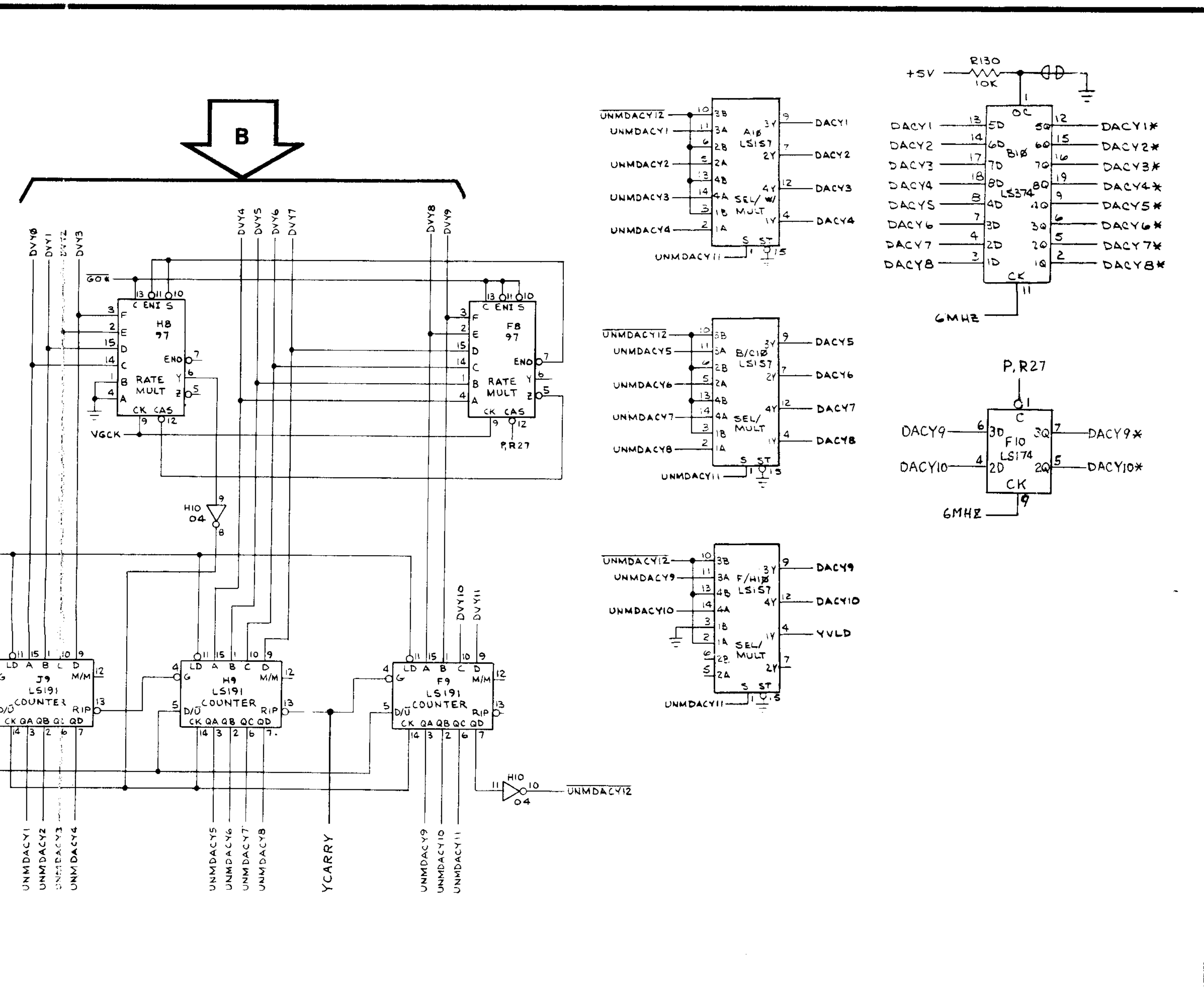
Latch 1 is cleared by ALPHANUM.

ndicated revision



ers count up or down. DVX11 is used
multiplexers D10, E10, and F10.
ACX10 (X axis unmultiplexed digital-
are transferred to the output of the
outputs of the latches on each rising
the microcomputer clock circuitry).
gnals are sent to the digital-to-analog
eo output.
outputs represent the physical place-
tor. The far left of the monitor screen
e far right is 1023. Therefore, if the
was greater than 1023, the monitor
ide of the screen and start again on
wraparound" condition. To prevent a
select input from UNMDACX11 goes
than 1023 or less than 0. This selects
n the multiplexers to the DACs, forc-
thus keeping the beam on the ap-
nstead of allowing it to wraparound.
valid) outputs from the X and Y posi-
atched and gated together to enable
valid).

Sheet 2, Side A

COCKTAIL ASTEROIDS

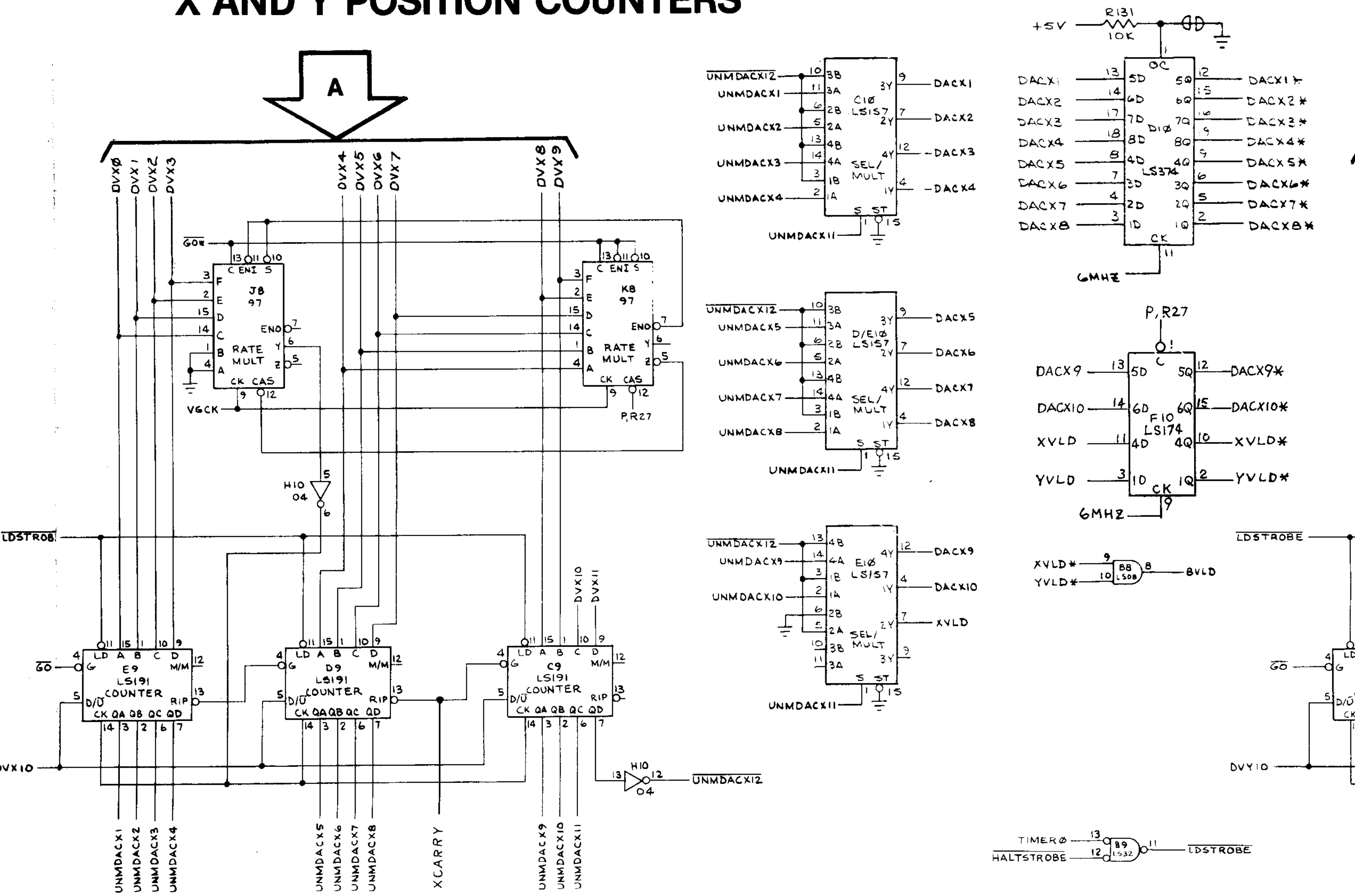
Video Generator

Section of 034986-XX G

ATARI®

denotes change by indic

X AND Y POSITION COUNTERS



ounter: are two identical circuits. criptiondiscusses only the X position

contai rate multipliers (J8 and K8), d E9), multiplexers (D10, E10, and F10), nd H10. The output of the down/up umber that represents the horizontal monitor screen (or X axis), with 0 being and 1023 being the far right side of the asing this binary number output will e right or left, respectively. The vector codes instructions from its memory, that data to alter the binary count of ways.

eset these counters to an entirely dif- evious contents. This will cause the cation on the monitor screen instan- new vector from a different starting ious vector ended. While the beam is "jumping" to this new position, the beam itself is turned off to prevent unwanted lines from appearing on the screen. To preset this new position into the counters, the state generator causes $\overline{\text{LDSTROBE}}$ to go low. At this time, a new 12-bit number (DVX0-11) is loaded into the counters from the vector generator memory data latches.

The state machine can also instruct these counters to count up or down any specific number of counts. This will cause the beam to move to the left or to the right a specific distance relative to where it was. During this beam movement, the beam is turned on with the desired intensity. This is the procedure used to draw a vector on the monitor screen. The direction (to the left or right) and length (0 to 1023) of the vector to be drawn relative to the beam's current position is determined by DVX0-11 (from the vector generator memory data latches). This data contains information that determines how many clock pulses the counters will receive and whether the counters will count up or down.

DVX0-9 memory data is loaded into rate multipliers J8 and K8. The function of these devices is to space the desired number of counter clock pulses at equal intervals over the time period that it will take to draw the desired vector. This insures that vectors of different lengths will still be displayed with the same relative beam intensity. DVX10 and 11 are loaded directly into the counters. DVX10 determines whether the counter to control the select input of m

The UNMDACX1 thru UNMDA to-analog converter signals) ar multiplexers and stored at the o edge of the 6 MHz clock (from The DACX1* thru DACX10* sign converters (DACs) in the X vide

The DACX1* thru DACX10* ou ment of the beam on the monito is 0, the center is 512, and the DACX1* thru DACX10* signal v beam would go off the right sid the left side of the screen, a "w wraparound, the multiplexers' s high when the count is greater t UNMDACX12 to be output from ing all zeros or all ones, and t propriate side on the screen, in

The XVLD and YVLD (X and Y v tion counter multiplexers are la the Z axis output, BVLD (beam xis out-

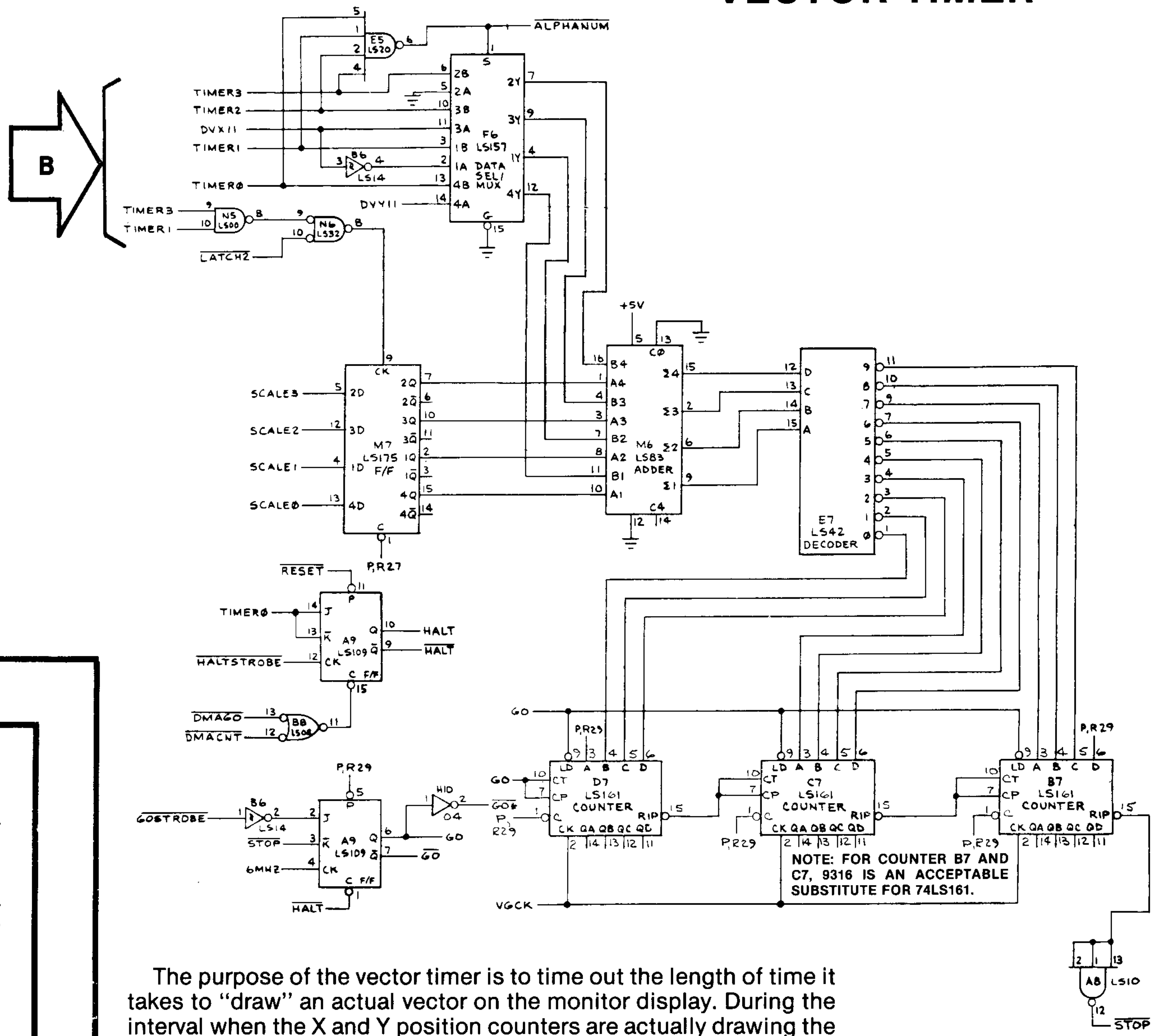
ts in a
ars the
0 thru
ns exe-
0.

le for a
ing the
ation.

d the X
vector
nput to
r timer
hine is
he vec-
te from
hrough
nd GO

.5 MHz
me fre-
er. The
ressing
s low).
gh until

VECTOR TIMER

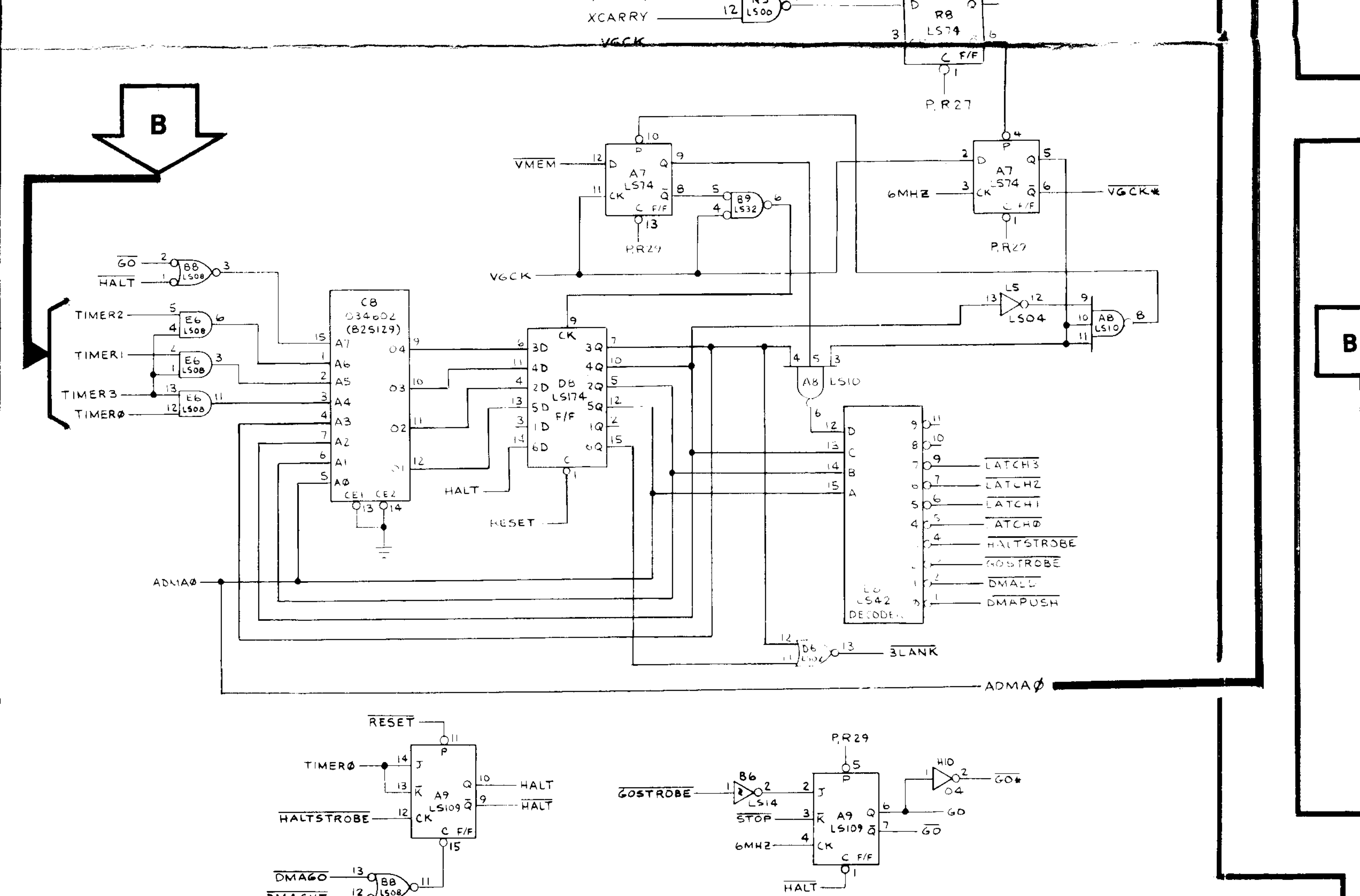


The purpose of the vector timer is to time out the length of time it takes to "draw" an actual vector on the monitor display. During the interval when the X and Y position counters are actually drawing the vector, STOP is high. This prevents the vector generator state machine from advancing to its next state until the vector currently being drawn is completed. As soon as the vector has been drawn, STOP goes low, allowing the state machine to advance to the next state in its intended sequence.

The vector timer consists of multiplexer F6, decoder E7, LATCH M7, ADDER M6, and counters B7, C7, and D7. M7 contains a scale factor which is added in M6 to the four timer signals. If TIMER0 thru TIMER3 inputs are any state but all high, decoder E7 directly decodes the sum and loads the decoded low into one of the counters. When GO goes low, the counters count from the loaded count until the counters all reach their maximum count. This count is a maximum length of 1024. At this time STOP goes low and clears the GO flip-flop of the state machine.

If the TIMER signals are all high, ALPHANUM goes low and data signals DVX11 and DVY11 are decoded by decoder E7. This is added to the scale factor and loaded into the counters.

The X and Y position counter
Therefore, the following descriptior
counters.
The X position counters contain
down/up counters (C9, D9 and E9), mul
and associated gates (B8 and H10
counters is a 12-bit binary number th
location of the beam on the monitor s
the far left side of the screen and 10:3
screen. Increasing or decreasing thi
cause the beam to move to the right or
generator state machine decodes ns
and then is capable of using that data
these counters in one of two ways.
The state machine can preset thse
ferent number from their previous co
beam to "jump" to a new location or
taneously, i.e., for drawing a new ve
position than where the previous vect



STATE MACHINE

The state machine is the "master controller" of the vector generator circuitry. It receives instructions from the game MPU, via the vector generator RAM. It carries out these instructions by accessing the appropriate sections of the vector generator ROM memory, using the vector generator program counter to do so. The state machine reads the vector generator ROM data (via Timer 0-3) and decodes this information to determine how it should use this data: 1) to draw a vector; 2) to move the monitor beam to a new position on the monitor display; 3) to "jump" to a new vector memory address; 4) to return to a previous vector memory address; or 5) to tell the game MPU that it has completed its current instructions, and is waiting for its next command.

The state machine consists of input gates B8 and E6, ROM C8, latch D8, clock circuitry A7, and decoder E8. Four bit input TIMER0 thru TIMER3 is the operation code input to the state machine. The A4 thru A6 address input to ROM C8 tells the ROM which instructions to perform. Address inputs A0 thru A3 from latch D8 tells the ROM which state was last performed. The address A7 input $\overline{\text{GO}}$ tells the ROM that the position counters are presently drawing a vector. The $\overline{\text{HALT}}$ input to A7 tells the ROM that the vector generator has completed its operations.

During initial power-up of the game, the $\overline{\text{HALT}}$ signal is preset low. The microcomputer reads the high HALT signal through its switch input port (buffer M10) on data line DB0. This tells the microcomputer that the vector generator is halted and waiting for an instruction. To ensure that the beam is off when the state machine is halted, the high HALT, clocked through latch D8, results in a low $\overline{\text{BLANK}}$ to the Z axis output.

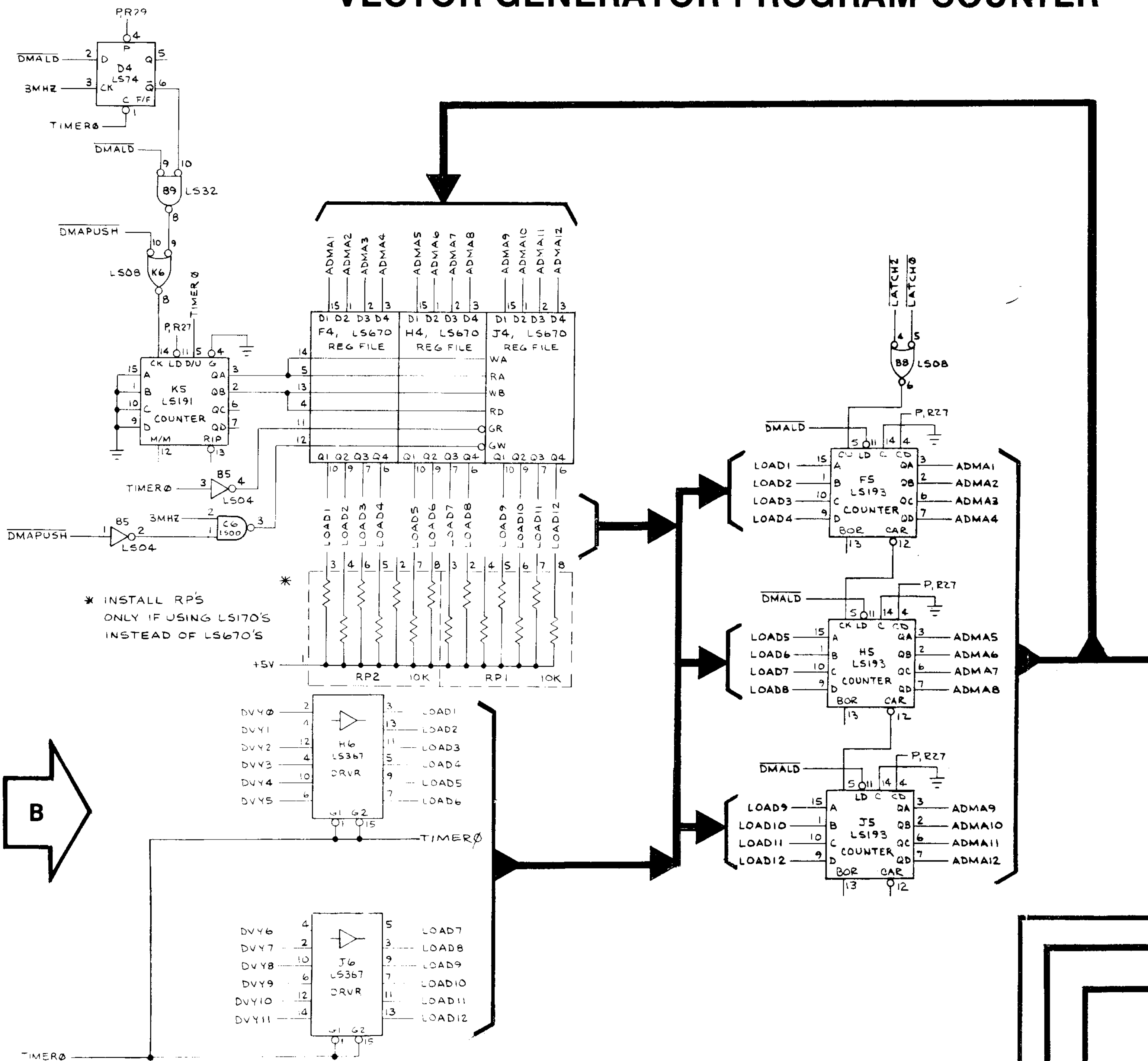
The microcomputer outputs an address that results in a $\overline{\text{DMAGO}}$ signal that causes $\overline{\text{HALT}}$ to go high, and clears the vector generator data latches. This makes TIMER0 thru TIMER3 signals all low. The state machine now begins executing instructions, starting at vector memory location 0.

When the state machine receives the operation code for a $\overline{\text{HALT}}$ instruction, it outputs a low $\overline{\text{HALTSTROBE}}$, setting the HALT flip-flop A9, and suspending state machine operation.

The GO signals load and enable the vector timer and the X and Y position counters and tell the ROM that the vector generator is now actively drawing a vector. The $\overline{\text{HALT}}$ input to GO flip-flop A9 sets the outputs to ensure that the vector timer and position counters are not active when the state machine is halted. When a low $\overline{\text{GOSTROBE}}$ is clocked through A9, the vector timer and X and Y position counters begin to operate from the GO, $\overline{\text{GO}}$ and $\overline{\text{GO*}}$ signals. When $\overline{\text{STOP}}$ is clocked through A9, the vector timer has reached its maximum count, and $\overline{\text{GO}}$ goes high. This means the vector has been drawn.

The VGCK input to the clock circuitry is a buffered 1.5 MHz clock signal from the microcomputer. This is the same frequency used to clock the MPU of the microcomputer. The signal clocks latch D8 unless the microcomputer is addressing the vector RAM or ROM memories (when $\overline{\text{VMEM}}$ goes low). Then the clock input to latch D8 goes high and stays high until $\overline{\text{VMEM}}$ goes high.

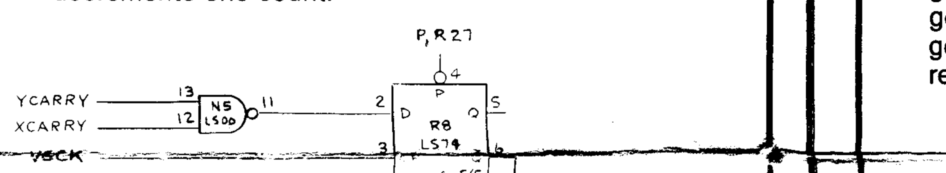
VECTOR GENERATOR PROGRAM COUNTER



Counters F5, H5 and J5 contain the address of the next data byte (instruction) to be fetched from the Vector Generator memory. Because these counters point to the next instruction in memory to be retrieved and performed, they are called the program counter. This program counter is incremented one count (to the next sequential address) each time the information at its current address is loaded into data latch 0 or data latch 2.

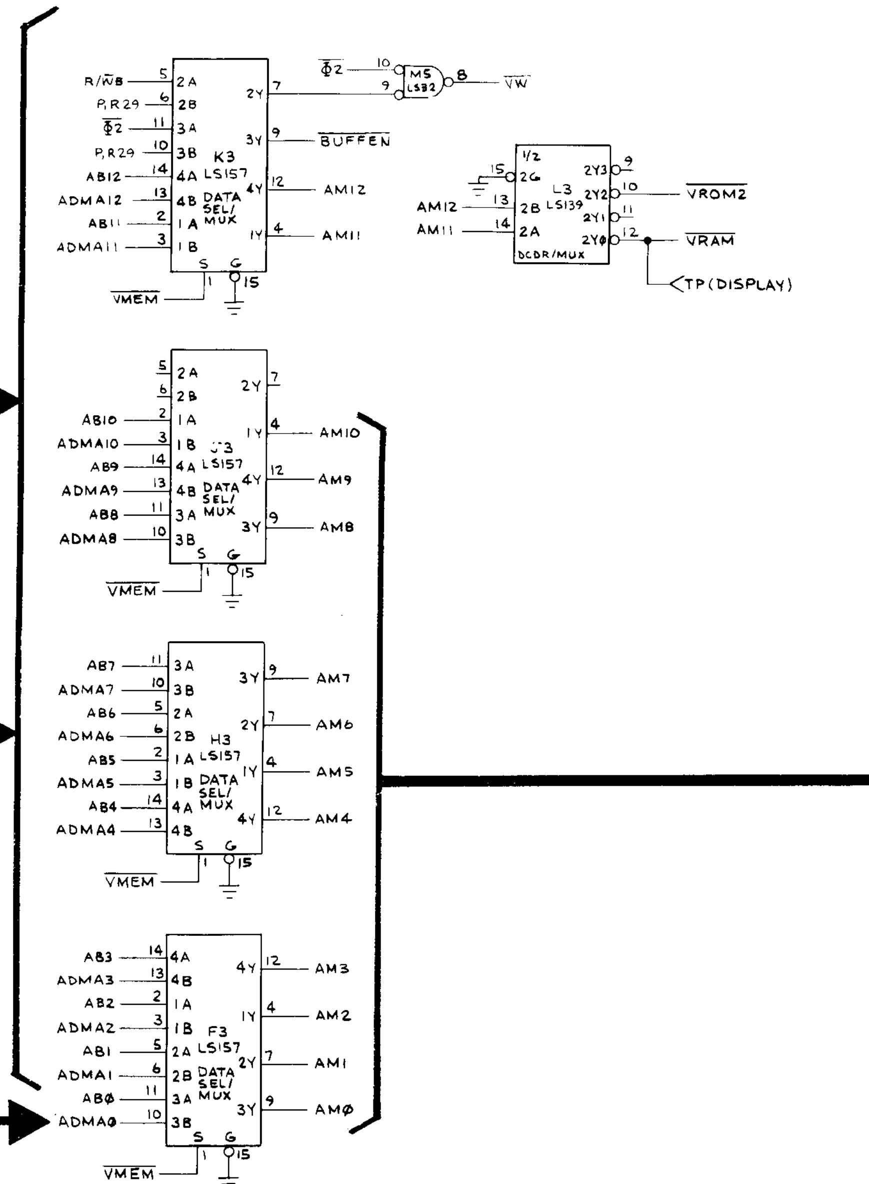
The program counter may also be preset to "jump" to a new address. This new address can be loaded into the program counter from the vector generator memory via data latches F7 and H7 and buffers H6 and J6.

The program counter may also be preset to "return" to a previous address which it had stored in its "stack". The stack consists of register files F4, H4, & J4, and down/up counter K5. The stack is a 4-word 12-bit memory, used to save the contents of the program counter for future reference. It is loaded when $\overline{\text{DMAPUSH}}$ is low. Immediately after information is written into the stack, counter K5 increments one count. Immediately before loading the program counter from the stack, counter K5 decrements one count.



FROM
MICROCOMPUTER
SHEET 1, SIDE B

VECTOR GENERATOR MEMORY ADDRESS SELECTOR

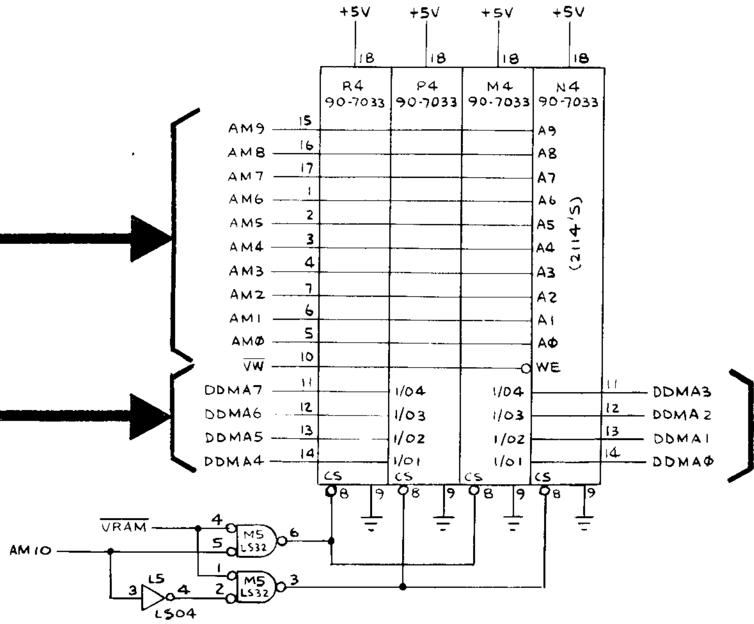


The address selector consists of multiplexers F3, H3, J3 and K3. When $\overline{\text{VMEM}}$ is low, the MPU of the microcomputer gains access to the address inputs of the vector generator memory. In this state, $\overline{\text{BUFFEN}}$ is from $\overline{\Phi 2}$ and $\overline{\text{VW}}$ (vector generator write) is low when $\overline{\Phi 2}$ and $\overline{\text{R/WB}}$ are both low. When $\overline{\text{VMEM}}$ is high, the address input to the vector generator memory is from the vector generator program counter and state machine. In this state, $\overline{\text{BUFFEN}}$ and $\overline{\text{VW}}$ are both held high by the pullup resistors connected to the 2B and 3B inputs of multiplexer K3.

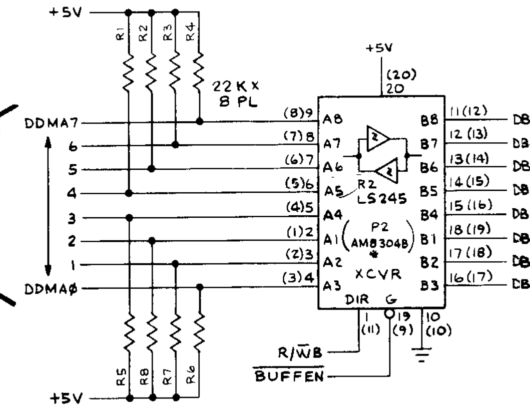
Address decoder L3 decodes address bits A11 and A12, and selects the RAM or one of three ROMs of the vector generator memory.

This address-selecting arrangement allows the game MPU to access the vector generator memory, i.e., write data into the vector generator RAM to instruct the vector generator what it should do next. The address selector can then allow the vector generator program counter and state machine to access this same area of RAM also, and read what instructions were sent to it by the game MPU.

ENERATOR RAM

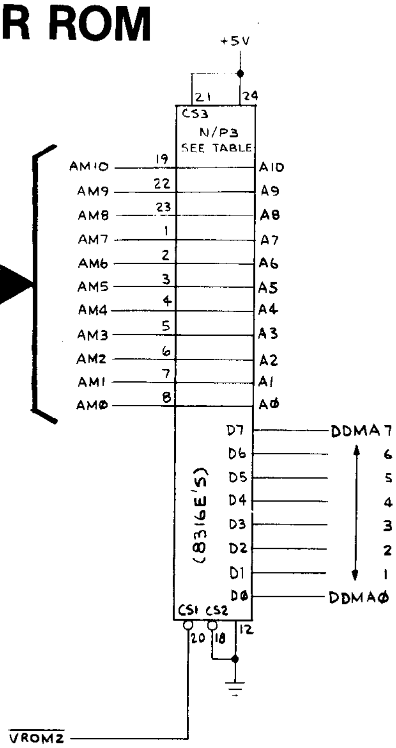


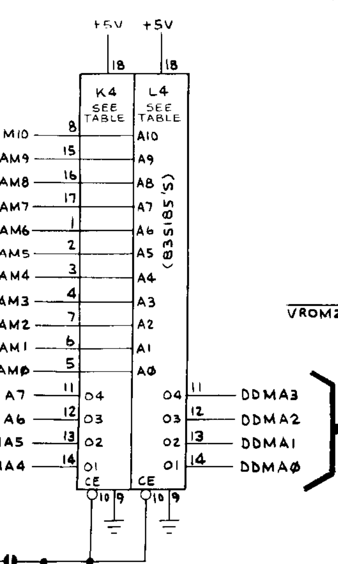
VECTOR GENERATOR DATA BUFFER



* USE PIN NUMBERS IN PAREN-
THESES IF BOARD CONTAINS
AM8304B INSTEAD OF 74LS245

R GENERATOR ROM



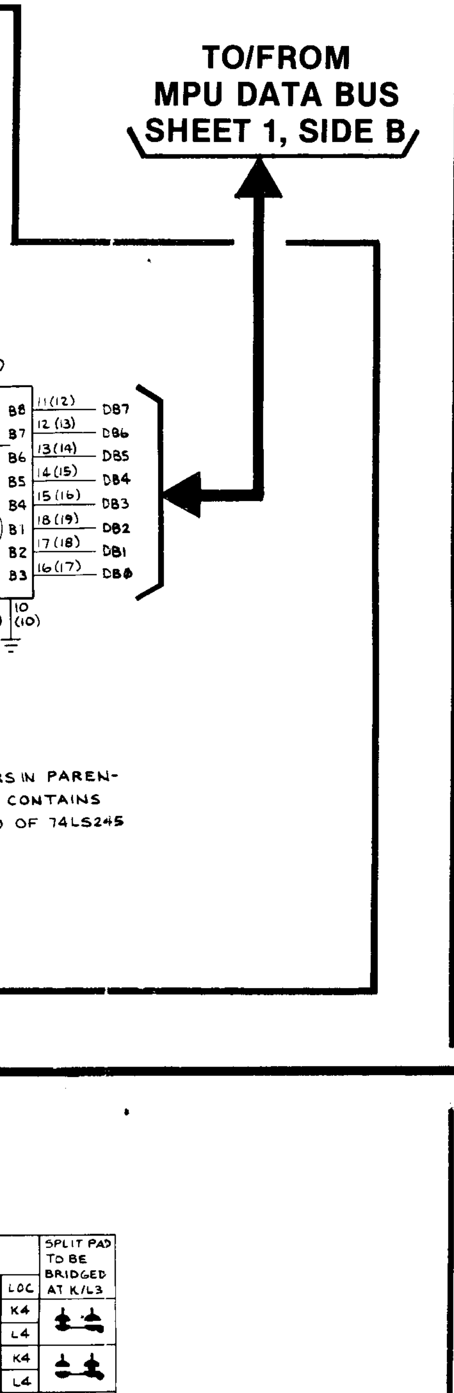


VECTOR MEMORY ROM/PROM SUBSTITUTION				SPLIT PAD TO BE BRIDGED AT K/L3
ROM #	LOC	PROM #	LOC	
035127	N/P3	035129	K4	
		035130	L4	
	R3		K4	
			L4	

The vector generator memory consists o
of RAM and 4K of ROM. It may be directl
cessed by the MPU of the microcomp
through the direct memory access pro
(DMA). Data is written in from the microc
puter thru data buffer R2 when BUFFEN
R/WB are low.

The 2k×8 vector generator prog
memory chip N/P3 may be substituted
two equivalent 1K×8 chips in location K4
L4.

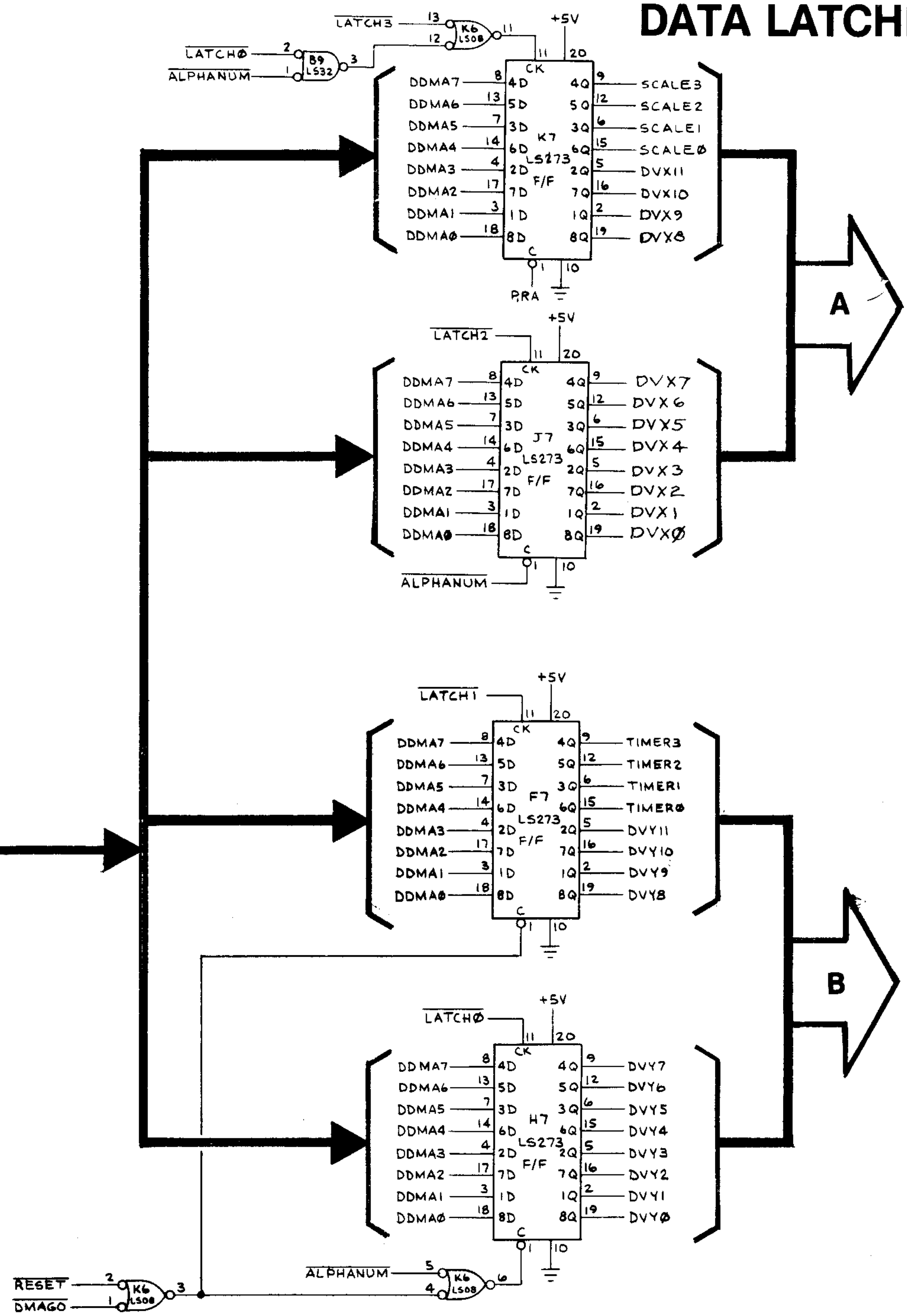
denotes change by indicated rev



y consists of 2K
y be directly ac-
microcomputer
access process
the microcom-
n $\overline{\text{BUFFEN}}$ and

rator program
ubstituted with
location K4 and

VECTOR GENERATOR MEMORY DATA LATCHES



The data latches consist of latch 0 (H7), latch 1 (F7), latch 2 (J7), and latch 3 (K7). Inputs DDMA0 thru DDMA7 are the data outputs from the vector generator memory.

Latches 0 thru 2 are directly clocked by the rising edge of the $\overline{\text{LATCH0}}$, $\overline{\text{LATCH1}}$, and $\overline{\text{LATCH2}}$ outputs from the vector generator's state machine. Latch 3 is clocked by $\overline{\text{LATCH3}}$ or by $\overline{\text{LATCH0}}$, if $\overline{\text{ALPHANUM}}$ is low. Latch 0 is cleared when $\overline{\text{RESET}}$, $\overline{\text{DMAGO}}$, or $\overline{\text{ALPHANUM}}$ goes low. Latch 1 is cleared by $\overline{\text{ALPHANUM}}$.

indicated revision