RAM Access 4004

It's hardly possible to go back further than the Intel 4004, at least if you are only considering single chip CPUs (at the time of its conception in the early 70s, there were already famous multi-chip CPUs with comfortable orthogonal instruction sets, notably the PDPs). Indeed, it was the *first* widely available single chip CPU. This little thing was a 4-bit CPU with some strange quirks, which we will explore further. Overall, it bears little to no resemblance to its successor in name, the Intel 8008 (except for the internal stack, which both had–I will cover that in another posting).

But let's just look at the code for writing a value of 3 into the memory location at 5 first:

```
FIM P0, 5; load address 05h into pair R0,R1
SRC P0   ; set address bus to contents of R0,R1
LDM 3    ; load 3 into accumulator
WRM      ; write accumulator content to memory
```

That looks a bit strange.

As a 4 bit CPU, the 4004 has 4 bit wide registers and addresses 4 bit nibbles as words in memory. It has only one accumulator on which the majority of operations is performed, but *sixteen* index registers (R0-R15).

Those index registers are handy for accessing memory: Besides loading values directly from ROM, an instruction exists to load data indirectly, which sets the address bus to the ROM cell's content. Another instruction performs an indirect *jump* instead. Other than that, you can just increment index registers, albeit there is the interesting "ISZ" instruction that not only increments, but also branches if the result is not 0.

Because the 4004 uses 8 bits to address the 4 bit nibbles, every two consecutive index registers form a pair, which is then used for memory references.

Note that I explicitly said ROM above. This is because in the 4004 architecture, ROM and RAM are actually vastly different beasts, at least from the assembly programmer's perspective. You can *not* directly access RAM. It always involves index register pairs, manually sending their content to the address bus (with a strangely named instruction "SRC", which for some reason spells out *send register control*) and then issuing another instruction which transfers from or to the accumulator.

Interestingly, accessing regular RAM nibbles is not your only choice among the transfer instructions. You can also fetch from and to I/O ports. But the CPU does not have any direct I/O port, instead they are available on both RAM and ROM! You can also read and write "RAM status characters", which to me look like plain regular RAM cells within another namespace. If someone knows, I'd love to hear what they were used for (and if they maybe did behave differently to normal RAM).

Take a look at the data sheet. Within its only 9 pages, the instruction set is depicted on page 4 and 5. Especially in the light that fairly reasonable orthogonal instruction sets appear to have been available in multi-chip CPUs, this first single-chip CPU is clearly a strange specialization towards the desk calculator it was meant for (the Busicom 141-PF). It has the aforementioned index register-centered RAM access, separate ROM (although there is a transfer instruction which strangely refers to some optional "read/write program memory"), a *three* level internal stack which is almost useless

for general purpose programming and a lone special purpose instruction for “*keyboard process*” (KBP).

Original 4004 CPUs go from anything from a few to a few thousand dollars on eBay, depending on their packaging and revision. If you’d like to, you can instead play around with a virtual one in this java-script based, fully fledged assembler, disassembler and emulator, or read the rescued source code of the Busicom 141-PF calculator. There’s lots more of schematics, data sheets and other resources on the Intel’s anniversary project page.