Birla Institute of Technology and Science – Pilani, Hyderabad/Goa Campus
Second Semester 2018-19
CS F342: Computer Architecture Mid-Term Test (Closed Book)

Time: 4:00PM – 5:30PM **Date:** 15th March 2019 **Max. Marks:** 60

Note: Answer all the questions in the same sequence.

1. The table below shows the instruction type breakdown and the CPI of each instruction type of a given application running on a 3GHz StoneBear 438 processor. Assume the application has a total of 2×10^9 instructions.

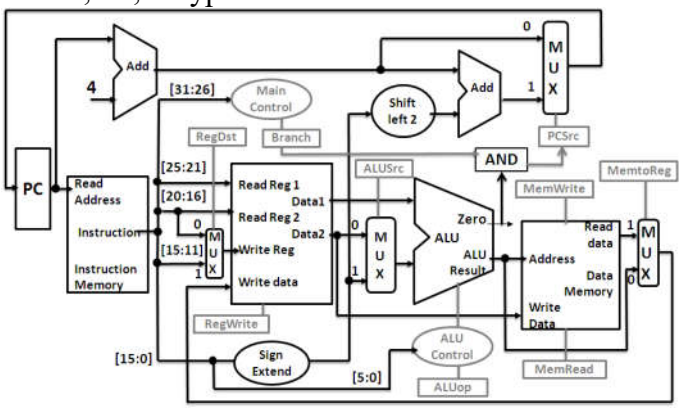
Instruction Type	Percentage of Instruction Count	CPI
Integer	55%	1
Load/Store	30%	3
Branch	15%	4

(i) What is the total execution time of the given application?

(ii) A revised version of StoneBear 438 processor raises the clock rate to 4GHz, but also increases the CPI of Load/Store instructions to 12. What's the speedup of this revised version vs the original? **[8]**

2. We want to perform an operation called **bcp**, which stands for block-copy where a sequence of words are copied from one portion in memory to another. Assume $t1 has the starting address of the destination, $t2 has the starting address of the source, and $t3 has the number of words to be copied. Given that contents of the three registers need not be preserved, Write the assembly code to translate this pseudo-instruction to an appropriate block of MIPS assemble code. Your code must be as efficient (not more than 8 instructions) as possible, and commented appropriately. **[10]**

3. Data path and control for a lw, sw, R-type and Branch instructions of MIPS is shown in the figure.



The design has to be modified to support a new instruction **beqzr reg1, reg2** (register indirect conditional branch if zero) which is not part of MIPS ISA. The example usage of this instruction is as below:

beqzr $t2, $s0;
L1:

This instruction checks if **$t2** (in this example) contains zero. If (**$t2==0**) then the instruction branches to an address which is specified in the register **$s0** (in this example), else next instruction i.e. instruction at L1 is executed.

(a) Determine and explain a suitable instruction code format for supporting **beqzr** instruction.

(b) Draw the datapath (digital blocks, additional control signals) that is required to support the **beqzr** instruction in simplest manner possible. (Draw only **extra datapath blocks** and show the additional control signals. For the extra datapath blocks clearly mention from which block(s) are the inputs coming and to which block(s) are the outputs going). **[12]**

4. Consider a hypothetical (not MIPS) instruction pipeline with five stages without any branch prediction. The five stages are Instruction Fetch (IF), Instruction Decode (ID), Fetch Operand (FO), Execute Instruction (EI) and Write Operand (WO). The stage delays for IF, ID, FO, EI and WO are 5 ns, 7 ns, 10 ns, 8 ns and 6 ns, respectively. There are intermediate storage registers after each stage and the delay of each register is 1 ns. A program consisting of 10

instructions I1, I2... I10 is executed in this pipelined processor. Instruction I4 is the only branch instruction and its branch target is I9. The branch is Taken during the execution of this program.

(i) Draw the pipeline diagram/layout with respect to time for the above program. (your answer should be in tabular form as shown in example below)

Instruction	Clock Cycle 1	Clock Cycle 2	Clock Cycle 3	Clock Cycle 4	Clock Cycle 5	……..			
I1	IF	ID	FO	EI	WO				
I2		IF	ID	FO	…	….			

(ii) Compute the time (in ns) required to complete the program.**[13]**

5. Assume that following is the outcome of a particular branch when it was executed within a loop:

Table 1

Branch Occurrence	1	2	3	4	5	6	7	8	9	10	11	12
Outcomes	T	T	T	NT	T	T	T	NT	T	T	T	NT
Predictions	?	?	?	?	?	?	?	?	?	?	?	?

For the above outcomes:

(i) Determine the predictions and percentage of mis-prediction for a static predictor which always predicts “Branch Taken”?

(ii) Determine the predictions and percentage of mis-prediction for a static predictor which always predicts “Branch Not Taken”?

(iii) Draw the FSM for a 2-bit dynamic branch predictor (such that the prediction changes from NT to T or T to NT only if two successive predictions are wrong). Determine the predictions and percentage of mis-prediction for a 2-bit dynamic predictor which is initialized to Strongly Not-Taken?

(iv) Assume that in a typical execution, 80% of your instructions are branches that follow the above outcome pattern. Currently there is a Static “Branch Not Taken” predictor in your machine. In order to optimize the program, you can either change the predictor to a 2-bit predictor or you can change the code and reduce the number of branch instructions by half. Compute the speedup for each alternative. Which optimization is the best choice?**[17]**
(The predictions for different branch predictors should be written in tabular from as shown in Table 1)