

Esercizi su FPGA



UNIVERSITÀ DEGLI STUDI
DI PERUGIA



- 1** Setup del Laboratorio
 - Accesso ed Ambiente
 - Evaluation Board
 - Software
- 2** Circuiti Combinatori
 - Porte Logiche
- 3** I/O
- 4** Esercizi completi



UNIVERSITÀ DEGLI STUDI
DI PERUGIA

Setup del Laboratorio

Mirko Mariotti



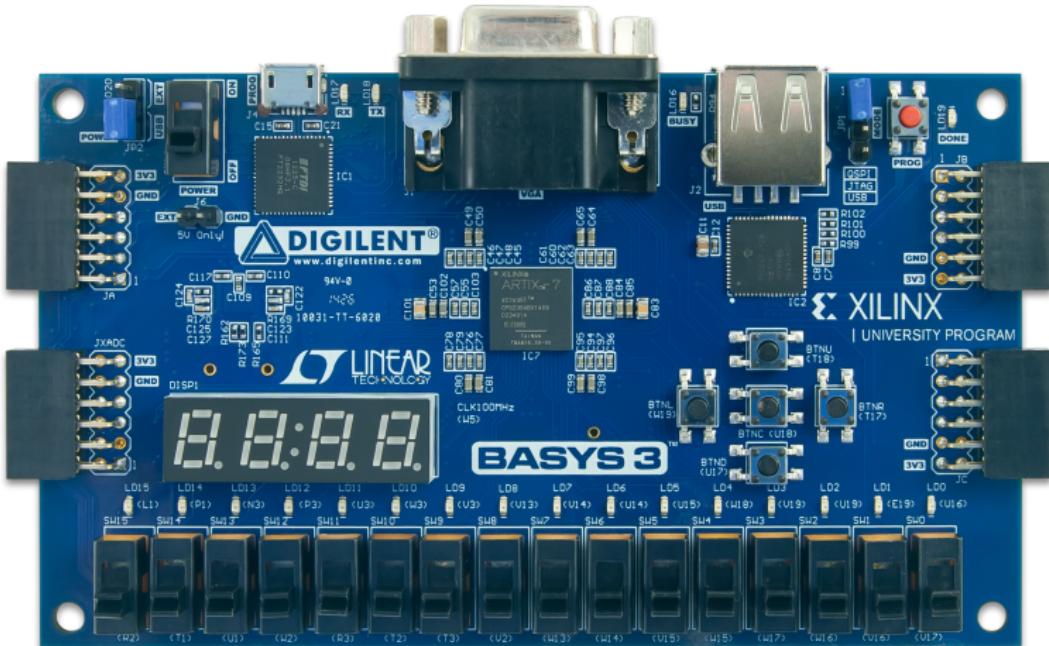
- Al laboratorio si può accedere con le credenziali di Ateneo (se autorizzati).
- Per questi esercizi si possono usare gli account lab2userXY.
- Il sistema è Linux con una distribuzione Debian 7 (Wheezy) e Gnome classico con Desktop Environment.



- Al laboratorio si può accedere con le credenziali di Ateneo (se autorizzati).
- Per questi esercizi si possono usare gli account lab2userXY.
- Il sistema è Linux con una distribuzione Debian 7 (Wheezy) e Gnome classico con Desktop Environment.



- Al laboratorio si può accedere con le credenziali di Ateneo (se autorizzati).
- Per questi esercizi si possono usare gli account lab2userXY.
- Il sistema è Linux con una distribuzione Debian 7 (Wheezy) e Gnome classico con Desktop Environment.

Evaluation Board
Basys3

Caratteristiche:

- Marca: Digilent
- Chip: Xilinx Artix 7 - XC7A35T-1CPG236C
- Celle: 33280
- Frequenza: 450 MHz
- 16 user switches
- 16 user LEDs
- Part number: **xc7a35ticpg236-1L**
- altre ...



Il software utilizzato per la programmazione dei device Xilinx è Vivado.



UNIVERSITÀ DEGLI STUDI
DI PERUGIA

Circuiti Combinatori

Mirko Mariotti



Sono circuiti il cui funzionamento riguarda solo la relazione ingresso-uscita.



Obiettivi dell'esercizio:

- Creazione di un progetto con Vivavo.
- Inserimento di un modulo Verilog per realizzare la porta logica Not.
- Sintesi del circuito.
- Verifica dello schematico risultante.
- Verifica dei valori di verità.



Obiettivi dell'esercizio:

- Creazione di un progetto con Vivavo.
- Inserimento di un modulo Verilog per realizzare la porta logica Not.
- Sintesi del circuito.
- Verifica dello schematico risultante.
- Verifica dei valori di verità.



Obiettivi dell'esercizio:

- Creazione di un progetto con Vivavo.
- Inserimento di un modulo Verilog per realizzare la porta logica Not.
- Sintesi del circuito.
- Verifica dello schematico risultante.
- Verifica dei valori di verità.



Obiettivi dell'esercizio:

- Creazione di un progetto con Vivavo.
- Inserimento di un modulo Verilog per realizzare la porta logica Not.
- Sintesi del circuito.
- Verifica dello schematico risultante.
- Verifica dei valori di verità.

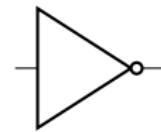


Obiettivi dell'esercizio:

- Creazione di un progetto con Vivavo.
- Inserimento di un modulo Verilog per realizzare la porta logica Not.
- Sintesi del circuito.
- Verifica dello schematico risultante.
- Verifica dei valori di verità.



- Simboli
- Tavola di verità
- Codice verilog





- Simboli
- Tavola di verità
- Codice verilog

Table: Tavola Verità

A	B
0	1
1	0



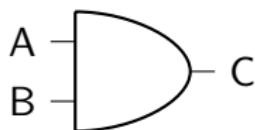
- Simboli
- Tavola di verità
- Codice verilog

```
module  notm (
    input wire A,
    output wire B
);

    assign B = ~ A;

endmodule
```

Provare a ripetere l'esercizio con una porta And:



```
module andm (
    input wire A,
    input wire B,
    output wire C
);
    assign C = A & B;
endmodule
```

Provare a ripetere l'esercizio con una porta Or:



```
module orm (
    input wire A,
    input wire B,
    output wire C
);

    assign C = A | B;

endmodule
```

Vedere cosa succede con reti piu' complesse.

```
module complex(
    input wire A,
    input wire B,
    input wire C,
    output wire D,
    output wire E
);

    assign D = A & (B | C);
    assign E = B | C;
endmodule
```

Vedere cosa succede con reti piu' complesse.

```
module complex(
    input wire A,
    input wire B,
    input wire C,
    input wire D,
    input wire E,
    input wire F,
    input wire G,
    output wire H
);

    assign H = A & ((B | C) & (!D & (F | !G)) | E);
endmodule
```



UNIVERSITÀ DEGLI STUDI
DI PERUGIA

I/O

Mirko Mariotti

- Il file di constraints mappa gli input-output effettivi agli elementi in verilog.
- Un esempio è fornito dal produttore della evaluation board.
- Va incluso nel progetto.

Obbiettivi dell'esercizio:

- Importare il clock dentro il progetto come input.
- Importare un led dentro il progetto come output.
- Far lampeggiare il led.

```
module blink(
    input  clk,
    output reg [7:0] led
);

/* always */
always @ (posedge clk) begin
    led[0] <= clk;
end

endmodule
```

Obbiettivi dell'esercizio:

- Importare il clock dentro il progetto come input.
- Importare un led dentro il progetto come output.
- Far lampeggiare il led.

```
module blink(  
    input  clk,  
    output reg [7:0] led  
>;  
  
/* always */  
always @ (posedge clk) begin  
    led[0] <= clk;  
end  
  
endmodule
```



Obbiettivi dell'esercizio:

- Importare il clock dentro il progetto come input.
- Importare un led dentro il progetto come output.
- Far lampeggiare il led.

```
module blink(
    input  clk,
    output reg [7:0] led
);

/* always */
always @ (posedge clk) begin
    led[0] <= clk;
end

endmodule
```

Obbiettivi dell'esercizio:

- Importare il clock dentro il progetto come input.
- Importare un led dentro il progetto come output.
- Far lampeggiare il led.
-

```
module blink(
    input  clk,
    output reg [7:0] led
);

/* always */
always @ (posedge clk) begin
    led[0] <= clk;
end

endmodule
```

Far lampeggiare il LED0 con una frequenza di circa 2 Hz.

```
module counter(
    input  clk,
    output reg [7:0] led
);

initial
    counter <= 0;

reg [32:0] counter;

/* always */
always @ (posedge clk) begin
    led[0] <= counter[23];

    counter <= counter + 1;
end

endmodule
```



UNIVERSITÀ DEGLI STUDI
DI PERUGIA

Esercizi completi

Mirko Mariotti

- Utilizzare lo switch0 per pilotare il LED0 (lampeggia se lo switch è attivo altrimenti no)
- Regolare la frequenza con switch 1 (se è attivo raddoppiare la frequenza)
- Realizzare coi led da 1 a 8 un counter binario.
- Partendo con uno dei 16 led acceso e gli altri spenti, fare in modo che premendo il tasto destra il led acceso si sposti a destra. Stessa cosa a sinistra.