

### **Computer Architecture Experiments**

# Topic 1. Pipelined CPU supporting RISC-V RV32I Instructions

College of Computer Science & Technology

2022年9月

#### **Outline**



- Experiment Purpose
- Experiment Task
- Basic Principle
- Operating Procedures
- Checkpoints







- Understand RISC-V RV32I instructions
- Master the design methods of pipelined CPU executing RV32I instructions
- Master the method of Pipeline Forwarding Detection and bypass unit design
- Master the methods of 1-cycle stall of Predict-not-taken branch design
- Master methods of program verification of Pipelined CPU executing



# **Experiment Task**



- Design of Pipelined CPU executing RV32I instructions.
  - Design datapath
  - Design Bypass Unit
  - Design CPU Controller

Verify the Pipelined CPU with program and observe the execution of program





31 30	5 24 21	20	19	15 14 12	2 11 8	7	6 0	
funct7	rs2		rs1	funct3	rd	l	opcode	R-type
imm[	1:0]		rs1	funct3	rd	l	opcode	I-type
imm[11:5]	rs2		rs1	funct3	imm[	4:0]	opcode	S-type
imm[12] imm[10:5]	rs2		rs1	funct3	imm[4:1]	imm[11]	opcode	B-type
	imm[31	:12]			rd	l	opcode	U-type
imm[20] imm[	0:1]	imm[11]	imn	1[19:12]	ro	l	opcode	J-type





31	25	24 20	19	15 14	12	11	76 0	
		imm[31:12]				rd	0110111	U lui
		imm[31:12]				rd	0010111	U auipc
	imn	n[20 10:1 11 19:	12]			rd	1101111	J jal
	imm[11:0]		rs1	0	000	rd	1100111	I jalr
	imm[12 10:5]	rs2	rs1	0	000	imm[4:1 11]	1100011	B beq
	imm[12 10:5]	rs2	rs1	0	001	imm[4:1 11]	1100011	B bne
	imm[12 10:5]	rs2	rs1	1	.00	imm[4:1 11]	1100011	B blt
	imm[12 10:5]	rs2	rs1	1	.01	imm[4:1 11]	1100011	B bge
	imm[12 10:5]	rs2	rs1	1	10	imm[4:1 11]	1100011	B bltu
	imm[12 10:5]	rs2	rs1	1	11	imm[4:1 11]	1100011	B bgeu





31 2	5 24 2	0 19 1	5 14 12	11	7 6 0	_
imm[11:0	]	rs1	000	rd	0000011	I lb
imm[11:0	]	rs1	001	rd	0000011	I lh
imm[11:0	]	rs1	010	rd	0000011	I lw
imm[11:0	]	rs1	100	rd	0000011	I lbu
imm[11:0	]	rs1	101	rd	0000011	I lhu
imm[11:5]	rs2	rs1	000	imm[4:0]	0100011	S sb
imm[11:5]	rs2	rs1	001	imm[4:0]	0100011	S sh
imm[11:5]	rs2	rs1	010	imm[4:0]	0100011	S sw





31 25	5 24 20	19 1	5 14 12	11 7	6 0	
imm[11:0]	· 	rs1	000	rd	0010011	I addi
imm[11:0]		rs1	010	rd	0010011	I slti
imm[11:0]		rs1	011	rd	0010011	I sltiu
imm[11:0]		rs1	100	rd	0010011	I xori
imm[11:0]		rs1	110	rd	0010011	I ori
imm[11:0]		rs1	111	rd	0010011	I andi
0000000	shamt	rs1	001	rd	0010011	I slli
0000000	shamt	rs1	101	rd	0010011	I srli
0100000	shamt	rs1	101	rd	0010011	I srai





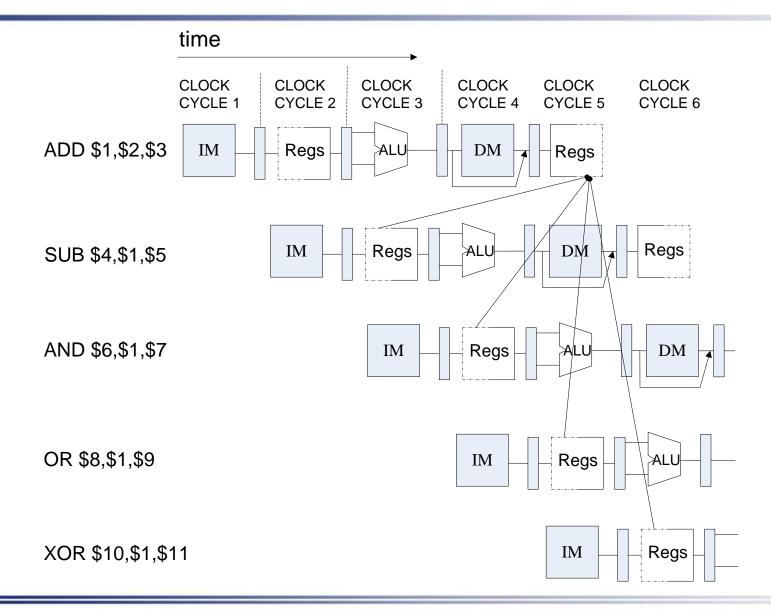
31	25 24 20	19 15	14 12	11 7	6 0	ı
0000000	rs2	rs1	000	rd	0110011	R add
0100000	rs2	rs1	000	rd	0110011	R sub
0000000	rs2	rs1	001	rd	0110011	R sll
0000000	rs2	rs1	010	rd	0110011	R slt
0000000	rs2	rs1	011	rd	0110011	Rsltu
0000000	rs2	rs1	100	rd	0110011	R xor
0000000	rs2	rs1	101	rd	0110011	R srl
0100000	rs2	rs1	101	rd	0110011	R sra
0000000	rs2	rs1	110	rd	0110011	R or
0000000	rs2	rs1	111	rd	0110011	R and



#### Instruction Demo

#### Data Hazard的条件: 最多间隔一条指令

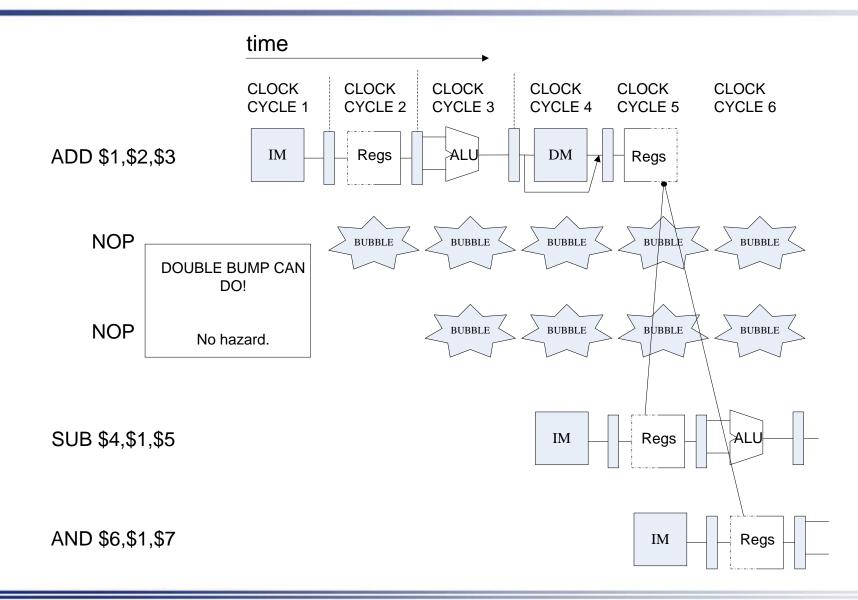






#### **Data Hazard Causes Stalls**



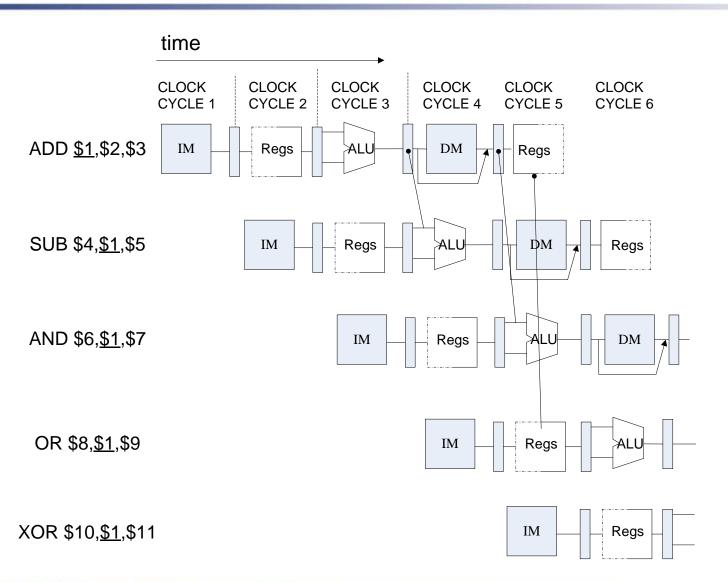








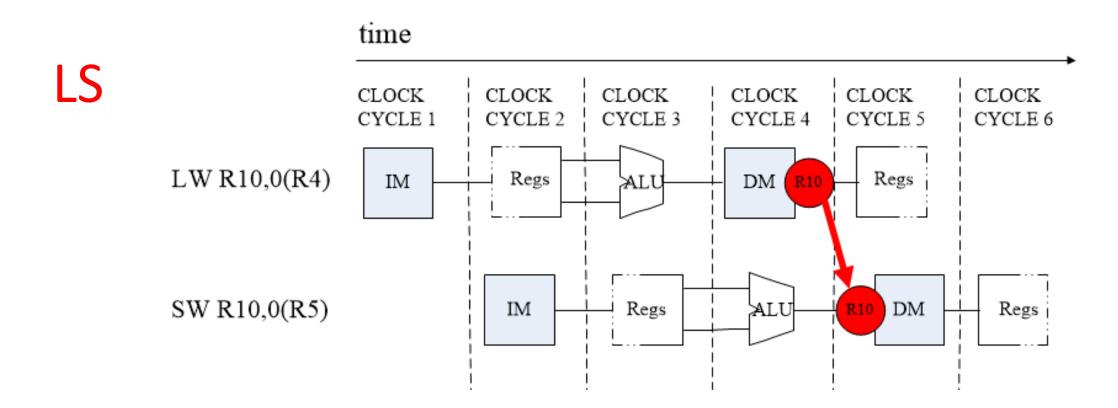
RR





### Special case: SW After LW

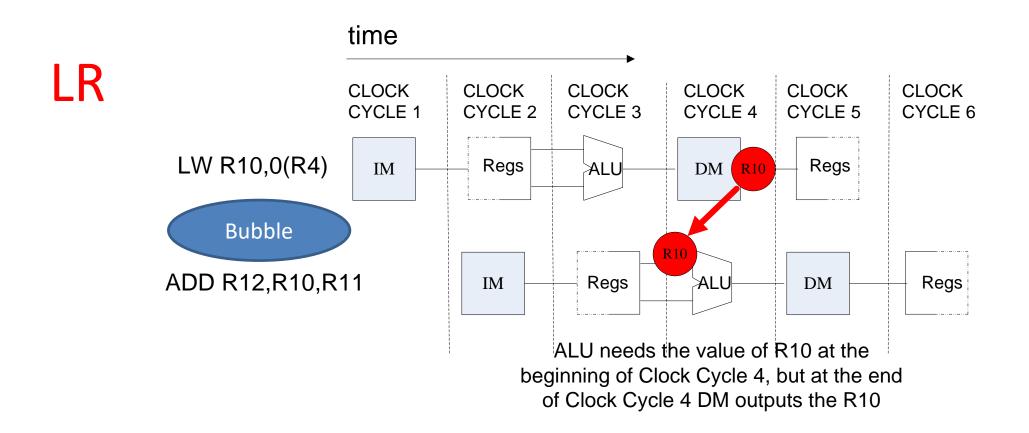








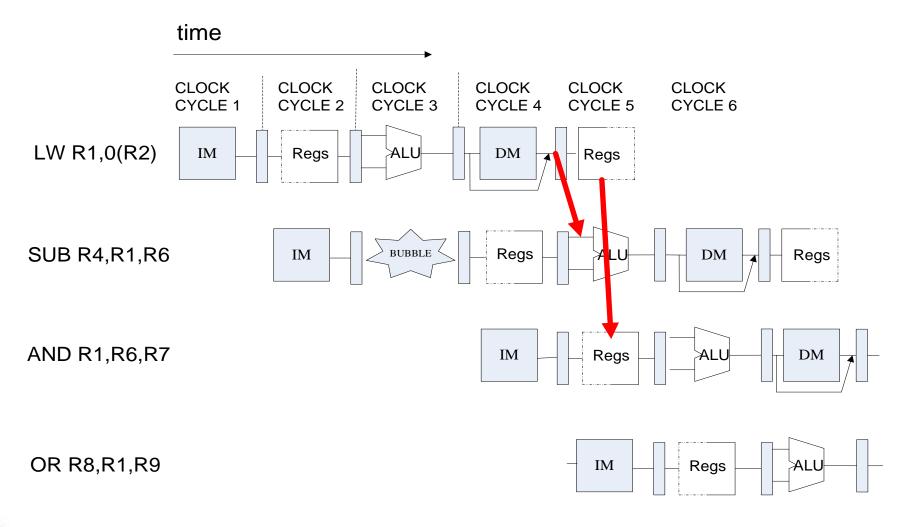
### Condition in Which Bypass Unit doesn't work





### Pipeline Stalls





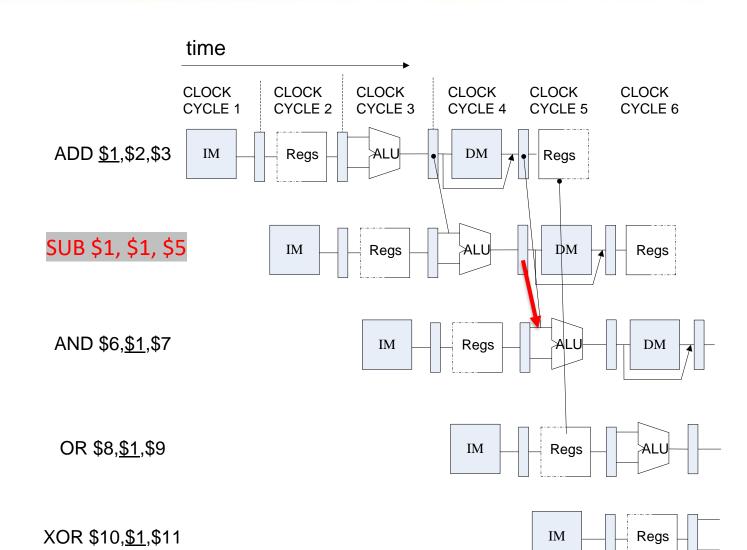


#### **Double Data Hazard**



```
1 add x1, x2, x3
2 add x1, x5, x1 # 采用此x1
3 add x6, x1, x1
```

```
1 lw x1, 0(x6)
2 add x1, x3, x4 # 采用此x1
3 add x5, x1, x1
```





### **Instruction Demo**



	Address Instruction					
		36	Nop			
		40	Add R30,R30,R30			
Branch to 72→		44	Beq R1,R3,24			
		48	And R12,R2,R5			
		52	Or R13,R6,R2			
if R1!=R3, it executes in sequence		56	And R14,R2,R2			
		60				
		64				
		68				
If D1_D2 it avacutes these		72	Lw R14,50(R7)			
If R1=R3, it executes these instruction		76				



#### **Execution result**



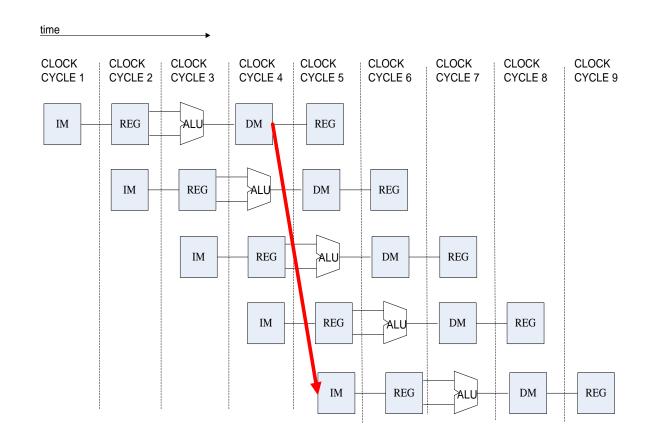
44 BEQ R1,R3,24

48 AND R12,R2,R5

52 OR R13,R6,R2

56 ADD R14,R2,R2

60 OR 72 (DEPENDING ON BRANCH)





# Methods of resolving Control hazards



Freeze or flush the pipeline

Predict-not-taken

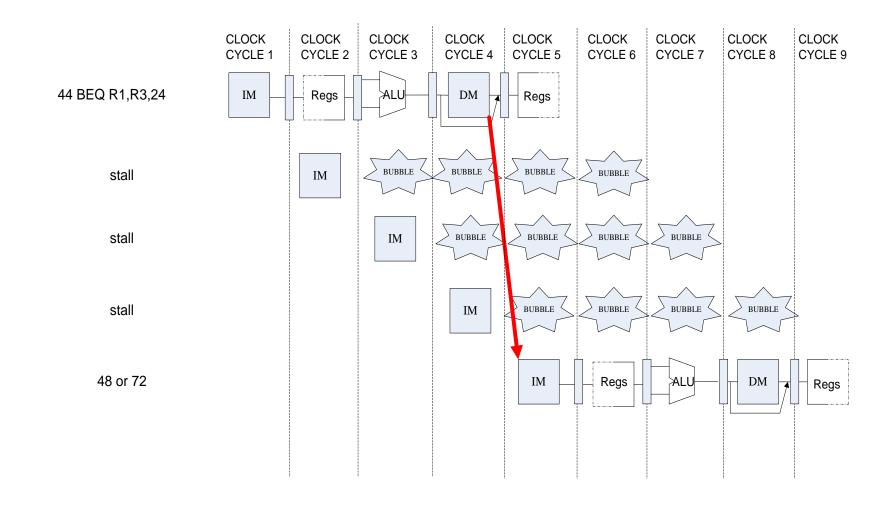
Predict-taken

Delayed branch



#### Freeze method

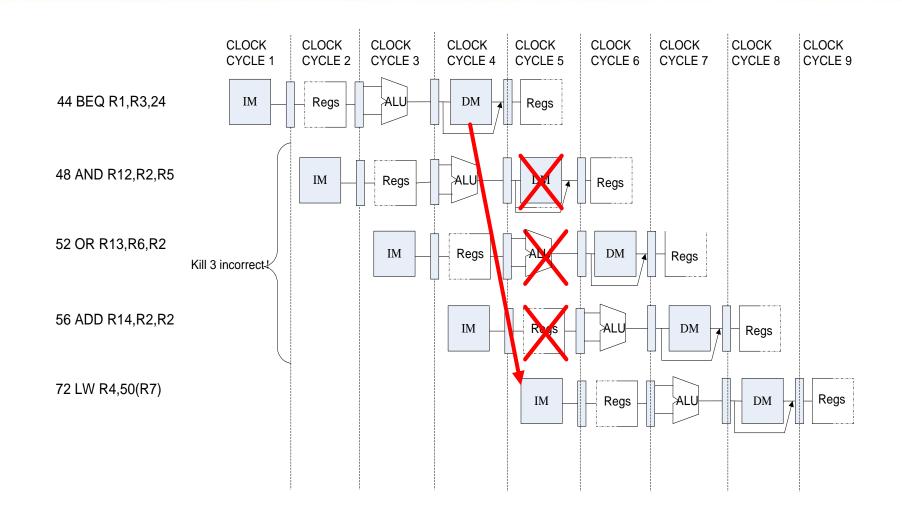














#### Predict-taken method



- 60% of branch result is taken
- Bring forward calculation of branch condition from MEM
   Stage to EX Stage, stall reduce from 3-cycle to 2-cycle.
- Then bring forward from EX to ID, stall reduce from 2-cycle to 1-cycle.
- 1-cycle stall may be used for 1 delay slot



#### Stalls of Predict Methods for Control Hazard

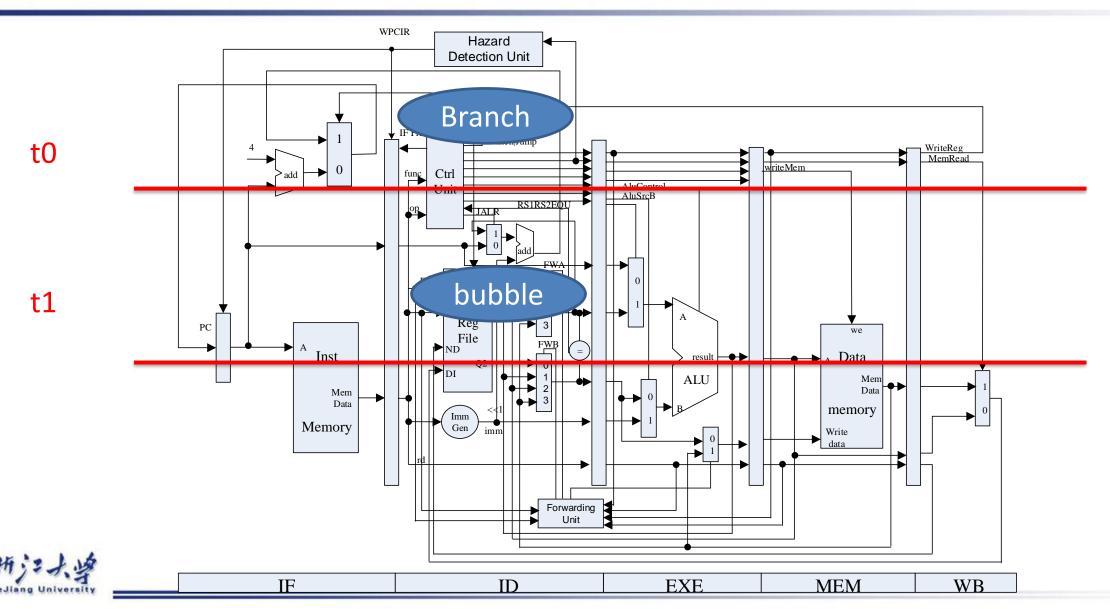


Predict Methods	Status	Original Datapath	Address Calculation Forward	Condition Comparison Forward
Predict-taken	Hit (Need Branch)	3 stall	1 stall	1 stall
Predict-taken	Miss (No Branch)	3 stall	3 stall	1 stall
Predict-not-taken	Hit (No Branch)	go on	go on	go on
Predict-not-taken	Miss (Need Branch)	3 stall	3 stall	1 stall



#### **Control Hazard**







#### Data Hazard & Control Hazard in the meantime

#### RB

```
1 add x1, x2, x3
2 beq x1, x2, label_1
```

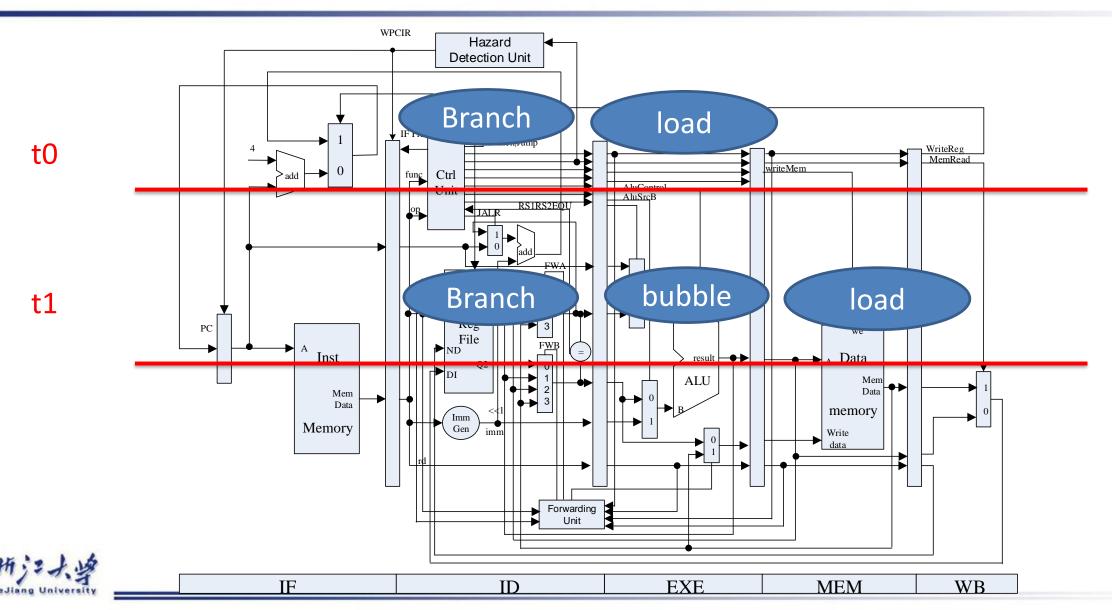
#### LB

```
1 lw x1, 0(x6)
2 jal x1, 12
```



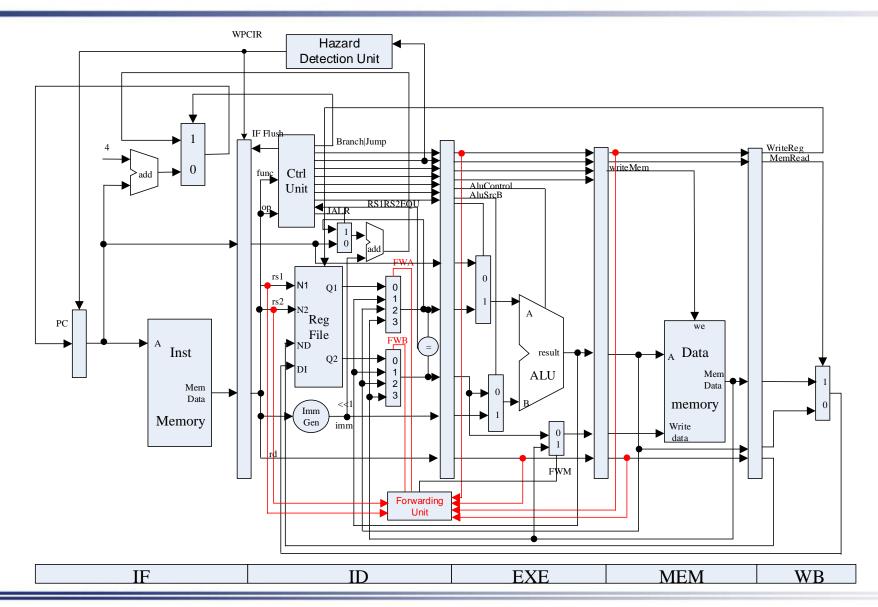


#### Data Hazard & Control Hazard in the meantime



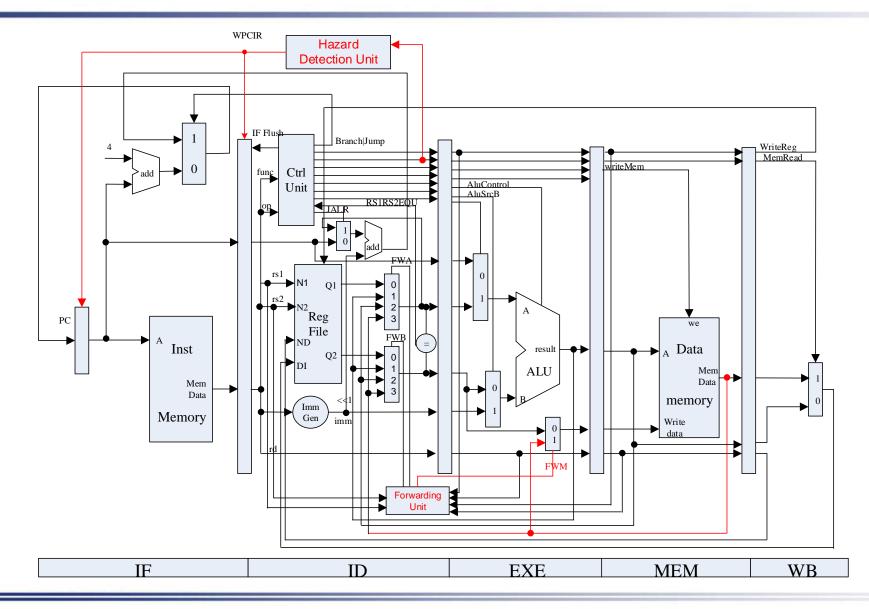


### Pipelined CPU supporting RV32I instructions





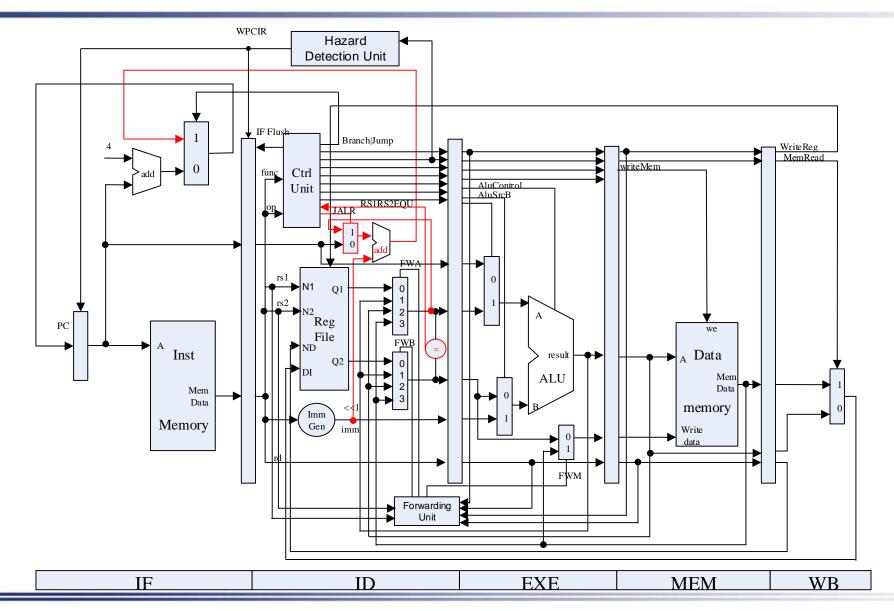
### Pipelined CPU supporting RV32I instructions







### Pipelined CPU supporting RV32I instructions





# Instr. Mem.(1)

NO.	Instruction	Addr.	Label	ASM	Comment
0	0000013	0	start:	addi x0, x0, 0	
1	00402103	4		lw x2, 4(x0)	
2	00802203	8		lw x4, 8(x0)	
3	004100b3	С		add x1, x2, x4	
4	fff08093	10		addi x1, x1, -1	
5	00c02283	14		lw x5, 12(x0)	
6	01002303	18		lw x6, 16(x0)	
7	01402383	1C		lw x7, 20(x0)	
8	402200b3	20		sub x1,x4,x2	
9	002270b3	24		and x1,x4,x2	
10	002260b3	28		or x1,x4,x2	
11	002240b3	2C		xor x1,x4,x2	
12	002210b3	30		sll x1,x4,x2	
13	002220b3	34		slt x1,x4,x2	
14	004120b3	38		slt x1,x2,x4	





# Instr. Mem.(2)

NO.	Instruction	Addr.	Label	ASM	Comment
15	002350b3	3C		srl x1, x6, x2	
16	402350b3	40		sra x1, x6, x2	
17	4023d0b3	44		sra x1, x7, x2	
18	007330b3	48		sltu x1, x6, x7	
19	0063b0b3	4C		sltu x1, x7, x6	
20	00000033	50		add x0,x0,x0	
21	ffd50093	54		addi x1,x10,-3	
22	00f27093	58		andi x1,x4,15	
23	00f26093	5C		ori x1,x4,15	
24	00f24093	60		xori x1,x4,15	
25	00f22093	64		slti x1,x4,15	
26	00121093	68		slli x1,x4,1	
27	00225093	6C		srli x1,x4,2	
28	40c35093	70		srai x1, x6, 12	
29	fff33093	74		sltiu x1, x6, -1	





# Instr. Mem.(3)

NO.	Instruction	Addr.	Label	ASM	Comment
30	fff3b093	78		beq x4,x5,label0	
31	00520863	7C		beq x4,x4,label0	
32	00420663	80		addi x0,x0,0	
33	0000013	84		addi x0,x0,0	
34	0000013	88	label0:	bne x4,x4,label1	
35	00421863	8C		bne x4,x5,label1	
36	00521663	90		addi x0,x0,0	
37	0000013	94		addi x0,x0,0	
38	0000013	98	label1:	blt x5,x4,label2	
39	0042c863	9C		blt x4,x5,label2	
40	00524663	A0		addi x0,x0,0	
41	0000013	A4		addi x0,x0,0	
42	0000013	A8	label2:	bltu x6,x7,label3	
43	00736863	AC		bltu x7,x6,label3	
44	0063e663	В0		addi x0,x0,0	





# Instr. Mem.(4)

NO.	Instruction	Addr.	Label	ASM	Comment
45	0000013	B4		addi x0,x0,0	
46	0000013	B8	label3:	bge x4,x5,label4	
47	00525863	ВС		bge x5,x4,label4	
48	0042d663	CO		addi x0,x0,0	
49	0000013	C4		addi x0,x0,0	
50	0000013	C8	label4:	bgeu x7,x6,label5	
51	0063f863	CC		bgeu x6,x7,label5	
52	00737663	D0		addi x0,x0,0	
53	0000013	D4		addi x0,x0,0	
54	0000013	D8	label5:	bge x4,x4,label6	
55	00425663	DC		addi x0,x0,0	
56	0000013	EO		addi x0,x0,0	
57	0000013	E4	label6:	lui x1,4	
58	000040b7	E8		jal x1,12	
59	00c000ef	EC		addi x0,x0,0	





# Instr. Mem.(5)

NO.	Instruction	Addr.	Label	ASM	Comment
60	0000013	FO		addi x0,x0,0	
61	0000013	F4		lw x8, 24(x0)	
62	01802403	F8		sw x8, 28(x0)	
63	00802e23	FC		lw x1, 28(x0)	
64	01c02083	100		sh x8, 32(x0)	
65	02801023	104		lw x1, 32(x0)	
66	02002083	108		sb x8, 36(x0)	
67	02800223	10C		lw x1, 36(x0)	
68	02402083	110		lh x1, 26(x0)	
69	01a01083	114		lhu x1, 26(x0)	
70	01a05083	118		lb x1, 27(x0)	
71	01b00083	11C		lbu x1, 27(x0)	
72	01b04083	120		auipc x1, 0xffff0	
73	ffff0097	124		jalr x1,0(x0)	
74	000000e7	128			





### Data Mem.

NO.	Data	Addr.	Comment	NO.	Instruction	Addr.	Comment
0	000080BF	0		16	00000000	40	
1	8000000	4		17	00000000	44	
2	0000010	8		18	00000000	48	
3	0000014	С		19	00000000	4C	
4	FFFF0000	10		20	A3000000	50	
5	0FFF0000	14		21	2700000	54	
6	FF000F0F	18		22	79000000	58	
7	F0F0F0F0	1C		23	15100000	5C	
8	0000000	20		24	00000000	60	
9	0000000	24		25	00000000	64	
10	0000000	28		26	00000000	68	
11	0000000	2C		27	00000000	6C	
12	0000000	30		28	00000000	70	
13	00000000	34		29	00000000	74	
14	00000000	38		30	00000000	78	
15	0000000	3C		31	00000000	7C	

### Test Bench



```
RV32core core(
    .debug_en(1'b0),
    .debug_step(1'b0),
    .debug_addr(7'b0),
    .debug_data(),
    .clk(clk),
    .rst(rst),
    .interrupter(1'b0)
 initial begin
   clk = 0;
    rst = 1;
   #2 rst = 0;
 end
 always #1 clk = ^{\sim}clk;
```



# Simulation (1)



П	Name	Value	0 ns	2 ns	4 ns	6 ns	8 ns		10 ns	12	ns	14 ns	16 <del>.</del>	ıs	18 ns		0 ns	2	2 ns
	¼ dk	0																	
2	> <b>₩</b> PC_IF[31:0]	00000000	00000000	0000	0004 000	80000	00000000	X	000000	010	000	00014	00000018	000	0001c	0000002	20	000000	24 00
2	> <b>₩</b> inst_IF[31:0]	00000013	00000013	0040	2103 008	02203	004100Ъ3	X	fff080	093	000	02283	01002303	014	02383	4022001	ьз 📉	002270	ъз 00
	> ₩ PC_ID[31:0]	00000000		00000000	000	00004	00000008	X	000000	00c	000	00010	00000014	000	00018	000000	1c	000000	20 00
	> <b>₩</b> inst_ID[31:0]	00000000	00000000	0000	0013 004	02103	00802203	X	004100	0Ъ3	fff	08093	00c02283	010	02303	0140238	83	402200	ьз 💢 оо
	> ₩ PC_EXE[31:0]	00000000		00000000		X	00000004	00000	8000		0000000c	X	00000010	000	00014	000000	18	000000	1c 00
	> <b>₩</b> inst_EXE[31:0]	00000000		00000000	000	00013	00402103	00802	2203	00000000	004	100ъ3	fff08093	000	02283	0100230	03	014023	83 40
	▶ ₩ PC_MEM[31:0]	00000000			00000000			00000	0004	00000008	X	0000000	,	000	00010	000000	14	000000	18 00
	> ₩ inst_MEM[31:0]	00000000		00000000		X	00000013	00402	2103	00802203	000	00000	004100Ъ3	fff	08093	0000228	83	010023	03 01
	> <b>₩</b> PC_WB[31:0]	00000000			00000000	)			X	00000004	000	00008		0000000c		000000	10	000000	14 00
	> <b>₩</b> inst_WB[31:0]	00000000			00000000			00000	0013	00402103	008	02203	00000000	004	100Ъ3	fff0809	93	00c022	83 01
	Branch_ctrl	0																	
	I JALR	0																	
	RegWrite_ctrl	0																	
	¼ mem_w_ctrl	0																	
	MIO_ctrl	0																	
	ALUSrc_A_ctrl	0																	
	ALUSrc_B_ctrl	0																	
	DatatoReg_ctrl	0																	
	l rs1use_ctrl	0																	
	ors2use_ctrl	0																	
	> ₩ hazard_orl[1:0	0	0	X	X	2		X		1		X			2		=		1
2	> <b>₩</b> ImmSel_ctrl[2:0]	0	0	X		1		X	0		X			1			=X		0
	> ₩ cmp_ctrl[2:0]	0								0									
	> W ALUContrl[3:0]	0	0	X						1							=	2	Х 3
314	> ♥ forwardA[1:0]	0			0			χ :	X	0	X	1				0			
ie.li	> ♥ forwardB[1:0]	0			0				X	3	X				0				
H								χ :	X		X	1 /			0	0			

# Simulation (2)



Name	Value	20 55	22 ns	24 ns	126 pe	128 25	130	30 pe	34 ns	36 pe	138 55	M0 ne	lao
<sup>1</sup> ₄ dk	0	20 ns	22 115	24 115	26 ns	28 ns	30 ns	32 ns	J4 115	36 ns	38 ns	40 113	42
> W PC_IF[31:0]	00000000	00000020	00000024	00000028	X 0000002c	00000030	00000034	00000038	00000036	00000040	00000044	00000048	0000004c
> <b>W</b> inst_IF[31:0]	00000013	402200Ъ3	002270Ь3	002260ъ3	002240ъ3	002210Ь3	002220ь3	004120ъ3	002350ъ3	402350Ъ3	402340ъ3	007330Ъ3	00635053
> <b>W</b> PC_ID[31:0]	00000000	0000001c	00000020	00000024	00000028	0000002c	00000030	00000034	00000038	0000003c	00000040	00000044	00000048
> <b>II</b> inst_ID[31:0]	00000000	01402383	402200Ъ3	002270ъ3	002260Ъ3	002240Ъ3	002210Ь3	002220Ъ3	004120Ъ3	002350ъ3	402350ъ3	4023d0ъ3	007330Ъ3
> <b>W</b> PC_EXE[31:0]	00000000	00000018	0000001c	00000020	00000024	00000028	00000020	00000030	00000034	00000038	00000030	00000040	00000044
> <b>W</b> inst_EXE[31:0]	00000000	01002303	01402383	402200Ъ3	002270Ъ3	002260Ъ3	002240Ь3	002210Ъ3	002220ъ3	004120Ъ3	002350ъ3	402350ъ3	4023d0ь3
> W PC_MEM[31:0]	00000000	00000014	00000018	0000001c	00000020	00000024	00000028	0000002c	00000030	00000034	00000038	0000003c	00000040
> <b>W</b> inst_MEM[31:0]	00000000	00c02283	01002303	01402383	402200Ъ3	002270Ъ3	002260Ъ3	002240Ъ3	002210Ъ3	002220ъ3	004120Ъ3	002350ъ3	402350ъ3
> <b>W</b> PC_WB[31:0]	00000000	00000010	00000014	00000018	00000010	00000020	00000024	00000028	00000020	00000030	00000034	00000038	0000003e
> <b>W</b> inst_WB[31:0]	00000000	fff08093	00e02283	01002303	01402383	402200Ъ3	002270ъ3	002260Ъ3	002240ъ3	002210Ъ3	002220ъ3	004120ъ3	002350ъ3
Branch_ctrl     Branc	0												
I JALR	0												
RegWrite_ctrl	0												
la mem_w_ctrl	0												
MIO_ctrl	0												
ALUSrc_A_ctrl	0												
ALUSrc_B_ctrl	0												
	0												
⅓ rs1use_ctrl	0												
lars2use_ctrl	0												
> W hazard_orl[1:0	0	2	χ '					1					
> W ImmSel_ctrl[2:0	0	1	χ					0					
> <b>W</b> cmp_ctrl[2:0]	0							)					
> W ALUContrl[3:0	0	1	2	3	4	X 5	6	χ	8	7	χ	a	9
> <b>W</b> forwardA[1:0]	0							)					
> 😻 forwardB[1:0]	0							1					

# Simulation (3)



Name	Value	40	ns ,	42 ns	44	ns ,	46 ns	48	ns ,	50 ns	52	ns ,  5	4 ns	56 ns	58 ns	60 ns	62
¼ dk	0																
> <b>W</b> PC_IF[31:0]	00000000	0000004	8	0000004c	00000050		0000054	00000058	X 00	00005c	00000060	000000	64	00000068	0000006c	00000070	00000074
> <b>W</b> inst_IF[31:0]	00000013	007330Ъ	3	0063Ь0Ь3	00000033		f450093	00f27093	00	f26093	00f24093	00f220	93	00121093	00225093	40c35093	fff33093
> <b>W</b> PC_ID[31:0]	00000000	0000004	4	00000048	0000004c		0000050	00000054	00	000058	0000005c	000000	60	00000064	00000068	0000006c	00000070
▶ Inst_ID[31:0]	00000000	4023d0b	3	007330Ъ3	00635053		0000033	ffd50093	00	27093	00f26093	00f240	93	00f22093	00121093	00225093	40035093
₩ PC_EXE[31:0]	00000000	0000004	0	00000044	00000048	X	000004c	00000050	00	000054	00000058	000000	5c X	00000060	00000064	00000068	00000066
→ <b>W</b> inst_EXE[31:0]	00000000	402350Ъ	3	4023d0b3	007330Ъ3		063Ь0Ь3	00000033	ff	150093	00f27093	00f260	93	00f24093	00f22093	00121093	00225093
₩ PC_MEM[31:0]	00000000	0000003	ic X	00000040	00000044	χ (	0000048	0000004	00	000050	00000054	000000	58	0000005c	00000060	00000064	00000068
	00000000	002350Ъ	3	402350Ъ3	4023d0b3	X	07330Ъ3	00635053	00	000033	ffd50093	00f270	93	00f26093	00f24093	00f22093	00121093
> ₩ PC_WB[31:0]	00000000	0000003	8	0000003c	00000040	X (	0000044	00000048	00	00004c	00000050	000000	54	00000058	00000050	00000060	00000064
▶ W inst_WB[31:0]	00000000	004120Ъ	3	002350Ъ3	402350ъ3	4	02340Ъ3	007330Ъ3	00	31-01-3	00000033	ffd500	93	00f27093	00f26093	00f24093	00f22093
Branch_ctrl     Branc	0																
1⊌ JALR	0																
1⊌ RegWrite_ctrl	0																
¼ mem_w_ctrl	0																
MIO_ctrl	0																
la ALUSrc_A_ctrl	0																
ALUSrc_B_ctrl	0																
DatatoReg_ctrl     Data	0									0							
l rs1use_ctrl	0									T							
¹⊌ rs2use_ctrl	0																
₩ hazard_orl[1:0	0																
₩ ImmSel_ctrl[2:0	0				0			χ					1				
cmp_ctrl[2:0]	0																
M ALUContrl[3:0	0	9.	X		9	$-\chi$		1	X	3	X 4	5	$\overline{}$	8	X 6	7	\ a
■ forwardA[1:0]	0																
	0																

# Simulation (4)



Name	Value		60 ns		62 ns	64	4 ns	6	66 ns	لىىيا	68 ns	لىىيا	70 ns		72 ns		74 ns		76 ns	سبيا	78 ns		80 ns	82
¼ dk	0																							
> • PC_IF[31:0]	00000000	00000	070	00000	074	000000	78	000000	)7c	00000	080	00000	084	00000	08c	0000	0090	00000	0094	0000	009c	0000	00a0	000000a4
> <b>W</b> inst_IF[31:0]	00000013	40c35	093	fff33	093	fff3b0	93	005208	863	00420	663	00000	013	00421	863	0052	1663	00000	013	0042	2c863	00524	4663	00000013
> W PC_ID[31:0]	00000000	00000	06c	00000	070	000000	74	000000	78	00000	07c	$\langle \hspace{0.2cm} \rangle$	00000	080		0000	008c	X	00000	0090		0000	009c	000000a0
> <b>W</b> inst_ID[31:0]	00000000	00225	093	40035	093	fff330	93	fff3b0	93	00520	863	00420	663	00000	013	0042	1863	00521	1663	0000	00013	0042	863	0052466
> ₩ PC_EXE[31:0]	00000000	00000	068	00000	06c	000000	70	000000	74	00000	078	00000	07c		00000	080		00000	008c		0000	0090		0000009
> <b>W</b> inst_EXE[31:0]	00000000	00121	093	00225	093	400350	93	fff330	93	fff3b	093	00520	863	00420	663	0000	0013	00421	1863	0052	1663	0000	0013	0042086
▶ ₩ PC_MEM[31:0]	00000000	00000	064	00000	068	000000	6c	000000	70	00000	074	00000	078	00000	07c	( <u> </u>	0000	080		0000	0008c	X	000000	190
▶ Inst_MEM[31:0]	00000000	00f22	093	00121	093	002250	93	400350	93	fff33	093	fff3b	093	00520	863	0042	0663	00000	0013	0042	1863	0052	1663	0000001
> W PC_WB[31:0]	00000000	00000	060	00000	064	000000	68	000000	)6c	00000	070	00000	074	00000	078	0000	007c	X	00000	080		0000	008c	0000009
▶ Inst_WB[31:0]	00000000	00f24	093	00f22	093	001210	93	002250	193	40035	093	fff33	093	fff3b	093	0052	0863	00420	1663	0000	00013	0042	1863	0052166
Branch_ctrl     Branc	0																							
¹⊌ JALR	0																							
RegWrite_ctrl	0																					1		
le mem_w_ctrl	0									0														
MIO_ctrl	0																							
la ALUSrc_A_ctrl	0																							
ALUSrc_B_ctrl	0															1						1		
DatatoReg_ctrl	0																							
⅓ rs1use_ctrl	0																							
¹⊌ rs2use_ctrl	0																			1				
₩ hazard_orl[1:0	0				1							0		1		<u></u>		0		$\overline{}$	1	<u> </u>	0	
₩ ImmSel_ctrl[2:0	0				1					$\overline{}$		2		$\equiv$		$\geq$		2		$\equiv$	1	$\widehat{\mathbf{x}}$	2	
cmp_ctrl[2:0]	0				0				==	$\vdash$			==	$\equiv$	==	$\geq$		2	==	$\geq$	0	ý—	3	
₩ ALUContrl[3:0	0	7		•	$\overline{}$		9			$\overline{}$				Ť		$\geq$		0		$\geq$	1	Ŷ <u> </u>	0	
♥ forwardA[1:0]										$\overline{}$				÷								^	· ·	
forwardB[1:0]	0												0											

# Simulation (5)



Name	Value	80 ns	82 ns	84 ns	86 ns		90 ns	92 ns	94 ns	96 ns	98 ns	100 ns	102
¼ dk	0												
> W PC_IF[31:0]	00000000	000000a0	000000⊾4	00000040	000000ь0	000000ъ4	000000Ъс	00000000	000000c4	00000000	00000040	00000044	00000040
> 😻 inst_IF[31:0]	00000013	00524663	00000013	00736863	0063e663	00000013	00525863	00424663	00000013	0063f863	00737663	00000013	00425663
> W PC_ID[31:0]	00000000	0000009c	0000	00a0	000000ac	0000	0050	000000ъс	0000	000c0	000000cc	00000	0040
> <b>W</b> inst_ID[31:0]	00000000	0042c863	00524663	00000013	00736863	0063e663	00000013	00525863	00424663	00000013	0063f863	00737663	00000013
> ₩ PC_EXE[31:0]	00000000	00000090	00000090	0000	00a0	000000ac	0000	0050	000000Ъс	0000	0000	00000000	00000040
> <b>W</b> inst_EXE[31:0]	00000000	00000013	00420863	00524663	00000013	00736863	0063e663	00000013	00525863	00424663	00000013	0063f863	00737663
> ₩ PC_MEM[31:0]	00000000	00000	0090	0000009c	00000	00a0	000000ac	0000	0090	000000ъс	0000	100c0	000000cc
> <b>W</b> inst_MEM[31:0]	00000000	00521663	00000013	00420863	00524663	00000013	00736863	0063e663	00000013	00525863	00424663	00000013	0063f863
> W PC_WB[31:0]	00000000	00000080	0000	0090	00000090	0000	00a0	00000040	0000	000Ъ0	000000Ъс	00000	J0c0
> <b>W</b> inst_WB[31:0]	00000000	00421863	00521663	00000013	0042c863	00524663	00000013	00736863	0063e663	00000013	00525863	00424663	00000013
Branch_ctrl     Branc	0												
1₫ JALR	0												
¹₀ RegWrite_ctrl	0												
le mem_w_ctrl	0												
MIO_ctrl	0												
ALUSrc_A_ctrl	0												
ALUSrc_B_ctrl	0												
	0												
¹⊌ rs1use_ctrl	0												
¹₀ rs2use_ctrl	0												
> W hazard_orl[1:0	0		)	X i	X	0	X	X	0	X i	X	0	1
> <b>W</b> ImmSel_ctrl[2:0]	0		2	X i	Χ	2	X 1	X	2	X 1	X	2	1
> <b>W</b> cmp_ctrl[2:0]	0		3	X 0	Χ	4	X 0	X	5	X 0	X	6	Χ ο
> W ALUContrl[3:0]	0			X 1	X	0	\i	X	0	X 1	XX	0	1
> <b>W</b> forwardA[1:0]	0						0						
> 😽 forwardB[1:0]	0						0						

# Simulation (6)



Name	Value	1,,,,,,  100,	ns. I	102 ns	104 ns		106 ns	108 ns		0 ns. 1	112 ns	114 r	s.   116 ns	118 ns	120 ns	
<b>¼</b> dk	0															
> W PC_IF[31:0]	00000000	000000d4	0000	000dc	000000e0	00000	0e8	000000ec	00000010	o X	000000f8	000000fc	00000100	00000104	00000108	00000100
> <b>W</b> inst_IF[31:0]	00000013	00000013	004	25663	00000013	00004	0ъ7	00c000ef	00000013	3 \	01802403	00802e23	01c02083	02801023	02002083	02800223
> W PC_ID[31:0]	00000000	0	0000040		0000	00dc	$\overline{}$	000000e8	$\overline{}$	000000	ec	000000f8	000000fc	00000100	00000104	00000108
> W inst_ID[31:0]	00000000	00737663	0000	00013	00425663	00000	013	000040ъ7	00c000ef	f	00000013	01802403	00802e23	01c02083	02801023	02002083
> W PC_EXE[31:0]	00000000	00000000	_X	00000	0040	X	00000	0 dc	000000e8	8	0000	00ec	000000f8	000000fc	00000100	00000104
> <b>W</b> inst_EXE[31:0]	00000000	0063f863	007:	37663	00000013	00425	663	00000013	000040Ъ7	7 X	00c000ef	00000013	01802403	00802e23	01c02083	02801023
> W PC_MEM[31:0]	00000000	000000c0	000	000cc	0000	00d0	$\overline{}$	00000	0de	$\overline{}$	000000e8	0	0000ec	000000f8	000000fc	00000100
> <b>W</b> inst_MEM[31:0]	00000000	00000013	006	3f863	00737663	00000	013	00425663	00000013	3	000040Ъ7	00c000ef	00000013	01802403	00802e23	01c02083
> W PC_WB[31:0]	00000000		0000000		00000000	X	00000	040		000000	do	000000e8	000	)00ec	000000f8	000000fc
> <b>W</b> inst_WB[31:0]	00000000	00424663	000	00013	0063f863	00737	663	00000013	00425663	3	00000013	000040ъ7	00c000ef	00000013	01802403	00802e23
Branch_ctrl	0															
1⊌ JALR	0															
RegWrite_ctrl	0															
le mem_w_ctrl	0															
MIO_ctrl	0															
ALUSrc_A_ctrl	0															
ALUSrc_B_ctrl	0															
DatatoReg_ctrl     Data	0															
l₀ rs1use_ctrl	0															
¹⊌ rs2use_ctrl	0								0							
> <b>W</b> hazard_orl[1:0	0	0	$\sim$	1	0	X						2	3	2	3	2
> W ImmSel_ctrl[2:0	0	2	$\sim$	1	2	X 1		5	3	$=\chi$		1	4	X 1	4	1
> <b>W</b> cmp_ctrl[2:0]	0	6	$\sim$	0	5	X						0				
> W ALUContrl[3:0	0	0	$\sim$	1	0	X 1		•	ь	$-\chi$				1		
> 🐶 forwardA[1:0]	0									0						
he-> 😻 forwardB[1:0]	0									0						

# Simulation (7)



Name	Value	1	120 ns		122 ns		124 ns	1	26 ns	1:	28 ns	1	130 ns	13	2 ns	134	ns	136 ns		138 ns	14	10 ns. 1	142
¼ clk	0																						
> W PC_IF[31:0]	00000000	00000	108	00000	10c	00000	110	000001	14	000001	18	00000	11c	0000012	0	00000124	00	000128	000	0012c	0000000	00	00000004
> <b>W</b> inst_IF[31:0]	00000013	020020	083	02800	223	02402	083	01a010	83	01a050	83	01500	083	01Ъ0408	3	ffff0097	00	0000e7	иии	хххх	0000001	13	00402103
> W PC_ID[31:0]	00000000	00000	104	00000	108	00000	10c	000001	10	000001	14	00000	118	0000011	c X	00000120	00	000124	X	0000	0128	$\square$ X	00000000
> <b>W</b> inst_ID[31:0]	00000000	02801	023	02002	083	02800	223	024020	83	01a010	83	01a05	083	01Ъ0008	3	01Ъ04083	fi	ff0097	000	000e7	X	000000	13
> <b>W</b> PC_EXE[31:0]	00000000	00000	100	00000	104	00000	108	000001	0c X	000001	10	00000	114	0000011	8	00000110	00	000120	000	00124	X	000001	28
> <b>W</b> inst_EXE[31:0]	00000000	01c02	083	02801	023	02002	083	028002	23	024020	83	01a01	083	01a0508	3	01Ъ00083	01	Ъ04083	fff	f0097	000000	e7 X	00000013
> <b>W</b> PC_MEM[31:0]	00000000	000000	0fc	00000	100	00000	104	000001	08	000001	Oc X	00000	110	0000011	4	00000118	00	00011c	000	00120	0000012	24	00000128
> <b>W</b> inst_MEM[31:0]	00000000	00802	e23	01002	083	02801	023	020020	83	028002	23	02402	083	01a0108	3	01a05083	01	ъ00083	01Ъ	04083	ffff009	97	000000e7
> <b>W</b> PC_WB[31:0]	00000000	000000	0f8	00000	Ofc	00000	100	000001	04	000001	08 X	00000	10c	0000011	0 X	00000114	00	000118	000	0011c	0000012	20	00000124
> <b>W</b> inst_WB[31:0]	00000000	018024	403	00802	e23	01002	083	028010	23	020020	83	02800	223	0240208	3	01a01083	01	a05083	01Ъ	00083	0150408	83	ffff0097
Branch_ctrl	0																						
1₫ JALR	0																						
RegWrite_ctrl	0																						
mem_w_ctrl	0																						
MIO_ctrl	0																						
ALUSrc_A_ctrl	0																				Щ		
ALUSrc_B_ctrl	0																						
DatatoReg_ctrl     Data	0																						
ars1use_ctrl	0																	$\perp$					
I rs2use_ctrl	0																						
hazard_orl[1:0	0	3	X	2		3		$\overline{}$				2					X			1			
→ W ImmSel_ctrl[2:0	0	4	X	1		X 4		$\overline{}$				1					_X_	5	X		1		
> ₩ cmp_ctrl[2:0]	0												0										
> W ALUContrl[3:0]	0									1									X	Ъ	X	1	
> <b>V</b> forwardA[1:0]	0												0										
> 😻 forwardB[1:0]	0												0										

# Checkpoints



#### • CP 1:

Waveform Simulation of the Pipelined CPU with the verification program

#### • CP 2:

FPGA Implementation of the Pipelined CPU with the verification program





# Thanksl

