

# CPU Dojo 3

## Subroutines

In this dojo we're going to add some new opcodes for jumping to subroutines and jumping back again. This will then give us the ability to create methods to call a number of times.

To add subroutines we're going to need to add a new register to hold a reference to the stack pointer. In a 6502 CPU the stack has a reserved memory location and the stack pointer will point to the last entry in that memory space. Each time something is pushed onto the stack the stack pointer is decreased and therefore is increased when an item is popped.

For ease of use on constructing our CPU we're going to make the stack be the last section of our memory. The stack pointer will point to the very last *item* in memory (array length - 1). Also for simplicity we're not going to have a limit to the stack and we'll just need to make sure the amount available is enough to fit your application, the stack and any other reserved memory space such as the ASCII text section we reserved in Dojo 2. For this reason I'm suggesting a memory space of 640 *items*, which should be enough for anybody.

### Registers:

- a
- x
- y
- program counter
- flags (*we're just going to be using the equal bit flag*)
- stack pointer - *start it at the last memory address, in our case this will be 639*

**Memory size:** 640 *items* in length

**Operations:** We're building on top of the operations we defined in the previous two dojo's so the new operations will start at 11.

| OpCode | Length | Name | Description                                                                                                                                                                                                                                                                                                        |
|--------|--------|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 11     | 2      | JSR  | This operation does the following: <ul style="list-style-type: none"><li>• stores the current program counter in the memory address pointed to by the stack pointer</li><li>• subtracts 1 from the stack pointer</li><li>• sets the program counter to be the value -1 stored in the next memory address</li></ul> |
| 12     | 1      | RTS  | This operation does the following: <ul style="list-style-type: none"><li>• adds 1 to the stack pointer</li><li>• sets the program counter to be the value stored in the memory address +1, pointed to by the stack pointer</li></ul>                                                                               |

See blog post and GitHub repo for more information:

<http://www.8bitpickles.com/writing-a-cpu-emulator-as-a-doj/>

<https://github.com/timpickles/cpu-doj>