

# 1092 Digital System Design Final Project

## Pipelined RISC-V Design

*Announced at May 20, 2021*

### TA Information

**Baseline/ Extension:** 馬咏治  
**kane@access.ee.ntu.edu.tw**



### 1. Project Description

Table 1. Required Instruction Set

Name	Description
ADD	Addition, overflow detection for signed operand is not required*
ADDI	Addition immediate with sign-extension, without overflow detection*
SUB	Subtract, overflow detection for signed operand is not required*
AND	Boolean logic operation
ANDI	Boolean logic operation with 12bit of immediate
OR	Boolean logic operation
ORI	Boolean logic operation with 12bit of immediate
XOR	Boolean logic operation
XORI	Boolean logic operation with 12bit of immediate
SLLI	Shift left logical (zero padding)
SRAI	Shift right arithmetic (sign-digit padding)
SRLI	Shift right logical (zero padding)
SLT	Set less than, comparison instruction
SLTI	Set less than variable, comparison instruction
BEQ	Branch on equal, conditional branch instruction
BNE	Branch on not equal, conditional branch instruction
JAL	Unconditionally jump and link (Save next PC in \$rd)
JALR	Jump and link register(Save next PC in \$rd)
LW	Load word from data memory (assign word-aligned)
SW	Store word to data memory (assign word-aligned)
NOP	No operation(addi \$r0 \$r0 0)

\* Different from definition in [1], the exception handler **for arithmetic overflow** is not required.

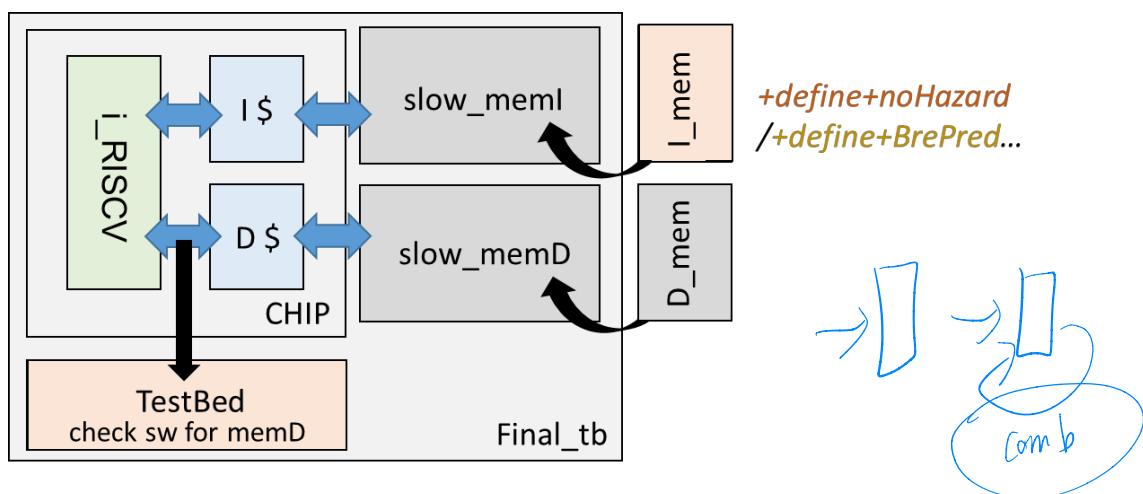
In final project, you are asked to design a **pipelined RISC-V processor** (**synchronous active low reset**) with **instruction cache** and **data cache**. This processor should at least support the instruction set defined in Table 1. The instruction set is referenced from Chapter2 (RV32I base integer instruction set) of [1], and we encourage you read it in detail.

The whole module hierarchy is shown in Figure 1. And the processor architecture is in Figure 2. As you see, this is modified from single-cycle architecture of our HW3. Your design should follow this **5-stage pipelined architecture**. You need to modify several parts to fit our specifications. For example, you need to add the path for **J-type instructions**.

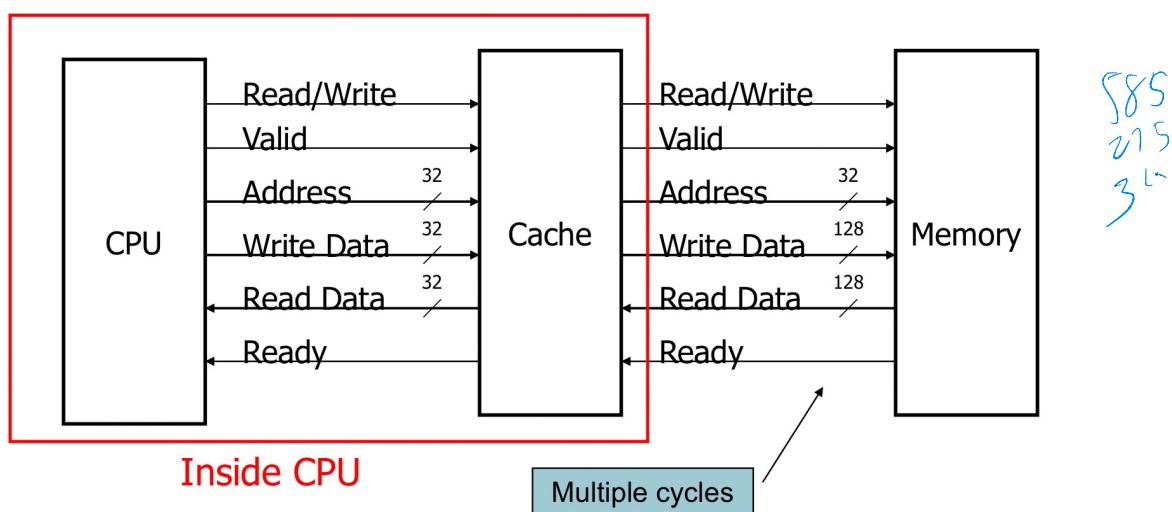
Also, you should **solve the hazards** by adding some circuits. There are 3 hazard categories should be properly handled in your pipelined processor:

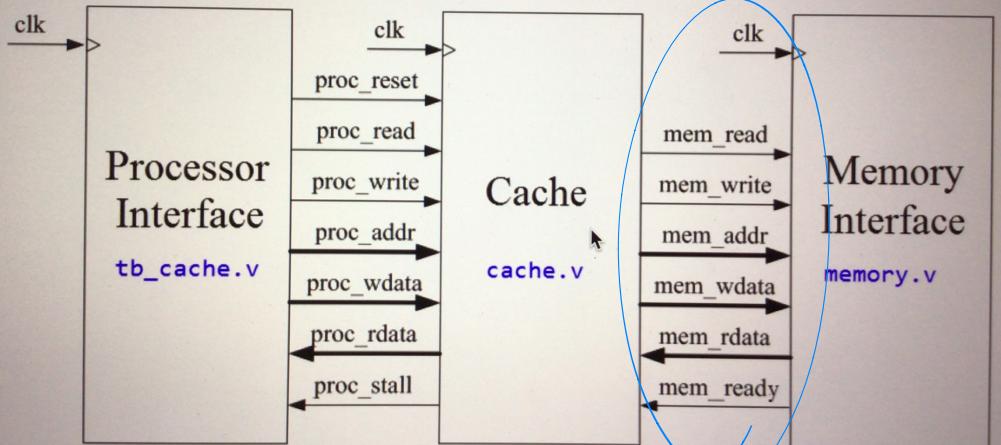
- 1) Structure hazard
- 2) Data hazard
- 3) Branch hazard

Although all of these hazards can be solved by insert NOP manually or automatically in your test program, we ask you to implement **data forwarding unit** and **pipeline stall unit** to solve these hazards.

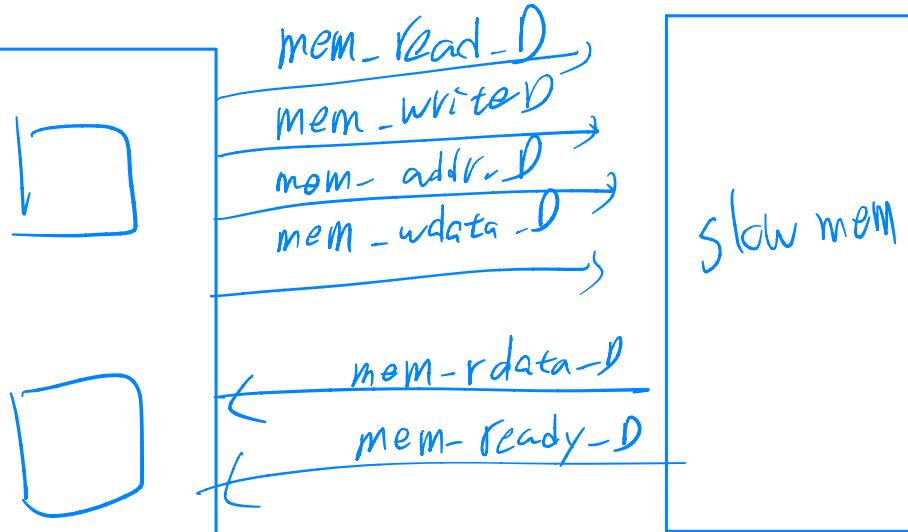
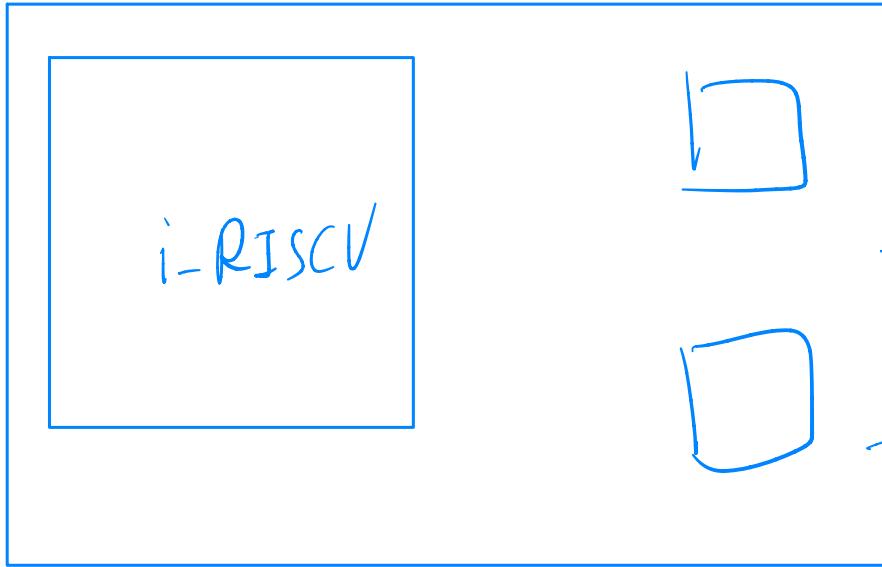


**Figure 1.** Module Hierarchy

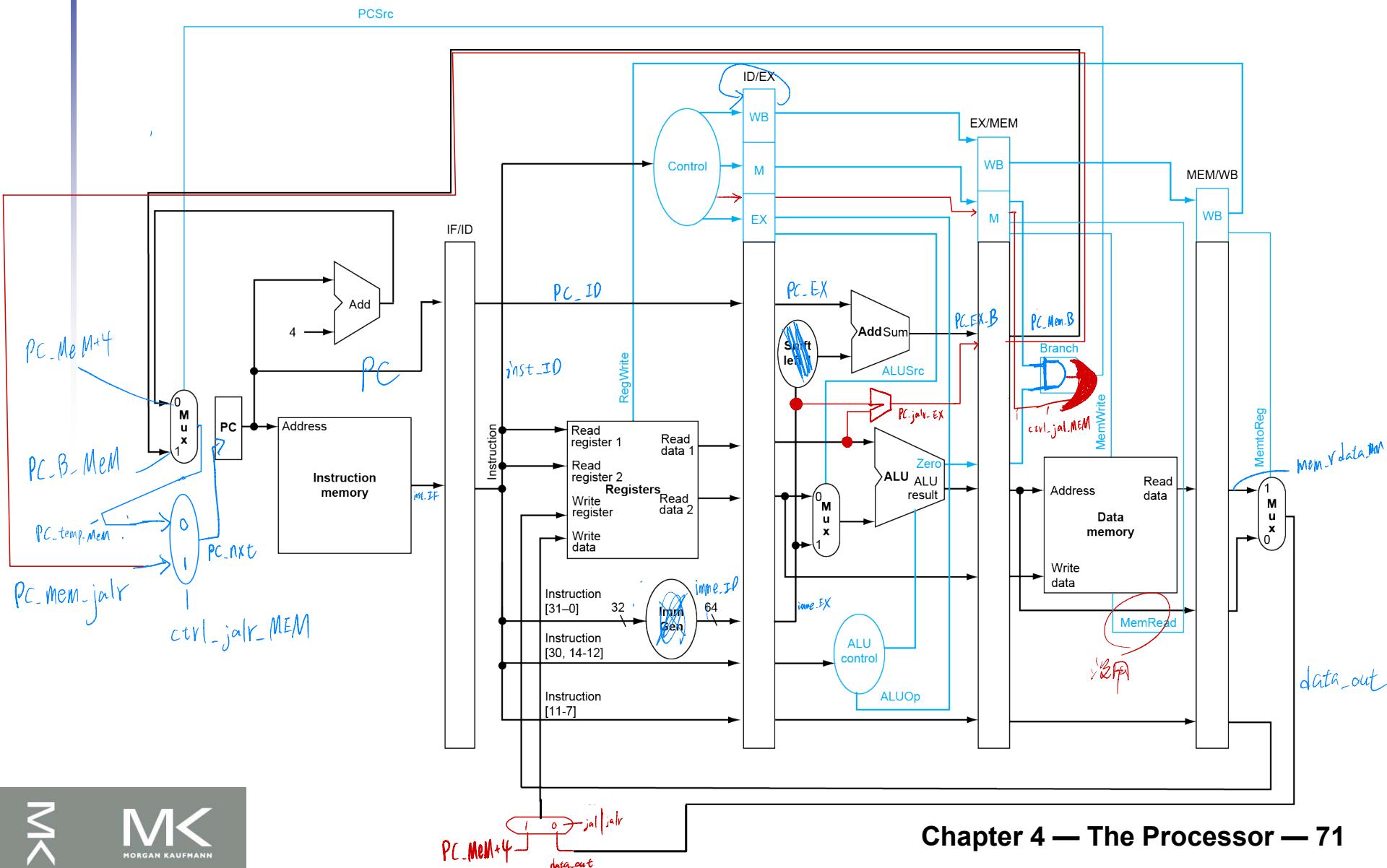




**Figure 2.** I/O specification and interface of the cache unit.



# Pipelined Control EX Mem WB



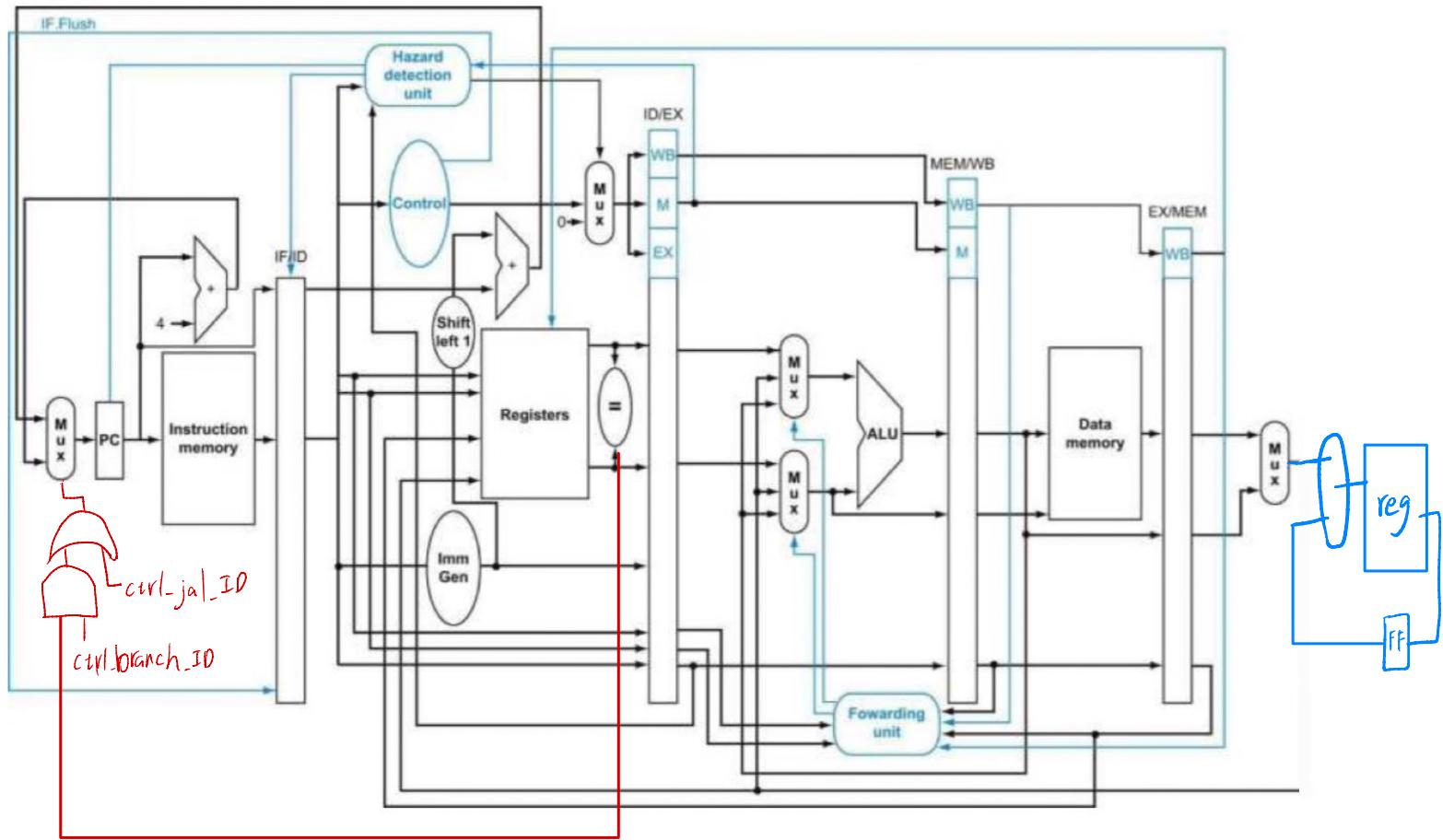
IF

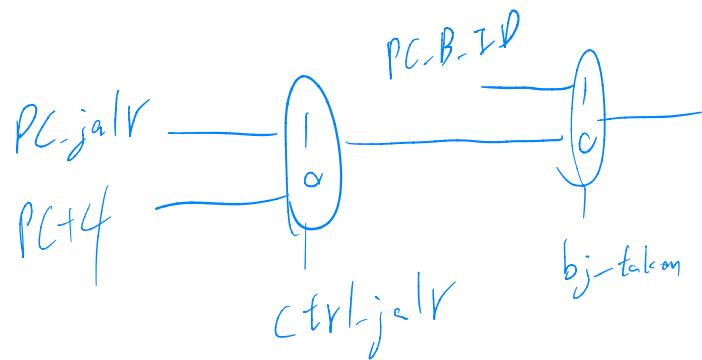
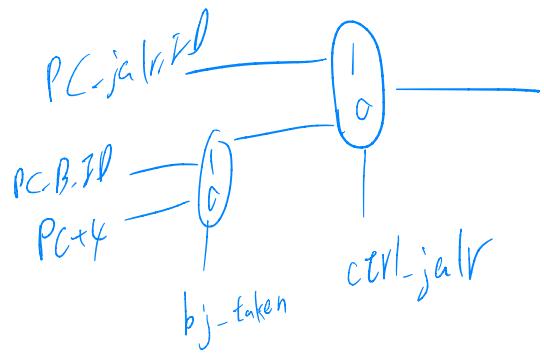
ID

EX

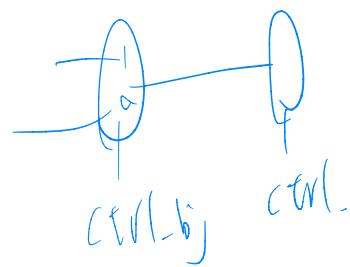
MEM

WB





$\left| \begin{matrix} 0 \\ 0 \\ 1 \\ 0 \end{matrix} \right|$



```

67
68 ADD 0000000 rs2 rs1 000 rd 0110011 R-type +
69 SUB 0100000 rs2 rs1 000 rd 0110011 R-type -
70 AND 0000000 rs2 rs1 111 rd 0110011 R-type &
71 OR 0000000 rs2 rs1 110 rd 0110011 R-type |
72 XOR 0000000 rs2 rs1 100 rd 0110011 R-type ^
73 SLT 0000000 rs2 rs1 010 rd 0110011 R-type <

74
75 ADDI imme[11:0] rs1 000 rd 0010011 I-type +
76 ANDI imme[11:0] rs1 111 rd 0010011 I-type &
77 ORI imme[11:0] rs1 110 rd 0010011 I-type |
78 XORI imme[11:0] rs1 100 rd 0010011 I-type ^
79 SLLI 0000000 sh rs1 001 rd 0010011 I-type <<
80 SRAI 0100000 sh rs1 101 rd 0010011 I-type >>>
81 SRLI 0000000 sh rs1 101 rd 0010011 I-type >>
82 SLTI imme[11:0] rs1 010 rd 0010011 I-type <
83 LW imme[11:0] rs1 010 rd 0000011 I-type +
84 JALR imme[11:0] rs1 000 rd 1100111 I-type +
85 NOP(addi)imm[11:0] rs1 000 rd 0010011 I-type +

86
87 BEQ imme[12|10:5] rs2 rs1 000 imm[4:1|11] 1100011
88 BNE imme[12|10:5] rs2 rs1 001 imm[4:1|11] 1100011

89
90 JAL imme[20|10:1|11|19:12] rd 1101111

91
92 SW imme[11:5] rs2 rs1 010 imm[4:0] 0100011
93
94

```

$\begin{array}{|c|c|} \hline 0 & 1 \\ \hline 0 & 1 \\ \hline \end{array}$  |  $\begin{array}{|c|} \hline b \\ \hline d \\ \hline \end{array}$   
 4b 2

$\begin{array}{|c|c|} \hline 0 & 1 \\ \hline 0 & 1 \\ \hline \end{array}$  |  $\begin{array}{|c|} \hline 0 \\ \hline 1 \\ \hline \end{array}$

0	0	0	0	0	0	add	
1	0	0	0	1	1	sub	
1	C	1	1	1	1	and	
3	0	0	1	1	1	or	
2	C	0	1	0	0	XOR	
4	0	1	0	0	<		
8	1	0	0	0	<<		
14	1	1	1	0	>		
15	1	1	1	1	>>		
func7[5]				func3		op[6:2]	
ADD	0	rs2	rs1	000	rd	01100	R-type +
ADDI	i		rs1	000	rd	00100	I-type +
LW	i		rs1	010	rd	00000	I-type +
JALR	i		rs1	000	rd	11001	I-type +
SW	i	rs2	rs1	010	imme	01000	S-type +
SUB	1	rs2	rs1	000	rd	01100	R-type -
BEQ	i	rs2	rs1	000	imm	11000	B-type
BNE	i	rs2	rs1	001	imm	11000	B-type
AND	0	rs2	rs1	111	rd	01100	R-type &
andi	i		rs1	111	rd	00100	I-type &
ORI	i		rs1	110	rd	00100	I-type
OR	0	rs2	rs1	110	rd	01100	R-type
XOR	0	rs2	rs1	100	rd	01100	R-type ^
XORI	i		rs1	100	rd	00100	I-type ^
SLT	0	rs2	rs1	010	rd	01100	R-type <
SLTI	i		rs1	010	rd	00100	I-type <
SLLI	0	sh	rs1	001	rd	00100	I-type <<
SRAI	1	sh	rs1	101	rd	00100	I-type >>>
SRLI	0	sh	rs1	101	rd	00100	I-type >>

$$alu(3) = !func3[1] \& func3[0] \& !op(6)$$

$$alu(2) = [func3[2] \& func3[0]] \mid \begin{cases} (!func2 \& func1[1] \& !func1[0]) \\ 1 \times 1 \end{cases}$$

!op(4)

$$alu(1) = func3(2)$$

$$alu(0) = func1[5] \& [0] \mid \begin{cases} 1 \times 1 \mid \begin{cases} func1[5] \& (!func1[1]) \\ op(5) \end{cases} \\ func1[5] \left( [0] \mid \begin{cases} X \times X & op(5) \end{cases} \right) \end{cases}$$

			$op[6:2]$
0	0 0 0		0 1 1 0 0
2	0   0		0 0 1 0 0
			0 0 0 0 0
1	c 0 1		1 1 0 0 1
3	c   1		c   0 0 0
6	0   0		1 1 0 0 0

$$type[2] = op[3]$$

$$type[1] = !op[6] \& !op[5] \quad | \quad op[6] \& op[5]$$

$$type[0] = \quad op[5] \quad \& \quad !op[4] \& !op[1]$$

alu[3] = func3(10) & fun1(5)

`alu[2] = func3: 100 | c/c & op[4] | func[0] & !func[65]`

$$\text{alu}[1] = \left( \text{func3}[2] \& \left( \text{func3}[1:0] = 11 \mid 10 \right) \right) \mid \begin{array}{l} \text{func3}[0] \neq \text{func1}[5] \\ \& !\text{op}[6] \end{array}$$

$$a/b(c)$$

$$a = b$$

$\Rightarrow a(c) \wedge b(a)$

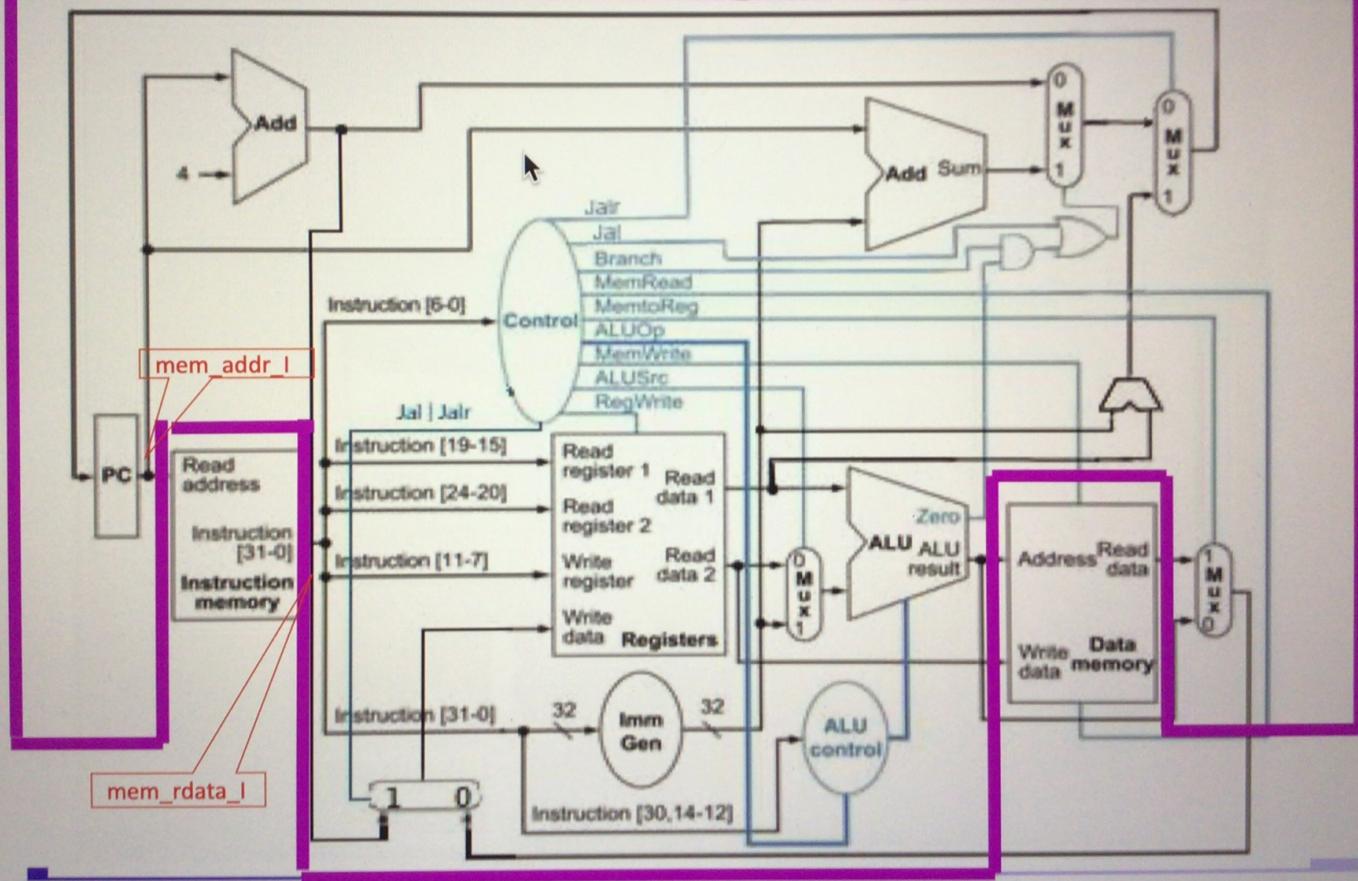
$\Rightarrow f(a(1) \wedge b(1))$

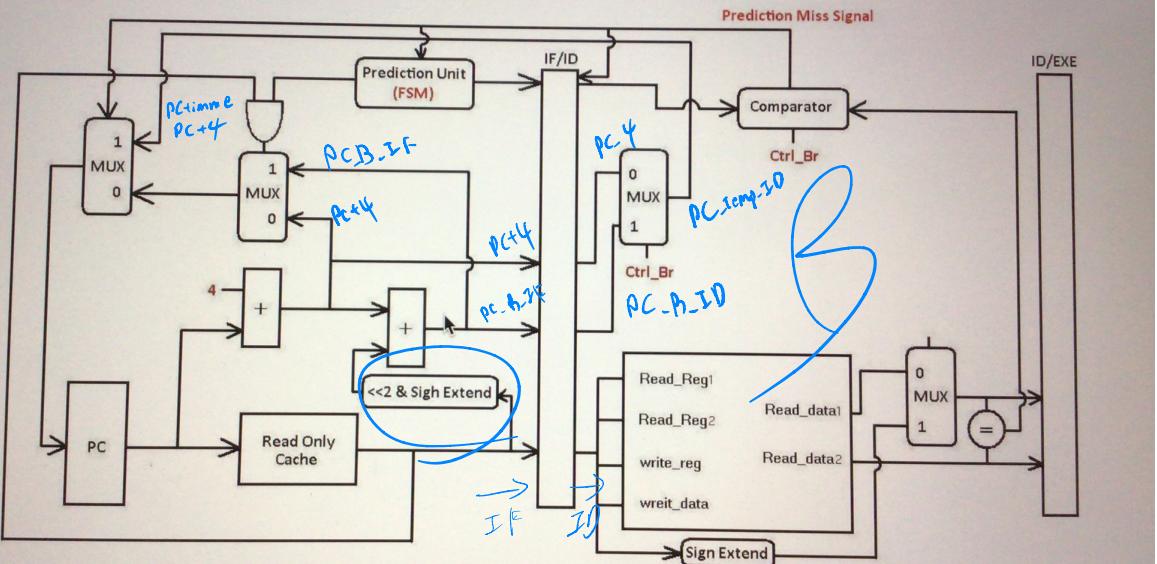
$\Rightarrow f(a(2) \wedge b(2))$

$\Rightarrow f(a(3) \wedge b(3))$

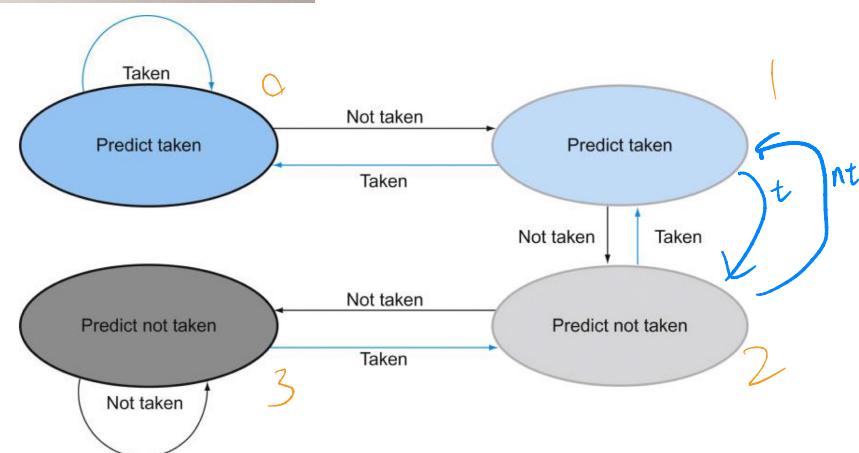
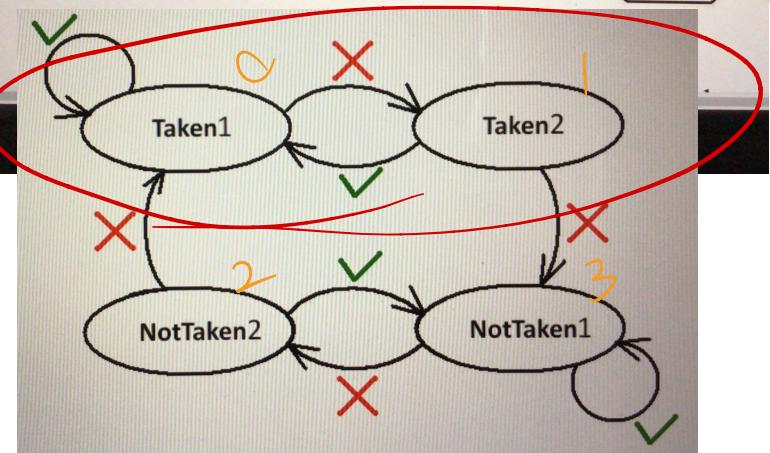
$\Rightarrow f(a(4) \wedge b(4))$

## Block Diagram(2/2)



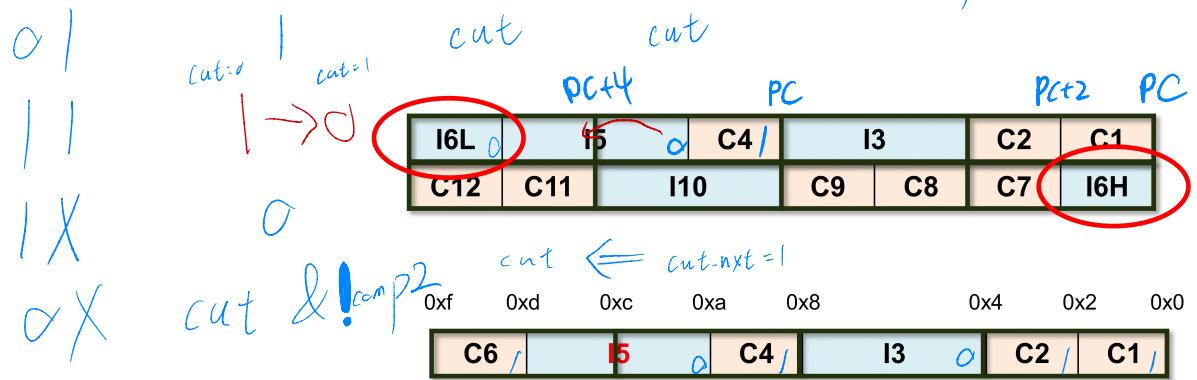
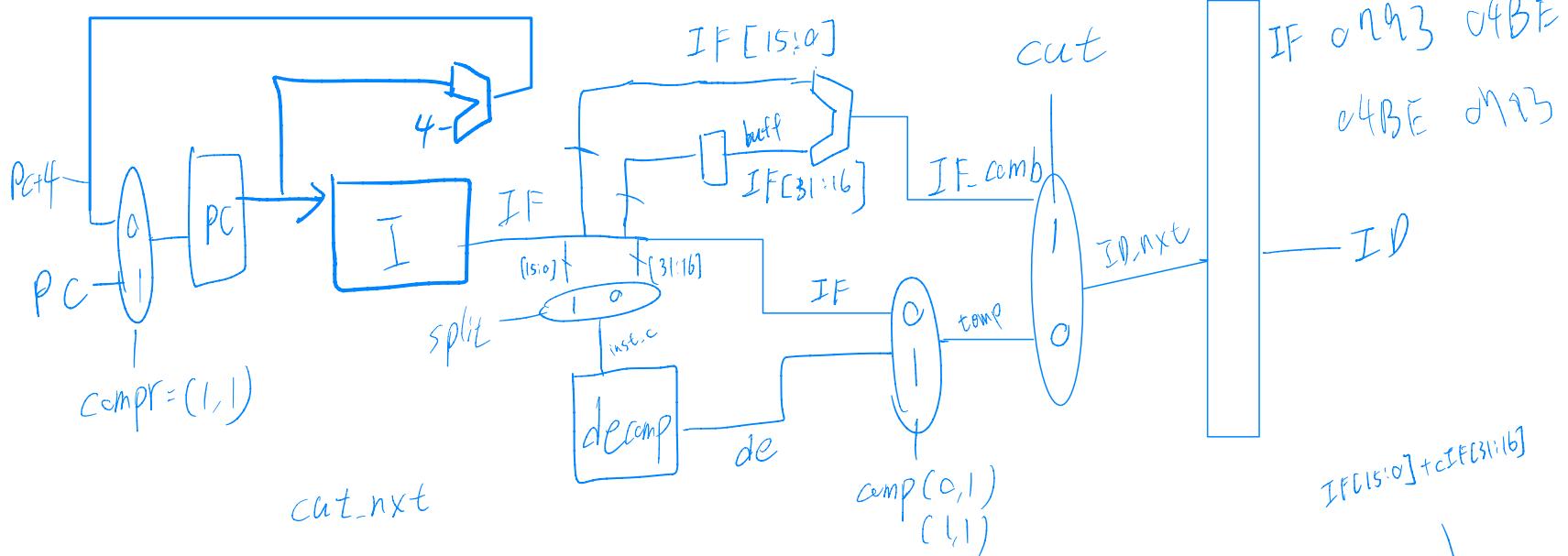


PC X  
branch  
PC+4



```
000000000000_00000_00000_0010011 //0x00// nop
/ 00000001001_00000_000_01000_0010011 //0x04// addi x8 x0 0x009 // a = 9
/ 00000001111_00000_000_01001_0010011 //0x08// addi x9 x0 0x00F // b = 15
/ 000000010010_00000_000_01010_0010011 //0x0C// addi x10 x0 0x012 // c = 18
/ 000000000000_00000_000_01011_0010011 //0x10// addi x11 x0 0x000
/ 000000000000_00000_000_01100_0010011 //0x14// addi x12 x0 0x000
/ 000000000000_00000_000_01101_0010011 //0x18// addi x13 x0 0x000
/ 000000000001_01011_000_01011_0010011 //0x1C// addi x11 x11 0x001 // Part_A
/ 0_00000_01011_01000_000_0100_0_1100011 //0x20// beq x8 x11 0x004 (to 0x28) // to Part_A_e
/ 1_111111100_1_1111111_00000_1101111 //0x24// jal x0 0xFFFFFC (to 0x1C) // to Part_A
/ 0100000_01011_01000_000_01110_0110011 //0x28// sub x14 x8 x11 // Part_A_end
/ 0000000_01110_00000_010_00000_0100011 //0x2C// sw x14 x0 0x000
/ 000000000001_01100_000_01100_0010011 //0x30// addi x12 x12 0x001 // Part_B
/ 0_00000_01100_01001_001_1000_0_1100011 //0x34// bne x9 x12 0x008 (to 0x44) // to Part_B_e
/ 0100000_01100_01001_000_01110_0110011 //0x38// sub x14 x9 x12
/ 0000000_01110_00000_010_00000_0100011 //0x3C// sw x14 x0 0x000
/ 0_0000001000_0_00000000_00000_1101111 //0x40// jal x0 0x00008 (to 0x50) // to Part_C
/ 0_00000_01011_01000_001_1000_0_1100011 //0x44// bne x8 x11 0x008 (to 0x4C) // Part_B_end,
/ 1_1111110100_1_1111111_00000_1101111 //0x48// jal x0 0xFFFF4 (to 0x30) // to Part_B
/ 0000000_01000_00000_010_00000_0100011 //0x4C// sw x8 x0 0x000 // Part_B_error
/ 000000000001_01101_000_01101_0010011 //0x50// addi x13 x13 0x001 // Part_C
/ 1_111111_01101_01010_001_1110_1_1100011 //0x54// bne x10 x13 0xFFE (to 0x50) // to Part_C
/ 0100000_01110_01010_000_01101_0110011 //0x58// sub x14 x10 x13 3 3 3 2 0 | 0
/ 0000000_01110_00000_010_00000_0100011 //0x5C// sw x14 x0 0x000
N N N
X T T T
N N N N
00 01 10 11
```

N T N T N T



$$IF[15:0] + c \cdot IF[31:16]$$

	cut	cut	nxt
/ X	1	0	0
1 1 0	0	0	0
X 0	1	1	1
0 X	1	1	1
0 1 0	1	1	1

0193

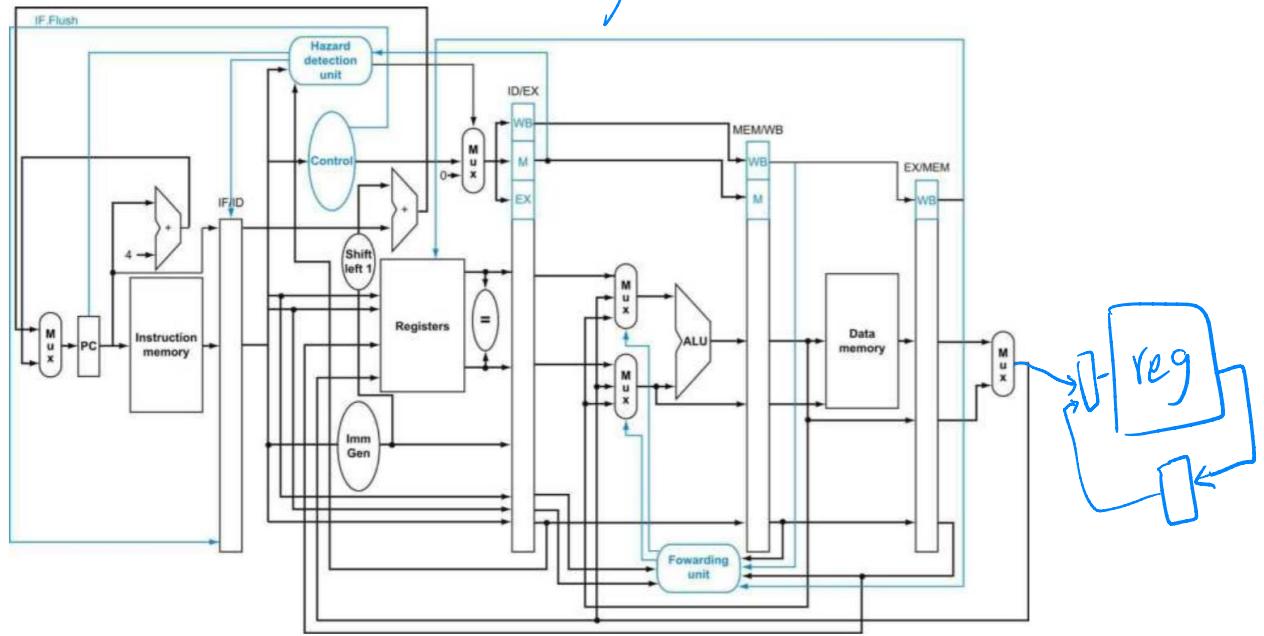
000\_0111\_1001\_0011

SW

SW

0100\_1000  
A81D 0001

1010\_1000\_0001\_1101



**Figure 2.** Simplified Pipeline Architecture of RISCV

## 2. Cache and Memory Interface

The instruction memory and data memory will not be contained in your design. The memory interface is left as module I/O. You have to use the provided slow memory model. **Do not synthesize the slow memory.**

The cache units are suggested to have the same block number (8) and block size (4) as in HW4. Besides, we do not restrict the replacement policy and writing policy of the cache design. You are encouraged to optimize the cache units to fit your RISC-V design.

## 3. Synthesis Notes

You should synthesize your design using TSMC 0.13 cell library, and the relevant files, e.g. *.synopsys\_dc.setup*, can be copied from previous HW or use the attached file. The design constraints are specified in “*CHIP\_syn.sdc*”. Note that the pipelined RISC-V, instruction cache and data cache are included in the *CHIP.v*. They should be synthesized together.

The post-synthesis simulation is required and all involved Verilog files should be all modeled by gate-level. Note that the maximum clock frequency must be verified by post-synthesis gate-level simulation. **And you are recommended to buffer the input signal to avoid timing violation.**

PC:

B8

$$x8 = 168$$

BC

jal (0xFFFFA6) × 1

08

$$sw \frac{x8}{0} \times 0 + 3FC$$

0C

jalr → × 1

C0

$$x1b = x0 + B8$$

C4

$$x17 = x0 + B8$$

C8

jalr ( $\frac{x1b + 0xF58}{B8}$ ) × 2

] F

10

$$x11 = x0 + 10$$

14

$$x9 = 0$$

18

$$x10 = 1$$

1C

$$x15 = 0$$

20

$$sw \frac{x9}{0} (\frac{x15 + 0}{0})$$

24

$$\frac{x15}{4} = \frac{x15 + 4}{0}$$

] F

28

$$sw \frac{x10}{1} (\frac{x15 + 0}{4})$$

2C

$$\frac{x8}{0} = \frac{x9}{0}$$

30

jal (0xFFFFEC) × 1

34



$$08 \quad sw \frac{x8}{0} \xrightarrow{\times 0 + 3FC}$$

0C

jalr → × 1

$$34 \quad \frac{x8}{1} = \frac{x10}{1}$$

38

jal (0xFFFFE8) × 1

$$08 \quad sw \frac{x8}{1} \xrightarrow{\times 0 + 3FC}$$

0C

jalr → × 1

$$3C \quad x12 = 2$$

$$40 \quad \frac{x10}{1} = \frac{x10 + x9}{0} \xrightarrow{\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}}$$

$$44 \quad \frac{x9}{1} = \frac{x10 - x9}{0} \xrightarrow{\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}}$$

$$48 \quad \frac{x15}{8} = \frac{x15 + 4}{4}$$

$$4C \quad sw \frac{x10}{1} \xrightarrow{(x15 + 0)}$$

50

$$\frac{x8}{1} = \frac{x10}{1}$$

54

jal (0xFFFFDA) × 1

$$08 \quad sw \frac{x8}{1} \xrightarrow{\times 0 + 3FC}$$

0C

jalr → × 1

$$58 \quad \frac{x12}{3} = \frac{x12 + 1}{2}$$

$$5C \quad \frac{x12}{3} \neq \frac{x11}{10} ? (FF2)$$

mem 0 : 0 |

4 : 1 |

8 : 1 2

C : 2 0

10 : 3 :

x13 x14

40	$\frac{x10}{1} = \frac{x10+9}{1}$	60 jalr $x0 \times 2$	84 SW $\times 14 \xrightarrow{\text{?}} (\times 11 + 0)$
44	$\frac{x9}{1} = \frac{x10}{1} - \frac{x9}{0}$	CC jalr $\times 2 (\times 17 + \text{FAC})$	88 SW $\times 13 \xrightarrow{\text{?}} (\times 11 + 4)$
48	$\frac{x15}{c} = \frac{x15}{8} + 4$	64 $x9 = 3C$	8C $\frac{x11}{4} = \frac{x11}{0} + 4$
4C	SW $\frac{x10}{1} (\times 15 + 0)$	68 $x10 = 0$	90 bne $\frac{x11}{4} \neq \frac{x12}{3C} ? (\text{FF2})$
50	$x8 = \frac{x10}{1}$	6C $\frac{x12}{3C} = \frac{x9}{3C} - \frac{x10}{0} \leftarrow$	94 $\frac{x10}{4} = \frac{x10}{0} + 4$
54	jal (FFFF0A) $\times 1$	70 $x11 = 0$	98 bne $x10 \neq \frac{x9}{3C} ? \text{ FEA}$
08	SW $\frac{x8}{1} \xrightarrow{\text{?}} \times 0 + 3FC$	74 lw $\times 13 \left( \frac{x11+0}{0} \right) \boxed{F}$	9C $x9 = 40$
0C	jalr $\rightarrow \times 1$	WB78 lw $\times 14 \left( \frac{x11+4}{0} \right) \boxed{J}$	A0 $x10 = 0$
58	$\frac{x12}{4} = \frac{x12}{3} + 1$	MEM 	A4 lw $\times 8 \xrightarrow{\text{?}} (\times 10 + c)$
5C	$\frac{x12}{4} \neq \frac{x11}{10} ? (\text{FF2}) \rightsquigarrow$	EX7Cslt $\frac{x15}{1}, \frac{x13}{5}, \frac{x14}{8}$	A8 jal FFFF00 $\times 1$
		IV80 beg $x15 = x0 ? (0 \times 8C)$	

08 SW  $\times 8$   $\times 0 + 3FC \leftarrow$

0C jal  $r_3 \rightarrow x1$

A0  $x10 = \frac{x10}{4} + 4$

B0  $x10 \neq \frac{x9}{40} ? FFA$

A4 lw  $\times 8$   $(\times 10 + c)$

A8 ja| FFFFBO x1

hazard

$0 \rightarrow 1 \rightarrow 1 \rightarrow 2 \rightarrow 1 \rightarrow 2 \rightarrow 3$   
 $\rightarrow 0 \rightarrow 3 \rightarrow 0 \rightarrow 3 \rightarrow 4$   
 $\rightarrow 5 \rightarrow 0 \rightarrow 5 \rightarrow 6$

$4 \rightarrow 5 \rightarrow 0 \rightarrow 5 \rightarrow$

$\rightarrow 4 \rightarrow 5 \rightarrow 0 \rightarrow 5 \rightarrow$

$\rightarrow 4 \rightarrow 5 \rightarrow 0 \rightarrow 5 \rightarrow$

Fibonacci loop

$7 \rightarrow 8 \rightarrow 9 \rightarrow$   
 $\rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow$

bubble inner  
loop

$11 \rightarrow 10 \rightarrow 0 \rightarrow 10$   
 $\rightarrow 11 \rightarrow 10 \rightarrow 0 \rightarrow 10$

bubble output

$\rightarrow 13$  finish

	0	1
0	0 → 4	12 → 0
1	1	5
2	2	6
3	11	3

2-way

→ mem prefetch

11	1	2	3	4	5	6	7	8	9	10
addV	0	11	→ 12	1	→ 2	→ 3	→ 4	→ 5	0	6
modulo	0	3	0	1	2	3	0	1	0	2

$$\text{miss} = 10$$

block	0	1	2	3	4	5	6	7	full
0	11	12	1	2	3	4	5		
miss = 9	6								

	0	1	2	3
0	0	12 → 6	2	4
1	11	1	3	5

$$\text{miss} = 9$$

0	$4 \rightarrow 12$	$0 \rightarrow 8$
1	$1 \rightarrow 13$	$5 \rightarrow 9$
2	2	6
3	$11 \rightarrow 7$	3

2-way

順序 1 2 3 4 5  
 addr 12 → 13 7 → 8 → 9  
 modulo 0 1 3 0 1

→ mem prefetch

miss = 5

block	0	1	2	3	4	5	6	7	full
0	6	12	1	2	3	4	5		

miss = 4

13 7 8 9

	0	1	2	3
0	0	6	2	4
1	11	1	3	5

miss = 9

C	$12 \rightarrow 0$	8
I	$13 \rightarrow 1$	9 $\rightarrow 13$ finish
2	$2 \rightarrow 10$	6
3	7	$3 \rightarrow 11$

2-way

順序  
addr  
modulo

1	2	3	4
10	0	1	11
2	0	1	3

miss = 4

finish

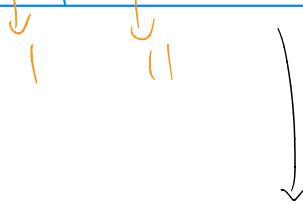
13

1

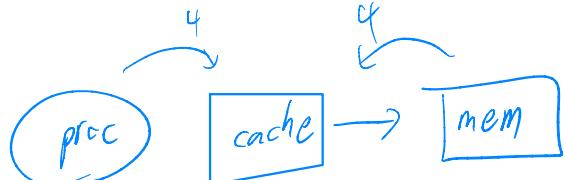
miss + 1

block	0	1	2	3	4	5	6	7
	0	6	12	13	7	8	9	5

miss = 3

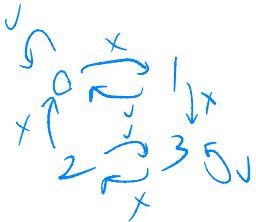


finish : miss + 0



$\begin{smallmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & c & 0 \\ 3 & 2 & 1 & 0 \end{smallmatrix}$

3021



04	$x8 = 3$	34	$x9 \pm x12 ? 0x044$
08	$x9 = 6$		$\frac{6}{1}$
0C	$x10 = 8$	38	$x14 = x9 - x12$
10	$x11 = 0$	3C	sw x14 0x00
14	$x12 = 0$	40	jal 0x50
18	$x13 = 0$	44	$x8 \pm x11 ? 0x4\cancel{C}$
1C	$x11 = \frac{x11+1}{0}$		$\frac{54}{3} \leftarrow$
20	$\frac{x8}{3} = \frac{x11}{1} ? 0x2\cancel{8}$	48	jal 0x30
24	jal 0x1C	4C	sw x8 0x00
28	$\frac{x14}{0} = \frac{x8 - x11}{0}$	50	$\frac{x13}{1} = \frac{x13+1}{0}$
2C	sw x14 0x00	54	$\frac{x10}{8} = \frac{x13}{0} 0x50 \leftarrow$
30	$\frac{x12}{1} = \frac{x12+1}{0}$		

58  $\frac{x14 = x10 - x13}{f} 10 \frac{1}{1}$

5C sw x14 0x00

e8  
 $\begin{array}{r} 1110 \\ 0001 \\ 0001 \\ 0001 \end{array} \rightarrow 1000$   
 $\begin{array}{r} 0111 \\ 1011 \\ 1011 \\ 1011 \end{array}$   
 $\begin{array}{r} 11 \\ 11 \\ 11 \\ 11 \end{array}$

B80d

$$64 \times 8 = 0x168$$

$$68 \times 14 = 0$$

6C jal output  $\times 1$

$$0C sw \frac{x8}{0x168} \frac{(\times 14 + 0)}{0}$$

$$10 jalr x0 \times 1+0 \text{ WB}$$

$$70 \times 8 = 0xDE \text{ MEM}$$

$$74 \times 8 = \times 8 \ll 8 \text{ EX}$$

$$78 \times 8 = \times 8 + 0xAD \text{ ID}$$

7C sw  $\frac{x8}{(\times 0+c)}$

$$8C jal \text{ output } \times 1$$

$$0C sw \frac{x8}{(\times 14 + 0)}$$

$$10 jalr x0 \times 1+0$$

$$84 \times 8 = 0xf6$$

$$88 \times 8 = \times 8 \ll 8$$

$$8C \times 8 = \times 8 + 0x25 \text{ MEM}$$

$$90 sw \frac{x8}{(\times 0 + 0x04)} \text{ EX}$$

$$94 jal \text{ output } \text{ ID}$$

$$0C sw \frac{x8}{(\times 14 + 0)}$$

$$10 jalr x0 \times 1+0$$

$$98 \times 16 = \text{ Mult}$$

$$9C jal \times 16(\text{Mult}) \times 1$$

$$14 \times 2 = \times 1 + 4 (\text{A4})$$

$$18 \times 9 \leftarrow (\times 14 + 0)$$