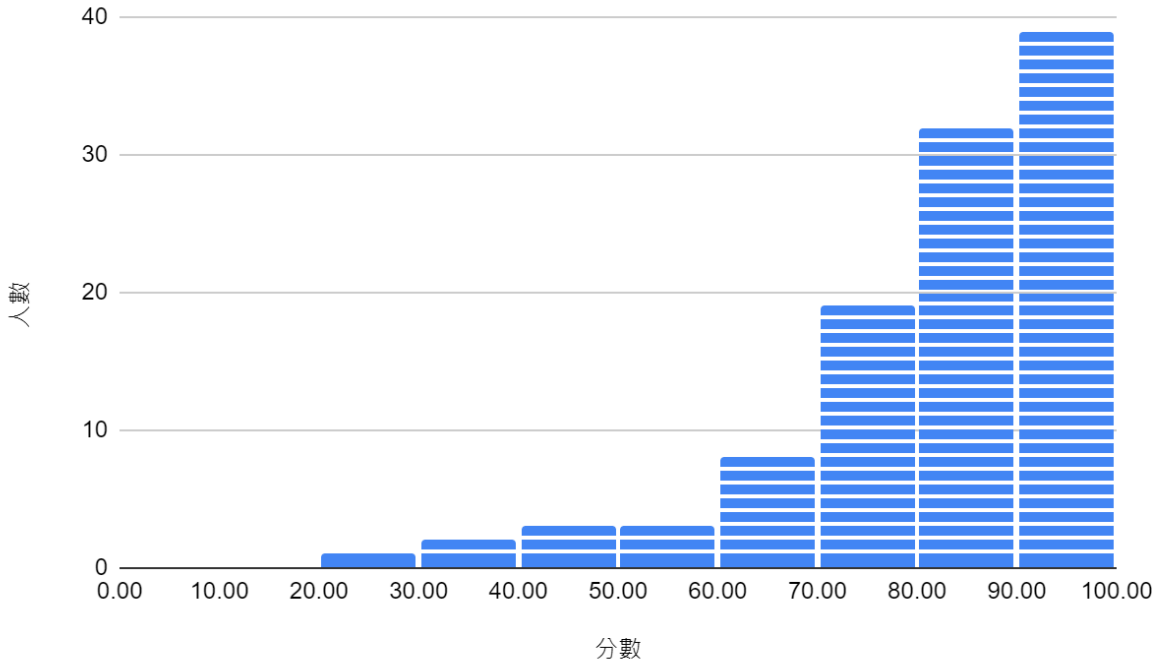
CA midterm

Grade Distribution

- 平均:81.3



Problem 1a

Compilers can have a profound impact on the performance of an application. For a program running on the processor X, compiler A results in a dynamic instruction count of 4×10^9 and an execution time of 12 s, while compiler B results in a dynamic instruction count of 3.6×10^9 and an execution time of 18 s.

a. Find the average CPI for each program given that the processor X has a clock cycle time of 1 ns.

Problem 1a

- CPI(A):12/($4 \times 10^9 \times 1 \times 10^{-9}$)=3
- CPI(B):18/($3.6 \times 10^9 \times 1 \times 10^{-9}$)=5
- 評分標準：
- 全對5分 錯一個扣兩分 有列出CPI=CPU time/(Instruction數×cycle time)得2分

Problem 1b

Compilers can have a profound impact on the performance of an application. For a program running on the processor X, compiler A results in a dynamic instruction count of 4×10^9 and an execution time of 12 s, while compiler B results in a dynamic instruction count of 3.6×10^9 and an execution time of 18 s.

b. Assume the compiler A's codes and the compiler B's codes run on another two different processors Y and Z respectively. The only difference among processor X, Y and Z is the clock rate. If the execution times on the processor Y and Z are the same. What is the ratio of the clock rate of processor Y to the clock rate of the processor Z?

Problem 1b

$$\frac{IC_Y * CPI_Y}{Clock\ Rate_Y} = \frac{IC_Z * CPI_Z}{Clock\ Rate_Z} \rightarrow \frac{Clock\ Rate_Y}{Clock\ Rate_Z} = \frac{4*3}{3.6*5} = \frac{2}{3}$$

評分標準：

答案正確得五分 答案寫成cycle time的比例得三分 算式列對計算錯誤得三分 有列出式子的概念但題目看錯得兩分

Problem 1c

Compilers can have a profound impact on the performance of an application. For a program running on the processor X, compiler A results in a dynamic instruction count of 4×10^9 and an execution time of 12 s, while compiler B results in a dynamic instruction count of 3.6×10^9 and an execution time of 18 s.

c. A new compiler is developed such that it uses only 6.0×10^8 instructions and has an average CPI of 5 to execute this program. What is the speedup of using this new compiler versus using compiler A or B on the original processor X?

Problem 1c

New execution time $=6\times10^8\times5\times1\text{ns}=3\text{sec}$

Speed up A: $18\div3=6$

Speed up B: $12\div3=4$

評分標準：

答對得5分 錯一個扣兩分 加速比例計算正確但speedup理解錯誤得三分 有計算出new execution time =3 sec得兩分

Problem 2a

Suppose you are trying to improve the performance of processor X, which spends 20% of its CPU time executing floating point (FP) operations and 40% of its CPU time executing the load/store operations (40% for other operations). The first improvement method is to make the FP operations run 4 times faster, and the second improvement method is to make the load/store operations run 2 times faster.

a. Which method (the first or the second) can get higher speed-up?

Problem 2a

Speed up 1 : 1/(0.20/4+0.8)=1.18

Speed up 2 : 1/(0.40/2+0.6)=1.25

評分標準:

答案正確得五分，錯一個答案扣兩分，計算錯誤得三分

Problem 2b

Suppose you are trying to improve the performance of processor X, which spends 20% of its CPU time executing floating point (FP) operations and 40% of its CPU time executing the load/store operations (40% for other operations). The first improvement method is to make the FP operations run 4 times faster, and the second improvement method is to make the load/store operations run 2 times faster.

b. Suppose the floating point operations can be improved by a factor of infinity (∞), what is the speed-up?

Problem 2b

Speed up= $1/(0.20/\infty+0.8)=1.25$

評分標準：

答案正確得五分，計算錯誤(式子列對得三分)，Speedup理解錯誤得三分

Problem 3

Assume for a given processor the CPI of arithmetic instructions is 1, the CPI of load/store instructions is 10, and the CPI of branch instructions is 3. Assume a program has the following instruction breakdowns: 500 million arithmetic instructions, 300 million load/store instructions, 100 million branch instructions. Suppose that new, more powerful arithmetic instructions are added to the instruction set. On average, through the use of these more powerful arithmetic instructions, we can reduce the number of arithmetic instructions needed to execute a program by 25%, while increasing the clock cycle time by only 10%. Is this a good design choice? Why?

Problem 3

No.

Original CPU requires 500×1+300×10+100×3=3800 million cycles

New CPU requires 500×75%×1+300×10+100×3=3675 million cycles

However, the new CPU's cycles is 10% longer than original CPU's cycles. The new CPU's 3675 million cycles will take as long as 4042.5 million cycles on the original CPU

評分標準:

答案正確得五分

計算錯誤得三分

列出cycle數=$\sum$ CPI * instruction概念得兩分

Problem 4a

For the following C statement, write a sequence of RISC-V assembly instructions that performs the identical operation. Assume x6 = A, and x17 is the base address of B. Assume that all registers are 64 bits, and B is a long integer array, where long is 8 bytes in C.

```
A = B[0] << 8;
```

Problem 4a

```
ld x6, 0(x17)
slli x6, x6, 8
```

- 評分標準：
  - 正確存取B[0]: 2分
  - 正確對B[0] << 8: 2分
  - 正確將B[0] << 8存進A: 1分

Problem 4b

Provide a ***minimal*** set of RISC-V instructions that could be used to implement the following pseudo instructions:

```
not x5, x6
```

Problem 4b

```
xori x5, x6, -1
```

- 評分標準：
  - 答案完全正確得5分
  - 用2行完成bit-wise invert得2分
  - 沒有支援所有bit數的bit-wise invert且在兩行內完成得1分

Problem 5

Assume that long is 8 bytes in C and all registers are 64 bits. Translate the following loop into C. Assume that the C-level integer *i* is held in register `x5`, `x6` holds the C-level integer called *result*, and `x10` holds the base address of the long integer `MemArray`

```
addi x5, x0, 0
addi x29, x0, 200
LOOP:
 ld x7, 0(x10)
 slli x7, x7, 1
 addi x7, x7, 1
 add x6, x6, x7
 addi x10, x10, 8
 addi x5, x5, 1
 blt x5, x29, LOOP
```

Problem 5

```
for (i = 0, i < 200, i++)
     result += (MemArray[i] * 2 + 1)
```

評分標準：

- 本題中result與MemArray型態不同，但我們不對type casting做扣分。
- 迴圈次數正確: 3分
- 正確讀取MemArray[i]: 3分
- MemArray[i] * 2 + 1式子正確: 3分
- 將MemArray[i] * 2 + 1正確存回result: 1分

Problem 6

Assume that we would like to expand the RISC-V register file to 128 registers. Please answer the following questions.

a. How would this affect the size of each of the bit fields in the R-type instructions?
b. How would this affect the size of each of the bit fields in the I-type instructions?
c. How would the proposed change affect the size of a RISC-V assembly program?

Problem 6

a.(5%) rd, rs1, rs2 增加 2 bits, 其他不變。錯一個field扣3分

b.(5%) rs1, rd 增加 2 bits，其他不變。錯一個field扣3分

c.(3%) 答案合理且完整（完整=有解釋原因）得3分，只寫到一項得2分，沒解釋沒分

- Increasing the size of each bit field potentially makes each instruction longer, potentially increasing the code size overall.
- However, increasing the number of registers could lead to less register spillage, which would reduce the total number of instructions, possibly reducing the code size overall.

Problem 7

Assume a single-cycle RISC-V CPU implementation as shown in *Figure 1* (page 4). What are the control signal values (1/0/X) for the instructions listed in the table below? In the answer sheet, please have your answers organized as the following table.

instruction	Branch	MemRead	MemtoReg	MemWrite	ALUSrc	RegWrite
addi						
add						
slli						
ld						
blt						

Problem 7

- 評分標準：
- 每個row兩分，除MemRead及MemWrite外，全對才給分
- MemRead 0寫成X：-2分
- MemWrite 0寫成X：-2分

instruction	Branch	MemRead	MemtoReg	MemWrite	ALUSrc	RegWrite
addi	0	0	0	0	1	1
add	0	0	0	0	0	1
slli	0	0	0	0	1	1
ld	0	1	1	0	1	1
blt	1	0	X	0	0	0

Problem 8

Assume a single-cycle RISC-V CPU shown in *Figure 1* (Page 4). Answer the following questions for instructions add, addi, sd and beq:

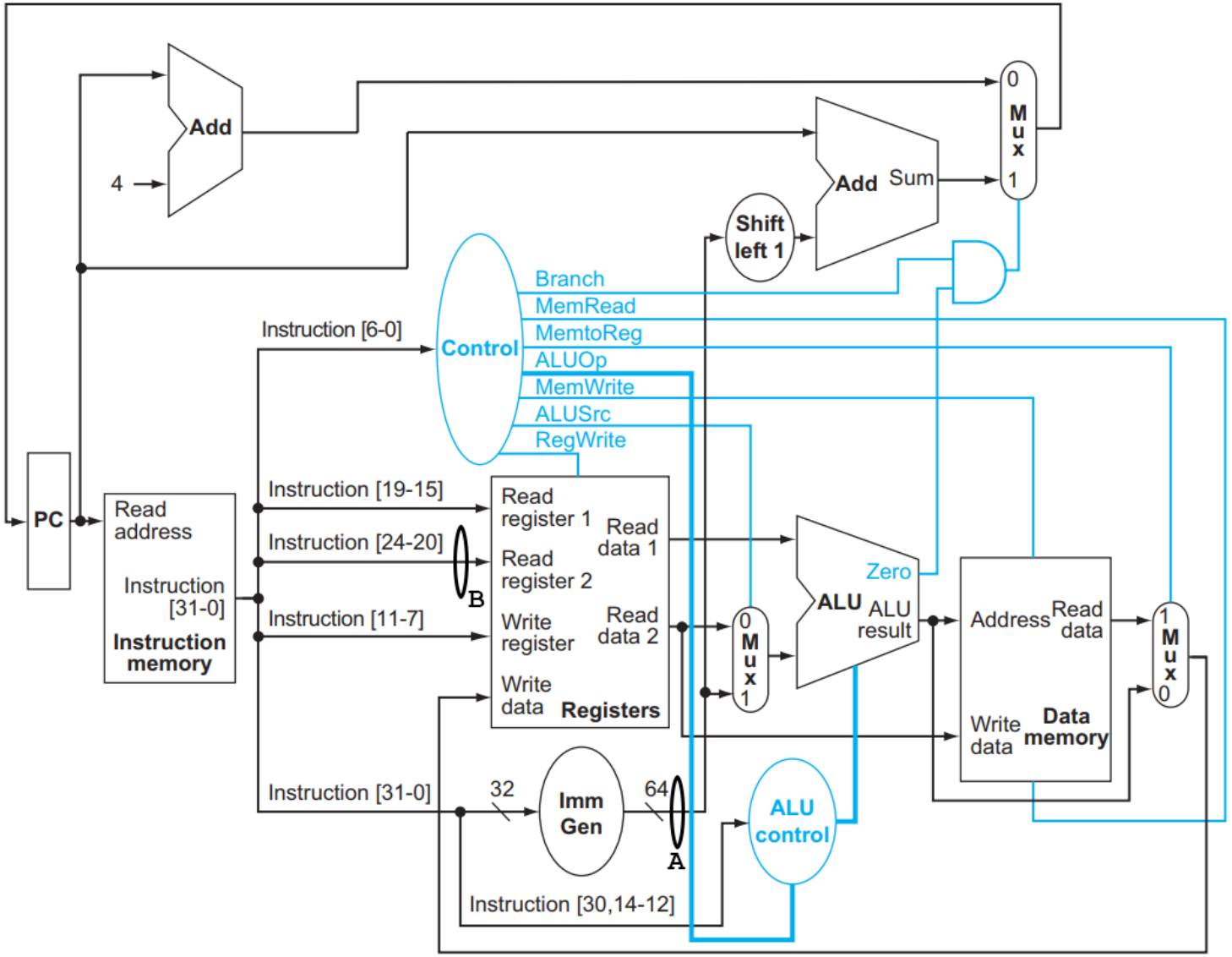
a. If the path labeled A has been cut, which of the instructions can still run correctly?

b. If the path labeled B has been cut, which of the instructions can still run correctly?

c. If the control signal ALUsrc is stuck on 1, which of the instructions can still run correctly?

d. If the control signal MemtoReg is stuck on 0, which of the instructions can still run correctly?

Problem 8


Add
4
0
M
u
x
1
Add
Sum
Shift
left 1
Branch
MemRead
MemtoReg
ALUOp
MemWrite
ALUSrc
RegWrite
Control
Instruction [6-0]
PC
Read
address
Instruction
[31-0]
Instruction
memory
Instruction [19-15]
Instruction [24-20]
Instruction [11-7]
Instruction [31-0]
Read
register 1
Read
register 2
Write
register
Write
data
Read
data 1
Read
data 2
Registers
B
A
32
Imm
Gen
64
Instruction [30,14-12]
ALU
control
Zero
ALU
ALU
result
Address
Read
data
Write
data
Data
memory
1
M
u
x
0

Problem 8

a. `add`

b. `addi`

c. `addi, sd`

d. `add, addi, sd, beq`

評分標準：每小題3分，全對才給分

Problem 9

Examine the difficulty of adding a proposed swap rs1, rs2 instruction to RISC-V.

```
swap rs1, rs2
// Reg[rs2]=Reg[rs1]; Reg[rs1]=Reg[rs2];
```

Assume this is an R-type instruction, and `rd=rs2` (i.e. `instruction[24-20]=instruction[11-7]`)

a. Which new functional blocks (if any) do we need for this instruction?
b. Which existing functional blocks (if any) require modification?
c. What new data paths do we need (if any) to support this instruction?
d. What new signals do we need (if any) from the control unit to support this instruction?
e. Modify *Figure 1* to demonstrate an implementation of this new instruction.

Problem 9

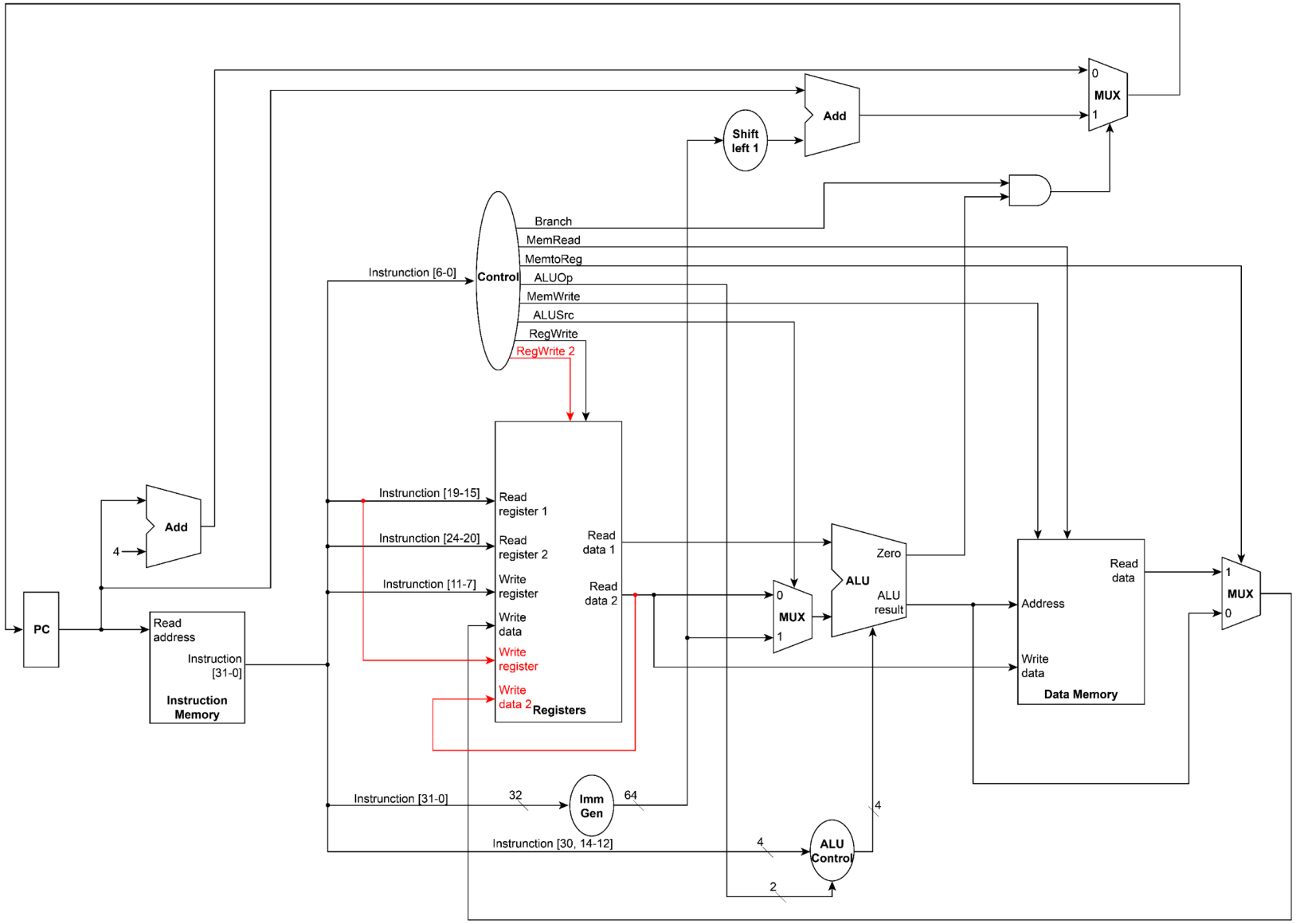
1) Registers should be added two input port: `write register 2`, `write data 2`

2) `ALU` should be able to pass `data1` directly

3) Connect `instruction[19:15]` to `write register 2`, `read data 2` to `write register 2`

4) Add `RegWrite 2` to control the second set of write ports of registers

Problem 9


Add
Shift left 1
MUX
0
1
Branch
MemRead
MemtoReg
ALUOp
MemWrite
ALUSrc
RegWrite
RegWrite 2
Control
Instrunction [6-0]
Add
4
PC
Read address
Instruction [31-0]
Instruction Memory
Instrunction [19-15]
Instrunction [24-20]
Instrunction [11-7]
Read register 1
Read register 2
Write register
Write data
Write register
Write data 2
Registers
Read data 1
Read data 2
MUX
0
1
ALU
Zero
ALU result
Address
Read data
Write data
Data Memory
MUX
1
0
Instrunction [31-0]
32
Imm Gen
64
Instrunction [30, 14-12]
4
ALU Control
4
2

Problem 9

- Register改成2個write：4分
- Datapath足夠且正確(rs1, rs2, Reg[rs1], Reg[rs2]都有正確連回Registers)：4分
- Control signal足夠(有連到能控制Reg[rs1], Reg[rs2]的path)：4分
- 支援原本的ISA：3分

- 其他扣分項目：
  - 寫到imm=0但沒定義ALU要用什麼operation: -2分
  - 多餘且會造成問題的memory/register: -2分
  - 沒寫到也沒交代write register 2：-2分