

## **CSE 30321 – Computer Architecture I – Fall 2011**

### **Homework 08 – Multiprocessing – Please answer on this handout**

**Assigned:** November 22, 2011, **Due:** December 6, 2011, **Total Points:** 60

#### **Problem 1: (20 points)**

You are trying to decide whether or not (i) a slower, dual core processor or (ii) a faster, single core processor performs better for a given workload. The specifications and benchmark data relevant to each machine/core are given below.

#### **Dual core specs:**

To execute your application of choice on the 2 GHz dual core machine, you will need to execute 10 billion total instructions:

- 30% of the instructions will run on Core 1
- 70% of the instructions will run on Core 2
- Execution will proceed in parallel.

Of the instructions that run on Core 1:

- 10% are branch instructions
- 70% are ALU instructions
- 20% of instructions reference memory

Of the instructions that run on Core 2:

- 10% are branch instructions
- 10% are ALU instructions
- 80% of instructions reference memory

On either core:

- A branch instruction requires 7 CCs to execute
- An ALU instruction requires 10 CCs to execute

On Core 1:

- A memory reference instruction requires:
  - o 7 CCs if there is an L1 cache hit
  - o 17 CCs if there is an L1 cache miss but data is found in the L2 cache
  - o 125 CCs if the memory reference misses in L1 and L2
- The L1 miss rate is 15%, and the L2 miss rate is 10%

On Core 2:

- A memory reference instruction requires:
  - o 7 CCs if there is an L1 cache hit
  - o 17 CCs if there is an L1 cache miss but data is found in the L2 cache
  - o 125 CCs if the memory reference misses in L1 and L2
- The L1 miss rate is 3%, and the L2 miss rate is 5%

#### **Single core specs:**

To execute your application of choice on the 3 GHz **single core machine**, you will also need to execute 10 billion total instructions:

- 10% are branch instructions
- 28% are ALU instructions
- 62% of instructions reference memory

On the single core machine:

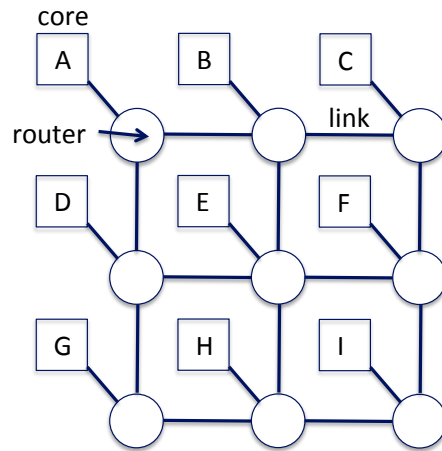
- A branch instruction requires 6 CCs to execute
- An ALU instruction requires 8 CCs to execute
- A memory reference instruction requires:
  - o 7 CCs if there is an L1 cache hit
  - o 20 CCs if there is an L1 cache miss, but data is found in the L2 cache
  - o 155 CCs if the memory reference misses in L1 and L2
- The L1 miss rate is 15%, and the L2 miss rate is 2%

For the 10 billion instruction workload, which processor is better and by how much?

**Problem 2: (9 points)**

Assume that 9 processor cores are connected via a 3 x 3 mesh network (see below).

- Note: Circles are routers, squares (with letters) are cores



Assume that you want to send a 1200 bit message from Node A to Node I. The total bandwidth between nodes/routers is 32 bits per link. Every packet will traverse the same path. Assume that the time spent in the links between routers is 2 CCs. The total time required to send the 1200 bit message is 66 CCs. How much time is spent in a router? (You may assume that each router has the same latency.)

**Problem 3: (18 points)**

Consider the multiprocessor cache state shown in the tables below:

**CACHES**

**P0:**

Block Number	Coherence State	Address Tag	Data	
Block 0	Shared	200	01	02
Block 1	Modified	220	03	04
Block 2	Shared	210	05	06
Block 3	Invalid	204	07	08

**P1:**

Block Number	Coherence State	Address Tag	Data	
Block 0	Shared	200	01	02
Block 1	Modified	150	09	10
Block 2	Shared	700	11	12
Block 3	Modified	204	13	14

**P2:**

Block Number	Coherence State	Address Tag	Data	
Block 0	Shared	200	01	02
Block 1	Shared	300	15	16
Block 2	Shared	700	11	12
Block 3	Invalid	204	07	08

(note that you will only need to consider cache state for these 3 processing nodes.)

Given the initial state shown above, explain what happens to cache block state given the events described below. You should treat each **part** below (there are 3 in all) independently (i.e. as you begin a new part, as a starting point, you should assume the initial state given above).

Describe any changes that occur in a given node for the listed event, any generated bus traffic, etc. (Note that in some cases, some cache blocks in a given node may not change at all, no bus traffic may be generated, etc.) **You should assume an MSI protocol and a centralized shared memory machine for this problem.**

**Part 1**

Question A:

P0 requests data from memory with address tag 100. The data maps to Block 0. What happens?

Question B:

P0 writes to Block 2. What happens?

## **Part 2**

### Question A:

Assume a new node – P3 – is introduced. It too has a cached copy of data associated with address tag 200 in the shared state, and now does a write to a word within the block. What happens?

### Question B:

P1 now requests data from memory with address tag 500. This data would map to Block 0. What happens?

### Question C:

P1 now requests data from memory with address tag 400. This data would map to Block 3. What happens?

## **Part 3**

### Question A:

P2 requests data from memory with address tag 150. What happens? (You do not have to comment on what block 150 might map to for this question.)

(There is no Question B.)

**Problem 4: (18 points)**

Consider the multiprocessor cache state shown in the tables below:

**CACHES**

**P0:**

Block Number	Coherence State	Address Tag	Data	
Block 0	Shared	200	01	02
Block 1	Modified	220	03	04
Block 2	Exclusive	210	05	06
Block 3	Invalid	204	07	08

**P1:**

Block Number	Coherence State	Address Tag	Data	
Block 0	Shared	200	01	02
Block 1	Modified	150	09	10
Block 2	Shared	700	11	12
Block 3	Modified	204	13	14

**P2:**

Block Number	Coherence State	Address Tag	Data	
Block 0	Shared	200	01	02
Block 1	Exclusive	300	15	16
Block 2	Shared	700	11	12
Block 3	Invalid	204	07	08

**Repeat Question 3 – but now assume an MESI protocol.**

**Part 1**

Question A:

P0 requests data from memory with address tag 100. The data maps to Block 0. What happens?

Question B:

P0 writes to Block 2. What happens?

## **Part 2**

### Question A:

Assume a new node – P3 – is introduced. It too has a cached copy of data associated with address tag 200 in the shared state, and now does a write to a word within the block. What happens?

### Question B:

P1 now requests data from memory with address tag 500. This data would map to Block 0. What happens?

### Question C:

P1 now requests data from memory with address tag 400. This data would map to Block 3. What happens?

## **Part 3**

### Question A:

P2 requests data from memory with address tag 150. What happens? (You do not have to comment on what block 150 might map to for this question.)

(There is no Question B.)