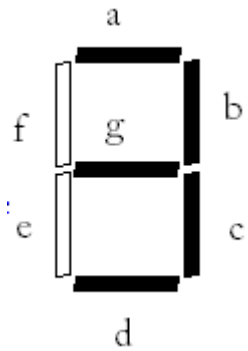


# BCD to Seven segment display

- It is a digital circuit that decodes BCD numbers into 7 segment numbers that can be used for 7 segment displays and other applications.



INPUTS	OUTPUT						
	a	b	c	d	e	f	G
ABCD	a	b	c	d	e	f	G
0000	1	1	1	1	1	1	0
0001	0	1	1	0	0	0	0
0010	1	1	0	1	1	0	1
0011	1	1	1	1	0	0	1
0100	0	1	1	0	0	1	1
0101	1	0	1	1	0	1	1
0110	1	0	1	1	1	1	1
0111	1	1	1	0	0	0	0
1000	1	1	1	1	1	1	1
1001	1	1	1	1	0	1	1

# Code converter in behavioral

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
entity bcd is
    Port ( i : in STD_LOGIC_VECTOR (0 to 3);
          g : out STD_LOGIC_VECTOR (0 to 6));
end bcd;
architecture beh_f of bcd is
begin
    process(i)
    begin
        case i is
            when "0000" => g <= "1111110";
            when "0001" => g <= "0110000";
            when "0010" => g <= "1101101";
            when "0011" => g <= "1111001";
            when "0100" => g <= "0110011";
            when "0101" => g <= "1011011";
            when "0110" => g <= "1011110";
            when "0111" => g <= "1110000";
            when "1000" => g <= "1111111";
            when "1001" => g <= "1111011";
            when others => g <= "1111111";
        end case;
    end process;
end Beh_f;
```

## Code convertor using structural code

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
entity bcd is
    Port ( i : in  STD_LOGIC_VECTOR (0 to 3);
          g : out STD_LOGIC_VECTOR (0 to 6));
end bcd;
architecture stur_f of bcd is
    Component deco_g
    Port( in: in bit_vector(0 to 3);
          o1: out bit_vector(0 to 6));
    End component;
    Begin
    V0: decod_g port map (I,g);
    End stur_f;
```

## Data flow

entity bs is

```
Port ( i : in STD_LOGIC_VECTOR (3 downto 0);  
      o : out STD_LOGIC_VECTOR (6 downto 0));
```

end bs;

architecture Behavioral of bs is

begin

```
o <= "1111110" when i ="0000" else  
     "0110000" when i ="0001" else  
     "1101101" when i ="0010" else  
     "1111001" when i ="0011" else  
     "0110011" when i ="0100" else  
     "1011011" when i ="0101" else  
     "1011111" when i ="0110" else  
     "1110000" when i ="0111" else  
     "1111111" when i ="1000" else  
     "1111011" when i ="1001";
```

end Behavioral;