Attempt all of the following questions.

Q(1) **(12 points)**

State however the following statements are true or false and if false mention the truth:

A. It is an organization issue whether a computer will have one or multiple buses.

B. The implementation technique of a multiply instruction is an architectural issue.

C. Utilizing faster processor results in an increase in the performance regardless of the memory speed.

D. Pipelining decreases CPU instruction throughput but reduces the execution time of each individual instruction.

E. Variable length instructions make the pipelining much easier.

F. Cache memory is better implemented using DRAM for its superior speed.

Ans.

A. False: it is an architectural issue.

B. False: is an Organizational issue.

C. False: The memory speed is a real limitation for overall CPU performance.

D. False: it does not reduce the individual instruction execution time.

E. False: Fixed length instructions make the pipelining easier.

F. False: is better implemented using SRAM.

Q(2) **(12 points)**

A) Briefly explain why the computer needs to utilize a memory hierarchy?

b) Give at least two reasons for why there is an optimum pipeline depth after which the performance remains the same or even get worse?

c) Explain with the aid of an example how can the MIPS leads to a false performance indication?

d) Give an assembly example for the WAR hazard? Explain why this kind of hazards cannot appear in the MIPS-5 stage pipelined architecture?

Ans.

A. Computer needs need memory to fit very large programs & data and to work at a speed comparable to that of the microprocessors. The main issue is that memories are much slower than processors and the faster the memory the greater the cost per bit. The solution is to build a composite memory system which combines a small fast memory and a large slow main memory; which behaves (most of the time) like a large fast memory. This is called memory hierarchy.

B. The two reasons are:

   I. As more pipeline stages are added, extra delays are expected due to increasing overhead of inter-pipeline stages buffering.

   II. The pipeline hazards are getting more complex to be resolved (takes much more time = extra delays).

C. MIPS as a performance measure may be high but the performance is low which is called false indication. This can be demonstrated by several examples hereafter an example: Use of a Floating Point Unit (FPU) vs. S/W routines for floating point operations. FPU requires less time and less instructions, S/W uses many simple integer instructions leading to what seems like higher MIPS rating; even though it takes more time than the FPU to perform the task.

D. Here is an assembly example of the WAR data hazard:

I: sub **r4, r1, r3**

J: add **r1, r2, r3**

K: mul **r6, r1, r7**

- It is pretty clear that the add instruction is trying to write the result of the addition to r1 while the sub instruction is utilizing r1 as a source. In some processors it is allowed that instructions are completed out of order. Hence, for these processors if the add instruction is completed before the sub instruction then the sub will read wrong data from r1.
- This kind of hazards can't happen in MIPS 5-stages pipelined processor for the following reasons:
  - All instructions take 5-stages, and

- Reads are always in stage number 2, and
- Writes are always in stage number 5.

Q(3) **(12 points)**

A) Explain, in points, why pipelining is much difficult in CISC than RISC architectures?

b) List the micro operations of the following two instructions: Jal L2, Jr $s1?

c) Explain why Harvard architecture is favoured over Von Neumann architecture for pipelined processor?

Ans.

A. Pipelining in CISC is much more difficult than it in RISC for the following:
   1. CISC has variable length instructions that make pipelining difficult; while RISC has fixed length instructions.
   2. CISC has large number of instructions that have different CPI (one, two, etc.) which make the pipelining more difficult.

B. Here are the micro operation:

	3 bus	**Single bus**
Jal l1	**T3: ra ← pc** **T4: pc ← pc(31-28) || IR (25-0) || 00**	**T3 : ra ← pc , A ← pc(31-28)** **T4: pc ← pc(31-28) || IR (25-0) || 00**
Jr r1	**T3 :Pc ← r1**	**T3 :Pc ← r1**

C. Harvard architecture is based on separate memories for both data and instructions. Thus, it is used at cache level to avoid resource hazards in pipelined processors. While Von Neumann is based on unified memory for both data and instructions and hence is better utilized in main memory to allow the programmer to dynamically allocate memory space for data and instructions as needed.

Q(4) Given the following code segment: **(12 points)**

```
Add     R1, R0, #20
Mul     R3, R2, #3
And     R4, R1, #3A
```

Add R5, R0, R2

A) Draw the Gantt chart for executing the previous code segment on a 4-stage pipelined processor (F: Fetch, D: Decode, E: Execute, and W: Write result back)?

B) If there is any hazard during the execution of this code segment mention it and its type? What is the proper way to handle it?

C) Trace the contents of the three inter-pipeline stage buffers during the execution clock cycles?

ANS.

A. Hereafter is the Gantt chart.

F1	D1	E1	W1				
	F2	D2	E2	W2			
		F3	Stall	D3	E3	W3	
			Stall	F4	D4	E4	W4

B. Data hazard may occur between I1 and I3 through the utilization of R1 as a destination in I1 and reusing it as a source in I3 Read After Write data dependency (RAW). Also a resource hazard may occur between I3 and I4 in the as the stall in I3 prevent the use of the B1 buffer by the Fetch of I4.

C. See the table below:

CLK	B1	B2	B3
1	-	-	-
2	I1	-	-
3	I2	ADD, R0, 20	-
4	I3	MUL, R2, 3	2020, R1
5	I3	MUL, R2, 3	150, R3
6	I4	AND, R1, $3A_H$	150, R3
7	-	ADD, R0, R2	$1A_H$, R4
8	-	-	2050, R5

The grey shaded cells are due to the pipeline stalls.

Q(5) **(12 points)**

Suppose a computer using direct mapped cache has 2^{32} words of main memory, and a cache of 1024 blocks, where each cache block contains 32 words.

a. How many blocks of main memory are there?

b. What is the format of a memory address as seen by the cache, i.e., what are the sizes of the tag, block, and word fields?
c. To which cache block will the memory reference $(000063FA)_{16}$ map?
d. What is the role of the valid bit in a direct mapped cache?

Ans.

a. Number of blocks of MM= $2^{32} / 2^5 = 2^{27}$
b. 32 bit addresses with 17 bits in the tag field, 10 in the block field, and 5 in the word field.

Tag	Block	Word
17	10	5

c. 000063FA = **00000000000000000 1100011111 11010**, which implies Block **799.**
d. The role of the valid bit in a direct mapped cache is to indicate if that block is filled with any valid data from the MM or not i.e. if valid bit =1 then this block have valid data but if valid = 0 then no further checks for the tag or the word is needed because there is no valid data in this block.

Best wishes,
Dr. Mostafa Elsayed