DMA Controller

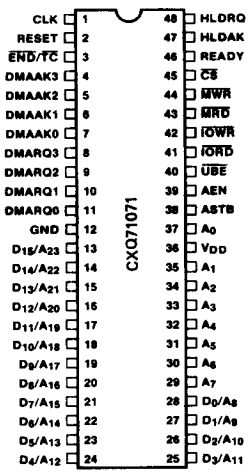
Description

The CXQ71071 is a high-speed, high-performance Direct Memory Access (DMA) controller that provides high speed data transfers between peripheral devices and memories. A programmable bus width allows bidirectional data transfer in both 8- and 16-bit systems. In addition, CMOS technology reduces power consumption.

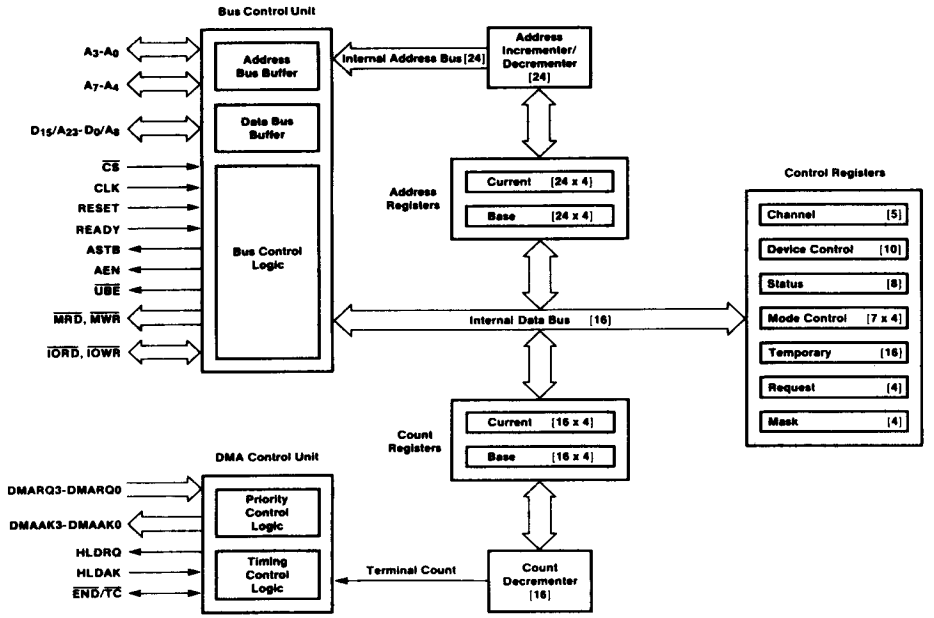
Features

- Four independent DMA channels
- 24 address lines to access 16M byte memory
- 64K byte/word transfer count
- 8- or 16-bit programmable data bus width
- Abundunt transfer functions
  - — Byte/word transfer
  - — Three transfer modes: single, demand, and block
  - — Two bus modes: bus release and bus hold
  - — Two transfer timings: normal and compressed
  - — Memory-to-memory, memory-to-I/O, or I/O-to-memory transfer
- Enable/disable of individual DMA requests
- Software DMA requests
- Autoinitialize enable/disable of individual DMA channels
- Address increment/decrement
- Fixed/rotational DMA channel priority
- Terminal count output
- Forced transfer termination input
- Cascade capability
- Programmable DMA request and acknowledge signal polarities
- High performance: transfers up to 5.33M bytes/second
- 8 MHz operation
- CXQ70108/70116 CPU system compatible
- CMOS technology
- Single power supply
- 48-pin plastic/ceramic DIP
- NEC μPD71071 compatible

Pin Configuration



Block Diagram



Pin Functions

No.	Symbol	Name	Type	Function
1	**CLK**	Clock	Input	Controls internal operation and data transfer rate.
2	**RESET**	Reset	Input	Initializes the controller's internal registers and enters the controller in the Idle cycle (CPU controls the bus). Active high.
3	**$\overline{END}$/$\overline{TC}$**	End DMA Transfer/ Terminal Count	Input/Output	The DMA service can be terminated either internally or externally. The CXQ71071 allows an external signal to end a DMA service by pulling the $\overline{END}$ low. The CXQ71071 also generates a pulse ($\overline{TC}$) when the designated terminal count for any channel has been reached. Open drain, it requires external pull-up resistor. Active low.
4–7	**DMAAK3–DMAAK0**	DMA Acknowledge	Output	Indicate to peripheral devices that DMA service has been acknowledged. DMAAK3–DMAAK0 respond respectively to DMA Channels 3–0. The sense of these line is programmable.
8–11	**DMARQ3–DMARQ0**	DMA Request	Input	Accept DMA service requests from peripheral devices. DMARQ3–DMARQ0 respond respectively to DMA Channels 3–0. The sense of these lines is programmable.
12	**GND**	Ground		Ground
13–20	**A23–A16/D15–D8**	Address/ Data	Input/Output 3-State	In 8-bit systems, these pins output the upper 8 bits of the address. In 16-bit systems, the address is multiplexed with the upper byte of the data bus. In the Idle mode, these pins become data bus to communicate with the CPU. In the DMA cycles, these lines become address/data bus.
21–28	**A15–A8/D7–D0**	Address/ Data	Input/Output 3-state	These pins output the middle 8 bits of the address multiplexed with the lower byte of the data bus. In the Idle mode, these lines function as data bus. In the DMA cycle, they become address/data bus.

No.	Symbol	Name	Type	Function
29–35, 37	A_7–A_0	Address	Input/Output 3-state	Function as the lower 8 bits of the address bus. A_7–A_4 output memory address during the DMA cycle and become high impedance in the Idle cycle. A_3–A_0 function as the lower 4 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 address.
36	V_{DD}	Power Supply		+5V power supply.
38	ASTB	Address Strobe	Output	Latches address information into an external address latch on the falling edge of ASTB during a DMA cycle. Latches A_{23}–A_8 in 16-bit mode and A_{15}–A_8 in 8-bit mode. Active high.
39	AEN	Address Enable	Output	Enables the output of an external latch that holds DMA addresses. AEN becomes high during the DMA cycle. Active high.
40	$\overline{UBE}$	Upper Byte Enable	Input/Output 3-state	Indicates the upper byte of the data bus is valid during 16-bit mode. In the Idle cycle during data transfer, it receives $\overline{UBE}$ from the CPU when the upper data byte is valid. During a DMA cycle, $\overline{UBE}$ 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. Active low.
41	$\overline{IORD}$	I/O Read	Input/Output	In the Idle cycle, it inputs a read signal from the CPU. In the DMA cycle, it outputs a read signal to I/O devices. Active low.
42	$\overline{IOWR}$	I/O Write	Input/Output	In the Idle cycle, it inputs a write signal from the CPU. In the DMA cycle, it outputs a write signal to I/O devices. Active low.
43	$\overline{MRD}$	Memory Read	Output 3-state	During the DMA cycle, it outputs a read signal to memory. $\overline{MRD}$ is high impedance during the Idle cycle. Active low.
44	$\overline{MWR}$	Memory Write	Output 3-state	During the DMA cycle, it outputs a write signal to memory. $\overline{MWR}$ is high impedance during the Idle cycle. Active low.

No.	Symbol	Name	Type	Function
45	$\overline{CS}$	Chip Select	Input	During the Idle cycle, it selects the CXQ71071 as an I/O device. Active low.
46	READY	Ready	Input	During a DMA operation, it 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 and memory, the CXQ71071 inserts a wait state (if READY is low), to extend the bus cycle until READY becomes high. Active high.
47	HLDAK	Hold Acknowledge	Input	Accepts the bus hold acknowledge signal from the CPU. Active high.
48	HLDRQ	Hold Request	Output	Outputs a bus hold request to the CPU. Active high.

Absolute Maximum Ratings (Ta=25°C)

Parameter	Symbol	Rating Value	Unit
Voltage on any pin with respect to Ground	V	−0.5 to +7.0	V
Operating temperature	Topr	−40 to +85	°C
Storage temperature	Tstg	−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 reliablility.

DC Characteristics (Ta=−40 to +85°C, V_{DD}=5V±10%)

Parameter	Symbol	Limits		Unit	Test Conditions
		Min.	Max.		
Input high voltage	V_{IH}	3.3	V_{DD}+0.5	V	CLK input pin
		2.2	V_{DD}+0.5	V	Other inputs
Input low voltage	V_{IL}	−0.5	0.8	V	
Output high voltage	V_{OH}	0.7V_{DD}		V	I_{OH}=−400μA
Output low voltage	V_{OL}		0.4	V	I_{OL}=2.5mA
Input leakage current	I_{LI}		±10	μA	0V≤V_I≤V_{DD}
Output leakage currnet	I_{LO}		±10	μA	0V≤V_O≤V_{DD}
Supply current (dynamic)	I_{DD1}		30	mA	
Supply current (static)	I_{DD2}		100	μA	

Capacitance (Ta=25°C)

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

AC Characteristics (DMA Mode)

(Ta=−40 to −85°C, V_{DD}=5V±10%)

Parameter	Symbol	Limits		Unit	Test Conditions
		Min.	Max.		
Clock cycle	t_{CYK}	125		ns	
Clock pulse width high	t_{KKH}	44		ns	
Clock pulse width low	t_{KKL}	55		ns	
Clock rise time	t_{KR}		10	ns	1.5V→3.0V
Clock fall time	t_{KF}		10	ns	3.0V→1.5V
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}	35		ns	SI, S0, S3, SW, S4w
HLDRQ high delay time from CLK low	t_{DHQH}		100	ns	SI, S4w
HLDRQ low delay time from CLK low	t_{DHQL}		100	ns	SI, S0, S4w
HLDRQ low level period	t_{HQHQL}	$2t_{CYK}-50$		ns	S4w
HLDAK high setup time to CLK low	t_{SHA}	35		ns	S0, S4, S4w
AEN high delay time from CLK low	t_{DAEH}		90	ns	S1, S2
AEN low delay time from CLK low	t_{DAEL}		90	ns	SI, S4w
ASTB high delay time from CLK low	t_{DSTH}		70	ns	S1
ASTB low delay time from CLK high	t_{DSTL}		70	ns	S1
ASTB high level period	t_{STSTH}	$t_{KKL}-15$		ns	
ADR/$\overline{UBE}$/$\overline{RD}$/$\overline{WR}$ active delay time from CLK low	t_{DA}		100	ns	S1, S2
ADR/$\overline{UBE}$/$\overline{RD}$/$\overline{WR}$ float time from CLK low	t_{FA1}		70	ns	SI, S4w
ADR setup time to ASTB low	t_{SAST}	$t_{KKL}-50$		ns	
ADR hold time from ASTB low	t_{HSTA}	$t_{KKH}-20$		ns	
ADR/$\overline{UBE}$ hold time from CLK high	t_{HA}	10		ns	S2, S4
ADR float time from CLK low	t_{FA2}	0	70	ns	S1, S2
$\overline{RD}$ low delay time from ADR float	t_{DAR}	−10		ns	

Parameter	Symbol	Limits		Unit	Test Conditions
		Min.	Max.		
Input data setup time to CLK low	t_{SID}	35		ns	S14
Input data hold time from CLK low	t_{HID}	10		ns	S14
Output data delay time from CLK low	t_{DOD}	10	100	ns	S22
Output data hold time from CLK high	t_{HOD}	10		ns	S24
Output data hold time from $\overline{MWR}$ high	t_{HMWOD}	$t_{KKL}-50$		ns	
$\overline{RD}$ low delay time from CLK low	t_{DKLR}		70	ns	S2 Normal Timing
$\overline{RD}$ low delay time from CLK high	t_{DKHR}		70	ns	S2 Compressed Timing
$\overline{RD}$ low level period	t_{RRL1}	$2t_{CYK}-50$		ns	Normal Timing
	t_{RRL2}	$t_{CYK}+t_{KKH}-50$			Compressed Timing
$\overline{RD}$ high delay time from CLK low	t_{DRH}	15	100	ns	S4
ADR delay time from $\overline{RD}$ high	t_{DRA}	$t_{CYK}-40$		ns	
$\overline{WR}$ low delay time from CLK low	t_{DWL1}	10	70	ns	S3 Normal Write
	t_{DWL2}	10	70	ns	S2 Extended Write, Normal Timing
$\overline{WR}$ low delay time from CLK high	t_{DWL3}	10	70	ns	S2 Compressed Timing
$\overline{WR}$ low level period	t_{WWL1}	$t_{CYK}-50$		ns	Normal Write
	t_{WWL2}	$2t_{CYK}-50$		ns	Extended Write, Normal Timing
	t_{WWL3}	$t_{CYK}+t_{KKH}-50$		ns	Extended Write, Compressed Timing
WR high delay time from CLK low	t_{DWH}	10	80	ns	S4
DMAAK setup time to $\overline{RD}$, $\overline{WR}$ low	t_{SDARW}	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}	10	70	ns	S1

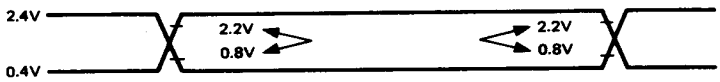
Parameter	Symbol	Limits		Unit	Test Conditions
		Min.	Max.		
DMAAK delay time from CLK low	t_{DKLDA}	10	115	ns	S1 Cascade Mode
DMAAK inactive delay time from CLK high	t_{DDAI1}	10	70	ns	S4
DMAAK inactive delay time from HLDAK low	t_{DDAI2}	5	−80	ns	S4 Cascade Mode
$\overline{TC}$ low delay time from CLK high	t_{DTCL}		100	ns	S3
$\overline{TC}$ off delay time from CLK high	t_{DTCF}		40	ns	S4
$\overline{TC}$ high delay time from CLK high	t_{DTCH}		$t_{KKH}+t_{CYK}-10$	ns	0V→2.2V*1
$\overline{TC}$ low level period	t_{TCTCL}	$t_{CYK}-15$		ns	
$\overline{END}$ low setup time to CLK high	t_{SED}	35		ns	S2
$\overline{END}$ low level period	t_{EDEDL}	100		ns	
READY setup time to CLK high	t_{SRY}	35		ns	S3, SW
READY hold time from CLK high	t_{HRY}	20		ns	S3, SW

Notes: *1 $\overline{END}/\overline{TC}$ has a 75 pF maximum input capacitance. To meet t_{DTCH}, use a 2.2K ohm or greater pull-up resistor with a load capacitance of 75 pF. The maximum output load capacitance for pins other than $\overline{END}/\overline{TC}$ is 100 pF.

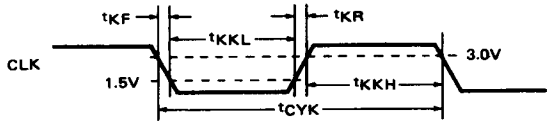
AC Characteristics (Programming Mode and RESET)

Parameter	Symbol	Limits		Unit	Test Conditions
		Min.	Max.		
$\overline{IOWR}$ low level period	t_{IWIWL}	100		ns	
$\overline{CS}$ low setup time to $\overline{IOWR}$ high	t_{SCSIW}	100		ns	
$\overline{CS}$ hold time from $\overline{IOWR}$ high	t_{HIWCS}	0		ns	
ADR/$\overline{UBE}$ setup time to $\overline{IOWR}$ high	t_{SAIW}	100		ns	
ADR/$\overline{UBE}$ hold time from $\overline{IOWR}$ high	t_{HIWA}	0		ns	
Input data setup time to $\overline{IOWR}$ high	t_{SIDIW}	100		ns	
Input data hold time from $\overline{IOWR}$ high	t_{HIWID}	0		ns	
$\overline{IORD}$ low level period	t_{IRIRL}	150		ns	
ADR/$\overline{CS}$ setup time to $\overline{IORD}$ low	t_{SAIR}	35		ns	
ADR/$\overline{CS}$ hold time from $\overline{IORD}$ high	t_{HIRA}	0		ns	
Output data delay time from $\overline{IORD}$ low	t_{DIROD}		120	ns	
Output data float time from $\overline{IORD}$ high	t_{FIROD}		100	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}	200		ns	

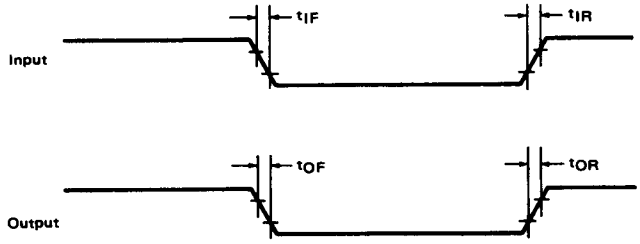
AC Test Waveforms



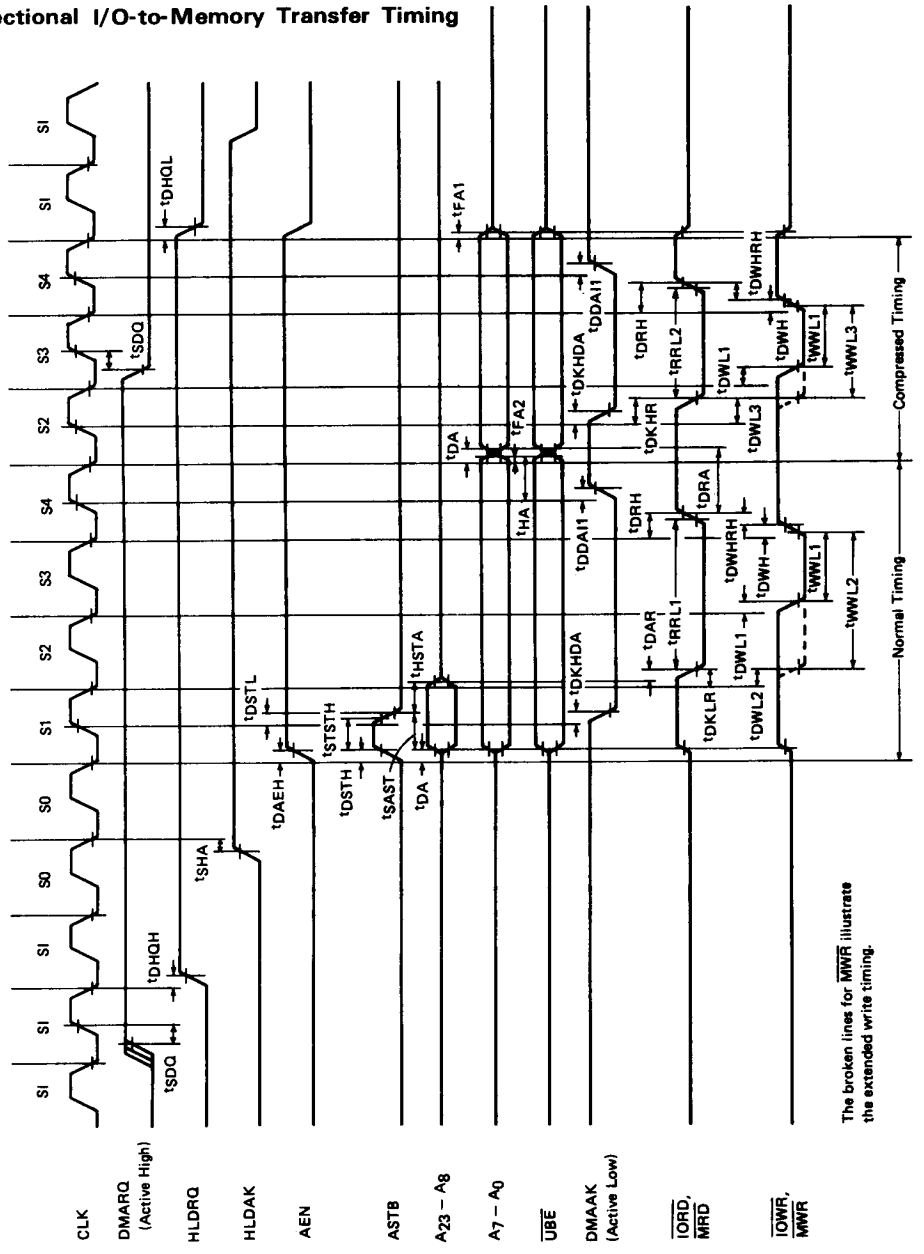
Clock Timing



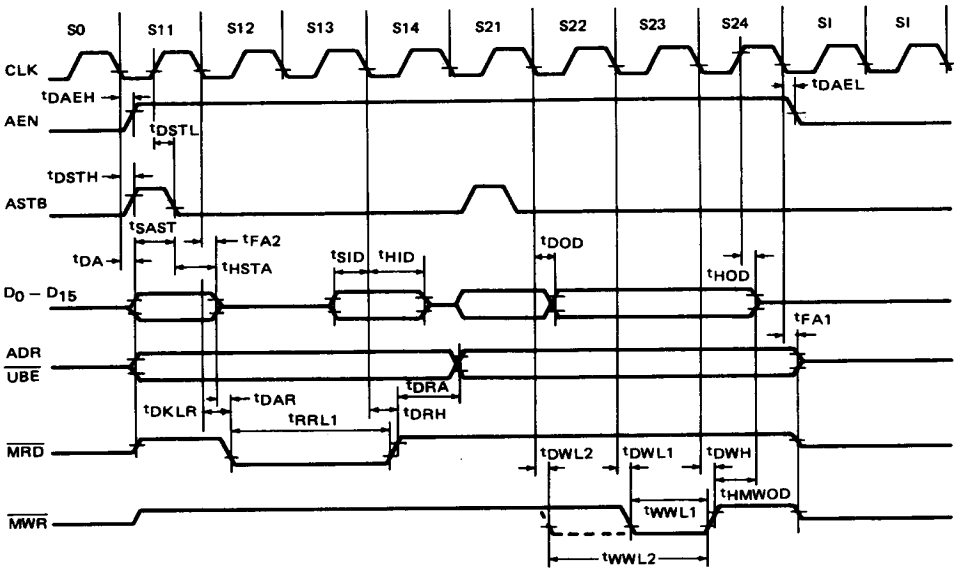
Input/Output Edge Timing



Directional I/O-to-Memory Transfer Timing

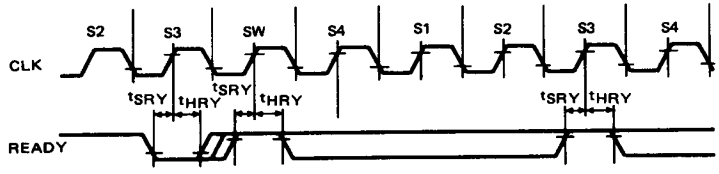


Memory-to-Memory Transfer Timing

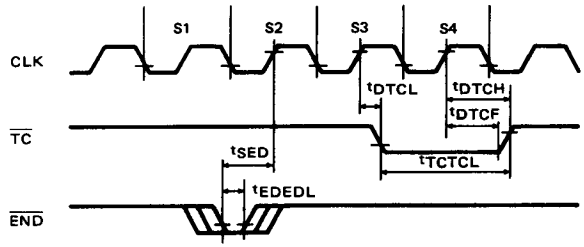


The broken line for $\overline{MWR}$ illustrates the extended write timing.

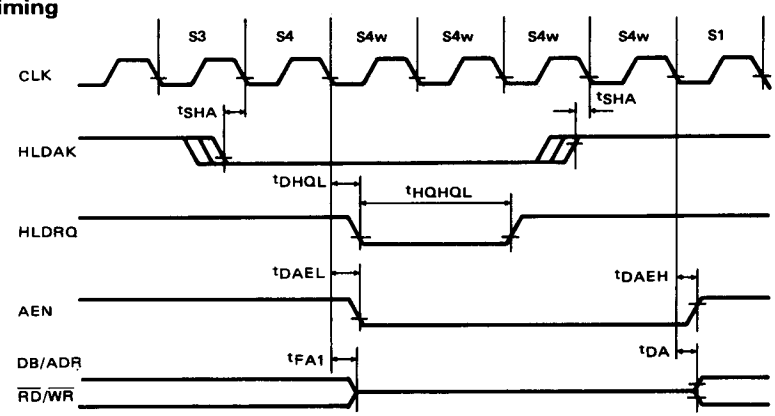
Ready Timing



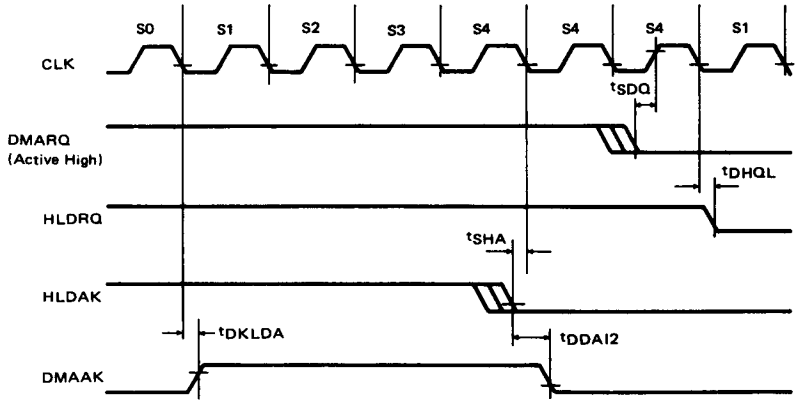
$\overline{END}/\overline{TC}$ Timing



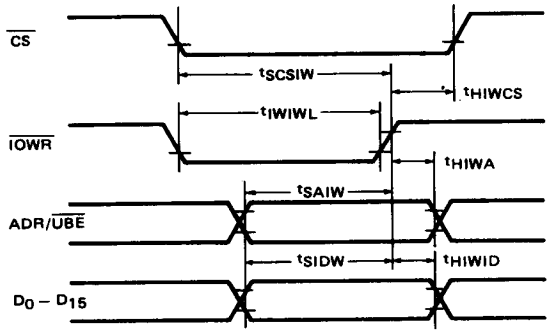
Bus Wait Timing

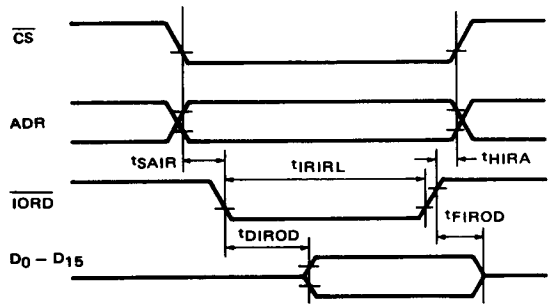


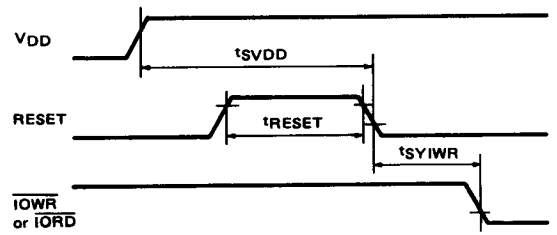
Cascade Timing

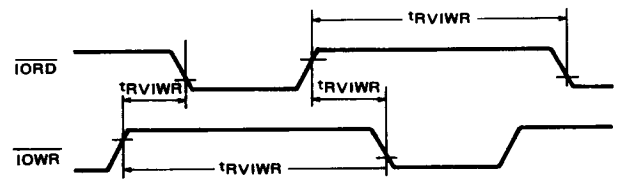


Programming Mode and RESET Timing









Functional Description

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 address 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 system bus according to the priority level. The timing control logic provides internal timing and controls DMA operations.

Address Registers

Each DMA channel has one 24-bit Base Address Register and one 24-bit Current Address Register. The Base Address Registers hold the value written by the CPU and transfer the value to the corresponding Current Address Register during autoinitialization (address and count are automatically initialized). The Current Address Register is automatically incremented/decremented for each transfer and always contains the address to be transferred next.

Address Incrementer/Decrementer

The Address Incrementer/Decrementer updates the contents of the Current Address Register whenever a DMA transfer for one bus cycle has finished.

Count Registers

Each DMA channel has one 16-bit Base Count Register and one 16-bit Current Count Register. The Base Count Register holds the value written by the CPU and transfers the value to the corresponding Current Count Register during autoinitialization. The Current Count Register is automatically decremented for each transfer and generates a terminal count when it reaches zero.

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 has finished.

Contol Registers

The CXQ71071 contains the following control registers:

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

These registers control bus mode, pin active level, DMA operation mode, mask bits, and other CXQ71071 operating functions.

DMA Operation

The CXQ71071 operates in two cycles: Idle and DMA. In an Idle cycle, the CPU uses the system bus, while in a DMA cycle, the CXQ71071 uses it.

Idle Cycle

In an Idle cycle, there are no DMA requests active or there are one or more active DMA requests, but the CPU has not released the bus. The CXQ71071 will sample the four DMARQ inputs every clock cycle to determine if any channel is requesting a DMA service. If one or more inputs are active, the corresponding DMA request bits (RQ) are set and the CXQ71071 will output a bus hold request (HLDRQ) to the CPU. The CXQ71071 continues to sample DMA requests until it obtains the bus.

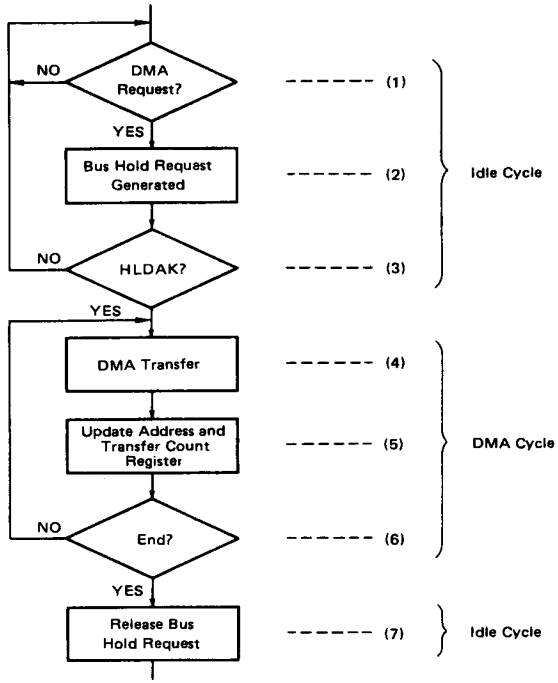
After the CPU returns a HLDAK signal and the CXQ71071 obtains the bus, the CXQ71071 stops DMA sampling and selects the highest priority channel from the valid DMA request signals.

The CXQ71071 will also sample $\overline{CS}$ pin, checking an attempt by the CPU to read or write the internal registers of the device. Address lines A0-A3 select which registers will be read or written and $\overline{IORD}$ and $\overline{IOWR}$ are used to select reading or writing.

DMA Cycle

In a DMA cycle, the CXQ71071 controls the bus in order to perform DMA transfer operations based on the programmed information. Figure 1 outlines the sequential flow of a DMA operation.

Figure 1. DMA Operation Flow



Data Bus Width

In order to allow an easy interface with an 8- or 16-bit CPU, the data bus width of the CXQ71071 is user programmable for 8 or 16 bits. A 16-bit data bus allows access to the 16-bit internal registers in one I/O bus cycle.

The initial bus width after reset is set to 8 bits.

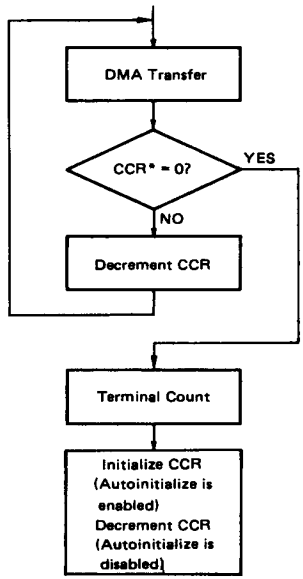
The following table shows the relationship of the data bus width, A_0, $\overline{UBE}$, and the internal registers.

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

Terminal Count

The CXQ71071 ends DMA service when it internally generates a terminal count ($\overline{TC}$) or when $\overline{END}$ externally becomes active. A terminal count is produced when the contents of the Current Count Register become 0 and output a low level pulse to the $\overline{TC}$ pin. Figure 2 shows the relationship between the generation of the Terminal Count and the Current Count Register. The Current Count Register is tested after each DMA transfer and prior to decrementing it so that the DMA transfer is actually performed one more than the programmed value of the Current Count Register.

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



*CCR: Current Count Register

Unless a channel is programmed for autoinitialize when DMA service ends, the corresponding bit of the Mask Register is set, and the DMARQ input of the channel is masked.

DMA Transfer Type

The type of transfer the CXQ71071 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 CXQ71071 can perform memory-to-I/O transfer (one transfer in one bus cycle) and memory-to-memory transfer (one transfer in two bus cycles).

Memory-to-memory transfer is enabled, when bit 0 of the Device Control Register is set to 1. The DMA channel used for memory-to-memory transfer is fixed, with Channel 0 as the source channel and Channel 1 as the destination channel. For this reason, 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. When DMARQ0 (Channel 0) becomes active by software, the transfer is initiated. The CXQ71071 performs the following operations until a Channel 1 terminal count or $\overline{END}$ input is present:

- The memory data pointed to by the Current Address Register of Channel 0 is read into the temporary register and the address and count of Channel 0 are updated.
- 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 while memory-to-memory transfer is enabled, the CXQ71071 will perform memory-to-I/O transfer. Since this may cause erroneous memory-to-memory transfers, bit 1 of the Mask Register should be set to 1.

During the memory-to-memory transfers, the source side (Channel 0) can be programmed to retain the same address by setting bit 1 of the Device Control Register to 1. In this manner, a range of memory can be initialized with the same value.

During memory-to-memory transfer, the DMAAK signals and Channel 0's terminal count (TC) pulse are not output.

Direction of Memory-to-I/O Transfers

All DMA transfers use memory as a reference point. A DMA Read reads data from memory and writes to an I/O port. A DMA Write reads data from an I/O port and writes to memory. In memory-to-I/O transfer, use the Mode Control Register to set one of the following transfer directions for each channel and activate the appropriate control signals.

Type	Transfer Direction	Activated Signals
DMA Read	Memory ⟶ I/O	$\overline{IOWR}$, $\overline{MRD}$
DMA Write	I/O ⟶ Memory	$\overline{IORD}$, $\overline{MWR}$
Verify	Verify transfer outputs address only and does not perform actual transfer	—

Transfer Modes

In a 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. The following table shows the various transfer modes and termination conditions:

Transfer Mode	Transfer End 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

Note: DMA transfer operations using memory-to-memory transfers are identical to the block transfer mode.

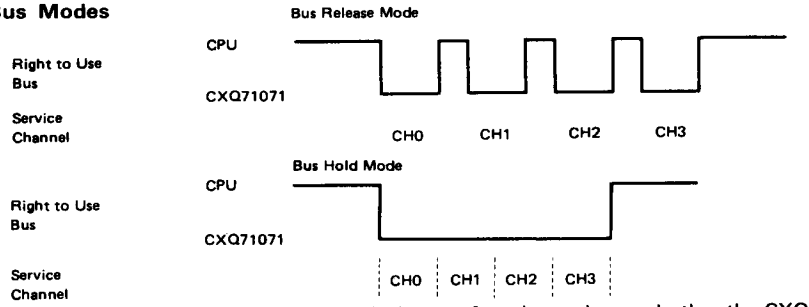
Bus Mode

The Device Control Register selects either the Bus Release or Bus Hold mode. The bus mode determines how the CXQ71071 returns the bus to the CPU.

Figure 3 shows that in Bus Release mode, only one channel can receive service for each DMA operation. Whenever DMA service terminates (transfer end conditions depend on the transfer mode), the channel returns the bus to the CPU (regardless of the state of other DMA requests) and the CXQ71071 enters the Idle cycle. When the CXQ71071 regains the bus, another DMA operation will begin.

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. Transfer end conditions depend on the transfer mode.

Figure 3. Bus Modes



The operation of Single, Demand, and Block Mode transfers depends on whether the CXQ71071 is in Bus Release or Bus Hold mode.

Single Mode Transfer

In Bus Release mode, when a channel completes the transfer of a single byte or word, the CXQ71071 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 CXQ71071 terminates the channel's service even if it is still asserting a DMA request signal. The CXQ71071 will then service the highest priority requesting channel. If there are no requests from any other channel, the CXQ71071 releases the bus and enters the Idle cycle.

Demand Mode Transfer

In Bus Release mode, the current 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 channel in service goes inactive, the CXQ71071 releases the bus to the CPU and enters the Idle state, even if the DMA requests from other channels are active.

In Bus Hold mode, when the active channel completes a single transfer, the CXQ71071 checks the other DMA request lines without ending the state of the current service. If there is a higher priority request, the CXQ71071 suspends servicing the current channel and starts servicing the highest priority channel requesting service, without releasing the bus. If there is no higher request than the current one, the CXQ71071 continues to service the currently active channel. Lower priority DMA requests are honored without releasing the bus after the current channel service is completed.

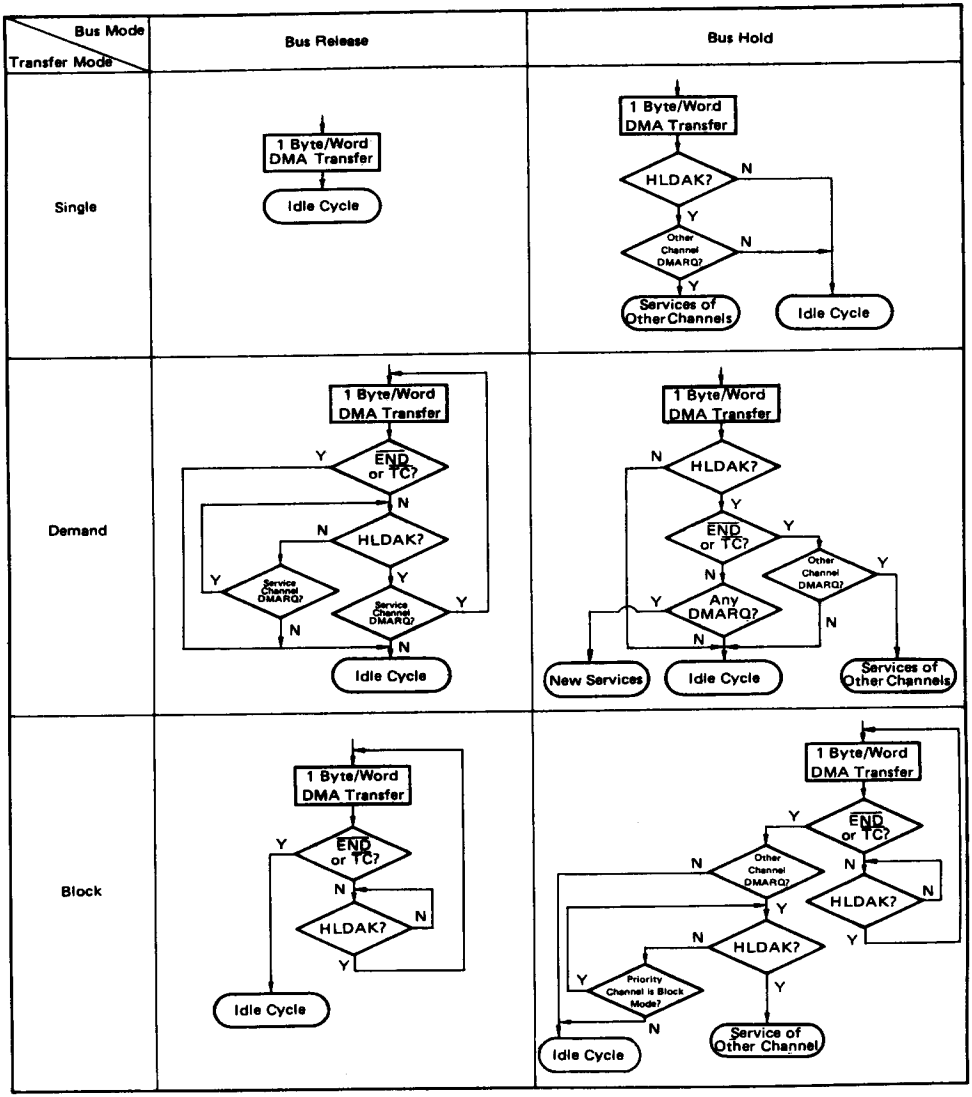
Block Mode Transfer

In Bus Release mode, the current channel continues data transfer until a terminal count or the external $\overline{\text{END}}$ signal becomes active. During this time, the CXQ71071 ignores all other DMA requests. After completion of the block transfer, the CXQ71071 releases the bus and enters the Idle cycle even if the DMA requests from other channels are active.

In Bus Hold mode, the current channel transfers data until an internal or external $\overline{\text{END}}$ signal becomes active. When the service is completed, the CXQ71071 checks all DMA requests without releasing the bus. If there is an active request, the CXQ71071 immediately begins servicing the request. The CXQ71071 releases the bus after it honors all DMA requests.

Figure 4 shows the operation flow for the six possible transfer and bus mode operations in DMA transfer.

Figure 4. Transfer and Bus Modes Operations



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. The following table shows the byte count by which the Address and Count Registers are incremented or decremented during byte/word transfer.

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 the initial value of the programmed address is odd, transfer is started after decrementing the address by 1. For this reason, always select even addresses as the initial value to avoid destroying data. Byte and word transfers are controlled by the A_0 and $\overline{UBE}$ signals.

The following table shows the relationship between the data bus width, A_0 and $\overline{UBE}$ signals, and data bus status.

Data Bus Width	A_0	$\overline{UBE}$	Data Bus Status
8 bits	X	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

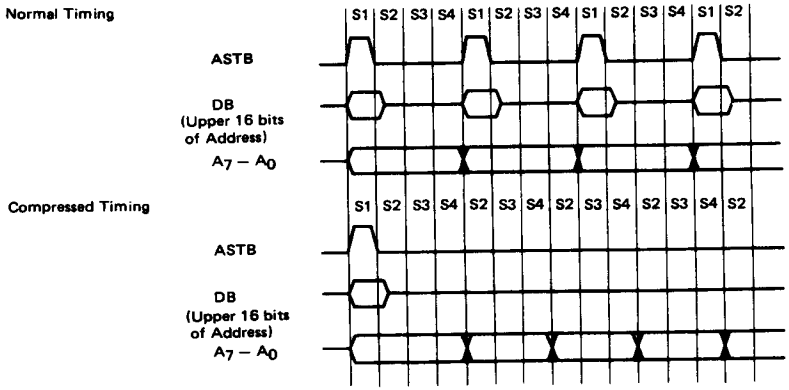
*Note: Always 1

Compressed Timing

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 three clock cycles. Compressed timing is only available in block mode (except memory-to-memory) and in demand mode during Bus Release mode, for 33% more efficiency.

The CXQ71071 is able to omit one clock during compressed timing by not updating the upper 16 bits of the latched address. In Block mode and Demand mode during bus release, 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) can be omitted in the bus cycles except during the first bus cycle and when the upper 16 bits of an address is changed. Figure 5 shows wave forms for normal and compressed timing.

Figure 5. Normal and Compressed Timing Waveforms



Software DMA Requests

The CXQ71071 can accept software DMA requests in addition to DMA requests from the four DMARQ pins. Software DMA requests are generated by setting the appropriate bit in the Request Register. Software DMA requests are not masked by the Mask Register and 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 will be serviced, and all bits of the Request Register will be cleared when the service is over. There may be a chance that other software DMA requests will be cancelled.

When Bus Hold mode is set, only the corresponding bit of the Request Register will be cleared after a DMA service is over. All software DMA requests will be serviced in the sequence of their priority level.

Precaution must be taken for software DMA requests for cascade channels (See the Cascade Connection) in Bus Hold mode. While a cascade channel is serviced, the master CXQ71071 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 completed and all bits of the Request Register are cleared, the cascade channel masks can be cleared.

Transfer Mode

When Single or Demand mode is programmed, the corresponding request bits are cleared and software DMA service ends with the transfer of one byte/word. When Block mode (memory-to-memory) is programmed, service continues until $\overline{\text{END}}$ is input or a terminal count is generated. The corresponding request bits are cleared when service ends.

Autoinitialize

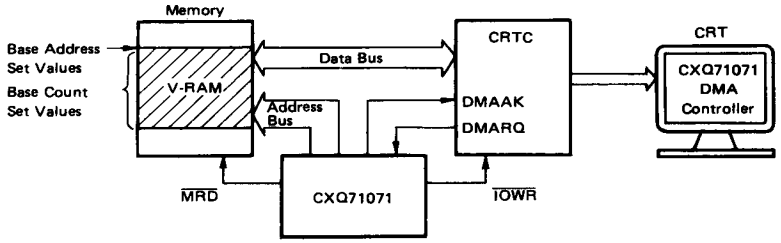
When the Mode Control Register selects autoinitialize for a channel, the CXQ71071 automatically initializes the address and count when $\overline{END}$ is input or a terminal count is generated. Then the contents of the Base Address and Base Count Registers are transferred to the Current Address and Current Count Registers, respectively. The corresponding bit of the Mask Register is cleared. However, the bit of the Mask Register is set for channels not programmed for autoinitialize.

Use the autoinitialize function for the following types of transfers:

Repetitive Input/Output of Memory Area

Figure 6A shows an example of DMA transfer between a CRT controller and memory. After setting the same value in the Base and Current Registers, autoinitialize allows repetitive DMA transfer without CPU involvement.

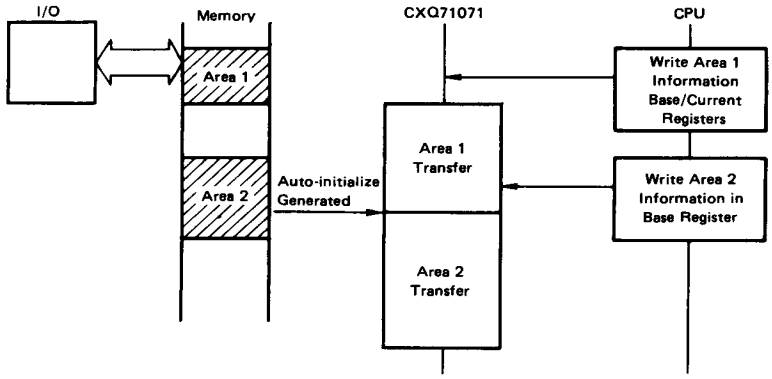
Figure 6A. Autoinitialize, Application 1



Continuous Transfer of Several Memory Areas

The CPU can write the CXQ71071 only to the Address/Count Base Registers for programming the address/count information. The autoinitialize function can be used to perform continuous transfer of several contiguous or non-contiguous memory areas during Single or Demand mode in the Bus Release mode. If the CPU sets information for Area 2 in Base Registers during the previous transfer of data for Area 1, the Current Registers will be automatically restored from the Base Registers following the generation of a terminal count. Figure 6B illustrates this procedure.

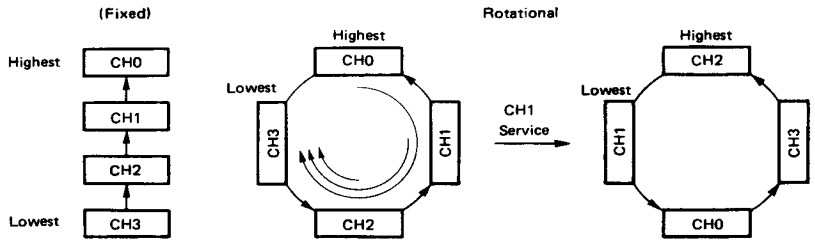
Figure 6B. Autoinitialize, Application 2



Channel Priority

Each of the CXQ71071'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 two types of priority encoding: fixed and rotational priority. In fixed priority, channel 0 is assigned the highest priority through channel 3 the lowest. In rotational priority, priority order is rotated so that the last channel to get service is assigned the lowest priority with the others rotating accordingly. This method prevents exclusive servicing of some channel (s). Figure 7 shows the two priority order methods.

Figure 7. Priority Order

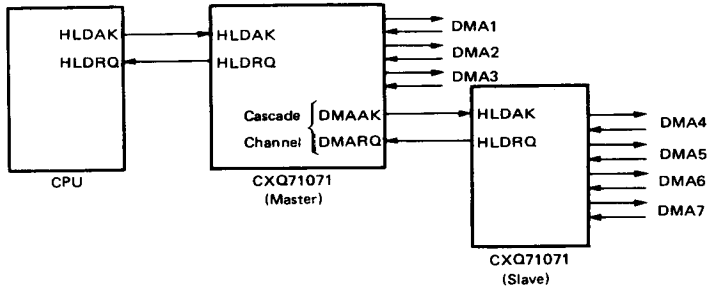


Cascade Connection

The CXQ71071 can be cascaded to expand the system DMA channel capacity. To connect a CXQ71071 for cascading (Figure 8),

1. Connect pins HLDRQ and HLDAK of the slave CXQ71071 to pins DMARQ and DMAAK of any channels of the master CXQ71071.
2. Set bits 7 and 6 of the Mode Control Register to 11 in order to select Cascade mode for the Master's channel.

Figure 8. Cascade Connection Example



In the master CXQ71071, DMARQ, DMAAK, HLDRQ, and HLDAK only become valid during the DMA service of the channel set to the Cascade mode. The other signals are disabled so as not to conflict with the outputs of the active channel in the slave CXQ71071. The master cascade channel only propagates hold request/hold acknowledge between the slave and CPU.

The master CXQ71071 always operates in the Bus Release mode while 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 CXQ71071 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 the other non-cascaded channels normally.

Bus Waiting Operation

In systems using a V40™/V50™ as the CPU, even during a DMA cycle, the on-chip refresh control unit in the CPU may lower the level of the HLDAK signal to inactive and use the bus. Therefore, the CXQ71071 automatically performs a bus waiting operation in a system that has a bus master whose priority level is higher than that of the CXQ71071.

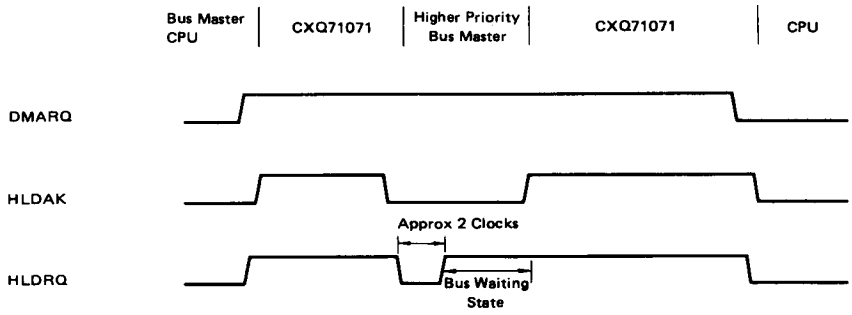
The CXQ71071 executes the bus waiting operation if the HLDAK signal becomes inactive during service in an operating mode where transfer is executed continuously during block transfer mode; during demand transfer mode in bus release mode; and during memory-to-memory transfer.

When HLDAK becomes inactive during service in other operating modes, the device returns to the Idle cycle and passes the control of the bus to the higher bus master.

Figure 9 shows that when the HLDAK signal becomes inactive during service for continuous transfer, the CXQ71071 is set up in an S4w state (bus waiting). Operation moves to an Idle cycle if DMARQ is inactive in the demand transfer mode.

The HLDRQ signal becomes 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.

Figure 9. Bus Waiting Operation



Programming the CXQ71071

To prepare a channel for DMA transfer, the following information must be defined:

- starting address for the transfer
- number of transfer cycles
- DMA operating modes
- data bus widths
- active levels of the DMARQ and DMAAK signals

The CXQ71071 contain 395 bits of internal memory in the form of registers. The address lines A_3—A_0 are used to address the register to be read or written.

The following tables show the register and command configurations.

Register Configuration

Register	Bit size	Number
Channel	5	1
Base Address	24	4
Current Address	24	4
Base Count	16	4
Current Count	16	4
Mode Control	7	4
Device Control	10	1
Status	8	1
Request	4	1
Mask	4	1
Temporary	16	1

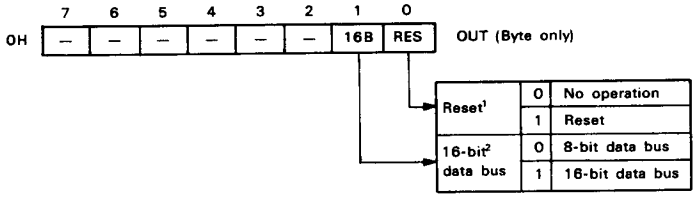
Note: When using a 16-bit CPU and selecting a 16-bit data bus, use the word IN/OUT instruction to read/write information two bytes at a time. However, the commands suffixed with "(B)" in the following table, must be issued with the byte IN/OUT instruction.

Command Configuration

Address	R/W	Command Name	Format MSB							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(B)	Status Register Read	RQ3	RQ2	RQ1	RQ0	TC3	TC2	TC1	TC0
0BH	R/W(B)	Mode Control Reg. Read/Write	TMODE		ADIR	AUTI	TDIR		—	W/B
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

Initialize

The following figure shows the CXQ71071 initialize process.



Notes: 1. The CXQ71071 initializes as follows:

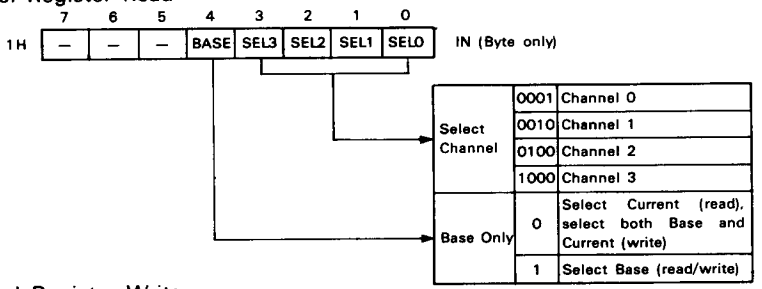
Register	Initialization Operation
Initialize	Clears all bits
Address	No change
Count	No change
Channel	Selects Channel 0
Mode Control	Clears all bits
Device Control	Clears all bits
Status	Clears all bits
Request	Clears all bits
Mask	Sets all bits (masks all channels)
Temporary	Clears all bits

2. When using the CXQ71071 in a 16-bit system, set this bit immediately after a hardware reset since the CXQ71071 always initializes in the 8-bit data bus mode.

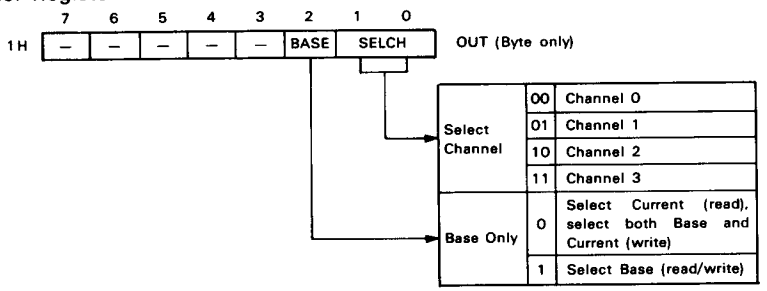
Channel Register

This command reads and writes the Channel Register that selects one of four DMA channels for programming by the Address, Count and Mode Control Registers. This command must be issued by the byte IN/OUT instruction.

Channel Register Read



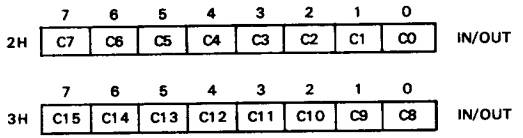
Channel Register Write



Count Register Read/Write

If bit 2 of the Channel Register 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 is set, a write to the Count Register only affects the Base Count Register.

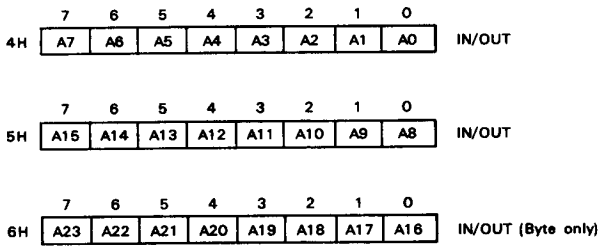
The Base Count Registers hold the initial count value until a new count is specified. If autoinitialization is enabled, this value is transferred to the Current Count Register when an $\overline{END}$ or $\overline{TC}$ is generated. For each DMA transfer, the Current Count Register is decremented by one.



Address Register Read/Write

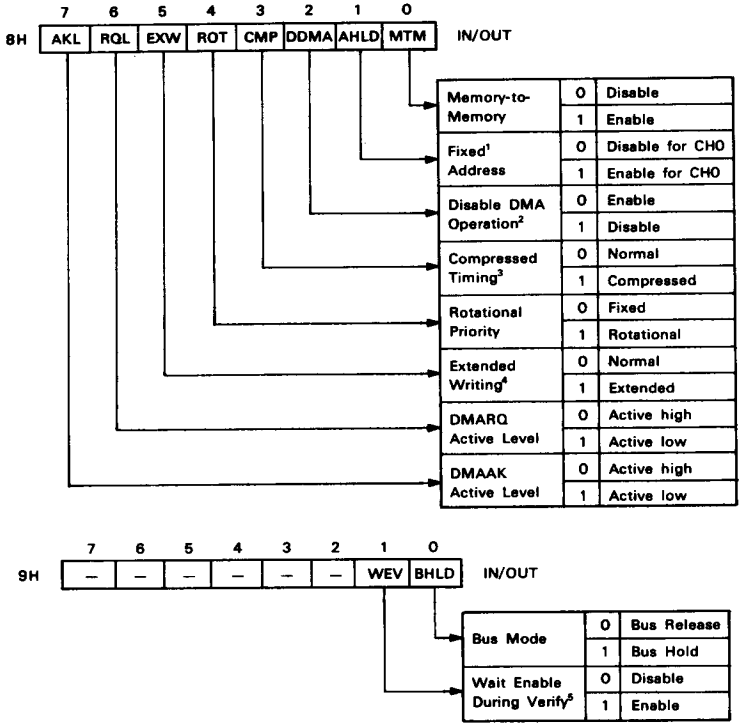
The word IN/OUT instruction is used to specify the lower two bytes (4H and 5H) of the register if a 16-bit data bus width is selected. The byte IN/OUT instruction must be used to specify the upper byte (6H) of the register. When bit 2 of the Channel Register is cleared, a write to the Address Register 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 value is specified and this value is transferred to the Current Address Register during autoinitialization. For each DMA transfer, the Current Address Register is incremented or decremented by two during word transfer and by one during byte transfer.



Device Control Register Read/Write

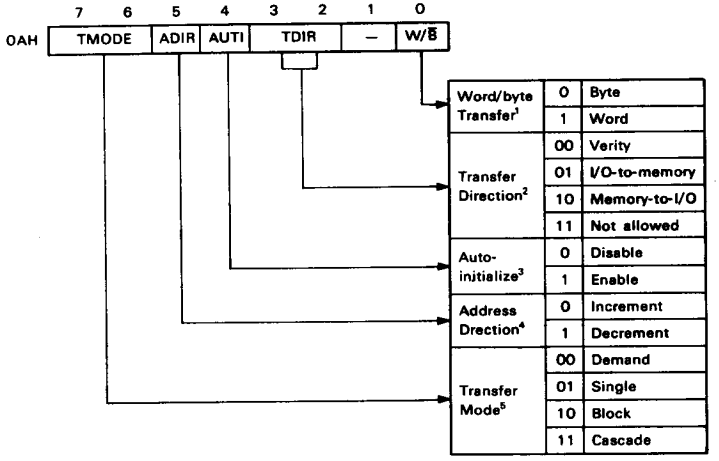
The Device Control Register Read/Write command reads from and writes to the Device Control Register. If using a 16-bit data bus, the word IN/OUT instruction is used to read and write 16-bit data.



Notes:
1. This bit is meaningless when MTM = 0.
2. Disables HLDRQ to the CPU to prevent incorrect DMA operation while the CXQ71071's registers are being initialized or modified.
3. Performs compressed timing DMA transfer in block or demand mode during bus release.
4. When EXW is 0, the write signal becomes active (normal write) during S3 and SW (see the timing waveforms). When 1, the write signal becomes active during S2, S3, and SW (like the read signal).
5. Wait states are generated by the READY signal during a verity transfer.

Mode Control Register Read/Write

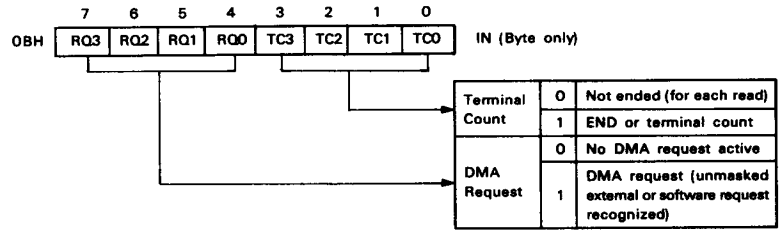
This command reads from and writes to the Mode Control Register to specify the operating mode for each channel. The Channel Register selects the Mode Control Register. This command must be issued by the byte IN/OUT instruction.



Notes:
1. When a 16-bit data bus is selected, this bit selects DMA transfer by word or byte.
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.

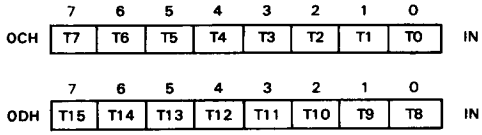
Status Register Read

This command reads the Status Register for the individual DMA channel that has DMA request states and terminal count or END information. This command must be issued by the byte IN instruction.



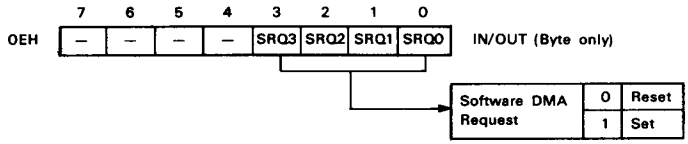
Temporary Register Read

If a 16-bit data bus is selected, the word IN instruction is used to read 16-bit data with this command. The last data transferred in memory-to-memory transfer is stored in the temporary register. If an 8-bit data bus is selected, the value of the upper byte becomes undefined.



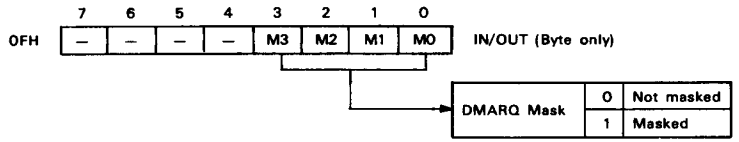
Request Register Read/Write

This command reads from and writes to the Request Register to generate DMA requests by software for the four corresponding DMA channels. This command must be issued by the byte IN/OUT instruction.



Mask Register Read/Write

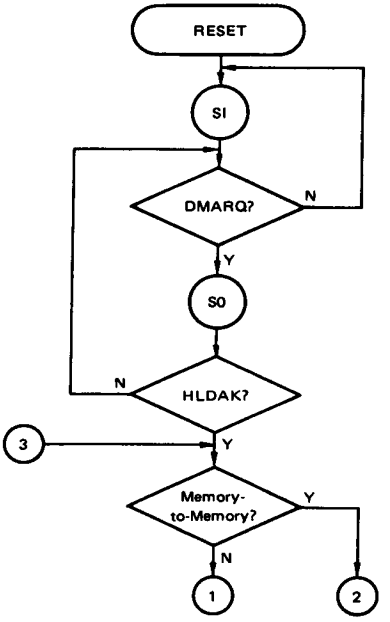
This command read from and writes to the Mask Register to control DMA request for the corresponding four DMA channels by DMARQ3—DMARQ0. This command must be issued by the byte IN/OUT instruction.



DMA Transfer Modes

Figures 10 through 15 show state transition diagrams for the different modes of DMA transfer.

Figure 10. Idle Cycle



SI: DMA request idle cycle
S0: HLDAK wait state
S1: Address latch state
S2: Read signal output state
S3: Write signal output state
S4: Read/Write signal recovery state
SW: READY wait state
S4w: Bus wait state

1 : Memory-I/O Transfer
2 : Memory-to-Memory Transfer

Figure 11. DMA Cycle, Cascade Mode

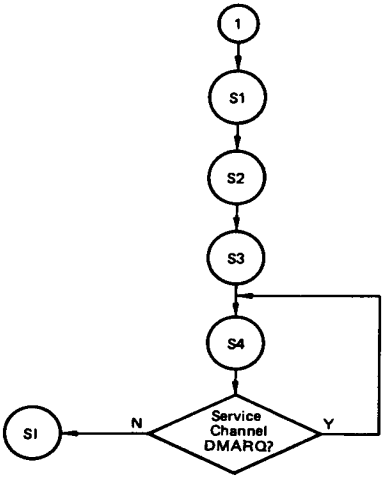


Figure 12. DMA Cycle, Single Mode

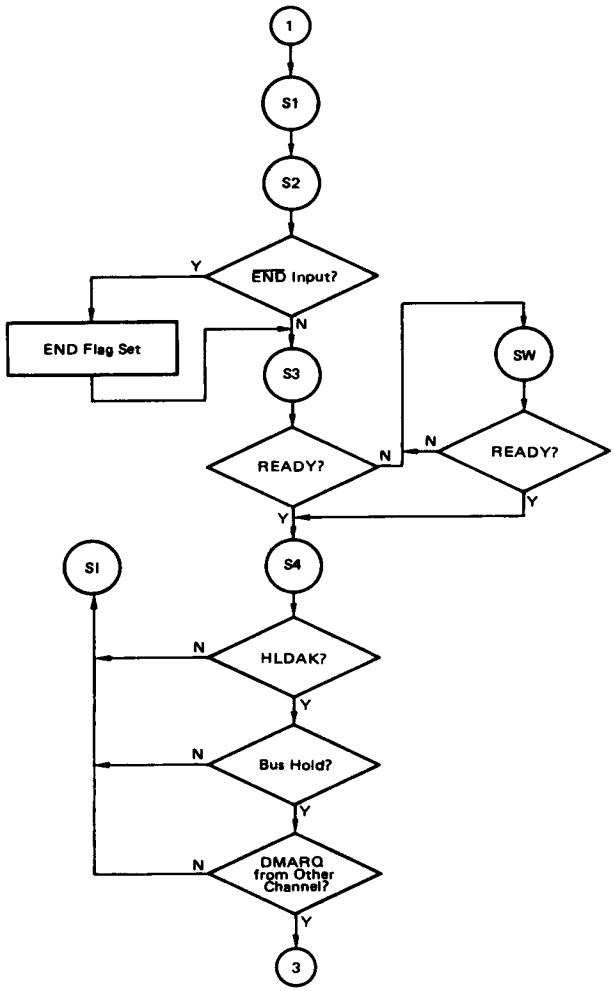
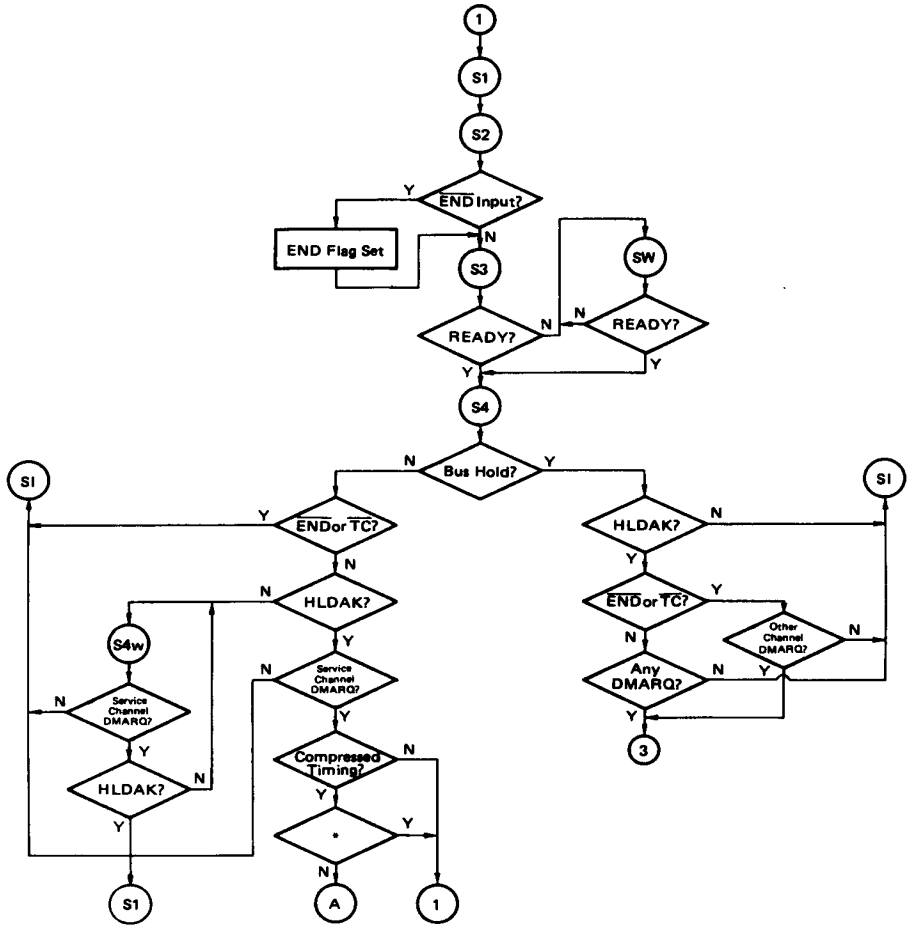


Figure 13. DMA Cycle, Demand Mode



* Carry or Borrow to Upper Two-Bytes of Address?

Figure 14. DMA Cycle, Block Mode

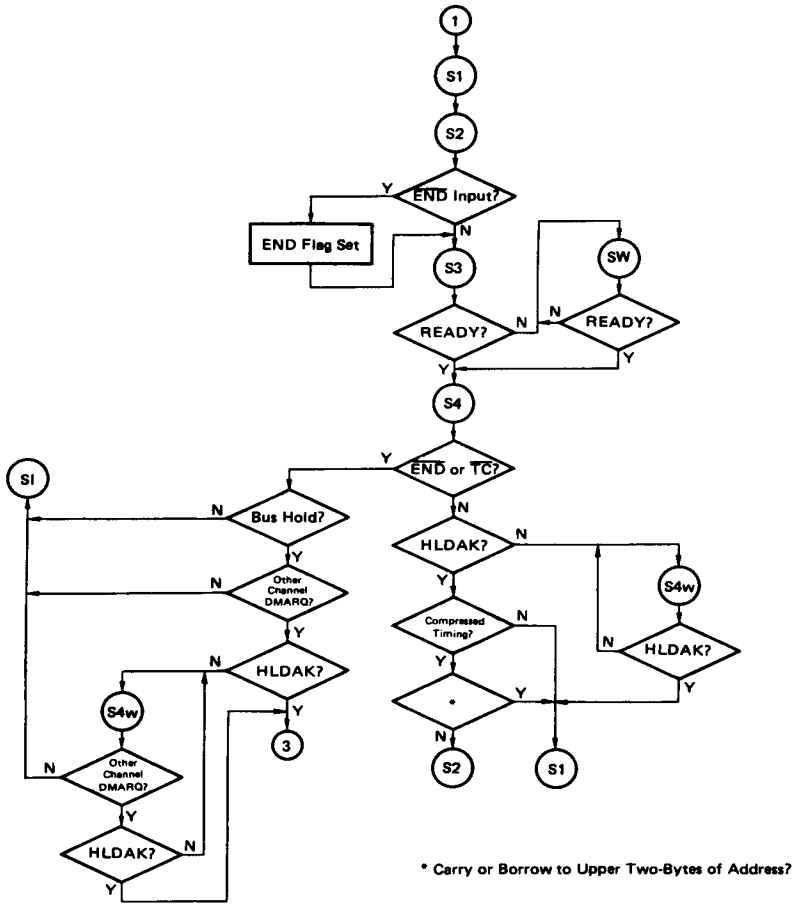
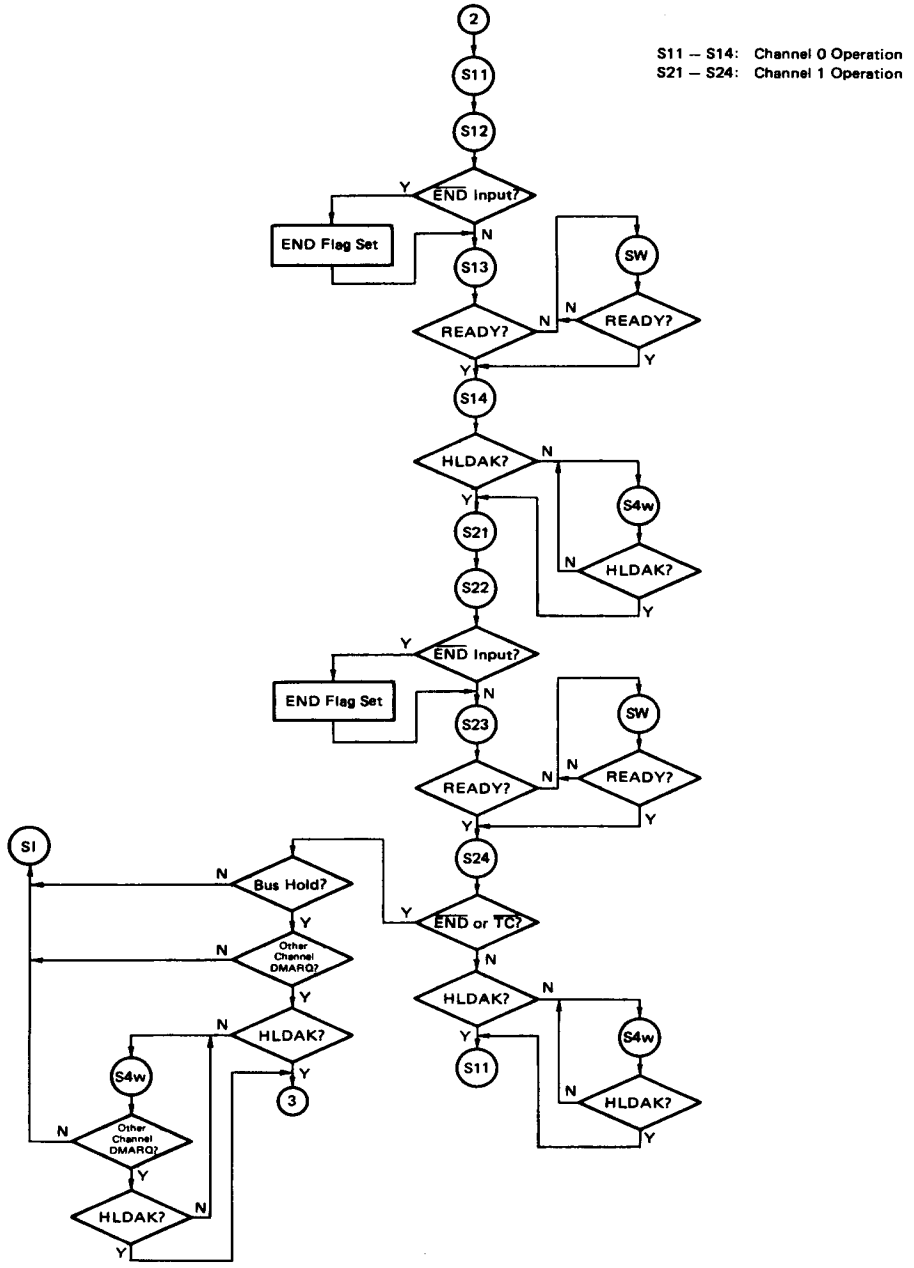
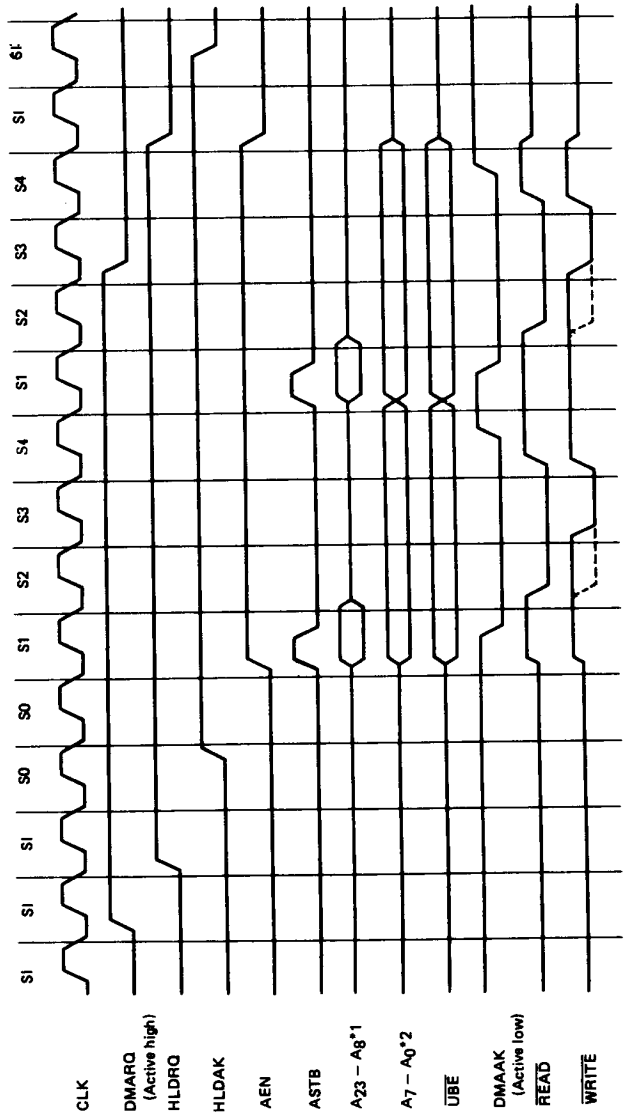


Figure 15. DMA Cycle, Memory-to-Memory Transfer



Transfer Timing

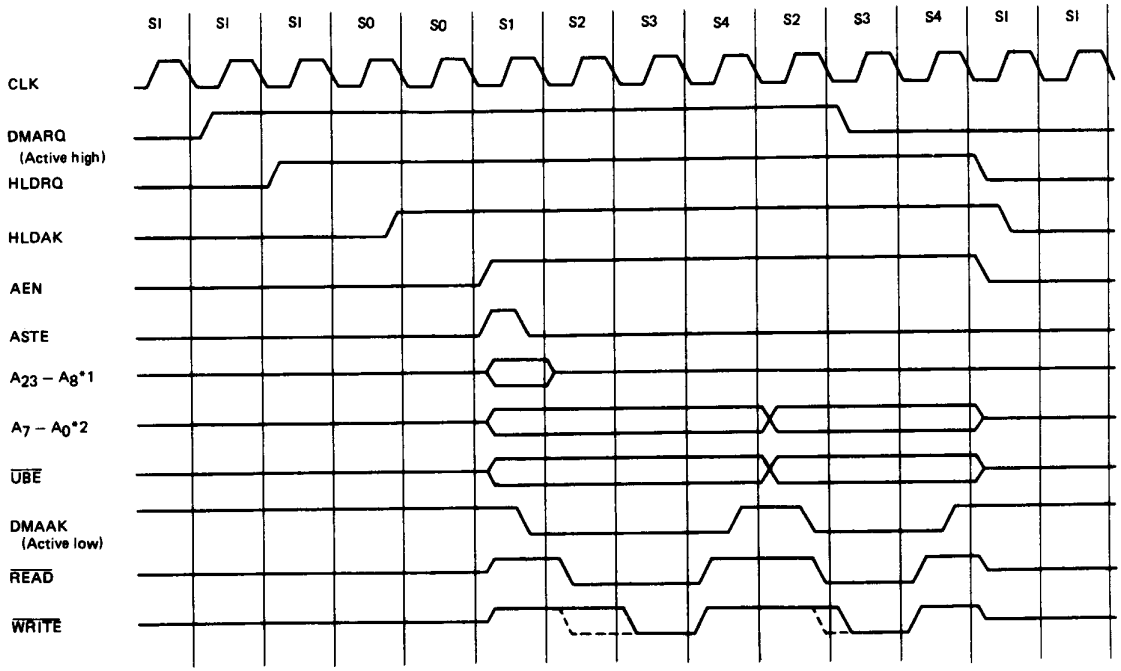
Figures 16 through 18 show CXQ71071 timing waveforms.



Figures 16. Memory-I/O Transfer, Normal Timing

Notes: If an 8-bit data bus is selected,
1: A_{15}—A_8
2: A_{23}—A_{16}, A_7—A_0
The broken lines of the $\overline{WRITE}$ signal are for extended write timing.

Figure 17. Memory-I/O Transfer, Compressed Timing



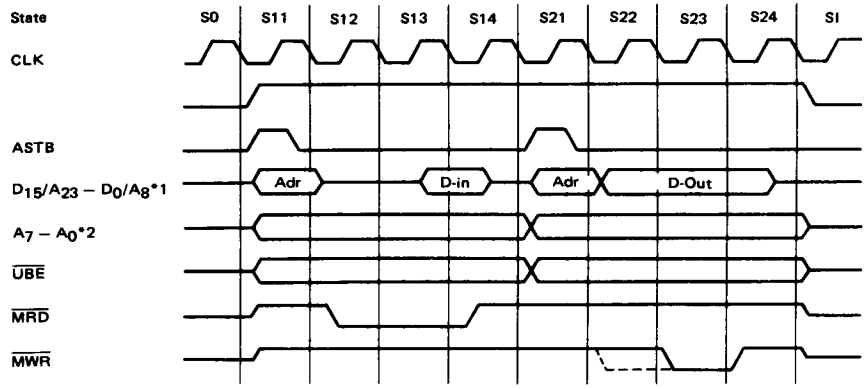
Notes: If an 8-bit data bus is selected,

1: A_{15}—A_8

2: A_{23}—A_{16}, A_7—A_0

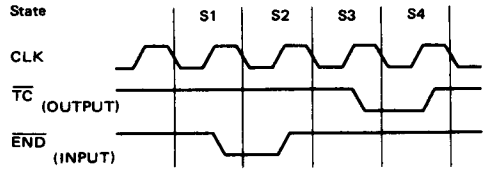
The broken lines of the $\overline{\text{WRITE}}$ signal are for extended write timing.

Figure 18. Memory-to-memory Transfer



Notes: If an 8-bit data bus is selected,
1: D_7/A_{15}—D_0/A_8
2: A_{23}—A_{16}, A_7—A_0
The broken lines of the $\overline{MWR}$ signal are for extended write timing.

Figure 19. $\overline{END}/\overline{TC}$ Timing



Examples of System Configuration

Figures 20 through 22 show system configuration examples using the 8-bit CXQ70108 CPU and the 16-bit CXQ70116 CPU. The CXQ71082 externally latches addresses and data.

Figure 20. System Configuration with CXQ70108

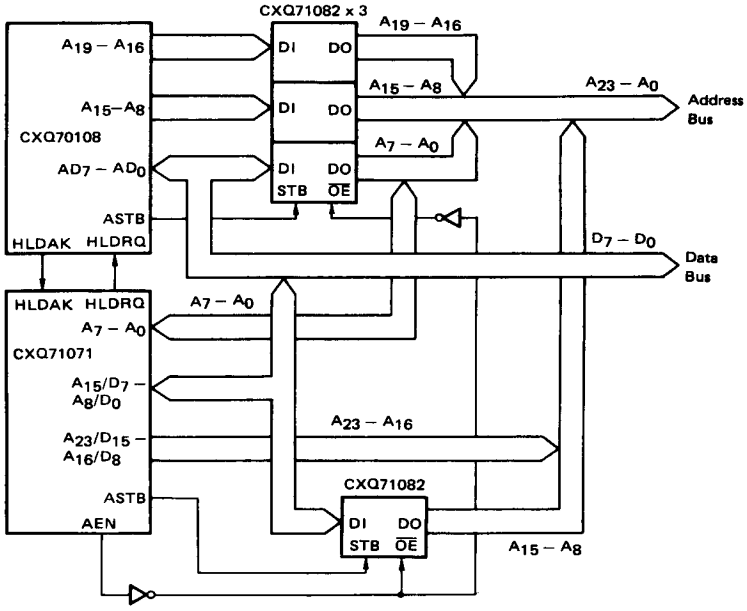


Figure 21. System Configuration with CXQ70116, Byte Transfer

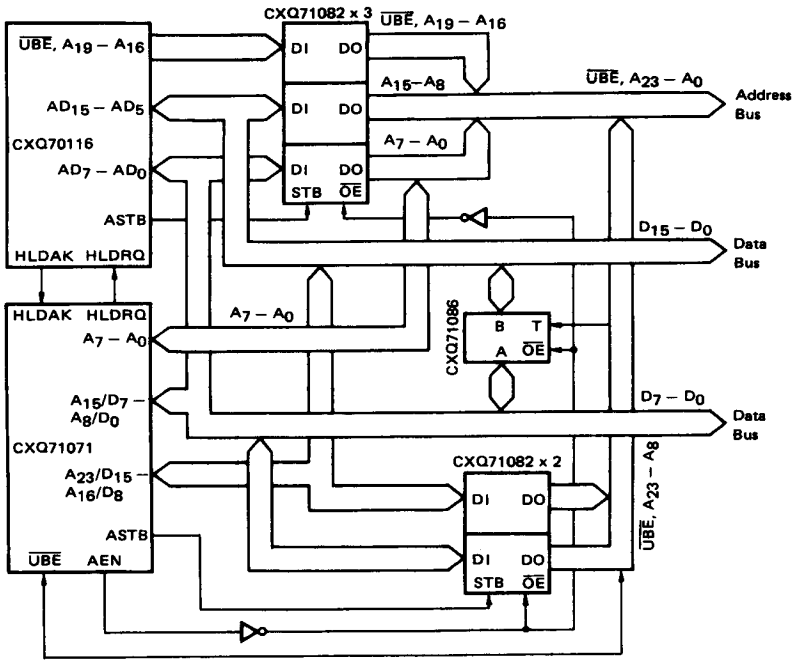
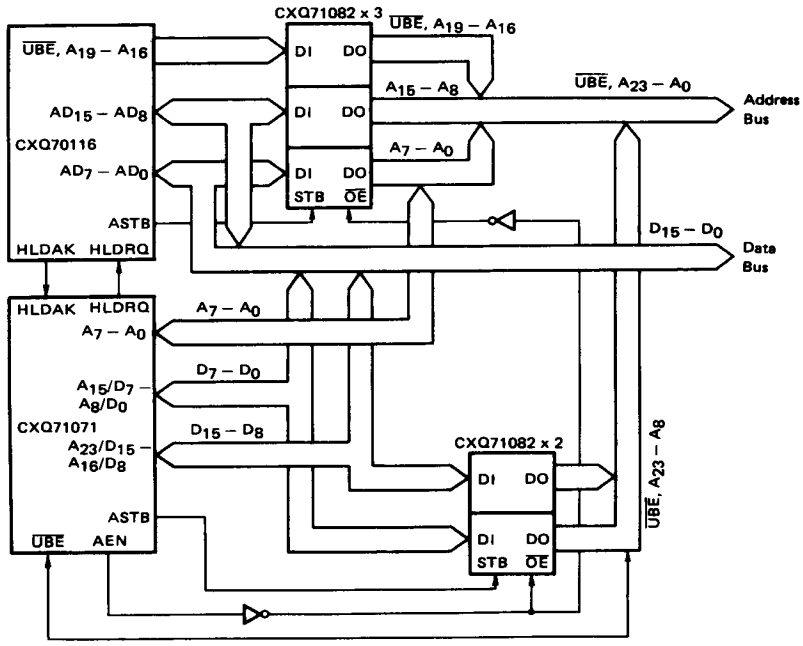


Figure 22. System Configuration with CXQ70116, Word Transfer



Package Outline

Unit: mm