Roll No CS23I1025 | Name Raghul Vishwa . S

Indian Institute of Information Technology, Design and Manufacturing, Kancheepuram

Mid Semester Examination, February 2025

Course Code: CS2007 | Course Title: Computer Organization and Architecture
Batch: CS, CS-AI, DD 2023 | Category: Core
Date of Examination: 25th February 2025
Duration: 1 hour 30 minutes | Max. Marks: 25

Instructions to students (if any):

- Use of calculator is prohibited.
- Show the steps wherever necessary

1. There is a malfunctioning keyboard which doesn't accept any input other than 0's and 1's. Another problem with the keyboard is that if we press a key (0 or 1) it gets stuck and it will accept the input atleast 2 times. Eg. If a person wants to give input as 01 he may end up giving (accepting) 0011 as input because of the malfunction. Consider this scenario and accordingly choose the most suitable hardware algorithm for multiplication for the accepted input. Also show the steps of multiplication in detail for the following accepted input (Because of Malfunction). (4 marks)
   A: 00001100 B:00111100

2. Subtract a floating point number from the other (A-B) given in IEEE-754 form and give the result in Floating point Hex form: (3 marks)
   A: 42068000 (Hex Representation)
   B: 41A90000 (Hex Representation)

3. Divide B by A where A: 01001, B: 10001 using a suitable division algorithm used in hardware. (3 marks)

4. Write the equivalent RISC code for the following C Program: (5 marks)

```
void main(){
int a=5, b, n=3;
b=mul(a,n);
}

int mul(int a, int n) {
if (n == 0) {
return 1;
}
return a + mul(a, n - 1);
}
```

Note:

i. Define the list of variables that might be pushed into the stack on each function (procedure) call
ii. Before every procedure call, create the required space in the stack to store the registers/variables/ arguments (prologue) and delete it at the end of the procedure call (epilogue).

Roll No:	Name:	

5. The floating point standard IEEE 754 was introduced in 1985. Micro-processors earlier to 1985, such as 80286, which used 16-bit words, did not have a 16 bit floating point standard. Assuming there was a floating point as below, please answer the following questions.

(3 marks)

1 bit	5 bits	10 bits
S	Exponent	Mantissa

a. How much would be the bias value of the exponent?
b. What is the smallest positive normalized number that can be represented using this format?
c. What is the smallest negative normalized number that can be represented using this format?

6. Assume that you are designing a RISC processor (32 bit instruction set) that uses a register file containing 128 registers. The ISA for processor supports an instruction as follows: **opcode rD, rA, rB** where rD is the destination register, rA and rB denote the registers that store the two operands of the operation. Each register has to be represented uniquely along with an instruction. Eg. add rD, rA, rB where add is one such instruction that can be executed.

Answer the following questions. **(2 marks)**

a. Assuming that the ISA mandates fixed-length opcodes (i.e. code for every operation (instruction) is of same size in terms of bits), how many instructions (operations) can this ISA support?
b. How would you modify the above instruction in such a way that the number of instructions supported can be increased?

7. Consider two processor designs that implement the same ISA, P1 and P2. Assume there are two instructions in the ISA, namely I1 and I2. The throughput is measured in terms of clocks per instruction (CPI). P1 takes 2 and 4 CPI for I1 and I2 respectively, while P2 takes 3 and 3 CPI for I1 and I2 respectively. P1 runs at a clock rate of 2.5 GHz, while P2 runs at a clock rate of 2GHz. Assume a program with 30% of I1 instructions and 70% of I2 instructions.

(3 marks)

a. Find out the global CPI for each implementation.
b. Find the clock cycles required in running the program in both cases.

8. Analyze the complexity of the following algorithms based on add operations for n bit input

(2 marks)

a. Best and Worst case complexity of booths algorithm
b. Wallace tree multiplication