Clock

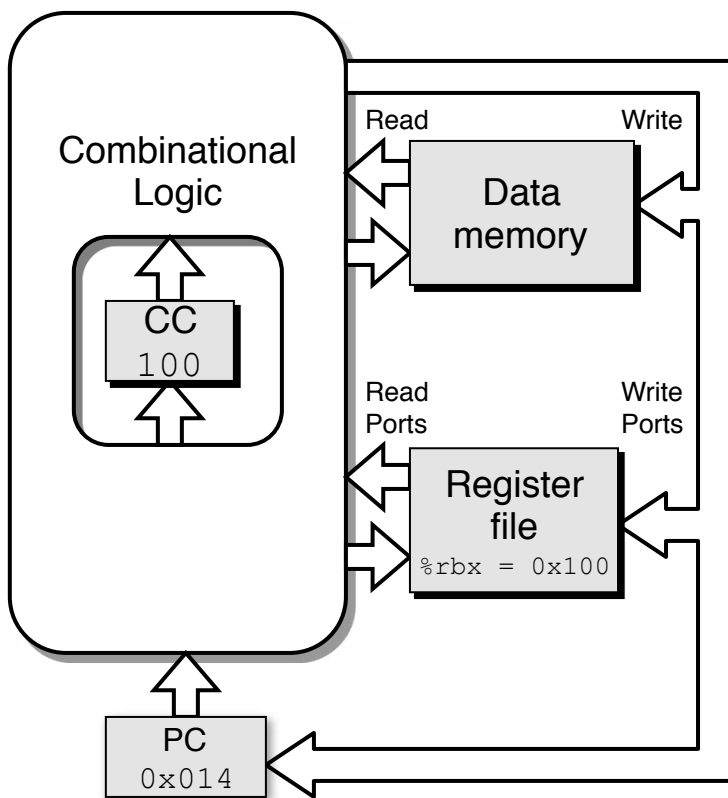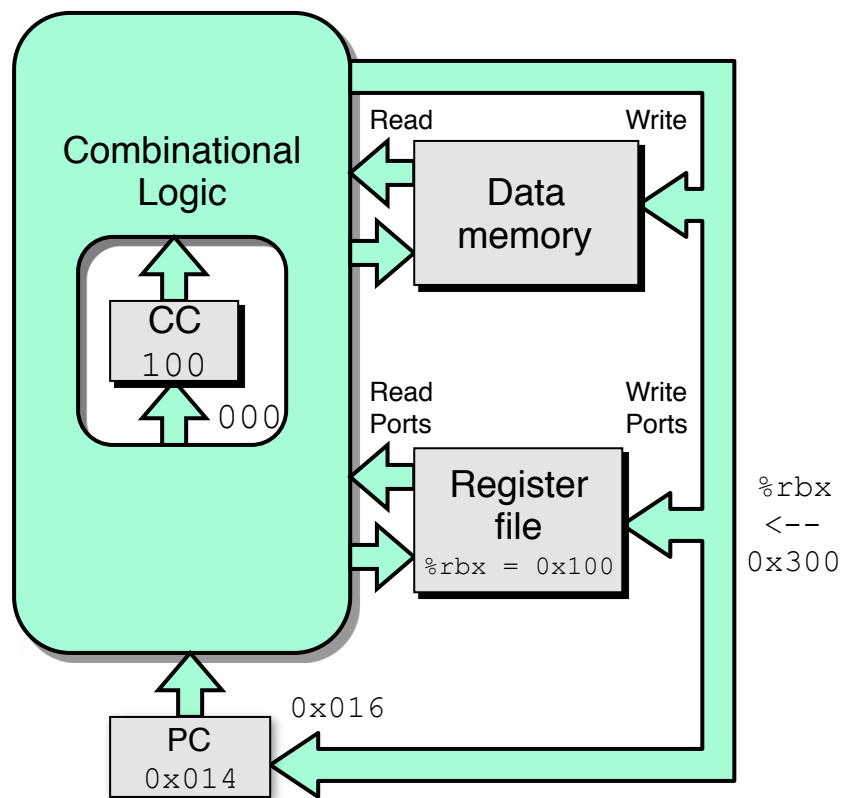| | Cycle 1 | Cycle 2 | Cycle 3 | Cycle 4 |

① ② ③ ④

Cycle 1: `0x000:    irmovq $0x100,%rbx  # %rbx <-- 0x100`

Cycle 2: `0x00a:    irmovq $0x200,%rdx  # %rdx <-- 0x200`

Cycle 3: `0x014:    addq %rdx,%rbx      # %rbx <-- 0x300 CC <-- 000`

Cycle 4: `0x016:    je dest            # Not taken`
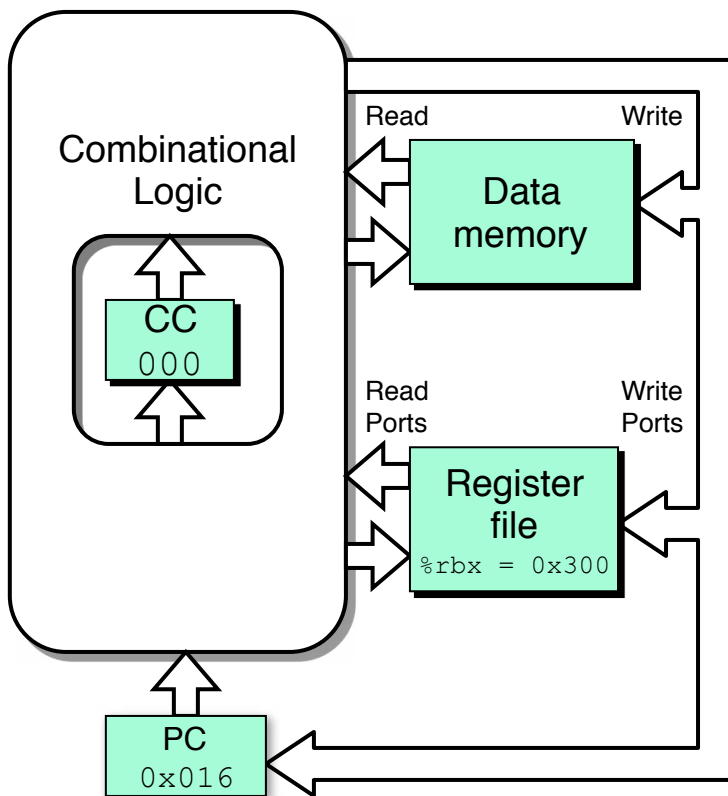
Cycle 5: `0x01f:    rmmovq %rbx,0(%rdx) # M[0x200] <-- 0x300`

① Beginning of cycle 3

Combinational Logic

CC 100

Read   Write

Data memory

Read Ports   Write Ports

Register file
%rbx = 0x100

PC 0x014

② End of cycle 3

Combinational Logic

CC 100
000

Read   Write

Data memory

Read Ports   Write Ports

Register file
%rbx = 0x100

PC 0x014

0x016

%rbx <-- 0x300

③ Beginning of cycle 4

Combinational Logic

CC 000

Read   Write

Data memory

Read Ports   Write Ports

Register file
%rbx = 0x300

PC 0x016

④ End of cycle 4

Combinational Logic

CC 000

Read   Write

Data memory

Read Ports   Write Ports

Register file
%rbx = 0x300

PC 0x016

0x01f