quarch

caches

Assume a program exhibits perfect temporal locality (e.g., program with memory blocks A, B and C accesses them in such a sequence: A, …, A, B, …, B, C, …, C). Which of the following types of cache misses can be observed when running this program?
i.Conflict misses
ii.Capacity misses
iii.Cold (compulsory) misses [answer]

bit serial arithmetic

Assume there are n bit-serial adders available in the hardware system. To perform an addition between two M-bit inputs, what is the theoretically minimum latency to perform the addition? Assume n>=1, M>0.
i. M/n
ii. M/((n-1)/2+1) [answer]
iii. 2M/n

cache coherence

Select all the stable state transitions that don't exist in corresponding cache coherence protocol.
i. S->E in MESI [answer]
ii. O->M in MOESI
iii. E->O in MOESI [answer]
iv. M->S in MOESI [answer]
v. S->F in MESIF [answer]
vi. I->S in MESIF [answer]

Assume a multi-core cache coherent system where memory latency is 60 cycles and cache-to-cache forwarding latency is 80 cycles, is running a program with extensive data sharing (both read and writes). Assume memory write is off the critical path and the network has high bandwidth. Which of the following protocol is most suitable in terms of latency for the system and workload?
i. MSI [answer]
i. MESI

ii. MOESI
iii. MESIF

cache consistency

Consider a program with four threads running on separate cores in a quad-core shared-memory multiprocessor. Assume that all memory contents are initially 0, and there are no private data caches in this system. Select the program outcome (i.e., set of register contents after executing all threads) that is impossible if the multiprocessor enforces sequential consistency.

```
# thread A
A0:  lui  $t0, 0xC000
A1:  addi $t7, $zero, 1
A2:  sw   $t7, 0($t0)     # (0xC0000000) <- 1
# thread B
B0:  lui  $t1, 0xC000
B1:  ori  $t1, $t1, 0x1000
B2:  addi $t7, $zero, 1
B3:  sw   $t7, 0($t1)     # (0xC0001000) <- 1
# thread C
C0:  lui  $t1, 0xC000
C1:  add  $t0, $zero, $t1
C2:  ori  $t1, $t1, 0x1000
C3:  lw   $s0, 0($t0)     # [$s0] <- (0xC0000000)
C4:  lw   $s1, 0($t1)     # [$s1] <- (0xC0001000)
# thread D
D0:  lui  $t0, 0xC000
D1:  add  $t1, $zero, $t0
D2:  ori  $t1, $t1, 0x1000
D3:  lw   $s2, 0($t1)     # [$s2] <- (0xC0001000)
D4:  lw   $s3, 0($t0)     # [$s3] <- (0xC0000000)
```

i. $s0 = 0, $s1 = 0, $s2 = 0, $s3 = 0
ii. $s0 = 0, $s1 = 0, $s2 = 0, $s3 = 1
iii. $s0 = 0, $s1 = 1, $s2 = 0, $s3 = 1
iv. $s0 = 1, $s1 = 0, $s2 = 0, $s3 = 0
v. $s0 = 1, $s1 = 0, $s2 = 1, $s3 = 0 [answer]
vi. $s0 = 1, $s1 = 0, $s2 = 0, $s3 = 1
vii. $s0 = 1, $s1 = 1, $s2 = 0, $s3 = 0

vii. $s0 = 1, $s1 = 1, $s2 = 0, $s3 = 1
viii. $s0 = 1, $s1 = 1, $s2 = 1, $s3 = 0

NoCs

Which of the following optimizations reduce packet serialization delay? Select all that apply.
i. Change topology to reduce network diameter
ii. Use larger links to increase link bandwidth [answer]
iii. Apply crossbar speedup to increase router switch bandwidth
iv. Shrink cache blocks to reduce number of flits per packet [answer]