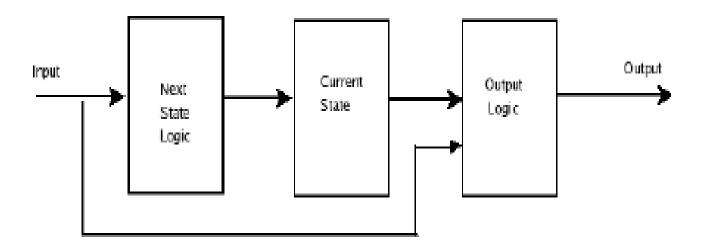
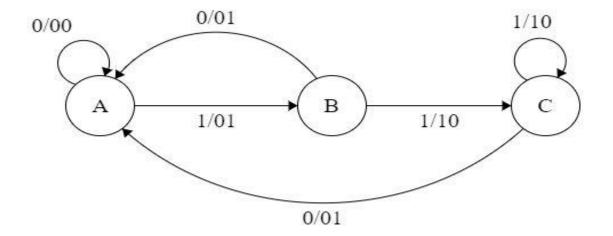
محمد مهدی نظری ۹۹۳۱۰۶۱ – آرمین ابراهیمی صبا ۹۹۳۱۰۸۶

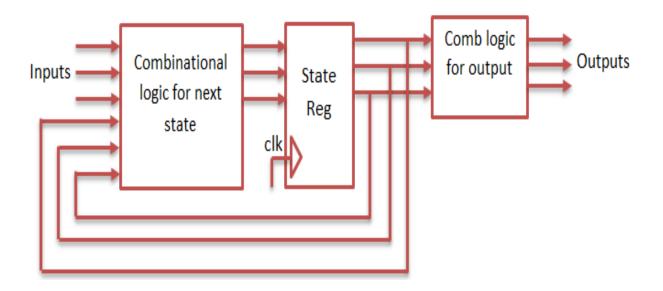
۱ – ماشین های میلی و مور

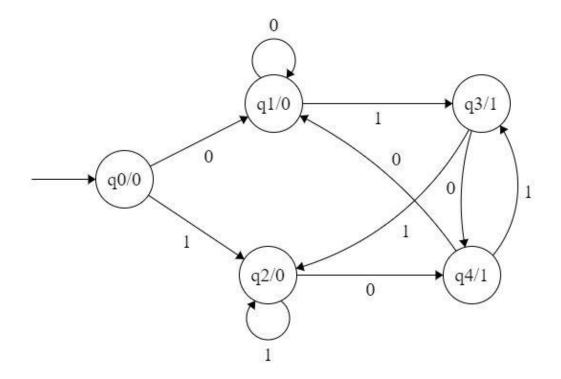
در ماشین های میلی (Mealy) خروجی های مدار به هردوی ورودی ها و حالت فعلی مدار بستگی دارند و ترکیبی از آن ها خروجی را میسازد . دیاگرام حالت و شکل کلی مدار آن ها مانند زیر است :



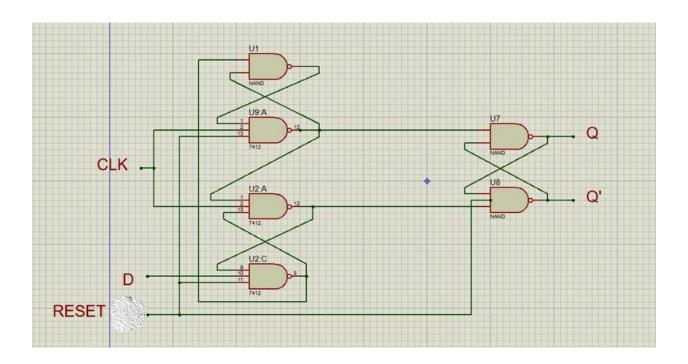


در ماشین های مور (Moore) خروجی هارا فقط حالات فعلی مدار یا رجیستر ها تعیین میکنند و ورودی ها تاثیر مستقیم در تولید خروجی ندارند . دیاگرام حالت و شکل کلی مدار آن ها مانند زیر است :



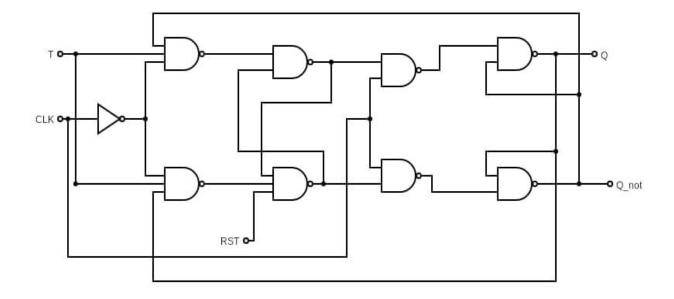


DFF - Y



D Flip Flop					
Input	Output				
D	Q	Q^			
0	0	1			
1	1	0			

TFF - $^{\circ}$



Q	Т	Q(t+1)	
О	0	0	
0	1	1	
1	О	1	
1	1	0	

"1101" sequence detector - $^{\circ}$

میلی:

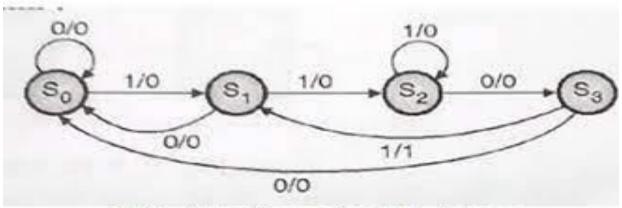
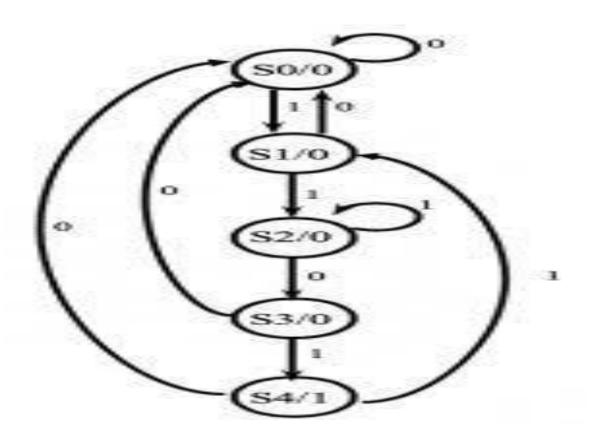
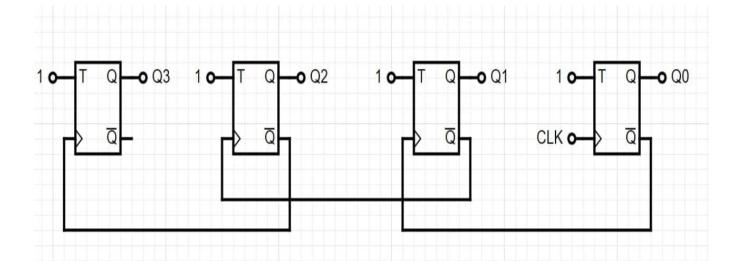


Fig8. State diagram for 1101 detector

مور :



Ripple Counter using TFF $-\delta$



Q_3	Q_2	Q_1	Q_0	Decimal Value
0	0	0	0	O
O	0	0	1	1
0	0	1	0	2
0	0	1	ı	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	61 4	15
0	0	0	0	full reset

Process Block & Sequential Circuits $-\hat{\gamma}$

```
Process statement is a concurrent ofatement.
  but the statements within the process
                                                                    use IEEESTD_LOGIC_1164.ALL;
                                                                    entity moore machine is
                                                                              input : in std_logic ;
                                                                              output : out std_logic_vector(1 downto 0);
                                                                              clk: in std_logic
                                                                    architecture Behavioral of moore machine is
                                                                         type state_t is (s0, s1, s2);
                                                                         signal state : state t := s0;
                                                                         signal next_state : state_t := s0;
      1. sensitivity list
                                                                        begin
                                                                              case state is
      2. sequential circuit
                                                                                    when s0=>
                                                                                         if(input = '1') then
                                                                                               next_state <= s1;
                                                                                                                                moore machine
     3 - combinationer circuit
                                                                                               next_state <= state :
                                                                                    when s1=>
                                                                                                                                                                    M
sequential circuits propertys:
                                                                                    when s2=>
                                                                                                                                                                       comb .
                                                                                                                               comp.
                                                                                    when others=>
                                                                                         next_state <= s0;
      1. need a memory
                                                                                           -sensitivit
                                                                         end process; 🗸
                                                                         REG: process(clk)
                                                                                                                                             current state
      2.400 both current state and input for output
                                                                                   if(clk'event and clk='1') then sequent in cineuit
                                                                                         state <= next_state;
                                                                         end process;
      3. has a clock
                                                                         output <= "00" when state = s0 else
                                                                                "10" when state = s1 else
                                                                                "11" when state = s2;
    4. has a feedback
```