1. What changes would need to be made to the basic pipeline CPU to reduce the branch hazard penalty from three clock cycles to one? Describe the changes and/or sketch a simple diagram indicating the changes.

2. What changes would be needed to reduce or eliminate data hazards from the pipeline as much as possible. Describe the changes and/or sketch a simple diagram indicating the changes. What data hazard cannot be completely eliminated?
3. What are a cache "hit" and a cache "miss"? How are they detected?

4. How does a set-associative cache differ from a direct-mapped cache? Why are there usually fewer cache misses with a set-associative cache than with a direct-mapped cache if both are the same size?
5. What are two reasons for using virtual memory?

6. How is the logical (virtual) address translated into the physical address in a virtual memory system? How does the TLB (Translation Look-aside Buffer) speed up this process?
7. In a virtual memory system with a TLB, what is the sequence of actions when a processor accesses data memory when a physically addressed cache is used? Describe the actions and how the sequence can result in a first-level cache hit or miss.

8. What is meant by "base" or "displacement" addressing, also called "register indirect with base or displacement"? Give an example of a ARMv8 instruction using this addressing mode. Give an example of a data structure which might be accessed conveniently using this addressing mode.

9. What is an "Instruction Set Architecture"?
10. Assume that we have the instruction mix shown below.

<table>
<thead>
<tr>
<th>Instruction</th>
<th>condition</th>
<th>frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-type</td>
<td></td>
<td>40%</td>
</tr>
<tr>
<td>Branch/jump</td>
<td>taken</td>
<td>15%</td>
</tr>
<tr>
<td>Branch/jump</td>
<td>not taken</td>
<td>15%</td>
</tr>
<tr>
<td>store word</td>
<td></td>
<td>15%</td>
</tr>
<tr>
<td>load word</td>
<td>no dependency</td>
<td>10%</td>
</tr>
<tr>
<td>load word</td>
<td>data dependency</td>
<td>5%</td>
</tr>
</tbody>
</table>

Consider the pipeline and multicycle architectures studied in class. Assume that for the pipeline architecture, a branch not taken incurs no penalty but a branch taken incurs a three cycle stall, and that data forwarding resolves all data hazards, except when the instruction following a load word depends on the data loaded, in which case a one cycle stall occurs. For (a) and (b) assume that memory accesses take one cycle.

a) What is the average CPI (cycles per instruction) for the multicycle CPU?

b) What is the average CPI for the pipeline CPU?

Now suppose that we include cache miss penalties into the calculation. Assume that both instruction cache and data cache misses occur on 2% of the memory accesses (reads and writes) and that a cache miss incurs a penalty of 10 clock cycles. (Note: every instruction is a potential instruction cache miss, but only sw and lw instructions can cause a data cache miss.)

c) What is the average CPI for the multicycle CPU, taking into account cache misses?

d) What is the average CPI for the pipeline CPU, taking into account cache misses?
11. Consider the following code segment.

\[ X_{10} = \text{starts at address of last element of array } A \]

\[ X_{19} = \text{length of array } A, \ X_{20} \text{ is base address of array } A \]

**loop:**
- \( \text{ldur } X_{11}, [X_{10}, 0] \) // \( X_{11} = A[k] \)
- \( \text{add } X_{12}, X_{10}, X_{19} \) // \( X_{12} = \text{address of } B[k] \)
- \( \text{ldur } X_{13}, [X_{12}, 0] \) // \( X_{13} = B[k] \)
- \( \text{add } X_{13}, X_{13}, X_{11} \) // \( X_{13} = A[k] + B[k] \)
- \( \text{sub } X_{10}, X_{10}, -8 \) // \( k-- \)

**test:**
- \( \text{sub } X_{9}, X_{10}, X_{20} \) // loop test
- \( \text{cbnz } X_{9}, \text{loop} \) // loop f not done

Assume that we work with the five-stage pipeline with accelerated branch and data-forwarding. Assume that if a branch is taken, there is a stall after the branch instruction.

(a) Insert the needed \text{nop}(s) and calculate the number of cycles for this loop for arrays with 1000 entries (assume no cache misses). Just indicate on the above listing where the \text{nop}(s) need to be inserted, with no reordering. Insert “stall if taken” for a branch. How many cycles are needed to execute the loop?

(b) Use loop unrolling (up to four times), register renaming, and reordering to reduce the total number of cycles needed for execution. How many cycles are needed now?
11 (c). Assume a dual pipeline, one pipe does memory instructions (ldur and sdur) only. The other does all other instructions. Use loop unrolling (up to four times), register renaming, and reordering to reduce the total number of cycles needed for execution. How many cycles?
12. In an SMP (Shared memory Processor) parallel machine, the processors address the same memory but have their own 1st level caches. What control bits are needed to provide a means for maintaining cache coherence using a “snoopy” bus protocol? How do these define three states for each cache line? How would a write to a given cache line by processor 1 affect the state of the same line in processor 2? How would a read of this line by processor 2 following the write by processor 1 affect the state of the line in processor 1?
13. Describe three types of hardware-supported multi-threading. What additional hardware is needed for each type?
14. (a) Describe how a SMP (Shared Memory Multi-processor) coordinates actions by several processors.

(b) Describe how a message-passing multiprocessor (cluster) coordinates actions by several processors.
15. State Amdahl's law and give a simple example. How does weak-scaling compensate for Amdahl's law for parallel processing?