



The following document contains information on Cypress products. The document has the series name, product name, and ordering part numbering with the prefix “MB”. However, Cypress will offer these products to new and existing customers with the series name, product name, and ordering part number with the prefix “CY”.

How to Check the Ordering Part Number

1. Go to www.cypress.com/pcn.
2. Enter the keyword (for example, ordering part number) in the **SEARCH PCNS** field and click **Apply**.
3. Click the corresponding title from the search results.
4. Download the Affected Parts List file, which has details of all changes

For More Information

Please contact your local sales office for additional information about Cypress products and solutions.

About Cypress

Cypress is the leader in advanced embedded system solutions for the world's most innovative automotive, industrial, smart home appliances, consumer electronics and medical products. Cypress' microcontrollers, analog ICs, wireless and USB-based connectivity solutions and reliable, high-performance memories help engineers design differentiated products and get them to market first. Cypress is committed to providing customers with the best support and development resources on the planet enabling them to disrupt markets by creating new product categories in record time. To learn more, go to www.cypress.com.

Traveo™ Family, MB9D560 Setting Clock System

Associated Part Family:	Series Name	Product Number
	MB9D560	MB9DF564MAE/F565MAE/F566MAE MB9DF564MGE/F565MGE/F566MGE

This application note is intended for persons who are considering the use of Traveo family MB9D560 and describes how to set up the clock system when it operated at maximum frequency (200MHz) to MB9D560.

Contents

1	Introduction.....	1	3.2	System Clock Setting.....	6
1.1	Environment of Development.....	1	4	Reference.....	14
2	Clock System.....	1	5	Document History.....	15
3	System Clock Change Operating	3			
3.1	The Wait Cycle Count of TCFLASH.....	4			

1 Introduction

This application note is intended for persons who are considering the use of Traveo family MB9D560.

It is necessary to use a specific procedure to set the clock system of MB9D560.

This application note describes how to set up the clock system when it operated at maximum frequency (200MHz) to MB9D560.

1.1 Environment of Development

Contents is described in this application note has been developed in the environment shown in Table 1.

Table 1. Environment of Development

Microcomputer	MB9DF566MGB
Integrated Development Environments	MULTI v6.1.4
Evaluation Board	MB2198-770-02-E0
Optimization	Optimize for Speed

2 Clock System

The clock system supplies the clocks for MCU operation.

The external/built-in oscillating clocks of the MCU are generically called source clocks. A source clock is a clock used to generate an internal operating clock for MCU operation.

The clock system generates the following source clocks.

- Fast-CR clock
- Slow-CR clock
- Main clock/Main clock divided by 2
- PLL clock

External/built-in oscillation circuits generate a Fast-CR clock, Slow-CR clock and Main clock. A Main clock divided by 2 is generated by the division of the main clock.

And the main clock and the built-in PLL oscillation circuits generate a PLL clock.

When using the main clock and PLL clock can monitor clock by clock supervisor.

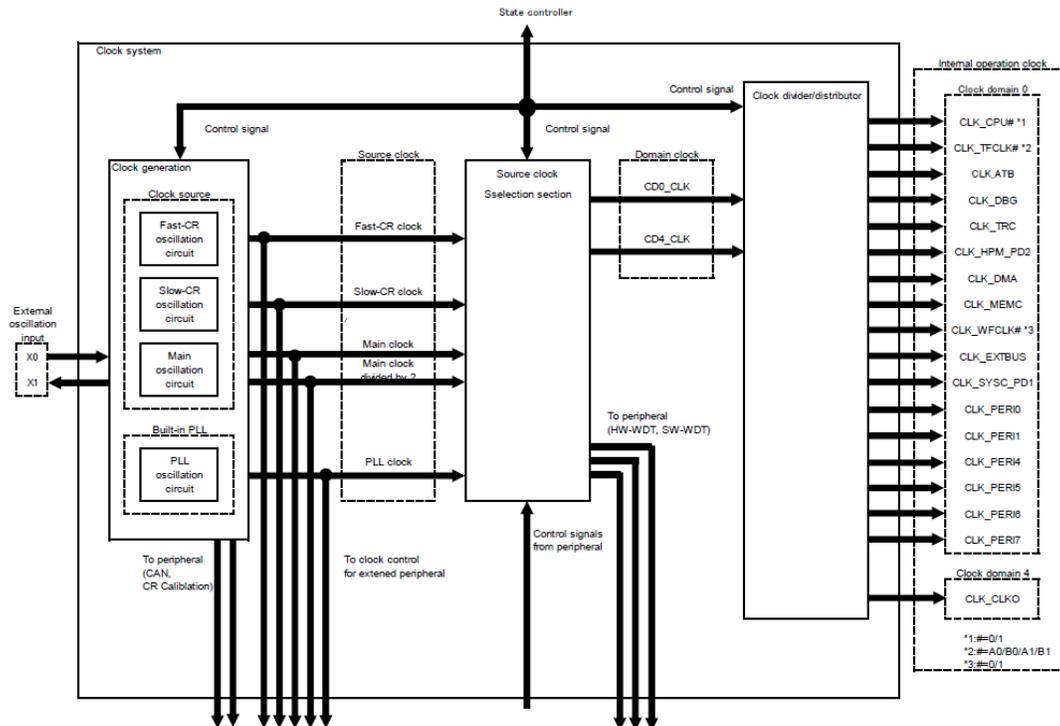
An area that uses internal operating clocks generated from the same source clock is called a clock domain. A clock domain contains 1 or more internal operating clocks.

The clock system supplies clocks to the following clock domains.

- Clock domain 0 (system area)
- Clock domain 4 (external output clock area)

The source clock used in each clock domain can be selected. In each clock domain, the selected source clock is divided, and the resulting clocks are distributed within the MCU as internal operating clocks.

Figure 1. Clock System Block Diagram

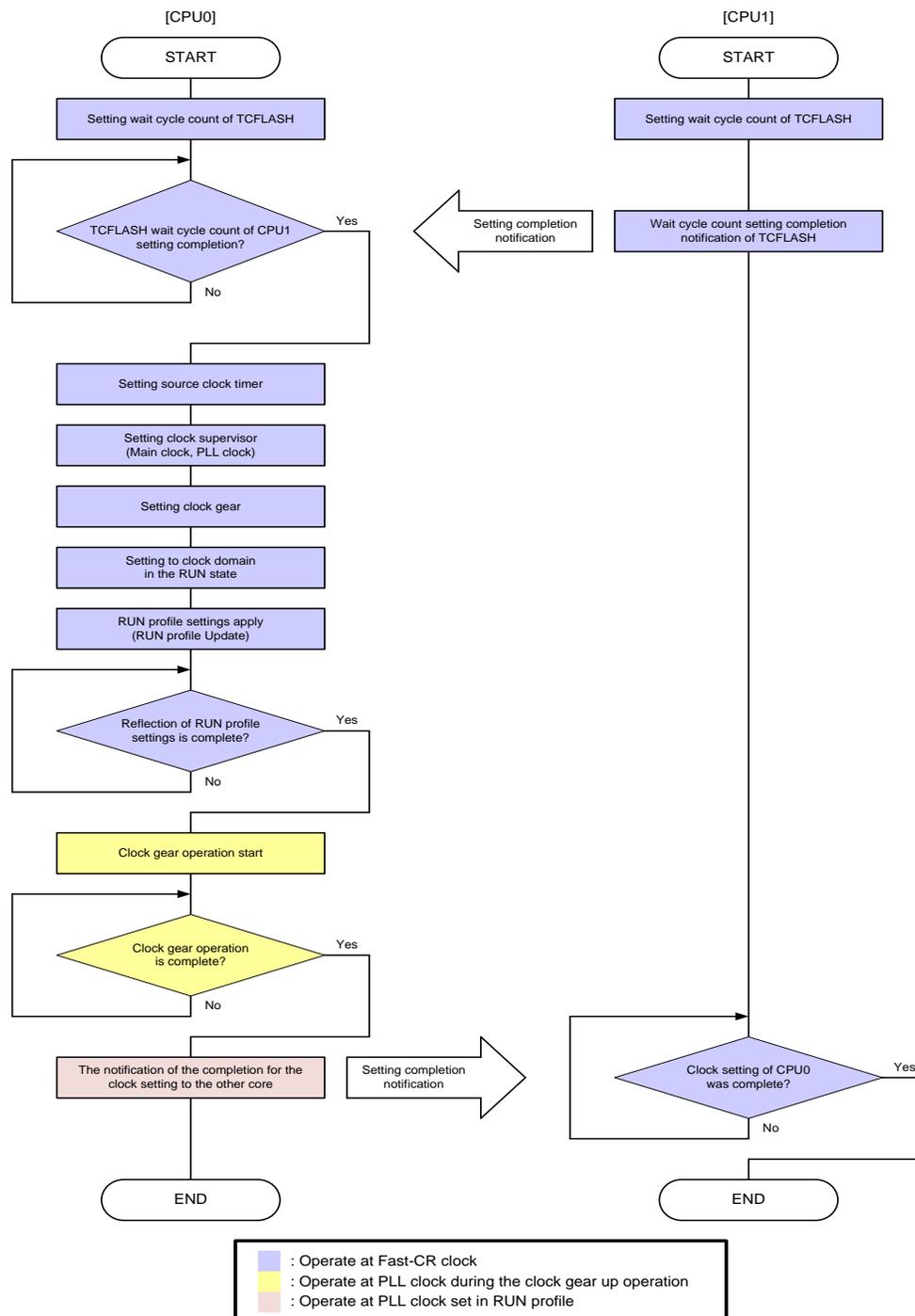


3 System Clock Change Operating

This is the procedure to change the system clock of RUN mode in Figure 2.

If it makes a change of system clock, it must be synchronized between the cores. See the application note of caution of 2CPU mode.

Figure 2. System Clock Change Procedure



3.1 The Wait Cycle Count of TCFLASH

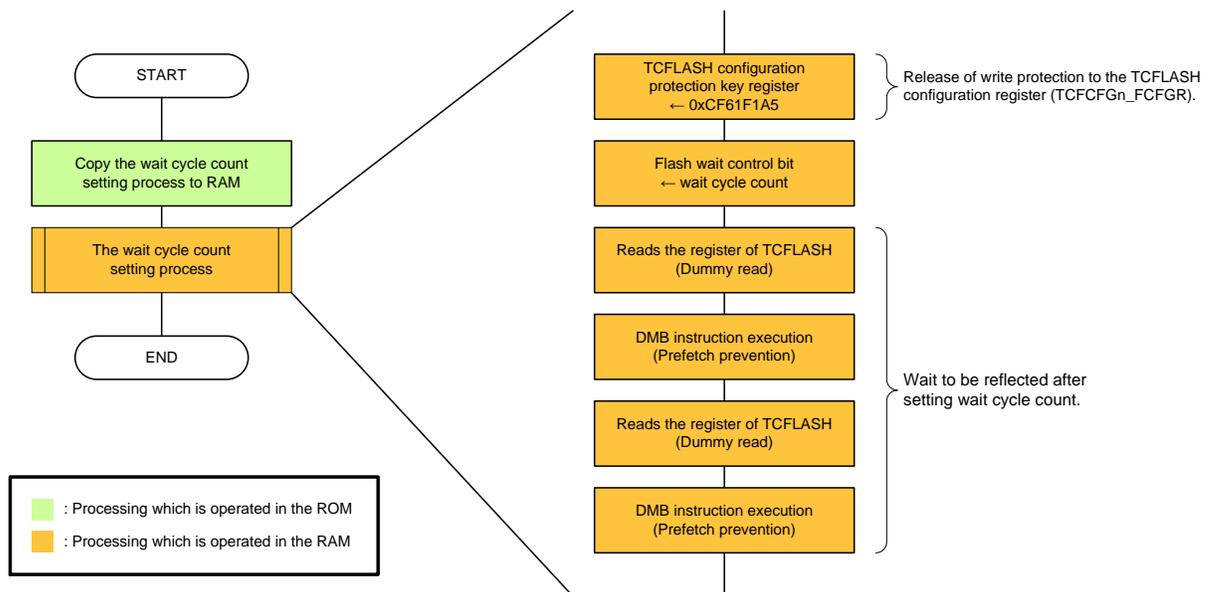
If the operating frequency of the system is higher than the maximum operating frequency of the flash memory (80MHz), it is necessary to insert wait cycles when accessing the flash memory. Wait cycles can be inserted by setting an appropriate value in FAWC [1:0] in the TCFCFGn_FCFGR register.

The procedure is described in Figure 3.

Two TCFLASH is equipped in this microcomputer, it is necessary to set wait cycle count of TCFLASH connected with each core.

It is necessary to set wait cycle count before the change of system clock.

Figure 3. TCFLASH Wait Cycle Count Change Procedure



3.1.1 Advance Preparation of Wait Cycle Configuration

Access to TCFLASH (Data readout or program execution) is prohibited during setting wait cycle count. Therefore, the wait cycle count setting process is necessary to operate in RAM. It describes the procedure to operate wait cycle count setting process in RAM in the following.

1. Prepare a section of wait cycle count setting process in ROM and RAM.
2. Implement the wait cycle count setting process while specifying the sections (RAM) that was created in step 1.
3. Implement the process of copying the data of the section (ROM) that was created in step 1 to section that you created in step 1 (RAM).
4. Implement a process to run the copied wait cycle count setting process in section that you created in step 3 (RAM) after executing the processes that have been implemented in step 3

The example is described in Figure 4, Figure 5.

Figure 4. Configuration Example of TCFASH Wait Cycle Count (CPU0)

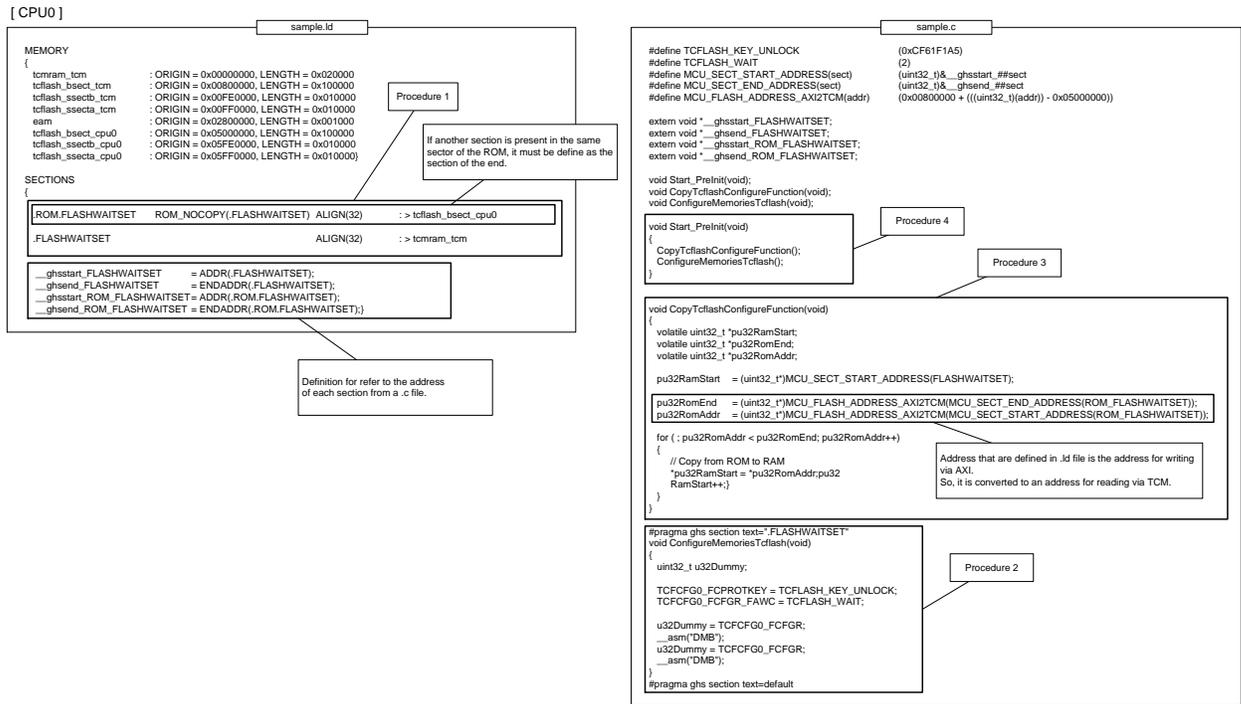
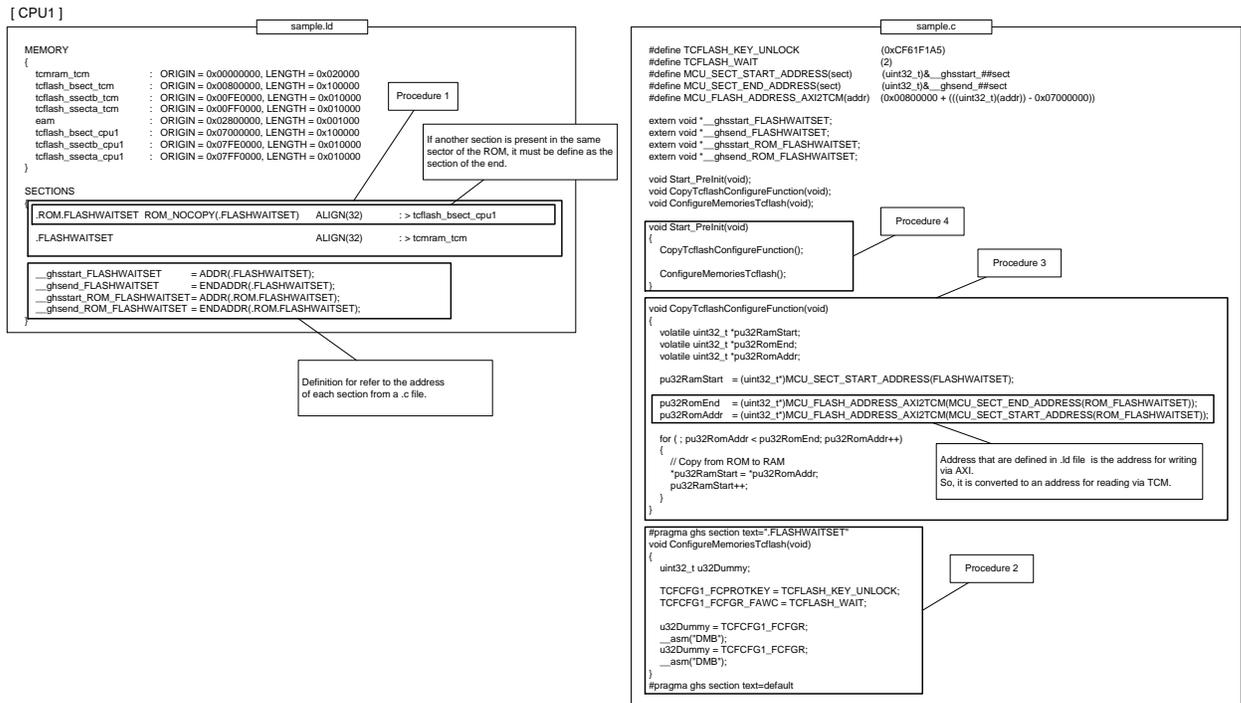


Figure 5. Configuration Example of TCFASH Wait Cycle Count (CPU1)



3.1.2 The Wait Cycle Count Calculation

The wait cycle count is calculated from the following equation.

$(\text{System operating frequency} / \text{Maximum operating frequency of TCFLASH}) - 1 = \text{Wait cycle count} (*)$

If the calculated result is a decimal, integer values are determined by rounding up the decimal part.

Ex.) System operating frequency: 200MHz, Maximum operating frequency of TCFLASH: 80MHz

$(\text{System operating frequency} / \text{Maximum operating frequency of TCFLASH}) - 1$

$= 200 / 80 - 1$

$= 3 - 1$

$= 2$

3.2 System Clock Setting

This microcontroller operates in the Fast-CR clock after power-on. It performs the setting mentioned in this chapter for changing frequency of the system clock. After that, it is necessary to reflect the setting to the RUN profile. After RUN profile settings, the frequency of the system clock change gradually over a number of steps that is set by operating the clock gear.

All cores sets do not access the peripherals during the frequency change of the system clock. It may cause a malfunction.

All registers that are described in this chapter is locked the writing. It is necessary to perform after unlock to write the unlock code (0x5CACCE55) to the protection key setting register (SYSC_PROTKEYR) when writing to the register. Every time unlock is required when performing the writing.

3.2.1 Source Clock Timer

The source clock timer is for gating the clock output until the end of the clock oscillation stabilization wait time. In order to use the source clock timer, it is necessary to set Table 2 register. Example of this setting is described in Table 2 when changing the system clock to the PLL clock (200MHz) from the high-speed CR clock.

Table 2. Example of Source Clock Timer Setting (PLL clock 200MHz when used)

Register Abbreviation	Bit name	Settings	Value
SYSC_MOCTCPR *1	PSCL	Select the division ratio of the input clock of the main clock timer.	0110(b) *3
	CMPR *2	The compare value of the main clock timer.	1D4C(h) *3
SYSC_MOCTTRGR *4	CGCPT	Change the timer settings / Start the timer counting.	1(b)
SYSC_PLLSTCNTR	PLLSTABS *5	PLL stabilization wait time selection.	1111(b)

- Setting is reflected by writing a "1" to CGCPT of SYSC_MOCTTRGR.
It is necessary to complete the setting before writing "1" to CGCPT of SYSC_MOCTTRGR.
- It is prohibited to be changed during timer operation.
- The main clock is the setting that assumes the 16MHz.
- Bits which are not described are not required change.
- Do not change setting after PLL clock oscillation is set to enable

3.2.2 Clock Supervisor

If you use the clock supervisor, the clock supervisor can detect the stopped clocks and occurrences of frequency-range abnormalities. To detect the abnormal using the clocks of 2 systems are used: one clock becomes the monitoring target, and the other clock is a reference clock for measuring a certain time period.

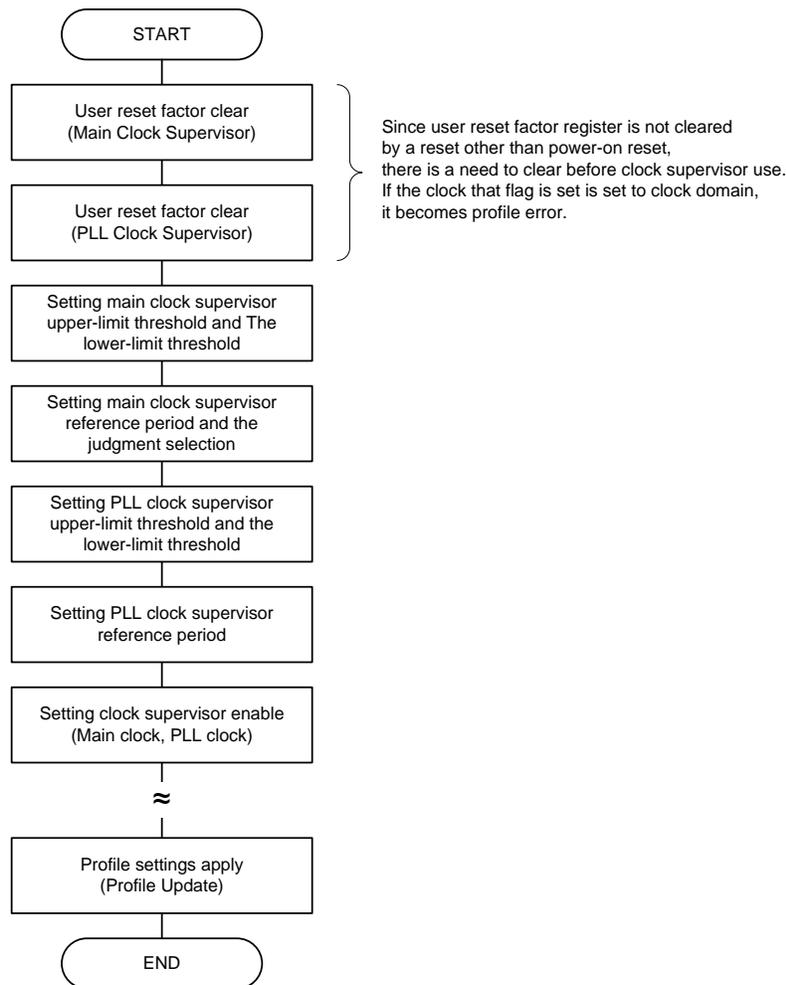
The clock supervisor is equipped with the following supervisors. Each of them is independent and can be separately set to enabled or disabled.

- Main clock supervisor
- PLL clock supervisor
- Sub-system PLL clock supervisor (Flex Ray/RDC PLL)

This application note describes how to use main clock supervisor and PLL clock supervisor. It refers to the hardware manual when using sub-system PLL clock supervisor.

Setting procedure is described in Figure 6 when using the Main clock supervisor and PLL clock supervisor. Clock supervisor start operating after the oscillation stabilization wait of clock to be monitored.

Figure 6. Clock Supervisor Setting Procedure



Clock Supervisor Setting Example

In order to use clock supervisor, it is necessary to set the register of Table 3. The example is described in Table 3 when using clock supervisor under the following conditions.

It is not include setting at the time of PSS state in Table 3. The reference clock of the main clock supervisor at the time of PSS is possible to select the Slow-CR clock. Method of using Slow-CR clock as a reference clock refers to the hardware manual.

- Clock supervisor to be used : Main clock supervisor, PLL clock supervisor
- Main clock supervisor use conditions
 - Main clock frequency : 16MHz
 - Fast-CR clock frequency : 6~10MHz (typ : 8MHz)
 - Acceptable main clock frequency range : 4~20MHz
 - Reference clock count period : 60
- PLL clock supervisor use conditions
 - PLL clock frequency : 200MHz
 - Main clock frequency : 8MHz
 - Acceptable PLL clock frequency range : 80~500MHz
 - Reference clock count period : 12

Table 3. Clock Supervisor Setting Example (Main clock/PLL clock when used)

Register Abbreviation	Bit name	Settings	Value (decimal)
SYSC_CSMOCFGR0 *1、*2	UPTHR	Frequency range upper-limit threshold value of Main clock (Reference clock : Fast-CR clock)	200 *3
	LOWTHR	Frequency range lower-limit threshold value of Main clock (Reference clock : Fast-CR clock)	24 *3
SYSC_CSMOCFGR01 *1	JDGSEL	Selection of Operation at abnormal state detection of Main clock (Reference clock : Fast-CR clock)	0 *4
	REFCLKWND	Reference clock count period of Main clock supervisor (Reference clock : Fast-CR clock)	60
SYSC_CSVPLLCFGR0 *2、*5	UPTHR	Frequency range upper-limit threshold value of PLL clock (Reference clock : Main clock)	375 *3
	LOWTHR	Frequency range lower-limit threshold value of PLL clock (Reference clock : Main clock)	60 *3
SYSC_CSVPLLCFGR1 *5	REFCLKWND	Reference clock count period of PLL clock supervisor (Reference clock : Main clock)	12
SYSC_RUNCSVCFGR	PLLCSSVE	Enable the PLL clock supervisor	1
	MOCSVE	Enable the main clock supervisor	1

1. This register must not be modified during operation of the main clock supervisor.
2. Set the upper-limit threshold value bit value to be greater than the lower-limit threshold value bit value.
3. Calculation method of the threshold value refer to “Calculation of the Clock Supervisor Threshold Value”.
4. At the abnormal state detection, it generates a reset if the main clock is selected as clock domain 0 or the software watchdog timer. If this is not the case an interrupt is generated.
5. This register must not be modified during operation of the PLL clock supervisor.

Calculation of the Clock Supervisor Threshold Value

Threshold value to be set in the clock supervisor is calculated from the following equation.

Acceptable the maximum value of monitoring target clock frequency / Minimum value of Reference clock frequency x Reference clock count period

= Upper-limit threshold value

Acceptable the minimum value of monitoring target clock frequency / Maximum value of Reference clock frequency x Reference clock count period

= Lower-limit threshold value

If it was used in the conditions in the “0.0.0Clock Supervisor Setting Example”, the setting is following.

- Upper-limit threshold value of main clock supervisor

Acceptable the maximum value of main clock frequency / Minimum value of Fast-CR clock frequency x Reference clock count period

= $20 / 6 \times 60 = 200$

- Lower-limit threshold value of main clock supervisor

Acceptable the minimum value of main clock frequency / Maximum value of Fast-CR clock frequency x Reference clock count period

= $4 / 10 \times 60 = 24$

- Upper-limit threshold value of PLL clock supervisor

Acceptable the maximum value of PLL clock frequency / Minimum value of Main clock frequency x Reference clock count period

= $500 / 16 \times 12 = 375$

- Lower-limit threshold value of PLL clock supervisor

Acceptable the minimum value of PLL clock frequency / Maximum value of Main clock frequency x Reference clock count period

= $80 / 16 \times 12 = 60$

Notes of Clock Supervisor

- Profile settings are used to enable/disable the clock supervisor. Writing to the clock supervisor configuration register alone setting is not reflected. So it must be RUN update.
- If clock supervisor want stopped, it is necessary to stop the oscillation of the monitoring clock.
- There is a profile error, if the clock supervisor reset detection bit of the user reset factor register (SYSC_RSTCAUSEUR) is already set is selected as the clock monitor domain.
- User reset factor register (SYSC_RSTCAUSEUR) is not cleared except during power-on reset. It must clear the clock supervisor reset detection bit before starting clock supervisor operation.
- After a reset generated by the detection of a clock abnormality, the main clock supervisor reference clock returns to the fast-CR clock.

3.2.3 Clock Gear

The frequency of the clock gear can be gradually changed by step-by-step output of the input clock to the clock gear circuit.

The frequency fluctuates abruptly at the time of switching from the main clock to the PLL clock, so the power supply current also fluctuates considerably. A clock gear operation is necessary to prevent any possible occurrence of overshoot/undershoot of the power supply current at the clock switching time.

In order to use the clock gear, it is necessary to set the register of Table 4. The example is described in Table 2 when using clock gear under the following conditions.

- Main clock frequency : 16MHz
- PLL clock frequency : 200MHz
- Loop count : 256times
- Step width : 1
- Start step : 0

Table 4. Clock Gear Setting Example

Register Abbreviation	Bit name	Settings	Value
SYSC_PLLCGNTR	PLLCGLP *1	Loop count per one step of PLL clock gear operation	FF(h)
	PLLCGSTP *1	Step width at the PLL clock gear-up/down	00(b)
	PLLCGSSN *1	Step at the start of PLL clock gear operation	000000(b)
	PLLCGSTR	Start gear operation	01(b) *2
	PLLCGEN *1	Clock gear operation enabled	1(b)

1. Do not change setting after PLL clock oscillation is set to enable.
2. It is necessary to set at the timing of operating the clock gear after application of the profile settings.

Procedure of Clock Gear Operation Start

Clock gear starts the operation by performing the following steps after setting in Table 4 (Except PLLCGSTR). It is necessary to wait until the clock gear is completed after clock gear operation start.

1. Select the PLL clock as the domain clock.
2. Set "01" to PLLCGSTR of PLL clock gear control register (SYSC_PLLCGNTR).
3. Poll the value of the clock gear status flag (PLLCGNTR.PLLCGSTS), and wait until the clock indicates the stop state.

Notes of Clock Gear

- Setting of the clock gear control register (SYSC_PLLCGNTR) must satisfy the condition of the "Source Clock for the Clock Domain 0".
- The smaller step width and the larger loops count, the more gradually the frequency changes.
- When clock gear operation is enabled, after the oscillation stabilization wait time of the clock gear input clock, the clock according to the setting of PLLCGSSN is output. But it does not gear up/gear-down until the clock gear operation start setting is performed.

3.2.4 Clock Domain of RUN Mode

Enable Setting of the Source Clock

A hardware reset results in the source clocks from the external/built-in oscillation circuits (fast-CR circuit clock, slow-CR clock and main clock/main clock divided by 2) entering the oscillation-enabled state. Conversely, the PLL clock enters the oscillation-disabled state. It is necessary to enable the oscillation of the PLL clock when changing the source clock of the system clock to the PLL clock (200MHz). To enable PLL oscillation, the oscillation for the main clock must already be enabled. If the oscillation for the main clock has not been enabled, a profile error occurs.

It can be set to enable the oscillation of the corresponding clock during RUN by writing "1" to the bits of register shown in the Table 5. However, control for enabling/disabling the source clock oscillation is available only when the system is not using the source clock. The fast-CR clock/slow-CR clock oscillation cannot be disabled in RUN.

The example is described in Table 5 when changing the source clock of the system clock to the PLL clock (200MHz).

Table 5. Source Clock Oscillation Enabled Setting Example

Register Abbreviation	Bit name	Settings	Value (decimal)
SYSC_RUNCKSRER	PLEN	Enable PLL clock oscillation	1
	MOSCEN	Enable Main clock oscillation	1
	SCROSCEN	Enable Slow-CR clock oscillation	1
	CROSCEN	Enable Fast-CR clock oscillation	1

PLL Clock Divider

The PLL clock is generated from the main clock. The example is described in Table 6 when generating a PLL clock 200MHz from the main clock 16MHz under the following conditions.

- N-multiplier : Multiply by 25
- M-divider : Divided by 2
- Input clock divider : No division

Table 6. PLL Clock Divider Setting Example

Register Abbreviation	Bit name	Settings	Value (decimal)
SYSC_RUNPLLCNTR	PLLDIVN	PLL clock N-multiplier setting	24
	PLLDIVM	PLL clock M-divider setting	1
	PLLDIVL	PLL input clock divider setting	0

Source Clock for the Clock Domain 0

Hardware reset results in the fast-CR clock being selected as the clock used in all clock domains. It is necessary to set the PLL clock to the source clock of clock domain 0 when changing the source clock of the system clock to the PLL clock (200MHz). Source clock of the system clock become the PLL clock by setting the source clock of clock domain 0 to PLL clock.

Source clock of clock domain 0 during RUN state can be selected by setting the register of Table 7. The example is described in Table 7 when setting the source clock of clock domain 0 to PLL clock.

When switching to PLL clock, be sure to use clock gear. Switch with 50MHz or less and activate gear-up operation

Table 7. Clock Domain 0 Source Clock Setting Example

Register Abbreviation	Bit name	Settings	Value (decimal)
SYSC_RUNCKSELR0	CD0CSL	Clock domain 0 source clock selection	4

Internal Clock Oscillation Enable

Setting internal clock oscillation enable during RUN state can be changed by setting the register of Table 8. However, there are restrictions on the following clock.

- SYSCPD1 internal operation clock: Oscillation disable cannot set.
- HPMPD2 internal operation clock: Oscillation disable cannot set.
- CPU1 internal operation clock: Oscillation disable cannot set.
- CPU0 internal operation clock: Oscillation disable cannot set.
- EXTBUS internal operation clock : Set this bit to "0" due to this microcomputer has no external bus I/F.(*)

Note: Setting of EXTBUS internal operation clock become enabling by default. It is necessary to set the oscillation disable.

The setting example is described in Table 8 when enabling other than EXTBUS internal operation clock.

Table 8. Example of Setting Internal Clock Oscillation Enable

Register Abbreviation	Bit name	Settings	Value (decimal)
SYSC_RUNCKER	ENCLKPERI7	PERI7 internal operation clock oscillation enable	1
	ENCLKPERI6	PERI6 internal operation clock oscillation enable	1
	ENCLKPERI5	PERI5 internal operation clock oscillation enable	1
	ENCLKPERI4	PERI4 internal operation clock oscillation enable	1
	ENCLKPERI1	PERI1 internal operation clock oscillation enable	1
	ENCLKPERI0	PERI0 internal operation clock oscillation enable	1
	ENCLKSYSCPD1	SYSCPD1 internal operation clock oscillation enable	1 *1
	ENCLKEXTBUS	EXTBUS internal operation clock oscillation enable	0 *2
	ENCLKMEMC	MEMC internal operation clock oscillation enable	1
	ENCLKDMA	DMA internal operation clock oscillation enable	1
	ENCLKHPMPD2	HPMPD2 internal operation clock oscillation enable	1 *1
	ENCLKATB	ATB internal operation clock oscillation enable	1
	ENCLKDBG	DBG internal operation clock oscillation enable	1
	ENCLKCPU1	CPU1 internal operation clock oscillation enable	1 *1
ENCLKCPU0	CPU0 internal operation clock oscillation enable	1 *1	

1. "0" writing is invalid.
2. Since the initial value "1" is set, changing always "0".

Internal Clock Divider

Division ratio of the source clock of each internal operation clock during RUN state can be changed by setting the registers of Table 10. However, it is necessary to set the division ratio so that the supplied clock does not exceed the maximum operating frequency of any internal operating clock. The maximum operating frequency is described in Table 9.

Table 9. Maximum Operating Frequencies of Internal Operating Clock

Internal Operating Clock	Maximum Operating Frequency
CLK_HPMPD2	200MHz
CLK_DMA	200MHz
CLK_MEMC	200MHz
TRC clock	100MHz
CLK_ATB	100MHz
DBG clock	50MHz
System clock	200MHz
SYSC_PD1 clock	100MHz
EXTBUS clock	200MHz
PERI7 clock	50MHz
PERI6 clock	50MHz
PERI5 clock	100MHz
PERI4 clock	100MHz
PERI1 clock	50MHz
PERI0 clock	100MHz

The setting example is described in Table 10 when operating at the maximum operating frequency. Source clock of the system clock is the PLL clock (200MHz).

Table 10. Internal Clock Divider Setting Example

Register Abbreviation	Bit name	Settings	Value (decimal)
RUNCKDIVR0	HPMDIV	Division ratio of the HPM clock, DMA clock and MEMC clock from the system clock	0
	TRCDIV	Division ratio of the TRC clock from the system clock	1
	ATBDIV	Division ratio of the ATB clock from the system clock	1 *1
	DBGDIV	Division ratio of the DBG clock from the ATB clock	1 *1
	SYSDIV	Division ratio of the system clock from the source clock	0
RUNCKDIVR1	SYSCPD1DIV	Division ratio of the SYSC_PD1 clock from the HPM clock	1
	EXTBUSDIV	Division ratio of the EXTBUS clock from the HPM clock	1 *2
RUNCKDIVR2	PERI7DIV	Division ratio of the PERI7 clock from the PERI5 clock	1
	PERI6DIV	Division ratio of the PERI6 clock from the PERI4 clock	1
	PERI5DIV	Division ratio of the PERI5 clock from the HPM clock	1
	PERI4DIV	Division ratio of the PERI4 clock from the HPM clock	1
	PERI1DIV	Division ratio of the PERI1 clock from the PERI0 clock	1
	PERI0DIV	Division ratio of the PERI0 clock from the HPM clock	1

1. Write invalid
2. It is not affect to change the settings because this microcontroller not equipped EXTBUS

3.2.5 RUN Profile Settings Update

Setting is not reflected in the writing to the register when changing the settings of RUN / PSS profile setting parameters (*). It is necessary to do the following steps to reflect the setting of RUN profile.

Note: See Hardware Manual.

1. Write the settings to the register change the setting.
2. Clear the RUN profile (main status control) update completion flag at system status flag and interrupt clear register (SYSC_SYSCICLR).
3. Write "0xAB" to the RUN profile update trigger register of the system controller (SYSC_TRGRUNCNTR).
4. Poll the RUN profile update completion flag bit of System Status Register, and to wait until the profile update completed (RUNDFO becomes "1").

If the contents of the new RUN profile do not present any problems, the control circuit will reflect the contents of the profile, as follows.

1. "1" is set in the system status register (SYSC_SYSSTSR: RUNSTS0).
2. The contents of the RUN profile are copied to the APPLIED profile.
3. Reflect the following setting in order
 - Clock oscillation enable/stop (including waiting for oscillation stabilization)
 - Clock supervisor setting changes
 - LVD setting changes
 - Clock operation settings (source clock changes, division, and ON/OFF of each clock source)
 - Clock stop settings (source clock stop)
4. When RUN profile update is completed, the system status register (SYSC_SYSSTSR: RUNSTS0) is cleared to "0".
5. "1" is set in the system status register (SYSC_SYSSTSR: RUNDFO).

If present setting is problems, Profile error occurs, system error interrupt factor register 1 (SYSC_SYSERRIR1: RUNERRIF0) is set. The contents of the new profile are discarded, and the circuit operates with the contents of the profile currently in use.

See the hardware manual, about the profile error.

Notes of RUN Profile Settings

- Profile Update is prohibited during the profile update. If the profile update again while profile is being updated, a system error interrupt (SYSC_SYSERRIR1: RUNTRGERRIF) will be generated, and the attempted update profile will be disabled.
- Before updating the RUN profile, it is necessary to confirm the flag of the profile status register (SYSC_SYSPROSTSR:RUNPSTS) is already cleared. When it is updated the RUN profile during a profile error status, an NMI interrupt will be generated, and the RUN profile settings will be discarded.
- During RUN profile update, it is prohibited to write access to RUN profile register group. A write access is made to the RUN profile register group during a RUN profile update. The write data will be disabled.

4 Reference

1. 32-BIT MICROCONTROLLER Spansion Traveo Family MB9D560 HARDWARE MANUAL
2. MB9D560 Series Application Notes (Dual CPU Operation)

5 Document History

Document Title: AN204450 - Traveo™ Family, MB9D560 Setting Clock System

Document Number: 002-04450

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	-	KHAS	07/16/2015	Initial Release
*A	5041474	KHAS	12/08/2015	Migrated Spansion Application Note from MB9D560_AN708-00008-1v0-E to Cypress format

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at [Cypress Locations](#).

Products

Automotive	cypress.com/go/automotive
Clocks & Buffers	cypress.com/go/clocks
Interface	cypress.com/go/interface
Lighting & Power Control	cypress.com/go/powerpsoc
Memory	cypress.com/go/memory
PSoC	cypress.com/go/psoc
Touch Sensing	cypress.com/go/touch
USB Controllers	cypress.com/go/usb
Wireless/RF	cypress.com/go/wireless
Spansion Products	spansion.com/products

PSoC® Solutions

psoc.cypress.com/solutions

[PSoC 1](#) | [PSoC 3](#) | [PSoC 4](#) | [PSoC 5LP](#)

Cypress Developer Community

[Community](#) | [Forums](#) | [Blogs](#) | [Video](#) | [Training](#)

Technical Support

cypress.com/go/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.



Cypress Semiconductor Phone : 408-943-2600
198 Champion Court Fax : 408-943-4730
San Jose, CA 95134-1709 Website : www.cypress.com

© Cypress Semiconductor Corporation, 2015. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

This Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges. Use may be limited by and subject to the applicable Cypress software license agreement.