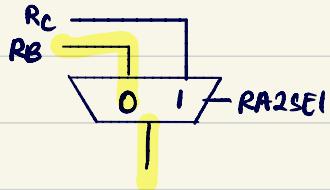


Problem: RA2SEL is always 0

↳ ST is affected



label = 0X00AC

• = 0X000

CMOVE(3, R5)

STC R5, label, R31 → ST RC Ra | C
011001 00101 1111 | 0000 0000 1010 1100
RB = RO

Fix: CMOVE(3, R0)

STC R[~]X, label, R31)

ignored

if : label = 0x AABBB
not 0

ST	RC	Ra			C
011001	xxxxxx	11111		1010 1010 1011 1011	
					$\underbrace{\qquad}_{RB = R_{21}}$

Fix: CMOVE(3, R21)

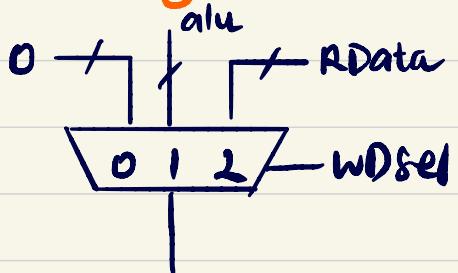
STC RX, label, R31)

ignored

Problem: WDSEL 0 mux is stuck at 0

↳ we cannot store return address automatically

which code can "check" this error?



(a) $\cdot = 0X000$

$BEQ(R31, \text{label}, LP)$

$\text{label: ADDC(RI, 4, RI)}$

$JMP(LP)$

(b) $\cdot = 0X000$

$BEQ(R31, \cdot + 4, LP)$

$JMP(LP)$

$ADDC(RI, 4, RI)$

run for 1 cycle.

Both (a) & (b) \rightarrow after BEQ , content of LP is 0 instead of 4

Problem : WDSSEL 0 mux is stuck at 0

↳ we cannot store return address automatically

· = 0X000

BEQ(R31, label, LP)

label : ADDC(R1, 4, R1)

JMP(LP)

any fix ?

· = 0X000

CMOVE(· + 8, LP)

BEQ(R31, label, R31)

label : ADDC(R1, 4, R1)

JMP(LP)

Problem : BSEL 0 mux is always 0

all OPC operations are affected. No fix.

Which code can check for this error?

(a) . = 0X00

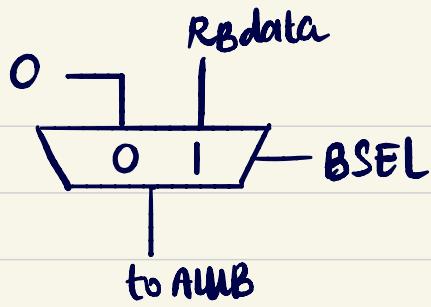
CMOVE (0XF000, R0) → R0 contains content of R30 or R31 depends
MUL (R0, 4, R0) → R0 doesn't contain 16 on RA2SEL

(b) . = 0X00

ADD (R31, R3, R3)

JMP (R3)

} no change



Can we do this in 1 cycle?

ADDMUL (R0, R1, R2) : $\text{Reg}[R2] \leftarrow \text{Reg}[R0] * \text{Reg}[R1]$
 $\text{Reg}[R0] \leftarrow \text{Reg}[R0] + \text{Reg}[R2]$

* No, we only have 1 ALU.

ADDST (R0, label, R2) : $\text{Reg}[R0] \leftarrow \text{Reg}[R0] + \text{Reg}[R2]$
 $\text{Mem}[\text{label}] \leftarrow \text{Reg}[R0]$

* No, output of ALU cannot be directly stored to memory as data

can we do this in 1 cycle?

$$WASEL = 0$$

$$PCSEL = 0$$

$$ASEL = 0$$

$$BSEL = 0$$

$$RA \angle SEL = 0$$

$$WR = 0$$

$$WERF = 1$$

$$WDSEL = 2$$

AUJFN = "+"

← yes. give all values

What is the Reg transfer language for this?

Mem [PC+4+4*SXT(C)] ← Reg[RC]

Reg[RC] ← PC + 4

PC ← Reg[RA]

