Computer Architecture Lab

Lab11 – Week #11

MPLAB Media Processing Laboratory

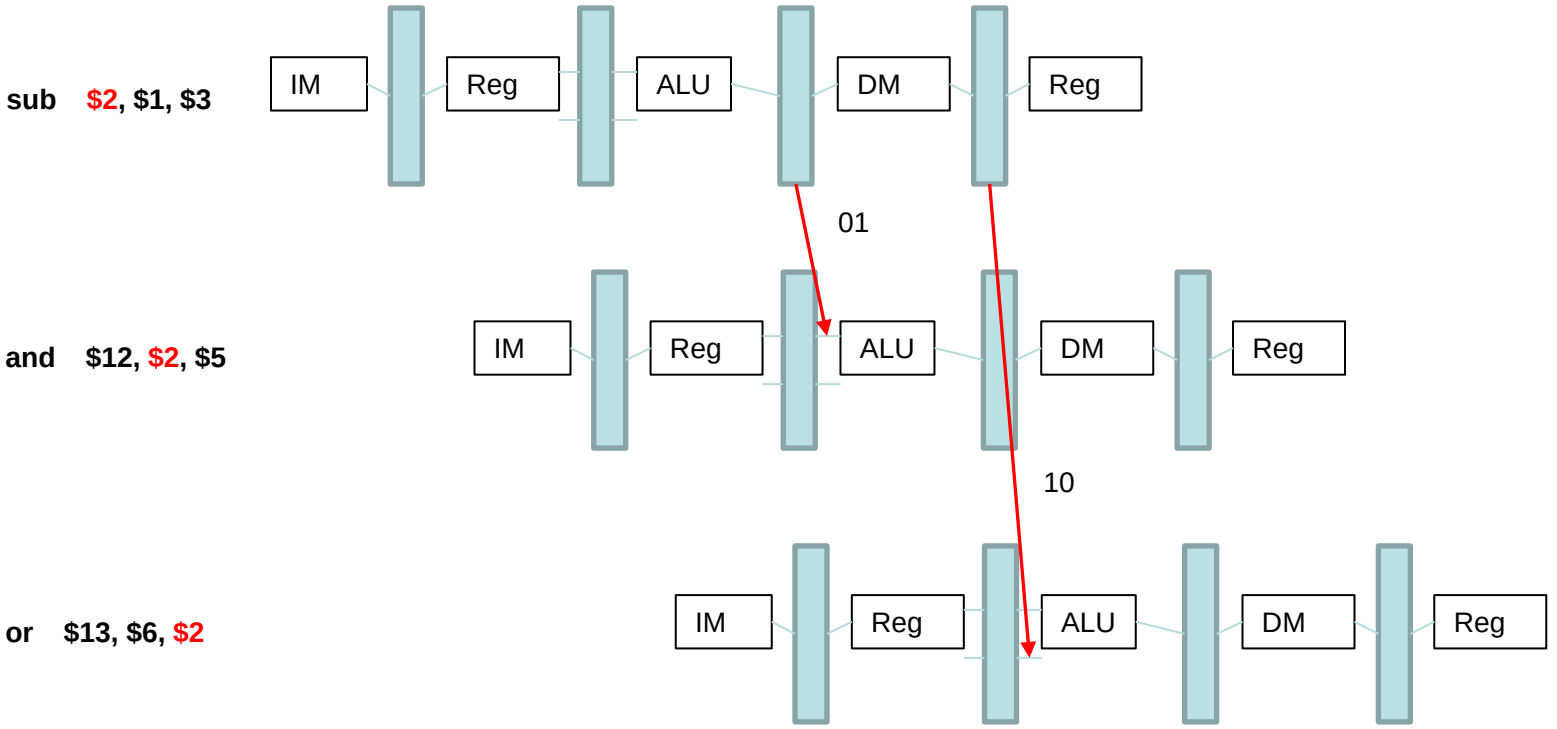
컴퓨터 공학과 이성원 교수님 연구실

ALU forwarding

Table 2 – Signals for M_TEXT_FWD.txt

Signal Names	Descriptions
FWD_ALU_Ai	**00:** From Register file of ID stage to EX stage **01:** From ALU output of MM stage **10:** From Writeback data of WB stage
FWD_ALU_Bi	**00:** From Register file of ID stage to EX stage **01:** From ALU output of MM stage **10:** From Writeback data of WB stage
* For both signals, 11 is not allowed.	

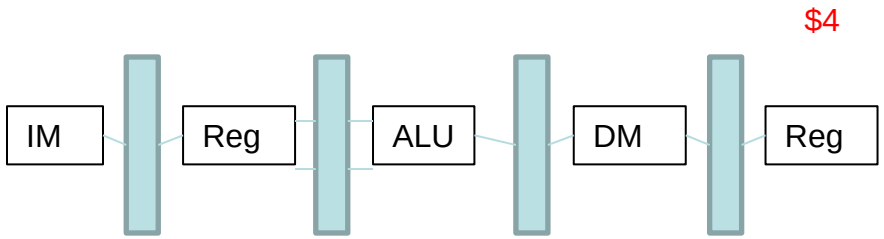
ALU forwarding figure

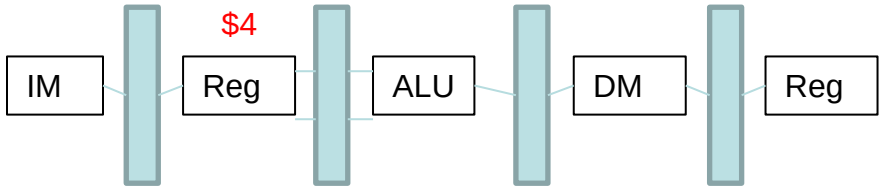


Hazard example - ALU forwarding (01)

insertion_sort.asm

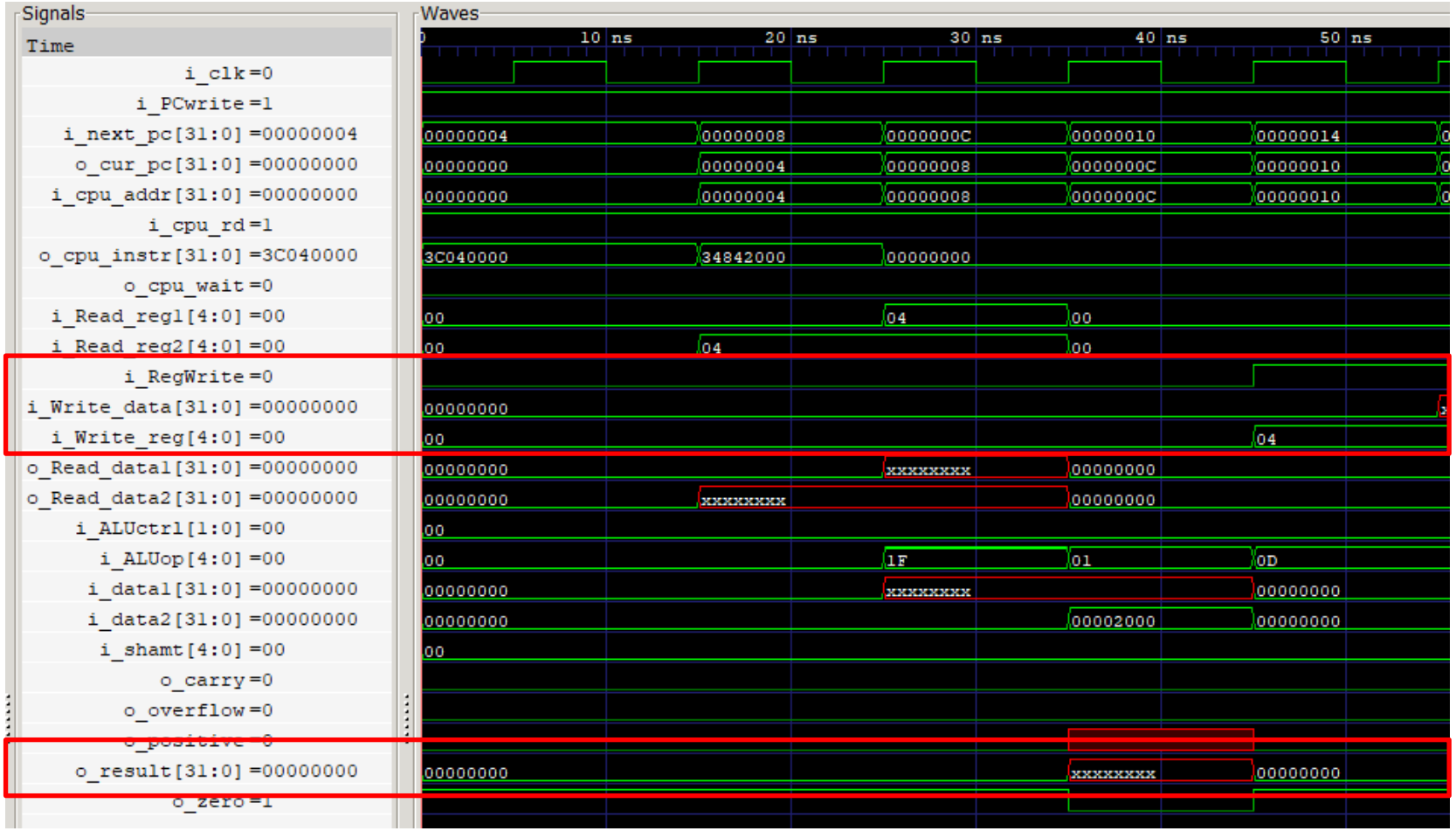
```
1  .text
2  main: lui   $4, 0x0000
3        ori   $4, $4, 0x2000
```





Hazard example - ALU forwarding (01)

lui $4 0x0000	Fetch	Decode	Exe	Mem	Wb
ori $5 $0 0x2000		Fetch	Decode	Exe	Mem

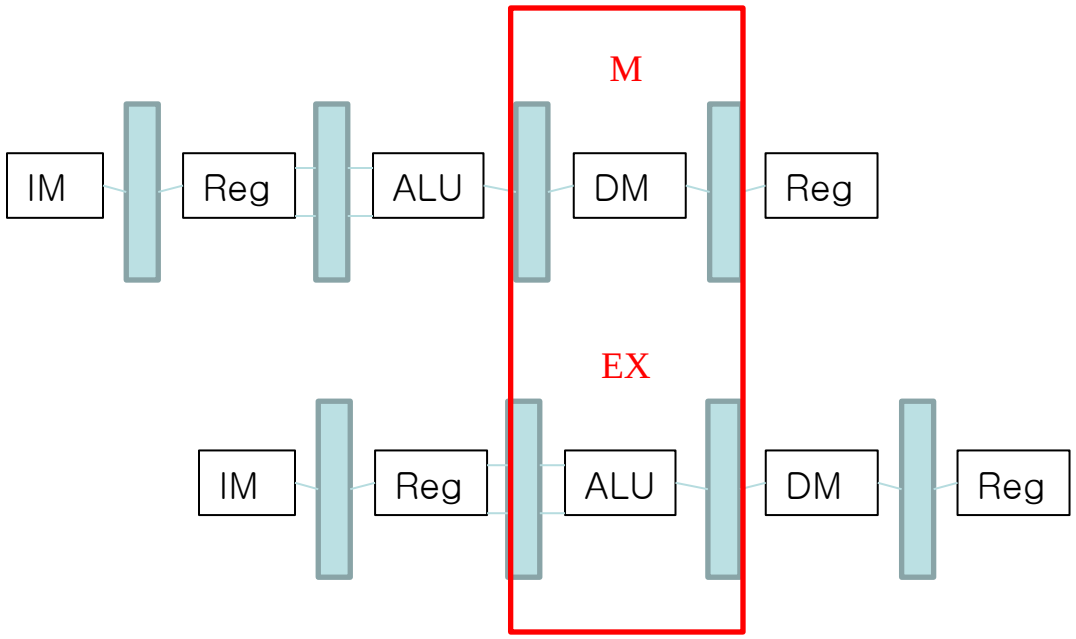


MPLAB
Media Processing Laboratory

Hazard example - ALU forwarding (01)

insertion_sort.asm

```
1  .text
2  main: lui   $4, 0x0000
3        ori   $4, $4, 0x2000
```



Hazard example - ALU forwarding (01)

M_TEXT_SEG.txt

lui
ori

```
1   00111100000001000000000000000000
2   00110100100001000010000000000000
3   00000000000000000000000000000000
4   00000000000000000000000000000000
5   00000000000000000000000000000000
6   00000000000000000000000000000000
7   00110100000001010000000001100100
8   00000000000000000000000000000000
9   00000000000000000000000000000000
10  00000000000000000000000000000000
11  00000000000000000000000000000000
12  00100000000010000000000000000001
13  00000000000000000000000000000000
14  00000000000000000000000000000000
```

M_TEXT_FWD.txt

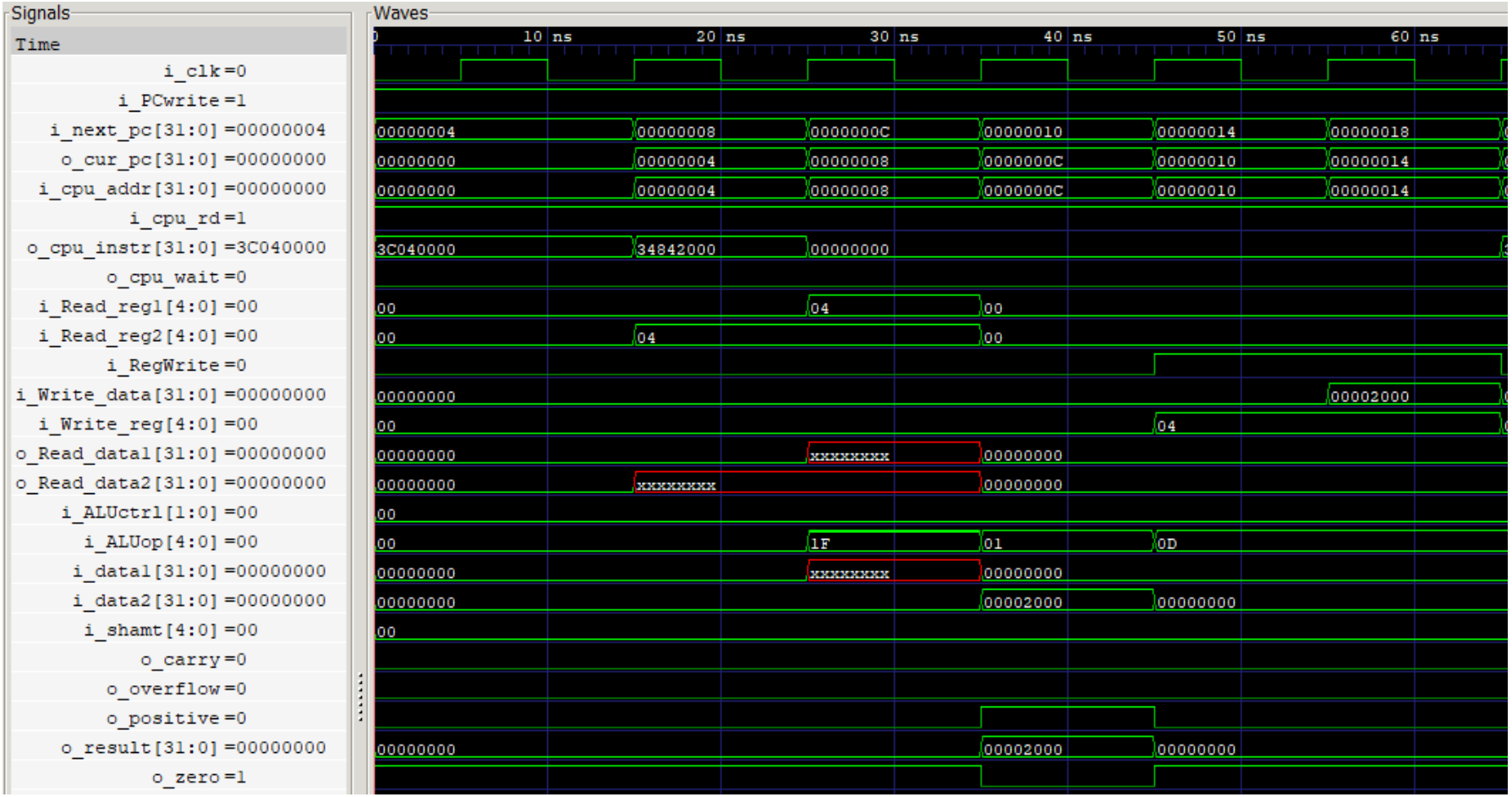
lui
ori

```
1   01_00  //  0x000
2   00_00  //  0x004
3   00_00  //  0x008
4   00_00  //  0x00C
5   00_00  //  0x010
6   00_00  //  0x014
7   00_00  //  0x018
8   00_00  //  0x01C
9   00_00  //  0x020
10  00_00  //  0x024
11  00_00  //  0x028
12  00_00  //  0x02C
13  00_00  //  0x030
14  00_00  //  0x034
```

Hazard example - ALU forwarding (01)

lui $4 0x0000 Fetch Decode Exe Mem Wb

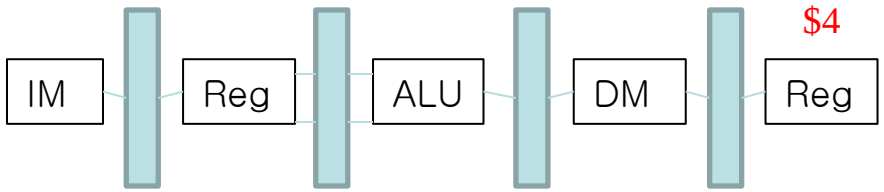
ori $5 $0 0x2000 Fetch Decode Exe Mem Wb

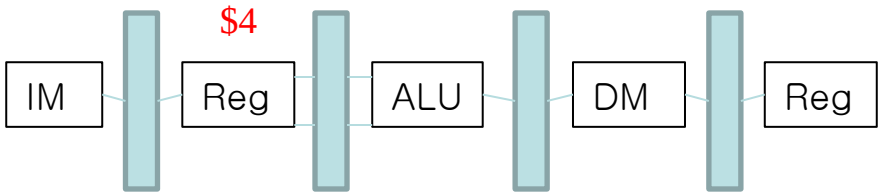


Hazard example - ALU forwarding (10)

insertion_sort.asm

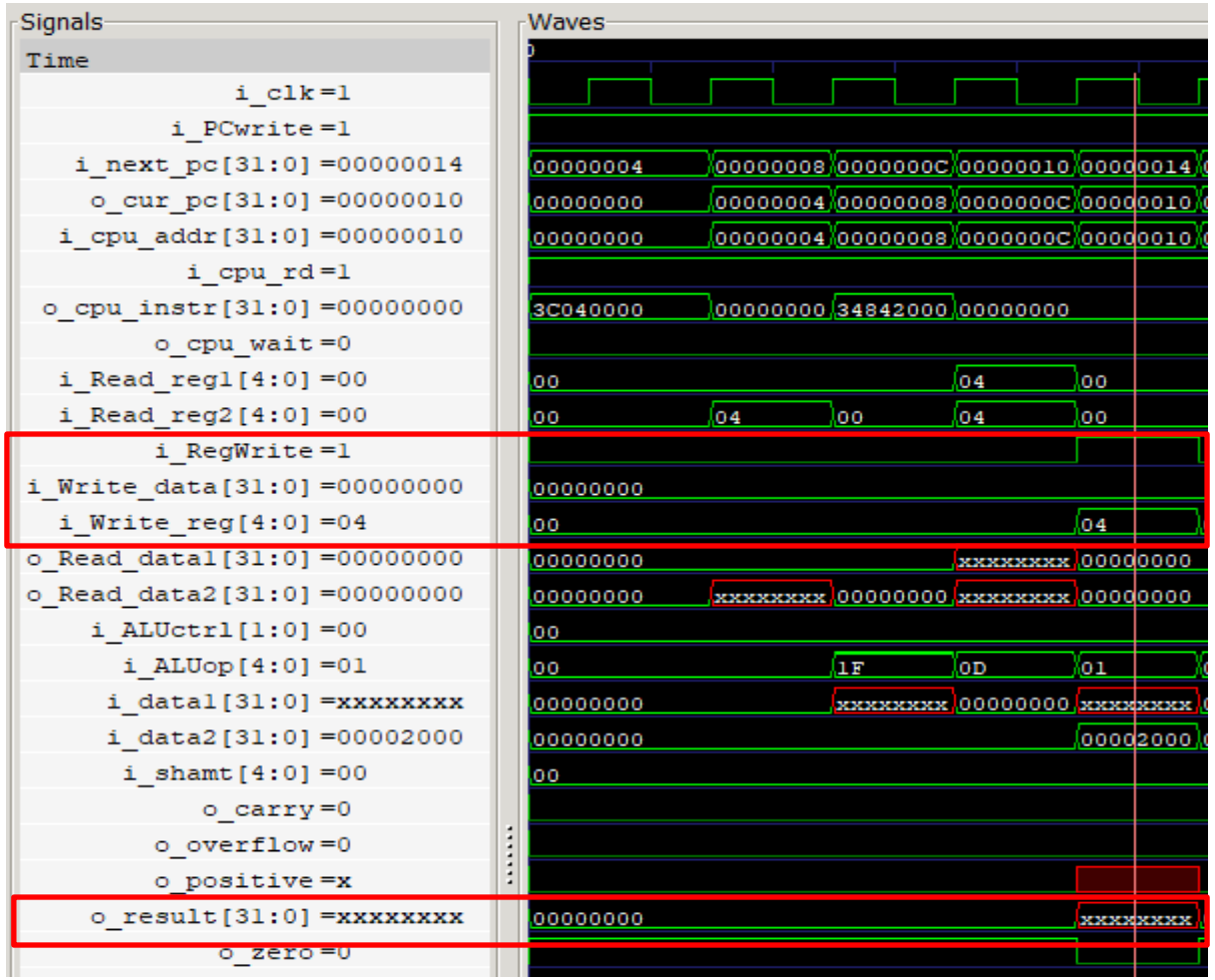
```
1  .text
2  main: lui   $4, 0x0000
3        nop
4        ori   $4, $4, 0x2000
```





Hazard example - ALU forwarding (10)

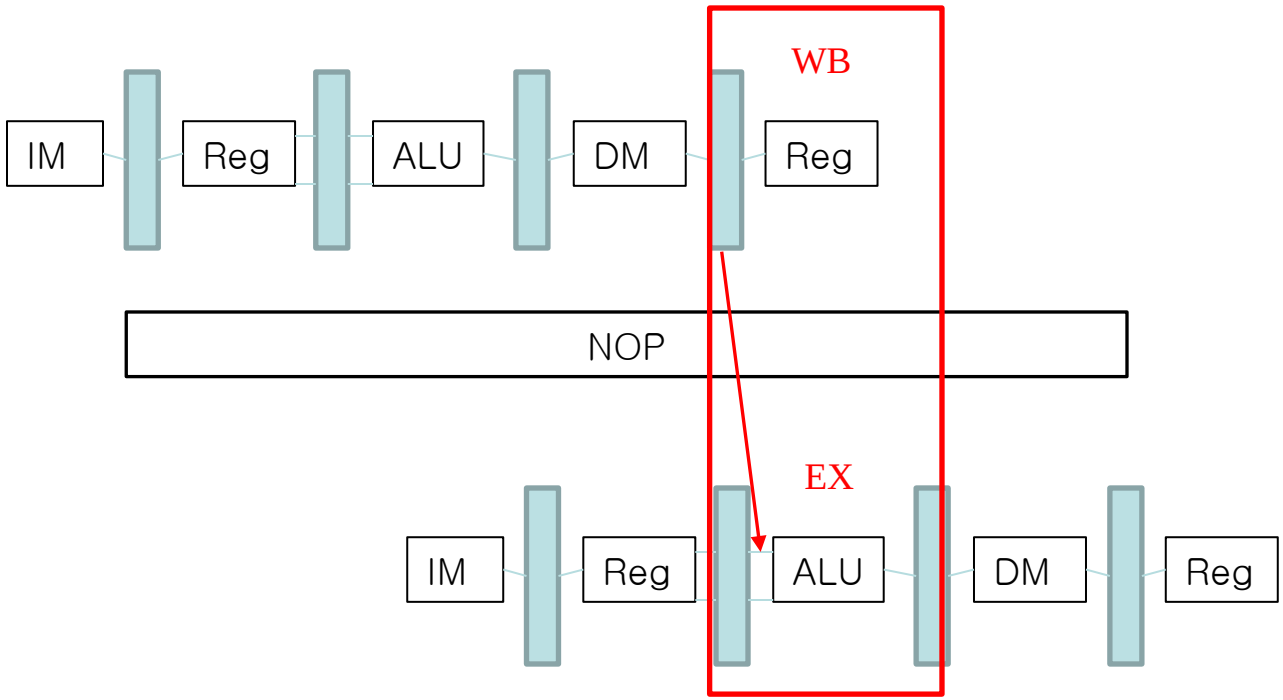
lui $4 0x0000 Fetch Decode Exe Mem Wb



Hazard example - ALU forwarding (10)

insertion_sort.asm

```
1   .text
2   main:  lui   $4, 0x0000
3          nop
4          ori   $4, $4, 0x2000
```



Hazard example - ALU forwarding (10)

M_TEXT_SEG.txt

lui

ori

```
1   00111100000001000000000000000000
2   00000000000000000000000000000000
3   00110100100001000010000000000000
4   00000000000000000000000000000000
5   00000000000000000000000000000000
6   00000000000000000000000000000000
7   00000000000000000000000000000000
8   00110100000001010000000001100100
9   00000000000000000000000000000000
10  00000000000000000000000000000000
11  00000000000000000000000000000000
12  00000000000000000000000000000000
13  00100000000010000000000000000001
14  00000000000000000000000000000000
```

M_TEXT_FWD.txt

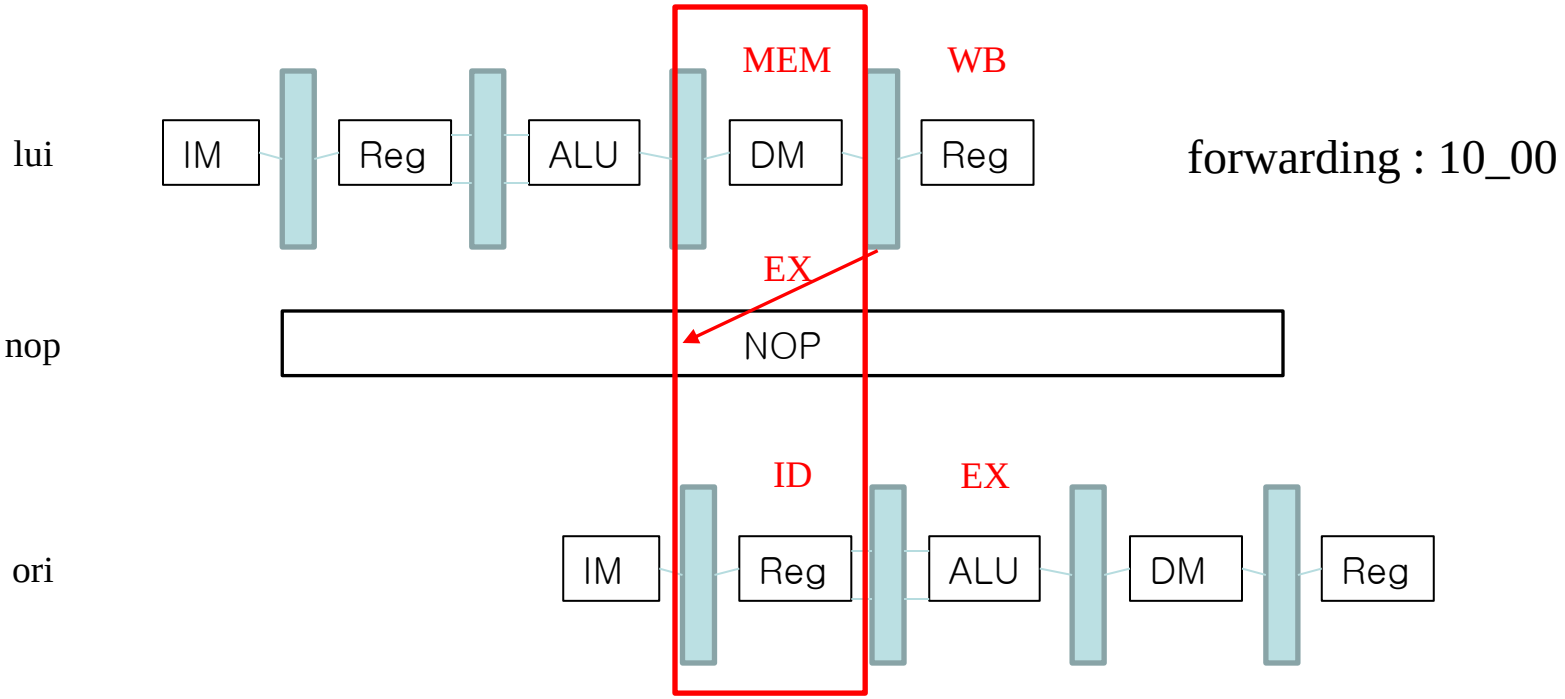
lui

ori

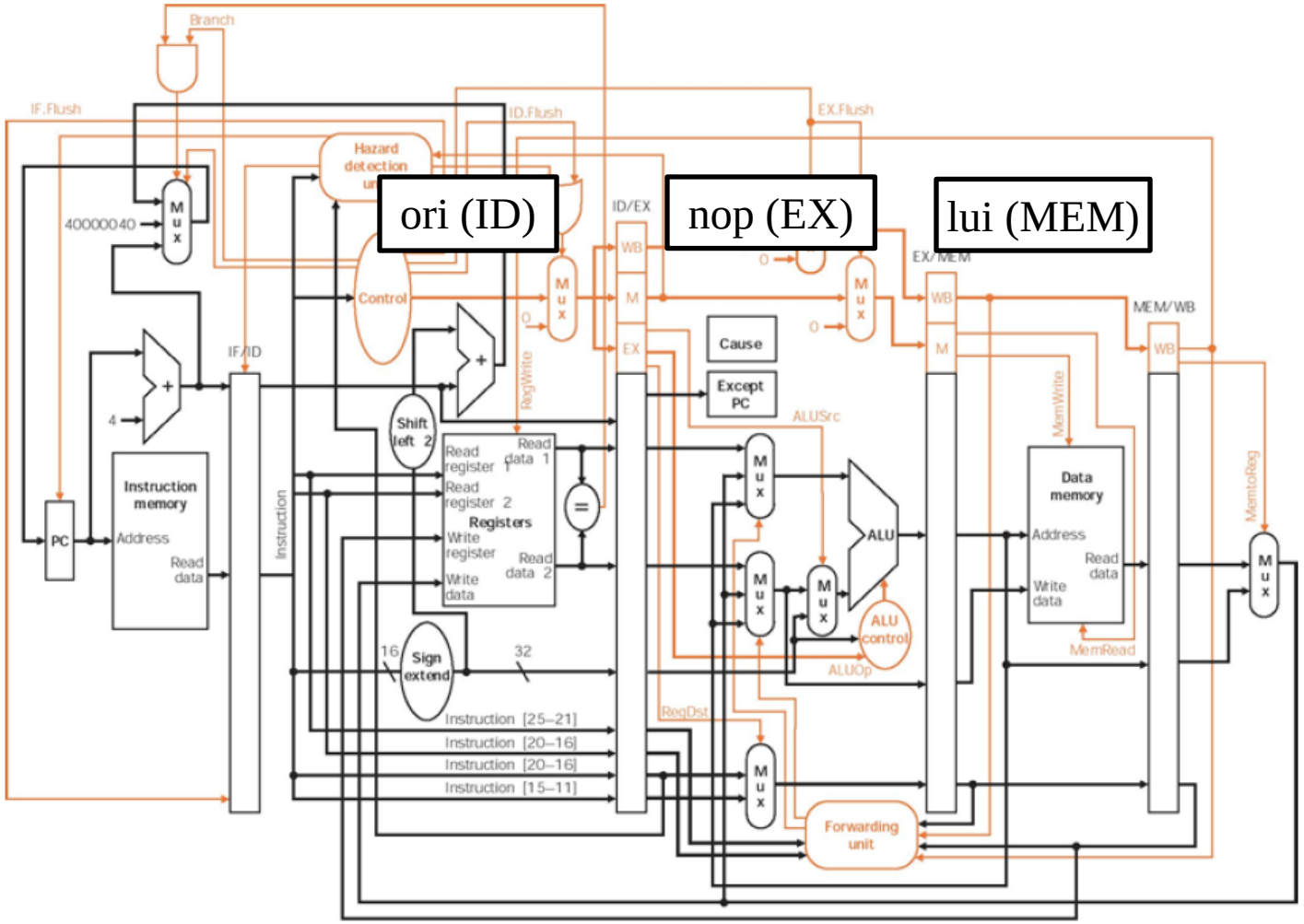
```
1   00_00  //  0x000
2   10_00  //  0x004
3   00_00  //  0x008
4   00_00  //  0x00C
5   00_00  //  0x010
6   00_00  //  0x014
7   00_00  //  0x018
8   00_00  //  0x01C
9   00_00  //  0x020
10  00_00  //  0x024
11  00_00  //  0x028
12  00_00  //  0x02C
13  00_00  //  0x030
14  00_00  //  0x034
```

Hazard example - ALU forwarding (10)

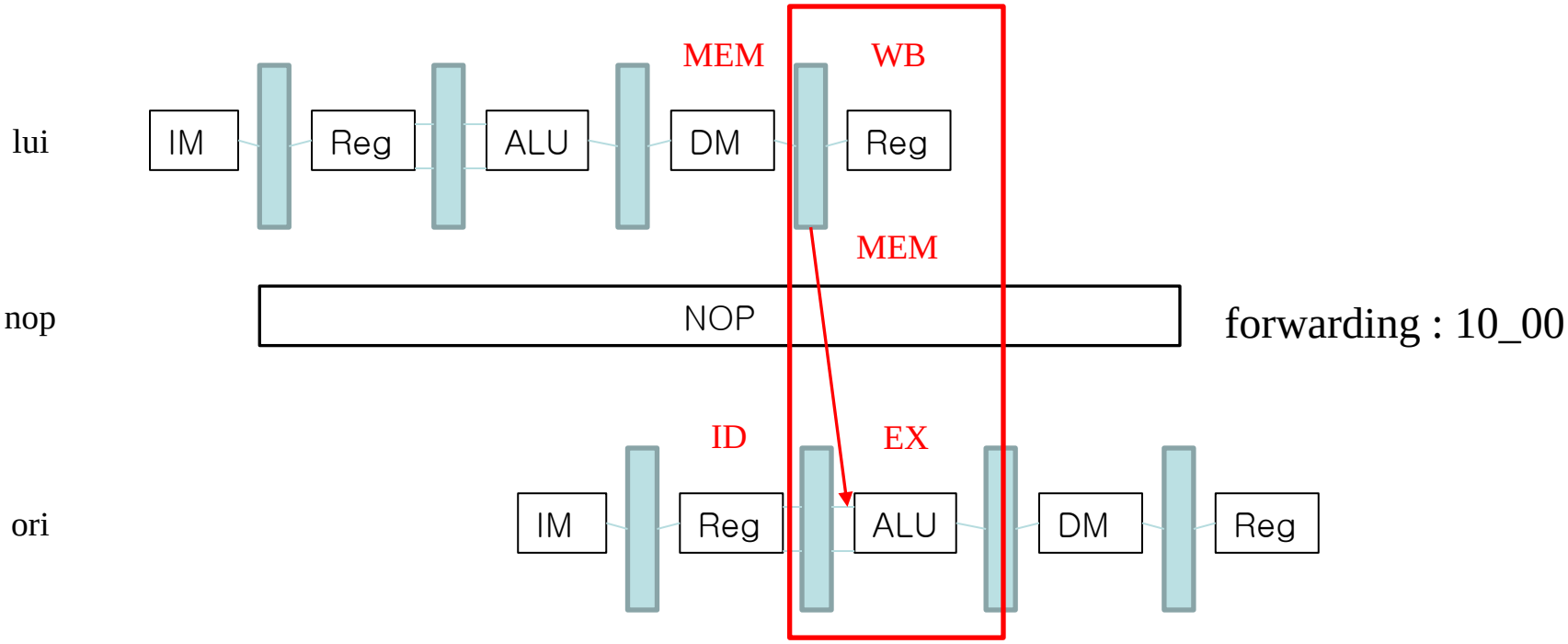
- Why is the address line like that?



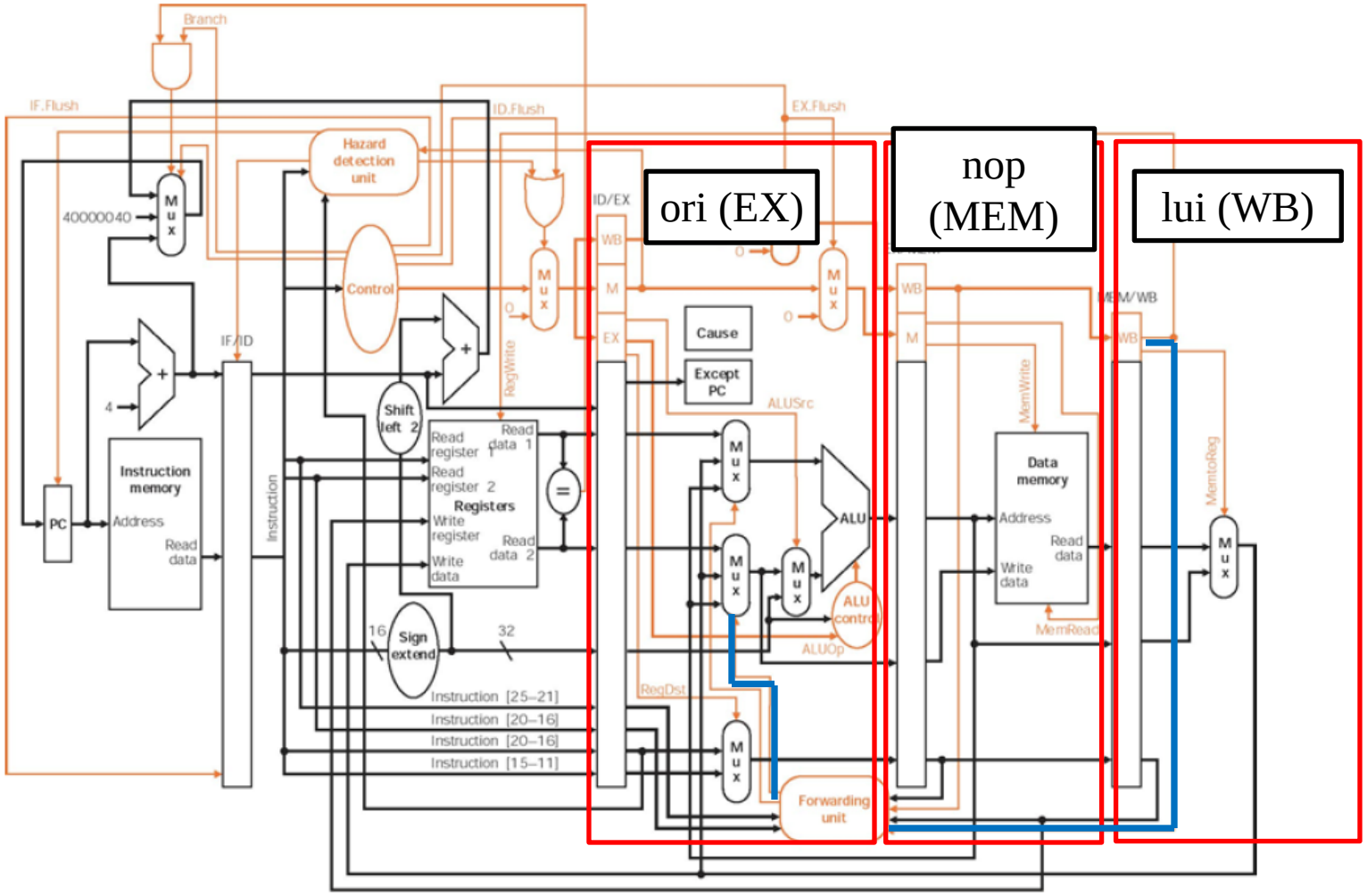
Hazard example - ALU forwarding (10)



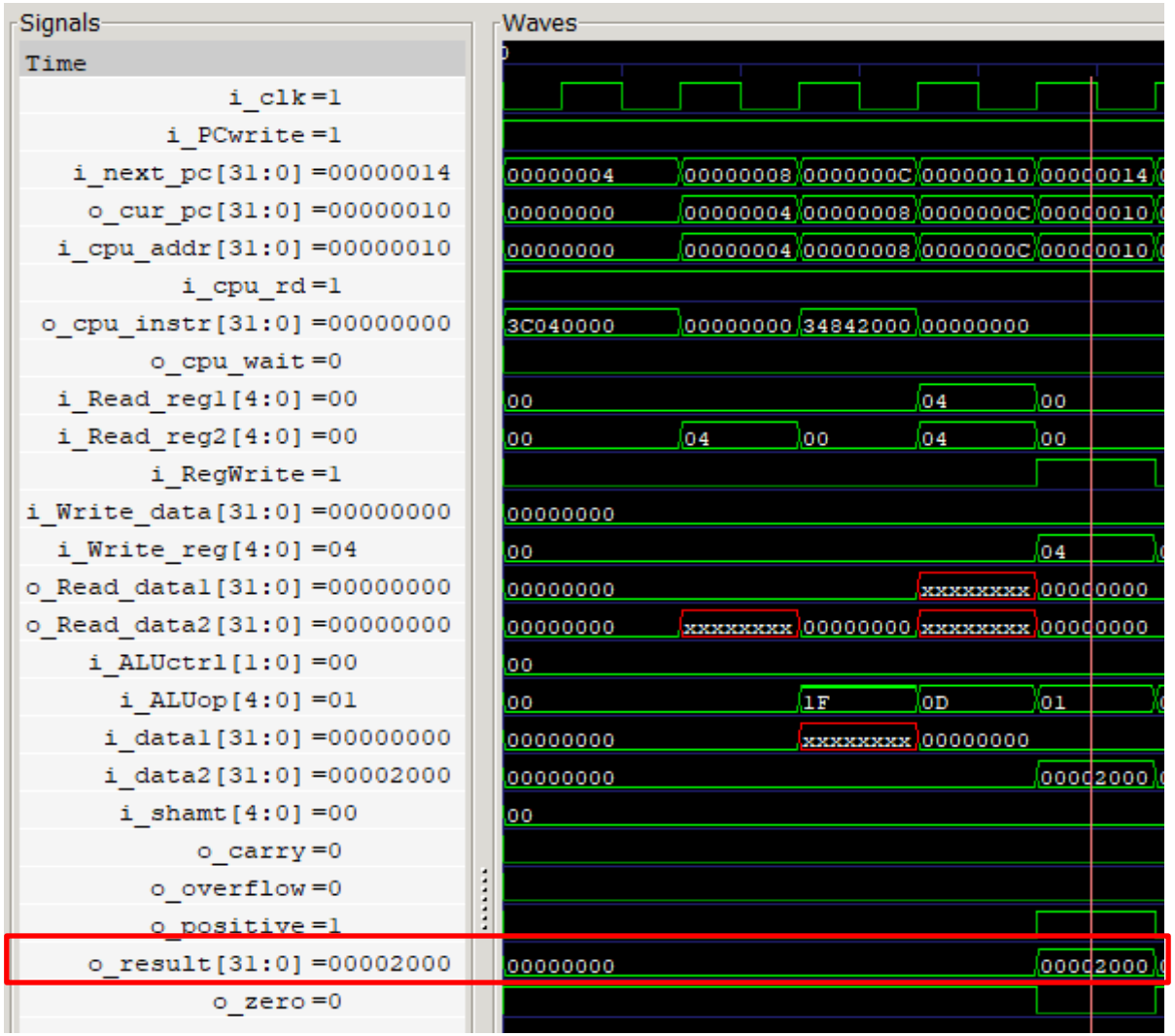
Hazard example - ALU forwarding (10)



Hazard example - ALU forwarding (10)



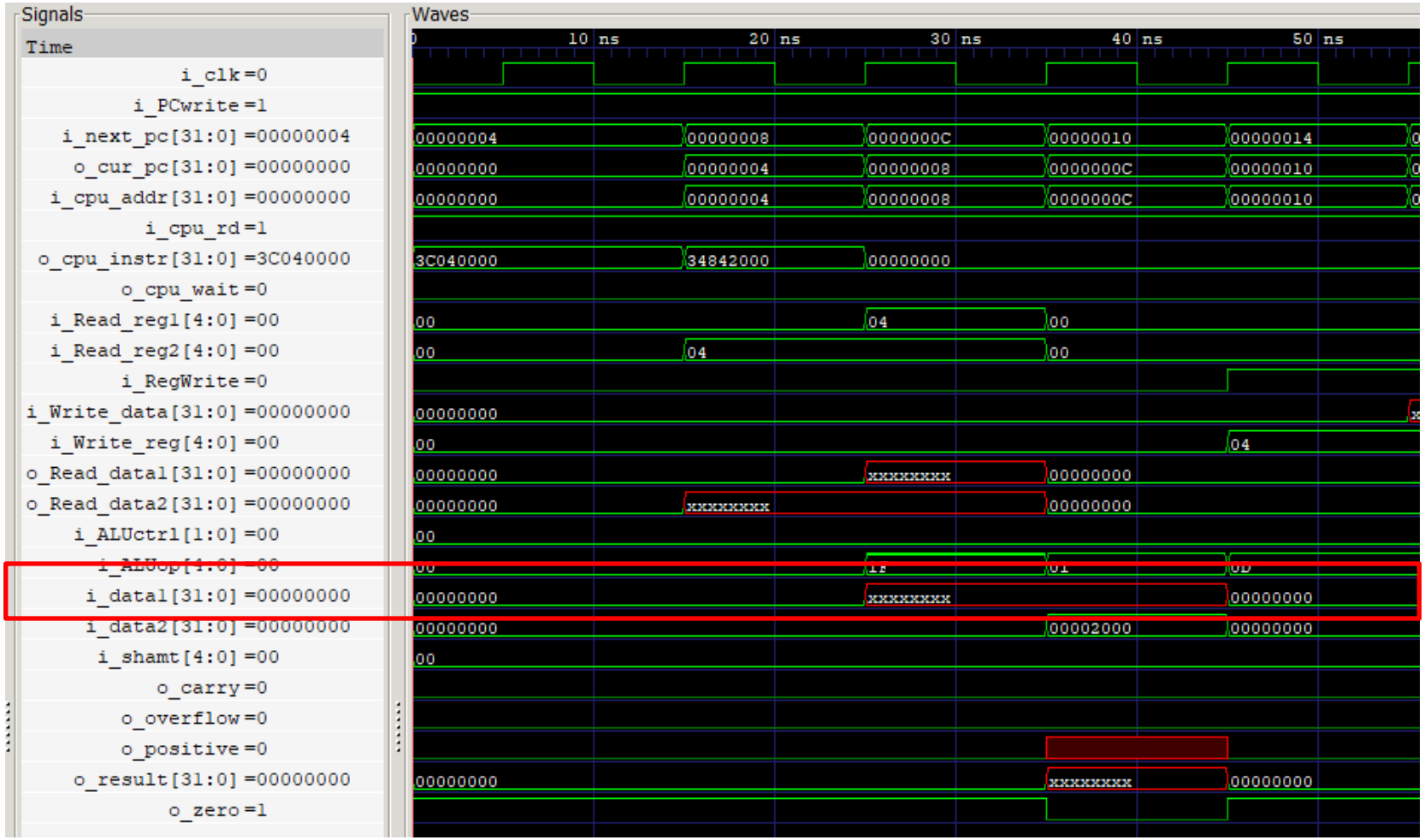
Hazard example - ALU forwarding (10)



MPLAB
Media Processing Laboratory

Check forwarding A or B

- You should check ALU port position when you forwarding



A port
B port

Check forwarding A or B

- You should check ALU port position when you forwarding

M_TEXT_FWD.txt

lui
ori

```
1    01_00  //  0x000
2    00_00  //  0x004
3    00_00  //  0x008
4    00_00  //  0x00C
5    00_00  //  0x010
6    00_00  //  0x014
7    00_00  //  0x018
8    00_00  //  0x01C
9    00_00  //  0x020
10   00_00  //  0x024
11   00_00  //  0x028
12   00_00  //  0x02C
13   00_00  //  0x030
14   00_00  //  0x034
```

A port_B port

Thank You