B.E. COMPUTER SCIENCE AND ENGINEERING SECOND YEAR SECOND SEMESTER -2019
COMPUTER ARCHITECTURE

Time: Three Hours **Full Marks:100**

Group-A

Answer twenty-eight (28) questions. **28 × 3 = 84**

Choose the unique correct answer

For Q1..Q4 : Consider a 5-stage instruction pipeline IIF, ID, EX, MEM, WB). The first instruction b0 is a branch instruction and it occupies the IF stage in cycle-1. The subsequent instructions are b1, b2, b3, etc. The instruction at the branch target is bt and its successors are bt1, bt2, bt3, etc.

For Q1..Q2: Suppose the pipeline is stalled whenever a branch instruction is detected in the ID stage. However, the branch is not taken.

1. In cycle-3, b1 performs
(a) IF
(b) ID
(c) EX
(d) MEM

2. In cycle-5, b2 performs
(a) IF
(b) ID
(c) EX
(d) MEM

For Q3..Q4: An alternative scheme, which is more efficient than stalling on every branch (See Q1-Q2), is to treat every branch as NOT TAKEN. However, if it is detected in the ID stage of b0 (where decoding and reading registers are done simultaneously), that the branch will be taken, bt performs IF in the following cycle. Then b1 is converted to a no-op.

3. If the branch is taken, then in cycle-4, bt performs
(a) IF
(b) ID
(c) EX
(d) MEM

4. If the branch is taken, then at the end of cycle-9,
(a) only b1 and bt will be completed
(b) only b1, bt, and bt1 will bw completed
(c) only b1,bt, bt1, and bt2 will be completed
(d) none of the above

For Q5..Q7: Consider the MIPS floating point unit using Tomasulo's Algorithm for Dynamic Scheduling. In the following code sequence, all instructions have issued; however only the first instruction has written its result. The second instruction has completed execution bot not written its result.

```
L.D     F6, 32(R2)
L.D     F2, 44(R3)
MUL.D   F0,F2,F4
SUB.D   F8, F2, F6
DIV.D   F10, F0, F6
ADD.D   F6,F8,F2
-----------------------
```

5. For SUB.D, Vk is
(a) Load1, i.e. L.D F6, 32(R2)
(b) Mem[32 + Regs[R2]]
(c) Load2, i.e. L.D F2, 44(R3)
(d) Mem[44 + Regs[R3]]

6. For Add2, i.e. ADD.D instruction, Qj is
(a) Load2
(b) Regs[F8]
(c) Regs[F2]
(d) Add1

7. For Mult1, i.e. MUL.D instruction, Qj is
(a) Regs[F2]
(b) Add2, i.e. ADD.D instruction
(c) Load2, i.e. L.D F2,44(R3)
(d) Add1, i.e. SUB.D instruction

For Q8..Q10: Consider the MSI Protocol for Cache coherence.

8. If a cache in state M observes a BusRdX transaction, it
(a) moves to state S
(b) remains in state M
(c) flushes its contents
(d) flushes its contents and moves to state I

9. If a cache in state I receives a PrWr request, it
(a) moves to state M
(b) moves to state S
(c) remains in state I
(d) generates a BusRdX transaction and moves to state M

10. If a cache in state M observes a BusRd transaction, it
(a) moves to state S
(b) moves to state I
(c) remains in state M
(d) none of the above

For Q11..Q14: Consider a 2-ary 3-fly butterfly network with 3 stages and 2 outputs per crossbar-switch. The switches are numbered d_1d_2 where $0 \le d_1 \le 2$ and $0 \le d_2 \le 3$. Here d_1 represents the stage-number (0 for input, 1 for middle , and 2 for output). There are 8 inputs and 8 outputs. An input is represented by $a_2a_1a_0$ and an output by $b_2b_1b_0$, where a_i and b_j $(0 \le i,j \le 2)$ are binary bits.

For each stage, there are 8 links represented by a 3-bit string $e_2e_1e_0$. The topology of the network is specified by the following connections:

Stage-0 switch	Inputs
i $(0 \le i \le 3)$	i and 2i+1

Stage-0 output-link	Stage-1 input-link
$x_1x_2x_3$	$x_3x_2x_1$

Stage-1 output-link	Stage-2 input-link
$y_1y_2y_3$	$y_1y_3y_2$

Stage-2 switch	Output
j	2j and 2j + 1

Any switch	Input-links	Output-links
d_1d_2	$2d_2$ and $2d_2$ +1	$2d_2$ and $2d_2$ +1

Now consider the routing of input $a_2a_1a_0$ to output $b_2b_1b_0$.

11. The stage-0 output link to be selected is
(a) $b_0a_1a_0$
(b) $a_1a_2a_0$
(c) $a_2a_1b_2$
(d) $a_2b_1a_0$

12. The stage-1 input-link to be selected is
(a) $b_2a_1a_2$
(b) $a_2b_1a_0$
(c) $a_2a_1b_0$
(d) $a_1b_1b_0$

13. The stage-1 output link to be selected is
(a) $a_2b_1b_0$
(b) $a_1b_2b_0$
(c) $b_1b_0a_2$
(d) $b_2a_1b_1$

14. The stage-2 input-link to be selected is
(a) $b_0b_1a_0$
(b) $b_2b_1a_1$
(c) $b_2a_2b_0$
(d) $a_1b_0b_2$

---

15. An interconnection network is a system because it consists of the following component:
(a) buffers
(b) channels
(c) switches
(d) all of the above

16. Circuit-switching is a form of
(a) bufferless flow-control
(b) buffered flow-conbtrol
(c) crossbar switching
(d) multistage switching

17. To advance to the next stage, a poacket must be allocated
(a) a switch
(b) a buffer
(c) bandwidth on an input-channel
(d) a semaphore

18. Packet-buffer flow-control is inefficient because
(a) buffers are allocated in units of packets
(b) contention latency is increased
(c) both (a) and (b)
(d) none of the above

19. Virtual-channel flow-control overcomes the blocking problems of wormhole flow-control
(a) by forcing senders to send routing requests
(b) by reallocating a blocked channel
(c) by associating several virtual channels with a single physical channel
(d) by maintaining large buffers

20. A significant advantage of distributed memory systems is that
(a) load balancing is achieved
(b) contention problem is not severe
(c) deadlocks can never occur
(d) all of the above

21. A disadvantage of shared memory systems is
(a) there is no need to physically move data when processes communicate
(b) lack of scalability
(c) synchronization is very simple to achieve
(d) uniprocessor programming techniques cannot easily be adapted

22. Shared memory systems are basically classified according to their
(a) synchronization primitives
(b) interconnection network
(c) cache-coherent protocol
(d) memory organization

23. A multistage network is a
(a) switching network
(b) shared-path network

24. In centralized arbitration, if a requesting master obtains a grant from the arbiter in response to its request, it
(a) immediately starts bus transaction
(b) deactivates its request
(c) activates the bus busy line
(d) both (b) and (c)

25. In a daisy-chained grant for bus-arbitration, the priority of a master
(a) is determined by its position in the grant chain
(b) changes dynamically with time
(c) is determined by distributed voting
(d) is determined by the arbiter according to the requests received

26. The B-registers in the address section of the CRAY X-MP
(a) hold addresses referenced in memory operations
(b) are used as index registers
(c) are used for saving registers during subroutine linkage
(d) are connected directly to the functional units

For Q27..Q29 Consider the following CRAY X-MP program:

```
A1      53
VL      A1
V4      V2 + V3
```

The vector integer adder is a 3-stage pipeline

27. The first result emerges after
(a) 3 cycles
(b) 6 cycles
(c) 9 cycles
(d) none of the above

28. All the 53 results emerge at the end of
(a) 55 cycles
(b) 52 cycles
(c) 53 cycles
(d) 58 cycles

29. The functional unit, i.e. adder, can be reused after
(a) 59 cycles
(b) 61 cycles
(c) 56 cycles
(d) 57 cycles

For Q30..Q31: Consider a directory-based cache-coherence with MSI protocol:

30. There is a read miss at node-I and the request goes to the home node for the block. If the dirty bit is ON,
(a) the identity of the owner node is sent to the requestor
(b) the block is supplied to the requestor

31. There is a write-miss at node-I and the dirty bit is OFF. Then
(a) invalidation requests are sent to all nodes
(b) invalidation requests are sent to all nodes j for which presence[j] is ON
(c) requestor obtains the block from the home node and places it in its cache in dirty state
(d) both (b) and (c)

---

32. In the MESI protocol, if a cache controller receives a PrRd request while in the I-state, it
(a) moves to the S-state if the shared signal is inactive
(b) move to the E-state if the shared signal is active
(c) moves to the E-state if the shared signal is inactive
(d) moves to the M-state

33. In the MESI protocol, if a cache controller observes a BusRd while in the E-state, it
(a) moves to the S-state
(b) moves to the M-state
(c) moves to the I-state
(d) remains in the E-state

For Q34..Q35: Consider the following MIPS code:

```
Loop: L.D      F0,0(R1)     ; F0 = array element
      ADD.D    F4, F0, F2   ; add scalar in F2
      S.D      F4, 0(R1)    ; store result
      DADDUI   R1, R1, #-8  ;decrement pointer 8 bytes (per DW)
```

R1 is initially the address of the last (highest address) element of the array. R2 points to the element preceding the first one (i.i., 8 + R1 = address of the first element).

There is a latency constraint of 1 cycle between L.D and ADD.D, 2 cycles between ADD.D and S.D, and 1 cycle between DADDUI and BNE.

Suppose the loop is unrolled 4 times. The registers used for load and store are (F0, F4), (F6,F8), (F10,F12), (F14,F16) for the four consecutive elements of the array in the unrolled loop.

34. The stalls take
(a) 14 cycles
(b) 4 cycles
(c) 8 cycles
(d) 13 cycles

35. The execution time of the unrolled loop
(a) can never be reduced
(b) can be reduced by bypassing
(c) can be reduced by pipeline scheduling
(d) can be reduced by register renaming

Group-B

36. A superscalar processor has four(4) execution units. Each execution unit can execute any operation. Load operations have a 2-cycle latency, and all other operations have a 1-cycle latency. (The latency between an instruction I_1, and an instruction I_2 dependent on I_1, is the time-delay (cycles) between their issue cycles) Each processor has a 5-stage pipeline [IF(Fetch instruction),ID(Decode instruction), RR(Read registers), EX(execute), WB (write result back into registers)]. An instruction is said to have issued when it passes from the RR stage into the EX-stage. Instructions cannot be executed out-of-order (i.e., the instructions cannot be reordered). Now consider the following instruction sequence:

```
-------------------------
ADD   r1, r2, r3      ; r1 := r2 + r3
SUB   r5, r4, r5      ; r5 := r4 – r5
LD    r4, (r7)        ; r4 := (r7)
MUL   r4, r4, r4
ST    (r7), r4
LD    r9, (r10)
LD    r11, (r12)
ADD   r11, r11, r12
MUL   r11, r11, r11
ST    (r12), r11
-------------------------
```

How long would the program take to issue ? **16**