

# Reti Logiche

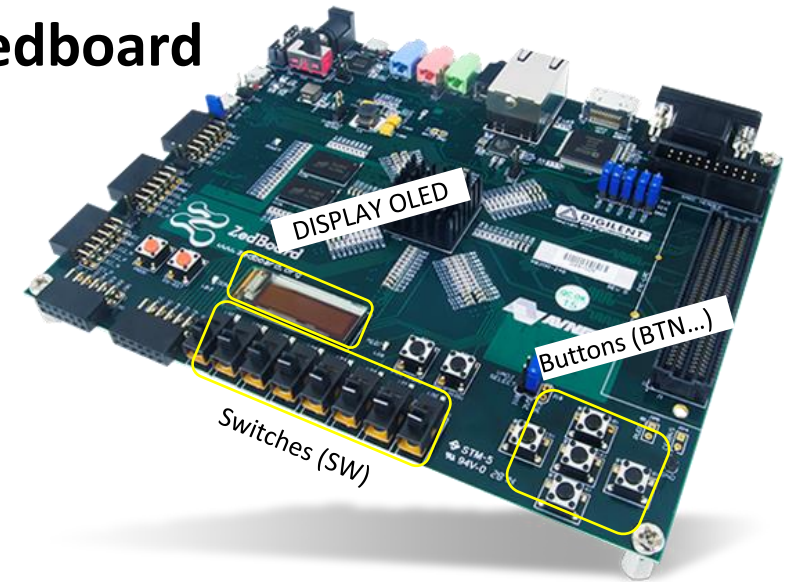
## Laboratorio #1 Calcolatrice

Enrico Manuzzato

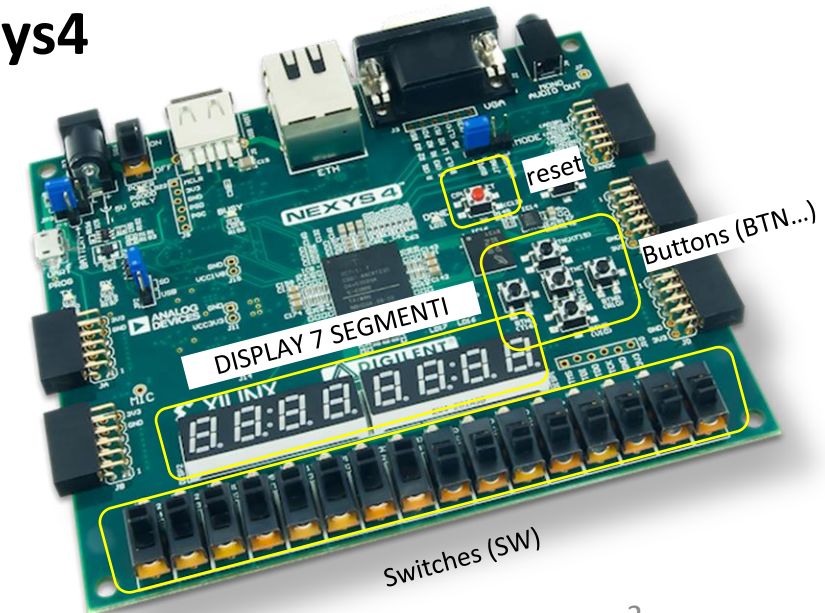
# Descrizione

- Progettare una semplice calcolatrice con accumulatore
  - Implementare le seguenti operazioni :  
addizione, sottrazione e moltiplicazione  
eseguite da una Arithmetic Logic Unit (ALU)
  - Il primo operando viene inserito tramite gli  
switch sulla board in codice binario
  - Il secondo operando è il risultato  
dell'operazione precedente ed è interno alla  
FPGA, memorizzato da un accumulatore  
(all'inizio è zero)
  - Il risultato viene visualizzato sul display (a 7  
segmenti sulla NEXYS4, OLED sulla ZedBoard)

Zedboard

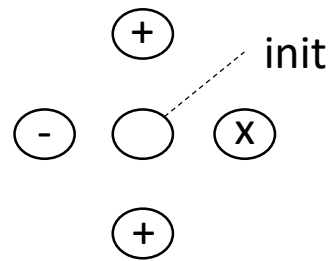


Nexys4

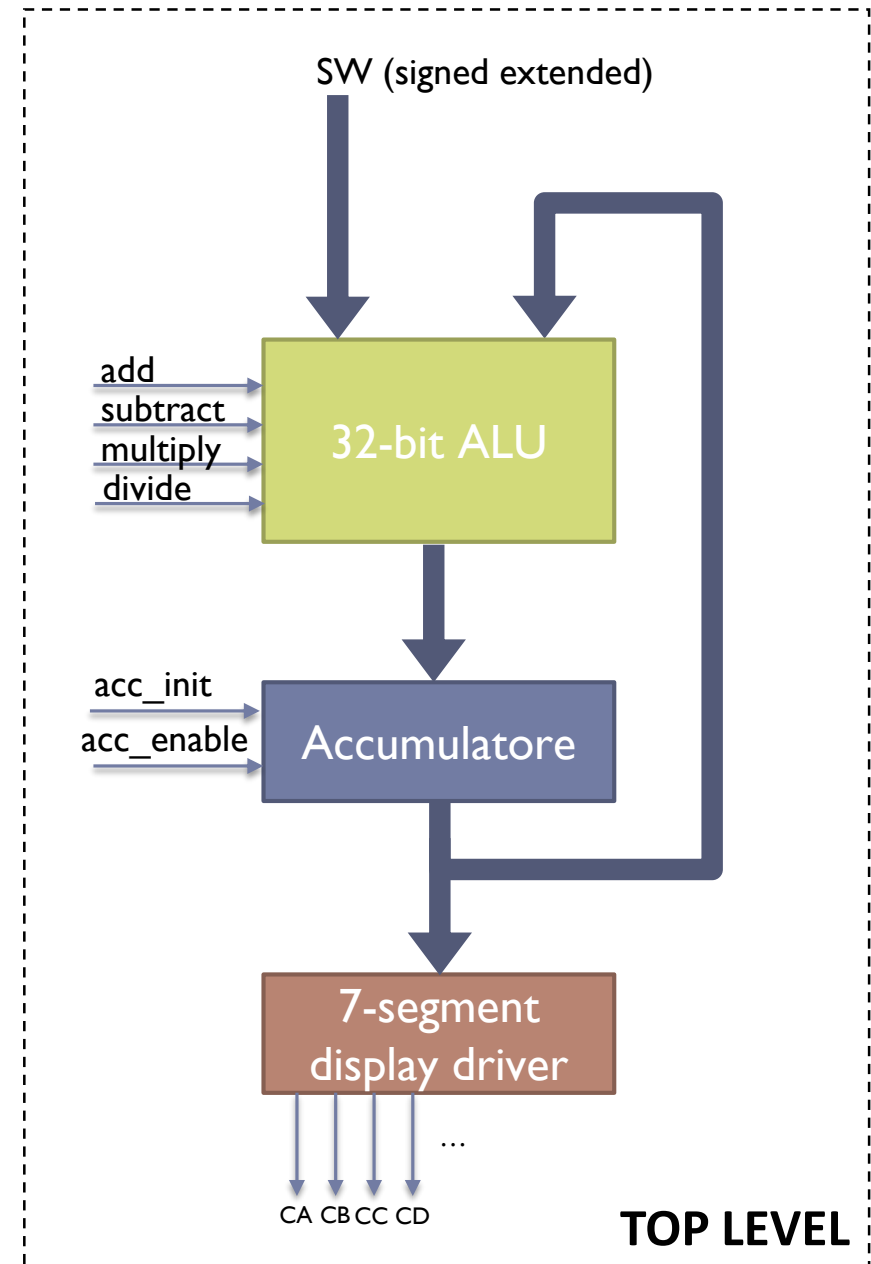


# Architettura

- Utilizzare i bottoni per selezionare l'operazione secondo il seguente schema.

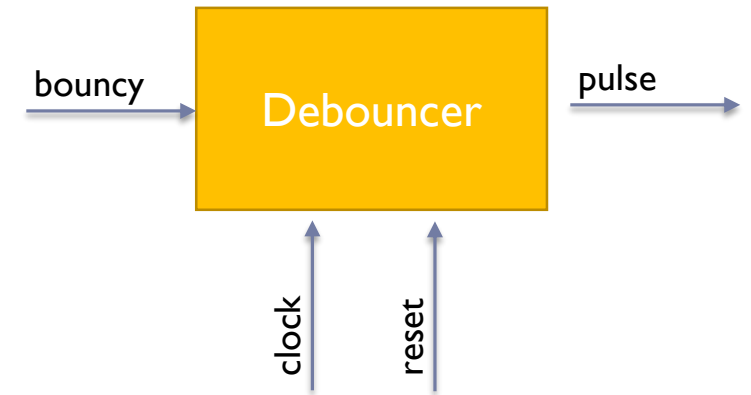


- Usare degli edge detector che implementino la funzione di debouncer. L'uscita dovrà essere attiva per un solo ciclo di clock per ogni pressione del bottone.



# Debouncer

- Il bottone è un componente elettromeccanico rumoroso. La chiusura contatto genera treni di impulsi spurii.
- Debouncer: produce un solo impulso stabile in presenza di un ingresso elettricamente rumoroso.
  - Come si fa?
    - Ogni volta che il bottone cambia di valore si resetta un contatore.
    - Per ogni ciclo di clock in cui il valore del bottone è uguale al ciclo di clock precedente sottrae 1 al contatore, altrimenti aggiorna il valore e resetto il contatore.
    - Quando il valore del contatore è zero, allora considera valore stabile.
    - Si attiva l'uscita per un periodo di clock solamente quando il valore stabile differisce dal valore che aveva il ciclo di clock precedente.



# ALU

- Esegue i calcoli
- Due ingressi a 32 bit ciascuno per gli operandi (di tipo signed)
  - Il primo viene dagli switch sulla board
  - Il secondo viene dall'accumulatore
- Ha ulteriori 3 ingressi per ciascuna delle operazioni che vogliamo implementare (ognuno di questi proviene dal relativo pushbutton con debouncer)
- Una uscita (signed 32 bit) per il risultato
- Occhio al numero di bit necessari per l'operazione di moltiplicazione!

# Accumulator

- È il blocco più semplice
- Riceve in ingresso il risultato della ALU e ad ogni fronte del clock, se un segnale di enable lo permette, lo mette in uscita.
- Il segnale di enable è a '1' ogni qualvolta si vuole eseguire una operazione, potete quindi farlo come la OR tra i vari debouncer dei pushbutton delle operazioni.
- Ha inoltre un segnale di inizializzazione proveniente da un pushbutton (per azzerare la memoria).

# Suggerimenti

- Simulare prima le seguenti entità separatamente e poi integrare il tutto:
  - Debouncer per i pushbutton
  - ALU
  - Accumulator
- Utilizzare la traccia e testbench forniti

# Note

- **NEXYS4**

- Codice FPGA: xc7a100tcsg324-1
- Utilizzare il driver per display a 7 segmenti fornito.

- **ZedBoard**

- Codice FPGA: xc7z020clg484-1
- 8 switch invece di 16 → Ingresso a 8 bit
- Utilizzare il driver per display OLED fornito.
- Usare il bottone in basso BTND come display shutdown, invece che collegarlo alla ALU.