Subject: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by plasmo on Wed, 02 Aug 2017 00:27:58 GMT
View Forum Message <> Reply to Message

A decade ago I purchased a box of salvaged single board computers based on Motorola MC68302. They were made by ADC Communication as a controller for their Soneplex communication product line. I figured out most of the wirings, wrote a monitor and built a few home projects using these boards. With my recent experiences of CP/M 68K I realize the board also has enough memories and resources to run CP/M.

The board has 3 banks of flash memories. The first bank is 256K bytes and hold the monitor. The other two banks are 1 megabytes each. Collectively they formed a read-only disk to hold the CP/M 68K ver 1.3 distribution files. The RAM is 1 megabyte with the top 256K backed up with a super capacitor. The backed up RAM may serve as a small RAM disk. CP/M 68k resides in $15000 - $1FFFF and the Transient Program Area is from $20000 to BFFFF, or 640K--which "ought to be enough for anybody"
I copied the CP/M distribution files into flash and configured the BIOS for Flash disk and RAM disk and this is what I have:

The 256K RAM disk is obviously too small to be really useful. There is a 82C55 on board and all it's outputs are available on the 96-pin DIN 41612 connector. It should not be difficult to add a CompactFlash adapter for the DIN connector. Available on the DIN connector are also a number of discrete I/O, 4 serial ports and a SPI. The processor clock is 16MHz. This is a fairly capable 68000 single board computer. Periodically they are available on eBay for $15-$20 like this one: http://www.ebay.com/itm/122306457847

Disclaimer: I do not know the above seller and haven't bought anything from him. It appears to be the same board as mine, but I don't know whether it will work exactly like mine or not.

File Attachments
1) adc mpu_component side_F2.jpg, downloaded 2825 times
2) cpm_on_adc_mpu.jpg, downloaded 2820 times

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by computerdoc on Wed, 02 Aug 2017 02:20:22 GMT
View Forum Message <> Reply to Message

Hi Plasmo,
That is a pretty cool looking board! Congratulations on getting CP/M 68K running on it. How did you get the CP/M 68K software into the flash? Do you have some development software for this board? What have you done with it? Will you be creating a wiki entry for this board and software you have adapted to it? I'd like to read up on this.

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board

Posted by plasmo on Wed, 02 Aug 2017 04:18:13 GMT
View Forum Message <> Reply to Message

I wrote about some of the projects I've done here: http://www.easy68k.com/EASy68Kforum/viewtopic.php?f=10&t =1570 I also have a Xbee network around the house and garden using these boards. These boards were cheap & plentiful on eBay 5-10 years back as they were scrapped. Perhaps few people here have bought some and interested in how to re-use them. I'll create a wiki page about the board and how to program custom software into it. It is a brute-force approach, I'm hoping someone more familiar with the board design can provide board design information & better way to upload new software.

CP/M 68K software into flash is a bit complicated. This is the condensed version: Modify the BIOS so CP/M68K will run in the EASy68K environment. Use EASy68K's file I/O calls to read in the image of CPM distribution files which is created with cpmtools. Run CP/M in EASy68K simulator, convert .rel command files to .68k and get rid of unnecessary files so they'll fit in 1.8meg of disk space. Save the modified disk image and convert them to S-record. In the actual hardware, load the S-record and program the flash. Now the disk image is in the flash, load the CPM15000.SR & associated BIOS and run.

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by norwestrzh on Thu, 03 Aug 2017 17:36:53 GMT
View Forum Message <> Reply to Message

Hi Plasmo,
Interesting! My 68k SBCs have operational CF, with the CPU running as fast as 12 MHz. They seem to be working fine, but I can't get the Digital Research BASIC compiler (CB68) to work reliably. I can compile something simple like:
FOR I = 1 TO 50
IF I < 5 THEN GOTO 50
IF I > 45 THEN GOTO 50
PRINT I
50 NEXT I
to compile and run OK, but not something more complex like your "ASCII art" program (I get $03 exception).
I'm wondering if you have ever tried CB68 on this platform, or one of your others? Can you get it to work?
Roger

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by Andrew B on Thu, 03 Aug 2017 18:15:34 GMT
View Forum Message <> Reply to Message

Picked up one of these boards for $20 + $10 S/H off eBay, so there are still a few around anyway.

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by plasmo on Fri, 04 Aug 2017 18:07:29 GMT
View Forum Message <> Reply to Message

norwestrzh wrote on Thu, 03 August 2017 10:36Hi Plasmo,
Interesting! My 68k SBCs have operational CF, with the CPU running as fast as 12 MHz. They seem to be working fine, but I can't get the Digital Research BASIC compiler (CB68) to work reliably. I can compile something simple like:
FOR I = 1 TO 50
IF I < 5 THEN GOTO 50
IF I > 45 THEN GOTO 50
PRINT I
50 NEXT I
to compile and run OK, but not something more complex like your "ASCII art" program (I get $03 exception).
I'm wondering if you have ever tried CB68 on this platform, or one of your others? Can you get it to work?
Roger
Roger,
The "ASCII art" was produced by EhBasic which may have different syntax than CB68.
I am able to download the binary for cb68 on cpm.z80.de, but I can't find the programmer's guide for the CP/M68K version of CB68. Could you give me a quick tutorial on how to compile & run CB68 programs? I can run your short program on my board and then a longer program that didn't work for you and compare results.

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by norwestrzh on Fri, 04 Aug 2017 23:26:41 GMT
View Forum Message <> Reply to Message

>> The "ASCII art" was produced by EhBasic which may have different syntax than CB68.

Yes, I know. I got EhBASIC going on my 68k, and was able to benchmark the ascii.art thing (very nice trial to test out various compilers and assemblers, BTW!

>> ...but I can't find the programmer's guide for the CP/M68K version of CB68

Same here. I think I found a guide for CP/M 80 and 86, but nothing for 68k. I can tell you what I did, but given my results, I may be doing something very wrong! *grin*

Just create a file with your BASIC code, let's say "ASCII.BAS". Their BASIC compiler appears to be very flexible as far as syntax is concerned. Line numbers are only required on lines that you reference in your code, and feel free to indent and structure your code as you see fit. CB68 doesn't seem to mind.

Then:
CB68 ASCII.BAS
and:

LINK68 ASCII.O, CB68.L68
that ought to produce a .68K file that (theoretically) can be run from the command line.
[Note that CB68.L68 is (I gather) the link library for the thing.]

Thanks for looking at this. I'm *very* interested in your results!

Roger

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by norwestrzh on Fri, 04 Aug 2017 23:31:03 GMT
View Forum Message <> Reply to Message

>> The "ASCII art" was produced by EhBasic which may have different syntax than CB68.

Yes, I know. I got EhBASIC going on my 68k, and was able to benchmark the ascii.art thing [very nice trial to test out various compilers and assemblers, BTW!]

>> ...but I can't find the programmer's guide for the CP/M68K version of CB68

Same here. I think I found a guide for CP/M 80 and 86, but nothing for 68k. I can tell you what I did, but given my results, I may be doing something very wrong! *grin*

Just create a file with your BASIC code, let's say "ASCII.BAS". Their BASIC compiler appears to be very flexible as far as syntax is concerned. Line numbers are only required on lines that you reference in your code, and feel free to indent and structure your code as you see fit. CB68 doesn't seem to mind.

Then:
CB68 ASCII.BAS
and:
LINK68 ASCII.O, CB68.L68
that ought to produce a .68K file that (theoretically) can be run from the command line.
[Note that CB68.L68 is (I gather) the link library for the thing.]

Thanks for looking at this. I'm *very* interested in your results!

Roger

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by norwestrzh on Fri, 04 Aug 2017 23:37:31 GMT
View Forum Message <> Reply to Message

>> The "ASCII art" was produced by EhBasic which may have different syntax than CB68.

Yes, I know. I got EhBASIC going on my 68k, and was able to benchmark the ascii.art thing [very nice trial to test out various compilers and assemblers, BTW!]

>> ...but I can't find the programmer's guide for the CP/M68K version of CB68

Same here. I think I found a guide for CP/M 80 and 86, but nothing for 68k. I can tell you what I did, but given my results, I may be doing something very wrong! *grin*

Just create a file with your BASIC code, let's say "ASCII.BAS". Their BASIC compiler appears to be very flexible as far as syntax is concerned. Line numbers are only required on lines that you reference in your code, and feel free to indent and structure your code as you see fit. CB68 doesn't seem to mind.

Then:
CB68 ASCII.BAS
and:
LINK68 ASCII.O, CB68.L68
that ought to produce a .68K file that (theoretically) can be run from the command line.
[Note that CB68.L68 is (I gather) the link library for the thing.]

Thanks for looking at this. I'm *very* interested in your results!

Roger

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by norwestrzh on Fri, 04 Aug 2017 23:40:14 GMT
View Forum Message <> Reply to Message

Sorry for the duplication -- the site is doing something very strange!

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by plasmo on Sat, 05 Aug 2017 02:07:11 GMT
View Forum Message <> Reply to Message

Roger,
Good to hear you are having problem with the website as well. I've had an intermittent internet connection last 2-3 days due to weather and I thought the problem was with my connection.

Thanks for your instruction on how to compile and run cb68.

I used EASy68K simulator to convert cb68.rel to cb68.68k and created a second flash drives, drive B, that contains cb68.68k, cb68.l68, and cb68.doc. I only have 2 meg of flash on the ADC SPX-MPU board, so I'm trying to save space. So now drive A contains CP/M 68K distribution files, drive B contains the cb68 files, and drive C is the 256K RAM drive

The next 2 screen shots shows the content of drive B, compiling your short basic program with cb68, and link it, and then run it. (you are correct that output of the link68 is a .68k executable)

I need to change the original ascii art code so statement 110 is now:
110 if (a*a+b*b)>4 then goto 200

With that change, I'm able to compile, link, and run it. See next 2 screen shots.
It takes 104 seconds to display the ascii art. This is compiled cb68 code running on 16MHz 68302, 16-bit wide data bus and zero wait state RAM. Compare that to 102 second on 8MHz 68000 running Lee Davison's EhBasic interpreter.

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by plasmo on Sat, 05 Aug 2017 02:11:22 GMT
View Forum Message <> Reply to Message

The internet connection is too flaky to upload pictures. It'll have to wait for another day.

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by plasmo on Sat, 05 Aug 2017 04:29:42 GMT
View Forum Message <> Reply to Message

Here are the two screen shots of the short.bas compiling, linking & executing

Here are the two screen shots of the asciiart.bas compiling, linking & executing

File Attachments
1) short_1.jpg, downloaded 2613 times
2) short_2.jpg, downloaded 2497 times
3) asciiart_1.jpg, downloaded 2586 times
4) asciiart_2.jpg, downloaded 2546 times

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by norwestrzh on Sat, 05 Aug 2017 17:18:02 GMT
View Forum Message <> Reply to Message

Hmmm .....

Here's what happens when I try that:

```
10A>cb68 plasmo.bas
------------------------------------------------
CB68 CBASIC Compiler                Version 1.0
Serial No. 3123-0000-000061    All Rights Reserved
Copyright (c) 1983          Digital Research, Inc.
------------------------------------------------
end of pass 1
end of pass 2
    1:     10 for y=-12 to 12
    2:     20 for x=-39 to 39
    3:     30 ca=x*0.0458
    4:     40 cb= y*0.08333
    5:     50 a=ca
    6:     60 b=cb
    7:     70 for i=0 to 15
    8:     80 t=a*a-b*b+ca
    9:     90 b=2*a*b+cb
   10:     100 a=t
   11:     110 if (a*a+b*b)>4 then goto 200
   12:     120 next i
   13:     130 print " ";
   14:     140 goto 210
   15:     200 if i>9 then i=i+7
   16:     205 print chr$(48+i);
   17:     210 next x
   18:     220 print
   19:     230 next y
end of compilation
no errors detected
code area size:    738       000002e2h
data area size:    88        00000058h
common area size:  0         00000000h
symbol table space remaining: 49687

10A>link68 plasmo.o,cb68.l68
------------------------------------------------
LINK68 Overlay Linker               Release 0.f
Serial No. XXXX-0000          All Rights Reserved
Copyright (c) 1983          Digital Research, Inc.
------------------------------------------------

plasmo.o,cb68.l68

10A>plasmo


Exception $03 at user address $000205E0.  Aborted.
10A>
```

How are you building the CB68.REL file in the simulator? I thought that RELOC had to use a memory configuration exactly like the run time environment.

Just for grins -- here are some ASCII art run times:

MBASIC(interpreter) (Z80 @ 10 MHz) 1:54
MBASIC(interpreter) (FPGA Z80 :46 (equiv. to ~25 MHz)]
CB68(compiled) (68k @ 6 MHz) 4:23 <== I can get it to work on one of my slower 68ks
EhBASIC(interpreter) (68k @ 12 MHz) 0:55
EhBASIC(interpreter) (CP/M 68k @ 12 MHz) 1:26
EhBASIC(interpreter) (68k @ 10 MHz) 1:06
EhBASIC(interpreter) (68010 @ 10 MHz) 1:02
EhBASIC(interpreter) (8-bit strapped 68EC00 @ 10MHz) 2:55
GCC(compiler) (68k @ 12 MHz) 0:20

I've got the GCC toolchain running on my Linux box, and it seems to generate very efficient 68k code. Don't know why I worry about CB68, but it just bugs me!

Roger

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by plasmo on Sat, 05 Aug 2017 18:25:00 GMT
View Forum Message <> Reply to Message

Roger,
The BIOS for the EASy68K simulator is same as the BIOS running in the ADC MPU board. I used EASy68K trap #15 service calls for console I/O for both BIOS. Drives A, B, C are all mapped to same areas in memory. The TPA is the same (starts at 0x20000, length of 0xA0000). So whatever I see in the simulator, I should see in the hardware. I just ran the ascii art in simulator and captured the disk image of the program from 0x20000 to 0x26000 (program is 23K). It is attached below.
Assuming your TPA starts at 0x20000, the data at location 0x205E0 should be 0x00010042 06AA0000 00010046 60D44E75 I think these information may be useful with DDT debugger, then again, I never used DDT before so I can be very wrong!

I can upload the BIOS if you are interested. I'm starting a retrobrew wiki page on the SPX-MPU board and eventually all the software will be uploaded there, but it is a work-in-progress.
https://www.retrobrewcomputers.org/doku.php?id=builderpages: plasmo:spx-mpu

Thanks for you benchmark results. The EhBASIC running on your 68k @ 12MHz is fast! By my calculation if I replace the 8Mhz with 12MHz on my 68000, the benchmark time should be 68 seconds vs your 55 seconds. That's quite a difference for running the same program. The only difference I can think is the trap #15 services for console I/O and more efficient serial port.

File Attachments

1) asciiart.zip, downloaded 499 times

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by wsm on Sat, 05 Aug 2017 18:40:34 GMT
View Forum Message <> Reply to Message

Just for giggles, I tried that out on my 33MHz Z8S180. For reference:

MBASIC 4.51 interpreter - 0:32
MBASIC 5.21 interpreter - 0:30
BASCOM compiler - 0:12

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by plasmo on Sat, 05 Aug 2017 18:45:52 GMT
View Forum Message <> Reply to Message

Andrew B wrote on Thu, 03 August 2017 11:15Picked up one of these boards for $20 + $10 S/H off eBay, so there are still a few around anyway.

There seem to be some interests with the ADC SPX-MPU boards (A number of boards were purchased off eBay in the last few days) so I started a wiki page about the SPX-MPU board. As I walked through the steps in repurposing the board, I realized the process is somewhat complex and software is certainly klugy (it was my first 68K assembly program after 25+ years hiatus). I fear it would be discouraging to people wanting to repurpose their boards, so I want to make this offer to retrobrew forum members: I will repurpose your boards and install CP/M68K free of charge if you would pay for shippings both way. I live in New Mexico, USA. Please PM me for shipping details. In the meantime I'll continue to work on the wiki page so you'll have all the info I have on SPX-MPU.
Bill Shen

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by norwestrzh on Sun, 06 Aug 2017 03:43:29 GMT
View Forum Message <> Reply to Message

Hi Plasmo --

What is that .zip file? I wanted to try it out on my SBC, but when I un-zipped it and got asciiart.bin, I discovered that it isn't a bin file at all. Seems to have some directory entries at the beginning, a Digital Research copyright embed after that, and then quite a distance down in the file, some references to ASCIIART.BAK, -.BIN, -.BAS, and -.O !! Can't figure out how to extract the goodies.

If you can generate an S-record file (SENDC68) for it, that would be great (for me anyway). Loading and running at $20000 is fine.

Thanks.

Roger

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by norwestrzh on Sun, 06 Aug 2017 03:47:37 GMT
View Forum Message <> Reply to Message

>> Just for giggles, I tried that out on my 33MHz Z8S180. For reference:

>> MBASIC 4.51 interpreter - 0:32
>> MBASIC 5.21 interpreter - 0:30
>> BASCOM compiler - 0:12

Thanks, Bill. I'll add those times to my [growing] list. 33MHz Z8S180 is a hot number!!!

[Not sure if my MBASIC runs were ver. 4.51 or 5.21. Gotta' check that.]

Roger

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by plasmo on Sun, 06 Aug 2017 05:20:52 GMT
View Forum Message <> Reply to Message

Roger,
That bin file was the image of asciiart.68k when it was loaded into TPA. I thought you may have memory corruption issues (location 0x205E0 in particular), so that was a correctly working image for comparison.

It occurs to me now that perhaps the problem is not loading of asciiart.68k into TPA, but the program asciiart.68k itself. Perhaps something is wrong with link68 so the output of asciiart.68k is not correct. So attached is the zipped asciiart.68k

You mentioned that you can use s-record of asciiart.68k (as generated by sendc68). I also attach the srecord file (asciiart_srecord.txt). Out of curiosity, how do you load and execute this s-record in the CP/M environment?
Bill

Edit, I noticed you did all the compiling and linking as user 10. Have you try it as user 0?

File Attachments
1) asciiart_68k.zip, downloaded 468 times

2) asciiart_srecord.TXT, downloaded 495 times

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by norwestrzh on Sun, 06 Aug 2017 19:08:47 GMT
View Forum Message <> Reply to Message

Hi Bill --

Thanks for the asciiart files. I'll try them ASAP. We both use the same starting address for TPA [$20000], so I anticipate that all will be fine, but we'll see!

>> how do you load and execute this s-record in the CP/M environment?

I wrote a little program to load S-records...well, actually I swiped the S-record loader code from my monitor (modified version of the Antonakos monitor), and wrote some CP/M code around it to save the file to "disk". Want a copy? Is it permissible to post something like that here -- it is about 500 lines of code? [I feel compelled to warn you that I'm terrible at 68k assembler coding! *grin*]

>> I noticed you did all the compiling and linking as user 10. Have you try it as user 0?

No. I used cpmtools to create that image of the filesystem, and user 10 was the first empty user area (other than user 0). So, I put it there. I could copy it elsewhere, even to an empty "disk", but I wanted to avoid the possibility of corruption during a copy. You are correct that the problem could be anywhere -- CB68, its link library (CB68.L68), or even LINK68. I'm trying to narrow down the list, so thanks again for the asciiart code. I'm hoping that it will help.

As an aside, user areas in a CP/M filesystem are really an artifice -- files that go in user 0 can be stored right alongside those in user 10. It is all one giant directory area. The only difference is the first byte of the directory entry (user 0 has that byte 0, and user 10 has an $a there), so I'm betting that the actual user area doesn't make a whole lot of difference. Hope to find out.

Roger

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by plasmo on Sun, 06 Aug 2017 22:46:55 GMT
View Forum Message <> Reply to Message

norwestrzh wrote on Sun, 06 August 2017 12:08Hi Bill --

I wrote a little program to load S-records...well, actually I swiped the S-record loader code from my monitor (modified version of the Antonakos monitor), and wrote some CP/M code around it to save the file to "disk". Want a copy? Is it permissible to post something like that here -- it is about 500 lines of code? [I feel compelled to warn you that I'm terrible at 68k assembler coding! *grin*]

Yes, I love to have a copy of your program that saves S-record file to disk.

Since our systems have the same TPA setup, you can also send me your version of asciiart.o and asciiart.68k. I can run it on my board and compare results.

Bill

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by norwestrzh on Mon, 07 Aug 2017 00:04:16 GMT
View Forum Message <> Reply to Message

>> Yes, I love to have a copy of your program that saves S-record file to disk.

Here ya' go .... just remember that it *completely* ignores the address on the S-record. If your execution address is different than $20000, you are going to have to become familiar with DDT. You might have to anyway. *grin*

For example, if you upload an S-record file with code that needs to be at, say, $50000 [just for example], then you would have to start up DDT, and use the "r" command to read the file:
r<myfile>.bin
DDT will spit the starting and ending addresses of the file back at you. Then you do:
m<begin>,<end>,50000
and then start it up with:
g50000
That's assuming that you don't want to set any breakpoints.

If your file is a .68K file [i.e. it has valid headers], then use "e" instead of "r" to read it in.

Hope this helps. My apologies if you are a DDT pro (I'm not)!

Roger

P.S. I encountered a *ahem* small problem with your asciiart files. I think that your UART is at a different address than mine? Is it a 68681 DUART? At what address? Mine is at $700001. Maybe I can patch things up so that they would work?

File Attachments

1) srin.s, downloaded 546 times

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by norwestrzh on Mon, 07 Aug 2017 00:47:23 GMT
View Forum Message <> Reply to Message

Thinking about it, that last comment about the UART doesn't really make any sense. BASIC ought to be independent of the CONIN/CONOUT hardware, I think, and just use BIOS calls???

Here is what happens when I uploaded the .68K file you sent:

file loaded successfully.

```
3A>dir
A: CLIB        : CLINKE   SUB : CLINKF   SUB : CLINK    SUB : ED       REL

A: LIBE     A   : LIBF     A   : LO68     REL : RELOC3   SUB : S        O

A: ASCII    BIN : EHBASIC  BIN : STORE    S   : SUDOKU   BIN : STORE    68K

A: SIEVE    S   : SIEVE    68K : SRIN     S   : ASCIIART BIN : SRIN     68K

A: ASCIIART 68K  <== there it is

3A>asciiart
Insufficient memory or bad file header        <== won't run directly

3A>ddt

*********************************************************
* DDT-68K        9/20/84                  Version 1.3 *
* Serial #XXXX-0000               All Rights Reserved *
* Copyright 1982,1983,1984,1985  Digital Research Inc. *

*********************************************************

-rasciiart.68k                          <== try it as a binary file

Start = 00020000     End = 00025B7F
-g20000
Address Error at 00020806
Address Error at 001FA0B6
PC=4279001F USP=EBC22E8E SSP=0697FFFF ST=FF7C=>TR SUP IM=7 EXT NEG ZER

D  610002D4 2D40FF78 67EE206E FF780C10  002D660E 33FC0001 001FEE94 52AEFF78

A  60064279 001FEE94 206EFF78 10104880  600001BC 2EAEFF78 61000346 600001C8

Address Error at 001FBF46
Illegal Instruction at 001F814C
Address Error at 001FA1CA
?
Line 1111 Emulator at EC260000
Bus Error at 001FA1CA
```

[and many, many more of the same.]

```
3A>ddt

*******************************************************
* DDT-68K       9/20/84                 Version 1.3 *
* Serial #XXXX-0000             All Rights Reserved *
* Copyright 1982,1983,1984,1985  Digital Research Inc. *
*******************************************************

-easciiart.68k                  <== try to load it as an executable file
Insufficient memory or bad file header
-
```

Can't even move it and run it at the *real* TPA starting address:

```
3A>ddt

*******************************************************
* DDT-68K       9/20/84                 Version 1.3 *
* Serial #XXXX-0000             All Rights Reserved *
* Copyright 1982,1983,1984,1985  Digital Research Inc. *

*******************************************************

-rasciiart.68k
Start = 00020000     End = 00025B7F
-m20000,25B7F,20100
-g20100
Bus Error at 00000FFC
Address Error at 001FA0B6
PC=4279001F USP=EBC22E8E SSP=0697FFFF ST=FF7C=>TR SUP IM=7 EXT NEG ZER

D  610002D4 2D40FF78 67EE206E FF780C10  002D660E 33FC0001 001FEE94 52AEFF78

A  60064279 001FEE94 206EFF78 10104880  600001BC 2EAEFF78 61000346 600001C8

Address Error at 001FBF46
Illegal Instruction at 001F814C
Address Error at 001FA1CA
?
Line 1111 Emulator at EC260000
Bus Error at 001FA1CA
?
Line 1111 Emulator at EC260000
Bus Error at 001FA1CA
```

I'm disappointed.

Roger

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by plasmo on Tue, 08 Aug 2017 17:53:14 GMT
View Forum Message <> Reply to Message

Roger,
I tried to run the asciiart_68k.zip that I uploaded 8/5/17 and I also experienced the same problem. That file was derived by saving the image of a disk that contains a working asciiart.68k and use cpmtools (cpmcp) to transfer asciiart.68k to my PC. I believe my disk definition file associated with cpmtools was incorrect. I made corrections and transferred asciiart.68K to my PC again. It is attached below.

I tried two methods to verify asciiart.68k:
1, I used cpmtools and copy it into a CPM disk image and execute it in CPM simulator. This works for one disk image and does not work for a different disk (the two disks have different block size and disk capacity). I'm not quite sure why.
2, Use bin2mot to convert asciiart.68k to s-record starting from 0x20000 then use your srin.s (changed the DUART address to match my hardware) to copy the s-record to CPM file then execute it. This works. I'm pretty confident the asciiart.68k is correct.

I tried sendc68 on asciiart.68k. The s-record generated starts from address 0, and is quite a bit smaller than the original asciiart.68k file in size. When I change the address offset to 0x20000 then copy to CPM using srin, the resulting file is not executable. I don't understand the purpose of sendc68, I don't think it generate the exact s-record image of the executable.

Edit: I just realize as I went through your srin.s code that it is not necessary to generate S record with address offset of 0x20000. No offset works just fine, but you already know this.

File Attachments
1) asciiart.zip, downloaded 424 times

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by norwestrzh on Wed, 09 Aug 2017 02:39:52 GMT
View Forum Message <> Reply to Message

Bill,
Learning new things about the strange world of CP/M 68k every day. *smile* I discovered that sendc68 skips over the CP/M 68k header. Using the Digital Research version of DUMP seems to give a true snapshot of a file. BUT, then if you grab a copy of the DUMP output, you have to have a way to convert it back to binary [and from there convert to S-record?]. I can give you a copy of a utility I wrote [pretty rude and crude] to convert a dump file back to binary. It is written in C [I run it

on my Linux box], called "eatdmp". It is a little picky about the format of the dump file.

I want to be sure I understand correctly -- are you saying that, with method #2, that you can get my copy of asciiart.68k to run??

I'll try to run the latest asciiart.zip file that you uploaded -- see what happens.

As a side note, I picked up an ACC Danube router/protcol converter today at a second hand store for $5[!!!]. It contains a 68302, some flash, some DRAM [both flash and DRAM are SIMMs], and a pair of 27c010 EPROMs. Looks like there is a UART to drive a console [connection on back] and line drivers [1488/1489]. A couple of 20 MHz oscillators on board. And, of course, it has a RJ45 connection for network. I'm still trying to figure it out. It wants 24v A.C. at a little less than half an amp. Looks very clean inside. I can't believe my luck! Of course, I couldn't open it up in the store, so I just bought it on a hunch! Don't know [yet] if it works. Connectors aren't ones that I commonly use.

Roger

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by norwestrzh on Wed, 09 Aug 2017 03:47:41 GMT
View Forum Message <> Reply to Message

A little bit later .....

I converted the file that you sent (asciiart.68k) to S-records, and loaded it using "srin". When I tried to execute it directly from the command line, I got the same old "$03 exception". BUT, just for grins I loaded the file via DDT. DDT checks the header of a file if you load it with the "e" command. It echoed back what looked to me to be correct values extracted from the header. So I tried to run it. It worked!!!! I don't think I've ever tried that before [running it via DDT and the "e" command], so I did the same with a version of the program that I had created previously [and that gave "$03 exception" when I tried to run it directly from the command line]. That worked too. So, for some strange reason, these files that I create with CB68 will run via DDT, but not directly from the command line. Very odd!

Thanks for all your efforts on this. It helped immensely!!

Roger

File Attachments

1) aug8.log, downloaded 523 times

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by plasmo on Wed, 09 Aug 2017 18:14:48 GMT

View Forum Message <> Reply to Message

Roger,
I'm really new to CP/M so I don't understand how asciiart.68k won't work at the command level, but working with DDT's help. There must be some differences between our hardware.

Thank you for your kind offer regarding "eatdmp" (cute name! but I don't have a Linux box running right now so I won't be able to use it.

That's how I started with my ADC spx-mpu boards. I bought a couple boards from a local electronic surplus store for $5 each. They worked so I went back and bought out his inventory of a couple dozen boards. I bought more off eBay, but wind up paying more than $5 per board with shipping and defective boards. Too bad the local store is closed, it had so many interesting surplus electronic items, including Altair S-100 boards.

Are the 27c010 socketed? Check the connection from chip selects of the 27c010 to 68302's CS0 (pin 128 of plastic quad flat pack). If so, they are the boot EPROM. If they are socketed, then repurposing the hardware should be relatively straightforward.
Bill

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by norwestrzh on Wed, 09 Aug 2017 18:54:06 GMT
View Forum Message <> Reply to Message

Bill,
> I'm really new to CP/M so I don't understand how asciiart.68k won't work at the command level, but working with DDT's help.
I don't understand it either. At a very high level, there must be some difference in how the operating system loads an executable file, and the way DDT does it. Perhaps different stack allocations? Maybe I'll figure it out ..... eventually?

On my "Danube" router ...
>> Are the 27c010 socketed?
Yes, I plan on popping them out of the sockets and dumping them.
>> Check the connection from chip selects of the 27c010 to 68302's CS0 (pin 128 of plastic quad flat pack).
Thanks for the tip! I'll check it out. They are almost certainly the boot code.
I hope I can get this thing working as a general purpose 68k. It is in excellent shape!
Roger

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by plasmo on Tue, 15 Aug 2017 20:01:26 GMT
View Forum Message <> Reply to Message

plasmo wrote on Tue, 01 August 2017 17:27
.

.
.
This is a fairly capable 68000 single board computer. Periodically they are available on eBay for $15-$20 like this one: http://www.ebay.com/itm/122306457847

I have had PM conversations with buyers of the SPX-MPU boards on eBay that I referred to in my original post of this thread. It turns out that these boards are not the same as mine. All they have is 29F010 flash chips, no 29F040 for bank 2 and bank 3. This is bad news because 29F010 is too small to put CP/M68K distribution files, even a drastically pared down version. I felt responsible for claiming that spx-mpu can be repurposed to run CP/M68K, but didn't anticipate there are boards out there with 29F010 flash only. So I had privately offered to exchange (one-for-one) my boards with 29f040 and CP/M68K already installed for their boards with 29f010 only. I understand there are more recent buyers of the spx-mpu boards and I want to offer this exchange publicly: Send me a PM with your shipping address that you want to exchange your spx-mpu with 29F010 and I'll send you my address and ship you a board with 29F040 and CP/M68K already installed. No money will be involved in this exchange; we each pay for the shipment of our own boards. You can find more information about the board with 29F040 and CP/M68K installed here:
https://www.retrobrewcomputers.org/doku.php?id=builderpages: plasmo:cmp68k_mpu302
Bill

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by plasmo on Sat, 19 Aug 2017 17:54:16 GMT
View Forum Message <> Reply to Message

MicroEMACS running on repurposed spx-mpu. The C sources are downloaded from cpm.z80.de and compiled/linked use the CP/M68K C tool chain. There are no instruction on how to compile them. This is how I did it:
* Use the included C.SUB to compile all C programs (there are 17 C programs)
* use the included ASM.SUB to assemble the included BIOS.S
* use the included ARCHIV.SUB to form library, ME.LIB
* use the included MERLINK.SUB to generate ME.REL
* use RELOC.68K to convert ME.REL to ME.68K.

When moving cursor up or down with up arrow or down arrow, don't hold down the arrow key. At high repetition rate some of the arrow key strokes are read as character inputs instead of cursor manipulation command and corrupt the data file. Use CTL-V and ESC-V to move down/up pages.

Zipped ME.REL binary and ME.REL in S-record format are attached

File Attachments
1) microEMAC.jpg, downloaded 2340 times
2) me_rel_s68.zip, downloaded 433 times
3) me_rel_bin.zip, downloaded 426 times

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by mikemac on Tue, 22 Aug 2017 14:34:41 GMT
View Forum Message <> Reply to Message

Nice!

plasmo wrote on Sat, 19 August 2017 10:54
When moving cursor up or down with up arrow or down arrow, don't hold down the arrow key. At high repetition rate some of the arrow key strokes are read as character inputs instead of cursor manipulation command and corrupt the data file. Use CTL-V and ESC-V to move down/up pages.

Sounds like buffer overflow on the UART. Are you sure you have HW flow control working? Years ago we had a 68K Unix system hooked up to a Beehive terminal. The function keys, including the arrow keys, send a 3 bytes escape sequence which would overflow the 1 byte buffer of the UART. Manually sending the escape sequence or using the control key equivalent would work everytime.

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by plasmo on Tue, 22 Aug 2017 18:33:05 GMT
View Forum Message <> Reply to Message

Mike,
Thank you for that insight! I have no hardware flow control currently. Looking at the output of the down arrow key, I see 3-byte output, <ESC>,[, B (0x1B, 0x5B, 0x42). When down arrow key is held down, I see the character 'B' inserted intermittently in the text. So you are right, it is a case of buffer overflow. If I use Ctrl-N repeatly (equivalent to using down arrow key in microEMACS), I see no spurious text inserted. So I should either have hardware flow control or an interrupt driven console I/O.

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by plasmo on Fri, 25 Aug 2017 20:52:45 GMT
View Forum Message <> Reply to Message

Mike,
Confirming your observation that it is buffer overflow on the UART that causes extraneous characters being inserted when holding down the keyboard up/down arrows keys: I enabled RTS/CTS hardware handshake on the serial port. Now I can hold down the up/down arrow keys and no extraneous characters are generated. Thanks!
Another bonus with hardware handshake is that gkermit downloaded from
ftp://kermit.columbia.edu/kermit/archives/gkermit68.tar.gz
is working now. Prior to this gkermit can send file to my PC correctly but can't receive data--too many retries. With hardware handshake, it can send & receive text data just fine.

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by plasmo on Sat, 02 Sep 2017 18:44:18 GMT
View Forum Message <> Reply to Message

I kludge up a IDE-CF interface board to the SPX-MPU board via the 96-pin DIN connector. The IDE-CF is driven by the 82C55 on the SPX-MPU board. The BIOS is updated with 4 more drives so drives A & B are read-only flash drives, drive C is RAM drive, drives D thru G are 8 megabyte each on IDE-CF:
A>stat dsk:

```
     A: DRIVE CHARACTERISTICS
12,288: 128 BYTE RECORD CAPACITY
 1,536: KILOBYTE DRIVE  CAPACITY
   256: 32 BYTE  DIRECTORY ENTRIES
     0: CHECKED  DIRECTORY ENTRIES
   128: 128 BYTE RECORDS / DIRECTORY ENTRY
    16: 128 BYTE RECORDS / BLOCK
 1,024: 128 BYTE RECORDS / TRACK
     0: RESERVED  TRACKS

     B: DRIVE CHARACTERISTICS
 3,072: 128 BYTE RECORD CAPACITY
   384: KILOBYTE DRIVE  CAPACITY
   256: 32 BYTE  DIRECTORY ENTRIES
     0: CHECKED  DIRECTORY ENTRIES
   128: 128 BYTE RECORDS / DIRECTORY ENTRY
    16: 128 BYTE RECORDS / BLOCK
 1,024: 128 BYTE RECORDS / TRACK
    12: RESERVED  TRACKS

     C: DRIVE CHARACTERISTICS
 2,048: 128 BYTE RECORD CAPACITY
   256: KILOBYTE DRIVE  CAPACITY
   128: 32 BYTE  DIRECTORY ENTRIES
     0: CHECKED  DIRECTORY ENTRIES
   128: 128 BYTE RECORDS / DIRECTORY ENTRY
     8: 128 BYTE RECORDS / BLOCK
 1,024: 128 BYTE RECORDS / TRACK
     0: RESERVED  TRACKS

     D: DRIVE CHARACTERISTICS
64,512: 128 BYTE RECORD CAPACITY
 8,064: KILOBYTE DRIVE  CAPACITY
   512: 32 BYTE  DIRECTORY ENTRIES
     0: CHECKED  DIRECTORY ENTRIES
   256: 128 BYTE RECORDS / DIRECTORY ENTRY
    32: 128 BYTE RECORDS / BLOCK
 1,024: 128 BYTE RECORDS / TRACK
     1: RESERVED  TRACKS
```

```
     E: DRIVE CHARACTERISTICS
64,512: 128 BYTE RECORD CAPACITY
 8,064: KILOBYTE DRIVE  CAPACITY
   512: 32 BYTE  DIRECTORY ENTRIES
     0: CHECKED  DIRECTORY ENTRIES
   256: 128 BYTE RECORDS / DIRECTORY ENTRY
    32: 128 BYTE RECORDS / BLOCK
 1,024: 128 BYTE RECORDS / TRACK
    64: RESERVED  TRACKS

     F: DRIVE CHARACTERISTICS
64,512: 128 BYTE RECORD CAPACITY
 8,064: KILOBYTE DRIVE  CAPACITY
   512: 32 BYTE  DIRECTORY ENTRIES
     0: CHECKED  DIRECTORY ENTRIES
   256: 128 BYTE RECORDS / DIRECTORY ENTRY
    32: 128 BYTE RECORDS / BLOCK
 1,024: 128 BYTE RECORDS / TRACK
   128: RESERVED  TRACKS

     G: DRIVE CHARACTERISTICS
64,512: 128 BYTE RECORD CAPACITY
 8,064: KILOBYTE DRIVE  CAPACITY
   512: 32 BYTE  DIRECTORY ENTRIES
     0: CHECKED  DIRECTORY ENTRIES
   256: 128 BYTE RECORDS / DIRECTORY ENTRY
    32: 128 BYTE RECORDS / BLOCK
 1,024: 128 BYTE RECORDS / TRACK
   192: RESERVED  TRACKS
```

On paper, IDE access via the 82C55 is quite slow; it takes four instructions for byte access and 5 instructions for word access; compare that to single instruction for byte or word access for bus-connected IDE interface. I did a quick benchmark of disk operation:

Pip 250K bytes of files from flash drive to RAM drive takes 3.5 seconds,
Pip 250K bytes of files from flash drive to IDE-CF takes 14.5 seconds,
Pip 250K bytes of files from IDE-CF to RAM drive takes 4.5 seconds.

So write to IDE-CF is slow, but read from IDE-CF is fairly quick compare to the RAM drive access. I think the bottleneck is in the CP/M overhead and IDE write itself. The access time of the CF registers is a small part of the overall performance.

File Attachments

1) DSC_29660902.jpg, downloaded 2287 times

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by wsm on Sat, 02 Sep 2017 19:06:16 GMT
View Forum Message <> Reply to Message

Quote:
So write to IDE-CF is slow, but read from IDE-CF is fairly quick compare to the RAM drive access. I think the bottleneck is in the CP/M overhead and IDE write itself. The access time of the CF registers is a small part of the overall performance.
One thing that you may be overlooking is that file creation in CP/M takes at least 3 write operations: directory allocation, data and finally directory completion. On your CF card there may also any delays after the writes before the card is ready to accept new commands.

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by plasmo on Sat, 02 Sep 2017 22:18:28 GMT
View Forum Message <> Reply to Message

You are right that not only there are multiple writes, there are also flash erasure-before-write delay that is not deterministic due to remap of flash sectors by the CF controller. I'd logged the delays from write command to flash ready to accept write data and I can see intermittently a large delay that is 2 orders of magnitude greater than the normal delay. This makes performance measurements difficult.

Early on I was hung up on how many instructions it takes to wiggle the CF control lines via the 82c55. Beside, it was a downright primitive way of interfacing to hardware. I thought about a more efficient hardware interface, but the 82c55 is already there and all port signals are available on the 96-pin connector, so I thought I try it first. My feeling now is that a bus-connected IDE-CF interface may be a tad faster, but hardly justify the board modifications it requires. So I'm going ahead with the layout a simple DIN-to-IDE pc board.

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by wsm on Sun, 03 Sep 2017 01:12:11 GMT
View Forum Message <> Reply to Message

You got me kind of curious since I've played around with a couple of different flash interfaces on my 33MHz Z180 system. So far I've only been comparing relative times within this system. Using a 264K file (as reported by STAT), I got the following:

RAM disk to CF card (2GB Verbatim) : 2.8 seconds using DMA mode 2
RAM disk to serial flash chip array : 3.5 seconds using I/O instructions

Reads were harder to measure with a stopwatch ... about 1 second from CF to RAM disk.

Both interfaces used a data buffer of the appropriate size.

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by plasmo on Sun, 03 Sep 2017 05:30:58 GMT
View Forum Message <> Reply to Message

That z180 is an impressive machine. Judging by the gap in performance, I think there are a lot of room for improvement for the 68000 running CPM as well. The write algorithm in the BIOS can certainly uses some refinement.

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by plasmo on Mon, 04 Sep 2017 22:04:15 GMT
View Forum Message <> Reply to Message

If there are interests in the repurposed SPX-MPU board for CPM-68K, I can offer boards at $20 each plus shipping ($8 in the USA). The board will have CPM-68K v1.3 distribution files installed in drive A (read only), utilities in drive B (read only), and RAMdisk in drive C (read/write). It is ready to run CPM-68K out of the box. You'll need a 5V power at 500mA, min, null modem cable for DB25, and a terminal software running at 38400 baud, 8,1,N with hardware flow controls. So far I've modified 8 SPX MPU boards successfully, 3 for my own testings, 5 to replace the SPX-MPU boards purchased on eBay that do not have large enough flash memories. I'm currently working on a CompactFlash daughter board. It is a simple design with just one 7400 IC, so I expect the pc board to be well smaller than 100mm x 100mm and costs $1.50 for the bare board. Please let me know if you are interested.

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by norwestrzh on Mon, 04 Sep 2017 22:33:35 GMT
View Forum Message <> Reply to Message

Hi Plasmo,
I'd be interested in one of those boards.
Please let me know the details: norwestrzh at aol
Roger

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by plasmo on Fri, 08 Sep 2017 19:49:23 GMT
View Forum Message <> Reply to Message

For those interested in playing with the repurposed SPX-MPU board running CP/M 68K in simulation, attached is the S-record of programs loaded in the repurposed SPX-MPU board. The same programs can also be loaded into EASy68K's Sim68K and execute in simulation. You will see the same results (with some limitations explained later) in simulation as you'll see with the actual hardware. However, you will have much greater visibility in the simulated environment: you can display and change the memory contents and you can single-step BIOS code. Since the disks A, B, &C are all in memory (disk A&B are read-only flash memory, disk C is a RAMdisk), the contents of disks can be displayed as memory. Simulation on my 2.4GHz PC is faster than the actual hardware running at 16MHz. For an example, the ASCII art program runs in 41 seconds in simulation vs 104 seconds on the hardware.

For people already have the actual SPX-MPU hardware, the simulated environment is still useful because if the hardware behaves unexpectly, you can run the same code in the simulated environment and compare results. Another use for the simulated environment is developing codes for the CPM-68K BIOS.

As mentioned before, there are limitation to the simulation, specifically the simulated console is a virtual console so there are no easy way to upload or download files from console to/from CPM. Kermit does not work in simulation which is a bother because currently the gkermit program can't receive binary files from CPM (it can send binary & text files to PC, and it can also receive text files from PC without problems), and I'm unable to debug that problem in simulation. Similarly, microEMACS expects the console to be a VT52 terminal and uses escape sequences to manipulate the cursor position. That doesn't work with simulated console.

This is the step-by-step instruction for simulating CP/M68K in SPX-MPU board:
1. run Sim68K (the program can be downloaded from easy68k.com)
2. in the drop-down menu under 'File', select 'Open...' and pick 'MPUBIOS.S68'
3. In the drop-down menu under 'File', select 'Open Data...' and pick 'CPM15000.s68' <--this is CPM15000.sr binary of CCP and BDOS.
4. In the drop-down menu under 'File', select 'Open Data...' and pick SIMDSKA.S68 <-- this is drive A image
5. In the drop-down menu under 'File', select 'Open Data...' and pick SIMDSKB.S68 <--this is drive B image
6. In the Registers area, change 'PC=' field to 00015000 <-- this is the start address of CPM-68K
7. Press 'run' icon (or F9). You should see the A> prompt in the console. You are now running CP/M-68K.

File Attachments

1) cpm68k_4_sim68k.zip, downloaded 600 times

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by norwestrzh on Sun, 10 Sep 2017 21:06:09 GMT
View Forum Message <> Reply to Message

Bill --
Got your package today. It was very well padded, so the board was in excellent shape. Powered

it up, and ran CP/M for a while. Everything works. Thanks.
It will take me a while to look it over more carefully, and become familiar with everything.
Any idea what the 3 position DIP switch and hex encoder (lower left corner) do?
Roger

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by gkaufman on Mon, 11 Sep 2017 00:09:14 GMT
View Forum Message <> Reply to Message

Likewise, my board arrived safely and powered up easily.

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by plasmo on Mon, 11 Sep 2017 01:51:19 GMT
View Forum Message <> Reply to Message

Roger & Gary,
I'm glad you've received the board in great shape. Did you take a look at drive C? Drive C is a RAMdrive backed up with a super capacitor. The capacitor can keep its content intact for well over a week on my bench, but I'm curious whether the anti-static bag would discharge the capacitor during shipping.

I'll answer your question about the switches directly, but it is also an action item for me to update the wiki page with better organized info regarding memory map & I/O schematics.

The hex switch and 3-pos toggle DIP switch are buffered via UG2 (HC244) and appear as a memory location as follow:
DIP SW1 is d0, with 10k pull up,
DIP SW2 is d1, with 10k pull up,
DIP SW3 is d2, with 10k pull up,
d3 is pulled up with 10K,
Hex switch SK3 pin 1 to d4, with 10K pull up,
Hex switch SK3 pin 4 to d5, with 10K pull up,
Hex switch SK3 pin 3 to d6, with 10K pull up,
Hex switch SK3 pin 6 to d7, with 10K pull up,
Note: the 4-bit input value for SK3, d4-d7, is invert of the value printed on the switch.

Attached is a hand-sketched schematic of the switches (You can see the envelop glue lines because it was literally on the back of an envelop).

For completeness sake, the high byte of the discrete inputs are buffered through UG11 (HC244) & UG3 as follow:
d8 is 96-pin DIN pin B27 pulled up with 10K, 10K series resistor to UG11,
d9 is 96-pin DIN pin A27 pulled up with 10K, 10K series resistor to UG11,

d10 is 96-pin DIN pin B28 pulled up with 10K, 10K series resistor to UG11,
d11 is 96-pin DIN pin A28 pulled up with 10K, 10K series resistor to UG11,
d12 is 96-pin DIN pin B29 pulled up with 10K, 10K series resistor to UG3,
d13 is 96-pin DIN pin A29 pulled up with 10K, 10K series resistor to UG3,
d14 is 96-pin DIN pin B30 pulled up with 10K, 10K series resistor to UG3,
d15 is 96-pin DIN pin A30 pulled up with 10K, 10K series resistor to UG3,

The memory map of the I/O resources is as follow:
0x300000-0x30001F, 82C55
0x300020-0x30003F, 68692 DUART
0x300040-0x30005F, real time clock, RTC72423 <- backed up with super capacitor
0x300060-0x30007F, switch inputs <- this is the hex switch/DIP switch/DIN connector
0x300080-0x30009F, special function, still need to figure it out.

The connection of the DB25 connector on the face plate:
DB25-2 RX buffered via UH12 to 68692-35 (RxDA)
DB25-3 TX buffered via UH12 to 68692-33 (TXDA)
DB25-5 RTS buffered via UH12 to 68692-32 (OP0)
DB25-4 CTS buffered via UH12 to 68692-42 (IP5) & 68692-8 (IP0) <- these are engineering changes, the blue wire on 68692 and a long blue wire at the back of the board.

You may also be interested in the pin definition of the 96-pin DIN connector. I attached a text file below. It needs to be clean up and included in the wiki, but it is information you and other users (there are 5 users right now) of this board may want to have now.

File Attachments

1) ug2-sch_F.jpg, downloaded 3610 times
2) Din96.txt, downloaded 604 times

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by plasmo on Mon, 11 Sep 2017 02:10:08 GMT
View Forum Message <> Reply to Message

The fact there are 7 user definable switch inputs and 8 inputs from DIN connector is quite useful. The software can change mode based on the switch inputs. When I add the Compactflash IDE daughter board I will take one of the DIN discrete input and tell the BIOS that external IDE-CF is attached so to enable CF drives.

In my XBEE application, I use the hex switch to assign address of each XBEE station and the DIP switches for different modes of operation.

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by norwestrzh on Mon, 11 Sep 2017 17:26:46 GMT
View Forum Message <> Reply to Message

Drive C: on my board looks thus:
A>dir c:
C: AS68SYMB DAT : CONC SUB : CLINKONC SUB : BASIC SUB : BLINK SUB
C: ME 68K : ASM SUB : HELLO C : HELLO 68K : HELLO O
C: ASCIIART BAS : ASCIIART 68K : ASCIIART O : FCOMP S : FCOMP O
C: FCOMP 68K
I think that's OK? I wondered about the anti-static bag as well. Seemed not to have drained my super cap.

Roger

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by plasmo on Mon, 11 Sep 2017 17:41:50 GMT
View Forum Message <> Reply to Message

Roger,
These files look good. They were used or created to check out the board and software on drive A & B.
  Bill

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by etchedpixels on Wed, 13 Sep 2017 18:13:21 GMT
View Forum Message <> Reply to Message

Yes I'm interested in one to play with - what would shipping to the UK be like ?

On the IDE performance by way of comparison a 4MHz Z80 driving an SD card via a hardware SPI interface (bytewise so it just does INI INI INI INI INI INI ..... comes out at a similar speed to the 33MHz Z180 bit banging it which shows the value of having hardware to do the job . For IDE you should be able to read 16bits/instruction and you can run most CF cards at PIO4 if you negotiate the speed with it. That's supposed to top out at 16MB/second (120ns cycle time) which should be adequate for any retro hardware

---

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by plasmo on Thu, 14 Sep 2017 00:56:33 GMT
View Forum Message <> Reply to Message

etchedpixels,
Thank you for your interest in the repurposed SPX-MPU board for CPM-68K. I have not shipped package internationally for quite a while. According to the US Postal Service, First Class International to UK is $22.50. I need your postal code in UK to get a accurate quote from UPS,

and if you have a preferred carrier, I'd happy to use it.

The board does have a 16-bit interface to IDE via the 82C55. To read a CF sector of data (512 bytes), the processor needs to:
1. assert nCS, IDE data register (addr 0x0) lines on 82C55 port C,
2. assert DIOR line (82C55 port C),
3. read in low byte IDE data (82C55 port B),
4. read in high byte IDE data (82C55 port A),
5. negate DIOR line.

Repeat steps 1-5 256 times to read a CF sector. I measure 8.6uS to do step 1-5, so the max transfer rate is 230k byte/sec.
Bill

Edit: Looking at the code again, I can optimize by using address pointers and movep instruction. So the loop time is reduced to 3.1uS for an effective transfer bandwidth of 650k bytes/sec

---

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by rhkoolstar on Thu, 14 Sep 2017 11:17:09 GMT
View Forum Message <> Reply to Message

Wouldn't it be faster to use the IDE in 8 bit mode? saves on (slow) CPU overhead.

Rienk

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by plasmo on Fri, 15 Sep 2017 02:02:32 GMT
View Forum Message <> Reply to Message

The reason IDE access thru 82C55 is slow is because the processor needs to manually wiggle the controls and address lines of IDE which are connected to 82C55's port C. It is the same number of instruction to wiggle addresses/controls for 8-bit mode as for the 16-bit mode. Having invested in the same number of instructions to wiggle the addresses&controls, I might as well read 16-bit of data instead of 8-bit.

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by plasmo on Fri, 29 Sep 2017 19:06:54 GMT
View Forum Message <> Reply to Message

Spent this morning looking over the SPX-MPU boards that were recently purchased from eBay which only have 29F010 flashes (e.g., http://www.ebay.com/itm/122306457847 . I determined they have the same pc board layout as my collection of boards that have two banks of 29F040. So it is possible to switch out the 29F010 with 29F040.

I also have plenty of the SPX-MPU board with 2 banks of 29F040 already installed so anyone interested in running CPM-68K or just want to have a general 68000 development board with 1 meg or RAM, 2.25 meg of flash and plenty of peripherals for $20 (plus $5 shipping in USA) please contact me.

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by etchedpixels on Wed, 04 Oct 2017 16:09:29 GMT
View Forum Message <> Reply to Message

I guess the 29F010 is sufficient anyway once you have the IDE or SD interface available ?

(I went digging - the difference comes from the boards and their original OS - at some point the OS update on the cards stopped fitting and they introduced a new/upgraded board requirement.

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by plasmo on Wed, 04 Oct 2017 18:52:23 GMT
View Forum Message <> Reply to Message

You are right that CPM68K can run on the repurposed SPX-MPU board with just the boot flash (256K bytes) and an expansion daughter board containing an IDE or SD drive. Currently the monitor and the CPM image (including BIOS) are reside in the boot flash with plenty of room for growth. The 2nd & 3rd bank of flash (2megabyte total) contain disk images, but that can be relocated to the CF or SD cards. My original motivation is to have a low cost working CPM68K all in a single PC board, but expansion boards with IDE-CF and other hardware will certainly add significantly more flexibility and features to the base board. It is desirable to have a runtime configurable BIOS that detects the system parameters and enable the appropriate drivers, which dovetailed nicely with the current discussion on modular CP/M BIOS.

While digging into SPX-MPU, have you run across hardware design documents with schematics and/or theory of operation? What about a way to upgrade software? I used a brute-force hardware programming header to "hijack" the boot flash and overwritten the factory code, but it is so much easier to reprogram using existing software if such capability exists.
Bill

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by etchedpixels on Tue, 10 Oct 2017 12:39:13 GMT
View Forum Message <> Reply to Message

I found nothing about theory of operation or schematics, not even the pinouts. There *is* a remote update facility on the cards but there's nothing helpful in the documentation. The manuals dated around 2000 show a system where you connect to the board either via serial or over a network and trigger an ftp from an internet host for the firmware (which by then at least is vxWorks) so presumably you'd need a fully functional soneplex to do so, although there is a reference

elsewhere to xmodem.

And by way of other hits

https://www.linkedin.com/in/eric-nelson-525ab96a

seems to be one of the folks who worked on it - and this also confirms the 68000 stuff was in the market in 2000

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by plasmo on Tue, 17 Oct 2017 04:42:01 GMT
View Forum Message <> Reply to Message

Received the pc board from Seeed Studio this morning. Here is the assembled board and how it plugs into the SPX-MPU board. It appears to be working. The updated BIOS adds 4x 8-megabyte CF drives, D:, E:, F:, G:, to the existing flash drives A:, B: and ramdrive C:. I will update the wiki page with the design files.

File Attachments
1) DSC_30871016.jpg, downloaded 3354 times
2) DSC_30901016.jpg, downloaded 3110 times
3) DINIDECF_scm.pdf, downloaded 485 times

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by plasmo on Sat, 25 Nov 2017 14:55:02 GMT
View Forum Message <> Reply to Message

I completed the manual tracing of the pc board and updated the SPX-MPU schematics in the wiki page:
https://www.retrobrewcomputers.org/doku.php?id=builderpages: plasmo:spx-mpu:mpuschematic
It is likely the schematics contain errors. Please let me know if you've found them.

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by plasmo on Wed, 29 Nov 2017 23:39:29 GMT
View Forum Message <> Reply to Message

The wiki page for the CF IDE daughterboard is completed.
https://www.retrobrewcomputers.org/doku.php?id=boards:other: idecf2din:start

The daughterboard works very well. I encountered no problems with different brands of CF. Four additional 8-meg disks are added with the daughterboard so a MPU302 with the daughter board will have 7 disks: drive A and B are flashdisks, drive C is the RAM disk, drive D to G are the additional CF disks on the daughterboard. The current owners of MPU302 need to update their BIOS to accommodate the daughterboard. The updated BIOS will detect the presence of daughterboard and adjust the number of CF disks accordingly. The BIOS update procedure is documented in the MPU302 wiki:
https://www.retrobrewcomputers.org/doku.php?id=boards:sbc:mp u302:mpu302updates

At this moment I have 6 bare CF IDE daughterboards. If you are interested in bare PCB, it is $1.50 plus $1.50 for shipping in the USA. For assembled and tested board, it is $8 plus $3 shipping in the USA.

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by plasmo on Fri, 22 Dec 2017 18:01:52 GMT
View Forum Message <> Reply to Message

A number of forum members have purchased the MPU302 (AKA repurposed SPX-MPU board), I thought it maybe useful to document the frequently asked questions here.

wiki page for MPU302 is here:
https://www.retrobrewcomputers.org/doku.php?id=boards:sbc:mp u302

Powering up: If you have the CF daughterboard/MPU302 combo, you MUST power up via the 2.5mm power jack on the daughterboard (center lead is 5V, barrel is ground). This is because the 5V is shorted to ground when 2.5mm power plug is not inserted. Expected power consumption is 300mA @ 5V

Serial port connection: Terminal program should be configured to 38400 baud, N,8,1, with RTS/CTS hardware handshake. The serial port voltage level is RS232, not TTL, so it won't work with the cheap serial-to-USB adapters with TTL level I/O. A DB25 to DB9 (assuming serial port cable out of the PC is DB9) is needed with the following wiring:
DB9 DB25
2---------3
3---------2
7---------4
8---------5
5---------7

As many MPU302 I've plugged/unplugged, I find the DB25 connector difficult to unplug, so I actually used the pigtail kludge in the picture on my bench.

Boot to CP/M-68K:
When power is applied initially, the MPU302 will sign on with:

AMBug v2.07aHardware handshake with CTS/RTS, 8/30/2017
RAM at $0 and alias at $200000, boot ROM at $600000. Type "he" for help

>

Type 'bo' to boot into CP/M-68K <--This is obscure because 'bo' command is not shown in the 'help' menu. I'll fix that in next revision.

File Attachments

1) DSC_32951222.jpg, downloaded 516 times

Subject: Re: Porting CP/M 68K to a repurposed ADC Soneplex MPU board
Posted by b1ackmai1er on Sun, 13 May 2018 10:52:32 GMT
View Forum Message <> Reply to Message

norwestrzh wrote on Sat, 05 August 2017 10:18Hmmm .....
MBASIC(interpreter) (Z80 @ 10 MHz) 1:54
MBASIC(interpreter) (FPGA Z80 :46 (equiv. to ~25 MHz)]
CB68(compiled) (68k @ 6 MHz) 4:23 <== I can get it to work on one of my slower 68ks
EhBASIC(interpreter) (68k @ 12 MHz) 0:55
EhBASIC(interpreter) (CP/M 68k @ 12 MHz) 1:26
EhBASIC(interpreter) (68k @ 10 MHz) 1:06
EhBASIC(interpreter) (68010 @ 10 MHz) 1:02
EhBASIC(interpreter) (8-bit strapped 68EC00 @ 10MHz) 2:55
GCC(compiler) (68k @ 12 MHz) 0:20

asciiart.bas benchmark for Rienk's sbc-2g-512 7.3728Mhz Z80 board running NASCOM ROM BASIC Ver 4.7:

2m43s

```
10 FOR Y=-12 TO 12
20 FOR X=-39 TO 39
30 CA=X*0.0458
40 CB= Y*0.08333
50 A=CA
60 B=CB
70 FOR I=0 TO 15
80 T=A*A-B*B+CA
90 B=2*A*B+CB
100 A=T
110 IF (A*A+B*B)>4 THEN GOTO 200
120 NEXT I
130 PRINT " ";
140 GOTO 210
200 IF I>9 THEN I=I+7
205 PRINT CHR$(48+I);
210 NEXT X
```

```
220 PRINT
230 NEXT Y
Ok
run
00000001111111111111111122222233347E7AB32222211100000000000000000000000000
00
00000111111111111111112222222233557BF754332222111110000000000000000000000
00
00011111111111111111122222222233445C     643332222111110000000000000000000000000
01111111111111111122222223344456C     654433332211111100000000000000000000000
11111111111111112222233346 D978 BCF    DF9 6556F42211111100000000000000000000000
11111111111122223333334469             D   6322111111000000000000000000000
11111111122233333334457DB                  85332111111100000000000000000000
11111122234B744444455556A                  96532211111100000000000000000000
122222233347BAA7AB776679                   A3221111110000000000000000000
2222233334567       9A                     A5322211111100000000000000000
222333346679                               943222111111000000000000000000
234445568  F                               B54322211111100000000000000000
                                    8643322211111100000000000000000
234445568  F                               B54322211111100000000000000000
222333346679                               943222111111000000000000000000
2222233334567       9A                     A5322211111100000000000000000
122222233347BAA7AB776679                   A3221111110000000000000000000
11111122234B744444455556A                  96532211111100000000000000000
11111111122233333334457DB                  85332111111100000000000000000
11111111111122223333334469             D   6322111111000000000000000000
11111111111111112222233346 D978 BCF    DF9 6556F42211111100000000000000000000000
01111111111111111122222223344456C     654433332211111100000000000000000000000
00011111111111111111222222233445C     643332222111110000000000000000000000000
00000111111111111111112222222233557BF754332222111110000000000000000000000
00
00000001111111111111111122222233347E7AB32222211100000000000000000000000000
00
```