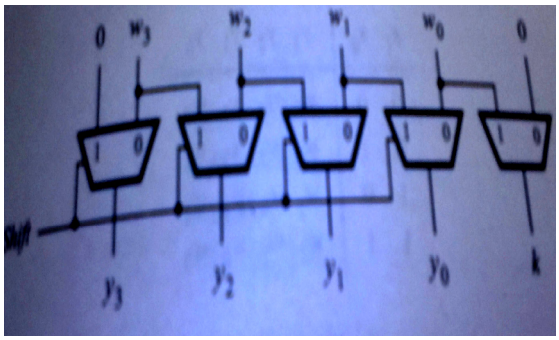
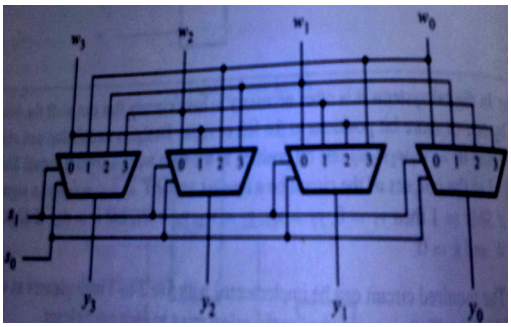
1. Design and implement a full adder using Verilog HDL.

a) gate level modelling
b) gate level modelling (using half adder)
c) behavioural modelling
d) data flow modelling

2. Write the Verilog code for a 2:4 decoder circuit using gate level, data flow and behavioural.
3. Write the Verilog code for a 3:8 decoder circuit using 2:4 decoder using gate level and data flow modelling.
4. Write the Verilog code for a BCD to 7-segment display code converter circuit using gate level, data flow and behavioural.
5. Write the Verilog code for a binary to Gray code converter circuit using gate level, data flow and behavioural.
6. Write the Verilog code for an 8:3 binary encoder circuit using gate level, data flow and behavioural.
7. Write the Verilog code for a shifter circuit and barrel shifter circuit shown in figure using gate level, data flow and behavioural.



Shifter Circuit



Barrel Shifter Circuit

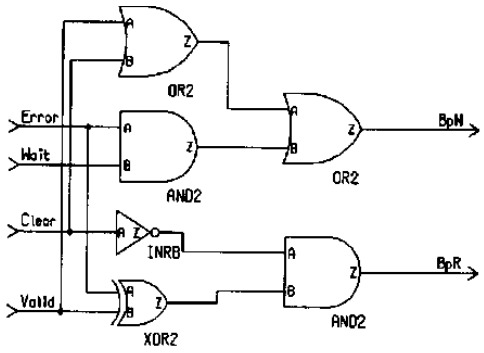
8. Write the Verilog code for a 4 bit comparator circuit using gate level, data flow and behavioural.
9. Write the Verilog code for an one hot encoder circuit using gate level, data flow and behavioural.
10. Show how a 16:1 multiplexer is built using five 4:1 multiplexers also write Verilog HDL using gate level, data flow and behavioural modelling
11. Write the Verilog code for a 4 bit BCD circuit using gate level, data flow and behavioural.
12. Write the Verilog code for a 2 bit Comparator circuit using gate level, data flow and behavioural.
13. Find out the expressions of the following functions and write Verilog HDL using gate level, data flow and behavioural modelling

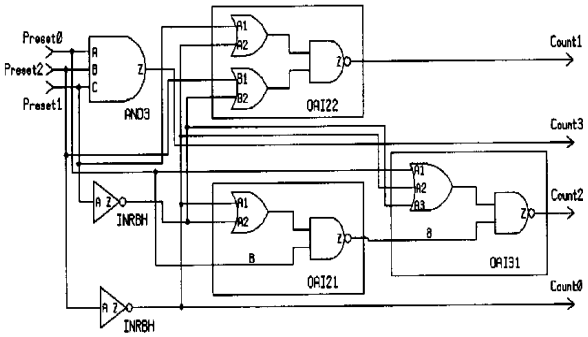
a) $f(x_1, x_2, x_3) = \sum m(2,3,4,6,7)$
b) $f(x_1, x_2, x_3) = \prod M(0,1,5)$
c) $f(x_1, x_2, x_3, x_4) = \sum m(3,7,9,12,13,14,15)$
d) $f(x_1, x_2, x_3) = \prod M(0,2,3,7)$

14. Implement the following functions by using Multiplexer and write the Verilog code using gate level, data flow

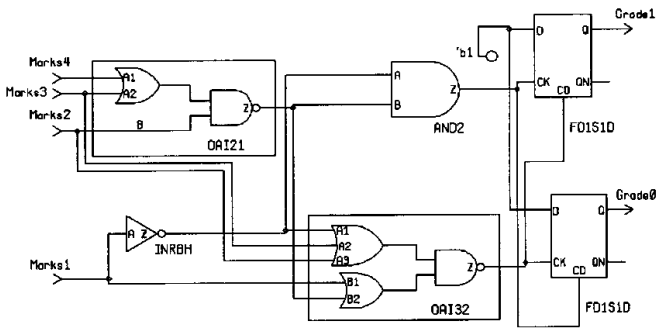
a) $f = \overline{w_1} w_3 + w_2 \overline{w_3}$
b) $f = \overline{w_1}\,\overline{w_3} + w_1 w_2 + w_1 w_3$

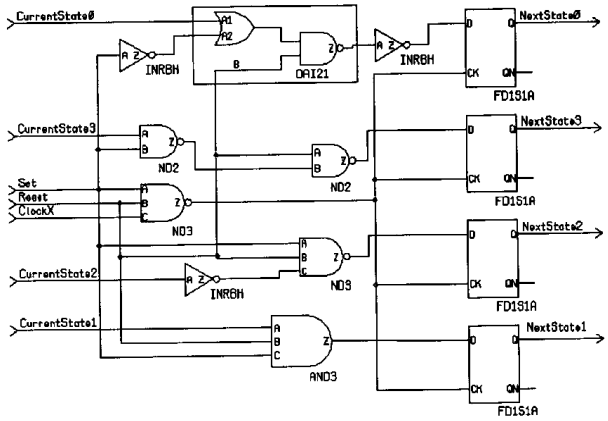
15. Design and implement 16:1 Multiplexer using two 8:1 MUX in Verilog HDL using gate level, data flow and behavioural.
16. Write the Verilog code for the circuit shown in figures, using gate level, data flow modelling

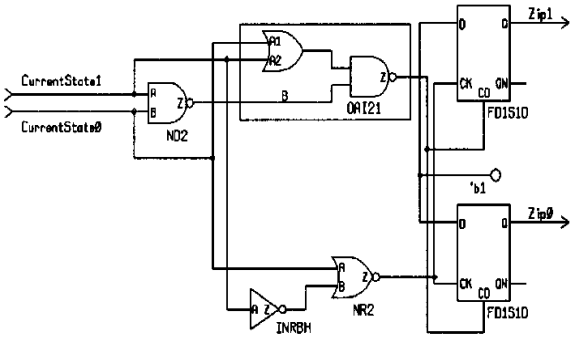


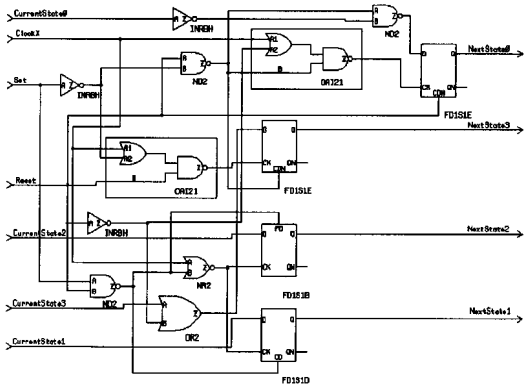


17. Model a 4 bit linear feedback shift register using Verilog HDL using data flow and behavioural modelling.
18. Write the Verilog code for the circuit shown in figures, using mixed design style modelling



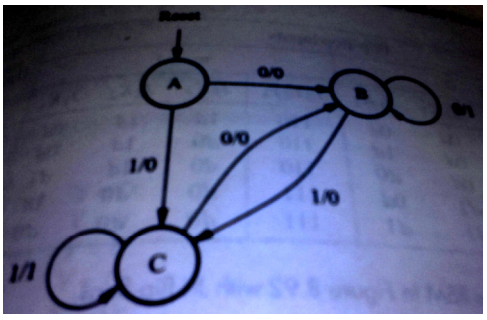






19. Model a 4 bit binary multiplier using Verilog HDL using data flow and behavioural modelling.
20. Write a Verilog code to verify the functionality of T flip-flop with test bench and synthesize the above design with the following constraints (Clock, Input delay, Output delay, Rise time, Fall time) to meet the timing.
21. Write a Verilog code to verify the functionality of 4-bit Asynchronous counter with test bench and synthesize the above design with the following constraints (Clock, Input delay, Output delay, Rise time, Fall time) to meet the timing.

22. Write a Verilog code to verify the functionality of 4-bit Synchronous counter with test bench and synthesize the above design with the following constraints (Clock, Input delay, Output delay, Rise time, Fall time) to meet the timing.
23. Write a Verilog code to verify the functionality of JKMS flip-flop with test bench and synthesize the above design with the following constraints (Clock, Input delay, Output delay, Rise time, Fall time) to meet the timing.
24. Write a Verilog code to verify the functionality of JK flip-flop with test bench and synthesize the above design with the following constraints (Clock, Input delay, Output delay, Rise time, Fall time) to meet the timing.
25. Write a Verilog code to verify the functionality of D flip-flop with test bench and synthesize the above design with the following constraints (Clock, Input delay, Output delay, Rise time, Fall time) to meet the timing.
26. Write a Verilog code to verify the functionality of RS flip-flop with test bench and synthesize the above design with the following constraints (Clock, Input delay, Output delay, Rise time, Fall time) to meet the timing.
27. Write a Verilog code to verify the functionality of Serial adder with test bench and synthesize the above design with the following constraints (Clock, Input delay, Output delay, Rise time, Fall time) to meet the timing.
28. Write a Verilog code to verify the functionality of Parallel adder with test bench and synthesize the above design with the following constraints (Clock, Input delay, Output delay, Rise time, Fall time) to meet the timing.
29. Write a Verilog code to verify the functionality of Universal gates and Transmission gates with test bench and synthesize the above design.
30. Write a Verilog code to verify the functionality of Inverter and Buffer with test bench and synthesize the above design.
31. Design a FSM (Finite State Machine either Moorey type or Mealy Type) to detect a sequence 10110.
32. Design a FSM (Finite State Machine either Moorey type or Mealy Type) to detect more than one "1"s in last 3 samples. Overlapping input patterns are allowed.
33. Design an FSM that has input w and output z. The machine is a sequence detector that produces z=1 when the previous two values of w were 00 or 11: otherwise z=0. Also write the Verilog code to implement the FSM.
34. Write Verilog code to implement the FSM shown in the figure.



State Diagram

35. Write Verilog code for the FSM shown in the table.

Present State $y_2\ y_1$	Next State w=0 $Y_2\ Y_1$	Next State w=1 $Y_2\ Y_1$	Output z
00	10	11	0
01	01	00	0
10	11	00	0
11	10	01	1

State Assigned Table

36. A sequential circuit has two inputs, w_1 and w_2 and one output z. Its function is to compare the input sequences on the two inputs. If $w_1=w_2$ during any four consecutive clock cycles, the circuit produces z=1: otherwise z=0. For example

W_1=0110111000110

W_2=1110101000111

Z= 0000100001110

Write Verilog code for the FSM discussed above.