Birla Institute of Technology and Science – Pilani, Hyderabad/Goa Campus
Second Semester 2019-20
CS F342: Computer Architecture Mid-Term Test (Closed Book)

Time: 9:00AM – 10:30AM **Date:** 6th March 2020 **Max. Marks:** 60

Note: Answer all the questions in the same sequence.

1. Server farms such as Google and AWS provide enough compute capacity for the highest request rate of a day. Imagine that most of the time these servers operate at only 60% capacity. Assume further that the power does not scale linearly with load; that is, when the servers are operating at 60% capacity, they consume 90% of the maximum power (as the remaining servers consume power even in idle state). The servers could be turned off while in idle state, but they would take long to restart in response to more load. A new system has been proposed that allows for a quick restart but requires only 20% of maximum power while in this "barely alive" state

(i) What % of the maximum power will be consumed if 40% of the servers are turned off?
(ii) What % of the maximum power will be consumed if 40% of the servers are placed in "barely alive state"?
(iii) What % of the maximum power will be consumed if 20% of the servers are in "barely alive state" and 20% of the servers are turned off? **[2+3+3]**

2. Analyze the RISC V assembly program shown below and determine the array values stored at location "Num" after the execution of the program for **(i) N = 0x02 (ii) N = 0x05 and (iii) N = 0x08** (Fill in the values of the array after execution of assembly code in the tabular format shown, for three cases separately. No partial marking. Marks will be given only if all the entries of Num for given N are correct) **[2+6+8]**

```
.globl main
.data
Num: .word 0x45, 0x20, 0x32, 0x10, 0x42, 0x06, 0x59, 0x23
N: .word ???
.text
main:  lw t1, N
       srai t2, t1, 1
       add s4, t2, zero
       add s5, t2, zero
       slli t2, t2, 2
L4:    la t0, Num
       add t3, t2, t0
       beq t3, t0, Exit
       addi s5, s5, -1
       beq s5, zero, Exit
       add s4, s5, zero
L3:    lw s0, 0(t0)
       lw s1, 4(t0)
       lw s2, 0(t3)
       lw s3, 4(t3)
       bge s0, s1, L1
L5:    bge s2, s3, L2
L6:    addi t0, t0, 4
       add t3, t2, t0
       addi s4, s4, -1
       bne s4, zero, L3
       jal zero, L4
Exit:  ecall
L1:    sw s0, 4(t0)
       sw s1, 0(t0)
       jal zero, L5
L2:    sw s2, 4(t3)
       sw s3, 0(t3)
       jal zero, L6
```

Before execution:

Num:

0x45
0x20
0x32
0x10
0x42
0x06
0x59
0x23

After execution: **N= 0x02 / 0x05 / 0x08**

Num:

3. In the RISC V assembly program given in **Question 2**, assume that the five branches are at the locations as given in the table below:

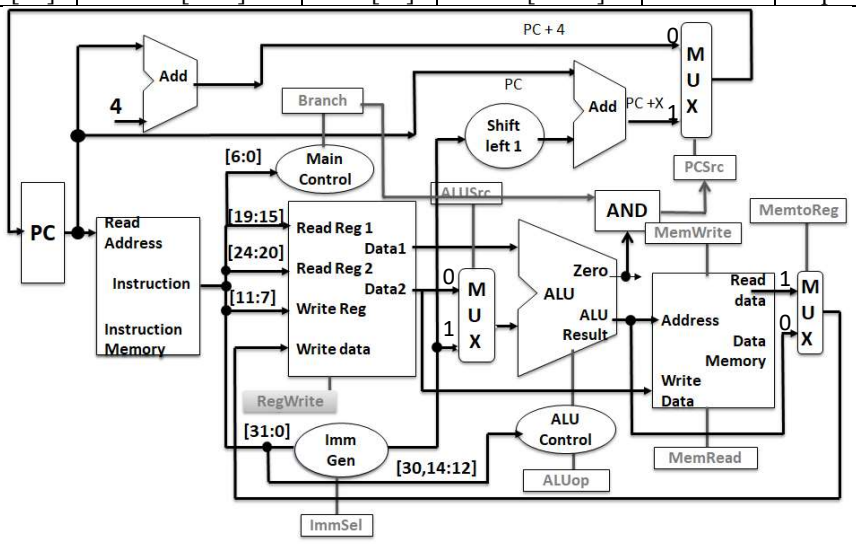
Branch	Instruction	Instruction Address location
1	**beq t3, t0, Exit**	0x00400024
2	**beq s5, zero, Exit**	0x0040002C
3	**bge s0, s1, L1**	0x00400044
4	**bge s2, s3, L2**	0x00400048
5	**bne s4, zero, L3**	0x00400058

Assume that the processor executing the program in **Question 2** uses a **4-entry 2-bit saturation counter (with all entries initialized to Strongly Taken state S3→11)** for branch prediction. As last two bits (0^{th} and 1^{st} bit) of all the instruction address are 0s, the 3^{rd} and 2^{nd} lower significant bits are used to access the 4-entry branch prediction table. For **N = 0x08,** determine the branch outcomes and predictions for branches 4 and 5, i.e **bge s2, s3, L2@** 0x00400048 and **bne s4, zero, L3 @**0x00400058. (Your answer should be in tabular format as shown below. Your answer will be considered for partial marks, only if all the branch outcomes are correct) **[8+8]**

Branch occurrence **bge s2, s3, L2@** 0x00400048 / **bne s4, zero, L3 @**0x00400058	1	2	3	……
Present State	**11**	??	??	
Prediction	**T**	??	??	……
Outcome	??	??	??	……
Next State				

4. Data path and control for lw, sw, R-type and Branch instructions of RISC V is shown in the figure below. The design has to be modified to support **jal** (Jump and link) instruction which is part of RISC V ISA. The instruction format for **jal** instruction is given below.

Bits	31	30:21	20	19:12	11:7	6:0
Field	Imm[20]	Imm[10:1]	Imm[11]	Imm [19:12]	Rd	Opcode (1101111)



(i) With the help of **jal** instruction format, find the instruction code (in hexadecimal format) for instruction **jal zero, L5** in the RISC V assembly program given in **Question 2**. (Assume the instruction code for **ecall** requires 32-bits and entire assembly code is stored consecutively in memory. The assembly code from **L1:sw s0, 4(t0) to jal zero, L6 is stored after ecall** consecutively). **[3]**

(ii) Draw the datapath (digital blocks, additional control signals) that is required to support the **jal** 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). Also list the values of all control signals (except ImmSel) while executing **jal** instruction. (You don't have to show the changes required within the Imm Gen block. Assume that Imm Gen block modifies the immediate field as required for **jal**) **[9]**

5. The assembly code given below is run on 5-stage pipelined processor.

```
add t0, zero, 4
add t1, t0, s0
lw t2, 0(t0)
add t4, t2, t3
sw t4, 0(t1)
```

Rewrite the assembly code after resolving the data hazards by adding least number of **nop** instructions, when assembly code is run on a processor (i) with no forwarding and (ii) with forwarding allowed to input of EX stage from output of EX stage only (forwarding to input of EX from output of MEM stage is not allowed). **[4+4]**