

Progetto corso di PASD

Filippo Landi

30 giugno 2021

Sommario

Il mio progetto per il corso di Progetto Automatico di Sistemi Digitali (PASD) consiste nello studio statistico di alcuni possibili difetti di un circuito di tipo "multiply and accumulate".

1 Introduzione al progetto

Il progetto si avvale:

- Di alcuni concetti inerenti il collaudo dei circuiti digitali.
- Del software libero "hope" un software di electronic design automation usato per il collaudo dei circuiti, esso sfrutta una descrizione dei circuiti (netlist) in formato .bench.

2 Circuito multiply and accumulate

Mi è stato richiesto di realizzare questo circuito a 8 bit. Ho ricevuto uno schema dal professore di un circuito di questo tipo a 4 bit, riporto quindi la documentazione.

2.1 Documentazione fornita

Il multiply and accumulate è una unità ampiamente utilizzata nei DSP e nel machine learning per realizzare funzioni del tipo $y = \sum_{i=0}^{n-1} x_i w_i$. La dimensione degli ingressi sia n , per cui quella dell'uscita del moltiplicatore sia $2n$.

Qui c'è lo schema di un moltiplicatore parallelo a 4 bit che può essere esteso per valori più grandi di n .

2.2 Considerazioni ulteriori

Il ripple carry adder è quello standard, ha ingressi quindi $2n$, l'uscita va ad un registro realizzato con flip flop d che retroaziona l'uscita in modo da realizzare la somma dei vari risultati. Potrebbe essere necessario un'ulteriore porta AND ai fini del collaudo ma prendo questo in considerazione successivamente.

3 Approccio alla descrizione in formato .bench

La stesura manuale del circuito a 8 bit non mi sembrava un approccio intelligente al problema: è facile commettere errori sia (banalmente) di battitura sia dovuti ad una scarsa comprensione dell'architettura del circuito.

Il mio approccio quindi è stato quello di scrivere uno script in python che descrivesse e collegasse le varie parti del circuito.

In partenza ho scritto alcuni circuiti di prova, che trovate nella cartella omonima, per comprendere come eseguire questa operazione.

Credo sia interessante notare come questo approccio abbia permesso di avere una dimensione in bit del circuito arbitraria.

Il risultato è il file "multiplier_generator.py": basta eseguirlo con python versione 3 (io ho Python 3.8.5) con argomento il numero di bit del circuito desiderati.