

ANEJO 1: PROGRAMACIÓN PLD

1.1 TABLA DE VERDAD

Para generar el programa en VHDL o el archivo JEDEC para programar el PLD, en el programa “BooleanDeusto” es necesario introducir la tabla de verdad del circuito selector. Dicho circuito tendrá 6 entradas, las cuales son:

- Swi0: determina si se encuentra funcionando en modo motor (0) o en modo freno (1).
- Swi1: determina si el convertidor debe trabajar en modo boost (0) o en modo buck (1).
- Swi2: determina si la tensión de salida es mayor (0) o menor (1) que 32 V.
- Swi3: determina si la tensión de salida es mayor (0) o menor (1) que 34 V.
- Swi4: determina si la tensión de salida es mayor (0) o menor (1) que 38 V.
- Swi5: determina si la tensión de salida es mayor (0) o menor (1) que 42 V.

El PLD tendrá que tener 4 salidas para llevar a cabo el circuito lógico diseñado. Son las siguientes:

- Ena0: adquirirá el valor 0 si el convertidor debe trabajar como un boost y un 1 si debe trabajar como buck.
- Ena1: adquirirá el valor 1 si el convertidor debe trabajar en modo motor.
- Ena2: adquirirá el valor 1 si el convertidor debe trabajar en modo freno.
- Ena3: adquirirá el valor 1 si el convertidor si se debe permitir poner los polos a 0 (abiertos).

Presentadas las variables del circuito combinacional, la tabla de verdad realizada es la que se muestra a continuación.

Nº	Swi0	Swi1	Swi2	Swi3	Swi4	Swi5	Ena0	Ena1	Ena2	Ena3
0	0	0	0	0	0	0	1	0	0	1
1	0	0	0	0	0	1	1	0	1	0
2	0	0	0	0	1	0	X	X	X	X
3	0	0	0	0	1	1	1	1	0	0
4	0	0	0	1	0	0	X	X	X	X
5	0	0	0	1	0	1	X	X	X	X
6	0	0	0	1	1	0	X	X	X	X
7	0	0	0	1	1	1	1	1	0	0
8	0	0	1	0	0	0	X	X	X	X
9	0	0	1	0	0	1	X	X	X	X
10	0	0	1	0	1	0	X	X	X	X
11	0	0	1	0	1	1	X	X	X	X
12	0	0	1	1	0	0	X	X	X	X
13	0	0	1	1	0	1	X	X	X	X
14	0	0	1	1	1	0	X	X	X	X
15	0	0	1	1	1	1	1	1	0	0
16	0	1	0	0	0	0	0	0	0	1
17	0	1	0	0	0	1	0	0	1	0
18	0	1	0	0	1	0	X	X	X	X
19	0	1	0	0	1	1	0	1	0	0
20	0	1	0	1	0	0	X	X	X	X
21	0	1	0	1	0	1	X	X	X	X

Nº	Swi0	Swi1	Swi2	Swi3	Swi4	Swi5	Ena0	Ena1	Ena2	Ena3
22	0	1	0	1	1	0	X	X	X	X
23	0	1	0	1	1	1	1	1	0	0
24	0	1	1	0	0	0	X	X	X	X
25	0	1	1	0	0	1	X	X	X	X
26	0	1	1	0	1	0	X	X	X	X
27	0	1	1	0	1	1	X	X	X	X
28	0	1	1	1	0	0	X	X	X	X
29	0	1	1	1	0	1	X	X	X	X
30	0	1	1	1	1	0	X	X	X	X
31	0	1	1	1	1	1	0	1	0	0
32	1	0	0	0	0	0	0	0	1	0
33	1	0	0	0	0	1	0	0	1	0
34	1	0	0	0	1	0	X	X	X	X
35	1	0	0	0	1	1	0	0	1	0
36	1	0	0	1	0	0	X	X	X	X
37	1	0	0	1	0	1	X	X	X	X
38	1	0	0	1	1	0	X	X	X	X
39	1	0	0	1	1	1	0	1	0	0
40	1	0	1	0	0	0	X	X	X	X
41	1	0	1	0	0	1	X	X	X	X
42	1	0	1	0	1	0	X	X	X	X
43	1	0	1	0	1	1	X	X	X	X

Nº	Swi0	Swi1	Swi2	Swi3	Swi4	Swi5	Ena0	Ena1	Ena2	Ena3
44	1	0	1	1	0	0	X	X	X	X
45	1	0	1	1	0	1	X	X	X	X
46	1	0	1	1	1	0	X	X	X	X
47	1	0	1	1	1	1	0	0	0	1
48	1	1	0	0	0	0	1	0	1	0
49	1	1	0	0	0	1	1	0	1	0
50	1	1	0	0	1	0	X	X	X	X
51	1	1	0	0	1	1	1	0	1	0
52	1	1	0	1	0	0	X	X	X	X
53	1	1	0	1	0	1	X	X	X	X
54	1	1	0	1	1	0	X	X	X	X
55	1	1	0	1	1	1	1	1	0	0
56	1	1	1	0	0	0	X	X	X	X
57	1	1	1	0	0	1	X	X	X	X
58	1	1	1	0	1	0	X	X	X	X
59	1	1	1	0	1	1	X	X	X	X
60	1	1	1	1	0	0	X	X	X	X
61	1	1	1	1	0	1	X	X	X	X
62	1	1	1	1	1	0	X	X	X	X
63	1	1	1	1	1	1	1	0	0	1

1.2 CÓDIGO VHDL

El código VHDL generado por el BooleanDeusto lo adjunto a continuación.

```
-- Generated by Boole-Deusto.  
-- Timestamp: 2017-07-10 21:45:25  
-- Computer name: fvV  
-- User name: Jose
```

```
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;  
use IEEE.STD_LOGIC_ARITH.ALL;  
use IEEE.STD_LOGIC_UNSIGNED.ALL;
```

```
entity base is
```

```
    Port (
```

```
        clk : in std_logic;
```

```
        led0 : inout std_logic;
```

```
        led1 : inout std_logic;
```

```
        led2 : inout std_logic;
```

```
        led3 : inout std_logic;
```

```
        led4 : inout std_logic;
```

```
        led5 : inout std_logic;
```

```
        led6 : inout std_logic;
```

```
        led7 : inout std_logic;
```

```
        ena0 : inout std_logic;
```

```
    ena1 : inout std_logic;
    ena2 : inout std_logic;
    ena3 : inout std_logic;

    seg0 : inout std_logic;
    seg1 : inout std_logic;
    seg2 : inout std_logic;
    seg3 : inout std_logic;
    seg4 : inout std_logic;
    seg5 : inout std_logic;
    seg6 : inout std_logic;

    dot : inout std_logic;

    but0 : in std_logic;
    but1 : in std_logic;
    but2 : in std_logic;
    but3 : in std_logic;

    swi0 : in std_logic;
    swi1 : in std_logic;
    swi2 : in std_logic;
    swi3 : in std_logic;
    swi4 : in std_logic;
    swi5 : in std_logic;
    swi6 : in std_logic;
    swi7 : in std_logic;
    swi8 : in std_logic;
    swi9 : in std_logic
  );
end base;
```

architecture behavioral of base is

begin

 ena0<=((not(swi0) and not(swi1)) or (swi0 and swi1) or (not(swi0) and not(swi2) and swi3));

 ena1<=((not(swi2) and swi3) or (not(swi0) and swi4));

 ena2<=((not(swi4) and swi5) or (swi0 and not(swi3)));

 ena3<=((not(swi0) and not(swi5)) or (swi0 and swi2));

end behavioral;