**Lec 2 Introduction**

Q1. Why does memory access time increases with the size of the memory for any type of the memory (cache, main memory, ...)?

A1. Access time is dominated by wire delays - larger memories have longer wires and therefore longer access time.

Q2. It possible to have speedup better than P where P is the number of available processors.

A2. Yes

Q3. In vector processors, does one instruction perform many operations?

A3. Yes

Q4. Why is it difficult to further increase the number of pipelining stages in the processors?

A4. Breaking the instructions up into stages of approximately equal length and designing hardware to implement these stages is difficult. The different tasks rarely take the same time. It also takes time to move data between stages and this overhead will increase the time it takes for a single instruction to execute. The stages require hardware to be able to work independent, to communicate and coordinate with the other stages. This size and complexity of this controlling hardware grows exponentially with the number of stages. This increase in complexity makes it even more difficult to increase the number of stages.

Q5. How is it possible to prevent against transient faults in memories?

A5. DRAMs and SRAMs usually have some form of error detection and correction capabilities.

**Caches and Parallel programming models**

Q1. How many bits per cache set are needed to support LRU replacement policy for 16-way set-associative caches? (here I mean the total number of bits needed for LRU for one set in all 16 lines)

A1. 64

Q2. What is prefetch degree?

A2. The number of blocks that are prefetched

Q3. We need to apply mutual exclusion when:

A3. Two processes write to the same memory location

**Caches and shared memory model**

Q1. What is the reason for using Barrier in parallel programming?

A1. To provide a point for synchronization for multiple threads

Q2. The sequence of 10 byte accesses for a direct-mapped cache with 4 blocks of 4 bytes is shown below. Accesses number 8, 9 and 10 will result in the following sequence of hits/misses:

A2.

Block size is 4 bytes and the memory is byte addressed, hence offset is the least significant 2 bits

Cache Size is 4 blocks with direct mapping, i.e. 4 cache lines, so index is the two bits in the middle

Tag is the 2 most significant bits

|  |  |  |
| --- | --- | --- |
|  | **Address (binary)** | **Hit/Miss** |
| 1. | 110001 | Miss |
| 2. | 100111 | Miss |
| 3. | 001111 | Miss |
| 4. | 001100 | Hit |
| 5. | 010001 | Miss |
| 6. | 110010 | Miss |
| 7. | 100101 | Hit |
| 8. | 001110 | Hit |
| 9. | 100001 | Miss |
| 10. | 110101 | Miss |

Answer: H, M, M

**Message passing and bus-based shared memory systems**

Q1. One of the architectures of bus-based shared memory systems relies on shared L2 caches. What is the reason for having shared L2 caches?

A1. This organization is used in multicore systems where the memory is outside of the chip. The goal is to reduce memory access latency gap between the first-level cache and the main memory.

Q2. In the following program assume that SEND takes 500 cycles to finish and unrelated computation takes 600 cycles. How long does it take to execute this program? Assume that both threads can send their messages at the same time?

**CODE FOR THREAD T1:                    CODE FOR THREAD T2:**

  A = 10;                                                        B = 5;

  ASEND(&A,sizeof(A),T2,SEND\_A);            ASEND(&B,sizeof(B),T1,SEND\_B);

   <Unrelated computation;>                            <Unrelated computation;>

  SRECV(&B,sizeof(B),T2,SEND\_B);              SRECV(&A,sizeof(B),T1,SEND\_A);

A2. 600

Q3. Is OpenMP language or Application Programming Interface (API)? If it is API, what languages is it specified for? Is it for shared memory or for message passing systems?

A3. API for C, C++ and Fortran for shared memory systems

**Quiz\_GPU**

Q1. Each kernel instance is a \_\_\_\_\_\_\_\_\_\_ or thread

A1. work item

Q2. What are the main benefits of OpenCL vs. CUDA? Select all that apply.

A2. OpenCl is made for heterogeneous computing platforms.

OpenCl is hardware independent.

Q3. How does the GPU hide memory latency?

A3. By spreading the lookup latency across a group of parallel threads.

**Snooping cache coherence implementation**

Q1. What is the purpose of wired-NOR bus line for the MESI protocol?

A1. To check if the data in read transaction is present in any other cache (Shared state). The ‘Shared’ signal is generated by a wired-NOR line driven by all processors with an open-drain and pulled up in the bus controller.

Q2. What are the information that have to be exchanged between two caches (L1 and L2) for the snooping protocol with multilevel caches? Assume that the caches are inclusive. Describe information exchange both in L1->L2 and L2->L1 directions.

A2. Snooping cache coherence control needs to take place in the L2 cache since the L1 cache is a subset of L2 and data is propagated explicitly.

Q3. Disadvantages of the MESI protocol in comparison with MSI protocol are (list all that apply):

A3. additional bus lines

more complex state machine

waiting for all processors to set the shared line before it is possible to change state after the read miss

**Directory based protocols**

Q1. The full map (or baseline directory protocol) directory can cause significant memory overhead. For a fixed size of the main memory, the overhead of the directory will increase if we increase the size of cache blocks.

A1. False

Q2. Consider a multiprocessing system with 8 processors that have their local caches and they are connected to the main memory.

Assume that Full Map Directory cache coherence protocol with Centralized Directory Invalidate is implemented. Assume that directory for address X contained all 0s at the beginning. Fill the following table for the following sequence of instructions:

|  |  |  |
| --- | --- | --- |
| Time instant | Operation | Content of the directory for X |
| 1 | Processor 0 – read X | [1] |
| 2 | Processor 5 – read X | [2] |
| 3 | Processor 0 – writes to X | [3] |

Content of the directory include only presence bits. LSB corresponds to the cache of processor 0 .

*Centralized Directory Invalidate protocol description:*

*Invalidating signals and a pointer to the requesting processor are forwarded to all processors that have a copy of the block. Each invalidated cache sends an acknowledgment to the requesting processor. After the invalidation is complete, only the writing processor will have a cache with a copy of the block.*

A2. 1. 00000001

2. 00100001

3, 00000001

**Virtualization and Networks**

Q1. What is the maximum degree of the switches in 4x4 mesh network?

A1. It is 5 if we consider the connection with the local processor. If we consider only connections with the other nodes then the degree is 4.

Q2. Why is hypervisor type 2 less efficient than type 1?

A2. Type 1 hypervisor runs on bare metal whereas the type 2 hypervisor runs on the operating system and every request for hardware from the guest system have to pass through the host system.

Q3. What are events in Xen?

A3. Virtual interrupts

**Interconnection network delays**

Q1. Consider 4x4 mesh network. Switching strategy is store-and-forward. The network clock frequency is 1 GHz and the phit size is 8 bits. The routing address occupies 2 phits (header of the packet), a payload of a packet is 128 bits. Assume that the sender and receiver overheads are zero and that wire delay is also zero. End-to-end network latency for sending a packet from the lower left to the upper right corner in nanoseconds is [1]. The minimum size buffer in each switch in bytes is [2].

A1.

[1] 108 / 107

[2] 18

**Interconnection network topologies**

Q1. Consider simple comparison between 16x16 Omega network and 16x16 crossbar network. While the crossbar uses cross points, the Omega network is using 2x2 switching elements (SE). Assume that the cost of the SE is four times that of a cross point. How many times is crossbar network more expensive than Omega network if we assume that the cost of the Omega network is determined only by its switching elements and the cost of the crossbar network is determined only by its cross points.

A1.

Cross points in 16x16 Crossbar =256

Switches in 16x16 omega = # of stages \* # of switches per stage = 4 \* 8 =32

Cost of crossbar network = 256\*1 = 256

Cost of Omega network = 32 \* 4 =128

Cost of Crossbar/Cost of Omega = 256/128 = 2

Q2. How many processor nodes Np and how many switches Ns are there in a 4-ary tree (4-ary means that each intermediate node has 4 children) which height is 3 (3 levels of links and 4 levels of nodes)?

A2.

Np=64

Ns=21

**Networks and Synchronization**

Q1. If reducing the amount of memories and buffers in a system-on-a-chip is a main design goal, which switching technique would you use?

A1. Wormhole

Q2. What techniques/transactions in bus-based networks can be used where fast and slow devices need to communicate?

A2. Split-transactions

Q3. Barrier is used to provide

A3. global synchronization among all threads

Q4. Diameter of 4x4 2D mesh, where each node is also connected to additional 3 nodes in the form of a linear array, is:

A4. 12

**Synchronization**

Q1. In case the Store Conditional (SC) instruction fails, bus will not be accessed. How is that achieved? Describe what component needs to be added and what actions need to be performed in order for LL-SC to work.

A1.

Initially, a load-linked/load-locked operation will place the value of a memory location into the cache. The store-conditional will act on the same memory location and will only change the value if it has remained unchanged since the load-linked/load-locked. The bus will not be in use as the store-conditional is only comparing the cache value and the store will not be taking place.

We need to support a hardware lock flag and address register for each processing element wishing to implement this SC operation. An initial load-linked/load-locked operation will set the flag and will place the memory address of the corresponding data value into the special lock address register. The snooper will look for changes to that address and if a change occurs, then the flag is reset. The flag is also reset if the matching cache block is swapped out or replaced.

Q2. Test\_and\_test&set causes larger number of invalidations in MSI invalidate protocol than test&set.

A2. False

**Multithreading**

Q1. What is the common switching delay in RISC-based blocked multithreading processors?

A1. The delay corresponds to pipeline depth.

Q2. Select resources that need to be replicated to support blocking multithreading implementation.

A2.

Register file

Program counter

**Multithreading1**

Q1. Will context switching take longer for the OoO superscalar processor than in-order in case of blocked multithreading implementation? Why?

A1. Yes, it will take longer. OoO processor has fetches more instructions so that it can analyze their dependencies before dispatching them. It will take more time to flush these instructions and all the stages of the OoO processor.

Q2. SMT increases the processor area approximately by

A2. 5%