Name : 김도현

Student ID: 2019112920

교수님 한 학기동안 유익한 강의 해주셔서 감사합니다!

COMP41107, Fall 2020

Final Exam

8 December 2020

Time Limit: 80 minutes

Professor : Seokin Hong

Notes

1. This exam contains 7 pages and 8 questions. Check to see if any pages are missing.
2. Put your name and student ID on the top of each page.
3. Write clearly.
4. Mysterious or unsupported answer will not receive full points for questions that require some calculations. A correct answer, but unsupported by calculations and explanation will receive no point; an incorrect answer supported by substantially correct calculations and explanation will receive partial points.
5. You can answer each question in Korean or English.

Question	Points	Score
1	15	12
2	10	8
3	17	4.5
4	10	10
5	13	13
6	13	~~5.5~~ 6
7	12	6.5
8	10	0
Total	100	60

Name : ______________________ Student ID: ______________________

Q1. [15 points] Answer each question.

3 1. [3 points] Define the term “temporal locality”.

한번 액세스한 메모리에 자주 액세스할 가능성이 높다.

?

3 2. [3 points] Describe four ways to reduce the miss rate of a cache?

✓ 1. 큰 block size

✓ 2. associativity를 늘림

✓ 3. 큰 캐시 size

✓ 4. SW적인 최적화 기법

3 3. [3 points] 32KB fully associative cache is usually slower than 32KB direct mapped cache. Why?

fully associative cache는 hit/miss 판별할 때 캐시의 모든 index를 탐색해야 하기 때문에 search space가 크다. direct mapped cache는 index 하나만 보면 된다.

3 4. [3 points] How does the NMRU replacement policy select a victim block?

가장 최근에 접근한 block을 제외한 모든 block이 victim block이 될수 있다.

5. [3 points] What are the advantage and disadvantage of using write-through policy?

advantage

- ~~쓰기 속도가 빠르다 (block 전체를 캐시에 load하지 않아도 되기에)~~ ?

- 메모리 액세스의 랜덤성이 강할 경우, AMAT이 다른 policy보다 작다.

disadvantage

- memory access의 locality가 큰 경우, hit 하지 못해 AMAT이 커진다.

Name : ____________________ Student ID: ____________________

8

Q2. [10 points] Consider the following instruction mix of a program.

add	ld	sd	beq	jal
30%	20%	20%	20%	10%

1. [2 points] What fraction of all instructions use data memory (in percentage)? ld, sd

   40%

2. [2 points] What fraction of all instructions use instruction memory (in percentage)? 전부다

   100%

3. [2 points] What fraction of all instructions require the sign-extension (in percentage)? beq, jal

   30%

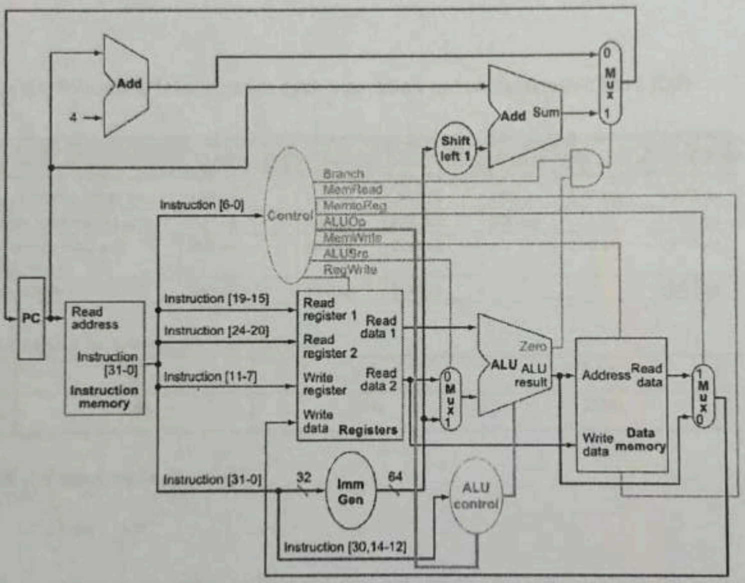
4. [2 points] What fraction of all instructions use the ALU (in percentage)? 전부다

   100%

5. [2 points] What fraction of all instructions write back a value into the register file (in percentage)? add, ld

   50%

Q3. [17 points] Answer each question about the following single-cycle datapath of the RISC-V processor



Consider the following sequence of instructions.

Inst#1	**LOOP:** ld x1, 8(x2)
Inst#2	add x4, x1, x2
Inst#3	sd x4, 24(x2)
Inst#4	beq x2, x3, LOOP

1. [7 points] What are the values of control signals generated by the (main) control unit in above datapath for these four instructions. Fill the blanks in the following table. (0 or 1, X for don't care)

1.5

	ALUSrc	MemtoReg	RegWrite	MemRead
Inst #1	(1) 1	(2) 1	(3) 0	(4) 0
Inst #2	0	(5) X	(6) 1	(7) X
Inst #3	(8) 0	(9) X	(10) X	(11) 1
Inst #4	(12) X	X	(13) X	(14) X

Name : ____________________ Student ID: ____________________

3

2. [3 points] Following table shows how the ALU control signal is generated by the ALU control unit. Fill the blank in this table

Opcode	ALUOp	ALU Function	ALU Control Signal
ld	00	(1) add	(4) 0010
sd	00	(2) add	(5) 0010
beq	01	(3) subtract	(6) 0110
R-type	10	add	0010
		subtract	0110

3. [7 points] The above datapath cannot execute the jal instruction. Explain clearly what modifications are needed to support the jal instruction. Write the list of modifications. You can draw the required modification in the figure above.

Q4. [10 points] Consider the following latencies for each operation of the datapath above (Q3).

Instruction class	Instruction fetch	Register read	ALU operation	Data access	Register write	Total time
Load doubleword (ld)	200 ps	100 ps	200 ps	200 ps	100 ps	800 ps → 600
Store doubleword (sd)	200 ps	100 ps	200 ps	200 ps		700 ps → 500
R-format (add, sub, and, or)	200 ps	100 ps	200 ps		100 ps	600 ps
Branch (beq)	200 ps	100 ps	200 ps			500 ps

Assume that a program has the following instruction mix.

add	ld	sd	beq	jal
30% → 70%	20%	20%	20%	10%

2

1. [2 points] What would the clock cycle time be?

800 ps

2. [8 points] If we modify ld and sd instructions so that they do not use the ALU to calculate the memory address, we can reduce the clock cycle time. However, it would also increase the number of instructions because an add instruction would be used to calculate the memory address before calling the ld and sd instructions.

3

(1) [3 points] What would the clock cycle time be with this modification?

600 ps

5

(2) [5 points] With this modification, would the program having above instruction mix run faster or slower? By how much? Assume every ld/sd instruction is replaced with a sequence of two instructions (add and ld/sd).

더 느려진다. 기존에 비해 소요시간은 1.05 배 늘어남.

ld	~~800ps x 0.2 = 160~~
sd	~~700 ps x 0.2 = 140~~
add	~~600 ps x 0.3 = 180~~
beq ~~etc.~~	~~500 ps x 0.2 = 100~~
jal	~~x 0.1~~

before : 800 x N = 800N

after : 600 x 1.4N = 840N

↓ 손해!

$\frac{840}{800} = \frac{21}{20} \fallingdotseq 1.05$

Name : ______________________ Student ID: ______________________

Q5. [13 points] Assume that individual stages of the 5-stage pipelined datapath have the following latencies.

IF	ID	EX	MEM	WB
250ps	300ps	200ps	300ps	200ps

3

1. [3 points] What is the clock cycle time of the pipelined and non-pipelined datapath?
(Assume that non-pipelined datapath is the single-cycle design and there are no pipeline hazards)

pipelined : 300 ps

non-pipelined : 1250 ps

4

2. [4 points] What is the total execution time for the 10000 instructions on the pipelined and non-pipelined datapath?
And how much is the pipelined datapath faster than the non-pipelined datapath?
(Assume that non-pipelined datapath is the single-cycle design and there are no pipeline hazards)

non-pipelined: ~~300ps~~ 1250 ps × 10000 = ~~3000 ns~~ 12500 ns

pipelined : 300 ps × 10004 = 3001.2 ns

pipelined datapath가 약 4.2배 더 빠름

10004
300
300120.

$\frac{12500}{3001} = \frac{25}{6}$

$6 \overline{)25}$ 4.1 4
24
10
6
40
36
40

6

3. [6 points] Assume that the pipelined datapath uses a static branch prediction technique with an accuracy of 70%, and the branch misprediction penalty is one instruction. And also assume the fraction of branch instructions is 20% of the total instructions and there are no data and structure hazards. What is the total execution time for the 10000 instructions on this pipelined datapath?

3181.2 ns

hit 70%

branch 20% < miss 30% — penalty (one instruction)

3181.200

instructions (N)

80%

N = 10000

0.8N + 0.2N × 0.7 + 0.2N × 0.3 × 2

= (0.8 + 0.14 + 0.1 × 2)N = ~~2.14N = 21400~~

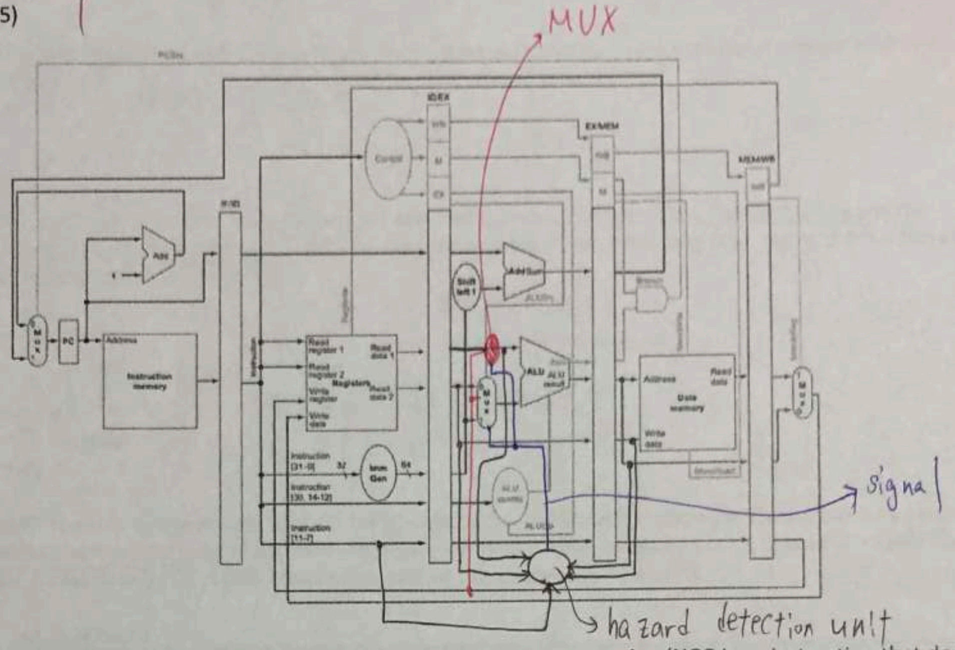
= 1.06 N = 10600

300 ps × 10604 = 3181.2 ns

Name : ______________________ Student ID: ______________________

Q6. [13 points] Consider the following assembly code, and assume that it is executed on a five-stage pipelined datapath below.

```
1개 → add x15, x12, x11     0.5
      ld  x13, 4(x15)
      ld  x12, 0(x2)
2개 → → or  x13, x15, x13
      sd  x13, 0(x15)        1
```



1.5

1. [3 points] Insert minimum number of NOP instructions to ensure correct execution (NOP is an instruction that does nothing).

3

2. [4 points] Change and/or rearrange the code to reduce the number of NOP instructions needed. You can use the additional registers.

4

```
add x15, x12, x11
ld  x13, 4(x15)
or  x13, x15, x13
ld  x12, 0(x2)       1          stall 2회로 줄임?
sd  x13, 0(x15)
```

0.5

3. [6 points] Extend the above pipelined datapath to support the data forwarding and the hazard detection. Write the list of modifications. You can draw the required modification in the figure above.

ID/EX 또는 ~~Ex/MEM~~ 파이프라인 레지스터의 Rs 값이 IF/ID 의 Write register에 해당하는 값과 같을 때, forwarding 발생

forwarding 방식은, data memory에 쓰려고 하는 값을 바로 ALU의 input으로 당겨오는 방식.

0.5

Name : ______________________ Student ID: ______________________

Q7. [12 points] Suppose a 32-bit memory address generated by CPU is split into three fields as follows.

4 5

[31-9]	[8-5]	[4-0]
Tag	Index	Offset

3

1. [3 points] What is the block size in byte?

32 byte

2. [3 points] If a cache is the 8-way set associative cache, what is the cache size in byte (total amount of data stored in the cache)?

~~128 byte~~ 2kB

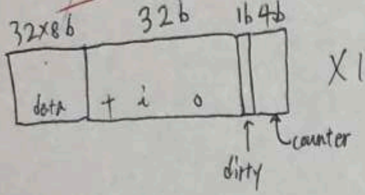
16 × 8 = 128, 7

7+5 = 12

3.5

3. [6 points] Suppose that a cache is the 16-way set associative cache, and uses the write-back policy and the counter-based LRU replacement policy. What is the total number of bits (including data, tag, and other bits used for cache management) in the cache?

4688 bits



$(256 + \cancel{32} + \cancel{8}1 + 4) \times 16$

$= 293 \times 16$

1 1 0.5

293 × 16 = 1758, 293, 4688

Q8. [10 points] Consider a memory hierarchy consisted of L1 data cache, L1 instruction cache, and main memory. Assume that main memory access time is 70 ns and that the clock cycle time is determined by the hit time of L1 cache. The following table shows data for L1 caches attached to each of two processors P1 and P2.

36 × 63 = 108, 218, 2.268

36 × 52 = 72, 180, 1.872

5.6

0.7

L1 data cache

	cache size	miss rate (misses/accesses)	hit time
P1	2 KiB	8.0%	0.7 ns
P2	4 KiB	6.0%	1 ns

L1 instruction cache

	cache size	miss rate (misses/accesses)	hit time
P1	2 KiB	3.0%	0.7 ns
P2	4 KiB	1.0%	1 ns

1. [5 points] Assume a base CPI of 1.0 and that 36% of total instructions are load and store instructions. What is the total CPI for P1 and P2? Which processor is faster and how much? (Base CPI of 1.0 means that an instruction is executed per a clock cycle if there are no pipeline stalls)

P2 가 $\frac{5068}{3512}$ 배 빠름

P1: 0.7 + 0.03 × 70 + 0.36 × (0.7 + 0.08 × 70) = 0.7 + 2.1 + 0.36 × 6.3 = 5.068

P2: 1 + 0.01 × 70 + 0.36 × (1 + 0.06 × 70) = 1 + 0.7 + 0.36 × 5.2 = 3.512

2. [5 points] If 75% of total memory accesses is for instructions and 25% of total memory accesses is for data, what are the Average Memory Access Time (AMAT) for the processors P1 and P2 (in cycles)?

P1: 0.7 + 2.1 + 0.25 × 6.3 = 4.375 ns

P2: 1 + 0.7 + 0.25 × 5.2 = 3 ns

25 × 63 = 75, 150, 1575

25 × 52 = 50, 125, 1300

Name : 김도현 Student ID: 2019112920

COMP41107, Fall 2020
Midterm Exam
20 October 2020
Time Limit: 120 minutes

Professor : Seokin Hong

Notes

1. This exam contains 12 pages and 8 questions. Check to see if any pages are missing.
2. Put your name and student ID on the top of each page.
3. Write clearly. If I can't read it I can't grade it.
4. Mysterious or unsupported answer will not receive full points for questions that require some calculations. A correct answer, unsupported by calculations, explanation will receive no point; an incorrect answer supported by substantially correct calculations and explanation will receive partial points.
5. You can answer each question in Korean or English.
6. Appendix provides information about some commonly used RISC-V instructions and registers for your reference.

Question	Points	Score
1	10	9
2	13	~~8~~ 9
3	12	7
4	18	10
5	16	11
6	9	9
7	12	10
8	10	3
Total	100	68

Name : 강도현

Student ID: 2019112920

Q1. [10 points, 1 point each] True or False Questions (fill in T or F).

1. (F) Moore's law is often used in parallel computing to predict the theoretical maximum speedup achieved with multiple processors (cores).

2. (T) Technology scaling results in reduction of the size of transistors.

3. (T) In RISC-V, conditional branch instructions use the register addressing to calculate the destination address.

4. (F) CPU Execution time includes time spent waiting for I/O.

5. (F) In RISC-V, arithmetic instruction can use memory operands.

6. (T) Accessing registers uses less energy than accessing memory.

7. (T) When adding two negative numbers, the overflow occurs if the sign bit of the result is 0. $(-) + (-) \overset{\checkmark}{\rightarrow} (+)$

8. (F) In RISC-V, register x1 always contain value 1. It cannot be overwritten.

9. (F) In RISC-V, load and store instructions use the same instruction format.

10. (T) In IEEE 754 floating-point format, the infinity is represented by a non-zero fraction and the largest biased exponent (255 for single-precision and 2047 for double-precision).

Name : 김도현　　　　Student ID: 2019112920

Q2. [13 points] Answer each question.

1 1. [1 points] 추상화 helps us deal with complexity in computer hardware and software by hiding low-level details and offering a simpler model at higher levels.

2. [2 points] List five components in the Von Neumann Architecture.

Input

Output

Control Unit

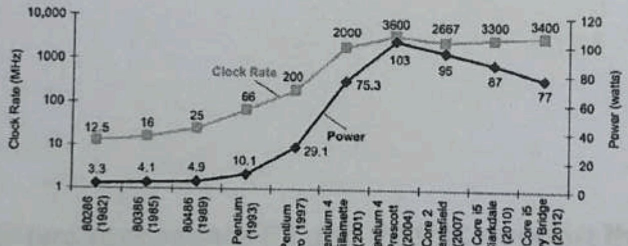
Processor

ALU

0 3. [2 points] The following figure shows the clock rate (frequency) and power consumption of Intel X86 processors over eight generations. *Why doesn't the clock rate increase after the Pentium 4 Prescott processor?*

서의 집적도를 일정수준이상 높일수

기 때문



4. [1 points] ISA is the interface between hardware and software. ISA specifies the set of basic operations, data types, storages, and memory addressing mode, etc. Fill in the blanks.

2 5. [2 points] What are the advantages and disadvantages of the RISC compared to the CISC?

장점
- 인스트럭션 길이가 고정적
- CPU 설계가 용이

단점
- 인스트럭션 길이가 길어짐

Name : 깅조헌 Student ID: 2019112920

3

6. [3 points] Assume that **the size of a register is 8 bits**, and registers x5 and x6 hold the values 10011100_2 and 10000000_2, respectively. (a) **What is the value of x3** for the following assembly code? (b) **Is the result in x3 the desired result**, or **has there been an overflow**?

add x3, x5, x6

```
  10011100
+ 10000000
----------
  00011100  → overflow
```

2

7. [2 points] What is the decimal number of each of the following 4-bit two's complement signed numbers (4th bit is MSB).

1) A = 0111_2 7

2) B = 1110_2 −2

-0010

Q3. [12 points] Consider two processors P1 and P2 that support the same ISA.

- There are four classes of instructions in the ISA.
- P1 has a clock rate of 3 GHz and P2 has a clock rate of 4 GHz.
- The counts of each instruction class for a program, and the CPI for each instruction class in the two processors are as below:

	Class	CPI on P1	CPI on P2	Instruction Counts
√	Arithmetic	1	2	500
	Branch	1	3	200
√	Load/Store	2	4	200
	Floating-point	4	6	100

1000

1. [1 point] What is the clock period (Clock cycle time) of each processor?

P1: 500 + 200 + 400 + 400 = 1500 (clock cycle)

P2: 1000 + 600 + 800 + 600 = 3000 (clock cycle)

P1: $\frac{1500}{3G} = 500$ ns

P2: $\frac{3000}{4G} = 750$ ns

Name : 김조현 Student ID: 2019112920

1 △ 2. [2 points] What is the total clock cycles for the program on each processor?

P1: 1500 풀이X

P2; 3000

2 3. [2 points] What is the global CPI for the program on each processor?

$P1; \frac{1500}{1000} = 1.5$

$P2; \frac{3000}{1000} = 3.0$

○ 4. [3 points] Which processor is faster, and by how much?

P1이 2배 빠름

P2에 비해

4 5. [4 points] Assume that the given program can be parallelized to run over multiple processors. When parallelizing the program, the **number of arithmetic and load/store instructions per processor is divided by p** (where p is the number of processors) but the **number of branch and floating-point instructions per processor remains the same.** What is the **maximum speedup** that can be achieved with **multiple P1 processors** compared to using a single P1 processor?

$\text{multiple P1; } \frac{500}{p} + 200 + \frac{400}{p} + 400 = 600 + \frac{900}{p} \text{ (clock cycle)}$

$p \to \infty, \text{ multiple P1} = 600$

$\text{single P1} = 1500$

$\frac{\overset{5}{\cancel{1500}}}{\underset{2}{\cancel{600}}} = 2.5$

∴ multiple P1 은 single P1에 비해

최대 2.5 배 speedup 가능.

Name : 김도헌　　　　　　　　Student ID: 2019112920

Q4. [18 points] Answer each question about RISC-V ISA.

6

2 1. [2 points] Write **a single C statement** that corresponds to the three RISC-V assembly instructions below. Use variable i, j, k, n, and m for registers x2, x3, x4, x5, and x6, respectively.

i j k n m

```
add  x4, x2, x3     k=i+j
addi x5, x2, 24     n=i+24    → m=(i+j)-(i+24)
sub  x6, x4, x5     m=k-n       =j-24
```

$\boxed{m = j - 24;}$ ← 세미콜론

2 2. [4 points] Write RISC-V assembly code that does the following computations.

B[12] = A[i-j]

Assume that A and B are arrays of 8-byte values and the variable i and j are assigned to registers x5 and x6, respectively. Base address of the array A is in register x7 and the base address of the array B is in register x8.

x5 x6 x7 x8

```
sub x5 x5 x6 # x5 = i-j          0.5
add x5 x5 x7 # x5 = A[i-j]       0.5
sd x5 96(x8) # B[12] = x5        1
```

3. [3 points] Suppose a 64-bit data (0xFFAB12343412CDAB) is stored at address 0x0 of the memory as follows. **When loading the data to register x2, what data will be stored in x2 for each of the following load instructions?** Suppose that **x1 and x2 are 64-bit registers**.

a. lb x2 1(x1)

b. lbu x2 1(x1)

a, b 둘 다 0x0000000000010010 가 저장됨

?

Registers

x1	0x0000000000000004
x2	0

Memory

Address	Data
0x00000000	AB
0x00000001	CD
0x00000002	12
0x00000003	34
0x00000004	34
→ 0x00000005	12
0x00000006	AB
0x00000007	FF

$12_{10} \rightarrow 00010010_2$

Name : 김조현 Student ID: 2019112920

3

4. [3 points] Show how does the assembler convert the following branch instruction into a pair of instructions if its target address (**L1**) is too far to be accommodated in the 12-bit immediate field.

```
      beq x2, x3, L1
        ..
        ..
L1:     ..
        ..
```

```
    bne  x2, x3, L2
    addi x5, x0, L1
    jalr x0, 0(x5)
L2:
    ,
    ,
    ,
L1:
```

3

5. [3 points] Provide a RISC-V assembly language instruction for the following binary value:
0000 000|0 0001| 0000 1|000|0000 1|011 0011

R-type

f7 rs2 rs1 . f3 rd →add

x1 x1 x1

⇒ add x1, x1, x1

0

6. [3 points] What is the value of x7 for the following assembly code? Assume the following register contents:

x5 = 0xF0000000FFFFFFFF, x6=0x1234567812345678

```
srai  x7,  x5, 4
or    x7, x7, x6
```

$F_{16} = 1111_2$

x6 = 1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8
x7 = F F F F F F F F F 0 0 0 0 0 0 0
or
x7 = 1 2 3 4 5 6 7 8 1 0 0 0 0 0 0 0 ← result

Name : 강조현　　　　　　　　Student ID: 2019112920

11

Q5. [16 points] Consider the following RISC-V loop:

Low address ↓ High address

Address (decimal)	Instructions	x6	x5
24	LOOP: beq x6, x0, DONE	6	0
28	addi x6, x6, -1	5	2
32	addi x5, x5, 2	4	4
36	jal x0, LOOP	3	6
40	DONE:	2	8
		1	10
		0	12

2 1. [2 points] What is the address of "jal x0, LOOP" instruction?

36

3 2. [3 points] Assume that the register x6 is initialized to the value 6 and register x5 is initialized to the value zero. What is the final value in register x5?

6 0

12

0 3. [3 points] Rewrite the above assembly code to replace the instruction "beq x6, x0, DONE" with the instruction "blt x6, x0, DONE".

if(x6 < 0) → jump to DONE

3 4. [3 points] Rewrite the above assembly code to use **jalr** instruction instead of **jal** instruction in the above assembly code.

```
LOOP ; beq  x6,x0, DoNe
       addi x6,x6,-1
       addi x5,x5, 2
       addi x7, x0, LOOP
       jalr x0, 0(x7)
DONE ;
```

3 5. [5 points] Translate "beq x6, x0, DONE" assembly instruction into a 32-bit machine instruction (binary value).

0 0 0 0 0 _ 1 00000 00110 000 0 1 0 0 0 1100011

12 10 5 ... 4 1 11

0.8 0.5 0.5 1.5

SB

op; 1100011

f3; 000

DONE = 10100~~X~~

32 8

5 4 3 2 1

Name : 김조헌

Student ID: 2019112920

1 8 23

Q6. [9 points] Answer each question about the floating-point numbers.

3

1. [3 points] Write the IEEE 754 binary representation of a decimal number -0.625 in single-precision.

1 01000000 11111111111111111111111

0.5 $\frac{1}{2}$

0.125 $\frac{1}{8}$

$-0.101 \rightarrow -1.010 \times 2^{-1}$

$-1 = -0...001$
$= 1...11$

3

2. [3 points] Calculate the following binary floating-point addition. **Show every step** of the floating-point addition. Assume we can store only 4 digits of significant.

$1.011_2 \times 2^{-1} + 1.00_2 \times 2^{-2} = ?$

$\Rightarrow 1.011_2 \times 2^{-1} + 0.100_2 \times 2^{-1}$ 지수 맞춤

$= 1.111_2 \times 2^{-1}$ 가수끼리 연산

$= 1.1110_2 \times 2^{-1}$ normalize, 반올림은 필요X

3

3. [3 points] Calculate the following binary floating-point multiplication. **Show every step** of the floating-point multiplication. Assume we can store only 4 digits of significant.

$1.001_2 \times 2^{-1} \times -1.11_2 \times 2^{-2} = ?$

$\Rightarrow$ 가수 $= 2^{-3}$

지수 $= 1.11111$

$\Rightarrow 1.1111$ (반올림)

$\Rightarrow 1.1111 \times 2^{-3}$ (normalize) $\Rightarrow -1.1111 \times 2^{-3}$ (부호)

```
   1.001
 ×  1.11
 -------
    1001
   1001
  1001
 -------
  1.11111
```

Q7. [12 points] Answer each question about multiplication.

2

1. [2 points] Calculate the following binary multiplication.
$1001_2 \times 1010_2 = ?$

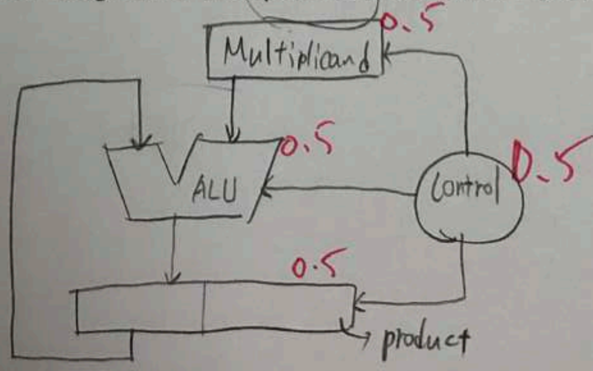
1011010

```
     1001
     1010
   ------
    1001
  1001
  -------
  1011010
```

3

2. [5 points] Draw the block diagram of the optimized version of 4-bit multiplier that uses a 4-bit ALU.

bit 없음



connection 1

Name : 강조헌 Student ID: 2019112920

5

3. [5 points] We will perform the above binary multiplication (1001_2 x 1010_2) with the multiplier drawn in Q7-2. Fill in the following table that shows the value of each register of the multiplier for each iteration.

- In this table, value in each register is in binary

Iteration	Multiplicand	Product
0	1001	0000 1010
1	1001	0000 0101
2	1001	0100 1010
3	1001	0010 0101
4	1001	0101 1010
5	1001	0101 1010 (done)
6	1001	

3

Q8. [10 points] Translate the following C code to RISC-V assembly code.

- You can use any RISC-V registers

sp==x1

```
main(){
    int y;
    y=func(10);
}

int func(int a){
    int i;
    int b;
    for(i=0; i<a; i++){
            b+= i;
    }
    return b;
}
```

```
addi x1, x1, -16 ?
addi x5, x0, FUNC
sd   x5, 0(x1)
addi x6, x0, 10    → 2
sd   x5, 8(x1)
jal  x1, FUNC      → 1

FUNC:
```