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	H8SX
Core Size	32-Bit Single-Core
Speed	50MHz
Connectivity	I²C, IrDA, SCI, SmartCard, USB
Peripherals	DMA, PWM, WDT
Number of I/O	92
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	40K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 8x10b; D/A 2x8b
Oscillator Type	Internal
Operating Temperature	-20°C ~ 75°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LFQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/df61664w50fpv

Bit	Bit Name	Initial Value	R/W	Description
2	Z	Undefined	R/W	Zero Flag Set to 1 to indicate zero data, and cleared to 0 to indicate non-zero data.
1	V	Undefined	R/W	Overflow Flag Set to 1 when an arithmetic overflow occurs, and cleared to 0 otherwise.
0	C	Undefined	R/W	Carry Flag Set to 1 when a carry occurs, and cleared to 0 otherwise. A carry has the following types: • Carry from the result of addition • Borrow from the result of subtraction • Carry from the result of shift or rotation The carry flag is also used as a bit accumulator by bit manipulation instructions.

2.5.4 Extended Control Register (EXR)

EXR is an 8-bit register that contains the trace bit (T) and three interrupt mask bits (I2 to I0).

Operations can be performed on the EXR bits by the LDC, STC, ANDC, ORC, and XORC instructions.

For details, see section 4, Exception Handling.

Bit	Bit Name	Initial Value	R/W	Description
7	T	0	R/W	Trace Bit When this bit is set to 1, a trace exception is generated each time an instruction is executed. When this bit is cleared to 0, instructions are executed in sequence.
6 to 3	—	All 1	R/W	Reserved These bits are always read as 1.
2	I2	1	R/W	Interrupt Mask Bits
1	I1	1	R/W	These bits designate the interrupt mask level (0 to 7).
0	I0	1	R/W	

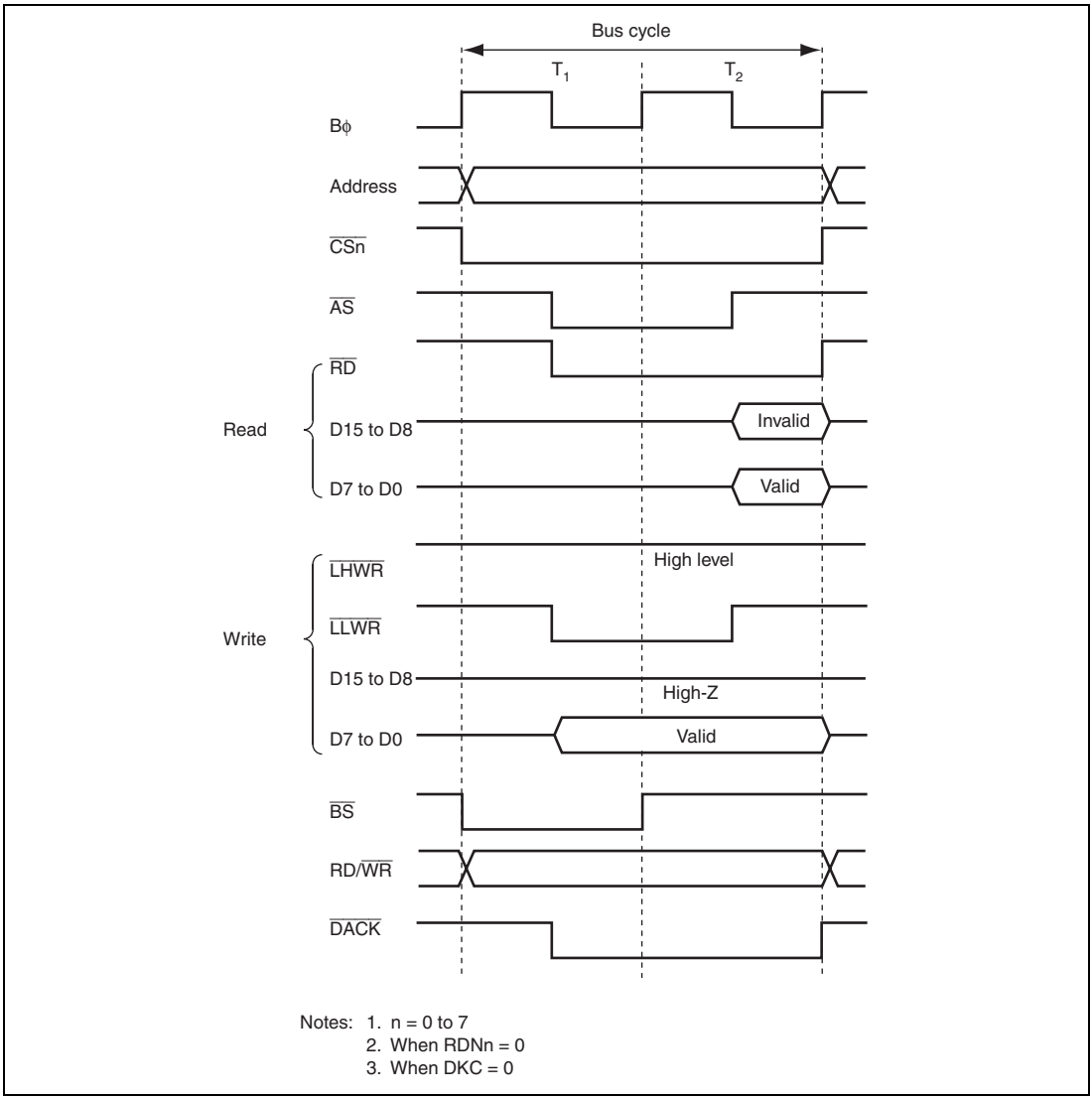


Figure 6.16 16-Bit 2-State Access Space Bus Timing (Byte Access for Odd Address)

(4) When CKSPE = 1

When the SDRAM space is read, the read data settling cycle can be inserted for one cycle using the clock suspend mode. To enter the clock suspend mode, set the OEE bit to 1, and connect the CKE pin.

Figure 6.87 shows an output timing example of the $\overline{\text{DACK}}$ signal when CKSPE = 1 with DDS = 1 and DKC = 0.

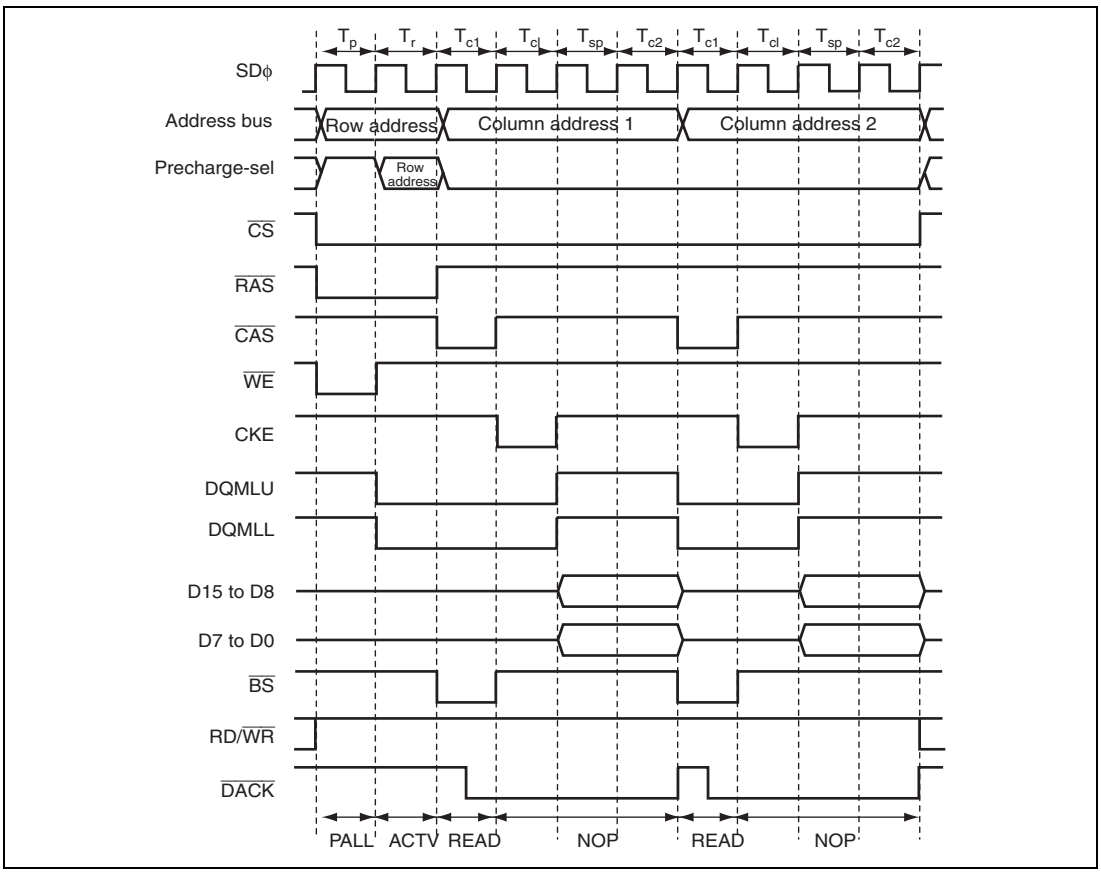


Figure 6.87 Output Timing Example of $\overline{\text{DACK}}$ when CKSPE = 1 (Read, CAS Latency = 2)

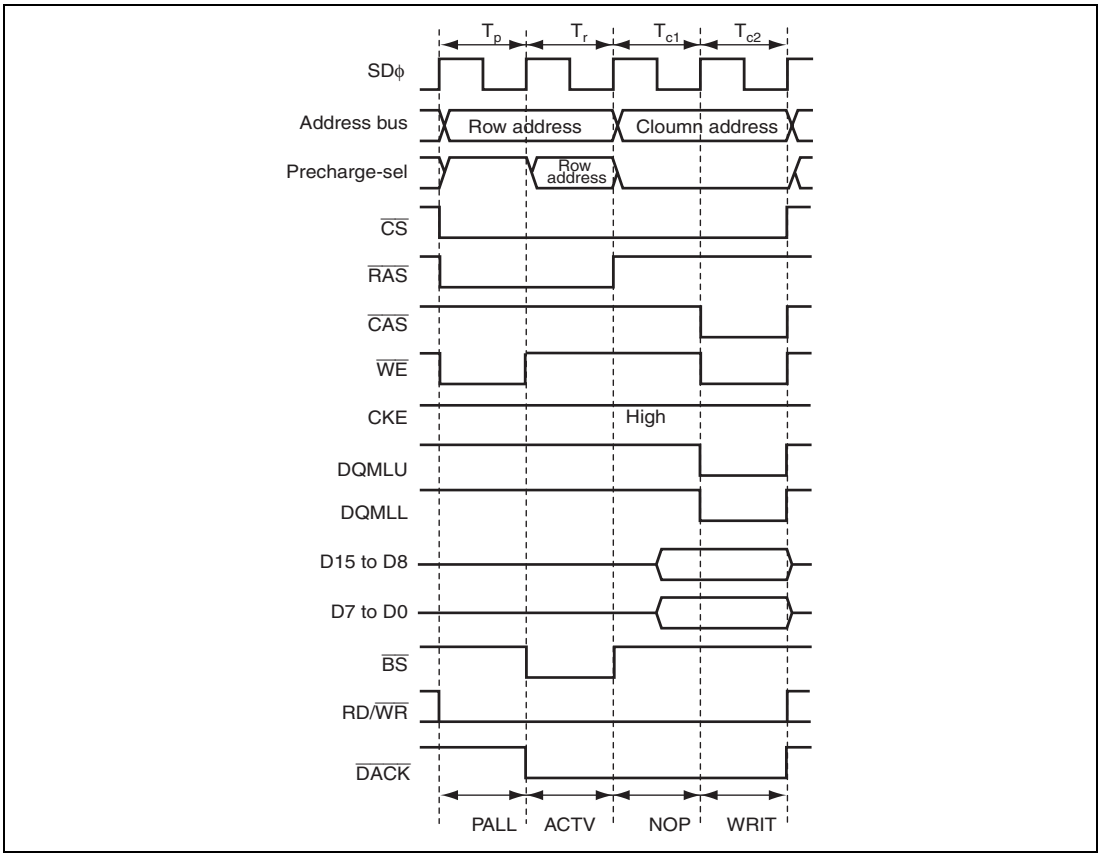


Figure 6.89 Output Timing Example of $\overline{\text{DACK}}$ when DKC = 1 and DDS = 0 (Write)

(1) Consecutive Reads in Different Areas

If consecutive reads in different areas occur while bit IDLS1 in IDLCR is set to 1, idle cycles specified by bits IDLCA1 and IDLCA0 when bit IDLSELn in IDLCR is cleared to 0, or bits IDLCB1 and IDLCB0 when bit IDLSELn is set to 1 are inserted at the start of the second read cycle (n = 0 to 7).

Figure 6.90 shows an example of the operation in this case. In this example, bus cycle A is a read cycle for ROM with a long output floating time, and bus cycle B is a read cycle for SRAM, each being located in a different area. In (a), an idle cycle is not inserted, and a conflict occurs in bus cycle B between the read data from ROM and that from SRAM. In (b), an idle cycle is inserted, and a data conflict is prevented.

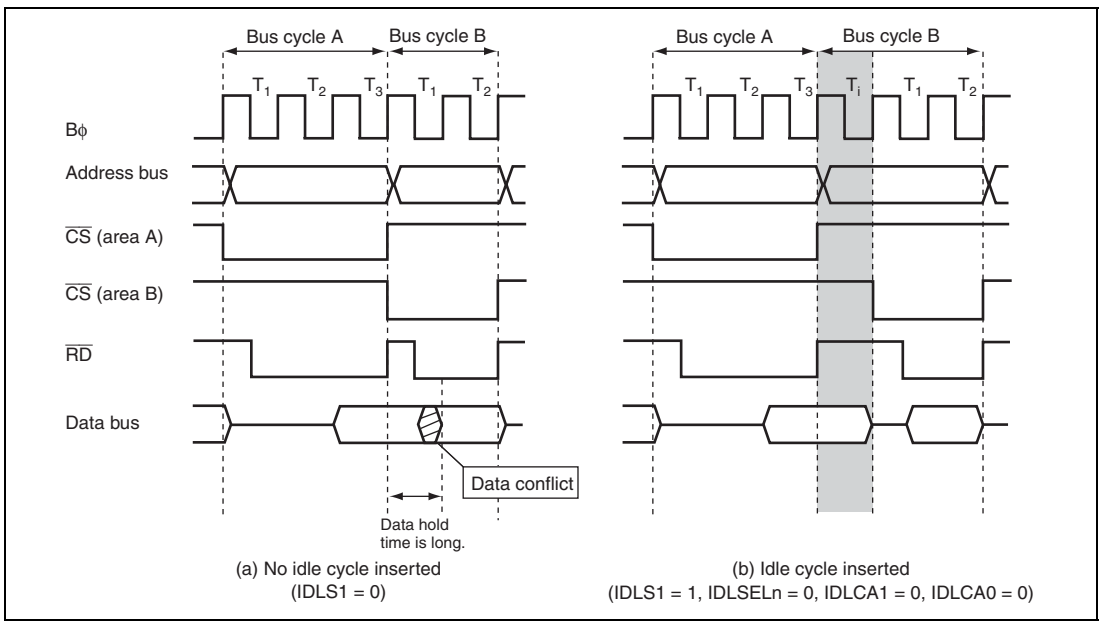


Figure 6.90 Example of Idle Cycle Operation (Consecutive Reads in Different Areas)

7.5.11 Bus Cycles in Single Address Mode

(1) Single Address Mode (Read and Cycle Stealing)

In single address mode, one byte, one word, or one longword of data is transferred at a single transfer request and after the transfer the bus is released temporarily. One bus cycle or more by the CPU or DTC are executed in the bus released cycles.

In figure 7.34, the $\overline{\text{TEND}}$ signal output is enabled and data is transferred in bytes from the external 8-bit 2-state access space to the external device in single address mode (read).

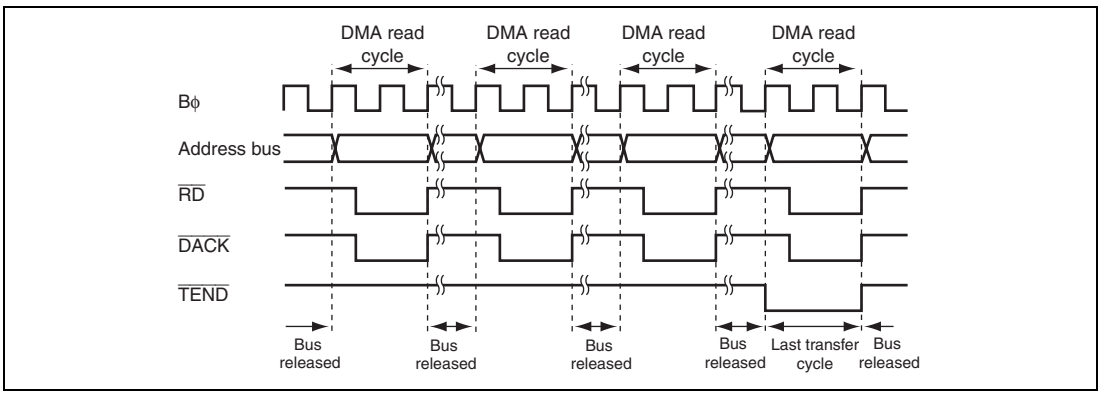


Figure 7.34 Example of Transfer in Single Address Mode (Byte Read)

(5) P33/PO11/TIOCC0/TIOCD0/TCLKB-A/$\overline{\text{DREQ1}}$-B

The pin function is switched as shown below according to the combination of the TPU and PPG register settings and P33DDR bit setting.

		Setting		
		TPU	PPG	I/O Port
Module Name	Pin Function	TIOCD0_OE	PO11_OE	P33DDR
TPU	TIOCD0 output	1	—	—
PPG	PO11 output	0	1	—
I/O port	P33 output	0	0	1
	P33 input (initial setting)	0	0	0

(6) P32/PO10/TIOCC0/TCLKA-A/$\overline{\text{DACK0}}$-B

The pin function is switched as shown below according to the combination of the DMAC, TPU, and PPG register settings and P32DDR bit setting.

		Setting			
		DMAC	TPU	PPG	I/O Port
Module Name	Pin Function	DACK0B_OE	TIOCC0_OE	PO10_OE	P32DDR
DMAC	$\overline{\text{DACK0}}$-B output	1	—	—	—
TPU	TIOCC0 output	0	1	—	—
PPG	PO10 output	0	0	1	—
I/O port	P32 output	0	0	0	1
	P32 input (initial setting)	0	0	0	0

9.2.14 Port M

(1) PM4

The pin function is switched as shown below according to the combination of the USB register setting and PM4DDR bit setting.

		Setting	
		USB	I/O Port
Module Name	Pin Function	PULLUP_E	PM4DDR
USB	PULLUP control output	1	—
I/O port	PM4 output	0	1
	PM4 input (initial setting)	0	0

(2) PM3

The pin function is switched as shown below according to the combination of the PM3DDR bit setting.

		Setting
		I/O Port
Module Name	Pin Function	PM3DDR
I/O port	PM3 output	1
	PM3 input (initial setting)	0

(3) PM2

The pin function is switched as shown below according to the combination of the PM2DDR bit setting.

		Setting
		I/O Port
Module Name	Pin Function	PM2DDR
I/O port	PM2 output	1
	PM2 input (initial setting)	0

(b) Free-running count operation and periodic count operation

Immediately after a reset, the TPU's TCNT counters are all designated as free-running counters. When the relevant bit in TSTR is set to 1 the corresponding TCNT counter starts up-count operation as a free-running counter. When TCNT overflows (changes from H'FFFF to H'0000), the TCFV bit in TSR is set to 1. If the value of the corresponding TCIEV bit in TIER is 1 at this point, the TPU requests an interrupt. After overflow, TCNT starts counting up again from H'0000.

Figure 10.3 illustrates free-running counter operation.

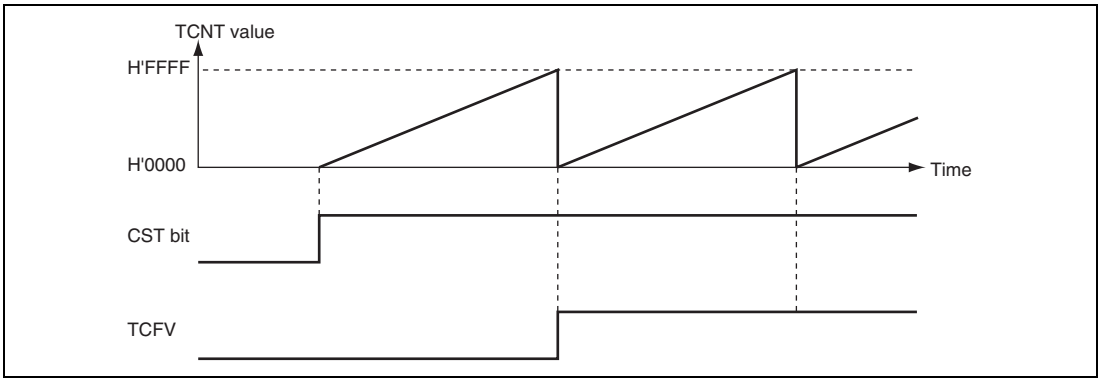


Figure 10.3 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT counter for the relevant channel performs periodic count operation. The TGR register for setting the period is designated as an output compare register, and counter clearing by compare match is selected by means of bits CCLR2 to CCLR0 in TCR. After the settings have been made, TCNT starts count-up operation as a periodic counter when the corresponding bit in TSTR is set to 1. When the count value matches the value in TGR, the TGF bit in TSR is set to 1 and TCNT is cleared to H'0000.

If the value of the corresponding TGIE bit in TIER is 1 at this point, the TPU requests an interrupt. After a compare match, TCNT starts counting up again from H'0000.

14.6.5 Internal Reset in Watchdog Timer Mode

This LSI is not reset internally if TCNT overflows while the RSTE bit is cleared to 0 during watchdog timer mode operation, but TCNT and TCSR of the WDT are reset.

TCNT, TCSR, and RSTCR cannot be written to while the $\overline{\text{WDTOVF}}$ signal is low. Also note that a read of the WOVF flag is not recognized during this period. To clear the WOVF flag, therefore, read TCSR after the $\overline{\text{WDTOVF}}$ signal goes high, then write 0 to the WOVF flag.

14.6.6 System Reset by $\overline{\text{WDTOVF}}$ Signal

If the $\overline{\text{WDTOVF}}$ signal is input to the $\overline{\text{RES}}$ pin, this LSI will not be initialized correctly. Make sure that the $\overline{\text{WDTOVF}}$ signal is not input logically to the $\overline{\text{RES}}$ pin. To reset the entire system by means of the $\overline{\text{WDTOVF}}$ signal, use a circuit like that shown in figure 14.6.

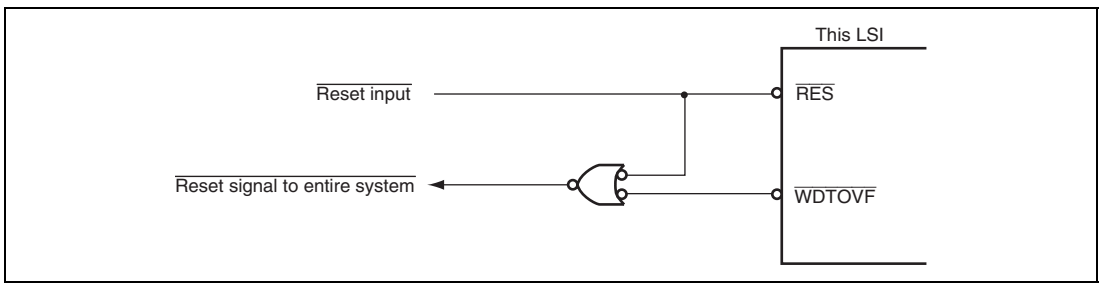


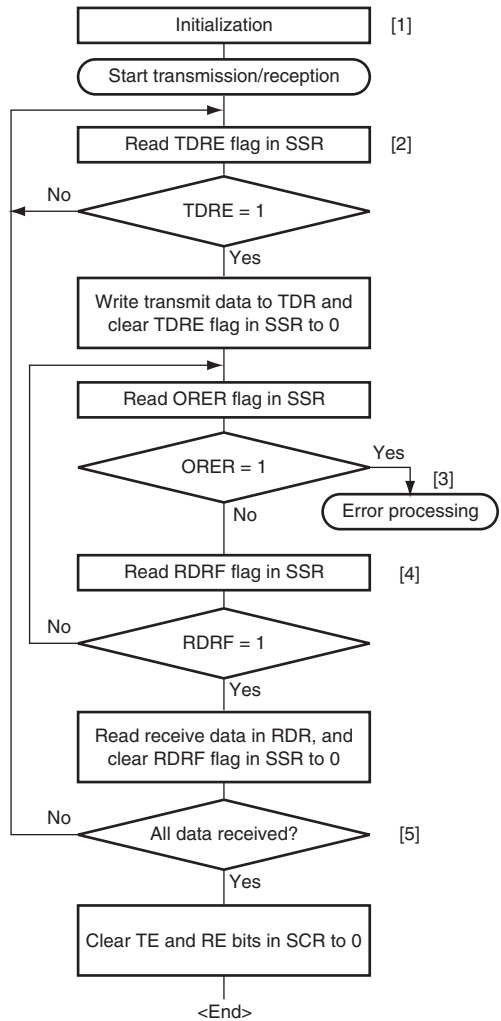
Figure 14.6 Circuit for System Reset by $\overline{\text{WDTOVF}}$ Signal (Example)

14.6.7 Transition to Watchdog Timer Mode or Software Standby Mode

When the WDT operates in watchdog timer mode, a transition to software standby mode is not made even when the SLEEP instruction is executed when the SSBY bit in SBYCR is set to 1. Instead, a transition to sleep mode is made.

To transit to software standby mode, the SLEEP instruction must be executed after halting the WDT (clearing the TME bit to 0).

When the WDT operates in interval timer mode, a transition to software standby mode is made through execution of the SLEEP instruction when the SSBY bit in SBYCR is set to 1.



Note: When switching from transmit or receive operation to simultaneous transmit and receive operations, first clear the TE bit and RE bit to 0, then set both these bits to 1 simultaneously.

[1] SCI initialization:
The TxD pin is designated as the transmit data output pin, and the RxD pin is designated as the receive data input pin, enabling simultaneous transmit and receive operations.

[2] SCI state check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0. Transition of the TDRE flag from 0 to 1 can also be identified by a TXI interrupt.

[3] Receive error processing:
If a receive error occurs, read the ORER flag in SSR, and after performing the appropriate error processing, clear the ORER flag to 0. Reception cannot be resumed if the ORER flag is set to 1.

[4] SCI state check and receive data read:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial transmission/reception continuation procedure:
To continue serial transmission/reception, before the MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0. Also, before the MSB (bit 7) of the current frame is transmitted, read 1 from the TDRE flag to confirm that writing is possible. Then write data to TDR and clear the TDRE flag to 0.
However, the TDRE flag is checked and cleared automatically when the DMAC or DTC is initiated by a transmit data empty interrupt (TXI) request and writes data to TDR. Similarly, the RDRF flag is cleared automatically when the DMAC or DTC is initiated by a receive data full interrupt (RXI) and reads data from RDR.

Figure 15.23 Sample Flowchart of Simultaneous Serial Transmission and Reception

Bit	Bit Name	Initial Value	R/W	Description
4	EP2EMPTY	1	R	EP2 FIFO Empty This bit is set when at least one of the dual endpoint 2 transmit FIFO buffers is ready for transmit data to be written. This is a status bit, and cannot be cleared.
3	SETUPTS	0	R/W	Setup Command Receive Complete This bit is set to 1 when endpoint 0 receives successfully a setup command requiring decoding on the application side, and returns an ACK handshake to the host. (When the CPU is used to clear this flag by writing 0 while the corresponding interrupt is enabled, be sure to read the flag after writing 0 to it.)
2	EP0oTS	0	R/W	EP0o Receive Complete This bit is set to 1 when endpoint 0 receives data from the host successfully, stores the data in the FIFO buffer, and returns an ACK handshake to the host. (When the CPU is used to clear this flag by writing 0 while the corresponding interrupt is enabled, be sure to read the flag after writing 0 to it.)
1	EP0iTR	0	R/W	EP0i Transfer Request This bit is set if there is no valid transmit data in the FIFO buffer when an IN token for endpoint 0 is received from the host. A NACK handshake is returned to the host until data is written to the FIFO buffer and packet transmission is enabled. (When the CPU is used to clear this flag by writing 0 while the corresponding interrupt is enabled, be sure to read the flag after writing 0 to it.)
0	EP0iTS	0	R/W	EP0i Transmit Complete This bit is set when data is transmitted to the host from endpoint 0 and an ACK handshake is returned. (When the CPU is used to clear this flag by writing 0 while the corresponding interrupt is enabled, be sure to read the flag after writing 0 to it.)

16.3.17 EP1 Receive Data Size Register (EPSZ1)

EPSZ1 is a receive data size resister for endpoint 1. EPSZ1 indicates the number of bytes received from the host. The FIFO for endpoint 1 has a dual-buffer configuration. The size of the received data indicated by this register is the size of the currently selected side (can be read by CPU).

Bit	7	6	5	4	3	2	1	0
Bit Name	—	—	—	—	—	—	—	—
Initial Value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	—	All 0	R	Number of received bytes for endpoint 1

16.3.18 Trigger Register (TRG)

TRG generates one-shot triggers to control the transfer sequence for each endpoint.

Bit	7	6	5	4	3	2	1	0
Bit Name	—	EP3 PKTE	EP1 RDFN	EP2 PKTE	—	EP0s RDFN	EP0o RDFN	EP0i PKTE
Initial Value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	—	W	W	W	—	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	Reserved The write value should always be 0.
6	EP3 PKTE	Undefined	W	EP3 Packet Enable After one packet of data has been written to the endpoint 3 transmit FIFO buffer, the transmit data is fixed by writing 1 to this bit.

Table 21.3 Registers/Parameters and Target Modes

Register/Parameter		Down-load	Initiali-zation	Program-ming	Erasure	Read	RAM Emulation
Programming/ erasing interface registers	FCCS	O	—	—	—	—	—
	FPCS	O	—	—	—	—	—
	FECS	O	—	—	—	—	—
	FKEY	O	—	O	O	—	—
	FTDAR	O	—	—	—	—	—
Programming/ erasing interface parameters	DPFR	O	—	—	—	—	—
	FPFR	—	O	O	O	—	—
	FPEFEQ	—	O	—	—	—	—
	FMPAR	—	—	O	—	—	—
	FMPDR	—	—	O	—	—	—
	FEBS	—	—	—	O	—	—
RAM emulation	RAMER	—	—	—	—	—	O

21.6.1 Programming/Erasing Interface Registers

The programming/erasing interface registers are 8-bit registers that can be accessed only in bytes. These registers are initialized by a power-on reset.

(1) Flash Code Control/Status Register (FCCS)

FCCS monitors errors during programming/erasing the flash memory and requests the on-chip program to be downloaded to the on-chip RAM.

Bit	7	6	5	4	3	2	1	0
Bit Name	—	—	—	FLER	—	—	—	SCO
Initial Value	1	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	(R)/W

Bit	Bit Name	Initial Value	R/W	Description
1	WA	—	R/W	Write Address Error Detect When the following items are specified as the start address of the programming destination, an error occurs. • An area other than flash memory • The specified address is not aligned with the 128-byte boundary (lower eight bits of the address are other than H'00 and H'80) 0: Setting of the start address of the programming destination is normal 1: Setting of the start address of the programming destination is abnormal
0	SF	—	R/W	Success/Fail Returns the programming result. 0: Programming has ended normally (no error) 1: Programming has ended abnormally (error occurs)

(c) Erasure

FPFR indicates the return value of the erasure result.

Bit	7	6	5	4	3	2	1	0
Bit Name	—	MD	EE	FK	EB	—	—	SF

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	—	Unused Returns 0.
6	MD	—	R/W	Erasure Mode Related Setting Error Detect Detects the error protection state and returns the result. When the error protection state is entered, this bit is set to 1. Whether the error protection state is entered or not can be confirmed with the FLER bit in FCCS. For conditions to enter the error protection state, see section 21.8.3, Error Protection. 0: Normal operation (FLER = 0) 1: Error protection state, and programming cannot be performed (FLER = 1)

21.7.2 USB Boot Mode

USB boot mode executes programming/erasing of the user MAT by means of the control command and program data transmitted from the externally connected host via the USB.

In USB boot mode, the tool for transmitting the control command and program data, and the program data must be prepared in the host. The system configuration in USB boot mode is shown in figure 21.9. Interrupts are ignored in USB boot mode. Configure the user system so that interrupts do not occur.

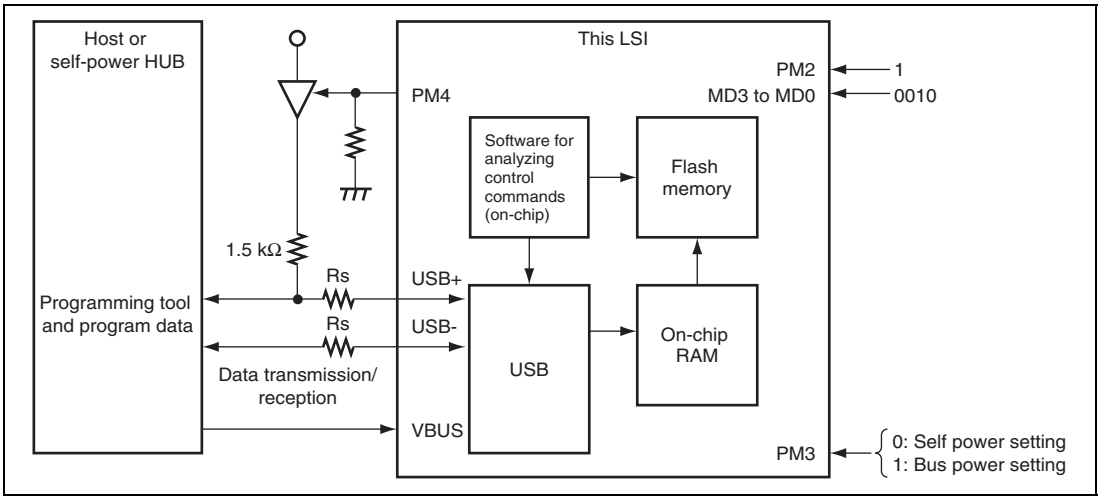


Figure 21.9 System Configuration in USB Boot Mode

Error Response	H'D0	ERROR

- Error response, H'D0, (one byte): Error response for 128-byte programming
- ERROR: (one byte): Error code

  H'11: Checksum Error

  H'2A: Address error

  The address is not in the specified MAT.

  H'53: Programming error

  A programming error has occurred and programming cannot be continued.

The specified address should match the unit for programming of data. For example, when the programming is in 128-byte units, the lower eight bits of the address should be H'00 or H'80. When there are less than 128 bytes of data to be programmed, the host should fill the rest with H'FF.

Sending the 128-byte programming command with the address of H'FFFFFFFF will stop the programming operation. The boot program will interpret this as the end of the programming and wait for selection of programming or erasing.

Command	H'50	Address	SUM

- Command, H'50, (one byte): 128-byte programming
- Programming Address (four bytes): End code is H'FF, H'FF, H'FF, H'FF.
- SUM (one byte): Checksum

Response	H'06

- Response, H'06, (one byte): Response to 128-byte programming
  On completion of programming, the boot program will return ACK.

Error Response	H'D0	ERROR

- Error Response, H'D0, (one byte): Error response for 128-byte programming
- ERROR: (one byte): Error code

  H'11: Checksum error

  H'53: Programming error

  An error has occurred in programming and programming cannot be continued.

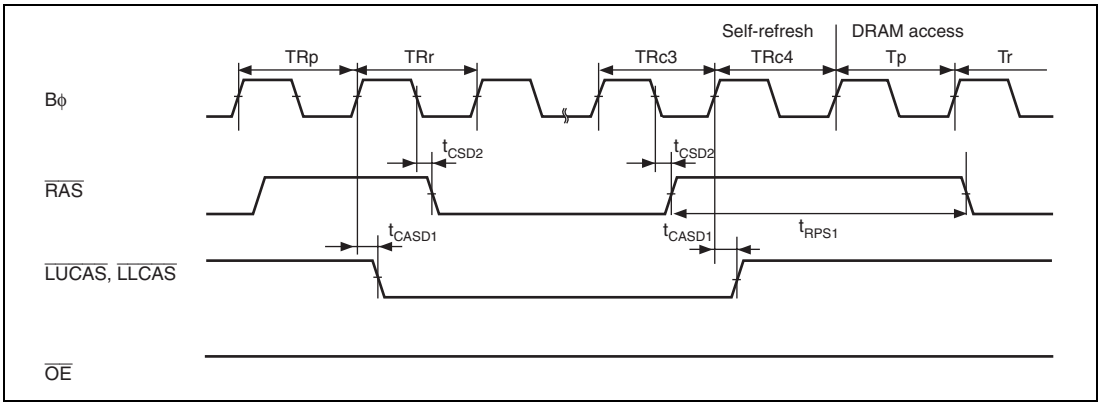


Figure 25.28 Self-Refresh Timing (After Leaving Software Standby: RAST = 1)

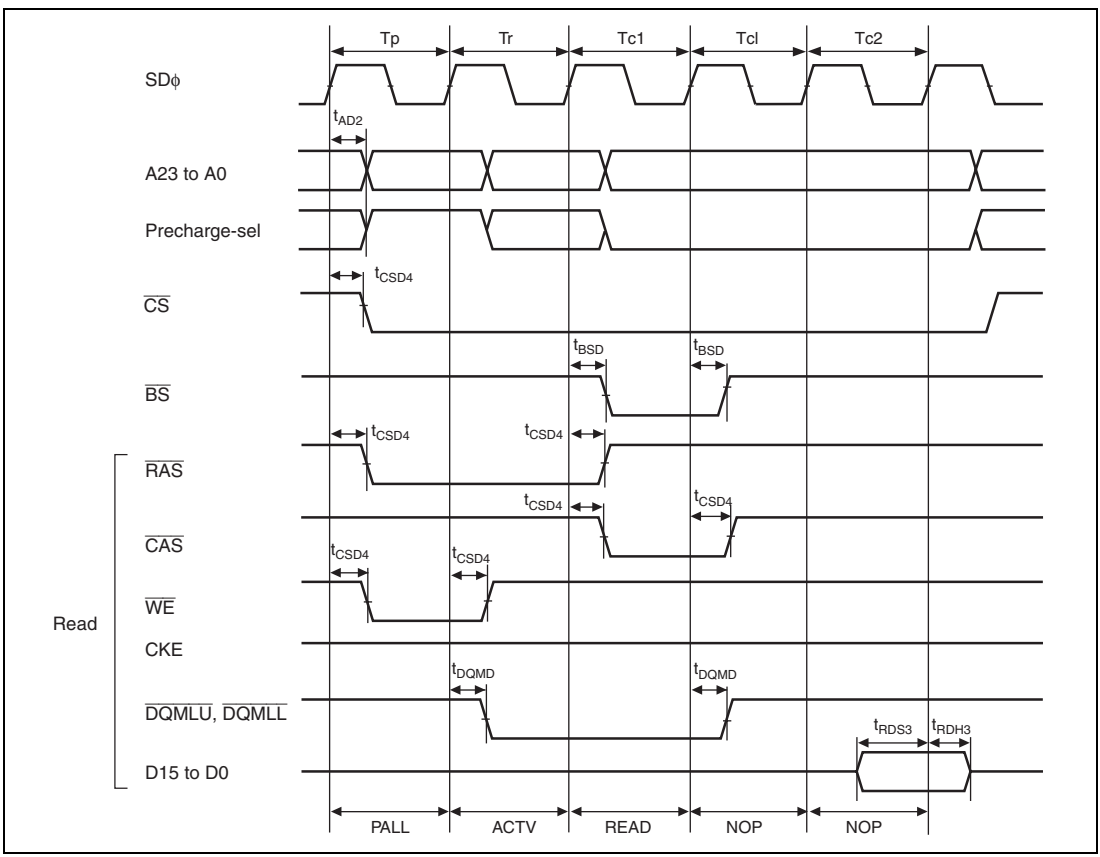


Figure 25.29 Synchronous DRAM Basic Read Access Timing (CAS Latency 2)