Indian Institute of Technology Delhi
ELL 201: Digital Electronic Circuits Laboratory
Experiment 7: Designing Counters

October 16, 2019

Counters are used in a variety of applications in the digital domain. In this experiment, you will be learning the design of two types of counters.

- Synchronous 4-bit Gray-Code Counter. A Gray-Code counter counts in such a manner that the difference between any two consecutive states only differs by 1 bit. It is used in a variety of scenarios especially where bit variations/ state changes are prohibitive in terms of resource costs. // You can count in any manner, but it should cover all 16 states of the counter in a cyclic manner.
  - First, make a state table for the scheme that your counter follows.
  - Calculate the number of SR Flip-Flops required.
  - Assign values to the inputs S,R for each of the Flip-Flops.
  - Use Karnaugh Maps to achieve minimized expressions for the inputs.
  - Write Verilog/ VHDL code for this counter.

  (Hint: Make an entity for SR Flip-Flop, use these as components in your counter, and assign inputs to these Flip-Flops depending on the minimized expressions achieved as a function of the current outputs. Ensure your Flip-Flop is either positive- or negative- edge triggered, and does not function as a Latch)

- Synchronous Ring Counter A Ring counter behaves like a Shift-Register, except that the first Flip-Flop gets an input which can be a combination of any of the current states.
  - First, calculate the number of D Flip-Flops required.
  - Assign values to the inputs D for each of the Flip-Flops looking at the state table (Fig. 1). Note that for the intermediate ip-ops, the behaviour is the output of one fed as input to the other. Be careful about the input of the MSB of the states.

– When implementing the counter, let it start from the state 0001.
– Write Verilog/ VHDL code for this counter.

Q3	Q2	Q1	Q0	D3	D2	D1	D0
0	0	0	1	1	0	0	0
1	0	0	0	0	1	0	0
0	1	0	0	0	0	1	0
0	0	1	0	1	0	0	1
1	0	0	1	1	1	0	0
1	1	0	0	0	1	1	0
0	1	1	0	1	0	1	1
1	0	1	1	0	1	0	1
0	1	0	1	1	0	1	0
1	0	1	0	1	1	0	1
1	1	0	1	1	1	1	0
1	1	1	0	1	1	1	1
1	1	1	1	0	1	1	1
0	1	1	1	0	0	1	1
0	0	1	1	0	0	0	1

Figure 1: State Table for Synchronous Ring Counter

(Hint: Make an entity for D Flip-Flop, use these as components in your counter, and assign inputs to these Flip-Flops depending on the behaviour deduced from the state table. Ensure your Flip-Flop is either positive- or negative- edge triggered, and does not function as a Latch)
Does the counter cover all 16 possible states? Try to figure out what the counter is trying to do, and how it depends on which state the counter starts from.