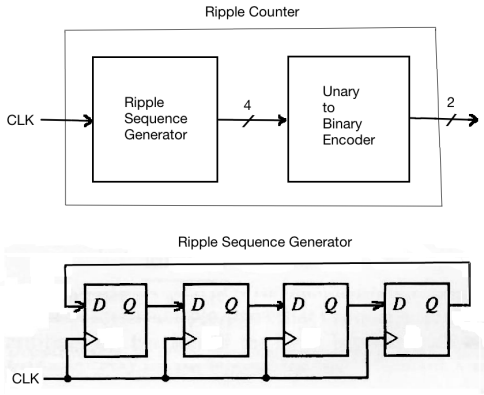
CSc 137 Computer Organization
Sacramento State, Spring 2019
Verilog HDL Assignment 4
Sequential Logic: Counter

- The use of memory constitutes a sequential-logic circuit. The memory in Verilog programming is the use of `reg` to declare bit storage which can contain a *state* of the electronics, and can be used as part of the input to generate the next state in time driven by clock cycles. Simple examples of sequential circuits are counters.

- This assignment we will code a Verilog programs for simulation of a Ripple Counter which is one of various types of counters. The basic code and runtime output are in **Demo Folder**.

- The way a ripple counter is constructed: a single value 1 is being passed among several bit registers, and the collective output of these registers will thus form a *unary* pattern. This pattern is then given to an *encoder*, and the output of the encoder will be binary. That is said to be a binary two-bit counter: 1000->00_2, 0100->01_2, 0010->10_2, 0001->11_2, and repeat.



Suppose the output of the 4 registers above are Q0, Q1, Q2, and Q3. Then, the truth table for encoding:

```
Q0 Q1 Q2 Q3  C1 C0
1  0  0  0   0  0  Therefore, C1 is 1 when either Q2 or Q3 is 1; otherwise 0.
0  1  0  0   0  1  And, C0 is 1 when either Q1 or Q3 is 1; otherwise 0.
0  0  1  0   1  0  This defines how the encoder module processes internally.
0  0  0  1   1  1
```

- Your goal is to program **a counter that counts from 0 to 15** and repeat. (Expand from count4.v which counts from 0 to 3 and repeat). That is , it counts binary: 0000 to 1111, and then back to 0000. See the demo to know the runtime period to stop the repetition.

- You **MUST USE ARRAYS** for registers or wires. The only single-bit item is the clock signal *CLK* as shown in the given code.

- See the given example code to program the clock signal generation correctly. In the ripple counter, all counting registers are initially set zero except the first one is 1 so to pass it to the next register upon a new clock cycle. Your Verilog programs should generate the same output as how the demo runs.

- Submit your source file only into your login-named folder (that you make yourself) under the usual dropbox 4thPrgAssig folder on host Voyager. Again, if resubmitting a corrected version is needed, use command at the smb prompt: "mkdir V2" to make a new folder and to put in there. If still more correction, use V3, V4, etc. as a new folder name.