64K DRAM BOARD

Mucking around in memory? Seeking space? Look no further, here's a bounty (no connection with those distracting TV ads) of bits, rapacious in real-estate, for your 6502 or 6800 system to gorge itself on. Design and development by Bob Campbell.

Most microcomputer users find out fairly quickly that there is no such thing as too much memory. But even today with memory as cheap as it is, many systems are on sale with less, often considerably less than the 64K that most eight-bit microprocessors are capable of addressing.

The independent suppliers are usually very quick to provide units to fill this gap, but one system not well covered in this respect is the Tangerine Micron/Microtan 65. Until recently, there was only the TANRAM, but now there is the CMOS alternative. However, despite advantages in power consumption and battery back-up, the CMOS unit, like the TANRAM, is large and fairly expensive. More than one board is required to provide the maximum possible memory.

The approach here is to use the highest density dynamic RAM chips readily available and allow the user to access all of it except where it would clash with essential EPROM, I/O or CPU board RAM. This leads to an extremely flexible and cost effective system. Although specifically designed for the Microtan 65 computer together with either a disc system or TUG's Eprom Storage Card (the MOS Disc concept) the design retains enough flexibility to accommodate almost any desired configuration of computer and operating system, the only prerequisite is a 6502 or 6800 CPU.

Design

The board uses the latest 64K by 1 bit dynamic RAM chips, TMS 4164-15. These are decoded into 64 1K blocks, with all but four of the blocks used in its standard configuration. Making almost 6¼% of the RAM effectively redundant may at first sight seem a little extravagant, however even allowing for this the cost per K is less than £1.00. If one adds the other savings on hardware, sockets, power supply requirements board space etc., the 64K chip route stands out above all the other alternatives.

The heart of the system is the 74LS608 **memory cycle controller (MCC).** This chip generates all the signals the RAM requires to perform the two types of cycles necessary for proper operation. The MCC generates these signals from the CPU's clocks $\phi 1$ and $\phi 2$ together with the decoded signal RE, RAM enable. It is important not to confuse this signal with the Tanbus signal RAME. The only signals used from the bus are the address and data lines together with R/$\overline{\text{W}}$, $\phi 1$ and $\phi 2$, and because of this and the use of a PROM address decoder, this board is very flexible in design and easily adapted to suit other systems.

Dynamic RAMs

The two great advantages of dynamic RAM are its extremely low power consumption and its packing density. This is achieved by the design of the actual memory element which is in fact a very small capacitor. The logic level stored being defined by the presence or absence of a charge on that capacitor. Because all capacitors have a finite leakage, the charge on the capacitors must be periodically topped up. This procedure is called *refreshing* and is accomplished by performing what is known as a $\overline{\text{RAS}}$ only refresh cycle.

This $\overline{\text{RAS}}$ only refresh cycle consists of first setting up an eight bit address at the input latches and strobing $\overline{\text{RAS}}$ low, while maintaining $\overline{\text{CAS}}$ high. The complete chip is refreshed when all 256 row addresses have been treated similarly. Data retention is assured if all these 256 cycles are completed at least once every 4 msec.

Apart from the necessity to refresh every 4 msec there is one other penalty to pay for the 16 pin packing density and that is the multiplexed address bus. Figure 1 shows the internal architecture of the 4164.

To address every memory element within the IC, 16 address bits must be applied. These are separated into the row address and the column address, each latched onto the multiplexed address bus

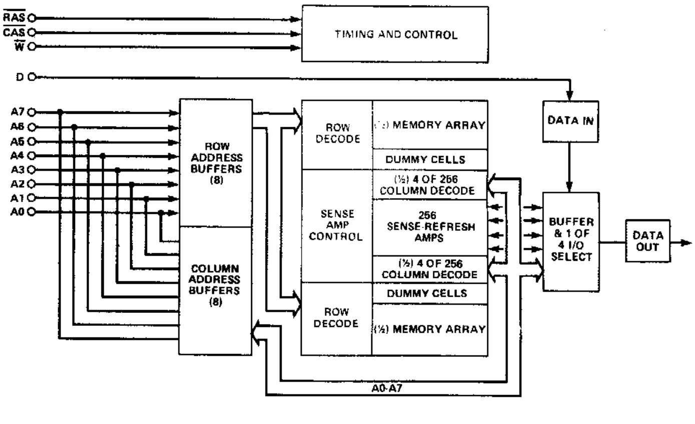


Fig.1 Internal architecture of the TMS4164 DRAMs used in the project.

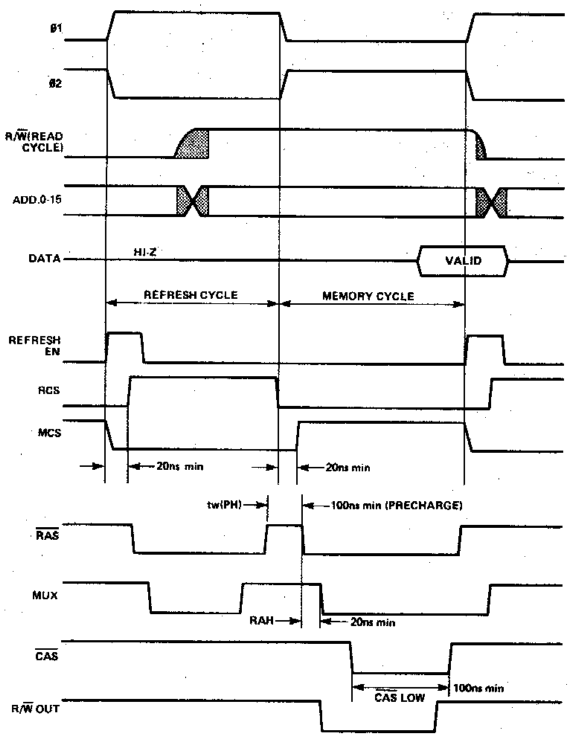


Fig. 2 Processor and memory cycle controller timing.

upon the appropriate signal $\overline{RAS}$ or $\overline{CAS}$.

In full, the memory cycle consists of five stages. Firstly the row address is set up on the eight address pins and $\overline{RAS}$ pulled low. Then the address multiplexers are switched placing the other eight bits, the column address, onto the address pins and $\overline{CAS}$ pulsed low. This last operation enables the chip and, depending upon status of the R/$\overline{W}$ line, enables the input or output buffers, thus completing a read or write cycle.

There are two other possible types of cycle, the *page mode* . *read/write* and the *read modify write cycles*. However since neither of these apply to the 6502 or 6800 type of processor it is not necessary to consider them further here.

It is important to note that the 6502 operates in what is known as the **early write cycle** where the R/$\overline{W}$ line is set up long before $\overline{CAS}$ goes low. This enables the data in (D) and data out (Q) pins to be connected together and thus have a common data bus. Obviously the sequence and timing of the two cycles, refresh and memory, is extremely important. The $\overline{RAS}$ only refresh cycle is particularly significant for two reasons: firstly, it is necessary to perform it regularly (256 times every 4 msec), and secondly, it is effectively a dead cycle, when the CPU cannot access memory.

Refresh cycles can be carried out in either *burst mode* or *hidden transparent mode*. Burst refresh is a technique where all the memory elements are refreshed consecutively whilst the processor is held in a wait or halted state. This dead time is called the **refresh overhead**, which, more accurately, is defined as the ratio of the time taken to refresh all the memory elements and the maximum refresh interval. In well-designed systems with the 4 msec 64K rams this overhead can be as low as 2%. As the circuitry needed to maintain this type of refresh system is complex it is not commonly used outside the realms of very fast microcomputors, minis and mainframe systems.

The other technique, hidden refresh, is the more commonly used. This technique relies upon the fact that the CPU will always have a period within any instruction or machine cycle when it will not access the system bus, and one refresh cycle can be accomplished during this period. Thus after a maximum of 256 instruction cycles all the memory elements will have been serviced. This technique has the great advantage of a zero refresh overhead rate and is totally transparent to the CPU and thus the user.

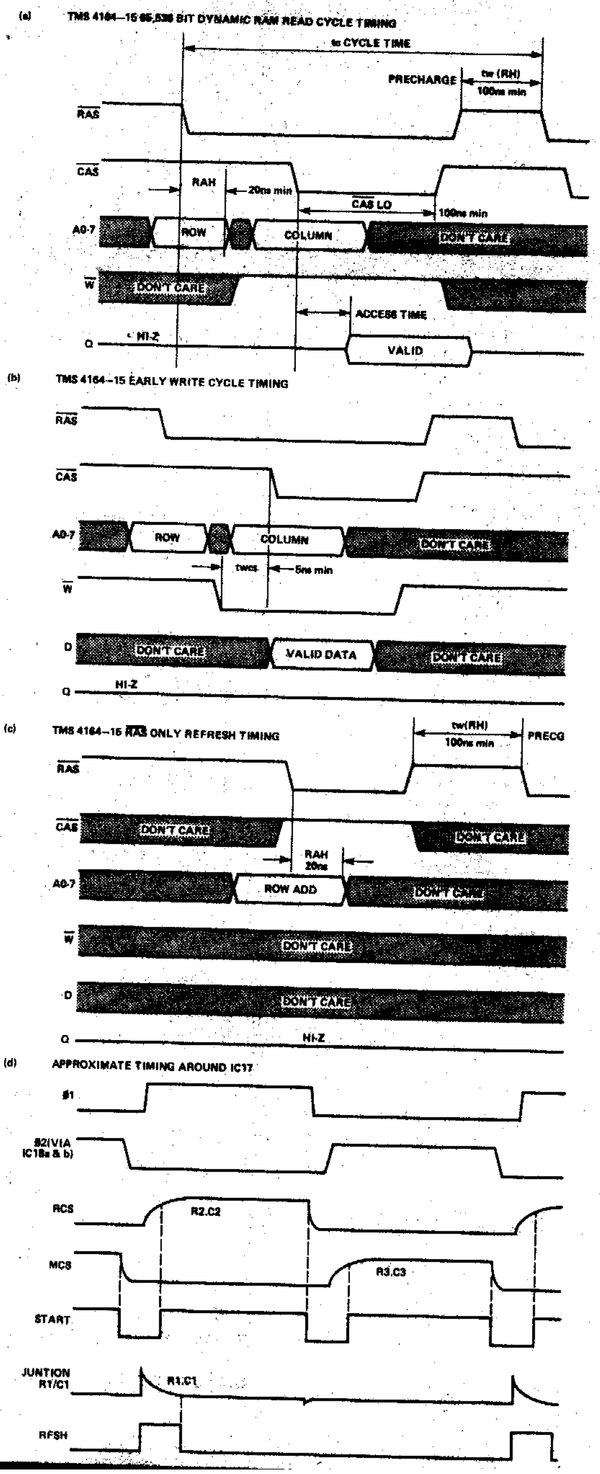
In this design the two cycles, refresh and memory, are sequenced by the main CPU clocks ϕ1 and ϕ2. While ϕ1 is high, the CPU sets up the R/$\overline{W}$ and address lines, the rising edge of ϕ2 signifying a valid memory address. This edge of ϕ2 is normally used to enable the address and data buffers. Thus while ϕ1 is high, the CPU is normally isolated from the system bus, and the refresh cycle can be accomplished during this period. In addition by using ϕ1 to clock the eight bit refresh row address counter all 256 row addresses can be refreshed sequentially. Figure 3 shows exactly the relationship and timing of these events.

PROM Program Design

The memory map of the RAM board is controlled directly by the TBP24S10 PROM, which acts as a complex address decoder. Before programming the PROM, the desired memory map must be established. The minimum requirement for most systems will be the system monitor, the I/O area and unless there is a serial VDU as the screen, some screen memory. Some systems use a relocatable area of memory for the screen RAM, the video controller accessing the system bus directly. If the target system is of this type then no provision should be made for the screen RAM in the PROM program. Remember the overriding factor when designing the memory map is that there must not be two components within the system which have the same address. Taking the standard configuration of the Microtan as our worked example, the minimum memory map is as shown in Fig 4.

Once you've determined the memory map(s) required, the upper six address lines should be written out bit fashion (bit by bit . . .?). Each bit corresponds to a PROM address bit; however because of the PCB board layout, the one-to-one cor-

Fig. 3 Memory timing for various operations and approximate timing round the MCC.



respondance is not in numerical order.

In addition, by using the two extra PROM address lines A7 and A8, there is the facility to have up to four programs and therefore four memory maps resident on the board at one time, selectable by means of the DIL switch SW1. Using the two tables 1 and 2 it is possible to calculate all the PROM addresses which are required to be 'blown'.

Remember that PROMs are not erasable, once a memory location is altered from the "all 1's" condition, it cannot be reversed. There is however an escape route if a mistake is made during programming. The program is created by blowing only the operative bits within the data word from a 1 to a 0. In this design, only one of the four bits available is used (bit 4). If an error is made during programming, then it is possible to use an alternative bit by breaking the PCB track at pin 9 IC15, installing a link to either pin 10, 11 or 12 (bits 1-3 inc.) and reprogramming the PROM using the appropriate data word. (Alternatively, this would make it possible to hold a total of 16 memory maps in the PROM).

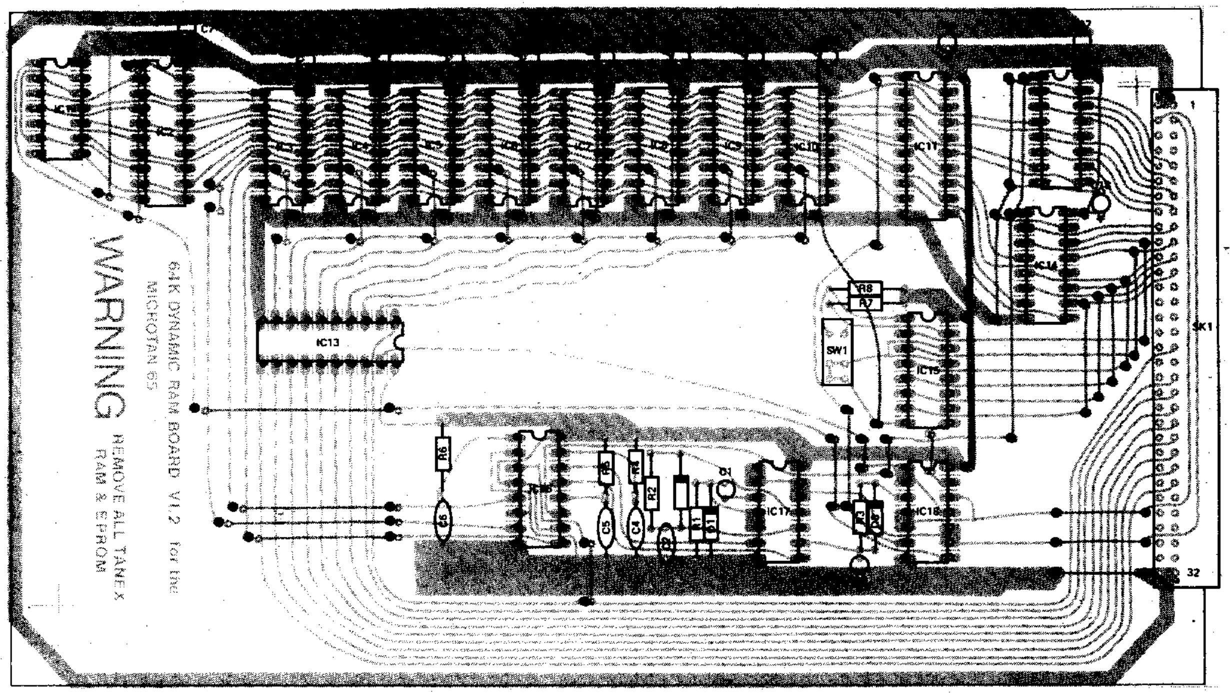
It is beyond the scope of this article to describe the methods for actually programming the PROM, suffice it to say that the amount of programming by the nature of it's use, is small, so it would be feasible to use the switchbox type of programmer.

Construction and Setting Up

The construction of the board is very straightforward, particularly if the PCB design presented here is followed exactly: there are, after all, only 18 ICs. The PCB is a double-sided design but to keep costs down it doesn't use plated-through holes. To make the necessary interconnections, track pins or short lengths of wire must be soldered between the two in the positions marked on the overlay diagram with a black dot. These pins must be soldered in first,

AREA	HEX ADD.	SIZE
A) TANBUG	FFFF F800	2K
RAM	F7FF C000	14K
B) I/O	BFFF BC00	1K
RAM	BBFF 0400	46K
C) CPU BOARD RAM	03FF 0000	1K

Fig. 4 Minimum memory map for the Microtan.



PARTS LIST

RESISTORS

R1, 2, 3	1k0 hystab metal film 1%
R4, 6	4k3 hystab metal film 1%
R5	1k1 hystab metal film 1%
R7, 8	1k0 carbon ¼W 5%

CAPACITORS

C1	100p ceramic plate 2% or better, or silvered mica 1%
C2	150p ceramic plate 2% or better, or silvered mica 1%
C3	68p ceramic plate 2% or better, or silvered mica 1%
C4	220p ceramic plate 2% or better, or silvered mica 1%
C5	68p ceramic plate 2% or better, or silvered mica 1%
C6	120p ceramic plate 2% or better, or silver mica 1%
C7, C17	10u tant or low leakage solid aluminium electrolitic
C8-C16, C18	1u tantalum

SEMICONDUCTORS

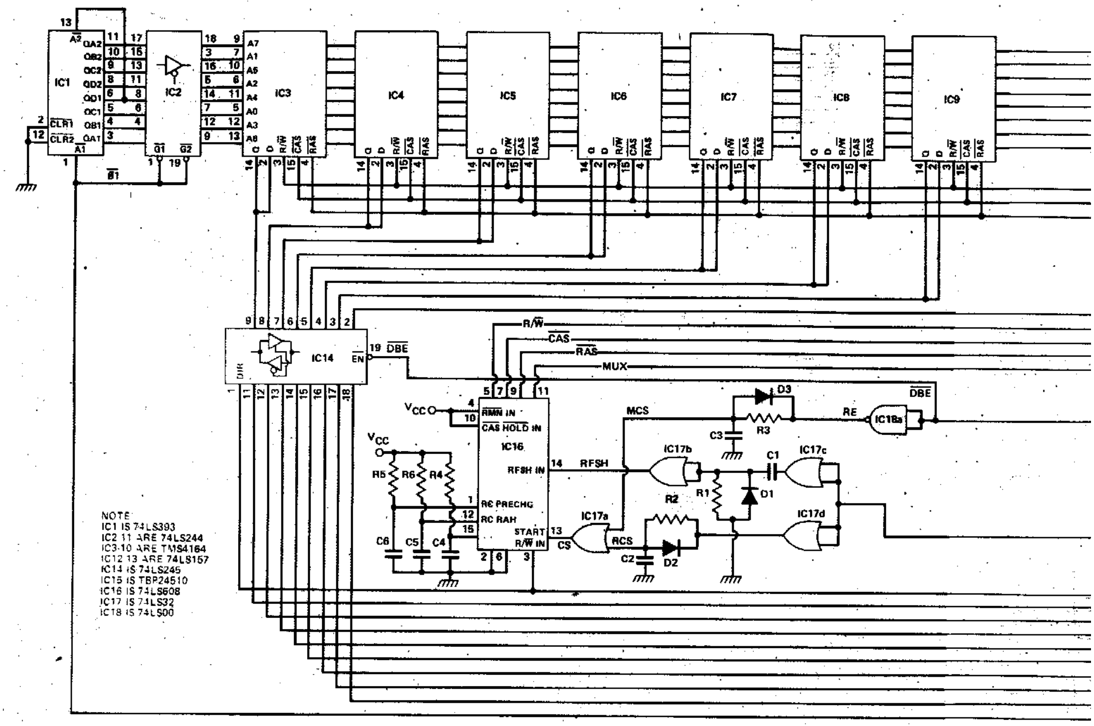
IC1	74LS393
IC2, 11	74LS244
IC3-10	TMS4164-15p
IC12, 13	74LS157
IC14	74LS245
IC15	TBP24S10
IC16	74LS608
IC17	74LS32
IC18	74LS00
D1, 2, 3	1N4148

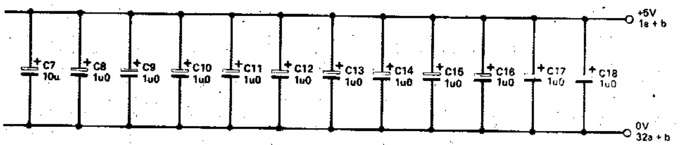
MISCELANEOUS

DIN 41612 64 way double-sided connector; DIL 2 pole on/off switch; DIL sockets: 3 off 20 pin, 12 off 16 pin, 3 off 14 pin; PCB.

BUYLINES

One or two not so easy to obtain items here. The TBP24S10 PROM and 74LS608 memory cycle controller chip were tracked down to Farnell Electronic Components Ltd, Canal Road, Leeds LS12 2TU. At £2.42 and £6.44 respectively plus 55p p&p plus VAT these shouldn't break the bank. The 1% metal film resistors are available from Rapid, Cricklewood, Watford and many others. The specified memories and other TTL devices are advertised by Midwich Computer Company Ltd. And, in case you hadn't guessed, the PCB will be available through our own service (see page 77).





POWER LINES + DECOUPLING CAPACITORS
1u0 ALL TANT
10u TANT OR LOW LEAKAGE
SOLID ELECTROLYTIC

Fig. 5 Circuit diagram of the complete project.

prior to any other components, as there are some beneath the DIL sockets; I advise checking the continuity of each one thoroughly, as mistakes are difficult to rectify later. The remainder of the soldered components can be assembled in almost any order, but I've found that it pays to be systematic and to follow a list, checking off each component as it is soldered in.

All the usual checks should be carried out before the ICs are inserted into their sockets. Particular attention should be given to avoiding solder bridges in the daisy-chained RAM area of the board.

It is useful to insert the chips in three stages and perform some functional checks on the system at each stage. The first of these stages is to insert the PROM and all the TTL, with the exception of the 74LS608 (IC16) and the 74LS245 data bus buffer (IC14). Now powering up the board on the bus can be performed with all the Tanex RAM and EPROM still resident without the risk of any memory conflict occurring. This procedure will allow you to check the following items with the system running.

A dual beam oscilloscope is really desirable particularly if you have deviated from the timing component values for any reason. However it should be possible if you don't have access to a 'scope to use a good logic probe to check that all the apropriate signals are present.

The most relevant signals to check first are $\phi 1$, $\phi 2$ and their complements $\overline{\phi 1}$, $\overline{\phi 2}$. RE and $\overline{\text{DBE}}$ should be active only when a valid address within your programmed memory map is accessed. Next check that the two address buffers, IC2 and IC11, are switching correct-

SYSTEM ADDRESS						HEX ADD.	COMMENTS
A15	A14	A13	A12	A11	A10		
1	1	1	1	1	1	FFFF	TANBUG
1	1	1	1	1	0	F800	
1	0	1	1	1	1	BFFF	I/O
1	0	1	1	1	1	BC00	
0	0	0	0	0	0	03FF	CPU BOARD RAM
0	0	0	0	0	0	0000	

Table 1 Revised system memory map.

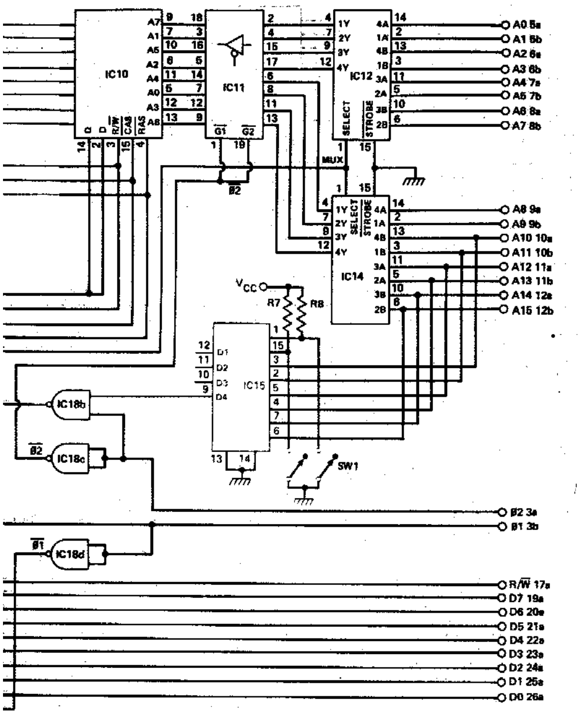


Table 2 Programming sheet for the PROM.

64K DYNAMIC RAM BOARD PROM PROGRAMMING SHEET . . . OF . . . 4									
SYSTEM ADDRESS	SW1	SW2	11	10	13	14	15	12	HEX PROM ADD.
PROM ADDRESS	8	7	6	5	4	3	2	1	BLOW TO 07
	0	0	0	0	0	0	0	0	00
	0	0	1	1	1	1	1	1	3F
	0	0	1	0	1	1	1	1	2F
	0	0	1	1	1	0	1	1	3B

ly, ie exactly 180° out of phase with each other, and that the refresh address counter IC1 is functioning correctly as an eight-bit counter.

The final check at this stage is to measure the pulse delay and shaper circuits formed by the diode/resistor networks and IC17. The three signals $\overline{RAS}$ cycle start (RCS), memory cycle start (MCS) and refresh (RFSH) should all correspond to the timing diagram in Figure 3. Any deviation should be adjusted by altering the value of the capacitor and/or resistor within the relevant RC network. However if the stated tolerances of the components are adhered to there should be no problems.

Having completed all the checks and adjustments so far the next stage is to insert the 74LS608 memory cycle controller, which should produce the necessary signals $\overline{RAS}$, $\overline{CAS}$, MUX and R/$\overline{W}$. These four main signals should be checked against the timing diagrams in Figures 2 and 3. The important factors are the relationships between cycle start, CS, and $\overline{RAS}$, MUX, $\overline{CAS}$ sequence and the $\overline{RAS}$ refresh cycle. The row address hold time RAH, CAS low and the precharge time are the major controlling times and are all programmable via the three RC networks on the 74LS608. Under standard conditions with the 750KHz Microtan system clock these times have quite a large latitude. However with faster clock rates the times become proportionally more critical. All these times can be calculated from the memory data sheets.

One fault which may occur at this point has the symptoms $\overline{RAS}$ permanently low, $\overline{CAS}$, MUX and R/$\overline{W}$ permanently high. If this situation exists try shorting **very briefly** pin 12 to ground. If the controller then starts to function correctly then the 74LS608 is at fault. I understand from Texas that on a number of the older batches of chips there is a fault with the power-on-reset circuit, newer batches, I am assured, are all O.K.

Having checked that all the relevant signals are present at the RAM chip sockets, the RAM chips themselves can now be inserted. **Power down first.** These are very static sensitive so take all the usual precautions, they are also **upside down** in relation to the other ICs on the board. Be warned that if they

HOW IT WORKS

As so much detail has been given in the general section, this 'How It Works' is going to be fairly brief. During $\phi 1$ high the main bus buffers IC11 and IC13 are disabled, removing the RAM from the system bus. The refresh row address counter IC1 is connected directly to the RAM ICs (IC3-10) via the enabled buffer IC2. The rising edge of $\phi 1$ is first buffered by two OR gates and then, via the pulse generator network D1, C1, R1, IC17, it applies a pulse to the REFRESH ENABLE pin (14) of the memory cycle controller IC16. The same rising edge is delayed by D2, C2, R2, IC17, before reaching the CYCLE START pin 13 of IC16. This delay is necessary to satisfy the refresh hold time of the memory cycle controller, and must be maintained at 20 ns minimum. The MCC then responds by pulsing $\overline{RAS}$ low for a period of time determined by the RC network at pin 12, the row address hold time. The rising edge of $\overline{RAS}$ is the end of the refresh cycle.

The memory cycle starts with the rising edge of $\phi 2$ (falling edge of $\phi 1$) at which point the address bus buffer is enabled directly by $\phi 2$ and assuming the address is within the memory map, the PROM output D4 is already high. This output combined with $\phi 2$ produces via IC18 two signals $\overline{DBE}$ and RE.

$\overline{DBE}$ enables the data buffer IC13; RE delayed via D3, C3, R3, IC17 is fed to the CYCLE START input of IC16 the memory cycle controller. This last event causes the MCC to start the actual memory access cycle. The $\overline{RAS}$ output (pin 7) goes low then, after the programmed RAH time, the R/$\overline{W}$ line is allowed to pass through and the MUX output then goes low switching over the address multiplexers IC12 and IC14 to the column address. $\overline{CAS}$ then goes low for a period of time $\overline{CAS}$ LO. All three outputs $\overline{RAS}$, $\overline{CAS}$, MUX then go high. This point should coincide with the falling edge of $\phi 2$ when the data from or to the RAM is latched by either the CPU or the memory depending on the status of the R/$\overline{W}$ line.

The next refresh cycle then occurs on the rising edge of $\phi 1$ and so the system carries on until the power is removed.

are inserted with pin 1 to the upper edge of the board they will be irrevocably damaged, and at £4 each a mistake could be very expensive. Finally insert the data buffer IC13.

With construction and testing completed there is still one task to finish before the board is inserted back into the rack and powered up. **Remove all Tanex RAM and EPROM**, and all other memory map conflicts, for example the hires graphics board, failure to do this will probably destroy ALL the memory components in the system.

After powering up the board in the now "minimised" system, unless you've chosen to create a memory map option which retains the Tanex EPROM, your system will be running in Tanbug or TUG bug. The quickest way to check the RAM from here is to boot up Basic and XBUG from disc or ESC and let it do the check. 47103 BYTES FREE should appear as the message header. Note some difficulties may be experienced because the F7F7 error jump will not exist immediately. This will show up only if an error occurs during the boot up procedure e.g. miss keying; simply RESET and start again to recover. Assuming this initial check appears to be OK then a more comprehensive memory test routine should be performed; the one published in the November 1981 issue of Computing Today is most suitable. However it should be noted that these types of test do not pick out the periodic bit drop out and only extensive usage in BASIC or similar will show up this problem.

Other Systems

The board relies only upon signals derived directly from the CPU i.e. $\phi 1$, $\phi 2$, $R/\overline{W}$ and the address and data buses. Since all these signals will be present in any 6502-6800 system, conversion is relatively simple. The only component that needs to be altered In any way is the PROM which does all the decoding. The essential considerations are those concerning the design of the memory map and, in particular, possible address conflicts. Remember no two components, be they RAM or I/O should have the same address! A suggestion for those with a Microtan but no discs or ESC is to leave the XBUG EPROM resident (F000-F7FF) and use the tape routines instead.

Those who design their own PCB should take care to heed the memory manufacturer's recommendations on decoupling and PCB layout around those chips. Particular attention should be given to the ground and power supply lines, which effectively surround each chip; the arrangement of interlocked fingers on the typical breadboard is definitely out.

Similarly the decoupling of the TTL chips should be comprehensive enough to avoid too much power supply noise, a major culprit of periodic bit drop out. Lastly, the 74LS608 MCC gets hot, but since the lead-frame is directly coupled with both the substrate and the ground pin, a large area of copper around pin 8 should alleviate the problem and improve reliability.

With regards to systems employing faster clock rates than 1MHz, as long as the $\overline{RAH}$, PRECHARGE $\overline{CAS}$ low times and the refresh hold time for the MCC are satisfied (calculating them from the manufacturer's data sheets), no significant problems should occur. **ETI**