
ANALISI E VALUTAZIONE DI UN SET DI MOLTIPLICATORI APPROSSIMATI

Daniele Zinghirino - 1000050613

INTRODUZIONE

Obiettivi del lavoro

Lo scopo del lavoro presentato nel seguente elaborato è quello di prendere sotto esame e valutare in termini di accuratezza e prestazioni un set di **moltiplicatori approssimati**. Nello specifico si vuole valutare l'andamento, al variare del livello di approssimazione, di due moltiplicatori: **un moltiplicatore senza segno ad 8-bit** e un **moltiplicatore senza segno a 16-bit**. L'analisi verrà effettuata valutando determinati parametri presentati in seguito.

Per poter svolgere accuratamente tale valutazione, nel lavoro eseguito si è scelto di utilizzare un set di modelli di moltiplicatori forniti dalla libreria **EvoApproxLib Lite**.

Modelli di moltiplicatori

EvoApproxLib Lite (<https://github.com/ehw-fit/evoapproxlib>) è una libreria open-source che fornisce un'ampia collezione di implementazioni approssimative di operazioni aritmetiche a basso livello. La libreria mette a disposizione approssimazioni per sommatore e moltiplicatori - con e senza segno - di molteplici dimensioni, fornendo per ognuno di essi i parametri di design ottenuti considerando un design compiler **Synopsys DC** con un process design kit a **45nm**, con una tensione di **1V** e temperatura di **25°C**. In tali condizioni di lavoro vengono dunque forniti i valori di:

- **Consumo di potenza** (in mW)
- **Area nel chip** (in μm^2)
- **Delay** (in ns)

Al fine di valutare l'impatto che ha l'approssimazione sul circuito in termini di accuratezza e precisione, è utile considerare una serie di parametri quali:

- **MAE (Errore Assoluto Medio)** → Misura dell'errore medio tra i valori delle moltiplicazioni esatte e i valori delle moltiplicazioni approssimate. Calcolato come il valore assoluto della differenza tra i due valori.
- **WCE (Errore Assoluto nel Worst-Case)** → Misura dell'errore massimo tra tutti i valori delle moltiplicazioni esatte e i valori delle moltiplicazioni approssimate effettuate con gli stessi operandi. Rappresenta il momento di peggior funzionamento del moltiplicatore.
- **MSE (Errore Quadratico Medio)** → Misura dell'errore quadratico medio tra i valori delle moltiplicazioni esatte e i valori delle moltiplicazioni approssimate. Calcolato sommando i quadrati delle differenze tra ciascuna moltiplicazione approssimata e il corrispondente valore della moltiplicazione esatta e dividendolo per il totale di moltiplicazioni.
- **MRE (Errore Relativo Medio)** → Misura dell'errore percentuale medio tra le moltiplicazioni approssimate e le moltiplicazioni esatte. Si calcola sommando le differenze percentuali assolute tra ogni moltiplicazione approssimata e il valore corrispondente di moltiplicazione esatta e dividendolo per il numero di moltiplicazioni.
- **WCRE (Errore Relativo nel Worst-Case)** → Errore percentuale massimo che si verifica tra tutte le moltiplicazioni esatte e i valori delle moltiplicazioni approssimate. Rappresenta la peggiore discrepanza percentuale tra due moltiplicazioni.
- **EP (Probabilità di Errore)** → Probabilità che il valore della moltiplicazione approssimata sia diverso dal valore della moltiplicazione esatta.

La libreria consente di scegliere, per ogni moltiplicatore, tra diversi **dataset** che rappresentano un sottoinsieme di soluzioni ottime rispetto a due parametri in conflitto tra loro (uno di questi è sempre la potenza del circuito): in tal modo è possibile analizzare un determinato set di moltiplicatori e analizzarne le sue prestazioni hardware e accuratezza, per un determinato dataset, al variare del livello di approssimazione applicata al circuito: in questo lavoro si utilizzerà un dataset che metta in contrapposizione il **consumo di potenza** con l'**errore assoluto Worst-Case (WCE)**.

Il livello di approssimazione applicata ai moltiplicatori messi a disposizione da *EvoApproxLib Lite* si basa su cinque diverse metodologie:

- Approssimazione della generazione di prodotti parziali che utilizza una struttura più semplice per generare prodotti parziali;
- Approssimazione nell'albero dei prodotti parziali ignorando alcuni prodotti parziali, dividendo i prodotti parziali in più moduli e applicando l'approssimazione nei moduli meno significativi;
- Utilizzo di contatori o compressor approssimativi nell'albero dei prodotti parziali;

- Utilizzo di moltiplicatori che utilizzano l'algoritmo del prodotto di Booth approssimativi;
- Composizione di moltiplicatori approssimativi complessi per mezzo di moltiplicatori approssimativi più semplici;

In questo elaborato verranno presi sotto esame sei moltiplicatori a 8-bit senza segno e sei moltiplicatori a 16-bit senza segno aventi, per ciascuna categoria, cinque diversi livelli di approssimazione applicati in ordine crescente di imprecisione (oltre ad analizzare il moltiplicatore senza alcuna approssimazione applicata).

Procedura di valutazione

EvoApproxLib Lite offre modelli hardware e software compatibili in diversi ambienti, come *MATLAB*, *Verilog* o *C*. In questo lavoro si utilizzerà un ambiente **Python** nella seguente maniera:

1. Tramite la libreria *EvoApproxLib Lite* verranno selezionate 2 tipologie di moltiplicatore: un **moltiplicatore a 8-bit** e un **moltiplicatore a 16-bit**, entrambi **senza segno**;
2. Per ogni tipologia verrà selezionato un determinato **dataset**, secondo le modalità precedentemente descritte;
3. Per ogni tipologia di moltiplicatore, scelto il dataset, verranno selezionati sei diversi modelli: il modello del **moltiplicatore non approssimato** e **cinque modelli** di moltiplicatore con **crescente livello di approssimazione**;
4. Una volta individuati i dodici modelli utili, tramite **Cython** verranno importati i rispettivi modelli scritti in **C** all'interno di un programma **Python**, in un modulo apposito rappresentante il determinato moltiplicatore;
5. All'interno di ogni modulo verrà creato un file *Python* **analysis.py** nel quale avverrà l'analisi del rispettivo moltiplicatore;
6. L'analisi avverrà facendo operare il moltiplicatore con tutti i numeri da esso supportati: verranno dunque eseguite moltiplicazioni tra una coppia di numeri interi, ognuno dei quali appartenente all'intervallo **$[0, 2^8 - 1]$** per i moltiplicatori ad **8-bit** e all'intervallo **$[0, 2^{16} - 1]$** per i moltiplicatori a **16-bit**;
7. Durante queste iterazioni verranno rilevati gli **errori** dovuti all'approssimazione del moltiplicatore e la **discrepanza tra i risultati** delle moltiplicazioni esatte e di quelle approssimate;
8. Una volta concluse tutte le operazioni, verranno mostrati **valori finali** ottenuti per quella determinata simulazione;
9. La procedura verrà ripetuta per ogni moltiplicatore scelto nella prima fase.

Per ogni circuito il codice *Python* di tale procedura sarà del tipo:

```
# Definizione del numero di operandi
op_len = 2**8 #(per i moltiplicatori a 8-bit, altrimenti 2**16)
mult_len = op_len * op_len

# Definizione ed inizializzazione delle variabili per calcolare le metriche
wce = e = e_sq = e_count = e_rel = wce_rel = diff_rel = 0

# Loop per eseguire tutte le moltiplicazioni
for i in range(0, op_len):
    for j in range(0, op_len):
        # Calcolo dell'errore assoluto tra il risultato della
        # moltiplicazione approssimata e il risultato della moltiplicazione
        # esatta tra i due numeri i e j
        diff = abs(MOLTIPLICATORE_APPROSSIMATO.mul(i,j) - (i*j))

        # Verifica se i e j sono entrambi maggiori di zero per evitare
        # divisioni per zero
        if i > 0 and j > 0:
            # Calcolo dell'errore relativo, normalizzato rispetto
            # alla moltiplicazione esatta
            diff_rel = abs(MOLTIPLICATORE_APPROSSIMATO.mul(i,j) - (i*j))/(i*j)

            # Aggiornamento del massimo errore assoluto
            if diff > wce:
                wce = diff

            # Aggiornamento del massimo errore relativo
            if diff_rel > wce_rel:
                wce_rel = diff_rel

            # Aggiornamento del numero di errori
            if diff != 0:
                e_count += 1

            # Aggiornamento delle somme cumulative per calcolare le medie
            # (assoluta, quadratica e relativa)
            e += diff
            e_sq += diff * diff
            e_rel += diff_rel

# Visualizzazione a schermo dei valori delle metriche in percentuale
print('mean_absolute_error (MAE) = ', e * 100 / (mult_len * e_count))
print('worst-case error (WCE) = ', wce * 100 / mult_len)
print('mean_squared_error (MSE) = ', e_sq * 100 / mult_len)
print('mean_relative_error (MRE) = ', e_rel * 100 / mult_len)
print('error_probability (EP) = ', e_count * 100 / mult_len)
```

DATI OTTENUTI - MOLTIPLICATORI 8-BIT

No approssimazione:

WCE	MAE	MRE	MSE	EP	Potenza	Area	Delay
0.00%	0.00%	0.00%	0	0.00%	0.391mW	709.6 μm^2	1.43ns

Approssimazione 1:

WCE	MAE	MRE	MSE	EP	Potenza	Area	Delay
0.017%	0.0046%	0.18%	17	75%	0.370mW	637.8 μm^2	1.40ns

Approssimazione 2:

WCE	MAE	MRE	MSE	EP	Potenza	Area	Delay
0.18%	0.057%	2.56%	2008	97.72%	0.302mW	542.5 μm^2	1.44ns

Approssimazione 3:

WCE	MAE	MRE	MSE	EP	Potenza	Area	Delay
0.66%	0.18%	4.42%	21037	98.37%	0.206mW	395.6 μm^2	1.34ns

Approssimazione 4:

WCE	MAE	MRE	MSE	EP	Potenza	Area	Delay
2.41%	0.56%	10.85%	209723	98.99%	0.104mW	239.3 μm^2	1.00ns

Approssimazione 5:

WCE	MAE	MRE	MSE	EP	Potenza	Area	Delay
8.21%	2.15%	39.78%	3086510	99.16%	0.034mW	110.8 μm^2	0.58ns

DATI OTTENUTI - MOLTIPLICATORI 16-BIT

No approssimazione:

WCE	MAE	MRE	EP	Potenza	Area	Delay
0.00%	0.00%	0.00%	0.00%	2.202mW	3203.0 μm^2	3.11ns

Approssimazione 1:

WCE	MAE	MRE	EP	Potenza	Area	Delay
0.0000027%	0.00000086%	0.00013%	97.72%	2.106mW	2992.3 μm^2	3.09ns

Approssimazione 2:

WCE	MAE	MRE	EP	Potenza	Area	Delay
0.00042%	0.000075%	0.0067%	99.84%	1.648mW	2404.2 μm^2	2.44ns

Approssimazione 3:

WCE	MAE	MRE	EP	Potenza	Area	Delay
0.058%	0.011%	0.46%	100%	0.791mW	1284.9 μm^2	2.03ns

Approssimazione 4:

WCE	MAE	MRE	EP	Potenza	Area	Delay
0.63%	0.16%	3.06%	100%	0.401mW	642.0 μm^2	1.97ns

Approssimazione 5:

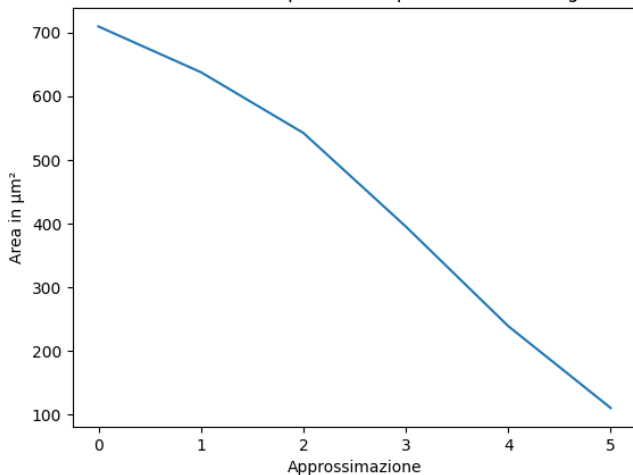
WCE	MAE	MRE	EP	Potenza	Area	Delay
7.03%	1.76%	19.30%	100%	0.101mW	244.0 μm^2	1.03ns

VALUTAZIONE DATI

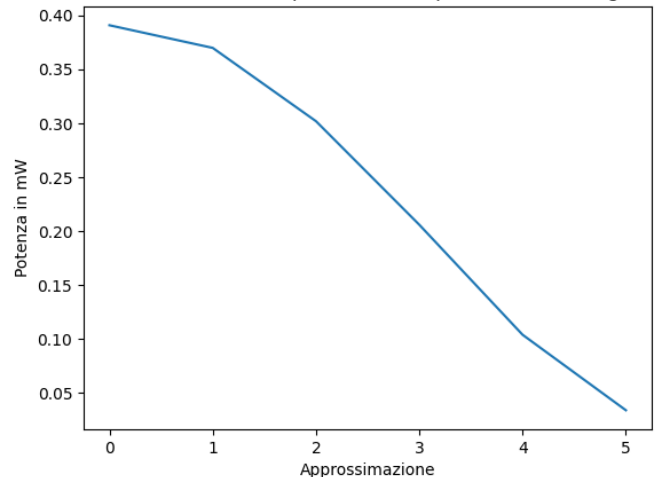
I principali dati ricavati dalle simulazioni vengono mostrati di seguito in forma di diagramma, ottenuti tramite l'utilizzo della libreria *Python Matplotlib*.

Per quanto riguarda i **moltiplicatori senza segno a 8-bit**:

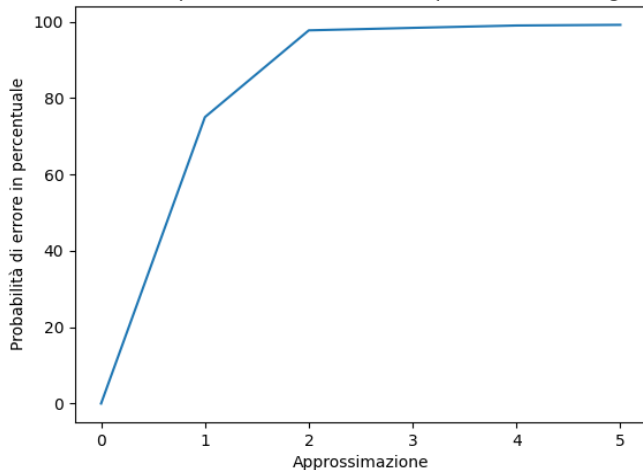
Andamento dell'area occupata - moltiplicatore senza segno a 8-bit



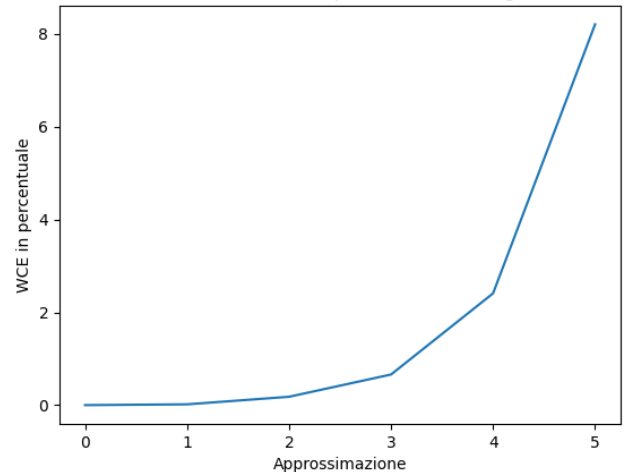
Andamento del consumo di potenza - moltiplicatore senza segno a 8-bit



Andamento della probabilità di errore - moltiplicatore senza segno a 8-bit



Andamento del WCE - moltiplicatore senza segno a 8-bit

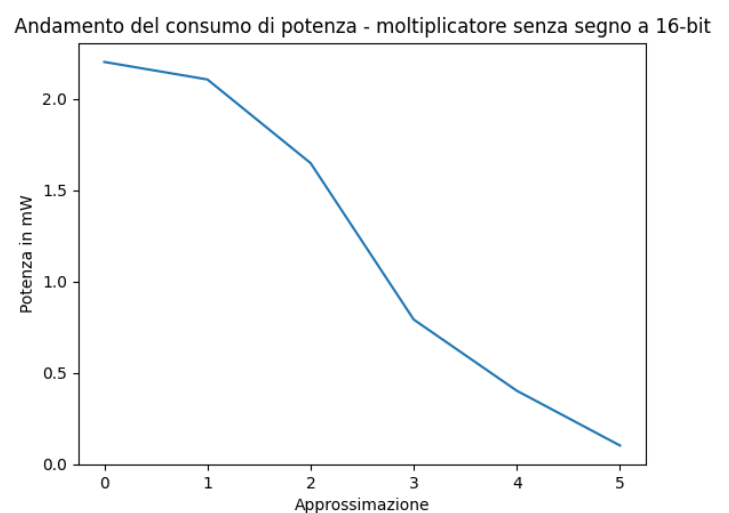
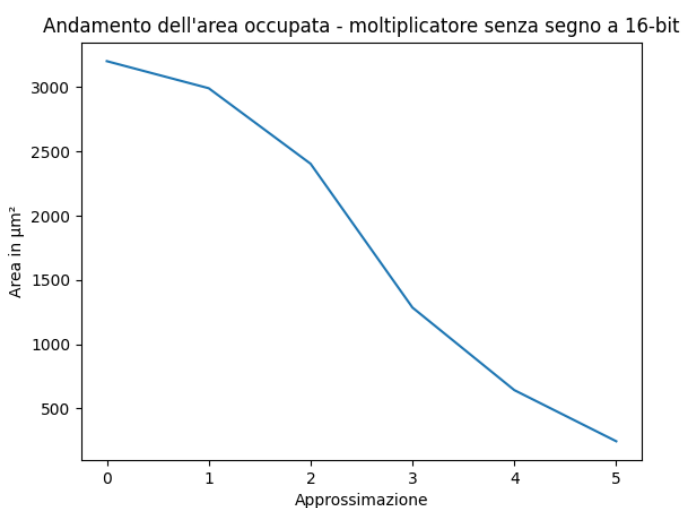


Avendo come riferimento il moltiplicatore non approssimato, è subito evidente come sin dal **primo livello di approssimazione** vi è un repentino aumento della **probabilità di errore**, seppur ancora in valori accettabili: nel moltiplicatore col minor numero di approssimazioni attuate il risultato di **una moltiplicazione su quattro** è uguale al valore ottenuto da una moltiplicazione esatta operante con gli stessi numeri. In tale configurazione non viene però rilevata una grossa riduzione della potenza consumata.

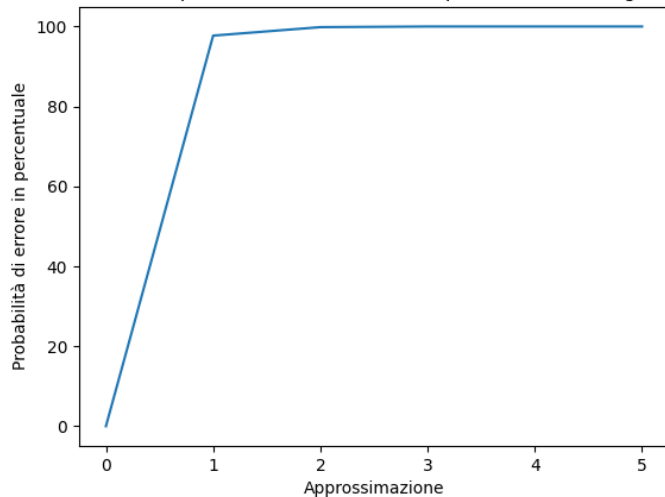
Dal **secondo livello di approssimazione** in poi la probabilità di errore è prossima al **100%** nonostante non venga mai raggiunto tale valore. Dunque, vengono rilevate (pochissime) moltiplicazioni approssimate con valore identico al risultato di una moltiplicazione esatta: per queste configurazioni è utile confrontare il **WCE**, che dà un'indicazione sulla **massima inesattezza** riscontrata nei risultati delle moltiplicazioni. Il valore di WCE cresce vertiginosamente all'aumentare del livello di approssimazione applicata, rendendo le **approssimazioni 4 e 5 difficilmente realizzabili** in applicazioni reali, nonostante in tali configurazioni si registri una grande diminuzione di area occupata e di consumo di potenza: la **potenza erogata** dal moltiplicatore col massimo livello di approssimazione risulta essere di **un ordine di grandezza inferiore** rispetto a quella del moltiplicatore esatto, così come l'**area occupata** dal circuito risulta essere circa **sei volte inferiore**.

Un compromesso accettabile tra accuratezza e vantaggi in termini di design all'interno di questo set di moltiplicatori potrebbe risiedere nel circuito con **approssimazione di livello 2** per le applicazioni che necessitano di più **accuratezza**, o nella configurazione con **livello di approssimazione 3** laddove i vincoli di precisione siano meno stringenti o si abbia la necessità di ridurre l'area occupata e il consumo di potenza, ad esempio all'interno di **dispositivi mobili**.

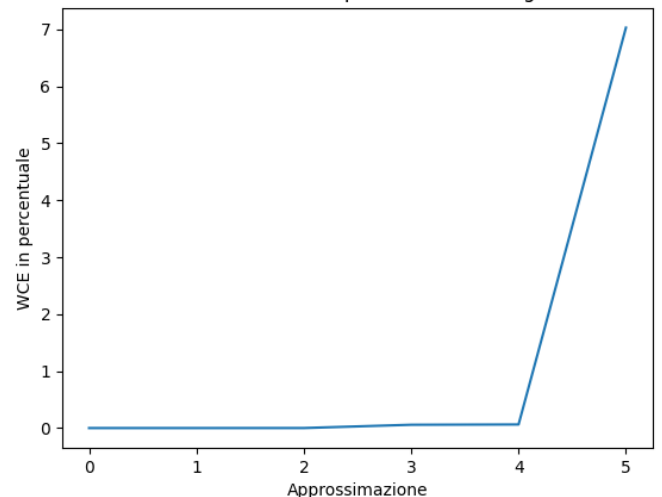
Le valutazioni sono leggermente diverse per quanto riguarda i **moltiplicatori a 16-bit**:



Andamento della probabilità di errore - moltiplicatore senza segno a 16-bit



Andamento del WCE - moltiplicatore senza segno a 16-bit



Per questo set di moltiplicatori la **probabilità di errore**, in caso di approssimazione, è in qualsiasi caso **prossima al 100%** e dal livello di approssimazione 3 in poi addirittura la probabilità di errore raggiunge tale valore; ciò significa che **ogni** moltiplicazione approssimata attuata con questi metodi comporta un risultato che differisce dalla moltiplicazione effettuata dal circuito accurato, motivo per il quale è necessario analizzare il **WCE**. Per questo set di circuiti tale valore si mantiene ad un livello accettabile, salvo poi aumentare repentinamente nel circuito con **livello di approssimazione 5**: tale valore rende di fatto il circuito inutilizzabile in applicazioni reali nonostante riduca il **consumo di energia di circa venti volte** e l'**area occupata di circa tredici volte** rispetto al moltiplicatore non approssimato.

Data la presenza di errore in quasi la totalità dei moltiplicatori approssimati di tale set, negli scenari dove l'**accuratezza** rappresenta un vincolo da rispettare questi circuiti approssimati non sono applicabili e **non rappresentano una soluzione**. Nelle applicazioni dove invece è possibile tollerare gli errori dei risultati sotto un limite massimo di scostamento dai valori di moltiplicazioni esatte, il circuito con **livello di approssimazione 4** permette di ridurre notevolmente l'area occupata, la potenza erogata e il delay, permettendone l'utilizzo in una situazione reale.