Desmond’s C64/MAX RAM Test Thing

(DesTestMAX)

Matthew Desmond

factorofmatt.com

Revision 2

Table of Contents

Introduction

The DesTestMAX project aims to improve upon Commodore's Dead Test cartridge by using a more comprehensive memory test strategy and provide easier to interpret diagnostic output.

DesTestMAX utilizes the MAX (or ULTIMAX) cartridge mode as it allows us to place code directly in the $E000-$FFFF block of memory that gets executed as soon as the system resets. Such a situation allows the DesTestMAX code to take complete control of the system right from the get-go. Unlike the regular Commodore 64 boot process, DesTestMAX makes very few assumptions about the state of the machine and can produce usable results even with a severely broken machine.

The 6510 CPU, VIC-II and PLA need to be functional, though the ROMs, CIAs and SID are not required and can be removed if socketed. See the C64 Hardware Considerations section for more details.

The big drawback to MAX mode is that only the first 4K of system memory is available to the CPU. DesTestMAX then can't be a comprehensive memory test, but it can be a great help when faced with a Commodore 64 that shows nothing but a black screen.

A DesTestMAX cartridge will also work in a Commodore 128. DesTestMAX will automatically run in ‘64 mode yet requires the ‘128 kernal, a working Z80 and a working MMU. The VDC is not required.

Building a DesTestMAX Cartridge

The DesTestMAX code runs from a standard Commodore 64 cartridge configured in the following way:

- /GAME pulled low to enable MAX mode (/EXROM can be pulled high or left floating)
- 8K EPROM enabled by /ROMH ($E000-$FFFF)
- A reset button (if available) is highly recommended.

You can find myriad C64 cartridge PCB solutions and any of them should work for this purpose. In the simplest of cases, you just need to program and install an EPROM then configure a couple of jumpers and you're all set.

If you already have a Dead Test style cartridge it may be simple task of replacing the ROM/ERPOM with a new one containing the DesTestMAX code.

It is recommended that you first test your new DesTestMAX cartridge with a working C64 or C128 so you'll know what to expect when using it to test an unknown machine.

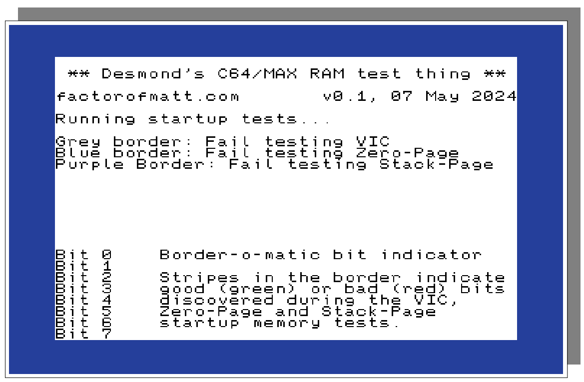
It is also possible to use the DesTestMAX code in place of the C64 kernal ROM. See the Kernal ROM section for details.

Running the Diagnostics

Exactly what you see when you power on your machine with a DesTestMAX cartridge installed depends on exactly what is wrong with the system. The DesTestMAX cartridge requires the CPU, VIC-II, PLA and supporting circuitry to be functional in order to be of any help. An entirely blank screen is a good indication that one of the big-three chips or their support logic is malfunctioning and the cartridge will be of little use.

Startup Tests

When the cartridge first starts it tests the VIC-II, Zero-Page, the Stack Page and the checksum of its own ROM. The startup screen will be displayed immediately upon cartridge startup and remains only for a few seconds while the startup tests run.

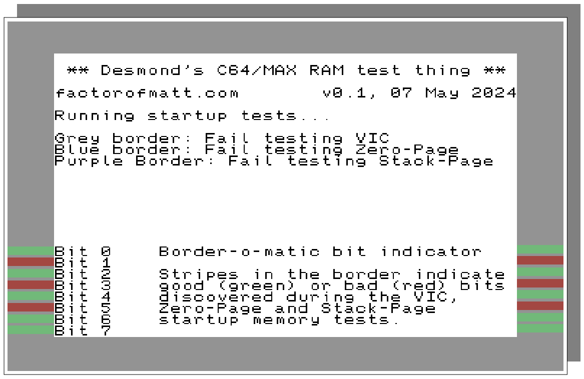


- The VIC-II has 47 registers mapped into the $D000 block of address space. We don't use all the features of the VIC-II during testing, so many of these registers can be considered as general read/write 'memory'. Testing is performed on these registers to verify that they can be written and read as expected.
- The Zero Page is one of the two 256-byte memory pages treated specially by the 6510. It would be impractical to write a full memory test without using at least some of zero-page, so we test it early here. The test does not use zero-page (or the stack) to do so.
- The Stack Page is the other 256-byte memory page treated specially by the 6510. The stack allows the use of subroutines (JSR/RTS). We test the stack page (without using it or zero-page) so that we have some confidence we can use subroutines for the more comprehensive tests.
- A 16-bit checksum is calculated for the entire contents of the DesTestMAX code ($E000-$FFFF). If the checksum is incorrect the EPROM image could be corrupt or could indicate that address decoding logic in the C64 is faulty.

An error detected in any of these first 4 tests will cease testing and the display updated to indicate the failure:

VIC Test Failure

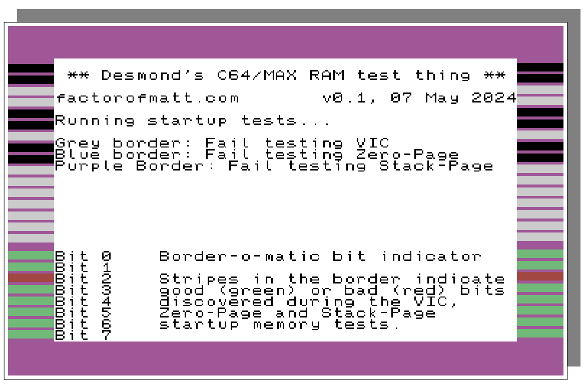
A failure during the VIC test results in a grey border with the failing bits displayed in the bottom part of the screen:



The border-stripes show data bits (D0-D7, top to bottom) that showed inconsistencies while testing the VIC-II registers. Here we see that data bits 1, 3 and 5 were detected as bad. This may be of diagnostic use if another data attached chip in the system is corrupting the bus – or it may simply indicate a marginal VIC-II.

Zero Page or Stack Page Failure

A failure during either of these tests results in a blue (zero-page) or purple (stack page) border plus stripes indicating the fail-address and failed bits:



The border stripes show address bits (A0-A15, top-to-bottom) and data bits (D0-D7, top to bottom). A black address bit is 1, light grey is 0. A green data bit indicates a good bit, red indicates bad. Here we see that data bit 2 was detected as bad and that the most recent memory location found to be bad was $01B3.

Code Checksum Failure

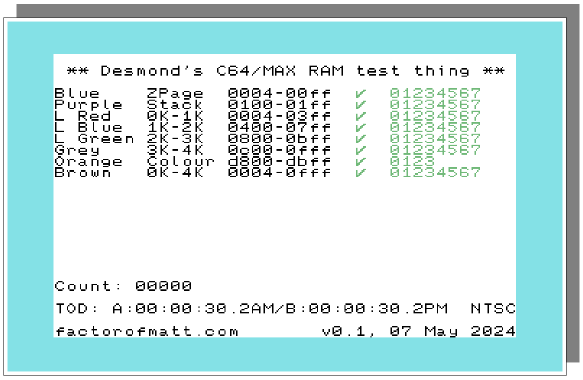
A checksum failure is indicated by a white border and a message indicating the checksum error. Under certain circumstances, the text of this screen may be garbled or otherwise unintelligible. See the Limitations section for details.

Main Tests

After the startup tests complete, we have enough confidence in the system to go ahead and test the entire 4K of memory. Depending on how much memory is good/present or if the VIC-II can "see" it properly this screen may be garbled. There are 8 tests here:

- Zero Page (blue border): Tested again
- Stack Page (purple border): Tested again
- 0K-1K (light red border) : first 1024 bytes (including the Zero and Stack pages)
- 1K-2K (light blue border) : second 1024 bytes
- 2K-3K (light green border): third 1024 bytes
- 3K-4K (dark grey border): last 1024 bytes. This VIC-II gets the display matrix from this region so there will be activity on the display during this test.
- Colour RAM (orange border): VIC-II colour information is stored in this region
- All RAM (brown border): All 4K is tested at once

It isn't strictly necessary to split the tests up as above, but it allows a little bit of visual flair for some tests and also means we can use a little pre-tested memory for storage of address/bit-error information.



At the bottom of the screen is the count of the number of times we've looped through all the tests. Each cycle takes less than 30 seconds. The TOD information (derived from the CIAs if they're present) is mostly for decoration though may indicate CIA issues if the times look crappy.

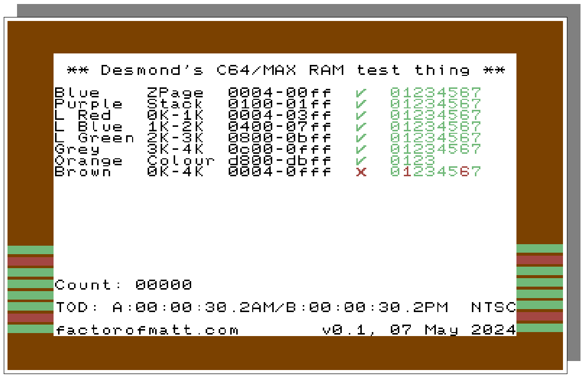
The tests will loop forever if no failures are determined.

Any test failure will halt testing, though tests in the current cycle (except possibly All RAM) are finished first: we might as well collect as much information as we can.

The Zero Page, Stack Page, 0K-1K and All RAM tests do not collect error information (since it would be overwritten by the destructive nature of the tests).

Error Detected, No information saved

If no error information was collected, the border stripes show data bits (D0-D7 top to bottom) green for good, red for bad.



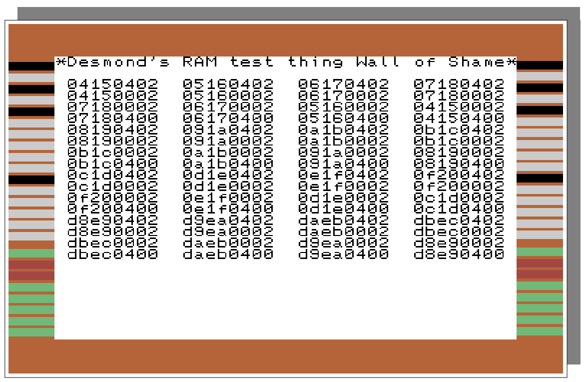
Error Detected, Information saved.

In the event that we were able to collect some address/bit-error information it is displayed on the following screen. Each error record is 8 hex digits wide:

- First 4 digits: The address at which an error was detected
- Next 2 digits: An indication of bits that were written as 0 but were read as 1
- Last 2 digits: An indication of bits that were written 1 but were read as 0

An error record of 04150402 indicates that at address $0415 bit 2 ($04) was mistakenly read as 1 and bit 1 ($02) was mistakenly read as 0. Patterns may emerge both in addresses and bad bits that can indicate the source of problems.

Note: some addresses may appear multiple times since the testing algorithm walks through memory both backwards and forwards a couple of times each.

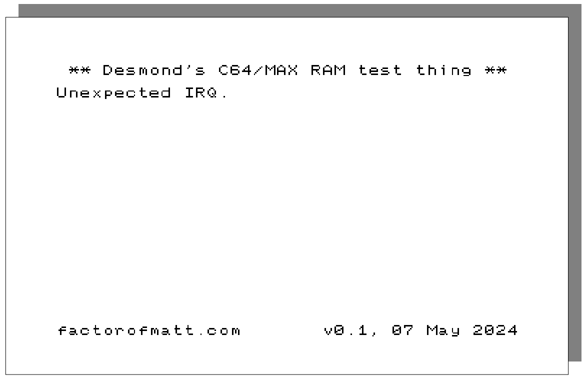


Interrupts

An unwanted interrupt could cause havoc during the middle of a test: interrupts write to the stack which really wouldn't be great if we're testing that memory or if that memory is suspect.

IRQ

Under normal circumstances no IRQs should occur during normal DesTestMAX operation. The CPU is left free to respond to IRQs and will display the following screen should any occur.



The receipt of an IRQ during the execution of tests could very well indicate something wrong with the interrupt-signal path or a misbehaving interrupt source (VIC-II, CIA etc).

NMI

NMIs cannot be disabled. The best we can hope is to ensure that any sources of an NMI are disabled. Unfortunately, the RESTORE key when tapped will always generate an NMI and there's really nothing that can be done about it. If an NMI is received, the following screen is displayed:

If you see this screen and did not hit RESTORE then something (CIA) generated an NMI or something is wrong in the NMI signal path.

If you hit the RESTORE key (possibly in frustration if nothing seems to be happening) and you see this screen, then at least you know the CPU works.

Kernal Rom

The code for DesTestMAX is located from $E000 to $FFFF in MAX mode. It turns out that the ROM kernal code resides there too. It is possible, then, to replace the standard Commodore 64 kernal ROM with an EPROM programmed with DesTestMAX. Breadbin units have a 2364 ROM containing the kernal so you'll likely require some kind of DIP Adapter. 64C units have a combined basic/kernal

ROM so you'll likely want a 16K EPROM/EEROM solution and program DesTestMAX into the top-half.

There are some differences when running code as a kernal instead of MAX mode:

- All 64K is available whereas MAX mode only has 4K.
- The VIC-II chip accesses RAM at $F000-$FFFF whereas MAX mode will access the ROM.
- The VIC-II requires CIA-2 to pick which 16K region of memory it accesses whereas MAX mode always accesses the last 4K in each region from the ROM effectively meaning there's no need for CIA-2.

This version of DesTestMAX does not account for the extra memory and has only minor contingencies built-in to ensure both MAX and kernal mode perform similarly.

While running as a kernal ROM, DesTestMAX is a little more dependent on system RAM to generate display output. Failed or marginal system RAM may produce garbled output. In such cases the border stripes will still relay failure information.

Limitations

While the VIC-II can be persuaded in MAX mode to display characters from the $F000-$FFFF ROM area, if we want to be able to update the screen to show useful information then it's going to have to come from the system RAM. If the RAM isn't functional in the $0800-$0fff region then the VIC-II display will be garbled. DesTestMAX uses this memory either way: it doesn't hurt and just maybe it'll produce something useful.

Memory Errors might not be due to the memory at all. The shared address and data busses in the Commodore 64 are susceptible to accidental hijack by malfunctioning ICs. If a chip writes data to one of these busses when it shouldn't then it can very easily seem like a RAM error when in reality the bus is being corrupted elsewhere. A useful technique can be to remove all non-essential chips in case they are dirtying a bus. The ROMs, CIAs and SID can all be removed (if socketed) and might provide clues if symptoms change after. DesTestMAX is happy to run without those chips. See the C64 Hardware Considerations section for more details.

Border-o-matic bit indicator

This silly name refers to the method used to display address and bit information should the VIC-II otherwise not be trusted to generate a useful text display. This simple technique has been used for other retro computer systems and it seems like a cool way to impart diagnostic information when all else fails.

The border is split into 24 'stripes' each that represents a bit:

- A0-A15 (top to bottom) – Address bits.

- Black for ‘1’, Grey for ‘0’.
- This number represents an address where the memory-test most recently found an error.
- In cases where no address information can be captured, these 16 stripes are absent.

- D0-D7 (top to bottom) – Data bits.
  - Green for ‘good’, Red for ‘bad’.
  - This value represents a map of RAM data-bits bits where an error was detected. Multiple bits can be flagged as ‘bad’ and indeed individual bad bits could come from different addresses.

The address shown (if present) will be that of the most-recent error found. This will usually correspond to the lowest faulty memory address of the region being scanned by nature of how the memory is tested. The good/bad bits do not represent any specific byte in memory, rather just an indication of bit-positions that showed an error at some address or other. Different revisions of the C64 use different configurations of physical RAM chips for storage. To map a bit number to a specific chip, See the C64 Hardware Considerations section.

If the border-stripes are shown then testing ceases so the information conveyed may be recorded. A power-cycle or reset is required to re-run the tests.

Testing methodology

The memory testing algorithm used in DesTestMAX is called March-B. A good description of common memory problems and test methodologies can be found here: <https://redirect.cs.umbc.edu/~reza2/courses/418/Slides/15MemoryTest.pdf>

The March B test performs 4 testing passes over the memory-region-under-test and ultimately verifies that any read or operation performed on a given bit is correct and doesn’t affect any other bits in the region. The test is order 17N meaning that each bit under test is written and read a total of 17 times during the test. The test of the entire 4K region available in MAX mode takes about 10 seconds.

Good care has been taken to ensure that no assumptions are made about the validity of memory before it has been tested. Neither Zero Page nor the Stack are used before those two memory regions have been verified since errors in either would cause havoc with the running code.

C64 Hardware Considerations

Commodore released multiple revisions of the C64 motherboard over the years. While these revisions remain mostly compatible with each other, there are a few differences that should be considered when attempting to diagnose a faulty machine.

RAM data bit to IC mapping

DesTestMAX helpfully indicates which RAM data-bits seem to be misbehaving but does not indicate the specific ICs that need to be replaced or investigated. This is for the simple reason that different revisions of the motherboard have different arrangements of RAM ICs and differing part identification numbers. This table will help you map bit numbers to specific ICs on your motherboard.

Assy# \ Bit	326298	KU1419HB	250407	250425	240441	250466	250469	C128
0	U21	U21	U21	U21	U21	U10	U10	U38
1	U9	U9	U9	U9	U9			U39
2	U22	U22	U22	U22	U22			U40
3	U10	U10	U10	U10	U10			U41
4	U23	U23	U23	U23	U23	U9	U11	U42
5	U11	U11	U11	U11	U11			U43
6	U24	U24	U24	U24	U24			U44
7	U12	U12	U12	U12	U12			U45

Shortboard CIA incompatibility

Elsewhere in this document we've discussed the fact that DesTesMAX doesn't require the CIAs to be installed in order to operate correctly. Unfortunately that isn't quite true for the most recent version of the C64 motherboard. The Assy 250469 "shortboard" as found in later C64Cs operates a little differently from the other motherboards when it comes to a missing CIA#2 ($DD00-$DDFF).

The PA0 and PA1 signals from CIA#2 control which of the 4 16K blocks of memory the VIC-II will address. The logic is inversed, so to select block 0 ($0000-$3FFF) both bits are set high.

For the first set of C64 motherboards, these two signals float high when the CIA isn't installed. [Though I don't see any specific pull-up resistors, my assumption is that the 74LS258 used to pick the correct 16K block floats its input pins to high internally if not driven to 0, as is the way with most TTL chips].

The shortboard motherboards don't use a 74LS258 to select the VIC-II block, rather it uses the 64-pin super-PLA for that job. For some reason, the PA0 signal no longer seems to float to high so the default block for the VIC-II is no longer 0 when CIA#2 is removed. The upshot is that the VIC-II looks at the wrong memory area during the latter stages of DesTestMAX and the screen is garbled. A 1K to 10K resistor placed between /VA14 (pin 2) and +5v (pin 20) on the CIA#2 (U2) socket should be enough to overcome this limitation in the short term.

From Matt

If you've ever used the Commodore Dead Test then chances are you'll find DesTestMAX a useful addition to your diagnostics arsenal. Please do give it a try. And tell me what you think.

Though I've worked hard to ensure that DesTestMAX will give reliable, accurate results under the widest set of circumstances I just haven't been able to physically test much more than removing chips and inducing incorrect behaviour with jumper wires.

I'd like to see how DesTestMAX works out there in the real world. I'd like to hear about your experiences:

- Does it work at all for you?
- Does it reliably show a specific, traceable fault?
- Does it give misleading or plainly incorrect results?
- Could it be made more useful?
- Does the font make your eyes hurt?

Please send feedback to destest@factorofmatt.com

Thank you, -M@