# **UK Vintage Radio Repair and Restoration**

Search this website Powered By Google™ Custom Search

Vintage Radio Service Data Service Data DVD-ROM From £24.99 Instant Download Data Just £1.99 Each

🐩 UK Vintage Radio Repair and Restoration Discussion Forum > Specific Vintage Equipment > Vintage Computers

Gemini 80-Bus System

Welcome, john newcombe.

You last visited: Today at 2:28 pm Private Messages: Unread 0, Total 13.

**User CP** 

FAO

**Forum Rules** 

Community

Calendar

**New Posts** 

Search V

Ouick Links V

Log Out

Vintage Computers Any vintage computer systems, calculators, video games etc., but with an emphasis on 1980s and earlier equipment.



Page 7 of 8 **«** First < 3 4 5 6**7** $8 > <math>\nabla$ 

Thread Tools Search this Thread Rate Thread

23rd Jun 2019, 7:07 am

# john newcombe

Tetrode

Join Date: Apr 2019 Location: Skipton, North

Yorkshire, UK. Posts: 75

Re: Gemini 80-Bus System

Thank you, I will check these as you suggest.

One other symptom I have just noticed is that with the IVC plugged in, the serial version of CP/M boots but does not show the A> prompt.

Attached Files —



G812 Circuit.pdf (95.4 KB, 14 views)









23rd Jun 2019, 9:02 am

#122

# john newcombe

Tetrode

Join Date: Apr 2019 Location: Skipton, North

Yorkshire, UK. Posts: 75

Re: Gemini 80-Bus System

The voltage across C44, once charged sits at at just over 4.7V. Shorting this does not seem to change things. I can see the reset signal on the CPU go low when I do this. Also I can see C44 charge back up to 4.7V. Just for fun I checked the HALT pin and the CPU isn't in a halted state.

I do have the option of emulating an eprom using a Dataman S4, although I have never done this before so it adds anther variable. Whilst I ponder that one, I will disassemble the IVC ROM.











# iohn newcombe

Tetrode

Join Date: Apr 2019 Location: Skipton, North

Yorkshire, UK. Posts: 75



#### Re: Gemini 80-Bus System

Disassembled IVC ROM, just in case you fancy some light reading... 🙆

- Attached Files -

[2] IVC-MON V2.0.pdf (117.6 KB, 14 views)









25th Jun 2019, 10:50 pm

#124

### NealCrook

Diode

Join Date: May 2019

Location: Reading, Berkshire,

UK. Posts: 9

#### Re: Gemini 80-Bus System

Hi John,

I think Julie's suggestion of a custom EPROM is a good one. The best Z80 debug tip I ever got was from someone at NASCOM when my '2 stopped working. He told me to hold RESET low and then follow the address bus from the CPU through address decode to the ROM, and follow the data (0x31 naturally) back from the ROM to the CPU. My test equipment was a TIL209 LED and a 270R resistor.

When I brought my '2 back to life last year, I used a similar technique (though with my 'scope instead of the LED)... turned out to be a bad socket, causing an open-circuit which a TTL gate was considering to be a 1, causing the address bus to skip. I'd almost forgotten about those TTL behaviours.

With some very short custom EPROMS you can toggle all of the address and data bits, do I/O writes to the other devices on the bus. If you just do (eq) 1 I/O write in your loop it will be easy to trigger a scope or analyser and work your way along the bus. Sounds like a fun project for a rainy afternoon (Hmm. Weather's getting hotter..)

regards, Neal.











26th Jun 2019, 8:24 am

#125

# iohn newcombe

Tetrode

Join Date: Apr 2019

#### Re: Gemini 80-Bus System

Ok, thanks. I have been disassembling the ROM and learning Z80 assembler as I go as I spent all of my early days with 6502 assembler on Apple IIs and BBC Micros. When I switched to CP/M it was Pascal and C so I missed out Z80 assembler (apart from a space game on the ZX81, but I'll not count that). With that in mind and with a view to

Location: Skipton, North

Yorkshire, UK. Posts: 75

reducing the number of variables, do you have any suggestion as to what the code might be?

The memory map is divided into 8K blocks as follows;

#### Code:

```
0000 - 1FFF ROM
2000 - 3FFF VDU RAM
4000 - 5FFF CHAR GEN
6000 - 7FFF STATUS (RD)
8000 - 9FFF KEYBOARD
A000 - BFFF DATA PORT
C000 - DFFF STATUS (WR)
E000 - FFFF RAM
```

I guess reading or writing from/to each of the blocks would allow me to pop a logic probe on the various chip selects.

The basic issue is that I don't really know what I am doing . One thing I am fairly certain of though is that, in this case at least, it isn't a faulty socket as there aren't any (a).









29th Jun 2019, 12:23 am

NealCrook

Diode

Join Date: May 2019

Location: Reading, Berkshire, UK.

Posts: 9

Re: Gemini 80-Bus System

Quote:

The basic issue is that I don't really know what I am doing

Oh, you fooled me good and proper! I'll have another look at the schematics over the weekend and suggest a tiny program.

Neal.









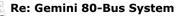
#127



NealCrook

Diode

Join Date: May 2019



John,

Location: Reading, Berkshire, UK.

Posts: 9

here is a simple program loop. If you have a 2-channel scope you should be able to trigger on /IOREQ. It should loop after <100 clock cycles.

If it's looping OK, you can use the other channel to probe around to see chip selects on each of the memories, on the 6845, on the buffers used for the kbd and status ports and on the latch used for the write status port. You should also be able to see the low 6 bits of the address bus toggling, and see the data toggling and propagating though the buffers.

I'm assuming you can figure out the details of what to look at on the schematic, but if you need more details, let me know.

#### Code:

```
# File ivc test1.asm
0000
                       ;;; ivc test1.asm
0000
                       ;;; simple loop for hw debug of once-working now faulty I
0000
                       ;;;
0000
0000
                              ;; in ROM and executes from reset.
0000
                              ;; assume no stack/working RAM so no subroutines
0000
                              ORG 0
0000
0000 db 00
                      loop:
                              in a, (0)
                                                   ; should generate /CS to
                                                  ; video RAM
                              ld hl, 0x2000
0002 21 00 20
0005 7e
                              ld a, (hl)
                                                   ; read then write
0006 77
                              ld (hl), a
0007 21 00 40
                              ld hl, 0x4000
                                                 ; char gen ROM or RAM
000a 7e
                              ld a, (hl)
                                                  ; read then write
000b 77
                              ld (hl), a
000c 21 00 60
                              ld hl, 0x6000
                                                   ; status read
000f 7e
                              ld a, (hl)
                                                   ; read
0010 21 00 80
                              ld hl, 0x8000
                                                   ; keyboard
0013 7e
                              ld a, (hl)
                                                   ; read
0014 21 00 a0
                              ld hl, 0xa000
                                                   ; host comms data port
0017 7e
                              ld a, (hl)
                                                   ; read then write
0018 77
                              ld (hl), a
0019 21 00 c0
                              ld hl, 0xc000
                                                    ; status write
001c af
                              xor a
001d 77
                              ld (hl), a
                                                   ; write 0
                                                 ; workspace RAM
                              ld hl, 0xe000
001e 21 00 e0
0021 7e
                              ld a, (hl)
                                                   ; read then write
0022 77
                              ld (hl), a
0023 18 db
                              jr loop
0025
# End of file ivc test1.asm
```

The processor sub-system is very straightforward so if this code doesn't even loop there's a fundamental problem with the CPU/clock/reset/address/data signals. If that's the case the next experiment is to take the R24/D1 junction to GND (hold the chip in reset) then probe to see:

/MREQ /RD both asserted Address bus 0 ROM chip-selected data bus reads 0xDB (if this ROM is fitted) or 0x31 if the standard ROM is fitted.

You should also check that you've got CLK on the Z80 and both CLK and E on the 6845.

regards,

Neal









30th Jun 2019, 12:44 pm

#128

### NealCrook

Diode

Join Date: May 2019 Location: Reading, Berkshire,

UK. Posts: 9

### Re: Gemini 80-Bus System

I use a PERL module CPU::Z80: isassembler to explore/disassemble Z80 code. Given a binary and an entry point it traces all paths of execution through the code from that point. Using that result as a starting point you can give it hints of other entry points. As you explore the code the "hints" can get more and more detailed: giving names to labels and back-annotating comments, without ever modifying the binary or reassembling.

Here's a partial exploration of the IVC ROM using this tool, dis ivc,txt is a PERL program that contains all of the hints, ivc.asm.txt is the result. I did a most of this "blind" in a couple of hours and then read the (excellent) software manual for the board and corrected some wrong guesses that I had made.

This code makes extensive use of the alternative HL/BC/DE register set. The exx instruction swaps between the two, but it's super-tricky to inspect the code and know which of the two sets is in use at any time. I expect the original source file/author had some "rules" about what was the default swap and how to handle use of exx but it would take more inspection that I have time for.

Another strange thing is that the code uses NMI from the vertical retrace. There is code at the NMI entry point, but there is no occurrence of RETN (return from NMI) anywhere in the code. Reading the Z80 manual, I conclude that RETN is not strictly necessary and that RET will do instead, PROVIDED that you never use interrupts (only the nonmaskable interrupt) -- which is true for this code.

#### Neal

Attached Files



dis ivc.txt (5.3 KB, 9 views)











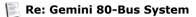
#129

# <u>iohn</u> newcombe

Tetrode

Join Date: Apr 2019 Location: Skipton, North Yorkshire, UK.

Posts: 75



Thank you so much. I am away this week but I will code up a rom with your code I do have a dual channel scope so I will do what you say.

CLK, seems to be good on z80, and IC31 where it seems to be gated with the iorg to provide crtc E, but as I am not seeing any change in jorg I am not seeing E change.

Thanks again for the time you have spent on this, I will post an update as soon as I have been able to try your code.











#130

### john newcombe

Tetrode

Join Date: Apr 2019 Location: Skipton, North

Yorkshire, UK. Posts: 75



#### Re: Gemini 80-Bus System

Just a quick update, I have only been able to spend a few minutes so far, just enough time to see your code looping with IORO triggering the scope. Will keep you posted.









#131

# iohn newcombe

Tetrode

Join Date: Apr 2019 Location: Skipton, North

Yorkshire, UK. Posts: 75



### Re: Gemini 80-Bus System

One thing that may be significant though is that the code only runs once I have grounded the reset on the CPU. It would not run from power up. I quickly replaced your ROM for the IVC MON and grounded reset, however this would not run.









### NealCrook

Diode

Join Date: May 2019 Location: Reading, Berkshire,

UK. Posts: 9

#### Re: Gemini 80-Bus System

Well, it's progress and good news to know that the CPU can execute code and loop. Odd that the power-on-reset circuit does not work. As John Hanson suggested, maybe that capacitor C44 is leaky. If you are in a card rack with the rest of the working system you should also be able to do a reset by an I/O read or write to port 0xB3 (that would also be a way to do some kind of a test on the ?IC36? decode PROM.

While watching The Cure at Glastonbury (on TV..) I made more progress on the disassembly; it's a nice puzzle a bit like a crossword. I found the jump tables that search a list of escape codes and invoke a code fragment as a result, and I made a bit more sense of some other parts of the code. I'll post the result once I've checked that it assembles back to an identical binary.

Neal.











4th Jul 2019, 8:17 am

#133

### john newcombe

Tetrode

Join Date: Apr 2019 Location: Skipton, North

Yorkshire, UK. Posts: 75





#### 🖳 Re: Gemini 80-Bus System

Thanks Neal, really appreciate the help. I will swap C44 when I get back from Reading on Sunday (your neck of the woods if my memory serves me). I should have done this back when JBH first mentioned it.







#### Yesterday, 8:54 pm

#134

# john newcombe

Tetrode

Join Date: Apr 2019 Location: Skipton, North Yorkshire, UK.

Posts: 75

### Re: Gemini 80-Bus System

I have been able to spend some more time with the test code provided by Neil and have a few results. Before starting I changed C44 in the reset circuitry as suggested by John, following this, the code now runs from power up. I did retest the original ROM and a second that I now have with no success.

So with the test code running I had a probe around with the scope. This is what I found.

VIDEO RAM IC9

/CS was permanently low, this is correct as it is tied to ground.

/WE shows negative going pulses at around 33Khz

Code:

e.g.	
/OE shows positive going pulses at the same frequency	
Code:	
e.g.	
CHAR GEN ROM/RAM	
/CS Low on the ROM and high on the RAM this is as I would have expected.	
CRTC	
/CS shows negative going pulses at around 33Khz as per the VIDEO RAM /OE shows negative going pulses at around 667Khz /WE shows pairs of negative pulses at around 250Khz	
Code:	
e.g.	
WORKSPACE RAM	
/CS shows negative going pulses at around 33Khz Code:	
e.g.	

I did notice that the DOT clock is all over the place and noticed that IC25 Pin 9 shows the same negative 33Kz pulses. Pin 9 seems to select either the crystal controlled clock or the VFO. I am thinking that this is something to do with the status port access of the test program. If that is the case from what I can see, which is guite limited, it all seems to be working as expected, or am I missing something?











#135



# iohn newcombe

Tetrode

Join Date: Apr 2019 Location: Skipton, North

Yorkshire, UK. Posts: 75



#### 🖳 Re: Gemini 80-Bus System

Sorry... Neal not Neil.









Yesterday, 10:46 pm

#136

### NealCrook

Diode

Join Date: May 2019 Location: Reading, Berkshire,

UK.

Posts: 9



#### Re: Gemini 80-Bus System

video RAM looks correct: it is output-enabled by default for access by the 6845 or for a read by the Z80, the /OE only negates to allow the data bus to be driven for a write, so the /WE short low pulse should correspond to the /OE short high pulse.

You should also be able to look at the MUX select on IC10,11,12 (pin 1) which should have 2 pulses per loop a bit like you show for the CRTC -- these pulses correspond to the Z80 taking priority over the 6845 and generating the address (once for the read and once for the write).

char gen rom/ram looks correct: you should also see /OE and /WE behave in the same ways as for the video RAM.

You should also be able to look at the MUX select on IC15,16,17 (pin 1) as for the video RAM

crtc: I'm not sure about your comments here because you refer to /CS, /OE and /WE. In the code loop there should be 1 read of the CRTC and no writes. That read should correspond to the /IORQ that you're triggering on. If you see it OK, change to trigger on /CS and use the other probe to look at E. You should see a pulse on E that is overlapping and out of phase with the pulse on /CS - generated by the 'flop IC31b.

workspace RAM: I would expect to see pairs of negative pulses on /CS - the first one matching a negative pulse on /OE and the second matching a negative pulse on /WE.

Quote:

I did notice that the DOT clock is all over the place and noticed that IC25 Pin 9 shows the same negative 33Kz pulses. Pin 9 seems to select either the crystal controlled clock or the VFO. I am thinking that this is something to do with the status port access of the test program.

I started out thinking this was a problem, but then I realised this part is a 273 which is a transparent latch, and the CLK pin is only a decode of /MREQ and the address bus. The data bus might not be stable at the \*start\* of /MREQ so there could be a glitch on the outputs at this time, which stabilises before the latch closes. Not very clean but this port probably only gets written 1 time at startup so it would likely not cause a problem.

If IC25 is suspect you can use a dedicated test program like this:

```
Code:

ld hl, 0xc000 ; status write
loop: ld (hl), a ; write
inc a
jr loop
```

you should see square waves on the outputs of IC25, with pin 12 at the highest frequency and each next pin at 1/2 the frequency. However, if my guess above is correct you will also see a 33kHz "glitch" superimposed on some if not all of the signals.

What have we learned? Looks as though the CPU is executing code OK and the data but and at least some of the address bus is OK. The memory decoding seems OK.

#### I suggest

1/ putting the proper ROM back in and seeing if IC25/9 is static. OR blow a ROM in which you set its output to 1, which selects the crystal dot clock:

```
Code:

ld hl, 0xc000 ; status write

ld a, 0x02 ; crystal dot clock

ld (hl), a ; write

loop: jr loop
```

In either case, check that IC3/6 is stable 16MHz, While you're at it, take a look at the rest of the dot clock logic. With a 16MHz dot clock you should see 1/4 (4MHz) on IC4/21 and then, on the 74165 shift register (can't read the IC number) you should see a 16MHz clock and a load pulse that blips once every 8 clocks.

2/ if you can do jo reads and writes from the main CPU to this board, check that you can pulse the CPU reset by a read or write to port 0xb3.

3/ if all that looks OK, I think it would be worth trying to check out the communications between the main CPU and this board -- the logic in the top left quarter of schematic sheet 1.

BR,

Neal.

Last edited by NealCrook: Yesterday at 10:56 pm, Reason; hit post too soon!





### john newcombe

Tetrode

Today, 9:02 am

Join Date: Apr 2019 Location: Skipton, North Yorkshire, UK.

Posts: 75

### Re: Gemini 80-Bus System

Hi Neal, wow!, thank you for your assistance.

The anomaly around the CRTC is down to my notes being a tad messy. The /OE /WE stated for the CRTC were in fact the signals for the Workspace RAM. The IORQ is signalling the CRTC as expected.

I will check the MUX lines as you suggest and then move on to check the DOT clock and IC25.

I have the RP/M monitor installed in the main CPU so should be able to install and run some code to do some I/O on port 0xb3.

I will let you know how I get on...









#### Today, 9:35 am

#138

#### john newcombe Tetrode

Join Date: Apr 2019 Location: Skipton, North

Yorkshire, UK.

Posts: 75

#### Re: Gemini 80-Bus System

The RP/M monitor, as I am sure you know behaves like a cassette based CP/M and so the transient area starts at 0100. Would the following code, loaded at 0100h be enough to pulse the IVC's CPU reset from the CPU Board?

Code:







ld hl, 0x00b3 21 B3 00 loop: ; port B3 7E ld a, (hl) ; read 77 ld (hl), a ; then write 18 F9 jr loop











#139



### JohnBHanson

Hexode

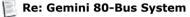
Join Date: Aug 2009 Location: Worthing, Sussex,

UK.

Posts: 369







No the port b3 is in io space. You need to do an in a,0b3h to do the reset.

also out (0b3), a will also work as a reset.







#140



### john newcombe

Tetrode

Join Date: Apr 2019 Location: Skipton, North

Yorkshire, UK. Posts: 75



### Re: Gemini 80-Bus System

OK, thanks, time to get my Rodney ZAKs books out I think...





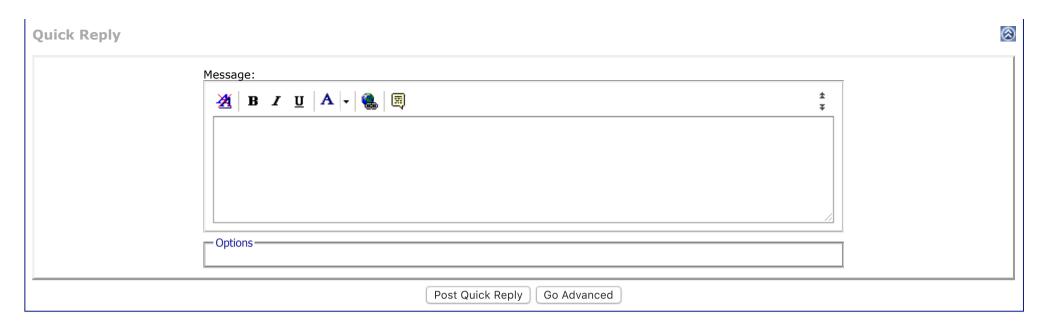




Page 7 of 8 **«** First < 3 4 5 6**7** $8 > <math>\nabla$ 

**Quick Reply** 





« Previous Thread | Next Thread »

All times are GMT +1. The time now is 5:48 pm.

Contact Us - Home - Forum Rules, Privacy & Disclaimer - Top

All information and advice on this forum is subject to the WARNING AND DISCLAIMER located at <a href="https://www.vintage-radio.net/rules.html">https://www.vintage-radio.net/rules.html</a>.

Failure to heed this warning may result in death or serious injury to yourself and/or others.

Powered by vBulletin®
Copyright ©2000 - 2019, vBulletin Solutions, Inc.
Copyright ©2002 - 2019, Paul Stenning.