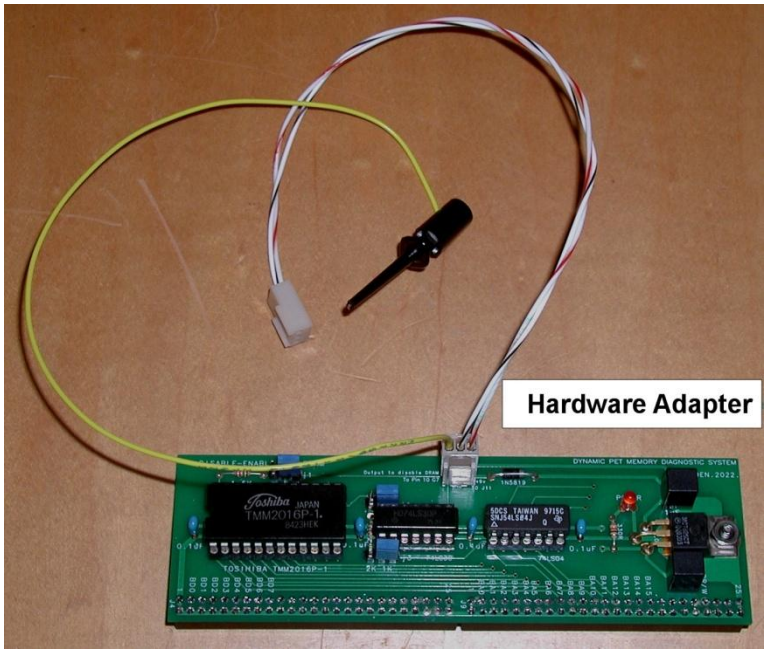
THE 2k SRAM DIAGNOSTIC PET:

DRAM MEMORY TEST SYSTEM FOR THE DYNAMIC PET COMPUTER.

H. Holden, August, 2022.



Introduction:

This article relates to 2001 model Dynamic PET computer, which use the 4116 DRAM Memory IC's. In the case of the 32K PET, sixteen 16384 x 1 bit 4116 IC's (or their equivalents) are used.

This article describes how to fault find and identify defective 4116's IC's in any of the 8K, 16K or 32K versions, specifically in the case where the IC's are soldered into the pcb and are not socketed.

Generally, the Dynamic PET computers were supplied with 8k, 16k or 32k of DRAM. In the 8K case it has been said that these IC's were actually 4116 IC's where ½ of the IC was defective internally and not used and only 1k of the capacity deployed. In most cases the Dynamic PETs have been upgraded from the 8K and 16K machine to the 32k option, using a total of sixteen 4116 DRAM IC's.

Although, sometimes, Commodore made the 16K to 32K upgrade very awkward by unforgivably drilling holes in the motherboard where the additional eight 4116 DRAM IC's would be fitted.

The 4116 DRAM IC's & the PET:

The Mostek 4116 DRAM's operate on the principle of charge stored on a capacitor. This charge represents the state of the "cell" or individual logic storage element. There are 16384 storage cells in the 4116. To address these might take 14 bits or 14 address pins (2^14= 16384), however, to keep the package size low and pin count down, 7 pins are used. The row and column addresses are generated by IC's E3,E4,E5 & E6 (74153's) in the PET and latched inside the 4116 IC's. The first control signal is the Row address strobe or /RAS signal, this latches the 7 bit row address into the 4116.

The second strobe is the Column address or /CAS and this latches the column address into the 4116, creating the unique address of the memory cell. Inside the IC a sequence of events is triggered using clocks that are internal to the IC and timed so that the input multiplexing is not disrupted. The 4116 is a complex IC.

Reading the individual cell depletes the stored charge and the stored charge also dissipates over time (some minutes due to electrical leakage). Therefore the logic state of the cell must be continually refreshed. The 4116 data sheet says that refreshing is accomplished by performing a memory cycle at each of the 128 row addresses within a 2mS time interval. And that a normal memory cycle will complete a refresh function too, though the refresh is generally achieved with the /RAS only signal.

Data to be **written** into the 4116 is strobed from the data bus via IC I11 & IC I10 (74LS244's) to pin 2's of the bank of 4116's. The data there is latched into an on chip register by a combination of /WRITE & /CAS while /RAS is active.

In the case of the PET design, the 4116's output data pin (pin 14) and input data pin (pin 2) are connected together. Commodore appear to have used the "early write cycle" method to use the 4116. This option references the input Data timing to /CAS by bringing the /WRITE input low prior to /CAS going low. This timing was confirmed checking with the scope.

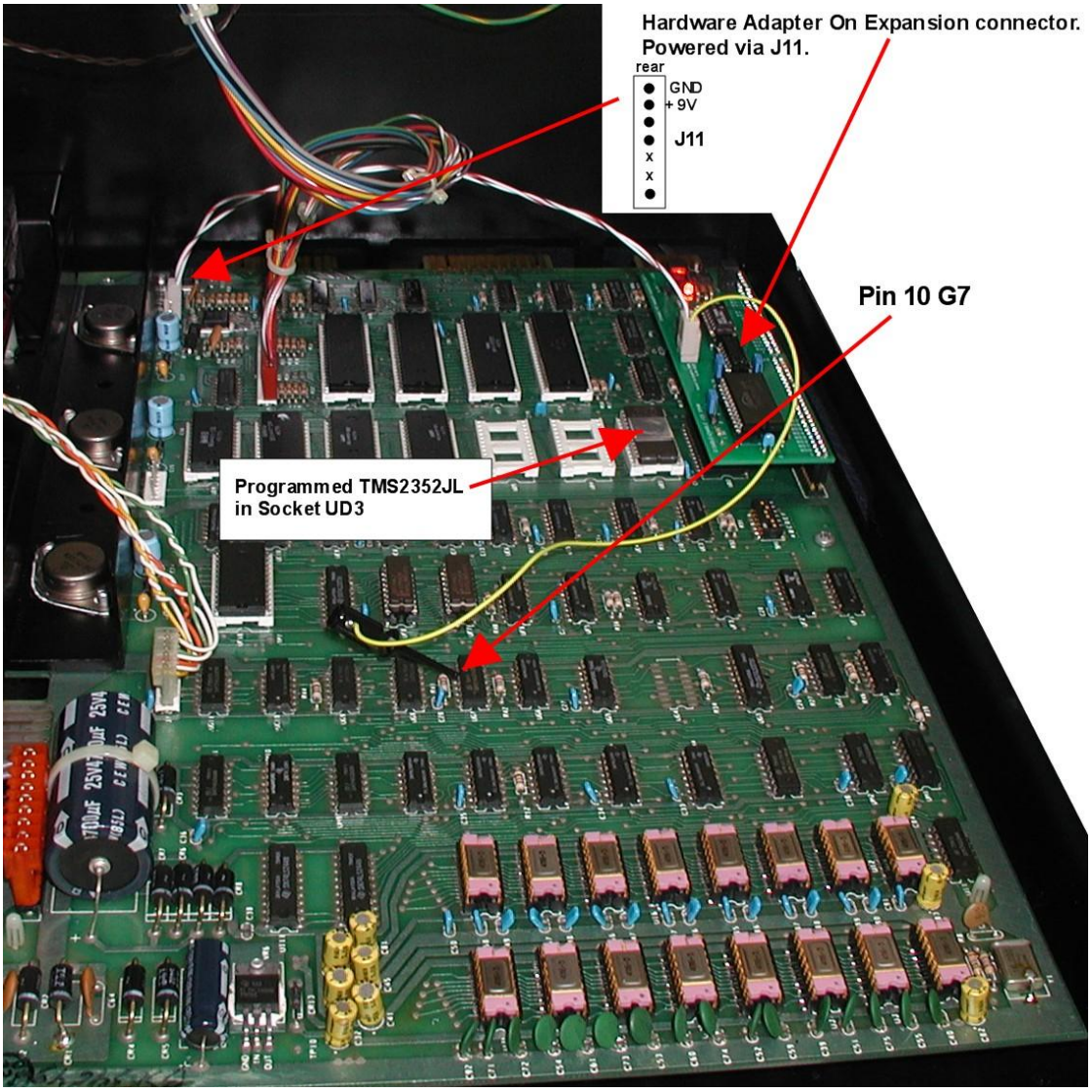
Data is **read** from the memory by keeping /WRITE high throughout the portion of the memory cycle when /CAS is low and around the time /CAS is low the output data present on pin 14 of the 4116 IC's is strobed onto the data bus by the same pair of 74LS244's which handle the writes.

The output of the IC internally is via a tri-state gate. This can only assume a high, low (depending on the logic state of the cell) or be effectively open circuit. In this latter state though, as will be shown later, in the event of faulty IC's this open circuit voltage level is borderline for a logic low or high and can be interpreted as either at different times when that level is strobed onto the PET's data bus. This has some interesting implications in determining which IC's are faulty on testing.

Diagnostic System Background:

The diagnostic system presented here, for the PET DRAM analysis, leaves no stone unturned in locating defective DRAM IC's. Also it allows detection of faults in the DRAM's support circuity.

This system uses a combination of Firmware in a 2532 ROM placed in socket UD3, containing four programs and uses a Hardware Adapter device which plugs on to the computer's expansion connector:



If it appears the DRAM is malfunctioning, it could be assumed that the DRAM IC's are faulty, when in fact it is not the actual DRAM IC's at fault. Unlike SRAM, DRAM requires

a good number of auxiliary IC's to address it correctly and make it function normally. DRAM requires constant refreshing.

SRAM, on the other hand, does not require the support circuitry and holds it data as long as it remains powered.

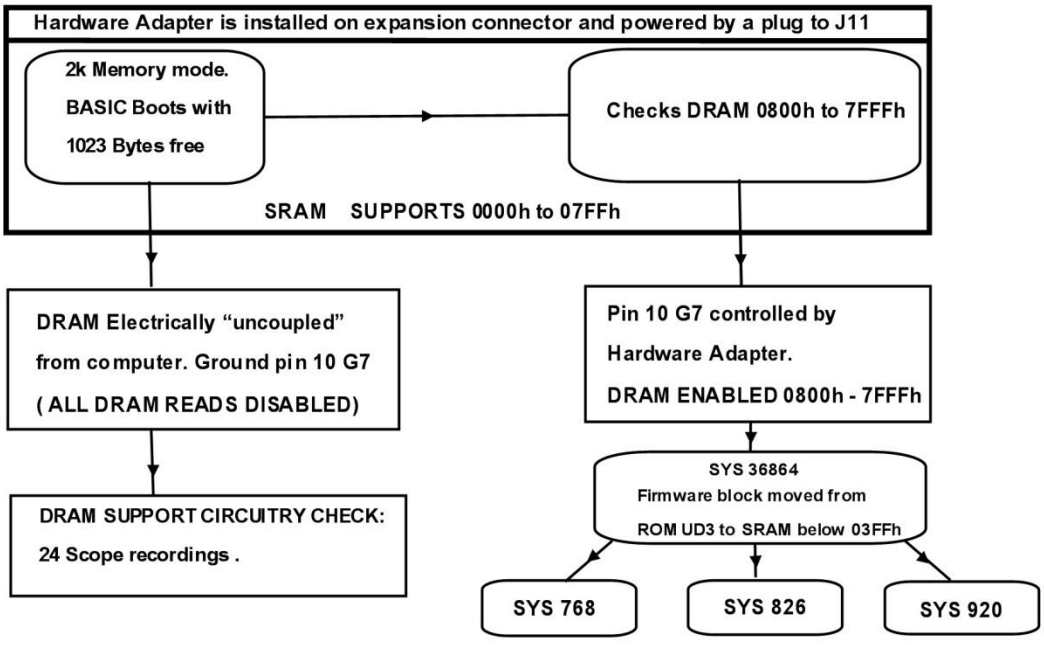
This means that to adequately diagnose and repair faulty DRAM completely requires that the functions of the support circuitry and DRAM are split into two separate entities for testing.

The Hardware Adapter & Firmware Diagnostic System:

The block diagram below summarises the basic features of the diagnostic system presented here.

Essentially one feature allows checking of the support circuitry, purely based on a hardware approach. The other feature is based on firmware to check the DRAM IC's themselves, with the goal of pinpointing which particular DRAM IC's are defective.

DRAM Memory & DRAM Support Circuitry: Test system overview:



Firmware utilitilies for detailed testing of DRAM Final tests done in 1K mode

Test Sequence:

It is preferable to check the DRAM support circuitry first by totally disabling the DRAM IC's.

This is achieved by simply linking pin 10 of IC G7 to ground with a wire, with a clip on each end suited to grabbing an IC pin. Then with the hardware adapter system running on the expansion connector, **set for 2k mode**, creates an operational "2k PET" supported only by SRAM. All of the important signals for "driving the DRAM" are then checked against the 24 scope recordings contained in this document.

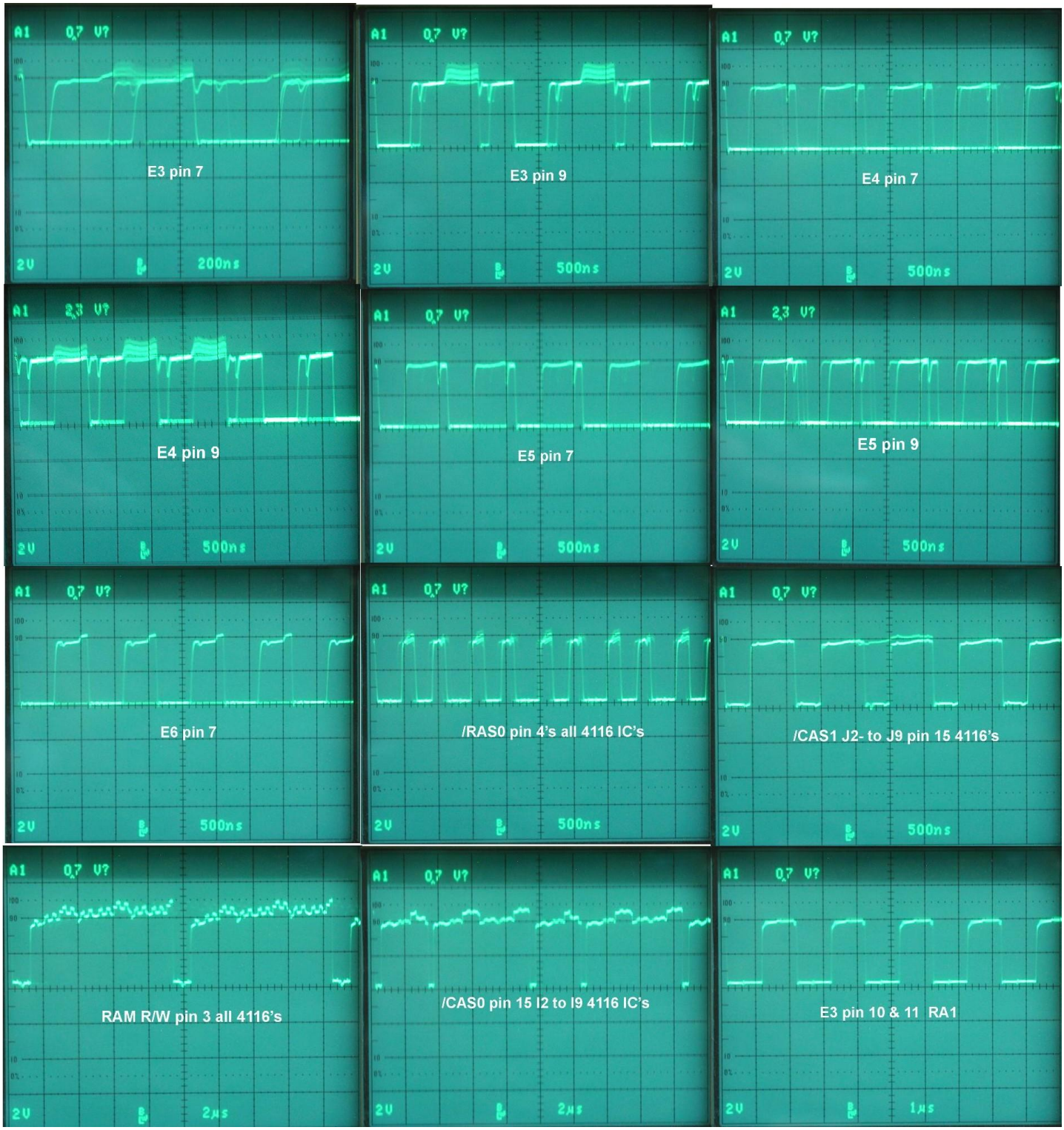
When the computer boots in this mode the following screen is reported.

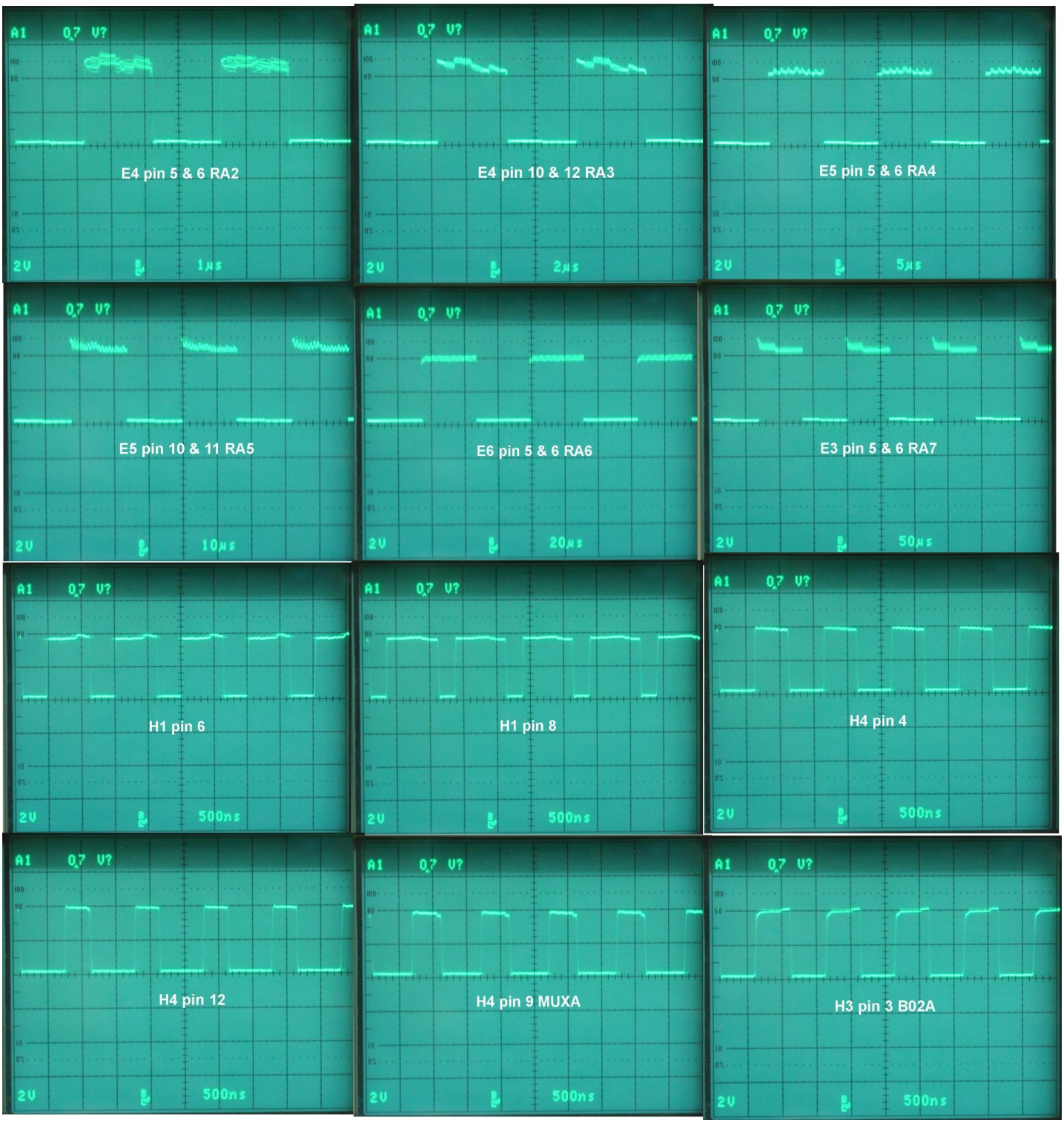
Of note, if the machine was booted with the two select jumpers on the hardware adapter accidentally set to 1k and Pin 10 of IC G7 grounded (full DRAM diable), BASIC finds no memory at 0400h & above and reports: **? OUT OF MEMORY ERROR IN 255***. Sometimes in the 1k mode it can cause boot problems if the DRAM is badly damaged. So it is recommended that the hardware adapter is always run in 2k mode, except for final testing of the DRAM. This way a stable boot with 1023 Bytes free always occurs regardless of any severity of DRAM damage.*

Fortunately, with the computer booted with the total of 2k of SRAM memory (provided by a TMS2016 on the hardware adapter) all of the support circuitry for the DRAM remains continuously active. This is helpful for checking the logic pulses in the support circuitry. It is not necessary to run any specific program to check the DRAM's support circuitry. Depending on the exact trigger settings on the particular scope, the recordings may look a little different in some cases. It pays to have the time-base setting (noted on each recording) the same for each test point checked as shown in the recordings.

Set the scope for DC coupling and the vertical gain at 2V/cm. All pulses should measure as standard TTL logic level pulses with no anomalous logic levels. The 24 important test points are shown on the two pages below. In all cases with no signal (zero volts) the scope trace was in the middle of the graticule. Of note the DRAM address multiplexers

are IC's E3, E4, E5 & E6. These provide row and column addresses created by the combination of the main buffered CPU address lines and also the signals RA1 to RA7.





One final test point to be checked is the pin 2's of the IC's E3,E4,E5,E6 which is simply a 1Mhz logic level square wave labelled **CLK1** which originates from pin 7 of IC G5 and appears similar to H3 pin 3.

After the support circuitry is checked and repaired, if required, then the testing moves to using the ROM firmware.

Pin 10 of IC G7 is then un-grounded and instead connected to the output on the hardware adapter. And as noted the adapter is set for 2k mode. This adapter dynamically deactivates DRAM reads over the range 0000h to 07FFh and uses the SRAM to support the operation of BASIC.

The DRAM IC's are checked with Firmware Diagnostic Programs over the address range 0800h to 7FFFh (or less if there is less DRAM memory installed).

The programmed ROM is placed in socket UD3. Initially, to start the process, the firmware is block moved (by typing **SYS 36864**) into the active SRAM memory which is operating below page 4.

(The schematic of the hardware adapter and the pcb layout are shown at the end of this article)

The 1K Memory below $0400 (page 4) is used extensively by BASIC. Page 0 and page 1 are critical to the computer's BASIC operating system. Fortunately there is some RAM space in high page 3 area where the Cassette Data Buffer area is, to run machine code programs without upsetting BASIC in any way. This is the address area where the three programs **SYS 768, SYS 826** and **SYS 920** are moved to and run from. These programs must run in RAM, not ROM, because they use self modifying code.

Various DRAM Defects:

Stuck high, stuck low or open circuit or non responsive DRAM IC output pins:

It is worth mentioning this initially, before more details on DRAM testing. One thing to note is that due to the fact these are one bit memory IC's, then every DRAM IC, be it in the 8K,16K or 32K PET, can contribute to the value of the Byte retrieved from memory.

For example, if *any* of the DRAM IC's in the entire array have an output that is stuck low, or stuck high the computer cannot run and will boot to a black screen. (In this condition though, once the VDU has warmed up, switching the computer off & on will briefly show the scrambled screen, while the 555 reset circuit is still in its timing cycle).

On the other hand, if the DRAM IC has an open circuit or floating output pin 14, the effects then are different, depending on what bank, I or J, the defective IC is in and if that floating pin is "interpreted" as being high or low - both are possible.

Case 1: If an IC with this effective open circuit defect is in Row J (upper 16k), it will allow the computer to boot, because it won't normally* interfere with the Row I output on the same pin. In this case the boot message will report 15359 bytes free (in the 32k PET), almost as though the whole upper bank of 8 IC's is not present.

(* = in some cases it can interfere see Failure example 5 later)

And in this case, since BASIC is still operational, Peeks & Pokes could be used into the addresses $4000 and above to check the RAM there, but it is better to do this with the programs in firmware provided.

CASE 2: The defective IC is in the lower bank and the computer won't boot:

This is where the **Hardware Adapter** comes into play, because hardware adapter dynamically disables the PET's on-board DRAM below $0800 (running in 2k mode), and switches in, in its place, known good SRAM. If this allows a normal boot and rehabilitates the BASIC operating system, then obviously there is a DRAM defect in row I or a defective DRAM in Row J with a stuck high or stuck low output in the same column.

(The hardware adapter can also be set to disable and electrically substitute in for DRAM memory below $0400, by altering two jumpers on it to the 1K position and connecting its output to G7 pin 10. This has value in final testing).

In any case, the **Hardware Adapter** may or may not be required to initially get the computer running, depending on the extent of the defects in the 4116 DRAMs, if they don't happen to be in row I and in the case they were in row J, that they do not interfere with he operation of the row I DRAM IC's. It is better to run the adapter though in all cases in 2k mode until all the DRAM defects are resolved.

Once the BASIC operating system is running (Hardware adapter supporting it) then the firmware in the TMS2532JL ROM, in socket D3 can be deployed.

Using the firmware:

The ***Initialisation Program*** is deployed first with the command **SYS 36864** to move the three test programs to the cassette buffer area (currently supported by working SRAM).

Once this is done the three programs **SYS 768, SYS 826** and **SYS 920** can be deployed. After each of these is run, the Ready prompt & cursor reappears.

SYS 768: When run, this program fills the memory from 0400h to 7FFFh with the same byte. The default value when it is run is AA (170 decimal). This byte can be altered by

poking its decimal value into address 822 and running the program again. For example POKE 822,187 would fill the memory with byte BB or POKE 822,204 would fill the memory with CC etc.

(After much work on memory checking, I found that rather than writing a program that writes to DRAM memory and quickly reads memory back, reporting errors as specific address on the screen, a better result could be attained by using the M/L monitor to inspect the memory a time after it has been written. There are two main reasons: One is the M/L monitor works very quickly as a tool to "scan the memory" and visually see a geometric pattern defect in the bytes as they scroll by, especially when they are the same byte, or a pattern from a checkerboard of bytes 00's and FF's. It takes about 4 & 1/2 minutes to display the memory from 0400h to 7FFFh. It is also very easy to stop the process where defects are observed by pressing the run-stop button to see the faulty bytes and their addresses. But, secondly, the delay it takes between filling the entire memory with bytes, or a byte pattern, then inspecting it with the M/L monitor later, is very helpful in confirming that the DRAM memory refresh function is normal and they are not losing data over a time frame of some minutes at least)

Once the program is run, the memory contents are then inspected with the M/L monitor. This is invoked with **SYS 64785.**

Then typing **M** 0800 7FFF (ENTER)

The bytes scroll by for inspection.

SYS 826: This program fills the memory 0400h to 7FFFh with a pattern of bytes. Firstly it fills the memory with byte 00. Once that is done, it then fills odd locations with byte FF. So the pattern in memory becomes FF 00 FF 00 etc. There is a specific reason for this. The idea is to not only to check that the correct bytes returned from the locations, but that writing to adjacent memory locations does not corrupt the byte values previously written in nearby locations.

Again the memory contents are inspected with the M/L monitor.

SYS 920: This program is similar, but instead it initially fills the entire memory with byte FF. Once done it then overwrites the entire memory at odd locations with byte 00 ,to create an inverted checkerboard pattern compared to **SYS 826.** The byte pattern created is 00 FF 00 FF etc. Again this is inspected with the M/L monitor for defects.

Physically Locating the Defective DRAM IC's from defective byte values returned on memory testing:

Essentially there are four ways the DRAM IC's can misbehave, assuming faults in its support circuitry are excluded:

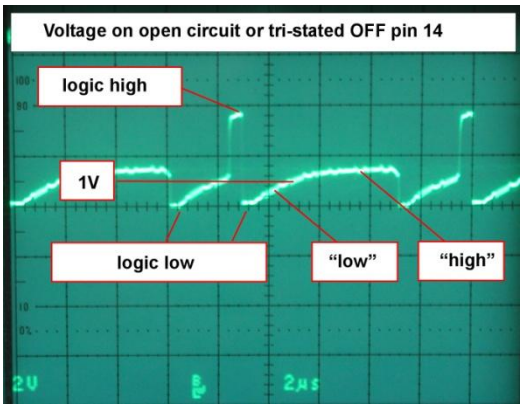
A) Because they have Tri-stated output (on the IC's pin 14), if faulty they can behave as though the chip is "not there", or in other words "non-responsive". In these cases piggybacking a good IC can work.

The defective IC in this case does not interfere with the function of the one in the other row that its output pin is connected to as a "general rule". However on testing actual faulty DRAM IC's I have been collecting, sometimes they can intermittently interfere with each other. (see example 5 later)

There is an interesting complication that one needs to be prepared for when interpreting the results of defective byte values retuned from memory testing:

It could be tempting to think that if an IC was either non responsive, or had an open circuit output pin (its tri-state output not working or effectively tri-stated off) then the logic state would be assumed to be high, because it is feeding a 74LS244 TTL buffer IC and normally the inputs of TTL IC's, when left open circuit, assume a high logic level. But, they don't over a short time frame of a few uS. It takes time for an input pin to charge to close to a high logic level, when left "floating".

Testing indicates that the voltage level seen at the 4116 IC's output pin 14, when it is tri-stated off, or open circuit, is a borderline logic level, depending on the timing. This is shown below:



Therefore, depending on the timing of a memory read, the output of the defective IC can either appear as a logic high, or a logic low.

So one needs to be “mentally prepared” for this variation when interpreting byte results returned from memory, while trying to figure out which IC is defective. If Commodore had used something like 10k pull up on the data outputs of the 4116 DRAM IC’s, this interesting ambiguity may have been avoided and all defective DRAM’s that were non responsive would have generated a consistent logic high.

B) The output can also be stuck logic high. This is a different scenario than above, because it interferes with the IC in the other row (same bit column) in every case.

C) The output can be stuck logic low, again interfering between DRAM IC’s rows I & J.

D) The IC output tri-state system may be working, but the data in the IC is corrupt due to various internal failures in the chip itself or refresh problems.

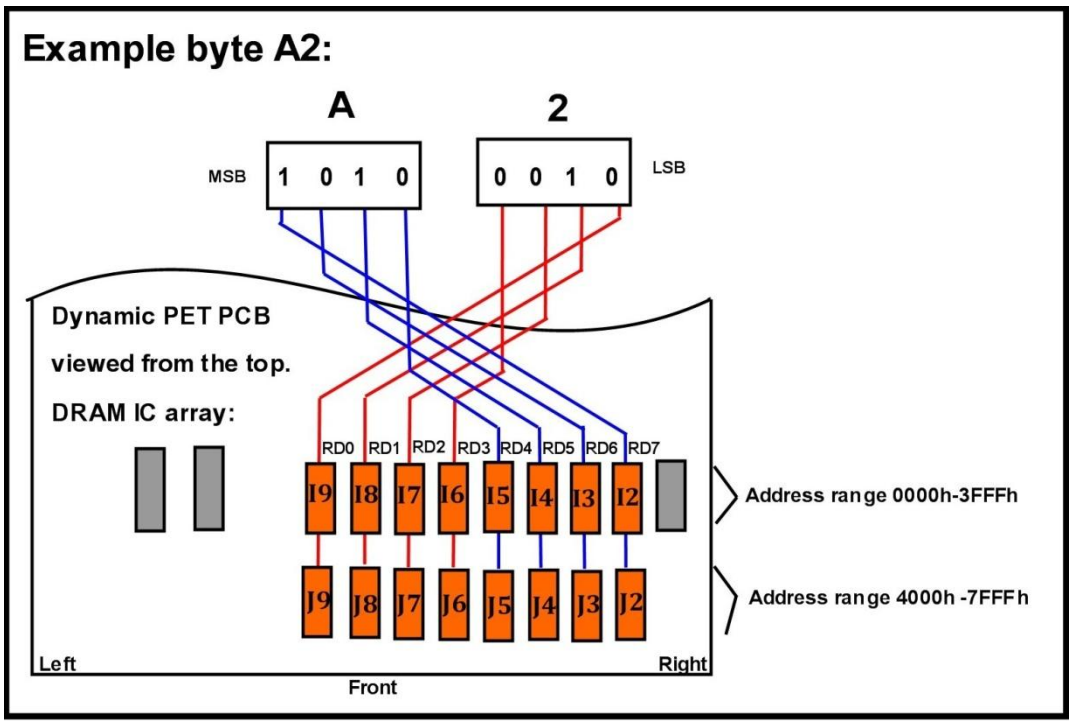
In addition, there could also be a problem with either IC’s I10, or I11 (the 74LS244’s), which strobe the DRAM data to & from the data bus BD0—BD7, however, in general these IC’s are more reliable IC’s than the 4116’s. And they can be checked on the scope to verify they are working properly.

Of course there can be more than one DRAM IC defective in row I or row J in the same or different column, so there can be a number of patterns of corrupt data to interpret. But in all cases it is possible to decode the returned bytes to determine which IC’s are defective at least down to two possibilities in the worst case. This is because the outputs of the IC’s in row I and row J in the same column have their pin 14’s directly connected to each other and one may develop a stuck high or low output pin.

A number of examples of DRAM failures will be given to get the idea of how to interpret the defective bytes returned from memory and find the IC’s responsible.

Examples of DRAM test scenarios:

When fault finding the DRAM, it pays to consult the schematic to see which IC pairs (in the 32k PET) are responsible for which bit of the Byte returned from memory. The physical layout of the DRAM’s on the actual pcb is in reverse order compared to the way the schematic was laid out which was more logical, so it helps to have the following diagram in mind:



Failure Example 1:

Initially the computer does not boot & has a black screen.

Running the hardware adapter in 2K mode and the adapter connected to pin 10 G7 the computer boots and reports: 1023 bytes free. Confirming there is a DRAM issue that was inhibiting booting BASIC.

This means the DRAM in row I is likely defective, or DRAM in row J is defective with a stuck output inhibiting row I, or combinations of both.

Firmware diagnostics run:

SYS 36864 deployed to initialise system.

SYS 768 deployed (default value AA written to memory)

SYS 64785 M/L monitor to inspect memory:

M 0800 7FFF shows all bytes are AE, not AA.

Implication: a value of 4 has been “added” to the lower nibble of the Byte.

SYS 826 deployed (values FF 00 FF etc written)

M 0800 7FFF shows byte array FF 04 FF 04 etc confirming the above test.

From the schematic RD2 is high (adding 4 to the lower nibble).

Analysis:

IC I7 pin14 is high, or pin14 of J7 is high. This can be determined which by examining the boundary across address 3FFFh.

If the memory dump shows that the bytes are normal above 3FFFh and defective below it, then it is possible to narrow it down exactly to a defective non responsive IC I7 with an open output pin 4 being interpreted as logic high. In this case a piggyback of a good IC would likely correct the problem.

On the other hand if the test had shown that the bytes were normal below 3FFFh and abnormal above, this would indicate that IC J7 was responsible, being unresponsive.

(if this was the case BASIC would have also reported 15359 Bytes free on booting because memory up to 3FFFh would be functional)

On the other hand (running out of hands) if the bytes are abnormal across the full range 0800h-7FFFh, then likely I7 or J7 have a stuck high pin 14. And it would be necessary to disconnect at least one of the pin 14's on I7 or J7 to find out.

In the case that a pair of IC's in the same column and row I & J has a stuck high, or low output pin, it can only be narrowed down to two IC's and piggybacking cannot help resolve the problem to less than one of two possible IC's being defective.

Failure Example 2:

Second example with another type of fault affecting UI7 or UJ7 and again:

Initially the computer does not boot & has a black screen.

Running the hardware adapter in 2K mode and the adapter connected to pin 10 G7 the computer boots and reports: 1023 bytes free. Confirming there is a DRAM issue.

This means the DRAM in row I is defective, or DRAM in row J is defective (with a stuck output inhibiting row I) or combinations of both.

Firmware diagnostics run:

SYS 36864 deployed to initialise system.

SYS 768 deployed (default value AA written to memory)

SYS 64785 M/L monitor to inspect memory.

M 0800 7FFF shows in this case all bytes are correct AA AA AA…

M/L monitor exited (type X) and **SYS 826** deployed. Re-enter the M/L monitor

M 0800 7FFF shows all bytes are FB 00 FB 00, not FF 00 FF 00 as they should be.

Implication: RD2 (bit 3) of the lower nibble is stuck low and decimal value 4 has been subtracted from the lower nibble.

Analysis:

Either I7 or J7 has stuck low pin 14. It did not show up on the test with byte AA, and it does not corrupt byte 00, because, in both cases bit 3 has a zero value for the bytes AA and 00 anyway.

It is a case where piggybacking a RAM IC would have failed to correct or detect the problem.

Failure Example 3:

The 32k computer initially boots with or without the hardware adapter running in the 2k mode, BASIC reports 15359 bytes free.

This suggests the memory above 3FFFh is not working, but the DRAM memory below 3FFFh is working.

This indicates that there is one or more defective (non responsive) IC's in Row J. It is not a stuck bit issue in row J, or row I's function would have been corrupted and BASIC would not have booted without the support of the hardware adapter. (It is better to run the hardware adapter in all cases though).

Firmware diagnostics run:

SYS 36864 deployed to initialise system.

SYS 768 deployed (default value AA written to memory)

SYS 64785 M/L monitor to inspect memory.

M 0800 7FFF shows all bytes 0800h to 3FFFh are normal AA, but at 4000h and above all bytes are EA.

Implication: a value of 4 has been added to the upper nibble of the byte. Suggesting RD6 (bit 7) of the 4116 IC location J3 is non responsive (open) and in this case is being interpreted as high. The reason the pin 14 IC J3 must be open or a non responsive IC

(not stuck high) is that if the latter were the case, it would have corrupted the function of IC I3 and BASIC would have booted to 1023 Bytes free with the hardware adapter, not 15359 bytes free which only happens if DRAM is working below 4000h. In this example, piggybacking J3 would have found the problem.

Intermediate summary:

So far we have learnt from this that row I DRAM IC failures, regardless of the failure mode (excluding late data loss) prevent the computer booting to BASIC and require the support of the hardware adapter to make the computer run and use the diagnostic programs.

On the other hand, row J failures may or may not prevent the computer booting, depending if there is a stuck bit issue (high or low). A non responsive DRAM IC issue, with what amounts to an open circuit IC output; the computer will boot normally with or without the hardware adapter and 15359 Bytes free are reported by BASIC. If any row J bits are stuck (just as any row I bits too), the computer will not boot without the hardware adapter.

A stuck pin or "stuck bit" issue is **not** amenable to piggybacking DRAM as a test. Also, with the stuck bit, it is indeterminate which of the two IC's with their pins 14 connected together in the same column of IC's is the cause and ultimately one pin 14 at least will need to be disconnected to find out.

However, in the non responsive IC cases, which appear common with faulty DRAM IC's I have tested, with an open output in a row J, or an open circuit output in a row I, it is possible to determine exactly which IC/s are the cause by looking at the bytes across the address boundary 3FFFh. In the row I case (unresponsive IC), the computer won't boot and requires the hardware adapter and BASIC reports 1023 bytes free. Memory examination, after running the diagnostic programs, shows that memory is defective up to the address 3FFFh, but normal above that, assuming there are normal DRAM IC's in row J.

In the row J case (unresponsive IC), the computer does boot even without the aid of the hardware adapter. But it is better to have the adapter in place. BASIC reports 15359 Bytes free. Memory examination, after running the diagnostic programs, shows that memory is normal up to the address 3FFFh, but abnormal above that.

In both of the above non responsive IC cases, one (or more of the bits) which make up the byte, can have assumed a logic high (or low) value as this is how an open pin is interpreted by the 74LS244 hardware interface to the bus. Unfortunately it is not high or low definitely as there are no pull up, or pull down resistors, and the output voltage at pin 14 can be a borderline logic level at the time of the Data read. This is shown in a real life situation in example 5 with defective IC's installed in my PET.

Failure Example 4:

A more complicated case: Two IC failures in row I and two in row J in different places (columns).

This example involves unresponsive IC's.

The computer won't boot. The hardware adapter allows it to boot to BASIC with 1023 bytes free.

Firmware diagnostics run:

SYS 36864 deployed to initialise system.

SYS 768 deployed (default value AA written to memory)

SYS 64785 M/L monitor to inspect memory.

M 0800 3FFF shows all bytes are corrupt reported as BE BE BE BE… not AA

M 4000 7FFF shows all bytes 22 22 22 22…not AA

M/L monitor exited (type X) and SYS 826 deployed. Re-enter the M/L monitor:

M 0800 3FFF shows all bytes are FF 14 FF 14… not FF 00 FF 00 as they should be.

M 4000 7FFF shows byte array is 77 00 77 00… not FF 00 FF 00 as they should be.

Analysis:

In the case of the byte AA, the upper nibble has become B. In the case of the byte 00 the upper nibble has become 1 in the address range 0800h to 3FFFh .

RD4 (bit 5) is high, suggesting initially pin 14 of I5 or J5 or both is possibly either open and interpreted high, or stuck high.

Above addresses 3FFFh, AA is reported as 22 and FF is reported as 77.

Since it is impossible to have a 22 reported from the row J IC's (which look after memory above addresses 3FFFh) then the pin 14 of J5 cannot be stuck high, or open and being interpreted as high.

Therefore RD4 (bit 5) being high, suggests pin 14 of I5 is open interpreted as high, or possibly stuck high. However, since byte also 22 appeared in the upper address range, pin 14 of I5 cannot be "stuck". Meaning that RD4 (bit 5) pin14 of I5 is high, due to its output being non responsive and interpreted as high.

Therefore First faulty IC discovered is I5.

Since in the upper address range 4000h to 7FFF (looked after by row J), FF is reported a 77, this suggests both bit 4 (RD3) and bit 8 (RD7) are low, being non responsive and interpreted as low. They cannot be stuck low, but only being interpreted as low, or it would have been impossible to have the byte pattern BE BE BE returned from DRAM.

This indicates that the outputs of IC's J6 and J2 are non responsive or open circuit.

Therefore the second and third faulty IC's are J6 and J2.

However, this does not account for the "E" in the byte BE over the address range 0800h to 3FFFh when it should have been A. This suggests that RD2 is high, but not stuck high or the returned bytes 22 and 00 would not have been possible. However, being open circuit (non responsive) and interpreted as high in this case.

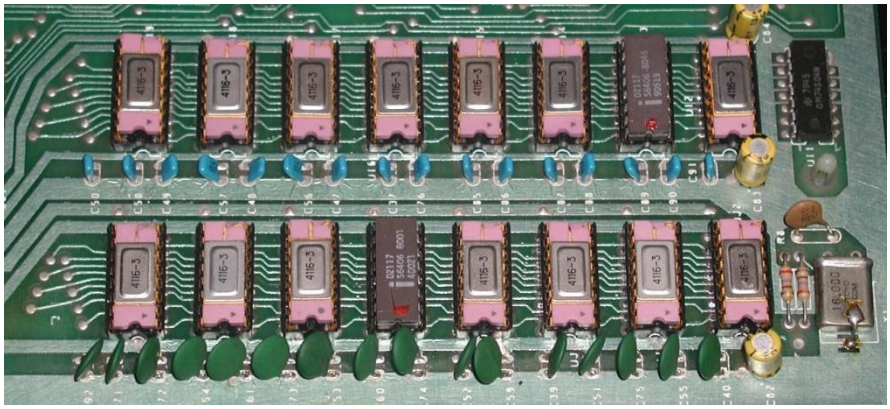
This implies that the fourth defective IC is I7.

Therefore the total number of defective DRAM IC's in this scenario are four, being:

J6,J2,I5 and **I7.**

Failure Example 5:

For this example two known defective 4116 DRAM IC's which failed in an independent 4116 DRAM tester are placed, one each bank as shown in the photo, to see what the testing system here makes of it:



Known faulty Intel 2117 IC's installed in position I3 and J6 (these are equivalent to 4116's. I mark all defective IC's with red paint)

Can we determine that these parts are faulty with the test system presented here?

Sequence:

Computer fails to boot, screen remains black.

Hardware adapter fitted in 2k mode:

Output connected to pin 10 G7.

Computer boots to BASIC showing 1023 bytes free.

Firmware diagnostics run:

SYS 36864 deployed to initialise system.

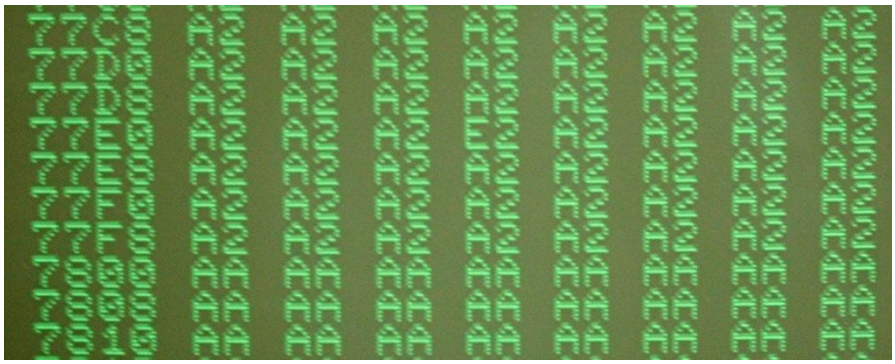
SYS 768 deployed (default value AA written to memory)

SYS 64785 M/L monitor to inspect memory.

M 0800 3FFF shows all bytes are corrupt reported as EA EA EA EA… not AA

M 4000 7FFF shows all bytes A2 A2 A2 A2…not AA

However in this upper address range, sometimes blocks of memory return AA correctly and the byte E2 *infrequently* appears too:

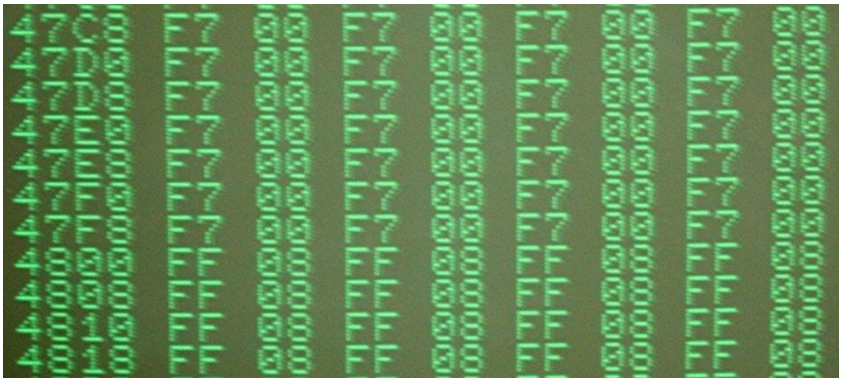


M/L monitor exited (type X) and **SYS 826** deployed. Re-enter the M/L monitor:

M 0800 3FFF shows all bytes are FF 40 FF 40… not FF 00 FF 00 as they should be.

M 4000 7FFF shows byte array is F7 00 F7 00… not FF 00 FF 00 as they should be.

However, from 3FFFh and above, it is mainly F7 00, but occasionally FF 08 FF 08 gets reported:



Analysis:

The fact that the computer recovered function with the hardware adapter indicates that DRAM is defective and very likely there is defective DRAM in row I.

Looking at the bytes returned from address range that row I supports it appears that a value of 4 decimal has been added to the upper nibble as EA is reported, not AA. This suggests that RD6 is high when it should not be. However, it is not stuck permanently high, otherwise it would have been impossible to display byte 00 which appeared on the **SYS 826** program test.

From this we can conclude that IC I3 is defective and non responsive possibly with a floating output that has been interpreted as logic high in this case. So we have already discovered **the first faulty IC I3.**

In the upper address range looked after by row J, the programmed FF is displayed as F7 at some addresses and normally at FF at others. And 00 is displayed as 08 at some address and normally at 00 in others. This suggests a decimal value of 8 is being added or subtracted from the byte value's lower nibble. This implies that RD3 is low sometimes when it should not be and high sometimes when it should not be. It indicates that IC in position J6 is defective and is unresponsive with a borderline logic output level, interpreted as sometimes high and sometimes low.

Now we have found the second defective DRAM IC **J6**.

What about the other "evidence" which might not seem at a glance to be a perfect fit:

Looking at the returned bytes from the upper address range 3FFFh to 7FFFh results:

AA is displayed as A2 or E2 at some infrequent addresses yet normally as AA at other addresses.

The A2 is accounted for again by the decimal value 8, corresponding to RD3 being low, when it should not be, pointing to IC J6.

But what about the intermittently occurring byte E2 in the upper address range when the E should have been an A? This suggests that a value of 4 has been added to the upper nibble of the byte by RD6, connected to IC's I3 and J3. And in theory, in this address range only IC J3 should be active, not the defective IC I3. The explanation here is that the output of the defective IC I3 is sometimes interfering with the output of IC J3 and mostly not at other times.

Summary:

We can learn from the above that the DRAM failure modes can be complex and even appear to be intermittent. However, close study of the returned defective bytes will lead us to the defective IC's. More frequently occurring byte errors have more significance finding the defective IC's than very infrequently occurring ones. Failed DRAM IC's can

intermittently interfere with the outputs of other IC’s they are connected to in the same column.

Wrapping it up:

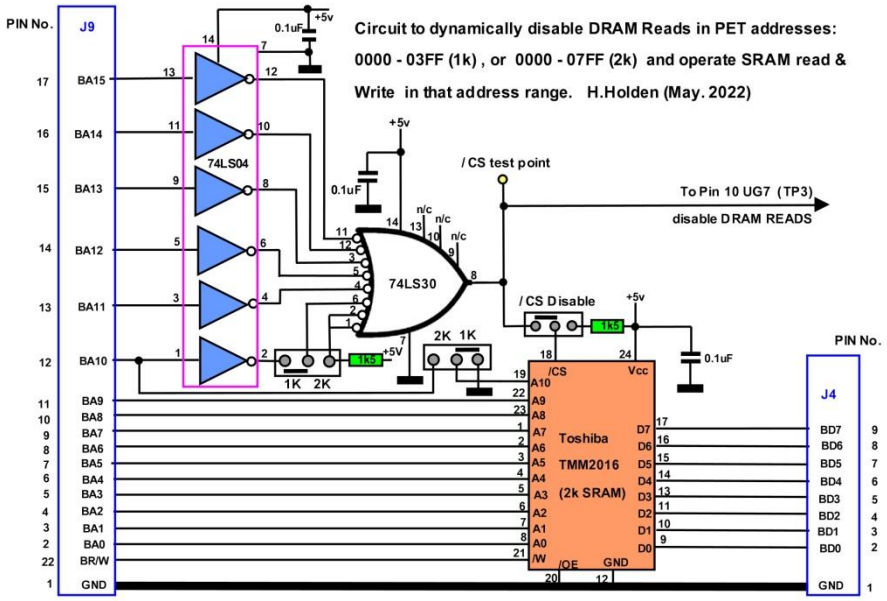
Finally when the DRAM system is repaired and running normally, the hardware adapter can be set for the 1k mode and its output to pin 10 G7. Then the three programs **SYS 768, SYS 826**, and **SYS 920** can be used for a full DRAM check from 0400h to 7FFFh.

Then finally the hardware adapter removed and the firmware run again for another check.

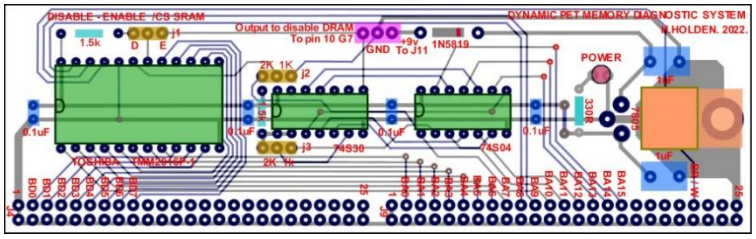
If there happened to be any unusual problems suspicious of DRAM faults, isolated to bytes only below 0400h, this would show up as the fault/s would clear when the hardware adapter was running and substituting in for DRAM below 0400h.

Since the hardware adapter provides 1023 Bytes of free memory and supports BASIC, small BASIC programs can also be run in the usual way above 0400h supported by SRAM.

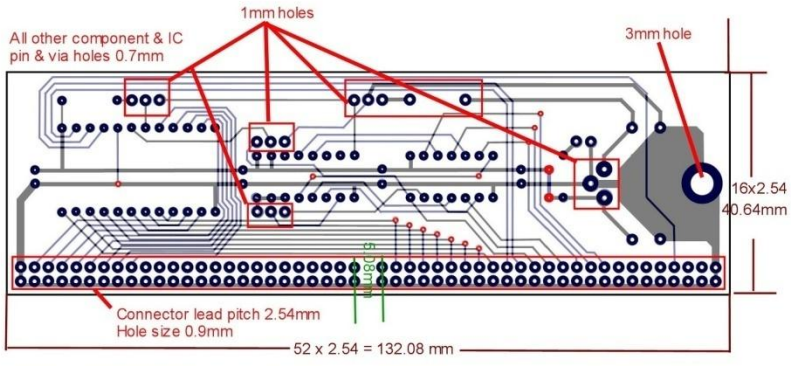
The hardware adapter schematic:

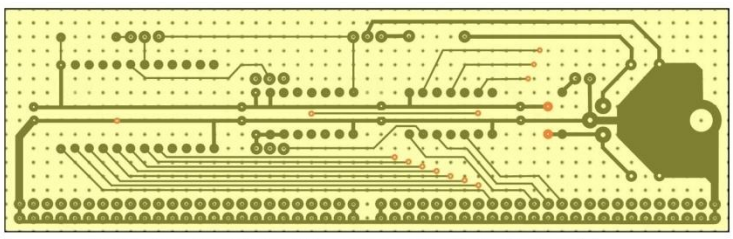


The SRAM can be jumpered with */*CS disable to deactivate it, if required, rating than having to unplug the adapter from the expansion connector to do it.



PCB size & HOLE SIZE CHART:





Most holes except most vias are on a 2.54 x 2.54 mm grid

PET DRAM MEMORY DIAGNOTIC (Firmware Held in TMS2532JL ROM in socket UD3 ,file = mmm.bin).

Step 1: Initialize	SYS 36864	Transfers memory diagnostic programs from ROM in socket D3 to cassette buffer area high page 3 (supported by SRAM on the Hardware adapter).
Memory Diagnostic 1	SYS 768	Fills entire memory $0400 to 7FFF with Byte AA. Or, you can modify the byte value by poking the decimal value to decimal addresss 822 and re-deploying the program.
Memory Diagnostic 2	SYS 826	Fills memory $0400 to $7FFF initially with byte 00 all locations. After that writes to every odd byte an FF. Byte pattern becomes FF 00 FF 00 FF 00 etc
Memory Diagnostic 3	SYS 920	Fills memory $0400 to $7FFF initially with byte FF all locations. After that writes to every odd byte an 00. Byte pattern becomes 00 FF 00 FF 00 FF etc.

Firmware placed in a TMS2532JL:

These Bytes start at address 0000 in ROM (block move program)ROM place in socket UD3.

08 48 8E 22 90 8C 23 90 A2 00 BD 00 93 9D 00 03

E8 8A C9 F6 F0 03 4C 0A 90 AC 23 90 AE 22 90 68

28 60 00 00

These bytes start at address 0300 and end at address 03F5 in the ROM. These are the three “SYS” programs.

08 48 8E 38 03 8C 39 03 A9 04 8D 37 03 A2 00 AD

36 03 9D 00 04 E8 D0 FA EE 37 03 AD 37 03 C9 80

F0 06 8D 14 03 4C 0F 03 A9 04 8D 14 03 AC 39 03

AE 38 03 68 28 60 AA 00 00 00 08 48 8E 95 03 8C

96 03 A9 00 8D 93 03 A9 04 8D 94 03 A2 00 AD 93

03 9D 00 04 C9 00 F0 01 E8 E8 D0 F5 EE 94 03 AD

94 03 C9 80 F0 06 8D 53 03 4C 4E 03 A9 04 8D 53

03 8D 94 03 A2 00 AD 93 03 C9 FF F0 08 A9 FF 8D

93 03 4C 4E 03 A9 04 8D 53 03 AC 96 03 AE 95 03

68 28 60 00 00 00 00 00 08 48 8E F3 03 8C F4 03

A9 FF 8D F1 03 A9 04 8D F2 03 A2 00 AD F1 03 9D

00 04 C9 FF F0 01 E8 E8 D0 F5 EE F2 03 AD F2 03

C9 80 F0 06 8D B1 03 4C AC 03 A9 04 8D B1 03 8D

F2 03 A2 00 AD F1 03 C9 00 F0 08 A9 00 8D F1 03

4C AC 03 A9 04 8D B1 03 AC F4 03 AE F3 03 68 28

60 00 00 00 00 00

The program bytes were placed starting at 0300h in the ROM file as it was less technically difficult to move them to start at 0300h in RAM and simplified the mover program. (All other bytes in the TMS2532JL ROM remain as FF)

****