

A Retro Computer, built in 1987 and still running!



In 1987, the thing powered on and worked on the first try, with maybe 2000 (?) wire wrap connections!

In 2021, it still runs!!!

The NS32032 was significantly faster than PCs of its day!

Processor: NS32032

- 68 pin leadless chip carrier, 8MHz
- 32 bit data bus; 24 bit address bus; 0 wait-state
- highly-orthogonal C optimized instruction set
- processor family eventually reached RISC-like 1.4 CPI with NS32532

EPROM: 32kB at offset 0

- 4x 2764 (8k x 8bit)
- Intel hex bootloader from boot.q

RAM: 32kB at offset 32kB

- 4x 6264 (8k x 8bit)

Peripherals:

- 16450 serial port
- 32202 vectored interrupt controller
- 32081 floating point unit
- 32082 memory management unit
- DMA address and data switches and LEDs (for bootstrap before bootloader was working)

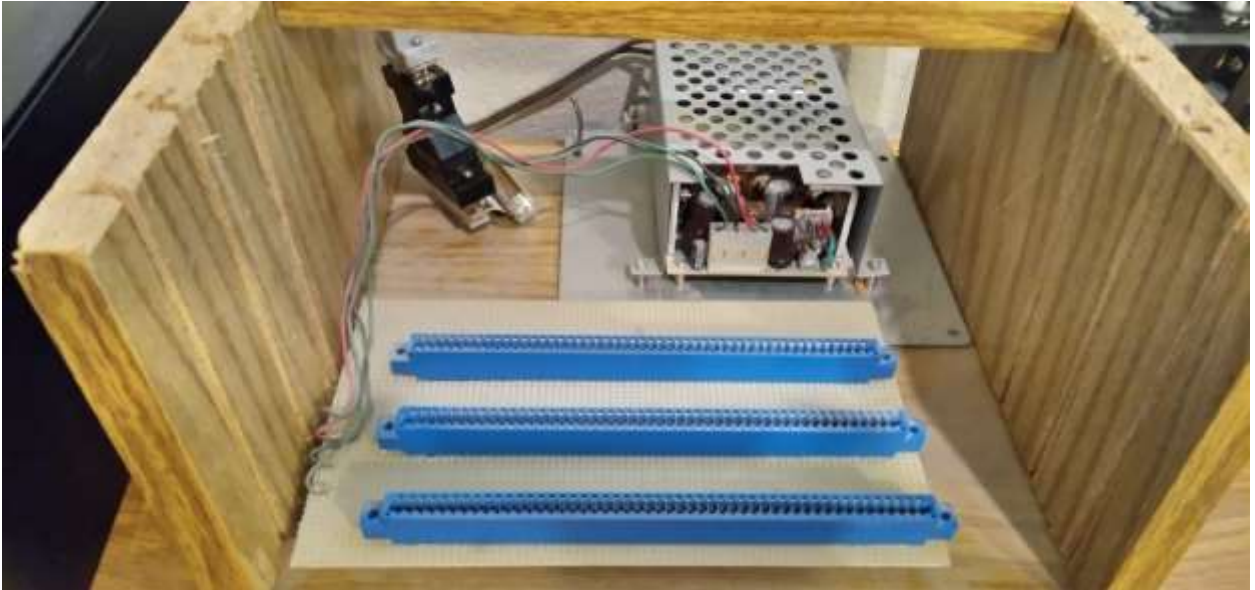
Boards:

- 3x S100 boards, wire-wrap
- Memory/Peripheral, CPU, and DMA boards shown left to right



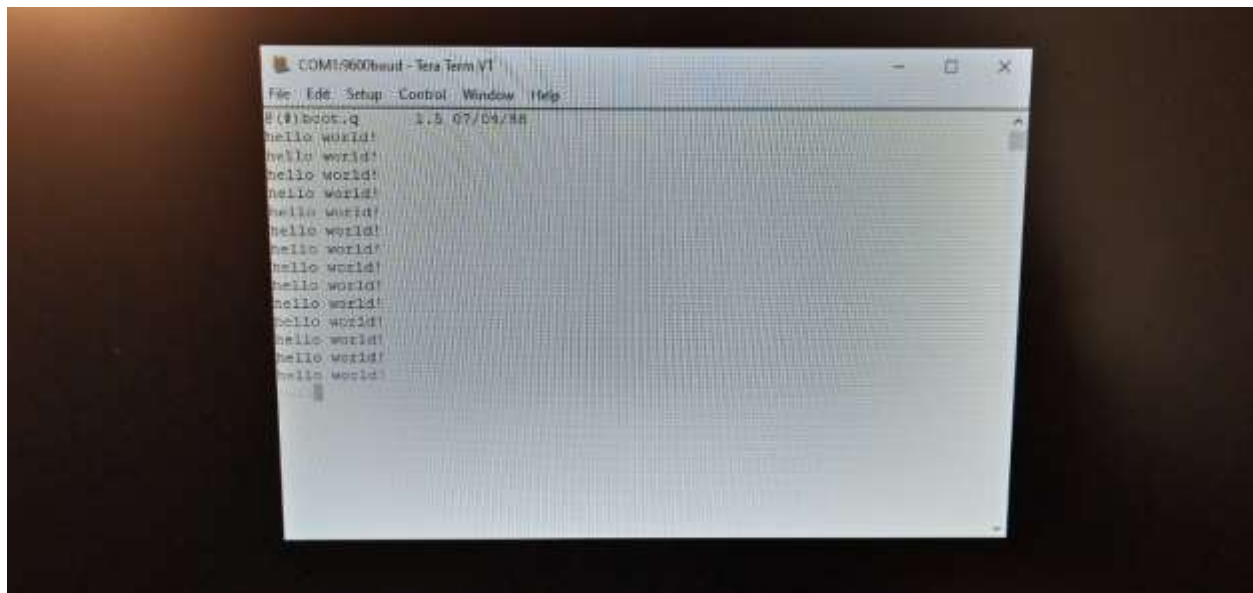
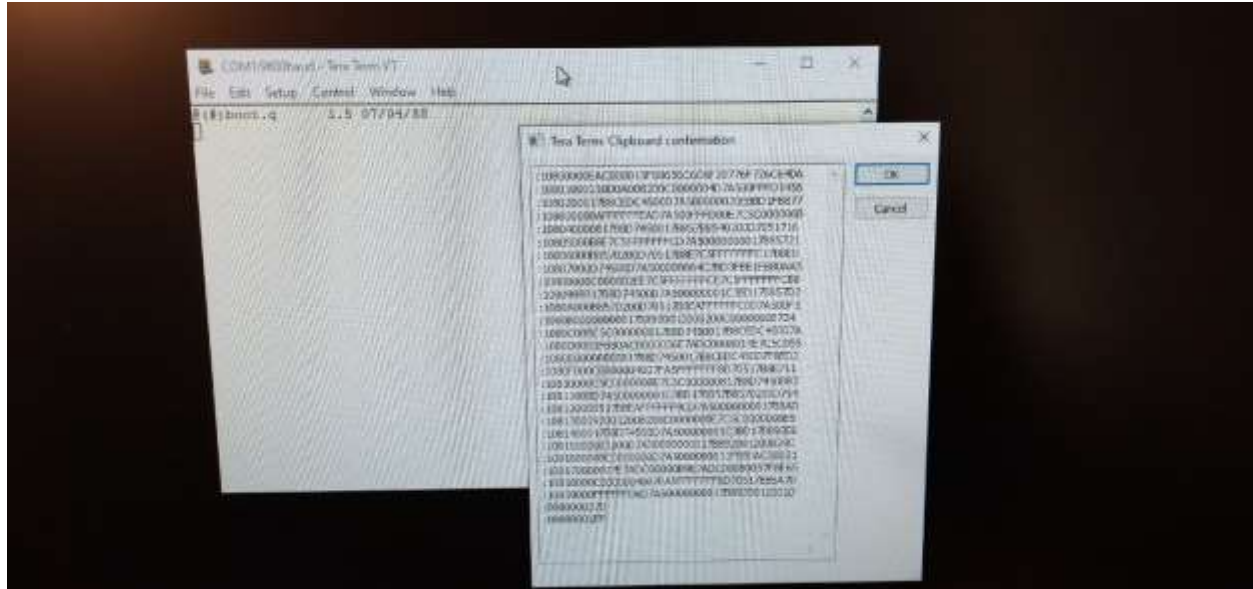
Card cage:

- 3x S100 connectors, wire-wrap
- framed with particle-board college bookshelf shelf, cut by hand
- switching power supply +/-5V, +12V (replaced once since 1987)



Demo:

- In 2021, using a 9600 baud serial port:



Compiler:

- pre-ansi-C yacc and lex source code still compiles!
- qc.exe -- a queer optimization-less binary file compiler (from when "queer" actually meant "queer"! :-)
- intel.exe -- binary file to intel hex converter

Test program:

- source, test.q

```
/****** test.q *****/

entry main

/****** SIO defines *****/

#define SIORBR [0xFFFD00]. /* receive buffer register */
#define SIOTHR [0xFFFD00]. /* transmit holding register */
#define SIOLSR [0xFFFD14]. /* line status register */

#define DR 0x01 /* receive data ready */
#define THRE 0x20 /* transmit holding register empty */

/****** putc *****/

define putc(c)
begin
  repeat
    /* null */
  until SIOLSR & THRE
  SIOTHR := c
end

/****** puts *****/

define puts(s)
begin
  while [s]. do
    call putc([s].)
    s := s + 1
  end
end

/****** main *****/

string hello "hello world!\r\n\0"

define main()
begin
  while 1 do
    call puts(@hello)
  end
end
```

- assembly, debug

cl /EP test.q >test.qq & qc.exe -d -o 32768 <test.qq

| | | |
|----------------------|-------------------------|-------------------------|
| test.q | addr 8(fp),tos | beq 56 {80bd} |
| 8000 | 805a | 80bd |
| branch ffff8000 {0} | movd tos,r0 | movd 0,tos |
| | movd 0(r0),tos | 80c3 |
| | 805f | movd tos,r0 |
| 8005 | movd tos,r0 | 80c5 |
| PROCEDURE putc | movxbd 0(r0),tos | exit [] |
| | 8065 | return 0 |
| enter [],0 | cmpqd tos,0 | 80c9 |
| 800b | beq ffff7f99 {0} | 8000 |
| movd fffd14,tos | 806c | branch d8 {80d8} |
| 8011 | addr @8005,tos | |
| movd tos,r0 | 8072 | |
| movxbd 0(r0),tos | addr 8(fp),tos | 80d8 |
| 8017 | 8078 | PROCEDURE main |
| movd 20,tos | movd tos,r0 | |
| 801d | movd 0(r0),tos | enter [],0 |
| andd tos,tos | 807d | 80de |
| 801f | movd tos,r0 | movd 1,tos |
| cmpqd tos,0 | movxbd 0(r0),tos | 80e4 |
| beq fffffffea {800b} | 8083 | cmpqd tos,0 |
| 8026 | jsr 0(4(sp)) | beq ffff7f1a {0} |
| movd fffd00,tos | adjspd fffffff8 | 80eb |
| 802c | 8090 | addr @804e,tos |
| addr 8(fp),tos | movd r0,tos | 80f1 |
| 8032 | 8092 | addr @80c9,tos |
| movd tos,r0 | movd tos,r0 | 80f7 |
| movd 0(r0),tos | 8094 | jsr 0(4(sp)) |
| 8037 | addr 8(fp),tos | adjspd fffffff8 |
| movd tos,r0 | 809a | 8104 |
| movd tos,r1 | addr 8(fp),tos | movd r0,tos |
| movb r0,0(r1) | 80a0 | 8106 |
| 803e | movd tos,r0 | movd tos,r0 |
| movd r0,tos | movd 0(r0),tos | 8108 |
| 8040 | 80a5 | branch fffffffd6 {80de} |
| movd tos,r0 | movd 1,tos | 80e4 |
| 8042 | 80ab | cmpqd tos,0 |
| movd 0,tos | addd tos,tos | beq 27 {810d} |
| 8048 | 80ad | 810d |
| movd tos,r0 | movd tos,r0 | movd 0,tos |
| 804a | movd tos,r1 | 8113 |
| exit [] | movd r0,0(r1) | movd tos,r0 |
| return 0 | 80b4 | 8115 |
| | movd r0,tos | exit [] |
| | 80b6 | return 0 |
| 804e | movd tos,r0 | |
| PROCEDURE puts | 80b8 | |
| | branch fffffff9c {8054} | 119 bytes generated. |
| enter [],0 | 8065 | |
| 8054 | cmpqd tos,0 | |

- binary, test.i

```
cl /EP test.q >test.qq & qc.exe -o 32768 <test.qq >test.x & intel.exe -o 32768 -s 32768 <test.x
```

```
:10800000EAC00000D88200C0000000D7A500FFFD34
:108010001417B8CEDC4500D7A500000020EBBD1F2B
:10802000B80AFFFFFFEAD7A500FFFD00E7C5C000C3
:10803000000817B8D7450017B857B8540200D7053D
:1080400017B8D7A50000000017B8920012008200F0
:10805000C0000000E7C5C000000817B8D7450017EA
:10806000B8CEDC45001FB80AC0000056E7ADC0001E
:108070008005E7C5C000000817B8D7450017B8CE7F
:10808000DC45007F8EC0000004007FA5FFFFFFF8E5
:10809000D70517B8E7C5C0000008E7C5C00000084D
:1080A00017B8D74500D7A500000001C3BD17B857C2
:1080B000B8570200D70517B8EAF9CD7A50005
:1080C00000000017B89200120068656C6C6F207792
:1080D0006F726C64210D0A008200C0000000D7A5F9
:1080E0000000000011FB80AC0000027E7ADC00080F3
:1080F0004EE7ADC00080C97F8EC0000004007FA5A0
:10810000FFFFFFFF8D70517B8EAF9FD6D7A50096
:0981100000000017B892001200F3
:008000037D
:00000001FF
```

Source code:



32032.zip