HACKADAY.IO

ISA bus slave NS32016 processor board

Back to overview

By Dave Rand and George Scolaro, 1985.

Keith

Back to overview Files 14 Components 0 Logs 13 Instructions0 Discussion 3

« Back to project details

SORT BY:

Newest

Tech data for build your own PD32

02/16/2024 at 14:31 • 0 comments

I've added tech data so you can build your own PD32 or at least the reconstruction of it. There are schematics, PCB layout, two BOMs, 3D rendering images, and Gerber files.

If anyone decides to make their own PD32 using the supplied Gerber files please post here with your results. Especially if any changes are necessary and/or you find improvements.

Would really like to see your results of a PD32 build. Thanks!

Cross-assembler recovery

02/11/2024 at 01:26 • 0 comments

T' ı Definicon cross-assembler but it is an MSDOS executable binary so I can't run it c ux PC nor can I compile it for Linux because there is no source-code.

There is C source code for one in Dr. Dobbs' Journal. The OCR text is not good because all the OCR tools I can use for free try to automatically recognise text in columns, with no way to specify single-column operation. Which mangles the order of source code scans.

Several days of tedious salvage work later, I've got the 32000 cross-assembler to the point where it compiles, but it crashes with a memory segmentation fault. Probably trying to do something that used to be okay on MSDOS but not on a modern OS.

I don't have enough spare time to get it working, but there are many capable coders out there, and I am sure one of them could get it running fairly quickly.

I notice that most of the code is doing lexical analysis, splitting the source into lines and words, turning words into symbols and numbers.

I have just started learning Python, and it has vast libraries for doing this kind of work. If I were to write one from scratch, I would start with Python to make the job easier and quicker.

PCB trace routing

02/07/2024 at 14:24 • 0 comments

The PD32 PCB routes in in FreeRouting as a 2-layer board, however, based on what I've read the NS32K chipset seems rather sensitive to signal quality. A 2-layer board is less expensive, but if a 4-layer design is more likely to work, then I think that's the way to go.

It seems there were some articles and/or letters in MicroCornicopia after the original PD32 article suggesting PCB quality caused a lot of problems to the initial roll out in 1986. They also used an improved 4-layer board to improve signal quality, so I think it is prudent to do the same.

The good news is the PCB easily routes as a 2-layer board so transitioning to a 4-layer PCB should be no problem. Just add two internal copper layers, one for GND and the other for VCC.

Component sourcing

02/04/2024 at 17:01 • 0 comments

C[illegible] links to chips in BOM

```
 13    U1, U3   PAL20L8 or better e.g. 22V10
 14    U11,U12  27256
```

15	U14,U15	74ALS573	Transparent latch, 8-bit
16	U2	74LS74	
17	U4	74LS32	
18	U5	74LS646	Octal bus transceivers and register
19	U6	74ALS00	
20	U7	74LS175	
21	U8	NS32202	Interrupt Control Unit
22	U9	74LS30	
23	U10	74LS04	
24	U13	NS32081	Floating-Point Co-processor
25	U16	NS32201	Timing Control Unit
26	U17	NS32016	Processor
27	U18	DP84412	Replace with PAL16R6 or better e.g. 16V8
28	U19	NS32082	
29	U20	DP8419	DRAM address multiplexer and refresh row counter
30	U21	74S32	
31	U22	74LS08	

The PAL chips are no problem, I have original PAL chips and the logic can be ported to modern GAL chips.

UT Source have the DP8419 and DP8409

Unix System V software

02/03/2024 at 22:03 • 0 comments

I feel completely content if a hardware project boots up and says "Hello world!".

After that, I hand over to software hobbyists. Hardware tends to have a completion point, software does not.

If you have the time and skills, it is possible to get this board running Unix, though I wonder what you could do to impress someone through a text-only interface.

H[illegible] software

C[illegible] at 21:28 • 0 comments

The board and Host PC communicate through an 8-bit bus, although the board does not have any 8-bit latches to store data. Maybe those would be on the Z80 board?

I'm pretty sure this is mainly the user interface terminal with the ability to transfer something like intel hex files. This should not be too hard to create.

I feel it could be replaced by a USB-FIFO type interface.

Toolchain

02/03/2024 at 21:19 • 0 comments

The design includes the source code, but no binary image. We need a 32000 assembler for that. Which I thought would take some finding because the 32000 family never took off commercially. Thankfully, I was wrong!

http://cpu-ns32k.net/Crossass.html

I tried using this to assemble the PD32 firmware - without success. It seemed to want carriage returns with linefeed, and not have colons before equates. Conditional assembly didn't work. I think the syntax is significantly different, and I don't have a manual to tell me what it is. If it can't do conditional assembly it may not be up to modern standards.

Hex numbers are not the usual $FF, or C style 0xFF, but x'FF which is very strange. So the h'FF needs changing to x'FF.

I don't have time to investigate this at the moment.

It isn't hard to split the firmware into odd and even bytes when required.

Later, we will need a C compiler. Also hard to find. Here seems a good place to start:

http://cpu-ns32k.net/Gnutool.html

Checking the KiCAD PCB

02/03/2024 at 21:13 • 0 comments

Start by asking what are the worst mistakes that could happen, and work downwards from there.

For example, check if a hole is in the right place before you check it is the right size.

I' [illegible] boards with the wrong size, the components back to front or upside-down, so I know what to look for!

Checking the KiCAD schematic

02/03/2024 at 21:04 • 0 comments

This involves printing it and the original, then marking checked tracks with a highlighter pen. Not difficult but you do have to be patient and meticulous. I can do this.

2024-02-05

I printed the old and new schematics on A3 paper.

First job will be to check the pin numbers of each part match.

2024-02-16

Checked final sheet today. All sheets now checked.

No discrepancies, although the original numbers header pins in an unconventional way.

The capacitor for BBG looked wrong (+ve to ground) until noting that it is the decoupler for a negative bias generator.

Original circuit omits power and ground pins, so I have not been able to correlate them.

Checking the KiCAD library parts

02/03/2024 at 20:59 • 0 comments

I used to do CAD librarian work so I can certainly do this.

Next >

⇡ Going up?

About Us Contact Hackaday.io Give Feedback Terms of Use Privacy Policy Hackaday API