

Re: homebrew computer – where to start?

Source: <http://sci.tech–archive.net/Archive/sci.electronics.basics/2005–09/msg00113.html>

- *From:* Jonathan Kirwan <jkirwan@xxxxxxxxxxxxxxxx>
 - *Date:* Sat, 03 Sep 2005 19:34:18 GMT
-

On Sat, 3 Sep 2005 20:10:11 +0200, "petrus bitbyter"
<pieterkraltlaatditweg@xxxxxxxxxxxxxxxx> wrote:

>A starting point is a matter of choice. Some early calculating machines used
>cogwheels and functioned pure mechanically. The first electronic computers
>used electron tubes by the dozens and required more power than your mains
>connection can provide. The first computer I worked on was built with
>discrete transistors, ferrite cores and lots of wire all packed in five 19"
>rack enclosures higher than a man's length. The first one I built for myself
>has a Z80 processor on 4MHz, 2k of RAM and 2k of EPROM. The latter contained
>a monitor program derived from the NASCOM. I built, also from scratch, a
>separate I/O card for it containing a UART that communicated with a dumb
>terminal. The next step was an I/O card that could write to – and read from
>cassette tape. Still works when I hook up a PC running a terminal emulator.

My first build-up was the Altair 8800 from a kit of parts. Years
before the Z80 was available. I remember looking with envy at the
8085 when it finally came out, because it had been so simplified from
the complex clocking monster that the 8080 was. And that was before
the Z80. The Z80 was so simple to design a board for. Nice!

>At about the same time you could buy Apples or one of its clones. You could
>buy an empty board and fill it with components. Some time later you could
>buy empty PC– and peripheral boards to do the same. AFAIK the last computer
>building that required soldering. These days you can assemble your own
>machine even without a screwdriver. IMHO you can't go back but to the first
>microprocessors like the 8085, Z80, 6800, 6502 and some others I don't know
>well. Some stuff, like the Z80, is still available. Don't know about the
>others.

Well, I've some 65SC02's laying around. And I'm sure I could find an
8080A and a clocking chip for it, if I looked. ;)

>Nevertheless, I don't think this is the way to go. The old times
>will not come back you know. I advise to look around in the world of
>microcontrollers. They have processor, RAM, ROM and I/O in one package but
>fiddling with the bits, assembler programming and even soldering are still
>required. There is a wide range of them from six pins SOT–23 to forty and
>more pins DIP all with eight bits processors. The latter at least as

Re: homebrew computer – where to start?

>powerfull as the old Z80 and its contemporaries. If you want more there are
>much more powerfull sixteen bits micros available as well. You'll find more
>info then you ever can read on the web but you still can do plenty of things
>others have not done before.

The only problem I have with your answer to the OP is that the OP specifically asked this:

>> i'd like to try to build a computer from scratch – build my own
>> processor, etc. – to gain a greater familiarity with the underlying
>> technology.

It was the 'build my own processor' part of the above phrase that really sounded to me like asking about how to design an ALU with latched input paths and output, access to registers for latching, a memory address latch, tri–state buffers on a common internal bus, etc. I could have been wrong, though.

But that is a great deal of fun to learn about and do -- especially in this day of cheap FPGA boards from a variety of sources in attractive variations on a theme. It's almost dead simple to get started now writing VHDL and getting a simple cpu up and going, entirely of your own making. You don't even have to worry much about routing and floor planning, as the tools will do a uniformly lousy job automatically but one that still gets something working for you. And with the huge resources available on these chips today, who cares if it all gets ruthlessly squandered by the planner while you are learning VHDL?

It will be interesting to see if the OP is really more about learning the internal basics of how a cpu works inside or more about learning the larger picture of a cpu–memory–i/o system. There are so many choices now that can teach at almost any level of interest -- from internal cpu design all the way up to pasting down one–chip–wonders that include all the I/O, code space, data space, and cpu processing power and leave very little left to learn about.

Jon
.

- **Follow–Ups:**

- ◆ **Re: homebrew computer – where to start?**

◇ From: jm

- **References:**

- ◆ **homebrew computer – where to start?**

◇ From: racter

- ◆ **Re: homebrew computer – where to start?**

◇ From: petrus bitbyter

Re: homebrew computer – where to start?

- Prev by Date: [**Re: homebrew computer – where to start?**](#)
- Next by Date: [**Re: Hitachi TV power problem**](#)
- Previous by thread: [**Re: homebrew computer – where to start?**](#)
- Next by thread: [**Re: homebrew computer – where to start?**](#)
- Index(es):
 - ◆ [**Date**](#)
 - ◆ [**Thread**](#)