DesTestKernal

Matthew Desmond
factorofmatt.com

Revision 1

Table of Contents

Introduction

The DesTestKernal solution can test the entire 64K of available memory in your Commodore 64 using the same March-B algorithm as the DesTestMAX 4K testing cartridge.

While your Commodore 64 doesn't need to be fully working in order to run DesTestKernal, the CPU, VIC-II and the PLA must be functional. The BASIC ROM, character ROM, SID and CIA#1 (U1) are not required. Longboard C64s do not require CIA #2 (U2), whereas shortboard C64s do require it (see the C64 Hardware Considerations section for more details).

While a DesTestKernal ROM will work in a Commodore 128 (when replacing the combined C64 BASIC/kernal ROM) please not that the Z80 and MMU also need to be working to.

About a DesTestKernal PROM

As the name implies, the DesTestKernal code replaces that of the kernal. This generally means that the kernal ROM (U4 in all C64s and U32 in C128s) need to be replaced with a PROM that contains the DesTestKernal code.

De-soldering the original ROM and replacing it with a socket are beyond the scope of this manual though advice is not hard to find on the internet.

Ideally you'd be able to program the perfect PROM to fit in the kernal socket. Unfortunately the correct EPROM for the 8K kernal ROMs (2564) aren't particularly commonplace these days so some kind of adapter or alternate technology is required:

2364 to 2764 DIP adapters are available on the web and will allow 2764 compatible PROMs to replace the kernal ROM.

For the C128 and shortboard C64s a common 27128 compatible PROM is, happily, directly plug-in compatible with the original 23128

Additionally, kernal-switching solutions (that use a larger PROM than necessary to allow multiple 8K kernal images) can also be used. Just program DesTestKernal into one of the available 'slots' for kernal images.

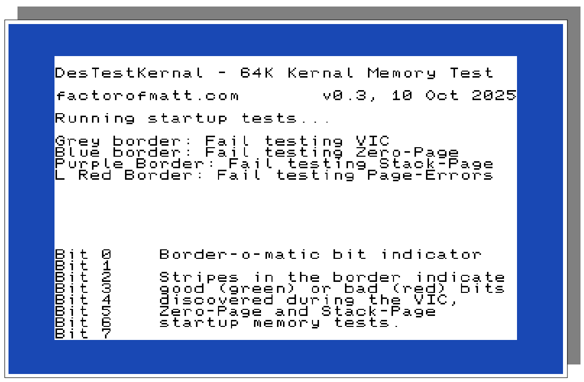
If you only have larger, pin-compatible, PROMs, you may need to program the device with 2, 4 or 8 duplicates of the 8K DesTestKernal binary image depending on the adapter solution you choose.

Running the Diagnostics

Exactly what you see when you power on your machine with a DesTestKernal cartridge installed depends on exactly what is wrong with the system. The DesTestKernal cartridge requires the CPU, VIC-II, PLA, the kernal ROM, a small amount of memory 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.



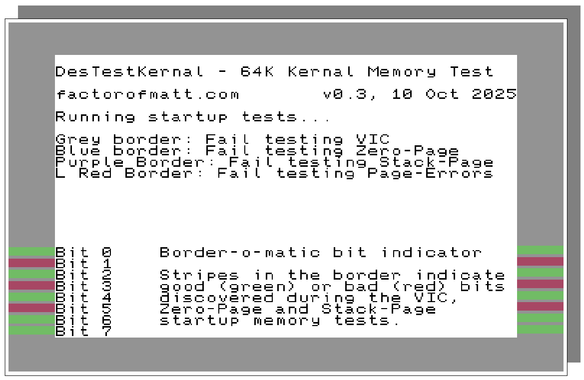
- The VIC-II has 47 registers mapped into the $D000 block of address space. We don't use all the features of the VIC-II during testing, so many of these registers can be considered as general read/write 'memory'. Testing is performed on these registers to verify that they can be written and read as expected.
- 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.
- Failures in the C64's memory address multiplexer circuitry can lead to seemingly random memory corruption known as Page Errors. Since we do use a little of the 64K memory during tests, we test the multiplexers early to avoid possible corruption.

- A 16-bit checksum is calculated for the entire contents of the DesTestKernal 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

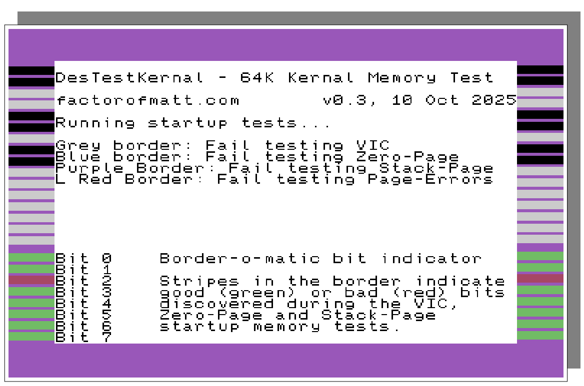
A failure during the VIC test results in a grey border with the failing bits displayed in the bottom part of the screen:



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

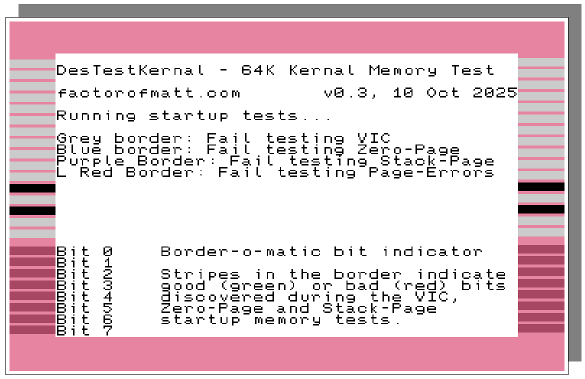
A failure during either of these tests results in a blue (zero-page) or purple (stack page) border plus stripes indicating the fail-address and failed bits:



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.

Page Error Failure

A failure during the page-error (multiplexer) tests results in a light-red (pink) border plus stripes indicating the fail-address-bits and failed bits:



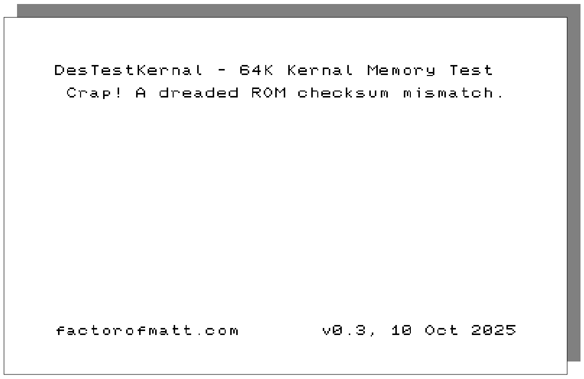
Page errors occur when some set of addresses gets incorrectly mapped to a different set of addresses. This can happen both inside a memory chip or in the case of a C64 when the address multiplexers fail. Imagine that address-bit 2 is always stuck ‘on’: a read or write at address 0 would actually be a read or write at address 4 (plus 1 maps to 5, 2 to 6 and 3 to 7 and so on every 4 bytes throughout memory). Similar situations exists if a bit is stuck ‘off’ or accidentally tied to another bit entirely.

The March-B tests used to detect memory errors are totally capable of detecting page errors, though we choose to explicitly check for them early to avoid accidental memory corruption. While every effort is made during the main set of tests to not actually use memory, a couple of addresses are used and they are at risk of corruption if the multiplexers are bad. This test hopefully eliminates the possibility of the multiplexers being bad and allows us to use some (previously tested) locations with a measure of confidence.

Unlike other tests, the fail-address-bits indicate which address-bits seem to be faulty (bits 9 and 11 in the above example) rather than a specific address. Such information can be useful (with help from the C64 schematics) to determine which multiplexer is likely faulty.

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.

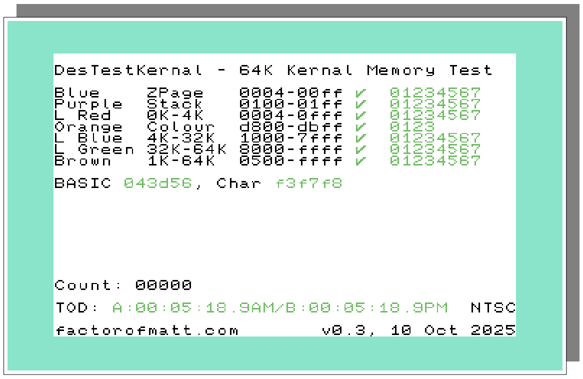


Main Tests

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-4K (light red border) : first 4096 bytes (including the Zero and Stack pages)
- Colour RAM (orange border): VIC-II colour information is stored in this region
- 4K-32K (light blue): the lower half of the C64 RAM.
- 32K-64K (light green): the upper half of the C64 RAM.
- 1K-64K (brown): the entire C64 RAM.

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 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).

Interference in the Border

As each of the memory tests is running you'll notice that the border flickers slightly (it may look like interference). This is entirely intentional and acts to inform you that the tests are still running.

ROM Checksum Display

After each test, the extended (24-bit) checksum of the BASIC and character ROMs are calculated and displayed. If the checksums match those of known 'good' ROMs, then the checksum is displayed in green. If the checksum is unknown, then it is displayed in RED. Should a bad-checksum seem to change after each test, then that is a good indication that the ROM in question is having electrical issues.

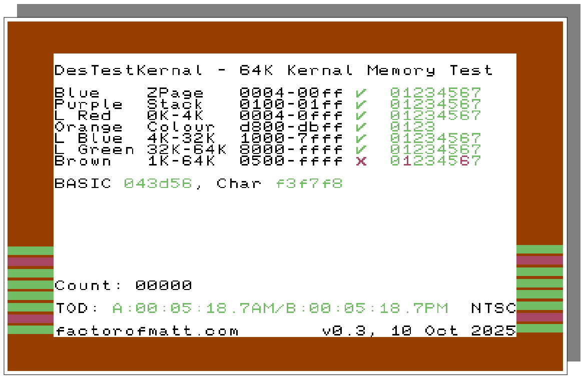
Note: a red checksum doesn't necessarily indicate a bad ROM, just that DesTestKernal doesn't recognise it.

CIA Time Of Day Display

The TOD information (derived from the CIAs if they're present) shows how long the tests have been running. Throughout the tests the times displayed should remain consistent between the two CIAs. Should the times not be consistent (ignoring AM/PM) then the times will be displayed in red instead of the usual green. Red times may indicate faulty or missing CIAs.

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.



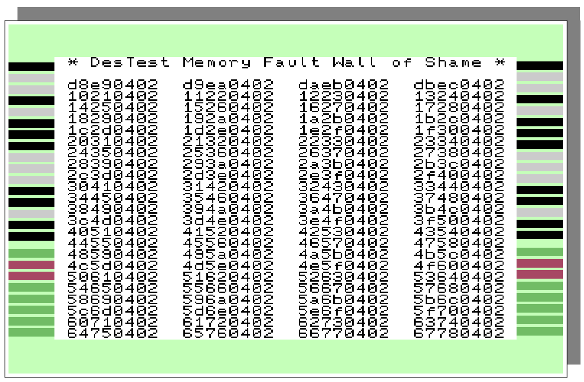
Error Detected, Information saved.

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.

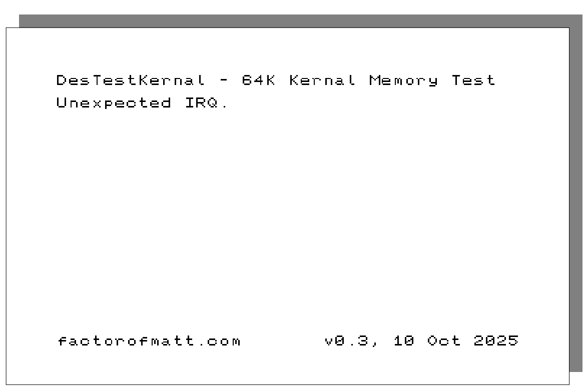


Interrupts

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

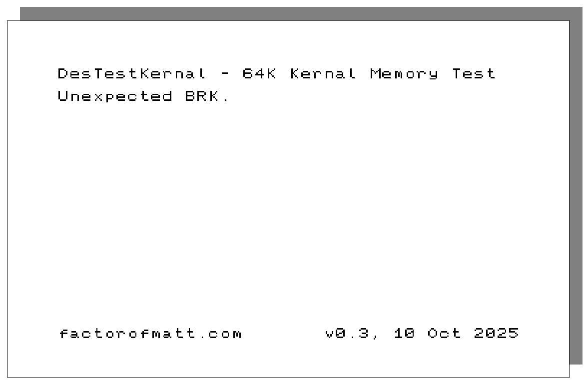
Under normal circumstances no IRQs should occur during normal DesTestMAX operation. The CPU is left free to respond to IRQs and will display the following screen should any occur.



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.).

BRK

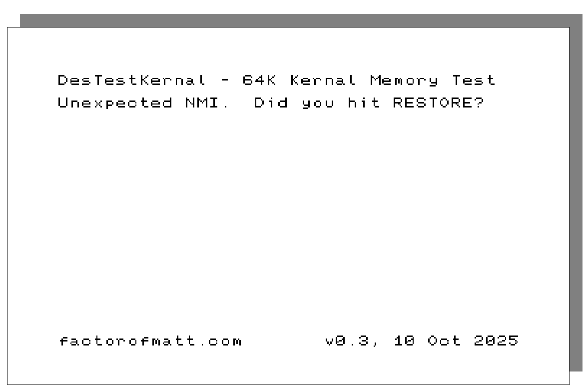
The BRK instruction of the 6502/6510 will interrupt the CPU in much the same way that an IRQ does – though it is straightforward to differentiate the two.



If you see this screen then the CPU has somehow executed a BRK instruction – something that won't happen during the normal operation of DesTestKernal. This might indicate a marginal 6510, PLA or other chip that is affecting data reads.

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.

Limitations

DesTestKernal does assume that the first 4K of memory is mostly healthy. The screen display and the character bitmaps are stored in this RAM. If the first 4K of memory is not healthy, all may not be lost: there's some testing that can continue – but the display may not be readable. Don't despair though, since the border indicators (see below) will still give a strong indication of failing memory-bits.

Memory Errors might not be due to the memory at all. The shared address and data buses in the Commodore 64 are susceptible to accidental hijack by malfunctioning ICs. If a chip writes data to one of these buses 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 bitpositions 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 DesTestKernal 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 64K region available in MAX mode takes about

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

DesTestKernal 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.

<table>
<tr><th>Assy# \ Bit</th><th>326298</th><th>KU1419HB</th><th>250407</th><th>250425</th><th>240441</th><th>250466</th><th>250469</th><th>C128</th></tr>
<tr><td>0</td><td>U21</td><td>U21</td><td>U21</td><td>U21</td><td>U21</td><td rowspan="4">U10</td><td rowspan="4">U10</td><td>U38</td></tr>
<tr><td>1</td><td>U9</td><td>U9</td><td>U9</td><td>U9</td><td>U9</td><td>U39</td></tr>
<tr><td>2</td><td>U22</td><td>U22</td><td>U22</td><td>U22</td><td>U22</td><td>U40</td></tr>
<tr><td>3</td><td>U10</td><td>U10</td><td>U10</td><td>U10</td><td>U10</td><td>U41</td></tr>
<tr><td>4</td><td>U23</td><td>U23</td><td>U23</td><td>U23</td><td>U23</td><td rowspan="4">U9</td><td rowspan="4">U11</td><td>U42</td></tr>
<tr><td>5</td><td>U11</td><td>U11</td><td>U11</td><td>U11</td><td>U11</td><td>U43</td></tr>
<tr><td>6</td><td>U24</td><td>U24</td><td>U24</td><td>U24</td><td>U24</td><td>U44</td></tr>
<tr><td>7</td><td>U12</td><td>U12</td><td>U12</td><td>U12</td><td>U12</td><td>U45</td></tr>
</table>

Shortboard CIA incompatibility

Elsewhere in this document we’ve discussed the fact that DesTestKernal 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 inverted, 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 DesTestKernal and the screen is garbled. A 1K to 10K resistor placed between /VA14 and /VA15 (pins 2 and 3) 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 DesTestKernal 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 DesTestKernal 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 DesTestKernal 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@