بسمه تعالی

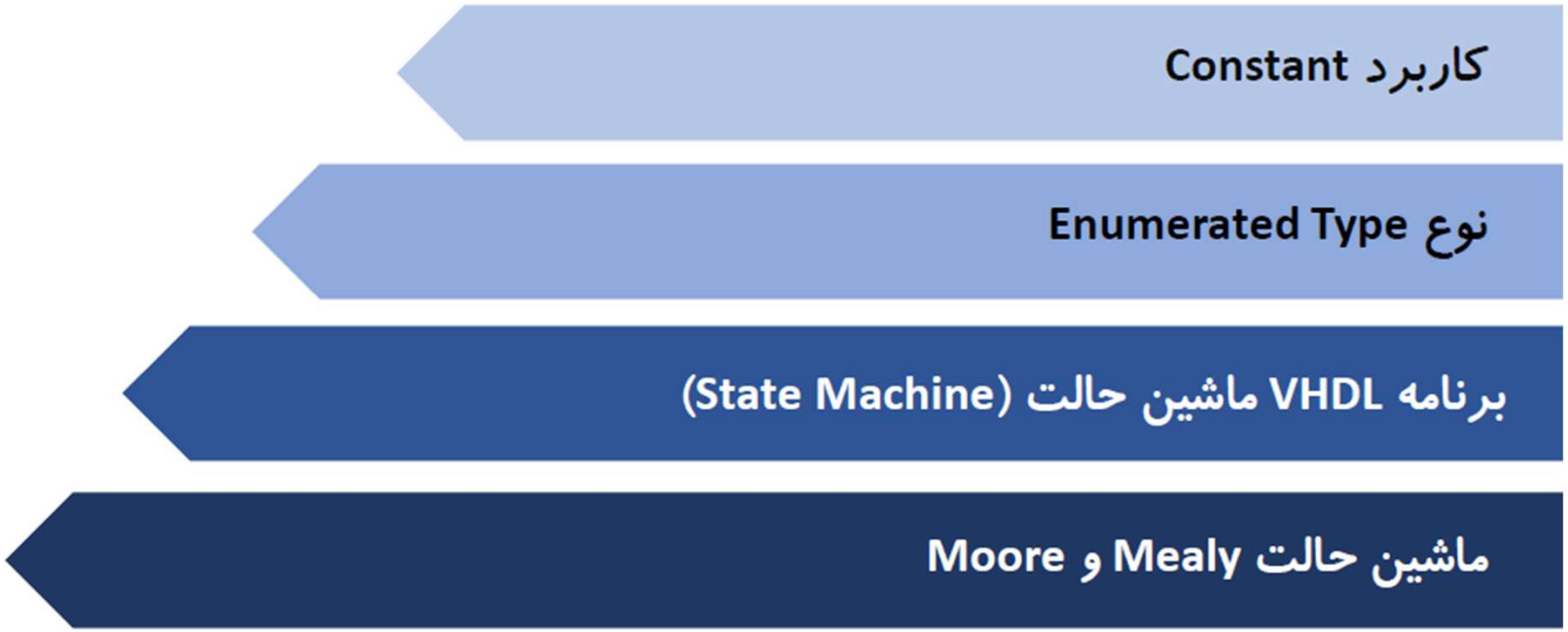
طراحی کامپیوتری سیستمهای دیجیتال

Computer-Aided Digital System Design

مدرس
فاطمه دارائی

f_daraei@semnan.ac.ir
https://fdaraei.profile.semnan.ac.ir

چشم‌انداز



اسلایدها برگرفته از جزوه دکتر محمد علی شفیعیان
https://shafieian-education.ir/computer-aided-digital-system-design-3/

کاربرد Constant

- زمانی که یک مدار را توصیف می‌کنیم، هر object یا ارتباط یا ... ممکن است signal، variable یا constant باشد که هر کدام داری نوع و نام است.
- constant یک نام سمبولیک اختیاری به یک مقدار، نسبت می‌دهد که در زمان شبیه‌سازی و ابتدا تا انتهای برنامه VHDL ثابت می‌ماند و تغییر نمی‌کند.
- constant برای توصیف بهتر یک طرح استفاده می‌شود و تغییرات طرح را در آینده آسان می‌سازد.
- معمولاً constant در محل اعلانات (Declarations) برنامه VHDL معرفی می‌شود.

کاربرد Constant

دانشگاه سمنان
Semnan University
پردیس فرزانگان

```
architecture sample1 of const is
      constant VCC:bit:='1';
      constant error_flag:Boolean:=true;
      constant period:time:=50 ns;
      constant SRAM:bit_vector(7 downto 0):=x"60F0";
      constant A:std_logic_vector (7 downto 0):=b"10101010"
begin
      a <= VCC;
      process
      constant count_limit:integer:=205;
      begin
      if counter = counter_limit then
            b <= '0';
      - - -
      wait for period;
      - - -
      end process;
end
```

کاربرد Constant

```
نام constant : نوع [ :=مقدار ثابت ]
```

constant دارای یک مقدار ثابت است و نوع (Type) آن هر نوعی می‌تواند باشدو در قسمت اعلانات قرار می‌گیرد.

process بدون لیست سیگنال‌ها

- process دارای یک لیست سیگنال‌ها است که اگر هر یک از سیگنال‌ها تغییر نمایند، process اجرا می‌شود.
- این حالت معمول‌ترین روش توصیف فلیپ‌فلاپ‌ها، ثبات‌ها و ... هستند.
- روش دیگر این است که process دارای لیست سیگنال‌ها نباشد.
- در این حالت، عبارات process پشت سر هم اجرا می‌شوند تا به یک عبارت wait برخورد کنند که در این شرایط مطابق با شرایط wait عمل می‌شود.

process بدون لیست سیگنال‌ها

```
architecture … of … is

begin
process نام:process
      Declaration
      عبارت ۱
      عبارت ۲
      ---
      wait until (condition)
      عبارت ۳
      عبارت ۴
      ---
      wait for (time)
end process [process نام]
```

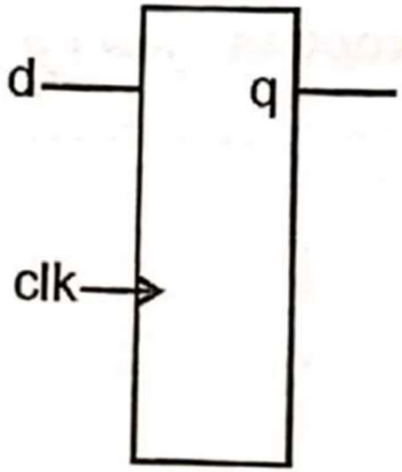
wait until
wait on
wait for (Time)

} عبارات wait

عبارات wait until و wait on

مثال

برنامه VHDL برای فلیپ‌فلاپ D با عبارت wait بنویسید به‌طوری‌که زمانی که پالس ساعت clk برابر ۱ شد، ورودی D به خروجی Q منتقل شود.



عبارات wait until و wait on

```
library ieee;
use ieee.std_logic_1164.all;
entity flipwait is
     port(clk : in std_logic;
          d : in std_logic;
          q: out std_logic);
end flipwait;
architecture flipwait_arch of flipwait is
begin
     process
     begin
          wait until clk='1';
          q <= d;
     end process;
end
```

عبارات wait for

مثال

برای ساختن پالس ساعت با دوره تناوب ۵۰ نانوثانیه از process و wait for استفاده کنید.

```
clock: process
     variable clktime:bit:='0';
begin
     clktime:= NOT clktime;
     clk <= clktime;
     wait for 50 ns;
end process clock;
```

نوع Enumerated Type

- نوع enumerated ابزار بسیار قوی است که با آن می‌توان مدارهای دیجیتال را متناسب با عملیاتی که انجام می‌دهند مدلسازی نمود.
- به عبارت دیگر، نوع یا Type سیگنال را متناسب با نوع مسأله تعریف کرد .

نام نوع Type is (لیستی از حروف یا نام‌ها)

```
type state_type is (S0,S1,S2,S3);
signal state : state_type;
```

Enumerated Type نوع

دانشگاه سمنان
Semnan University
پردیس فرزانگان

```
type instruction is (add,sub,lda,sta,inc);

entity ALU is
     port(Istr:in instruction;
          data2:inout Integer);
end;
```

```
type my_type is (Reset,Start,Executed);

type current_state,next_state : my_type;
```

Enumerated Type نوع

در طراحی ماشین حالت، باید نوع (Type) آن در قسمت اعلانات آرشیتکت توصیف شوند.

همچنین، سیگنالی شبیه state یا next_state و current_state برای قرار دادن حالت‌های مدار در آن تعریف شود.

برنامه VHDL برای ماشین حالت (State Machine) با استفاده از نوع Enumerated

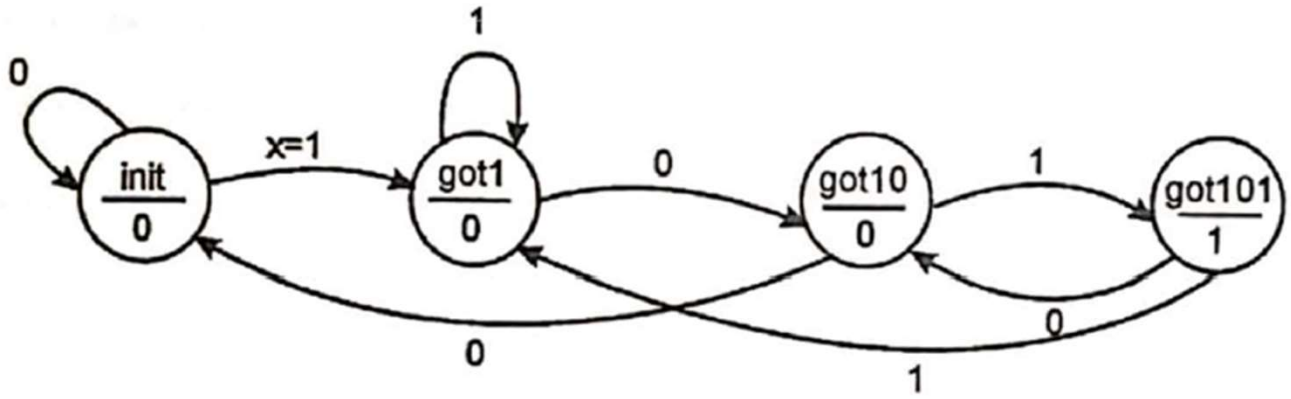
ماشین حالت برای کنترل سیستم‌های دیجیتال و کامپیوتر به‌کار می‌رود.

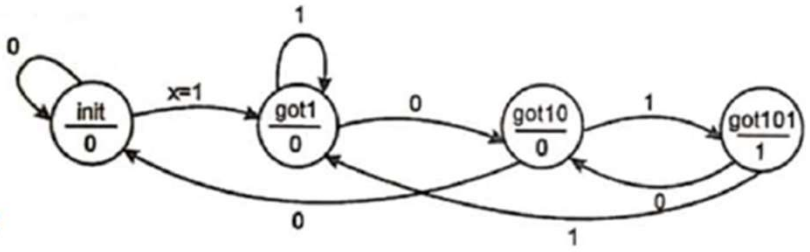
ماشین حالت
- ماشین حالت Moore: خروجی فقط تابع حالت فعلی مدار است.
- ماشین حالت Mealy: خروجی تابع حالت فعلی و ورودی‌های مدار است.

برنامه VHDL برای ماشین حالت (State Machine) با استفاده از نوع Enumerated

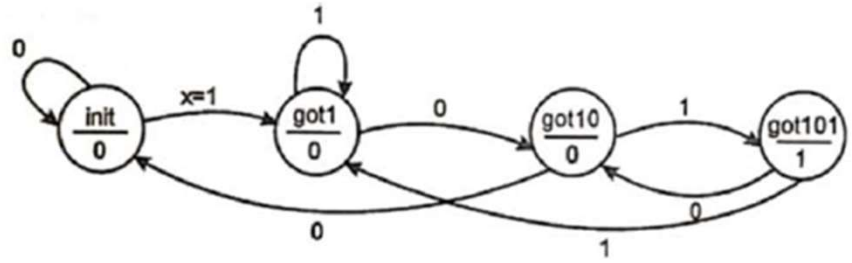
مثال

برنامه VHDL برای ماشین حالت Moore بنویسید که اگر در ورودی x به‌ترتیب مقادیر 101 آمد، خروجی g برابر 1 شود.

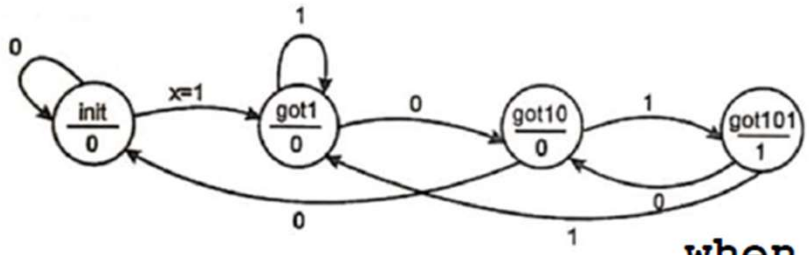




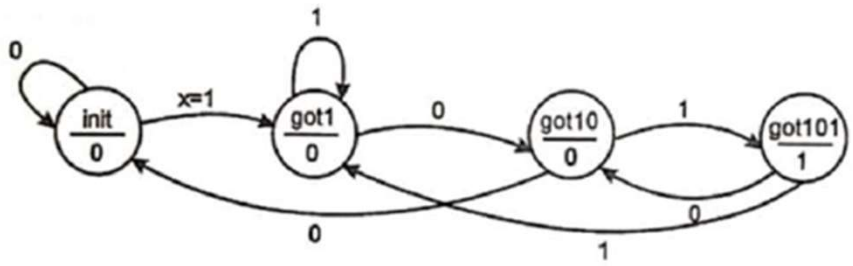

```
library ieee;
use ieee.std_logic_1164.all;
entity detector is
     port(x,clk : in std_logic;
          g: out std_logic);
end entity;
architecture behavior of detector is
     type states is (init,got1,got10,got101);
     signal current : states;
begin
     clock:process(clk)
     begin
          if (clk'event and clk='1') then
               case current is
```



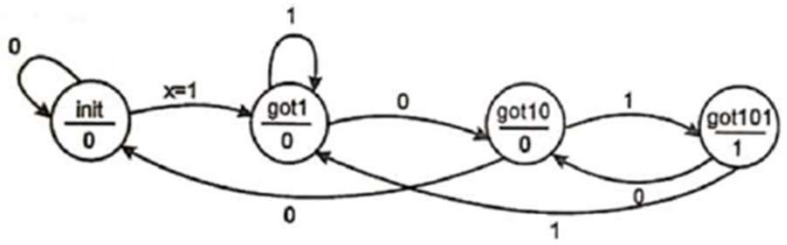

```
when init => if(x='0') then
                current <= init;
             else
                current <= got1;
             end if;
when got1 => if(x='1') then
                current <= got1;
             else
                current <= got10;
             end if;
```

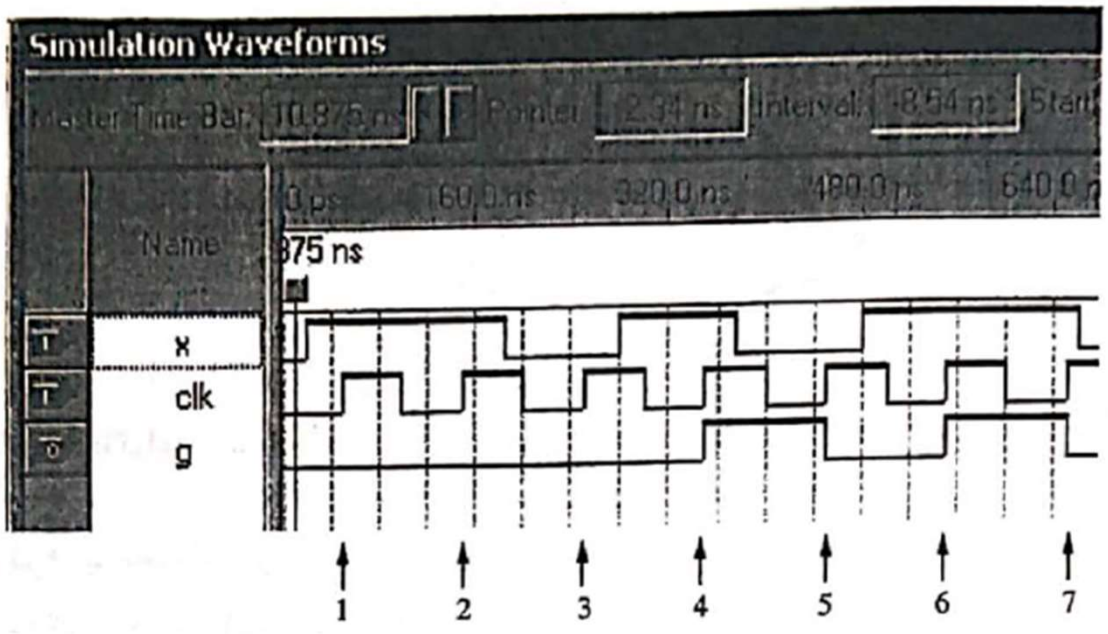



```
                when got10 => if(x='1') then
                                  current <= got101;
                              else
                                  current <= init;
                              end if;
                when got101 => if(x='1') then
                                  current <= got1;
                              else
                                  current <= got10;
                              end if;
            end case;
        end if
    end process;
```




```
output: process(current)
        begin
            case current is
                when init => g <= '0';
                when got1 => g <= '0';
                when got10 => g <= '0';
                when got101 => g <= '1';
            end case;
end architecture;
```

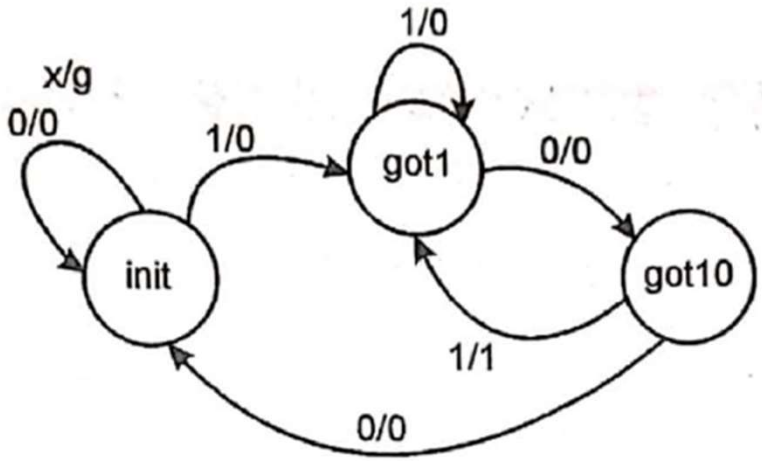

init
0
got1
0
got10
0
got101
1
x=1
0
1
0
1
0
1
0
1


Simulation Waveforms
Master Time Bar:
10.875 ns
Pointer:
2.34 ns
Interval:
-8.54 ns
Start:
0 ps
160.0 ns
320.0 ns
480.0 ns
640.0 n
Name:
875 ns
x
clk
g
1
2
3
4
5
6
7

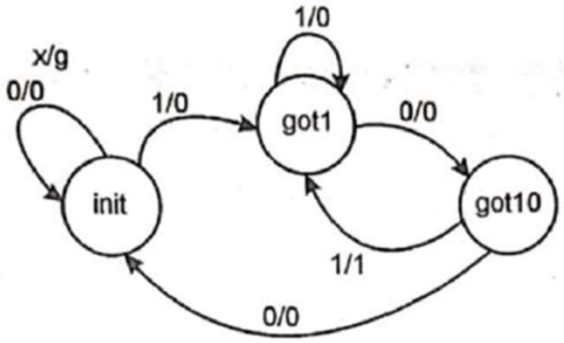
برنامه VHDL برای ماشین حالت (State Machine) با استفاده از نوع Enumerated

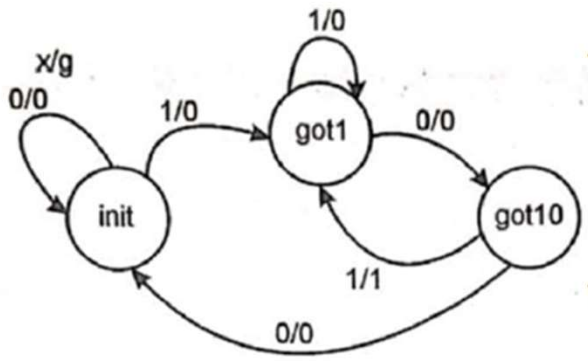
مثال

برنامه VHDL برای ماشین حالت Mealy بنویسید که اگر در ورودی x به‌ترتیب مقادیر 101 آمد، خروجی g برابر 1 شود.



```
library ieee;
use ieee.std_logic_1164.all;
entity mealy_detector is
     port(x,clk : in bit;
          g: out bit);
end entity;
architecture beh of mealy_ detector is
     type states is (init,got1,got10);
     signal current : states;
begin
     clock:process(clk)
     begin
          if (clk'event and clk='1') then
               case (current) is
```

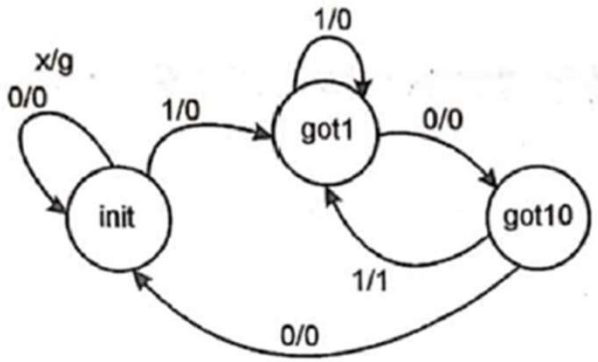




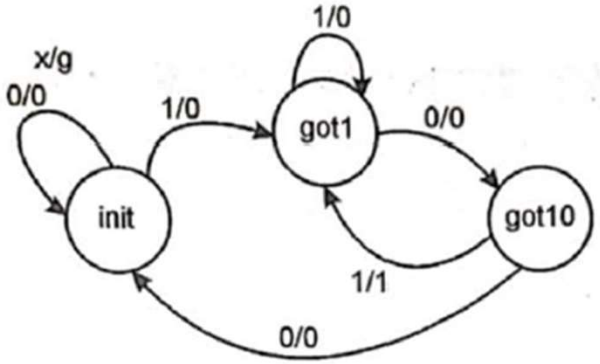

```
                when init =>
                     if(x='0') then current <= init;
                     else current <= got1;
                     end if;
                when got1 =>
                     if(x='1') then current <= got1;
                     else current <= got10;
                     end if;
                when got10 =>
                     if(x='1') then current <= got1;
                     else current <= init;
                     end if;
           end case;
      end if;
end process;
```

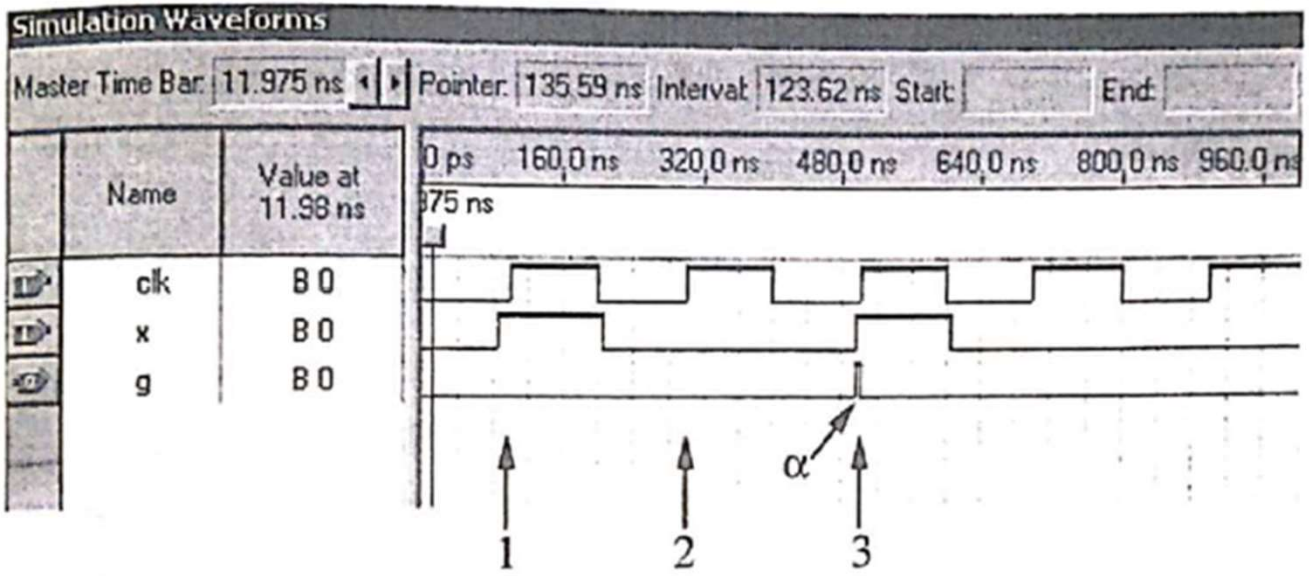
دانشگاه سمنان
Semnan University
پردیس فرزانگان

```
    output:process(current)
    begin
    case (current) is
    when init =>
         if (x='0') then g<='0';
         else g <= '0';
         endif;
    when got1 =>
         if (x='0') then g<='0';
         else g <= '0';
         endif;
    when got10 =>
         if (x='0') then g<='0';
         else g <= '1';
         endif;
    end case;
    end process;
end;
```



دانشگاه سمنان
Semnan University
پردیس فرزانگان


x/g
0/0
1/0
1/0
got1
0/0
init
got10
1/1
0/0


Simulation Waveforms
Master Time Bar: 11.975 ns
Pointer: 135.59 ns
Interval: 123.62 ns
Start:
End:
Name
Value at 11.98 ns
clk
x
g
B 0
0 ps
160.0 ns
320.0 ns
480.0 ns
640.0 ns
800.0 ns
960.0 ns
α
1
2
3

دانشگاه سمنان
Semnan University
پردیس فرزانگان

طراحی کامپیوتری سیستمهای دیجیتال

Computer-Aided Digital System Design

مثال های بیشتر

نمونه هایی از برنامه های زبان VHDL

مثال

برنامه VHDL بنویسید که تابع منطقی زیر را پیاده‌سازی کند.

$$F3 = \overline{LM}N + LM$$

L	M	N	F3
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

$$F3(L, M, N) = \sum(1, 6, 7)$$

مثال

برنامه VHDL بنویسید که تابع منطقی زیر را پیاده‌سازی کند.

$$F3(L,M,N) = \sum(1,6,7)$$

```
-- library declaration
library IEEE;
use IEEE.std_logic_1164.all;
-- entity
entity my_ckt_f3 is
port ( L,M,N : in  std_logic;
       F3    : out std_logic);
end my_ckt_f3;
-- architecture
architecture f3_8 of my_ckt_f3 is
signal t_sig : std_logic_vector(2 downto 0); -- local bundle
begin
t_sig <= (L & M & N); -- concatenation operator

with (t_sig) select
    F3 <= '1' when "001" | "110" | "111",
          '0' when others;
end f3_8;
```

مثال

با استفاده از if برنامه VHDL بنویسید که تابع منطقی زیر را پیاده‌سازی کند.

$$F_OUT(A,B,C) = A\overline{BC} + BC$$

```
-- library declaration
library IEEE;
use IEEE.std_logic_1164.all;
-- entity
entity my_ex is
port     (A,B,C : in  std_logic;
          F_OUT : out std_logic);
end my_ex;
-- architecture
architecture silly_example of my_ex is
begin
    proc1: process(A,B,C) is
    begin
        if (A = '1' and B = '0' and C = '0') then
            F_OUT <= '1';
        elsif (B = '1' and C = '1') then
            F_OUT <= '1';
        else
            F_OUT <= '0';
        end if;
    end process proc1;
end silly_example;
```

مثال

با استفاده از if برنامه VHDL بنویسید که تابع منطقی زیر را پیاده‌سازی کند.

$$F_OUT(A,B,C) = A\overline{BC} + BC$$

```
-- architecture
architecture bad_example of my_ex_7 is
begin
    proc1: process(A,B,C)
    begin
        if (A='0' and B='0' and C='0') or (B='1' and C='1') then
            F_OUT <= '1';
        else
            F_OUT <= '0';
        end if;
    end process proc1;
end bad_example;
```

مثال

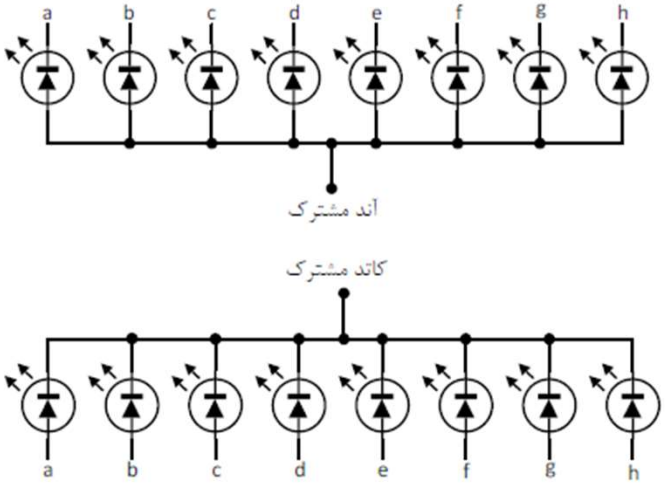
با استفاده از case برنامه VHDL بنویسید که تابع منطقی زیر را پیاده‌سازی کند.

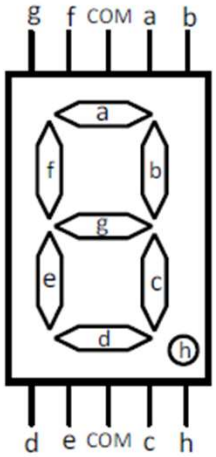
$$F_OUT(A,B,C) = A\overline{BC} + ABC + \overline{A}BC$$

```
-- library declaration
library IEEE;
use IEEE.std_logic_1164.all;
-- entity
entity my_example is
port (A,B,C : in  std_logic;
      F_OUT : out std_logic);
end my_example;
-- architecture
architecture my_soln_exam of my_example is
    signal ABC: std_logic_vector(2 downto 0);
begin
    ABC <= A & B & C; -- group signals for case statement
    my_proc: process (ABC)
    begin
        case (ABC) is
        when "100"  => F_OUT <= '1';
        when "011"  => F_OUT <= '1';
        when "111"  => F_OUT <= '1';
        when others => F_OUT <= '0';
        end case;
    end process my_proc;
end my_soln_exam;
```

مثال

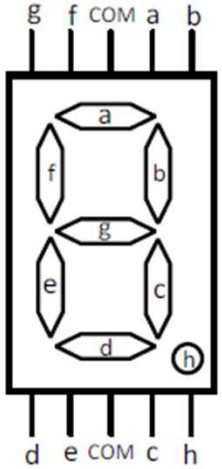
دیکدر نمایشگر هفت قسمتی (7-Segment)

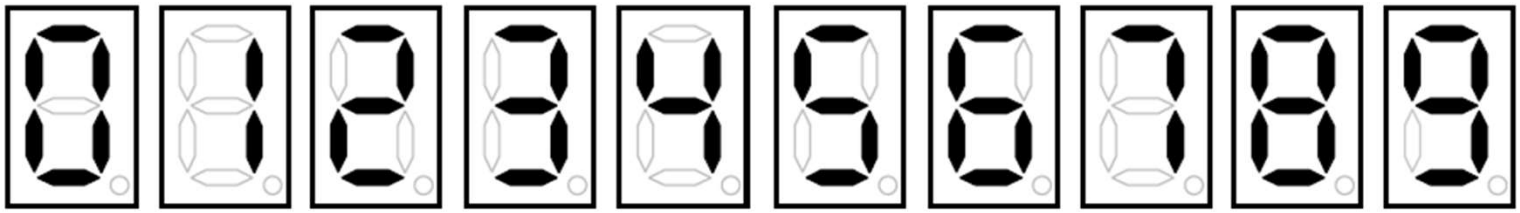




مثال

دیکدر نمایشگر هفت قسمتی (7-Segment)



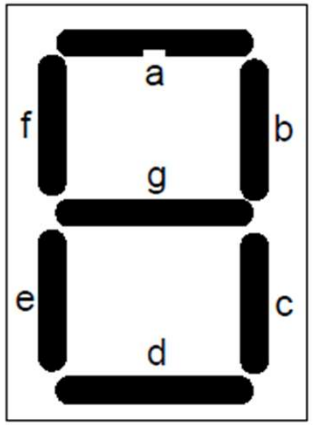


مثال

دیکدر نمایشگر هفت قسمتی (7-Segment)

آند مشترک

عدد	g	f	e	d	c	b	a
0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	1
2	0	1	0	0	1	0	0
3	0	1	1	0	0	0	0
4	0	0	1	1	0	0	1
5	0	0	1	0	0	1	0
6	0	0	0	0	0	1	0
7	1	1	1	1	0	0	0
8	0	0	0	0	0	0	0
9	0	0	1	0	0	0	0

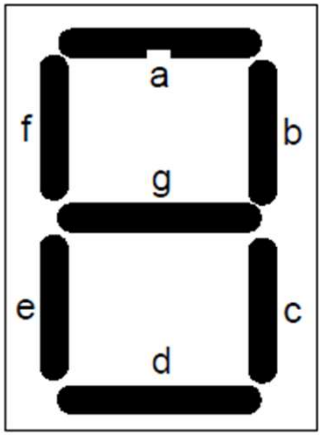


مثال

دیکدر نمایشگر هفت قسمتی (7-Segment)

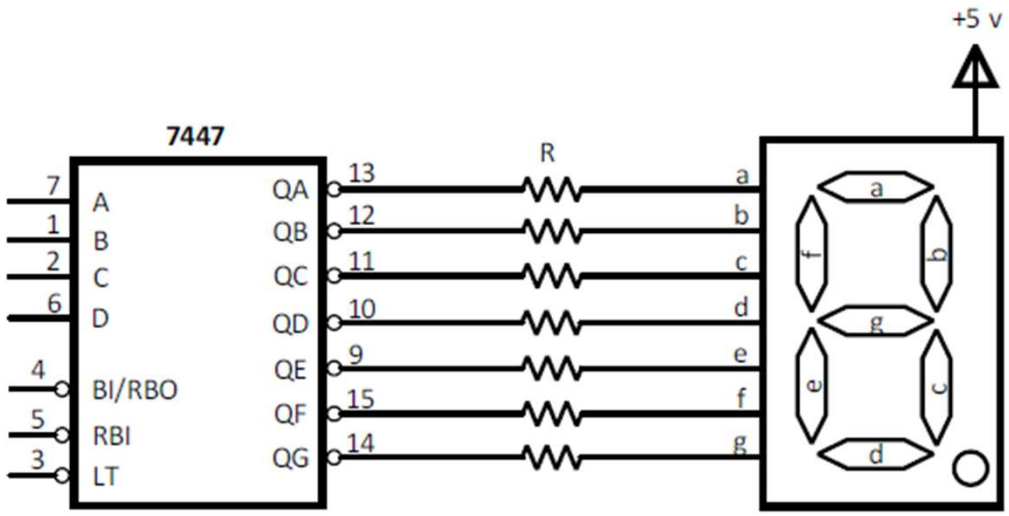
کاتد مشترک

عدد	g	f	e	d	c	b	a
0	0	1	1	1	1	1	1
1	0	0	0	0	1	1	0
2	1	0	1	1	0	1	1
3	1	0	0	1	1	1	1
4	1	1	0	0	1	1	0
5	1	1	0	1	1	0	1
6	1	1	1	1	1	0	1
7	0	0	0	0	1	1	1
8	1	1	1	1	1	1	1
9	1	1	0	1	1	1	1



مثال

دیکدر نمایشگر هفت قسمتی (7-Segment)



مثال

دیکدر نمایشگر هفت قسمتی (7-Segment)

```
--LED Decoder for 7 segment LED
--
library IEEE;
use IEEE.std_logic_1164.all;
--
-- Defining interface with entity
--
entity LEDDCD is
   port (
      D: in STD_LOGIC_VECTOR (3 downto 0);
      S: out STD_LOGIC_VECTOR (6 downto 0)
   );
end LEDDCD;
```

مثال

دیکدر نمایشگر هفت قسمتی (7-Segment)

```
--
-- Defining architecure of entity
--
--S=abcdefg
architecture LEDDCD_arch of LEDDCD is
begin
with D select
S <= "1111110" when "0000",--   Display 0
    "0110000" when "0001",--   Display 1
    "1101101" when "0010",--   Display 2
    "1111001" when "0011",--   Display 3
    "0110011" when "0100",--   Display 4
    "1011011" when "0101",--   Display 5
    "0011111" when "0110",--   Display 6
    "1110000" when "0111",--   Display 7
    "1111111" when "1000",--   Display 8
    "1110011" when "1001",--   Display 9
   "0000000" when others;--   Blank for non decimal
end LEDDCD_arch;
```

مثال

برنامه VHDL برای یک ثبات هشت بیتی با فلیپ‌فلاپ D بنویسید که:

- اگر ورودی reset برابر 1 شود، خروجی ثبات q صفر شود، در غیر این‌صورت:
  - در لبه بالارونده پالس ساعت اگر:
  - اگر initial برابر 1 گردد خروجی ثبات q مساوی 11111111 شود و در غیر این‌صورت ورودی d به خروجی q منتقل شود.

```
library IEEE;
use IEEE.std_logic_1164.all;

entity registdr is
   port (
      d: in STD_LOGIC_VECTOR (0 to 7);
      clk,reset,initial: in STD_LOGIC;
      q: out STD_LOGIC_VECTOR (0 to 7)
   );
end registdr;
```

```
architecture registdr_arch of registdr is
begin
process(clk,reset)
        begin
           if reset='1' then
           q<=b"00000000";
           elsif (clk'event and clk='1') then
              if initial ='1' then
              q<=b"11111111";
              else
              q<=d;
              end if;
            end if;
end process;
end registdr_arch;
```