Clock — Cycle 1 — Cycle 2 — Cycle 3 — Cycle 4

① ② ③ ④

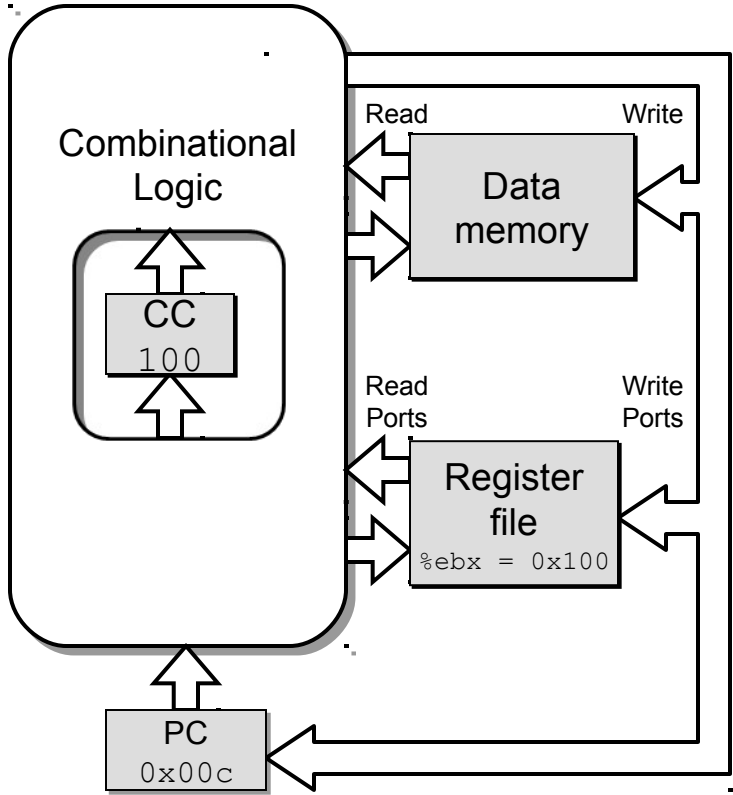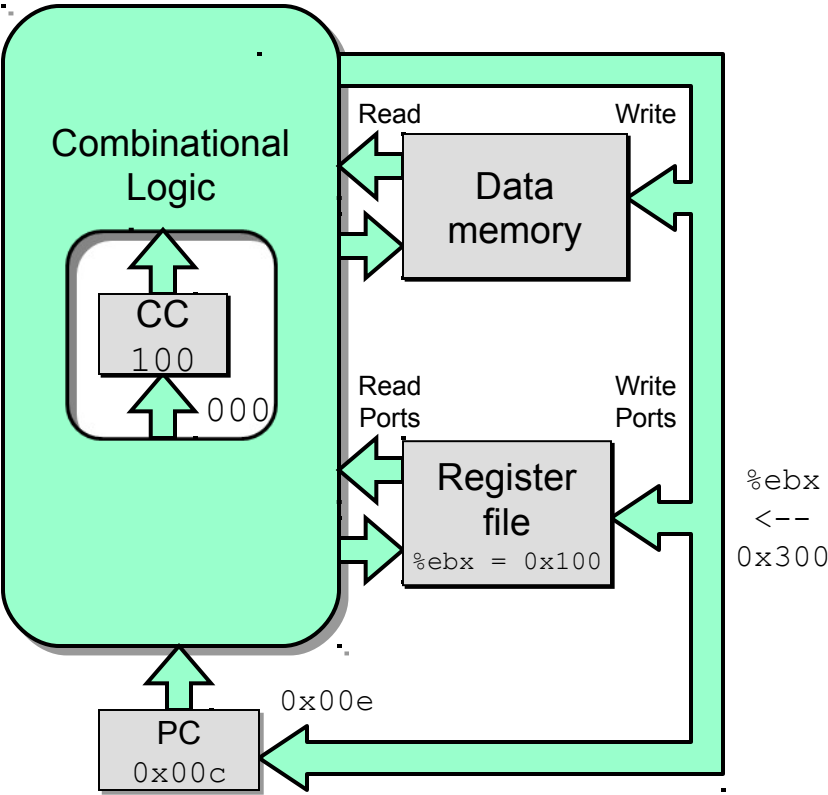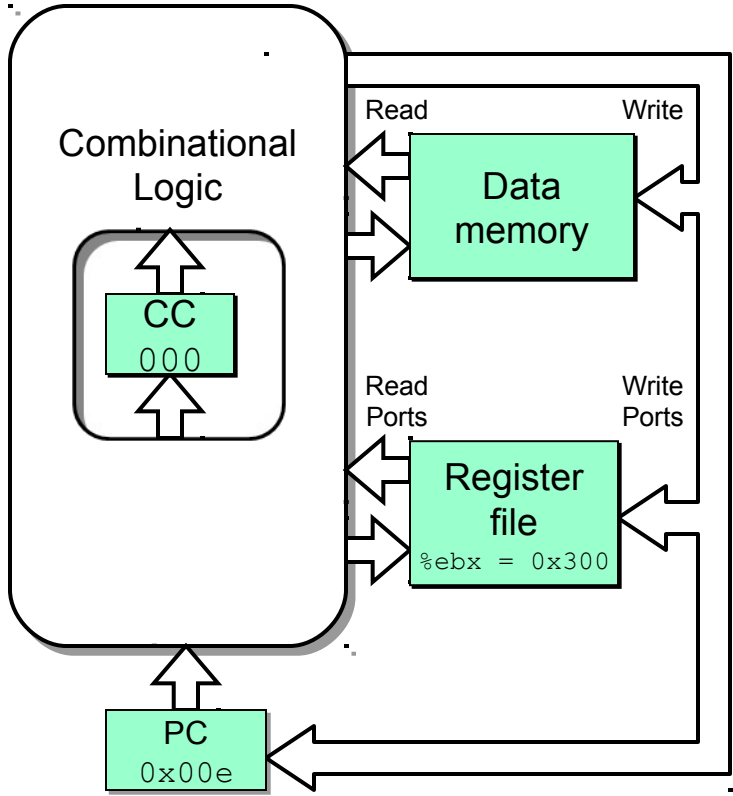|		
Cycle 1:	`0x000:    irmovl $0x100,%ebx   # %ebx <-- 0x100`	
Cycle 2:	`0x006:    irmovl $0x200,%edx   # %edx <-- 0x200`	
Cycle 3:	`0x00c:    addl %edx,%ebx       # %ebx <-- 0x300 CC <-- 000`	
Cycle 4:	`0x00e:    je dest              # Not taken`	
Cycle 5:	`0x013:    rmmovl %ebx,0(%edx) # M[0x200] <-- 0x300`	

① Beginning of cycle 3

Combinational Logic

CC 100

Read   Write
Data memory

Read Ports   Write Ports
Register file
%ebx = 0x100

PC 0x00c

② End of cycle 3

Combinational Logic

CC 100   000

Read   Write
Data memory

Read Ports   Write Ports
Register file
%ebx = 0x100

PC 0x00c

0x00e

%ebx <-- 0x300

③ Beginning of cycle 4

Combinational Logic

CC 000

Read   Write
Data memory

Read Ports   Write Ports
Register file
%ebx = 0x300

PC 0x00e

④ End of cycle 4

Combinational Logic

CC 000

Read   Write
Data memory

Read Ports   Write Ports
Register file
%ebx = 0x300

PC 0x00e

0x013