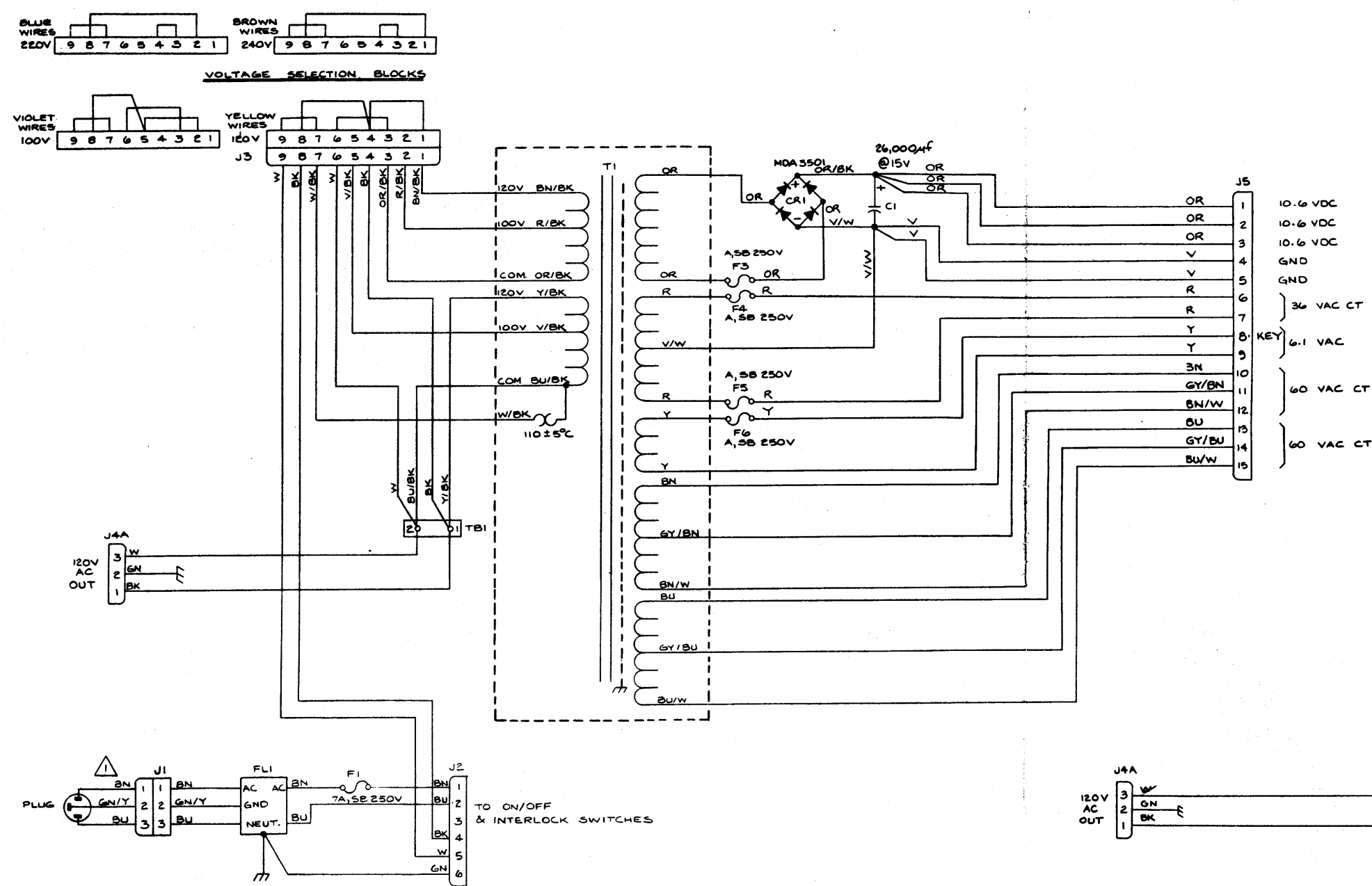
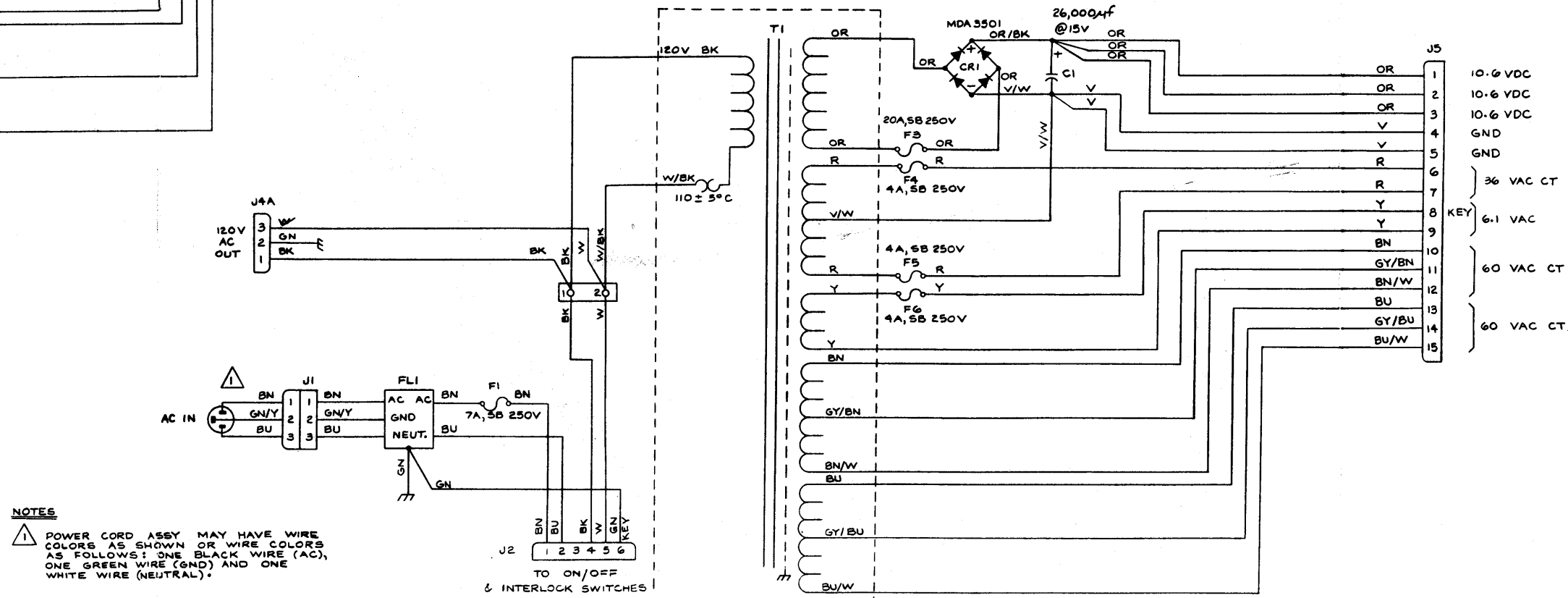


# INTERNATIONAL X-Y POWER SUPPLY WIRING DIAGRAM (035887-01) A)



# U.S. X-Y POWER SUPPLY WIRING DIAGRAM (036352-01) A)



# REGULATOR/AUDIO I PCB SCHEMATIC (034485-03) A)

## Regulator/Audio I PCB

The Regulator/Audio I PCB has the dual functions of regulating the +5 VDC logic power to the game PCB and amplifying the audio from the game PCB.

## Regulator Circuit

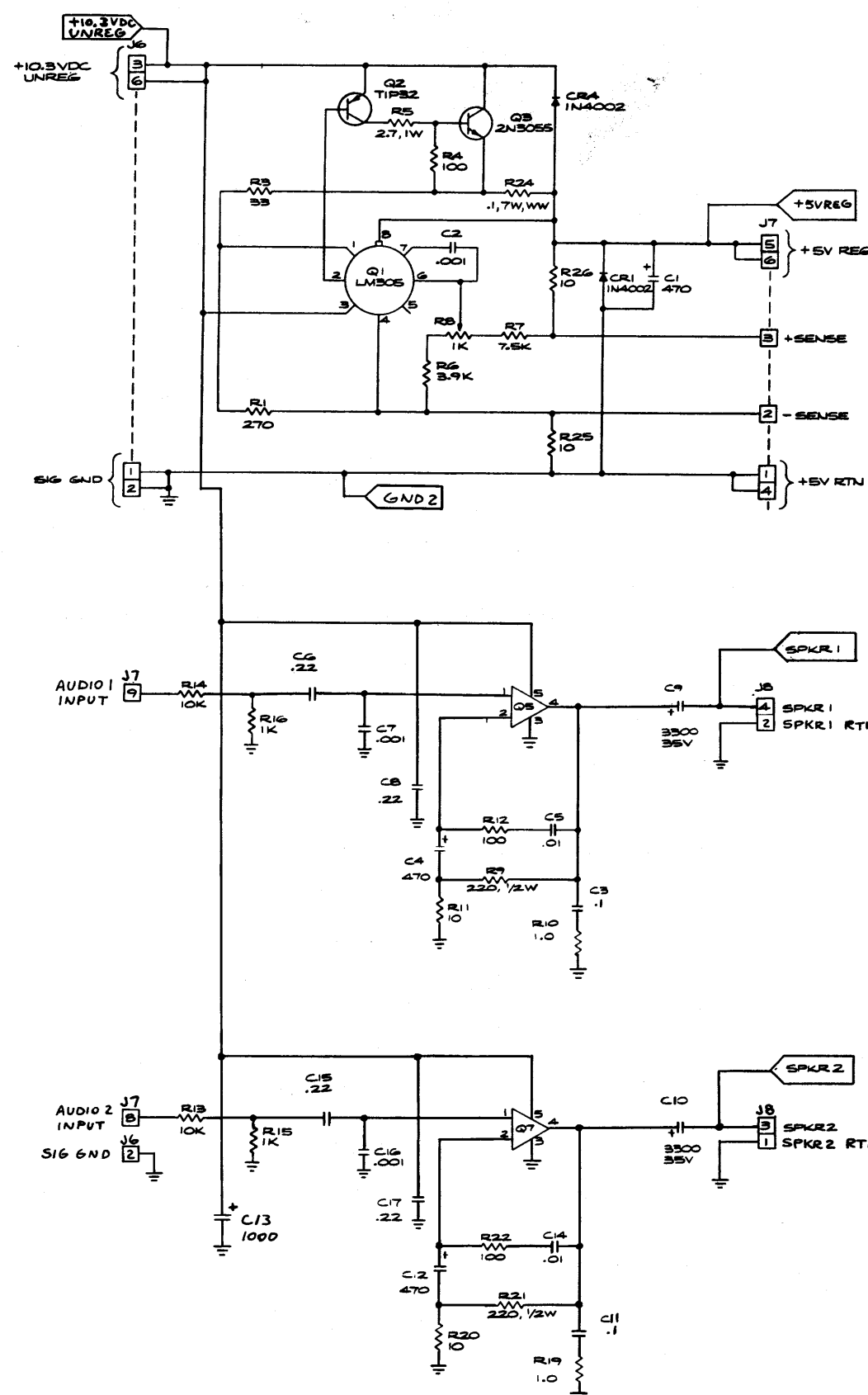
The regulator consists of voltage regulator Q1, current source power transistor Q3 and Q3's bias transistor Q2. The regulator accurately regulates the logic power input to the game PCB by monitoring the voltage through high-impedance inputs +SENSE and -SENSE. The inputs are directly from the +5 VDC and ground inputs to the game PCB. Therefore, the regulator regulates the voltage on the game PCB. This eliminates a reduced voltage due to IR buildup on the wire harness between the regulator and the game PCB. Variable resistor R8 is adjusted for the +5 VDC on the game PCB. Once adjusted, the voltage at the input of the game PCB will remain constant at this voltage.

## Regulator Adjustment

1. Connect a voltmeter between +5 V and GND test points of the game PCB.
2. Adjust variable resistor R8 on the Regulator/Audio I PCB for +5 VDC reading on the voltmeter.
3. Connect a voltmeter between +5 V REG and GND on the Regulator/Audio I PCB. Voltage reading must not be greater than +5.5 VDC. If greater, try cleaning edge connectors on both the game PCB and the Regulator/Audio I PCB.
4. If cleaning PCB edge connectors doesn't decrease voltage difference, connect minus lead of voltmeter to GND test point of Regulator/Audio I PCB and plus lead to GND test point of game PCB. Note the voltage. Now connect minus lead of voltmeter to +5 REG test point on Regulator/Audio I PCB and plus lead to +5 V test point on game PCB. From this you can see which harness circuit is dropping the voltage. Troubleshoot the appropriate harness wire or harness connector.

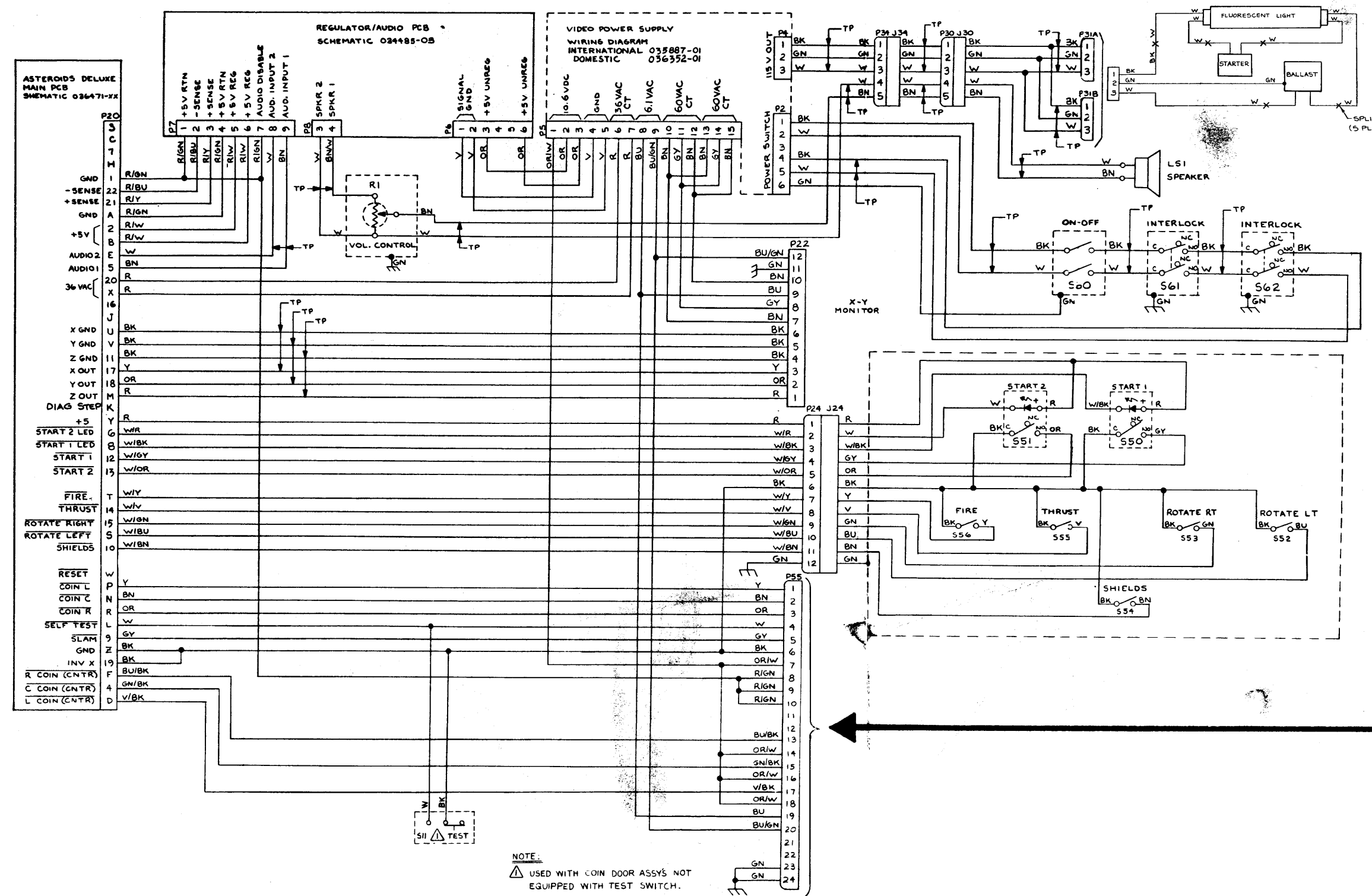
## Audio Circuit

The audio circuit contains two independent audio amplifiers. Each amplifier consists of a TDA2002AV amplifier with a gain of ten.

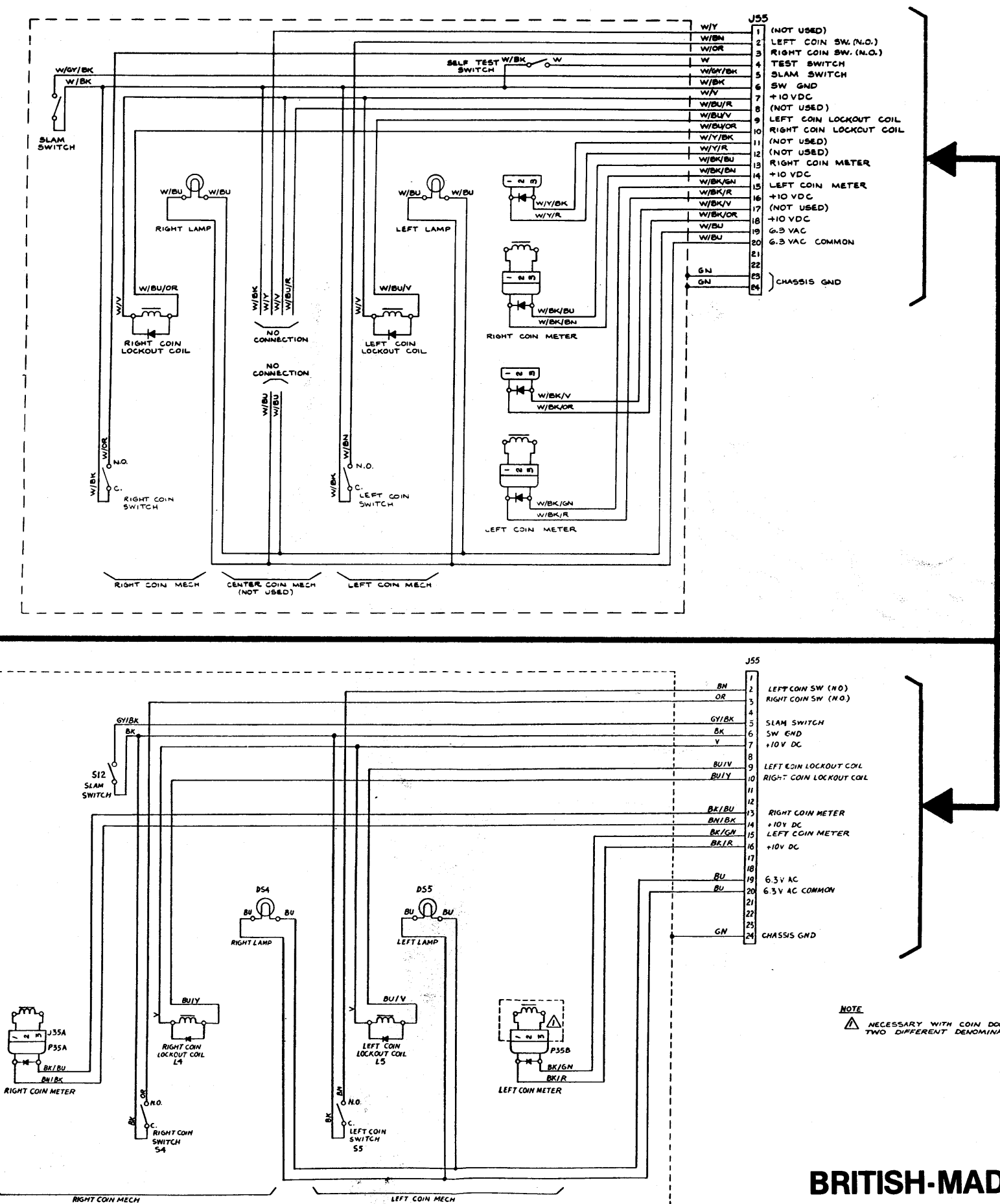


Denotes a test point

# ASTEROIDS DELUXE™ WIRING DIAGRAM (036687-01) A)



# U.S. COIN DOOR SCHEMATIC (034988-01)



# BRITISH-MADE COIN DOOR SCHEMATIC (037050-01) A)

## Drawing Package Supplement

to

## ASTEROIDS DELUXE™

## Operation, Maintenance and Service Manual

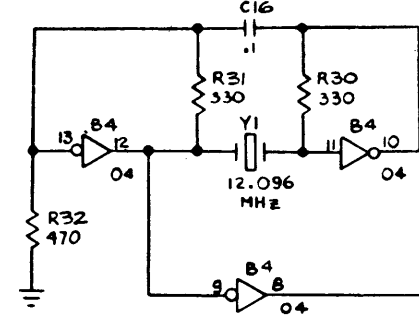
## Contents of this Drawing Package

Game Wiring Diagram, Coin Door and Power Supply Sheet 1, Side A  
Microprocessor Sheet 1, Side B  
Video Generator Sheet 2, Side A  
Switch Inputs, Coin Counter, LED and Audio Outputs Sheet 2, Side B



© 1981 Atari, Inc.

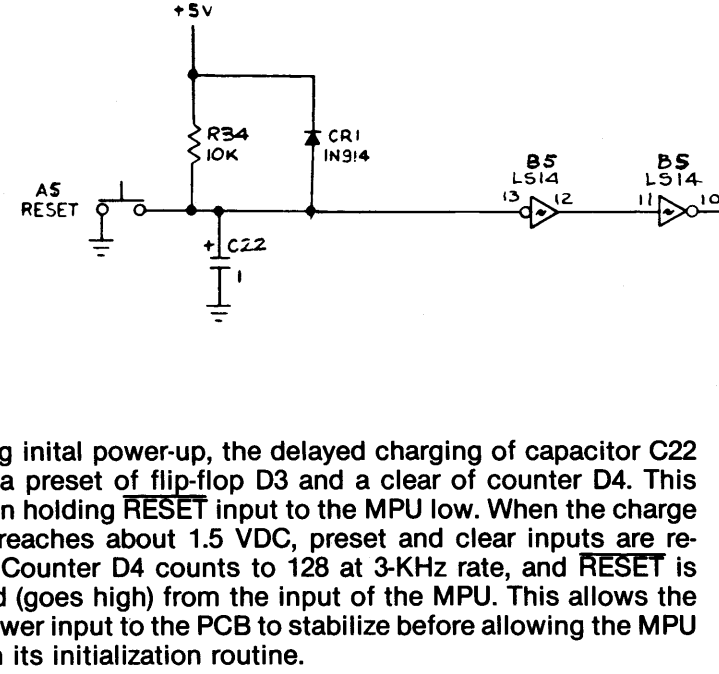
## CLOCK CIRCUIT



The clock circuit consists of crystal Y1 and associated inverters and counters C3 and B3. Counters C3 and B3 count the crystal frequency down to the frequencies necessary for the Asteroids Deluxe™ game.

NOTE:  
The MPU in this game operates at a frequency of 1.5 MHz. Therefore the MPU chip must be the 6502A. The 6502's maximum frequency is 1 MHz and is not compatible with this game.

## POWER RESET AND WATCHDOG COUNTER



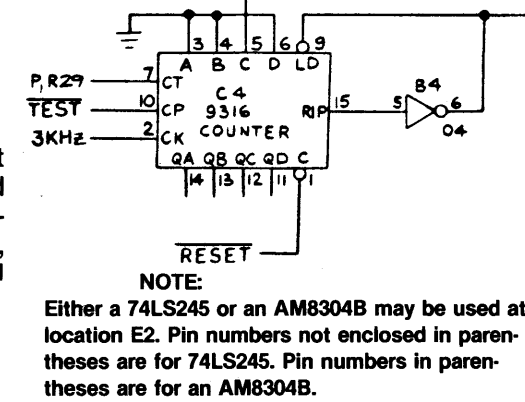
During initial power-up, the delayed charging of capacitor C22 causes a preset of flip-flop D3 and a clear of counter D4. This results in holding RESET input to the MPU low. When the charge of C22 reaches about 1.5 VDC, preset and clear inputs are removed. Counter D4 counts to 128 at 3-KHz rate, and RESET is removed (goes high) from the input of the MPU. This allows the logic power input to the PCB to stabilize before allowing the MPU to begin its initialization routine.

If the MPU program is operating properly, the MPU address decoding circuitry will output the WDCLR (Watchdog clear) signal at predetermined intervals. This serves to clear counter D4 before it counts up to the state that will create the RESET condition. If the MPU program strays from its intended sequence and does not output the WDCLR signal, counter D4 will count up to the RESET state and cause the MPU to return to its initialization routine.

Denotes a test point

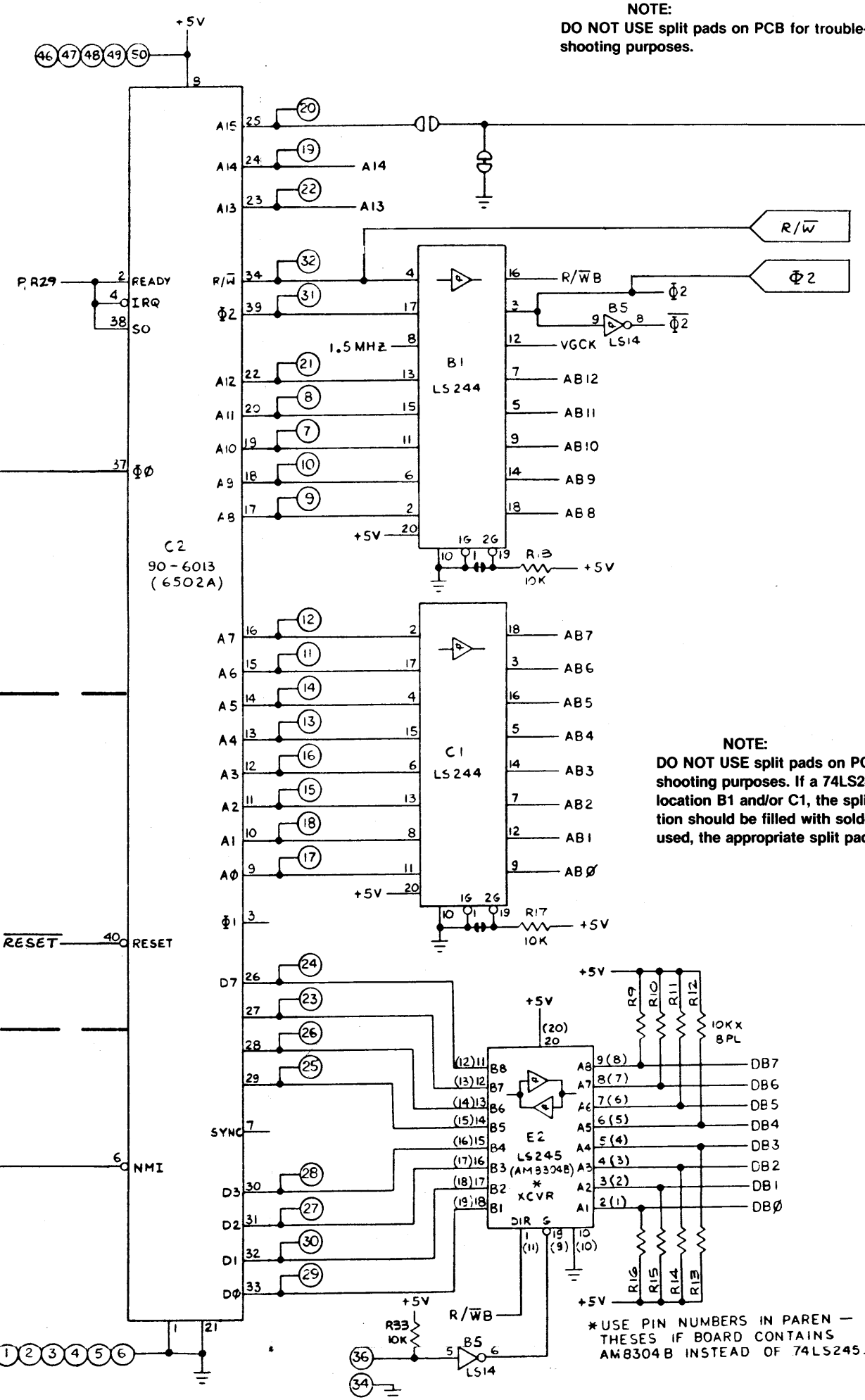
## NMI COUNTER

The NMI (non-maskable interrupt) counter causes an interrupt at the NMI input of the MPU every 4 msec. The interrupt is derived by dividing 3 KHz by a factor of 12 through counter C4. The interrupt occurs when pin 6 of inverter B4 goes low. During power-up, the NMI counter is disabled by RESET. During Self-Test, the NMI is disabled by TEST.



NOTE:  
Either a 74LS245 or an AM8304B may be used at location E2. Pin numbers not enclosed in parentheses are for 74LS245. Pin numbers in parentheses are for an AM8304B.

## MPU CIRCUITRY

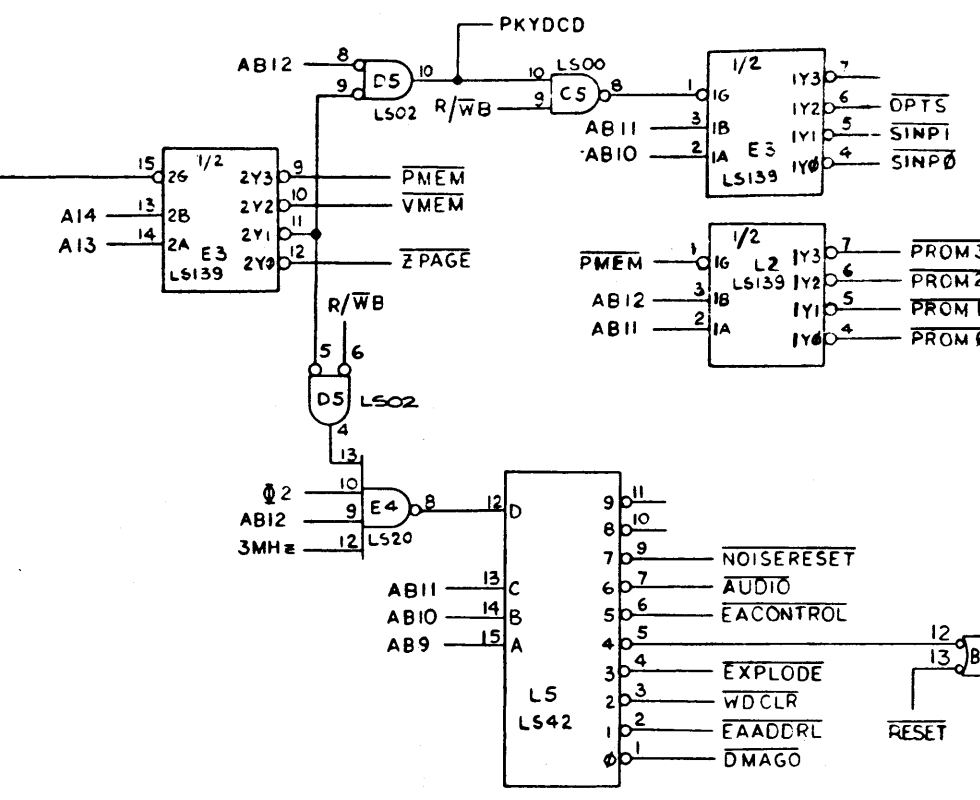


NOTE:  
DO NOT USE split pads on PCB for trouble-shooting purposes.

NOTE:  
DO NOT USE split pads on PCB for trouble-shooting purposes. If a 74LS244 is installed at location B1 and/or C1, the split pad for that location should be filled with solder. If a 74LS241 is used, the appropriate split pad should be open.

NOTE:  
USE PIN NUMBERS IN PAREN — THESE IF BOARD CONTAINS AM8304B INSTEAD OF 74LS245.

## ADDRESS DECODING CIRCUITRY



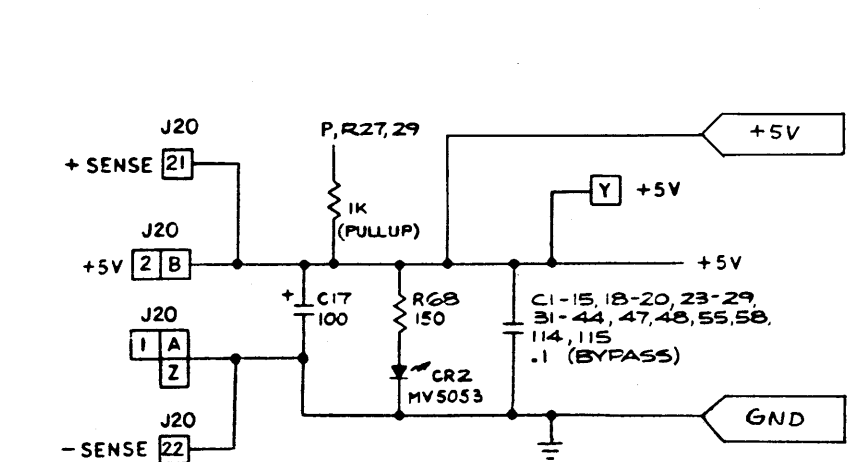
The address decoder performs the function of turning on or enabling the appropriate circuitry at the critical time, so that information can be transferred back and forth between the game circuitry and the MPU. The memory map below is for the Asteroids Deluxe™ game.

If you are going to use the Automatic RAM/ROM Tester, please remember to remove MPU C3 and ground the WDOG DISABLE test point.

## MEMORY MAP

HEXADECIMAL	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	R/W	D7	D6	D5	D4	D3	D2	D1	D0	FUNCTION
0-3FF	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Scratch RAM
2000	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Not used
2001	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									3 KHz
2002	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									VG halted
2003	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Shield switch
2004	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Fire switch
2006	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Slam switch
2007	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Self-test switch
2400	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Left coin switch
2401	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Center coin switch
2402	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Right coin switch
2403	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									1-player start switch
2404	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									2-player start switch
2405	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Thrust switch
2406	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Rotate left switch
2407	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Rotate right switch
2800	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Option switch 8, 7 (at R8)
2801	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Option switch 8, 5
2802	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Option switch 4, 3
2803	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Option switch 2, 1
2C00	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									POKEY
2C40	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									EAROM read
3000	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Start VG
3001	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Latch EA address/data
3002	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Watchdog reset
3003	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Explosion pitch
3004	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Explosion volume
3005	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									3800 VG reset
3006	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									EA control latch
3007	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									1-player start LED
3008	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									2-player start LED
3009	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Not used
300A	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Not used
300B	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Ship thrust sound
300C	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Bank select
300D	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Left coin counter
300E	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Center coin counter
300F	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Right coin counter
3E00	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Noise generator reset
4000	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Vector RAM
4001	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Vector RAM
4002	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Vector RAM
4003	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Vector RAM
4004	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Program ROM
4005	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Program ROM
4006	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Program ROM
4007	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Program ROM
4008	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Program ROM
4009	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Program ROM
400A	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Program ROM
400B	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Program ROM
400C	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Program ROM
400D	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Program ROM
400E	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Program ROM
400F	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R									Program ROM

## POWER INPUT



This circuitry consists of the PCB inputs and outputs for the +5 VDC logic power and 25 VAC input to the on board regulators. The +5 VDC inputs and outputs are discussed on Sheet 1, Side A of this schematic set.

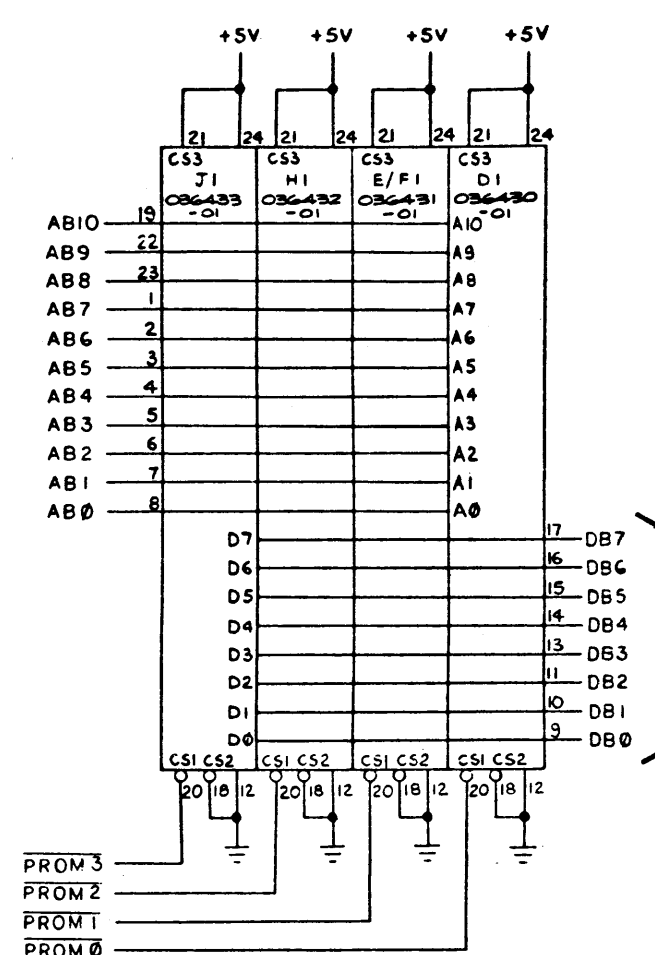
The 25 VAC inputs are received by two full wave rectifiers. Diodes CR9 and CR8 rectify the negative cycle of the input and the 7915 regulates the voltage at -15 VDC. Diodes CR6 and CR10 rectify the positive pulse of the 25 VAC input and the 7815 regulates the voltage at +15 VDC. The 7805 regulates an additional 5 VDC for the DACs. Zener diode CR12 supplies the +8.2 VDC for the sample and hold circuit. The +22V (unregulated) is used to power operational amplifiers P11 and L8 in the audio output.

Denotes a test point

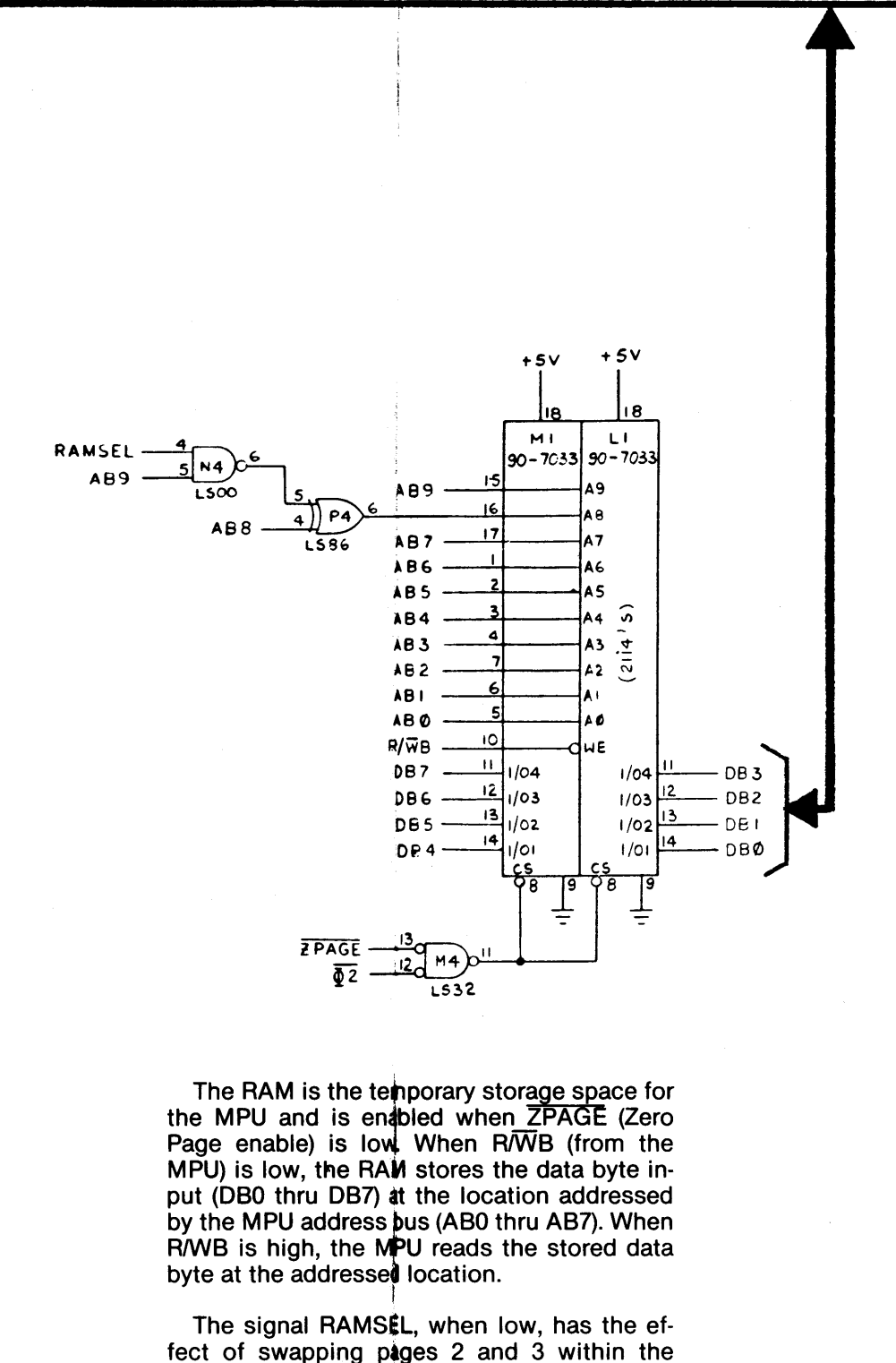
## ROM/PROM CIRCUITRY

FROM SWITCH INPUTS  
SHEET 2, SIDE B

Program memory for the Asteroids Deluxe™ game is contained in three ROMs.



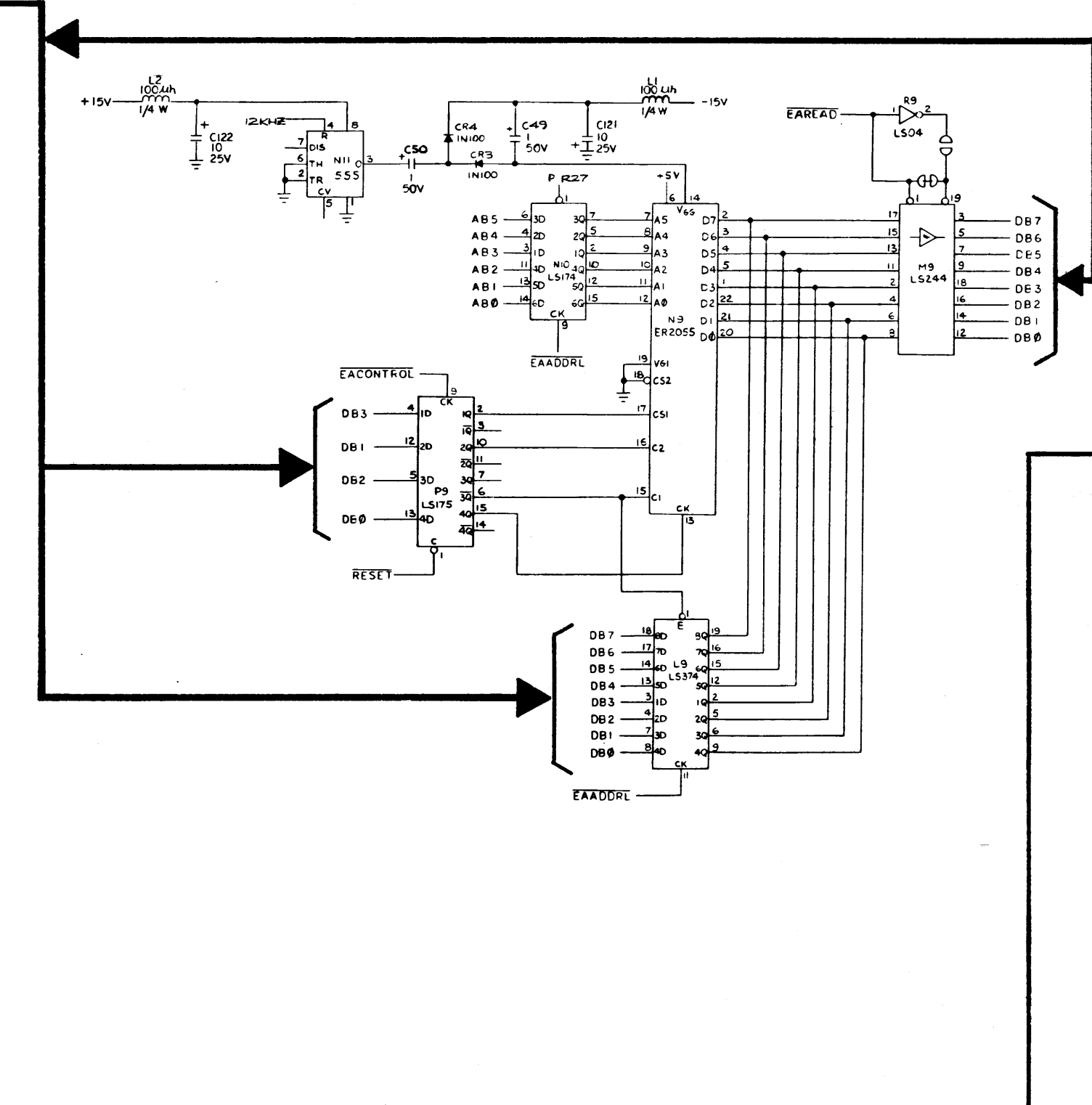
## RAM CIRCUITRY



The RAM is the temporary storage space for the MPU and is enabled when ZPAGE (Zero Page enable) is low. When R/WB (from the MPU) is low, the RAM stores the data byte input (DB0 thru DB7) at the location addressed by the MPU address bus (AB0 thru AB7). When R/WB is high, the MPU reads the stored data byte at the addressed location.

The signal RAMSEL, when low, has the effect of swapping pages 2 and 3 within the RAM. This allows greater programming flexibility.

## HIGH SCORE CIRCUITRY



The High Score Memory circuit consists of an erasable re-programmable ROM N9, latches L9, P9, N10 buffer M9, and timer N11.

N11 produces a 12KHz O-15V squarewave. This signal when +15, forward biases diode CR4 and allows capacitor C50 to charge to -29V. When it's OV, CR4 is then cut-off and CR3 is forward biased which causes C49 to develop a charge. C49 charges to approximately -28V. This is the potential required for EAROM N9 to operate.

The MPU addresses the EAROM (AB0-AB5) via latch N10, when EXADDR1 goes high, and data is latched into the EAROM on DB0-DB7 through latch L9.

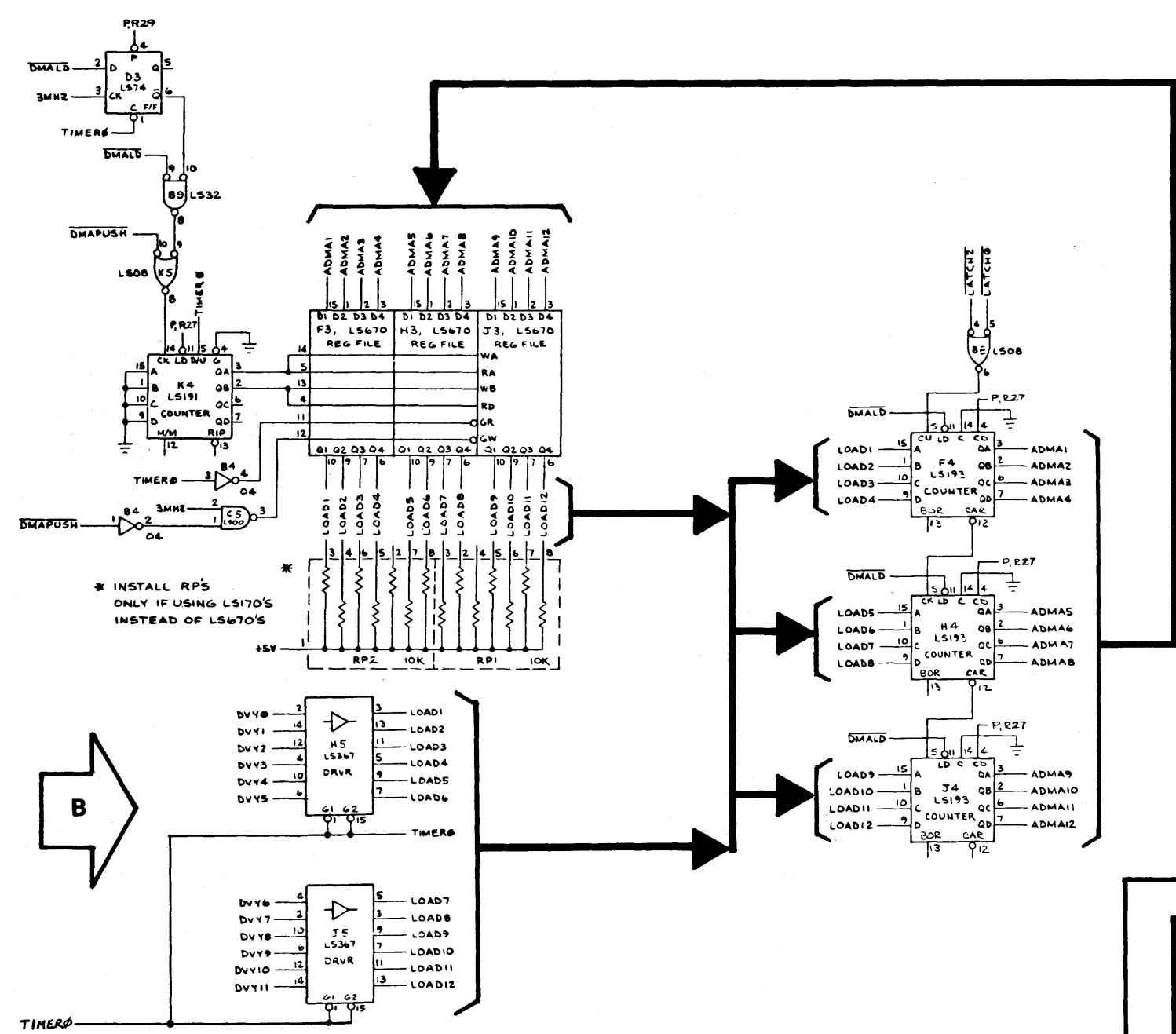
The function of the EAROM (read, write or erase) is determined by the MPU on data lines DB0-DB3. Latch D9 receives a high EACONTROL signal from the MPU address decoder and function data is passed to the EAROM.

Data in the EAROM is read by the MPU when the EXAREAD is addressed by the MPU after a reset pulse or during self-test.

NOTICE TO ALL PERSONS RECEIVING THIS DRAWING  
CONFIDENTIAL: Reproduction forbidden without the specific written permission of Atari, Inc., Sunnyvale, CA. This drawing is only conditionally issued, and neither the drawing nor the information contained herein is to be released, in whole or in part, to any person or organization without the specific written permission of Atari, Inc. Except for manufacture by vendors of Atari, Inc. and for manufacture under the corporation's written license, no right to reproduce this drawing is granted or the subject matter thereof unless by written agreement with or written permission from the corporation.



## PROGRAM COUNTER

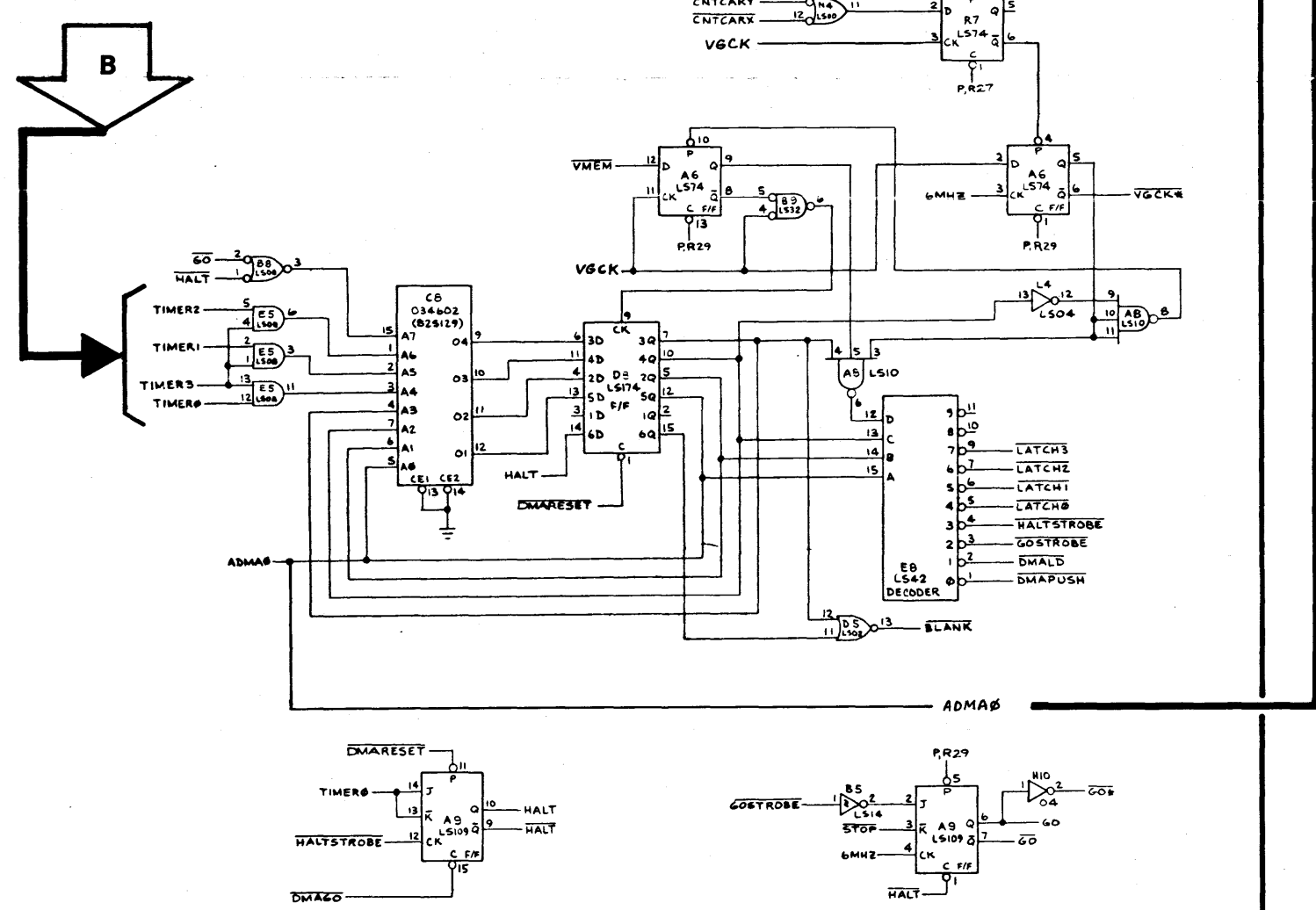


Counters F4, H4 and J4 contain the address of the next data byte (instruction) to be fetched from the Vector Generator memory. Because these counters point to the next instruction in memory to be retrieved and performed, they are called the program counter. This program counter is incremented one count 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 F6 and H6 and buffers H5 and J5.

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 F3, H3, J3, and down/up counter K4. 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 DMARST is low. Immediately after information is written into the stack, counter K4 increments one count. Immediately before loading the program counter from the stack, counter K5 decrements one count.

## 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 U-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 E5, ROM C8, latch D8, clock circuitry A6, 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 GO tells the ROM that the position counters are presently drawing a vector. The HALT input to A7 tells the ROM that the vector generator has completed its operations.

During initial power-up of the game, the HALT signal is preset low. The microcomputer reads the high HALT signal through its switch input port (sel/mux L10) on data line DB7. 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 BLANK to the Z-axis output.

The microcomputer outputs an address that results in a DMAGO signal that causes 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.

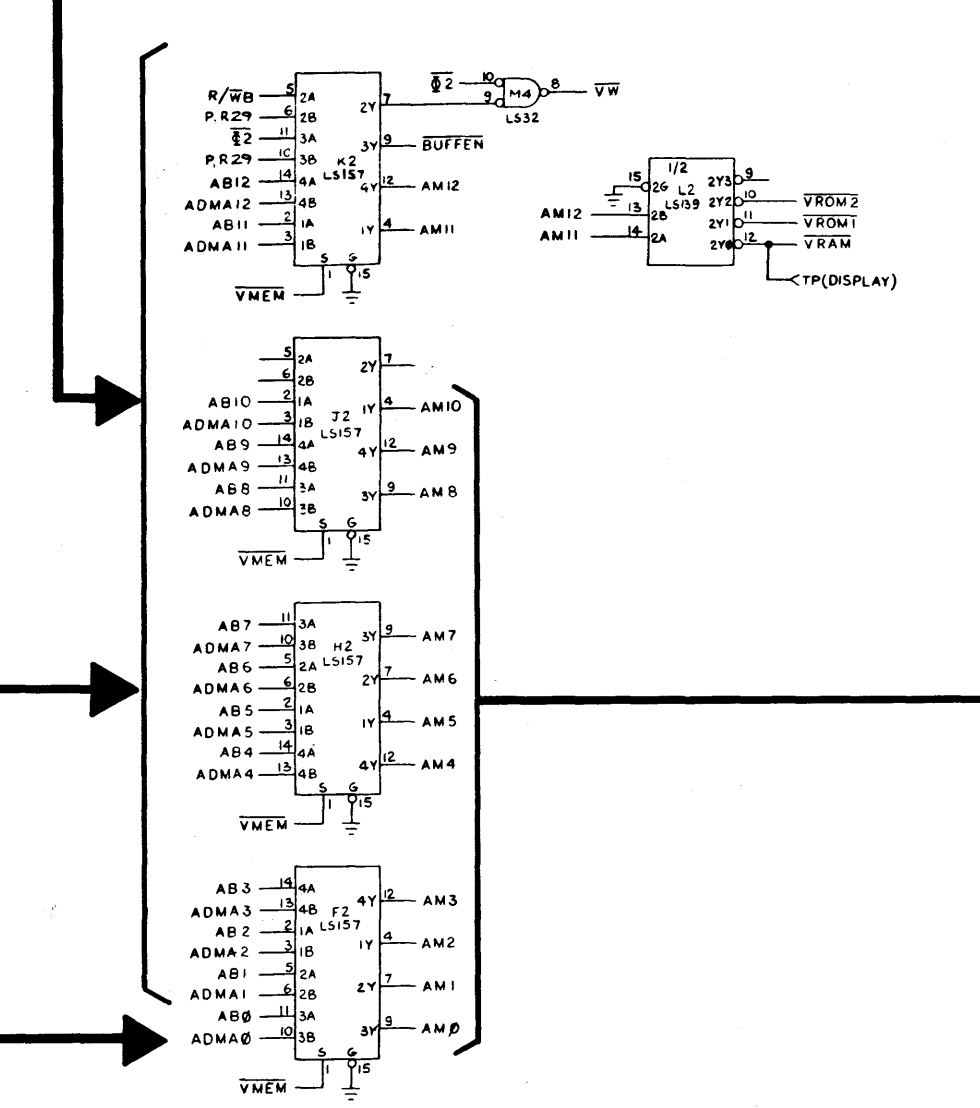
When the state machine receives the operation code for a HALT instruction, it outputs a low HALTSTROBE, setting the HALT flip-flop A9, and suspending state machine operation.

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 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 GOSTROBE is clocked through A9, the vector timer and X- and Y-position counters begin to operate from the GO, GO and GO+ signals. When STOP is clocked through A9, the vector timer and position counters are suspended, and 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 VMEM goes low). Then the clock input to latch D8 goes high and stays high until VMEM goes high.

## FROM MICROCOMPUTER SHEET 1, SIDE B

### VECTOR GENERATOR MEMORY ADDRESS SELECTOR

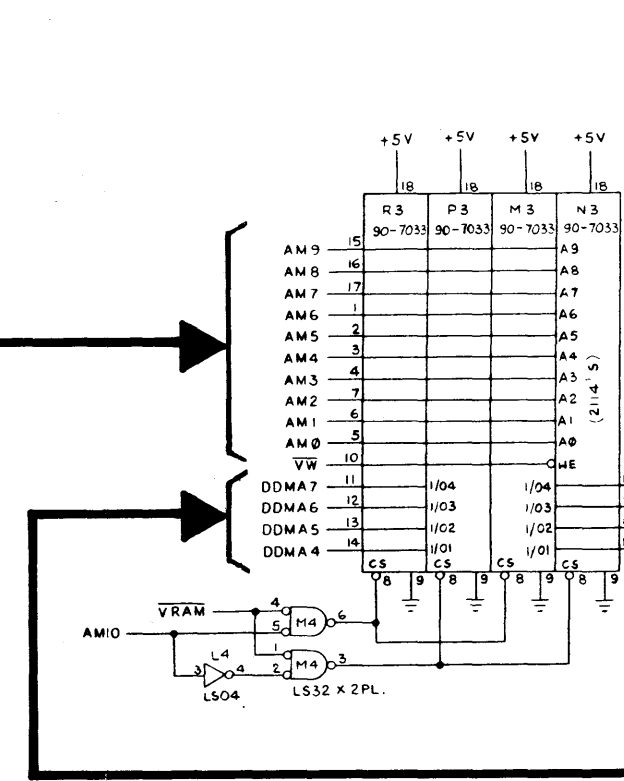


The address selector consists of multiplexers F2, H2, J2 and K2. When VMEM is low, the MPU of the microcomputer gains access to the address inputs of the vector generator memory. In this state, BUFFEN is from Q2 and VW (vector generator write) is low when Q2 and RWB are both low. When VMEM is high, the address input to the vector generator memory is from the vector generator program counter and state machine. In this state, BUFFEN and VW are both held high by the pullup resistors connected to the 2B and 3B inputs of multiplexer K2.

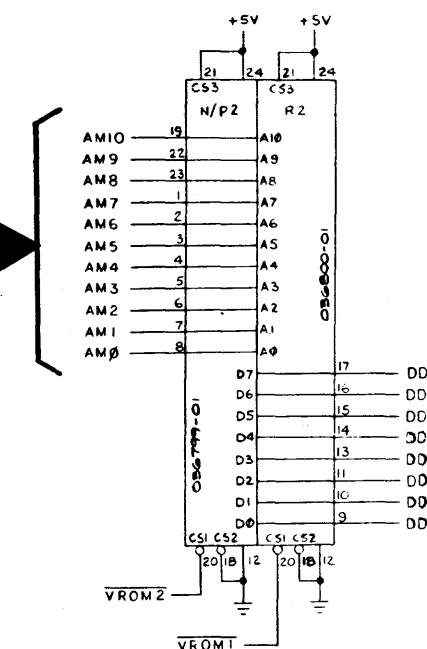
Address decoder L2 decodes address bits A11 and A12, and selects the RAM or one of three ROMs of the vector-generator memory.

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.

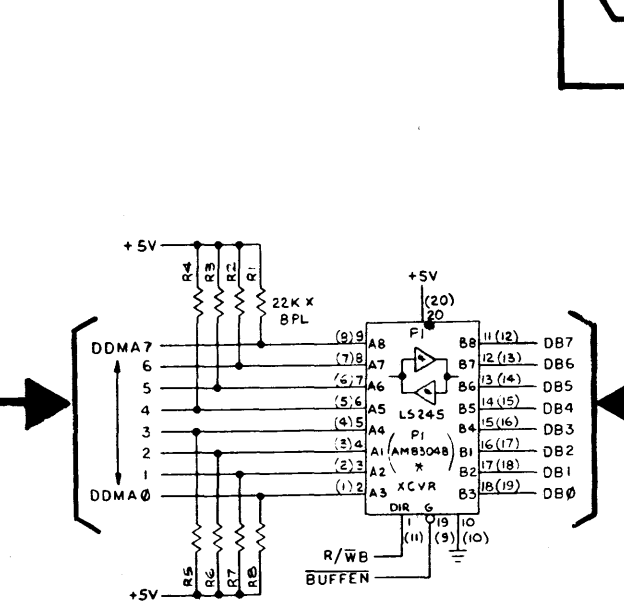
## VECTOR GENERATOR RAM



## VECTOR GENERATOR ROM



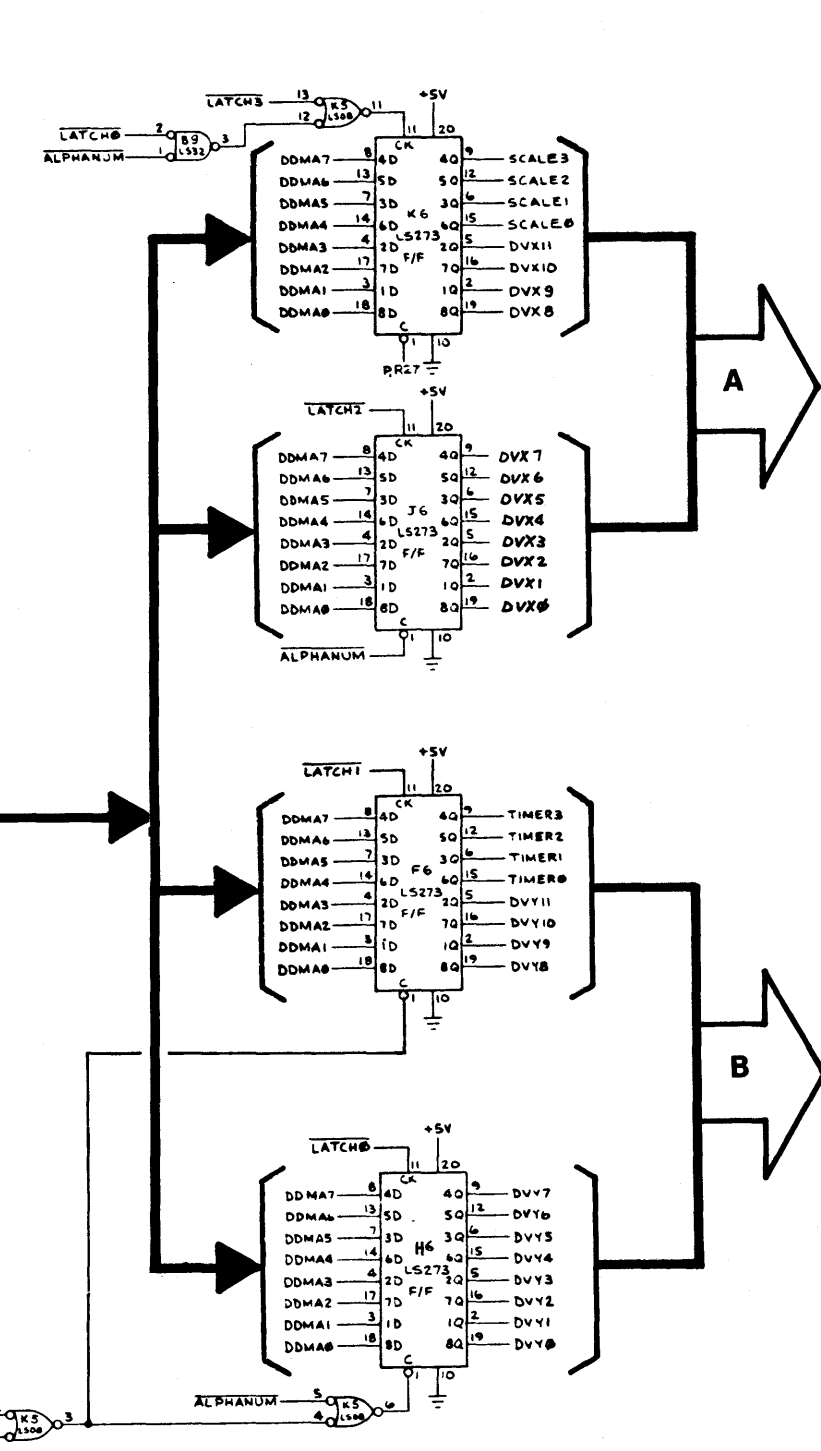
## VECTOR GENERATOR DATA BUFFER



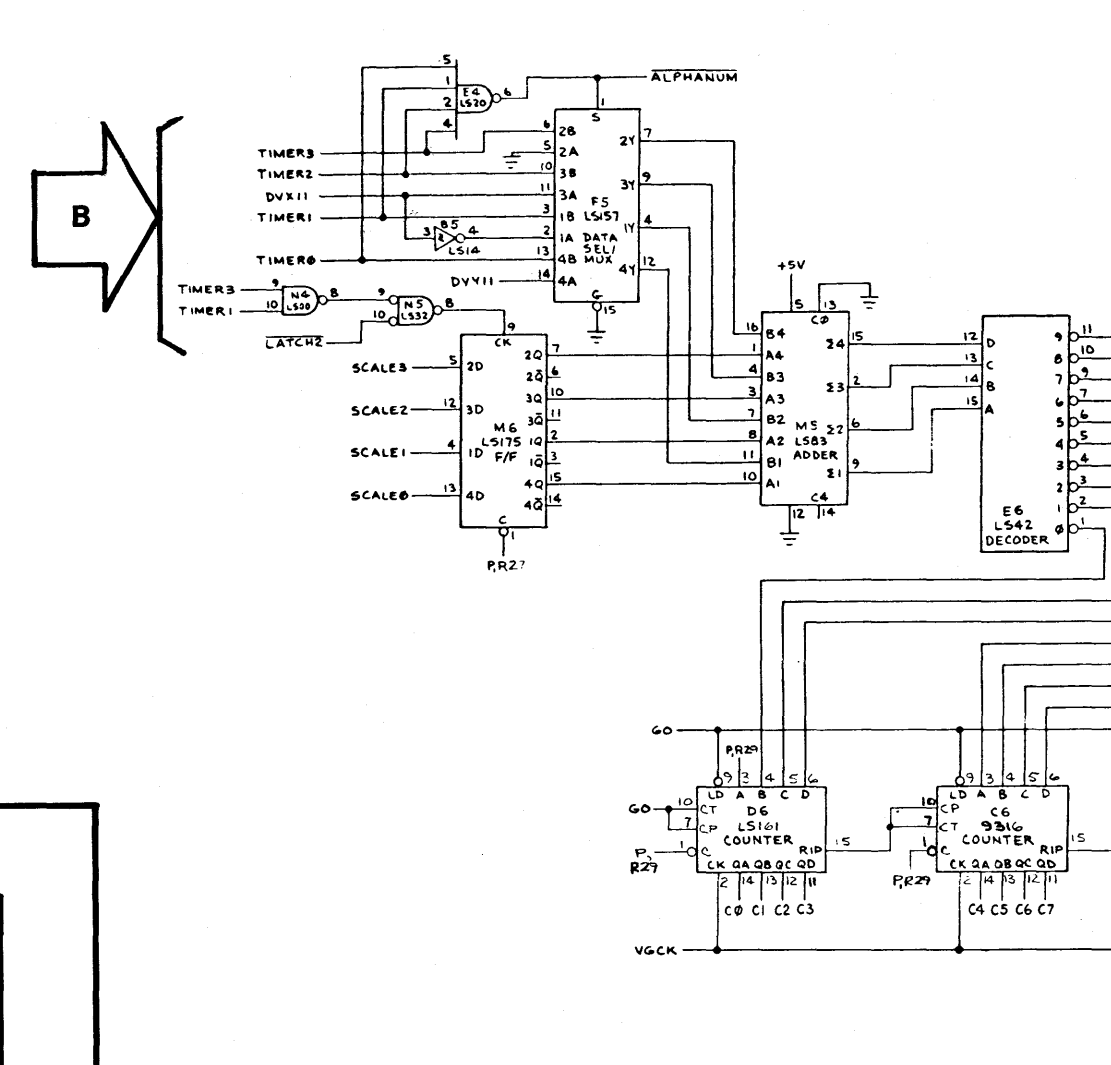
The data latches consist of latch 0 (H6), latch 1 (F6), latch 2 (J6), and latch 3 (K6). Inputs DDMA0 thru DDMA7 are the data outputs from the vector-generator memory.

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

## VECTOR GENERATOR MEMORY DATA LATCHES



## VECTOR TIMER

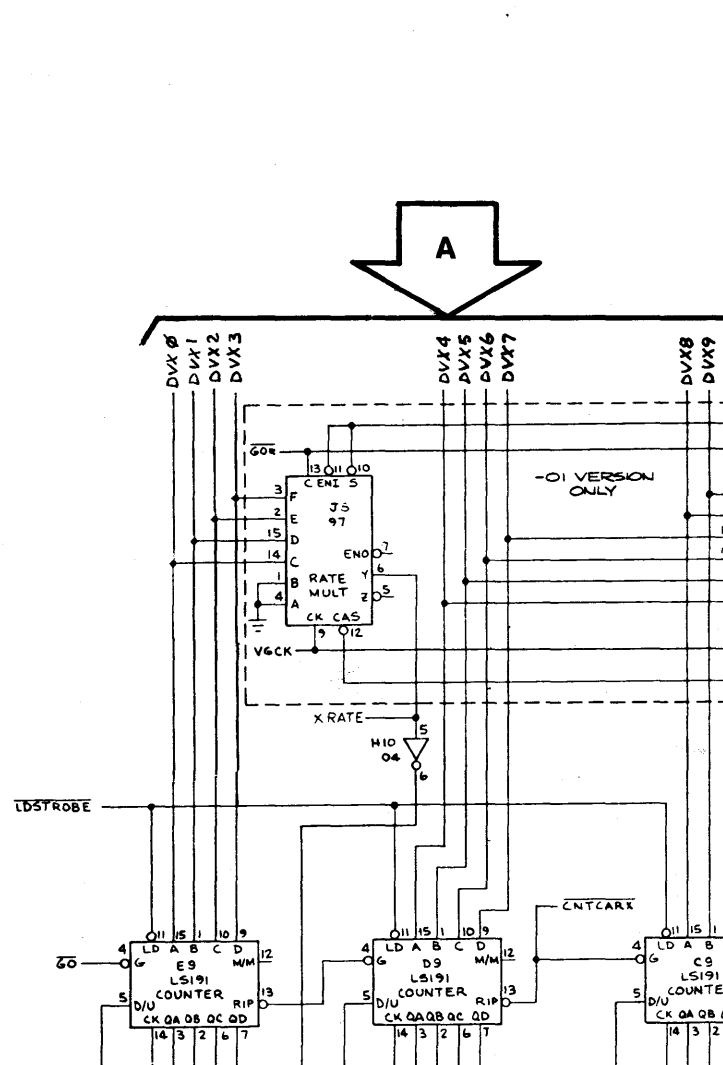


The purpose of the vector timer is to time out the length of time it takes to "draw" an actual vector on the monitor display. During the interval when the X- and Y-position counters are actually drawing the vector, STOP is high. This prevents the vector-generator state machine from advancing to its next 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 F5, decoder E6, latch M6, adder M5, and counters B6, C6, and D6. M6 contains a scale factor which is added in M5 to the four timer signals. If TIMER0 thru TIMER3 inputs are any state but all high, decoder E6 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 DVI11 are decoded by decoder E6. This is added to the scale factor and loaded into the counters.

## X- AND Y-POSITION COUNTERS



The X- and Y-position counters are two identical circuits. Therefore, the following description discusses only the Y-position counters. The X-position counters contain rate multipliers (J8 and K8), down/up counters (C9, D9 and E9), multiplexers (C10, D10, E10), latch (F10), and associated gates (B8 and H10). The output of the down/up counters is a 12-bit binary number that represents the horizontal location of the beam on the monitor screen (or X axis), with 0 being the far left side of the screen and 1023 being the far right side of the screen. Increasing or decreasing this binary number output will cause the beam to move to the right or left, respectively. The vector generator state machine decodes instructions from its memory, and then is capable of using that data to alter the binary count of these counters in one of two ways.

The state machine can preset these counters to an entirely different number from their previous contents. This will cause the beam to "jump" to a new location on the monitor screen instantaneously, i.e., for drawing a new vector from a different starting position than where the previous 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 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 counters count up or down. DVX11 determines the quadrant of the vector being drawn.

The UNMDACX1 thru UNMDACX10 (X-axis unmultiplexed digital-to-analog converter signals) are transferred and stored at the output of the multiplexers on each rising edge of the 6-MHz clock (from the microcomputer clock circuitry). The DACX1 thru DACX10 signals are sent to the digital-to-analog converters (DACs) in the X video output. The DACX1 and DACX10 outputs represent the physical placement of the beam on the monitor. The far left of the monitor screen is 0, the center is 512, and the far right is 1023. Therefore, if the DACX1 thru DACX10 signal was greater than 1023, the monitor beam would go off the right side of the screen and start again on the left side of the screen, a "wraparound" condition. To prevent a wraparound, the multiplexers' select input from UNMDACX11 goes high when the count is greater than 1023 or less than 0. This selects UNMDACX12 to be output from the multiplexers to the DACs, forcing all zeros or all ones, and thus keeping the beam on the appropriate side on the screen, instead of allowing it to wraparound.

The XLVD and YLVD (X and Y valid) outputs from the X- and Y-position counter multiplexers are latched (F10) and gated together to enable the Z axis output, BVLD (beam valid).

Sheet 2, Side A  
ASTEROIDS DELUXE™  
Video Generator  
Section of 036471-02 C





