

The Web Site to Remember National Semiconductor's Series 32000 Family

Home	Chips	Systems (1)	Systems (2)	M32632	Software
Miscellaneous	Imprint				

Opus

NS32016/NS32032 based systems

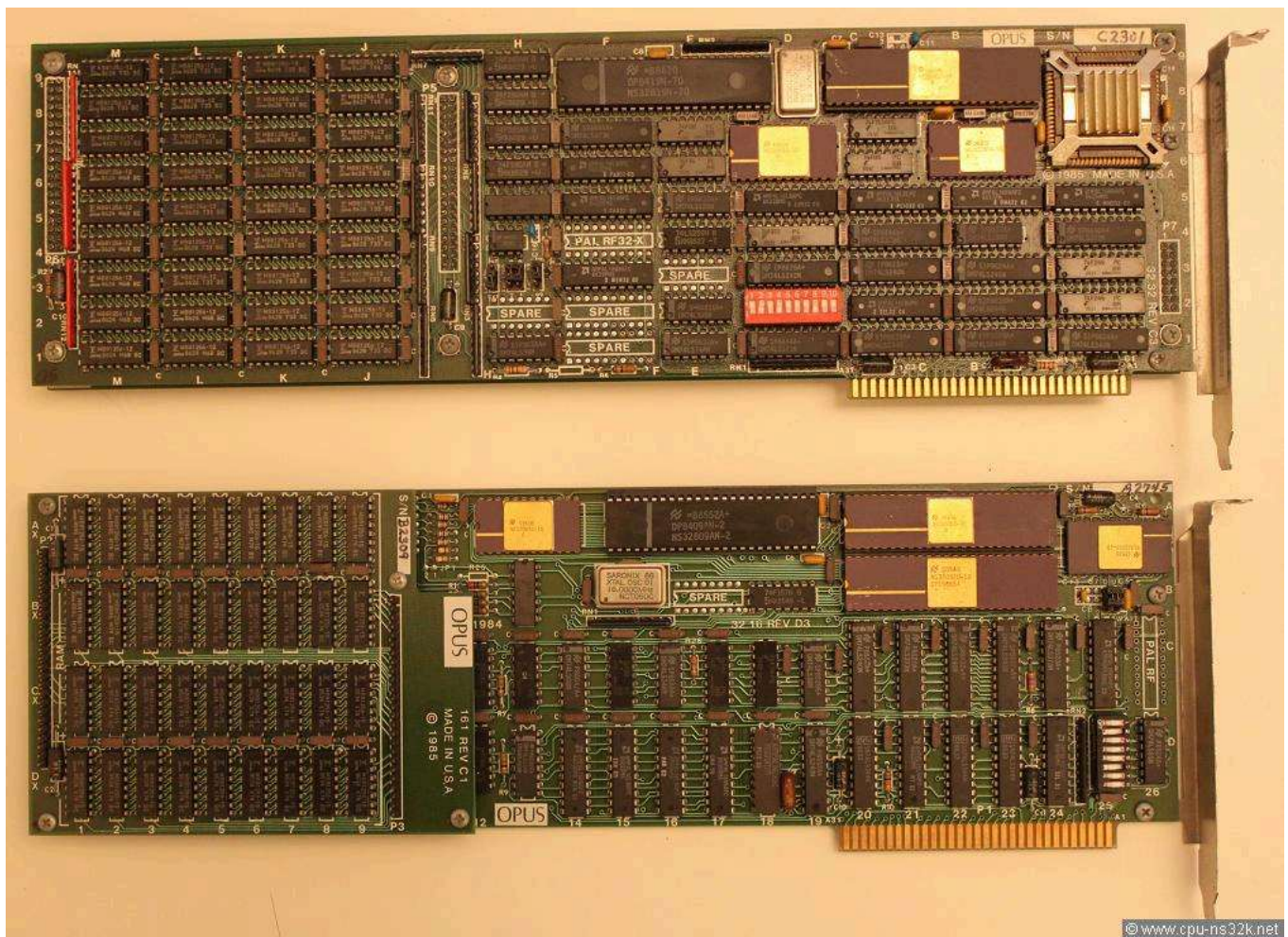
The Opus board from 1985 is based on the NS32016D-10 CPU with nearly all the peripherals of the family : MMU, FPU and TCU. Main memory is 2 Mbytes of DRAM with parity protection. 1 Mbytes is placed on an add-on board screwed to the main board. The system does not have a boot ROM. Therefore everything is programmed from the host which is obviously a PC. The connector is used for 8-bit ISA slots.



Fig. 1. PC add-on board from Opus.

The photo in Figure 2 shows two boards from Opus in comparison. The upper board is based on the NS32032 CPU. This board uses a large memory expansion board for a total memory size of 4 Mbytes. The lower board is of the same type as in Figure 1 (please note the different series numbers in the upper right corner of the boards). The NS32032 CPU is placed in the upper right corner just below the series number C2301.

The Opus boards were used in CAD systems. For this kind of application they were sold with a large memory array. 144 DRAM chips, each 256-kbit in size, were used for the memory of the NS32032 system. 36 of them are placed on the main board. For the remaining 108 chips an own board was used which equals the size of the main board. Both boards are screwed together.



© www.cpu-ns32k.net

Fig. 2. Two PC add-on boards from Opus: the upper one is based on the NS32032 CPU which is placed under the heatsink.

The photo in Figure 2 is available in higher resolution [here](#).

Since September 2018 some software for the system will be available soon - thanks to Simon, see below. This is great and I will test my board in Richard's hardware.

The application note [AN-587](#) contains a hint to Opus boards on page 4.

In July 2017 Alexander informed me about a surprising discovery. The hardware of the National Semiconductor SYS32/20 Development Package was made by Opus. This way National Semiconductor saved a lot of effort. Whether the hardware of the NS32332 based SYS32/30 is also made by Opus is not known.

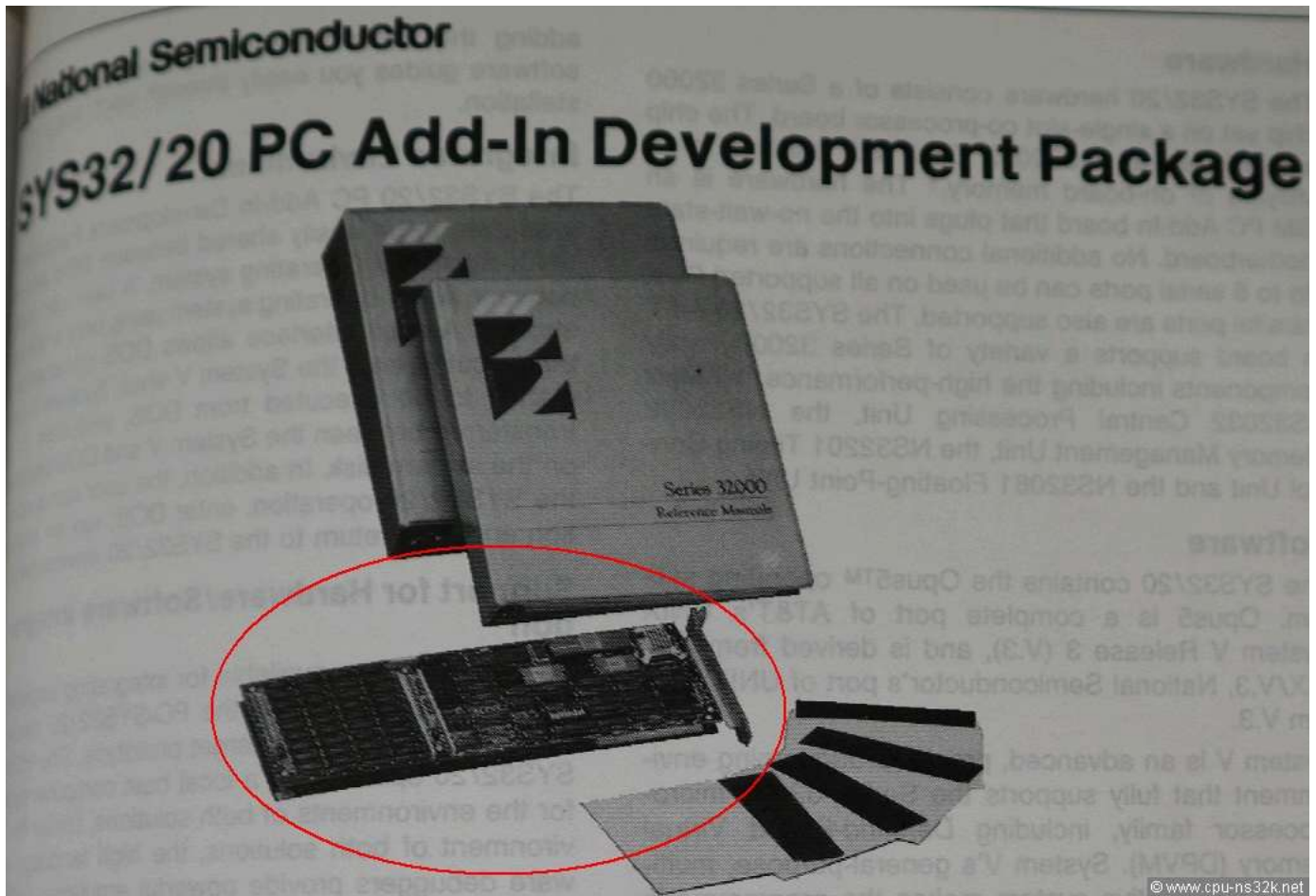


Fig. 3. It is obvious that the board of the SYS32/20 is identical to the Opus NS32032 board in Figure 2.

■ High Performance, 10 MHz, no-wait state, 32-bit expansion board for an IBM-PC/AT or compatible system ■ An Operating System derived from AT&T's UNIX® System V.3 ■ The Series 32000 GNX (GENIX Native and Cross-Support) Language tools including the Series 32000 assembler, linker, monitors and debuggers ■ Hardware that supports the NS32032 CPU, NS32082 MMU, NS32201 TCU and the NS32081 FPU ■ Two available on-board memory configurations: — 2-Mbyte RAM — 4-Mbyte RAM	■ Software available on 1.2-MByte floppies ■ Complete support for the following application tools: — SPLICE — National's Series 32000 Development Board Family — Compilers for C, FORTRAN77, Pascal and Ada — Complete System V Documentation — 4.2 "bsd" Utilities — Tools for Documentors (TFD), a derivative of AT&T's DWB™ utilities — Multiuser environment
---	--

Description

National Semiconductor's SYS32/20 is a complete, high performance development package that converts an IBM-PC/AT or compatible system into an ideal environment for the support of Series 32000®-based applications. The SYS32/20 PC Add-In Development Package allows mainframe-size programs to run on a personal computer at speeds similar to those of a

VAX 780. The SYS32/20 consists of a 32-bit PC Add-In board based on the Series 32000 chip set, a complete port of AT&T's UNIX® System V.3 specially developed software that integrates the UNIX and DOS operating systems, and National's Series 32000 development tools (GNX).

6

© www.cpu-ns32k.net

Fig. 4. The 1988 edition of the Series 32000 Microprocessor Data Book describes the content of the SYS32/20 Development Package.

NS32532 based system

Until June 2017 I was believing that the german company Mikron has built an NS32532 based add-on board for the PC. Now I learned from John C. that the board shown in an advertisement from Mikron is an Opus board. Mikron is "only" a systemintegrator.

In 1988 Mikron offered the board with 4 Mbytes of DRAM together with software for ~10,000 € which is very high from today's perspective. For this money you got at the end of the 1980's a new VW Golf. A complete system with a Compaq Deskpro 386 PC as a host, a graphic board and a 19" Sony color monitor was offered for ~40,000 €. This was the price of a new Porsche!



Fig. 5. The NS32532 based PC add-on board for the 16-bit ISA bus.

The board runs a variety of System V. The two Xilinx FPGAs (XC2018-70, 100 logic cells) are loaded at MSDOS startup. UNIX hard disk access was through a helper-program which ran under MSDOS.



Fig. 6. A partially populated daughter board for memory expansion screwed to the CPU board.

The daughter board in Figure 4 has DRAMs on both sides of the PCB. Fully populated it has a capacity of 16 MBytes. Together with the 4 MBytes of the CPU board a system may have 20 MBytes in total. The memory is parity protected.



© John C.

Fig. 7. A detailed view of the CPU and FPU.



Fig. 8. The board was designed and made in the USA. Please note that the manufacturer of the DRAMs was Motorola.

An Opus board still running in 2018

In September 2018 I got an email from Simon telling me that he has a system with an Opus board. Not very exciting. But one of the next sentences said: **"It still works."** Wow - this was a true surprise. Until now nobody seems to have something like this. He wrote about the system:

"... The system is an IBM 3270 PC AT/GX. The system was assembled and sold by Valid Logic Systems. The Opus card allowed the IBM PC to boot SYS V UNIX and run the Valid GED schematic entry software. The system was loaned by Valid to the company I worked for at the time, LSI Logic. When we were done with it, Valid said they didn't want it back. My boss told me, "Get that thing out of here," so I did. It still works. The UNIX still boots, and Valid GED still runs. The monitor weighs 35 Kilos, the base system weighs another 23 Kilos. The Display Attachment Unit is a separate case almost as big and almost as heavy as the base system. The DAU is what we now refer to as a GPU, which is just a card in current desktop systems. Only IBM could build such a monster. At the Vintage Computer Fair in San Jose, CA in 2000, this system took Best in Class (post 1981), Best Presentation (completeness), and Best of Show."

The Opus board inside the PC is identical to the one in Figure 1.



Fig. 9. Below the PC containing the Opus board is the DAU, the Display Attachment Unit.

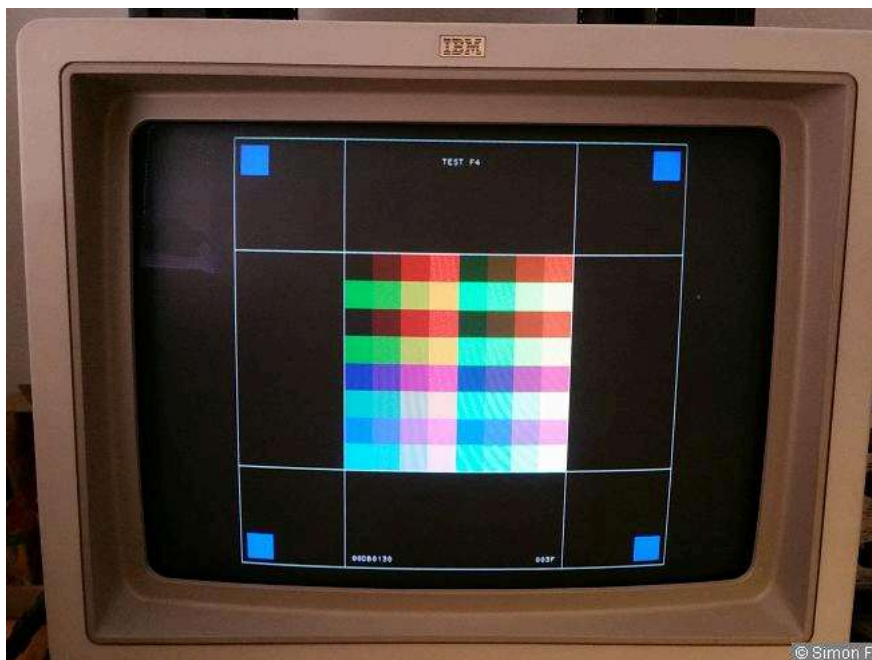


Fig. 10. The high resolution monitor of the system shows a test picture.

Simon gave me more details about the system:

"Valid specifically chose this hardware configuration and ported their software to it. I believe they also did the UNIX port. The 3270 AT GX has what's called an All Points Addressable card that allows the system to write to any point on the screen at 1024x1024 resolution. I do not have any other software that uses this capability except for GED. IBM intended for this system to be used as a graphics terminal connected to a mainframe. In 3270 mode, it was a text-based terminal for the mainframe. In APA mode, it was also a graphics display system. They just let the Opus card replace the mainframe. The system first boots to DOS, then you invoke the UNIX boot program and the Opus card takes over the screen. In text mode, UNIX looks no different from DOS, just being lines of text that I think are 24x80 characters. This is extremely limiting on such a large monitor! Once GED is invoked, then the 1024x1024 graphics mode is started. I think it only does 4 colors at 1024x1024. The Valid SCALD station had a monochrome green monitor, so GED runs on Opus with green graphics to look the same."

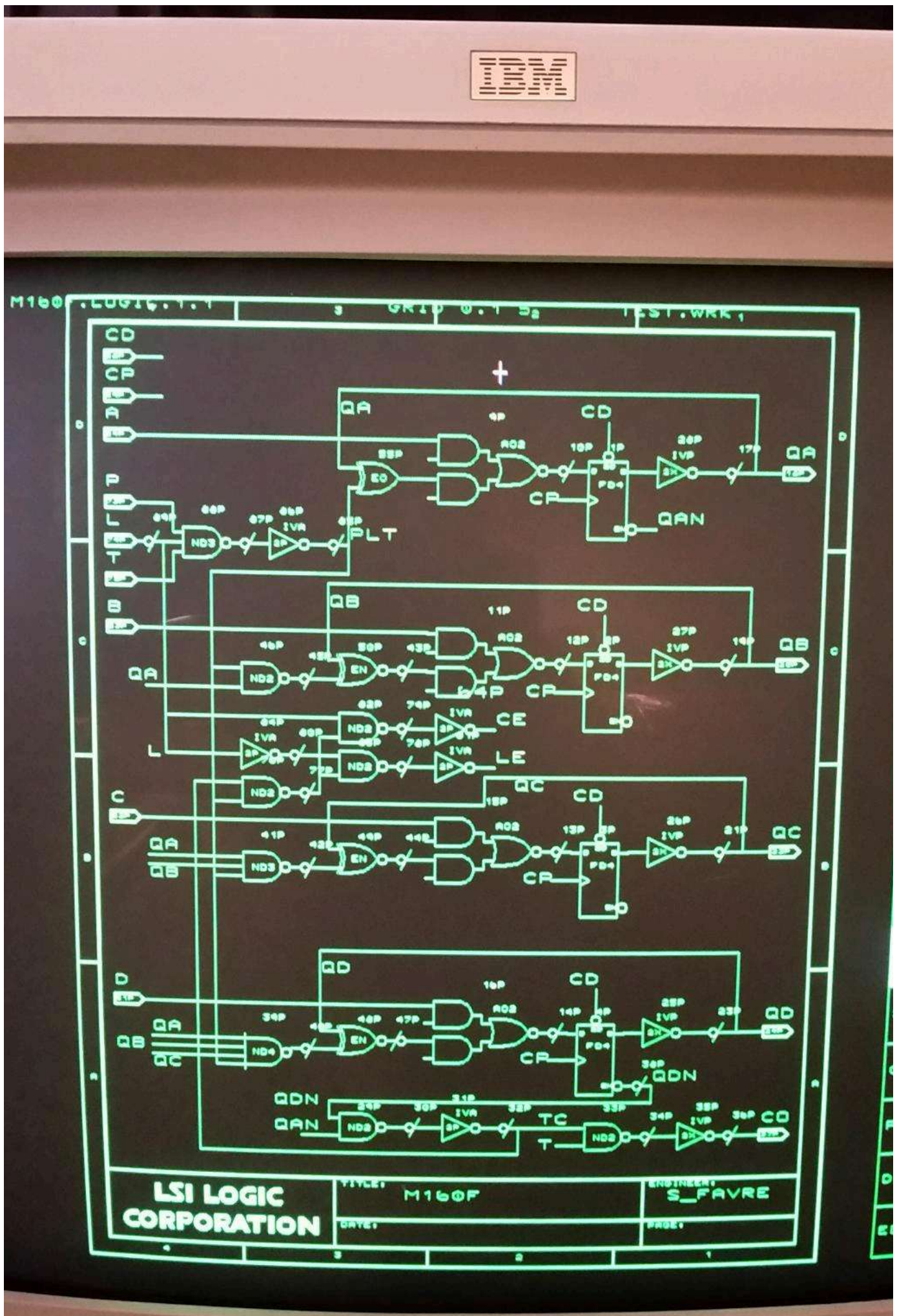


Fig. 11. A detailed view of the output of the Valid GED schematic entry software showing a 4 bit decimal counter.

Starting in 1988 I was sitting in front of a similar system from Daisy and doing schematic entry. Unfortunately I cannot remember any further technical details about Daisy...

It is fascinating to see a system which has done useful work running 30 years later. And still all transistors and bits are in a good shape. I should ask Simon whether he can make a small YouTube video working with the system.

My Opus board is running in 2022

To be honest only the debugger program is running in April. But it is the first step and it required already some help and some good ideas. It will be a long way to go to get UNIX running on this machine.

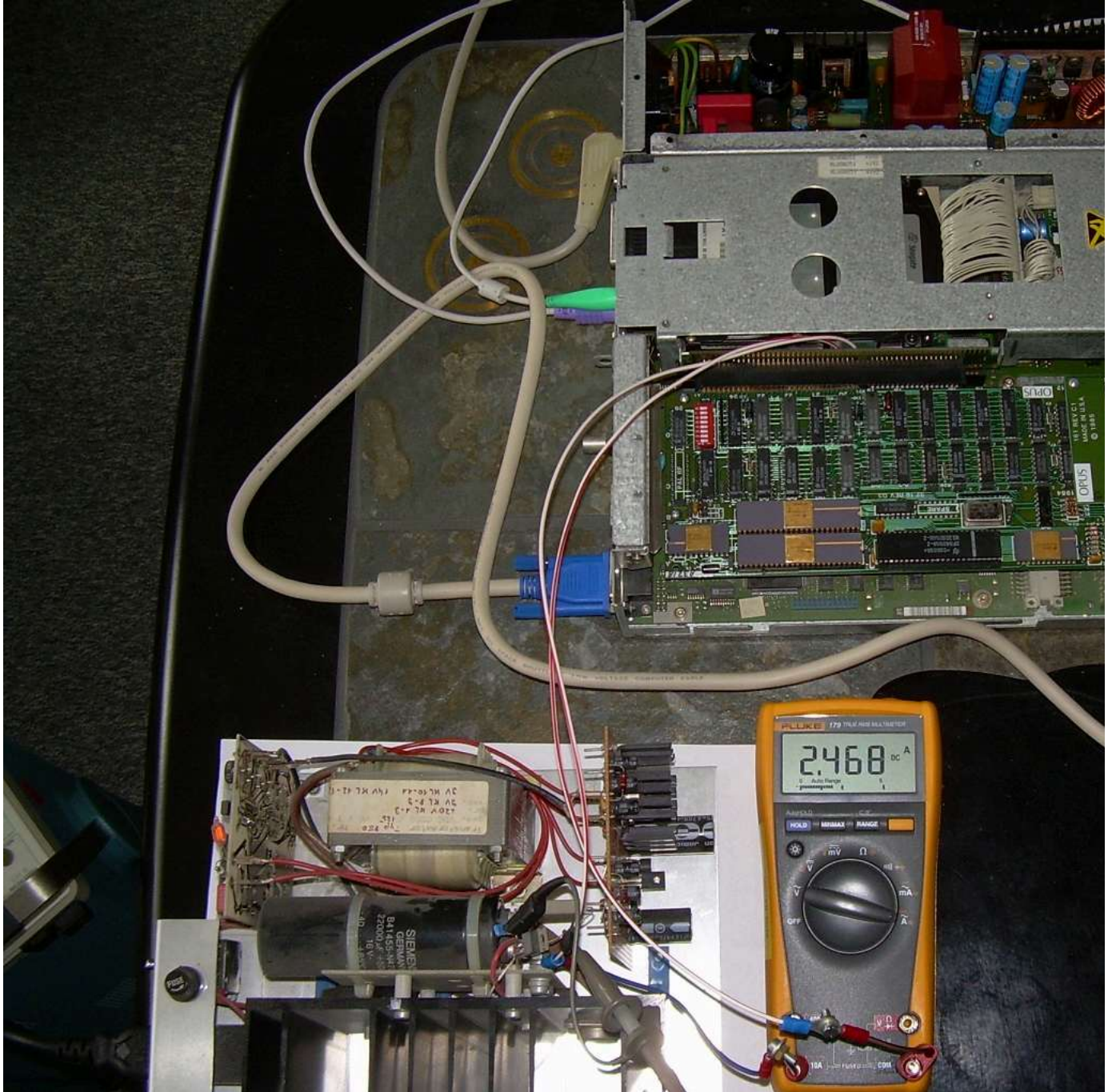


Fig. 12. A view of the experimental set up.

I use an old PC model from Siemens-Nixdorf as the host machine. It's name is PCD-4L. It is based on the 486SL processor running at 33 MHz. The slim design of the computer is nice but it has some drawbacks. First there is not much room inside. And there is no fan. So maybe the Opus board will get hot if the case is closed. I'm not sure if the ventilation slots at the side are sufficient.

Second the power supply of the PCD-4L is weak. It only delivers 5A at 5V. Before the test, I had no idea how much current the Opus board would draw. My guess was around 3A. Therefore I decided to supply the 5V for the Opus board from externally. As you can see in reality it is around 2.5A. The external power supply in Figure 12 is the old power supply for my first National System TITAN1, built in the 1980's.

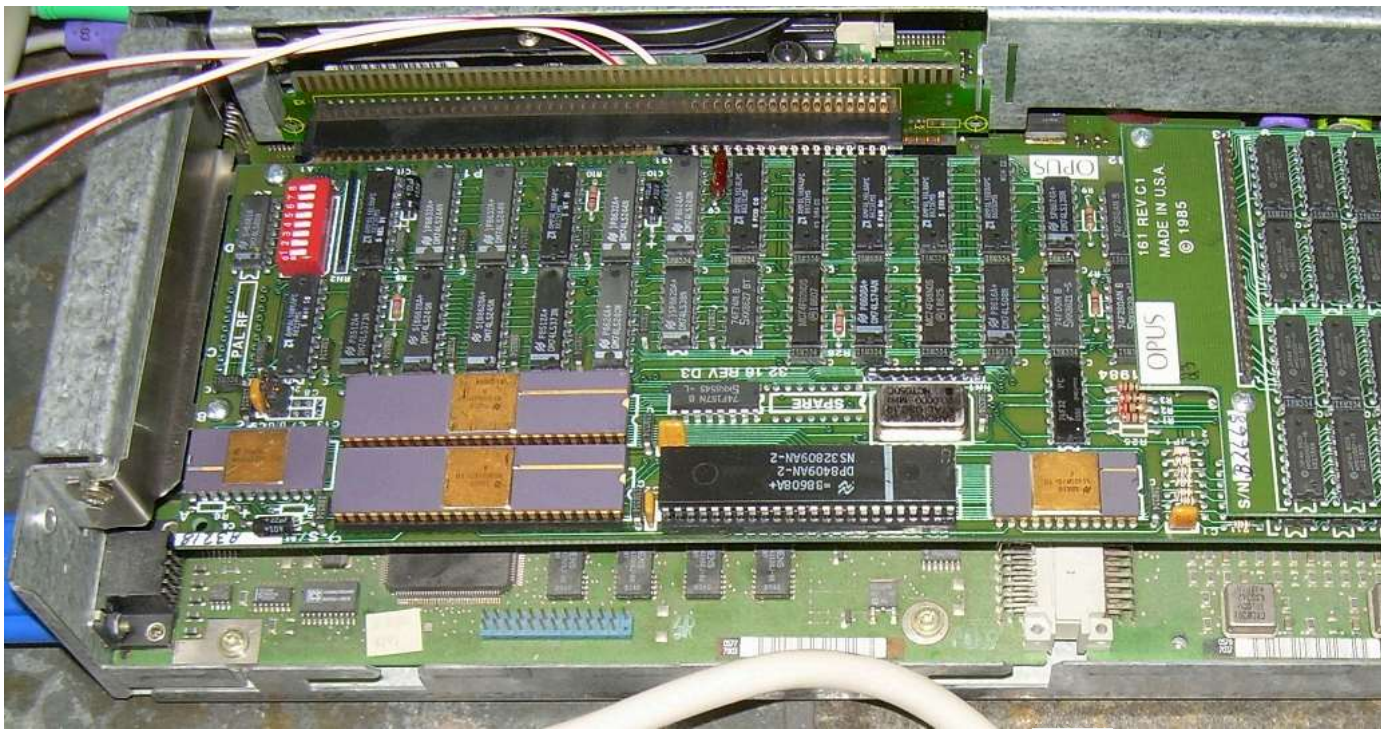


Fig. 13. A detailed view of the Opus board inside the PC. It is a good feeling to see old hardware working again 😊

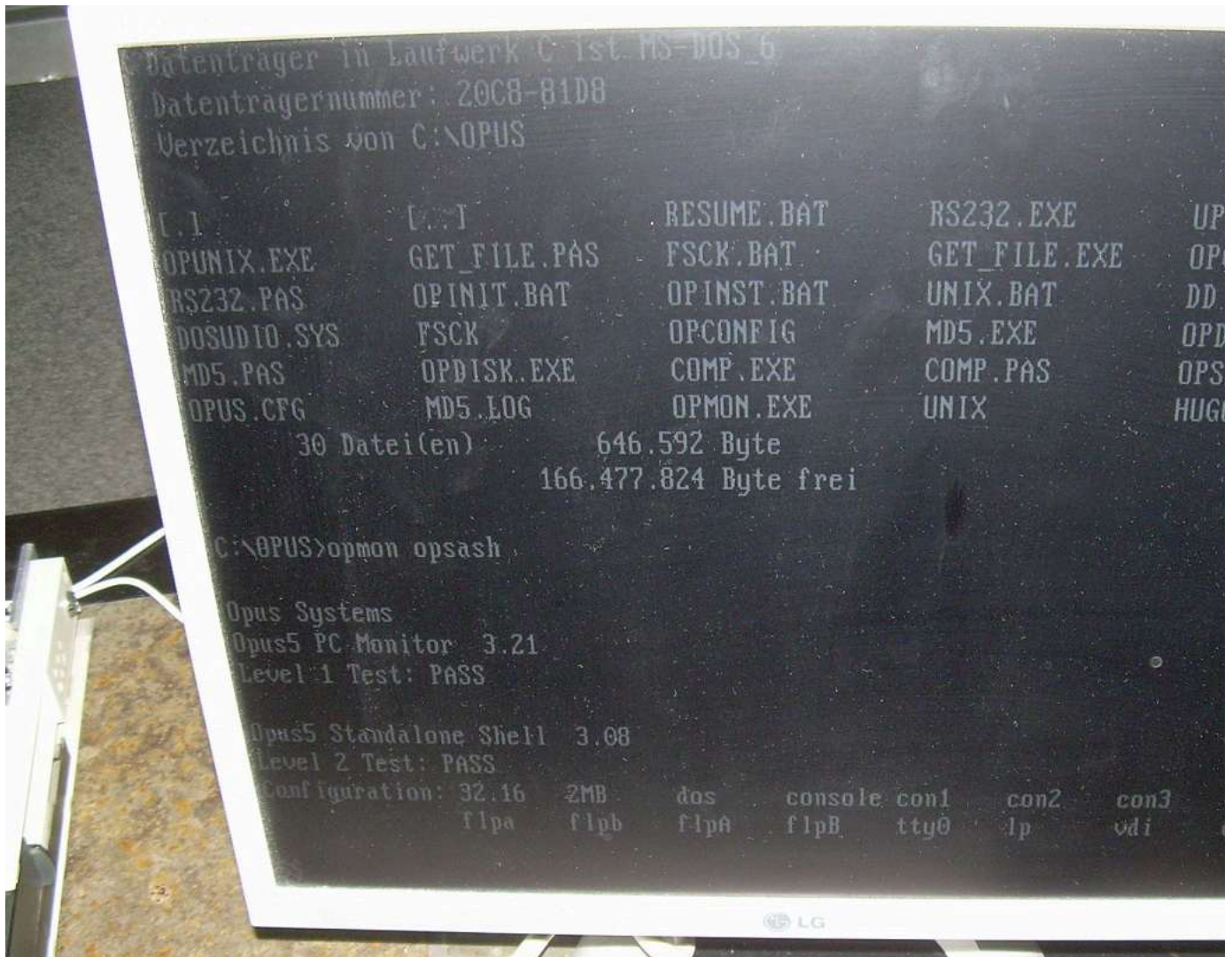


Fig. 14. First to note: the tests are PASS! The first program used was "opsash" = Opus Standalone Shell. But nothing useful could be done with it.

The old PC has only a floppy disk drive to transfer data. No USB. So the basic software for the Opus board was transferred serially from TITAN3. For this purpose I had to program the PC in Turbo Pascal which a friend still has on floppy disks - thanks to Manfred!

There is a big challenge with the remaining software for UNIX: it is stored on 40 disks not formatted in DOS format. I think an emulation of

the floppy drive is a good solution.

The next step is to save the content of the hard disk drive. Then I will try to configure the system. This step is pretty unsafe because I have not the user manual. There are some hints but I'm not sure that this is enough. But if something goes wrong I can restore the saved HDD content.

Does anybody have the Opus532 User Manual? This would be great! => Update of 22 June 2022: I found the manual at bitsavers.org. It was placed there just some hours before. Surprisingly the name was changed in 1987 to **Opus 100PM User Manual**. The NS32016 board was named **Opus 108PM** and the NS32032 board was named **Opus 110PM**. Anyhow every open question shall find an answer now...

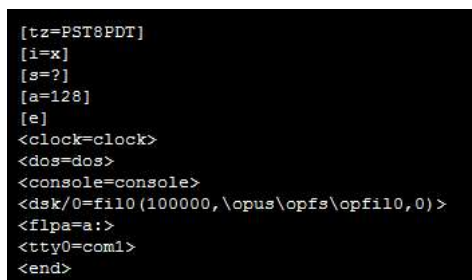
Job done: Opus board is running UNIX!

6 weeks later the job is surprisingly already done. It took 34 disk writes and reads to install UNIX. The most important information was that the **OPFIL*** file in each directory of the Opus zip file is not an image file. This was my assumption because the size of the files was nearly 360 kB which is the size of a double density disk. But it turned out that I could simply copy **OPFIL** to any MSDOS disk. At least I needed 6 disks at the same time for the biggest package which is the manual package. Of course I had to transfer each **OPFIL** serially to the PC at 38400 Baud...

*) file names written in capital letters are MSDOS files, file names written in lower-case are UNIX files.

Alexander told me later that **OPFIL** is a **cpio** archive.

The first step of the installation was the modification of the configuration described in the file **OPUS.CFG**. It looks like the configuration was written for the Valid system described in the previous section. I took everything out which seems not to be useful. My hope was that an MSDOS file could be used as the UNIX partition. And this became true - see Figure 15.



```
[tz=PST8PDT]
[i=x]
[s=?]
[a=128]
[e]
<clock=clock>
<dos=dos>
<console=console>
<disk/0=fil0(100000,\opus\opfs\opfil0,0)>
<flpa=a:>
<tty0=com1>
<end>
```

Fig. 15. The current content of the Opus configuration file **OPUS.CFG**.

The size of the UNIX partition disk/0 is 100,000 blocks. 15,000 of them are used for swap space.

One mystery for me is the file **DOSUDIO.SYS** in the Opus directory of the PC, see Figure 14. I included it in **CONFIG.SYS**. But I'm not sure whether it has any effect. I did not made a test without it.

After the modification of **OPUS.CFG** I started the configuration with **OPCONFIG.BAT** without any parameters. A disk with the file **OPFIL** of the ROOT directory in the zip file should be already in the disk drive. The configuration process is straightforward and self-explanatory. At the end it reads the disk and installs a small UNIX system in the swap space.

The configuration process generated a file named **SWAPBOOT.BAT**. I run it simply because I didn't know what else to do. I was asked for three more disks for the kernel (Directories K-1, K-2 and K-3 in the zip file). Afterwards my feeling was that it is time to run **UNIX.BAT**.

The capability of the fresh system is very limited. Therefore it is necessary to install at least 30 other disks with the program **/opus/bin/opload**.

This UNIX is a System V UNIX from AT&T. It has a C compiler and a Fortran compiler. Instead of **more** it uses the command **pg** for the same purpose. Many people do not know this including myself. It is really necessary to read the documentation because many commands and options are different from the Berkeley UNIX.

Two programs are used for file transfer between MSDOS and UNIX, **udcp** and **ducip**. Both generate an I/O error message at the beginning of the transfer. But the data is always right. What to do? I have no idea.

I use the program **sync** to update the UNIX file system super block. Then I switch off the PC.

After I found out that I can use the function `mclock()` of the Fortran compiler to measure execution times I was able to compile the benchmark program Linpack. The output of a Linpack run is shown in Figure 16. The run took 11 minutes and 6 seconds. The column mflops shows that the pair NS32016 and NS32081 performs 36,700 double-precision (64 bit) floating-point operations per second. This is an expected result.

```

Please send the results of this run to:

Jack J. Dongarra
Computer Science Department
University of Tennessee
Knoxville, Tennessee 37996-1300

Fax: 865-974-8296

Internet: dongarra@cs.utk.edu

This is version 29.5.04.

      norm. resid      resid      machep      x(1)      x(n)
1.25501937E+00  1.39332990E-14  2.22044605E-16  1.00000000E+00  1.00000000E+00

      times are reported for matrices of order 100
      dgefa      dgesl      total      mflops      unit      ratio      b(1)
times for array with leading dimension of 201
1.813E+01  5.500E-01  1.868E+01  3.675E-02  5.442E+01  3.336E+02  -9.159E-15
1.813E+01  5.000E-01  1.863E+01  3.685E-02  5.427E+01  3.327E+02  1.000E+00
1.818E+01  5.000E-01  1.868E+01  3.675E-02  5.442E+01  3.336E+02  1.000E+00
1.817E+01  5.383E-01  1.871E+01  3.671E-02  5.449E+01  3.340E+02  5.298E+02

times for array with leading dimension of 200
1.818E+01  5.500E-01  1.873E+01  3.665E-02  5.456E+01  3.345E+02  1.000E+00
1.818E+01  5.500E-01  1.873E+01  3.665E-02  5.456E+01  3.345E+02  1.000E+00
1.818E+01  5.500E-01  1.873E+01  3.665E-02  5.456E+01  3.345E+02  1.000E+00
1.816E+01  5.383E-01  1.870E+01  3.672E-02  5.447E+01  3.340E+02  5.298E+02
end of tests -- this version dated 05/29/04

```

Fig. 16. To see first time the output of my preferred test program Linpack on a system is always a proud moment - if the result is correct...

Later I learned that my Opus board was running with one wait state during Linpack. I took out the wait state and performance increased to 39 kflops, 6% more. Running the system without a wait state is in my view a bit critical because the access time of the DRAM is 150 ns. But up to now there were no problems.

Now I'm waiting for any reports of Mattis who owns the two boards shown in Figure 2. He has PC hardware like me. Does he also has the time to do the same? Let's see.

The Opus boards of Mattis

Yes, Mattis has the time. He could successfully boot UNIX on his NS32016 based board. His PC is an Ericsson AT computer with a 12 MHz 286 processor. He replaced the old MFM disk and controller with an ISA IDE card and a compact flash card. The floppy is replaced with a Gotek.

This is the nice part of the story. Here comes the sad part: the NS32032 based board is not running...

He replaced the 20 MHz quartz oscillator with a 16 MHz part. But this didn't help much. He tried then to analyze what's happening on the board with a logic analyzer, a HP 16500B. The results were weird. At the end, he thinks that the NS32032 CPU is broken.

By the way, Mattis owns a not so small computer museum with very old hardware, PDP8 and so on. There is also a website about it. The address is www.datormuseum.se. Both are worth to have a look at it!

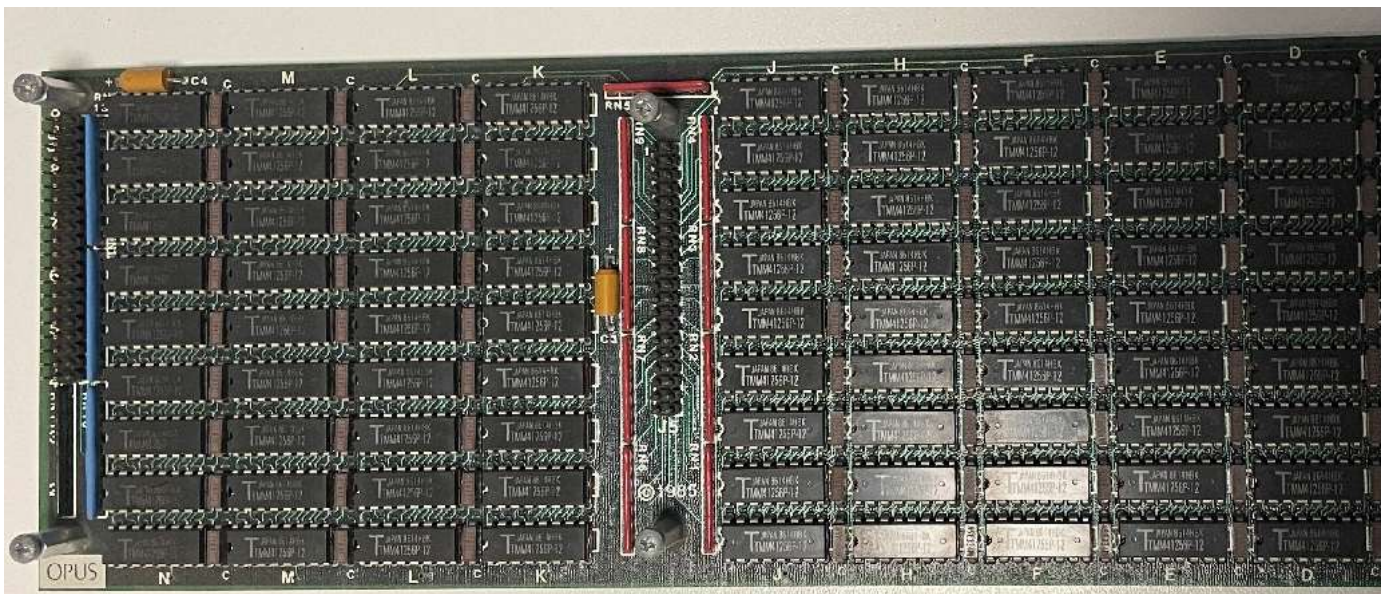


Fig. 17. The memory controller DP8419 has to drive 144 DRAM chips. The specification is only defined for 88 devices. More devices means slower signals...

To analyze the problem, Mattis took off the memory expansion board, see the Figure above. This frees a connector on the CPU board containing the whole 32 bit data bus. This was the perfect place to hook the LA, see the next Figure. The CPU data bus is multiplexed with the addresses, which gives additional informations.

This Opus board should run at 10 MHz without a wait state. For me it is somehow unbelievable because of the length of the unbuffered address/data bus. The DRAMs are fast with 120 ns access time. Maybe no wait state is possible under typical conditions. Hopefully we will

see the board in action soon.

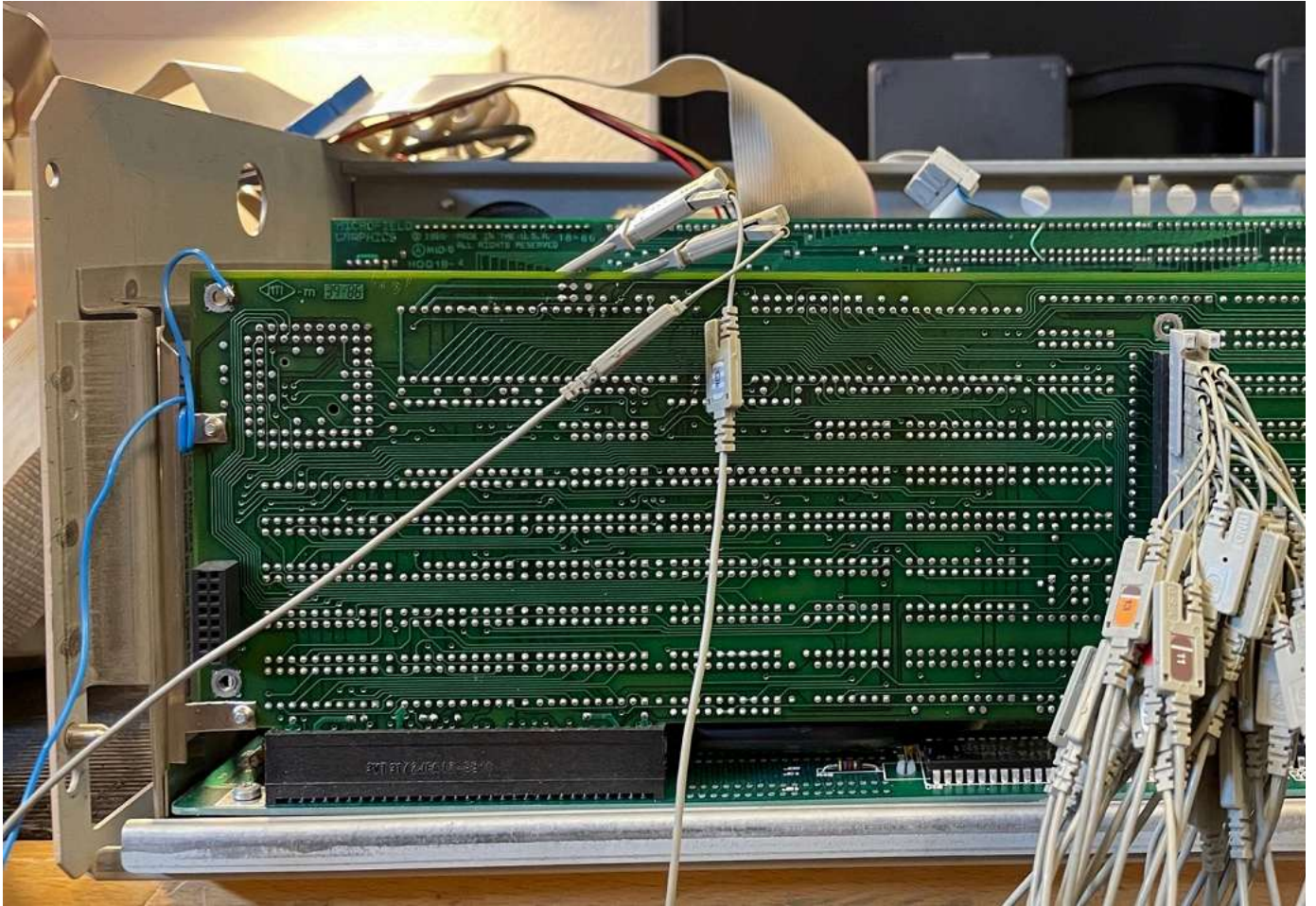


Fig. 18. The backside of the CPU board with the connections to the logic analyzer. The blue cable at the left side is the ground connection.

A new CPU could do

In the meantime, Mattis got a new (old) CPU. He wrote about what happened next:

"At first I had trouble with the memory expansion board on the 32032 board. Almost directly after the boot it was short-circuiting the computer and it smelled weird.

Removed the board and checked the base 32032 board which still worked just fine. Thought it was one of three tantalum capacitors that had shorted. I removed those but they measured fine. One memory chip smelled weird. I removed that one and replaced it with a new chip. Then tested the board with a 5V lab supply. It consumed only 0.9 A so that looked fine. When testing the board it reported memory errors. A bit was bad in bank 2. (RAS 2). Tracked it down to bit D25 which is chip B7. Replaced that one and the board came up with the full 4 meg."

The next Figure shows that everything is now okay:

```

Internet: dongarra@cs.utk.edu
This is version 29.5.04.

      norm. resid      resid      machep      x(1)      x(n)
1.25501937E+00  1.39332990E-14  2.22044605E-16  1.00000000E+00  1.00000000E+00

      times are reported for matrices of order 100
      dgefa      dgesl      total      mflops      unit      ratio      b(1)
times for array with leading dimension of 201
1.383E+01  4.500E-01  1.428E+01  4.807E-02  4.160E+01  2.551E+02  -9.159E-15
1.383E+01  4.500E-01  1.428E+01  4.807E-02  4.160E+01  2.551E+02  1.000E+00
1.383E+01  4.333E-01  1.427E+01  4.813E-02  4.155E+01  2.548E+02  1.000E+00
1.385E+01  4.067E-01  1.426E+01  4.815E-02  4.153E+01  2.546E+02  5.298E+02

times for array with leading dimension of 200
1.383E+01  4.500E-01  1.428E+01  4.807E-02  4.160E+01  2.551E+02  1.000E+00
1.385E+01  4.333E-01  1.428E+01  4.807E-02  4.160E+01  2.551E+02  1.000E+00
1.390E+01  3.833E-01  1.428E+01  4.807E-02  4.160E+01  2.551E+02  1.000E+00
1.385E+01  4.067E-01  1.426E+01  4.816E-02  4.153E+01  2.546E+02  5.298E+02
end of tests -- this version dated 05/29/04
# -

```

© www.cpu-ns32k.net

Fig. 19. The result of the Linpack benchmark on the NS32032 Opus board running with no wait states: 48 kflops.

The result of the Linpack benchmark depends not only on the FPU (which is the same) but also on the data bus width. The 32-bit system achieves 23% more flops. This is the first time that I see a trustable comparison between NS32032 and NS32016. All other tests depend more or less on the performance of the IO system, for example the compilation of Linpack. The next Figure below shows the result.

```

# time f77 linpack.f -o linpack
linpack.f:
MAIN:
matgen:
dgefa:
dgesl:
daxpy:
ddot:
dscal:
idamax:
epslon:
dmxpy:
ran:
second:

real      1:09.9
user      1:01.5
sys       5.6
#

```

© www.cpu-ns32k.net

Fig. 20. The NS32032 compiles the Linpack benchmark.

My Opus system with the NS32016 CPU achieved the following times:

- real 1:46.8
- user 1:26.2
- sys 10.4

Mattis has build his system with an extra Opus partition on the disk. I use only a file on the DOS partition. Maybe that is the reason why his N32032 has a much faster real time. The user time could be a real indication of the true CPU performance. Here NS32032 is 28.6% faster than the NS32016.

Another well known CPU benchmark was Dhrystone. It was very famous in the 1980's and 1990's. Dhrystone tests only the CPU and not the IO system. Therefore it was used to compare CPU performance. Unfortunately the result also depends on the quality of the compiler generated code. Today it is outdated because it is too simple for modern CPUs.

The actual version is 2.1 but we did the tests of our systems with the version 1.1. Dhrystone results come in two flavours. The difference is the use of registers with the **-DREG=register** option. This yields better results = higher numbers.

```

/* Compiler dependent options */
#undef NOENUM          /* Define if compiler has no enum's */
#undef NOSTRUCTASSIGN  /* Define if compiler can't assign structures */

/* define only one of the next two defines */
/*#define TIMES        /* Use times(2) time function */
#define TIME           /* Use time(2) time function */

/* define the granularity of your times(2) function (when used) */
#define HZ 100         /* times(2) returns 1/60 second (most) */

"dhrystone.c" 772 lines, 27341 characters
# cc -DREG=register dhrystone.c -o dhrystone
# ./dhrystone
Dhrystone(1.1) time for 50000 passes = 47
This machine benchmarks at 1063 dhrystones/second
# time dhrystone
Dhrystone(1.1) time for 50000 passes = 48
This machine benchmarks at 1041 dhrystones/second

real    48.5
user    48.4
sys     0.1
#

```

Fig. 21. The screen shot shows the result for the Dhrystone benchmark. The **time** command eats some dhrystones/second away.

The following table shows some NS32016 based results. PC-MX2 is a Siemens computer running also at 10 MHz. The number of wait states is unknown.

Option	0 Wait State	1 wait state	PC-MX2
no Option	744	656	687
-DREG=register	785	700	720

One wait state makes the NS32016 around 13% slower. I guess that the PC-MX2 uses also one wait state because it has a bigger main memory of 4 Mbytes. Then the Siemens compiler must generate a slightly faster code. In this test the NS32032 is 26% faster if I compare 1063 dhrystones/second versus 785 dhrystones/second.

Some other Dhrystone results are shown at the end of the chapter [Systems/Richard](#) . These results are much better for both CPUs. Obviously the Opus compiler is not so good. I don't know the version of it because **cc -V** shows no result.

This chapter was last modified on 29 September 2022. Next chapter: [PC532](#)