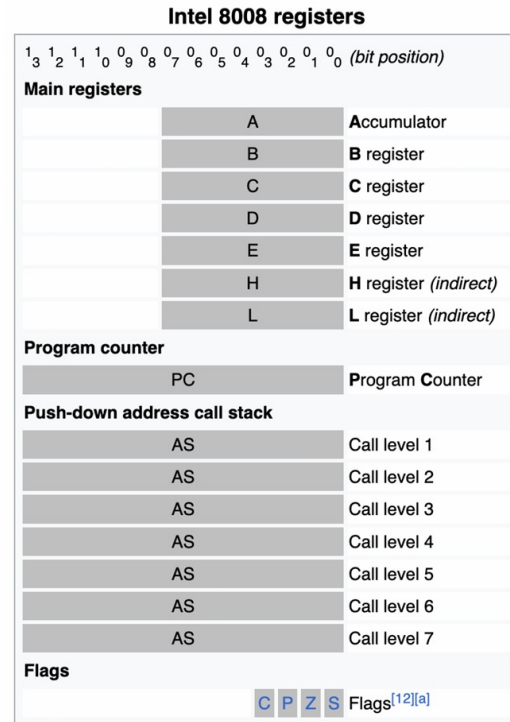
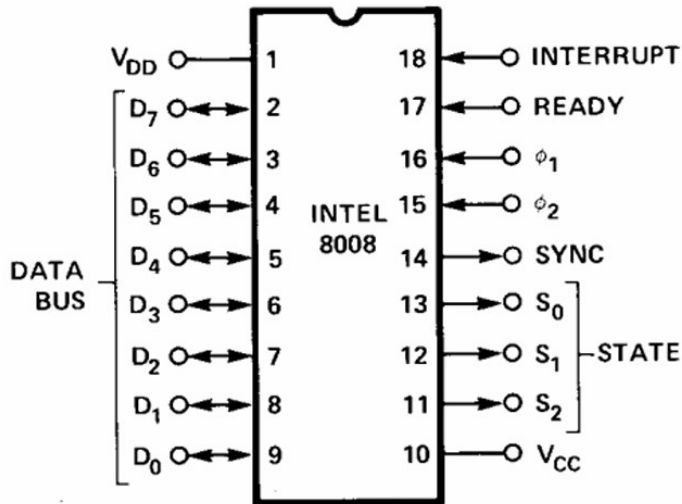
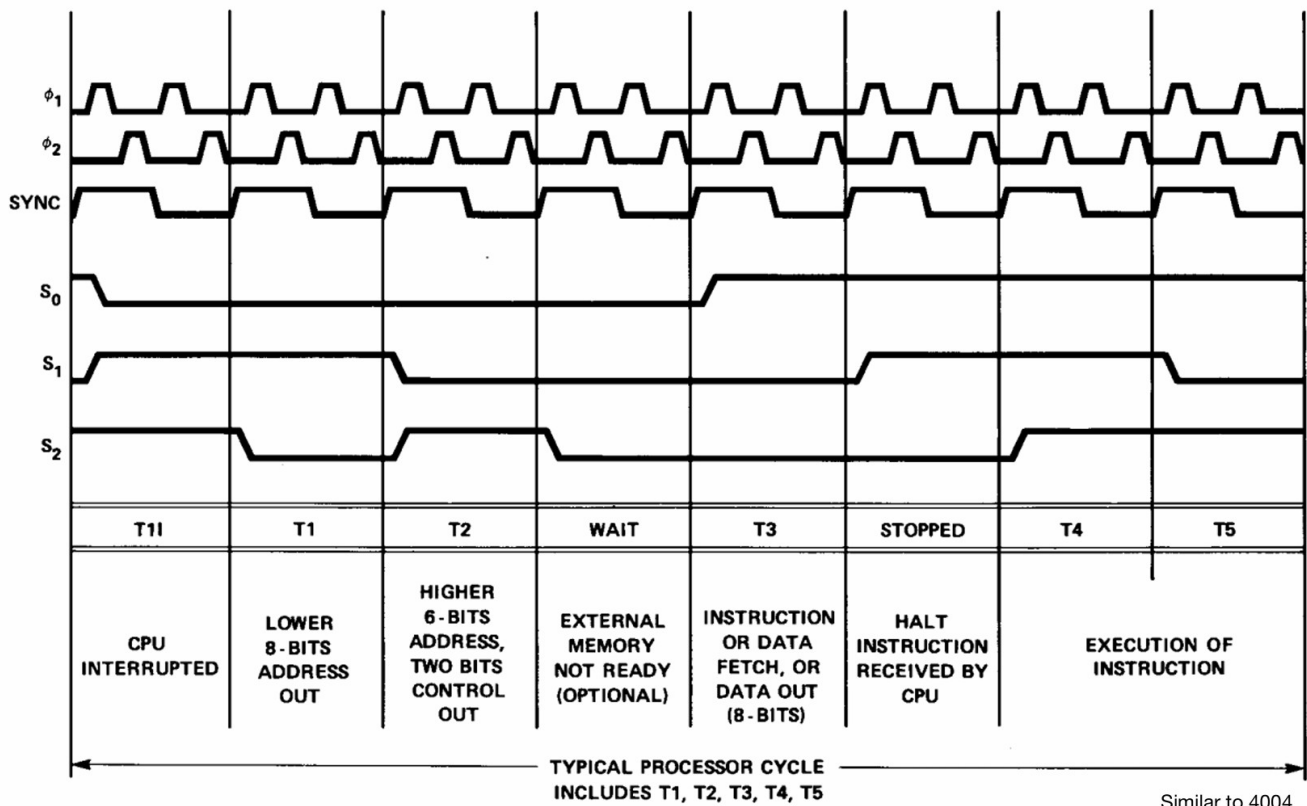


HARDWARE



- 8008 was housed in a 18-pin ceramic package.
- 8008 requires two power supplies: +5V and -9V. Please note there is no GND pin.
- Unlike 4004, **8008's inputs and outputs are TTL-compatible**. For inputs, datasheet recommends +5V pull-up resistors to reach V_{ih} levels.
- **There is no RESET input to 8008**. When 8008 is powered-on, it resets PC to 0x0000, then generates a HALT instruction and halts at address 0x0000. To kick-start 8008 out of this halted state, interrupt must be asserted and the initial instruction fed to it. See next for how this works.
- Interrupt implementation is very basic. When INTERRUPT is asserted, 8008 will take an external instruction from the databus and execute it. This instruction can be any instruction. For **interrupt functionality**, this instruction was a “**CALL to subroutine**” which saved return address to stack so 8008 can go back to the interrupted address using RET. For **reset purposes**, this instruction was usually a single byte **RESTARTx** instruction which jumped to predetermined addresses or **JMP xxxx** or even **NOP**. In the NOP scenario, 8008 would continue executing at address 0x0000 which was mapped to ROM, although in Erturk's opinion, NOP method is not recommended because PC is reset to 0x0000 only when power is cycled. As a result, for subsequent resets without power cycle, NOP would just continue executing.
- 8008 required multiple clock cycles to execute instructions. A typical Processor Cycle (also called Machine Cycle) included multiple states: **T1/T2 for address**, **T3 for instruction fetch**, then **T4/T5 for execution**. SYNC signal indicated the beginning of each processor cycle. This multicycle implementation was similar to 4004.



USES and SOFTWARE (That I/Erturk know of):

INTELLEC 8/MOD 8

Intel provided INTELLEC 8/MOD 8 computers for development. These included a ROM-resident monitor code and Assembler/Editor on paper tapes. Retroschild4004 runs the resident monitor-code unfortunately I (Erturk) could not find the paper tape code to make the assembler/editor software alive.

SCELBI (SCientific-ELectronics-BIology)

Computer Consulting was a personal computer hardware and software manufacturer founded in 1973 by Nat Wadsworth and Bob Findley. Their first product was the SCELBI-8H microcomputer powered by the intel 8008 microprocessor. (From wikipedia)

SCELBAL (SCientific ELementary BASIC Language).

This software was distributed in book format for 8008 and 8080. The code was very well commented. SCELBAL had floating point support, math functions. Programs were saved on cassette. The source was made opensource by the author Mark G. Arnold (MGA) (with late Nat Wadsworth). It is also worth mentioning Mark's late wife Karolyn as she was very supportive and instrumental in recovering SCELBAL/MEA code together w/ Mark.

<https://www.scelbi.com/>

The introduction in the assembly code is worth reading:

```
;;; SCELBAL is the only open-source, floating-point
```

```
;;; high-level language ever implemented on Intel's first
;;; general-purpose microprocessor, the 8008. It was
;;; published in book form:
;;;
;;; SCELBAL: A Higher-Level Language for 8008/8080 Systems
;;;
;;; (Tiny BASIC only used 16-bit integers; the MCM\70
;;; was a closed system; calculators implemented with 8008
;;; were floating-point, but not high-level.)
```

<EA (Monitor Editor Assembler)

MEA was an assembly editor/assembler/monitor running on SCELBI for 8008 code development. It used cassette as a storage device to assemble the source code. (Side note, I'm working w/ MGA to get this released on Retroschild8008. Once I clean up the code and get his final approval, we'll make it available.

<https://www.scelbi.com/software>