Indian Institute of Technology, Ropar

FALL, 2012

COMPUTER ARCHITECTURE

Midterm Examination

(TIME: 2 Hours)

NOTE: All answers need to be brief and to the point.
Please make any assumptions that you deem to be reasonable.
Total Marks: 50
Number of Pages : 3

Easy

1. Answer the following. (No Part Marking)
Just state the answers. You do not have to prove your results. (10 marks)

(a) What will happen if the number of processors in Amdahl's Law approaches infinity?

(b) What is the largest positive floating point denormal number?

(c) What is the asymptotic complexity of the Wallace Tree Multiplier? State your answer in terms of n. n is the number of bits used to represent both the multiplier and multiplicand.

(d) What is the asymptotic complexity of the non-restoring algorithm for division? State your answer in terms of n. n is the number of bits used to represent both the divisor and dividend.

(e) Create a one to one mapping between the sets (IPC, Frequency, #insts) with (compiler, technology, architecture).

2. Convert the following C program to ARM assembly. Store the integer, i, in register $R0$. Assume that the starting address of array A is saved in register $R1$, and the starting address of array B is saved in register $R2$. (5 marks)

```
int i;
int B[500];
int A[500];
for(i=0; i < 500; i++) {
      B[i] = A[A[i]];
}
```

Medium

3. Consider two designs $\mathcal{D}_1$ and $\mathcal{D}_2$. Their instruction mix and CPI per instruction type is given in Table 1. The total number of instructions are the same. The frequency of $\mathcal{D}_1$ is 1.5 GHz, and the frequency of $\mathcal{D}_2$ is 1.2 GHz. Compute $Performance(\mathcal{D}_1)/performance(\mathcal{D}_2)$. Write each and every step in your derivation cleanly. Mention all the formulae that you are using and justify all of your computations. (4 marks)

Operation	Fraction	CPI in $\mathcal{D}_1$	CPI in $\mathcal{D}_2$
Branch	20	2	1.5
ALU	40	1	2
Load	30	3	4
Store	10	2	1

Table 1: CPI and Instruction Types

4. Consider the floating point number given by Equation 1.

$$N = -1 \times \left(1.75 \times 2^{-29} + 2^{-40} + 2^{-45}\right) \tag{1}$$

Let us represent this number using the 32 bit IEEE 754 format. What is the hexadecimal representation of this number. Show the exponent and mantissa calculation in detail and explain your steps. Clearly indicate your final answer. (5 marks)

5. Let us consider two numbers, u, and v ($u, v \geq 1$). Here is the core of the algorithm to compute their gcd.

$$gcd(u,v) = \begin{cases} 2 \times gcd(\frac{u}{2}, \frac{v}{2}) & \text{both are even} \\ gcd(\frac{u}{2}, v) & \text{u is even and v is odd} \\ gcd(u, \frac{v}{2}) & \text{u is odd v is even} \\ gcd(\frac{|u-v|}{2}, v) & \text{both are odd} \end{cases} \tag{2}$$

Now, assuming that u is given in register $R0$, v is given in register $R1$, write an ARM assembly program to compute the gcd of the numbers, and place the result in register $R7$. You need to use ARM assembly instructions only. You are not allowed to use external functions including division and mod functions. Neatly write the ARM code with an adequate amount of spacing, indentation, and comments. (5 marks)

6. Answer the following:

(a) Write the smallest possible ARM assembly program to load the constant $0xEFFFFFF2$ into register $R0$. (2 marks)

(b) Write the smallest possible ARM assembly program to load the constant $0xFFFD67FF$ into register $R0$. (2 marks)

(c) We know that floating point representation is approximate. Let us consider two real numbers A and B. The closest numbers in the IEEE 754 format are A' and B' respectively. The error in representation is given by $\mid A - A' \mid$ and $\mid B - B' \mid$ respectively. Comment on the total error of the product $A'B'$ as compared to AB. How is it related to the individual values of error for A and B ? (2 marks)

Hard

7. Consider an n bit binary number. Find the fastest possible algorithm in hardware to find out if it is divisible by five. Clearly explain the following.

- Mathematical foundations
- Operation of the algorithm
- Evaluation of the asymptotic complexity

(5 marks)

8. We wish to compute the square root of a floating point number in hardware. You need to find the fastest possible algorithm for it. You need to elaborate on the mathematical foundations, operation of the algorithm, and the asymptotic complexity. The student is free to choose the algorithm.

We outline a part of one algorithm here that might potentially lead to a solution. Let us use the Newton-Raphson method similar to the one that was used to compute the reciprocal of a FP number. Note that the FP reciprocal algorithm to compute $\frac{1}{b}$, uses two functions. One is the objective function, $f(x) = \frac{1}{x} - b$, and the other is the error function, $E(x) = bx - 1$. Here are the steps of the proposed algorithm for the square root.

- Find an appropriate objection function.
- Find the equation of the tangent, and the point at which it intersects the x-axis.
- Find an error function (HINT : $E(x) = x - \sqrt{b}$).
- Calculate an appropriate initial guess for x.
- Prove that the magnitude of the error is less than 1.
- Prove that the error decreases at least by a constant factor per iteration.
- Evaluate the asymptotic complexity of the algorithm.

(5 marks)

Research

9. Assume that we want to create a computer that has just one instruction in its instruction set. We want this single instruction to be powerful enough to represent all possible programs that can be represented by C or ARM assembly. What is this instruction? Prove that it is equally powerful as C or ARM assembly. Try to make the instruction as simple as possible. (5 marks)