Computer Architecture (CA)

## Final Term Fall 2018

## Total Marks: 50 Time Allowed: 180 Minutes

## Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Registration Number: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Instructions:**

1. **Solve paper on answer sheet provided.**
2. **This is a close book close notes exam. Cheat sheet is not allowed.**
3. **Attempt all questions.**
4. **Show your work clearly to get full credit**.

**Question #1: [5]**

In Memory hierarchy, when a block of information is not found in Cache, CPU has to fetch that block from main memory. How does Cache improve the overall system performance, if the CPU has to fetch the block from the main memory when the block is not available in Cache?

**Question #2: [7+8]**

**LRU (Least recently used)** and **LFU** **(least frequently used)** are two well-known Cache replacement algorithms which defines the block to be replaced in Cache. You are required to give a **comparison of these two algorithms (in terms of Hit Rate)** when the main memory has storage capacity of 128 bytes (data given in Table-1) and Cache memory has data storage capacity of 8 bytes (initially empty). **Compare the results by drawing the Cache memory table and filling the data in Cache** when Cache is mapped as **2-way set Associative** for **block size 1-byte** and CPU generates addresses: 0, 1, 2, 3, 127, 126, 125, 124, 0, 64, 10, 20, 41, 126. **Clearly label the data that is replaced in Cache lines.**

**Table 1- Main memory block**

|  |  |
| --- | --- |
| Address | Data |
| 0000000 | A0 |
| 0000001 | A1 |
| 0000010 | A2 |
| 0000011 | A3 |
| **.** |  |
| **.** |  |
| **.** |  |
| **.** |  |
| . | . |
| 1111111 | A127 |

**Question #3: [10]**

Consider the main memory has storage capacity of 128 bytes (data given in Table-2) and Cache memory have data storage capacity of 16 bytes (initially empty). As a computer architect designer, you are required to **select an optimal Block Size** which will give **maximum Hit Rate** when Cache is mapped as **Fully Associative** for the given code. Justify your answer by drawing the table and fill-in the data, for each of the block size you chose for comparison. Give at least two such tables.

**Note: Consider the separate cache for data and instruction. Only calculate the hit rate of data cache.**

**Table 2- Main memory block**

|  |  |
| --- | --- |
| Address | Data |
| 0000000 | A[0] |
| . | . |
| . | . |
| 0001001 | A[9] |
| 0001010 | B[0] |
| **.** | . |
| **.** | . |
| 0010011 | B[9] |
| 0010100 | Sum[0] |
|  | . |
|  | . |
| 0011101 | Sum[9] |
| 0011110 | **i** |
| . | . |
| 1111111 | A127 |

Code Example:

char A[10], B[10], sum[10];

char i;

for(i=0; i<10; i++)

{

A[i]=B[10-i]

}

**Question #4: [20]**

**Consider the following set of instructions**

**I1: R0 R1 + 2**

**I2: R2 R1 \* 4**

**I3: R1 R2 AND R3**

**I4: R2 R4 + 1**

**I5: R0 R4 AND R5**

**I6: R5 R0 \* 2**

1. Identify the **data dependencies** in the above piece of code. Also write down the cases when **write after write (WAW**) **and write after read (WAR**) dependencies can occur in the above code fragment, **assume that (for this part only)** all instructions can be executed all at once if there is no dependency. **[5 marks]**
2. The above piece of code needs to run on a superscalar machine with following specifications:
   * Machine can Fetch/decode and write back 2 instructions at a time
   * The machine has 3 functional units with the following capabilities:
     + - Only 1 functional unit out of 3 can perform addition with a constant and addition takes 2 clock cycles
       - Only 1 functional unit out of 3 can perform “\*” and it takes 3 clock cycles.
       - Only 1 out of 3 functional units can perform “AND” operation and it takes 1 clock cycles.
       - 1 instruction of addition, 1 instruction of “\*” and 1 instruction of “AND” can execute in parallel.
     1. Assuming Write after Write (WAW) and Write after Read (WAR) dependencies are already resolved by the processor using register renaming technique, list down all the constraints (Read after write (RAW) or any structural hazard (functional conflicts) in the given code fragment? **[5 marks]**
     2. How many clock cycles will be required to complete the code fragment using **in-order issue** and **out-of-order completion**? Make a table of Decode, Execute and Write back stage for the code fragment. **[10 marks]**