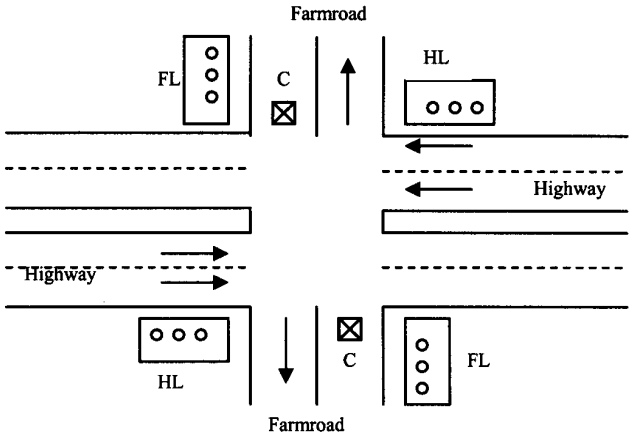
answers the phone (i.e. when A goes to 1). If a person answers the phone at any point, A will become 1, and the circuit should reset. Assume that S is not changed while the phone is counting rings.

(a) **(20 points)** Give a **Moore** state graph and state table for this circuit. (Note: A "state graph" is the same as a "state diagram".)
(b) **(20 points)** Write a Verilog module to implement the Moore machine from Part (a) in **behavioral Verilog**.

Problem 24 (60 points):
A busy highway is intersected by a little-used farmroad, as shown in the figure below. Detectors are placed along the farmroad to raise the signal C as long as a vehicle is waiting to cross the highway. The traffic light controller should operate as follows. As long as no vehicle is detected on the farmroad, the lights should remain green in the highway direction. If a vehicle is detected on the farmroad, the highway lights should change from yellow to red, allowing the farmroad lights to become green. The farmroad lights stay green only as long as vehicle is detected on the farmroad and never longer than a set interval to allow the traffic to flow along the highway. If these conditions are met, the farmroad lights change from green to yellow to red, allowing the highway lights to return to green. Even if vehicles are waiting to cross the highway, the highway should remain green for a set interval. You may assume there is an external timer that, once set via the control signal ST (set timer), will assert the signal TS after a short time has expired (used for timing yellow lights) and TL after a long interval (for green lights). The timer is automatically reset when ST is asserted.



We want to implement this traffic light controller as a **Mealy machine**.

(a) **(5 points)** Understand the problem specification. Clearly define the input and output signals that you use and give a description in English, of what each signal does or achieves.

(b) **(5 points)** Define the states that you will use and give a description of each state. Try to use the minimum number of states that you can. (No formal state minimization procedure is required in this problem. If you do not use the minimum number of states, it will be fine, too.)

(c) **(10 points)** Draw the state diagram.

(d) **(10 points)** Draw the state table.

(e) **(10 points)** Implement this controller using only D flip-flops. (Draw the schematic.)

(f) **(20 points)** Write a Verilog module to implement this Mealy machine in **behavioral Verilog**.

(Notes: In practice, we write the behavior of the machine like this in Verilog and rely on synthesis tools to synthesize a circuit which will automatically produce a circuit such as in part (e). Since this example is simple, we can do the D flip-flop implementation by hand as in Part (e) and see both approaches. It is illustrative to compare the code in Part (f) with the implementation in Part (e).)

Problem 25:

Reduce the following state table to a minimum number of states.

(a) **(10 points)** First, use row matching to reduce as far as you can.

(b) **(10 points)** Then, use the implication chart method on the output of the row matching procedure and perform any further reductions.

Present State	Next State X=0	Next State X=1	Present Output X=0	Present Output X=1
A	A	E	1	0
B	C	F	0	0
C	B	H	0	0
D	E	F	0	0
E	D	A	0	1
F	B	F	1	1
G	D	H	0	1
H	H	G	1	0

(c) **(15 points)** You are given two identical sequential circuits which realize the preceding state table. One circuit is initially in state B and the other circuit is initially in state G. Specify an input sequence of length three which could be used to distinguish between the two circuits and give the corresponding output sequence from each circuit.

CR# 24

Problem :- 8

(a)

Input Signal	Description
Reset	Place controller in initial state
C	Detects vehicle on farmroad in either direction
TS	Short timer interval has expired
TL	Long timer interval has expired.

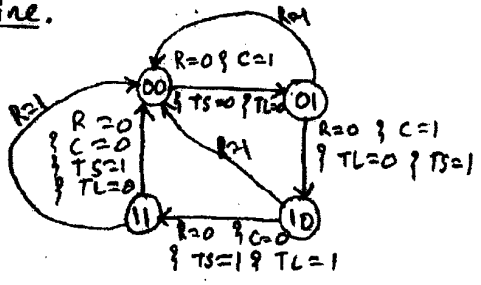
Output signal	Description
HG, HY, HR	Assert green, yellow & red highway lights
FG, FY, FR	Assert green, yellow & red farmroad lights
ST	Commence timing a long or short interval

(b)

S0 → highway green (& farmroad red)
S1 → highway yellow (& farmroad red)
S2 → farmroad green (& highway red)
S3 → farmroad yellow (& highway red)

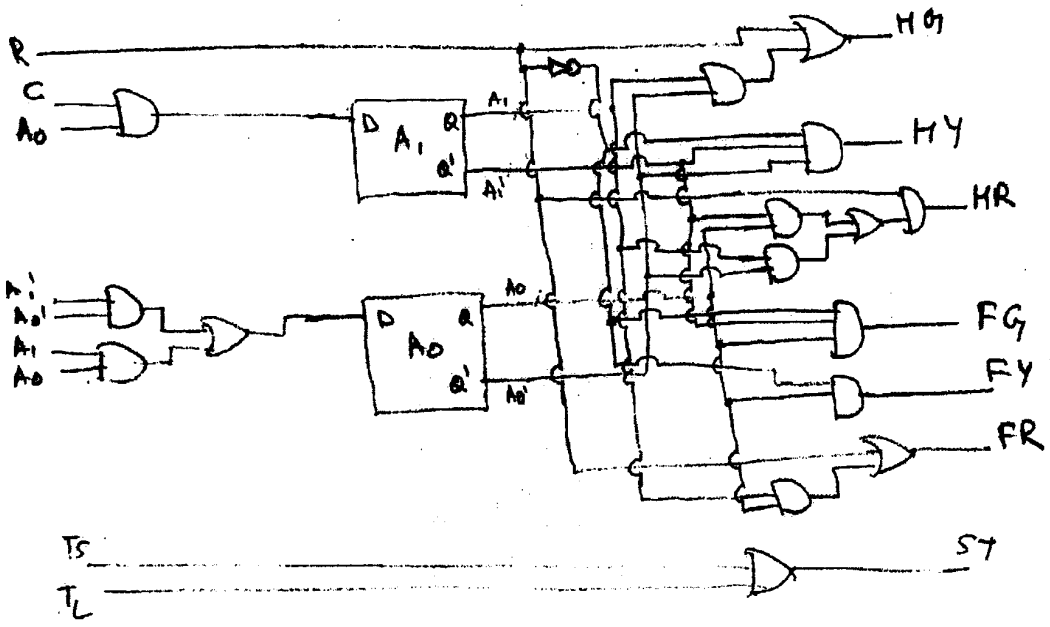
(c) State diagram :-

To represent 4 states, shown in (b), we need two bits, viz A_1 & A_0. Also, we are designing a Mealy machine.



CR# 24

Schematic :-



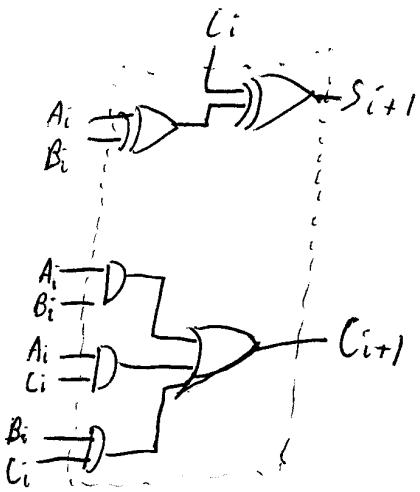
CR #24 Verilog

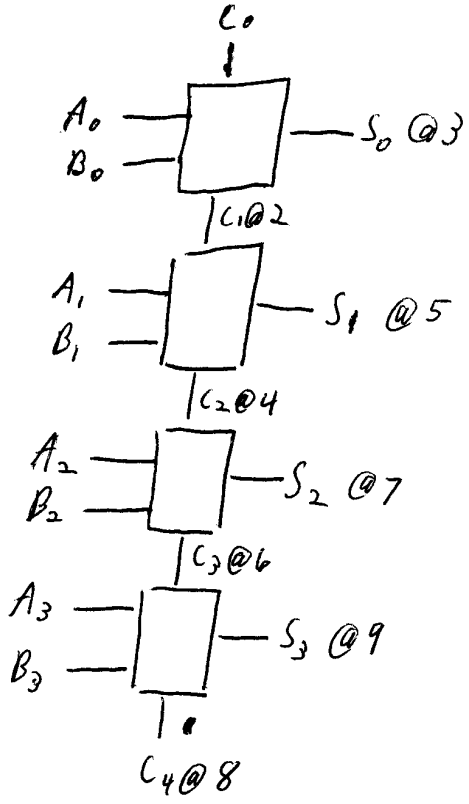
```
Module  traffic light Controller(Reset, C, TS, TL, HG, HY, HR, FG, FY, FR,
                                 CLK, ST);
Input Reset, C, TS, TL, CLK;
Output HG, HY, HR, FG, FY, FR, ST;
reg [1:0] state;


always @ (posedge CLK)
begin
    case x (state)
        2'b00 : if (R==0 & C==1 & TS==0 & TL==0) state <= 2'b01;
        2'b01 : if (R==0 & C==1 & TL==0 & TS==1) state <= 2'b10;
                else if (R==1)                   state <= 2'b00;
        2'b10 : if (R==0 & C==0 & TS==1 && TL==1) state <= 2'b11;
                else if (R==1)                    state <= 2'b00;
        2'b11 : if (R==0 & C==0 & TS==1 && TL==0) state <= 2'b00;
                else if (R==1)                    state <= 2'b00;
    end
end Module;
```

Adders

(a) Build a 4 bit Carry Ripple Adder with AND, OR, INV having delay 1 and XOR having delay 3. Minimize critical path.





(b) what is the critical path delay? $t = 9$

(c) Construct a 3-bit carry ~~save~~ lookahead adder. What is the critical path?

$P_i = A_i \oplus B_i$ $\quad G_i = A_i \cdot B_i$

$C_i = G_i + P_i C_i$

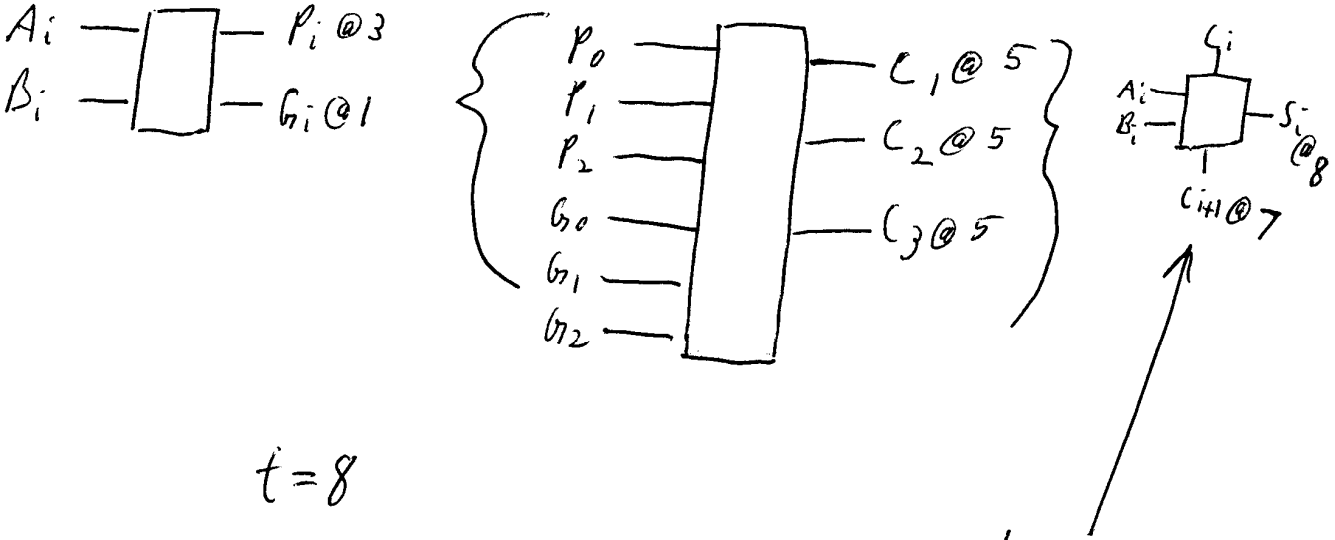
$C_1 = G_0 + P_0 C_0$

$C_2 = G_1 + P_1 C_1 = G_1 + P_1 G_1 + P_1 P_0 G_0$

$C_3 = G_2 + P_2 C_2 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$

~~A_0~~
~~B_0~~
~~A_1~~
~~B_1~~

Critical Path:



$t = 8$

delay of 1-bit carry ripple adder

(d) will the 3rd bit of the carry look ahead be valid before 3rd bit of the carry ripple adder?

no $S_{2,CLA}$ @8 . $S_{2,CRA}$ @7

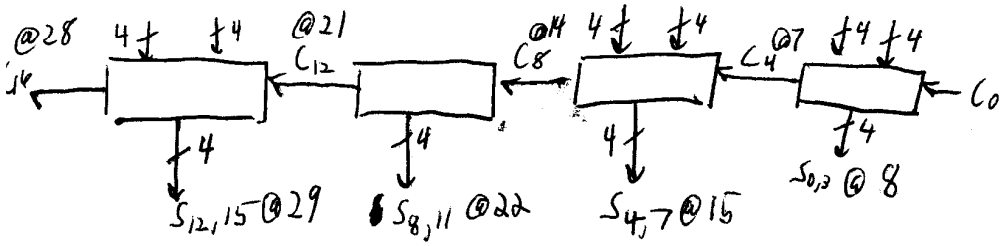
(e) what is the critical path delay of a 4-bit carry lookahead adder?

$t = 8$

(f) is S_4 on the CLA valid before the CRA?

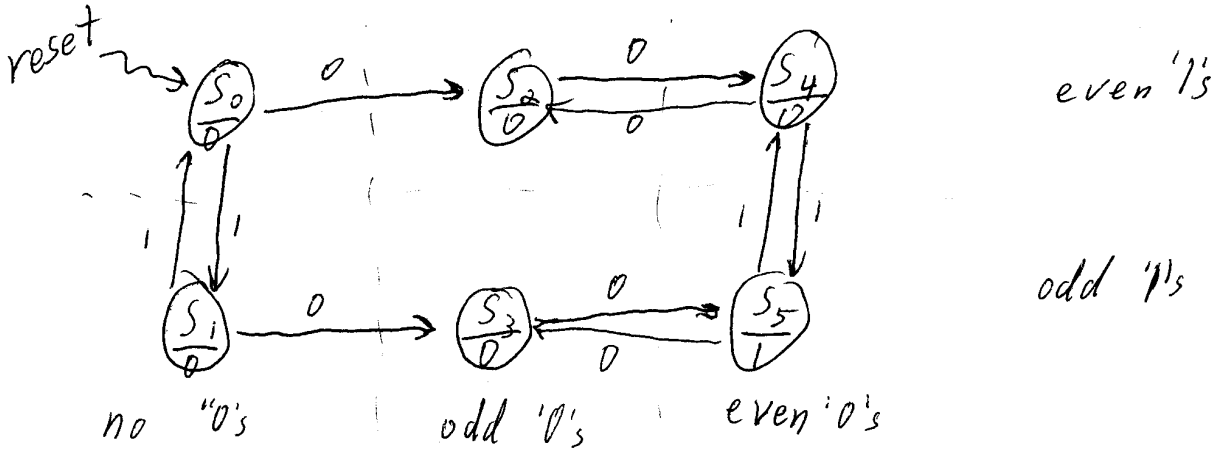
$t_{CLA} = 8$ $t_{CRA} = 9$ $t_{CLA} < t_{CRA}$ so yes

(g) construct a 16-bit adder with 4-bit carry lookahead adders. what is the critical path delay?



$t_c = 29$

A Moore sequential circuit has one input and one output. Output is "1" iff number of "1"'s received is odd and total number of "0"s is an even number greater than zero. Derive the state ~~table~~ diagram and Verilog.




```
module fsm(clk, reset, x, y);
  input clk, reset, x;
  output y;
  reg y;
  reg [2:0] state, next_state;
  parameter [2:0] S0=3'b000, S1=3'b001, S2=3'b010, S3=3'b011,
                  S4=3'b100, S5=3'b101;
```

```
always @(reset or x or state)
  if (reset) next_state<=S0;
  else begin
    case (state)
      S0: next_state <= x ? S1 : S2;
      S1: next_state <= x ? S0 : S3;
```

```
always @ (reset or x or state)
    if (reset) begin
        next_state <= 0;
        y <= 0;
    end
    else begin
      case(state)
        S0 : next_state <= x ? S1 : S2;
        S1 : next_state <= x ? S0 : S3;
        S2 : next_state <= x ? S3 : S4;
        S3 : next_state <= x ? S2 : S5;
        S4 : next_state <= x ? S5 : S2;
        S5 : next_state <= x ? S4 : S3;
        default : next_state <= 3'bxxx;
      endcase;
      y <= (state == S5) ? 1 : 0;
    end

always @ (posedge clk)
    state <= next_state
end module
```