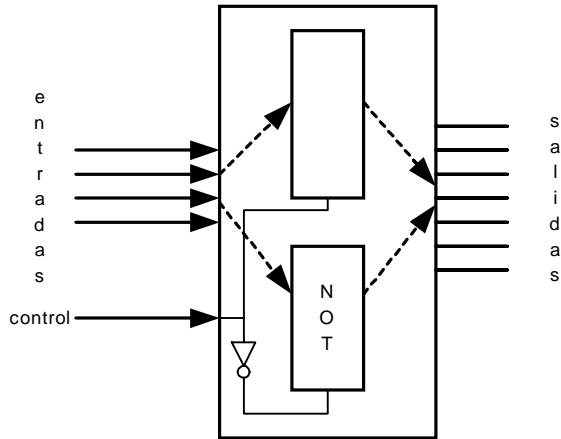


DECODIFICADOR DE 7 SEGMENTOS CON UNA ENTRADA DE CONTROL

Si control = '0' → ánodo común

Si control = '1' → cátodo común



```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity ckto8 is
    Port ( entrada : in std_logic_vector(3 downto 0);
          salida : out std_logic_vector(6 downto 0);
          control : in std_logic);
end ckto8;
```

```
architecture flujo of ckto8 is
    signal paso:std_logic_vector(6 downto 0);
begin
```

```
    with entrada select
        paso<="1000000" when "0000",
            "1111001" when "0001",
            "0100100" when "0010",
            "0110000" when "0011",
            "0011001" when "0100",
            "0010010" when "0101",
            "0000010" when "0110",
            "1111000" when "0111",
            "0000000" when "1000",
            "0010000" when "1001",
            "0001000" when "1010",
            "0000010" when "1011",
```

```

                "1000110" when "1100",
                "0100001" when "1101",
                "0000110" when "1110",
                "0001110" when others;
--para catodo comun control=1
--para anodo comun control=0
-- es un decoder para catodo comun... los segmentos se
activan con nivel bajo
-- de binario a hexadecimal
                salida <= paso when control='1' else
not paso;
end flujo;

```

PARA SIMULAR EL CIRCUITO

```

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY testbench IS
END testbench;

ARCHITECTURE behavior OF testbench IS

    COMPONENT ckto8
    PORT(
        entrada : IN std_logic_vector(3 downto 0);
        control : IN std_logic;
        salida : OUT std_logic_vector(6 downto 0)
    );
    END COMPONENT;

    SIGNAL entrada : std_logic_vector(3 downto 0);
    SIGNAL salida : std_logic_vector(6 downto 0);
    SIGNAL control : std_logic;

BEGIN

    uut: ckto8 PORT MAP(
        entrada => entrada,
        salida => salida,
        control => control
    );

-- *** Test Bench - User Defined Section ***
tb : PROCESS
BEGIN

```

```
    entrada <= "0000";control <= '0';
    wait for 20 ns;
    entrada <= "0001";control <= '0';
    wait for 20 ns;
    entrada <= "0010";control <= '0';
    wait for 20 ns;
    entrada <= "0011";control <= '0';
    wait for 20 ns;
    entrada <= "0100";control <= '0';
    wait for 20 ns;
    entrada <= "0101";control <= '0';
    wait for 20 ns;
    entrada <= "0110";control <= '0';
    wait for 20 ns;
    entrada <= "0111";control <= '0';
    wait for 20 ns;
    entrada <= "1000";control <= '0';
    wait for 20 ns;
    entrada <= "1001";control <= '0';
    wait for 20 ns;
    entrada <= "1010";control <= '0';
    wait for 20 ns;
    entrada <= "1011";control <= '0';
    wait for 20 ns;
    entrada <= "1100";control <= '0';
    wait for 20 ns;
    entrada <= "1101";control <= '0';
    wait for 20 ns;
    entrada <= "1111";control <= '0';
    wait for 20 ns;
--para catodo comun
    entrada <= "0000";control <= '1';
    wait for 20 ns;
    entrada <= "0001";control <= '1';
    wait for 20 ns;
    entrada <= "0010";control <= '1';
    wait for 20 ns;
    entrada <= "0011";control <= '1';
    wait for 20 ns;
    entrada <= "0100";control <= '1';
    wait for 20 ns;
    entrada <= "0101";control <= '1';
    wait for 20 ns;
    entrada <= "0110";control <= '1';
    wait for 20 ns;
    entrada <= "0111";control <= '1';
```

```

wait for 20 ns;
entrada <= "1000";control <= '1';
wait for 20 ns;
entrada <= "1001";control <= '1';
wait for 20 ns;
entrada <= "1010";control <= '1';
wait for 20 ns;
entrada <= "1011";control <= '1';
wait for 20 ns;
entrada <= "1100";control <= '1';
wait for 20 ns;
entrada <= "1101";control <= '1';
wait for 20 ns;
entrada <= "1111";control <= '1';
wait for 20 ns;

```

```
wait; -- will wait forever
```

```
END PROCESS;
```

```
-- *** End Test Bench - User Defined Section ***
```

```
END;
```

