SGS-THOMSON MICROELECTRONICS

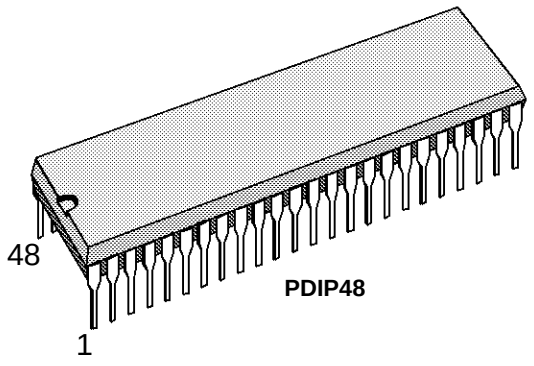
TS68HC901

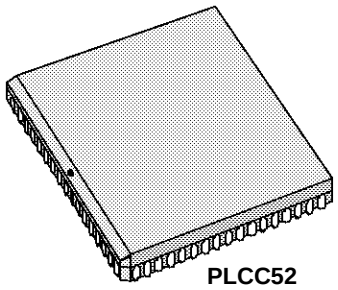
HCMOS MULTI-FUNCTION PERIPHERAL

The TS68HC901 CMFP (CMOS Multi-Function Peripheral) is a combination of many of the necessary peripheral functions in a microprocessor system.

Included are :

- 8 INPUT/OUTPUT PINS
  - Individually programmable direction
  - Individual interrupt source capability
    - Programmable edge selection
- 16 SOURCE INTERRUPT CONTROLLER
  - 8 Internal sources
  - 8 External sources
  - Individual source enable
  - Individual source masking
  - Programmable interrupt service modes
    - Polling
    - Vector generation
    - Optional In-service status
  - Daisy chaining capability
- FOUR TIMERS WITH INDIVIDUALLY PROGRAMMABLE PRESCALING
  - Two multimode timers
    - Delay mode
    - Pulse width measurement mode
    - Event counter mode
  - Two delay mode timers
  - Independent clock input
  - Time out output option
- SINGLE CHANNEL USART
  - Full Duplex
  - Asynchronous to 65 kbps
  - Byte synchronous to 1 Mbps
  - Internal/External baud rate generation
  - DMA handshake signals
  - Modem control
  - Loop back mode
- 68000 BUS COMPATIBLE





(Ordering Information at the end of the Datasheet

DESCRIPTION

The use of the CMFP in a system can significantly reduce chip count, thereby reducing system cost. The CMFP is completely 68000 bus compatible, and 24 directly addressable internal registers provide the necessary control and status interface to the programmer.

The CMFP is a derivative of the MK3801 STI, a Z80 family peripheral.

INTRODUCTION

The TS68HC901 multi-function peripheral (CMFP) is a member of the 68000 peripherals. The CMFP directly interfaces to the 68000 processor via an asynchronous bus structure. Both vectored and polled interrupt schemes are supported, with the CMFP providing unique vector number generation for each of its 16 interrupt sources. Additionally, handshake lines are provided to facilitate DMAC interfacing. Refer to block diagram of the TS68HC901.

The TS68HC901 performs many of the functions common to most microprocessor-based systems.

The resources available to the user include:

- Eight Individually Programmable I/O Pins with Interrupt Capability
- 16-Source Interrupt Controller with Individual Source Enabling and Masking
- Four Timers, Two of which are Multi-Mode Timers
- Timers may be used as Baud Rate Generators for the Serial Channel
- Single-Channel Full-Duplex Universal Synchronous / Asynchronous Receiver-Transmitter (USART) that Supports Asynchronous and with the Addition of a Polynomial Generator Checker Supports Byte Synchronous Formats

By incorporating multiple functions within the CMFP, the system designer retains flexibility while minimizing device count.

From a programmer's point of view, the versatility of the CMFP may be attributed to its register set. The registers are well organized and allow the CMFP to be easily tailored to a variety of applications. All of the 24 registers are also directly addressable which simplifies programming. The register map is shown in Figure 2.

Figure 1: TS68HC901 Block Diagram

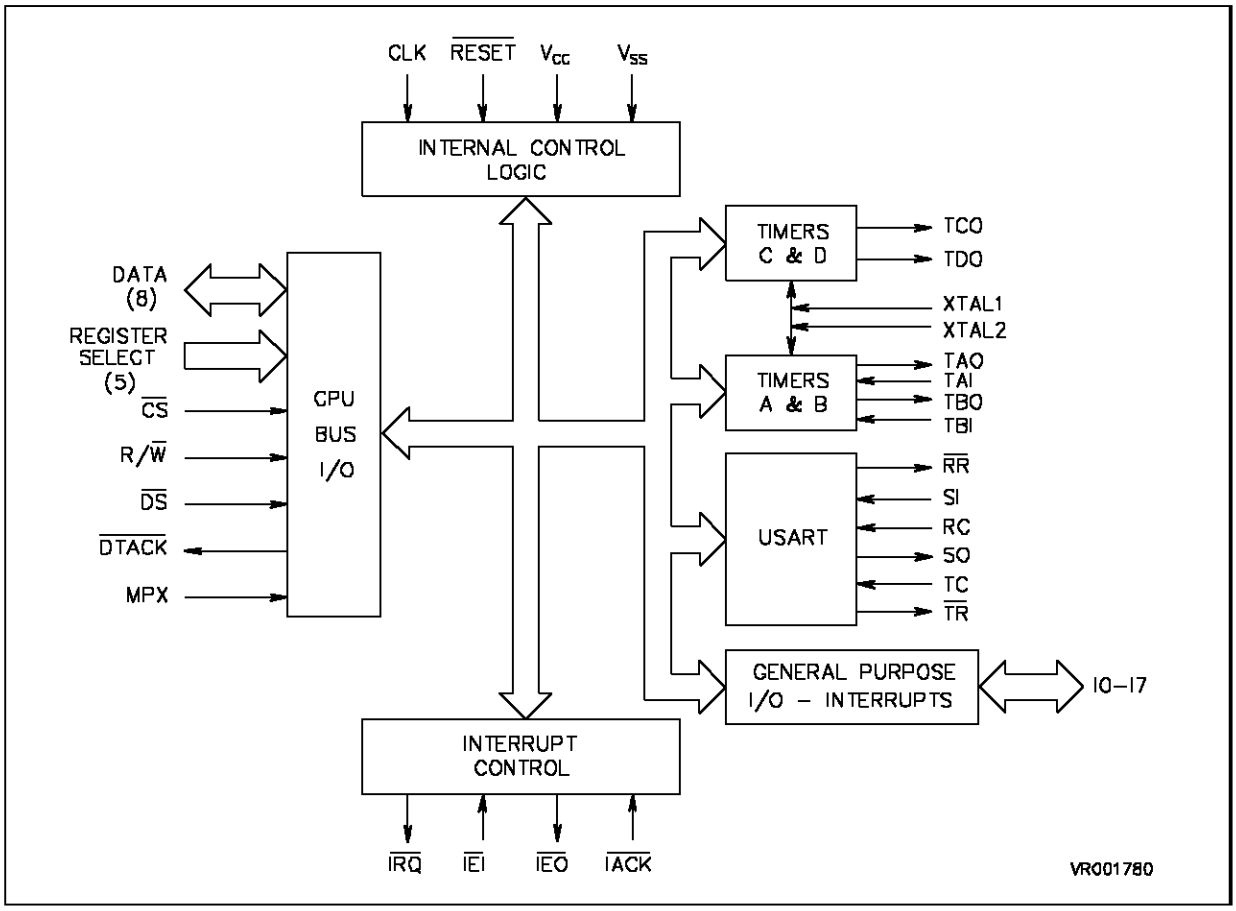


Figure 2 : CMFP Register Map.

Address						Abbreviation	Register Name
Hex	Binary						
	RS5	RS4	RS3	RS2	RS1		
01	0	0	0	0	0	GPIP	General Purpose I/O Register
03	0	0	0	0	1	AER	Active Edge Register
05	0	0	0	1	0	DDR	Data Direction Register
07	0	0	0	1	1	IERA	Interrupt Enable Register A
09	0	0	1	0	0	IERB	Interrupt Enable Register B
0B	0	0	1	0	1	IPRA	Interrupt Pending Register A
0D	0	0	1	1	0	IPRB	Interrupt Pending Register B
0F	0	0	1	1	1	ISRA	Interrupt In-service Register A
11	0	1	0	0	0	ISRB	Interrupt In-service Register B
13	0	1	0	0	1	IMRA	Interrupt Mask Register A
15	0	1	0	1	0	IMRB	Interrupt Mask Register B
17	0	1	0	1	1	VR	Vector Register
19	0	1	1	0	0	TACR	Timer A Control Register
1B	0	1	1	0	1	TBCR	Timer B Control Register
1D	0	1	1	1	0	TCDCR	Timers C and D Control Register
1F	0	1	1	1	1	TADR	Timer A Data Register
21	1	0	0	0	0	TBDR	Timer B Data Register
23	1	0	0	0	1	TCDR	Timer C Data Register
25	1	0	0	1	0	TDDR	Timer D Data Register
27	1	0	0	1	1	SCR	Synchronous Character Register
29	1	0	1	0	0	UCR	USART Control Register
2B	1	0	1	0	1	RSR	Receiver Status Register
2D	1	0	1	1	0	TSR	Transmitter Status Register
2F	1	0	1	1	1	UDR	USART Data Register

Note : Hex addresses assume that RS1 connects with A1, RS2connects with A2, etc... and that DS is connected to LDS on the 68000 or DS is connect to DS on the 68008.

Figure 3 : PDIP Pin connection

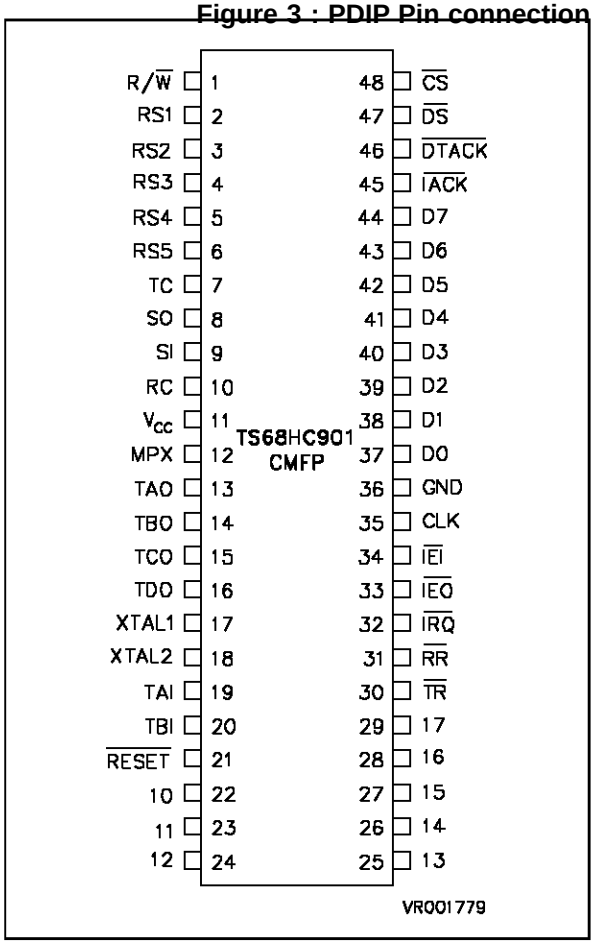
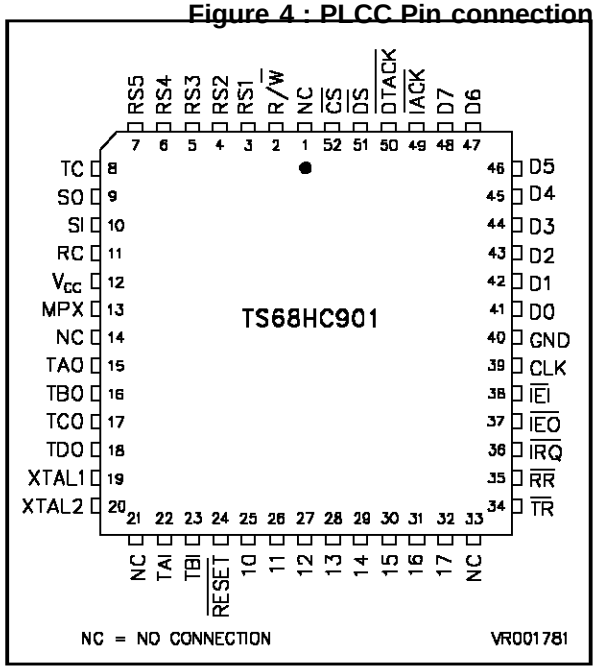


Figure 4 : PLCC Pin connection



Pin	MOTOROLA 6800 Type	MOTOROLA Multiplexed	INTEL
48	$\overline{CS}$	$\overline{CS}$	$\overline{CS}$
47	E	$\overline{DS}$	$\overline{RD}$
1	R/$\overline{W}$	R/$\overline{W}$	$\overline{WR}$
35	V_{SS}	AS	ALE

PIN DESCRIPTION

GND : Ground

V_{CC} : +5 volts (± 5%)

R/$\overline{W}$: Read/Write (input). This input defines a data transfert as a Read (High) or Write (Low) cycle. This signal is used as $\overline{WR}$ with an Intel processor type.

$\overline{DTACK}$: This output signals the completion of the operation phase of a bus cycle to the processor. If the bus cycle is a processor read, the CMFP asserts DTACK to indicate that the information on the Data bus is valid. If the bus cycle is a processor to the CMFP, DTACK acknowledges the acceptance of the data by the CMFP. DTACK will be asserted only by an CMFP that has CS or IAK (and IEI) asserted. This signal is not used with a 6800 processor type.

$\overline{CS}$: Chip Select (input, active low). $\overline{CS}$ is used to select the TS68HC901 CMFP for accesses to the internal registers. $\overline{CS}$ and IACK must not be asserted at the same time.

$\overline{DS}$: Data Stobe (input, active low).This Input is part of the internal chip select and interrupt acknowledge functions.
The CMFP must be located on the lower portion of the 16-bit data-bus so that the vector number passed to the processor during an interrupt acknowledge cycle will be located in the low byte of the data word. As a result, DS must be connected to the processor's lower data strobe if vectored interrupt are to be used. Note that this forces all registers to be located at odd addresses and latches data on the rising edge for writes. This signal is used as RD with an Intel processor type.

RS1-RS5: (A1-A5) Register Address Bus (inputs). The address bus is used to address one of the internal registers during a read or write cycle.

D0-D7 : Data Bus (bi-directional, tri-stateable). This bus is used to receive data from or transmit data to the MFP's internal registers during a processor read or write cycle. During an interrupt acknowledge cycle, the data bus is used to pass a vector number to the processor. Since the MFP is an 8-bit peripheral, the MFP could be located on either the upper or lower portion of the 16-bit data bus (even or odd address). However, during an interrupt acknowledge cycle, the vector number passed to the processor must be located in the low byte of the data word. As a result, D0-D7 of the MFP must be connected to the low eight bits of the processor data bus, placing MFP registers at odd addresses if vectored interrupt are to be used.

CLK : The clock input is a single-phase TTL compatible signal used for internal timing . This input should not be gated off at any time and must conform to minimum and maximum pulse width times. The clock is not necessarily the system clock in frequency nor phase. When the bus is multiplexed (MPX=1), an address strobe signal is connected to this pin. In the non multiplexed mode (MPX=0), this input is connected to the system clock when used with a 68000 processor type or to V_{SS} ($0V_{DC}$) when used with a 6800 processor type.

$\overline{RESET}$: Device reset. (input, active low). Reset disables the USART receiver and transmitter, stops all timers and forces the timer outputs low, disables all interrupt channels and clears any pending interrupts. The General Purpose Interrupt/I/O lines will be placed in the tri-state input mode. All internal registers (except the timer, USART data registers, and transmit status register) will be cleared.

MPX : This input selects the data bus mode:

MPX = 0 : non multiplexed mode

MPX = 1 : multiplexed mode. The register select lines RS1-RS5 and the data bus D0-D7 are multiplexed. An address strobe must be connected to the CLK pin.

$\overline{IRQ}$: Interrupt Request (output, active low, open drain). This output signals the processor that an interrupt is pending from the CMFP. These are 16 interrupt channels that can generate an interrupt request. Clearing the interrupt pending registers (IPRA and IPRB) or clearing the interrupt mask registers (IMRA and IMRB) will cause $\overline{IRQ}$ to be negated. $\overline{IRQ}$ will also be negated as the result of an interrupt acknowledge cycle, unless additional interrupts are pending in the CMFP. Refer to paragraph INTERRUPTS for further information.

$\overline{IACK}$: Interrupt Acknowledge (input, active low). $\overline{IACK}$ is used to signal the TS68HC901 that the CPU is acknowledging an interrupt. $\overline{CS}$ and IACk must not be asserted at the same time.

$\overline{IEI}$: Interrupt Enable In (input, active low). $\overline{IEI}$ is used to signal the TS68HC901 that no higher priority device is requesting interrupt service.

$\overline{IEO}$: Interrupt Enable Out (output, active low). $\overline{IEO}$ is used to signal lower priority peripherals that neither the TS68HC901 nor another higher priority peripheral is requesting interrupt service.

I0-I7 : General Purpose Interrupt I/O lines. These lines may be used as interrupt inputs and/or I/O lines. When used as interrupt inputs, their active edge is programmable. A data direction register is used to define which lines are to be Hi-Z inputs and which lines are to be push-pull TTL compatible outputs.

SO : Serial Output. This is the output of the USART transmitter.

SI : Serial Input. This is the input to the USART receiver.

RC : Receiver Clock. This input controls the serial bit rate of the USART receiver.

TC : Transmitter Clock. This input controls the serial bit rate of the USART transmitter.

$\overline{RR}$: Receiver Ready. (output, active low) DMA output for receiver, which reflects the status of Buffer Full in port number 15.

$\overline{TR}$: Transmitter Ready. (output, active low) DMA output for transmitter, which reflects the status of Buffer Empty in port number 16.

TAO,TBO, TCO,TDO: Timer Outputs. Each of the four timers has an output which can produce a square wave. The output will change states each timer cycle ; thus one full period of the timer out signal is equal to two timer cycles. TAO or TBO can be reset (logic "O") by a write to TACR, or TBCR respectively.

XTAL1, XTAL2 : Timer Clock inputs. A crystal can be connected between XTAL1 and XTAL2, or XTAL1 can be driven with a TTL level clock. When driving XTAL1 with a TTL level clock, XTAL2 must be allowed to float. When using a crystal, external capacitors are required. See figure 35. All chip accesses are independent of the timer clock.

TAI,TBI : Timer A, B inputs. These inputs are control signals for timers A and B in the pulse width measurement mode and event count mode. These signals generate interrupts at the same priority level as the general purpose I/O interrupt lines I4 and I3, respectively. I4 and I3 do not have interrupt capability when the timers are operated in the pulse width measurement mode or the event count mode - under these conditions I4 and I3 may only be used for I/O. Refer to paragraph TIMERS for further information.

SIGNAL SUMMARY.

Signal Name	Mnemonic	I/O	Active
Power Input	V_{CC}	Input	High
Ground	GND	Input	Low
Clock	CLK	Input	N/A
Chip Select	$\overline{CS}$	Input	Low
Data Strobe	$\overline{DS}$	Input	Low
Read/Write	R/$\overline{W}$	Input	Read-High / Write-Low
Data tranfer Acknowledge	$\overline{DTACK}$	Output	Low
Register Select Bus	RS1-RS5	Input	N/A
Data Bus	D0-D7	I/O	N/A
Reset	$\overline{RESET}$	Input	Low
Interrupt Request	$\overline{IRQ}$	Output	Low
Interrupt Acknowledge	$\overline{IACK}$	Input	Low
Interrupt Enable In	$\overline{IEI}$	Input	Low
Interrupt Enable Out	$\overline{IEO}$	Output	Low
General Purpose I/O - Interrupt Lines	I0-I7	I/O	N/A
Timer Clock	XTAL1, XTAL2	Input	High
Timer Inputs	TAI, TBI	Input	N/A
Timer Outputs	TAO, TBO, TCO, TDO	Output	N/A
Serial Input	SI	Input	N/A
Serial Output	SO	Output	N/A
Receiver Clock	RC	Input	N/A
Transmitter Clock	TC	Input	N/A
Receiver Ready	$\overline{RR}$	Output	Low
Transmitter Ready	$\overline{TR}$	Output	Low
MPX	$\overline{MPX}$	Input	N/A

BUS OPERATION

The following paragraphs explain the control signals and bus operation during data transfer operations and reset.

DATA TRANSFER OPERATIONS.

Transfer of data between devices involves the following pins: Register Select Bus - RS5 through RS1 Data Bus - D0 through D7 Control Signals The address and data buses are separate parallel buses used to transfer data using an asynchronous bus structure. In all cycles, the bus master assumes responsibility for deskewing all signals it issues at both the start and end of a cycle. Additionally, the bus master is responsible for deskewing the acknowledge and data signals from the peripheral devices.

Read Cycle. To read a CMFP register, $\overline{CS}$ and $\overline{DS}$ must be asserted, and R/$\overline{W}$ must be high. The CMFP will place the content of the register which is selected by the register select bus (RS1 through RS5) on the data bus (D1 through D7) and then assert $\overline{DTACK}$. The register addresses are shown on Figure 2. After the processor has latched the data, $\overline{DS}$ is negated. The negation of either $\overline{CS}$ or $\overline{DS}$ will terminate the read operation. The CMFP will drive $\overline{DTACK}$ High and place it in the high-impedance state. The timing for a read cycle is shown in figure 21.

Write Cycle. To write a register $\overline{CS}$ and $\overline{DS}$ must be asserted, and R/$\overline{W}$ must be low. The CMFP will decode the address bus to determine which register is selected. Then the register will be loaded with the contents of the data bus and $\overline{DTACK}$ will be asserted. When the processor recognizes $\overline{DTACK}$, $\overline{DS}$ will be negated. The write cycle is terminated when either $\overline{CS}$ or $\overline{DS}$ is negated. The CMFP will drive $\overline{DTACK}$ high and place it in the high-impedance state. The timing for a write cycle is shown in figure 22.

INTERRUPT ACKNOWLEDGE OPERATION.

The CMFP has 16 interrupt sources, eight internal and eight external. When an interrupt request is pending, the CMFP will assert $\overline{IRQ}$. In a vectored interrupt scheme, the processor will acknowledge the interrupt request by performing an interrupt acknowledge cycle. $\overline{IACK}$ and $\overline{DS}$ will be asserted. The CMFP responds to the $\overline{IACK}$ signal by placing a vector number on the lower eight bits of the data bus. This vector number corresponds to the $\overline{IRQ}$ handler for the particular interrupt requesting service. The format of this vector number is given in figure 6.

When the CMFP asserts $\overline{DTACK}$ to indicate that valid data is on the bus, the processor will latch the data and terminate the bus cycle by negating $\overline{DS}$. When either $\overline{DS}$ or $\overline{IACK}$ are negated, the CMFP will terminate the interrupt acknowledge operation by driving $\overline{DTACK}$ high and placing it in the high-impedance state. Also, the data bus will be placed in the high-impedance state. $\overline{IRQ}$ will be negated as a result of the $\overline{IACK}$ cycle unless additional interrupts are pending.

The CMFP can be part of a daisy-chain interrupt structure which allows multiple CMFPs to be placed at the same interrupt level by sharing a common $\overline{IACK}$ signal. A daisy-chain priority scheme is implemented with IEI and IEO signals. IEI indicates that no higher priority device is requesting interrupt service. IEO signals lower priority devices that neither this device nor any higher priority devices is requesting service. To daisy-chain CMFPs, the highest priority CMFP has its IEI tied low and successive CMFPs have their IEI connected to the next higher priority device's IEO. Note that when the daisy-chain

interrupt structure is not implemented, the $\overline{IEI}$ of all CMFPs must be tied low.

When the processor initiates an interrupt acknowledge cycle by driving IACK and $\overline{DS}$, the CMFP whose $\overline{IEI}$ is low may respond with a vector number if interrupt is pending. If this device does not have a pending interrupt, $\overline{IEO}$ is asserted which allows the next lower priority device to respond to the interrupt acknowledge. When an CMFP propagates IEO, it will not drive the data bus nor DTACK during the interrupt acknowledge cycle. The timing for an IACK cycle is shown in figure 23 and 24.

RESET OPERATION

The reset operation will initialize the CMFP to a known state. The reset operation requires that the RESET input be asserted for a minimum of two microseconds. During a device reset condition, all internal CMFP registers are cleared except for the timer data registers (TADR, TBDR, TCDR and TDDR), the USART data register (UDR), the transmitter status register (TSR) and the interrupt vector register. All timers are stopped and the USART receiver and transmitter are disabled. The interrupt channels are also disabled and any pending interrupts are cleared. In addition, the general purpose interrupt I/O lines are placed in the high- impedance input mode and the timer outputs are driven low. External CMFP signals are negated. The interrupt vector register is initialized to a 0Fh.

NON MULTIPLEXED MODE

In this mode, the MPX input must be set to zero, and the TS68HC901 can be used with a 68000 processor type or a 6800 processor type. Refer to figure 21 to 24 for the electrical characteristics.

With a 6800 processor type the $\overline{DS}$ pin is connected to the E signal of the processor, the $\overline{DTACK}$ signal is not used and the CLK must be zeroed.

MULTIPLEXED MODE

The CMFP can be used either on a MOTOROLA or INTEL bus type. In this case the MPX pin is connected to Vcc. The table page 4 gives the signification of the different signals used. A dummy access to the TS68HC901 has to be done before any valid access in order to set up the internal logic of sampling.

INTERRUPT STRUCTURE

In a 68000 system, the CMFP will be assigned to one of the seven possible interrupt levels. All interrupt service requests from the CMFP's 16 interrupt channels will be presented at this level. Although, as an interrupt controller, the CMFP will internally prioritize its 16 interrupt sources. Additional interrupt sources may be placed at the same interrupt level by daisy-chaining multiple CMFPs. The CMFPs will be prioritized by their position in the chain.

INTERRUPT PROCESSING

Each CMFP provides individual interrupt capability for its various functions. When an interrupt is received on one of the external interrupt channels or from one of the eight internal sources, the CMFP will request interrupt service. The 16 interrupt channels are assigned a fixed priority so that multiple pending interrupts are serviced according to their relative importance. Since the CMFP can internally generate 16 vector numbers, the unique vector number which corresponds to the highest priority channel that as a pending interrupt is presented to the processor during an interrupt acknowledge cycle. This unique vector number allows the processor to immediately begin execution of the interrupt handler for the interrupt source, decreasing interrupt latency time.

INTERRUPT CHANNEL PRIORITIZATION

The 16 interrupt channels are prioritized as shown in figure 5. General purpose interrupt 7 (I7) is the highest priority interrupt channel and I0 is the lowest priority channel. Pending interrupts are presented to the CPU in order of priority unless they have been masked off. By selectively masking interrupts, the channel are in effect re-prioritized.

INTERRUPT VECTOR NUMBER FORMAT

During an interrupt acknowledge cycle, a unique 8-bit vector number is presented to the system which corresponds to the specific interrupt source which is requesting service. The format of the vector is shown in figure 6. The most significant four bits of the interrupt vector number are user programmable. These bits are set by writing the upper four bits of the vector register which is shown in figure 7. The low order bits are generated internally by the TS68HC901. Note that the binary channel number shown in figure 5 corresponds to the low order bits of the vector number associated with each channel.

Figure 5 : Interrupt Channel Prioritization

Figure 9 : Interrupt Channel Prioritization

Priority	Channel	Description
HIGHEST	1111	General Purpose Interrupt 7(I7)
	1110	General Purpose Interrupt 6(I6)
	1101	Timer A
	1100	Receive Buffer Full
	1011	Receive Error
	1010	Transmit Buffer Empty
	1001	Transmit Error
	1000	Timer B
	0111	General Purpose Interrupt 5(I5)
	0110	General Purpose Interrupt 4(I4)
	0101	Timer C
	0100	Timer D
	0011	General Purpose Interrupt 3(I3)
	0010	General Purpose Interrupt 2(I2)
	0001	General Purpose Interrupt 1(I1)
LOWEST	0000	General Purpose Interrupt 0(I0)

Figure 6 :

V7	V6	V5	V4	IV3	IV2	IV1	IV0

V7-V4 The four most significant bits are copied from the register

IV3-IV0 These bits are supplied by the CMFP. They are the binary channel number of the highest priority channel that is requesting interrupt service.

Figure 7 :

VECTOR REGISTER

	7							0
VR (17h)	V7	V6	V5	V4	S	0*	0*	0*

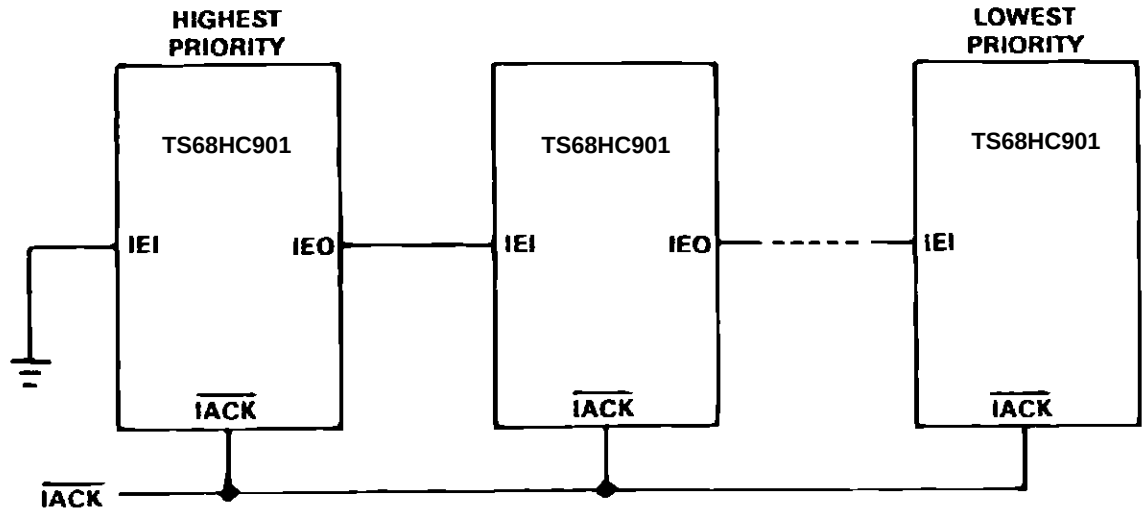
Writing 0 : CLEARED
Writing 1 : SET
CLEARED on RESET

V7-V4 The upper four bits of the vector register are written by the user. These bits become the most significant four bits of the interrupt vector number.

S In-Service Register Enable. When the S bit is zero, the CMFP is in the automatic end-of-interrupt mode and the In-Service register bits are forced low. When the S bit is a one, the CMFP is in the software end-of-interrupt mode and the In-Service register bits are enabled.

* Unused bits, read as zero.

Figure 8 : Daisy Chaining



DAISY-CHAINING CMFPs

As an interrupt controller, the TS68HC901 CMFP will support eight external interrupt sources in addition to its eight internal interrupt sources. When a system requires more than eight external interrupt sources to be placed at the same interrupt level, sources may be added to the prioritized structure by daisy-chaining CMFPs. Interrupt sources are prioritized internally within each CMFP and the CMFPs are prioritized by their position in the chain. Unique vector numbers are provided for each interrupt sources.

The $\overline{IEI}$ and $\overline{IEO}$ signals implement the daisy-chained interrupt structure. The $\overline{IEI}$ of the highest priority CMFP is tied low and the $\overline{IEO}$ output of this device is tied to the next highest priority CMFP's $\overline{IEI}$. The $\overline{IEI}$ and $\overline{IEO}$ signals are daisy-chained in this manner for all CMFPs in the chain, with the lowest priority CMFP's $\overline{IEO}$ left unconnected. A diagram of an interrupt daisy-chain is shown in figure 8.

Daisy-chaining requires that all parts in the chain have a common $\overline{IACK}$. When the common $\overline{IACK}$ is asserted during an interrupt acknowledge cycle, all parts will prioritize interrupts in parallel. When the $\overline{IEI}$ signal to a CMFP is asserted, the part may respond to the $\overline{IACK}$ cycle if it requires interrupt service. Otherwise, the part will assert $\overline{IEO}$ to the next lower priority device. Thus, priority is passed down the chain via $\overline{IEI}$ and $\overline{IEO}$ until a part which has appending interrupt is reached. The part with the pending interrupt passes a vector number to the processor and does not propagate $\overline{IEO}$.

Figure 9a :

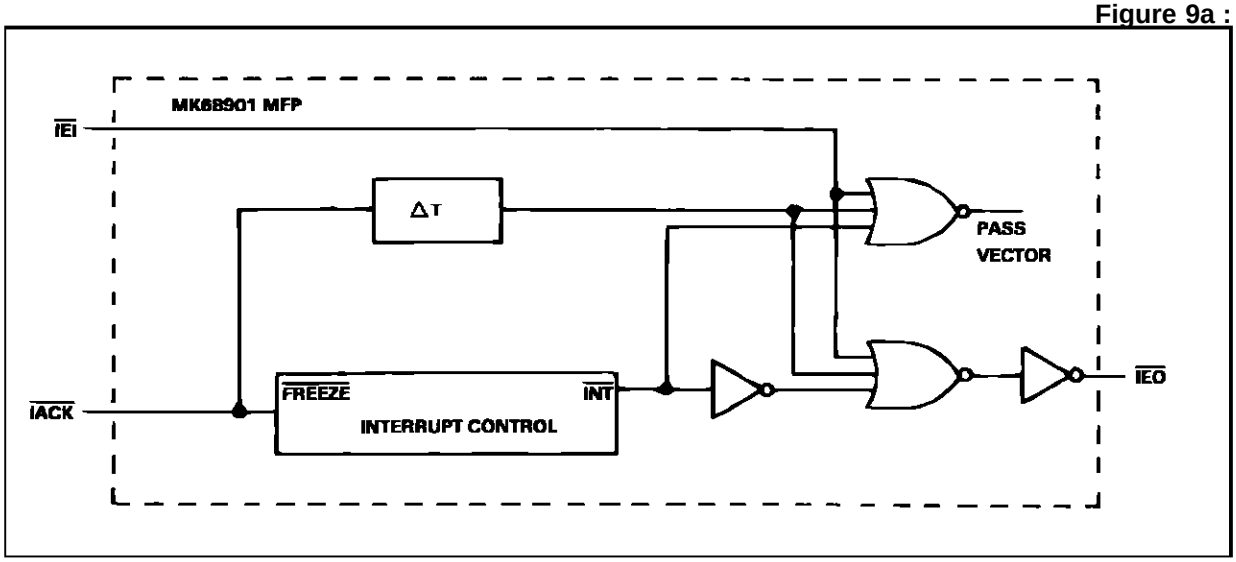
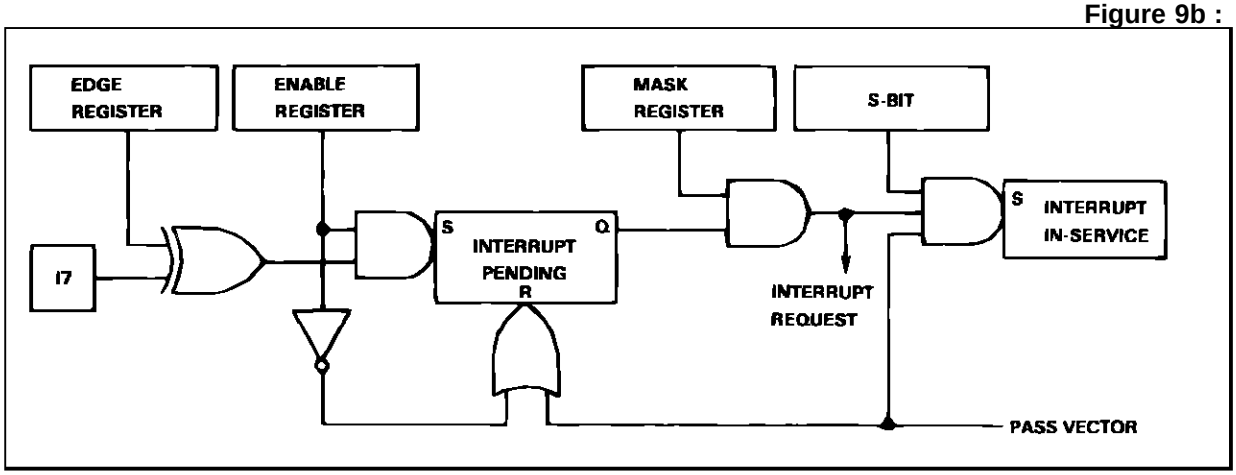


Figure 9b :



INTERRUPT CONTROL REGISTERS

CMPF interrupt processing is managed by the interrupt enable registers A and B, interrupt pending registers A and B, and interrupt mask registers A and B. These registers allow the programmer to enable or disable individual interrupt channels, mask individual interrupt channels, and access pending interrupt status information. In-service registers A and B allow interrupts to be nested as described hereafter. The interrupt control registers are shown in figure 10.

INTERRUPT ENABLE REGISTERS

The interrupt channels are individually enabled or disabled by writing a one or zero, respectively, to the appropriate bit of interrupt enable register A (IERA) or interrupt enable register B (IERB). The processor may read these registers at any time.

When a channel is enabled, interrupts received on the channel will be recognized by the CMFP and IRQ will be asserted to the processor, indicating that interrupt service is required. On the other hand, a disabled channel is completely inactive; interrupts received on the channel are ignored by the CMFP.

Writing a zero to a bit of interrupt enable register A or B will cause the corresponding bit of interrupt pending register A or B to be cleared. This will terminate all interrupt service requests for the channel and also negate IRQ, unless interrupts are pending from other sources. Disabling a channel, however, does not affect the corresponding bit in interrupt in-service registers A or B. So, if the CMFP is in the software end-of-interrupt mode and an interrupt is in service when a channel will remain set until cleared by software.

INTERRUPT PENDING REGISTERS

When an interrupt is received on an enabled channel, the corresponding interrupt pending bit is set in interrupt pending register A or B (IPRA or IPRB). In a vectored interrupt scheme, this bit will be cleared when the processor acknowledges the interrupting channel and the CMFP responds with a vector number. In a polled interrupt system, the interrupt pending registers must be read to determine the interrupting channel and then the interrupting pending bit is cleared by the interrupt handling routine without performing an interrupt acknowledge sequence.

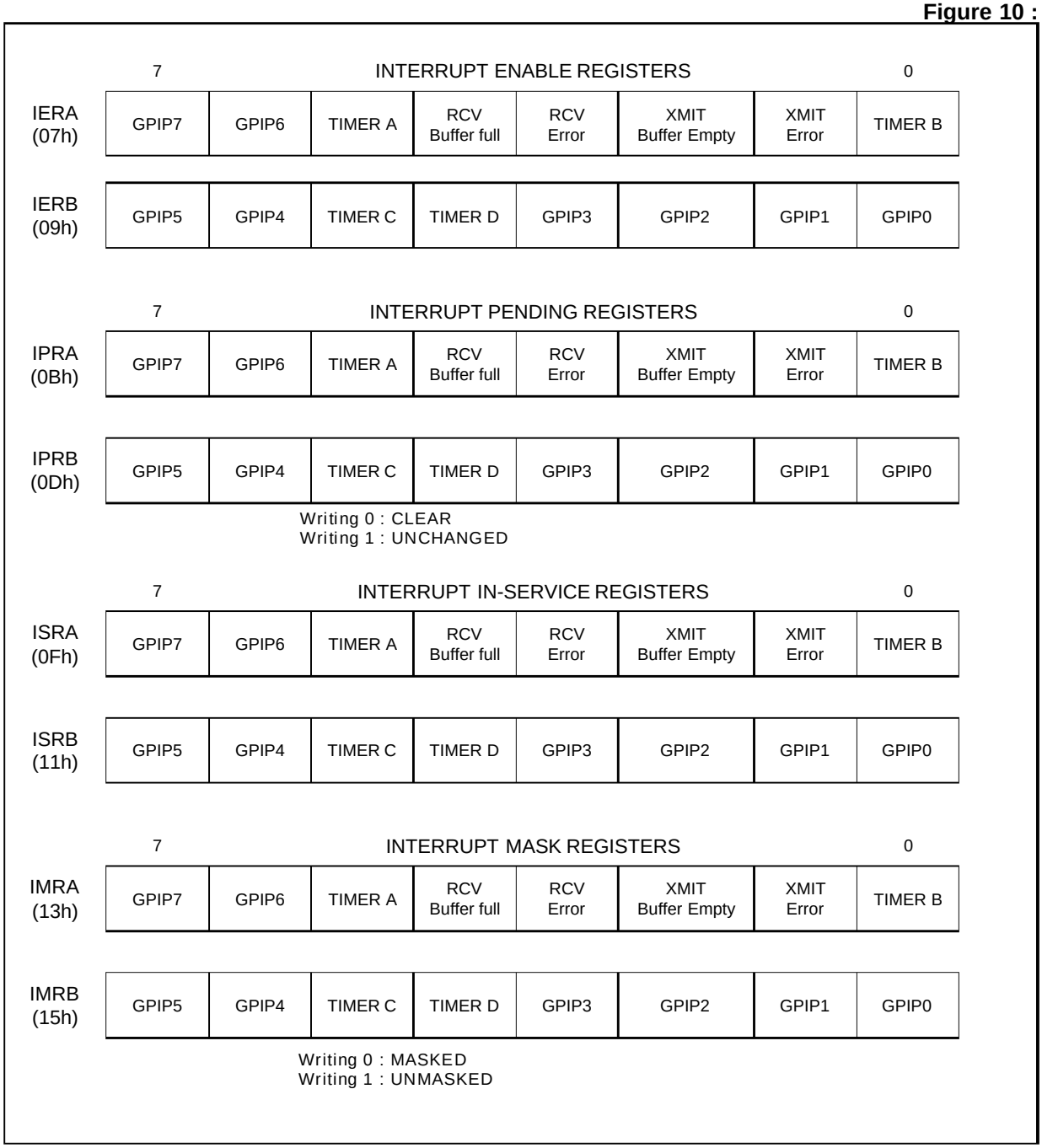
A single bit of the interrupt pending registers is cleared in software by writing ones to all bit positions except the bit to be cleared. Note that writing ones to IPRA and IPRB has no effect on the contents of the register. A single bit of the interrupt pending registers is also cleared when the corresponding channel is disabled by writing a zero to the appropriate bit of IERA or IERB.

INTERRUPT MASK REGISTERS

Interrupts are masked for a channel by clearing the appropriate bit in interrupt mask register A or B (IMRA or IMRB). Even though an enabled channel is masked, the channel will recognize subsequent interrupts and set its interrupt pending bit. However, the channel is prevented from requesting interrupt service (IRQ to the processor) as long as the mask bit for that channel is cleared.

If a channel is requesting interrupt service at the time that its corresponding bit in IMRA or IMRB is cleared, the request will cease and IRQ will be negated, unless another channel is requesting interrupt service. Later, when the mask bit is set, any pending interrupt on the channel will be processed according to the channel's assigned priority. IMRA and IMRB may be read at any time.

Figure 10 :


7
INTERRUPT ENABLE REGISTERS
0
IERA (07h)
GPIP7
GPIP6
TIMER A
RCV Buffer full
RCV Error
XMIT Buffer Empty
XMIT Error
TIMER B
IERB (09h)
GPIP5
GPIP4
TIMER C
TIMER D
GPIP3
GPIP2
GPIP1
GPIP0
7
INTERRUPT PENDING REGISTERS
0
IPRA (0Bh)
GPIP7
GPIP6
TIMER A
RCV Buffer full
RCV Error
XMIT Buffer Empty
XMIT Error
TIMER B
IPRB (0Dh)
GPIP5
GPIP4
TIMER C
TIMER D
GPIP3
GPIP2
GPIP1
GPIP0
Writing 0 : CLEAR
Writing 1 : UNCHANGED
7
INTERRUPT IN-SERVICE REGISTERS
0
ISRA (0Fh)
GPIP7
GPIP6
TIMER A
RCV Buffer full
RCV Error
XMIT Buffer Empty
XMIT Error
TIMER B
ISRB (11h)
GPIP5
GPIP4
TIMER C
TIMER D
GPIP3
GPIP2
GPIP1
GPIP0
7
INTERRUPT MASK REGISTERS
0
IMRA (13h)
GPIP7
GPIP6
TIMER A
RCV Buffer full
RCV Error
XMIT Buffer Empty
XMIT Error
TIMER B
IMRB (15h)
GPIP5
GPIP4
TIMER C
TIMER D
GPIP3
GPIP2
GPIP1
GPIP0
Writing 0 : MASKED
Writing 1 : UNMASKED

NESTING CMFP INTERRUPTS

In a 68000 vectored interrupt system, the CMFP is assigned to one of seven possible interrupt levels. When an interrupt is received from the CMFP, an interrupt acknowledge for that level is initiated. Once an interrupt is recognized at a particular level, interrupts at that same level or below are masked by 68000. As long as the processor's interrupt mask is unchanged, the 68000 interrupt structure will prohibit the nesting of interrupts at the same interrupt level. However, additional interrupt requests from the CMFP can be recognized before a previous channel's interrupt service routine is completed by lowering the processor's interrupt mask to the next lower interrupt level within the interrupt handler.

When nesting CMFP interrupts, it may be desirable to permit interrupts on any CMFP channel, regardless of its priority, to preempt or delay interrupt processing of an earlier channel's interrupt service request. Or, it may be desirable to only allow subsequent higher priority channel interrupt requests to supersede previously recognized lower priority interrupt requests. The CMFP interrupt structure provides this flexibility by offering two end-of-interrupt options for vectored interrupt schemes. Note that the end-of-interrupt modes are not active in a polled interrupt scheme.

SELECTING THE END-OF-INTERRUPT MODE

In a vectored interrupt scheme, the CMFP may be programmed to operate in either the automatic end-of-interrupt mode or the software end-of-interrupt mode. The mode is selected by writing the S bit of the vector register (see figure 7). When the S bit is programmed to a one, the CMFP is placed in the software end-of-interrupt mode and when the S bit is a zero, all channels operate in the automatic end-of-interrupt mode.

AUTOMATIC END-OF-INTERRUPT

When an interrupt vector number is passed to the processor during an interrupt acknowledge cycle, the corresponding channel's interrupt pending bit is cleared. In the automatic end-of-interrupt mode, no further history of the interrupt remains in the CMFP. The in-service bits of the interrupt in-service registers (ISRA and ISRB) are forced low. Subsequent interrupts which are received on any CMFP channel will generate an interrupt request to the processor, even if the current interrupt's service routine has not been completed.

SOFTWARE END-OF-INTERRUPT

In the software end-of-interrupt mode, the channel's associated interrupt pending bit is cleared and in addition, the channel's in-service bit of in-service register A or B is set when its vector number is passed to the processor during an IACK cycle. A higher priority channel may subsequently request interrupt service and be acknowledged, but as long as the channel's in-service bit is set, no lower priority channel may request interrupt service nor pass its vector during an interrupt acknowledge sequence.

While only higher priority channels may request interrupt service, any channel can receive an interrupt and set its interrupt pending bit. Even the channel whose in-service bit is set can receive a second interrupt. However, no interrupt service request is made until its in-service bit is cleared.

The in-service bit for a particular channel can be cleared by writing a zero to its corresponding bit in ISRA or ISRB and ones to all other bit positions. Since bits in the in-service registers can only be cleared in software and not set, writing ones to the register does not alter their contents. ISRA and ISRB may be read at any time.

GENERAL PURPOSE INPUT/OUTPUT INTERRUPT PORT

The general purpose interrupt input/output (I/O) port (GPIP) provides eight I/O lines (I0 through I7) that may be operated as either inputs or outputs under software control. In addition, these lines may optionally generate an interrupt on either a positive transition or negative transition of the input signal. The flexibility of the GPIP allows it to be configured as an 8-Bit I/0 port or for bit I/O. Since interrupts are enabled on a bit-by-bit basis, a subset of the GPIP could be programmed as handshake lines or the port could be connected to as many as eight external interrupt sources, which would be prioritized by the CMFP interrupt controller for interrupt service.

6800 INTERRUPT CONTROLLER

The CMFP interrupt controller is particularly useful in a system which has many 6800-type devices. Typically, in a vectored 68000 system, 6800-type peripherals use the autovector which corresponds to their assigned interrupt level since they do not provide a vector number in response to an AC cycle. The autovector interrupt handler must then poll all 6800-type devices at that interrupt level to determine which device is requesting service. However, by tying the IRQ output from a 6800-type device to the general purpose I/O interrupt port (GPIP) of a CMFP, a unique vector number will be provided to the processor during an interrupt acknowledge cycle. This interrupt structure will significantly reduce interrupt latency for 6800-type devices and other peripheral devices which do not support vector-by-device.

GPIP CONTROL REGISTERS

The GPIP is programmed via three control registers shown in figure 11. These registers control the data direction provide user access to the port, and specify the active edge for each bit of the GPIP which will produce an interrupt. These registers are described in detail in the following paragraphs.

GPIP DATA REGISTER

The general purpose I/O data register is used to input or output data to the port. When data is written to the GPIP data register, those pins which are defined as inputs will remain in the high-impedance state. Pins which are defined as outputs will assume the state (high or low) of their corresponding bit in the data register. When the GPIP is read, data will be passed directly from the bits of the data register for pins which are defined as outputs. Data from pins defined as inputs will come from the input buffers.

ACTIVE EDGE REGISTER

The active edge register (AER) allows each of the GPIP lines to produce an interrupt on either a one-to-zero or a zero-to-one transition. Writing a zero the appropriate edge bit of the active edge register causes the associated input to generate an interrupt on the one-to-zero transition. Writing a one to the edge bit will produce an interrupt on the zero-to-one transition of the corresponding GPIP line.