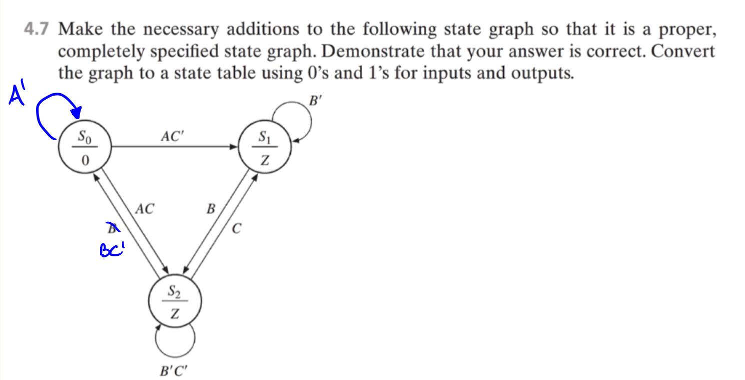
Engr 378

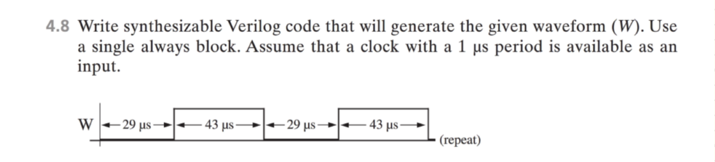
Homework #8

Problems 4.7, 4.8 & 4.10



**4.7** Make the necessary additions to the following state graph so that it is a proper, completely specified state graph. Demonstrate that your answer is correct. Convert the graph to a state table using 0's and 1's for inputs and outputs.

|  | ABC |  |  |  |  |  |  |  | Z |
|---|---|---|---|---|---|---|---|---|---|
| State | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 | output |
| $S_0$ | $S_0$ | $S_0$ | $S_0$ | $S_0$ | $S_1$ | $S_2$ | $S_1$ | $S_2$ | 0 |
| $S_1$ | $S_1$ | $S_1$ | $S_2$ | $S_2$ | $S_1$ | $S_1$ | $S_2$ | $S_2$ | 1 |
| $S_2$ | $S_2$ | $S_1$ | $S_0$ | $S_1$ | $S_2$ | $S_1$ | $S_0$ | $S_1$ | 1 |



**4.8** Write synthesizable Verilog code that will generate the given waveform ($W$). Use a single always block. Assume that a clock with a 1 μs period is available as an input.

```
Modole waveForm(w);
 outpot w;
 reg state, clk;
   integer count
   initial begin
      state = 0;
      count = 0;
      clk = 0;
   end
 assign w = state;
  always begin
   #5 = ~clk; // 1μs clock period
  end
  ...
```

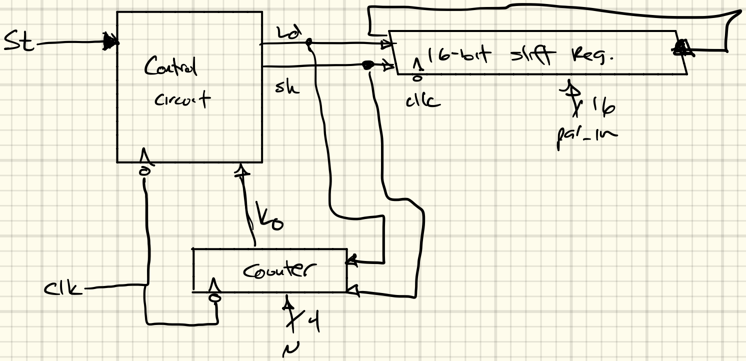
```
... always @(negedge clk)
      begin
        case(state)
      0: begin
    if(count == 29) begin
      count <= 1;
      state <= 1;
    end
    else begin
      count <= count + 1;
   end
   end
 1: begin
 ...
```

```
 ...
if(count==43) begin
count <= 1;
state <= 0;
end
else begin
count <= count + 1;
end
end
end case
end
end module
```

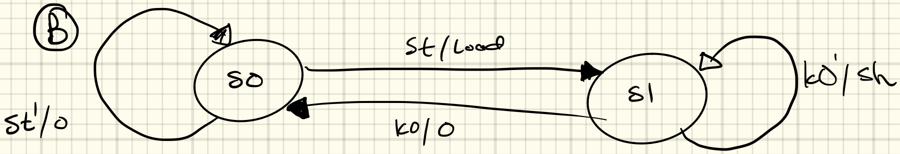
4.10 Write Verilog code for a shift register module that includes a 16-bit shift register, a controller, and a 4-bit down counter. The shifter can shift a variable number of bits depending on a count provided to the shifter module. Inputs to the module are a number *N* (indicating shift count) in the range 1 to 15, a 16-bit vector *par_in*, a clock, and a start signal, *St*. When *St* = 1, *N* is loaded into the down counter, and *par_in* is loaded into the shift register. Then the shift register does a cycle left shift *N* times, and the controller returns to the start state. Assume that *St* is only 1 for one clock time. All operations are synchronous on the falling edge of the clock.

**(a)** Draw a block diagram of the system and define any necessary control signals.
**(b)** Draw a state graph for the controller (two states).
**(c)** Write Verilog code for the shift-register module. Use two always blocks (one for the combinational part of the circuit and one for updating the registers).

A



B


S0
S1
St/Load
K0/0
K0'/sh
St'/0