Name: _____

# ES 4 sequential logic practice problems

After solving the problems, look at the solutions posted on the course website and categorize your work for each problem on the following scale:

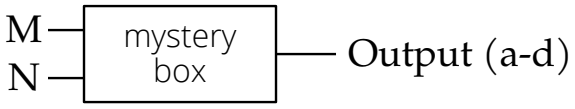
- ● Completely correct
- ◕ Nearly correct, but made a small mathematical or copying error
- ◒ Solved part of the problem correctly
- ◔ Started some work in the right direction
- ○ Incorrect, or didn't even know where to start on the problem
- Include a question mark (?) in addition to one of the above symbols if you don't feel like you understand the question or the solution well enough to make a definite judgement.

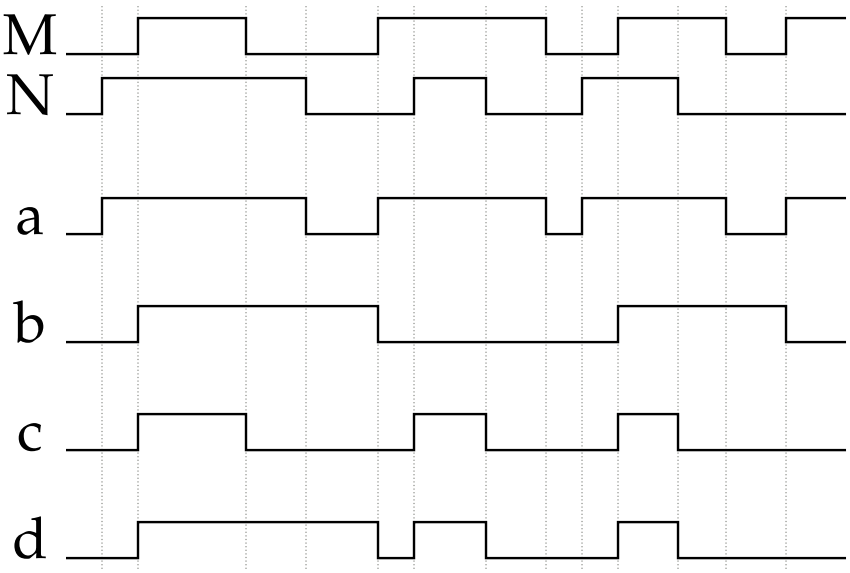
You do not need to turn anything in for these problems, although we're happy to discuss your work and answer any questions you have!

| 1 | 2 | 3* | 4 | 5 | 6* | 7 |
|---|---|---|---|---|---|---|
| | | | | | | |

## Part 1: What's in the box?

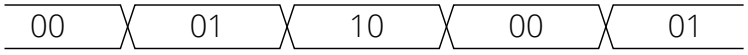
For each of the four waveforms below (a-d), identify what is in the box. It could be an SR latch, a D latch, a D flip-flop, or a single combinational logic gate (NAND, XOR, etc).



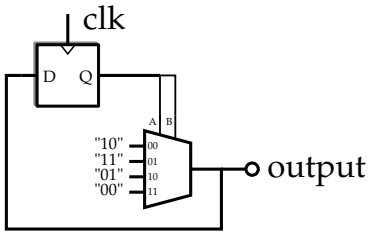



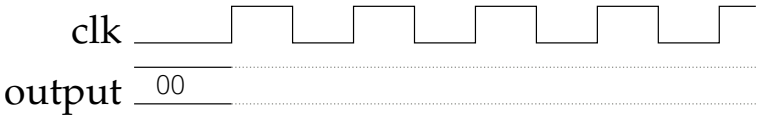
## Part 2: A sequential circuit

Below is a small sequential circuit which contains a 2-bit register. Draw its output as a function of time on the waveform diagram below, assuming it starts at "00". The conventional way to draw a bus of signals is to draw the signal as both high and low (since some bits are high and others are low), and to write the numerical value (in whatever base) in the spaces in between:



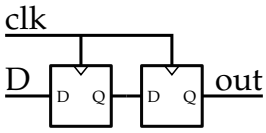

*Note: The multiplexer is actually two multiplexers stacked together, which collectively drive the bus. The same select lines go to each multiplexer, so the result on the output is the bit pattern shown on the left. This is how Radiant (and other tools) typically draw multiplexed buses.*



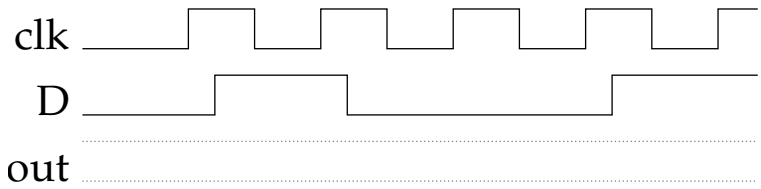



## Part 3: Another sequential circuit

The circuit below is a 2-bit shift register (briefly discussed in section 5.4.2 of the textbook).

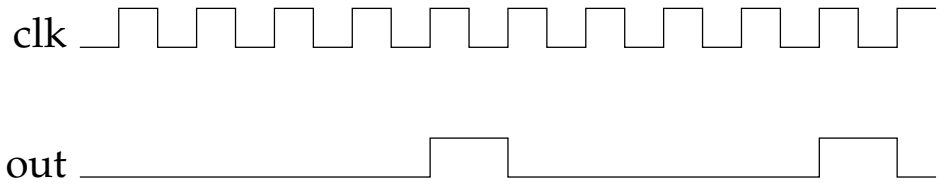



Draw the output as a function of the input signals below. Be sure to clearly mark the section of time when the output is unknown.



### Part 4: 5th-cycle high

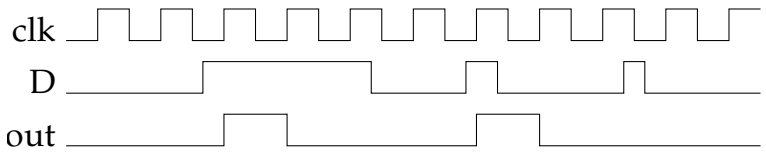
Design a circuit which takes a clock signal and produces an output which is high every fifth cycle:




Complete the corresponding problem on VHDLweb to implement your design. The logic diagram synthesized from VHDL should match the logic diagram you drew on paper.

## Part 5: Clocked rising-edge detector

Design a circuit which takes a clock signal and a data input, and produces an output which is high for one cycle whenever the data has a rising edge (i.e., was low for the previous clock cycle and high for this cycle). Note that pulses on D which are too short to be caught by a rising clock edge should be ignored.




*This circuit is super useful when you want to register button presses or other external events, but only want to count them once, instead of incrementing every time that the input is high and the clock has a rising edge.*

Complete the corresponding problem on VHDLweb to implement your design. The logic diagram synthesized from VHDL should match the logic diagram you drew on paper.

## Optional book problems

These are selected problems from the book which may be helpful for practice and review. The answers to these problems are online at `https://booksite.elsevier.com/9780128000564/solutions.php`

- 3.1, 3.3, 3.5 (Latches and flip-flops)
- 5.47 (Up-down counter)
- 5.48/5.49 (A load-able counter) *Note: The book doesn't say it, but what you're building here is actually a simple but crucial piece of a processor.*
- 5.51 (Scan-chain flip-flops)