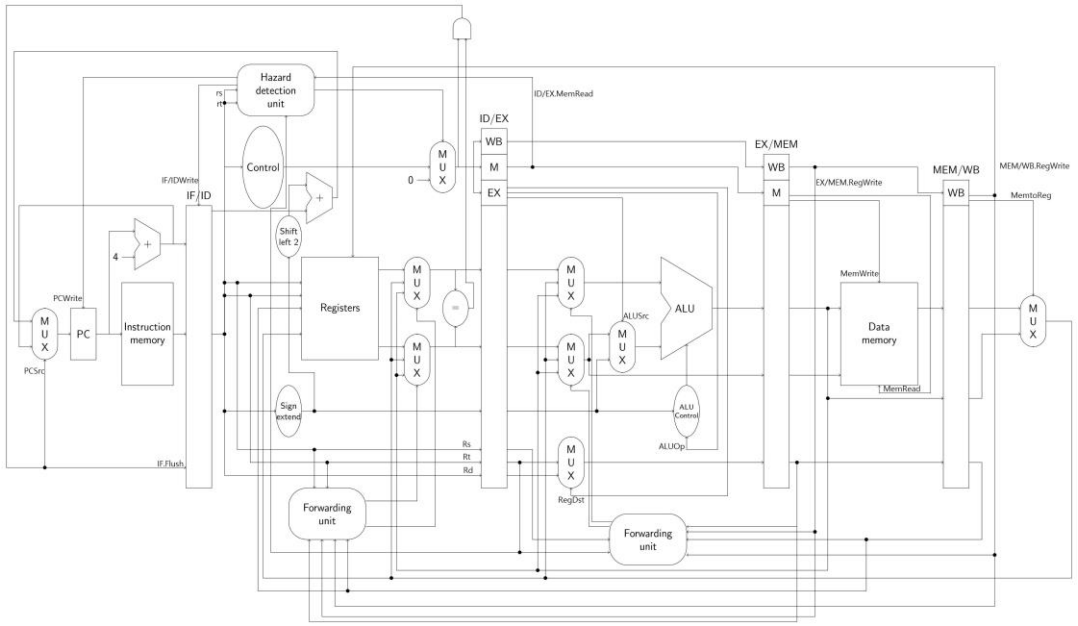
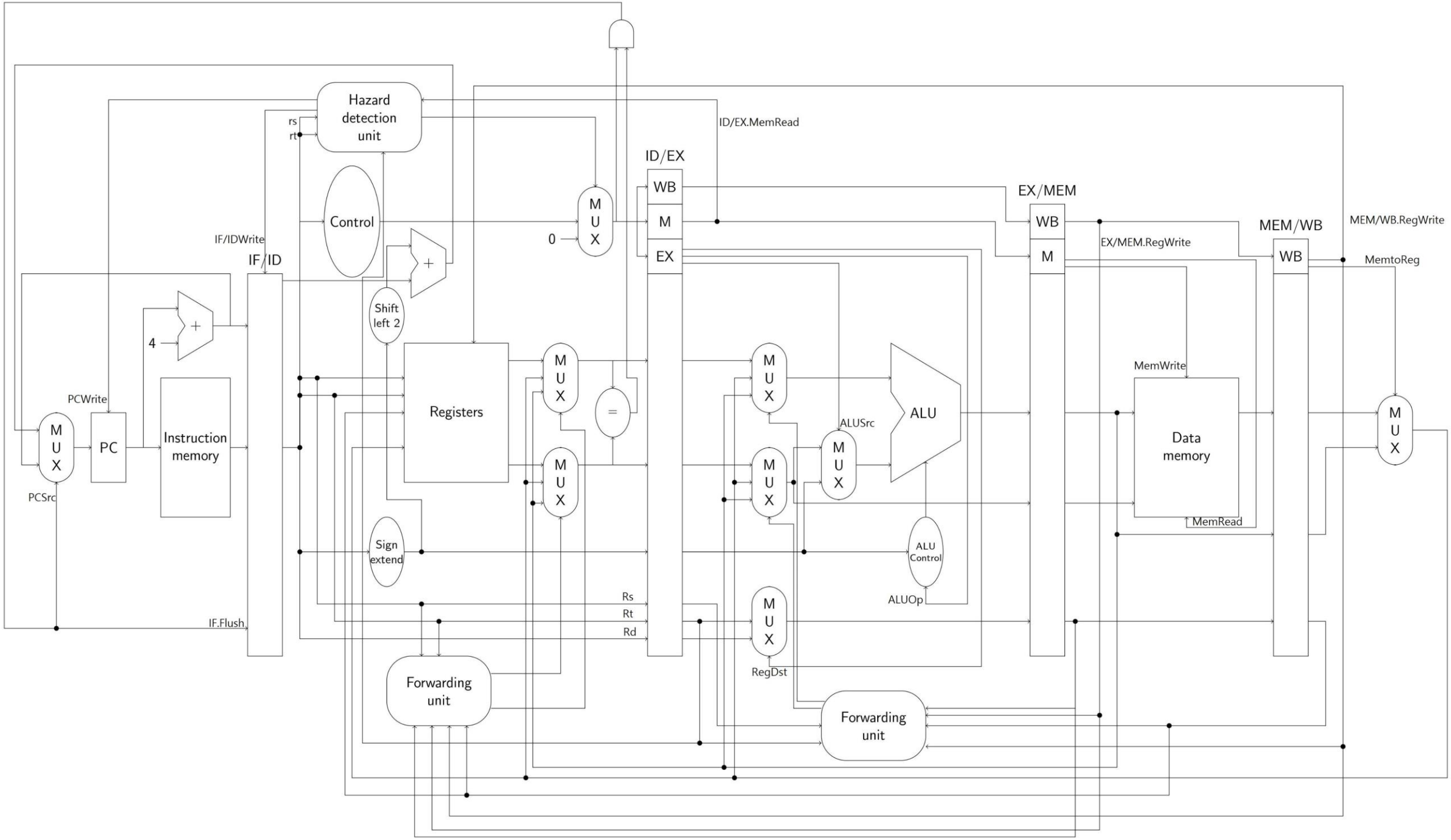
컴퓨터 구조 HW5

20181653 이강희

1번

- ID 단계의 beq명령이 EX 단계의 명령에 의존성이 있거나 MEM 단계의 lw명령에 의존성이 있으면 한사이클 지연시키도록 Hazard Detection Unit이 하나 더 필요하지만(또는 기존 유닛을 개선), 회로가 너무 복잡해져서 생략했습니다.




Hazard detection unit
rs
rt
ID/EX.MemRead
ID/EX
WB
M
EX
EX/MEM
WB
M
MEM/WB
WB
MEM/WB.RegWrite
EX/MEM.RegWrite
MemtoReg
Control
0
MUX
IF/IDWrite
IF/ID
+
4
Shift left 2
PCWrite
PC
Instruction memory
Registers
=
MUX
ALUSrc
ALU
MemWrite
Data memory
MemRead
Sign extend
ALU Control
ALUOp
Rs
Rt
Rd
RegDst
IF.Flush
Forwarding unit
Forwarding unit
PCSrc

2번

IF/ID – 64bit

ID/EX – 114bit

EX/MEM – 73bit

MEM/WB – 71bit

IF/ID

Instruction – 32bit
PC+4 – 32bit

Total: 64bit

ID/EX

RegDst, ALUSrc, MemtoReg, RegWrite, MemRead, MemWrite, ALUOP1, ALUOP2 – 8bit (branch는 ID단계에서 사용됨)
Read data1 – 32bit
Read data2 – 32bit
Sign-extended immed – 32bit
Rs, Rt, Rd – 15bit

Total: 119bit
하지만 Sign-extended immed 에서 Rd를 얻을 수 있으므로 114bit로 개선할수 있음.
Total: 114bit

EX/MEM

MemtoReg, RegWrite, MemRead, MemWrite – 4bit
ALU 결과 – 32bit
Read data2 – 32bit
Rd (Rt, Rd 중 선택된 것) – 5bit

Total: 73bit

MEM/WB

MemtoReg, RegWrite – 2bit
ALU 결과 – 32bit
Memory read data – 32bit
Rd – 5bit

Total: 71bit

3번

lw $8, 0($gp)

100011 11100 01000 0000000000000000

add $9, $8, $0

000000 01000 00000 01001 00000 100000

beq $8, $9, L

000100 01000 01001 0000000000000001

lw $10, 4($gp)

100011 11100 01010 0000000000000100

slt $11, $9, $0

000000 01001 00000 01011 00000 101010

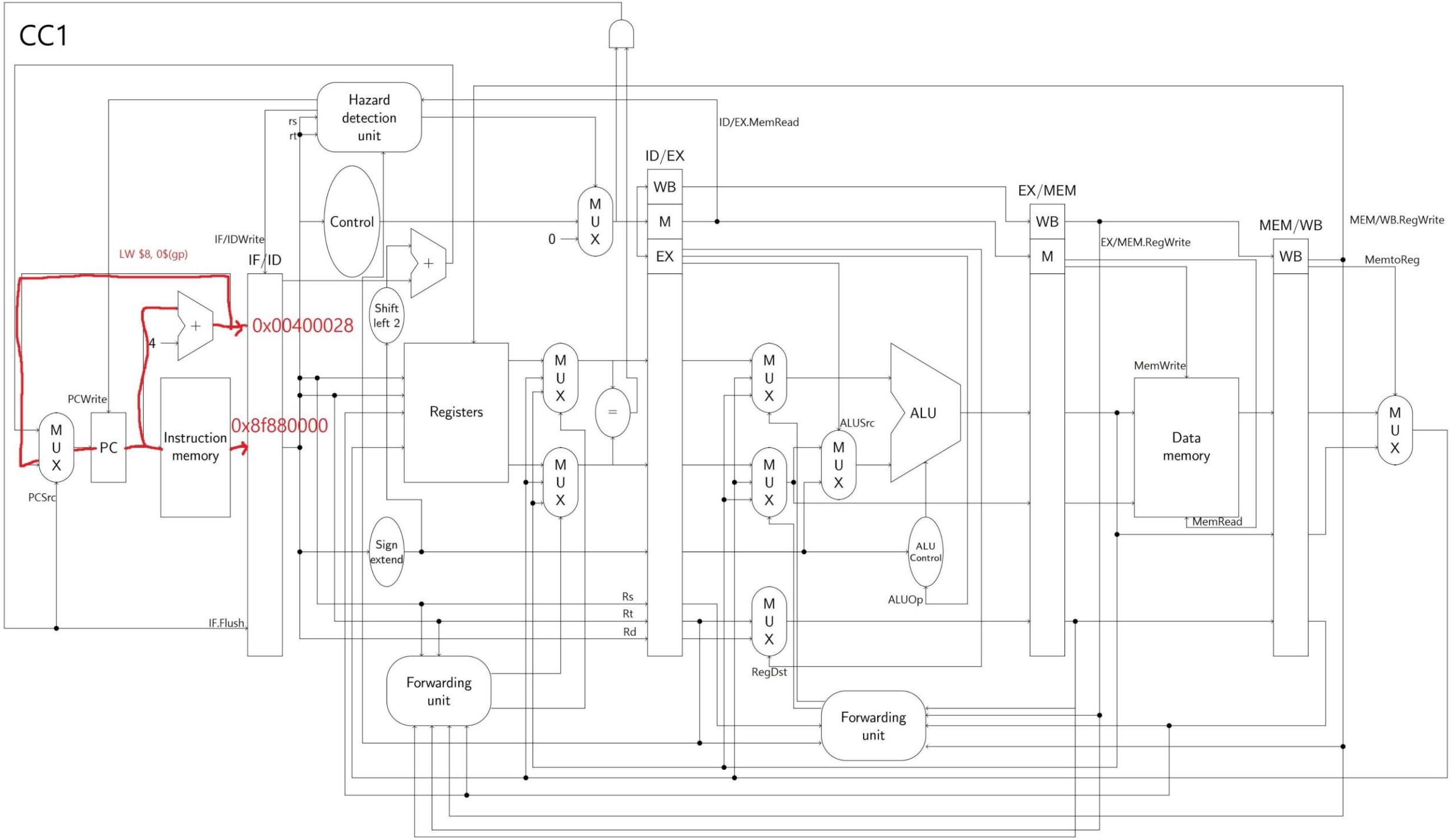
beq immed field에 관하여

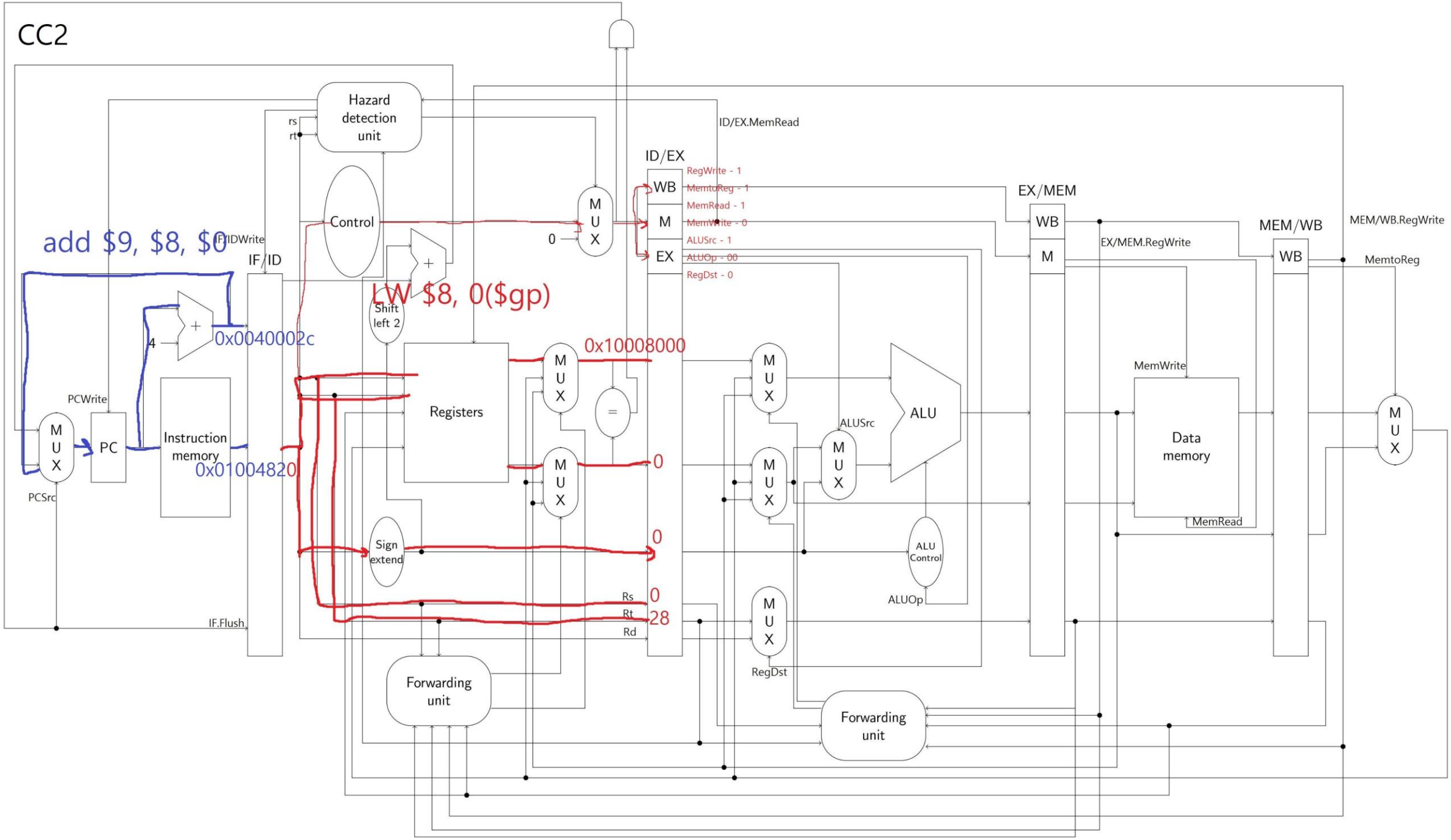
회로상에서는 PC+4에 offset을 더하기 때문에, QtSpim에서 어셈블된 것과 1만큼 차이납니다.

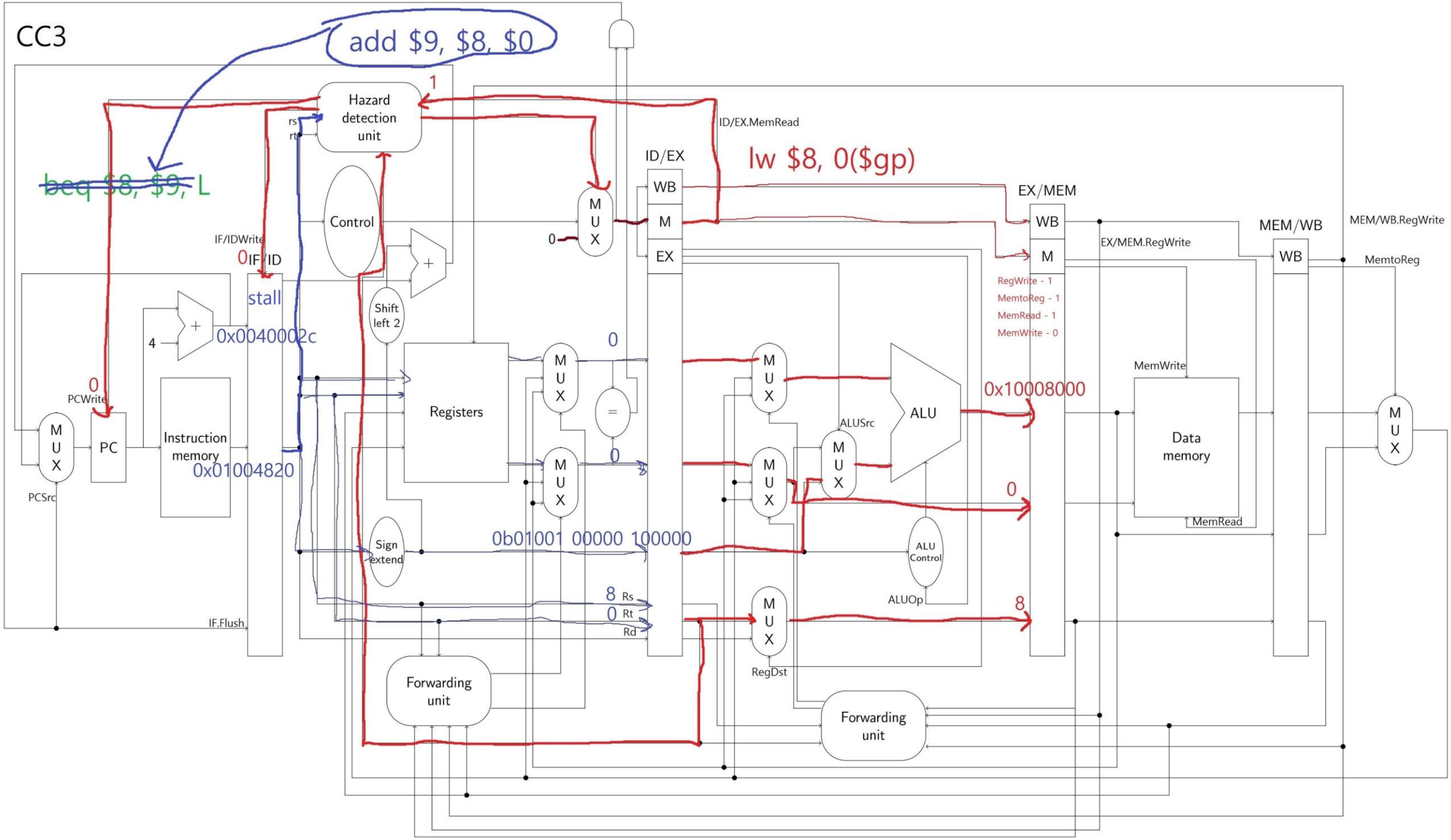
beq $8, $9, L

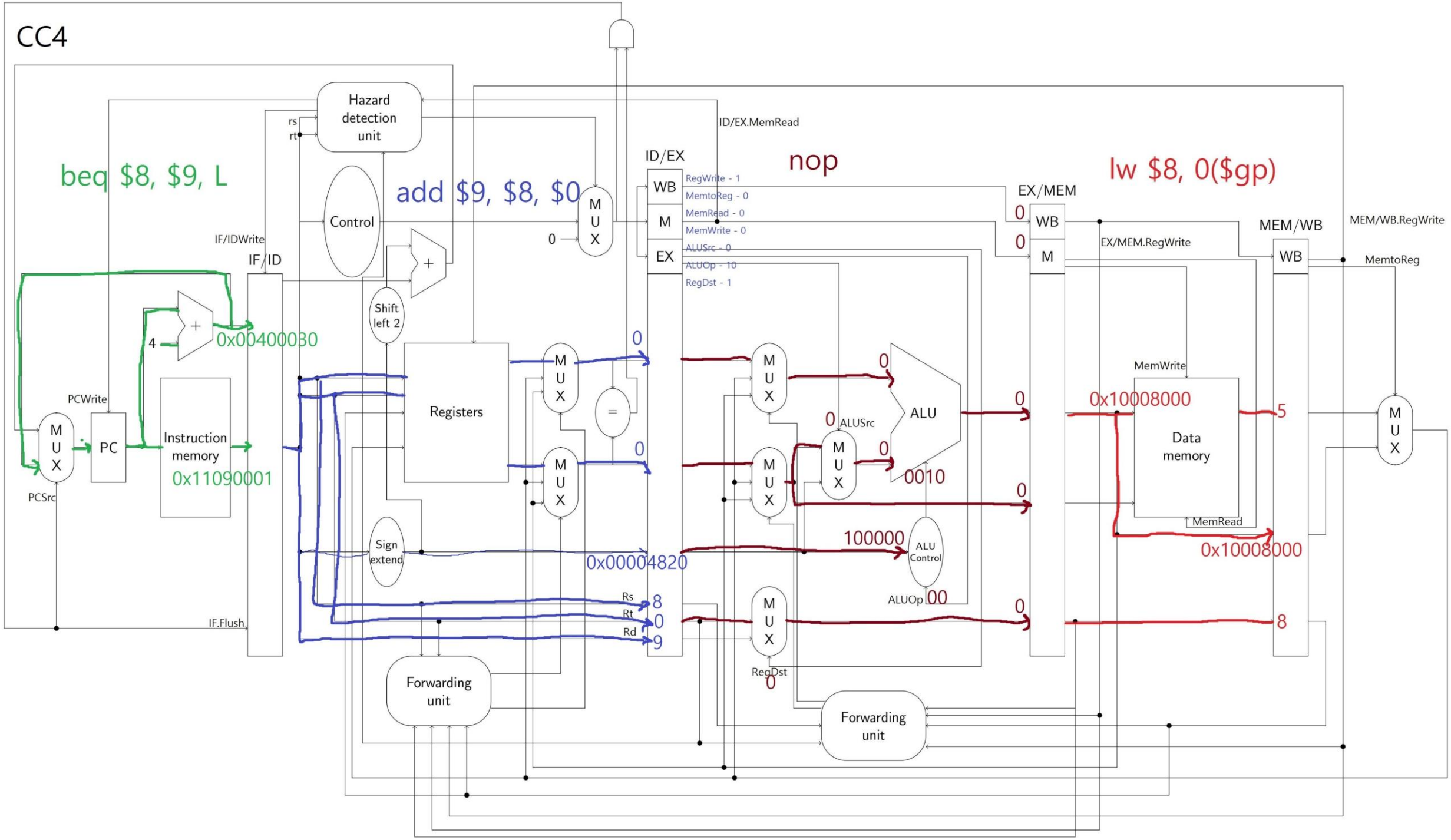
000100 01000 01001 0000000000000001

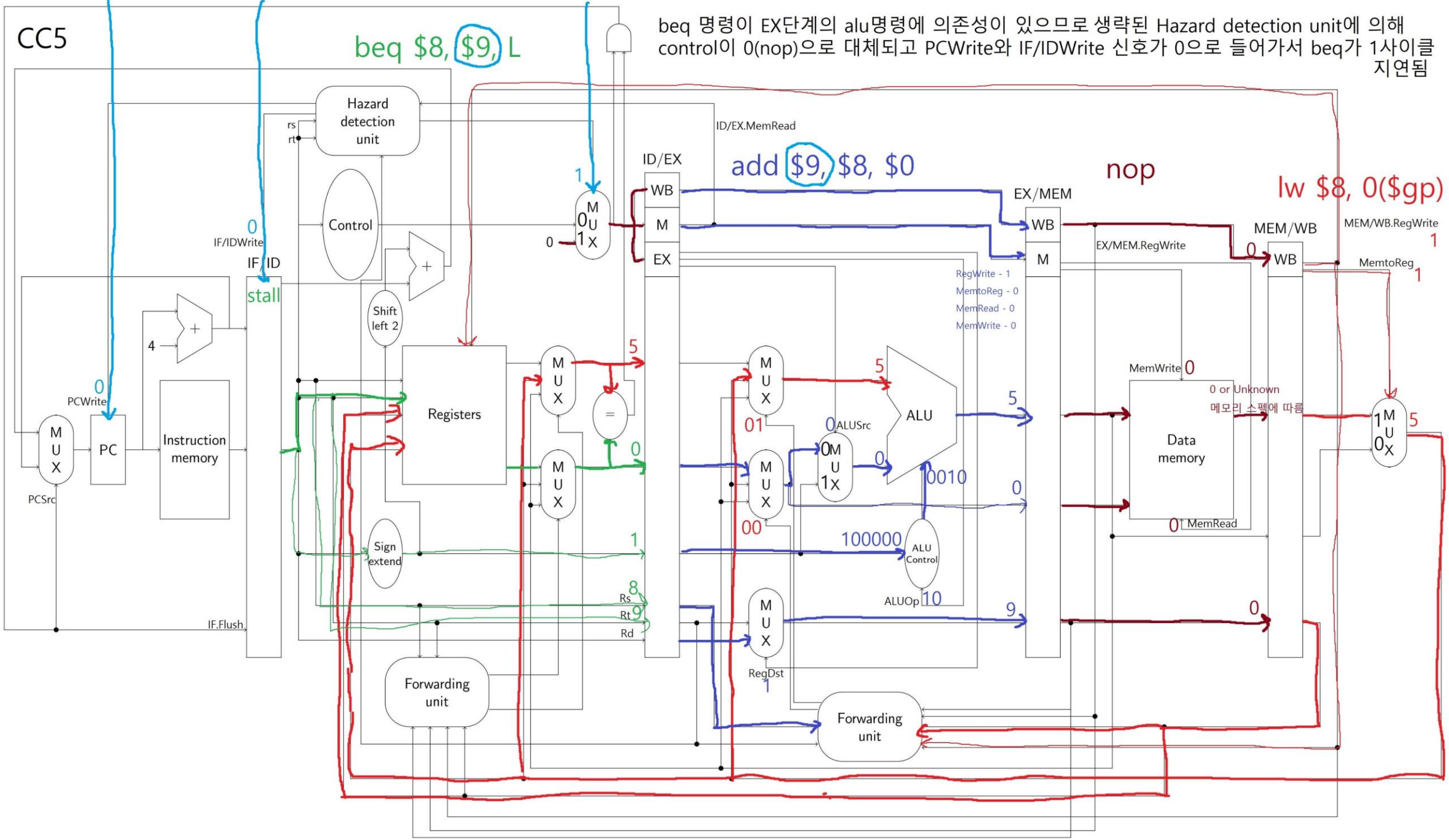
```
[0040002c] 11090002  beq $8, $9, 8 [L-0x0040002c]; 8: beq $8, $9, L
[00400030] 8f8a0004  lw $10, 4($28)              ; 9: lw $10, 4($gp)
[00400034] 0120582a  slt $11, $9, $0             ; 11: slt $11, $9, $0
```


CC1
Hazard detection unit
rs
rt
ID/EX.MemRead
ID/EX
WB
M
EX
EX/MEM
WB
M
MEM/WB
WB
MEM/WB.RegWrite
MemtoReg
EX/MEM.RegWrite
Control
IF/IDWrite
IF/ID
LW $8, 0$(gp)
0
MUX
+
4
0x00400028
Shift left 2
PCWrite
PC
Instruction memory
0x8f880000
PCSrc
Registers
=
ALUSrc
ALU
MemWrite
Data memory
MemRead
ALU Control
ALUOp
Sign extend
Rs
Rt
Rd
IF.Flush
RegDst
Forwarding unit
Forwarding unit


CC2
Hazard detection unit
rs
rt
ID/EX.MemRead
ID/EX
RegWrite - 1
MemtoReg - 1
MemRead - 1
MemWrite - 0
ALUSrc - 1
ALUOp - 00
RegDst - 0
WB
M
EX
EX/MEM
MEM/WB
MEM/WB.RegWrite
EX/MEM.RegWrite
MemtoReg
add $9, $8, $0
LW $8, 0($gp)
Control
PCWrite
IF/IDWrite
IF/ID
Shift left 2
4
0x0040002c
0x10008000
0x01004820
PC
PCSrc
Instruction memory
Registers
Sign extend
Rs
Rt
Rd
0
0
0
28
IF.Flush
ALUSrc
ALU
ALU Control
ALUOp
MemWrite
MemRead
Data memory
RegDst
Forwarding unit
Forwarding unit
MUX
=


CC3
add $9, $8, $0
beq $8, $9, L
lw $8, 0($gp)
Hazard detection unit
rs
rt
1
ID/EX.MemRead
ID/EX
EX/MEM
MEM/WB
MEM/WB.RegWrite
MemtoReg
WB
M
EX
Control
IF/IDWrite
0
IF/ID
stall
Shift left 2
4
0x0040002c
PCWrite
PC
Instruction memory
0x01004820
PCSrc
Registers
0
0
0b01001 00000 100000
Sign extend
8 Rs
0 Rt
Rd
IF.Flush
Forwarding unit
Forwarding unit
RegWrite - 1
MemtoReg - 1
MemRead - 1
MemWrite - 0
EX/MEM.RegWrite
MemWrite
MemRead
ALUSrc
ALU
ALU Control
ALUOp
RegDst
0x10008000
0
8
Data memory
MUX


CC4
beq $8, $9, L
add $9, $8, $0
nop
lw $8, 0($gp)
Hazard detection unit
rs
rt
ID/EX.MemRead
Control
IF/IDWrite
IF/ID
ID/EX
WB
M
EX
RegWrite - 1
MemtoReg - 0
MemRead - 0
MemWrite - 0
ALUSrc - 0
ALUOp - 10
RegDst - 1
EX/MEM
MEM/WB
MEM/WB.RegWrite
EX/MEM.RegWrite
MemtoReg
Shift left 2
4
0x00400030
PCWrite
PC
Instruction memory
0x11090001
PCSrc
Registers
Sign extend
0x00004820
Rs
Rt
Rd
8
0
9
IF.Flush
Forwarding unit
Forwarding unit
ALU
ALUSrc
0010
100000
ALU Control
ALUOp
00
RegDst
MemWrite
MemRead
Data memory
0x10008000
0x10008000
5
8
MUX
=


CC5
beq $8, $9, L
beq 명령이 EX단계의 alu명령에 의존성이 있으므로 생략된 Hazard detection unit에 의해 control이 0(nop)으로 대체되고 PCWrite와 IF/IDWrite 신호가 0으로 들어가서 beq가 1사이클 지연됨
Hazard detection unit
rs
rt
ID/EX.MemRead
Control
IF/IDWrite
IF/ID
stall
ID/EX
add $9, $8, $0
nop
lw $8, 0($gp)
EX/MEM
MEM/WB
MEM/WB.RegWrite
MemtoReg
EX/MEM.RegWrite
WB
M
EX
RegWrite - 1
MemtoReg - 0
MemRead - 0
MemWrite - 0
Shift left 2
4
PCWrite
PC
Instruction memory
PCSrc
Registers
Sign extend
Rs
Rt
Rd
IF.Flush
Forwarding unit
ALUSrc
ALU
0010
100000
ALU Control
ALUOp
10
RegDst
MemWrite
MemRead
0 or Unknown
메모리 스펙에 따름
Data memory
MUX

IF/ID (beq $8, $9, L)

PC+4 – 0x00400030

Instruction – 0x11090001

ID/EX (add $9, $8, $0)

RegWirte – 1, MemtoReg – 0, MemRead – 0, MemWrite – 0, ALUSrc – 0, ALUOp – 10, RegDst – 1

Read data 1 – 0

Read data 2 – 0

Sign-extend immed – 0x00004820 (Rd – 9 포함)

Rs – 8

Rt - 0

EX/MEM (nop)

MemtoReg, RegWrite, MemRead, MemWrite – 0

ALU 결과 – 0

Read data 2 – 0

Rd - 0

MEM/WB (lw $8, 0($gp))

MemtoReg – 1, RegWrite – 1

ALU 결과 – 0x10008000

Memory read data – 5

Rd - 8