Subject: Retrocomputer projects using EPM240 development board
Posted by plasmo on Sat, 01 Jun 2024 04:56:14 GMT
View Forum Message <> Reply to Message

EPM240 development board is an inexpensive (~$10) CPLD learning board based on Intel (Altera) Max2 EPM240 CPLD. I bought mine from this eBay seller, https://www.ebay.com/itm/354417399620. The board has a 100-pin EPM240 with 50MHz oscillator and four 2x10 headers. EPM240 is a 3.3V device and not 5V tolerant. A mezzanine board can mate with the headers to host additional electronics. There are enough room on such mezzanine board for two 600-mil DIP side-by-side within the space of four 2x20 headers. That's enough room for a retro processor, memory and associated connectors.

EPM240 has 240 macrocells which is much larger than the 5V EPM7XXX CPLD that I've been using for many projects. While I have larger design space, I have to deal with the 3.3V limitation which restricts the retro computer families I can work with.

I plan to develop mezzanine boards for Z80, Z180, 65C02, 68SEC000, and 68EZ328.
Bill

File Attachments
1) EPM240_dev_board_F.jpg, downloaded 1506 times

Subject: Re: Retrocomputer projects using EPM240 development board
Posted by plasmo on Sun, 02 Jun 2024 04:30:25 GMT
View Forum Message <> Reply to Message

RomWBW-capable Z80 mezzanine board.

The first mezzanine board is based on Z80. The design was successfully prototyped previously, so this is not a new design but instead used to check out mechanical fits. I can't find the mechanical drawing of the EPM240 dev board, but I can see the four 2x10 connectors are not on 0.1" grid. I also want a cut out for the on/off power switch; so the resulting Z80 mezzanine board is 2.1" x 3" with a cut out for on/off power switch that's slightly off-center. The four 2x10 connectors do fit correctly with EPM240 dev board.

The Z80 mezzanine board has a CMOS Z80 and 512K RAM. EPM240 contains the bootstrap ROM, emulated ACIA, memory bank logic for sixteen 32K banks, RTC interface, SD card interface, and I2C interface. EPM240 dev board has a 50MHz oscillator that's divided by 3 to 16.67MHz as Z80 clock and that is further divided by 9 to 1.852MHz which is 0.5% faster than the standard 1.8432MHz clock for 115200 baud serial port. The CPLD logic is 60% utilized.

EPM240 has an internal 1K flash which is large enough to bootstrap from SD card, but currently I bootstrap via the serial port and load monitor and RomWBW serially every power cycle. It is working, however, I've noticed that while I've tried a dozen CMOS Z80 that all worked at 3.5V, many of them won't run or failed zexall at 3.3V. The current Z80 mezzanine board design used the 3.3V generated by the regulator on EPM240 dev board, but a better solution may be a local

regulator on the mezzanine board that generate 3.5V from 5V to power the Z80. As long as the mezzanine voltage is within 0.3V of EPM240 dev board, there won't be any interface issues. Power consumption while running RomWBW is 49mA @ 5V.

I still need to figure out how to boot from SD card.
Bill

File Attachments

1) 3VZ80_RomWBW_annotated_F.jpg, downloaded 1326 times
2) 3VZ80_with_RTC_F.jpg, downloaded 103 times
3) 3VZ80_romwbw_zexall.jpg, downloaded 1289 times
4) 3VZ80_rev0_scm.pdf, downloaded 148 times

Subject: Re: Retrocomputer projects using EPM240 development board
Posted by greghol on Tue, 04 Jun 2024 07:21:39 GMT
View Forum Message <> Reply to Message

Very nice Bill... I did pick up a couple of these EMP240 modules to play with. Is the Z80 mezzanine board stable?

Greg

Subject: Re: Retrocomputer projects using EPM240 development board
Posted by plasmo on Tue, 04 Jun 2024 16:54:25 GMT
View Forum Message <> Reply to Message

The design is stable if populated with a suitable 3.3V Z80. This is a Z80 that will boot RomWBW and run the hour-long zexall.com without errors, which is about 1 out 3 CMOS Z80 (these are collection of used (many refer them as fake) Z80 I acquired on eBay). If Z80 supply voltage can be raised to 3.5V, then vast majority of CMOS Z80 will run and be stable.
Bill

Subject: Re: Retrocomputer projects using EPM240 development board
Posted by greghol on Mon, 10 Jun 2024 04:04:57 GMT
View Forum Message <> Reply to Message

Bill,

Are you going to put the design files on retrobrew? I would like to build one to play with and I have a batch of boards queued up for JLCPCB.

Another board that would be fun to adapt to the CPLD board is the W65C02 which as you know

runs at 3.3V. Perfect match.

Greg

Subject: Re: Retrocomputer projects using EPM240 development board
Posted by plasmo on Mon, 10 Jun 2024 14:16:05 GMT
View Forum Message <> Reply to Message

I don't have much time lately, so I have not created a homepage for 3VZ80. I will soon. I'm actually pretty excited about using EPM240 dev board for other applications, such as the beam racing 65C02 you've mentioned. EPM240 chip itself is pretty cheap ($1.50) on eBay.

I think you are in USA, right? PM me your address and I'll drop off a blank 3VZ80 pc board in letter mail. You may need to screen for a Z80 that can run at 3.3V, which is about 1 in 3 CMOS Z80. I found most Z80 will run at 3.5V, however. So I plan to revise 3VZ80 with local voltage regulator adjusted to 3.5V.

EPM240's internal 1KB flash is very cool. It is enough to load programs from SD, so external boot ROM is not necessary. It is booting a small file loader now; I'm working on a SD boot loader slowly only when I have time...

Bill

Edit, another interesting possibility is 68SEC000 with SDRAM; two chip 68K mezzanine board with EPM240 handles bootstrapping, IO, and SD disk.

Subject: Re: Retrocomputer projects using EPM240 development board
Posted by greghol on Tue, 11 Jun 2024 04:30:54 GMT
View Forum Message <> Reply to Message

Bill,

I sent an email to the @<starts with s with five letters> if that is still active. Thanks.

I'll see if I can find a CMOS Z80 that will work but if not then the 3.3V could be cut with a regulator on the board for split supply. The 5V could be tapped. Looking at the schematic for the EPM240 board there is a +5V pin to get the regulator input voltage.

Yeah, The 65C02 would be real fun with this as well as the 68K. The "SEC" must be a 3.3.V supplied part.

Thanks,
Greg

Subject: Re: Retrocomputer projects using EPM240 development board
Posted by plasmo on Tue, 18 Jun 2024 04:02:43 GMT
View Forum Message <> Reply to Message

EPM240 has an embedded 1KB flash memory. While it is 1KB, it is actually organized as a word-wide memory, 512x16. It is somewhat painful to use word-wide memory as boot ROM for Z80, so I only use half of the embedded flash or 512 bytes to boot Z80.

The task now is fitting SD bootstrap code in 512 bytes. This is my first foray into bare metal SD access and it was painful! I started out with Wayne Warthen's RomWBW SD code and have made so many simplifications that it is unrecognizable, but it does fit 512 bytes. However, it can only handle V1 type SD card, the standard SD card in microSD format. This is the card up to 2GB in capacity.

To be compatible with RomWBW, I configure the hardware SD interface to emulate the bit-bang PIO interface. Unfortunately that's rather slow interface, only capable of 25KB/sec transfer with 16MHz Z80 clock. To boot into RomWBW from reset/power up, the 512-byte bootstrap copies 480KB RomWBW stored in SD to RAM then execute RomWBW. The picture shows RomWBW being loaded, each dot is a sector read. It takes 23 seconds to load RomWBW. I'm not happy about that.

EPM240 is 80% utilized, so I should be able implement the SD_GET and SD_PUT software routines in hardware and potentially speed up the SD access significantly.

Bill

File Attachments
1) bootSD_RomWBW.jpg, downloaded 1133 times

Subject: Re: Retrocomputer projects using EPM240 development board
Posted by wsm on Tue, 18 Jun 2024 21:03:49 GMT
View Forum Message <> Reply to Message

Bill - Some food for thought:

Having 1K of user flash certainly opens up some interesting possibilities. I've only taken a cursory look at the EPM240 and I don't use Quartus. Although the UFM is serial, the documents state there is interface logic for parallel address and data so I'm not sure what the difficulty would be with using 512x16 flash as a 1024x8 ROM. The difficulty with x16 is on writes whereas x8 reads can simply use A0 as a multiplexer input and I've done that several times using standard x16 SRAM and/or x16 flash chips in 8-bit systems.

With 1024 bytes of Z80 code you should be able to include a very generalized MMC & SD I/O routine as mine is about 3/4 of that size and handles all the different uSD cards I've tried. Hopefully with the free EPM logic you could add a HW SPI routine at 25 or 50 MHz and possibly a

DMA controller on top of that.

I think the biggest performance gain will come from a HW SPI controller instead of bit-bang PIO. The limiting factor with a fast SPI interface would be the 16 MHz Z80 INIR instruction where the theoretical transfer rate is about 762 KB/S, although there'll also be some setup overhead. If you're short of logic cells, the much slower I2C interface at 400 KHz is better suited for bit-banging as each bit would be about 40 T-states.

I'm not sure if it's a typo, but is it really necessary to read 480KB of RomWBW? The whole purpose of mass storage is to only R/W what is necessary.

Subject: Re: Retrocomputer projects using EPM240 development board
Posted by plasmo on Wed, 19 Jun 2024 18:53:56 GMT
View Forum Message <> Reply to Message

Great food for thought. I'm new to EPM240 and SD card interfacing, so there are plenty room for tradeoffs and optimization. EPM240 is a nice big (and cheap) sandbox. There is also a pin-compatible EPM570.

My past designs used CF disk for program storage because it only needs a few instructions to load the bootstrap code in CF disk. SD card needs many more instructions, too many to fit in the logic fabric of CPLD. So I'm interested in Max2 family of CPLD such as EPM240 that all have 1KB embedded flash which opens the door for SD card based bootstrap. The embedded flash is serial eprom, but like you've said, there is a library function for parallel interface. No free lunch, however, parallel interface for 16-bit wide data needs more logic than 8-bit, so I want to save the logic for faster SPI and other circuits, or just generally playing around.

ROMless RomWBW is indeed 480KB. One reason for its size is a 256KB RAM disk preloaded with the essential CP/M programs. Since SD card will always be there, an alternative is blank 256KB RAM disk but populate the SD card with CP/M programs. This should speed up the RomWBW loading to around 10 seconds.
Bill

Subject: Re: Retrocomputer projects using EPM240 development board
Posted by wsm on Fri, 21 Jun 2024 16:50:18 GMT
View Forum Message <> Reply to Message

I understand the desire to limit macrocells as it's easy to get carried away adding features. Even more fun :(doing upgrades when the device has been pin locked which can then force a choice of significant PCB rework or having to go to a larger device ... been there & done that.

I'd offer you my HDL SPI code like on the MinZ-C but it's in ABEL for Xilinx CPLDs and somewhat confusing as it supports both 5 & 25 MHz operation plus DMA. One way it speeds up reads is when the host finishes reading a SPI byte, the CPLD immediately starts another SPI read. Other than the first read of a sector, data should always be available for a 20MHz Z80's INIR without

wait states if the SPI data rate is about 10 MHz or faster. This doesn't cause any SD issues if MOSI is FFh which is not a valid SD command since they always start with a high order bit of 0.

The BASIC concept of my SPI routine is a fairly simply state machine clocked at 2X the SPI rate. However because of the two clock domains (host & CPLD) it was necessary to be very careful doing the startup synchronization with 36 MHz Z180s.

FYI: It doesn't seem to have hit the Altera chips yet but Xilinx CPLDs have been going up in price significantly since AMD bought them and then decided to end-of-life ALL their CPLDs. XC95xx CPLDs are still fairly reasonable but the 128 cell or larger XCR3xxx and XC2Cxxx CoolRunners are getting really expensive, especially the VQ100 packages which is what I've primarily been using. Some of the eBay prices for questionably "new" parts are now higher than Mouser & Digi-Key. Although Lattice ispMACH4000 CPLDs are still available, they've now decided to charge $699+ per year for ispLEVER, the development SW that previously had free one year licenses. Definitely a show-stopper for me even though I have a possible project that could really benefit from their 32 cell 2.5nS device.

Subject: Re: Retrocomputer projects using EPM240 development board
Posted by plasmo on Sat, 22 Jun 2024 14:48:48 GMT
View Forum Message <> Reply to Message

On 6502.org forum, there is a SPI design for ATF1504 CPLD, 65SPI. I've ported it to Altera. The core design is quite simple, so I should be able to fit it into EPM240. I don't necessarily want to change the SPI interface of RomWBW. CP/M programs are generally small, so 25KB/sec bit-bang is adequate to load CP/M programs; it is the loading of 480KB RomWBW image I want to speed up. I can have a separate, faster SPI interface dedicated to loading RomWBW then run RomWBW unmodified with bit-bang SPI interface.

Alternatively, I've found RomWBW system code is only 128KB, RAM disk image is 256KB, and there are significant unused space in the 480KB image. So I can just load the 128KB system code and leave the RAM disk un-initialized which takes 6 seconds to boot. I can also crank up the CPU clock to 25MHz and drop the bootstrap time to 4 seconds. Now that's tolerable bootstrap time.

CPLD are getting rather expensive now. I do have a stock pile of CPLD sufficient to last my life, but others interested in reproducing my designs may face significantly cost and availability issues. That is regrettable, but that's a constant challenge of working with obsolete technology. EPM240 remains affordable, that's probably why EPM240 development board is lesser than $10, for now.

Bill

PS, homepage for 3VZ80 is here,
https://www.retrobrewcomputers.org/doku.php?id=builderpages:plasmo:epm240dev:3vz80:3vz80rev0

Subject: Re: Retrocomputer projects using EPM240 development board
Posted by greghol on Mon, 01 Jul 2024 22:06:59 GMT
View Forum Message <> Reply to Message

I ordered two of the EPM240 boards off of ebay. One worked for about 1 min of on time cycled a half dozen times or so. Then nothing. 3.3V short. Did visual inspection of board and removed 3.3V regulator since it was getting hot and didn't know if this was the issue. One would think that the regulator would do a thermal shutdown. The short persisted after the reg was pulled so I the pulled the EPM240 and verified it was the culprit. Ordered a lot of EPM240 based on Plasmo's eBay link from another thread for about US$1.50 each in lots of 10. Ali has these for US$1.00 each in lots of 10. I poked around and did some research on these boards and devices and I found this thread:

KrzysioTesterEPM240 - the EPM240 CPLD chip tester, the fear of sellers from alieexpress
https://forums.nesdev.org/viewtopic.php?t=23635

An interesting read.

Greg

Subject: Re: Retrocomputer projects using EPM240 development board
Posted by plasmo on Tue, 02 Jul 2024 21:44:59 GMT
View Forum Message <> Reply to Message

I redesigned the 3VZ80 mezzanine board so it has a local adjustable 3.3V regulator. The local 3.3V is adjusted to 3.5V based on my past observations that majority of CMOS 20MHz Z80 will run OK down to 3.5V while only about 1/3 of the same Z80 will run to 3.3V. Mezzanine board operating at 3.5V won't affect the 3.3V EPM240 CPLD because EPM240 can tolerate I/O voltage up to 0.3V higher.

I tested 10 used CMOS Z80 that I recently purchased on eBay and sure enough at 3.5V and 16.7MHz CPU clock, every Z80 will boot into RomWBW and passed first few tests of zexall (zexall is a long test, but once a processor passed the first 3-4 tests it is likely to pass the remaining tests).

I'm done with Z80 for now. I have two other mezzanine boards (6502, 68K) already fabricated for EPM240 dev board.
Bill

File Attachments
1) 3VZ80_rev1_Mezzanine_board.jpg, downloaded 93 times
2) 3vz80_epm240_side_view.jpg, downloaded 104 times

Subject: Re: Retrocomputer projects using EPM240 development board
Posted by greghol on Thu, 11 Jul 2024 23:43:19 GMT
View Forum Message <> Reply to Message

Bill... I got the Rev 0 of the 3VZ80 put together and tried out with a fresh 20MHz Z80 and it works fine.
I didn't have the MCP130-315 on hand so I just used a 4.7K and 2.2Uf cap which has a TC of about 10ms. The board
appears to be stable if left alone but it will come up after one or two rapid button presses. Maybe the MCP130 will help
this out and I will get some in my next Mouser order. Maybe an artifact of running at 3.3V but it is not a hassle or anything.

I wanted to get zexall on the SD card so I was trying to use some the ROMWBW xfer programs from the image. I always had luck with XM
with my other Z80 RC2014 and your builds but not with the 3VZ80. I'm using the same serial USB module as you are using. Have you had luck getting
XM to work? Short of that I could use cpmtools and xfer images over or just pull some source for xmodem that I have in a Z180 monitor with tweaks.

HW wise I would like to try adding another SPI bus to the EPM240.

Tnx,
Greg

Subject: Re: Retrocomputer projects using EPM240 development board
Posted by plasmo on Fri, 12 Jul 2024 04:59:39 GMT
View Forum Message <> Reply to Message

Greg,
Good to hear you are successful with 20MHz Z80 operating at 3.3V. My feeling is that Z80 is much better than its data sheet due to many years of process improvements, that the more recent Z80 are likely to run 3.3V at 16.7MHz. My earlier observation that only 1/3 of used Z80 can run at 3.3V is based on my collection of used Z80 acquired on eBay with their original markings painted over, so I don't know their datecode and whether more recent datecode CMOS Z80 performed better than older CMOS Z80. I've placed an order with Mouser for two 20MHz Z80, but they are backordered and I'm still waiting. I plan to do overclock and low voltage testing with these new Z80.

XM is working on my hardware. You do need to enable hardware handshake for it to work.

I have updated the 3VZ80 homepage with a CPLD design file that can load and boot RomWBW in 6 seconds. I didn't change the hardware design, I only changed the SD bootstrap code to only load 128KB of RomWBW system but not the 256KB RAM disk contents. You'll find drive A (RAMdisk) blank but drive B-H still have RomWBW files.

The homepage is also updated with design files of rev1.1 of 3VZ80.

https://www.retrobrewcomputers.org/doku.php?id=builderpages:plasmo:epm240dev:3vz80:3vz80rev1_1
Bill

Subject: Re: Retrocomputer projects using EPM240 development board
Posted by plasmo on Fri, 12 Jul 2024 19:26:49 GMT
View Forum Message <> Reply to Message

While operating Z80 at 3.3V is clearly not specified in the Z80 datasheet, operating W65C02 at 3.3V is specified in its datasheet. W65C02 datasheet indicates that 14MHz operation is possible at 3.3V, but in reality it is capable of much higher frequency. I'm shoot for the VGA frequency because I like to race the VGA beam with 25MHz 6502.

Here is rev0 3V6502 for EPM240 dev board. The 25MHz 6502 fetches a byte of image data every 8 clock cycles; the data is feed to a 8-bit shift register operating at 25MHz to drive the VGA display with 25MHz pixel clock. So the display is monochrome 640x480 with 6502 busily feeding the data during the active video period. During the vertical retrace period the 6502 can do non video work.

It is a simple design, but VGA and PS2 connectors take additional room so the mezzanine board overhang the EPM240 dev board. There is also a mistake with the PS2 placement such that it is pointing inward rather than outward. To work around the mistake, the PS2 connector is mounted tilting 30 degress up so it still can accept a PS2 plug.

Currently the 6502 is working with 25MHz clock passing memory diagnostic, so the hardware seems to work OK at 3.5V 25MHz. There are quite a bit of software works ahead.
Bill

File Attachments
1) 3V6502rev0_annotated.jpg, downloaded 974 times
2) 3V6502_on_EPM240DevBoard_sideview.jpg, downloaded 80 times

Subject: Re: Retrocomputer projects using EPM240 development board
Posted by greghol on Fri, 12 Jul 2024 23:09:09 GMT
View Forum Message <> Reply to Message

Bill,

Yeah, I didn't mention that I already tried to turn handshaking on via TeraTerm and that didn't help much. I do get the first xmodem block in and that is it. I'll tweak the EPM240 code and try at 57.6k baud and see what happens. I'll try another serial to USB too.

The second EMP240 board, that had a shorted CPLD, got fix with another part so I will try the fast load CPLD image on that one.

Nice that the 6502 version working out so far!

Thanks,
Greg

Subject: Re: Retrocomputer projects using EPM240 development board
Posted by plasmo on Sun, 14 Jul 2024 00:40:36 GMT
View Forum Message <> Reply to Message

Greg,
Have you tried sending file to PC (XM S filename)? It should work without handshake.

RomWBW's "i 0 57600" command won't work because serial port in CPLD is fixed to 115200. However, I can change the CPLD serial prescaler to support 57600. Try the attached programming file, all it did is changing the serial prescaler to support 57600 (divide the system clock by 18). You should be able to XM without hardware handshake at 57600.
Bill

File Attachments
1) 3VZ80_57600N81.zip, downloaded 82 times

Subject: Re: Retrocomputer projects using EPM240 development board
Posted by greghol on Sun, 14 Jul 2024 07:53:17 GMT
View Forum Message <> Reply to Message

Bill,

"xm s <filename>" works fine.

Thanks for making the 57.6k baud image as this works just fine in receiving the files via xmodem with no flow control. Also tried zmrx and it works too. HW flow control on and it will not send a chars out of teraterm and I dont see the led flash on the usb serial confirming. The board will send chars to teraterm just fine with HW flow control on.

I've tried with minicom to same result as well as a FTDI based USB to serial dongle with same results.
The CP210x USB to serial is the same as yours but mine has no part number markings, only a pin one dot marker. But both the FTDI and CP210x based units are doing the same thing so that is not the issue.

DTR is always low measured by my DVM and not a scope. I'm not going to get hung up this for now. It will be slower but that's ok for now as I want to move onto the EPM240 design.

Thanks,
Greg

Subject: Re: Retrocomputer projects using EPM240 development board
Posted by plasmo on Sun, 14 Jul 2024 15:02:55 GMT
View Forum Message <> Reply to Message

Modification to CP2102 are needed to use hardware handshake with CP2102. This is a picture of the mod

File Attachments
1) 544C580A-97A0-4C47-A800-63F8E8D696A8.jpeg, downloaded 928 times

---

Subject: Re: Retrocomputer projects using EPM240 development board
Posted by greghol on Sun, 14 Jul 2024 19:20:03 GMT
View Forum Message <> Reply to Message

Ahhhhh... The missing puzzle piece. I was wondering why CTS wasn't being used but figured that since I was using the same module it was ok with DTR as is. Working at 115.2k baud now with HW control!

Greg

Subject: Re: Retrocomputer projects using EPM240 development board
Posted by plasmo on Wed, 17 Jul 2024 18:14:03 GMT
View Forum Message <> Reply to Message

My focus is getting 3V6502 to do 25MHz VGA beam racing, but I have another mezzanine board based on 68SEC000 which is capable of 3.3V operation. UTSource has new 68SEC000, but they renegaded on the published price and wanted much higher $14 each. So instead, I bought some from this eBay vendor and I want to test the device immediately upon receival so I can return them if they are not working.

The board design is fairly simple, consists of 68SEC000 and 512Kx16 RAM. EPM240 will provide the bootstrap code, serial port, SD, I2C, and RTC interface. RAM is not needed to run "scream" test where a small program resides in CPLD writes incrementing data to the serial port. The board has a few design errors, but it does passed "scream" test at 16.7MHz CPU clock. So I have at least one good 68SEC000.

Now back to 3V6502 project.
Bill

File Attachments
1) 68KSEC_initial_test_annotated.jpg, downloaded 922 times
2) 68KSEC_scream_test.jpg, downloaded 77 times

Subject: Re: Retrocomputer projects using EPM240 development board
Posted by plasmo on Tue, 23 Jul 2024 00:59:22 GMT
View Forum Message <> Reply to Message

The goal of 3V6502 is a standalone 6502 computer with own 640x480 monochrome VGA display capable of text and graphic, PS2 keyboard, serial port, and SD card for mass storage. While the 6502 processor is operating at 25MHz, it spends 90% of its throughput driving the 640x480 display. During the active video phase, it is servicing the 31.5KHz interrupt from the horizontal sync and output 80 bytes of video data (640 pixels) to the display. It also look for PS2 keyboard input and serial port input during the interrupt service. The processor is fully utilized during the active video phase, but every 16.7mS it has 1.4mS of vertical retrace period where it can do real work. This is not much time; it is effectively a 2MHz 6502.

I've implemented a simple monitor that can edit memory, display memory, execute program using PS2 keyboard as input and VGA monitor as output. The attached animated GIF file shows display/edit/go commands of the simple monitor running on 3V6502. The 6502 clock is 25MHz, not 25.175MHz, so the screen as captured by the video capture hardware is shifted by one character. I can adjust the video monitor to compensate for the shifted screen, but the video capture hardware is not adjustable. Anyway, the demo video shows the speed of the hardware handling scrolling and displaying of data. While the hardware can handle scrolling and displaying reasonably fast, the monitor software is quite crude and complicated due to processor spent most of its time handling video display. I want to port EhBASIC to it, but I'll need to think about how to deal with juggling multiple tasks of video, keyboard, SD card, serial port, and EhBASIC application.
Bill

Edit, This is the setup with PS2 and VGA interfaces to run the above monitor. SD card is not operational so the serial port is needed to load and run the simple monitor software. There was a layout mistake with PS2 connector, so it was assembled with 30 degree tilt so PS2 cable can be inserted.

File Attachments
1) 3V6502_simple_monitor.gif, downloaded 869 times
2) 3V6502_with_PS2_VGA_serial.jpg, downloaded 792 times

Subject: Re: Retrocomputer projects using EPM240 development board
Posted by plasmo on Sun, 04 Aug 2024 00:18:16 GMT
View Forum Message <> Reply to Message

I've made good progress bootstrapping 3V6502 using SD card. The internal flash in EPM240 is a serial flash so it is rather slow, about 5uS to access a byte. It is more efficient to copy the contents of the internal flash to RAM and execute in RAM. The internal flash contains a SD card initialization and SD read routines that copies and executes a 4K monitor program stored in known sectors in the SD card. The interface to SD card is software bit-banging, but it is fairly fast, about 100KB/sec, because 6502 is running at 25MHz.

6502's interrupt vector is located at top of its memory where internal flash is located. Because the flash access is so slow, it is necessary to page out the flash and replace with RAM to improve the interrupt response. So a page register is added in CPLD. Once flash is paged out, the memory is all zero-wait RAM except the 256-byte I/O space located at $FB00-$FBFF.

I now have a pretty decent development environment to load/run/debug applications for 3V6502. The next goal is rework the VGA/keyboard software as background tasks, so I can port applications such as EhBASIC as foreground task for the standalone 3V6502.

Bill

File Attachments
1) 3V6502_with_SD_card.jpg, downloaded 775 times

Subject: Re: Retrocomputer projects using EPM240 development board
Posted by plasmo on Wed, 07 Aug 2024 03:54:44 GMT
View Forum Message <> Reply to Message

3V6502 is working well. I want to correct various PC board mistakes on 3V6502 and 68KSEC and also layout a Z180 mezzanine board. Before I send out these designs to JLCPCB, I want to make sure 68KSEC will work with 1meg RAM and pass memory diagnostic. I'm glad I did that because I've discovered another layout error with 68KSEC in the process. This is the patched up 68KSEC that now can boot from serial port, load and run memory diagnostic of the 1 meg RAM. Current consumption of the board while running memory diagnostic is 50mA. 68SEC000 is clocked at 16.7MHz at 3.3V.

Bill

File Attachments
1) DSC_77380806.jpg, downloaded 750 times

Subject: Re: Retrocomputer projects using EPM240 development board
Posted by plasmo on Sun, 25 Aug 2024 19:02:31 GMT

View Forum Message <> Reply to Message

This is another mezzanine board for EPM240 dev board. It is 3VZ180, based on Z8S180 processor and a 1/2 meg RAM. Since Z180 is already highly integrated with its own serial ports, SPI, and memory bank registers, the EPM240 is not highly utilized. EPM240 provides the bootstrap flash and memory decode for I2C and RTC. To help software development, a serial port is temporarily added to EPM240 which allows high-speed (115200) serial interface with 25MHz or 50MHz CPU clock. This is because Z180's internal baud generator has limited capability that can't generate clock for 115200 bps with 25 or 50MHz CPU clock.

The 3VZ180 mezzanine board is designed for RomWBW. Currently the RomWBW image is loaded into RAM via the serial port. The goal is load RomWBW image from SD card to RAM using EPM240's internal flash.

The board is working at 3.3V 50MHz; current consumption is 110mA. RTC board is supported and it measure the effective CPU clock as 49.992MHz.
Bill

File Attachments

1) 50mhz3vz180_rtc.jpg, downloaded 67 times
2) DSC_77430825.jpg, downloaded 642 times

Subject: Re: Retrocomputer projects using EPM240 development board
Posted by plasmo on Mon, 26 Aug 2024 04:29:43 GMT
View Forum Message <> Reply to Message

It has been suggested that lightly loaded Z180 is capable of 70MHz operation at 3.3V. I'm skeptical, but I do have a 64MHz oscillator so I thought I give it a try. Well, it actually run at 64MHz!

Picture shows a 64MHz oscillator wired to 3VZ180 which has a DS1302 RTC module plugged in. I verified via scope that Z8S180 is driven with 64MHz clock. It boots into RomWBW and CPU clock is measured at 63.992MHz by RomWBW. I also do the mandelbrot benchmark with MBASIC which completed in 16.8 seconds, that is the correct time for 64MHz Z180 with 0 wait state RAM. Current consumption is 185mA. This is actually quite amazing.
Bill

File Attachments

1) 64mhz3vz180_rtc.jpg, downloaded 620 times
2) 3VZ180_with_external_64MHz_osc.jpg, downloaded 590 times

Subject: Re: Retrocomputer projects using EPM240 development board

Posted by wsm on Mon, 26 Aug 2024 17:12:50 GMT
View Forum Message <> Reply to Message

I agree that it's quite amazing as 64MHz at 3.3V is more than three times the Zilog specifaction of 20MHz maximum at 3.3V.

I am curious about "16.8 seconds that is the correct time for 64MHz". I use MBASIC 5.21 and my version of ASCIIART runs in 30 seconds at 33.333MHz and 27 seconds at 36.864MHz on my MinZ systems. Those systems also do some minimal processing of once-per-second RTC interrupts. By my calculations, that should correlate to 15.6 seconds at 64MHz.

Subject: Re: Retrocomputer projects using EPM240 development board
Posted by plasmo on Mon, 26 Aug 2024 23:40:36 GMT
View Forum Message <> Reply to Message

I don't have the best reflex, but I measured either 16.8 or 16.9 seconds with a stopwatch. The duration is from "run" to "ok". I'm using MBASIC under RomWBW. According to the sign on of the program it is "BASIC-80 Rev5.21". Attached is a screen shot of the run. The serial port is non-standard 50Kbps due to 64MHz clock does not divide nicely to standard baud rates. I wonder whether slow serial port can have an impact on run time. Seems unlikely...
Bill

File Attachments
1) 64MHz_asciiart_mbasic.jpg, downloaded 71 times

Subject: Re: Retrocomputer projects using EPM240 development board
Posted by wsm on Tue, 27 Aug 2024 01:19:37 GMT
View Forum Message <> Reply to Message

Yep, that appears to be exactly the same program I'm using. I added "999 SYSTEM" as the last statement and by naming it ASCIIART.BAS I can then simply enter "MBASIC ASCIIART" and it'll return to the CP/M prompt when finished.

I just tried it with the console on ASCI 0 at 57600 to a CP2105 (320 byte FIFOs) and couldn't see any noticeable difference. Using Chrome's Stopwatch, I can usually get +/- .05 second over a couple of runs and I do it from the ENTER of "MBASIC ASCIIART" to the CP/M prompt where MBASIC and ASCIIART are on an SST39SF040 flash "disk".

Hmmm ...
I don't think RomWBW's redirection would cause a roughly 7% difference when there's only about 1760 bytes of output. Perhaps it might be interesting to get the 25MHz time. I would expect that to be (33.333/25)*30 = 40 seconds.

Subject: Re: Retrocomputer projects using EPM240 development board

Posted by plasmo on Tue, 27 Aug 2024 02:44:50 GMT
View Forum Message <> Reply to Message

RomWBW is pretty complex and may have introduced some overhead. I have a simple CPM2.2 without interrupt, so I'll run the benchmark with that.

Subject: Re: Retrocomputer projects using EPM240 development board
Posted by plasmo on Sat, 31 Aug 2024 04:48:21 GMT
View Forum Message <> Reply to Message

I ran the ascii mandelbrot benchmark on a Z8S180 board with 36.8MHz clock. With my simple CP/M2.2, it is 27.1 seconds, which is similar to your MinZ result. However, with the same hardware running mandelbrot benchmark under RomWBW, it is 29.1 seconds. So RomWBW is definitely introducing some overhead. I verify RomWBW turned off the refresh and enable zero wait. I think the overhead may be due to RomWBW's 50Hz timer interrupt.
Bill

Subject: Re: Retrocomputer projects using EPM240 development board
Posted by wsm on Sat, 31 Aug 2024 22:01:16 GMT
View Forum Message <> Reply to Message

Again, that's about a 7% overhead for RomWBW on what should be a CPU-intensive program. I think you should talk to Wayne about that as it would be even more noticable at lower frequencies i.e. about 10 full SECONDS at 7.3728MHz.

The 7% at 36.864MHz is 2,580,480 * 29.1 T states for about 1.7K of ASCI output and a 50Hz timer that the documention describes as simply a tick counter. With a VERY rough estimate of 10 T states per instruction that would be an extra 7,509,197 instructions for about 3,215 events (ASCI + PRT) and likely most of that is in the Console Output BIOS routine ... possibly about an extra 4,200 instructions per byte of output.

FWIW: My MinZ reference system has a 1Hz RTC interrupt and uses IOBYTE console redirection to ASCI0. I enabled a 60Hz PRT interrupt with minimal processing and it did not make any noticeable difference in run times. Thus my assumption that the extra overhead is primarily in RomWBW's Console Output routine.

Subject: Re: Retrocomputer projects using EPM240 development board
Posted by plasmo on Wed, 04 Sep 2024 04:12:08 GMT
View Forum Message <> Reply to Message

I was poking around RomWBW CP/M BIOS and patching out the various console I/O routined and timer interrupt routine to find the impact to mandelbrot benchmark. I was surprised to find that replacing CP/M BIOS with simple console output routine and disabling 50Hz interrupt have no measurable impact, but replacing RomWBW's CP/M BIOS console status with simple console

status routine trimmed off 2 seconds from mandelbrot benchmark, so the benchmark went from 29.1 seconds to 27.1 seconds which is roughly same as the benchmark on your MinZ.

RomWBW version of CP/M BIOS does have more overhead, but not anything extraordinary, as far as I can tell. Now I wonder whether MBASIC calls console status frequently, far more than console output or 50hz interrupt, such that added overhead in console status routine can resulted in 7% degradation of performance.

Bill

Edit, Maybe I'm missing something reading Wayne Warthen's CP/M BIOS code. I'll ask him if CBIOS's console status does more works than console input or output.