

VLSI 系統設計

-期末專題報告

組別：This Time CP UUUUUUUUU		
姓名	學號	授課教師
余雅瑄	E94106054	李昆忠
宋晉誠	E94101119	
游宗謀	E94106151	
褚福為	E94104036	

1. 系統簡介

- 指令集格式

RV32I Base Integer Instruction Set

- 指令集格式欄位的名稱、長度、說明

31	30	25	24	21	20	19	15	14	12	11	8	7	6	0	
funct7				rs2		rs1	funct3		rd			opcode		R-type	
imm[11:0]						rs1	funct3		rd			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			opcode			U-type
imm[20]	imm[10:1]			imm[11]	imm[19:12]				rd			opcode			J-type

a. R -type (Register-Register 操作)

31	25	24	20	19	15	14	12	11	7	6	0
funct7				rs2	rs1	funct3		rd	opcode		
7				5	5	3		5	7		
0000000				src2	src1	ADD/SLT/SLTU		dest	OP		
0000000				src2	src1	AND/OR/XOR		dest	OP		
0000000				src2	src1	SLL/SRL		dest	OP		
0100000				src2	src1	SUB/SRA		dest	OP		

從 rs1 和 rs2 取值，進行操作後存入 rd

b. I 型 (Immediate 操作)

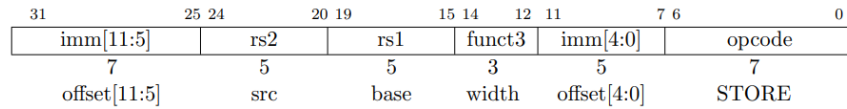
31	20	19	15	14	12	11	7	6	0
imm[11:0]			rs1	funct3		rd	opcode		
12			5	3		5	7		
I-immediate[11:0]			src	ADDI/SLTI[U]		dest	OP-IMM		
I-immediate[11:0]			src	ANDI/ORI/XORI		dest	OP-IMM		

31	25	24	20	19	15	14	12	11	7	6	0
imm[11:5]		imm[4:0]		rs1	funct3		rd	opcode			
7		5		5	3		5	7			
0000000		shamt[4:0]		src	SLLI		dest	OP-IMM			
0000000		shamt[4:0]		src	SRLI		dest	OP-IMM			
0100000		shamt[4:0]		src	SRAI		dest	OP-IMM			

31	20	19	15	14	12	11	7	6	0
imm[11:0]			rs1	funct3		rd	opcode		
12			5	3		5	7		
offset[11:0]			base	width		dest	LOAD		

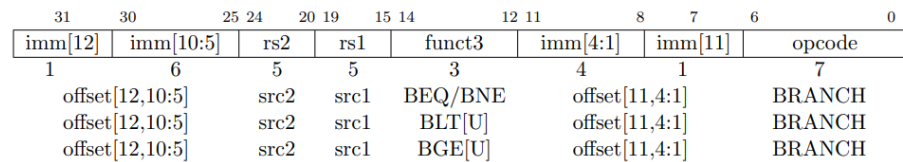
從 rs1 取值，和 imm 進行操作，將結果寫回 rd

c. S 型 (Store 操作)



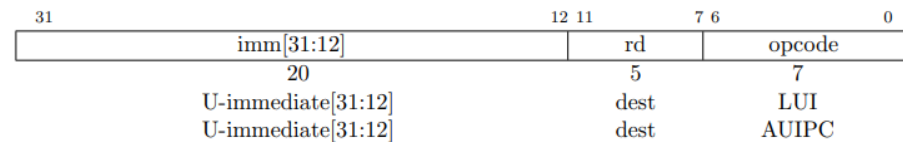
rs1 和 imm 分別為 base 和 offset，將 rs2 的內容儲存進 memory 相對應的位置

d. B 型 (Branch 操作)



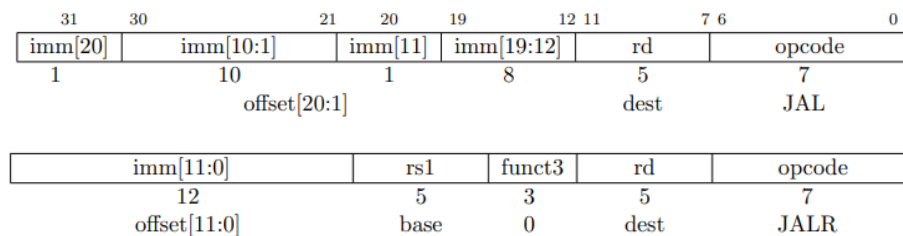
針對 rs1 和 rs2 的值進行判斷，若是判斷為真，則跳到 $pc + imm$ 的位置

e. U 型 (Upper Immediate 操作)



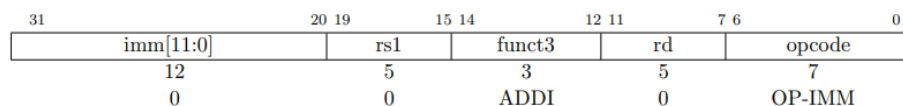
針對 imm 的 upper bits 寫入 rd

f. J 型 (Jump 操作)



跳到 $pc + imm$ 的位置，並且把要跳回來的位置 $pc+4$ 存進 rd

g. NOP

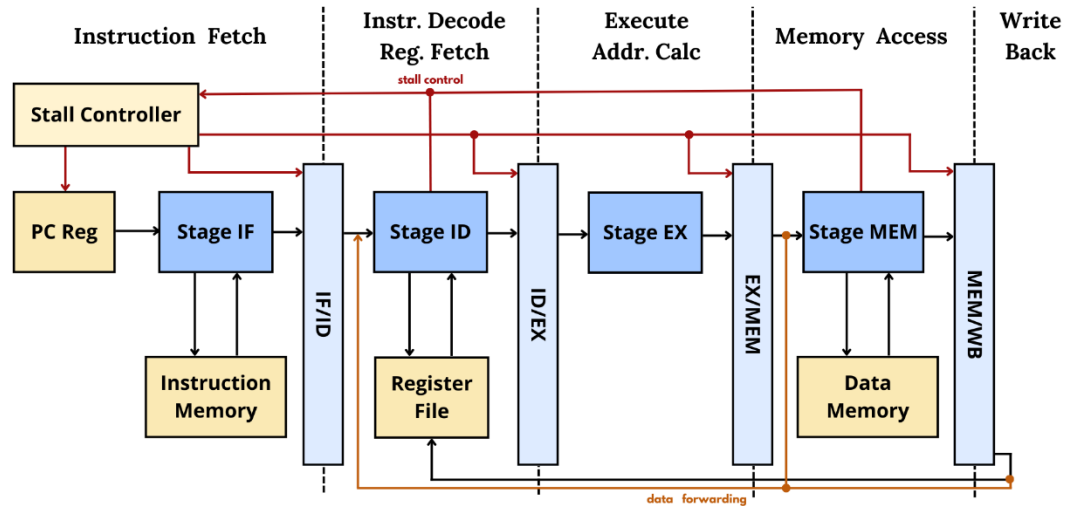


• Branch 指令與 Jump 指令的定址

- JAL 把 offset(20 bits)後補一個 0，跳躍範圍為 $\pm 1M$ bytes。
- JALR 將 offset(12 bits)加上 rs 的值當作跳躍地址。
- Branch 將 instruction 中的 offset(12 bits)後補一個 0，跳躍範圍為 $\pm 4K$ bytes。

- 架構說明

1. 架構圖及說明



2. 個別元件說明

Pipeline register(IF/ID、ID/EX、EX/MEM、MEM/WB):

將前一個 stage 的資料在 posedge clk 時傳送下去

Stall Controller:

接收 Stage ID 和 Stage MEM 的 stall request，控制各 pipeline register 和 PC Reg 的行為

PC Reg:

輸出 PC 給 Stage IF

Stage IF:

從 Instruction Memory 取得 instruction

Instruction Memory:

儲存指令的 memory，根據 pc 的位置將指令傳回 CPU

Stage ID:

解碼 instruction，將後續運算和邏輯判斷會需要的數值往後傳送

Register File:

共 32 個 32 bits 暫存器，第 0 個暫存器數值為 0 且不能被更改

Stage EX:

根據 Opcode 去做相對應的運算

Stage MEM:

決定 data 要存回 MEM 還是 register file

Data Memory:

儲存寫入 MEM 的 data，並在 load 時取出

2. 系統目前可執行之指令

- R 型

31	25 24	20 19	15 14	12 11	7 6	0
funct7	rs2	rs1	funct3	rd	opcode	
7	5	5	3	5	7	
0000000	src2	src1	ADD/SLT/SLTU	dest	OP	
0000000	src2	src1	AND/OR/XOR	dest	OP	
0000000	src2	src1	SLL/SRL	dest	OP	
0100000	src2	src1	SUB/SRA	dest	OP	

- I 型

31	20 19	15 14	12 11	7 6	0
imm[11:0]	rs1	funct3	rd	opcode	
12	5	3	5	7	
I-immediate[11:0]	src	ADDI/SLTI[U]	dest	OP-IMM	
I-immediate[11:0]	src	ANDI/ORI/XORI	dest	OP-IMM	

31	25 24	20 19	15 14	12 11	7 6	0
imm[11:5]	imm[4:0]	rs1	funct3	rd	opcode	
7	5	5	3	5	7	
0000000	shamt[4:0]	src	SLLI	dest	OP-IMM	
0000000	shamt[4:0]	src	SRLI	dest	OP-IMM	
0100000	shamt[4:0]	src	SRAI	dest	OP-IMM	

31	20 19	15 14	12 11	7 6	0
imm[11:0]	rs1	funct3	rd	opcode	
12	5	3	5	7	
offset[11:0]	base	width	dest	LOAD	

註:LOAD 包含 LW/LH/LHU/LB/LBU

- S 型

31	25 24	20 19	15 14	12 11	7 6	0
imm[11:5]	rs2	rs1	funct3	imm[4:0]	opcode	
7	5	5	3	5	7	
offset[11:5]	src	base	width	offset[4:0]	STORE	

註:STORE 包含 SW/SH/SB

- B 型

31	30	25	24	20	19	15	14	12	11	8	7	6	0
imm[12]	imm[10:5]	rs2	rs1	funct3	imm[4:1]	imm[11]	opcode						
1	6	5	5	3	4	1	7						
offset[12,10:5]		src2	src1	BEQ/BNE	offset[11,4:1]		BRANCH						
offset[12,10:5]		src2	src1	BLT[U]	offset[11,4:1]		BRANCH						
offset[12,10:5]		src2	src1	BGE[U]	offset[11,4:1]		BRANCH						

- U 型

31	12	11	7	6	0
imm[31:12]	rd	opcode			
20	5	7			
U-immediate[31:12]	dest	LUI			
U-immediate[31:12]	dest	AUIPC			

- J 型

31	30	21	20	19	12	11	7	6	0
imm[20]	imm[10:1]	imm[11]	imm[19:12]	rd	opcode				
1	10	1	8	5	7				
offset[20:1]				dest	JAL				

imm[11:0]	rs1	funct3	rd	opcode
12	5	3	5	7
offset[11:0]	base	0	dest	JALR

- NOP

31	20	19	15	14	12	11	7	6	0
imm[11:0]	rs1	funct3	rd	opcode					
12	5	3	5	7					
0	0	ADDI	0	OP-IMM					

3. 系統驗證方法與結果分析

- 驗證方法：

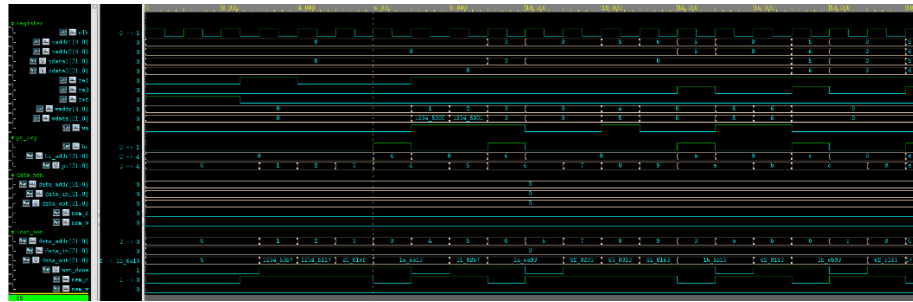
將組合語言轉成機器語言，用 testbench 執行，並觀看波型的變化是否正確，同時用指令：\$monitor()，在 terminal 顯示資料變化。

- 結果分析(需附上 nWave 截圖)

A. “單一”指令正確性

Assembly Language: instructions_test.txt

Machine Code: machine code instructions_test.txt



Start:

LUI 和 AUIPC 測試

LUI x1, 0x12345 # x1 = 0x12345000

AUIPC x2, 0x12345 # x2 = PC + 0x12345000

JAL 和 JALR 測試

JAL x3, jump1 # 跳轉並保存地址到 x3

ORI x20, x20, 1 # 如果沒跳, x20 = 1

jump1:

JALR x4, x3, 3 # x3 = x2 + 12, 並跳轉

ORI x21, x21, 1 # 如果沒跳, x21 = 1

branch 指令測試

ADDI x5, x5, 5

ADDI x6, x6, 6

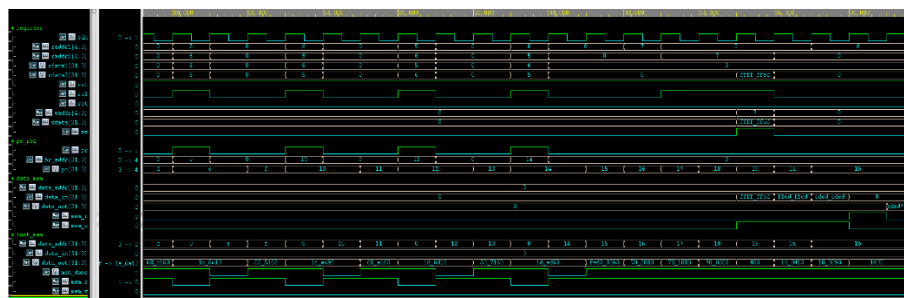
BEQ x5, x5, branch1 # 如果 x5 == x6, 跳轉

ORI x22, x22, 1 # 如果沒跳, x22 = 1

branch1:

BNE x5, x6, branch2 # 如果 x5 != x6, 跳轉

ORI x23, x23, 1 # 如果沒跳, x23 = 1

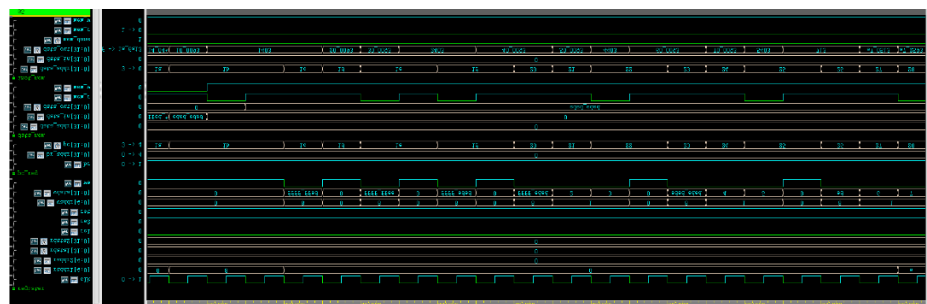


branch2:

```

BLT    x5, x6, branch3    # 如果 x5 < x6，跳轉
ORI    x24, x24, 1         # 如果沒跳，x24 = 1
branch3:
BGE    x6, x5, branch4    # 如果 x6 >= x5，跳轉
ORI    x25, x25, 1         # 如果沒跳，x25 = 1
branch4:
BLTU   x5, x6, branch5    # 如果 x5 < x6（無符號），跳轉
ORI    x26, x26, 1         # 如果沒跳，x26 = 1
branch5:
BGEU   x6, x5, branch6    # 如果 x6 >= x5（無符號），跳轉
ORI    x27, x27, 1         # 如果沒跳，x27 = 1
branch6:
# STORE 指令測試
ADDI   x7, x7, 0xFED
SW      x7, 0(x0)          # 存儲字
SH      x7, 0(x0)          # 存儲半字
SB      x7, 0(x0)          # 存儲字節

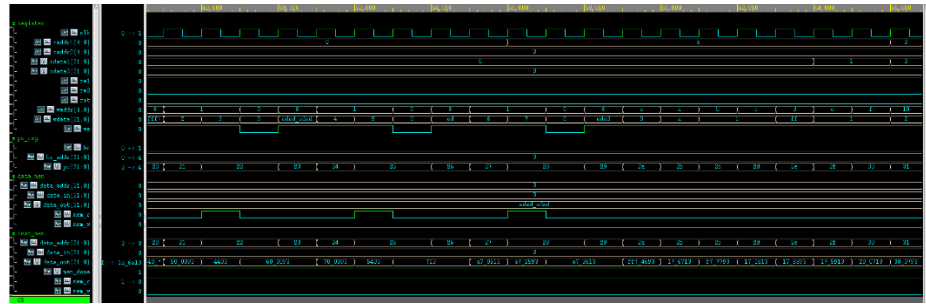
```



```

# LOAD 指令測試
LB      x8, 0(x0)          # 讀取字節
ADDI    x8, x8, 1          # to make stall[0] = 1
LH      x8, 0(x0)          # 讀取半字
LW      x8, 0(x0)          # 讀取字
LBU     x8, 0(x0)          # 讀取半字節(unsigned)

```

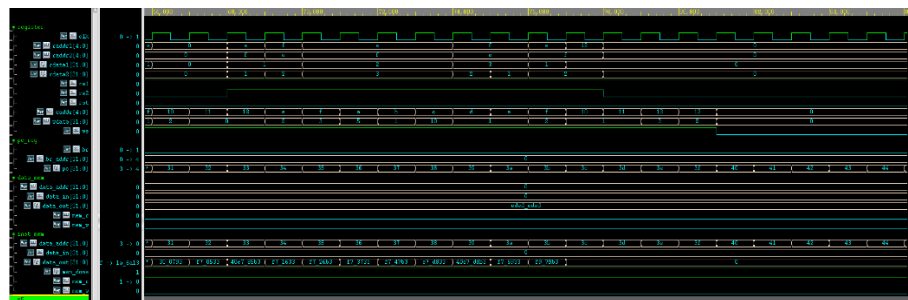



```

LHU    x8, 0(x0)           # 讀取半字(unsigned)

# I-type 指令測試
ADDI   x14, x0, 0
ADDI   x10, x14, 10        # x10 = x14 + 10
SLTI   x11, x14, 10        # x11 = (x14 < 10) ? 1 : 0
SLTIU  x12, x14, 10        # x12 = (x14 < 10 unsigned)
                                ? 1 : 0
XORI   x13, x14, 0xFF      # x13 = x14 ^ 0xFF
ORI    x14, x14, 0x01      # x14 = x14 | 0x01
ANDI   x15, x14, 0x0F      # x15 = x15 & 0x0F
SLLI   x16, x14, 1         # x16 = x14 << 1

```



```

SRLI   x17, x14, 1         # x17 = x14 >> 1
SRAI   x18, x14, 1         # x18 = x14 算術右移 1

# R-type 指令測試
ADDI   x14, x0, 2          # x14 = 2
ADDI   x15, x0, 3          # x15 = 3
ADD     x10, x14, x15       # x10 = x14 + x15
SUB     x11, x15, x14       # x11 = x15 - x14
SLL     x12, x14, x15       # x12 = x14 << x15

```

```

SLT    x13, x14, x15    # x13 = (x14 < x15) ? 1 : 0
SLTU   x14, x14, x15    # x14 = (x14 < x15 unsigned)
                        ? 1 : 0
XOR     x15, x14, x15    # x15 = x14 ^ x15
SRL     x16, x15, x14    # x16 = x15 >> x14
SRA     x17, x15, x14    # x17 = x15 算術右移 x14
OR      x18, x14, x15    # x18 = x14 | x15
AND     x19, x18, x15    # x19 = x14 & x15

```

B. 程式正確性

I. 排序(需自行撰寫組合語言)

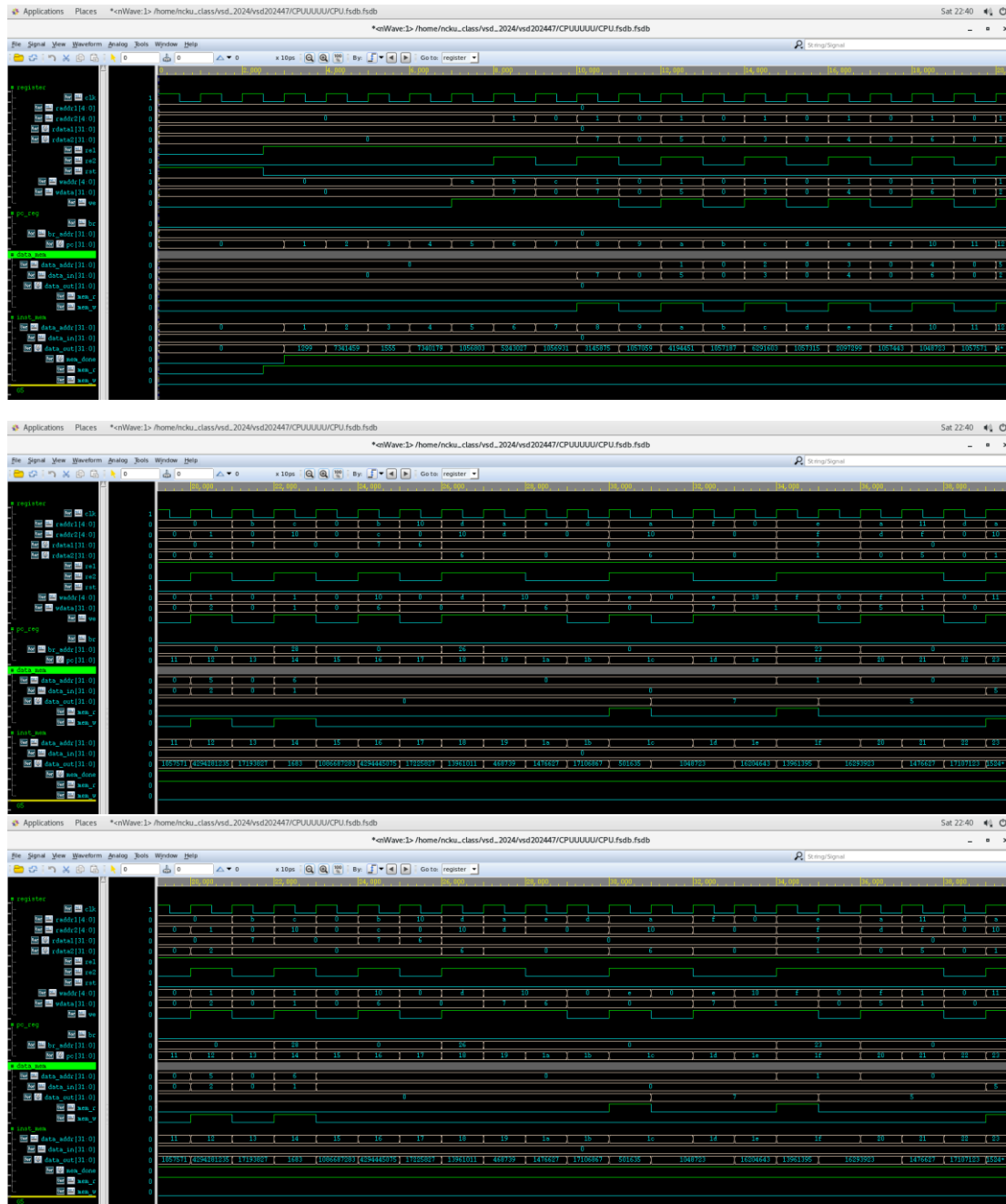
Assembly Language: bubble_sort.txt

Machine Code: machine code bubble_sort.txt

Before: 7, 5, 3, 4, 6, 2, 1

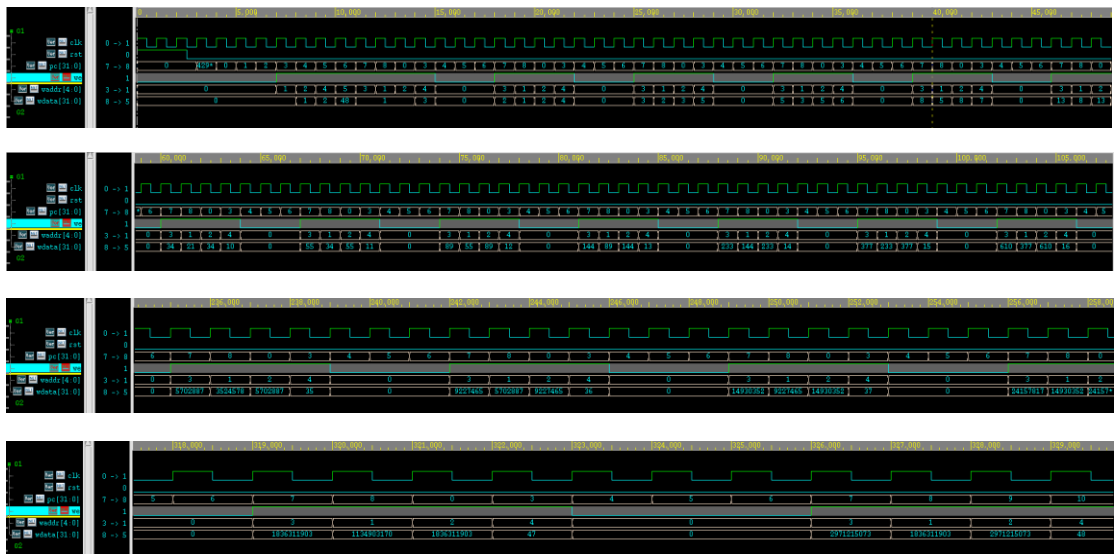
After: 1, 2, 3, 4, 5, 6, 7

Applications Places Terminal									
vsd202447@tsr18:~/CPUUUUU									
File	Edit	View	Search	Terminal	Help				
5190	41	1	2	3	4	5	6	7	
5200	17	1	2	3	4	5	6	7	
5210	18	1	2	3	4	5	6	7	
5220	19	1	2	3	4	5	6	7	
5230	20	1	2	3	4	5	6	7	
5240	40	1	2	3	4	5	6	7	
5250	41	1	2	3	4	5	6	7	
5260	42	1	2	3	4	5	6	7	
5270	43	1	2	3	4	5	6	7	
5280	44	1	2	3	4	5	6	7	
5290	45	1	2	3	4	5	6	7	
5300	46	1	2	3	4	5	6	7	
5310	47	1	2	3	4	5	6	7	
5320	48	1	2	3	4	5	6	7	
5330	49	1	2	3	4	5	6	7	
5340	50	1	2	3	4	5	6	7	
5350	51	1	2	3	4	5	6	7	
5360	52	1	2	3	4	5	6	7	
5370	53	1	2	3	4	5	6	7	
5380	54	1	2	3	4	5	6	7	
5390	55	1	2	3	4	5	6	7	
5400	56	1	2	3	4	5	6	7	
5410	57	1	2	3	4	5	6	7	
5420	58	1	2	3	4	5	6	7	
5430	59	1	2	3	4	5	6	7	
5440	60	1	2	3	4	5	6	7	
5450	61	1	2	3	4	5	6	7	
5460	62	1	2	3	4	5	6	7	
5470	63	1	2	3	4	5	6	7	
5480	64	1	2	3	4	5	6	7	
5490	65	1	2	3	4	5	6	7	
5500	66	1	2	3	4	5	6	7	
5510	67	1	2	3	4	5	6	7	
5520	68	1	2	3	4	5	6	7	
5530	69	1	2	3	4	5	6	7	
5540	70	1	2	3	4	5	6	7	
5550	71	1	2	3	4	5	6	7	
5560	72	1	2	3	4	5	6	7	
5570	73	1	2	3	4	5	6	7	
5580	74	1	2	3	4	5	6	7	
5590	75	1	2	3	4	5	6	7	
5600	76	1	2	3	4	5	6	7	
5610	77	1	2	3	4	5	6	7	
5620	78	1	2	3	4	5	6	7	
5630	79	1	2	3	4	5	6	7	
5640	80	1	2	3	4	5	6	7	
5650	81	1	2	3	4	5	6	7	
5660	82	1	2	3	4	5	6	7	
5670	83	1	2	3	4	5	6	7	
5680	84	1	2	3	4	5	6	7	
5690	85	1	2	3	4	5	6	7	
5700	86	1	2	3	4	5	6	7	
5710	87	1	2	3	4	5	6	7	
CPUUUUU vsd202447@tsr18:~/CPUUUUU testbench.v (~CPUUUUU) - gedit									



File	Edit	View	Search	Terminal	Help
0				0	0
30				1	0
40				2	0
50				3	0
60				4	0
70				5	0
80				6	0
90				7	0
100				8	0
110				9	0
120				10	1
130				4	1
140				5	1
150				6	1
160				7	1
170				8	1
180				9	1
190				10	2
200				4	2
210				5	2
220				6	2
230				7	2
240				8	2
250				9	2
260				10	3
270				4	3
280				5	3
290				6	3
300				7	3
310				8	3
320				9	3
330				10	5
340				4	5
350				5	5
360				6	5
370				7	5
380				8	5
390				9	5
400				10	8
410				4	8
420				5	8
430				6	8
440				7	8
450				8	8
460				9	8
470				10	13
480				4	13
490				5	13
500				6	13
510				7	13
520				8	13
530				9	13
540				10	11

File	Edit	View	Search	Terminal	Help
2840				9	102334155
2850				10	165580141
2860				4	165580141
2870				5	165580141
2880				6	165580141
2890				7	165580141
2900				8	165580141
2910				9	165580141
2920				10	267914296
2930				4	267914296
2940				5	267914296
2950				6	267914296
2960				7	267914296
2970				8	267914296
2980				9	267914296
2990				10	433494437
3000				4	433494437
3010				5	433494437
3020				6	433494437
3030				7	433494437
3040				8	433494437
3050				9	433494437
3060				10	701408733
3070				4	701408733
3080				5	701408733
3090				6	701408733
3100				7	701408733
3110				8	701408733
3120				9	701408733
3130				10	1134903170
3140				4	1134903170
3150				5	1134903170
3160				6	1134903170
3170				7	1134903170
3180				8	1134903170
3190				9	1134903170
3200				10	1836311903
3210				4	1836311903
3220				5	1836311903
3230				6	1836311903
3240				7	1836311903
3250				8	1836311903
3260				9	1836311903
3270				10	2971215073
3280				11	2971215073
3290				12	2971215073
3300				13	2971215073
3310				14	2971215073
3320				15	2971215073
3330				16	2971215073
3340				17	2971215073
3350				18	2971215073
3360				19	2971215073



(session_0) - Jasper Apps (.../CPUUU/jgproject) - Main

File Edit View Design Application Window Help

Superlint

Search

Violation Messages View

Rule Violation

Description (Order by domain)

Analysis Browser

MAC_NR_DMUL_300

```

1 >> define Inst 31:0
2   define InstAddr 31:0
3   module stage_if(rst,pc_i,inst_in,br,br_wait,mem_read,mem_ac
4
5   input rst;
6   input [InstAddr]pc_i;
7   input [Inst] inst_in;
8   //input mem_busy;
9   input br;
10  input br_wait; // from pc_reg(after it receive br)
11
12  output reg mem_read;
13  output reg [InstAddr]mem_addr;
14  output reg [InstAddr]pc_o;
15  output reg [Inst] inst_out;

```

session_0

INFO (ISL018): Started building clock tree
 INFO (ISL018): Finished building clock tree
 INFO (ISL018): Started building reset tree
 INFO (ISL018): Finished building reset tree
 INFO (ISL018): Violation Count: Errors = 0 Warnings = 271 Info = 5
 INFO (ISL018): Started grouping violations
 INFO (ISL018): Finished grouping violations
 313
 [embedded] % save -force -script /home/ncku_class/vsd_2024/vsd202471/CPUUU/superlint.tcl
 [embedded] %

Waiver List

<No waiver data>

Console Warnings / Errors Proof Messa

No proofs running Console input ready

• ICC 檢查結果

Module			Toggle
CPU	***		93% 959 / 1026
Module	Block	Toggle	
pc_reg	*** 100% 12 / 12	98% 68 / 69	
Module	Block	Toggle	
reg_ex_mem	*** 100% 5 / 5	97% 217 / 224	
Module	Block	Toggle	
reg_mem_wb	*** 100% 15 / 15	99% 79 / 80	
Module	Block	Toggle	
reg_if_id	*** 100% 17 / 17	93% 154 / 166	
Module	Block	Expression	Toggle
reg_id_ex	*** 100% 5 / 5	100% 4 / 4	80% 235 / 294
Module	Block	Expression	Toggle
register	*** 100% 16 / 16	100% 17 / 17	99% 115 / 116
Module	Block	Expression	Toggle
stage_id	*** 100% 102 / 102	100% 24 / 24	82% 624 / 763
Module	Block	Expression	Toggle
stage_if	*** 100% 19 / 19	100% 6 / 6	98% 162 / 165
Module	Block	Expression	Toggle
stage_ex	*** 100% 32 / 32	100% 3 / 3	83% 318 / 384
Module	Block	Toggle	
stage_mem	*** 100% 45 / 45	98% 245 / 249	
Module	Block	Toggle	
stall_ctrl	*** 100% 7 / 7	80% 4 / 5	

5. 合成結果

- 速度

Max: 0.00 ns

```
Applications Places Terminal

vsd202447@ttr18:TSMC180

File Edit View Search Terminal Help
# A fanout number of 1000 was used for high fanout net computations.

Operating Conditions: slow Library: slow
Wire Load Model Mode: top

Startpoint: reg_id ex0/ex_opv2 reg_0
(rising edge-triggered flip-flop clocked by clk)
Endpoint: reg_id0/id_inst_reg_12_
(rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type: max

Des/Clust/Port Wire Load Model Library
-----
CPU tsmc18_wl10 slow

Point Incr Path
-----
clock clk (rise edge) 5.00 5.00
clock network delay (ideal) 0.00 5.00
reg_id ex0/ex_opv2_reg_0 /CK (EDFFTRX1) 0.00 # 5.00 r
reg_id ex0/ex_opv2_reg_0 /Q (EDFFTRX1) 0.54 5.54 r
reg_id ex0/ex_opv2[0] (reg_id_ex) 0.00 5.54 r
stage_ex0/opv2[0] (stage_ex) 0.00 5.54 r
stage_ex0/U429/Y (NOR2X2) 0.27 5.81 r
stage_ex0/U606/Y (OR2X2) 0.23 6.03 r
stage_ex0/U159/Y (CLKINX3) 0.14 6.18 f
stage_ex0/U169/Y (INWX1) 0.33 6.50 r
stage_ex0/U212/Y (INWX4) 0.40 6.96 f
stage_ex0/U200/Y (AOI22XL) 0.49 7.45 r
stage_ex0/U586/Y (OAI221XL) 0.52 7.97 f
stage_ex0/U532/Y (INWX1) 0.51 8.48 r
stage_ex0/U70/Y (AOI221X1) 0.33 8.81 f
stage_ex0/U742/Y (AOI222X1) 0.67 9.48 r
stage_ex0/U190/Y (NAND4BX2) 0.26 9.74 f
stage_ex0/reg_wdata[0] (stage_ex) 0.00 9.74 f
stage_id0/ex_reg_wdata[0] (stage_id) 0.00 9.74 f
stage_id0/U427/Y (INWX4) 0.14 9.88 r
stage_id0/U108/Y (NOR2X2) 0.09 9.97 f
stage_id0/ex_reg_wdata[0] (stage_id) 0.00 9.74 f
stage_id0/U427/Y (INWX4) 0.14 9.88 r
stage_id0/U108/Y (NOR2X2) 0.09 9.97 f
stage_id0/U251/Y (NOR2X2) 0.41 10.39 r
stage_id0/U457/Y (OAI221X2) 0.25 10.64 f
stage_id0/U206/Y (CLKINX3) 0.10 10.74 r
stage_id0/U1805/Y (OAI221X2) 0.14 10.87 f
stage_id0/r458/B[0] (stage_id DW01_cnp6_0) 0.00 10.87 f
stage_id0/r458/U221/Y (INWX4) 0.15 11.02 r
stage_id0/r458/U271/Y (XNOR2X4) 0.30 11.32 r
stage_id0/r458/U133/Y (NAND2X4) 0.07 11.39 f
stage_id0/r458/U194/Y (INWX2) 0.14 11.53 r
stage_id0/r458/U129/Y (OAI21X4) 0.10 11.63 f
stage_id0/r458/U121/Y (AOI21X4) 0.29 11.83 r
stage_id0/r458/U105/Y (AOI21X4) 0.10 11.93 f
stage_id0/r458/U73/Y (AOI21X4) 0.18 12.11 r
stage_id0/r458/U9/Y (AOI21X4) 0.10 12.21 f
stage_id0/r458/U186/Y (INWX2) 0.15 12.36 r
stage_id0/r458/LT (stage_id DW01_cnp6_0) 0.00 12.36 r
stage_id0/U1011/Y (AOI21X4) 0.11 12.47 f
stage_id0/U384/Y (NOR4X4) 0.32 12.79 r
stage_id0/U1013/Y (NAND3BX4) 0.20 12.99 r
stage_id0/br (stage_id) 0.00 12.99 r
U4/Y (INWX4) 0.07 13.06 f
U3/Y (CLKINX3) 0.11 13.17 r
reg_id0/br (reg_id_id) 0.00 13.17 r
reg_id0/U62/Y (OR2X4) 0.22 13.39 r
reg_id0/U12/Y (NAND3BX4) 0.22 13.61 r
reg_id0/U70/Y (INWX20) 0.32 13.93 f
reg_id0/U63/Y (AOI222X4) 0.71 14.64 r
reg_id0/U48/Y (INWX4) 0.04 14.68 f
reg_id0/id_inst_reg_12 /D (DFFHQX4) 0.00 14.68 f
data arrival time 14.68

clock clk (rise edge) 15.00 15.00
clock network delay (ideal) 0.00 15.00
reg_id0/id_inst_reg_12 /CK (DFFHQX4) 0.00 15.00 r
library setup time -0.32 14.68
data required time 14.68
data arrival time -14.68
slack (MET) 0.00

*****
Report : timing
-path full
-delay min
-max paths 1
-sort by arrival
```

Min: 0.59 ns

```
Applications Places Terminal

vsd202447@tsr118:TSMC180

File Edit View Search Terminal Help

Version: 0-2019.12
Date : Sat Jan 4 20:09:11 2025
*****

# A fanout number of 1000 was used for high fanout net computations.

Operating Conditions: fast Library: fast
Wire Load Model Mode: top

Startpoint: reg_pc0/pc_reg_3_
(rising edge-triggered flip-flop clocked by clk)
Endpoint: mem_addr_i[3]
(output port clocked by clk)
Path Group: clk
Path Type: min

Des/Clust/Port Wire Load Model Library
-----
CPU tsmc18_wl10 slow

Point Incr Path
-----
clock clk (rise edge) 5.00 5.00
clock network delay (ideal) 0.00 5.00
reg_pc0/pc_reg_3_/_CK (DFFRHQX4) 0.00 # 5.00 r
reg_pc0/pc_reg_3_/_Q (DFFRHQX4) 0.29 5.29 f
reg_pc0/pc[3] (pc_reg) 0.00 5.29 f
stage_if0/pc_i[3] (stage_if) 0.00 5.29 f
stage_if0/0B5/r (AMB2X2) 0.19 5.49 f
stage_if0/mem_addr[3] (stage_if) 0.00 5.49 f
mem_addr_i[3] (out) 0.00 5.49 f
data arrival time 5.49

clock clk (rise edge) 5.00 5.00
clock network delay (ideal) 0.00 5.00
output external delay -0.10 4.90
data required time 4.90
-----
data required time 4.90
data arrival time -5.49
-----
slack (MET) 0.59

Writing Verilog file '/home/ncku_class/vsd_2024/vsd202447/CPUUUUU/CPU_syn.v'.
Warning: Verilog writer has added 3 nets to module CPU using SYNOPSIS UNCONNECTED as prefix. Please use the change_names command to make the correct changes before invoki
Information: Annotated 'cell' delays are assumed to include load delay. (UID-282)
Information: Writing timing information to file '/home/ncku_class/vsd_2024/vsd202447/CPUUUUU/TSMC180/CPU_syn.sdf'. (WT-3)
design_vision> Current design is 'CPU'.
write_sdc -version 1.5 CPU_syn.sdc
1
design_vision> [
```

- 面積(需附上截圖): 1349804.055211 μm^2 (Gate counts=134980)

```
vsd202447@tsr118:TSMC180

File Edit View Search Terminal Help

Note: Symbol # after min delay cost means estimated hold TNS across all active scenarios

Optimization Complete
-----
Warning: Design 'CPU' contains 1 high-fanout nets. A fanout number of 1000 will be used for delay calculations involving these nets. (TIN-134)
Net 'reg_mem_wb1/clk': 1415 load(s), 1 driver(s)

*****
Report : area
Design : CPU
Version: 0-2019.12
Date : Sat Jan 4 20:09:10 2025
*****

Information: Updating design information... (UID-85)
Warning: Design 'CPU' contains 1 high-fanout nets. A fanout number of 1000 will be used for delay calculations involving these nets. (TIN-134)
Library(s) Used:

slow (File: /home/ncku_class/vsd_2024/vsd202400/CIC/180/SynopsysDC/slow.db)

Number of ports: 3149
Number of nets: 12674
Number of cells: 9562
Number of combinational cells: 8106
Number of sequential cells: 1415
Number of macros/black boxes: 0
Number of buf/inv: 2610
Number of references: 15

Combinational area: 147336.235959
Buf/Inv area: 25370.453118
Noncombinational area: 82135.470284
Macro/Black Box area: 0.000000
Net Interconnect area: 1120332.348969

Total cell area: 229471.706242
Total area: 1349804.055211
Loading db file '/home/ncku_class/vsd_2024/vsd202400/CIC/180/SynopsysDC/slow.db'
Information: Propagating switching activity (low effort zero delay simulation). (PWR-6)
Warning: Design has unannotated primary inputs. (PWR-414)
Warning: Design has unannotated sequential cell outputs. (PWR-415)

*****
Report : power
-analysis_effort low
Design : CPU
Version: 0-2019.12
Date : Sat Jan 4 20:09:11 2025
*****
```

- 功耗(需附上截圖)
Dynamic Power: 8.8860 mW
Static Power: 8.1929 uW


```
Applications Places Terminal
vsd202447@ts

File Edit View Search Terminal Help
Library(s) Used:

slow (File: /home/ncku_class/vsd_2024/vsd202400/CIC/100/SynopsysDC/slow.db)

Operating Conditions: slow Library: slow
Wire Load Model Mode: top

Design Wire Load Model Library
-----
CPU tsmc18_wl10 slow

Global Operating Voltage = 1.62
Power-specific unit information :
Voltage Units = 1V
Capacitance Units = 1.000000pf
Time Units = 1ns
Dynamic Power Units = 1mW (derived from V,C,T units)
Leakage Power Units = 1pW

Cell Internal Power = 7.9489 mW (89%)
Net Switching Power = 937.1257 uW (11%)
-----
Total Dynamic Power = 8.8860 mW (100%)
Cell Leakage Power = 0.1929 uW

Power Group Internal Power Switching Power Leakage Power Total Power ( % ) Attrs
-----
io_pad 0.0000 0.0000 0.0000 0.0000 ( 0.00%)
memory 0.0000 0.0000 0.0000 0.0000 ( 0.00%)
black_box 0.0000 0.0000 0.0000 0.0000 ( 0.00%)
clock_network 0.0000 0.0000 0.0000 0.0000 ( 0.00%)
register 7.6621 9.2483e-02 2.6374e+06 7.7572 ( 87.22%)
sequential 0.0000 0.0000 0.0000 0.0000 ( 0.00%)
combinational 0.2868 0.8446 5.5555e+06 1.1370 ( 12.78%)
-----
Total 7.9489 mW 0.9371 mW 8.1929e+06 pW 8.8942 mW

*****
Report : timing
path full
-delay max
-max_paths 1
-sort_by group
Design : CPU
Version: Q-2019.12
Date : Sat Jan 4 20:09:11 2025
*****
vsd202447@tsu118.TSMC180 CPU_syn.v (-CPUUUUUU) - gedit Design Vision
```

6. Layout 結果

```
#####
##          CALIBRE SYSTEM          ##
##          LVS REPORT              ##
#####

REPORT FILE NAME: lvs.rep
LAYOUT NAME: CHIP.sp1 ('chip')
SOURCE NAME: CHIP.sp1 ('chip')
RULE FILE: Calibre-lvs-cur
HCELL FILE: (-automatch)
CREATION TIME: Sat Jan 4 20:36:56 2025
CURRENT DIRECTORY: /project/dr562/pj12/pj1221/layout/Calibre/lvs
USER NAME: pj1221
CALIBRE VERSION: v2020.2_14.12 Thu Apr 2 15:39:27 PDT 2020

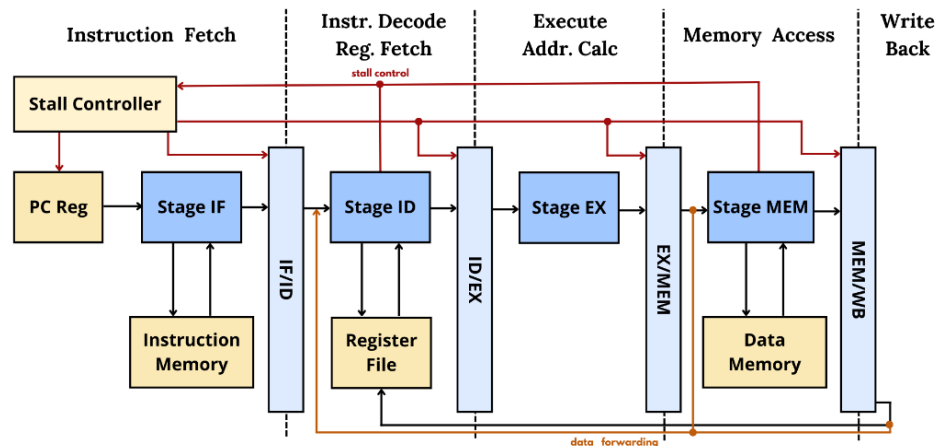
OVERALL COMPARISON RESULTS

# # # # #
# # CORRECT # #
# # # # #

Warning: Ambiguity points were found and resolved arbitrarily.
Warning: LVS property resolution maximum exceeded.
Warning: Source and layout refer to the same data.
```

7. 管線化

- 設計說明：五級管線化



提高吞吐量：通過讓多條指令同時在不同階段執行，實現指令執行的重疊，增加處理器每秒執行的指令數（Instruction Per Cycle, IPC）。降低執行延遲：分割指令執行過程，將每個階段的執行時間最小化，縮短整體的時鐘週期。保持正確性：通過處理資料危障、控制危障以及結構危障，確保指令執行的正確性。

- 階段的劃分：

指令擷取階段（Instruction Fetch, IF）：從指令記憶體中取出下一條指令。解碼階段（Instruction Decode, ID）：解析指令，讀取暫存器的操作數，準備執行控制信號。執行階段（Execute, EX）：根據操作碼執行算術、邏輯、位移或分支運算。存取記憶體階段（Memory Access, MEM）：對資料記憶體進行讀寫操作（僅針對存取類指令）。寫回階段（Write Back, WB）：將結果寫回暫存器檔案。

- 管線化的危障(需說明解決方法)

Structural Hazards: 在同一個 Clock Cycle 中，若 Pipeline 有兩個以上的指令需要使用同一個資源，便會產生 Structural Hazards 的問題。如此一來，後面的指令便會需要延後放入流水線，造成處理器無法在理想狀況下運行(一個 Cycle 處理一條指令)。

Data Hazards:假設有兩個算數指令，分別是 ADD 和 SUB，並且 SUB 的輸入資料是 ADD 的輸出資料。若 SUB 在執行時，ADD 處理的結果還沒有被寫回暫存器，便會造成 Data Hazards。

解決方法:Forwarding:當 Hazard 情況發生時，將 EX 階段算出來的結果直接傳回要執行指令的 register，不用等到結果存回 register file 再存 register file 中取出。

8. 特殊設計

Static Branch prediction: 一律猜測不會進行 branch，當猜測錯誤時，會將原本要執行的指令 flush，並重新去讀取正確的指令執行。

9. 問題與討論

1. 結構危障 (Structural Hazard)

問題描述：結構危障是由於多個指令同時訪問共享硬體資源而引起。

例如，在五級管線架構中，如果指令擷取 (IF) 階段和存取記憶體 (MEM) 階段同時需要訪問單一記憶體，則會產生資源衝突，導致管線執行效率下降。

解決方式：分離指令記憶體和資料記憶體：採用 Harvard 架構，分別使用兩組記憶體來存儲指令和資料，避免衝突發生。雙埠記憶體 (Dual-Port Memory)：允許同時進行讀寫操作，進一步提升資源利用率。時序優化：對可能產生結構危障的階段進行調整，避免多階段同時訪問相同資源的情況。

2. 未來改進方向

(1) 管線飽和度

問題描述：當指令流中包含大量跳轉指令或彼此間具有高度依賴性時，管線利用率會大幅降低，導致處理器吞吐量下降。

改進方向：指令重新排序 (Instruction Scheduling)：利用編譯器在程式編譯階段重排指令，減少危障影響。動態指令發射 (Dynamic Instruction Issue)：允許指令非順序執行，優化管線的資源利用率。

(2) 更先進的分支預測

問題描述：現行的靜態分支預測容易因預測錯誤導致管線清空，增加執行延遲。

改進方向：動態分支預測 (Dynamic Branch Prediction)：採用兩級預測器或分支目標緩衝區 (BTB)，根據執行歷史提高預測準確性。全局歷史預測 (Global History Prediction)：記錄所有分支指令的歷史資訊，基於整體行為進行預測。

(3) 擴展功能

Integer Multiplication and Division (整數乘法): 增加整數乘法指令支援, 例如 MUL 和 DIV, 提升處理器在數值運算方面的性能。

Atomic Instructions: 增加原子性操作 (如 LR/SC 指令) 的支援, 為多執行緒環境提供更有效的同步機制。

浮點運算 (Floating-Point Operations): 引入浮點數指令集 (如 FADD、FMUL 等), 支援高精度科學計算和多媒體應用。

快取記憶體 (Cache Memory): 設計並整合一級快取 (L1 Cache), 降低記憶體訪問延遲, 提升整體系統效能。

總結: 未來的改進將專注於解決結構性瓶頸、提升管線利用率、增加分支預測精準度, 並通過引入更多指令集支援和快取記憶體, 進一步提升處理器的多功能性和效能表現。

10. 心得

- 褚福為: 透過這次的 final project 與組員們實作出一顆 pipeline CPU, 最後有正確的實作出來讓我十分有成就感。在實作的期間, 會遇到很多問題, 需要透過與組員之間的溝通才能解決。這次的作業讓我學到了很多東西, 包括 CPU 相關的知識與學習如何與同學進行有效的溝通, 很開心能順利完成這次作業。
- 余雅瑄: 經過了實作一顆 pipeline CPU, 我學到了如何撰寫測試檔使 Coverage 的百分比提高, 也更加了解了 pipeline CPU 是如何正確運作, 最後也很開心能順利跑完 layout 完成這次的作業。
- 宋晉誠: 這次和組員一起完成了一顆 pipeline CPU, 雖然過程有點匆忙, 但幸好還是趕出來了, 這次的 final project 讓我認知到設計一個系統的困難之處, 和之前寫過的程式不同, 這次不僅要把每個 module 寫好, 更重要的是將每個部份組合在一起, 還能正常運作, 此外還學習到各種 tool, 可以如何輔助我們設計, 收穫滿滿。
- 游宗謀: 感謝教授、助教、TSRI 讓我可以利用這些平時無法接觸到的 EDA tool 完成一顆簡易的 pipeline CPU。過程中我研究了分支預測和浮點數運算的算法與電路結構, 但很遺憾無法在時間內實做出來。我也學會使用許多 EDA tool, 用 Python 寫出編譯器將寫好的組合語言轉成機器語言, 並用 Python 產出在 layout 中需要的檔案。最後感謝組員們願意一起完成它。

11. 分工

姓名	學號	負責項目	貢獻度 (%)
游宗謀	E94106151	Stage EX、測試和 layout 檔案、編譯器	25
宋晉誠	E94101119	Stage ID、測試檔案、系統正確度確認	25
余雅瑄	E94106054	Stage MEM、WB、ICC、layout、superlint	25
褚福為	E94104036	Stage IF、合成檔案、layout	25

12. 參考資料

GitHub - Evensgn/RISC-V-CPU: RISC-V CPU with 5-stage pipeline, implemented in Verilog HDL.

<https://github.com/Evensgn/RISC-V-CPU/tree/master>

iT 邦幫忙::一起幫忙解決難題，拯救 IT 人的一天

<https://ithelp.ithome.com.tw/m/articles/10265705>

Building a MIPS 5-stage Pipeline processor in Verilog (Part 2) | by Lena | Medium

<https://medium.com/@LambdaMamba/building-a-mips-5-stage-pipeline-processor-in-verilog-6d627a31127c>