Q1: define the following terms:

MDR:
Memory data register

MAR:
Memory address register

Immediate addressing mode

Q2: Drew the figure of the following: 1- Cpu with one bus structure.
Q3: explain in details the major hazards in pipelined execution.

الجواب

Structural hazards (Resource Conflicts) Hardware Resources required by the instructions in simultaneous overlapped execution cannot be met Data hazards (Data Dependency Conflicts) An instruction
scheduled to be executed in the pipeline requires the result of a previous instruction, which is not yet available

**Control hazards**

Branches and other instructions that change the PC make the fetch of the next instruction to be delayed

---

**Q4: give one reason for the following**

1- Using comparison instruction CMP in SAFE POP routine of stack.
   التعبير: لضمان عدم السحب من ستاك فارغ
2- Why pipelining is possible in RISC?
   الجواب
   Because all of instructions execution in a uniform amount of time (i.e. one clock ), so pipeline is possible.

**Q5: construct the format of the following assembly program in PDP-11:**

```
A:. WORD 20.
C:. WORD 15
B:.WORD 28
```
CLR R2
ADD A, R2
SUB @C, R2
ADD #5, R2
MOV R2, @#B
HLT
END
الجواب لسؤال 5
Q6: SOLVE THE FOLLOWING

1. BIS #007321, #001010
   
   | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
   | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
   
   | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |

2. MUL 2:3, 5:2 IN RISC approach
   
   LOAD A, 2:3
   LOAD B, 5:2
   PROD A, B
   STORE 2:3, A

Q7: compare between CISC and RISC (GIVE 6 POINTS)

<table>
<thead>
<tr>
<th></th>
<th>CISC</th>
<th>RISC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emphasis on hardware</td>
<td>Emphasis on software</td>
<td></td>
</tr>
<tr>
<td>Includes multi-clock complex instructions</td>
<td>Single-clock, reduced instruction only</td>
<td></td>
</tr>
<tr>
<td>Memory-to-memory:</td>
<td>Load to register:</td>
<td>Register to register:</td>
</tr>
<tr>
<td>&quot;LOAD&quot; and &quot;STORE&quot; incorporated in instructions</td>
<td>&quot;LOAD&quot; and &quot;STORE&quot; are independent instructions</td>
<td></td>
</tr>
<tr>
<td>Small code sizes, high cycles per second</td>
<td>Low cycles per second, large code sizes</td>
<td></td>
</tr>
<tr>
<td>Transistors used for storing complex instructions</td>
<td>Spends more transistors on memory registers</td>
<td></td>
</tr>
</tbody>
</table>

Q8: explain briefly the factors that make interfacing difficult between cpu and input-output devices

• The encoding of the transmitted word must be that which is
employed by the I/O device.

- Operating Rates
  - The CPU and Main Memory operate at many times the speed of I/O devices

- Timing and Control
  - Exchange of status signals between CPU and device.
  - Rate of transmission from device to CPU or vice-versa.

- Communication Link (Word Length)
  - There are at least 25 different word lengths used in computers. The word lengths vary from 4 to 128 bits. The separation (or combination) of words (because of word length) into characters, bytes or other units presents a "word assembly" problem.

Q9: EXPLAIN Briefly:

1. CPU FUNCTIONS
   - Fetch a word from memory
   - Store a word to memory
   - Register transference
   - Perform arithmetic and logic operation

2. Transmission type of word between CPU and I/O devices
   - Serial by character(byte) quasiparallel
   - Serial by word(parallel)
   - Serial by bit

4. Q10: write the control sequence to execute the instruction ADD @R2,R1

5. 

<table>
<thead>
<tr>
<th>STEP</th>
<th>ACTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PC OUT, MAR IN, READ, CLEAR Y, SET CURRY IN ALU, ADD, Z IN</td>
</tr>
<tr>
<td>2</td>
<td>Z OUT, PC IN, WAIT MFC</td>
</tr>
<tr>
<td>3</td>
<td>MDR OUT, IR IN</td>
</tr>
</tbody>
</table>
Q11: EXPLAIN by using graph how CPU asks the main memory (main store) the data and write it in the cache