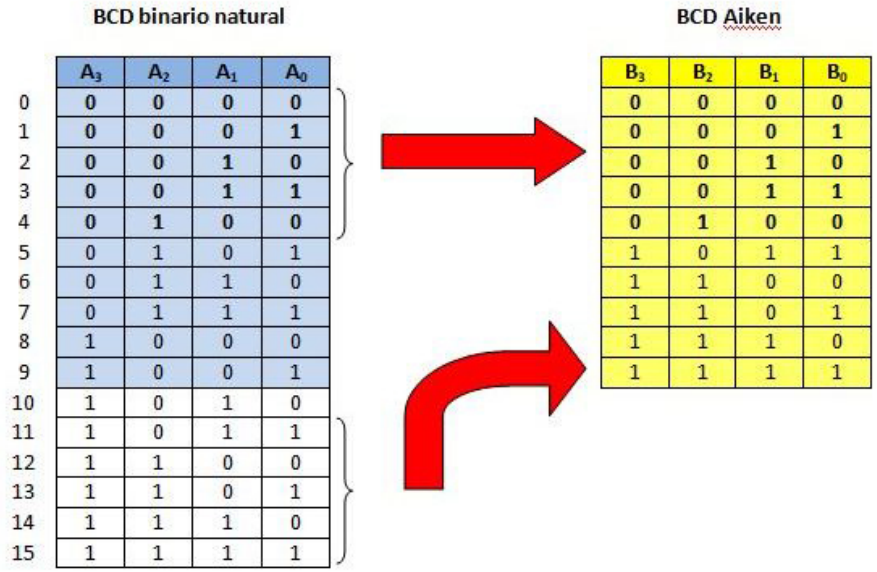
Presentador: Victor Miguel Barrera Peña

Tema: 12 Conversor de BCD Natural a Aiken

Teoría

Hay que recordar como es electrónicamente, existe.



	A_3	A_2	A_1	A_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

B_3	B_2	B_1	B_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

Hay que recordar la códificación Aiken, y para ello veremos la tabla que muestre la codificación.

Vamos a verlo a detalle en la siguiente imagen:

- El cuadro rojo se ignora
- En BCD aiken los números en rojos son el número de BCD natural que le corresponde.

BCD binario natural

	A_3	A_2	A_1	A_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

BCD Aiken

B_3	B_2	B_1	B_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

0 1 2 3 4 11 12 13 14 15

Decimal	BCD natural (8421)	BCD Aiken (2421)
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	1011
6	0110	1100
7	0111	1101
8	1000	1110
9	1001	1111

Veamos una tabla para transcodificar.

Funcionamiento

Recuerda la idea de como funciona un decodificador, es lo mismo , pero en este caso no sólo códifica y listo, sino que la entrada ya esta codificada y sólo necesita transformarse en otra codificación

Veamos el código

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all; --nuevos tipos de datos usigned
```

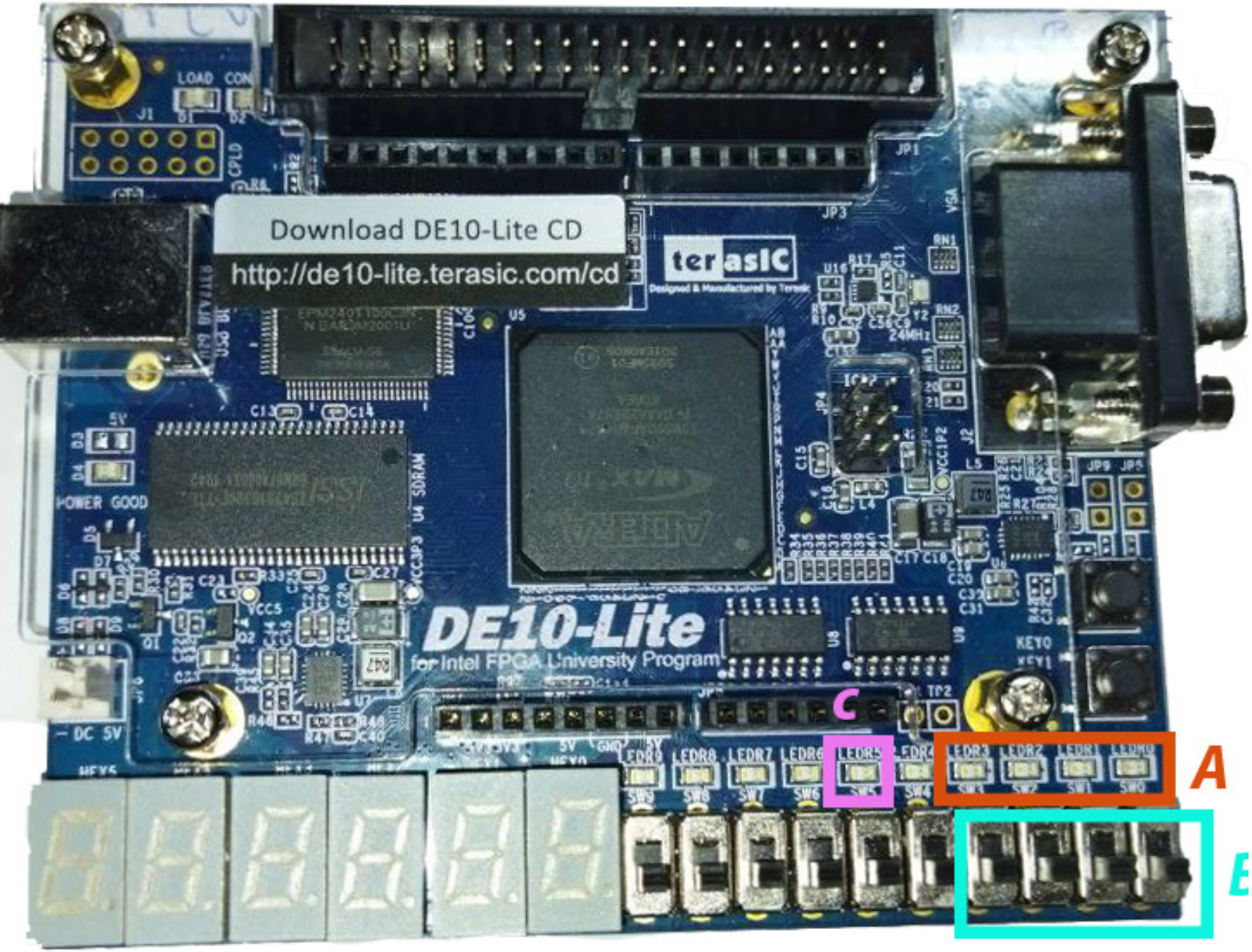
```
entity p12 is
        port(
                n_i : in std_logic_vector(3 downto 0);
                a_o : out std_logic_vector(3 downto 0);
                error : out std_logic
                );
end entity p12;
```

```
architecture behavior of p12 is begin

        a_o <= n_i when unsigned(n_i) < 5  else
                "0000"  when  9 <  unsigned(n_i)
                else std_logic_vector(unsigned(n_i) + 6);

        -- saber error
        error <= '1'  when  9 <  unsigned(n_i)
                          else '0';

end architecture behavior;
```



Asignación

- **A** Son los led de salida.
- **B** Son los switch de entrada.
- **C** Error codificiación

Veamos su comportamiento

Muchas gracias por ver el video