

# Pràctica 1. Sistemes Digitals

## Interfície a un teclat PS/2 amb VHDL.

Pere Palà - Alexis López

Febrer de 2016

En aquesta pràctica connectarem un teclat d'ordinador de tipus PS/2 (els que tenen un connector circular que no és USB) a una FPGA i farem que aquesta interpreti algun dels codis rebuts. A diferència de la darrera pràctica del curs passat, detectarem les pulsacions de les tecles 0..9 i A..F i les mostrarem a un display de 7 segments. Alhora, el disseny el farem amb VHDL a l'entorn Quartus II d'Altera.

Recordeu de fer l'estudi previ! Això és imprescindible per poder accedir al laboratori. Recordeu també que, com sempre, al laboratori us donarem els components necessaris i també tindreu a la vostra disposició cable rígid.

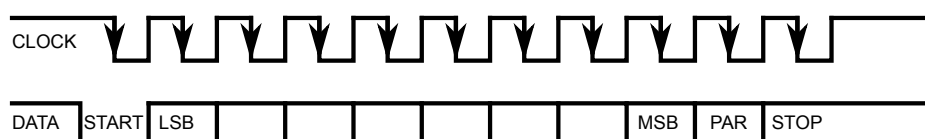
## 1 Sobre el teclat PS/2

Recordem que un teclat d'ordinador del tipus PS/2 té un connector mini-DIN de 6 pins, dels quals 4 estan cablejats:

- VCC. Típicament 5 V, però molts funcionen a 3.3 V.
- GND.
- CLK.
- DATA.

Quan es prem una tecla, el teclat envia 11 polsos de rellotge, actius per flanc descendent, i simultàniament transmet 11 bits:

- 1 bit de Start que sempre val 0.
- 8 bits de dades.
- 1 bit de paritat (completa la paritat per aconseguir que el nombre de 1 sigui senar).
- 1 bit de Stop que sempre val 1.



Els bits es transmeten amb el bit menys significatiu (LSB) en primer lloc. Cada tecla té un codi associat. Per exemple, la tecla <RETURN> té com el codi 5A (en hexadecimal), la tecla <A> el codi 1C ...

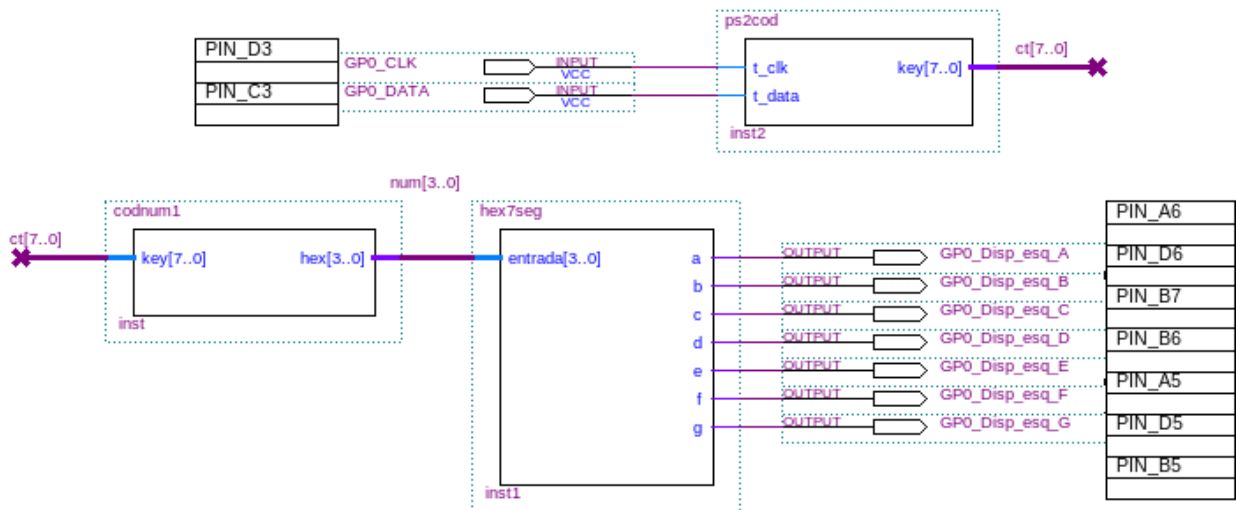
Algunes tecles, com les tecles dels cursors, envien dos o més codis seguits: <UP> envia E0 i 6B...

Quan es deixa anar una tecla normalment s'envien dos codis: F0 seguit del codi de la tecla premuda. Així, quan deixem anar la tecla <RETURN> s'envia F0 i 5A.

## 2 Descripció del muntatge

A la darrera pràctica del curs passat, ja es va fer un primer prototipus capaç de detectar la pulsació d'una tecla concreta. Ara, volem anar un xic més enllà i ser capaços de descodificar les tecles corresponents als codis hexadecimal: 0..9 i A..F.

L'estructura que proposem és la següent:



Els blocs que intervenen en aquest disseny estan implementats en VHDL. El mòdul ps2cod té el codi següent:

```

1 library ieee ;
2 use ieee.std_logic_1164.all ;
3
4 entity ps2cod is
5     port( t_clk, t_data: in std_logic;
6           key: out std_logic_vector(7 downto 0));
7 end ps2cod;
8
9 architecture arch of ps2cod is
10     signal q : std_logic_vector (10 downto 0);
11 begin
12     shift_reg : process(t_clk)
13     begin
14         if falling_edge(t_clk) then
15             q <= t_data & q(10 downto 1);
16         end if;

```

```

17     end process shift_reg;
18     key <= q(8 downto 1);
19 end arch;

```

En essència, incorpora un registre de desplaçament (línia 15) (l'operador & és un concatenador).

Després tenim el bloc anomenat `codnum1` que té per codi:

```

1  library ieee ;
2  use ieee.std_logic_1164.all ;
3
4  entity codnum1 is
5      port ( key : in std_logic_vector (7 downto 0);
6            hex : out std_logic_vector (3 downto 0));
7  end codnum1 ;
8
9  architecture num of codnum1 is
10     begin
11         with key select
12             hex <= x"0" when x"45",
13                  x"1" when x"16",
14                  x"2" when x"1E",
15                  x"3" when x"26",
16                  x"4" when x"25",
17                  x"5" when x"2E",
18                  x"6" when x"36",
19                  x"7" when x"3D",
20                  x"8" when x"3E",
21                  x"9" when x"46",
22                  x"A" when x"1C",
23                  x"B" when x"32",
24                  x"C" when x"21",
25                  x"D" when x"23",
26                  x"E" when x"24",
27                  x"F" when x"2B",
28                  "----" when others;
29 end num;

```

Essencialment aquest bloc fa una traducció del codi rebut al seu valor hexadecimal. La línia 28 fa que el resultat sigui don't care per altres valors.

Per acabar, sols cal un bloc que faci la traducció del codi hexadecimal al codi de 7 segments (de forma similar a com ho fan els integrats CD4511 que vàrem fer servir el curs passat). El codi del bloc `hex7seg`, que realitza precisament aquesta funció, és el següent:

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  entity hex7seg is
5      port(entrada : in std_logic_vector(3 downto 0);
6            a,b,c,d,e,f,g : out std_logic);
7  end hex7seg;
8
9  architecture arc of hex7seg is
10     signal bus_out : std_logic_vector(6 downto 0);
11 begin
12     with entrada select
13
14         bus_out <= "1111110" when x"0",

```

```

15         "0110000" when x"1",
16         "???????" when x"2",
17         "???????" when x"3",
18         "???????" when x"4",
19         "???????" when x"5",
20         "???????" when x"6",
21         "???????" when x"7",
22         "???????" when x"8",
23         "???????" when x"9",
24         "???????" when x"A",
25         "???????" when x"B",
26         "???????" when x"C",
27         "???????" when x"D",
28         "???????" when x"E",
29         "???????" when x"F",
30         "-----" when others;
31     (a,b,c,d,e,f,g) <= bus_out;
32 end arc;

```

*Previ 1.* Haureu observat que a les línies 16 a 29 hi apareixen uns interrogants. Es tracta de que esbrineu els valors que calen per aconseguir l'objectiu perseguit. Cerqueu la informació que us calgui i tingueu en compte que els noms dels ports s'han triat de forma racional.

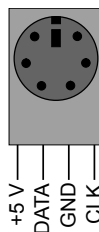
*Previ 2.* Algú podria pensar que els dos darrers blocs es podrien combinar en un bloc únic. S'ha triat fer-ho de forma separada per millorar la claredat del disseny. D'altra banda, durant el procés de síntesi, l'optimitzador ja s'encarregarà de trobar la forma més simplificada de realitzar la funció final.

Malgrat aquests comentaris, proposeu un únic bloc VHDL que substitueixi la funcionalitat dels dos darrers blocs.

### 3 Experimentant

Es tracta de reproduir la funcionalitat que s'ha proposat en aquest document. Heu d'aconseguir que el display de 7 segments mostri la tecla premuda.

*Tasca 1.* Implementeu el disseny general del *previ 1*. És a dir, el codi donat a l'enunciat de la pràctica però amb els interrogants substituïts pels valors corresponents. Connecteu la placa que us proporcionem juntament amb la placa de desenvolupament amb la FPGA. El connector PS/2 pel teclat es reproduïx a l'imatge que hi ha a continuació.



*Tasca 2.* Implementeu el disseny més compacte que heu aconseguit en el *previ 2* amb el vostre codi per al decodificador de 7 segments.

*Previ 3.* Cada vegada que manipulem una tecla es genera un missatge quan la pitgem i un altre quan la deixem anar. Al final del procés veurem el codi associat al darrer missatge enviat que coincidirà amb la tecla manipulada, però entremig hi haurà un parpelleig degut a que també s'envia un codi F0.

Penseu una manera d'evitar aquest fenomen.

*Tasca 3.* Experimenteu la solució proposada al problema del parpelleig.