CHAPTER 6 DMA CONTROLLER

The μPD70325 and 70335 each contain a two-channel DMA controller to enable direct addressing of the 1-Mbyte memory space.

6.1 Pin Functions

The following pins are provided for the DMA controller. Because all of these pins are also used as ports, the corresponding bits of the port 2 mode control register (PMC2) must be set to 1 to use these pins for the DMA controller.

(1) DMARQ0 and DMARQ1 (P20 and P23)

These are active-high DMA request input pins.

To stop the DMA transfer in the demand release mode using the DMARQn pin, set so that a wait state of 2-state can be inserted in the space corresponding to the DMA transfer. Also, set the DMARQn to low level within 1-clock of the falling edge of $\overline{\text{DMAAKn}}$.

(2) $\overline{\text{DMAAK0}}$ and $\overline{\text{DMAAK1}}$ (P21 and P24)

These are active-low DMA response output pins.

However, the signals are not output during DMA transfer between memory and memory (burst mode or single-step mode).

(3) $\overline{\text{TC0}}$ and $\overline{\text{TC1}}$ (P22 and P25)

These are active-low DMA completion output pins.

Output occurs when a borrow is generated in TC0 and TC1 decrements.

6.2 DMA Operation

The μPD70325 and 70335 each have four DMA transfer modes. Table 6-1 lists the transfer mode functions, etc.

In DMA transfer, internal data area cannot be accessed. When any address corresponding to internal data area is accessed, the external memory location at the same address as the internal data area is accessed.

Table 6-1. Transfer Mode Functions

Mode name	Transfer between	Function	DMA start	Stop method	Interrupt	During HALT	DMA request during DMA operation
Single step	Memory and memory	When one DMA request occurs, executions of one bus cycle and one DMA transfer are repeated alternately as many times as specified.	• On the DMARQ rising edge • When the DMA control register's TDMA bit is set	By software (clear EDMA bit)	All are acknowledged	DMA transfer is executed consecutively as many times as specified.	Channel 1 DMA is pending or stopped and channel 0 DMA is executed.
Burst	Memory and memory	When one DMA request occurs, DMA transfer is executed consecutively as many times as specified.	• On the DMARQ rising edge • When DMA control register TDMA bit is set	None	• Not acknowledged during DMA transfer	DMA transfer is executed consecutively as many times specified.	All other DMA is held pending until DMA transfer terminates.
One transfer	Memory and I/O	Each time a DMA request occurs, one DMA transfer is executed.	• On the DMARQ rising edge	By software (clear EDMA bit)	All are acknowledged	As usual	After one DMA transfer, requested DMA is executed.
Demand release	Memory and I/O	While the DMARQ pin remains high, DMA transfer is executed.	• When DMARQ is set high	• By setting DMARQ low during DMA • By software in other cases (clear EDMA bit)	• Not acknowledged during DMA transfer • All are acknowledged • except during DMA except during DMA transfer	As usual	Channel 1 DMA is pending or stopped and channel 0 DMA is executed.

In DMA transfer between memory and memory, the $\overline{\text{DMAAK}}$ signal is not asserted. In DMA transfer between memory and I/O, the $\overline{\text{DMAAK}}$ signal is asserted for each DMA cycle. Use the $\overline{\text{DMAAK}}$ signal instead of the $\overline{\text{IOSTB}}$ signal to access I/O during DMA transfer.

The programmable wait function (see section **5.1**) is effective even during DMA transfer. During DMA transfer between memory and memory, the specified number of wait states are inserted for each destination and source. During DMA transfer between memory and I/O, one transfer is completed with each bus cycle, thus the specified number of wait states are inserted for memory or I/O, whichever is slower.

The bus hold function and refresh function are also effective during DMA transfer, and DMA transfer is temporarily stopped when either of these functions is executed.

If DMA transfer is requested, it is also executed when a block servicing (transfer, comparison, or retrieval input/output) instruction with a repeat prefix is being executed. At that time, execution of the block servicing instruction is temporarily stopped.

Likewise, DMA transfer is also executed if a BUSLOCK prefix is added.

During DMA transfer, no interrupts are acknowledged and all are held pending.

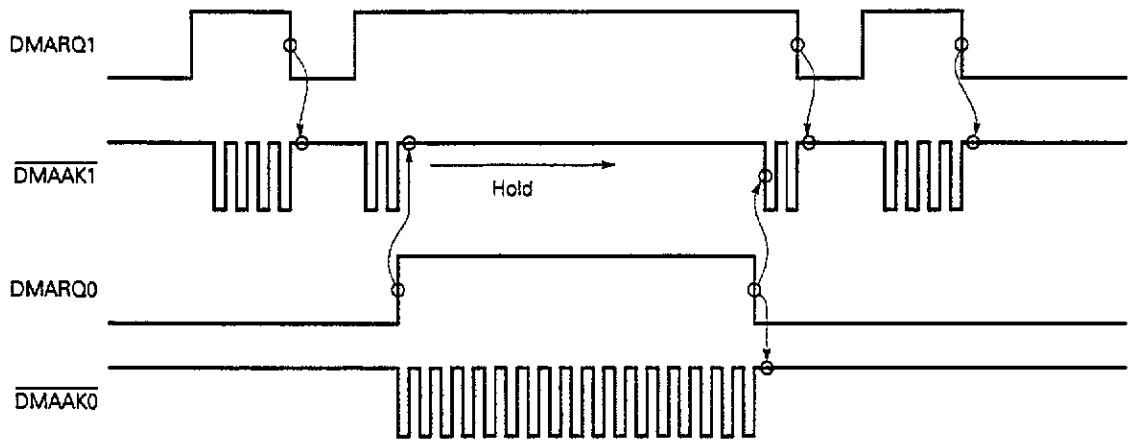
During the HALT mode, if DMA transfer is requested, it is executed. A return is made to the HALT mode when DMA transfer terminates. If a DMA transfer completion interrupt occurs after a return is made to the HALT mode, the HALT mode is released.

If DMA requests occur at the same time, channel 0 takes precedence over channel 1.

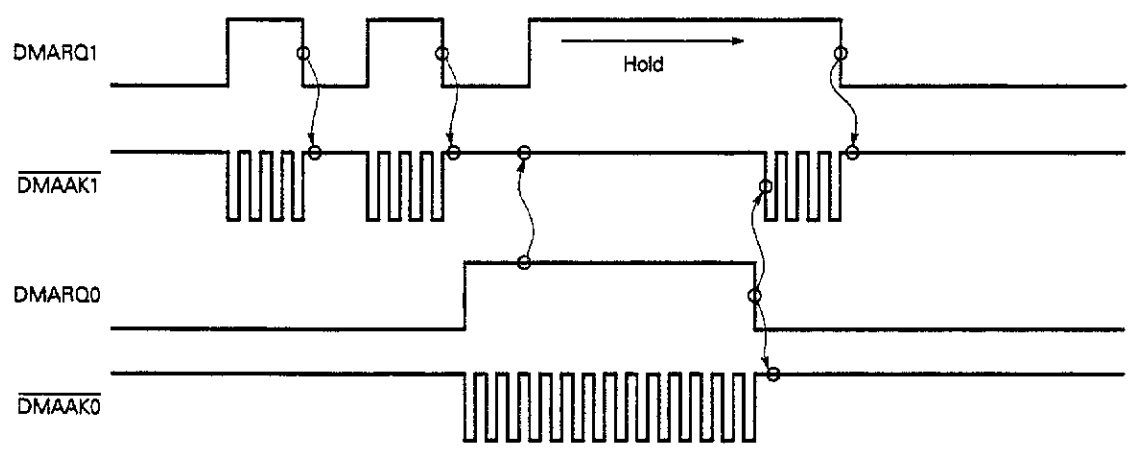
When DMA transfer terminates, the following operations occur.

- Output to the TC pin goes low.
- The EDMA bit is cleared (reset to 0).
- A DMA completion interrupt request occurs.

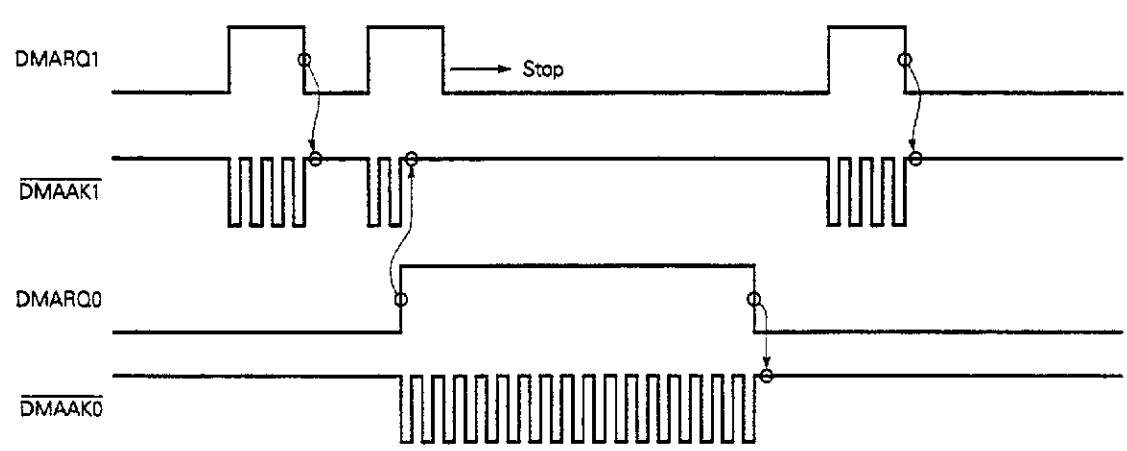
Example 1. When in demand release mode: held pending by DMA request



Example 2. When in demand release mode: held pending by DMA request



Example. When in demand release mode: stopped by DMA request



The DMA transfer rate t of the μPD70325 in the demand release mode is as follows.

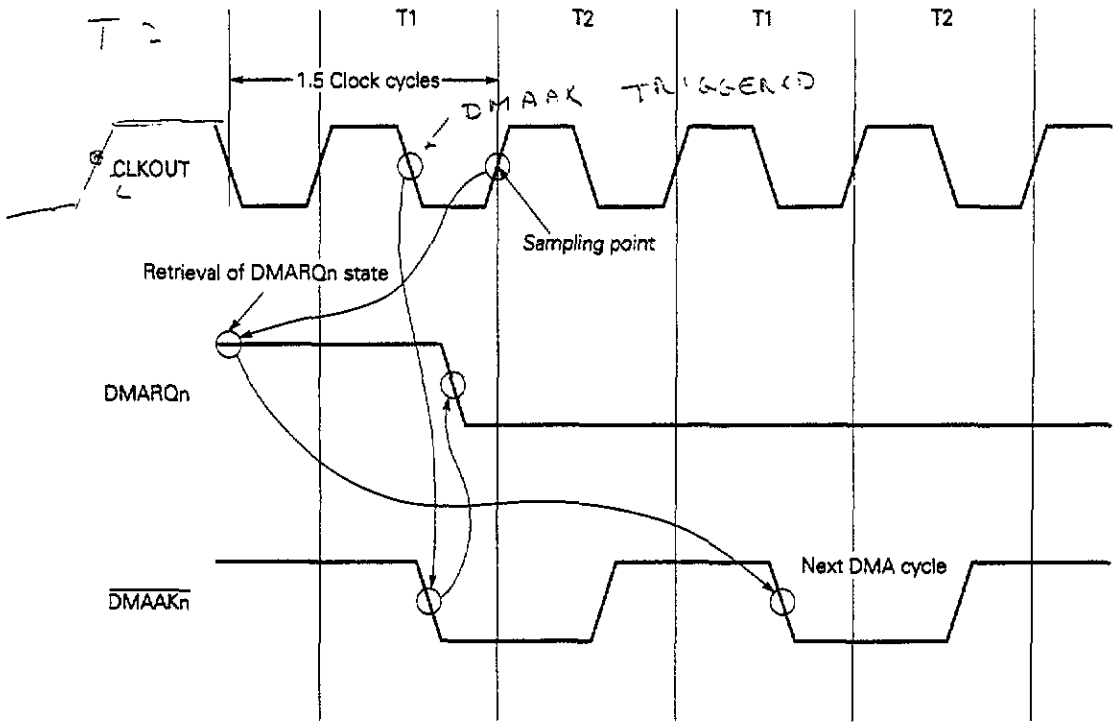
$t = f_{CLK}/(2 + w)$ [byte/s] f_{CLK}: System clock [Hz]
w: Number of wait cycles per one DMA cycle

And, the DMA transfer rate t of the μPD70335 is as follows.

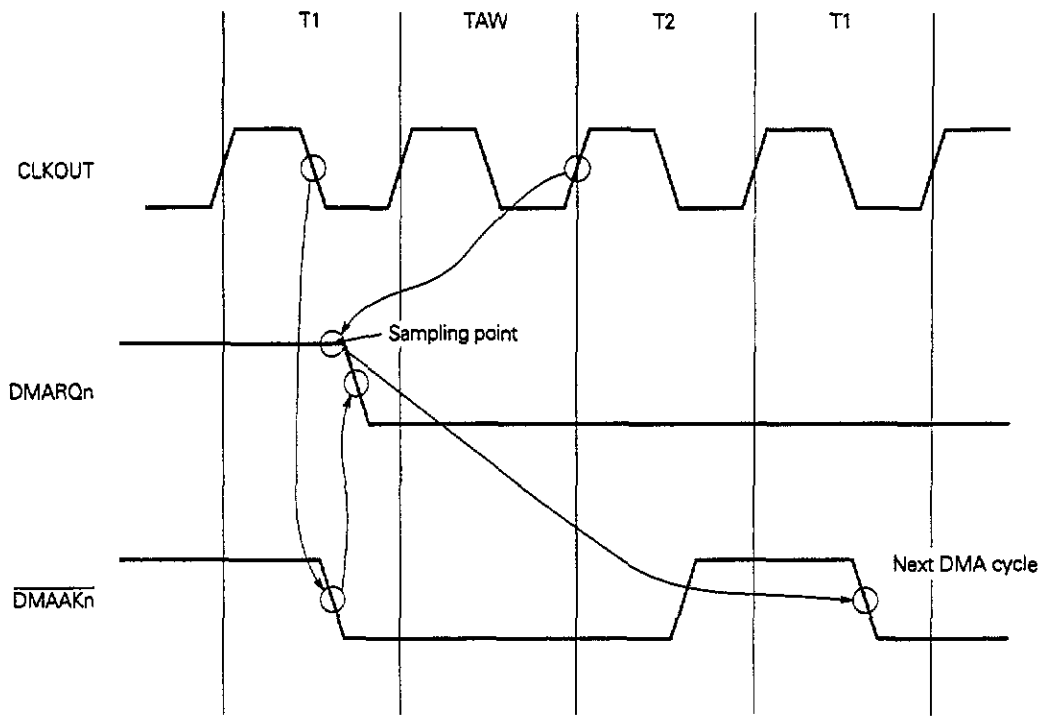
$t = f_{CLK}/(3 + w)$ [byte/s]

The operation timing in this mode is described below.

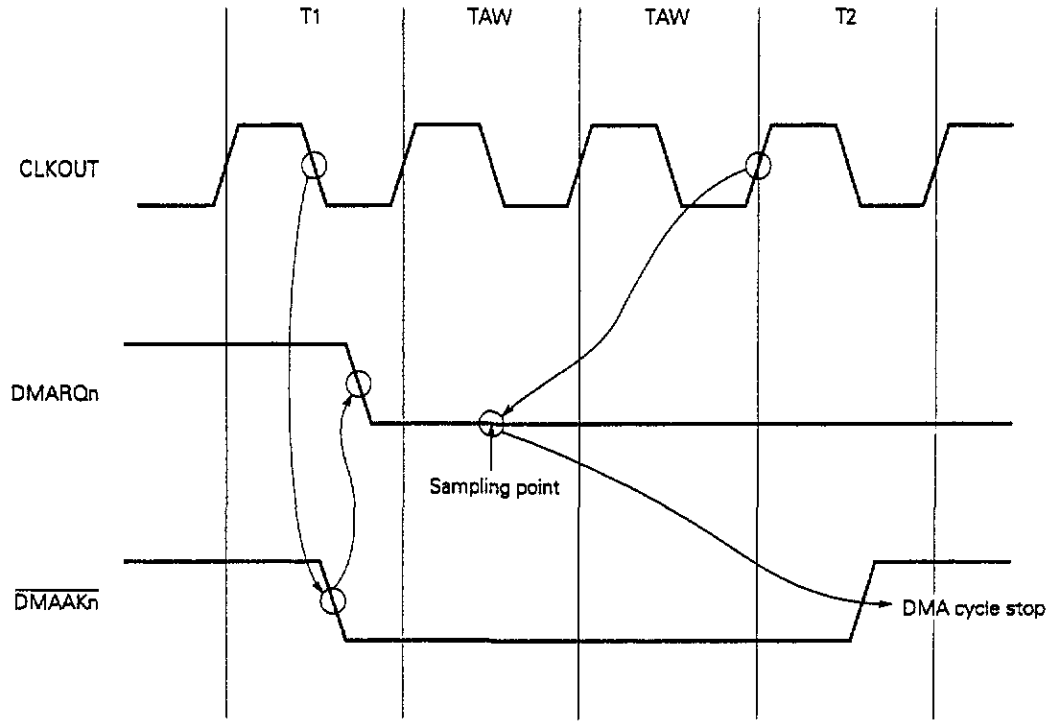
(1) No wait (μPD70325)



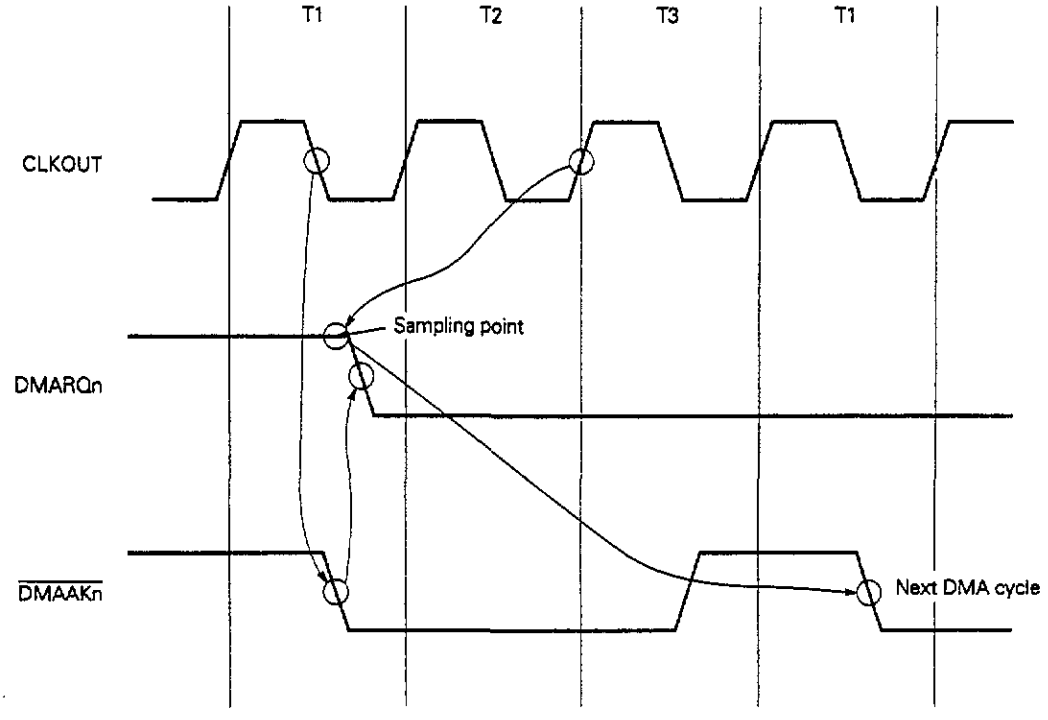
(2) One wait (μPD70325)



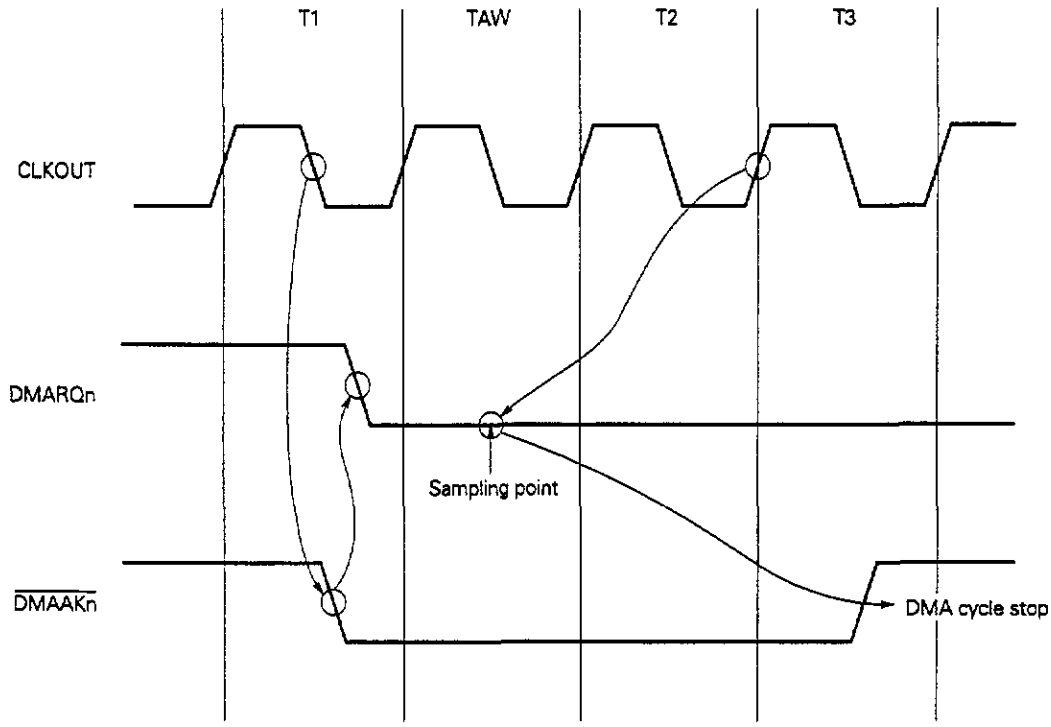
(3) Two wait (μPD70325)



(4) No wait (μPD70335)



(5) Two wait (μPD70335)



The DMA controller samples the DMARQn input each time a bus cycle occurs in the demand release mode, and decides whether to start DMA cycles or to interrupt the DMA cycle according to the resulting state.

As the DMARQn input is a level-sense and the reception of DMA request by the DMARQn is indicated externally by the $\overline{\text{DMAAK}}$n output, the DMARQn input should be held until the falling edge of the $\overline{\text{DMAAK}}$n.

However, with the μPD70325, as the retrieval of DMARQn is executed 1.5 clocks before the T2 cycle, the DMARQn, once raised, cannot be correctly dropped before the next DMA cycle sampling in no-wait or 1-wait. (See (1) and (2).)

Therefore, when using the demand release mode, it is necessary to insert a wait cycle with more than 2 states into a memory or I/O that is to be used for the transfer. (See (3).)

As with the μPD70325, in the μPD70335, as the retrieval of DMARQn is executed 1.5 clocks before the T3 cycle, it is necessary to insert a wait cycle with more than 1 state into a memory or I/O that is to be used for the transfer. (See (4) and (5).)

However, when allowing the TCn register to execute a specified number of DMA transfers by holding the DMARQn once raised until $\overline{\text{TC}}$n output (Terminal count occurs.) without manipulating it during a series of DMA transfer, a higher speed DMA transfer of 1-wait or 0-wait can be set. (Equivalent to the burst mode between memory and I/O)

6.3 DMA Control Registers

DMA mode registers and DMA control registers are provided for DMA transfer mode specification, etc. DMA interrupt request control registers are also provided. These registers are provided for each channel.

6.3.1 DMA mode registers (DMAM0 and DMAM1)

The DMA mode registers are 8-bit registers that specify the DMA transfer mode, etc. The registers can be written/read by making an 8-bit or 1-bit memory access. DMAM0 and DMAM1 correspond to channels 0 and 1, respectively.

When $\overline{\text{RESET}}$ is asserted, the DMAMn register contents are initialized to 00H.

7	6	5	4	3	2	1	0
MD2	MD1	MD0	W	EDMA	TDMA	0	0

MD2, MD1, and MD0 : Transfer mode specification bits

MD2	MD1	MD0	Transfer mode
0	0	0	Single-step mode
0	0	1	Demand release mode (I/O to memory)
0	1	0	Demand release mode (memory to I/O)
0	1	1	Setting prohibited
1	0	0	Burst mode
1	0	1	One transfer mode (I/O to memory)
1	1	0	One transfer mode (memory to I/O)
1	1	1	Setting prohibited

W : Bit specifying whether transfer processing is performed in byte or word units

When this bit is set to 0, byte transfer is specified; when set to 1, word transfer is specified.

Be sure to specify the byte transfer (W = 0) for the μPD70325.

Set even number to both the source and destination initial addresses for the μPD70335 word transfer.

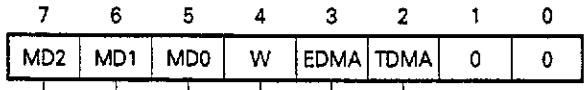
EDMA : DMA transfer enable/disable specification bit

Set this bit to 1 to enable DMA transfer and to 0 to disable DMA transfer. When set to 0 (disable), DMA requests are ignored and are not held pending. When the DMA service channel terminal counter (TC) is set to FFFFH, this bit is automatically cleared (reset to 0).

TDMA : Transfer start bit

This bit is valid only in the single-step or burst mode. DMA transfer is started by writing 1 for this bit, as long as the EDMA bit is set to 1. The bit read level is always 0. This bit has no meaning in the demand release mode or one transfer mode.

Figure 6-1. DMAM0 and DMAM1



7	6	5	4	3	2	1	0	When reset	R/W
MD2	MD1	MD0	W	EDMA	TDMA	0	0	00H	R/W

EDMA	TDMA	Transfer start
0	×	DMA disable state
1	0	State hold
1	1	DMA transfer start

EDMA	DMA transfer enable/disable
0	DMA disable state
1	DMA enable state

W	Transfer processing mode
0	Byte transfer
1	Word transfer

MD2	MD1	MD0	Transfer mode
0	0	0	Single-step mode
0	0	1	Demand release mode (I/O to memory)
0	1	0	Demand release mode (memory to I/O)
0	1	1	Setting prohibited
1	0	0	Burst mode
1	0	1	One transfer mode (I/O to memory)
1	1	0	One transfer mode (memory to I/O)
1	1	1	Setting prohibited

6.3.2 DMA control registers (DMAC0 and DMAC1)

The DMA control registers are 8-bit registers that specify the source address and destination address update modes. The registers can be written/read by making an 8-bit or 1-bit memory access.

The DMACn register contents are maintained even when $\overline{\text{RESET}}$ is asserted.

As shown in Figure 6-2, the source address offset value update mode is specified by setting DMACn register bits 1 and 0 (PS1 and PS0) and the destination address offset value update mode is specified by setting bits 5 and 4 (PD1 and PD0).

Figure 6-2. DMAC0 and DMAC1

7	6	5	4	3	2	1	0	When reset	R/W
0	0	PD1	PD0	0	0	PS1	PS0	Retention[Note]	R/W

PS1/PD1	PS0/PD0	Address offset value update mode
0	1	Increments address offset value
1	0	Decrements address offset value
0	0	No update
1	1	

Note When power-on reset: Undefined

6.3.3 DMA interrupt request control registers (DIC0 and DIC1)

The DMA interrupt request control registers are 8-bit registers that control interrupts which occur upon completion of a DMA transfer. The interrupt occurs when the terminal counter (TC) is set to FFFFH.

The registers can be written/read by making an 8-bit or 1-bit memory access, in which case one wait state is inserted.

When $\overline{\text{RESET}}$ is asserted, the DICn register contents are initialized to 47H.

These interrupts do not support the macro service function. Channel 0 (INTD0) and channel 1 (INTD1) DMA transfer completion interrupts make up one group and channel 0 is specified a higher interrupt priority level than channel 1. INTD0 is controlled by setting the DIC0 register and the vector becomes 20. INTD1 is controlled by setting the DIC1 register and the vector becomes 21. (See section **3.5.5.**)

Interrupt priority levels: DF0 > DF1

Figure 6-3. DIC0 and DIC1

7	6	5	4	3	2	1	0
DF0	DMK0	0	ENCS	0	PR2	PR1	PR0
DF1	DMK1	0	ENCS	0	1	1	1

Caution DIC1 register bits 2 to 0 are fixed to 1.
The DIC1 register interrupt request priority level conforms to the settings of DIC0 register bits PR2 to PR0.

The DF0/DF1 bit is a DMA transfer completion interrupt request flag. The DMK0/DMK1 bit is a DMA transfer completion interrupt mask bit.

See section **4.7** for details of other bit fields.

6.4 DMA Service Channels

In the μPD70325 and 70335, a source address pointer, destination address pointer, and terminal counter are provided in the special function register for each channel to specify the source, destination, and number of DMA transfer. This enables up to 64-KB data to be transferred in the DMA processing.

6.4.1 Source address pointer (SAR0, SAR1)

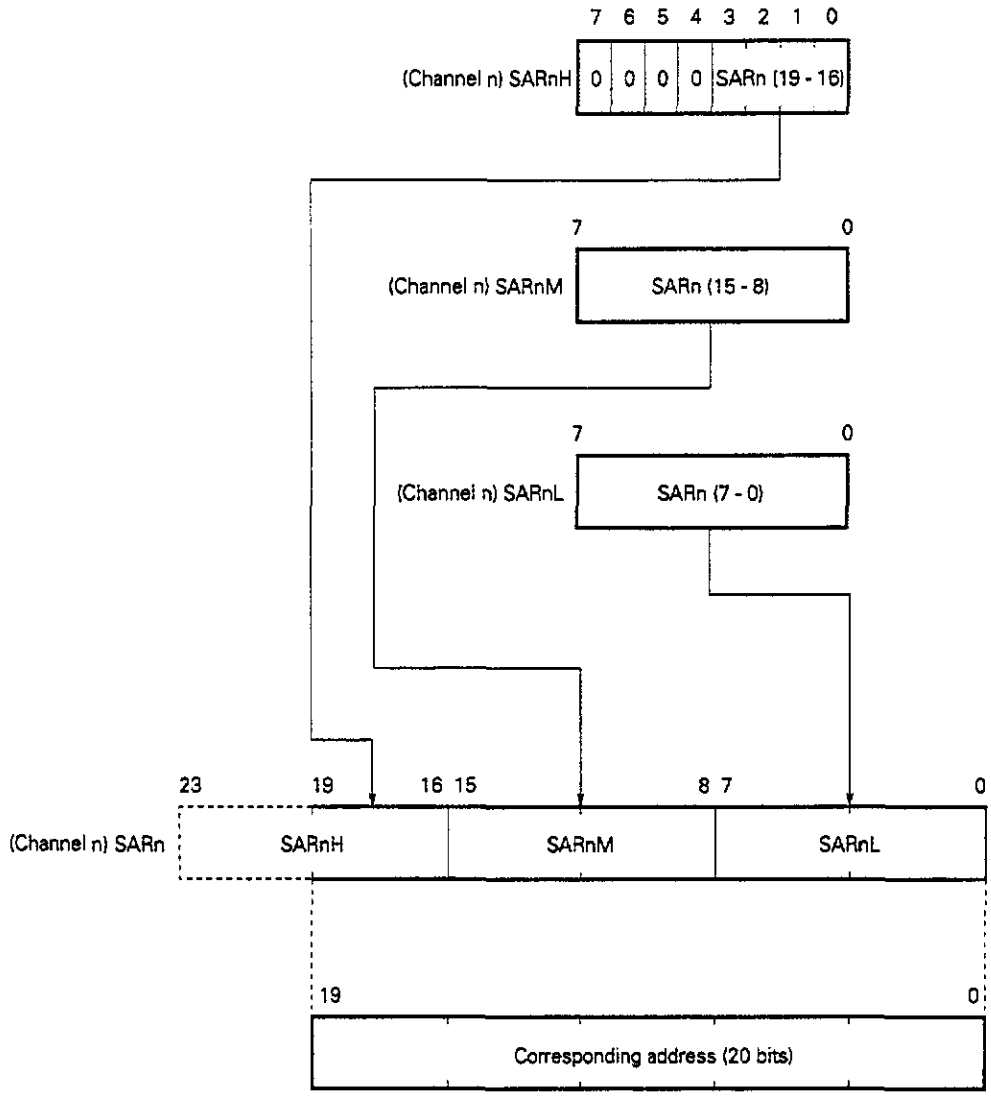
This is a 20-bit pointer that specifies the transfer source address in DMA transfer. It directly specifies the lower 8 bits, higher 8 bits, and highest 4 bits of 20 bits using the SARnL register, SARnM register, and SARnH register (n = 0, 1). The higher 4 bits of the SARnH register (n = 0, 1) are fixed at 0.

The SARnL, SARnM, and SARnH registers (n = 0, 1) execute reading/writing by a memory access (only 8-bit manipulation for SARnH register) of 16/8-bit manipulation.

The pointer value is automatically updated according to the mode specified by the DMA control register (DMACn) when DMA transfer is executed. A byte data is set to ±1 or not changed, and a word data is set to ±2 or not changed (Only byte transfer is possible for the μPD70325.).

It is linearly executed for the updated 20 bits.

Figure 6-4. SAR0 (SAR0H, SAR0M, SAR0L), SAR1 (SAR1H, SAR1M, SAR1L)



Remark n = 0, 1

6.4.2 Destination address pointer (DAR0, DAR1)

This is a 20-bit pointer that specifies a transfer target address in DMA transfer.

- DARnL : bit 7 to bit 0
- DARnM : bit 15 to bit 8
- DARnH : bit 19 to bit 16

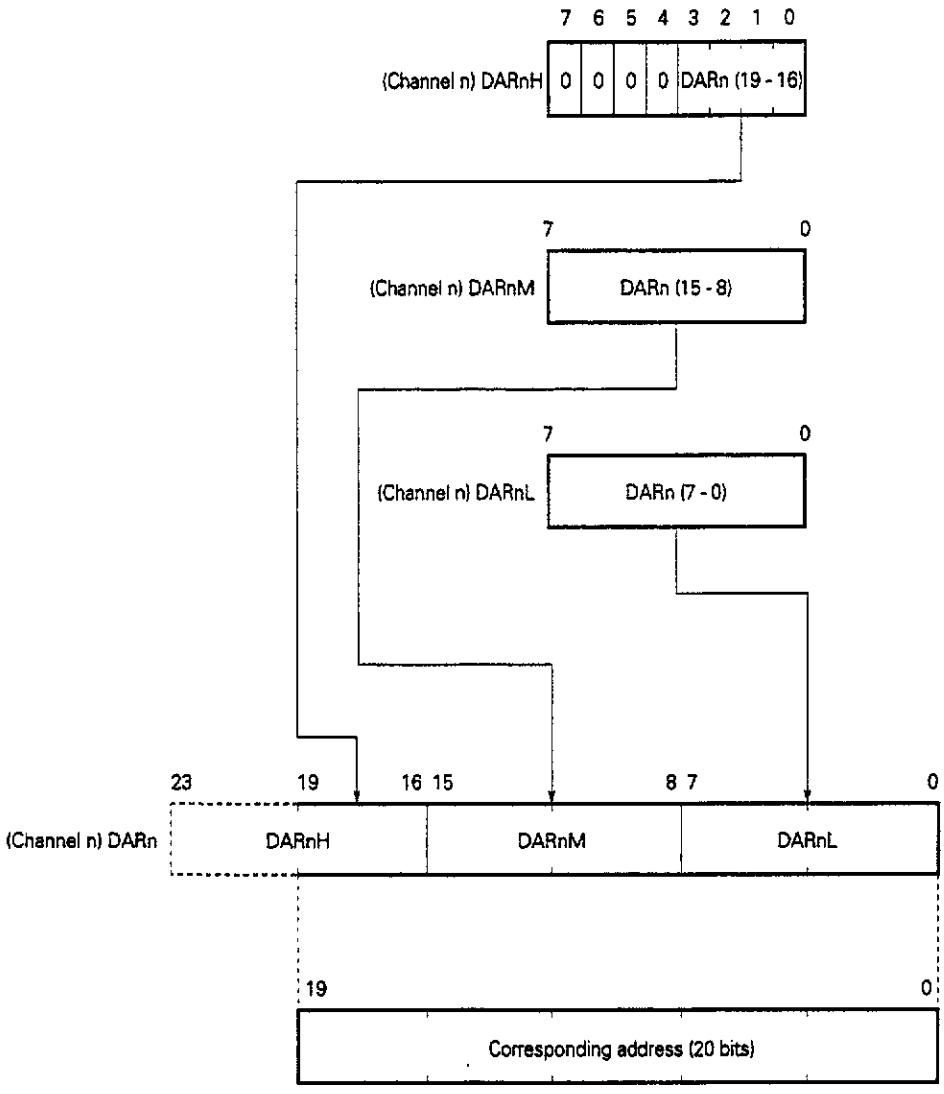
The higher 4 bits of the DARnH register (n = 0, 1) is fixed.

The DARnL, DARnM, and DARnH registers (n = 0, 1) execute reading/writing by a memory access (only 8-bit manipulation for DARnH register) of 16/8-bit manipulation.

The pointer value is automatically updated according to the mode specified by the DMA control register (DMACn) when DMA transfer is executed. A byte data is set to ±1 or not changed, and a word data is set to ±2 or not changed (Only byte transfer is possible for the μPD70325.).

It is linearly executed for the updated 20 bits.

Figure 6-5. DAR0 (DAR0H, DAR0M, DAR0L), DAR1 (DAR1H, DAR1M, DAR1L)



Remark n = 0, 1

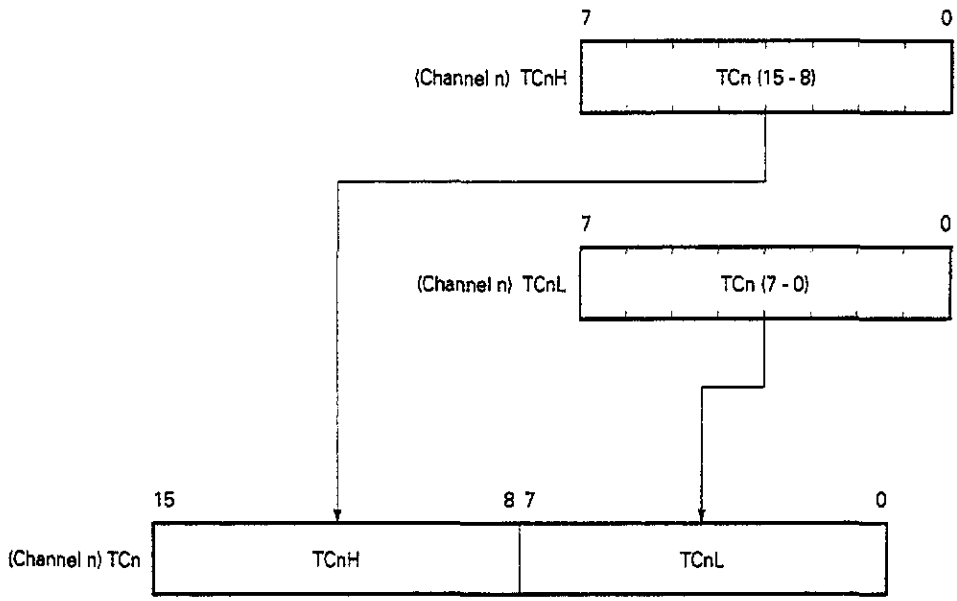
6.4.3 Terminal counter (TC0, TC1)

This is a 16-bit counter that specifies the transfer number in DMA transfer. The lower 8 bits and higher 8 bits are specified by the TCnL and TCnH registers (n = 0, 1). Specify transfer number – 1 to it.

The TCnL and TCnH registers (n = 0, 1) can execute reading/writing by a memory access of 16/8-bit manipulation.

The counter value is automatically updated by the DMA operation, and it is decremented by 1 each time a DMA transfer is executed.

Figure 6-6. TC0 (TC0H, TC0L), TC1 (TC1H, TC1L)



Remark n = 0, 1

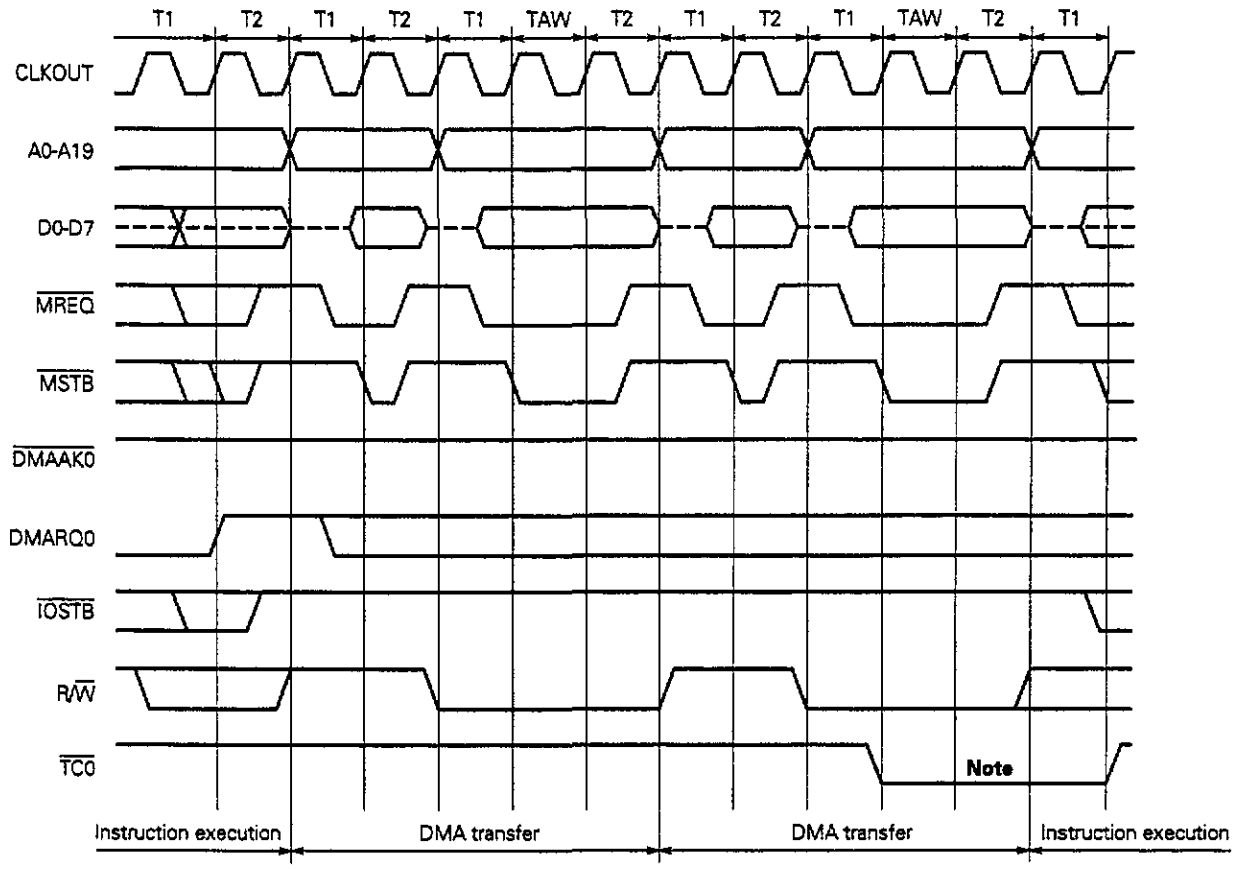
6.5 DMA Transfer Timings

Figures 6-7 to 6-15 show the main DMA transfer timings.

6.5.1 DMA transfer timing of μPD70325

Figure 6-6 shows the burst mode timing with no wait state insertion for the source memory bank and 1-wait state insertion for the destination when DMA is started by the DMARQ signal with TC =1.

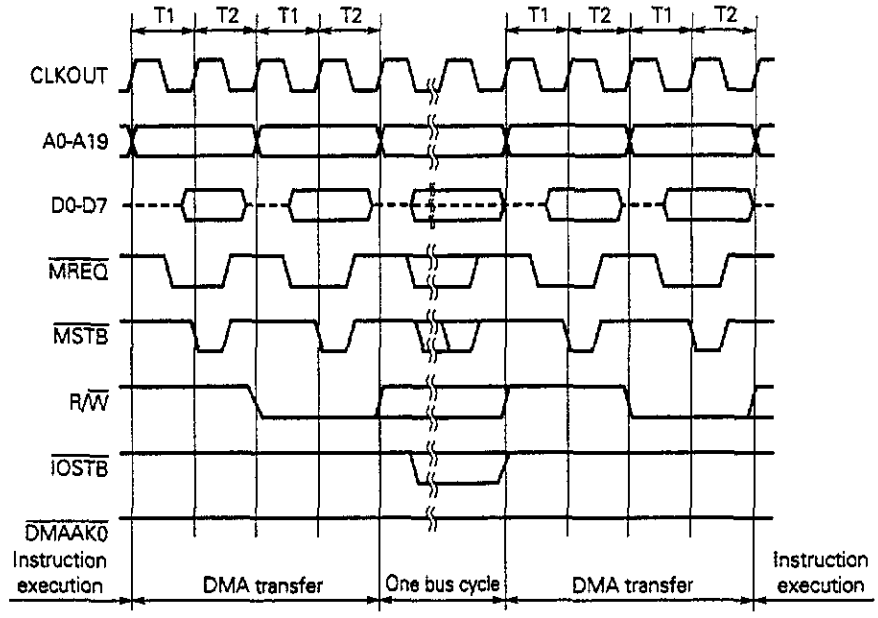
Figure 6-7. When in Burst Mode



Note The active period of $\overline{TC0}$ is 2 + W clock cycles (W: number of wait states).

Remark The broken lines indicate high impedance.

Figure 6-8. When in Single-Step Mode (μPD70325)



Remark *The broken lines indicate high impedance.*

Figure 6-9. One Transfer Mode on μPD70325 (memory to I/O with no wait state insertion)

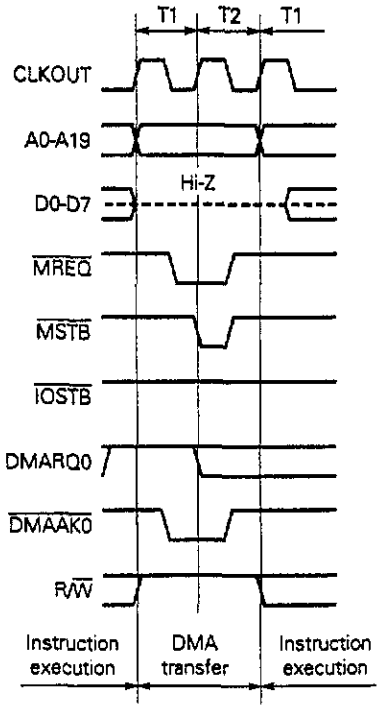
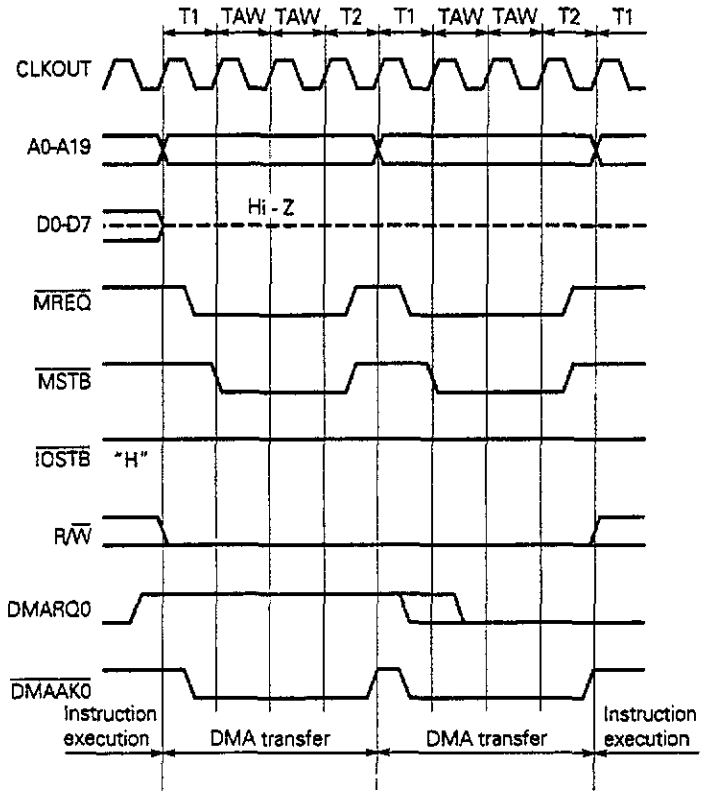


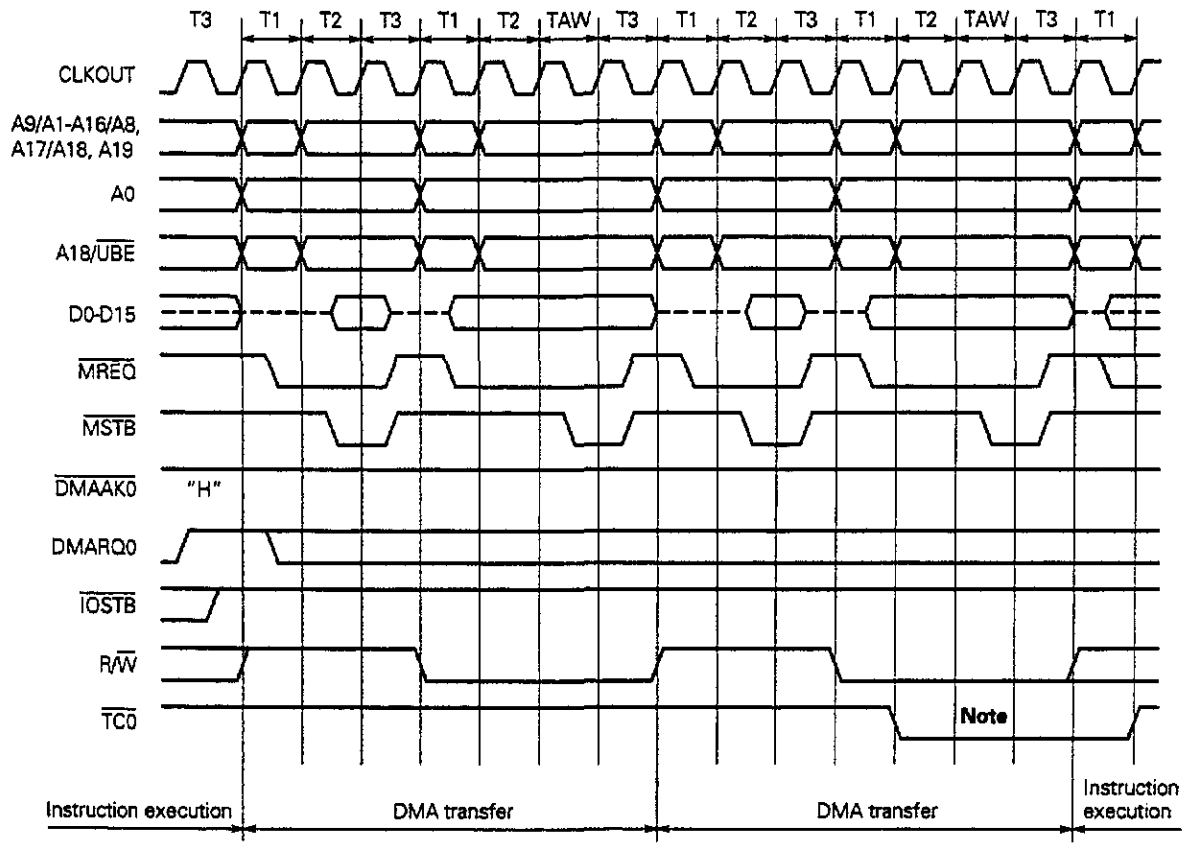
Figure 6-10. Demand Release Mode (I/O to memory, I/O: two wait state insertion, memory: no wait state insertion)



6.5.2 DMA transfer timing of μPD70335

Figure 6-11 shows the burst mode (byte transfer) timing with no wait state insertion for the source memory bank and one wait state insertion for the destination when DMA is started by the DMARQ signal with TC = 1.

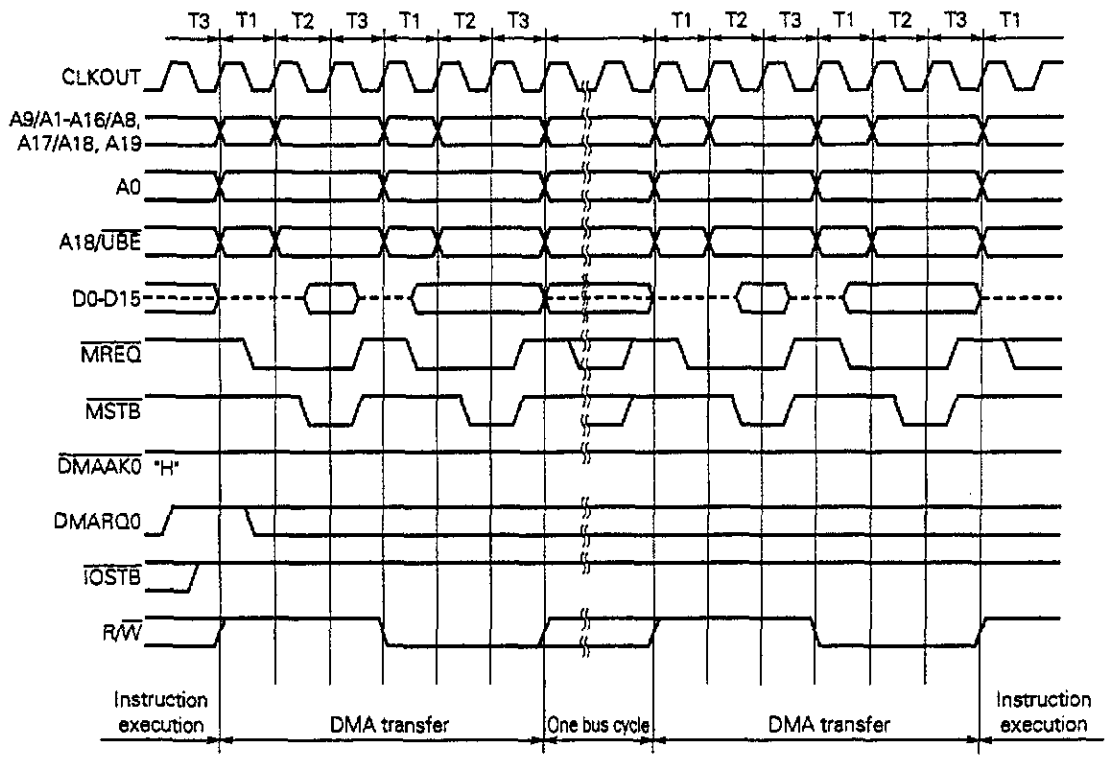
Figure 6-11. When in Burst Mode (byte transfer)



Note The active period of $\overline{TC0}$ is 3 + W clock cycles.

Remark The broken lines indicate high impedance.

Figure 6-12. When in Single-Step Mode (μPD70335)



Remark The broken lines indicate high impedance.

Figure 6-13. One Transfer Mode on μPD70335 (memory to I/O with no wait state insertion)

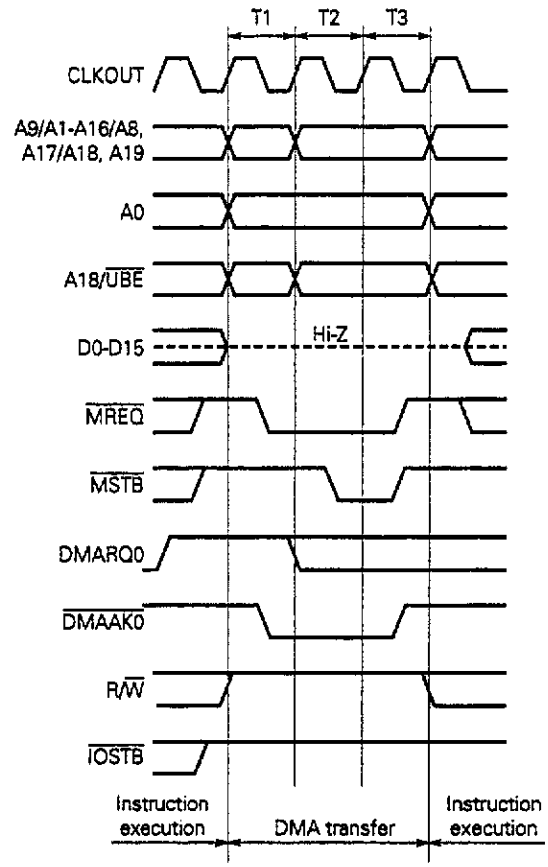
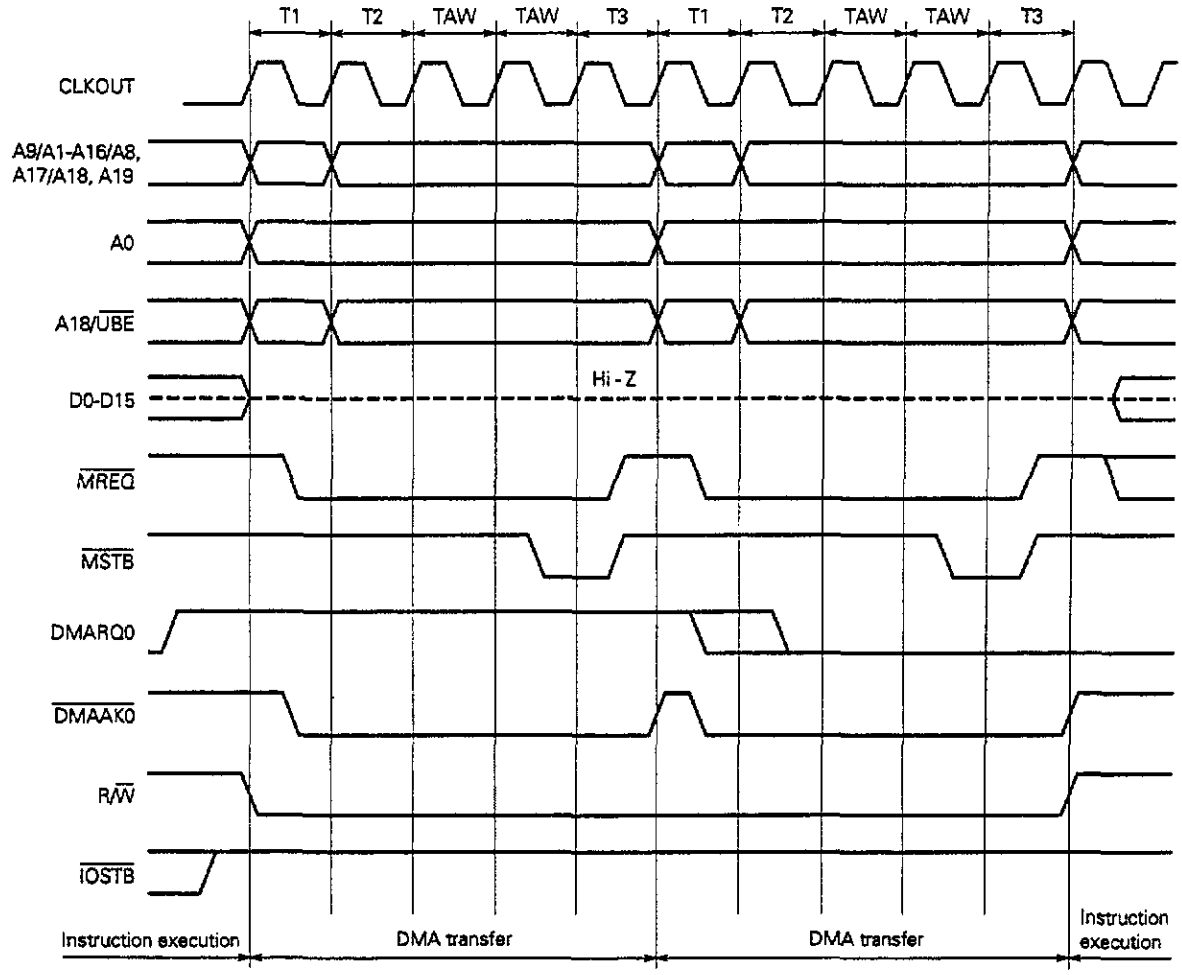


Figure 6-14. Demand Release Mode (I/O to memory, I/O: two wait state insertion, memory: no wait state insertion)



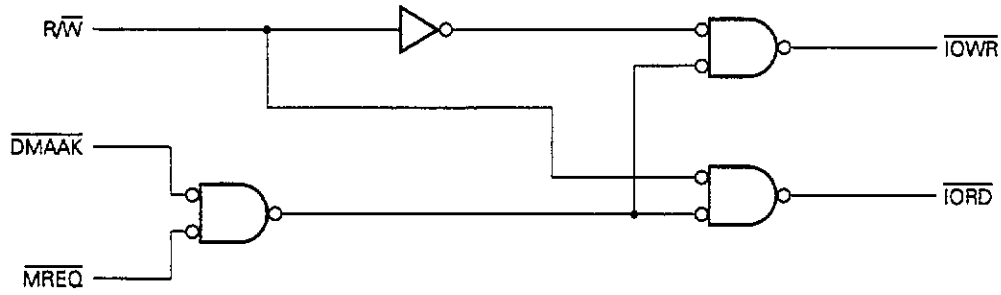
In DMA transfer between I/O and memory, the $\overline{IOSTB}$ signal is not output ($\overline{IOSTB}$ = high) and the $\overline{DMAAK}$ signal is output ($\overline{DMAAK}$ = low).

Therefore, the $\overline{DMAAK}$ signal is used for I/O access decision. In DMA transfer between memory and memory, the $\overline{DMAAK}$ signal is not output.

The delay time of $\overline{MREQ}$ ↑ and $\overline{DMAAK}$ ↑ during memory to I/O cannot be specified. Design the circuit by considering a case where $\overline{DMAAK}$ ↑ is delayed for $\overline{MREQ}$ ↑.

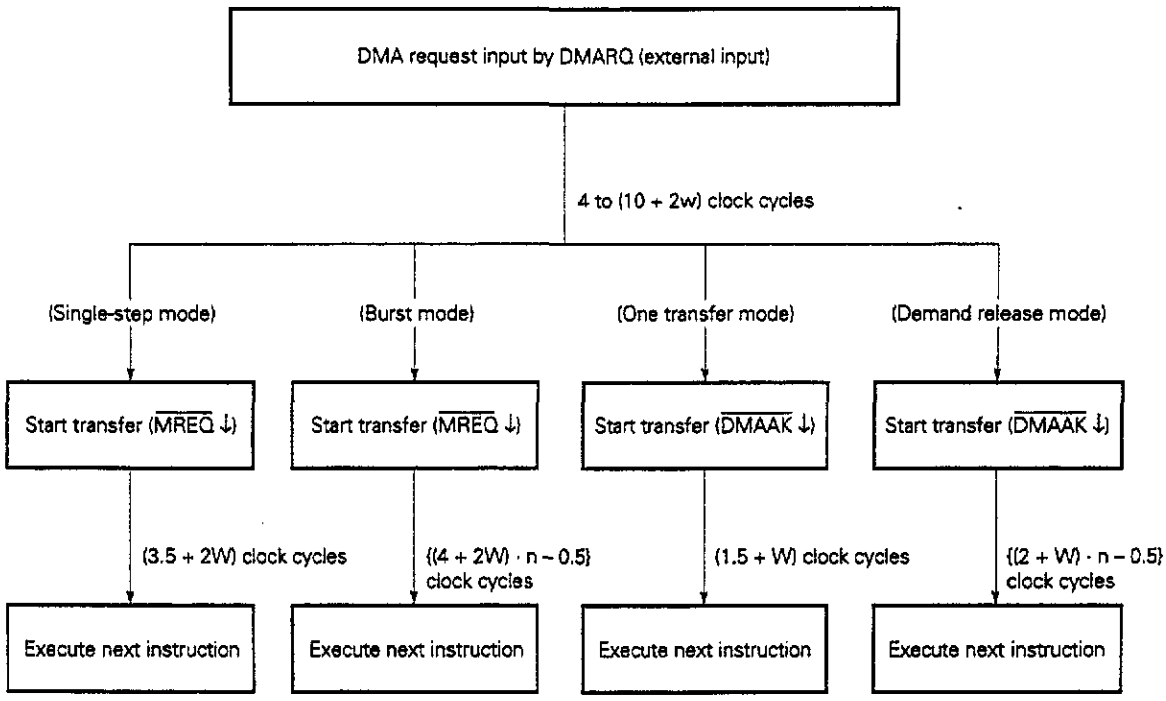
Figure 6-15 shows a circuit example in which the $\overline{IORD}$ and $\overline{IOWR}$ signals are generated from the R/$\overline{W}$, $\overline{IOSTB}$, $\overline{DMAAK}$ and $\overline{MREQ}$ signals.

Figure 6-15. Example of $\overline{IORD}$ and $\overline{IOWR}$ Signal Generation Circuit



6.6 DMA Execution Time

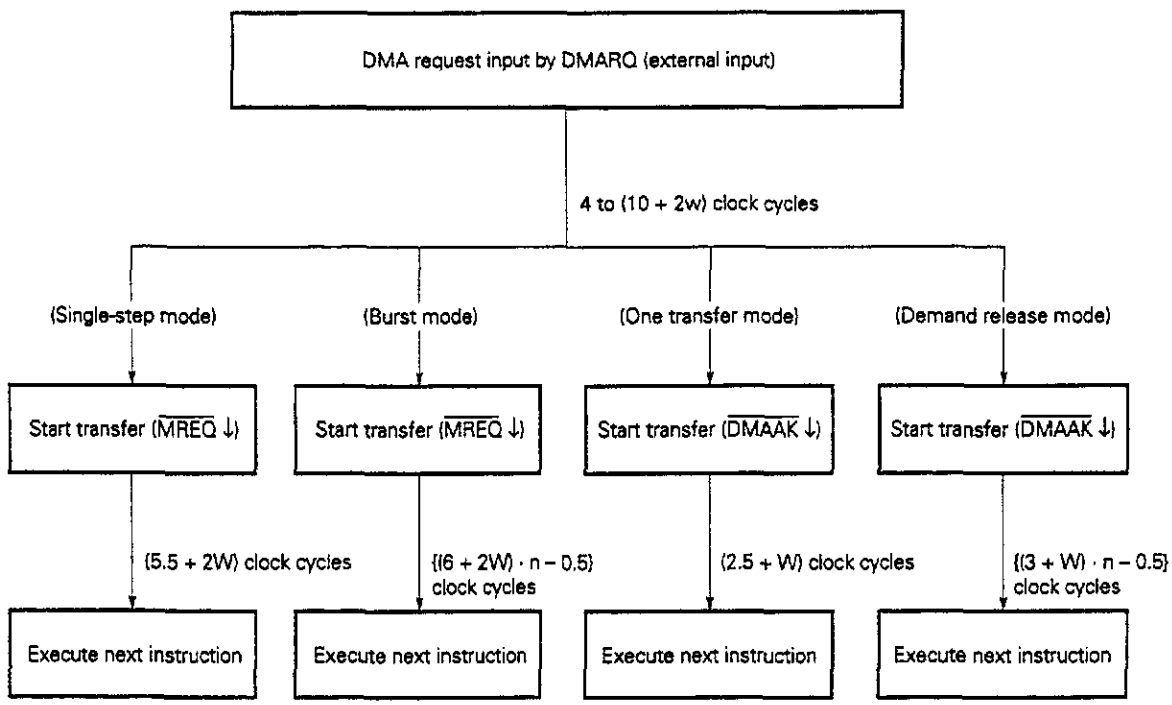
6.6.1 V25+'s DMA execution time (number of system clock cycles)



w : Number of bus cycles waited at that time
W: Number of wait cycles per DMA transfer bus cycle
n : DMA transfer count

Cautions Refresh cycles, hold requests, interrupt requests, and other DMA requests are not considered.

6.6.2 V35+'s DMA execution time (number of system clock cycles)



w : Number of bus cycles waited at that time
W: Number of wait cycles per DMA transfer bus cycle
n : DMA transfer count

Cautions Refresh cycles, hold requests, interrupt requests, and other DMA requests are not considered.