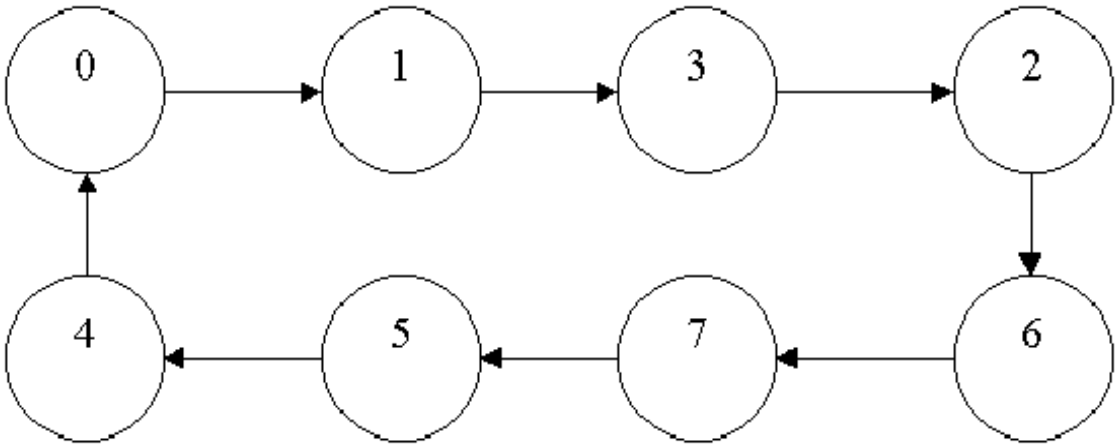
**CS150**
**Fall 1998** **by Brian Choi**

# Problem Set #3 Solutions

## 1. Counter Design

**State Diagram:** The current state represents the output of the system.




**State Table:** The Q[2:0] lines represent the current state as well as the output of the counter.

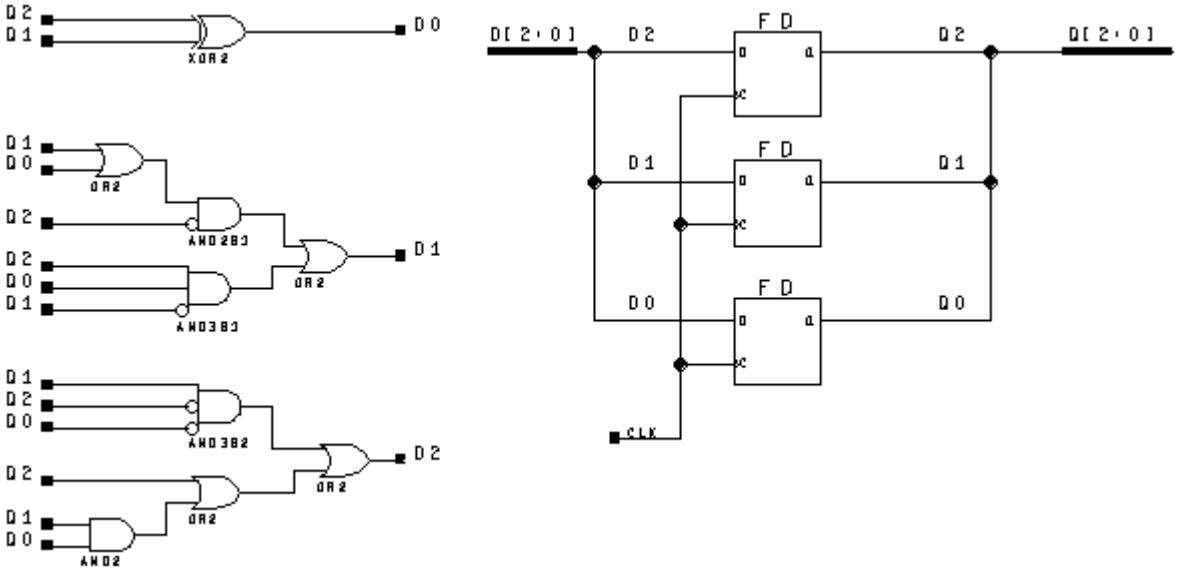
| Q2 | Q1 | Q0 | D2 | D1 | D0 |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 |

$D0=\overline{Q2 \oplus Q1}$

$D1=\overline{Q2}\cdot(Q1+Q0) + Q2\cdot Q1\cdot\overline{Q0}$

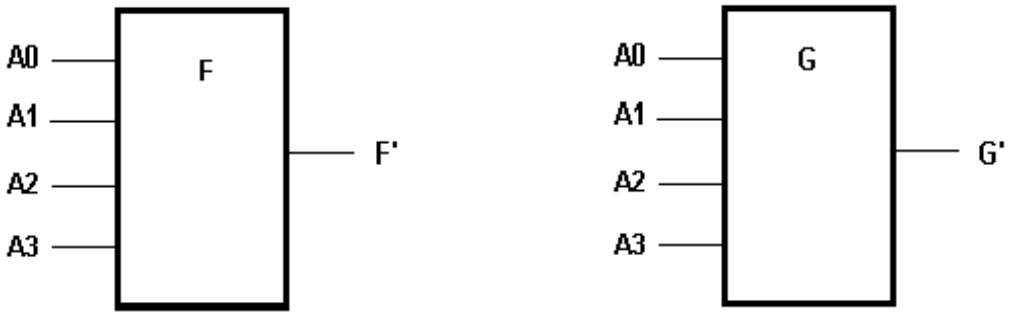
$D2=\overline{Q2}\cdot Q1\cdot\overline{Q0} + Q2\cdot(Q1+Q0)$

**Schematic:**

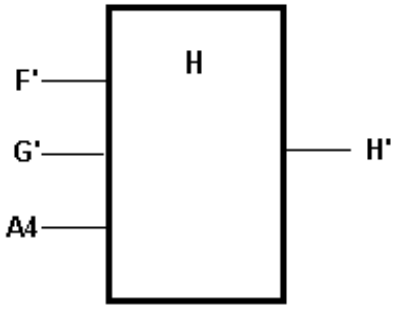



## 2. Xilinx Combinational Logic Block (CLB)

The output of each LUT (Look Up Table) will be represented with a (‘) (e.g. the output of the F LUT will be F’). Let the 5 inputs be shown as signals A[4:0]. Assign the first four inputs as shown below.




F’ and G’ are functions of 4 variables and independent of each other. These are fed into the H block along with the fifth input. As shown in the next figure.




The H block can opt to pass either F' or G', depending on the Value ofA4. This can be modeled by the expression:

**$H' = (\overline{A4}) \cdot F' + (A4) \cdot G'$**

This effectively creates a boolean function of 5 variables.

**Truth Table Visualization**

| A4 | A3 | A2 | A1 | A0 | Output | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | ? | |
| 0 | 0 | 0 | 0 | 1 | ? | |
| 0 | 0 | 0 | 1 | 0 | ? | |
| 0 | 0 | 0 | 1 | 1 | ? | F' |
| | | : | | | ? | |
| | | : | | | ? | |
| 0 | 1 | 1 | 1 | 0 | ? | |
| 0 | 1 | 1 | 1 | 1 | ? | |
| 1 | 0 | 0 | 0 | 0 | ? | |
| 1 | 0 | 0 | 0 | 1 | ? | |
| 1 | 0 | 0 | 1 | 0 | ? | |
| | | : | | | ? | G' |
| | | : | | | ? | |
| 1 | 1 | 1 | 1 | 0 | ? | |
| 1 | 1 | 1 | 1 | 1 | ? | |

Inputs A3-A0 will go to both the "F" and "G" LUT. The A4 input will go the "H" LUT and decide which of the F' or G' output to follow for the final H' output. This shows that a single CLB can produce a function for 5 variables.

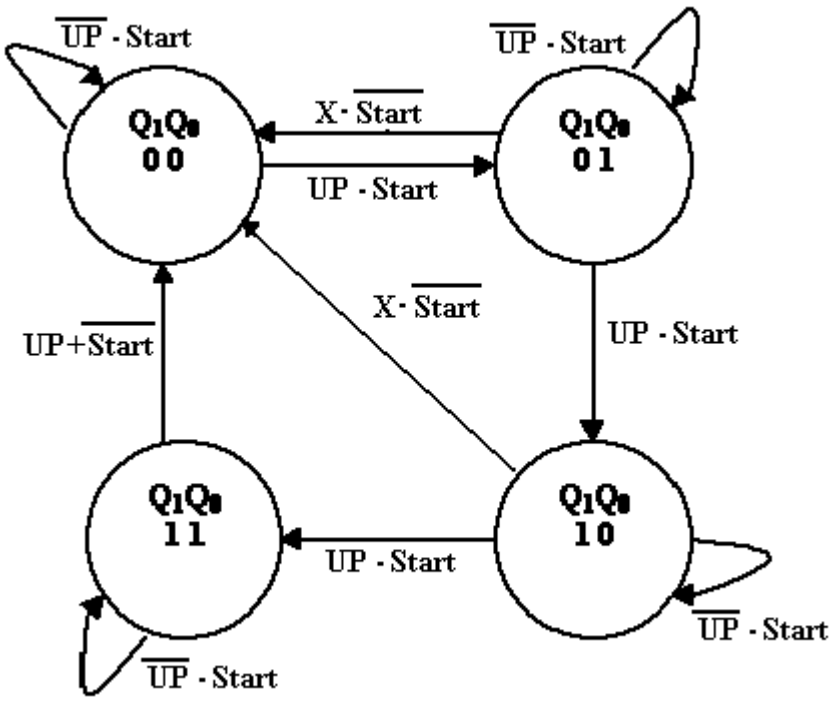
## 3. FSM Analysis

**State Transition Table**

| Start | Q1 | Q0 | A0 | D1 | D0 |
|---|---|---|---|---|---|
| 0 | X | X | X | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 |

If **start** is not asserted, then the decoder will not function and the system will return to the zero state.

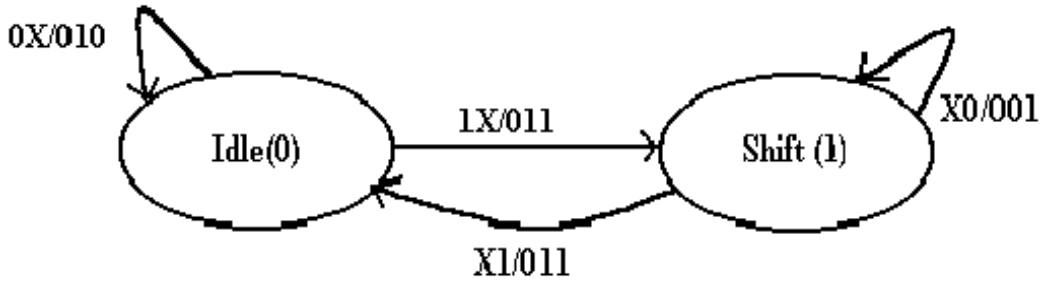
**State Diagram**




The system will go to the next state when up and start are asserted. This machine is nothing more than a really fancy counter.

## 4. Parallel-to-Serial Converter

**State Diagram**




The bit-fields for the state transitions are as follows

**Start,CNT7/Done, Init, Count**

**Start = Start signal is selected**

**CNT7 represents an internal signal where the counter has reached the value 7 (111/b). Since thecounter starts at 0, 8 bits will pass when the counter hits 7.**

**Done is the Done signal**

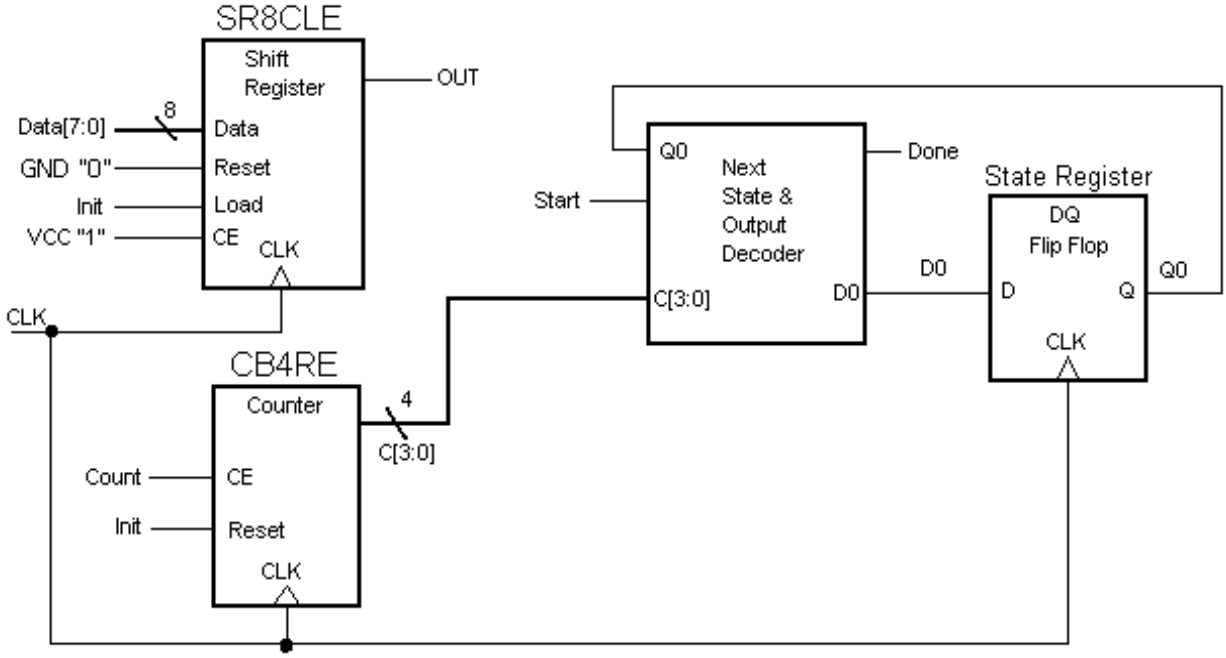
**Init will reset the the counter and load the shift register.**

**Count is the clock enable line for the Counter.**

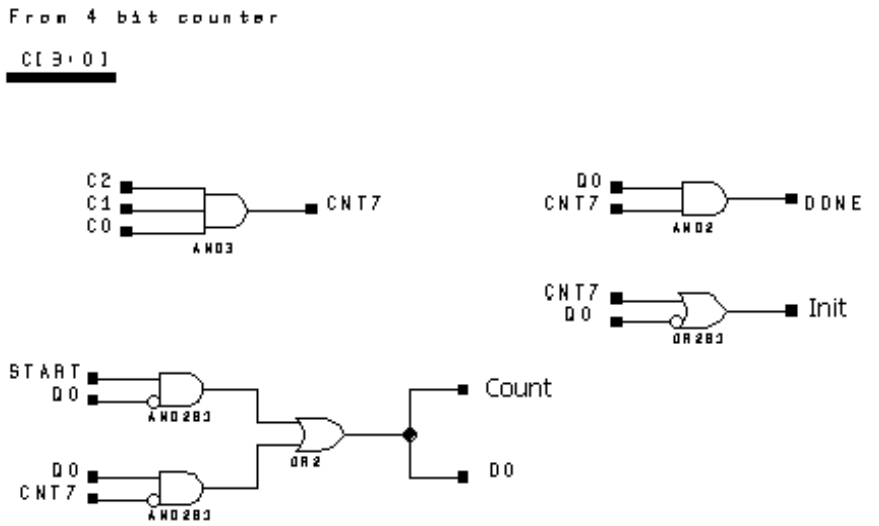
**State Transition and Output Table**

| Start | CNT7 | Q0 | Done | Init | Count | D0 |
|---|---|---|---|---|---|---|
| 0 | X | 0 | 0 | 1 | 0 | 0 |
| 1 | X | 0 | 0 | 1 | 1 | 1 |
| X | 0 | 1 | 0 | 0 | 1 | 1 |
| X | 1 | 1 | 1 | 1 | 0 | 0 |

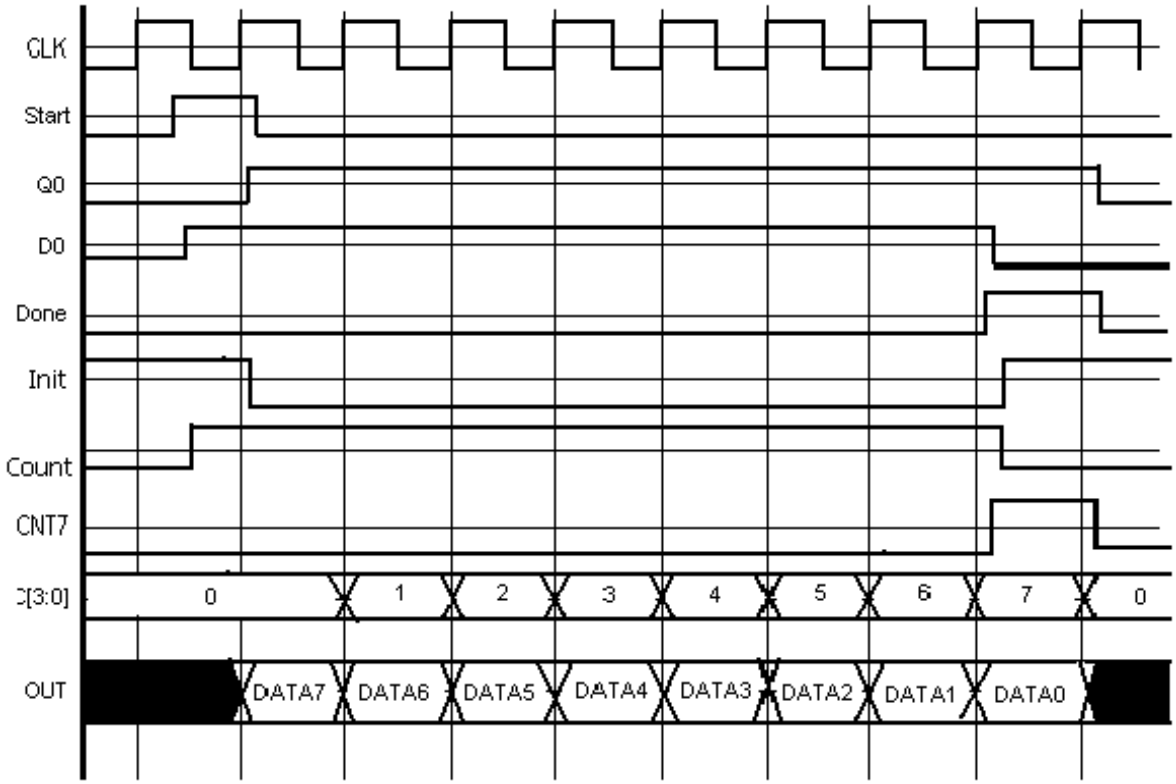
**Schematics: Datapath**



## Schematics: Next State And Ouptut Decoder




## Timing:



The Parallel to Serial converter is a fairly simple machine, which only requires two states. When Start is asserted, on the next clock cycle, the shift register will load in the data values and output the bits. The “OUT” signal is actually the highest index output bit from the shift register (Q7). The counter is then started. On each subsequent cycle, the lower bits are shifted up and the highest bit is assigned as the OUT signal. While the system is shifting, a counter is counting from 0 to s7. When it hits seven, the CNT7 is asserted (by means of simple combinatorial logic). When CNT7 is asserted, all 8 bits have been sent through the line. The system will assert done, then go back to the idle state where it will wait for another start signal.