Indian Institute of Technology, Delhi

FALL, 2012

CSL 211: COMPUTER ARCHITECTURE

Minor 2

One and Half Hours

NOTE:
- All answers need to be brief and to the point.
- Please make any assumptions that you deem to be reasonable.
- We need exquisite detail, and clarity in every answer.
- You need to write your answer in the answer box (Ans:). We will **NOT GRADE** your answer, if it is not written in the box.
- Every answer needs to be written neatly and cleanly in the space provided for it.
- Use proper handwriting, and do not write anything in the margins.
- Note that in most questions, there is no part marking.
- You can use rough sheets. Do not attach them with this paper.
- You are not allowed to carry any electronic gadget including calculators.
- This question paper **NEEDS TO BE SUBMITTED**. Do not take it with you.

Total Marks: 50
Total Number of Pages : 8

Name:	Group No:	Entry No:

Marks:	1	2	3	4	5	6	7	8	Total

Easy

1. Answer briefly (1-2 lines) ***NO PART MARKING***

(a) Why don't we require forwarding from the WB to the ID/RF stage?
The WB stage writes its result in the earlier half of the clock cycle, and the ID/RF stage reads its result in the latter half of the clock cycle. The values are thus transferred through the register file

(b) What is the bias for double precision numbers? Ans: 1023

(c) What is the smallest denormal number? Ans: 2^{-149} - 2^{-126}

(d) How many unique denormalized numbers are there in the 32-bit IEEE 754 format Ans: 2^{24} - 2

(e) If I have two branch predictors – predict always taken, and predict always not taken – which one is expected to have higher accuracy for generic programs? Ans: always taken

(10 marks)

2. Consider the table given below. Here the *Base CPI* is calculated by assuming 100% branch predictor accuracy. Assume that 30% of the instructions are branch instructions. ***NO PART MARKING***

Design	Branch Predictor Accuracy	Base CPI	Branch Penalty	Processor Frequency
D_1	80%	1.05	2	1
D_2	90%	1	3	1.5
D_3	95%	1.5	6	2

What is the CPI of:

D_1	Ans: 1.17
D_2	Ans: 1.09
D_3	Ans: 1.59

What is the performance of D_2 and D_3 relative to D_1:

D_1	Ans: 1
D_2	Ans: 1.61
D_3	Ans: 1.47

(6 marks)

Medium

3. (7 marks)

(a) Give the simplest possible algorithm to compare two 32 bit IEEE 754 floating point numbers. Do not consider $\pm\infty$ and NAN. (3 marks)

Compare them as regular integers. (3) Default comparison (if-else based) (1)

(b) Prove it. Every step of the proof must be described in exquisite detail. (3 marks)

See that the mantissa and exponents are compared properly. Based on answer to question (a)

(c) Derive its complexity. (1 mark)

O(log(n)) if they have used a carry lookahead adder. Based on answer to (b)

4. Convert the following floating point numbers into the IEEE 32 bit 754 format. Write your answer in the hexadecimal format. (4 marks)

NO PART MARKING

-1 * (1.75 * 2^{-29} + 2^{-40} + 2^{-45})	Ans: 0xB160 1080
52	Ans: 0x 4250 0000

5. (6 marks)

NO PART MARKING

(a) Assume that 20% of the dynamic count of the instructions executed on a computer are branch instructions. Delayed branching is used with one delay slot. Estimate the CPI if the compiler is able to use 85% of the delay slots. Assume that the *base* CPI is 1.5. In the *base* case, we do not use any delay slot. We stall the pipeline for the total number of delay slots. (3 marks)

CPI	Ans: 1.33

(b) Now, assume that there are two delay slots. The compiler is able to fill the first delay slot 85% of the time, and the second slot the 20% of the remaining time. Estimate the new CPI. The base CPI is 1.5. (3 marks)

CPI	Ans: 1.324

6. Let the total algorithmic work be A. Let the pipeline have k stages, and let the latch delay be d. Let the cost of implementing a pipeline be $\alpha k + \beta$. Assume that we want to minimize the objective function: $minimum\_clock\_cycle\_time \times cost$. (5 marks)

(a) What is the optimal number of stages in the pipeline? Ans: $\sqrt{\frac{A\beta}{\alpha d}}$ (2 marks)

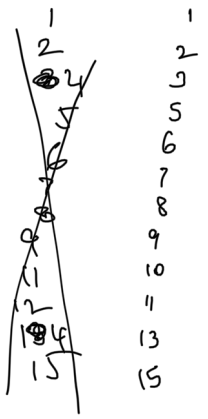
NO PART MARKING

(b) Derive and justify your answer. Every single step must be shown in great detail. (3 marks)
Formulation of clock cycle time, product, differentiation

Hard

7. *NO PART MARKING*

```
add r1, r2, r3
sub r4, r1, r6
lw r5, 10(r4)
add r6, r5, r5
sub r8, r8, r9
mul r10, r10, r11
beq r8, r10, label
add r5, r6, r8
sw r3, 20(r5)
lw r6, 20(r5)
lw r7, 20(r6)
slt r7, r7, r10
```



(6 marks)

(a) Assuming a traditional MIPS pipeline, how many cycles does this code take to execute? Assume that time starts when the first instruction reaches the WB stage. This means that if I had just one instruction, then it would take exactly 1 cycle to execute (Not 5). Assume that there is forwarding only between the WB and ID stage. Otherwise, forwarding is turned off. Assume that the branch evaluates to not taken, and there are no delayed branches. Furthermore, branches are evaluated in the EX stage. Ans: 28 (2), 29 (1) , 27(1) (2 marks)

(b) How many cycles will this code take to execute, if we have forwarding turned on, and we do not have delayed branches? Ans: 17(2), 16(1), 18(1) (2 marks)

(c) How many cycles with this code take to execute, if we have forwarding and delayed branches? In this case, assume that the two instructions after *beq* occupy the delay slots. Ans: 15(2), 14(1), 16(1) (2 marks)

Research

8. Assume that we want to design a pipeline that will have the IF, ID, and WB stages conceptually similar to the MIPS pipeline. However, we want to change the behavior of the RF, EX, and MEM stages, and possibly add many more stages in between. Let us now make the following assumptions:

(a) The memory latency differs for different instances of load-store instructions. It can vary between 1 to 100 cycles.

(b) If an instruction is independent of other instructions that are already there in the pipeline, then we should be able to execute it. For example, consider the sequence:

```
lw r1, 20 (r2)
add r4, r5, r6
add r10, r11, r4
add r20, r9, r10
```

If the load instruction gets stuck in the MEM stage, then we should still be able to execute all the add instructions.

(c) Instructions exit the WB stage in the same order that they entered the IF stage.

(d) You can execute upto n instructions in parallel. There should not be any structural hazards if less than or equal to n instructions are in the EX stage or in the MEM stage in the same cycle.

(e) Assume that the branch predictor has 100% accuracy.

(f) Do not assume any compiler support.

Design a machine according to these assumptions. We wish to maximize the IPC. (6 marks)

NOTE

- Do not give a very low level description of the data path and control path.
- We want to see a high level view of the design, description of some key structures that you use, and the way they work.
- Evaluation is on a case-by-case basis. We are looking for:
  - The high level structures.
  - A broad view of their operation including the computation and messages exchanged.
  - Agreement with the assumptions.
  - Efficiency and simplicity.
  - Please attempt this question, only if you think that you have something close to the correct answer and a precise design in mind. The bar is very high.
  - The final discretion lies with the evaluator.