HDL FSM Examples

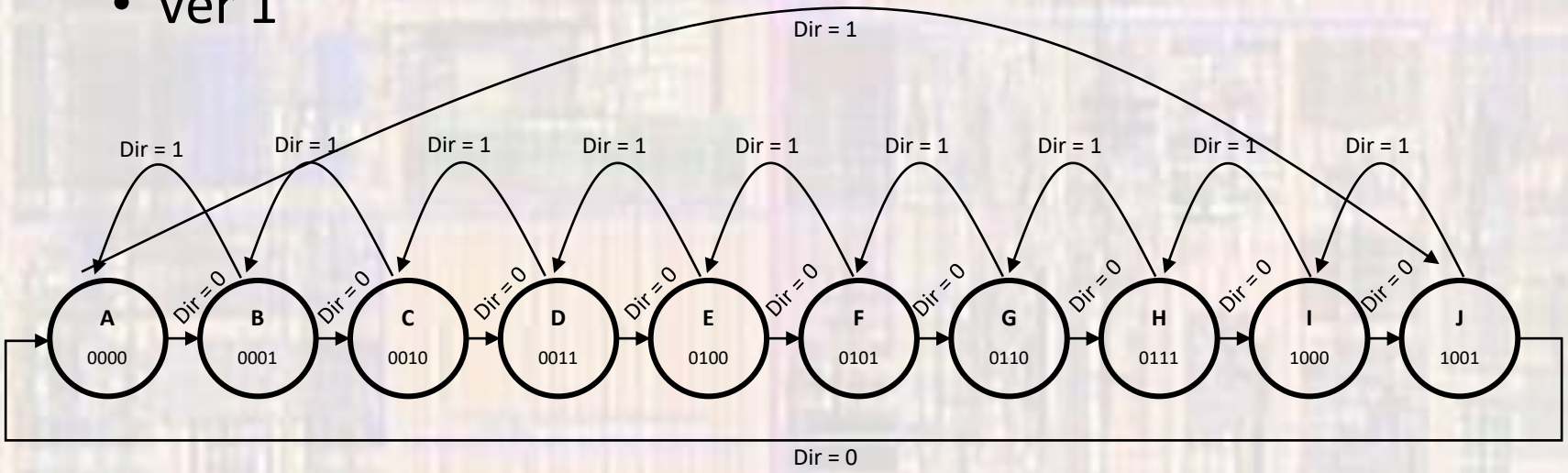
Last updated 1/22/21

State Machine Intro

- These slides present VHDL FSM examples using enumerated states

HDL FSM Examples

- Mod 10 u/d Counter – Enumerated States
  - Ver 1



Note: this is for illustrative purpose - we would never make a counting state machine like this

HDL FSM Examples

- Mod 10 u/d Counter – Enumerated States
  - Ver 1

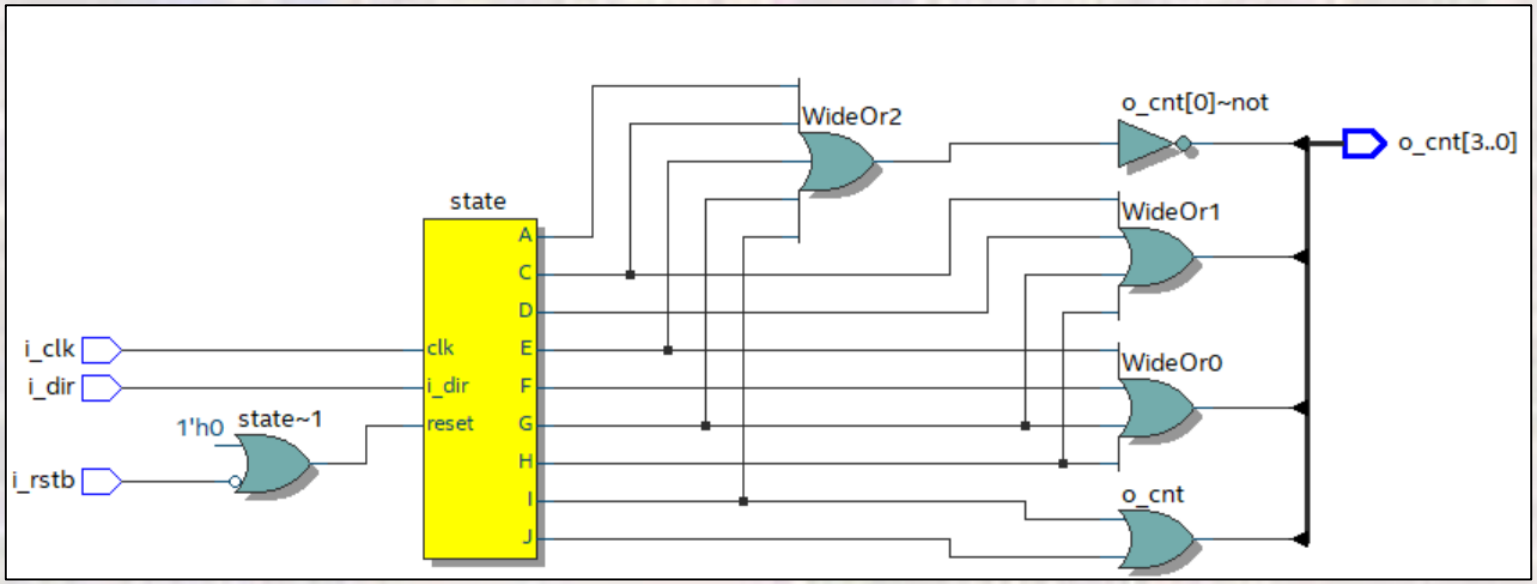
```
----------------------------------------
--
-- counter_mod10_ud_fsm.vhdl
--
-- created 3/30/18
-- tj
--
-- rev 0
----------------------------------------
--
-- mod 10 up/dn-counter - state machine with case statement
--                        enumerated states
----------------------------------------
--
-- Inputs: i_rstb, i_clk, i_dir (0=up)
-- Outputs: o_cnt[3:0]
--
----------------------------------------
library ieee;
use ieee.std_logic_1164.all;

entity counter_mod10_ud_fsm is
   port (
         i_clk :  in std_logic;
         i_rstb : in std_logic;
         i_dir :  in std_logic;

         o_cnt :  out std_logic_vector(3 downto 0)
    );
end entity;
```

```
architecture behavioral of counter_mod10_ud_fsm is

   --
   -- internal signals
   --
   type STATE_TYPE is (A, B, C, D, E, F, G, H, I, J);
   signal state:       STATE_TYPE;
   signal state_next:  STATE_TYPE;

begin

   --
   -- next state logic
   --
   process(all)
   begin
     if(i_dir = '0') then
        case state is
           when A => state_next <= B;
           when B => state_next <= C;
           when C => state_next <= D;
           when D => state_next <= E;
           when E => state_next <= F;
           when F => state_next <= G;
           when G => state_next <= H;
           when H => state_next <= I;
           when I => state_next <= J;
           when J => state_next <= A;
           when others => state_next <= A;
        end case;
     else
        case state is
           when A => state_next <= J;
           when B => state_next <= A;
           when C => state_next <= B;
           when D => state_next <= C;
           when E => state_next <= D;
           when F => state_next <= E;
           when G => state_next <= F;
           when H => state_next <= G;
           when I => state_next <= H;
           when J => state_next <= I;
           when others => state_next <= A;
        end case;
     end if;
   end process;
```

```
   --
   -- Register logic
   --
   process(i_clk, i_rstb)
   begin
      -- reset
      if (i_rstb = '0') then
         state <= A;
      -- rising i_clk edge
      elsif (rising_edge (i_clk)) then
         state <= state_next;
      end if;
   end process;

   --
   -- Output logic
   --
   process(all)
   begin
      case state is
         when A => o_cnt <= "0000";
         when B => o_cnt <= "0001";
         when C => o_cnt <= "0010";
         when D => o_cnt <= "0011";
         when E => o_cnt <= "0100";
         when F => o_cnt <= "0101";
         when G => o_cnt <= "0110";
         when H => o_cnt <= "0111";
         when I => o_cnt <= "1000";
         when J => o_cnt <= "1001";
         when others => o_cnt <= "0000";
      end case;
   end process;

end behavioral;
```

HDL FSM Examples

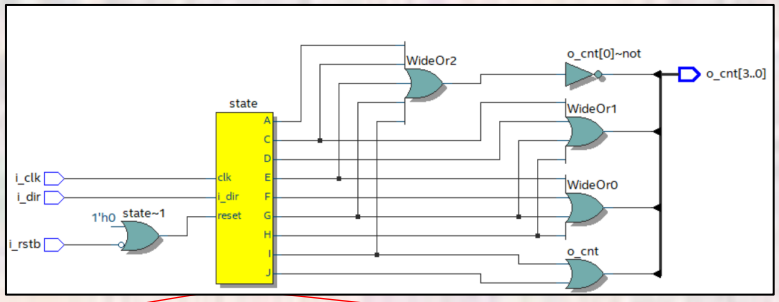
- Mod 10 u/d Counter – Enumerated States
  - Ver 1

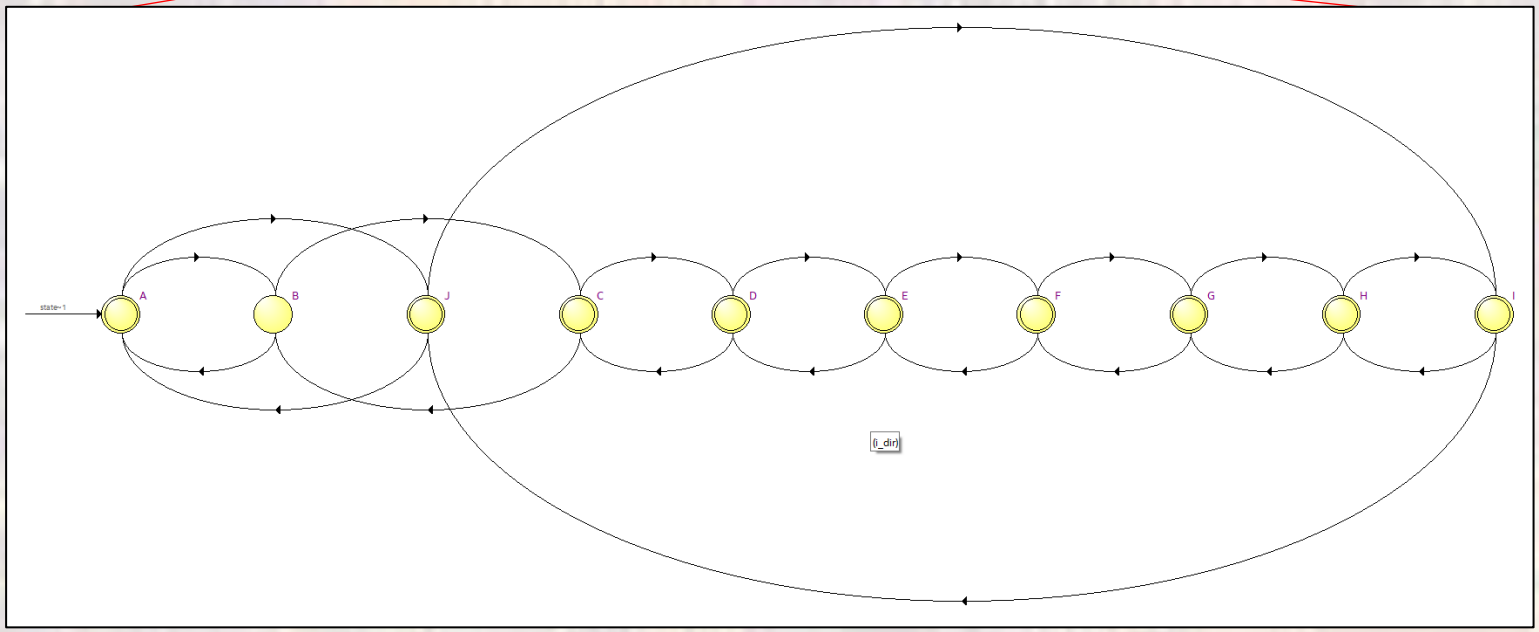


Quartus recognized part of the code as a state machine

HDL FSM Examples

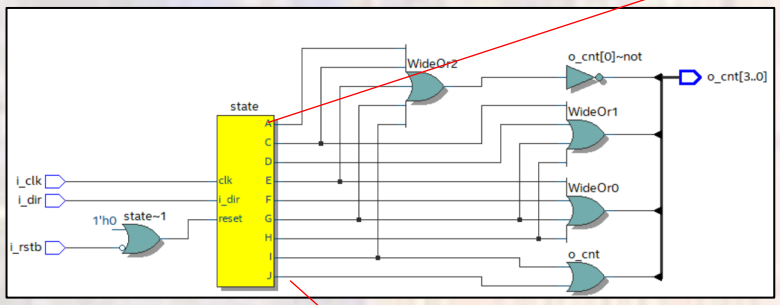
- Mod 10 u/d Counter – Enumerated States
  - Ver 1





HDL FSM Examples

- Mod 10 u/d Counter – Enumerated States
  - Ver 1



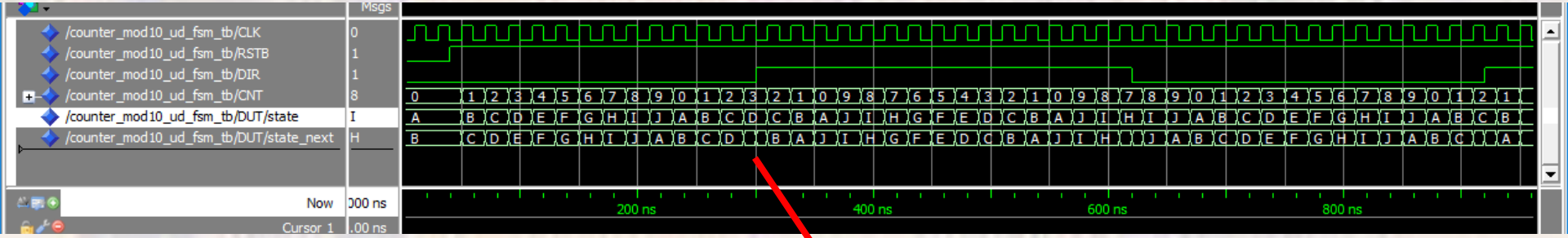
	Source State	Destination State	Condition
1	A	J	(dir)
2	A	B	(!dir)
3	B	C	(!dir)
4	B	A	(dir)
5	C	D	(!dir)
6	C	B	(dir)
7	D	E	(!dir)
8	D	C	(dir)
9	E	F	(!dir)
10	E	D	(dir)
11	F	G	(!dir)
12	F	E	(dir)
13	G	H	(!dir)
14	G	F	(dir)
15	H	I	(!dir)
16	H	G	(dir)
17	I	J	(!dir)
18	I	H	(dir)
19	J	I	(dir)
20	J	A	(!dir)

Transitions | Encoding

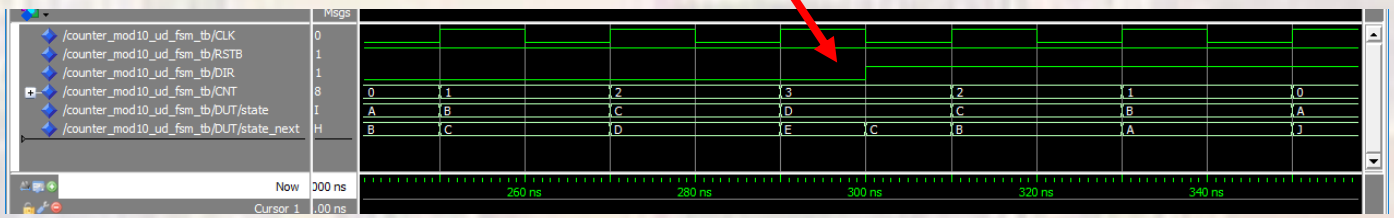
HDL FSM Examples

- Mod 10 u/d Counter – Enumerated States
  - Ver 1

After adding states and restarting

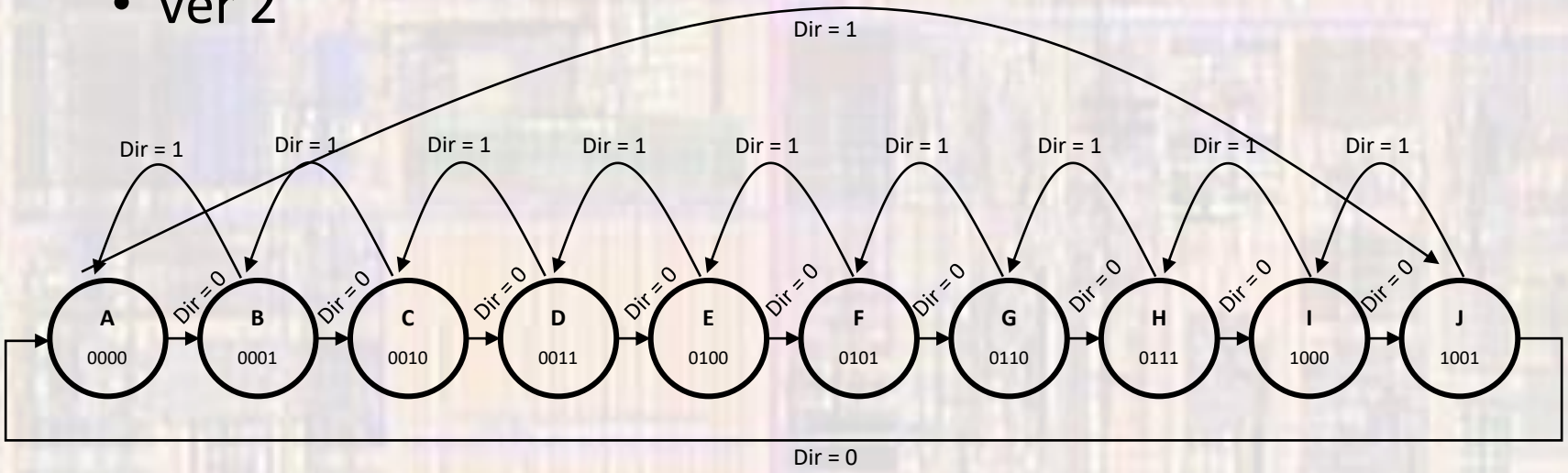


note – due to use of “all” in process sensitivity list



HDL FSM Examples

- Mod 10 u/d Counter – Enumerated States
  - Ver 2



Note: this is for illustrative purpose - we would never make a counting state machine like this

HDL FSM Examples

- Mod 10 u/d Counter – d States
  - Ver 2

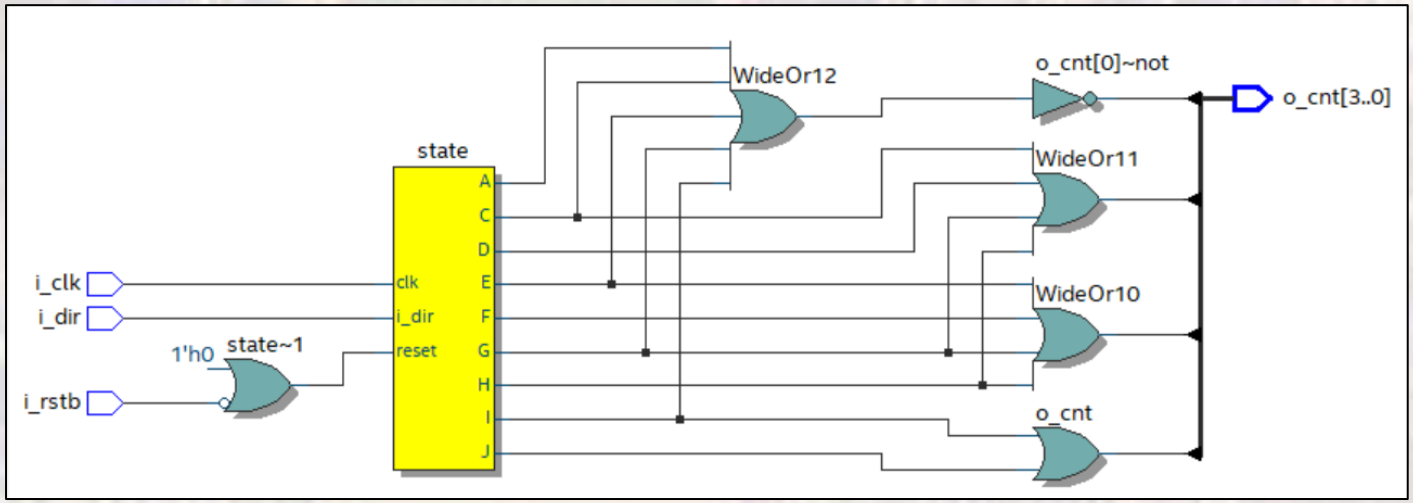
```
----------------------------------------
--
-- counter_mod10_ud_fsmA.vhdl
--
-- created 3/30/18
-- tj
--
-- rev 0
----------------------------------------
--
-- mod 10 up/dn-counter - state machine with case statement
--                        enumerated states
----------------------------------------
--
-- Inputs: i_rstb, i_clk, i_dir(0=up)
-- Outputs: o_cnt[3:0]
--
----------------------------------------
library ieee;
use ieee.std_logic_1164.all;

entity counter_mod10_ud_fsmA is
   port (
         i_clk :  in std_logic;
         i_rstb : in std_logic;
         i_dir :  in std_logic;

         o_cnt :  out std_logic_vector(3 downto 0)
    );
end entity;

architecture behavioral of counter_mod10_ud_fsmA is

   --
   -- internal signals
   --
   type STATE_TYPE is (A, B, C, D, E, F, G, H, I, J);
   signal state:       STATE_TYPE;
   signal state_next:  STATE_TYPE;

begin
```

```
--
-- next state logic
--
process(all)
begin
   case state is
      when A =>
         if(i_dir = '0') then
            state_next <= B;
         else
            state_next <= J;
         end if;
      when B =>
         if(i_dir = '0') then
            state_next <= C;
         else
            state_next <= A;
         end if;
      when C =>
         if(i_dir = '0') then
            state_next <= D;
         else
            state_next <= B;
         end if;
      when D =>
         if(i_dir = '0') then
            state_next <= E;
         else
            state_next <= C;
         end if;
      when E =>
         if(i_dir = '0') then
            state_next <= F;
         else
            state_next <= D;
         end if;
      when F =>
         if(i_dir = '0') then
            state_next <= G;
         else
            state_next <= E;
         end if;
      when G =>
         if(i_dir = '0') then
            state_next <= H;
         else
            state_next <= F;
         end if;
      when H =>
         if(i_dir = '0') then
            state_next <= I;
         else
            state_next <= G;
         end if;
      when I =>
         if(i_dir = '0') then
            state_next <= J;
         else
            state_next <= H;
         end if;
      when J =>
         if(i_dir = '0') then
            state_next <= A;
         else
            state_next <= I;
         end if;
      when others =>
         state_next <= A;
   end case;
end process;
```

```
   --
   -- Register logic
   --
   process(i_clk, i_rstb)
   begin
      -- reset
      if (i_rstb = '0') then
         state <= A;
      -- rising i_clk edge
      elsif (rising_edge (i_clk)) then
         state <= state_next;
      end if;
   end process;

   --
   -- Output logic
   --
   process(all)
   begin
      case state is
         when A => o_cnt <= "0000";
         when B => o_cnt <= "0001";
         when C => o_cnt <= "0010";
         when D => o_cnt <= "0011";
         when E => o_cnt <= "0100";
         when F => o_cnt <= "0101";
         when G => o_cnt <= "0110";
         when H => o_cnt <= "0111";
         when I => o_cnt <= "1000";
         when J => o_cnt <= "1001";
         when others => o_cnt <= "0000";
      end case;
   end process;

end behavioral;
```

HDL FSM Examples

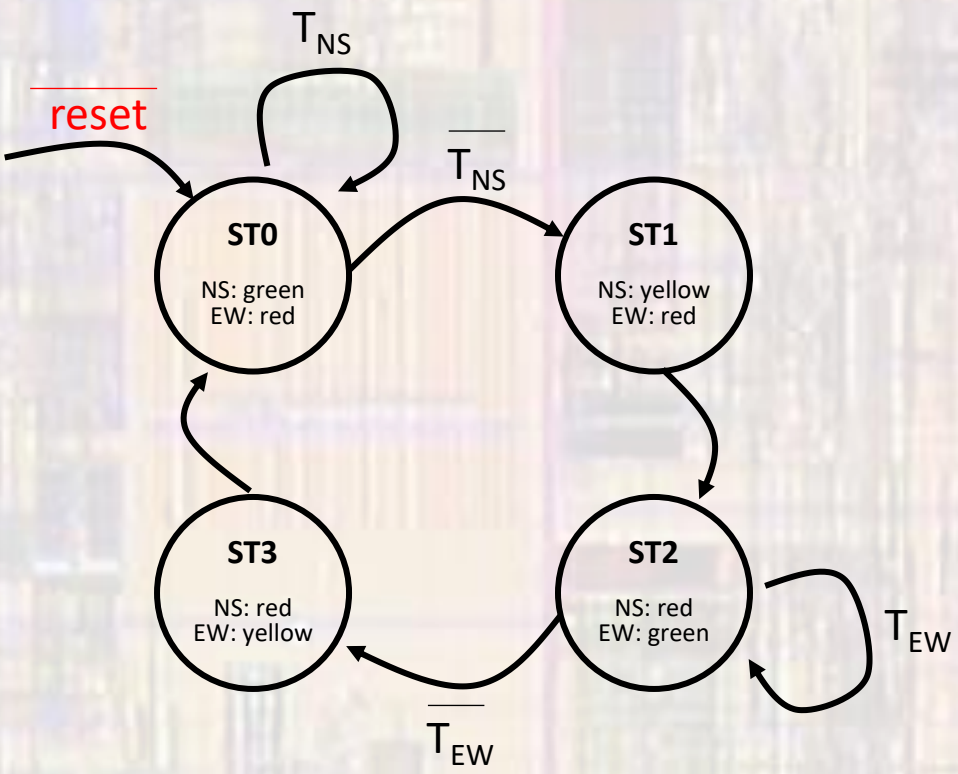
- Mod 10 u/d Counter – Enumerated States
  - Ver 2



Quartus recognized part of the code as a state machine

HDL FSM Examples

- Priority Stoplight - revisited



HDL FSM Examples

- Priority Stoplight - revisited

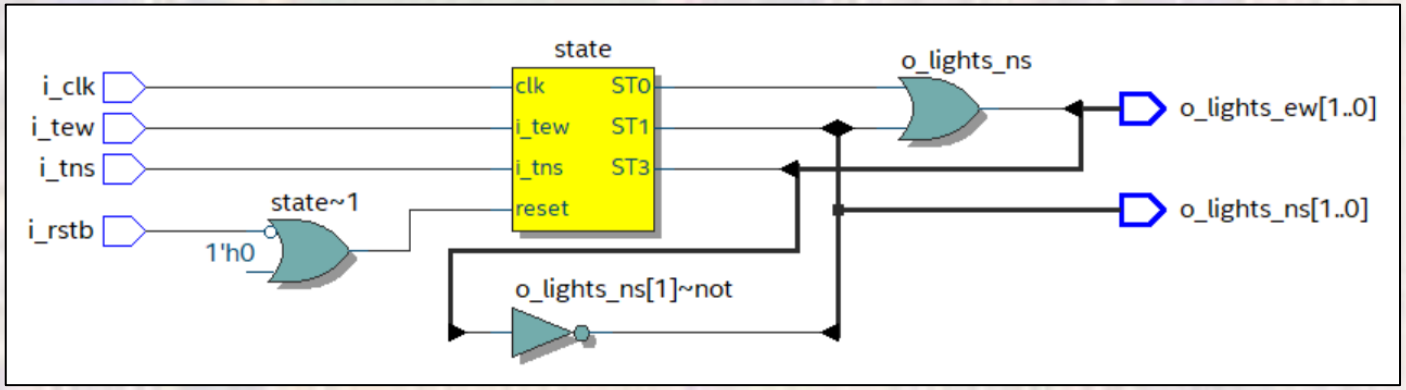
```
----------------------------------------
--
-- stoplight_nsew_fsm.vhdl
--
-- created 3/30/18
-- tj
--
-- rev 0
----------------------------------------
--
-- NS/EW Stoplight
----------------------------------------
--
-- Inputs: i_rstb, i_clk, i_tns, i_tew
-- Outputs: o_lights_ew, o_lights_ns
--
----------------------------------------
library ieee;
use ieee.std_logic_1164.all;

entity stoplight_nsew_fsm is
   port (
         i_clk :      in std_logic;
         i_rstb :     in std_logic;
         i_tns :      in std_logic;
         i_tew :      in std_logic;

         o_lights_ns : out std_logic_vector(1 downto 0);
         o_lights_ew : out std_logic_vector(1 downto 0)

    );
end entity;
```

```
architecture behavioral of stoplight_nsew_fsm is
   --
   -- internal signals
   --
   type STATE_TYPE is (ST0, ST1, ST2, ST3);
   signal state:       STATE_TYPE;
   signal state_next:  STATE_TYPE;

   constant green:   std_logic_vector  := "00";
   constant yellow:  std_logic_vector  := "01";
   constant red:     std_logic_vector  := "10";

begin
   --
   -- next state logic
   --
   process(all)
   begin
      case state is
         when ST0 =>
            if(i_tns = '1') then
               state_next <= ST0;
            else
               state_next <= ST1;
            end if;

         when ST1 =>
            state_next <= ST2;

         when ST2 =>
            if(i_tew = '1') then
               state_next <= ST2;
            else
               state_next <= ST3;
            end if;

         when ST3 =>
            state_next <= ST0;
      end case;
   end process;
```

```
   --
   -- Register logic
   --
   process(i_clk, i_rstb)
   begin
      -- reset
      if (i_rstb = '0') then
         state <= ST0;
      -- rising i_clk edge
      elsif (rising_edge(i_clk)) then
         state <= state_next;
      end if;
   end process;

   --
   -- Output logic
   --
   process(all)
   begin
      case state is
         when ST0 =>
            o_lights_ns <= green;
            o_lights_ew <= red;
         when ST1 =>
            o_lights_ns <= yellow;
            o_lights_ew <= red;
         when ST2 =>
            o_lights_ns <= red;
            o_lights_ew <= green;
         when ST3 =>
            o_lights_ns <= red;
            o_lights_ew <= yellow;
      end case;
   end process;

end behavioral;
```

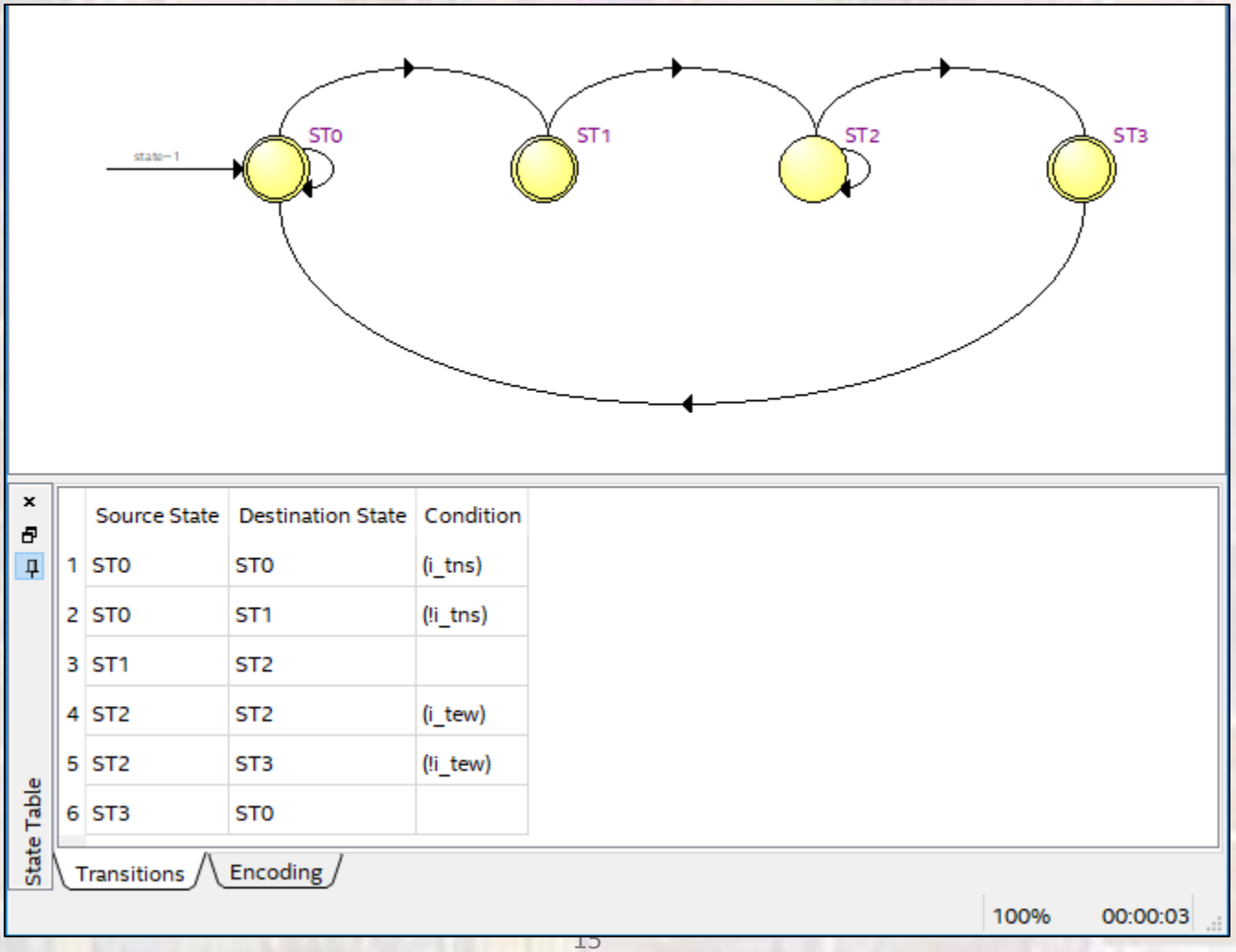
HDL FSM Examples

- Priority Stoplight - revisited



HDL FSM Examples

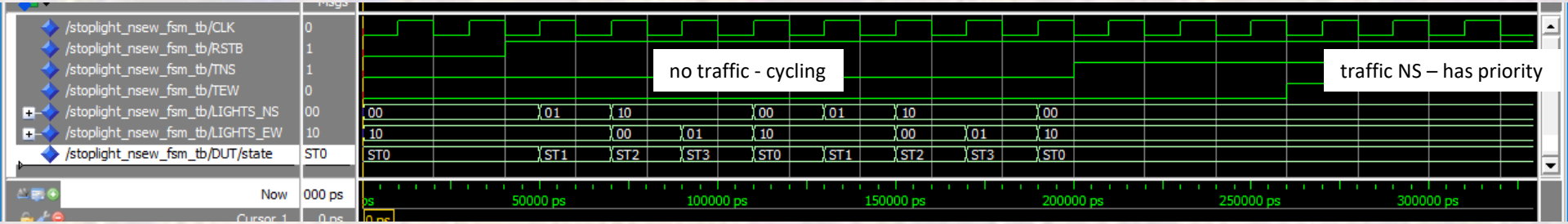
- Priority Stoplight - revisited

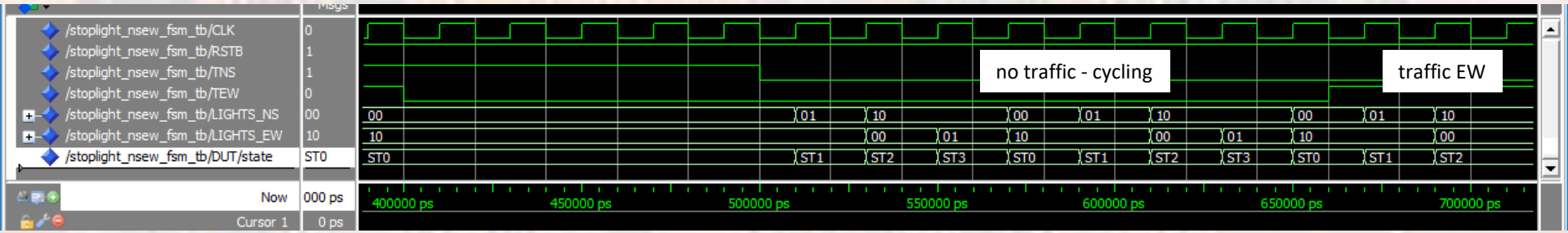


	Source State	Destination State	Condition
1	ST0	ST0	(i_tns)
2	ST0	ST1	(!i_tns)
3	ST1	ST2	
4	ST2	ST2	(i_tew)
5	ST2	ST3	(!i_tew)
6	ST3	ST0	

HDL FSM Examples

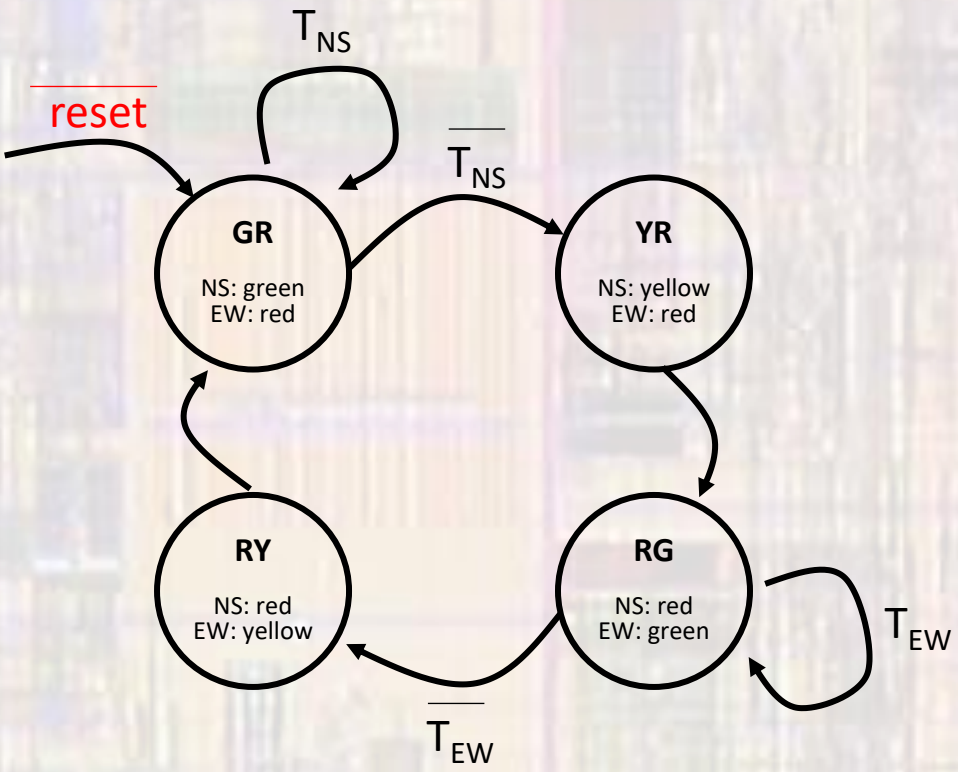
- Priority Stoplight - revisited





HDL FSM Examples

- Priority Stoplight – revisited - revisited



HDL FSM Examples

- Priority Stoplight – revisited - revisited

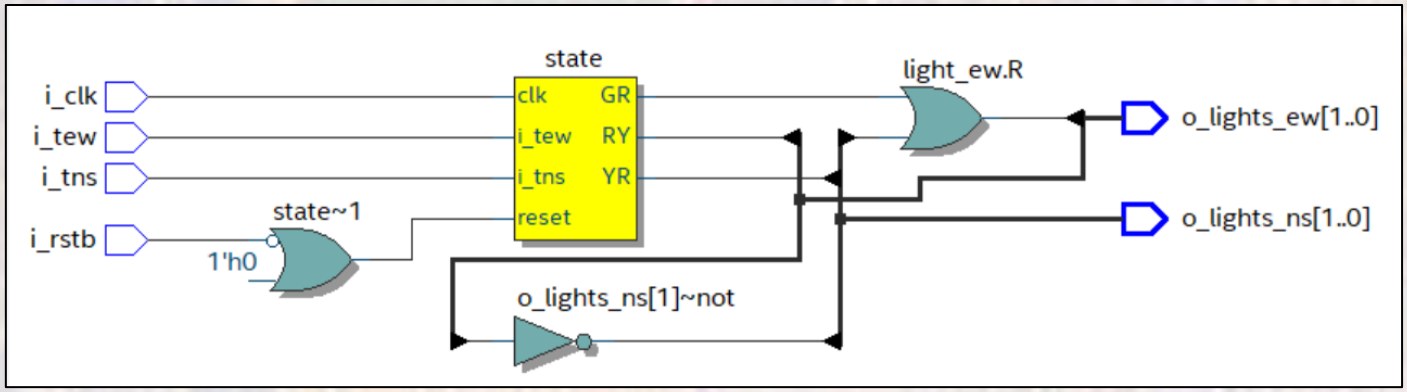
```
----------------------------------------
--
-- stoplight_nsew_fsm2.vhdl
--
-- created 3/30/18
-- tj
--
-- rev 0
----------------------------------------
--
-- NS/EW Stoplight with o_lights enumerated
----------------------------------------
--
-- Inputs: i_rstb, i_clk, i_tns, i_tew
-- Outputs: o_lights_ew, o_lights_ns
--
----------------------------------------
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity stoplight_nsew_fsm2 is
   port (
         i_clk :     in std_logic;
         i_rstb : in std_logic;
         i_tns :     in std_logic;
         i_tew :     in std_logic;

         o_lights_ns : out std_logic_vector(1 downto 0);
         o_lights_ew : out std_logic_vector(1 downto 0)

    );
end entity;
```

```
architecture behavioral of stoplight_nsew_fsm2 is
   --
   -- internal signals
   --
   type STATE_TYPE is (GR, YR, RG, RY);
   signal state:      STATE_TYPE;
   signal state_next:   STATE_TYPE;

   type LIGHT is (G, Y, R);
   signal light_ns:  LIGHT;
   signal light_ew:  LIGHT;

begin
   --
   -- next state logic
   --
   process(all)
   begin
      case state is
         when GR =>
            if(i_tns = '1') then
               state_next <= GR;
            else
               state_next <= YR;
            end if;
         when YR =>
            state_next <= RG;
         when RG =>
            if(i_tew = '1') then
               state_next <= RG;
            else
               state_next <= RY;
            end if;
         when RY =>
            state_next <= GR;
      end case;
   end process;
```

```
   --
   -- Register logic
   --
   process(i_clk, i_rstb)
   begin
      -- reset
      if (i_rstb = '0') then
         state <= GR;
      -- rising clk edge
      elsif (rising_edge(i_clk)) then
         state <= state_next;
      end if;
   end process;

   --
   -- Output logic
   --
   process(all)
   begin
      case state is
         when GR =>
            light_ns <= G;
            light_ew <= R;
         when YR =>
            light_ns <= Y;
            light_ew <= R;
         when RG =>
            light_ns <= R;
            light_ew <= G;
         when RY =>
            light_ns <= R;
            light_ew <= Y;
      end case;
   end process;

   o_lights_ns <= std_logic_vector(to_unsigned(LIGHT'pos(light_ns), 2));
   o_lights_ew <= std_logic_vector(to_unsigned(LIGHT'pos(light_ew), 2));

end behavioral;
```

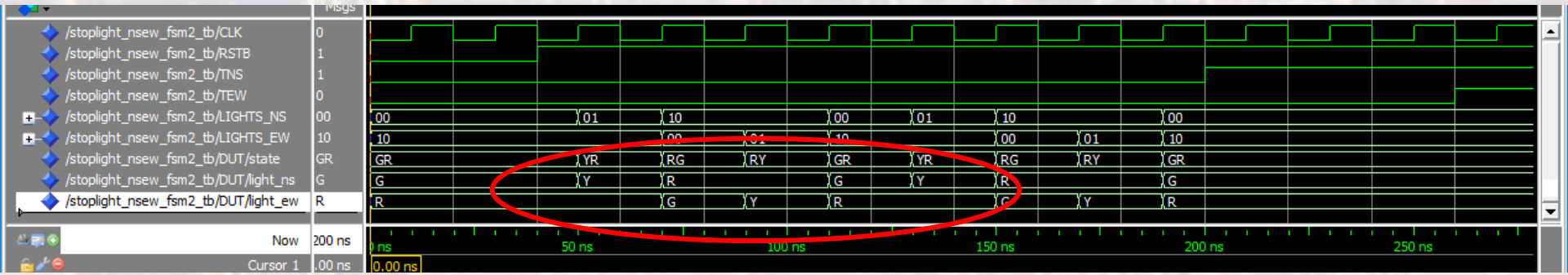
HDL FSM Examples

- Priority Stoplight – revisited - revisited



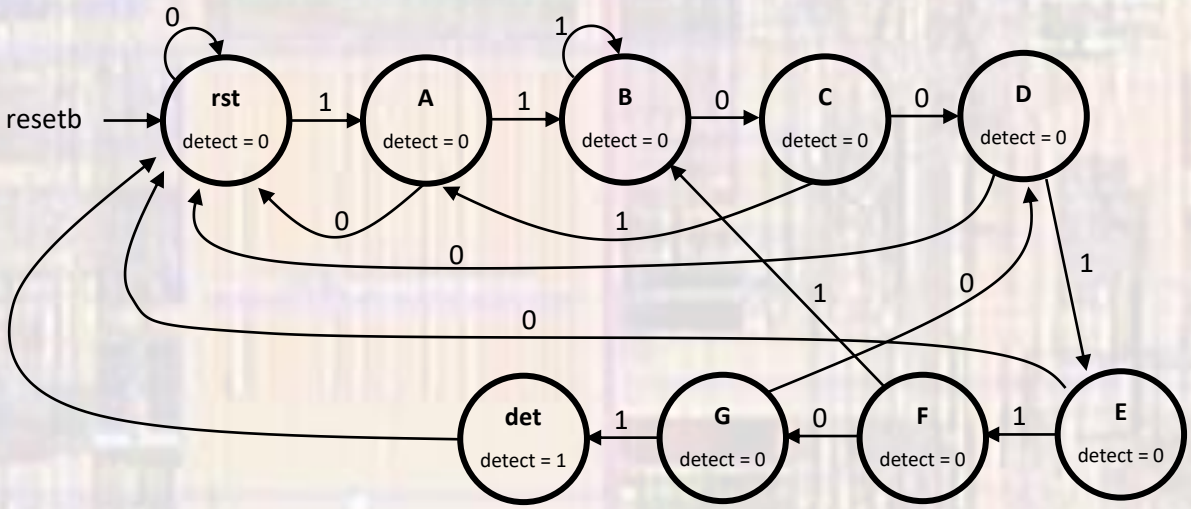
HDL FSM Examples

- Priority Stoplight – revisited - revisited



HDL FSM Examples

- Sequence detector – revisited
  - Detecting 11001101



HDL FSM Examples

- Sequence detector -

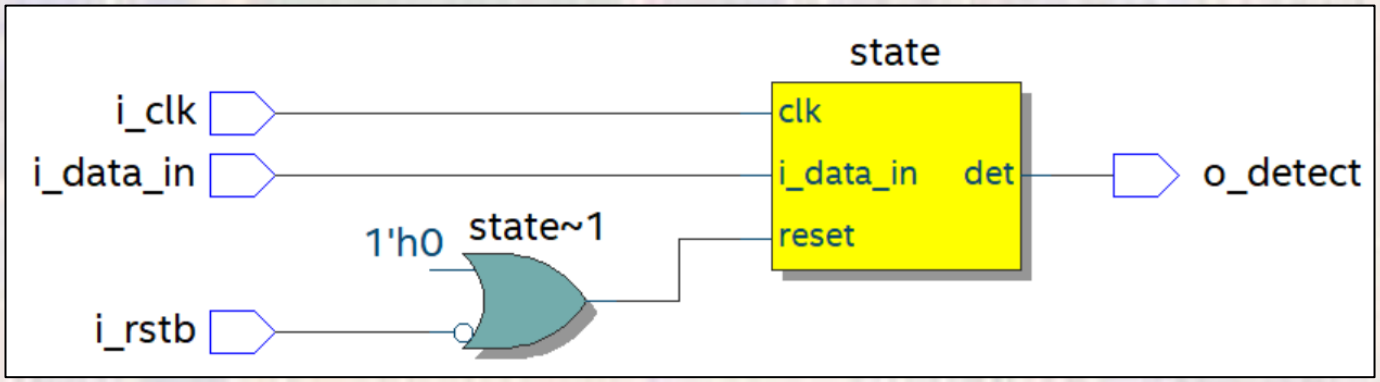
```
----------------------------------------
--
-- sequence_detector_CD_fsm.vhdl
--
-- created 4/12/2018
-- tj
--
-- rev 0
----------------------------------------
--
-- sequence detector state machine
--
----------------------------------------
--
-- Inputs: i_rstb, i_clk, i_data_in
-- Outputs: Detect
--
----------------------------------------
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity sequence_detector_CD_fsm is
   port (
      i_clk:      in std_logic;
      i_rstb:     in std_logic;
      i_data_in:  in std_logic;

      o_detect:   out std_logic
   );
end entity;

architecture behavioral of sequence_detector_CD_fsm is

   --
   -- internal signals
   --
  type STATETYPE is (rst,A,B,C,D,E,F,G,det);
  signal state:       STATETYPE;
  signal state_next:  STATETYPE;

begin
```

```
   --
   -- next state logic
   --
   process(all)
   begin
      case state is
         when rst =>
            if i_data_in = '1' then
               state_next <= A;
            else
               state_next <= rst;
            end if;
         when A =>
            if i_data_in = '1' then
               state_next <= B;
            else
               state_next <= rst;
            end if;
         when B =>
            if i_data_in = '0' then
               state_next <= C;
            else
               state_next <= B;
            end if;
         when C =>
            if i_data_in = '0' then
               state_next <= D;
            else
               state_next <= A;
            end if;
         when D =>
            if i_data_in = '1' then
               state_next <= E;
            else
               state_next <= rst;
            end if;
         when E =>
            if i_data_in = '1' then
               state_next <= F;
            else
               state_next <= rst;
            end if;
         when F =>
            if i_data_in = '0' then
               state_next <= G;
            else
               state_next <= B;
            end if;
         when G =>
            if i_data_in = '1' then
               state_next <= det;
            else
               state_next <= D;
            end if;
         when det =>
            state_next <= rst;
      end case;
   end process;
```

```
   --
   -- Register logic
   --
   process(i_clk, i_rstb)
   begin
      -- reset
      if (i_rstb = '0') then
         state <= rst;
      -- rising i_clk edge
      elsif (rising_edge (i_clk)) then
         state <= state_next;
      end if;
   end process;

   --
   -- Output logic
   --
   process(state)
   begin
      case state is
         when rst => o_detect <= '0';
         when A => o_detect <= '0';
         when B => o_detect <= '0';
         when C => o_detect <= '0';
         when D => o_detect <= '0';
         when E => o_detect <= '0';
         when F => o_detect <= '0';
         when G => o_detect <= '0';
         when det => o_detect <= '1';
      end case;
   end process;

end behavioral;
```

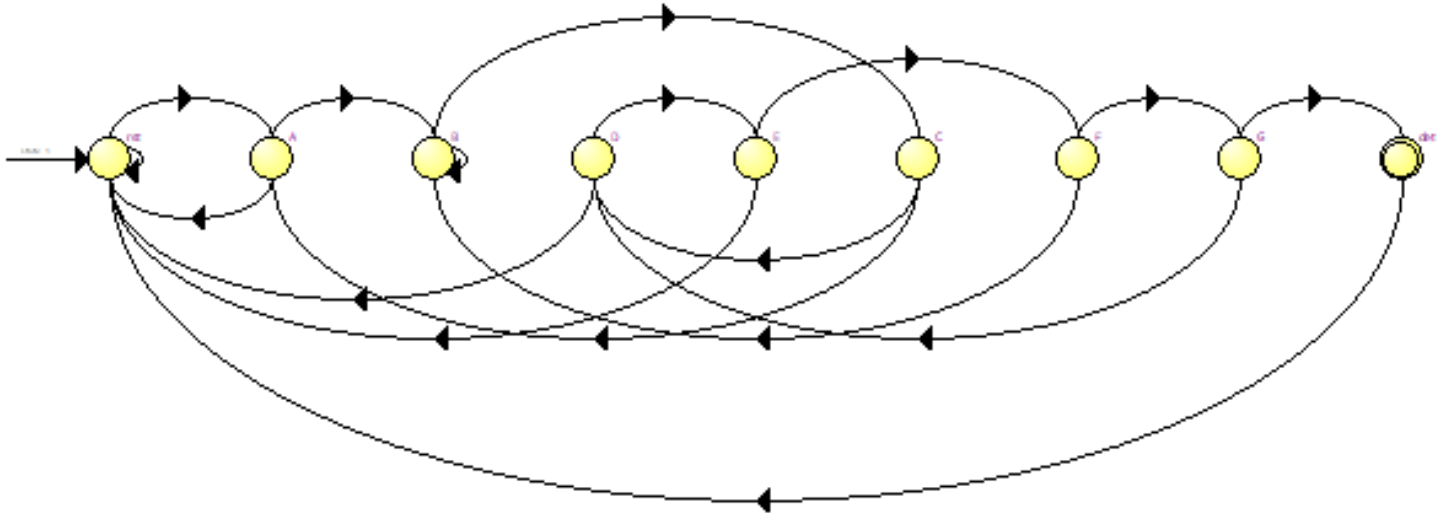
HDL FSM Examples

- Sequence detector - revisited



HDL FSM Examples

- Sequence detector - revisited



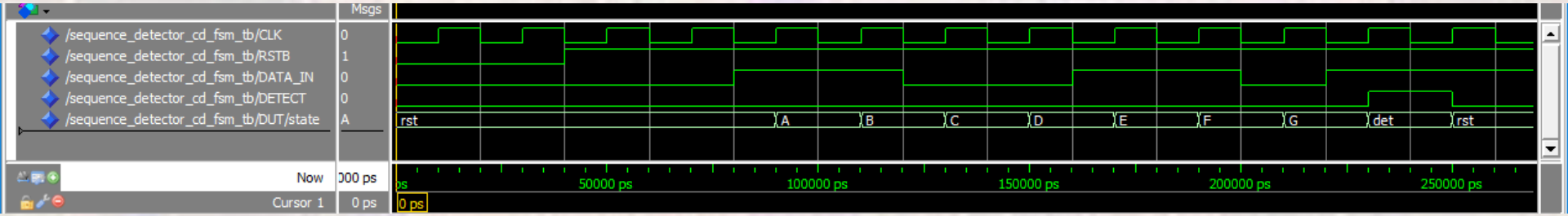
State Table

	Source State	Destination State	Condition
1	A	rst	(!i_data_in)
2	A	B	(i_data_in)
3	B	C	(!i_data_in)
4	B	B	(i_data_in)

Transitions | Encoding

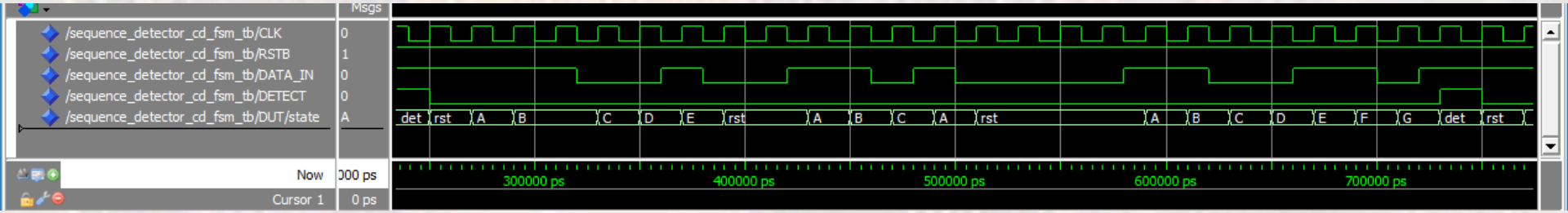
HDL FSM Examples

- Sequence detector - revisited



full sequence

re-start



repeat B

Back to start

Back to A

Back to start

full sequence