SGS-THOMSON MICROELECTRONICS

MK68901

MULTI–FUNCTION PERIPHERAL

- 8 INPUT/OUTPUT PINS
  - Individually programmable direction
  - Individual interrupt source capability
    - Programmable edge selection
- 16 SOURCE INTERRUPT CONTROLLER
  - 8 Internal sources
  - 8 External sources
  - Individual source enable
  - Individual source masking
  - Programmable interrupt service modes
    - Polling
    - Vector generation
      - Optional In-service status
  - Daisy chaining capability
- FOUR TIMERS WITH INDIVIDUALLY PROGRAMMABLE PRESCALING
  - Two multimode timers
    - Delay mode
    - Pulse width measurement mode
    - Event counter mode
  - Two delay mode timers
  - Independent clock input
  - Time out output option
- SINGLE CHANNEL USART
  - Full Duplex
  - Asynchronous to 65 kbps
  - Byte synchronous to 1 Mbps
  - Internal/External baud rate generation
  - DMA handshake signals
  - Modem control
  - Loop back mode
- 68000 BUS COMPATIBLE
- 48 PIN DIP OR 52 PIN PLCC

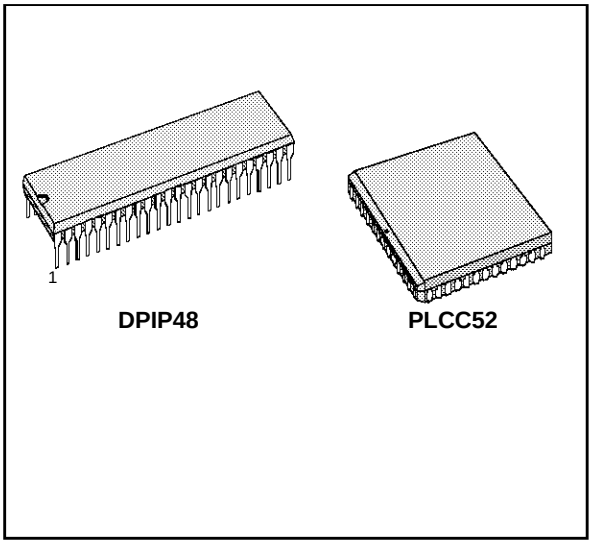
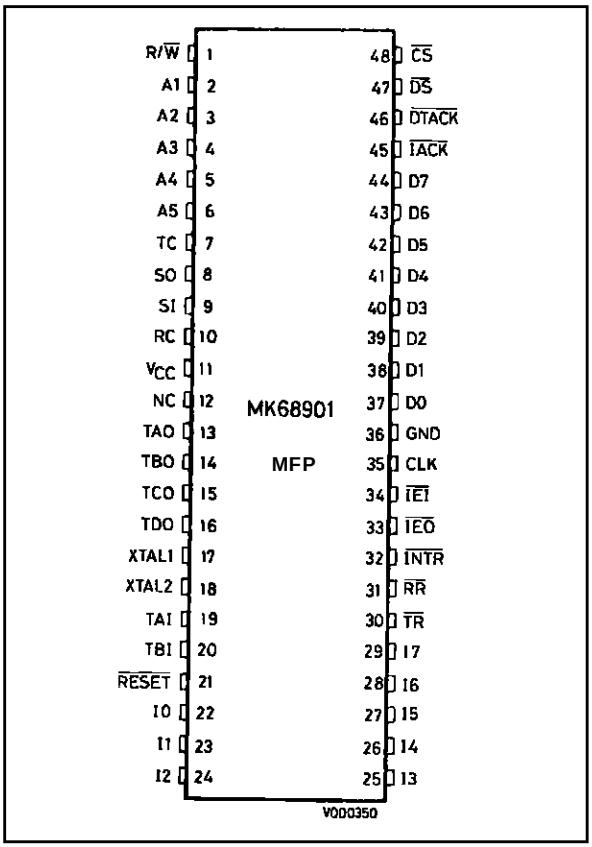


Figure 1 : Pin connections.



DESCRIPTION

The MK68901 MFP (Multi-Function Peripheral) is a combination of many of the necessary peripheral functions in a microprocessor system.
Included are :

Eight parallel I/O lines
Interrrupt controller for 16 sources
Four timers
Single channel full duplex USART

The use of the MFP in a system can significantly reduce chip count, thereby reducing system cost. The MFP is completely 68000 bus compatible, and 24 directly addressable internal registers provide the ne-

necessary control and status interface to the programmer.

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

PIN DESCRIPTION

GND : Ground

V_{CC} : +5 volts (± 5%)

$\overline{CS}$: Chip Select (input, active, low). $\overline{CS}$ is used to select the MK68901 MFP for accesses to the internal registers. $\overline{CS}$ and IACK must not be asserted at the same time.

$\overline{DS}$: Data Strobe (input, active low). $\overline{DS}$ is used as part of the chip select and interrupt acknowledge functions.

R/$\overline{W}$: Read/Write (input). R/$\overline{W}$ is the signal from the bus master indicating whether the current bus cycle is a Read (High) or Write (Low) cycle.

$\overline{DTACK}$: Data Transfer Acknowledge. (output, active low, tri-stateable) $\overline{DTACK}$ is used to signal the bus master that data is ready, or that data has been accepted by the MK68901 MFP.

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

D0-D7 : Data Bus (bi-directional, tri-stateable). The data bus is used to receive data from or transmit data to one of the internal registers during a read or write cycle. It is also used to pass a vector during an interrupt acknowledge cycle.

CLK : Clock (input). This input is used to provide the internal timing for the MK68901 MFP.

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

$\overline{INTR}$: Interrupt Request (output, active low, open drain). INTR is asserted when the MK68901 MFP is requesting an interrupt. $\overline{INTR}$ is negated during an interrupt acknowledge cycle or by clearing the pending interrupt(s) through software.

$\overline{IACK}$: Interrupt Acknowledge (input, active low). IACK is used to signal the MK68901 MFP that the CPU is acknowledging an interrupt. $\overline{CS}$ and $\overline{IACK}$ must not be asserted at the same time.

$\overline{IEI}$: Interrupt Enable In (input, active low). $\overline{IEI}$ is used to signal the MK68901 MFP that no higher priority device is requesting interrupt service.

$\overline{IEO}$: Interrupt Enable Out (output, active low). $\overline{IEO}$ is used to signal lower priority peripherals that neither the MK68901 MFP nor another higher priority peripheral is requesting interrupt service.

10-17 : General Purpose Interrupt I/O lines. These lines may be used as interrupt inputs and/or I/O lines. When used as interrupt inputs, their active edge is programmable. A data direction register is used to define which lines are to be Hi-Z inputs and which lines are to be push-pull TTL compatible outputs.

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

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

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

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

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

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

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

XTAL1, XTAL2 : Timer Clock inputs. A crystal can be connected between XTAL1 and XTAL2, or XTAL1 can be driven with a TTL level clock. When driving XTAL1 with a TTL le-

vel clock, XTAL2 must be allowed to float. When using a crystal, external capacitors are required. See figure 33. All chip accesses are independent of the timer clock.

TAI,TBI : Timer A, B inputs. Used when running the timers in the event count or the pulse width measurement mode. The interrupt channels associated with 14 and 13 are used for TAI and TBI, respectively. Thus, when running a timer in the pulse width

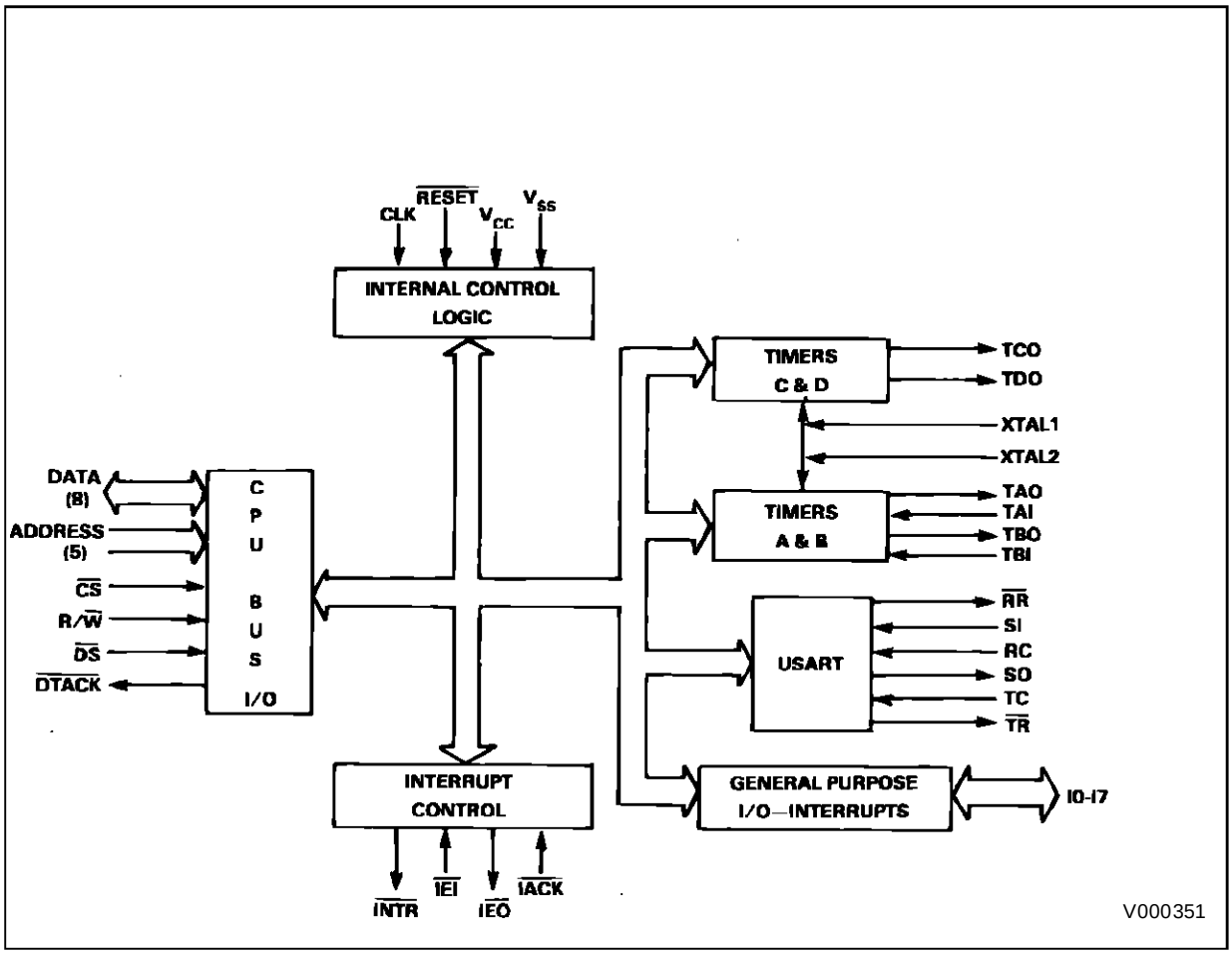


Figure 4 : Register Map.

Address Port N°.	Abbreviation	Register Name
0	GPIP	GENERAL PURPOSE I/O
1	AER	ACTIVE EDGE REGISTER
2	DDR	DATA DIRECTION REGISTER
3	IERA	INTERRUPT ENABLE REGISTER A
4	IERB	INTERRUPT ENABLE REGISTER B
5	IPRA	INTERRUPT PENDING REGISTER A
6	IPRB	INTERRUPT PENDING REGISTER B
7	ISRA	INTERRUPT IN-SERVICE REGISTER A
8	ISRB	INTERRUPT IN-SERVICE REGISTER B
9	IMRA	INTERRUPT MASK REGISTER A
A	IMRB	INTERRUPT MASK REGISTER B
B	VR	VECTOR REGISTER
C	TACR	TIMER A CONTROL REGISTER
D	TBCR	TIMER B CONTROL REGISTER
E	TCDCR	TIMERS C AND D CONTROL REGISTER
F	TADR	TIMER A DATA REGISTER
10	TBDR	TIMER B DATA REGISTER
11	TCDR	TIMER C DATA REGISTER
12	TDDR	TIMER D DATA REGISTER
13	SCR	SYNC CHARACTER REGISTER
14	UCR	USART CONTROL REGISTER
15	RSR	RECEIVER STATUS REGISTER
16	TSR	TRANSMITTER STATUS REGISTER
17	UDR	USART DATA REGISTER

INTERRUPTS

The General Purpose I/O-Interrupt Port (GPIP) provides eight I/O lines that may be operated either as inputs or outputs under software control. In addition, each line may generate an interrupt in either a positive going edge or a negative going edge of the input signal.

The GPIP has three associated registers. One allows the programmer to specify the Active Edge for each bit that will trigger an interrupt. Another register specifies the Data Direction (input or output) associated with each bit. The third register is the actual data I/O register used to input or output data to the port. These three registers are illstrated in figure 5.

The Active Edge Register (AER) allows each of the General Purpose Interrupts to provide an interrupt on either a 1-0 transition or a 0-1 transition. Writing a zero to the appropriate bit of the AER causes the associated input to produce an interrupt on the 1-0 transition. The edge bit is simply one input to an exclusive-or gate, with the other input coming from the input buffer ant the output going to a 1-0 transition detector. Thus, depending upon the state of the input, writing the AER can cause an interrupt-producing transition, which will cause an interrupt on the associated channel, if that channel is enabled. One would then normally configure the AER before enabling interrupts via IERA and IERB.

Note : Changing the edge bit, with the interrupt enabled, may cause an interrupt on that channel.

The Data Direction Register (DDR) is used to define 10-17 as inputs or as outputs on a bit by bit basis. Writing a zero into a bit of the DDR causes the corresponding Interrupt-I/O pin to be a Hi-Z input. Writing a one into a bit of the DDR causes the corresponding pin to be configured as a push-pull output. When data is written into the GPIP, those pins defined as inputs will remain in the Hi-Z state while those pins defined as outputs will assume the state (high or low) of their corresponding bit in the GPIP. When the GPIP is read, the data read will come directly from the corresponding bit of the GPIP register for all pins defined as output, while the data read on all pins defined as inputs will come from the input buffers.

Each individual function in the MK68901 is provided with a unique interrupt vector that is presented to the system during the interrupt acknowledge cycle. The interrupt vector returned during the interrupt acknowledge cycle is shown in figure 6, while the vector register is shown in figure 7.

There are 16 vector addresses generated internally by the MK68901, one for each of the 16 interrupt channels.

The Interrupt Control Registers (figure 8) provide control of interrupt processing for all I/O facilities of the MK68901. These registers allow the programmer to enable or disable any or all of the 16 interrupts, providing masking for any interrupt, and provide access to the pending and in-service status of the interrupt. Optional end-of-interrupt modes are available under software control. All the interrupts are prioritized as shown in figure 9.

Figure 5 : General Purpose I/O Registers.

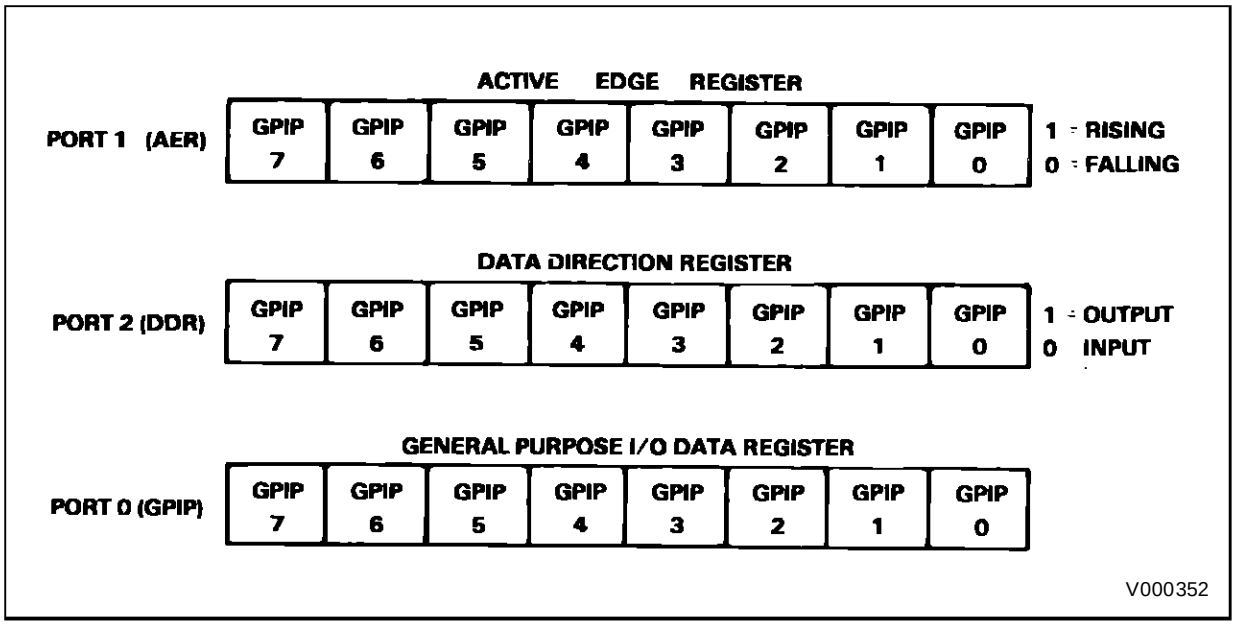


Figure 6 : Interrupt Vector.

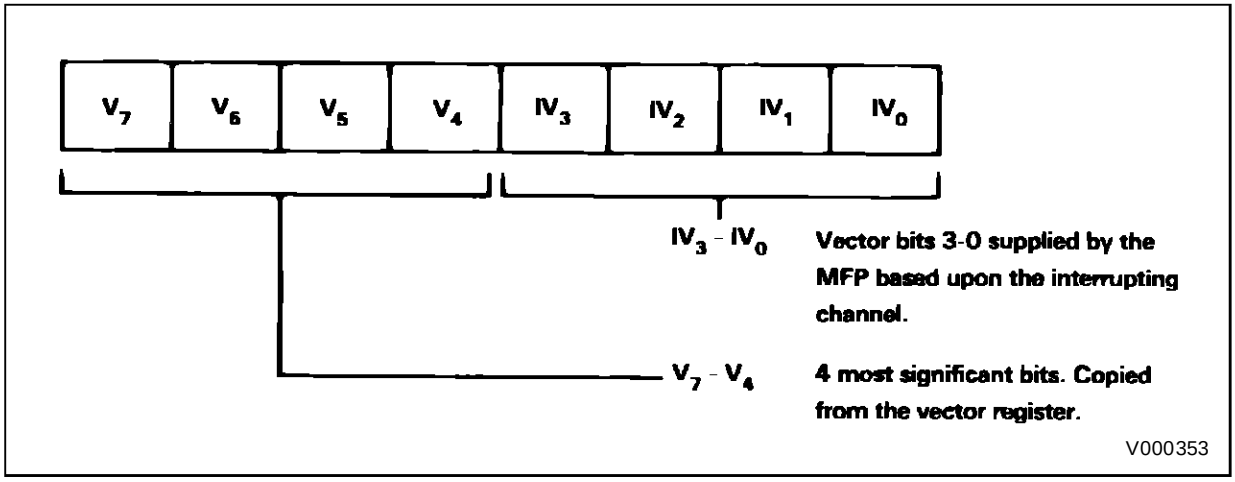


Figure 7 : Vector Register.

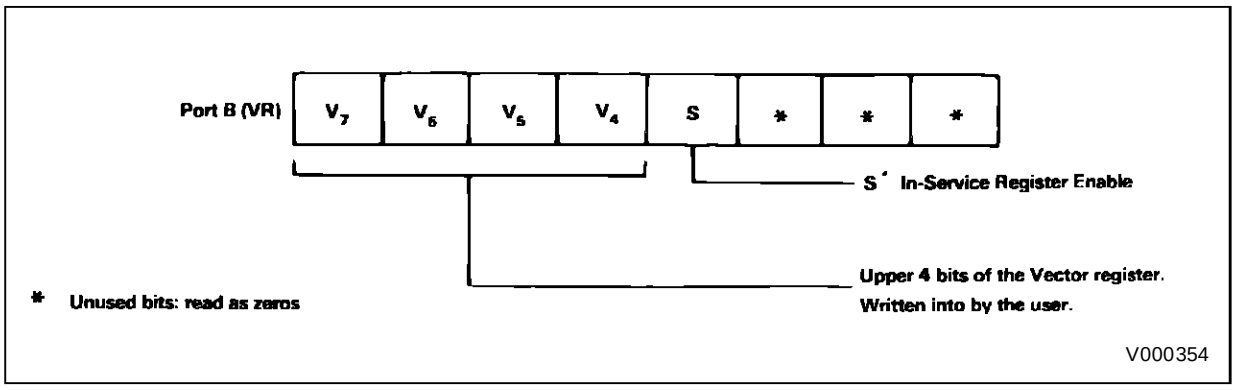


Figure 8 : Interrupt Control Registers.

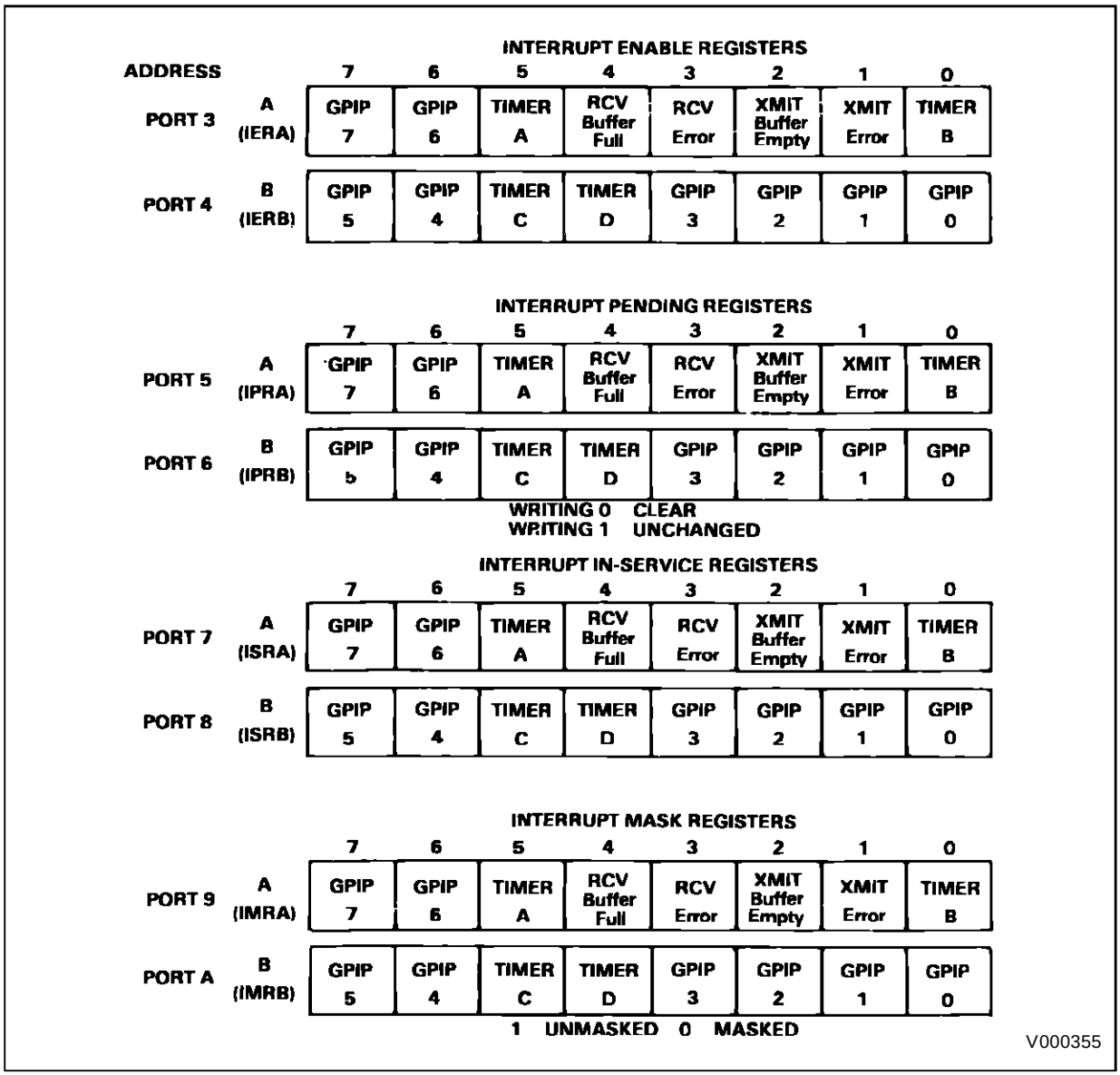


Figure 9 : Interrupt Control Register Definitions

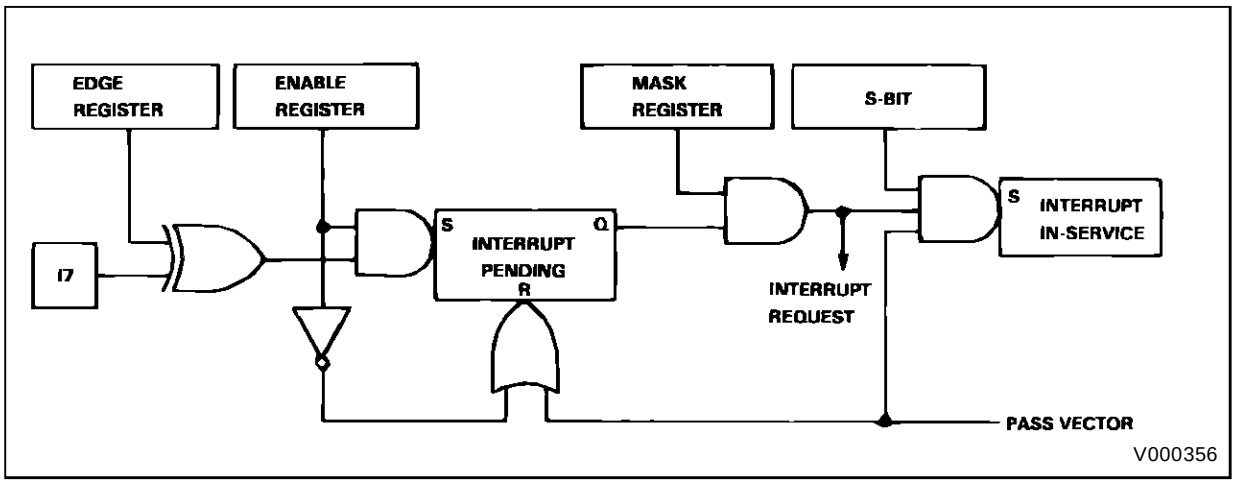
Priority	Channel	Description
HIGHEST	1111	General Purpose Interrupt 7(I7)
	1110	General Purpose Interrupt 6(I6)
	1101	Timer A
	1100	Receive Buffer Full
	1011	Receive Error
	1010	Transmit Buffer Empty
	1001	Transmit Error
	1000	Timer B
	0111	General Purpose Interrupt 5(I5)
	0110	General Purpose Interrupt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

Interrupts may be either polled or vectored. Each channel may be individual enabled or disabled by writing a one or a zero in the appropriate bit of Interrupt Enable Registers (IERA, IERB - see figure 8 for all registers in this section). When disabled, an interrupt channel is completely inactive. Any internal or external action which would normally produce an interrupt on that channel is ignored and any pending interrupt on that channel will be cleared by disabling that channel. Disabling an interrupt channel has no effect on the corresponding bit in Interrupt In-Service Registers (ISRA, ISRB) ; thus, if the In-service Registers are used and an interrupt is in service on that channel when the channel is disabled, it will remain in service until cleared in the normal manner. IERA and IERB are also readable.

When an interrupt is received on an enabled channel, its corresponding bit in the pending register will be set. When that channel is acknowledged it will pass its vector, and the corresponding bit in the Interrupt Pending Register (IPRA or IRPB) will be cleared. IPRA and IPRB are readable ; thus by polling IPRA and IPRB, it can be determined whether a channel has a pending interrupt. IPRA and IPRB are also writeable and a pending interrupt can be cleared without going through the acknowledge sequence by writing a zero to the appropriate bit. This allows any one bit to be cleared, without altering any other bits, simply by writing all ones except for the bit position to be cleared to IPRA or IPRB. Thus a fully polled interrupt scheme is possible. Note : writing a one to IPRA, IPRB has no effect on the interrupt pending register.

The interrupt mask registers (IMRA and IMRB) may be used to block a channel from making an interrupt request. Writing a zero into the corresponding bit of the mask register will still allow the channel to receive an interrupt and latch it into its pending bit (if that channel is enabled), but will prevent that channel from making an interrupt request. If that channel is causing an interrupt request at the time the corresponding bit in the mask register is cleared, the request will cease. If no other channel is making a request, INTR will go inactive. If the mask bit is re-enabled, any pending interrupt is now free to resume its request unless blocked by a higher priority request for service. IMRA and IMRB are also readable . A conceptual circuit of an interrupt channel is shown in figure 10.

Figure 10 : A Conceptual Circuit of an Interrupt Channel.



There are two end-of-interrupt modes : the automatic end-of-interrupt mode and the software end-of-interrupt mode. The mode is selected by writing a one or a zero to the S bit of the Vector Register (VR). If the S bit of the VR is a one, all channels operate in the software end-of-interrupt mode. If the S bit is a zero, all channels operate in the automatic end-of-interrupt mode, and a reset is held on all in-service bits. In the automatic end-of-interrupt mode, the pending bit is cleared when that channel passes its vector. At that point, no further history of that interrupt remains in the MK68901 MFP. In the software end-of-interrupt mode, the in-service bit is set and the pending bit is cleared when the channel passes its vector. With the in-service bit set, no lower priority channel is allowed to request an interrupt or to pass its vector during an acknowledge sequence ; however, a lower priority channel may still receive an interrupt and latch it into the pending bit. A higher priority channel may still request an interrupt and be acknowledged. The in-service bit of a particular channel may be cleared by writing a zero to the corresponding bit in ISRA or ISRB. Typically, this will be done at the conclusion of the interrupt routine just before the return. Thus no lower priority channel will be allowed to request service until the higher priority channel is complete, while channels of still higher priority will be allowed to request service. While the in-service bit is set, a second interrupt on that channel may be received and latched into the pending bit, though no service request will be made in response to the second interrupt until the in-service bit is cleared. ISRA and ISRB may be read at any time. Only a zero may be written into any bit of ISRA and ISRB ; thus the in-service bits may be cleared in software but cannot be set in software. This allows any one bit to be cleared, without altering any other bits, simply by writing all ones except for the bit position to be cleared to ISRA or ISRB, as with IPRA and IPRB.

Figure 11 a : A Conceptual Circuit of the MK68901 MFP Daisy Chaining.

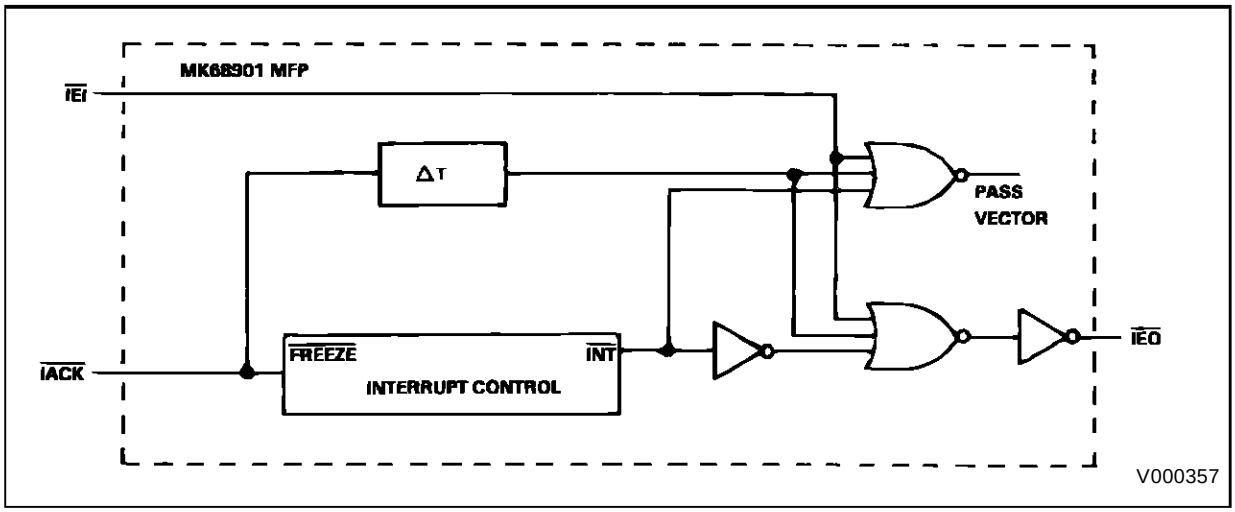
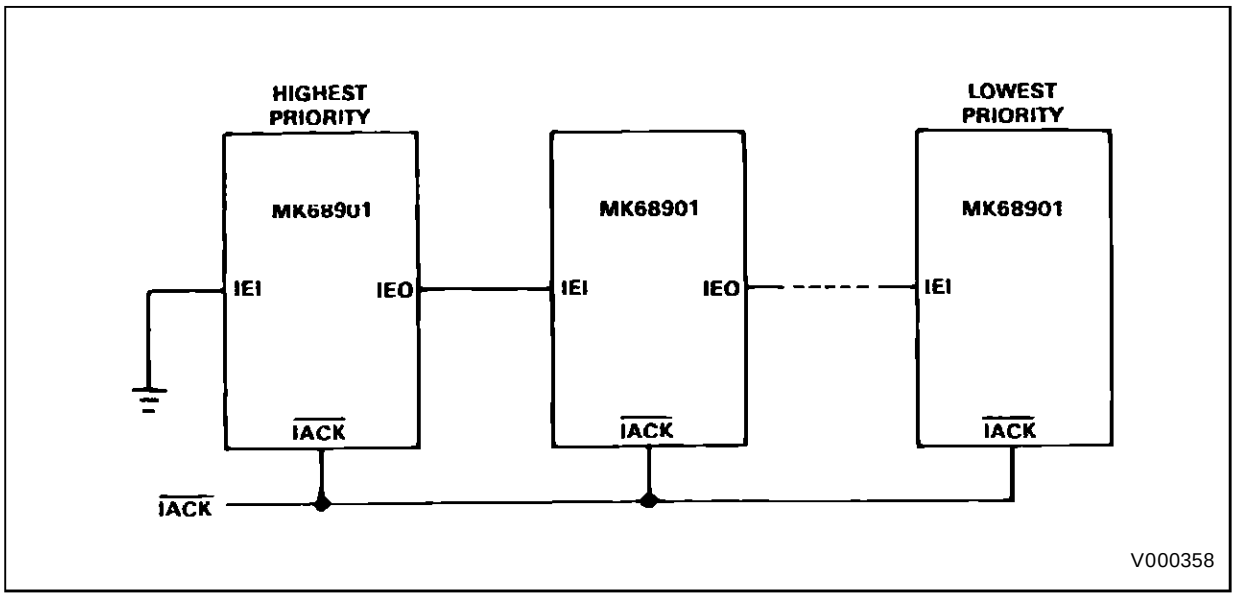


Figure 11 b : Daisy Chaining.



Each interrupt channel responds with a discrete 8-bit vector when acknowledged. The upper four bits of the vector are set by writing the upper four bits of the VR. The four low order bits (bit 3-bit 0) are generated by the interrupting channel.

To acknowledge an interrupt, IACK goes low, the IEI input must go low (or be tied low) and the MK68901 MFP must have an acknowledgeable interrupt pending. The Daisy Chaining capability (figure 11) requires that all parts in a chain have a common IACK. When the common IACK goes low, all parts freeze and prioritize interrupts in parallel. Then 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, does not propagate IEO, and generates DTACK.

Figure 9 describes the 16 prioritized interrupt channels. As chown, General Purpose Interrupt 7 has the highest priority, while General Purpose Interrupt 0 is assigned the lowest priority. Each of these channels may be reprioritized, in effect, by selectively masking interrupts under software control. The binary numbers under "channel" correspond to the modified bits IV3, IV2, IV1 and IV0, respectively, of the Interrupt Vector for each channel (see figure 6).

Each channel has an enable bit contained in IERA or IERB, a pending latch contained in IPRA or IPRB, a mask bit contained in IMRA or IMRB, and an in–service latch contained in ISRA or ISRB. Additionally, the eight General Purpose Interrupts each have an edge bit contained in the Active Edge Register (AER), a bit to define the line as input or output contained in the Data Direction Register (DDR) and an I/O bit in the General Purpose Interrupt-I/O Port (GPIP).

TIMERS

There are four timers on the MK68901 MFP. Two of the timers (Timer A and Timer B) are full function timers which can perform the basic delay function and can also perform event counting, pulse width measurement, and waveform generation. The other two timers (Timer C and Timer D) are delay timers only. One or both of these timers can be used to supply the baud rate clocks for the USART. All timers are prescaler/counter timers with a common independent clock input (XTAL1, XTAL2). In addition, all timers have a time-out output, function that toggles each time the timer times out.

The four timers are programmed via three Timer Control Registers and four Timer Data Registers. Timers A and B are controlled by the control registers TACR and TBCR, respectively (see figure 12), and by the data registers TADR and TBDR (figure 13). Timers C and D are controlled by the control register TCDCR (see figure 14) and two data registers TCDR and TDDR. Bits in the control registers allow the selection of operational mode, prescale, and control white the data registers are used to read the timer or write into the time constant register. Timer A and B input pins TAI and TBI, are used for the event and pulse width modes for timers A and B.

With the timer stopped, no counting can occur. The timer contents will remain unaltered while the timer is stopped (unless reloaded by writing the Timer Data Register), but any residual count in the prescaler will be lost.

Figure 12 : Timer A and B Control Registers.

Port C (TACR)	*	*	*	TIMER A RESET	AC_3	AC_2	AC_1	AC_0
Port D (TBCR)	*	*	*	TIMER B RESET	BC_3	BC_2	BC_1	BC_0

C_3	C_2	C_1	C_0	
0	0	0	0	Timer Stopped
0	0	0	1	Delay Mode. : 4 Prescale
0	0	1	0	Delay Mode. : 10 Prescale
0	0	1	1	Delay Mode. : 16 Prescale
0	1	0	0	Delay Mode. : 50 Prescale
0	1	0	1	Delay Mode. : 64 Prescale
0	1	1	0	Delay Mode. : 100 Prescale
0	1	1	1	Delay Mode. : 200 Prescale
1	0	0	0	Event Count Mode
1	0	0	1	Pulse Width Mode. : 4 Prescale
1	0	1	0	Pulse Width Mode. : 10 Prescale
1	0	1	1	Pulse Width Mode. : 16 Prescale
1	1	0	0	Pulse Width Mode. : 50 Prescale
1	1	0	1	Pulse Width Mode. : 64 Prescale
1	1	1	0	Pulse Width Mode. : 100 Prescale
1	1	1	1	Pulse Width Mode. : 200 Prescale

V000359

* Unused bits : read as zeros.

In the delay mode, the prescaler is always active. A count pulse will be applied to the main timer unit each time the prescribed number of timer clock cycles has elapsed. Thus, if the prescaler is programmed to divide by ten, a count pulse will be applied to the main counter every ten cycles of the timer clock.

Each time a count pulse is applied to the main counter, it will decrement its contents. The main counter is initially loaded by writing to the Timer Data Register. Each count pulse will cause the current count to decrement. When the timer has decremented down to "01" , the next count pulse will not cause it to decrement to "00". Instead, the next count pulse will cause the timer to be reloaded from the Timer Data Register. Additionally, a "Time out" pulse will be produced. This Time Out pulse is coupled to the timer interrupt channel, and, if that channel is enabled, an interrupt will be produced. The Time Out pulse is also coupled to the timer output pin and will cause the pin to change states. The output will remain in this new state until the next Time Out pulse occurs. Thus the output will complete one full cycle for each two Time Out pulses.

If, for example, the prescaler were programmed to divide by ten, and the Timer Data Register were loaded with 100 (decimal), the main counter would decrement once for every ten cycles of the timer clock. A Time Out pulse will occur (hence an interrupt if that channel is enabled) every 1000 cycles of the timer clock, and the timer output will complete one full cycle every 2000 cycles of the timer clock.

The main counter is an 8-bit binary down counter. It may be read at any time by reading the Timer Data Register. The information read is the information last clocked into the timer read register when the DS pin had last gone high prior to the current read cycle. When written, data is loaded into the Timer Data Register, and the main counter, if the timer is stopped. If the Timer Data Register is written while the timer is running, the new word is not loaded into the timer until it counts through H"01". However, if the timer is written while it is counting through H"01", an indeterminate value will be written into the timer constant register. This may be circumvented by ensuring that the data register is not written when the count is H"01".

If the main counter is loaded with "01", a Time Out Pulse will occur every time the prescaler presents a count pulse to the main counter. If loaded with "00", a Time Out pulse will occur every 256 count pulses.

Figure 13 : Timer Data Registers (A, B, C, and D).

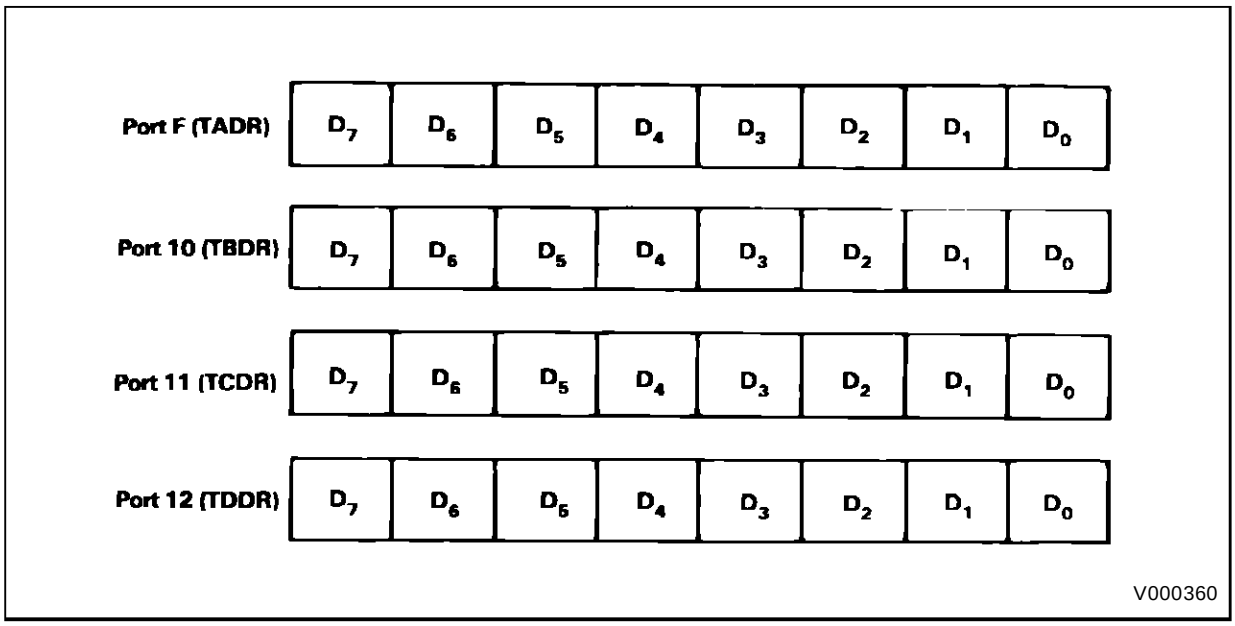
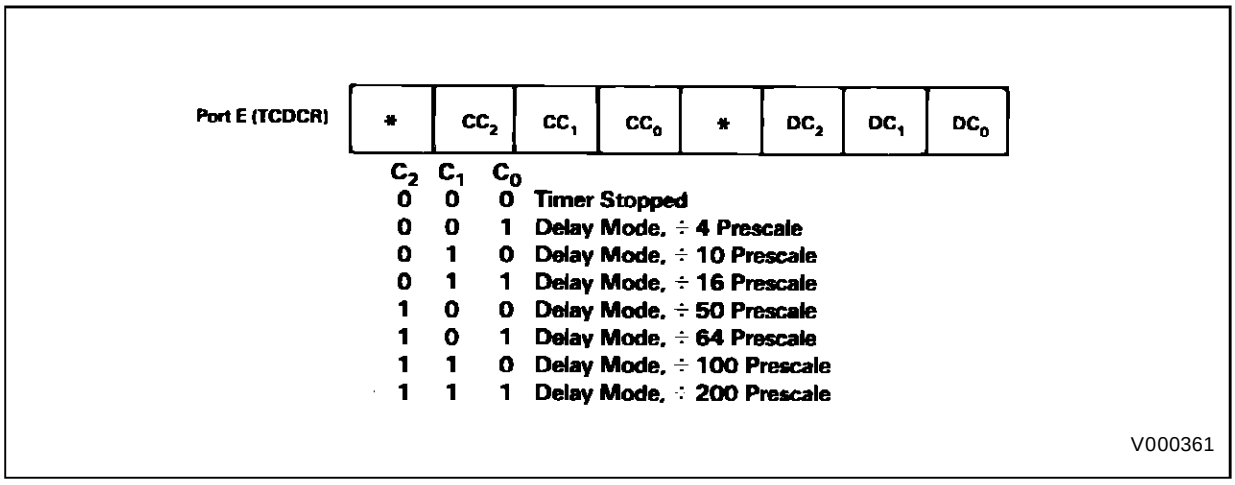
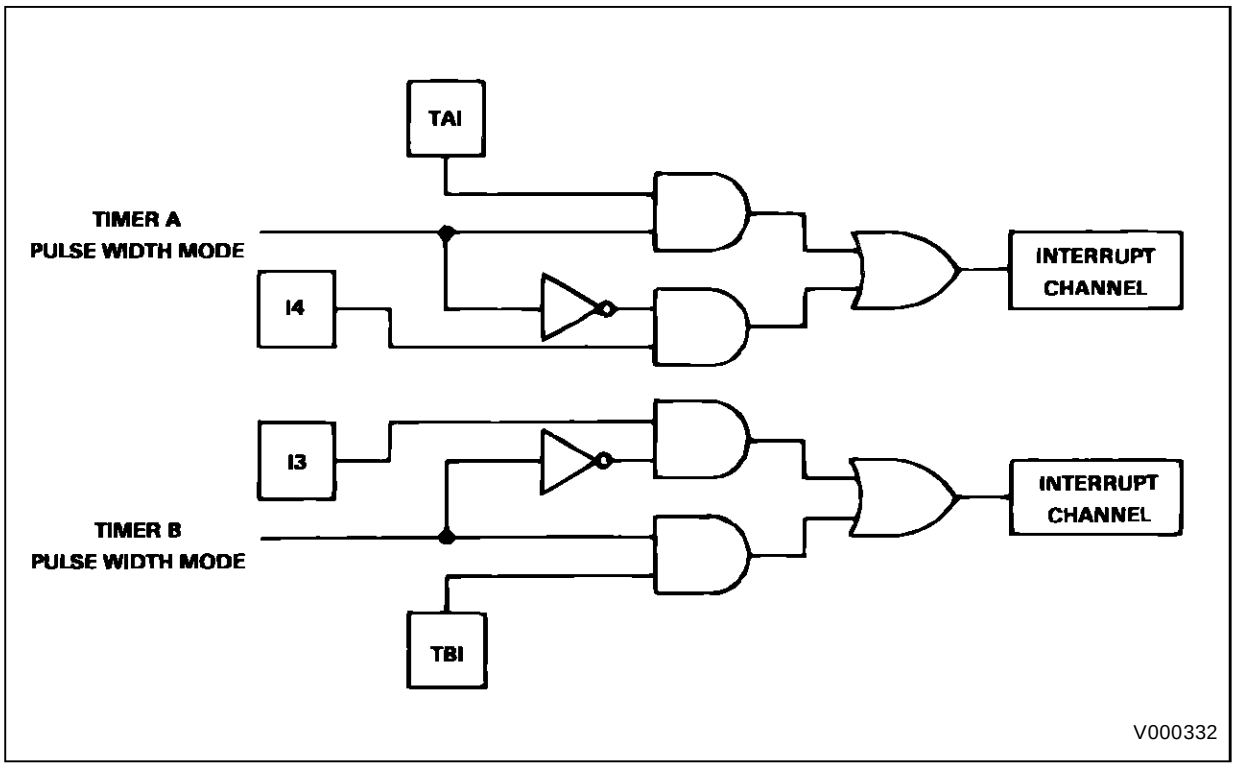


Figure 14 : Timer C and D Register.



* Unused bits : read as zeros.

Figure 15 : A Conceptual Circuit of the MFP Timers in the Pulse Width Measurement Mode.



Changing the prescale value with the timer running can cause the first Time Out pulse to occur at an indeterminate time, (no less than one nor more than 200 timer clock cycles times the number in the time constant register), but subsequent Time Out pulses will then occur at the correct interval.

In addition to the delay mode described above, Timers A and B can also function in the Pulse Width Measurement mode or in the Event Count mode. In either of these two modes, an auxiliary control signal is required. The auxiliary control input for Timer A is TAI, and for Timer B, TBI is used. The interrupt channels associated with 14 and 13 are used for TAI and TBI, respectively, in Pulse Width mode. See Figure 15.

The pulse width measurement mode functions much like the delay mode. However, in this mode, the auxiliary control signal on TAI or TBI acts as an enable to the timer. When the control signal on TAI or TBI is inactive, the timer will be stopped. When it is active, the prescaler and main counter are allowed to run. Thus the width of the active pulse on TAI or TBI is determined by the number of timer counts which occur while the pulse allows the timer to run. The active state of the signal on TAI or TBI is dependent upon the associated Interrupt Channel's edge bit (GPIP 4 for TAI and GPIP 3 for TBI : see Active Edge Register in figure 5). If the edge bit associated with the TAI or TBI input is a one, it will be active high ; thus the timer will be allowed to run when the input is at a high level. If the edge bit is a zero, the TAI or TBI input will be active low. As previously stated, the interrupt channel (13 or 14) associated with the input still functions when the timer is used in the pulse width measurement mode. However, if the timer is programmed for the pulse width measurement mode, the interrupt caused by tran-sitions on the associated TAI or TBI input will occur on the opposite transition.

For example, if the edge bit associated with the TAI input (AER-GPIP 4) is as one, an interrupt would normally be generated on the 0-1 transition of the 14 input signal. If the timer associated with this input (Timer A) is placed in the pulse width measurement mode, the interrupt will occur on the 1-0 tran-sition of the TAI signal instead. Because the edge bit (AER-GPIP 4) is a one, Timer A will be allowed to count while the input is high. When the TAI input makes the high to low transition, Timer A will stop, and it is at this point that the interrupt will occur (assuming that the channel is enabled). This allows the interrupt to signal the CPU that the pulse being measured has terminated ; thus Timer A may now be read to determine the pulse width. (Again note that 13 and 14 may still be used for I/O when the timer is in the pulse width measurement mode). If Timer

A is reprogrammed for another mode, interrupts will again occur on the transition, as normally defined by the edge bit. Note that, like changing the edge bit, placing the timer into or taking it out of the pulse width mode can produce a transition on the signal to the interrupt channel and may cause an interrupt. If measuring consecutive pulses, it is obvious that one must read the contents of the timer and then reinitialize the main counter by writing to the timer data register. If the timer data register is written while the pulse is going to the active state, the write operation may result in an indeterminate value being written into the main counter. If the timer is written after the pulse goes active, the timer counts from the previous contents, and when it counts through H"01", the correct value is written into the timer. The pulse width then includes counts from before the timer was reloaded.

In the event count mode, the prescaler is disabled. Each time the control input on TAI or TBi makes an active transition as defined by the associated Interrupt Channel's edge bit, a count pulse will be generated, and the main counter will decrement. In all other respects, the timer functions as previously described. Altering the edge bit while the timer is in the event count mode can produce a count pulse. The interrupt channel associated with the input (I3 for I4 for TAI) is allowed to function normally. To count transitions reliably, the input must remain in each state (1/O) for a length of time equal to four periods of the timer clock ; thus signals of a frequency up to one fourth of the timer clock can be counted.

The manner in which the timer output pins toggle states has previously been described. All timer outputs will be forced low by a device RESET. The output associated with Timers A and B will toggle on each Time Out pulse regardless of the mode the timers are programmed to. In addition, the outputs from Timers A and B can be forced low at any time by writing a "1" to the reset location in TACR and TBCR, respectively. The output will be forced to the low state during the WRITE operation, and at the conclusion of the operation, the output will again be free to toggle each time a Time Out pulse occurs. This feature will allow waveform generation.

During reset, the Timer Data Registers and the main counters are not reset. Also, if using the reset option on Timers A or B, one must make sure to keep the other bits in the correct state so as not to affect the operation of Timers A and B.

USART

Serial Communication is provided by a full-duplex double-buffered USART, which is capable of either asynchronous or synchronous operation. Variable word length and start/stop bit configurations are available under software control for asynchronous operation. For synchronous operation, a Sync Word is provided to establish synchronization during receive operations. The Sync Word will also be repeatedly transmitted when no other data is available for transmission. Moreover, the MK68901 allows stripping of all Sync Words received in synchronous operation. The handshake control lines RR (Receiver Ready) and TR (Transmitter Ready) allow DMA operation. Separate receive and transmit clocks are available, and separate receive and transmit status and data bytes allow independent operation of the transmit and receive sections.

The USART is provided with three Control/Status Registers and a Data Register. The USART Data Register form is illustrated in figure 16. The programmer may specify operational parameters for the USART via the Control Register, as shown in figure 17. Status of both the Receiver and Transmitter sections is accessed by means of the two Status Registers, as shown in figures 18 and 19. Data written to the Data Register is passed to the transmitter, while reading the Data Register will access data received by the USART.

Figure 16 : USART Data Register.

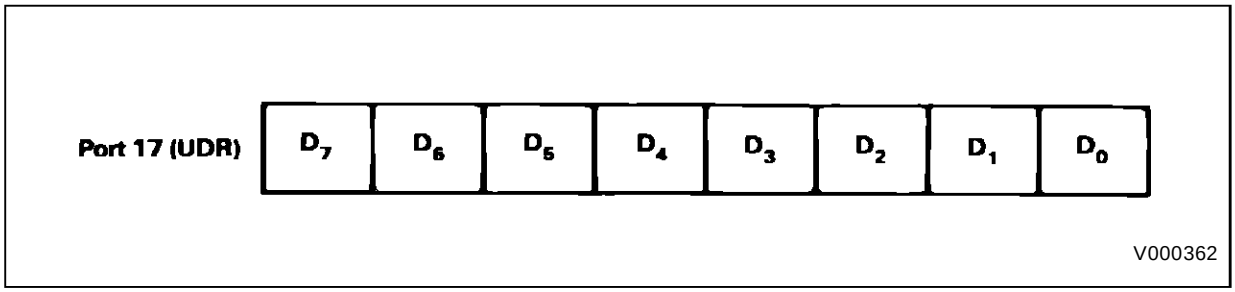
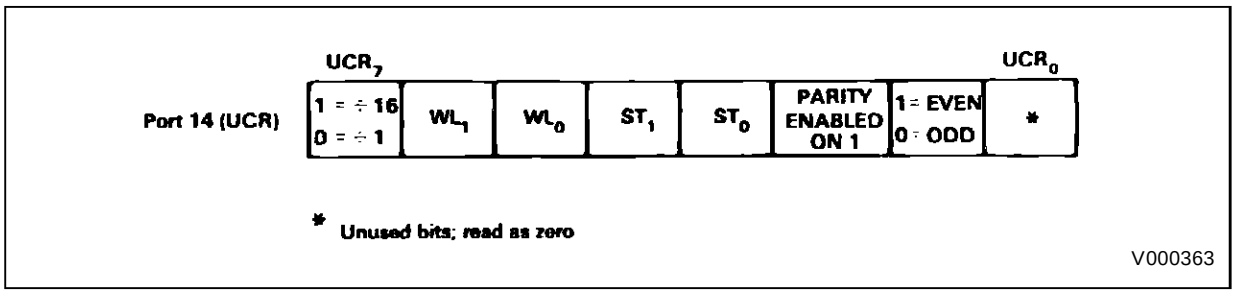


Figure 17 : USART Control Register (UCR).



÷ 16/÷ 1 : When this bit is zero, data will be clocked into and out of the receiver and transmitter at the frequency of their respective clocks. When this bit is loaded with a one, data will be clocked into and out of the receiver and transmitter at one sixteenth the frequency of their respective clocks. Additionally, when placed in the divide by sixteen mode, the receiver data transition resynchronization logic will be enabled.

WL0-WL1 :Word Length Control. These two bits set the length of the data word (exclusive of start bits, stop bits, and parity bits as follows:

WL1	WL0	Word Length
0	0	8 Bits
0	1	7 Bits
1	0	6 Bits
1	1	5 Bits

ST0-ST1 : Start/stop bit control (format control). These two bits set the format as follows

ST1	ST0	Start Bits	Stop Bits	Format
0	0	0	0	SYNC
0	1	1	1	ASYNC
†1	0	1	$1^1/_2$	ASYNC
1	1	1	2	ASYNC

:

PARITY : Parity Enabled. When set (”1”), parity will be checked by the receiver, parity will be calculated, and a parity bit will be inserted by the transmitter. When cleared (”0”) no parity check will be made and no parity bit will be inserted for transmission. For a word length of 8 the MFP calculates the parity and appends it when transmitting a sync character. For shorter lengths, the parity must be stored in the Sync Character Register (SCR) along with the sync character.

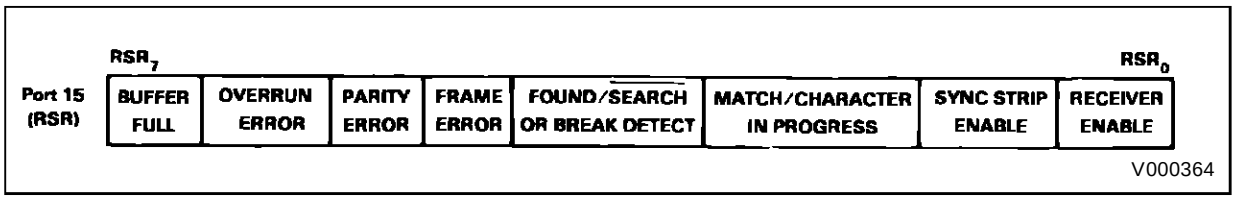
E/O : Even-Odd. When set (”1”), even parity will be used if parity is enabled. When cleared (”0”), odd parity will be used if parity is enabled.

Note that the synchronous or asynchronous format may be selected independently of a ÷ 1 or ÷ 16 clock. Thus it is possible to clock data synchronously into the device but still use start and stop bits. In this mode, all normal asynchronous format features still apply. Data will be shifted in after a start bit is encountered, and a stop bit will be checked to determine proper framing. If a transmit underrun condition occurs, the output will be placed in a marking state, etc. It is conversely possible to clock data in asynchronously using a synchronous format. There is data transition detection logic built into the receive clock circuitry which will re-synchronize the internal shift clock on each data transition so that, with sufficienty frequent data transitions, start bits are not required. In this mode, all other common synchronous features function normally. This re-synchronization logic is only active in ÷ 16 clock mode.

RECEIVER

The receiver section of the USART is configured by the UCR as previously described. The status of the receiver can be determined by reading and writing to the Receiver Status Register (RSR). The RSR is configured as follows :

Figure 18 : Receiver Status Register (RSR).



BF : Buffer Full. This bit is set when the incoming word is transferred to the receive buffer. The bit is cleared when the receive buffer is read by reading the UDR. This bit of the RSR is read only.

OE : Overrun Error. This flag is set if the incoming word is completely received and due to be tranferred to the receive buf-fer, but the last word in the receive buf-fer has not yet been read. When this condition occurs, the word in the receive buffer is not overwritten by the new word. Note that the status flags always reflect the status of the data word currently in the receive buffer. As such, the OE flag is not actually set until the good word currently in the buffer has been read. The interrupt associated with this error will also not be generated unti the old word in the receive buffer has been read.

OE flag is cleared by reading the receiver status register, and new data words cannot be shifted to the receive buffer until this is done.

PE : Parity Error. This flag is set if the word received has a parity error. The flag is set when the received word is tranferred from the shift register to the receive buffer if the error condition exists. The flag is cleared when the next word which does not have a parity error is tranferred to the receive buffer.

FE : Frame Error. This flag only applies to the asynchronous format. A frame error is defined as a non-zero data word which is not followed by a stop bit. Like the PE flag, the FE flag is set or cleared when a word is transferred to the receive buffer.

F/S : Found/$\overline{\text{Search}}$. This combination control bit and flag bit is only used with the synchronous format. It can be set or cleared by writing to this bit of the RSR. When this bit is cleared, the receiver is placed in the search mode. In this mode, a bit by bit comparison of the incoming data to the character in the Sync Character Register (SCR) is made. The word length counter is disabled. When a match is found, this bit will be set automatically, and the word length counter will start as sync has not been achieved. An interrupt will be generated on the receive error channel when the match occurs. The word just shifted in will, or necessity, be equal to the sync character, and it will not be transferred to the receive buffer.

B : Break. This flag is used only when the asynchronous format is selected. This flag will be set when an all zero data word, followed by no stop bit, is received. The flag will stay set until both a non-zero bit is received and the RSR has been read at least once since the flag was set. Break indication will not occur if the receive buffer is full.

M/CIP : Match/Character in Progress. If the synchronous format is selected, this flag is the Match flag. It will be set each time the word transferred to the receive buffer matches the sync character. It will be reset each time the word transferred to the receive buffer does not match the sync character. If the asynchronous format is selected, this flag represents Character in Progress. It will be set upon a start bit detect and cleared at the end of the word.

SS : Sync Strip Enable. If this bit is set to a one, data words that match the sync character will not be loaded into the receive buffer, and no buffer full signal will be generated.

RE : Receiver Enable. This control bit is used to enable or disable the receiver. If a zero is written to this bit of the RSR, the receiver will turn off immediately. All flags including the F/S bit will be cleared. If a one is written to this bit, normal receiver operation is enabled. The receive clock has to be running before the receiver is enabled.

There are two interrupt channels associated with the receiver. One channel is used for the normal Buffer Full condition, while the other channel is used whenever an error condition occurs. Only one interrupt is generated per word received, but dedicating two channels allows separate vectors : one for the normal condition, and one for an error condition. If the error channel is disabled, an interrupt will be gen-erated via the Butter Full Channel, whether the word received is normal or in error. Those conditions which produce an interrupt via the error channel are : Overrun, Parity Error, Frame Error, Sync Found, and Break. If a received word has an error associated with it, and the error interrupt channel is enabled, an interrupt will occur on the error channel only.

Each time a word is transferred into the receive buffer, a corresponding set of flags is latched into the RSR. No flags (except CIP) are allowed to change until the data word has been read from the receive buffer. Reading the receive buffer allows a new data word to be transferred to the receive buffer when it is received. Thus one should first read the RSR then read the receive buffer (UDR) to ensure that the flags just read match the data word just read. If done in the reverse order, it is possible that subsequent to reading the data word from the receive buffer, but prior to reading the RSR, a new word may be received and transferred to the receive buffer and, with it, its associated flags latched into the RSR. Thus, when the RSR is read, those flags may actually correspond to a different data word. It is good practice, also to read the RSR prior to a data read as, when an overrun error occurs, the receiver will not assemble new characters until the RSR has been read.

As previously stated, when overrun occurs, the OE flag will not be set and the associated interrupt will not be generated until the receive buffer has been read. If a break occurs, and the receive buffer has not yet been read, only the B flag will be set (OE will not be set). Again, this flag will not be set until the last valid word has been read from the receive buffer. If the break condition ends and another whole data word is received before the receive buffer is read, both the B and OE flags will be set once the receive buffer is read.

If a break occurs while the OE flag is set, the B flag will also be set.

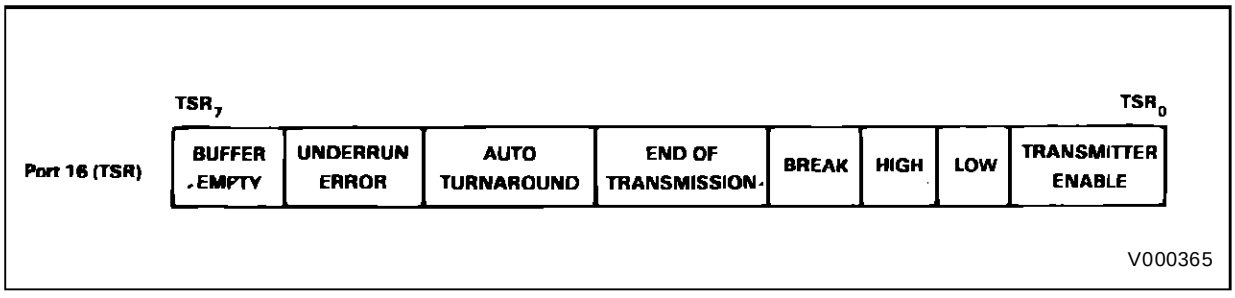
A break generates an interrupt when the condition occurs and again when the condition ends. If the break condition ends before it is acknowledged by reading the RSR, the receiver error interrupt indicating end of break will be generated once the RSR is read.

Anytime the asynchronous format is selected, start bit detection is enabled. New data is not shifted into the shift register until a zero bit is detected. If a ÷ 16 clock is selected, along with the asynchronous format, false start bit detection is also enabled. Any transition has to be stable for 3 positive going edges of the receive clock to be called a valid transition. For a start bit to be good, a valid 0-1 transition must not occur for 8 positive clock transitions after the initial valid 1-0 transition.

After a good start bit has been detected, valid transitions in the data are checked for continously. When a valid transition is detected, the counter is forced to state zero, and no more transition checking is started 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 stream. This logic also makes the unit more tolerant of clock skew for normal asynchronous communications than a device which employs only start bit synchronization.

Figure 19 : Transmitter Status Register (TSR).



TRANSMITTER

The transmitter section of the USART is configured as to format, word length, etc. by the UCR, as previously described. The status of the transmitter can be determined by reading or writing the Transmitter Status Register (TSR). The TRS is configured as follows :

BE : Buffer Empty. This status bit is set when the word in the transmit buffer is transferred to the output shift register and thus the transmit buffer may be reloaded with the next data word. The flag is cleared when the transmit buffer is reloaded. The transmit buffer is loaded by writing to the UDR.

UE : This bit is set when the last word has been shifted out of the transmit shift register before a new word has been loaded into the transmit buffer. It is not necessary to clear this bit before loading the UDR.

This bit may be cleared by either reading the TSR or by disabling the transmitter. After the setting of the UE bit, one full transmitter clock cycle is required before this bit can be cleared by a read. The timing in some systems may allow a read of the TSR before the required clock cycle has been completed. This would result in the UE bit not being cleared until the following read. To avoid this problem, a dummy read of the TSR should be performed at the end of he UE service routine.

Only one underrun error may be generated between loads of the UDR regardless of the number of transmitter clock cycles between UDR loads.

AT : This bit causes the receiver to be enabled at the end of the transmission of the last word in the transmitter if the transmitter has been disabled.

END : End or Transmission. When the transmitter is turned off with a character still in the output shift register, transmission will continue until that character is shifted out. Once it has cleared the output register, the END bit will be set. If no character is being transmitted when the transmitter is disabled, the transmitter will stop at the next rising edge of the internal shift clock, and END will immediately be set. The END bit is cleared by re-enabling the transmitter.

B : Break. This control bit will cause a break to be transmitted. When a ”1” is written to the B bit of the TSR, a break will be transmitted upon completion of the character (if any) currently being transmitted. A break will continue to be transmitted until the B bit is cleared by writing a ”0” tot his bit of the TSR. At that time, normal transmission will resume. The B bit has no function in the synchronous format. Setting the ”B” bit to a one keeps the ”BE” bit from being set to a one. So, if there were a word in the buffer at the start of break, it would remain there until the end of break, at which time it would be transmitted (if the transmitter is still enabled). If the buffer were not full at the start of break, it could be written at any time during the break. If the buffer is empty at the end of break, the underrun flag will be set (unless the transmitter is disabled).

The BREAK bit cannot be set until the transmitter has been enabled and the transmitter has had sufficient time (one clock cycle) to perform the internal reset and initialization functions.

H,L : High and Low. These two control bits are used to configure the transmitter output, when the transmitter is disabled, as follows :

H L Output State

0 0 Hi-Z

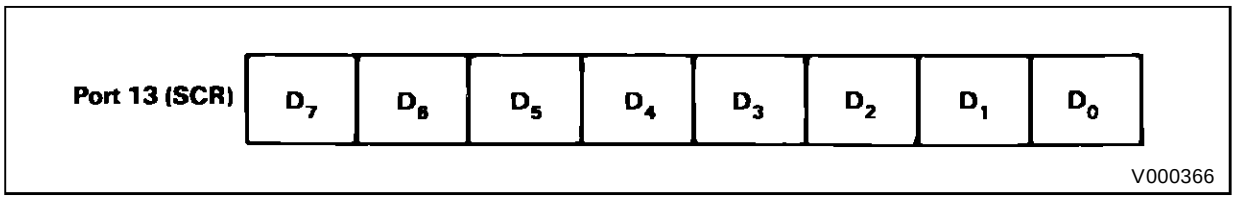
0 1 Low (”0”)

1 0 High

1 1 Loop-Connects transmitter output to receiver input, and TC to Receiver Clock (RC and SI are not used ; they are bypassed internally). In loop back mode, transmitter output goes high when disabled.

Altering these two bits after Transmitter

Figure 20 : SYNC Character Register.



Enable (XE) is set will alter the output state until END is false. These bits should be set prior to enabling the transmitter. The state of these bits determine the state of the first transmitted character after the transmitter is enabled. If the high impedance mode was selected prior to the transmitter being enabled, the first bit transmitted is indeterminate.

XE : Transmitter Enable. This control bit is used to enable or disable the transmitter. When set, the transmitter is enabled. When cleared, the transmitter will be disabled. If disabled, any word currently in the output register will continue to be transmitted when XE is cleared, the transmitter will turn off at the end of the break character boundary, and no end of break stop bit is transmitted. The transmit clock must be running before the transmitter is enabled A "one" bit always precedes the first word out of the transmitter after the transmitter is enabled. There is a delay between the time the transmitter enable bit is written an when the transmitter reset goes low ; therefore, the H & L bits should be written with the desired state prior to enabling the transmitter.

Like the receiver section, there are two separate interrupt channels associated with the transmitter. The buffer Empty condition causes an interrupt via one channel, while the Underrun and END conditions will cause an interrupt via the second channel. When underrun occurs in the synchronous format, the character in the SCR will be transmitted until a new words is loaded into the transmit buffer. In the asynchronous format, a "Mark" will be continuously transmitted when underrun occurs.

The transmit buffer can be loaded prior to enabling the transmitter. When the transmitter is disabled, any character currently in the process of being transmitted will continue to conclusion, but any character in the transmit buffer will not be transmitted and will remain in the buffer. Thus no buffer empty interrupt will occur nor will the BE flag be ste. If the buffer were already empty, the BE flag would be set and would remain set. When the transmitter is disabled with a character in the output register but with no character in the transmit buffer, an Underrun Error will not occur when the character in progress concludes.

Often it is necessary to send a break for some particular period. To aid in timing a break transmission, a transmission, a transmit error interrupt will be generated at every normal character boundary time during a break transmission. The status register information is unaffected by this error condition interrupt. It should be noted that an underrun error, if present, must be cleared from the TSR, and the interrupt pending register must be cleared of pending transmitter errors at the beginning of the break transmission or no interrupts will be generated at the character boundary time.

It the synchronous format is selected, the sync character should be loaded into the Sync Character Register (SCR) as shouwn in figure 20. This character is compared to the received serial data during a Search, and will be continuously transmitted during an underrun condition.

All flags in the RSR or TSR will continue to function as described whether their associated interrupt channel is disabled or enabled. All interrupt channels are edge triggered and, in many cases, it is the actual output of a flag bit or flag bits which is coupled to the interrupt channel. thus, if a normal interrupt producing condition occurs while the interrupt channel is disabled, no interrupt would be produced even if the channel was subsequently enabled, because a transition did not occur while the interrupt channel was enabled. that particular flag bit would have to occur a second time before another "edge" was produced, causing an interrupt to be generated.

Error conditions in the USART are determined by monitoring the Receive Status Register and the Transmitter Status Register. These error conditions are only valid for each word boundary and are not latched. When executing block tranfers or data, it is necessary to save any errors so that they can be checked at the end of a block. In order to save error conditions during data transfer, the MK68901 MFP interrupt controller may be used by enabling error interrupt for the desired channel (Receive error or Transmit error) and by masking these bits off. Once the tranfer is complete, the Interrupt Pending Register can be polled, to determine the precence of a pending error interrupt, and therefore an error.

Unused bits in the sync character register are zeroed out ; therefore, word length should be set up prior to writing the sync word in some cases. Sync word length is the word length plus one when parity is enabled. The user has to determine the parity of the sync word when the word length is not 8 bits. The MK68901 MFP does not add a parity bit to the sync word if the word length is less than 8 bits. The extra bit in the sync word is transmitted as the parity bit. With a word length of eight, and parity selected, the parity bit for the sync word is computed an added on by the MK68901 MFP.

RR RECEIVER READY

$\overline{RR}$ 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 TRANSMITTER READY

$\overline{TR}$ is asserted when the Buffer Empty bit is set in the TSR unless a break is currently being transmitted.

REGISTER ACCESSES

All register accesses are dependent on CLK as shown in the timing diagrams. To read a register, $\overline{CS}$ and $\overline{DS}$ must be asserted, and R/$\overline{W}$ must by high. The internal read control signal is essentially the combination of $\overline{CS}$, $\overline{DS}$, and RD/$\overline{WR}$. Thus, the read operation will begin when $\overline{CS}$ and $\overline{DS}$ go active and will end when either $\overline{CS}$ or $\overline{DS}$ goes inactive. The address bus must be stable prior to the start of the operation and must remain stable until the end of the operation. Unless a read operation or interrupt acknowledge cycle is in progress the data bus (D_0-D_7) will remain in the tri-state condition.

To write a register, $\overline{CS}$ and $\overline{DS}$ must be asserted and R/$\overline{W}$ must be low. The address must be stable prior to the start of the operation and must remain stable until the end of the operation. After the MK68901 asserts $\overline{DTACK}$, the CPU negates $\overline{DS}$,. At this time, the MFP latches the data bus and writes the contents into the appropriate register. Also when $\overline{DS}$ is negated, the MFP rescinds $\overline{DTACK}$.

For an interrupt acknowledge, the operation starts when IACK goes low, and ends when IACK goes high. The data bus is tri-stated when either IACK or $\overline{DS}$ goes high.

When $\overline{CS}$ or $\overline{IACK}$ are asserted the MFP starts an internal cycle. $\overline{DS}$ is needed to enable the address and data buffers. It is recommended taht $\overline{CS}$ and IACK be gated by $\overline{DS}$ so that DS is always present whenever an MFP bus cycle starts.

MK68901 ELECTRICAL SPECIFICATIONS – PRELIMINARY

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
T_A	Temperature under Bias	– 25 to + 100	°C
T_{stg}	Storage Temperature	– 65 to + 150	°C
V_I	Voltage on Any Pin with Respect to Ground	– 0.3 to + 7	V
P_D	Power Dissipation	1.5	W

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect reliability.

D.C. CHARACTERISTICS

T_A = 0°C to 70°C ; V_{CC} = + 5V ± 5% Unless Otherwise Specified

Symbol	Parameter	Test Condition	Min.	Max.	Unit
V_{IH}	Input High Voltage		2.0	V_{CC} + .3	V
V_{IL}	Input Low Voltage		– 0.3	0.8	V
V_{OH}	Output High Voltage (except $\overline{DTACK}$)	I_{OH} = – 120µA	2.4		V
V_{OL}	Output Low Voltage (except $\overline{DTACK}$)	I_{OL} = 2.0mA		0.5	V
I_{LL}	Power Supply Current	Outputs Open		180	mA
I_{LI}	Input Leakage Current	V_{IN} = 0 to V_{CC}		± 10	µA
I_{LOH}	Tri-state Output Leakage Current in Float	V_{OUT} = 2.4 to V_{CC}		10	µA
I_{LOL}	Tri-state Output Leakage Current in Float	V_{OUT} = 0.5V		– 10	µA
I_{OH}	$\overline{DTACK}$ Output Source Current	V_{OUT} = 2.4		– 400	µA
I_{OL}	$\overline{DTACK}$ Output Sink Current	V_{OUT} = 0.5		5.3	mA

All voltages are referenced to ground.

CAPACITANCE

T_A = 25°C, f = 1MHz unmeasured pins returned to ground.

Symbol	Parameter	Test Condition	Max.	Unit
C_{IN}	Input Capacitance	Unmeasured pins returned to ground	10	pF
C_{OUT}	Tri-state Output Capacitance		10	pF

AC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0Vdc ± 5%, GND = 0Vdc, T_A = 0°C to 70°C)

Number	Characteristic	Value				Unit	Fig.	Note
		MK68901-4		MK68901-5				
		Min.	Max.	Min.	Max.			
1	$\overline{CS}$, $\overline{DS}$ Width High	50		35		ns	21,22	5
2	R/$\overline{W}$, A1-A5 Valid to Falling $\overline{CS}$ (setup)	0		0		ns	21,22	
3	Data Valid Prior to Falling CLK	280		0		ns	22	
4	$\overline{CS}$, $\overline{IACK}$ Valid to Falling Clock (setup)	50		45		ns	21-24	3
5	CLK Low to $\overline{DTACK}$ Low		220		180	ns	21,22	
6	$\overline{CS}$, $\overline{DS}$ or $\overline{IACK}$ High to $\overline{DTACK}$ High		60		55	ns	21-24	
7	$\overline{CS}$, $\overline{DS}$ or $\overline{IACK}$ High to $\overline{DTACK}$ Tri-state		100		95	ns	21-24	
8	$\overline{DTACK}$ Low to Data Invalid (hold time)	0		0		ns	22	
9	$\overline{CS}$, $\overline{DS}$ or $\overline{IACK}$ High to Data Tri-state		50		50	ns	21,23,24	
10	$\overline{CS}$ or $\overline{DS}$ High to R/$\overline{W}$, A1-A5 Invalid (hold time)	0		0		ns	21,22	
11	Data Valid from $\overline{CS}$ Low		310		260	ns	21	3,6
12	Read Data Valid to $\overline{DTACK}$ Low (setup)	50		50		ns	21	
13	$\overline{DTACK}$ Low to $\overline{DS}$, $\overline{CS}$ or $\overline{IACK}$ High (hold time)	0		0		ns	21-23	
14	$\overline{IEI}$ Low to Falling $\overline{CLK}$ (setup)	50		50		ns	23,24	
15	$\overline{IEO}$ Valid from Clock Low (delay)		180		180	ns	23	1
16	Data Valid from Clock Low (delay)		300		300	ns	23	
17	$\overline{IEO}$ Invalid from $\overline{IACK}$ High (delay)		150		150	ns	23, 24	
18	$\overline{DTACK}$ Low from Clock High (delay)		180		165	ns	23, 24	
19	$\overline{IEO}$ Valid from $\overline{IEI}$ Low (delay)		100		100	ns	24	1
20	Data Valid from $\overline{IEI}$ Low (delay)		220		220	ns	24	
21	Clock Cycle Time	250	1000	200	1000	ns	21	
22	Clock Width Low	110		90		ns	21	
23	Clock Width High	110		90		ns	21	
24	$\overline{CS}$, $\overline{IACK}$ Inactive to Rising Clock (setup)	100		80		ns	21-23	4,5
25	I/O Minimum Active Pulse Width	100		100		ns	25	
26	$\overline{IACK}$ Width High	2		2		T_{CLK}	23-24	2
27	I/O Data Valid from Rising $\overline{CS}$ or $\overline{DS}$		450		450	ns	26	
28	Receiver Ready Delay from Rising RC		600		600	ns	27	
29	Transmitter Ready Delay from Rising TC		600		600	ns	28	
30	Timer Output Low from Rising Edge of $\overline{CS}$ or $\overline{DS}$ (A & B) (reset T_{OUT})		450		450	ns	29	7
31	T_{OUT} Valid from Internal Timeout		2 t_{CLK} + 300		2 t_{CLK} + 300	ns	29	2
32	Timer Clock Low Time	110		90		ns	29	

AC ELECTRICAL CHARACTERISTICS (continued)
(V_{CC} = 5.0Vdc ± 5%, GND = 0Vdc, T_A = 0°C to 70°C)

Number	Characteristic	Value				Unit	Fig.	Note
		MK68901-4		MK68901-5				
		Min.	Max.	Min.	Max.			
33	Timer Clock High Time	110		90		ns	29	
34	Timer Clock Cycle Time	250	1000	200	1000	ns	29	
35	$\overline{\text{RESET}}$ Low Time	2		1.8		µs	30	
36	Delay to Falling $\overline{\text{INTR}}$ from External Interrupt Active Transition		380		380	ns	25	
37	Transmitter Internal Interrupt Delay from Falling Edge of TC		550		550	ns	28	
38	Receiver Buffer Full Interrupt Transition Delay from Rising Edge of RC		800		800	ns	27	
39	Receiver Error Interrupt Transition Delay from Falling Edge of RC		800		800	ns	27	
40	Serial in Set Up Time to Rising Edge of RC (divide by one only)	80		70		ns	27	
41	Data Hold Time from Rising Edge of RC (divide by one only)	350		325		ns	27	
42	Serial Output Data Valid from Falling Edge of TC (÷1)		440		420	ns	28	
43	Transmitter Clock Low Time	500		450		ns	28	
44	Transmitter Clock High Time	500		450		ns	28	
45	Transmitter Clock Cycle Time	1.05	∞	0.95	∞	µs	28	
46	Receiver Clock Low Time	500		450		ns	27	
47	Receiver Clock High Time	500		450		ns	27	
48	Receiver Clock Cycle Time	1.05	∞	0.95	∞	µs	27	
49	$\overline{\text{CS}}$, $\overline{\text{IACK}}$, $\overline{\text{DS}}$ Width Low		80		80	T_{CLK}	29	2
50	Serial Output Data Valid from Falling Edge of TC (÷16)		490		370	ns	28	

Notes :
1. $\overline{\text{IEO}}$ only goes low if no acknowledgeable interrupt is pending. If $\overline{\text{IEO}}$ goes low, $\overline{\text{DTACK}}$ and the data bus remain tri-stated.
2. T_{CLK} refers to the clock applied to the MFP CLK input pin. t_{CLK} refers to the timer clock signal, regardless of whether that signal comes from the XTAL 1/XTAL2 crystal clock inputs or the TAI or TBI timer inputs.
3. If the setup time is not met, $\overline{\text{CS}}$ or IACK will not be recognized until the next falling CLK.
4. If this setup time is met (for consecutive cycles), the minimum hold-off time of one clock cycle will be obtained. If not met, the hold-off will be two clock cycles.
5. $\overline{\text{CS}}$ is latched internally, therefore if spec's 1 and 24 are met then $\overline{\text{CS}}$ may be reasserted before the rising clock and still terminate the current bus cycle. The new bus cycle will be delayed by the MK68901 until all appropriate internal operations have completed.
6. Although $\overline{\text{CS}}$ and $\overline{\text{DTACK}}$ are synchronized with the clock, the data out during a read cycle is asynchronous to the clock, relying only on $\overline{\text{CS}}$ for timing.
7. Spec. 30 applies to timer outputs TAO and TBO only.

TIMER A.C. CHARACTERISTICS

Definitions :
Error = Indicated Time Value - Actual Time Value
tpsc = t_{CLK}x Prescale Value
Internal Timer Mode

Single Interval Error (free running) (note 2)	± 100ns
Cumulative Internal Error	0
Error between Two Timer Reads	± (tpsc + $4t_{CLK}$)
Start Timer to Stop Timer Error	+ ($2t_{CLK}$ + 100ns) to - (tpsc + $6t_{CLK}$ + 100ns)
Start Timer to Read Timer Error	+ 0 to – (tpsc + $6t_{CLK}$ + 400ns)
Start Timer to Interrupt Request Error (note 3)	- $2t_{CLK}$ to – ($4t_{CLK}$ + 800ns)

PULSE WIDTH MEASUREMENT MODE

Measurement Accuracy (note 1)	+ $2t_{CLK}$ to – (tpsc + $4t_{CLK}$)
Minimum Pulse Width	$4t_{CLK}$

EVENT COUNTER MODE

Minimum Active Time of TAI, TBI	$4t_{CLK}$
Minimum Inactive Time of TAI, TBI	$4t_{CLK}$

Notes : 1. Error may be cumulative if repetitively performed.
2. Error with respect to T_{OUT} or INT if note 3 is true.
3. Assuming it is possible for the timer to make an interrupt request immediately.

Figure 21 : Read Cycle.

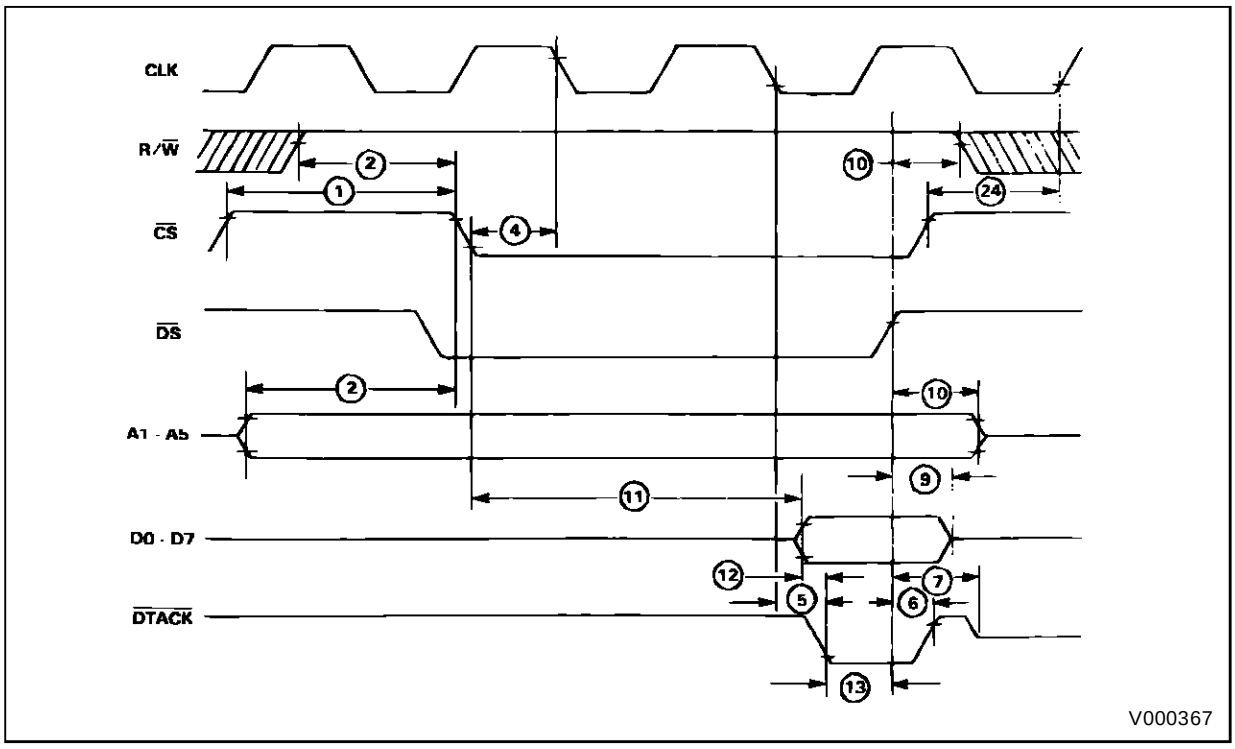
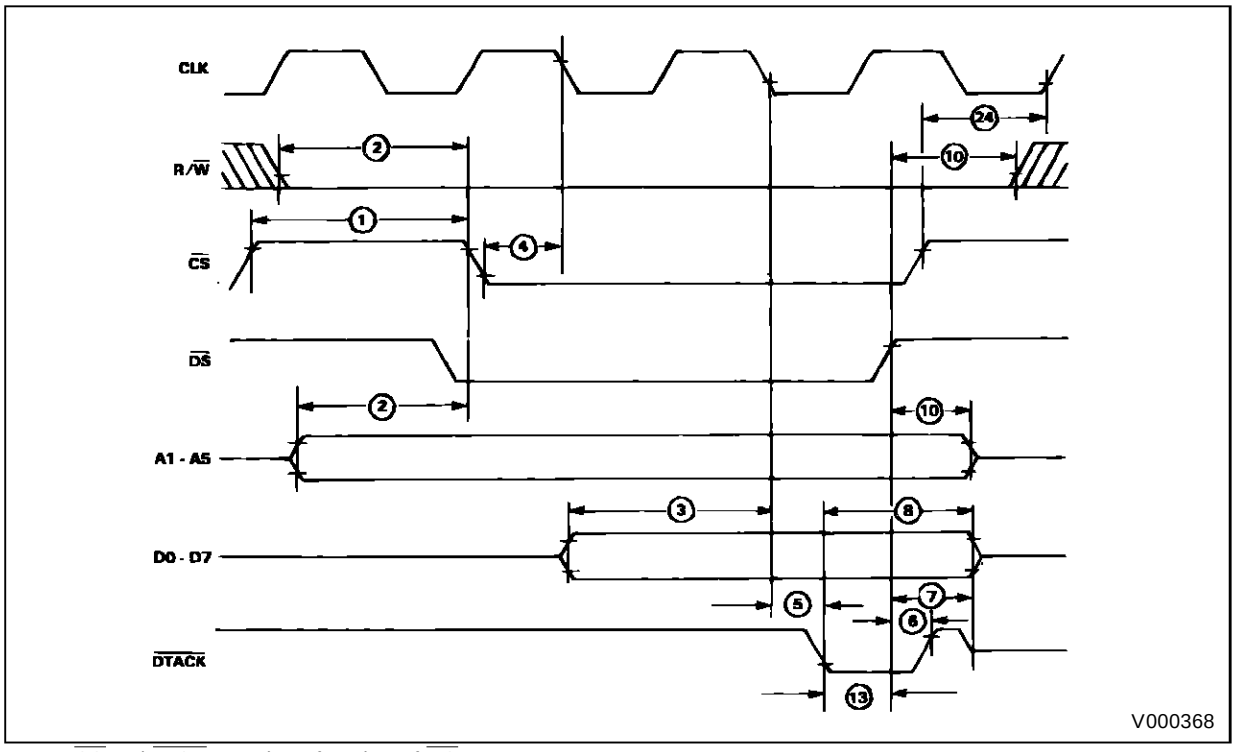


Figure 22 : Write Cycle.



Note : $\overline{CS}$ and $\overline{IACK}$ must be a function of $\overline{DS}$.

Figure 23 : Interrupt Acknowledge ($\overline{IEI}$ low).

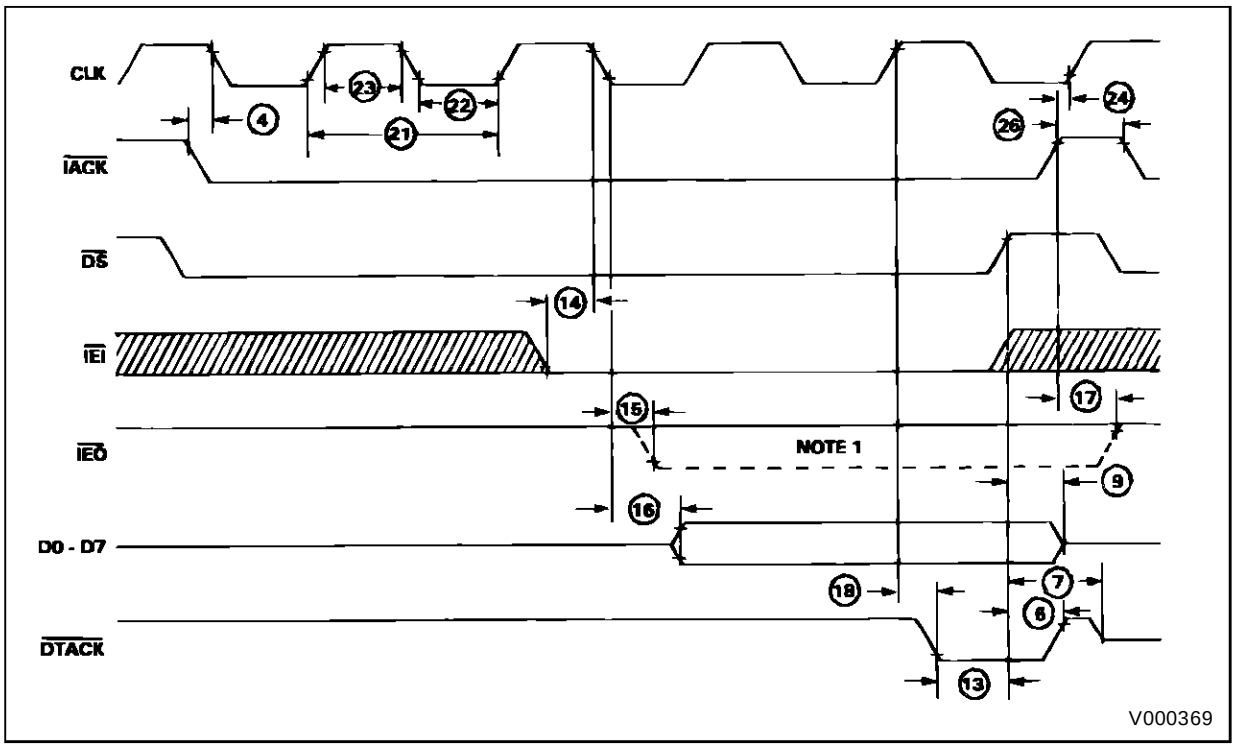
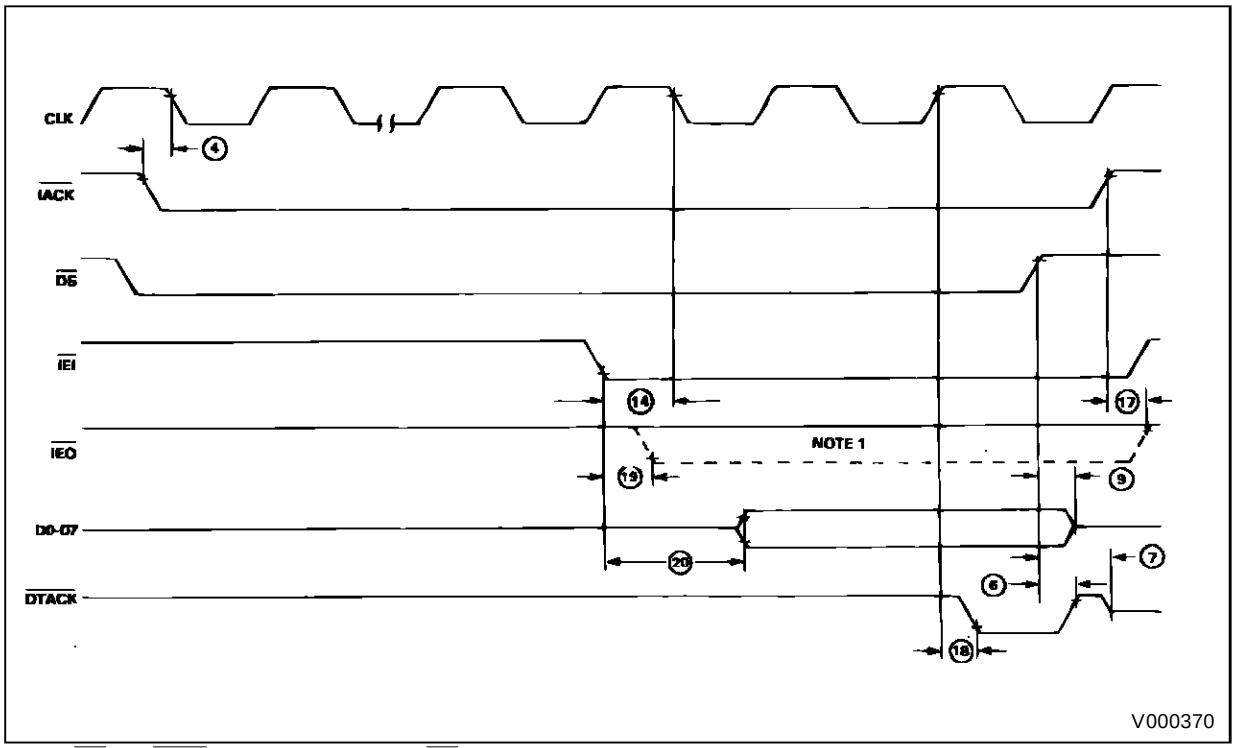
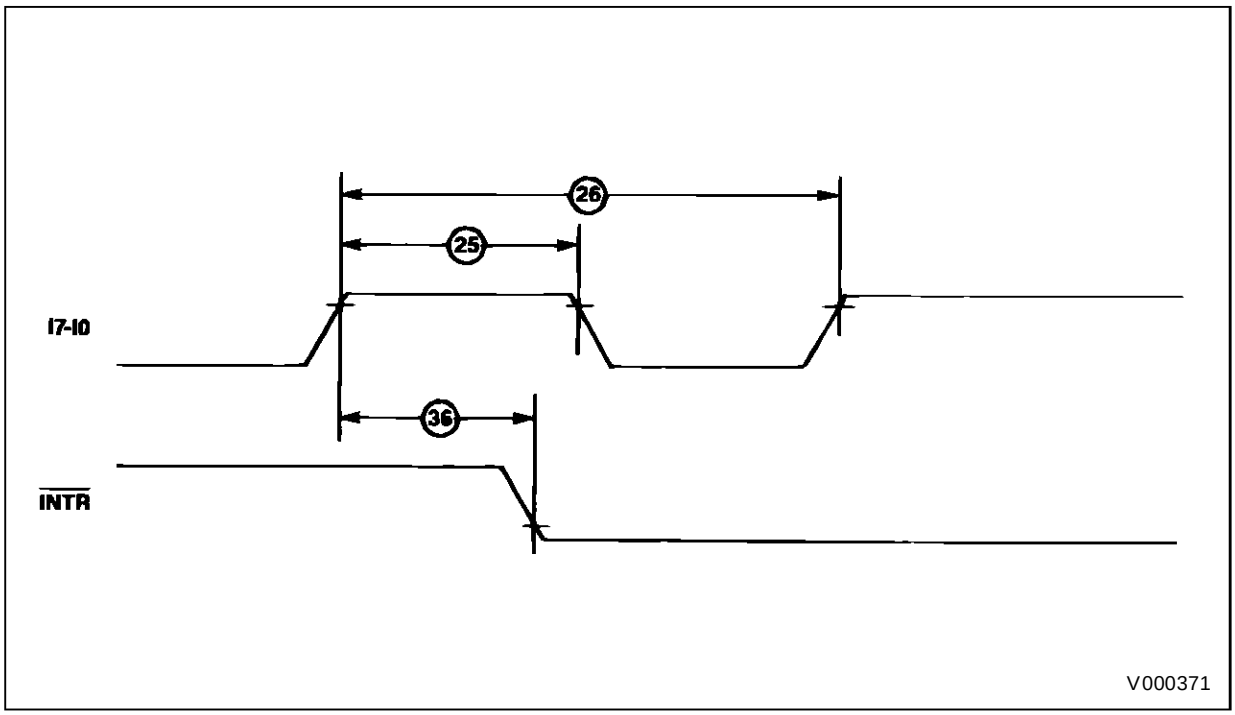


Figure 24 : Interrupt Acknowledge Cycle ($\overline{IEI}$ high).



Note : $\overline{CS}$ and $\overline{IACK}$ must be a function of $\overline{DS}$.

Figure 25 : Interrupt Timing.



Note : Active edge is assumed to be the rising edge.

Figure 26 : Port Timing.

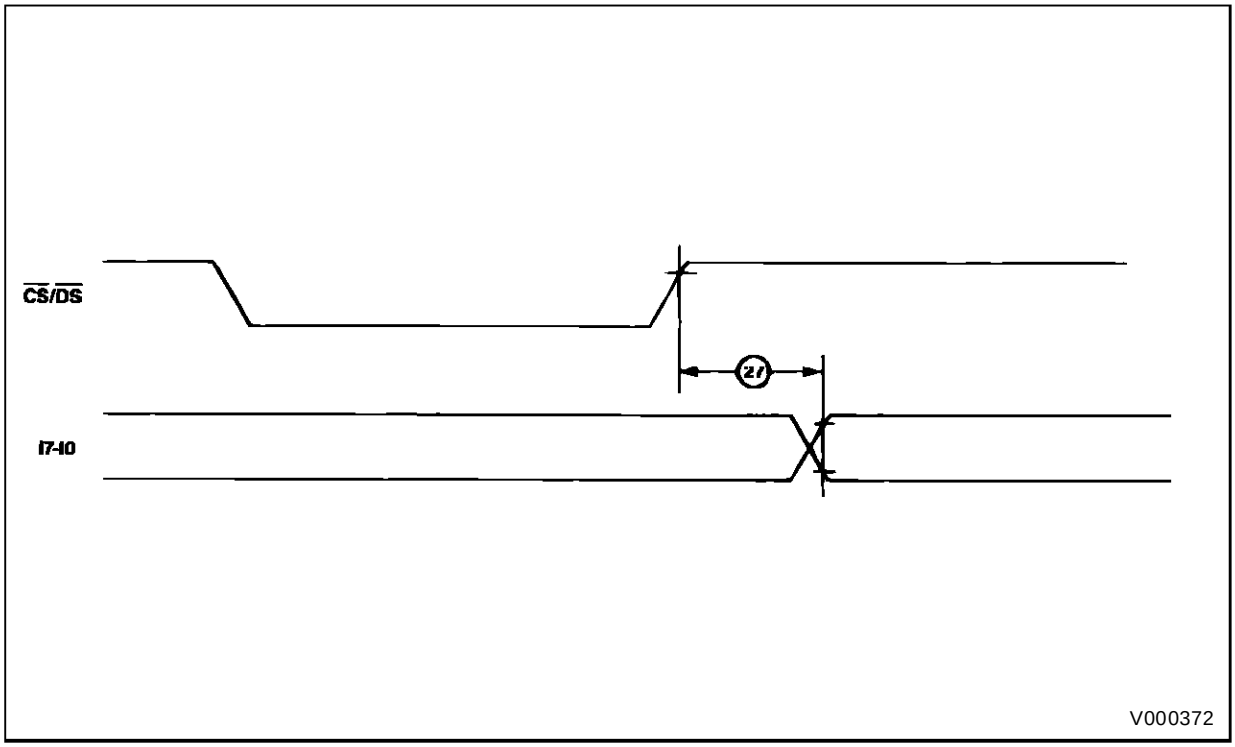


Figure 27 : Receiver Timing.

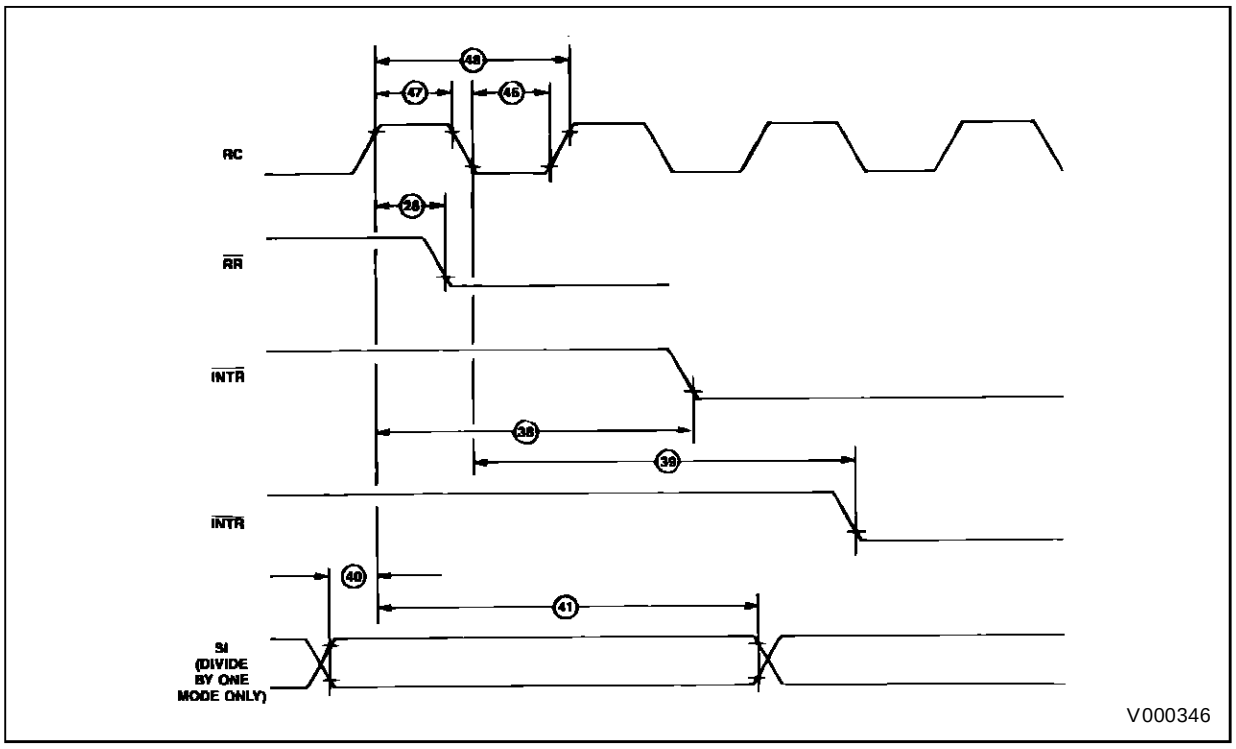


Figure 28 : Transmitter Timing.

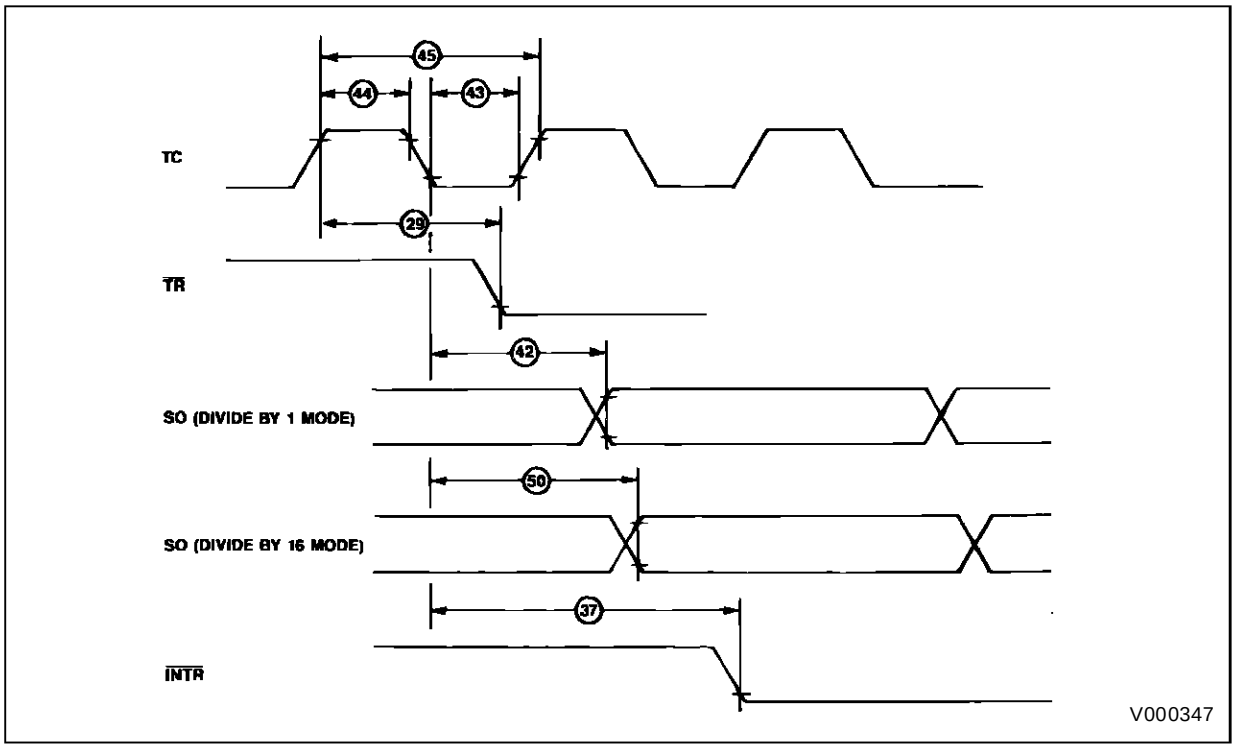


Figure 29 : Timer Timing.

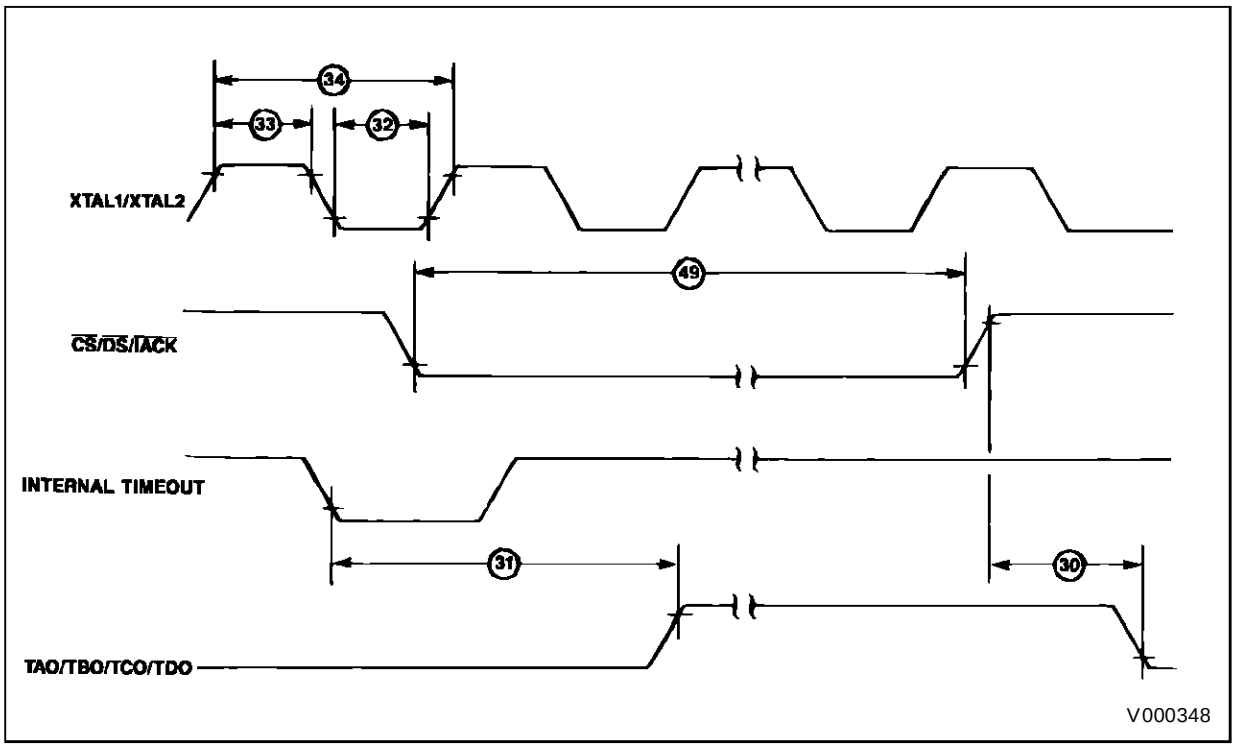


Figure 30 : Reset Timing.

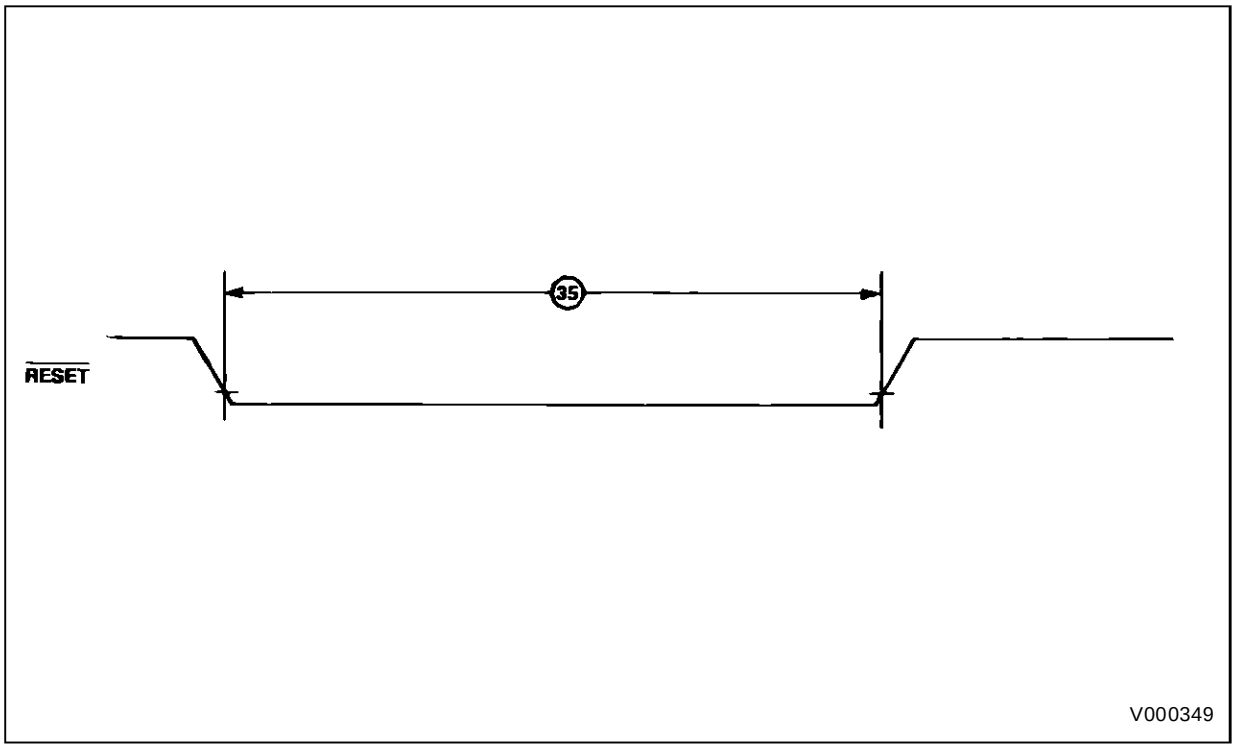


Figure 31 : Typical Output.

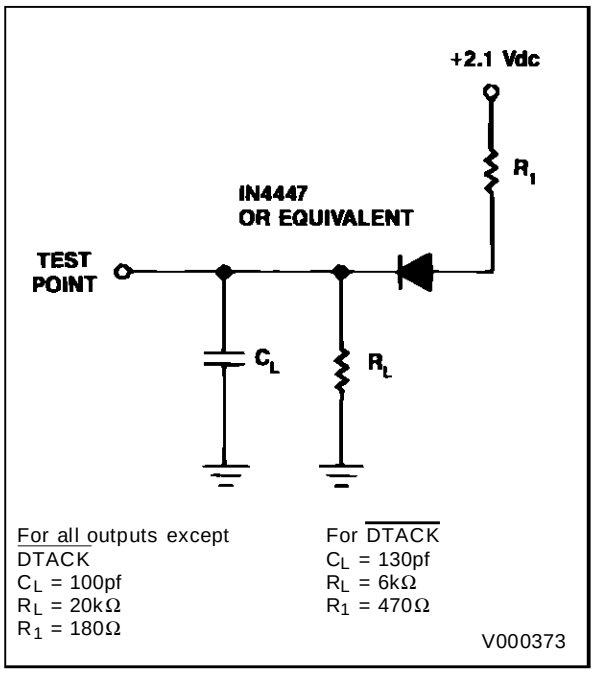


Figure 32 : INTR Test Load.

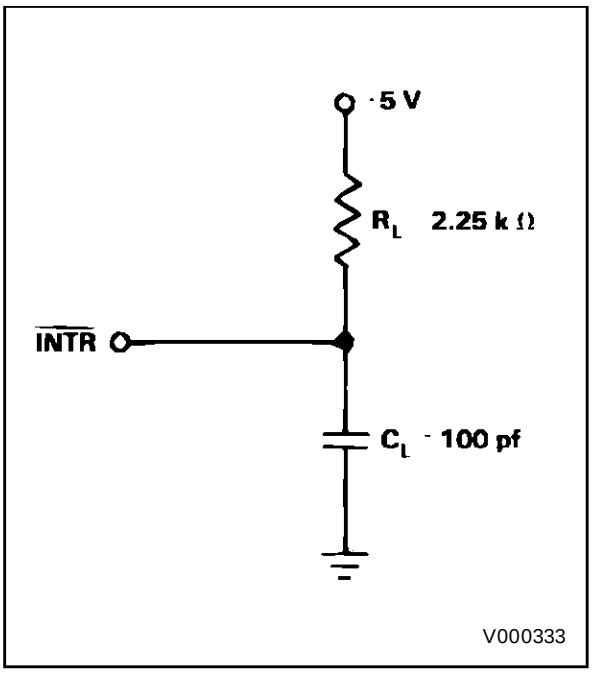
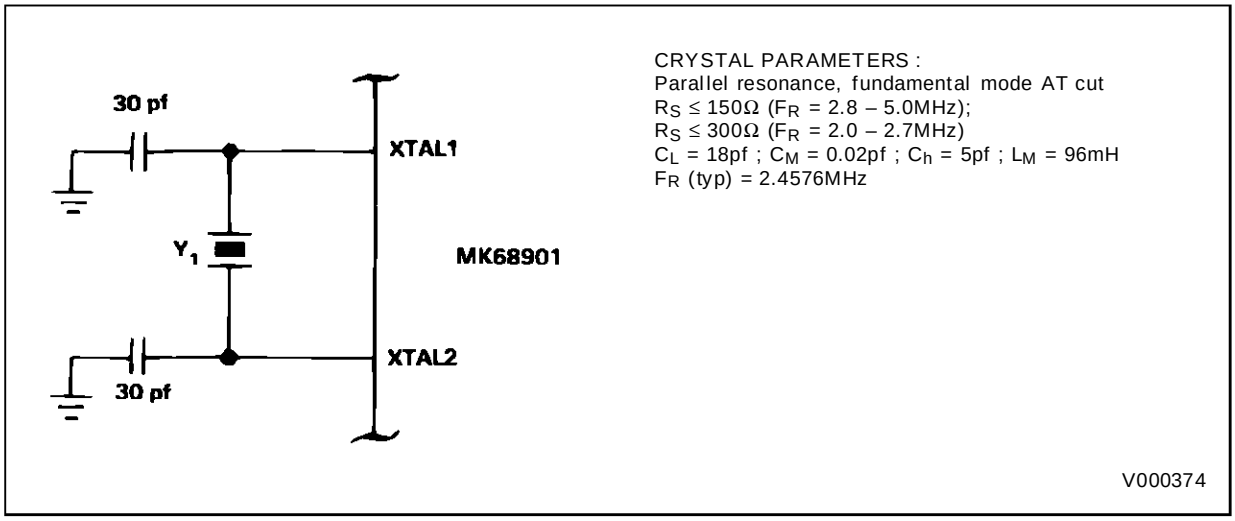


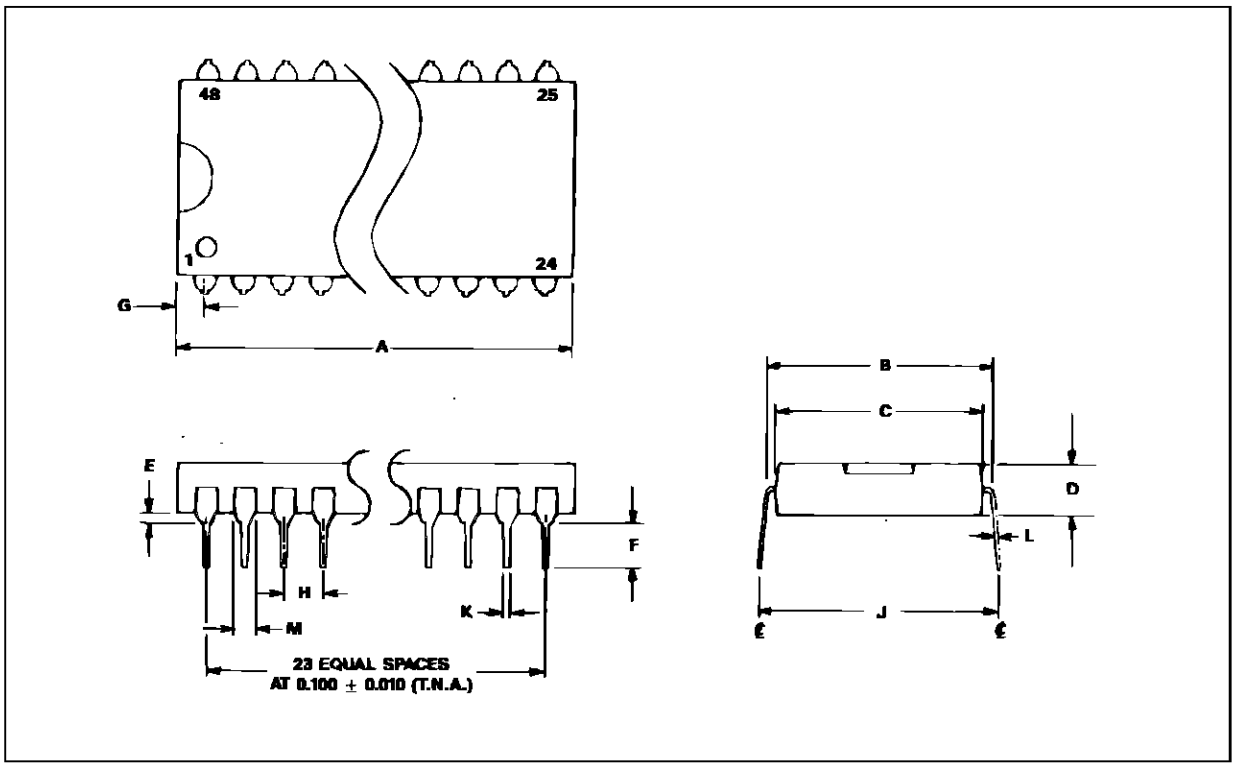
Figure 33 : MK68901 MFP External Oscillator Components.



MK68901 ORDER CODES

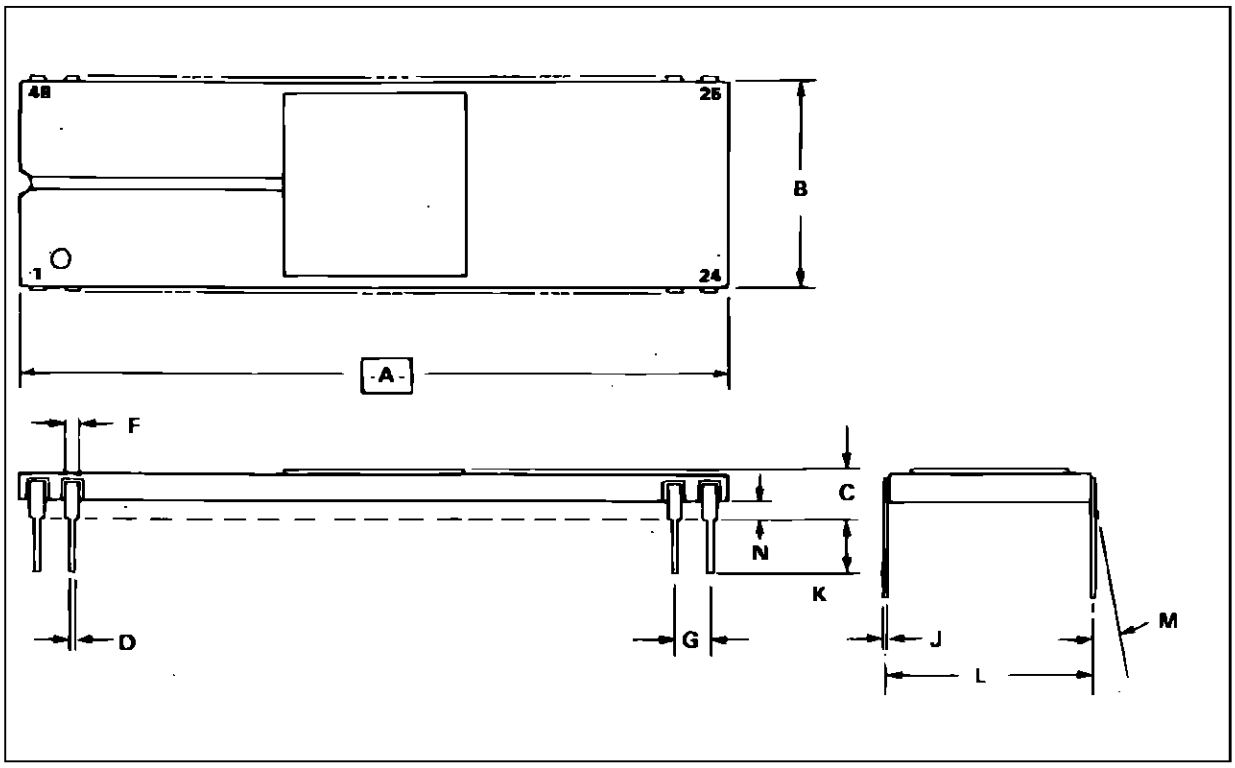
Part Number	Package Type	Max. Clock Frequency	Temperature Range
68901P04	Ceramic DIP	4.0MHz	0° to 70°C
68901P05	Ceramic DIP	5.0MHz	0° to 70°C
68901N04	Plastic DIP	4.0MHz	0° to 70°C
68901N05	Plastic DIP	5.0MHz	0° to 70°C
68901Q04	Plastic PLCC	4.0MHz	0° to 70°C
68901Q05	Plastic PLCC	5.0MHz	0° to 70°C

MK68901 48–PIN PLASTIC DUAL–IN–LINE PACKAGE (N)



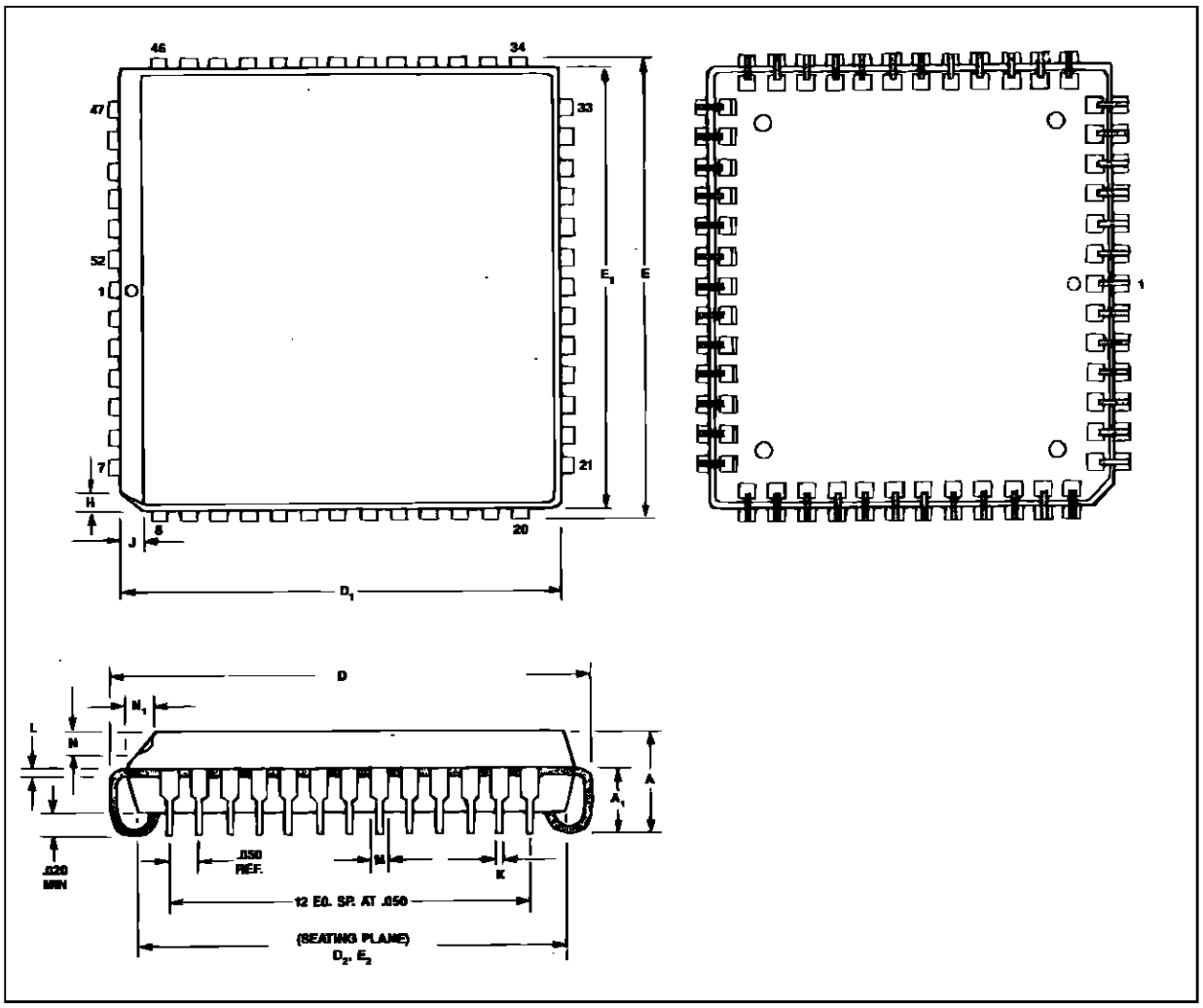
Dim	Millimeters		Inches	
	Min.	Max.	Min.	Max.
A	61.468	62.738	2.420	2.470
B	14.986	16.256	.590	.640
C	13.462	13.97	.530	.550
D	3.556	4064	.140	.160
E	0.381	1.524	.015	.060
F	3048	3.81	.120	.150
G	1.524	2.286	.060	.090
H	1.186	1.794	.090	.110
J	15.24	17.78	.600	.700
K	0.381	0.533	.015	.021
L	0.203	0.305	.008	.012
M	1.143	1.778	.045	.070

MK68901 48–PIN CERAMIC DUAL–IN–LINE PACKAGE (P)



Dim	Inches	
	Min.	Max.
A	2.376	2.424
B	0.576	0.604
C	0.120	0.160
D	0.015	0.021
F	0.030	0.055
G	0.100 BSC	
J	0.008	0.013
K	0.100	0.165
L	0.590	0.616
M	0°	10°
N	0.040	0.060

MK68901 52–PIN PLASTIC LEADED CHIP CARRIER (Q)



Dim	Inches	
	Min.	Max.
A	.165	.180
A_1	.090	.130
D	.785	.795
D_1	.750	.756
D_2	.690	.730
E	.785	.795
E_1	.750	.756
E_2	.690	.730
H	.042	.048
J	.042	.048
K	.013	.024
L	.008	.014
M	.026	.032
N/N_1	.043	.048

MK68901 PIN CONNECTIONS

PLCC	DIP	FUNC.	PLCC	DIP	FUNC.	PLCC	DIP	FUNC.
1	–	NC	19	17	XTAL1	37	33	$\overline{IEO}$
2	1	R/$\overline{W}$	20	18	XTAL2	38	34	IEI
3	2	A1	21	–	NC	39	35	CLK
4	3	A2	22	19	TAI	40	36	GND
5	4	A3	23	20	$\overline{TBI}$	41	37	D0
6	5	A4	24	21	$\overline{RESET}$	42	38	D1
7	6	A5	25	22	IO	43	39	D2
8	7	TC	26	23	I1	44	40	D3
9	8	SO	27	24	I2	45	41	D4
10	9	SI	28	25	I3	46	42	D5
11	10	RC	29	26	I4	47	43	D6
12	11	V_{CC}	30	27	I5	48	44	$\overline{D7}$
13	–	NC	31	28	I6	49	45	$\overline{IACK}$
14	12	NC	32	29	I7	50	46	$\overline{DTACK}$
15	13	TAO	33	–	$\overline{NC}$	51	47	$\overline{DS}$
16	14	TBO	34	30	$\overline{TR}$	52	48	CS
17	15	TCO	35	31	$\overline{RR}$			
18	16	TDO	36	32	$\overline{INTR}$			

Note : NC – No Connection