MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

Order this data sheet
by 68902/D

Military 68902

Advance Information

Enhanced Multi-Function Peripheral

The 68902 is a single chip peripheral device that allows a microprocessor system to perform the following system functions: serial and parallel data transfers, timer functions, interrupt control and general I/O handling.

AVAILABLE AS

1) JAN: N/A
2) SMD: N/A
3) 883: PLANNED
PGA: 68902/BZAJC
CLCC: 68902/BYCJC

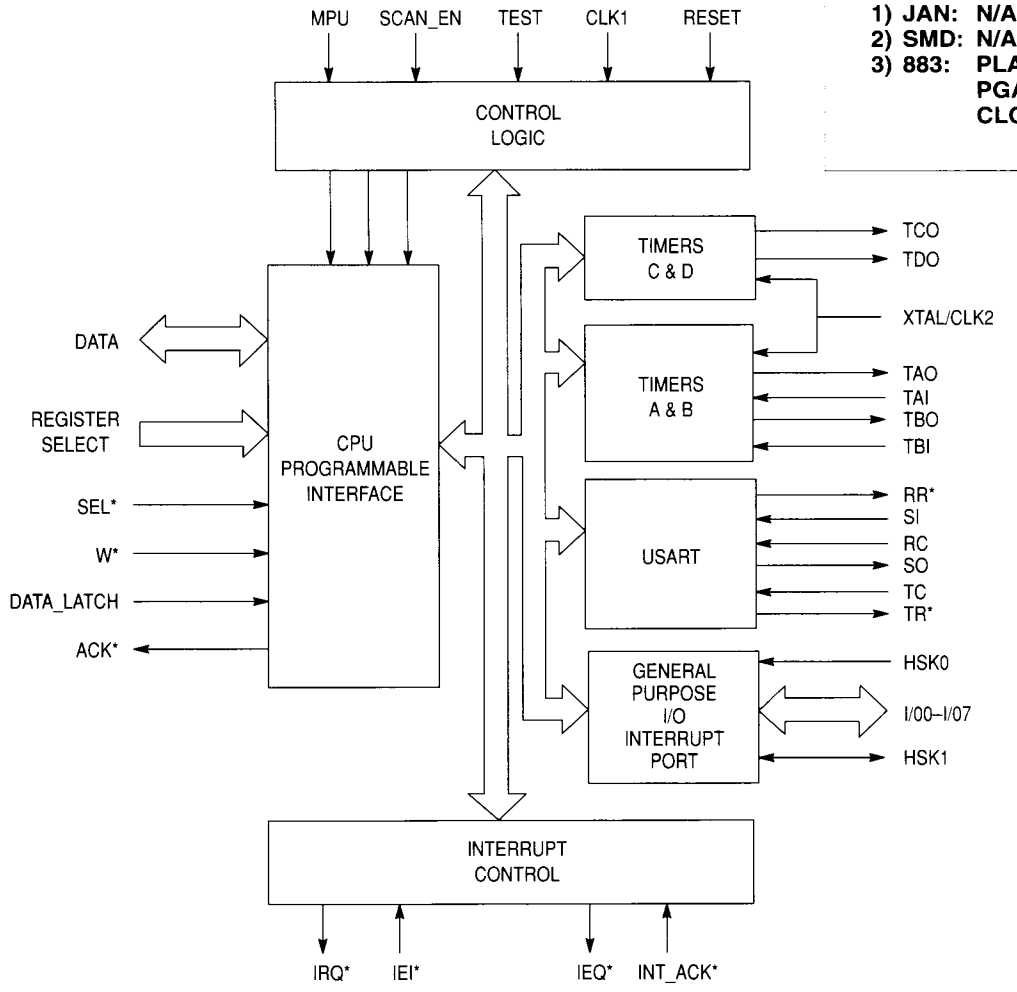


Figure 1. Block Diagram

This document contains information on a new product. Specifications and information herein are subject to change without notice.

INTRODUCTION

Features:

- Eight individually Programmable I/O pins with Interrupt capability. Two parallel handshaking lines can be used to control parallel data transfers over the 8-bit I/O port.
- The 68902 contains 16 sources of interrupts. These can be individually enabled/disabled or masked via the microprocessor interface. The interrupt controller can be programmed to supply individual vector numbers as used by the M68000 family of microprocessors.
- Four 8-bit timers with prescalers provide many of the timing functions typically required in microprocessor based applications. Two of the times can be used to divide a crystal frequency, or a clocked input, to derive baud rates for the serial transmitter and receiver channels. The timers can operate in delayed mode to produce waveform or generate interrupts at programmed time periods. In addition to delayed mode the timers can be programmed to operate in pulse width measurement and event count mode. In pulse width measurement an input is used to start a counter. When the input changes value this stops the counter. The value in the counter then represents the width of the pulse. In event count mode a value loaded into a timer is allowed to expire causing an output pin to toggle. The timer can then be programmed to operate in a polled interrupt environment.
- The 68902 contains a full-duplex Universal Synchronous/Asynchronous Receiver-Transmitter (USART) circuit capable of being clocked at up to 5 M bits/s. This USART contains a separate serial input and serial output character capable of being individually programmed with different baud rates.
- A versatile hardware select mechanism to allow synchronous interfacing to the M68000 family of microprocessors. The 68902 will operate initially at frequencies up to 20 MHz.

The 68902 offers a high performance integrated solution to interfacing to a number of peripheral functions in one device. In addition the 68902 offers an upward grad path for existing designs based on the 68901 or 68HC901. As far as the software is concerned the two devices are identical, in other words the same registers are used and the same bit position in the registers replicated.

A block schematic of the 68902 is shown in Figure 1. The 68902, although a complete function in its own right, is comprised of five ASIC mega functions with interconnect and control logic. Designing functional blocks as mega-functions allows them to be used separately in custom designs and allows a greater amount of functional independency. The five mega ASIC functions are listed below:

USART901

An ASIC macro of the Universal Synchronous/Asynchronous Receiver-Transmitter found on the MC68901 multi-functional peripheral. This macro is software compatible however is hardware enhanced to allow faster transmission rates.

TIMER901

An ASIC macro compatible to the timer found on the MC68901 multi-functional peripheral. This macro is software compatible however it has a number of hardware enhanced to allow higher clocking frequencies to be used.

INT901

This ASIC macro is functionality identical to the interrupt control logic found on the MC68901, with the exception of higher operating speeds. This interrupt logic allows an external interrupt to generate a vectored interrupt number and/or to prioritize interrupts. The interrupt controller allows multiple 68902 device interrupts to be daisy-chained effectively.

GPIP

General Purpose I/O Port. This is a general purpose 8-bit port with a three way multiplexor to allow the following functions: general I/O port with interrupt capability on correct edge detection; parallel port control with handshaking; and hardware control lines for CCITT V.24 protocol to be used in conjunction with the USART.

MPUINT

A general purpose interface allowing a number of Motorola microprocessors to be interfaced to the 68902. When a particular microprocessor is selected the interface is automatically reconfigured to allow it to operate with the minimum number of wait states and external glue logic.

SIGNAL AND BUS OPERATION DESCRIPTION

This section describes the various signals on the 68902 and the operation of the microprocessor interface bus for normal and interrupt acknowledge cycles.

NOTE

The term assertion and negation will be used extensively. This is done to avoid confusion when dealing with a mixture of "active low" and "active high" signals. The term "assert" or "assertion" is used to indicate that a signal is active or true, independent of whether that level is represented by a high or low voltage. The term "negate" or "negation" is used to indicate that a signal is inactive or false. If the signal has a '*' appended to it, e.g. W*, then this signal is active at the low voltage level. If no '*' is appended then the signal is active at the high voltage level.

SIGNAL DESCRIPTION

All input and output and bi-directional signals of the 68902 are shown in Figure 2. The 68902 is packaged in a 68 pin PLCC.

V_{DD} and V_{SS}

The total number of active pins on the 68902 is 52, however for good noise immunity and fast operating speeds the 68902 comes in a 68 pin package. Therefore the 68902 has a total of eight V_{DD} and V_{SS} pin pairs.

CLK1

This clock input is the system clock frequency. It is used to clock all the internal logic in the 68902. The 68902 has the capability to interface synchronously to microprocessors at up to 20 MHz. Higher microprocessor frequencies can be used if

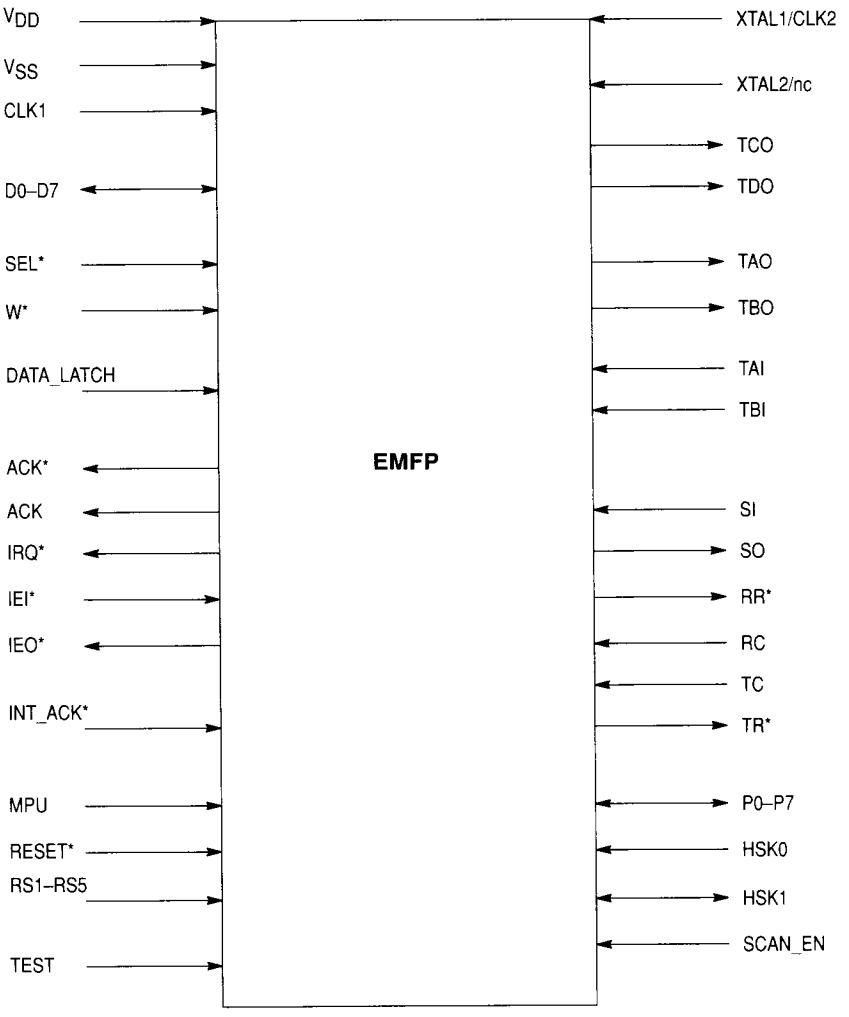


Figure 2. Pin Description

the 68902 is driven from a lower frequency clock derived from the microprocessor clock.

Data Bus D0–D7

Internally the 68902 contains 8-bit control registers. Data is written to and read from these registers over the 8-bit data bus. In addition to the normal read and write operations, the 8-bit data bus is also used to transfer an interrupt vector number to the processor. This mode is used on processors that support the vectored interrupt mode of operation.

Chip Select (SEL*)

The SEL* is used to chip select the 68902. The SEL* is asserted by a microprocessor when a read or write to the internal registers is required. To ensure correct operation and prevent contention on the data bus the SEL* and INT_ACK* signals should not be asserted simultaneously.

Register Select (RS1–RS5)

The register select lines are used to access the internal registers on the 68902. A total of 27 registers are implemented on the device as can be seen in Figure 3. The first 24 registers have the same function, and are software compatible to the MC68901. Two registers are used for handshaking control on the general purpose I/O port. These are the HSK0CR and HSK1CR registers. The operation of these registers is described in section 4. **Unless otherwise mentioned all bits in the 68902 registers are set to 0 at reset.**

Unlike the other registers the RESET register does not return or latch data on the data bus. Writing to this register will cause a software reset. A software reset function is useful in fault diagnostics to allow the 68902 to be individually reset separately from other peripheral devices in the system if a suspected fault occurs. The information presented on the data bus is irrelevant. The register is accessed in the same manner as a normal write cycle. If this write cycle is at least four clocks in duration a software reset will occur for the 68902. Reading the register has no effect, the bus cycle is terminated with spurious data on the data bus in the same manner as a normal read cycle. The software reset will reset all the internal modules except the bus interface module. This being previously hard-wired selected on initial power-up.

Figure 3. 68902 Register Map

Address						Abbreviation	Register Name
Hex	Binary RS5	RS4	RS3	RS2	RS1		
01	0	0	0	0	0	GPDR	General Purpose I/O Data 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	Timer 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
31	1	1	0	0	0	HSK0CR	Hand Shake 0 Control Register
33	1	1	0	0	1	HSK1CR	Hand Shake 1 Control Register
39	1	1	1	0	0	RESET	Software Reset Register

Write Cycle (W*)

This signal is used to control the direction of transfers on the data bus. When the W* is high and the microprocessor is chip selected then the contents of the internal registers are placed on the data bus. When the W* is asserted the information on the data bus is latched into the 68902's registers. **For correct operation of a write cycle, the W* signal must be asserted before or at the same time as SEL* is asserted.**

Reset (RESET*)

The reset input is used to place the 68902 in a predetermined state after the power is applied. Assertion of Reset ensures that all control and status registers are placed in a known default mode of operation.

Latch Data (DATA_LATCH)

During a read and write cycle by the microprocessor the DATA_LATCH signal is used to indicate that the processor has placed a valid address on the bus, and therefore data can be written or read by the 68902. During a read or an interrupt acknowledge cycle the DATA_LATCH signal is used to indicate that the microprocessor has latched data and therefore the 68902 can tri-state the data bus and give control back to the microprocessor.

Transfer Cycle complete (ACK*)

The ACK* signal is used to terminate a normal bus cycle.

Interrupt Request (IRQ*)

This open drain output signals to the processor indicates that an interrupt is pending from the 68902. There are a total of 16 interrupt channels that can generate an interrupt request. Clearing the Interrupt Pending Register (IPRA and IPRB) or clearing the Interrupt Mask Register (IMRA and IMRB) will cause the IRQ* pin to be negated. IRQ* will also be negated as a result of an interrupt acknowledge cycle, INT_ACK* asserted, being activated.

Interrupt Enable In (IEI*)

This input together with the IEO* provides a daisy-chained interrupt structure for a vectored interrupt scheme. IEI* indicates that no higher priority device is requesting interrupt service. The highest priority 68902 in the chain should have its IEI* pin tied low. During an interrupt acknowledge cycle, a 68902 with a pending interrupt is allowed to pass a vector number to the processor until its IEI* pin is asserted. When the

daisy-chain option is not implemented, all 68902s should have their IEI* pin tied low.

Interrupt Enable Out (IEO*)

This input together with the IEI* provides a daisy-chained interrupt structure for a vectored interrupt scheme. The IEO* of a particular 68902 signals lower priority devices that neither it nor any other higher priority device is requesting interrupt service. The lowest priority 68902's IEO* is not connected. When the daisy-chain option is not implemented, IEO* is not connected. To ensure proper recognition of an interrupt request by the microprocessor the IEO* signal is asserted one clock period on the rising edge of the clock before the IRQ* to the microprocessor. This prevents other daisy-chained devices gaining access to the bus during an interrupt acknowledge cycle.

Interrupt Acknowledge (INT_ACK*)

If both IRQ* and IEI* are asserted, the 68902 will begin an interrupt acknowledge when INT_ACK* is asserted. The 68902 will supply a unique vector number to the processor which corresponds to the particular channel requesting interrupt service. In a daisy-chained interrupt structure, all devices in the chain must have a common INT_ACK*.

Microprocessor Unit select (MPU)

This pin is used to perform a hard-wired select of the microprocessor connected to the 68902 during reset. The pin selects between two different bus structures. The following encoding shows the defined value on the pin and the microprocessor that interfaces directly to the 68902.

MPU

GND ------- MC68000, MC68008, MC68010, MC68001, MC68020 and MC68030 (default case).

V_{CC} ------- MC68040.

Timer Clock (XTAL1/CLK2 and XTAL2/nc)

This input provides the timing signals for the four timers. A crystal can be connected between the timer clock inputs, XTAL1/CLK2 and XTAL2/nc, or XTAL1/CLK2 can be driven with a TTL-level clock while XTAL2/nc remains unconnected.

Timer Outputs (TAO, TBO, TCO and TDO)

Each timer has an associated output which toggles when its main counter counts through 01 (hexadecimal) regardless of which operational mode is selected. When in the delay mode, the timer output will be a square wave with a period equal to two timer cycles. This output may be used to supply the Universal Synchronous/Asynchronous Receiver-Transmitter (USART) baud rate clocks.

Timer Inputs (TAI and TBI)

These inputs are control signals for the timers A and B in the pulse width measurement mode and the event count mode. These signals generate interrupts at the same priority level as the general purpose I/O interrupts lines P4 and P3, respectively, when in the pulse width measurement mode. While P4 and P3 do not have interrupt capability when the timers are operating in this mode, they may still be used for I/O.

Serial Input (SI)

This input line is the USART receiver data input. This input is not used in the USART loopback mode.

Serial Output (SO)

This output line is the USART transmitter data output. This output is in a high-impedance state after a device reset.

Receiver Clock (RC)

This input controls the serial bit rate of the receiver. The signal may be supplied by the timer output lines or by an external TTL-level clock which meets the following trimming characteristics: max. receiver clock 5 MHz (4 * Receiver Clock = < 20 MHz). This clock is not used in the USART loopback mode.

Transmitter Clock (TC)

This input controls the serial bit rate of the transmitter. The signal may be supplied by the timer output lines or by an external TTL-level clock which meets the following trimming characteristics: max. transmitter clock 5 MHz (4 * Transmitter Clock = < 20 MHz).

Receiver Ready (RR*)

This active low output reflects the receiver buffer full status (bit 7 in the Receiver Status Register) for DMA operations.

Transmitter Ready (TR*)

This active low output reflects the transmitter buffer empty (bit 7 in the Transmitter Status Register) for DMA operations.

General I/O or Interrupt lines (P0–P7)

These 8 lines can be used for three functions:

i) As general purpose input output lines. The direction being controlled by the DDR.

ii) As interrupt lines. Each line can be individually programmed to generate an interrupt on a low to high, or high to low transition.

iii) As a parallel transfer port of 8-bits wide. The latching of data in this parallel port being controlled by the handshaking control lines.

Handshaking Control Lines (HSK0 and HSK1)

These signals are used to control the direction and latching of data in the 8-bit I/O port of the 68902.

Scan Test select (SCAN_EN)

Assertion of the SCAN_EN enables a scan tests to be performed on the internal 68902 logic. The results of the scan tests are produced on external pins. In test mode these pins have a different function than in normal operating mode and are only used for testing the 68902 internal logic. This signal is active high.

Inverse Transfer Cycle Complete (ACK)

The ACK signal is the inverse of ACK*, the cycle termination signal from the 68902. The ACK can be used for example to supply the DSACK1* termination signal in a MC68020 or MC68030 system to indicate communication over a 8-bit wide bus.

Test pins (TEST1, TEST2)

These active high pins are used for internal test purposes only. When not in use they can be tied to V_{SS} or left uncon-

nected, this being the default case internally with internal pull-down resistors.

BUS OPERATION

The operation of the microprocessor interface bus is designed to interface synchronously to a number of Motorola microprocessors.

M68000 Bus etc.

The bus structure on the MC68000, MC68HC000, MC68HC001 and MC68010 processors have a four clock access to memory. Therefore it is possible to operate at zero wait state to the 68902 by synchronizing to the microprocessor bus. To allow the 68902 to operate with the MC68020 and MC68030, in addition to the MC68000, the cycle times are extended as shown in the following diagrams. For the MC68020 and MC68030 running at up to 20 MHz a simple interface can be designed. For higher clock frequencies the system clock should be divided down to supply the clock for the 68902 to allow it to operate with faster processors.

M68000 to MC68030 Read Cycle

The MC68000 through to the MC68030 bus timing is shown in Figure 4 and Figure 5. Assuming the SEL* is generated from address decode logic it should be stable before the rising edge of the clock. The addresses A1–A5 are assumed to select the registers on the chip.

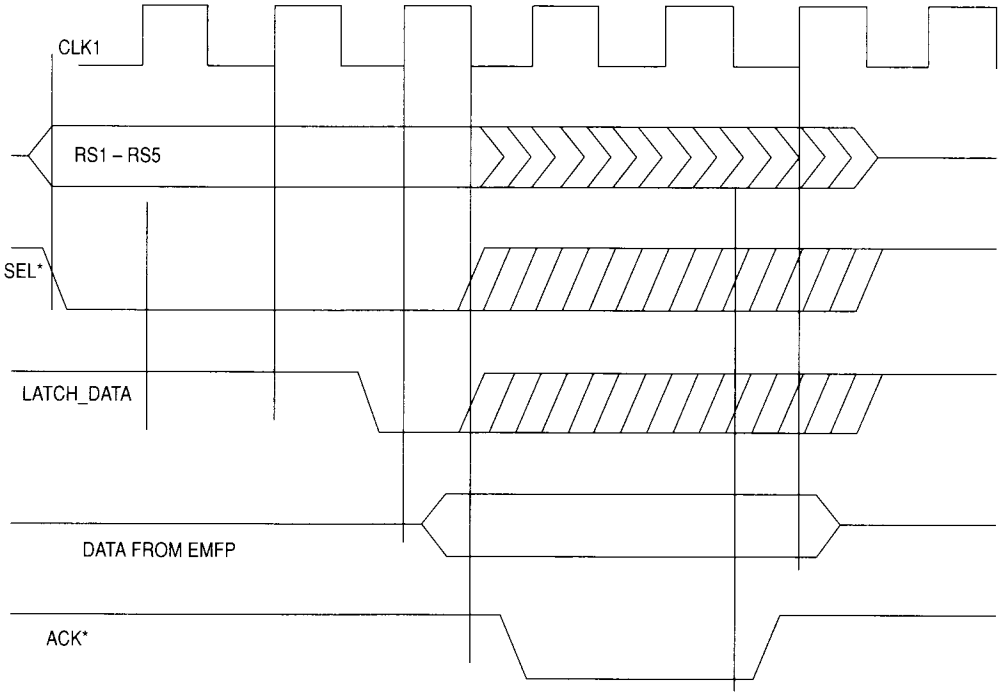


Figure 4. MC68000 through to MC68030 Read Cycle

The addresses are not verified internally until the DATA_LATCH is asserted. This should come at the earliest some time after the next rising edge of the clock. Delaying the assertion of this signal results in additional wait states being added to the access time. Therefore when SEL* is asserted and DATA_LATCH is asserted a register access is valid.

After the cycle has started and a register access is valid, the next rising edge of the clock results in the 68902 driving the data from the registers on the data bus. If the 68902 can not supply date on this edge it will drive data on the next rising edge that it can, thereby asserting a wait state. The next falling edge after the 68902 drives data results in the assertion of the ACK* and ACK signals from a tri-state to active state. Two clock periods after the ACK and ACK* are asserted they are again negated. The 68902 will then tri-state the address bus on the next rising edge of the clock.

NOTE

The cycle of the MC68000 to MC68030 microprocessors is optimized for synchronous operation. This offers the maximum performance for the device. The data from the 68902 however remains on the DATA bus for a period of two clocks. This allows it to transfer through any buffer circuits that may lie between the processor and the 68902.

MC68000 through to MC68030 Write Cycle

The write cycle is very similar to that of the read. A write by the microprocessor is indicated with the assertion of the W* signal as can be seen in Figure 5. This is used to select the data direction in the 68902 and in addition to know where in the cycle to latch data. If the microprocessor cannot supply data in the time period shown in Figure 5. Wait states can be inserted in the cycle by delaying the assertion of the

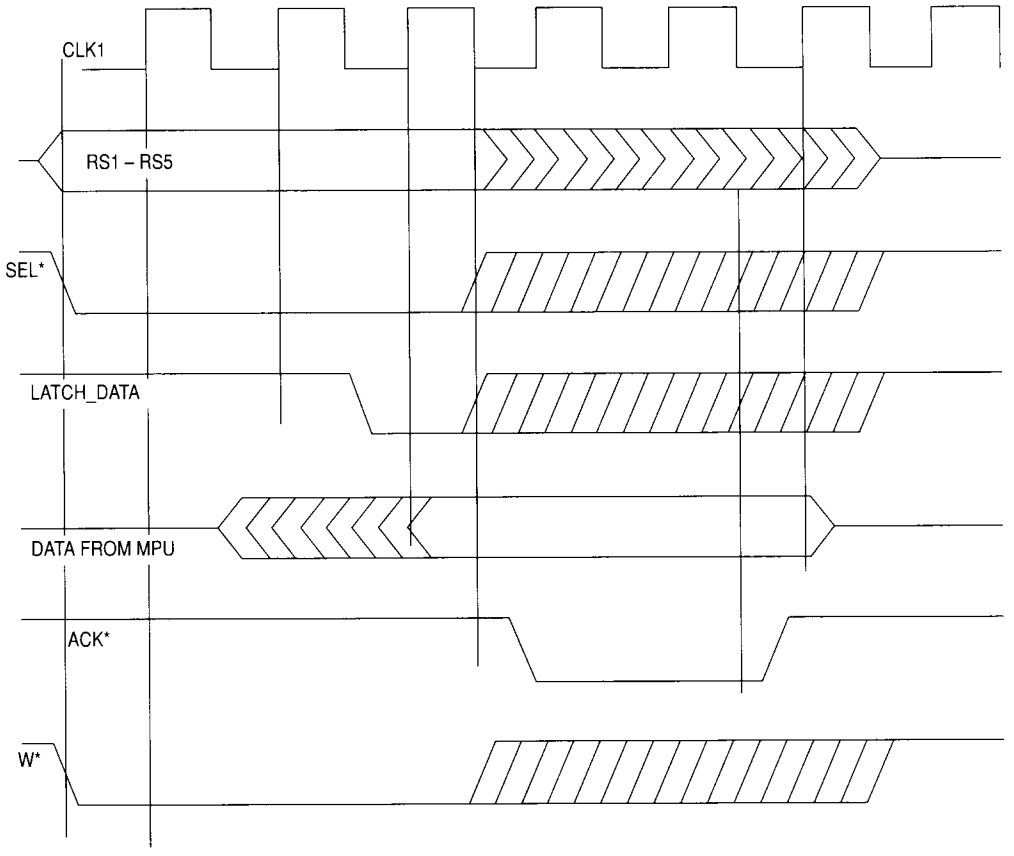


Figure 5. MC68000 to MC68030 Write Cycle

DATA_LATCH signal to another rising edge of the clock when there is valid data on the data bus.

MC68000 to MC68030 Interrupt Acknowledge Cycle

The interrupt acknowledge cycle involves placing an interrupt number on the data bus. The cycle is treated as a normal read cycle with the exception that the INT_ACK* is asserted instead of the SEL*. In addition the IEI* must be asserted if the 68902 is to respond to the interrupt. The diagram of the interrupt acknowledge cycle is shown in Figure 6.

MC68040 Bus Interface

The MC68040 possesses a bus structure that better matches the requirement of the 68902 internal operation. The processor bus is a synchronous bus with signals being asserted on the rising edges of the clock. Although a much simpler bus structure can be achieved in the 68902 with the MC68040 microprocessor, additional select and DATA_LATCH logic may be required externally. External logic would for example be used to extend the duration of the MC68040's transmission start signal, TA*. This is only asserted at the start of each cycle for the duration of one clock period. However this should still be possible to design in only one external PAL device. Initial versions of the 68902 will be operational up to 20 MHz, the initial MC68040 devices will be sampled at 25 MHz. However even if the microprocessor clock is divided by 2 to drive the 68902 at 12.5 MHz. This still offers a considerable system improvement over an equivalent 68901 device.

MC68040 Read Cycle

The MC68040 read cycle is initiated when the SEL* is asserted before the rising edge of the clock to the 68902, CLK1. The exact set up and hold times are given in the Electrical Characteristics tables. After the SEL* is asserted the DATA_LATCH* should be asserted on the next or later rising edge(s) of the clock to validate the addresses. Delaying the assertion of the DATA_LATCH* results in wait states being added in the cycle.

After the DATA_LATCH* is asserted the 68902 will drive the data on the data bus on the next rising edge of the clock. If there are other internal operations in progress this may be delayed by a number of clocks. On the next rising edge after the data bus is driven the 68902 will assert the ACK* signal to terminate the cycle. On the next rising edge the ACK* will be negated and the data bus tri-stated, to ensure enough hold time is present to satisfy the timing requirements of the MC68040.

NOTE

The cycle of the MC68040 is optimized for synchronous operation to the microprocessor. This offers the maximum performance for the device. The data from the 68902 however remains on the DATA bus for a period of two clocks. This allows it to transfer through any buffer circuits that may be between the processor and the 68902 in the application.

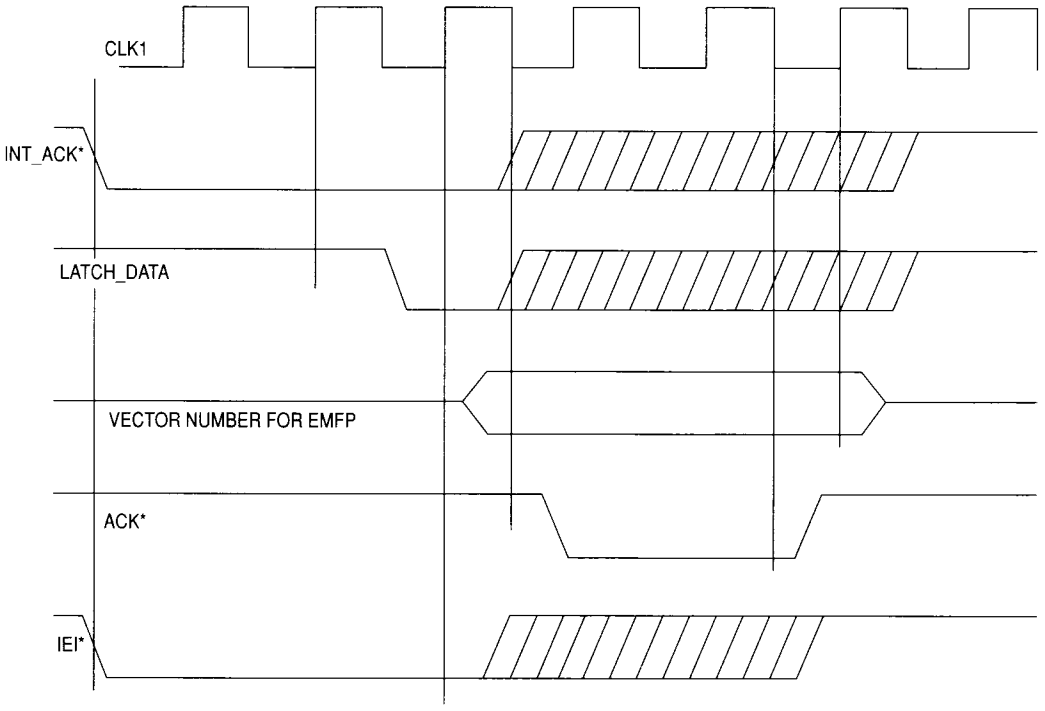


Figure 6. MC68020 or MC68030 Interrupt Acknowledge Cycle

MC68040 Write Cycle

The write cycle is very similar to that of the read. The additional W* input is used to select the direction of the data buffers and qualify the access.

MC68040 Interrupt Acknowledge Cycle

The interrupt acknowledge cycle is the same as that of the read cycle. Instead of data being placed on the data bus, a vector number is placed on the bus. The interrupt acknowledge cycle is initiated when INT_ACK* and IEI* asserted simultaneously. This is shown in Figure 9.

ACK* and ACK Considerations

The ACK and ACK* signals produce the bus cycle acknowledge to the processor. Similar bus termination signals are typically driven by a number of devices in the system. Therefore to speed up the system access time, these acknowledges can be wired-ORed together.

To allow this to be achieved the 68902 must negate the ACK and ACK* signals at the end of the bus cycle before tri-stating. This will ensure that the lines are charged to a logic level high. Although not shown in the previous figures, the operation of the ACK* and ACK signals are shown in Figure 10.

The ACK and ACK* signals are used to signal the end of a bus cycle. If the 68902 cannot respond to a bus cycle in the minimum number of clock the cycle will be automatically extended by one or more clocks by delaying the assertion of ACK and ACK*.

INTERRUPT STRUCTURE

The interrupt structure on the 68902 is the same as that on the MC68901. The 68902 has the added advantage of performing parallel handshaking over the I/O port. This handshaking scheme requires the generation of an additional interrupt vector to the processor.

INTERRUPT PROCESSING

Each 68902 provides individual interrupt capability for its various functions. When an interrupt is received on one of the external channels or from one of the eight internal sources, the 68902 will request interrupt service. Sixteen interrupt channels are assigned a fixed priority so that multiple pending interrupts are serviced according to their relative importance. Since the 68902 can internally generate vector numbers for these interrupts, the unique vector number which corresponds to the highest priority channel that has 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 interrupting source, decreasing interrupt latency. In addition to the 16 interrupt channels that are assigned fixed priority one additional interrupt is generated separately with the I/O handshaking scheme with no fixed priority. When multiple interrupts occur this interrupt takes the lowest priority.

Interrupt Channel Prioritization

The 16 interrupt channels are prioritized from highest to lowest, with General Purpose Interrupt 7 (P7) being the highest and P0 the lowest. The priority of the interrupt is determined by the least significant four bits in the interrupt vector number which are internally generated by the 68902. Pending interrupts are presented to the processor in order of priority unless they have been masked. By selectively masking interrupts, the channels are in effect re-prioritized.

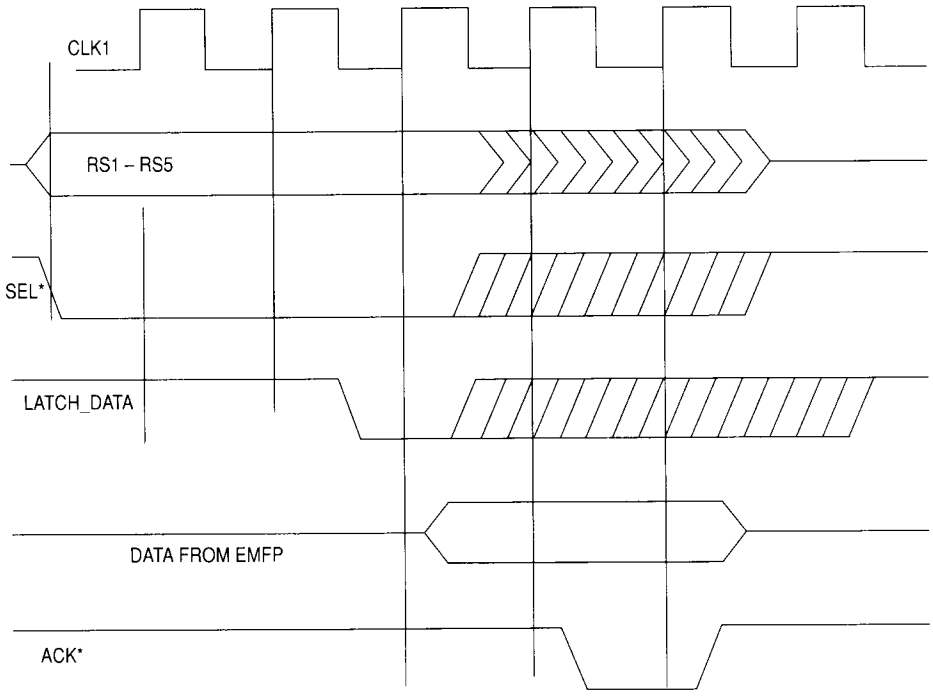


Figure 7. MC68040 Read Access

Interrupt Vector Number

During an interrupt acknowledge cycle, a unique 8-bit interrupt vector number is presented to the system which corresponds to the specific interrupt source that is requesting service.

7	6	5	4	3	2	1	0
V7	V6	V5	V4	IV3	IV2	IV1	IV0

V7–V4 — Written by user to set the most-significant four bits of the interrupt vector number.

IV3–IV0 — Determine highest priority channel Requiring Interrupt

IV3	IV2	IV1	IV0	Description
1	1	1	1	General Purpose Interrupt 7 (P7)
1	1	1	0	General Purpose Interrupt 6 (P6)
1	1	0	1	Timer A
1	1	0	0	Receiver Error
1	0	1	1	Receiver Empty
1	0	1	0	Transmit Buffer Empty
1	0	0	1	Transmit Error
1	0	0	0	Timer B
0	1	1	1	General Purpose Interrupt 5 (P5)
0	1	1	0	General Purpose Interrupt 4 (P4)
0	1	0	1	Timer C
0	1	0	0	Timer D
0	0	1	1	General Purpose Interrupt 3 (P3)
0	0	1	0	General Purpose Interrupt 2 (P2)
0	0	0	1	General Purpose Interrupt 1 (P1)
0	0	0	0	General Purpose Interrupt 0 (P0)

Figure 11. Interrupt Channels

When an interrupt from the handshaking mechanism on the I/O port requires servicing. The contents of bits 4 through to 7 of the register HSK1CR are used to generate the upper four bits of the vector number. The lower four bits being 0001 or 0000, depending upon which of the two handshaking interrupts are being serviced. The value 0001 corresponds to handshaking line HSK1 and 0000 corresponds to handshaking line HSK0.

Vector Register (VR)

This 8-bit register determines the four most-significant bits in the interrupt vector format and which end-of-interrupt mode is used in a vectored interrupt scheme. The vector register should be written to before writing to the interrupt mask or enable register to ensure that the 68902 responds to an interrupt acknowledge cycle with a vector number that is in the range of allowable user vectors.

7	6	5	4	3	2	1	0
V7	V6	V5	V4	S	*	*	*

V7–V4 — Written by user to set the most-significant four bits of the interrupt vector number.

S — In-Service Register Enable
1 = Software end-of-interrupt mode and in-service register bits enabled.
0 = Automatic end-of-interrupt mode and in-service register bits forced low.

2–0 — Not Used

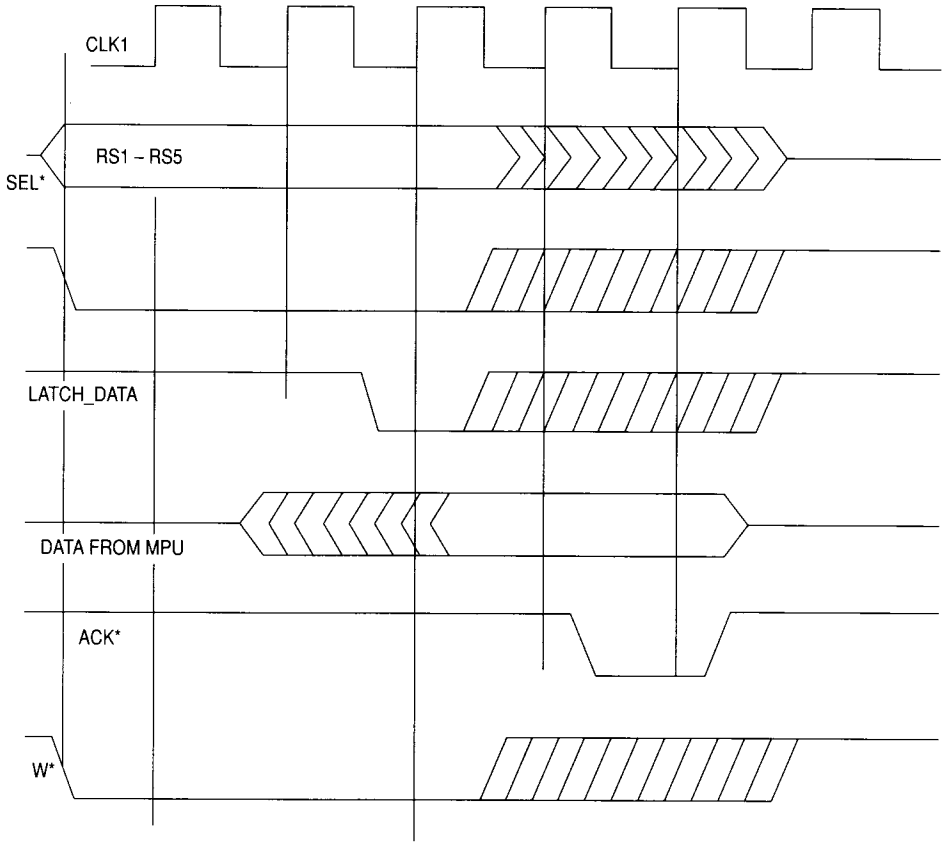


Figure 8. MC68040 Write Access

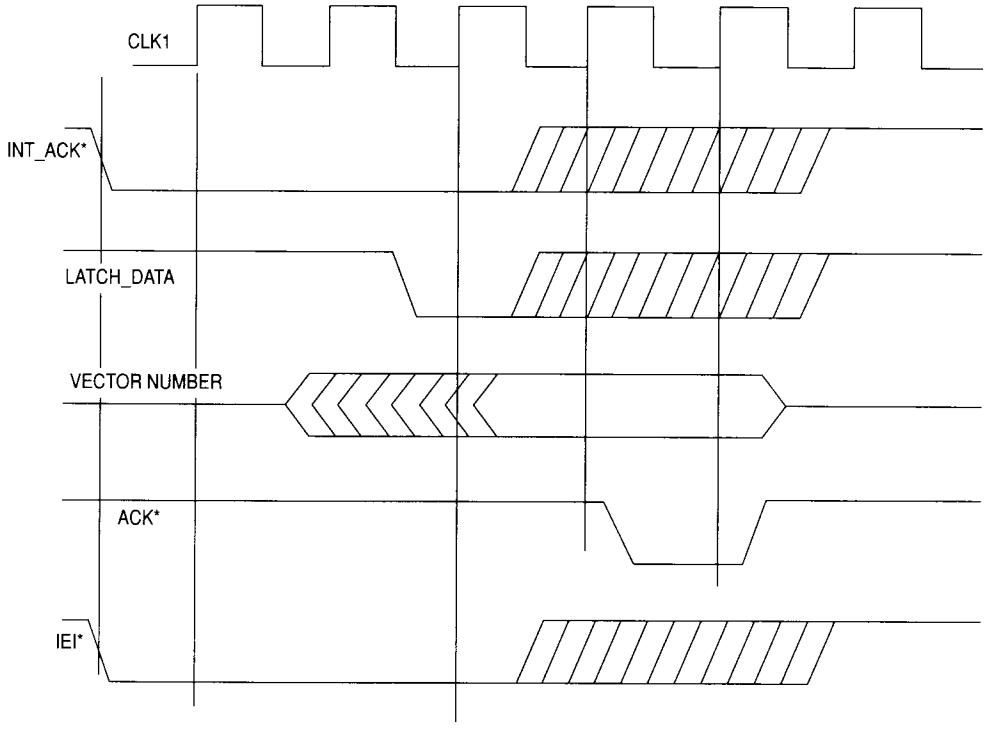


Figure 9. MC68040 Interrupt Acknowledge Cycle

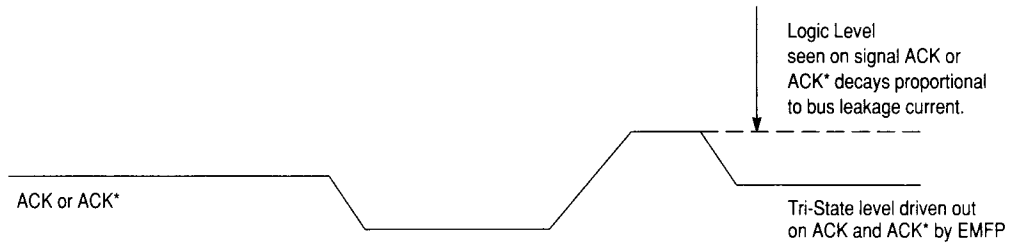


Figure 10. Operation of the ACK and ACK* Signals

DAISY-CHAINING 68902s

As an interrupt controller, the 68902 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, additional sources may be added to the prioritized structure by daisy-chaining 68902s. Interrupt sources are prioritized internally within each 68902, and the 68902s are prioritized by their position in the chain. Unique vector numbers are provided for each interrupt source.

The IEI* and IEO* signals implement the daisy-chained structure. The IEI* of the highest priority 68902 is tied low and the IEO* output of the device is tied to the next highest priority 68902's IEI*. The IEI* and IEO* signals are daisy-chained in this manner for all the 68902s in the chain with the lowest priority 68902s IEO* left unconnected.

Daisy-chaining requires that all parts in the chain have a common INT_ACK*. When the common INT_ACK* is asserted during an interrupt acknowledge cycle, all parts will prioritize interrupts in parallel. When the IEI* signal to an 68902 is asserted, the part may respond to the INT_ACK* cycle if it requires interrupt service.Otherwise, the part will assert IEO* to the next lower priority device. Thus, priority is passed down the chain via IEI* and IEO* until a part which has a pending interrupt is reached. The part with the pending interrupt passes a vector number to the processor and does not propagate IEO*.

If the IEI* to the 68902 is not asserted, then under no condition will an IRQ* be issued to the microprocessor. This improves the daisy-chaining scheme and acts as a hardware interrupt enable signal. In addition any pending or in-service interrupts, not masked in the internal mask register, will cause the IEO* to remain negated until all pending interrupts are serviced.

INTERRUPT CONTROL REGISTER

Interrupt processing is managed by the 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 in **Nesting 68902 Interrupts**. The interrupt control registers are shown in the following paragraphs.

Interrupt Enable Registers (IREA, IERB)

The interrupt channels are individually enabled or disabled by writing a one or zero, respectively, to the appropriate bit of Interrupt Enable Register A or B (IERA or 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 68902, 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 68902.

Writing a zero to a bit of interrupt enable register A or B will cause the corresponding bit of the interrupt pending register 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 register A or B. So if the 68902 is in software end-of-interrupt mode (see Software End-Of-Interrupt) and an interrupt is in service when a channel is disabled, the in-service bit of that channel will remain set until cleared by software.

7	6	5	4	3	2	1	0
GPIB7	GPIP6	Timer A	RCV Buffer Full	RCV Error	XMIT Buffer Empty	XMIT Error	Timer B

IERA

7	6	5	4	3	2	1	0
GPIB5	GPIP4	Timer C	Timer D	GPIP3	GPIP2	GPIP1	GPIP0

IERB

Interrupt Pending Register (IPRA, IPRB)

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 68902 responds with a vector number. In a polled interrupt system, the interrupt pending register must be read to determine the interrupting channel, and then the interrupt pending bit is cleared by the interrupt handling routine. No interrupt acknowledge cycle is used in this mode.

7	6	5	4	3	2	1	0
GPIB7	GPIP6	Timer A	RCV Buffer Full	RCV Error	XMIT Buffer Empty	XMIT Error	Timer B

IPRA

7	6	5	4	3	2	1	0
GPIB5	GPIP4	Timer C	Timer D	GPIP3	GPIP2	GPIP1	GPIP0

IPRB

Interrupt Mask Register (IMRA, IMRB)

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 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 IMBRB 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.

7	6	5	4	3	2	1	0
GPIB7	GPIP6	Timer A	RCV Buffer Full	RCV Error	XMIT Buffer Empty	XMIT Error	Timer B

IMRA

7	6	5	4	3	2	1	0
GPIB5	GPIP4	Timer C	Timer D	GPIP3	GPIP2	GPIP1	GPIP0

IMRB

Interrupt In-Service Register (ISRA, ISRB)

These registers indicate whether interrupt processing is in progress for a certain channel. A bit is set whenever an interrupt vector number is passed for an interrupt channel during an interrupt acknowledge cycle and the S bit of the vector register is a one. The bit is cleared whenever interrupt service is complete for an associated interrupt channel, the S bit of the vector register is cleared, or the processor writes a zero to the bit.

7	6	5	4	3	2	1	0
GPIB7	GPIP6	Timer A	RCV Buffer Full	RCV Error	XMIT Buffer Empty	XMIT Error	Timer B

ISRA

7	6	5	4	3	2	1	0
GPIB5	GPIP4	Timer C	Timer D	GPIP3	GPIP2	GPIP1	GPIP0

ISRB

NESTED 68902 INTERRUPTS

In the M68000 family vectored interrupt system, the 68902 is assigned to one of seven possible interrupt levels. When an interrupt is received from the 68902, an interrupt acknowledge for that level is initiated. Once an interrupt is recognized at a particular level, interrupts at the same level or below are masked by the processor. As long as the processor's interrupt mask is unchanged, the M68000 interrupt structure will prohibit nesting the interrupts at the same interrupt level. However, additional interrupt requests from the 68902 can be recognized before a previous channel's interrupt service routine is finished by lowering the processor's interrupt mask to the next lower interrupt level within the interrupt handler.

When nesting 68902 interrupts, it may be desirable to permit interrupts on any 68902 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 allow subsequent higher priority channel interrupt requests to supersede previously recognized lower priority interrupt requests. The 68902 interrupt structure provides the flexibility of offering two end-of-interrupt options for the vectored interrupt scheme. 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 68902 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. When the S bit is programmed to a one, the 68902 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 Mode

When an interrupt vector 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 68902. The in-service bits of the interrupt in-service registers (ISRA and ISRB) are forced low. Subsequent interrupts, which are received on any 68902 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 Mode

In the software end-of-interrupt mode, the channel's associated interrupt pending bit is cleared. 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 the interrupt acknowledge 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 with its in-service bit 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 zero to its corresponding bit in ISRA or ISRB and ones to all other bit positions. Since bits in the in-service register 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.

HANDSHAKING I/O PORT INTERRUPT MECHANISM

The handshaking port interrupt mechanism is similar to the scheme adopted for the general purpose parallel ports such as that found on many single chip micro-controller devices. In general the interrupt line to the processor for this scheme is cleared by writing a 0 to bits 6 or 7 of the HSKCR0 register during an interrupt service routine in the polled interrupt mechanism. However, if the processor has the capability of generating an interrupt acknowledge cycle then when the

INT_ACK* is asserted the interrupt line to the processor is automatically negated according to the priority scheme and the selected end of interrupt mode (see bit 3 of the HSKCR1 register).

GENERAL PURPOSE INPUT/OUTPUT PORT

The general purpose input/output port operates in three configurations to offer a high degree of functionality:

In the normal GPIP mode the port operates as a simple 8-bit input/output port. The operation of this port is software compatible to the MC68901.

In hardware handshaking mode the 8-bit port operates together with two handshaking control lines are used to offer parallel port control. In this mode the functions of the I/O port are similar to those found on many 8-bit single chip microcontrollers. The handshaking on the I/O port has the capability to generate two interrupt vectors during an interrupt acknowledge cycle. The four upper bits of the interrupt vector number are made up of four bits in the HSKCR1 register, D7–D4. The remaining bits in D3–D0 are either 0000 or 0001 depending on if the HSK0 or HSK1 pins respectively caused the interrupt.

A third mode exist to allow the USART to adhere to the hardware CCITT V.24 specification for serial data transmission. This protocol is ideally suited to such systems using RS-232-C and RS-422 etc.

AUTO-VECTORED INTERRUPT CONTROLLER

The 68902 interrupt controller is particularly useful in a system which has many interrupt source that are non-vectored. These interrupts are referred to as Auto-Vectored interrupts. Auto-vectors have no capability of supplying a vector number to the microprocessor. On receiving an Auto-vectored interrupt the Auto-vectored interrupt handler must then poll all the Auto-vectored sources at the interrupt requesting priority level to determine which device is requesting service. However, by connecting the IRQ* output from one of these interrupt sources to the general purpose I/O port (GPIP) of the 68902, a unique vector number will be provided to the processor during an interrupt acknowledge cycle. This interrupt structure will significantly reduce interrupt latency for the requesting devices that do not support any vectored interrupt capability.

GPIP CONTROL REGISTERS

The GPIP is programmed via three control registers. 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.

General Purpose I/O Data Register (GPDR)

The general purpose I/O data register is used to input data from or output data to the port. When data is written to the GPDR, 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 GPDR 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 buffer.

7	6	5	4	3	2	1	0
GPIP7	GPIP6	GPIP5	GPIP4	GPIP3	GPIP2	GPIP1	GPIP0

GPDR

In the handshaking mode of operation the data in this register is latched on a read or write operation. The latching of data is dependent on the hardware handshaking mode.

Active Edge Register (AER)

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 to the appropriate edge bit of the active edge register will cause 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 line. When the processor sets a bit, interrupts will be generated on the rising edge of the associated input signal. When the processor clears a bit, interrupts will be generated on the falling edge of the associated input signals.

7	6	5	4	3	2	1	0
GPIP7	GPIP6	GPIP5	GPIP4	GPIP3	GPIP2	GPIP1	GPIP0

AER

NOTE

The AER should be configured before enabling interrupts via the Interrupt Enable Register (IERA and IERB). Also, changing the edge bit while interrupts are enabled may cause an interrupt on the corresponding channel.

Data Direction Register (DDR)

The Data Direction Register (DDR) allows the programmer to define P0 through P7 as inputs or outputs by writing to the corresponding bit. When a bit of the data direction register is written as a zero, the corresponding interrupt I/O pin will be a high-impedance input.

7	6	5	4	3	2	1	0
GPIP7	GPIP6	GPIP5	GPIP4	GPIP3	GPIP2	GPIP1	GPIP0

DDR

I/O HARDWARE HANDSHAKING MODE

Selecting the I/O hardware handshaking mode on the 68902 allows the 8-bit I/O port to be configured for handshaking operations. This configuration can be used to latch data in a simplex and half duplex manner. In this mode the I/O lines cannot be used to generate interrupts.

Handshake Control Register 0 (HSKCR0)

This control register is used to control the operation of the handshaking mechanism when in this mode of operation. A diagram of the control register is given in the following pages. Pulse widths on the HSK0 and HSK1 lines must have a duration of two or more internal clock periods to be recognized by the 68902. The function of these bits are listed as follows:

Bit 0

When this bit is a 1 an interrupt will be generated on an active transition of HSK0. When this bit is a 0 the interrupt is disabled.

Bit 1

This bit is used to determine which active edge of HSK0 causes the data to be latched and the interrupt flag 1 for HSK0 to be set. When this bit is a 0 a high-to-low transaction on HSK0 causes the flag to be set. When this bit is a 1 a low-to-high transaction on HSK0 causes this flag to be set.

Bit 2

When this bit is a 0 the port handshaking line HSK1 is configured for a read operations only. When this bit is a 1 the port handshaking line HSK1 is configured as a write port only. This bit is used to determine if HSK1 will be asserted during a read operation or write operation. As an example if this bit is set for a read and a write occurs to the port then the HSK1 line is not activated. If a read occurs then the HSK1 line is activated depending upon the operating mode given below. This bit is a 0 at reset.

Bits 3, 4, 5

These bits are used to configure the operation of the handshaking line HSK1. The encoding of these bit means the following:

Bit 5	Bit 4	Bit 3	
0	0	0	Interrupt disabled for HSK1
0	0	1	Generate Interrupt on high-to-low transition of HSK1
0	1	0	Interrupt Disabled for HSK1
0	1	1	Generate Interrupt on low-to-high transition of HSK1
1	0	0	HSK1 goes low on first rising edge of the clock after the GPDR register is read or written to by the MPU (dependent on bit 2). It is returned high on the first rising edge of the clock in which HSK0 is asserted.
1	0	1	HSK1 goes low on first low-to-high of the clock following completion of a MPU read or write (dependent on bit 2) of the GPDR. It is returned high on the next rising edge of the clock. This generates a pulse on the HSK1 line of one internal clock duration.
1	1	0	HSK1 goes low as MPU writes Bits 5, 4, 3 = 110 into the register.
1	1	1	HSK1 goes high as MPU writes Bits 5, 4, 3 = 111 into the register.

Bit 6

Interrupt flag for HSK1. This bit indicates that an interrupt is pending for the GPIP HSK1. The bit is cleared automatically when INT_ACK* is asserted and the vector number of this interrupt has been given to the processor. In the polled interrupt mode of operation this bit is cleared by writing a 0 to this bit.

Bit 7

Interrupt flag for HSK0. This bit indicates that an interrupt is pending for the GPIP HSK0. The bit is cleared automatically when INT_ACK* is asserted and the vector number of this interrupt has been given to the processor. In the polled interrupt mode of operation this bit is cleared by writing a 0 to this bit.

NOTE

Like the other interrupts on in the 68902 the interrupts for the GPIP can operate in either polled or vectored mode.

(i) In vector mode a unique vector number is returned for either HSK0 or HSK1. In the case of a simultaneous interrupt request form HSK0 and HSK1, HSK0 is always the highest priority and is serviced first.

(ii) In polled interrupt mode the user has control over which channel has priority. Therefore the user can assign his own update mechanism to the two channels. The interrupt status flags are cleared in the interrupt service routine by writing a 0 to bit 6 or bit 7. It is not possible to write a 1 to either bit 6 or bit 7. If this is performed the bit remains unchanged.

INT. FLAG FOR HSK0	INT. FLAG FOR HSK1	HSK1 CONTROL	READ/ WRITE	HSK0 CONTROL

HSKCR0 CONTROL REGISTER

CCITT V.24 HARDWARE HANDSHAKING MODE

Selecting this mode of operation under software via the HSKCR1 control register reorganizes the I/O port to conform with the serial hardware protocol of the CCITT V.24 recommendation. When this mode is selected the hardware handshaking lines will automatically control the serial flow of information. In this mode of operation the I/O port is reconfigured as follows:

P0 = RTS* — Request to send information.

P1 = CTS* — Clear to send information.

P2 = DSR — Data Set Ready.

P3, P4 = Used as normal I/O as in the normal operation of the GPIP, or used as inputs to the timers in pulse width measurement mode.

P5 = DCD* — Data Carrier Detect.

P6 = DTR* — Data Terminal Ready.

P7 = Used as normal I/O as in the normal operation of operation of the GPIP.

Interface of the Port Lines to the USART

The above hardware handshaking lines are controlled automatically by signals generated from the logic in the USART when this mode of operation is selected. The generation of these signals is as listed below:

RTS*:

The request to send is asserted when the transmitter register is loaded with data. It is negated at reset and when the transmitter register is empty and there is no new data being transferred to this shift register.

CTS*:
When this mode of operation for the port is selected the transmitter shift register will not transmit data unless the CTS* is asserted. This is used as an acknowledgment to the RTS* signal by the receiving slave device.

DSR:
When the CCITT V.24 mode of operation is selected this input is used as an external hardware select for the USART. In the V.24 mode of operation if this input is not selected the USART will be disable, this being the default case after reset.

DCD*:
This input is used as a hardware select for the receiver shift register. When asserted serial data will be shifted in to the receiver register. When negated data appearing on the receiver pin will not be shifted in to the register. After reset the receiver register is enabled if DCD* is asserted and disabled if DCD* is negated.

DTR*:
This is asserted by the 68902 after reset and remains asserted until the receiver shift register is full. When the contents of the receive shift register are transferred to the receiver register (indicating that the shift register is now empty) the DTR* is again asserted.

HANDSHAKE CONTROL REGISTER 1 (HSKCR1)

This control register is used to supply the upper 4-bit of the vector number in vector interrupt mode if an interrupt source was generated from the I/O port. In addition two flags are used in this register for interrupts on the I/O port. The remaining two bits are used to select the operating mode of the 8-bit port. A description together with diagram of the bits in this register is given below:

Bit 0, 1
These bits are cleared at reset. They are used to select one of three modes of operation for the GPIP. The coding on these bits is as given below:

Bit 1	Bit 0	
0	0	GPIP mode
0	1	GPIP mode
1	0	CCITT V.24 mode
1	1	Handshaking mode

When Bits 0 and 1 are not programmed for the handshaking mode of operation the handshake lines are placed in the tri-state mode. The HSKCR0 control register can only be modified when the handshaking mode (bits 1 and 0 are both programmed with 1) is not selected. If this is not adhered to a spurious internal interrupt may be generated.

Bit 2
GPIP_in_Service bit. This has the same function as that in the interrupt controller, except it is only for the GPIP port.

TIMERS

The 68902 contains four 8-bit timers which provide many functions typically required in microprocessor systems. The timers can supply the baud rate clocks for the on-chip serial I/O channel, generate periodic interrupts, measure elapsed time, and count signal transitions. In addition, two timers have waveform generation capability.

All timers are prescaler/counter timers with a common independent clock input (XTAL1 and XTAL2) and are not required to be operated from the system clock (CLK1). Each timer's output signal toggles when the timer's main counter times out. Additionally, timers A and B have auxiliary control signals which are used in two of the operation modes. An interrupt channel is assigned to each timer, and when the auxiliary control signals are used in the pulse width measurement mode, a separate interrupt channel will respond to transitions on these inputs.

OPERATION MODES
Timer A and B are full function timers which, in addition to the delay mode, operate in the pulse width measurement mode and the event count mode. Timers C and D are delay timers only. A brief discussion of each of the timer modes follows.

Delay Mode Operation
All timers may operate in the delay mode. In this mode, the prescaler is always active. The prescaler specifies the number of timer clock cycles which must elapse before a count pulse is applied to the main counter. A count pulse causes the main counter to decrement by one. When the timer has decremented down to 01 (hexadecimal), the next pulse will cause the main counter to be reloaded from the timer data register and a time out pulse will be produced. The time out pulse is coupled to the timer's interrupt channel and, if the channel is enabled, an interrupt will occur. The time out pulse also causes the timer output pin to toggle. The output will remain in this new state until the next time out pulse occurs.

For example, if delay mode with a divide-by-10 prescaler is selected and the timer data register is loaded with 100 (decimal), the main counter will decrement once every 10 timer clock cycles. After 1000 timer clocks, a time out pulse will be produced. This time out pulse will generate an interrupt if the channel is enabled (IERA, IERB), and in addition, the timer's output line will toggle. The output line will complete one full period every 2000 cycles of the timer clock.

If the prescaler value is changed while the timer is enabled, the first time out pulse will occur at an indeterminate time no less than one nor more than 200 timer clock cycles. Subsequent time out pulses will then occur at the correct interval.

If the main counter is loaded with 01 (hexadecimal), a time out pulse will occur every time the prescaler presents a count pulse to the main counter. If the main counter is loaded with 00, a time out pulse will occur every 256 count pulses.

Pulse Width Measurement Operation

Besides the delay mode, timers A and B may be programmed to operate in the pulse width measurement mode. In this mode, an auxiliary control input is required; timers A and B auxiliary input lines are TAI and TBI. Also, in the pulse width measurement mode, interrupt channels normally associated with P4 and P3 will instead respond to transitions on TAI and TBI, respectively. General purpose lines P3 and P4 may still be used for I/O, but may not be used as interrupt generating inputs. A conceptual circuit of the selection of the interrupt source is shown in Figure 12.

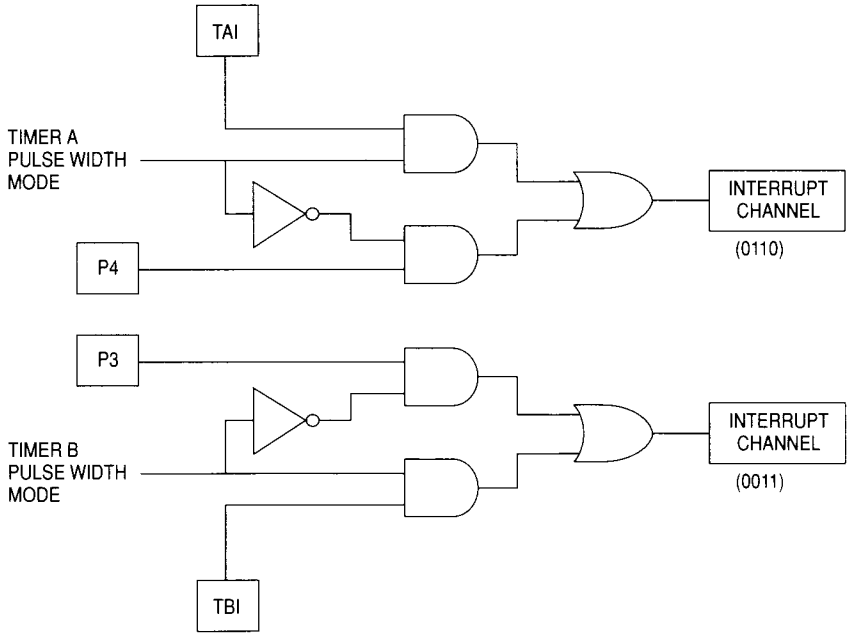


Figure 12. Conceptual Circuit of Interrupt Source Selection

The pulse width measurement mode functions similarly to the delay mode, with the auxiliary control signal acting as an enable to the timer. When the control signal is active, the prescaler and main counter are allowed to operate. When the control signal is negated, the timer is stopped. So, the width of the active pulse on TAI or TBI is measured by the number of timer counts which occur while the timer is allowed to operate.

The active state of the auxiliary input lines is defined by the associated interrupt channel's edge bit in the Active Edge Register (AER). GPIP4 of the AER is the edge bit associated with TAI, and GPIP3 is associated with TBI. When the edge bit is a one, the auxiliary input will be active high, enabling the timer while the input signal is at a high level. If the edge bit is zero, the auxiliary input will be active low and the timer will operate while the input signal is at a low level.

The state of the active edge bit also specifies whether a zero-to-one transition or a one-to-zero transition of the auxiliary input pin will produce an interrupt when the interrupt channel is enabled. In normal operation, programming the edge bit to a one will produce an interrupt on the zero-to-one transition of the associated input signal. Alternately, programming the edge bit to a zero will produce an interrupt on the one-to-zero transition of the input signal. However, in the pulse width measurement mode, the interrupt generated by a transition on TAI or TBI will occur on the opposite transition as that normally defined by the edge bit.

For example, in the pulse width measurement mode, if the edge bit is a one, the timer will be allowed or run while the auxiliary input is high. When the input transitions from high to low, the timer will stop and, if the interrupt channel is enabled, an interrupt will occur. By having the interrupt occur on the one-to-zero transition instead of the zero-to-one transition, the processor will be interrupted when the pulse being measured has been terminated and the width of the pulse from the timer is available.

After reading the contents of the timer, the processor must reinitialize the main counter by writing to the timer data register to allow consecutive pulses to be measured. If the data register is written after the auxiliary input signal become active, the timer will count from the previous contents of the timer data register until it counts through 01 (hexadecimal). At that time, the main counter is loaded with the new value from the timer data register, a time out pulse is generated which will toggle the timer output, and an interrupt may be optionally generated on the timer interrupt channel. Note that the pulse width measured includes counts from before the main counter was reloaded. If the timer data register is written while the pulse is transitioning to the active state, an indeterminate value may be written into the main counter.

Once the timer is reprogrammed for another mode, interrupts will again occur as normally defined by the edge bit. Note that an interrupt may be generated as the result of placing the timer into the pulse width measurement mode or by reprogramming the timer for another mode. Also, an interrupt may be generated by changing the state of the edge bit while in the pulse width measurement mode.

Event Count Mode Operation

In addition to the delay mode and the pulse width measurement mode, timer A and B may be programmed to operate in the event count mode. Like the pulse width measurement mode, the event count mode requires an auxiliary input signal, TAI or TBI. General purpose lines P3 and P4 can be used for I/O or as interrupt producing inputs.

In the event count mode, the prescaler is disabled allowing each active transition on TAI and TBI to produce a count pulse. The count pulse causes the main counter to decrement by one. When the timer counts through 01 (hexadecimal), a time out pulse is generated which will cause the output signal to toggle and may optionally produce an interrupt via the associated timer interrupt channel. The timer's main counter is also reloaded from the timer data register. To count transitions reliably, the input signal may only transition once every four timer periods. For this reason, the input signal must have a maximum frequency of one-fourth that of the timer clock.

The active edge of the auxiliary input signal is defined by the associated channel's edge bit. GPIP4 of the AER specifies the active edge for TAI, and GPIP3 defines the active edge for TBI. When the edge bit is programmed to a one, a count pulse will be generated on the zero-to-one transition of the auxiliary input signal. When the edge bit is programmed to a zero, a count pulse will be generated on the one-to-zero transition. Also, note that changing the status of the edge bit while the timer is in the event mode may produce a count pulse.

TIMER REGISTERS

The four timers are programmed via three control registers and four data registers. The following paragraphs describe the different registers.

Timer Data Register (TxDR)

The four Timer Data Registers (TDRs) are designated as Timer A Data Register (TADR), Timer B (TBDR), Timer C (TCDR), and Timer D (TDDR). Each timer's main counter is an 8-bit binary down counter. The timer data register contain the value of their respective main counter. This value was captured on the last low-to-high transition of the data strobe pin.

The main counter is initialized by writing to the TDR. If the timer is stopped, data is loaded simultaneously into both the TDR and main counter. If the TDR is written to while the timer is enabled, the value is not loaded into the timer until the timer counts through 01 (hexadecimal). If a write is performed while the timer is counting through 01, then an indeterminate value will be loaded into the timer's main counter.

Typical Timer Data Register

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0

TADR, TBDR, TCDR, TDDR

Timer Control Register (TxCR)

Timer Control Register A (TACR) and Timer Control Register B (TBCR) are associated with timer A and B, respectively. Timers C and D are programmed using one control register — the Timer C and D Control Register (TCDCR). The bits in the control register select the operation mode, prescaler value, and disable the timers. Both control registers have bits which allow the programmer to reset output lines TA0 and TB0.

7	6	5	4	3	2	1	0
*	*	*	RESET TA0	AC3	AC2	AC1	AC0

TACR

7	6	5	4	3	2	1	0
*	*	*	RESET TB0	BC3	BC2	BC1	BC0

* Unused bits read as zero

TBCR

Reset TAO/TBO — Reset Timer A and B Output Lines

TAO and TBO may be forced low at any time by writing a one to the reset location in TACR and TBCR. The output is held low during the write operation, and at the end of the bus cycle the output is allowed to toggle in response to a time-out pulse. When resetting TAO and TBO, the other bits in the TCR must be written with their previous value to avoid altering the operating mode.

AC3–AC0, BC3–BC0 — Select Timer A and B Operating Mode

When the timer is stopped, counting is inhibited. The contents of the timer's main counter is not affected, although any residual count in the prescaler is lost.

AC3 BC3	AC2 BC2	AC1 BC1	AC0 BC0	Operation Mode
0	0	0	0	Timer Stopped
0	0	0	1	Delay Mode, /4 Prescaler
0	0	1	0	Delay Mode, /10 Prescaler
0	0	1	1	Delay Mode, /16 Prescaler
0	1	0	0	Delay Mode, /50 Prescaler
0	1	0	1	Delay Mode, /64 Prescaler
0	1	1	0	Delay Mode, /100 Prescaler
0	1	1	1	Delay Mode, /200 Prescaler
1	0	0	0	Event Count Mode
1	0	0	1	Pulse Width Mode, /4 Prescaler
1	0	1	0	Pulse Width Mode, /10 Prescaler
1	0	1	1	Pulse Width Mode, /16 Prescaler
1	1	0	0	Pulse Width Mode, /50 Prescaler
1	1	0	1	Pulse Width Mode, /64 Prescaler
1	1	1	0	Pulse Width Mode, /100 Prescaler
1	1	1	1	Pulse Width Mode, /200 Prescaler

7	6	5	4	3	2	1	0
*	CC2	CC1	CC0	*	DC2	DC1	DC0

* Unused bits read as zero

TCDCR

CC2–CC0, DC3–DC0 — Select Timer C and D Operating Mode

When the timer is stopped, counting is inhibited. The contents of the timer's main counter is not affected, although any residual count in the prescaler is lost.

CC3 DC3	CC2 DC2	CC1 DC1	CC0 DC0	Operation Mode
0	0	0	0	Timer Stopped
0	0	0	1	Delay Mode, /4 Prescaler
0	0	1	0	Delay Mode, /10 Prescaler
0	0	1	1	Delay Mode, /16 Prescaler
0	1	0	0	Delay Mode, /50 Prescaler
0	1	0	1	Delay Mode, /64 Prescaler
0	1	1	0	Delay Mode, /100 Prescaler
0	1	1	1	Delay Mode, /200 Prescaler

USART — UNIVERSAL SYNCHRONOUS/ ASYNCHRONOUS RECEIVER-TRANSMITTER

The Universal Synchronous/Asynchronous Receiver-Transmitter (USART) is a single, full-duplex serial channel with a double-buffered receiver and transmitter. There are separate receive and transmit clocks and, also, separate receive and transmit status and data bytes. The receiver and transmit sections are also assigned separate interrupt channels. Each section has two interrupt channels: one for normal conditions and the other for error conditions. All interrupt channels are edge-triggered. Generally, it is the output of a flag bit or bits which is coupled to the interrupt channel. Thus, if an interrupt-producing event occurs while the associated interrupt channel is disabled, no interrupt would be produced, even if the channel was subsequently enabled because a transition did not occur while the channel was enabled. That particular event would have to occur again, generating another edge, before an interrupt would be generated. The interrupt channel may be disabled and instead, a DMA device can be used to transfer the data via the control signals Receive Ready (RR*) and Transmitter Ready (TR*). See **DMA Operation** for more information.

On reset the 68902 USART is configured to support a simple serial protocol transfer. Typically the receiver and transmitter are always enabled. In this mode the speed that the processor updates the USART registers should be faster than the USART receives or transmits data. If this is not the case then the I/O port can be configured to support serial hardware handshaking. This relives the software requirement of the processor and allows a fully hardware automatic method of transferring data serially by the hardware on the 68902. This handshaking operation is ideal for RS-232C type applications conforming to the V.24 protocol. See **CCITT V.24 Hardware Handshaking Mode** for more information.

Character Protocols

The 68902 USART supports asynchronous and, with the help of a polynomial generator checker, byte synchronous character formats. These formats are selected independently of the divide-by-one and divide-by 16 clock modes. It is possible to clock data synchronously into the 68902 but still use start and stop bits. After a start bit is detected, data will be shifted in and a stop bit will be checked to determine proper framing. In this mode, all normal asynchronous format features apply.

When the divide-by-one clock mode is selected, synchronization must be accomplished externally. The receiver will sample serial date on the rising edge of the receiver clock. In the divide-by-16 clock mode, the data is sampled at mid-bit time to increase noise rejection.

Also, when the divide-by-16 clock mode is selected the USART resynchronization logic is enabled. This logic increases the channels clock skew tolerance. Refer to **Asynchronous Format** for more information on the resynchronization logic.

Asynchronous Format

Variable character length and start/stop bit configurations are available under software control for asynchronous operation: The user can choose a character length from five to eight bits and a stop bit of one, one and a half, or two bits. The user can also select odd, even, or no parity.

In the asynchronous format, start bit detection is always enabled. New data is not shifted into the receive shift register until a zero bit is received. When the divide-by-16 clock mode is selected, the false start bit logic is also active. Any transition must be stable for three positive receive clock edges to be considered valid. For a start bit to be good, a valid zero-to-one transition must not occur for eight positive receiver clock transitions after the initial one-to-zero transition. After a valid start bit has been detected, the data is checked continuously for valid transitions. When a valid transition is detected, an internal counter is forced to state zero, and no further transition checking is initiated until state four. At state eight, the "previous state" of the transition checking logic is clocked into the receiver. As a result of this resynchronization logic, it is possible to run with asynchronous clocks without start and stop bits if there are sufficient valid transitions in the data streams.

Synchronous Format

When the synchronous character format is selected, the 8-bit synchronous character loaded into the Synchronous Character Register (SCR) is compared to receiver data until a match is found. Once synchronization is established, incoming data is clocked into the receiver. The synchronous word will be continuously transmitted during an underrun condition. All synchronous characters can be optionally stripped from the receive buffer (i.e., taken out of the data stream and thrown away) by clearing the appropriate bit in the Receive Status Register (RSR).

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0

SCR

The synchronous character should be written after the character length is selected, since unused bits in the synchronous character register are zeroed out. When parity is enabled, the synchronous word length is the character length plus one. The 68902 will compute and append the parity bit for the synchronous character when a character length of eight is selected. However, if the character length is less than eight, the user must determine the synchronous word parity and write it into the synchronous character register along with the synchro-

nous character. The parity bit must be the most-significant bit. The 68902 will then transmit the extra bit in the synchronous word as a parity bit.

USART Control Register (UCR)

This register selects the clock mode and the character format for the receive and transmit sections.

7	6	5	4	3	2	1	0
CLK	CL1	CL0	ST1	ST0	PE	E/O	*

* Unused bits read as zero

UCR

CLK — Clock Mode

1 = Data clocked into and out of receiver and transmitter at one sixteenth the frequency of their respective clocks.

0 = Data clocked into and out of receiver and transmitter at the frequency of their respective clocks.

CL1, CL0 — Character Length

These bits specify the length of the character exclusive of start bits, and parity.

CL1	CL0	Character Length
0	0	8 Bits
0	1	7 Bits
1	0	6 Bits
1	1	5 Bits

ST0–ST1 — Start/Stop Bit and Format Control

These bits select the number of start and stop bits and specify the character format.

ST1	ST0	Start Bits	Stop Bits	Format
0	0	0	0	Synchronous
0	1	1	1	Asynchronous
1	0	1	1-1/2	Asynchronous*
1	1	1	2	Asynchronous

* Used with Divide-by-16 mode only

PE — Parity Enable

Parity is not automatically appended to the synchronous character for character lengths of less than eight bits. Therefore, parity should be written into the SCR along with the synchronous character.

1 = Parity checked by receiver and parity calculated and inserted during data transmission.

0 = No parity check and no parity bit computed for transmission.

E/O — Even/Odd Parity

1 = Even parity is selected.

0 = Odd parity is selected.

Bit 0 — Not used

RECEIVER

As data is received on the Serial Input line (SI), it is clocked into an internal 8-bit shift register until the specified number of data bits have been assembled. The character will then be transferred to the receive buffer, assuming that the last word in the receive buffer has been read. This transfer will set the buffer full bit in the Receiver Status Register (RSR) and produce a buffer full interrupt to the processor, assuming this interrupt has been enabled.

Reading the receive buffer satisfies the buffer full condition and allows a new data word to be transferred to the receive buffer when it is assembled. The receive buffer is accessed by reading the USART Data Register (UDR). The UDR is simply an 8-bit register used when transferring data between the 68902 and the main processor.

Each time a word is transferred to the receiver buffer, its status information is latched into the Receiver Status Register (RSR). The RSP is not updated again until the data word in the receive buffer has been read. When a buffer full condition exists, the RSR should always be read before the receive buffer (UDR) to maintain the correct correspondence between data and flags. Otherwise, it is possible that after reading the UDR and prior to reading the RSR, a new word could be received and transferred to the receive buffer. Its associated flags would be latched into the RSR, writing over the flags for the previous data word. Thus, when the RSR was read to access the status information for the first data word, the flags for the new word would be retrieved.

Receiver Interrupt Channels

The USART receiver section is assigned two interrupt channels. One indicates the buffer full condition while the other channel indicates an error condition. Error conditions include overrun, parity error, synchronous found, and break. These interrupting conditions correspond to the OE, PE, FE, and F/S or B bits of the receiver status register. These flags will function as described in **Receiver Status Register** whether the receive interrupt channels are enabled or disabled.

While only one interrupt is generated per character received, two dedicated interrupt channels allow separate vector numbers to be assigned for normal and abnormal receiver conditions. When a receiver word has an error associated with it and the error interrupt channel is enabled, an interrupt will be generated on the error channel only. However, if the error channel is disabled, an interrupt for an error condition will be generated on the buffer full interrupt channel along with interrupts produced by the buffer full condition. The receiver status register must always be read to determine which error condition produced the interrupt.

Receiver Status Register (RSR)

The RSR contains the receiver buffer full flag, the synchronous strip enable, the various status information associated with the data word in the receive buffer. The RSR is latched each time a data word is transferred to the receive buffer. RSR flags cannot change again until the new data word has been read. However, the M/CIP bit is allowed to change.

7	6	5	4	3	2	1	0
BF	OE	PE	FE	F/S or B	M/CIP	SS	RE

RSR

BF — Buffer Full

1 = Receiver word is transferred to the receive buffer.
0 = Receiver buffer is read by accessing the USART data register.

OE — Overrun Error

Overrun error occurs when a receiver word is to be transferred to the receive buffer, but the buffer is full. Neither the receiver buffer not the RSR is overwritten.

1 = Receiver buffer full.
0 = Read by RSR or 68902 reset.

PE — Parity Error

1 = Parity error detected on character transfer to receive buffer.
0 = No parity error detected on character transfer to receive buffer.

FE — Frame Error

A frame error exists when a non-zero data character is not followed by a stop bit in the asynchronous character format.

1 = Frame error detected on character transfer to receive buffer.
0 = No frame error detected on character transfer to receive buffer.

F/S or B — Found/Search or Break Detect

The F/S bit is used in the synchronous character format. When set to zero, the USART receiver is placed in the search mode. F/S is cleared when the incoming character does not match the synchronous character.

1 = Match found. Character length counter enabled.
0 = Incoming data compared to SCR. Character length counter disabled.

The B bit is used in the asynchronous character format. This flag indicates a break condition which continues until a non-zero data bit is recovered.

1 = Character transferred to the receive buffer is a break condition.
0 = Non-zero data bit received and break condition was acknowledged by reading the RSR at least once.

M or CIP — Match/Character in Progress

The M bit is used in the synchronous character format and indicates a synchronous character has been received.

1 = Character transferred to the receiver buffer matches the synchronous character.
0 = Character transferred to the receiver buffer does not match the synchronous character.

The CIP bit is used in the asynchronous character format and indicates that a character is being assembled.

1 = Start bit is detected.
0 = Final stop bit has been received.

SS — Synchronous Strip Bit

1 = Characters that match the synchronous character will not be loaded into the receiver buffer and no buffer full condition will be produced.
0 = Characters that match the synchronous character will be transferred to the receive buffer and a buffer full condition will be produced.

RE — Receiver Enable

This bit should not be set until the receiver clock is active. When the transmitter is disabled in auto-turnaround mode this bit is set.

1 = Receiver operation is enabled.
0 = Receiver is disabled.

Special Receive Conditions

Certain receive conditions relating to the overrun error flag and the break detect flag require further explanation. Consider the following examples:

1) A break is received while the receive buffer is full. This does not produce an overrun condition. Only the B flag will be set after the receiver buffer is read.
2) A new word is received, and the receive buffer is full. A break is received before the receive buffer is read. Both the B and OE flags will be set when the buffer full condition is satisfied.

Transmitter Interrupt Channels

The USART transmit section is assigned two interrupt channels. The normal channel indicates a buffer empty condition, and the error channel indicates an underrun or end condition. These interrupting conditions correspond to the BE, UE, and END flags in the TSR. The flag bits will function as described in **Transmitter Status Register** whether their associated interrupt channel is enabled or disabled.

Transmitter Status Register

The TSR contains various transmitter error flags and transmitter control bits for selecting auto-turnaround and loopback mode.

7	6	5	4	3	2	1	0
BE	UE	AT	END	B	H	L	TE

TSR

BE — Buffer Empty

1 = Character in the transmit buffer transferred to TSR.
0 = Transmit buffer reloaded by writing to the USR.

U — Underrun Error

One full transmitter clock is required after UE bit is set before it can be cleared. This bit does not require clearing before writing to the UDR.

1 = Character in the TSR was transmitted before a new word was loaded into the transmit buffer.
0 = Transmitter disabled or read performed to TSR.

AT — Auto-Turnaround

When set, the receiver will be enabled automatically after the transmitter has been disabled and the last character being transmitted is complete.

END — End of Transmission

If the transmitter is disabled while a character is being transmitted, this bit is set after transmission is complete. If no character was being transmitted, then this bit is set immediately. Re-enabling the transmitter clears this bit.

B — Break

This bit only functions in the asynchronous format. When B is set, BE cannot be set. A break consists of all zeros with no stop bit. This bit cannot be set until the transmitter is enabled and internal reset and initialization is complete.

1 = Break transmitted and transmission stops.
0 = Break ceases and normal transmission resumes.

H, L — High and Low

These bits configure the Transmitter Output (SO) when the transmitter is disabled. Changing these bits after the transmitter is enabled will alter the output state until END is cleared.

H	L	Output State
0	0	High Impedance
0	1	Low
1	0	High
1	1	Loopback Mode

TE — Transmitter Enable

The serial output will be driven according to H and L bits until transmission begins. A one bit is transmitted before character transmission in the TSR begins.

1 = Transmitter enabled.
0 = Transmitter disabled. UE bit cleared and END bit set.

DMA OPERATION

USART error conditions are valid only for each character boundary. When the USART performs block data transfers by using the DMA handshake lines Receive Ready (RR*) and Transmitter Ready (TR*), errors must be saved and checked at the end of a block. This is accomplished by enabling the error channel for the receiver or transmitter and by masking interrupts for this channel. Once the transfer is complete, interrupt pending register A is read. Any pending receiver or transmitter error indicates an error in the data transfer.

RR* is asserted when the buffer full bit is set in the RSR unless a parity error or frame error is detected by the receiver. TR* is asserted when the buffer empty bit is set in the TSR unless a break is currently being transmitted.

ELECTRICAL CHARACTERISTICS

This section contains the electrical specifications and associated timing information for the 68902.

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	– 0.5 to 7.0	V
Input Voltage	V_{in}	–1.5 to V_{CC} + 1.5	V
Operating Temperature Range	T_A	– 40 to 85	°C
Operating Temperature Hi-Rel	T_A	– 55 to 125	°C
Storage Temperature Range	T_{stg}	– 65 to 150	°C
Power Dissipation	P_D	TBD	W
Power Dissipation Hi-Rel	P_D	TBD	W

This device contains circuitry to protect the inputs against damage due to high static Voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than the maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{CC} or GND). To prevent unecessary power consumption there are no internal pull-up or pull-down resistors on these inputs. The user must therefore take care of the voltage levels on these pins.

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Symbol	Value	Rating
Thermal Resistance	θ_{JA}		θ_{JC}		°C/W
Ceramic		TBD		TBD	
Plastic		TBD		TBD	

TABLE 1. AC ELECTRICAL CHARACTERISTICS

V_{DD} = 5.0 V ± 10%, V_{SS} = 0 V; T_C = – 55°C to 125°C

Number	Characteristic	Min	Max	Unit
1	CLK1 Cycle Time	50	—	ns
2	CLK1 Width High	20	—	ns
3	CLK1 Width Low	20	—	ns
4	RS1 to RS5 Setup Time to Rising Edge of CLK1	8	—	ns
5	SEL* Setup time to Rising Edge of CLK1	11	—	ns
6	DATA_LATCH* Setup Time to Rising Edge of CLK1	17	—	ns
7	Data Bus Driven from the Rising Edge of CLK1	—	31	ns
8	Time for ACK* Asserted from Falling Edge of CLK1	—	46	ns
9	Time Taken to Negate ACK* from the Falling Edge of CLK1	—	34	ns
10	Hold Time before Data Bus is Tri-Stated from the Rising Edge of CLK1	—	31	ns
11	DATA_LATCH* Hold Time from the Rising Edge of CLK1	7	—	ns
12	SEL* Hold Time from the Rising Edge of CLK1	7	—	ns
13	RS1–RS5 Hold Time from Rising Edge of CLK1	9	—	ns
14	CLK1 Low to ACK* High Impedance	—	73	ns

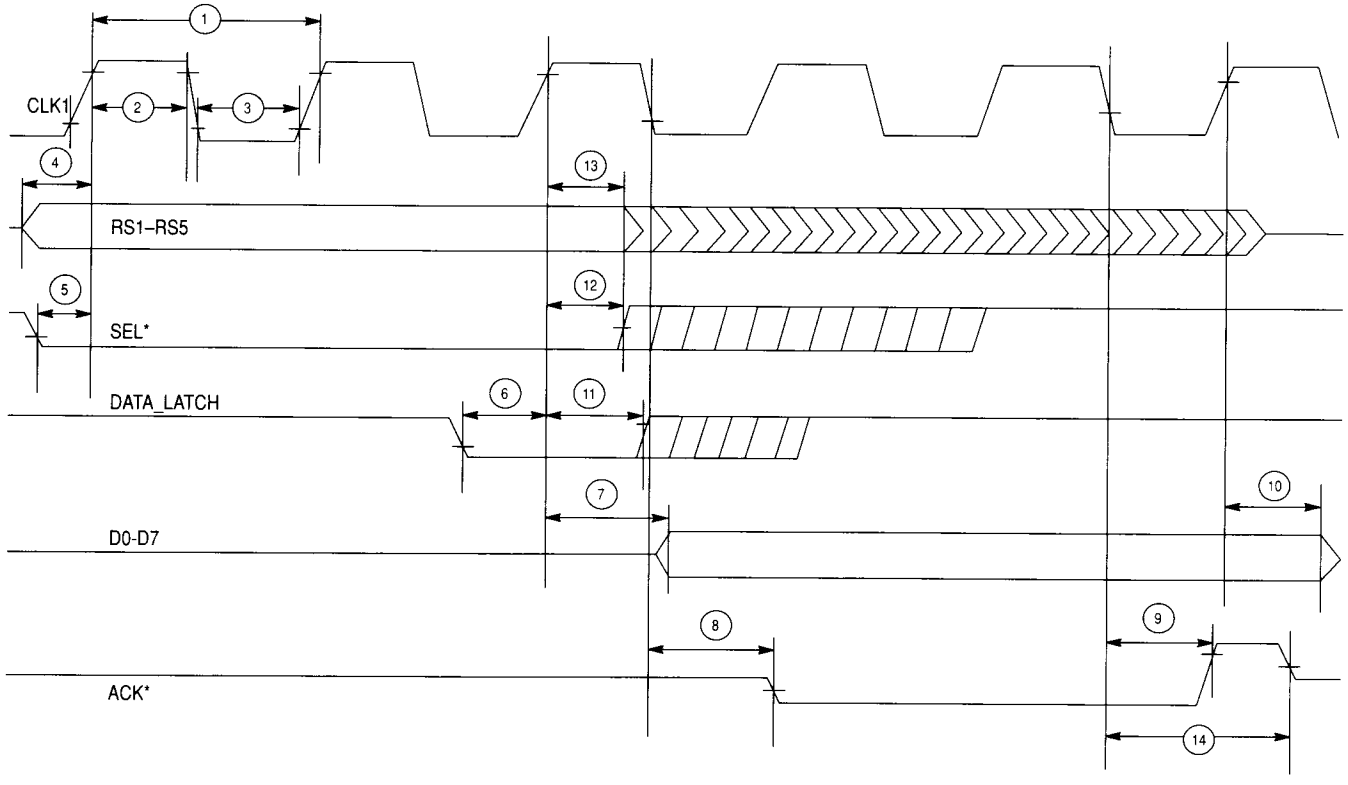


Figure 13. MC68000 through to MC68030 Read Cycle

TABLE 2. AC ELECTRICAL CHARACTERISTICS

V_{DD} = 5.0 V ± 10%, V_{SS} = 0 V; T_C = – 55°C to 125°C

Number	Characteristic	Min	Max	Unit
15	W* Setup Time before the Rising Edge of CLK1	9	—	ns
16	Data Bus Hold Time from the Rising Edge of CLK1	8	—	ns
17	W* Hold Time from the Rising Edge of CLK1	7	—	ns
18	Data Bus Setup Time from the Rising Edge of CLK1	19	—	ns

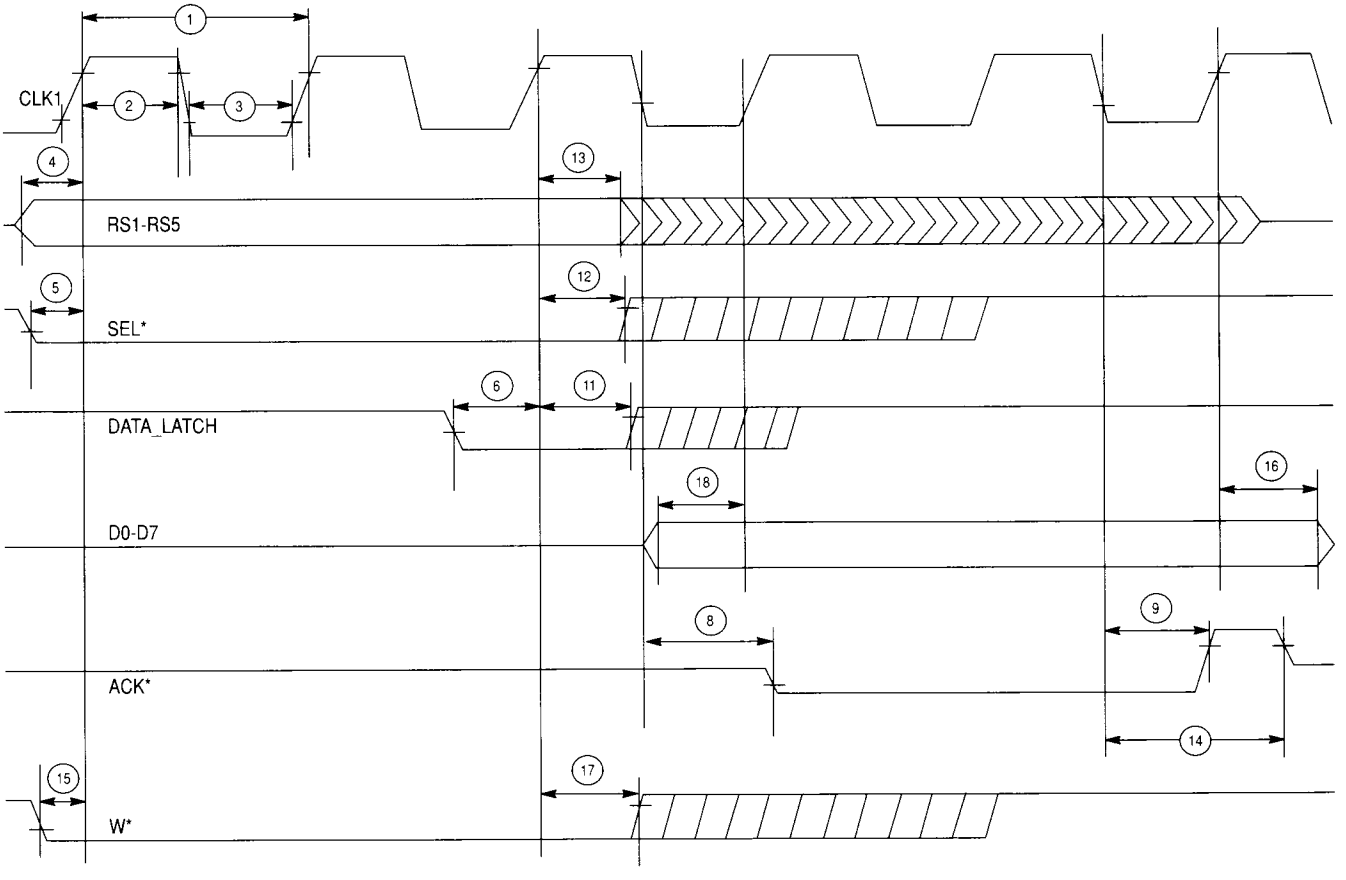


Figure 14. MC68000 through to MC68030 Write Cycle

TABLE 3. AC ELECTRICAL CHARACTERISTICS

V_{DD} = 5.0 V ± 10%, V_{SS} = 0 V; T_C = – 55°C to 125°C

Number	Characteristic	Min	Max	Unit
19	INT_ACK* Setup Time before the Rising Edge of CLK1	9	—	ns
20	INT_ACK* Hold Time after the Rising Edge of CLK1	8	—	ns
21	IEI* Setup Time before the Rising Edge of CLK1	14	—	ns
22	IEI* Hold Time after the Rising Edge of CLK1	8	—	ns

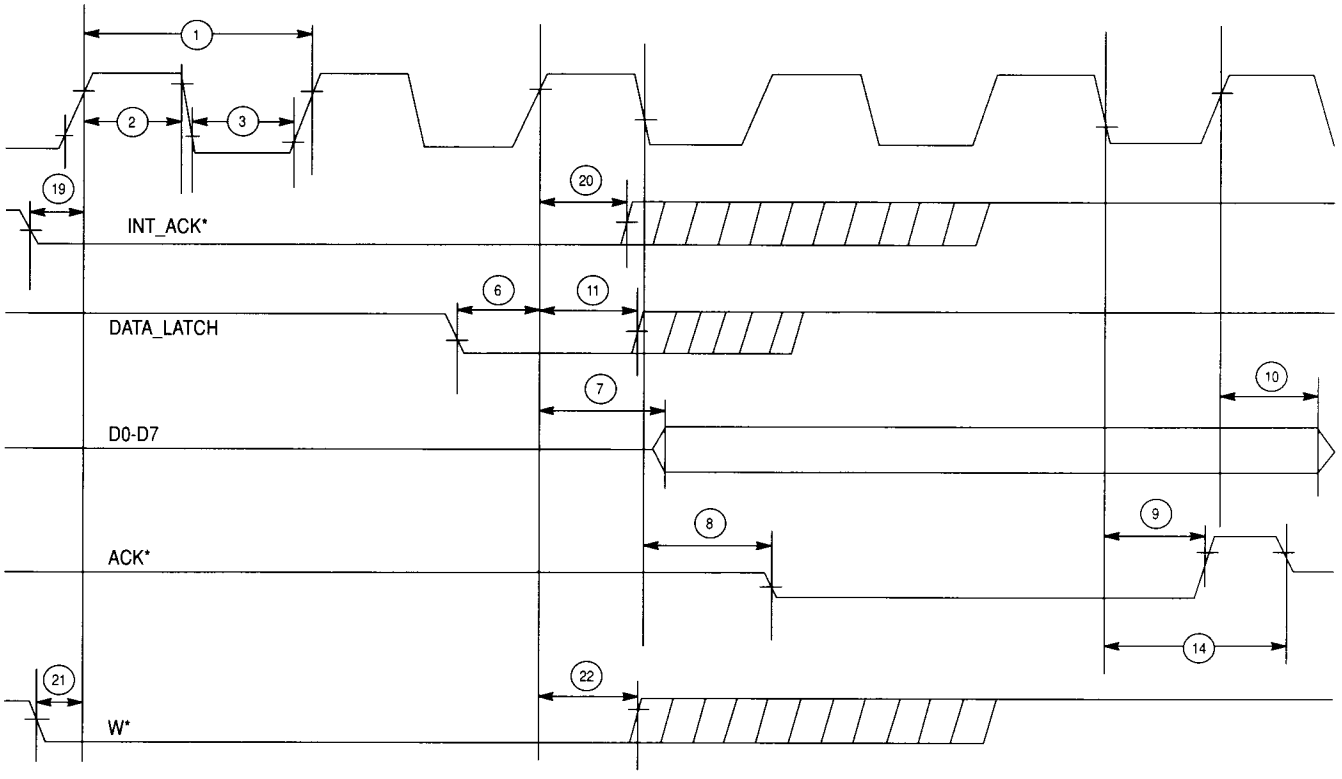


Figure 15. MC68000 through to MC68030 Interrupt Cycle

TABLE 4. AC ELECTRICAL CHARACTERISTICS

V_{DD} = 5.0 V ± 10%, V_{SS} = 0 V; T_C = – 55°C to 125°C

Number	Characteristic	Min	Max	Unit
23	RS1 to RS5 Setup Time to Rising Edge of CLK1	8	—	ns
24	SEL* Setup Time to Rising Edge of CLK1	11	—	ns
25	DATA_LATCH* Setup Time to Rising Edge of CLK1	17	—	ns
26	Data Bus Driven from the Rising Edge of CLK1	31	—	ns
27	Time for ACK* Asserted from Rising Edge of CLK1	—	41	ns
28	Time Taken to Negate ACK* from the Rising Edge of CLK1	—	35	ns
29	Hold Time before Data Bus is Tri-Stated from the Rising Edge of CLK1	—	31	ns
30	DATA_LATCH* Hold Time from the Rising Edge of CLK1	7	—	ns
31	SEL* Hold Time from the Rising Edge of CLK1	7	—	ns
32	RS1–RS5 Hold Time from Rising Edge of CLK1	9	—	ns
33	CLK1 High to ACK* High Impedance	—	67	ns

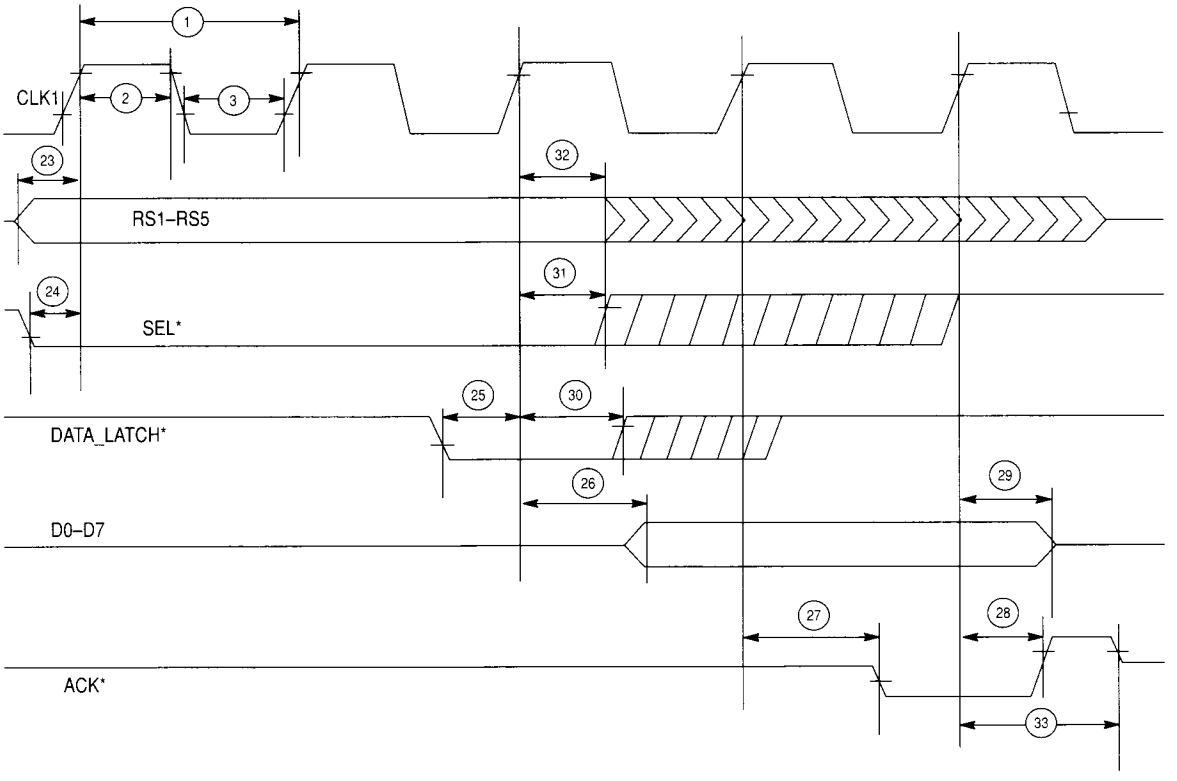


Figure 16. MC68040 Read Cycle

TABLE 5. AC ELECTRICAL CHARACTERISTICS

V_{DD} = 5.0 V ± 10%, V_{SS} = 0 V; T_C = – 55°C to 125°C

Number	Characteristic	Min	Max	Unit
34	W* Setup Time before the Rising Edge of CLK1	9	—	ns
35	Data Bus Hold Time from the Rising Edge of CLK1	9	—	ns
36	W* Hold Time from the Rising Edge of CLK1	7	—	ns
37	Data Bus Setup Time from the Falling Edge of CLK1	19	—	ns

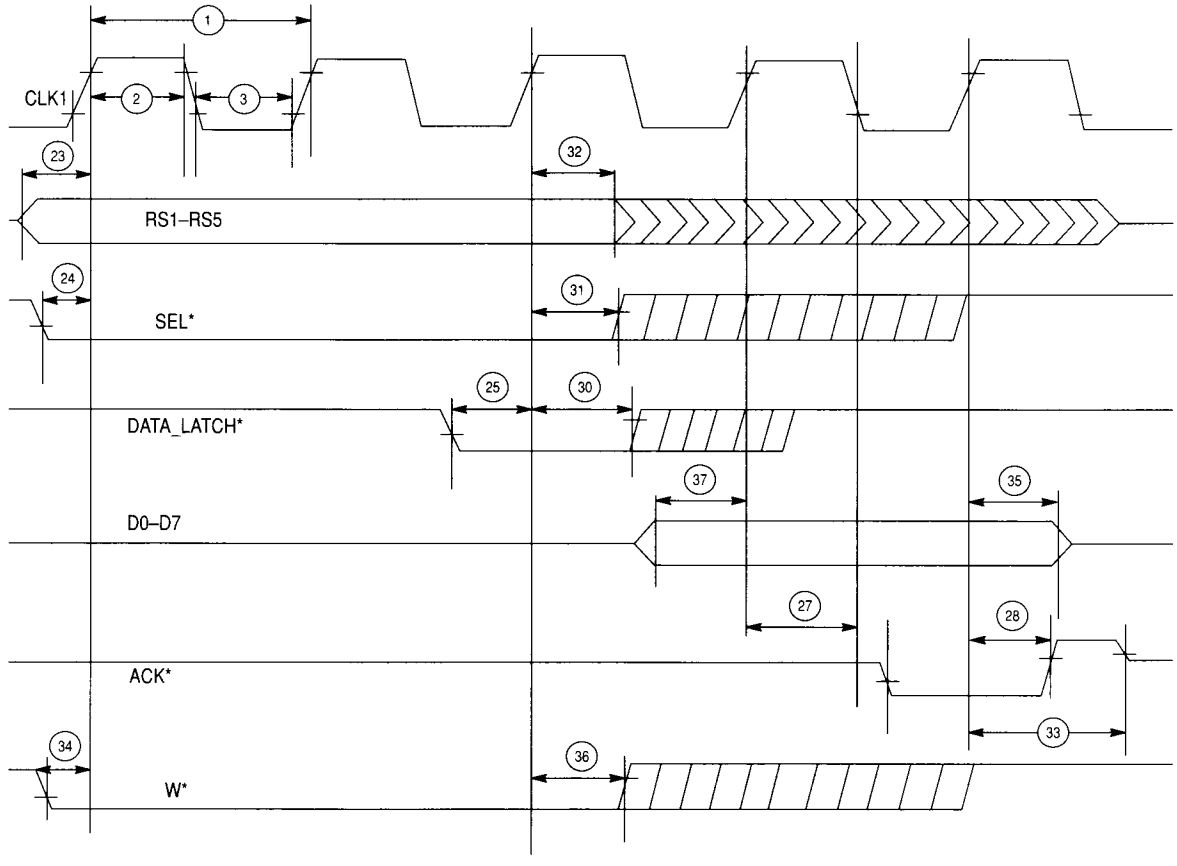


Figure 17. MC68040 Write Cycle

TABLE 6. AC ELECTRICAL CHARACTERISTICS

V_{DD} = 5.0 V ± 10%, V_{SS} = 0 V; T_C = – 55°C to 125°C

Number	Characteristic	Min	Max	Unit
38	INT_ACK* Setup Time before the Rising Edge of CLK1	9	—	ns
39	INT_ACK* Hold Time after the Rising Edge of CLK1	8	—	ns
40	IEI* Setup Time before the Rising Edge of CLK1	14	—	ns
41	IEI* Hold Time after the Rising Edge of CLK1	8	—	ns

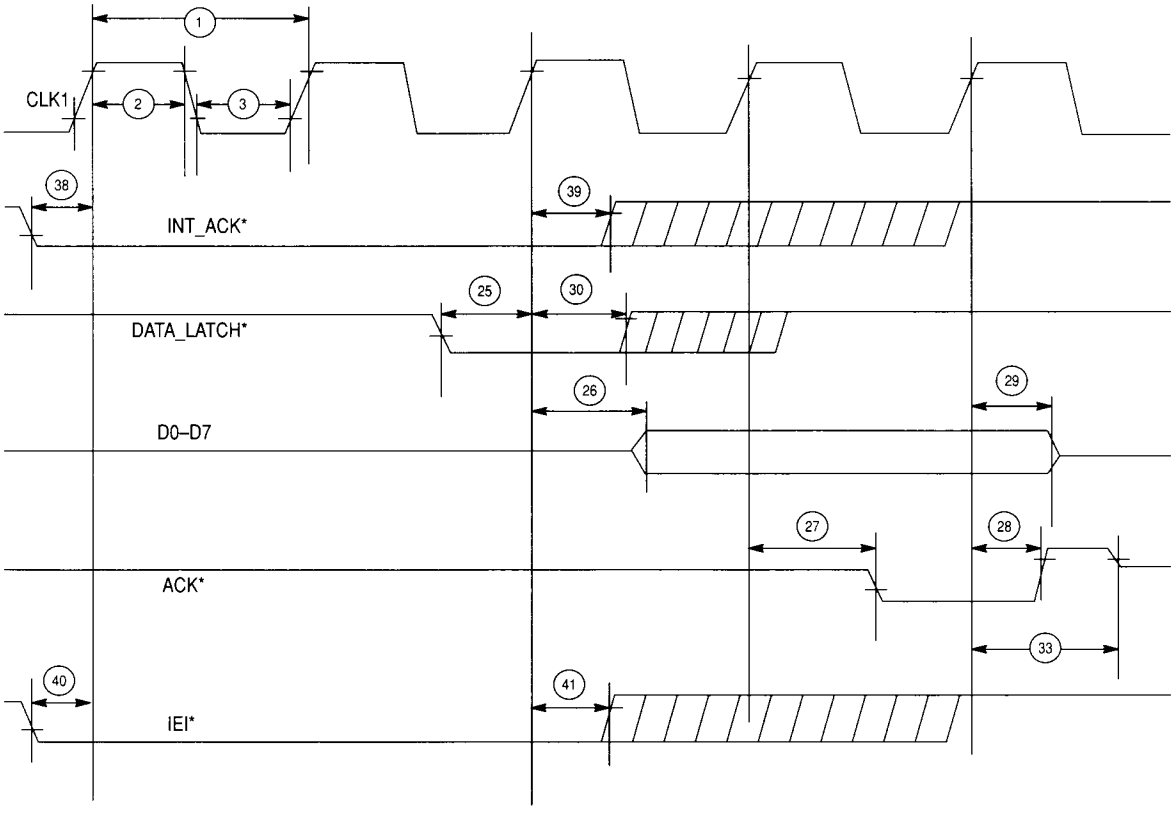


Figure 18. MC68040 Interrupt Acknowledge Cycle

TABLE 7. AC ELECTRICAL CHARACTERISTICS

V_{DD} = 5.0 V ± 10%, V_{SS} = 0 V; T_C = – 55°C to 125°C

Number	Characteristic	Min	Max	Unit
42	Setup Time for P0–P7 from the Rising Edge of CLK1	6	—	ns
43	P0–P7 Pulse Duration to Ensure Interrupt Recognition	2	—	t clk1
44	Duration of Time Before Interrupt from a Port Line is Asserted	—	2	t clk1
45	Assertion of IRQ* from the Rising Edge of CLK1	—	32	ns
46	Output on P0–P7 from the Rising Edge CLK1	—	20	ns
47	Power on RESET Pulse Duration	2	—	µs
48	Warm RESET Pulse Duration	10	—	t clk1
49	Skew Between ACK* (High to Low) and ACK (Low to High)	—	10	ns
50	Skew Between ACK* (Low to High) and ACK (High to Low)	—	10	ns
51	IEO* Asserted from the Falling Edge of IEI*	—	23	ns
52	IEO* Negated from the Rising Edge of IEI*	—	18	ns

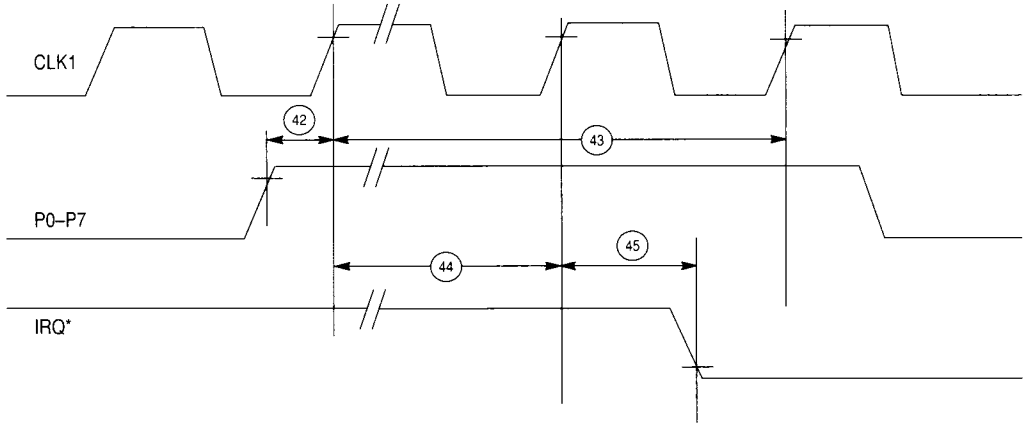


Figure 19. Interrupt Timing for Normal Port Operation

Note: 1) Active Edge is assumed to be the rising edge
2) If spec. 42 is not satisfied the signal is not recognized until the next rising edge.
3) The port line should be stable for a duration of spec. 43 for the Interrupt to be recognized.
4) The interrupt will be asserted after spec. 44.
5) The interrupt line is asserted after spec. 45 after the rising edge of the clock.

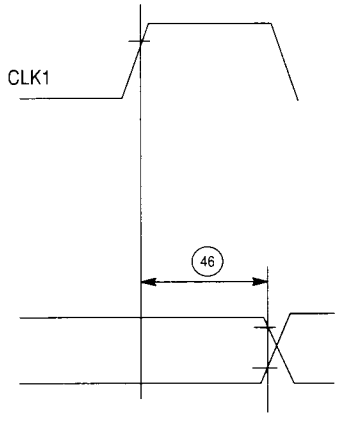
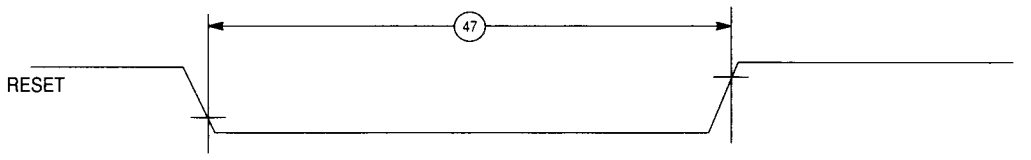
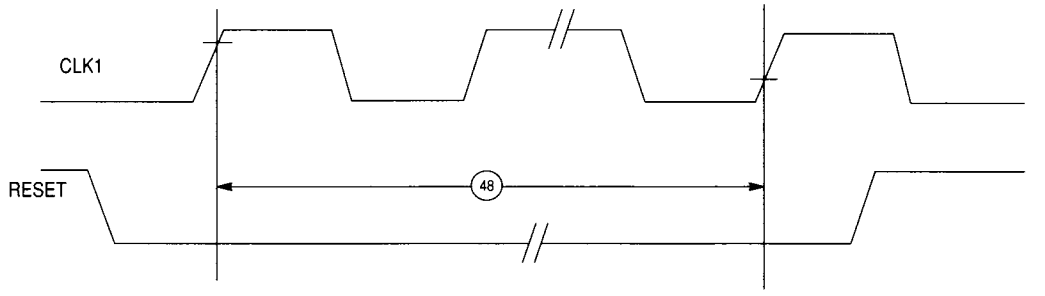


Figure 20. Port Timing



Note: The signal MPU should be hardwired during RESET.

Figure 21. Power on RESET



Note: 1) A WARM RESET require a minimum time of spec. 48.

Figure 22. WARM RESET

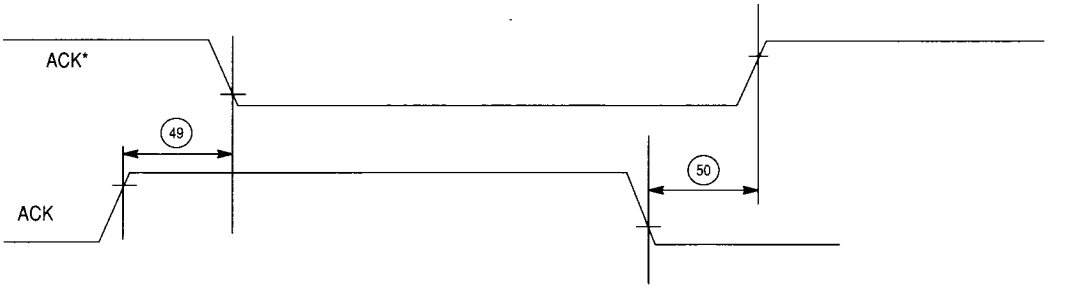


Figure 23. ACK Timing

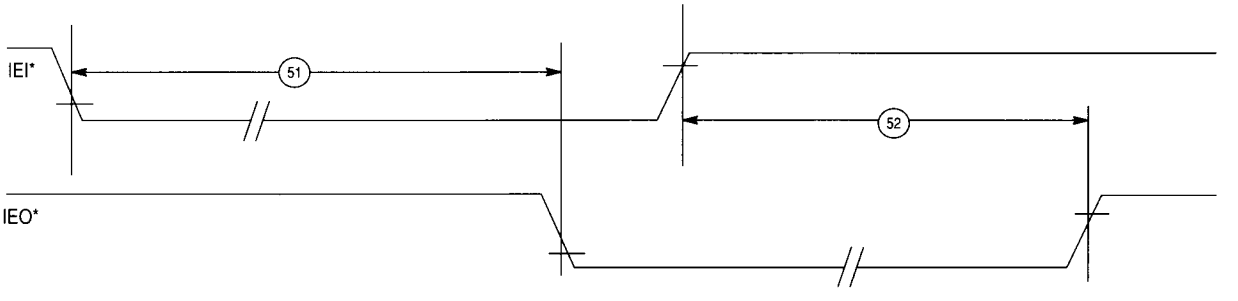


Figure 24. IEO* Timing

TABLE 8. AC ELECTRICAL CHARACTERISTICS

V_{DD} = 5.0 V ± 10%, V_{SS} = 0 V; T_C = – 55°C to 125°C

Number	Characteristic	Min	Max	Unit
53	Receiver Clock (RC) Cycle Time	4	—	t clk1
54	Receiver Clock (RC) High Time	1.5*	—	t clk1
55	Receiver Clock (RC) Low Time	1.5*	—	t clk1
56	Max. No. CLK1 Periods from Receive Register Full to RR* Asserted	—	1	t clk1
57	RR* Asserted from Rising Edge of CLK1	—	27	ns
58	Max. No. CLK1 Periods from Error Condition to IRQ* Asserted	—	3	t clk1
59	IRQ* Asserted from Rising Edge of CLK1 (Receiver)	—	32	ns
60	Setup Time for the Serial Input Data from the Rising Edge of RC	9	—	ns
61	Hold Time for the Serial Input Data from the Rising Edge of RC	tclk1+10	—	ns
62	Transmitter Clock (TC) Cycle Time	4	—	t clk1
63	Transmitter Clock (TC) High Time	1.5*	—	t clk1
64	Transmitter Clock (TC) Low Time	1.5*	—	t clk1
65	Serial Output Data after the Rising Edge of TC	3tclk1+38	—	ns
66	Max. No. CLK1 Periods from Buffer Empty to IRQ* Asserted	—	2	t clk1
67	IRQ* Asserted from Rising Edge of CLK1 (Transmitter)	—	32	ns
68	TR* Asserted from the Rising Edge of CLK1	—	26	ns
69	Timer Crystal Input (XTAL1/XTAL2) Cycle Time	50	—	ns
70	Timer Crystal Input (XTAL1/XTAL2) High Time	20	—	ns
71	Timer Crystal Input (XTAL1/XTAL2) Low Time	20	—	ns
72	TxO Asserted from Rising Edge of Timer Clock	—	—	—
73	Max. No. XTAL Periods from RESET Write Cycle to TAO/TBO Asserted	—	2	t xtal
74	TAO/TBO Asserted from Rising Edge of XTAL (Timer RESET Mode)	—	51	ns

* 1 = t clk1 + t setup + t hold

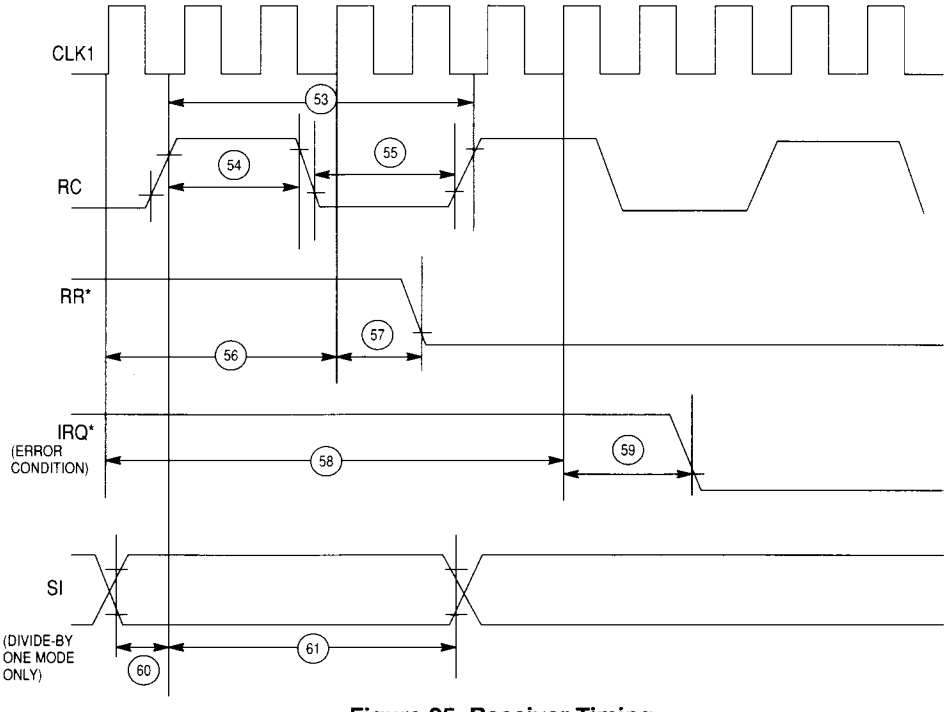


Figure 25. Receiver Timing

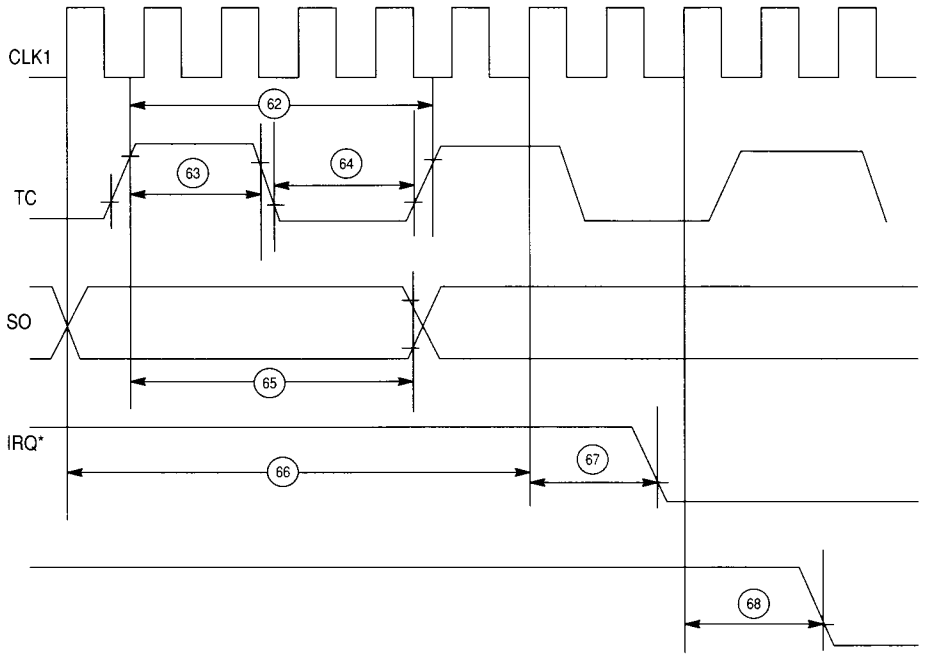


Figure 26. Transmitter Timing

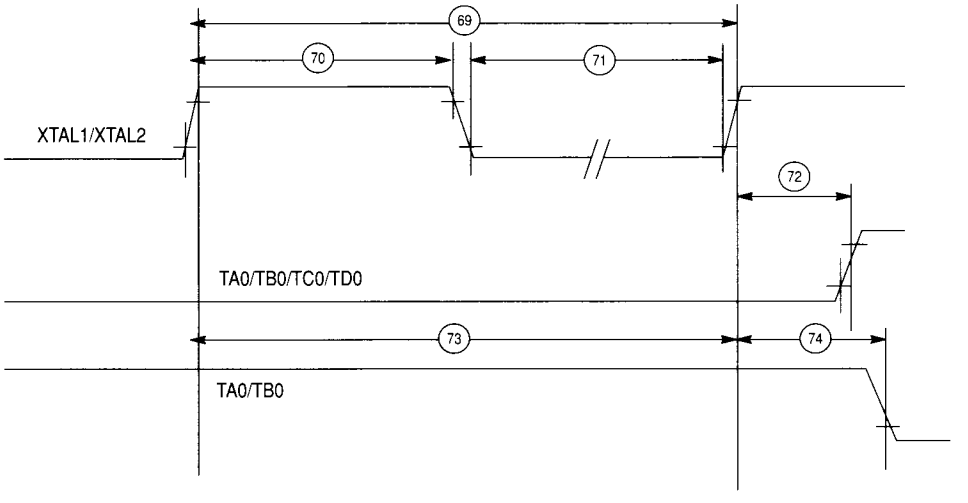
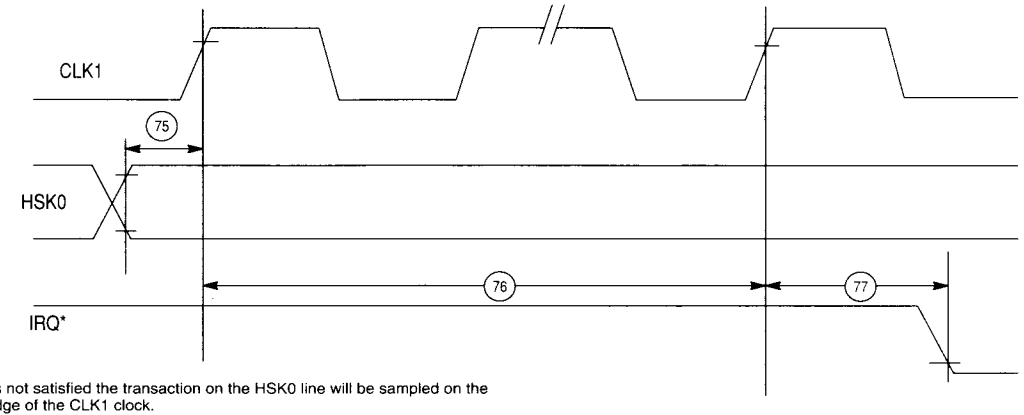


Figure 27. Timer Timing

TABLE 9. AC ELECTRICAL CHARACTERISTICS

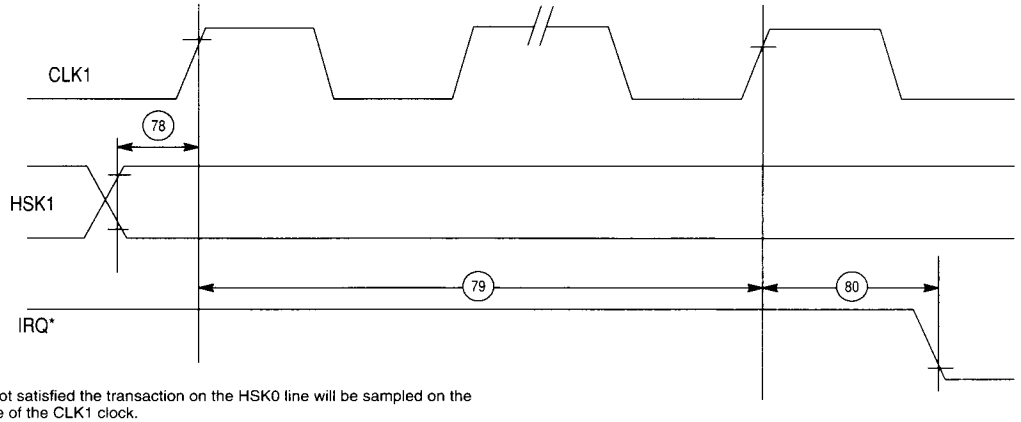
V_{DD} = 5.0 V ± 10%, V_{SS} = 0 V; T_C = – 55°C to 125°C

Number	Characteristic	Min	Max	Unit
75	Setup Time for HSK0 before the Rising Edge of CLK1	9	—	ns
76	Number of CLK1 Clock Periods required before IRQ* Recognized	2	—	t clk1
77	IRQ* Asserted from the Rising Edge of CLK1	—	32	ns
78	Setup Time for HSK1 before the Rising Edge of CLK1	8	—	ns
79	Number of CLK1 Clock Periods required before IRQ* Recognized	2	—	t clk1
80	IRQ* Asserted from the Rising Edge of CLK1	—	32	ns
81	HSK1 Asserted from the Rising Edge of CLK1	—	27	ns
82	HSK0 Setup Time to Rising Edge CLK1	9	—	ns
83	HSK1 Negation from Rising Edge of CLK1	—	25	ns
84	HSK1 Asserted from Rising Edge of CLK1	—	27	ns
85	HSK1 Negation from Rising Edge of CLK1	—	25	ns
86	No. CLK1 Periods Required from loading HSKCR0 to HSK1 Asserted	—	1	t clk1
87	HSK1 Asserted from the Rising Edge of CLK1	—	27	ns



Note: If timing 75 is not satisfied the transaction on the HSK0 line will be sampled on the next rising edge of the CLK1 clock.

Figure 28. HSK0 Timing



Note: If timing 78 is not satisfied the transaction on the HSK0 line will be sampled on the next rising edge of the CLK1 clock.

Figure 29. HSK1 Timing

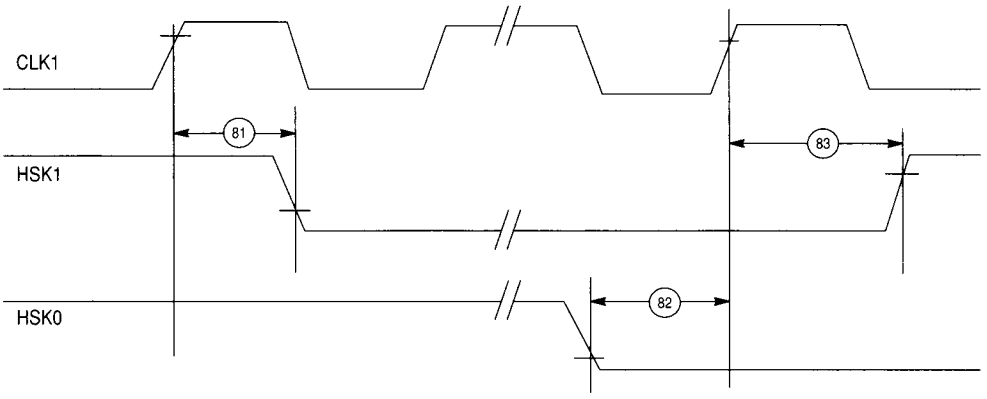


Figure 30. HSK1/HSK0 Timing
(Mode bits 5, 4, 3 = 100 in HSKCR0 Register)

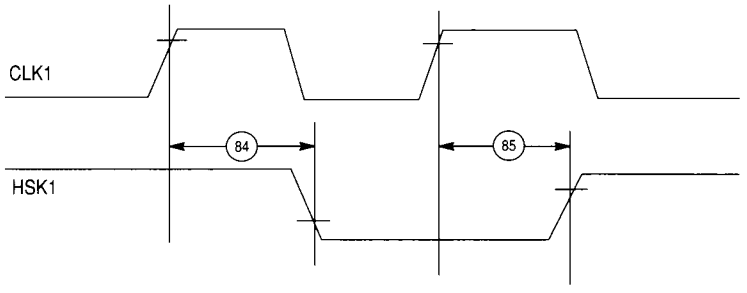


Figure 31. HSK1 Timing
(Mode bits 5, 4, 3 = 101 in HSKCR0 Register)

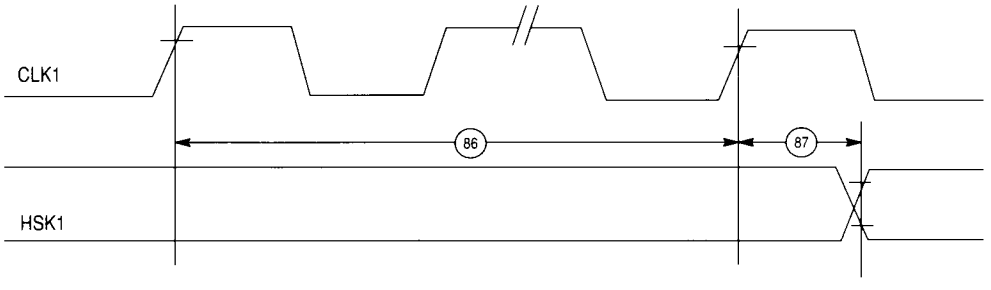
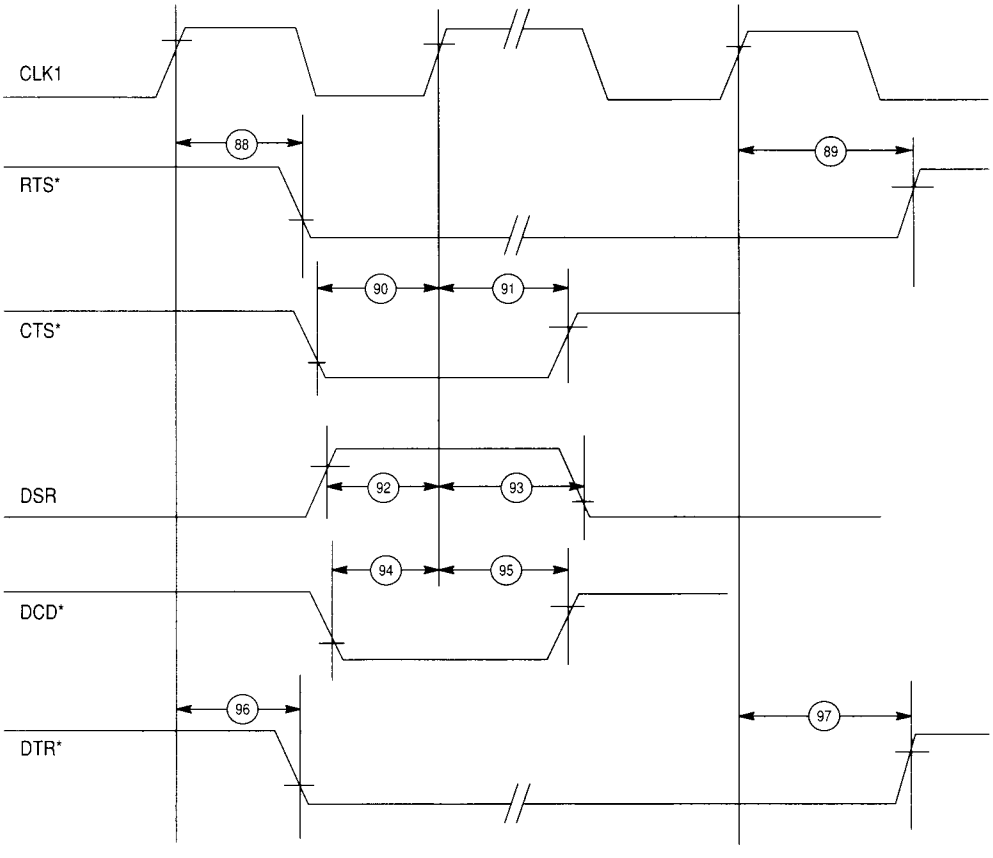


Figure 32. HSK1 Timing
(Mode bits 5, 4, 3 = 110 or 111 in HSKCR0 Register)

TABLE 10. AC ELECTRICAL CHARACTERISTICS

V_{DD} = 5.0 V ± 10%, V_{SS} = 0 V; T_C = – 55°C to 125°C

Number	Characteristic	Min	Max	Unit
88	RTS* Asserted from Rising Edge of CLK1	—	36	ns
89	RTS* Negated from Rising Edge of CLK1	—	34	ns
90	CTS* Setup Time before Rising Edge of CLK1	24	—	ns
91	CTS* Hold Time after the Rising Edge of CLK1	6	—	ns
92	DSR Setup Time before Rising Edge of CLK1	13	—	ns
93	DSR Hold Time after the Rising Edge of CLK1	5	—	ns
94	DCD* Setup Time before Rising Edge of CLK1	38	—	ns
95	DCD* Hold Time after the Rising Edge of CLK1	6	—	ns
96	DTR* Asserted from Rising Edge of CLK1	—	29	ns
97	DTR* Negated from Rising Edge of CLK1	—	26	ns



Note: The pulse width of CTS*, Pw(CTS*), must be greater than or equal to the high period of the Transmitter clock, TC. Therefore: Pw(CTS*) >= t high (TC)

Figure 33. V.24 Handshaking Control Lines Timing

PACKAGING INFORMATION

The 68902 is available to full MIL-STD-883C Class B specification and to Motorola Enhanced Processing (MEP) flows in a 68 Ceramic Leaded Chip Carrier (CLCC) and a 68 Ceramic Pin Grid Array.

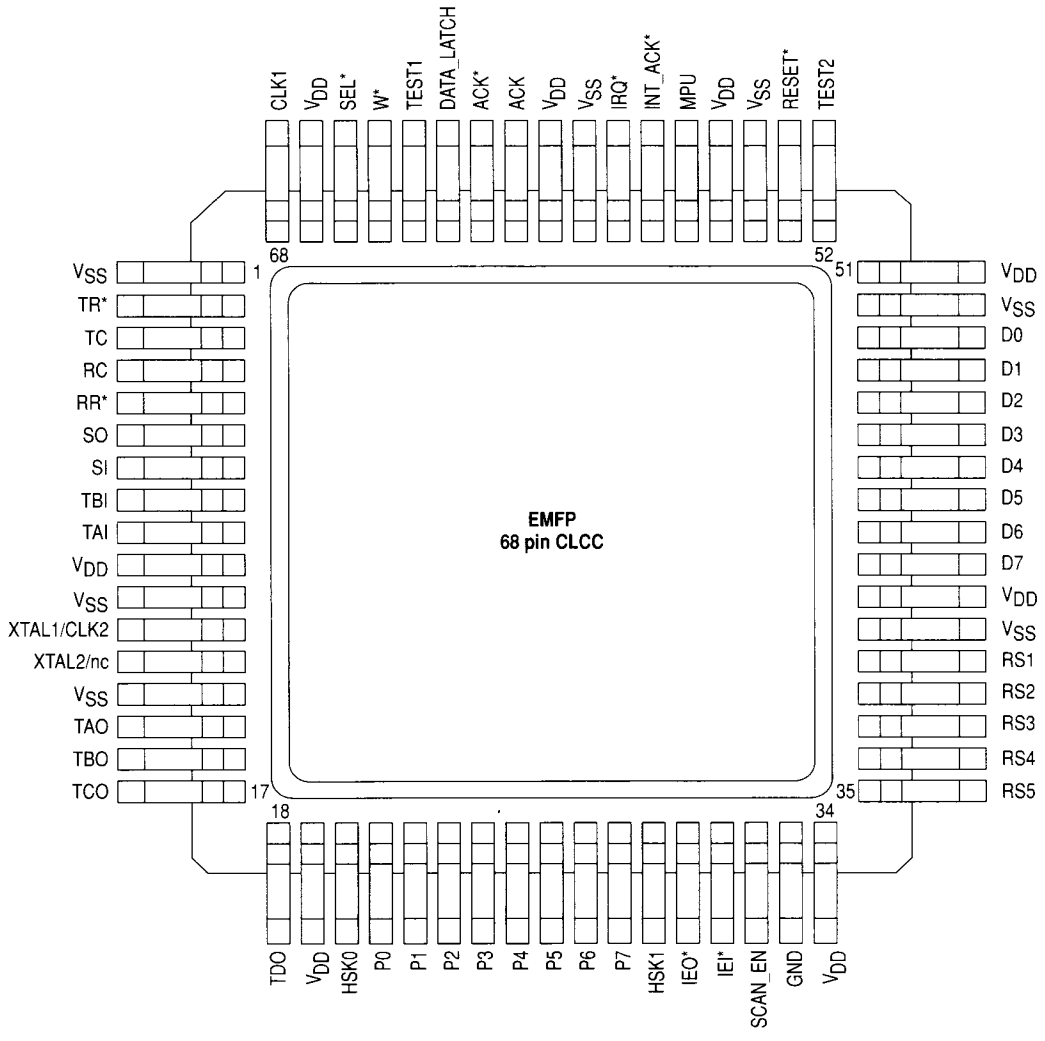
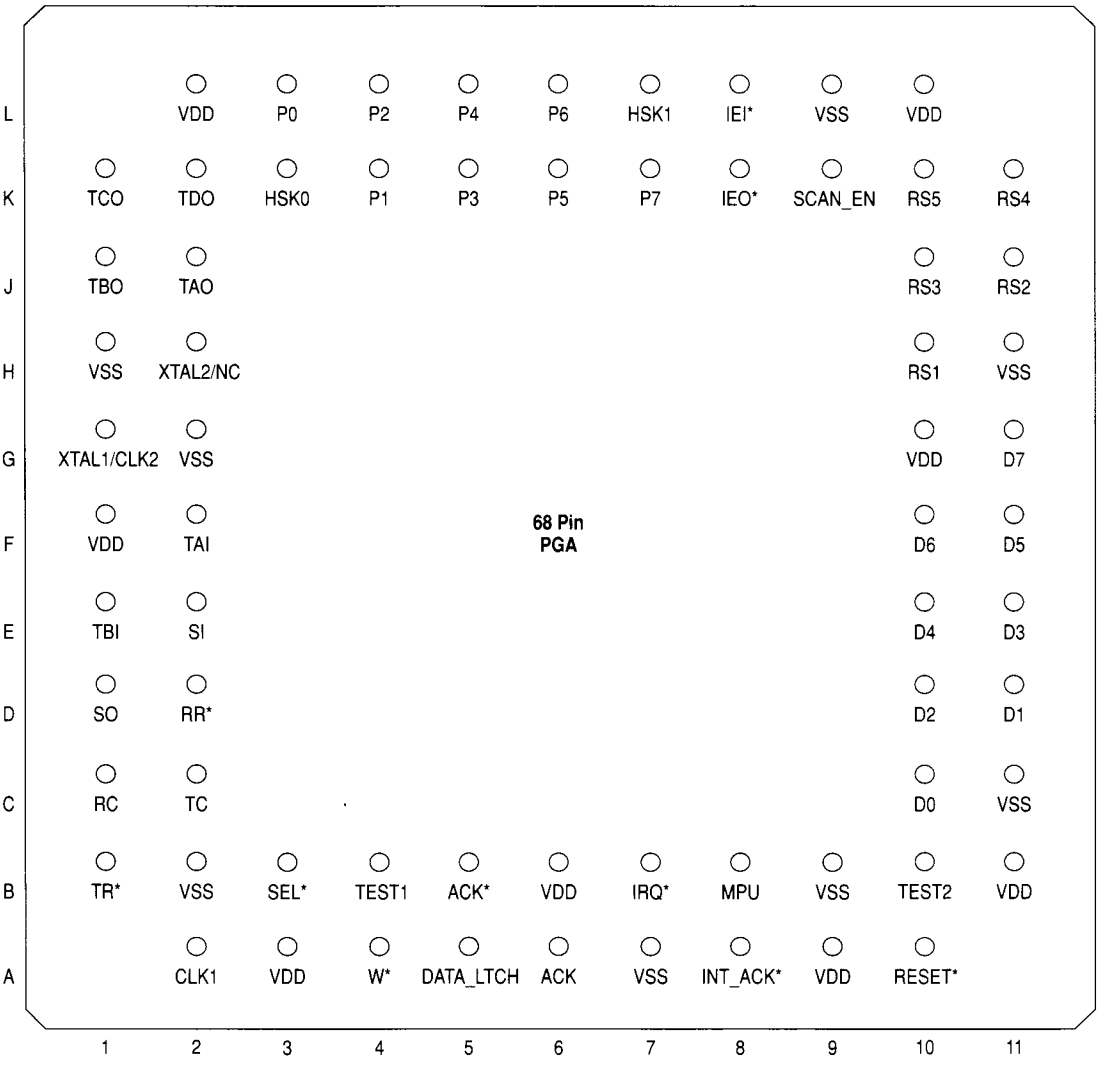


Figure 34. Pin Assignment

Packaging Information — Consult sales office.

Ordering Information — Consult sales office.

PACKAGING INFORMATION



Literature Distribution Centers:
USA: Motorola Literature Distribution; P.O. Box 20912; Phoenix, Arizona 85036.
EUROPE: Motorola Ltd.; European Literature Center; 88 Tanners Drive, Blakelands, Milton Keynes, MK14 5BP, England.
JAPAN: Nippon Motorola Ltd.; 4-32-1, Nishi-Gotanda, Shinagawa-ku, Tokyo 141 Japan.
ASIA-PACIFIC: Motorola Semiconductors H.K. Ltd.; Silicon Harbour Center, No. 2 Dai King Street, Tai Po Industrial Estate, Tai Po, N.T., Hong Kong.