MTU-130

MONOMEG SINGLE BOARD COMPUTER

HARDWARE MANUAL

FEBRUARY, 1982 REV. A

TABLE OF CONTENTS

1. FEATURE SUMMARY

The Monomeg Processor Board is the primary component of the MTU-130 Computer System. It is truly a single-board computer requiring only a keyboard matrix, a video monitor, and mass storage to form a fully functional computer system.

1.1 6502 MICROPROCESSOR

The Monomeg Processor Board uses the powerful and popular 6502 microprocessor as its CPU. Processor and system bus clock speed is 1.0MHz and there are no wait states introduced by any system components. Since the 6502's CPU cycle frequency is the same as its clock frequency, that 1.0MHz is equivalent to a 3MHz 8080 or Z-80 CPU even before the 6502's advanced instruction set is taken into account.

1.2 18 BIT EXTENDED ADDRESSING

The Monomeg supports an extended 18 bit address bus which allows direct addressing of 256K of memory without need for special bank-switched memory boards. Address space expansion is based on the concept of separation of instructions and data which allows a program as large as 64K to manipulate 64K of data without constantly referencing bank selection registers. Even larger programs and/or data spaces can be accomodated if the bank selection registers are manipulated.

1.3 READ/WRITE MEMORY

The Monomeg has 48K of RAM for general system use addressed from 00000 to 0BFFF and 16K of RAM associated with the display addressed from 1C000-1FFFF for an on-board total of 64K. When used in the MTU-130 system, the Disk Controller adds another 16K from 0C000-0FFFF for a system total of 80K. A software controlled write protect feature is available for a portion of on-board RAM from 08000-0BFFF. Although dynamic RAM ICs are used, the display section keeps all on-board RAM refreshed without introducing wait states or other types of system slowdowns.

1.4 ROM/PROM SOCKETS

On-board the Monomeg Processor Board are four sockets which accept standard 32K bit (4K byte) ROMs and PROMs. These are fixed addressed from 0C000-0FFFF and are provided for custom applications of the Monomeg without mass storage. These sockets are typically unused in the MTU-130 system but are available for use. A programmable I/O bit and on-board jumper determine whether the ROM sockets or the external bus responds to addresses in the range of 0C000-0FFFF.

1.5 DISPLAY GENERATOR

The on-board display generator operates in one of two modes. The high resolution black and white mode produces a 480 wide by 256 high bit mapped dot matrix. The gray scale mode produces a 240 wide by 256 high dot matrix where 2 bits define one of 4 possible shades of gray for each dot. Either mode uses 15K of on-board memory fixed addressed from 1C000-1FBFF. With suitable character generation software, the high resolution mode is capable of 25 lines of 80 characters per line. The display may also be blanked under software control without affecting the contents of display memory. Video output is standard composite video suitable for a standard video monitor.

1.6 LIGHT PEN INTERFACE

Special registers on the Monomeg will latch the display memory address and pixel number within the currently addressed byte in response to an input pulse. Typically this pulse would come from an external high speed photodiode/amplifier based light pen. The information provided by these registers can be easily converted into X and Y coordinates to within 1 pixel resolution with software. This light pen capability gives a digitizing speed of 60 points per second which is suitable for real-time drawing and tracking as well as the usual menu selection applications.

1.7 KEYBOARD INTERFACE

The Monomeg is designed to interface with a nearly bare key matrix of up to 96 keys arranged in a 6 by 16 array. Other than a crosspoint matrix, the external keyboard circuitry need only have a 4 bit resettable counter (type 74LS93) and a 1-of-16 decoder (type 74159) to select the columns. A software routine is responsible for scanning the key matrix and implementing rollover, repeat, shift, and other features. In addition to the standard matrix, interface for 4 normally open individual keys is provided for break, interrupt, and reset console functions.

1.8 AUDIO D-A CONVERTER

An 8 bit digital-to-analog converter coupled to a sharp low-pass filter and 1 watt power amplifier provides the primary sound generation capability of the Monomeg. With the DAC and appropriate driver software, signal tones, high quality multi-voice music, and speech sounds may be generated. The CB2 output of one of the parallel interface chips is also tied into the audio system for simple rectangular waveform generation without processor intervention.

1.9 CASSETTE INTERFACE

An audio cassette interface is provided for software and data exchange with existing microcomputer systems. The write circuit consists of a flip-flop that can be toggled under software control followed by an attenuator and waveshaping filter. The read circuit consists of noise filter followed by a zero crossing detector. With appropriate programming, it is possible to read and write nearly all audio cassette formats in common use.

1.10 SERIAL PORT

A type 6551 serial I/O chip provides serial RS-232 compatible I/O capability. The baud rate is programmable from 50 to 19,200 baud and the data format is programmable from 5 to 8 data bits, 1, 1.5, and 2 stop bits, and parity. A number of modem control signals are implemented as well. Level shifters provide a standard + and - signal RS-232 interface.

1.11 PARALLEL PORT

A type 6522 parallel I/O chip is reserved for the user. The parallel port connector is simply connected to the 20 I/O pins on that 6522. Additionally, regulated DC voltages of +12, +5, and -12 volts are provided to power customer interfaces and the system Reset signal is provided.

1.12 RESERVED ON-BOARD I/O

Several additional bits of parallel I/O terminate in two connectors on the Monomeg. These are intended to connect to two optional off-board (but still inside the MTU-130 case) driver boards for special features. One of these is a Network Transceiver board that allows a very inexpensive implementation of a 50K baud inter-processor notwork. The other is a miniature cartridge tape drive. Both connectors have regulated power as well as the I/O bits. The I/O and power are available to the customer if the corresponding driver board is not being used.

1.13 EXPANSION BUS

A set of 44 edge fingers forms the system bus connector. The 8 data lines, 18 address lines, bus control signals, and power connections to the Monomeg board are made through these edge fingers. The external bus is used for additional memory and I/O functions in the MTU-130 system. The physical dimensions and pin assignments of the bus connector are designed to be compatible with the KIM-1 bus structure.

1.14 POWER CONSUMPTION

The MONOMEG CPU is designed for low power consumption. Current requirements are as follows:

+5 Regulated at 1.2 amps
+12 Regulated at .5 amps, .75 amps at full audio volume
-12 Regulated at .1 amps

2. INTEGRATION INTO THE MTU-130 COMPUTER

The Monomeg Single Board Computer is housed in the Keyboard Module. It resides in the bottom slot of the 5 slot card file inside the module case. The bottom slot position is required because the Monomeg is considerably deeper (10.5 inches) than the Disk Controller and optional expansion boards.

2.1 CONNECTORS

Connection to the system bus, which goes to the other 4 slots, is with a 44 pin edge connector. The Monomeg also receives regulated DC power from the Keyboard Module power supply through the system bus.

Small separate connectors near the back edge of the Monomeg connect to various I/O devices within the Keyboard case and to the I/O connectors on the rear panel. These connectors are as follows:

A. Keyboard connector - A 16 pin single row pin header which provides power to the keyboard circuit board and connects to the keyswitch matrix and console control keys (INT, MOD, RESET, BREAK).

B. Light pen connector - A 4 pin single row pin header which provides power to the light pen amplifier board and receives the "hit" signal from it.

C. Parallel I/O connector - A 26 pin dual row pin header which connects through a flat cable to the parallel I/O connector on the rear panel.

D. Serial I/O connector - A 10 pin single row pin header which connects through a cable to the serial I/O connector on the rear panel.

E. Miscellaneous I/O connector - A 10 pin single row pin header which connects to the internal speaker, cassette I/O jacks, and video jack on the rear panel. It also provides separated video and sync signals which are not connected to anything.

F. MTUNET connector - A 10 pin single row pin header which provides power and receive/transmit signals to an optional MUNET Transceiver board which mounts inside the keyboard case if present.

G. MTUTAPE connector - A 14 pin single row pin header which provides power, data, and control signals to an optional miniature tape cartridge drive. This connector can be used as a third user I/O port if the tape drive is not used.

2.2 POWER SUPPLY

The MTU-130 power supply provides regulated DC power (+12, +5, and -12) to the Monomeg and unregulated DC power to other boards in the system. The Monomeg receives its DC power from the system bus connector.

3. PIN CONNECTIONS AND JUMPER OPTIONS

The following sections detail the pin assignments of all connectors on the Monomeg CPU board.

3.1 SYSTEM BUS CONNECTOR

The system bus connector is a 22 position double-sided (44 pins total) PC edge connector. Pins 1, 22, A, and Z are identified in etch.

BUS PIN	SIGNAL	
1	I/O SELECT	Logic 1 when 0BE00-0BFFF and I/O is enabled.
2	ADDRESS BUS 16	Bank Select least significant bit.
3	ADDRESS BUS 17	Bank Select most significant bit.
4	$\overline{\text{IRQ}}$	Maskable interrupt request, has 3K pullup.
5	$\overline{\text{SET}}$ OVERFLOW	Input to pin 38 of 6502, has 3K pullup.
6	$\overline{\text{NMI}}$	Non-maskable interrupt request, has 3K pullup.
7	$\overline{\text{RESET}}$	Logic zero during power up, cold, or warm reset
8	BUFFERED DATA BUS BIT 7	
9	BUFFERED DATA BUS BIT 6	
10	BUFFERED DATA BUS BIT 5	
11	BUFFERED DATA BUS BIT 4	
12	BUFFERED DATA BUS BIT 3	
13	BUFFERED DATA BUS BIT 2	
14	BUFFERED DATA BUS BIT 1	
15	BUFFERED DATA BUS BIT 0	
16	-12 VOLTS REGULATED	Regulated power input
17	+12 VOLTS REGULATED	Regulated power input
18	--NC--	(bus carries +8 volts unregulated on this pin)
19	--NC--	
20	--NC--	
21	+5 VOLTS REGULATED	Regulated power input
22	GROUND	
A	BUFFERED ADDRESS BUS 0	
B	BUFFERED ADDRESS BUS 1	
C	BUFFERED ADDRESS BUS 2	
D	BUFFERED ADDRESS BUS 3	
E	BUFFERED ADDRESS BUS 4	
F	BUFFERED ADDRESS BUS 5	
H	BUFFERED ADDRESS BUS 6	
J	BUFFERED ADDRESS BUS 7	
K	BUFFERED ADDRESS BUS 8	
L	BUFFERED ADDRESS BUS 9	
M	BUFFERED ADDRESS BUS 10	
N	BUFFERED ADDRESS BUS 11	
P	BUFFERED ADDRESS BUS 12	
R	BUFFERED ADDRESS BUS 13	
S	BUFFERED ADDRESS BUS 14	
T	BUFFERED ADDRESS BUS 15	
U	PHASE 2	Phase 2 clock for bus data transfers, 1.0MHz
V	$\overline{\text{READ/WRITE}}$	Logic high during bus read cycles
W	READ/WRITE	Logic low during bus read cycles
X	--NC--	(bus carries +16 volts unregulated on this pin)
Y	$\overline{\text{PHASE 2}}$	Inverse of Phase 2
Z	$\overline{\text{RAM WRITE}}$	Logical NAND of $\overline{\text{READ/WRITE}}$ and PHASE 2

3.2 KEYBOARD CONNECTOR

The keyboard connector is designated J3. It is a polarized dual row pin header with pin 1 identified in printed circuit etch.

PINS	SIGNALS
1	+5V. regulated power
2	Ground
3	--polarize--
4	--unused--
5	Column count clear
6	Interrupt key
7	Column count clock
8	Modify key

PINS	SIGNALS
9	Break key
10	Reset key
11	Matrix row 1
12	Matrix row 4
13	Matrix row 3
14	Matrix row 6
15	Matrix row 5
16	Matrix row 2

3.3 LIGHT PEN CONNECTOR

The light pen connector is designated J7. It is a polarized single row pin header with pin 1 identified in printed circuit etch.

PINS	SIGNALS
1	+12 V. regulated power
2	Hit sense
3	--unused--
4	Ground

3.4 PARALLEL I/O CONNECTOR

The parallel I/O connector is designated J5. It is a double row pin header with odd numbered pins in one row and even numbered pins in the other row. Pin 1 is identified in printed circuit etch.

PINS	SIGNALS
1	+5 Volts regulated
2	GROUND
3	CB2 SOUND OFF
4	+12 Volts regulated
5	RESET
6	-12 Volts regulated
7	CA2
8	CA1
9	PA1
10	PA0
11	PA3
12	PA2
13	PA5

PINS	SIGNALS
14	PA4
15	PA7
16	PA6
17	PB1
18	PB0
19	PB3
20	PB2
21	PB5
22	PB4
23	PB7
24	PB6
25	CB2
26	CB1

3.5 SERIAL I/O CONNECTOR

The serial I/O connector is designated J4. It is a polarized single row pin header with pin 1 identified in printed circuit etch.

PINS	SIGNALS
10	GROUND
9	--unused--
8	--polarize--
7	Request To Send
6	Data Terminal Ready

PINS	SIGNALS
5	Transmit Data
4	Receive Data
3	Data Set Ready
2	Data Carrier Detect
1	Clear To Send

3.6 MISCELLANEOUS I/O CONNECTOR

The miscellaneous I/O connector is designated J6. It is a polarized single row pin header with pin 1 identified in printed circuit etch.

PINS	SIGNALS	PINS	SIGNALS
1	Cassette Output	6	Video Ground
2	Cassette Input	7	--polarize--
3	Cassette Ground	8	Composite Video
4	Sep Horiz Sync	9	Audio Ground
5	Sep Vert Sync	10	Audio Output

3.7 MTUNET I/O CONNECTOR

The MUNET I/O connector is designated J1. It is a single row pin header with pin 1 identified in printed circuit etch.

PINS	SIGNALS	PINS	SIGNALS
10	MTUNET Clock	5	--polarize--
9	MTUNET Data	4	Power Ground
8	MTUNET Active	3	+12 Power
7	MTUNET Talk	2	-12 Power
6	--unused--	1	+5 Power

3.8 MTUTAPE I/O CONNECTOR

The MTUTAPE I/O connector is designated J2. It is a single row pin header with pin 1 identified in printed circuit etch.

PINS	SIGNALS	PINS	SIGNALS
14	MTUTAPE Clock	7	MTUTAPE Data
13	MTUTAPE Go	6	MTUTAPE Signal Ground
12	MTUTAPE Write Enable	5	--polarize--
11	MTUTAPE Fast	4	Power Ground
10	MTUTAPE BOT	3	-12 Power
9	MTUTAPE Wrtie Protect	2	+12 Power
8	MTUTAPE Wafer in Place	1	+5 Power

3.9 JUMPER OPTIONS

There is only one jumper option on the Monomeg. This jumper determines whether Addresses between 0C000 and 0FFFF refer to the on-board ROM sockets or to the external bus (and presumably the Disk Controller) immediately after power-up or Reset. This is important because it determines where the reset/interrupt vectors for the 6502 microprocessor are located. The normal setting of this jumper is for external bus enabled and ROM sockets disabled which allows automatic boot from disk when power is applied. In specialized applications of the Monomeg without disk, this jumper would be put into the opposite position for power-up directly into the ROM.

The jumper is located 1" up from the back and 4.5" from right edge of the board (viewed with the edge fingers pointing away and the component side up). It consists of 4 wire-wrap posts arranged as two rows of two posts each. A small push-on jumper connects two of the posts together. The PC etch identifies one pair as "R" for ROM and the other pair as "B" for Bus. (Note: For owners of boards through Rev. C, these are misidentified.) The push-on jumper should be installed on the post pair corresponding to the desired operation.

MICROPROCESSOR PROGRAMMING

The microprocessor used in the Monomeg is a standard 6502 with peripheral logic added to support an 18 bit address bus. There will be no attempt here to explain the 6502's instruction set. The reader is referred to one of the following publications for a detailed description and explanation of the 6502 instruction set:

A. MOS Technology 6502 Programming Manual
B. Programming the 6502 (Third Ed.), by Rodney Zaks.
C. 6502 Assembly Language Programming, by Lance A. Leventhal.

4.1 6502 INSTRUCTION SET SUMMARY

Instructions		Immediate			Absolute			Zero Page			Accum.			Implied			(IND, X)			(IND), Y			Z, Page, X			ABS, X			ABS, Y			Relative			Indirect			Z, Page, Y			Condition Codes					
Mnemonic	Operation	OP	N	#	OP	N	#	OP	N	#	OP	N	#	OP	N	#	OP	N	#	OP	N	#	OP	N	#	OP	N	#	OP	N	#	OP	N	#	OP	N	#	OP	N	#	N	Z	C	I	D	V
A D C	A+M+C → A (4) (1)	69	2	2	6D	4	3	65	3	2							61	6	2	71	5	2	75	4	2	7D	4	3	79	4	3										✓	✓	✓			✓
A N D	A ∧ M → A (1)	29	2	2	2D	4	3	25	3	2							21	6	2	31	5	2	35	4	2	3D	4	3	39	4	3										✓	✓				
A S L	C ← 7 0 ← 0				0E	6	3	06	5	2	0A	2	1										16	6	2	1E	7	3													✓	✓	✓			
B C C	BRANCH ON C=0 (2)																															90	2	2											
B C S	BRANCH ON C=1 (2)																															B0	2	2											
B E Q	BRANCH ON Z=1 (2)																															F0	2	2											
B I T	A ∧ M				2C	4	3	24	3	2																															M7	✓				M6
B M I	BRANCH ON N=1 (2)																															30	2	2											
B N E	BRANCH ON Z=0 (2)																															D0	2	2											
B P L	BRANCH ON N=0 (2)																															10	2	2											
B R K	(See Fig. 1)													00	7	1																													
B V C	BRANCH ON V=0 (2)																															50	2	2											
B V S	BRANCH ON V=1 (2)																															70	2	2											
C L C	0 → C													18	2	1																											0			
C L D	0 → D													D8	2	1																												0		
C L I	0 → I													58	2	1																											0			
C L V	0 → V													B8	2	1																														0
C M P	A-M (1)	C9	2	2	CD	4	3	C5	3	2							C1	6	2	D1	5	2	D5	4	2	DD	4	3	D9	4	3										✓	✓	✓			
C P X	X-M	E0	2	2	EC	4	3	E4	3	2																																✓	✓	✓			
C P Y	Y-M	C0	2	2	CC	4	3	C4	3	2																																✓	✓	✓			
D E C	M-1 → M				CE	6	3	C6	5	2													D6	6	2	DE	7	3													✓	✓				
D E X	X-1 → X													CA																												✓	✓				
D E Y	Y-1 → Y													88																												✓	✓				
E O R	A ∨ M → A (1)	49	2	2	4D	4	3	45	3	2							41	6	2	51	5	2	55	4	2	5D	4	3	59	4	3										✓	✓				
I N C	M+1 → M				EE	6	3	E6	5	2													F6	6	2	FE	7	3													✓	✓				
I N X	X+1 → X													E8	2	1																											✓	✓				
I N Y	Y+1 → Y													C8	2	1																											✓	✓				
J M P	JUMP TO NEW LOC				4C	3	3																												6C	5	3								
J S R	(See Fig. 2) JUMP SUB				20	6	3																																							
L D A	M → A (1)	A9	2	2	AD	4	3	A5	3	2							A1	6	2	B1	5	2	B5	4	2	BD	4	3	B9	4	3										✓	✓				

Mnemonic	Operation	Immediate			Absolute			Zero Page			Accum.			Implied			(IND, X)			(IND), Y			Z, Page, X			ABS, X			ABS, Y			Relative			Indirect			Z, Page, Y			Condition Codes					
		OP	N	#	OP	N	#	OP	N	#	OP	N	#	OP	N	#	OP	N	#	OP	N	#	OP	N	#	OP	N	#	OP	N	#	OP	N	#	OP	N	#	OP	N	#	N	Z	C	I	D	V
L D X	M → X (1)	A2	2	2	AE	4	3	A6	3	2																			B9	4	3							B6	4	2	✓	✓				
L D Y	M → Y (1)	A0	2	2	AC	4	3	A4	3	2													B4	4	2	BC	4	3													✓	✓				
L S R	0 → 7 0 → C				4E	6	3	46	5	2	4A	2	1										56	6	2	5E	7	3													0	✓	✓			
N O P	NO OPERATION													EA	2	1																														
O R A	A ∨ M → A	09	2	2	0D	4	3	05	3	2							01	6	2	11	5	2	15	4	2	1D	4	3	19	4	3										✓	✓				
P H A	A → M_S S-1 → S													48	3	1																													
P H P	P → M_S S-1 → S													08	3	1																													
P L A	S+1 → S M_S → A													68	4	1																										✓	✓				
P L P	S+1 → S M_S → P													28	4	1																										(RESTORED)					
R O L	→ 7 0 → C →				2E	6	3	26	5	2	2A	2	1										36	6	2	3E	7	3													✓	✓	✓			
R O R	→ C 7 0 →				6E	6	3	66	5	2	6A	2	1										76	6	2	7E	7	3													✓	✓	✓			
R T I	(See Fig. 1) RTRN INT													40	6	1																										(RESTORED)					
R T S	(See Fig. 2) RTRN SUB													60	6	1																														
S B C	A-M-C → A (1)	E9	2	2	ED	4	3	E5	3	2							E1	6	2	F1	5	2	F5	4	2	FD	4	3	F9	4	3										✓	✓	(3)			✓
S E C	1 → C													38	2	1																												1			
S E D	1 → D													F8	2	1																													1		
S E I	1 → I													78	2	1																												1			
S T A	A → M				8D	4	3	85	3	2							81	6	2	91	6	2	95	4	2	9D	5	3	99	5	3														
S T X	X → M				8E	4	3	86	3	2																															96	4	2					
S T Y	Y → M				8C	4	3	84	3	2													94	4	2																					
T A X	A → X													AA	2	1																											✓	✓				
T A Y	A → Y													A8	2	1																											✓	✓				
T S X	S → X													BA	2	1																											✓	✓				
T X A	X → A													8A	2	1																											✓	✓				
T X S	X → S													9A	2	1																															
T Y A	Y → A													98	2	1																											✓	✓				

NOTES

1. Add 1 to "N" if Page Boundary is Crossed
2. Add 1 to "N" if Branch Occurs to Same Page
   Add 2 to "N" if Branch Occurs to Different Page
3. Carry Not = Borrow
4. If in Decimal Mode Z Flag is Invalid
   Accumulator Must be Checked for Zero Result

X Index X

Y Index Y

A Accumulator

M Memory Per Effective Address

M_S Memory Per Stack Pointer

\+ Add

− Subtract

∧ AND

∨ OR

∀ Exclusive OR

✓ Modified

\- Not Modified

M_7 Memory Bit 7

M_6 Memory Bit 6

N No Cycles

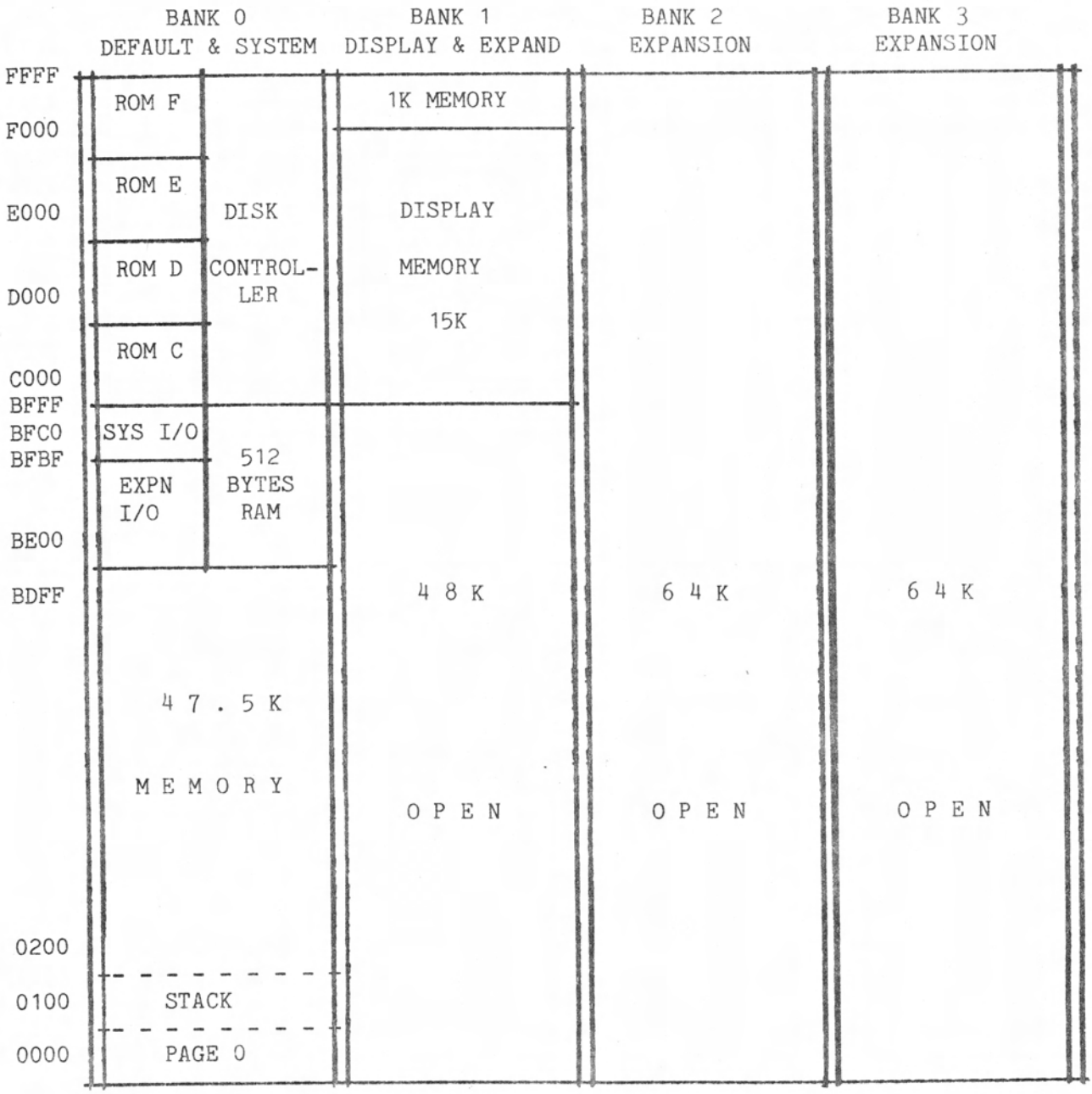
No Bytes

4.2 MEMORY MAP

Following is a reference summary of the disposition of all addresses in the MTU-130 system. Nearly all of these are fixed by logic design and cannot be easily changed. Generally speaking, all 64K addresses in bank 0 and the upper 16K of bank 1 are assigned to specific functions. The remaining 48K of bank 1 is unassigned while the 128K in banks 2 and 3 would typically be used by the 128K RAM/Math board.

4.2.1 Overall Memory Map

Below is a representation of the overall MTU-130 system memory map. Note that dual use may be made of addresses from BE00 to BFFF by manipulating the I-O/RAM control bit. Dual use may also be made of addresses C000 to FFFF in bank 0 by manipulating the ROM/BUS control bit. See section 4.3 for further information on the control bits.



```
00000
 ---     Page zero
000FF
00100
 ---     Stack
001FF
00200
 ---     General RAM
0BDFF
0BE00
 ---     Reserved for external bus I/O
0BFBF
0BFC0    Light Pen bit number and hit flag
0BFC1    Light Pen address low 8 bits
0BFC2    Light Pen address high 6 bits and PIX 8 flag
0BFC3    -Reserved-
0BFC4    Reset light pen hit flag
0BFC5    Reset keyboard scan counter
0BFC6    Toggle cassette output
0BFC7    -Reserved-
0BFC8    Read serial I/O received data, Write serial I/O transmit data
0BFC9    Read serial I/O status, Write serial reset
0BFCA    Read and write serial I/O command register (parity, modem cntl, etc.)
0BFCB    Read and write serial I/O control register (baud rate, serial format)
0BFCC
  --     BFC8-BFCB duplicated once
0BFCF
0BFD0    User 6522 Port B data register
0BFD1    User 6522 Port A data register
0BFD2    User 6522 Port B direction register
0BFD3    User 6522 Port A direction register
0BFD4    User 6522 timer 1
0BFD5    User 6522 timer 1
0BFD6    User 6522 timer 1
0BFD7    User 6522 timer 1
0BFD8    User 6522 timer 2
0BFD9    User 6522 timer 2
0BFDA    User 6522 shift register
0BFDB    User 6522 auxiliary control register
0BFDC    User 6522 peripheral control register
0BFDD    User 6522 interrupt flag register
0BFDE    User 6522 interrupt enable register
0BFDF    User 6522 Port A without handshake
0BFE0    System 1 6522 System control port data register
0BFE1    System 1 6522 Keyboard port data register
0BFE2    System 1 6522 System control port direction register
0BFE3    System 1 6522 Keyboard port direction register
0BFE4    System 1 6522 timer 1
0BFE5    System 1 6522 timer 1
0BFE6    System 1 6522 timer 1
0BFE7    System 1 6522 timer 1
0BFE8    System 1 6522 timer 2
0BFE9    System 1 6522 timer 2
0BFEA    System 1 6522 shift register (MTUNET)
0BFEB    System 1 6522 auxiliary control register
0BFEC    System 1 6522 peripheral control register
0BFED    System 1 6522 interrupt flag register
0BFEE    System 1 6522 interrupt enable register
0BFEF    System 1 6522 Keyboard port data register without handshake
```

```
0BFF0    System 2 6522 MTUTAPE/cassette port data register
0BFF1    System 2 6522 Audio DAC port data register
0BFF2    System 2 6522 MTUTAPE/cassette port direction register
0BFF3    System 2 6522 Audio DAC port direction register
0BFF4    System 2 6522 timer 1
0BFF5    System 2 6522 timer 1
0BFF6    System 2 6522 timer 1
0BFF7    System 2 6522 timer 1
0BFF8    System 2 6522 timer 2
0BFF9    System 2 6522 timer 2
0BFFA    System 2 6522 shift register  (MTUTAPE)
0BFFB    System 2 6522 auxiliary control register
0BFFC    System 2 6522 peripheral control register
0BFFD    System 2 6522 interrupt flag register
0BFFE    System 2 6522 interrupt enable register
0BFFF    System 2 6522 Audio DAC port data register without handshake
0C000
 ---     4K of Disk Controller RAM or ROM C
0CFFF
0D000
 ---     4K of Disk Controller RAM or ROM D
0DFFF
0E000
 ---     4K of Disk Controller RAM (can be write protected) or ROM E
0EFFF
0F000
 ---     3.75K of Disk Controller RAM (can be write protected) or ROM F
0FDFF
0FF00
 ---     Disk Controller boot ROM or ROM F
0FFE7
0FFE8    Disk Control hardware status read and control write or ROM F
0FFE9    Duplicate of 0FFE8 or ROM F
0FFEA    Disk Control DMA address register or ROM F
0FFEB    Duplicate of 0FFEA or ROM F
0FFEC    Disk Control undefined or ROM F
0FFED    Disk Control undefined or ROM F
0FFEE    Disk Control Main Status Register or ROM F
0FFEF    Disk Control Data Register or ROM F
0FFF0    Disk Control boot ROM or ROM F
 ---
0FFFD
0FFFE    Disk Control boot ROM or ROM F read, disable I/O at 0BE00-0BFFF write
0FFFF    Disk Control boot ROM or ROM F read, enable I/O at 0BE00-0BFFF write
10000
 ---     --Unused--
1BFFF
1C000
 ---     Display memory
1FBFF
1FC00
 ---     General RAM
1FFFF
20000
 ---     --Unused--
3FFFF
```

4.2.3 I/O Area Address Map

Following is a summary of all of the I/O resources on the MONOMEG board. For details, see the section on the respective device or interface.

4.2.3.1 0BFC0 - 0BFC7 Miscellaneous I/O

0BFC0 - Read light pen bit number and hit flag, bits 0-2 are bit number, bit 3 is is hit flag.
0BFC1 - Read light pen address, low 8 bits
0BFC2 - Read light pen address, high 6 bits
0BFC3 - Unused input enable
0BFC4 - Any write resets the light pen hit flag and bit number.
0BFC5 - Any write resets the keyboard scan counter
0BFC6 - Any write toggles the cassette output signal
0BFC7 - Unused pulse output

4.2.3.2 0BFC8 - 0BFCB Serial Port, 6551 chip, see section 5.7

0BFC8 Transmit data (write) and receive data (read) registers
0BFC9 Reset (write) and status register (read)
0BFCA Command register (read and write) (parity, interrupt enables, RTS, DTR)
0BFCB Control register (read and write) (stop bits, data bits, baud rate select)

4.2.3.3 0BFD0 - 0BFDF User parallel port, 6522 chip, see section 5.6

4.2.3.4 0BFE0 - 0BFEF SYS1 internal 6522 chip #1, see section 5.6

		1 0
PA0 Modify key	PB0 Data bank select 0	0 0 = bank 3
PA1 MTUNET Active (level sense)	PB1 Data bank select 1	0 1 = bank 2
PA2 Keyboard row 1	PB2 Program bank select 0	1 0 = bank 1
PA3 Keyboard row 2	PB3 Program bank select 1	1 1 = bank 0
PA4 Keyboard row 3	PB4 Black & white display mode select	
PA5 Keyboard row 4	PB5 Display enable	
PA6 Keyboard row 5	PB6 ROM ENABLE (see note 1)	
PA7 Keyboard row 6	PB7 RAM 08000-0BFFF Write Enable	
CA1 Keyboard BRK key	CB1 MTUNET clock	
CA2 Keyboard counter clock	CB2 MTUNET data	

4.2.3.5 0BFF0 - 0BFFF SYS2 internal 6522 chip #2, see section 5.6

PA0 Audio DAC bit 0 LSB	PB0 MTUTAPE Motor On (output)
PA1 Audio DAC bit 1	PB1 MTUTAPE Fast (output)
PA2 Audio DAC bit 2	PB2 MTUTAPE Write Enable (output)
PA3 Audio DAC bit 3	PB3 MTUTAPE BOT Sensor (input)
PA4 Audio DAC bit 4	PB4 MTUTAPE Write Protect Sensor (input)
PA5 Audio DAC bit 5	PB5 MTUTAPE Wafer Present Sensor (input)
PA6 Audio DAC bit 6	PB6 MTUNET Talk Mode (output)
PA7 Audio DAC bit 7 MSB	PB7 Audio cassette input (input)
CA1 MTUNET active (interrupt on change)	CB1 MTUTAPE Clock (bidirectional)
CA2 Vertical Interval Interrupt	CB2 MTUTAPE Data (bidirectional)

NOTE 1: Jumper option makes this ROM ENABLE

NOTE 2: Note that after power-on or Reset all 20 ports on each of the two system 6522's revert to inputs. All of those that will function as outputs in the system therefore come up in a logical one state.

4.3 I-O/RAM SELECT

In order to accomodate special applications that might demand over 48K of contiguous RAM in bank 0 (such as the UCSD Pascal system), provision is made to disable the I/O address space from BE00 through BFFF and let the on-board RAM that it normally masks show through. Since the I/O might be switched off, the control bit must be manipulated in a somewhat unusual manner. To enable the I/O address space (and therefore mask off the RAM there), the programmer should write into location FFFE in bank 0. To disable the I/O address space (and therefore uncover the RAM), the programmer should write into location FFFF. The data written is not significant; the hardware simply keys off the address. Note that these locations are one of the vectors and therefore will always be some kind of ROM which of course ignores write cycles. The RAM contents are preserved when I/O is selected and manipulated. Power-on or Reset automatically selects I/O.

4.4 ROM/BUS SELECT

The Monomeg CPU can conceivably be used in two kinds of system configurations. In a ROM-based system, the primary system software or even the whole application permantly resides in ROMs. In a RAM-based system, such as the MTU-130, a very small ROM loads the operating software from an external storage device when power is applied. In order to accomodate either system organization, the Monomeg can be jumpered to assign the top 16K of bank 0 to either the on-board ROM sockets or to the external bus. Since the top 16K of bank 0 holds the reset and interrupt vectors, this selection has a profound effect on system organization. However, regardless of the setting of this jumper, a running program can freely switch between ROM and external bus for the top 16K.

The bit that selects between on-board ROM and external bus is bit 6 of port B of the SYS1 I/O chip. The port B data register can be reached at address BFE0. The port B direction register can be reached at address BFE2 and bit 6 of that address must be a 1 before bit 6 of the data register at BFE0 can be altered. It is important to note that the sense of this bit is affected by the ROM/BUS jumper on the Monomeg. If the jumper is set for BUS, then power-up or Reset selects the bus and a 1 in the control bit also selects the bus while a zero selects ROM. This is the normal situation in the MTU-130 system. If the jumper is set for ROM, then power-up or Reset selects the on-board ROM and a 1 in the control bit also selects the ROM while 0 selets the bus. This operation is a natural consequence of the 6522 I/O chips coming up with all ports set to inputs (which float up to logic ones). The ROM/BUS jumper in fact simply inserts or bypasses an inverter between SYS1 port B bit 6 and the ROM/BUS control circuitry.

There is a built-in one instruction delay between when the ROM/BUS select bit is changed and when it is effective. This allows a JMP instruction to follow the STA that changed the bit to make a controlled jump into the altered environment.

4.5 ON-BOARD MEMORY WRITE PROTECT

On-board RAM between 8000 and BFFF may be write protected under software control. Bit 7 of port B of the SYS1 I/O chip controls this feature. The port B data register can be reached at address BFE0. The port B direction register can be reached as address BFE2 and bit 7 of that address must be a 1 before bit 7 of the data register at BFE0 can be altered. Writing is allowed when the control bit is a one. Writing is inhibited when the control bit is a zero. Writing is allowed after power-up or a reset because the I/O port reverts to an input which floats up to a one logic level. Note that there is no indication of an attempted write into protected memory; the memory is simply not written into. Note also that this bit has no effect on writing into I/O registers (when I/O is selected) even though they are in the protected address range.

4.6 EXTENDED ADDRESSING

In order to provide room for growth in systems based on the MONOMEG CPU, provision is made for addressing up to 256K bytes of memory through use of an 18 bit address bus and extended addressing hardware logic. Address space is thus divided up into four banks of 64K bytes each. Full addresses are expressed as 5 hex digits with the leftmost designating the bank number. Since display memory is in bank 1, all normal applications of the Monomeg board will make use of the extended addressing feature.

4.6.1 Extended Addressing Philosophy

Extended addressing in other microprocessor-based systems either uses a very rudimentary scheme of bank selection performed by an output port or a very complex segmentation scheme using block maps. The first method does not increase the amount of memory conveniently addressed by a single program; instead it is most suitable for time sharing systems where a number of relatively small, independent programs must co-exist in memory simultaneously. The segmentation/block map scheme requires a microprocessor whose instruction set design includes a segment field in all memory reference instructions and thus is not directly applicable to the 6502 or other existing 8 bit microprocessors.

Extended addressing in the Monomeg is based on the concept of separation of program and data memory. With such separation, program and data can be up to 64K each in size with little if any loss in programming ease. This is particularly applicable to interpretive languages (BASIC, PASCAL, FORTH) where the "program" is actually data for the interpreter. However it is also applicable to assembly language and compiler language programs which manipulate large data arrays.

Although the 6502 microprocessor chip does not provide any signals to distinguish between data references and program references, some study of its instruction set reveals that only the indirect addressing modes ((indirect,X) and (indirect),Y) are usable to access large blocks of data (assuming no instruction self-modification). All other addressing modes, including instruction fetches, can therefore be assumed to refer to program memory. Thus the extended addressing logic in the Monomeg board keys on the addressing mode being used and steers indirect references (except the indirect jump) to the current data bank and all others to the current program bank. To provide additional flexibility, instructions executed from page 0 or page 1 (presumably self-modified code) using absolute addressing will be interpreted as indirect references as well and refer to the data bank.

4.6.2 Extended Addressing Terminology

Before describing the rules governing extended addressing, several terms must be defined:

1. The System Bank is always bank 0 which is all of the standard MTU-130 memory except the display memory.

2. The current Program Bank is defined by the setting of two I/O register bits.

3. The current Data Bank is defined by the setting of two I/O register bits.

4. Indirect memory references are defined as:
   A. The data fetch or store cycle of instructions using (Indirect,X) addressing.
   B. The data fetch or store cycle of instructions using (Indirect),Y addressing.
   C. The data fetch or store cycle of instructions whose op-code was fetched from page 0 or 1 using ABS, ABS,X, or ABS,Y addressing.

5. Direct memory references are all other memory references by the 6502.

4.6.3 Extended Addressing Rules

Using these definitions, operation of the extended addressing logic can be described as below:

1. Direct memory references (except those covered by 3 and 4 below) refer to the current Program Bank.

2. Indirect memory references refer to the current Data Bank.

3. If a direct memory reference refers to page 0 or 1, the System Bank (Bank 0) is always refered to. Thus the only way to refer to locations 0000-01FF in the other banks is through an indirect reference.

4. Recognition of a Maskable or Non-Maskable Interrupt or execution of a BRK instruction sets an Interrupt Mode flip-flop which forces direct memory references to Bank 0. Indirect references still refer to the current Data Bank. Execution of an RTI instruction resets the Interrupt Mode flip-flop.

5. Direct references by instructions fetched from page 0 or 1 refer to the system bank.

4.6.4 Using Extended Addressing

The Data and Program Bank select bits are found in port B of SYS1 which is at address BFE0. In order to change these bank select bits, the least significant 4 bits of the corresponding direction register at BFE2 must be ones. Bits 0 and 1 select the Data Bank. Their value in binary is the complement of the bank number thus bank 0 is selected by 11, bank 1 by 10, bank 2 by 01, and bank 3 by 00. Bits 2 and 3 select the program bank with similar coding.

After power-on or Reset, the I/O port controlling the bank selection is set to inputs which means that the bank select bits are all ones which in turn selects bank 0 for all memory references. Thus the 6502 behaves normally as if there was no extended addressing logic at all. If a programmer wishes to have a program run in a bank other than zero but does not wish to differentiate between program and data memory, then both pairs of bits should be set to select the same bank.

In assembly language programming the usual case will be Program Bank set to 0 and Data Bank set to something else. In many cases it will be sufficient to simply set the Data Bank once on entry and leave it set. The program code, individual variables, and small tables (such as would be accessed by ABS,X or ABS,Y addressing) would all be in bank 0. The very large data arrays that would be accessed by (IND,X) and (IND),Y addressing would be in the selected data bank. The only really awkward case occurs when moving a large block of data between banks. In such cases it is usually more efficient to use an intermediate buffer in the program bank rather than switching the data bank twice for each byte to be moved.

The best way to transfer control from a program in one bank to a program in another bank is with a small piece of code located in page 0 or 1. This code would change the program bank select bits and then jump to the desired address. This works since direct references to page 0 or 1 always refer to bank zero and thus the code segment is not affected by the setting of the program bank control bits.

Since interrupts automatically defer program execution to bank 0 until they are serviced, it is highly desirable to locate interrupt service routines there. In many cases the interrupt service routine would not use any indirect references so there would be no need to manipulate the bank selection register. If indirect references are used during interrupt service, then the state of the data bank select bits should be saved, set as desired, and restored before returning from the interrupt. Execution of the RTI instruction will restore normal program bank operation.

5. INPUT/OUTPUT PROGRAMMING

The following sections describe programming techniques for the various I/O devices built into the Monomeg CPU. Only in very specialized cases should the user or programmer need to program these devices himself since the CODOS operating system and its associated I/O drivers perform all of the standard I/O functions needed. The main exception is the User Parallel Port which is described in section 5.6.

5.1 DISPLAY GENERATOR PROGRAMMING

Conceptually the display generator is very simple but functionally it is extremely versatile. One simplifying factor is the existence of only two operating modes: two level black and white (which is the power-up default) and 4 level gray scale. SYS1 port B (BFE0) bit 4 selects black and white when a one and gray scale when a zero. Most display applications will use the black and white mode because of its higher resolution. Besides a bit to select between these modes, there is a display blank bit which will blank the screen without disturbing the contents of display memory. The blank bit is SYS1 port B (BFE0) bit 5 which blanks the display if it is a zero. Note that the corresponding direction register bits at BFE2 must be ones in order to change the control bits. Other than the effect of these two bits, the display appearance is governed solely by the contents of display memory.

Other than residing in bank 1 and showing up on the screen, display memory is exactly like any other memory in the system. It may be accessed by the microprocessor at any time with no wait states or other slowdowns. Also, accessing the display memory does not produce any kind of interference on the screen regardless of when or how often it is accessed. Only addresses 1C000-1FBFF are displayed; the remaining 1K from1 FC00-1FFFF are reserved for future use by MTUNET software.

The display is essentially a matrix of dots with 256 rows of 480 dots per row. For addressing purposes the dots can be numbered from 0 to 122,879 with dot 0 being the upper left-hand corner dot, dot 479 being at the upper right corner, dot 480 being the leftmost dot on the next row down, and 122,879 being the lower right-hand corner dot. Eight horizontally adjacent dots make up one byte of memory with the left-right position of the dots on the display corresponding to the natural order of bits in the byte. Thus dot 0 is the leftmost bit (bit 7) of the first byte in the display memory (address 1C000). Conversely dot 479 would be the rightmost bit (bit 0) of the sixtieth byte (address 1C03B).

Usually graphics programming is performed using the X-Y method of identifying a particular dot position. Although the origin of the coordinate system can be assumed to be anywhere, it is convenient to place it at the lower left corner of the display. Thus all of the displayable points are in the first quadrant and X and Y are always positive numbers. To convert from X-Y point coordinates to a byte address and bit number within the byte is a simple matter of evaluating the following equations:

```
BYTE ADDRESS = $C000+(255-Y)*60+INT(X/8)
  BIT NUMBER = 7-REM(X/8)
```

Note that multiplication by 60 can be accomplished in steps as follows: A*60=A*64-A*4 where multiplication by 4 and 64 is accomplished by shifting left 2 and 6 positions respectively. Divison by 8 is accomplished by shifting right 3 positions. Once the byte address and bit number has been found, the ORA instruction can be used to turn the dot on, AND to turn it off, EOR to flip it, and so forth. Lines, circles, and other figures must be drawn one dot at a time by evaluating the equations that define these figures.

Text characters may be generated in two ways. One method is to consider characters as collections of lines ("vectors") where a list of line endpoints that make up each character shape in the font are stored in a "font table" in memory. A line drawing routine would be called to literally draw the character in the desired position on the screen. The advantage of this method is that the endpoint coordinates are easily manipulated to scale and rotate the characters for different applications. The disadvantage is slow drawing speed and large font table size. The other method considers characters to be made of dots in a fixed matrix of for example 5 dots wide and 7 dots high. The font table entry for each character is simply a string of several bytes where bits represent the dots. Placing characters on the screen then is a fairly simple matter of copying the pattern bytes from the font table into the display memory with perhaps some shifting for horizontal positioning. The advantage of this method is high speed and a compact font table. The disadvantage is that scaling and rotation (in 90 degree increments) is difficult and likely to be slower than the vector approach.

In the MTU-130 system, the Graphics and Text I/O driver routines in the operating system have entry points to perform all of the text and graphics functions commonly needed. Please refer to the Graphic and Text Driver section of the CODOS manual for a list of these functions and entry points.

5.2 LIGHT PEN PROGRAMMING

The Light Pen is a graphic input device that can be used to determine the X and Y position of the light sensitive pen when it detects light from the displayed CRT image. A single light pen "hit" is sufficient to establish the precise X and Y coordinates of the hit in 1/60 of a second. This is possible because the light pen circuitry can recognize light and store the state of display generator counters in registers in less than 100NS. In this respect the light pen capability of the Monomeg is much greater than that typically offered in personal computers. In many cases the light pen can eliminate the need for a costly graphic digitizer.

The light pen is controlled via a single bit. The data latched when light is detected is held in 3 registers. A typical light pen use sequence is as follows:

1. Wait for the beginning of vertical retrace. This can be detected as a positive edge on SYS2 CA2 (see section 5.6 for details on programming the 6522 I/O chips).

2. Reset the light pen hit flag by writing anything into 0BFC4.

3. Wait for a light pen hit by waiting for bit 7 of 0BFC0 to go to a one. If desired, a "failure" exit could be taken if there is no hit before the next vertical retrace.

4. At this time the three light pen registers have meaningful data latched which will persist until the hit flag is reset.

The next step is to convert the light pen register content into X and Y screen coordinates. Two of the registers contain the low 14 bits of the display memory address while the third contains the dot number and a "skew flag". One complicating factor however is a FIFO buffer between the display memory and the video shift register. What this means is that the latched display memory address may be 0, 1, 2, or 3 counts ahead of the pixel that caused the light pen hit and latched the registers. Thus the latched address must be corrected based on its value, the dot number, and the state of the skew flag before it can be used with the dot number to compute X and Y coordinates. The full procedure for doing this is as follows:

1. A is defined as the 14 bit latched memory address, low byte at 0BFC1 and high byte at 0BFC2. Be sure to mask off the upper 2 bits of the high byte.

2. B is defined as the 3 bit latched dot number at bits 0-2 of 0BFC0. Be sure to mask off bits 3-7.

3. S is the skew flag which is bit 7 of the byte at BFC2.

4. Decrement A by 2 to form A'.

5. Divide A' by 60 and define C as the quotient (range 0-255) and D as the remainder (range 0-59).

6. Divide D by 10 and define E as the remainder. Discard the quotient.

7. Compute a table lookup address as 16*E+8*S+B (range 0-159).

8. Lookup the correction factor (either -1, 0, +1 or +2) in the table below and add it to C to form C'.

	0	1	2	3	4	5	6	7	8	9
0	X	X	X	X	X	X	-1	-1	0	0
10	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	+1	+1	X	X	X	X
30	X	X	+1	+1	+1	+1	X	X	X	X
40	X	X	0	0	0	0	0	0	X	X
50	X	X	0	0	0	0	+1	+1	+1	+1
60	+1	+1	X	X	X	X	X	X	X	X
70	X	X	X	X	X	X	X	X	X	X
80	0	0	0	0	0	0	0	0	X	X
90	X	X	X	X	-1	-1	+1	+1	+1	+1
100	X	X	X	X	0	0	0	0	0	0
110	0	0	X	X	X	X	0	0	0	0
120	+1	+1	+1	+1	X	X	X	X	+1	+1
130	+1	+1	+1	+1	+1	+1	+2	+2	+2	+2
140	0	0	0	0	0	0	0	0	0	0
150	-1	-1	-1	X	X	X	-1	-1	-1	-1

X=Should not occur.

9. Final X coordinate = 8*C'+B and final Y coordinate = 255-C. It may be desirable to subtract approximately 3 from the X coordinate to compensate for circuit, display monitor, and photodetector response delays.

In the MTU-130 system a complete light pen subroutine is available which returns X and Y coordinates using the method outlined above. Please refer to the Graphics Driver section of the CODOS manual for a listing of available functions and entry point addresses.

Because of the finite field of view of the light pen and circuit noise in the highly sensitive photodectector amplifier, there will be some uncertainty in the coordinate output, particularly when the pen is moving. This uncertainty is usually on the order of 1 coordinate unit in either direction but may reach 2 units under some conditions. If the application involves direct drawing of curves on the screen, it will probably be necessary to apply a smoothing function to the list of coordinates received from the light pen. In addition, the pen will not reliably detect a feature less than 2 pixels wide. In particular a single dot on the screen is generally invisible to the pen unless the monitor brightness is very high. Finally the light pen may produce a spurious hit once in a great while. If this is a potential problem, you might want to require a confirming hit before taking action.

The MTU-130 keyboard is completely software driven which gives great flexibility to programs using the keyboard. In most cases the standard keyboard routines in CODOS are quite adequate but in some applications it may be desirable to address the keyboard circuitry directly. For a description of the standard keyboard routine, see the Text I/O Driver section of the CODOS manual.

All of the keyboard keys (except INT, MOD, RESET, and BREAK) are connected in a matrix having 16 columns and 6 rows. This includes the SHIFT keys, the CTRL key, and even the REPEAT key. The columns are numbered from 0 to 15 and are addressed by the state of a 4 bit counter that can be directly controlled. The counter is reset to zero by writing anything to address 0BFC5. The counter is clocked by a 1-to-0 transition on CA2 of the SYS1 on-board 6522 I/O chip (see section 5.6 for details on programming the 6522 I/O chips). The state of the 6 keyboard rows appears on bits 2 through 7 of port A of the SYS1 I/O chip. If a key in the currently selected column (as determined by the counter) is pressed, the bit associated with the row it resides in will be read as a zero. The other row bits will remain ones. Keyboard routines typically scan the 16 columns looking for a zero to appear in the row bits and take appropriate action when that situation is found. With proper programming of the SYS1 I/O chip, the column counter can be made to automatically increment each time the rows are read and thus speed up a keyboard scan loop. The BREAK key is connected directly to CA1 of the SYS1 I/O chip such that pressing the key produces a logic 1 level.

The matrix arrangement used in the MTU-130 keyboard allows any two keys to be down at once without ambiguity. However when 3 or more keys are pressed simultaneously, there is an excellent possibility of additional "phantom keys" appearing to be pressed when in fact they are not. This effect is avoided with the SHIFT, CTRL, REPEAT, CAPS LOCK and SPACE keys by adding series diodes in the matrix circuitry.

The matrix arrangement of the keyboard keys is shown below. Note that all of the mode keys (shift, etc.) have been placed in column 0 so they may be quickly addressed simply by resetting the column counter.

LEGEND	COL	ROW	LEGEND	COL	ROW	LEGEND	COL	ROW	LEGEND	COL	ROW
f1	1	6	TAB	1	3	J	8	4	up	14	6
f2	2	6	Q	2	3	K	9	4	left	13	6
f3	3	6	W	3	3	L	10	4	HOME	12	6
f4	4	6	E	4	3	; :	11	4	right	10	6
f5	5	6	R	5	3	' "	12	4	down	9	6
f6	6	6	T	6	3	[{	13	4	PF1	13	7
f7	7	6	Y	7	3	RETURN	14	4	PF2	8	7
f8	8	6	U	8	3	DELETE	13	5	x	6	7
ESC	1	2	I	9	3	L. SHIFT	0	5	÷	4	7
1 !	2	2	O	10	3	Z	3	5	7	12	7
2 @	3	2	P	11	3	X	4	5	8	14	7
3 #	4	2	] }	12	3	C	5	5	9	7	7
4 $	5	2	\ |	13	3	V	6	5	-	1	7
5 %	6	2	LINE FEED	14	3	B	7	5	4	10	7
6 ^	7	2	RUB OUT	15	3	N	8	5	5	5	7
7 &	8	2	CTRL	0	3	M	9	5	6	2	7
8 *	9	2	CAPS LOCK	0	4	, >	10	5	+	3	7
9 (	10	2	A	2	4	. <	11	5	1	9	7
0)	11	2	S	3	4	/ ?	12	5	2	15	7
_ -	12	2	D	4	4	R. SHIFT	0	6	3	11	7
= +	13	2	F	5	4	REPEAT	0	2	0	15	6
` ~	14	2	G	6	4	INSERT	14	5	.	15	5
BACKSPACE	15	2	H	7	4	SPACE	2	5	ENTER	11	6

5.4 SOUND GENERATOR PROGRAMMING

The Monomeg CPU board has two different methods of generating sounds. The primary means is a digital-to-analog converter which is capable of producing tones, multi-part music, and even speech with the proper programming. Many of the capabilities of the converter are illustrated by demonstration programs supplied with the MTU-130 system. A secondary means is provided by a programmable oscillator which is part of the User Parallel I/O port and is capable of producing single tones without constant attention by the program. This oscillator is also supported by the MTU BASIC TONE statement.

5.4.1 Digital-to-Analog Converter

The digital-to-analog converter is connected to port A of the SYS2 I/O chip. The data register is at address 0BFFF and the direction register is at address 0BFF3. All 8 bits of the direction register must be set to ones before the data register can be used. The D-to-A converter is offset binary coded which means that a zero (or baseline) value is $80 or decimal 128. Positive full scale is $FF or decimal 255 while negative full scale is $00. When not making sound with the converter, the contents of the data register should be left at the baseline value of $80 to avoid an initial pop when sound generation is resumed.

Making sound with the D-to-A converter is simply a matter of sending bytes to the data register at a high rate of speed, typically 8000 or more bytes per second. The key to producing a particular sound then lies in rapidly calculating the values these bytes should have or reading them from a disk file. This topic is much too involved to describe here so the reader is referred to one of the following MTU documents for more information:

1. K-1002-1ART - 16 page article reprint describes simple 4-voice music synthesis.
2. K-1002-6ART - 22 page article reprint describes more advanced music synthesis.
3. K-1002-BOOK - 653 page book which discusses all aspects of music synthesis.

5.4.2 Programmable Oscillator

A programmable oscillator is provided for generating simple sounds automatically without requiring constant attention from the CPU. The oscillator in fact uses Timer 2 and the shift register in the User 6522 I/O chip. The output of the oscillator is connected into the same audio amplifier used by the D-to-A converter. Note that the oscillator feature uses the CA2 and CB2 signals of the User 6522 I/O chip therefore these must not be connected to external equipment if the oscillator is to be used. Conversely, if the oscillator will not be used and external equipment uses the CB2 signal, extraneous sounds while the equipment is running may be suppressed by grounding the CB2 SOUND OFF signal of the parallel I/O connector.

Three sound parameters are programmable: the frequency, the waveform, and the volume. The frequency parameter (labelled FRQ in later examples) is an 8 bit value. The actual tone frequency is 62500/FRQ. The actual waveform generated consists of 8 segments, each of which may be either +1 or -1 in value. An 8 bit waveform parameter (labelled WAV) determines the polarity of each segment where 1 represents +1 and 0 represents -1. The volume is determined by the content of the D-to-A register (see 5.4.1 above). A value of $00 gives silence while $FF gives maximum volume. The usual quiescent value of $80 gives an intermediate volume level.

Use the following code sequence to start the oscillator with a desired frequency, waveform, and volume:

```
LDA  #$10     ;SELECT FREE RUN SHIFT REGISTER MODE
STA  $BFDB
LDA  FRQ      ;SET TONE FREQUENCY
STA  $BFD8
LDA  WAV      ;SET TONE WAVEFORM
STA  $BFDA
LDA  VOL      ;SET TONE VOLUME (PORT A OUTPUT MODE ASSUMED)
STA  $BFFF
LDA  #0       ;START TONE
STA  $BFD9
```

After the tone has started it will continue sounding indefinitely. The frequency, waveform, and volume may be dynamically altered without stopping and restarting the tone simply by storing new values into the respective registers. There may be some extraneous noise associated with changing the waveform however. Very sudden volume changes should be avoided as well or else a "thumping" sound may be introduced. When sound generation is complete, the oscillator should be stopped by storing zeroes in bit positions 2-4 of address 0BFDB. Setting the volume register to zero should not be used to permanently silence the oscillator because any subsequent use of the D-to-A converter (such as by the CODOS keyboard routine) will make the oscillator audible again.

5.5 CASSETTE INTERFACE PROGRAMMING

The audio cassette interface consists of independent read and write sections. With proper programming, it should be possible to read and write nearly all audio cassette formats in common use. It is expected that audio cassettes will serve as an interchange medium between the MTU-130 and other computers although it could also be used as the primary system storage in systems configured without disks. No cassette motor control signals are explicitly provided. If it is necessary to program control the cassette motor, the User Parallel Port in conjunction with a relay circuit should be used for that purpose.

The write circuit consists of a flip-flop to produce precise program controlled rectangular waves, a filter to remove high frequency harmonics, and an attenuator to provide the proper signal level for the auxiliary input of a cassette recorder. By writing anything to address OBFC6, the programmer can toggle the flip-flop and thus make the audio waveform to the recorder change in polarity. To write a particular cassette format, it is first necessary to know in detail how bits are encoded into waveform polarity reversals.

For example, one method of encoding bits is to mark the beginning of each bit with a polarity reversal. If the bit is to be a one, another reversal is inserted after half a bit time. If the bit is to be a zero, nothing happens during the bit time. This would be programmed simply by writing to OBFC6 at the beginning of a bit, waiting half a bit time, writing to OBFC6 again if the bit is to be a one, waiting half a bit time, and then looping for the next bit. To make readback decoding easier and more reliable, it is desirable to compensate for variable software delays so that the bits are spaced uniformly and the halfway point is indeed halfway between bit boundaries. The method of grouping bits into bytes and bytes into data records is determined by the format specification and is independent of the cassette interface hardware.

The read circuit consists of a filter and zero-crossing detector connected to bit 7 of port B of the SYS2 6522 I/O chip. This register is at address OBFF0 and since the sign bit is of interest, it is very easy to test with the BIT instruction. When the audio signal from the cassette unit is negative, the sign bit will be on and a BIT $BFF0 followed by a BMI will branch. When the audio signal is positive, there would be no branch. In theory, every polarity reversal recorded on the tape would cause a sign reversal of the byte at OBFF0. Note that only the reversal itself is of significance; the direction could be either - to + or + to -.

In practice the reversal timing read from the cassette unit will be somewhat different from that written on the tape. Generally there will be some bias that depends on the recorder itself and some random jitter that depends on the data pattern and on the tape quality. Thus a cassette read program will have to be somewhat tolerant of timing variations if good data accuracy is to be achieved. Best results are obtained with self-clocking bit encoding schemes (such as described above) in conjunction with a read program that simulates a phase-locked loop. The timers in the I/O chips on the Monomeg are quite helpful in this regard although standard timed loop techniques will also work since there are no memory wait states or other timing uncertainties in the Monomeg circuitry. For more details on programming the timers, please refer to section 5.6 which describes the 6522 I/O chips.

The 6522 VIA (Versatile Interface Adapter) chip used to implement the User parallel port provides 2 independent 8 bit ports plus 4 control lines for standard port-oriented I/O. In addition, it has two counter/timers and an 8 bit shift register. This section will describe how to use the parallel port for simple I/O interfacing functions. The reader should refer to the 6522 data sheet in this section for detailed programming instructions for the control lines, counter/timers, and the shift register.

The User 6522 VIA on the Monomeg uses 16 I/O addresses. The table below gives the function of each of these addresses:

ADDRESS	FUNCTION
0BFD0	Port B data register.
0BFD1	Port A data register (normal handshake operation).
0BFD2	Port B direction register.
0BFD3	Port A direction register.
0BFD4	Write timer 1 low latch. Read timer 1 low count and clear timer 1 interrupt flag.
0BFD5	Write timer 1 high latch and high count, transfer low latch to low count. Read timer 1 high count.
0BFD6	Write timer 1 low latch. Read timer 1 low count.
0BFD7	Write timer 1 high latch. Read timer 1 high count.
0BFD8	Write timer 2 low latch. Read timer 2 low count, clear timer 2 int. flag.
0BFD9	Write timer 2 high count, transfer low latch to low count, clear timer 2 interrupt flag. Read high count.
0BFDA	Shift Register.
0BFDB	Auxiliary Control Register (counter & shift register. mode).
0BFDC	Peripheral Control Register (CA1, CA2, CB1, CB2 control).
0BFDD	Interrupt Flag Register.
0BFDE	Interrupt Enable Register.
0BFDF	Port A data register (no effect on handshake).

For simple port-oriented I/O, it is important that the Auxiliary Control Register, Peripheral Control Register, Interrupt Flag Register, and Interrupt Enable Register all be zeroes. Normally system reset can be counted upon to establish this condition. However if the I/O program must be completely self-initializing after a crash, it can simply write 00 into these registers.

The next step is to determine whether the port is to be an input port or an output port. If Port A is to be input, for example, then the program should write 00 into the Port A direction register. This is also accomplished by system reset which sets both direction registers to 00 and thus programs both data registers for inputs. To program a port for all outputs, $FF must be written into its corresponding direction register. One can also mix inputs and outputs on the same port if desired; simply set bits in the direction register to ones that are desired to be outputs in the data register.

Examination of the chart above will reveal that the Port A data register can be reached at two different addresses. With the Peripheral control register set to 00 there is no significant difference between the action of the two addresses. However when the Peripheral control register is set to trigger actions on CA1 and CA2 when port A is manipulated, using the BFDF address will not trigger those actions.

Besides the User parallel port, the Monomeg CPU uses two other 6522 I/O chips to control most of its on-board I/O functions. The user is free to use the timers on the SYS2 chip which uses addresses BFF0 - BFFF.

$\overline{IRQ}$ (Interrupt Request)

The Interrupt Request output goes low whenever an internal interrupt flag is set and the corresponding interrupt enable bit is a logic 1. This output is "open-drain" to allow the interrupt request signal to be "wire-or'ed" with other equivalent signals in the system.

PA0-PA7 (Peripheral A Port)

The Peripheral A port consists of 8 lines which can be individually programmed to act as inputs or outputs under control of a Data Direction Register. The polarity of output pins is controlled by an Output Register and input data may be latched into an internal register under control of the CA1 line. All of these modes of operation are controlled by the system processor through the internal control registers. These lines represent one standard TTL load in the input mode and will drive one standard TTL load in the output mode. Figure 7 illustrates the output circuit.

CA1, CA2 (Peripheral A Control Lines)

The two Peripheral A control lines act as interrupt inputs or as handshake outputs. Each line controls an internal interrupt flag with a corresponding interrupt enable bit. In addition, CA1 controls the latching of data on Peripheral A port input lines. CA1 is a high-impedance input only while CA2 represents one standard TTL load in the input mode. CA2 will drive one standard TTL load in the output mode.

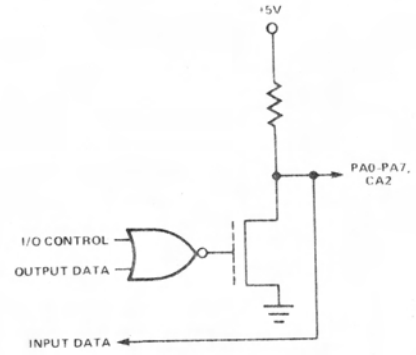


Figure 7. Peripheral A Port Output Circuit

PB0-PB7 (Peripheral B Port)

The Peripheral B port consists of eight bi-directional lines which are controlled by an output register and a data direction register in much the same manner as the PA port. In addition, the polarity of the PB7 output signal can be controlled by one of the interval timers while the second timer can be programmed to count pulses on the PB6 pin. Peripheral B lines represent one standard TTL load in the input mode and will drive one standard TTL load in the output mode. In addition, they are capable of sourcing 1.0mA at 1.5VDC in the output mode to allow the outputs to directly drive Darlington transistor circuits. Figure 8 is the circuit schematic.

CB1, CB2 (Peripheral B Control Lines)

The Peripheral B control lines act as interrupt inputs or as handshake outputs. As with CA1 and CA2, each line controls an interrupt flag with a corresponding interrupt enable bit. In addition, these lines act as a serial port under control of the Shift Register. These lines represent one standard TTL load in the input mode and will drive one standard TTL load in the output mode. Unlike PB0-PB7, CB1 and CB2 cannot drive Darlington transistor circuits.

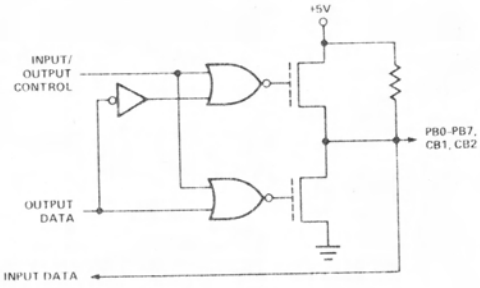


Figure 8. Peripheral B Port Output Circuit

FUNCTIONAL DESCRIPTION

Port A and Port B Operation

Each 8-bit peripheral port has a Data Direction Register (DDRA, DDRB) for specifying whether the peripheral pins are to act as inputs or outputs. A 0 in a bit of the Data Direction Register causes the corresponding peripheral pin to act as an input. A 1 causes the pin to act as an output.

Each peripheral pin is also controlled by a bit in the Output Register (ORA, ORB) and an Input Register (IRA, IRB). When the pin is programmed as an output, the voltage on the pin is controlled by the cor-

responding bit of the Output Register. A 1 in the Output Register causes the output to go high, and a "0" causes the output to go low. Data may be written into Output Register bits corresponding to pins which are programmed as inputs. In this case, however, the output signal is unaffected.

Reading a peripheral port causes the contents of the Input Register (IRA, IRB) to be transferred onto the Data Bus. With input latching disabled, IRA will always reflect the levels on the PA pins. With input latching enabled, IRA will reflect the levels on the PA pins at the time the latching occurred (via CA1).

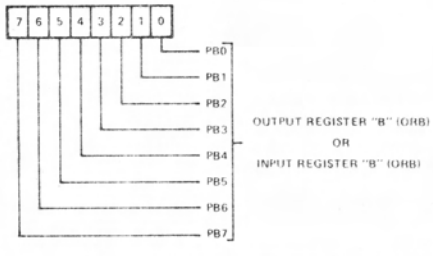
The IRB register operates similar to the IRA register. However, for pins programmed as outputs there is a difference. When reading IRA, the level on the pin determines whether a 0 or a 1 is sensed. When reading IRB, however, the bit stored in the output register, ORB, is the bit sensed. Thus, for outputs which have large loading effects and which pull an output "1" down or which pull an output "0" up, reading IRA may result in reading a "0" when a "1" was actually programmed, and reading a "1" when a "0" was programmed. Reading IRB, on the other hand, will read the "1" or "0" level actually programmed, no matter what the loading on the pin.

Figures 9, 10, and 11 illustrate the formats of the port registers. In addition, the input latching modes are selected by the Auxiliary Control Register (Figure 16.)

Handshake Control of Data Transfers

The SY6522 allows positive control of data transfers between the system processor and peripheral devices through the operation of "handshake" lines. Port A lines (CA1, CA2) handshake data on both a read and a write operation while the Port B lines (CB1, CB2) handshake on a write operation only.

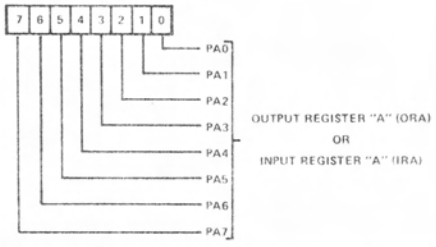
REG 0 – ORB/IRB



Pin Data Direction Selection	WRITE	READ
DDRB = "1" (OUTPUT)	MPU writes Output Level (ORB)	MPU reads output register bit in ORB. Pin level has no affect.
DDRB = "0" (INPUT) (Input latching disabled)	MPU writes into ORB, but no effect on pin level, until DDRB changed	MPU reads input level on PB pin.
DDRB = "0" (INPUT) (Input latching enabled)		MPU reads IRB bit, which is the level of the PB pin at the time of the last CB1 active transition.

Figure 9. Output Register B (ORB), Input Register B (IRB)

REG 1 – ORA/IRA



Pin Data Direction Selection	WRITE	READ
DDRA = "1" (OUTPUT) (Input latching disabled)	MPU writes Output Level (ORA)	MPU reads level on PA pin
DDRA = "1" (OUTPUT) (Input latching enabled)		MPU reads IRA bit which is the level of the PA pin at the time of the last CA1 active transition.
DDRA = "0" (INPUT) (Input latching disabled)	MPU writes into ORA, but no effect on pin level, until DDRA changed	MPU reads level on PA pin.
DDRA = "0" (INPUT) (Input latching enabled)		MPU reads IRA bit which is the level of the PA pin at the time of the last CA1 active transition.

Figure 10. Output Register A (ORA), Input Register A (IRA)

REG 2 (DDRB) AND REG 3 (DDRA)

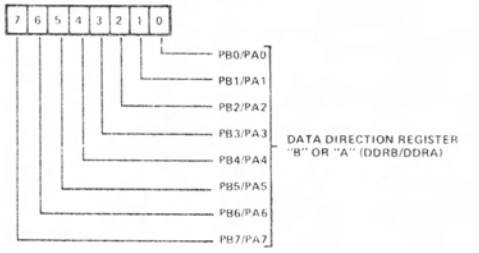


Figure 11. Data Direction Registers (DDRB, DDRA)

Read Handshake

Positive control of data transfers from peripheral devices into the system processor can be accomplished very effectively using Read Handshaking. In this case, the peripheral device must generate the equivalent of a "Data Ready" signal to the processor signifying that valid data is present on the peripheral port. This signal normally interrupts the processor, which then reads the data, causing generation of a "Data Taken" signal. The peripheral device responds by making new data available. This process continues until the data transfer is complete.

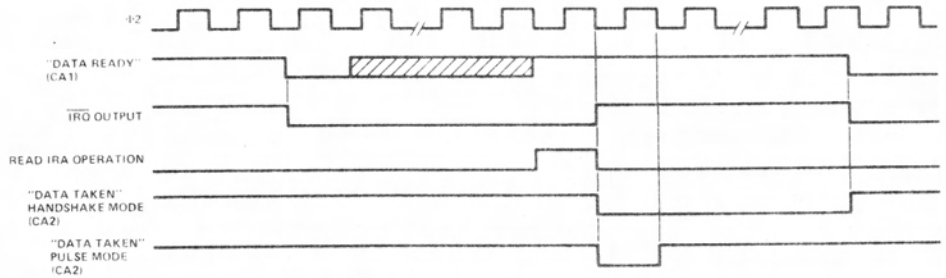


Figure 12. Read Handshake Timing (Port A, Only)

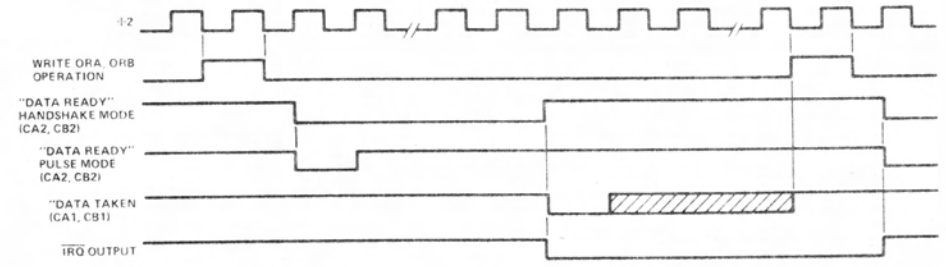


Figure 13. Write Handshake Timing

In the SY6522, automatic "Read" Handshaking is possible on the Peripheral A port only. The CA1 interrupt input pin accepts the "Data Ready" signal and CA2 generates the "Data Taken" signal. The "Data Ready" signal will set an internal flag which may interrupt the processor or which may be polled under program control. The "Data Taken" signal can either be a pulse or a level which is set low by the system processor and is cleared by the "Data Ready" signal. These options are shown in Figure 12 which illustrates the normal Read Handshaking sequence.

Write Handshake

The sequence of operations which allows handshaking data from the system processor to a peripheral device is very similar to that described for Read Handshaking. However, for Write Handshaking, the SY6522 generates the "Data Ready" signal and the peripheral device must respond with the "Data Taken" signal. This can be accomplished on both the PA port and the PB port on the SY6522. CA2 or CB2 act as a "Data Ready" output in either the handshake mode or pulse mode and CA1 or CB1 accept the "Data Taken" signal from the peripheral device, setting the interrupt flag and cleaning the "Data Ready" output. This sequence is shown in Figure 13.

Selection of operating modes for CA1, CA2, CB1, and CB2 is accomplished by the Peripheral Control Register (Figure 14).

Timer Operation

Interval Timer T1 consists of two 8-bit latches and a 16-bit counter. The latches are used to store data which is to be loaded into the counter. After loading, the counter decrements at $\phi 2$ clock rate. Upon reaching zero, an interrupt flag will be set, and $\overline{IRQ}$ will go low if the interrupt is enabled. The timer will then disable any further interrupts, or will automatically transfer the contents of the latches into the counter and will continue to decrement. In addition, the timer may be programmed to invert the output signal on a peripheral pin each time it "times-out". Each of these modes is discussed separately below.

The T1 counter is depicted in Figure 15 and the latches in Figure 16.

REG 12 – PERIPHERAL CONTROL REGISTER

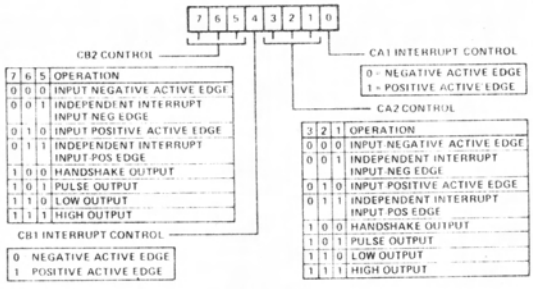
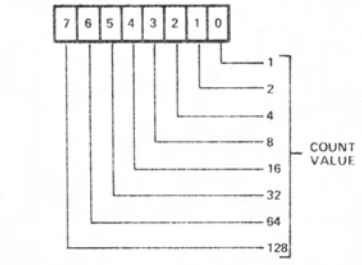


Figure 14. CA1, CA2, CB1, CB2 Control

Two bits are provided in the Auxiliary Control Register (bits 6 and 7) to allow selection of the T1 operating modes. The four possible modes are depicted in Figure 17.

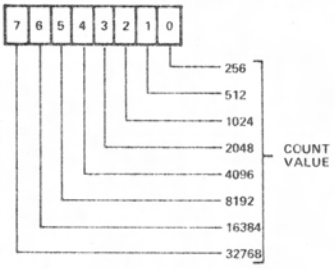
REG 4 – TIMER 1 LOW-ORDER COUNTER



WRITE – 8 BITS LOADED INTO T1 LOW-ORDER LATCHES. LATCH CONTENTS ARE TRANSFERRED INTO LOW-ORDER COUNTER AT THE TIME THE HIGH-ORDER COUNTER IS LOADED (REG 5).

READ – 8 BITS FROM T1 LOW-ORDER COUNTER TRANSFERRED TO MPU. IN ADDITION, T1 INTERRUPT FLAG IS RESET (BIT 6 IN INTERRUPT FLAG REGISTER).

REG 5 – TIMER 1 HIGH-ORDER COUNTER

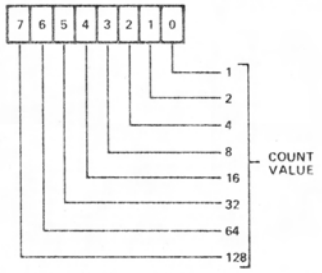


WRITE – 8 BITS LOADED INTO T1 HIGH-ORDER LATCHES. ALSO, AT THIS TIME BOTH HIGH AND LOW-ORDER LATCHES TRANSFERRED INTO T1 COUNTER. T1 INTERRUPT FLAG ALSO IS RESET.

READ – 8 BITS FROM T1 HIGH-ORDER COUNTER TRANSFERRED TO MPU.

Figure 15. T1 Counter Registers

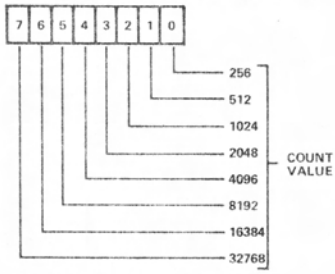
REG 6 – TIMER 1 LOW-ORDER LATCHES



WRITE – 8 BITS LOADED INTO T1 LOW-ORDER LATCHES. THIS OPERATION IS NO DIFFERENT THAT A WRITE INTO REG 4.

READ – 8 BITS FROM T1 LOW-ORDER LATCHES TRANSFERRED TO MPU. UNLIKE REG 4 OPERATION, THIS DOES NOT CAUSE RESET OF T1 INTERRUPT FLAG.

REG 7 – TIMER 1 HIGH-ORDER LATCHES

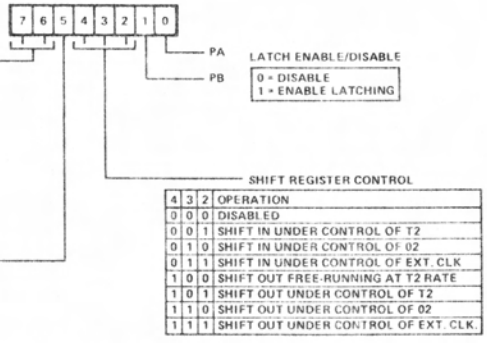


WRITE – 8 BITS LOADED INTO T1 HIGH-ORDER LATCHES. UNLIKE REG 4 OPERATION NO LATCH-TO-COUNTER TRANSFERS TAKE PLACE.

READ – 8 BITS FROM T1 HIGH-ORDER LATCHES TRANSFERRED TO MPU.

Figure 16. T1 Latch Registers

REG 11 – AUXILIARY CONTROL REGISTER



7	6	OPERATION	PB7
0	0	TIMED INTERRUPT EACH TIME T1 IS LOADED	DISABLED
0	1	CONTINUOUS INTERRUPTS	
1	0	TIMED INTERRUPT EACH TIME T1 IS LOADED	ONE-SHOT OUTPUT
1	1	CONTINUOUS INTERRUPTS	SQUARE WAVE OUTPUT

4	3	2	OPERATION
0	0	0	DISABLED
0	0	1	SHIFT IN UNDER CONTROL OF T2
0	1	0	SHIFT IN UNDER CONTROL OF 02
0	1	1	SHIFT IN UNDER CONTROL OF EXT. CLK
1	0	0	SHIFT OUT FREE-RUNNING AT T2 RATE
1	0	1	SHIFT OUT UNDER CONTROL OF T2
1	1	0	SHIFT OUT UNDER CONTROL OF 02
1	1	1	SHIFT OUT UNDER CONTROL OF EXT. CLK.

5	OPERATION
0	TIMED INTERRUPT
1	COUNT DOWN WITH PULSES ON PB6

Figure 17. Auxiliary Control Register

Note: The processor does not write directly into the low order counter (T1C-L). Instead, this half of the counter is loaded automatically from the low order latch when the processor writes into the high order counter. In fact, it may not be necessary to write to the low order counter in some applications since the timing operation is triggered by writing to the high order counter.

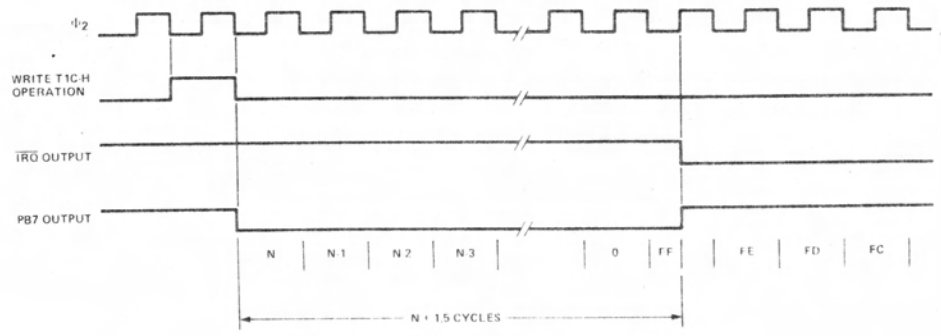


Figure 18. Timer 1 and Timer 2 One-Shot Mode Timing

Timer 1 One-Shot Mode

The interval timer one-shot mode allows generation of a single interrupt for each timer load operation. As with any interval timer, the delay between the "write T1C-H" operation and generation of the processor interrupt is a direct function of the data loaded into the timing counter. In addition to generating a single interrupt, Timer 1 can be programmed to produce a single negative pulse on the PB7 peripheral pin. With the output enabled (ACR7=1) a "write T1C-H" operation will cause PB7 to go low. PB7 will return high when Timer 1 times out. The result is a single programmable width pulse.

In the one-shot mode, writing into the high order latch has no effect on the operation of Timer 1. However, it will be necessary to assure that the low order latch contains the proper data before initiating the count-down with a "write T1C-H" operation. When the processor writes into the high order counter, the T1 interrupt flag will be cleared, the contents of the low order latch will be transferred into the low order counter, and the timer will begin to decrement at system clock rate. If the PB7 output is enabled, this signal will go low on the phase two following the write operation. When the counter reaches zero, the T1 interrupt flag will be set, the $\overline{IRQ}$ pin will go low (interrupt enabled), and the signal on PB7 will go high. At this time the counter will continue to decrement at system clock rate. This allows the system processor to read the contents of the counter to determine the time since interrupt. However, the T1 interrupt flag cannot be set again unless it has been cleared as described in this specification.

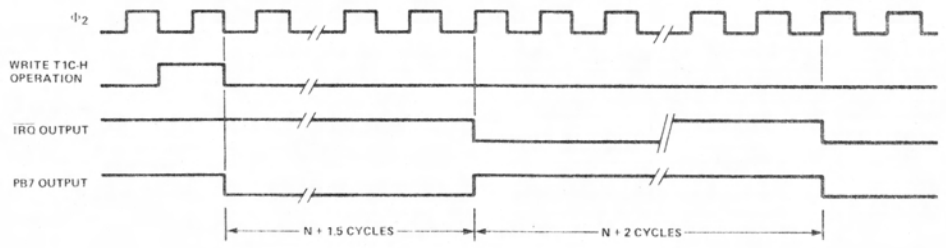
Timing for the SY6522 interval timer one-shot modes is shown in Figure 18.

Timer 1 Free-Run Mode

The most important advantage associated with the latches in T1 is the ability to produce a continuous series of evenly spaced interrupts and the ability to produce a square wave on PB7 whose frequency is not affected by variations in the processor interrupt response time. This is accomplished in the "free-running" mode.

In the free-running mode, the interrupt flag is set and the signal on PB7 is inverted each time the counter reaches zero. However, instead of continuing to decrement from zero after a time-out, the timer automatically transfers the contents of the latch into the counter (16 bits) and continues to decrement from there. The interrupt flag can be cleared by writing T1C-H, by reading T1C-L, or by writing directly into the flag as described later. However, it is not necessary to rewrite the timer to enable setting the interrupt flag on the next time-out.

All interval timers in the SY6522 are "re-triggerable". Rewriting the counter will always re-initialize the time-out period. In fact, the time-out can be prevented completely if the processor continues to rewrite the timer before it reaches zero. Timer 1 will operate in this manner if the processor writes into the high order counter (T1C-H). However, by loading the latches only, the processor can access the timer during each down-counting operation without affecting the time-out in process. Instead, the data loaded into the latches will determine the length of the next time-out period. This capability is particularly valuable in the free-running mode with the output enabled. In this mode, the signal on PB7 is inverted and the interrupt flag is set with each time-out. By responding to the interrupts with new data for the latches, the processor can determine the period of the next half cycle during each half cycle of the output signal on PB7. In this manner, very complex waveforms can be generated. Timing for the free-running mode is shown in Figure 19.



Note: A precaution to take in the use of PB7 as the timer output concerns the Data Direction Register contents for PB7. Both DDRB bit 7 and ACR bit 7 must be 1 for PB7 to function as the timer output. If one is 1 and the other is 0, then PB7 functions as a normal output pin, controlled by ORB bit 7.

Figure 19. Timer 1 Free-Run Mode Timing

Timer 2 Operation

Timer 2 operates as an interval timer (in the "one-slot" mode only), or as a counter for counting negative pulses on the PB6 peripheral pin. A single control bit is provided in the Auxiliary Control Register to select between these two modes. This timer is comprised of a "write-only" low-order latch (T2L-L), a "read-only" low-order counter and a read/write high order counter. The counter registers act as a 16-bit counter which decrements at Φ2 rate. Figure 20 illustrates the T2 Counter Registers.

Timer 2 One-Shot Mode

As an interval timer, T2 operates in the "one-shot" mode similar to Timer 1. In this mode, T2 provides a single interrupt for each "write T2C-H" operation. After timing out, the counter will continue to decrement. However, setting of the interrupt flag will be disabled after initial time-out so that it will not be set by the counter continuing to decrement through zero. The processor must rewrite T2C-H to enable setting of the interrupt flag. The interrupt flag is cleared by reading T2C-L or by writing T2C-H. Timing for this operation is shown in Figure 18.

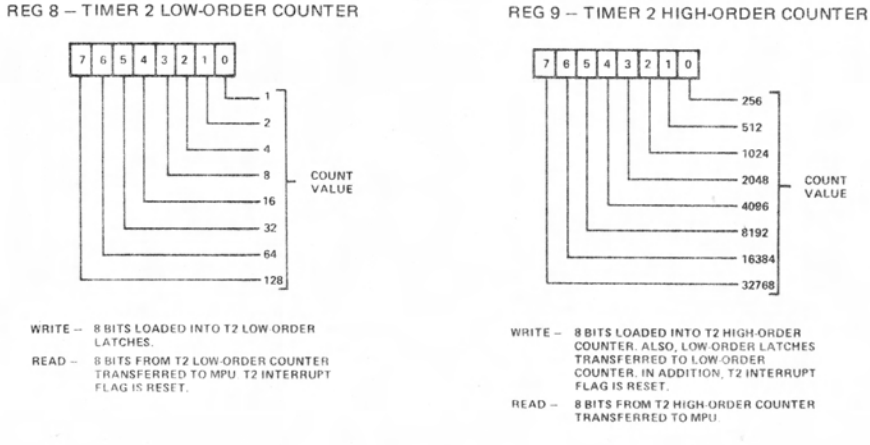


Figure 20. T2 Counter Registers

Timer 2 Pulse Counting Mode

In the pulse counting mode, T2 serves primarily to count a predetermined number of negative-going pulses on PB6. This is accomplished by first loading a number into T2. Writing into T2C-H clears the interrupt flag and allows the counter to decrement each time a pulse is applied to PB6. The interrupt flag will be set when T2 reaches zero. At this time the counter will continue to decrement with each pulse on PB6. However, it is necessary to rewrite T2C-H to allow the interrupt flag to set on subsequent down-counting operations. Timing for this mode is shown in Figure 21. The pulse must be low on the leading edge of $\Phi 2$.

Shift Register Operation

The Shift Register (SR) performs serial data transfers into and out of the CB2 pin under control of an internal modulo-8 counter. Shift pulses can be applied to the CB1 pin from an external source or, with the proper mode selection, shift pulses generated internally will appear on the CB1 pin for controlling external devices.

The control bits which select the various shift register operating modes are located in the Auxiliary Control Register. Figure 22 illustrates the configuration of the SR data bits and the SR control bits of the ACR.

Figures 23 and 24 illustrate the operation of the various shift register modes.

Interrupt Operation

Controlling interrupts within the SY6522 involves three principal operations. These are flagging the interrupts, enabling interrupts and signaling to the processor that an active interrupt exists within the chip. Interrupt flags are set by interrupting conditions which exist within the chip or on inputs to the chip. These flags normally remain set until the interrupt has been serviced. To determine the source of an interrupt, the microprocessor must examine these flags in order from highest to lowest priority. This is accomplished by reading the flag register into the processor accumulator, shifting this register either right or left and then using conditional branch instructions to detect an active interrupt.

Associated with each interrupt flag is an interrupt enable bit. This can be set or cleared by the processor to enable interrupting the processor from the corresponding interrupt flag. If an interrupt flag is set to a logic 1 by an interrupting condition, and the corresponding interrupt enable bit is set to a 1, the Interrupt Request Output ($\overline{IRQ}$) will go low. $\overline{IRQ}$ is an "open-collector" output which can be "wire-or'ed" with other devices in the system to interrupt the processor.

In the SY6522, all the interrupt flags are contained in one register. In addition, bit 7 of this register will be read as a logic 1 when an interrupt exists within the chip. This allows very convenient polling of several devices within a system to locate the source of an interrupt.

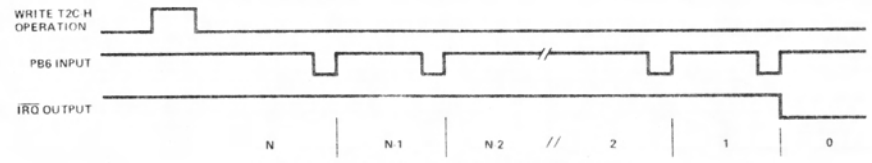
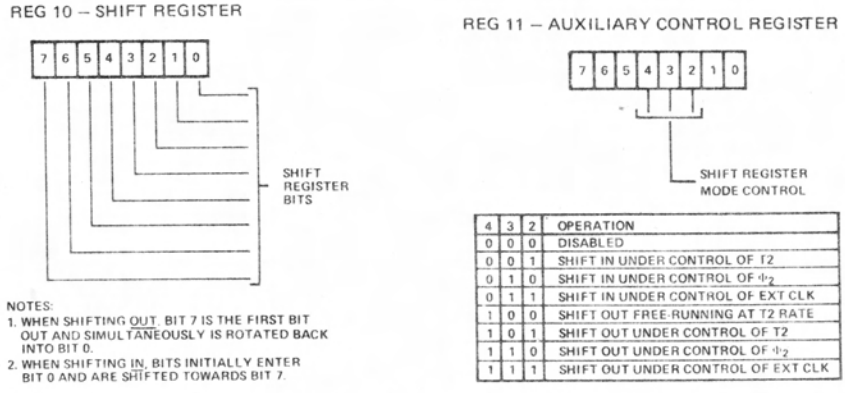


Figure 21. Timer 2 Pulse Counting Mode



4	3	2	OPERATION
0	0	0	DISABLED
0	0	1	SHIFT IN UNDER CONTROL OF T2
0	1	0	SHIFT IN UNDER CONTROL OF Φ_2
0	1	1	SHIFT IN UNDER CONTROL OF EXT CLK
1	0	0	SHIFT OUT FREE-RUNNING AT T2 RATE
1	0	1	SHIFT OUT UNDER CONTROL OF T2
1	1	0	SHIFT OUT UNDER CONTROL OF Φ_2
1	1	1	SHIFT OUT UNDER CONTROL OF EXT CLK

Figure 22. SR and ACR Control Bits

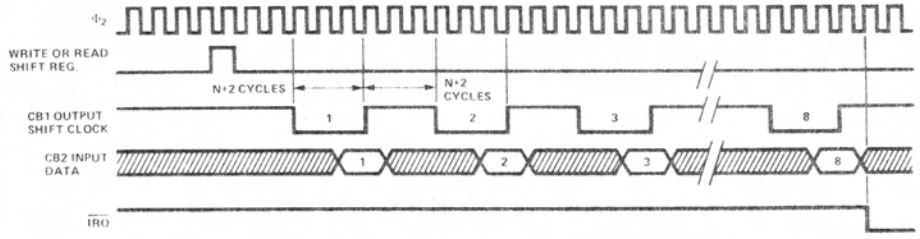
SR Disabled (000)

The 000 mode is used to disable the Shift Register. In this mode the microprocessor can write or read the SR, but the shifting operation is disabled and operation of CB1 and CB2 is controlled by the appropriate bits in the Peripheral Control Register (PCR). In this mode the SR Interrupt Flag is disabled (held to a logic 0).

Shift in Under Control of T2 (001)

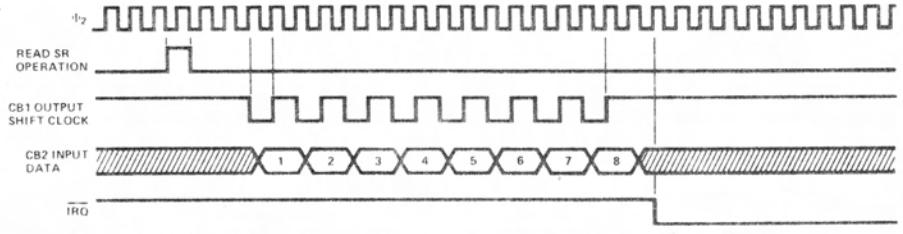
In the 001 mode the shifting rate is controlled by the low order 8 bits of T2. Shift pulses are generated on the CB1 pin to control shifting in external devices. The time between transitions of this output clock is a function of the system clock period and the contents of the low order T2 latch (N).

The shifting operation is triggered by writing or reading the shift register. Data is shifted first into the low order bit of SR and is then shifted into the next higher order bit of the shift register on the negative-going edge of each clock pulse. The input data should change before the positive-going edge of the CB1 clock pulse. This data is shifted into the shift register during the ϕ_2 clock cycle following the positive-going edge of the CB1 clock pulse. After 8 CB1 clock pulses, the shift register interrupt flag will be set and $\overline{IRQ}$ will go low.



Shift in Under Control of ϕ_2 (010)

In mode 010 the shift rate is a direct function of the system clock frequency. CB1 becomes an output which generates shift pulses for controlling external devices. Timer 2 operates as an independent interval timer and has no effect on SR. The shifting operation is triggered by reading or writing the Shift Register. Data is shifted first into bit 0 and is then shifted into the next higher order bit of the shift register on the trailing edge of each ϕ_2 clock pulse. After 8 clock pulses, the shift register interrupt flag will be set, and the output clock pulses on CB1 will stop.



Shift in Under Control of External CB1 Clock (011)

In mode 011 CB1 becomes an input. This allows an external device to load the shift register at its own pace. The shift register counter will interrupt the processor each time 8 bits have been shifted in. However, the shift register counter does not stop the shifting operation; it acts simply as a pulse counter. Reading or writing the Shift Register resets the Interrupt flag and initializes the SR counter to count another 8 pulses.

Note that the data is shifted during the first system clock cycle following the positive-going edge of the CB1 shift pulse. For this reason, data must be held stable during the first full cycle following CB1 going high.

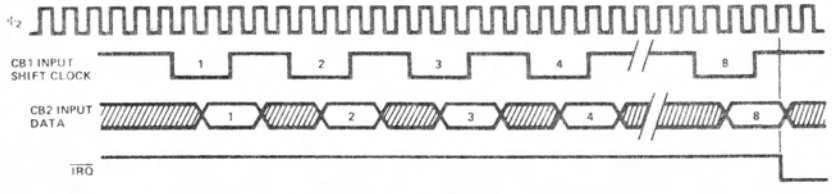
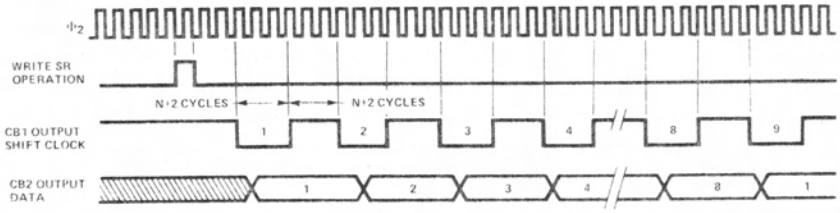


Figure 23. Shift Register Input Modes

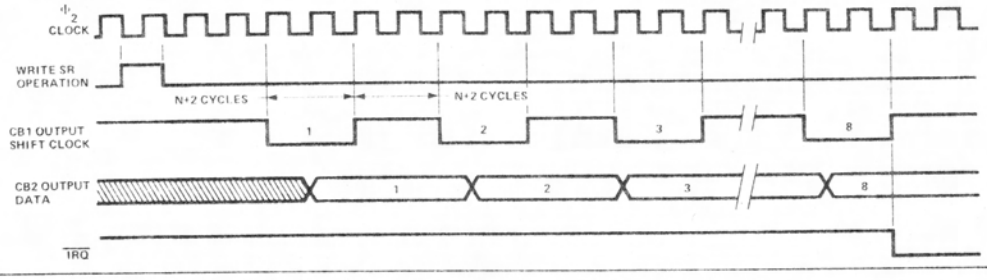
Shift Out Free-Running at T2 Rate (100)

Mode 100 is very similar to mode 101 in which the shifting rate is set by T2. However, in mode 100 the SR Counter does not stop the shifting operation. Since the Shift Register bit 7 (SR7) is recirculated back into bit 0, the 8 bits loaded into the shift register will be clocked onto CB2 repetitively. In this mode the shift register counter is disabled.



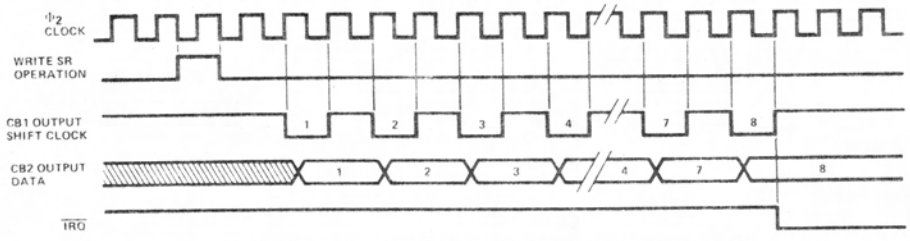
Shift Out Under Control of T2 (101)

In mode 101 the shift rate is controlled by T2 (as in the previous mode). However, with each read or write of the shift register the SR Counter is reset and 8 bits are shifted onto CB2. At the same time, 8 shift pulses are generated on CB1 to control shifting in external devices. After the 8 shift pulses, the shifting is disabled, the SR Interrupt Flag is set and CB2 remains at the last data level.



Shift Out Under Control of ϕ_2 (110)

In mode 110, the shift rate is controlled by the ϕ_2 system clock.



Shift Out Under Control of External CB1 Clock (111)

In mode 111 shifting is controlled by pulses applied to the CB1 pin by an external device. The SR counter sets the SR Interrupt flag each time it counts 8 pulses but it does not disable the shifting function. Each time the microprocessor writes or reads the shift register, the SR Interrupt flag is reset and the SR counter is initialized to begin counting the next 8 shift pulses on pin CB1. After 8 shift pulses, the interrupt flag is set. The microprocessor can then load the shift register with the next byte of data.

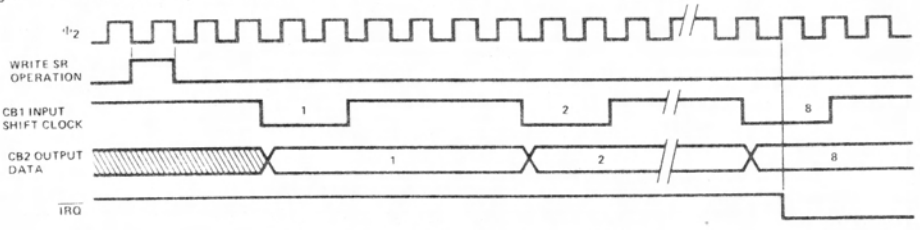


Figure 24. Shift Register Output Modes

The Interrupt Flag Register (IFR) and Interrupt Enable Register (IER) are depicted in Figures 25 and 26, respectively.

The IFR may be read directly by the processor. In addition, individual flag bits may be cleared by writing a "1" into the appropriate bit of the IFR. When the proper chip select and register signals are applied to the chip, the contents of this register are placed on the data bus. Bit 7 indicates the status of the IRQ output. This bit corresponds to the logic function: IRQ = IFR6 x IER6 + IFR5 x IER5 + IFR4 x IER4 + IFR3 x IER3 + IFR2 x IER2 + IFR1 x IER1 + IFR0 x IER0. Note: X = logic AND, + = Logic OR.

The IFR bit 7 is not a flag. Therefore, this bit is not directly cleared by writing a logic 1 into it. It can only be cleared by clearing all the flags in the register or by disabling all the active interrupts as discussed in the next section.

REG 13 – INTERRUPT FLAG REGISTER

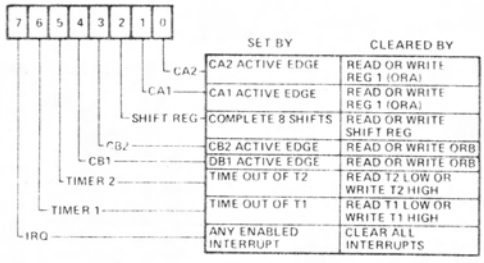


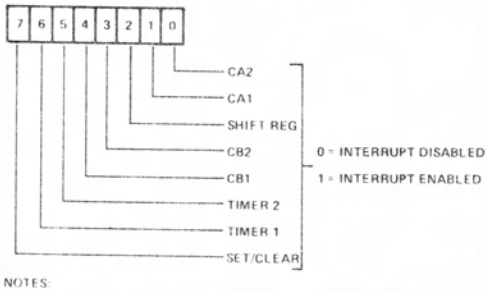
Figure 25. Interrupt Flag Register (IFR)

For each interrupt flag in IFR, there is a corresponding bit in the Interrupt Enable Register. The system processor can set or clear selected bits in this register to facilitate controlling individual interrupts without affecting others. This is accomplished by writing to address 1110 (IER address). If bit 7 of the data placed on the system data bus during this write operation is a 0, each 1 in bits 6 through 0 clears the corresponding bit in the Interrupt Enable Register. For each zero in bits 6 through 0, the corresponding bit is unaffected.

Setting selected bits in the Interrupt Enable Register is accomplished by writing to the same address with bit 7 in the data word set to a logic 1. In this case, each 1 in bits 6 through 0 will set the corresponding bit. For each zero, the corresponding bit will be unaffected. This individual control of the setting and clearing operations allows very convenient control of the interrupts during system operation.

In addition to setting and clearing IER bits, the processor can read the contents of this register by placing the proper address on the register select and chip select inputs with the R/W line high. Bit 7 will be read as a logic 0.

REG 14 – INTERRUPT ENABLE REGISTER



NOTES:
1. IF BIT 7 IS A "0", THEN EACH "1" IN BITS 0 - 6 DISABLES THE CORRESPONDING INTERRUPT.
2. IF BIT 7 IS A "1", THEN EACH "1" IN BITS 0 - 6 ENABLES THE CORRESPONDING INTERRUPT.
3. IF A READ OF THIS REGISTER IS DONE, BIT 7 WILL BE "0" AND ALL OTHER BITS WILL REFLECT THEIR ENABLE/DISABLE STATE.

Figure 26. Interrupt Enable Register (IER)

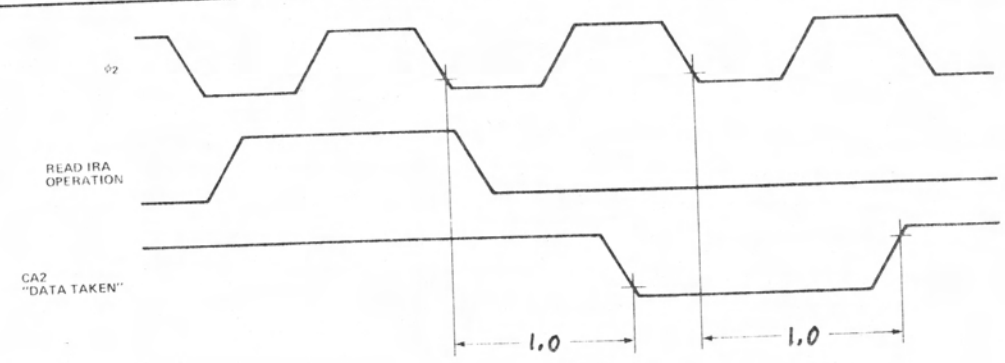


Figure 5a. CA2 Timing for Read Handshake, Pulse Mode

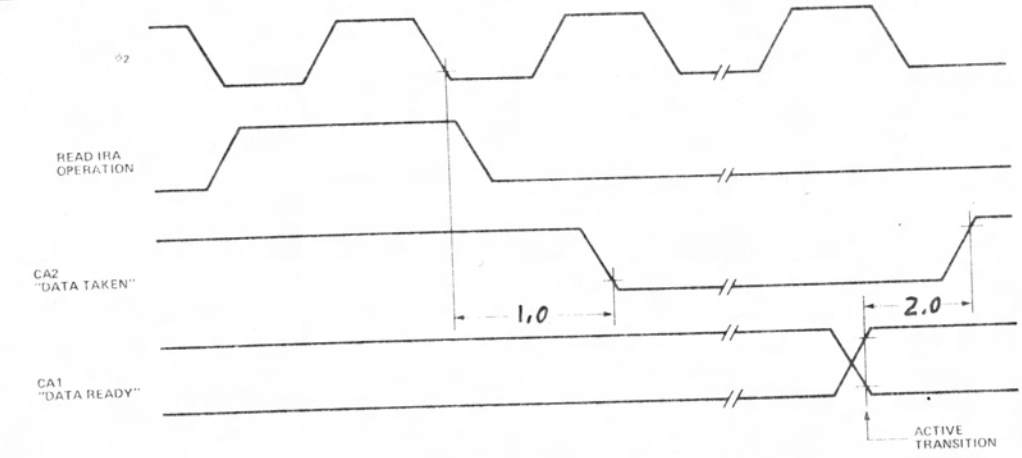


Figure 5b. CA2 Timing for Read Handshake, Handshake Mode

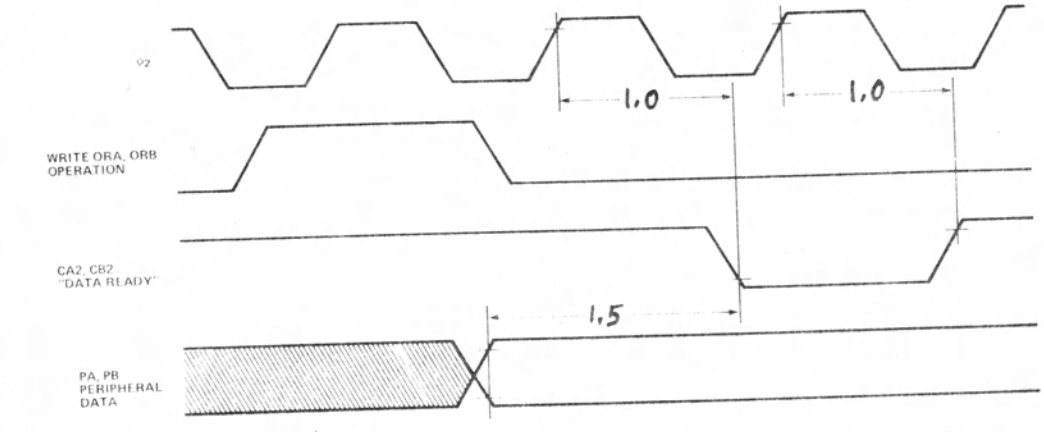


Figure 5c. CA2, CB2 Timing for Write Handshake, Pulse Mode

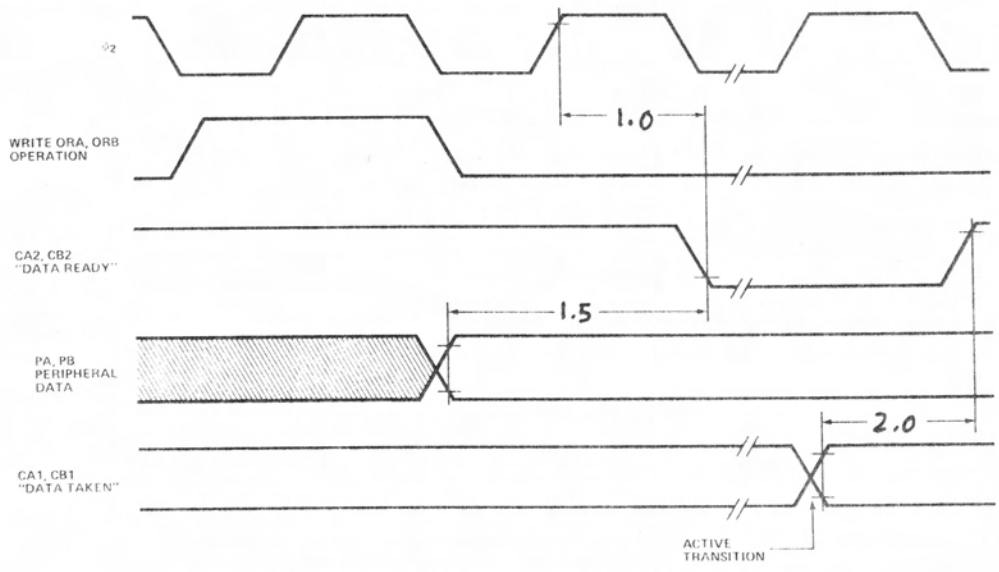


Figure 5d. CA2, CB2 Timing for Write Handshake, Handshake Mode

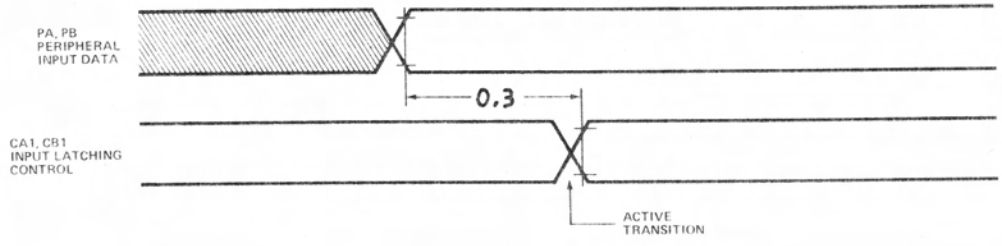


Figure 5e. Peripheral Data Input Latching Timing

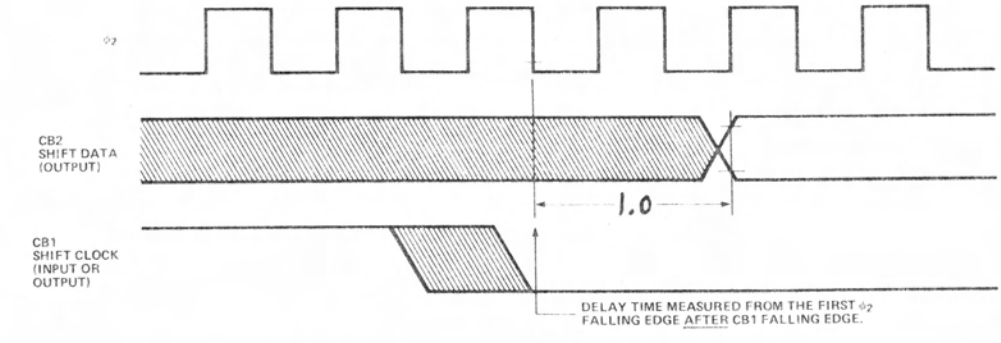


Figure 5f. Timing for Shift Out with Internal or External Shift Clocking

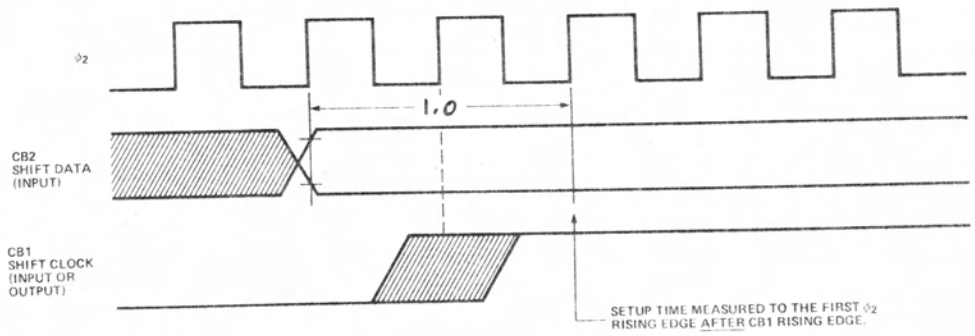


Figure 5g. Timing for Shift In with Internal or External Shift Clocking

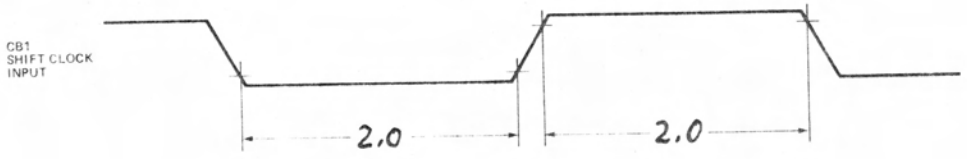


Figure 5h. External Shift Clock Timing

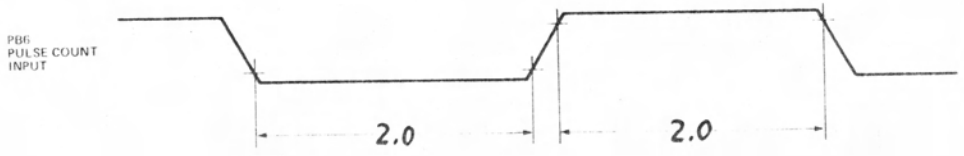


Figure 5i. Pulse Count Input Timing

5.7 SERIAL PORT PROGRAMMING

The serial port on the Monomeg is implemented using a 6551 Asynchronous Communications Adapter (ACIA) integrated circuit. Besides performing the usual transmitter and receiver functions, this chip has its own baud rate generator so the baud rate is programmable as well. In addition to these features, the most important modem control signals are implemented for applications involving a telephone data set.

The first step in using the serial port is to reset the chip and then set up the various transmission parameters. Reset is accomplished by writing anything to address 0BFC9. Two registers are involved in specifying the transmission parameters and speed. The standard setting of the Command Register at address 0BFCA is $0B which gives no parity, independent transmitter and receiver operation (no echo), no receive/transmit interrupts, and asserts the DTR (data terminal ready) and RTS (request to send) modem control signals. The value $05 is the same except that interrupts from the transmitter and receiver are enabled.

The Control Register at address 0BFCB determines the byte format and the baud rate. The standard setting of the upper 4 bits (left hex digit) of this register is $1 which gives 8 data bits, 1 stop bit, and selects the on-chip baud rate generator. The value $9 is similar except for two stop bits which is usually required when talking to a teletype at 110 baud. The lower 4 bits (right hex digit) determines the baud rate. Typical values are $6 for 300 baud, $A for 2400 baud, and $E for 9600 baud. Settings for other rates are shown on the 6551 data sheet.

After resetting the chip and setting the Command and Control registers, the serial port is ready for transmitting and receiving. The Data Register is at address 0BFC8. When read, its contents represent the character last received. When written to, its contents specify the character to be sent. The chip is ready to transmit a character whenever bit 4 (mask value $10) of the Status Register at address 0BFC9 is a one. A character is transmitted simply by storing it at 0BFC8 which then clears bit 4 until the chip is ready for another character. A character has been received when bit 3 (mask value $08) of the Status Register is a one. Reading the character from the Data Register resets bit 3 to zero until another character is received. (Caution: resetting the chip does not clear the received data register therefore if the status register indicates that a character has been received immediately after reset, it should be read and discarded.)

Interrupts my originate from 4 different sources in the serial interface. One of these is generated by the transmitter when bits 3 and 2 in the Command Register are 0 and 1 respectively and a character has been transmitted. Another is generated by the receiver when Command Register bit 1 is a zero and a character has been received. The remaining two interrupts are generated whenever the Data Set Ready or Data Carrier Detect modem control signals change state and the receiver is enabled. Thus a program using the receiver must be prepared to handle these latter two interrupts or steps must be taken to prevent signal level changes on the DSR and DCD lines (including noise cross-coupled in the connecting cable). Additionally, when a program has finished using the serial port, it should reset the chip (by writing into the Status Register), and then writing $02 into the Command Register to prevent spurious interrupts from noise. Unexplained "IRQ" interrupts when a program is run are usually caused by a prior program failing to "idle" the serial port and can be cleared by pressing the Reset key.

The Status Register also has a number of bits for error conditions and the modem control signals. Please refer to the 6551 data sheet for the meaning of these bits. Note that if nothing is connected to the CTS, DSR, and DCD input pins on the serial connector all three of these signals will appear to be active because of pullup resistors on the Monomeg.

CONTROL REGISTER (BFCB)

The Control Register is used to select the desired mode for the SY6551. The word length, number of stop bits, and clock controls are all determined by the Control Register

CONTROL REGISTER

7	6	5	4	3	2	1	0

STOP BITS

0 = 1 Stop Bit
1 = 2 Stop Bits
1 Stop Bit if Word Length = 8 Bits and Parity*
1½ Stop Bits if Word Length = 5 Bits and No Parity.

WORD LENGTH

BIT 6	BIT 5	DATA WORD LENGTH
0	0	8
0	1	7
1	0	6
1	1	5

RECEIVER CLOCK SOURCE

0 = External Receiver Clock
1 = Baud Rate Generator

BAUD RATE GENERATOR

0	0	0	0	16x EXTERNAL CLOCK
0	0	0	1	50 BAUD
0	0	1	0	75
0	0	1	1	109.92
0	1	0	0	134.58
0	1	0	1	150
0	1	1	0	300
0	1	1	1	600
1	0	0	0	1200
1	0	0	1	1800
1	0	1	0	2400
1	0	1	1	3600
1	1	0	0	4800
1	1	0	1	7200
1	1	1	0	9600
1	1	1	1	19,200

*This allows for 9 bit transmission (8 data bits plus parity)

	7	6	5	4	3	2	1	0
HARDWARE RESET	0	0	0	0	0	0	0	0
PROGRAM RESET	–	–	–	–	–	–	–	–

COMMAND REGISTER (BFCA)

The Command Register is used to control Specific Transmit/Receive functions

COMMAND REGISTER

7	6	5	4	3	2	1	0

PARITY CHECK CONTROLS

BIT 7	BIT 6	BIT 5	OPERATION
–	–	0	Parity Disabled - No Parity Bit Generated - No Parity Bit Received
0	0	1	Odd Parity Receiver and Transmitter
0	1	1	Even Parity Receiver and Transmitter
1	0	1	Mark Parity Bit Transmitted, Parity Check Disabled
1	1	1	Space Parity Bit Transmitted, Parity Check Disabled

DATA TERMINAL READY

0 = Disable Receiver and All Interrupts ($\overline{DTR}$ high)
1 = Enable Receiver and All Interrupts ($\overline{DTR}$ low)

RECEIVER INTERRUPT ENABLE

0 = $\overline{IRQ}$ Interrupt Enabled from Bit 3 of Status Register
1 = $\overline{IRQ}$ Interrupt Disabled

TRANSMITTER CONTROLS

BIT 3	BIT 2	TRANSMIT INTERRUPT	$\overline{RTS}$ LEVEL	TRANSMITTER
0	0	Disabled	High	Off
0	1	Enabled	Low	On
1	0	Disabled	Low	On
1	1	Disabled	Low	Transmit BRK

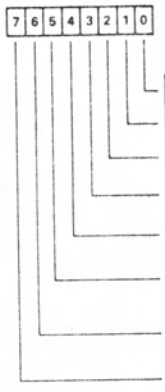
NORMAL/ECHO MODE FOR RECEIVER

0 = Normal
1 = Echo (Bits 2 and 3 must be "0")

	7	6	5	4	3	2	1	0
HARDWARE RESET	0	0	0	0	0	0	0	0
PROGRAM RESET	–	–	–	0	0	0	0	0

STATUS REGISTER (BFC9)

The Status Register is used to indicate to the processor the status of various SY6551 functions



	STATUS	SET BY	CLEARED BY
7	Parity Error*	0 = No Error 1 = Error	Self Clearing**
6	Framing Error*	0 = No Error 1 = Error	Self Clearing**
5	Overrun*	0 = No Error 1 = Error	Self Clearing**
4	Receive Data Register Full	0 = Not Full 1 = Full	Read Receive Data Register
3	Transmit Data Register Empty	0 = Not Empty 1 = Empty	Write Transmit Data Register
2	$\overline{DCD}$	0 = $\overline{DCD}$ Low 1 = $\overline{DCD}$ High	Not Resettable Reflects $\overline{DCD}$ State
1	$\overline{DSR}$	0 = $\overline{DSR}$ Low 1 = $\overline{DSR}$ High	Not Resettable Reflects $\overline{DSR}$ State
0	IRQ	0 = No Interrupt 1 = Interrupt	Read Status Register

*NO INTERRUPT GENERATED FOR THESE CONDITIONS.
**CLEARED AUTOMATICALLY AFTER A READ OF RDR AND THE NEXT ERROR FREE RECEIPT OF DATA.

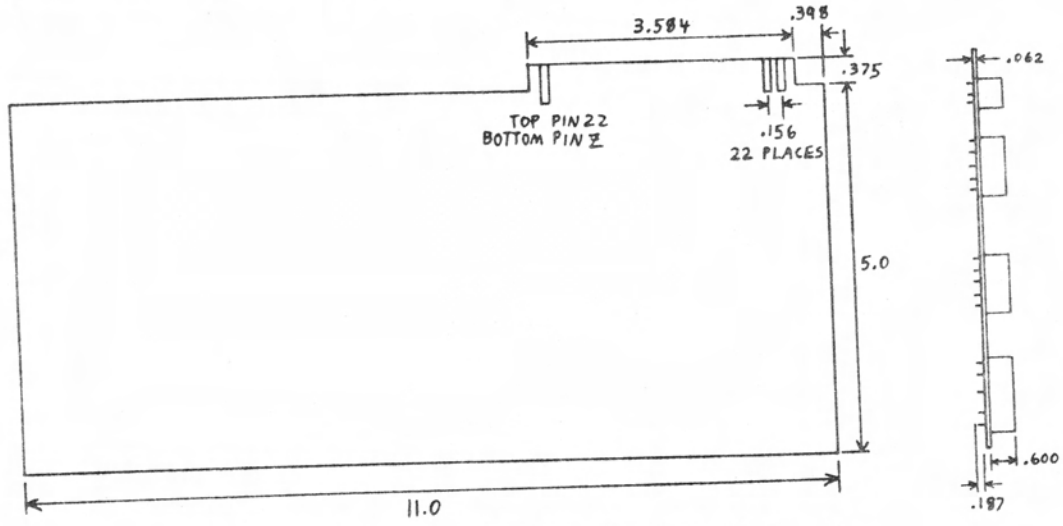
	7	6	5	4	3	2	1	0
HARDWARE RESET	0	–	–	1	0	0	0	0
PROGRAM RESET	–	–	–	–	–	0	–	–

6. USER INTERFACING TO THE MTU-130 BUS

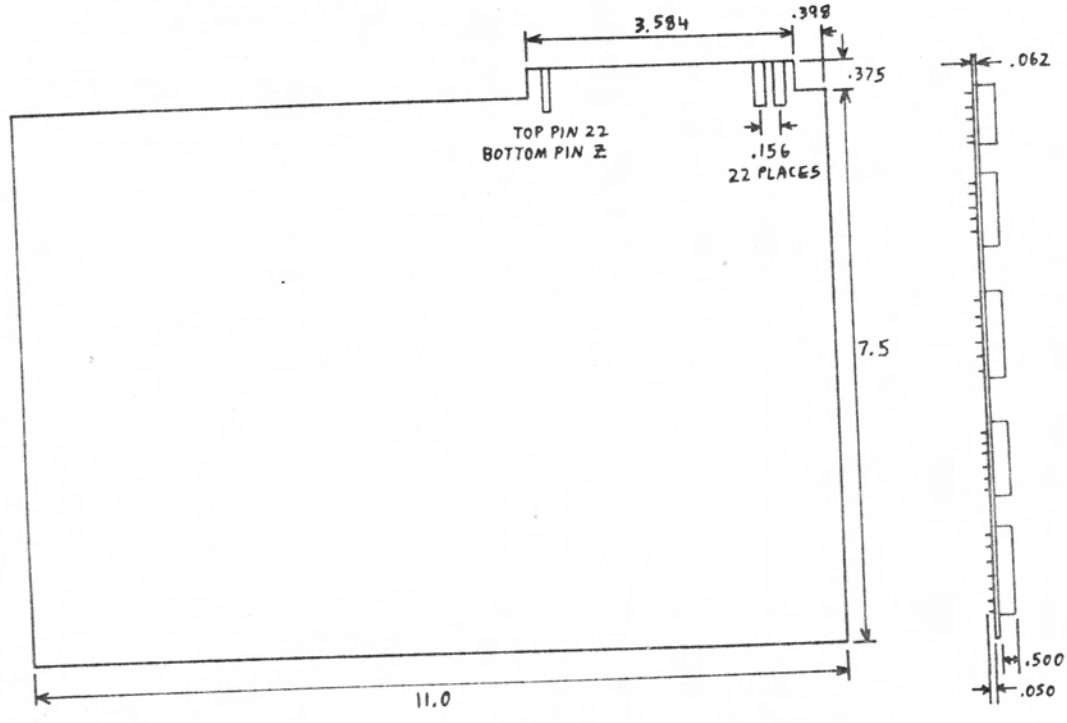
This section provides the information necessary for the design of custom bus interface boards. Such boards typically contain I/O interfaces that are not conveniently handled through the standard I/O ports provided. Specialized memory expansion, such as large amounts of ROM or non-volitale read/write memory might also be accommodated with a custom bus interface board.

6.1 PHYSICAL INTERFACE BOARD REQUIREMENTS

The MTU-130 bus normally contains 3 empty slots intended for bus interface boards. All 3 have indentical bus connectors but the physical board dimensions allowed vary as shown by the sketches below:



Dimensions for Either of the Top 2 Slots.



Dimensions for the Third Slot.

The model K-1020 wire-wrap prototype board will fit any of the three available slots. Note that very little depth is allowed for the wire-wrap pins so as a practical matter, only the top slot may be used for wire-wrapped prototype boards. Two small (1" by 1.5") cutouts and blank panels are provided on the rear of the MTU-130 just behind the empty slots for mounting customer supplied I/O connectors.

6.2 POWER PROVISIONS

DC power for customer interfaces is available on the bus connector as outlined below:

6.2.1 Pin 22 GROUND - This is the return point for all power currents and signals on the interface board. It must be connected directly to any power input bypass capacitors and bus logic signal buffers used on the board.

6.2.2 Pin 21 +5 Regulated - This pin may be used to supply power to a simple interface board that requires less than 50MA of current and can tolerate some noise (the Monomeg draws its power from this pin). For boards requiring more current or better regulation, you should use the +8 volt unregulated source and an on-board 5 volt IC regulator.

6.2.3 Pin 18 +8 Unregulated - This pin may be used to supply up to .66 amp of raw power to a custom interface board. If more current is required, the sum total of all three expansion slots must be less than 2 amps. The actual voltage found at this pin with no additional load and nominal line voltage is approximately 9 volts average with 1 volt peak-to-peak of ripple superimposed. Five volt IC regulators such as LM340T-5 or 7805 types may be used to provide +5 volt logic power from this source.

6.2.4 Pin 17 +12 Regulated - This pin may be used to supply power to small interface boards using op-amps and 12 volt memory circuits if the current drain is less than 35MA. There may be some noise on this pin because the Monomeg uses it to power its memory and analog circuits. For boards requiring more current or better regulation, you should use the +16 volt unregulated source and an on-board +12 volt regulator.

6.2.5 Pin 16 -12 Regulated - This pin may be used to supply up to 66MA of regulated -12 to analog circuitry such as op-amps. If more current is required, the sum total of all 3 expansion slots must be less than .2 amps.

6.2.6 Pin X +16 Unregulated - This pin may be used to supply up to .25 amp of raw high voltage power to a custom interface board. If more current is required, the sum total of all 3 expansion slots must be less than .75 amps. The actual voltage found at this pin with no additional load and nominal line voltage is approximately 19 volts average with 1 volt peak-to-peak of ripple superimposed. Twelve volt IC regulators such as LM340T-12 or 7812 types may be used to provide +12 volt analog power from this source. It is recommended that you not attempt to derive +15 volts from this source because there will be very little margin left for line voltage fluctuations.

6.2.7 Power Input Bypassing

You should use input bypass capacitors on all power voltages used. Most IC regulators require at least 47uF of aluminum electrolytic capacitor at their input to suppress oscillation. To prevent large circulating ripple currents, you should not use an input bypass capacitor larger than 1000uF.

6.2.8 Deriving Other Voltages

In special cases, voltages other than those listed above may be required. The best way to obtain these voltages is by means of "charge pump" circuits. For maximum efficiency, it is recommended that the +16 unregulated voltage be used to power the charge pump. When an outside source of power is used, you should restrict power consumption on a single board to 10 watts to insure adequate cooling.

6.3 BUS SIGNALS

The paragraphs below describe the function of each bus signal used in the MTU-130. For readers familiar with the bus structure used in KIM-1, SYM-1, and AIM-65 computers, you will find that significant differences are confined to the addition of 2 address bus lines and the I/O ENABLE signal. Unless otherwise specified, all lines driven by the Monomeg are buffered and are capable of driving 5 standard TTL loads (8MA in the low state). Pin connections for the bus signals may be found in paragraph 3.1.

6.3.1 Address Bus

The address bus consists of 18 lines designated AB0 - AB17 with AB0 being the least significant. These lines are buffered and are capable of driving up to 10 standard TTL loads. They are always driven by the Monomeg and are <u>not</u> tri-state. There is nothing special about the AB16 and AB17 lines, they should be decoded and handled just like the other 16 address lines.

6.3.2 I/O Select

This signal is associated with the address bus and is provided to simplify the design of I/O interfaces. This signal will be high when the address bus contains an address in the range of 0BE00-0BFFF and the I/O address space has been enabled. This signal is driven by the Monomeg and changes coincident with address bus changes.

6.3.3 Data Bus

The data bus consists of 8 lines designated DB0 - DB7 with DB0 being the least significant. These lines are buffered and are capable of driving up to 5 TTL loads (25 low power Shottky loads). They are tri-state as well which allows peripheral boards to drive them during read cycles. The Monomeg drives the Data Bus only during write cycles to off-board addresses. During read cycles in which the Monomeg addresses on-board functions, the data bus is ignored and during on-board write cycles, it is not driven. Interface boards should only use low power bus drivers such as the 74LS245 to avoid generating excessive system bus noise.

6.3.4 Read/Write

This signal is driven by the Monomeg and distinguishes between read and write bus cycles. During a bus read cycle, this signal will be high and valid coincident with the address bus. During a bus write cycle, it will be low and valid coincident with the address bus. The opposite polarity signal is also available.

6.3.5 Phase 2

This signal is driven by the Monomeg and is the master timing signal for bus data transfers. It has a 50% duty cycle and is derived directly from the Phase 2 output of the 6502 CPU chip to insure correct timing. Its frequency is jitter-free and is within .02% of 1.0MHz. The MTU-130 bus cycle consists of a setup portion and a data transfer portion. The setup portion occurs while Phase 2 is low and is when the address bus and read/write line may change. The data transfer portion occurs while Phase 2 is high. The address bus and read/write line is guaranteed to be stable during this time. Data is actually latched in the CPU, memory, or peripheral on the falling edge of Phase 2. The opposite polarity signal is also available.

6.3.6 RAM Write

This is a convenience signal that goes low only when Phase 2 is high and a bus write cycle is in progress. It is high at all other times. This signal may be used to trigger TTL output registers or may be connected directly to the Write Enable input of static RAM chips.

6.3.7 Reset

This signal goes low whenever the keyboard Reset key is pressed. It also is held low for approximately 1 second after power is applied to the MTU-130 system. This signal should initialize or reset everything on the interface board that must come up in a "known" or "safe" state. This line is driven by the Monomeg board.

6.3.8 Interrupt Request (IRQ)

This is a wire-or line that is connected to all devices capable of generating interrupts in the MTU-130 system. A 3K pull-up resistor is provided on the Monomeg. Any device requesting interrupt service should pull this line to ground with an open-collector TTL gate or an open-drain MOS output. The line should be held low until program interrupt service instructions clear the cause of the interrupt in the device or interface. Interrupt source identification is performed by software that checks the status of every device that might request an interrupt. This line may change state at any time.

6.3.9 Non-Maskable Interrupt Request (NMI)

This is a wire-or line that is connected to all devices capable of generating non-maskable interrupts in the MTU-130 system. A 3K pull-up resistor is provided on the Monomeg. Use of the non-maskable interrupt should be restricted to "panic" situations because proper re-entry into the interrupted program cannot be guaranteed if it uses the SVC facility of the CODOS operating system. The 6502 CPU responds to negative-going edges on this line and unconditionally interrupts the program when an edge is sensed. This line is connected to the INT key on the MTU-130 keyboard. This line may change state at any time. The minimum pulse width is 2 microseconds.

6.3.10 Set Overflow

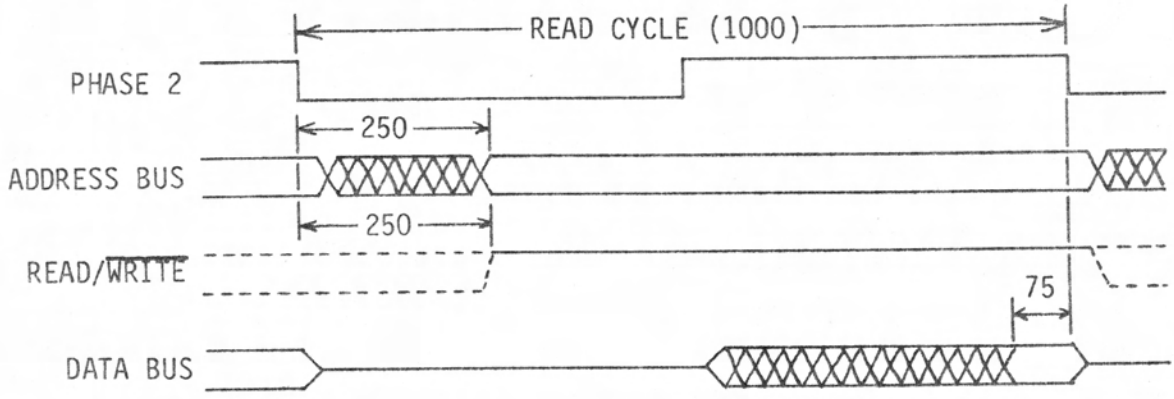
This is a wire-or line that is connected to all devices capable of setting the overflow flag in the 6502's status register. A 3K pull-up resistor is provided on the Monomeg. This signal is negative edge sensitive. When an edge is detected, the overflow (V) flag in the 6502's status register is unconditionally set. The flag will stay set until an arithmetic or pull status instruction changes it. This line may change state at any time. The minimum pulse width is 2 microseconds. This signal is not used in any way by the Monomeg CPU or any current or contemplated MTU peripherals and is completely free for use by the customer.

6.3.11 Missing Signals

The MTU-130 bus does not provide the 6502 Ready, Sync, or Phase 1 signals. Ready was omitted because all modern memories and I/O chips are quite fast enough to function without wait states. Sync is generally necessary only for interfacing "tricks" (such as the 18 bit address logic on the Monomeg). Inclusion of either of these signals would make the potential future design of a faster CPU board while retaining 1MHz bus timing nearly impossible. Phase 1 is so close to the inverse of Phase 2 that the latter can be used in any reasonable interfacing situation.

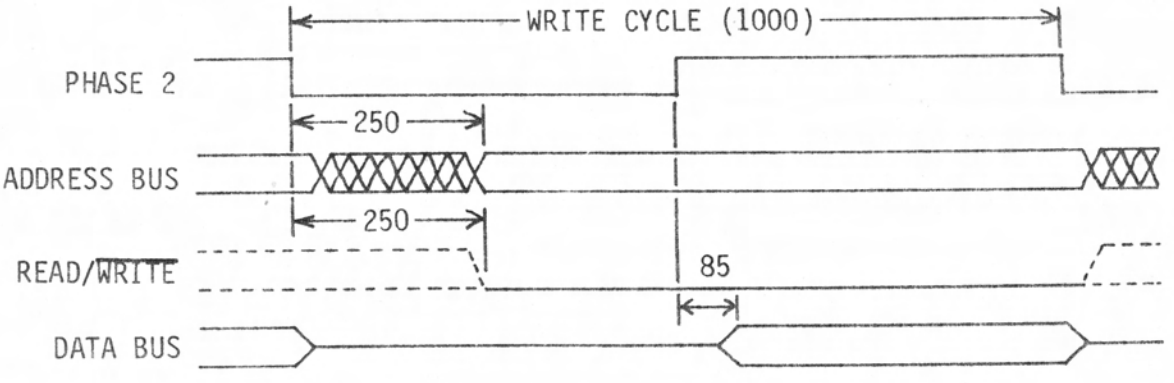
Timing relationships and requirements of the 6502 microprocessor used in the MTU-130 system are the simplest and least critical of any microprocessor. Three factors make this true. The first is that the timing is regular; all bus cycles are the same length and are always executed "bumper-to-bumper" with no dead time between them. The next is that only one timing signal and one control signal are used to coordinate bus operation and they do it in the simplest way possible. The last is that very little "happens" during a bus cycle which means that even though the cycle rate exceeds that of a 3MHz 8080 or Z-80 based system, noise generation is much less and timing tolerances are much greater. These factors just about guarantee a hassle-free bus interface design and debug cycle for the customer. In particular, the bus timing allows direct interface of all 650X and 680X family I/O ICs (except that data bus buffers are required).

The MTU-130 bus executes just two kinds of cycles, a read cycle and a write cycle. The timing for a read cycle is shown below:



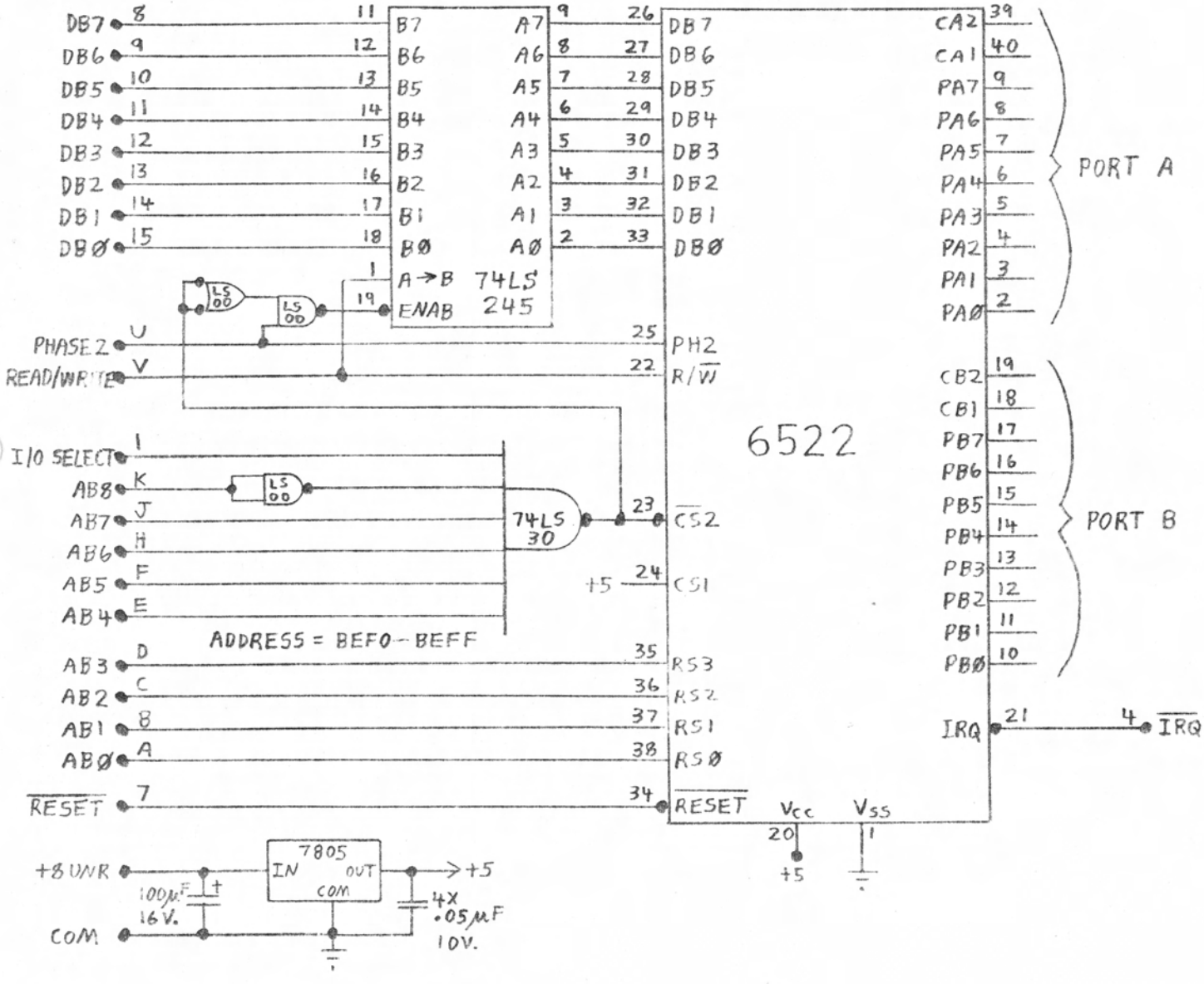
The read cycle starts (and the previous cycle ends) by Phase 2 going low. While Phase 2 is low, the address bus changes and settles to the address that the cycle will refer to. Simultaneously, the Read/Write line goes high (or stays high) to specify a read cycle. After 500NS Phase 2 goes high. The addressed device at this time is expected to drive the data bus with the data to be read. The data need not be valid on the bus until 75NS before Phase 2 goes low again which is 425NS after it goes high. The cycle ends with Phase 2 going low at which instant the CPU latches the data internally. The addressed device is expected to cease driving the data bus when Phase 2 goes low again. Data "hold time" after Phase 2 goes low is of no concern because buffering delays on the Monomeg insure adequate hold time to the 6502 even if the addressed device stops driving the data bus instantly when Phase 2 goes low.

The timing for a write cycle is very similar as shown below:



When a write cycle is executed, the Read/Write line goes low during the time Phase 2 is low. When Phase 2 goes high, the Monomeg starts driving the data bus with the data to be written. Within 100NS of Phase 2 going high, this data will be valid. The data remain valid on the data bus for the remainder of Phase 2 high time and for at least 20NS after Phase 2 goes low. The addressed device can successfully latch the data either with a negative edge-triggered clock connected to Phase 2 or with an active-high level triggered clock connected to Phase 2.

The schematic below shows an example MTU-130 interface circuit diagram. The device function is simply an interface to a 6522 Versatile Interface Module which in turn can be connected to other on-board user circuitry.



Example MTU-130 Bus Interface to 6522

7. PRINCIPLES OF OPERATION

This section describes the Monomeg circuitry block-by-block. The intention is to provide sufficient information for an experienced technician to understand and troubleshoot the Monomeg CPU board down to the component level. Additionally, it will serve to answer almost any detail level question that may arise about programming and interfacing the Monomeg or MTU-130. The timing and schematic diagrams that will be constantly referred to may be found in sections 9 and 13 respectively. All logic signal terminology and conventions to be used are consistent with "standard" usage in IC literature.

7.1 CLOCK AND SYSTEM TIMING GENERATION

All timing functions on the Monomeg are controlled ultimately by a single crystal controlled 10MHz oscillator which is shown at the bottom of page 5 of the schematic diagrams. The oscillator network using U121 (a 7404), resistors, and 10MHz crystal has been found to be highly reliable through years of experience and can be expected to give a jitter-free symmetrical square wave output. Three additional stages give true and complement polarities of the clock plus a delayed true version (U21-12, 2, and 4 respectively).

For system and memory timing generation, the non-inverted 10MHz clock drives U78 which is connected for divide-by-10 operation. This is accomplished by preloading the counter with a 3 and letting it count through 12 at which point it is reloaded with 3 to start another cycle. Nand gate U98-6 detects that the count is equal to 12 and causes the counter to reload a 3 on the next clock pulse. The system timing diagram shows that this count sequence results in a symmetrical 5MHz square wave at the A output and a symmetrical 1MHz square wave at the D output. The A output is used primarily by the video generator when in the half-resolution gray scale mode. The 1MHz D output is called PHASE 0 and goes directly to the 6502 CPU which uses it to generate the master PHASE 2 clock which is essentially a delayed (50NS) replica of PHASE 0. The B and C outputs have a scrambled appearance and are used in the memory timing circuitry to be described later.

7.2 MEMORY TIMING AND CYCLE CONTROL

Also at the bottom of page 5 is the memory cycle timing control. For memory timing purposes, the 1 uS system cycle period is divided into a 400NS CPU portion and a 600NS display portion. There either may or may not be a memory cycle executed in each portion. The CPU cycle, if taken, is always a single, full memory cycle. The display cycle, if taken, may either be a single cycle or a double (page mode) cycle. There are no special refresh cycles since the display cycles occur frequently enough and with a regular addressing pattern so that they serve to keep the memory refreshed. If a cycle is not taken, none of the memory ICs are clocked which minimizes power consumption. When a CPU cycle is taken, only the addressed row of 8 memory ICs is clocked. When a display cycle is taken, the display row and another row (in rotation) are clocked but the other 2 rows are not clocked. These measures keep system noise down.

Flip-flop U7-9 generates the RAS (Row Address Strobe) signal for CPU memory cycles. Note that it is connected upsidedown so the "off" state is set. Nand gate U79-8 conditions the flop to turn-on at the proper time if the RAM ADRD signal is true. RAM ADRD is generated by the address decoding circuitry and is true only if the 6502 CPU is addressing on-board RAM in its current cycle. Flip-flop U60-6 generates the CAS (Column Address Strobe signal for CPU memory cycles. It will follow U7-9 with a delay of about 50NS by virtue of being clocked by the opposite polarity of the 10MHz clock. The address multiplex signal for the RAMs follows RAS and is timed with normal logic delays to occur midway between RAS and CAS turning on as shown in the timing diagram.

The CPU RAS signal is terminated at the end of a CPU memory cycle by the network made of C7, R7, and R8. This turns-off U7-9 by coupling negative edges of PHASE 0 to the preset input. The address multiplex and CPU CAS signals turn-off in sequence later just like they were turned-on earlier. Since the 4116 type RAM chips retain valid data out until CAS is turned off, sufficient hold time at the end of the cycle for the 6502 to read the memory data is provided.

Timing generation for display cycles is somewhat more complex because of the provision for page mode cycles. Flip-flop U7-7 generates DISP RAS when conditioned to turn-on by Nand gate U79-6. This occurs at the proper time in the 1uS system cycle when enabled through pin 3 by the display sync circuits. Once triggered, DISP RAS always stays on for the full display cycle duration regardless of whether it will be a single cycle or a double cycle. After U7-7 turns-on, address multiplexing follows just as in CPU cycles. Flip-flop U60-8 begins the raw display CAS signal 50NS after DISP RAS begins and holds it until 50NS after DISP RAS ends. DISP RAS is turned-off at the proper time by U79-12 and the multiplex and raw display CAS signals follow in sequence.

In single display cycles, the raw display CAS signal simply pases through U53-6 and goes directly to memory row 3 which is the display memory. In double cycles, U77-6 triggers for 100NS which temporarily blocks the raw display CAS signal through U53-4. The resulting output at U53-6 is a double CAS pulse to the display RAMs which is interpreted as a page mode cycle. Decoder U3 in conjunction with the horizontal sync generator determines when double cycles are to occur. Only 2 out of 8 display cycles are double cycles which allows 10 bytes to be displayed in 8uS.

Circuitry for controlling the Write Enable signal to the RAMs is on schematic page 6. Writing is only allowed during CPU write cycles and thus the correct timing is simply the logical And of PHASE 2, CNTC, and the inverse of the 6502's R/W signal. Write enable for RAM row 2, which occupies addresses 08000-0BFFF, also factors in the WRITE ALLOW signal from one of the on-board 6522 I/O ICs.

This same page also shows circuitry for controlling the combination and distribution of CPU RAS and DISP RAS to the 4 rows of memory IC's. For CPU cycles, CPU RAS is directed to the addressed row according to the AB14 and AB15 address lines from the 6502 CPU. For display cycles, RAM row 3 always gets DISP RAS while one of the other rows also gets DISP RAS according to the state of display address bits 8 and 9. Since only row 3 also gets CAS during display cycles, only row 3 is fully activated; the other row is simply refreshed.

7.3 MEMORY ADDRESS MULTIPLEXOR

The memory address multiplexor is at the right edge of schematic page 2. In the MTU-130, the memory may receive addresses from two sources, the 6502 CPU and the display address counter. Further, since 4116 type RAMs are employed, the 14 address lines needed to address 16K bits must be multiplexed into the RAM's as two sets of 7 bits each. It is possible to combine these two multiplexing functions into a single level of 4-input multiplexors. The type 74LS153 dual 4-input multiplexors used have two selection inputs to select one of the 4 data inputs. One of these (SEL0, pin 14) is used to select between the row address (when high) and the column address (when low). The other (SEL1, pin 2) is used to select between display address (when low) and the CPU address (when high). The actual bit numbers used for row and column addresses were selected for successful memory refreshing by the display; logically it makes no difference as long as the selection is the same for CPU and display.

The bare outputs of the multiplexor ICs are not powerful enough to drive 32 memory ICs while the use of more powerful 74153s would overload the 6502's address lines too much for reliable operation. Thus an octal buffer (type 81LS96) is inserted between the multiplexors and the RAM ICs. Using inverting buffers delays the signal somewhat less and uses less power. The RAMs do not care about the polarity of their address inputs as long as it is the same for both write and read.

7.4 MEMORY ARRAY AND DATA BUFFERS

Page 3 of the schematics shows the memory array, CPU data buffer, and video data register. The 32 type 4116 memory ICs are arranged in 4 rows of 8 ICs each making the total array capacity 64K bytes. Logically all four memory rows are equivalent. Separation into 48K of main memory and 16K of display memory in a different bank is a function of external clocking and address decoding circuitry.

The data input and output lines of the RAM chips are separated in order to increase timing margins during write cycles. Since only the 6502 CPU can write into memory, the data input lines are tied directly to the 6502 data bus. The RAM data output lines go to the RAM Data Out Buffer and the Video Data Register. Since the memory cycle timing has been arranged to coincide with the 6502's bus cycle, data read during CPU cycles is gated directly onto the 6502's data bus through U74 which is an octal tri-state buffer. Display cycles are less well lined up with video circuit requirements so data read during display cycles is latched into octal latch U91. The output of this latch then goes to the video shift register where it is converted into serial form for the display.

7.5 HORIZONTAL COUNTER

The Horizontal Counter is shown on page 5 of the schematics. The purpose of the horizontal counter is to generate the horizontal sync frequency of 16,129 Hz and proper timing of horizontal sync pulses and horizontal unblank time. U2 is the counter proper and consists of a 6 bit counter arranged to divide the 1MHz PHASE 0 frequency by 62. This is accomplished by detecting the instant when the counter reaches 62 and then resetting the counter to 0. U8-6, U19-11, and the decoder U3 all participate in the detection of state 62.

Essentially the video generator runs with an 800NS cycle time while the display memory runs with a 1000NS cycle. Periodically this requires a double cycle from the memory to keep up with the display as described in section 7.1. Theorectically every fourth memory cycle should be a double cycle but the requirements of page mode cycles dictate that there be an even number of single cycles between double cycles. Thus double cycles are taken at intervals of 3, 5, 3, 5, ... (thus giving 2, 4, 2, 4, ... single cycles between them) as shown in the Horizontal Display Period timing diagram. This gives an alignment recurrence period of 8uS. Decoder U3 decodes this alignment recurrence period and a number of display timing signals are derived from it.

The actual horizontal sync pulses are derived by decoding the horizontal counter state with nand gate U19-3. Dual flip-flop U9 also participates in producing the correct sync duration. A sync pulse width of 4uS is produced by using a 4uS period square wave from the horizontal counter to clock U9. Actually, U9-6 generates an "inverse" horizontal sync and U9-9 generates the regular horizontal sync signal. The inverse signal is only used during the vertical retrace interval. Besides being of inverted polarity, it leads the regular horizontal sync signal by 4uS. This is important to maintain proper horizontal synchronization during vertical retrace.

Individual pixels within a display cycle time are counted by the least significant bit of U78 in the System Timing section and by U4-10 and 11 in the Horizontal Timing section. Together these function as a divide-by-8 counter and count out the 8 pixels in a display memory byte. Nand gate U5 detects when it is time to reload the video shift register with the next 8 pixels. It also recognizes the active horizontal and vertical display periods and suppresses shift register loading during blanking times. Note that at the end of every horizontal period the pixel counter just described is resynchronized to the horizontal counter by being reset at the same time.

7.6 VERTICAL COUNTER

The vertical counter functions in a manner similar to the horizontal counter except that it is triggered by inverse horizontal sync pulses. The vertical counter divides the 16,129 Hz horizontal frequency by 269 to obtain 60Hz. This is accomplished by a 9 bit counter that is allowed to count up to 269 and then is promptly reset to 0. Nand gate U20-8 detects the 269 state and generates the reset pulse. Since there are exactly 256 active scan lines, the VERTICAL ENABLE signal is simply the time that the nineth counter bit is off (U110-6). The 13 cycles when it is on is the vertical retrace time. Nand gate U81-12 is satisfied for 4 horizontal cycles immediately after VERTICAL ENABLE turns off and generates the actual vertical sync pulse. Starting vertical sync immediately at the end of the visible sweep gives the display monitor maximum time for vertical retrace. The remaining circuitry around the vertical counter is part of the composite video generator described in paragraph 7.8.

7.7 VIDEO SHIFT REGISTER

The video shift register can be found at the bottom left corner of schematic page 5. It is a type 74LS166 8 bit synchronous load shift register. It is constantly clocked by the 10MHz pixel clock. Displayed bits are shifted out and zeroes are shifted in. Every 8 pixel times it is enabled to load a new set of 8 bits to display. Ahead of the video shift register is the video data register which is on schematic page 3. This register acts as a 1 stage first-in-first-out buffer to match the somewhat erratic read rate of data from display memory to the constant data rate required by the shift register. Video retrace blanking is accomplished by stopping the shift register load pulses. This works because zeroes continue to be shifted in which are converted into a black video level.

7.8 COMPOSITE VIDEO SIGNAL GENERATOR

The composite video signal generator is at the right side of schematic page 5. When the video circuits are in black and white (2 level) mode, three different video voltage levels must be generated. This is accomplished with a discrete digital-to-analog converter and the converted voltage appears on the node connected to R51 and R26. Sync level is about .2 volts and is forced whenever open-collector nand gates U58-6 or U58-8 turn on. U58-6 turns on for normal horizontal sync pulses and U58-8 turns on for inverse horizontal sync pulses during the vertical sync pulse. When in black and white mode, only nand gate U96-6 is enabled. Serial video data is delayed one clock pulse by flip-flop U110-9 and passes through U96-6 and U96-11 to drive the video summing node through R57. Black pixels tend to pull the summing node voltage down to about +1.1 volts while white pixels pull it up to about +2.1 volts.

In gray-scale (4 level) mode, the two flip-flops in U97 in conjunction with U110-9 serve to convert the 10MHz serial bit stream into a 2 bit wide 5MHz parallel stream. U96-6 is disabled and U96-3 and U111-8 are enabled when in gray-scale mode. The gate outputs together with R57 and R58 form a 2-bit weighted resistor digital-to-analog converter. The black video levels is +1.1 volts as before while the dim level is +1.6, normal is +2.1, and the bright level is +2.6 volts.

Q2 functions as an emitter follower and serves to convert the relatively high D-to-A converter node impedance into a low impedance for driving the video cable. Series resistor R45 provides an effective source impedance very close to 75 ohms. For practical purposes the emitter follower has a unity gain but it does shift all of the video voltages up about +.7 volts.

7.9 DISPLAY ADDRESS COUNTER

The display address counter counts out display memory addresses and is located at the bottom of schematic page 2. It is essentially a resettable 14 bit ripple counter made with 74LS393 dual 4 bit counters. It is counted up by negative edges of DISP A CNT which is generated by the Memory Timing Generator. DISP A CNT is really the same as CAS during display cycles so the address counter counts at the end of every display memory cycle (each page mode cycle if a double cycle). The counter is reset by DISP A CLR which is the same signal that resets the vertical sync counter.

7.10 LIGHT PEN INTERFACE AND REGISTERS

One of the unique features of the Monomeg CPU is the inclusion of light pen registers that latch sufficient information to calculate light pen position to the pixel. Schematic page 6 shows the initial signal conditoning circuitry for the output of the off-board light pen amplifier. Schmidt trigger inverters are used to insure a fast risetime signal since the lightpen amplifier output has a risetime of approximately 100NS for a solid hit and somewhat slower for light seen at the edge of its field of view.

The sharpened light pen pulse then goes to the left edge of schematic page 2 into the K input of flip-flop UU77-9. This flop serves to synchronize the light pen signal to the 10MHz pixel clock. The flop is constantly clocked by the pixel clock and if the light pen signal is present, it will be reset on the next clock pulse. Since it is a J-K type of flip-flop, it will then remain reset regardless of what the light pen does until the program recognizes the hit and sets it. The $\overline{Q}$ output of the flop, which is now synchronized with the pixel clock, triggers U20 which latches the 3 bit pixel counter value. The fourth bit, which is the hit flag the CPU sees, is unconditionally set high at the same time. The $\overline{Q}$ output also goes to the right edge of the page and triggers a flip-flop that latches the state of PIX4. The latched PIX4 signal later becomes the skew flag.

In order to reliably latch the display memory address, the light pen hit signal must be further synchronized to the 1uS system cycle. This is accomplished by U64-5. This flop is constantly clocked by PHASE 0 and synchronizes the pixel aligned hit signal from U77-9 to the positive edge of phase 0. Its Q output thus synchronized triggers latches U34 and U47 which latch the state of the display address counter and the skew flag described earlier.

Each of these 3 registers has tri-state outputs. This allows them to be connected directly to the CPU bus and be read individually by the program. The pixel synchronizer flip-flop and the pixel register (which includes the hit flag) may be cleared at any time with one of the "bit I/O" strobes generated under program control.

7.11 6502 CPU

The circuitry around the 6502 CPU is actually one of the simpler aspects of the Monomeg. It occupies the premiere position at the top left of schematic page 1. Clocking is accomplished simply by connecting its phase 0 input to the symmetrical 1MHz PHASE 0 signal generated by the system timing generator. This clock input signal is absolutely regular with no "stretching" of any kind ever. The 6502 uses this signal to generate the "real" PHASE 2 clock. Both true and complement PHASE 2 are derived from the 6502's Phase 2 output and sent to other circuitry and the bus with very little differential skew.

The 5 control inputs are disposed of very simply. RDY is not used so it is simply pulled up with a resistor. Set Overflow is likewise pulled up and then connected to the bus for external use. The IRQ (Interrupt ReQuest) is also pulled up and then distributed to on-board devices capable of generating interrupts and to the bus connector. The NMI (Non-Maskable Interrupt) input is pulled up and goes to the bus as well. The keyboard INT key is debounced with R52, R55, and C107 then squared up by U94-10 and inverted by U43-8. CR5 simulates an open-collector output for U43-8 thus allowing the keyboard interrupt to be connected to the NMI input.

The debounce network for KEYB RST serves the dual purpose of debouncing and as a power-on reset. R53, R56, and C106 form a delay network with a very slow rise time and somewhat slow fall time due to the large size of C106. When power is off, C106 discharges through CR9 into the 5 volt bus which would be a ground potential. When power is applied, C106 charges very slowly through RR53 and R56. During this charging interval, a low level is held on the Reset input to the 6502 as well as other portions of the Monomeg. After approximately 1 second, the positive-going threshold of U94 is reached and the reset signal is released. Pressing the keyboard RESET key will also discharge C106 thus simulating a power off-on cycle. Because of R53 the discharge requires about 1/4 second which is sufficient to "lock out" unintentional depression of the RESET key.

The address and data busses of the 6502 are sufficiently powerful to drive everything on-board the Monomeg without special buffering. Of course they are buffered before going off-board as described in paragraph 7.13. The read/write line however is buffered and inverted and connects to other on-board circuitry and the bus. The SYNC output is used only on-board and does not require any buffering.

7.12 ADDRESS DECODING

The primary address decoding circuits are on schematic page 2. One-of-4 decoder U38-4 looks at address bits AB16 and AB17 to determine the bank that is addressed. Another one-of-4 decoder U38-12 looks at AB14 and AB15 to determine the 16K block that is being addressed. These decoded signals in conjunction with other individual address bits are used to generate the various select signals in the Monomeg.

I/O SELECT should go high only when Bank 0 is selected, the address is in the range of BE00-BFFF and I/O is enabled. This is accomplished by nand gates U37 and U8-12. Flip-flop U56-6 is the I/O enable flip-flop. The I/O SELECT signal then goes to the bus and to U13-8 where it is further qualified to determine if any on-board I/O (which occupies BFC0-BFFF) has been addressed. If on-board I/O has been addressed, one-of-4 decoder U40-12 subdivides it into four sets of 16 addresses each; one for each of the 3 6522s and the last for miscellaneous.

The I/O enable flip-flop (U56-6) is clocked by a write cycle to either FFFE or FFFF in bank 0. When clocked, it is set to the state of address AB0. System reset forces it into the I/O enabled state.

U53-8 determines when RAM on-board the Monomeg is addressed. The RAM is addressed if Bank 1 and Block 3 is addressed (display memory) or Bank 0 and not I/O and not Block 3 (main memory). On-board ROM is addressed if Bank 0 and Block 3 is addressed and the ROM is enabled. Flip-flops U63-9 and U56-9 delay the bus/ROM signal from one of the I/O chips by one instruction time by virtue of being clocked by SYNC from the 6502 which pulses on every op-code fetch. A single gate at the bottom of schematic page 1 activates the external bus if nothing on-board is addressed.

7.13 EXTERNAL BUS BUFFERS

In order to drive the large system configurations that 18 bit addressing implies, the 6502 address and data bus lines are buffered on the Monomeg before going off-board. This circuitry is shown at the lower right corner of schematic page 1. The address buffers for the lower 16 address bits are U16 and U45 which are octal buffers. Since the Monomeg is the only source of addresses in the MTU-130 system, these buffers are always enabled. The extended memory address bits need no separate buffering since they are generated by TTL circuitry.

U30 is the data bus buffer which is an octal transceiver. The read/write line from the 6502 controls the transceiver's data flow direction: drive out on write cycles and drive in on read cycles. The transceiver is not activated however unless nothing on-board is addressed (as determined by U61-6) and then only during PHASE 2 high time. Both the address and data buffers are low power devices which reduce system noise generation while still providing adequate drive for the full 5 slot backplane bus.

7.14 EXTENDED ADDRESSING

One of the truly unique features of the Monomeg CPU is its extended addressing capability up to 256K using a standard 6502 microprocessor. Extended addressing essentially works by defining three different "segements" (or banks) called System, Program, and Data. The Program and Data segments can be assigned to any of the four 64K blocks of addresses in the system independently. The System segment is always assigned to the lowest 64K of addresses. The segment actually used for a 6502 bus cycle depends on the instruction being executed and other factors. The System segment for example is used for all base page and stack references. It is also selected whenever interrupts are serviced. The Data segment is used only for the data transfer cycles of indirect-X and indirect-Y instructions. The Program segment covers all other references which includes normal instruction fetch and the absolute addressing modes. Even the base MTU-130 system requires proper functioning of the extended addressing circuitry because the display memory is located in bank 1.

Circuitry for implementing the extended addressing is in the right half of schematic page 1. Most of the work is performed by a 256x4 bit ROM (U55) but some logic is needed to provide its inputs. The ROM output is the segment number to be used in the current cycle with 00 specifying Data, 01 specifying Program, and 10 or 11 specifying System. The segment number then drives the select inputs of a dual 4-input multiplexor (U1) which gates the selected bank number from one of the on-board I/O ports to the bus as AB16 and AB17. Since system reset clears all of the on-board I/O to ones, bank 0 (after inversion by U17-6 and 8) will be selected for all 3 segments which gives standard 6502 operation.

The remaining circuitry is concerned with generating the 8 input signals the ROM needs to decide which segment is being referenced. The least significant 3 bits (U55 5,6,7) expect the execution cycle number of the currently executed instruction. This is provided by counter U42-11 which counts up at the beginning of every 6502 bus cycle. Since it is reset by SYNC from the 6502 , which goes high during the fetch cycle of every instruction, its count will reflect the execution cycle count. Note that the counter will be 0 for both the fetch cycle and the first execution cycle, so the count is truly the execution cycle number.

ROM address bit 3 (U55-4) indicates whether the current cycle is referencing addresses between 0000 and 01FF. Since these addresses are page zero and the stack, they have special significance. Gates U44-6 and U8-8 generate a 0 if these special addresses are being referenced.

The next 3 ROM address bits give information about the actual operation-code of the instruction currently executing. This information is latched by U54 when the instruction is fetched and persists until the next instruction is fetched. ROM A6 tells whether the instruction itself was fetched from page 0 or 1. ROM A4 and A5 have a code that classifies the type of instruction being executed. A code of 00 identifies an instruction using the indirect-X addressing mode while a code of 02 identifies indirect-Y addressing. Code 11 identifies the absolute addressing mode and code 10 is used for all other instructions that do not fall into these classes. Decoder U14 and gates U12-3, U12-11, and U13-6 look at the data bus to make this determination and latch U54 registers the decision on the trailing edge of Phase 2 when SYNC is active.

The last ROM address bit is connected to the Interrupt Mode flip-flop. This flip-flop is normally off but is on whenever an interrupt service routine is in control. The flop must turn on when an interrupt acknowledge sequence is executed by the 6502 and turn back off when a return from interrupt (RTI) instruction is executed. Unfortunately, the 6502 has no specific interrupt acknowledge pin. Fortunately, the internal acknowledge sequence is unique in that it is the only operation that produces 3 write cycles in a row. Counter U43-3 counts consecutive write cycles and sets U57-10 when 3 in a row have occured. The flip-flop is reset by recognizing the RTI instruction op-code with gates U44-5 and U12-8.

7.15 ROM SOCKETS

The Monomeg CPU has 4 ROM/EPROM sockets included to make it suitable for applications where external mass storage to load programs into RAM is not desired. Fundamentally, these are very simple consisting of four 24 pin sockets with the 12 address lines, 8 data lines, and 2 power lines connected in parallel and to the internal address and data buses as shown on schematic page 1. The Output Enable lines of all 4 are also paralleled and connected to Phase 2. The Chip Enable lines are individually connected to outputs of the address decoder on page 2. If the ROM is enabled and addressed as determined by U61-8, then decoder U40-4 decodes the the top 16K of Bank 0 into 4 blocks of 4K each for the ROM sockets. This two-level method of enabling the ROM ICs ensures minimum power consumption by inactive ROMs and minimum noise generation by delaying driving the data bus until all other on-board circuitry has been thoroughly deselected. These sockets are intended only for type 2732 EPROMs or type 2333 masked ROMs. Other EPROM or ROM types are either missing the Output Enable function or have non-standard pinouts. Type 2716 may be used by making an adapter socket which removes pin 21 from the monomeg pin 21 and connects it to pin 24.

7.16 I/O CHIP ADDRESSING

I/O functions on-board the Monomeg use 64 addresses in the range of BFC0 to BFFF in bank 0. Fourty-eight of these 64 addresses are assigned to 3 type 6522 I/O interface ICs at the rate of 16 addresses each. These three 6522s may be found on schematic page 4. Each one has 4 address inputs as well as an active-high chip select and an active-low chip select. The active-high chip select is always enabled and the active-low chip select is connected to the address decoder. Since the 6522 is a 650X family member, its clock and read/write lines may be connected directly to PHASE 2 and read/write on the 6502.

The fourth group of 16 addresses is divided equally between a 6551 serial communications IC and a TTL based "bit I/O" circuit. This circuitry may be found near the middle of schematic page 2. Each function actually uses only 4 addresses but the decoding is incomplete so in effect each uses 8. Decoding to split the group of 16 addresses into two groups of 4 and associated images is done entirely by appropriate connection of the multiple chip select inputs.

7.17 CASSETTE INTERFACE

The cassette interface consists of signal conditioning circuits connected to two port bits on one of the system 6522's. The purpose of the cassette interface is to provide convenient data interchange with other systems that may lack a disk or serial communications capability. Since no motor control is provided, the cassette interface is not really suitable as a substitute for disk based mass storage.

The cassette write circuit may be found in the middle of schematic page 2. Flip-flop U59-9 changes state whenever a particular bit in the bit I/O circuit is addressed (see paragraph 7.21). The flip-flop output is attenuated to 1 volt and amplitude stabilized by the voltage divider formed by R36 and R37. Capacitor C100 attenuates high frequency energy above 10KHz thus preventing interference with the recorder's bias frequency. Series resistor R46 allows the user to simply parallel a low value (100 ohm) resistor with the recorder input if a low-level (mic input) signal is needed. It has little effect when the signal is plugged directly into a high level (aux) input.

The cassette read circuit is at the bottom left of schematic page 4. Basically it is a zero-crossing detector connected to port B bit 7 on the SYS2 6522. Capacitor C4 DC isolates the input signal and attenuates low frequency (hum) signal components. R4 and C5 attenuate high frequency (noise) signal components. R5, R6, CR1, and CR2 form a limiter that prevents saturation of the LM339 comparator. The signal is limited to about .5 volts peak-to-peak when the input level is 1 volt peak-to-peak or greater. Comparator U6 is connected as a sensitive comparator with about 50MV of hysteresis supplied by R63 and R2. Resistor R3 connected to -12 volts makes the two thresholds (+25MV and -25MV) symmetrical around ground since the comparator's output swings only between +5 volts and ground.

7.18 KEYBOARD INTERFACE

Since the keyboard is software scanned, its interface circuitry is actually quite simple. On the external keyboard itself is a 4 bit counter and one-of-16 decoder that drives the 16 columns of the keyswitch matrix. The 6 rows of the keyboard come on-board the Monomeg and are connected directly to port A bits 2-7 on the SYS1 6522. The only control signals needed by the counter are Clock and Reset. Count is supplied by CA2 on the SYS1 6522. CA2 can be programmed to generate a pulse whenever port A is read thus providing for automatic stepping to the next column when a scan routine looks at the rows. Reset is provided by one of the bit I/O outputs.

Also associated with the keyboard interface is signal conditioning for the INT, RESET, BREAK, and MOD keys. The INT and RESET circuitry is described in paragraph 7.11. The BRK key is debounced by R13, R14, and C17 in the upper left corner of page 4. Schmidt trigger U18 then provides a clean edge to the CA1 input of the SYS1 6522. Under program control, the CA1 input can be made to generate an interrupt when the BRK key is pressed.

The MOD key debouncing circuit is fairly complex. During power-up, the MOD key must appear to be depressed so that the bootstrap load program will perform a "cold reset" sequence. However after the operating system has been loaded, the MOD key should act normally without any significant activate/deactive delay. During power-up, C13 is initially discharged and keeps the voltage at the base of Q1 near ground through CR3. During an approximately 2 second period, C13 charges through R12 and CR3. After C13 is charged and Q1's base is at a high voltage, pressing the keyboard MOD key (which grounds the KEY MOD signal) will instantly ground the base of Q1 with no delay and without disturbing the charge on C13. When power is removed, C13 quickly discharges through CR4. Q1 is an emitter follower which drives Schmidt trigger gate U18-4. This allows C13 to be much smaller than would otherwise be necessary. C14 prevents oscillation in the Schmidt trigger gate which can be a problem when driven by a high AC source impedance.

7.19 AUDIO D-A CONVERTER

The audio digital-to-analog converter is shown at the bottom right corner of schematic page 4. The entire circuit consists of an integrated 8-bit digital-to-analog converter connected to port A of the SYS2 6522, a sharp cutoff low-pass filter, and a power amplifier stage capable of 1 watt output. In addition, a path is provided to the CB2 pin of the user 6522 for generating square wave "CB2 sound".

U62 is a standard DAC08 type of integrated circuit DAC chip selected for its low cost and ability to provide a voltage output without an OP-amp current-to-voltage converter. A regulated, noise-free reference current into the REF+ input is provided by zener diode CR8, R21, and C74. R31 supplies operating current to the zener regulator. The reference current may be modulated about 50% above and below its .9MA quiescent value by the CB2 output of the user 6522 through R64, R65, and C143. The series capacitor only allows the AC component of the CB2 output from affecting the reference current. This setup allows the DAC to be used as a programmable volume control when using CB2 as an oscillator yet at the same time allows normal audio DAC operation when CB2 is not active. The user may disable CB2 sound altogether by grounding the CB2 OFF signal in the user parallel connector. Regardless of the source, the audio voltage is developed by the IOUT+ DAC output current flowing through R22. At maximum signal level, this is about 2 volts peak-to-peak with the positive peak at +9.1 volts.

The low-pass active filter is a 6-pole 1/2dB Chebyshev design, or in other words, a very sharp 60dB per octave (a Butterworth of equivalent complexity would only give 36dB per octave). It is realized as three sections of 2 poles each cascaded where each section is a Sallen and Key VCVS configuration. U92-8 is the lowest Q section and has a unity gain. Each of the other two sections has a gain of +2. Since the gain determining feedback network to the inverting inputs is returned to +9.1 volts instead of ground, the positive peak of the signal remains at +9.1 volts but now extends down to +1 volt because of the gain of 4. Note that if it becomes necessary to replace any of the capacitors in the filter (C73, 75, 99, 101, 102, 103) that they should be replaced with 5% polystyrene capacitors for best sound quality. C75 and C99 are especially critical since they are in the highest Q section.

The power amplifier stage consists of op-amp U92-1 and a complementary pair of booster transistors connected as emitter followers. The entire stage has unity gain and a low output impedance suitable for driving an 8 ohm speaker. Resistor R50 between the bases and emitters eliminates crossover distortion at high signal levels and low load impedances by filling in somewhat the "dead band" that would otherwise exist at zero crossing time. When driving a high impedance load (higher than 2K), the signal simply flows through R50 and Q3 and Q4 never turn on at all. C133 is the output coupling capacitor and is made large enough to give good bass response (down to 30Hz) when an external speaker is used. R43 and C127 suppress possible oscillation when inductive loads are driven. The output transistors should be replaced with the same type or with type MPSU01 (NPN) and MPSU51 (PNP) power transistors, both of which have high current gain (100) at high currents (1 amp) in a small package.

7.20 SERIAL INTERFACE

The serial interface is shown at the bottom of schematic page 2. At its heart is a type 6551 Serial Communications Interface Adapter with internal programmable baud rate generator. The baud rate generator requires a 1.84MHz input frequency. Since this is not simply related to the 10MHz system clock frequency, a separate oscillator is used. U75-2 in conjunction with ceramic resonator Y1 and R34 make a 3.68MHz oscillator. Resonators (or quartz crystals) for the 3.68MHz frequency are much smaller and less expensive than would be required for the 1.84MHz frequency. Dual capacitor C76 is required for oscillation and also temperature compensates the ceramic resonator. Flip-flop U59-6 divides the 3.68MHz oscillator frequency by 2 to produce the required 1.84MHz clock.

Standard serial interface logic levels are -12 volts for a logic 1 and +12 volts for a logic zero. Line receiver and level shifter U76 receives the serial input data and CTS, DSR, and DCD signals from off-board and converts them the TTL compatible levels required by the 6551. R18, 19, and 20 bias the line receviers for CTS, DSR, and DCD so that if they are not driven by external signals, they will "float" to true levels. U93 translates the TTL level data, RTS and DTR signals generated by the 6551 into high voltage bipolar signals for the interface. U93 is actually a quad op-amp chosen for its low power consumption and controlled risetime characteristics. R35 and R42 set a +2.5 volt input threshold as the op-amp acts like a comparator. R39-R41 set an output impedance of 300 ohms.

7.21 INTERNAL BIT I/O

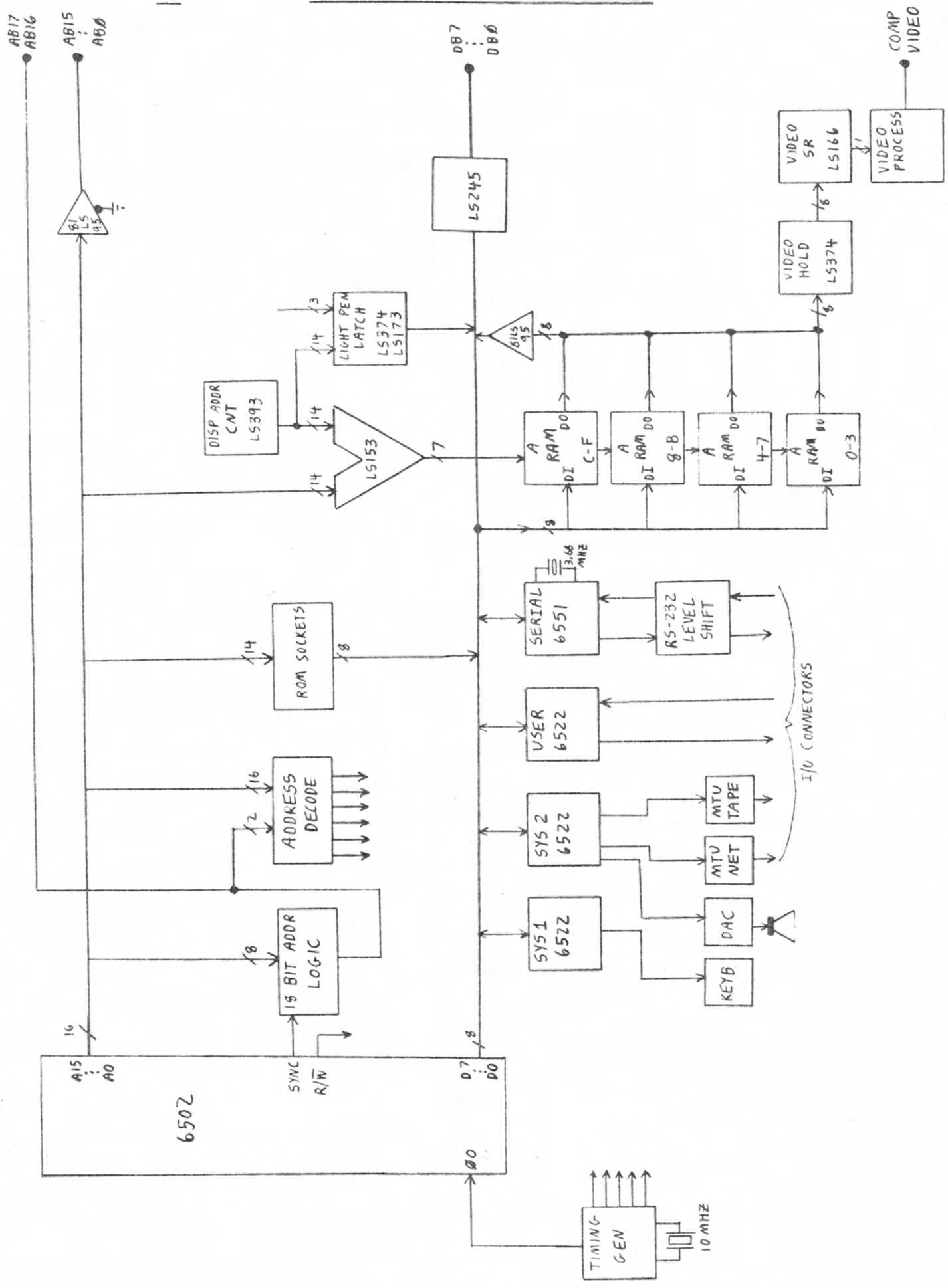
A number of I/O functions on-board the Monomeg either require simply triggering something or require reading a read-only register like one of the light pen registers. For these purposes, 4 output strobes and 4 input enables are generated by one-of-8 decoder U21. Outputs 0-3 are .5uS wide strobes that are generated when one of the 4 bit I/O addresses is written to. One of these resets the light pen, another clears the keyboard scan counter and another toggles the cassette output. Outputs 4-7 go low for .5uS when one of the 4 bit I/O addresses is read. Three of these are used to read the 3 light pen registers. There is one spare output strobe and input enable signal left over for use in testing.

7.22 POWER SUPPLY

The Monomeg CPU is designed to be operated from +12, +5, and -12 volt regulated power supplies. These input voltages are simply decoupled with 100uF capacitors at the boards edge fingers and then distributed to on-board circuitry. The 4116 memory ICs however require a -5 volt input at very low power which is derived from the -12 volt input with a zener diode regulator. Additionally, the audio D-to-A circuit requires +9.1 volts which is produced with another zener diode regulator.

8.

MONOMEG CPU BOARD BLOCK DIAGRAM



TIMING DIAGRAMS

FIGURE 9-1 OVERALL SYSTEM TIMING

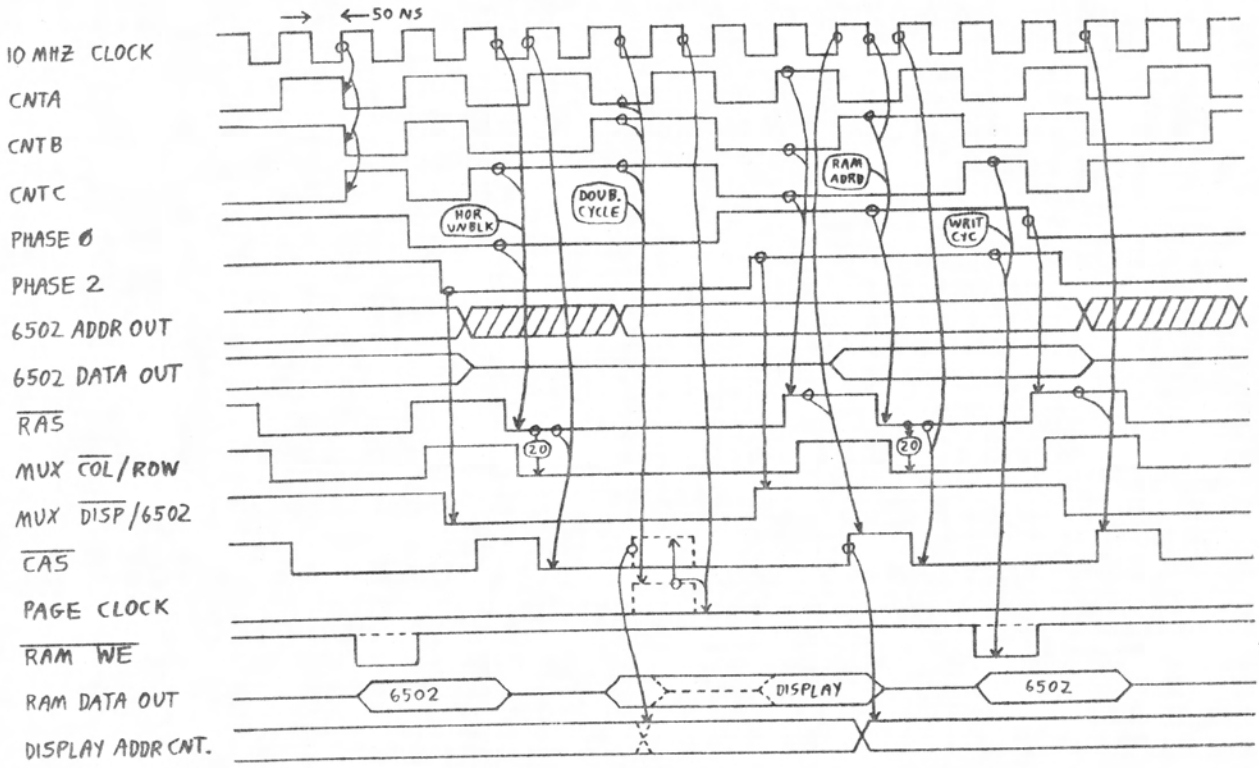


FIGURE 9-2 VIDEO RECURRANCE PERIOD TIMING

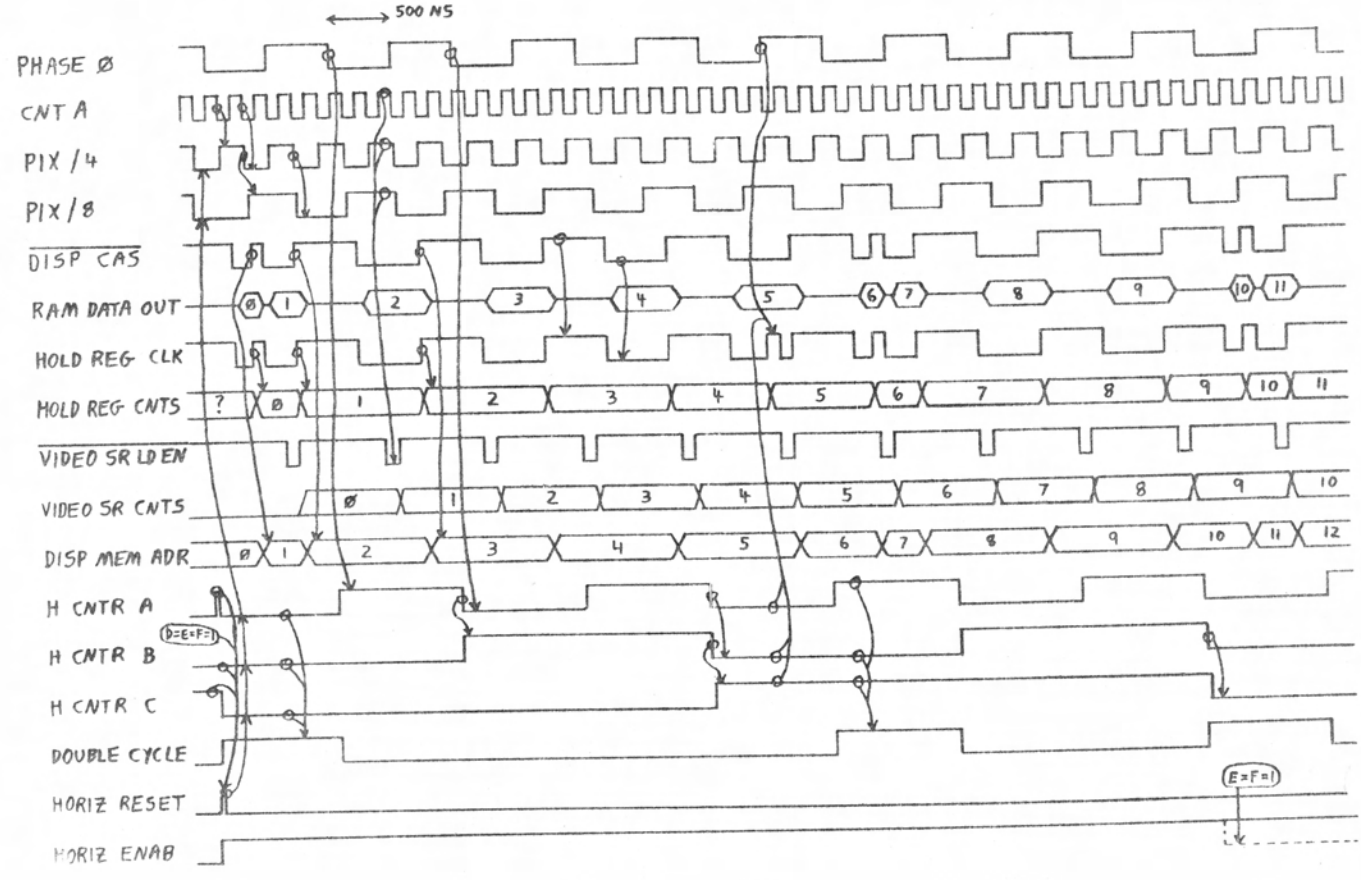


FIGURE 9-3 HORIZONTAL DISPLAY PERIOD TIMING

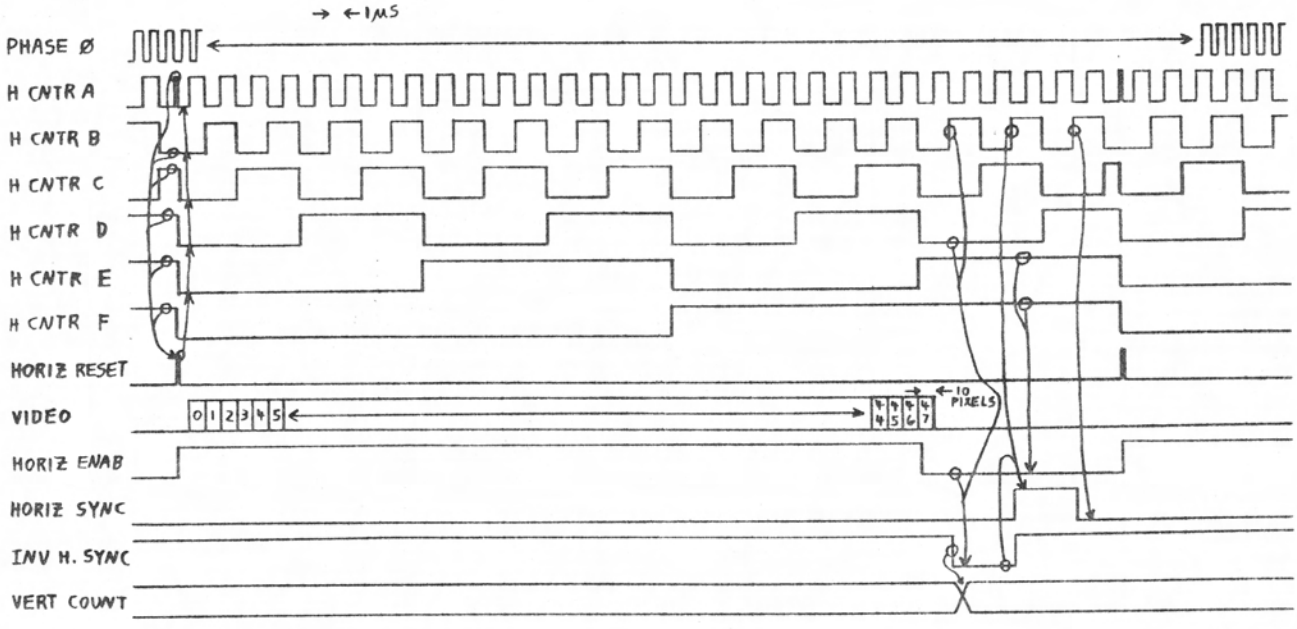
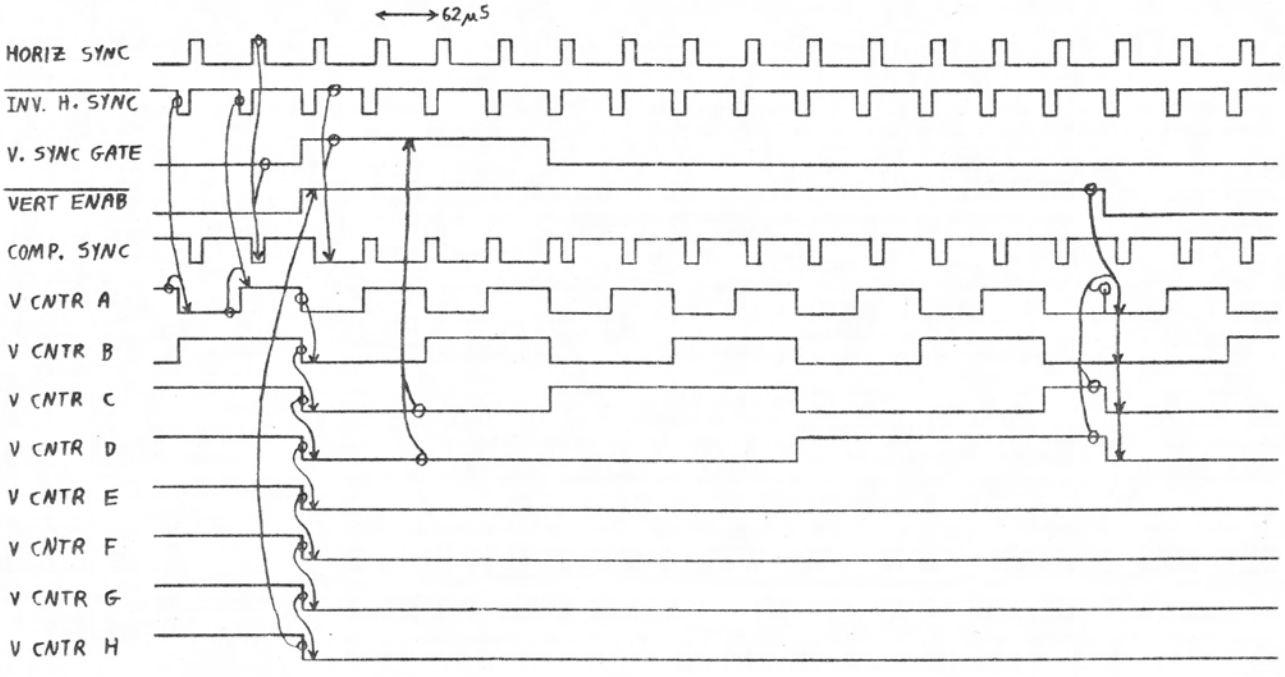


FIGURE 9-4 VERTICAL SYNC DETAIL



10. TROUBLESHOOTING GUIDE

The following discussion is intended to outline a "method of attack" for locating and correcting malfunctions on the Monomeg CPU board. It is assumed that the Disk Controller and other bus interface boards in the MTU-130 have at least tentatively been ruled out as the cause of system malfunction. It is also assumed that the Monomeg has functioned satisfactorily in the past. The initial steps require only the use of a volt meter while the later ones involve use of an oscilloscope.

SUPPLY VOLTAGE CHECK

The first troubleshooting check is always to check the power supply voltages actually reaching the Monomeg board. Be careful not to let the voltmeter probes slip while making these measurements. First check the 5 volt supply accross any of the TTL IC's on the board. U1 (a 74LS153) is easiest to reach when the Monomeg and disk controller are installed into the MTU-130 chassis. Be suspicious if the voltage is less than +4.7 or more than +5.2 volts. If the voltage is low, check the power cable and connectors. If it is zero or too high, the 5 volt regulator in the power supply may have failed. +12 volts is most easily checked on pin 4 of U92 (a TL084 op-amp). Be very careful not to let the probe slip! The voltage reading should be between +11.5 and +12.5 volts. -12 volts can be found on pin 11 of the same IC and its acceptable range is -11.5 to -12.5 volts. If any of the voltages are out of range, find the cause and correct before continuing.

GROSS TIMING CHAIN CHECK

The general function of the timing curcuits can be quickly checked by connecting a video monitor to the video output and then powering up the system. From a cold start like this, the monitor display should be stable (after horizontal and vertical hold controls are adjusted on the monitor) and should display several somewhat irregular vertical bars. If there is no display at all, either the 10MHz crystal clock oscillator has failed, one of the counters in the timing chain has failed, or the some component in the video generator has failed. Check each of these areas in the sequence just mentioned.

If there is a display, examine it closely for evidence of horizontal or vertical jitter. If such jitter is seen, one of the timing chain counters is perhaps miscounting or misresetting. If the overall image is stable but some of the pixels (dots) are turning on and off, the problem is most likely in the memory array itself or the circuitry that times the RAS, address multiplex, and CAS memory timing signals. If none of the display defects discussed here are seen, the timing chain circuits are most likely functioning correctly

CPU FUNCTION CHECK

It is assumed here that the system gives absolutely no response to the keyboard Reset and Mod keys. If the disk drive loads its head when the Reset key is pressed and released (with the MOD key down), then the CPU is probably functioning and the problem is in the disk system. For looking at the CPU signals, it will be necessary to temporarily remove the Monomeg board and plug it into one of the upper slots above the disk controller board (leave the disk controller installed). You need reconnect only the audio/video connector (J6) and the keyboard connector (J3). Now apply power and check the logic level at pin 40 (Reset) of the 6502 microprocessor. It should rise suddenly to more than 3 volts about 1 second after power is applied and should also follow the action of the Reset keyboard key. With an oscilloscope, look at the PHASE 2 output of the microprocessor on pin 39 of U29. It should be a 1MHz square wave with good logic levels. If absent, look at the PHASE 0 input on pin 37. If that is also absent, there is a problem in the timing chain.

Now look at the SYNC output on pin 7. This will pulse high for 1uS whenever an instruction is fetched. If it is not pulsing, press Reset (with the MOD key doun) and release. If SYNC does not pulse even once, the 6502 chip is probably bad. If it pulses even once then stops, the microprocessor is getting garbage on its data bus which can be due to either a bad IC elsewhere loading the data bus or the address bus or to a non-functional disk controller board (see Disk Controller manual). Remember that the bootstrap load ROM is on the Disk Controller board. With a scope, look at each of the 16 address bus pins (U29 9-10 and 22-25). Each address line should swing smartly between near ground and +3 volts (substantial overshoot above 3 volts up to 6 volts is normal) with changes occuring approximately 100NS after the PHASE 2 output takes a high-to-low transition. If an address line is not changing, repeatedly press Reset until a change is seen. If no change can be obtained and the line is solidly at 0 or +5 volts, suspect a short. If the address lines seem OK, examine the data bus lines (U29 26-33). It is helpful to synchronize the scope to PHASE 2 since the data bus signals are significant only when PHASE 2 is high. Towards the end of PHASE 2 high time, the logic levels on each data bus line should be either near ground or greater than +3 volts (again substantial overshoot is normal). If an intermediate voltage level or logic level change is seen within 75NS of the end of PHASE 2, then some system component is not responding to its address or two components are responding simultaneously and the address decoding circuitry should be checked. It is also possible for a failed IC connected to the data bus to be loading it excessively. If there is any possibility that a board has been removed or installed in the MTU-130 while power was on, be sure to check the address, data, and timing signals that connect to the bus on all boards for electrical damage.

VIDEO SYSTEM

If the timing chain is functioning but the video output is weak, distorted, or absent, then the video output transistor (Q2) may have been blown from external voltages (such as a CRT arc-over). With a scope look at the signal at its base (U58-8 is the same signal) which should be a 2 volt amplitude video signal. If the signal is OK, look at the video output signal to the monitor (with monitor disconnected); it should be the same but shifted up +.7 volt. If either test fails, replace Q2. If there is still no video signal, there could be trouble in the sync pulse generation circuitry.

If the image intermittantly tears or shifts horizontally or shifts vertically and the monitor can be ruled out as the cause, then replacing one of the following counter ICs may solve the problem: U35, U46, U2, U108. They are all the same type (74LS393) so you can try swapping them around to see if the symptom changes before ordering new ones.

AUDIO SYSTEM

The audio power amplifier transistors Q3 and Q4 are not protected from direct short circuits to the audio output jack. If this jack has been used and there is now no audio output, then these transistors may have been blown. Start the music demonstration program and look at the voltage on the collector tab of either transistor (they are tied together). If it shows a sound waveform, the problem is off-board. If it is constant at about +5 volts, then the problem is further back in the audio circuitry. Follow the audio signal through from U62-4 to U92-8, U92-14, U92-7, and U92-1 to determine where it becomes lost. If there is no signal even on U62-4, then check the bit inputs to the DAC chip (U62). If none of the 8 bits are changing, there may be a problem with the 6522 I/O chip (U33) that drives it or a problem with address decoding. Note that a slight raspiness and lack of very high frequencies in the reproduced sound is normal since 8 bit D-to-A conversion at 8KHz sample rate used by the MTU-130 to reproduce sound is not really "high fidelity". Also the relatively powerful amplifier and speaker used in the MTU-130 may cause internal components to rattle when music is played at high volume levels.

11.

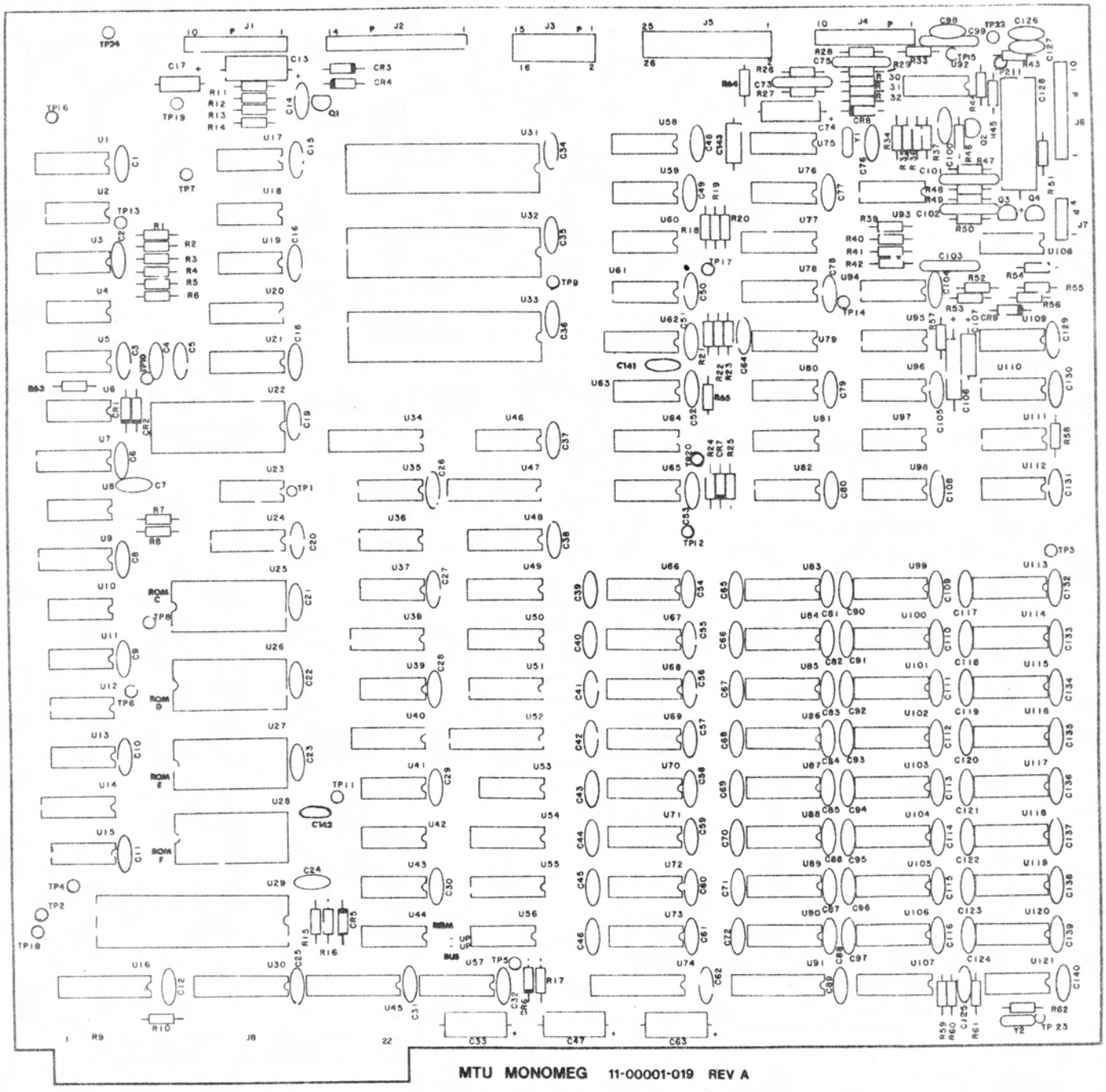
MONOMEG CPU BOARD PARTS LIST

QTY	COMPONENT	COMPONENTS DESIGNATION(S)
4	LOGIC,74LS00	U12,19,96,98
8	LOGIC,74LS04	U10,15,17,39,43,80,95,112
2	LOGIC,74LS08	U11,111
3	LOGIC,74LS10	U8,79,81
2	LOGIC,74LS14	U18,94
3	LOGIC,74LS20	U13,61,109
1	LOGIC,74LS26	U58
3	LOGIC,74LS30	U5,36,37
1	LOGIC,74LS42	U14
3	LOGIC,74LS51	U53,65,82
7	LOGIC,74LS74	U56,59,60,63,64,97,110
4	LOGIC,74LS109	U7,9,57,77
2	LOGIC,74LS138	U3,21
2	LOGIC,74LS139	U38,40
5	LOGIC,74LS153	U1,48-51
1	LOGIC,74LS161	U78
1	LOGIC,74LS166	U107
1	LOGIC,74LS173	U20
1	LOGIC,74LS174	U54
1	LOGIC,74LS245	U30
1	LOGIC,74LS260	U44
3	LOGIC,74LS374	U34,47,91
7	LOGIC,74LS393	U2,4,23,35,42,46,108
1	LOGIC,7404	U121
4	LOGIC,81LS95	U16,45,52,74
1	LOGIC,4069 CMOS	U75
32	LOGIC,4116 16K RAM	U66-73,83-90,99-106,113-120
2	LOGIC,6301 256X4 PROM	U24,55
1	LOGIC,6502 MICRO	U29
3	LOGIC,6522 I/O	U31-33
1	LOGIC,6551 COMM	U22
1	LOGIC,LM339 OP	U6
2	LOGIC,TL084 OP	U92,93
1	LOGIC,DAC0808 DAC	U62
1	LOGIC,1489 DRVR	U76
50	SOCKET,PC,14 PIN	XU2,4-6,8,10-13,15,17-19,23, 35-37,39,41,42-44,46,53,56, 58-61,63-65,75,76,79-82, 92-98,108-112
54	SOCKET,PC,16 PIN	XU1,3,7,9,14,20,21,24,38,40, 48-51,54,55,57,62,66-73,77, 78,83-90,99-107,113-121
8	SOCKET,PC,20 PIN	XU16,30,34,45,47,52,74,91
4	SOCKET,PC,24 PIN	XU25-28
1	SOCKET,PC,28 PIN	XU22
4	SOCKET,PC,40 PIN	XU29,31-33
1	CONN.HDR,SIL 4PIN	J7
3	CONN.HDR,SIL 10PIN	J1,4,6
1	CONN.HDR,SIL 14PIN	J2
1	CONN.HDR,DIL 16PIN	J3
1	CONN.HDR,DIL 26PIN	J5

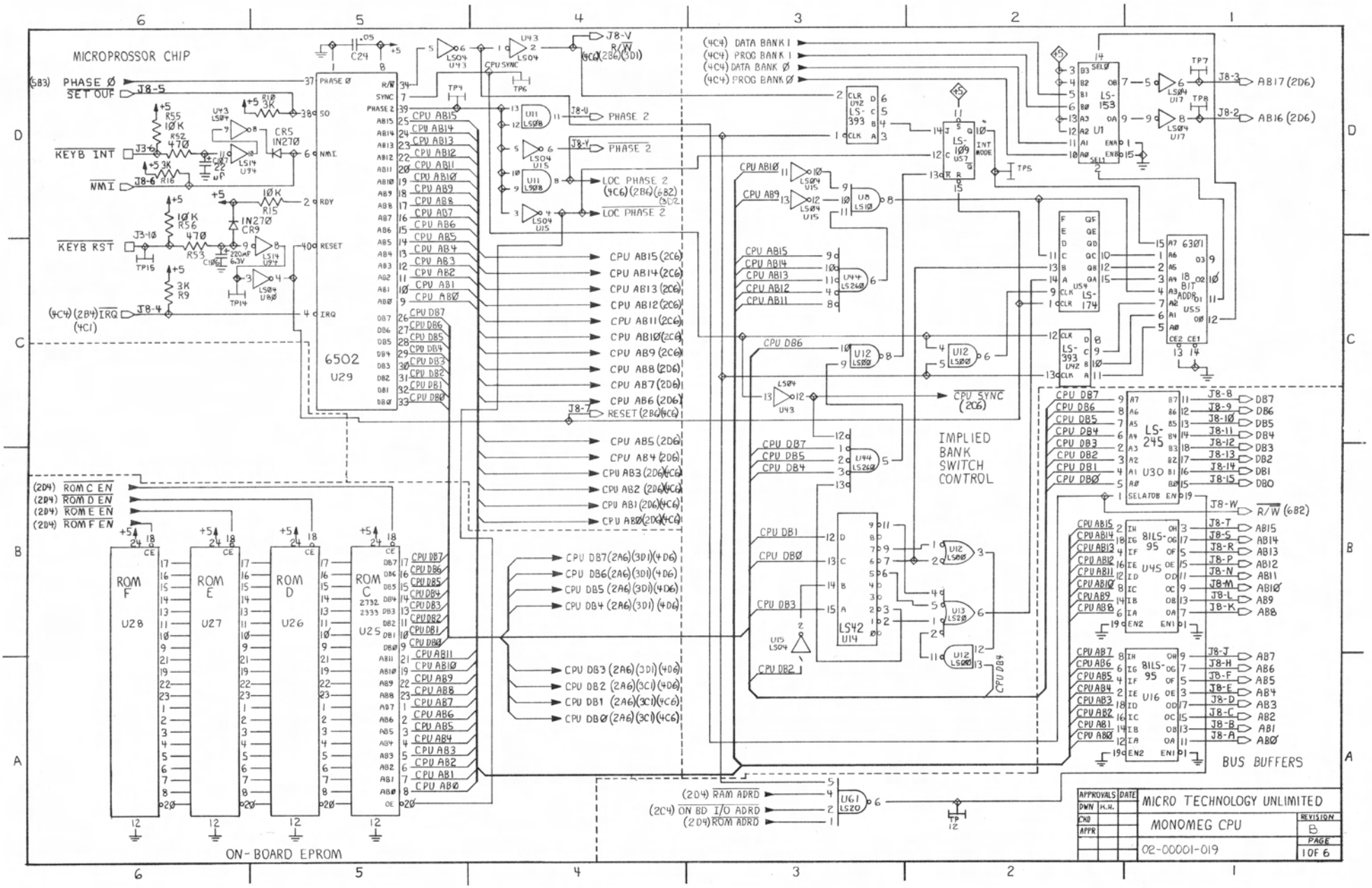
QTY	COMPONENT	COMPONENTS DESIGNATION(S)
1	RES,2.7 1/4W 5%	R43
2	RES,68 1/4W 5%	R44,45
3	RES,220 1/4W 5%	R50,59,61
3	RES,330 1/4W 5%	R39-41
5	RES,470 1/4W 5%	R2,14,31,52,53
1	RES,560 1/4W 5%	R60
4	RES,1K 1/4W 5%	R11,17,54,65
3	RES,1.6K 1/4W 5%	R51,57,62
7	RES,2.2K 1/4W 5%	R1,4,7,22,24,25,37
4	RES,3K 1/4W 5%	R9,10,16,26
2	RES,3.3K 1/4W 5%	R8,58
2	RES,4.7K 1/4W 5%	R5,36
1	RES,6.8K 1/4W 5%	R64
12	RES,10K 1/4W 5%	R6,13,15,18-20,21,35,42,46,55,56
1	RES,16K 1/4W 5%	R48
5	RES,27K 1/4W 5%	R29,30,32,33,47
1	RES,36K 1/4W 5%	R28
2	RES,47K 1/4W 5%	R12,63
1	RES,62K 1/4W 5%	R23
1	RES,68K 1/4W 5%	R49
1	RES,100K 1/4W 5%	R27
1	RES,240K 1/4W 5%	R3
1	RES,1M 1/4W 5%	R34
1	CAP,DISK,NPO 68PF 12V	C142
2	CAP,DISK,Y5F 100PF 12V	C7,64
1	CAP,DISK,Y5F 1000PF 12V	C141
2	CAP,POLY,1000PF 12V	C73,99
4	CAP,POLY,2000PF 12V	C75,101-103
2	CAP,DISK,Z5U .01UFD 12V	C100,5
120	CAP,DISK,Z5U .05UFD 12V	C1-4,6,8-12,14-16,18-32,34-46, 48-62,65-72,77-98,104,105, 108-127,129-140
3	CAP,ELECT,22UFD 10V AX	C17,107,143
5	CAP,ELECT,100UFD 16V AX	C13,33,47,63,74
1	CAP,ELECT,220UFD 6V AX	C106
1	CAP,ELECT,1000UFD 10VAX	C128
1	CAP,DUAL,RESONATOR COMP	C76
1	TRANS,PN2222	Q1
1	TRANS,PN2907	Q2
1	TRANS,PU01	Q3
1	TRANS,PU51	Q4
4	DIODE,1N270	D1,2,5,9
3	DIODE,1N4148	D3,4,7
1	DIODE,ZENER 5% 5.1V	D6
1	DIODE,ZENER 5% 9.1V	D8
1	CRYSTAL,3.68MHZ RESON.	Y1
1	CRYSTAL,10.0MHZ	Y2
23	TEST POINTS, .062	TP1-23
1	PCB, MONOMEG	

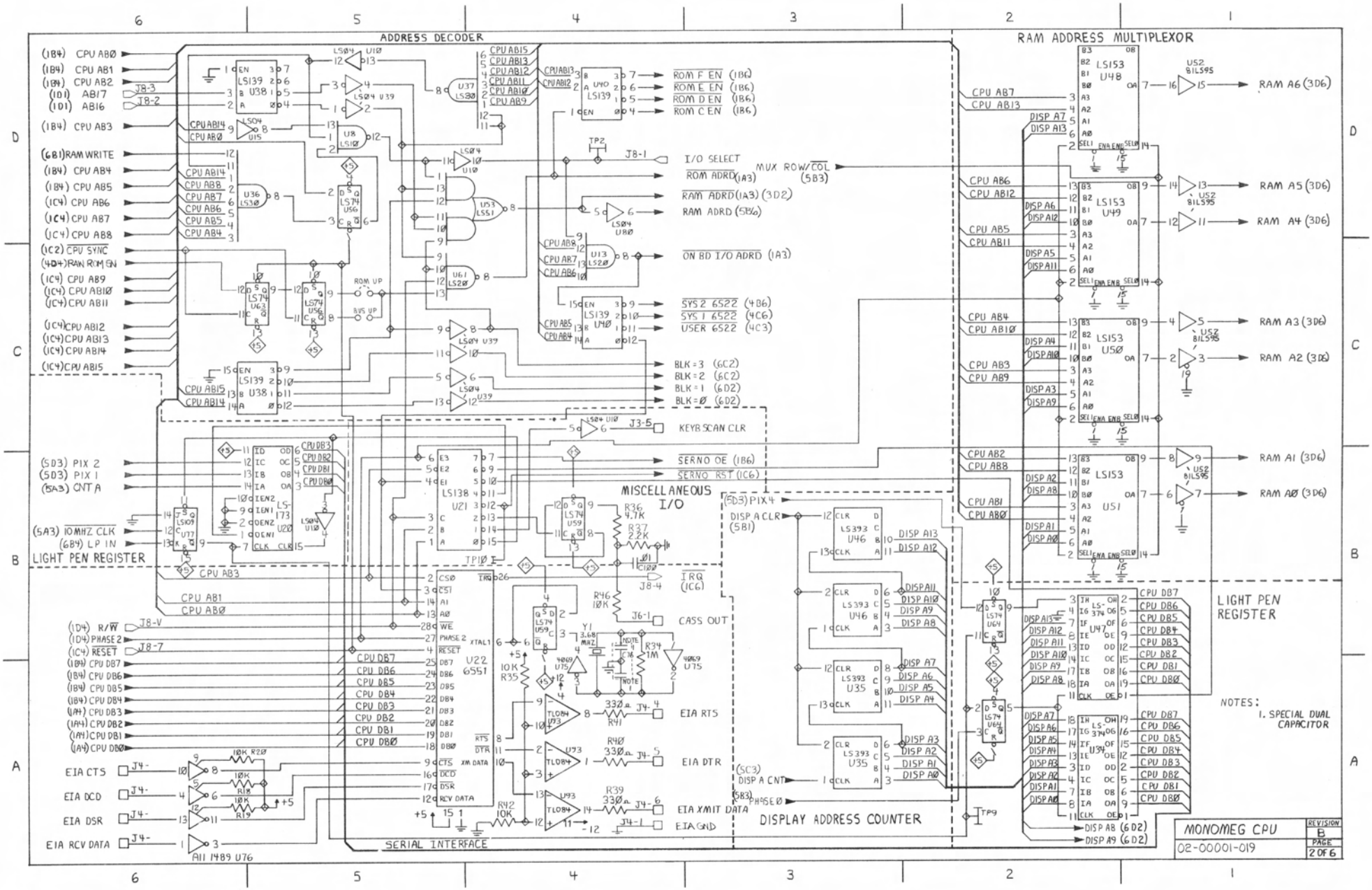
12.

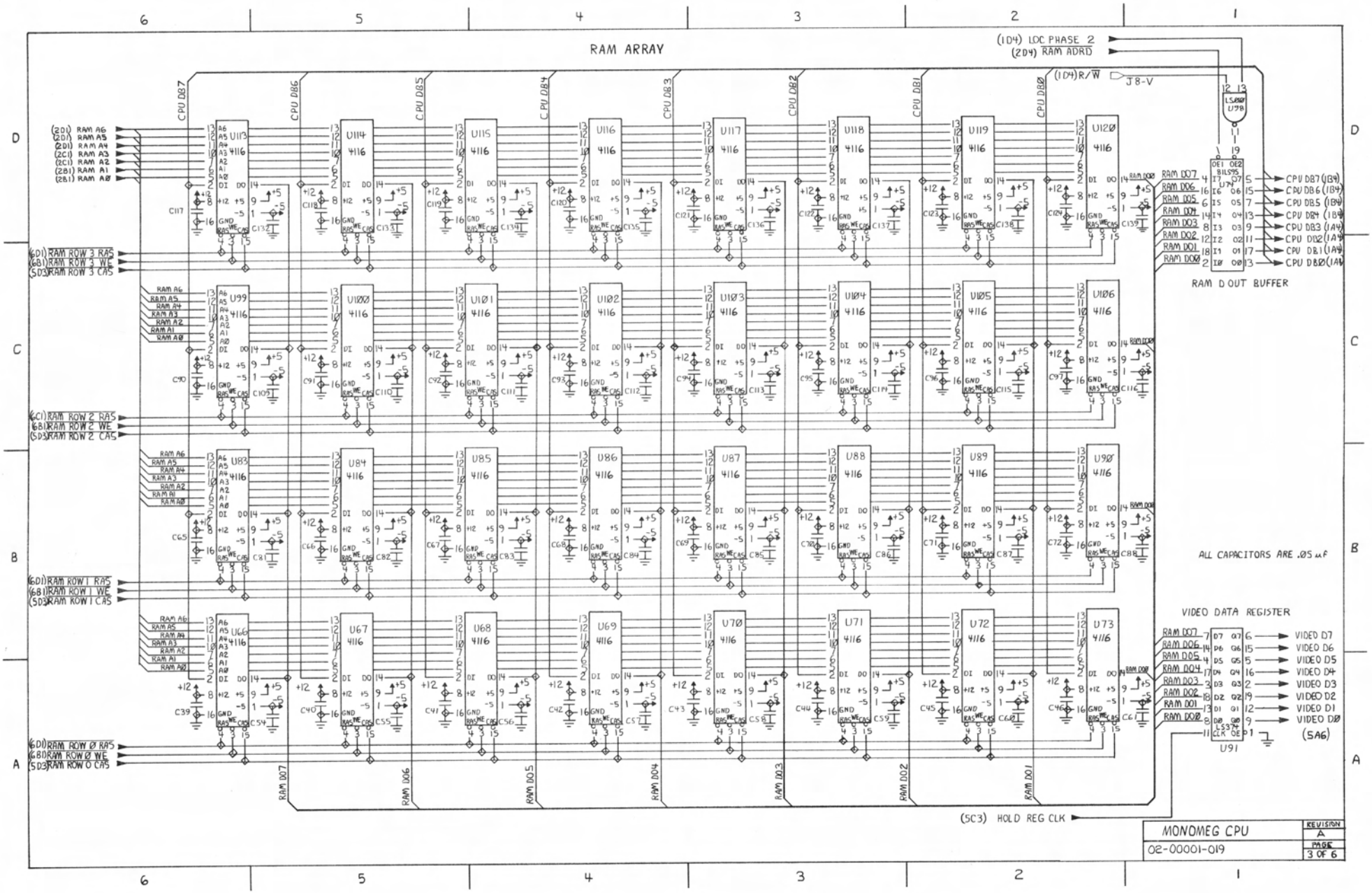
MONOMEG CPU BOARD PARTS LAYOUT

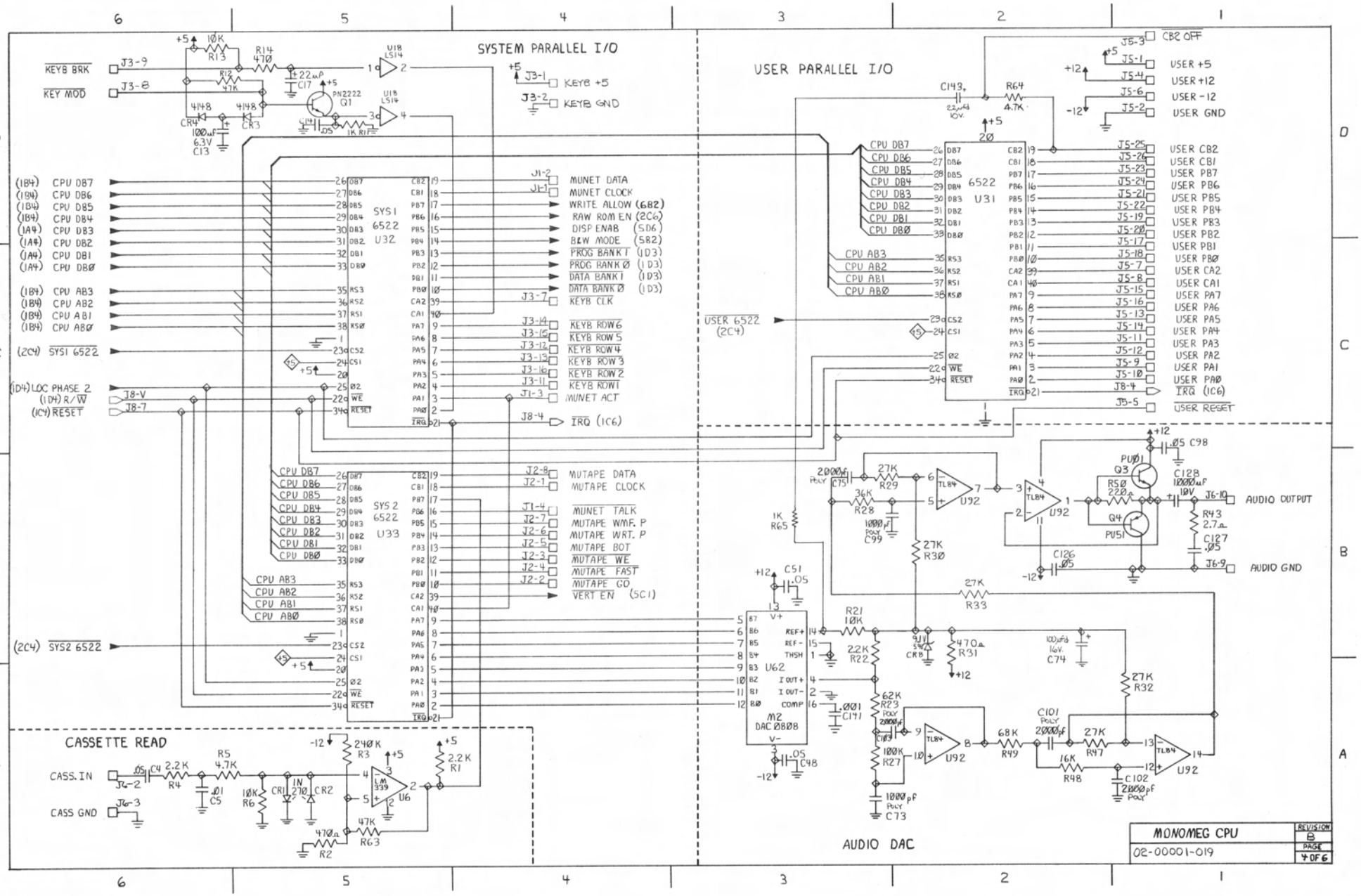


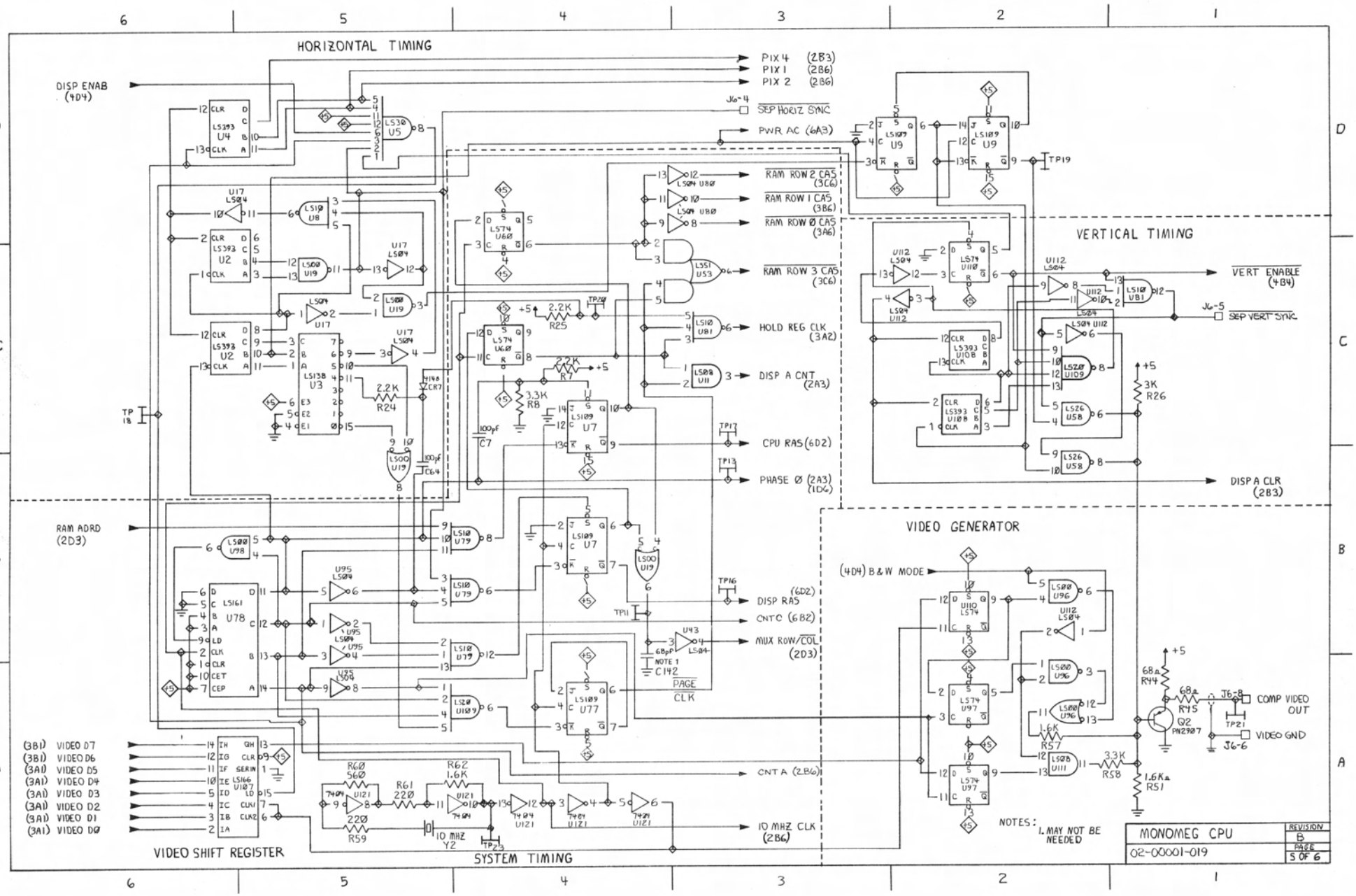
13. MONOMEG CPU BOARD SCHEMATIC DIAGRAMS

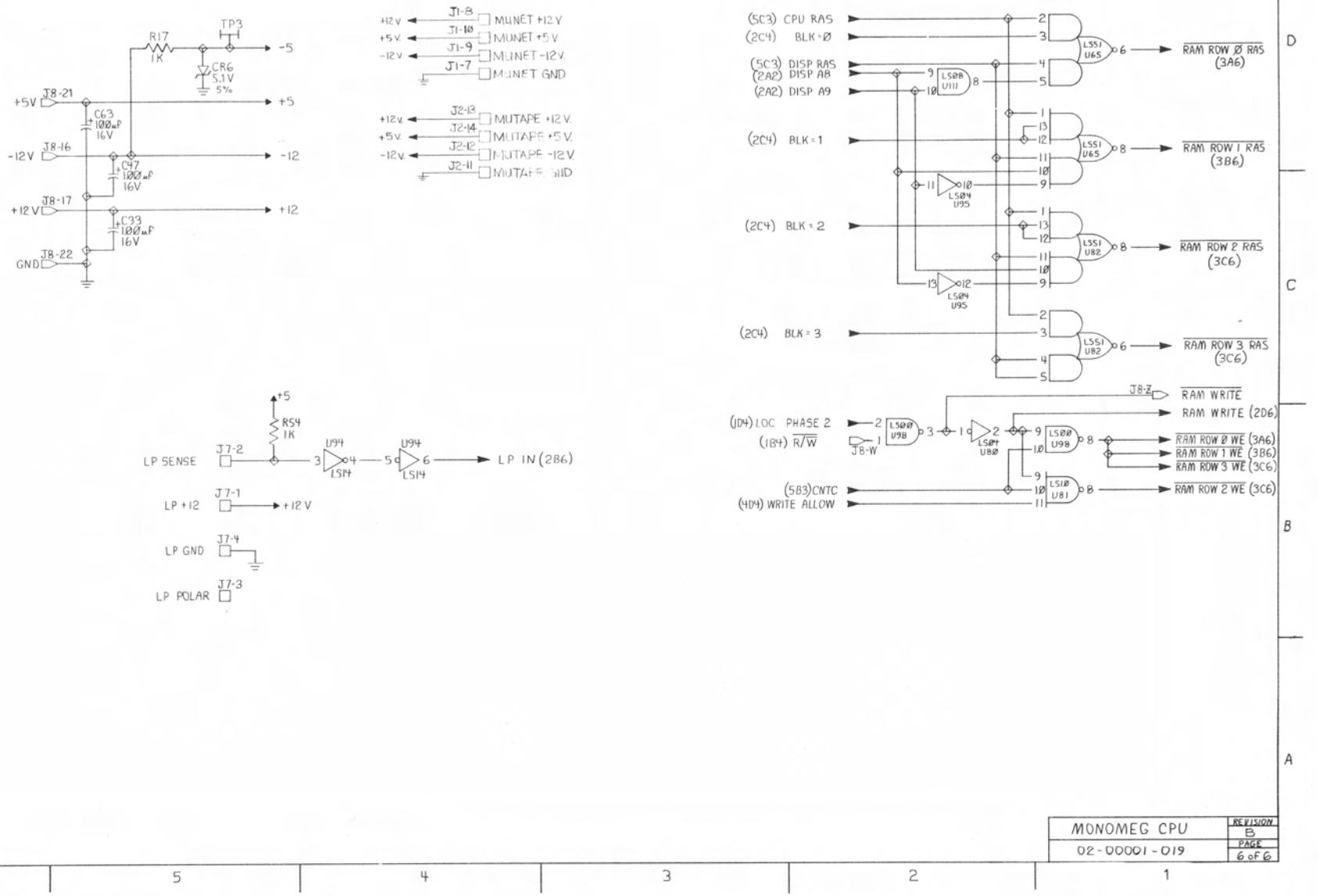

MICROPROSSOR CHIP
6502 U29
PHASE Ø
SET OUF
KEYB INT
NMI
KEYB RST
IRQ
CPU AB15 (2C6)
CPU AB14 (2C6)
CPU AB13 (2C6)
CPU AB12 (2C6)
CPU AB11 (2C6)
CPU AB10 (2C6)
CPU AB9 (2C6)
CPU AB8 (2D6)
CPU AB7 (2D6)
CPU AB6 (2D6)
RESET (2B6)(4C6)
CPU AB5 (2D6)
CPU AB4 (2D6)
CPU AB3 (2D6)(4C6)
CPU AB2 (2D6)(4C6)
CPU AB1 (2D6)(4C6)
CPU AB0 (2D6)(4C6)
PHASE 2
LOC PHASE 2 (4C6)(2B6)(6B2)
LOC PHASE 2
(4C4) DATA BANK 1
(4C4) PROG BANK 1
(4C4) DATA BANK Ø
(4C4) PROG BANK Ø
CPU SYNC (2C6)
IMPLIED BANK SWITCH CONTROL
ROM C EN
ROM D EN
ROM E EN
ROM F EN
ROM F U28
ROM E U27
ROM D U26
ROM C 2732 2333 U25
CPU DB7 (2A6)(3D1)(4D6)
CPU DB6 (2A6)(3D1)(4D6)
CPU DB5 (2A6)(3D1)(4D6)
CPU DB4 (2A6)(3D1)(4D6)
CPU DB3 (2A6)(3D1)(4D6)
CPU DB2 (2A6)(3C1)(4D6)
CPU DB1 (2A6)(3C1)(4C6)
CPU DB0 (2A6)(3C1)(4C6)
ON-BOARD EPROM
RAM ADRD
ON BD I/O ADRD
ROM ADRD
BUS BUFFERS
MICRO TECHNOLOGY UNLIMITED
MONOMEG CPU
02-00001-019
REVISION B
PAGE 1 OF 6


ADDRESS DECODER
RAM ADDRESS MULTIPLEXOR
MISCELLANEOUS I/O
LIGHT PEN REGISTER
SERIAL INTERFACE
DISPLAY ADDRESS COUNTER
LIGHT PEN REGISTER
MONOMEG CPU
02-00001-019
REVISION B
PAGE 2 OF 6
NOTES:
1. SPECIAL DUAL CAPACITOR


RAM ARRAY
(1D4) LDC PHASE 2
(2D4) RAM ADRD
(1D4) R/W
J8-V
CPU D87
CPU D86
CPU D85
CPU D84
CPU D83
CPU D82
CPU D81
CPU D80
(2D1) RAM A6
(2D1) RAM A5
(2D1) RAM A4
(2C1) RAM A3
(2C1) RAM A2
(2B1) RAM A1
(2B1) RAM A0
(6D1) RAM ROW 3 RAS
(6B1) RAM ROW 3 WE
(5D3) RAM ROW 3 CAS
(6C1) RAM ROW 2 RAS
(6B1) RAM ROW 2 WE
(5D3) RAM ROW 2 CAS
(6D1) RAM ROW 1 RAS
(6B1) RAM ROW 1 WE
(5D3) RAM ROW 1 CAS
(6D1) RAM ROW 0 RAS
(6B1) RAM ROW 0 WE
(5D3) RAM ROW 0 CAS
RAM D07
RAM D06
RAM D05
RAM D04
RAM D03
RAM D02
RAM D01
RAM D00
RAM D OUT BUFFER
CPU D87 (1B4)
CPU D86 (1B4)
CPU D85 (1B4)
CPU D84 (1B4)
CPU D83 (1A4)
CPU D82 (1A4)
CPU D81 (1A4)
CPU D80 (1A4)
ALL CAPACITORS ARE .05 µF
VIDEO DATA REGISTER
VIDEO D7
VIDEO D6
VIDEO D5
VIDEO D4
VIDEO D3
VIDEO D2
VIDEO D1
VIDEO D0
(5A6)
(5C3) HOLD REG CLK
MONDMEG CPU
02-00001-019
REVISION
A
PAGE
3 OF 6


SYSTEM PARALLEL I/O
USER PARALLEL I/O
CASSETTE READ
AUDIO DAC
KEYB BRK
KEY MOD
J3-9
J3-8
R13 10K
R14 470
R12 47K
C17 22uF
Q1 PN2222
CR4 4148
CR3 4148
C13 100uF 6.3V
U18 LS14
C14 .05
1K R15
J3-1 KEYB +5
J3-2 KEYB GND
SYS1 6522 U32
SYS2 6522 U33
6522 U31
(1B4) CPU DB7
(1B4) CPU DB6
(1B4) CPU DB5
(1B4) CPU DB4
(1A4) CPU DB3
(1A4) CPU DB2
(1A4) CPU DB1
(1A4) CPU DB0
(1B4) CPU AB3
(1B4) CPU AB2
(1B4) CPU AB1
(1B4) CPU AB0
(2C4) SYS1 6522
(2C4) SYS2 6522
(1D4) LOC PHASE 2
(1D4) R/W
(1C4) RESET
J8-V
J8-7
J1-2 MUNET DATA
J1-1 MUNET CLOCK
WRITE ALLOW (6B2)
RAW ROM EN (2C6)
DISP ENAB (5D6)
B&W MODE (5B2)
PROG BANK1 (1D3)
PROG BANK0 (1D3)
DATA BANK1 (1D3)
DATA BANK0 (1D3)
J3-7 KEYB CLK
J3-14 KEYB ROW6
J3-15 KEYB ROW5
J3-17 KEYB ROW4
J3-18 KEYB ROW3
J3-16 KEYB ROW2
J3-11 KEYB ROW1
J1-3 MUNET ACT
J8-4 IRQ (1C6)
J2-8 MUTAPE DATA
J2-1 MUTAPE CLOCK
J1-4 MUNET TALK
J2-7 MUTAPE WMF. P
J2-6 MUTAPE WRT. P
J2-5 MUTAPE BOT
J2-3 MUTAPE WE
J2-4 MUTAPE FAST
J2-2 MUTAPE GO
VERT EN (5C1)
J5-3 CB2 OFF
J5-1 USER +5
J5-4 USER +12
J5-6 USER -12
J5-2 USER GND
C143 22uF 16V.
R64 4.7K
USER 6522 (2C4)
J5-25 USER CB2
J5-26 USER CB1
J5-23 USER PB7
J5-24 USER PB6
J5-21 USER PB5
J5-22 USER PB4
J5-19 USER PB3
J5-20 USER PB2
J5-17 USER PB1
J5-18 USER PB0
J5-7 USER CA2
J5-8 USER CA1
J5-15 USER PA7
J5-16 USER PA6
J5-13 USER PA5
J5-14 USER PA4
J5-11 USER PA3
J5-12 USER PA2
J5-9 USER PA1
J5-10 USER PA0
J8-4 IRQ (1C6)
J5-5 USER RESET
C98 .05
C128 1000uF 10V
J6-10 AUDIO OUTPUT
R43 2.7
C127 .05
J6-9 AUDIO GND
C126 .05
Q3 PU01
Q4 PU51
R50 220
C75 2000pF
R29 27K
R28 36K
C99 1000pF
U92 TL84
R65 1K
R30 27K
R33 27K
C51 .05
U62 DAC 0808
R21 10K
R22 2.2K
CR8 9.1V 5%
R31 470
C74 100uF 16V.
R32 27K
R23 62K
C71 .001
C73 1000pF
R27 100K
R49 68K
C101 2000pF
R47 27K
R48 16K
C102 2000pF
C48 .05
R3 240K
R1 2.2K
R4 2.2K
R5 4.7K
C4 .05
C5 .01
R6 10K
CR1 1N 270
CR2
U6 LM 339
R63 47K
R2 470
J6-2
J6-3
CASS.IN
CASS GND
MONOMEG CPU
02-00001-019
REVISION B
PAGE 4 OF 6


HORIZONTAL TIMING
VERTICAL TIMING
VIDEO GENERATOR
VIDEO SHIFT REGISTER
SYSTEM TIMING
DISP ENAB (4D4)
PIX 4 (2B3)
PIX 1 (2B6)
PIX 2 (2B6)
SEP HORIZ SYNC
PWR AC (6A3)
RAM ROW 2 CAS (3C6)
RAM ROW 1 CAS (3B6)
RAM ROW Ø CAS (3A6)
RAM ROW 3 CAS (3C6)
HOLD REG CLK (3A2)
DISP A CNT (2A3)
CPU RAS (6D2)
PHASE Ø (2A3) (1D4)
DISP RAS (6D2)
CNTC (6B2)
MUX ROW/COL (2D3)
CNT A (2B6)
10 MHZ CLK (2B6)
VERT ENABLE (4B4)
SEP VERT SYNC
DISP A CLR (2B3)
RAM ADRD (2D3)
(4D4) B&W MODE
COMP VIDEO OUT
VIDEO GND
(3B1) VIDEO D7
(3B1) VIDEO D6
(3A1) VIDEO D5
(3A1) VIDEO D4
(3A1) VIDEO D3
(3A1) VIDEO D2
(3A1) VIDEO D1
(3A1) VIDEO DØ
PAGE CLK
NOTES:
1. MAY NOT BE NEEDED
MONOMEG CPU
02-00001-019
REVISION B
PAGE 5 OF 6



MONOMEG CPU	REVISION B
02-00001-019	PAGE 6 of 6