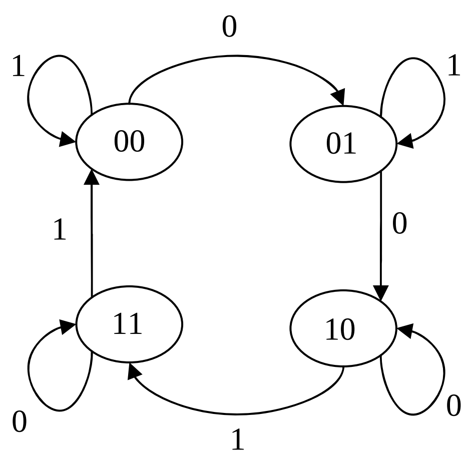
COS 116
The Computational Universe
Homework 4

Q1. Consider the following 4-state, 1-input finite state machine:

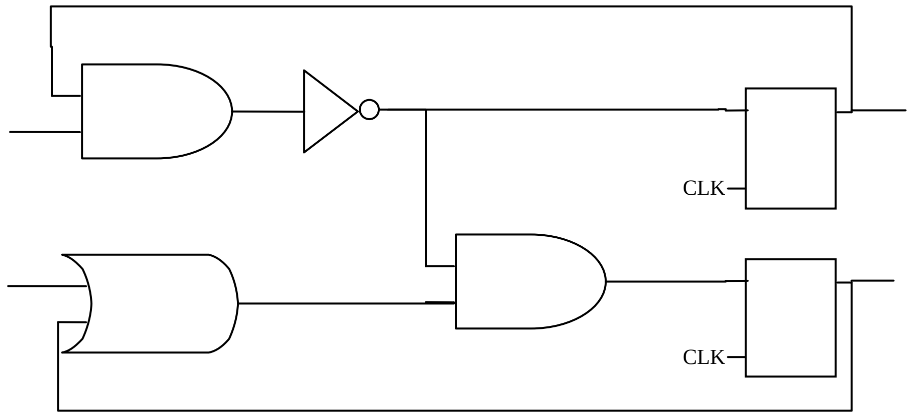


a) Give the truth table for this finite state machine. The table should specify the next state for each combination of input and current state.
b) Draw a circuit design that implements this finite state machine. Your circuit will consist of a combinational circuit and some "rising-edge triggered" D flip-flops (like you used in lab). The combinational portion should implement the truth table from part (a). Make your circuit as simple as you can.

Q2. A computer manufacturer lists its processor speed as 3.2 GHz. Explain what they mean by this and comment in a line or two in what ways it is a meaningful measure of speed and in what ways it is not.

Q3. The *propagation delay* of a gate is the time taken for the digital signal to travel from the input of the gate to the output. Before this delay has elapsed, the output of a gate is not stable, and may be incorrect.

Now consider the following sequential circuit. Note that the write controls of the "rising-edge triggered" D-flip-flops (like you used in lab) are all connected to a clock signal.



On every rising edge of the clock signal, the inputs to the flip-flops are written to the outputs. The outputs are then fed back into the circuit. To ensure proper functioning of the circuit, the new inputs to the flip-flops must be stable before the next rising edge of the clock signal.

Suppose the propagation delay of an AND/OR gate is 2 nanoseconds (2×10^{-9}), and the delay of a NOT gate is 1 nanosecond. Also, assume that flip-flops have zero delay. Calculate the maximum frequency for the clock signal which still ensures that this circuit will function correctly.

Q4. Suppose we have a computer with a main memory that can hold 10 data items, and a cache that can hold 3 data items. Initially, the main memory contains the numbers 1 through 10, and the cache is empty:

<u>Main memory</u>:

1	2	3	4	5	6	7	8	9	10

<u>Cache</u>:

Recall that, whenever a program requests a data item, it first checks to see whether it is in the cache. If it is, the request proceeds normally. If it's not, the data item is first copied from the main memory to the cache. If the cache is full, some other data item in the cache must be overwritten.

Assume our computer uses the "Least Recently Used" algorithm to decide which data item will be overwritten. Now suppose we run a program that requests data items in the following sequence: 2, 4, 7, 2, 1, 3, 4, 2, 4, 10, 4, 9, 2, 5, 2, 2. Show the contents of the cache after each request. Also, say how many times during this sequence the requested item was not in the cache.

Q5. Consider a computer with three levels of memory: cache, RAM, and hard disk. The table below gives the time required to access a data item from each of these memories.

Memory Type	*Access Time*
Cache	5×10^{-9}s
RAM	2×10^{-8}s
Hard disk	5×10^{-3}s

Suppose that 98% of data requests are satisfied by the cache, 1.9% by the RAM, and 0.1% by the hard disk. Calculate the average time to access a data item in this computer.