Reti Logiche

Laboratorio #2
Modulatore PWM

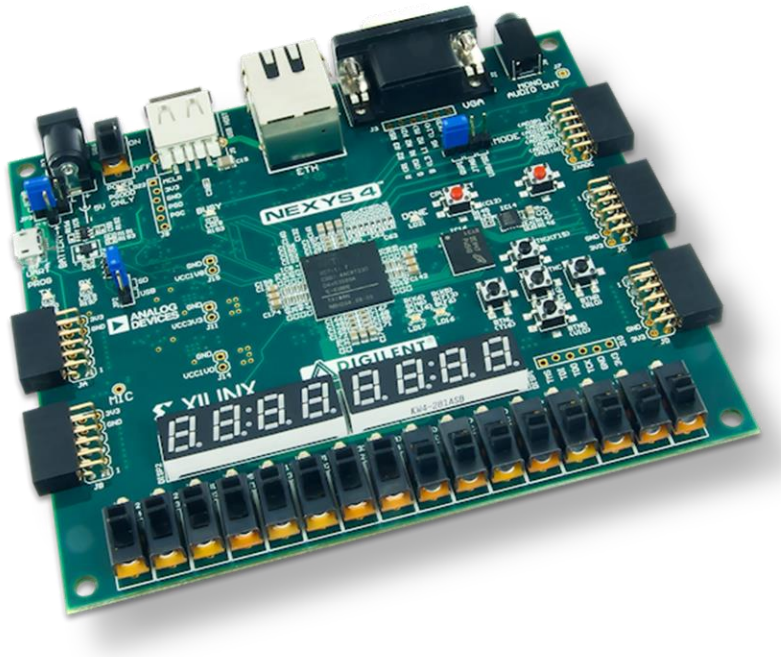
Enrico Manuzzato

Codice FPGA per Vivado

- Chi ha la NEXYS4-DDR: xc7a100tcsg324-1
- Chi ha la ZedBoard : xc7z020clg484-1

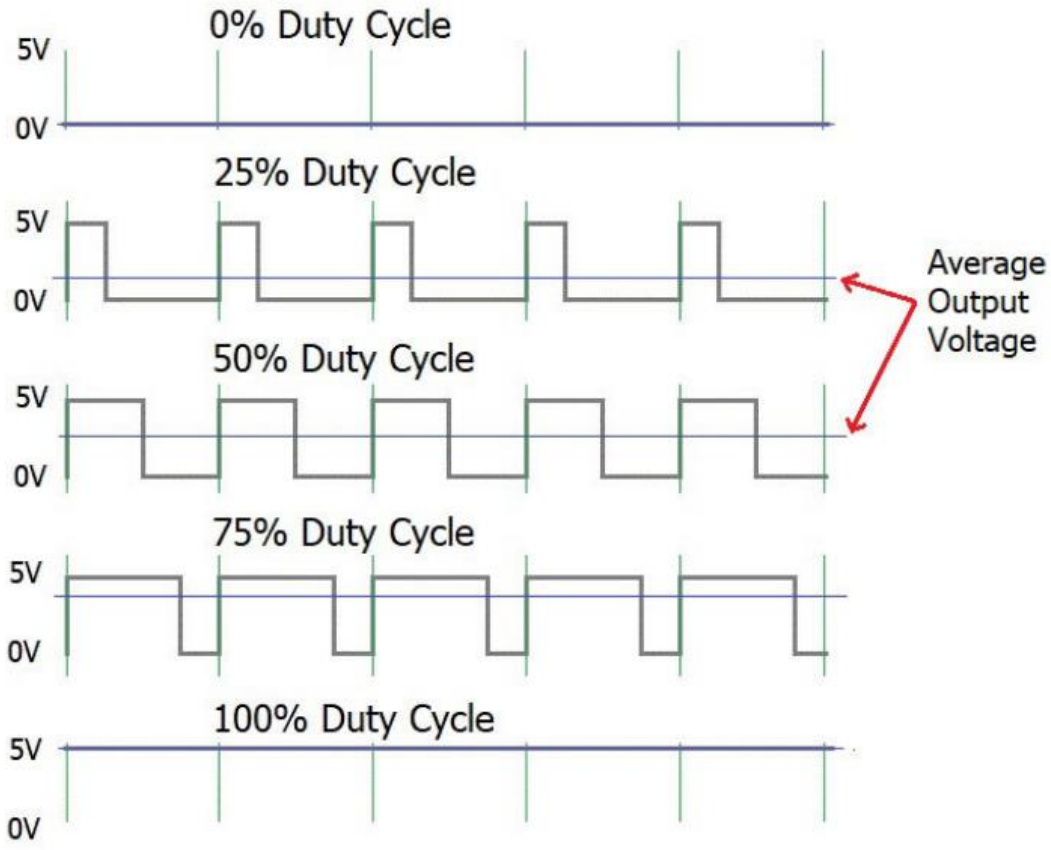
Descrizione

- Implementerete un controller per gestire degli effetti di luce
- Il blocco di base è il modulatore PWM che controlla il livello di luminosità dei LED
- Poi ci sarà un blocco che gestisce la variazione di luminosità dei LED, per creare effetti di luce



Modulazione PWM

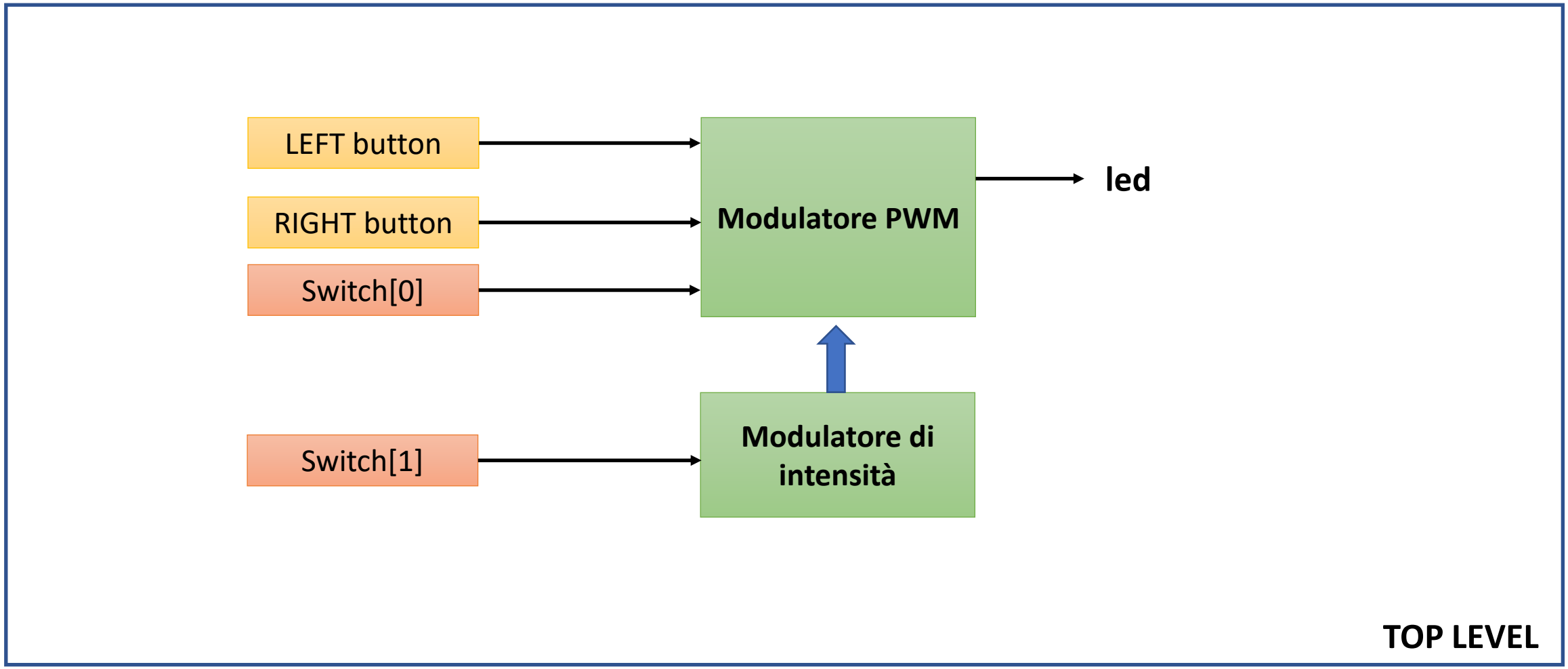
- Serve per poter controllare in modo digitale il livello medio di tensione fornito a un carico (es: motori elettrici, LED)
- Per aumentare/diminuire la tensione media fornita al carico, viene fatto variare il duty cycle (e non il livello)



Modulatore PWM

- Bisogna quindi generare un segnale a onda quadra il cui duty cycle deve poter essere cambiato da 0% (LED spento) al 100% (LED acceso alla massima luminosità)
- Come si fa?
- Quanti livelli di luminosità vogliamo?

Schema a blocchi completo



Descrizione del funzionamento

- LEFT button serve per **incrementare manualmente il livello di luminosità dei LED** (ad ogni pressione si incrementa di +1, quando raggiunge il massimo riparte da 0)
- RIGHT button serve per **decrementare manualmente il livello di luminosità dei LED** (ad ogni pressione si decrementa di -1, quando raggiunge il minimo riparte da 127)
- Switch[0] serve per selezionare tra **luminosità manuale** (LEFT/RIGHT button) oppure luminosità "**automatica**" (generata dal blocco di modulazione intensità)
- Switch[1] serve per selezionare tra due diversi **pattern di variazione di luminosità** (a dente di sega oppure triangolare)

Entità base

- Dovrete descrivere e simulare le seguenti entità:
  - Modulatore PWM
  - Modulatore di intensità
- Il VHDL per i bottoni con debouncing vi viene già fornito. Il display è opzionale, vi potrebbe tornare utile per visualizzare il valore corrente di luminosità di un LED.
- Utilizzare il template fornito

Modulatore PWM

- Possiamo usare un contatore che conta da 0 a 127 e poi riparte da 0 (free running).
- Il duty cycle si imposta confrontando il contatore con un valore di soglia.
- Il valore di soglia lo scelgo anch'esso a 7 bit --> in questo modo ho 128 possibili livelli di luminosità.

```
if (counter < soglia) then
        LED <= '1';
else
        LED <= '0';
end if;
```

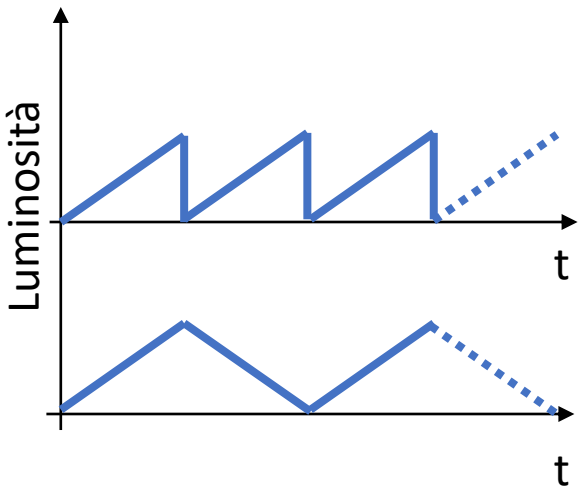
- Per valori alti di soglia: la condizione *counter < soglia* sarà soddisfatta per più tempo => duty cycle alto
- Per valori bassi di soglia: la condizione *counter < soglia* sarà soddisfatta per meno tempo => duty cycle basso

Modulatore PWM

- Per variare la luminosità del LED, potete incrementare il valore di soglia ogni volta che viene premuto un bottone (ad ogni click: *soglia <= soglia + ‘1’;)*
- Usate il VHDL del bottone fornito (button.vhd)
- **Poi, prevedete anche un input a 7 bit per fornire il valore di soglia dall’esterno (servirà poi con il modulatore di intensità per gestire gli effetti luminosi).**

Modulatore di intensità

- Serve per creare un certo pattern di luminosità
- Ci sono tante possibilità, quelle proposte sono le seguenti:
  - **Pattern a dente di sega**: la luminosità del LED aumenta linearmente da 0 fino al massimo (127). Poi riparte da 0.
  - **Pattern triangolare**: la luminosità del LED aumenta linearmente da 0 fino al massimo, poi diminuisce gradualmente fino a 0.
- Perché gli effetti siano apprezzabili dall'occhio umano, la luminosità deve variare **lentamente** (ad esempio, dato che abbiamo un totale di 128 possibili livelli, si può incrementare/decrementare di 1 ogni 1/128 di secondo, in modo da coprire tutti i valori nell'arco di un secondo)



Modulatore di intensità

- Come si fa?
- Una realizzazione semplice può essere tramite macchina a stati:
  - Uno stato per il pattern a dente di sega (dove banalmente si fa contare un contatore)
  - Uno stato per la rampa 0->max del pattern triangolare (dove si conta da 0 a 127)
  - Uno stato per la rampa max->0 del pattern triangolare (dove si conta da 127 a 0)

- L'output di questo blocco (il valore di luminosità a 7 bit) andrà poi connesso all'input per il valore di soglia del modulatore PWM!