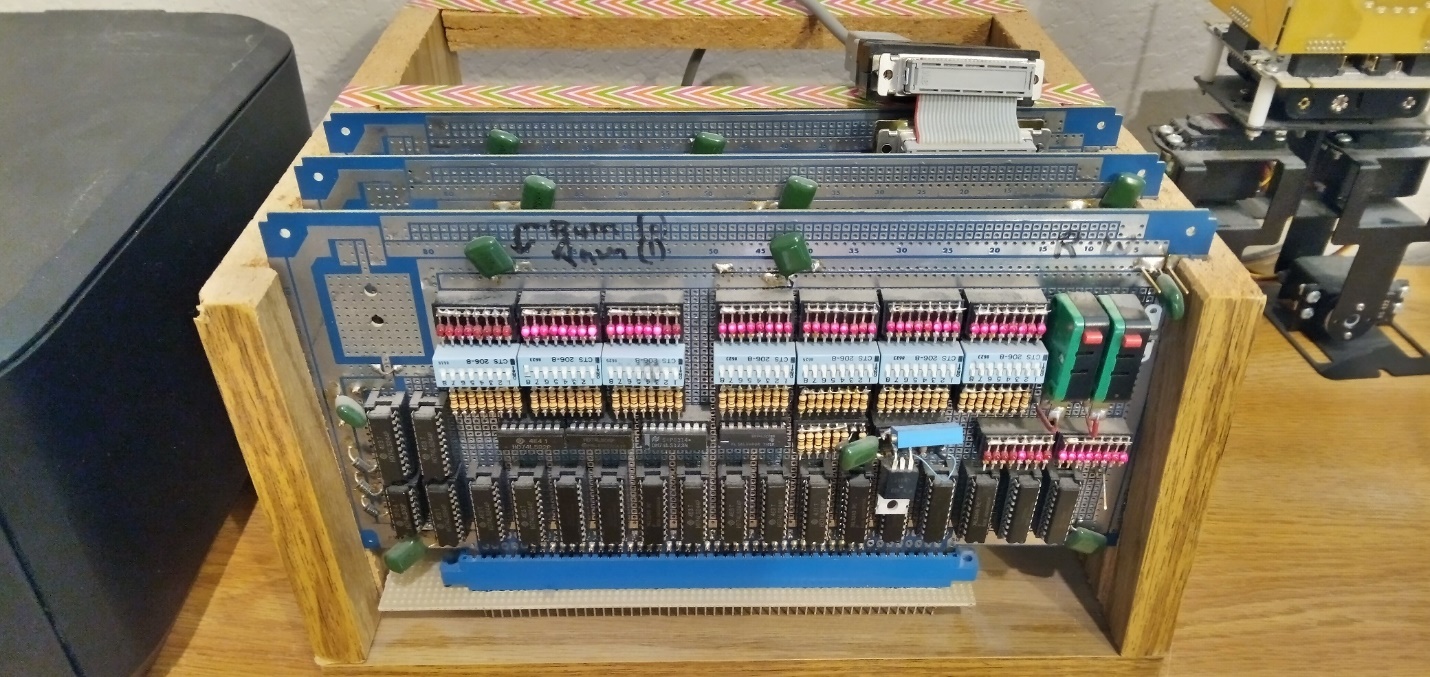
A Retro Computer, built in 1987 and still running!

See: https://github.com/rtestardi/retro



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)

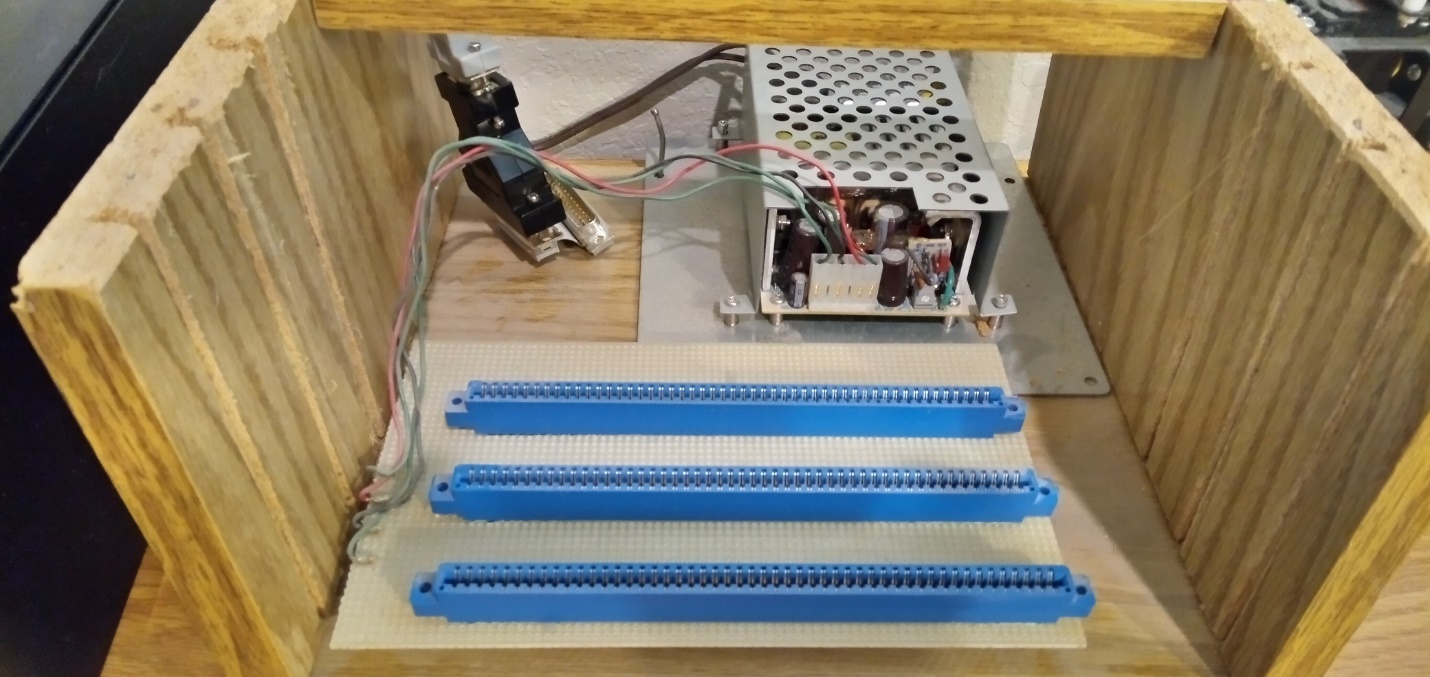
# Boards:

* 3x S100 boards, wire-wrap
* Memory/Peripheral, CPU, and DMA boards shown left to right  
  A picture containing text, electronics, circuit

  Description automatically generated  
  A picture containing text

  Description automatically generated

# 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:  
  A picture containing text, receipt

  Description automatically generated  
    
  A picture containing text

  Description automatically generated

# 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

8000

branch ffff8000 {0}

8005

PROCEDURE putc

enter [],0

800b

movd fffd14,tos

8011

movd tos,r0

movxbd 0(r0),tos

8017

movd 20,tos

801d

andd tos,tos

801f

cmpqd tos,0

beq ffffffea {800b}

8026

movd fffd00,tos

802c

addr 8(fp),tos

8032

movd tos,r0

movd 0(r0),tos

8037

movd tos,r0

movd tos,r1

movb r0,0(r1)

803e

movd r0,tos

8040

movd tos,r0

8042

movd 0,tos

8048

movd tos,r0

804a

exit []

return 0

804e

PROCEDURE puts

enter [],0

8054

addr 8(fp),tos

805a

movd tos,r0

movd 0(r0),tos

805f

movd tos,r0

movxbd 0(r0),tos

8065

cmpqd tos,0

beq ffff7f99 {0}

806c

addr @8005,tos

8072

addr 8(fp),tos

8078

movd tos,r0

movd 0(r0),tos

807d

movd tos,r0

movxbd 0(r0),tos

8083

jsr 0(4(sp))

adjspd fffffff8

8090

movd r0,tos

8092

movd tos,r0

8094

addr 8(fp),tos

809a

addr 8(fp),tos

80a0

movd tos,r0

movd 0(r0),tos

80a5

movd 1,tos

80ab

addd tos,tos

80ad

movd tos,r0

movd tos,r1

movd r0,0(r1)

80b4

movd r0,tos

80b6

movd tos,r0

80b8

branch ffffff9c {8054}

8065

cmpqd tos,0

beq 56 {80bd}

80bd

movd 0,tos

80c3

movd tos,r0

80c5

exit []

return 0

80c9

8000

branch d8 {80d8}

80d8

PROCEDURE main

enter [],0

80de

movd 1,tos

80e4

cmpqd tos,0

beq ffff7f1a {0}

80eb

addr @804e,tos

80f1

addr @80c9,tos

80f7

jsr 0(4(sp))

adjspd fffffff8

8104

movd r0,tos

8106

movd tos,r0

8108

branch ffffffd6 {80de}

80e4

cmpqd tos,0

beq 27 {810d}

810d

movd 0,tos

8113

movd tos,r0

8115

exit []

return 0

119 bytes generated.

* 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  
  :1080B000B8570200D70517B8EAFFFFFF9CD7A50005  
  :1080C00000000017B89200120068656C6C6F207792  
  :1080D0006F726C64210D0A008200C0000000D7A5F9  
  :1080E000000000011FB80AC0000027E7ADC00080F3  
  :1080F0004EE7ADC00080C97F8EC0000004007FA5A0  
  :10810000FFFFFFF8D70517B8EAFFFFFFD6D7A50096  
  :0981100000000017B892001200F3  
  :008000037D  
  :00000001FF