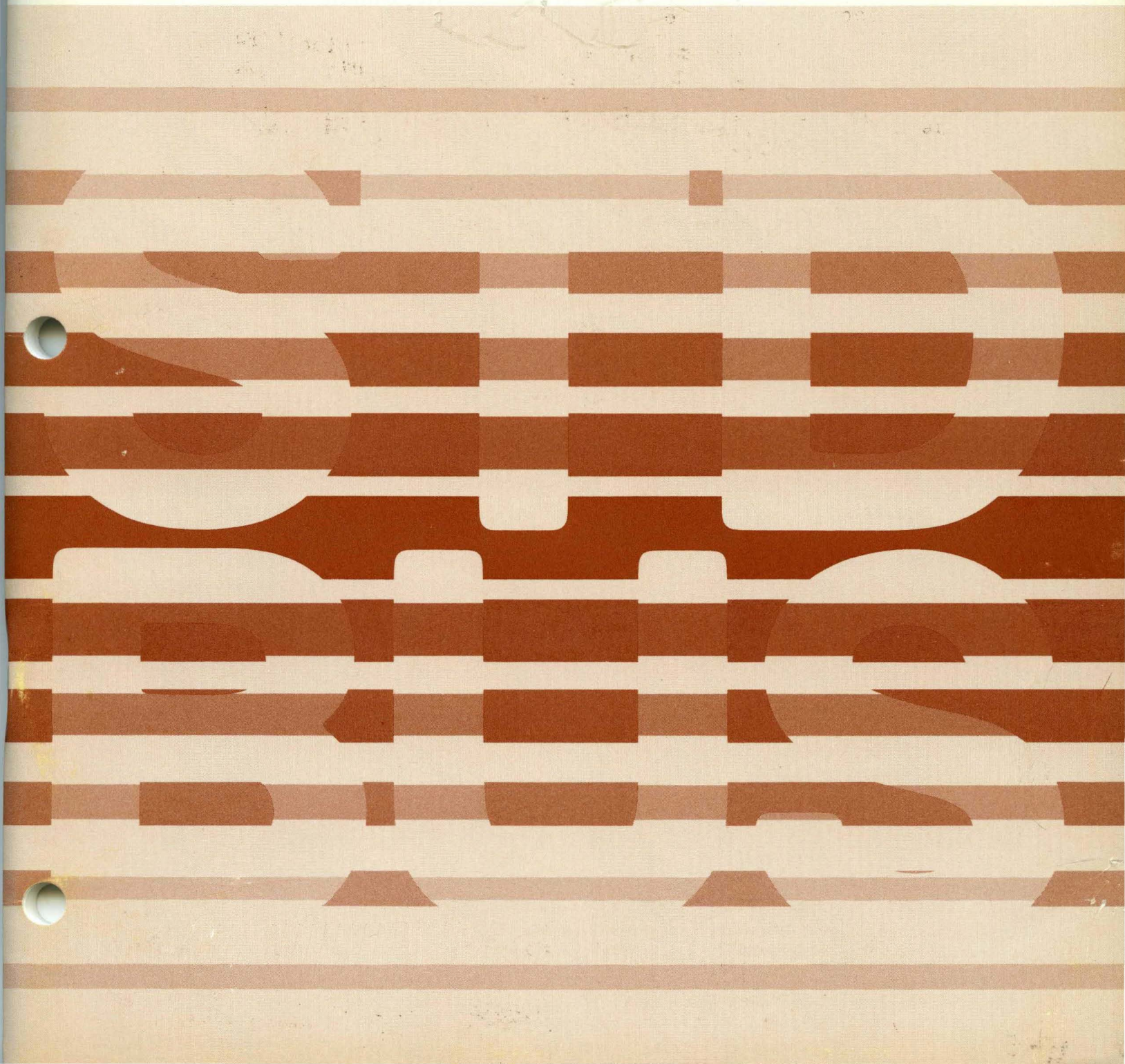
PRO-LOG
CORPORATION

STD 7000

7802
6800 Processor 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.

PRO-LOG
CORPORATION

STD 7000

7802
6800 Processor Card

USER'S MANUAL

2/82

FOREWORD

This manual explains how to use Pro-Log's 7802 6800 Processor Card. It is structured to reflect the answers to basic questions that you, the user, might ask yourself about the 7802. We welcome your suggestions on how we can improve our instructions.

The 7802 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

Figures

SECTION 1
Product Overview

This card provides a buffered and fully expandable 6800 microprocessor with onboard RAM and PROM sockets.

The 7802 includes 1K byte of RAM with sockets for up to 4K, and sockets for up to 8K bytes of ROM or EPROM. An STD BUS system using the 7802 may be expanded to the full memory and I/O capacity of the 6800. The STD BUS interface may be disabled for DMA and multiprocessor applications.

Main Features

- **6800 Processor**
- **4096 bytes RAM capacity onboard**
- **1024 bytes RAM included**
- **8192 bytes ROM capacity onboard**
- **3-state Address, Data, Control Buses**
- **Crystal-controlled 1μs clock**
- **Power-on reset and pushbutton reset input**
- **STD BUS compatible I/O mapping**
- **Standard and custom memory and I/O remapping options**
- **Single +5V operation**
- **Use Pro-Log D1004 1Kx8 memories (two 2114L's)**

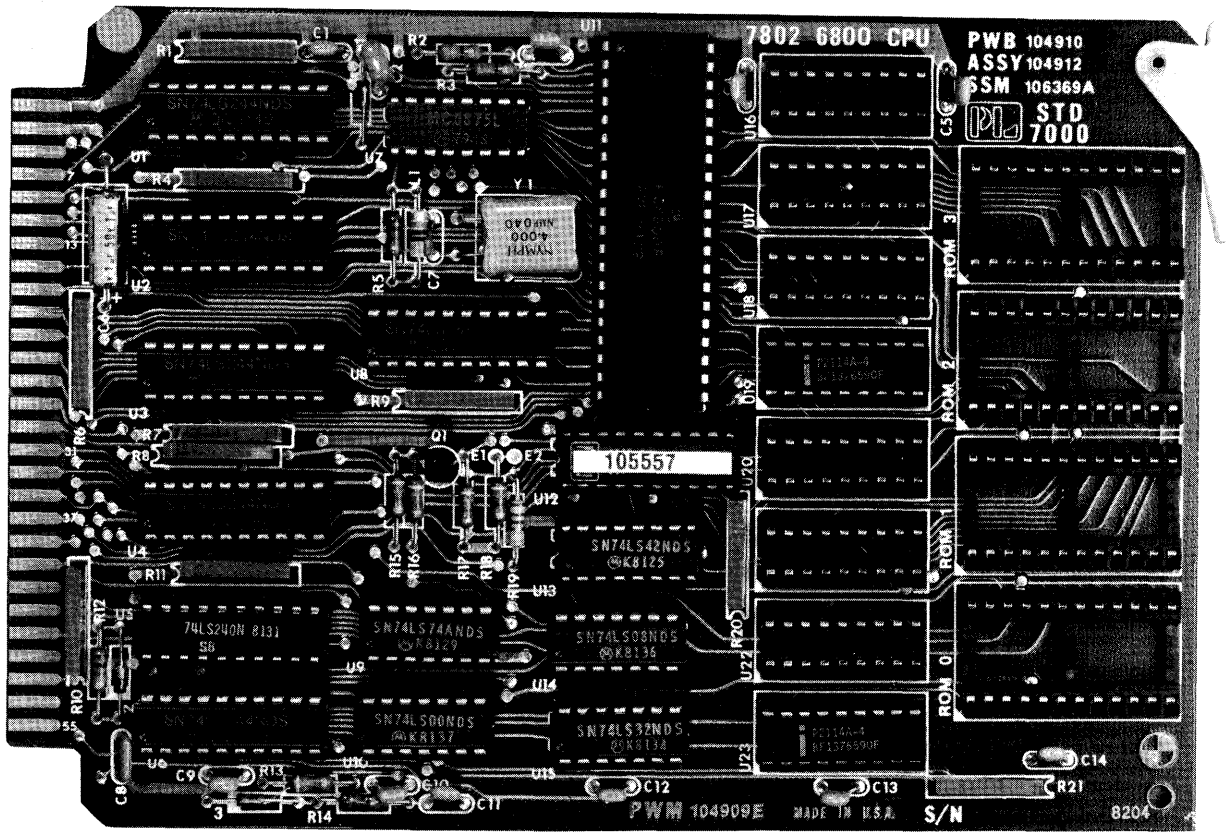


Figure 1-1. 7802 6800 CPU Card.

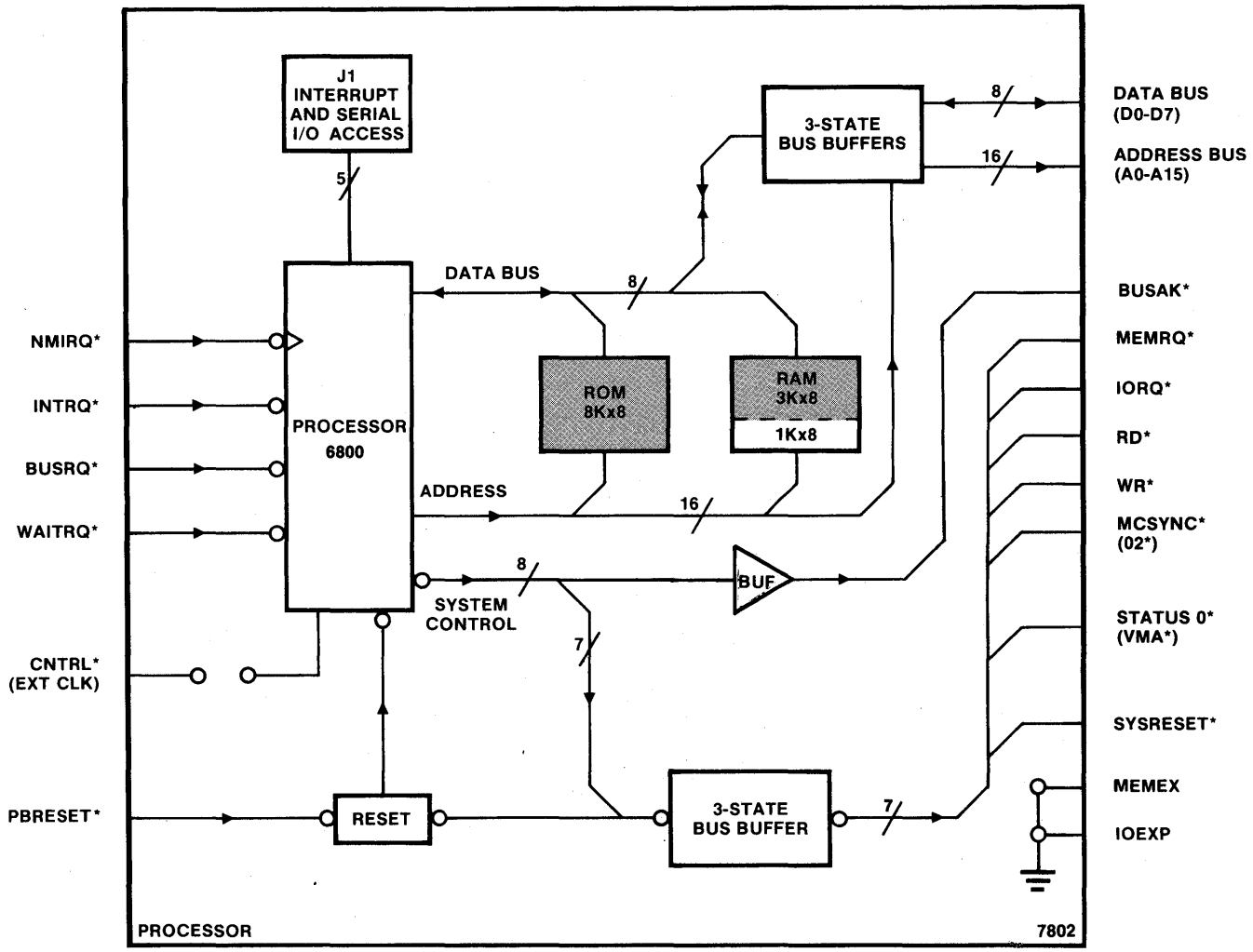


Figure 1-2. Block Diagram of 7802 CPU Card.

SECTION 2
The STD BUS

The STD BUS standardizes the physical and electrical aspects of modular 8-bit microprocessor card systems, providing a dedicated, orderly interconnect scheme. The STD BUS is dedicated to internal communication and power distribution between cards, with all external communication made via I/O connectors which are suitable to the application. The standardized pinout and 56-pin connector lends itself to a bused motherboard that allows any card to work in any slot.

As the sytem processor and primary system control card, the 7802 is responsible for maintaining the signal functionally defined by the STD BUS standard.

A complete copy of the STD BUS standard is contained in the *Series 7000 STD BUS Technical Manual*, available from Pro-Log Corporation, 2411 Garden Road, Monterey, California 93940.

STD BUS Summary

The 56-pin STD BUS is organized into five functional groups of backplane signals:

1. Logic Power Bus — pins 1-6
2. Data Bus — pins 7-14
3. Address Bus — pins 15-30
4. Control Bus — pins 31-52
5. Auxiliary Power — pins 53-56

Figure 2-1 shows the organization and pinout of the STD BUS with mnemonic function and signal flow relative to the 7802 processor card.

STD BUS

	COMPONENT SIDE				CIRCUIT SIDE			
	PIN	MNEMONIC	SIGNAL FLOW	DESCRIPTION	PIN	MNEMONIC	SIGNAL FLOW	DESCRIPTION
LOGIC POWER BUS	1	+5V	In	+5 Volts DC (Bussed)	2	+5V	In	+5 Volts DC (Bussed)
	3	GND	In	Digital Ground (Bussed)	4	GND	In	Digital Ground (Bussed)
	5	-5V	In	-5 Volts DC	6	-5V	In	-5 Volts DC
DATA BUS	7	D3	In/Out	Low Order Data Bus	8	D7	In/Out	High Order Data Bus
	9	D2	In/Out	Low Order Data Bus	10	D6	In/Out	High Order Data Bus
	11	D1	In/Out	Low Order Data Bus	12	D5	In/Out	High Order Data Bus
	13	D0	In/Out	Low Order Data Bus	14	D4	In/Out	High Order Data Bus
ADDRESS BUS	15	A7	Out	Low Order Address Bus	16	A15	Out	High Order Address Bus
	17	A6	Out	Low Order Address Bus	18	A14	Out	High Order Address Bus
	19	A5	Out	Low Order Address Bus	20	A13	Out	High Order Address Bus
	21	A4	Out	Low Order Address Bus	22	A12	Out	High Order Address Bus
	23	A3	Out	Low Order Address Bus	24	A11	Out	High Order Address Bus
	25	A2	Out	Low Order Address Bus	26	A10	Out	High Order Address Bus
	27	A1	Out	Low Order Address Bus	28	A 9	Out	High Order Address Bus
	29	A0	Out	Low Order Address Bus	30	A 8	Out	High Order Address Bus
CONTROL BUS	31	WR*	Out	Write to Memory or I/O	32	RD*	Out	Read to Memory or I/O
	33	IORQ*	Out	I/O Address Select	34	MEMRQ*	Out	Memory Address Select
	35	IOEXP*	Out	I/O Expansion	36	MEMEX*	In/Out	Memory Expansion
	37	REFRESH*	Out	Refresh Timing	38	MCSYNC*	Out	CPU Machine Cycle Sync.
	39	STATUS 1*	Out	CPU Status	40	STATUS 0*	Out	CPU Status
	41	BUSAK*	Out	Bus Acknowledge	42	BUSRQ*	In	Bus Request
	43	INTAK*	—	Interrupt Acknowledge	44	INTRQ*	In	Interrupt Request
	45	WAITRQ*	—	Wait Request	46	NMIRQ*	In	Non-Maskable Interrupt
	47	SYSRESET*	Out	System Reset	48	PBRESET*	In	Push Button Reset
	49	CLOCK*	—	Clock from Processor	50	CNTRL*	In	AUX Timing
	51	PCO	Out	Priority Chain Out	52	PCI	In	Priority Chain In
POWER BUS	53	AUXGND	In	AUX Ground (Bussed)	54	AUXGND	In	AUX Ground (Bussed)
	55	AUX +V	In	AUX Positive(+12 Volts DC)	56	AUX -V	In	AUX Negative (-12 Volts DC)

Figure 2-1. The STD BUS.

STD BUS Pin Utilization by 7802

Since the STD BUS standard does not specify timing or require that all available pins be used, the timing and signal allocation assumes many of the characteristics of the microprocessor type used. The characteristics of the 7802 are dictated by its 6800 microprocessor, with LSTTL buffering added to enhance the card's drive capability. The buffers decrease memory and I/O access time slightly.

The allocation of STD BUS lines for the 7802 is given below:

1. **Logic Power Bus**: +5V (pins 1,2) and Logic Ground (pins 3,4) supply operating power to the 7802. Pins 5 and 6 are open.

2. **Data Bus:** Pins 7 through 14 form an 8-bit bidirectional 3-state data bus as shown in Figure 2-1. High level active data flows between the 7802 and its peripheral cards over this bus. When the 7802 fetches data from its onboard memory sockets, this data also appears on the STD Data Bus.

   With the exception of Direct Memory Access (DMA) operations, the 7802 controls the direction of data flow with its MEMRQ*, IORQ*, RD*, and WR* control signal outputs. Peripheral cards are required to release the data bus to the high impedance state except when addressed and directed to drive the data bus by the 7802. The 7802 releases the Data Bus when BUSAK* is active in response to BUSRQ*, as in DMA operations.

3. **Address Bus:** Pins 15 through 30 form a 16-bit 3-state address bus as shown in Figure 2-1. The 7802 drives high level active 16-bit memory addresses over these lines, and 8-bit I/O port addresses over the eight low-order address lines (A0 through A7 on pins 15, 17, 19, 21, 23, 25, 27 and 29).

   The 7802 releases the Address Bus when BUSAK* is active in response to BUSRQ*, as in DMA operations.

4. **Control Bus**: Pins 31 through 52 provide control signals for memory, I/O, interrupt, and fundamental system operations. Figure 2-2 summarizes these signals and shows how they are derived from 6800 signals.

   The 7802 releases the Control Bus during BUSAK* in response to BUSRQ*, except for the following output signals: MEMEX*, IOEXP*, BUSAK*, PCO.

5. **Auxiliary Power Bus**: Pins 53 through 56 are not used by the 7802 and are electrically open.

The 7802 does **not** meet all of the signal requirements of the STD BUS standard. The standard states that VMA* is STATUS 1* and R/W* is STATUS 0*. The 7802 defines STATUS 0* as VMA* with STATUS 1* jumper selectable to R/W*. Detailed timing information and specifications are in Section 3.

MNEMONIC	PIN	IN/OUT	FUNCTION	HOW DERIVED: 6800 NAME
WR*	31	OUT#	Write to memory or I/O	[RD] masked with Ø2TTL
RD*	32	OUT#	Read from memory or I/O	[RD]* masked with Ø2TTL
IORQ*	33	OUT#	A0-A7 hold valid I/O address	Address decode
MEMRQ*	34	OUT#	A0-A15 hold valid memory address	Address decode
IOEXP	35	OUT	I/O expansion control	User-removable ground
MEMEX	36	OUT	Memory expansion control	User-removable ground
REFRESH*	37	OUT#	Provide Refresh timing	Pull-up optional jumper connection
MCSYNC*	38	OUT#	One transition per machine cycle	[Ø2TTL]* from MC6875
STATUS 1*	39	OUT#	Read/Write from/to memory or I/O	[RD]—user optional jumper connection
STATUS 0*	40	OUT#	Valid address on address bus	[VMA]*
BUSAK*	41	OUT	Acknowledges BUSRQ*	[BA]*
BUSRQ*	42	IN	Bus request (DMA); synchronous processor halt and 3-state driver disable	[HALT*]
INTAK*	43		Not used	Electrically open
INTRQ*	44	IN	Maskable interrupt request	[IRQ*]
WAITRQ*	45		Not used	Electrically open
NMIRQ*	46	IN	Nonmaskable interrupt request	[NMI*]
SYSRESET	47	OUT#	System power-on and pushbutton reset one-shot output	[R*] from MC6875
PBRESET*	48	IN	Pushbutton reset input	[R*] on MC6875
CLOCK*	49		Not used	Electrically open
CNTRL*	50	IN	External clock input	[EXT CLK] on MC6875 — user optional jumper connection
PCI/PCO	52/51	IN/OUT	Priority chain. (Note: Trace on 7802 connects PCI to PCO to maintain chain continuity.)	

* Low level active # Output buffer disabled when BUSAK* active [ ] Denotes equivalent 6800 signal name.

Figure 2-2. 7802 Control Bus Signals.

7802 Processor Status: MCSYNC*, STATUS 0*, STATUS 1* Signals

MCSYNC*, STATUS 0*, and STATUS 1* signals provide encoded status information which is peculiar to the 6800 microprocessor. These signals are useful for displaying processor status in logic signal analyzers, and can be used to drive certain peripheral chips and systems designed to work with the 6800 specifically. The use of these signals is not recommended in systems where microprocessor device-type independence is a design goal.

MCSYNC* is the logical inversion of the MC6875ϕ2TTL signal. This signal is used as an enable for memory devices and as a synchronization signal for some of the 6800's special peripheral devices. MCSYNC*'s leading edge denotes the approximate start of a machine cycle. Counting the MCSYNC* transitions allows a logic signal analyzer to select a specific machine cycle within a multi-cycle instruction for analysis.

STATUS 0* is the logical inversion of the 6800 VMA signal. This signal indicates to peripheral devices that the address bus holds a valid address.

STATUS 1* is the logical inversion of the 6800 RD signal. This signal is activated by inserting a jumper in the provided pad, thus enabling the R*/W signal. This signal is used to control the direction of data flow between the CPU and memory or I/O devices.

SECTION 3
7802 Specifications

Power Requirements

PARAMETER	RECOMMENDED OPERATING LIMITS			ABSOLUTE NON-OPERATING LIMITS		UNITS
	MIN	TYP	MAX	MIN	MAX	
Vcc (Note 1)	4.75	5.00	5.25	0	5.50	Volts
Icc (Note 2)		1.25	1.85			Ampere

Notes:
1. In order to guarantee correct operation, the following power supply considerations apply:
   a. Vcc rise must be monotonic, rising from +0.50V to +4.75V in 10ms or less.
   b. If Vcc drops below +4.75V at any time, it must be returned to less than +0.50V before restoration to the specified operating range.
2. Icc specification assumes that all EPROM and RAM sockets on the 7802 are loaded. Subtract 75mA per 2716 EPROM and 50mA per 2114L RAM for each device not used (typical values).

Figure 3-1. 7802 Power Supply Specification.

The 6800 requires the passage of at least eight time states after power reaches 4.75V before it will place the reset address on the address bus. 2114L RAM devices require 10 milliseconds minimum after initial power-on for stabilization of internal bias oscillators. The 7802's power-on reset one-shot provides adequate stabilization delay if Vcc risetime is less than 10 milliseconds.

Address, Data and Control Buses meet STD BUS general electrical specifications, except:

WAITRQ*: This line is not connected on the 7802 because its function is duplicated by the BUSRQ* line.

PBRESET*: Input capacitance 0.5μF nominal. This line is recommended for momentary grounding by pushbutton.

MEMEX, IOEXP: These bus lines are grounded on the 7802 by user-removable jumper traces.

Drive Capability and Loading

The 7802's STD BUS Edge Connector Pin List (Figure 3-2) gives input loading and output drive capability in LSTTL loads as defined by the *Series 7000 Technical Manual.*

In general, input lines and disabled 3-state outputs present 5 LSTTL loads maximum (one LSTTL or MOS input plus 4.7K pullup resistor). Output lines can drive a minimum of 50 LSTTL loads. Pins which are unspecified in Figure 3-2 are electrically open.

Exceptions to the general loading rules are:

PBRESET* input capacitance which is 0.5μF typical.
PCI and PCO which are connected to each other but to nothing else on the 7802.
CNTRL* (EXT CLK) input which is 1 LSTTL load.

PIN NUMBER	OUTPUT (LSTTL DRIVE)	INPUT (LSTTL LOADS)	MNEMONIC	PIN NUMBER	OUTPUT (LSTTL DRIVE)	INPUT (LSTTL LOADS)	MNEMONIC
+5 VOLTS	IN		2	1		IN	+5 VOLTS
GROUND	IN		4	3		IN	GROUND
-5V			6	5			-5V
D7	5	55	8	7	55	5	D3
D6	5	55	10	9	55	5	D2
D5	5	55	12	11	55	5	D1
D4	5	55	14	13	55	5	D0
A15	5	55	16	15	55	5	A7
A14	5	55	18	17	55	5	A6
A13	5	55	20	19	55	5	A5
A12	5	55	22	21	55	5	A4
A11	5	55	24	13	55	5	A3
A10	5	55	26	25	55	5	A2
A9	5	55	28	27	55	5	A1
A8	5	55	30	29	55	5	A0
RD*	5	55	32	31	55	5	WR*
MEMRQ*	5	55	34	33	55	5	IORQ*
MEMEX (GROUND)		OUT	36	35	OUT		IOEXP (GROUND)
MCSYNC* (Ø2*)	5	55	38	37	[60]		REFRESH* (DRIVER)
STATUS 0* (VMA*)	5	55	40	39			STATUS 1*
BUSRQ*	5		42	41	55	5	BUSAK*
INTRQ*	5		44	43			INTAK*
NMIRQ*	5		46	45			WAITRQ*
PBRESET*	0.5μF		48	47	55	5	SYSRESET*
CNTRL*	[1]		50	49			CLOCK*
PC1	IN		52	51		OUT	PC0
AUX GND			54	53			AUX GND
AUX -V			56	55			AUX +V

*** Active low-level logic**

[ ] Brackets indicate user optional connection.

Figure 3-2. Edge Connector Pin List for the 7802.

Clock Generator

The 7802 uses the MC6875 two-phase microprocessor clock as its primary timing element. The 6875 provides the non-overlapping 2-phase clock needed by the 6800. It uses a 4MHz crystal oscillator circuit input to generate an operating frequency of 1MHz. This allows a cycle time (or time state) of 1μs. The time state is the shortest program-related period of interest in the system. Instruction execution times are computed as whole multiples of the time state clock period (Section 5).

The 7802 is shipped with a 4MHz crystal installed which sets the system's time state period to 1μs. If desired, the user can substitute a different crystal or he can replace the crystal with a TTL-compatible clock signal generated externally. The cycle time specification for an MC6800 microprocessor calls for a minimum cycle time of 1μs through a maximum of 10μsec.

Timing Specifications (Based on 1μs $\pm$0.5% time states)

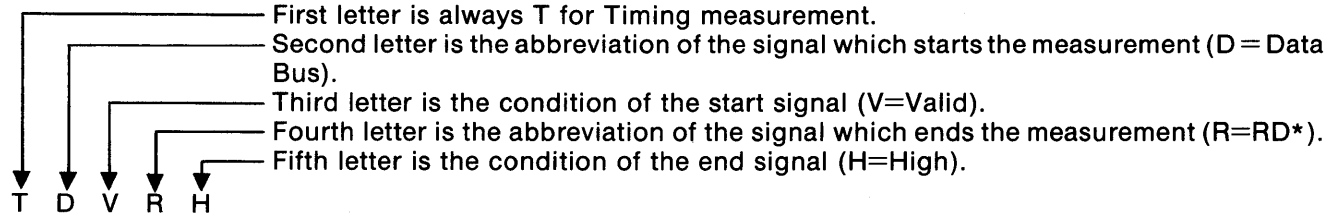
An understanding of the 7802's signal timing characteristics is necessary for the selection of speed-compatible memory devices, I/O functions, and other peripheral STD BUS cards, and for real-time logic analysis of 7802-based STD BUS card systems.

The 7802's timing characteristics are established by its 6800 microprocessor with additional delays added by LSTTL buffers. The basic operations performed by the 7802 and the signals controlling these operations are shown in Figure 3-3.

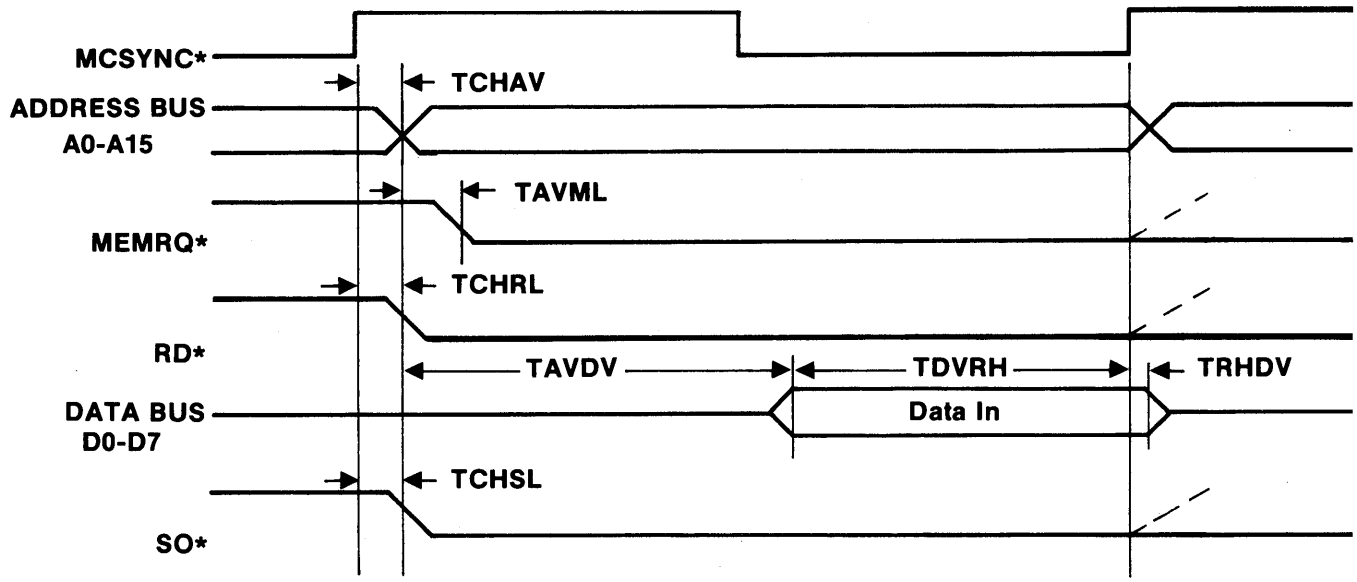
SIGNALS	OPERATION	WAVEFORM
MEMRQ* RD* A0-A15	Read from memory	Figure 3-4
MEMRQ* WR* A0-A15	Write to memory	Figure 3-5
IORQ* RD* A0-A7	Read from an input port	Figure 3-6
IORQ* WR* A0-A7	Write to an output port	Figure 3-7

Figure 3-3. Basic 7802 Operations.

The waveforms on the following pages show timing measurements as a 5-letter code as follows:



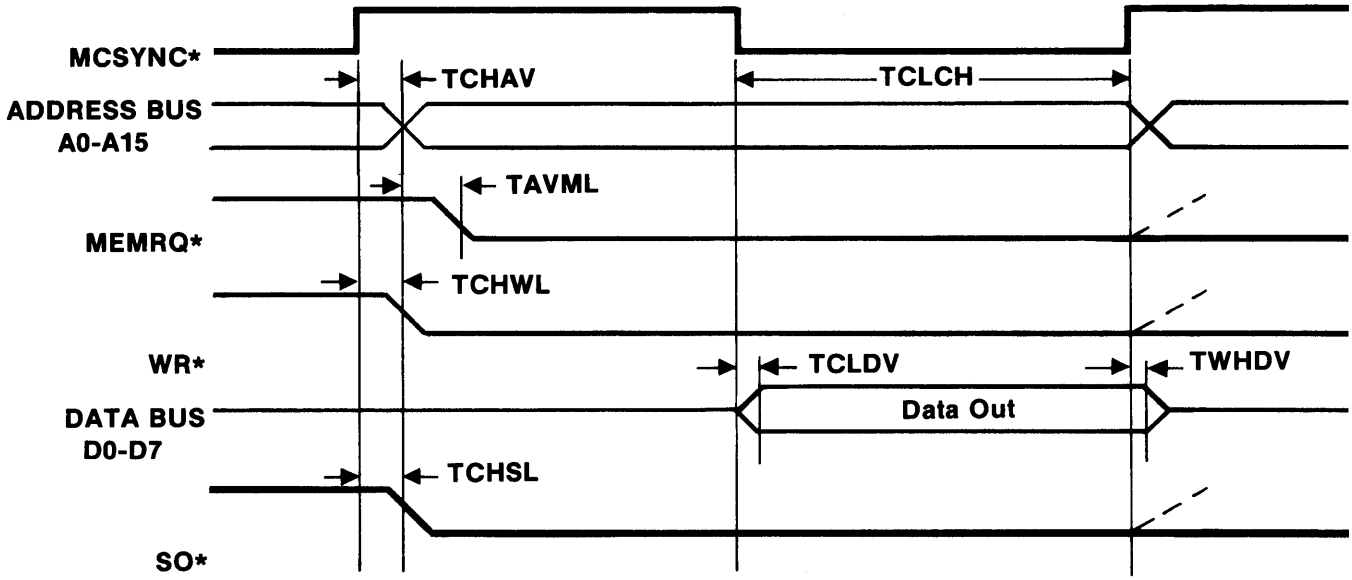
For example, TDVRH stands for **T**ime from **D**ata **V**alid until **R**D* (READ) **H**igh inactive. Specific abbreviations are given in the legend on each page of the specification.



	LEGEND
C	MCSYNC*
A	Address, A0-A15
M	MEMRQ*
R	RD*
D	Data, D0-D7
S	SO*
V	Valid
H	High
L	Low

SYMBOL	PARAMETER	NANOSECONDS		
		MIN	TYP	MAX
TCHAV	Address delay		220	300
TCHRL	Read active		220	300
TCHSL	VMA active		220	300
TAVML	Address valid before MEMRQ* active*			109
TAVDV	Address valid before Data Bus active		540	
TDVRH	Data set up time (Read)	100		
TRHDV	Data hold time	10		

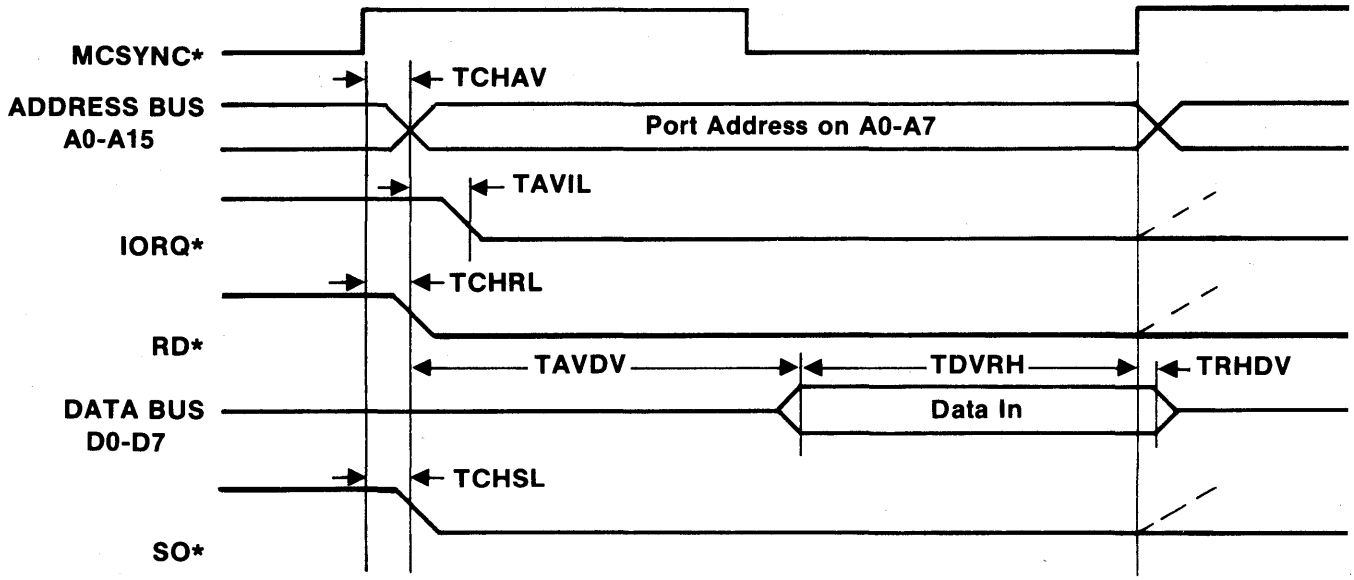
Figure 3-4. Read from Memory, 7802.



	LEGEND
C	MCSYNC*
A	Address, A0-A15
M	MEMRQ*
W	WR*
D	Data, D0-D7
S	SO*
V	Valid
H	High
L	Low

SYMBOL	PARAMETER	NANOSECONDS		
		MIN	TYP	MAX
TCHAV	Address valid		220	300
TCHWL	Write active		220	300
TCHSL	VMA active		220	300
TAVML	Address valid before MEMRQ* active*			109
TCLCH	Data Bus enable delay	450		
TCLDV	Data delay time (Write)		165	225
TWHDV	Data hold time	10		

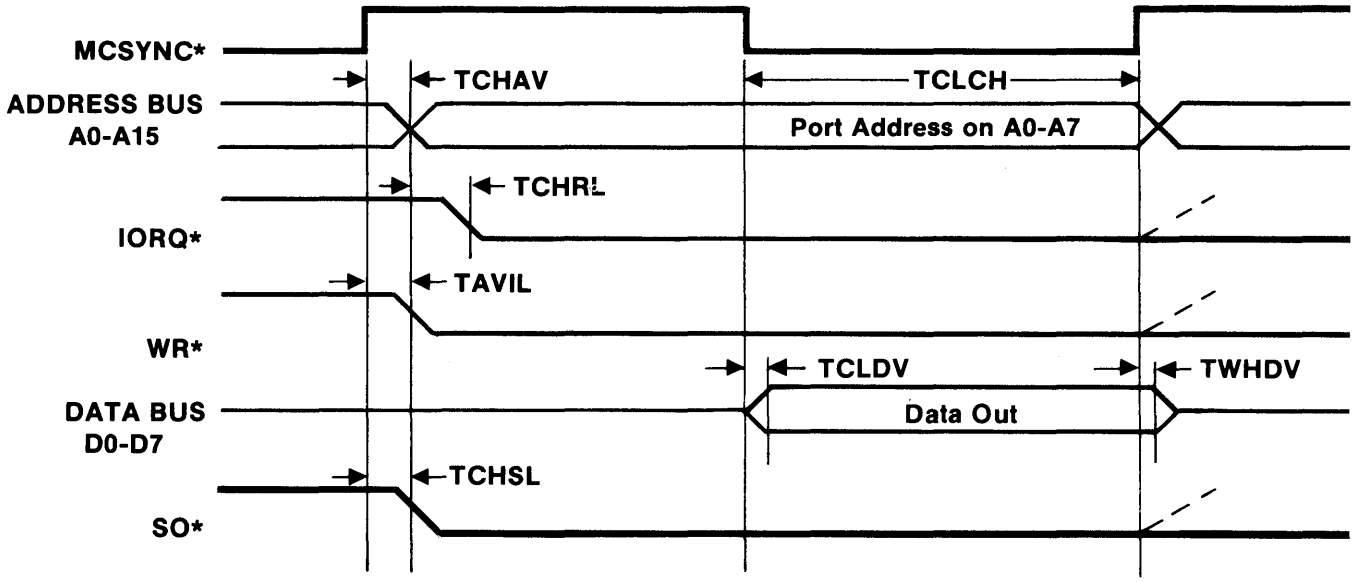
Figure 3-5. Write to Memory, 7802.



	LEGEND
C	MCSYNC*
A	Address, A0-A15
I	IORQ*
R	RD*
D	Data, D0-D7
S	SO*
V	Valid
H	High
L	Low

SYMBOL	PARAMETER	NANOSECONDS		
		MIN	TYP	MAX
TCHAV	Address delay		220	300
TCHRL	READ active		220	300
TCHSL	VMA active		220	300
TAVIL	Address valid before IORQ* active			109
TAVDV	Address valid before Data Bus active			540
TDVRH	Data set up time (Read)	100		
TRHDV	Data hold time	10		

Figure 3-6. Read from Input Port, 7802.



	LEGEND
C	MCSYNC*
A	Address, A0-A15
I	IORQ*
W	WR*
D	Data, D0-D7
S	SO*
V	Valid
H	High
L	Low

SYMBOL	PARAMETER	NANOSECONDS		
		MIN	TYP	MAX
TCHAV	Address valid		220	300
TCHWL	Write active		220	300
TCHSL	VMA active		220	300
TAVIL	Address valid before IORQ* active*			109
TCLCH	Data Bus enable delay	450		
TCLDV	Data delay time (Write)		165	225
TWHDV	Data hold time	10		

Figure 3-7. Write to Output Port, 7802.

Mechanical Specifications

The 7802 meets all STD BUS mechanical specifications. Refer to the *Series 7000 Technical Manual* for outline dimensions.

Environmental Specifications

PARAMETER	MIN	TYP	MAX	UNITS
Free Air Ambient Operating Temperature	0	25	55	°Celsius
Absolute Non-operating Free Air Ambient Temperature	-40		75	°Celsius
Relative Humidity Non-condensing	5		95	%
Absolute Non-operating Relative Humidity, Non-condensing	0		100	%

Figure 3-8. Environmental Specifications.

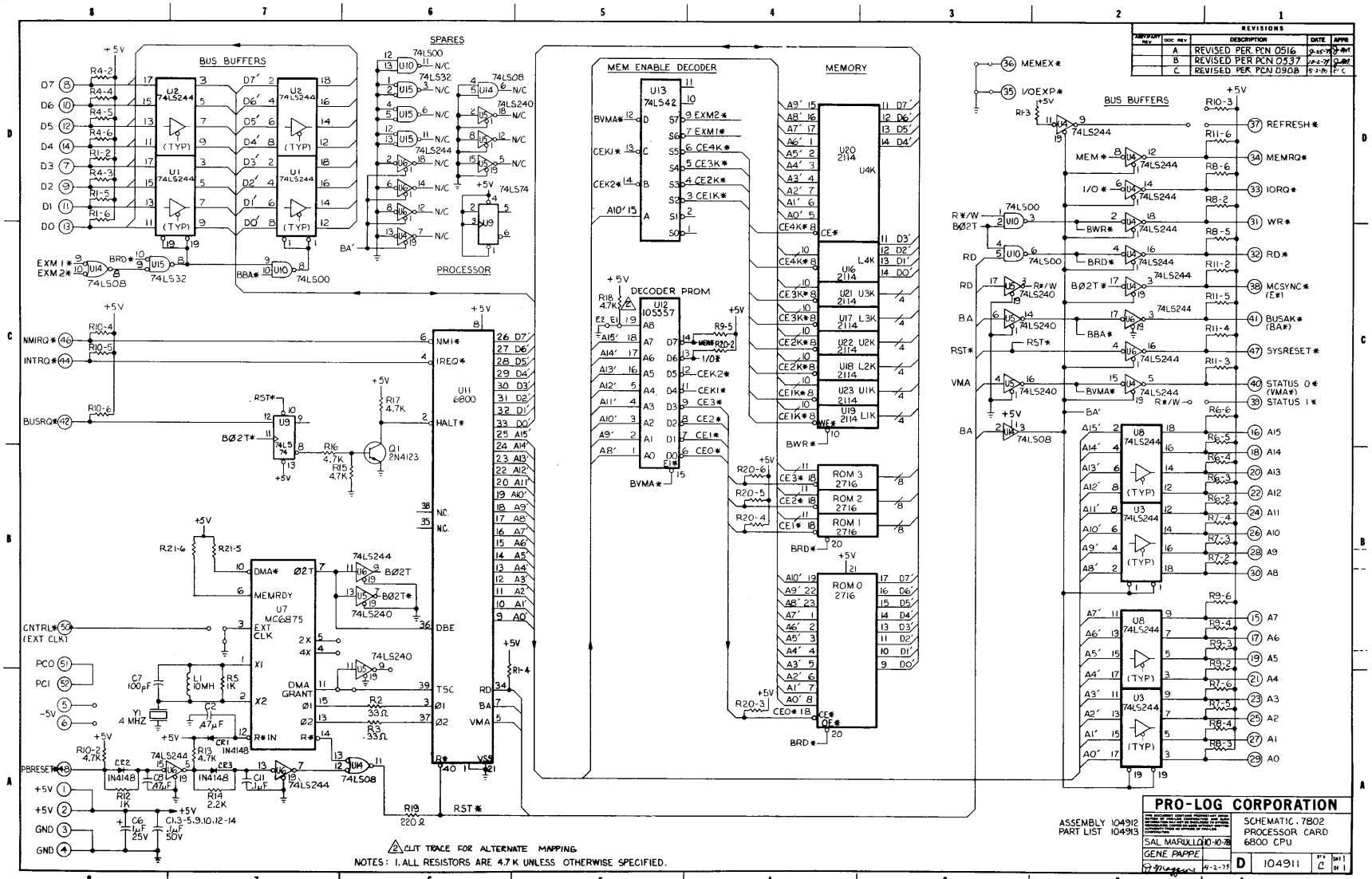


Figure 3-9. Schematic, 7802.

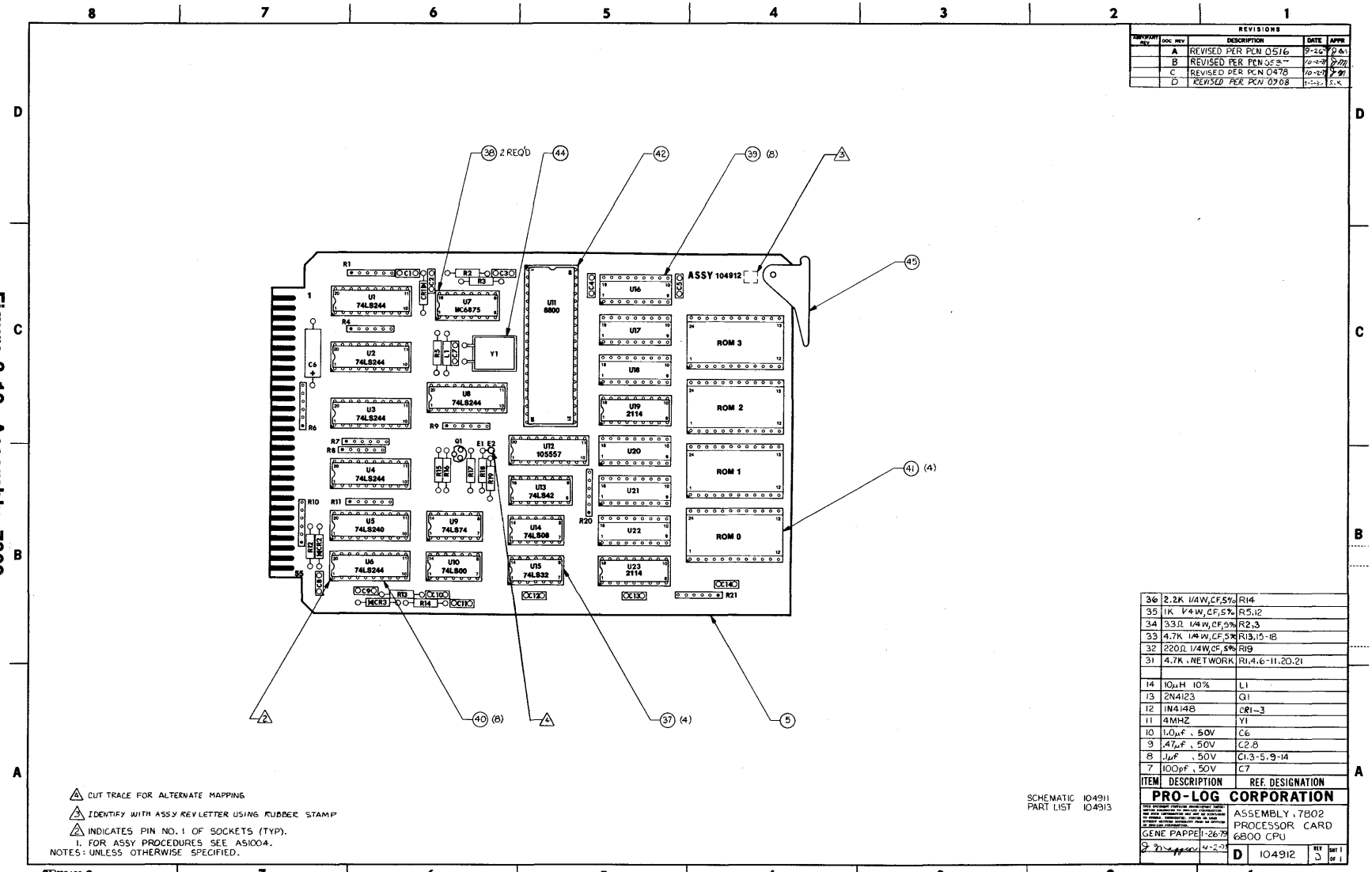


Figure 3-10. Assembly, 7802.

SECTION 4
Architecture and Instruction Set

6800 ARCHITECTURE

The 6800 architecture shown in Figure 4-1 consists of an 8-bit instruction register, a 16-bit stack pointer register, a 16-bit index register, an address ALU, and an 8-bit data ALU with two accumulators and a 6-bit Flag register.

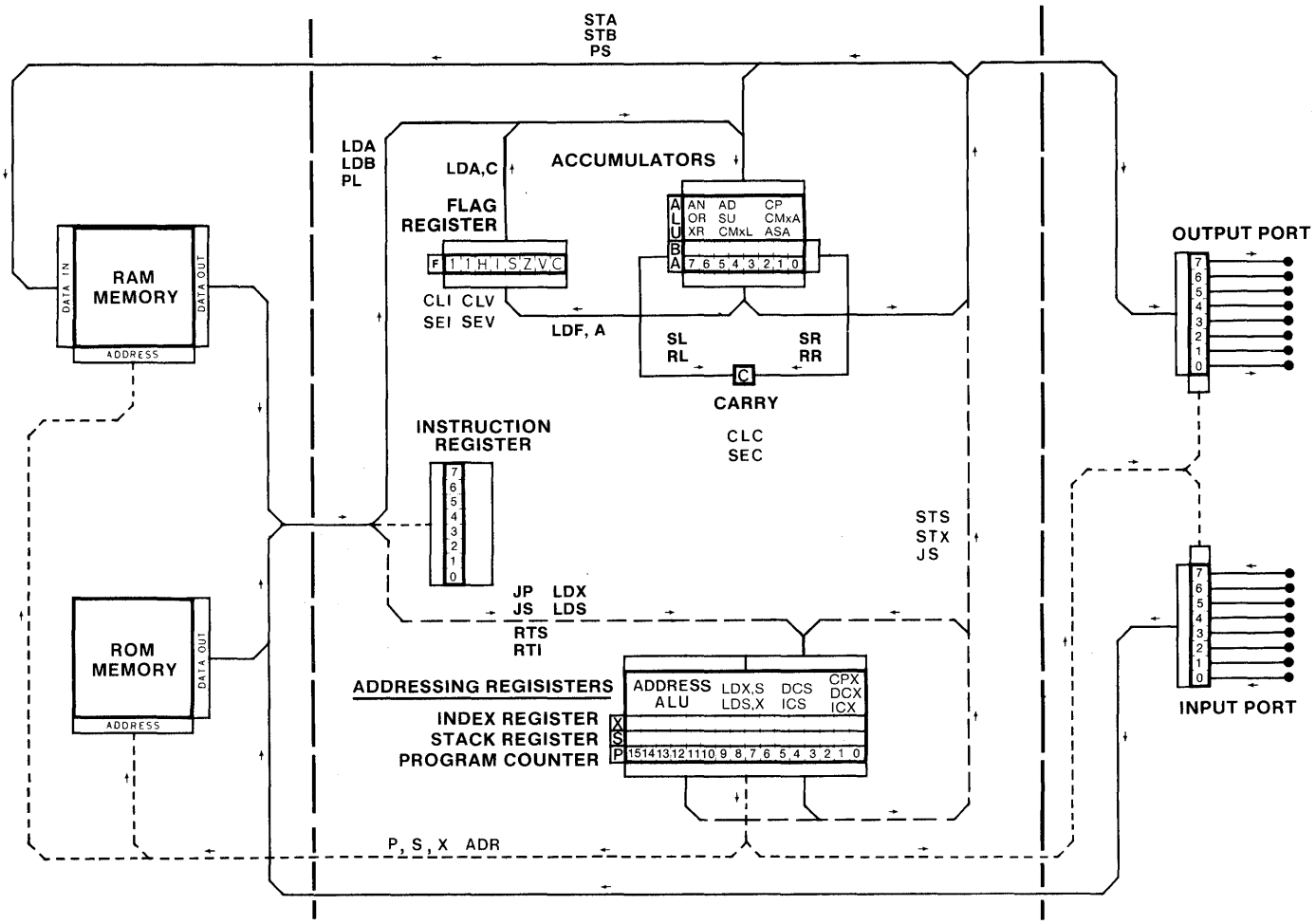


Figure 4-1. 6800 System Data Flow.

Instruction Register

The 8-bit instruction register provides storage and decoding for the instruction words as they are received from program memory.

The second and third words of multiple-byte instructions bypass the instruction register and are either data for the accumulators or a direct or offset address for the addressing registers.

Program Address Counter

The 16-bit program address counter keeps track of the location of the next instruction to be executed from the program memory. The program address counter increments automatically for each instruction word unless the instruction is a JUMP or RETURN which modifies the count by loading a new address. Relative and indexed instructions do not affect the program address counter except for JUMPs.

Stack Pointer

A 16-bit auto-counting stack pointer provides the address of the stack location in external RAM. The stack pointer is used for controlling subroutines, interrupts, and data handling.

Subroutine return addresses are automatically stored on the stack by the Jump to Subroutine instruction and retrieved by a Return from Subroutine.

Interrupt return addresses plus the Index register, Flag register, and both accumulators are automatically stored on the stack by an interrupt response and retrieved by a Return from Interrupt.

Accumulator data is stored on the stack with a PUSH instruction and retrieved by a PULL instruction.

Index Register

The 16-bit index register provides the address for the indexed and base relative addressing modes. The 8-bit offset from the second instruction byte is added with the Index value to provide the memory address.

The index register can be loaded from, stored to, and compared with memory, either directly, indexed, or immediately. It can also be counted up or down and loaded with the stack pointer.

Data ALU and Accumulators

The two accumulators and their associated carry bit are part of the data ALU. The data ALU provides add and subtract with or without carry, AND, OR, exclusive OR, compare and complement. The arithmetic and logic operations can be performed on the accumulators from memory either indexed, short, direct, or immediately from the second word of the instruction. The accumulators can also be shifted and rotated to the left or to the right either with or without carry. The A accumulator can also be Decimal Adjusted.

Flag Register (Status)

The flag register contains six status flags; Carry (C), Zero (Z), Sign (S), Overflow (V), Decimal (D), and Interrupt (I). The C, Z, S and V flags can be tested by the relative JUMP instructions to make program decisions. The D flag affects the operation of the Decimal Adjust and the I flag controls the maskable interrupt. The C, I, and V flags can be SET and CLEARED by separate instructions.

Memory

The 16-bit addressing capability allows 64K of memory which can be any combination of ROM or RAM. RAM is required for stack operations to allow the use of subroutines and interrupt. Memory is shared by data, programs, and I/O. The instruction set allows immediate, relative, indexed, short, and direct addressing.

The relative and indexed memory modes are complete, allowing "relocatable programs" and "dynamic memory allocation" techniques to be implemented. The short form of direct addressing is limited to page 00 indicating that this space should be used for workspace when efficient programs are desired. There are no separate indirect operations thus requiring the use of an indexed operation with a 00 offset byte.

Inputs and Outputs

I/O is memory-mapped (i.e., there are no separate I/O instructions). I/O ports use the same data, address, and control lines as memory. All memory instructions can be used for manipulating the I/O ports. In Figure 4-1, the I/O can be seen to share the same data and addressing paths as the memory.

Interrupt

The 6800 has two wired interrupt inputs each of which causes the program to retrieve a fixed address from Page FF. One of these wired interrupts is non-maskable.

The program address counter, the index register, both accumulators, and the flag register are automatically pushed to the stack by either interrupt or the HALT instruction.

Summary

The 6800 architecture has an external stack and full interrupt servicing features for multi-processing and includes relative and indexed addressing modes for "relocatable programs" and "dynamic memory allocation." The architecture is memory-oriented with no 8-bit data registers other than the two accumulators.

Some operations are performed between accumulators, but most are between either accumulator and memory or direct with memory.

6800 INSTRUCTIONS

This section contains an explanation of the 6800 addressing modes and the 6800 instruction set presented in concise tabular form. The instruction tables give a cross-reference between the mnemonic (abbreviated form of the instruction) and the hexadecimal operation code, together with a brief description of the instruction and its effect on the condition code flags.

Unassigned Operation Codes

Not all of the 256 possible 8-bit instruction operation codes are defined for the 6800 microprocessor. These unassigned codes do not appear in this manual. They are **not** NOP instructions and should not be used. In general, these codes are untested and unsupported by manufacturers of the 6800 microprocessor chip and the functions resulting from the use of these codes are liable to change without advance notice as design improvements are made.

STD Instruction Mnemonics

The STD Instruction Mnemonics are a standard set of processor instruction abbreviations suitable for use as an assembly language for writing programs.

These mnemonics are standard in that they do not change but keep the same meaning regardless of the processor they are applied to. They are also standard in that they are derived from a set of easily understood rules.

The instruction mnemonic is an abbreviated action statement containing an operator, a locator, and a qualifier plus a supplemental and separate modifier.

1. The operator is a unique two letter abbreviation that suggests the action.
2. The locator follows the operator and designates the operand or data to be operated on. Instructions without operands ignore the locator.
3. The qualifier states the addressing mode or provides further qualifying information for compound instructions.
4. modifier carries detailed support information; labels, conditions, addressing, and data.

The operator, locator, and qualifier letters are strung together to form the instruction mnemonic. The modifier, when needed, stands alone either in its own separate column or separated by spaces or additional lines in written text.

	OPERATOR	LOCATOR	QUALIFIER	MODIFIER	INSTRUCTION DESCRIPTION
RTS	RT	S			Return from Subroutine
CLA	CL	A			Clear A
LDAD	LD	A	D		Load A Direct
LDA B	LD	A		B	Load A with B
LDAN (BC)	LD	A	N	(BC)	Load A indirect using BC as an Address Pointer
JS (LABEL)	JS			(LABEL)	Jump to Subroutine Located at (LABEL)

Figure 4-2. Examples of STD Instruction Mnemonic Structure.

6800 Addressing Modes

The instruction's addressing mode tells the Processor how to find the data to work with or how to find the address to jump to. Some instructions have only one addressing mode. Others allow the designer to select from more than one mode, giving an opportunity to trade execution speed and program storage requirements to the best advantage. The 6800 offers the following modes:

Inherent Addressing

One-byte instructions which operate on unique elements in the system. These elements (the Accumulators, Flags, Index Register, and Stack Pointer) need no identifying address, so the instruction simply defines the operation to be performed. Inherent instructions (also called implied instructions) require the least memory space and are often the fastest in execution; however, they are generally limited to working with the data already stored in the Processor element operated on.

Inherent data instructions include those which set and clear each of the condition code flags, those which operate on the A and B Accumulators together, Stack Pointer data operations, and those which operate on the content of the Stack Pointer and Index Register.

Inherent Program Control Operations include RTI and RTS, which obtain jump addresses via the Stack Pointer, and the machine control instructions NOP and HLT.

Examples:	ADA	B	Adds the content of Accumulator B to Accumulator A.
	LDS	X	Copies the data from the Index Register into the Stack Pointer.
	RTS		Loads the Program Counter with a previously stored return address identified by the Stack Pointer effecting a jump.

Immediate Addressing

Two-byte and three-byte instructions which themselves contain the data to be operated on; i.e., the data is immediately adjacent to the operation code as the next one or two bytes and is regarded as part of the instruction. Immediate instructions are often used to preload a register or memory location with a numeric value or bit pattern in preparation for subsequent operations. When an immediate instruction is stored in ROM, it necessarily loads a constant. The modifier I is used to designate immediate addressing.

Examples:	ADAI		Adds 25 HEX to the content of Accumulator A.
	—	25	
	LDSI		Loads the Stack Pointer with the address 0123 HEX.
	—	01	
	—	23	

Direct Addressing (Extended)

Three-byte instructions which themselves contain the full memory address of data to be operated on, or a jump address. The second byte contains the memory page number and the third byte contains the memory line number. The modifier D is used to designate direct addressing.

Examples:	ADAD		Adds the content of memory address 0123 to Accumulator A.
	—	01	
	—	23	
	JP		Causes the program to jump to address 0123 for an instruction.
	—	01	
	—	23	

Short Addressing (Direct)

Two-byte instructions which use the second byte to specify a line address (00-FF HEX) in memory page 00. Not all 6800-based systems implement memory page 00. If present, it may consist of 256 bytes of RAM storage, ROM as lookup tables, or I/O ports. The short instructions are provided in lieu of internal Processor scratch-pad workspace. If a 256-byte RAM is addressed as page 00 in the system, the short instructions provide a fast method of manipulating the data stored in memory addresses 0000 through 00FF. The designer may also wire I/O ports to these addresses and use the direct instructions for high speed I/O bit manipulation, or ROM as lookup tables for high speed data conversion. Any combination of these elements may be addressed in page 00 as required by the application. When a short instruction is used, Address Bus bits 8-15 are forced to zero (page 00) and bits 0-7 assume the states designated by the second byte of the direct instruction. The modifier S is used to designate short addressing.

Examples:	ADAS		The content of memory address 0025 is added to Accumulator A
	—	25	
	LDSS		The 16-bit Stack Pointer is loaded from two consecutive memory locations: SP bits
	—	25	8-15 are loaded from memory location 0025, and bits 0-7 are loaded from memory location 0026 (the double work operation is controlled automatically by the Processor).

Indexed Addressing

Two-byte instructions which form a data address or jump address by adding the second byte of the instruction (called the "offset") to the content of the Index Register.

The offset value is an 8-bit unsigned binary number which can be from 0 to +255 (00 to FF HEX). Thus while the 16-bit index register can specify any of 65K memory addresses, the indexed instruction can only modify this address by adding from 0 to 255 locations. Indexed instructions can address memory and I/O ports for data, or they can be used to form unconditional or subroutine jump addresses. The X modifier is used to designate indexed addressing.

Relative Addressing

Two-byte instructions used only for program jumps (also called branches). Like Indexed Addressing, the second byte is an 8-bit signed binary offset value giving a relative range of -128 to +127 memory locations. This is the full range allowed for relative jumps. Note that many of the relative jump instructions are conditional; thus the jump may or may not occur, depending on the current state(s) of the flag(s) tested. The modifier R is used to designate relative addressing.

Examples:	JPR	Z0	Causes a jump five locations forward in the program if the Z flag is clear (last opera-
	—	05	tion resulted in non-zero).
	JPR	Z1	Causes a jump five locations backward in the program if the Z flag is set (last oper-
	—	FB	ation resulted in zero).

The Relative Addressing tables are used to convert a decimal number of memory locations to a signed binary offset value given in hexadecimal. Figure 4-5 gives values for forward relative addressing and for backward relative addressing. Use these values for both Relative and Indexed addressing modes.

MOTOROLA	STD
ADC A or B	ACA, ACB
ADD A or B ABA	ADA, ADB
AND A or B	ANA, ANB
BIT A or B	ANAV, ANBV
BLT, BGT	JPR CX
BSR, BRA	JS, JP
CPA A or B CBA	CPA, CPB
EOR A or B	XRA, XRB
LDA A or B TAB, TBA	LDA, LDB
ORA A or B	ORA, ORB
PSH, PUL	PS, PL
SBC A or B	SCA, SCB
STA A or B	STA, STB
SUB A or B SBA	SUA, SUB
SWI	JI
TAP, TPA	LDF A, LDA F
TST	CPxZ

Figure 4-3. Mnemonic Cross Reference Table, Motorola to STD.

8-BIT OPERATIONS

INSTR.	LOC. QUAL. MOD.	ACC A (1 Byte)	ACC B (1 Byte)	MEMORY, M ADDRESSING MODE I dd (2 Byte)	S mL (2 Byte)	X +rr (2 Byte)	D mPmL (3 Byte)	FLAG STATUS	DESCRIPTION OF OPERATION
LDA	x	—	17	86	96	A6	B6	s,z,V0	Load A with x
LDB	x	16	—	C6	D6	E6	F6	s,z,V0	Load B with x
STA	x	—	—	—	97	A7	B7	s,z,V0	Store Accumulator A
STB	x	—	—	—	D7	E7	F7	s,z,V0	Store Accumulator B
PSx		36	37	—	—	—	—		Push Data, Stack -1
PL x		32	33	—	—	—	—		Pull Data, Stack +1
ICx	m	4C	5C	—	—	6C	7C	s,z,v	Increment
DCx	m	4A	5A	—	—	6A	7A	s,z,v	Decrement
ADA	x	—	1B	8B	9B	AB	BB	d,s,z,v,c	ADD to A
ADB	x	—	—	CB	DB	EB	FB	d,s,z,v,c	ADD to B
ACA	x	—	—	89	99	A9	B9	d,s,z,v,c	ADD to A with carry
ACB	x	—	—	C9	D9	E9	F9	d,s,z,v,c	ADD to B with carry
SUA	x	—	10	80	90	A0	B0	s,z,v,c	Subtract from A
SUB	x	—	—	C0	D0	E0	F0	s,z,v,c	Subtract from B
SCA	x	—	—	82	92	A2	B2	s,z,v,c	Subtract from A with Carry
SCB	x	—	—	C2	D2	E2	F2	s,z,v,c	Subtract from B with Carry
ANA	x	—	—	84	94	A4	B4	s,z,V0	AND with A
ANB	x	—	—	C4	D4	E4	F4	s,z,V0	AND with B
ORA	x	—	—	8A	9A	AA	BA	s,z,V0	OR with A, Inclusive
ORB	x	—	—	CA	DA	EA	FA	s,z,V0	OR with B, Inclusive
XRA	x	—	—	88	98	A8	B8	s,z,V0	OR with A, Exclusive
XRB	x	—	—	C8	D8	E8	F8	s,z,V0	OR with B, Exclusive
CPA	x	—	11	81	91	A1	B1	s,z,v,c	Compare with A
CPB	x	—	—	C1	D1	E1	F1	s,z,v,c	Compare with B
ANAT	x	—	—	85	95	A5	B5	s,z,V0	AND with A Test
ANBT	x	—	—	C5	D5	E5	F5	s,z,V0	AND with B Test
CMxL	m	43	53	—	—	63	73	s,z,V0,C1	Complement 1's (Logical)
CPxZ	m	4D	5D	—	—	6D	7D	s,z,V0,C0	Compare with ZERO
CLx	m	4F	5F	—	—	6F	7F	S0,Z1,V0,C0	Clear
CMxA	m	40	50	—	—	60	70	s,z,v,c	Complement 2's (Arithmetic)
AJAD	—	19	—	—	—	—	—	s,z,v,c	Adjust A Decimal
RLx C	m	49	59	—	—	69	79	s,z,v,c	Rotate left through carry
RRx C	m	46	56	—	—	66	76	s,z,v,c	Rotate right through carry
SLxA	m	48	58	—	—	68	78	s,z,v,c	Shift left Arithmetic
SRxA	m	47	57	—	—	67	77	s,z,v,c	Shift right Arithmetic
SRxL	m	44	54	—	—	64	74	S0,z,v,c	Shift right Logical

16-BIT OPERATIONS

INSTR.	LOC. QUAL. MOD.	REG,R S (1 Byte)	REG,R X (1 Byte)	MEMORY, M ADDRESSING MODE I dHdL (3 Byte)	S mL (2 Byte)	X +rr (2 Byte)	D mPmL (3 Byte)	FLAG STATUS	DESCRIPTION OF OPERATION
LDX	x	30	—	CE	DE	EE	FE	s,z,V0	Load Index Register
LDS	x	—	35	8E	9E	AE	BE	s,z,V0	Load Stack Pointer
STX	m	—	—	—	DF	EF	FF	s,z,V0	Store Index Register
STS	m	—	—	—	9F	AF	BF	s,z,V0	Store Stack Pointer
DCx		34	09	—	—	—	—	z	Decrement Register
IC x		31	08	—	—	—	—	z	Increment Register
CPX	m	—	—	8C	9C	AC	BC	s,z,v	Compare Index Register

Figure 4-4. 6800 Programming Aid Tables.

PROGRAM ADDRESS CONTROL INSTRUCTIONS

INSTR.	QUAL. MOD.	1 Byte	R ±rr	X +rr	D mPmL	FLAG STATUS	DESCRIPTION OF OPERATION
			2 Byte		3 Byte		
JP	m	—	20	6E	7E		Jump Unconditional
JS	m	—	8D	AD	BD		Jump to Subroutine, Stack -2
RTS		39					Return from Subroutine, Stack +2
JI		3F				I1	Jump to Interrupt, Stack -7
RTI		3B				d,i,s,z,v,c	Return from Interrupt, Stack +7
HLT		3E				I1	Wait for Interrupt (Halt)
NOP		01					No Operation

Program Address Control Instructions

CONDITIONAL BRANCHING INSTRUCTIONS (Relative Only: 2 Bytes)

ARITHMETIC MEANING					LOGICAL MEANING			
INSTR.	MOD.	OP CODE	DESCRIPTION		INSTR.	MOD.	OP CODE	DESCRIPTION
JPR	H	22	Higher	>	JPR	C0•Z0	22	C=0 and Z=0
JPR	NH	23	Lower or same	≤	JPR	C1+Z1	23	C=1 or Z=1
JPR	NL	24	Higher or same	≥	JPR	C0	24	C=0
JPR	L	25	Lower	<	JPR	C1	25	C=1
JPR	NE	26	Not equal zero	≠ 0	JPR	Z0	26	Z=0
JPR	EQ	27	Equal zero	= 0	JPR	Z1	27	Z=1
JPR	NV	28	Overflow clear		JPR	V0	28	V=0
JPR	V	29	Overflow set		JPR	V1	29	V=1
JPR	P	2A	Plus	+	JPR	S0	2A	S=0
JPR	N	2B	Minus	-	JPR	S1	2B	S=1
JPR	GE	2C	≥ zero	≥ 0	JPR	S=V	2C	S1•V1 or S0•V0
JPR	LT	2D	< zero	< 0	JPR	S≠V	2D	S1•V0 or S0•V1
JPR	GT	2E	> zero	> 0	JPR	Z0•(S=V)	2E	
JPR	LE	2F	≤ zero	≤ 0	JPR	Z1+(S≠V)	2F	

Conditional Branching Instructions

CONDITION REGISTER INSTRUCTIONS (One Byte)

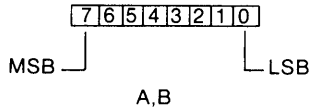
INSTR.	MOD.	OP CODE	FLAG STATUS	DESCRIPTION OF OPERATION
LDF	A	06	d,i,s,z,v,c	Load Flag Register with Accumulator A
LDA	F	07		Load Accumulator A with Flag Register
CLV	—	0A	V0	Clear Overflow
SEV	—	0B	V1	Set Overflow
CLC	—	0C	C0	Clear Carry
SEC	—	0D	C1	Set Carry
DSI	—	0E	I0	Disable Interrupt
ENI	—	0F	I1	Enable Interrupt

Condition Register Instructions

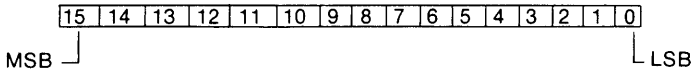
JI, HLT or INTERRUPT	STACK LOCATIONS	RTS
SP - 7 AFTER →	(AVAILABLE)	← INITIAL SP
SP - 6	1 1 D I S Z V C	SP + 1
SP - 5	ACC. B	SP + 2
SP - 4	ACC. A	SP + 3
SP - 3	XH	SP + 4
SP - 2	XL	SP + 5
SP - 1	PCH	SP + 6
INITIAL SP →	PCL	← SP + 7 AFTER

JSR	STACK LOCATIONS	RTS
SP - 2 AFTER →	(AVAILABLE)	← INITIAL SP
SP - 1	PCH	SP + 1
INITIAL SP →	PCL	← SP + 2 AFTER

RESTART POINTERS		
FUNCTION	**ADDRESS**	**CONTAINS**
MASKABLE INTERRUPT	FFF8	PCH
	FFF9	PCL
SOFTWARE INTERRUPT	FFFA	PCH
	FFFB	PCL
NONMASKABLE INTERRUPT	FFFC	PCH
	FFFD	PCL
RESET	FFFE	PCH
	FFFF	PCL



A,B



X,SP,PC

Data Transfer To/From Memory	BITS 15-8	M
	BITS 7-0	M+1

FORWARD RELATIVE OFFSET

ADDRESS FOR 0 THRU 127 COUNTS

0	00	16	10	32	20	48	30	64	40	80	50	96	60	112	70
1	01	17	11	33	21	49	31	65	41	81	51	97	61	113	71
2	02	18	12	34	22	50	32	66	42	82	52	98	62	114	72
3	03	19	13	35	23	51	33	67	43	83	53	99	63	115	73
4	04	20	14	36	24	52	34	68	44	84	54	100	64	116	74
5	05	21	15	37	25	53	35	69	45	85	55	101	65	117	75
6	06	22	16	38	26	54	36	70	46	86	56	102	66	118	76
7	07	23	17	39	27	55	37	71	47	87	57	103	67	119	77
8	08	24	18	40	28	56	38	72	48	88	58	104	68	120	78
9	09	25	19	41	29	57	39	73	49	89	59	105	69	121	79
10	0A	26	1A	42	2A	58	3A	74	4A	90	5A	106	6A	122	7A
11	0B	27	1B	43	2B	59	3B	75	4B	91	5B	107	6B	123	7B
12	0C	28	1C	44	2C	60	3C	76	4C	92	5C	108	6C	124	7C
13	0D	29	1D	45	2D	61	3D	77	4D	93	5D	109	6D	125	7D
14	0E	30	1E	46	2E	62	3E	78	4E	94	5E	110	6E	126	7E
15	0F	31	1F	47	2F	63	3F	79	4F	95	5F	111	6F	127	7F

COUNT ZERO AT SECOND ADDRESS AFTER JUMP MNEMONIC

BACKWARD RELATIVE OFFSET

ADDRESS FOR 1 THRU 128 COUNTS

1	FF	17	EF	33	DF	49	CF	65	BF	81	AF	97	9F	113	8F
2	FE	18	EE	34	DE	50	CE	66	BE	82	AE	98	9E	114	8E
3	FD	19	ED	35	DD	51	CD	67	BD	83	AD	99	9D	115	8D
4	FC	20	EC	36	DC	52	CC	68	BC	84	AC	100	9C	116	8C
5	FB	21	EB	37	DB	53	CB	69	BB	85	AB	101	9B	117	8B
6	FA	22	EA	38	DA	54	CA	70	BA	86	AA	102	9A	118	8A
7	F9	23	E9	39	D9	55	C9	71	B9	87	A9	103	99	119	89
8	F8	24	E8	40	D8	56	C8	72	B8	88	A8	104	98	120	88
9	F7	25	E7	41	D7	57	C7	73	B7	89	A7	105	97	121	87
10	F6	26	E6	42	D6	58	C6	74	B6	90	A6	106	96	122	86
11	F5	27	E5	43	D5	59	C5	75	B5	91	A5	107	95	123	85
12	F4	28	E4	44	D4	60	C4	76	B4	92	A4	108	94	124	84
13	F3	29	E3	45	D3	61	C3	77	B3	93	A3	109	93	125	83
14	F2	30	E2	46	D2	62	C2	78	B2	94	A2	110	92	126	82
15	F1	31	E1	47	D1	63	C1	79	B1	95	A1	111	91	127	81
16	F0	32	E0	48	D0	64	C0	80	B0	96	A0	112	90	128	80

COUNT ONE AT FIRST ADDRESS AFTER JUMP MNEMONIC

Figure 4-5. Relative Offset Tables.

SYSTEM CONTROL

Reset

The 6800 uses the RESET signal to reset and start from a power-down condition. It is used after power failure, for initial start-up of the sytem, or to reinitialize the system after start-up.

When the 6800 receives a RESET input, the following occurs:

The address of the reset routine is fetched from the Vector Table in memory; the contents of locations FFFE and FFFF are placed into the program counter.

The Interrupt Mask bit in the condition code register is set to prevent interrupts. (The program must later clear this bit if the system is to be able to recognize interrupt requests through the IRQ input).

The reset routine is processed; the program runs.

Interrupts

The 6800 has three types of interrupt, two driven from external device requests and one software driven.

When the 6800 receives an interrupt request, the following actions occur:

The internal registers (the program counter (PC), index register (IX), accumulators (A and B), and the condition code, or flag, register (F) are pushed onto the stack.

The Interrupt Mask bit is set to prevent further interrupts.

The address of the interrupt service routine is fetched from the Vector Table in memory.

FFF8, FFF9 for IRQ
FFFA, FFFB for JI (hex code 3F)
FFFC, FFFD for NMI

The interrupt service routine is then processed.

Upon completion of the interrupt service routine, the RTI instruction will cause the following actions:

The internal registers (F, B, A, IX, PC) are pulled off the stack.

The Interrupt Mask bit is restored to its condition prior to the interrupt.

The program continues from the point at which it was interrupted.

The 7802 has two interrupt request inputs, NMIRQ* and INTRQ*, available at the STD BUS backplane. The interrupt inputs have different characteristics as summarized in Figure 4-6.

STD BUS Interrupt Name	6800 Interrupt Name	Input Sensitivity	Priority	Processor Action Upon Response	Maskable?
NMIRQ*	NMI	Low level	High	Push internal registers onto stack. Set Interrupt Mask bit. Get interrupt service routine start address from FFFC,FFFD. Jump to service routine.	No
INTRQ*	IRQ	Low level	Low	Same as NMIRQ* response, except that interrupt service routine start address is fetched from FFF8,FFF9.	Yes

Figure 4-6. 7802 Interrupt Summary.

NMIRQ* (pin 46) is nonmaskable and cannot be disabled by the program. It is often used for catastrophic system events such as impending power failure, but it may be used for any interrupt function.

INTRQ* (pin 44) is a maskable interrupt and is enabled/disabled by the program with the ENI and DSI instructions which control the Interrupt Mask bit of the condition code register.

Pro-Log's 7320 Priority Interrupt Controller may be used to expand any of the 7802's interrupt inputs. Each 7320 card expands the interrupt line by a factor of eight and provides additional program control over the interrupt system while handling expansion signal protocol. See Figure 4-7.

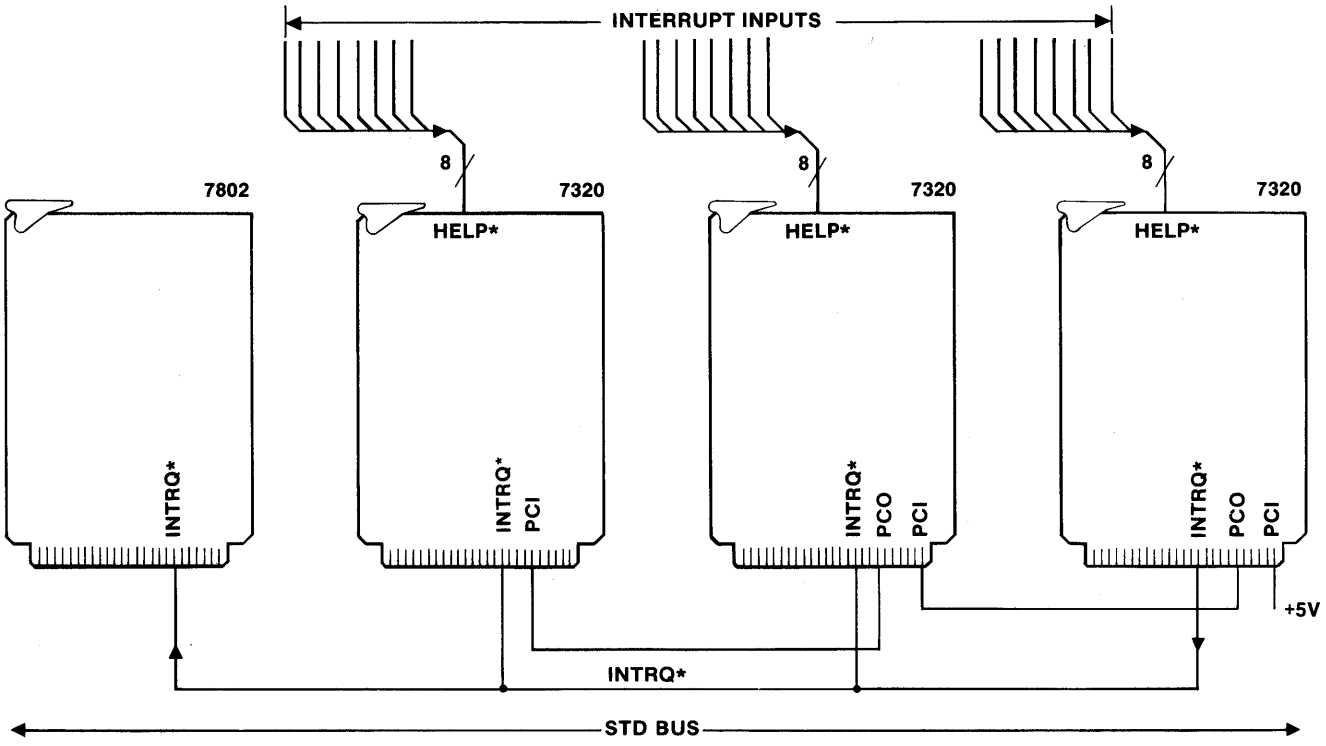


Figure 4-7. Example of 7802 Interrupt Inputs Expanded with three 7320 Cards.

SECTION 5
PROGRAM INSTRUCTION TIMING

Instruction Cycle Time

The 7802 6800 CPU Card operates by periodically retrieving groups of bits from program memory and performing operations defined by the bit patterns of these instruction words. The instruction cycle time of addressing the program memory, fetching the instruction word, and executing the operation requires multiple time intervals. The total time required to complete the instruction cycle is called the "instruction cycle time."

The instruction cycle time for the 7802 varies, with some instructions requiring more time to execute than others.

Instruction Words

The instruction set includes single-, double-, and triple-byte instruction words which require from two to twelve machine cycles to execute.

Machine Cycle

A machine cycle is the basic cycle time of the 6800 microprocessor. During each machine cycle, one of the following operations occurs: Read data from memory, read data from peripherals, write data to memory, write data to peripherals, or execute operation internally.

The time required for a machine cycle is determined by the frequency of the two-phase clock. 7802 systems operate with 1.0 microsecond machine cycles.

Instruction or Data Cycle

A machine cycle can be an instruction cycle or a data cycle, depending on whether memory is addressed by program counter for an instruction or by the program itself, the Stack Pointer, or the Index Register for data.

The first machine cycle is always an instruction cycle which fetches the instruction code from program memory. It is followed by one to eleven cycles according to the type of instruction and the addressing mode used.

Summary of Cycle by Cycle Operation

Figure 5-1 provides a detailed description of the information present on the Address Bus, Data Bus, Valid Memory Address line (VMA), and the Read/Write line (R/W) during each cycle for each instruction.

Figure 5-2 provides specific timing information for each of the 6800 instructions. The information is categorized in groups according to Addressing Mode and number of cycles per instruction. (In general, instructions with the same Address Mode and number of cycles execute in the same manner; exceptions are indicated in the table.)

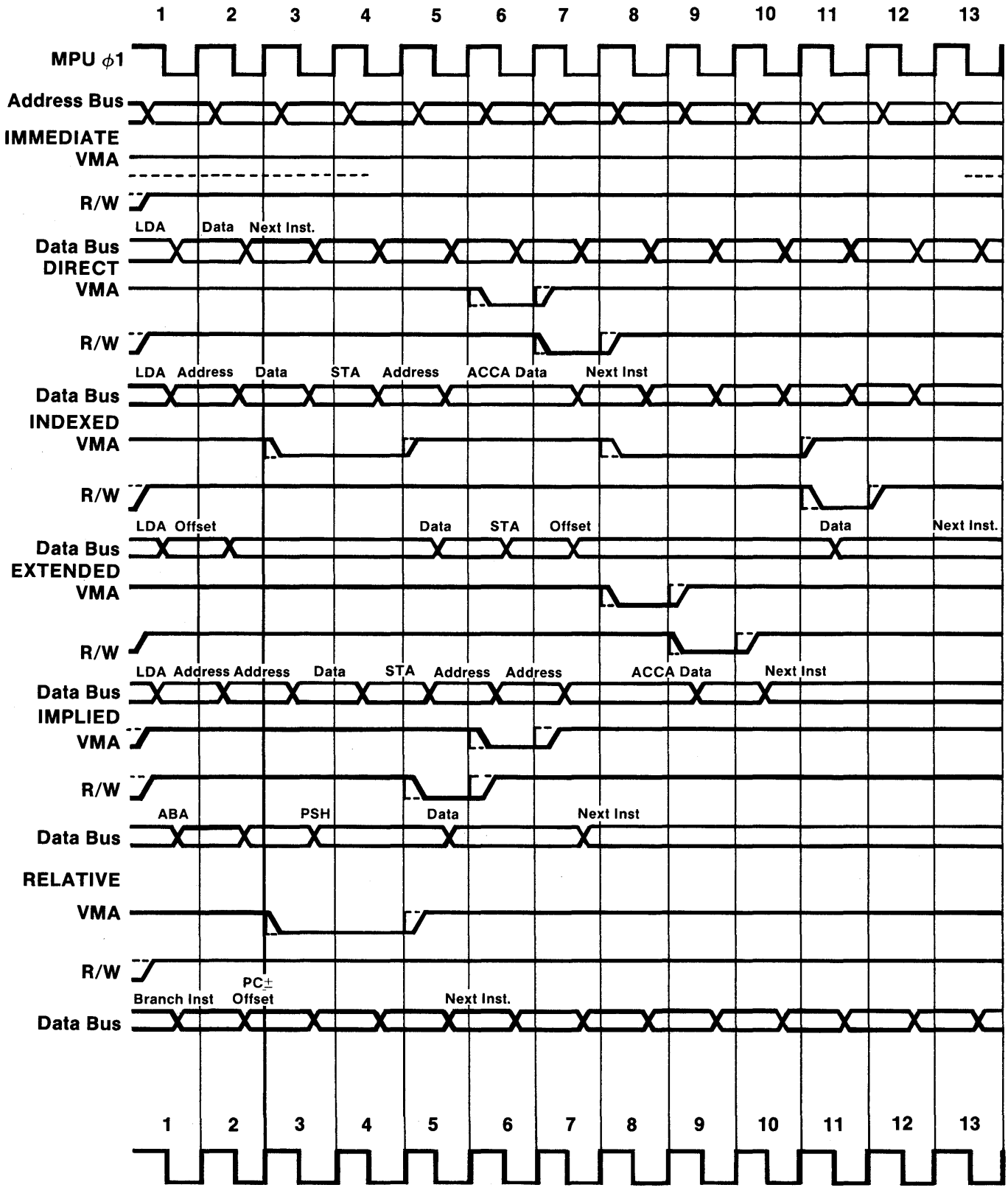


Figure 5-1. 6800 Instruction Timing.

OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#
01	NOP	INHER	2	1	26	JPR NE	REL	4	2	40	CMAA	INHER	2	1
06	LDF A		2	1	27	JPR EQ		4	2	43	CMAL		2	1
07	LDA F		2	1	28	JPR NV		4	2	44	SRAL		2	1
08	ICX		4	1	29	JPR V		4	2	46	RRAC		2	1
09	DCX		4	1	2A	JPR P		4	2	47	SRAA		2	1
0A	CLV		2	1	2B	JPR N		4	2	48	SLAA		2	1
0B	SEV		2	1	2C	JPR GE		4	2	49	RLAC		2	1
0C	CLC		2	1	2D	JPR LT		4	2	4A	DCA		2	1
0D	SEC		2	1	2E	JPR GT		4	2	4C	ICA		2	1
0E	DSI		2	1	2F	JPR LE	REL	4	2	4D	CPAZ		2	1
0F	ENI		2	1	30	LDX S	INHER	4	1	4F	CLA		2	1
10	SUA B		2	1	31	ICS		4	1	50	CMBA		2	1
11	CPA B		2	1	32	PLA		4	1	53	CMBL		2	1
16	LDB A		2	1	33	PLB		4	1	54	SRBL		2	1
17	LDA B		2	1	34	DCS		4	1	56	RRBC		2	1
19	AJAD		2	1	35	LDS X		4	1	57	SRBA		2	1
1B	ADA B	INHER	2	1	36	PSA		4	1	58	SLBA		2	1
20	JPR	REL	4	2	37	PSB		4	1	59	RLBC		2	1
22	JPR H		4	2	39	RTS		6	1	5A	DCB		2	1
23	JPR NH		4	2	3B	RTI		10	1	5C	ICB		2	1
24	JPR NL		4	2	3E	HLT		9	1	5D	CPBZ		2	1
25	JPR L	REL	4	2	3F	JI	INHER	12	1	5F	CLR	INHER	2	1

OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#
60	CMMA	INDXD	7	2	7E	JP	EXTND	3	3	98	XRA	DIR	3	2
63	CMML		7	2	7F	CLM	EXTND	6	3	99	ACA		3	2
64	SRML		7	2	80	SUA	IMMED	2	2	9A	ORA		3	2
66	RRMC		7	2	81	CPA		2	2	9B	ADA		3	2
67	SRMA		7	2	82	SCA		2	2	9C	CPX		4	2
68	SLML		7	2	84	ANA		2	2	9E	LDS		4	2
69	RLMC		7	2	85	ANAT		2	2	9F	STS	DIR	5	2
6A	DCM		7	2	86	LDA		2	2	A0	SUA	INDXD	5	2
6C	ICM		7	2	88	XRA		2	2	A1	CPA		5	2
6D	CPMZ		7	2	89	ACA		2	2	A2	SCA		5	2
6E	JPX		4	2	8A	ORA		2	2	A4	ANA		5	2
6F	CLM	INDXD	7	2	8B	ADA		2	2	A5	ANAT		5	2
70	CMMA	EXTND	6	3	8C	CPX	IMMED	3	3	A6	LDA		5	2
73	CMML		6	3	8D	JSR	REL	3	2	A7	STA		5	2
74	SRML		6	3	8E	LDS	IMMED	3	3	A8	XRA		5	2
76	RRMC		6	3	90	SUA	DIR	3	2	A9	ACA		5	2
77	SRMA		6	3	91	CPA		3	2	AA	ORA		6	2
78	SLML		6	3	92	SCA		3	2	AB	ADA		6	2
79	RLMC		6	3	94	ANA		3	2	AC	CPX		6	2
7A	DCM		6	3	95	ANAT		3	2	AD	JSX		8	2
7C	ICM		6	3	96	LDA		3	2	AE	LDS		6	2
7D	CPMZ	EXTND	6	3	97	STA	DIR	4	2	AF	STS	INDXD	7	2

OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#
B0	SUA	EXTND	4	3	C9	ACB	IMMED	2	2	E6	LDB	INDXD	5	2
B1	CPA		4	3	CA	ORB		2	2	E7	STB		6	2
B2	SCA		4	3	CB	ADB		2	2	E8	XRB		5	2
B4	ANA		4	3	CE	LDX	IMMED	3	3	E9	ACB		5	2
B5	ANAT		4	3	D0	SUB	DIR	3	2	EA	ORB		5	2
B6	LDA		4	3	D1	CPB		3	2	EB	ADB		5	2
B7	STA		5	3	D2	SCB		3	2	EE	LDX		6	2
B8	XRA		4	3	D4	ANB		3	2	EF	STX	INDXD	7	2
B9	ACA		4	3	D5	ANBT		3	2	F0	SUB	EXTND	4	3
BA	ORA		4	3	D6	LDB		3	2	F1	CPB		4	3
BB	ADA		4	3	D7	STB		4	2	F2	SCB		4	3
BC	CPX		8	3	D8	XRB		3	2	F4	ANB		4	3
BD	JS		9	3	D9	ACB		3	2	F5	ANBT		4	3
BE	LDS		5	3	DA	ORB		3	2	F6	LDB		4	3
BF	STS	EXTND	6	3	DB	ADB		3	2	F7	STB		5	3
C0	SUB	IMMED	2	2	DE	LDX		4	2	F8	XRB		4	3
C1	CPB		2	2	DF	STX	DIR	5	2	F9	ACB		4	3
C2	SCB		2	2	E0	SUB	INDXD	5	2	FA	ORB		4	3
C4	ANB		2	2	E1	CPB		5	2	FB	ADB		4	3
C5	ANBT		2	2	E2	SCB		5	2	FE	LDX		5	3
C6	LDB		2	2	E4	ANB		5	2	FF	STX	EXTND	5	3
C8	XRB	IMMED	2	2	E5	ANBT	INDXD	5	2					

~ Number of time states.
# Number of bytes.

Figure 5-2. Cycle timming Tables for 6800

Instruction Timing Example

The execution time for any routine or program segment is found by totalling all of the time states in all of the instructions executed. The factors affecting the execution time of a program segment are:

a. The clock frequency, which determines the time state period

b. The specific instructions used, which determine the number of time states in the segment

c. The number of instruction loops within the instruction sequence, and the number of times each loop is executed (loop iterations)

d. If the program segment has more than one entrance or exit, every combination of routes though the segment that are used by the program should be considered.

The following example shows how to compute execution times in a program segment. The 6800 is programmed to generate a series of five short pulses at an output port bit line. Determine the overall execution time of the program segment and the period of the pulses generated.

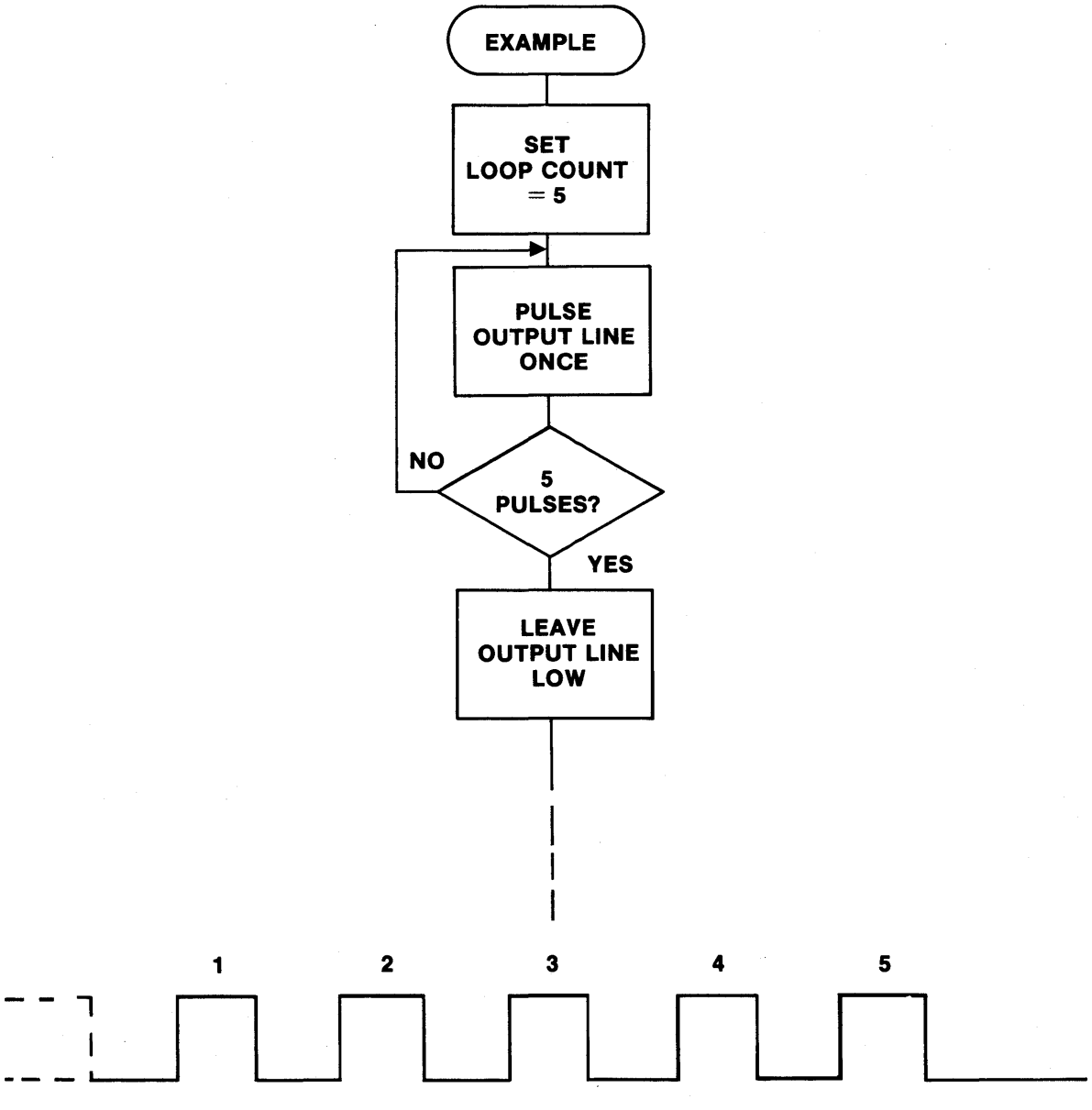


Figure 5-3. Instruction Segment Timing Example.

In the example in Figure 5-3, six of the program segment's nine instructions are within the loop and are executed 5 times each.

FLOW DIAGRAM FUNCTION	TIMES PERFORMED	INSTRUCTIONS		CYCLES	EXECUTION TIME IN 1 μs 7802 SYSTEM
Set loop count=5	Once	LDBI		2 05	2 μs
Pulse output line once	Five times	CLA STAS LDAI STAS	 PORT 00 80 PORT 00	2 4 2 4	2 μs 4 μs 2 μs 4 μs
Test for end	Five times	DCB JPR	 NE LOOP	2 4	2 μs 4 μs
Leave output line low	Once	CLA STAS	 PORT 00	2 4	2 μs 4 μs

Figure 5-4. Sample Timing Calculation.

The total execution time for the functions performed once, outside the loop, is

$$2 + 2 + 4 = 8\ \mu s$$

One pass through the loop requires:

$$2 + 4 + 2 + 4 + 2 + 4 = 18\ \mu s$$

Five loop iterations requires (5) (18) = 90 μs

The total time for the program segment (instructions inside and outside the loop) is

$$90 + 8 = 98\ \mu s$$

The period of the pulses is found by adding the time the pulse is low to the time the pulse is high. The pulse is low from the end of the first output instruction to the end of the second ($2 + 4 = 6\ \mu s$). The pulse is high from the end of the second output instruction to the end of the first (or third) output instruction as the program flows around (or exits) the loop ($2 + 4 + 2 + 4 = 12\ \mu s$). The period of the pulses is therefore

$$6 + 12 = 18\ \mu s$$

SECTION 6
Memory and I/O Mapping and Control

The 6800 is a memory-oriented processor. It accesses 65,536 bytes of memory via its 16-bit address bus. I/O is memory-mapped; there are no I/O control signals generated by the 6800 (no input/output instructions). I/O uses the same address, data, and control signals as memory.

The 7802 decodes address and control signals from the 6800 to generate separate memory and I/O control signals for the STD BUS. This allows the 7802 to interface directly with other Series 7000 memory and I/O cards. Appendix A explains the how decoding of the addresses is accomplished.

Memory Addressing

A specific memory location is addressed when these conditions are met:

a. The Address Bus contains the specific address of the memory location (0000 through FFFF hexadecimal);

b. MEMRQ* (Memory Request) and RD* (Read) or WR* (Write) control signals are active;

c. MEMEX (Memory Expansion) has the polarity required by the specific memory card.

The 7802 disconnects from the STD BUS and enters a HALT state while Direct Memory Access (DMA) operations are conducted by an alternate system controller card. DMA is controlled by the BUSRQ*/BUSAK* (Bus Request/Bus Acknowledge) signals.

A typical memory implementation is shown in Figure 6-1.

The 7802 card has a combined EPROM/ROM and RAM memory on the card which is large enough to store the program and variable data required in many applications, without the need for additional external memory cards. The card is shipped with 1K of RAM and sockets which allow the user to add up to 8K of EPROM or masked ROM devices and to expand the RAM to 4K. The onboard memory sockets have addressing restrictions (Figure A-2) and are not accessible in DMA operations.

The onboard memory is organized as follows:

a. **EPROM/ROM sockets:** provide capacity for four 2716 or equivalent single +5V supply EPROM devices which can be mixed in any combination with 2316E or equivalent masked ROMs. Each device is a 2048-byte (2K) read-only memory for a total capacity of 8192 (8K) bytes. All of these devices are supplied by the user.

b. **RAM and RAM Sockets:** provide two 2114L or equivalent RAM devices organized as a 1024-byte (1K) memory, and sockets for six additional user-supplied 2114 RAMs. The 2114 is a 1024x4 device and two chips are required for each 1K of RAM added to the card. The total RAM capacity of the 7802 with all sockets loaded is 4096 (4K) bytes.

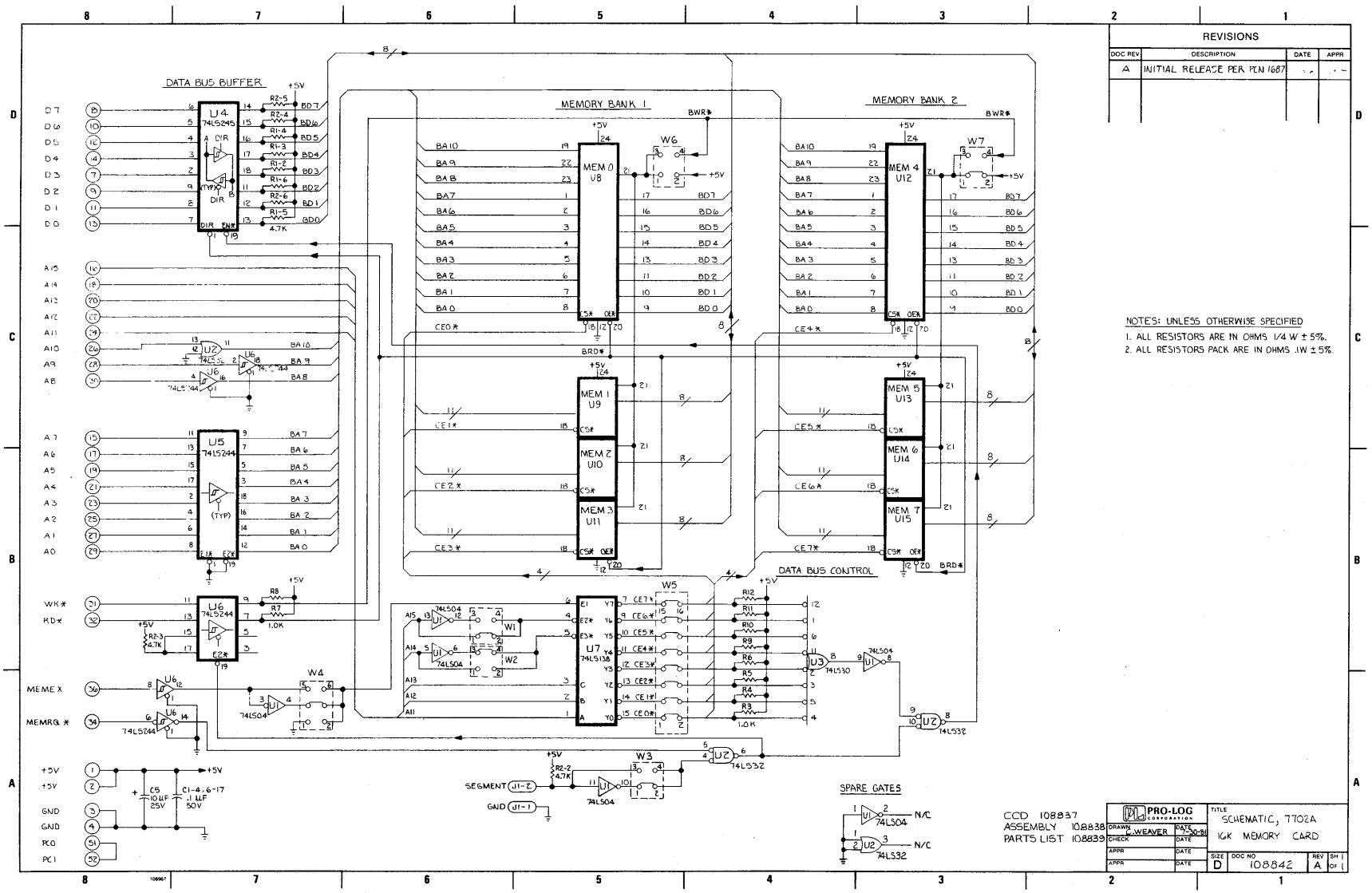


Figure 6-1. Typical External Memory Implementation (7702 EPROM Card).

Input/Output (I/O) Port Addressing

The 7802 can address up to 256 each input ports and output ports. The port address appears on the low-order half of the Address Bus (A0-A7). The high-order half of the Address Bus (A8-A15) is decoded by the 7802 to generate an I/O Request signal for one dedicated page of memory. A specific I/O port is addressed when the following conditions are met:

a. The Address Bus (A0-A7) contains the specific address of the I/O port (00 through FF hexadecimal);

b. IORQ* (I/O Request) is active;

c. IOEXP (I/O Expansion) has the polarity required by the specific I/O card;

d. RD* (Read) is active to select an input port, or WR* (Write) is active to select an output port.

The 8-bit input ports provide a means for reading data or status lines into the processor to take part in programmed operations. The 8-bit output ports provide a means for outputting program-generated data or control states. Typical input and output port circuits are shown in Figure 6-2.

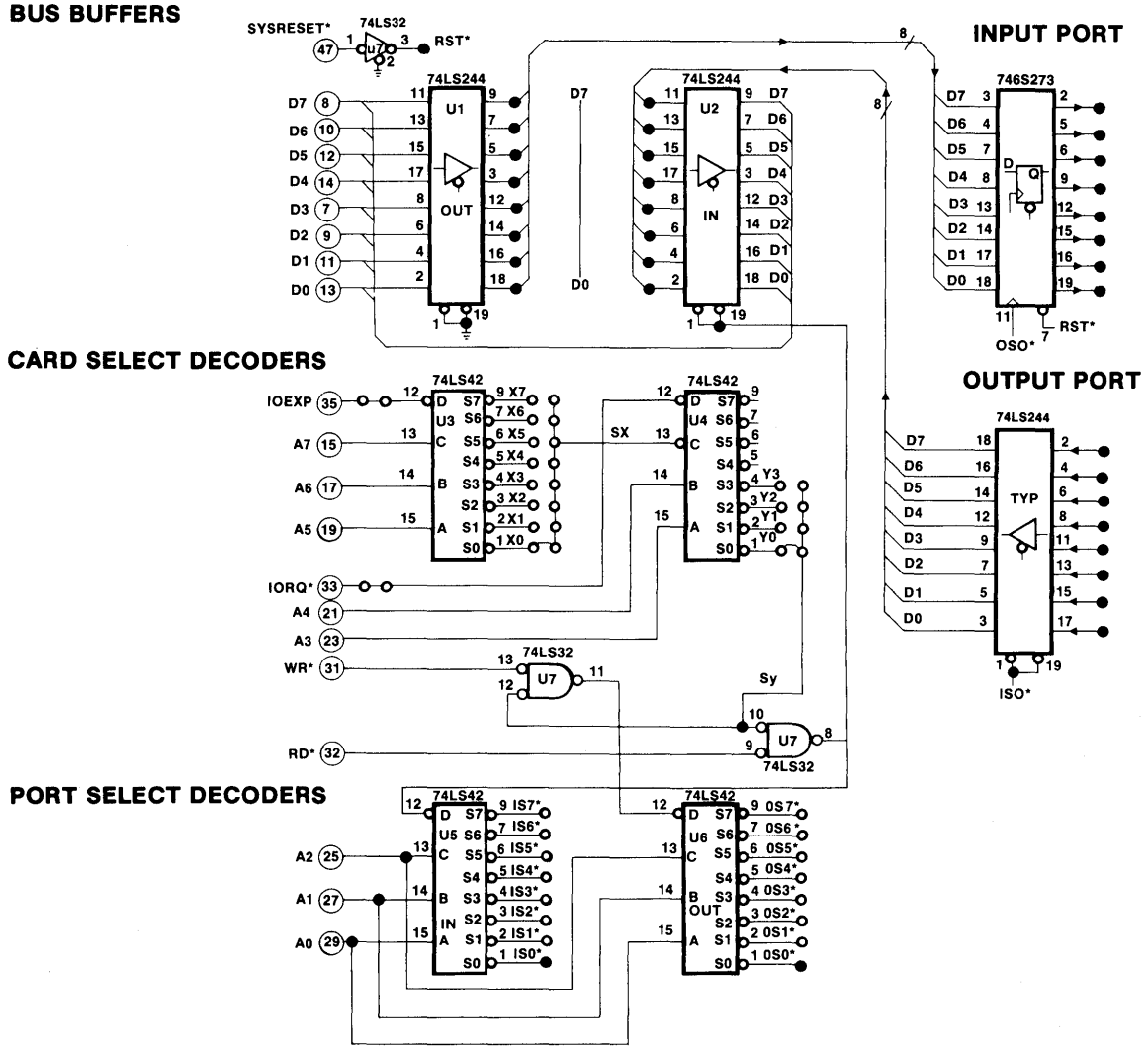


Figure 6-2. Typical Input & Output Port Implementation.

This figure illustrates the STD BUS Interface, I/O port address decoding circuitry, and device types typically used to implement I/O ports. Pro-Log's 7500, 7600, and 7900 Series I/O modules are similar to this example.

SECTION 7
Program and Hardware Debugging

Microprocessor Logic State Analysis

An attempt at monitoring the execution of a microprocessor program in real time using a conventional multi-trace oscilloscope will be found to be impossible for practical purposes. The capacity of the scope and the operator will be quickly exhausted by the magnitude of the problem because of the following characteristics:

a. **Parallel Data and Addresses.** Data is transferred as byte-parallel information (the address bus is 2 bytes wide). Individual bits on these buses have little meaning in program debugging. It is necessary to see the full content of both buses at once.

b. **Display Trigger Qualification.** As many as 20 signals (combined address and control signals) may be used simultaneously to qualify the enabling of a peripheral memory card, for example. In order to capture this event, the test instrumentation must also be trigger-qualified by the same group of signals. Conventional oscilloscopes lack the number of trigger channels and operating modes needed to interface with a processor system such as the 7802.

c. **Data Bus Voltage Levels and Timing.** The 7802 and all of its peripheral cards in a given system will drive the Data Bus at different times, and will do so with a variety of logic high and logic low levels, all of which are different but within specification. This presents two problems: the operator will find it difficult to identify the source of any given waveform on the scope display; and in order to see a specific data segment on the Data Bus, the operator will find it necessary to synchronize the display with the processor's **software program** rather than with the voltage output of any one element of system hardware. The logic state analyzer solves these problems by displaying formatted high/low logic states rather than analog waveforms, and by offering enough trigger channels and coincidence logic to allow literal program/display synchronization.

A microprocessor system analyzer or logic state analyzer is considered an essential troubleshooting aid for both program development and system maintenance in any 7802-based system where the needs of the Manufacturing Test and Field Service organizations are important considerations.

The analyzer performs these basic functions:

a. Tracks the actual instruction sequence as the program executes, facilitating program debugging.

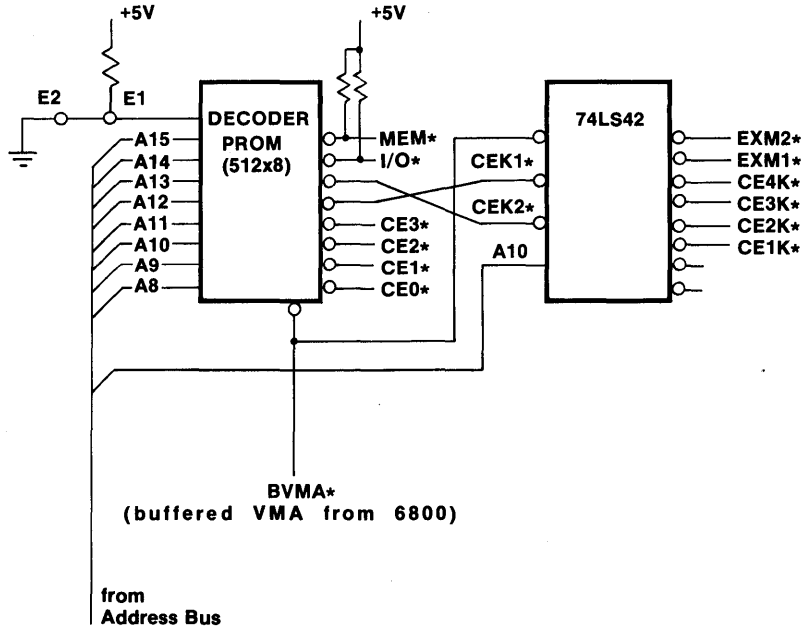
b. Monitors control states and data passing between the processor and the system it controls, allowing the system external to the processor card to be observed at the same time as the program flow, using the same display.

c. Provides a multi-qualified trigger to a conventional oscilloscope when analog measurements are unavoidable (e.g. propagation delay through a suspected memory device).

APPENDIX A
7802 Address Decoding

The 7802 uses a 512x8 PROM to decode the 6800's address bus. Chip selects, memory control signals, and/or I/O control signals are generated from the decoder PROM depending upon the upper half (A8-A15) of the address bus. This decoding scheme allows the 6800 to interface to I/O-mapped I/O rather than just memory-mapped I/O; thus, the 7802 can interface directly with the Series 7000 I/O cards.

The decoding circuit is shown in Figure A-1. The eight most significant bits of the address bus along with a most significant input or GND or +5V defines a memory location in the PROM. The data read out from that location defines a chip select for one of the onboard ROMs (CE0*, CE1*, CE2*, CE3*) and/or some control signals which are further decoded. Two of the signals (CEK1* and CEK2*) are used to generate a chip select for onboard RAM (CE1K*, CE2K*, CE3K*, CE4K*) or an external memory control signal which is used to enable input buffers for the STD BUS data bus. MEM* and I/O* are buffered to generate the STD BUS signals MEMRQ* and IORQ*, respectively.



	DECODE SIGNALS	SYSTEM SIGNALS
Onboard PROM	CE3* CE2* CE1* CE0*	Chip enables for PROMs
Onboard RAM	CE4K* CE3K* CE2K* CE1K*	Chip enables for RAMs
External Memory	EXM2* EXM1* MEM*	Enables for STD BUS Data Bus input buffers MEMRQ*
External I/O	I/O*	IORQ*

Figure A-1. 7802 Decode Scheme.

The decoder PROM allows for two system mapping schemes. Figure A-2 depicts the possible system configurations. In the as-shipped configuration, I/O is at page 00. "Short" addressing then allows for faster input/output operations. To select the optional mapping scheme, simply cut the trace between E1 and E2 (grid BC-6 on the Assembly Drawing). The option allows page 00 to be used as RAM which is a common memory configuration for 6800 system. "Short" addressing allows for fast data manipulation.

MEMORY DEVICE DESIGNATION	ADDRESS AS SHIPPED	ADDRESS OPTION
ROM 0 ROM 1 ROM 2 ROM 3	F800-FFFF F000-F7FF E800-EFFF E000-E7FF	F800-FFFF F000-F7FF E800-EFFF E000-E7FF
RAM 1K U19 U23 RAM 2K U18 U22 RAM 3K U17 U21 RAM 4K U16 U20	D000-D3FF D400-D7FF D800-DBFF DC00-DFFF	0000-03FF 0400-07FF 0800-0BFF 0CFF-0FFF
External I/O	0000-00FF	8000-80FF
External Memory	0100-CFFF	1000-7FFF 8100-DFFF

Figure A-2. 7802 System Mapping.

If the as-shipped and optional mapping schemes do not meet the 7802 user's needs, he may change the system mapping scheme by changing the contents of the decoder PROM.

Figure A-3 gives the programming information for the decoder PROM used by the 7802 for address decoding. A "0" represents an active condition, a "1" is inactive. The desired signals are selected by appropriate setting and clearing of data bits in the PROM's memory location. For example, in order to access memory location FFFE which is in PROM 0, a chip select and memory control signal need to be generated. At location OFF in the decoder PROM, there should be a data word which will enable the ROM 0 chip. In this case, the data word is Hex 4E (see Decoder PROM listing) which activates the CE0*, CEK1*, CEK2*, MEM* signals thus selecting ROM 0 and generating MEMRQ*.

FUNCTION	PROM OUTPUT BIT								HEX
	7	6	5	4	3	2	1	0	
	MEM*	I/O*	CEK2*	CEK1*	CE3*	CE2*	CE1*	CE0*	
Enable PROM 0	0	1	0	0	1	1	1	0	4E
Enable PROM 1	0	1	0	0	1	1	0	1	4D
Enable PROM 2	0	1	0	0	1	0	1	1	4B
Enable PROM 3	0	1	0	0	0	1	1	1	47
Enable RAM, lower 2K	0	1	1	0	1	1	1	1	6F
Enalbe RAM, upper 2K	0	1	0	1	1	1	1	1	5F
Enable external memory	0	1	1	1	1	1	1	1	7F
Enable external I/O	1	0	1	1	1	1	1	1	BF

NOTES: (1) "Lower 2K" refers to RAM chips U19, U23, U18, and U22.
(2) "Upper 2K" refers to RAM chips U17, U21, U16, and U20.

Figure A-3. Decoder PROM Programming Information.

PRO-LOG CORPORATION

PROGRAM ASSEMBLY FORM

TITLE 7802 DECODER PROM DATE

HEXADECIMAL PAGE ADR	HEXADECIMAL LINE ADR	HEXADECIMAL INSTR.	MNEMONIC LABEL	MNEMONIC INSTR.	MNEMONIC MODIFIER	COMMENTS
0	00	BF	I/O		I/O*	GENERATE IORQ*
	1	7F	EXT. MEM		MEM*	GENERATE MEMRQ* (EXTERNAL MEMORY)
	2					
	3					
	4					
	5					
	6					
	7					
	8					
	9					
	A					
	B					
	C					
	D					
	E					
	F					
	10					
	1					
	2					
	3					
	4					
	5					
	6					
	7					
	8					
	9					
	A					
	B					
	C					
	D					
	E					
	F					
	20					
	1					
	2					
	3					
	4					
	5					
	6					
	7					
	8					
	9					
	A					
	B					
	C					
	D					
	E					
	F					
	30					
	1					
	2					
	3					
	4					
	5					
	6					
	7					
	8					
	9					
	A					
	B					
	C					
	D					
	E					
	F					

HEXADECIMAL PAGE ADR	HEXADECIMAL LINE ADR	HEXADECIMAL INSTR.	MNEMONIC LABEL	MNEMONIC INSTR.	MNEMONIC MODIFIER	TITLE / DATE / COMMENTS
0	40	7F	EXT. MEM		MEM*	GENERATE MEMRQ* (EXTERNAL MEMORY)
	41					
	42					
	43					
	44					
	45					
	46					
	47					
	48					
	49					
	4A					
	4B					
	4C					
	4D					
	4E					
	4F					
	50					
	51					
	52					
	53					
	54					
	55					
	56					
	57					
	58					
	59					
	5A					
	5B					
	5C					
	5D					
	5E					
	5F					
	60					
	61					
	62					
	63					
	64					
	65					
	66					
	67					
	68					
	69					
	6A					
	6B					
	6C					
	6D					
	6E					
	6F					
	70					
	71					
	72					
	73					
	74					
	75					
	76					
	77					
	78					
	79					
	7A					
	7B					
	7C					
	7D					
	7E					
	7F					

HEXADECIMAL PAGE ADR	HEXADECIMAL LINE ADR	HEXADECIMAL INSTR.	MNEMONIC LABEL	MNEMONIC INSTR.	MNEMONIC MODIFIER	TITLE / DATE / COMMENTS
0	80	7F	EXT.MEM		MEM*	GENERATE MEMRQ* (EXTERNAL MEMORY)
	81					
	82					
	83					
	84					
	85					
	86					
	87					
	88					
	89					
	8A					
	8B					
	8C					
	8D					
	8E					
	8F					
	90					
	91					
	92					
	93					
	94					
	95					
	96					
	97					
	98					
	99					
	9A					
	9B					
	9C					
	9D					
	9E					
	9F					
	A0					
	A1					
	A2					
	A3					
	A4					
	A5					
	A6					
	A7					
	A8					
	A9					
	AA					
	AB					
	AC					
	AD					
	AE					
	AF					
	B0					
	B1					
	B2					
	B3					
	B4					
	B5					
	B6					
	B7					
	B8					
	B9					
	BA					
	BB					
	BC					
	BD					
	BE					
	BF					

HEXADECIMAL PAGE ADR	HEXADECIMAL LINE ADR	HEXADECIMAL INSTR.	MNEMONIC LABEL	MNEMONIC INSTR.	MNEMONIC MODIFIER	TITLE / COMMENTS	DATE
0	C0	7F	EXT. MEM		MEM*	GENERATE MEMRQ* (EXTERNAL MEMORY)	
	1						
	2						
	3						
	4						
	5						
	6						
	7						
	8						
	9						
	A						
	B						
	C						
	D						
	E						
	F						
	D0	6F	RAM 1K		CE1*·MEM*	GENERATE MEMRQ* AND ENABLE RAM U23,U19 D000-D3FF	
	1						
	2						
	3						
	D4		RAM 2K			GENERATE MEMRQ* AND ENABLE RAM U22,U18 D400-D7FF	
	5						
	6						
	7						
	D8	5F	RAM 3K		CE2*·MEM*	GENERATE MEMRQ* AND ENABLE RAM U21,U17 D800-DBFF	
	9						
	A						
	B						
	DC		RAM 4K			GENERATE MEMRQ* AND ENABLE RAM U20,U16 DC00-DFFF	
	D						
	E						
	F						
	E0	47	PROM 3	CE3*·CE1*·CE2*·MEM*		GENERATE MEMRQ* AND ENABLE ROM 3 E000-E7FF	
	1						
	2						
	3						
	4						
	5						
	6						
	7						
	E8	4B	PROM 2	CE2*·CE1*·CE2*·MEM*		GENERATE MEMRQ* AND ENABLE ROM 2 E800-EFFF	
	9						
	A						
	B						
	C						
	D						
	E						
	F						
	F0	4D	PROM 1	CE1*·CE1*·CE2*·MEM*		GENERATE MEMRQ* AND ENABLE ROM 1 F000-F7FF	
	1						
	2						
	3						
	4						
	5						
	6						
	7						
	F8	4E	PROM 0	CE0*·CE1*·CE2*·MEM*		GENERATE MEMRQ* AND ENABLE ROM 0 F800-FFFF	
	9						
	A						
	B						
	C						
	D						
	E						
	F						

HEXADECIMAL PAGE ADR	HEXADECIMAL LINE ADR	INSTR.	MNEMONIC LABEL	MNEMONIC INSTR.	MNEMONIC MODIFIER	TITLE / COMMENTS
1	00	6F	RAM 1K		CEK1*. MEM*	GENERATE MEMRQ* AND ENABLE RAM U23, U19 0000-03FF
	01					
	02					
	03					
	04		RAM 2K			GENERATE MEMRQ* AND ENABLE RAM U22, U18 0400-07FF
	05					
	06					
	07					
	08	5F	RAM 3K		CEK2*. MEM*	GENERATE MEMRQ* AND ENABLE RAM U21, U17 0800-0BFF
	09					
	0A					
	0B					
	0C		RAM 4K			GENERATE MEMRQ* AND ENABLE RAM U20, U16 0C00-0FFF
	0D					
	0E					
	0F					
	10	7F	EXT. MEM		MEM*	GENERATE MEMRQ* (EXTERNAL MEMORY)
	11					
	12					
	13					
	14					
	15					
	16					
	17					
	18					
	19					
	1A					
	1B					
	1C					
	1D					
	1E					
	1F					
	20					
	21					
	22					
	23					
	24					
	25					
	26					
	27					
	28					
	29					
	2A					
	2B					
	2C					
	2D					
	2E					
	2F					
	30					
	31					
	32					
	33					
	34					
	35					
	36					
	37					
	38					
	39					
	3A					
	3B					
	3C					
	3D					
	3E					
	3F					

HEXADECIMAL PAGE ADR	HEXADECIMAL LINE ADR	HEXADECIMAL INSTR.	MNEMONIC LABEL	MNEMONIC INSTR.	MNEMONIC MODIFIER	TITLE / COMMENTS
1	40	7F	EXT. MEM		MEM*	GENERATE MEMRQ* (EXTERNAL MEMORY)
	41					
	42					
	43					
	44					
	45					
	46					
	47					
	48					
	49					
	4A					
	4B					
	4C					
	4D					
	4E					
	4F					
	50					
	51					
	52					
	53					
	54					
	55					
	56					
	57					
	58					
	59					
	5A					
	5B					
	5C					
	5D					
	5E					
	5F					
	60					
	61					
	62					
	63					
	64					
	65					
	66					
	67					
	68					
	69					
	6A					
	6B					
	6C					
	6D					
	6E					
	6F					
	70					
	71					
	72					
	73					
	74					
	75					
	76					
	77					
	78					
	79					
	7A					
	7B					
	7C					
	7D					
	7E					
	7F					

HEXADECIMAL PAGE ADR	HEXADECIMAL LINE ADR	HEXADECIMAL INSTR.	MNEMONIC LABEL	MNEMONIC INSTR.	MNEMONIC MODIFIER	TITLE / COMMENTS
1	80	BF	I/O		I/O*	GENERATE IORQ*
	1	7F	EXT. MEM		MEM*	GENERATE MEMRQ* (EXTERNAL MEMORY)
	2					
	3					
	4					
	5					
	6					
	7					
	8					
	9					
	A					
	B					
	C					
	D					
	E					
	F					
	90					
	1					
	2					
	3					
	4					
	5					
	6					
	7					
	8					
	9					
	A					
	B					
	C					
	D					
	E					
	F					
	A0					
	1					
	2					
	3					
	4					
	5					
	6					
	7					
	8					
	9					
	A					
	B					
	C					
	D					
	E					
	F					
	B0					
	1					
	2					
	3					
	4					
	5					
	6					
	7					
	8					
	9					
	A					
	B					
	C					
	D					
	E					
	F					

HEXADECIMAL PAGE ADR	HEXADECIMAL LINE ADR	INSTR.	MNEMONIC LABEL	MNEMONIC INSTR.	MNEMONIC MODIFIER	TITLE / COMMENTS
1	C0	7F	EXT. MEM		MEM*	GENERATE MEMRQ* (EXTERNAL MEMORY).
	1					
	2					
	3					
	4					
	5					
	6					
	7					
	8					
	9					
	A					
	B					
	C					
	D					
	E					
	F					
	D0					
	1					
	2					
	3					
	4					
	5					
	6					
	7					
	8					
	9					
	A					
	B					
	C					
	D					
	E					
	F					
	E0	47	PROM 3	CE3*.CEK1*.CEK2*.MEM*		GENERATE MEMRQ* AND ENABLE ROM 3 E000-E7FF
	1					
	2					
	3					
	4					
	5					
	6					
	7					
	E8	4B	PROM 2	CE2*.CEK1*.CEK2*.MEM*		GENERATE MEMRQ* AND ENABLE ROM 2 E800-EFFF
	9					
	A					
	B					
	C					
	D					
	E					
	F					
	F0	4D	PROM 1	CE1*.CEK1*.CEK2*.MEM*		GENERATE MEMRQ* AND ENABLE ROM 1 F000-F7FF
	1					
	2					
	3					
	4					
	5					
	6					
	7					
	F8	4E	PROM 0	CE0*.CEK1*.CEK2*.MEM*		GENERATE MEMRQ* AND ENABLE ROM 0 F800-FFFF
	9					
	A					
	B					
	C					
	D					
	E					
	F					

7802 Decoder PROM Specifications

The decoder PROM used on the 7802 is a 512x8 low power Schottky TTL device. Figure A-5 gives the recommended parts and the Pro-Log Personality Modules needed to program each with a Pro-Log Series 90 PROM Programmer.

PART	MANUFACTURER	PERSONALITY MODULE
63LS481	MMI	PM9066, PA20-2, 512x8(L)
DM74LS472	National	PM9047, PA20-1, 512x8(L)
82S147	Signetics	PM9059, PA20-1, 512x8(L)

Figure A-5. Recommended Decoder PROM Parts.

If these parts cannot be found, the second recommendations are 512x8 Schottky TTL PROMs. Figure A-6 gives information concerning these parts.

PART	MANUFACTURER	PERSONALITY MODULE
63PS481	MMI	PM9066, PA20-2, 512x8(L)
63S481	MMI	PM9066, PA20-2, 512x8(L)
82S147	Signetics	PM9059, PA20-1, 512x8(L)
AM27S29C	AMD	PM9058, PA20-4, 512x8(L)
DM74S472	National	PM9047, PA20-1, 512x8(L)
TBP18S42N	T.I.	PM9046B, PA20-1, 512x8(L)

Figure A-6. Optional Decoder PROM Parts.

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.
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 Rd.,
   Monterey, Calif. 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 two years 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.

USER'S MANUAL

PRO-LOG
CORPORATION

2411 Garden Road
Monterey, California 93940
Telephone: (408) 372-4593
TWX: 910-360-7082
Telex: 171879

108793A 3.5K 2/82