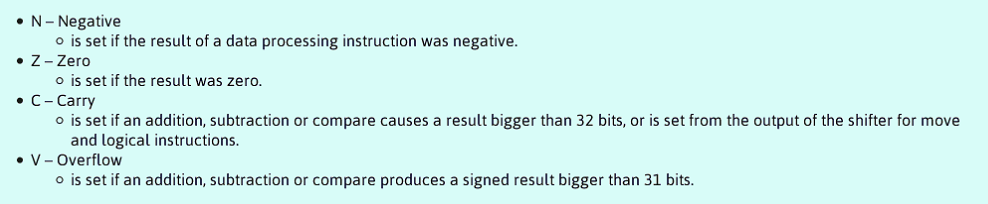
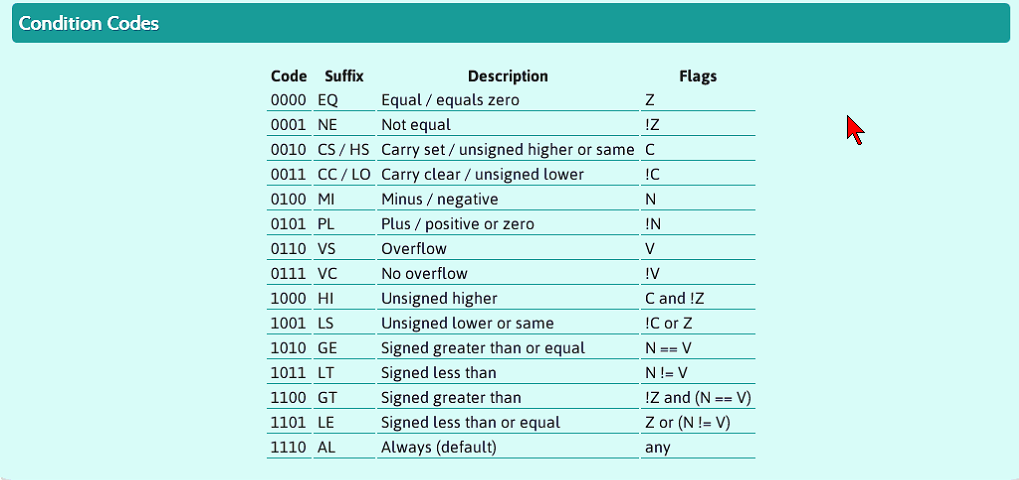
10/17/16 Assembly

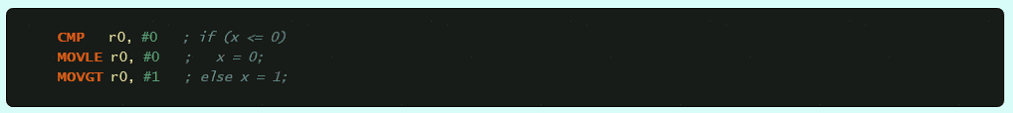
Class Notes

Condition codes for flags:





* Condition codes are appended to code for flag set
  + Ex Asm)



* + More ex) 
* If 5 and 5 are compared, program subtracts it so it == 0; Zero flag is set and lets the program know that it’s equal.
  + Ex Asm) r0 = 5, r1 = 5

cmp r0, r1 //Compare

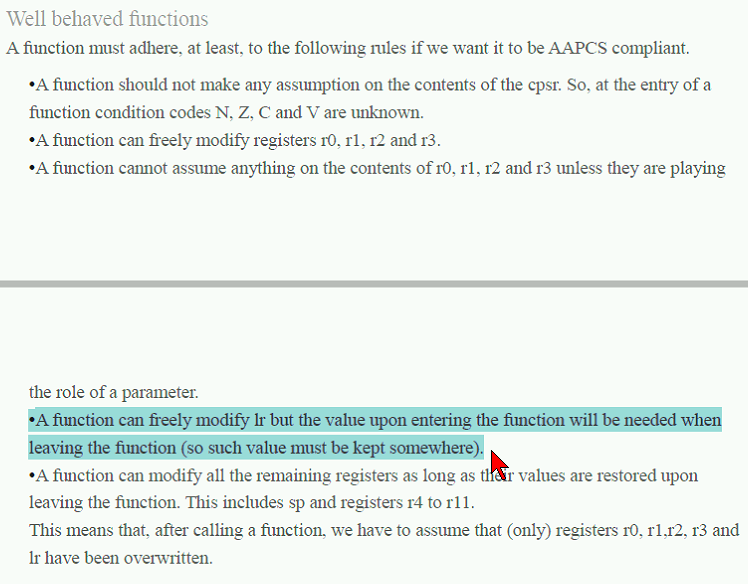
//EQ Flag set

* + Ex Asm) beq r0, r1
    - //EQ Flag set

Pipeline coding

* Def.) Allows the program to run multiple processes at one time

Functions

* Stack frames:
  + R0-r4 contains the 1st,2nd,3rd, and 4th params for the function
* Every function should start with push {lr} and exit with pop {pc}
* Rules: 
* Ex c++) int add\_two\_ints(int x, int y){

int sum;

sum = x + y;

return sum;

}

Creating a label in assembly

* Step 1: Create a label with the same name as the function
* Step2: Include the entry and exit instructions
  + At a minimum, have push {lr} and pop {pc}
  + To preserve r4-r11 values:
    - Push {r4-r11}
    - Pop {r4-r11}
      * Use only if you use these registries
* Ex Asm)
  + add\_two\_ints:

push {lr}

adds r0, r1 //r0 = r0 + r1 (sum = x + y)

pop {pc}

Calling a function

* Ex Asm)
  + mov r0, #5
  + mov r1, #6
  + bl add\_two\_ints