SONY **CXQ71059**

Interrupt Control Unit

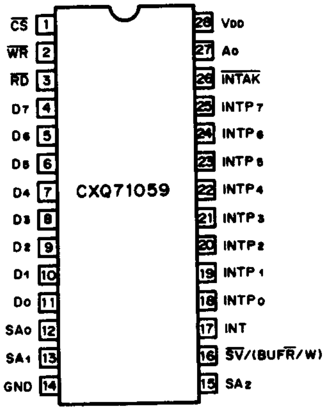
Description

The CXQ71059 is a low-power CMOS programmable interrupt control unit for microcomputer systems. It can process eight interrupt requests and can be expanded to 64 interrupt requests by adding other CXQ71059s. It transfers the interrupt request with the highest priority to the CPU, along with the interrupt address information.

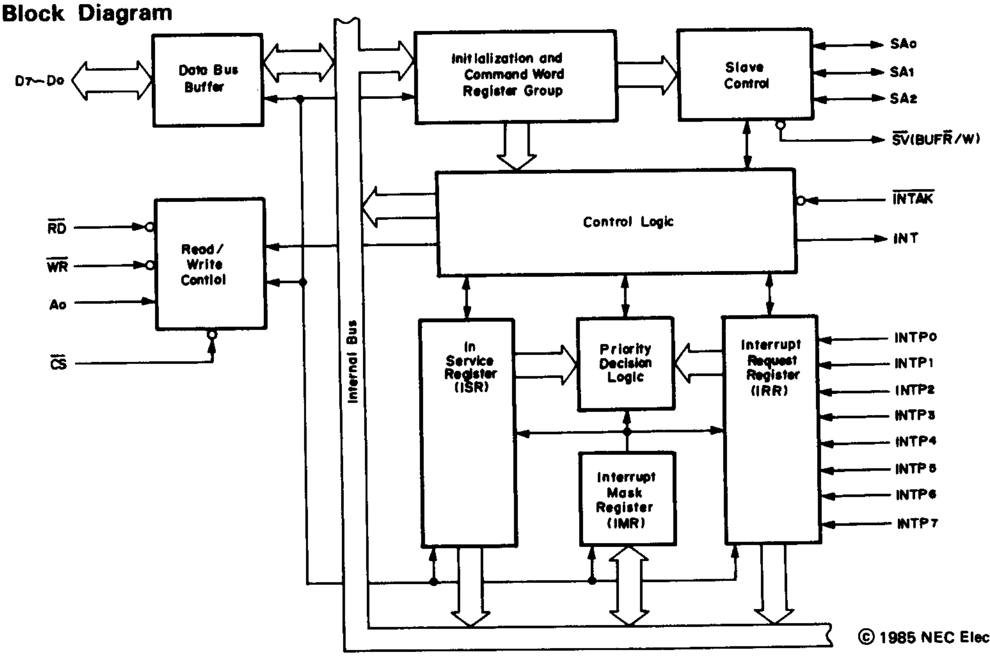
Features

- 8085A compatible (CALL mode)
- CXQ70108/70116 compatible (vector mode)
- Eight interrupt request inputs per chip
- Up to 64 interrupt requests inputs per system (extended mode)
- Edge- or level-triggered interrupt request inputs
- Each interrupt maskable
- Programmable priority level
- Polling operation
- CMOS technology
- +5V single power supply
- 28-pin plastic DIP (600 mil)
- NEC μPD71059 compatible

Pin Configuration (Top View)



Block Diagram



© 1985 NEC Electronics

Pin Identification

No.	Symbol	Direction	Function
1	$\overline{CS}$	In	Chip select input
2	$\overline{WR}$	In	Write strobe input
3	$\overline{RD}$	In	Read strobe input
4-11	D_7–D_0	In/Out	Data bus
12-13	SA_0–SA_1	In/Out	Slave address, bits 0, 1
14	GND		Ground potential
15	SA_2	In/Out	Slave address, bit 2
16	$\overline{SV}$/(BUF$\overline{R}$/W)	In/Out	Slave (Buffer read write)
17	INT	Out	Interrupt output
18-25	$INTP_0$–$INTP_7$	In	Interrupt inputs
26	$\overline{INTAK}$	In	Interrupt acknowledge input
27	A_0	In	Address input
28	V_{DD}		Power supply

Pin Functions

D_7–D_0 [Data Bus]

The 8-bit, three-state bidirectional bus is used to interface the 71059 to the system bus. The control words, status information, and interrupt-vector data are transferred.

$\overline{CS}$ [Chip Select]

The CPU sets $\overline{CS}$ low when selecting CXQ71059 to read from (IN instructions) or write to (OUT instructions). The $\overline{RD}$ and $\overline{WR}$ signals to the CXQ71059 are enabled when $\overline{CS}$ is low. $\overline{CS}$ is not used for the INTAK sequence.

$\overline{RD}$ [Read Strobe]

The CPU sets $\overline{RD}$ low when reading the internal registers IMR, IRR and ISR, and the polling data. during polling operations.

$\overline{WR}$ [Write Strobe]

The CPU sets $\overline{WR}$ low when writing initializing words IW1–IW4 and command words IMW, PFCW and MCW.

A_0 [Address]

A_0 is used with $\overline{CS}$, $\overline{RD}$, and $\overline{WR}$ to read or write to the CXQ71059. Normally, A_0 is connected to A_0 of the system address bus. Table 1 shows the relationship between read/write operations and the control signals ($\overline{CS}$, $\overline{WR}$, $\overline{RD}$ and A_0).

Table 1. Read/Write Operations

$\overline{\text{CS}}$	$\overline{\text{RD}}$	$\overline{\text{WR}}$	A_0	Other Conditions	CXQ71059 Operation	CPU Operation
0	0	1	0	IRR set by MCW	IRR to Data bus	IRR read
				ISR set by MCW	ISR to Data bus	ISR read
				Polling phase[1]	Polling data to Data Bus	Polling
0	0	1	1		IMR to Data bus	IMR read
0	1	0	0	$D_4=1$	Data bus to IW1 register	IW1 write
				D_4, $D_3=0$	Data bus to PFCW register	PFCW write
				$D_4=0$, $D_3=1$	Data bus to MCW register	MCW write
0	1	0	1	Note 2	Data bus to IW2 register	IW2 write
					Data bus to IW3 register	IW3 write
					Data bus to IW4 register	IW4 write
				After initializing	Data bus to IMR	IMW write
0	1	1	X		Data bus: high impedance	
1	X	X	X			
0	0	0	X		Illegal	

Notes: 1. In the polling phase, polling data is read instead of IRR and ISR.
2. Refer to Control Words section for IW2–IW4 writing procedure.

INTP_7–INTP_0 [Interrupt Request from Peripheral]

INTP_7–INTP_0 are eight asynchronous interrupt request inputs. They can be set to be either edge- or level-triggered. These pins are pulled up by an internal resistance. Their power consumption is lower at high-level input than at low-level input.

INT [Interrupt]

INT is the interrupt request output from a CXQ71059 to the CPU or master CXQ71059. When an interrupt from a peripheral is input to an INTP pin and acknowledged, the CXQ71059 asserts INT high to generate an interrupt request to the CPU or master CXQ71059.

$\overline{\text{INTAK}}$ [Interrupt Acknowledge]

$\overline{\text{INTAK}}$ informs the CXQ71059 that its interrupt request is being acknowledged by the CPU. During this acknowledgement, the CPU returns three low-level pulses (CALL mode) or two low-level pulses (vector mode).

$\overline{\text{SV}}$/($\text{BUF}\overline{\text{R}}$/W) [Slave, Buffer Read/Write]

This pin has two functions. When in non-buffer made, it is the $\overline{\text{SV}}$ input to designate a slave ($\overline{\text{SV}}$=0) or master ($\overline{\text{SV}}$=1). $\overline{\text{SV}}$ has no meaning when the CXQ71059 is set to signal mode.

This signal allows a bus transceiver to be controlled by the CXQ71059, if buffer mode is used. This pin becomes a $\text{BUF}\overline{\text{R}}$/W output. When the CXQ71059 changes its data bus to output, $\text{BUF}\overline{\text{R}}$/W goes low. $\text{BUF}\overline{\text{R}}$/W goes high when the data bus changes to input.

SA_2–SA_0 [Slave Address]

These pins are only used in systems with cascaded CXQ71059s. The master CXQ71059 uses these pins to address up to eight slave CXQ71059s. These pins are output pins for masters, and input pins for slaves.

Note: In the single mode, SA_2-SA_0 are output pins, but the output data has no meaning.

V_{DD} [Power Supply]

This is the positive power supply.

GND [Ground]

This is the ground potential.

Absolute Maximum Ratings

Ta=25°C

Parameter	Symbol	Rating Value	Unit
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
Power dissipation	P_{DMAX}	500	mW
Operating temperature	Topr	−40 to +85	°C
Storage temperature	Tstg	−65 to +150	°C

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

DC Characteristics

Ta=−40 to +85°C; V_{DD}=5V±10%

Parameter	Symbol	Min.	Max.	Unit	Test Conditions
Input voltage high	V_{IH}	2.2	V_{DD}+0.3	V	
Input voltage low	V_{IL}	−0.5	0.8	V	
Output voltage high	V_{OH}	0.7×V_{DD}		V	I_{OH}=−400 μA
Output voltage low	V_{OL}		0.4	V	I_{OL}=2.5 mA
Input leakage current high	I_{LIH}		10	μA	V_I=V_{DD}
Input leakage current low	I_{LIL}		−10	μA	V_I=0V
Output leakage current high	I_{LOH}		10	μA	V_O=V_{DD}
Output leakage current low	I_{LOL}		−10	μA	V_O=0V
INTP input leakage current high	I_{LIPH}		10	μA	
INTP input leakage current low	I_{LIPL}		−300	μA	
Supply current	I_{DD1}		9	mA	Operation
	I_{DD2}		50	μA	Stand-by Mode

Capacitance

$T_a=25°C$; $V_{DD}=0V$

Parameter	Symbol	Min.	Max.	Unit	Test Conditions
Input capacitance	C_I		10	pF	$f_C=1$ MHz Unmeasured pins returned to 0V
I/O capacitance	C_{IO}		20	pF	

AC Characteristics ($T_a=-40$ to $+85°C$; $V_{DD}=5V\pm10\%$)

Read Timing

Parameter	Symbol	Min.	Max.	Unit	Test Conditions
A_0, $\overline{CS}$ set-up to $\overline{RD}$ ↓	t_{SAR}	0		ns	
A_0, $\overline{CS}$ hold from $\overline{RD}$ ↑	t_{HRA}	0		ns	
$\overline{RD}$ pulse width low	t_{RRL}	160		ns	
$\overline{RD}$ pulse width high	t_{RRH}	120		ns	
Data delay from $\overline{RD}$ ↓	t_{DRD}		120	ns	$C_L=150$ pF
Data float from $\overline{RD}$ ↑	t_{FRD}	10	85	ns	$C_L=100$ pF
Data delay from A_0, $\overline{CS}$	t_{DAD}		200	ns	$C_L=150$ pF
BUF$\overline{R}$/W delay from $\overline{RD}$ ↓	t_{DRBL}		100	ns	
BUF$\overline{R}$/W delay from $\overline{RD}$ ↑	t_{DRBH}		150	ns	

Write Timing

Parameter	Symbol	Min.	Max.	Unit	Test Conditions
A_0, $\overline{CS}$ set-up to $\overline{WR}$ ↓	t_{SAW}	0		ns	
A_0, $\overline{CS}$ hold from $\overline{WR}$ ↑	t_{HWA}	0		ns	
$\overline{WR}$ pulse width low	t_{WWL}	120		ns	
$\overline{WR}$ pulse width high	t_{WWH}	120		ns	
Data set-up to $\overline{WR}$ ↑	t_{SDW}	120		ns	
Data hold from $\overline{WR}$ ↑	t_{HWD}	0		ns	

Interrupt Timing

Parameter	Symbol	Min.	Max.	Unit	Test Conditions
INTP pulse width	t_{IPIPL}	100		ns	See note
SA set-up to second, third $\overline{\text{INTAK}}$ ↓	t_{SSIA}	40		ns	Slave
$\overline{\text{INTAK}}$ pulse width low	t_{IAIAL}	160		ns	
$\overline{\text{INTAK}}$ pulse width high	t_{IAIAH}	120		ns	$\overline{\text{INTAK}}$ Sequence
INT delay from INTP ↑	t_{DIPI}		300	ns	C_L=150 pF
SA delay from first $\overline{\text{INTAK}}$ ↓	t_{DIAS}		360	ns	Master, C_L=150 pF
Data delay from $\overline{\text{INTAK}}$ ↓	t_{DIAD}		120	ns	C_L=150 pF
Data float from $\overline{\text{INTAK}}$ ↑	t_{FIAD}	10	85	ns	C_L=100 pF
Data delay from SA	t_{DSD}		200	ns	Slave, C_L=150 pF
BUF$\overline{\text{R}}$/W delay from $\overline{\text{INTAK}}$ ↓	t_{DIABL}		100	ns	C_L=150 pF
BUF$\overline{\text{R}}$/W delay from $\overline{\text{INTAK}}$ ↑	t_{DIABH}		150	ns	

Note: The time to clear the input latch in edge-trigger mode.

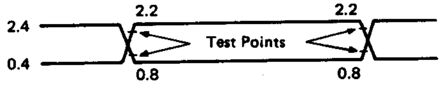
Other Timing

Parameter	Symbol	Min.	Max.	Unit	Test Conditions
Command recovery time	t_{RV1}	120		ns	Note 1
$\overline{\text{INTAK}}$ recovery time	t_{RV2}	250		ns	Note 2
$\overline{\text{INTAK}}$/command recovery time	t_{RV3}	250		ns	Note 3

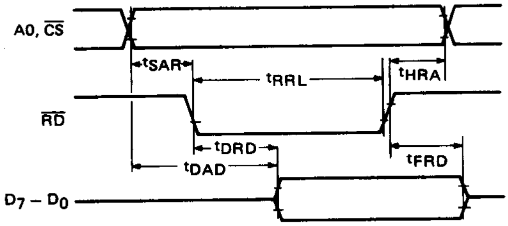
Notes: 1. The time to move from read to write operation.
2. The time to move $\overline{\text{INTAK}}$ to the next $\overline{\text{INTAK}}$ operation.
3. The time to move $\overline{\text{INTAK}}$ to/from command (read/write).

Timing Waveforms

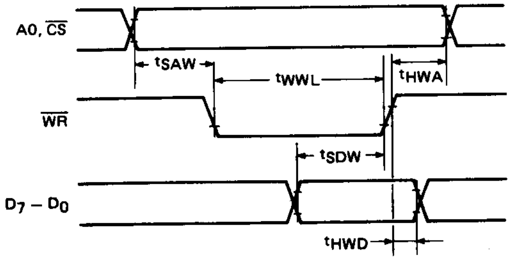
AC Test Input/Output Waveform



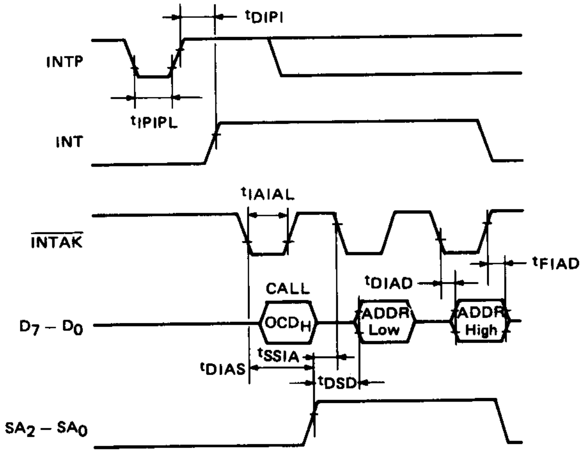
Read Cycle



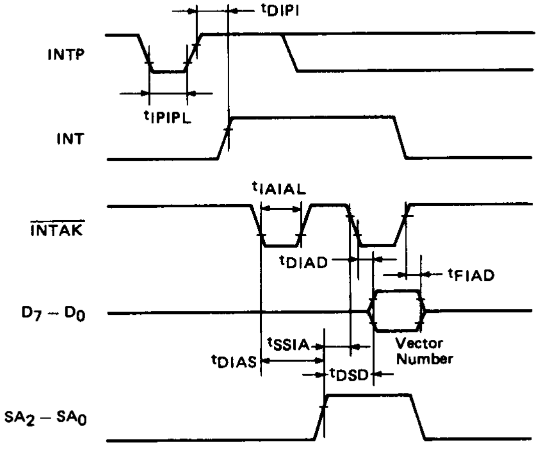
Write Cycle



$\overline{\text{INTAK}}$ Sequence (CALL Mode) 8085A

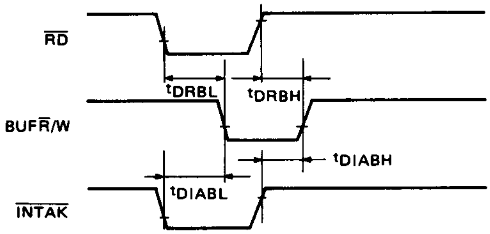


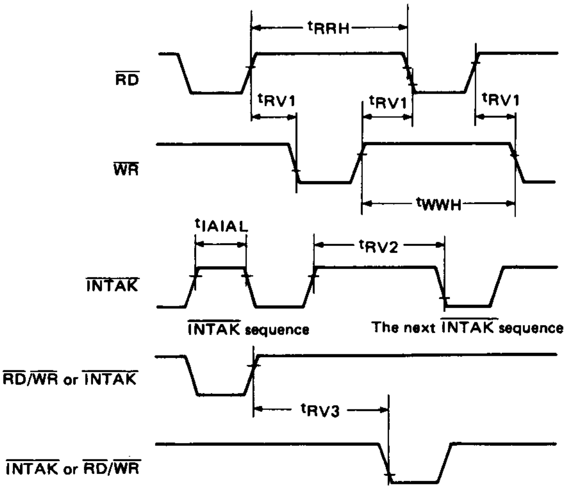
$\overline{\text{INTAK}}$ Sequence (Vector Mode) CXQ70108/70116



INTP Input should be maintained at high level until the leading edge of the 1st $\overline{\text{INTAK}}$ pulse.

Other Timing





Block Diagram Functions

Data Bus Buffer

The data bus buffer is a buffer between D_7–D_0 and the CXQ71059's internal bus.

Read/Write Control

The read/write control controls the reading from and writing to the CXQ71059 registers.

Initialization and Command Word Registers

There registers store initializing words IW1-IW4 and command words PFCW (priority and finish control word) and MCW (mode control word). The CPU cannot read these registers.

Interrupt Mask Register (IMR)

The interrupt mask register stores the interrupt mask word (IMW). Each bit masks an interrupt. If bit n of this register is 1, the interrupt request $INTP_n$ is masked and cannot be accepted by the CXQ71059. The CPU can read this register by performing an IN instruction with $A_0=1$.

Interrupt Request Register (IRR)

The interrupt request register shows which interrupt levels are currently being requested. If bit n of the IRR is 1, $INTP_n$ is requesting service .The CPU can read this register.

In-Service Register (ISR)

The in-service register shows all interrupt levels currently in service. If bit n of this register is 1, the interrupt routine corresponding to $INTP_n$ is currently being executed. The CPU can read this register.

Priority Decision Logic

The priority decision logic decides the highest priority interrupt request in IRR. The decision is based on the state of IMR, ISR, and the mode setting.

Control Logic

The control logic receives and generates the signals that control the sequence of events in an interrupt.

Slave Control

Slave control is used in systems with cascaded CXQ71059s. A master CXQ71059 uses it to control slave CXQ71059s, and a slave uses it to interface with the master CXQ71059.

Interrupt Operation

Almost all microcomputer systems use interrupts to reduce the overhead when controlling peripherals. However, the number of interrupt pins on a CPU is limited. When the number of interrupt lines increases beyond that limit, external circuits like the CXQ71059 become necessary.

The CXQ71059 can process eight interrupt request according to an allocated priority order and transmit the signal with the highest priority to the CPU. It also supplies the CPU with information to ascertain the interrupt routine start address. Cascading CXQ71059s by connecting up to eight "slave" CXQ71059s to a single "master" CXQ71059 permits expansion up to 64 interrupt request signals.

Interrupt system scale, interrupt routine addresses, interrupt request priority, and interrupt request masking are all programmable, and can be defined by the CPU.

Normal interrupt operation for a single CXQ71059 is as follows. First, the initialization registers are set-up by a sequence of initialization words. When the CXQ71059 detects an interrupt request from a peripheral to an INTP pin, it sets the corresponding bit of the interrupt request register (IRR). The interrupt is checked with the interrupt mask register (IMR) and the interrupt service register (ISR). If the interrupt is not masked and there is no other interrupt with a higher priority in service or requesting service, it generates an INT signal to the CPU.

The CPU acknowledges the interrupt by bringing the $\overline{\text{INTAK}}$ line low. The CXQ71059 sets the corresponding bit in its ISR to indicate that this interrupt is in service and to disable interrupts with lower priority. It resets the bit in the IRR at this point. The CXQ71059 then peaces interrupt CALL or vector data onto the data bus in response to $\overline{\text{INTAK}}$ pulses. When the CPU has finished processing the interrupt, it will inform the CXQ71059 by sending a finish interrupt (FI) command. This resets the bit in the ISR and allows the CXQ71059 to accept interrupts with lower priorities. If the CQ71059 is programmed in the self-FI mode, the ISR bit is reset automatically and this step is not necessary.

Software Features

The CXQ71059 has the following software features:

- Interrupt types: CALL/vector
- Interrupt masking: Normal/extended nesting
- End of interrupt: Self-FI/normal FI/specific FI
- Priority rotation: Normal nested/extended nested/exceptional nested
  Automatic priority rotation
  Rotate to specific priority
- Polled mode
- CPU-readable registers

Hardware Configurations

The CXQ71059 has the following hardware configurations:

- Interrupt input: Edge/level sensitive
- Cascading CXQ71059s: Single/extended (master/slave)
- Output driver control: Buffered/non-buffered

Mode Control

These features and configurations are selected and controlled by the four initialization words (IW1—IW4) and the three command words (IMW, PFCW, and MCW). The format of these words are shown in figures 2 and 3, respectively.

Control Words

These are two types of CXQ71059 control words: initialization words and command words.

There are four intialization words: IW1–IW4. These words must be written to the CXQ71059 to initialize it prior to normal operation. They must be written in a sequence of two to four bytes.

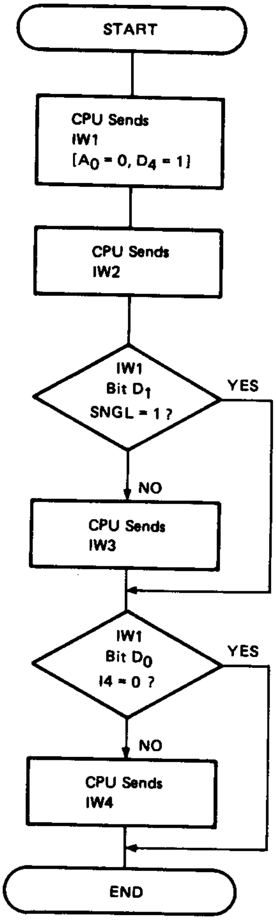
There are three types of command words: interrupt mask word (IMW), priority and finish control word (PFCW), and the mode control word (MCW). These words can be written freely after initialization.

Initialization Words

Initialization sequence. When a data is written to a CXQ71059 with $A_0=0$ and $D_4=1$, the data is always accepted as IW1. This results in a default initialization as shown below. See figure 1.

(1) The edge-trigger circuit of the INTP input is reset. IRR is cleared in the edge-trigger mode.
(2) ISR and IMR are cleared.
(3) INTP$_7$ receives the lowest priority; INTP$_0$ receives the highest.
(4) The exceptional nesting mode is cleared. IRR is set as the register to be read.
(5) Register IW4 is cleared. The normal nesting mode, non-buffer mode, FI command mode, and CALL mode are entered.

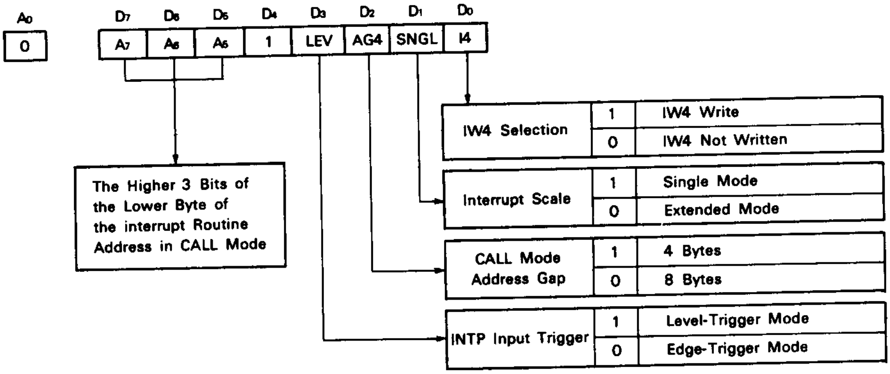
Figure 1. Initialization Sequence



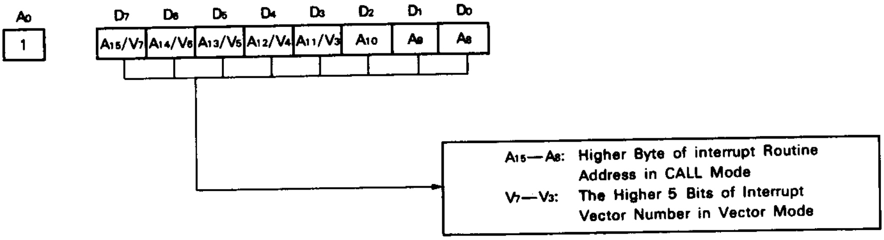
The initialization words are written in a consecutive format. The first two, IW1 and IW2 must be written for any mode of the operation, and designate the interrupt address or vector. IW3 specifies which interrupts have slaves if the CXQ71059 is used for a master, and defines the slave number for a slave. Therefore, IW3 is only required in extended systems. The CXQ71059 will only accept it if bit D_1 of IW1, SNGL=0. IW4 is only accepted if bit D_0 IW1, I4=1. See figure 2 for the format of the initialization words.

Figure 2. Initialization Word Format

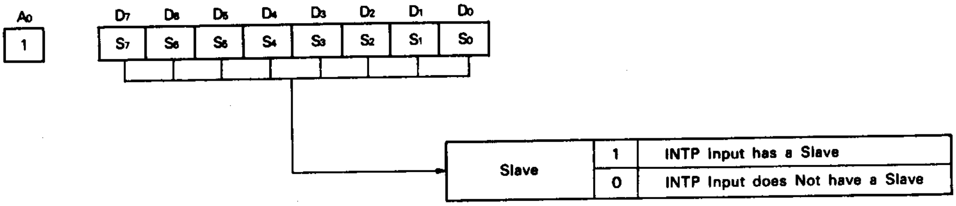
IW1 [Initialization Word 1]



IW2 [Initialization Word 2]



IW3 [Initialization Word 3] Master Mode



IW3 [Initialization Word 3] Slave Mode

A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	SN_2	SN_1	SN_0

Slave Number	0	0	0	0
	1	0	0	1
	2	0	1	0
	3	0	1	1
	4	1	0	0
	5	1	0	1
	6	1	1	0
	7	1	1	1

IW4 [Initialization Word 4]

A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	EXTN	BUF	$\overline{\text{BSV}}$	SFI	V/$\overline{\text{C}}$

Interrupt Mode Select	1	Vector Mode
	0	CALL Mode

FI Mode	1	Self-FI Mode
	0	FI Command Mode

	BUF	$\overline{\text{BSV}}$	
Buffer Mode Select	0	X	Non-Buffer Mode
	1	1	Buffer Mode Master
	1	0	Buffer Mode Slave

Extended Nest	1	Extended Nest Mode
	0	Normal Nest Mode

IW1:

A_7–A_5 are the higher three bits of the lower byte of an interrupt routine address given to the CPU in CALL mode.

LEV is used to select the trigger mode of the INTP inputs. If LEV=0, the edge-trigger mode is selected. If LEV=1, the level-trigger mode is selected.

AG4 (address gap 4 bytes) sets the spacing of interrupt routine addresses in the CALL mode to either 4 or 8. For example, when the address of INTP$_0$'s interrupt routine is 1000H, INTP$_1$'s address is 1004H when AG4=1 and 1008H when AG4=0.

SNGL is used to designate the scale of the interrupt system. If SNGL=1, only one CXQ71059 is used in a system, and IW3 will not be accepted.

IW4 will be written when I4=1. It will not be written when I4=0.

IW2:

A_{15}—A_8 is a higher byte of the interrupt routine address given to the CPU in the CALL mode.

V_7—V_3 are the five higher bits of the interrupt vector number given to the CPU in the vector mode.

IW3:

IW3 has meaning only in the extended mode.

When the CXQ71059 is the master in an extended mode system, S_7–S_0 define whether INTP$_7$–INTP$_0$ have slaves or peripherals. For example, if S_2=1, that indicates that interrupts to pin INTP$_2$ are from slave CXQ71059. The master CXQ71059 will output the slave number 2 on SA$_2$–SA$_0$ during the $\overline{\text{INTAK}}$ cycle and will not place the interrupt address or vector number on the data bus. The slave then outputs the interrupt address or vector number. When S_2=0, the master will place the interrupt address or vector numbers on the data bus for INTP$_2$ interrupts.

In the slave mode, the lower three bits of IW3 set the slave number of the CXQ71059. This number will be compared with the number put on pins SA$_2$-SA$_0$ output by the master CXQ71059 during as $\overline{\text{INTAK}}$ cycle. If there is a match, the slave knows that its interrupt is being honored and issues the interrupt address or vector number onto the data bus.

IW4:

EXTN (extended mode) sets the nesting mode. When EXTN=0, the normal nesting mode is set. When EXTN=1, the extended nesting mode is set.

SFI (self-finish interrupt mode) set the FI mode. When SFI=0, the FI (finish interrupt) command mode is set. In FI command mode, an FI command must be sent to the CXQ71059 to terminate the interrupt. When SFI=1, the self-FI mode is set and the CXQ71059 will automatically perform an FI command at the end of every $\overline{\text{INTAK}}$ cycle.

BUF (buffer) is used to designate the buffer mode. If BUF=1, the buffer mode is set.

$\overline{\text{BSV}}$ (buffered slave) is used with BUF. If BUF=1, $\overline{\text{BSV}}$ defines whether the CXQ71059 is a master or a slave. When $\overline{\text{BSV}}$=0, the CXQ71059 is master; when $\overline{\text{BSV}}$=1, a slave.

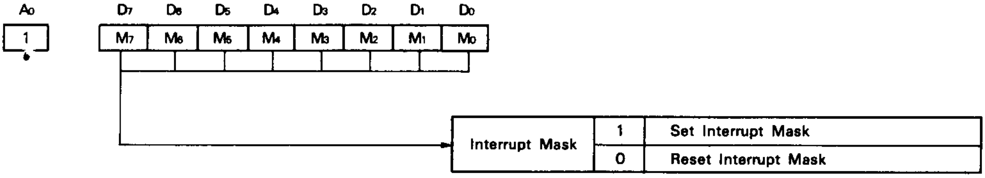
V/$\overline{\text{C}}$ sets the vector or CALL mode. The vector mode is set when V/$\overline{\text{C}}$=1 and the CALL mode is set when V/$\overline{\text{C}}$=0. These modes should be programmed to match the system's CPU.

Command Words

The command words give various commands to a CXQ71059 during its operation to change interrupt masks and priorities, to end interrupt processing, etc. See figure 3.

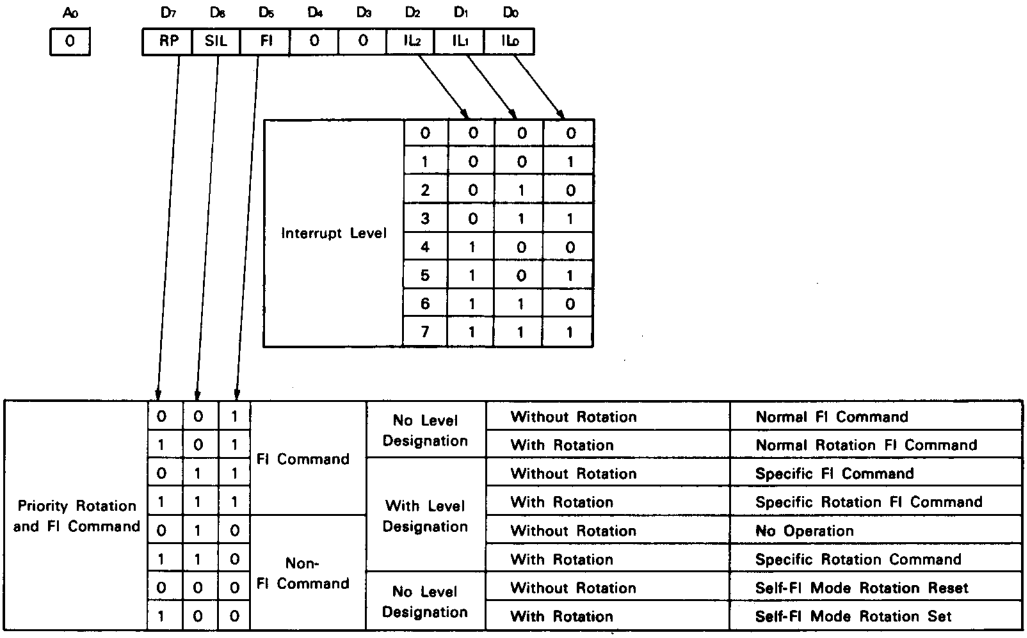
Figure 3. Command Word Format

IMW [Interrupt Mask Word]



Interrupt Mask	1	Set Interrupt Mask
	0	Reset Interrupt Mask

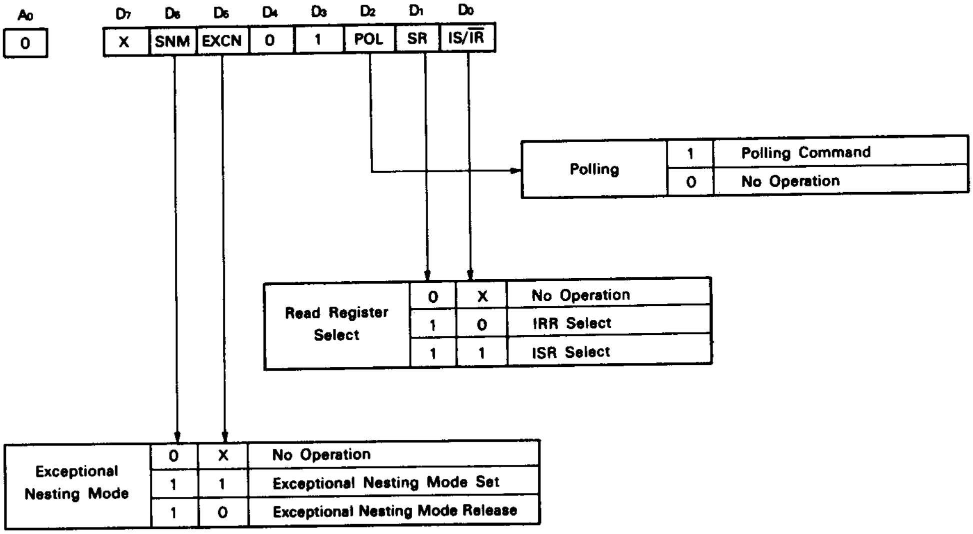
PFCW [Priority Finish and Control Word]



Interrupt Level	0	0	0	0
	1	0	0	1
	2	0	1	0
	3	0	1	1
	4	1	0	0
	5	1	0	1
	6	1	1	0
	7	1	1	1

Priority Rotation and FI Command	0	0	1	FI Command	No Level Designation	Without Rotation	Normal FI Command
	1	0	1			With Rotation	Normal Rotation FI Command
	0	1	1		With Level Designation	Without Rotation	Specific FI Command
	1	1	1			With Rotation	Specific Rotation FI Command
	0	1	0	Non-FI Command		Without Rotation	No Operation
	1	1	0			With Rotation	Specific Rotation Command
	0	0	0		No Level Designation	Without Rotation	Self-FI Mode Rotation Reset
	1	0	0			With Rotation	Self-FI Mode Rotation Set

MCW [Mode Control Word]



Polling	1	Polling Command
	0	No Operation

Read Register Select	0	X	No Operation
	1	0	IRR Select
	1	1	ISR Select

Exceptional Nesting Mode	0	X	No Operation
	1	1	Exceptional Nesting Mode Set
	1	0	Exceptional Nesting Mode Release

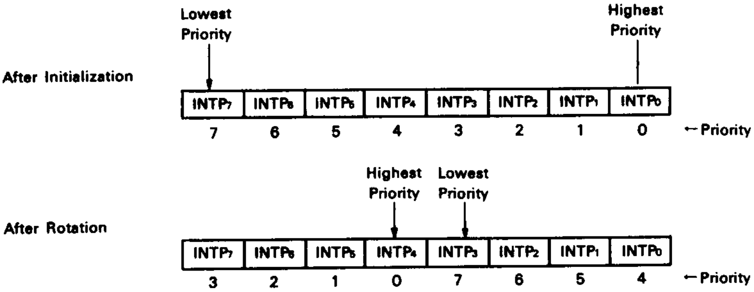
IMW (Interrupt Mask Word). This word masks the IRR and disables the corresponding INTP interrupt requests. It also maskes the ISR in the exceptional nesting mode. Bits M_7–M_0 correspond to the interrupt levels of INTP$_7$–INTP$_0$, respectively. IMW is stored in the IMR.

Interrupts corresponding to the bits of IRR and ISR in the exceptional nesting mode are masked if the M_n bit is set to 1.

PFCW (Priority and Finish Control Word). This word designates the FI command that defines the way how interrupts are ended, and the commands that change interrupt request priorities.

When RP (rotate priority) is set to 1, the priorities of the interrupt request change (rotate). The priority order of the 8 INTP pins is circular, as shown in Figure 4. If a level is assigned to the lowest priority, the priorities for all the other levels will be defined correspondingly. For example, if INTP$_3$ becomes the lowest priority, INTP$_4$ will be the highest. (INTP$_7$ has the lowest priority after initialization).

Figure 4. INTP Priority Order



SIL (specify interrupt level) is set to 1 to change the priority order or designate an interrupt level. It is used with the RP and FI bits (bits D_7 and D_5). When SIL =1 and RP=1, the level identified by IL_2—IL_0 is designated as the lowest priority level. The other priorities will be assigned correspondingly. When SIL=1 used with FI=1, it resets the ISR bit corresponding to the interrupt level IL_2—IL_0. If SIL=0, the IL_2—IL_0 bits have no meaning.

MCW (Mode Control Word). This word is used to enter the exceptional nesting mode, to poll the CXQ71059, and to read the ISR and IRR registers.

Bits SR (set register) and $IS/\overline{IR}$ (ISR/IRR) are used to read the contents of the IRR and ISR registers. When SR=0, no operation is performed. To read IRR or ISR, set A_0=0 and select the IRR or ISR register by writing to MCW. To select the IRR register, write MCW with SR=1 and $IS/\overline{IR}$=0. To select the ISR, write MCW with SR=1 and $IS/\overline{IR}$=1. The selection is retained, and MCW does not have to be rewritten to read the same register again. IRR and ISR are not masked by the IMR.

Bits SNW (set nesting mode) and EXCN (exceptional nesting mode) are used to set or clear the exceptional nesting mode. If SNW=0, EXCN executes no operation. If SNW=1 with EXCN=1, the exceptional nesting mode is set. If with EXCN=0, it is cleared.

POL (polling) is used to enable the polling operation. If POL=1, the polling command is issued. If POL=0, it is not issued.

CALL OR VECTOR MODES

The CXQ71059 passes interrupt routine address data to the CPU in two modes, depending on the CPU type. This mode is selected by bit $V/\overline{C}$ in initialization word IW4. $V/\overline{C}$ is set to one to to select the vector mode for CXQ70108/70116 CPUs, and reset to zero to select the CALL mode for 8085A CPUs.

CALL MODE (8085A CPUs)

In this mode, when an interrupt is acknowledged by the CPU, the CXQ71059 outputs three bytes of interrupt data to the data bus in its $\overline{INTAK}$ sequence. During the first $\overline{INTAK}$ pulse from the CPU, the CXQ71059 outputs the CALL opcode 0CDH. During the next $\overline{INTAK}$ pulse, it outputs the lower byte of a two-byte interrupt routine address. During the third INTAK pulse, it outputs the upper byte of the address. The CPU interprets these three bytes as a CALL instruction and executes the CALL interrupt routine. See figure 5.

Interrupt routine addresses are defined by using words IW1 and IW2 through the initialization. However, only the higher 10 or 11 bits of the interrupt addresses are designated: A_{15}—A_6 or A_{15}—A_5. The remaining lower bits (D_5—D_0 or D_4—D_0) are set to zero to get the address of $INTP_0$'s interrupt routine. The addresses for $INTP_1$–$INTP_7$ are set at equally speced gaps besed on it. The gap between interrupt addresses is determined by the AG4 bit (address gap 4 bytes) of IW1. When AG4=1, the interrupt routine starting addresses are 4 bytes apart. Therefore, the starting address for $INTP_n$ is the starting address for $INTP_0$ plus four times n. When AG4=0, starting addresses are eight bytes apart, so the starting address for $INTP_n$ is the starting address for $INTP_0$ plus eight times n. See figure 6.

Figure 5. CALL Mode Interrupt Sequence

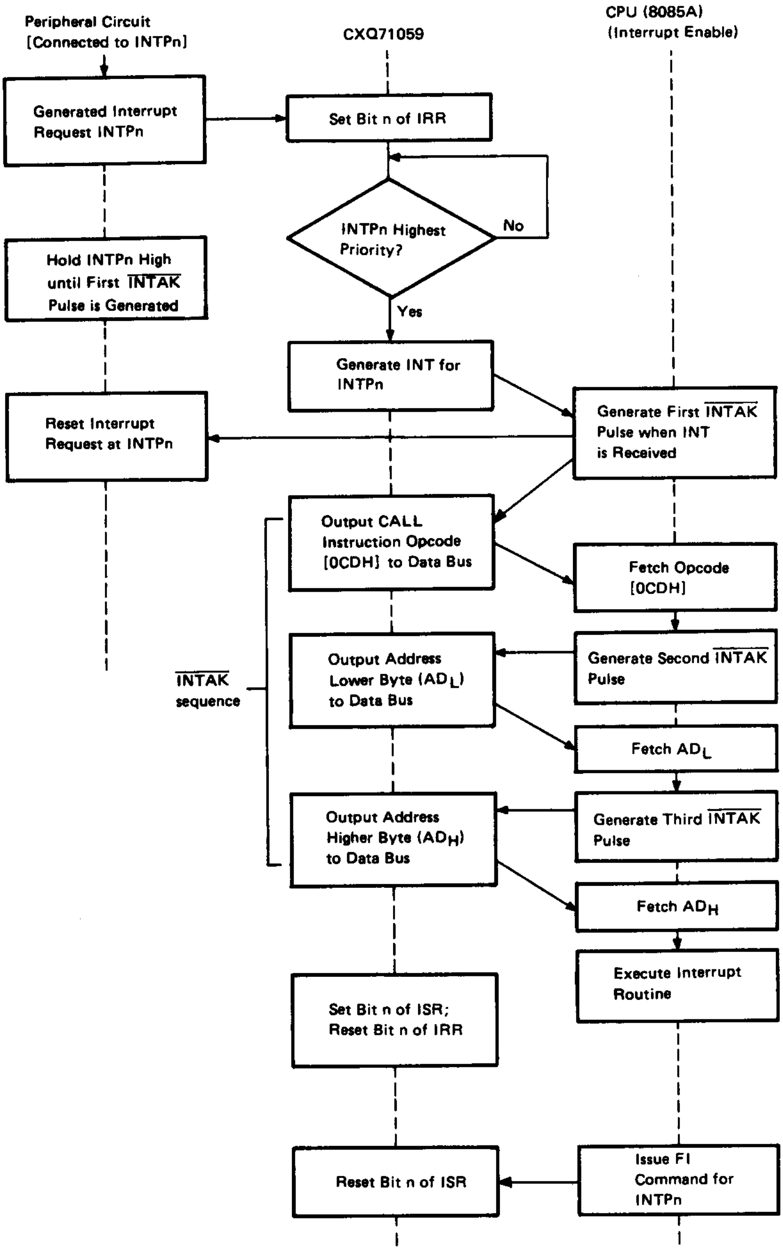


Figure 6. CALL Mode Interrupt Address Sequence

Address Lower Byte [AD$_L$] During Second $\overline{\text{INTAK}}$

AG4=1 (4-Byte Spacing Address)

Interrupt Level	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
INTP$_0$	A_7	A_6	A_5	0	0	0	0	0
INTP$_1$	A_7	A_6	A_5	0	0	1	0	0
INTP$_2$	A_7	A_6	A_5	0	1	0	0	0
INTP$_3$	A_7	A_6	A_5	0	1	1	0	0
INTP$_4$	A_7	A_6	A_5	1	0	0	0	0
INTP$_5$	A_7	A_6	A_5	1	0	1	0	0
INTP$_6$	A_7	A_6	A_5	1	1	0	0	0
INTP$_7$	A_7	A_6	A_5	1	1	1	0	0

AG4=0 (8-Byte Spacing Address)

Interrupt Level	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
INTP$_0$	A_7	A_6	0	0	0	0	0	0
INTP$_1$	A_7	A_6	0	0	1	0	0	0
INTP$_2$	A_7	A_6	0	1	0	0	0	0
INTP$_3$	A_7	A_6	0	1	1	0	0	0
INTP$_4$	A_7	A_6	1	0	0	0	0	0
INTP$_5$	A_7	A_6	1	0	1	0	0	0
INTP$_6$	A_7	A_6	1	1	0	0	0	0
INTP$_7$	A_7	A_6	1	1	1	0	0	0

Note: When AG4=0, bit A_5 is ignored.

Address Higher Byte [AD$_H$] During Third $\overline{\text{INTAK}}$

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8

Vector Mode (CXQ70108/70116 CPUs)

In the vector mode, the CXQ71059 outputs a one-byte interrupt vector number to the data bus in the $\overline{\text{INTAK}}$ sequence. The CPU uses that vector number to generate an interrupt routine address. See figure 7.

The higher five bits of the vector number, V_7–V_3, are defined by IW2 during initialization. The CXQ71059 assigns the remaining three bits to successive interrupt vector numbers for the eight interrupts. See figure 8.

Figure 7. Vector Mode Interrupt Sequence

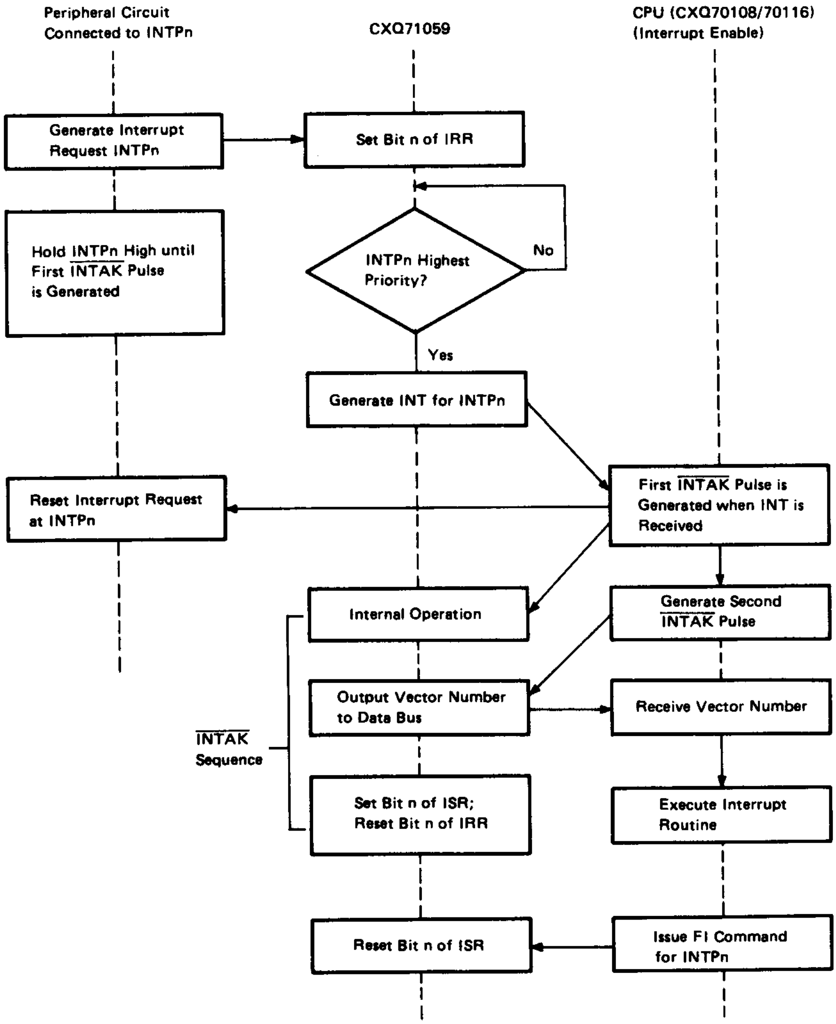


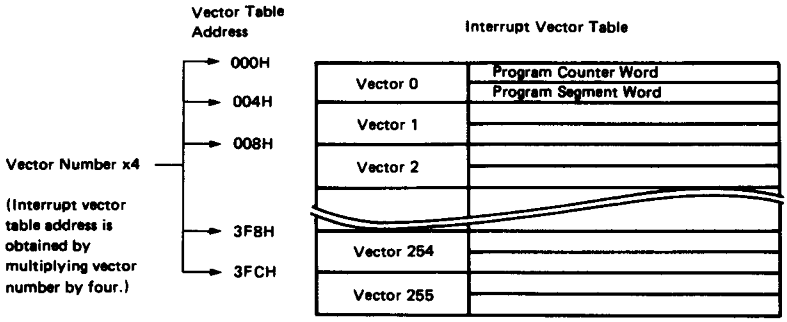
Figure 8. Vector Numbers Output in Vector Mode

Output During the Second $\overline{\text{INTAK}}$

Interrupt Level	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
$INTP_0$	V_7	V_6	V_5	V_4	V_3	0	0	0
$INTP_1$	V_7	V_6	V_5	V_4	V_3	0	0	1
$INTP_2$	V_7	V_6	V_5	V_4	V_3	0	1	0
$INTP_3$	V_7	V_6	V_5	V_4	V_3	0	1	1
$INTP_4$	V_7	V_6	V_5	V_4	V_3	1	0	0
$INTP_5$	V_7	V_6	V_5	V_4	V_3	1	0	1
$INTP_6$	V_7	V_6	V_5	V_4	V_3	1	1	0
$INTP_7$	V_7	V_6	V_5	V_4	V_3	1	1	1

The CPU generates an interrupt vector by multiplying the vector number by four, and using the result as the address of a location in an interrupt vector table located at addresses 000H–3FFH. See figure 9.

Figure 9. Interrupt Vectors for the CXQ70108/70116



System Scale Modes

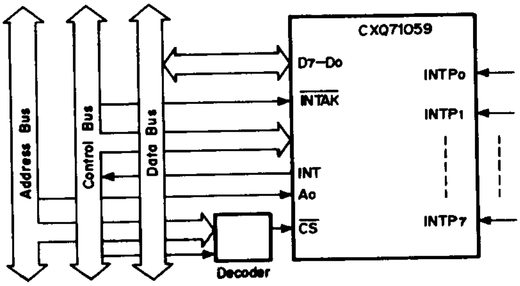
The CXQ71059 can operate in either single mode, with up to eight interrupt lines or extended mode, with more than one CXQ71059 and more than eight interrupt lines. In extended mode a CXQ71059 is in either master or slave mode.

Bit D_1 of the first initialization word, IW1, SNGL (single mode) designates the scale of the interrupt system. SNGL=1 designates that only one CXQ71059 is being used (single mode system). SNGL=0 designates an extended mode system with a master and slave CXQ71059s. In the single mode (SNGL=1), the SV input and IW4 buffer mode bits D_3 and D_2 are ignored by the CXQ71059.

Single Mode

This mode is the normal mode of CXQ71059 operation. It has been described in the Interrupt Operation section. See figure 10 for a system exmple.

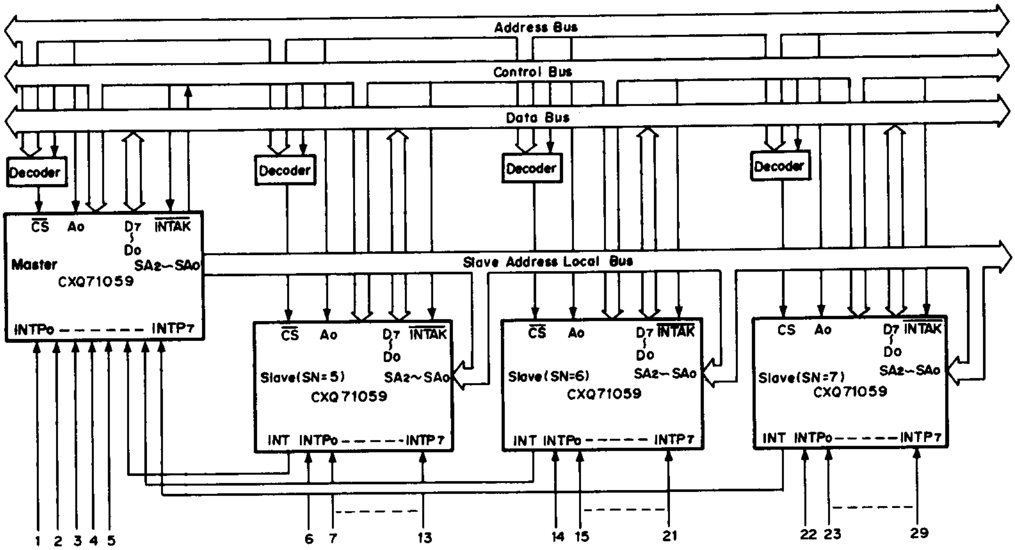
Figure 10. Single Mode System



Extended Mode

In this mode, up to 64 interrupt requests can be processed using a master connected to a maximum of eight slaves. See figure 11 for a system example.

Figure 11. Extended System Example with Three Slaves



Master Mode

In the master mode, S_7–S_0 are defined by IW3, identifying which INTP inputs have slaves and which do not. Let us consider an interrupt request from $INTP_n$.

If $S_n=0$, the interrupt is not from a slave (for example, $INTP_0$ of the master in Figure 11), and the CXQ71059 treats it the same way it would if it were in the single mode. SA_2–SA_0 outputs are low.

If $S_n=1$, the interrupt is from a slave (for example, $INTP_7$ of the master). The master sends an interrupt to the CPU if the slave requesting the interrupt has priority. The master then outputs slave address n to pins SA_2–SA_0 on the first $\overline{INTAK}$ pulse by the CPU. It lets the slave n perform the rest of the $\overline{INTAK}$ sequence.

Slave Mode

When a slave receives an interrupt request from a peripheral (assuming this request is higher than other requests and in-service levels on the slave,), it sends an interrupt request to the master through its INT output. When the interrupt is accepted by the CPU through the master, the master outputs the slave's address on pins SA_2–SA_0. Each slave compares the address on SA_2–SA_0 to its own address. The slave that has sent interrupt will find a match. It completes the $\overline{INTAK}$ sequence the same way as a single CXQ71059 would.

The master outputs slave address 0 when it is processing a non-slave interrupt. Therefore, do not use 0 as a slave address if there are less than eight slaves connected to the master.

Figures 12 and 13 show the interrupt operating sequences for slaves in the extended mode.

Figure 12. Interrupt from Slave (CALL Mode)

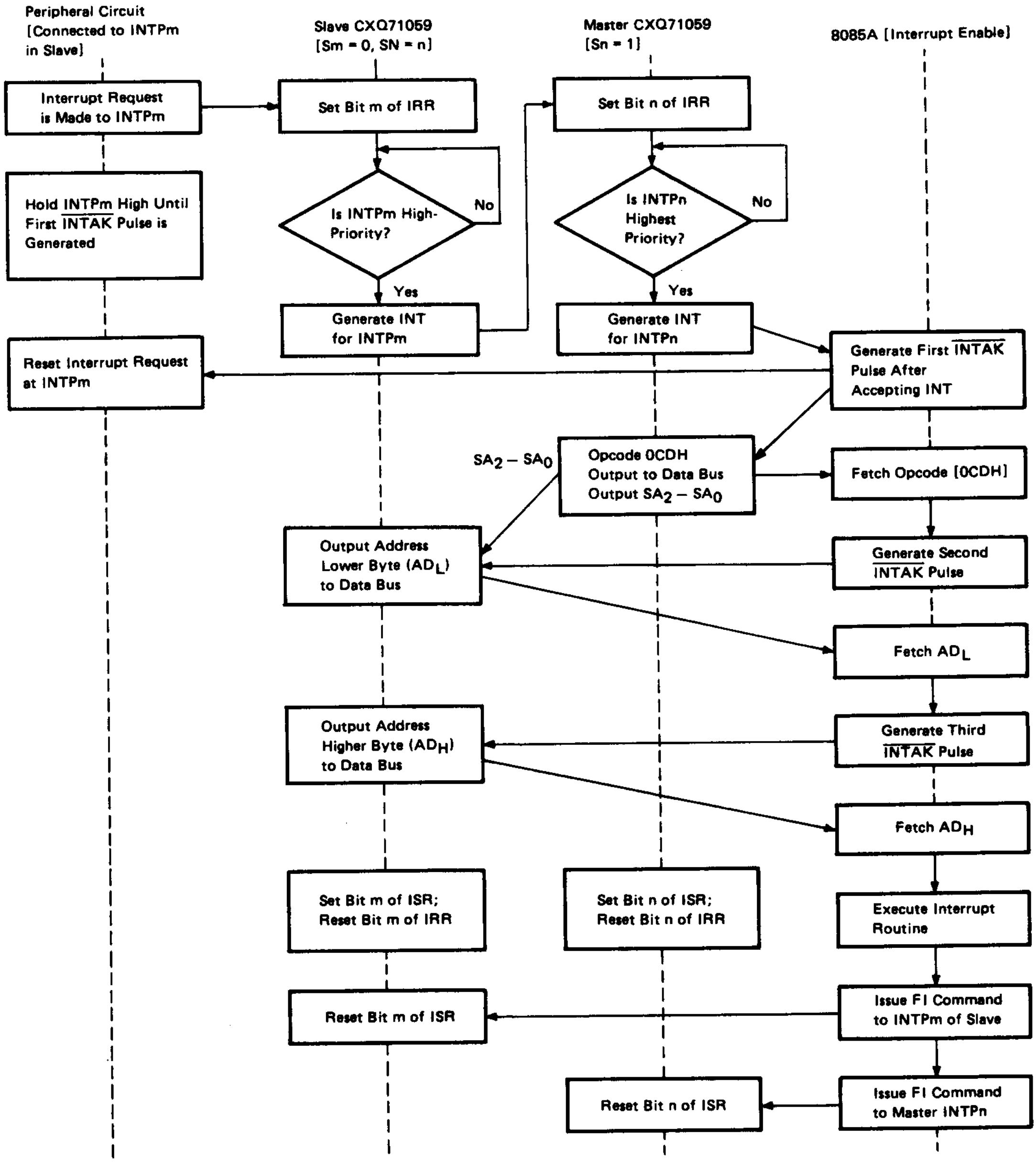
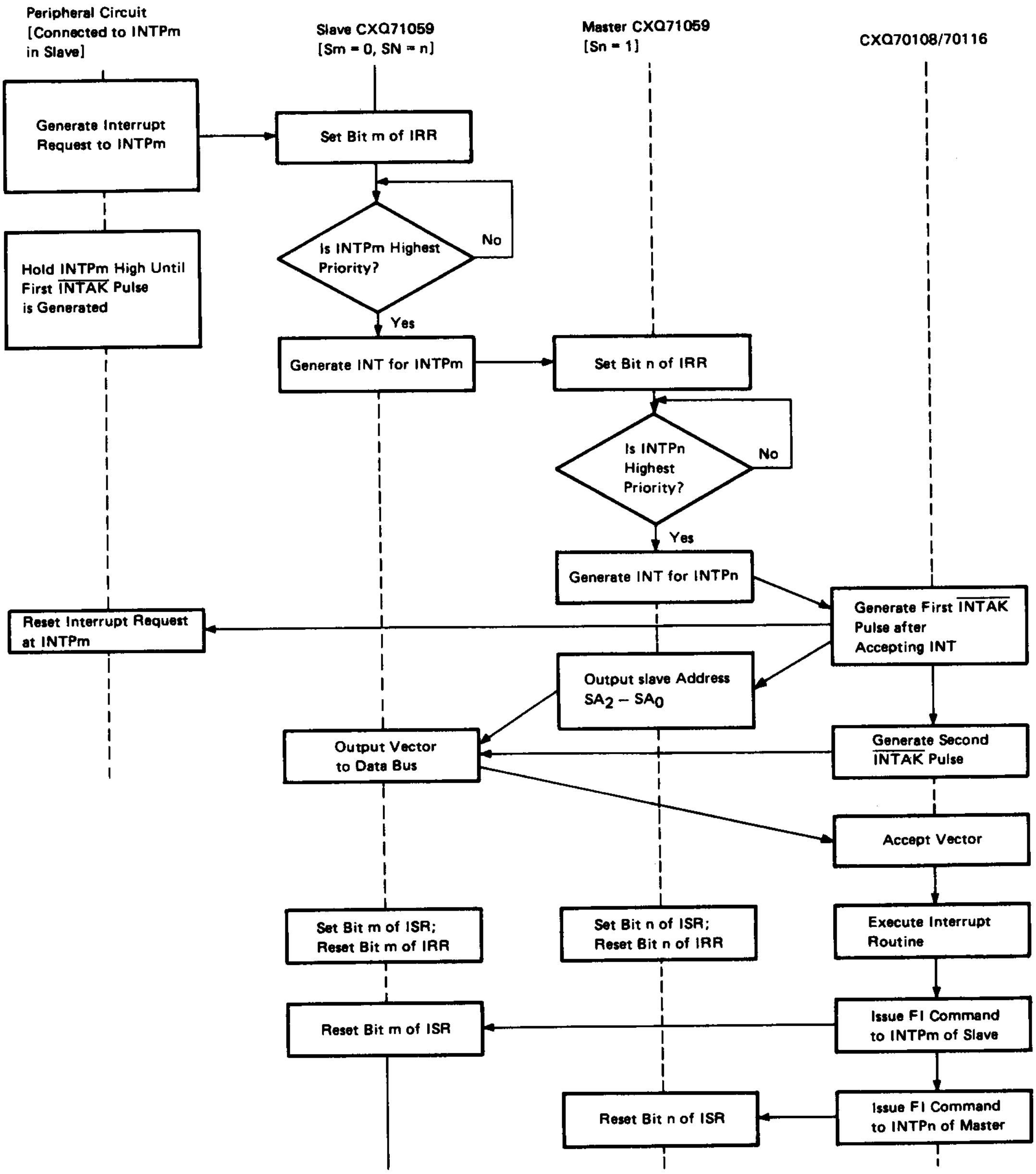


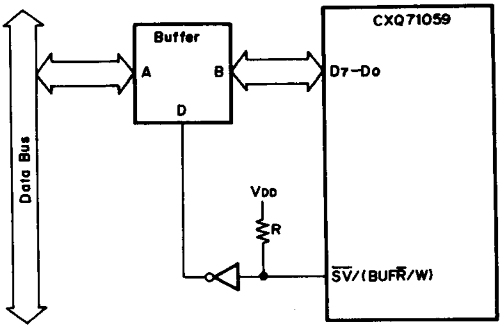
Figure 13. Interrupt from Slave (Vector Mode)



BUFFER AND NON-BUFFER MODES

When the system scale is large, a buffer may be needed for the CXQ71059 to drive the data bus. A buffer mode is supplied, with a signal to specify the buffer direction. In the buffer mode, $\overline{SV}$/(BUF$\overline{R}$/W) is used to select the buffer direction and $\overline{SV}$ cannot be used to specify the master/slave mode. The master/slave selection must be done by IW4. IW4 bit D_3, BUF (buffer) and D_2, $\overline{BSV}$ (buffered slave) are used together to define the buffer mode and master/slave relation. When BUF=0, the non-buffer mode is designated and $\overline{BSV}$ has no meaning. When BUF=1, the buffer mode is designated. In buffer mode, the CXQ71059 is a master when $\overline{BSV}$=1, a slave when $\overline{BSV}$=0. See figure 14.

Figure 14. Buffer Mode



Note 1: D determines data direction
Low Level: A → B
High Level: A ← B

2: The CXQ71059 is set to input $\overline{SV}$ in its initial state and is pulled up by R to set D to the low level during initialization.

NESTING MODES

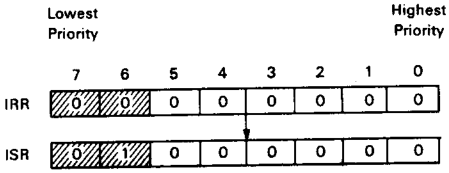
A variety of nesting modes are available for supporting a multilevel-interrupt structure.

Normal Nesting Mode

This mode is entered when IW4 is not written or when IW4 has EXTN=0. It is the most common nesting mode. See figure 15.

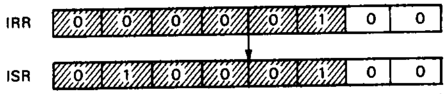
While an interrupt is being executed in this mode (corresponding bit of ISR=1), only interrupt requests of higher priority can be accepted and the requests of the same or lower priority cannot.

Figure 15. Normal Nesting Mode



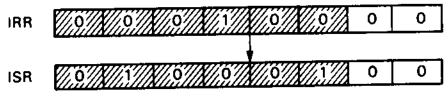
Interrupts that can be accepted are $INTP_5$ through $INTP_0$ during execution of interrupt Level 6.

Request Occurred in Level 2



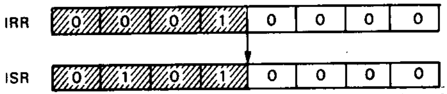
Interrupt Level 2 has been accepted and is being executed.

Request Occurred in Level 4



Level 4 requests cannot be accepted.

Level 2 FI Command Issued



Level 4 request can be accepted after processing of Level 2 has been ended, when high level is maintained at $INTP_4$ until $INTP_4$ is accepted.

Extended Nesting Mode

This mode is only applicable to a master in the extended mode. Eight interrupt priority levels of a slave become only one priority level when viewed by the master.

Therefore, a request made by a slave with a higher priority than one in service from the same slave will not be accepted. This is because the master's ISR bit is set, ignoring all requests of equal or lower priority in normal nesting mode. This cannot be called complete nesting since priority ranking within slave loses its significance.

When the extended nesting mode is entered (EXTN=1 in IW4 of the master), interrupt requests of the same priority level can be accepted only in interrupt requests by slaves which permet complete nesting operations.

Care should be taken when issuing an FI (finish interrupt) command in the extended nesting mode. Upon an interrupt by a slave, the CPU first issued as FI command to the slave. Then, the CPU reads the slave's inservice register (ISR) to see if that slave still has interrupts in service. If there are no interrupts in service (ISR=00H), an FI command can be issued to the master, as in the single mode when as interrupt is made by a peripheral.

Exceptional Nesting Mode

A CXQ71059 in the normal or extended nesting mode cannot accept interrupts of a lower priority than the interrupt in service. Sometimes, however, it is desirable that requests with lower priority be accepted while higher-priority interrupt is being serviced. It is enabled by using the exceptional nesting mode. Once the exceptional nesting mode is entered, it is retained until reset. When it is reset, the previous nesting mode is resumed.

The exceptional nesting mode is programmed by SNM (set nesting mode) and EXCN (exceptional nesting mode) (bit D_6 and D_5 of MCW), used in pairs. They set and clear the exceptional nesting mode. The mode will not changed when SNW=0. Excepitonal nesting mode is entered if EXCN= and cleared if EXCN=0 when SNM=1.

In the exceptional nesting mode IMR masks the corresponding bits of IRR and ISR (though in other mode only IRR is masked). Therefore, if a bit in the IMR is set to a 1 in the exceptional nesting mode, it inhibits interrupts to that level and any other masked bit, but it allows interrupts to all other levels, higher or lower priority.

Cautions must be taken when issuing an FI command to a level masked in the exceptional nesting mode. Since the ISR bit is masked, the normal FI command will not work. For this reason, a specific FI command specifying the ISR bits must be issued. After the exceptional mode is released, the normal FI command may be used.

The procedure for the exceptional nesting mode is as follows:

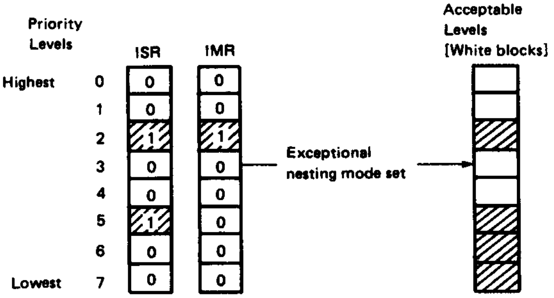
(1) Read the ISR.

(2) Write the read data to the IMR.

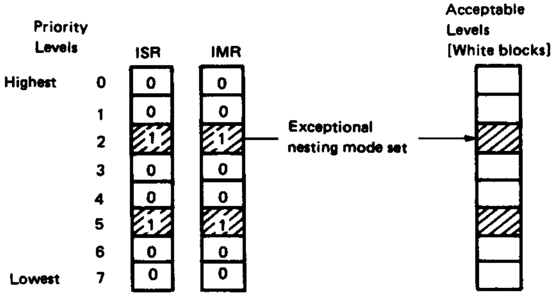
(3) Set the exceptional nesting mode.

In this way, all interrupt requests not currently in service will be enabled. See figure 16.

Figure 16. Exceptional Nesting Mode



Requests from INTP$_6$ and INTP$_7$ cannot be accepted when only bit 2 of the IMR is set to 1.



All requests other than those being executed can be accepted when the IMR is set the same as the ISR, in exceptional nesting mode.

Finishing Interrupts and Changing the Priority Levels

The priority and finish control word (PFCW) issues FI commands and changes interrupt priorities.

Normal FI Command

	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
PFCW=	0	0	1	0	0	X	X	X

When a normal FI command is issued, the CXQ71059 resets the ISR bit corresponding to the highest priority level selected from the interrupts in service, This operation confirms to the CXQ71059 that the highest priority routine (assuming it is the last) of the routines in service is finished. If an interrupt routine changes the priority level or the exceptional nesting mode is used, this command will not operate correctly because the highest priority interrupt is not necessarily the last interrupt in service.

Specific FI Command

	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
PFCW=	0	1	1	0	0	IL_2	IL_1	IL_0

When the specific FI command is issued, the CXQ71059 resets the ISR bit designated by bits IL_2—IL_0 of the PFCW. This command is used when the CXQ71059 cannot automatically determine the level of a completed service routine.

Self-FI Mode

When SFI (of IW4)=1, the CXQ71059 is set to the self-FI mode. In this mode, the ISR bit corresponding to the interrupt is set at the leading edge of the last $\overline{\text{INTAK}}$ pulse in the $\overline{\text{INTAK}}$ sequence. The same bit is reset on the trailing edge of the last INTAK pulse. Therefore, the CPU does not have to issue an FI command when the interrupt routine ends. In this mode, however, the ISR stores no information to specify the routine in service. Unless interrupts are disable by the interrupt routine, newly generated interrupt requests are granted without priority limitation by the ISR. This can cause a stack overflow when frequent interrupt requests occur, or when the interrupt is level-trigger

Self-FI Rotation

With Rotation:

	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
PFCW=	1	0	0	0	0	X	X	X

Without Rotation:

	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
PFCW=	0	0	0	0	0	X	X	X

Rotation of interrupt priorities can be added to the self-FI mode. In this case, the corresponding interrupt is assigned to the lowest priority level when the bit is reset in the ISR at the end of the $\overline{\text{INTAK}}$ sequence.

Normal Rotation FI Command

	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
PFCW=	1	0	1	0	0	X	X	X

When the normal rotation FI command is issued, the CXQ71059 resets the ISR bit corresponding to the highest priority level selected from the interrupts in service, then rotates the priority levels so that the interrupt just completed becomes the lowest priority.

Specific Rotation FI Command

	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
PFCW=	1	1	1	0	0	IL_2	IL_1	IL_0

When the specific rotation FI command is issued, the CXQ71059 resets the ISR bit designated by bits IL_2—IL_0 of the PFCW and rotates the interrupt priorities so that the interrupt just reset becomes the lowest priority. As this change in priority levels is different from the normal nesting mode, it is the user's responsibility to manage nesting.

Specific Rotation Command

	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
PFCW=	1	1	0	0	0	IL_2	IL_1	IL_0

When the specific rotation command is issued, the CXQ71059 specifies the interrupt designated by bits IL_2—IL_0 of the PFCW to the lowest priority and rotates the priority levels. In this case also, the user must manage nesting.

Triggering Mode

Bit D_3 of the first initilalization word, IW1, is LEV (level-trigger mode bit). LEV defines the trigger mode of the INTP input. The level-trigger mode is set when LEV=1. The edge-trigger mode is set when LEV=0.

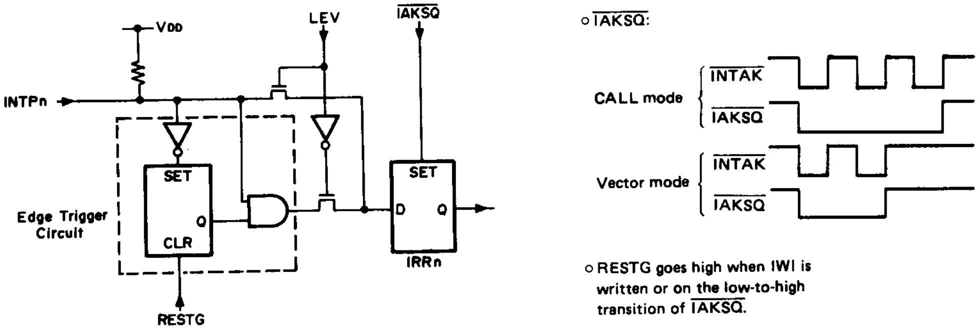
Edge-Trigger Mode

In the edge-trigger mode, an interrupt is detected by the rising edge of the signal on an INTP input. Although the corresponding IRR bit becomes '1' when INTP is high, the IRR bit is not latched until after the CPU returns an $\overline{\text{INTAK}}$ pulse. The INTP input must be maintained high until after $\overline{\text{INTAK}}$ is received. The IRR bit will be unlatched only after the INTP input goes low. To send the next interrupt request, temporarily lower the INTP input, then raise it.

Level-Trigger Mode

In the level-trigger mode, an IRR bit is set by the INTP input being at high level. As in the edge-trigger mode, the INTP must be maintained high until after the $\overline{\text{INTAK}}$ is received. Interrupts are requested as long as the INTP input remains high. Care should be taken so as not to cause a stack overflow in the CPU. See figure 17.

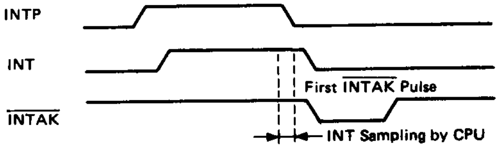
Figure 17. INTP Input



o RESTG goes high when IWI is written or on the low-to-high transition of $\overline{\text{IAKSQ}}$.

Note: In both the edge- and level-trigger modes the INTP requests must be maintained high until after the $\overline{\text{INTAK}}$ is received. If on any INTP inputs the CXQ71059 INT output goes low before the first $\overline{\text{INTAK}}$ pulse, the CXQ71059 operates as if the INTP$_7$ interrupt occurred. This incomplete interrupt request will not set the bit 7 of ISR. In case this will occur, the interrupt routine for the INTP$_7$ must see whether the current request is a complete one or not by reading the ISR. The FI command should not be issued for incomplete interrupts. See figure 18.

Figure 18. Incomplete Interrupt Request



Reading Registers

The contents of the internal registers, IRR, ISR, and IMR, can be read.

The IMR can be read when a direct read with $A_0=1$ is done.

To read the IRR or ISR, MCW must be issued to select either the IRR or ISR. Then a read operation with $A_0=0$ will issue the data of the selected register. Once the MCW is issued, the CXQ71059 will retain the selection. Therefore, no MCW is required when the same register with the previous one is to be read. The data of the IRR or ISR to be read is not masked by the IMR.

Polling Operation

When polling, the CPU should first disable its INT input. Next, it issues a polling command to the CXQ71059 using MCW with POL=1. This command sets the CXQ71059 in the polling mode until the CPU reads one of the CXQ71059's registers. When the CPU performs a read operation with $A_0=0$ in the polling mode, polling data as shown in figure 19 is read instead of ISR or IRR. The CXQ71059 then ends the polling mode.

If INT in the polling data is 1, the CXQ71059 sets the ISR bit corresponding to the interrupt level shown by bits PL_2-PL_0 of the polling data and considers that interrupt as being executed.The CPU then processes the interrupt routine accordingly, based on the polling data read. An FI command should be issued when this processing ends.

Note: When a read is performed with $A_0=1$ after the polling command is sent to the CXQ71059, the IMR will be read instead of polling data. However, after the polling command is issued, the CXQ71059 operates in the same manner when $A_0=1$ as it would when $A_0=0$. This means that although A_0 was set to 1, the CXQ71059 will send the contents of the IMR, but it will also set an ISR bit just as it would if A_0 had been set to zero. This may disturb the nesting. Therefore, performing a read operation with $A_0=1$ immediately after sending the polling command should be avoided.

Figure 19. Polling Data

	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
MCW =	INT	0	0	0	0	PL_2	PL_1	PL_0

INT (Interrupt)

This bit has the same meaning as the INT pin. When it is set to 1, it means that the CXQ71059 has accepted an INTP input.

PL_2-PL_0 (Permitted Level)

These bits show which INTP input requested an interrupt when INT=1.

Package Outline

Unit: mm