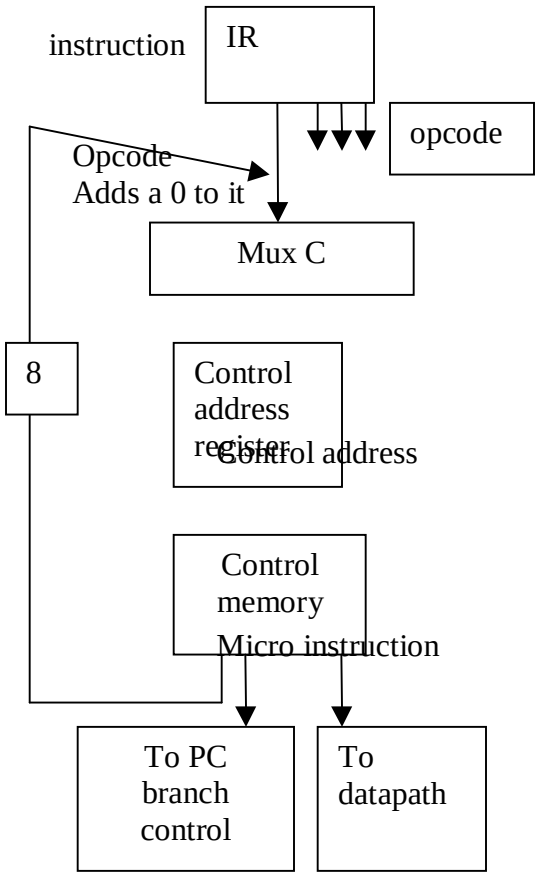
Micro-Programming

Instruction Execution:

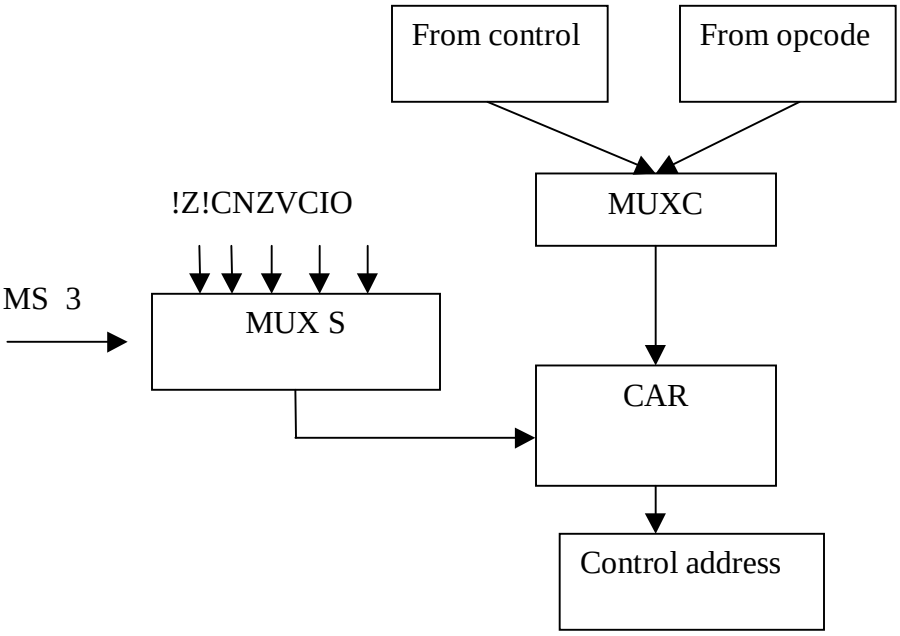
- Address from PC
- Lookup instruction in memory
- Execute
  - If we are doing micro-programming run the micro-program for that opcode.
  - Microinstruction execution:
- Control address from?
  - Look up microinstruction in control memory
- Execute
  - Send out control signals
- Where does the control address come from??
  - In the first cycle of an instruction it's the opcode with a 0 (in the example)
  - In the next cycle(s), it will be specified as part of the previous microop

- To start executing the instruction in the IR, set MC←1
- To execute the next microinstruction set MC←0
  - Control address of next microop specified in the previous microop



Micro-programs & status Bits

- We need to be able to check the status bits as part of a micro-program
  - To implement conditional branch instructions
  - Also to let the status of previous calculations
- We will let he status bits control the CAR (control address register)
  - The CAR will either load or increment in each cycle, depending on the value of its control signal
  - That will allow micro-programs to branch selectively
  - A mux will let it select the branch condition



- If MUXS outputs is 0, then CAR will increment
- If MUXS outputs is 1, it will load
- The MS Signal will be another output of the control ROM
  - Ie part of the micro instruction
- Example instruction: shift & increment
  - First cycle: shift left
  - MB=?
  - FS=11000
  - MD= 0
  - RW=1
  - NA= address of second microinstruction
  - MC=0
  - MS=001 (CAR should load)
  - IL=0 (don't' fetch)
  - PI=1 (increment PC)
  - PL=0 (don't load PC)

DATAPATH CONTROL STUFF

DATAPATH CONTROL STUFF

- Second cycle: increment
    - FS= 000001(?)
    - …..
    -
    - NA=control address of fetch microop
    - MC=0, MS=001, IL=0
    - PI=0, PL=0
- Third cycle: fetch
    - IL =1;
    - MC=1;
    - MS=001;