Name: _____

ES 4 FSM practice problems

Practice problems - For Review

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`.

- 3.23, 4.33 (Designing logic for an FSM, VHDL implementation)
- 3.25, 4.35 (Mealy FSM, VHDL implementation)
- 3.29, 4.39 (Deriving an FSM from equations, VHDL ipmlementation)
- 4.25 (FSMs in VHDL)

Gene analysis

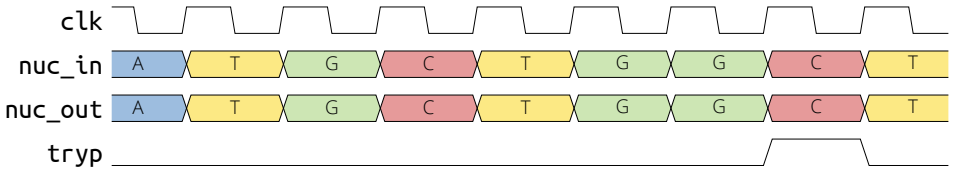
The next few problems are about gene analysis. DNA consists of chains of four nucleotides which form a coded sequence to create protiens or control operations within the cell. We can represent each of the four nucleotides with 2 bits:

Adenine	A	00
Thymine	T	01
Guanine	G	10
Cytosine	C	11

In each problem below, you'll design a system that reads in a sequence of nucleotides and analyzes it in some way. Since genomes are huge (hundreds of megabytes) and your hardware is small, you can't store the sequence. Instead, the sequence will be "streamed" in: on each clock cycle the next nucleotide is presented at the input, and a few clock cycles later your hardware should present it at the output, along with any signals indicating the resulting analysis.

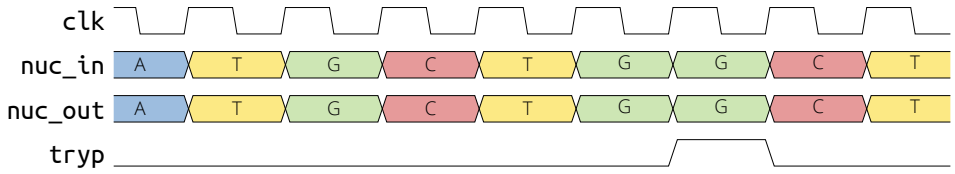
Part 1: Detecting tryptophan — Moore

(a) Draw the state diagram for a Moore state machine which detects the code 'TGG', which codes for the amino acid tryptophan. The output `nuc_out` should be a copy of the input, with no delay. The `tryp` output should be high for one clock cycle after the sequence is detected:



Part 2: Detecting tryptophan — Mealy

(a) Draw the state diagram for a Mealy state machine which detects the code 'TGG'. The output `nuc_out` should be a copy of the input, with no delay. The `tryp` output should be high for one clock cycle, synchronized with the second 'G':



Part 3: Comparing Moore and Mealy

Take a moment to compare your designs for the two previous problems, and answer the questions below.

(a) Which machine has more states? Assuming the synthesis tool uses a binary encoding, does this require extra flip-flops?

(b) Which machine has simpler output logic?

(c) Which machine computes the result faster (i.e., in fewer clock cycles)?

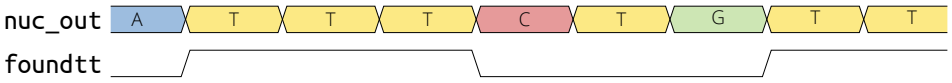
Part 4: Evaluating skin cancer risk

A common cause of skin cancer is when two adjacent thymine ('T') base pairs are fused together by an incoming ultraviolet ray. This causes the strand to be mis-read, which under the right circumstances causes the cell to go haywire.

(a) Draw the state diagram for a **Mealy** state machine which detects adjacent thymine base pairs. As before, the input should be echoed to the output, possibly with some delay.

The `doubleT` signal should be high whenever the module emits a 'T' which is part of a pair (i.e., it should be high for two cycles when a pair comes out). *Hint: This means you'll need some delay. You can consider any values in the delay chain to be FSM inputs; this will greatly simplify your FSM.*

It should also be able to handle multiple adjacent pairs (e.g., 'TTT' or 'TTTT').



(b) Implement your state machine in VHDL.