CPU Dojo 1

Journey into the ever growing CPU that rules from the centre of the Ultraworld

In the language of your choice create a virtual CPU which will have 2 registers and 16 *items* of memory. Using the list of operations listed below create a program which does the following:

- store the number 100 into register *a*
- add the number 7 to register *a*
- store the value of register *a* into memory address 15
- stop the program

Notes:

- Your program will be stored directly in memory from address 0.
- The CPU's program counter will start at 0 and should increase after each operation by the length of the operation stored at that memory location.
- Notice the length of the operations. An operation with a length of 2 will store a value required by the operation in the next memory address.
- You may want to write some code to inspect the contents of your registers and memory after each iteration of the run loop.

Registers:

- a
- program counter

Memory size: 16 *items* in length of non floating point number types only!

Operations:

OpCode	Length	Name	Description
0	1	BRK	Stops the program
1	2	LDA	Load the value in the next memory address into register A
2	2	ADC	Add the value in the next memory address to the value in register A
3	2	STA	Store the value of register A into the memory location specified by the value in the next memory address

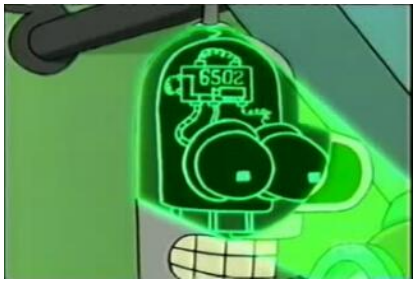
The equivalent program in 6502 Assembly:

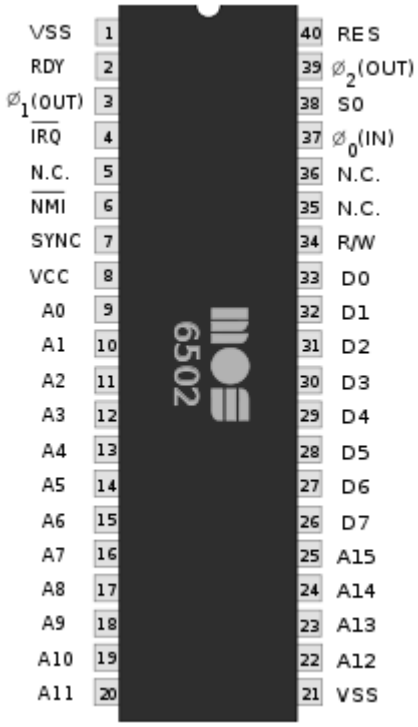
```
LDA #$64      ; 100 = 64 in Hexadecimal
ADC #$07
STA $15
BRK
```

Try the above assembly in a browser and learn more here: http://skilldrick.github.io/easy6502/

MOS 6502

The 6502 processor was massive in the seventies and eighties, powering famous computers like the BBC Micro, Atari 2600, Commodore 64, Apple II, and the Nintendo Entertainment System. Bender in Futurama has a 6502 processor for a brain. Even the Terminator was programmed in 6502.





MOS 6502 registers

$^{1}_{5}$ $^{1}_{4}$ $^{1}_{3}$ $^{1}_{2}$ $^{1}_{1}$ $^{1}_{0}$ $^{0}_{9}$ $^{0}_{8}$ $^{0}_{7}$ $^{0}_{6}$ $^{0}_{5}$ $^{0}_{4}$ $^{0}_{3}$ $^{0}_{2}$ $^{0}_{1}$ $^{0}_{0}$ *(bit position)*

Main registers

	A	**A**ccumulator

Index registers

	X	**X** index
	Y	**Y** index
0 0 0 0 0 0 0 1	SP	**S**tack **P**ointer

Program counter

PC		Program **C**ounter

Status register

N V - B D I Z C		**P** Processor flags

Apple IIe

BBC Micro

Atari 2600

Atari 800

Commodore VIC-20

Commodore 64

Nintendo Family Computer

Ohio Scientific Challenger 4P

See blog post and GitHub repo for more information:
http://www.8bitpickles.com/writing-a-cpu-emulator-as-a-dojo/
https://github.com/timpickles/cpu-dojo