Advanced Computer Architecture (CS G524)
Semester II, 2022-23
Comprehensive Examination
Department of Computer Science and Information Systems (CSIS)
BITS-Pilani, K K Birla Goa Campus, Goa, India.

May 8, 2023: 2 pm Duration: 3 hrs # of Questions: 6 Marks: 90 [10 + 10 + 18 +10 + 38 + 4]

Instructions: **Attempt all the questions. No marks will be awarded for the questions if no reasoning is found in the answer script. You must continuously write all answers to a question [not here and there within the answer script].**

1. We want to add the swap instruction, ***swap rs, rt***, in the Multi-Cycle MIPS microprocessor. The interpretation of the instructions is `Reg[rs] = Reg[rt]; Reg[rt] = Reg[rs].`

(a) Show the modification in the microprocessor's datapath. More precisely, only the modified part, **not the entire datapath**.

(b) Show the modification in the controller, i.e., only the table for the new instructions.

(c) Compute the ***swap rs, rt*** using other instructions of MIPS ISA.

[**Marks**: 4 + 4 + 2]

2. (a) In pipelined MIPS microprocessor, what will happen if the write operation to the register file of the R-type instruction (ADD/SUB) happens from the EXE stage? Explain it with an example.

(b) Write down the dependency detection condition clearly [with diagram] for data hazard in the 5-stage pipelined MIPS processor and find out the data dependency among the following code

```
sub $2, $1, $3
and $12, $2, $5
or $13, $6, $2
add $14, $2, $2
sw $15, 100($2)
```

(c) Rearrange the above-mentioned code considering there is no hazard detection in the 5-stage MIPS pipeline microprocessor, including no-operation instruction.

[**Marks**: 2 + (4+2) + 2]

3. (a) Find a string where the prediction accuracy for a 1-bit predictor is 50% and for a 2-bit saturation counter, with state's change an only single bit and the initial state is NT (01), is 0%, in steady-state. Can you think of a programming pattern that fits this type of string?

(b) Write the C code for *int getPrediction()* and *void updatePred()*, the *n-bit* branch predictor, assuming branch traces come from a file, *btrace.txt*. The trace format is B_PC<\t><1/0>, and each trace is started in a new line. B_PC is the PC value of the branch instruction, <\t> is a tab, and '1' is if the branch decision is taken; otherwise, '0'. The *getPrediction()* returns a '1' or '0', and *updatePred()* returns nothing but updates the predictor. Choose the arguments of the functions properly.

[**Marks**: (3 + 3) + 4 + 8]

4. (a) CPI equation can be used to model the performance of in-order processors with multilevel cache hierarchies. Compute the CPI for such a processor with the following parameters:
Infinite cache CPI of 1.15, L1 cache miss penalty of 12 cycles, L2 cache miss penalty of 50 cycles, L1 instruction cache per-instruction miss rate of 3%, L1 data cache per-instruction miss rate of 2%, and L2 local cache miss rate of 25%.

(b)Consider a processor with a 32-bit virtual address, 4K-byte pages, and a 36-bit physical address. Assume memory is byte-addressable (i.e., the 32-bit virtual address specifies a byte in memory).
L1 instruction cache: 64K bytes, 128-byte blocks, four-way set-associative, indexed and tagged with the virtual address.
L1 data cache: 32K bytes, 64-byte blocks, two-way set-associative, indexed and tagged with the physical address, writeback.
Four-way set-associative TLB with 128 entries in all. Assume the TLB keeps a dirty bit for each entry.
Find out the number of offset bits, index bits, tag bits, size of the tag array, and size of the data array for L1-I, L1-D, and TLB cache.

[**Marks**: 4 + 6]

5. (a) What's the memory wall problem? How can we minimize it?

(b) Is the tag available in the MIPS register file? Justify your answer.

(c) What's the primary miss? Describe the MSHR's different fields briefly.

(d) Consider the following code and a virtually-addressed direct-mapped cache of capacity 8K-byte and 64-byte blocks. Consider that all variables are stored in the registers, and the arrays, A, B, and C, are placed consecutively in memory. Calculate the cold and conflict misses and show all cache snapshots clearly.

```
double A[1024], B[1024], C[1024];
for ( int i=0; i < 1000; i +=2 )
    A[i] = 35.0 * B[i] + C[i+1];
```

(e) Consider a cache with 256 bytes. The word size is 4 bytes, the block size is 16 bytes, and the replacement algorithm is LRU. Show the values in the cache and tag bits after each memory access operation for the two-way associative cache memory. Also, mention whether the access was a hit or miss. Justify. The address is presented in hexadecimal.

1. Read 0010, 2. Read 001C, 3. Read 0018, 4. Write 0010, 5. Read 0484, 6. Read 051C, 7. Read 001C, 8. Read 0210, 9. Read 051C

(f) Why is it called "pseudo" for PLRU cache replacement policy? Justify it with an example of a 3-bit tree PLRU.

(g) Consider the problem of **5(d)** for fully-associative cache with a tagged prefetcher, the LRU replacement policy, and loop condition, $i < 200$. Find out the cache misses.

[**Marks**: 2 + 2 + (1+3) + 8 + 5 + 5 + 12]

6. Consider a 16 GB DIMM, consisting of x8 DRAM devices of capacity 4 Gb. How many ranks are present on this DIMM if the channel (for transferring the data from main memory to cache and vice-versa) width is 64? Show all the calculations.

[Marks: 4]

-: END :-