7320
PRIORITY INTERRUPT CARD

PRO-LOG
CORPORATION

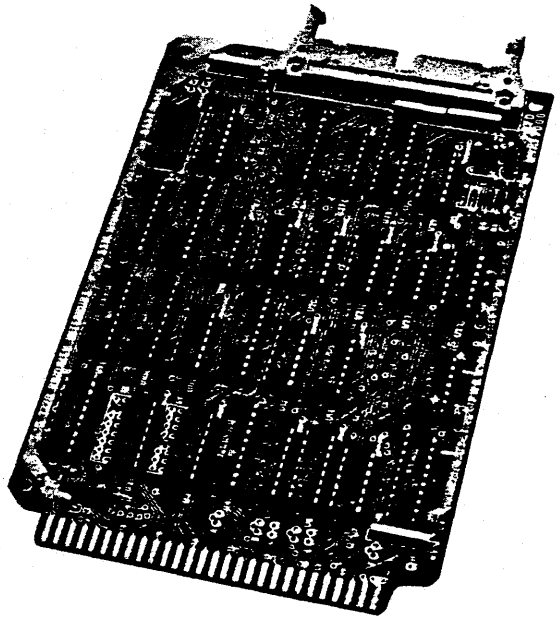
TABLE OF CONTENTS

7000

PRELIMINARY 7320 - PRIORITY INTERRUPT CARD

The 7320 is a universal, 8-input, Priority Interrupt Controller for TTL-compatible interrupt requests originating in the same card rack. The card accepts interrupt requests from other cards, prioritizes them, and generates a single interrupt request to the system microprocessor card. Polled interrupts, 8080-family RESTART instructions, and high-speed single-byte vectored interrupts are all supported with automatic resetting of the interrupt request latches to minimize program service time.

A card level priority system allows 7320s to be cascaded at the user interface connector or across the STD BUS priority chain. Hysteresis buffers, edge-sensitive latches, and interrupt system freeze circuitry are combined to provide error-free operation.



FEATURES

- **Eight, Prioritized, Edge-sensitive Inputs**
- **Program Control of Interrupt Polling, Masking, and Clearing**
- **Programmable Priority Chain Participation**
- **Universal: 1-byte Vector PROM Socket for Z80 (Mode 2) and 2650 RESTART generator for 8085 and Z80 (Mode 0)**
  Polled Mode for 6800 Family and Others
- **Automatic Latch Clearing for Minimized Service Routine**

PRELIMINARY
Document #_____
Eng. _____ Mkt. _____

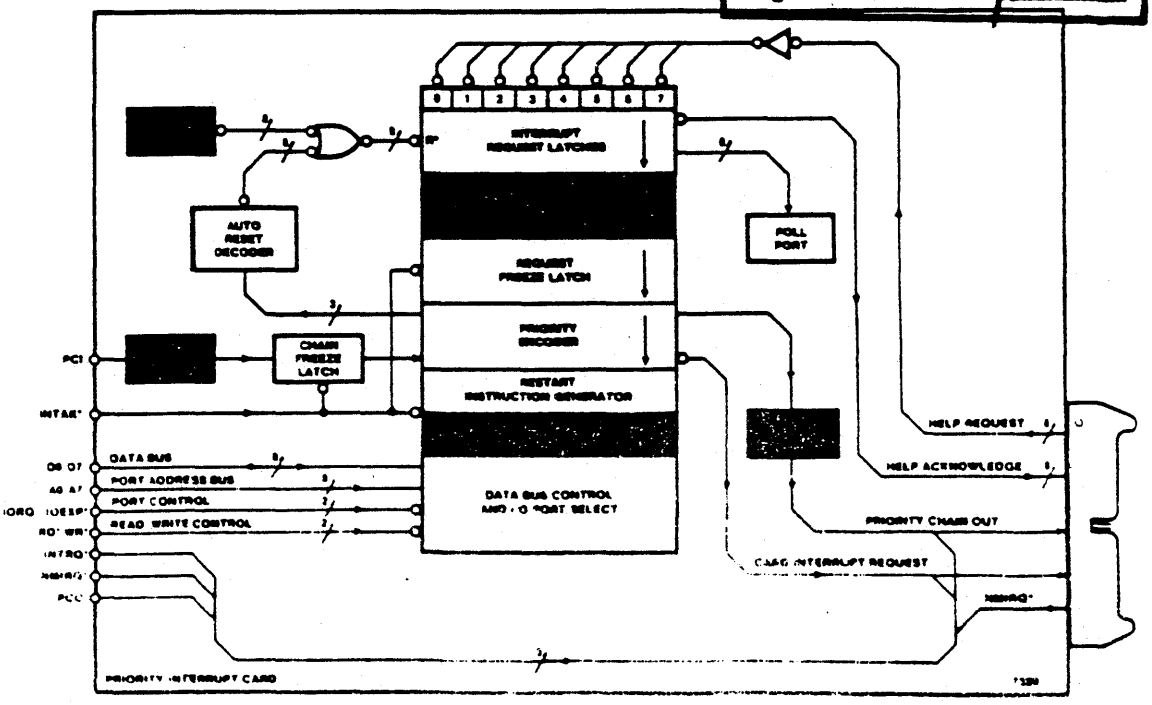


FIG. 1

PRO-LOG CORPORATION

SECTION I -

B. INTRODUCTION TO INTERRUPT SYSTEMS

> INTERRUPT: To pre-empt a process in such a way that it can be resumed; . . . a function which, by reason of pre-established priority, is able to seize the process in progress and cause to be performed a process of higher priority.
>
> (IEEE Standard Dictionary)

Interrupt is a time-oriented function that allows the processor to take action outside of its normal sequence of operations with little or no decision making on its part. The interrupt request signal, if enabled by the processor's program, is able to override the program instruction flow and force a jump to a different set of instructions. It is essential that the processor be able to resume its previous activity once the interrupt function is complete, without any alteration in the original function except the passage of time.

Interrupt systems have been used to perform the following kinds of operations, among others:

a. Servicing asynchronous, unpredictable events without using program overhead to test for the event.
b. Sequencing complex, multifunction programs in an orderly fashion (e.g. timesharing a processor system among several users, or providing a time interval to step multitasking real-time software along).
c. Simplifying program design by moving some decision-making from software to the interrupt hardware.
d. Shortening the processor's reaction time to an external event and improving time resolution in fast control systems.

Since the need to implement an interrupt system is usually based on a shortage of available processing time, the interrupt scheme generally must minimize the response time required to service an interrupt request. An ideal scheme would require no program overhead to prepare the processor for the interrupt; would perform the interrupt without altering any of the interrupted program's data, including the content of registers, memory, and I/O ports; and would restore the processor to its pre-interrupt condition in zero time once the interrupt was completed. Naturally some time is required for these functions in real microprocessors, but the time expenditure is quite small with most microprocessor types.

Kinds of Interrupt Systems

Microprocessor chips are limited in the number of signals that can flow in and out of the circuit because of a limited number of pins on the chip's package. 40-pin chips typically dedicate 2 to 6 pins to the interrupt function. An application with more interrupting devices than the processor has interrupt request pins will require some form of communication between the requestor and the processor. The identifying information passing from the requestor to the processor is called the vector because it provides a direction for program flow. The vector's nature defines the kind of interrupt system being implemented, as follows:

a. Implied Vector Interrupts. In the simplest case, there are few enough external interrupt-causing devices in the system that a different interrupt request pin on the processor chip can be dedicated to each requestor. When a particular input has an active request, the processor can go directly to the interrupt service routine written for the specific device causing the interrupt. No further decision-making or information exchange is necessary to identify the requestor.

Processor chip interrupt pins with Implied Vector characteristics always go to the same specific memory address when that input is active. The processor does not attempt to read a vector code from the interrupting device. The table in Figure 2 shows the Implied Vector address for several processors with this type of input.

PROCESSOR TYPE	INTERRUPT NAME	IMPLIED VECTOR ADDRESS (HEX)
8048	INT*	003
8085	TRAP	0024
	5.5	002C
	6.5	0034
	7.5	003C
Z80	NMI*	0066
	INT*(mode 1)	0038

*Low level active

FIGURE 2: IMPLIED VECTOR ADDRESSES

Figure 3 shows a conceptual diagram of an STD BUS processor system which uses Implied Vector Interrupts. In this case the various interrupt request pins are each driven by only one requesting device. Thus, when an interrupt occurs there can be no doubt as to which device caused it. The processor is forced to a different memory address in each case.

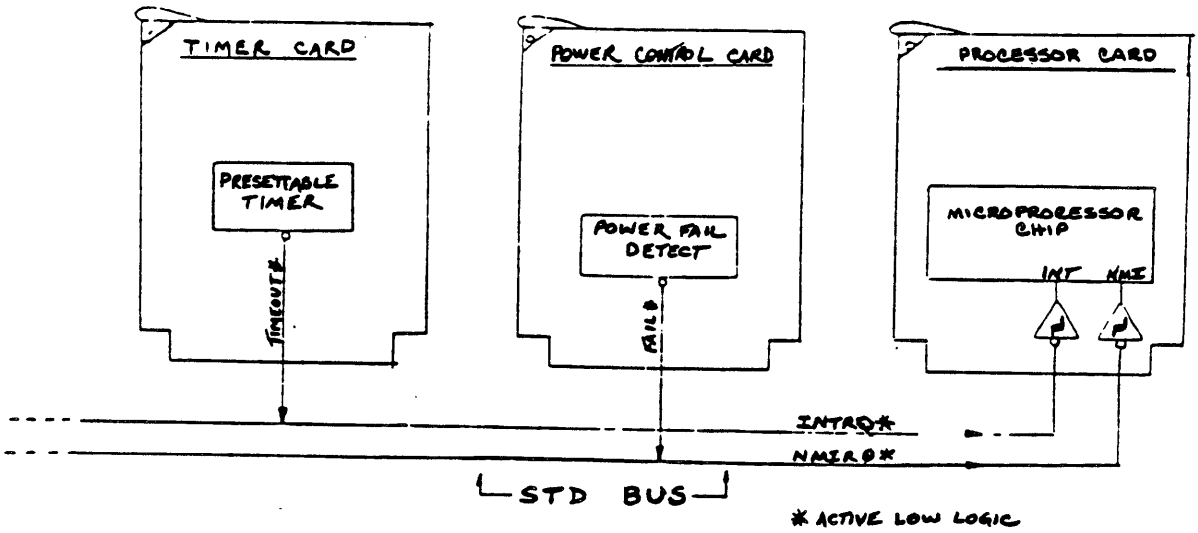


FIGURE 3: CONCEPTUAL DIAGRAM OF STD BUS SYSTEM: IMPLIED VECTORS

b. Supplied Vector Interrupts In this type of interrupt system, multiple devices can all drive the same interrupt request pin on the processor chip. When the processor responds to the interrupt, it issues the Interrupt Acknowledge signal (INTAK*) or a combination of signals from which INTAK* can be derived. INTAK* causes the interrupting device to supply an identifying code (vector) to the processor.

Depending on the microprocessor type, and the particular mode of interrupt operation; the vector code can be a full or partial memory address employing direct, indirect, or relative addressing which the processor uses to find the start of the proper interrupt service routine; or the vector can be an instruction to be executed out of sequence (see 8080-Family Considerations). Since the identifying vector code is supplied by the interrupting device itself (or by a Priority Interrupt Controller, such as the Pro-Log 7320), the processsor expends little or no time identifying the requestor.

In the implementation of Supplied Vector interrupts, the INTAK* signal itself is used as a "read memory" strobe which causes the vector memory (RAM, ROM, PROM, switch register, etc.) to place the vector code onto the system data bus. The Supplied Vector system clearly depends on the generation of an INTAK* signal to inhibit the processors's main memory and to read the vector memory. Supplied Vector interrupt systems for several microprocessor types are summarized in Figures 4 and 5.

PROCESSOR TYPE	INTERRUPT NAME	VECTOR DESCRIPTION	COMMENT
8080 8085	INT INTR	Any 1,2,3-byte instruction. See 8080-family considerations.	Processor generates one INTAK* pulse for each byte of multibyte instructions.
Z80	INT* (mode 0)	Any 1,2,3,4-byte instruction. See 8080-family considerations	Generates INTAK* only on opcode of instruction.
Z80	INT* (mode 2)	1-byte vector is low order indirect address for implied jump-to-subrouting instruction: internal interrupt register supplies the high order byte providing a 16 bit address	Generates one INTAK* pulse
6800/02/08/09	IRQ* NMI* FIRQ* (6809)	Vector stored in 2 specific addresses per interrupt input.	Reads from 2 sequential locations in page FF of memory.
2650	INTREQ	1-byte vector is relative offset for memory page 00 table.	Generates one INTAK* pulse

FIGURE 4: SUPPLIED VECTOR INTERRUPT SUMMARY

Figure 5 shows how the INTAK* signal is generated by the processors in Figure 4:

PROCESSOR TYPE	SOURCE OF INTAK
8080	Directly available from Processor Status Word latch, which is a standard support component in 8080 systems.
8085	Available at a processor chip pin
Z80	The M1* signal is ANDed with the IORQ* signal external to the processor chip.
6800/02/08	Address decoder external to the processor chip.
6809	The BA signal is ANDed with the BS signal external to the processor chip.
2650	Available at a processor chip pin.

FIGURE 5:

SUMMARY OF INTERRUPT ACKNOWLEDGE (INTAK*) GENERATION

8080-Family Considerations: The 8080, 8085, and Z80 (mode 0) interrupt systems all allow the processor to read and execute an out-of-sequence instruction from the data bus when responding to an interrupt. As noted in Figure 4, INTAK* reads this instruction into the processor via the data bus. The processor issues INTAK* in place of the normal "read memory" signal combination, so the processor's main memory is disabled during INTAK*.

The instruction read into the processor can be any 1, 2, 3, or (Z80 only) 4-byte instruction in the instruction set. If this happens to be any type of JUMP instruction, the processor is vectored to a specific memory address. Alternately any logical, arithmetic, or machine control instruction (such as increment a register or memory, halt, etc.) could be executed.

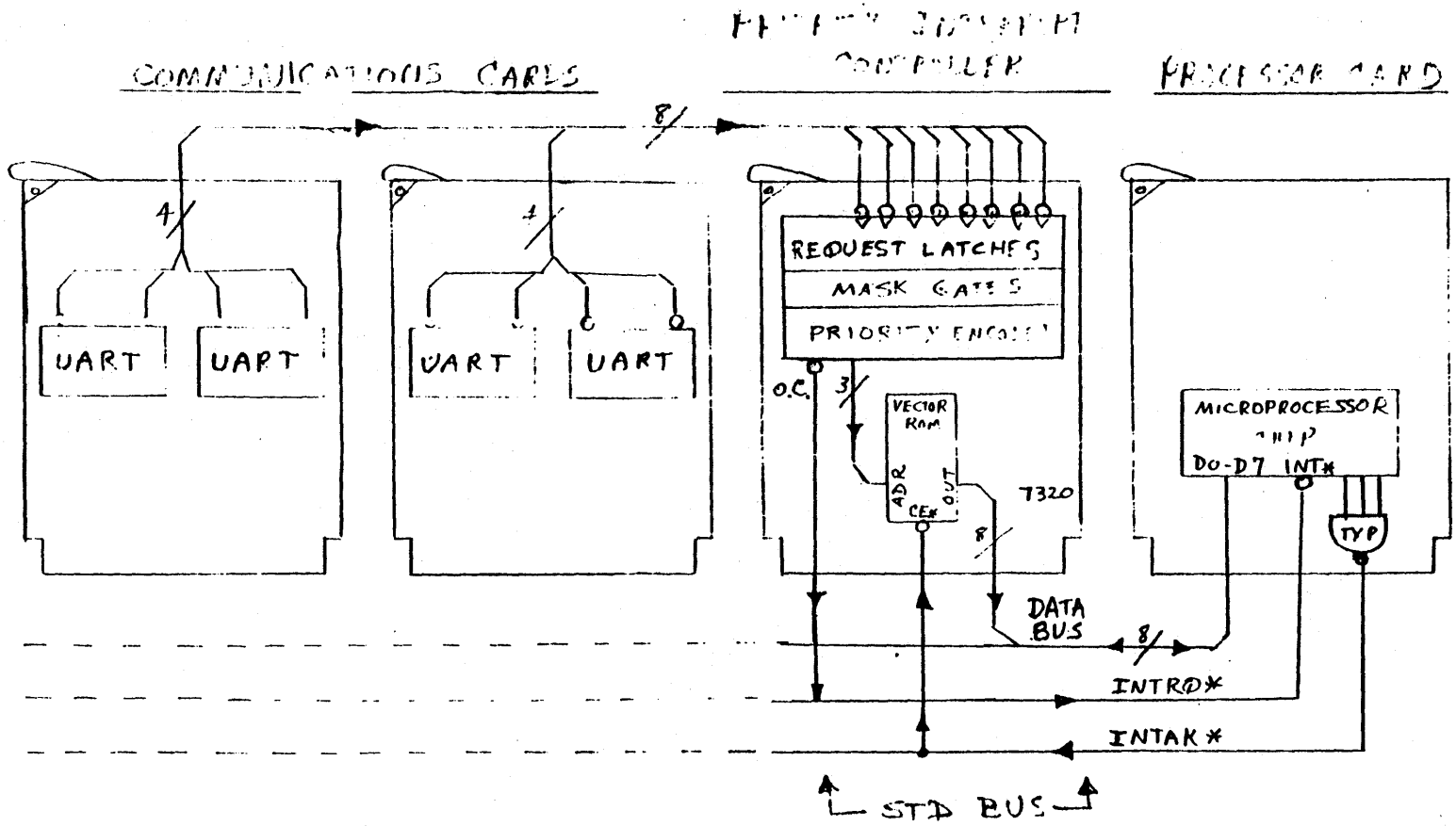
A special set of RESTART instructions are provided. These are 1-byte implied-address subroutine jump instructions.

With RESTART instructions, however, only 8 interrupt requests can be vectored by without additional polling by the program. Note that the RESTART instructions are limited to eight specific memory locations (hex 0000, 0008, 0010, 0018, 0020, 0028, 0030, and 0038). For unlimited vectoring anywhere in memory, the multibyte jump instructions are required.

System Example

Figure 6 shows a conceptual diagram of an STD BUS system using Supplied Vector interrupts provided by a Priority Interrupt Card (7320). In this example, four UART devices on two cards can generate a total of eight interrupt request signals. The Priority Interrupt card latches the interrupt requests from the UARTs to alleviate any critical pulse timing specifications for the UARTs. The Priority Interrupt card also contains circuitry to selectively control interrupt masking, eliminate timing ambiguities, assign priorities, and generate a single ORed interrupt request (INTRQ*) signal from the eight inputs. It also contains a vector memory device which supplies the highest active priority vector code to the processor via the Data Bus during INTAK*.

* Low level active logic



* LOW LEVEL ACTIVE
O.C. OPEN COLLECTOR

FIGURE 6

c. Polled Interrupt System

Polled interrupts represent the "computed vector" approach to interrupt systems, in contrast to Implied Vector or Supplied Vector systems which are predominately hardware-controlled when responding to an interrupt.

Polled interrupt systems replace most of the interrupt hardware with programmed instruction sequences which determine the source of the interrupt via input ports and perform other logical operations, such as the assignment and arbitration of priority. The vector is fetched or generated by the program as a result of these operations, generally from a lookup table where the start addresses of the various device service routines are stored.

In interrupt polling, multiple devices drive the same interrupt request pin on the processor chip, but the processor lacks the capability to read a supplied vector, or that capability is not cost-effective in the application.

For example, if several devices drive one of the Implied Vector interrupt inputs shown in Figure 1, the processor must be programmed to poll (read) the various interrupt requestors to determine which one caused the interrupt. If more than one is found to be active, the polling routines make decisions about the relative priority of the active devices and respond to them in sequence.

Polling can be combined with vectoring to produce a flexible and cost-effective interrupt system. For example, the designer may wish to use the 8080-family RESTART instructions in a system with more than eight interrupting devices. The RESTART vectors produce fast response and can be implemented at low cost. Only a few polling instructions at each of the eight RESTART memory addresses are needed to make additional tests to determine which is the active requestor. Any spare input port lines are suitable for this function.

In general, polled interrupts offer unlimited flexibility since any interrupt scheme can be implemented in software. Response time to the interrupt is slower (by the time required to execute the polling instructions) than an equivalent Supplied Vector interrupt. However, polling offers low cost, simplified system and program design, flexibility, and a high degree of processor-independent standardization in comparison with other vectored interrupt schemes.

Figure 7 shows a simple polled interrupt system in which two UARTs and two timing circuits can generate a total of six interrupt request signals, each requiring a different service routine in the program. These signals are ORed together by gates on the card modules and again by wire-ORable drivers at the INTRQ* trace on the STD BUS backplane. The processor's INT input is activated by any of the six signals. When it responds to the interrupt, the processor is programmed to read the local input ports on each of the card modules, then determine what situation exists in the system and what to do about it. The program vectors itself to the proper service routine using a combination of table lookup and indirect jump techniques.

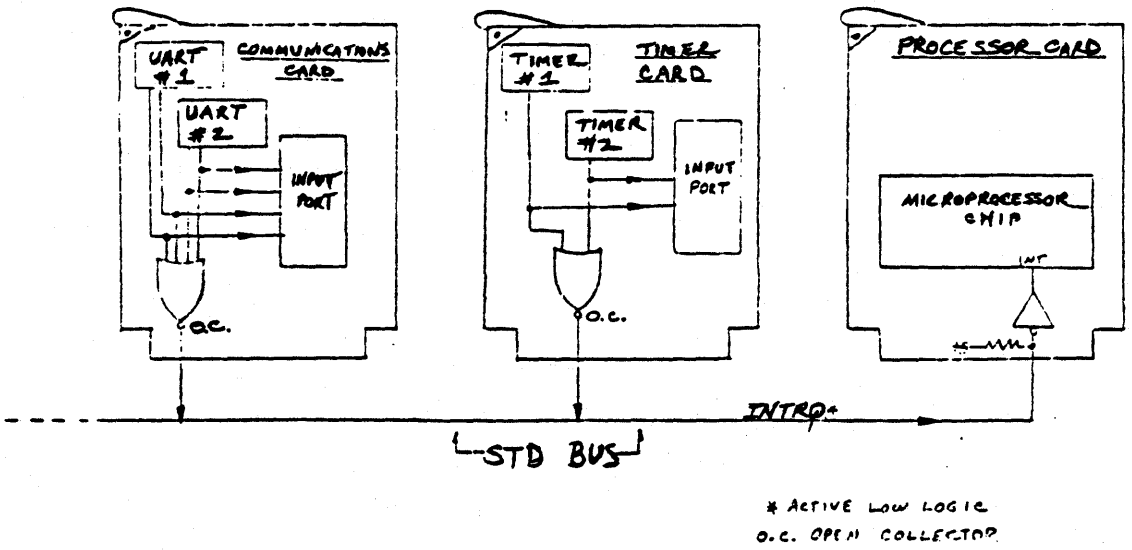


FIGURE 7: POLLED INTERRUPT SYSTEM

Automatic Processor Activity at Interrupt

Interruptable microprocessors share certain key characteristics which give them the ability to be interrupted. Among these are:

1. Instruction to enable/disable interrupt allows the processor to ignore the interrupt request signals from maskable interrupt inputs. This instruction is used anytime the processor is performing a time-critical function that shouldn't be interrupted.

2. Ability to complete the instruction in progress before responding to an enabled interrupt. The processor could not be reasonably expected to recover from an instruction that did not completely execute prior to response to the interrupt. Accordingly, processors respond to the interrupt only at the completion of the current instruction. Exception: macro-type instructions (e.g. Z8000's multiply,divide) are extremely long iterative-loop instructions which use other registers for counting and memory control. These can sometimes be interrupted at the completion of any given loop, since the control registers can be saved and restored when the interrupt is over.

3. Use of the subroutine stack circuitry to save the address of the next instruction in the interrupted program. The requirements of interrupt save/restore functions are similar to those of subroutines, so the program instruction counter's content is generally saved automatically at interrupt on the subroutine stack. Most processors also provide instructions (PUSH and PULL or POP) which allow the programmer to save other data on the stack at interrupt. The return-from-interrupt function is usually just a variation on the return-from-subroutine function.

4. Automatic disabling of DC level sensitive interrupt inputs by the processor eliminates crucial timing requirements for the interrupt request signal by preventing erroneous multiple interrupts from the same request.

The following table lists specific activity by several processors when an interrupt is responded to and gives other features provided for fast response.

PROCESSOR TYPE	INTERRUPT ACTION
8080	Interrupt disabled by reset and automatically upon response to an interrupt; must execute ENI (enable interrupt) before next interrupt can be accepted. Saves current address on the subroutine address stack. No other data saved automatically, but provides PUSH and PULL (POP) instructions for saving all internal registers.
8085	Identical to 8080, but has four additional interrupt inputs (nonmaskable TRAP and maskable 5.5, 6.5, 7.5). Provides RIM and SIM instructions which monitor and control the interrupt system.
Z80	Mode 0 is identical to 8080; also gives a choice of program selectable mode 1 (implied vector) or mode 2 (supplied vector). Resets to 8080-compatibility. Interrupt service time is improved by additional internal register banks. Switching banks (2 fast instructions) saves storing registers on stack with 4 to 6 long instructions. RTI (return from interrupt) instruction is provided to control serial priority chain among Z80's peripheral chips.
6800	Automatically stores all internal registers on the stack when interrupted. Also does this with WAI (wait-for-interrupt, or halt instruction) so that it's ready for instant response to an interrupt. Automatically disables IRQ* upon response, and by reset.
6809	Identical to 6800, but adds FIRQ (fast interrupt request input) which stores only the program counter and flag register on the stack to provide faster response than the 6800- type request inputs. Also requires less circuitry than the 6800 to generate INTAK*

*Low level active

FIGURE 8: AUTOMATIC INTERRUPT ACTIVITY BY PROCESSOR TYPE

CHECKLISTS

The following checklists represent problems that must be addressed to some extent by the designer when implementing an interrupt system.

a. Interrupt Circuit Characteristics

1. Is the microprocessor's interrupt input edge sensitive, level sensitive, or both? (i.e. Is the input latched or gated?) Is it protected against noise and multiple responses to the same signal?
2. Is the interrupt request input high or low level active, and is it voltage-compatible or is voltage translation required?
3. Are RC networks and/or hysteresis buffers and/or optical isolators necessary to prevent false interrupts caused by noise?
4. Is the interrupt system speed comptabible with the processor? Is the request pulse too short to be polled? Is the vector memory's access time as fast as the system's main memory? If a card-level priority propagation system (daisy chain) is used, can a signal propagate along the chain in the alotted time or is look-ahead gating required?
5. If a Priority Interrupt Controller (PIC) is used to augment the processor's interrupt capability, is the PIC universal or processor signal dependent? What support signals does it require to operate? If used with the wrong processor type, what tradeoffs are involved? If the system has more interrupts than the PIC has inputs, how is expansion accomplished?
6. If system speed/cost requirements do not warrant vectored interrupts, can each of the interrupting devices be polled and masked by the program? Is the interrupt request signal driver wire-ORable so that all devices can drive the processor's input pin?

b. Interrupt Service Routine

In general the interrupt must not destroy or alter the data that was being used by the interrupted program. The things to save or protect include:

1. Data in the processor's internal accumulator(s), flag register, and other data registers.
2. Address register content, including the program counter, stack pointer, index registers, and address modification (offset) registers.
3. Data contained in latching output and input ports.
4. Bank selection, mode selection, and enable/disable decisions made by the interrupted program.
5. Data contained in the processor's RAM.
6. Conditions and data existing in hardware external to the processor chip prior to the interrupt, where applicable.

Potential Problems With System Timing

1. What happens if devices with different priorities request an interrupt at the same instant?
2. What happens if a high priority device requests an interrupt while a lower priority device is being serviced?
3. Can a high priority device request interrupts so frequently that a lower priority device will never be serviced?
4. In response to a low-priority device interrupt, the processor may issue INTAK at the same instant that a high priority device interrupts. Will the processor see the high order vector, the low order vector, or simply the ripple effect of vector bits changing? What defines the last possible instant that a new interrupt request can change the system response?
5. Perhaps most importantly, what happens to the system hardware under the processor's control during an interrupt? Will a disaster occur if the interrupt comes at the wrong time or lasts too long? Must the interrupt routine also contain instructions which protect or "keep alive" some operation being controlled by the interrupted program?

Priority Interrupt Controllers such as Pro Log's 7320 card offer solutions for items 1 and 4 above (when application recommendations are followed or modified appropriately to suit special circumstances). The other items, however, are entirely application dependent and require the careful attention of the system design engineer.

SECTION 2: FUNCTIONAL DESCRIPTION

The 7320's interrupt request (HELPn*) inputs are falling edge sensitive and buffered with hysteresis gates for improved noise rejection. The eight identical request latches can be polled, masked, and cleared by onboard I/O ports.

Priority is fixed with HELP0* having the highest priority and HELP7* having the lowest. The HELPn* input connections are made via signal/ground .025" post connectors approved by the STD BUS Manufacturer's Group. These connections allow for the following applications:

a. wire (or twisted pair) inputs from adjacent cards;
b. Mass terminated ribbon cables with ground-signal-ground conductor allocation from adjacent cards; and
c. Mass terminated or discrete wiring to a signal conditioning card (Opto-22 or equivalent).

The 7320 supports single byte vectored interrupts for processors capable of generating the INTAK* signal, and polled interrupts for other processors. 8080-family processors (including 8085 and Z80 mode 0) are additionally supported by a special RESTART instruction generator which allows poll-free vectoring of up to eight interrupts. The vector generator chip is a user-supplied 32 x 8 PROM and requires removal of the RESTART generator chip if used.

The 7320 supports the Priority Chain as an STD BUS daisy-chained slot dependent priority propagation scheme using the PCI/PCO backplane traces. Alternately the user may connect the daisy chain across the card front for card slot independence. Priority propagation to another 7320, Z80 peripheral, or other Priority Interrupt Controller device is controlled both in hardware and in software.

Circuit Description

The 7320 consists of three major circuits: (Refer to schematic 105942)

A. Onboard Control Ports. Input/Output ports and associated address decoding and card select circuitry provide program control of the interrupt latches, allowing latch reading (polling), masking, and clearing plus card level priority chain control.

These ports allow the program to exercise logical control over the interrupt inputs, augmenting the microprocessor's interrupt control instructions and allowing for logical operations such as interrupt priority rotation. For example, a high priority input may be held off indefinitely or cleared according to logical decisions in the program.

The 7320 appears to the system processor as an array of three output ports and one or two input ports. (See Interrupt Response Circuitry, below) The second input port is available if the processor does not generate INTAK*.

The port-address mapping selected when the 7320 is shipped can be found in Figure 11, Section 4. The user may change the port address assignments by changing jumper wires SX and SY. Refer to Section 3 (Mapping).

B. Interrupt Request Latches (U19 through U22)

Interrupt Request Latches form the user interface to the 7320. These eight identical TTL circuits are pulled up to Vcc by a 1K resistor and buffered with hysteresis buffers to improve noise immunity. Each input is falling edge sensitive: the input must be returned high, then fall low to set the interrupt request latch a second time. Each of the eight interrupt request inputs (HELP0* through HELP7*) produces an associated HELP Acknowledge output (HACK0* through HACK7*). Use of these outputs is optional; they are provided for situations where a 2-wire handshake with the interrupting device is required, or for convenient indication that the interrupt request has been received and latched by the 7320 (the HACK signals can sink 20mA LEDs.)

The eight interrupt request latches are set only by the eight user inputs, and can be reset in one of three ways:

a. By the SYSRESET* Signal generated by the system processor card at power-on or in response to the PBRESET* signal.
b. By the program at any time a decision is made to cancel an interrupt request or hold a latch in the reset state.
c. Automatically by the 7320 itself when the processor generates the INTAK* (interrupt acknowledge) signal in response to the specific interrupt request.

Each of the eight interrupt request latches has an associated Mask Enable Gate (U11 and U13) which is available to the program. If the gate bit is zero, the corresponding interrupt request latch is prevented from generating an interrupt. Masking the gate does not clear the latch nor prevent it from being set by an incoming request signal, and the state of the latch can always be polled by the program regardless of whether or not it is enabled.

C. Interrupt Response Circuitry

The response provided by the 7320 depends on the application. As shipped, the card is capable of unlimited interrupt polling as described above, and is equipped with a RESTART generator device (74LS244 at U18) which generates the eight 8080-family RESTART instructions used in the 8080 8085, and Z80 mode 0 interrupt vectoring schemes.

Also provided is a socket at U24 for a 74S288 or equivalent 32 x 8 bipolar PROM which is supplied by the user to support any single-byte address or instruction vector, depending on the microprocessor and interrupt mode in use. (See Appendix A) The vector PROM is recommended

for Z80 mode 2 systems, 2650 or similar microprocessors requiring a single-byte interrupt vector, or 8080-family processors where single byte instructions (in place of the RESTART instructions and vectors) are desired at interrupt. This PROM allows up to four interrupt schemes to be stored in the same PROM for manufacturing simplicity, with the four separate 8-byte vector groups being strap selectable on the 7320.

In all cases the 7320 responds to any enabled interrupt request from the user with a single wire ORable INTRQ* (interrupt request) signal to the system processor card. If the system processor card generates the INTAK* (interrupt acknowledge) signal, the 7320's vector generator or RESTART generator may be used and the RESTART instruction or address/instruction vector byte will be placed on the STD data bus during INTAK* to be read by the system processor. If the processor card does not generate INTAK* and the equivalent cannot be derived elsewhere in the system, then interrupt polling is the only available alternative. In this case the 7320's RESTART generator or vector generator can be use as an auxilary input port such that the vector information or octal-encoded top priority request code can be read into the processor as input port data. If used as input ports, the 7320's interrupt response generators still automatically clear the corresponding interrupt request latch to improve the polled interrupt response time.

The 7320 can generate an interrupt only if the PCI(Priority Chain In) signal is active, and the card drops the PCO(Priority Chain Output) signal when it or a higher level card driving it are requesting an interrupt. Both the PCI and PCO signals are duplicated by software controlled port bits, allowing the program to disable either the 7320 card or all low-order interrupting cards participating in the chain Note that either the card-slot-dependent priority chain at the STD BUS backplane, or a card-slot-independent scheme at the user interface connector may be implemented. PCO must be reset by program instruction at the end of the interrupt. (1)

Both the interrupt request outputs and the priority chain are held constant by onboard freeze latches (U12) during INTAK* or input port read times. This feature is provided to prevent address vector misreads if a higher order interrupt is accepted during the time the 7320 is supplying an address vector to the system processor.

A spare input to the interrupt system freeze circuit is provided to allow for special processor requirements. For example, Z80-based systems normally connect this input to the STATUS 1* (M1*) line to prevent interrupt system changes during M1 (instruction opcode fetch) to allow for priority chain propagation delay across several interrupting cards. Note that systems with a large number of prioritized interrupt

(1) RTI (Return from Interrupt) Z80 instruction is not decoded by the 7320, so an output instruction is required to reset the priority chain.

(2) Not required for Pro-Log 7803 (Z80) Processor Card

requests or a high speed processor may require the inclusion of wait states (implemented outside the 7320 using the WAITRQ* signal) to allow adequate priority chain settle time. Refer to the dynamic characteristics of the 7320 in Section 5 , and to the data sheet for the processor card used.

Other user options include the defeat of the automatic latch resetting circuit, defeat of the 7320's priority chain participation, and generation of NMIRQ* in place of INTRQ* (allowing prioritization of the nonmaskable interrupt, or the 8085's 5.5, 6.5, and 7.5 interrupt inputs or equivalent in different microprocessor types) with provision for chopping edge-sensitive interrupt inputs. Refer to Appendix C.

SECTION 3 - MAPPING

I/O PORT ADDRESS SELECTION

The 7320 occupies four input port addresses and four output port addresses. When shipped these port addresses are CC, CD, CE, and CF hexadecimal. If only one 7320 card is used in the system and no existing ports are already assigned these addresses, then the address selection is generally arbitrary and no change need be made. However the port addresses may be assigned any four sequential addresses in the range 00-FF hexadecimal as follows:

a. Locate card select decoders U2 and U3 (74LS138) adjacent to the STD BUS edge connector. Each decoder device has a dual row of pads which form decoder output select matrices as shown in figure 9 . Make one and only one connection to each of the matrices; the wire jumper supplied with the 7320 at each matrix must be removed if a new connection is to be made.

b. Figure 10 shows where to place jumper straps to obtain any four sequential port addresses. Use the lowest of the four addresses desired when determining the jumper positions, and note that this lowest address must end in one of the following hexadecimal numbers: 0, 4, 8, or C (eg 28 is the lowest of the group 28, 29, 2A, 2B). Find the most significant hex address digit along the vertical axis, then find the intersection along the horizontal axis which coincides with the desired group of least significant hex digits. At that intersection, find the two connections to to be made to matrix SX (adjacent to U2) and matrix SY (adjacent to U3).

   For example, port addresses 28, 29, 2A, and 2B are selected by connecting jumpers to X1 and Y2, soldered in place.

The jumper matrix pads are on 0.10 inch (0.25 cm) centers. Wirewrap posts may be soldered in if frequent changes are anticipated. For permanent connections use short wire jumpers.

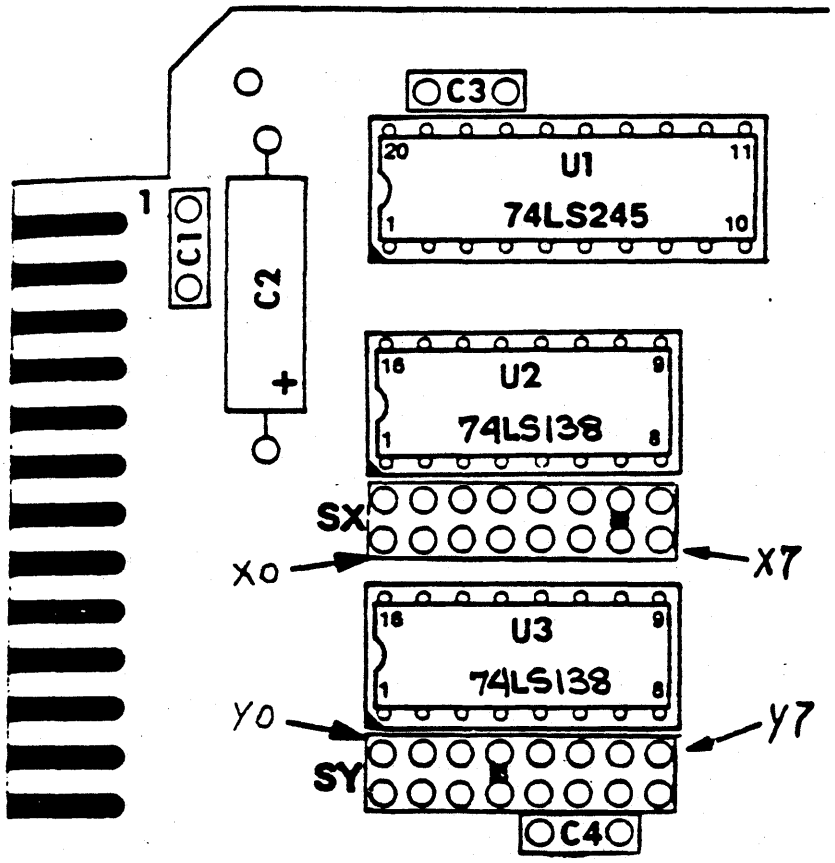


FIGURE 9: LOCATION OF 7320 ADDRESS DECODER OUTPUT SELECT MATRICES

MOST SIGNIFICANT HEX ADDRESS	LEAST SIGNIFICANT HEX ADDRESS																JUMPER SELECTION X & Y
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	X0 Y0				X0 Y1				X0 Y2				X0 Y3				X AND Y
1	X0 Y4				X0 Y5				X0 Y6				X0 Y7				
2	X1 Y0				X1 Y1				X1 Y2				X1 Y3				
3	X1 Y4				X1 Y5				X1 Y6				X1 Y7				
4	X2 Y0				X2 Y1				X2 Y2				X2 Y3				
5	X2 Y4				X2 Y5				X2 Y6				X2 Y7				
6	X3 Y0				X3 Y1				X3 Y2				X3 Y3				
7	X3 Y4				X3 Y5				X3 Y6				X3 Y7				
8	X4 Y0				X4 Y1				X4 Y2				X4 Y3				
9	X4 Y4				X4 Y5				X4 Y6				X4 Y7				
A	X5 Y0				X5 Y1				X5 Y2				X5 Y3				
B	X5 Y4				X5 Y5				X5 Y6				X5 Y7				
C	X6 Y0				X6 Y1				X6 Y2				X6 Y3				
D	X6 Y4				X6 Y5				X6 Y6				X6 Y7				
E	X7 Y0				X7 Y1				X7 Y2				X7 Y3				
F	X7 Y4				X7 Y5				X7 Y6				X7 Y7				

FIGURE 10: 7320 ADDRESS DECODER JUMPER PROGRAMMING

SECTION 4 - OPERATION AND PROGRAMMING

PROGRAMMING INFORMATION

The 7320 occupies four consecutive I/O port addresses with port addresses CC through CF hexadecimal selected by jumper straps when shipped. The user may select any other four consecutive addresses (see Section 3). Of the port addresses occupied by the card, only three output ports and one or two input ports are used. Figure 11 shows the bit/function of each of the ports.

A. Poll Port 0: Address CC Hex (Input to Processor Card)

This 8-bit port allows the states of the eight interrupt request (HELF) latches to be monitored by the program. Reading this port always returns the actual states of the latches, regardless of the control conditions imposed by the other ports. If the latch is set, a logical one is returned; if reset, a logical zero is returned.

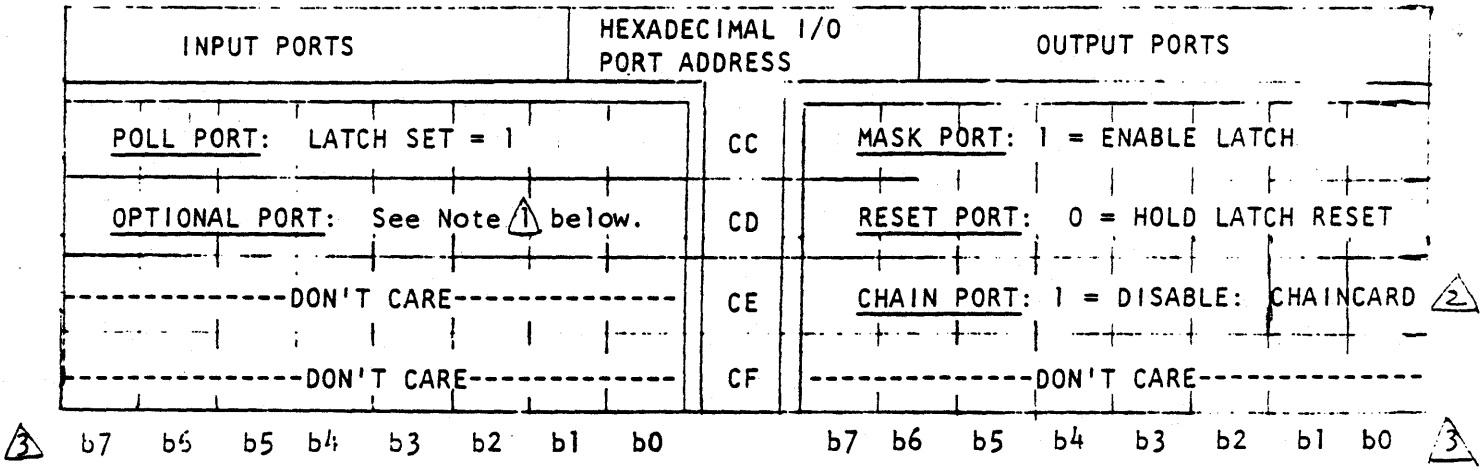
B. Mask Enable Port: Address CC Hex (Output from Processor card)

This 8-bit port allows the program to prevent the interrupt request latches from generating an interrupt by blocking their outputs. If a bit is written high by the program, the corresponding interrupt request latch is enabled and allowed to generate an interrupt request INTRQ*; if the bit is written low, the latch's output is blocked.

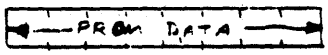
Note that the SYSRESET* signal (at power-on, for example) clears this port, effectively blocking all interrupt requests from the card. Note also that disabling the latch output by writing its mask enable bit low or by resetting the card does not prevent the latch from being set nor does it prevent input Port CC from reading the true state of the latch.

C. Poll Port 1 (Vector or Restart): Address CD Hex (Input to Processor card)

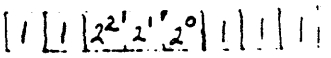
This optional 8-bit port is available if the system microprocessor does not generate the INTAK* signal or if interrupt vectoring is not required in the system. Moving a jumper strap on the 7320 allows either the



NOTES: 1 Input Port CD is present if user makes jumper-selectable card option. See APPENDIX C. Optional data formats are:



Vector Generator



RESTART Generator

2 Output Port CE bits 2 thru 7 are DON'T CARE Set bit 1 to disable 7301's PC0, or set bit 0 to disable both PC1 and PC0.

3 Bit numbers indicated correspond to interrupt requests HELP0* - HELP7*.

FIGURE 11: I/O PORT BIT FUNCTION AND PROGRAMMING TABLE

Vector Generator PROM or the RESTART instruction generator to become an input port. Reading this input port gives a vector or RESTART code corresponding to the highest priority request input that is currently active.

If the RESTART Generator is used, reading Input Port CD will return the 3-bit octal-encoded number corresponding to the priority (lowest numbered) enabled interrupt request in bits 3, 4, and 5.

Whether the Vector Generator or RESTART Generator is used, the highest enabled interrupt-request latch producing the Input Port CD response will be automatically cleared at the completion of the read operation.

D. Reset Port: Address CD Hex (Output from Processor card)

This 8-bit port allows the program to selectively reset one or any combination of the interrupt request latches. It may simultaneously be used to hold selected latches reset and prevent them from becoming set. If an output port bit is written low by the program, the latch is held reset; if high, the latch is enabled to be set by the corresponding HELP signal.

Note that the SYSRESET* signal clears this port and holds the eight interrupt request latches reset until initialized by the program.

E. Card Control Port: Address CE Hex (Output from Processor card)

This 2-bit output port allows the 7320 to be disabled by setting a single bit (bit 0), and also allows all lower-order priority interrupting cards to be disabled from the 7320 (bit 1). When these bits are set high, the corresponding function is disabled; when cleared, the functions are enabled. A multicard interrupt system can be disabled by setting bit 0 in the highest order 7320 card.

Note that these bits are cleared by the SYSRESET* signal, thus enabling the priority chain at power-on reset time.

F. Unused Ports: Address CE Hex (Input) and CF Hex (Input and Output)

The 7320 occupies these port addresses even though they are not used, and no other I/O card in the system can share these addresses or bus contention with unpredictable read data will result from input reads.

THEORY OF OPERATION (Refer to 7320 Schematic 105944)

The 7320 Priority Interrupt Card consists of three major functions:

- I/O Control Ports including card select and read/write circuitry, and the STD BUS interface.
- User Interface (Figure 12) consisting of buffered interrupt request latches and latch reset gating.
- Priority Control (Figure 13) consisting of mask gates priority chain control, the priority encoder, latch reset decoding, and vector generator.

Control Ports

The 7320 has three output ports and one or two input ports to control the card. The card and one of its port addresses are selected by the Processor when the IORQ* and IOEXP* control inputs are both active, one of the user-selected address bit combinations (A0-A7) is present and either RD* or WR* is active.

DATA BUS TRANSCEIVER

Card selection is defined as the activation of the 7320's Data Bus Transceiver (74LS245; U1), a process requiring only the presence of IORQ*, IOEXP*, and a user-selected combination of address lines A2-A7. The user chooses the card-select address by connecting one and only one jumper to each of the two pad matrices adjacent to

ADDRESS DECODERS

address decoders U2 and U3 (74LS138). Note that the 7320 is shipped with jumpers at X6 (U2 pin 9) and Y3 (U3 pin 12). This decodes I/O addresses CC, CD, CE, and CF hexadecimal and results in the Data Bus transceivers being enabled any time the Processor Card addresses one of these four port addresses.

In addition to enabling the Transceiver, the CARD* select signal also enables the Port Select Decoders (74LS138; U5 and U6). U6 selects input ports and U5 selects output ports. The input port selector is additionally enabled by the RD* input, and the output port selector is enabled by the WR* signal. Each circuit can decode four ports, of which only three

DATA BUS TRANSCEIVER

output ports and one or two input ports are used. Note, however, that even though some ports are not implemented, the Transceiver remains enabled when the unused ports are addressed. Consequently these port addresses may not be implemented elsewhere in the system or bus contention will result due to multiple devices driving the Data Bus.

The decoded outputs of the Port Decoders are used to gate input ports and strobe output ports via the STD data bus. The RD* input is buffered and used to control the direction of the Data Bus Transceiver.

* low level active

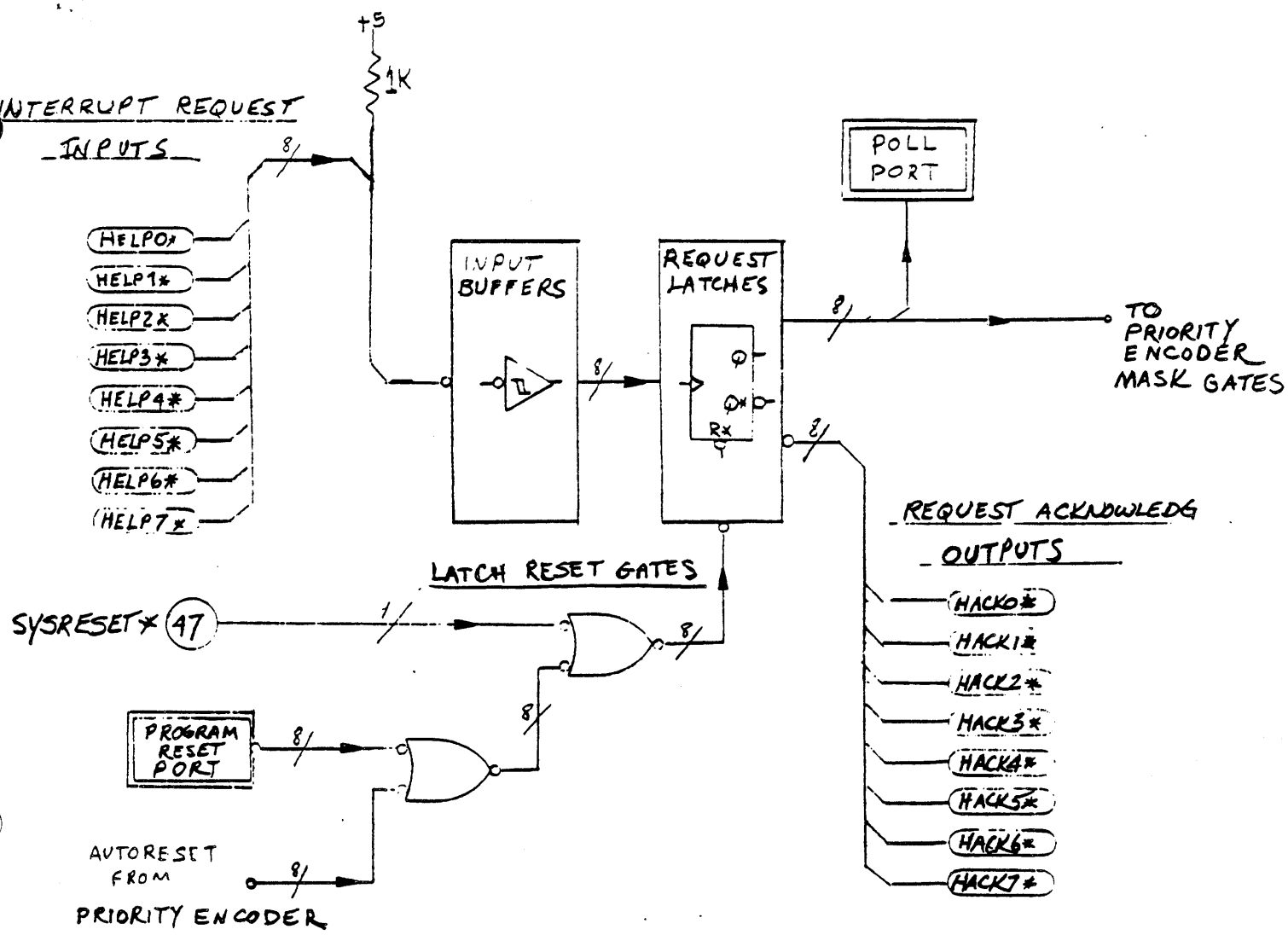


FIGURE 12 : 7320 USER INTERFACE CIRCUIT

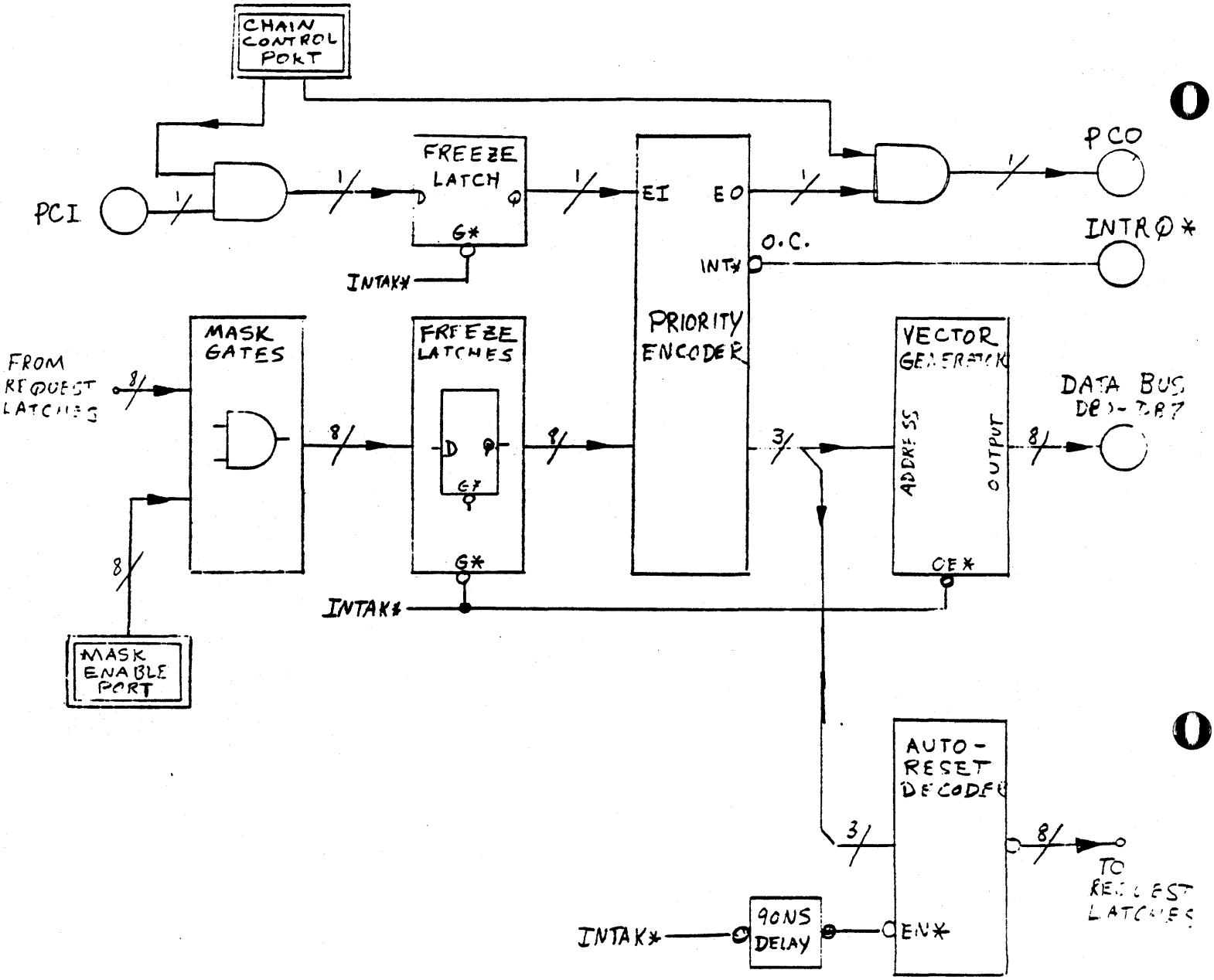


FIGURE 13: 7320 PRIORITY CONTROL CIRCUIT

OUTPUT PORTS

Output Port CC (74LS273; U9) is a latching 8-bit output port strobed by the OP0* output from the Decoder and cleared by the SYSRESET* signal at power-on. The 8 output bits from this port provide the interrupt enable mask.

Output Port CD (74LS27 ; U17) is a latching 8-bit output port used by the program to control the reset input to each of the Request Latches. Bits written low at this port hold the corresponding latches in reset.

Output Port CE (74LS74, U23) is a latching 2-bit output port used to control the Priority Chain path through the 7320. Bits 0 and 1 from this port correspond to the PCI and PCO signals, respectively, and provide the same functions under program control.

Note that U9, U17, and U23 are rising-edge sensitive devices and accept new data at the rising (trailing) edge of the WR* signal.

INPUT PORTS

Input port CC (74LS244; U10) is a gated 8-bit input port used to poll (read) the states of the eight Request Latches. This port drives the system Data Bus through the Transceiver at the falling (leading) edge of the RD* signal.

Input Port CD is similar to Input Port CC, but is used only under special circumstances. See Vector Generator discussion below.

User Interface

HELP REQUEST INPUT BUFFERS

The 7320 accepts up to eight interrupt request inputs (HELP0* through HELP7*). These inputs are pulled up to Vcc (+5V) through 1K resistors to improve noise immunity and to allow each input to be driven by multiple wire-ORed open collector drivers. Each input drives an inverting buffer (74LS240; U28) with 200mV minimum hystersis (400mV typical) for additional noise immunity. The combination of 1K pullup resistor and 74LS240 buffer presents 15 LSTTL input loads maximum. The resistor networks used to pull up the inputs (R7 and R8) may be replaced with higher value networks to reduce input loading if desired.

REQUEST LATCHES

The input buffers drive the clock inputs of eight clocked latches (74LS74: U19, U20, U21, and U22). These are rising edge sensitive clock inputs, so a falling edge at each of the HELPn* inputs causes the corresponding latch to set after inversion by the input buffer. Once set, the latch's inverted output is buffered (74LS244; U26) and provided at User Interface Connector J1 as HELP acknowledge signals (HACK 0* thru HACK 7*).

The use of these signals is optional; they may be used to sink 20mA LED indicators, or as part of a 2-wire handshake with the interrupting device if necessary. Note that by removing and exchanging U26 (74LS244) and U28 (74LS240) both the HELPn* and HACKn* signals may be made high level active.

POLL PORT

The true outputs of the eight interrupt request latches are routed to the Priority Control circuit and to the Poll Port (Input Port CC; 74LS244; U10). By reading the Poll Port the program can determine the states of all eight interrupt request latches simultaneously (1=set, 0=reset).

LATCH RESET GATES

Low level OR gates (74LS08; U14, U18) are used to reset the request Latches. The latches can be reset as follows:

PROGRAM RESET PORT

- The Program Reset Port (Output Port CD; 74LS273; U17) which allows any combination of Request Latches to be selectively reset by an output instruction sequence or to be held reset.

- The Autoreset Decoder (74LS42; U27) automatically selects and resets a specific Request Latch when that request is being acknowledged by the Processor.

- SYSRESET*, issued by the processor card, through it the eight request latches.

C. Priority Control

This section provides the program with control over the interrupt system, synchronizes the interrupt system with the Processor Card, and generates the vector information (optional) needed by the Processor.

MASK GATES

The Mask Gates (74LS00; U11, U13) determine which interrupt request latches are allowed to drive the Priority Encoder. They gate the outputs of the Request Latches with the outputs of the Mask Enable Port (output Port CC; 74LS273; U7). Any logical "1" written to this port by the program will enable the corresponding Request Latch output (bit 0 enables MASK GATE 0.) Since the Mask Enable Port is cleared by SYSRESET* at power on, the mask word must be written to this port before any interrupts can be generated by the 7320.

MASK ENABLE PORTS

FREEZE LATCHES

Two sets of Freeze Latches are provided on the 7320. The first (74LS373; U12) is used to hold the eight outputs of the Mask Gates unchanged, and the second (74LS00; U29) holds the state of the Priority Chain constant. Both Freeze Latches are activated by the INTAK* signal; they act as gates when INTAK* is inactive and as latches when INTAK* is active. Both serve to freeze the conditions present in the interrupt system while the Processor Card uses INTAK* to read the Vector Generator (or RESTART Generator).

INPUT PORT CONVERSION

If the Processor Card does not generate the INTAK* signal (or if vectored interrupts are not wanted in the system) the Vector or RESTART generator can be converted to an input port by opening jumper B and connecting jumper C (see Appendix C). With that modification, the above discussions regarding the INTAK* signal refer instead to the Input Port CD select signal IPI*. Reading Input Port CD then supplies the same codes to the program that would otherwise have been provided to the Processor Card with INTAK*.

PRIORITY CHAIN

Card level priority control (the PCI and PCO signals) are important when more than one 7320 is used in the system, or when the 7320 is used with any other card (particularly those with Z80-family peripheral chips) which can generate a vectored interrupt.

PCI (STD BUS pin 52) and INH* input (J1 pin 2) both serve to inhibit the 7320 if a higher priority interrupt card is requesting an interrupt. PCI is driven if the card is being used in the STD BUS priority chain; INH* can be used if a card-slot-independent priority chain scheme is implemented by the user. Both signals serve the same purpose and only one is generally used for priority chain control.

FREEZE LATCH

The ECI* signal (Output Port CE bit 0) is ANDed with PCI* (74LS32; U7) to produce EPE*, a signal which enables the Priority Encoder (74LS148; U25). See Figure 14 for the Priority Encoder Truth Table. EPE* must be frozen by INTAK* to stabilize the interrupt system while the processor is in the act of responding to the interrupt. A one-bit Freeze Latch (74LS00; U29) is provided to freeze EPE* and produce FEPE* which drives the Priority Encoder's enable input.

EDGE SENSITIVE PROCESSORS

The INH* signal input also serves a purpose unrelated to the Priority Chain. Some microprocessor interrupt-request input pins are edge sensitive. The 7320's normal mode of operation is suitable only for level-sensitive inputs, since its INTRQ* driver transistor will keep the INTRQ* pin active as long as any of the eight Request Latches are set. Thus if two latches were set initially, no signal transition will occur on the INTRQ* output when the processor moves from servicing the first to servicing the second. The INH* signal provides a convenient input for the user to "chop" the INTRQ* signal and produce one transition per processor machine cycle, allowing the card to be used with edge-sensitive microprocessor inputs. Pads are provided on the STD BUS pins MCSYNC*, STATUS 1*, STATUS 0*, and spare pins 5 and 6. Normally connecting INH* by jumper to MCSYNC* will produce a pulsed request signal, providing all other priority conditions are satisfied on the card.

INPUTS									OUTPUTS						SEE NOTE
PCI (note 1)	Interrupt Request Latch 0	1	2	3	4	5	6	7	Priority Code P4	P2	P1	= Hex	INTRQ* (note 2)	PCO (note 3)	
0	X	X	X	X	X	X	X	X	X	X	X	X	1	0	4
1	0	0	0	0	0	0	0	0	X	X	X	X	1	1	5
1	1	X	X	X	X	X	X	X	0	0	0	0	0	0	
1	0	1	X	X	X	X	X	X	0	0	1	1	0	0	
1	0	0	1	X	X	X	X	X	0	1	0	2	0	0	
1	0	0	0	1	X	X	X	X	0	1	1	3	0	0	6
1	0	0	0	0	1	X	X	X	1	0	0	4	0	0	
1	0	0	0	0	0	1	X	X	1	0	1	5	0	0	
1	0	0	0	0	0	0	1	X	1	1	0	6	0	0	
1	0	0	0	0	0	0	0	1	1	1	1	7	0	0	

X = DON'T CARE

NOTES:

1. PCI is high level active at the board edge; PCI = 1 corresponds to EI* = 0 at U25 pin 5.
2. Corresponds to GS* at U25 pin 14.
3. PCO is high level active at the card edge; PCO = 1 corresponds to EO* = 0 at U25 pin 15.
4. 7320 is disabled because a higher-order card in the Priority Chain has an active interrupt.
5. 7320 enabled but idle because it has no active interrupts; Priority Chain signal is passed on to next lower-order card in the chain.
6. 7320 is enabled and has an active interrupt request; Priority Chain is disabled to next lower-order card in the chain.

FIGURE 14 : 7320 PRIORITY ENCODER TRUTH TABLE

Interrupt Request Latches 0	1	2	3	4	5	6	7	RESTART Generator Output	VECTOR GENERATOR ADDRESS SELECTED Jumpers: J1,K1	J1,K0	J0,K1	J0,K0 (note)
1	X	X	X	X	X	X	X	C7	00	08	10	18
0	1	X	X	X	X	X	X	CF	01	09	11	19
0	0	1	X	X	X	X	X	D7	02	0A	12	1A
0	0	0	1	X	X	X	X	DF	03	0B	13	1B
0	0	0	0	1	X	X	X	E7	04	0C	14	1C
0	0	0	0	0	1	X	X	EF	05	0D	15	1D
0	0	0	0	0	0	1	X	F7	06	0E	16	1E
0	0	0	0	0	0	0	1	FF	07	0F	17	1F

1 = SET, 0 = RESET

X = Don't care

NOTE: Jx,Kx refers to condition of jumpers J and K
J1 and K1 indicate jumpers are connected.
J0 and K0 indicate jumpers are open

The Vector Generator addresses shown are the hexadecimal addresses selected according to the combination of jumper conditions and active interrupt request latches.

FIGURE 15 : RESTART GENERATOR / VECTOR GENERATOR ACTION

SECTION 5 - SPECIFICATIONS

7320 PIC ENVIRONMENTAL SPECIFICATIONS

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

FIGURE 16

7320 PIC ELECTRICAL SPECIFICATIONS

RECOMMENDED OPERATING LIMITS				ABSOLUTE NON OPERATING LIMITS		
PARAMETER	MIN	TYP	MAX	MIN	MAX	UNITS
Vcc	4.75	5.00	5.25	0.0	5.50	Volt

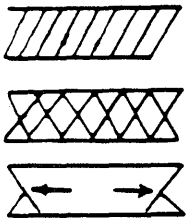
FIGURE 17

STD BUS ELECTRICAL CHARACTERISTICS OVER RECOMMENDED OPERATION LIMITS

PARAMETER	MIN	TYP	MAX	UNITS
Vcc Supply Current		460	770	mA
STD BUS Input Load	See Figure 23			
STD BUS Output Load	See Figure 23			

FIGURE 18

WAVEFORM LEGEND



Transition to high

Don't Care

Must be valid

T = TIMING MEASUREMENT	
SIGNALS	STATUS
P = any HELP* input K = any HACK* output N = INTRQ* A = INTAK* D = Data Bus D0-D7 I = PCI O = PCO	H = High L = Low X = Don't Care V = Valid

7320 DYNAMIC SPECIFICATIONS (See Figure 20)

PARAMETER		SEE NOTE	NANOSECONDS MIN	TYP	MAX
TPLPH	Minimum HELPn* request pulse width		30		
TPLKL	HACKn* (HELP acknowledge) active after any HELPn*			30	55
TPLNL	INTRQ* active after HELPn*	1		100	160
TPLOL	PCO inactive after HELPn*	1		80	150
TALKH	HACKn* inactive after INTAK*	2		175	250
TALPX	HELP0-7* and PCI frozen after INTAK*	3		20	40
TALDV	RESTART instruction access time	4		60	100
TALDV	Vector access time	5		60	100
TILOL TIHOH	Priority chain propagation time	6		90	160

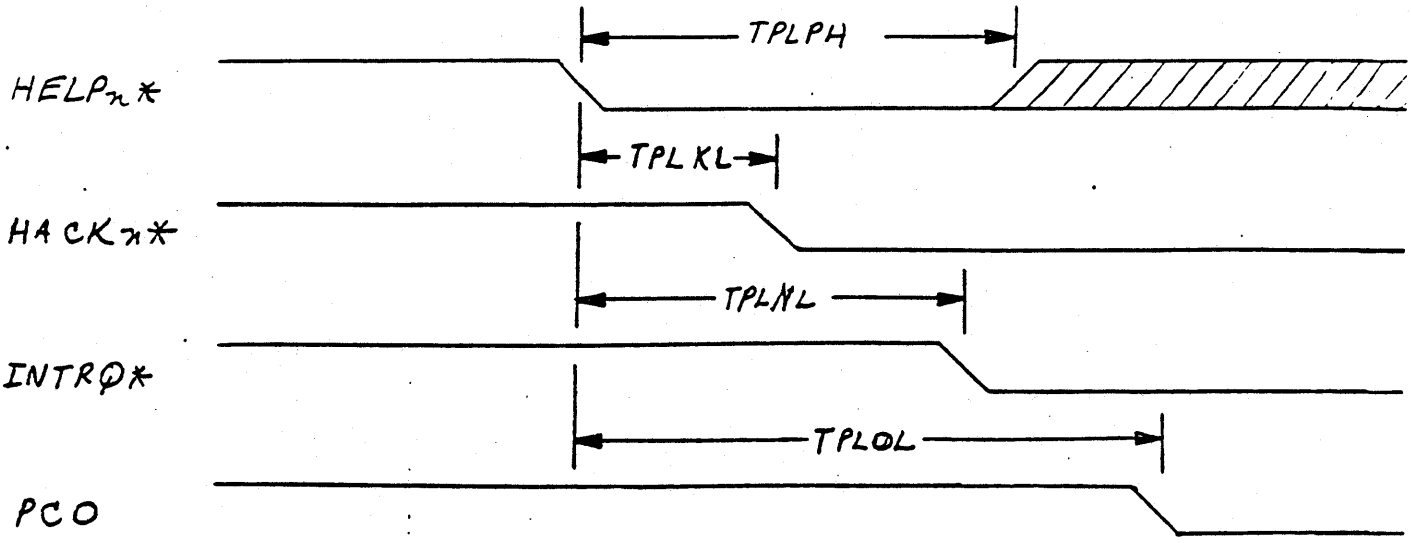
NOTES (Jumpers: refer to schematic 105942)

1. PCI = 1; Card control port Bits 0,1 = 0; Mask enable bit n = 1.
2. Automatic latch reset feature (jumper J) connected.
3. Idle HELPn* latches may be set during this period, but the RESTART instruction or vector will not change.
4. Remove vector generator device.
5. Remove RESTART generator device; vector generator specification is for user-supplied 74S288 device. Several equivalent PROM types may be substituted.
6. 7320 participation jumper (jumper D) connected.

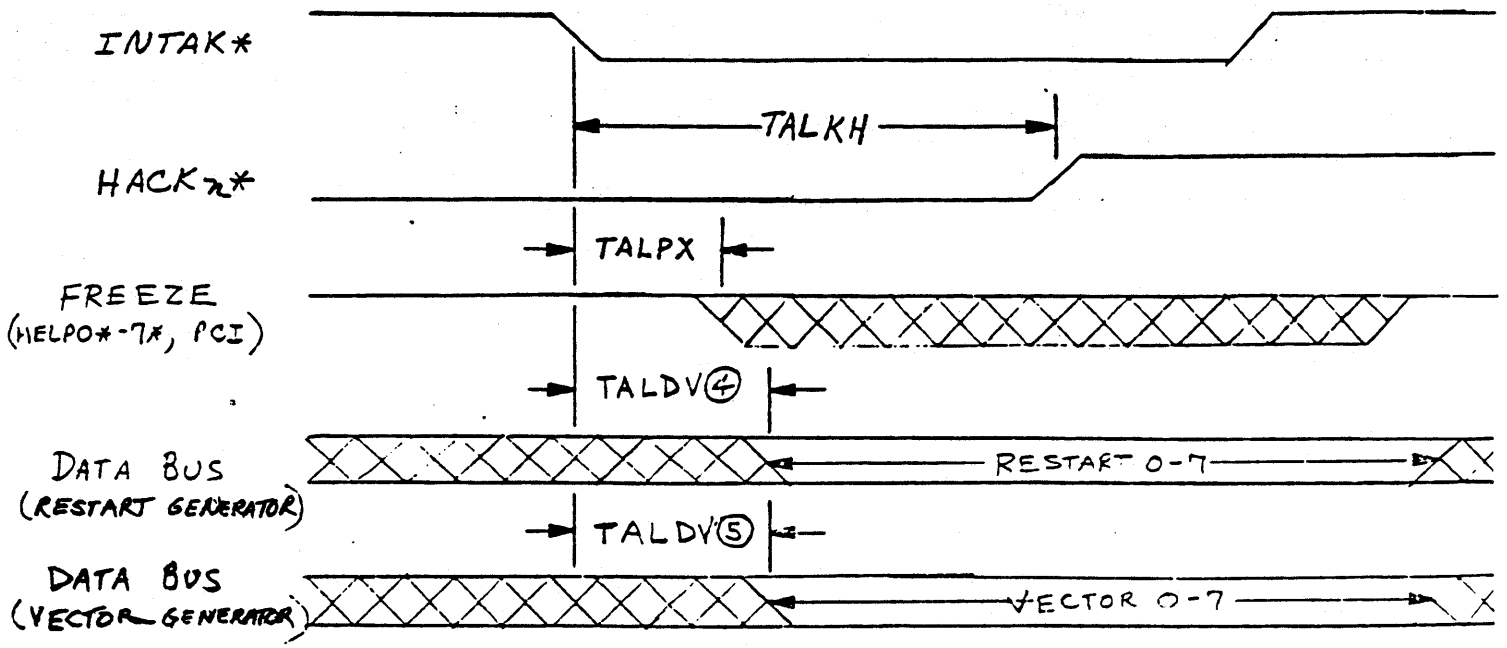
FIGURE 19

C. 7320 TIMING WAVEFORMS

INTERRUPT REQUEST



INTERRUPT RESPONSE



PRIORITY CHAIN PROPAGATION

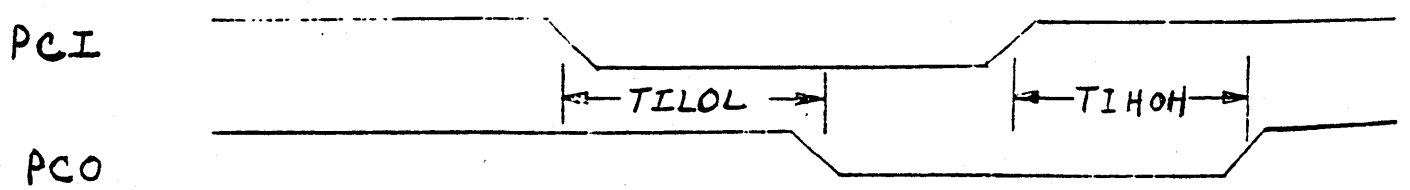


FIGURE 20

D. MECHANICAL

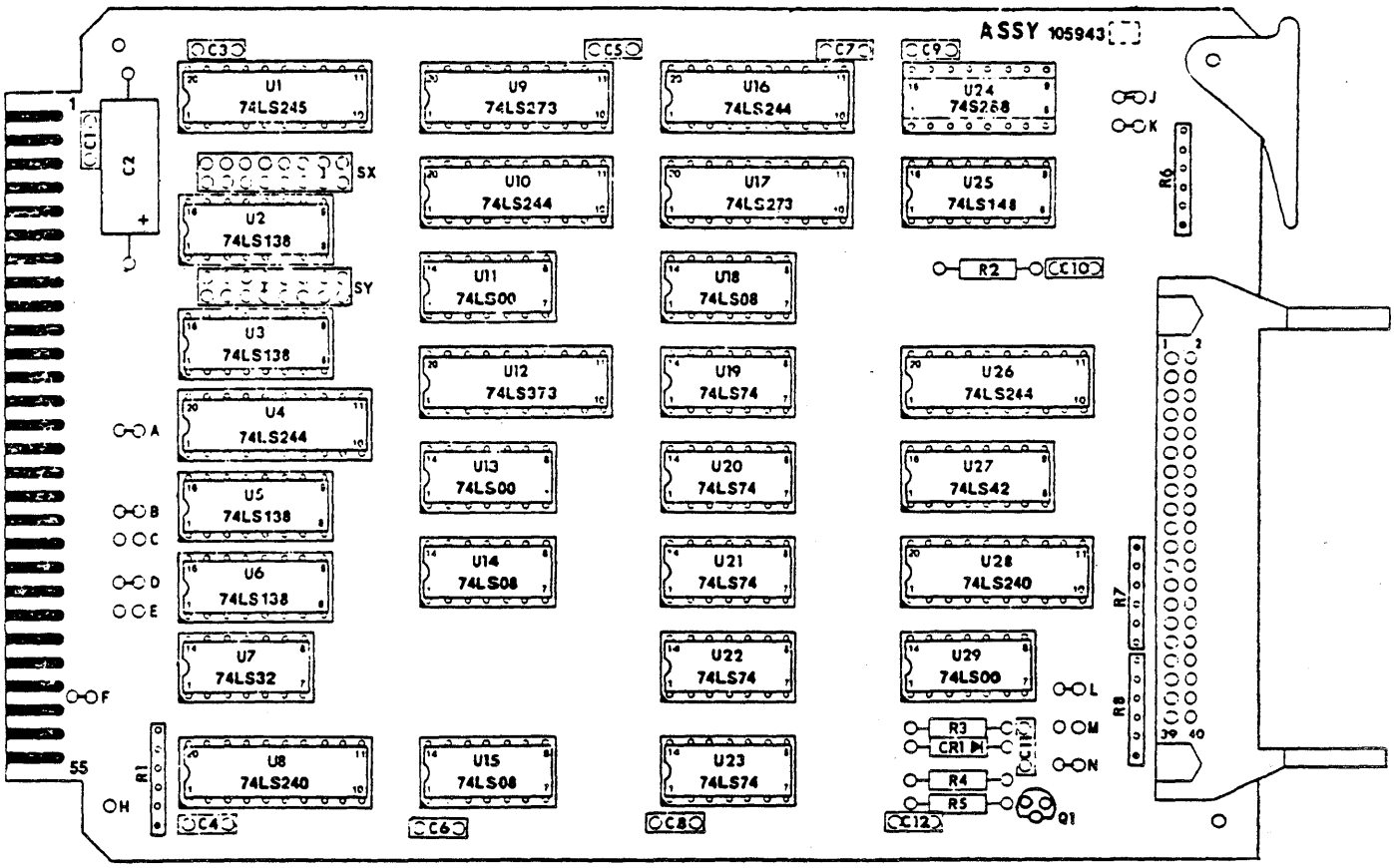


FIGURE 21

The 7320 meets all STD BUS mechanical specifications

USER INTERFACE MATING CONNECTOR

40-PIN FEMALE RIBBON SOCKET	MANUFACTURER	
	3M	SAE
Part #	3324-0001	RD6340

FIGURE 22

STD/7320 EDGE CONNECTOR PIN LIST								
PIN NUMBER				PIN NUMBER				
OUTPUT (LSTTL DRIVE)						OUTPUT (LSTTL DRIVE)		
INPUT (LSTTL LOADS)							INPUT (LSTTL LOADS)	
MNEMONIC								MNEMONIC
+5 VOLTS	VCC		2	1		VCC		+5 VOLTS
GROUND	GND		4	3		GND		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		IORQ*
MEMEX*			36	35		1		IOEXP*
MCSYNC*			38	37				REFRESH*
STATUS 0*			40	39				STATUS 1*
BUSRQ*			42	41				BUSAK*
INTRQ*		20#	44	43		1		INTAK*
NMIRQ*		OUT	46	45				WAITRQ*
PBRESET*			48	47		1		SYSRESET*
CNTRL*			50	49				CLOCK*
PC1	5		52	51	55			PC0
AUX GND			54	53				AUX GND
AUX-V (-12V)			56	55				AUX +V

***Designates Active Low Level Logic**

\# Designates open collector driver

FIGURE 23

7320 USER INTERFACE CONNECTOR PIN LIST							
PIN NUMBER				PIN NUMBER			
OUTPUT (LSTTL DRIVE)					OUTPUT (LSTTL DRIVE)		
INPUT (LSTTL LOADS)						INPUT (LSTTL LOADS)	
MNEMONIC							MNEMONIC
INH*	5		2	1	OUT		GROUND
PCO		55	4	3	OUT		GROUND
NMIRQ*	IN		6	5	OUT		GROUND
HACK7*		55	8	7	OUT		GROUND
HACK6*		55	10	9	OUT		GROUND
HACK5*		55	12	11	OUT		GROUND
HACK4*		55	14	13	OUT		GROUND
HACK3*		55	16	15	OUT		GROUND
HACK2*		55	18	17	OUT		GROUND
HACK1*		55	20	19	OUT		GROUND
HACK0*		55	22	21	OUT		GROUND
HELP7*	15		24	23	OUT		GROUND
HELP6*	15		26	25	OUT		GROUND
HELP5*	15		28	27	OUT		GROUND
HELP4*	15		30	29	OUT		GROUND
HELP3*	15		32	31	OUT		GROUND
HELP2*	15		34	33	OUT		GROUND
HELP1*	15		36	35	OUT		GROUND
HELP0*	15		38	37	OUT		GROUND
INT*		20#	40	39	OUT		GROUND

* Designates active low level logic
\# Designates open collector driver

FIGURE 24

SECTION 6

7320 OPERATING FIRMWARE

The following routines are provided for the 7320 user. They will operate in either Z80 or 8085 systems as outlined below. The flow diagrams provided allow conversion to another processor's assembly language, although modifications may be necessary according to the interrupt characteristics of differing processor types.

These routines may be used without licensing from Pro Log, or they may be used as models for modification. Although they have been tested and are believed to be correct, they are not represented to be appropriate to any specific application or free from errors or patent infringement.

Because of the large number of possible mapping and application variations, the coding forms are tagged with notes as follows:

NOTE	COMMENT
△1	Memory addresses are compatible with Pro Log's 7801 (8085A) and 7803 (Z80) processor cards. Change for systems with different memory mapping.
△2	Port addresses are compatible with 7320 as shipped. Change if the 7320 is remapped.
△3	User-supplied 16-bit memory address appropriate to the application.
△4	Mask's bit pattern = hex FF enables all eight interrupt request latches. Remove bits for permanent latch disable.
△5	If interrupt occurs here, program goes to address 0038, then eventually returns to the instruction following this one with the processor's interrupt system disabled (8085A and Z80).

Figure 25 Firmware Notes

For example, changing the 7320's address mapping allocation will require a change in all program lines with the △2 notation.

A. 7320 Operating Subroutines

These subroutines provide basic system functions by manipulating the I/O ports on the 7320.

In some applications it may be advantageous to combine two or more functions in one subroutine, or modify the subroutine's operation.

In systems with more than one 7320, the cards must be mapped in different I/O port address ranges and separate sets of subroutines are needed to manipulate each card.

Instructions to implement the subroutines listed in Figure 26 can be found in Figure 27. Use the notes as indicated in Figure 25 to modify the program if the memory or port addresses are remapped.

NAME AND DESCRIPTION	REGISTERS CHANGED	START ADDRESS
(INITIALIZE 7320) Removes reset bits and enables the onboard interrupt latches	A	1200
(ENABLE ALL) Enables all onboard interrupts	A	1204
(MASK) Enter with accumulator bits = 1 to disable corresponding latches	A	1209
(RESTORE CHAIN) Restores the 7320's priority chain (PCO output) at the conclusion of an interrupt.	A,F	120D
(DISABLE 7320) Inhibit 7320 & lower order cards Restore with (RESTORE CHAIN)	A	1211
(DISABLE CHAIN) Inhibits lower order cards Restore with (RESTORE CHAIN)	A	1216
(POLL LATCHES) Reads 7320 latch states to processor's accumulator.	A	121B

Figure 26 - Subroutines

HEXADECIMAL PAGE ADR	HEXADECIMAL LINE ADR	HEXADECIMAL INSTR.	MNEMONIC LABEL	MNEMONIC INSTR.	MNEMONIC MODIFIER	TITLE / COMMENTS
12	00	3E	(INITIALIZE 7320)	LDAI		REMOVE ALL LATCH RESET SIGNALS
4	1	FF			FF	
	2	D3		OPA		
2	3	CD			7320 RESET PORT	
	04	3E	(ENABLE ALL)	LDAI		ENABLE ALL LATCHES TO DRIVE THE
4	5	FF			FF	7320's PRIORITY ENCODER
	6	D3		OPA		
2	7	CC			7320 MASK PORT	
5	8	C9		RTS		
	09	2F	(MASK)	CMAL		COMPLEMENT MASK BITS & WRITE TO MASK PORT
	A	D3		OPA		
2	B	CC			7320 MASK PORT	
5	C	C9		RTS		
	0D	AF	(RESTORE CHAIN)	CLAC		SET PCI = PCO = ACTIVE ON 7320 & TO NEXT
	E	D3		OPA		LOWER CARD IN CHAIN
2	F	CE			7320 CHAIN PORT	
5	10	C9		RTS		
	1	3E	(DISABLE 7320)	LDAI		MAKE 7320's PCI = INACTIVE
	2	01			01	
	3	D3		OPA		
2	4	CE			7320 CHAIN PORT	
	5	C9		RTS		
	16	3E	(DISABLE CHAIN)	LDAI		MAKE 7320's PCO = INACTIVE
	7	02			02	
	8	D3		OPA		
2	9	CE			7320 CHAIN PORT	
	A	C9		RTS		
	1B	DB	(POLL LATCHES)	IPA		READ 7320's POLL PORT
2	C	CC			7320 POLL PORT	
	D	C9		RTS		
	E					
	F					

FIGURE 27

B. Program Example: Polled Interrupt Operating System with Rotating Priority

The following example is not a subroutine, but a complete operating system which illustrates:

a. A polled interrupt system not requiring INTAK* (interrupt acknowledge) or vectored interrupt hardware.

b. An interrupt-driven system which is halted unless servicing an interrupt.

c. Programmed priority rotation (each input becomes the highest priority input, in turn).

This program can be used without modification by combining a 7801 (8085A) or 7803 (Z80) processor card with the 7320 Priority Interrupt Card, using the as-shipped configurations of the cards. The user need only supply eight interrupt service routines for up to eight interrupting devices (an example of a service routine is at address 0030 in the program) and any additional hardware or card modules needed to operate the eight interrupting devices. Place the start addresses of the eight routines in the vector lookup table beginning at program address 0020.

The flow diagram of the system, figure 28 , can be used as a model to implement a similar system in any STD BUS processor system or to modify this routine to suit a special application.

Use the notes as indicated in figure 25 to modify the program if the memory or port addresses are remapped.

Note that the flexibility of polled interrupt systems allow unlimited variations within the interrupt response time constraints of the system, since unlimited program manipulation of the interrupt system is possible with the 7320.

Typical Program requirements (times based on 2.5 MHz Z80 system)

a. Response time to an interrupt signal from the halt condition varies approximately from 60 microseconds to 240 microseconds, depending on the current state of the enable mask on which HELP-request latch is active.

b. Uses registers A, D, E, F, H, L; reserves memory page 23 for the stack and location 2300 as the interrupt rotate mask.

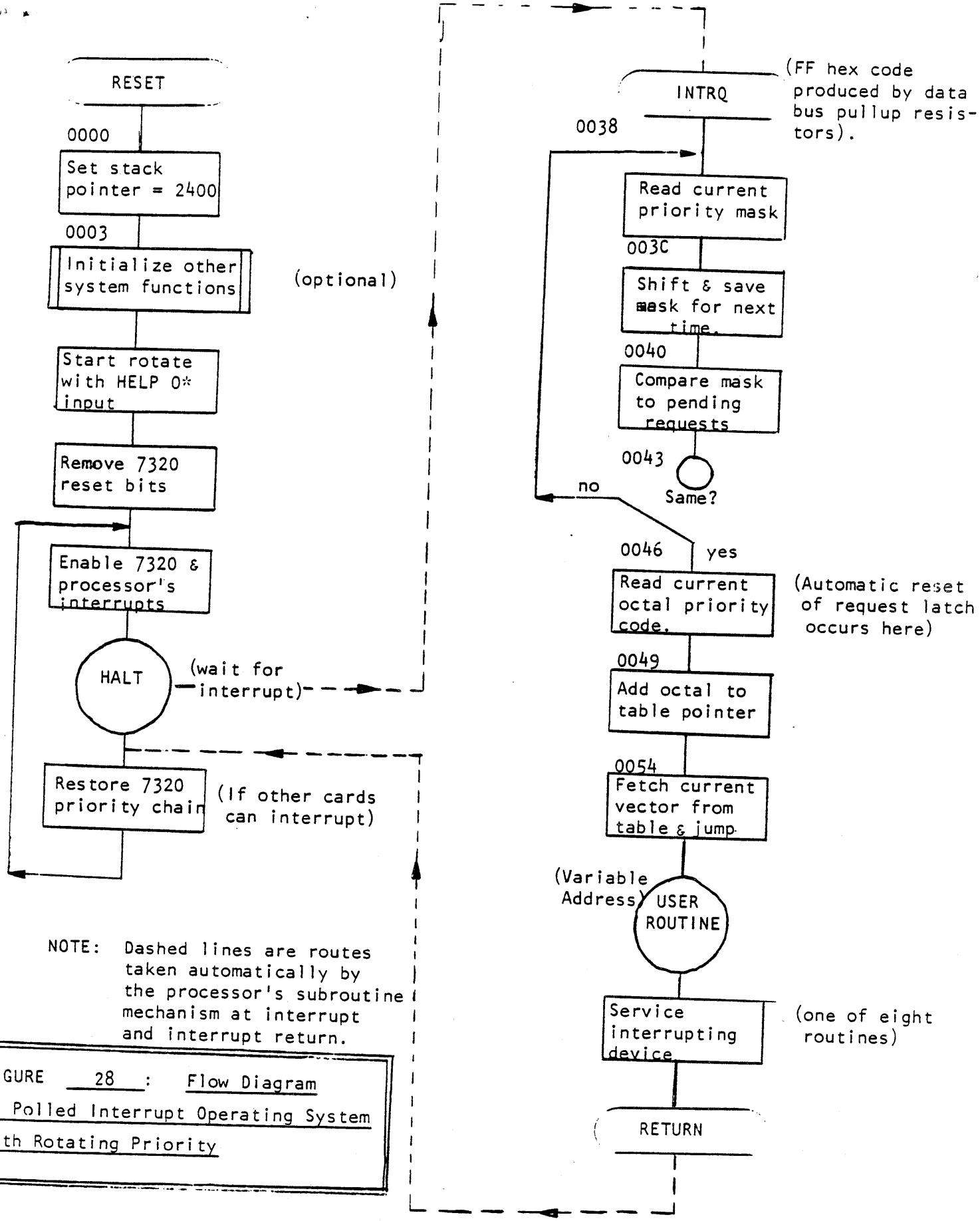
c. The user-supplied interrupt service routines are treated as subroutines and are terminated with RTS (return-from-subroutine, hex C9), unless used in a Z80 system with Z80 peripheral chips such as the PIO, in which case the routines are terminated with RTI (return-from-interrupt, hex ED,4D).

RTI executes as RTS in the Z80, but the peripheral chips decode the RTI opcode sequence and use it to restore the priority chain (PCO) signal.

Use RTS in all 8085A systems.

d. The system hardware is configured as follows:

1. Leave vector generator socket U24 empty.
2. Leave RESTART generator U16 (74LS244) installed.
3. Make RESTART generator conversion to Input Port CD as detailed in Appendix C. (Remove jumper B install jumper C)
4. Make sure processor card has data bus pullup resistors so that processor sees the hex FF opcode during interrupt acknowledge. Alternately (Z80 only), insert the SEIM 1 (set interrupt mode 1 instruction, hex ED, 56) in front of the ENI instruction at line 0013 in the program. (7801 and 7803 cards have pullup resistors)



NOTE: Dashed lines are routes taken automatically by the processor's subroutine mechanism at interrupt and interrupt return.

FIGURE 28 : Flow Diagram of Polled Interrupt Operating System with Rotating Priority

PRO-LOG CORPORATION

PROGRAM ASSEMBLY FORM

TITLE

DATE

HEXADECIMAL PAGE ADR	HEXADECIMAL LINE ADR	HEXADECIMAL INSTR.	MNEMONIC LABEL	MNEMONIC INSTR.	MNEMONIC MODIFIER	COMMENTS
00	00	31	RESET	LDPI	SP	SET STACK POINTER AT TOP OF RAM SUPPLIED
	1	00			RAM 2400	WITH 7801, 7803 PROCESSOR CARDS
1	2	24			–	
	3	CD		JS		PRESET OTHER SYSTEM FUNCTIONAL MODULES
	4				(INITIALIZE)	(E.G. TIMER, I/O LINES) IF APPLICABLE
3	5				–	
	6	3E		LDAI		START WITH HELP0* INTERRUPT REQUEST
	7	01			01	
	8	32		STAD		
	9	00			ROTATE MASK	
1	A	23			–	
	B	3E		LDAI		REMOVE RESET SIGNALS FROM HELP-REQUEST
4	C	FF			FF	LATCHES
	D	D3		OPA		
2	E	CD			7320 RESET PORT	
	0F	3E	INTERRUPT LOOP	LDAI		ENABLE HELP-REQUEST LATCHES TO DRIVE
4	10	FF			FF	7320's PRIORITY ENCODER
	1	D3		OPA		
2	2	CC			7320 MASK PORT	
	3	00		NOP		INSERT (ED SEIM 1 IF NECESSARY PER SECTION 5.4.1(4)
5	4	00		NOP		(56 ↓
	5	3E		LDAI		RESTORE THE STD BUS PRIORITY CHAIN
	6	00			00	(OMIT IF NOT APPLICABLE)
	7	D3		OPA		
2	8	CE			7320 CHAIN PORT	
	9	C3		JP		RESTORE 7320 MASK BITS AND REPEAT
	A	0F			INTERRUPT LOOP	
1	B	00			–	
	C					
	D					
	E					
	F					

FIGURE 20A

PRO-LOG CORPORATION

PROGRAM ASSEMBLY FORM

HEXADECIMAL PAGE ADR	HEXADECIMAL LINE ADR	HEXADECIMAL INSTR.	MNEMONIC LABEL	MNEMONIC INSTR.	MNEMONIC MODIFIER	TITLE / DATE / COMMENTS
00	20		VECTOR TABLE		HELP0 VECTOR	ADDRESS TABLE OF SERVICE ROUTINES OF
3	1				–	UP TO EIGHT INTERRUPTING DEVICES
	2				HELP1 VECTOR	
3	3				–	
	4				HELP2 VECTOR	
3	5				–	
	6				HELP3 VECTOR	
3	7				–	
	8				HELP4 VECTOR	
3	9				–	
	A				HELP5 VECTOR	
3	B				–	
	C				HELP6 VECTIR	
3	D				–	
	E	30			HELP7- VECTOR	} VECTORS TO 0030 AS EXAMPLE.
3	F	00			–	
00	30	xx	HELP 7	xx		INSTRUCTIONS AS REQUIRED TO SERVICE
	1	xx		xx		DEVICE #7. NOTE: TERMINATE ROUTINE
	2	xx		xx		WITH RTI (HEX ED,4D) ONLY IF Z80
	3	C9		RTS		PERIPHERAL CHIPS ARE USED IN SYSTEM'S
	4	.				PRIORITY CHAIN ALONG WITH 7320.
	5					
	6					
	7					
00	38	3A	INTRQ	LDAD		FETCH NEXT HELP-REQUEST TO BE SAMPLED
	9	00			ROTATE MASK	
1	A	23			–	
	B	47		LDB	A	
	C	07		RLA		MOVE TO NEXT HIGHER HELP INPUT FOR
	D	32		STAD		NEXT TIME; REPLACE MASK
	E	00			ROTATE MASK	
	F	23			–	

FIGURE 29B

PRO-LOG CORPORATION

PROGRAM ASSEMBLY FORM

HEXADECIMAL PAGE ADR	HEXADECIMAL LINE ADR	HEXADECIMAL INSTR.	MNEMONIC LABEL	MNEMONIC INSTR.	MNEMONIC MODIFIER	COMMENTS
						TITLE DATE
00	40	DB		IPA		POLL TO SEE WHICH REQUESTS ARE PENDING
(2)	1	CC			7320 POLL PORT	↓
	2	A0		ANA	B	IF CURRENT HELP REQUEST IS NOT ACTIVE, MOVE
	3	CA		JP	Z1	↓ MASK ONE POSITION AND TRY AGAIN.
	4	38			INTR0	
(1)	5	00			—	↓
	6	78		LDA	B	FOUND AN ACTIVE HELP-REQUEST; DISABLE ALL
	7	D3		OPA		↓ OTHER REQUESTS BUT THAT ONE
(2)	8	CC			7320 MASK PORT	↓
	9	DB		IPA		READ OCTAL VALUE OF CURRENT REQUEST, AND
(2)	A	CD			7320 RESTART PORT	↓ AUTOMATICALLY RESET THE LATCH.
	B	E6		ANAI		REMOVE "1" BITS FROM RESTART OPCODE, LEAVING
	C	38			38	↓ THE OCTAL CODE.
	D	OF		RRA		POSITION OCTAL CODE & MODIFY LOOKUP TABLE
	E	OF		RRA		↓ ADDRESS POINTER BY ADDING THE
	F	C6		ADAI		OCTAL VALUE TO THE TABLE START
00	50	20	(1)		TABLE LINE	ADDRESS
	1	6F		LDL	A	
	2	26		LDH	I	
(1)	3	00			TABLE PAGE	↓
	4	5E		LDE	M	READ THE CORRECT SERVICE ADDRESS
	5	23		ICP	HL	↓ FROM THE LOOKUP TABLE
	6	56		LDD	M	↓
	7	EB		XCP	DE, HL	INDIRECT JUMP TO THE APPROPRIATE
	8	E9		JPN	HL	↓ INTERRUPT SERVICE ROUTINE.
	9					
	A					
	B					
	C					
	D					
	E					
	F					

FIGURE 29C.

APPENDIX A

7320 VECTOR PROM PROGRAMMING

The 7320's Vector Generator PROM is a 74S288 or any 32 x 8 device with a compatible pinout and address access time or chip select access time able to satisfy the following equation:

$$T \leq t_{cpu} - 100ns$$

where T is the worst case of address or chip select access time, and t_{cpu} is the maximum access time required for memory devices during INTAK* (or equivalent) for the processor card used.

The following table lists devices which may be used and the Personality Module required for use with Pro-Log's PROM programmers to program each device type.

PROM	MANUFACTURER	PERSONALITY MODULE	COMMENT
74S288	T.I.	PM9046 , PA16-4, 32x8(L)	Generic; see note
74S288	National	PM9047, PA16-2 or -4, 32x8(L)	Generic; see note
N82S123	Signetics	PM9059, PA16-2 or -4, 32x8(L)	Generic; see note
5610	Intersil	PM9016	Single PROM Module
MB7051	Fujitsu	PM9016	
7603-5	Harris	PM9039 , PA16-2 or -4, 32x8(H)	Generic; see note
6331-1	MMI	PM9037, PA16-2, 32x8(H)	Generic; see note
63S081 63LS081	MMI	PM9066, PA16-2, 32x8(L)	Generic; see note

Figure 30

Note: Generic Personality Modules require a Pinout Adapter (PA16-2 or -4) and Configurator (32 x 8 (X)) which are listed after the PM type in the table. Generic modules may be used for other PROMs within a family as well as the type shown.

The above list is not intended to be complete and does not constitute a recommendation of any particular device type.

If the user lacks the ability to program these PROMs, the manufacturer's distributors for these devices frequently offer the capability of supplying devices that are preprogrammed with code submitted by the user.

Vector PROM Organization

Only eight of the 32 locations in the Vector PROM may be used by a single 7320 Card. Consequently, up to four complete vector tables may be stored in each PROM. This can simplify the programming process in user systems which are manufactured using up to four 7320 cards. Jumper pads K and L are provided on the card (see assembly diagram) to allow the user to choose which of the four vector tables is selected on a specific 7320 card as follows:

JUMPERS	SELECT PROM ADDRESSES
K connected, J connected	00-07 hexadecimal
K connected, J open	08-0F hexadecimal
K open, J connected	10-17 hexadecimal
K open, J open	18-1F hexadecimal

Figure 31

HEXADECIMAL			MNEMONIC			TITLE 7320 VECTOR PROM DATE
PAGE ADR	LINE ADR	INSTR	LABEL	INSTR.	MODIFIER	COMMENTS
	0 0					JUMPERS J CONNECTED, K CONNECTED
	0 1					
	0 2					
	0 3					
	0 4					
	0 5					
	0 6					
	0 7					
	0 8					JUMPERS J CONNECTED, K OPEN
	0 9					
	0 A					
	0 B					
	0 C					
	0 D					
	0 E					
	0 F					
	1 0					JUMPERS J OPEN, K CONNECTED
	1 1					
	1 2					
	1 3					
	1 4					
	1 5					
	1 6					
	1 7					
	1 8					JUMPERS J OPEN, K OPEN
	1 9					
	1 A					
	1 B					
	1 C					
	1 D					
	1 E					
	1 F					

FIGURE 32: 7320 VECTOR GENERATOR PROM PROGRAMMING FORM

100001 2/77

APPENDIX B

7320 SIGNAL GLOSSARY

STD BUS Edge Connector Signals (also see STD BUS Pin List, Figure 23)

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.
SYS-RESET*	System Reset	47	Low Active	Reset signal originating at processor card in response to power-on or PBRESET*.
IOEXP*	I/O port Expansion	35	Low Active	Bank select signal originating at processor card or other control card. Must be low to activate 7320.
IORQ*	I/O Request	33	Low Active	Indicates that the Address Bus holds a valid I/O port address.
RD*	Read	32	Low Active	Indicates that the processor wants to read data from the addressed input port.
WR*	Write	31	Low Active	Indicates that the processor wants to write data to the addressed output port.
INTAK*	Interrupt Acknow-ledge	43	Low Active	The processor is in the act of responding to an interrupt; causes the 7320 to place a vector code or RESTART instruction on the Data Bus.
INTRQ*	Interrupt Request	44	Low Active	Maskable interrupt input to the processor card.
NMIRQ*	Non-maskable Interrupt Request	46	Low Active	Nonmaskable, highest-priority interrupt input to the processor card.
PCI	Priority Chain In	52	High Active	Card-level priority signal used as a card enable input to the 7320 from cards of higher priority.
PCO	Priority Chain Out	51	High Active	Card-level priority signal output from the 7320 used to disable cards of lower priority
MCSYNC* STATUS 1* STATUS 0*	Auxilary processor-dependent timing signals available for use on the 7320 in special timing situations. Consult the processor card data sheet of interest for specific timing information.			

Figure 33

. User Interface Connector (J1) Signals (also see User Interface Connector Pin List, Figure 24)

MNEMONIC	MEANING	PIN(s)	DESCRIPTION	FUNCTION
HELP0*-HELP7*	Interrupt Request	24,26, 28,30, 32,34, 36,38	Edge Sensitive (1→0 only)	Eight interrupt request inputs from devices in user's system which can interrupt the processor.
HACK0*-HACK7*	HELP Acknowledge	8,10, 12,14, 16,18, 20,22	Low Active	Acknowledges that the corresponding request latch has been set by the HELPn* signal; optionally used for LED indicator or handshake with interrupting device in user's system. Not normally required.
INT*	Interrupt	40	Low Active	Open collector output from 7320; allows 7320 to prioritize interrupt requests for applications different from normal INTRQ* drive on STD BUS.
NMIRQ*	Non-Maskable Interrupt Request	6	Low Active	See NMIRQ* Figure 33; provides for external drive of NMIRQ* line via J1.
INH*	Inhibit	2	Low to inhibit, High to enable.	Same function as PCI Figure 33, may be used as a second inhibit in addition to PCI, and to pulse INTRQ* for edge sensitive processor interrupt inputs.
PCO	Priority Chain Out	4	Active High	Identical to PCO, Figure 33 May be used in combination with INH* to provide a card-slot-independent priority chain in place of PCI/PCO at the backplane.

Figure 34

c. Figure 35 - Internal 7320 Signals (See schematic 105942)

⚠1 Note: Unless inhibited by the program

MNEMONIC	MEANING	DESCRIPTION	ORIGIN COORDINATES	COMMENTS
AR0* to AR7*	Automatic Reset	Active low	A2	Automatic latch reset signal developed from IRSP* and LP1,2,4 decoded.
BACK*	Buffered INTAK*	Active low	B6	
BD0 to BD7	Buffered Data Bus	Active high	D7	Internal 3-state bidirectional bus.
BRD*	Buffered RD*	Active low	B7	
CARD*	Card Select	Active low	B6	Decoded A2-A7 plus IORQ* and IOEXP*.
EO*	Enable Out	Active low	C3	Active when PC1 = 1⚠1 and 7320 has no active interrupt requests (PC0 = EO*)⚠.
EPE*	Enable Priority Encoder	Active low	A6	Active when PC1 = 1⚠
EN0 to EN7	Enable Mask bit	Active high	D7,8	Programmed interrupt mask enable bits.
FEPE*	Frozen EPE*	Active low	A2,3	Enable Priority Encoder frozen by INTAK
FIR0* to FIR7*	Frozen Interrupt Request	Active low	A3	Enabled interrupt requests frozen by INTAK*.
IL0 to IL7	Interrupt Latches	Active high	ABCD4	Interrupt latch states to Poll Port.
IP0* IP1*	Input Port	Active low	B7	Input port strobes.
IR0* to IR7*	Masked Interrupt Request	Active low	ABCD4	Enabled interrupt requests to freeze latch.
IRSP*	Interrupt Response	Active low	D3	Activated by INTAK* only when INTRQ* generated by this 7320.
P1 P2 P4	Priority Code	Active high	C3	3-bit code representing top priority active enabled interrupt on this 7320.
OP0* OP1* OP2*	Output Port	Active low	B7	Output port strobes.
PC1*	Buffered PC1	Active low	A6	
PR0* to PR7*	Program Reset	Active low	D6	Programmable resets to the interrupt request latches.
RST*	Reset	Active low	B8	Buffered SYSRESET*. GENERAL INTERNAL RESET

APPENDIX C

INTRODUCTION

The 7320 is shipped in a configuration which allows it to operate without alteration in the majority of applications involving the 7801 (8085A), 7802 (6800), and 7803 (Z80) Processor cards. However, a number of options are provided which can enhance the card's performance in certain applications. These options are selected by the removal and replacement of jumper wires or integrated circuits as shown in this Appendix. The options include:

a. 7320 onboard I/O port address selection
b. Priority Chain (card level) control
c. Interrupt response generators (vector or instruction)
d. Automatic interrupt request latch resetting
e. Interrupt request signal input and acknowledge output polarity
f. Special processor control
g. Interrupt request signal output control
h. Vector PROM addressing

In order to make these changes properly the user should have the 7320 schematic (Pro-Log document number 105942) and assembly diagram (105943) for reference.

PRIORITY CHAIN CONTROL

The 7320's priority chain circuit allows card-level control of interrupt priorities where more than one card in the system is capable of generating the INTRQ* signal.

The following jumpers control the 7320's priority chain:

JUMPERS	FUNCTION	OPTIONAL FUNCTION
F	Connects PCI to the 7320's priority encoder.	Remove jumper C if: a. Only this 7320 is capable of generating an interrupt request. b. This 7320 has the highest priority in a chain of other cards. c. The PCI/PCO scheme is to be defeated and a card-slot-independent scheme is to be substituted by the user at the front of the card rack (7320 user-interface connector pin 1, INH*, is provided for that purpose and is equivalent to PCI).
D,E	Jumper D connected, E open, places the 7320 in the chain. If the card generates an interrupt, PCO will go inactive.	Jumper D open, E connected, removes the 7320 from the priority chain but allows PCI to propagate through the card to PCO.

Figure 36

In systems where the INTAK* signal is present, the latch which caused the interrupt presently being serviced will be automatically cleared by the delayed leading edge of INTAK*. The autoreset decoder (74LS42 U27) performs this function which can be defeated by removing jumper A.

In systems where no INTAK* exists and polling is used, the autoreset feature can be retained by turning the vector generator or the RESTART generator into an input port readable by the program. When the program reads this input port, it can find the highest priority pending interrupt in a system with one or more 7320's and automatically clears the corresponding HELP* request latch. The vector generator or RESTART generator is changed to an input port as follows:

JUMPERS	FUNCTION	OPTIONAL FUNCTION
B, C	Jumper B connected, C open, connects the INTAK* signal to the response generators and autoreset decoder	Jumper B open, C connected, replaces the INTAK* signal with the IP1* signal so that the response generators and autoreset decoder respond as Input Port CD. (1)

FIGURE 37

(1) If the RESTART generator is retained on the card, reading Input Port CD (or equivalent user-selected address) will return one of eight binary codes to the system processor corresponding to the eight HELP* inputs as shown in figure 5 . These codes contain the octal encoded priority level in data bits 5,4,3. In addition, the corresponding HELP* request latch will be cleared by the operation.
If the vector generator is retained on the card, it will respond as above except that the data returned will be the corresponding data programmed into the vector PROM by the user. This may be a numeric representation of the HELP* request input as 00-07, or other data, or part of a jump address to be formulated by the program.

HELP* and HACK* POLARITY

The eight HELP* inputs and eight HACK* outputs are each buffered by octal buffer devices, 74LS240 U28 and 74LS244 U26 respectively. These devices result in low level active inputs and outputs at the user interface connector for these signals.

By interchanging the two devices U28 and U26, both the HELP* and HACK* signal groups will become high level active signals.

SPECIAL PROCESSOR CONTROL

The combination of interrupt requests and state of the PCI signal into the 7320's priority encoder are frozen by the INTAK* signal (or IP1*, see AUTOMATIC INTERRUPT REQUEST LATCH RESETTING above). This prevents a higher order incoming interrupt request from changing the interrupt response signals while the microprocessor is in the act of reading them, thus preventing possible erroneous misreads.

An additional input to this freeze circuit (jumper-pad H) is provided to allow the user to freeze these signals in specific applications. A low level on jumper pad H effects a freeze, causing the 7320 to ignore incoming changes at any of the HELP* inputs or PCI. Note that this does not prevent the HELP* latches from being set, so no interrupt requests are lost.

It is recommended that when used with the Z80-based processor cards other than Pro-Log's 7803, jumper-pad H be connected to the STATUS 1* (M1*) signal at the pad provided next to STD BUS connector pin 39. This freezes the interrupt system during the Z80's M1 time in accordance with the manufacturer's recommendation without losing any interrupt requests.

Additional pads are provided at card edge pins 38 (MCSYNC*), 40 (STATUS 0*), 5 (spare) to be used if required. No connections are required for any of Pro-Log's 7800 Series.

INTERRUPT REQUEST SIGNAL CONTROL

DC stable

The 7320 is designed to generate a single open-collector/interrupt request signal to the INTRQ* line on the STD BUS backplane, and provides an unbuffered path from the user interface connector (pin 5) to the NMIRQ* (nonmaskable interrupt request) line. The following options are available:

Prioritized NMIRQ*

By opening jumpers L and N and connecting jumper M, the 7320's priority encoder drives the NMIRQ* line. Note, however, that in many microprocessors the INTRQ* input is level sensitive while the NMIRQ* input is edge sensitive.

Edge Sensitive Interrupt Inputs

User interface connector pin 2 (INH*) can be jumpered to MCSYNC*, STATUS 0*, STATUS 1*, or another user supplied signal to provide at least one edge transition in each memory cycle to trigger the NMIRQ* or other edge sensitve input and prevent loss of requests.

Card Front Interrupt Inputs

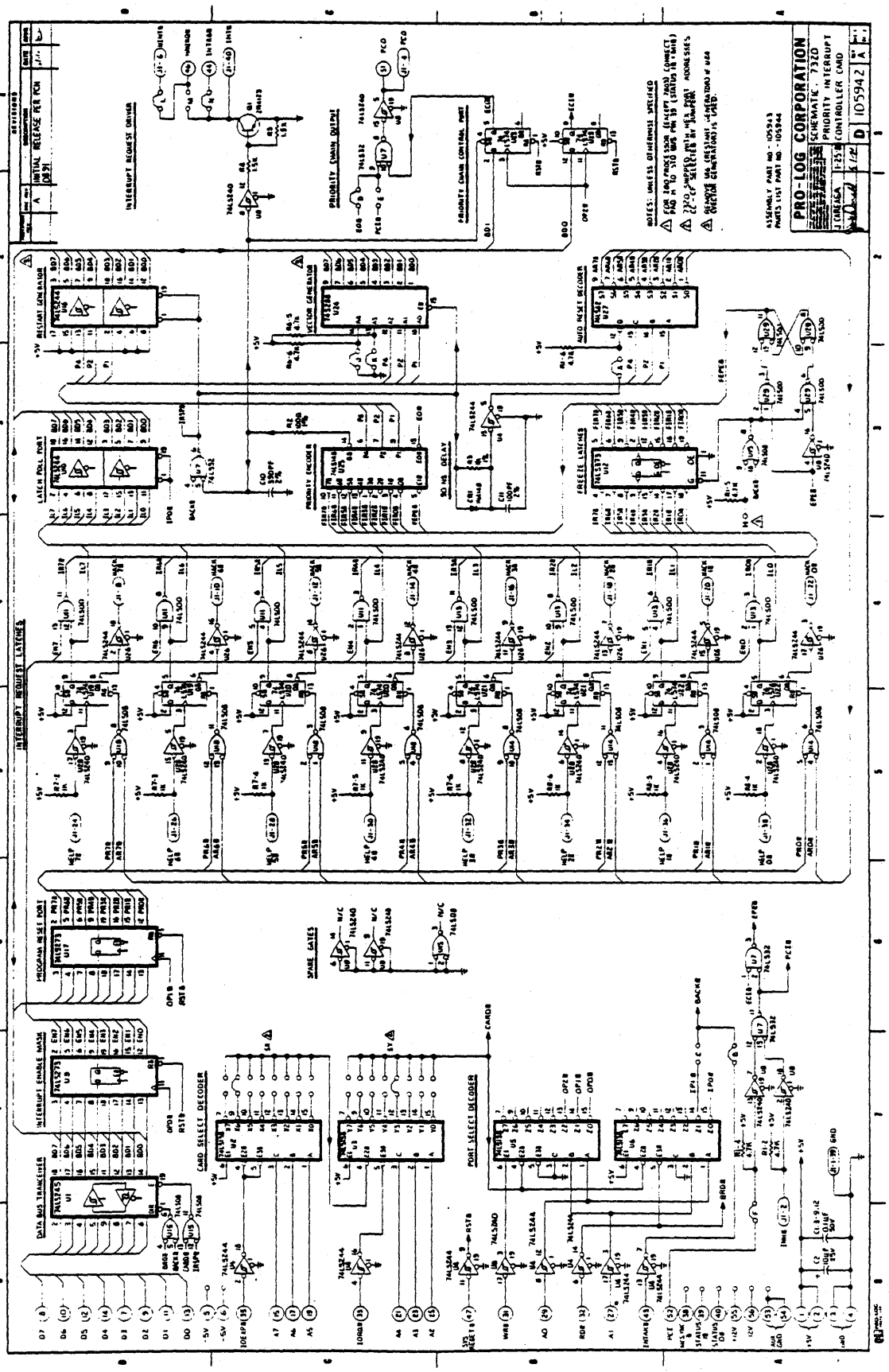
Cards such as Pro-Log's 7801 (8085A-based) processor provide interrupt request inputs (in addition to INTRQ* and NMIRQ*) which are available at the card front. Opening jumper N will prevent the 7320 from driving INTRQ*. J1 pin 40 then supplies the INT* signal. Note that the 7801 7.5 interrupt is edge and level sensitive, while the 6.5 and 5.5 interrupts are level sensitive.

Vector PROM Addressing

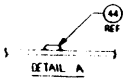
Refer to Figure 31 in Appendix A for use of jumpers J and K to control addressing of the appropriate 8 byte section of the 32 byte user supplied vector PROM if installed.

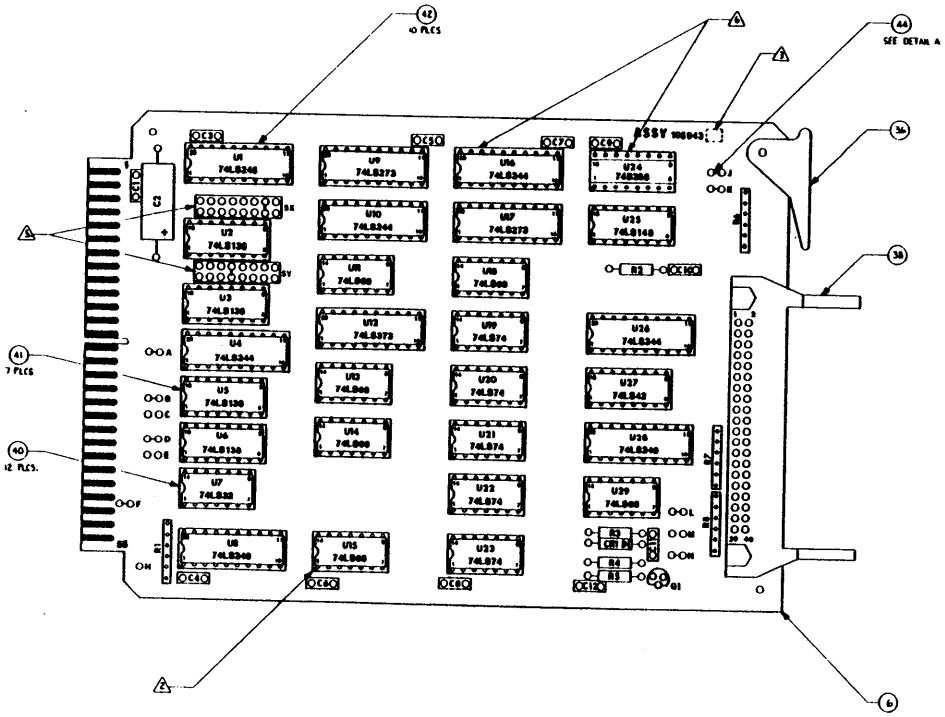
APPENDIX D

REFERENCE ONLY



REVISIONS				
ENG. ORDER NO.	DOC. REV.	DESCRIPTION	DATE	APPR.
	A	INITIAL RELEASE PER PLN 981		
	B	REVISED PER ECN 1051		





NOTES: UNLESS OTHERWISE SPECIFIED

1. REFERENCE DESIGNATIONS ARE FOR LOCATING PURPOSES ONLY AND MAY NOT APPEAR ON ACTUAL PART
2. DENOTES PIN 1 END OF IC'S.
3. IDENTIFY WITH ASSEMBLY REV LETTER USING RUBBER STAMP.
4. BOARD TO CONFORM WITH ASSEMBLY PROCEDURES AS1004
5. 7320 SHIPPED WITH HEX PORT ADRESSES 1C CF SELECTED BY JUMPER
6. REMOVE U16 IF U24 IS USED

ITEM	DESCRIPTION	REF DESIGNATION
21	100Ω .25W 1% MF	R2
20	1M .25W 1% MF	R3
19	1.5K 1/4W 5% CF	RN 5
18	47K 1W 5% NETWORK	RN 6
17	1K 1W 5% NETWORK	RN 7,8
16		
15	2N4123	Q1
14		
13	IN4148	CR1
12		
11	.1UF. 50V	C1-3, 9-12
10	100PF. 2% 500V	C11
9	390PF 2% 100V	C10
8	10UF. 25V	C2
7		
6	105940	PWB

SCHEMATIC NO - 105948
PARTS LIST NO. - 105944

PRO-LOG CORPORATION

ASSEMBLY 7320
PRIORITY INTERRUPT
CONTROLLER CARD

J. CARBAGA 4-1-80

D 105943 B

PRO-LOG
CORPORATION 2411 Garden Road Monterey, California 93940 Telephone (408) 372-4593

106178B 6/81 200
TWX: 910-360-7082