This is a RC2014 board, but I believe it has wider usage than the RC2014 world.

I'm interested in a minimum system to explore the 8-bit microprocessors of 1970's and 1980's. The targeted processors are the traditional 5V microprocessors that need memory, address decode logic, and peripherals, not the integrated micro-controllers. Processors such as 680x, Z80, 808x, 6502, and even 16-bit processors with 8-bit bus such as 68008 and 32008. I want a common development platform that supplies the memory and logic and basic peripheral so I can just concentrate on the processors. I also don't want to burn EPROM over and over again, it is a pain as we all know.

The development platform will revolve around a 40 or 48-pin 600-mil socket for processors. Since these processors all have different pinout, including power & ground, the 40/48 pin socket is completely uncommitted. Another word, there will always be 40-48 fly wires, but the rest can be printed circuits. Some processors require specialized clock logic such as 8080 & 32008, so there need be extra room for the special logic.

The development platform is named G8PP for Generic 8-bit Processors Prototype. These are its features:
* ROM-less
* 512K battery-backed RAM
* Altera EPM7128 CPLD
* Serial bootstrap capability
* Banked memory
* Prototype area supporting 48-pin, 600mil wide DIP
* Compact flash interface
* RC2014 bus compatible

A 5V programmable logic (CPLD) is needed to handle different control signals with different timing requirements and most 5V CPLD can handle that adequately. However, unlike 3V FPGA, 5V CPLD of 1990's are not very complex, it does not have initializable RAM that can serve as boot ROM. So a ROM-less design requires a way to load boot ROM into nonvolatile RAM and protect the bootstrap code so it is not overwritten in normal usage. I am inspired by Z280's UART bootstrap method and come up with a similar concept in CPLD. It is basically a pre-configured serial port with simple DMA and a bank memory scheme to protect the bootstrap code. This serial bootstrap mode, when enabled at power up, will receive a pre-determined number of serial data and store them in designated RAM locations. The support logic will tri-state the target processor's bus during serial bootstrapping. When the pre-determined number of data have received, the support logic will release the target processor to run. The bootstrap software will load a more sophisticated bootstrap software which will copy itself into a banked memory region that becomes the boot memory region when reset or power up. It may sound complicated, but it easily fit in a modest CPLD like EPM7128 and have spare logic for other functions. It will even fit the smaller EPM7064 which is interesting from hobbyist build-able point of view.

First photo is the picture of G8PP without processor. The 2nd & 3rd pictures are G8PP with Z80.
There are about 50 manual wires. The Z80 is overclocked to 22MHz only because I don't have a 20MHz clock right now. It passed ZEXALL.COM, so I'm confident it is working properly. It runs CP/M2.2 as well as CP/M3. The CF interface is surprisingly robust. In the picture is a particular brand of CF disk that I have problem with most of my SBC, but it works just fine in this setup.

I want to do 8085 next, but I don’t have the chip yet, so I’ll work on 68008 next.

Bill

File Attachments
1) DSC_40691207.jpg, downloaded 621 times
2) DSC_40781208.jpg, downloaded 540 times
3) DSC_40801208.jpg, downloaded 526 times

Subject: Re: Generic 8-bit Processors Prototype Board
Posted by plasmo on Sun, 09 Dec 2018 18:09:18 GMT
View Forum Message <> Reply to Message

I have started a web page on G8PP, but it is pretty sparse. I'll fill it up with design information as I go.
https://www.retrobrewcomputers.org/doku.php?id=builderpages:plasmo:g8pp

Subject: Re: Generic 8-bit Processors Prototype Board
Posted by mikemac on Mon, 10 Dec 2018 20:49:07 GMT
View Forum Message <> Reply to Message

Cool!

Subject: Re: Generic 8-bit Processors Prototype Board
Posted by b1ackmai1er on Mon, 10 Dec 2018 23:10:10 GMT
View Forum Message <> Reply to Message

Nice work Bill.

What is that type of wire called? I’ve see it around a lot -t gives a lot needer finish than multistrand hookup wire.

Regards Phil

Subject: Re: Generic 8-bit Processors Prototype Board
Phil,
They are 30 ga wirewrap wires, commonly referred as kynar wire. They come in all different color. I bought a big spool of it 10+ years ago that should last my life time.

I was curious to find how fast I can overclock the 20MHz Z80. It is flaky at 28MHz at 5V, but if I raise the voltage to 5.5V, it will run reliably at 29.5MHz at room temperature. The ASCIIART benchmark runs in 39 seconds, and the ZEXALL.COM completes in 26 minutes 25 seconds and 100% pass. Running CP/M at 29.5MHz definitely has a snap to it. The current consumption for the board is 250mA at 29.5MHz. The CPLD is warm, but everything else is cool. Pretty amazing.

a>mbasic80 asciiart.txt
BASIC-80 Rev. 5.21
[CP/M Version]
Copyright 1977-1981 (C) by Microsoft
Created: 28-Jul-81
32824 Bytes free
00000011111111111111111122222233347E7AB32222211110000000000000000000000000000 00
00000111111111111111112222222333557BF754332222111110000000000000000000000000 00
0001111111111111111122222233347E7AB32222211110000000000000000000000000000 00
001111111111111111112222223334456C 64333222211111000000000000000000000000000000
011111111111111111112222223334456C 654433322211111100000000000000000000000000000
11111111111111111111222222333469 D978 BCF DF9 6556F4221111110000000000000000000000
111111111111111111112222223334469 D 632211111100000000000000000000000000000
111111111111111111112222223334457DB 853211111100000000000000000000000000000
111111222234B744444455556A 96532211111100000000000000000000000000000
12222233347BAA7B776679 A3221111110000000000000000000000000000
2222223334567 9A A5322211111110000000000000000000000
2222223334679 94322211111110000000000000000000000
234445568 F B54322211111110000000000000000000000
234445568 F 86432211111110000000000000000000000
234445568 F 94322211111110000000000000000000000
2222223334679 94322211111110000000000000000000000
2222223334567 9A A5322211111110000000000000000000000
12222233347BAA7B776679 A32211111110000000000000000000000000000
111111122223333334457DB 853211111110000000000000000000000000000
11111111111111222222333334469 D 632211111100000000000000000000000000000
11111111111111111111222222333469 D978 BCF DF9 6556F4221111110000000000000000000000
11111111111111111111222222333469 D632211111100000000000000000000000000000
a>zexall
Z80 instruction exerciser
<adc,sbc> hl,<bc,de,hl,sp>.... OK
add hl,<bc,de,hl,sp>........ OK
add ix,<bc,de,ix,sp>......... OK
add iy,<bc,de,iy,sp>......... OK
aluop a,nn.................. OK
aluop a,<b,c,d,e,h,l,(hl),a>.. OK
aluop a,<ixh,ixl,iyh,iyl>..... OK
aluop a,(<ix,iy>+1)......... OK
bit n,(<ix,iy>+1)........... OK
bit n,<b,c,d,e,h,l,(hl),a>.... OK
cpd<r>........................ OK
cpi<r>........................ OK
<daa,cpl,scf,ccf>............ OK
<inc,dec> a................... OK
<inc,dec> b................... OK
<inc,dec> bc................... OK
<inc,dec> c................... OK
<inc,dec> d................... OK
<inc,dec> de................... OK
<inc,dec> e................... OK
<inc,dec> h................... OK
<inc,dec> hl................... OK
<inc,dec> ix................... OK
<inc,dec> iy................... OK
<inc,dec> l................... OK
<inc,dec> (hl)................ OK
<inc,dec> sp.................. OK
<inc,dec> (<ix,iy>+1)........ OK
<inc,dec> ixh................ OK
<inc,dec> ixl................ OK
<inc,dec> iyh................ OK
<inc,dec> iyl................ OK
ld <bc,de>,(nnnn)......... OK
ld hl,(nnnn)............... OK
ld sp,(nnnn)................. OK
ld <ix,iy>,(nnnn)......... OK
ld (nnnn),<bc,de>........... OK
ld (nnnn),hl............... OK
ld (nnnn),sp............... OK
ld (nnnn),<ix,iy>........... OK
ld <bc,de,hl,sp>,nnnn...... OK
ld <ix,iy>,nnnn............. OK
ld a,<bc),(de)>........... OK
I'm taking a different approach to the G8PP: Instead of manually wiring up a G8PP board for every processor, I decided I should emulate the RC2014 approach where there are multiple simple CPU boards and a common G8PP baseline board connected to the target CPU via a backplane.

Because each CPU has different control signals and timing requirements, the backplane signal assignments will change depending on the processor used. The addresses and data are the same, so are power/ground, clock, reset, and interrupt. The five Z-80 specific control signals (nM1, nMREQ, nRD, nWR, nIORQ) and four spare signals of RC2014 are re-assigned depending on the specific processor. Each processor board is quite simple consists of unbuffered CPU signals connected directly to the 40-pin RC2014 connector.

The common board is the G8PP in baseline configuration. It provides the RESET, clock, battery-backed 512K paged RAM, serial port with bootstrap function, compact flash interface and an Altera EPM7128S CPLD. It also has a prototype area to accommodate additional functions.
such as RTC, SPI, I2C, etc. The CPLD can be reprogrammed to support different processor's control signals. Such two-board solution has enough functionalities to get CP/M running for the Z80/8085/68008 processors.

As I gained more experience, the baseline configuration of G8PP should not have any manual wirings. The five RC2014 control signals plus the 4 spare pins are reassigned different signals depending on the processor card in use. Reprogramming the CPLD should handle the different signal assignments.

What I hope to achieve when changing processor is
* Reprogram the CPLD on G8PP,
* Change oscillator,
* Plug in the targeted processor card,
* Serially load bootstrap associated with the processor,
* Load applications and run.
* No manual wiring changes and no EPROM burning.

The goals for Z80, 8085, and 68008 are CP/M. For 6502 it is Lee Davison’s EhBASIC. I don’t know what the goals are for 1802, 6809, and 32008. Any suggestions?

Bill

Edit, In the first picture (6502 CPU), the G8PP looks different because the CF interface was not installed when the picture was taken.

More edits: The attached pictures are:
1. 6502 with G8PP running at 3.68MHz
2. Z80 with G8PP running at 14.7MHz
3. 8085 with G8PP at 7.37MHz (processor speed is 1/2 of clock or 3.68MHz)
4. backside of the baseline G8PP
5. backside of 6502 CPU card

Subject: Re: Generic 8-bit Processors Prototype Board
Posted by etchedpixels on Mon, 07 Jan 2019 18:32:57 GMT
View Forum Message <> Reply to Message

For 6809 there are a bunch of options

- Grant's 6809 machine has a Microsoft BASIC akin to the Z80 setup
- CamelForth 6809 seems another popular environment

- FLEX was the CP/M of the 6800/6809/68HC11 world. It's in some ways neater and in other ways brain-numbingly dumb but like CP/M has a ton of useful software ready to go.

- OS/9 (nowdays NitrOS/9) is what real people run. Can use all the banking out of the box, mostly realtime. Rather a bigger project though.

- There's also CUBIX: http://www.classiccmp.org/dunfield/d6809/index.htm, which is Dave Dunfield's homebrew OS for his own build 6809 machine and which has since been ported to other stuff like multicomp09. It's an amazingly tiny OS (smaller than CP/M).

For the 1802 there is a tinybasic and also IDIOT (a tiny monitor in 512 bytes) from the Membership card 1802 plus other stuff

See: http://www.retrotechnology.com/memship/mship_soft.html

On the 32008 I have no idea. All the 320xx software I've seen has either been Unix systems or Acorn co-processor software nothing standalone. There were 320xx S100 boards but I've no idea what anyone actually did with them

Alan

Subject: Re: Generic 8-bit Processors Prototype Board
Posted by plasmo on Mon, 07 Jan 2019 19:17:12 GMT

Alan,
Thanks for the suggestion on 6809 and 1802. FLEX for 6809 seems a good goal. 1802 is slow so a tinybasic may be all it can do.
There is a website on NS32xxx, http://cpu-ns32k.net, it has cross assembler and examples of 32xxx systems. I hope to draw some inspirations from there.
I'm setting aside 32008 and 1802 for next iteration, I'm turning the 6809, 68008, Z80, 6502 and 8085 designs into printed circuit boards. They can be optionally plugged directly into a G8PP board without the RC2014 backplane.

Bill

Subject: Re: Generic 8-bit Processors Prototype Board
Posted by plasmo on Wed, 23 Jan 2019 18:37:24 GMT

Alan,
Thanks for the suggestion on 6809 and 1802. FLEX for 6809 seems a good goal. 1802 is slow so a tinybasic may be all it can do.
There is a website on NS32xxx, http://cpu-ns32k.net, it has cross assembler and examples of 32xxx systems. I hope to draw some inspirations from there.
I'm setting aside 32008 and 1802 for next iteration, I'm turning the 6809, 68008, Z80, 6502 and 8085 designs into printed circuit boards. They can be optionally plugged directly into a G8PP board without the RC2014 backplane.

Bill
An update of G8PP progress:

I made pc boards for 5 processors: Z80, 8085, 6502, 6809, and 68008. The processor pc boards have an extra 40-pin female header so a G8PP board can plug directly into one of the 5 processor boards without a RC2014 backplane. Attached photo is G8PP + Z80 without the RC2014 backplane. G8PP + Z80 is in good shape. It is running at 22MHz with CP/M 2.2 and CP/M 3. The wiki page is here:

https://www.retrobrewcomputers.org/doku.php?id=builderpages:plasmo:g8pp:g8ppbasez80

Bill

---

Subject: Re: Generic 8-bit Processors Prototype Board
Posted by plasmo on Wed, 23 Jan 2019 18:45:30 GMT
View Forum Message <> Reply to Message

This is G8PP plugs directly into a 68008 CPU card. CP/M68K is running on G8PP+68K8. However, it is slow compare to Tiny68K. This is because 68008 data bus is 8-bit instead of 16-bit and the compact flash interface is 8-bit instead of 16-bit. Screenshot of CP/M68K booting up, compiling & running "Hello World" C program. The wiki page is here:

https://www.retrobrewcomputers.org/doku.php?id=builderpages:plasmo:g8pp:g8ppbase68k8

---

File Attachments
1) DSC_41870123.jpg, downloaded 345 times

---

Subject: Re: Generic 8-bit Processors Prototype Board
Posted by plasmo on Wed, 23 Jan 2019 19:18:16 GMT
View Forum Message <> Reply to Message

This is G8PP plugs directly into a 68008 CPU card. This particular CPU gave me quite a bit of troubles. Eventually I figured out that it is NOT 10MHz W65C02S8P, but it is a variant of 6502 that run at 3.68MHz(another word, it is a fake part). This particular variant has no tri-state control so in order to transfer data from serial port to RAM, the CPLD has to overdrive the address lines. Fortunately, the CPLD of 1990's has more drive capability than the 6502 of 1970's, furthermore, I added 160 ohms series resistor to every address line from 6502. This is why there are 16 resistors on board which would not have been necessary if this is a real W65C02S8P. Even with a fake 6502 running at 3.68MHz, it is able to run Lee Davison's EhBASIC and ran the mandelbrot benchmark in 5 minutes and 55 seconds which is the correct time for 3.68MHz part.
6502 EhBASIC [C]old/[W]arm?

Memory size?

48383 Bytes free

Enhanced BASIC 2.22p4

Ready
10 MAXITER=20
20 LET C$=",.~!^:<&?oxOX# "
30 FOR Y=-39 TO 39
40 FOR X=-39 TO 39
50 CREAL=X/20
70 CIMAG=Y/20
80 ZREAL = CREAL
90 ZIMAG = CIMAG
95 COUNT = 1
100 ZM = ZREAL*ZREAL
105 ZN = ZIMAG*ZIMAG
107 ZL = ZM+ZN
110 IF ZL>4 THEN GOTO 170
120 ZR2=ZM-ZN+CREAL
130 ZIMAG=ZREAL*ZIMAG*2+CIMAG
140 ZREAL=ZR2
150 COUNT=COUNT+1
160 IF COUNT<MAXITER THEN GOTO 100
170 PRINT MID$(C$,1+COUNT,1);
180 NEXT X
185 PRINT ""
190 NEXT Y
200 END
RUN

...............................,,,,,,,,,,,,,,,,,...............................
...........................,,,,,,,,,,,,,,,,,,,,,,,,,...........................
........................,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,........................
......................,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,......................
....................,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,....................
..................,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,~~..................
.................,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,';~~'................
................,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,'~~~''~................
..............,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,'~~~''~,..............
.............................,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
..................................................................................
I like the combination of G8PP+processor. It makes a nice, compact system that feels like a re-configurable single-board computer.

I have not made quite as much progress with the remaining 2 CPU, 8085 and 6809. 8085 has a working bootstrap and monitor/debugger, but I'm having problems porting Z80 CP/M 2.2 to it. The differences between Z80 instructions and 8085 are small, but enough to trip me up repeatedly. A friend offers to take over the effort, I gladly passed it on.

I'm able to get 6809 to bootstrap. That's all I have for now. The wiki pages are here:

https://www.retrobrewcomputers.org/doku.php?id=builderpages:plasmo:g8pp:g8ppbase8085
https://www.retrobrewcomputers.org/doku.php?id=builderpages:plasmo:g8pp:g8ppbase6809

Bill
Subject: Re: Generic 8-bit Processors Prototype Board
Posted by mikemac on Wed, 23 Jan 2019 21:37:38 GMT

Dumb question: are you reprogramming the Altera for each processor board? It seems like the processors would be different enough that they'd require a customized FPGA load.

Subject: Re: Generic 8-bit Processors Prototype Board
Posted by plasmo on Wed, 23 Jan 2019 23:42:30 GMT

mikemac wrote on Wed, 23 January 2019 14:37:38

Dumb question: are you reprogramming the Altera for each processor board? It seems like the processors would be different enough that they'd require a customized FPGA load.

Yes the CPLD needs to be reprogrammed for different processor. Depending on the processors, there are up to 8 control signals on the RC2014 bus that need to be redefined.

Bill

Edit: Furthermore, different bootstrap software needs to be uploaded into RAM followed by different application software. CF disk needs to swap out for different operating systems.

Subject: Re: Generic 8-bit Processors Prototype Board
Posted by mikemac on Thu, 24 Jan 2019 15:04:58 GMT

I know you're a fan of the 7000 series but maybe this is a place for one of the FPGAs that has an external image flash. That way you could put the customized FPGA image on the processor board. Then when you swapped processor boards, you'd automatically swap in the right FPGA code.

You could put the CF connector on the processor board too but I think that would be overkill. You're probably going to be playing with what code is on the CF during normal usage, so you'll be swapping it in and out often anyway. The FPGA firmware probably doesn't change much once you have the initial version working.
Sometimes I wish I had used FPGA, so I don’t have to count gates and stingy with my resources. I use EPM7128SQC100 mainly because I have so many of them on hand. It is also because most FPGA are 3.3V devices. The comments I received are about making the board build-able, such as using PLCC84 version of EPM7128. I’m revising the G8PP with PLCC CPLD right now.

Bill

Nice work.

Would it be possible to put an attiny or similar and an serial eprom on board with all the configurations so it could be reprogrammed on the fly at board level?

Regards Phil

I believe Altera has published the details of their programming format and Altera JAM STPL players source are publicly available. Having said that, the Altera programmer is readily available for lesser than $5 and the CPLD is easily programmed in-situ with the standard 2x5 programming header on G8PP. It only has to programmed once for a given processor, although I must have programmed that particular G8PP board 100+ times to work out all the hardware bugs for the various processors.

Bill

I like the idea of RC2014-like expansion connectors. G8PP+CPU in current form can easily have
3 expansion connectors. So here is the idea:
* CPU board serves as the anchor with two expansion connectors on either side of the CPU and two more horizontal connectors
* G8PP plugs to one long side of the CPU board with another expansion connector
* A second processor with its own expansion connector plugs into the other long side of the anchor CPU board. It should be fun to tinker with multiprocessing.

G8PP+CPU re-configurable motherboard, quite a mouthful. Need to come up with a better name.

Bill

---

Subject: Re: Generic 8-bit Processors Prototype Board
Posted by plasmo on Tue, 03 Nov 2020 01:41:37 GMT
I've updated G8PP base board. This is updated G8pp with 6502 processor card.

Bill

---

Subject: Re: Generic 8-bit Processors Prototype Board
Posted by ale500 on Tue, 03 Nov 2020 09:13:12 GMT
Your board looks very neat, I like how you use 90° angles for the traces. I think there are no problems at these "low" frequencies.
I had a similar idea, after many re-designs I decided to stay in a very simple 16 bit version: RAM&ROM and '245s for the data BUS and '573 for the address BUS. But without any CPLD. I have to post a picture. I haven't populated any yet.