

Relazione di Reti Logiche - 1, 14 dicembre 2022

Obiettivi

Tale report andrà a coprire gli aspetti relativi alla progettazione e realizzazione di una semplice calcolatrice in VHDL, in grado di svolgere le principali operazioni aritmetiche, quali addizione, sottrazione e moltiplicazione. Per questo, abbiamo fatto uso delle conoscenze teoriche apprese durante le lezioni, tra cui le applicazioni dei *multiplexers*, delle macchine a stati finiti, dei *full adders* e dei *flip-flops*. La principale difficoltà nella progettazione di una calcolatrice, oltre all'implementazione del *debouncer*, consiste nell'essere limitati al solo sistema numerico binario, all'avere cioè come input soltanto uni e zeri inseriti attraverso gli *switches* di una **Nexys 4 DDR**.

Schema a blocchi

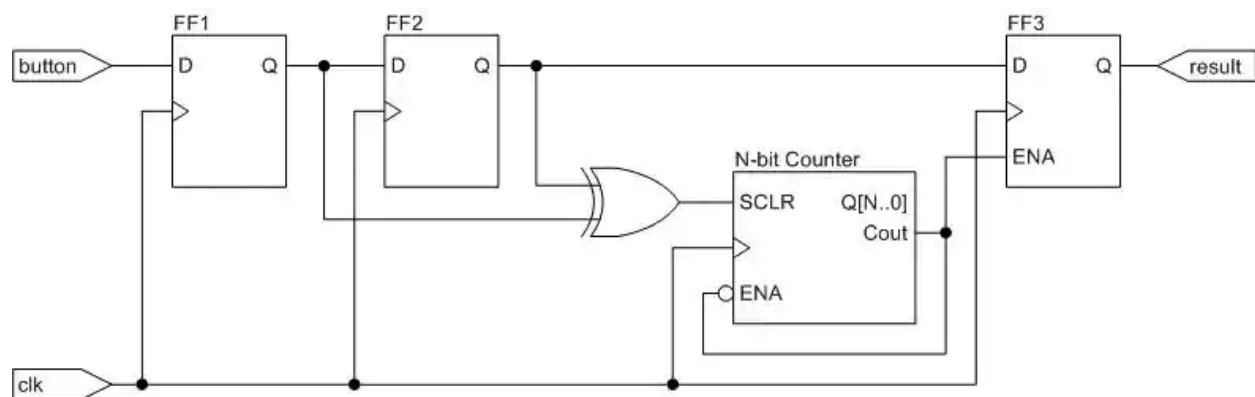


Fig. 1

Come mostrato in Fig. 1, i primi due flip-flops (FF1 e FF2) memorizzano lo stato del pulsante nel momento in cui viene premuto. Successivamente, una porta logica XOR confronta lo stato attuale del bottone (FF2) con lo stato precedente (FF1) e abilita l'uscita quando i due stati sono diversi. Inoltre, l'output della porta XOR è collegato ad un contatore (*N-bit Counter*), che viene resettato se l'uscita è vera. Altrimenti, se l'output della XOR è falso, per ogni ciclo di clock in cui i due stati sono uguali, viene sottratto 1 al valore del contatore.

Un terzo flip-flop (FF3), posto alla fine, viene attivato e memorizza lo stato di FF2 nel caso in cui il contatore non restituisca alcun valore.

Componenti

Per la realizzazione della calcolatrice, abbiamo utilizzato principalmente tre componenti: un accumulatore, un ALU e un debouncer. Il primo riceve in ingresso il risultato della ALU e ad ogni fronte del clock lo mette in uscita; il secondo è il processore che esegue i calcoli aritmetici, in questo caso addizione, sottrazione e moltiplicazione; il terzo invece produce un solo impulso stabile in presenza di un ingresso elettricamente rumoroso, eliminando quindi i vari impulsi spurii generati dai bottoni.

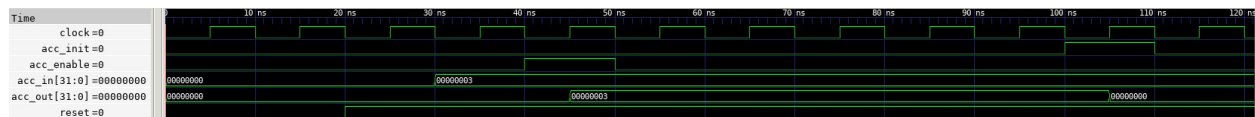


Fig. 2: diagramma temporale dell'accumulatore

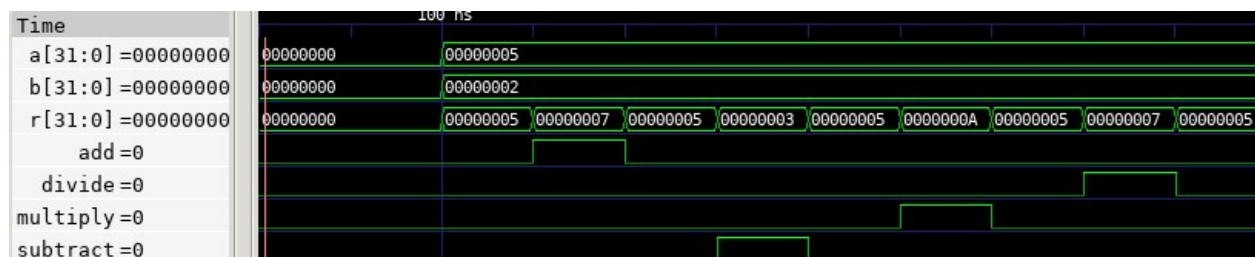


Fig. 3: diagramma temporale dell'ALU

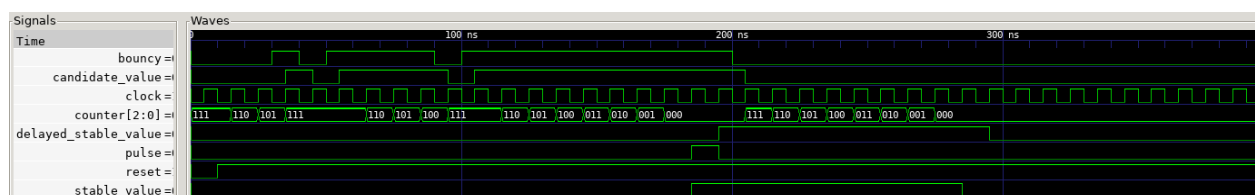


Fig. 4: diagramma temporale del debouncer

Risultati

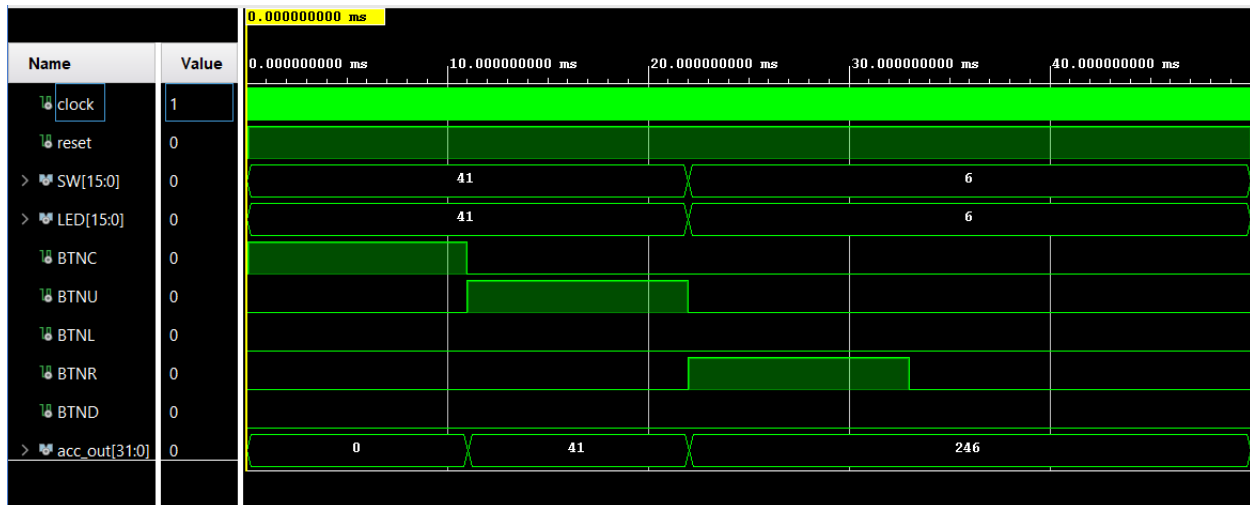


Fig. 5

Quello sopra riportato (Fig. 5) è il diagramma temporale (*testbench*) dell'architettura completa della calcolatrice. Inizialmente, il pulsante centrale (BTNC) si attiva ed inizializza a 0 il valore dell'accumulatore. In seguito, il pulsante superiore (BTNU) somma il valore attuale dell'accumulatore al numero in ingresso (SW): in questo caso risulta $0 + 41$, salvando quindi 41. Infine, viene attivato il pulsante destro (BTNR), il quale moltiplica il valore dell'accumulatore con quello in ingresso che, nel frattempo, è diventato 6; il valore visualizzato sul display (segnale *acc_out*) sarà quindi $6 \cdot 41 = 246$.

Il seguente (Fig. 6) è il testbench relativo all'architettura completa del sistema PWM.

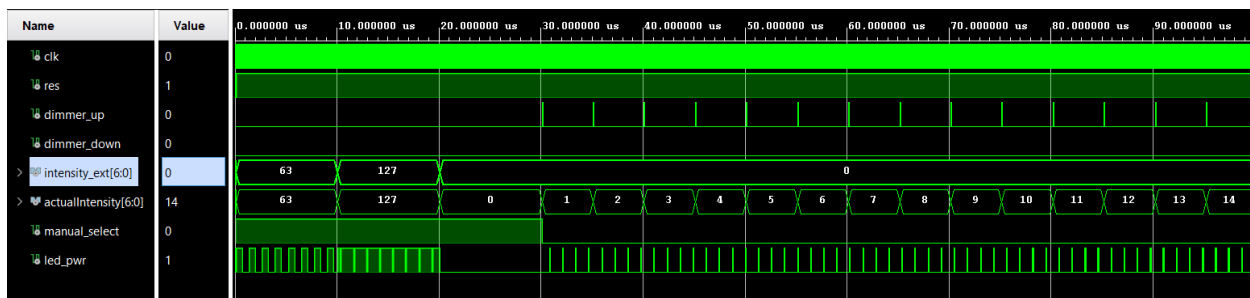


Fig. 6

Da questa simulazione si può osservare come inizialmente l'intensità del led vari da un valore pari a 63, dove il *duty cycle* è circa al 50%, ad uno pari a 127, dove il *duty cycle* è al 100% e quindi la luminosità del led è massima. Successivamente, lo switch[0] si attiva, avviando la modalità manuale della gestione della luminosità del led: si può vedere infatti come l'intensità incrementi da 0 ad 1 per ogni segnale di *dimmer up* (left button).

Crediti

- Materiale condiviso dai docenti
 - <https://www.digikey.it/en/articles/how-to-debounce-a-button-input-using-programmable-logic>
 - [https://en.wikipedia.org/wiki/Accumulator_\(computing\)](https://en.wikipedia.org/wiki/Accumulator_(computing))
 - https://en.wikipedia.org/wiki/Arithmetic_logic_unit
-

Report realizzato da

Demartin Gustavo,

Hangu David,

Pavona Tobia e

Repele Tommaso

Università degli Studi di

Trento - 2022/2023



**UNIVERSITÀ
DI TRENTO**