

Question Mid-1: (Solution, p 2) What are the two basic models for communication in multi-processor systems?

Question Mid-2: (Solution, p 2) Name and explain two advantages of shared memory over message passing.

Question Mid-3: (Solution, p 2) Name and explain two advantages of message passing over shared memory.

Question Mid-4: (Solution, p 2) There are two basic ways of implementing snooping to support cache coherency in shared memory systems: write broadcast and write invalidate. What is the primary advantage of write invalidate over write broadcast?

Question Mid-5: (Solution, p 2) What is an advantage of having a single processor with several cores over having several separate processors?

Question Mid-6: (Solution, p 2) Consider the following sequence of instructions to compute $x^2 + 4x + 1$ for each element x in a vector stored in `$v0`.

```
multv $v1, $v0, $v0
multsv $v2, 4, $v0
addv $v3, $v1, $v2
addsv $v4, $v3, 1
```

Let n represent the length of the vector supported on our processor, where $n \geq 8$. Assume that we have a vector processor with one vector multiplication unit whose latency is 7 and one vector addition unit whose latency is 6. If the processor does not avoid latency in subsequent instructions, but it does support forwarding between instructions (a.k.a. chaining), how many clock cycles with this instruction sequence take?

Provide a better instruction sequence accomplishing the same thing but requiring fewer clock cycles.

Solution Mid-1: (Question, p 1) Shared memory and message passing.

Solution Mid-2: (Question, p 1)

- Higher communication bandwidth, usually, since as soon as information is stored in memory, it is available to other processor. By contrast, message passing systems must copy the message from one processor's memory to another (at least in a naive implementation).
- Easier to administer systems, since notionally the system works as one integrated whole.

Solution Mid-3: (Question, p 1)

- Message passing is simpler to implement and thus less expensive, because there is no need to support the implicit synchronization important to shared memory systems.
- Systems are more scalable, since we are not limited by the bandwidth to memory.

Solution Mid-4: (Question, p 1) Write invalidate's primary advantage is that it requires less traffic for multiple writes to the same memory location: With write broadcast, every write to that memory location must be broadcast onto the bus, but with write invalidate, only the first write need be broadcast on the bus. (Of course, the value must be broadcast later when the cache decides no longer to include that memory location, or when another cache needs to read from that memory location.)

Solution Mid-5: (Question, p 1)

- The multiple cores can share some resources, such as circuitry to support L2 cache and communication with outside devices, thus leading to less total up-front expense and less power usage.
- The memory bus, being implemented on-chip, can run much more quickly.

Solution Mid-6: (Question, p 1) $26 + 4n$. The following instruction sequence requires only $20 + 2n$ cycles, though:

```
multsv $v2, 4, $v0
addsv  $v3, $v2, 1
multv  $v1, $v0, $v0
addv   $v4, $v1, $v3
```

Even better is the following based on the identity $x^2 + 4n + 1 = (x + 2)^2 - 3$. This requires only $12 + 2n$ cycles.

```
addsv  $v1, 2, $v0
multv  $v2, $v1, $v1
addsv  $v4, -3, $v2
```