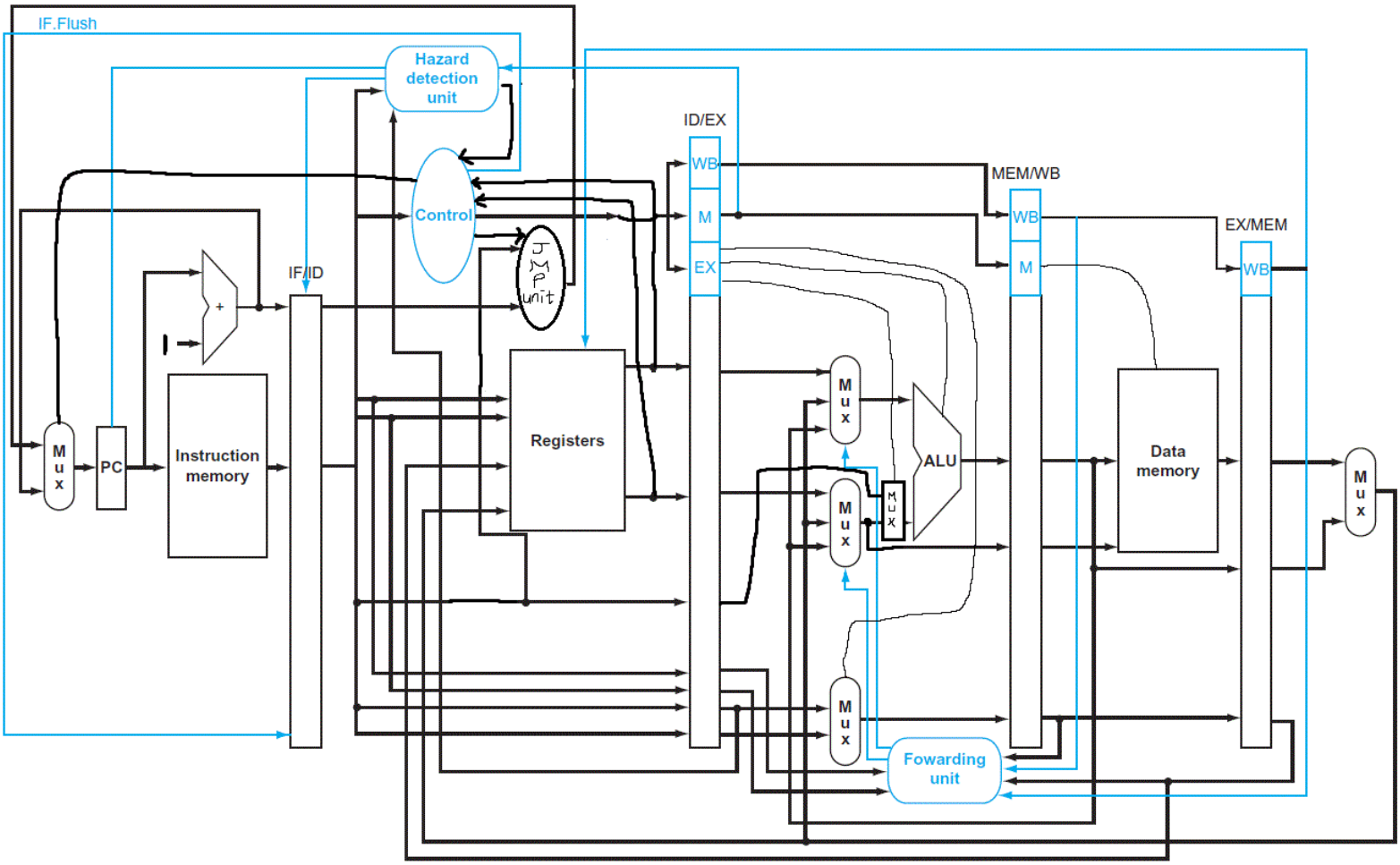
計算機組織

期末報告

電機三　吳亭葳

105501539



子module包含

PCmux, PC, INSTRUCTION, PCadder, IF_ID,

HDU, CONTROL, REG, JMPunit, ID_EX,

ALUmux0, ALUmux1, ALUmux2, REGdstmux, ALU, FU, MEM_WB

MEM, EX_MEM, WBmux

參考資料：計算機組織課本(Computer Organization And Design 5th Edition 2014)、上學期DSD上課投影片內容。

分工：獨自完成。

	IFflush	WBregwr	Wb regomem	MEMwr	EXalusrc	EXaluctrl	REGdst	PCsrc	MEMread
Lw	0	1	0	0	1	0	0	0	1
Sw	0	0	1	1	1	0	0	0	0
Add	0	1	1	0	0	0	1	0	0
Addi	0	1	1	0	1	0	1	0	0
Sub	0	1	1	0	0	1	1	0	0
And	0	1	1	0	0	2	1	0	0
or	0	1	1	0	0	3	1	0	0
slt	0	1	1	0	0	4	1	0	0
beq	cond	0	0	0	0	0	0	condi	0
Jmp	1	0	0	0	0	0	0	1	0
nop	0	0	0	0	0	0	0	0	0

control signal:

Instruction:

	CTRL(4)	Reg1(3)	Reg2(3)	Reg3(3)	constant (8)	
Lw	0			X		lw R2,constant(R1)
Sw	1			X		sw R2, constant(R1)
Add	2				x	add R3,R1,R2
Addi	3		x			addi R3,R1,constant
Sub	4				x	sub R3,r2,r1
And	5				x	and R3,R1,R2
or	6				x	Or R3,R1,R2
slt	7				x	slt R3,R1,R2
beq	8			x		beq R1,R2,constant
Jmp	9	x	x	x		jmp constant

nop	10	0	0	0	0	

(R0不使用)

注意：

做測試時要先寫出Instuction code，再將其輸入INSTRUCTION module的初始值中測試。

Pipeline test(without branch):

Addi r5,r5,8 0011_101_000_101_0000_1000

Addi r3,r2,12 0011_010_000_011_0000_1100

Addi r4,r4,16 0011_100_000_100_0001_0000

Lw r6,24(r5) 0000_101_110_000_0001_1000

Slt r1,r5,r3 0111_101_011_001_0000_0000

Sub r7,r3,r5 0100_011_101_111_0000_0000

Sw r4,0(r3) 0001_011_100_000_0000_0000

Reg[5]=8

Reg[3]=12

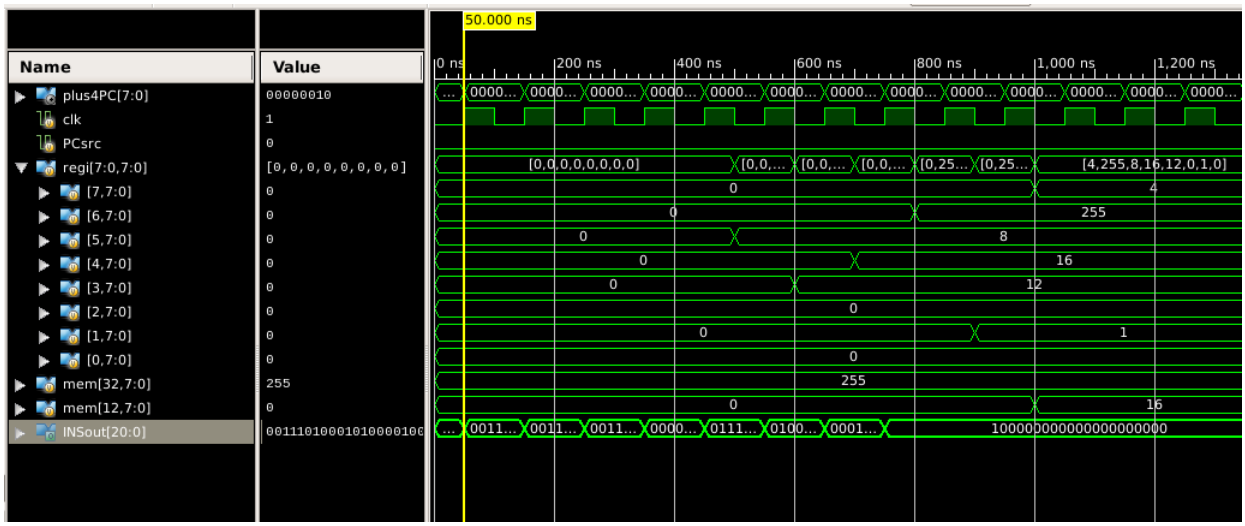
Reg[4]=16

Reg[6]=mem[32]

Reg[1]=1

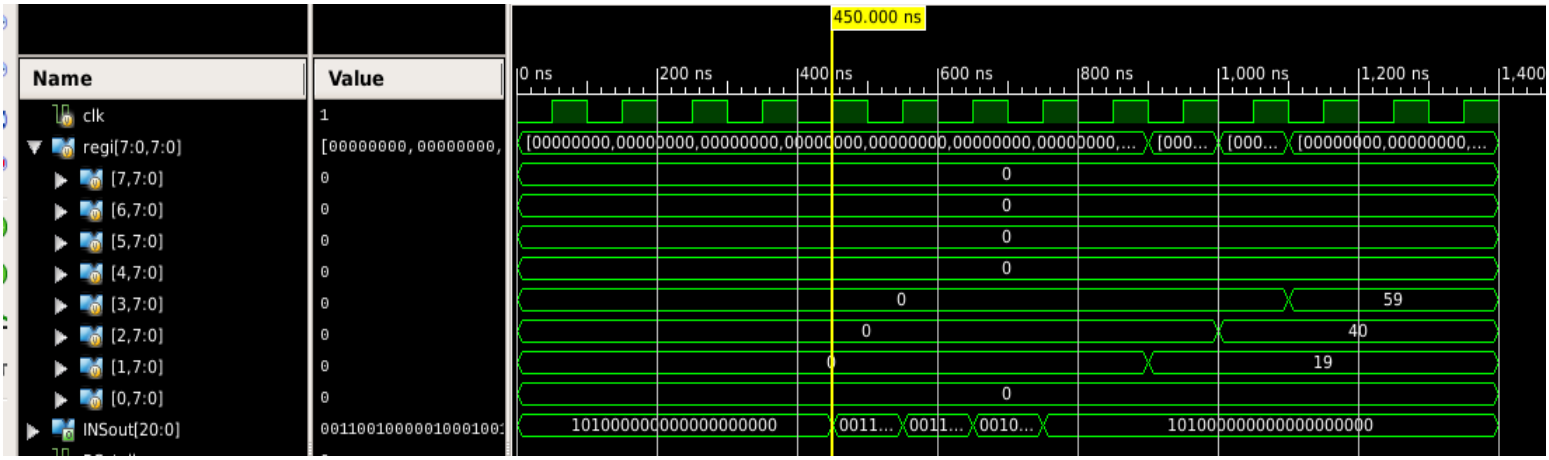
Reg[7]=4

Mem[12]=16



fowarding test:

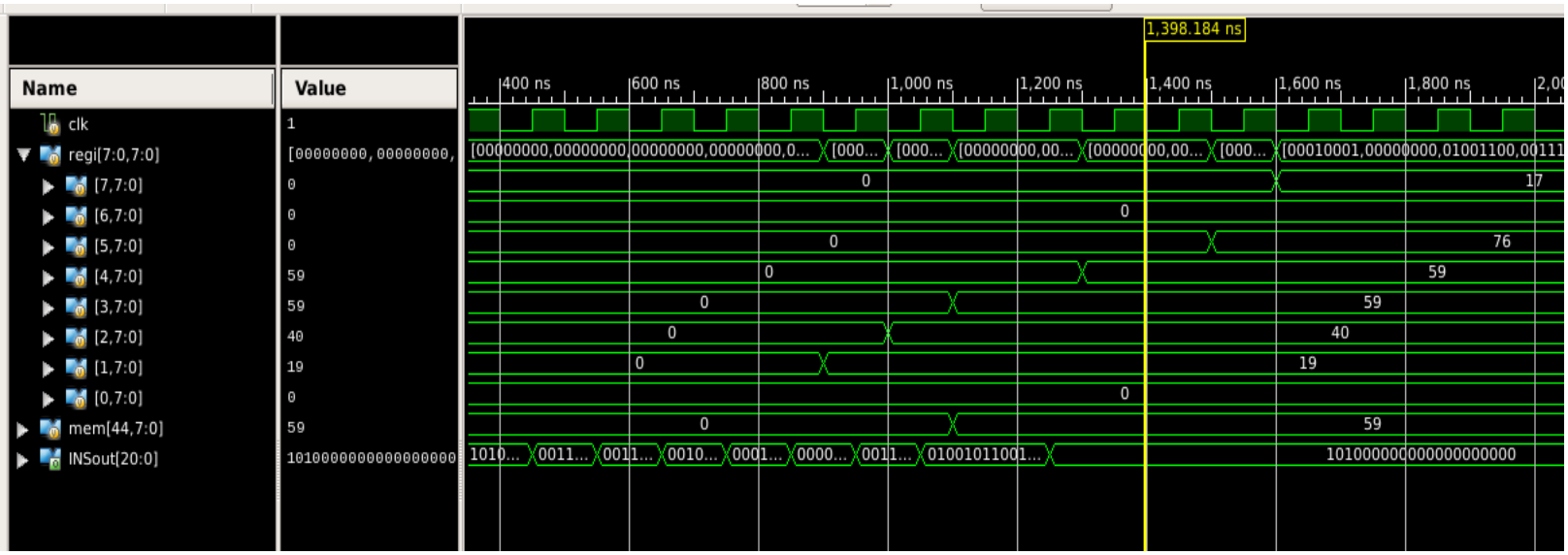
addi r1,r1,19	0011_001_000_001_0001_0011	R1=19
addi r2,r1,21	0011_001_000_010_0001_0101	R2=40
add r3,r2,r1	0010_010_001_011_0000_0000	R3=59



LW forwarding test:

addi r1,r1,19	0011_001_000_001_0001_0011	R1=19
addi r2,r1,21	0011_001_000_010_0001_0101	R2=40
add r3,r2,r1	0010_010_001_011_0000_0000	R3=59
sw r3,4(R2)	0001_010_011_000_0000_0100	mem[44]=59
lw r4,25(R1)	0000_001_100_000_0001_1001	R4=59
addi r5,r4,17	0011_100_000_101_0001_0001	R5=76

sub r7,r5,r4 0100_101_100_111_0000_0000 R7=17



Hazard with branch and jump:

[5]	addi r1,r1,1	0011_001_000_001_0000_0001	R1=1
[6]	addi r2,r1,5	0011_001_000_010_0000_0101	R2=6
[7]	slt r3,r1,r2	0111_001_010_011_0000_0000	R3=1
[8]	sw r2,0(R2)	0001_010_010_000_0000_0000	mem[6]=6
[9]	sub r2,r2,r1	0100_010_001_010_0000_0000	R2=5-1
[10]	beq r1,r3,-4	1000_001_011_000_1111_1101	[7~10] for loop
[11]	addi r2,r2,2	0011_010_000_010_0000_0010	R2=r2+2
[12]	lw r4,3(r2)	0000_010_100_000_0000_0011	R4=mem[r2+3]
[13]	and r5,r4,r2	0101_100_010_101_0000_0000	R5=0
[14]	or r6,r2,r4	0110_010_100_110_0000_0000	R6=0111=7
[15]	jmp 11	1001_000_000_000_0000_1011	jump to [11] infinite loop