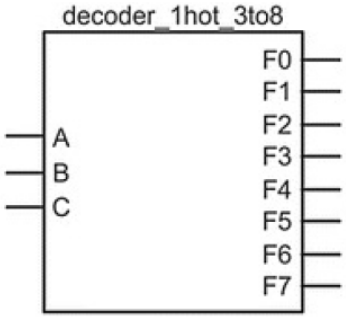
Пример “3-to-8 One-Hot” декодера (VHDL модел – конкурентна додела сигнала и логички оператори)



A	B	C	F7	F6	F5	F4	F3	F2	F1	F0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

$F0 = \sum_{A,B,C}(0) = A'\cdot B'\cdot C'$

$F1 = \sum_{A,B,C}(1) = A'\cdot B'\cdot C$

$F2 = \sum_{A,B,C}(2) = A'\cdot B\cdot C'$

$F3 = \sum_{A,B,C}(3) = A'\cdot B\cdot C$

$F4 = \sum_{A,B,C}(4) = A\cdot B'\cdot C'$

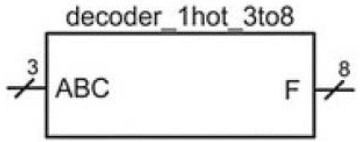
$F5 = \sum_{A,B,C}(5) = A\cdot B'\cdot C$

$F6 = \sum_{A,B,C}(6) = A\cdot B\cdot C'$

$F7 = \sum_{A,B,C}(7) = A\cdot B\cdot C$

```
entity decoder_1hot_3to8 is
   port (A,B,C                    : in  bit;
         F0,F1,F2,F3,F4,F5,F6,F7  : out bit);
end entity;

architecture decoder_1hot_3to8_arch of decoder_1hot_3to8 is

 begin

   F0 <= (not A) and (not B) and (not C);
   F1 <= (not A) and (not B) and (C);
   F2 <= (not A) and (B)     and (not C);
   F3 <= (not A) and (B)     and (C);
   F4 <= (A)     and (not B) and (not C);
   F5 <= (A)     and (not B) and (C);
   F6 <= (A)     and (B)     and (not C);
   F7 <= (A)     and (B)     and (C);

end architecture;
```

Пример “3-to-8 One-Hot” декодера (VHDL модел – условна/селелциона додела сигнала)



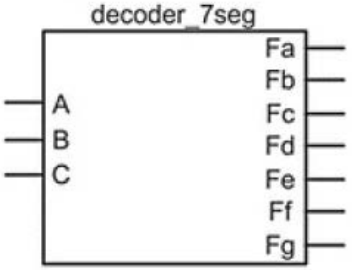
ABC	F(7)	F(6)	F(5)	F(4)	F(3)	F(2)	F(1)	F(0)
“000”	0	0	0	0	0	0	0	1
“001”	0	0	0	0	0	0	1	0
“010”	0	0	0	0	0	1	0	0
“011”	0	0	0	0	1	0	0	0
“100”	0	0	0	1	0	0	0	0
“101”	0	0	1	0	0	0	0	0
“110”	0	1	0	0	0	0	0	0
“111”	1	0	0	0	0	0	0	0

```
entity decoder_1hot_3to8 is
    port (ABC : in bit_vector(2 downto 0);
          F   : out bit_vector(7 downto 0));
end entity;
```

```
architecture decoder_1hot_3to8_arch of decoder_1hot_3to8 is

 begin

  F <= "00000001" when (ABC = "000") else
       "00000010" when (ABC = "001") else
       "00000100" when (ABC = "010") else
       "00001000" when (ABC = "011") else
       "00010000" when (ABC = "100") else
       "00100000" when (ABC = "101") else
       "01000000" when (ABC = "110") else
       "10000000" when (ABC = "111");

end architecture;
```

```
architecture decoder_1hot_3to8_arch of decoder_1hot_3to8 is

 begin

  with (ABC) select
     F <= "00000001" when "000",
          "00000010" when "001",
          "00000100" when "010",
          "00001000" when "011",
          "00010000" when "100",
          "00100000" when "101",
          "01000000" when "110",
          "10000000" when "111";

end architecture;
```

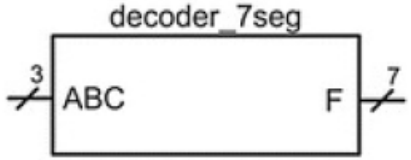
Декодер за 7-сегментни дисплеј (VHDL модел – конкурентна додела сигнала и логички оператори)



A	B	C	Fa	Fb	Fc	Fd	Fe	Ff	Fg
0	0	0	1	1	1	1	1	1	0
0	0	1	0	1	1	0	0	0	0
0	1	0	1	1	0	1	1	0	1
0	1	1	1	1	1	1	0	0	1
1	0	0	0	1	1	0	0	1	1
1	0	1	1	0	1	1	0	1	1
1	1	0	1	0	1	1	1	1	1
1	1	1	1	1	1	0	0	0	0

```
entity decoder_7seg is
   port (A,B,C                  : in  bit;
         Fa,Fb,Fc,Fd,Fe,Fg,Fg : out bit);
end entity;

architecture decoder_7seg_arch of decoder_7seg is

 begin

    Fa <= ((not A) and (not C)) or B or (A and C);
    Fb <= ((not B) and (not C)) or (not A) or (B and C);
    Fc <= A or (not B) or C;
    Fd <= ((not A) and (not C)) or ((not A) and B) or (B and (not C))
          or (A and (not B) and C);
    Fe <= ((not A) and (not C)) or (B and (not C));
    Ff <= ((not B) and (not C)) or (A and (not C)) or (A and (not B));
    Fg <= ((not A) and B) or (A and (not C)) or (A and (not B));

end architecture;
```

Декодер за 7-сегментни дисплеј (VHDL модел – условна/селелциона додела сигнала)



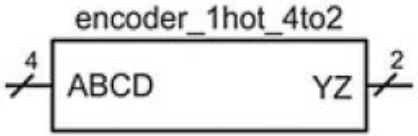
	a	b	c	d	e	f	g
ABC	F(6)	F(5)	F(4)	F(3)	F(2)	F(1)	F(0)
"000"	1	1	1	1	1	1	0
"001"	0	1	1	0	0	0	0
"010"	1	1	0	1	1	0	1
"011"	1	1	1	1	0	0	1
"100"	0	1	1	0	0	1	1
"101"	1	0	1	1	0	1	1
"110"	1	0	1	1	1	1	1
"111"	1	1	1	0	0	0	0

```
entity decoder_7seg is
  port (ABC  : in  bit_vector(2 downto 0);
        F    : out bit_vector(6 downto 0));
end entity;
```

```
architecture decoder_7seg_arch of decoder_7seg is
 begin
   F <= "1111110" when (ABC = "000") else
        "0110000" when (ABC = "001") else
        "1101101" when (ABC = "010") else
        "1111001" when (ABC = "011") else
        "0110011" when (ABC = "100") else
        "1011011" when (ABC = "101") else
        "1011111" when (ABC = "110") else
        "1110000" when (ABC = "111");
end architecture;
```

```
architecture decoder_7seg_arch of decoder_7seg is
 begin
   with (ABC) select
    F <= "1111110" when "000",
         "0110000" when "001",
         "1101101" when "010",
         "1111001" when "011",
         "0110011" when "100",
         "1011011" when "101",
         "1011111" when "110",
         "1110000" when "111";
end architecture;
```

Пример “4-to-2 One-Hot” кодера (VHDL модел – конкурентна/условна/селекциона додела сигнала)



ABCD	YZ
“0001”	“00”
“0010”	“01”
“0100”	“10”
“1000”	“11”

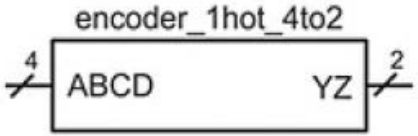
```
entity encoder_1hot_4to2 is
    port (ABCD  : in  bit_vector(3 downto 0);
          YZ    : out bit_vector(1 downto 0));
end entity;
```

```
architecture encoder_1hot_4to2_arch of encoder_1hot_4to2 is

 begin

  YZ(1) <= ABCD(3) or ABCD(2);
  YZ(0) <= ABCD(3) or ABCD(1);

end architecture;
```

```
architecture encoder_1hot_4to2_arch of encoder_1hot_4to2 is

 begin

  YZ <= "00" when (ABCD = "0001") else
        "01" when (ABCD = "0010") else
        "10" when (ABCD = "0100") else
        "11" when (ABCD = "1000") else
        “00”;

end architecture;
```

```
architecture encoder_1hot_4to2_arch of encoder_1hot_4to2 is

 begin

  with (ABCD) select
    YZ <= "00" when "0001",
          "01" when "0010",
          "10" when "0100",
          "11" when "1000",
          “00” when others;

end architecture;
```

Пример “4-to-2 One-Hot” кодера (VHDL модел – конкурентна/условна/селекциона додела сигнала)



ABCD	YZ
“0001”	“00”
“0010”	“01”
“0100”	“10”
“1000”	“11”

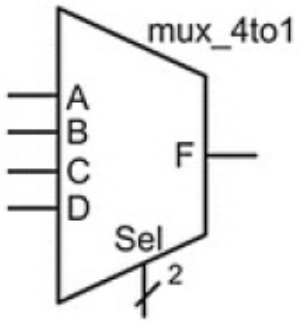
```
entity encoder_1hot_4to2 is
    port (ABCD  : in  bit_vector(3 downto 0);
          YZ    : out bit_vector(1 downto 0));
end entity;
```

```
architecture encoder_1hot_4to2_arch of encoder_1hot_4to2 is

 begin

  YZ(1) <= ABCD(3) or ABCD(2);
  YZ(0) <= ABCD(3) or ABCD(1);

end architecture;
```

```
architecture encoder_1hot_4to2_arch of encoder_1hot_4to2 is

 begin

  YZ <= "00" when (ABCD = "0001") else
        "01" when (ABCD = "0010") else
        "10" when (ABCD = "0100") else
        "11" when (ABCD = "1000") else
        “00”;

end architecture;
```

```
architecture encoder_1hot_4to2_arch of encoder_1hot_4to2 is

 begin

  with (ABCD) select
    YZ <= "00" when "0001",
          "01" when "0010",
          "10" when "0100",
          "11" when "1000",
          “00” when others;

end architecture;
```

Пример “2-to-1 мултиплексера (VHDL модел – конкурентна/условна/селекциона додела сигнала)



Sel	F
“00”	A
“01”	B
“10”	C
“11”	D

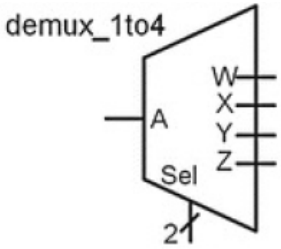
```
entity mux_4to1 is
    port (A,B,C,D : in  bit;
          Sel     : in  bit_vector(1 downto 0);
          F       : out bit);
end entity;
```

```
architecture mux_4to1_arch of mux_4to1 is
 begin
  F <= (A and not Sel(0) and not Sel(1)) or
       (B and not Sel(0) and     Sel(1)) or
       (C and     Sel(0) and not Sel(1)) or
       (D and     Sel(0) and     Sel(1));
end architecture;
```

```
architecture mux_4to1_arch of mux_4to1 is
 begin
  F <= A when (Sel = "00") else
       B when (Sel = "01") else
       C when (Sel = "10") else
       D when (Sel = "11");
end architecture;
```

```
architecture mux_4to1_arch of mux_4to1 is
 begin
   with (Sel) select
     F <= A when "00",
          B when "01",
          C when "10",
          D when "11";
end architecture;
```

Пример “1-to-2 демултиплексера (VHDL модел – конкурентна/условна/селекциона додела сигнала)



Sel	W	X	Y	Z
“00”	A	0	0	0
“01”	0	A	0	0
“10”	0	0	A	0
“11”	0	0	0	A

```
entity demux_1to4 is
    port (A       : in  bit;
          Sel     : in  bit_vector(1 downto 0);
          W,X,Y,Z : out bit);
end entity;
```

```
architecture demux_1to4_arch of demux_1to4 is

 begin

  W <=  A and not Sel(0) and not Sel(1);
  X <=  A and not Sel(0) and     Sel(1);
  Y <=  A and     Sel(0) and not Sel(1);
  Z <=  A and     Sel(0) and     Sel(1);

end architecture;
```

```
architecture demux_1to4_arch of demux_1to4 is

 begin

  W <= A when (Sel = "00") else '0';
  X <= A when (Sel = "01") else '0';
  Y <= A when (Sel = "10") else '0';
  Z <= A when (Sel = "11") else '0';

end architecture;
```

```
architecture demux_1to4_arch of demux_1to4 is

 begin

   with (Sel) select
     W <= A when "00",'0' when others;

   with (Sel) select
     X <= A when "01",'0' when others;

   with (Sel) select
     Y <= A when "10",'0' when others;

   with (Sel) select
     Z <= A when "11",'0' when others;

end architecture;
```