Department of Computer Science & Engineering,
National Institute of Technology, Jamshedpur

Course Title and Code: **Advanced Computer Architecture** CS 4201 (M. Tech) and CS 602 (B. Tech)

Practice questions for **mid-semester** examination **Spring Semester 2020**

To solve the given questions please refer to the

Text Book

A. Advanced Computer Architecture: A Quantitative approach by John L. Hennessy and David A. Patterson, 6th edition 2019, published by Morgan Kaufman Publishers (an Imprint of Elsevier).

Reference Book

B. Computer Organization and Architecture: Designing for Performance by William Stallings, 10th edition 2016, published by PHI, New Delhi.

Sample Questions:

1. Illustrate how a cache management policy of giving priority to read misses over write misses reduces the miss penalty.
2. Illustrate the meaning of *forwarding* in a pipeline by drawing the pipeline stages of few instructions.
3. What is a precise exception? Cite situations where imprecise exceptions may occur in MIPS pipeline.
4. What measures are taken by compiler to reduce the stalls due to branch (control hazard)?
5. Compare the performance of a processor for direct mapped and two-way set associative cache organizations. Assume that the CPI with a perfect cache is 1.5, the clock cycle time is 0.20 ns, there are 1.6 memory references per instruction, the size of both caches is 256 KB, and both have a block size of 64 bytes. Assume the processor clock cycle time is stretched 1.2 times to accommodate the selection multiplexor of the set-associative cache. The cache miss penalty is 50ns for either cache organization (usually rounded up or down to an integer number of clock cycles). First, calculate the average memory access time and then processor performance (CPU time taken to run a program). Assume the hit time is 1 clock cycle, the miss rate of a direct-mapped 2.4%, and the miss rate for a two-way set-associative cache of the same size is 1.6%.
6. Assume a disk subsystem with the following components and MTTF:
   8 disks, each rated at 500,000-hour MTTF
   1 SCSI controller, 400,000-hour MTTF
   1 power supply, 250,000-hour MTTF
   1 fan, 200,000-hour MTTF
   1 SCSI cable, 1,000,000-hour MTTF
   a. Using the simplifying assumptions that the lifetimes are exponentially distributed and those failures are independent, compute the MTTF of the system as a whole.
   b. Suppose we improve the reliability of the all the hard disks from 500,000-hour MTTF to 1, 000, 0000-hours MTTF and other components remain the same. Compute the improvement in the reliability of the entire system using Amdahl's Law.
7. Why the hardware *reorder buffer* (ROB) was added to the processor that implements dynamic scheduling of instruction execution using Tomasulo's algorithm?
8. What search technique is used in instruction/data caches? Illustrate.

9. Consider the following code that computes *for i = 1 to n* $A[i] = A[i] + B[i]$

```
Loop:   L.D      F1, 0(R1)   ; R1 points to the beginning of the first array A1
        L.D      F2, 0(R2)   ; R2 points to the beginning of the second array A2
        ADD.D    F3, F1, F2
        S.D      F3, 0(R1)
        DADDUI   R1, R1, #8  ; increment R1 to point to the next element in A1
        DADDUI   R2, R2, #8  ; increment R2 to point to the next element in A2
        BNE      R1, R0, Loop ; repeat for next array element if R1 != R0
```

The following table shows the latency of various CPU operations. The last column is the number of intervening clock cycles needed to avoid a stall.

Instruction producing result	Instruction using result	Latency in clock cycles
FP ALU op	Another FP ALU op	3
FP ALU op	Store double	2
Load double	FP ALU op	1
Load double	Store double	0
Load Integer	Any instruction	1
Integer ALU op	Any instruction	0

(a) Show how the loop would look on MIPS, both scheduled and unscheduled, including any stalls or idle clock cycles. Schedule (reorder without affecting the output of the code) the instructions for delays from floating-point operations. Ignore delayed branches. Computer the clock cycle count for one iteration of the loop for both scheduled and unscheduled cases.

(b) Suppose $n = 50$. Unroll the loop three times and show the code of unrolled loop and computer the total number of clock cycles required to execute the original loop n times. Note that three times unrolled loops would execute much less than n times to achieve the same effect.

(c) Show how multiple iterations of the loop can be executed in parallel using Tomasulo's algorithm without unrolling the loop. Show the snapshot of various tables used (*instruction status table, reservation station table, and register status table*) in this process..

10. Consider a RISC processor that implements dynamic scheduling and handles all the hazards using the scoreboard technique developed for CDC 6600 computer. It has two multipliers, one divider, one adder, and a single integer unit for all memory references, branches and integer operations. The scoreboard issues as many instructions as possible and the functional units execute them if there is no hazard. The following code sequence is to be executed with highest possible level of ILP:

```
L. D     F0, 12(R5)
L. D     F10, 12(R7)
MUL.D    F2, F0, F4
ADD. D   F4, F2, F10
DIV.D    F2, F10, F0
SUB.D    F0, F2, F4
S.D      F0, 20(R5)
```

Show the detailed data structure (*instruction status table, functional unit status table, and register result status table*) maintained in the scoreboard of this processor when the first L. D has completed and written result and the second L. D has completed but execution but has not yet written its result. You may make any reasonable assumption if required and mention it properly.

11. Suppose the code sequence of Q9 is to be dynamically scheduled using Tomasulo's algorithm developed for IBM 360/91 processor that contains additional registers called reservation stations, load buffers, and store buffers all of which are connected to CPU registers via common data bus (CDB). The processor has two load units, one store unit, two multiply/divide units and three add/subtract units. Show the data structure (*instruction status table, reservation station table, and register status table*) when
    (a) the first L. D has completed and written result and the second L. D has completed execution but has not yet written its result.
    (b) the first MUL. D is ready to write its result.

12. Suppose that following observations were made for a program running on a typical computer: Frequency of floating point (FP) operations = 20%, average CPI of FP operations = 5, average CPI of other instructions = 1.5, frequency of floating point square root (FPSQRT) operations = 4%, CPI of FPSQRT = 25. Assume that we have two design alternatives. The one is to decrease the CPI of FPSQRT to 4 and the other is to decrease the CPI of all FP operations to 3. Compare these two design alternatives (i.e., analyze and conclude which one is better and how much) using the processor performance equation. Alternatively you may use the Amdahl's law for this comparison.

13. State giving reason whether or not the processors with lower CPIs will always be faster. The IBM Power-5 processor is designed (in 2005) for high-performance integer and FP. It contains two processor cores each capable of sustaining four instructions per clock, including two FP and two load-store instructions. The highest clock rate for a Power-5 processor is 1.9 GHz. In comparison, the Pentium-4 offers a single processor with multithreading. The processor can sustain three instructions per clock with a very deep pipeline, and the maximum available clock rate (in 2005) was 3.8 GHz. Which of the two processors will be faster and why?

14. Consider the following code that multiplies the corresponding elements of two arrays and stores in a third array.

```
Loop:   L.D     F1, 0(R1)   ; R1 points to the beginning of the first array A1
        L.D     F2, 0(R2)   ; R2 points to the beginning of the second array A2
        MUL.D   F3, F1, F2
        S.D     F3, 0(R3)   ; R3 points to the beginning of the result array A3
        DADDUI  R1, R1, #4  ; increment R1 to point to the next element in A1
        DADDUI  R2, R2, #4  ; increment R2 to point to the next element in A2
        DADDUI  R3, R3, #4  ; increment R3 to point to the next element in A3
        BNE     R1, R0, Loop ; repeat for next array element if R1 != R0
```

The following table shows the latency of various CPU operations. The last column is the number of intervening clock cycles needed to avoid a stall.

Instruction producing result	Instruction using result	Latency in clock cycles
FP ALU op	Another FP ALU op	3
FP ALU op	Store double	2
Load double	FP ALU op	1
Load double	Store double	0
Load Integer	Any instruction	1
Integer ALU op	Any instruction	0

Show how the loop would look on MIPS, both scheduled and unscheduled, including any stalls or idle clock cycles. Schedule (reorder without affecting the output of the code) the instructions for delays from floating-point operations. Ignore delayed branches.

15. Arrange a 2-bit local branch predictor, a tournament branch predictor, a two level branch predictor and a correlating branch predictor in the order of increasing prediction accuracy. What can be the cost of misprediction in the worst case and why?

16. Consider a RISC processor that implements dynamic scheduling and handles all the hazards using the scoreboard technique developed for CDC 6600 computer. It has two multipliers, one divider, one adder, and a single integer unit for all memory references, branches and integer operations. The scoreboard issues as many instructions as possible and the functional units execute them if there is no hazard. The following code sequence is to be executed with highest possible level of ILP:

```
L. D      F0, 20(R5)
L. D      F10, 12(R7)
DIV.D     F2, F10, F0
ADD. D    F4, F2, F10
MUL.D     F2, F0, F4
MUL.D     F0, F3, F4
S.D       F5, 24(R5)
```

Show the detailed data structure (*instruction status table, functional unit status table, and register result status table*) maintained in the scoreboard of this processor when the first L. D has completed and written result and the second L. D has completed but execution but has not yet written its result. You may make any reasonable assumption if required and mention it properly.

17. Suppose the code sequence of Q5 is to be dynamically scheduled using Tomasulo's algorithm developed for IBM 360/91 processor that contains additional registers called reservation stations, load buffers, and store buffers all of which are connected to CPU registers via common data bus (CDB). The processor has two load units, one store unit, two multiply/divide units and three add/subtract units. Show the data structure (*instruction status table, reservation station table, and register status table*) when
    (c) the first L. D has completed and written result and the second L. D has completed execution but has not yet written its result.
    (d) the first MUL. D is ready to write its result.

18. In a typical two-level cache system, in 100 memory references, there are 5 misses in first-level cache and 3 misses in the second-level cache. What are the various miss rates? Assume the miss penalty from the L2 cache to memory is 150 clock cycles, the hit time of the L2 cache is 12 clock cycles, the hit time of L1 is 1 clock cycle, and there are 1.6 memory references per instruction. What is the average memory access time and average stall cycles per instruction? Ignore the impact of writes.
19. List various types of exceptions (also refereed as *faults* or *interrupts*) that describe exceptional situations where normal execution order of instructions is changed and interrupt service routine of the operating system has to be called in order to deal with the situation.
20. Compare the following types of exceptions:
    a. Synchronous versus asynchronous
    b. User requested versus coerced
    c. User maskable versus user nonmaskable
    d. Within versus between instructions
    e. Resume versus terminate

21. Why data is read from or written to the main memory as fixed size block of multiple bytes even if not all the bytes in the data block are required to execute the current instruction? What is aligned memory access?
22. Compare temporal locality and spatial locality patterns of data or instruction access.
23. How does a CPU identify instructions and data bytes stored in the main memory while both are similar sequences of zeros and ones?
24. What sequence of activities takes place within the CPU, cache, ROM, RAM, Hard disk (or any other booting device) and other peripherals attached with the computer when the user switch on the power button of a computer?
25. Why there are multi-level caches, such as L1-cache, L2-cahce, and L3 cache in modern computer systems? Compare this with only one level cache in older systems.
26. Compare unified cache with split cache.
27. Sketch the design of ROM to illustrate why it retains data even after power is switched off.
28. Can one replace the entire RAM by an equal sized cache? Given reasons for your answer.
29. What is advantage of using base plus displacement form of address in place of absolute address?
30. Why do we sometimes need index register in addition to the base register for specifying operands?
31. As an assembly level programmer, what information do you need about a compuer system in order to create a non-trivial application program for it?
32. Compare system programs versus application programs. Give examples.
33. Compare hardware interrupts versus software interrupts. Give examples.
34. Compare synchronous versus asynchronous data transfer. Give examples.
35. What do you mean by cache mapping functions?
36. Compare working, advantage and disadvantages of direct mapping, associate mapping and set associative mapping for caches.
37. Compare working, advantage and disadvantages of cache write policies write through, write back and write once.
38. Compare large cache line (number of bytes) versus small cache line. Note that all the data bytes in one main memory block are read together and stored in one cache line. So, a main memory block and a cache line would both have same number of bytes.
39. What is cache coherency problem? Explain, with necessary diagram and tables, any one method that is used to ensure cache coherency.
40. Comment on the following cache optimization techniques stating why/how:
    a. Larger cache size reduces miss rate.
    b. Large cache block (line size) reduces miss rate.
    c. Higher associativity reduces miss rate.
    d. Multilevel cache reduces miss penalty.
    e. Giving priority to read misses over writes reduces miss penalty.
    f. Avoiding address translation during indexing of the cache reduces hit time.
41. Compare superscalar processor versus super pipelined processor.
42. State the reason why the clock rate for pipelined processor is a little slower than the unpipelined processor with same set of hardware except the pipeline hardware.
43. Pipeline depth for two computer systems A and B are 5 and 10 respectively. Comment on the performances of these computers in terms of execution speed for the same program code.
44. State two reasons as why it is easier and more efficient to design pipeline for RISC systems than for the CISC system.
45. State one hardware or software technique to minimize
    a. Structural hazard
    b. Data hazard
    c. Control hazard
46. Explain RAW, WAR and WAW data hazards with examples.

47. How do CISC systems pipeline execution of an instruction which performs same operation on specified number of data items, such as move N bytes from specified source address to specified destination address in the main memory?
48. How do RISC systems pipeline execution of an instruction, such as multiply *or* divide, whose ALU operation prolongs of several cock cycles instead of just one clock cycle for most other instructions? Illustrate with necessary figure.
49. CISC instructions may be highly complicated so as to pose additional burden to the designer for maintaining the precise exceptions. Explain three such features of CISC systems which are not found in RISC systems. How precise exceptions are ensured with these complicated features?
50. What problems may arise in maintaining precise exceptions in RISC system due instructions, such as multiply *or* divide or floating point addition, whose ALU operation prolongs of several cock cycles instead of just one clock cycle? Explain the four possible techniques that a designer can choose to maintain precise exceptions in the presence of these problems.
51. Draw a neat sketch to illustrate how the data path is pipelined by adding a set of registers, one between each pair of pipe stages in a typical RISC system. You have to show the interconnection of various CPU registers, pipeline registers, ALU, multiplexers, instruction cache, data cache, instruction decoder, etc.
52. Based on the diagram for Q 52, give an modified diagram when, to minimize the impact of deciding whether a conditional branch is taken, we compute the branch target address in ID stage while doing the conditional test and final selection of next PC in EX stage of the RISC pipeline.
53. Give compiler techniques to minimize the
    a. Structural hazard
    b. Data hazard
    c. Control hazard
54. Explain the techniques of static branch prediction.
55. Explain the techniques of dynamic branch prediction.
56. What are the limitations of instruction level parallelism?
57. What is thread level parallelism? Why do the CPU designers go for it?

--Next set of questions to be given shortly--