Toshiba Semiconductor and Storage - TMP19A43FDXBG Datasheet

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Not For New Designs
Core Processor	TX19A
Core Size	16/32-Bit
Speed	40MHz
Connectivity	EBI/EMI, I²C, UART/USART
Peripherals	DMA, WDT
Number of I/O	143
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	24K x 8
Voltage - Supply (Vcc/Vdd)	1.35V ~ 3.6V
Data Converters	A/D 16x10b; D/A 2x8b
Oscillator Type	External
Operating Temperature	-20°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	193-FBGA
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/toshiba-semiconductor-and-storage/tmp19a43fdxbg

(6) 32-bit timer
- 32-bit input capture register : 4 channels
- 32-bit compare register : 8 channels
- 32-bit time base timer : 1 channel

(7) Clock timer : 1 channel

(8) General-purpose serial interface : 3 channels
- Selectable between the UART mode and the synchronization mode

(9) High-speed serial interface : 3 channels
- Selectable between the UART mode and the high-speed synchronization mode (maximum speed: 10 Mbps in the high-speed synchronization mode @40MHz)

(10) Serial bus interface : 1 channel
- Selectable between the I^2C bus mode and the clock synchronization mode

(11) 10-bit A/D converter (with S/H) : 16 channels
- Start by an external trigger, and the internal timer activated by a trigger
- Fixed channel/scan mode
- Single/repeat mode
- High-priority conversion mode
- Timer monitor function
- Conversion time 1.15 μsec(@ 40MHz)

(12) 8-bit D/A converter : 2 channels

(13) Watchdog timer : 1 channel

(14) Interrupt function
- CPU: 2 factors .....software interrupt instruction
- Internal: 46 factors.....The order of precedence can be set over 7 levels (except the watchdog timer interrupt).
- External: 48 factors .....The order of precedence can be set over 7 levels. Because 32 factors are associated with KWUP, the number of interrupt factors is one.

(15) Input and output ports .....143 terminals

(16) Standby function
- Three standby modes (IDLE, SLEEP, STOP)

(17) Clock generator
- Built-in PLL (multiplication by 4)
- Clock gear function: The high-speed clock can be divided into 3/4, 1/2, 1/4 or 1/8.
- Sub-clock: SLOW and SLEEP modes (32.768 kHz)

(18) Endian: Bi-endian (big-endian/little-endian)

(19) Maximum operating frequency
- 40 MHz (PLL multiplication)

(20) Operating voltage range
- Core: 1.35 V to 1.65 V
- I/O and ADC: 2.7 V to 3.6 V
- DAC: 2.3 V to 2.7 V

(21) Package

P-FBGA193 (12 mm × 12 mm, 0.65 mm pitch)

JEDEC compliant functions	Modified, added, or deleted functions
• Automatic programming • Automatic chip erase • Automatic block erase • Data polling/toggle bit	<Modified> Block protect (only software protection is supported) <Deleted> Erase resume - suspend function Automatic multiple block erase (supported to the chip level)

3.1.2 Block Diagram of the Flash Memory Section

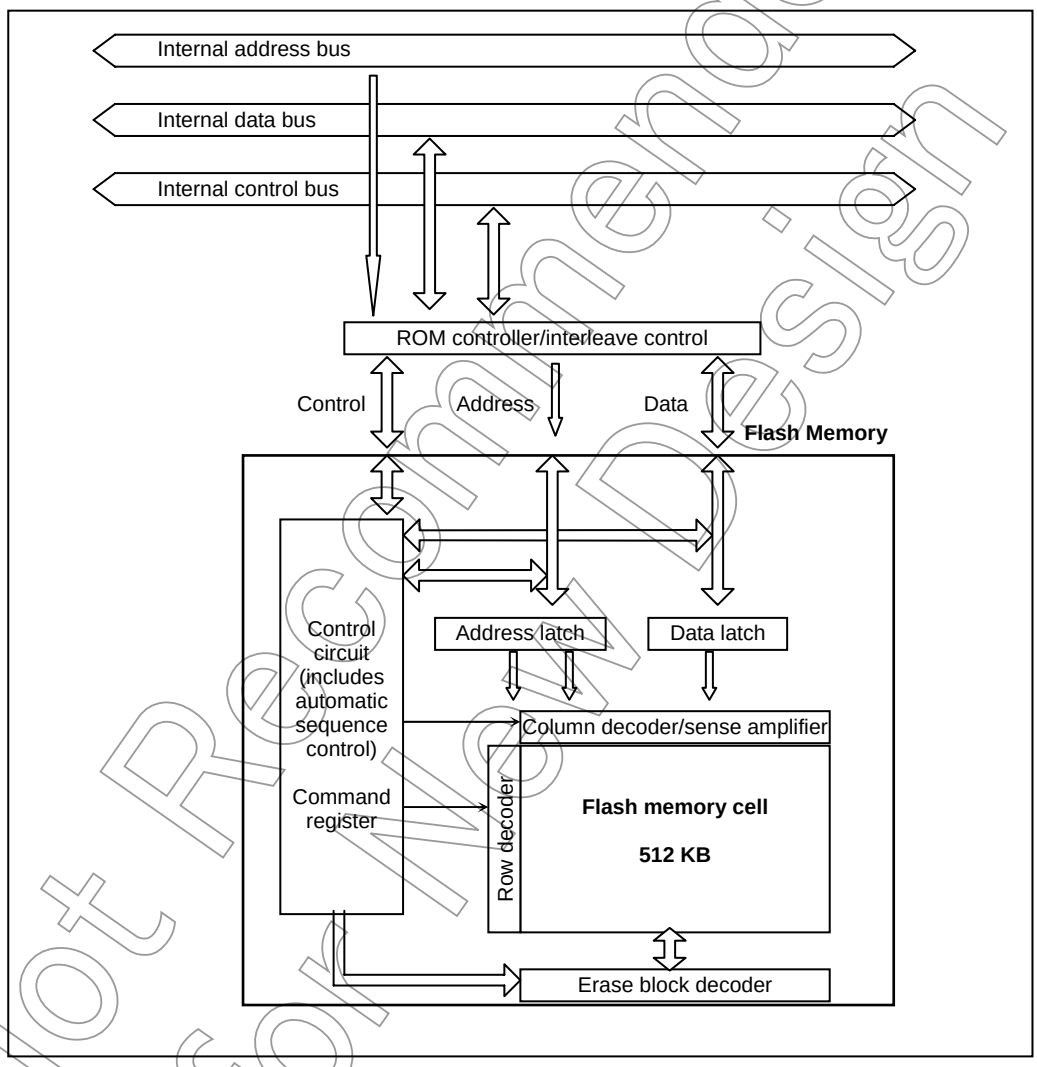


Fig. 3-1 Block Diagram of the Flash Memory Section

3.2 Operation Mode

This device has three operation modes including the mode not to use the internal flash memory.

Table 3-1 Operation Modes

Operation mode		Operation details
Single chip mode		After reset is cleared, it starts up from the internal flash memory.
	Normal mode	In this operation mode, two different modes, i.e., the mode to execute user application programs and the mode to rewrite the flash memory onboard the user's card, are defined. The former is referred to as "normal mode" and the latter "user boot mode."
	User boot mode	The user can uniquely configure the system to switch between these two modes. For example, the user can freely design the system such that the normal mode is selected when the port "00" is set to "1" and the user boot mode is selected when it is set to "0." The user should prepare a routine as part of the application program to make the decision on the selection of the modes.
Single boot mode		After reset is cleared, it starts up from the internal Boot ROM (Mask ROM). In the Boot ROM, an algorithm to enable flash memory rewriting on the user's set through the serial port of this device is programmed. By connecting to an external host computer through the serial port, the internal flash memory can be programmed by transferring data in accordance with predefined protocols.

Among the flash memory operation modes listed in the above table, the User Boot mode and the Single Boot mode are the programmable modes. These two modes, the User Boot mode and the Single Boot mode, are referred to as "Onboard Programming" modes where onboard rewriting of internal flash memory can be made on the user's card.

3.3.1 Configuring for Single Boot Mode

For on-board programming, boot the TMP19A43FDXBG in Single Boot mode, as follows:

$\overline{BOOT}$ = 0

$\overline{RESET}$ = 0 → 1

Set the $\overline{RESET}$ input at logic 0, and the BW0, BW1 and $\overline{BOOT}$ inputs at the logic values shown above, and then release $\overline{RESET}$ (high).

3.3.2 Memory Map

Figure 3.1 shows a comparison of the memory maps in Normal and Single Boot modes. In single Boot mode, the on-chip flash memory is mapped to physical addresses (0x4000_0000 through 0x4007_FFFF), virtual addresses (0x0000_0000 through 0x0007_FFFF), and the on-chip boot ROM is mapped to physical addresses 0x1FC0_0000 through 0x1FC0_1FFF.

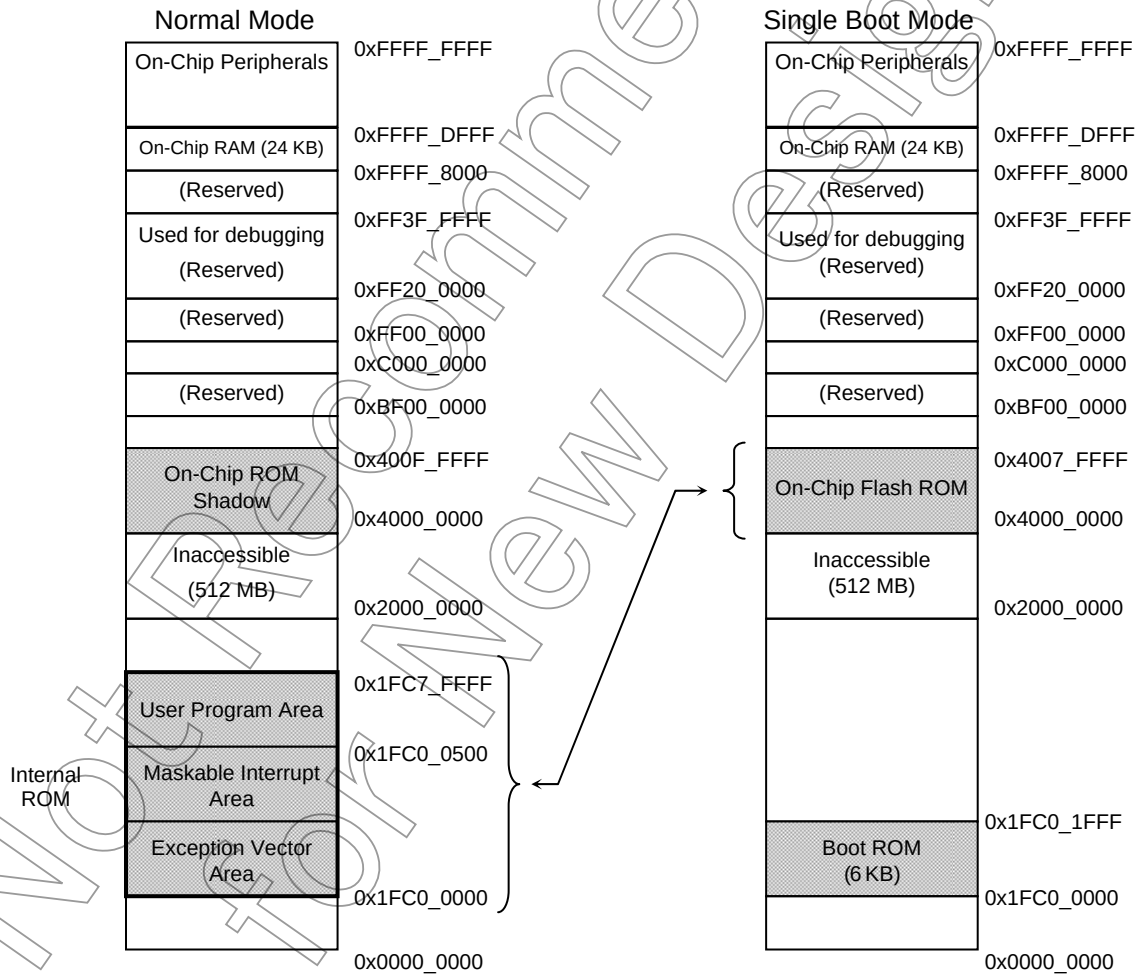


Figure 3.1 Memory Maps for Normal and Single Boot Modes (Physical Addresses)

- I/O Interface mode

  The boot program programs the SC0MOD0 and SC0CR registers to configure the SIO0 in I/O Interface mode (clocked by the rising edge of SCLK0), writes 30H to the SC0BUF. Then, the SIO0 waits for the SCLK0 signal to come from the controller. Following the transmission of the 1st byte, the controller should send the SCLK clock to the target board after a certain idle time (several microseconds). This must be done at 1/16 the desire baud rate. If the 2nd byte, which is from the target board to the controller, is 30H, then the controller should take it as a go-ahead. The controller must then delivers the 3rd byte to the target board at a rate equal to the desired baud rate. The boot program sets the RXE bit in the SC0MOD register to enable reception before loading the SIO transmit buffer with 30H.

(7) The 3rd byte, which the target board receives from the controller, is a command. The code for the RAM Transfer command is 10H.

(8) The 4th byte, transmitted from the target board to the controller, is an acknowledge response to the 3rd byte. Before sending back the acknowledge response, the boot program checks for a receive error. If there was a receive error, the boot program transmits x8H and returns to the state in which it waits for a command again. In this case, the upper four bits of the acknowledge response are undefined — they hold the same values as the upper four bits of the previously issued command. When the SIO0 is configured for I/O Interface mode, the boot program does not check for a receive error.

If the 3rd byte is equal to any of the command codes listed in Table 3.2, the boot program echoes it back to the controller. When the RAM Transfer command was received, the boot program echoes back a value of 10H and then branches to the RAM Transfer routine. Once this branch is taken, a password check is done. Password checking is detailed in Section 3.3.11.

If the 3rd byte is not a valid command, the boot program sends back x1H to the controller and returns to the state in which it waits for a command again. In this case, the upper four bits of the acknowledge response are undefined — they hold the same values as the upper four bits of the previously issued command.

(9) The 5th to 16th bytes, which the target board receives from the controller, are a 12-byte password. The 5th byte is compared to the contents of address 0x0000_03F4 in the flash memory; the 6th byte is compared to the contents of address 0x0000_03F5 in the flash memory; likewise, the 16th byte is compared to the contents of address 0x0000_03FF in the flash memory. If the password checking fails, the RAM Transfer routine sets the password error flag.

(10) The 17th byte is a checksum value for the password sequence (5th to 16th bytes). To calculate the checksum value for the 12-byte password, add the 12 bytes together, drop the carries and take the two's complement of the total sum. Transmit this checksum value from the controller to the target board. The checksum calculation is described in details in Section 3.3.13.

- The RAM storage start address must be within the range 0xFFFD_6000–0xFFFD_EFFF.

When the above checks have been successful, the RAM Transfer routine returns a normal acknowledge response (10H) to the controller.

(16) The 27th to mth bytes from the controller are stored in the on-chip RAM of the TMP19A43FDXBG. Storage begins at the address specified by the 19th–22nd bytes and continues for the number of bytes specified by the 23rd–24th bytes.

(17) The (m+1)th byte is a checksum value. To calculate the checksum value, add the 27th to mth bytes together, drop the carries and take the two's complement of the total sum. Transmit this checksum value from the controller to the target board. The checksum calculation is described in details in Section 3.3.13.

(18) The (m+2)th byte is a acknowledge response to the 27th to (m+1)th bytes.

First, the RAM Transfer routine checks for a receive error in the 27th to (m+1)th bytes. If there was a receive error, the RAM Transfer routine sends back 18H and returns to the state in which it waits for a command (i.e., the 3rd byte) again. In this case, the upper four bits of the acknowledge response are the same as those of the previously issued command (i.e., all 1s). When the SIO0 is configured for I/O Interface mode, the RAM Transfer routine does not check for a receive error.

(19) Next, the RAM Transfer routine performs the checksum operation to ensure data integrity. Adding the series of the 27th to (m+1)th bytes must result in zero (with the carry dropped). If it is not zero, one or more bytes of data has been corrupted. In case of a checksum error, the RAM Transfer routine sends back 11H to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again. When the above checks have been successful, the RAM Transfer routine returns a normal acknowledge response (10H) to the controller. If the (m+2)th byte was a normal acknowledge response, a branch is made to the address specified by the 19th to 22nd bytes in 32-bit ISA mode.

3.3.7 Show Flash Memory Sum Command

See Table 3.4.

(20) The processing of the 1st and 2nd bytes are the same as for the RAM Transfer command.

(21) The 3rd byte, which the target board receives from the controller, is a command. The code for the Show Flash Memory Sum command is 20H.

(22) The 4th byte, transmitted from the target board to the controller, is an acknowledge response to the 3rd byte. Before sending back the acknowledge response, the boot program checks for a receive error. If there was a receive error, the boot program transmits x8H and returns to the state in which it waits for a command again. In this case, the upper four bits of the acknowledge response are undefined — they hold the same values as the upper four bits of the previously issued command. When the SIO0 is configured for I/O Interface mode, the boot program does not check for a receive error.

If the 3rd byte is equal to any of the command codes listed in Table 3.2 on page 3-18, the boot program echoes it back to the controller. When the Show Flash Memory Sum command was received, the boot program echoes back a value of 20H and then branches to the Show Flash Memory Sum routine.

3.3.9 Acknowledge Responses

The boot program represents processing states with specific codes. Table 3.6 to Table 3.8 show the values of possible acknowledge responses to the received data. The upper four bits of the acknowledge response are equal to those of the command being executed. Bit 3 of the code indicates a receive error. Bit 0 indicates an invalid command error, a checksum error or a password error. Bit 1 and bit 2 are always 0. Receive error checking is not done in I/O Interface mode.

Table 3.6 ACK Response to the Serial Operation Mode Byte

Return Value	Meaning
86H	The SIO can be configured to operate in UART mode. (See Note)
30H	The SIO can be configured to operate in I/O Interface mode.

Note: If the serial operation mode is determined as UART, the boot program checks if the SIO can be programmed to the baud rate at which the operation mode byte was transferred. If that baud rate is not possible, the boot program aborts, without sending back any response.

Table 3.7 ACK Response to the Command Byte

Return Value	Meaning
x8H (See Note)	A receive error occurred while getting a command code.
x1H (See Note)	An undefined command code was received. (Reception was completed normally.)
10H	The RAM Transfer command was received.
20H	The Show Flash Memory Sum command was received.
30H	The Show Product Information command was received.

Note: The upper four bits of the ACK response are the same as those of the previous command code.

Table 3.8 ACK Response to the Checksum Byte

Return Value	Meaning
18H	A receive error occurred.
11H	A checksum or password error occurred.
10H	The checksum was correct.

3.3.14 General Boot Program Flowchart

Figure 3.6 shows an overall flowchart of the boot program.

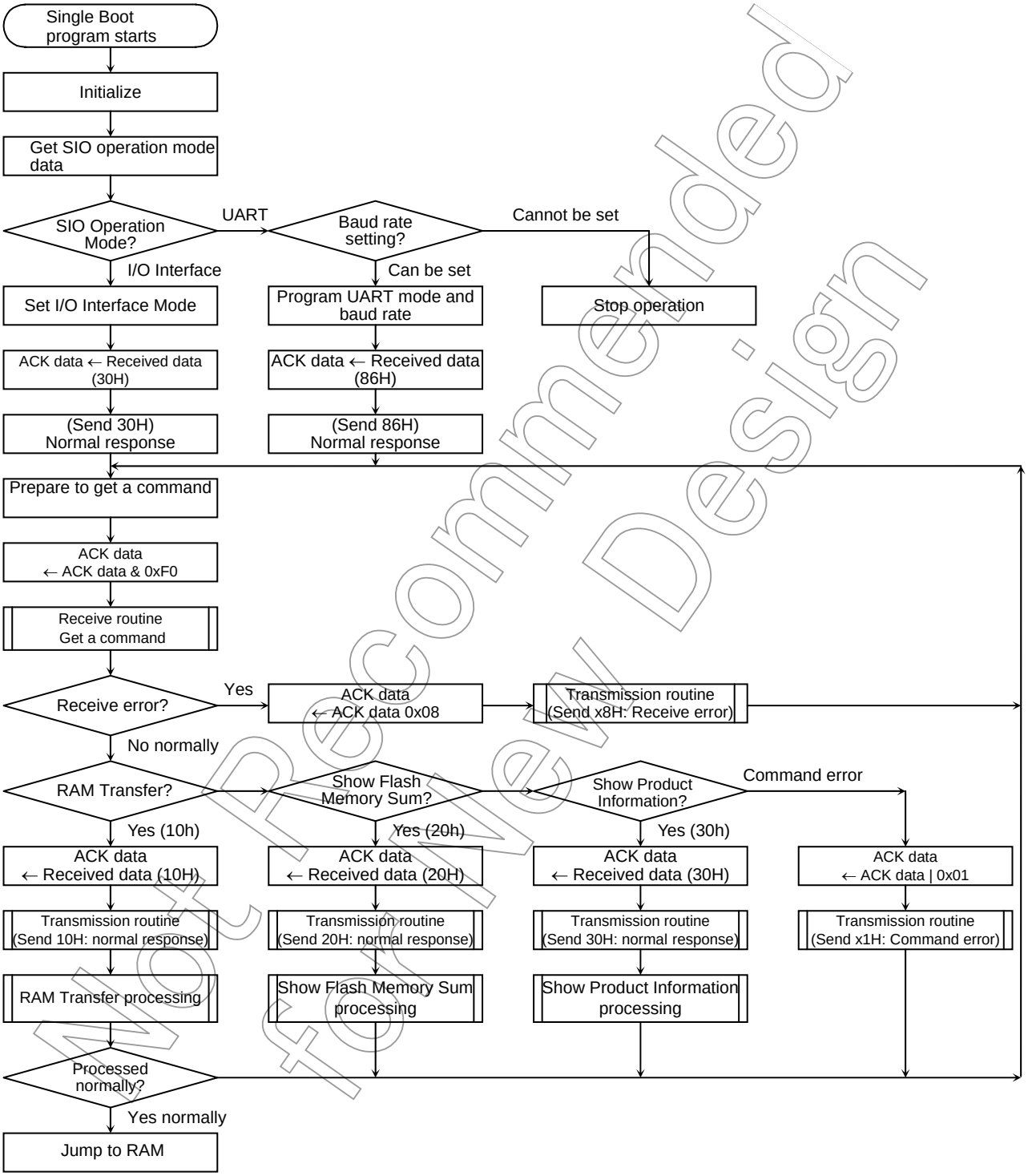


Figure 3.6 Overall Boot Program Flow

9) Flash control/ status register

This resister is used to monitor the status of the flash memory and to indicate the block protection status.

Table 3-6 Flash Control Register

FLCS (0xFFFF_E520)

	7	6	5	4	3	2	1	0
Bit Symbol	BLPRO3	BLPRO2	BLPRO1	BLPRO0		ROMTYPE		RDY/BSY
Read/Write	R				R	R	R	R
After reset	0	0	0	0	0	0	0	1
Function	Protection area setting (for each 128 kB) 0000: No blocks are protected xxx1: Block 0 is protected xx1x: Block 1 is protected x1xx: Block 2 is protected 1xxx: Block 3 is protected				Always reads "0."	ROM ID bit 0: Flash 1: MROM	Always reads "0."	Ready/Busy 0: In operation 1: Operation terminated

	15	14	13	12	11	10	9	8
Bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function								

	23	22	21	20	19	18	17	16
Bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function								

	31	30	29	28	27	26	25	24
Bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function								

Fig. 3-5

Bit 0: Ready/Busy flag bit
The RDY/BSY output is provided as a means to monitor the status of automatic operation. This bit is a function bit for the CPU to monitor the function. When the flash memory is in automatic operation, it outputs "0" to indicate that it is busy. When the automatic operation is terminated, it returns to the ready state and outputs "1" to accept the next command. If the automatic operation has failed, this bit maintains the "0" output. By applying a hardware reset, it returns to "1."

(note)Please issue it after confirming the command issue is always a ready state.
A normal command not only is sent when the command is issued to a busy inside but also there is a possibility that the command after that cannot be input. In that case, please return by system reset or the reset command.

Bit 2: ROM type identification bit
This bit is read after reset to identify whether the ROM is a flash ROM or a mask ROM.
Flash ROM: "0"
Mask ROM: "1"

Bits [7:4]: Protection status bits (can be set to any combination of blocks)
Each of the protection bits (4 bits) represents the protection status of the corresponding block. When a bit is set to "1," it indicates that the block corresponding to the bit is protected. When the block is protected, data cannot be written to it.

4.3 Protect Configuration and Protect Statuses

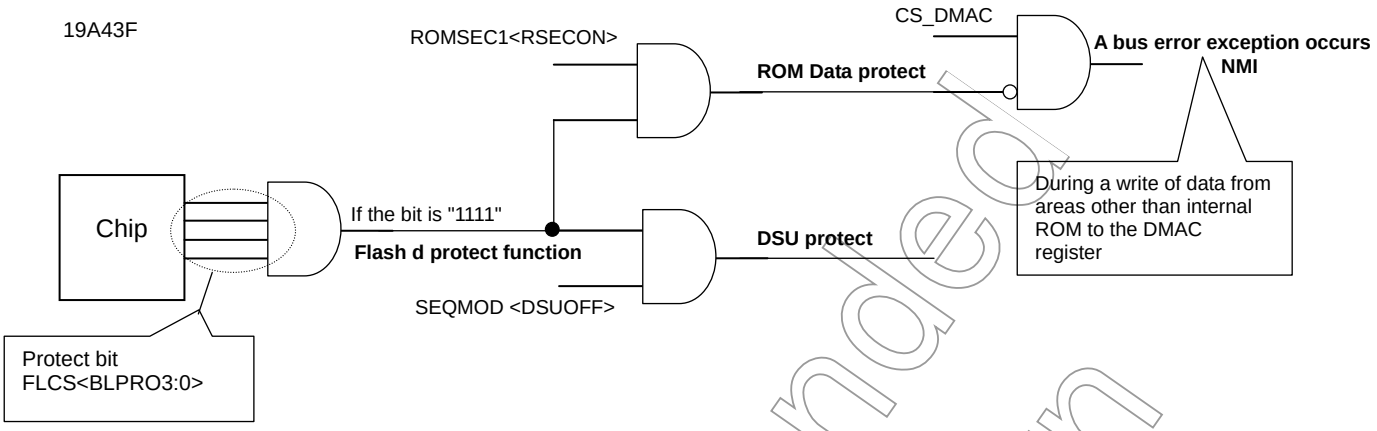


Fig. 4-1 Various Protect Statuses

Table 4-1 Protect Statuses in Each Mode

	Protect bit setting FLCS<BLPRO 3:0>	1111				≠ 1111
	ROM protect enable bit ROMSEC1<RSECON>	1		0		Don't Care
	DSU protect enable bit SEQMOD <DSUOFF>	1	0	1	0	Don't Care
	Flash read protect status	ON				OFF
	ROM protect status	ON		OFF		OFF
	DSU protect status	ON	OFF	ON	OFF	OFF
Single /single boot mode	Read of flash from internal ROM	○	○	○	○	○
	Read of flash from areas other than internal ROM	× *1	× *1	○	○	○
	Clearing of ROM protect enable status (from ROM)	○	○			○
	Clearing of ROM protect enable status (from areas other than ROM)	× *2	× *2			○
	Clearing of DSU protect enable status (from ROM)	○		○		○
	Clearing of DSU protect enable status (from areas other than ROM)	× *3		○		○
	Issuing of the command to erase protect bits	× *4	× *4	○ *8	○ *8	○
	Issuing of commands other than the command to erase protect bits	× *5	× *5	× *7	× *7	Δ *9
	Writing of data to the DMAC setting register (from ROM)	○	○	○	○	○
	Writing of data to the DMAC setting register (from areas other than ROM)	× *6	× *6	○	○	○

*1 : The data of address "0xBFC0_0000" or "0xBFC0_0002" can be read.
*2 : Stored data is masked. A write to registers cannot be executed (data in registers cannot be cleared).
*3 : Stored data is masked. A write to registers cannot be executed (data in registers cannot be cleared).
*4 : A command address is masked, and flash memory does not recognize commands.
*5 : A command address is masked, and flash memory does not recognize commands.
*6 : A bus error exception occurs (when making the DMAC register setting).
*7 : Because a read of flash memory is prohibited, commands are not recognized.
*8 : Because a read of flash memory is prohibited, issued commands are converted to the command for erasing the whole flash memory area and the command for erasing all protect bits.

4.5 Proted-related / Release Settings

If it is necessary to overwrite flash memory or protect bits in a protected state, "automatic protect bit deletion" must be executed or the ROM protect function must be disabled. DSU cannot be used if it is in a protected state.

Flash memory may go into a read-protected state after the automatic protect bit program is executed. In this case, it is necessary to set DSU-PROBE to "enable" before the automatic protect bit program is executed.

（The mask version is possible only the release of ROM security, and the protecting bit cannot be rewritten.）

If "automatic protect bit deletion" is executed when flash memory is in a read-protected state, flash memory is automatically initialized inside this device. Therefore, extra caution must be used when switching from one state to a read-protected state. (FLASH only)

4.5.1 Flash Protect Function

The flash protecting function cannot be released always effectively in the mask version.

It becomes effective by putting the block protecting on all of the four blocks in the flash version.

The flash memory command sequence and protect bit program commands are used to enable or disable the flash read protect function. For further information, refer to the command sequence explained in the chapter describing the operations of flash memory.

(notes concerning FLASH version)

The protecting bit is cleared after all the data of the flash is deleted when the protecting bit release command is executed with the flash protected, and the flash protecting is released.

In the state of ROM data protecting, explains as follows, the command execution to the flash is disregarded. It is necessary to release ROM data protecting first clearing the RSECON bit of ROM protecting register when the flash protecting is released with ROM protected.

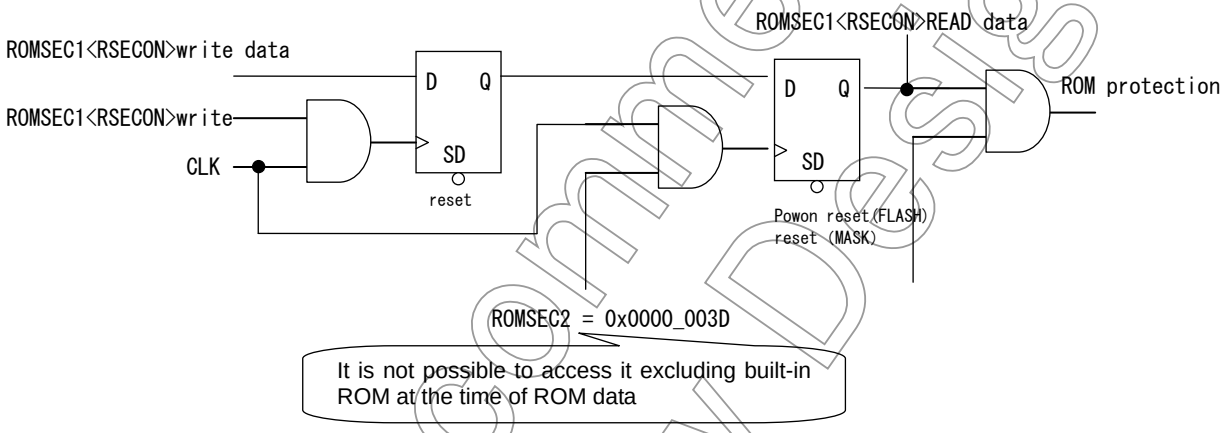
4.5.2 ROM data Protect

ROM data protecting is effective the flash protecting and becomes effective at ROM protecting register ROMSEC1<RSECON>=“1".

After releasing reset, the RSECON bit is initialized by "1". The flash protecting is sure to enter the state of ROM data protecting in the mask version after releasing reset because it is always effective.

It decides whether to enter the state of ROM data protecting by the state of the flash protecting in the flash version.

When ROM protecting register is rewritten with ROM data protected, rewriting can be executed only from the program put on built-in ROM. Therefore, it is necessary to prepare the release program of ROM data protecting on built-in ROM.



ROM data protecting is released by setting ROM protecting register ROMSEC1<RSECON>“0" when protecting is released, and writing protecting code “0x0000_003D" in ROM protecting lock register ROMSEC2. Moreover, ROM data protecting function can be set again by similarly setting ROM protecting register ROMSEC1<RSECON>“1" when ROM protecting is set, and writing protecting code “0x0000_003D" in ROM protecting lock register ROMSEC2.

It is necessary to note the ROMSEC2 register because the reading data is different from original write data because of the register only for writing.

The initialization of ROM protecting register is different in the flash version and the mask version.

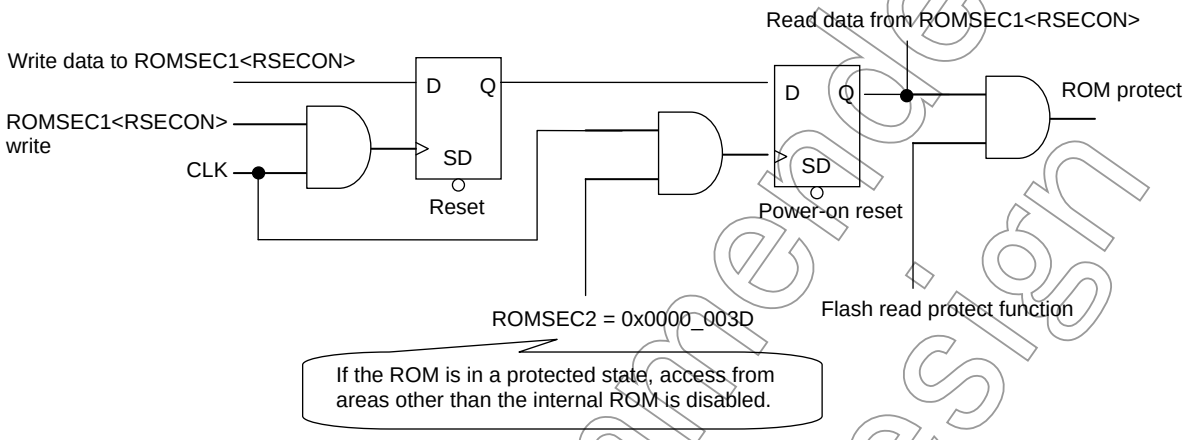
It provides with the power-on reset circuit in the flash version, ROM protecting register is initialized by power-on reset, and the value doesn't usually change in reset.

It is usually initialized by reset in the mask version because power-on reset is not provided.

It is necessary to note it in the mask version because it is usually initialized at each reset.

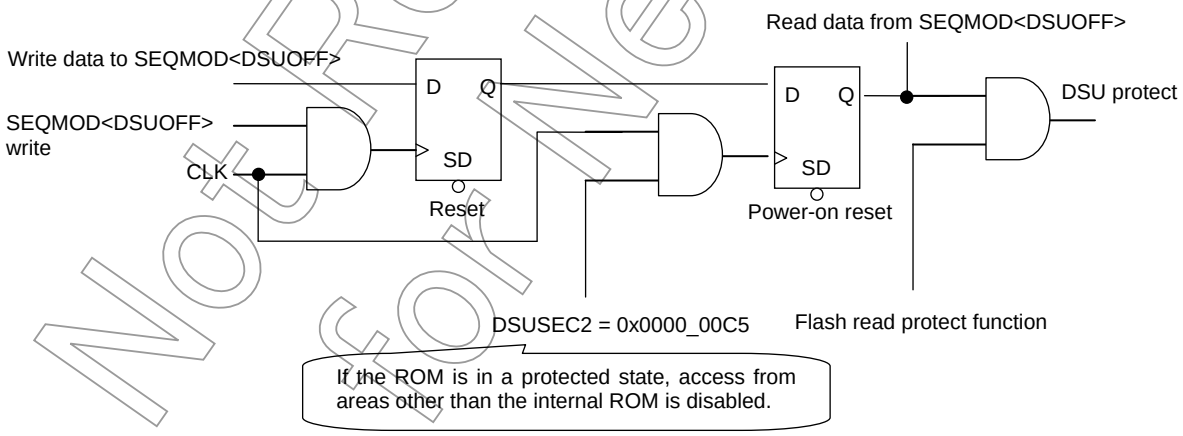
4.5.4 ROM Protect Register: ROMSEC1<RSECON>

The ROM protect register is equipped with a power-on reset circuit. Caution must be exercised as data read from the ROMSEC1<RSECON> bit is different from the actually written data. How data is processed is shown below. The mask version is usually initialized by reset though FLASH goods are initialized by power-on reset.



4.5.5 DSU Protect Mode Register: SEQMOD <DSUOFF>

The DSU protect mode register is equipped with a power-on reset circuit. Caution must be exercised as data read from the SEQMOD <DSUOFF> bit is different from the actually written data. How data is processed is shown below. The mask version is usually initialized by reset though FLASH goods are initialized by power-on reset.

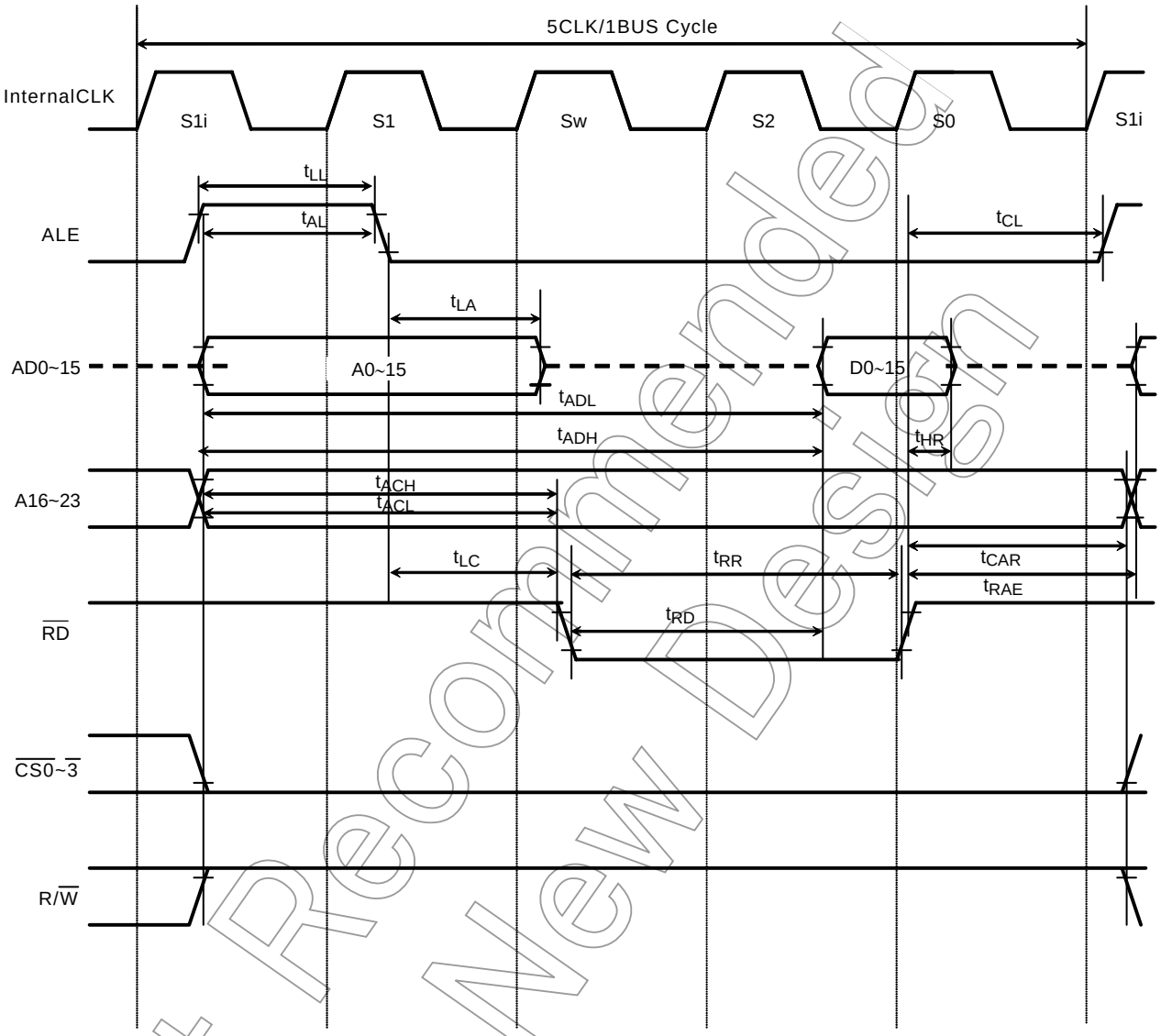


Ta＝－20 to 85℃

Parameter		Symbol	Rating		Min.	Typ. (Note 1)	Max.	Unit
High-level input voltage	P7 to P8 (Used as a port) Normal port	V_{IH1}	2.7V≦AVCC3≦3.6V		0.7 AVCC3		DVCC3 + 0.3 DVCC15+ 0.2 CVCCH+ 0.2 CVCCL+0.3	V
	Normal port	V_{IH2}	2.7V≦DVCC3≦3.6V		0.7 DVCC3			
	Schmitt-Triggered port	V_{IH3}	2.7V≦DVCC3≦3.6V		0.8 DVCC3			
	X1	V_{IH4}	1.35V≦CVCCH≦1.65V		0.9 CVCCH			
	XT2	V_{IH4}	2.7V≦CVCCL≦3.6V		0.9 CVCCL			
Low-level output voltage		V_{OL}	I_{OL} = 2mA	DVCC3≧2.7V			0.4	V
High-level output voltage		V_{OH}	I_{OH} = – 2mA	DVCC3≧2.7V	2.4			

Note 1: Ta = 25°C, DVCC15=1.5V,DVCC3= AVCC3=3.3V,DVCC=2.5V, unless otherwise noted

(1) Read cycle timing, ALE width = 1 clock cycle, 1 programmed wait state



5.7.2 Separate Bus mode

(1) DVCC15＝CVCCH＝1.35Vto1.65V, DVCC3＝AVCC3=2.7Vto3.6V,
DAVCC =2.3 Vto2.7V, Ta = －20 to 85°C

① SYSCR3<ALESEL> = “0”, 2 programmed wait state

No.	Parameter	Symbol	Equation		40 MHz (fsys)(Note)		Unit
			Min	Max	Min	Max	
1	System clock period (x)	t_{SYS}	25				ns
2	A0-A23 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{AC}	X(1＋ALE) –11		39.0		ns
3	A0-A23 hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated	t_{CAR}	x – 11		14.0		ns
4	A0-A23 valid to D0-D15 Data in	t_{AD}		x (2 + TW+ALE) – 43		82.0	ns
5	$\overline{RD}$ asserted to D0-D15 data in	t_{RD}		x (1 + TW) – 40		35.0	ns
6	$\overline{RD}$ width low	t_{RR}	x (1 +TW) –6		69.0		ns
7	D0-D15 hold after $\overline{RD}$ negated	t_{HR}	0		0		ns
8	$\overline{RD}$ negated to next A0-A23 output	t_{RAE}	x – 6		19.0		ns
9	$\overline{WR}$ /$\overline{HWR}$ width low	t_{WW}	x (1 + TW) –6		69.0		ns
10	$\overline{WR}$ or $\overline{HWR}$ asserted to D0-D15 valid	t_{DO}		9.7		9.7	ns
11	D0-D15 hold after $\overline{WR}$ or $\overline{HWR}$ negated	t_{DW}	x (1 + TW) – 11		64.0		ns
12	D0-D15 hold after $\overline{WR}$ or $\overline{HWR}$ negated	t_{WD}	x – 11		14.0		ns
13	A0-A23 valid to $\overline{WAIT}$ input	t_{AW}		x+ x (ALE)＋ x (TW-1)－32		43.0	ns
14	$\overline{WAIT}$ hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{CW}	x (TW - 3)-16	x (TW – 1) – 29	9.0	46.0	ns

Note 1: No. 1 to 14:

Internal 2 wait insertion，ALE “1” Clock, @40MHz

TW = W + 2N ,

W : Number of Auto wait insertion , 2N : Number of external wait insertion

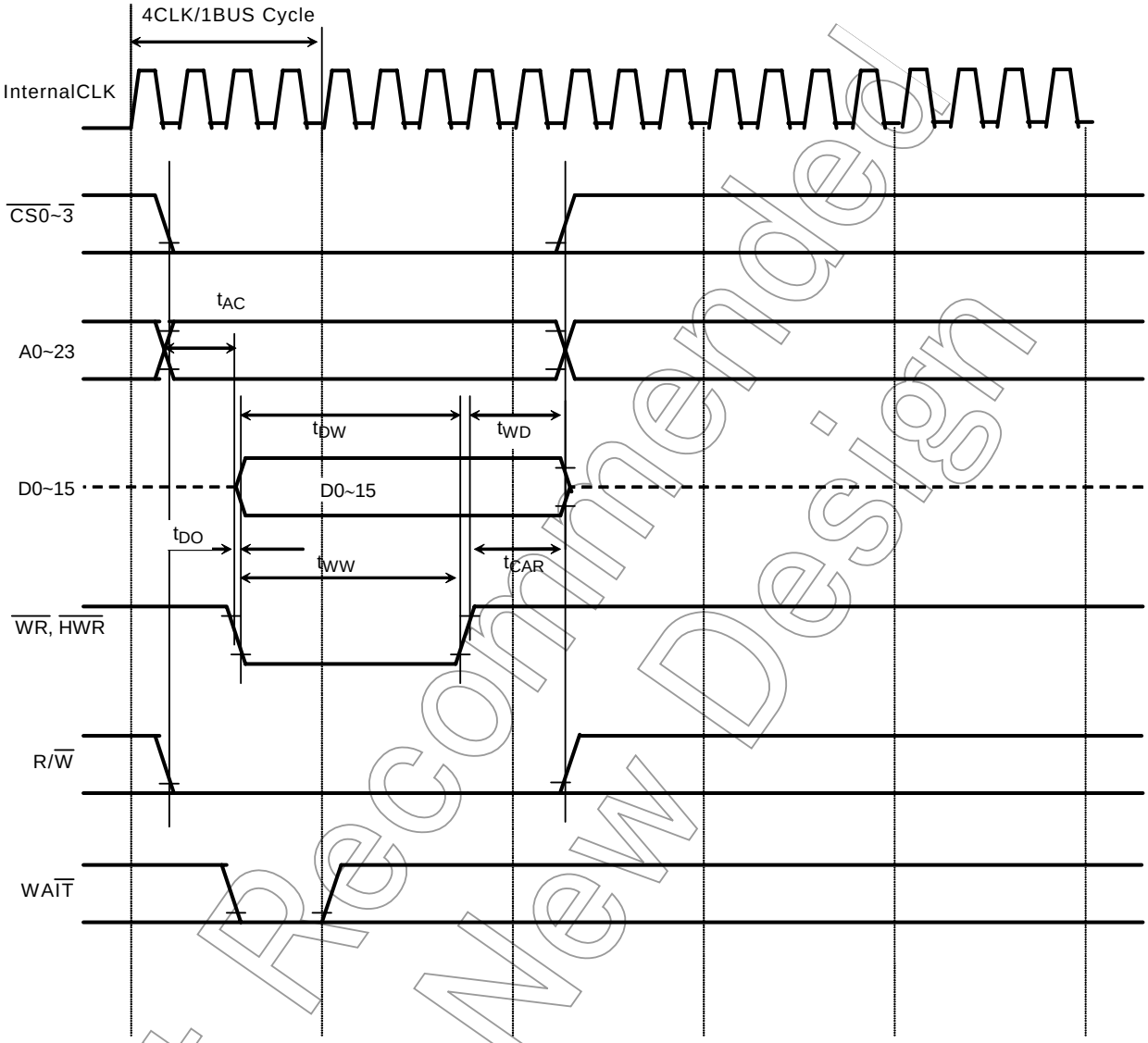
TW = 2 + 2*1 = 4

AC measurement conditions:

Output levels: High = 0.8DVCC3 V/Low 0.2DVCC3 V, CL = 30 pF

Input levels: High = 0.7DVCC3 V/Low 0.3DVCC3 V

(5) Write cycle timing (SYSCR3<ALESEL> =1, 4 wait state)



5.11 SBI Timing

(1) I2C mode

In the table below, the letters x represent the fsys periods, respectively.

n denotes the value of n programmed into the SCK (SCL output frequency select) field in the SBI0CR.

Parameter	Symbol	Equation		Standard mode		Fast mode		Unit
		Min	Max	Min	Max	Min	Max	
SCL clock frequency	tSCL	0		0	100	0	400	kHz
Hold time for START condition	tHD:STA			4.0		0.6		μs
SCL clock low width (Input) (Note 1)	tLOW			4.7		1.3		μs
SCL clock high width (Output) (Note 2)	tHIGH			4.0		0.6		μs
Setup time for a repeated START condition	tSU;STA	(Note 5)		4.7		0.6		μs
Data hold time (Input) (Note 3, 4)	tHD;DAT			0.0		0.0		μs
Data setup time	tSU;DAT			250		100		ns
Setup time for STOP condition	tSU;STO			4.0		0.6		μs
Bus free time between STOP and START conditions	tBUF	(Note 5)		4.7		1.3		μs

Note 1: SCL clock low width (output) is calculated with: $(2^{n-1}+58)/(fsys/2)$

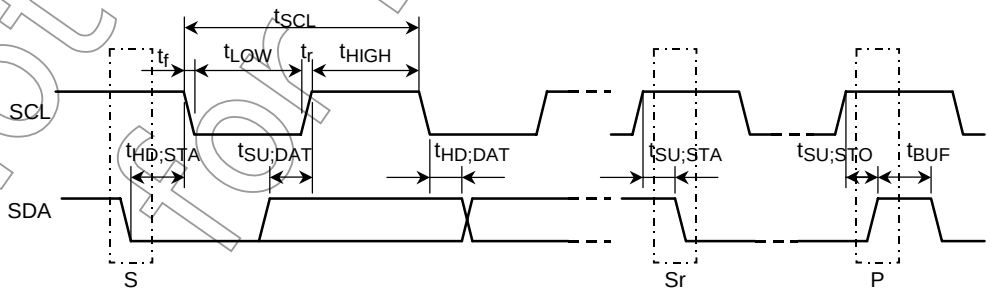
Note 2: SCL clock high width (output) is calculated with $(2^{n-1}+12)/(fsys/2)$

Notice: On I^2C-bus specification, Maximum Speed of Standard Mode is 100KHz ,Fast mode is 400Khz. Internal SCL clock Frequency setting should be shown above Note1 & Note2.

Note 3: The output data hold time is equal to 12x

Note 4: The Philips I^2C-bus specification states that a device must internally provide a hold time of at least 300 ns for the SDA signal to bridge the undefined region of the fall edge of SCL. However, this SBI does not satisfy this requirement. Also, the output buffer for SCL does not incorporate slope control of the falling edges; therefore, the equipment manufacturer should design so that the input data hold time shown in the table is satisfied, including tr/tf of the SCL and SDA lines.

Note 5: Software-dependent



S: START condition
Sr: Repeated START condition
P: STOP condition

5.12 Event Counter

In the table below, the letter x represents the fsys cycle period.

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
Clock low pulse width	t_{VCKL}	2X + 100		150		ns
Clock high pulse width	t_{VCKH}	2X + 100		150		ns

5.13 Timer Capture

In the table below, the letter x represents the fsys cycle period.

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
Low pulse width	t_{CPL}	2X + 100		150		ns
High pulse width	t_{CPH}	2X + 100		150		ns

5.14 General Interrupts

In the table below, the letter x represents the fsys cycle period.

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
Low pulse width for INT0-INTA	t_{INTAL}	X + 100		125		ns
High pulse width for INT0-INTA	t_{INTAH}	X + 100		125		ns

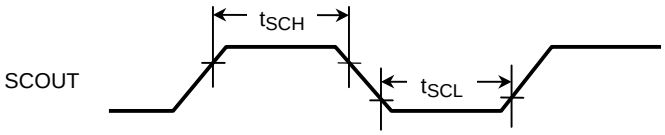
5.15 STOP /SLEEP/SLOW Wake-up Interrupts

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
Low pulse width for INT0-INTB	t_{INTBL}	100		100		ns
High pulse width for INT0-INTB	t_{INTBH}	100		100		ns

5.16 SCOUT Pin

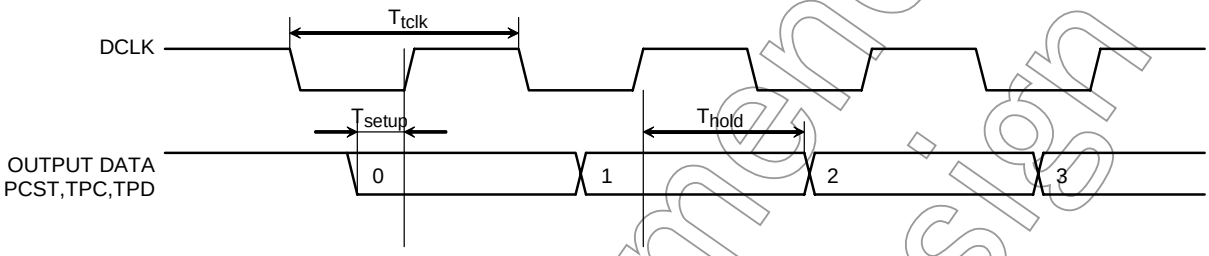
Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
Clock high pulse width	t_{SCH}	0.5T – 5		7.5		ns
Clock low pulse width	t_{SCL}	0.5T – 5		7.5		ns

Note: In the above table, the letter T represents the cycle period of the SCOUT output clock.



5.21 DSU

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
PCST valid to DCLK negated	Tsetup	11		11		ns
PCST hold after DCLK negated	Thold	0.5		0.5		ns
TPC valid to DCLK negated	Tsetup	11		11		ns
TPC hold after DCLK negated	Thold	0.5		0.5		ns
TPD valid to DCLK negated	Tsetup	11		11		ns
TPD hold after DCLK negated	Thold	0.5		0.5		ns



5.22 EJTAG

Parameter	Symbol	Equation		10 MHz(※)		Unit
		Min	Max	Min	Max	
TCK valid to TMS/TDI Data in	Ttsetup	40		40		ns
TMS/TDI hold after TCK negated	Tthold	50		50		ns
TDO hold after TCK asserted	Tt_{obt}		10		10	ns

※ Operating Frequency of TCK is 10MHz Only