as “TOSHIBA”.
Hardware, software and systems described in this document are collectively referred to as “Product”.

- TOSHIBA reserves the right to make changes to the information in this document and related Product without notice.
- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA's written permission, reproduction is permissible only if reproduction is without alteration/omission.
- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application with which the Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. **TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS.**
- **PRODUCT IS NEITHER INTENDED NOR WARRANTED FOR USE IN EQUIPMENTS OR SYSTEMS THAT REQUIRE EXTRAORDINARILY HIGH LEVELS OF QUALITY AND/OR RELIABILITY, AND/OR A MALFUNCTION OR FAILURE OF WHICH MAY CAUSE LOSS OF HUMAN LIFE, BODILY INJURY, SERIOUS PROPERTY DAMAGE AND/OR SERIOUS PUBLIC IMPACT ("UNINTENDED USE").** Except for specific applications as expressly stated in this document, Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, lifesaving and/or life supporting medical equipment, equipment used for automobiles, trains, ships and other transportation, traffic signaling equipment, equipment used to control combustions or explosions, safety devices, elevators and escalators, and devices related to power plant. **IF YOU USE PRODUCT FOR UNINTENDED USE, TOSHIBA ASSUMES NO LIABILITY FOR PRODUCT.** For details, please contact your TOSHIBA sales representative or contact us via our website.
- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.
- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.
- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**
- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the applicable export laws and regulations including, without limitation, the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.
- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. **TOSHIBA ASSUMES NO LIABILITY FOR DAMAGES OR LOSSES OCCURRING AS A RESULT OF NONCOMPLIANCE WITH APPLICABLE LAWS AND REGULATIONS.**