

13

```
library ieee;
use ieee.std_logic_1164.all;

entity decoder is
    port (d : IN std_logic_vector(2 downto 0);
          y : OUT std_logic_vector(7 downto 0))
end decoder;

architecture rtl_arc of decoder is
begin
    process(d)
    begin
        case d is
            when "000" => y <= "00000001";
            when "001" => y <= "00000010";
            when "010" => y <= "00000100";
            when "011" => y <= "00001000";
            when "100" => y <= "00010000";
            when "101" => y <= "00100000";
            when "110" => y <= "01000000";
            when "111" => y <= "10000000";
        end case;
    end process;
end rtl_arc;
```

```

library ieee;
use ieee.std_logic_1164.all;

entity encoder is
    port(d : IN std_logic_vector(7 downto 0);
         y : OUT std_logic_vector(2 downto 0);
         st : IN std_logic;
         yi, yo : OUT std_logic);
end encoder;

architecture rtl_arc of encoder is
begin
    process(d, st)
    begin
        if (st = '0') then
            if d(7) = '0' then y <= "000"; yi <= '0'; yo = '1';
            elsif d(6) = '0' then y <= "001"; yi <= '0'; yo = '1';
            elsif d(5) = '0' then y <= "010"; yi <= '0'; yo = '1';
            elsif d(4) = '0' then y <= "011"; yi <= '0'; yo = '1';
            elsif d(3) = '0' then y <= "100"; yi <= '0'; yo = '1';
            elsif d(2) = '0' then y <= "101"; yi <= '0'; yo = '1';
            elsif d(1) = '0' then y <= "110"; yi <= '0'; yo = '1';
            elsif d(0) = '0' then y <= "111"; yi <= '0'; yo = '1';
            else y <= "111"; yi <= '0'; yo = '0';
            end if
            y <= "111"; yi <= '1'; yo = '1';
        end if;
    end process;
end rtl_arc;

```

```
library ieee;
use ieee.std_logic_1164.all;

entity register4 is
    port(clk, clr: IN std_logic;
         d : IN std_logic_vector(3 downto 0);
         q : OUT std_logic_vector(3 downto 0));
end register4;

architecture rtl_arc of register4 is
begin
    process(clr, clk)
    begin
        if (clr = '1') then
            q <= "0000";
        elsif (clk'event and clk = '1') then
            q <= d;
        end if;
    end process;
end rtl_arc;
```

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity counter is
    port(clr, clk : IN std_logic;
          q : OUT std_logic_vector(3 downto 0) ;
          q : BUFFER std_logic_vector(3 downto 0));
end counter;

architecture rtl_arc of counter is
    signal temp : std_logic_vector(3 downto 0);
begin
    process(clr, clk, temp)
    begin
        if (clr = '1') then
            temp <= "0000";
        elsif (clk'event and clk = '1') then
            if (temp >= "1001") then
                temp <= "0000";
            else
                temp <= temp + 1;
            end if;
        end if;
    end process;
    q <= temp;
end rtl_arc;

```

```

library ieee;
use ieee.std_logic_1164.all;

entity counter is
    port(clr, clk, y: IN std_logic;
         q : OUT std_logic_vector(2 downto 0));
end counter;

architecture rtl_acr of counter is
    signal temp : std_logic_vector(2 downto 0);
begin
    process(clr, clk, y, temp)
    begin
        if (clr = '1') then temp = "000";
        else
            if (clk'event and clk = '1') then
                if (y = '1') then
                    case temp is
                        when "000" => temp <= "001";
                        when "001" => temp <= "011";
                        when "011" => temp <= "010";
                        when "010" => temp <= "110";
                        when "110" => temp <= "111";
                        when "111" => temp <= "101";
                        when "101" => temp <= "100";
                        when others => temp <= "000";
                    end case;
                else
                    case temp is
                        when "001" => temp <= "000";
                        when "011" => temp <= "001";
                        when "010" => temp <= "011";
                        when "110" => temp <= "010";
                        when "111" => temp <= "110";
                        when "101" => temp <= "111";
                        when "100" => temp <= "101";
                        when "000" => temp <= "100";
                        when others => temp <= "000";
                    end case;
                end if;
            end if;
        end if;
    end process;
end rtl_acr;

```

```

        end if;
    end process;
    q <= temp;
end rtl_arc;

```

21

```

library ieee;
use ieee.std_logic_1164.all;

entity DFA is
    port(clk, k : IN std_logic;
         q : OUT std_logic_vector(1 downto 0));
end DFA;

architecture rtl_arc of DFA is
    constant s0 : std_logic_vector(1 downto 0) := "00";
    constant s1 : std_logic_vector(1 downto 0) := "01";
    constant s2 : std_logic_vector(1 downto 0) := "10";
    constant s3 : std_logic_vector(1 downto 0) := "11";
    signal temp : std_logic_vector(1 downto 0) := "00";
begin
    process(clk, k)
    begin
        if (clk'event and clk = '1') then
            if (k = '0') then
                case temp is
                    when s0 => temp <= s1;
                    when s2 => temp <= s3;
                end case;
            else
                case temp is
                    when s1 => temp <= s2;
                    when s3 => temp <= s0;
                end case;
            end if;
        end if;
    end process;
    q <= temp;
end rtl_arc;

```