BALLY SYSTEM DESCRIPTION

INTRODUCTION

The Bally Professional Arcade is a full-color video game system based on the mass-ram-buffer technique. A mass-ram-buffer system is one in which one or more bits of RAM are used to define the color and intensity of a pixel on the screen. The picture on the screen is defined by the contents of RAM and can easily be changed by modifying RAM.

The system uses a Z-80 Microprocessor as it's main control unit. The system ROM has software for four games: Gunfight, Checkmate, Scribbling, and Calculator. Additional ROM can be accessed through the silicon cassette connector. Three custom chips are used for the video interface, special video processing functions, keyboard and control handle interface, and audio generation.

The system exists in both high-resolution and low-resolution models. The three custom chips can operate in either mode. The mode of operation is determined by bit 0 of output port 8H. It must be set to O for low-resolution and 1 for high-resolution. This bit is not set to O at power up and must be set by software before any RAM operation can be performed.

CURSOR
P.O. Box 266
N. Hollywood, CA 91603

Cursor's Bally System Description
Version 1.0 - Released Nov 04, 2000

This document has been scanned and converted to PDF format courtesy of the *Bally Alley* newsletter. For other reprints and more information visit:

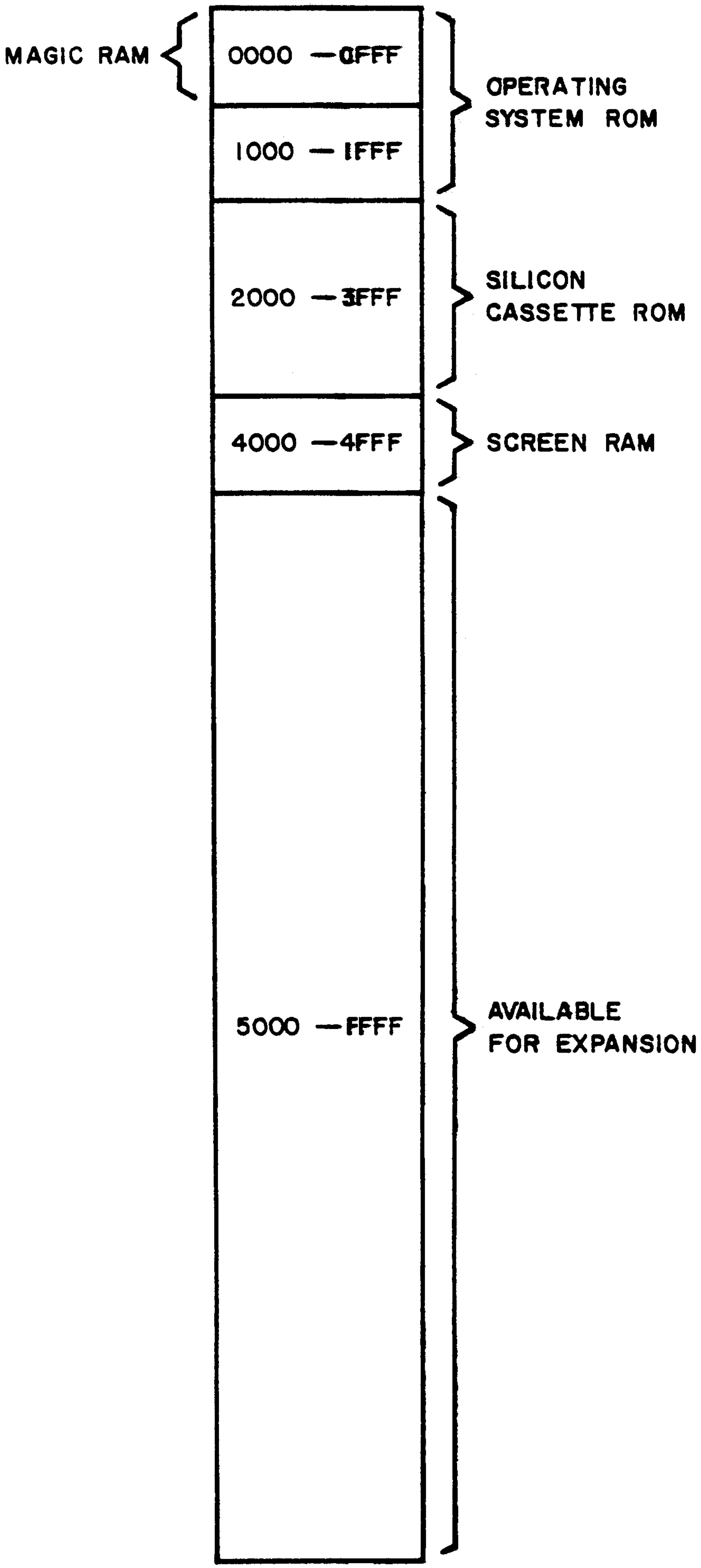
http://www.ballyalley.com

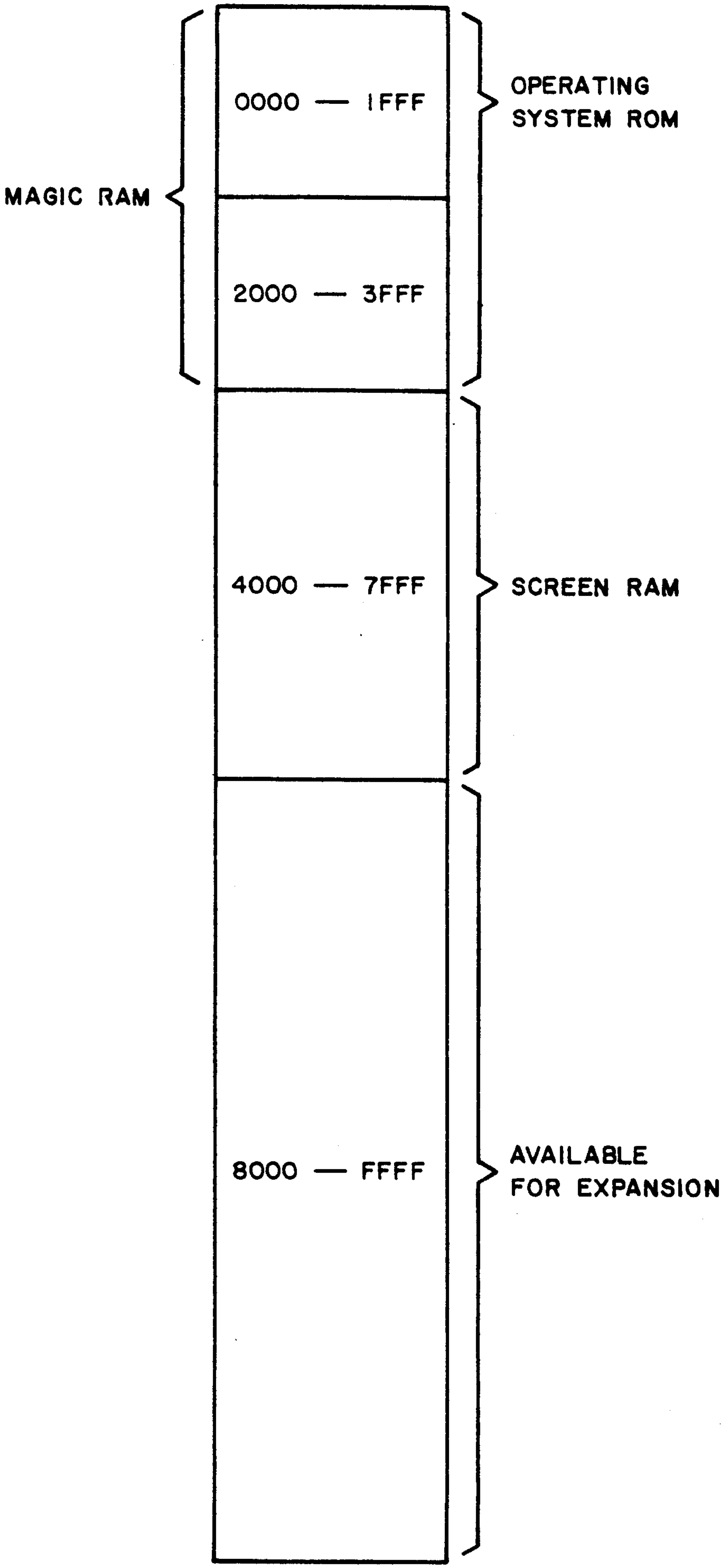
Corrections? Suggestions? Email Adam Trionfo at: ballyalley@hotmail.com

MEMORY MAP

In both the low and high resolution models, the operating system ROM is in the first 8K of memory space. The silicon cassette ROM is in the space from 8K to 16K. The standard screen RAM begins at 16K. In the low-resolution unit, standard screen RAM is 4K bytes; in the high-resolution unit it is 16K bytes. Magic screen RAM begins at location Ø. It is the same size as standard screen RAM. All memory above 32K is available for expansion. In the low-resolution unit, memory space 20K - 32K is available for expansion.

When data is read from a memory location between Ø and 16K the data comes from the ROM. When data is written in a memory location (X) between Ø and 16K, the system actually writes a modified from of the data in location X+16K. The modification is performed by the magic system in the Data Chip and Address Chip. Thus the RAM from 0 to 16K is called Magic Memory.


MAGIC RAM
0000 —0FFF
1000 — 1FFF
OPERATING SYSTEM ROM
2000 —3FFF
SILICON CASSETTE ROM
4000 —4FFF
SCREEN RAM
5000 —FFFF
AVAILABLE FOR EXPANSION



MEMORY MAP HIGH RESOLUTION

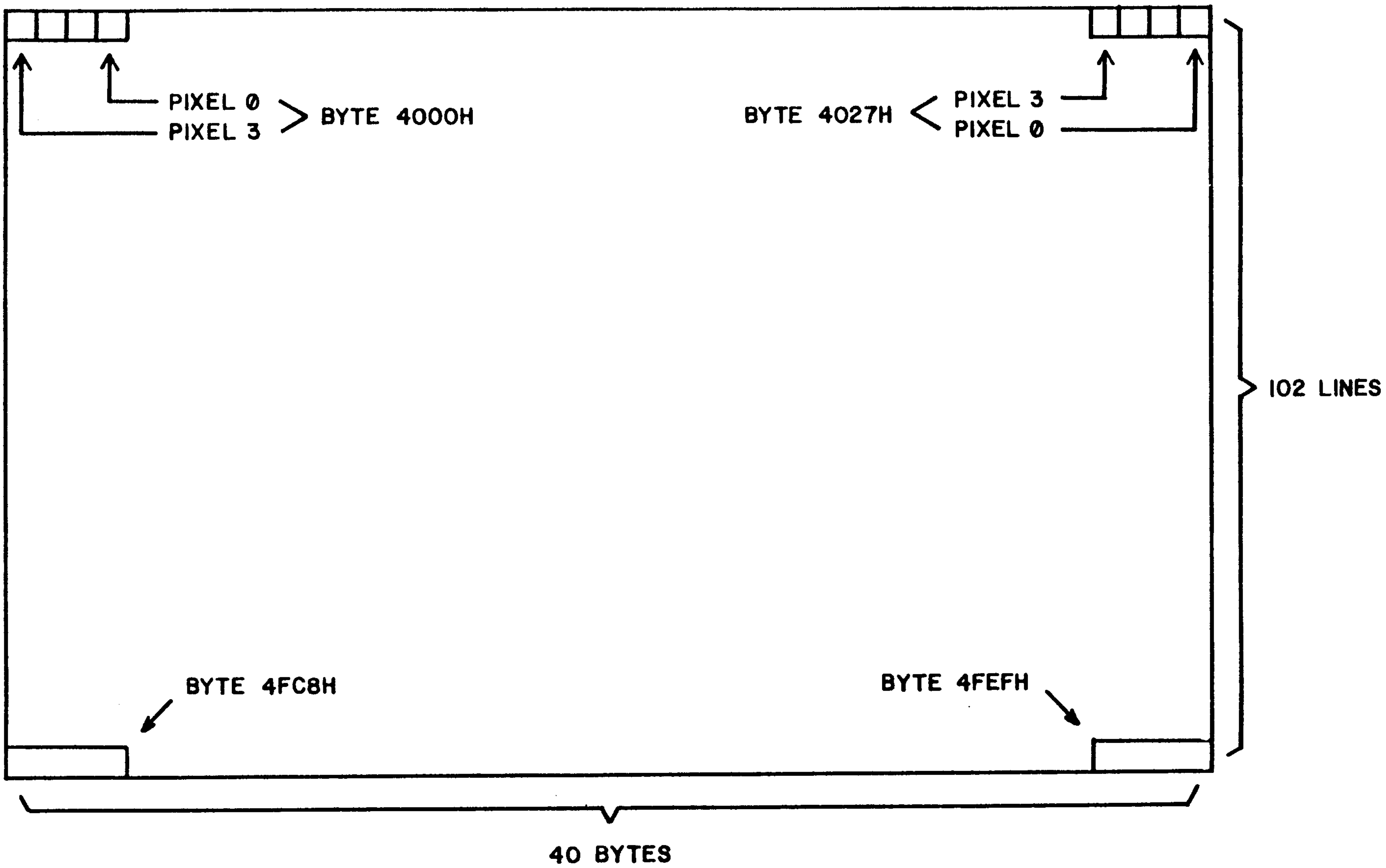
SCREEN MAP

In the Bally Professional Arcade, two bits of RAM are used to define a pixel on the screen. One 8-bit byte of RAM therefor defines four pixels on the screen.

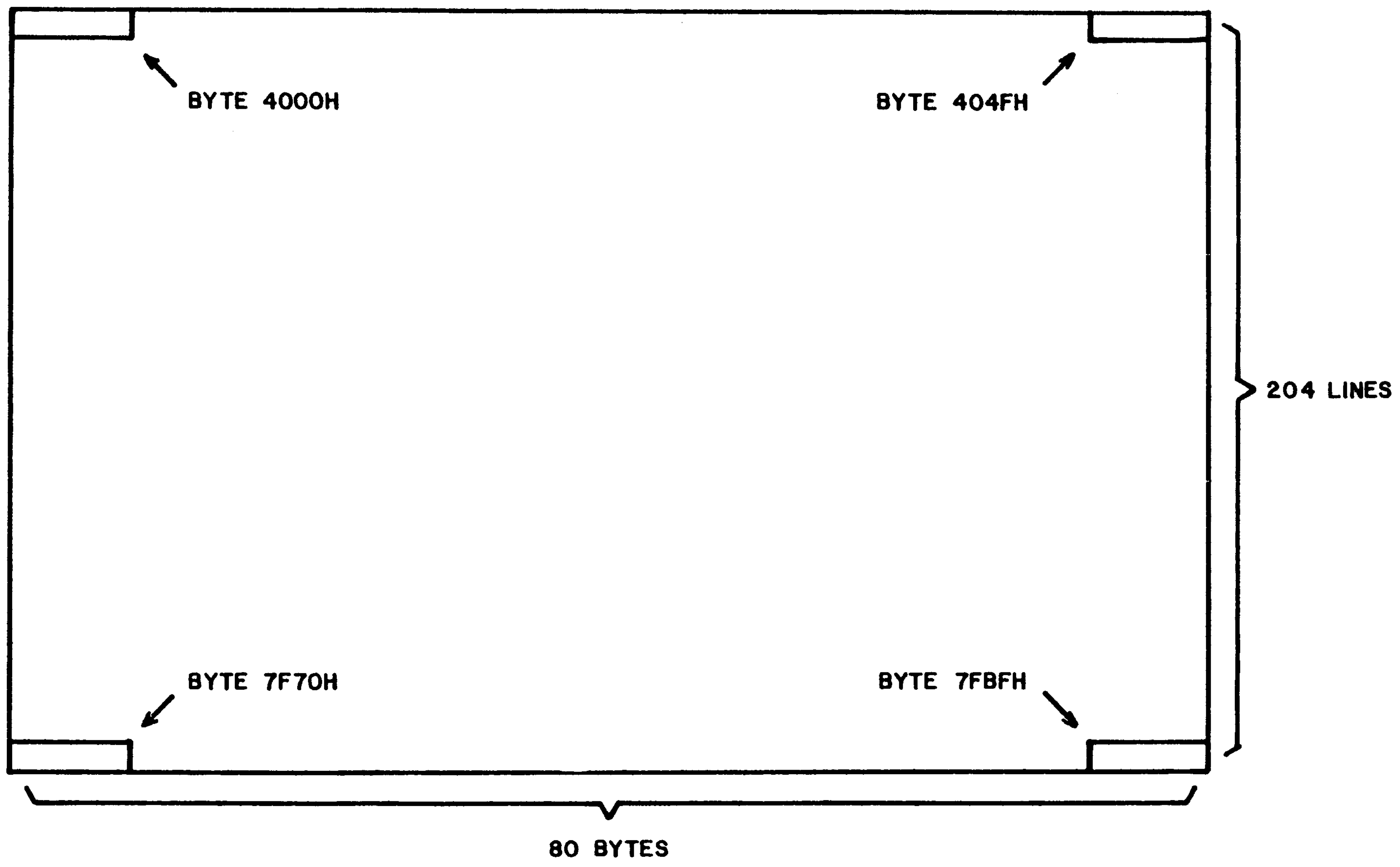
In the low-resolution model there are 40 bytes used to define a line of data. This gives a horizontal resolution of 160 pixels. The vertical resolution is 102 lines. The screen therefor requires 102 X 40 = 4,080 bytes. The remaining 16 bytes of the 4K RAM are used for scratch pad. More of the RAM can be used for scratchpad by blanking the screen before the 102nd line. This will be described later.

In the high-resolution model there are 80 bytes and 320 pixels per line. The 204 lines require 16,320 bytes of RAM. 64 bytes of the 16K RAM are left for scratch pad.

In both models the first byte of RAM is in the upper left-hand corner of the screen. As the RAM address increases, the position on the screen moves in the same directions as the TV scan; from left-to-right and from top-to-bottom. The four pixels in each byte are displayed with the least significant pixel, the one defined by bits Ø and 1, on the right.



SCREEN MAP LOW RESOLUTION


BYTE 4000H
BYTE 404FH
204 LINES
BYTE 7F70H
BYTE 7FBFH
80 BYTES

COLOR MAPPING

Two bits are used to represent each pixel on the screen. These two bits, along with the LEFT/RIGHT bit which is set by crossing the horizontal color boundary, map each pixel to one of eight different color registers. The value in the color register then defines the color and intensity of the pixel on the screen. The intensity of the pixel is defined by the three least significant bits of the register, ØØØ for darkest and 111 for lightest. The color is defined by the five most significant bits. The color registers are at output ports Ø through 7; register Ø at port Ø, register 1 at port 1, etc.

The color registers can be accessed as individual ports ar all eight can be accessed by one OTIR instruction. The OTIR instruction is to port BH (register C=BH) and register B should be set to 8. The eight bytes of data pointed to by HL will go to the color registers

HL → Memory Location	X	Color Register 7
	X+1	Color Register 6
	X+2	Color Register 5
	X+3	Color Register 4
	X+4	Color Register 3
	X+5	Color Register 2
	X+6	Color Register 1
	X+7	Color Register Ø

The horizontal color boundary (bits Ø-5 of port 9) defines the horizontal position of an imaginary vertical line on the screen. The boundary line can be position between any two adjacent bytes in the low-resolution system. The line is immediately to the left of the byte whose number is sent to bits Ø-5 of port 9. For example, if the horizontal color boundary is set to Ø, the line will be just to the left of byte Ø; if it is set to 20, the line will be between bytes 19 and 20 in the center of the screen.

If a pixel is to the left of the boundary, its LEFT/RIGHT bit is set to 1. The LEFT/RIGHT bit is set to Ø for pixels to the right of the boundary. Color registers Ø-3 are used for pixels to the right of th boundary and registers 4-7 are used for pixels to the left of the boundary.

In the high-resolution system, the boundary is placed in the same position on the screen but between different bytes. If the value X is sent to the horizontal color boundary, then the boundary will be between bytes 2X and 2X-1. If the value 20 is sent, the boundary wil be between 39 and 40, in the center of the screen.

To put the entire screen, including the right side background, on the left side of the boundary, set the horizontal color boundary to 40

BACKGROUND COLOR

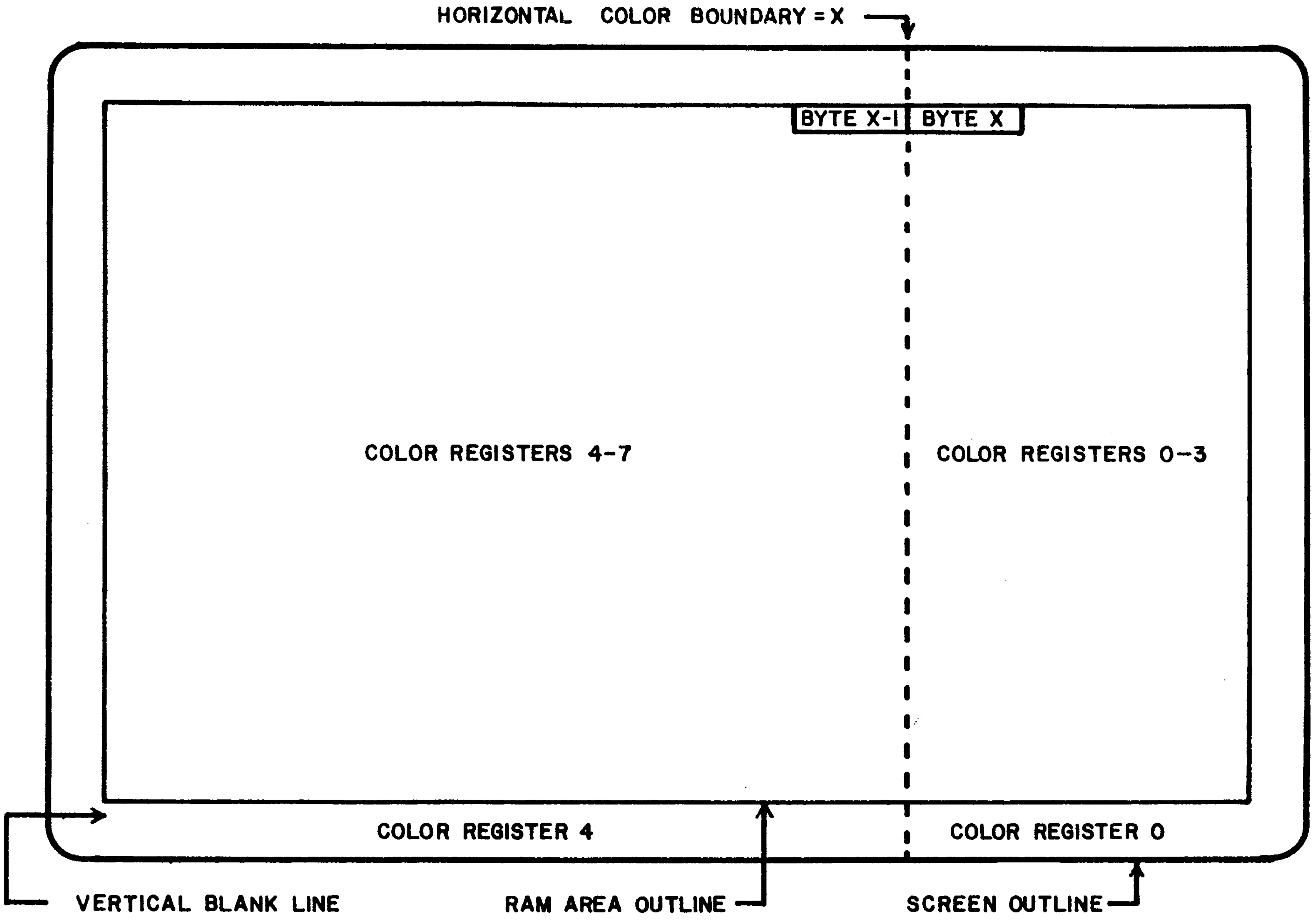
On most television the area defined by RAM is slightly smaller than t screen. There is generally extra space on all four sides of the RAM area. The color and intensity of this area is defined by the backgro color number (bits 6 and 7 of port 9). These two bits, along with the LEFT/RIGHT bit point to one of the color registers which determin the color and intensity.

VERTICAL BLANK

The Vertical Blank Register (output port AH) contains the line number on which vertical blanking will begin. In the low-resolution system bit Ø should be set to Ø and the line number should be in bits 1-7. In the high-resolution system the line number is in bits Ø-7. The background color will be displayed from the vertical blank line to the bottom of the screen. This allows the RAM that would normally be displayed in that area to be used for scratch pad. If the vertical blank register is set to Ø the entire RAM can be used for scratch pad. In a low-resolution system the register must be set to 101 or less; in a high-resolution system it must be set to 203 or less.

SUMMARY

The following color register map shows which color registers are used to define colors in different areas of the screen. The map assumes the background color is set to Ø. If it were set to 1 then color registers 1 and 5 would be used for background instead of Ø and 4. In the low-resolution system the color boundary is between bytes X and X-1. In the high-resolution system the boundary is between bytes 2X and 2X-1.



COLOR REGISTER MAP

INTERRUPT FEEDBACK

When the Z-80 acknowledges an interrupt it reads 8 bits of data from the data bus. It then uses this data as an instruction or an address. In the Bally Professional Arcade this data is determined by the contents of the interrupt feedback register (output port DH). In responding to a screen interrupt the contents of the interrupt feedback register are placed directly on the data bus. In responding to a light pen interrupt the lower four bits of the data bus are set to Ø and the upper four bits are the same as the corresponding bits of the feedback register.

INTERRUPT CONTROL BITS

In order for the Z-80 to be interrupted the internal interrupt enable flip-flop must be set by an EI instruction and one or two of the external interrupt enable bits must be set (output port EH). If bit 1 is set, light pen interrupts can occur. If bit 3 is set, screen interrupts can occur. If both bits are set, both interrupts can occur and the screen interrupt has higher priority.

The interrupt mode bits determine what happens if an interrupt occurs when the Z-80's interrupt enable flip-flop is not set. Each of the two interrupts may have a different mode. In mode Ø the Z-80 will continue to be interrupted until it finally enables interrupts and acknowledges the interrupt. In mode 1 the interrupt will be discarded if it is not acknowledged by the next instruction after it occured. If mode 1 is used the software must be designed such that the system will not be executing certain Z-80 instructions when the interrupt occurs. The opcodes of these instructions begin with CBH, DDH, EDH, and FDH.

The mode bit for light pen interrupt is bit Ø of port EH and the mode bit for screen interrupt is bit 2 of port EH.

SCREEN INTERRUPT

The purpose of the screen interrupt is to synchronize the software with the video system. The software must send a line number to the interrupt line register (output port FH). In the low-resolution system bit Ø is set to Ø and the line number is sent to bits 1-7. In the high-resolution system the line number is sent to bits 0-7. If the screen interrupt enable bit is set, the Z-80 will be interrupted when the video system completes scanning the line in the interrupt register. This interrupt can be used for timing since each line is scanned 60 times a second. It can also be used in conjunction with the color registers to make as many as 256 color-intensity combinations appear on the screen at the same time.

LIGHT PEN INTERRUPT

The light pen interrupt occurs when the light pen trigger is pressed and the video scan crosses the point on the screen where the light pen is. The interrupt routine can read two registers to determine the position of the light pen. The line number is read from the vertical feedback register (input port EH). In the high-resolution system the line number is in bits Ø-7. In the low-resolution system the line number is in bits 1-7, bit Ø should be ignored. The horizontal position of the light pen can be determined by reading input port FH and subtracting 8. In the low-resolution system the resultant value is the pixel number, Ø to 159. In the high-resolution system the resultant must be multiplied by two to give the pixel number, Ø to 358.

MAGIC REGISTER

As described earlier, the Magic System is enable when data is written to a memory location (X) from Ø to 16K. A modified form of the data is actually written in memory location X+16K. The magic register (output port CH) determines how the data is modified. The purpose of each bit of the magic register is shown below.

Bit Ø	LSB of shift amount
1	MSB of shift amount
2	Rotate
3	Expand
4	OR
5	XOR
6	Flop

The order is which magic functions are performed is as follows: Expansion is done first; rotating or shifting; flopping; OR or XOR. As many as four can be used at any one time and any function can be bypassed. Rotate and shift as well as OR and XOR cannot be done at the same time.

EXPAND

The expander is used to expand the 8 bit data bus into 8 pixels (or 16 bits). It.expands a Ø on the data bus into a two-bit pixel and a 1 into another two-bit pixel. Thus, two-color patterns can be stored in ROM in half the normal memory space.

During each memory write instruction using the expander, either the upper half or the lower half of the data bus is expanded. The half used is determined by the expand flip-flop. The flip-flop is reset by an output to the magic register and is toggled after each magic memory write. The upper half of the data bus is expanded when the flip-flop is Ø, and the lower half when the flip-flop is 1.

The expand register (output port 19H) determines the pixel values into which the data bus will be expanded. A Ø on the data bus will be expanded into the pixel defined by bits Ø and 1 of the expand register. A 1 on the data bus will be expanded into the pixel defined by bits 2 and 3 of the expand register.

The pixels generated by bit Ø or 4 of the data bus will be the least significant pixel of the expanded byte. The most significant pixel will come from bit 3 or 7.

SHIFTER

The shifter, flopper, and rotator operate on pixels rather than bits. Each byte is thought of as containing four pixels, each of which has one of four values. The four pixels are referred to as PØ, P1, P2, and P3. PØ is composed of the first two bits of the byte.

The shifter shifts data Ø, 1, 2, or 3 pixels to the right. The shift amount is determined by bits Ø and 1 of the magic register. The pixels that are shifted out of one byte are shifted into the next byte. Ø's are shifted into the first byte of a sequence. The shifter assumes the first byte of a sequence is the first magic memory write after an output to the magic register. Each sequence must be initialized by an output to the magic register and data cannot be sent to the magic register in the middle of a sequence.

FLOPPER

The output of the flopper is a mirror image of it's input. Pixel Ø and 3 exchange values as do pixel 1 and 2.

The diagrams on the following page show examples of shifting and flopping.

BYTE 0				BYTE 1				BYTE 2				
P3	P2	P1	P0	P7	P6	P5	P4	P11	P10	P9	P8	ORIGINAL DATA
0	P3	P2	P1	P0	P7	P6	P5	P4	P11	P10	P9	SHIFT 1
0	0	P3	P2	P1	P0	P7	P6	P5	P4	P11	P10	SHIFT 2
0	0	0	P3	P2	P1	P0	P7	P6	P5	P4	P11	SHIFT 3
P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	FLOPPED

SHIFTER — FLOPPER

ROTATOR

The rotator is used to rotate a 4 X 4 pixel image 90^0 in a clockwise direction. The rotator is initialized by an output to the magic register and will re-initialize itself after every eight writes to magic memory. To perform a rotation, the following procedure must be performed twice. Write the top byte of the unrotated image to a location in magic memory. Write the next byte to the first location plus 80, the next byte to the first location plus 160, and the last byte to the first location plus 240. After eight writes the data will appear in RAM and on the screen rotated 90^0 from the original image.

The rotator can only be used in commercial mode.

The diagram on the following page shows an example of rotating.

ROTATOR

P3	P2	P1	P0
P7	P6	P5	P4
P11	P10	P9	P8
P15	P14	P13	P12

ORIGINAL

P15	P11	P7	P3
P14	P10	P6	P2
P13	P9	P5	P1
P12	P8	P4	P0

ROTATED

OR AND XOR

These functions operate on a byte as 8-bits rather than four pixels. When the OR function is used in writing data to RAM, the input to the OR circuit is ORed with the contents of the RAM location being accessed. The resultant is then written in RAM.

The XOR function operates in the same way except that the data is XORed instead of ORed.

INTERCEPT

Software reads the intercept register (input port 8H) to determine if an intercept occured on an OR or XOR write. An intercept is defined as the writing of a non-zero pixel in a pixel location that previously contained a non-zero pixel. A non-zero pixel is a pixel with a value of Ø1, 1Ø, or 11. A 1 in the intercept register means an intercept has occured. Bits Ø - 3 give the intercept information for all OR or XOR writes since the last input from the intercept register. An input from the intercept register resets these bits. A bit is set to 1 if an intercept occurs in the appropriate position and will not be reset until after the next intercept register input.

Bit	
Ø	Intercept in pixel 3 in an OR or XOR write since last reset
1	Intercept in pixel 2 in an OR or XOR write since last reset
2	Intercept in pixel 1 in an OR or XOR write since last reset
3	Intercept in pixel Ø in an OR or XOR write since last reset
4	Intercept in pixel 3 in last OR or XOR write
5	Intercept in pixel 2 in last OR or XOR write
6	Intercept in pixel 1 in last OR or XOR write
7	Intercept in pixel Ø in last OR or XOR write

PLAYER INPUT

The system will accomodate up to four player control handles at once. Each handle has five switches and a potentiometer. The switches are read by the Z-80 on input ports 10H - 13H and are not debounced. The switches are normally open and normally feedback Ø's.

The signals from the potentiometers are changed to digital information by an 8-bit Analog-to-Digital Convertor. The four pots are on input ports 1CH - 1FH. All Ø's are fedback when the pot is turned fully counter-clockwise and all 1's when turned fully clockwise.

The 24-button keypad is read on bits Ø-5 of ports 14H-17H. The data is normally Ø and if more than one button is depressed, the data should be ignored. The keypad will not send back the proper data if any of the player control switches are closed. Here again, the buttons are not debounced.

Player control inputs are shown on the following page.

<table>
<tr><th>PORT ↓</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0 ←BIT</th><th></th></tr>
<tr><td>10H</td><td></td><td></td><td></td><td>TRIG</td><td>RIGHT</td><td>LEFT</td><td>DOWN</td><td>UP</td><td>PLAYER 1</td></tr>
<tr><td>11H</td><td></td><td></td><td></td><td>TRIG</td><td>RIGHT</td><td>LEFT</td><td>DOWN</td><td>UP</td><td>PLAYER 2</td></tr>
<tr><td>12H</td><td></td><td></td><td></td><td>TRIG</td><td>RIGHT</td><td>LEFT</td><td>DOWN</td><td>UP</td><td>PLAYER 3</td></tr>
<tr><td>13H</td><td></td><td></td><td></td><td>TRIG</td><td>RIGHT</td><td>LEFT</td><td>DOWN</td><td>UP</td><td>PLAYER 4</td></tr>
<tr><td>14H</td><td></td><td></td><td>=</td><td>+</td><td>−</td><td>X</td><td>÷</td><td>%</td><td>KEYPAD</td></tr>
<tr><td>15H</td><td></td><td></td><td></td><td>3</td><td>6</td><td>9</td><td>CH</td><td>∨</td><td>KEYPAD</td></tr>
<tr><td>16H</td><td></td><td></td><td>0</td><td>2</td><td>5</td><td>8</td><td>MS</td><td>∧</td><td>KEYPAD</td></tr>
<tr><td>17H</td><td></td><td></td><td>CE</td><td>1</td><td>4</td><td>7</td><td>MR</td><td>C</td><td>KEYPAD</td></tr>
<tr><td>1CH</td><td colspan="8">← POT →</td><td>PLAYER 1</td></tr>
<tr><td>1DH</td><td colspan="8">← POT →</td><td>PLAYER 2</td></tr>
<tr><td>1EH</td><td colspan="8">← POT →</td><td>PLAYER 3</td></tr>
<tr><td>1FH</td><td colspan="8">← POT →</td><td>PLAYER 4</td></tr>
</table>

PLAYER INPUT

MASTER OSCILLATOR

The frequency of the master oscillator is determined by the contents of several output ports. Port 1ØH sets the master frequency. It is given by the following formula:

$$F_m = \frac{1789}{\text{PORT 10H} + 1} \text{ Khz}$$

If bit 4 of output port 15H is set to 1, the master oscillator frequency will be modulated by noise. The amount of modulation will be set by the 8-bit noise volume register, output port 17H.

If bit 4 of output port 15H is set to Ø, the frequency of the master oscillator will be modulated by a constant value to give a vibrato effect. The amount of modulation will be set by the vibrato depth register (the first 6 bits of output port 14H). The speed of modulation is set by the vibrato speed register (upper 2 bits of output port 14H); ØØ for fastest and 11 for slowest.

Frequency modulation is accomplished by adding a modulation value to the contents of port 1ØH and sending the result to the master oscillator frequency generator. In noise modulation, the modulation value is an 8-bit word from the noise generator. If a bit in the noise volume register is set to Ø, the corresponding bit in the modulation value word will be set to Ø. In vibrato modulation, the modulation value alternates between Ø and the contents of the vibrato volume register.

Modulation can be completely disabled by setting the master volume to Ø if noise modulation is being used, or by setting the vibrato depth to Ø when vibrato is used.

TONES

The system contains three tone generators each clocked by the same master oscillator. The frequency of Tone A is set by output port 11H, Tone B by output port 12H, and Tone C by output port 13H. The frequency is given by the following formula:

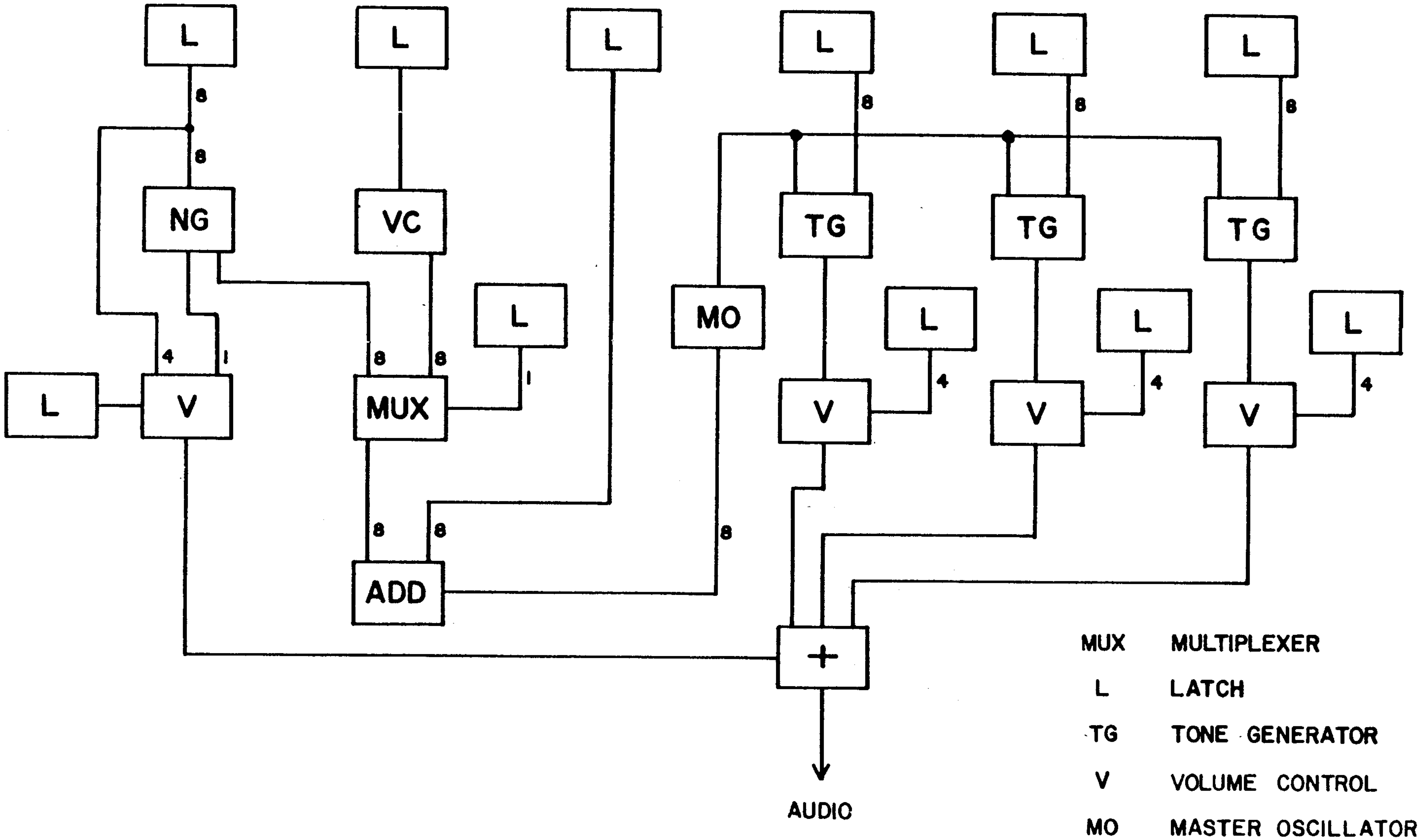
$$F_t = \frac{F_m}{2(\textbf{contents of TONE PORT}+1)} = \frac{894}{(\textbf{PORT 10H}+1)(\textbf{contents of TONE PORT}+1)}\ \textbf{Khz}$$

The tone volumes are controlled by output ports 15H and 16H. The lower 4 bits of port 16H set Tone A Volume, the upper 4 bits sets Tone B Volume. The lower 4 bits of port 15H sets Tone C Volume. Noise can be mixed with the tones by setting bit 5 of port 15H to 1. In this case the noise volume is given by the upper 4 bits of port 17H. In all cases a volume of Ø is silence and a volume of all 1's is loudest.

SOUND BLOCK TRANSFER

All 8 bytes of sound control can be sent to the audio circuit with one OTIR instruction. Register C should be sent to 18H, register B to 8H and HL pointing to the 8 bytes of data. The data pointed to by HL goes to port 17H and the next 7 bytes of data goes to ports 16H through 1ØH.

HL →	Memory Location	X	Data-to-port	17H
		X+1	Data-to-port	16H
		X+2	Data-to-port	15H
		X+3	Data-to-port	14H
		X+4	Data-to-port	13H
		X+5	Data-to-port	12H
		X+6	Data-to-port	11H
		X+7	Data-to-port	1ØH



MUX	MULTIPLEXER
L	LATCH
TG	TONE GENERATOR
V	VOLUME CONTROL
MO	MASTER OSCILLATOR
VC	VIBRATO CONTROL
NG	NOISE GENERATOR
ADD	ADDER

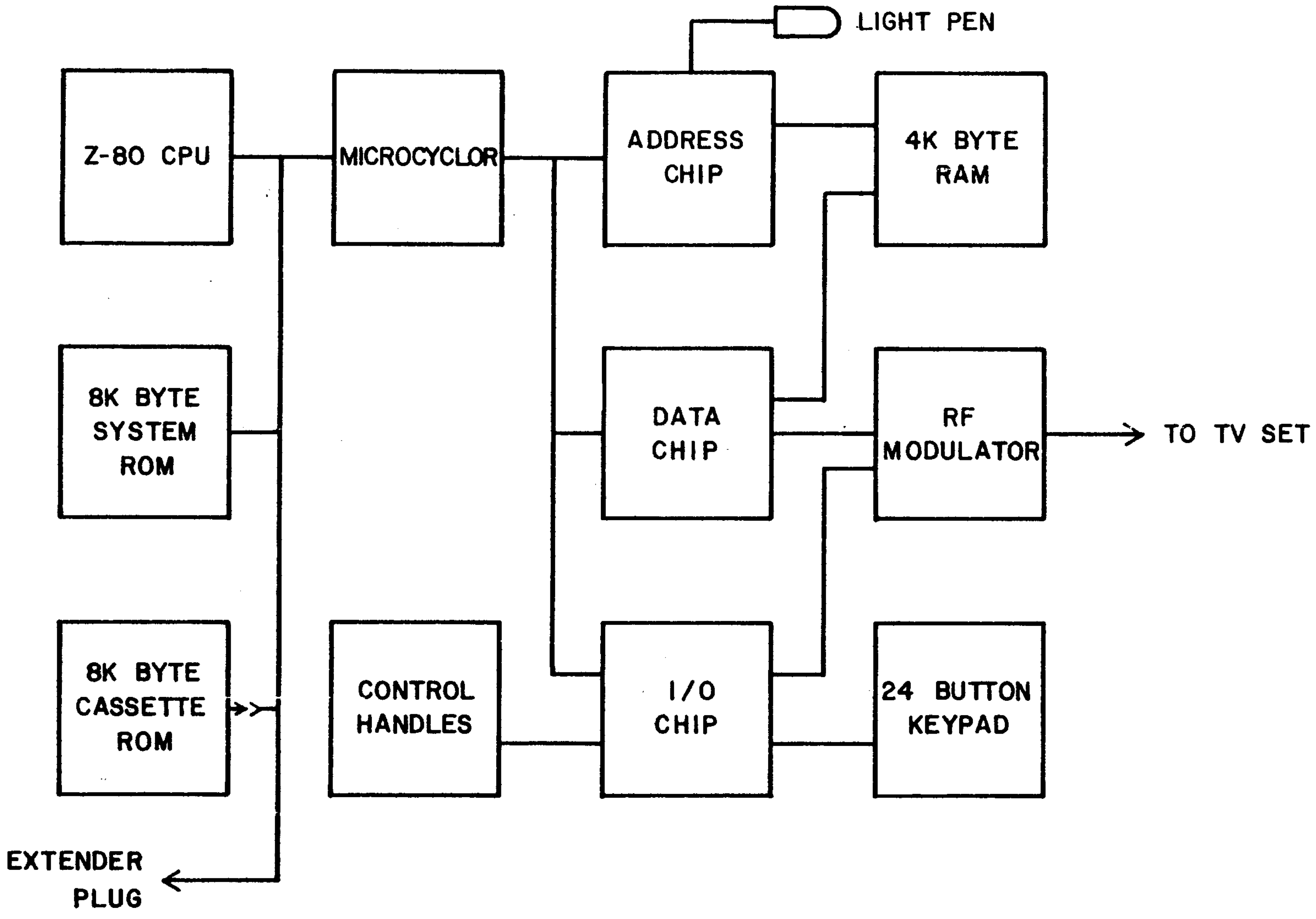
AUDIO GENERATOR BLOCK DIAGRAM

OUTPUT PORTS

PORT NUMBER	FUNCTION
ØH	Color Register Ø
1H	Color Register 1
2H	Color Register 2
3H	Color Register 3
4H	Color Register 4
5H	Color Register 5
6H	Color Register 6
7H	Color Register 7
8H	Low/High Resolution
9H	Horizontal Color Boundary, Background Color
AH	Vertical Blank Register
BH	Color Block Transfer
CH	Magic Register
DH	Interrupt Feedback Register
EH	Interrupt Enable and Mode
FH	Interrupt Line
1ØH	Master Oscillator
11H	Tone A Frequency
12H	Tone B Frequency
13H	Tone C Frequency
14H	Vibrato Register
15H	Tone C Volume, Noise Modulation Control
16H	Tone A Volume, Tone B Volume
17H	Noise Volume Register
18H	Sound Block Transfer
19H	Expand Register

INPUT PORTS

PORT NUMBER	FUNCTION
8H	Intercept Feedback
EH	Vertical Line Feedback
FH	Horizontal Address Feedback
1ØH	Player 1 Handle
11H	Player 2 Handle
12H	Player 3 Handle
13H	Player 4 Handle
14H	Keypad Column Ø (right)
15H	Keypad Column 1
16H	Keypad Column 2
17H	Keypad Column 3 (left)



SYSTEM BLOCK DIAGRAM

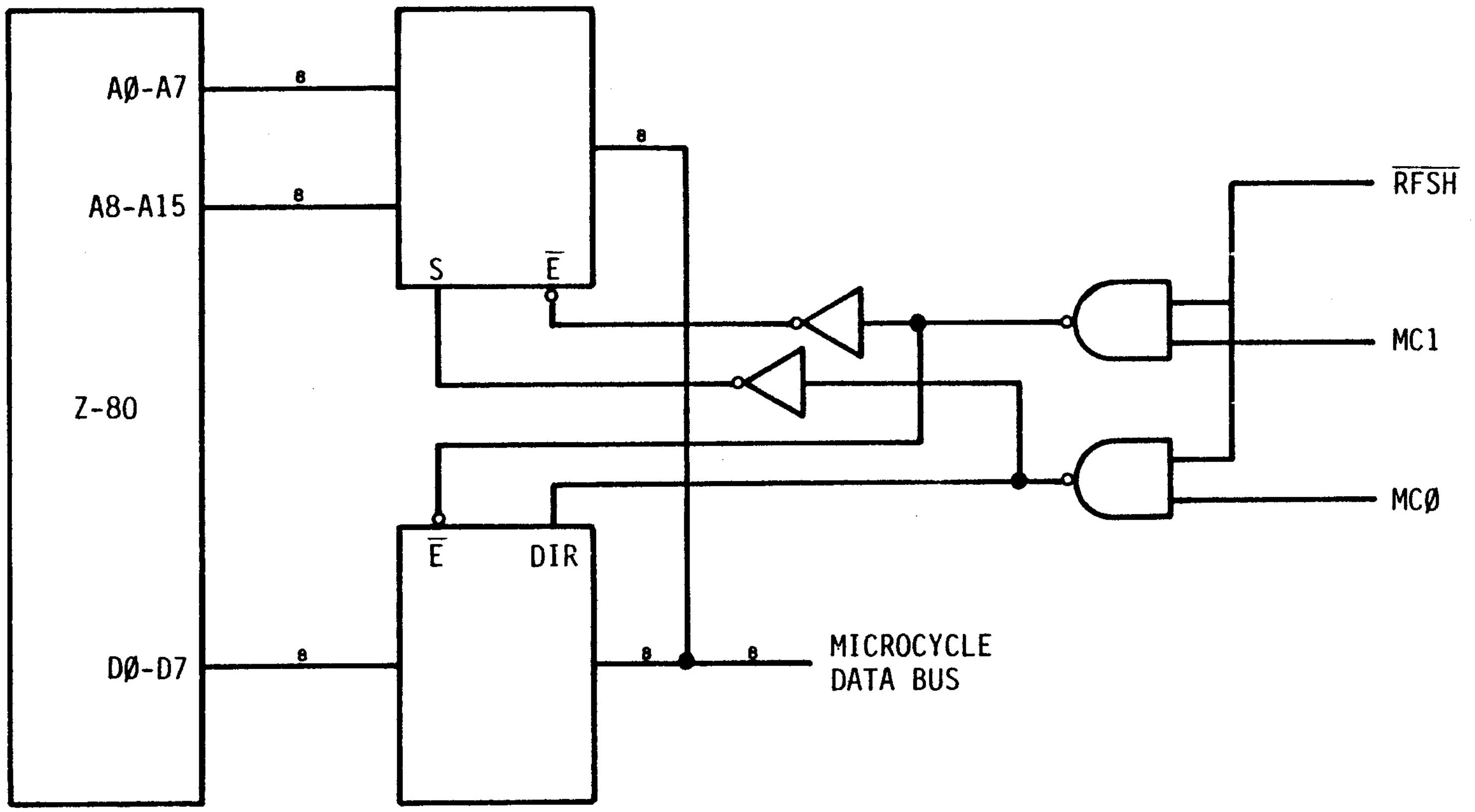
MICROCYCLER

The purpose of the microcycler is to combine the 16-bit Address Bus and the 8-bit Data Bus from the Z-8Ø into one 8-bit Microcycle Data Bus to the Data Chip, Address Chip, and I/O Chip. This was done to reduce the pin count on the custom chips.

The Microcycle Data Bus can be in any of four modes. Its mode is controlled by MCØ and MC1 coming from the Data Chip and $\overline{\text{RFSH}}$ from the Z-8Ø. The modes are shown below.

$\overline{\text{RFSH}}$	MCØ	MC1	Microcycle Data Bus Contents
Ø	Ø	Ø	AØ - A7 from Z-8Ø
Ø	Ø	1	AØ - A7 from Z-8Ø
Ø	1	Ø	AØ - A7 from Z-8Ø
Ø	1	1	AØ - A7 from Z-8Ø
1	Ø	Ø	AØ - A7 from Z-8Ø
1	Ø	1	A8 - A15 from Z-8Ø
1	1	Ø	DØ - D7 from Z-8Ø
1	1	1	DØ - D7 to Z-8Ø

MCØ and MC1 change 140 nsec after the rising edge of Φ. Their changes are shown in the timing diagrams of various instruction cycles.



MICROCYCLER BLOCK DIAGRAM

ADDRESS CHIP DESCRIPTION

The Microcycle Decoder generates twelve bits of Z-8Ø address from the 8-bit Microcycle Data Bus. This address is then fed through MUX I and MUX II to MAØ-5 which go to the RAM. The Scan Address Generator generates a 12-bit address which is used to read video data from the RAM. This address goes from Ø to FFFH once every frame (1/60 sec.).

MUX I sends either the Scan Address or Z-8Ø Address to its 12 outputs. An output of the Scan Address Generator controls MUX I. If the Scan Address Generator and the Z-8Ø request a memory cycle at the same time, the Scan Address Generator will have higher priority and the Z-8Ø will be required to wait (by the $\overline{\text{WAIT}}$ output). The Scan Address Generator never requires the memory for more than one consecutive memory cycle, so the Z-8Ø is never required to wait for the memory for more than one cycle. HORIZ DR and VERT DR synchronize the Scan Address Generator with the Data Chip and the TV Scan.

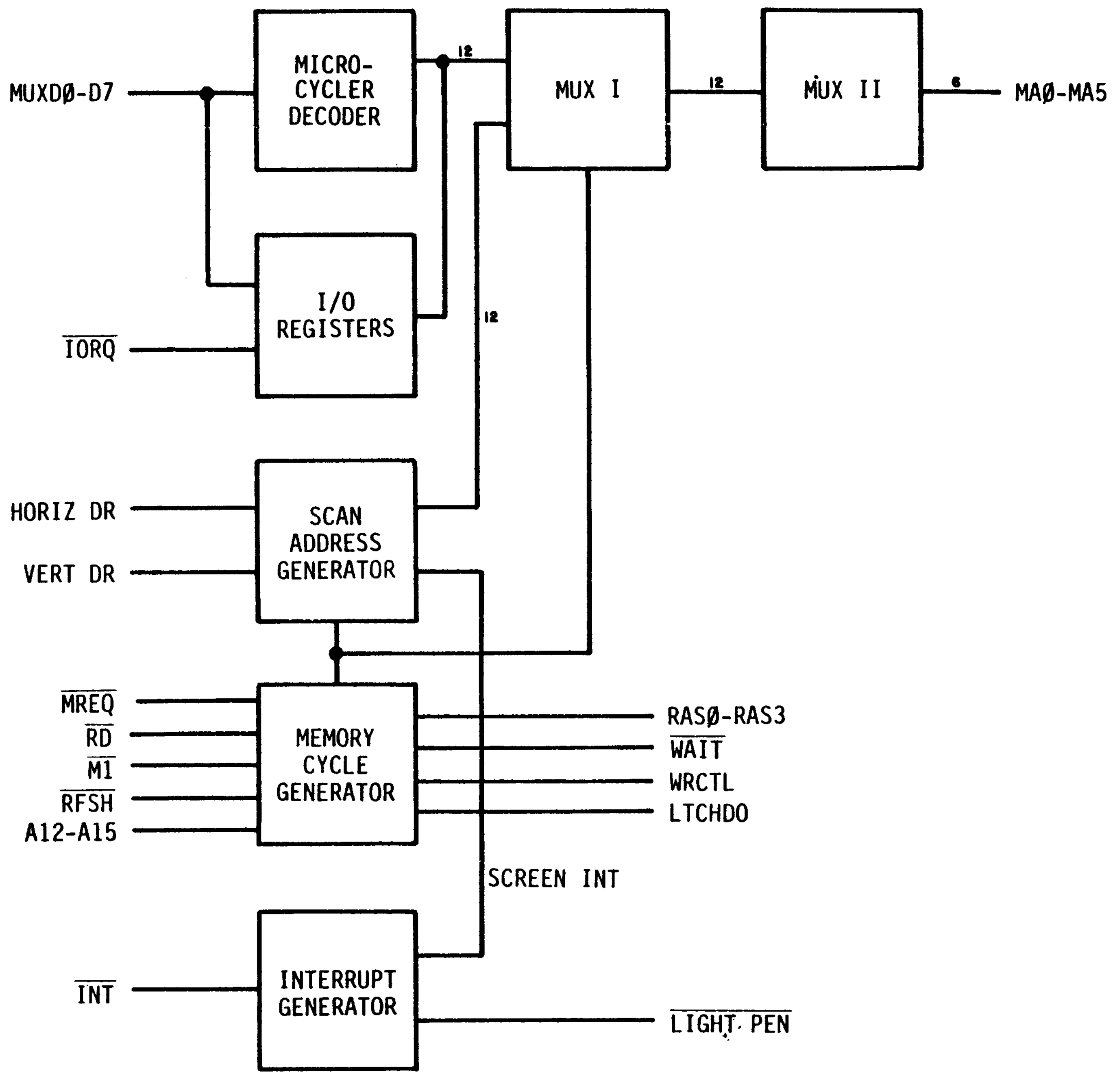
The purpose of MUX II is to multiplex its 12 inputs to the six address bits in the two time slices required for 4K x 1 16 pin RAMS.

The Memory Cycle Generator controls memory cycles generated by either the Z-8Ø or Scan Address Generator. $\overline{\text{MREQ}}$, $\overline{\text{RD}}$, $\overline{\text{M1}}$, $\overline{\text{RFSH}}$, and A12-A15 are from the Z-8Ø. A12-A15 are fed directly from the Z-8Ø because if they were brought out of the microcycle decoder, they would arrive too late in the memory cycle. The RASØ - RAS3 outputs are used to activate memory cycles. In the consumer game, only RASØ is used to one bank of RAM (4K x 8). In the commercial game, all four RAS's are used to control four banks of RAM (16K x 8). WRCTL and LTCHDO are control signals to the Data Chip. WRCTL tells the Data Chip when to place data to be written to memory on the memory data bus. LTCHDO tells the Data Chip when valid data from RAM is present on the memory data bus.

As mentioned earlier, $\overline{\text{WAIT}}$ is generated when the Z-8Ø and Scan Address Generator both request memory at the same time. $\overline{\text{WAIT}}$ is also generated for one cycle every time the Z-8Ø requests a memory access, even if there is no conflict with the Scan Address. This is because the microcycler slows down Z-8Ø memory accesses. The Z-8Ø address bus and data bus must time share the microcycle bus so the Z-8Ø data reaches the microcycle bus very late in the memory cycle.

The INT Generator generates two types of interrupts to the Z-8Ø; Light Pen and Screen interrupts. A screen interrupt is generated when screen interrupts are enabled and the TV scan completes a certain line on the screen (from Ø to 255). The line at which the interrupt will occur is determined by the Z-8Ø. This interrupt can be used for timing since the TV rescans every line once every 1/60 sec. A light pen interrupt occurs when the light pen interrupt is enabled and $\overline{\text{LIGHT PEN}}$ goes low. The current scan address is saved in latches in the Scan Address Generator. The Z-8Ø can read the contents of these latches to determine the scan address at the time $\overline{\text{LIGHT PEN}}$ was activated and thus the position of the light pen on the screen.

The I/O Decode circuit is used during Z-8Ø input and output instructions. Z-8Ø input instructions are used to read the scan address after light pen interrupts. Output instructions are used to enable the two interrupts and set the line number for screen interrupts.



ADDRESS CHIP BLOCK DIAGRAM

DATA CHIP DESCRIPTION

The TV Sync Generator uses 7M and $\overline{7M}$ (7.159090 Mhz square waves) to generate NTSC standard sync and blank to be sent to the Video Generator. It also generates HORIZ DR and VERT DR for synchronization with the Address Chip. HORIZ DR occurs once every horizontal line (63.5 usec), and VERT DR occurs once every frame (16.6 msec).

The Shift Register loads parallel data from the memory data bus (MDØ - MD7) and shifts it out of its two serial outputs. The TV sync Generator controls when data is loaded or shifted. In a consumer game,the two outputs of the shift register are sent through MUX I to MUX II. In a commercial game, SERIAL Ø and SERIAL 1 are sent through the MUX I to MUX II. The two bits from MUX I select 8 bits to be sent through MUX II to the Video Generator. These 8 bits then determine the analog values of VIDEO, R-Y, and B-Y. 2.5V is a 2.5V D C reference level.

The Clock Generator generates ØG and $\overline{PX}$ from 7M. These are the clocks for the rest of the system. The frequency of $\overline{PX}$ is half that of 7M and the frequency of ØG is half that of $\overline{PX}$.

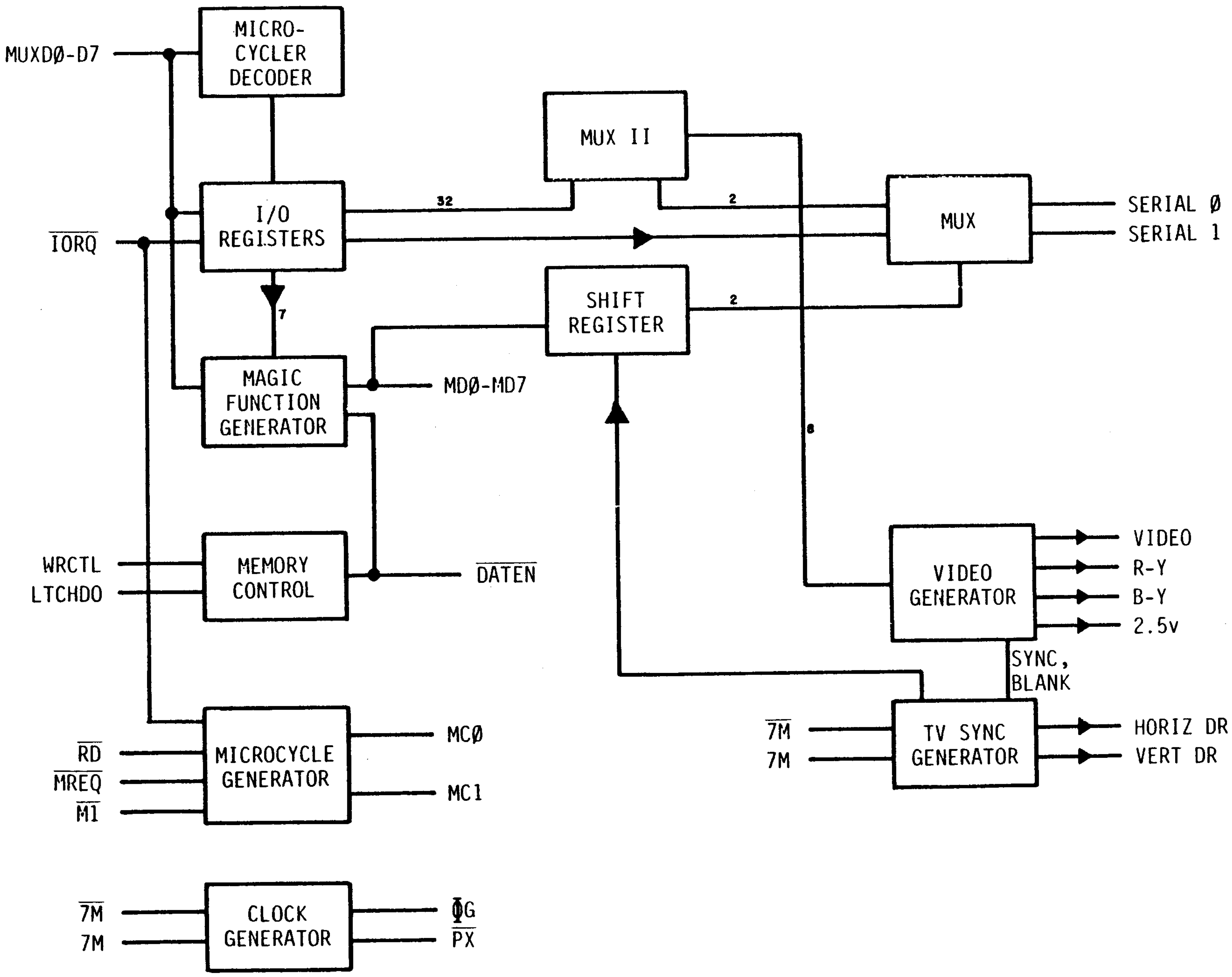
The Microcycle Generator generates the microcycle control bits, MCØ and MC1, from $\overline{IORQ}$, $\overline{MREQ}$, $\overline{RD}$, and $\overline{MI}$, all from the Z-8Ø.

In memory write cycles WRCTL is activated and the Memory Control circuit generates $\overline{DATEN}$. The Magic Function Generator takes the data from the Z-8Ø on MUXDØ - D7 and transfers it to MDØ - MD7. If a Magic write is being done, the Magic Function Generator will modify the data as required before it places it on the memory data bus.

A Magic write is a memory write cycle in which data is written to a location, (X) from Ø to 16K. All memory from Ø to 16K is ROM and cannot be modified. The data is modified by the Magic Function Generator and is written to location X + 16K. The way in which the data is modified is determined by the 7 bits coming from the I/O registers.

In memory reads, data is transferred from MDØ - MD7 to MUXDØ - MUXD7. Also, LTCHDO is activated which causes the data from RAM to be latched up in a register in the Magic Function Generator. This latched data is used in some magic functions.

The I/O registers are loaded by output instructions from the Z-8Ø just as in the Address Chip.



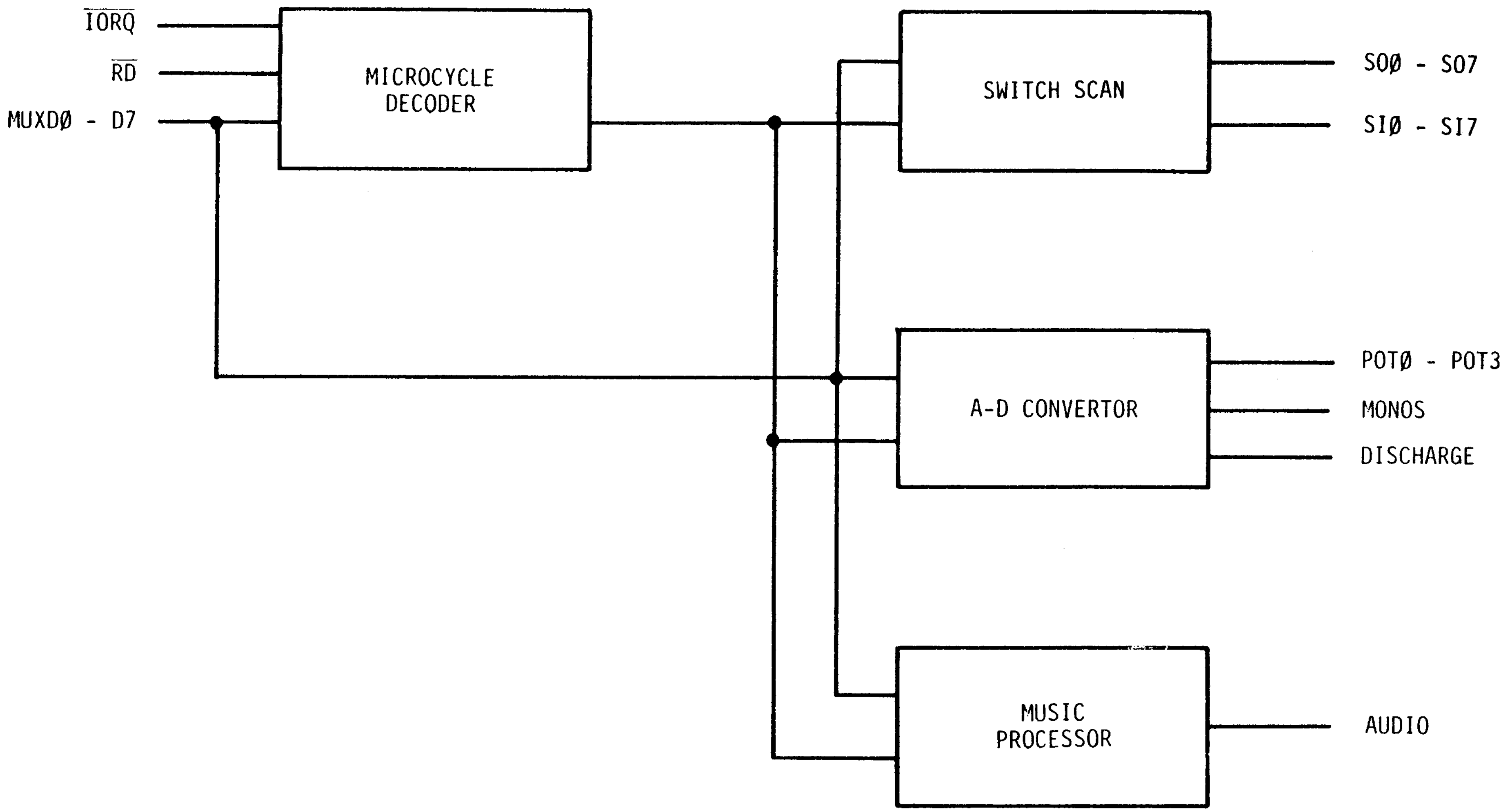
DATA CHIP BLOCK DIAGRAM

I/O CHIP DESCRIPTION

The Z-8Ø communicates with the I/O Chip through input and output instructions. The state of an 8 x 8 switch matrix can be read through the Switch Scan circuit. When an input instruction is executed, one of the SOØ-SO7 lines will be activated. When a line is activated, the switch matrix will feed back eight bits of data on SIØ-SI7. This data is in turn fed to the Z-8Ø through MUXDØ - MUXD7.

The Z-8Ø can read the position of four potentiometers (pots) through the A-D Converter circuit. The pots are continuously scanned by the A-D Converter and the results of the conversions are stored in a RAM in the A-D Converter circuit. The Z-8Ø simply reads this RAM with input instructions.

The Z-8Ø loads data into the Music Processor with output instructions. This data determines the characteristics of the audio that is generated. The Music Processor is described in detail below.



I/O CHIP BLOCK DIAGRAM

MUSIC PROCESSOR

The music processor can be divided into two sections. The first section generates the Master Oscillator Frequency and the second section uses the Master Oscillator Frequency to generate tone frequencies and the analog audio output. The contents of all registers in the Music Processor are set by output instructions from the Z-8Ø.

Master Oscillator Frequency is a square wave whose frequency is determined by the 8 binary inputs to the Master Oscillator. This 8-bit word is the sum of the contents of the Master Oscillator Register and the output of the MUX. The MUX is controlled by MUX REG.

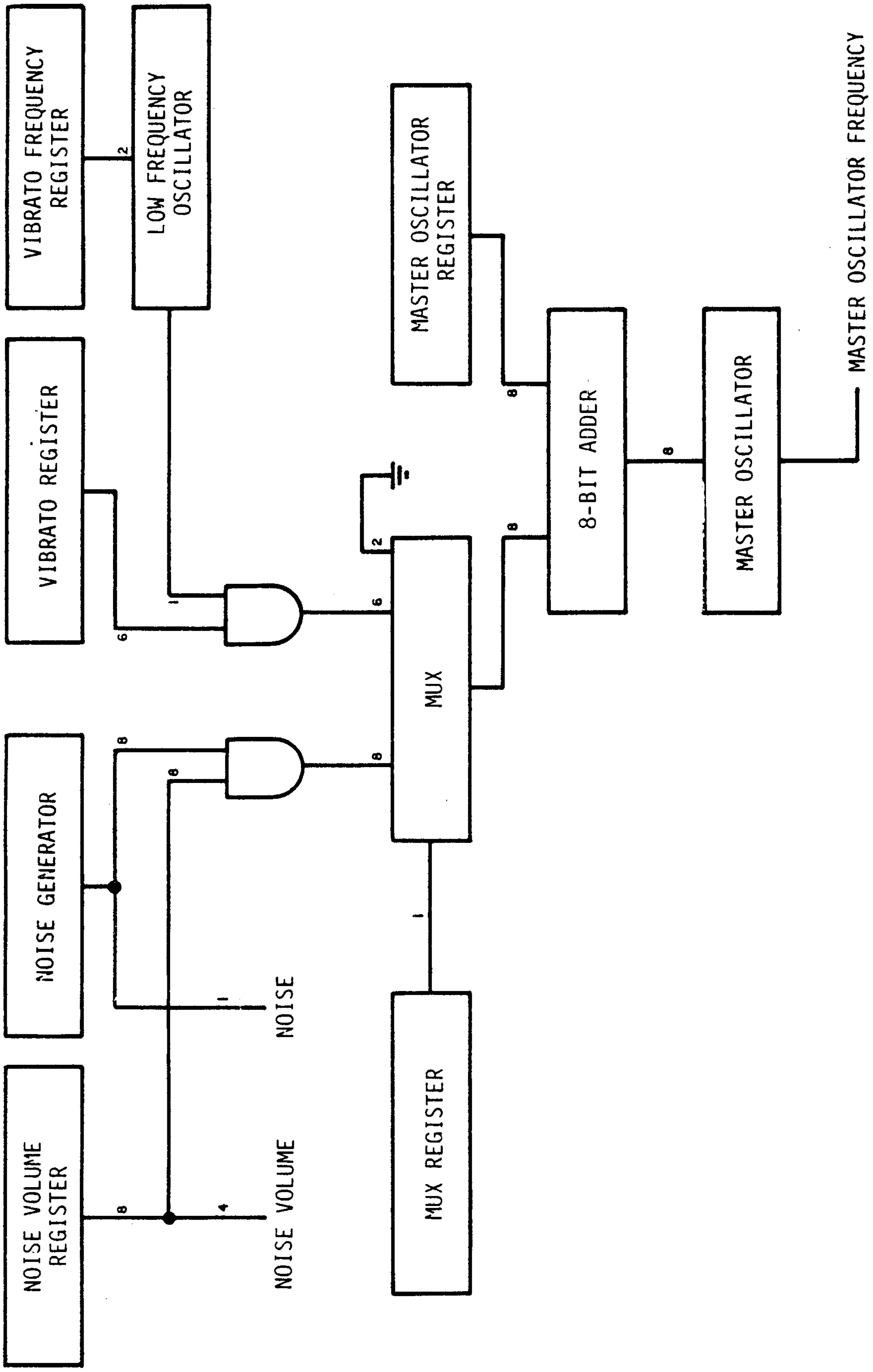
If MUX REG contains Ø,then data from the Vibrato System will be fed through the MUX. The two bits from the Vibrato Frequency Register determine the frequency of the square wave output of the Low Frequency Oscillator. The 6-bit word at the output of the AND gates oscillates between Ø and the contents of the Vibrato Register. The frequency of oscillation is determined by the contents of the Vibrato Frequency Register. The 6-bit word, along with two ground bits are fed through the MUX to the Adder. This causes the Master Oscillator Frequency to be modulated between two values thus giving a vibrato effect.

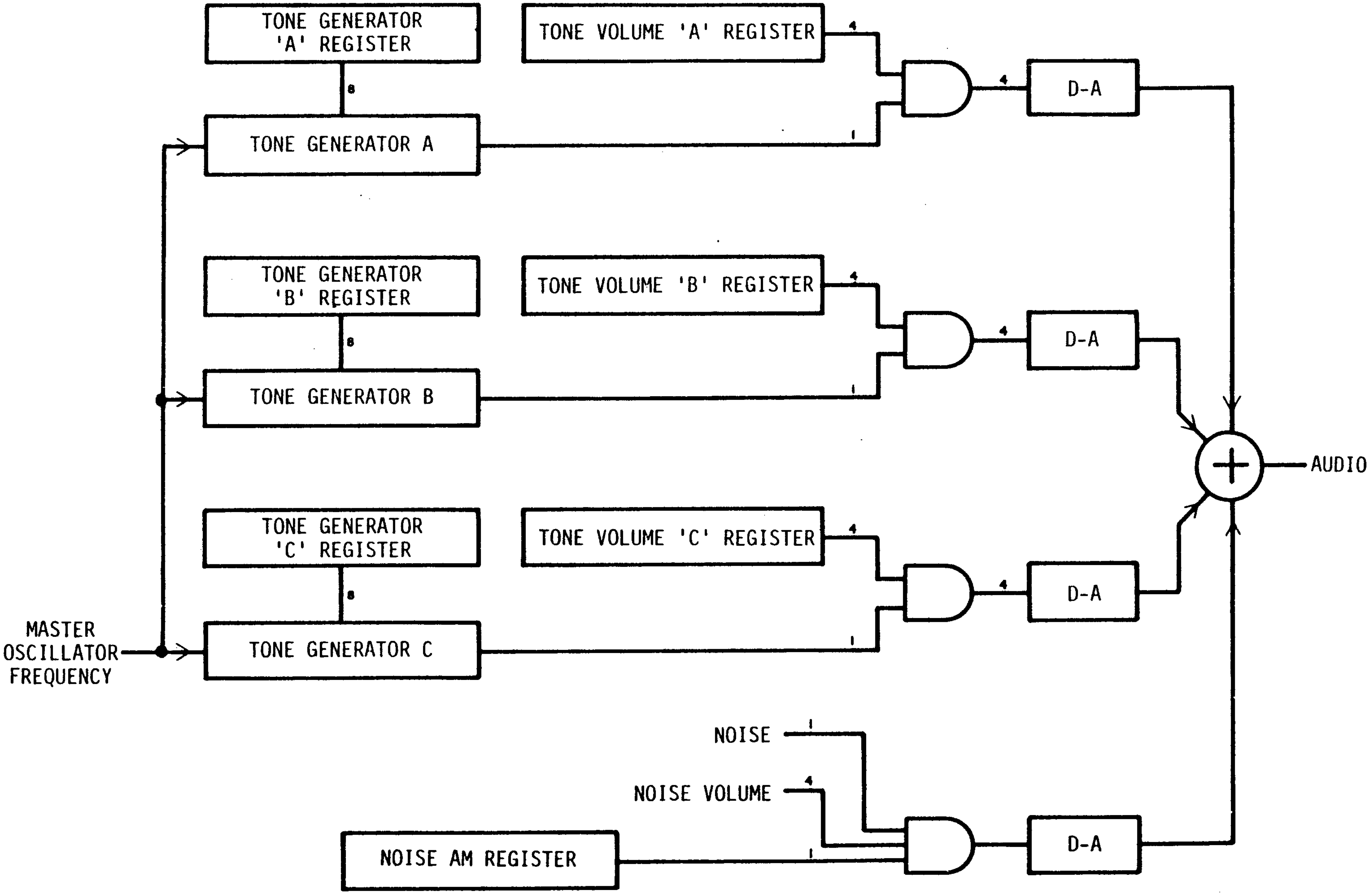
If MUX REG contains 1 ,then data from the Noise System will be fed through the MUX. The 8-bit word from the Noise Volume Register determines which bits from the Noise Generator will be present at the output of the AND gates.

If a bit in the Noise Volume Register is Ø, then the corresponding bit at the output of the AND gates will be Ø. If a bit in the Noise Volume Register is 1, then the corresponding bit at the output of the AND gates will be noise from the Noise Generator. This 8-bit word is sent through the MUX to the Adder. The Master Oscillator Frequency is modulated by noise.

In the second part of the Music Processor, the square wave from the Master Oscillator is fed to three Tone Generator circuits which produce square waves at their outputs. The frequency of their outputs is determined by the contents of their Tone Generator Register and Master Oscillator Frequency. The 4-bit words at the output of the AND gates oscillate between Ø and the contents of the Tone Volume Register. These 4-bit words are sent to D-A Converters whose outputs oscillate between GND and a positive analog voltage determined by the contents of the Tone Volume Register.

One Noise bit and four Noise Volume bits from the first section of the Music Processor are fed to a set of AND gates. This set of AND gates operates the same way as the AND gates for the tones, except that the Noise AM Register must contain a 1 for the outputs of the AND gates to oscillate. The analog outputs of the four D-A Converters are summed to produce the single audio output.


VIBRATO FREQUENCY REGISTER
LOW FREQUENCY OSCILLATOR
2
VIBRATO REGISTER
6
1
6
2
MASTER OSCILLATOR REGISTER
8
8-BIT ADDER
8
8
MASTER OSCILLATOR
MASTER OSCILLATOR FREQUENCY
NOISE GENERATOR
8
8
8
MUX
1
NOISE
1
NOISE VOLUME REGISTER
8
4
NOISE VOLUME
MUX REGISTER



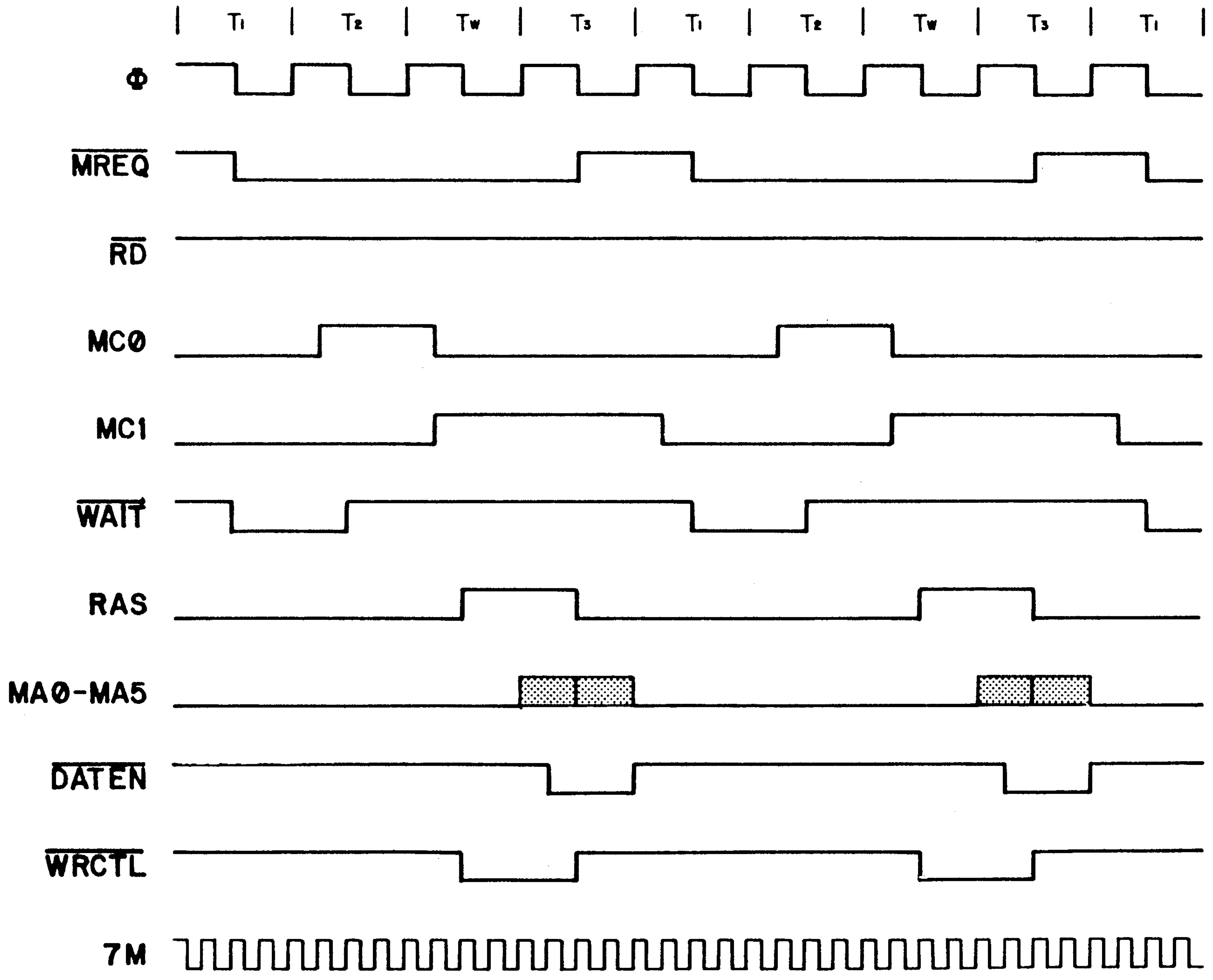
TONE GENERATORS

CUSTOM CHIP TIMING

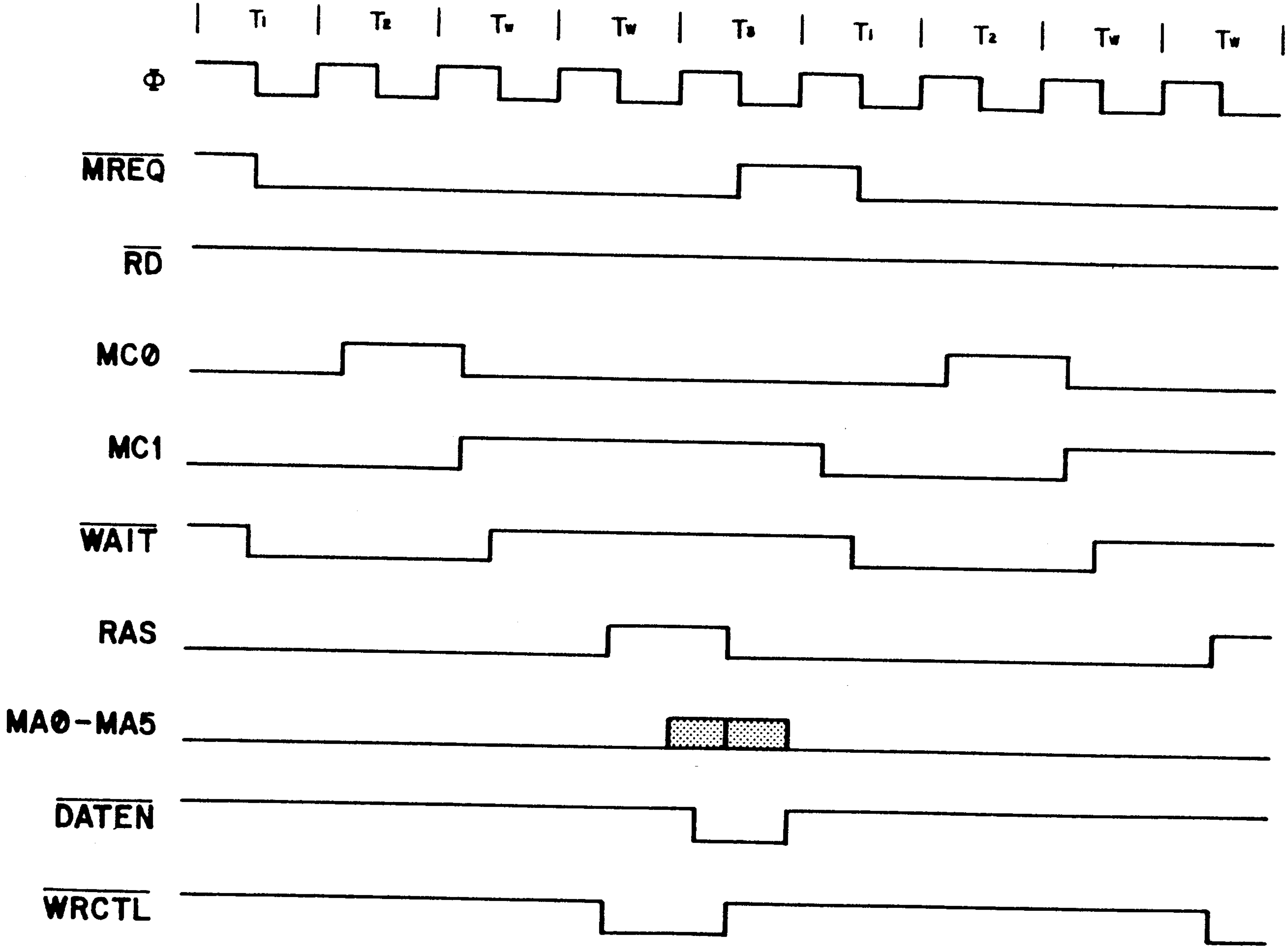
The following diagrams show the relationship of various signals in the system during different types of operations. Delays are shown to be zero nsec from the clock edge which causes the transition. The actual delay is given in "Electrical Specification for Midway Custom Circuits".

MUXDØ - MUXD7 is a 8-bit bidirectional address and data bus for the custom chips. By using this technique 16 bits of address and 8 bits of data can be sent to the custom chips on 8 wires. The state of the bus is determined by MCØ and MC1 from the data chip and $\overline{\text{RFSH}}$ from the Z-8Ø.

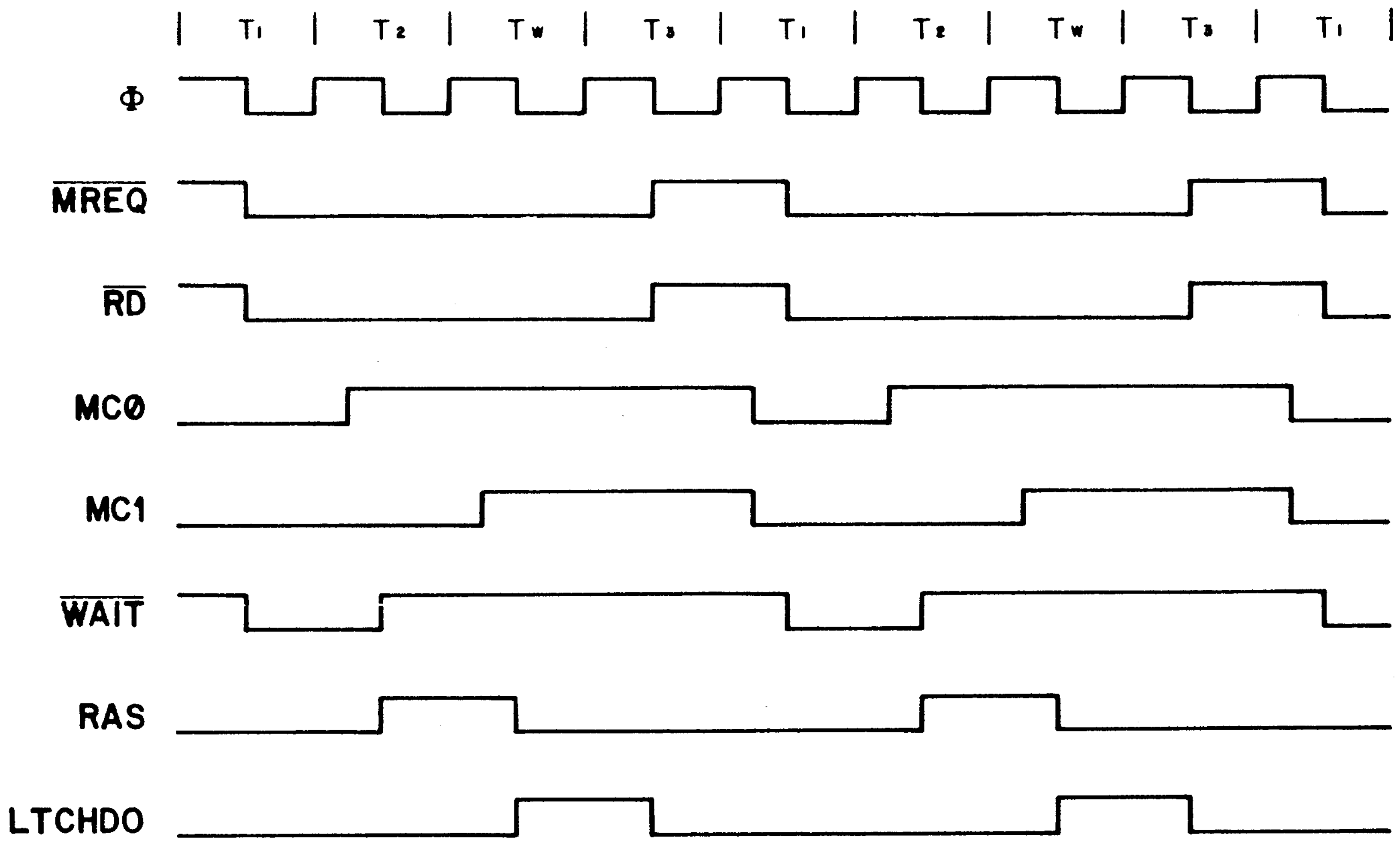
$\overline{\text{RFSH}}$	MC1	MCØ	
L	L	L	AØ - A7 to custom chips.
L	L	H	AØ - A7 to custom chips
L	H	L	AØ - A7 to custom chips
L	H	H	AØ - A7 to custom chips
H	L	L	AØ - A7 to custom chips
H	L	H	A8 - A15 to custom chips
H	H	L	DØ - D7 to custom chips
H	H	H	DØ - D7 from custom chips



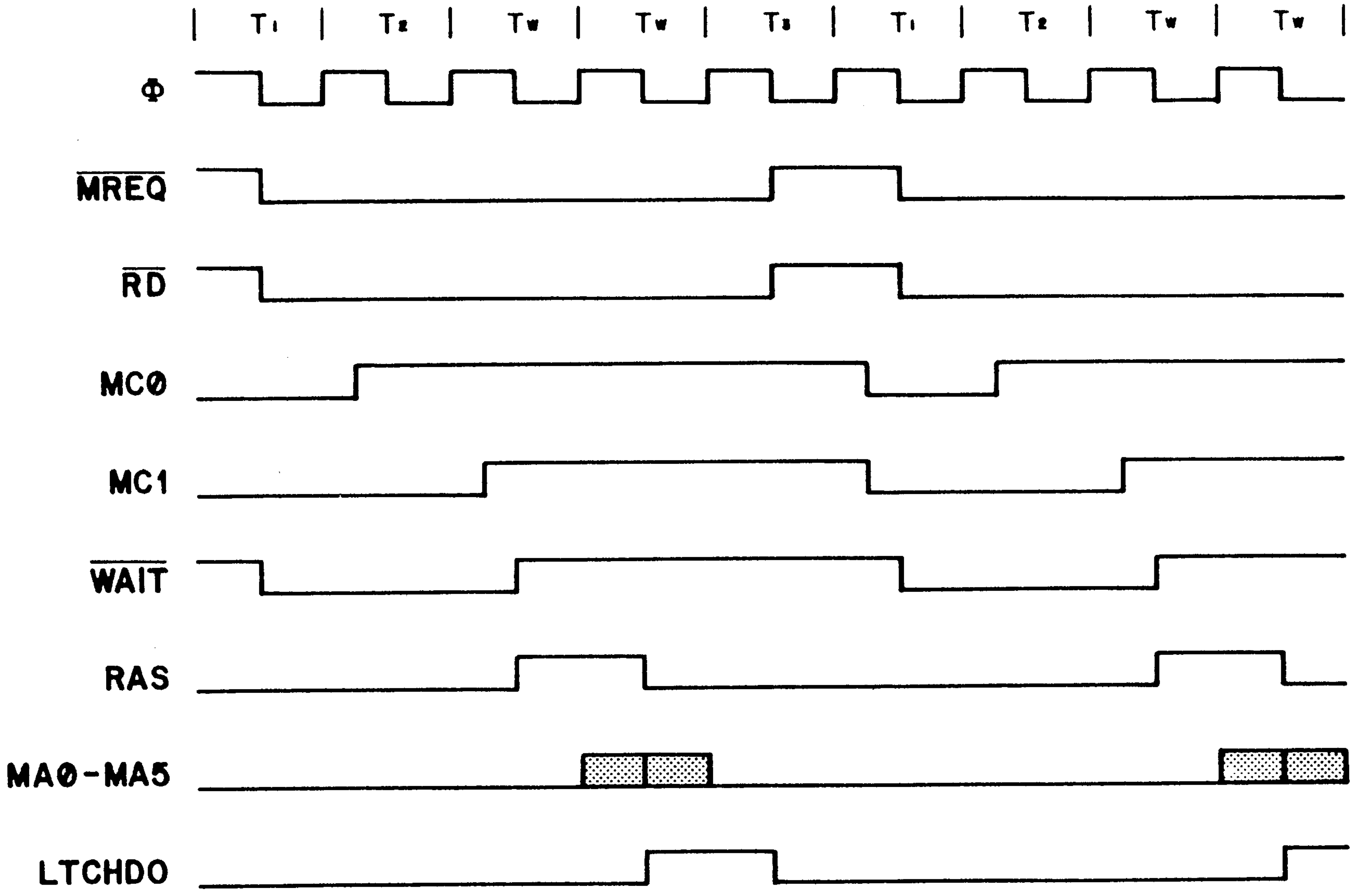
MEMORY WRITE WITHOUT EXTRA WAIT STATE



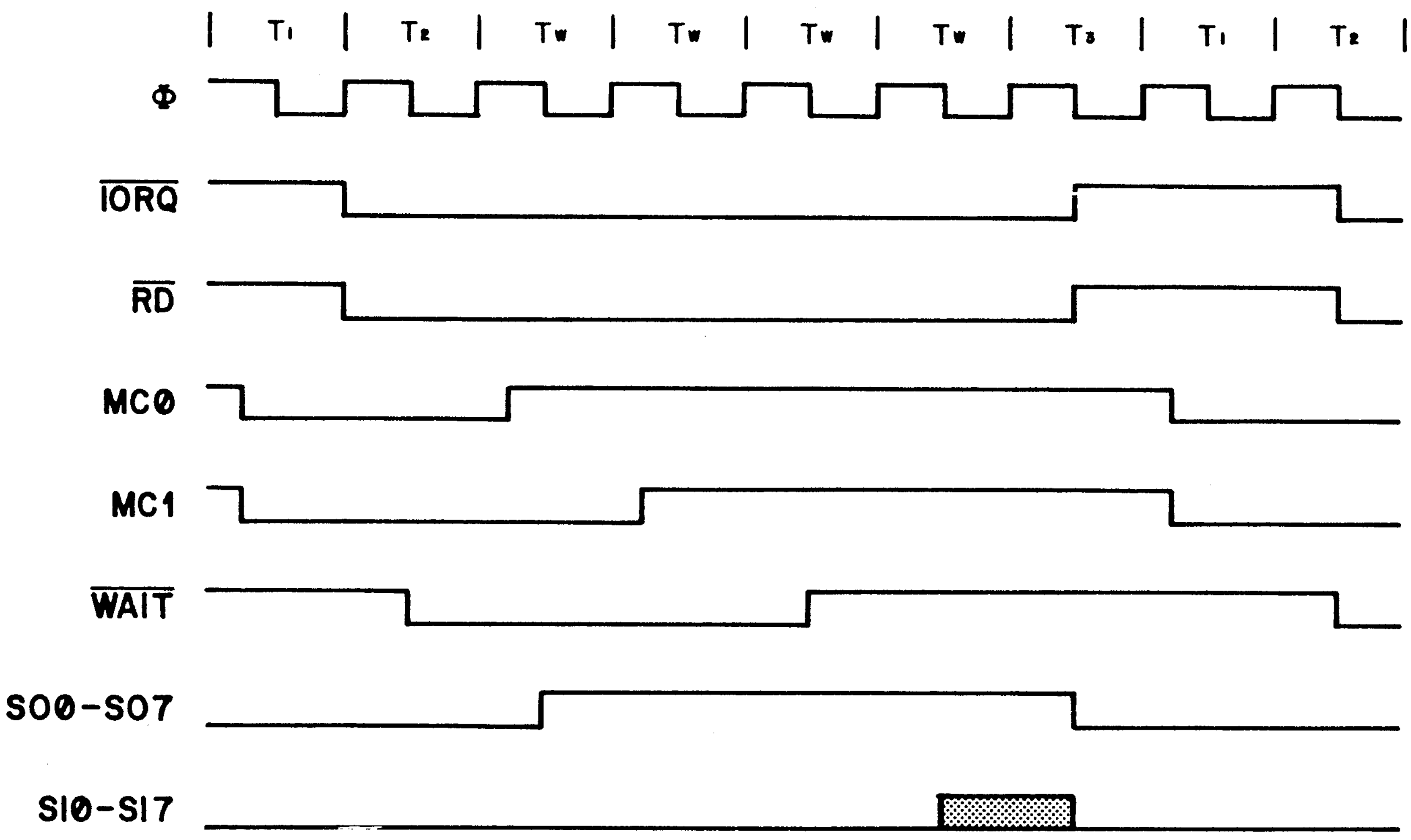
WRITE WITH VIDEO WAIT STATE



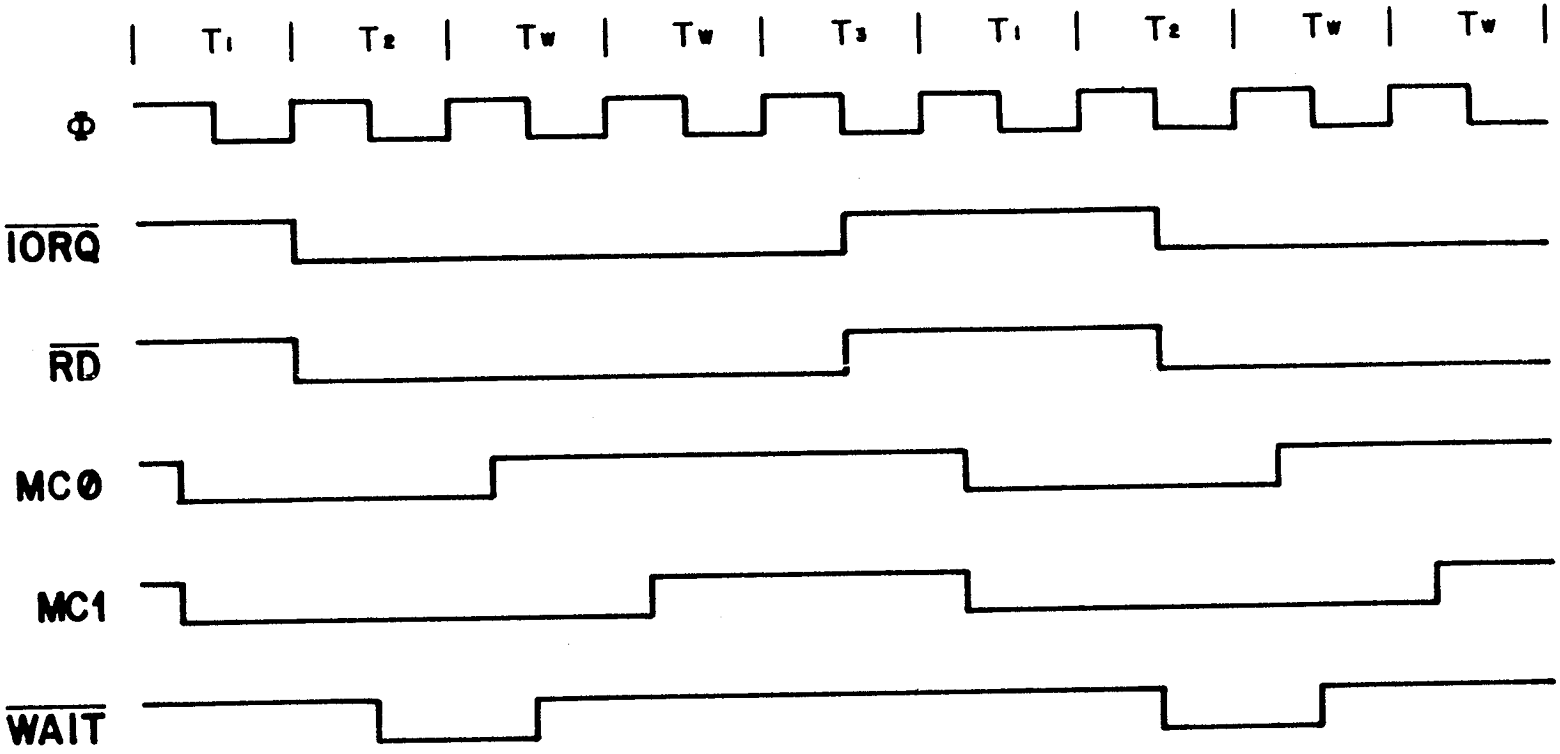
MEMORY READ WITHOUT EXTRA WAIT STATE

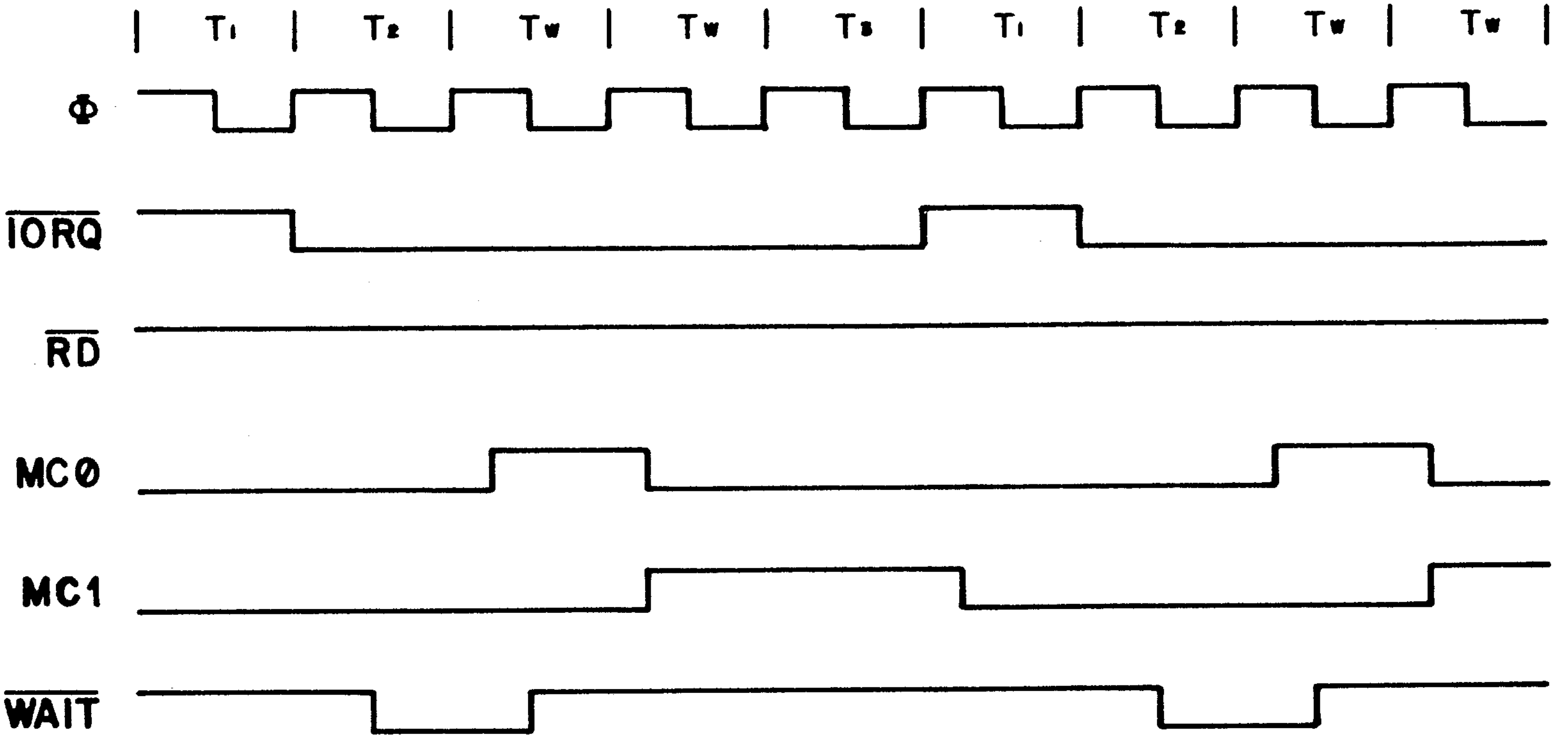


MEMORY READ WITH VIDEO WAIT STATE



I/O READ FROM PORT 10H-17H

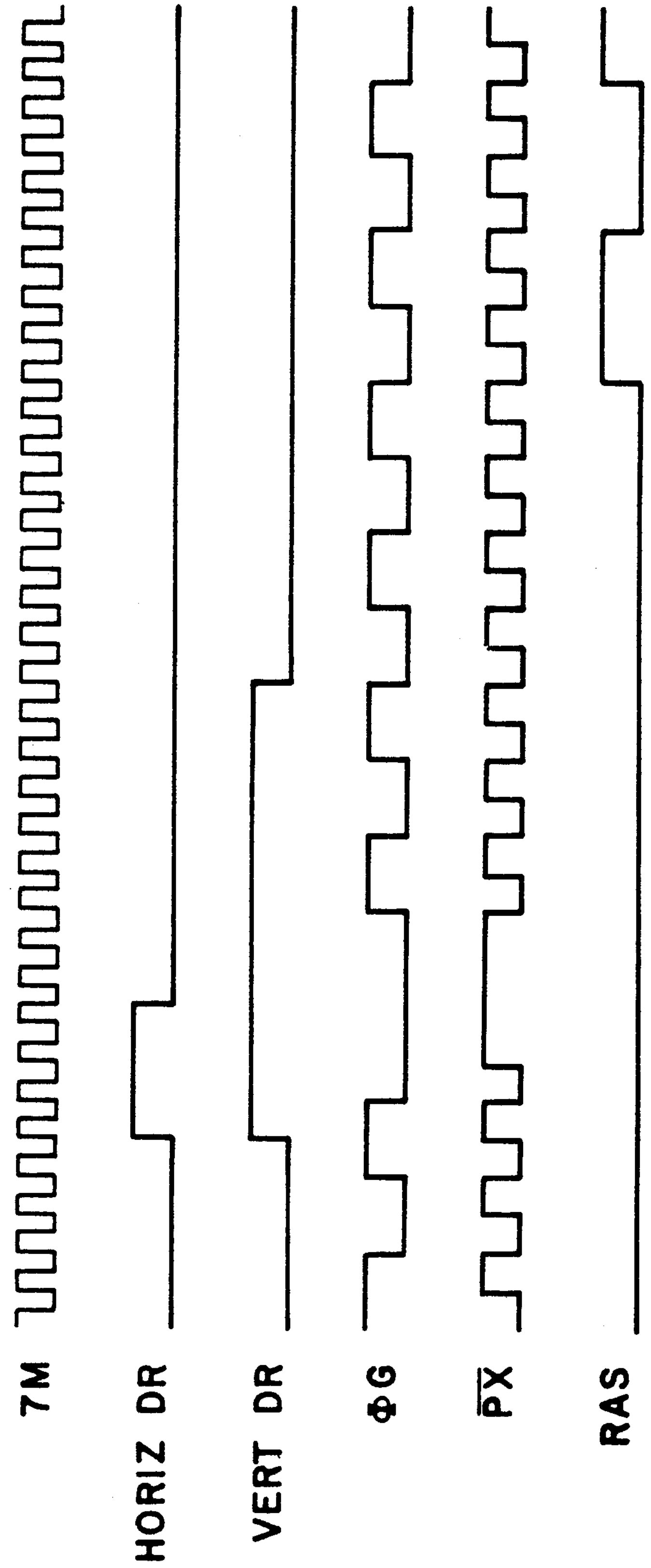

T1
T2
Tw
Tw
T3
T1
T2
Tw
Tw
Φ
IORQ
RD
MCØ
MC1
WAIT


T1
T2
Tw
Tw
T3
T1
T2
Tw
Tw
Φ
$\overline{\text{IORQ}}$
$\overline{\text{RD}}$
MCØ
MC1
$\overline{\text{WAIT}}$

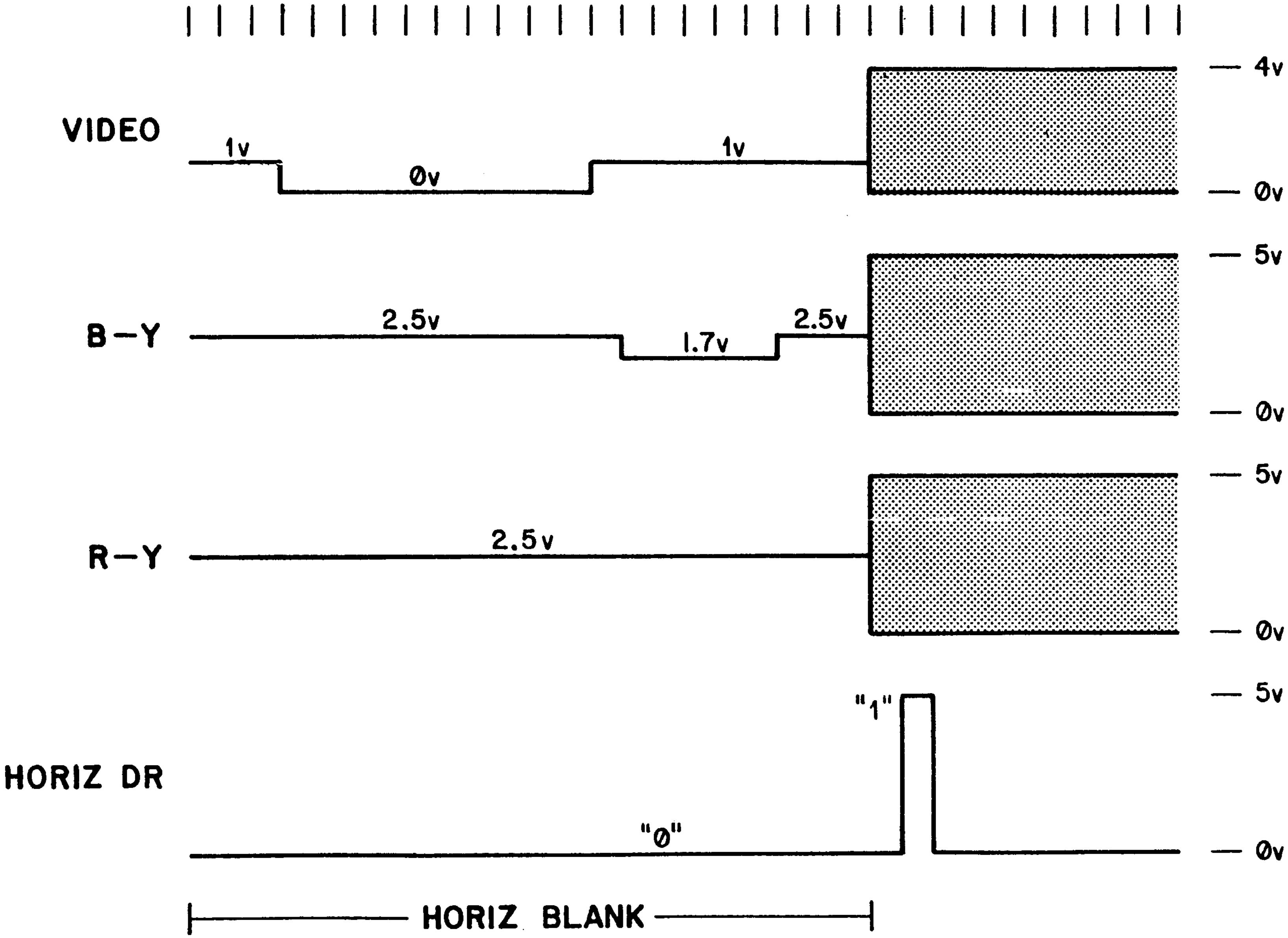
I/O WRITE

VIDEO TIMING

The frequency of $\overline{PX}$ is half that of 7M and the Ø is one-fourth 7M. There are 455 cycles of 7M per horizontal line and 113 3/4 Φ cycles per line. Because of the extra 3/4 cycle Ø must be resynchronized at the beginning of each line. This is done by stalling Ø for 3 cycles of 7M. $\overline{PX}$ is also stalled for the same amount of time. The timing relationship is shown below. The diagram also shows the relationship of VERT DR to HORIZ DR. The two RAS pulses shown are the first two video RAS signals of a line, each line contains forty.



RELATIONSHIP BETWEEN 7M, HORIZ DR, VERT DR, ΦG, $\overline{\text{PX}}$ AND RAS

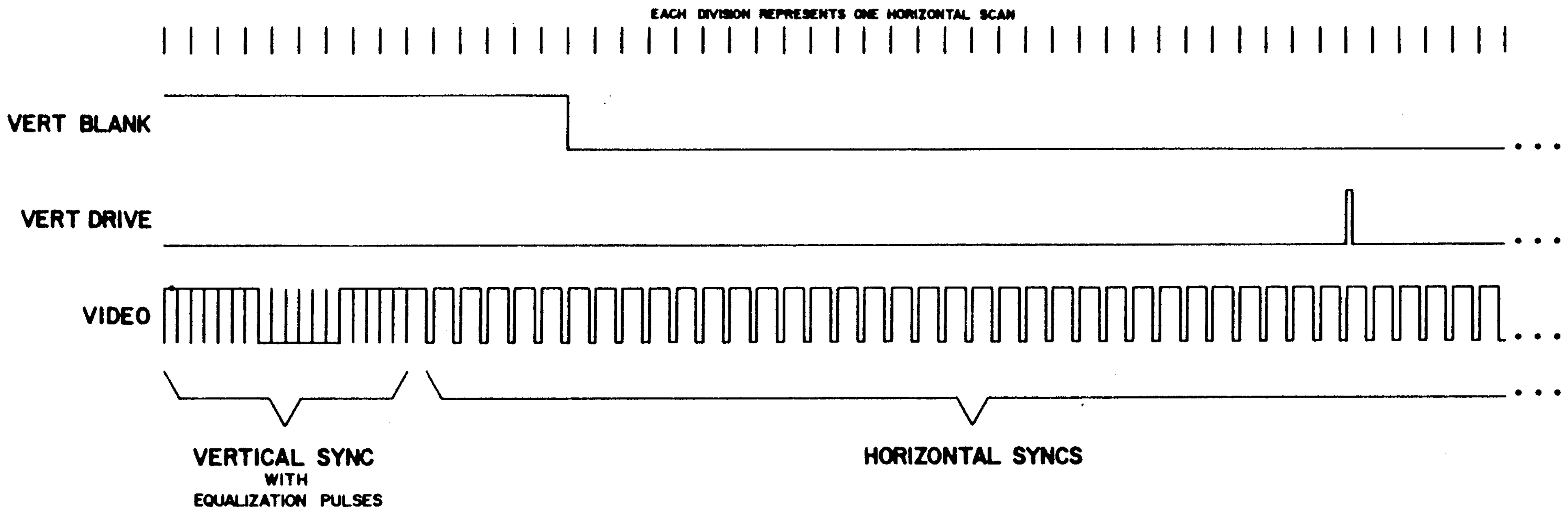


RELATIONSHIP BETWEEN HORIZ DR, HORIZ BLANK, HORIZ SYNC AND COLOR BURST

EACH HORIZONTAL DIVISION IS EQUAL TO 3½ CYCLES OF 7M

THE PATTERN REPEATS EVERY 455 CYCLES OF 7M

SHADED AREA VOLTAGE DETERMINED BY THE DATA IN RAM



RELATIONSHIP BETWEEN VERTICAL SYNC, VERTICAL BLANK AND VERTICAL DRIVE

EACH HORIZONTAL DIVISION REPRESENTS ONE HORIZONTAL SCAN

1/14/77
1/27/77
3/25/77
7/6/77

ELECTRICAL SPECIFICATION FOR MID[illegible] CUSTOM CIRCUITS

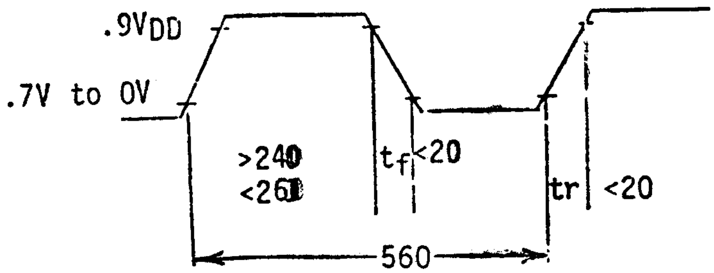
I. GENERAL SYSTEM PARAMETERS

I. A. Power Supplies

1. VDD=+5.0V ±5%
2. VGG=+10.0V ±5%
3. VSS=0.0V

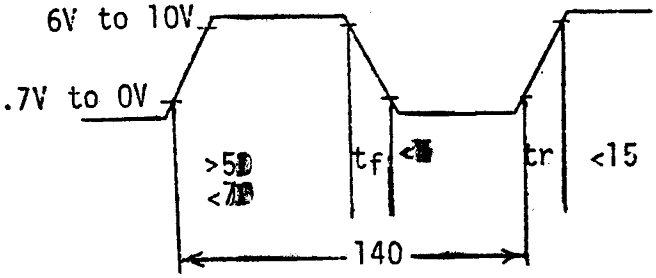
I. B. Timing Signals

1. Ø & $\overline{Ø}$; Period = 560nsec, High time* 240nsec to 260nsec.
   Ø and $\overline{Ø}$ have zero level crossover +1 volt -0 volts
   t_r, t_f* less than 20nsec



(Times are in

2. 7M & $\overline{7M}$; Period = 140nsec, High time$^+$ 50nsec to 70nsec
   7M & 7M have zero level crossover +1 volt -0 volt
   t_r, $t_f$$^+$ less than 15nsec



(Times are in

Dead time ≤ 5nsec
Max C Load = 20pf

+Note
1) High time is time clock at ≥ .6V.
2) Rise time for zero level to one level.

I. B. (Continued)

*Note:

1. High time is time between 50% points.

2. Clock signals are generated by low power Shottky Logic (series 74LS). Full level swing on clock signals to be achieved through external resistor to V_{DD}. Zero level .7V to 0V.

3. Rise time from zero level to .9V_{DD}.

I. C. <u>Z80 Data Bus (MUXDØ-MUXD7)</u>

1. Z80 Data Bus interface requires a three-state output/input buffer. The three states are defined below.

2. Logic 0: .5V + noise generated by chip, noise for address chip is .15V @ -430μA

3. Logic 1: 2.7V @ +70μA

4. High Impedance: Leakage at either logic 0 or 1 to be less than 5μA.

5. Transient Response: Transition from High Impedance to 0 or 1 will be complete within 442nsec of the 90% point of $\overline{Ø}$ of the last wait state of input cycle or 442nsec of the 90% point of Ø of the second wait state of the interrupt acknowledge cycle. The maximum load will be 80pf. This includes 14pfd for two custom chips.

6. Exception: The path through the Data chip connecting the RAM bus with the Z80 bus shall introduce a maximum of 160nsec of delay.

7. The low address byte will be valid on the Z80 Data Bus at least 62nsec before $\overline{Ø}$. The high address byte will be valid at least 79nsec before $\overline{Ø}$. The data byte will be valid 55nsec before $\overline{Ø}$.

I. D. RAM Data Bus (MD0-MD7) - Home Game

1. The RAM Data Bus will require three state logic buffers.

2. Logic 0: .5V @ -25μA

3. Logic 1: 2.7V @ +25μA

4. High Impedance: 5μA maximum leakage at either logic 0 or 1.

5. Transient Response: The outputs shall transition from High Impedance to 0 or 1 within 120nsec of 7M. The outputs shall transition from 1 or 0 to high impedance within 20nsec of 7M. Maximum load will be 20pf.

I. E. RAM Data Bus (MD0-MD7) - Commercial Game

1. The RAM Data Bus will require three state logic buffers.

2. Logic 0: .5V @ -200μA

3. Logic 1: 2.7V @ +25μA

4. High Impedance: 5μA maximum leakage of either logic 0 or 1.

5. Transient Response: The output shall transition from High Impedance to 0 or 1 within 120nsec of 7M. The output shall transition from 1 or 0 to High Impedance within 2nsec of 7M. Maximum load will be 10pf.

I. F. Ambient operating temperature $\geq 0^{o}C$, $\leq 55^{o}C$.

I. G. Storage temperature $\geq -65^{o}C$, $\leq 150^{o}C$.

I. H. Packing 40 pin plastic.

II. CUSTOM CIRCUIT SPECIFICATION

This specification defines the terminal characteristics for each of the custom circuits. These specifications shall take precedence in case of conflict. All Ø references refer to the Ø and $\overline{\text{Ø}}$ inputs to the address and I/O chip.

II. A. Data Chip

1. Input Pin List

Input Pin List	VO	V1	t_d (Low)[1]	t_d (High)[1]	Ref.
	(V)	(V)	(nsec)	(nsec)	
$\overline{\text{MREQ}}$	.5	2.45	132	6	7M
$\overline{\text{RD}}$	.5	2.45	12	6	7M
$\overline{\text{IORQ}}$	.5	2.45	112	126	7M
7M	See Section I.B.				
$\overline{\text{7M}}$	"	"			
$\overline{\text{WRCTL}}$	.5	3.1	82	82	7M
$\overline{\text{M1}}$	.5	2.45	12	82	7M
LTCHDO	.5	3.1	120	120	7M
Serial 0	.5	2.45	30	30	7M
Serial 1	.5	2.45	30	30	7M

2. Power Supplies

See Section I. A.

3. Bus Connections

MXD0	See Z80 Data Bus Spec. Section	I.C.
MXD1	"	"
MXD2	"	"
MXD3	"	"
MXD4	"	"
MXD5	"	"
MXD6	"	"
MXD7	"	"
MD0	See RAM Data Bus Spec Section	I.D.
MD1	"	"
MD2	"	"
MD3	"	"
MD4	"	"
MD5	"	"
MD6	"	"
MD7	"	"

4. Outputs	VO (V)	IO (μA)	V1 (V)	I1 (μA)	CAP (pf)	t_p (nsec)	Ref.
VIDEO*	*				10	100	7M
R-Y*	*				10	600	
B-Y*	*				10	600	
HORIZ DR	Note 4	400	2.7	20	20	20	7M
VERT DR	Note 4	400	2.7	20	20	20	7M
2.5V[6]	--	--	--	--	--	DC	
Ø	Note 4	400	2.7	20	10	100	7M
$\overline{PXCLK}$	Note 4	400	2.7	20	10	100	7M
MCO	Note 4	400	2.7	20	10	120	7M
MC1	Note 4	400	2.7	20	10	120	7M
$\overline{DATEN}$	Note 4	400	2.7	20	10	90	7M

*Video, R-Y, B-Y are analog outputs at 140nsec rate. Video, must switch from 10% to 90% of blank to white in 140nsec. R-Y and B-Y transitions not to exceed .6μsec.

[1] t_d (Low) and t_d (High) is maximum time in nsec except where a minimum is shown.

[2] For $\overline{IORQ}$ Ref. to $\overline{Ø}$ t_d (Low)=132nsec t_d (High)=6nsec.

[3] Serial 0 and Serial 1 will operate at 7MHz.

[4] .5V + noise generated by chip.

[5] Tap on both resistor chains for a capacitor. Will become test input with voltage applied > 8V.

[6] The Z80 Ø is generated by this signal with a clock driver which introduces a delay of <20nsec.

II. B. I/O Chip

1. Input Pin List

Input Pin List	VO	V1	Ref	t_d (High) (nsec)	t_d (Low) (nsec)
Reset	.5	2.45			
MONOS	Note 1				
$\overline{RD}$	.5	2.45	Ø or $\overline{Ø}$	166	172 Ø or $\overline{Ø}$
$\overline{IORQ}$	.5	2.45	Ø^6	146 $\overline{Ø}$	132 Ø
Ø	See Section I.B.				
$\overline{Ø}$	"	"	"		
SIØ	.5	3.3			Note 3
SI1	.5	3.3			Note 3
SI2	.5	3.3			Note 3
SI3	.5	3.3			Note 3
SI4	.5	3.3			Note 3
SI5	.5	3.3			Note 3
SI6	.5	3.3			Note 3
SI7	.5	3.3			Note 3
TEST	.5	5.0			DC

2. Power Supplies

See Section I.A.

3. Bus Connections

MUXD0 See Z80 Data Bus Spec Section I.C.
MUXD1 " "
MUXD2 " "
MUXD3 " "
MUXD4 " "
MUXD5 " "
MUXD6 " "
MUXD7 " "

4. Outputs

Outputs		VO (V)	IO (μA)	V1 (V)	I1 (μA)
Audio	Note 4	Fmax - 20KHz			
Discharge	Note 5	.5V	4V		
SOØ	Note 3	Note 7	200	4V	1650
SO1	Note 3	Note 7	200	4V	1650
SO2	Note 3	Note 7	200	4V	1650
SO3	Note 3	Note 7	200	4V	1650
SO4	Note 3	Note 7	200	4V	1650
SO5	Note 3	Note 7	200	4V	1650
SO6	Note 3	Note 7	200	4V	1650
SO7	Note 3	Note 7	200	4V	1650
POT Ø	Note 2		5	V_{DD}-.5	50
POT 1	Note 2		5	V_{DD}-.5	50
POT 2	Note 2		5	V_{DD}-.5	50
POT 3	Note 2		5	V_{DD}-.5	50

Note 1 MONOS triggers at 2.1 volts ±2% ± noise voltage when the supply is 5.25V.

Note 2 Open source-Voltage measured with 0.2ma.

Note 3 Time from load of address into microcycle register to date valid on MUX data bus from SI inputs (data path through address decoder, out on SO outputs, through closed switch and isolation diode, into SI input to MUX Data Bus) shall be 2μsec max. Drop of isolation diode will be 0.7V max. SO must drive 2kΩ in the high level. Max C load of SO shall be 300 pf. SI input shall have kill device enabled by $\overline{\text{INPUT}}$.

Note 4 Audio voltage oscillates between 0V and one of the following voltages; .33, .67, 1.00, 1.33, 1.67, 2.00, 2.33, 2.67, 3.00, 3.33, 3.67, 4.00,·4.33, 4.67 and 5.00. These voltages should be ±6%. The load shall be 1000pf and 100kΩ.

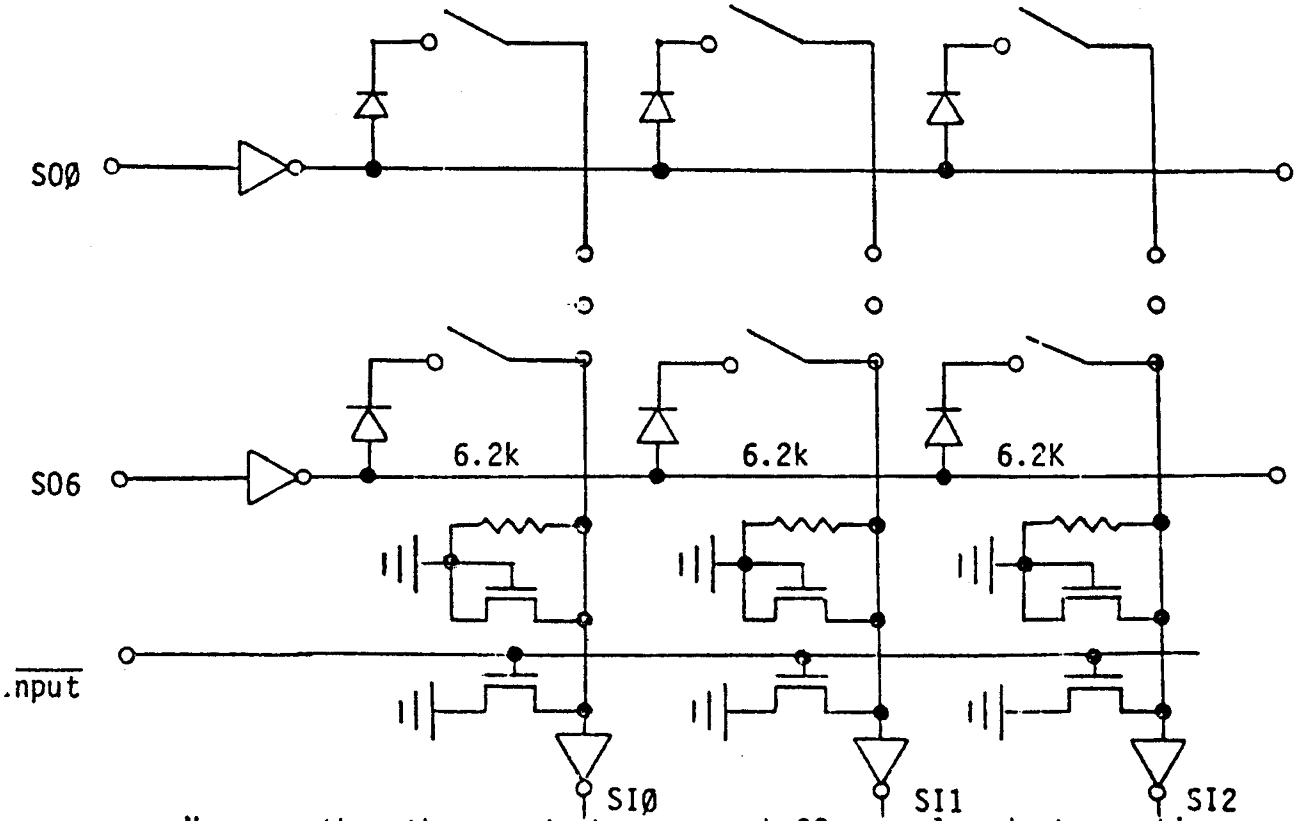
Note 5 Discharge is open drain to V_{SS}. Discharges .01μfd capacitor to .2V in 144μsec.

Note 6 For $\overline{\text{IOREQ}}$ Ref. to $\overline{\emptyset}$ t_d (Low)=152nsec t_d (High)=166nsec.

Note 7 .5V + noise generated by I/O chip.

Miscellaneous Timing

Time for MO Adder - 2∅ max



No more than three switches on each SO are closed at one time.

II. C. Address Chip

1. Input Pin List

Input Pin List	VO (V)	V1 (V)	tpd (Low) (nsec)	tpd (High) (nsec)	REF
$\overline{\text{RFSH}}$	.5	2.45	222 Ø	216	Ø
$\overline{\text{MREQ}}$	.5	2.45	152 $\overline{\text{Ø}}$	166	Ø or $\overline{\text{Ø}}$
$\overline{\text{RD}}$	.5	2.45	172 Ø or $\overline{\text{Ø}}$	166	Ø or $\overline{\text{Ø}}$
$\overline{\text{MI}}$	.5	2.45	176 Ø	242	Ø
A12[1]	.5	2.45			Ø
A13[1]	.5	2.45			Ø
A14[1]	.5	2.45			Ø
A15[1]	.5	2.45			Ø
$\overline{\text{IORQ}}$	.5	2.45	132 Ø	146	$\overline{\text{Ø}}$[2]
$\overline{\text{LIGHT PEN}}$	.5	2.45	Asyn		
$\overline{\text{TEST}}$	.5	5.0	DC		
HORIZ. DR.	.5	2.45	Note 3		$\overline{\text{Ø}}$
VERT. DR.	.5	2.45	Note 4		Ø
Ø	See Section I.B.				
$\overline{\text{Ø}}$	"	"	"		

2. Power Supplies

See Section I.A.

3. Bus Connections

MXD0	See Z80 Data Bus Spec Section I.E.
MXD1	" "
MXD2	" "
MXD3	" "
MXD4	" "
MXD5	" "
MXD6	" "
MXD7	" "

4. Outputs

Outputs	VO (V)	IO (μA)	V1 (V)	I1 (μA)	CAP (pf)	tpd(Low) (nsec)	tpd(High) (nsec)	REF
LATCHDO	Note 7	Note 6	3.1	Note 6	10	280	140	$\overline{\text{Ø}}$[5]
$\overline{\text{WAIT}}$	" "	400	2.4	20	25	490	490	$\overline{\text{Ø}}$
MA0-MA5	" "	400	2.4	20	20	242	240	$\overline{\text{Ø}}$ or Ø
$\overline{\text{INT}}$	" "	400	2.4	20	25	490	572	Ø
RAS0-RAS3	" "	400	2.4	20	20	382	382	$\overline{\text{Ø}}$
$\overline{\text{WRCTL}}$	" "	Note 6	3.1	Note 6	10	382	382	$\overline{\text{Ø}}$

1. Time from High Impedance to 1 or 0 is 200nsec. (from $Ø_1$ of T_1)
2. For $\overline{\text{IORQ}}$ Ref to $\overline{\text{Ø}}$ td (Low)=152nsec td (High)=166nsec. Ø
3. Horizontal Drive time from low to high is 40nsec after $\overline{\text{Ø}}$.
   Time from high to low is 100nsec before rising edge of Ø.
4. Vertical Drive will transition from low to high 40nsec after falling edge of Ø. Its width will be 2.1 μsec max. 1.54μsec min. It will go from high to low 100nsec before falling edge of Ø.
5. Reference tpd (High) is Ø.
6. MOS to MOS signal.
7. .5V + noise generated by Address C[illegible] (.15V) = .65V

III. I/O MODE DECODE

I/O Parts

HEX	Out	Input
0	Color Ø Right	
1	" 1 "	
2	" 2 "	
3	" 3 "	
4	" 0 Left	
5	" 1 "	
6	" 2 "	
7	" 3 "	
8	Consumer/Commercial	Intercept Feedback
9	Horiz Color Bndry	
A	Vertical Blank	
B	Color Block TX	
C	Magic Reg	
D	Interrupt Feedback	
E	Interrupt Mode	Vertical Addr Feedback
F	Interrupt Line	Horizontal Addr Feedback
10	Tone Master OSC	SW Bank 0
11	Tone A	1
12	" B	2
13	" C	3
14	Tremello	4
15	Tone C Volume	5
16	Tone A,B Volume	6
17	Noise Volume	7
18	Sound Block TX	
19		
1A		
1B		
1C		POT 0
1D		" 1
1E		" 2
1F		" 3
20		
21		
22		
23		
24		
.		
.		
2F		