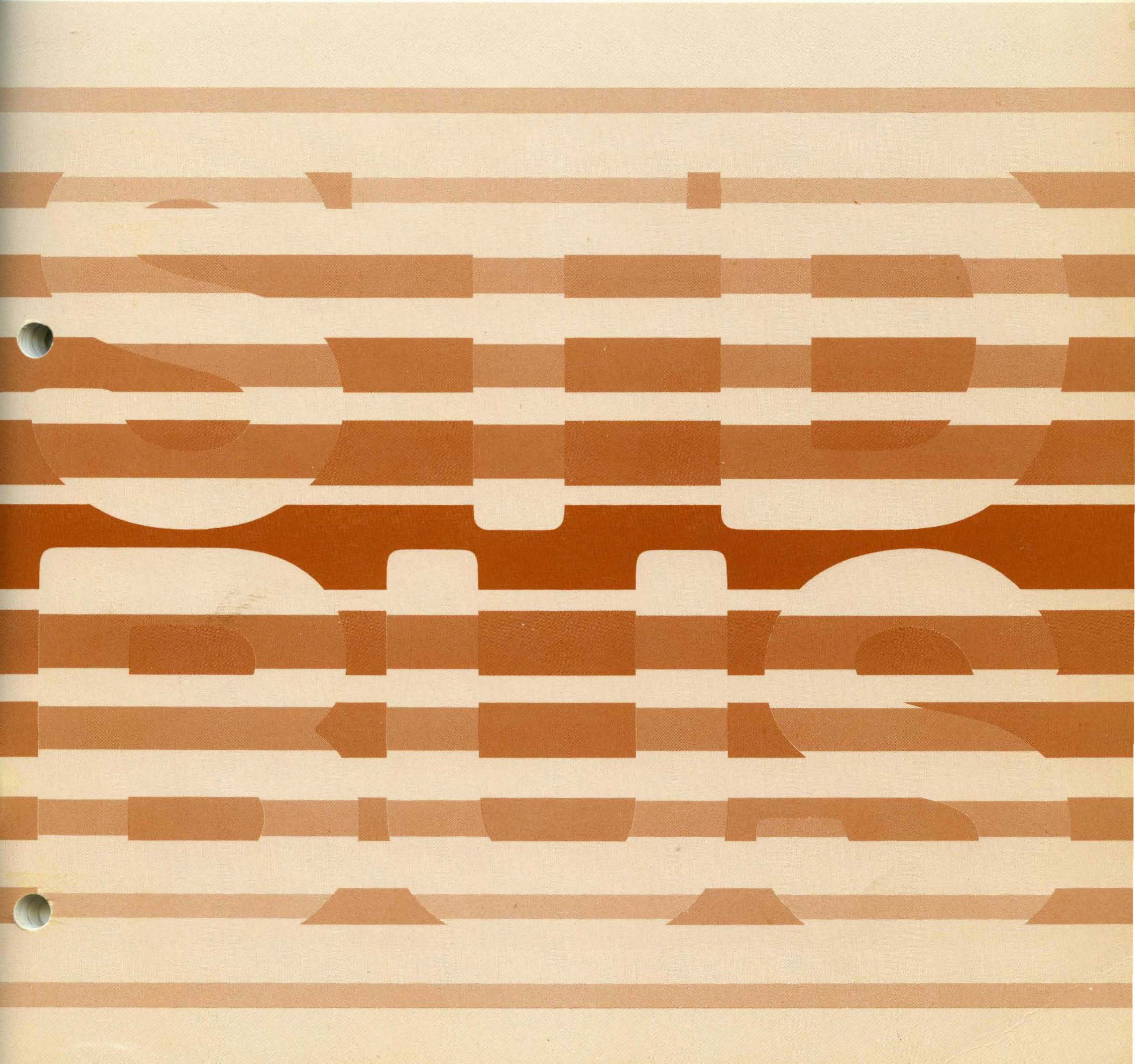
PRO-LOG
CORPORATION

STD 7000

7308 Counter/Timer Card

USER'S MANUAL



NOTICE

The information in this document is provided for reference only. Pro-Log does not assume any liability arising out of the application or use of the information or products described herein.

This document may contain or reference information and products protected by copyrights or patents and does not convey any license under the patent rights of Pro-Log, nor the rights of others.

7308
Counter/Timer Card
USER'S MANUAL

PRO-LOG
CORPORATION

FOREWORD

This manual explains how to use Pro-Log's 7308 Counter/Timer Card. It is structured to reflect the answers to basic questions that you, the user, might ask yourself about the 7308. We welcome your suggestions on how we can improve our instructions.

The 7308 is part of Pro-Log's Series 7000 STD BUS hardware. Our products are modular, and they are designed and built with second-sourced parts that are industry standards. They provide the industrial manager with the means of utilizing his own people to control the design, production, and maintenance of the company's products that use STD BUS hardware.

Pro-Log supports its products with thorough and complete documentation. Also, to provide maximum assistance to the user, we teach courses on how to design with, and to use microprocessors and the STD BUS products.

You may find the following Pro-Log documents useful in your work: *Microprocessor User's Guide* and the *Series 7000 STD BUS Technical Manual*. If you would like a copy of these documents, please submit your request on your company letterhead.

Contents

List of Figures

LIST OF TABLES

Section 1

Introducing the 7308 Multichannel Counter/Timer Card (Purpose and Main Features)

The 7308 is a fully programmable, multichannel counter/timer card (Fig. 1-1). It runs as a stand-alone peripheral after receiving setup instructions from the system processor card. An onboard interrupt and status polling system identifies completed operations. (See Fig. 1-2 for block diagram).

Based on the 8253-5 counter/timer, the 7308 provides three 16-bit counter/timer channels. The card has a crystal oscillator and tapped clock divider, an 8-input multiplexer for each channel, and programmable logic states at each clock, gate, and output signal.

The three channels are configured independently by the program. Each is suitable for event counting from DC to 2.5 MHz, for one-shot simulation with hardware and software triggering and retriggering, and for precisely timed interrupts. A special feature allows any channel to interrupt after the nth programmed event or loop iteration.

Main features of the 7308 are:

- Three independent 16-bit counter/timer channels with six operating modes each.
- Count source multiplexer, with eight inputs for each channel.
- Programmable logic polarity at user inputs and outputs.
- No adjustments - minimizes OEM effort and field errors.
- Onboard interrupt latches and masks for stand-alone operation; STDMG-approved interrupt connector for expansion with 7320 priority interrupt card.
- Onboard crystal oscillator for accurate programmed timing.
- Provision for external time-reference signal input, and clock output for other STD BUS cards.
- Universal processor compatibility: Z80, 8085A, 6800 and others.
- Multisourced industry-standard components.
- Single +5V operation.

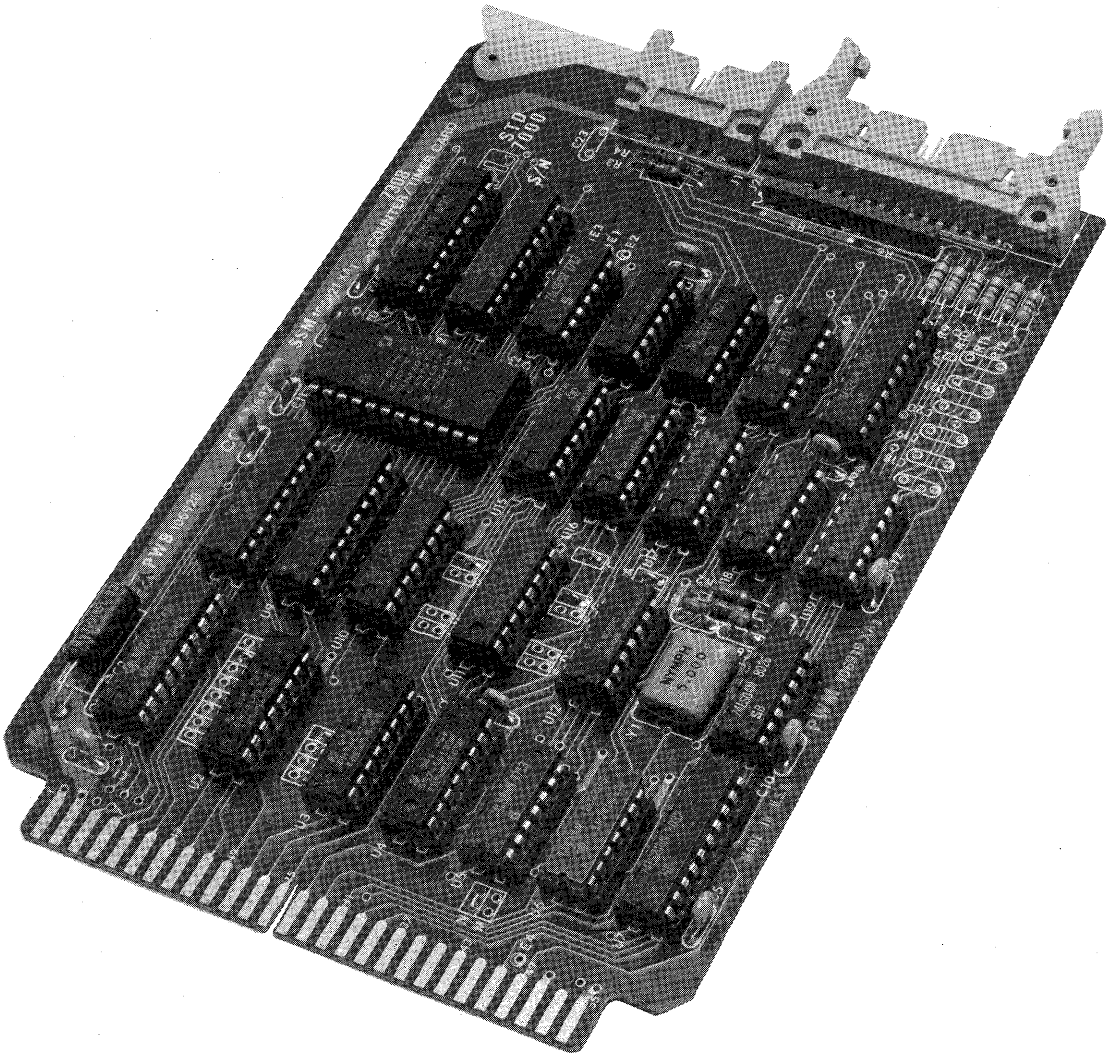
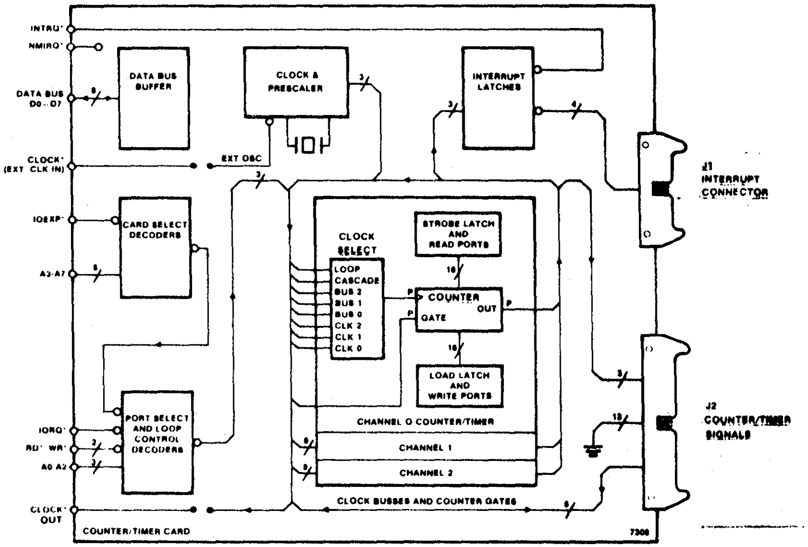


Figure 1-1. 7308 Multichannel Counter/Timer Card.



*INDICATES LOW LEVEL ACTIVE
P PROGRAMMABLE LOGIC POLARITY

Figure 1-2. Block Diagram of 7308 Multichannel Counter/Timer Card.

Section 2

How to Use the 7308 (Installation and Specifications)

The 7308 operates as part of an STD BUS card rack system. You can plug it directly into the STD BUS backplane (Fig. 2-1) or extend it from the motherboard with a 7901 card extender, or equivalent. In this configuration, the card is mapped at processor I/O port addresses.

I/O Mapped Card Addressing

(Figure 2-1: I/O Mapped Operation in Local Card Rack)
In its normal operation, the 7308 is addressed directly by the processor card. The 7308's input and output ports respond to single read and write instructions executed in the processor's operating program. The 7308 is enabled when a jumper-selected combination of address lines A0 through A7 is present, and when the following control lines are active: IORQ*, IOEXP*, and either RD* or WR*.

The 7308 occupies eight consecutive I/O addresses regardless of its mapping assignment. To facilitate testing, we ship the card with hexadecimal addresses D8 through DF connected. You may retain these addresses or change them by moving the installed jumper wires. While the card's port addresses are generally arbitrary, they must differ from all other I/O port addresses in the system. If they do not differ, multiple cards will respond to the same input instruction, resulting in bus contention.

Address Decoder Operation

(Figure 2-2: I/O Port Select Circuit)

Refer to the 7308 Schematic diagram (Figure B-1) in Appendix B.

Two cascaded 74LS138 decoders (U2 and U3) form the card select circuit which enables the 7308's data bus buffer (U1) and either the 74LS138 port select decoder (U4) or the 8253 device (U14).

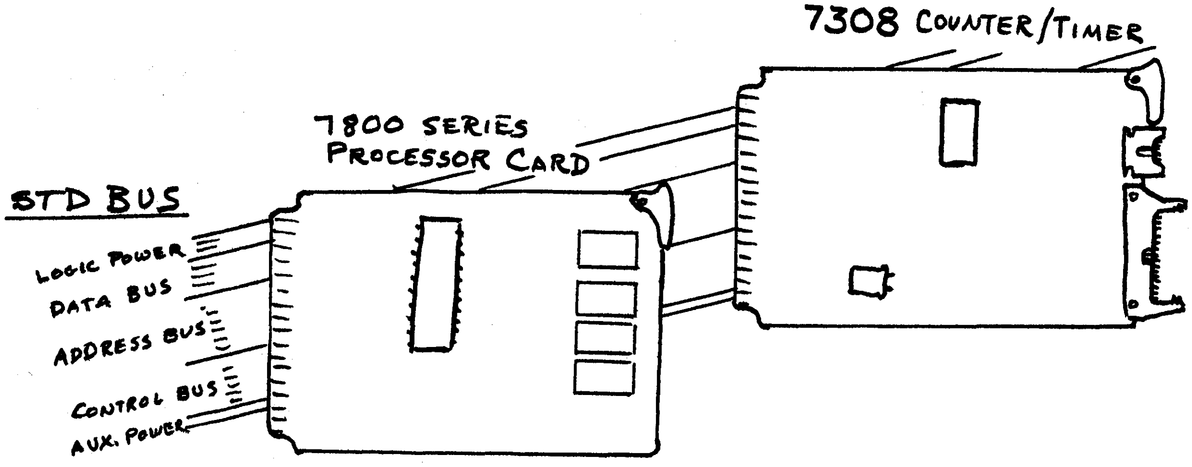


FIGURE 2-1 : I/O MAPPED OPERATION IN LOCAL CARD RACK

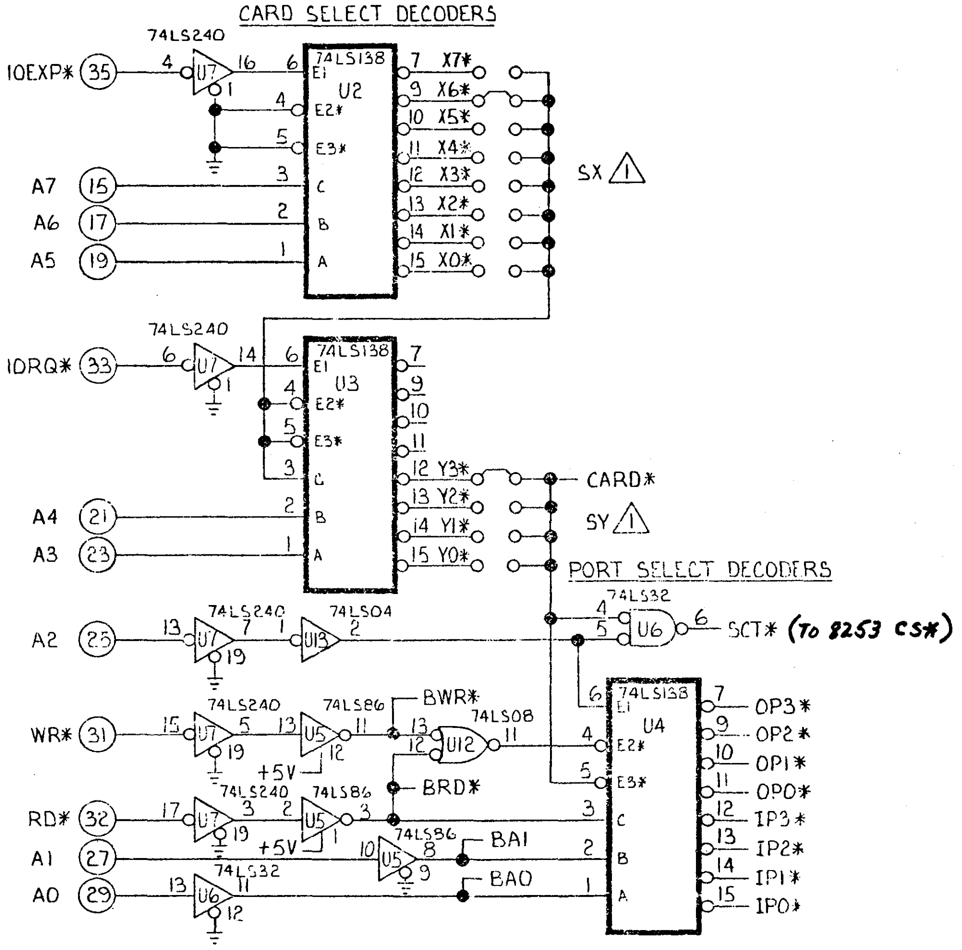


FIGURE 2-2: I/O PORT SELECT CIRCUIT (TYPICAL)

The 7308 occupies eight I/O port addresses, with four of the I/O ports internal to the 8253-5 (Ports D8-DB as shipped) and the other four ports. (DC-DF) controlled by the port select decoder (U4).

When address line A2 is low, the 8253-5 ports are selected. The device contains internal circuitry which decodes address lines A0 and A1 with RD* and WR* to control read and write operations for ports D8-DB.

When address line A2 is high, port select decoder U4 is enabled with RD* and WR* to generate input port strobes IP0* - IP3* and output port strobes OP0*-OP3*. These strobes control read and write operations for ports DC-DF.

The functions of the eight input and output ports are shown in Table 3-13.

Changing the 7308's Port Addresses

Refering to the 7308 assembly diagram, (Appendix B) locate decoders U2 and U3 (74LS138) next to the STD BUS edge connector. Each decoder has a dual row of pads that form decoder output-select matrices. Make one (and only one) connection to each matrix next to U2 and U3.

Figure 2-3 shows the numbering of the pads next to the decoder chips on the 7308. It also shows the jumpers (at X6 and Y3) that produce hexadecimal port addresses D8-DF (the selections made when the card is shipped).

(Figure 2-3: Decoder Jumper Pad Numbering)

Table 2-1 shows where to place jumper straps to obtain any eight sequential port addresses in the hexadecimal range of 00 through FF. You can change the 7308's address range to occupy any eight ports with the restriction that the least significant digit of the lowest of the eight addresses must be either a zero or an eight (the card can occupy addresses 20-27 or 58-5F, for example).

(Table 2-1: Port Address Decoder Jumper Selection)

Determine which block of addresses you want the card to occupy, then find the most significant digit of the address along the left axis of Table 2-1, then the block of eight least significant digits across the top of the

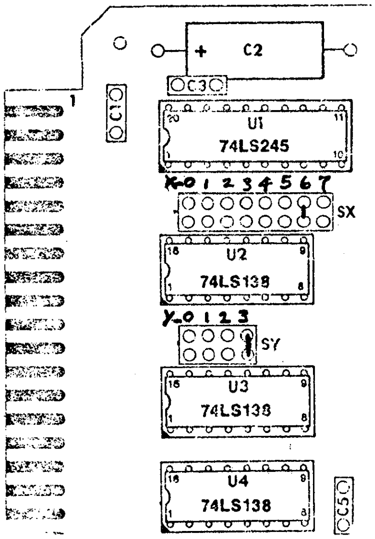


FIGURE 2-3: DECODER JUMPER PAD NUMBERING (TYPICAL)

MOST SIGNIFICANT HEX ADDRESS	LEAST SIGNIFICANT HEX ADDRESS																JUMPER SELECTION
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	X0 Y0								X0 Y1								X AND Y
1	X0 Y2								X0 Y3								
2	X1 Y0								X1 Y1								
3	X1 Y2								X1 Y3								
4	X2 Y0								X2 Y1								
5	X2 Y2								X2 Y3								
6	X3 Y0								X3 Y1								
7	X3 Y2								X3 Y3								
8	X4 Y0								X4 Y1								
9	X4 Y2								X4 Y3								
A	X5 Y0								X5 Y1								
B	X5 Y2								X5 Y3								
C	X6 Y0								X6 Y1								
D	X6 Y2								X6 Y3								
E	X7 Y0								X7 Y1								
F	X7 Y2								X7 Y3								

TABLE 2-1: PORT ADDRESS JUMPER SELECTION

table. Read the pair of jumper connections at the intersection of your selections. For example, obtaining addresses 20-27 requires jumpers X1 and Y0; addresses 58-5F require jumpers X2 and Y3. If you change the address jumpers, remember to remove the jumpers that were installed when you received the card (X6 and Y3).

The pad matrices next to U2 and U3 are on 0.10 inch (0.25 cm) centers. If you anticipate changing address selections frequently, you can replace the jumper wires with wire-wrap posts or equivalent.

Electrical Specifications

Vcc = +5V ± 5%

Icc = 750mA maximum, 450 mA typical

Address, data, and control busses meet all STD BUS general electrical specifications

User interface inputs are 5 LSTTL loads maximum. Counter/timer outputs can drive 55 LSTTL loads minimum, and interrupt outputs are open collector with 10 LSTTL load sink capability. Counter/timer inputs and outputs have programmable polarity; interrupt outputs are low level active. See Tables 2-3, 2-4 and 2-5.

Timing Specifications

Figure 2-4 shows the relationship of waveforms at the user's counter/timer signal interface connector J2, and Table 2-2 gives min/max timing specifications that apply to the clock, gate, and output signals of any one of the three channels over the specified Vcc and operating temperature ranges.

Note that since the clock and gate input polarities are programmable, Figure 2-4 and Table 2-2 assume that the clock is active on the falling edge and the gate is high level active (IPS=0 and GTS=1; see Table 3-11). This corresponds to the normal logic polarity at the inputs to the 8253-5 device.

(Figure 2-4: Waveforms at User Interface Connector J2)

(Table 2-2: User Interface Timing Specification)

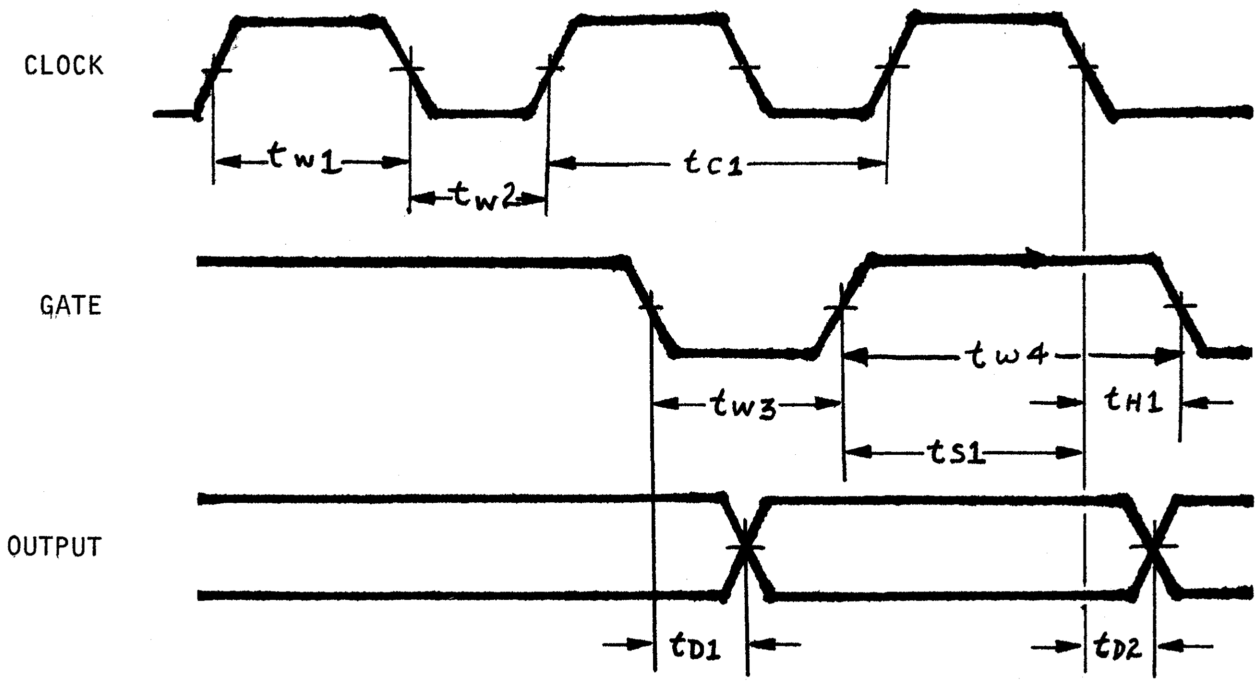


FIGURE 2-4: WAVEFORMS AT USER INTERFACE CONNECTOR J2

PARAMETER		MIN	MAX	UNITS
t_{C1}	Clock period asymmetrical waveform	400		ns
		0	2.5	MHz
	symmetrical waveform	500		ns
		0	2.0	MHz
t_{W1}	Clock high	250		ns
t_{W2}	Clock low	150		
t_{W3}	Gate low	125		
t_{W4}	Gate high	200		
t_{S1}	Gate setup time prior to Clock ↘	175		ns
t_{H1}	Gate hold time after to Clock ↘	25		
t_{D1}	Output delay after Gate ↘		525	ns
t_{D2}	Output delay after Clock ↘		400	

TABLE 2-2: USER INTERFACE TIMING SPECIFICATION

Signal Input Loading and Output Drive

Table 2-3 shows the STD BUS pins that the 7308 uses and the number of LSTTL loads represented by each pin. Pins not used by the card are electrically open. Tables 2-4 and 2-5 show the same information for the interrupt connector J1 and the counter/timer signal connector J2.

For the signal functions and complete electrical specifications of the STD BUS, see Reference 1.

(Table 2-3: STD BUS Edge Connector Pins for the 7308)
(Table 2-4: J1 Connector Pin List for Interrupt Status)
(Table 2-5: J2 Connector Pin List for Counter/Timer Channels)

User Vcc Output

User interface connector J2 provides system Vcc on pins 24 and 26 for powering user-added signal conditioning circuitry. These output pins are current limited by a one-ohm 1/4 watt resistor each to protect the connector. Output loading on these pins should be restricted to no more than 200 mA each; The output voltage relative to system Vcc will be reduced by 200 mA (typically) with a 200 mA load at each pin.

Mechanical Specifications

The 7308 meets all general mechanical specifications of the STD BUS, except for J1 and J2 connector protrusion, which is 0.25 inch (0.64 cm) maximum; and J2 mating connector latches, which may protrude up to 0.65 inch (1.65 cm) maximum.

7308 Environmental Specifications

(Table 2-6: 7308 Environmental Specifications)

STD 7308 EDGE CONNECTOR PIN LIST							
			PIN NUMBER	PIN NUMBER			
		OUTPUT (LSTTL DRIVE)			OUTPUT (LSTTL DRIVE)		
	INPUT (LSTTL LOADS)					INPUT (LSTTL LOADS)	
MNEMONIC							MNEMONIC
+5V	IN		2	1		IN	+5V
GROUND	IN		4	3		IN	GROUND
-5V			6	5			-5V
D7	1	55	8	7	55	1	D3
D6	1	55	10	9	55	1	D2
D5	1	55	12	11	55	1	D1
D4	1	55	14	13	55	1	D0
A15			16	15		1	A7
A14			18	17		1	A6
A13			20	19		1	A5
A12			22	21		1	A4
A11			24	23		1	A3
A10			26	25		1	A2
A9			28	27		1	A1
A8			30	29		1	A0
RD*	1		32	31		1	WR*
MEMRQ*			34	33		1	IORC*
MEMEX			36	35		1	IOEXP
MCSYNC*			38	37			REFRESH*
STATUS 0*			40	39			STATUS 1*
BUSRQ*			42	41			BUSAK*
INTRQ*		10[a]	44	43			INTAK*
NMIRQ*			46	45			WAITRQ*
PBRESET*			46	47			SYSRESET*
CNTRL*			50	49			CLOCK*
PC 1	IN		52	51		OUT	PC 0
AUX GND			54	53			AUX GND
AUX -V			56	55			AUX +V

* Active Low-level Logic [a] Open-Collector Driver

Table 2-3 **Edge Connector Pin List**

J1 CONNECTOR PIN LIST FOR INTERRUPT STATUS							
			PIN NUMBER	PIN NUMBER			
		OUTPUT (LSTTL DRIVE)			OUTPUT (LSTTL DRIVE)		
	INPUT (LSTTL LOADS)					INPUT (LSTTL LOADS)	
SIGNAL							SIGNAL
HELP 0*		10[a]	2	1	OUT		GROUND
HELP 1*		10[a]	4	3	OUT		GROUND
HELP 2*		10[a]	6	5	OUT		GROUND
GROUP*		10[a]	8	7	OUT		GROUND
TEST		0	10	9	OUT		GROUND

* Active Low-level Logic [a] Open-Collector Driver

Table 2-4: **Connector J1 Pin List for Interrupt Status**

J2 CONNECTOR PIN LIST FOR COUNTER/TIMER CHANNELS							
			PIN NUMBER	PIN NUMBER			
		OUTPUT (LSTTL DRIVE)			OUTPUT (LSTTL DRIVE)		
	INPUT (LSTTL LOADS)					INPUT (LSTTL LOADS)	
SIGNAL							SIGNAL
GATE 0*	5		2	1	OUT		GROUND
CLOCK 0*	5		4	3	OUT		GROUND
OUT 0*		55	6	5	OUT		GROUND
GATE 1*	5		8	7	OUT		GROUND
CLOCK 1*	5		10	9	OUT		GROUND
OUT 1*		55	12	11	OUT		GROUND
GATE 2*	5		14	13	OUT		GROUND
CLOCK 2*	5		16	15	OUT		GROUND
OUT 2*		55	18	17	OUT		GROUND
SPARE 1*	5		20	19	OUT		GROUND
SPARE 0*	5		22	21	OUT		GROUND
+5V		OUT	24	23	OUT		GROUND
+5V		OUT	26	25	OUT		GROUND

* Active Low-level Logic

NOTE: +5V outputs each have 1.0Ω, ¼W series resistors for connector protection.

Table 2-5 **Connector J2 Pin List for Counter/Timer Channels**

RECOMMENDED OPERATING LIMITS				ABSOLUTE NON-OPERATING LIMITS		
PARAMETER	MIN	TYP	MAX	MIN	MAX	UNITS
Free Air Temperature	0	25	55	-40	75	°C
Humidity *	5		95	0	100	%RH ·

* Non condensing

TABLE 2-6: 7308 ENVIRONMENTAL SPECIFICATIONS

Section 3

How the 7308 Functions
(Operation and Programming)

The 7308, in its function as a fully programmable, multichannel counter/timer card, operates as part of the STD BUS card rack system. Uses for the 7308 include:

- Event counting.
- Square-wave and marker pulse waveform generation,
- Time-interval measurements.
- One-shot simulation with hardware and software triggering and retriggering.
- Precisely timed interrupts.

The main elements of the 7308 are three 16-bit counter/timer channels, a crystal oscillator and tapped clock divider, an 8-input multiplexer for each channel, and programmable logic states, for the clock, gate, and output signals.

Counter/Timer Channels

The three channels of the 7308 are similar in logical organization (see Fig. 3-1 for single channel block diagram).

Each channel consists of the following elements:

a. 16-bit counter (down counter)
b. 16-bit channel read and load registers
c. Mode control register (6 bits)
d. Channel control port (8 bits)
e. 8-input multiplexer and clock polarity control
f. Gate-enable and polarity control
g. Output-polarity control circuit
h. Interrupt control circuit

Note in Figure 3-1 that items a, b, and c are part of the 8253-5 counter/timer device.

The 8-bit input and output ports, which the program uses to load data into the counter, read the counter's contents, and read the state of the counter's

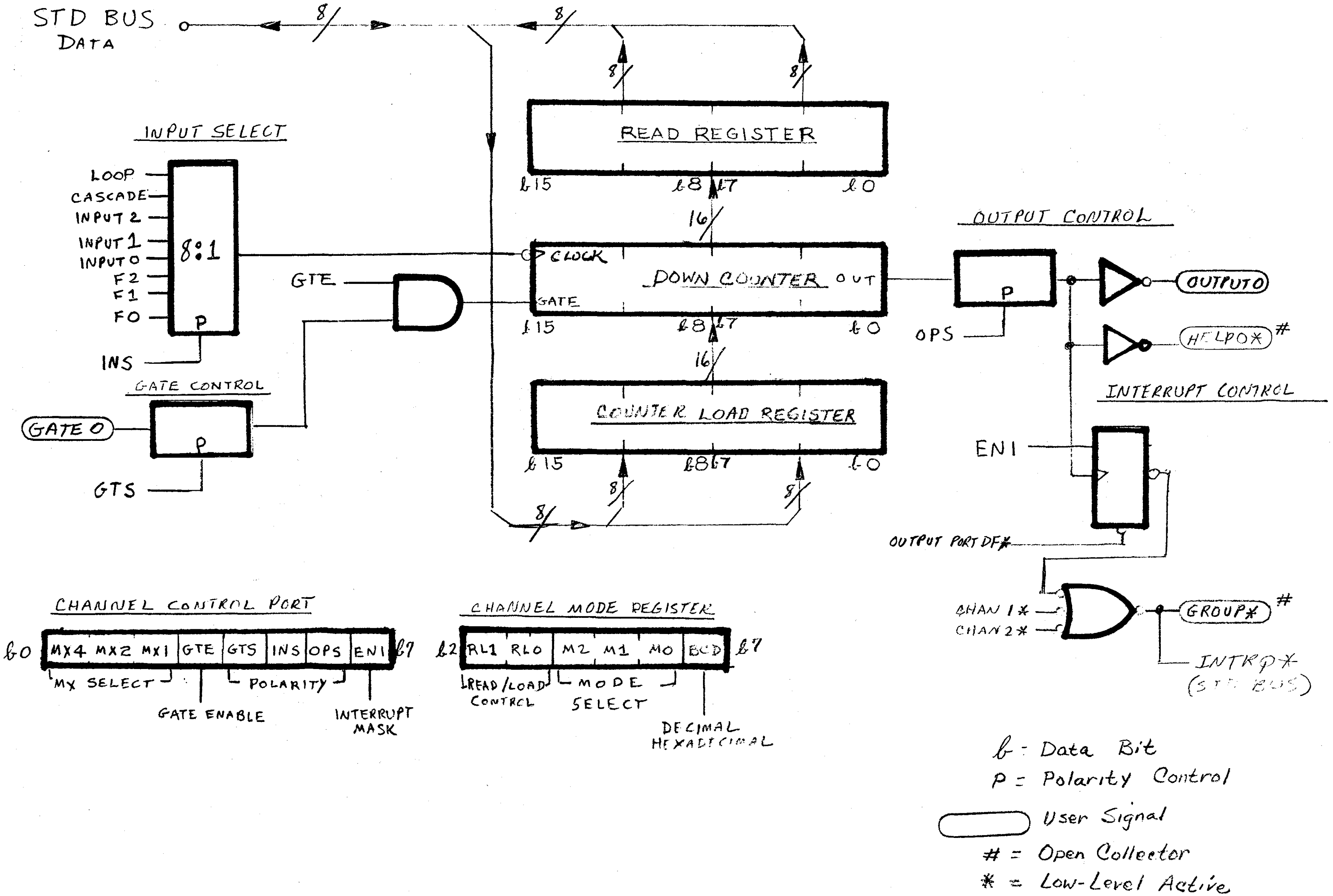


Figure 3-1. Logical Organization of 7308's Counter/Timer Channel 0
(Typical of all channels)

outputs and the interrupt system, are implied but not shown in Fig. 3-1.

Down Counter, Load Register, and Read Register. The central element in each channel is the 16-bit down counter. The remaining logic elements control and monitor the down counter and set its mode of operation.

You can preset the down counter to any value in the range of 0000-FFFF (hexadecimal) or 0000-9999 (decimal). Once all of the preset input conditions are met, the down counter decrements with each transition at its CLOCK input until terminal count is reached.

Terminal count is defined as the lowest number obtainable when down-counting. Depending on mode, this may be 0000, 0001, or 0002 (decimal or hexadecimal). When terminal count is reached, the OUTPUT circuit responds according to its mode and output-polarity program selections.

You can preload the down counter to any desired value through the load register, which is a 16-bit latch. Note that the program can only write to the load register, not to the down counter itself.

The preload value written to the load register does not enter the down counter immediately. Instead, it is clocked in by the first active transition at the CLOCK input (depending on mode, the GATE may or may not be required to be active to preload the down counter).

The preload value stays in the load register at all times. Depending on mode, this value may be automatically reloaded into the down counter either at terminal count, or at an active GATE transition (when GATE is used as a count trigger, as in the one-shot mode).

The down counter's read register is a transparent latch. It can operate in either the transparent mode, in which case the latch's outputs track exactly with the down counter's outputs and no latching action occurs, or it can synchronously latch and hold a dynamic count upon program command. In latch mode, internal gating synchronizes the read register and the down counter, so that the read register does not latch until the down counter is stable. Thus, a misread of the down counter from the ripple effect of carry propagation across the 16 bits cannot occur within the clock rate specification of the 7308 (DC to 2.5 MHz.).

IMPORTANT: Unless you are certain that a CLOCK transition cannot occur while the channel's count is being read by the program, we recommend the read register's latched mode for normal operation.

Data Input Ports and Output Ports. The 7308's I/O ports that read and load data to the counter's channels are listed in table 3-1. Note that these addresses reflect the as-shipped mapping of the card. Section 2 shows how the mapping can be changed to any group of eight sequential I/O port addresses.

(Table 3-1: Counter/Timer's Read/Load Ports)

Mode Control Register. Each channel has a 6-bit mode control register that selects the following parameters:

- One of six operating modes.
- One of three read/load formats for the data ports.
- Transparent or latched data readback.
- Binary or BCD (hexadecimal or decimal) down-counting.

The data ports described above are 8 bits wide, but each counter/timer channel is 16 bits wide. Consequently, a protocol must be established to handle the most significant or least significant 8 bits in each read or load operation. The mode register performs this function. It also controls the transparent or latched characteristics of the read register and establishes the basic operating mode for the channel (single count, one-shot, marker generator, etc.).

Detailed programming instructions for the mode control register are given in section 4.

COUNTER/TIMER	READ DATA	LOAD DATA
Channel 0	Input port D8	Output port D8
Channel 1	Input port D9	Output port D9
Channel 2	Input port DA	Output port DA

Table 3-1 Counter/Timer's Read/Load Port

Channel Control Port. Each channel has an 8-bit control port that selects the following parameters:

- One of eight CLOCK sources:

  0. Internal clock F0 = 0.500µs
  1. Internal clock F1 = 8.000µs } As Shipped
  2. Internal clock F2 = 128.0µs
  3. User CLOCK bus 0
  4. User CLOCK bus 1
  5. User CLOCK bus 2
  6. Cascade (output from adjacent channel)
  7. Loop control (input port strobe)

- Program controlled GATE
- State of the GATE, CLOCK, and OUTPUT user signals
- Channel-interrupt mask enable/disable

You can select channel clocking: from one of three crystal-controlled internal time references for time delays or waveform generation; from one of three user inputs for event counting; or from the adjacent channel (channel 0 drives channel 1, channel 1 drives channel 2, and channel 2 drives channel 0). Alternately, the channel can be clocked by the program. (Fig. 3-2-A)

The program clocks the channel by reading from an input port according to Table 3-2.

(Table 3-2 Input Port Clocking of Channels)

This feature allows the program to decrement a channel for loop control or limit control of programmed functions. The act of reading the input port causes the channel to down-count. The data read is irrelevant.

Figure 3-2-B shows the channel control port's counter-gate control circuit.

(Figure 3-2-B)

Interrupt System

The interrupt system (Fig. 3-2-C) is typical of all three channels. In an interrupt environment, you can use the 7308 in several ways:

- Stand-alone with polled response, in which the 7308's interrupt request output is wireOR'd to one of the processor's interrupt inputs (INTRQ*or NMIRQ* - See Appendix A) by plugging the card into the card rack. When the processor responds to an interrupt, the program must poll (read) the status of the 7308, along with other

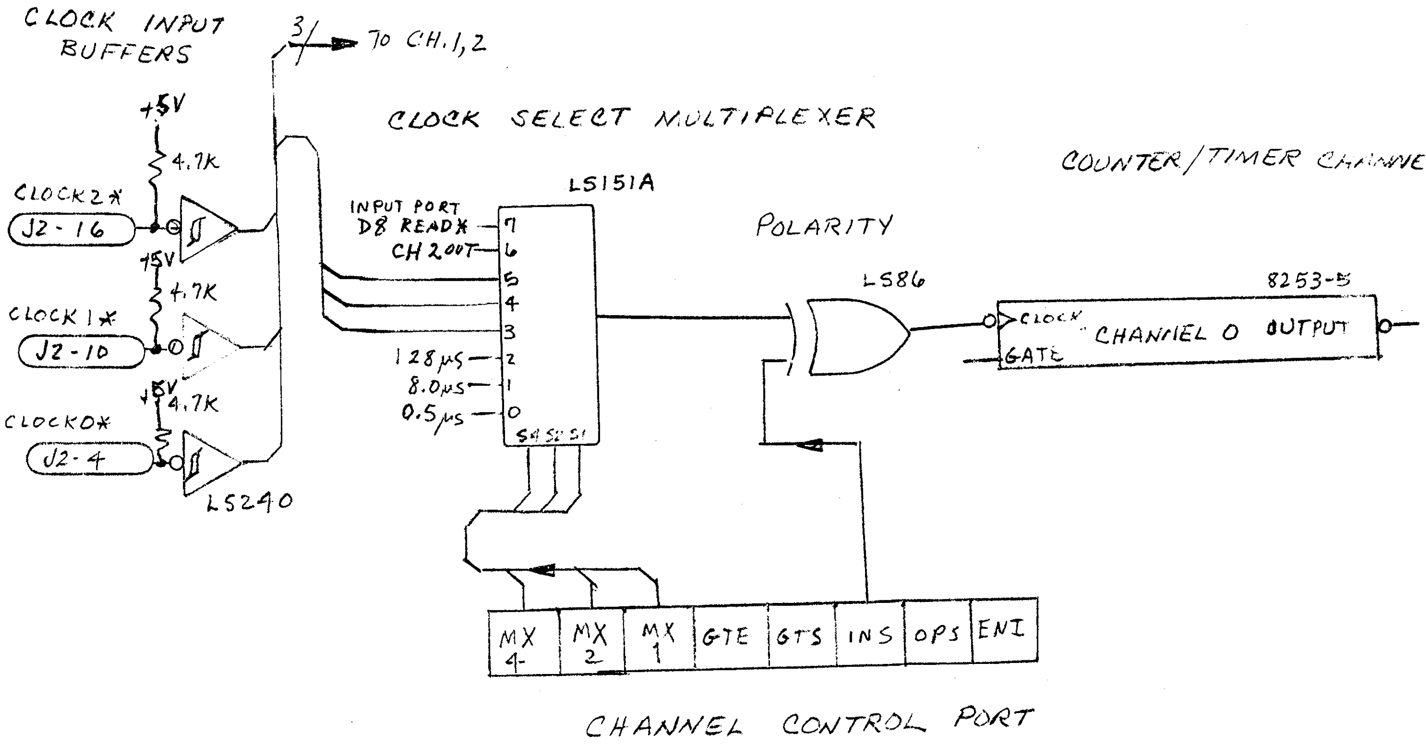


Figure 3-2-A Clock Control Circuit for Channel 0
(Typical of each channel.)

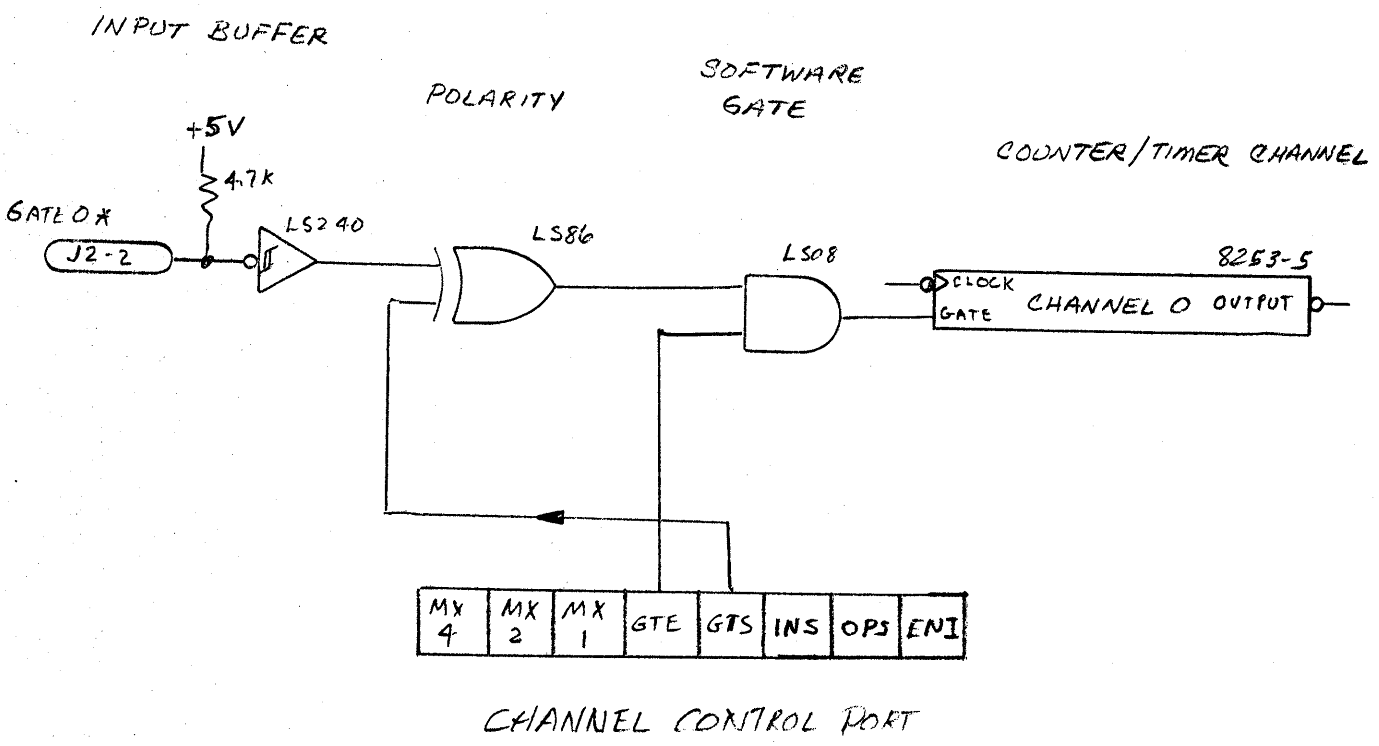


Figure 3-2-B. Gate Control Circuit for Channel 0
(Typical of each channel)

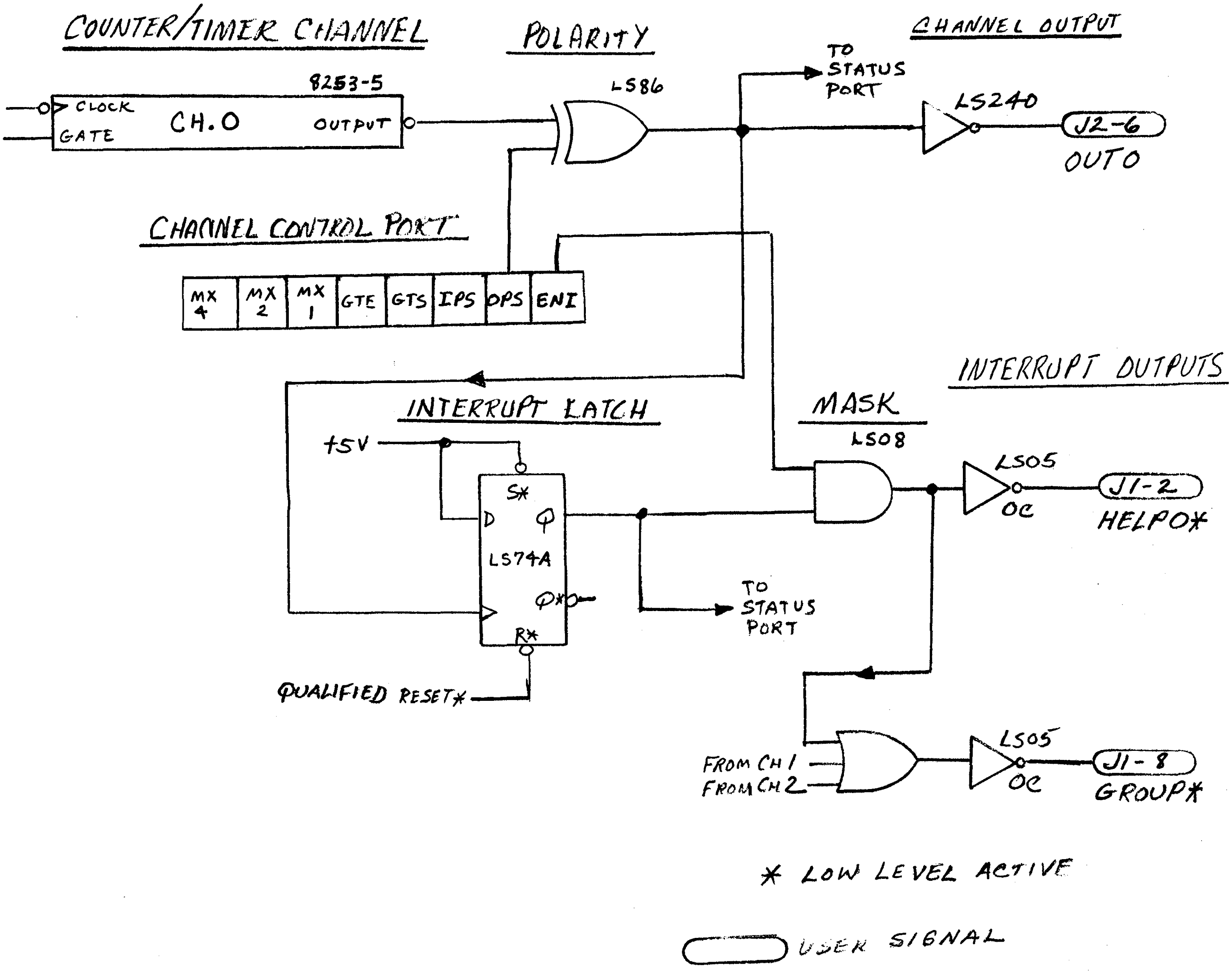


Figure 3-2:C Channel 0 Interrupt Circuit for the 7308 (typical of all channels)

COUNTER/TIMER	CLOCKED BY
Channel 0	Input port DC
Channel 1	Input port DD
Channel 2	Input port DE

Table 3-2. Input Port Clocking of Channels

interrupt-causing cards, to see which one has the active request; if more than one is active, the program assigns priority.

- Vectoring with the 7320 Priority Interrupt Controller (PIC), in which the PIC card or equivalent can be driven by the 7308's interrupt expansion connector J1. Each channel's HELP* output signal is available at J1 to drive one of the 7320's HELP* request inputs. Interrupt control and monitoring facilities are available on the 7320, along with automatic priority determination (according to which of the eight HELP inputs is driven) and a vector generator. When the processor responds to the interrupt, the 7320 releases a 1-byte vector or 8080-compatible restart instruction, which leads the processor directly to the instruction sequences that service the active requestor. The 7320 also has card-level priority circuitry, making the entire operation, from 7308-request to processor response, automatic. This is the fastest technique.

- Combined vectoring and polling, in which the 7308's interrupt expansion connector J1 provides a single GROUP* interrupt signal that is active when any one of the 7308's interrupt latches is set. This signal can drive one of the 7320's HELP inputs. The 7320 responds by vectoring the processor to a program sequence that services the 7308 as a single entity. At this point, additional instructions are needed to determine which channel caused the interrupt, and to assign priority if more than one channel is active. Thus, this technique combines high-speed vectoring with low-cost polling, to achieve a cost-effective solution in which a large number of interrupting cards are used.

The 7308's onboard interrupt system consists of:

- Interrupt latches: Each channel's output clocks a latch, which is edge sensitive (relative to the channel output signal at J1). Since the polarity of the output signal is programmable, the interrupt latch can be programmed to respond to the active-going or inactive-going transition at the channel output. A latch is necessary to

* Low level active

two reasons; (1) the channel's output signal can be active for as little as 400ns, which is much too fast to be read by the processor, and (2) the processor may be too busy to respond to the interrupt immediately. Note that the latches can be selectively reset by the program, and that this function normally occurs as soon the processor has determined the source of the interrupt.

- Mask gates. Each channel's mask gate is controlled by bit 0 of the channel control port (ENI bit). When ENI = 1, the corresponding interrupt request latch is enabled, to drive the processor's interrupt request, if the latch is set. If ENI = 0, the channel cannot interrupt the processor. Since the channel control port is reset (by the processor's SYSRESET* output) at power-on, the mask gates must be enabled by the program, or the 7308 cannot generate an interrupt.

- Status port. By using input port DF, the program reads the states of the interrupt request latches, the channel outputs, and the two uncommitted, general-purpose TTL input lines. When performing an interrupt polling operation, the processor reads the states of the status port's bits 0, 1, 2, which correspond to the three interrupt request latches (See Table 3-13 further on). IRQn = 1 means the corresponding interrupt latch is set.

- Interrupt latch reset port. You can use output port DF to selectively reset any or all of the three interrupt request latches. Any of bits 0, 1, 2 set to 1 when writing to output port DF, resets the corresponding latch.
  IMPORTANT: The usual technique for polling and resetting the interrupt latches is to read input port DF, then immediately write the same data back to output port DF. The program is then aware of any latches set at the time of the read, and only those latches are reset by the write. Thus, an interrupt that occured during the execution time of the read and write instructions is not lost -- it will be available the next time the status port is read.

- Interrupt request signals. The 7308 can generate the following interrupt output signals;

- INTRQ* (STD BUS maskable processor interrupt) - an open-collector output that is connected by jumper when the 7308 is shipped.
- NMIRQ* (STD BUS nonmaskable interrupt), which can be selected in place of INTRQ* above (see Appendix A).
- HELP0*, HELP1*, HELP2* - open-collector output signals used with the 7320 Priority Interrupt Card or equivalent to request a different interrupt vector for each of the 7308's channels.
- GROUP* - a single open-collector output used with the 7320 card to reqeust a single interrupt vector for the whole 7308; independently GROUP = HELP0 + HELP1 + HELP2.

For interrupt, and output-signal strapping options, see Appendix A.

Mode Control Registers: Output Port DB

Each of the three 16-bit counter/timer channels has an associated 6-bit mode control register. You can program different set of operating mode selections for each channel, allowing the channels to operate independently with no interaction.

The mode control registers select the following parameters:

- One of six operating modes
- One of three read/load formats for counter data
- Transparent or latched data readback
- Binary or BCD (hexadecimal or decimal) down-counting

The mode control registers are located inside the 8253-5 device. The program uses the single mode control port (output port DB as shipped) to write into the mode control registers. Two of the port's output bits (b6 and b7) select one of the three channels and the remaining six output bits are copied into the corresponding mode control register.

IMPORTANT: Bits 0-3 of each channel's mode control register can be changed only one time during operation in any given mode. Writing to the mode control register immediately cancels the channel's operation; this occurs even if identical bit states are recopied to bits 0-3 of the mode control register. After a write to the mode control register, channel operation cannot resume until a preset value is written to the channel's load register.

There is a single exception to this rule: The channel's read reqister can be changed from transparent mode to latched mode (RL bits b4 and b5 can be changed from nonzero to zero; see "Latched Read Mode") without cancelling channel operation, providing no other mode control register bits change.

The format and function of the mode control port's bits are shown in Table 3-3. They are defined in detail in the discussion that follows.
(Table 3-3 Mode Control Port Format)

Bits SC0 and SC1: Select Channel. These two bits are used to address one of the three mode control register, according to Table 3-4.
(Table 3-4 Channel Select Bits)

Bits RLØ and RL1: Read/Load Format. These two bits are used to select one of three techniques for communicating with the 16-bit channels from an 8-bit data bus; the first combination of the RL bits allows error-free reading of a high-speed count by a much slower microprocessor (See Table 3-5).
(Table 3-5 Read/Load Bit Function)

Data Read/Load Ports. Output ports D8, D9 and DA (addresses selected when the 7308 is shipped) are used to write to the load registers for channels 0, 1, and 2, respectively, to preload an initial count after the mode is selected. Input ports D8, D9, and DA are used to monitor the progress of a count or to read a final tally for event counting from the read registers.

Since the counter/timer channels are 16 bits each and the 7308's I/O ports are 8 bits wide, the program requires control over which half of a channel if being read from or written to.

BIT	b7	b6	b5	b4	b3	b2	b1	b0
FUNCTION	SC1	SC0	RL1	RL0	M2	M1	M0	BCD
	Address		Bits to Mode Control Register					

Table 3-3. Mode Control Port Format

SC1	SC0	Select Channel
0	0	Channel 0
0	1	Channel 1
1	0	Channel 2
1	1	Illegal

Table 3-4 Channel Select Bit

RL1	RL0	Read/Load Function
0	0	Latched read mode
1	0	Read/Load bits 0-7
0	1	Read/Load bits 8-15
1	1	Read/Load bits 0-7, then bits 8-15

Table 3-5 Read/Load Bit Function

The RL bits provide this control;

- RL = 10 causes only the most significant 8 bits to be read back or written into.
- RL = 01 selects only the least significant bits.
- RL = 11 selects a sequence: least significant bits followed by most significant bits.

IMPORTANT: When RL = 11, the program must complete the sequence of two writes, unless you cancel the sequence by selecting a new mode.

As noted before, writing to the mode control register forces the channel into the idle condition (except writing RL = 00). Channel operation cannot begin again until it has been preloaded to an initial count. Therefore, a write to the mode control port must be followed by one or two writes to the corresponding channel load port (output port D8, D9 or DA). The required number of writes is established by the value of the RL bits. If RL = 01 or 10, channel operation is enabled after one write. If RL = 11, channel operation is enabled after two writes. Thus, the number of CLOCK pulses received by the channel, before the second write when RL = 11, is irrelevant.

NOTE: The system may receive a push-button reset while the program is in the act of writing to the 8253-5, and the two data-preload write operations after setting RL = 11 may be interrupted. The 8253-5 device is unaffected by reset, but the next write to the channel's mode control register cancels the previous selection and effectively resets the channel's control logic.

Latched Read Mode. The program can read the content of any counter/timer channel in one of two ways:

* Transparent mode, in which the Data Read Ports (input ports D8, D9, DA) see the real-time counter content.

* Latched Mode, in which the program strobes an intermediate latch (the Read Register) prior to reading the data ports.

Transparent mode is selected when the RL bits are 01, 10, or 11. This is necessarily the initial mode for reading, since the channel can't be preloaded to an initial count value without selecting one of these combinations. Once set to 01, 10, or 11, the channel remembers the selection. The program can then change to RL = 00 and select the latched mode.

Writing RL=00 to the Mode Control Register creates a dynamic read strobe that synchronously latches the channel's count into the associated Read Register. The next time the program reads the count, it will actually read the output of the latched Read Register. This avoids the possibility of read errors introduced by asychronous carry propagation across the 16-bit counter during the act of reading. The Read Register will remain latched while the program reads one byte or two bytes (according to the previous state of the RL bits). After being read, the Read Register returns to the transparent mode. It may be restrobed at any time by again writing the RL bits = 00.

When in transparent mode, the real-time value of the counter is visible to the program. Thus, if the processor is halted by a logic analyzer while in the act of reading one of the counter/timer channels, each CLOCK transition will cause a new count to appear on the STD Data Bus; the counting operation can be monitored directly.

> IMPORTANT: If you use the transparent read mode while CLOCK transitions are possible, you should repeat the read operation a second time in the program and compare the values arithmetically. If the values are the same after both reads, it can be assumed that no carry propagation error has occurred.

The transparent mode is selected when the RL bits are 01, 10, or 11. This is necessarily the initial mode for reading, since the channel can't be preloaded to an initial count value without selecting one of these combinations. Once set to 01, 10, or 11, the channel remembers the selection. The program can then change to RL = 00 and select the latched mode.

The latched mode creates a dynamic read strobe when the data read ports are read by the program. The 8253-5 synchronously strobes the instantaneous data content of the channel being read into an intermediate holding latch (The read register). Then the program reads the output of this latch. The synchronous strobe prevents read errors caused by carry propagation across the 16-bit channel. Thereafter, each read operation restrobes the latch, allowing new data to be read by the program.

Having once selected the latched read mode, it is impossible to return to the transparent mode without interrupting the channel's operation by reselecting the same mode or a new mode, followed by a new channel data preload.

In the transparent mode, the real-time value of the counter is visible to the program. Thus, if the processor is halted by a logic analyzer while in the act of reading one of the channels, each CLOCK transition causes a new count to appear visibly on the STD data bus; the counting operation can be monitored directly.

> IMPORTANT: If you use the transparent read mode while CLOCK transitions are possible, we advise you to repeat the read operation twice and then arithmetically compare the values, to be certain that no carry propagation error has occurred in the first read (the values should be the same after both reads).

M2	M1	M0	Mode Select
0	0	0	Single count or time delay
0	0	1	Single count (external trigger), or retriggerable 1-shot
x	1	0	Multiple count/auto reload, or rate generator
x	1	1	Square wave generator
1	0	0	Delayed strobe - software trigger
1	0	1	Delayed strobe - hardware trigger

Table 3-6 Channel Operating Mode Codes

We use decimal and hexadecimal notations for explanations in this manual. Table 3-7 shows how the count proceeds according to the state of the BDC bit in the mode control register for each channel:
(Table 3-7 Example of BCD Bit Count Sequence)

Note that once the BCD bit has been set, or reset, it cannot be changed without interrupting the mode in progress. Any write to the mode control register (except setting the RL bits = 00) stops the count and requires a new channel data preload. The decimal/hexadecimal conversion to binary is given in Table 3-8.
(Table 3-8 Decimal/Hexadecimal/Binary Conversion Table)

Sequence of Operations in Each Mode

This subsection describes the waveforms and the sequence of operations in each of the six operating modes for the counter/timer channels. (Figures 3-3 through 3-8).

Since all user input/output signals have program-selectable polarity, the illustrated waveforms are those at the 8253-5 device pins. You can use these waveforms directly for card troubleshooting with an oscilloscope.

> NOTE: The signals at user interface connector J2 will match the waveforms shown on the following pages, if you program the channel control register as follows:
>
> GATE state: GTS = 1
> CLOCK state: INS = 0 (See Table 3-14)
> OUTPUT state: OPS = 1
>
> Programming the opposite state for these bits causes or requires the corresponding signal to be inverted at J2.

In all modes, the channel's output assumes a defined state immediately after the rising edge of the WR* signal, when the mode control register (Output Port DB) is loaded with the new mode selection for the channel. An output transition may occur at this time, depending on the output state remaining from the previous mode. You must take care not to inadvertently generate an interrupt at this time. You can prevent it by setting the channel control port's interrupt enable bit ENI = 0, until the new

	COUNT SEQUENCE	
BCD MODE BIT	BCD=0	BCD=1
	Hex	Decimal
	0500	0500
	04FF	0499
	04FE	0498
	.	.
	.	.
	0011	0011
	0010	0010
	000F	0009
	.	.
	.	.
	0001	0001
	0000	0000
	FFFF	9999
	FFFE	9998

Table 3-7 Example of BCD Bit Count Sequence

DECIMAL	HEXADECIMAL	BINARY
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
	A	1010
	B	1011
	C	1100
	D	1101
	E	1110
	F	1111

Table 3-8 Decimal/Hexadecimal/Binary Conversion Table

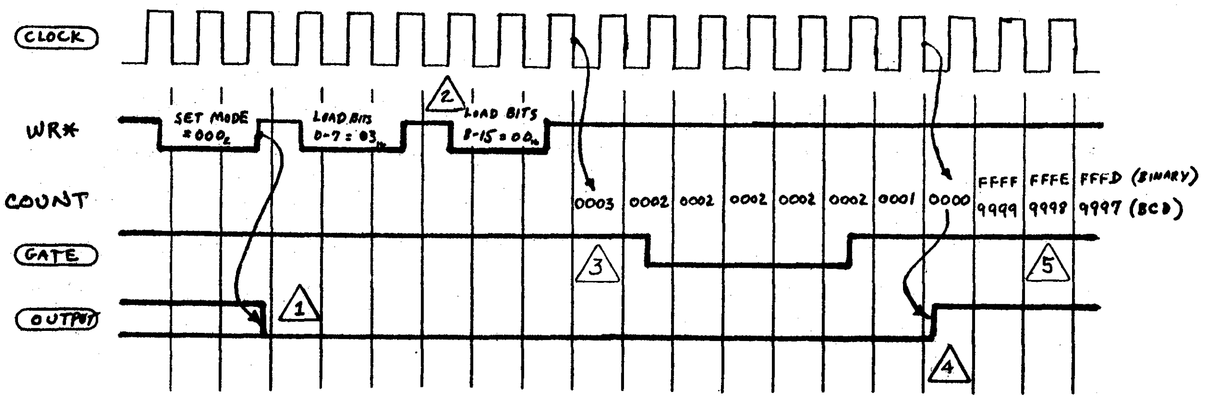
mode is selected and the output state is known.

When attempting to synchronize more than one channel, it is important to disable the CLOCK inputs to the channels (select multiplexer position 7; MX4=MX2=MX1=1) until all channels are preloaded. If counting is inhibited by disabling the GATE alone, the channels may fail to track by one count due to the Load Register/Counter transfer mechanism. After being preloaded, any or all channels may be inhibited by the GATE.

In describing the modes, we use these conventions:

- GATE, CLOCK, and OUTPUT are capitalized when they refer to one of the channel's input/output signals appearing at either the 8253-5 device pins or user interface connector J2.
- A signal available to the user is denoted by ⬭
- Low-level active logic is denoted by an asterisk (*).

(Figures 3-3 through 3-8)



Mode 0: Single Count, Single Delay, Software triggered one-shot

△1 OUTPUT goes active (low) with the rising edge of WR* as soon as Mode 0 is selected; counting stops regardless of CLOCK or GATE activity, and the count remaining from the previous mode is frozen in the counter. As a software triggered one-shot, OUTPUT goes active when Mode 0 is first selected and stays active until Terminal Count. Thereafter, as long as Mode 0 remains selected, OUTPUT goes active (1-shot is retriggered by software) when the first byte of the preload value is written in the Load Register.

△2 Counting cannot resume until one or two bytes of preload data (initial value) are written to the Load Register. The state of the Mode Control Register's RL bits determine whether one byte or two bytes are needed. CLOCK transitions occuring between write cycles are irrelevant until the channel is preloaded.

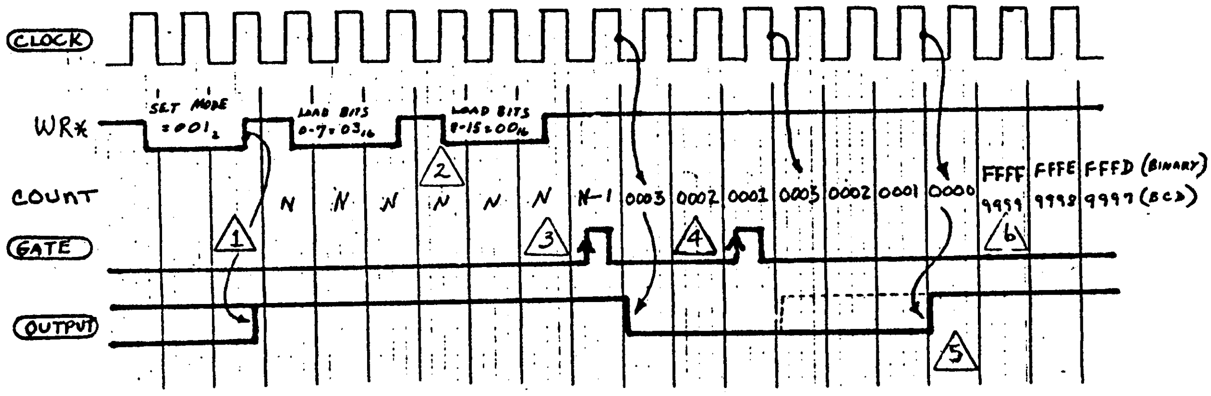
△3 After the preload operation, the first CLOCK falling edge with GATE active (high) transfers the preload value from the Load Register to the Counter. Subsequent CLOCK edges decrement the count. GATE may be used to modulate the count if desired. Counting can only occur while GATE is active. For an initial count of 0001, GATE has no effect on OUTPUT. If GATE is used to stop counting, GATE must be assured of going inactive at least two full clock cycles before Terminal Count to prevent OUTPUT from going inactive. GATE can stop the counter content at a value of 0001, but cannot prevent OUTPUT from going inactive with the next CLOCK edge.

△4 Terminal Count for Mode 0 is 0000 (decimal or hexadecimal). Immediately after the CLOCK edge causing Terminal Count, OUTPUT goes inactive (high) and remains inactive until a new mode is selected.

△5 After Terminal Count, the channel continues to decrement past 0000 according to the state of the BCD mode bit. The preload value is never reloaded to the counter. The only way to exit from Mode 0 is to write a new mode selection into the channel's Mode Control Register.

Figure 3-3. Mode 0 Waveforms

NOTE: The waveforms above show an example in which the Load Register is preset to the initial value 0003 (decimal or hexadecimal). WR* is an STD BUS signal (pin 31) which is active with IORQ* (pin 33) and IOEXP* (pin 35) during an output port write operation.



Mode 1: Hardware Triggered One-shot or Count (Retriggerable)

△1 OUTPUT goes inactive (high) with the rising edge of WR* when Mode 1 is selected; counting stops regardless of CLOCK or GATE activity, and the count remaing from the previous mode (N) is frozen in the counter.

Caution: Disable GATE by setting GTE=0 in the program prior selecting Mode 1, and do not re-enable GATE until the preload value has been loaded. The prevents inadvertent OUTPUT activity.

△2 Counting cannot resume until one or two bytes of preload date (initial value) are written to the Load Register. The state of the Mode Control Register's RL bits determine whether one byte or two bytes are needed. CLOCK transitions occurring between write cycles are irrelevant until the channel is preloaded.

Once the channel's initial value is set, CLOCK transitions decrement the counter regardless of the state of the GATE. However, counting proceeds from the value N (to N-1, N-2,...) left in the counter from the previous mode.

△3 GATE's rising edge triggers the Mode 1 one-shot action. The next CLOCK falling edge after the GATE rising edge transfers the preload value from the Load Register into the counter, and sets OUTPUT active (low). After triggering the count, GATE may remain high or fall low without affecting the one-shot action.

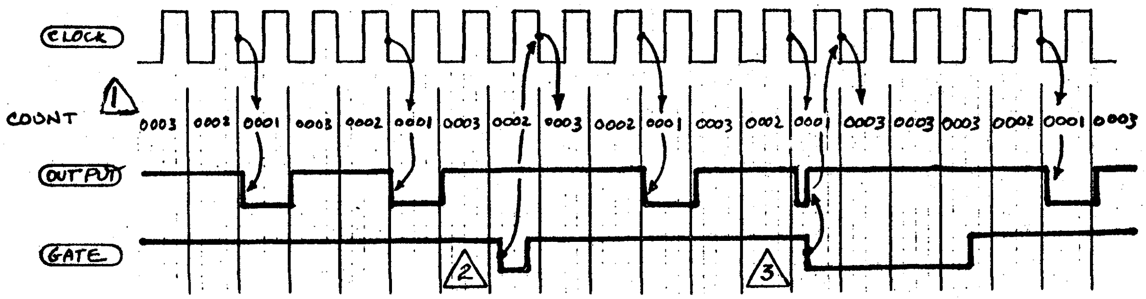
△4 If another GATE rising edge trigger occurs before Terminal Count, the one-shot is retriggered; the preload value is again transferred from the Load Register to the counter on the next CLOCK falling edge; counting proceeds from the preload value; and OUTPUT remains active (low) without glitching.

△5 Terminal Count for Mode 1 is 0000 (decimal or hexadecimal). Immediately after the CLOCK edge causing Terminal Count, OUTPUT goes inactive (high) and remains inactive until a new GATE trigger occurs or a new mode is selected.

△6 After Terminal Count, the channel continues to decrement past 0000 according to the state of the BCD mode bit. Decrementing continues freely with each CLOCK edge until the next GATE trigger.

FIGURE 3-4: MODE 1 WAVEFORMS

NOTE: The waveforms above show an example in which the Load Register is present to the initial value 0003 (decimal or hexadecimal). WR* is an STD BUS signal (pin 31) which is active with IORQ* (pin 33) and IOEXP (pin 35) during an output port write operation.



Mode 2: Rate Generator or Divide-by-N Counter with AutoReload

1 OUTPUT goes inactive (high) with the rising edge of WR* when Mode 2 is selected; counting stops regardless of CLOCK or GATE activity, and the count remaining from the previous mode is frozen in the counter. In the waveforms above, 0003 (decimal or hexadecimal) has already been preloaded after the selection of Mode 2 as an example.

If GATE is high after the preload operation, the first CLOCK falling edge transfers the preload value from the Load Register to the Counter, and subsequent CLOCK falling edges decrement the count toward Terminal Count.

Terminal Count for Mode 2 is 0001 (decimal or hexadecimal). The OUTPUT goes active (low) immediately after the CLOCK transition resulting in Terminal Count, and remains active for one clock period. The next CLOCK falling edge automatically transfers the content of the Load Register to the Counter, and returns the OUTPUT to the inactive (high) state.

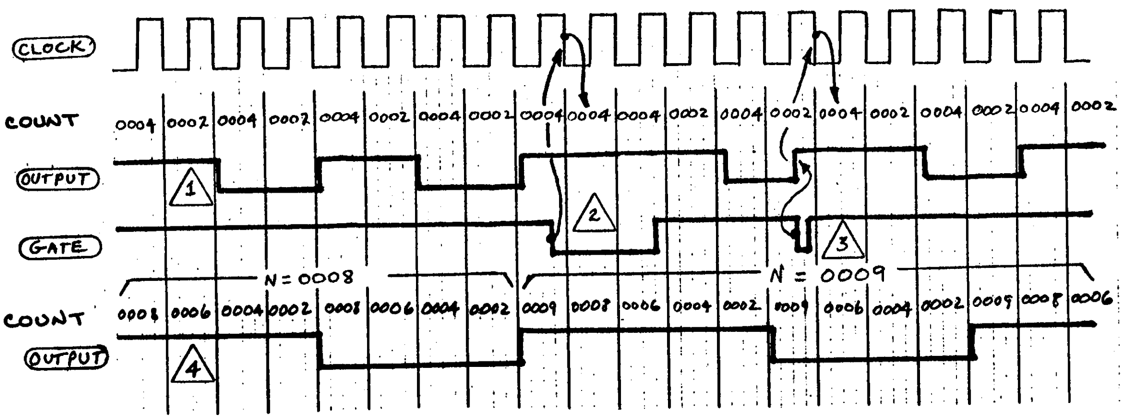
2 GATE may be regarded as either an output reset or synchronization input signal in Mode 2. When high, the channel is enabled to downcount. When low, the next CLOCK falling edge transfers the Load Register to the Counter, restoring the preload value and synchronizing the output with the GATE input. Counting resumes when GATE goes high.

3 If OUTPUT is active when GATE goes low, the output is immediately forced to the inactive state regardless of CLOCK transitions. At the next CLOCK falling edge, the preload value is restored to the counter and counting will resume as soon as GATE is again high.

FIGURE 3-5 : MODE 2 WAVEFORMS

Note: In Mode 2,

Preload value = 0000 is regarded as 10,000 (9999 + 1 decimal or FFFF + 1 hexadecimal). Preload value = 0001 holds the OUTPUT continuously inactive regardless of CLOCK or GATE activity.



See note on next sheet

Mode 3: Square Wave Generator

1 OUTPUT goes inactive (high) with the rising edge of WR* when Mode 3 is selected; counting stops regardless of CLOCK or GATE activity, and the count remaining from the previous mode is frozen in the counter. In the upper three waveforms above, 0004 (decimal or hexadecimal) has already been preloaded after the selection of Mode 3.

If GATE is high after the preload operation, the first CLOCK falling edge transfers the preload value from the Load Register to the Counter, and subsequent CLOCK falling edges decrement the counter toward Terminal Count.

Terminal Count for Mode 3 is 0002 (decimal or hexadecimal); unlike other modes, the OUTPUT state change occurs at the CLOCK falling edge after the edge causing Terminal Count. When the OUTPUT changes state, the Load Register value is automatically restored to the counter and the process repeats, creating an OUTPUT square wave.

Note that as long as the initial count value preloaded to the Load Register is an even number (bit 0 = 0), each CLOCK transition decrements the counter by 2.

2 GATE may be regarded as either an output reset or synchronization input signal in Mode 3. When high, the channel is enabled to downcount. When low, the next CLOCK falling edge transfers the Load Register to the counter, restoring the preload value and synchronizing the OUTPUT with the GATE input. Counting can resume when GATE is high.

3 If OUTPUT is active (low) when GATE goes low, OUTPUT is immediately forced inactive and the counter is reloaded from the Load Register.

4 The second OUTPUT waveform illustrates the difference between an even preload value and an odd value. The even value (0008 in the example) produces a symmetrical square wave output and causes every CLOCK falling edge to decrement the counter by two. The odd value (0009 in the example) produces an asymmetrica square wave with the OUTPUT inactive (high) for one clock period longer than it is active (low). This is achieved by decrementing by one the first time, then by two until the first Terminal count; then by three the first time with the output active, followed by two until the second Terminal Count completes the OUTPUT waveform period.

FIGURE 3-6 : MODE 3 WAVEFORMS

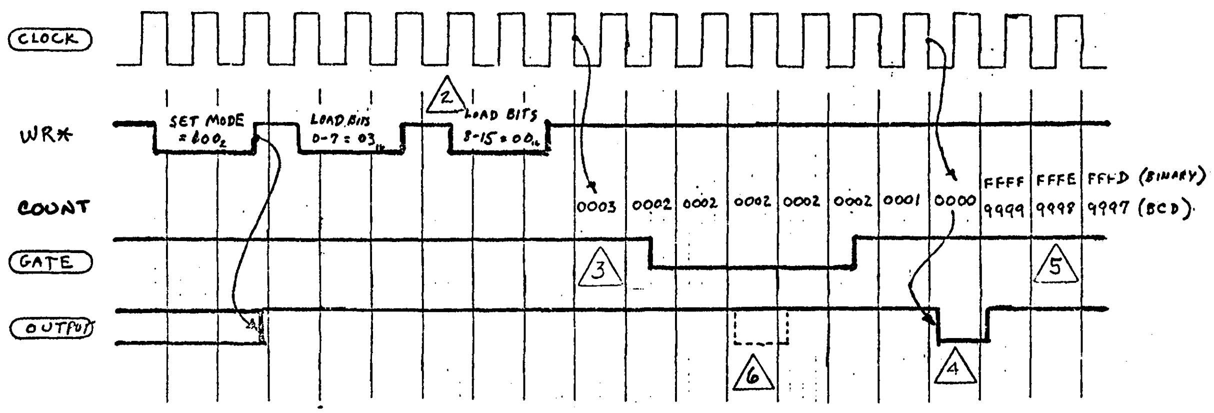
NOTE - Due to internal comparison logic in the 8253 for mode 3, certain initial values preloaded to the Load Register will cause unusual results:

0000 is regarded as 10,000 (9999 + 1 decimal or FFFF + 1 hexadecimal).

0001 is regarded as 10,001 (9999 + 2 decimal or FFFF + 2 hexadecimal).

0002 divides CLOCK by 2

0003 produces a waveform which is inactive for one CLOCK period and active for 10,000 (9999 + 1 decimal or FFFF + 1 hexadecimal) periods.



Mode 4: Software-Triggered Strobe

Mode 4 is identical to Mode 0, except that a single active pulse (strobe) is generated at Terminal Count. The "software trigger" is the preload operatio

△1 OUTPUT goes inactive (high) with the rising edge of WR* as soon as Mode 4 is selected; counting stops regardless of CLOCK or GATE activity. The count remaining from the previous mode is frozen in the counter.

△2 Counting cannot resume until one or two bytes of preload data (the initial value) are written to the Load Register. The state of the Mode Control Register's RL bits determine whether one byte or two bytes are needed. CLOCK transitions occurring between write cycles are irrelevant until the channel is preloaded.

△3 After the preload operation, the first CLOCK falling edge with GATE active transfers the preload value from the Load Register to the Counter. Subsequent CLOCK edges decrement the count. GATE may be used to modulate the count if desired. Counting can only occur while GATE is active.

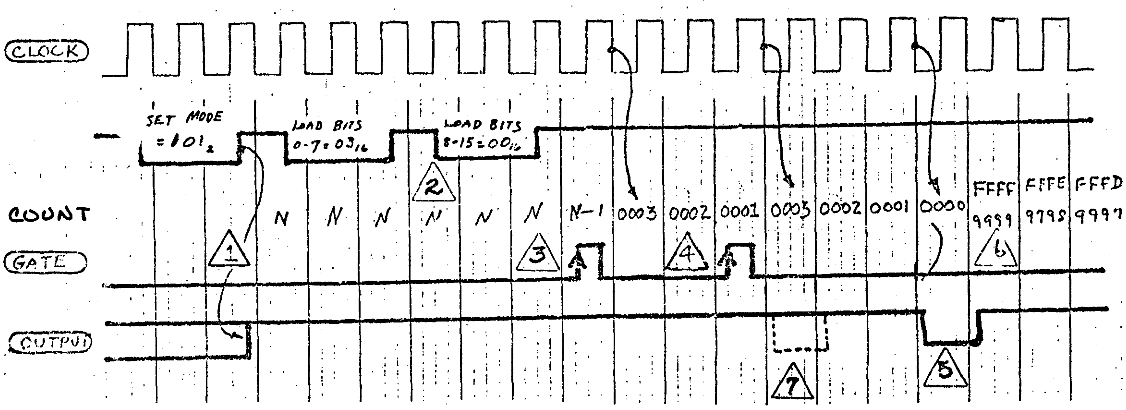
△4 Terminal Count for Mode 4 is 0000 (decimal or hexadecimal). Immediately after the CLOCK falling edge causing Terminal Count, the OUTPUT goes active (low) and remains active until the next CLOCK falling edge. It then goes inactive (high) and remains inactive until a new mode is selected.

Thus the act of preloading the channel in Mode 4 acts as a trigger which initiates a time delay. After a delay of N CLOCK pulses, a strobe pulse is generated at the OUTPUT.

△5 After Terminal Count, the channel continues to decrement past 0000 according to the state of the BCD mode bit. The preload value is never reloaded to the counter. The only way to exit from Mode 4 is to write a new mode selection into the channel's Mode Control Register.

△6 The dotted pulse shows how the OUTPUT would have responded if GATE had not gone inactive in the example.

FIGURE 3-7 : MODE 4 WAVEFORMS



Mode 5: Hardware Triggered Strobe (Retriggerable)

Mode 5 is identical to Mode 1 except that a single active pulse (strobe) is generated at Terminal Count. The "hardware trigger" is the GATE rising edge.

1 OUTPUT goes inactive (high) with the rising edge of WR* when Mode 5 is selected; counting stops regardless of CLOCK and GATE activity, and the count remaining from the previous mode (N) is frozen in the counter. Caution: Disable GATE by setting GTE=0 in the program prior to selecting Mode 5, and do not re-enable GATE until the preload value has been loaded. This prevents inadvertent OUTPUT activity.

2 Counting cannot resume until one or two bytes of preload data (initial value) are written to the Load Register. The state of the Mode Control Register's RL bits determine whether one byte or two bytes are needed. CLOCK transitions occurring between write cycles are irrelevant until the channel is preloaded.

3 GATE's rising edge triggers the delayed strobe. The next CLOCK falling edge after the GATE rising edge transfers the preload value from the Load Register to the counter (OUTPUT remains inactive high). After triggering the count, GATE may remain high or fall low without affecting the count.

4 If another GATE rising edge occurs before Terminal Count, retriggering occurs. The preload value is again transferred from the Load Register to the counter with the next CLOCK falling edge, and downcounting proceeds from the preload value. OUTPUT remains inactive without glitches.

5 Terminal Count for Mode 5 is 0000. Immediately after the CLOCK edge causing Terminal Count, OUTPUT goes active (low) and remains low until the next CLOCK edge. It then goes inactive (high) and remains high until the next GATE trigger or until a new mode is selected.

Thus triggering the channel via GATE produces a delayed strobe pulse.

6 After Terminal Count, the channel continues to decrement past 0000 according to the state of the BCD mode bit. Decrementing continues freely until the next GATE trigger.

7 The dotted pulse shows how the OUTPUT would have responded if the GATE retrigger had not occurred inthe example.

FIGURE 3-8: MODE 5 WAVEFORMS

BIT	b7	b6	b5	b4	b3	b3	b1	b0
FUNCTION	MX4	MX2	MX1	GTE	GTS	INS	OPS	ENI
	Input Multiplexer			Enable	Logic States			Interrupt

Table 3-9 Channel Control Port Bit Format

RANGE	MX4	MX2	MX1	CLOCK Source
0	0	0	0	Clock F0: internal 400 nanosecond rate
1	0	0	1	Clock F1: internal 6.4 microsecond rate.
2	0	1	0	Clock F2: internal 102.4 microsecond rate
3	0	1	1	INPUT 0 Bus
4	1	0	0	INPUT 1 Bus
5	1	0	1	INPUT 2 Bus
6	1	1	0	Cascade: Channel 0 - Channel 2's output Channel 1 - Channel 0's output Channel 2 - Channel 1's output
7	1	1	1	Loop Control: Channel 0 - Input Port DC (Decrement on Channel 1 - Input Port DD Port Read) Channel 2 - Input Port DE

Table 3-10 CLOCK Multiplexer Programming

Channel Control Ports: Output Ports DC, DD, and DE

Each of the three 16-bit counter/timer channels has an associated 8-bit channel control port. These ports give the program control over each channel's CLOCK input, interrupt mask, and states of the GATE, CLOCK, and OUTPUT signals. The selections are made independently for each channel so that independent operation is achieved unless the channels are cascaded.

Parameters selected by the channel control ports are:

- One of eight CLOCK sources.
- Channel count enable/disable (programmable GATE).
- Active state of the GATE, CLOCK, and OUTPUT signals.
- Interrupt mask enable/disable.

The channel control ports can be written to as often as necessary by the program, and at any time. However, if you program a change in the active logic state at the channel's inputs and output, the channel may trigger, or decrement, or set the interrupt latch. Thus, the order of setting bits in the channel control ports, or of writing to them and to the mode control port, is generally important.

The format and function of the three identical channel control ports are listed in Table 3-9.
(Table 3-9 Channel Control Port Bit Format)

Bits MX4, MX2, and MX1: Input Multiplexer Select. These bits select one of eight CLOCK inputs to the channel. Table 3-10 lists the bit codes for the eight possible CLOCK sources.
(Table 3-10CLOCK Multiplexer Programming)

Clocks F0, F1, and F2 are jumper selected at the 7308's internal clock prescaler. They are chosen for both wide range and convenient overlapping at often used delays in the 10 µs to 20 ms range. Figure 3-9 shows the range of each of the three prescaler taps. If a different prescaler tap, or external clock input is desired, Appendix A describes how to change the multiplexer jumper straps.

Input busses 0, 1, 2 are common to each counter channel. Any channel can select any input, or the same input can drive all channels.

Cascade selects the adjacent channel's output to create a 32-bit or 48-bit counter. Figure 3-9 shows the approximate range of two or three cascaded channels.

Loop control allows the channel to be clocked by the program. Reading one of the input ports (listed in Table 3-2) causes the corresponding channel to decrement once; this has the following applications:

- Limit control of programmed functions, and/or simplified programming Set the maximum number of times a programmed function (such as a loop iteration) is to be performed; then read the loop control input port at the conclusion of each function or loop. With the interrupt system enabled, an interrupt occurs as soon as the function has been performed N times.

- Simplified program module design and hardware/software integration. Pro-Log's Microprocessor User's Guide shows how to use the modular approach to program design. Specifically, single-function subroutines are designed and debugged first; then they are interconnected at system level by jump-to-subroutine instructions.

  The loop control input ports allow you to simulate the effect of a switch closure, or other external event signal within the program. Then you can debug a counter-control subroutine before connecting elements of the system hardware to the processor. Thus, you can reduce the number of unknowns during the design phase of program modules, by using this feature of the 7308 card.

GTE Bit: Gate Enable. This bit gives the 7308 program control over the gate function. With GTE = 0, the 8253-5's GATE input is inhibited (low) regardless of the GATE state at the user interface connector J2. Setting GTE = 1 transfers control of the counter gating function to the user's GATE signal for that channel (Table 3-11)

GTS, INS, and OPS Bits: Logic Polarity Programming. The program has control of the logic state polarity (high active or low active) at all of the counter/timer channel signals at J2, and at the inputs to the interrupt request latches provided for each channel.

These bits can be programmed as required;

- Set input signal polarity to avoid the need for external inverter gates.
- Program input polarity changes for duplicating the effect of input signals, to aid in program module development, to allow both hardware and software clocking, triggering, and gate control, or to facilitate 7308 card self-test routines for field testing.
- Set output signal polarity to match the application's requirement of the 7308's interrupt latches.

Table 3-11 shows the effect of the GTS and IPS program bits on input signals at connector J2. The mode bits are included, since the effect of these input signals on the channel is mode-dependent. Output signal polarity is shown in Table 3-12.

Port Bit Assignments and Programming Aid

Assignments of the 7308's 8-bit input and output ports are listed in Table 3-13. Definitions are given in Table 3-14.

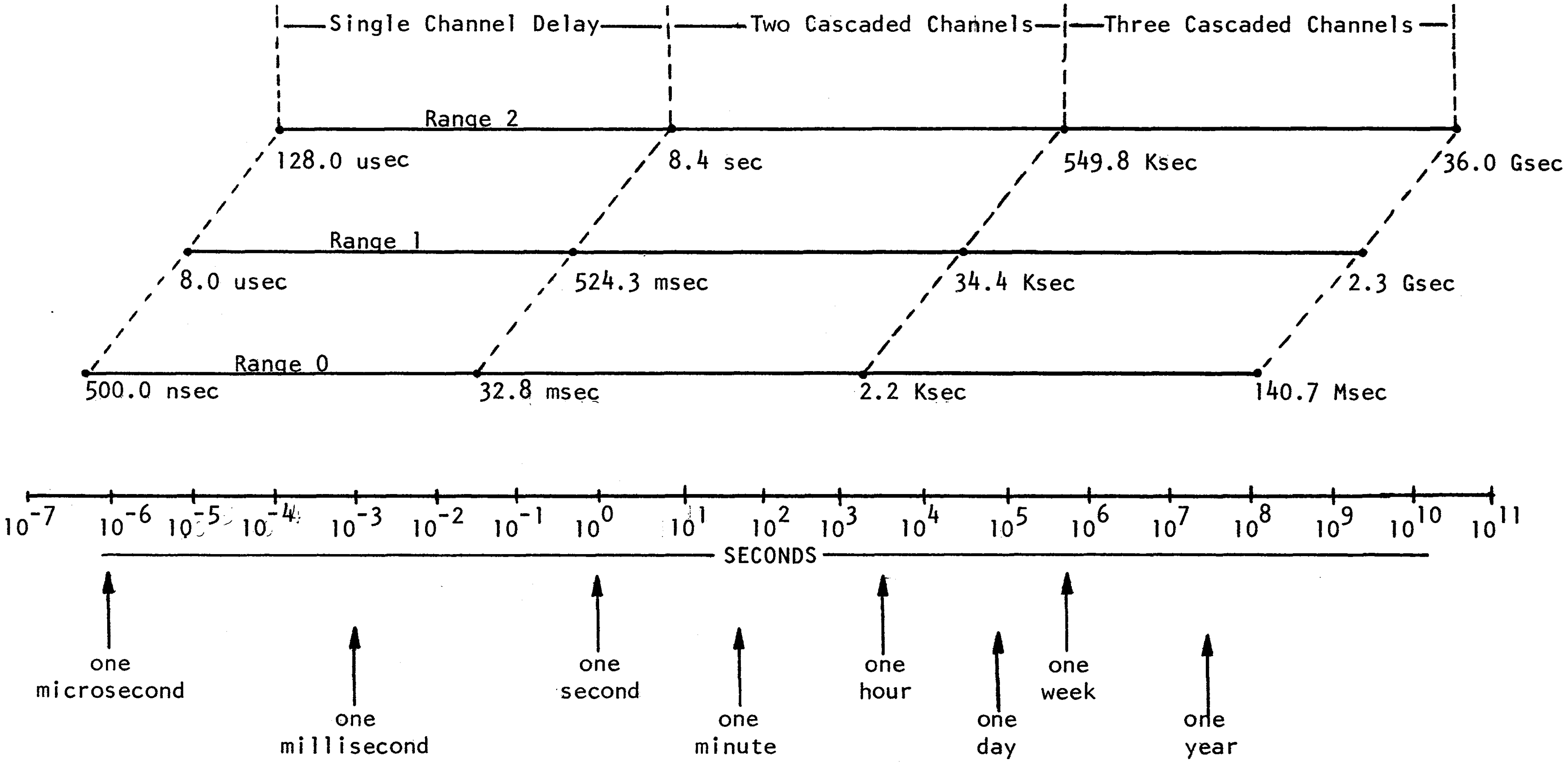


FIGURE 3-9: SINGLE CHANNEL AND CASCADED CHANNEL DELAY RANGES

MODE				CONTROL BIT					
M0	M1	M2	Name	GTE	(Gate Enable)	GTS	(Gate State)	IPS	(Input State)
0	0	0	Single count	0	Disable	x	Disabled	x	No counting
				1	Enable	0	Active = 0	0	Decrement on ↓
						1	Active = 1	1	Decrement on ↑
0	0	1	One-shot	0	Disable	x	Disabled		Note 1
				1	Enable, or trigger if Gate = active	0	Trigger = ↓	0	Decrement on ↓
						1	Trigger = ↑	1	Decrement on ↑
x	1	0	Rate Generator or Divide by N	0	Disable	x	Disabled	x	No counting
				1	Enable	0	Active = 0	0	Decrement on ↓
						1	Active = 1	1	Decrement on ↑
x	1	1	Square wave Generator	0	Disable	x	Disabled	x	No counting
				1	Enable	0	Active = 0	0	Decrement on ↓
						1	Active = 1	1	Decrement on ↑
1	0	0	Software Triggered Strobe	0	Disable	x	Disabled	x	No counting
				1	Enable	0	Active = 0	0	Decrement on ↓
						1	Active = 1	1	Decrement on ↑
1	0	1	Hardware Triggered Strobe	0	Disable	x	Disabled		Note 1
				1	Enable, or trigger if Gate = active	0	Trigger = ↓	0	Decrement on ↓
						1	Trigger = ↑	1	Decrement on ↑

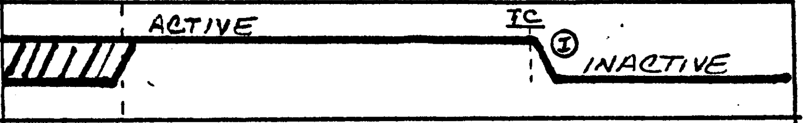
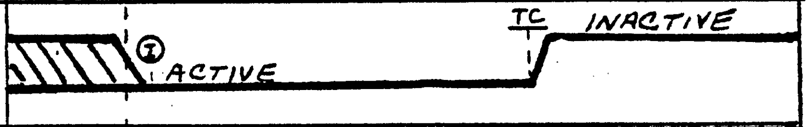
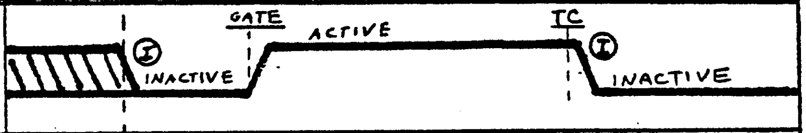
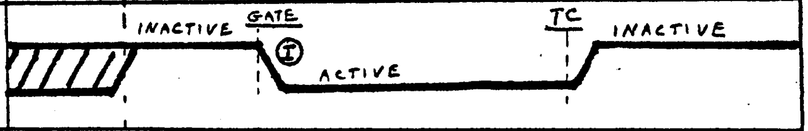
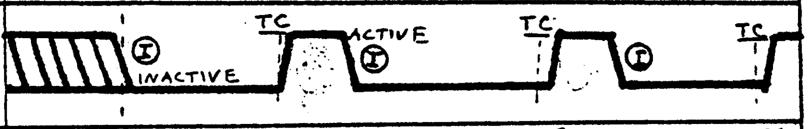
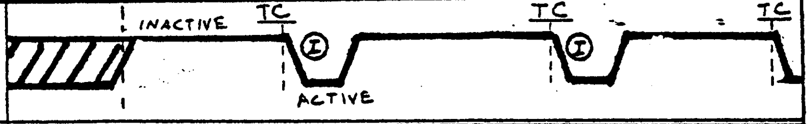
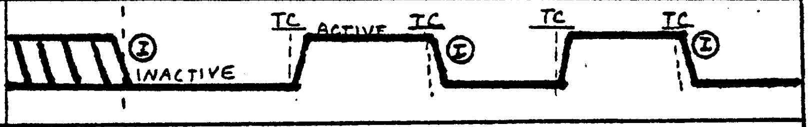
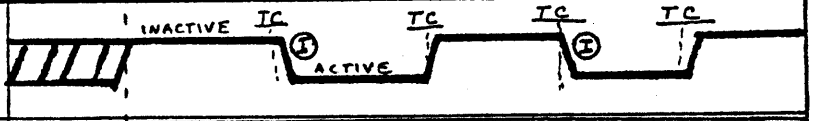
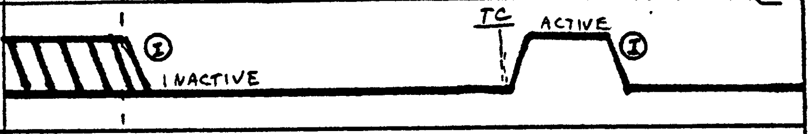
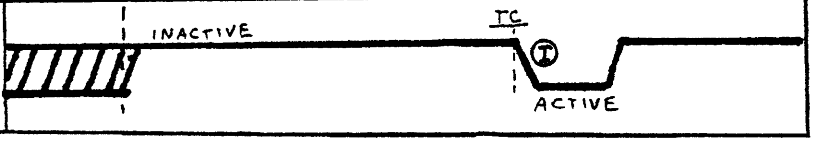
Table 3-11 : SUMMARY OF GATE AND CLOCK INPUT STATE CONTROL BITS (CHANNEL CONTROL REGISTER PROGRAMMING)

NOTES:

1. In Modes 1 and 5, counting proceeds freely with every CLOCK transition regardless of the GATE state; GATE controls the value of the count and the state of the OUTPUT.
2. In Modes 2 and 3, GATE also resets the OUTPUT state and restores the preloaded count value on the inactive-going edge.

x = Don't care, 0 or 1 state

TABLE 3-12 : OUTPUT WAVEFORM PROGRAMMING (OPS BIT) BY MODE

MODE					State of OPS	USER INTERFACE OUTPUT WAVEFORM (J2)
	M2	M1	M0	Name		Set Mode
0	0	0	0	Single Count or Software Triggered 1-shot	0	
					1	
1	0	0	1	Hardware Triggered 1-Shot (Re-Triggerable	0	
					1	
2	x	1	0	Repeated Count or Rate Generator or Divide-by-N	0	
					1	
3	x	1	1	Repeated Count or Square Wave Generator	0	
					1	
4	1	0	0	Software Delayed Strobe and Hardware Delayed Strobe	0	
5	1	0	1		1	

LEGEND

TC	Terminal Count - 0000, 0001, or 0002 depending on mode.
Ⓘ	Interrupt possible on this edge.
▨	The output assumes the state shown at the trailing edge of WR* when loading the Mode Control Register ; the state prior to this depends on the previous mode.

	OUTPUT (CONTROL) PORTS								HEX PORT ADDRESS	INPUT (STATUS & CONTROL) PORTS								
	b7	b6	b5	b4	b3	b2	b1	b0		b7	b6	b5	b4	b3	b2	b1	b0	
	WRITE PRESET VALUE TO CHANNEL 0								D8	READ CONTENT OF CHANNEL 0								
	WRITE PRESET VALUE TO CHANNEL 1								D9	READ CONTENT OF CHANNEL 1								
	WRITE PRESET VALUE TO CHANNEL 2								DA	READ CONTENT OF CHANNEL 2								
	SC1	SC0	RL1	RL0	M2	M1	M0	BCD	DB	-----------------NOT USED-----------------								
CH 0	MX4	MX2	MX1	GTE	GTS	INS	OPS	ENI	DC	READ DECREMENTS CHANNEL 0; DATA = DON'T CARE								CH 0
CH 1	MX4	MX2	MX1	GTE	GTS	INS	OPS	ENI	DD	READ DECREMENTS CHANNEL 1; DATA = DON'T CARE								CH 1
CH 2	MX4	MX2	MX1	GTE	GTS	INS	OPS	ENI	DE	READ DECREMENTS CHANNEL 2; DATA = DON'T CARE								CH 2
	ONES RESET INTERRUPT LATCHES					IRQ2	IRQ1	IRQ0	DF	SP1	SP0	OUT2	OUT1	OUT0	IRQ2	IRQ1	IRQ0	

Table 3-13: 7308 PORT BIT ASSIGNMENTS AND PROGRAMMING AID

(See Table 3-14 for definitions)

NOTE: Output Port DF bits 3-7 are Don't Care

OUTPUT PORT DB (CARD CONTROL PORT)

SC1	SC0	Select Channel
0	0	Channel 0
0	1	Channel 1
1	0	Channel 2
1	1	Illegal

RL1	RL0	Read/Load function
0	0	Dynamic read strobe
0	1	Read/load bits 8-15
1	0	Read/load bits 0-7
1	1	Read/load bits 0-7 then bits 8-15

M2	M1	M0	Mode select
0	0	0	Single count
0	0	1	One-shot
x	1	0	Rate generator
x	1	1	Square wave
1	0	0	Delayed strobe (software trigger)
1	0	1	Delayed strobe (hardware trigger)

BCD	Count radix select	
0	Binary	(Hexadecimal)
1	BCD	(Decimal)

INPUT PORT DF (CHANNEL STATUS AND SPARE INPUT LINES)	
BITS	FUNCTION
SP0, SP1	Spare high level active TTL input port bits
OUT0,OUT1,OUT2	Outputs of Channels 0,1,2; unaltered from 8253 output if corresponding OPS bit = 0, or inverted if corresponding OPS bit = 1
IRQ0,IRQ1,IRQ2	Outputs of interrupt request latches 0,1,2. Corresponding channel requests interrupt if IRQ = 1.

OUTPUT PORTS DC,DD,DE (CHANNEL CONTROL)

MX4	MX2	MX1	Multiplexed Input Select
0	0	0	0.500 microsecond clock
0	0	1	8.000 microsecond clock
0	1	0	128.0 microsecond clock
0	1	1	INPUT 0* Bus
1	0	0	INPUT 1* Bus
1	0	1	INPUT 2* Bus
1	1	0	Cascade from next channel
1	1	1	Loop control (Input Port DC,DD,DE)

INS	Input State
0	Decrement on falling edge
1	Decrement on rising edge

OBS	Output State
0	Output is inverted
1	Output is not inverted

GTE	Gate Enable - Software Control
0	Inhibit counting or disarm trigger
1	Permit counting or cause trigger

GTS	Gate State - Hardware Input
0	GATE is low level active
1	GATE is high level active

ENI	Interrupt Masking
0	Disable Interrupt
1	Enable Interrupt

Table 3-14 7308 PORT BIT PROGRAM ASSIGNMENTS

Section 4

How to Maintain the 7308
(Circuits and Signals)

Signal Glossary

This glossary identifies the signals associated with the 7308. It contains schematic coordinates for the orgin of signals generated by the 7308. The signals are arranges in four groups:

- STD BUS edge connector (Table 4-1)
- Interrupt connector J1 (Table 4-2)
- User interface connector J2 (Table 4-3)
- Internal signals (Table 4-4)

Table 4-1. STD BUS Edge Connector Signals used by the 7308

MNEMONIC	MEANING	PIN(S)	DESCRIPTION	FUNCTION
D0-D7	Data bus	7-14	High active	8-bit 3-state bidirectional data bus.
A0-A7	Address bus	15,17 19,21 23,25 27,29	High active	Low-order 8 bits of Address Bus, used for I/O port addressing. Responds to ports D8-DF when shipped.
SYSRESET*	System reset	47	Low active	Originates at processor card in response to power-on or PBRESET*.
IOEXP	I/O port expansion	35	High to Disable	Bank select; must be low for 7308; grounded by 7801,7802,7803 cards.
IORQ*	I/O request	33	Low active	Indicates the address bus has a valid port address on A0-A7.
RD*	Read	32	Low active	Indicates that the processor wants to read from the addressed input port.
WR*	Write	31	Low active	Indicates that the processor wants to write to the addressed output port.
INTRQ*	Interrupt	44	Low active open collector	Maskable interrupt to the processor card from the 7308 (optional).
NMIRQ*	Nonmaskable interrupt	46	Low active open collector	Nonmaskable interrupt to the process card from the 7308 (optional).
CLOCK*	Clock in	49	Low active	Clock signal; replaces crystal (optional).
CNTRL*	Clock out	50	Low active	Clock to other STD BUS cards (optional).
PCI PCO	Priority chain	52 51	High active	Interrupt card priority signal; trace maintains continuity only on 7308.

NOTE: Unused pins are open; pads are provided on some unused pins for user signals, NMIRQ*, CLOCK*, and CNTRL* require jumper straps - see Appendix A.

Table 4-2 Interrupt Connector J1 Signals

MNEMONIC	MEANING	PIN(S)	DESCRIPTION	FUNCTION
HELP0*	Interrupt	J1-2	Low active open collector	Channel 0 request latch set.
HELP1*	Interrupt	J1-4	Low active open collector	Channel 1 request latch set.
HELP2*	Interrupt	J1-6	Low active open collector	Channel 2 request latch set.
GROUP*	Interrupt	J1-8	Low active open collector	OR'd output of Channel 0,1,2 interrupt request latches.
TEST	Test signal (500 NS)	J1-10		Onboard oscillator output signal for card testing only. (2.0 MHz)
GND	Ground	All odd		Provides signal/ground pair for each interface signal.

Table 4-3 User Interface Connector J2 Signals

MNEMONIC	PIN(S)	FUNCTION
GATE0*	J2-2	Channel 0 gate or trigger input.
CLOCK0*	J2-4	Input 0 count-clock (bussed to all channel multiplexers).
OUT0*	J2-6	Channel 0 output signal.
GATE1*	J2-8	Channel 1 gate or trigger input.
CLOCK1*	J2-10	Input 1 count-clock (bussed to all channel multiplexers).
OUT1*	J2-12	Channel 1 output signal.
GATE2*	J2-14	Channel 2 gate or trigger input.
CLOCK2*	J2-16	Input 2 count-clock (bussed to all channel multiplexers).
OUT2*	J2-18	Channel 2 output signal.
SPARE0,1	J2-20,22	Spare high-active TTL-compatible input port lines.
+5 V	J2-24,26	+5V output to signal conditioning card (1 ohm each).
GND	All odd	Ground output to form signal/ground pair for all signals.

NOTE: Even numbered pins 2-18 have programmable polarity.

Table 4-4 Internal 7308 Signals (See appendix)

MNEMONIC	MEANING	DESCRIPTION	ORIGIN COORDINATES	COMMENTS
BA0 BA1	Buffered address	High active	B7	Drives port address decoders, 8253.
BD0 to BD7	Buffered data bus	High active	D7	Drives all ports and 8253.
BRD* BWR*	Buffered read,write	Low active	B7	Controls Port decoders, 8253, and direction of data bus tranceiver.
CK0,1,2	8253 clocks	Falling edge	B3,C3	8253 clock inputs; selected by multiplexer with polarity programmed.
EI0,1,2	Interrupt enable	High active	D5,D6	Mask/enable signal for each channel's interrupt. 1 = enable.
F0,F1,F2	Frequency	Rate source	B6,C6	Clock oscillator/scaler outputs selected by jumper straps.
G0,G1,G2	Gates	High active	B4,C4	Gate inputs to 8253, with polarity programmed.
GE0,1,2	Gate enables	High active	D5,D6	Programmable gate signal, output from channel control register; enables user gate signal.
GS0,1,2	Gate state	Programmable	D5,D6	Selects gate high/low active; 0 = low active at J2.
IL0,1,2	Interrupt latch	High active	A2,A3	Interrupt latch set = 1.
IN0,1,2	Input	High active	B4,C4	Buffered user CLOCK* inputs.
IP0* to IP3*	Input port strobe	Low active	B6	Processor reads input port on rising IP* edge; outputs from port decoder.
IRQ0 to IRQ2	Interrupt latch strobe	Rising edge	B2	Enabled interrupt requests from channels 0-2; rising edge sets latch.
IS0,1,2	Input state	Programmable	D5,D6	Selects clock input active edge; occurs after multiplex selection. 0 = falling edge at any CLOCK* input.

Table 4-4 (contînued)

MNEMONIC	MEANING	DESCRIPTION	ORIGIN COORDINATES	COMMENTS
MX0-1,2,4 MX1-1,2,4 MX2-1,2,4	Multiplexer select mode	High active	D5,D6	Select one of 8 inputs via 8:1 multiplexers. Lines form octal (3-bit binary) code. Outputs are from channel control registers.
OP0* to OP3*	Output port strobe	Low active	B6	Processor writes to output port with data latched on rising edge; outputs are from port select decoder.
OS0,1,2	Output state	Programmable	D5,D6	Selects output active logic state & Interrupt latch strobe timing. Output from channel control registers.
OUT0,1,2	Channel output	Programmable	B1,C1	Outputs from 8253 after programmed polarity conditioning; drive interrupt masks & card outputs.
RST RST*	Reset	High active low active	C7	Buffered SYSRESET*.
SCT*	Select 8253	Low active	B6	Chip select for 8253 from address decoder.
SX* SY* X0* to X7* Y0* to Y7*	Card address selection	Low active	B7,C7	Card select decoder busses. SX is the A5-A7 decoder output matrix and SY is the A3,A4 decoder output matrix. Make one selection at each matrix. 7308 is shipped with X6* and Y3* selected.
500 ns	Clock	Rate source	A6	Crystal oscillator output for testing

Return for Repair Procedures

Domestic Customers:

1. Call our factory direct at (408) 372-4593, and ask for CUSTOMER SERVICE.
2. Explain the problem and we may be able to solve it on the phone. If not, we will give you a Customer Return Order (CRO) number.

   Mark the CRO number on the shipping label, packing slip, and other paperwork accompanying the return. We cannot accept returns without a CRO.
3. Please be sure to enclose a packing slip with CRO number, serial number of the equipment, if applicable, reason for return, and the name and telephone number of the person we should contact (preferably the user), if we have any further questions.
4. Package the equipment in a solid cardboard box secured with packing material.

   CAUTION: Loose MOS integrated circuits, or any product containing CMOS integrated circuits, must be protected from electrostatic discharge during shipment. Use conductive foam pads or conductive plastic bags, and never place MOS or CMOS circuitry in contact with Styrofoam materials.
5. Ship prepaid and insured to:

   Pro-Log Corporation
   2411 Garden Road
   Monterey, California 93940

   Reference CRO #______________.

International Customers:

Equipment repair is handled by your local Pro-Log Distributor. If you need to contact Pro-Log, the factory can be reached at any time by TWX at 910-360-7082.

Limited Warranty: Seller warrants that the articles furnished hereunder are free from defects in material and workmanship and perform to applicable, published Pro-Log specifications for one year from date of shipment. This warranty is in lieu of any other warranty expressed or implied. In no event will Seller be liable for special or consequential damages as a result of any alleged breach of this warranty provision. The liability of Seller hereunder shall be limited to replacing or repairing, at its option, any defective units which are returned F.O.B. Seller's plant. Equipment or parts which have been subject to abuse, misuse, accident, alteration, neglect, unauthorized repair or installation are not covered by warranty. Seller shall have the right of final determination as to the existence and cause of defect. As to items repaired or replaced, the warranty shall continue in effect for the remainder of the warranty period, or for ninety (90) days following date of shipment by Seller or the repaired or replaced part whichever period is longer. No liability is assumed for expendable items such as lamps and fuses. No warranty is made with respect to custom equipment or products produced to Buyer's specifications except as specifically stated in writing by Seller and contained in the contract.

Section 5

Notes on Typical Applications

This section describes typical application examples for the 7308 counter/timer card, including suggestions for hardware and program support of the counter/timer. channels. We have tested the applications and we believe they are correct. However, we do not represent them as being appropriate for any specific application.

Program segments and subroutines are given as programming examples in some instances. They are available to the user without licensing from Pro-Log and may be used directly or adapted to a specific application. The 8080 instruction set is used exclusively, allowing the subroutines to run on 8080, 8085, Z80, and other code-compatible processors. Flow diagrams (which reference no processor characteristic) allow you to translate the routines into the assembly codes of other processor types.

Techniques appropriate for typical 7308 applications are described in terms of:

- Loading each channel's load register:
  - How to compute the preload value for time delays (1.5 ms Example)
  - How to compute time delay accurancy and select a clock range.
  - Examples: -Time delays: 50 ms, 2.345 seconds
    -Preload values for all common baud rates.
- Simple one-time delays
  - How to operate in mode zero.
  - How to poll the interrupt latches.
- Elapsed-time measurements
  - Software gating.
  - Hardware gating.
- Event counting
  - Totalizing.
  - Limit (alarm) counting.
- One-shot and delayed strobe modes
  - Software trigger (modes 0,4).
  - Hardware trigger (modes 1,5).

Loading Each Channel's Load Register

It is necessary to preload 7308's counter/timer channels to an initial value, or counting cannot proceed. This value is down-counted toward terminal count (the lowest obtainable count: 0000 in modes 0, 1, 4, and 5; 0001 in mode 2; and 0002 in mode 3). Obtain precise time delays by preloading the load register with a value N and counting a precisely timed clock. The delay is the product of N and the clock period.

Example: Determine the load register value for 1.5 ms time-out, using each of the 7308's three clock ranges:

- F0 = 0.500 µs ± 0.05%
- F1 = 8.000 µs ± 0.05%
- F2 = 128.0 µs ± 0.05%

For F0, 1.5 ms = (N)(0.500 µs)

$$N = \frac{1500\ \mu s}{0.5\ \mu s} = 3000$$

Thus the preload value for the channel's load register is 3000, if BCD (decimal) counting is selected. For binary (hexadecimal) counting, $3000_{10} = 0BB8_{16}$ is the preload value.

For F1, 1.5 ms = (N)(8.000 µs)

$$N = \frac{1500\ \mu s}{8.000\ \mu s} = 187.5 \cong 188$$

The preload value is 0188 for BCD operation, or $188_{10} = 00BC_{16}$ for binary operation.

For F2, 1.5 ms = (N)(128.0 µs)

$$N = \frac{1500\ \mu s}{128.0\ \mu s} = 11.72 \cong 12$$

The preload value is 0012 for BCD operation, or $0012_{10} = 000C_{16}$ for binary operation.

Time Delay Accuracy. The accuracy of a programmed time delay varies with the the ratio of the desired delay to the clock period (i.e., error increases as resolution decreases). In the previous example, the same time delay is programmed using each of the three clock ranges. Resulting accuracies for the three clock ranges are:

- For F0, the delay is
  (N)(0.500 µs) = (3000)(0.500 µs) = 1.500 µs
  Since the desired time delay is exactly divisible by the clock period, the overall error is equal to the clock accuracy, which is ±0.05%. Resulting delay is 1.49925 to 1.50075 ms.

- For F1, the delay is
  (N)(8.000 µs) = (188)(8.000 µs) = 1.504 ms.

  Ignoring clock accuracy, the error is

  $$(100)\ \frac{(1.504\text{ ms}) - (1.500\text{ ms})}{1.500\text{ ms}} = 0.267\%.$$

  Including clock error, the actual delay is then
  1.500 ms + 0.217% to 0.317%, or
  1.5033 ms + to 1.5048 ms.

- For F2, the delay is
  (N)(128.0 µs) = (12)(128.0 µs) = 1.536 ms

  Ignoring clock accuracy, the error is

  $$(100)\ \frac{(1.536\text{ ms}) - (1.500\text{ ms})}{1.500\text{ ms}} = +2.4\%.$$

  Including clock error, the actual delay is then
  1.5 ms +2.35% to +2.45%, or
  1.5353 to 1.5368 ms.

IMPORTANT: An error of up to one clock period can result in single delays (modes 0,1,4,5) or in the first output period of repetitive modes (modes 2,3) when the 7308's onboard oscillator clock ranges are selected; this stems from the fact that the 7308 oscillator is not synchronized to the processor clock (modes 0,2,3,4), or to the external gate trigger (modes 1,5). The program or external trigger may command the channel to start counting, but the channel cannot do so until the

the first clock falling edge occurs; without clock synchronization, this may require from 0 to 100% of the clock period. Note that this error amounts to only 0.0015% of full scale (1 count in 2^{16} or 65,536), and is the reason that high-resolution counting with the fastest possible clock produces the most accurate time delay. (This error source does not apply to multiplexer range 7, where the channel is clocked by programmed input port reads, or when the user triggers the one-shot modes with a gate signal that is synchronized to a user-supplied clock signal.)

Other Accuracy Considerations. The channel's output signal goes active when the program writes to the mode control register to select mode 0. But the actual counting process cannot start until the channel's load register is written to by the program. Thus, the time that the output is active is extended by the time required for the processor to execute the instructions that write to these registers, and by any instructions executed in between. You must consider this time if you are to account every fractional microsecond in your design.

You should consider the actual accuracy needed in a specific application. For example, the potential error described above may not be significant, if the time delay is being used to overshadow the bounce time of a relay contact closure. But it may be important when dealing with the fastest baud rates in a communications application. Looked at another way, it's possible for you to spend a lot of time calculating a very precise time delay, only to have the accuracy balloon to 1% or more by the execution of a few extra program instructions.

Also you should be aware that while the 7308's specified accuracy is ±0.05% over the operating temperature and Vcc range, this is not equal to the accuracy of a digital wristwatch maintained at 37°C by body heat, and operated by a lithium or NiCad battery. For example, a 30-day month contains 2,592,000 seconds and 0.05% of this value is 1295 seconds, or 21.6 minutes. A wristwatch that gains or loses 1 second per month has an accuracy of about ±0.00004%.

The 7308's oscillator can approach the timing accuracy of a digital wristwatch, but only if you are willing to:

a. Hold the operating temperature and power supply voltage constant.

b. Select an oscillator crystal, or trim the supplied crystal, for much greater accuracy.

A more practical approach, particularly in time-of-day applications, is to use an accurate, external time standard such as the local AC power line.

Examples of Computing Preload Values. Compute the preload value for (a) 50 ms and (b) 2.345µs. Find the best clock range and resulting error for each.

(a) 50 ms. F0 cannot be used for 50 ms, since $\frac{50 \text{ ms}}{0.5\ \mu\text{s}} = 100{,}000$ and any one channel can only count to 63,535 (in binary) or 9,999 decimal. However, F1 can be used:

$$\frac{50 \text{ ms}}{8.0\ \mu\text{s}} = 6250_{10} = 186A_{16}.$$

Either BCD (decimal) or binary (hexadecimal) counting modes may be used with identical results. The error is

$$(100)\ \frac{(6250)(8.0\ \mu\text{s})}{50 \text{ ms}} - 50 \text{ ms} = 0.0\%$$

Adding the clock error (±0.05%) gives a time delay in the range of 49.975 ms; to 50.025 ms.

(b) 2.345 sec. Figure 3-9 shows the delay ranges possible with one channel on each clock range (F0, F1, F2). Only F2 can achieve a delay above 549 ms:

$$\frac{2.345 \text{ s}}{128.0 \text{ ms}} = 18{,}320.3 \cong 18{,}320_{10} = 4790_{16}.$$

This time BCD (decimal) counting cannot be used, since the largest decimal number the channel can hold is 9999. Binary (hexadecimal) counting is required. The error is

$$(100)\quad \frac{(18{,}320)(128.0\ \mu s) - 2.345\ s.}{2.345\ s.} = -0.0017\%.$$

Adding the clock error (±0.05%) gives an overall error of -0.0517% to +0.0483%, or a time delay in the range of 2.3479 to 2.34513 seconds.

Example of Computing Preload Values for Common Baud Rates. The computed preload values for all of the common baud rates, used with UART and USART devices and their accuracies are listed in Table 5-1.

NOTE: A single channel using the F0 clock (0.5) and binary (hexadecimal) counting can handle all of these baud rates. We suggest channel mode 2 (marker pulse generator) for actual use with a UART or USART.

(Table 5-1. Computed Channel Preload Values for Common Baud Rates.)

Simple One-Time Delays (Mode 0)

Mode 0 is useful for program-initiated time delays that are performed only once. You can use the delays to time the processor's movement from function to function in the program, or to override and prevent a processor "hang-up" when the hardware malfunctions in the processor-controlled system. Examples of this feature are:

1. The processor detects what appears to be a switch closure, then programs a switch bounce time (e.g., 1 to 50 ms) to give the switch contacts time to settle. By rereading the switch contacts a second time, the processor can differentiate between a geniune switch closure and a noise pulse.

2. The processor has opened a valve to fill a tank and is waiting for a signal from a float to inform it that the tank is full. However, no fluid is flowing into the tank because of a malfunction. A fixed, overriding time delay then keeps the processor form waiting indefinitely for a "tank-full" signal that can never occur.

Mode 0 is setup as shown in Table 5-2. The preload value and range selection for a 2.345 s. delay (Example (b) in the load register discussion) are used to reload channel 1. We assume that channel 1's gate input is not connected and, therefore, pulled up to the logical 1 state; also, the channel's out-

BAUD RATES

iii. Compute preload values for all of the common baud rates used with UART and USART devices and give the accuracy.

NOTE: A single channel using the F0 clock (0.5 us) and binary (hexadecimal) counting can handle all of these baud rates. Channel Mode 2 (marker pulse generator) is suggested for actual use with a UART or USART.

		7308 PERFORMANCE		
BAUD RATE	DESIRED BIT PERIOD	ACTUAL BIT PERIOD	ERROR	HEXADECIMAL PRELOAD VALUE
50	20.000 ms	20.000 ms	0.05%	9C40
75	13.333 ms	13.332 ms	0.05%	682A
110	9.090 ms	9.091 ms	0.05%	4706
134.5	7.434 ms	7.435 ms	0.06%	3A16
150	6.666 ms	6.667 ms	0.06%	3415
300	3.333 ms	3.332 ms	0.06%	1A06
600	1.666 ms	1.665 ms	0.06%	0D05
1200	833.33 us	833.20 us	0.07%	0683
1800	555.56 us	555.20 us	0.06%	0457
2000	500.00 us	500.00 us	0.05%	03E8
2400	416.66 us	416.40 us	0.09%	0341
3600	277.77 us	277.60 us	0.13%	022C
4800	208.33 us	208.00 us	0.13%	01A1
7200	138.88 us	138.80 us	0.13%	0116
9600	104.16 us	104.00 us	0.21%	00D0
19200	52.08 us	52.00 us	0.21%	0068
38400	26.04 us	26.00 us	0.21%	0034

NOTE: The error given above includes 0.05% clock error.

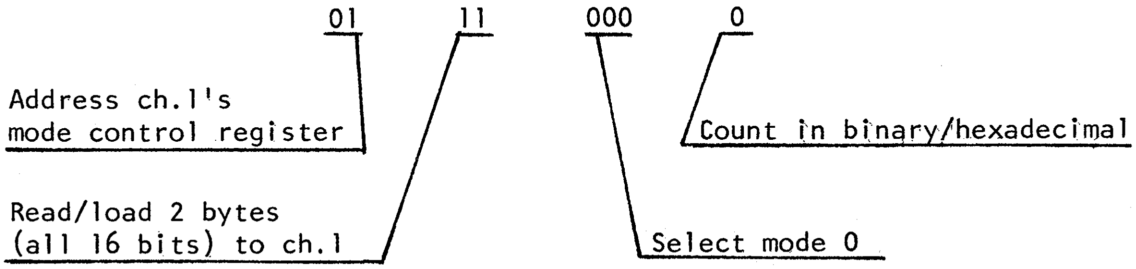
Table 5-1 : CHANNEL PRELOAD VALUES FOR COMMON BAUD RATES

put signal is used to set the channel 1 interrupt latch.

Figure 5-2 shows how the interrupt is serviced by polling the 7308's status port, assuming that an interrupt has vectored the processor to memory location 0038 hexadecimal. However, the same instruction sequence might be used for noninterruping application, in which the program occasionally tests the channel to see if it has reached terminal count yet.

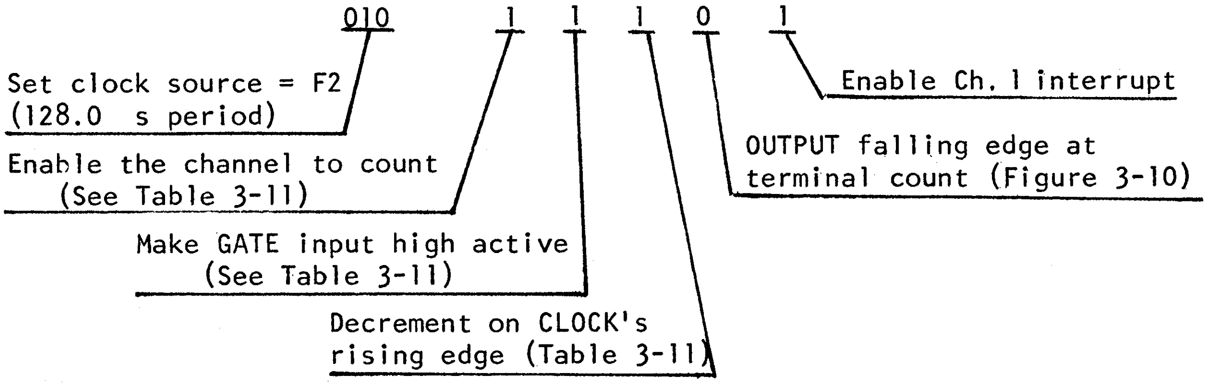
The hexadecimal data written to output ports DB, DD, and D9 in the program example in Table 5-2 is derived as follows:

(a) Data Output to the channel mode register (Output Port DB)


01
11
000
0
Address ch.1's mode control register
Read/load 2 bytes (all 16 bits) to ch.1
Select mode 0
Count in binary/hexadecimal

0111 0001 (binary) = 70 (hexadecimal)

(b) Data Output to the channel control port (Output Port DD for Ch.1)


010
1
1
1
0
1
Set clock source = F2 (128.0 s period)
Enable the channel to count (See Table 3-11)
Make GATE input high active (See Table 3-11)
Decrement on CLOCK's rising edge (Table 3-11)
OUTPUT falling edge at terminal count (Figure 3-10)
Enable Ch. 1 interrupt

0101 1101 (binary) - 5D (hexadecimal)

PRO-LOG CORPORATION PROGRAM ASSEMBLY FORM

HEXADECIMAL			MNEMONIC			TITLE DATE
PAGE ADR	LINE ADR	INSTR.	LABEL	INSTR.	MODIFIER	COMMENTS
XX	X0	3E	2.345 SEC. DELAY	LDAI		LOAD CH. 1's MODE CONTROL REGISTER (SEE
	1	70		–	01 11 000 0	TEXT FOR EXPLANATION OF BIT PATTERN)
	2	D3		OPA		
	3	DB		–	PORT DB	
	4	3E		LDAI		LOAD CH. 1's CHANNEL CONTROL PORT
	5	5D		–	010 1 1 1 01	
	6	D3		OPA		
	7	DD		–	PORT DD	
	8	3E		LDAI		RESET CH. 1's INTERRUPT LATCH WITHOUT
	9	02		–	BIT 1	DISTURBING THE CH. 0 + CH. 2 LATCHES
	A	D3		OPA		
	B	DF		–	PORT DF	
	C	FB		ENI		ENABLE THE PROCESSOR'S INTERRUPT INPUT
	D	3E		LDAI		PRESET CH. 1's LOAD REGISTER - BITS 0-7
	E	90		–	90	FOLLOWED BY BITS 8-15
	F	D3		OPA		
XX	Y0	D9		–	PORT D9	NOTE: $4790_{16} = 18,320_{10}$
	1	3E		LDAI		
	2	47			47	$(18,320)(128\mu S) = 2.345$ SEC
	3	D3		OPA	→	*** TIME DELAY STARTS AS THIS INSTRUCTION EXECUTES
	4	D9		–	PORT D9	
	5	•		•		PROGRAM CONTINUES; INTERRUPT OCCURS
	6	•		•		IN 2.345 SECONDS
	7	•		•		
	8					
	9					
	A					
	B					
	C					
	D					
	E					
	F					

Table 5-2. Example of Mode 0 Programming (Program Segment)

(c) Two Bytes of Data Output to Ch. 1's Load Register (Output Port D9)

Since step (a) above wrote 11 (binary) to channel 1's RL bits in the mode control register, the count cannot begin until two bytes of preload data are written to Channel 1's load register. Both halves of the load register are accessed through output port D9 (for Channel 1).

The first byte written to output port D9 goes to Channel 1's load register bits 0-7 (least significant 8 bits). The second byte goes to bits 8-15 (most significant 8 bits). Channel begins the delay at the conclusion of the second write operation to output port D9.

The hexadecimal preload value developed for 2.345 seconds earlier is 4790 (hexadecimal);this produces a 2.345 s delay (using the F2=128.0 µs clock range) with a maximum error of -0.0517%.

Elapsed Time Measurements

Figure 3-3 shows that, in mode 0, the counter continues to decrement with every clock transition, regardless of whether terminal count has occured. The channel also continues to respond to the gate after terminal count. You can use these characteristics to make elasped time measurements in the program, or to measure the pulse width of an external signal.

An important consideration in elapsed time measuremnts is the ability of the program to make error-free reads from the channel's read latch, even if the channel is being incremented at high speed by a fast clock. Section explains how to program the mode control register's RL (read/load) bits for latched, rather than transparent, reading of the channel's current count.

Software Gated Elapsed Time Measurements. These are made by simply allowing the channel to down-count continuously. The time measurement starts the first time the program reads the current count in the channel, and stops with the next read. By subtracting the initial value fromthe last value, the program can determine the elapsed time between two events. This operation is illustrated in Figure 5-2. Note that, if desired, the program can actually stop and start the count by controlling the GTE bit in the channel control port: setting GTE = 1 causes the channel to start counting (if the GATE

PRO-LOG CORPORATION **PROGRAM ASSEMBLY FORM**

HEXADECIMAL			MNEMONIC			TITLE DATE
PAGE ADR	LINE ADR	INSTR.	LABEL	INSTR.	MODIFIER	COMMENTS
	0					
	1					
	2					
	3					
	4					
	5					
	6					
	7					
00	3 8	DB	INTRQ	IPA		Read the 7308 card's status port bits; save in Register B
	9	DF			PORT DF	if needed later
	A	47		LDB	A	
	B	D3		OPA		Reset only the interrupt latches
	C	DF			PORT DF	a "1" in bits 0,1,2 will reset the corresponding interrupt latch
	D	1F		RRAC		Is Channel 0's interrupt set? Service Channel 0 if so.
	E	DC		JS	C1	NOTE: As written, this routine gives
	F	XX			(Service Ch. 0)	priority to the channels in 0-1-2 order.
	0	XX			-	(CH 0 service Routine at address XXXX)
	1	78		LDA	B	Is Channel 1's interrupt latch set?
	2	1F		RRAC		(CH 1 service routine at address YYYY)
	3	1F		RRAC		
	4	DC		JS	C1	
	5	YY			(Service Ch. 1)	
	6	YY			-	
	7	78		LDA	B	Is Channel 2's interrupt latch set?
	8	1F		RRAC		
	9	1F		RRAC		(Ch 2 service routine at address ZZZZ)
	A	1F		RRAC		
	B	DC		JS	C1	
	C	ZZ			(Service Ch. 2)	
	D	ZZ			-	
	E	.		.		No 7308 interrupt latches are set - continue on or poll
	F	.		.		other cards

Table 5-3. Interrupt Polling Program Segment

input is active) and clearing GTE to 0 stops the count. If the elapsed time is long enough for the counter to down-count past terminal count one or more times, the program must arrange to lengthen the channel to more than 16 bits. This is arranged by cascading two channels, or by propagating the count to a RAM or index register location. These techniques are discussed later in the subsection called "Extending the Count."

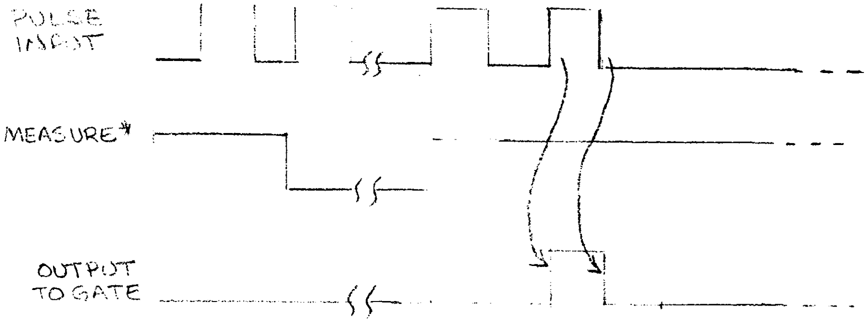
Hardware Gated Elapsed Time Measurements. These pulse-width measurements are made by connecting the signal to be measured to the channel's GATE input and programming the correct gate polarity. Counting can then occur only while GATE is active, thereby making a direct measurement of the pulse width. Figure 5-1 shows how the current count can be read twice and compared to determine whether the measurement is complete, i.e., whether the channel is still counting. An alternate method is to connect the signal being measured to both the GATE input and one of the spare TTL input lines available as bits 6 and 7 of input port DF (the 7308 status port). By sensing the state of the signal being measured, the program can determine whether or not the measurement is complete.

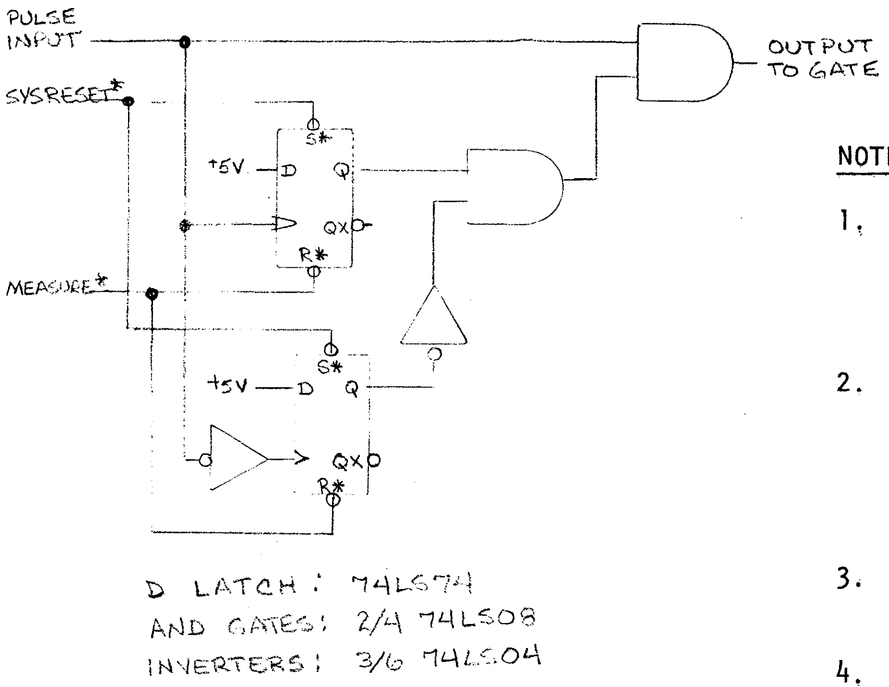
IMPORTANT: This techinque is accurate only if:

1. The microprocessor is substantially faster than the pulse being measured, so that it can control the measurement. If the pulse-repetition rate is too high, the channel may be re-enabled to count by a second incoming GATE pulse, before the processor can react to the first measurement, thereby causing an error; or,
2. The measurement is assisted by user-added hardware, such as an edge-sensitive latch and gate Figure 5-1), which allows a relatively fast pulse to be measured without errors.

Event Counting

You can use each of the 7308 channels for event counting up to 2.5 MHz, either to totalize an unknown number of events or to prevent an operation from happening more than N times by generating an interrupt. Note that with the 7308, the event may be represented by either an external signal or a programmed input-port read operation. The only difference between totalizing and limit counting is the initial value preloaded to the counter. Examples of the count sequence for totalizing and limit counting are shown in Figures 5-2 and 5-3, respectively.





NOTES

1. Use this circuit to interface a pulse train with high repetition rate (relative to the processor's instruction rate - see text) to the 7308's GATE input, for measuring a single pulse width.

2. The circuit passes the first high-active pulse from a pulse train following the rising edge of MEASURE*. Using 74LS devices, the leading edge of the pulse is delayed about 40ns and its width is streched by about 25ns (typical values).

3. MEASURE* is a user-supplied signal that could be a single, spare output port line for program control of the measurement.

4. For a low-active pulse, use an additional inverter at PULSE INPUT.

Figure 5-1: Circuit Example For Pulse-Width Measurement.

Totalizing. The channel is preset to a value representing zero; then the number of events is determined just by reading the channel value. Remember that since the 8253 can only decrement, either 2's complement or 10's complement arithmetic is needed to determine the number of events. Thus, the value representing zero is FFFF in hexadecimal (binary) operation and 9999 in decimal (BCD) operation. Alternately, the channel can be allowed to count freely from any initial value (as long as the initial value is stored by the program). The count can then be determined simply by subtracting the initial value from the final value. (See subsection called "Extending the Count").

Limit Counting. The channel is preloaded with the maximum number of events and the interrupt is enabled. An interrupt then informs the program that the limit has occurred. Alternately, you may want to use a number slightly in advance of the limit (smaller) to provide an early warning. For example, a system with mechanical intertia (such as a fast motor) may need an early warning to make prelimit adjustments (e.g., reversing the motor current to act as a brake) before the actual limit event.

DECIMAL TOTALIZING FROM INITIAL PRESET VALUE (10'S COMPLEMENT COUNTING)

STEP	COMMENT (Decimal Counting Example)	COUNTER CONTENT
1	Preload counter to 9999	9999 (initial count)
2	Enable event counting	Event.....9998 Event.....9997 Event.....9996 Event.....9995 Event.....9994 (final count)
3	Subtract final count from initial count:	9999 - 9994 5 total events

HEXADECIMAL TOTALIZING FROM RANDOM INITIAL VALUE (2's COMPLEMENT BINARY COUNTING)

Note: in order to save time, it is unnecessary to preload the channel prior to each count operation (although the channel must be preloaded the first time it is used after Mode 0 or any mode is selected).

STEP	COMMENT (Binary/hexadecimal Counting Example)	COUNTER CONTENT
1	Record the current content of the counter	ABCD (initial count)
2	Enable event counting	Event.....ABCC Event.....ABCB Event.....ABCA : : Event.....1235 Event.....1234 (final count)
3	Subtract final count from initial count:	ABCD - 1234 9999_{16} total events = $39,321_{10}$ total events

IMPORTANT: If the counter is allowed to restart from some previous value, as in this example, or if a large number of events is anticipated such that the channel may overflow (more than 9999_{10} or $FFFF_{16}$ counts), the program must arrange to propagate a borrow. The program may do this itself by decrementing an index register or RAM location upon Terminal Count interrupt, or two channels may be cascaded. so that the borrow is propagated into the adjacent channel by hardware. See (f) "Extending the Count" later in this section.

FIGURE 5-2 : DECIMAL AND HEXADECIMAL TOTALIZING

LIMIT COUNTING (HEX MODE) WITH INTERRUPT ALARM

STEP	COMMENT (Binary/Hex Counting)	COUNTER CONTENT
1	Select Mode 0 (or another mode)	⋮
2	Select appropriate output state to cause OUTPUT falling edge at Terminal Count, and enable interrupt.	⋮
3	Preload channel with desired limit. Example: for 123_{10} maximum events in hexadecimal counting mode, use...........	007B
4	Enable counting	Event....007A Event....0079 Event....0078 ⋮ Event....0002 Event....0001 Event....0000 (Terminal Count for Mode 0)
5	7308 interrupts processor - limit of 123_{10} events has occured	

IMPORTANT: Mode 0 is assumed in this example, but other modes might be more useful. For example, Mode 2 would automatically reload the counter with the initial count after interrupting, and therefore is preferable where the limit count is to be repeated. Note, however, that the value of Terminal Count varies with mode, as does the desired of the output signal for interrupting. See Section 3 .

FIGURE 5-3 LIMIT COUNTING EXAMPLE

Extending the Count

Sometimes it is necessary to extend the effective length of the 16-bit channel by propagating a borrow from the channel output to another counter. For example, a count extension is needed if:

- A time delay is more than 8.3 seconds.
- A count of more than 9,999 (decimal) or 65,535 (or FFFF hexadecimal) is anticipated, or the channel is allowed to count freely.
- The channel is being used to count a constant value repeatedly, e.g., the channel is programmed to generate an interval of 1 ms, and a second count is needed to find the number of 1 ms intervals that have occurred.

You can extend the count in two ways:

1 Connect the channel's output signal to the input of another counter. Multiplexer range 6 allows each channel to be locked by the output of the adjacent channel. The channels cascade as follows:

   - Channel 0 drives channel 1
   - Channel 1 drives channel 2
   - Channel 2 drives channel 0.

   NOTE: For normal cascade operation, set the driving channel's OPS bit and the driven channel's IPS bit to the same state when programming the two channel control ports. Figure 3-9 shows the range of delays possible with 2 or 3 channels cascaded.

2. Use the channel's interrupt output to propagate a borrow or a carry into one of the processor's registers or a RAM location. Table 5-3 shows how to service the interrupt that results from a counter/timer channel overflow. The output increments a register pair then exits back from the interrupt, using only a fraction of the available processor program time (about 0.7% of the processing time available to a 2.5MHz Z80 processor, with the 7308 channel set to generate 1.0 ms time intervals).

   IMPORTANT: If the channel is allowed to interrupt the program continually, to propagate a channel overflow into a processor register or RAM location, the time required for interrupt servicing should be considered when designing time-critical portions of the program. For example, the interrupt service routine might be written in such a way that its execution time is constant (even if any conditional jump instructions or loop sequences are included), thereby making it easier to accommodate the repetitive interrupt when designing the rest of the program.

e. One-Shot and Delayed Strobe Modes

Channel Modes 0,1,4 and 5 can all be used to generate signals which simulate one-shot circuits. For example, the TTL logic designer will find Mode 1's signal characteristics similar to those of the 74LS123, a retriggerable one-shot chip. The other modes are simply variations of Mode 1 which differ in triggering and output waveform characteristics as summarized in Table 5-4.

MODE	DESCRIPTION	TRIGGERED BY	OUTPUT RESPONSE	AUTORELOAD
0	Program Triggered One-shot	Program write to the channel's Mode Register	Output active during the full programmed delay plus program overhead	No
1	Hardware Triggered One-shot	Active-going edge of GATE signal	Output active during the full programmed delay	Yes, and Retriggerable
4	Program Triggered Delayed Strobe	Program write to the channel's Load Register	Output active for one clock period only, after full programmed delay	No
5	Hardware Triggered Delayed Strobe	Active-going edge of GATE signal	Output active for one clock period only, after full programmed delay.	Yes, and Retriggerable

TABLE 5-4 ONE-SHOT AND DELAYED STROBE MODE SUMMARY

IMPORTANT: Refer to Section 3 for detailed descriptions of the setup requirements and characteristics of the modes.

As shown in the waveforms in Figures 3-3 and 3-4, "one-shot" refers to a waveform which is active immediately after being triggered, and which is active for the full programmed time delay; "delayed strobe" is similar to a one-shot, except that the output doesn't go active until Terminal Count and then is active for only one clock period (figures 3-7,3-8).

7308 MODULES

(VARIABLE △)

This routine initializes channel 0 of the timer counter card. The channel setup information is provided in a lookup table ocntined in memory. Setup information is orgainzed in the table by the user as sets of data each consisting of four bytes. The first byte of a set is the mode register information; the second byte is the control port data. The third and fourth bytes are count data. Enter this routine with the number of desired data set in register A (00 is the first data set). A miximum of 64 data sets can be accomodated.

Altered Reg = A,F,D,E,H,L

(INTERRUPT CHK)

Upon an occurrence of an interrupt, INTERRUPT CHK can be used to check interrupt status of the 7308 card. Status Port DF is checked for interrupt activity. If a channel has interrupted, its service routine is executed and check is continued.

Altered Reg = A,F

(MS△)(EVENT COUNTER)

These modules demonstrate standard techniques for initializing internal timer (Channel 0 - Channel 2). These routines are identical except for the time delay values that are programmed.

Altered Reg = A,F

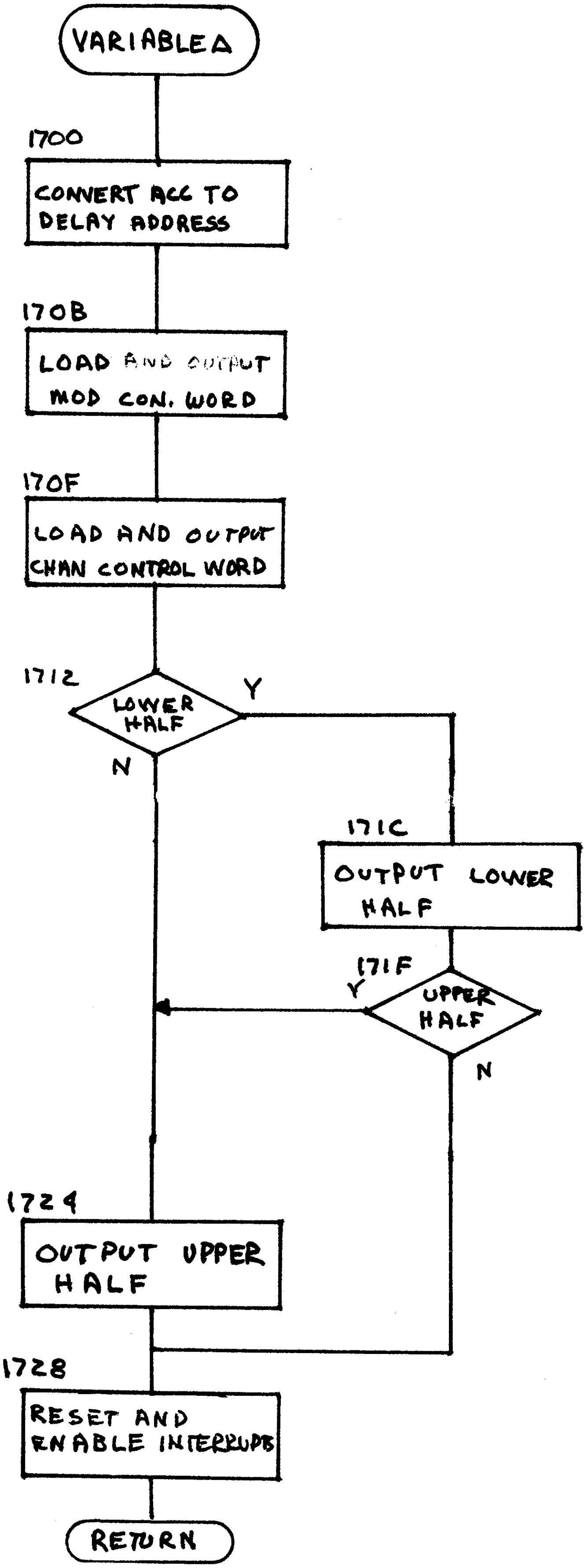
(MS•△•INTR)

Using this routine in conjunction with (MS△) and (INTERRUPT CHK) a variable milli-second delay may be derived. Every milli-second, this module decrements a memory location determined by user until count equals zero, at which time the service routine will be executed.

Altered Reg = A,F,B,C,H,L

PRO-LOG CORPORATION **PROGRAM ASSEMBLY FORM**

HEXADECIMAL			MNEMONIC			TITLE DATE
PAGE ADR	LINE ADR	INSTR.	LABEL	INSTR.	MODIFIER	COMMENTS
17	0 0	F3	(VARIABLEA)	DSI		DISABLE INTERRUPT STRUCTURE
	1	07		RLA		CONVERT REG A TO TABLE ADDRESS
	2	07		RLA		$0_{10} \leq$ REG A $\leq 63_{10}$
	3	5F		LDE	A	
	4	16		LDDI		
	5	00		—	00	
	6	21		LDPI	HL	
	7	34		—		
	8	17		—		
	9	19		ADP	HL,DE	
	A	7E		LDAN	(HL)	OUTPUT DATA TO CHANNELO
	B	5F		LDE	A	MODE REGISTER
	C	D3		OPA		
	D	DB		—		
	E	23		ICP	HL	OUTPUT DATA TO CHANNEL O
	F	7E		LDAN	(HL)	CONTROL PORT
17	1 0	D3		OPA		
	1	DC		—		
	2	23		ICP	HL	CHECK FOR LOWER HALF
	3	7B		LDA	E	WRITE
	4	07		RLA		
	5	07		RLA		
	6	07		RLA		
	7	07		RLA		
	8	5F		LDE	A	
	9	D2		JP	CO	
	A	1F		—	UPPERHALF	
	B	17		—		
	C	7E		LDAN	(HL)	OUTPUT LOWER HALF
	D	D3		OPA		
	E	D8		—		
	F	7B	UPPER HALF	LDA	E	CHECK FOR UPPER HALF WRITE
17	2 0	0F		RRA		
	1	D2		JP	CO	
	2	28		—	RST INTR	
	3	17		—		
	4	23		ICP	HL	OUTPUT UPPER HALF
	5	7E		LDAN	(HL)	
	6	D3		OPA		
	7	D8		—		
	8	3E	RST INTR	LDAI		RESET INTERRUPT LATCH
	9	01		—		
	A	D3		OPA		
	B	DF		—		
	C	FB		ENI		ENABLE INTERRUPT AND RETURN
	D	C9		RTS		
	E					
	F					
	0		LOOKUP			
	1					
	2					
	3					
	4		LOOKUP TABLE			MODE REGISTER } DATA SET O
	5					CONTROL PORT } TYP OF ALL
	6					COUNT LOWER BYTE } DATA SETS
	7					COUNT UPPER BYTE
	8					
	9					
	A					
	B					
	C					
	D					
	E					
	F					

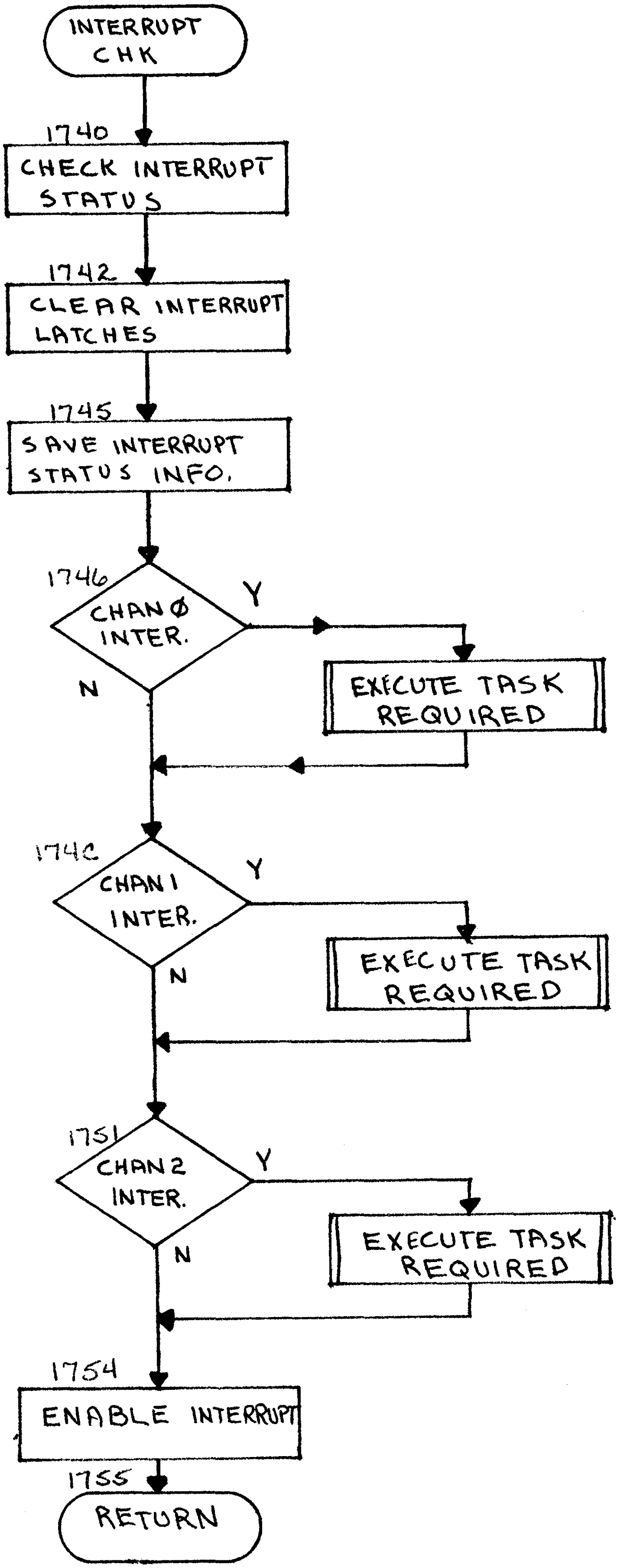

VARIABLEA
1700
CONVERT ACC TO DELAY ADDRESS
170B
LOAD AND OUTPUT MOD CON. WORD
170F
LOAD AND OUTPUT CHAN CONTROL WORD
1712
LOWER HALF
Y
N
171C
OUTPUT LOWER HALF
171F
UPPER HALF
Y
N
1724
OUTPUT UPPER HALF
1728
RESET AND ENABLE INTERRUPTS
RETURN

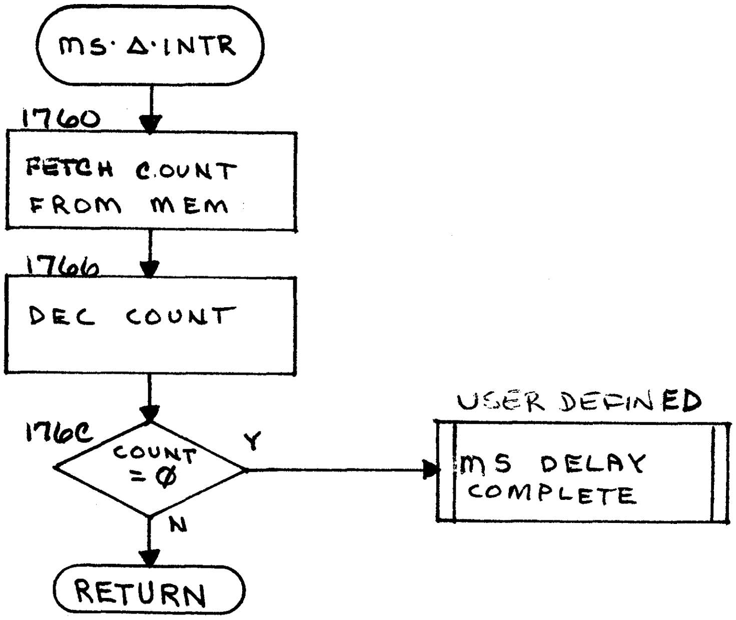
PRO-LOG CORPORATION

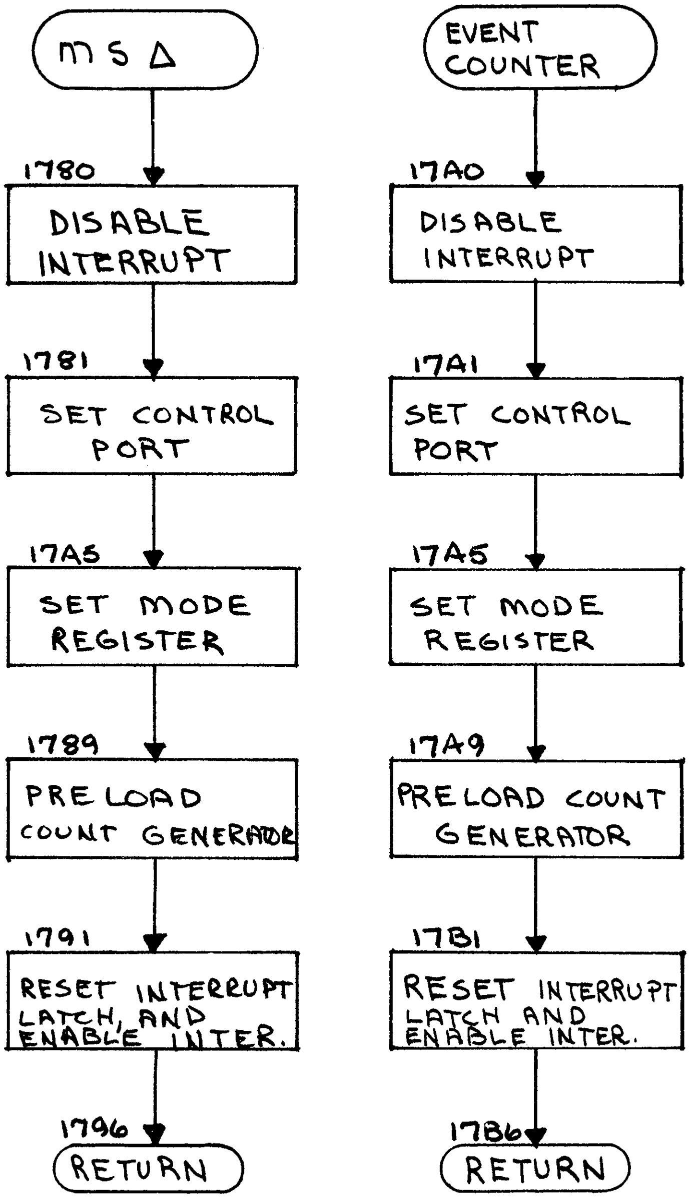
PROGRAM ASSEMBLY FORM

HEXADECIMAL			MNEMONIC			TITLE DATE
PAGE ADR	LINE ADR	INSTR.	LABEL	INSTR.	MODIFIER	COMMENTS
17	4 0	DB	(INTERRUPT·CHK)	IPA		INPUT INTERRUPT STATUS PORT
	1	DF		—	PORT DF	
	2	D3		OPA		CLEAR INTERRUPT LATCHES
	3	DF		—	PORT DF	
	4	OF		RRA		ADJUST AND SAVE STATUS INFO
	5	F5		PSP	A,F	
	6	DC		JS	C1	CHECK IF CHAN ZERO INTER.
	7			—	(INTR·CH·0)	
	8			—		
	9	F1		PLP	AF	RETRIEVE AND CHECK IF CHAN
	A	OF		RRA		ONE INTER.
	B	F5		PSP	A,F	
	C	DC		JS	C1	
	D				(INTR·CH· 1)	
	E					
	F	F1		PLP	AF	RETRIEVE AND CHECK IF CHAN
17	5 0	OF		RRA		TWO INTR.
	1	DC		JS	C1	
	2			—	(INTR·CH·2)	
	3			—		
	4	FB		ENI		ENABLE INTERRUPT AND
	5	C9		RTS		RETURN
	6					
	7					
	8					
	9					
	A					
	B					
	C					
	D					
	E					
	F					

17	6 0	21	(MS·A·INTR)	LDPI	H,L	LOAD COUNT TO B&C
	1			—	MS DELAY COUNT	
	2			RAM	RAM	
	3	46		LDBN	(HL)	
	4	23		ICP	H,L	
	5	4E		LDCN	(HL)	
	6	ØB		DCP	B,C	DECRIMENT REG. PAIR B&C
	7	78		LDA	B	AND STORE
	8	B1		ORA	C	
	9	71		STAN	(HL)	
	A	2B		DCP	H,L	
	B	7Ø		STAN	(HL)	
	C	CA		JS	Z1	IF COUNT ≠ 0 RETURN
	D					
	E			—		
	F	C9		RTS		
	0					
	1					
	2					
	3					
	4					
	5					
	6					
	7					
	8					
	9					
	A					
	B					
	C					
	D					
	E					
	F					


INTERRUPT CHK
1740
CHECK INTERRUPT STATUS
1742
CLEAR INTERRUPT LATCHES
1745
SAVE INTERRUPT STATUS INFO.
1746
CHAN Ø INTER.
Y
N
EXECUTE TASK REQUIRED
174C
CHAN 1 INTER.
Y
N
EXECUTE TASK REQUIRED
1751
CHAN 2 INTER.
Y
N
EXECUTE TASK REQUIRED
1754
ENABLE INTERRUPT
1755
RETURN


MS·Δ·INTR
1760
FETCH COUNT FROM MEM
1766
DEC COUNT
176C
COUNT = Ø
Y
N
USER DEFINED
MS DELAY COMPLETE
RETURN


MSΔ
1780
DISABLE INTERRUPT
1781
SET CONTROL PORT
17A5
SET MODE REGISTER
1789
PRE LOAD COUNT GENERATOR
1791
RESET INTERRUPT LATCH, AND ENABLE INTER.
1796
RETURN
EVENT COUNTER
17A0
DISABLE INTERRUPT
17A1
SET CONTROL PORT
17A5
SET MODE REGISTER
17A9
PRELOAD COUNT GENERATOR
17B1
RESET INTERRUPT LATCH AND ENABLE INTER.
17B6
RETURN

PRO-LOG CORPORATION

PROGRAM ASSEMBLY FORM

HEXADECIMAL			MNEMONIC			TITLE DATE
PAGE ADR	LINE ADR	INSTR.	LABEL	INSTR.	MODIFIER	COMMENTS
17	80	F3	(EVENT COUNTER)	DSI		DISABLE INTERRUPT
	1	3E		LDAI		SET CONTROL PORT
	2	BF		—	CH·3·CONTROL	
	3	D3		OPA		
	4	DE		—	PORT DE	
	5	3E		LDAI		SET MODE REGISTER
	6	B5		—	CH 3· MODE	
	7	D3		OPA		
	8	DB		—	PORT DB	
	9	3E		LDAI		PRELOAD COUNT GENERATOR
	A			—	LSB COUNT	
	B	D3		OPA		
	C	DA		—	PORT DA	
	D	3E		LDAI		
	E			—	MSB COUNT	
	F	D3		OPA		
17	90	DA		—	PORT DA	
	1	3E		LDAI		RESET INTERRUPT LATCHES
	2	Φ4		—	04	
	3	D3		OPA		
	4	DF		—		
	5	FB		ENI		ENABLE INTERRUPTS
	6	C9		RTS		
	7					
	8					
	9					
	A					
	B					
	C					
	D					
	E					
	F					

17	A0	F3	(ms Δ)	DSI		DISABLE INTERRUPT
	1	3E		LDAI		SET CONTROL PORT
	2	1F		—	CH·3·CONTROL	
	3	D3		OPA		
	4	DE		—	PORT DE	
	5	3E		LDAI		SET MODE REGISTER
	6	Ø7		—	MODE REGISTER	
	7	D3		OPA		
	8	DB		—	PORT DB	
	9	3E		LDAI		LOAD DELAY COUNT
	A	DØ		—	LSB COUNT	
	B	D3		OPA		
	C	DA		—	PORT DA	
	D	3E		LDAI		
	E	Ø7		—	MSB COUNT	
	F	D3		OPA		
17	B0	DA		—	PORT DA	
	1	3E		LDAI		RESET INTERRUPT LATCHES
	2	Ø4		—	Ø4	
	3	D3		OPA		
	4	DF		—	PORT DF	
	5	FB		ENI		ENABLE INTERRUPT·
	6	C9		RTS		RETURN
	7					
	8					
	9					
	A					
	B					
	C					
	D					
	E					
	F					

Appendix A

7308 Strapping Options

Several circuit options are available to the 7308 user. These are selected by the removal and replacement of wire jumpers as shown in this appendix. The options include:

- External clock time reference in place of internal oscillator.
- Clock scaler output changes.
- Generate nonmaskable interrupt (NMIRQ*) instead of INTRQ*.
- Provide clock drive to STD bus.

In order to make these changes you should have the schematic (Pro-Log Document #106922) and assembly diagram (106923) that were shipped with your 7308 card. The diagrams and figures in this manual are for reference only; your card may differ slightly due to design improvements.

External Clock Option

This option is provided to allow the 7308 to use the processor's clock (or equivalent) as a time reference signal. If used, this option places one LSTTL load on the STD bus CLOCK* line (PIN 49).

(Figure A-1: External Clock Input Option)

Locate jumpers A and B in figure A-1. Remove jumper wire A, and add a wire in the jumper B position. If desired you may also inhibit the 7308's internal oscillator by removing the crystal (Y1) and resistors R1, R2.

When programming the 7308 with an external clock, note that the frequency at F0 is 1/2 the input frequency.

Clock Scaler Output Changes

Table A-1 shows the frequencies available at the various clock scaler outputs with the 7308's internal crystal oscillator operating.

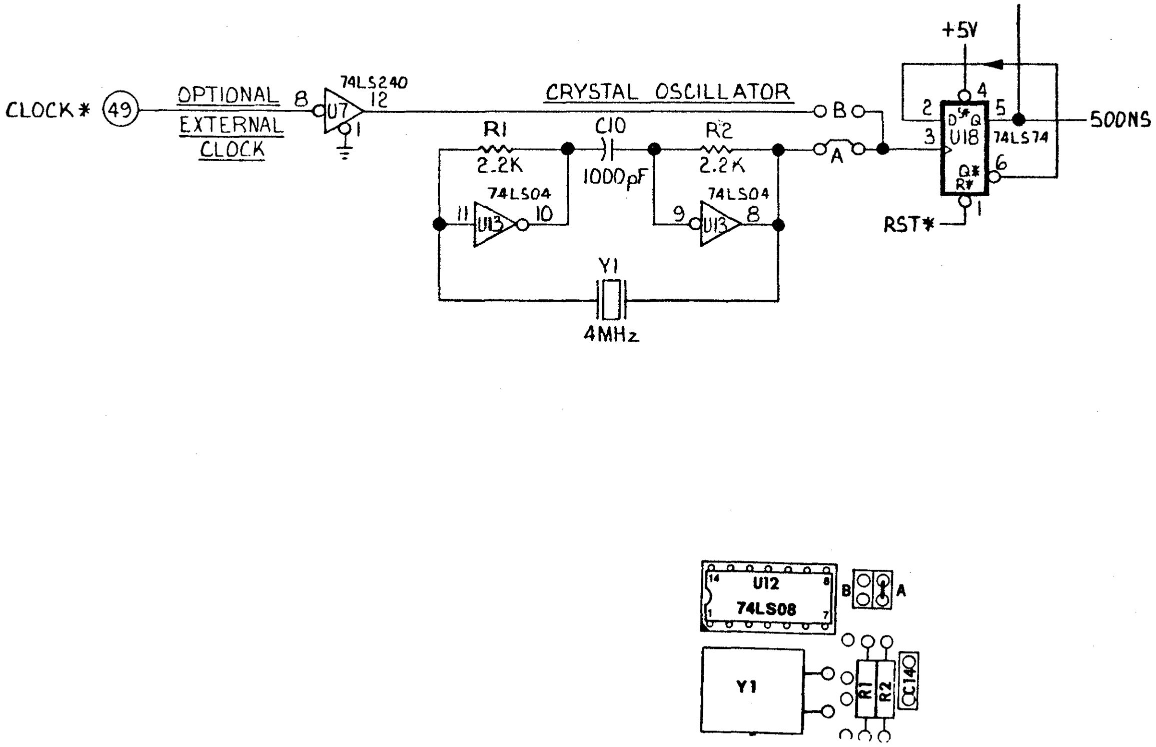


Figure A-1. External Clock Input Option

Table A-1

SCALER TAP	FREQUENCY	PERIOD
C	2.0 MHZ	500 µs
D	1.0 MHZ	1 µs
E	500.0 KHZ	2 µs
F	250.0 KHZ	4 µs
G	125.0 KHZ	8 µs
H	62.5 KHZ	16 µs
J	31.25 KHZ	32 µs
K	15.625 KHZ	64 µs

When shipped, clock input multiplexer position F0 is connected to the 500 ns source by jumper C; F1 to 8 µs by jumper G; and F2 to 128 µs by jumper L.

(Figure A-2: Clock Scaler Output Jumpers)

By removing and replacing jumper wires, the user may connect F0, F1, and F2 to any desired group of scaler outputs, or to STD bus signals, or user interface input signals as desired. Make one and only one connection each to F0, F1, and F2.

Nonmaskable Interrupt Request Output

When shipped, the 7308 is wired to drive the STD bus maskable interrupt line (INTRQ*). If desired, the user can drive the nonmaskable interrupt line (NMIRQ*) instead.

(Figure A-3: Interrupt Output Jumpers)

Locate jumpers M and N in figure A-3. Jumper N is connected when shipped; remove jumer N and connect jumper M to select the nonmaskable interrupt option.

NOTE: Be sure this option is compatible with your processor card. The 7308 is designed to drive DC levelsensitive interrupt inputs only. Some nonmaskable interrupt inputs are edge sensitive, in which case interrupts may be lost if multiple interrupting devices can drive NMIRQ*. For edge sensitive inputs, the interrupting signal should be gated with a signal that produces frequent edges, such as MCSYNC*. Alternately, Pro-Log's 7320 priority interrupt card may be used for edge-sensitive processor card types.

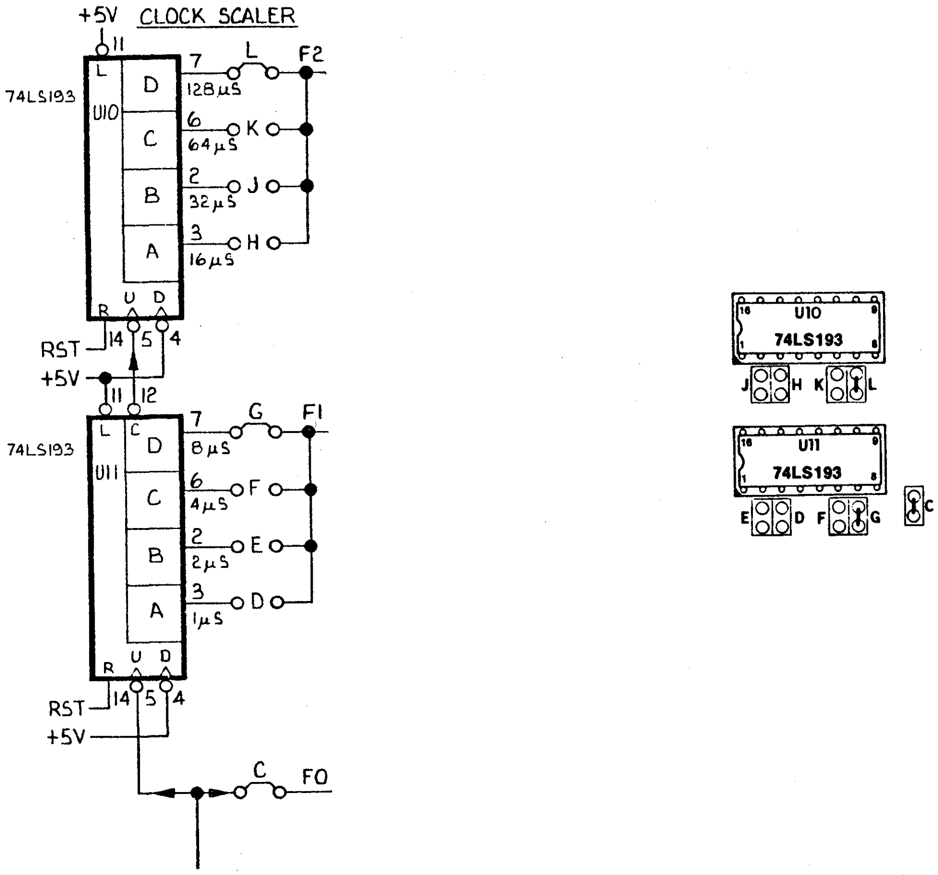


Figure A-2. Clock Scaler Output Jumpers

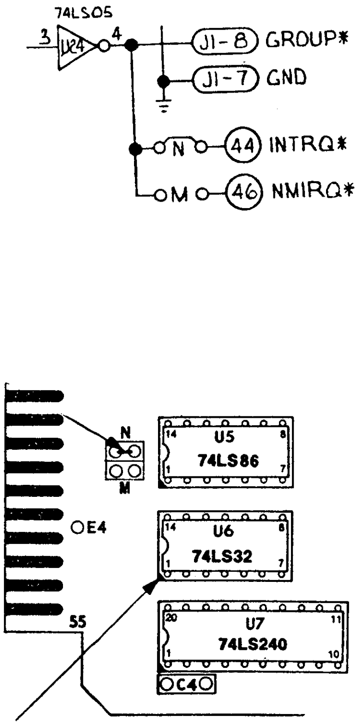


Figure A-3. Interrupt Output Jumpers

Each of the three counter/timer channel outputs is available at three pads: CH. 0 on E1, CH. 1 on E2, and CH. 2 on E3 as shown in figure A-4. Pad E4 is provided to allow any of the three output signals to be jumpered to the STD bus at pin 50 (CTRL*).

(Figure A-4: Optional Clock Drive to STD Bus From 7308)

The CTRL* line may be used to bus one of the 7308's output signals to other STD bus cards. For example, channel 2 can be used to generate an accurate baud rate marker signal as shown in figure A-4.
Channel 2's output would then be connected to the backplane by wiring a jumper from E3 to E4 for use by a UART/USART on another STD bus card.

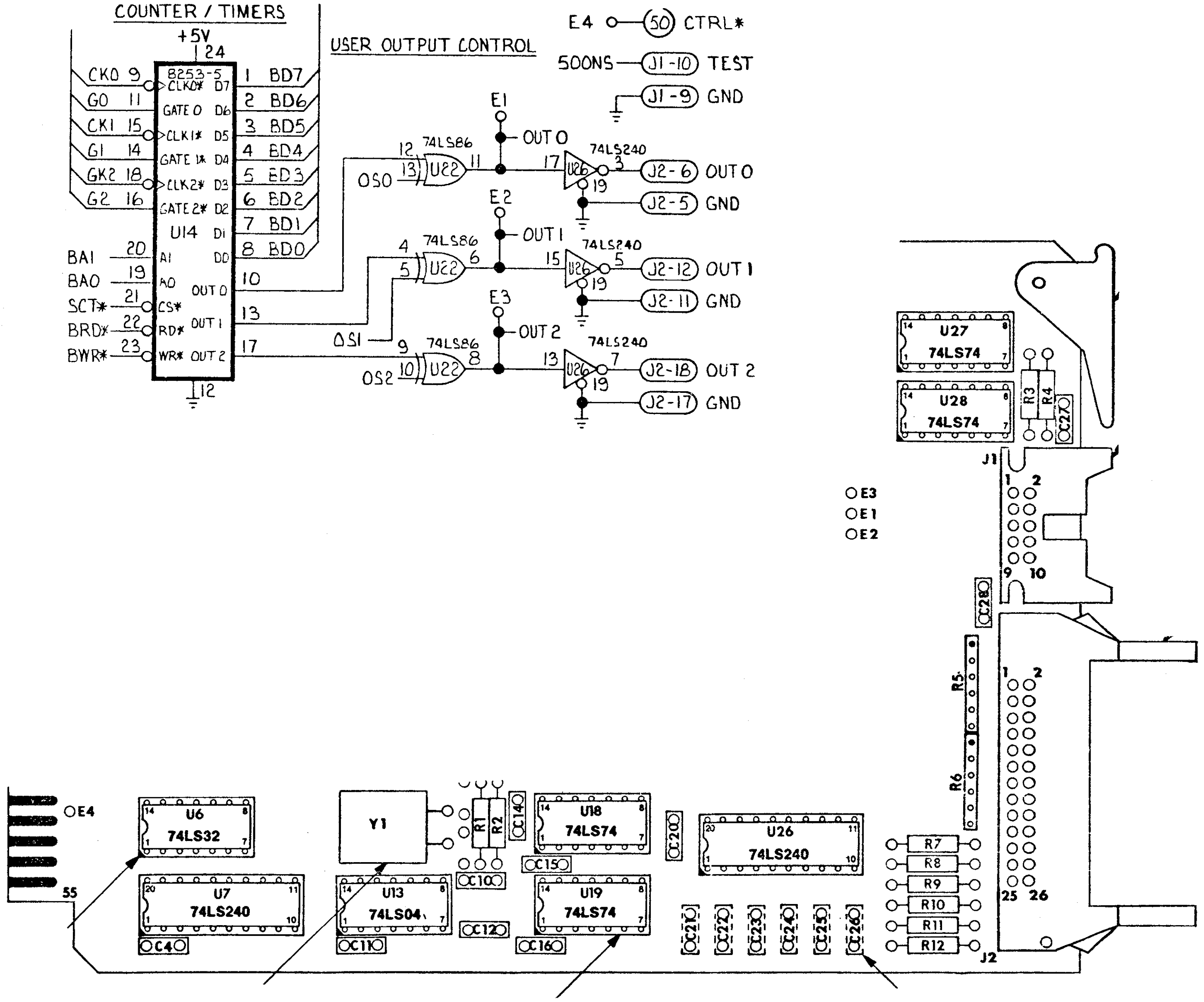
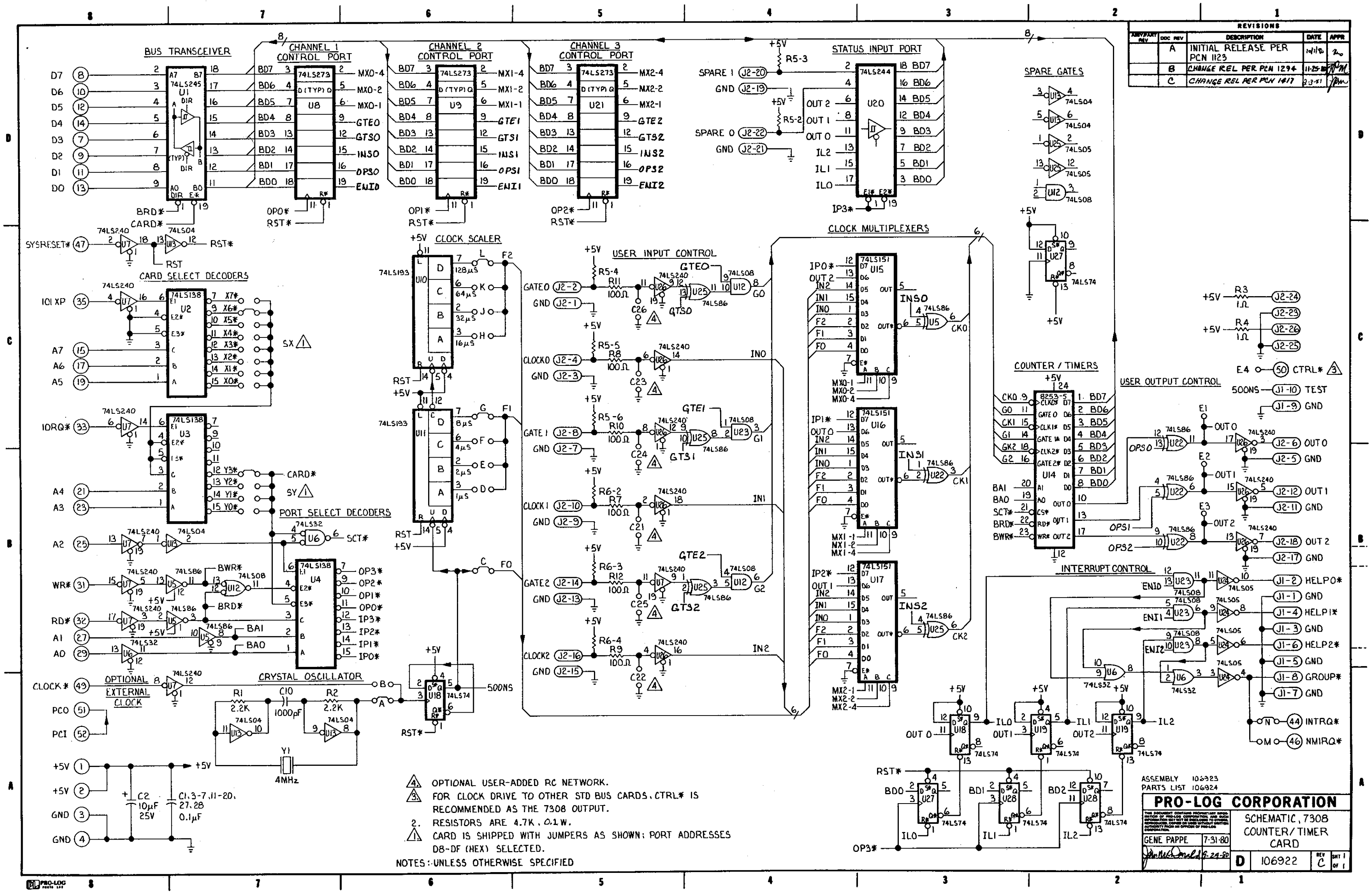


Figure A-4. Optional Clock Drive to STD BUS From 7308

Appendix B

Documentation



ASSY/PART REV	DOC REV	DESCRIPTION	DATE	APPR
	A	INITIAL RELEASE PER PCN 1123	10/7/80	
	B	CHANGE REL PER PCN 1294	11-25-80	
	C	CHANGE REL PER PCN 1417	2-3-81	

NOTES: UNLESS OTHERWISE SPECIFIED

1. CARD IS SHIPPED WITH JUMPERS AS SHOWN; PORT ADDRESSES D8-DF (HEX) SELECTED.
2. RESISTORS ARE 4.7K, 0.1W.
3. FOR CLOCK DRIVE TO OTHER STD BUS CARDS, CTRL* IS RECOMMENDED AS THE 7308 OUTPUT.
4. OPTIONAL USER-ADDED RC NETWORK.

ASSEMBLY 106923
PARTS LIST 106924

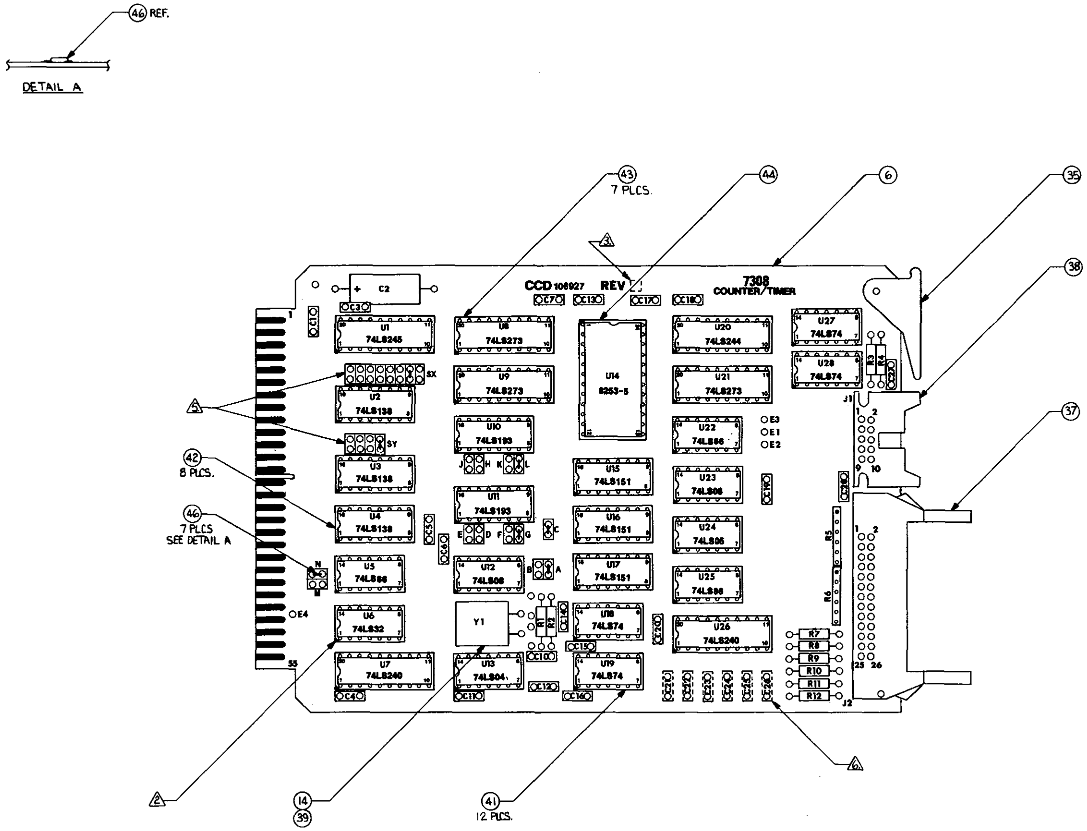
PRO-LOG CORPORATION

GENE PAPPE 7-31-80
6-29-80

SCHEMATIC, 7308
COUNTER/TIMER
CARD

D 106922 REV C SHT 1 OF 1

REVISIONS				
[illegible]	DOC REV	DESCRIPTION	DATE	APPR
	A	INITIAL RELEASE PER PCN 1123	[illegible]	[illegible]
	B	CHANGE RELEASE PER PCN 1274	11-25-80	[illegible]



NOTES: UNLESS OTHERWISE SPECIFIED

1. REFERENCE DESIGNATIONS ARE FOR LOCATING PURPOSES AND MAY NOT APPERR ON ACTUAL PART.

2. DENOTES PIN 1 OF IC'S SOCKETS.

3. IDENTIFY WITH C.C.D. REV LETTER USING RUBBER STAMP.

4. BOARD TO CONFORM WITH ASSEMBLY STANDARDS AS1004.

5. CARD IS SHIPPED WITH JUMPERS AS SHOWN; PORT ADDRESSES DB-DF (HEX) SELECTED.

6. C21 - C26 ARE CUSTOMER OPTION PARTS.

ITEM	DESCRIPTION	REF. DESIGNATION
19	100Ω 5%, 1/4W	R7-12
18	1Ω, 5%, 1/4W, CF	R3,4
17	2.2K, 5%, 1/4W, CF	R1,2
16	4.7K NETWORK	R5,6
15		
14	4 MHZ	Y1
13		
12	1000PF, 50V	C10
11		
10	0.1µF, 50V	C1,3-7,11-20,27,28
9	10µF, 25V	C2
8		
7		
6	106920	PWB

PARTS LIST 106924
SCHEMATIC 106922
CCD 106927

PRO-LOG CORPORATION

ASSY. 7308
COUNTER / TIMER CARD

J. CAREAGA	8-12-80				
CHK'D BY H. CHAVEZ	9-26-80				
Jim McDonald	9-29-80	D	106923	REV B	SHT 1 OF 1

USER'S MANUAL

PRO-LOG
CORPORATION

2411 Garden Road
Monterey, California 93940
Telephone: (408) 372-4593
TWX: 910-360-7082
Telex: 171829