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	Obsolete
Core Processor	H8/300H
Core Size	16-Bit
Speed	13MHz
Connectivity	SCI, SmartCard
Peripherals	DMA, PWM, WDT
Number of I/O	35
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x10b; D/A 2x8b
Oscillator Type	Internal
Operating Temperature	-20°C ~ 75°C (TA)
Mounting Type	Surface Mount
Package / Case	100-BFQFP
Supplier Device Package	100-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/d13007vfi13v

Preface

The H8/3006 and H8/3007 are a series of high-performance microcontrollers that integrate system supporting functions together with an H8/300H CPU core.

The H8/300H CPU has a 32-bit internal architecture with sixteen 16-bit general registers, and a concise, optimized instruction set designed for speed. It can address a 16-Mbyte linear address space.

The on-chip supporting functions include RAM, 16-bit timers, 8-bit timers, a programmable timing pattern controller (TPC), a watchdog timer (WDT), a serial communication interface (SCI), an A/D converter, a D/A converter, I/O ports, and a DMA controller (DMAC).

The address space is divided into eight areas. The data bus width and access cycle length can be selected independently in each area, simplifying the connection of different types of memory. Four MCU operating modes (modes 1 to 4) are provided, offering a choice of data bus width initial value and address space.

With these features, the H8/3006 and H8/3007 offer easy implementation of compact, high-performance systems.

This manual describes the H8/3006 and H8/3007 Group hardware. For details of the instruction set, refer to the H8/300H Series Software Manual.

Main Revisions for This Edition

Item	Page	Revision (See Manual for Details)
All	—	Company name and brand names amended (Before) Hitachi, Ltd. → (After) Renesas Technology Corp.
5.4.2 Interrupt Sequence Figure 5.7 Interrupt Sequence	102	Figure amended D_{15} to D_0
8.7.2 Register Configuration Table 8.14 Port A Pin Functions (Modes 1 to 4) $PA_3/TP_3/TIOCB_0/TCLKD$	274	Description and note amended Bit PWM0 in TMDR, bits IOB2 to IOB0 in TIOR0, bits TPSC2 to TPSC0 in 16TCR2 to 16TCR0 of the 16-bit timer, bits CKS2 to CKS0 in 8TCR2 of the 8-bit timer, bit NDER3 in NDERA, and bit PA_3DDR select the pin function as follows. Notes: 2. TCLKD input when TPSC2 = TPSC1 = TPSC0 = 1 in any of 16TCR2 to 16TCR0, or bits CKS2 to CKS0 in 8TCR2 are as shown in (3) in the table below.
$PA_2/TP_2/TIOCA_0/TCLKC$	275	Table amended

Pin **Pin Functions and Selection Method**

$PA_2/TP_2/TIOCA_0/TCLKC$ Bit PWM0 in TMDR, bits IOA2 to IOA0 in TIOR0, bits TPSC2 to TPSC0 in 16TCR2 to 16TCR0 of the 16-bit timer, bits CKS2 to CKS0 in 8TCR0 of the 8-bit timer, bit NDER2 in NDERA, and bit PA_2DDR select the pin function as follows.

16-bit timer channel 0 settings	(1) in table below	(2) in table below		
PA_2DDR	—	0	1	1
NDER2	—	—	0	1
Pin function	$TIOCA_0$ output	PA_2 input	PA_2 output	TP_2 output
		$TIOCA_0$ input*1		
	TCLKC input*2			

Notes: 1. TIOCA0 input when IOA2 = 1.

2. TCLKC input when TPSC2 = TPSC1 = 1 and TPSC0 = 0 in any of 16TCR2 to 16TCR0, or bits CKS2 to CKS0 in 8TCR0 are as shown in (3) in the table below.

16-bit timer channel 0 settings	(2)	(1)		(2)	(1)
PWM0	0				1
IOA2	0			1	—
IOA1	0	0	1	—	—
IOA0	0	1	—	—	—

8-bit timer channel 0 settings	(4)		(3)	
CKS2	0	1		
CKS1	—	0		1
CKS0	—	0	1	—

2.6 Instruction Set

2.6.1 Instruction Set Overview

The H8/300H CPU has 64 types of instructions, which are classified in table 2.1.

Table 2.1 Instruction Classification

Function	Instruction	Types
Data transfer	MOV, PUSH*[1], POP*[1], MOVTPE*[2], MOVFPE*[2]	5
Arithmetic operations	ADD, SUB, ADDX, SUBX, INC, DEC, ADDS, SUBS, DAA, DAS, MULXU, MULXS, DIVXU, DIVXS, CMP, NEG, EXTS, EXTU	18
Logic operations	AND, OR, XOR, NOT	4
Shift operations	SHAL, SHAR, SHLL, SHLR, ROTL, ROTR, ROTXL, ROTXR	8
Bit manipulation	BSET, BCLR, BNOT, BTST, BAND, BIAND, BOR, BIOR, BXOR, BIXOR, BLD, BILD, BST, BIST	14
Branch	Bcc*[3], JMP, BSR, JSR, RTS	5
System control	TRAPA, RTE, SLEEP, LDC, STC, ANDC, ORC, XORC, NOP	9
Block data transfer	EEPMOV	1
		Total 64 types

Notes: 1. POP.W Rn is identical to MOV.W @SP+, Rn.
PUSH.W Rn is identical to MOV.W Rn, @–SP.
POP.L ERn is identical to MOV.L @SP+, Rn.
PUSH.L ERn is identical to MOV.L Rn, @–SP.
2. Not available in the H8/3006 and H8/3007.
3. Bcc is a generic branching instruction.

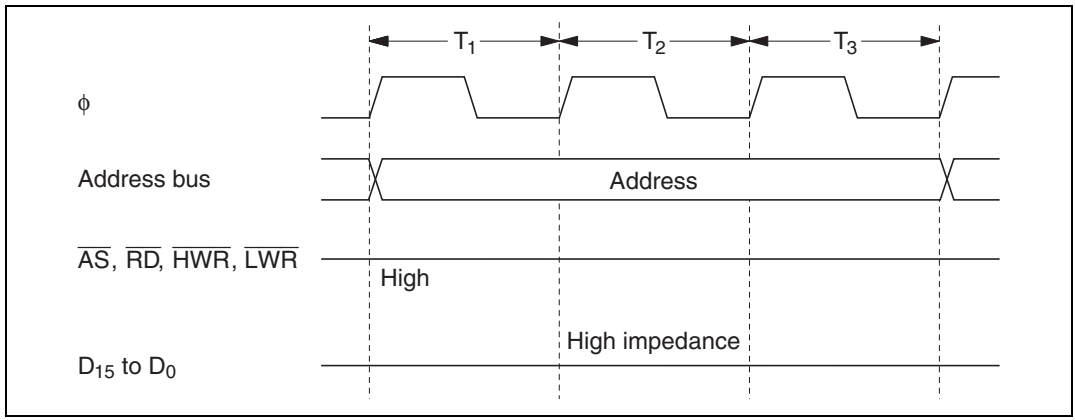


Figure 2.18 Pin States during Access to On-Chip Supporting Modules

2.9.4 Access to External Address Space

The external address space is divided into eight areas (areas 0 to 7). Bus-controller settings determine whether each area is accessed via an 8-bit or 16-bit bus, and whether it is accessed in two or three states. For details see section 6, Bus Controller.

- Burst ROM interface
  - Burst ROM interface can be set for area 0
  - Selection of two- or three-state burst access
- Idle cycle insertion
  - An idle cycle can be inserted in case of an external read cycle between different areas
  - An idle cycle can be inserted when an external read cycle is immediately followed by an external write cycle
- Bus arbitration function
  - A built-in bus arbiter grants the bus right to the CPU, DMAC, DRAM interface, or an external bus master
- Other features
  - The refresh counter (refresh timer) can be used as an interval timer

6.1.2 Block Diagram

Figure 6.1 shows a block diagram of the bus controller.

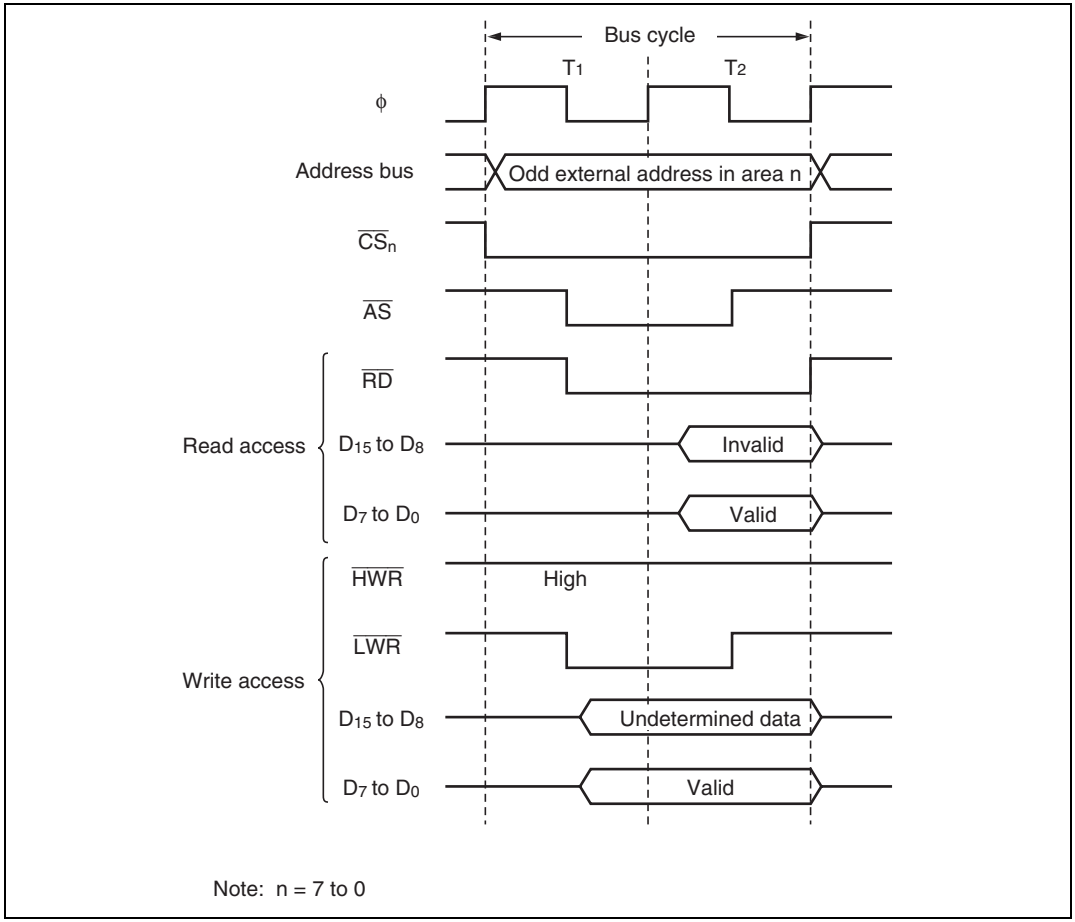


Figure 6.13 Bus Control Signal Timing for 16-Bit, Two-State-Access Area (2) (Byte Access to Odd Address)

Connection Examples

- Figure 6.29 shows typical interconnections when using two 2-CAS type 16-Mbit DRAMs using a × 16-bit organization, and the corresponding address map. The DRAMs used in this example are of the 10-bit row address × 10-bit column address type. Up to four DRAMs can be connected by designating areas 2 to 5 as DRAM space.

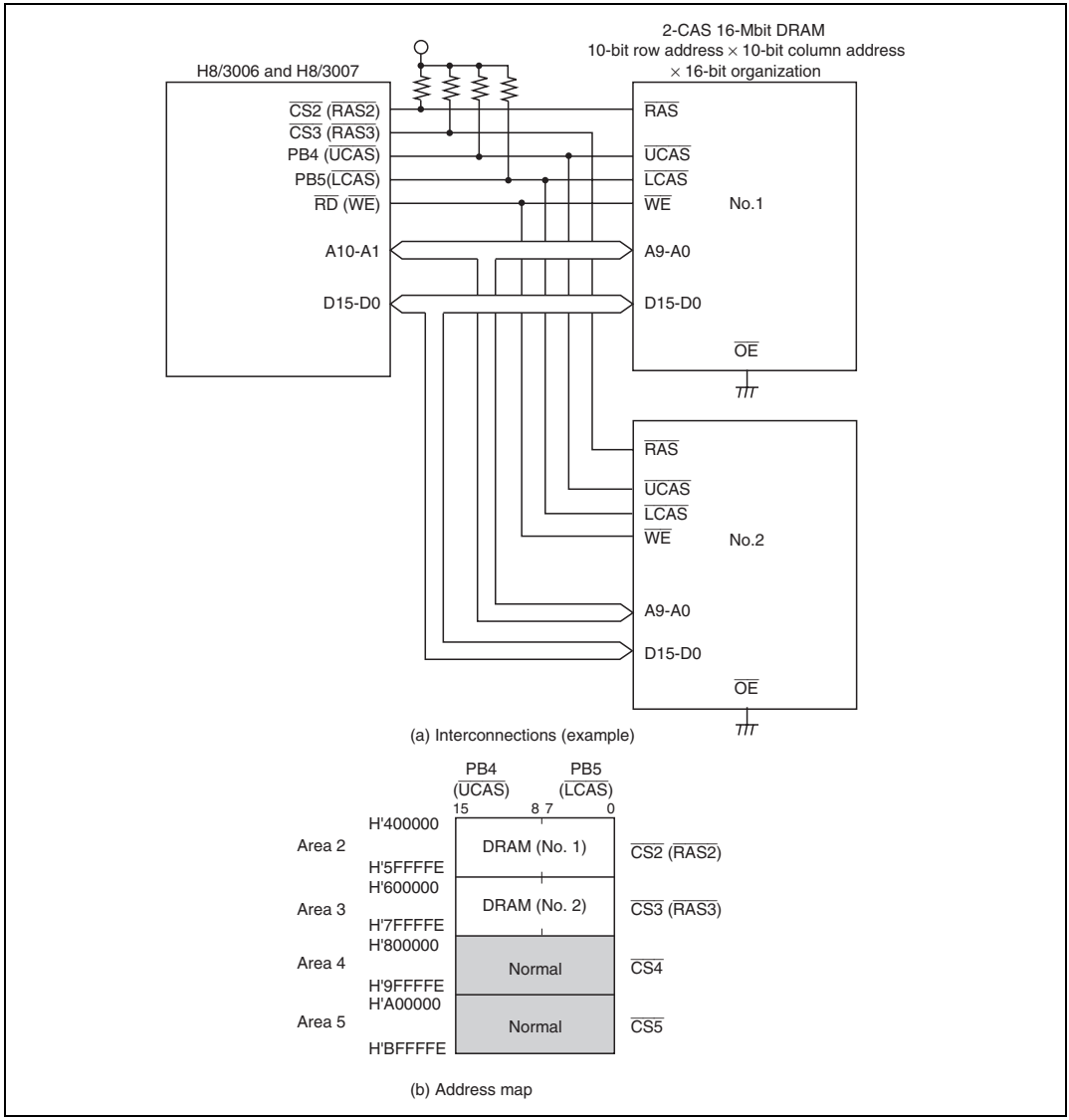


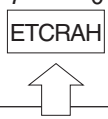
Figure 6.29 Interconnections and Address Map for 2-CAS 16-Mbit DRAMs with × 16-Bit Organization

7.4.6 Block Transfer Mode

In block transfer mode, the A and B channels are combined. One block of a specified size is transferred per request. A designated number of block transfers are executed. Addresses are specified in MARA and MARB. The block area address can be either held fixed or cycled.

Table 7.10 indicates the register functions in block transfer mode.

Table 7.10 Register Functions in Block Transfer Mode

Register	Function	Initial Setting	Operation
	Source address register	Transfer source start address	Incremented or decremented once per transfer, or held fixed
	Destination address register	Transfer destination start address	Incremented or decremented once per transfer, or held fixed
	Block size counter	Block size	Decremented once per transfer until H'00 is reached, then reloaded from ETCRL
	Initial block size	Block size	Held fixed
	Block transfer counter	Number of block transfers	Decremented once per block transfer until H'0000 is reached and the transfer ends

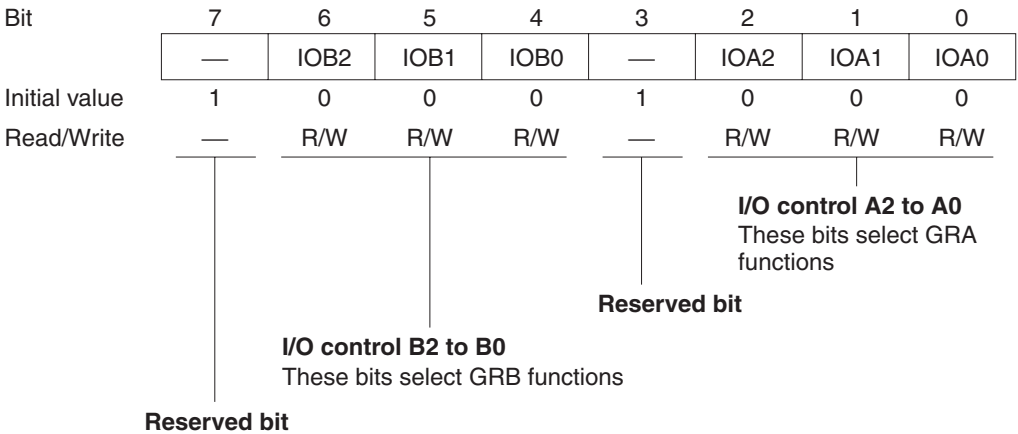
Legend:
MARA: Memory address register A
MARB: Memory address register B
ETCRA: Execute transfer count register A
ETCRB: Execute transfer count register B

The source and destination addresses are both 24-bit addresses. MARA specifies the source address. MARB specifies the destination address. MARA and MARB can be independently incremented, decremented, or held fixed as data is transferred. One of these registers operates as a block area register: even if it is incremented or decremented, it is restored to its initial value at the end of each block transfer. The TMS bit in DTCRB selects whether the block area is the source or destination.

9.2.10 Timer I/O Control Register (TIOR)

TIOR is an 8-bit register. The 16-bit timer has three TIORs, one in each channel.

Channel	Abbreviation	Function
0	TIOR0	TIOR controls the general registers. Some functions differ in PWM mode.
1	TIOR1	
2	TIOR2	



Each TIOR is an 8-bit readable/writable register that selects the output compare or input capture function for GRA and GRB, and specifies the functions of the TIORA and TIORC pins. If the output compare function is selected, TIOR also selects the type of output. If input capture is selected, TIOR also selects the edge or edges of the input capture signal.

TIOR is initialized to H'88 by a reset and in standby mode.

Bit 7—Reserved: This bit cannot be modified and is always read as 1.

16-bit timer Operating Modes

Table 9.7 (a) 16-bit timer Operating Modes (Channel 0)

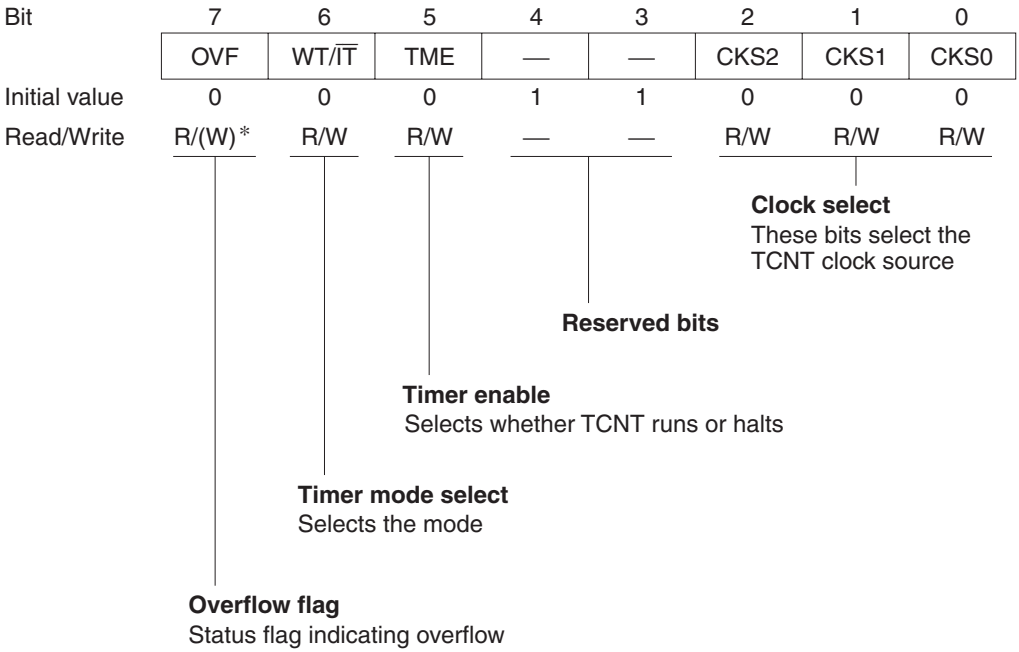
		Register Settings						
		TSNC	TMDR			TIOR0		16TCR0	
Operating Mode		Synchro-nization	MDF	FDIR	PWM	IOA	IOB	Clear Select	Clock Select
Synchronous preset		SYNC0 = 1	—	—	○	○	○	○	○
PWM mode		○	—	—	PWM0 = 1	—	○*	○	○
Output compare A		○	—	—	PWM0 = 0	IOA2 = 0 Other bits unrestricted	○	○	○
Output compare B		○	—	—	○	○	IOB2 = 0 Other bits unrestricted	○	○
Input capture A		○	—	—	PWM0 = 0	IOA2 = 1 Other bits unrestricted	○	○	○
Input capture B		○	—	—	PWM0 = 0	○	IOB2 = 1 Other bits unrestricted	○	○
Counter clearing	By compare match/input capture A	○	—	—	○	○	○	CCLR1 = 0 CCLR0 = 1	○
	By compare match/input capture B	○	—	—	○	○	○	CCLR1 = 1 CCLR0 = 0	○
	Syn-chronous clear	SYNC0 = 1	—	—	○	○	○	CCLR1 = 1 CCLR0 = 1	○

Legend:
○ Setting available (valid)
— Setting does not affect this mode

Note: * The input capture function cannot be used in PWM mode. If compare match A and compare match B occur simultaneously, the compare match signal is inhibited.

12.2.2 Timer Control/Status Register (TCSR)

TCSR is an 8-bit readable and writable register. Its functions include selecting the timer mode and clock source.



Notes: TCSR is write-protected by a password. For details see section 12.2.4, Notes on Register Access.
* Only 0 can be written, to clear the flag.

Bits 7 to 5 are initialized to 0 by a reset and in standby mode. Bits 2 to 0 are initialized to 0 by a reset. In software standby mode bits 2 to 0 are not initialized, but retain their previous values.

Bit 7—Overflow Flag (OVF): This status flag indicates that the timer counter has overflowed from H'FF to H'00.

Bit 7 OVF	Description	
0	[Clearing condition]	(Initial value)
	Cleared by reading OVF when OVF = 1, then writing 0 in OVF	
1	[Setting condition]	
	Set when TCNT changes from H'FF to H'00	

Bit 5—Overrun Error (ORER): Indicates that data reception ended abnormally due to an overrun error.

Bit 5 ORER	Description	
0	Receiving is in progress or has ended normally*1 [Clearing conditions] • The chip is reset or enters standby mode • Read ORER when ORER = 1, then write 0 in ORER	(Initial value)
1	A receive overrun error occurred*2 [Setting condition] Reception of the next serial data ends when RDRF = 1	

Notes: 1. Clearing the RE bit to 0 in SCR does not affect the ORER flag, which retains its previous value.
2. RDR continues to hold the receive data prior to the overrun error, so subsequent receive data is lost. Serial receiving cannot continue while the ORER flag is set to 1. In synchronous mode, serial transmitting is also disabled.

Bit 4—Framing Error (FER)/Error Signal Status (ERS): The function of this bit differs for the normal serial communication interface and for the smart card interface. Its function is switched with the SMIF bit in SCMR.

For serial communication interface (SMIF bit in SCMR cleared to 0): Indicates that data reception ended abnormally due to a framing error in asynchronous mode.

Bit 4 FER	Description	
0	Receiving is in progress or has ended normally*1 [Clearing conditions] • The chip is reset or enters standby mode • Read FER when FER = 1, then write 0 in FER	(Initial value)
1	A receive framing error occurred [Setting condition] The stop bit at the end of the receive data is checked and found to be 0*2	

Notes: 1. Clearing the RE bit to 0 in SCR does not affect the FER flag, which retains its previous value.
2. When the stop bit length is 2 bits, only the first bit is checked. The second stop bit is not checked. When a framing error occurs the SCI transfers the receive data into RDR but does not set the RDRF flag. Serial receiving cannot continue while the FER flag is set to 1. In synchronous mode, serial transmitting is also disabled.

Table 20.3 DC Characteristics (2)

Conditions: VCC = 2.7 to 5.5 V, AVCC = 2.7 to 5.5 V, VREF = 2.7 V to AVCC*1,
VSS = AVSS = 0 V*[1], Ta = –20°C to +75°C (regular specifications),
Ta = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Schmitt trigger input voltages	Port A, $P8_2$ to $P8_0$	V_T^-	$V_{CC} \times 0.2$	—	—	V	
		V_T^+	—	—	$V_{CC} \times 0.7$	V	
		$V_T^+ - V_T^-$	$V_{CC} \times 0.07$	—	—	V	
Input high voltage	$\overline{RES}$, $\overline{STBY}$, NMI, MD_2 to MD_0	V_{IH}	$V_{CC} \times 0.9$	—	V_{CC} +0.3	V	
	EXTAL		$V_{CC} \times 0.7$	—	V_{CC} +0.3	V	
	Port 7		$V_{CC} \times 0.7$	—	AV_{CC} +0.3	V	
	Ports 4, 6, $P8_3$, $P8_4$, $P9_5$ to $P9_0$, port B, D_{15} to D_8		$V_{CC} \times 0.7$	—	V_{CC} +0.3	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD_2 to MD_0	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
	NMI, EXTAL, Ports 4, 6, 7, D_{15} to D_8		–0.3	—	$V_{CC} \times 0.2$	V	V_{CC} < 4.0 V
	$P8_3$, $P8_4$, $P9_5$ to $P9_0$, port B				0.8	V	V_{CC} = 4.0 to 5.5 V
Output high voltage	All output pins (except $\overline{RESO}$)	V_{OH}	V_{CC} –0.5	—	—	V	I_{OH} = –200 μA
			V_{CC} –1.0	—	—	V	I_{OH} = –1 mA
Output low voltage	All output pins (except $\overline{RESO}$)	V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA
	A_{19} to A_0		—	—	1.0	V	I_{OL} = 5 mA (V_{CC} < 4.0 V) I_{OL} = 10 mA (V_{CC} = 4.0 to 5.5 V)
	$\overline{RESO}$		—	—	0.4	V	I_{OL} = 1.6 mA

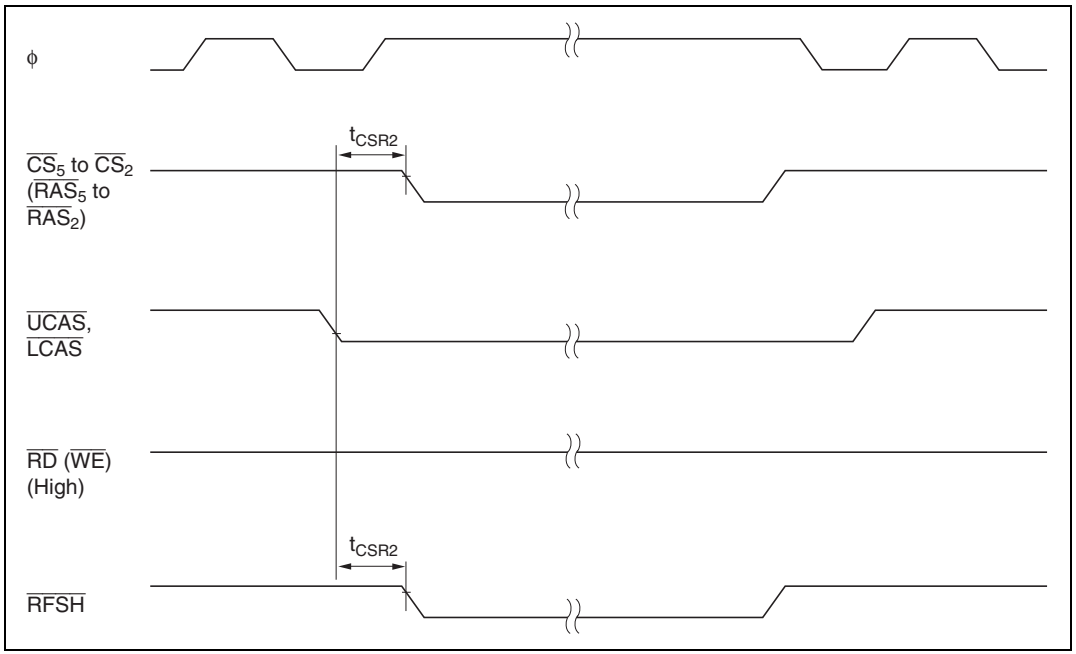


Figure 20.16 DRAM Bus Timing (Self-Refresh)

20.3.5 TPC and I/O Port Timing

Figure 20.17 shows the TPC and I/O port input/output timing.

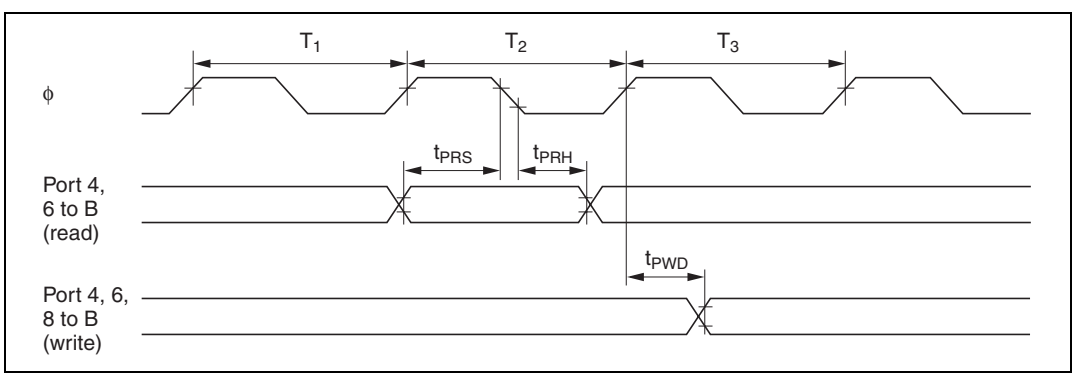


Figure 20.17 TPC and I/O Port Input/Output Timing

20.3.6 Timer Input/Output Timing

The timings of 16-bit and 8-bit timer are shown as follows:

- Timer input/output timing
  Figure 20.18 shows the timer input/output timing.
- Timer external clock input timing
  Figure 20.19 shows the timer external clock input timing.

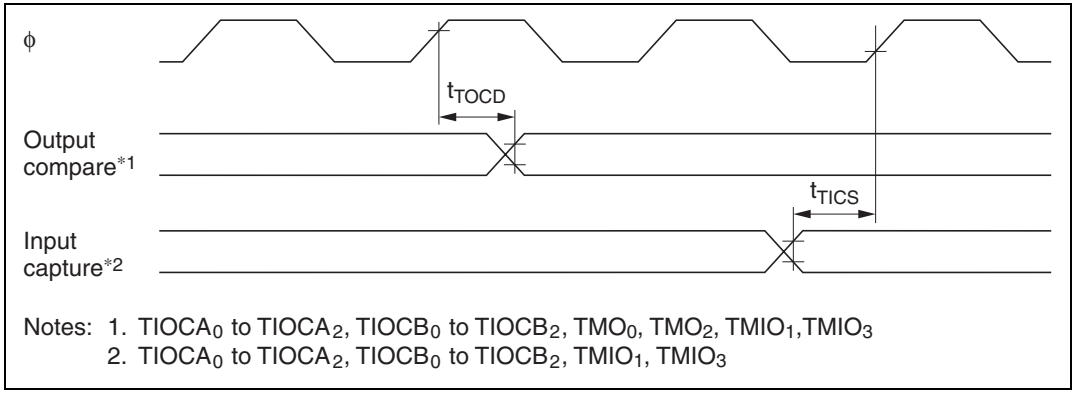


Figure 20.18 Timer Input/Output Timing

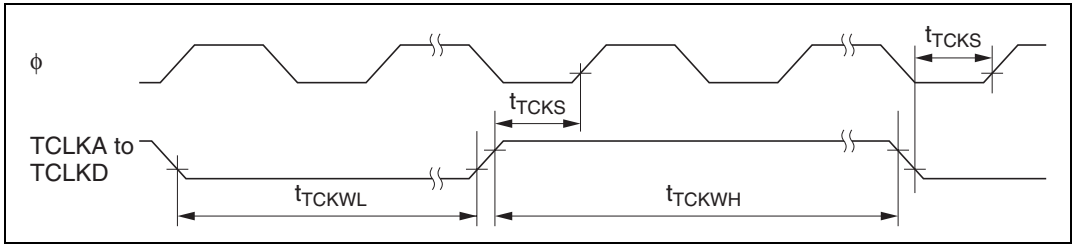


Figure 20.19 Timer External Clock Input Timing

Appendix A Instruction Set

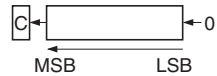
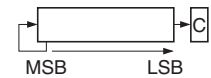
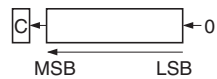
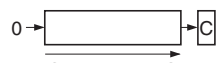
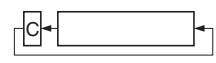
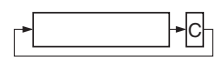
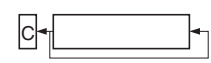
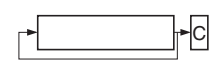
A.1 Instruction List

Operand Notation

Symbol	Description
Rd	General destination register
Rs	General source register
Rn	General register
ERd	General destination register (address register or 32-bit register)
ERs	General source register (address register or 32-bit register)
ERn	General register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
PC	Program counter
SP	Stack pointer
CCR	Condition code register
N	N (negative) flag in CCR
Z	Z (zero) flag in CCR
V	V (overflow) flag in CCR
C	C (carry) flag in CCR
disp	Displacement
→	Transfer from the operand on the left to the operand on the right, or transition from the state on the left to the state on the right
+	Addition of the operands on both sides
–	Subtraction of the operand on the right from the operand on the left
×	Multiplication of the operands on both sides
÷	Division of the operand on the left by the operand on the right
∧	Logical AND of the operands on both sides
∨	Logical OR of the operands on both sides
⊕	Exclusive logical OR of the operands on both sides
¬	NOT (logical complement)
( ), < >	Contents of operand

Note: General registers include 8-bit registers (R0H to R7H and R0L to R7L) and 16-bit registers (R0 to R7 and E0 to E7).

4. Shift instructions

Mnemonic	Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code						No. of States*1	
		#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	—		I	H	N	Z	V	C	Normal	Advanced
SHAL.B Rd	B		2									—	—	↕	↕	↕	↕	2	
SHAL.W Rd	W		2									—	—	↕	↕	↕	↕	2	
SHAL.L ERd	L		2									—	—	↕	↕	↕	↕	2	
SHAR.B Rd	B		2									—	—	↕	↕	0	↕	2	
SHAR.W Rd	W		2									—	—	↕	↕	0	↕	2	
SHAR.L ERd	L		2									—	—	↕	↕	0	↕	2	
SHLL.B Rd	B		2									—	—	↕	↕	0	↕	2	
SHLL.W Rd	W		2									—	—	↕	↕	0	↕	2	
SHLL.L ERd	L		2									—	—	↕	↕	0	↕	2	
SHLR.B Rd	B		2									—	—	↕	↕	0	↕	2	
SHLR.W Rd	W		2									—	—	↕	↕	0	↕	2	
SHLR.L ERd	L		2									—	—	↕	↕	0	↕	2	
ROTXL.B Rd	B		2									—	—	↕	↕	0	↕	2	
ROTXL.W Rd	W		2									—	—	↕	↕	0	↕	2	
ROTXL.L ERd	L		2									—	—	↕	↕	0	↕	2	
ROTXR.B Rd	B		2									—	—	↕	↕	0	↕	2	
ROTXR.W Rd	W		2									—	—	↕	↕	0	↕	2	
ROTXR.L ERd	L		2									—	—	↕	↕	0	↕	2	
ROTL.B Rd	B		2									—	—	↕	↕	0	↕	2	
ROTL.W Rd	W		2									—	—	↕	↕	0	↕	2	
ROTL.L ERd	L		2									—	—	↕	↕	0	↕	2	
ROTR.B Rd	B		2									—	—	↕	↕	0	↕	2	
ROTR.W Rd	W		2									—	—	↕	↕	0	↕	2	
ROTR.L ERd	L		2									—	—	↕	↕	0	↕	2	

Mnemonic	Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code						No. of States*1	
		#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	|		I	H	N	Z	V	C	Normal	Advanced
BLD #xx:3, @ERd	B			4							(#xx:3 of @ERd) → C	—	—	—	—	—	↕	6	
BLD #xx:3, @aa:8	B						4				(#xx:3 of @aa:8) → C	—	—	—	—	—	↕	6	
BILD #xx:3, Rd	B		2								¬(#xx:3 of Rd8) → C	—	—	—	—	—	↕	2	
BILD #xx:3, @ERd	B			4							¬(#xx:3 of @ERd) → C	—	—	—	—	—	↕	6	
BILD #xx:3, @aa:8	B						4				¬(#xx:3 of @aa:8) → C	—	—	—	—	—	↕	6	
BST #xx:3, Rd	B		2								C → (#xx:3 of Rd8)	—	—	—	—	—	—	2	
BST #xx:3, @ERd	B			4							C → (#xx:3 of @ERd24)	—	—	—	—	—	—	8	
BST #xx:3, @aa:8	B						4				C → (#xx:3 of @aa:8)	—	—	—	—	—	—	8	
BIST #xx:3, Rd	B		2								¬C → (#xx:3 of Rd8)	—	—	—	—	—	—	2	
BIST #xx:3, @ERd	B			4							¬C → (#xx:3 of @ERd24)	—	—	—	—	—	—	8	
BIST #xx:3, @aa:8	B						4				¬C → (#xx:3 of @aa:8)	—	—	—	—	—	—	8	
BAND #xx:3, Rd	B		2								C∧(#xx:3 of Rd8) → C	—	—	—	—	—	↕	2	
BAND #xx:3, @ERd	B			4							C∧(#xx:3 of @ERd24) → C	—	—	—	—	—	↕	6	
BAND #xx:3, @aa:8	B						4				C∧(#xx:3 of @aa:8) → C	—	—	—	—	—	↕	6	
BIAND #xx:3, Rd	B		2								C∧¬(#xx:3 of Rd8) → C	—	—	—	—	—	↕	2	
BIAND #xx:3, @ERd	B			4							C∧¬(#xx:3 of @ERd24) → C	—	—	—	—	—	↕	6	
BIAND #xx:3, @aa:8	B						4				C∧¬(#xx:3 of @aa:8) → C	—	—	—	—	—	↕	6	
BOR #xx:3, Rd	B		2								C∨(#xx:3 of Rd8) → C	—	—	—	—	—	↕	2	
BOR #xx:3, @ERd	B			4							C∨(#xx:3 of @ERd24) → C	—	—	—	—	—	↕	6	
BOR #xx:3, @aa:8	B						4				C∨(#xx:3 of @aa:8) → C	—	—	—	—	—	↕	6	
BIOR #xx:3, Rd	B		2								C∨¬(#xx:3 of Rd8) → C	—	—	—	—	—	↕	2	
BIOR #xx:3, @ERd	B			4							C∨¬(#xx:3 of @ERd24) → C	—	—	—	—	—	↕	6	
BIOR #xx:3, @aa:8	B						4				C∨¬(#xx:3 of @aa:8) → C	—	—	—	—	—	↕	6	
BXOR #xx:3, Rd	B		2								C⊕(#xx:3 of Rd8) → C	—	—	—	—	—	↕	2	
BXOR #xx:3, @ERd	B			4							C⊕(#xx:3 of @ERd24) → C	—	—	—	—	—	↕	6	
BXOR #xx:3, @aa:8	B						4				C⊕(#xx:3 of @aa:8) → C	—	—	—	—	—	↕	6	
BIXOR #xx:3, Rd	B		2								C⊕¬(#xx:3 of Rd8) → C	—	—	—	—	—	↕	2	
BIXOR #xx:3, @ERd	B			4							C⊕¬(#xx:3 of @ERd24) → C	—	—	—	—	—	↕	6	
BIXOR #xx:3, @aa:8	B						4				C⊕¬(#xx:3 of @aa:8) → C	—	—	—	—	—	↕	6	

TMDR—Timer Mode Register **H'FFF62** **16-Bit Timer (Common)**