1. [7 points]
   (i) (3 points) With the help of an example code, explain what is meant by control hazards in a pipelined processor.

   Control-changing instructions such as branches disrupt the straightline flow of control through a program. When a pipelined processor overlaps the execution of a control-changing instruction with that of instructions that sequentially follow it, control hazards are generated.

   Consider the following code and its potential execution in a standard 5-stage pipeline.

   ```
   BEQZ R1, Label1 # Branch to Label1 if R1 = 0
   R3 <-- R2
   Label1: R5 <-- R3 + R4
   ```

   In this pipeline execution, the second instruction enters the pipeline before the branch instruction’s outcome has been determined. If the second instruction is allowed to complete and the branch evaluates to be taken, then the third instruction obtains the incorrect value of R3.

(ii) (4 points) Describe two techniques to overcome control hazards. Mention at least one drawback of your proposed methods.

   a. A simple solution to deal with control hazards is to stall the pipeline when a control hazard is detected. Once the branch outcome is known, the pipeline continues fetching instructions from the correct path. The disadvantage of this method is loss of performance (branches form about 15-20% of the instructions).

   b. Another technique to overcome control hazards is to do branch prediction. That is, when a branch instruction is fetched, the CPU makes a prediction about the branch outcome and continues execution based on the prediction. If the prediction turns out to be incorrect, then the incorrectly fetched instructions are discarded from the pipeline, and instruction fetching is initiated from the correct address. One drawback of this method is the additional hardware needed to store the history of previous branches.
2. [7 points]
   (i) (2 points)
   Explain the purpose of using DMA controllers in computer systems.

   The DMA controller allows certain hardware subsystems within the computer system to access system memory independently of the CPU. Without a DMA controller, the CPU is directly involved in all I/O transfer operations. With a DMA controller, the CPU initiates the transfer, but performs other useful tasks while the DMA controller performs the intricate aspects of the actual transfer. Thus, the time required to perform the (slow) I/O transfer is overlapped with useful work done by the CPU, thereby improving the overall performance of the computer system.

   The DMA controller is useful especially for transferring large blocks of data to/from block-oriented devices such as disk drives.

   (ii) (3 points)
   Explain what is meant by **Kernel mode** and **User mode** and the need for the CPU to function in these two modes.

   When the CPU is in the Kernel mode, the executing program has unrestricted access to the underlying hardware. It can execute any instruction in the ISA and reference any memory address. The Kernel mode is used to run the trusted routines of the operating system.

   When the CPU is in the User mode, the executing program has limited access to the underlying hardware. It can execute only User mode instructions in the ISA and reference only User mode memory address space.

   Modern computer systems have two (or more) operation modes so that the operating system can support features such as virtual memory, protection, and multitasking.

   (iii) (2 points)
   Mention 3 situations that cause the CPU to switch from User mode to Kernel mode, and one situation that causes it to switch from Kernel mode to User mode.

   The CPU switches from User mode to Kernel mode:
   a. When the user process executes a **system call** (or trap) instruction.
   b. When the user process causes an exception (such as divide by zero and page fault).
   c. When an I/O device raises a hardware interrupt.

   The CPU switches from Kernel mode to User mode when the operating system schedules a user process to run.
3. [6 points]
A design team is developing a new processor. The memory system is 32-bit byte-addressable. The on-chip cache memory is 128 KByte 4-way set-associative, with a 64 byte block size.
(i) (1.5 points)
Draw a diagram showing how the cache controller will split the memory address; for each field, show its name and number of bits.

| +------------------------------------+---------------+-----------------+
| | | |
| | Tag | Set Number | Offset in Block |
| | | | |
| +------------------------------------+---------------+-----------------+

17 bits 9 bits 6 bits

(ii) (4.5 points)
The design team decided to change the cache architecture to a direct mapped one. For each of the parameters in the following table, indicate the impact of this design change (i.e., indicate whether it increases/decreases/stays same/cannot be determined). Feel free to give explanation for your answer in the last column
<table>
<thead>
<tr>
<th>No.</th>
<th>Parameter</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cache access latency</td>
<td>Decreases</td>
</tr>
<tr>
<td>2</td>
<td>Cache miss rate</td>
<td>Increases</td>
</tr>
<tr>
<td>3</td>
<td>Cache compulsory misses</td>
<td>Stays same</td>
</tr>
<tr>
<td>4</td>
<td>Cache conflict misses</td>
<td>Increases</td>
</tr>
<tr>
<td>5</td>
<td>Cache tag size</td>
<td>Decreases</td>
</tr>
<tr>
<td>6</td>
<td>Number of cache sets</td>
<td>Increases</td>
</tr>
<tr>
<td>7</td>
<td>Cache block size</td>
<td>Stays same</td>
</tr>
<tr>
<td>8</td>
<td>Total cache size (Data + Overhead)</td>
<td>Decreases</td>
</tr>
<tr>
<td>9</td>
<td>Average memory latency</td>
<td>Cannot be determined</td>
</tr>
</tbody>
</table>