# CENG 3420 Midterm (2024 Spring)

**Name:** _____ **ID:** _____

**Q0 (0 marks)**

1. What is your last digit of your SID (0 is regarded as 10)? This value is defined as `NUM_1` in the whole question paper.
2. What is your last two digits of your SID (00 is regarded as 100)? This value is defined as `NUM_2` in the whole question paper.
3. What is your last three digits of your SID? This value is defined as `NUM_3` in the whole question paper.

Example: if your SID is 12345678, then `NUM_1` $= 8$, `NUM_2` $= 78$, `NUM_3` $= 678$.

**Q1** (30 marks) **Select and fill** the **single** correct answer.

1. Which one is not a component of a computer?
   (A) processor
   (B) I/O
   (C) memory
   (D) fan
2. RV32I has a ____ instruction length with ____ general-purpose registers.
   (A) 32-bit 32
   (B) 32-bit 64
   (C) 32-bit 16
   (D) 64-bit 32
3. In the S-type instruction, why we split imm into different portions?

| imm[11:5] | rs2 | rs1 | funct3 | imm[4:0] | opcode | S-type |
|---|---|---|---|---|---|---|

   (A) To align most imm bits among different types
   (B) To follow IEEE standard
   (C) To follow ACM standard
   (D) To ease user understanding
4. Inside our CPU, do we need to implement substraction hardware? Why?
   (A) Yes; substraction hardware is via very clear design
   (B) Yes; we follow the same idea as adder hardware
   (C) No; we can use adder hardware to implement substraction
   (D) No; compiler will help to translate all substractions into addings
5. Which instruction is equivalent to the following program? (blt: branch less than; ble: branch if $\leq$; bgt: branch if $>$; bge: branch if $\geq$)
   **slt** `t0, s1, s2`
   **beq** `t0, zero, Label`

(A) **blt** `s1, s2, Label`

(B) **ble** `s1, s2, Label`

(C) **bgt** `s1, s2, Label`

(D) **bge** `s1, s2, Label`

6. When we compile a Recursive Procedure, the return address is ____ and stored in ____.

   (A) `ra`; stack

   (B) `ra`; heap

   (C) `a0`; stack

   (D) `a0`; heap

7. In signed binary representation, we use 2-complementary format, and which one is NOT the reason:

   (A) Single "0"

   (B) a + (-a) = "0"

   (C) Ease substraction

   (D) Ease multiplication

8. We have the registers `t1 = 0xFEDCBA98`, the value of `t2` for the following sequence of instructions is ____.

```
slli t2, t1, 24
srai t2, t1, 24
```

   (A) `0xFEDCBA98`

   (B) `0x00000098`

   (C) `0x0000BA98`

   (D) `0xFFFFFF98`

9. In datapath, will we conduct register file (RF) reading and decoding at the same time?

   (A) No; we should be decoding first so that we know where is the RF rading address

   (B) No; we do RF reading first

   (C) Yes; we read RF at the same time so that our datapath can be faster

   (D) Yes; in the instruction we dont encode RF address

10. Dividing $11100_2$ by $11_2$, the quotient is ____ and the reminder is ____.

    (A) $1000_2$ $01_2$

    (B) $1110_2$ $00_2$

    (C) $1001_2$ $01_2$

**A1** 1. D

2. A

3. A

4. C

5. D

6. A

7. D

8. X
9. C
10. C

**Q2** (10%) Consider the following RISC-V instructions. Please note that we treat `NUM_1%2` and `NUM_1%2+1` as decimal values.

```
li a1, NUM_1%2
li a2, NUM_1%2+1
li a3, 4
LOOP:
slti t0, a3, 1
bne t0, zero, DONE
add a4, a1, a2
addi a1, a2, 1
addi a2, a4, 1
addi a3, a3, -1
jal x0, LOOP
DONE:
# end of the program
```

1. How many times is the branch instruction (bne) executed? (4%)
2. What are the final values of `a1` and `a2`. (6%)

**A2** These are suggested solutions.

1. The branch instruction (bne) will be executed 5 times.
2. (a) If `NUM_1%2` = 0, a1 = 10, a2 = 16;
   (b) If `NUM_1%2` = 1, a1 = 15, a2 = 24;

**Q3** (10%) We have an `int` (32 bits) array named $arr1 = \{0x01, 0x23, 0x45, 0x67, 0x89\}$. The pointer of `arr1`'s first element stored in register `a1`. We also have the registers `t1 = 0x12345678`, `t2 = 0xFEDCBA98`
Please answer the following questions:

(Each question is stand-alone. Instructions in one question won't affect registers in other questions.)

1. What is the value of `t3` for the following sequence of instructions? (2%)

```
lw a2, 4(a1)
addi t3, a2, NUM_1
```

2. What is the value of `t1` for the following sequence of instructions? (4%)

```
slli t1, t1, 16
srli t1, t1, 20
```

3. What is the value of `t2` for the following sequence of instructions? (4%)

```
slli t2, t2, 24
srai t2, t2, 24
```

**A3** 1. `NUM_1` + 0x23

2. 0x00000567
3. 0xFFFFFF98

**Q4** (10%) A RISC-V assembly program is shown below.

```
main:
li a1, 5
jal ra, foobar
j exit

foobar:
addi sp, sp, -8
# Missing line 1
# Missing line 2
li a0, 1
blt a1, a0, exit_foobar
addi s0, a1, 0
addi a1, a1, -1
jal ra, foobar
mul a0, a0, s0

exit_foobar:
# Missing line 3
# Missing line 4
addi sp, sp, 8
jr ra

exit:
```

Note: `blt` is branch if less than.

1. Write down the missing lines to save and restore two registers. Hint: One of them is `s0`. The other one is critical for function call. (5%)
2. What is the final value of `a0`? (5%)

**A4**

1. 
   - sw s0, 0(sp)
   - sw ra, 4(sp)
   - lw s0, 0(sp)
   - lw ra, 4(sp)
2. 120 or 0x78

**Q5** (10%) What is $-2024.5_{10}$ in IEEE-754 single precision binary floating point format? Write your transformation process and convert the binary representation to a hexadecimal number with "0x" beginning.

**A5**

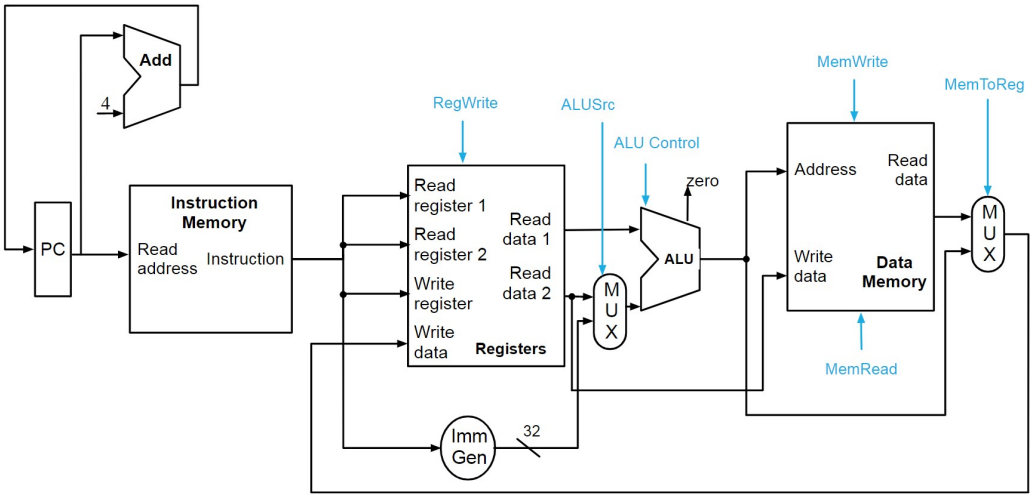
1. $-2024.5$ to binary 111 1110 1000.1000 0000 0000 0...
2. 111 1110 1000.1000 0... =1. 1111 1010 0010 0000 0000 000 $\times 2^{10}$
3. signal bit =1, exponent = 10+127 = 137 = 1000 $1001_2$
4. mantissa 1111 1010 0010 0000 0000 000 (hide the leftmost '1')
5. final result 1 1000 1001 1111 1010 0010 0000 0000 000 = 0xC4FD1000

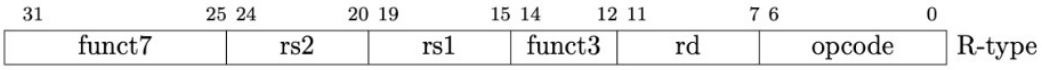
**Q6** (15%) The figure below shows the format and datapath of an R format instruction.

1. Assume we have an instruction whose machine code is 0x0040C1B3. Please write down the instruction in assembly language. The registers are `x0, x1, ..., x31`. (5%)
2. Please use lines to connect the hardware and the corresponding function.(4%)

| Function | Hardware |
|---|---|
| • Fetch and Updata PC | Registers |
| • Decode Instruction | Instruction Memory, PC and Add |
| • Execute | Data memory |
| • Write/Read Data | ALU |

3. • a.In the datapath, which is the factor will determine the cycle time of instructions? (3%)
   • b. Why do we need MUX in the datapath? (3%)




| Inst | Name | FMT | Opcode | funct3 | funct7 |
|---|---|---|---|---|---|
| add | ADD | R | 0110011 | 0x0 | 0x00 |
| sub | SUB | R | 0110011 | 0x0 | 0x20 |
| xor | XOR | R | 0110011 | 0x4 | 0x00 |
| or | OR | R | 0110011 | 0x6 | 0x00 |
| and | AND | R | 0110011 | 0x7 | 0x00 |
| sll | Shift Left Logical | R | 0110011 | 0x1 | 0x00 |
| srl | Shift Right Logical | R | 0110011 | 0x5 | 0x00 |
| sra | Shift Right Arith* | R | 0110011 | 0x5 | 0x20 |
| slt | Set Less Than | R | 0110011 | 0x2 | 0x00 |
| sltu | Set Less Than (U) | R | 0110011 | 0x3 | 0x00 |

**Q6** 1. xor x3,x1,x4

2. • Fetch and Updata PC —— Instruction Memory, PC and Add

- Decode Instruction —— Registers
- Execute —— ALU
- Write/Read Data ——Data memory

3. a.The longest path (or The critical path).(Other reasonable answers will also get the marks)
4. b.To share datapath elements between two different instruction classes. (Other reasonable answers will also get the marks)

**Q7** (15%) In $CPU_A$ with clock cycle equal to 1ns, assuming negligible delays (for muxes, control unit, sign extend, PC access, shift left2, wires) except:

- Instruction fetch and update PC (`IF`), Read/write data from/to data memory (`MEM`) ($3 + \texttt{NUM\_1}\%2$ ns)
- Execute R-type; Calculate memory address (`EXE`) (2 ns)
- Register fetch and instruction decode (`ID`), Write the result data into the register file (`WB`) (1 ns)

1. Calculate the delay time for the following instructions: `beq`, R/I-type instruction, `sw`, `lw`
2. Assuming the total number of instructions of $CPU_A$ are 10 (`beq`, `sw`, `lw` and 7 R/I-type instruction) and these instructions appear with equal probability in a program, what is the CPI of $CPU_A$?
3. Assuming that the clock rate (frequency) of $CPU_A$ is 1.2 times that of $CPU_B$, the CPI of $CPU_B$ is 0.75 times that of $CPU_A$ and the same number of instructions for both, which CPU is faster comparing to CPU time ($= CPI \times CC \times IC$)?

**A7** These are suggested solutions. Assume $\texttt{NUM\_1} = 0$, then

1. # of delay time `beq` $= \texttt{IF} + \texttt{ID} + \texttt{EXE} = 6$ ns
   # of delay time R/I-Type $= \texttt{IF} + \texttt{ID} + \texttt{EXE} + \texttt{WB} = 7$ ns
   # of delay time `lw` $= \texttt{IF} + \texttt{ID} + \texttt{EXE} + \texttt{MEM} + \texttt{WB} = 10$ ns
   # of delay time `sw` $= \texttt{IF} + \texttt{ID} + \texttt{EXE} + \texttt{MEM} = 9$ ns
2. CPI $= \frac{6+7*7+10+9}{10\times 1} = 7.4$
3. CPU Time$_B = 0.75 \times CPI_A \times 1.2 \times CC_A \times IC_A = 0.9 \times$ CPU Time$_A$, $CPU_B$ is faster because it has smaller CPU Time.

Assume $\texttt{NUM\_1} = 1$, then

1. # of delay time `beq` $= \texttt{IF} + \texttt{ID} + \texttt{EXE} = 7$ ns
   # of delay time R/I-Type $= \texttt{IF} + \texttt{ID} + \texttt{EXE} + \texttt{WB} = 8$ ns
   # of delay time `lw` $= \texttt{IF} + \texttt{ID} + \texttt{EXE} + \texttt{MEM} + \texttt{WB} = 12$ ns
   # of delay time `sw` $= \texttt{IF} + \texttt{ID} + \texttt{EXE} + \texttt{MEM} = 11$ ns
2. CPI $= \frac{7+7*8+12+11}{10\times 1} = 8.6$
3. CPU Time$_B = 0.75 \times CPI_A \times 1.2 \times CC_A \times IC_A = 0.9 \times$ CPU Time$_A$, $CPU_B$ is faster because it has smaller CPU Time.