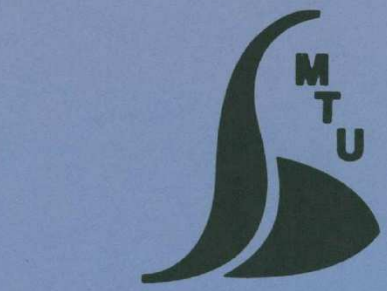


Micro
Technology
Unlimited

K-1008 VISIBLE MEMORY

8K BYTE MEMORY
200 HIGH BY 320 WIDE DOT MATRIX DISPLAY
FOR 6502 SYSTEMS

JANUARY 1, 1979

COPYRIGHT NOTICE

Micro Technology Unlimited
PO Box 4596
841 Galaxy Way
Manchester, NH 03108

K-1008 UNPACKING AND INSTALLATION

The K-1008 Visable Memory is a carefully engineered, manufactured, and tested product that should operate perfectly when handled and installed according to the following instructions. Note that the board is shipped in a black conductive plastic bag. Since MOS integrated circuits are used, damage from static discharge is possible. It is helpful to reduce static by working in an area with concrete floors and a reasonable humidity level. If this is impossible, at least avoid wearing rubber soled shoes and move slowly in the work area. When unpacking or handling the board, touch the screw sticking up in the middle of the heatsink first and release it last. Note that the preceeding comments apply equally to the KIM board which of course contains MOS IC's also.

Jumper socket S1 is shipped with jumpers installed for board addressing between 2000 and 3FFF and the full screen enabled. If at all possible, the board should be tested in the user's system with these jumpers intact. Following testing, they may be reconfigured as desired according to the table below:

ADDRESS RANGE	INSTALL JUMPERS BETWEEN S1		
2000-3FFF	1-16	3-14	6-11
4000-5FFF	1-16	4-13	5-12
6000-7FFF	1-16	4-13	6-11
8000-9FFF	2-15	3-14	5-12
A000-BFFF	2-15	3-14	6-11
C000-DFFF	2-15	4-13	5-12

To blank first 4K of the screen (lines 0-101 and part of 102) install a jumper between S1-7 and S1-10. To blank the second 4K (part of line 102 and lines 103-199) install a jumper between S1-8 and S1-9. Never install both jumpers.

If desired, the user may install DIP headers wired with the jumpers or a standard 8 pole dipswitch into S1.

Connection to the KIM-1 should be as indicated in the accompanying chart. The easiest method of connection to the KIM is with an MTU model K-1005 motherboard/cardfile. Alternatively the user may obtain two 2x22 pin printed circuit board edge connectors (.156" contact spacing) such as the one supplied with the KIM and wire them together except for contact X. Wire length should not exceed 4 inches. Plug the KIM expansion connector into one of the sockets, make the indicated connections to the application connector, and make the indicated power connections. The visable memory may then be plugged into the other connector.

Note that as shipped the board requires an unregulated voltage between +7 and +12 volts to operate the logic and another unregulated voltage between +14 and +20 volts to operate the memory chips such as provided by the expansion outputs of an MTU K-1000 power supply. The on-board regulators may be bypassed by shorting the two outside pins of each regulator IC together if the user wishes to use a regulated power source.

The video cable to the monitor should be high quality 75 ohm coax if the length exceeds 5 feet. A standard RCA phono plug is required at the VM end of the cable. For only one monitor along the cable, impedance matching at the monitor is not required. For maximum utilization of the high resolution capabilities of the Visable Memory, a video monitor or converted television is recommended. If a converted TV is used make sure that negative-going sync is expected and make doubly sure that the TV chasis is not hot!

With some monitors minor adjustment of the horizontal hold control will be necessary to obtain synchronization and to center the image horizontally. The video input level may need to be adjusted when using certain surplus computer terminal monitors. This may be accomplished with a 250 ohm pot accross the monitor video input or trial and error substitution of carbon resistors in the 50 to 250 ohm range. Excessive "swimming" of the image is either due to an external AC magnetic field such as from a computer power supply or is the fault of the monitor itself. The latter situation may be improved considerably by increasing the monitor's internal power supply filter capacitors.

After connecting the KIM, the monitor, and the power supply, the system may be turned on. The monitor should show a stable, semi-random pattern of memory contents. Adjust the horizontal hold, vertical hold, brightness, and contrast controls until a clear, stable and centered image is obtained. All corners of the image should be visable. If not adjust the monitor's height and width controls.

Pressing RESET on the KIM should initiate normal KIM operation. Set the address to 2000 and store different values there. The bit pattern in binary should show up in the upper left corner of the screen. The KIM data display should be stable and reflect the data stored. Go to 2001 and repeat.

If all is well at this point the test program supplied with the Visable Memory should be loaded through the KIM keyboard and dumped to cassette tape. The entry point is 0200 and the program should start by showing a series of different checkerboards. After 16 checkerboards are displayed, random bit patterns are generated and checked. After 16 of these the cycle repeats but with different patterns. The program should run indefinitely without stopping. If it does stop, locations 0000 and 0001 indicate the address of the failure and address 0002 shows the bit or bits in error. The checkerboard pattern is ideal for adjusting vertical linearity of the monitor also.

At this point checkout of the Visable Memory is complete and the user may now begin to write programs for it.

APPLICATION OF MULTIPLE K-1008 BOARDS

Besides use as a display board, the K-1008 outperforms the KIM manufacturer's 8K memory board in terms of power consumption and availability. It also does not require any external logic to connect directly to the KIM. When using multiple Visable Memories, it is advisable to remove U1, which is socketed, from all of the boards except one. This reduces address bus loading. The KIM bus is rated to drive three K-1008's and typically can easily drive four. The K-1000 power supply is rated to drive two K-1008's along with the KIM but can typically drive four of them also. In fact, the boards are tested four-at-a-time for 24 hours in this configuration.

Multiple Visable Memories may also be used for gray scale or color applications. Once synchronized, the boards will remain in perfect synchronization due to the fact that they all are synchronized to the same crystal controlled clock. Initial synchronization may be performed by force resetting the counter chains on all boards at power up. An application note detailing gray scale and color applications will be available shortly.

ADJUSTING THE DOT SYNC POTENTIOMETER

This adjustment was carefully made at the factory with the aid of an oscilloscope and should never require readjustment. However if the KIM display is unstable when examining VM contents or a <u>random</u> shimmy (not steady waver) is seen in the displayed image the pot may have fallen out of adjustment. Rotate the pot until a stable screen image is seen and the KIM data display is stable when examining a VM location. If a multimeter is available, further rotate the pot until a voltage reading at U8 pin 13 of 1.4 volts is achieved. The monitor and KIM displays should remain stable. If a meter is not available, note the extremes of rotation that provide stable displays and set the pot midway between the extremes. A spot of nail polish will serve to prevent future drifting of the adjustment.

SPECIFICATIONS

Display Format: 200 lines, 320 dots per line, non-interlace
Scanning Frequencies: (derived from KIM-1 crystal clock)
Horizontal: 15,625 Hz, Vertical: 60.1 Hz.
Required video bandwidth: 4 mHz minimum
Output: 1.25 V p-p composite video into 75 ohms, sync negative
Adjustments: One, dot sync (factory aligned on assembled units)
Power requirements: +7.5 volts unregulated .25 amp, +16 volts unregulated .25 amp.
Sockets: 16 memory IC's, address and blanking jumpers, and vector fetch gate (7430) are socketed.
Memory type: 22 pin 4K dynamic RAM, National Semi. MM5280 or equ.
Access time: greater than 100NS data stable time prior to fall of $\overline{\text{Phase 2}}$ clock
Cycle time: internally synchronized to 1.0mHz $\overline{\text{Phase 2}}$ clock from host system
Printed circuit board: 11" wide by 5" tall exclusive of gold-plated edge connector, plated-through holes
Inclusions: bare or assembled and tested board; instruction manual containing schematic, trouble-shooting tips, and memory diagnostic (fun to watch!)
Price: Assembled and tested - $289.00
Bare board - $40.00
Kits are not available.
Quantity discounts are available, please request on letterhead a current MTU price list.
Delivery: First retail delivery is January, 1978. Standard delivery schedule is stock to 2 weeks for retail orders. Delivery on larger quantities is individually negotiated.

PIN CONNECTIONS

Signal	KIM	K-1008	Signal	KIM	K-1008
SYNC	E-1	N.C.	ADDR BUS 0	E-A	A
RDY	E-2	N.C.	ADDR BUS 1	E-B	B
PHASE 1	E-3	N.C.	ADDR BUS 2	E-C	C
IRQ	E-4	N.C.	ADDR BUS 3	E-D	D
SET OVERFLOW	E-5	N.C.	ADDR BUS 4	E-E	E
NON-MASK INT.	E-6	N.C.	ADDR BUS 5	E-F	F
RESET	E-7	N.C.	ADDR BUS 6	E-H	H
DATA BUS 7	E-8	8	ADDR BUS 7	E-J	J
DATA BUS 6	E-9	9	ADDR BUS 8	E-K	K
DATA BUS 5	E-10	10	ADDR BUS 9	E-L	L
DATA BUS 4	E-11	11	ADDR BUS 10	E-M	M
DATA BUS 3	E-12	12	ADDR BUS 11	E-N	N
DATA BUS 2	E-13	13	ADDR BUS 12	E-P	P
DATA BUS 1	E-14	14	ADDR BUS 13	E-R	R
DATA BUS 0	E-15	15	ADDR BUS 14	E-S	S
K6	E-16	N.C.	ADDR BUS 15	E-T	T
SING. STP. OUT	E-17	N.C.	PHASE 2	E-U	N.C.
+7.5 UNREG	N.C.	18	$\overline{\text{READ/WRITE}}$	E-V	V
VECTOR FETCH	A-J	19	READ/WRITE	E-W	W
DECODE ENAB.	A-K	20	*+16 UNREG*	***	X
+5 REG.	E-21	N.C.	$\overline{\text{PHASE 2}}$	E-Y	Y
GROUND	E-22	22	RAM R/W	E-Z	N.C.

*** This signal must connect to the K-1008 only, not the KIM!

PROGRAMMING

Programming of the K-1008 to display text and graphics is very straightforward. The display is essentially a matrix of dots with 200 rows of 320 dots per row. For addressing purposes the dots can be numbered from 0 to 63,999 with dot 0 being the upper left-hand corner dot, dot 319 being at the upper right corner, dot 320 being the leftmost dot on the next row down, and 63,999 being the lower right-hand corner dot. Eight horizontally adjecant dots make up one byte of memory with the position of the dots on the display corresponding to the position of the bits in the byte. Thus dot 0 is the leftmost bit (bit 7) of the first byte in the visable memory (generally at memory address 2000_{16}). Conversely dot 319 would be the rightmost bit (bit 0) of the fourtieth byte (typically address 2037_{16}).

Usually graphics programming is performed using the X-Y method of identifying a particular dot position. Although the origin of the coordinate system can be assumed to be anywhere, it is convenient to place it at the lower left corner of the display. Thus all of the displayable points are in the first quadrant and X and Y are always positive numbers. To convert from X-Y point coordinates to a dot number is a simple matter involving evaluation of the equation: DOT # = (199-Y)*320+X . Conversion from the dot number to a byte address and bit number (assuming most significant bit is bit 0) is as follows: BYTE ADDR = VM BASE ADDR + INT(BIT #/8); BIT # = REM(BIT #/8) . Going directly from coordinates to byte address and bit number is as follows: BYTE ADDR = VM BASE ADDR + (199-Y)*40+INT(X/8); BIT # = REM(X/8) . Note that the multiplication by 40 can be accomplished in steps as follows: A*40=(A+A*4)*8 where multiplication by 4 and 8 is accomplished by shifting left 2 and 3 positions respectively. Divison by 8 is accomplished by shifting right 3 positions.

Once the byte and bit addresses are found, the dot may be turned on with the logical OR instruction, turned off with an AND instruction, or flipped with an EOR instruction. It is convenient to write subroutines that accept X and Y coordinates as input and set, reset, flip, write, or read a dot. These would in turn call a subroutine to compute the byte and bit addresses from X and Y coordinates. A more sophisticated subroutine would accept the coordinates of the endpoints of a line and fill in the points forming the closest approximation to the straight line between them. Characters may be drawn either as line segments or a dot matrix by using a font table and calls to the appropriate routine. In special cases drawing speed may be greatly increased by handling the 8 dots in a byte simultaneously.

Since the X coordinate may be as large as 319 which requires 9 bits to represent, the X coordinate must be a double-precision number. Although Y will fit into 8 bits, it too should be double precision for consistency and software compatibility with future display hardware upgrades. It is entirely possible that within two years from now we will see the introduction of a 640 wide by 400 high display using 16K dynamic RAM's!

Although it is a lot of fun to build up graphic subroutines yourself, it is possible that some users would prefer to have the work done for them. A set of utility routines including those discussed above plus some others and a full 320x200 LIFE game are under development and will be available shortly for $20.00 as printed, heavily commented source listings.

TROUBLESHOOTING GUIDE

In the unlikely event that the Visable Memory does not work properly the following suggestions should be tried before returning the board to the factory for repair. This is to the customer's benefit since shipping delays alone often amount to two weeks even if the repairs are made immediately upon receipt at the factory.

If the display is an unsynchronized mess first try adjusting the horizontal and vertical hold controls on the monitor. Some monitors may be super sensitive about the video amplitude so try to adjust that too with the pot or resistors as previously mentioned. A long length of severely mismatched coax cable may distort the sync pulses beyond recovery so try a short length first. Try a friend's monitor or a CCTV monitor at school.

If the display outline itself is stable but the individaul display dots are randomly changing and/or the KIM is unable to write and read data reliably in the VM check your power supply. Although unregulated input voltages are expected, the DC voltage minus the ripple must not be less than 14 volts and 7 volts for the memory and logic supplies respectively. If a voltmeter indicates less than 15 and 8 volts be suspicous. Try a larger filter capacitor in the power supply. If it makes <u>any</u> difference then that is the problem area. If the on-board regulators are bypassed, make sure that the supply voltages measured at the IC pins are within 4% of +12 and +5 and that ripple is less than 50 millivolts peak-to-peak. If the problem persists, carefully adjust the potentiometer according to the instructions on the previous page.

If the test program fails and consistantly points out the same bit at a consistently odd or even address then it is likely that a RAM chip is bad. Prior to shipment the board was continuosly checked with a similar program for 24 hours and no memory errors were allowed. Consult the accompanying chart to determine which RAM is bad and carefully remove it from the socket. Virtually any 22 pin 4K dynamic RAM with high-level clock and a 300NS access/470NS cycle speed may be substututed. Examples are MM5280 (NSC), TMS4060 (TI), 2107A, 2107B (Intel), 2604 (Sig.), and 9060 (AMD). Numbers to avoid are 2107plain and TMS4030. Also if parts are being obtained to populate a blank board it is recommended that the 2107B and the TMS4060 also be avoided. MM5280 RAM's for replacement or bare board population purposes may be obtained from MTU for $5.00 each.

Most other failures will require sharp eyes or an oscilloscope to trace. First examine the board underside to verify that unclipped excess component leads have not bent and shorted lines together. Also check the -5 supply voltage across D4; it should be between -4.5 and -5.5 volts.

Tracing with an oscilloscope is best done by checking the counter chain first. Look at the 8mHz oscillator output and the first 3 counter stages. Then look at the phase comparator output. Adjust the pot until waveforms like the diagram are seen. Check the remainder of the horizontal counter chain and verify proper horizontal unblank and sync signals. Their period should be 64uS exactly. Check the vertical counter chain. The most significant bit of this chain should be on for 256uS and repeat just a shade faster than 60Hz. Check the load enable input to the shift register. Look at the video output signal and verify 3 distinct voltage levels with 20NS transition periods from one level to the next. The video output transistor could have been zapped if the video signal is distorted.

The memory address counter chain should be checked next. Verify proper differentiation of the vertical enable pulse and proper resetting of the address register at the beginning of each vertical sweep. Check that every stage is counting. Check the address multiplexor for proper functioning of each bit. With the KIM monitor examining a VM location synchronize the scope to board addressed (U3-6). Check that the data register is being gated onto the KIM bus at this time. Check the RAM data outputs, they should be stable just prior to data register clocking which occurs 100-150 NS before the end of phase 2.

Check the clock waveform to the RAM chips, it should be a full 12 volts in amplitude and have 25NS or less transition times. If one of the clock driver transistors is bad, replace with the identical number.

If all of this fails to locate the problem, return the board to the factory.

PRINCIPLES OF OPERATION

The K-1008 Visable Memory is basically an 8K dynamic memory board. However instead of letting the memory refresh cycles go to waste, the data read is formatted into a video signal and sent out. Thus, depending on your point of view, it is either a dynamic board with "visable" refresh or a static video display board.

The key to the board's remarkable properties is the 6502 bus itself. A symmetrical 1.0 mHz two-phase clock is used by the KIM-1. The 6502 microprocessor really accesses memory only during Phase 2 with Phase 1 being used for setup. Thus the visable memory can use the 500NS period during Phase 1 to access the memory for display and then turn the memory over to the 6502 during phase 2. RAM chip access times approaching 300NS are required with this scheme but that figure is actually rather slow compared with modern 4K dynamic RAM standards. It is this "flip-flop" sharing between microprocessor and display that makes glitchless display quality possible under all operating conditions.

All of the board's timing is derived from an 8mHz oscillator which is phase-locked to the rising edge of $\overline{\text{PHASE 2}}$ from the KIM. Each cycle of this oscillator represents 1 dot on the display which is also 125 NS. U10 is the voltage controlled oscillator in the phase locked loop which is just a classic Schmidt trigger R-C oscillator. The 500 ohm pot determines the oscillator's free-running frequency and is set for a nominal frequency of 8.0mHz. This simple oscillator is made to act as a voltage controlled oscillator (VCO) by connecting a resistor (2.2K) to the R-C node. Changes in current through this resistor caused by voltage changes at its free end affect the oscillator's frequency. Although the linear VCO range is only 20% or so, this is ample for locking to a fixed crystal-controlled frequency.

The phase detector is also rather unique. Since the phase angle of the lock between the on-board oscillator/counter chain and the KIM's $\overline{\text{PHASE 2}}$ clock affects the data transfer timing, it had to be controlled more tightly than a typical exclusive-or phase detector would provide. A tri-state buffer (U8) fills the bill. A 250NS pulse at a 1.0mHz rate from the first three stages of the counter chain enables the tri-state buffer. The data input to the buffer is $\overline{\text{PHASE-2}}$ from the KIM. Ideal timing for data transfer between KIM and VM occurs when the trailing edge of $\overline{\text{PHASE 2}}$ occurs midway in the enable pulse. Under these conditions the output of the buffer floats for 3/4 of the cycle, is driven high for about 1/8 of the cycle, and then is driven low for the remaining 1/8 of the cycle before floating again. This wildly gyrating buffer output voltage is averaged by the low pass filter formed by R3 and C17. If $\overline{\text{PHASE 2}}$ turns off earlier in the enable window, the buffer output high time becomes less than the low time and the low-pass filter output voltage decreases thus speeding up the VCO which corrects for the error. The converse is true if $\overline{\text{PHASE 2}}$ becomes late. The exact equilibrium point can be changed by adjusting the 500 ohm pot, P1.

The 8.0mHz output of the oscillator is called DOT CLOCK and is used elsewhere to control generation of individual video dots. It is also fed to the counter chain which ultimately divides the 8.0mHz all the way down to 60Hz. The first three stages (part of U12) of the chain divide by 8 producing the DOT 4, DOT 2, and DOT 1 signals which are used to control the memory chip timing and loading of bytes into the video shift register for display. The remaining 6 stages (the remainder of U12 and part of U30) divide by 64 and produce the horizontal scan frequency of 15.625kHz which is a period of exactly 64uS. Decoding logic consisting of portions of U13, U31, and U16 produce two overlapping control signals. Pin 3 output of U13 is a horizontal display enable (unblank) signal. This signal is high for 40uS of the 64 and enables the generation of video data during that period. This of course represents 40 byte times or 320 bit times and sets the width of the image. Other decoding logic (parts of U31) generates a horizontal sync pulse which is 8uS wide and approximately centered in the 24uS interval that HORIZONTAL UNBLK is off. The decoded states of the counter were carefully chosen to insure that no glitches occurred on the horizontal sync pulse.

The trailing edge of the horizontal sync pulse drives the second half of the counter chain consisting of U32 and a portion of U45. Overall this counter divides by 260. Initially it starts with all 9 bits at zero. After 260 horizontal syncs it reaches a count of 260 which is detected by U31 pin 3 which then forces all 9 bits back to zero. The most significant bit of the counter (U45) is a one for only 4 horizontal sync periods so it is used as the vertical sync pulse. An exclusive-or equivalent formed from portions of U29 and U44 combines the horizontal and vertical syncs together to provide a simplified but perfectly adequate composite sync signal to the video signal generator.

U47, an inverter, and a flip-flop provide a glitch-free vertical display enable signal by decoding the second half of the counter chain. This signal is true for 200 horizontal scans and false for the remaining 60. Like the horizontal unblank and sync, vertical sync is intiated midway in the interval that vertical display enable is off. The leading edge of vertical display enable resets the memory scan address counter at the beginning of the frame through R12, R13, C33, and part of U46.

The video shift register, U9, is clocked continuously by the 8.0mHz oscillator. Any data in the register is shifted toward the output and zeroes are shifted in. After 8 shifts the register will start outputting zeroes or black if no new data is loaded. Nand gate U15 allows new data to be loaded only when VERTICAL ENABLE is true, HORIZONTAL UNBLK is true, and the dot counter portion of the counter chain is at STATE 7. When all of these conditions are satisfied, the next 8.0mHz clock pulse loads the shift register rather than shifting it. The memory timing has been carefully set up so that data from the memory is available when the shift register needs it. Since the 76LS166 is a synchronous load device, there is no problem with the first or last dot of a byte being wider or narrower than the other dots. A fourth input to the shift register load enable gate is normally always high but 2 of the jumpers at S1 allow it to be connected to true or complement of the most significant memory address counter bit. When in one of these positions, half of the screen is blanked and the other half works normally.

The video combiner consists of a resistor network and two open-collector gates from U14. Output 8 is controlled by the composite sync source and if it is on generates an essentially zero voltage level at the base of Q7. Video black is generated if output 3 is on which is a level of about .8 volts because of R16. If both gates are off the white level of 2.5 volts, set by voltage divider R17 and R18, is produced. Emitter follower Q7 buffers the video coax cable from the realtively high impedance video combiner insuring good signal quality regardless of cable length. Series termination of the line is provided by R14. The overall video amplitude into a 75 ohm standard video cable is about 1.2 volts P-P which doubles under open circuit conditions.

The display memory address counter is 13 bits long and consists of U19, U34, and a portion of U30. Every time the video shift register is loaded with data from memory, the counter increments by one in preparation for the next memory byte. The counter is reset immediately before the first byte is displayed at the upper left corner of the screen. Note that when the display frame is complete and VERTICAL ENABLE becomes false that the counter continues to count during those times that HORIZONTAL UNBLK is true. This maintains memory refresh action during the relatively long vertical blanking period.

A 12 bit 2 input address multiplexor is formed from U20, U33, and U35. This multiplexor selects addresses from the address counter when DOT 4 is high and selects addresses from the KIM when it is low. DOT 4 is roughly the inverse of KIM PHASE 2 but occurs about 50 to 100 NS earlier. The output of the address multiplexor drives the 12 address lines of the RAM array.

Looking now at the KIM side of the interface, U2 buffers the upper three KIM address bus bits and provides them in their true and complement sense. One 3-input gate from U3 in conjunction with 6 of the jumper positions at S1 produces the BOARD ADDRESSED signal when the board is actually addressed. Another gate in U3 also detects address references between E000 and FFFF and generates KIM DECODE ENABLE to allow the KIM monitor ROM's to function when A-K is disconnected from ground. U1, an 8-input nand, detects references between FF00 and FFFF and generates KIM VECTOR FETCH. A germanium diode in series with the gate output simulates the open-collector gate which is required.

The KIM data bus is buffered both to and from the actual RAM array. Data from the bus passes through U6 and U4 on its way to the RAM DATA INPUT pins. The inversion of the data is cancelled by the data inversion inside the RAM itself. Data output from the RAM enters a tri-state latch which is necessary because data from the RAM's has disappeared by the time the KIM uses it. The latches have new data clocked into them at the end of every memory cycle but their contents are gated onto the KIM data bus only when the board is addressed and a write cycle is not being performed.

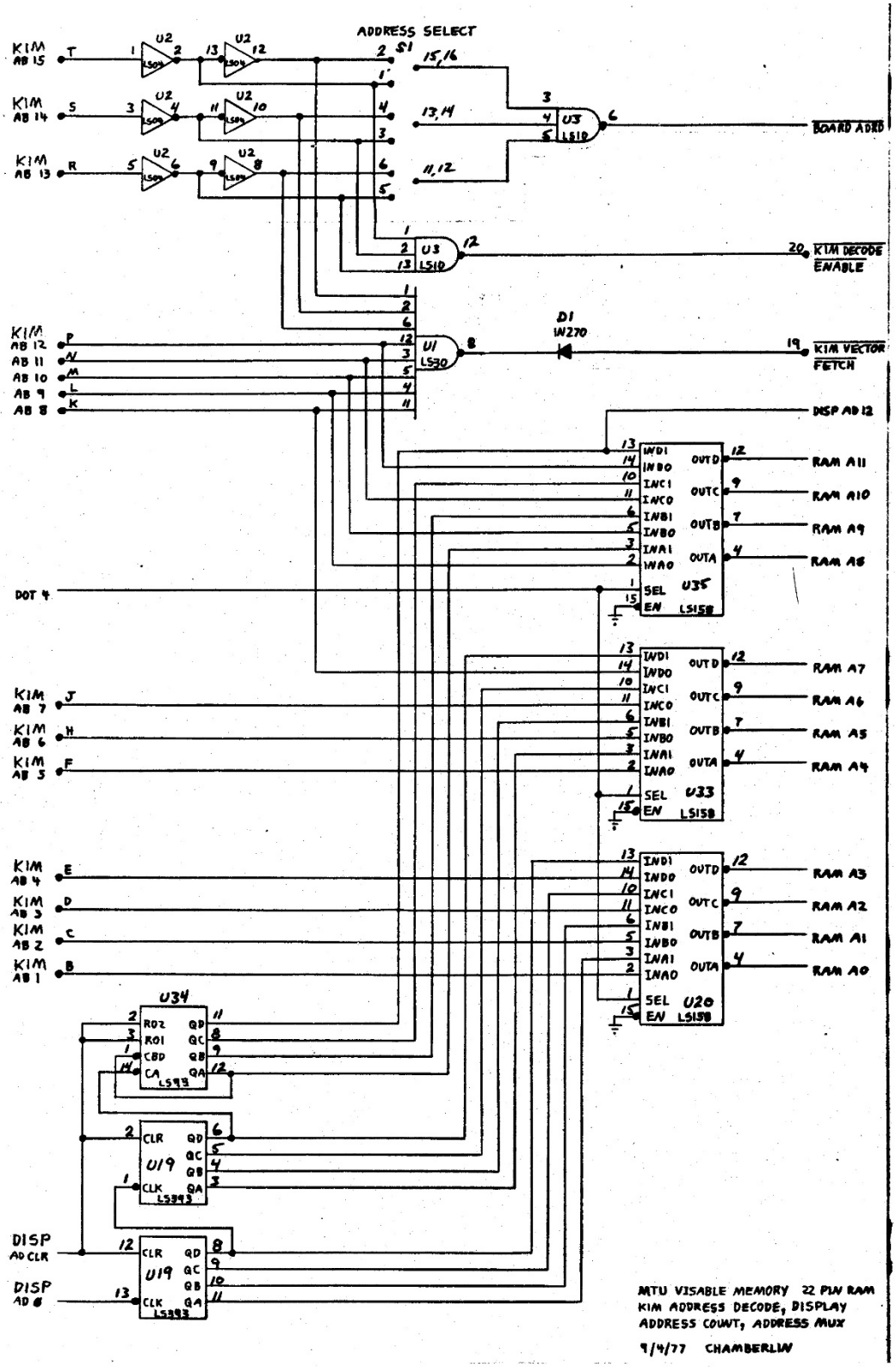
The memory array itself consists simply of 16 4K dynamic RAM chips of the 22 pin variety arranged in a 2 by 8 array. The primary reason for their use over other types of memory chips was cost and a long history of trouble-free reliable performance in large mainframe computers. Also they have the lowest average power consumption in this circuit of all available 4K RAM's. Although National Semiconductor MM5280's are used on factory assembled boards, many manufacturers produce compatible products. Exact details on the operation of 22 pin 4K dynamic RAM's may be found in the manufacturer's data sheets.

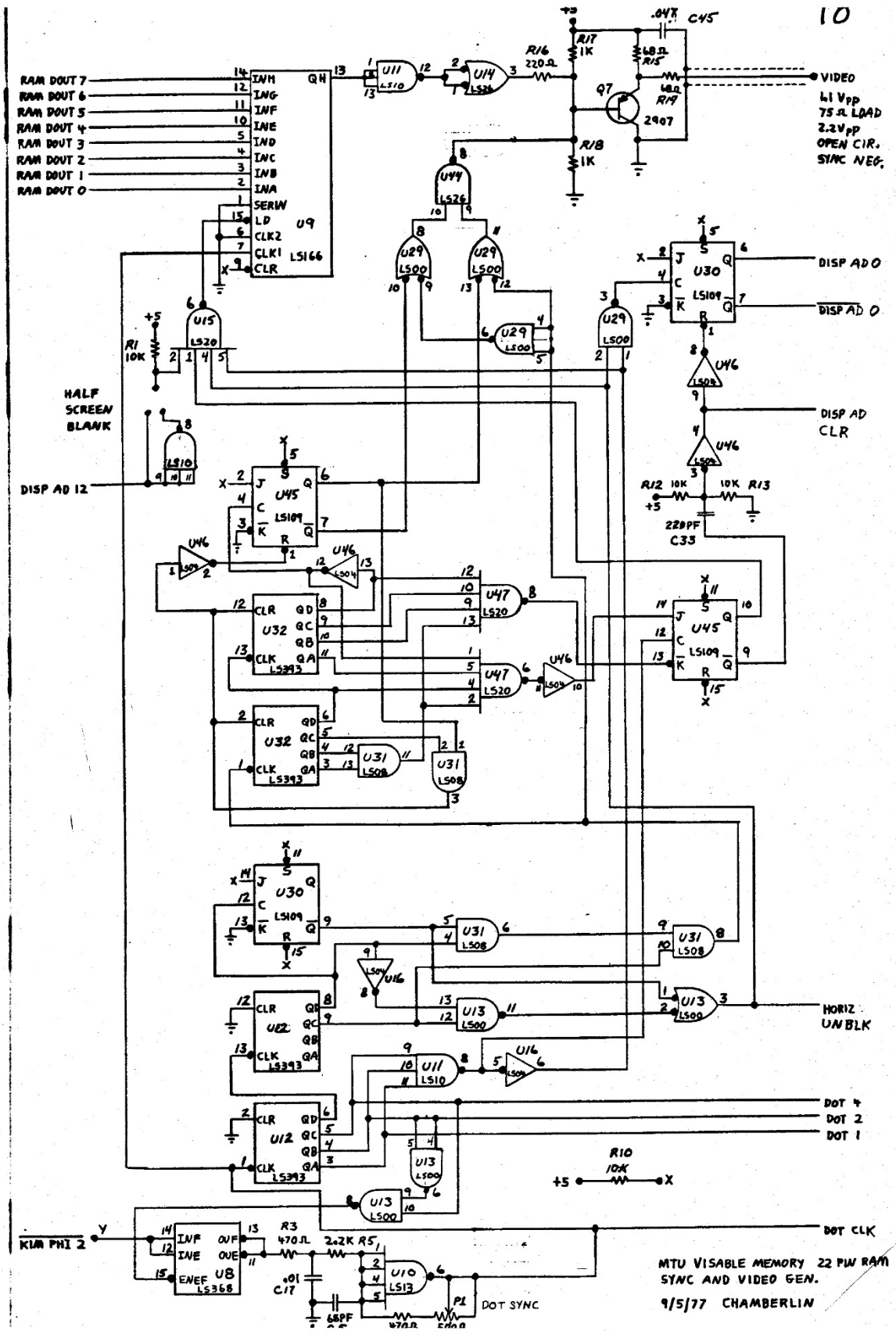
One signal required by the RAM chips is a clock signal that is 12 volts in amplitude. The leading edge of this signal causes the RAM's themselves to latch the state of the address inputs and hold it until clocked again. Data appears at the output after access time, which is typically 200NS, and remains until the clock returns to ground. When not clocked, the RAM's remain completely inactive, draw no power, and float their outputs. The power saver generates a clock pulse only when a memory cycle is actually needed and only clocks the row of RAM's that was actually addressed. At all other times the memory array draws no power at all. If the KIM is not accessing the board, less than 32% of the possible memory cycles are active which rises to about 81% if the KIM is in a tight loop fetching and executing solely on the VM board. An individual RAM chip will see about one half of this activity level. The result is that the memory array runs from stone cold when the KIM is executing elsewhere to just cold when fully utilized.

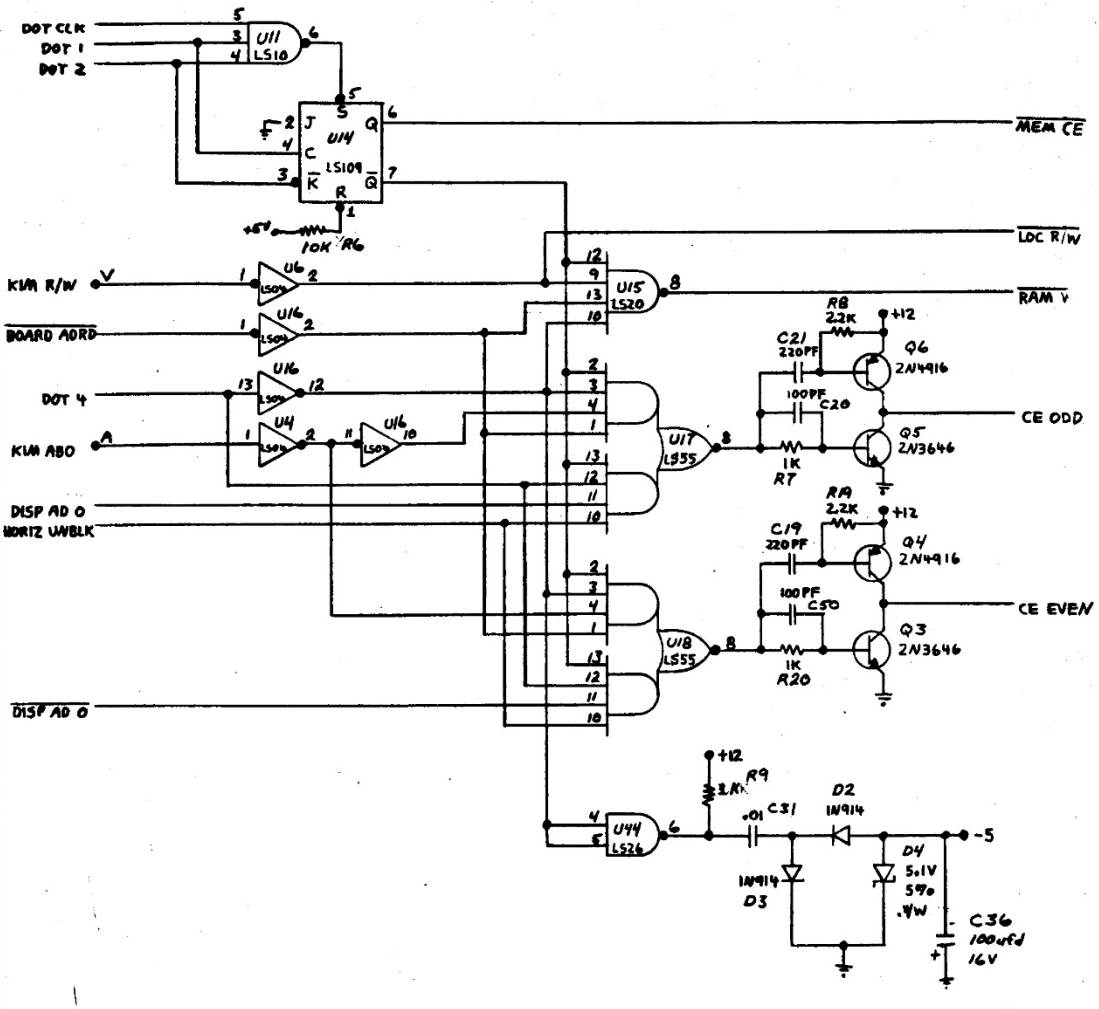
The clock driver circuit that accepts TTL levels from U17 and U18 and translates them to 12 volt levels is exceptionally simple, cheap, power conservative, and high in performance. Like the RAM array, the clock driver draws no power except when a clock pulse is being generated. Performance of the circuit when loaded by 8 RAM chips rivals that of $3 driver IC's with rise and fall times of less than 25NS.

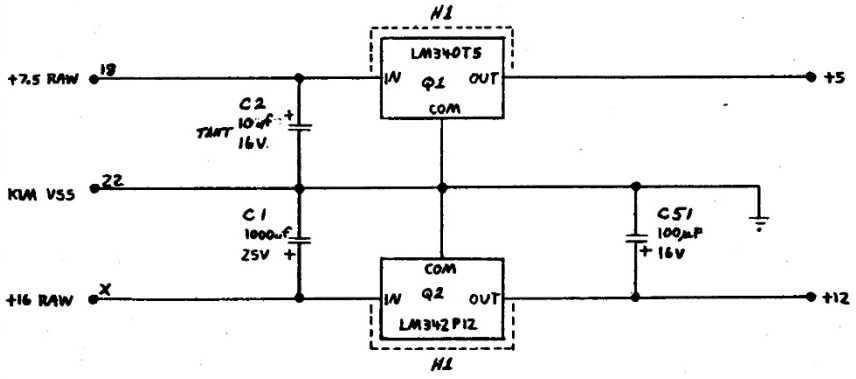
The clock timing generator uses a gate (part of U11) and a flip-flop to generate a precise clock pulse width for the RAM chips. The power saver gating is supplied by U17, U18, and some inverters. The power saver circuit combines clock timing, BOARD ADDRESSED, HORIZ UNBLK, and the least significant memory address bit together and determines which row of RAM's should be clocked if either. U15 generates a write enable pulse coincident with the clock when the conditions necessary for writing are satisfied.

Two 3-terminal regulators supply regulated +5 and +12 volts from unregulated input voltages. Minimal heatsinking is necessary due to the low power consumption of the board. The 1000uF filter capacitor on the +16 unregulated input allows the K-1000 power supply to power 2 Visable Memories as well as a KIM and K-1008 DAC all simultaneously. Negative 5 for the RAM chips is supplied by a charge pump and zener diode regulator. Output 6 of U44 provides a 12 volt P-P signal at 1mHz which drives the network consisting of D2, D3, and C31 which, without D4, would produce about -11 volts. D4 reduces this to -5 volts and in doing so limits the swing at U44-6 to about 6 volts P-P.


ADDRESS SELECT
S1
KIM AB 15
KIM AB 14
KIM AB 13
U2 LS04
15,16
13,14
11,12
U3 LS10
BOARD ADRS
KIM DECODE ENABLE
KIM AB 12
KIM AB 11
KIM AB 10
KIM AB 9
KIM AB 8
U1 LS30
D1 IN270
KIM VECTOR FETCH
DISP AD 12
RAM A11
RAM A10
RAM A9
RAM A8
DOT 4
U35 LS158
RAM A7
RAM A6
RAM A5
RAM A4
KIM AB 7
KIM AB 6
KIM AB 5
U33 LS158
RAM A3
RAM A2
RAM A1
RAM A0
KIM AB 4
KIM AB 3
KIM AB 2
KIM AB 1
U20 LS158
U34 LS393
U19 LS393
DISP AD CLR
DISP AD 0
MTU VISABLE MEMORY 22 PLN RAM
KIM ADDRESS DECODE, DISPLAY
ADDRESS COUNT, ADDRESS MUX
9/4/77 CHAMBERLIN

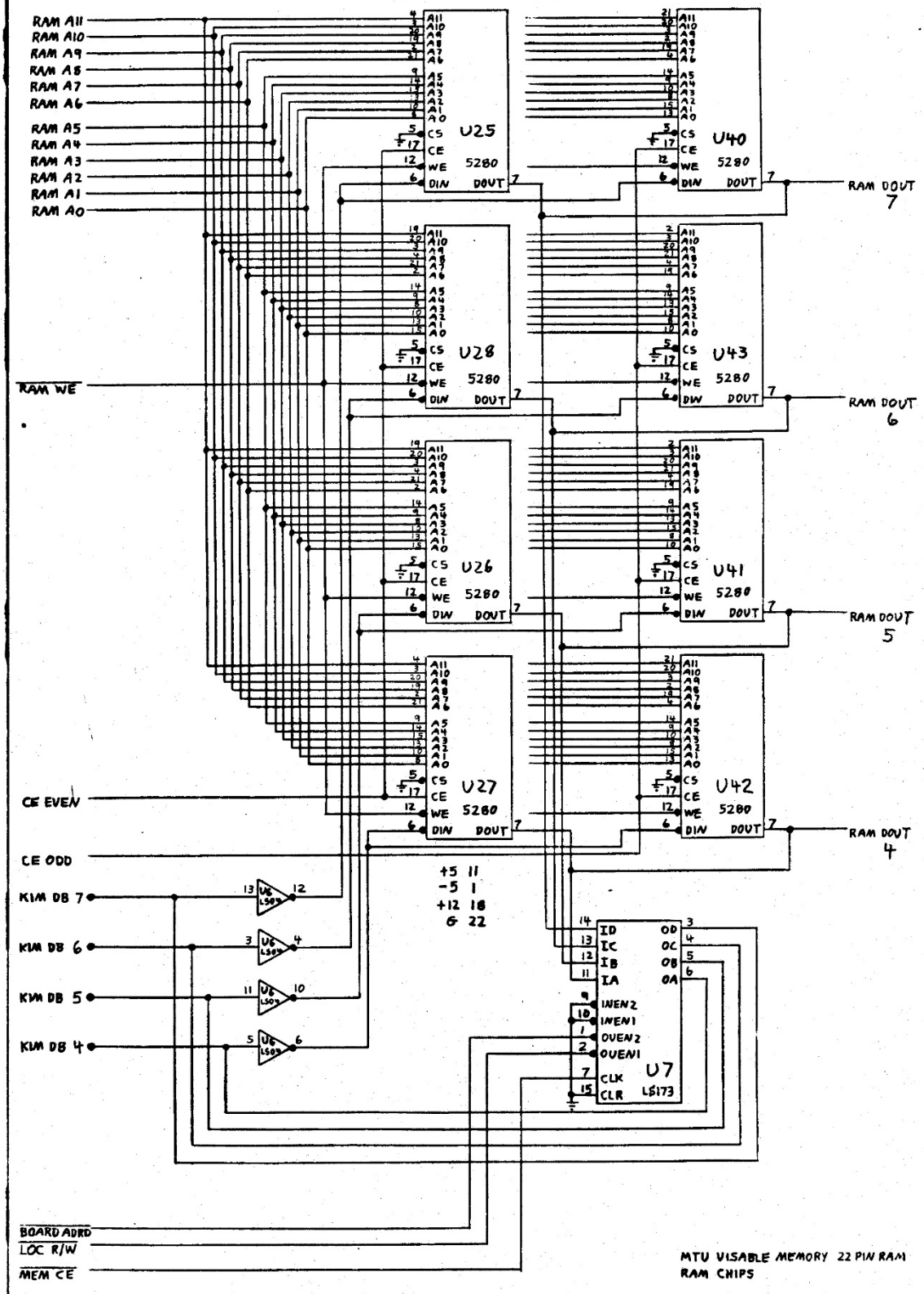

RAM DOUT 7
RAM DOUT 6
RAM DOUT 5
RAM DOUT 4
RAM DOUT 3
RAM DOUT 2
RAM DOUT 1
RAM DOUT 0
U9 LS166
U11 LS10
U14 LS26
R16 220Ω
R17 1K
R18 1K
.047 C45
68Ω R15
68Ω R19
Q7 2907
VIDEO
4.1 Vpp 75Ω LOAD
2.2Vpp OPEN CIR.
SYNC NEG.
U44 LS26
U29 LS00
U30 LS109
DISP AD 0
DISP AD 0
U46 LS04
DISP AD CLR
U15 LS20
R1 10K
HALF SCREEN BLANK
LS10
DISP AD 12
U45 LS109
R12 10K
10K R13
+5
22PF C33
U47 LS20
U32 LS393
U31 LS08
U45 LS109
U30 LS109
U31 LS08
U16 LS04
U13 LS00
HORIZ UNBLK
U32 LS393
U11 LS10
U16 LS04
DOT 4
DOT 2
DOT 1
U12 LS393
R10 10K
+5
X
U13 LS00
U13 LS00
DOT CLK
KIM PHI 2
Y
U8 LS368
R3 470Ω
2.2K R5
.01 C17
68PF
U10 LS13
P1
DOT SYNC
MTU VISABLE MEMORY 22 PW RAM
SYNC AND VIDEO GEN.
9/5/77 CHAMBERLIN

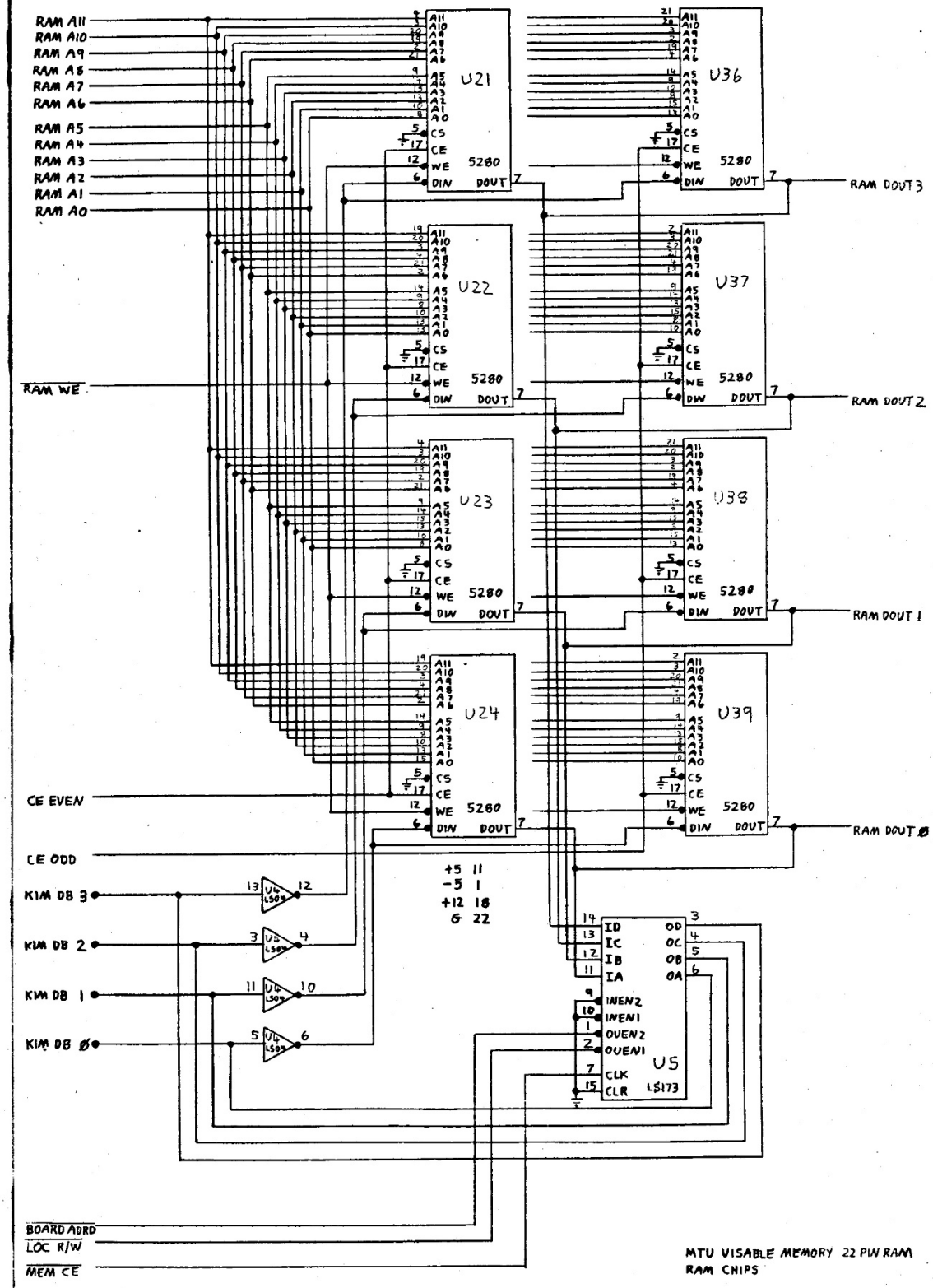




MTU VISABLE MEMORY 22 PIN RAM
TIMING GEN. & POWER SUPPLY

9/15/77 CHAMBERLIN

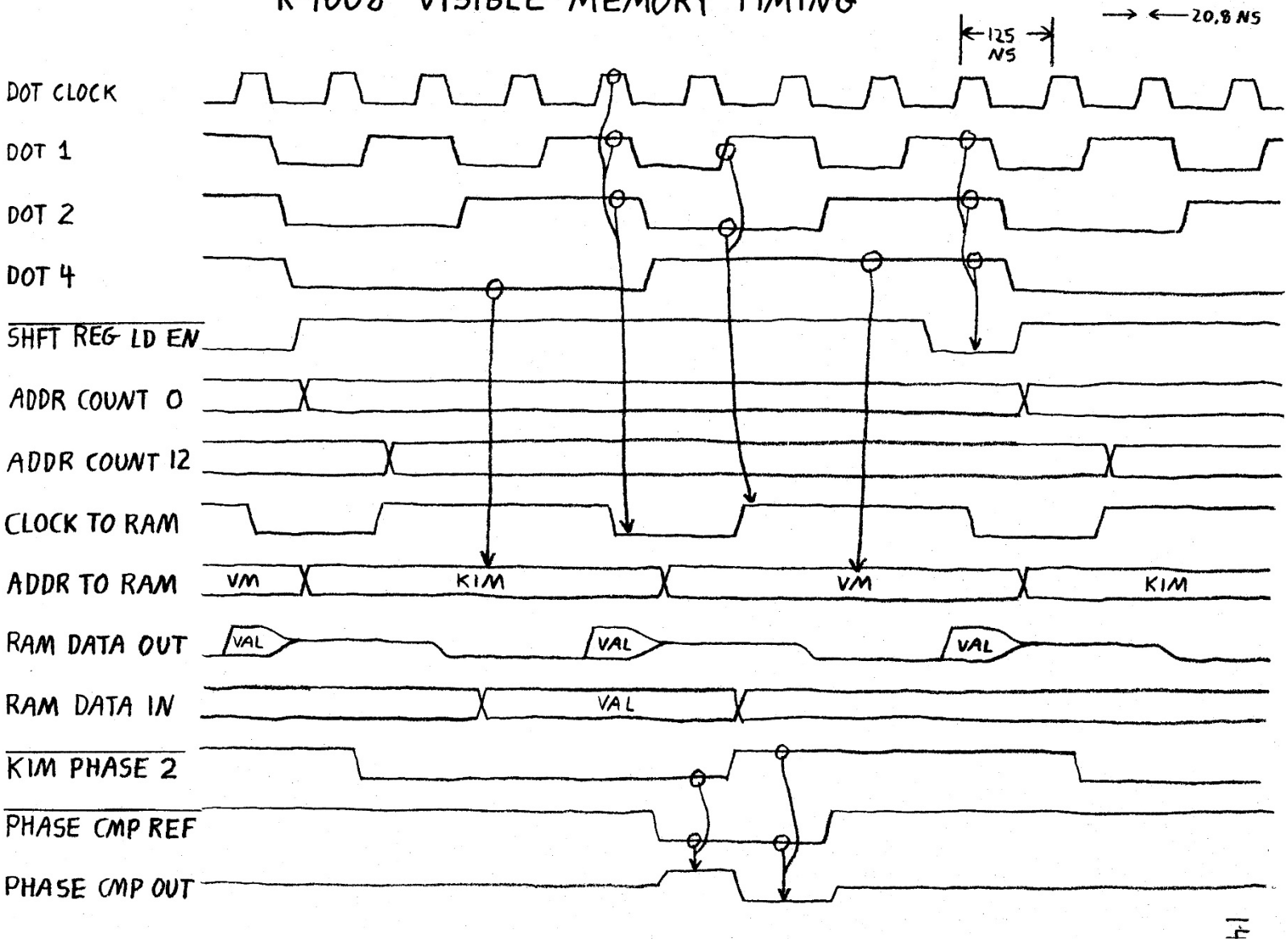

RAM A11
RAM A10
RAM A9
RAM A8
RAM A7
RAM A6
RAM A5
RAM A4
RAM A3
RAM A2
RAM A1
RAM A0
RAM WE
CE EVEN
CE ODD
KIM DB 7
KIM DB 6
KIM DB 5
KIM DB 4
BOARD ADRD
LOC R/W
MEM CE
U25
U26
U27
U28
U40
U41
U42
U43
5280
RAM DOUT 7
RAM DOUT 6
RAM DOUT 5
RAM DOUT 4
+5 11
-5 1
+12 18
G 22
U6 LS05
U7 LS173
MTU VISABLE MEMORY 22 PIN RAM
RAM CHIPS
9/4/77 CHAMBERLIN

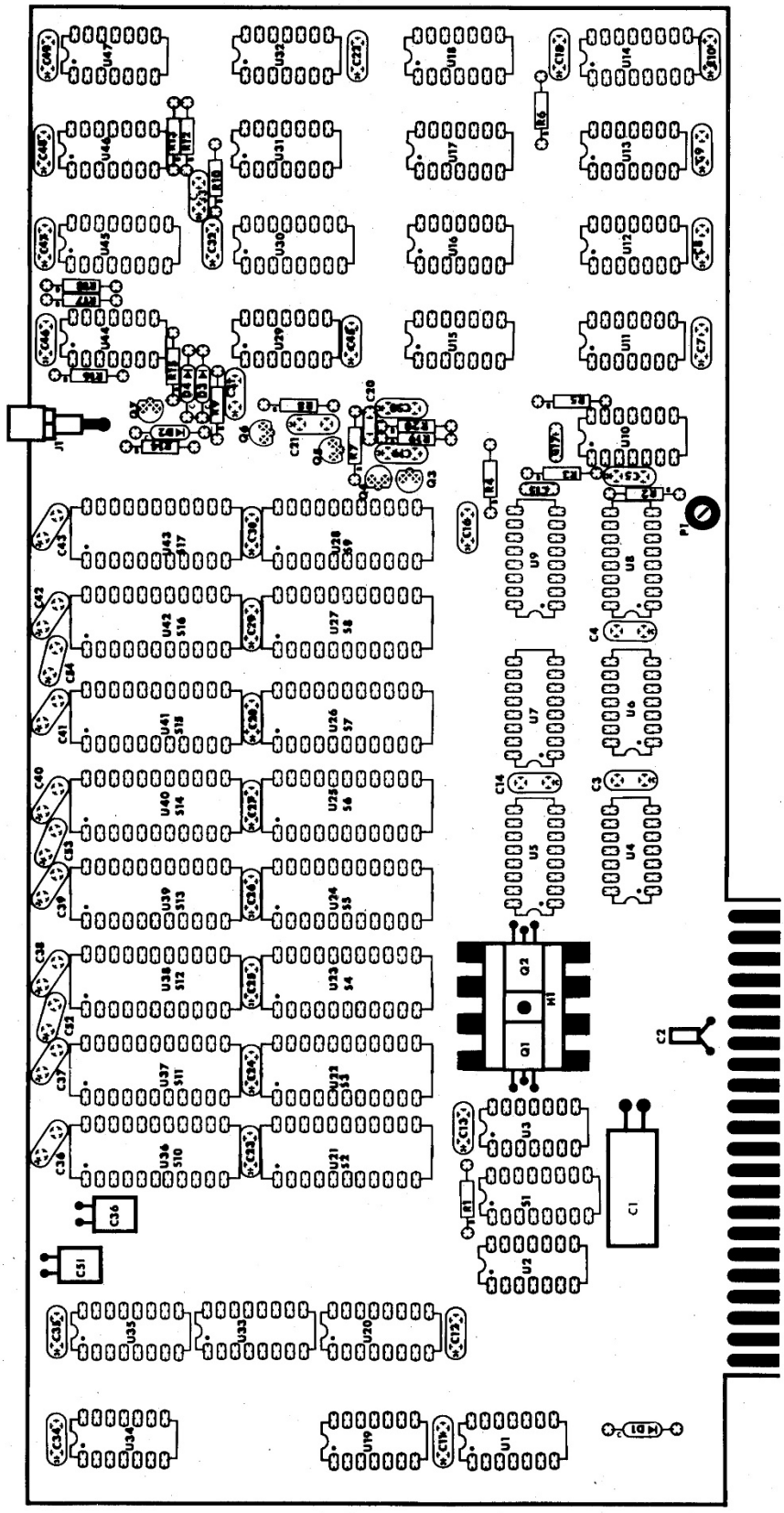


MTU VISABLE MEMORY 22 PIN RAM
RAM CHIPS

9/4/77 CHAMBERLIN

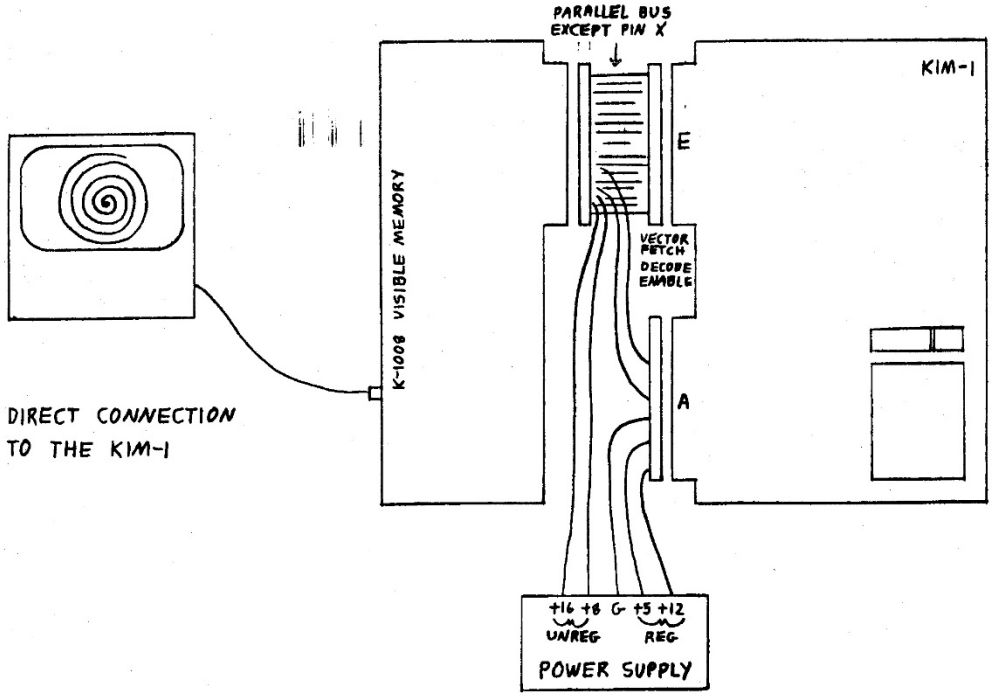
K-1008 VISIBLE MEMORY TIMING

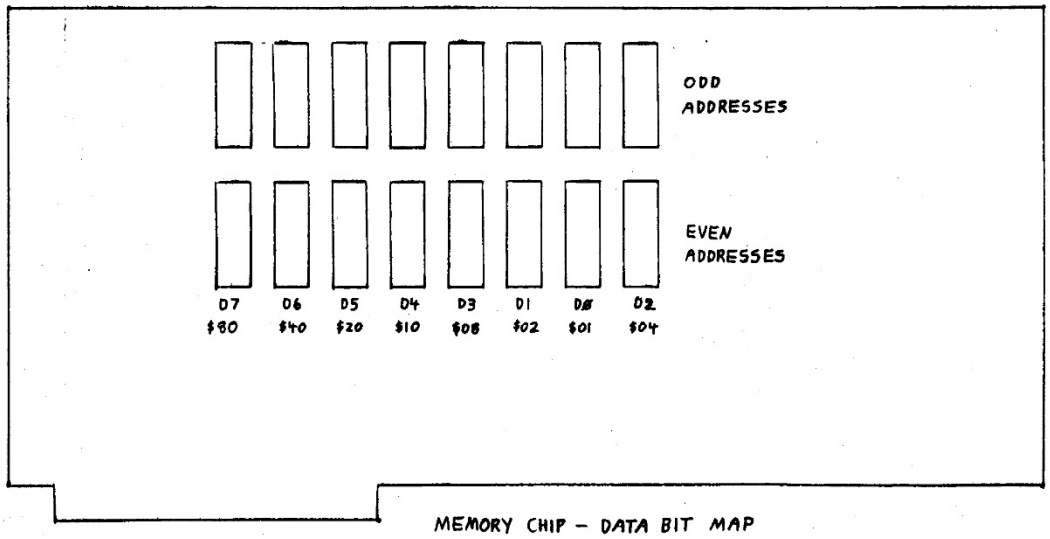




PARTS LIST

1	U1	74LS30
5	U2, U4, U6, U16 U46	74LS04
2	U3, U11	74LS10
2	U5, U7	74LS173
1	U8	74LS368
1	U9	74LS166
1	U10	74LS13
3	U12, U19, U32	74LS393
2	U13, U29	74LS00
3	U14, U30, U45	74LS109
2	U15, U47	74LS20
2	U17, U18	74LS55
3	U20, U33, U35	74LS158
16	U21-U28, U36-U43	MM5280 or equivalent
1	U31	74LS08
1	U34	74LS93
1	U44	74LS26
1	D1	1N270 OR EQU Ge DIODE
2	D2, D3	1N914 OR EUQ Si DIODE
1	D4	5.1V 5% .4W ZENER
1	Q1	LM340T5 VOLTAGE REG.
1	Q2	LM341P12 VOLTAGE REG.
2	Q3, Q5	2N3646 NPN HI SPEED
2	Q4, Q6	2N4916 PNP HI SPEED
1	Q7	2N2907 PNP MED CURRENT AMP
1	S1	16 PIN SOCKET FOR JUMPERS
6	R1, R4, R6, R10, R12, R13	10K 1/4 W. 5% RESISTOR
2	R2, R3	470 OHM 1/4 W. 5%
3	R5, R8, R19	2.2K 1/4W. 5%
5	R7, R9, R17, R18, R20	1K 1/4W. 5%
2	R14, R15	68 OHMS 1/4W. 5%
1	R16	220 OHMS 1/4 W. 5%
1	C1	1000UF 25V. ELECTROLYTIC
1	C2	10UF 16V. TANTALUM
1	C5	68PF NPO DISK CERAMIC
2	C17, C31	.01UF Z5U DISK CERAMIC
3	C19, C21, C33	220PF Z5U DISK CERAMIC
2	C20, C50	100PF Z5U DISK CERAMIC
1	C36	100UF 16V ELECTROLYTIC
1	C51	100UF 16V ELECTROLYTIC
41	C3, C4, C7-14, C15, C16, C18, C22-C30, C32, C34-C43, C45-C49 C52-C54	.047UF OR GREATER 12V Z5U CERAMIC DISK
1	H1	1" SQUARE HEATSINK
1	J1	RCA PHONO JACK
1	P1	500 OHM OR 1K TRIMPOT





MEMORY CHIP – DATA BIT MAP

```
                        .PAGE  'EQUATES AND DATA STORAGE'
                ;       TEST AND EXERCISE PROGRAM FOR THE K-1008 VISABLE MEMORY.  THIS
                ;       PROGRAM IMPLEMENTS TWO TESTS OF THE K-1008 DISPLAY BOARD.
                ;           THE FIRST TEST PERFORMS A GROSS CHECK OF MEMORY FUNCTION
                ;       AND DEMONSTRATES THE ACCURACY OR LACK THEREOF OF THE DISPLAY
                ;       GENERATOR CIRCUITS.  THE PATTERNS GENERATED ARE CKECKERBOARDS
                ;       OF VARIOUS SIZES.  THE DISPLAY MONITOR SHOULD MAINTAIN STABLE
                ;       SYNC EVEN WITH THE LARGE WHITE AND DARK AREAS THAT MIGHT
                ;       APPEAR.  THE DIMENSIONS OF THE RECTANGLES ARE RANDOM WITH AN
                ;       EXPONENTIAL DISTRIBUTION.  16 CHECKERBOARDS ARE DISPLAYED IN
                ;       TEST 1.
                ;           TEST 2 IS A MEMORY FUNCTION TEST.  RANDOM BITS ARE STORED
                ;       IN THE VM IN A SCRAMBLED ORDER WHICH IS ALSO RANDOMLY
                ;       DETERMINED.  AFTER EVERY MEMORY LOCATION IS FILLED, THE SAME
                ;       DATA AND SEQUENCE IS REGENERATED AND MEMORY CONTENTS ARE
                ;       CHECKED AGAINST IT.  THEN A NEW SEQUENCE IS TRIED.  THIS IS
                ;       ITERATED 16 TIMES WITH A SEVERAL SECOND PAUSE BETWEEN THE WRITE
                ;       AND VERIFY PHASE OF THE 16TH ITERATION INSERTED TO VERIFY THE
                ;       FUNCTIONALITY OF DYNAMIC RAM REFRESH.
                ;           THIS PROGRAM IS SPECIFICALLY WRITTEN TO TEST 8K OF
                ;       CONTIGUOUS MEMORY. MODIFICATION TO TEST OTHER SIZES IS POSSIBLE
                ;       BUT THE AMOUNT TESTED MUST BE A POWER OF 2.


                ;       KIM SYSTEM EQUATES

1C22            KIMMON  =       X'1C22          ; ADDRESS OF SAVE MACHINE STATE ENTRY POINT
2000            VMORG   =       X'2000          ; ADDRESS OF VISABLE MEMORY
2000            VMSIZ   =       8192            ; SIZE OF VISABLE MEMORY BOARD
001F            VMSGBT  =       X'1F            ; SIGNIFICANT UPPER ADDRESS BITS FOR VM

                ;       BASE PAGE DATA STORAGE
0000                    .=      0

                ;       MAIN PROGRAM DATA STORAGE

0000 0000       ERRADR: .WORD   0               ; ADDRESS OF DETECTED MEMORY ERROR
0002 00         ERRBTS: .BYTE   0               ; ONES REPRESENT ERROR BITS

0003 00         T1ITCT: .BYTE   0               ; ITERATION COUNT FOR TEST 1
0004 00         T2ITCT: .BYTE   0               ; ITERATION COUNT FOR TEST 2

                ;       DATA STORAGE FOR RANDOM PATTERN TEST

0005 D204       RANDNO: .WORD   1234            ; RANDOM NUMBER REGISTER
0007 0000       SEED:   .WORD   0               ; SAVES SEED FOR VERIFY
0009 0000       ADDRCT: .WORD   0               ; DOUBLE BYTE ADDRESS COUNTER
0008 0000       SCMEMA: .WORD   0               ; SCRAMBLED MEMORY ADDRESS AND ERROR ADDRES

                ;       DATA STORAGE FOR CHECKERBOARD TEST

000D 0000       VMADR:  .WORD   0               ; ADDRESS POINTER FOR VM DATA MANIPULATION
000F 00         VMDATA: .BYTE   0               ; DATA DESTINED FOR VM
0010 00         CKXSZ:  .BYTE   0               ; X SIZE (WIDTH) OF CHECKER RECTANGLE
0011 00         CKYSZ:  .BYTE   0               ; Y SIZE (HEIGHT) OF CHECKER RECTANGLE
```

VMTST K-1008 VISABLE MEMORY
EQUATES AND DATA STORAGE

```
0012 00         CKDTA:  .BYTE   0               ; COLOR OF UPPER LEFT CHECKER RECTANGLE
0013 00         CKDTAX: .BYTE   0               ; WORK COLOR DURING HORIZONTAL SCAN
0014 00         CKDTAY: .BYTE   0               ; WORK COLOR DURING VERTICAL SCAN
0015 00         CKYCT:  .BYTE   0               ; COUNT OF CHECKER HEIGHT DURING VERTICAL
                                                ; SCAN
0016 00         HBYTCT: .BYTE   0               ; BYTE COUNT DURING HORIZONTAL SCAN
```

```
                        .PAGE  'MAIN TEST PROGRAM'
  64 0017               .=     X'200          ; START PROGRAM CODE AT 200
  65
  66 0200 A9E0  MTEST:  LDA    #X'E0          ; INITIALIZE STACK POINTER
  67 0202 9A            TXS
  68 0203 D8            CLD                   ; INSURE BINARY ARITHMETIC
  69
  70            ;       TEST 1  16 CHECKERBOARD PATTERNS
  71
  72 0204 A910  MAIN:   LDA    #16            ; SET 16 ITERATION COUNT
  73 0206 8503          STA    T1ITCT
  74 0208 208B03 MAIN1:  JSR    RNDEXP         ; GET AN EXPONENTIALLY DISTRIBUTED RANDOM
  75                                           ; NUMBER IN A
  76 020B 8510          STA    CKXSZ          ; MAKE IT THE X CHECKER SIZE
  77 020D 208B03        JSR    RNDEXP         ; GET ANOTHER
  78 0210 8511          STA    CKYSZ          ; MAKE IT THE Y CHECKER SIZE
  79 0212 207103        JSR    RAND           ; RANDOMLY DETERMINE UPPER LEFT SQUARE
  80 0215 A505          LDA    RANDNO         ; COLOR
  81 0217 0A            ASLA
  82 0218 A900          LDA    #0             ; SET A TO ALL ZEROES OR ALL ONES ACCORDING
  83 021A E900          SBC    #0             ; TO SIGN OF RANDOM NUMBER
  84 021C 8512          STA    CKDTA
  85 021E 207F02        JSR    CKGEN          ; GENERATE A CHECKERBOARD
  86 0221 20D302        JSR    CKVER          ; VERIFY IT
  87 0224 D03F          BNE    CKERLG         ; GO TO ERROR LOG IF ERROR
  88 0226 C603          DEC    T1ITCT         ; DECREMENT AND CHECK 16 ITERATION COUNTER
  89 0228 D0DE          BNE    MAIN1          ; LOOP UNTIL 16 ITERATIONS DONE
  90
  91            ;       TEST 2 16 PASSES WITH RANDOM DATA, PAUSE IN 16TH PASS
  92
  93 022A A90F  MAIN10: LDA    #15            ; SET 16 ITERATION COUNT
  94 022C 8504          STA    T2ITCT
  95 022E 207103 MAIN11: JSR    RAND           ; NEW PASS, GET A RANDOM
  96 0231 A505          LDA    RANDNO         ; NUMBER IN RANDNO AND SAVE
  97 0233 8507          STA    SEED           ; AS SEED FOR VERIFY
  98 0235 A506          LDA    RANDNO+1
  99 0237 8508          STA    SEED+1
 100 0239 202903        JSR    RNDGEN         ; GENERATE A RANDOM DATA PATTERN IN VM
 101 023C A504          LDA    T2ITCT         ; TEST IF LAST PASS
 102 023E D011          BNE    MAIN15         ; SKIP OVER WAIT IF NOT
 103 0240 A200          LDX    #0             ; WAIT FOR ABOUT 5 SECONDS IN A TIGHT LOOP
 104 0242 A000  MAIN12: LDY    #0
 105 0244 A910  MAIN13: LDA    #16
 106 0246 18    MAIN14: CLC
 107 0247 69FF          ADC    #-1
 108 0249 D0FB          BNE    MAIN14
 109 024B 88            DEY
 110 024C D0F6          BNE    MAIN13
 111 024E CA            DEX
 112 024F D0F1          BNE    MAIN12
 113 0251 A507  MAIN15: LDA    SEED           ; RESTORE RANDOM SEED FOR VERIFY PHASE
 114 0253 8505          STA    RANDNO
 115 0255 A508          LDA    SEED+1
 116 0257 8506          STA    RANDNO+1
 117 0259 204603        JSR    RNDVER         ; VERIFY
```

VMTST K-1008 VISABLE MEMORY
MAIN TEST PROGRAM

```
 118 025C D014          BNE    RNERLG         ; GO TO ERROR LOG IF ERROR
 119 025E C604          DEC    T2ITCT         ; DECREMENT AND CHECK ITERATION COUNT
 120 0260 10CC          BPL    MAIN11         ; LOOP UNTIL 16 ITERATIONS DONE
 121 0262 4C0402        JMP    MAIN           ; REPEAT THE ENTIRE TEST WITH DIFFERENT
 122                                          ; DATA
 123
 124            ;       ERROR LOG ROUTINES
 125
 126 0265 8502  CKERLG: STA    ERRBTS         ; STORE ERROR BITS
 127 0267 A50D          LDA    VMADR          ; STORE ERROR ADDRESS
 128 0269 8500          STA    ERRADR
 129 026B A50E          LDA    VMADR+1
 130 026D 8501          STA    ERRADR+1
 131 026F 4C221C        JMP    KIMMON         ; GO TO KIM MONITOR
 132
 133 0272 8502  RNERLG: STA    ERRBTS         ; STORE ERROR BITS
 134 0274 A508          LDA    SCMEMA         ; STORE ERROR ADDRESS
 135 0276 8500          STA    ERRADR
 136 0278 A50C          LDA    SCMEMA+1
 137 027A 8501          STA    ERRADR+1
 138 027C 4C221C        JMP    KIMMON         ; GO TO KIM MONITOR
 139
```

```
                        .PAGE  'CHECKERBOARD PATTERN GENERATE AND VERIFY ROUTINES'
                ;       CHECKERBOARD PATTERN GENERATOR
                ;       STARTS AT UPPER LEFT CORNER OF SCREEN AND GENERATES A CHECKER-
                ;       BOARD PATTERN.
                ;       ENTER WITH CKXSZ SET TO CHECKER SQUARE WIDTH AND CKYSZ SET TO
                ;       CHECKER SQUARE HEIGHT AND CKDTA SET TO 0 FOR A BLACK UPPER LEFT
                ;       SQUARE OR SET TO X'FF FOR A WHITE UPPER LEFT SQUARE.
                ;       USES ALL REGISTERS, PRESERVES CKXSZ, CKYSZ, CKDTA

027F A920       CKGEN:  LDA     #VMORG/256      ; INITIALIZE ADDRESS POINTER TO BEGINNING
0281 850E               STA     VMADR+1         ; OF VM
0283 A900               LDA     #0
0285 850D               STA     VMADR
0287 A008               LDY     #8              ; INITIALIZE BIT COUNT
0289 A512               LDA     CKDTA           ; COPY CKDTA TO VERTICAL WORK LOCATION
028B 8514               STA     CKDTAY

                ;       START A ROW OF CHECKER BLOCKS

028D A511       CKGNV:  LDA     CKYSZ           ; SET Y SIZE IN CKYCT
028F 8515               STA     CKYCT

                ;       START A HORIZONTAL SCAN

0291 A514       CKGNH:  LDA     CKDTAY          ; COPY VERTICAL CKDTA TO HORIZONTAL WORK
0293 8513               STA     CKDTAX          ; LOCATION
0295 A928               LDA     #40             ; INITIALIZE COUNT OF BYTES GENERATED IN
0297 8516               STA     HBYTCT          ; A HORIZONTAL SCAN

0299 A610       CKGNH1: LDX     CKXSZ           ; SET X SIZE IN INDEX X
029B A513       CKGNH2: LDA     CKDTAX          ; GENERATE A DOT = TO CURRENT VALUE OF
029D 2A                 ROLA                    ; CKDTAX
029E 260F               ROL     VMDATA
02A0 88                 DEY                     ; COUNT DOTS GENERATED
02A1 D016               BNE     CKGNH4          ; SKIP AHEAD IF NOT 8 YET
02A3 A50F               LDA     VMDATA          ; STORE A COMPLETED BYTE IN VM
02A5 910D               STA     (VMADR),Y
02A7 E60D               INC     VMADR           ; INCREMENT VM ADDRESS
02A9 D008               BNE     CKGNH3
02AB E60E               INC     VMADR+1
02AD A50E               LDA     VMADR+1         ; TEST IF ENTIRE VM FILLED
02AF C940               CMP     #VMORG+8192/256
02B1 F01F               BEQ     CKGENF          ; JUMP OUT IF SO
02B3 A008       CKGNH3: LDY     #8              ; RESTORE 8 BIT COUNT
02B5 C616               DEC     HBYTCT          ; TEST IF FINISHED WITH A HORIZONTAL SCAN
02B7 F00C               BEQ     CKGNV1          ; JUMP IF SO
02B9 CA         CKGNH4: DEX                     ; DECREMENT SQUARE WIDTH COUNT
02BA D0DF               BNE     CKGNH2          ; GO GENERATE NEXT DOT
02BC A513               LDA     CKDTAX          ; AT SQUARE BOUNDARY, FLIP COLOR
02BE 49FF               EOR     #X'FF
02C0 8513               STA     CKDTAX
02C2 4C9902             JMP     CKGNH1          ; GO GENERATE NEXT DOT

                ;       FINISH VERTICAL SCAN
```



```

02C5 C615       CKGNV1: DEC     CKYCT           ; DECREMENT SQUARE HEIGHT
02C7 D0C8               BNE     CKGNH           ; GO GENERATE NEXT LINE
02C9 A514               LDA     CKDTAY          ; AT SQUARE BOUNDARY, FLIP COLOR
02CB 49FF               EOR     #X'FF
02CD 8514               STA     CKDTAY
02CF 4C8D02             JMP     CKGNV           ; GO GENERATE NEXT LINE

02D2 60         CKGENF: RTS                     ; RETURN

                ;       CHECKERBOARD PATTERN VERIFY
                ;       STARTS AT UPPER LEFT CORNER OF SCREEN AND VERIFIES A CHECKER-
                ;       BOARD PATTERN.
                ;       ENTER WITH CKXSZ SET TO CHECKER SQUARE WIDTH AND CKYSZ SET TO
                ;       CHECKER SQUARE HEIGHT AND CKDTA SET TO 0 FOR A BLACK UPPER LEFT
                ;       SQUARE OR SET TO X'FF FOR A WHITE UPPER LEFT SQUARE.
                ;       USES ALL REGISTERS, PRESERVES CKXSZ, CKYSZ, CKDTA
                ;       UPON FINDING A MISMATCH BETWEEN THE GENERATED PATTERN AND THE
                ;       CONTENTS OF THE VM, RETURNS WITH THE ERROR BIT SET IN A AND
                ;       THE ADDRESS OF THE ERROR IN VMADR AND VMADR+1

02D3 A920       CKVER:  LDA     #VMORG/256      ; INITIALIZE ADDRESS POINTER TO BEGINNING
02D5 850E               STA     VMADR+1         ; OF VM
02D7 A900               LDA     #0
02D9 850D               STA     VMADR
02DB A008               LDY     #8              ; INITIALIZE BIT COUNT
02DD A512               LDA     CKDTA           ; COPY CKDTA TO VERTICAL WORK LOCATION
02DF 8514               STA     CKDTAY

                ;       START A ROW OF CHECKER BLOCKS

02E1 A511       CKVRV:  LDA     CKYSZ           ; SET Y SIZE IN CKYCT
02E3 8515               STA     CKYCT

                ;       START A HORIZONTAL SCAN

02E5 A514       CKVRH:  LDA     CKDTAY          ; COPY VERTICAL CKDTA TO HORIZONTAL WORK
02E7 8513               STA     CKDTAX          ; LOCATION
02E9 A928               LDA     #40             ; INITIALIZE COUNT OF BYTES GENERATED IN
02EB 8516               STA     HBYTCT          ; A HORIZONTAL SCAN
02ED A610       CKVRH1: LDX     CKXSZ           ; SET X SIZE IN INDEX X
02EF A513       CKVRH2: LDA     CKDTAX          ; GENERATE A DOT = TO CURRENT VALUE OF
02F1 2A                 ROLA                    ; CKDTAX
02F2 260F               ROL     VMDATA
02F4 88                 DEY                     ; COUNT DOTS GENERATED
02F5 D018               BNE     CKVRH4          ; SKIP AHEAD IF NOT 8 YET
02F7 A50F               LDA     VMDATA          ; COMPARE THE COMPLETED BYTE WITH VM
02F9 510D               EOR     (VMADR),Y       ; CONTENTS
02FB D028               BNE     CKVERF          ; JUMP IF ERROR
02FD E60D               INC     VMADR           ; INCREMENT VM ADDRESS
02FF D008               BNE     CKVRH3
0301 E60E               INC     VMADR+1
0303 A50E               LDA     VMADR+1         ; TEST IF ENTIRE VM FILLED
0305 4940               EOR     #VMORG+8192/256
0307 F01F               BEQ     CKVERF          ; JUMP OUT IF SO
```

```
 249 0309 A008      CKVRH3:  LDY    #8              ; RESTORE 8 BIT COUNT
 250 030B C616               DEC    HBYTCT          ; TEST IF FINISHED WITH A HORIZONTAL SCAN
 251 030D F00C               BEQ    CKVRV1          ; JUMP IF SO
 252 030F CA        CKVRH4:  DEX                    ; DECREMENT SQUARE WIDTH COUNT
 253 0310 D0DD               BNE    CKVRH2          ; GO GENERATE NEXT DOT
 254 0312 A513               LDA    CKDTAX          ; AT SQUARE BOUNDARY, FLIP COLOR
 255 0314 49FF               EOR    #X'FF
 256 0316 8513               STA    CKDTAX
 257 0318 4CED02             JMP    CKVRH1          ; GO GENERATE NEXT DOT
 258
 259                ;        FINISH VERTICAL SCAN
 260
 261 031B C615      CKVRV1:  DEC    CKYCT           ; DECREMENT SQUARE HEIGHT
 262 031D D0C6               BNE    CKVRH           ; GO GENERATE NEXT LINE
 263 031F A514               LDA    CKDTAY          ; AT SQUARE BOUNDARY, FLIP COLOR
 264 0321 49FF               EOR    #X'FF
 265 0323 8514               STA    CKDTAY
 266 0325 4CE102             JMP    CKVRV           ; GO GENERATE NEXT LINE
 267
 268 0328 60        CKVERF:  RTS                    ; RETURN
 269
```



```
                            .PAGE  'RANDOM PATTERN GENERATE AND VERIFY ROUTINES'
 270                ;        RANDOM PATTERN STORED IN SCRAMBLED ORDER GENERATE ROUTINE
 271
 272 0329 A900      RNDGEN:  LDA    #0              ; INITIALIZE ADDRESS COUNTER
 273 032B 8509               STA    ADDRCT          ; TO 8192
 274 032D A920               LDA    #VMSIZ/256
 275 032F 850A               STA    ADDRCT+1
 276 0331 207103    STORPH:  JSR    RAND            ; GENERATE A RANDOM NUMBER
 277 0334 205F03             JSR    MADDR           ; FORM A SCRAMBLED MEMORY ADDRESS
 278 0337 A505               LDA    RANDNO          ; STORE A RANDOM BYTE
 279 0339 A200               LDX    #0              ; INDIRECTLY THROUGH SCRAMBLED MEMORY
 280 033B 810B               STA    (SCMEMA,X)      ; ADDRESS AT SCMEMA
 281 033D C609               DEC    ADDRCT          ; DECREMENT ADDRESS COUNTER
 282 033F D0F0               BNE    STORPH          ; AND LOOP IF NOT ZERO
 283 0341 C60A               DEC    ADDRCT+1
 284 0343 D0EC               BNE    STORPH
 285 0345 60                 RTS                    ; RETURN WHEN DONE
 286
 287
 288                ;        RANDOM PATTERN STORED IN SCRAMBLED ORDER VERIFY ROUTINE
 289
 290 0346 A920      RNDVER:  LDA    #VMSIZ/256     ; INITIALIZE ADDRESS COUNTER
 291 0348 850A               STA    ADDRCT+1
 292 034A 207103    VERFPH:  JSR    RAND            ; GENERATE A RANDOM NUMBER
 293 034D 205F03             JSR    MADDR           ; FORM SCRAMBLED MEMORY ADDRESS
 294 0350 A10B               LDA    (SCMEMA,X)      ; GET DATA FROM MEMORY INDIRECTLY
 295 0352 4505               EOR    RANDNO          ; THROUGH SCMEMA
 296 0354 D008               BNE    VERRET          ; GO RETURN ON UNEQUAL COMPARE
 297 0356 C609               DEC    ADDRCT          ; DECREMENT ADDRESS COUNTER
 298 0358 D0F0               BNE    VERFPH          ; AND LOOP IF NOT ZERO
 299 035A C60A               DEC    ADDRCT+1
 300 035C D0EC               BNE    VERFPH
 301 035E 60        VERRET:  RTS                    ; RETURN
 302                ;        SCRAMBLED MEMORY ADDRESS FORMATION ROUTINE
 303                ;        USES ADDRCT AND SEED TO FORM A SCRAMBLED ADDRESS IN SCMEMA
 304
 305 035F A507      MADDR:   LDA    SEED            ; GET LOWER BYTE OF RANDOM NUMBER
 306 0361 4509               EOR    ADDRCT          ; EXCLUSIVE-OR WITH LOWER ADDRESS
 307 0363 850B               STA    SCMEMA          ; LOWER BYTE OF RESULT
 308 0365 A508               LDA    SEED+1          ; GET UPPER BYTE OF RANDOM NUMBER
 309 0367 450A               EOR    ADDRCT+1        ; EXCLUSIVE-OR WITH UPPER ADDRESS
 310 0369 291F               AND    #VMSGBT         ; SAVE SIGNIFICANT BITS OF RESULT
 311 036B 18                 CLC
 312 036C 6920               ADC    #VMORG/256      ; ADD IN FIRST PAGE NUMBER OF BOARD
 313 036E 850C               STA    SCMEMA+1        ; BEING TESTED
 314 0370 60                 RTS                    ; RETURN
 315
 316
 317                ;        RANDOM NUMBER GENERATOR SUBROUTINE
 318                ;        ENTER WITH SEED IN RANDNO
 319                ;        EXIT WITH NEW RANDOM NUMBER IN RANDNO
 320                ;        USES 16 BIT FEEDBACK SHIFT REGISTER METHOD
 321                ;        DESTROYS REGISTER A AND Y
 322
 323 0371 A008      RAND:    LDY    #8              ; SET COUNTER FOR 8 RANDOM BITS
```

```
VMTST K-1008 VISABLE MEMORY
RANDOM PATTERN GENERATE AND VERIFY ROUTINES

  324 0373 A505      RAND1:  LDA     RANDNO          ; EXCLUSIVE-OR BITS 3, 12, 14, AND 15
  325 0375 4A                LSRA                    ; OF SEED
  326 0376 4505              EOR     RANDNO
  327 0378 4A                LSRA
  328 0379 4A                LSRA
  329 037A 4505              EOR     RANDNO
  330 037C 4A                LSRA
  331 037D 4506              EOR     RANDNO+1        ; RESULT IS IN BIT 3 OF A
  332 037F 4A                LSRA                    ; SHIFT INTO CARRY
  333 0380 4A                LSRA
  334 0381 4A                LSRA
  335 0382 4A                LSRA
  336 0383 2606              ROL     RANDNO+1        ; SHIFT RANDNO LEFT ONE BRINGING IN CARRY
  337 0385 2605              ROL     RANDNO
  338 0387 88                DEY                     ; TEST IF 8 NEW RANDOM BITS COMPUTED
  339 0388 D0E9              BNE     RAND1           ; LOOP FOR MORE IF NOT
  340 038A 60                RTS                     ; RETURN
  341
  342                ;       EXPONENTIALLY DISTRIBUTED RANDOM NUMBER SUBROUTINE
  343                ;       RULES OF USE SAME AS RAND, 8 BIT RESULT RETURNED IN A
  344                ;       AN EXPONENTIAL DISTRIBUTION MEANS THAT THE PROBABILITY OF A
  345                ;       RESULT BETWEEN 10 AND 20 IS THE SAME AS THE PROBABILITY OF A
  346                ;       RESULT BETWEEN 100 AND 200.
  347                ;       NOTE THAT THE PROBABILITY OF A ZERO RESULT IS ZERO.
  348
  349 038B 207103    RNDEXP: JSR     RAND            ; GET TWO NEW RANDOM BYTES
  350 038E 207103            JSR     RAND
  351 0391 A505              LDA     RANDNO          ; CONVERT ONE OF THE BYTES TO A RANDOM
  352 0393 2907              AND     #7              ; VALUE BETWEEN 0 AND 7 AND PUT IN Y AS A
  353 0395 A8                TAY                     ; SHIFT COUNT
  354 0396 C8                INY
  355 0397 A506              LDA     RANDNO+1        ; GET THE OTHER RANDOM NUMBER AND SHIFT IT
  356 0399 88        RNDXP1: DEY                     ; RIGHT ACCORDING TO Y
  357 039A F004              BEQ     RNDXP2
  358 039C 4A                LSRA
  359 039D 4C9903            JMP     RNDXP1
  360 03A0 0900      RNDXP2: ORA     #0              ; TEST FOR A ZERO RESULT
  361 03A2 F0E7              BEQ     RNDEXP          ; PROHIBIT ZERO RESULTS
  362 03A4 60                RTS                     ; RETURN
  363
  364 0000                   .END
NO ERROR LINES
```