Description

The μPD71071 is a high-speed, high-performance direct memory access (DMA) controller that provides high-speed data transfers between peripheral devices and memory. A programmable bus width allows bidirectional data transfer in both 8- and 16-bit systems. In addition, the μPD71071 uses CMOS technology to reduce power consumption.

The μPD71071 can perform a variety of transfer functions including byte/word, memory-to-memory, and transfers between memory and I/O. The μPD71071 also utilizes single, demand, and block mode transfers; release and bus hold modes; and normal and compressed timing.

Features

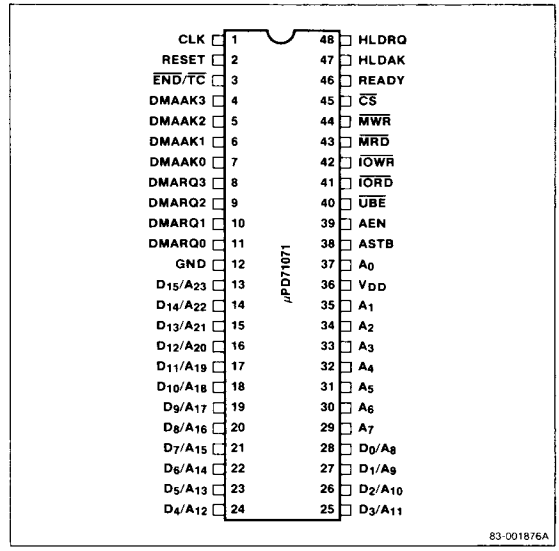
- ☐ Four independent DMA channels
- ☐ 16M-byte addressing
- ☐ 64K-byte/word transfer count
- ☐ 8- or 16-bit programmable data bus width
- ☐ Enable/disable of individual DMA requests
- ☐ Software DMA requests
- ☐ Enable/disable of autoinitialize
- ☐ Address increment/decrement
- ☐ Fixed/rotational DMA channel priority
- ☐ Terminal count output signal
- ☐ Forced transfer termination input
- ☐ Cascade capability
- ☐ Programmable DMA request and acknowledge signal polarities
- ☐ High performance: transfers to 5.33 Mbytes/s
- ☐ μPD70108/70116-compatible
- ☐ CMOS technology
- ☐ Low-power standby mode
- ☐ Single power supply, 5 V ±10%
- ☐ Industrial temperature range, −40 to +85 °C
- ☐ 10 MHz operation

Ordering Information

Part Number	Package
μPD71071C-10	48-pin plastic DIP
L-10	52-pin PLCC

Pin Configurations

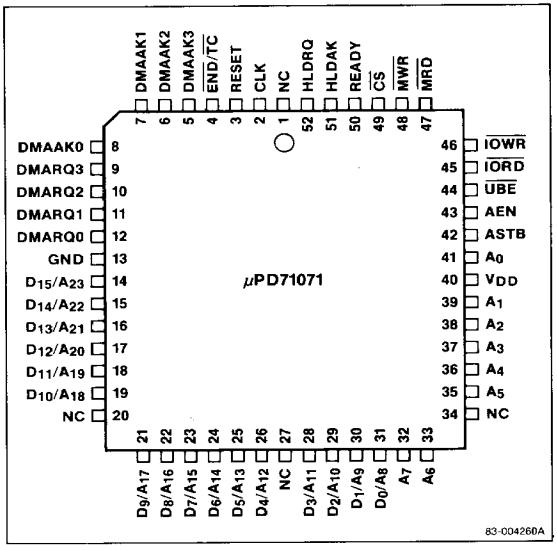
48-Pin Plastic DIP



5g

Pin Configurations (cont)

52-Pin Plastic Leaded Chip Carrier (PLCC)



Pin Functions

CLK [Clock]

CLK controls the internal operation and data transfer speed of the μPD71071.

RESET [Reset]

RESET initializes the controller's internal registers and leaves the controller in the idle cycle (CPU controls the bus). Active high.

$\overline{END}/\overline{TC}$ [End/Terminal Count]

This is a bidirectional pin. The $\overline{END}$ input is used to terminate the current DMA transfer. $\overline{TC}$ indicates the designated cycles of the DMA count transfer have finished. $\overline{END}/\overline{TC}$ is open drain and requires an external pull-up resistor. Active low.

DMAAK3-DMAAK0 [DMA Acknowledge]

DMAAK3-DMAAK0 indicates to peripheral devices that DMA service has been granted. DMAAK3-DMAAK0 respond respectively to DMA channels 3-0 and the polarities are user programmable.

Pin Identification

Symbol	Function
A_{23}-A_8/ D_{15}-D_0	Bidirectional address/data bus
IC	Internally connected; leave open
A_7-A_4	Address bus output
NC	Not connected
A_3-A_0	Bidirectional address bus
V_{DD}	Power supply
ASTB	Address strobe output
AEN	Address enable output
$\overline{UBE}$	Upper byte enable input/output
$\overline{IORD}$	I/O read input/output
$\overline{IOWR}$	I/O write input/output
$\overline{MRD}$	Memory read output
$\overline{MWR}$	Memory write output
$\overline{CS}$	Chip select input
READY	Ready input
HLDAK	Hold acknowledge input
HLDRQ	Hold request output
CLK	Clock input
RESET	Reset input
$\overline{END}/\overline{TC}$	End DMA transfer input/terminal count output
DMAAK3-DMAAK0	DMA acknowledge output
DMARQ3-DMARQ0	DMA request input
GND	Ground

DMARQ3-DMARQ0 [DMA Request]

DMARQ3-DMARQ0 accept DMA service requests from peripheral devices. DMARQ3-DMARQ0 respond respectively to DMA channels 3-0 and the polarities are user programmable. DMARQ must remain asserted until DMAAK is asserted.

GND [Ground]

GND connects to the power supply ground terminal.

A_{23}-A_8/D_{15}-D_0 [Address/Data Bus]

A_{23}-A_8/D_{15}-D_0 function as a 16-bit, multiplexed address/data bus when the μPD71071 is in the 16-bit data mode. In the 8-bit data mode, A_{23}-A_{16} (pins 13-20) become address bits only and A_{15}-A_8/D_7-D_0 (pins 21-28) remain an 8-bit multiplexed address/data bus. A_{23}-A_8/D_{15}-D_0 are three-state.

A_7-A_4, A_3-A_0 [Address Bus]

A_7-A_4, A_3-A_0 function as the lower eight bits of the address bus. A_7-A_4 output memory addresses during the DMA cycle and become high impedance in the idle cycle. A_3-A_0 function as the lower four bits of the address bus. In the idle cycle, A_3-A_0 become address inputs to select internal registers for the CPU to read or write. In the DMA cycle, A_3-A_0 output memory addresses.

V_{DD} [Power Supply]

V_{DD} connects to the +5-V power supply.

ASTB [Address Strobe]

ASTB latches address A_{23}-A_8 (16-bit mode)/A_{15}-A_8 (8-bit mode) from the address/data bus into an external address latch at the falling edge of ASTB during a DMA cycle. Active high.

AEN [Address Enable]

AEN enables the output of an external latch that holds DMA addresses. AEN becomes high during the DMA cycle.

$\overline{UBE}$ [Upper Byte Enable]

$\overline{UBE}$ indicates the upper byte of the data bus is valid during 16-bit mode. In the idle cycle during data transfer, the μPD71071 acknowledges data on D_{15}-D_8 when $\overline{UBE}$ is asserted. During a DMA cycle, UBE goes low to signify the presence of valid data on D_{15}-D_8. $\overline{UBE}$ has no meaning in 8-bit mode and becomes high impedance in the idle cycle and high level in the DMA cycle. Three-state, active low.

$\overline{IORD}$ [I/O Read]

In the idle cycle, $\overline{IORD}$ inputs a read signal from the CPU. In the DMA cycle, $\overline{IORD}$ outputs a read signal to an I/O device. Three-state, active low.

$\overline{IOWR}$ [I/O Write]

In the idle cycle, $\overline{IOWR}$ inputs a write signal from the CPU. In the DMA cycle, $\overline{IOWR}$ outputs a write signal to an I/O device. Three-state, active low.

$\overline{MRD}$ [Memory Read]

During the DMA cycle, $\overline{MRD}$ outputs a read signal to memory. $\overline{MRD}$ is high impedance during the idle cycle. Three-state, active low.

$\overline{MWR}$ [Memory Write]

During the DMA cycle, $\overline{MWR}$ outputs a write signal to memory. $\overline{MWR}$ is high impedance during the idle cycle. Three-state, active low.

$\overline{CS}$ [Chip Select]

During the idle cycle, $\overline{CS}$ selects the μPD71071 as an I/O device. Active low.

READY [Ready]

During a DMA operation, READY indicates that a data transfer for one cycle has been completed and may be terminated. To meet the requirements of low-speed I/O devices or memory, READY may be negated to insert wait states to extend the bus cycle until READY is again asserted.

HLDAK [Hold Acknowledge]

When active, HLDAK indicates that the CPU has granted the μPD71071 the use of the system bus. Active high.

HLDRQ [Hold Request]

HLDRQ outputs a bus hold request to the CPU. Active high.

Block Diagram Description

The μPD71071 has the following functional units.

- Bus control unit
- DMA control unit
- Address registers
- Address incrementer/decrementer
- Count registers
- Count decrementer
- Control registers

5g

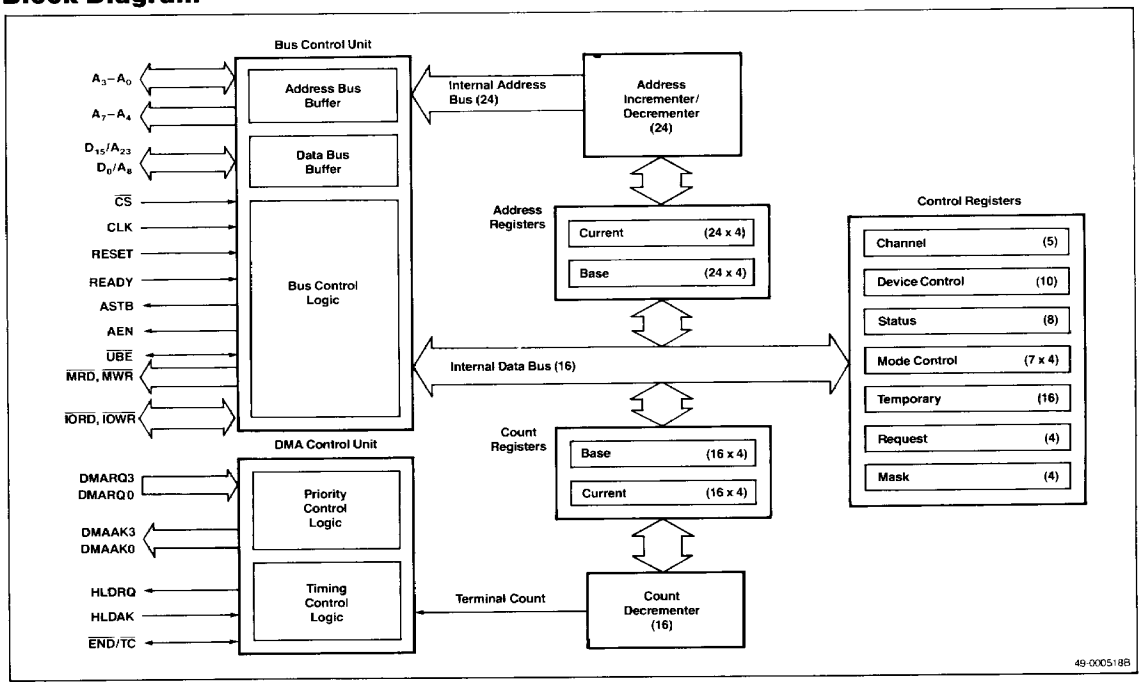
Bus Control Unit

The bus control unit consists of the address and data buffers, and bus control logic. The bus control unit generates and receives signals that control addresses and data on the internal address and data buses.

DMA Control Unit

The DMA control unit contains the priority and timing control logic. The priority control logic determines the priority level of DMA requests and arbitrates the use of the bus in accordance with this priority level. The DMA control unit also provides internal timing and controls DMA operations.

Block Diagram



Address Registers

Each of the four DMA channels has one 24-bit base address register and one 24-bit current address register. The base address register holds a value determined by the CPU and transfers this value to the current address register during autoinitialization (address and count are automatically initialized). The channel's current address register is incremented/decremented for each transfer and always contains the address of the data to be transferred next.

Address Incrementer/Decrementer

The address incrementer/decrementer updates the contents of the current address register whenever a DMA transfer completes.

Count Registers

Each of the four DMA channels has one 16-bit base count register and one 16-bit current count register. The base count register holds a value written by the CPU and transfers the value to the current count register during autoinitialization. A channel's current count register is decremented for each transfer and generates a terminal count when the count register is decremented to FFFFH.

Note: The number of DMA transfer cycles is actually the value of the current count register + 1. Therefore, when programming the count register, specify the number of DMA transfers minus one.

Count Decrementer

The count decrementer decrements the contents of the current count register by one when each DMA transfer cycle ends.

Control Registers

The μPD71071 contains the following control registers.

- Channel
- Device
- Status
- Mode
- Temporary
- Request
- Mask

These registers control bus mode, pin active levels, DMA operation mode, mask bits, and other μPD71071 operating functions.

Absolute Maximum Ratings

Power supply voltage, V_{DD}	−0.5 to +7.0 V
Input voltage, V_I	−0.5 to V_{DD} + 0.3 V
Output voltage, V_O	−0.5 to V_{DD} + 0.3 V
Operating temperature, T_{OPT}	−40 to +85 °C
Storage temperature, T_{STG}	−65 to +150 °C

Comment: Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of this specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Capacitance

T_A = 25 °C

Parameter	Symbol	Limits Typ	Limits Max	Unit	Test Conditions
Output capacitance	C_O	4	8	pF	f_c = 1.0 MHz unmeasured pins returned to 0 V
Input capacitance	C_I	8	15	pF	
I/O capacitance	C_{IO}	10	18	pF	

DC Characteristics

T_A = −40 to +85 °C, V_{DD} = 5 V ±10%

Parameter	Symbol	Limits Min	Limits Typ	Limits Max	Unit	Test Conditions
Input high voltage	V_{IH}	3.3		V_{DD} + 0.3	V	CLK input pin
		2.2		V_{DD} + 0.3	V	Other inputs
Input low voltage	V_{IL}	−0.5		0.8	V	
Output high voltage	V_{OH}	0.7 V_{DD}			V	I_{OH} = −400 μA
Output low voltage	V_{OL}			0.4	V	I_{OL} = 2.5 mA; 4.5 mA (TC)
Input leakage current	I_{LI}			±10	μA	0 V ≤ V_I ≤ V_{DD}
Output leakage current	I_{LO}			±10	μA	0 V ≤ V_O ≤ V_{DD}
Supply current (dynamic)	I_{DD1}		15	30	mA	
Supply current (stable)	I_{DD2}		10		μA	Inputs stable outputs open
Supply current (static)	I_{DD2}		10		μA	

5g

AC Characteristics

$T_A = -40$ to $+85°C$, $V_{DD} = 5\ V \pm 10\%$

Parameter	Symbol	Min	Max	Unit	Test Conditions
DMA Mode					
Clock cycle	t_{CYK}	100		ns	
Clock pulse width high	t_{KKH}	39		ns	
Clock pulse width low	t_{KKL}	49		ns	
Clock rise time	t_{KR}		10	ns	1.5 V → 3.0 V
Clock fall time	t_{KF}		10	ns	3.0 V → 1.5 V
Input rise time	t_{IR}		20	ns	
Input fall time	t_{IF}		12	ns	
Output rise time	t_{OR}		20	ns	
Output fall time	t_{OF}		12	ns	
DMARQ setup time to CLK high	t_{SDQ}	20		ns	S1, S0, S3, SW, S4w
HLDRQ high delay from CLK low	t_{DHQH}	5	70	ns	S1, S4w
HLDRQ low delay from CLK low	t_{DHQL}	5	70	ns	S1, S0, S4w
HLDRQ low level period	t_{HQHQL}	$2t_{CYK} - 50$		ns	S4w
HLDAK high setup time to CLK low	t_{SHA}	20		ns	S0, S4, S4w
AEN high delay from CLK low	t_{DAEH}	5	70	ns	S1, S2
AEN low delay time from CLK low	t_{DAEL}	5	70	ns	S1, S4w
ASTB high delay time from CLK low	t_{DSTH}	5	70	ns	S1
ASTB low delay time from CLK high	t_{DSTL}	5	70	ns	S1
ASTB high level period	t_{STSTH}	$t_{KKL} - 15$		ns	
ADR/$\overline{UBE}$/$\overline{RD}$/$\overline{WR}$ active delay from CLK low (Note 1)	t_{DA}	5	80	ns	S1, S2
ADR/$\overline{UBE}$/$\overline{RD}$/$\overline{WR}$ float time from CLK low	t_{FA}	0	70	ns	S1, S4w
ADR setup time to ASTB low	t_{SAST}	$t_{KKL} - 40$		ns	
ADR hold time to ASTB low	t_{HSTA}	$t_{KKH} - 20$		ns	

AC Characteristics (cont)

Parameter	Symbol	Min	Max	Unit	Test Conditions
DMA Mode (cont)					
ADR/$\overline{UBE}$ off delay time from CLK low	t_{DAF}	0	70	ns	S1, S2
$\overline{RD}$ low delay time from ADR float	t_{DAR}	−10		ns	
Input data delay time from $\overline{MRD}$ low	t_{DMRID}		$2t_{CYK} - 80$	ns	S12
Input data hold time from MRD high	t_{HMRID}	0		ns	S14
Output data delay time from CLK low	t_{DOD}	10	80	ns	S22
Output data hold time from CLK high	t_{HOD}	10		ns	S24
Output data hold time from $\overline{MWR}$ high	t_{HMWOD}	$t_{KKL} - 35$		ns	
$\overline{RD}$ low delay time from CLK high	t_{DKHR}			ns	S2 compressed timing
$\overline{RD}$ low level period	t_{RRL1}	$2t_{CYK} - 30$		ns	Normal timing
	t_{RRL2}	$t_{CYK} + t_{KKH} - 30$		ns	Compressed timing
$\overline{RD}$ high delay time from CLK low	t_{DRH}	10	70	ns	S4
ADR delay time from $\overline{RD}$ high	t_{DRA}	$t_{CYK} - 30$		ns	
$\overline{WR}$ low delay time from CLK low	t_{DWL1}	5	50	ns	S3 normal write
$\overline{WR}$ low delay time from CLK low	t_{DWL2}	5	50	ns	S2 extended write, normal timing
$\overline{WR}$ low delay time from CLK high	t_{DWL3}	5	50	ns	S2 extended write, compressed timing
$\overline{WR}$ low level period	t_{WWL1}	$t_{CYK} - 30$		ns	Normal write
	t_{WWL2}	$2t_{CYK} - 30$		ns	Extended write, normal timing
	t_{WWL3}	$t_{CYK} + t_{KKH} - 30$		ns	Extended write, compressed timing
$\overline{WR}$ high delay from CLK low	t_{DWH}	5	50	ns	S4
$\overline{RD}$ low delay time from CLK low	t_{DKLR}	5	50	ns	S2 normal timing
$\overline{RD}$, $\overline{WR}$ low delay from DMAAK active	t_{DDARW}	0		ns	S1, S2
$\overline{RD}$ high delay time from $\overline{WR}$ high	t_{DWHRH}	5		ns	
DMAAK delay time from CLK high	t_{DKHDA}	5	70	ns	S1 I/O memory timing
DMAAK delay time from CLK low	t_{DKLDA}	10	90	ns	S1 cascade mode
DMAAK inactive delay time from CLK high	t_{DDAI1}	5		ns	S4

5g

AC Characteristics (cont)

Parameter	Symbol	Min	Max	Unit	Test Conditions
DMA Mode (cont)					
DMAAK inactive delay time from	t_{DDAI2}	5	$t_{KKL} + 70$	ns	S4 cascade mode, HLDAK low HLDAK low in S4
	t_{DDAI3}		$4t_{KKL} + 70$	ns	S4 cascade mode, HLDAK low except in S4
DMAAK active level period	t_{DADA}			ns	Cascade mode
$\overline{TC}$ low delay time from CLK high	t_{DTCL}	5	70	ns	S3
$\overline{TC}$ off delay time from CLK high	t_{DTCF}		30	ns	S4
$\overline{TC}$ high delay time from CLK high	t_{DTCH}		$t_{KKH} + t_{CYK} - 10$	ns	0 to 2.2 V (Note 2)
$\overline{TC}$ low level period	t_{TCTCL}	$t_{CYK} - 15$		ns	
$\overline{END}$ low setup time to CLK high	t_{SED}	20		ns	S2
$\overline{END}$ low level period	t_{EDEDL}	50		ns	
READY setup time to CLK high	t_{SRY}	20		ns	S3, SW
READY hold time from CLK high	t_{HRY}	10		ns	S3, SW
Programming Mode and RESET					
$\overline{IOWR}$ low level period	t_{IWIWL}	80		ns	
$\overline{CS}$ low setup time to $\overline{IOWR}$ high	t_{SCSIW}	80		ns	
$\overline{CS}$ hold time from IOWR high	t_{HIWCS}	0		ns	
ADR/$\overline{UBE}$ setup time to $\overline{IOWR}$ high	t_{SAIW}	80		ns	
ADR/$\overline{UBE}$ hold time from $\overline{IOWR}$ high	t_{HIWA}	0		ns	
Input data setup time to $\overline{IOWR}$ high	t_{SIDIW}	80		ns	
Input data hold time from $\overline{IOWR}$ high	t_{HIWID}	0		ns	
$\overline{IORD}$ low level period	t_{IRIRL}	120		ns	
ADR/$\overline{CS}$ setup time to $\overline{IORD}$ low	t_{SAIR}	20		ns	
ADR/$\overline{CS}$ hold time from $\overline{IORD}$ high	t_{HIRA}	0		ns	
Output data delay time from $\overline{IORD}$ low	t_{DIROD}	10	100	ns	
Output data float time from $\overline{IORD}$ high	t_{FIROD}		80	ns	
RESET high level period	t_{RESET}	$2t_{CYK}$		ns	
V_{DD} setup time to RESET low	t_{SVDD}	500		ns	
$\overline{IOWR}$/$\overline{IORD}$ wait time from RESET low	t_{SYIWR}	$2t_{CYK}$		ns	RESET low to first read/write
$\overline{IOWR}$/$\overline{IORD}$ recovery time	t_{RVIWR}	160		ns	

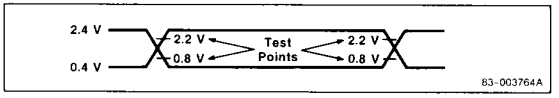
Notes:

(1) $\overline{RD}$/$\overline{WR}$ refers to $\overline{IORD}$ or $\overline{MRD}$ and $\overline{IOWR}$ or $\overline{MWR}$, respectively.

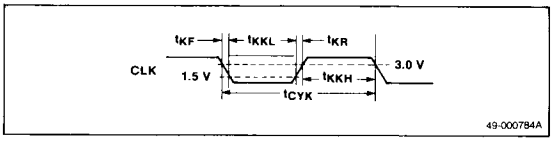
(2) For $\overline{END}$/$\overline{TC}$, output load capacitance = 75 pF maximum. To meet the t_{DTCH} parameter use a 2.2-kΩ pull-up resistor with a load capacitance of 75 pF. For other than $\overline{END}$/$\overline{TC}$, output load capacitance = 100 pF maximum.

Timing Waveforms

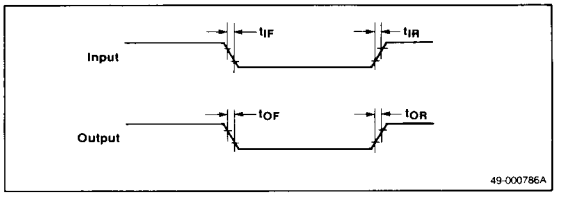
Timing Measurement Points



Clock Timing



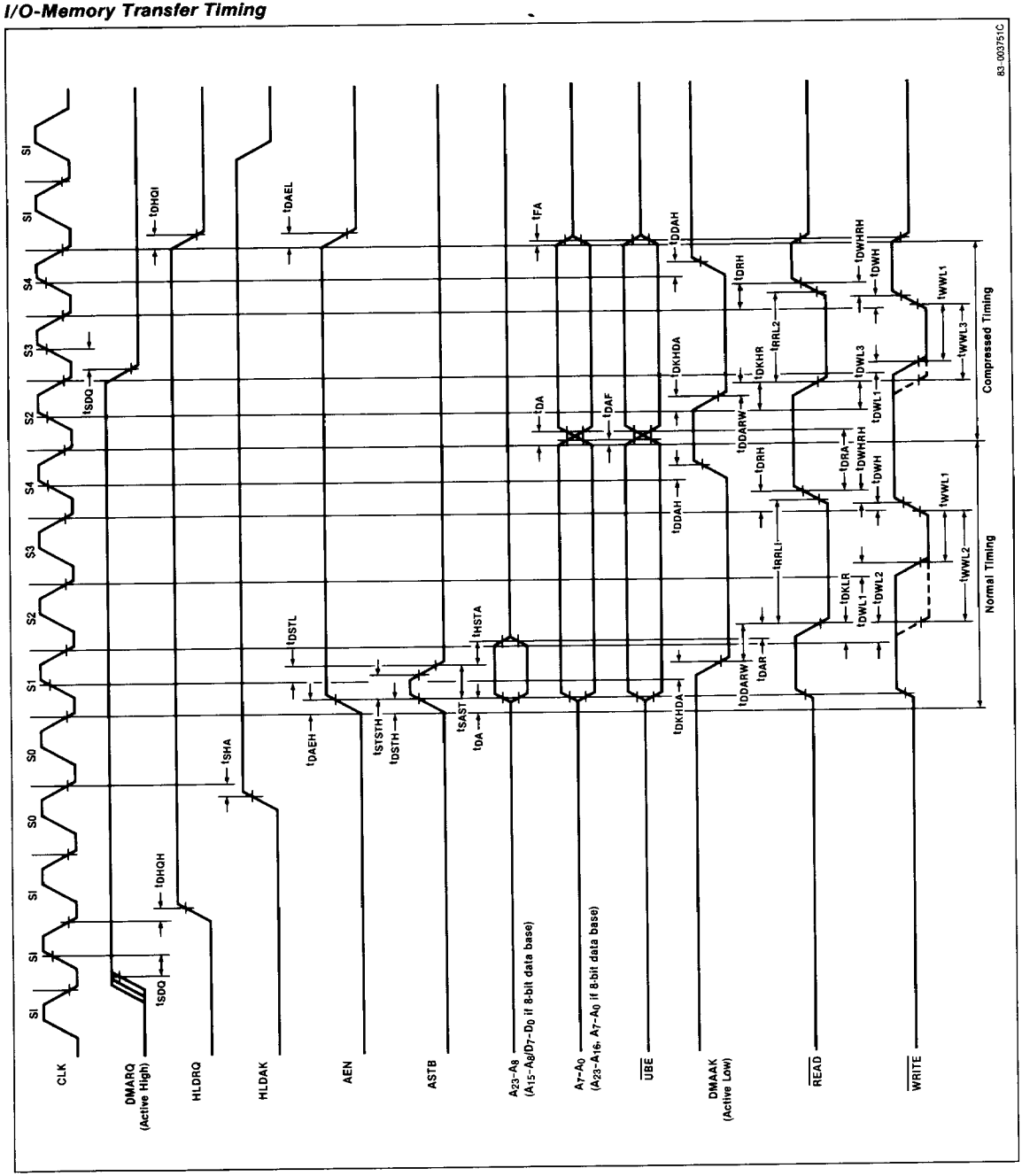
Input/Output Edge Timing



5g

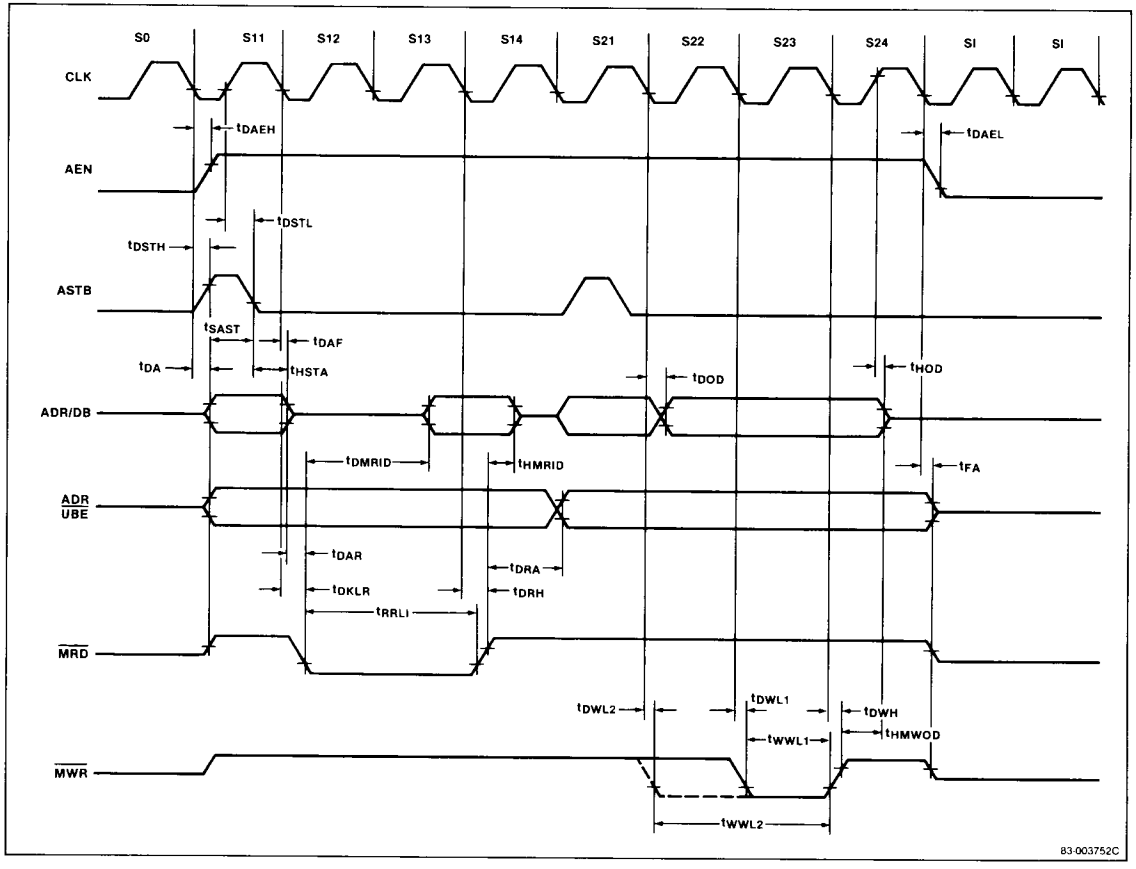
Timing Waveforms (cont)

I/O-Memory Transfer Timing

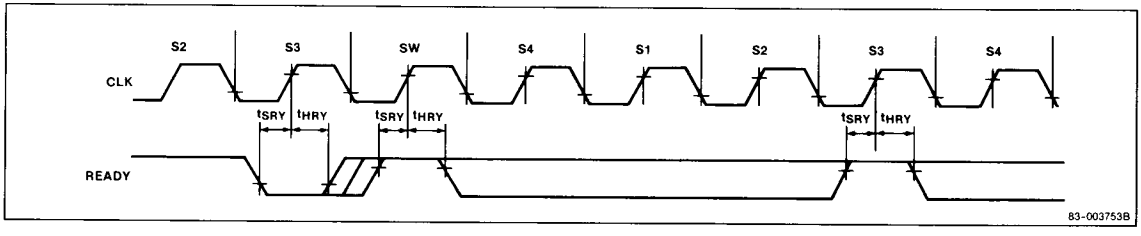


Timing Waveforms (cont)

Memory-to-Memory Transfer Timing



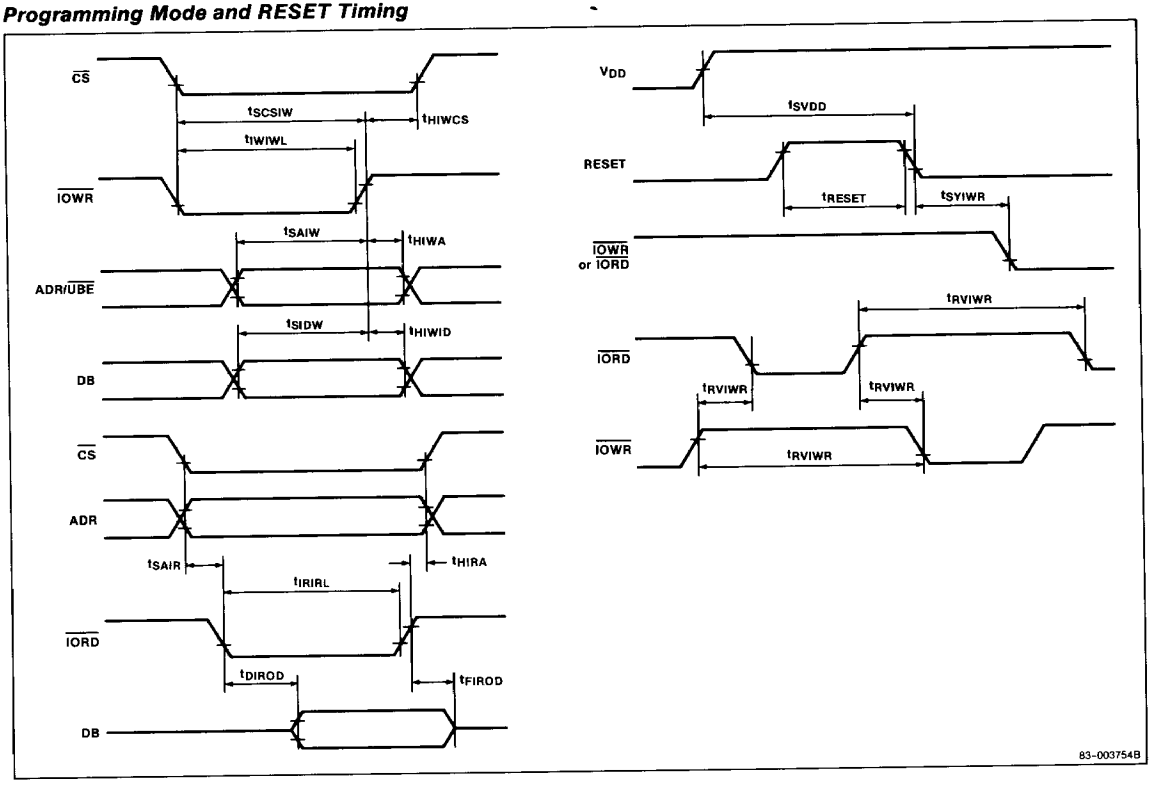
Ready Timing



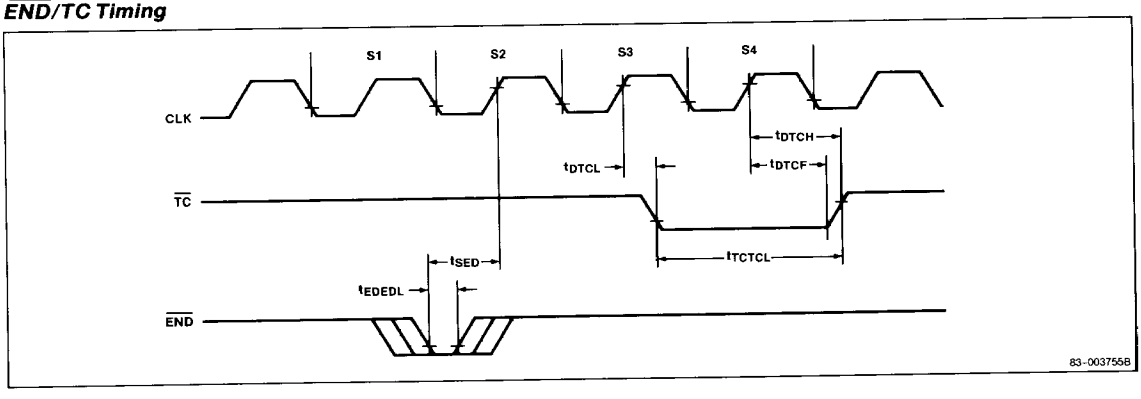
5g

Timing Waveforms (cont)

Programming Mode and RESET Timing

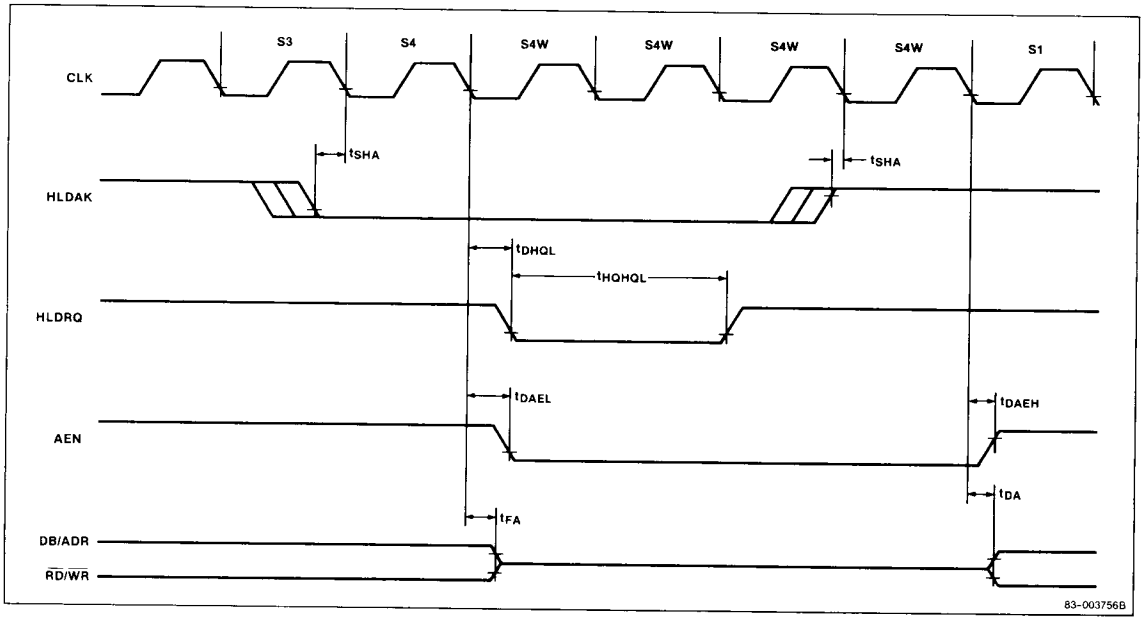


END/TC Timing



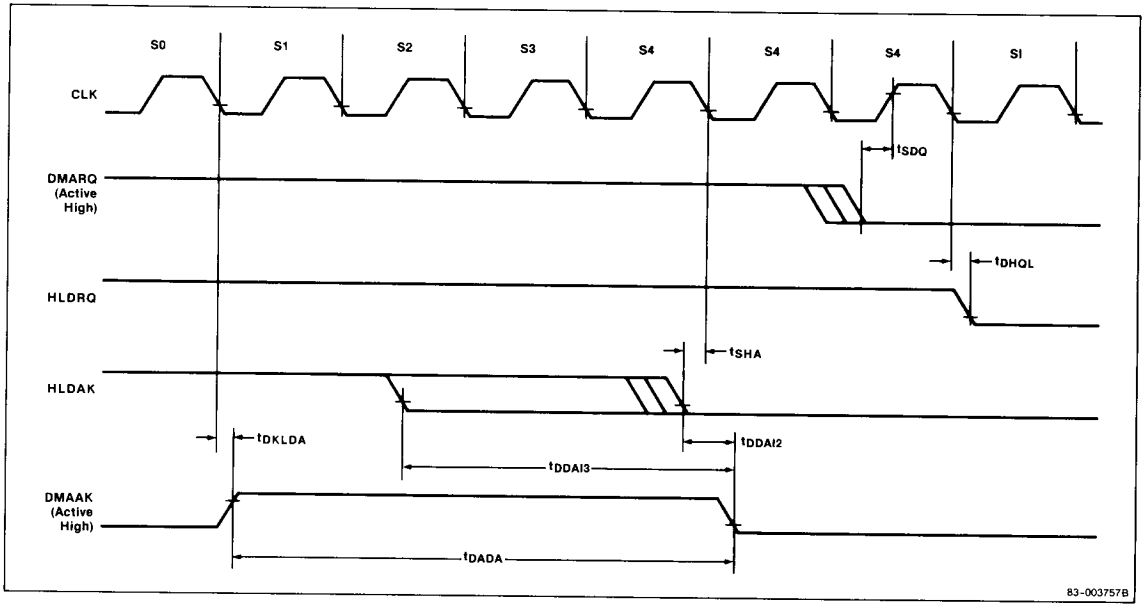
Timing Waveforms (cont)

Bus Wait Timing



5g

Cascade Timing



Functional Description

DMA Operation

The μPD71071 functions in three cycles: idle, DMA, and standby. In an idle or standby cycle, the CPU uses the bus, while in a DMA cycle, the μPD71071 uses it.

Idle Cycle. In an idle cycle, there are no DMA cycles active, but there may be one or more active DMA requests; however, the CPU has not released the bus. The μPD71071 will sample the four DMARQ input pins at every clock. If one or more inputs are active, the corresponding DMA request bits (RQ) are set in the status register and the μPD71071 sends a bus hold request to the CPU. The μPD71071 continues to sample DMA requests until it obtains the bus.

After the CPU returns a HLDAK signal and the μPD71071 obtains the bus, the μPD71071 stops DMA sampling and selects the DMA channel with the highest priority from the valid DMA request signals. Programming of the μPD71071 is done when the μPD71071 is in the idle cycle or the standby mode.

DMA Cycle. In a DMA cycle, the μPD71071 controls the bus and performs DMA transfer operations based on programmed information. Figure 1 outlines the sequential flow of a DMA operation.

Standby Mode. The μPD71071 can also be used in standby mode. It is in standby mode and consumes the static supply current (I_{DD2}) when the clock is turned off and no I/O read or write operations are being performed. All internal registers will retain their contents.

The μPD71071 can be programmed (using $\overline{IOWR}$) and read (using $\overline{IORD}$) with the clock off. The μPD71071 only uses the clock for the DMA data transfer cycles. The clock may be turned off without altering the internal registers when the μPD71071 is in the idle cycle. If the clock is turned off during a DMA transfer, the μPD71071 will not operate correctly. When the clock is off, the DMARQ inputs will not be recognized. The DMARQ inputs could be externally logically ORed and cause an interrupt to the CPU. The CPU could then turn on the clock, thus activating the μPD71071. If the previously programmed mode of operation is still valid, the μPD71071 does not have to be reprogrammed.

Data Bus Width

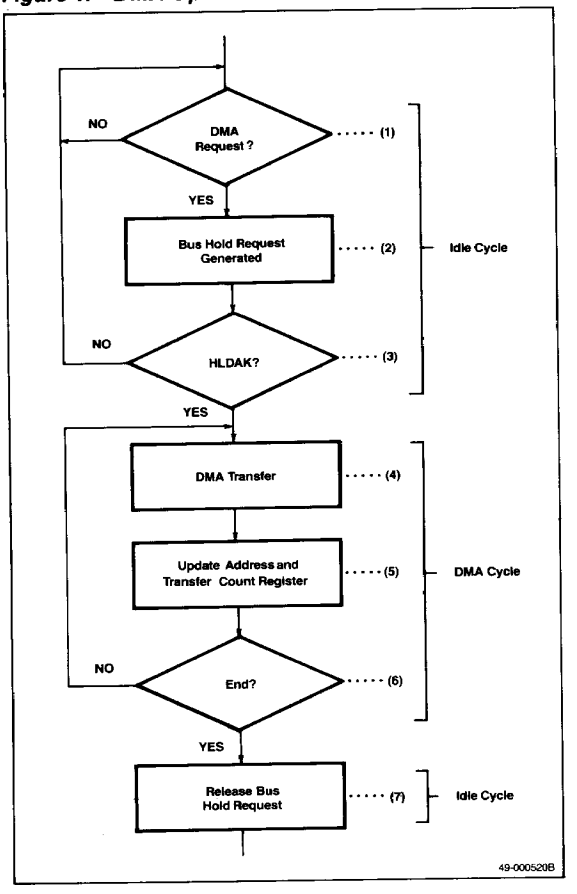
In order to allow an easy interface with an 8- or 16-bit CPU, the data bus width of the μPD71071 is user programmable for 8 or 16 bits. A 16-bit data bus allows 16-bit memory-to-memory DMA transfers and also provides a one-I/O bus cycle access to the 16-bit internal registers.

Table 1 shows the relationship of the data bus width, A_0, $\overline{UBE}$, and the internal registers.

Table 1. Data Bus Width

Bus Width	A_0	$\overline{UBE}$	Internal Read/Write Registers
8 bits	X	X	D_7-D_0 ⟷ 8-bit internal register
16 bits	0	1	D_7-D_0 ⟷ 8-bit internal register
	1	0	D_{15}-D_8 ⟷ 8-bit internal register
	0	0	D_{15}-D_0 ⟷ 16-bit internal register

Figure 1. DMA Operation Flow



Terminal Count

The μPD71071 ends DMA service when it generates a terminal count ($\overline{TC}$) or when the $\overline{END}$ input becomes active. A terminal count is produced when a borrow is generated by the current count register and a low-level pulse is output to the $\overline{TC}$ pin. Figure 2 shows that the current count register is tested after each DMA operation.

If autoinitialize is not set when DMA service ends, the mask register bit applicable to the channel where service ended is set, and the DMARQ input of that channel is masked.

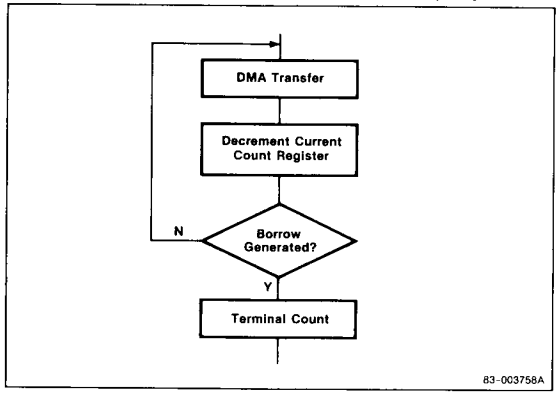
DMA Transfer Type

The type of transfer the μPD71071 performs depends on the following conditions.

- Memory-to-memory transfer enable
- Direction of memory-to-I/O transfer (each channel)
- Transfer mode (each channel)
- Bus mode

Memory-to-Memory Transfer Enable. The μPD71071 can perform memory-to-I/O transfers (one transfer cycle in one bus cycle) and memory-to-memory transfers (one transfer in two bus cycles). To select memory-to-memory transfer, set bit 0 of the device control register to 1. The DMA channels used in memory-to-memory transfers are fixed, with channel 0 as the source channel and channel 1 as the destination channel. Channels 2 and 3 cannot be used in memory-to-memory transfers. The contents of the count registers and word/byte transfer modes of channels 0 and 1 should be the same when performing memory-to-memory transfer.

Figure 2. Generation of Terminal Count ($\overline{TC}$)



For memory-to-memory byte transfer in 16-bit data bus mode, a read data from upper data bus is to be written to upper data bus, while a read data from lower data bus is to be written to lower data bus. Therefore, start addresses for source and destination must be the even-even or odd-odd. For word transfer, only even-even addresses are to be set for source and destination. (See Byte/Word Transfer paragraphs below.) When DMARQ0 (channel 0) becomes active, the transfer is initiated.

During memory-to-memory bus cycles in the 16-bit mode, data read from the DMAC's upper (lower) data bus is written to the upper (lower) data bus of the destination device. Thus, for word transfers, only even source and destination addresses should be used.

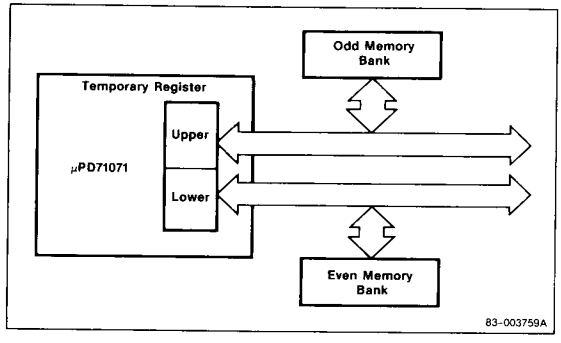
The DMA request input pin or a software DMA request to channel 0 may initiate memory-to-memory transfers. The μPD71071 performs the following operations until a channel 1 terminal count or $\overline{END}$ input is present:

- During the first bus cycle, the memory data pointed to by the current address register of channel 0 is read into the temporary register of the μPD71071 and the address and count of channel 0 are updated.
- During the second bus cycle, the temporary register data is written to the memory location shown by the current address register of channel 1, and the address and count of channel 1 are updated.

Note: If DMARQ1 (channel 1) becomes active, the μPD71071 will perform memory-to-I/O transfer even though memory-to-memory transfer is selected. Since this may cause erroneous memory-to-memory transfers, mask out channel 1 (DMARQ1) by setting bit 1 of the mask register to 1 before starting memory-to-memory transfers.

During memory-to-memory transfers, the addresses on the source side (channel 0) can be fixed by setting bit 1 of the device control register to 1. In this manner, a

Figure 3. Memory-to-Memory Transfer in 16-Bit Data Bus Mode



5g

range of memory can be initialized with the same value since the contents of the source address never change. During memory-to-memory transfer, the DMAAK signal and channel 0's terminal count ($\overline{TC}$) pulse are not output. (See figure 3.)

Direction of Memory-to-I/O Transfers. All DMA transfers use memory as a reference point. Therefore, a DMA read reads a memory location and writes to an I/O port. A DMA write reads an I/O port and writes the data to a memory location. In memory-to-I/O transfer, use the mode control register to set one of the transfer directions in table 2 for each channel and activate the appropriate control signals.

Table 2. Transfer Direction

Transfer Direction	Activated Signals
Memory → I/O (DMA read)	$\overline{IOWR}$, $\overline{MRD}$
I/O → memory (DMA write)	$\overline{IORD}$, $\overline{MWR}$
Verify (Outputs addresses only. Does not perform a transfer.)	—

Transfer Modes. In memory-to-I/O transfer, the mode control register selects the single, demand, or block mode of DMA transfer for each channel. The conditions for the termination of each transfer characterize each transfer mode. Memory-to-memory transfers have no relationship to single, demand, or block mode. Memory-to-memory transfers are a separate and distinct type of transfer mode. Table 3 shows the various transfer modes and termination conditions.

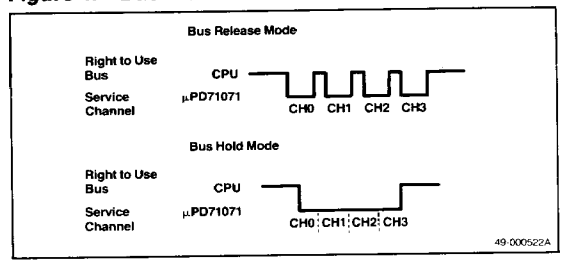
Table 3. Transfer Termination

Transfer Mode	End of Transfer Conditions
Single	After each byte/word
Demand	$\overline{END}$ input Generation of terminal count When DMA request of the channel in service becomes inactive When DMA request of a channel in higher priority becomes active (bus hold mode)
Block	$\overline{END}$ input Generation of terminal count
Memory-to-memory	$\overline{END}$ input Generation of terminal count

Bus Modes. The device control register selects either the bus release or bus hold mode. The bus mode determines when the μPD71071 returns the system bus to the CPU. The μPD71071 can be in either the release or hold modes for the single, demand, or block mode transfers. Therefore, there are six possible mode combinations.

Figure 4 shows that in bus release mode, only one channel can receive service after obtaining the bus. When DMA service ends (end of transfer conditions depend on the transfer mode), the channel returns the bus to the CPU (regardless of the state of other DMA requests) and the μPD71071 enters the idle cycle. When the μPD71071 regains use of the bus, a new DMA operation begins.

Figure 4. Bus Modes



In bus hold mode, several channels can receive service without releasing the bus after obtaining it. If there is another valid DMA request when a channel's DMA service is finished, the new DMA service can begin after the previous service without returning the bus to the CPU. End of transfer conditions depend on the transfer mode. A channel cannot terminate (end count) a transfer mode and immediately start on its next set of transfers. There must be another DMA channel service interleaved or the μPD71071 will put in an idle cycle. The following shows an example of the possible sequences for Channel 2.

CHAN2 → CHANn (n = 0,1,3) → CHAN2
or,
CHAN2 → idle → CHAN2

The operation of single, demand, and block mode transfers depends on whether the μPD71071 is in bus release or bus hold mode. In bus release mode, only one type of bus mode (single, demand, or block) is used each time the μPD71071 has the bus. In bus hold mode, multiple types of transfers are possible. Channel 0 might operate in the demand mode, and channel 1, which could get the bus immediately after channel 0, could operate in block mode.

Single Mode Transfer

In bus release mode, when a channel completes the transfer of a single byte or word, the μPD71071 enters the idle cycle regardless of the state of the DMA request inputs. In this manner, other devices will be able to access the bus on alternate bus cycles.

In bus hold mode, when a channel completes the transfer of a single byte or word, the μPD71071 terminates the channel's service even if it is still asserting a DMA request signal. The μPD71071 will then service the highest priority channel requesting the bus. If there are no requests from any other channel, the μPD71071 releases the bus and enters the idle cycle.

Demand Mode Transfer

In bus release mode, the currently active channel continues its data transfer as long as the DMA request of that channel is active, even though other DMA channels are issuing higher priority requests. When the DMA request of the serviced channel becomes inactive, the μPD71071 releases the bus and enters the idle state, even if the DMA request lines of other channels are active.

In bus hold mode, when the active channel completes a single transfer, the μPD71071 checks DMA request lines (other request lines when $\overline{END}$ or TC, all request lines including the last serviced channel when there is no $\overline{END}$ or TC). If there are active requests, the μPD71071 starts servicing the highest priority channel requesting service. If there is no request, the μPD71071 releases the bus and enters the idle state.

Block Mode Transfer

In bus release mode, the current channel continues data transfer until a terminal count or the external $\overline{END}$ signal becomes active. During this time, the μPD71071 ignores all other DMA requests. After completion of the block transfer, the μPD71071 releases the bus and enters the idle cycle even if DMA requests from other channels are active.

In bus hold mode, the current channel transfers data until a terminal count or the external $\overline{END}$ signal becomes active. When the service is complete, the μPD71071 checks all DMA requests without releasing the bus. If there is an active request, the μPD71071 immediately begins servicing the request. The μPD71071 releases the bus after it honors all DMA requests or a higher priority bus master requests the bus.

Figure 5 shows the operation flow for the six possible transfer and bus mode operations in DMA transfer.

Byte/Word Transfer

If the initialize command selects a 16-bit data bus width, the mode control register can specify DMA transfer in byte or word units for each channel. Table 4 shows the update of the address and count registers during byte/word transfer.

Table 4. Address and Count Registers

Register	Byte Transfer	Word Transfer
Address	±1	±2
Count	−1	−1

During word transfers, two bytes starting at an even address are handled as one word. If word transfer is selected and the initial value of the set address is odd, the μPD71071 will always decrement that address by 1, thus making the address even for the data transfer. For this reason, it is best to select even addresses when transferring words, to avoid destroying data. A_0 and $\overline{UBE}$ control byte and word transfers.

Table 5 shows the relationship between the data bus width, A_0 and $\overline{UBE}$ signals, and data bus status.

Table 5. Data Bus Status

Data Bus Width	A_0	$\overline{UBE}$	Data Bus Status
8 bits	X	1 (1)	D_7-D_0 valid byte
16 bits	0	1	D_7-D_0 valid byte
	1	0	D_{15}-D_8 valid byte
	0	0	D_{15}-D_0 valid word

Note:

(1) Always 1 for an 8-bit bus.

5g

Compressed Timing

In transfers between I/O and memory, a DMA transfer cycle is normally executed in four clocks. However, when the device control register selects compressed timing, one DMA cycle can be executed in a three-clock bus cycle. Compressed timing may be used in the release or hold modes when doing block transfers between I/O and memory. In the demand mode, only use compressed timing in the bus release mode. Compressed timing mode increases data transfer rates by 33%.

The μPD71071 is able to omit one clock period during compressed timing by not updating the upper 16 bits of the latched address. In block mode and demand bus release mode, addresses are output sequentially and the upper 16 bits of addresses latched in external latches need not be updated except after a carry or borrow from A_7 to A_8. For this reason, during compressed timing, the S1 state (output of upper 16 bits of an address for external latching) is omitted in the bus cycles except during the first bus cycle when the upper 16 bits of an address are changed. Figure 6 shows one word waveforms for normal and compressed timing.

Figure 5. Transfer and Bus Modes Operations

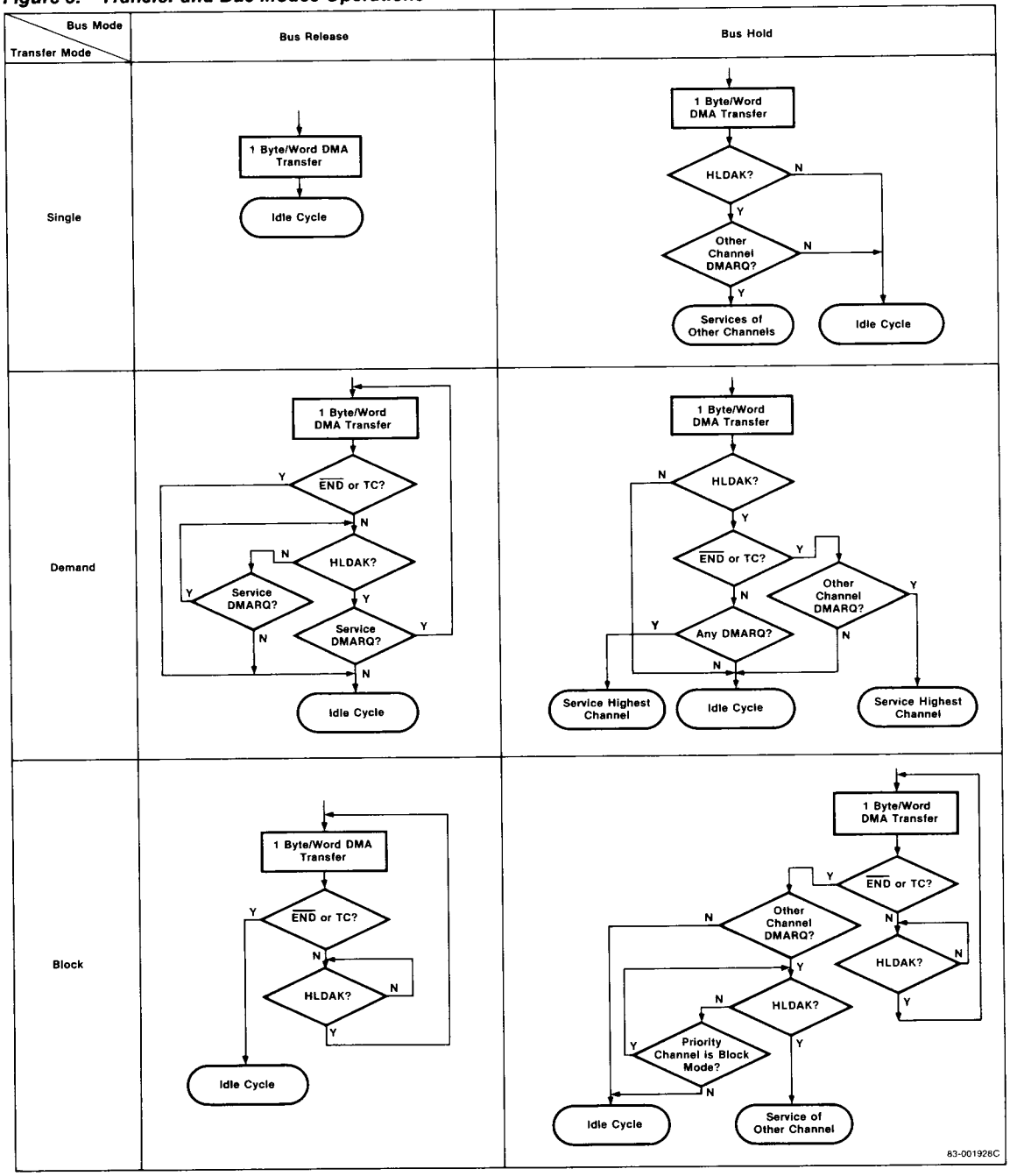
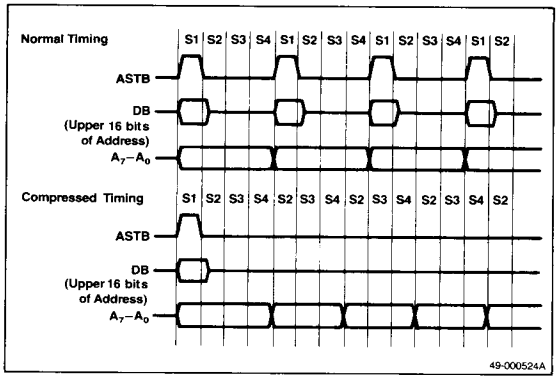


Figure 6. Normal and Compressed Timing Waveforms



Software DMA Requests

The μPD71071 can accept software DMA requests in addition to DMA requests from the four DMARQ pins. Setting the appropriate bit in the request register generates a software DMA request. The mask register does not mask software DMA requests. Software DMA requests operate differently depending on which bus or transfer mode is used.

Bus Mode. When bus release mode is set, the highest priority channel among software DMA requests and DMARQ pins is serviced, and all bits of the request register are cleared when the service is over. Therefore, there is a chance that other software DMA requests will be cancelled.

When bus hold mode is set, only the corresponding bit of the request register is cleared after a DMA service is over. Therefore, all software DMA requests will be serviced in the sequence of their priority level.

Software DMA requests for cascade channels (see Cascade Connection) must be performed in bus hold mode. When a cascade channel is serviced, the master μPD71071 operational mode is changed to bus release mode temporarily and all bits of the request register are cleared when the cascade channel service is over. To avoid this, it is necessary to mask any cascade channels before issuing a software DMA request. After confirming that all DMA software services are complete and all bits of the request register are cleared, the cascade channel masks can be cleared.

Transfer Mode. When single or demand mode is set, the applicable request bits are cleared and software DMA service ends with the transfer of one byte/word. When block mode or memory-to-memory modes are set, service continues until $\overline{END}$ is input or a terminal count is generated. Applicable request bits are cleared when service ends.

Autoinitialize

When the mode control register is set to autoinitialize a channel, the μPD71071 automatically initializes the address and count registers when $\overline{END}$ is input or a terminal count is generated. The contents of the base address and base count registers are transferred to the current address and current count registers, respectively. The applicable bit of the mask register is unaffected. The applicable bit of the mask register is set for channels not programmed for autoinitialize.

The autoinitialize function is useful for the following types of transfers.

Repetitive Input/Output of Memory Area. Figure 7 shows an example of DMA transfer between a CRT controller and memory. After setting the value in the base and current registers, autoinitialize allows repetitive DMA transfer between the CRT controller and the video memory area without CPU involvement.

Continuous Transfer of Several Memory Areas. The CPU can indirectly write to the address or count registers by writing to the base registers. New values can be written to the base registers. In the autoinitialize mode, the value in the base register will be transferred to the address/count registers when termination is reached in the address/count registers. Because of this, the autoinitialize function can perform continuous transfer of several contiguous or noncontiguous memory areas during single or demand bus release modes in the following manner.

During the transfer of data in area 1 (the first area being transferred), the CPU can write address and count information about area 2 (the second area to be transferred). Generation of a terminal count for area 1 results in the transfer of information of area 2 to the address and count registers. This will cause area 2 to be transferred. Figure 8 illustrates this procedure.

Channel Priority

Each of the μPD71071's four channels has its own priority. When there are DMA requests from several channels simultaneously, the channel with the highest priority will be serviced. The device control register selects one of two channel priority methods: fixed and rotational priority. In fixed priority, the priority (starting with the highest) is channel 0, 1, 2, and 3, respectively. In rotational priority, priority order is rotated so that the channel that has just been given service receives the lowest priority and the next highest channel number is given the highest priority. This method prevents exclusive servicing of some channel(s). Figure 9 shows the two priority order methods.

5g

Figure 7. Autoinitialize Application 1

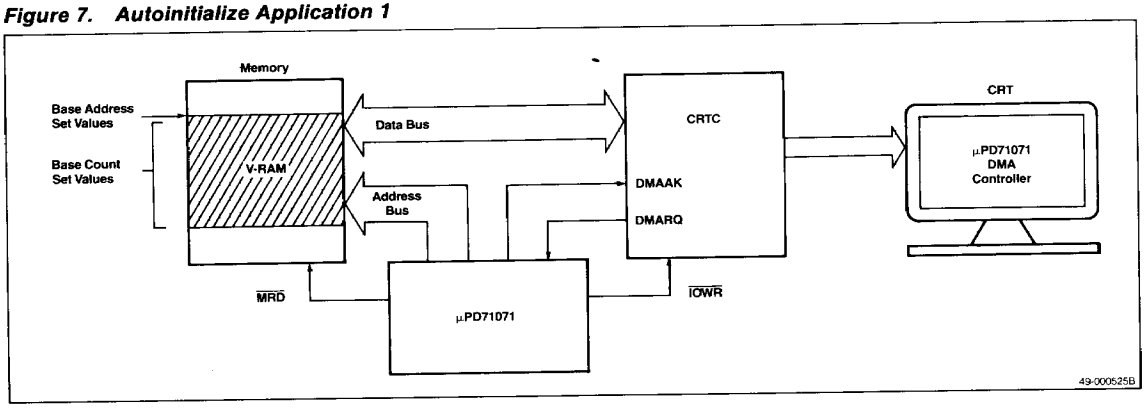


Figure 8. Autoinitialize Application 2

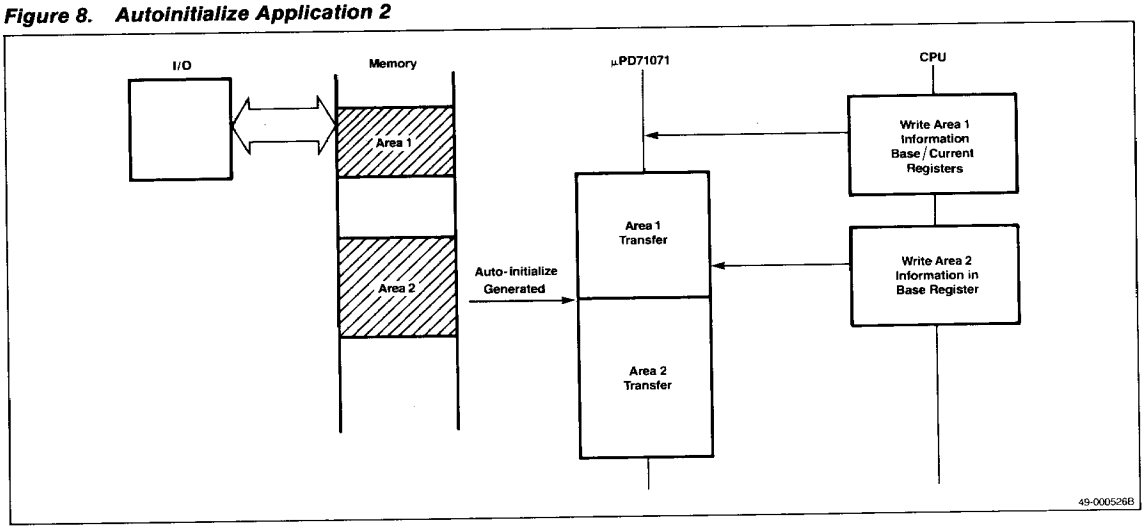
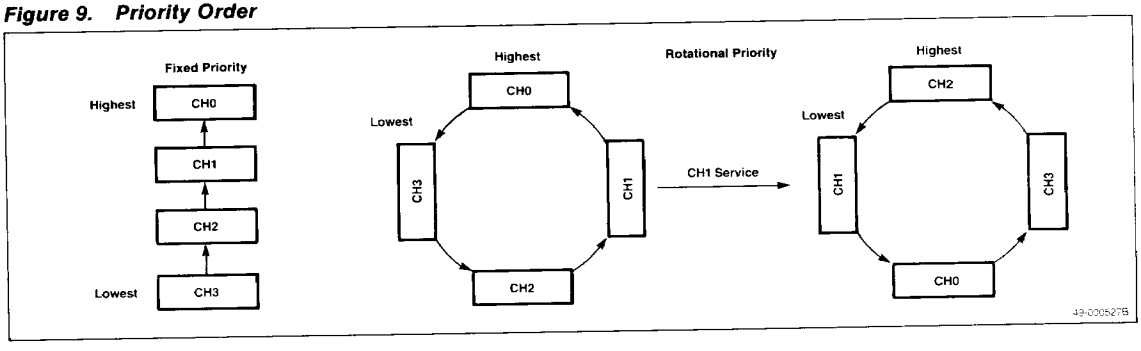


Figure 9. Priority Order



Cascade Connection

The μPD71071 can be cascaded to expand the system DMA channel capacity. To connect a μPD71071 for cascading (figure 10), perform the following operations.

(1) Connect pins HLDRQ and HLDAK of the second-stage (slave) μPD71071 to pins DMARQ and DMAAK of any channel of the first-stage (master) μPD71071.

(2) To select the cascade mode of a particular channel of a master μPD71071, set bits 7 and 6 of that channel's mode control register to 11.

When a channel is set to the cascade mode in a master μPD71071, DMARQ, DMAAK, HLDRQ, HLDAK, and RESET are the only valid signals in the master μPD71071. The other signals are disabled. The master cascade channel only intermediates hold request/hold acknowledge between the slave and CPU.

The master μPD71071 always operates in the bus release mode when a cascade channel is in service (even when the bus hold mode is set). Other DMA requests are ignored while a cascade channel is in service. When the slave μPD71071 ends DMA service and moves into an idle cycle, the master also moves to an idle cycle and releases the bus. At this time, all bits of the master's request register are cleared. The master operates its non-cascaded channels normally.

Bus Wait Operation

In systems using a μPD70208/70216 (V40/V50) as the CPU, the refresh control unit in the CPU changes the HLDAK signal to inactive (even during a DMA cycle) and uses the bus. Here, the μPD71071 automatically performs a bus wait operation. This system has a bus master (V40/V50) whose priority level is higher than that of the μPD71071.

The μPD71071 executes the bus wait operation when the HLDAK signal becomes inactive in an operating mode where transfer is executed continuously in block mode, during demand bus release mode, or during memory-to-memory transfer.

When HLDAK becomes inactive during service in other operating modes, the operation returns to the idle cycle and transfers control of the bus to the higher bus master.

Figure 11 shows that when the HLDAK signal becomes inactive during a continuous transfer, the μPD71071 is set up in an S4w state (bus wait). Operation moves to the idle cycle if DMARQ is inactive in the demand mode. The HLDRQ signal is made inactive for a period of about two clocks and the bus is released. The S4w state is repeated until the HLDAK signal again becomes active and the interrupted service is immediately restarted.

5g

Figure 10. Cascade Connection Example

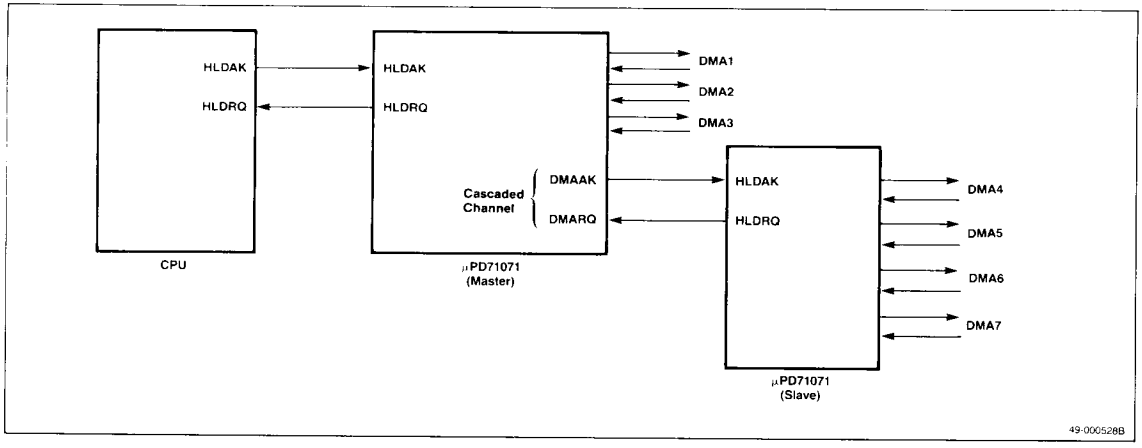
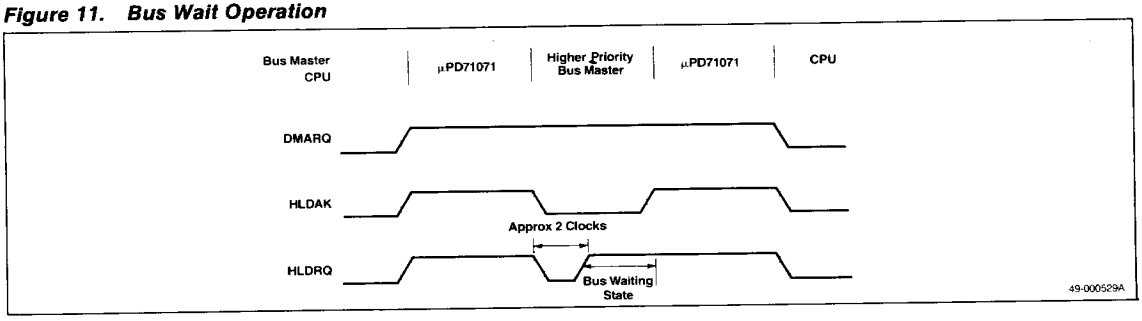


Figure 11. Bus Wait Operation



Programming the μPD71071

To prepare a channel for DMA transfer, you must select the following characteristics.

- Starting address for the transfer
- Number of byte/word transfers
- DMA operating modes
- Data bus widths
- Active levels of the DMARQ and DMAAK signals

When reading from or writing to a μPD71071 internal register, address lines A_3-A_0 select the register, $\overline{IORD}$ or $\overline{IOWR}$ select the data transfer direction, and $\overline{CS}$ enables the transfer. Table 6 shows the register and command configurations.

Table 6. Register Configuration

Register	Bit size
Channel	5
Base address	24 (4)
Current address	24 (4)
Base count	16 (4)
Current count	16 (4)
Mode control	7 (4)
Device control	10
Status	8
Request	4
Mask	4
Temporary	16

Note:

When using a 16-bit CPU and selecting a 16-bit data bus, the word IN/OUT instruction can be used to read/write information two bytes at a time. However, commands in table 7 suffixed with B must be issued with the byte IN/OUT instruction.

Initialize

Use the initialize command as a software initialize to the μPD71071 or to set the width of the data bus. When using a 16-bit CPU, set the data bus width to 16 bits first. Figure 12 shows the initialize command format.

Bit 0. When the RES bit is set, the internal state of the μPD71071 is initialized and will be the same as when a hardware reset is used (except for data bus width selection). A software reset leaves bit 16B intact whereas a hardware reset selects the 8-bit data bus. After initialization, the registers are as in table 8 and the RES bit is cleared automatically.

Table 8. Register Initialization

Register	Initialization Operation
Initialize	Clears bit 0 only
Address	No change
Count	No change
Channel	Selects channel 0, current and base
Mode control	Clears all bits
Device control	Clears all bits
Status	Clears bits 3-0 only
Request	Clears all bits
Mask	Sets all bits (masks all channels)
Temporary	Clears all bits

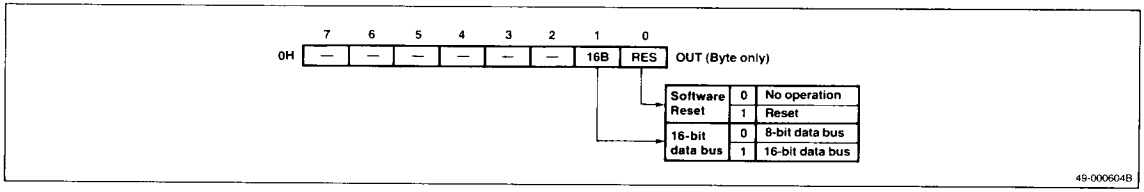
Bit 1. The 16B bit determines the data bus width. When using the μPD71071 in a 16-bit system, set this bit immediately after a hardware reset since a hardware reset always initializes it to the 8-bit data bus mode.

Table 7. Command Configuration

Address	R/W	Command Name	MSB Format LSB							
0H	W(B)	Initialize	—	—	—	—	—	—	16B	RES
1H	R(B)	Channel Register Read	—	—	—	BASE	SEL3	SEL2	SEL1	SEL0
	W(B)	Channel Register Write	—	—	—	—	—	BASE	SELCH	
2H	R/W	Count Register Read/Write	C7	C6	C5	C4	C3	C2	C1	C0
3H	R/W		C15	C14	C13	C12	C11	C10	C9	C8
4H	R/W	Address Register Read/Write	A7	A6	A5	A4	A3	A2	A1	A0
5H	R/W		A15	A14	A13	A12	A11	A10	A9	A8
6H	R/W(B)		A23	A22	A21	A20	A19	A18	A17	A16
8H	R/W	Device Control Reg. Read/Write	AKL	RQL	EXW	ROT	CMP	DDMA	AHLD	MTM
9H	R/W		—	—	—	—	—	—	WEV	BHLD
0AH	R/W(B)	Mode Control Reg. Read/Write	TMODE		ADIR	AUTI	TDIR		—	W/B
0BH	R(B)	Status Register Read	RQ3	RQ2	RQ1	RQ0	TC3	TC2	TC1	TC0
0CH	R	Temporary Reg. (lower) Read	T7	T6	T5	T4	T3	T2	T1	T0
0DH	R	Temporary Reg. (higher) Read	T15	T14	T13	T12	T11	T10	T9	T8
0EH	R/W(B)	Request Reg. Read/Write	—	—	—	—	SRQ3	SRQ2	SRQ1	SRQ0
0FH	R/W(B)	Mask Reg. Read/Write	—	—	—	—	M3	M2	M1	M0

49-000603B

Figure 12. Initialize Command Format



49-000604B

Channel Register

This command reads and writes the channel register that selects one of four DMA channels for programming the address, count, and mode control registers. Figure 13 shows the channel register read/write format.

Channel Register Read

SEL3-SEL0. These mutually exclusive bits show which of the four channels is currently selected for programming.

BASE. Base = 0. The current register may be read. During a write, the base and current registers will be written to simultaneously.

Base = 1. Only the base registers may be read or written to.

Channel Register Write

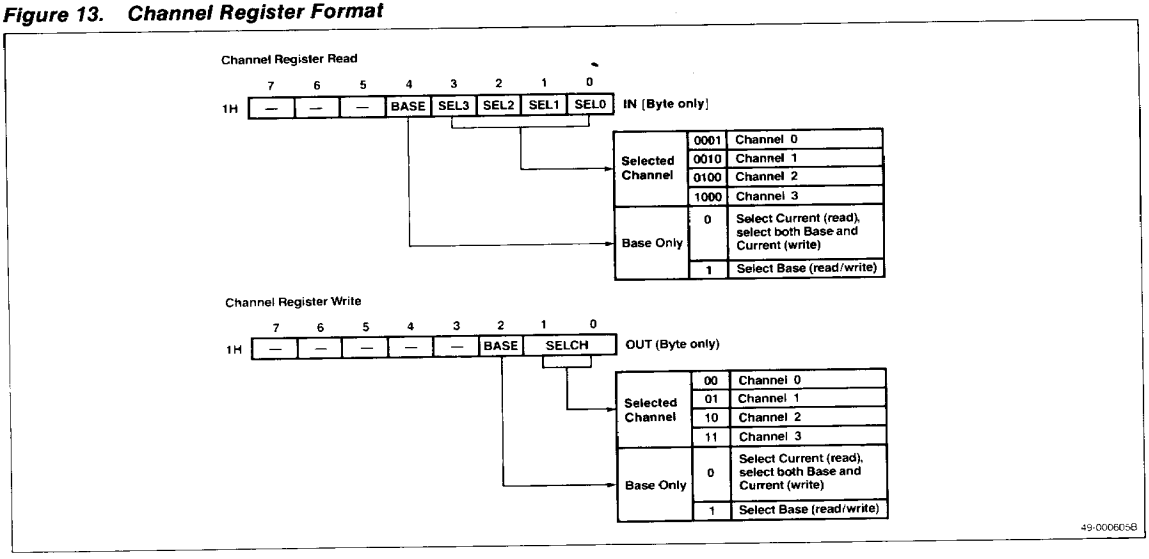
SELCH. This bit selects the channel to be programmed.

BASE. Base = 0. The current register may be read. During a write, the base and current registers will be written to simultaneously.

Base = 1. Only the base registers may be read or written to.

5g

Figure 13. Channel Register Format



Count Register Read/Write

When the 16-bit bus mode is selected, the IN/OUT instruction can directly transfer 16-bit data. The channel register selects one of the count registers. When bit 2 of the channel register write is cleared, a write to the count register updates both the base and current count registers with the new data. If bit 2 of the channel register write is set, a write to the count register only affects the base count register.

The base count registers hold the initial count value until a new count is specified. If autoinitialize is enabled, this value is transferred to the current count register when an $\overline{END}$ or $\overline{TC}$ is generated. For each DMA transfer, the current count register is decremented by one. Figure 14 shows the count register read/write format.

Address Register Read/Write

When a 16-bit data bus width is selected, the IN/OUT instruction can directly transfer the lower two bytes (4H and 5H) of the register. You must use the byte IN/OUT instruction with the upper byte (6H) of the register. The channel register selects one of the address registers. When bit 2 of the channel register is cleared, a write to the address register updates both the base and current address registers with the new data. If bit 2 of the channel register is set, a write to the address register only affects the base address register.

The base register holds the starting address value until a new setting is made and this value is transferred to the current address register during autoinitialization. For each DMA transfer, the current address register is updated ± 2 during word transfer and ± 1 during byte transfer. Figure 15 shows the address register read/write format.

Device Control Register Read/Write

The device control command reads from and writes to the device control register. When using a 16-bit data bus, use the word IN/OUT instruction to read and write 16-bit data. Figure 16 shows the device control register read/write format.

Figure 14. Count Register Read/Write Format

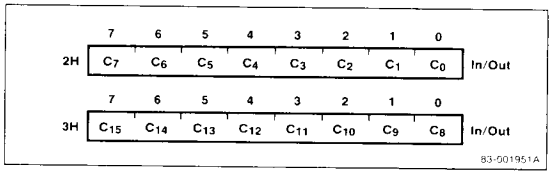


Figure 15. Address Register Read/Write Format

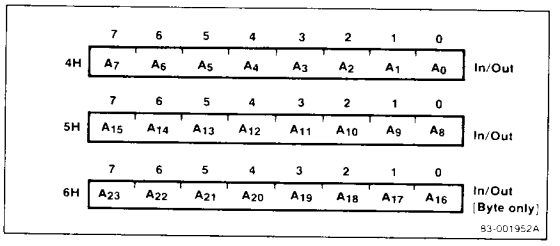
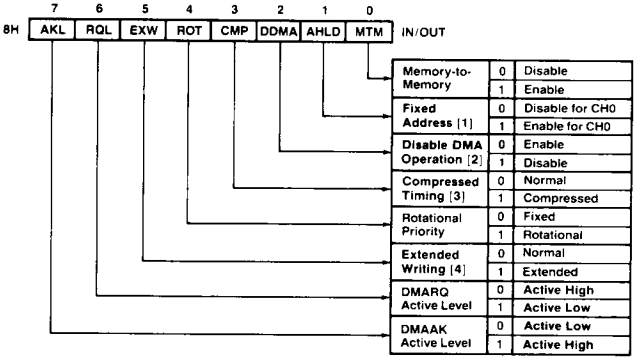
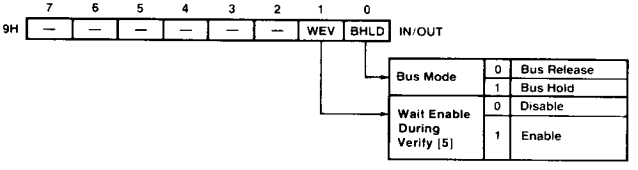


Figure 16. Device Control Register Read/Write Format





Note:

[1] This bit is only used when MTM = 1, [memory-to-memory transfers].

[2] Disables HLDRQ to the CPU to prevent incorrect DMA operation while the μPD71071's registers are being initialized or modified.

[3] When 1, causes the μPD71071 to use compressed timing in the demand bus release mode or in the block mode.

[4] When EXW is 0, the write signal becomes active [normal write] during S3 and SW. When 1, the write signal becomes active during S2, S3, and SW. See figures 27-29.

[5] This bit enables or disables the wait state generated by the READY signal during a verify transfer.

49-000606C

5g

Mode Control Register Read/Write

This command reads from and writes to the mode control register to specify the operating mode for each channel. The channel register selects the mode control register to be programmed. This command must be issued by the byte IN/OUT instruction. Figure 17 shows the mode control register read/write format.

Status Register Read

This command reads the status register for the individual DMA channels. The register has DMA request states and terminal count or $\overline{END}$ information. This command must be issued by the byte IN instruction. Figure 18 shows the status register read format.

Temporary Register Read

When a 16-bit data bus is selected, the IN instruction will read 16-bit data with this command. The last data transferred in memory-to-memory transfer is stored in the temporary register. Figure 19 shows the temporary register read format.

Request Register Read/Write

This command reads from and writes to the request register to generate DMA requests by software for the four corresponding DMA channels. This command may be issued by the byte IN/OUT instruction. Figure 20 shows the reguest register read/write format.

Mask Register Read/Write

This command reads from and writes to the mask register to mask or unmask external DMA requests for the corresponding four DMA channels (DMARQ3-DMARQ0). This command may be issued by the byte IN/OUT instruction. Figure 21 shows the mask register read/write format.

DMA Transfer Modes

Figures 22-27 show state transition diagrams for the different modes of DMA transfer.

Figure 23 shows the state of a master μPD71071 when an input from a slave μPD71071 (cascaded μPD71071) is using the system bus.

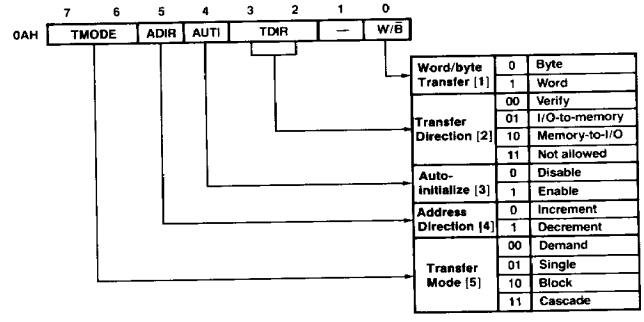
Transfer Timing

Figures 28-30 show μPD71071 timing waveforms.

Examples of System Configuration

Figures 31-32 show system configuration examples using the 8-bit μPD70108 CPU and the 16-bit μPD70116 CPU. The μPD71082 externally latches addresses and data.

Figure 17. Mode Control Register Read/Write Format



Note:

[1] This bit selects byte or word transfer for DMA transfers. This bit is used only in 16-bit data bus mode.

[2] These bits select the DMA transfer direction between memory and I/O. These bits are meaningless during memory-to-memory transfer.

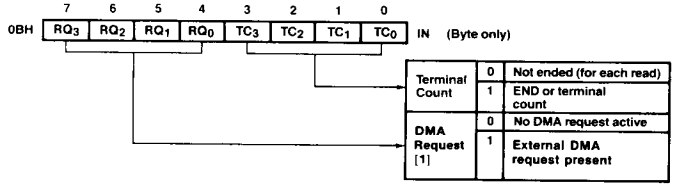
[3] Channel 0 and 1 must have the same AUTI bit value when performing memory-to-memory transfer.

[4] This bit decides the update direction of the Current Address Register. When ADIR is 0, the register increments by 1 for a byte transfer and by 2 for a word transfer. When ADIR is 1, the register decrements by 1 for a byte transfer and by 2 for a word transfer.

[5] These bits select the transfer mode during DMA transfer between memory and I/O and are meaningless during memory-to-memory transfer.

49-0006088

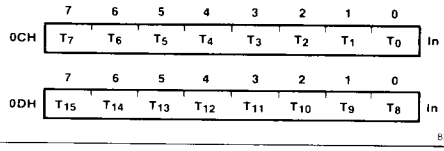
Figure 18. Status Register Read Format



Note:
[1] Bits RQ_3–RQ_0 will be set if an external hardware DMA request is pending even if its request bit is masked. Software-generated DMA requests, hardware reset, and software reset will not affect these bits.

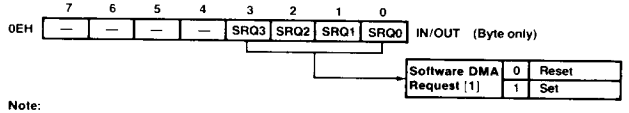
49-000607B

Figure 19. Temporary Register Read Format



83-001953A

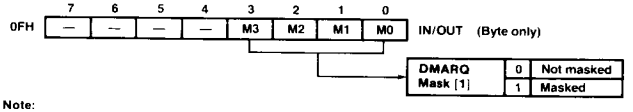
Figure 20. Request Register Read/Write Format



Note:
[1] In memory-to-memory applications, only bit SRQ0 will be cleared at terminal count or when an $\overline{END}$ input is present.

83-001887B

Figure 21. Mask Register Read/Write Format



Note:
[1] In memory-to-memory applications, only bits M0 and M1 will be set at terminal count or when an $\overline{END}$ input is present.

49-000609B

Figure 22. Idle Cycle

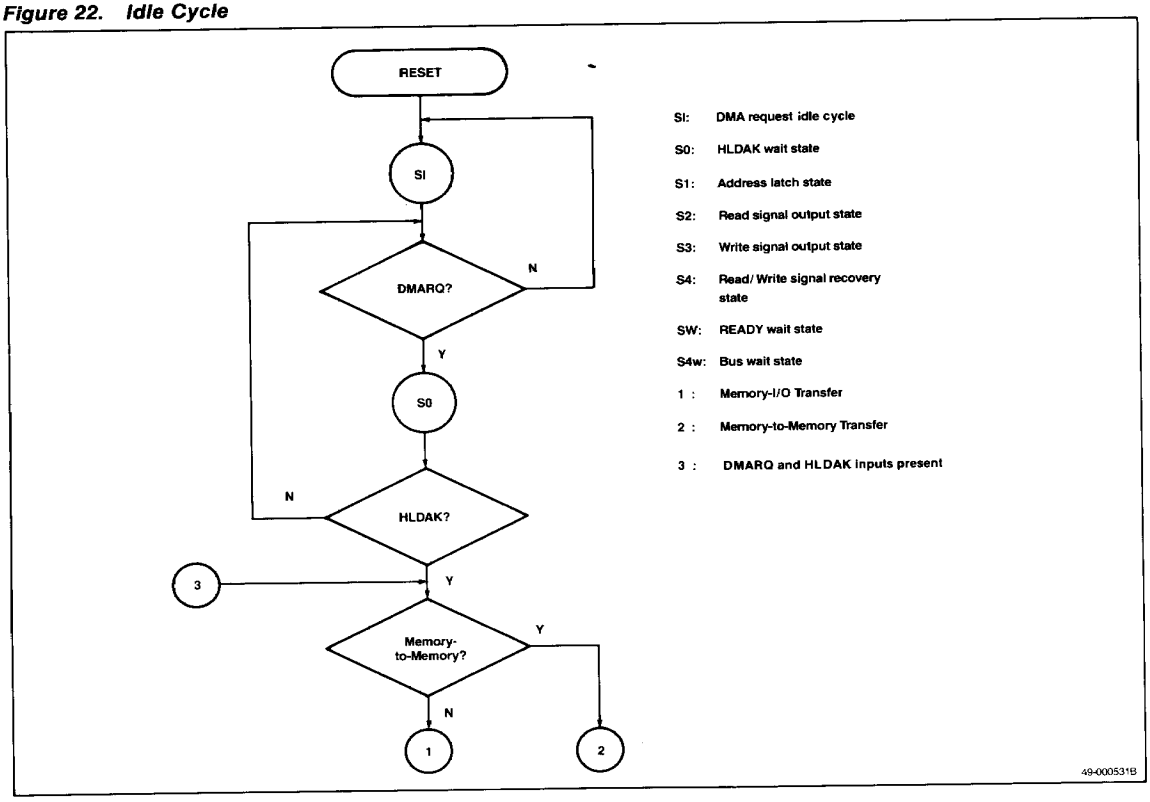


Figure 23. DMA Cycle, Cascade Mode

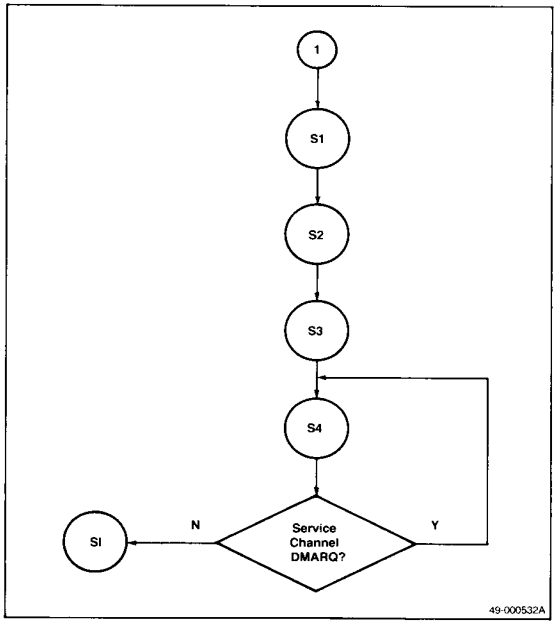
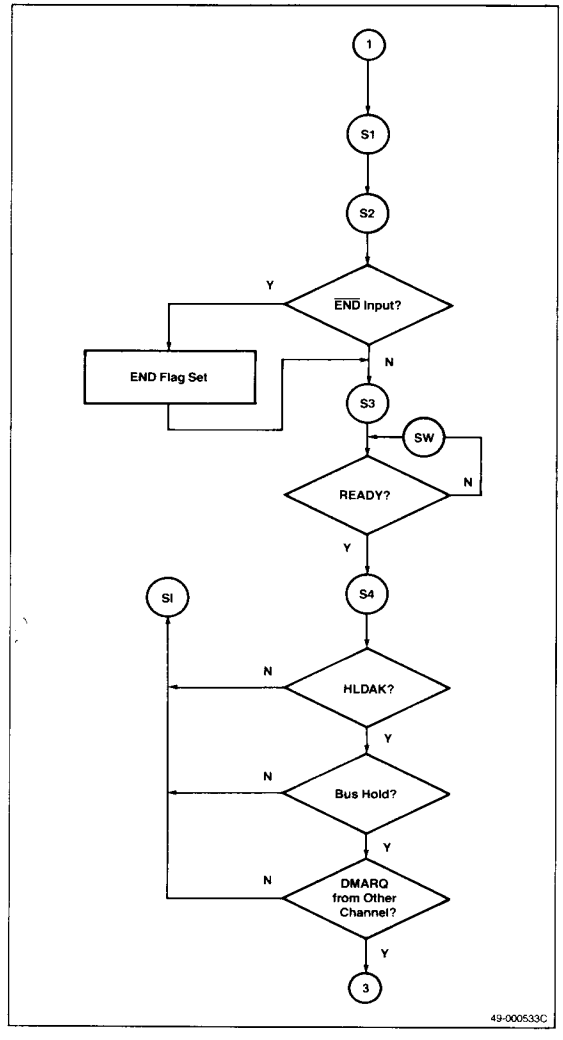
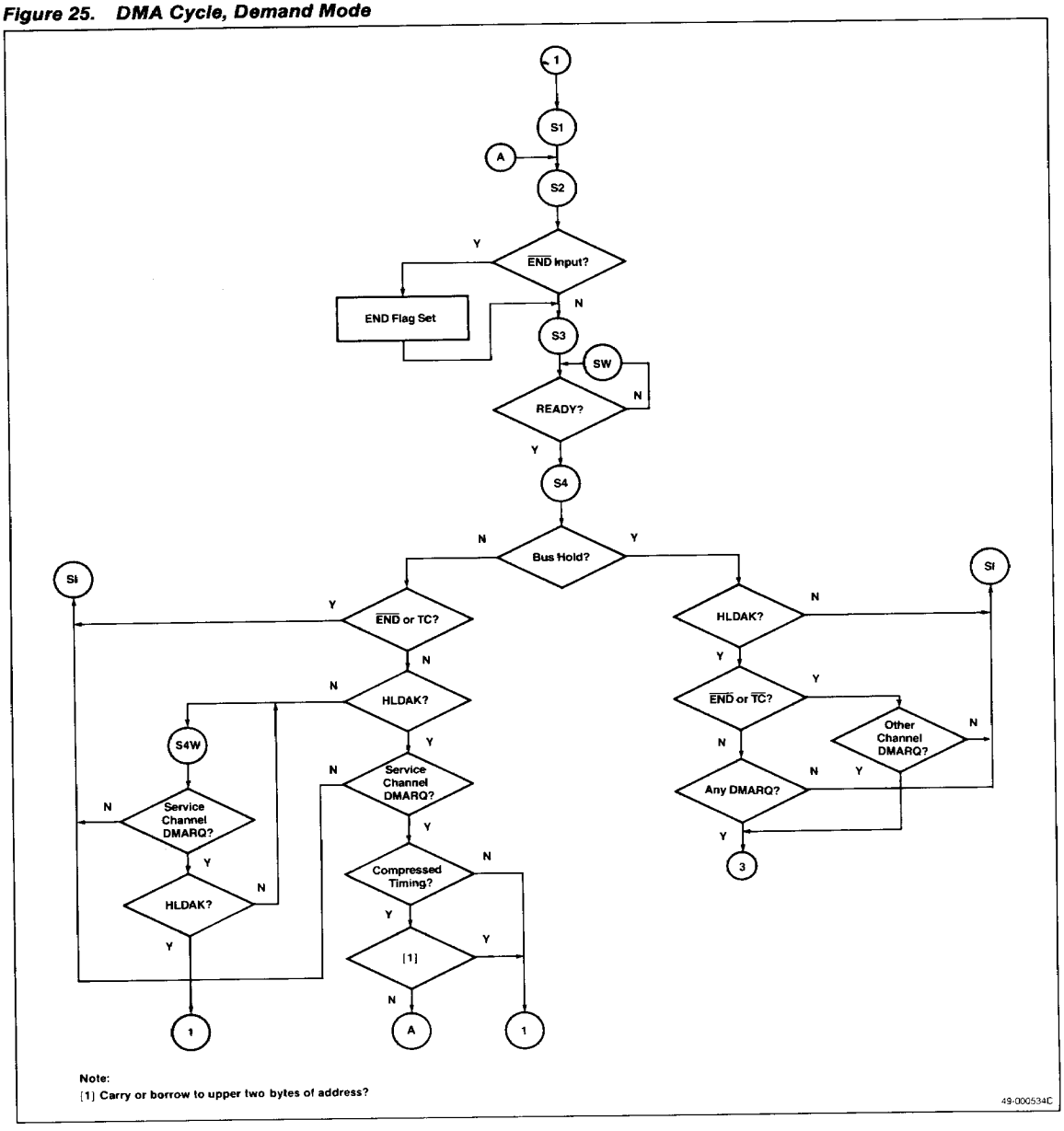


Figure 24. DMA Cycle, Single Mode



5g

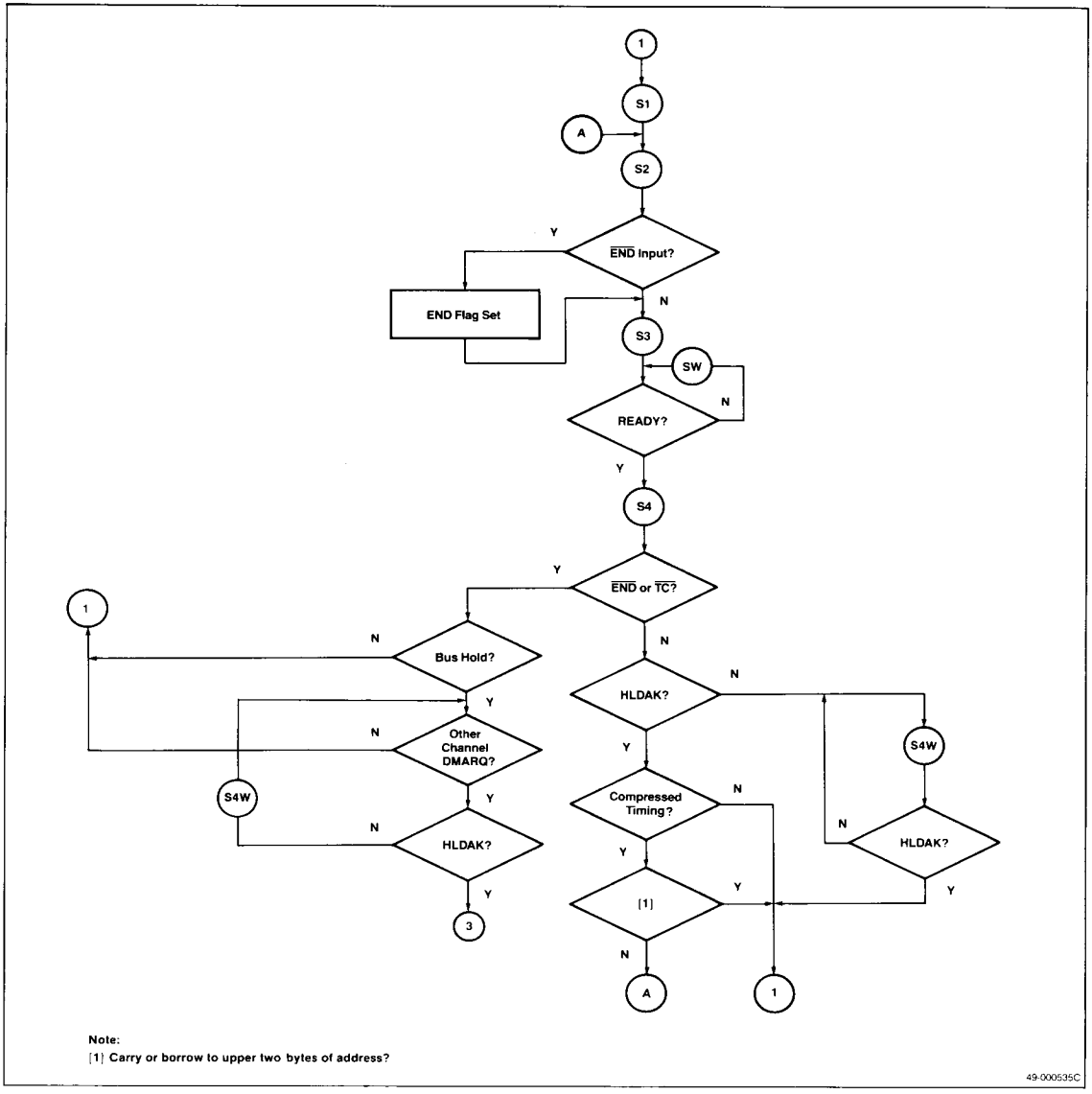
Figure 25. DMA Cycle, Demand Mode



Note:
[1] Carry or borrow to upper two bytes of address?

49-000534C

Figure 26. DMA Cycle, Block Mode



5g

Figure 27. DMA Cycle, Memory-to-Memory Transfer

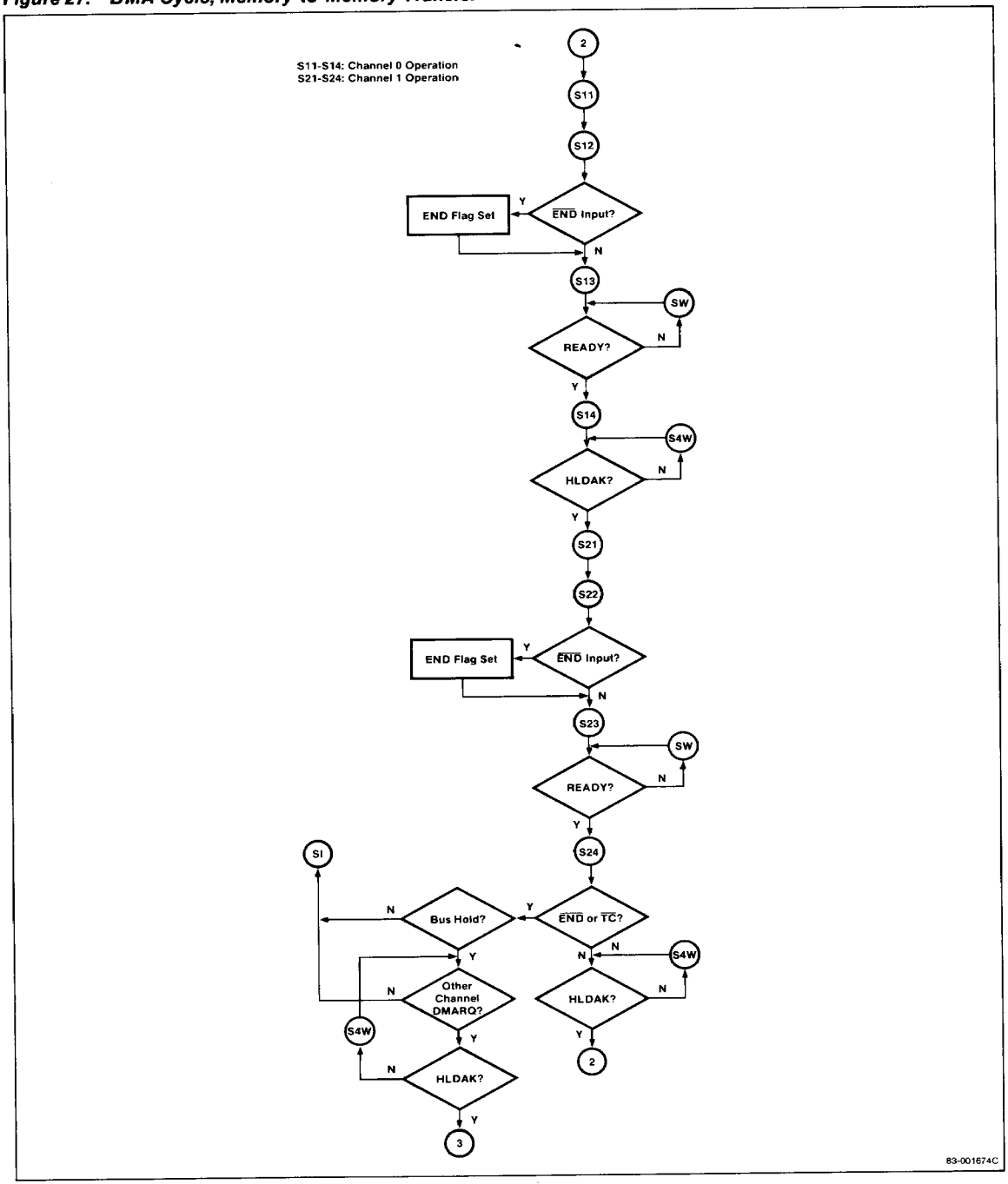
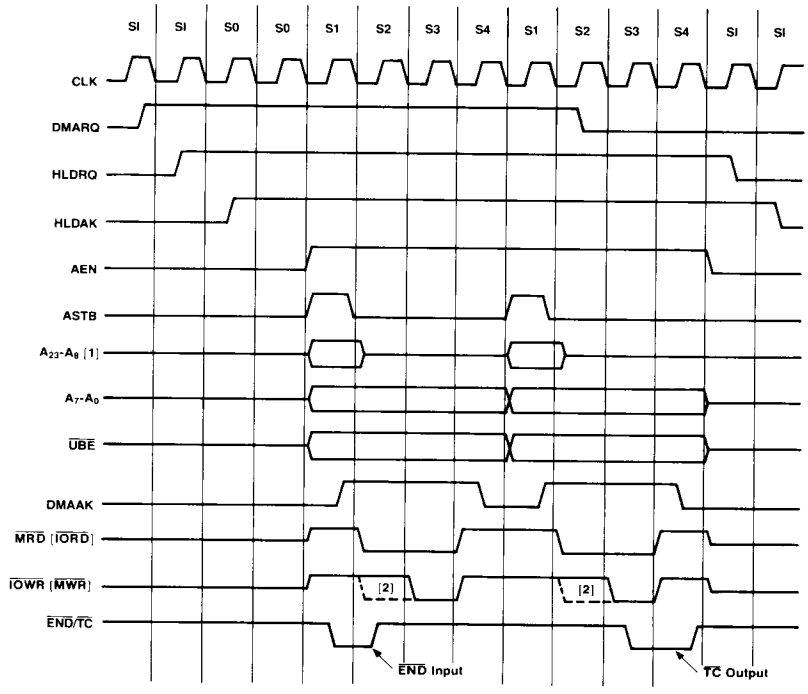


Figure 28. Memory-I/O Transfer, Normal Timing



Note:

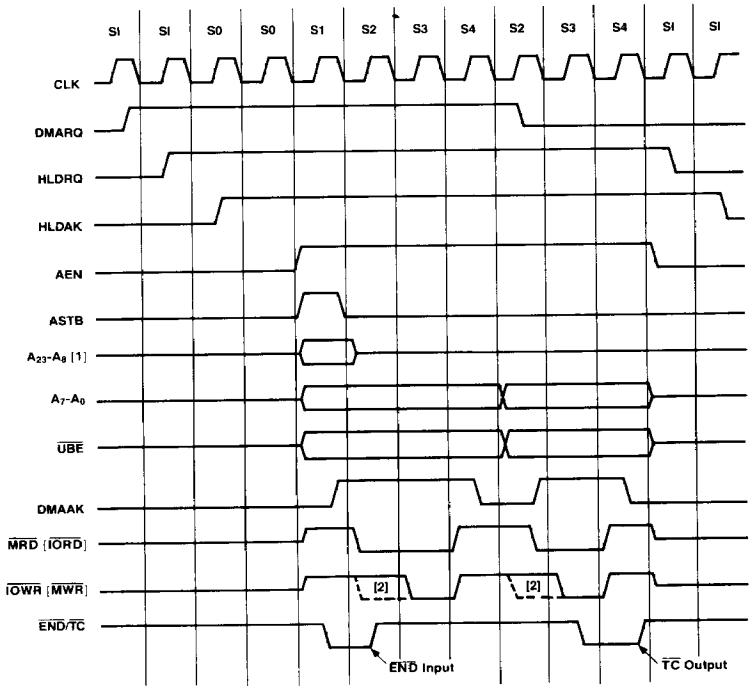
[1] When an 8-bit data bus is selected, D_{15}-D_8 are not used. Therefore, A_{23}-A_{16} are not multiplexed address/data signals and will have the same timing as A_7-A_0.

[2] The broken lines of the write signal are for extended write timing.

49-000537C

5g

Figure 29. Memory-I/O Transfer, Compressed Timing



Note:
[1] When an 8-bit data bus is selected, D_{15}-D_8 are not used. Therefore, A_{23}-A_{16} are not multiplexed address/data signals and will have the same timing as A_7-A_0.
[2] The broken lines of the write signal are for extended write timing.

49-000538C

Figure 30. Memory-to-Memory Transfer

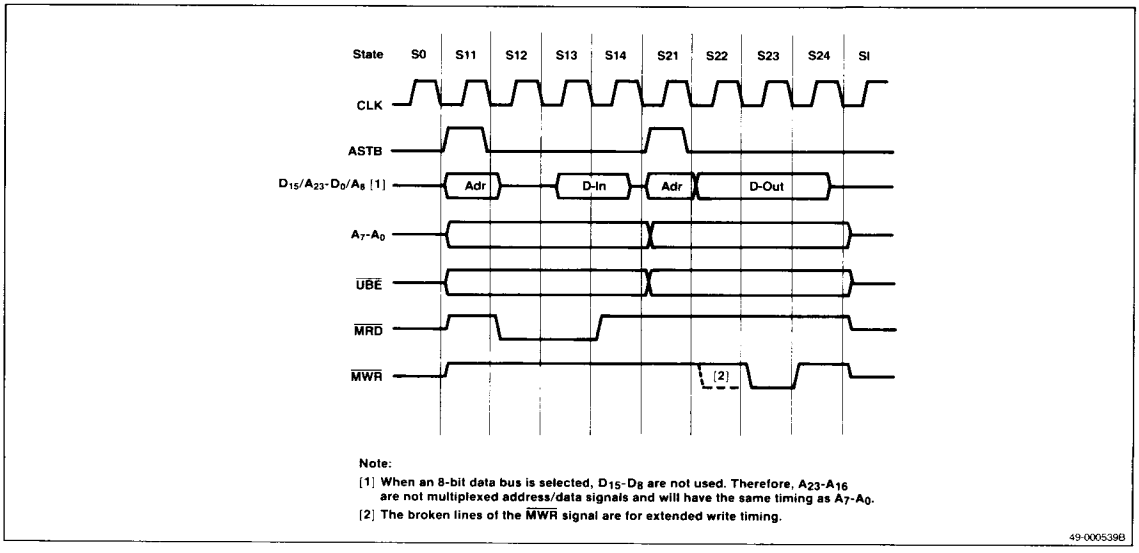
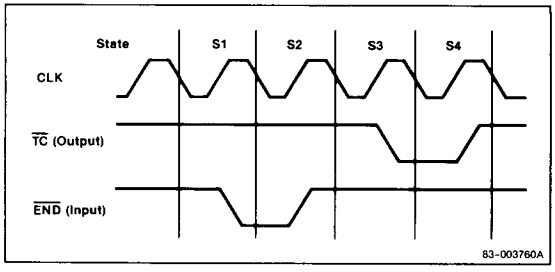


Figure 31. END/TC Input/Output



5g

Figure 32. System Configuration with μPD70108

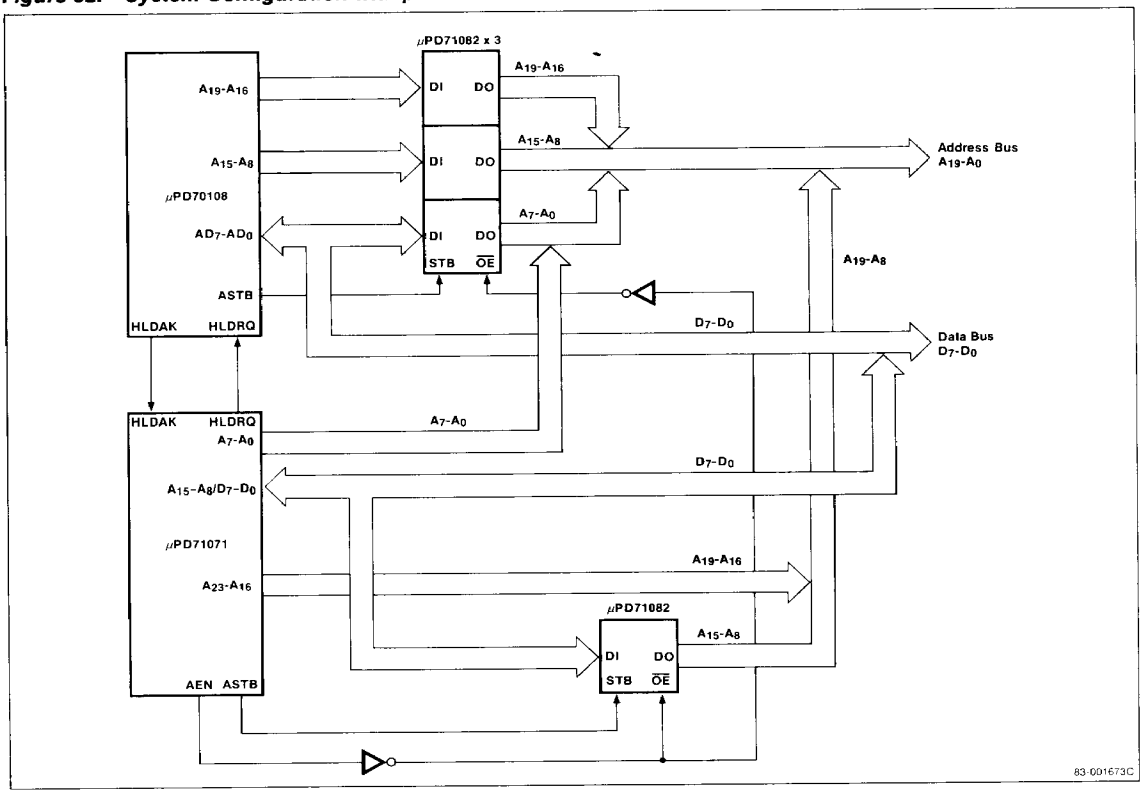
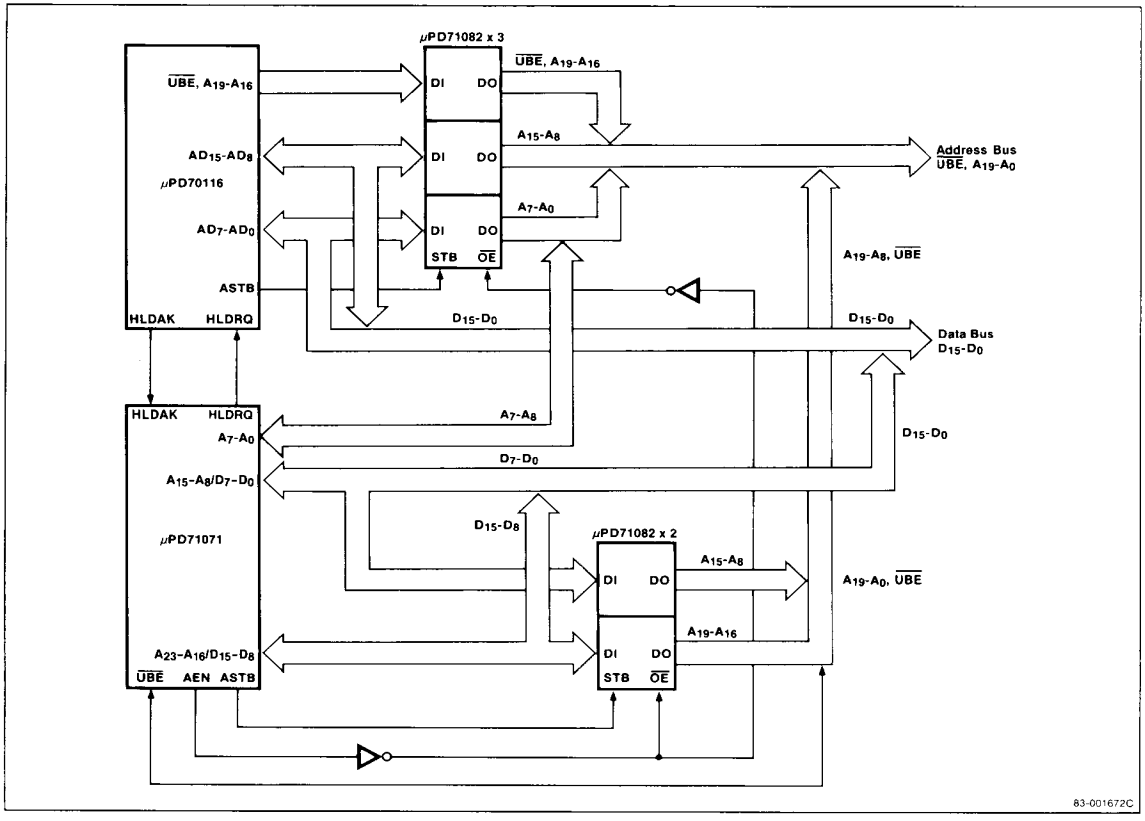


Figure 33. System Configuration with μPD70116



5g