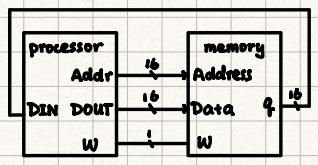
① Replace the counter with register inside the processor--r7 & give it a new name "Program Counter" (PC)

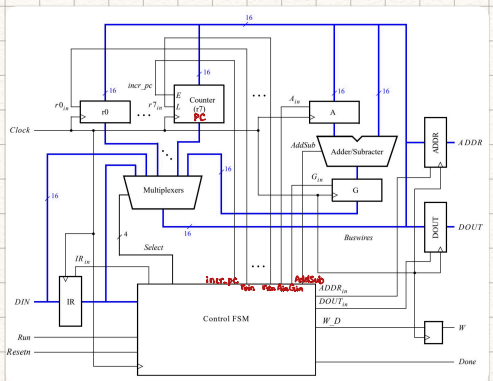
→PC holds the address of the next instruction to be executed.

② Generalize the connection to memory to be readable & writeable used for holding & fetching instruction & to hold & change our data.

The upper level picture of the processor system now looks like this



- to change / write set W=1
- to read set w=0 & q becomes the data at addr



Use these register to access memory:

1) fetch instructions

2). to read & write memory to/from register : two instructions to do these two things:

① LOAD instruction (read from memory & put into a register)

e.g: address 2 = 1025

r4 = 2

ld r1, [r4] r1 ← data in the memory at address given in r4 = 1025

in general, ld rX, [rY] rX ← contents / data of the address by rY.

② STORE instruction: copies the value in a register into a specific memory address.

e.g: if r1 = 68 and r2 = 6

st r1, [r2] would put 68 into memory, address 6

in general, st rX, [rY] puts the value in rX into the address indicated by rY.

③ A new operation in ALU, call "bitwise AND"

and r3, r4 r3 before ····010111
r4 before ····001110
r3 now 000110

in general, and rX, Op2

④ Instruction to control which instruction to execute next

unconditional branch e.g: b #100 pc ← 100

conditional branch beq #123 if G=0 (most recent ALU operation), set PC = 123,
else PC = PC+1