HD6844, HD68A44, HD68B44*

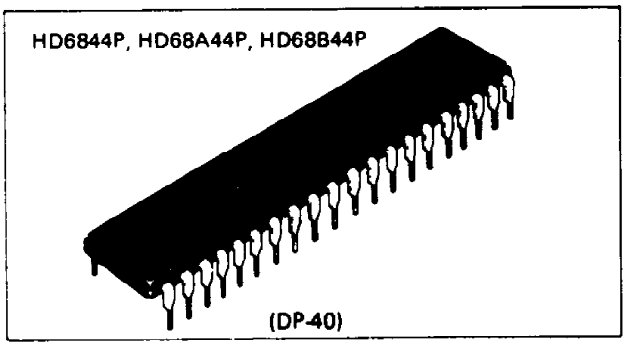
DMAC (Direct Memory Access Controller)

The HD6844 Direct Memory Access Controller (DMAC) performs the function of transferring data directly between memory and peripheral device controllers. It controls the address and data buses in place of the MPU in bus organized systems such as the HMCS6800 Microprocessor System.

The bus interface of the HD6844 includes select, read/write, interrupt, transfer request/grant, and bus interface logic to allow the data transfer over an 8-bit bidirectional data bus. The functional configuration of the DMAC is programmed via the data bus. The internal structure provides for control and handling of four individual channels, each of which is separately configured. Programmable control registers provide control for the transfer location and length, individual channel control and transfer mode configuration, priority of servicing, data chaining, and interrupt control. Status and control lines provide control to the peripheral controllers.

The mode of transfer for each channel can be programmed as cycle-stealing or a burst transfer mode.

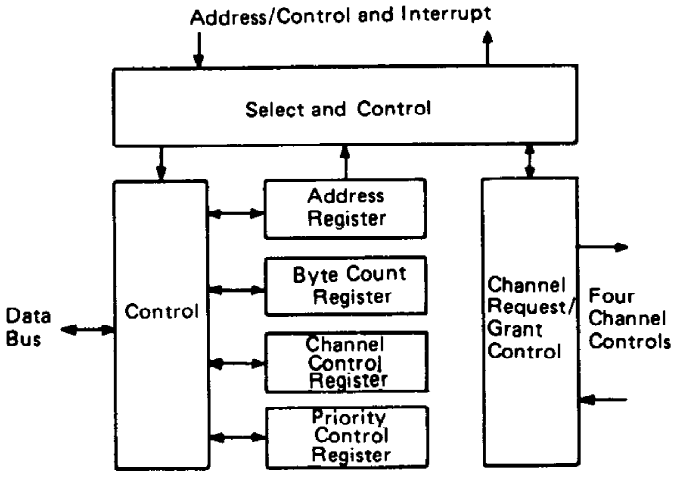
Typical applications would be with the Floppy Disk Controller (FDC), etc..



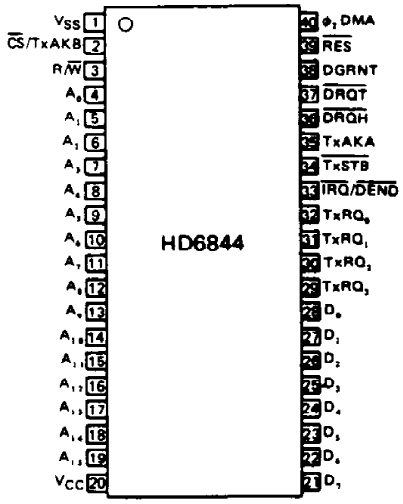
■ FEATURES

- Four DMA Channels, Each Having a 16-Bit Address Register and a 16-Bit Byte Count Register
- 1 M Byte/Sec (HD6844), 1.5 M Byte/Sec (HD68A44), 2.0 M Byte/Sec (HD68B44) Maximum Data Transfer Rate
- Selection of Fixed or Rotating Priority Service Control
- Separate Control Bits for Each Channel
- Data Chain Function
- Address Increment or Decrement Update
- Programmable Interrupts and DMA End to Peripheral Controllers
- Compatible with MC6844, MC68A44, MC68B44

■ BLOCK DIAGRAM



■ PIN ARRANGEMENT



(Top View)

***Old type name: HD46504, HD46504-1, HD46504-2.**

5

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC}*	-0.3 ~ +7.0	V
Input Voltage	V_{in}*	-0.3 ~ +7.0	V
Operating Temperature	T_{opr}	-20 ~ +75	°C
Storage Temperature	T_{stg}	-55 ~ +150	°C

* With respect to V_{SS} (SYSTEM GND)

(NOTE) Permanent LSI damage may occur if maximum ratings are exceeded. Normal operation should be under recommended operating conditions. If these conditions are exceeded, it could affect reliability of LSI.

■ RECOMMENDED OPERATING CONDITIONS

Item	Symbol	min	typ	max	Unit
Power Supply Voltage	V_{CC}*	4.75	5.0	5.25	V
Input Voltage	V_{IL}*	-0.3	–	0.8	V
	V_{IH}*	2.0	–	V_{CC}	V
Operating Temperature	T_{opr}	-20	25	75	°C

* With respect to V_{SS} (SYSTEM GND)

■ ELECTRICAL CHARACTERISTICS (V_{CC}=5V±5%, V_{SS}=0V, Ta=-20~+75°C, unless otherwise noted.)

• DC CHARACTERISTICS

Item		Symbol	Test Condition	min	typ*	max	Unit
Input "High" Voltage		V_{IH}		2.0	–	V_{CC}	V
Input "Low" Voltage		V_{IL}		-0.3	–	0.8	V
Input Leakage Current	$TxRQ_{0\sim3}$, ϕ_2DMA, $\overline{RES}$, DGRNT	I_{in}	V_{in}=0~5.25V	-2.5	–	2.5	μA
Three-State (off state) Leakage Current	A_0~A_{15}, D_0~D_7, R/$\overline{W}$	I_{TSI}	V_{in}=0.4~2.4V	-10	–	10	μA
Output "High" Voltage	D_0~D_7	V_{OH}	I_{OH}=-205μA	2.4	–	–	V
	A_0~A_{15}, R/$\overline{W}$		I_{OH}=-145μA	2.4	–	–	
	All Other Outputs		I_{OH}=-100μA	2.4	–	–	
Output "Low" Voltage		V_{OL}	I_{OL}=1.6mA	–	–	0.4	V
Source Current	$\overline{CS}$/TxAKB	I_{CSS}	V_{in}=0V, Fig. 10	–	10	16	mA
Power Dissipation		P_D		–	500	1000	mW
Input Capacitance	ϕ_2DMA	C_{in}	V_{in}=0V, T_a=25°C f=1.0MHz	–	–	20	pF
	D_0~D_7, $\overline{CS}$, A_0~A_4, R/$\overline{W}$			–	–	12.5	
	$TxRQ_{0\sim3}$, $\overline{RES}$, DGRNT			–	–	10	
Output Capacitance		C_{out}	V_{in}=0V, T_a=25°C, f=1MHz	–	–	12	pF

* V_{CC}=5.0V, T_a=25°C

• AC CHARACTERISTICS (Load Condition Fig. 9)

1. CLOCK TIMING

Item		Symbol	Test Condition	HD6844			HD68A44			HD68B44			Unit
				min	typ	max	min	typ	max	min	typ	max	
ϕ_2 DMA Cycle Time		$t_{cyc\phi}$	Fig. 2	1000	–	–	666	–	–	500	–	–	ns
ϕ_2 DMA Pulse Width	"High" Level	$PW_{\phi H}$	Fig. 2	450	–	–	280	–	–	235	–	–	ns
	"Low" Level	$PW_{\phi L}$	Fig. 2	400	–	–	230	–	–	210	–	–	ns
ϕ_2 DMA Rise and Fall Time		$t_{\phi r}$, $t_{\phi f}$	Fig. 2	–	–	25	–	–	25	–	–	25	ns

2. DMA TIMING (Load Condition Fig. 9)

Item		Symbol	Test Condition	HD6844			HD68A44			HD68B44			Unit
				min	typ	max	min	typ	max	min	typ	max	
TxRQ Setup Time	ϕ_2 DMA Rising Edge	t_{TQS1}	Fig. 3	120	–	–	120	–	–	120	–	–	ns
	ϕ_2 DMA Falling Edge	t_{TQS2}		210	–	–	210	–	–	155	–	–	
TxRQ Hold Time	ϕ_2 DMA Rising Edge	t_{TQH1}		20	–	–	10	–	–	10	–	–	ns
	ϕ_2 DMA Falling Edge	t_{TQH2}		20	–	–	10	–	–	10	–	–	
DGRNT Setup Time	DGRNT	t_{DGS}	Fig. 4	155	–	–	125	–	–	115	–	–	ns
DGRNT Hold Time	DGRNT	t_{DGH}		10	–	–	10	–	–	10	–	–	
Address Output Delay Time	A_0~A_{15}, R/$\overline{W}$, $\overline{TxSTB}$	t_{AD}	Fig. 6	–	–	270	–	–	180	–	–	160	ns
Address Output Hold Time	A_0~A_{15}, R/$\overline{W}$	t_{AHO}	Fig. 6	30	–	–	20	–	–	20	–	–	ns
	$\overline{TxSTB}$		Fig. 7	35	–	–	35	–	–	35	–	–	
Address Three-State Delay Time	A_0~A_{15}, R/$\overline{W}$	t_{ATSD}	Fig. 7	–	–	270	–	–	270	–	–	270	ns
Address Three-State Recovery Time	A_0~A_{15}, R/$\overline{W}$	t_{ATSR}	Fig. 7	–	–	270	–	–	270	–	–	270	ns
Delay Time	$\overline{DRQH}$, $\overline{DRQT}$	t_{DQD}	Fig. 5	–	–	375	–	–	250	–	–	210	ns
TxAK Delay Time	ϕ_2 DMA Rising Edge	t_{TKD1}	Fig. 5	–	–	400	–	–	310	–	–	250	ns
	DGRNT Rising Edge	t_{TKD2}	Fig. 8	–	–	190	–	–	160	–	–	150	
$\overline{IRQ}$/$\overline{DEND}$ Delay Time	ϕ_2 DMA Falling Edge	t_{DED1}	Fig. 6	–	–	300	–	–	250	–	–	210	ns
	DGRNT Rising Edge	t_{DED2}	Fig. 8	–	–	190	–	–	160	–	–	125	

3. BUS TIMING (Load Condition Fig. 9)

1) READ TIMING

Item		Symbol	Test Condition	HD6844			HD68A44			HD68B44			Unit
				min	typ	max	min	typ	max	min	typ	max	
Address Setup Time	A_0~A_4, R/$\overline{W}$, $\overline{CS}$	t_{AS}	Fig. 2	140	–	–	140	–	–	70	–	–	ns
Address Input Hold Time	A_0~A_4, R/$\overline{W}$, $\overline{CS}$	t_{AHI}		10	–	–	10	–	–	10	–	–	ns
Data Delay Time	D_7~D_7	t_{DDR}		–	–	320	–	–	220	–	–	180	ns
Data Access Time	D_0~D_7	t_{ACC}		–	–	460	–	–	360	–	–	280	ns
Data Output Hold Time	D_0~D_7	t_{DHR}		10	–	–	10	–	–	10	–	–	ns

5

2) WRITE TIMING

Item		Symbol	Test Condition	HD6844			HD68A44			HD68B44			Unit
				min	typ	max	min	typ	max	min	typ	max	
Address Setup Time	A_0~A_4, R/$\overline{W}$, $\overline{CS}$	t_{AS}	Fig. 2	140	–	–	140	–	–	70	–	–	ns
Address Input Hold Time	A_0~A_4, R/$\overline{W}$, $\overline{CS}$	t_{AHI}		10	–	–	10	–	–	10	–	–	ns
Data Setup Time	D_0~D_7	t_{DSW}		195	–	–	80	–	–	60	–	–	ns
Data Input Hold Time	D_0~D_7	t_{DHW}		10	–	–	10	–	–	10	–	–	ns

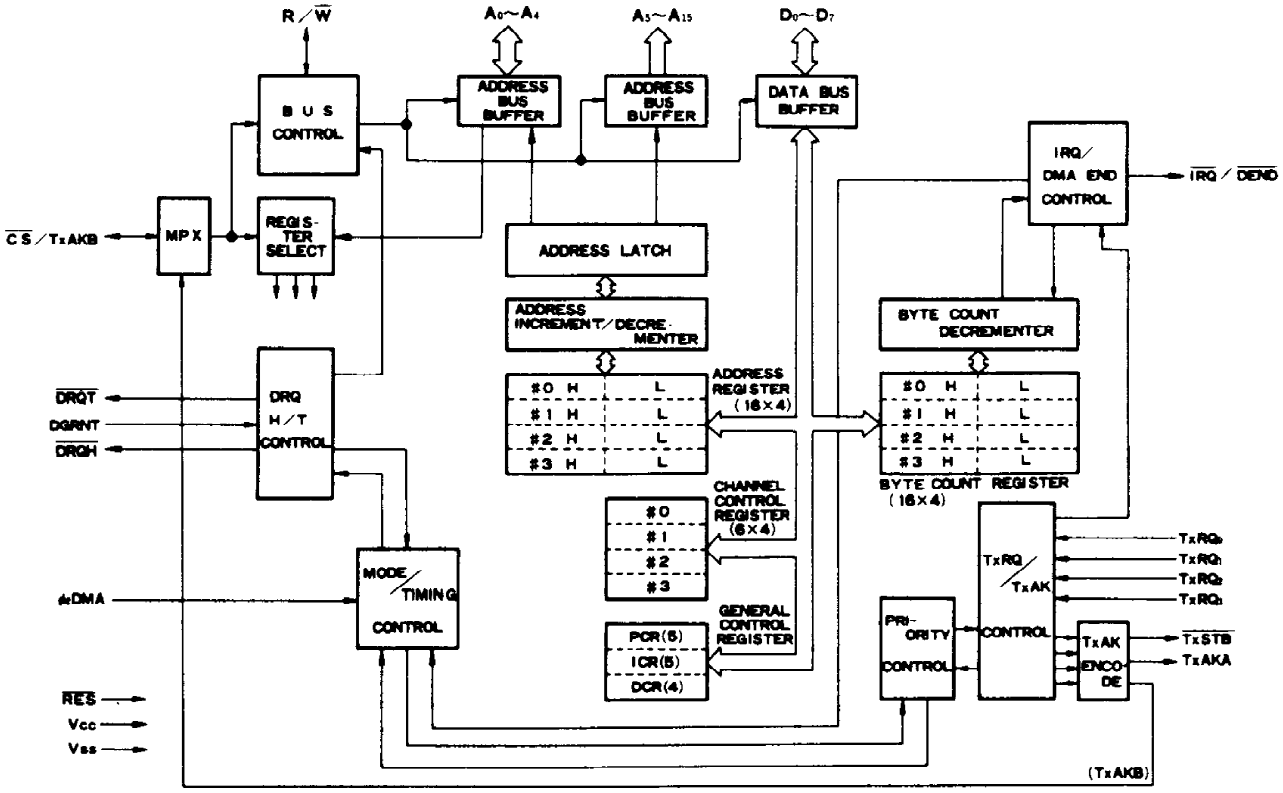


Figure 1 Expanded Block Diagram

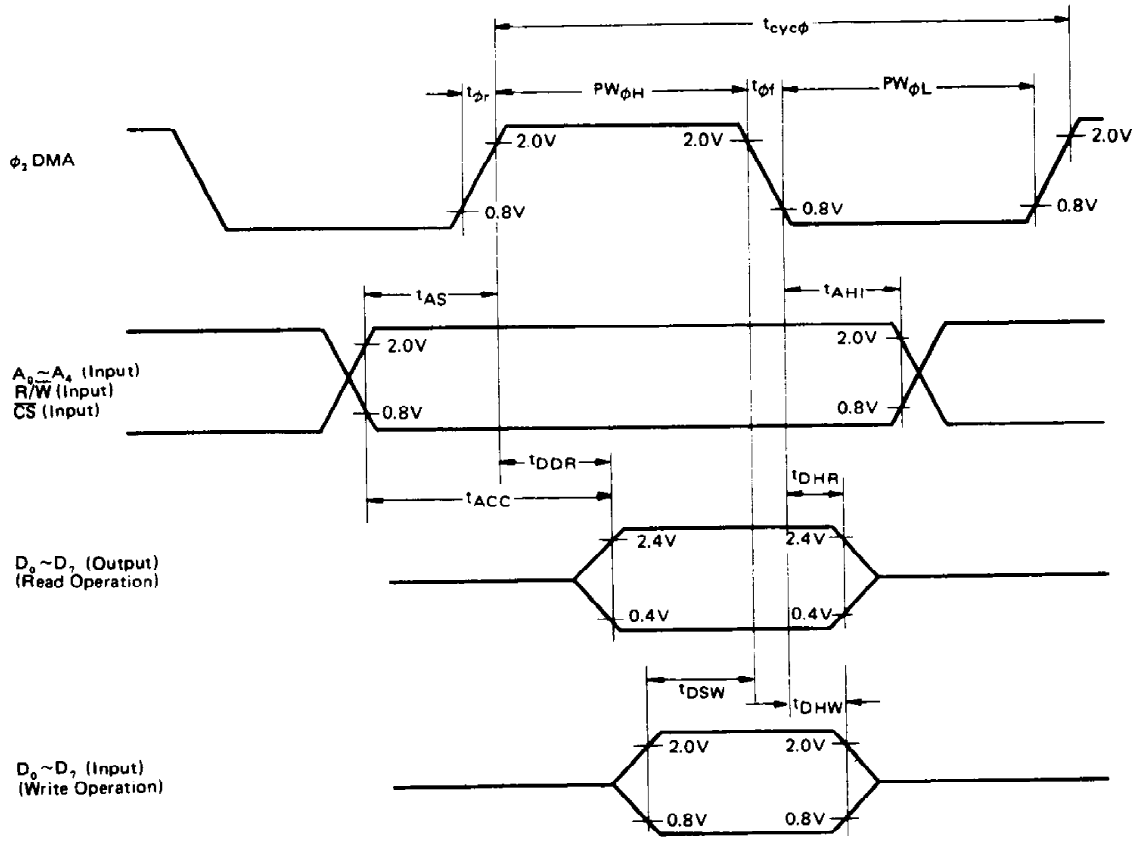


Figure 2 Read/Write Sequence

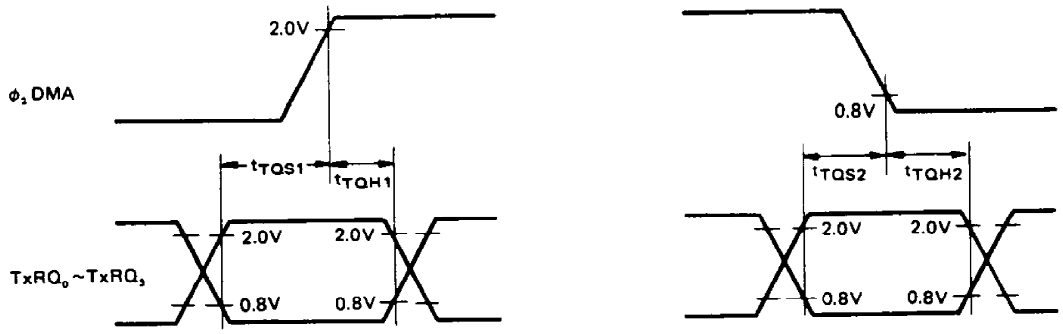


Figure 3 Timing of TxRQ Input

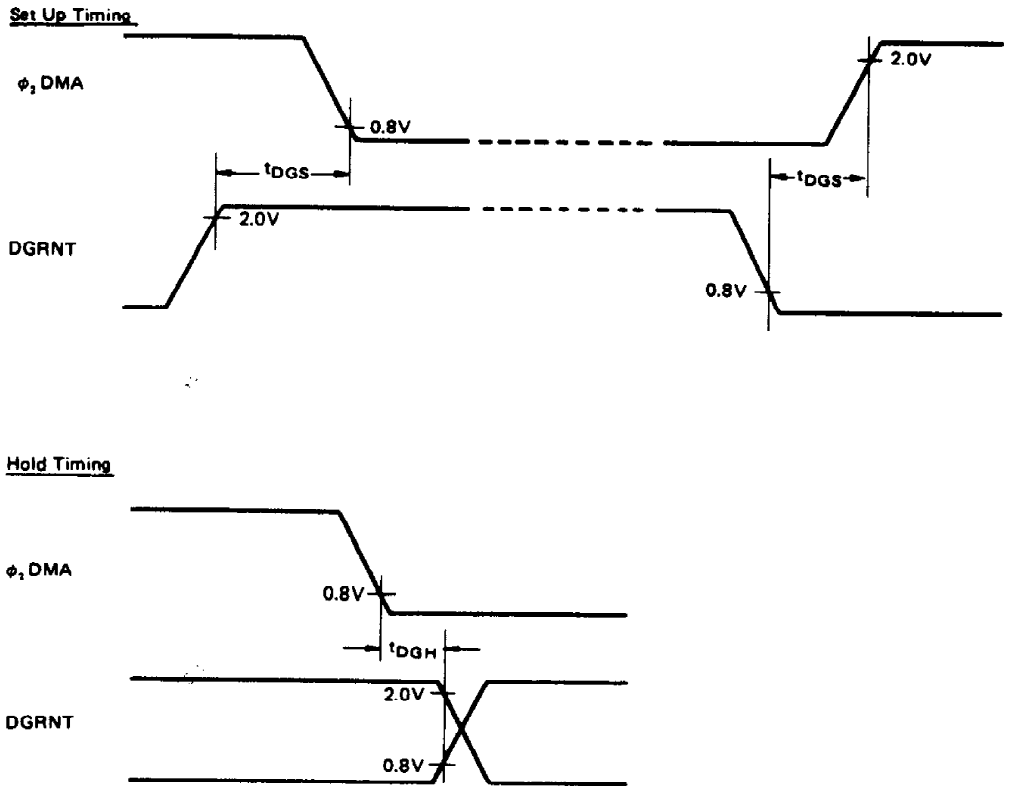


Figure 4 Timing of DGRNT Input

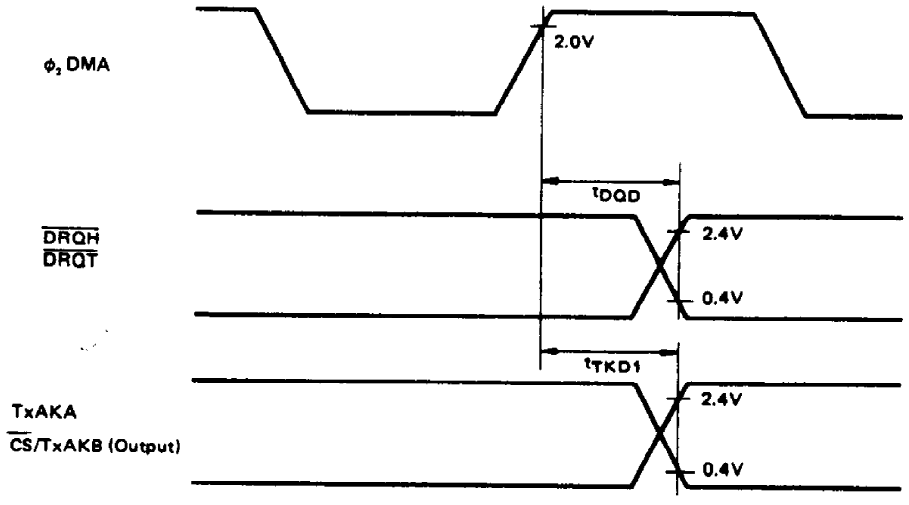


Figure 5 Timing of $\overline{DRQH}$, $\overline{DRQT}$, TxAK Outputs

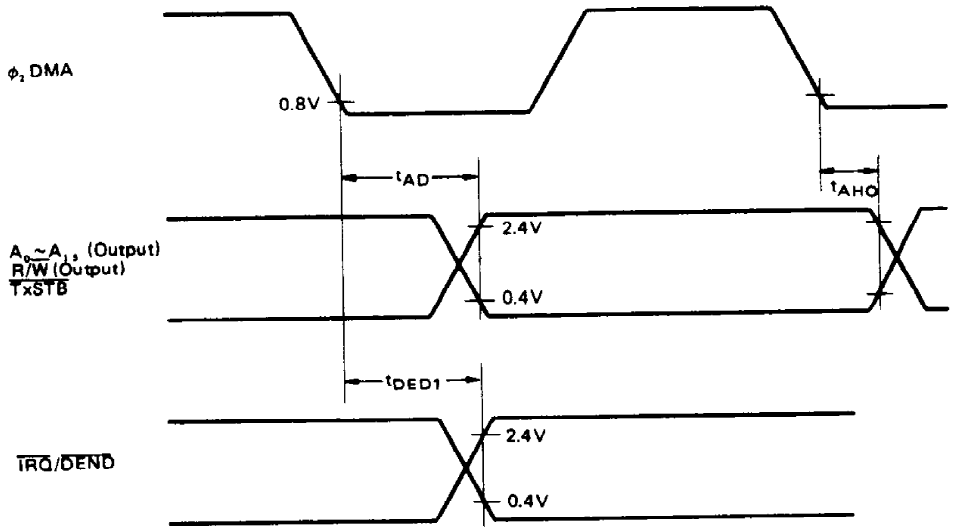
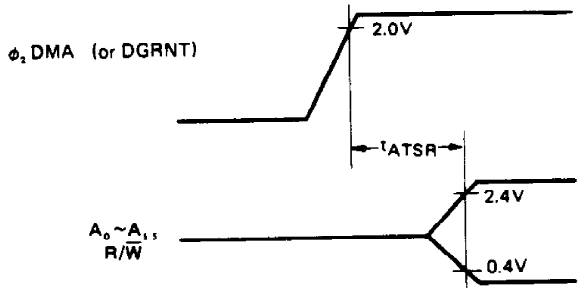


Figure 6 Timing of Address and IRQ/DEND Outputs

Recovery Time of Address Three-state



Delay Time of Address Three-state

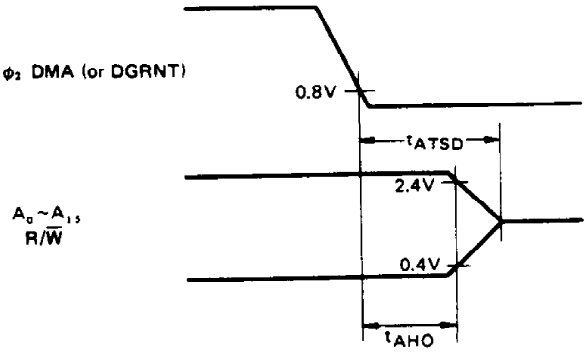


Figure 7 Timing of Address Three-state

5

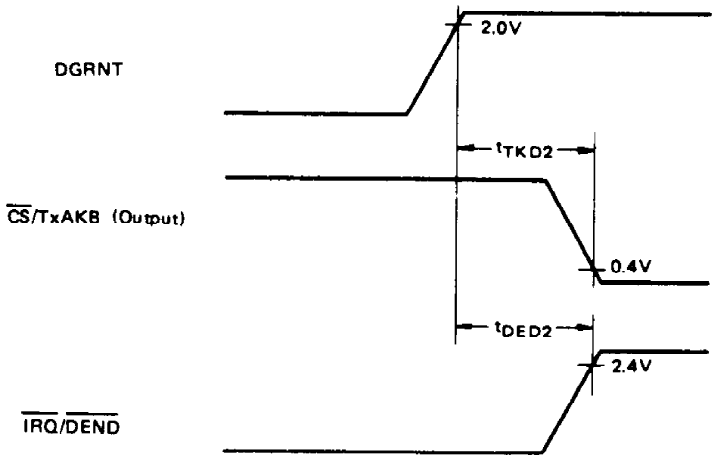
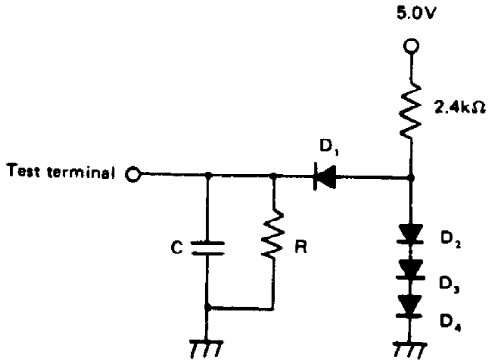


Figure 8 Timing of Synchronous DGRNT Output



Test terminal	C	R
D_0~D_7	130 pF	11 kΩ
A_0~A_{15}, R/$\overline{W}$	90 pF	16 kΩ
$\overline{CS}$/TxAKB	50 pF	24 kΩ
All other outputs	30 pF	24 kΩ

D_1~D_4 : 1S2074 Ⓗ or equivalent.

Figure 9 Load Circuit

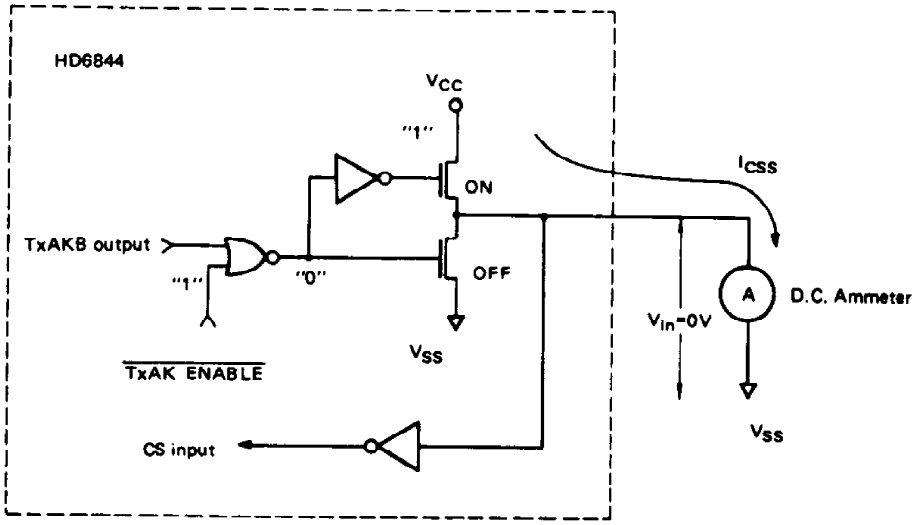


Figure 10 Source Current Measurement Circuit for $\overline{CS}$/TxAKB Terminal

■ DEVICE OPERATION

The DMAC has fifteen addressable registers, eight of them are sixteen bits in length. Each channel has a separate Address Register and a Byte Count Register, each of which is sixteen bits. There are also four Channel Control Registers. The three General Control Registers common to all four channels are the Priority Control Register, the Interrupt Control Register, and the Data Chain Register.

To prepare a channel for DMA, the Address Registers must be loaded with the starting memory address and the Byte Count Register loaded with the number of bytes to be transferred. The bits in the Channel Control Register establish the direction of the transfer, the mode, and the address increment or decrement after each cycle. Each channel can be set for one of three transfer modes: Three-State Control (TSC) Steal, Halt Steal, or Halt Burst. Two read-only status bits in the Channel Control Register indicate when the channel is busy transferring data and when the DMA transfer is completed.

The Priority Control Register enables the transfer requests from the peripheral controllers and establishes either a fixed priority or rotating priority scheme of servicing these requests.

When the DMA transfer for a channel is complete (the Byte Count Register is zero), a $\overline{\text{DMA End}}$ signal is directed to the peripheral controller and an $\overline{\text{IRQ}}$ goes to the MPU. Enabling of these interrupts is done in the interrupt Control Register. The IRQ flag bit is read from this register.

Chaining of data transfers is controlled by the Data Chain Register. When enabled, the contents of the Address and Byte Count Registers for channel #3 are put into the registers of the channel selected for chaining when its Byte Count Register becomes zero. This allows for repetitively reading or writing a block of memory.

During the DMA mode, the DMAC controls the address bus and data bus for the system as well as providing the R/$\overline{\text{W}}$ line and a signal to be used as VMA. When a peripheral device controller desires a DMA transfer, it is requested by a Transfer Request. Assuming this request is enabled and meets the test of highest priority, the DMAC will issue a DMA Request. When the DMAC receives the DMA Grant, it gives a Transfer Acknowledge to the peripheral device controller, at which time the data is transferred. When the channel's Byte Count Register equals zero, the transfer is complete and a $\overline{\text{DMA End}}$ is given to the peripheral device controller, and an $\overline{\text{IRQ}}$ is given to the MPU.

• Initialization

During a power-on sequence, the DMAC is reset via the $\overline{\text{RES}}$ input. All registers, with the exception of the Address and Byte Count Registers, are set to a logic "0" state. This disables all requests and the Data Chain function while masking all interrupts. The Address, Byte Count, and Channel Control Registers must be programmed before the respective transfer request bit is enabled in the Priority Control Register.

• Transfer Modes

There are three ways in which a DMA transfer may be done. The one used is determined by the data transfer rate required, the number of channels attached, and the hardware complexity allowable. Refer to Figures 12, 16 and 17.

Two of the modes, TSC Steal and Halt Steal, are done by cycle-stealing from the MPU. The Three-State Control (TSC) Steal mode is initiated by the DMAC bringing the $\overline{\text{DRQT}}$ line "Low". This line goes to the system clock driver which returns a "High" on DGRNT on the rising edge of the system ϕ_1 clock. The DGRNT signal must cause the address control and data lines to go to the high impedance state. The DMAC now supplies the address from the Address Register of the channel requesting. It also supplies the R/$\overline{\text{W}}$ signal as determined from the Channel Control Register. After one byte is transferred, control is returned to the MPU. This method stretches the ϕ_1 and ϕ_2 clocks while the DMAC uses the memory.

The second method of cycle-stealing is the Halt Steal mode. This method actually halts the MPU instead of stretching the ϕ_1 clock for the transfer period. This mode is initiated by the DMAC bringing the $\overline{\text{DRQH}}$ line "Low". This line connects to the MPU $\overline{\text{HALT}}$ input. The MPU Bus Available (BA) line is the DGRNT input to the DMAC. While the MPU is halted, its Address Bus, Data Bus, and R/$\overline{\text{W}}$ are in the high impedance state. The DMAC now supplies the address and R/$\overline{\text{W}}$ line. After one byte is transferred, the $\overline{\text{HALT}}$ line is returned "High" and the MPU regains control. In this mode, the MPU stops internal activity and is removed from the system while the DMAC uses the memory.

The third mode of transfer is the Halt Burst mode. This mode is similar to the Halt Steal mode, except that the transfer does not stop with one byte. The MPU is halted while an entire block of data is transferred. When the channel's Byte Count Register equals zero, the transfer is complete and control is returned to the MPU. This mode gives the highest data transfer rate, at the expense of the MPU being inactive during the transfer period.

■ INPUT/OUTPUT FUNCTIONS

• DMAC Interface Signals for the MPU

The DMAC interfaces with the HMCS6800 MPU through the eight-bit bidirectional data bus, the $\overline{\text{CS}}$ line, five address lines, an $\overline{\text{IRQ}}$ line, the Read/Write line, and the $\overline{\text{RES}}$ line. These signals, in conjunction with the HMCS6800 VMA output, permit the MPU to have access to the DMAC. Four other lines associated with the MPU and the clock driver are the $\overline{\text{DRQT}}$, $\overline{\text{DRQH}}$, DGRNT, and the ϕ_2 DMA.

Bidirectional Data (D_0~D_7)

I/O Pin No. 28 ~ 21

The Bidirectional Data lines ($D_0 \sim D_7$) allow for data transfer between the DMAC and the MPU. The data bus output drivers are three-state devices that remain in the high impedance state except when the MPU performs DMAC read operations.

Chip Select/Transfer Acknowledge B ($\overline{\text{CS}}$/T x AKB)

This line is multiplexed, serving both as an input and an output. $\overline{\text{CS}}$/TxAKB is an output in the four-channel mode during the DMA transfer. At all other times, it is a high impedance TTL compatible input used to address the DMAC. The DMAC is selected when $\overline{\text{CS}}$/TxAKB is "Low". VMA must be used in generating this input to insure that false selects will not occur. Transfers of data to and from the DMAC are then performed under the control of the ϕ2 DMA, Read/Write, and A_0~A_4 address lines. In the four-channel mode when TxAKB is needed, the $\overline{\text{CS}}$ gate must have an open-collector output (a pull-up resistor should not be used). In the two-channel mode, $\overline{\text{CS}}$/TxAKB is always an input.

Address Lines (A_0~A_4)

Address lines A_0~A_4 are both input and output lines. In the MPU mode, these are high impedance inputs used to address the DMAC registers. In the DMA mode, these lines are outputs which are set to the contents of the Address Register of the channel being processed.

5

Interrupt Request/DMA End ($\overline{IRA}/\overline{DEND}$)

Output Pin No. 33

$\overline{IRQ}/\overline{DEND}$ is a TTL compatible, active "Low" output that is used to interrupt the MPU and to signal the peripheral controller that the data block transfer has ended. If the Interrupt has been enabled, the $\overline{IRQ}/\overline{DEND}$ line will go "Low" after the last DMA cycle of a transfer. An open collector gate must be connected to DGRNT and $\overline{IRQ}/\overline{DEND}$ to prevent false interrupts from the DEND signal when interrupts are not enabled. Refer to the section of "DMA End Control".

- **Read/Write (R/$\overline{W}$)**

I/O Pin No. 3

Read/Write is a TTL compatible line that is a high impedance input in the MPU mode and an output in the DMA mode. In the MPU mode, it is used to control the direction of data flow through the DMAC's input/output data bus interface. When Read/Write is "High" (MPU read cycle) and the chip is selected, DMAC data output buffers are turned on and a selected register is read. When it is "Low", the DMAC output drivers are turned off and the MPU writes into a selected register.

In the DMA mode, Read/Write is an output to drive the memory and peripheral controllers. Its state is determined by bit 0 of the Channel Control Register for the channel being serviced. When Read/Write is "High", the memory is read and the data from the memory is written into the peripheral controller. When it is "Low", the peripheral controller is read and its data stored in the memory. In the DMA mode, the DMAC data buffers are off.

Rest ($\overline{RES}$)

Input Pin No. 39

The $\overline{RES}$ input provides a means of resetting the DMAC from an external source. In the "Low" state, the $\overline{RES}$ input causes all registers, with the exception of the Address and Byte Count Registers, to be reset to the logic "0" state. This disables all transfer requests, masks all interrupts, disables the data chain function, and puts each Channel Control Register into the condition of memory write, Halt Steal transfer mode, and address increment.

- **Transfer Signals to the MPU**

Two DMA request output lines and a DMA Grant input line, together with the system clock, synchronize the DMAC with the MPU system.

DMA Request Three-State Control Steal ($\overline{DRQT}$)

Output Pin No. 37

This active "Low" output requests a DMA transfer for a channel configured for the TSC Steal transfer mode. This line is connected to the system clock driver, requesting a ϕ_1 clock stretch. It will remain in the "Low" state until the transfer has begun.

DMA Request Halt ($\overline{DRQH}$)

Output Pin No. 36

This active "Low" output requests a DMA transfer for a channel programmed for the Halt Steal or Halt Burst mode transfer. This line is connected directly to the MPU $\overline{HALT}$ input and remains "Low" until the last byte has begun to be transferred.

DMA Grant (DGRNT)

Input Pin No. 38

This is a high impedance input to the DMAC, giving it control of the system busses. For the TSC Steal mode, the signal comes from the system clock drive circuit (DMA Grant), indicating that the clock is being stretched. For either of the Halt modes, the signal is the Bus Available from the MPU, indicating that the MPU has halted and turned control of its busses over to the DMAC. For a design involving TSC Steal and Halt mode transfers, this input must be the OR of the clock driven DMA Grant and the MPU BA.

ϕ_2 DMA

Input Pin No. 40

Transferring in and out of the DMAC registers, sampling of channel request lines and gating of other control signals to the system is done internally in conjunction with the ϕ_2 DMA high impedance input. This input must be the system memory clock (non-stretched ϕ_2 clock).

- **Transfer Signals From the Peripheral Controller**

Transfer Request ($T \times RQ_0 \sim T \times RQ_3$)

Input Pin No. 32 ~ 29

Each of the four channels has its own high impedance input request for transfer line. The peripheral controller requests a transfer by setting its T×RQ line "High" (a logic "1"). The lines are sampled according to the priority and enabling established in the Priority Control Register. In the Steal mode and the first byte of the Halt Burst mode, the T×RQ signals are tested on the positive edge of ϕ_2 DMA and the highest priority channel is strobed. Once strobed, the T×RQs are not tested until that channel's data transfer is finished. In the succeeding bytes of the Halt Burst mode transfer, the T×RQ is tested on the negative edge of ϕ_2 DMA, and data is transferred on the next ϕ_2 DMA cycle if T×RQ is "High".

- **Transfer Signals to the Peripheral Controller**

Two encoded lines select the channel to be serviced. A strobe line acknowledges the request and performs the transfer. The $\overline{DEND}$ line signals to the peripheral controller that the DMA transfer is completed.

Transfer Acknowledge A (T × AKA)

Output Pin No. 35

The Transfer Acknowledge LA (T×AKA) is a TTL compatible output used in conjunction with the $\overline{CS}$/T×AKB line to select the channel to be strobed for transfer and to give the DMA End Signal. In the two-channel mode, only T×AKA is used to select channel 0 or channel 1, and $\overline{CS}$/T×AKB is always an input.

Chip Select/Transfer Acknowledge B ($\overline{CST}$/ × AKB)

I/O Pin No. 2

In the DMA mode, this dual purpose line is encoded together with T×AKA to select the channel being serviced. Table 1 shows the encoding order.

$\overline{CS}$/T×AKB	T×AKA	Channel #
0	0	0
0	1	1
1	0	2
1	1	3

Transfer Strobe ($\overline{T \times STB}$)

Output Pin No. 34

The $\overline{T \times STB}$ causes acknowledgement to be given to the peripheral controller and transfers the data to or from the memory. This line is also intended to be the VMA signal for the system in the DMA mode. In a one-channel system, $\overline{T \times STB}$ may be inverted and run to the peripheral controller's Acknowledge input. In a two or four-channel system, $\overline{T \times STB}$ enables the decode of T×AKA and $\overline{CS}$/T×AKB to select the device controller to be acknowledged.

Interrupt Request/DMA End ($\overline{IRQ}/\overline{DEND}$)

In the DMA mode, this dual purpose line is "Low" for the last byte of transfer, indicating a DMA End. This occurs when the Byte Count register decrements to zero.

This line, through the decode of TxAKA and $\overline{CS}$/TxAKB, can be used to strobe a DMA End to each device controller.

- **Address Lines to the Memory**

Address Lines ($A_0 \sim A_{15}$)

These output lines are in the high impedance state during the MPU mode. In the DMA mode, these lines are outputs which are set to the contents of the Address Register of the channel being processed.

■ THE DMAC REGISTERS

The HD6844 (DMAC) has Address Register (ADR), Byte Count Register (BCR), Channel Control Register (CHCR), and General Control Register (GCR).

General Control Register (GCR) is composed of Priority Control Register (PCR) that controls priority among the channels, Interrupt Control Register (ICR) that controls interrupt and Data Chain Control Register (DCR) that controls data chain function. Refer to Table 2 and Figure 1.

These are Read/Write registers and MPU can exchange the data with DMAC when $\overline{CS}$ is at "Low" level. $A_0 \sim A_4$ specifies the address of the registers. How to specify the registers is shown in Table 2.

2-byte ADR and BCR can be read or written by one instruction, using 2-byte instruction of the MPU.

- **Function of Internal Registers**

ADR (Address Register)

Each channel has 16-bit Address Register. Initial address of memory used for DMA transfer is programmed to this register. The contents of ADR are output to address bus ($A_0 \sim A_{15}$) during DMA transfer operation. When 1-byte transfer has completed, the 16-bit address is incremented or decremented by one.

The address which the MPU reads out is the renewed one, that is, the memory address for the next transfer. When 1-block transfer has completed, final memory address +1 or -1 is read out.

Table 2 Internal Registers of the DMAC

Register Name	Symbol	Channel	Address Bus Signal					Address (Hexadecimal)
			A_4	A_3	A_2	A_1	A_0	
Address Register	ADRH	0	0	0	0	0	0	00
	ADRL	0	0	0	0	0	1	01
Byte Count Register	BCRH	0	0	0	0	1	0	02
	BCRL	0	0	0	0	1	1	03
Address Register	ADRH	1	0	0	1	0	0	04
	ADRL	1	0	0	1	0	1	05
Byte Count Register	BCRH	1	0	0	1	1	0	06
	BCRL	1	0	0	1	1	1	07
Address Register	ADRH	2	0	1	0	0	0	08
	ADRL	2	0	1	0	0	1	09
Byte Count Register	BCRH	2	0	1	0	1	0	0A
	BCRL	2	0	1	0	1	1	0B
Address Register	ADRH	3	0	1	1	0	0	0C
	ADRL	3	0	1	1	0	1	0D
Byte Count Register	BCRH	3	0	1	1	1	0	0E
	BCRL	3	0	1	1	1	1	0F
Channel Control Register	CHCR	0	1	0	0	0	0	10
	CHCR	1	1	0	0	0	1	11
	CHCR	2	1	0	0	1	0	12
	CHCR	3	1	0	0	1	1	13
Priority Control Register	PCR	–	1	0	1	0	0	14
Interrupt Control Register	ICR	–	1	0	1	0	1	15
Data Chain Control Register	DCR	–	1	0	1	1	0	16

(NOTE) 1) All the registers can be accessed by Read/Write operation. Unused bit of the register is read out "0".
2) H/L of ADR and BCR means the higher (H) 8 bits/the lower (L) 8 bits of a 16-bit register.
3) Being allocated to continuous address, 16-bit ADR and BCR can be read or written by one instruction, using MPU's 2-byte LOAD/STORE instruction.

e.g. LDX $\$\_\_\overline{0C}$ (Register Address / Address of DMAC) ..... [(ADRH 3) → (Index Register H)] [(ADRL 3) → (Index Register L)]

5

BCR (Byte Count Register)

Each channel has a 16-bit Byte Count Register. Number of DMA transfer words is programmed into this register. The content of the Byte Count Register is decremented by one everytime one-byte transfer has completed. When it becomes "0", $\overline{DEND}$ output goes "Low" level and informs I/O controller of the end of one-block DMA transfer. When $\overline{IRQ}$ is not masked, $\overline{IRQ}$ output goes "Low" level and MPU is interrupted to be informed of the end of DMA transfer. Moreover, $\overline{IRQ}$ and $\overline{DEND}$ signals are output, multiplexed with $\overline{IRQ}/\overline{DEND}$ pin.

CHCR (Channel Control Register)

Each channel has Channel Control Register. This register is used to program the control information of its corresponding channel. Structure of CHCR is shown in Table 3.

(1) R/W Control (specifies the direction of transfer)

Bit – CHCR Bit 0

This bit controls the direction of DMA transfer. When it is at "1", R/$\overline{W}$ signal of DMAC goes "High" level during DMA transfer operation. This means to read out memory and write into I/O controller, that is, data is transferred from memory to I/O controller.

When it is at "0", R/$\overline{W}$ output goes "Low" level and data is transferred from I/O controller to memory.

Table 3 Bit Structure of CHCR (Channel Control Register)

Bit No.	Name	Read/Write	Function	
			"1"	"0"
0	R/W	R/W	Transfer from memory to I/O controller (R/$\overline{W}$ output = "High")	Transfer from I/O controller to memory (R/$\overline{W}$ output = "Low")
1	Burst/Cycle Steal	R/W	Burst Mode	Cycle Steal Mode*
2	TSC/HALT	R/W	TSC Mode	HALT Mode*
3	Address down/up	R/W	Address: -1	Address: +1
4	Not used	–	–	–
5	Not used	–	–	–
6	Busy/Ready Flag	R	Busy (DMA Transfer Operation)	Ready (No DMA Transfer Operation)
7	DEND Flag	R	DMA End & Interrupt	No Interrupt

* Burst transfer in TSC mode is prohibited. R: Read, W: Write

Note that during DMA transfer operation, the function of R/$\overline{W}$ signal is accommodated to the memory Read/Write operation. Therefore, on the side of I/O device during DMA transfer operation, R/$\overline{W}$ input should be interpreted in inverse of the MPU Read/Write. That is, data should be output when R/$\overline{W}$ input is at "Low" level (In the case of MPU's read operation, I/O device outputs the data when it is at "High" level).

This arises from that during DMA transfer operation, I/O side performs data transfer independently instead of MPU. Moreover, such family LSI as HD6843 (FDC), etc. has this function and R/$\overline{W}$ signal is automatically interpreted inversely.

(2) Burst/Cycle Steal Bit – CHCR Bit 1

This bit is used to decide that DMA transfer should be performed in burst mode or cycle steal mode. When it is at "1", it specifies burst mode. That is, once DMA transfer is performed, MPU remains stopped until one-block data transfer is completed.

When this bit is "0", it specifies cycle steal mode. That is, everytime one-byte transfer has completed, MPU takes back the bus control, and DMA transfer and MPU operation are performed in time sharing.

(NOTE) Only in the case of HALT mode, burst mode can be specified. In TSC mode, burst mode cannot be specified.

(3) TSC/HALT Mode Bit – CHCR Bit 2

This bit is used to decide that DMA transfer should be performed by using MPU's TSC function or HALT function. When it is at "0", DMA transfer request signal is output from $\overline{DRQH}$ of DMAC.

When it is at "1", DMA transfer request signal is output from $\overline{DRQT}$ of DMAC.

(4) Address down/up Bit – CHCR Bit 3

This bit is used to decide that the address of memory region used for DMA transfer should be renewed up (increment of address) or down (decrement of address). When it is at "1", the address is decremented by one after one-byte transfer. When it is at "0", the address is incremented by one.

(5) Busy/Ready Flag Bit – CHCR Bit 6

This bit is a status flag to indicate whether its corresponding channel is performing DMA transfer or not. (READ only)

When it receives the first TxRQ of its corresponding channel, it goes to "1". When one-block transfer is completed and BCR becomes "0", it is reset to "0".

Also this flag is cleared when corresponding TxRQ Enable Bit in the PCR becomes "0".

(6) DEND Flag Bit – CHCR Bit 7

This bit is an interrupt flag to indicate that one-block DMA transfer of its corresponding channel has completed. (READ only).

When one-block transfer of its corresponding channel is completed and BCR becomes "0", it goes to "1". As soon as this flag is read out, i.e. CHCR of this channel is read

out, it is reset to "0".

Moreover, this bit is connected to $\overline{IRQ}$ output. When it is at "1" and IRQ enable bit (within ICR register described later) is at "1", $\overline{IRQ}$ output goes "Low" level.

PCR (Priority Control Register)

Priority Control Register is a 5-bit register to decide the operation mode of priority control circuit. Structure of PCR is shown in Table 4.

Table 4 Bit Structure of PCR (Priority Control Register)

Bit No.	Name	Read /Write	Function	
			"1"	"0"
0	TxRQ Enable #0 ($TxEN_0$)	R/W	TxRQ of Channel 0 is accepted.	TxRQ of Channel 0 is not accepted.
1	TxRQ Enable #1 ($TxEN_1$)	R/W	TxRQ of Channel 1 is accepted.	TxRQ of Channel 1 is not accepted.
2	TxRQ Enable #2 ($TxEN_2$)	R/W	TxRQ of Channel 2 is accepted.	TxRQ of Channel 2 is not accepted.
3	TxRQ Enable #3 ($TxEN_3$)	R/W	TxRQ of Channel 3 is accepted.	TxRQ of Channel 3 is not accepted.
4	Not used	–	–	–
5		–	–	–
6		–	–	–
7	Rotate Control	R/W	Rotate Mode	The order of priority is fixed at numerical order.

R: Read, W: Write

(1) TxRQ Enable Bit ($TxEN_0$~$TxEN_3$) – PCR Bit 0~3

Each channel has this TxRQ Enable bit. When it is at "1", TxRQ input of its corresponding channel is accepted to perform DMA transfer. When it goes to "0", TxRQ of its corresponding channel is masked not to be received and TxAK is not output. During DMA transfer operation, when this bit goes to "0" before BCR becomes "0", following TxRQ input is not accepted and DMA transfer is interrupted. Then contents of ADR and BCR remain unchanged. When it rises to "1" again, DMA transfer is reopened. Therefore, in the case of cycle steal DMA, it is possible for the program to change the priority of the specific channel temporarily by manipulating this bit.

(2) Rotate Control Bit – PCR Bit 7

When this bit is at "0", the order of priority among DMA channels is fixed at numerical order. That is, Channel 0 is given a first priority and then is followed by Channel $1 \rightarrow 2 \rightarrow 3$.

When this bit is at "1", priority control is due to rotate mode. That is, the channel that ended in the first time is given a first priority and the channel ended in the last time is controlled to be given a last priority.

ICR (Interrupt Control Register)

Interrupt Control Register is a 5-bit register to control $\overline{IRQ}$ output. Its structure is shown in Table 5.

(1) IRQ Enable Bit – ICR Bit 0~3

Each channel has IRQ Enable Bit. When this bit is at "1" and DEND Flag of its corresponding channel is set to "1", $\overline{IRQ}$ output goes "Low" level. But when it is at "0", $\overline{IRQ}$ output is masked not to be output even if DEND Flag is set to "1".

These bits enable to control to output only a necessary channel to $\overline{IRQ}$.

(2) IRQ Flag – ICR Bit 7

This is a read-only bit and the status of $\overline{IRQ}$ output is directly reflected on it. That is, when $\overline{IRQ}$ output goes to "Low" level, it becomes "1".

$\overline{IRQ}$ output of DMAC is output as logical OR of 4-channel DEND Flag according to the following equation.

$IRQ = (DEND_0 \cdot IRQ\ Enable_0) + (DEND_1 \cdot IRQ\ Enable_1) + (DEND_2 \cdot IRQ\ Enable_2) + (DEND_3 \cdot IRQ\ Enable_3)$

DCR (Data Chain Control Register)

Data Chain Control Register is a 4-bit register and three of those bits are used to control data chain function. Remaining one bit is used to specify 2-channel/4-channel mode.

Structure of DCR is shown in Table 6.

(1) Data Chain Enable Bit – DCR Bit 0

When this bit is at "1", data chain function of DMAC is enabled. That is, when DMA transfer of a specified channel has completed and BCR goes to "0", the contents of ADR and BCR of Channel #3 are automatically transferred to ADR and BCR of the specified channel.

(2) Data Chain Channel Bit – DCR Bit 1~2

These bits are used to specify which channel should be used for the data chain. How to specify the channel is shown in Table 7. Data Chain Channel bit specifies the channel to which data should be transfered from Channel #3. Channel #3 contains the data for replacement. Channel #3 is fixed and cannot be changed.

(3) 2/4-channel Mode Bit – DCR Bit 3

This bit has no relation to the data chain function.

It is used to specify whether $\overline{CS}$/TxAKB is used for only input pin or I/O pin. When this bit is "0", $\overline{CS}$/TxAKB becomes $\overline{CS}$ input pin in 2-channel mode since TxAKB output is not necessary for application up to 2-channel.

When this bit is "1", $\overline{CS}$/TxAKB becomes I/O pin in 4-channel mode (See Fig. 11).

5

Table 5 ICR (Interrupt Control Register)

Bit No.	Name	Read /Write	Function	
			"1"	"0"
0	IRQ Enable #0	R/W	$\overline{IRQ}$ of Channel 0 is able to be output.	$\overline{IRQ}$ output of Channel 0 is masked.
1	IRQ Enable #1	R/W	$\overline{IRQ}$ of Channel 1 is able to be output.	$\overline{IRQ}$ output of Channel 1 is masked.
2	IRQ Enable #2	R/W	$\overline{IRQ}$ of Channel 2 is able to be output.	$\overline{IRQ}$ output of Channel 2 is masked.
3	IRQ Enable #3	R/W	$\overline{IRQ}$ of Channel 3 is able to be output.	$\overline{IRQ}$ output of Channel 3 is masked.
4	Not used	–	–	–
5		–	–	–
6		–	–	–
7	IRQ Flag	R	$\overline{IRQ}$ output "Low"	$\overline{IRQ}$ output "High" (off state)

R: Read, W: Write

Table 6 Bit Structure of DCR (Data Chain Control Register)

Bit No.	Name	Read /Write	Function	
			"1"	"0"
0	Data Chain Enable	R/W	Data Chain is performed.	Data Chain is not performed.
1	Data Chain Channel	R/W	The channel which performs Data Chain is specified. (The channel where contents of ADR and BCR of Channel #3 are loaded.)	
2		R/W		
3	2/4-Channel Mode	R/W	4-Channel Mode ($\overline{CS}$/TxAKB is I/O pin.)	2-Channel Mode ($\overline{CS}$/TxAKB is designated to only input pin.)
4	Not used	–	–	–
5		–	–	–
6		–	–	–
7		–	–	–

R: Read, W: Write

Table 7 How to specify Data Chain Channel

DCR Bit 1	DCR Bit 2	Specified Channel
0	0	Channel #0
1	0	Channel #1
0	1	Channel #2
1	1	–

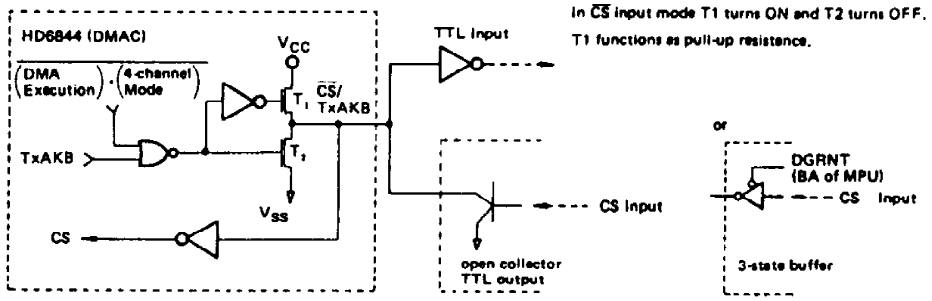


Figure 11 How to Use CS/TxAKB Pin

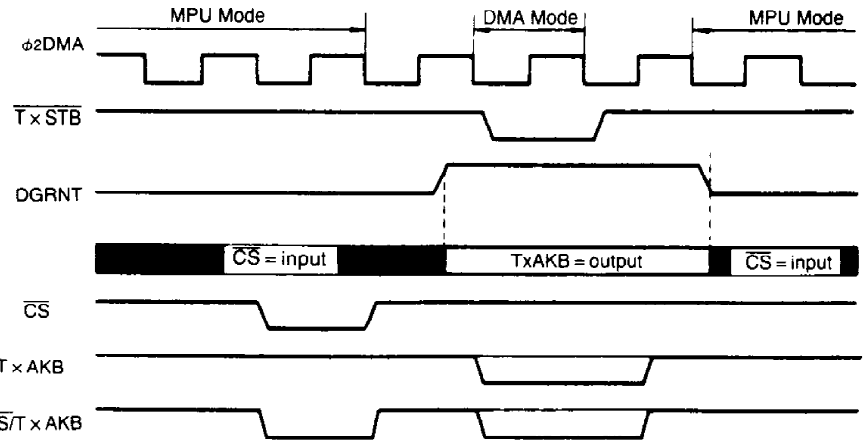


Figure 11A $\overline{CS}$/T × AKB input/output Timing

■ OPERATION OF THE DMAC

• Transfer Mode of the DMAC

There are three DMA transfer modes such as HALT Cycle Steal, HALT Burst and TSC Cycle Steal. Operation in each mode is explained in the following.

HALT Cycle Steal Mode

This is a basic DMA transfer mode utilizing HALT state of MPU. In this mode, everytime 1-byte transfer has completed, MPU takes back the bus control and executes instruction cycle. That is, DMA transfer and MPU operation are performed in time sharing.

Timing chart is shown in Fig. 12 and flow chart is shown in Fig. 13. Procedure of transfer operation is the following. (No. ① ~ ⑪ in Fig. 12 correspond to the following items.)

① $TxRQ_0$~$TxRQ_3$ input is checked at the rising edge of ϕ2DMA. When it is at "High" level, it gets into the following operation.
② $\overline{DRQH}$="Low" is output and MPU is requested to stop its operation.
③ TxAKA is driven (Level output).
④ MPU stops its operation and DMAC waits until DGRNT goes to "High" level.
⑤ When DGRNT goes to "High" level, DMAC drives TxAKB, A_0~A_{15} and R/$\overline{W}$ lines.
⑥ $\overline{TxSTB}$ is given to perform DMA transfer.
⑦ Address is incremented or decremented by one and number of transfer words is decremented by one.
⑧ When $\overline{DRQH}$ rises to "High" level, MPU gets into Instruction Cycle again.
⑨ TxRQ falls to "Low" level.
⑩ A_0~A_{15} and R/$\overline{W}$ get into high impedance state again.
⑪ DGRNT falls to "Low" level.

[Note] $TxRQ_0$~$TxRQ_3$ input in HALT cycle steal is, in principle as shown in Fig. 12, set to "High" every 1-byte transfer on account of I/O request. When $\overline{TxSTB}$ of the DMAC is driven, it is reset to "Low". Take care not to be against this principle, or the following states may happen.

(1) In the case where TxRQ becomes "High", but it is reset to "Low" before DGRNT becomes "High". In this case, the DMAC is in the wait state without sending out $\overline{TxSTB}$ until TxRQ rises to "High" again. As $\overline{DRQH}$ remains "Low" the MPU is forced to be stopped, and the system is in dead lock state until TxRQ rises to "High" again (Fig. 14).

(2) In the case where TxRQ is not reset to "Low" though $\overline{TxSTB}$ has been driven.
In this case, unless TxRQ returns to "Low" by the time ϕ_2DMA rises after $\overline{TxSTB}$ has risen to "High", it is considered as a new I/O request, which leads the above-mentioned operation ①,②→.
If TxRQ falls to "Low" immediately after that, the same state as (1) happens (Fig. 15).

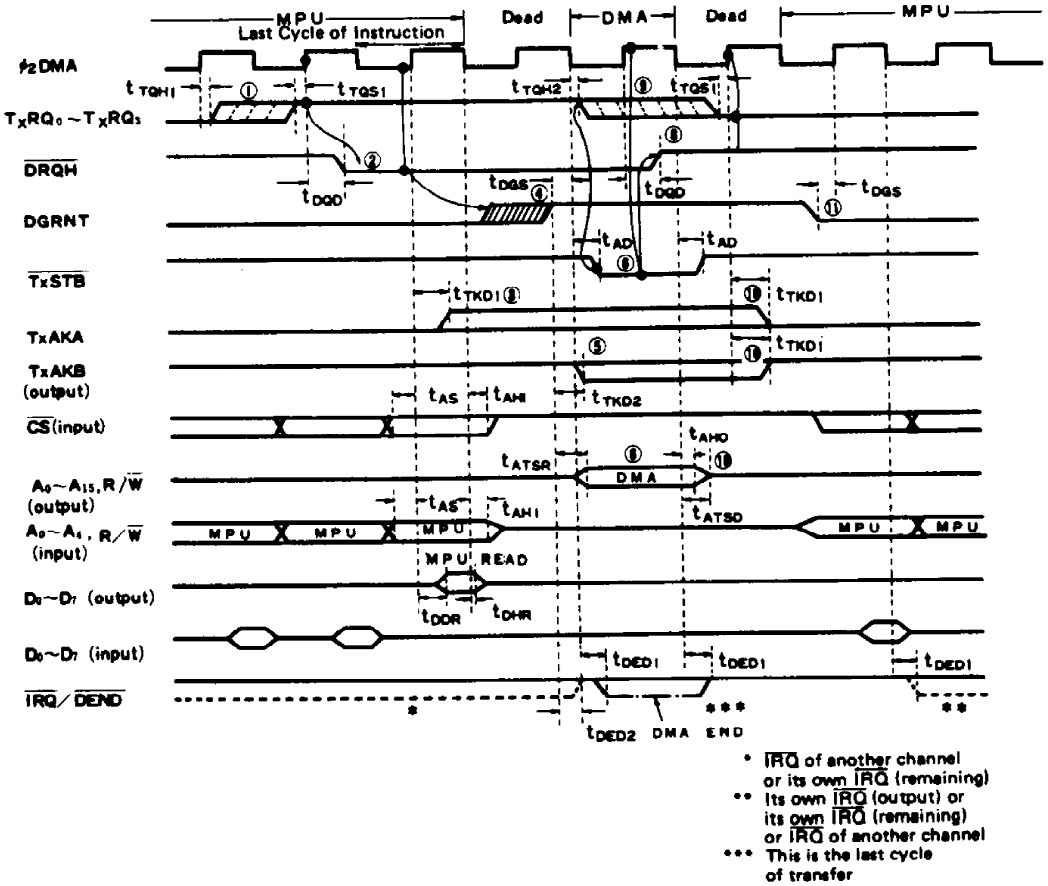


Figure 12 HALT Cycle Steal Mode

5

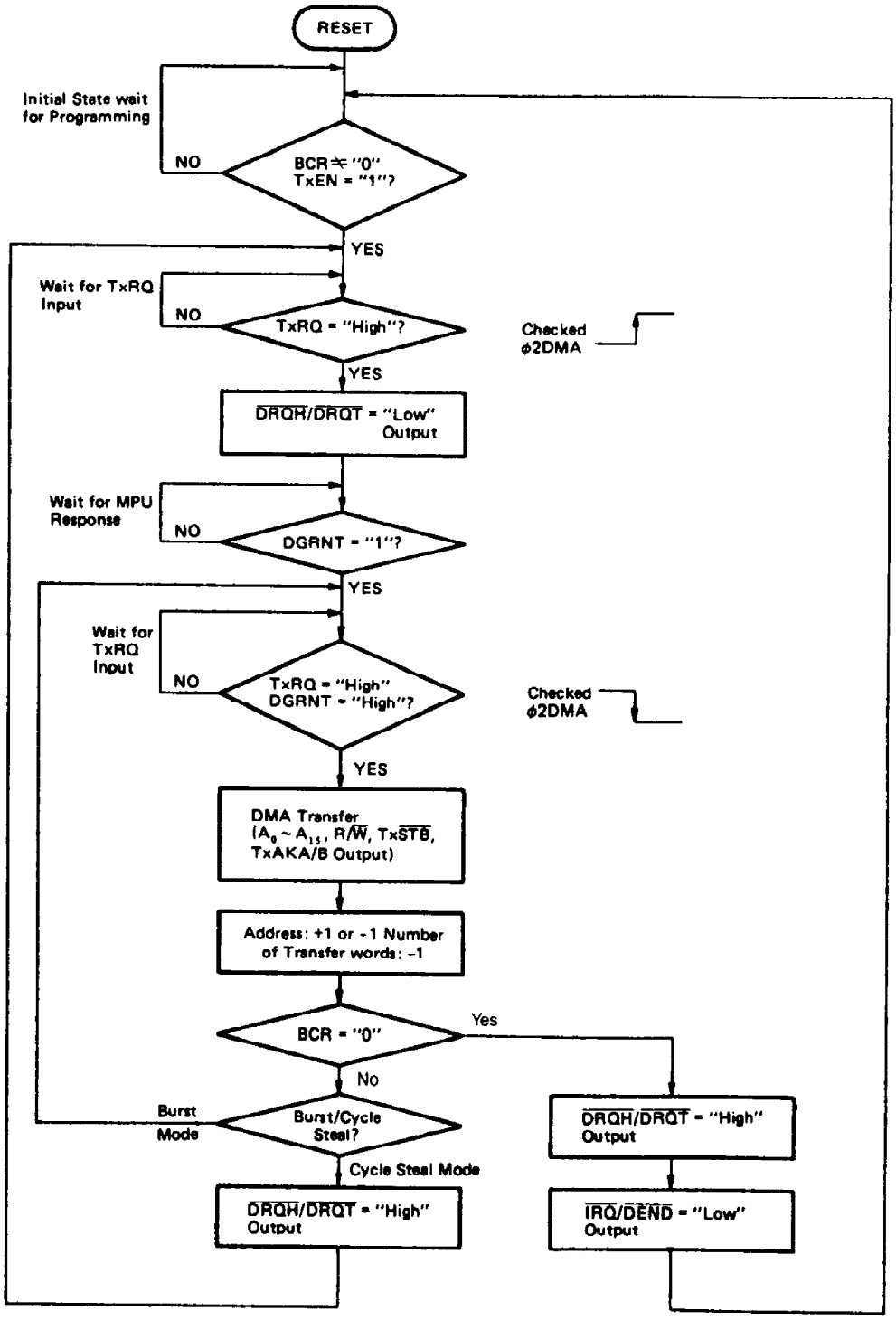


Figure 13 Flow Chart of DMAC Operation

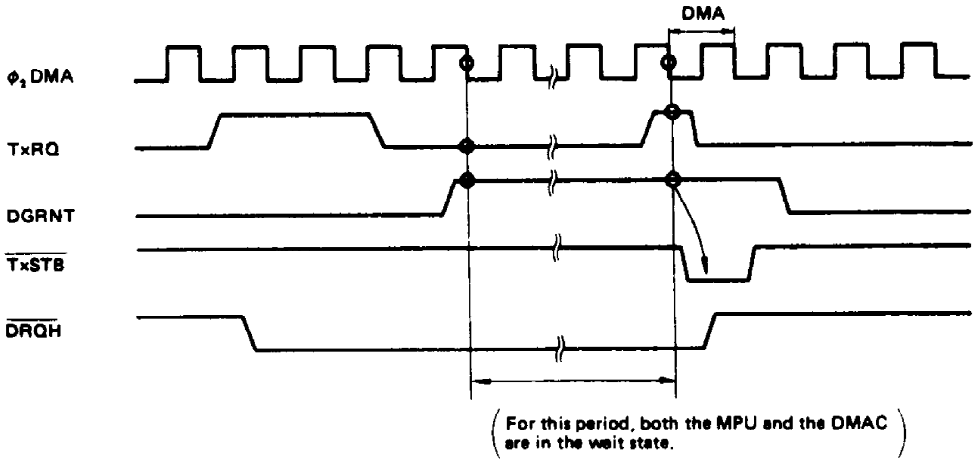


Figure 14 Extraordinary TxRQ Input (1)

In the case where TxRQ is reset to "Low" before the transfer

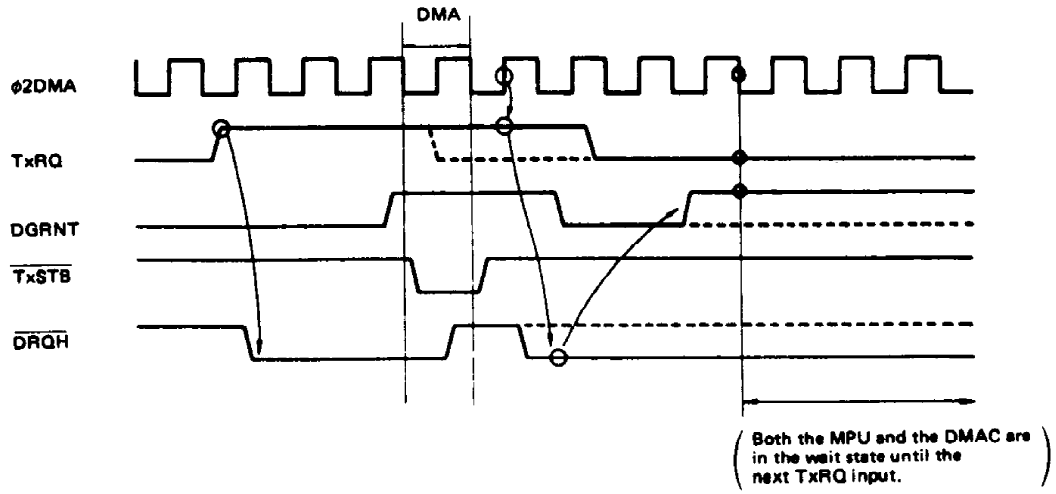


Figure 15 Extraordinary TxRQ Input (2)

In the case where TxRQ doesn't fall to "Low" after the transfer has been completed.

5

HALT Burst Mode

In the case of cycle steal mode, MPU gets into Instruction Cycle everytime 1-byte transfer has completed. But in the case of burst mode, MPU remains stopped until 1-block transfer is finished. That is, $\overline{DRQH}$ continues to be output "Low" level until BCR becomes "0".

Its timing chart and flow chart are shown in Fig. 16 and Fig. 13 respectively. Procedure of transfer is the following (No. ① ~ ⑭ in Fig. 16 correspond to the following items).

① TxRQ input is checked at the rising edge of ϕ_2 DMA. When it is at "High" level, it gest into the following operation.
② $\overline{DRQH}$="Low" level is given and MPU is requested to stop its operation.
③ TxAKA is driven.
④ MPU stops and DMAC waits for DGRNT rising "High" level.
⑤ When DGRNT rises "High" level, DMAC drives TxAKB, $A_0 \sim A_{15}$, and R/$\overline{W}$ lines.
⑥ $\overline{TxSTB}$ is sent out to perform DMA transfer.
⑦ Address is incremented or decremented by one and number of transfer words is decremented by one.
⑧ TxRQ falls to "Low" level.
⑨ When number of transfer words is 0, from ⑪ to ⑭ operations are performed.
⑩ When BCR is not "0", TxRQ is checked at the falling edge of ϕ_2 DMA.When TxRQ is at "High" level, DMA transfer is performed through ⑥ ~ ⑧ again. When TxRQ is not at "High" level, DMAC waits for becoming "High" level.
⑪ $\overline{IRQ}/\overline{DEND}$ output goes to "Low" level.
⑫ $\overline{DRQH}$ output rises to "High" level and MPU gets into Instruction Cycle again.
⑬ $A_0 \sim A_{15}$ and R/$\overline{W}$ get into high impedance state.
⑭ DGRNT falls to "Low" level.

The transfer of the first byte (① ~ ⑥) is performed in the same way as that in HALT cycle steal mode. But in the second-byte and subsequent transfer, TxRQ is checked at the falling edge of ϕ_2 DMA and if TxRQ is at "High" level, DMA transfer is performed at the following cycle. Therefore, a high-speed response (MAX. 1 byte/1 cycle) is feasible.

In burst mode, TxRQ should be also, in principle, set to "High" when I/O request is asserted, and reset to "Low" when $\overline{TxSTB}$ goes to "Low". If TxRQ is asserted as level input without being reset, DMA transfer is performed at all cycles of ϕ_2 DMA since TxRQ is always at "High" level at the falling edge of ϕ_2 DMA. Its example is shown in the second-byte and the third-byte transfer in Fig. 16.

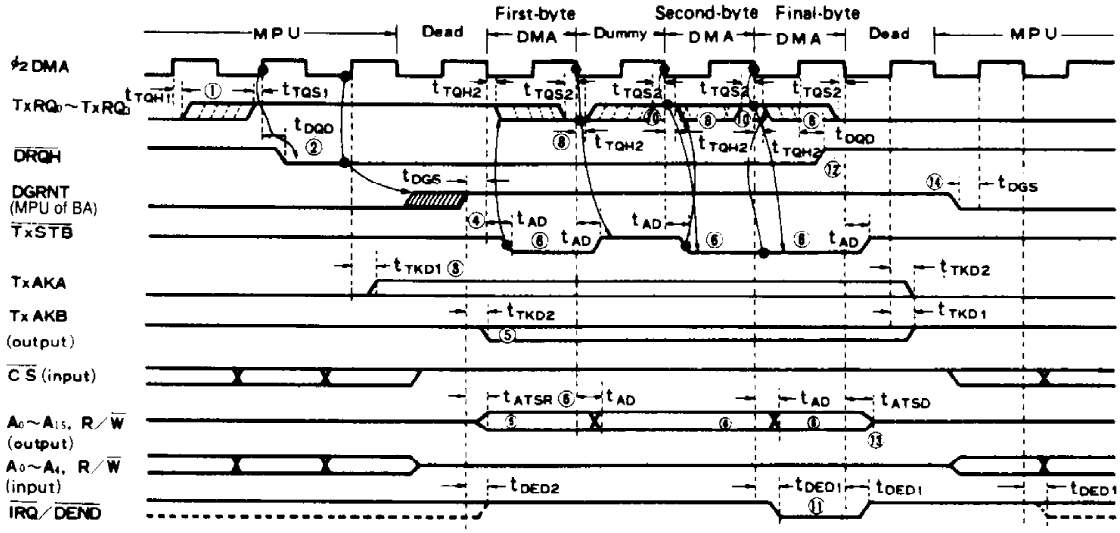


Figure 16 HALT Burst Mode

TSC Cycle Steal Mode

In the above-mentioned modes, DMA is performed by using the HALT function of the MPU. In TSC cycle steal mode, DMA is performed by using the TSC function of the MPU.

Its timing chart and flow chart are shown in Fig. 17 and Fig. 13 respectively.

Basic operation of the DMAC is the same as that in HALT cycle steal mode, but the detailed timing is different. The difference is explained in the following.

(1) $\overline{DRQT}$ is used for DMA transfer request instead of $\overline{DRQH}$.
(2) $\overline{DRQT}$ is sent to the external clock control circuit to extend clock E (ϕ_2) of MPU.
(3) To DGRNT, the external clock control circuit inputs response signals.

In TSC mode, there isn't a burst mode. Because the MPU clock cannot be extended for a long time because MPU performs dynamic operation. When TSC mode is specified, $\overline{DRQT}$ returns to "High" and the MPU gets into the instruction cycle everytime 1-byte transfer has finished.

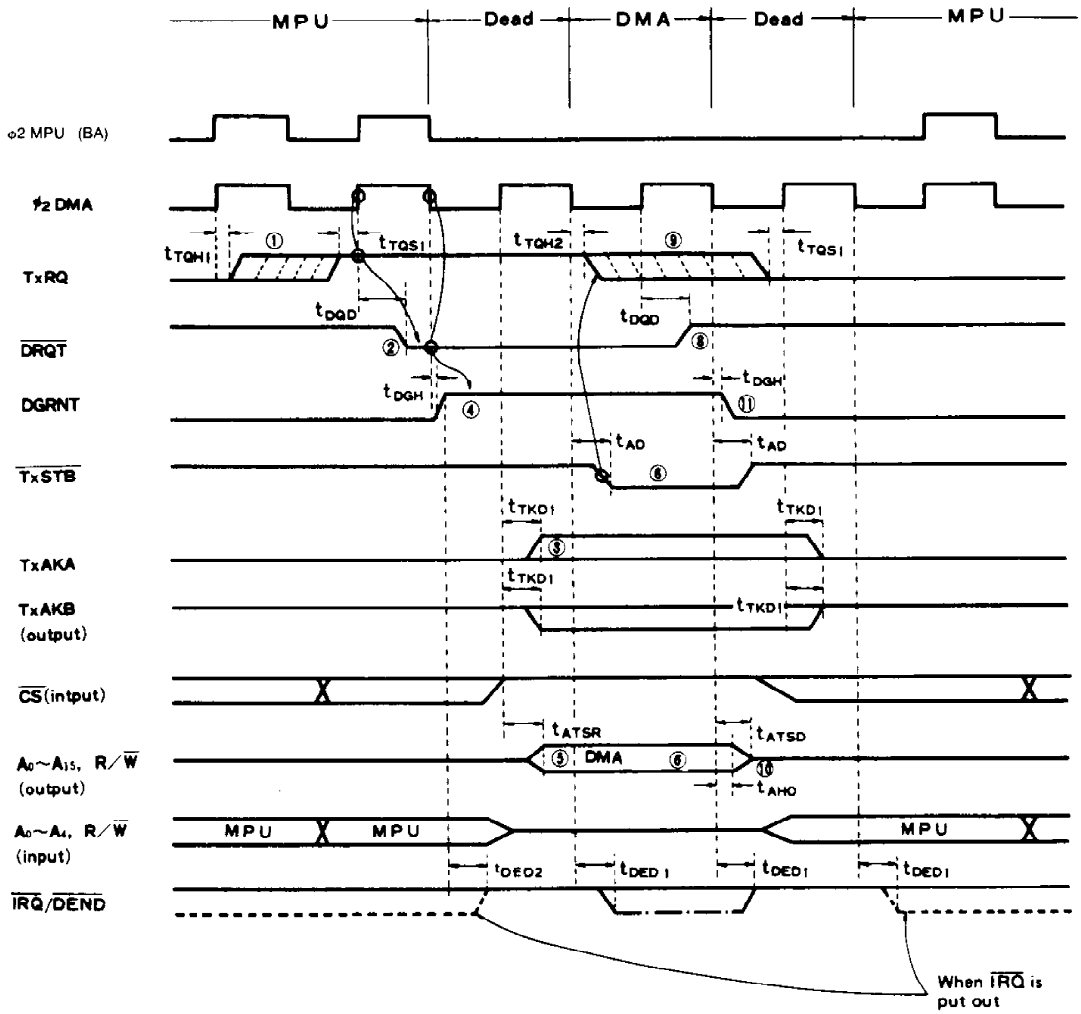


Figure 17 TSC Cycle Steal Mode

- **Priority Control**

Basic priority Control

There are two kinds of the DMAC priority control function. One is to mask TxRQ on each channel by TxRQ Enable bit of PCR. The other is priority-order-determining-circuit which the DMAC has as a hardware.

Moreover, the priority-order-determining-circuit has two operation modes (the rotate mode and the normal mode).

Structure of the priority control circuit is shown in Fig. 18. As shown in Fig. 18, TxRQ of the channel whose TxRQ Enable bit is at "1" level becomes an input of the priority-order-determining-circuit. Then it is checked whether TxRQ is at "High" level or not.

(Note) In this case, ZERO flag needs to be at "1" level. ZERO flag will be described later.

If one of $TxRQ_0$~$TxRQ_3$ is at "High" level, its channel is selected, being given a first priority. Then it is latched by an executing-channel-number-latch-circuit to perform DMA transfer. Once an executing channel is determined and latched, it is unchanged until its DMA transfer has been completed. That is, the channel number strobe signal of DMAC doesn't go to "1" and the contents of the channel-number-latch-circuit are unchanged. In the cycle steal mode, the channel is fixed until 1-byte transfer has completed. In the burst mode, it is fixed until BCR becomes "0".

5

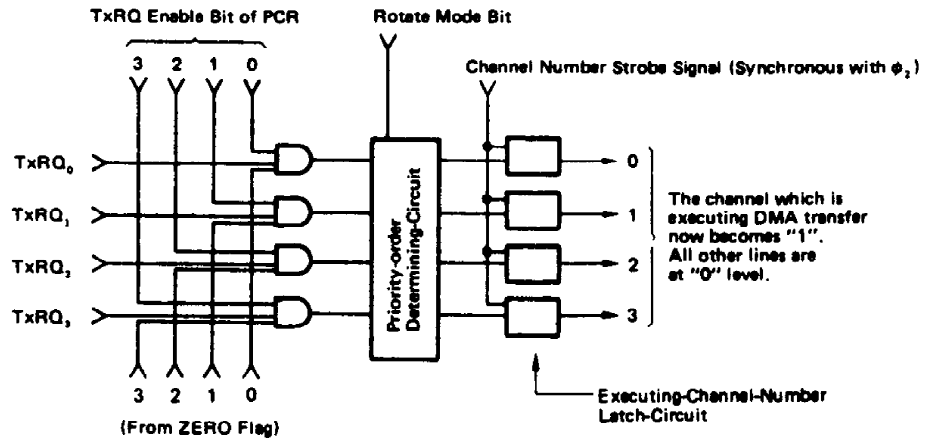


Figure 18 Structure of Priority Control Circuit

Therefore, once a long-period DMA transfer of a channel is performed in the burst mode, other channels need to wait until it has completed even if they have higher priority than the channel. Take much care to this point in designing response time to TxRQ of DMA channel.

(Note) As explained above, TxRQ input is latched internally. So once it is accepted and latched, the channel number cannot be changed even though it returns to "Low". But as explained in HALT Cycle Steal Mode, DMA transfer is not performed unless TxRQ rises to "High" again.

Strobe timing of executing-channel-number-latch-circuit which allow modification or decision of executing channel is shown in Fig. 19.

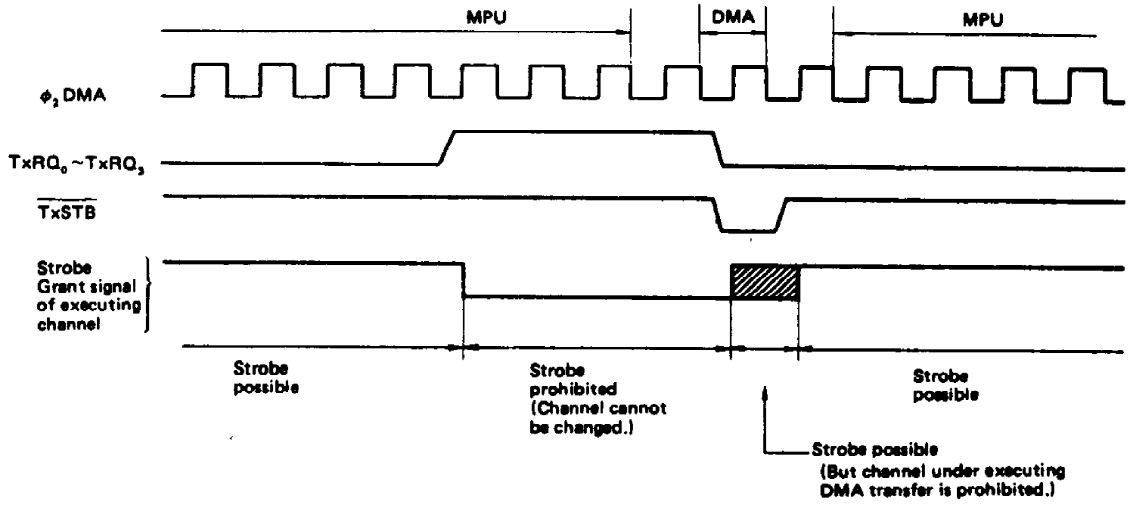


Figure 19 Strobe Timing of Executing-Channel-Number-Latch-Circuit (the cycle steal mode)

But, as shown in Fig. 19, only the channel under executing DMA transfer is prohibited to accept TxRQ during DMA transfer operation, in order that one more byte transfer may not be performed when the reset timing of TxRQ is delayed. Strobe timing in the burst mode is shown in Fig. 20.

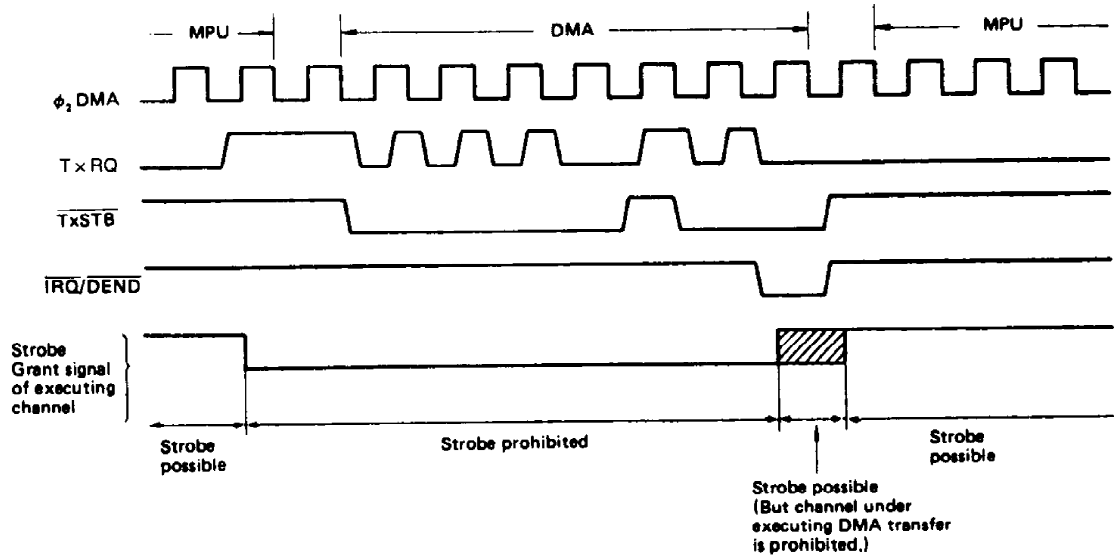


Figure 20 Strobe Timing of Executing-Channel-Number-Latch-Circuit (the burst mode)

Rotate Mode

There are two operation modes in priority-order-determining circuit. These are Normal Mode and Rotate Mode. In the normal mode, the order of priority is fixed at numerical order. (Channel 0 is given a first priority and then is followed by Channel 1 → 2 → 3.) In the rotate mode, the channel next to the channel with which DMA was executed in the last sequence, is given a first priority and the channel in the last sequence is given a last priority. But immediately after it gets into the reset state, the order of priority is the following: Channel 0 → 1 → 2 → 3.

An example of the rotate mode is shown in Fig. 21.

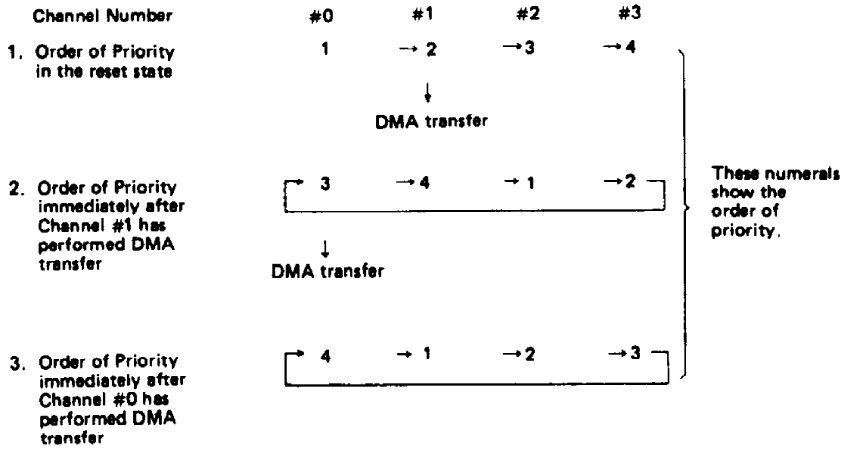
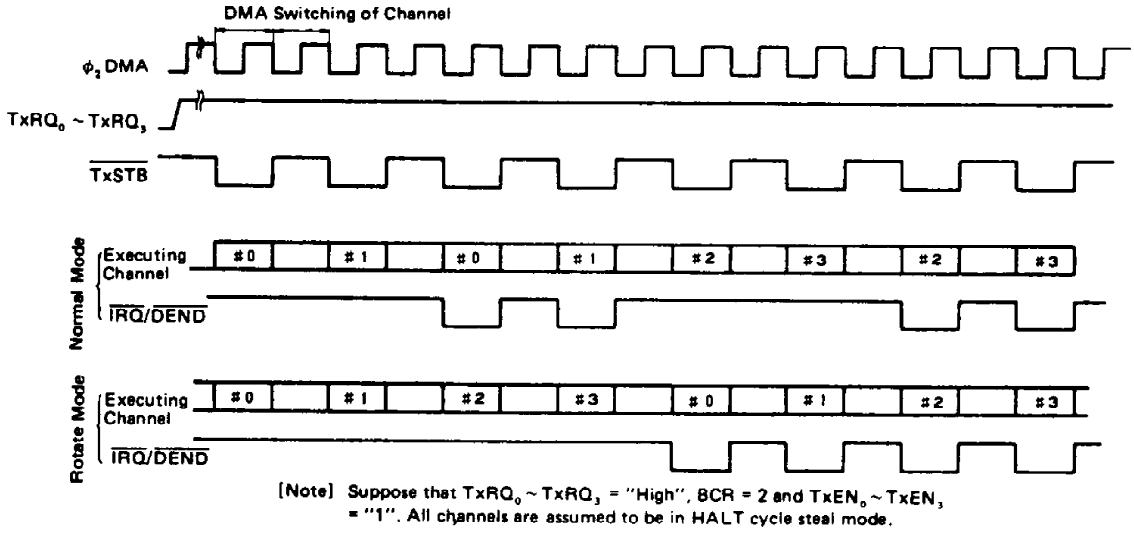


Figure 21 Example of Operation in the Rotate Mode

Next, Fig. 22 shows an example of the difference between the operation in the rotate mode and that in the normal mode. In this example, TxRQ of all channels is always at "High" level. Moreover, BCR=2 and TxEN=1 are assumed. As a transfer mode, HALT cycle steal mode is used.

5



[Note] Suppose that TxRQ$_0$ ~ TxRQ$_3$ = "High", 8CR = 2 and TxEN$_0$ ~ TxEN$_3$ = "1". All channels are assumed to be in HALT cycle steal mode.

Figure 22 Difference between the operation in the rotate mode and that in the normal mode

The reason why the order of priority is not #0 → #0 → #1 → #1 → ––– in the normal mode is that during DMA transfer operation, TxRQ of an executing channel is prohibited from being accepted.

DMA Operation Timing with priority control

When more than 2 channels perform DMA transfer in parallel, the abovementioned priority-order-determining-circuit is used to determine the priority. The channel with lower priority waits until the channel with higher priority completes the transfer. Then it gets into DMA transfer operation. In this case, The following combinations of transfer modes are conceivable.

(1) From HALT mode to HALT mode (Fig. 23)
(2) From TSC mode to TSC mode (Fig. 24)
(3) From HALT mode to TSC mode
(4) From TSC mode to HALT mode } (Fig. 25)

In changing from HALT mode to HALT mode, only one dead cycle is intervened. That is, even in the cycle steal mode, DMA transfer of the next channel is performed without returning the bus control to the MPU ($\overline{DRQH}$ remains "Low").

In changing from TSC mode to TSC mode, DMA transfer of the next channel is performed, after returning the bus control to MPU for one cycle.

In the case of HALT → HALT, it doesn't return the bus control to MPU in order not to increase the response time of DMA transfer and dead cycles of the system.

On the other hand, in the case of TSC → TSC mode, same mean cannot be applicable because MPU clock cannot remain stopped for a long time as in the case of HALT mode.

Both in the case of HALT → TSC mode and in the case of TSC → HALT mode, DMA operation timing is based on the same idea as the above two kinds of mode change. (In detail, see Fig. 25).

The timing in the case where the next byte is transfered without changing the channel is shown in Fig. 26. This is the case of HALT → HALT mode. In this case, the bus control returns to MPU, before the next byte is transfered. In the case of TSC → TSC mode, its timing is almost the same as than in Fig. 24, that is, after 1-byte transfer has completed, MPU executes the Instruction Cycle for one clock and then DMAC executes 1-byte transfer again.

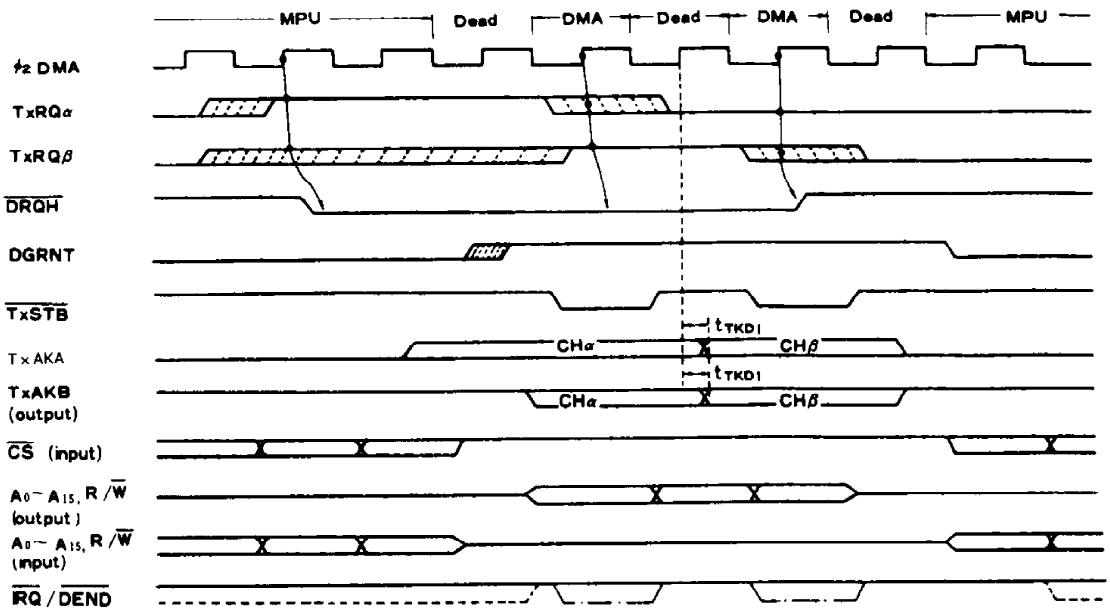


Figure 23 Channel Change (HALT Mode → HALT Mode)

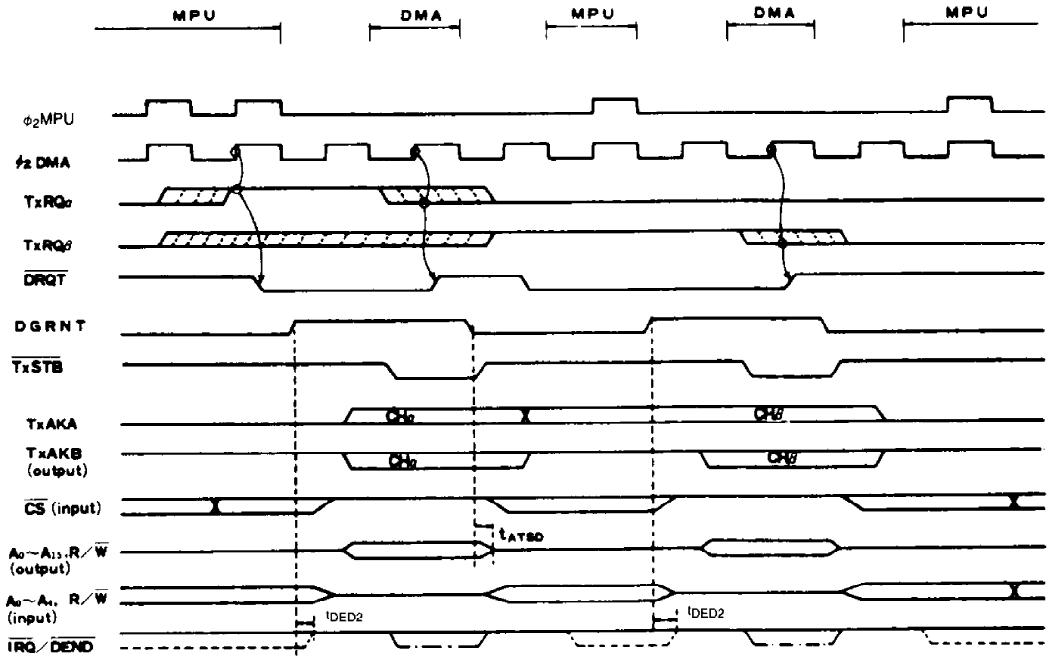


Figure 24 Channel Change (TSC Mode → TSC Mode)

5

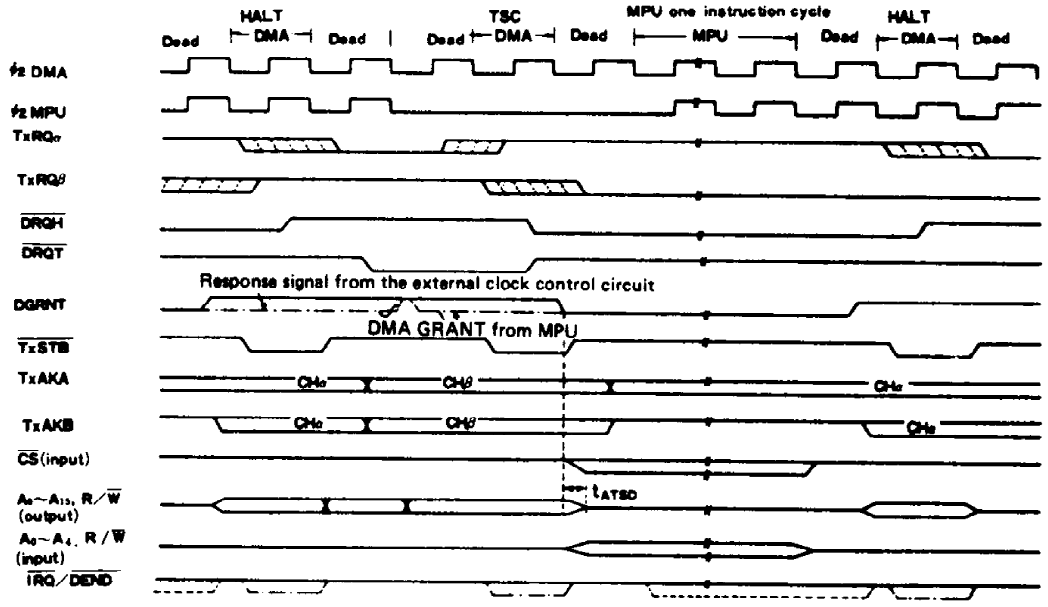
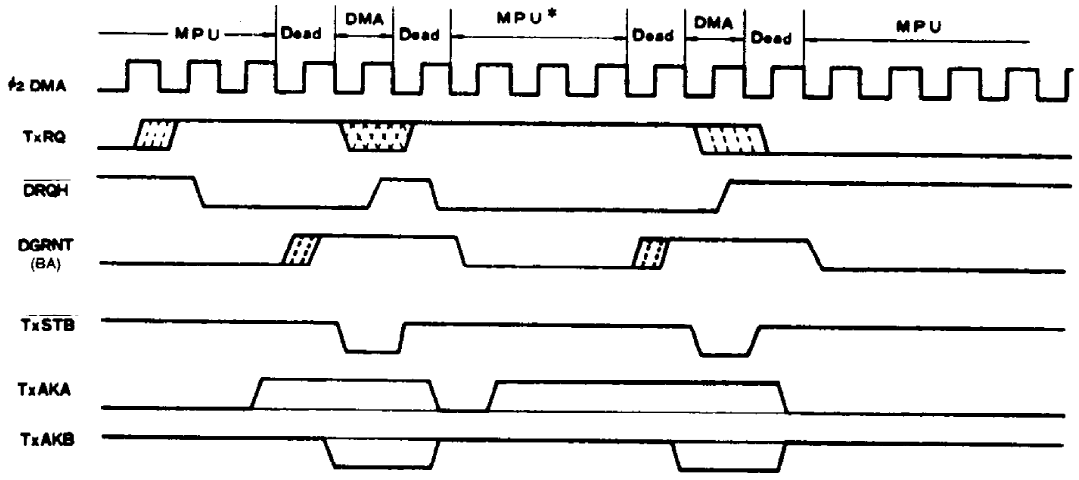


Figure 25 Channel Change (HALT Mode → TSC Mode → HALT Mode)



* Executing Period of One Instruction

Figure 26 Successive 2-byte Transfer of One Channel (HALT Cycle Steal Mode)
HALT → HALT (by one channel)

- **Status Flag**

DMAC has BUSY Flag, DEND Flag and ZERO Flag on each channel. The former two of these flags can be read out by MPU, but ZERO Flag cannot be read out. Set and reset timing of each flag are shown in Fig. 27.

BUSY/READY Flag

This flag is set to "1" when it accepts the first-byte TxRQ of its corresponding channel. After 1-block transfer has completed and BCR becomes "0", it is reset to "0". Therefore, while this flag is "1", that is, its corresponding channel is being used, the next block transfer cannot be performed.

Also this flag is cleared when corresponding TxRQ Enable Bit in the PCR becomes "0".

DEND Flag

This is the interrupt flag to indicate the end of DMA transfer of its corresponding channel. After 1-block transfer has completed and BCR becomes "0", this flag is set to "1". This flag is reset to "0" immediately after the Channel Control Register having this flag is read out.

ZERO Flag

This is the internal flag to indicate whether the data stored in the BCR is "0" or not (It cannot be read out).

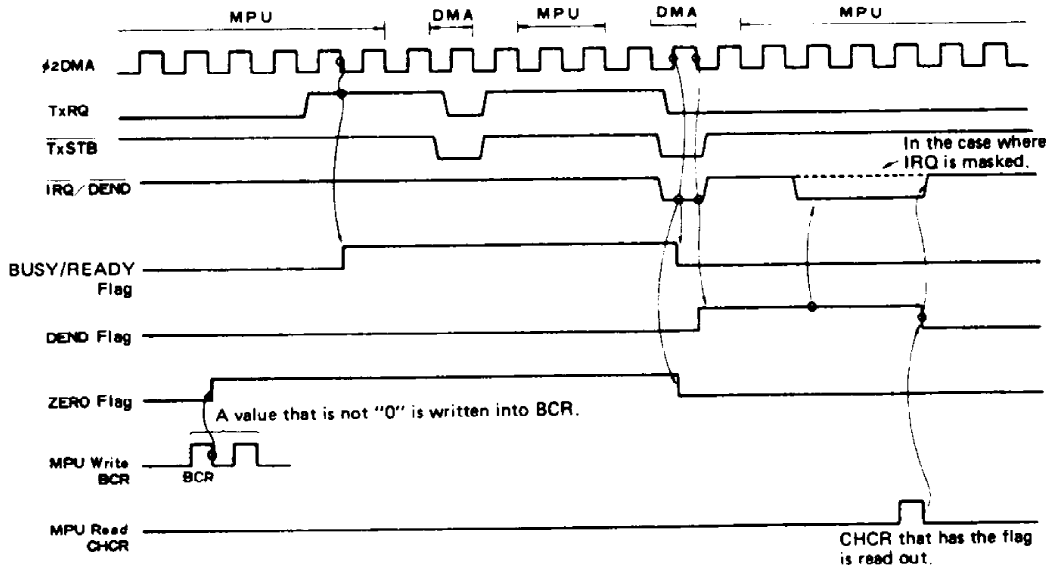


Figure 27 Timing of Status Flag (Suppose that BCR is 2 in the initial state)

When BCR is "0", ZERO Flag is "0". When BCR is not "0", it is "1".

In the reset state, this flag is "0". If data that is not "0" is written into BCR, this flag is set to "1". When BCR becomes "0" after 1-block data transfer has completed, or MPU writes "0" into BCR, this flag is reset to "0".

The function of ZERO Flag is to prohibit accepting TxRQ of its corresponding channel while this flag is "0" (that is, BCR is "0") (See Fig. 18). While ZERO Flag is "0", TxRQ is not accepted even if TxEN is "1". This function avoids an false operation even if "High" input is provided to TxRQ before the initialization of the register.

When $\overline{RES}$ pin goes to "Low", this flag becomes "0", but the number in BCR is not reset to "0". Therefore, the state of this flag and BCR are not the same. In this case new data should be written into BCR (Then ZERO Flag becomes "1").

- **DMA End Control**

Function of $\overline{IRQ}/\overline{DEND}$ Pin

DMAC has $\overline{IRQ}$ output and $\overline{DEND}$ output to perform DMA End Control. These are multiplexed outputs to $\overline{IRQ}/\overline{DEND}$ pin.

The function of $\overline{DEND}$ output is to inform I/O controller of the end of 1-block transfer. After 1-block transfer has been completed and BCR becomes "0", $\overline{DEND}$ output provides "Low" pulse whose cycle is one clock, being synchronous with the final 1-byte data transfer. 4 channels have only one $\overline{DEND}$ output in common, so each channel determines whether $\overline{DEND}$ output is its own output or not, combining with TxAK signal. When TxAK of the channel is "High" and $\overline{DEND}$ is "Low", it shows that the cycle is the last one of DMA (See Fig. 29 and 30).

The function of $\overline{IRQ}$ output is to inform MPU of the end of 1-block transfer by interrupting it. As shown in Fig. 28, $\overline{IRQ}$ output is logical AND-OR of the interrupt flag (DEND Flag) and IRQ Enable bit of each channel.

$\overline{IRQ}$ and $\overline{DEND}$ outputs are multiplexed. $\overline{IRQ}/\overline{DEND}$ pin is used as $\overline{DEND}$ output during DMAC cycle and $\overline{IRQ}$ output during MPU cycle. Moreover, DGRNT signal separates $\overline{DEND}$ and $\overline{IRQ}$ by its "High" or "Low". In detail, see Fig. 29 and Fig. 30.

5

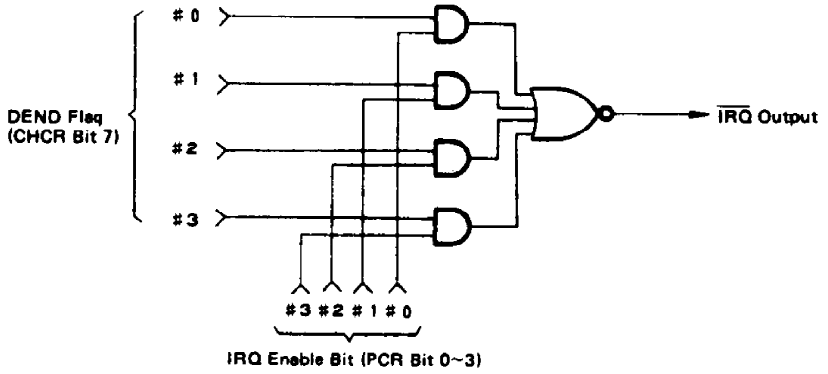


Figure 28 Logic of $\overline{IRQ}$ Output

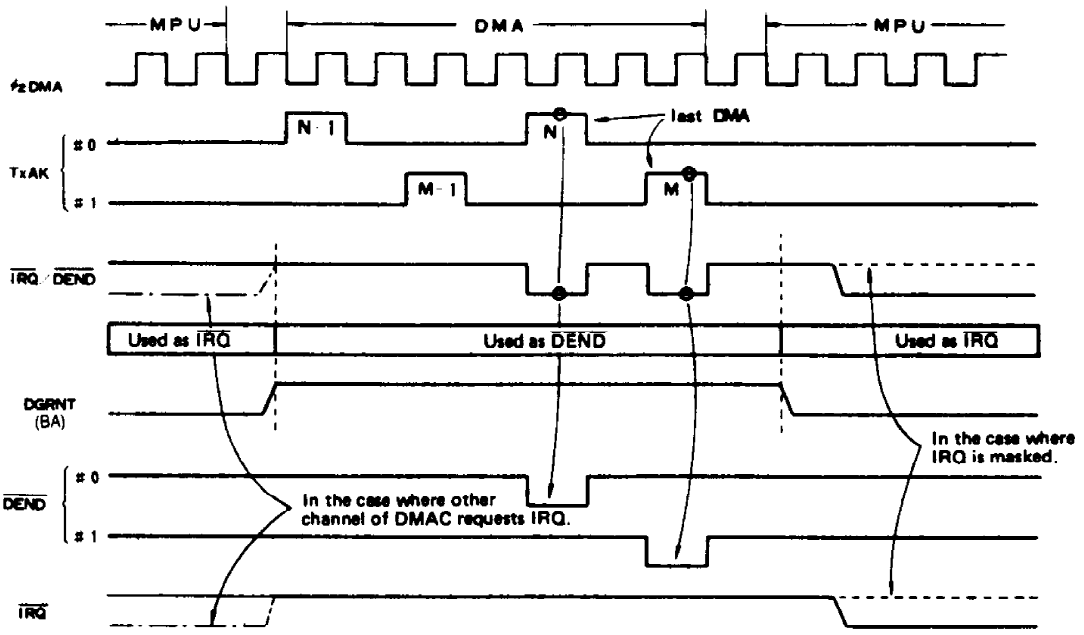


Figure 29 Timing of $\overline{IRQ}/\overline{DEND}$ Output

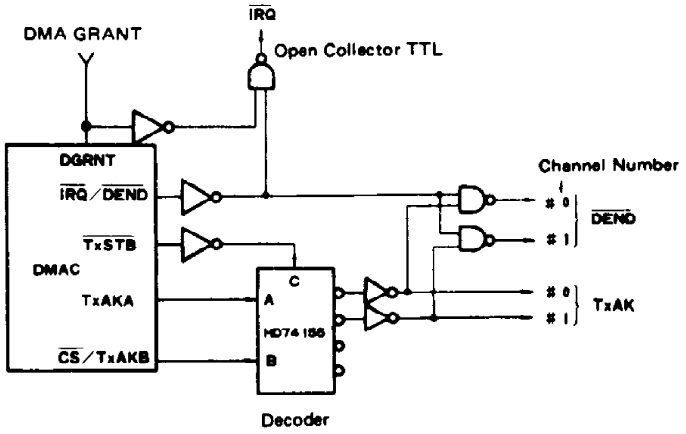


Figure 30 How to Use $\overline{IRQ}/\overline{DEND}$ Output Signal

Unusual DMA End

Following section describes how to terminate or change normal sequence of DMA transfer.

(1) When "0" is written into BCR
When "0" is written into BCR before it becomes "0", subsequent TxRQ are not accepted and this causes the termination of the DMA transfer since the internal ZERO Flag is reset to "0". In this case, note that $\overline{DEND}$ pulse is not provided.

(2) When "1" is written into BCR
When "1", instead of "0", is written into BCR, only the next TxRQ is accepted and 1-byte DMA transfer is performed. In this case, $\overline{DEND}$ pulse is provided, being synchronous with the last transfer.

(3) When another value is written into ADR & BCR during the transfer
When the data in ADR & BCR are changed during the transfer, the following transfer is performed according to the change of the data.

(4) When "0" is written into TxRQ Enable bit
When TxEN is reset to "0" during the transfer, this causes TxRQ comes not to be accepted and the transfer halts. But the state is different from that in the case (1), the number in BCR remains unchanged. Therefore, when TxEN is set to "1" again, the transfer is performed again.

(5) When $\overline{RES}$ pin is set to "Low"
When $\overline{RES}$ is provided during the transfer, the transfer stops.
Then all of the control registers and their internal flags are reset to "0". But the data in ADR & BCR are not reset.

(Supplement)

It is only in the cycle steal mode that DMAC registers such as BCR and ADR can be read or written during the transfer. In the burst mode, it is usually impossible (But special external circuits enable it).

- **Data Chain Function**

The data chain function of DMAC is to transfer the contents of ADR & BCR of Channel #3 to ADR & BCR of a specified channel automatically and renew the data of them after the channel has completed 1-block transfer.

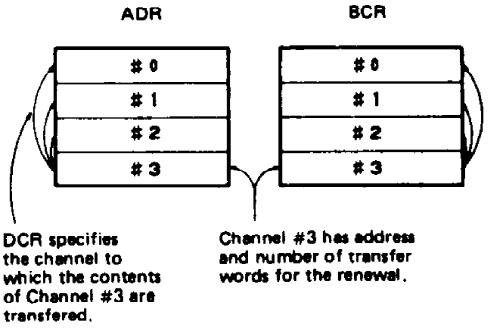


Figure 31 Data Chain Operation

Its detailed timing is shown in Fig. 32 and Fig. 33. As shown in these figures the contents of ADR & BCR of Channel #3 are transfered to the channel during the clock cycle next to the last one of 1-block transfer (which provides DEND pulse). Then $\overline{DRQH}$ or $\overline{DRQT}$ provides "Low" output for one more clock cycle than in the normal case. Therefore, MPU takes back the bus control again 1-clock later than in the normal case, that is, after the data renewal of the specified channel by the data chain from Channel #3.

In the TSC mode, the stretching period of clockϕ_1 is longer than in the normal case.

The contents of ADR & BCR of Channel #3 remain unchanged as long as new data are not written by MPU, even if the data chain is executed.

As for $\overline{DEND}$ output, DEND Flag and BUSY Flag in the case of data chain execution, they function in the same way as in the normal case. They provide $\overline{DEND}$ pulse everytime 1-block transfer has completed, and then DEND Flag is set to "1". Therefore, in the case where more than 3-block data chain is needed, DEND Flag is used for the execution. Its sequence is shown in Fig. 34. First, DEND Flag="1" that shows the end of the first-block data chain is read out. Next, the data of ADR & BCR for the third-block data chain need to be written into Channel #3, in parallel with the execution of the second-block data chain. (This data chain is feasible only in the cycle steal mode.)

5

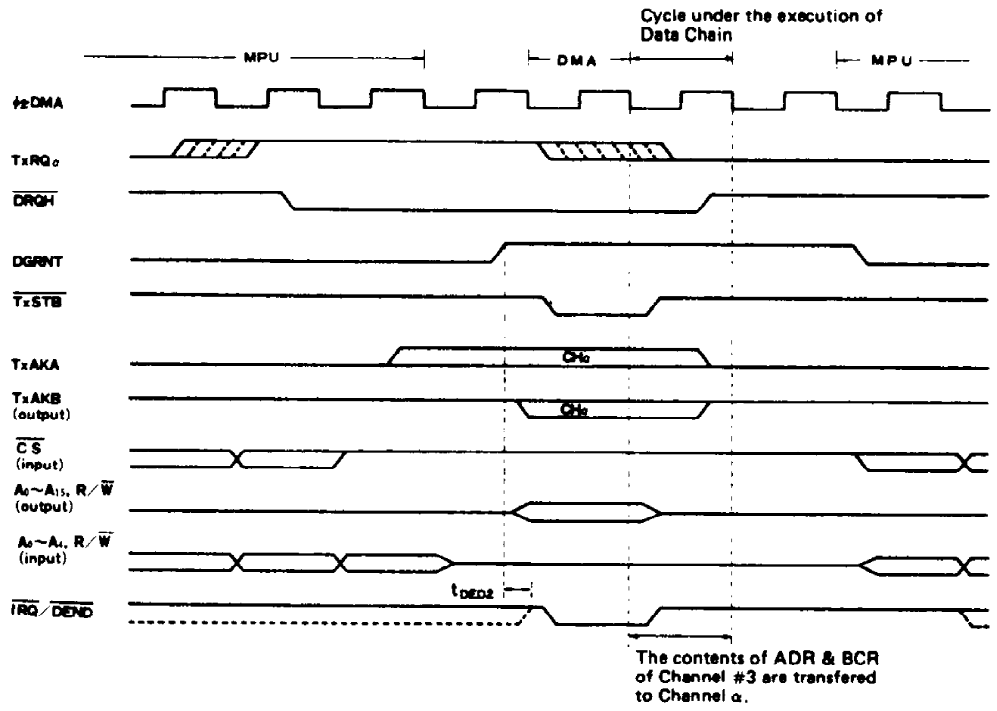


Figure 32 Data Chain Operation (HALT Mode)

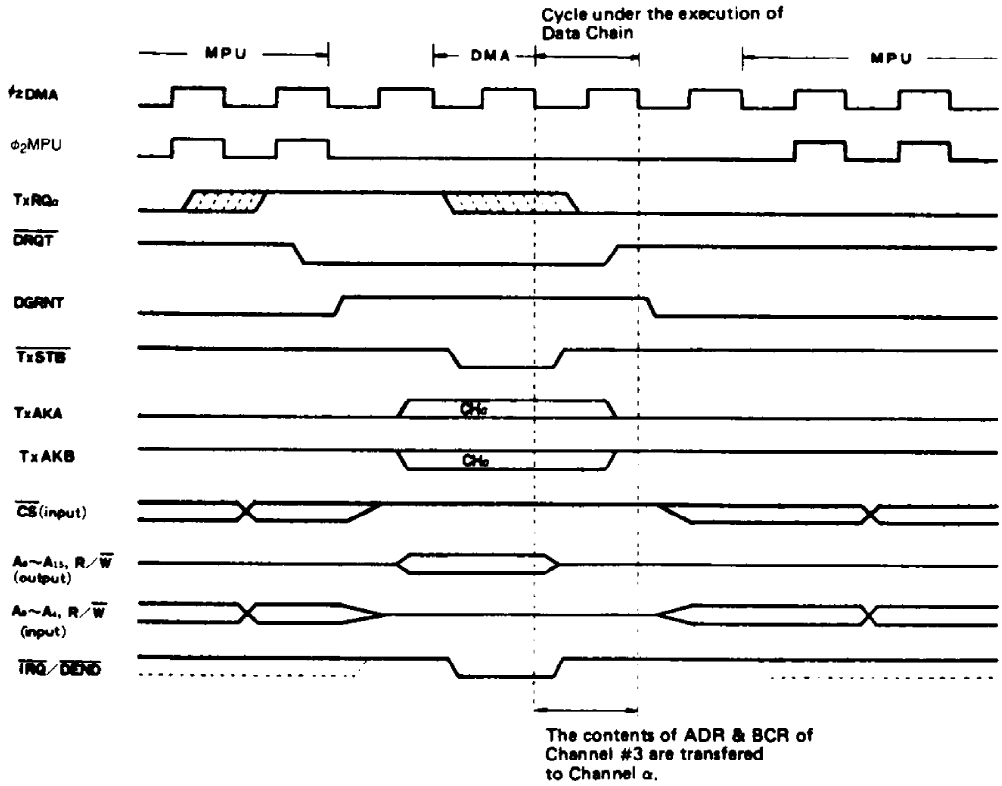


Figure 33 Data Chain Operation (TSC Mode)

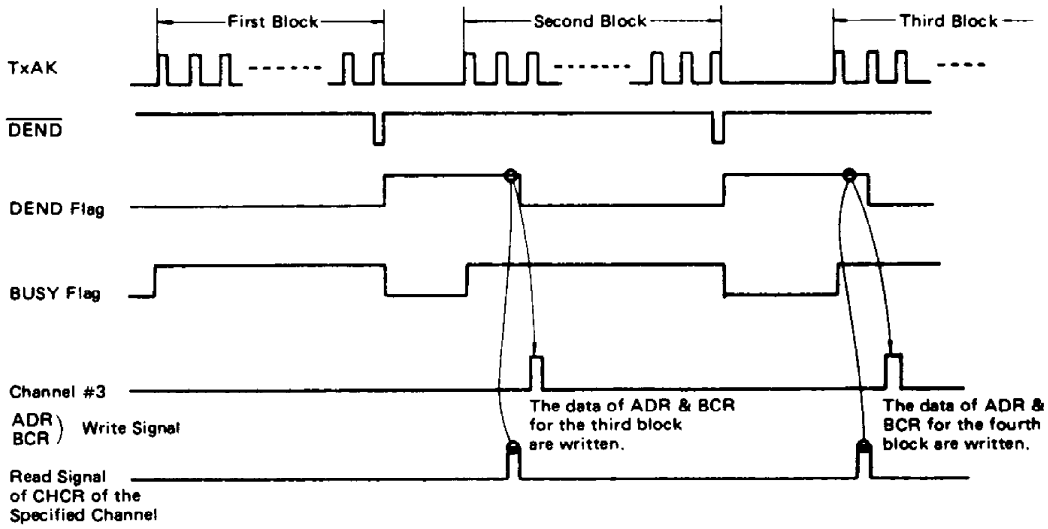


Figure 34 Sequence of More than 3-block Data Chain

■ DMAC PROGRAMMING

Preparation of a channel for a DMA transfer requires:

1) Load the starting address into the Address Register.
2) Load the number of bytes into the Byte Count Register.
3) Program the Channel Control Register for the transfer characteristics: direction (bit 0), mode (bits 1 and 2), and the address update (bit 3).

The channel is now configured. To enable the transfer request, set the appropriate enable bit (bits 0~3) of the Priority Control Register, as well as the Rotate Control bit.

If an interrupt on DMA End is desired, the enable bit (bits 0~3) of the Interrupt Control Register must be set.

If data chaining for the channel is necessary, it is programmed into the Data Chain Register and the appropriate data must be written into the Address and Byte Count Registers for channel #3.

Table 8 DMAC Programming Model

Register	Address (Hex)	Register Content							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Channel Control	1x*	DMA End Flag (DEND)	Busy/Ready Flag	Not Used	Not Used	Address Up/Down	TSC/ Halt	Burst/ Steal	Read/Write (R/W)
Priority Control	14	Rotate Control	Not Used	Not Used	Not Used	TxRQ Enable #3 (TxEN3)	TxRQ Enable #2 (TxEN2)	TxRQ Enable #1 (TxEN1)	TxRQ Enable #0 (TxEN0)
Interrupt Control	15	IRQ Flag	Not Used	Not Used	Not Used	IRQ Enable #3 (IE3)	IRQ Enable #2 (IE2)	IRQ Enable #1 (IE1)	IRQ Enable #0 (IE0)
Data Chain	16	Not Used	Not Used	Not Used	Not Used	Two/Four Channel Select (2/4)	Data Chain Channel Select B	Data Chain Channel Select A	Data Chain Enable

* The x represents the binary equivalent of the channel desired.

A comparison of the response times and maximum transfer rates is shown in Table 9. The data are shown for a system clock rate of 1 MHz.

The two 8-bit bytes that form the registers in Table 10 are placed in consecutive memory locations, making it very easy to use the MPU index register in programming them.

Fig. 38 shows an example of its minimum structure (1 channel, HALT mode, combination with FDC). Fig. 39 shows an example of its maximum structure. (but only one DMAC is used.)

5

Table 9 Maximum Transfer Speed & Response Time of the DMAC when $t_{cyc\phi}$ equals 1 μsec.

Mode		Maximum Transfer Speed (μsec/byte)	Response Time (μsec)	
			maximum	minimum
HALT Cycle Steal		(executing time of one instruction) + 3	(executing time of one instruction) +3.5 − t_{TQH1}	3.5 + t_{TQS1}
HALT Burst	first byte	1		
	since second byte		2 − t_{TQH2}	1 + t_{TQS2}
TSC Cycle Steal		4	3.5 − t_{TQH1}	2.5 + t_{TQH1}

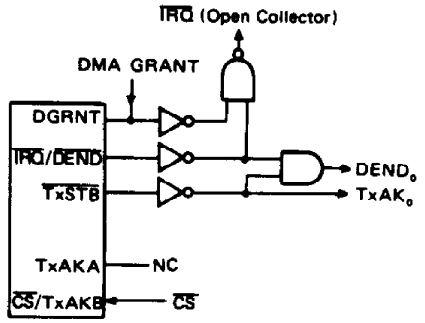


Figure 35 One Channel

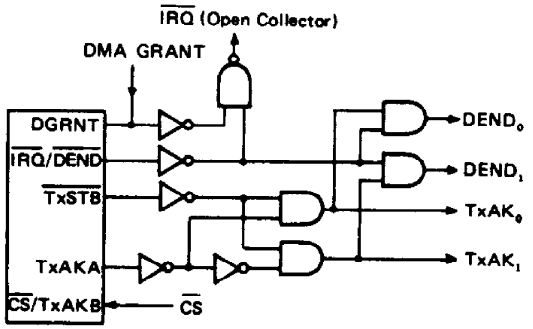


Figure 36 Two Channel

Table 10 Address and Byte Count Registers

Register	Channel	Address (Hex)
Address High	0	0
Address Low	0	1
Byte Count High	0	2
Byte Count Low	0	3
Address High	1	4
Address Low	1	5
Byte Count High	1	6
Byte Count Low	1	7
Address High	2	8
Address Low	2	9
Byte Count High	2	A
Byte Count Low	2	B
Address High	3	C
Address Low	3	D
Byte Count High	3	E
Byte Count Low	3	F

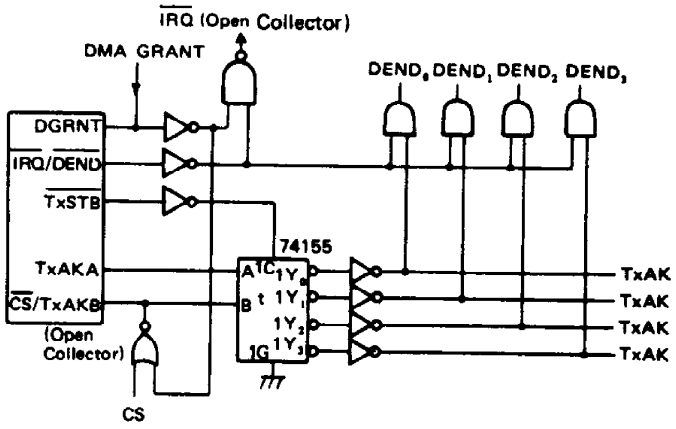


Figure 37 Four-Channel

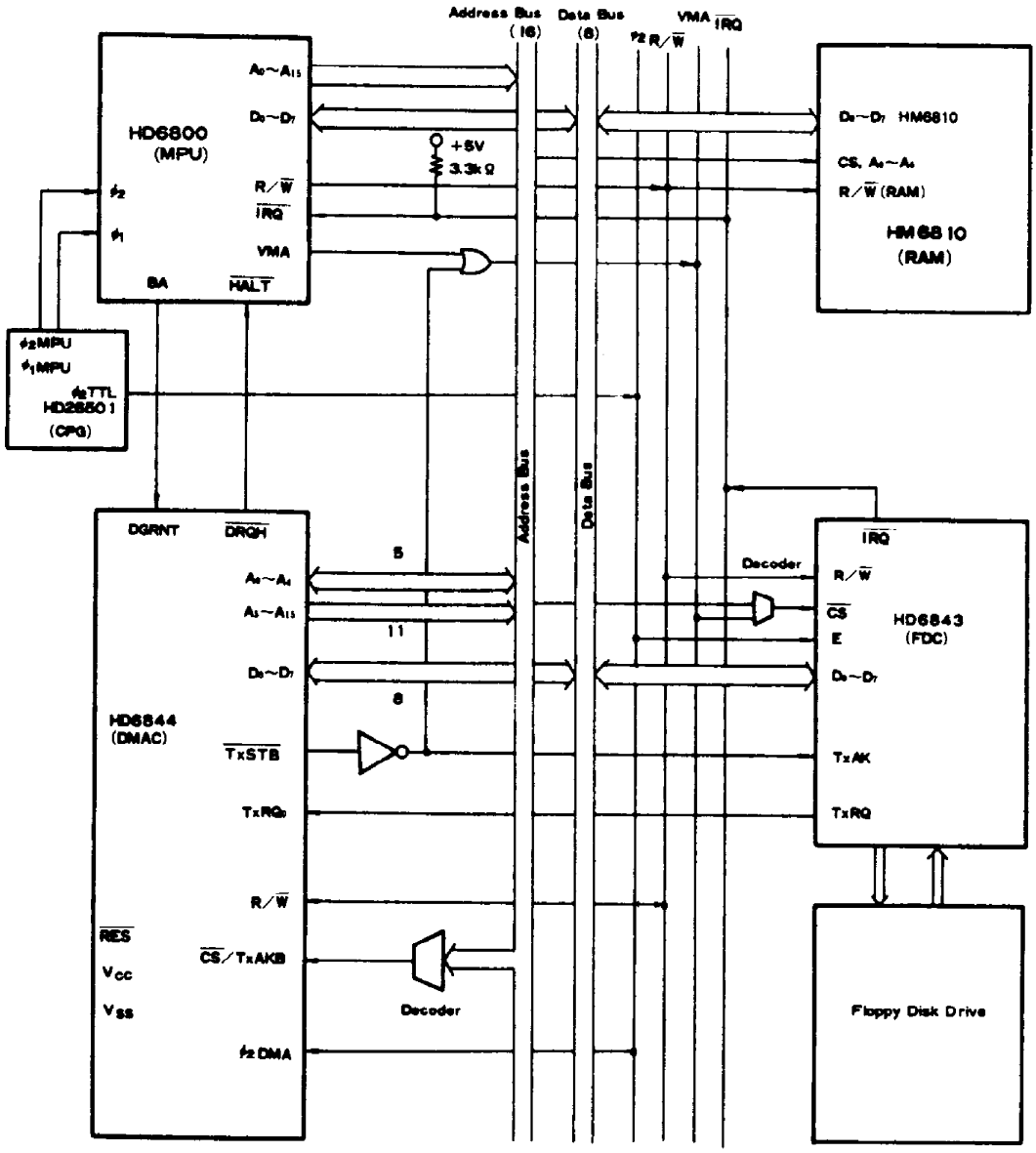


Figure 38 Example of DMA System Structure (1) (minimum)

5

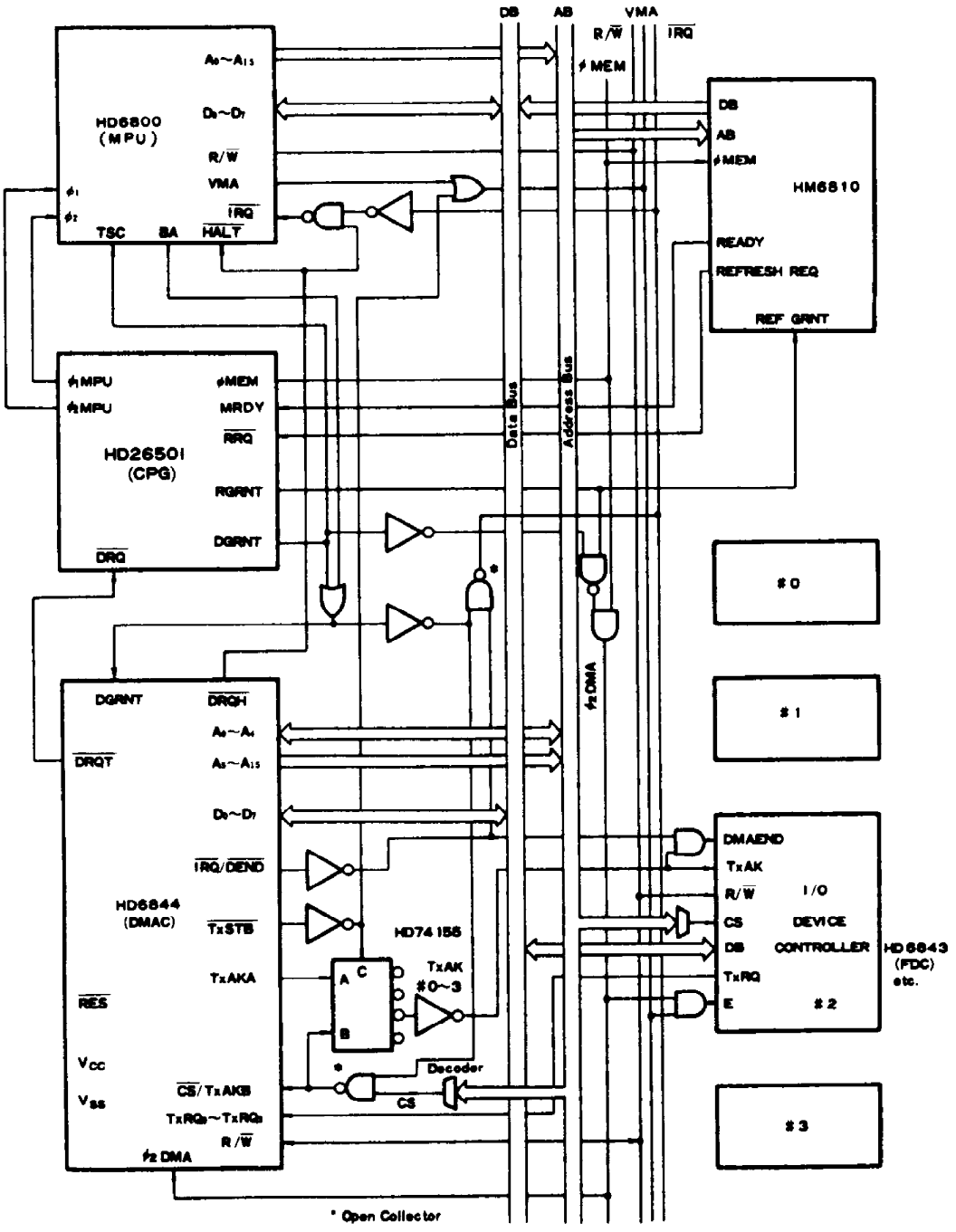
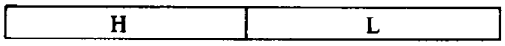


Figure 39 Example of DMA System Structure (2) (maximum)

■ APPENDIX

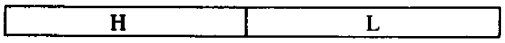
Contents of the DMAC Registers

(1) ADR0 ~ ADR3 (Address Register) (1 ADR on each channel)



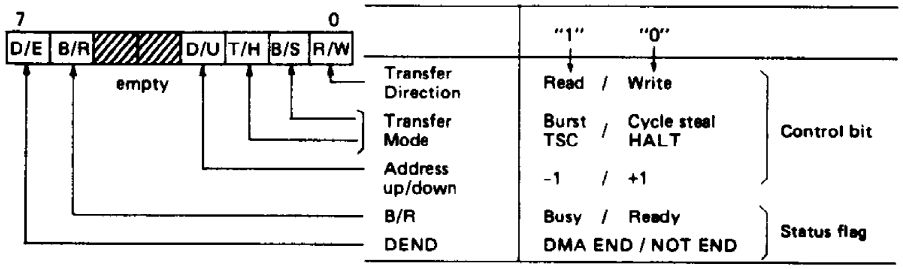
16 bit x 4

(2) BCR0 ~ BCR3 (Byte Count Register) (1 BCR on each channel)

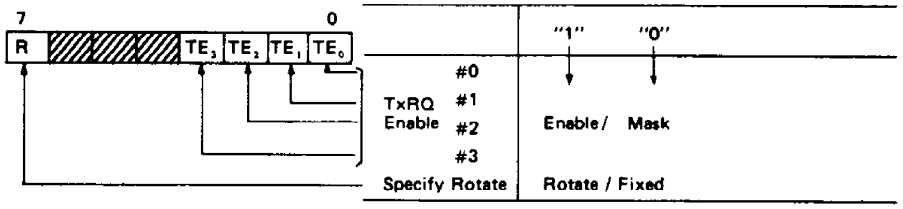


16 bit x 4

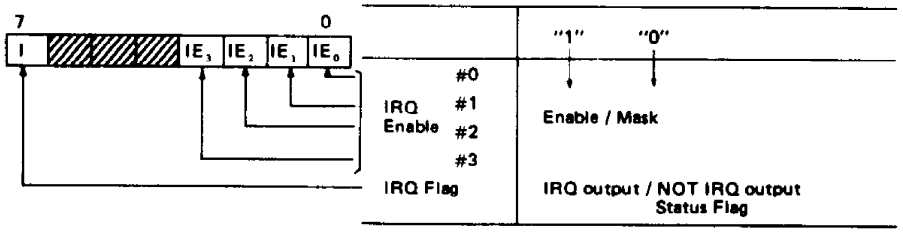
(3) CHCR0 ~ CHCR3 (Channel Control Register) (1 CHCR on each channel) (6 bit x 4)



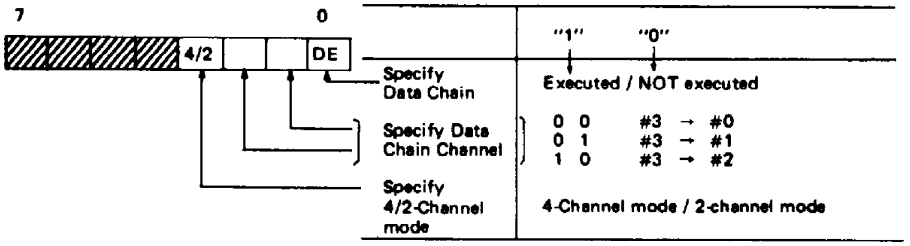
(4) PCR (Priority Control Register) (5 bit x 1)



(5) ICR (Interrupt Control Register) (5 bit x 1)



(6) DCR (Data Chain Control Register) (4 bit x1)



5