University of Sri Jayewardenepura

M.Sc. in Computer Science
First Semester Course Unit Examination – April, 2024

CSC 541 2.0 Computer System Architecture
(Time: 2 hours)

This paper consists of 4 questions on 3 pages. Answer all questions.
Some formulas are provided on the last page. You may freely use them.

Question 01 [*A total of* 25 *marks*]

(a) What are the types of memory present in a computer system? Briefly explain the purpose of each type and indicate where these memories are located within a computer system.

[08 Marks]

(b) Explain what is meant by Little endian memory of a 32-bit computer. Show how the following information are represented in such a computer.

Name of housing project (string): Green Life
Price per unit in thousand rupees (integer): 18000

[08 Marks]

(c) Consider the byte **01010100**. In order to calculate the Hamming error correcting code, insert the four parity bit positions into this byte and obtain their parity values.

[09 Marks]

Question 02 [*A total of* 20 *marks*]

(a) Providing an illustrative example, explain how a coprocessor could increase the performance of a computer.

[05 Marks]

(b) Consider a dual–pipelined processor that can issue two instructions per clock cycle. Let A, B and C be threads and A_i, B_i and C_i are the instructions issued by these three threads, respectively. Assume that there is a miss penalty of two clock cycles for level 1 cache, three clock cycles for level 2 cache, and 10 clock cycles for last level cache if an instruction misses in the cache. Fig 2.1 shows the instructions issued by these threads. A square represents a clock cycle and blank squares mean CPU stalls due to cache misses.

A_1			A_2			A_3	A_4			A_5	A_6	A_7				
B_1				B_2	B_3	B_4				B_5			B_6	B_7	B_8	
C_1	C_2			C_3			C_4	C_5	C_6			C_7	C_8	C_9		

Fig 2.1

Show the instructions issued in the first 15 clock cycles when fine-grained multithreading technique on a dual-pipelined processor is employed.

[15 Marks]

Question 03 [*A total of* 25 *marks*]

(a) A set of benchmark programs was executed on two computers, and the following measures were recorded. The clock frequency of Computer *A* is 0.2 GHz higher than the clock frequency of Computer *B*.

	Instruction Type	No. of Instructions Executed	Clock Cycles per Instruction
Computer *A*	ALU	18, 001, 547	1
	BRANCH	12, 412, 042	4
	OTHER	13, 945, 764	3
Computer *B*	ALU	24, 102, 334	1
	BRANCH	18, 247, 937	2
	OTHER	12, 001, 207	3

Calculate the clock frequency of Computer *B* if it is given that the above benchmark is executed in 1.02 times faster on Computer *B* than Computer *A*.

[15 Marks]

(b) Consider the specifications given below with respect to a computer system:

Cache memory size: `16MB`
Cache organization: `Single-level, direct-mapped`
Cache line size: `128 bytes`
Memory word size: `8 bytes`
Maximum amount of RAM accessible to the Operating System: `256GB`

(i) Show the components of the virtual memory address with their sizes.
(ii) What is the location in cache for the memory reference `EF000FE` (given in hexadecimal)?

[10 Marks]

Question 04 [*A total of* 30 *marks*]

(a) Providing illustrative examples describe the differences between immediate addressing mode and direct addressing mode.

[05 Marks]

(b) Let `A` be an array of size 40 of 8-byte integers whose base address is assigned to the register `x21`. Translate the following C code fragments to RISC-V assembly language.

(i)
```
A[1] = (A[12] + b)*8 + 5;
```

(ii)
```
while (A[i] >= 0)
    i += 1;
```

[20 Marks]

(c) What is the RISC-V assembly language instruction of the following RISC-V machine language instruction given in hexadecimal?

`060C0C93`

[05 Marks]

****

$$\text{Execution time}_{New} = \text{Execution time}_{Old} \times \left[\left(1 - \text{Fraction}_{Enhanced}\right) + \frac{\text{Fraction}_{Enhanced}}{\text{Speedup}_{Enhanced}} \right]$$

CPU execution time = (CPU clock cycles + Memory stall cycles) x Clock cycle time

Memory stall cycles = No. of misses x Miss penalty
= IC x Misses per instruction x Miss penalty
= IC x Memory references per instruction x Miss rate x Miss penalty

RISC-V instruction formats

	funct7	*rs2*	*rs1*	*funct3*	*rd*	*opcode*
Type R	7-bit	5-bit	5-bit	3-bit	5-bit	7-bit

	immediate	*rs1*	*funct3*	*rd*	*opcode*
Type I	12-bit	5-bit	3-bit	5-bit	7-bit

	immediate	*rs2*	*rs1*	*funct3*	*immediate*	*opcode*
Type S	7-bit	5-bit	5-bit	3-bit	5-bit	7-bit

RISC-V instruction encoding

Instruction	*funct7*	*funct3*	*opcode*
`add`	0	0	51
`sub`	32	0	51
`addi`	-	0	19
`ld`	-	3	3
`sd`	-	3	35