1. The `nor & nand` instructions are not part of the RISC-V instruction set because the same functionality can be implemented using existing instructions. Write RISCV code that performs a `nor` operation on registers `x8` and `x9` and places the result in register `x10`. Write RISCV code that performs a nand operation on registers `x5` and `x6` and places the result in register `x7`
2. Convert the following high-level language script into RISCV code. Assume the signed integer variables `g` and `h` are in registers `x5` and `x6` respectively.

(i)

```
if (g > h)
     g = g + 1;
else
     h = h - 1;
```

(ii)

```
if (g <= h)
 g = 0;
else
 h = 0;
```

3. One difference between a write-through cache and a write-back cache can be in the time it takes to write. During the first cycle, we detect whether a hit will occur, and during the second (assuming a hit) we actually write the data.

   Let's assume that 50% of the blocks are dirty for a write-back cache. For this question, assume that the write buffer for the write through will never stall the CPU (no penalty). Assume a cache read hit takes 1 clock cycle, the cache miss penalty is 50 clock cycles, and a block write from the cache to main memory takes 50 clock cycles. Finally, assume the instruction cache miss rate is 0.5% and the data cache miss rate is 1%. Assume that on average 26% and 9% of instructions in the workload are loads and stores, respectively.

   Estimate the performance of a write-through cache with a two-cycle write versus a write-back cache with a two-cycle write

4. Consider the delays from the Table below. Now, suppose that the ALU were 20% faster. Would the cycle time of the pipelined RISCV processor change? What if the ALU were 20% slower? Explain your answers .

Component Delay	Delay
Register Delay (Clk to Q)	40
Register Setup	50
Multiplexer	30
AND-OR gate	20
ALU	120
Decoder (Control Unit)	25
Sign Extend Unit	35
Memory Read	200
Register File Read	100
Register File Setup	60

5. Your Company describes their latest Processor with the following features:
• 95% of all memory accesses are found in the cache.
• Each cache block is two words, and the whole block is read on any miss.
• The processor sends references to its cache at the rate of 10^9 words per second.
• 25% of those references are writes.
• Assume that the memory system can support 10^9 words per second, reads or writes.
• The bus reads or writes a single word at a time (the memory system cannot read or write two words at once).
• Assume at any one time, 30% of the blocks in the cache have been modified.
• The cache uses write allocate on a write miss.

You are considering adding a peripheral to the system, and *you want to know how much of the memory system bandwidth is already used*.

5.1 Calculate the percentage of memory system bandwidth used assuming the cache is Write Back.

5.2 Calculate the percentage of memory system bandwidth used assuming the cache is Write Through.

Please be sure to state your assumptions and show all work

6. Assume that you have a computer with 1 clock cycle per instruction (CPI=1) when all accesses to memory are in cache. The only accesses to data come from load and store instructions. Those accesses account for 25 % of the total number of instructions. Miss penalty is 50 clock cycles and miss rate is 5 %. Determine the speedup obtained when there is no cache miss compared to the case when there are cache misses

7. Suppose that when Program A is run, the user CPU time is 3 seconds, the elapsed wall clock time is 4 seconds, and the system performance is 10 MFLOP/sec. Assume that there are no other processes taking any significant amount of time, and the computer is either doing calculations in the CPU, or doing I/O, but it can't do both at the same time. We now replace the processor with one that runs six times faster, but doesn't affect the I/O speed. What will the user CPU time, the wall clock time, and the MFLOP/sec performance be now?

8. Mark each statement below as true or false. Explain your reasoning. Provide a counterexample if the statement is false.

   (a) A two-way set associative cache always has a lower miss rate than a direct mapped cache with the same block size and total capacity.

   (b) A 16 KiB direct mapped cache always has a lower miss rate than an 8 KiB direct mapped cache with the same block size.

   (c) An instruction cache with a 32-byte block size usually has a lower miss rate than an instruction cache with an 8-byte block size, given the same degree of associativity and total capacity.