University of Sri Jayewardenepura

M.Sc. in Computer Science
First Semester Course Unit Examination – August, 2023

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) Explain the difference between static RAM (SRAM) and dynamic RAM (DRAM) with respect to computer memories. Describe where these types of memories are used in a computer system.

[08 Marks]

(b) What is the purpose of internal data bus in a processor? Consider a processor core with 64-bit wide data I/O bus running at the speed of 2133MHz. How much information can be transferred into the processor per second?

[08 Marks]

(c) Consider the byte **01110100**. In order to calculate the Hamming error correcting code, insert the four parity bit positions to this byte, and obtain their parity values.

[09 Marks]

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

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	A_8				
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 coarse-grained multithreading technique on a dual-pipelined processor is employed, assuming level 2 and last level cache misses as expensive stalls.

[20 Marks]

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

(a) Consider a processor with a split cache. Miss rates are 3% and 5% respectively for instruction and data caches. Miss penalty is 100 cycles for all misses. 40% of instructions are data accesses. The processor has a CPI (cycles per instructions) of 2.0 without considering memory stalls cycles. Compare the performance of this processor with a hypothetical processor that always hits in the cache.

[15 Marks]

(b) The advertised average seek time of a hard disk is 5ms, and the disk rotates at 7200 revolutions per minute. The disk can transfer data at the rate of 150MB per second. The disk controller overhead is 2ms. Calculate the average time to read two 512-byte sectors if

(i) Both sectors are in the same track
(ii) Two sectors are in two different tracks

[10 Marks]

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

(a) By providing illustrative examples describe the differences between direct addressing and register indirect addressing modes.

[10 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;
A[1] = A[1] + 5;
```

(ii)

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

[15 Marks]

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

```
0B5B3C23
```

[05 Marks]

****

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

$$\text{MIPS} = \frac{\text{IC}}{\text{Execution time} \times 10^6}$$

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

Average disk access time = Average seek time + average rotational delay
+ transfer time + controller overhead

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