Building a Computer

For fun. *There is no profit.*

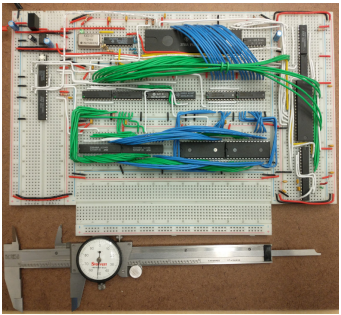
Ethan Blanton

April 2, 2022

Computer Homebrew

When I say “build a computer,” I don’t mean a gaming rig.

I mean build a computer from chips.



Difficulty

Building a computer is suprisingly easy to do!

It will require specific knowledge you may not (yet) have.

You can absolutely learn it, though!

It will also require patience and attention to detail.

The most relevant UB courses are probably:
CSE 241, CSE 341, CSE 321

Expense

The computer components to build a computer are relatively inexpensive.

However, you're going to need a lot of:

- Solderless breadboard area or PCB area
- Wire
- Miscellaneous connectors

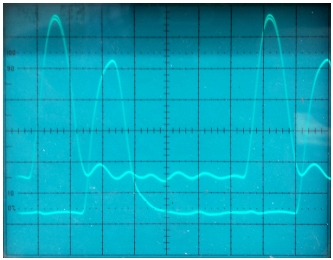
Those parts are not as cheap, due to quantity!

Plan to spend maybe $100, much of which is reusable.

Equipment

You will very much want some equipment.

- Wirestrippers/cutters
- Small pliers
- A ROM writer
- A logic analyzer
- An oscilloscope



An Arduino or other dev board can stand in for much of this.

Plan to spend maybe $100 more, all of which is reusable.

Software

Since you built the computer, there is no software!

You can find existing implementations for many platforms of:

- Programming languages (*e.g.*):
  - BASIC
  - Forth
  - Assemblers
  - Debug “monitors”
- Disk operating systems (*e.g.*):
  - CP/M
  - OS-9

You will have to do some porting.

What is a homebrew computer like?

You'll be building a retrocomputer using some older architecture.

The reasons for this are:

- Physically larger components with fewer pins
- Slower clock speeds
- Simpler support hardware requirements

Many 1980s CPUs need nothing but an oscillator and a single ROM chip!

Size and Speed

Most reasonable CPUs have 8 or 16 bit register widths.

Their address space is probably 16–24 bits.

Expect 16 kB–1 MB of memory.

For homebrew, 1 MHz is a fast clock, and 10 MHz is very fast.

Some CPUs are very efficient at these clock speeds.

You're still not going to win many races.

Comparisons

Compare to:

- Apple][(1 MHz, 16–128 kB RAM)
- Commodore 64 (1 MHz, 64 kB RAM)
- Sega Master System (4 MHz, 24 kB+ RAM)
- Super NES (3.5 MHz, 172 kB+ RAM)

(Note that the last three have significant video and audio hardware support.)

Interaction

Your first computer probably won't have a keyboard and monitor.

Many early computers used a serial terminal for interaction.

Your modern laptop/desktop provides this via a USB device!

This allows you to punt on:

- Producing video signals
- Using a large number of I/O lines
- Keeping a screen in (limited!) RAM

Image Copyright SparkFun, CC BY 2.0

Storage

Storage is relatively hard.

Disk interfaces are complicated.

USB is way too fast for a homebrew.

You will probably either:

- Be limited to ROM
- Emulate something on a microcontroller dev board
- Load and save via the serial console

The Net System

Keep your expectations in line!

Your system will be:

- Slow
- Short on RAM
- Limited in program size
- Text-only

But a lot of fun and a huge learning experience!

Some History

I really like unusual and defunct architectures.

I considered several architectures for a homebrew and selected the strangest.

That has been a terrible idea and I love it.

I started building my homebrew in Spring of 2020.

Things stalled for about a year, but are back in motion.

The Processor

I am using the TMS 9900 processor.
It is:

- 16-bit register width
- 15-bit address bus
- Word-addressed

It has:

- 64 pins
- No general purpose registers
- Memory-mapped “registers”
- Strange high-speed serial I/O

Size and Speed

As mentioned, the processor is 16 bit.

It runs at 3 MHz.

Most operations require 4–10 clock cycles.

I have 16 kB ROM and 16 kB RAM in modern-ish chips.

- The ROM chips are Atmega AT27C64 8kx8 EEPROMs.
- The RAM chips are 8kx8 6264 JEDEC-compatible SRAM.

Each of these are in pairs to form 16-bit words.

I will eventually have a full 64 kB or more of RAM.

Input and Output

The TMS 9900 has associated I/O chips.

I am using:

- The TMS 9902 serial UART for a “console”
- The TMS 9901 parallel I/O and interrupt controller

These communicate via the CRU interface.

It is an approximately 1.5 Mbps serial bus.

(Kind of like SPI, but weird.)

Storage

I have no mass storage.

I plan to include:

- Floppy disks (probably 5.25″ diskettes)
- Mass storage (probably IDE)

IDE is an interface from the IBM 286 era.

You know its descendants as SATA.

IDE is attractive on this system because IDE is 16 bit!

Sound and Video

I have no video or audio.

I plan to include:

- An 80x25 text video output
- FM synthesized audio

Software

I am aware of TMS 9900 implementations of:

- A ROM monitor (TIBUG)
- BASIC (Cortex BASIC)
- Forth (Fig-FORTH, TurboForth)

I will probably write:

- A line editor
- An assembler
- A disk operating system
- A Lisp or Scheme?

Time Frame

This will probably take years.

Where Do You Start?

There are numerous kits and plans!

You don’t have to build from scratch the first time.

Many kits are extensible.

There are also many descriptions of completed projects.

Search “homebrew computer” and start reading!

CPUs

There are many appropriate CPUs.

Some things to consider:

- Multiplexed I/O
- Data and address bus width
- Clock speed

8 bit with 16 bit address space is probably the sweet spot.

Kits and Projects Named Here

I am not endorsing any kits or projects mentioned here.

I have not built any of them!

Your mileage may, as they say, vary.

The Zilog Z80

The Z80 is a simplified and expanded Intel 8080-compatible CPU.

Modern Z80 processors are static logic:
The clock can run at 0 Hz without failure!

They have a non-multiplexed bus: 8 bit data, 16 bit address.

There are many kits and projects:

- RC 2014
- Steve Ciarcia's "Build Your Own Z80 Computer"

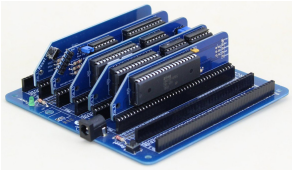


Image Copyright RFC2795 Ltd, Fair Use

The Z80 in History

Many, many computers used the Intel 8080 or Zilog Z80!

The Altair 8800, the TRS-80 line, the Osborne 1, …

The CP/M disk operating system runs on the Z80.

You can easily find:
BASIC, Forth, C, Pascal, games, applications, *etc.*!

Check out http://z80.info/homebrew.htm.

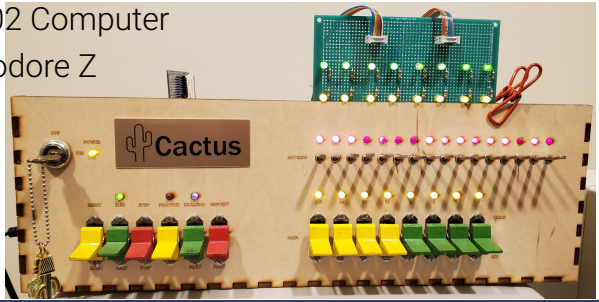
The MOS 6502

The MOS 6502 is a simplified and expanded Motorola 6800.

The 65C02 is also static logic, with a non-multiplexed bus!

Some kits and homebrew projects:

- Ben Eater's Build a 6502 Computer
- The Cactus by Commodore Z



Cactus image Copyright 2022 Commodore Z
(used with permission)

The MOS 6502 in History

Many famous computers used the 6502, too!

The Apple][, Commodore 64, NES, …

You can easily find:
BASIC, Forth, Pascal, Lisp, games, applications, …

Check out http://www.6502.org/

The Motorola 6809

The Motorola 6809 is also an expanded Motorola 6800.

It is dynamic logic, with a minimum clock of about 100 kHz.

Used in the Tandy Color Computer, Dragon 32 and 64, Vectrex.

OS-9 provides a capable disk operating system!

Image copyright Konstantin Lanzet CC BY-SA 3.0 (cropped)

Other CPUs

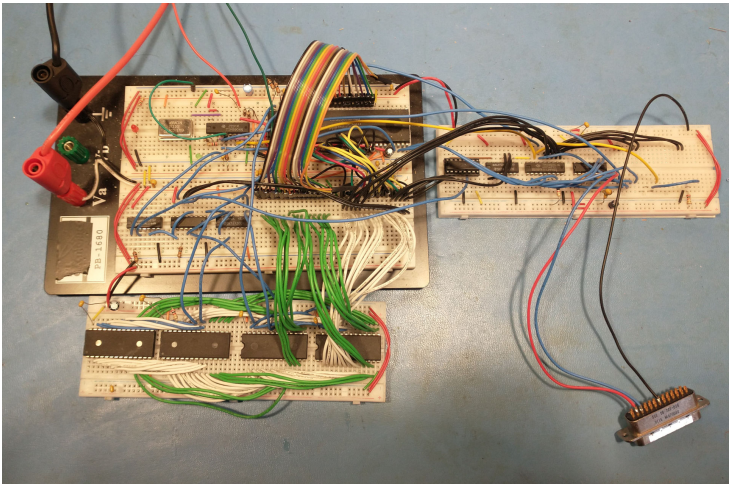
There are other CPUs that are more complicated because of: Multiplexed buses, faster clocks, less software, *etc.*

- WDC 65C816 (Apple IIGS, Super NES)
- TMS 9900 (TI 99/4 and 99/4A, Geneve 9640, my computer)
- Motorola 68000 (Sega Genesis, Sun 1, Macintosh)
- RCA 1802 (COSMAC ELF)

You probably shouldn't start with one of these.

You might enjoy seeing some of the existing homebrews, though!

Why I Started Over



Questions?

Please ask questions!

Resources

Some valuable links:

- The Cactus: http://commodorez.com/cactus.html
- Ben Eater's projects: https://eater.net/
- The RetroBrew Computers wiki: https://www.retrobrewcomputers.org/doku.php

License

Share the love of retrocomputing.