COMP1411 (Spring 2024) **Introduction to Computer Systems**

Individual Assignment 2 Duration: <u>00:00, 16-Mar-2024</u> ~ <u>23:59, 18-Mar-2024</u>

Name WANG Yuqi
Student number [redacted]

Question 1. [1 mark]
In this question, we use the Y86-64 instruction set (please refer to Lectures 4-6).
1(a) [1 mark]
Write the machine code encoding of the assembly instruction:
`rmmovq %rax, 0x1124(%rsp)`
Please write the bytes of the machine code in hex-decimal form, i.e., using two hex-decimal digits to represent one byte. You are allowed to leave spaces between adjacent bytes for better readability. The machine has a little-endian byte ordering.
Show your steps. Only giving the final result will NOT get a full mark of this question.
***Answer*:**

rmmovq : iCode = 4, ifun = 0
register IDs : %rax = 0
%rsp = 4
D = Extent_to_8Bytes (0x1124) = 0x 00 00 00 00 00 00 11 24
Full Instruction: 0x40 04 $\underbrace{24\ 11\ 00\ 00\ 00\ 00\ 00\ 00}_{\text{Reversed (Little Endian)}}$

1(b) [3 marks]
Consider the execution of the instruction "`rmmovq %rax, 0x1124(%rsp)`". Assume that the data in register `%rsp` is 0x456, the value of PC is 0x360. Use "**vm**" to represent the data that will be written to the main memory.
Describe the steps done in the following stages: Fetch, Decode, Execute, Memory, Write Back, PC update, by filling in the blanks in the table below.
Note that you are required to fill in the generic form of each step in the second column; and in the third column, fill in the steps for the instruction "`rmmovq %rax, 0x1124(%rsp)`" with the above given values.

0x456

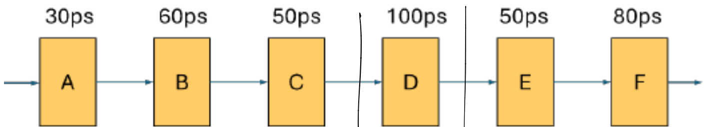
The symbol "←" means reading something from the right side and assign the value to the left side. X:Y means assign the highest 4 bits of a byte to X, and assign the lowest 4 bits of the byte to Y.

***Answer*:**

Stages	rmmovq rA, D(rB)	rmmovq %rax, 0x1124(%rsp)
Fetch	icode: ifun ← M_1[PC] rA:rB ← M_1[PC+1] valC ← M_8[PC+2] valP ← PC+10	icode: ifun ← M_1[0x360] = 4:0 rA:rB ← M_1[0x361] = 0:4 valC ← M_8[0x362] = 0x1124 valP ← 0x36A
Decode	valA ← R[rA] valB ← R[rB]	valA ← R[%rax] = vm valB ← R[%rsp] = 0x456
Execute	valE ← ValC + ValB	valE ← 0x456 + 0x1124 = 0x157A
Memory	M_8[ValE] ← valA/vm	M_8[0x157A] ← valA/vm
Write back		
PC update	PC ← ValP	PC ← 0x36A

Question 2. [2 marks]

Suppose a combinational logic is implemented by 6 serially connected components from A to F. The whole computation logic can be viewed as an instruction. The number on each component is the time delay spent on this component, in time unit ps, where 1ps = 10^{-12} second. Operating each register will take 20ps.



Throughput is defined as how many instructions can be executed on average in one second for a pipeline in the long run, and the unit of throughput is IPS, instructions per second.

Latency refers to the time duration starting from the very first component and ending with the last register operation finished, the time unit for latency is ps.

For throughput, please write the result in the form X.XX * 10^{Y} IPS, where X.XX means one digit before the dot and two fractional digits after the dot, and Y is the exponent.

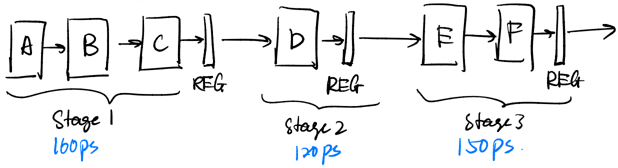
Make the computation logic a 3-stage pipeline design that has the maximal throughput. Note that a register shall be inserted after each stage to separate their combinational logics. By default, a register will be inserted after the last stage, i.e., after step F.

- Please answer how to partition the stages.
- Please compute the throughput and latency for your pipeline design, with steps.

Objective: Make longest section as short as possible.

For 1st section: A, AB, are both not optimal.

∴



Delay = 160ps (Because 160 > 150 > 120; Delay determined by the slowest stage)

Throughput = $\frac{1}{\text{Delay}} \times N$

$= \frac{1s}{160ps} \times 3$

$= \frac{1s}{160 \times 10^{-12} \times 3s} \times 3$

$= 6,250,000,000 = 6.25 \times 10^9$ IPS

Latency = 160ps × 3

= 480 ps

Question 3. [4 marks]

The following byte sequence is the machine codes of a C function compiled with the Y86-64 instruction set (refer to Lecture 6). The memory address of the first byte is 0x200. Note that the byte sequence is written in hex-decimal format, i.e., each number/letter is one hex-decimal number representing 4 binary bits, and two numbers/letters represent one byte. **Assume the machine is a little-endian byte order machine.** Assume that by default the value in register %rax will be returned.

30F3140000000000000030F102000000000000006300623373 2E020000
00000000603061137016020000000000000090

Please write out the assembly instructions (in Y86-64 instruction set) corresponding to the machine codes given by the above bytes sequence, and explain what this function is computing.

Address	Hex	Assembly
0x200	30 F3 1400000000000000	irmovq $0x14, %rbx
0x20A	30 F1 0200000000000000	irmovq $0x02, %rcx
0x214	63 00	xorq %rax, %rax
0x216	62 33	andq %rbx, %rbx
0x218	73 2E02000000000000	je 0x22E
0x221	60 30	addq %rbx, %rax
0x223	61 13	subq %rcx, %rbx
0x225	70 1602000000000000	jmp 0x216
0x22E	90	ret

Complete Assembly:

```
.pos 0x200
      irmovq $0x14, %rbx
      irmovq $0x02, %rcx
      xorq %rax, %rax
loop:
      andq %rbx, %rbx
      je done
      addq %rbx, %rax
      subq %rcx, %rbx
      jmp loop
done:
      ret
```

Explaination:

The function is computing the result of (0x14 + 0x12 + 0x10 + … + 0x02 + 0x00) = 110

Or, more generally:

$$\sum_{n=0}^{\left\lfloor \frac{X}{2} \right\rfloor} 2n + X \bmod 2$$

Where, X = 0x14 in this specific case.