Le ragazze e la scienza: Incoraggiamole ad intraprendere L'avventura

Laboratorio

Pierluigi Luciano

PhD student at UNICLAM
Fellow at Department of Physics
Associated with INFN

1° Esercizio: La Porta OR

Porta logica OR $Y = A+B$

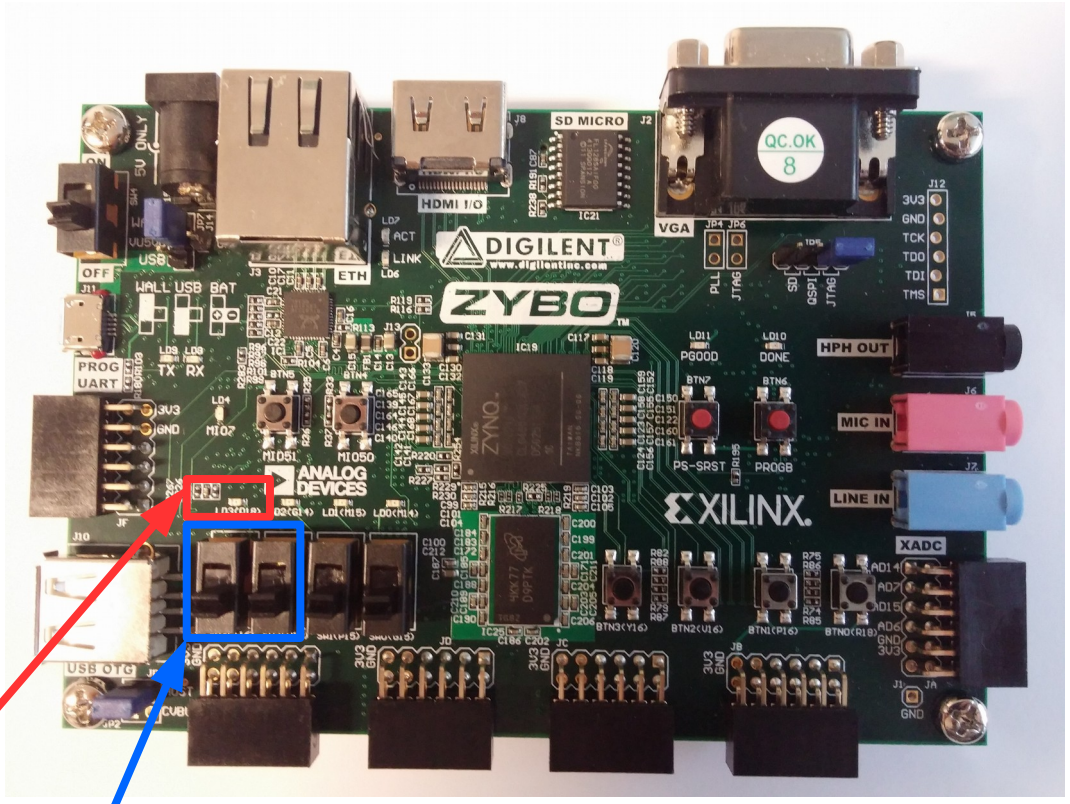
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1


```
input <= A & B;

porta_or_proc : process(input)

   begin
      case input is
         when "00" =>
            Y <= '0';
         when "01" =>
            Y <= '1';
         when "10" =>
            Y <= '1';
         when "11" =>
            Y <= '1';
         when others =>
            Y <= '0';
      end case;
end process;
```

```
entity porta_or is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           Y : out STD_LOGIC);
end porta_or;
```



2° Esercizio:
La Porta AND

Porta logica AND $Y = A \cdot B$

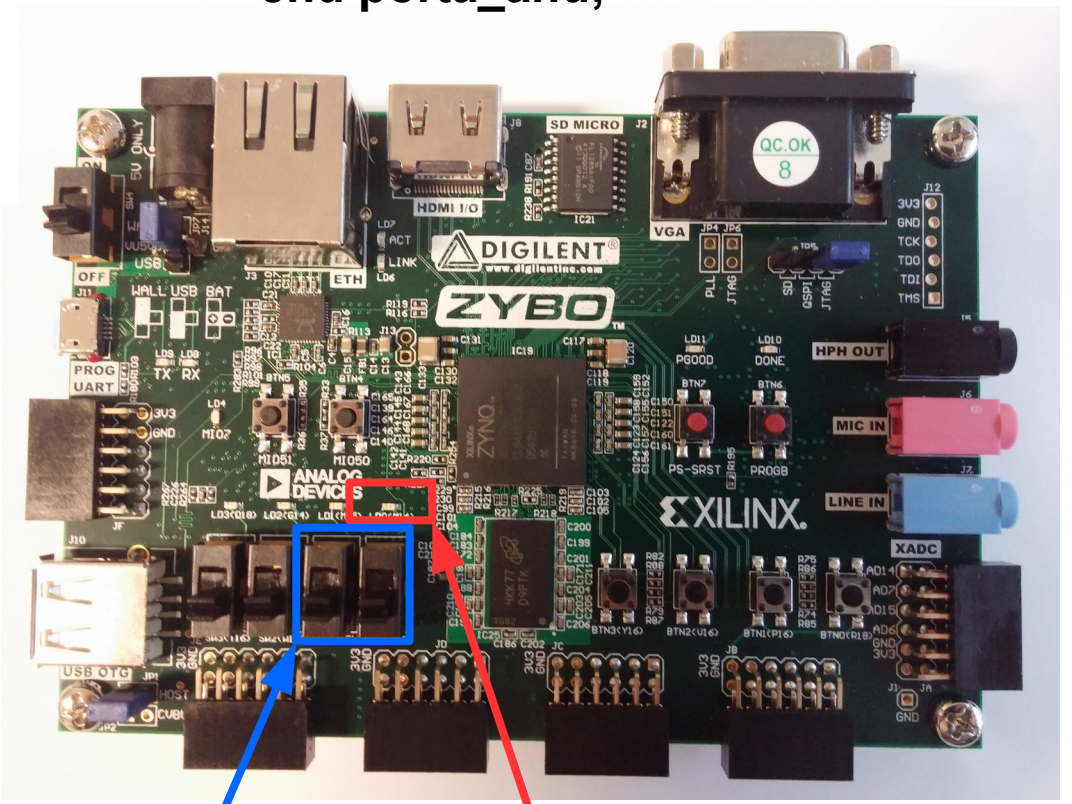
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1


```
Input <= A & B;

porta_and_proc : process(input)

   begin
      case input is
         when "00" =>
            Y <= '0';
         when "01" =>
            Y <= '0';
         when "10" =>
            Y <= '0';
         when "11" =>
            Y <= '1';
         when others =>
            Y <= '0';
      end case;
end process;
```

```
entity porta_and is
   Port ( A : in STD_LOGIC;
          B : in STD_LOGIC;
          Y : out STD_LOGIC);
end porta_and;
```



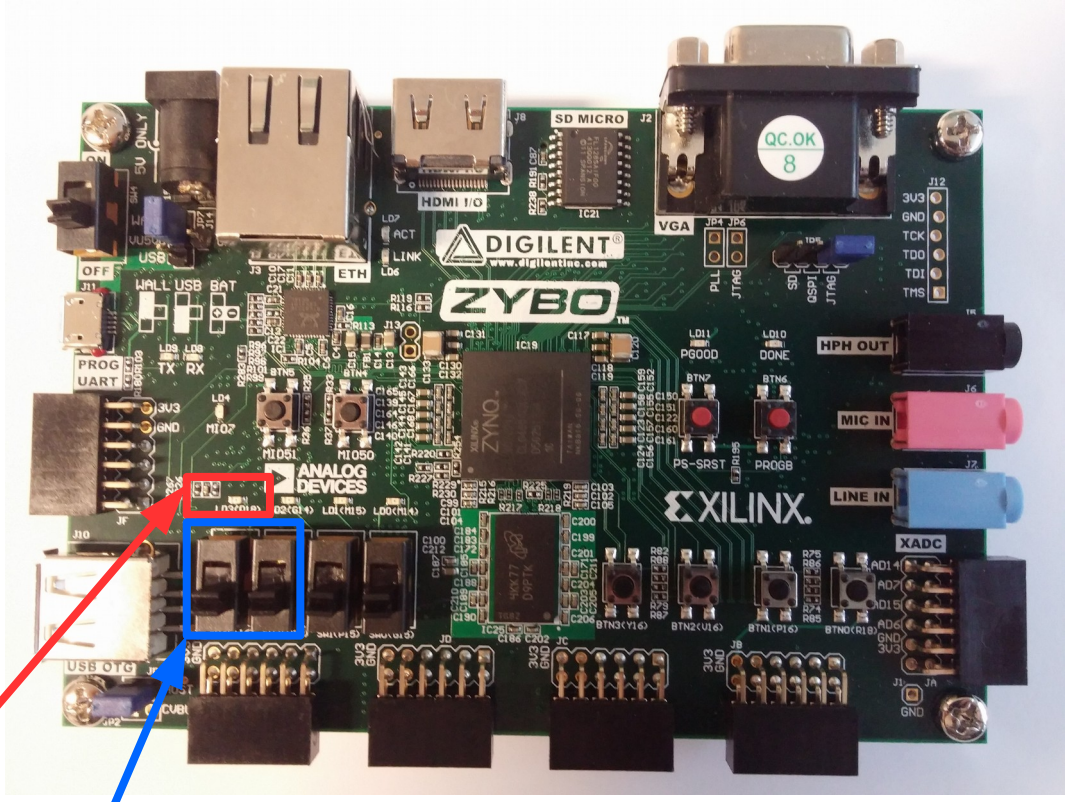
Esercizio Interattivo: La Porta XOR

Porta logica XOR $Y = A \oplus B$

A	B	Y
0	0	
0	1	
1	0	
1	1	


```
entity porta_xor is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           Y : out STD_LOGIC);
end porta_xor;
```

L'uscita è '1' quando in ingresso è presente un numero di '1' dispari!



Il Bignami del VHDL

Il VHDL mette a disposizione i costrutti OR e AND (vero? Che geni!!) per implementare un porta AND o OR!!

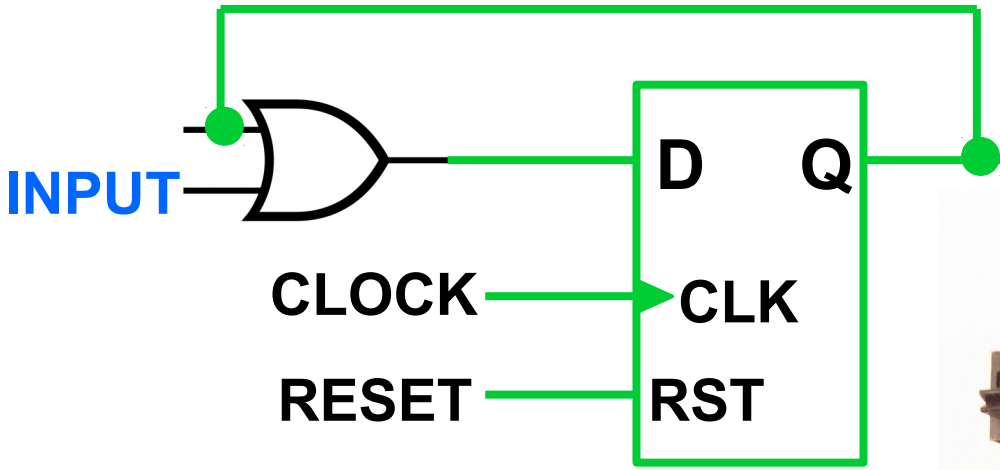
Y <= A and B;

Y <= A or B;

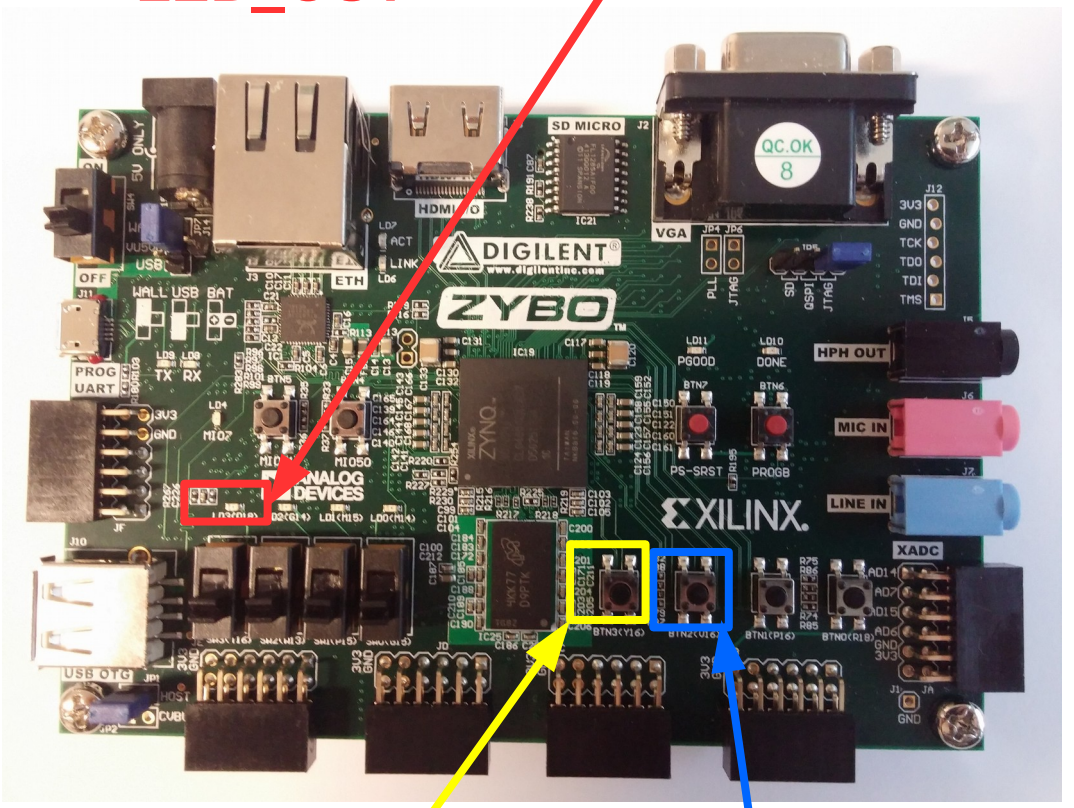
Y <= A xor B;

Ma sarebbe stato troppo facile!!!

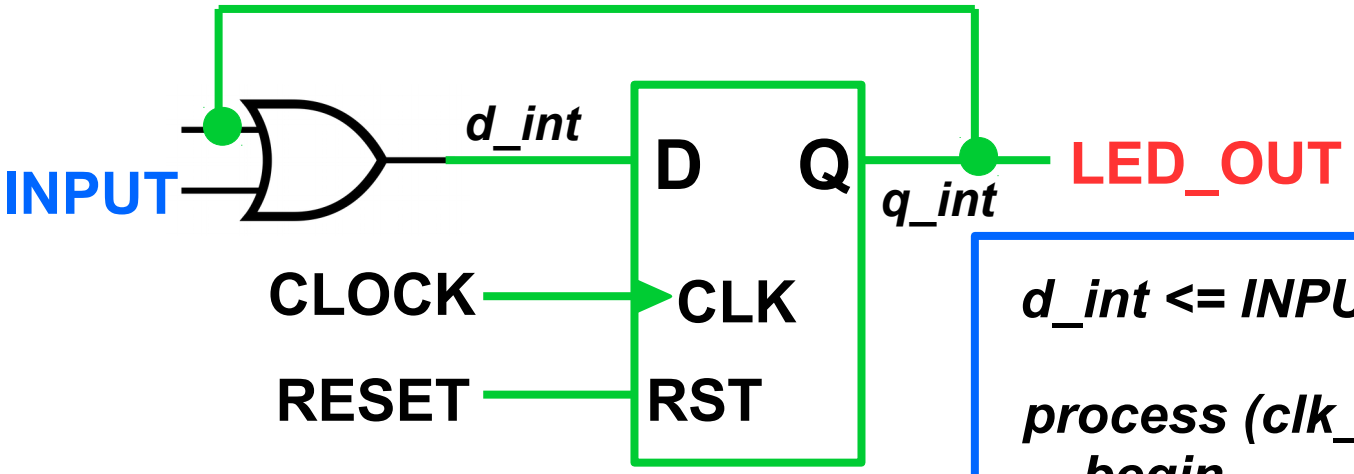
3° Esercizio:
Elemento di Memoria




```
entity elemento_di_memoria is
    Port (
        INPUT : in STD_LOGIC;
        RESET : in STD_LOGIC;
        CLOCK : in STD_LOGIC;
        LED_OUT : out
STD_LOGIC
    );
end elemento_di_memoria;
```



3° Esercizio: Elemento di Memoria



Semplice esempio che dimostra che il linguaggio VHDL è un linguaggio di descrizione hardware e NON un linguaggio di programmazione

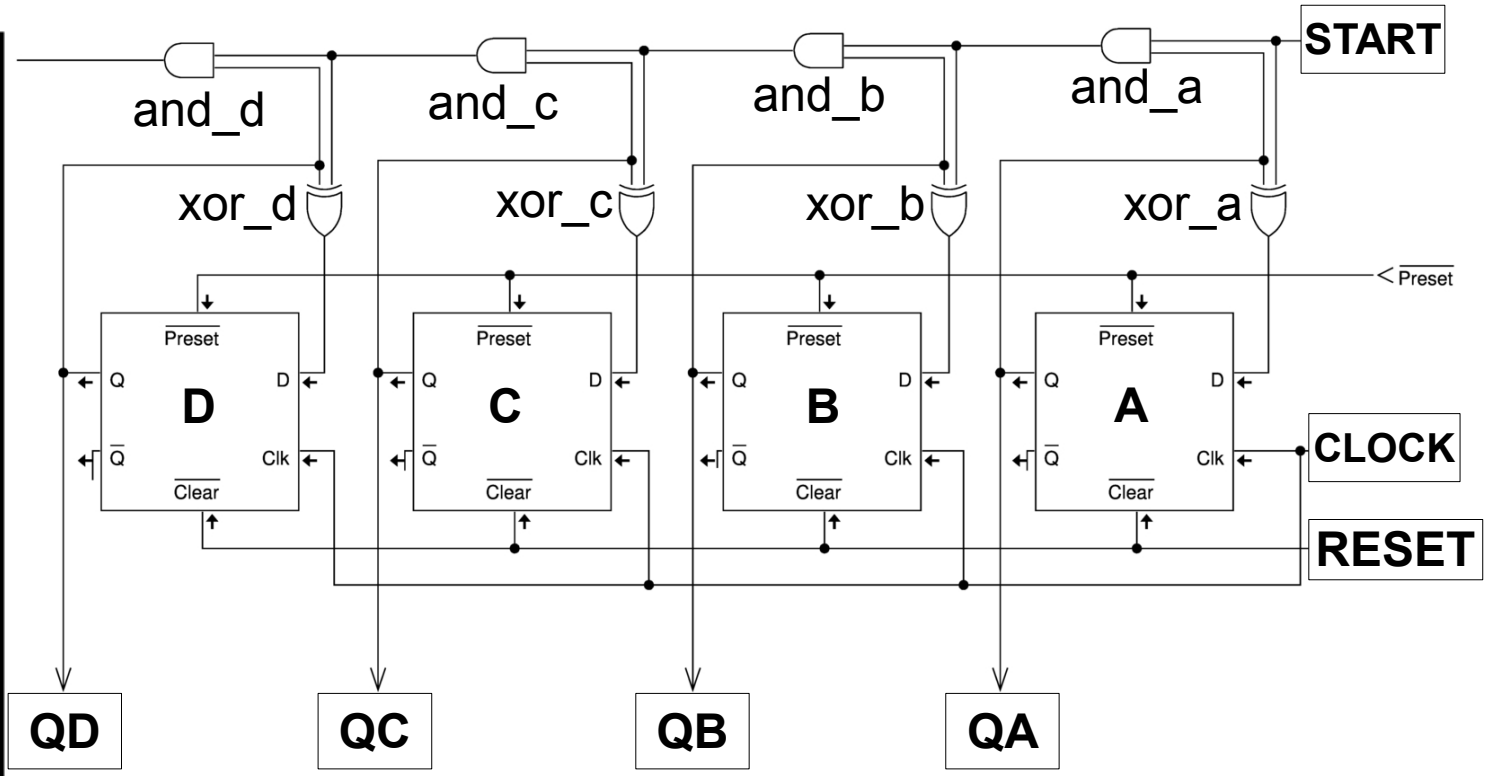
```
d_int <= INPUT or q_int;

process (clk_int)
    begin
        if clk_int'event and clk_int='1' then
            if RESET='1' then
                q_int <= '0';
            else
                q_int <= d_int;
            end if;
        end if;
end process;

LED_OUT <= q_int;
```

4° Esercizio: Contatore 4-bits

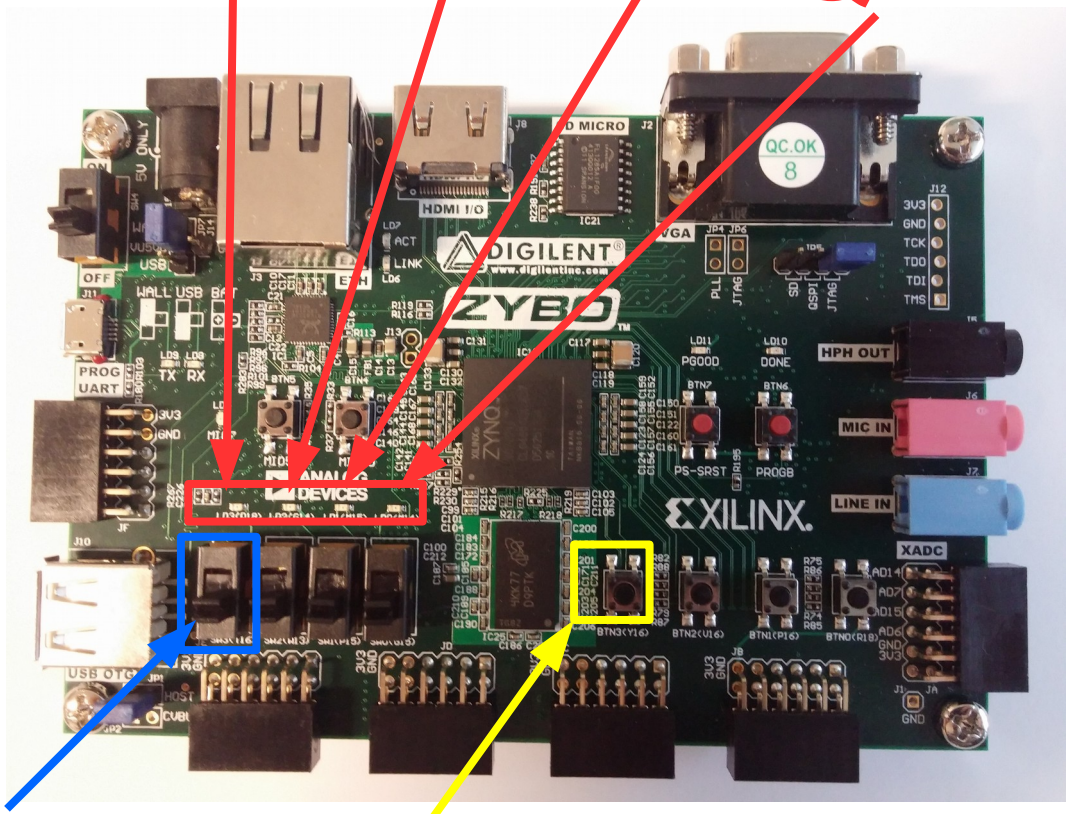
Count	Outputs			
	Q_D	Q_C	Q_B	Q_A
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	L	H	H	L
7	L	H	H	H
8	H	L	L	L
9	H	L	L	H
10	H	L	H	L
11	H	L	H	H
12	H	H	L	L
13	H	H	L	H
14	H	H	H	L
15	H	H	H	H



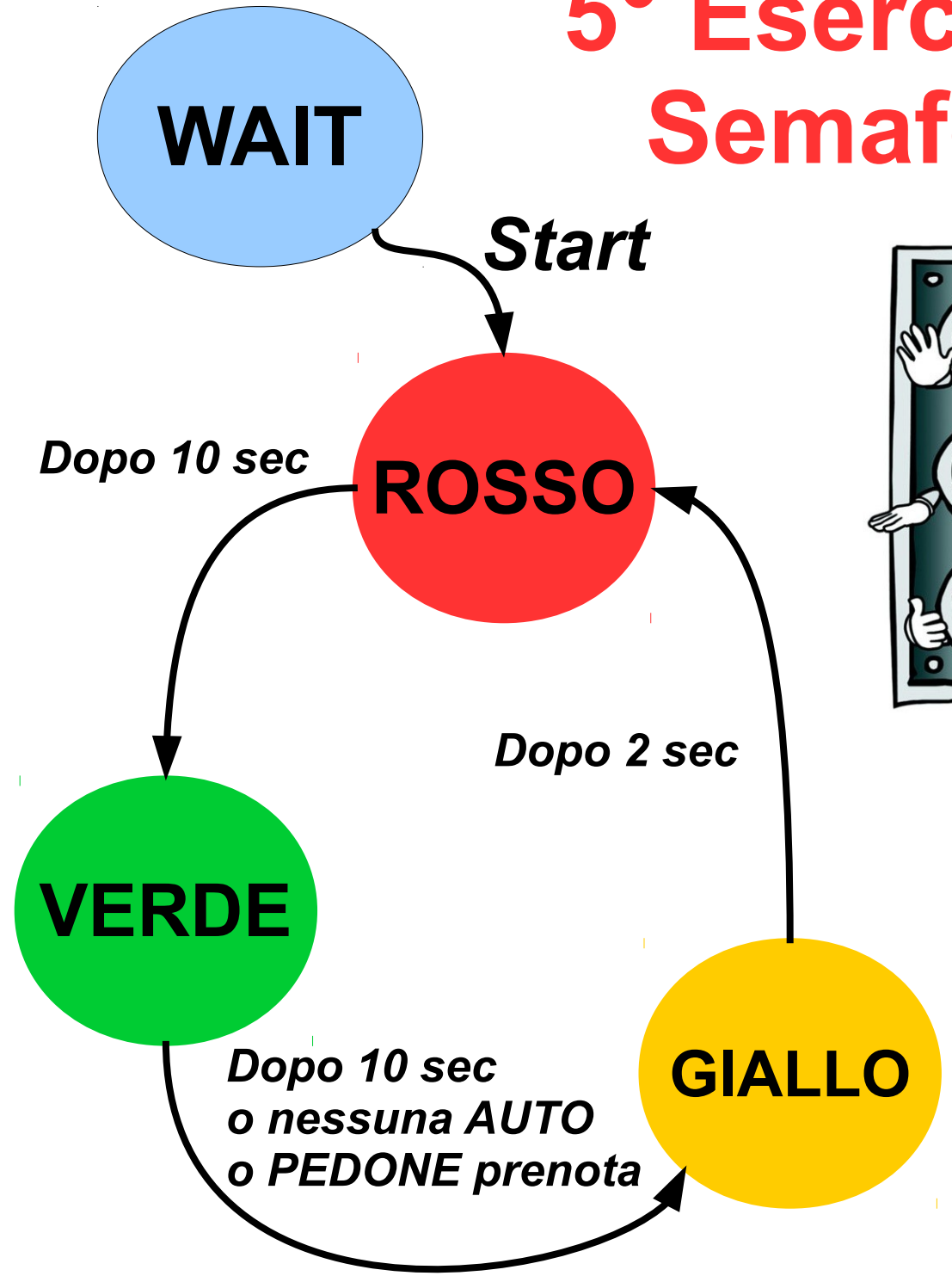
Combinando il "Flip-Flop D", la porta "AND" e la porta "XOR" possiamo realizzare un "CONTATORE 4-bits" per contare da 0 a 15 (in decimale)

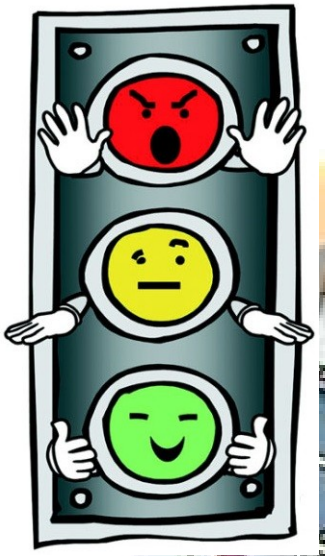
4° Esercizio:
Contatore 4-bits

```
entity contatore is
    Port (
       CLOCK : in STD_LOGIC;
       RESET  : in STD_LOGIC;
       START  : in STD_LOGIC;
       QD       : out STD_LOGIC;
       QC       : out STD_LOGIC;
       QB       : out STD_LOGIC;
       QA       : out STD_LOGIC
   );
end contatore;
```



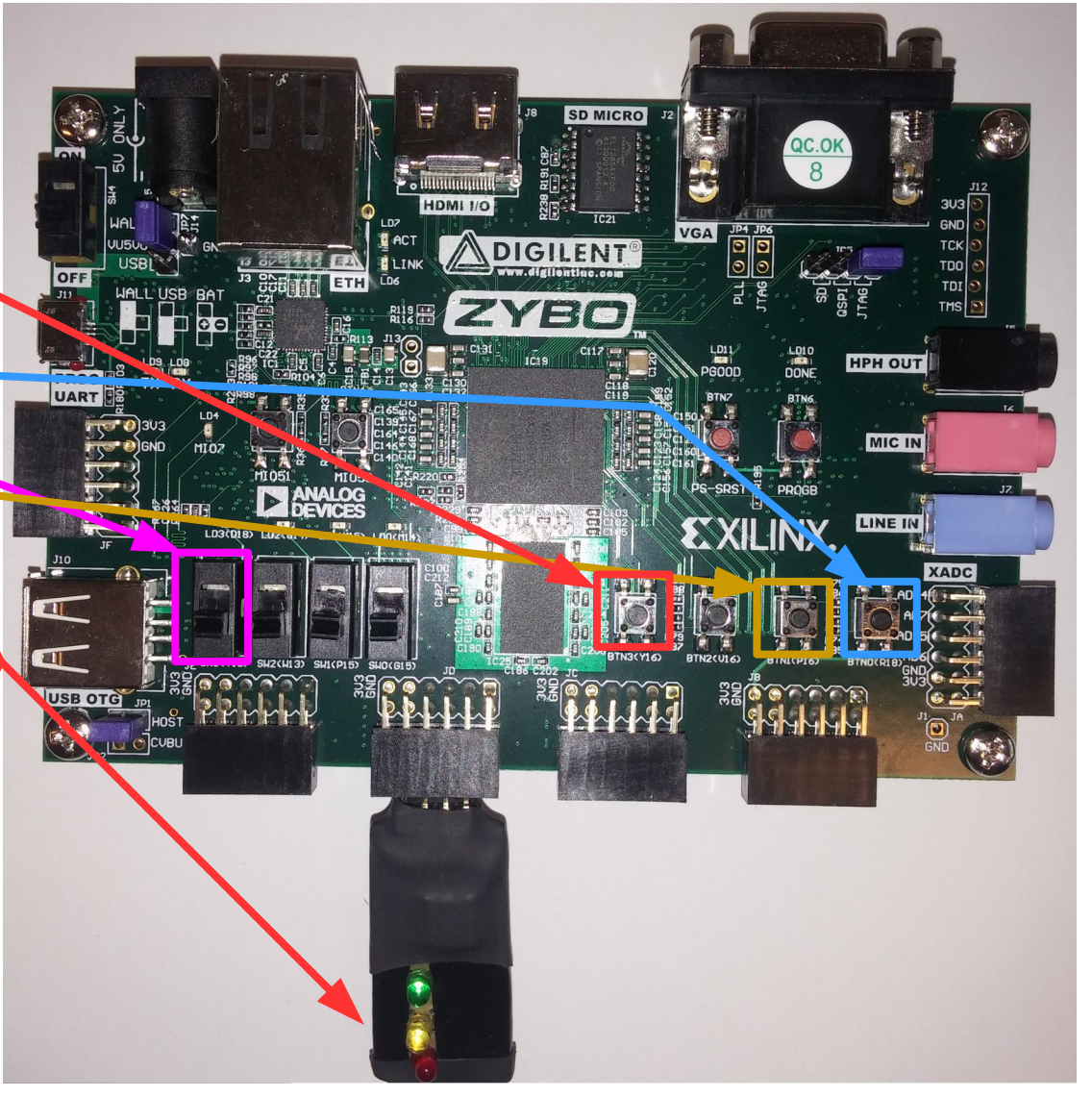
5° Esercizio: Semaforo





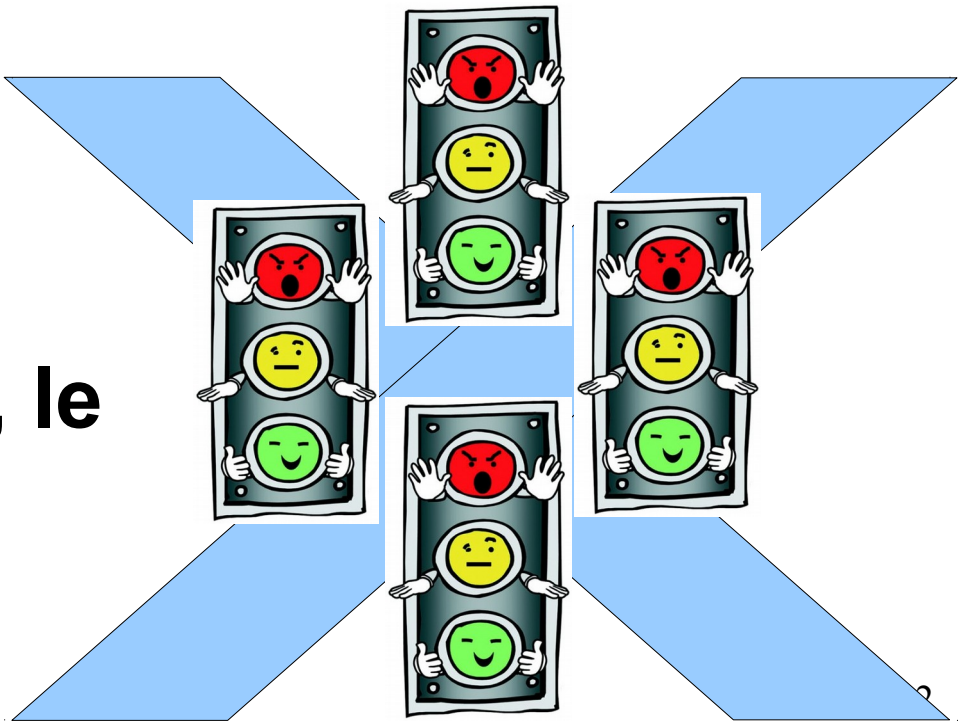
5° Esercizio: Semaforo

```
entity semaforo is
    Port (
        RESET   : in STD_LOGIC;
        CLOCK   : in STD_LOGIC;
        START   : in STD_LOGIC;
        AUTO    : in STD_LOGIC;
        PEDONE  : in STD_LOGIC;
        LED_G   : out STD_LOGIC;
        LED_Y   : out STD_LOGIC;
        LED_R   : out STD_LOGIC);
end semaforo;
```



Incrocio

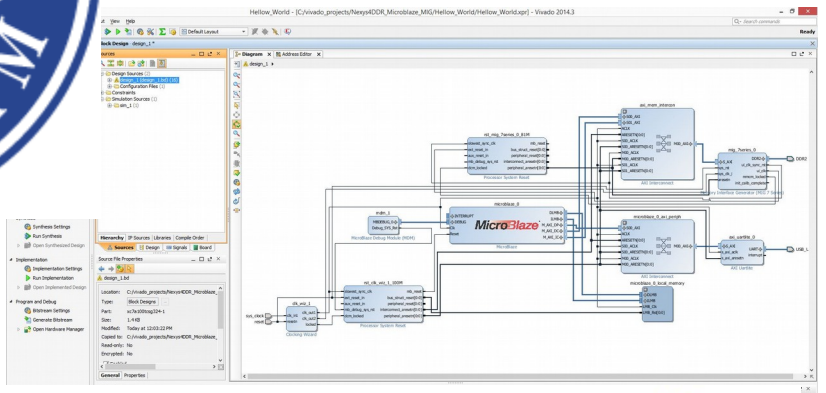
L'FPGA può diventare il controllore di un intero incrocio, gestendo tutti i semafori, le prenotazioni da parte dei pedoni, segnalazioni acustiche, ect ect



Come reperire il materiale

Esiste un modo GRATUITO E LEGALE per reperire sia il software che l'hardware che avete visto.

XILINX UNIVERSITY PROGRAM



DIGILENT®
BEYOND THEORY

VIVADO™