



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

FACOLTÀ DI INGEGNERIA ELETTRONICA

Materia
ELETTRONICA DEI SISTEMI DIGITALI - M

Progetto e caratterizzazione di un sommatore Kogge-Stone BLC a 8 bits

Professoressa: Eleonora Franchi Scarselli

Studenti:

Lorenzo Stella

Matricola:
0001139075

Giulio Centini

Matricola:
0001147941

Anno Accademico 2023-2024

Indice

1	<u>Introduzione</u>	2
2	<u>Ottimizzazione della latenza</u>	2
3	<u>Sistema digitale</u>	3
3.1	Rete Combinatoria	3
3.1.1	Celle standard	3
3.1.2	Blocco XOR	4
3.1.3	Blocco "single bit"	4
3.1.4	Blocco operatore "O"	4
3.2	Registri	5
3.2.1	Struttura interna	5
4	<u>Operazioni di somma e sottrazione</u>	6
5	<u>Caratterizzazione</u>	7
5.1	Prestazioni dinamiche	7
5.2	Potenza statica media	7
5.3	Energia media per ciclo di clock	7
5.3.1	Valori energetici relativi a rete combinatoria e registri.	7
5.4	Area occupata	7
6	<u>Considerazioni finali</u>	8

1 Introduzione

Il presente elaborato mostra il progetto di un circuito sommatore con $N=8$ numero di bit impiegati per operandi in ingresso e risultato in uscita. Tale progettazione è composta da varie fasi:

- 1) Scelta della tipologia circuitale del sommatore (associata alla specifica di progetto da ottimizzare).
- 2) Realizzazione e verifica funzionale (operazioni di somma e sottrazione).
- 3) Caratterizzazione del circuito, con particolare attenzione rivolta al consumo di potenza statica media e all'energia media ottenuta al variare del coefficiente di attività α (1, 0.5, 0).

2 Ottimizzazione della latenza

La scelta del tipo di architettura è dovuta alla specifica da ottimizzare. La rete combinatoria impiegata nel sistema digitale del sommatore è *Kogge-Stone BLC* modificato¹, ovvero un'architettura di tipo Carry-lookahead impiegata per ridurre la **latenza** (ovvero il tempo di propagazione, T_P). La sigla "BLC" sta per "Brent-Kung Linear Combination", ovvero una tecnica che mira ad ottenere la parallelizzazione delle linee dati bilanciando il trade-off latenza/(area e potenza).

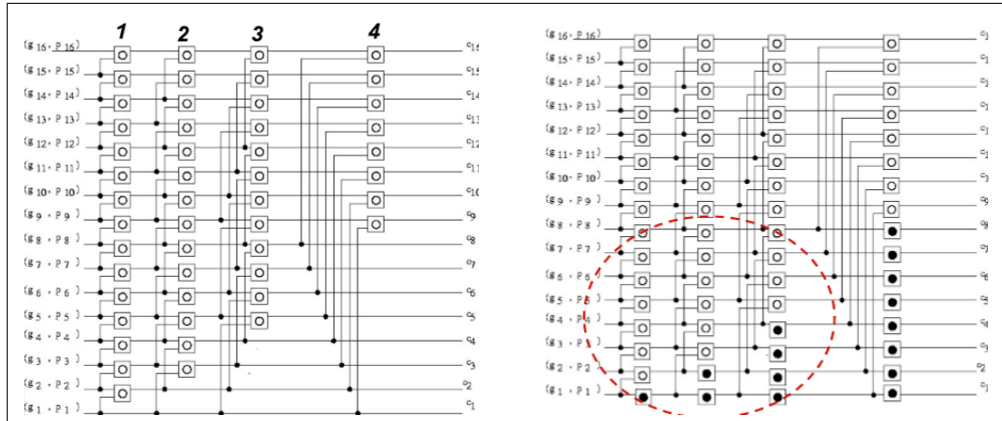


Figura 1: Modifica dell'albero binario di tipo Kogge-Stone, con sotto-blocco ad 8 bit evidenziato.

Nella figura di destra è possibile osservare due tipologie di blocchi:

- Blocchi "O" vuoti (bianchi), che rappresentano gli elementi utili a fornire segnali di generazione e propagazione;
- Blocchi "O" pieni (neri), che corrispondono agli operatori "O" impiegati come buffer, per favorire la somiglianza tra ciascuna delle linee dati in termini di latenza;

Di seguito è mostrato il tempo impiegato per la somma:

$$T_{adder} = t_{g,p} + \log_2[N] * T_{op,O} + t_{XOR} \quad (1)$$

Ipotizzando che i ritardi introdotti dai vari blocchi siano uguali tra di loro ($= \tau$):

$$T_{adder,est} = \tau(2 + \log_2[N]) \quad (2)$$

¹La modifica consiste nell'aggiunta dei blocchi "O" pieni

3 Sistema digitale

Il sistema digitale presenta una rete combinatoria di tipo Kogge-Stone modificato e tre registri, che ricevono in ingresso il segnale di clock indispensabile per il corretto campionamento dei dati.

3.1 Rete Combinatoria

La rete combinatoria del sommatore Kogge-Stone, ha una struttura ad albero, vale a dire una parallelizzazione delle linee dati, composte da blocchi logici di vario tipo. Nella figura seguente sono indicati:

- SB = Blocco "single bit";
- O = Operatore "O";
- O-G = Operatore "O" impiegato per il generate di gruppo;
- O-B = Operatore "O" impiegato come buffer;

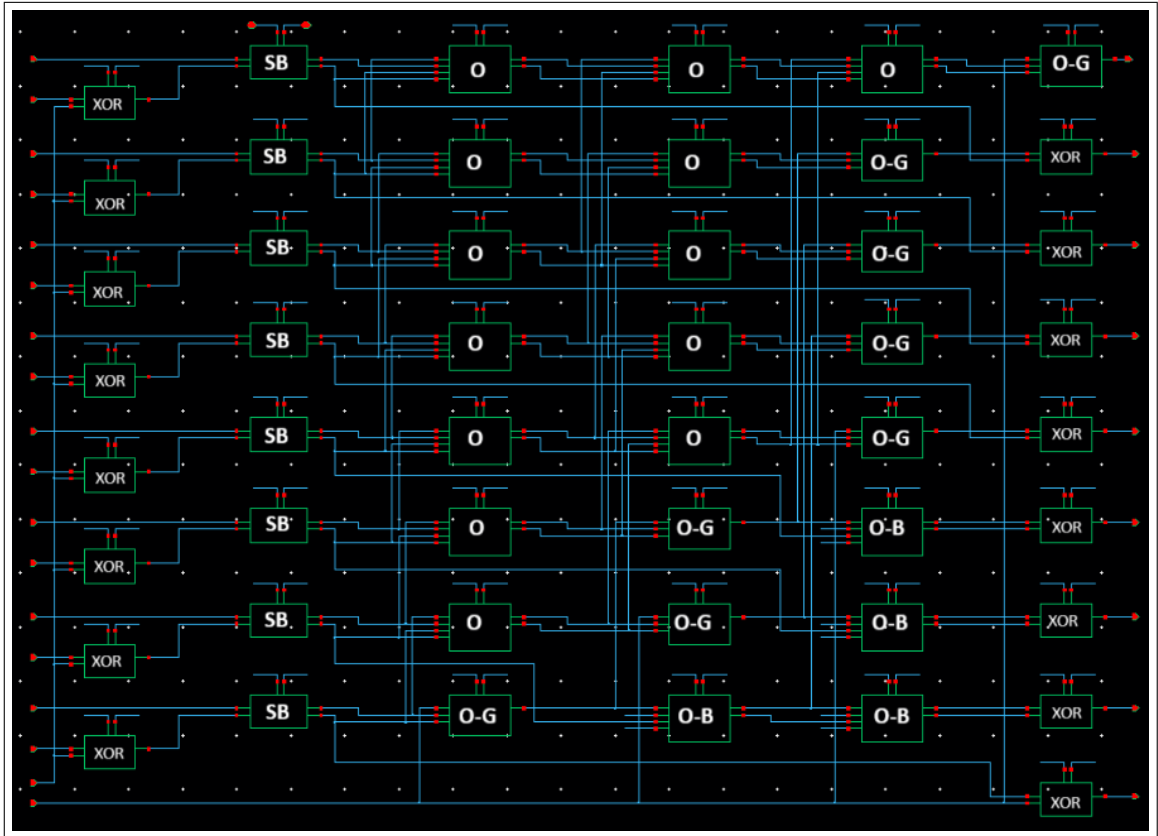


Figura 2: Vista complessiva della rete combinatoria.

3.1.1 Celle standard

Le celle standard impiegate nella realizzazione della rete combinatoria, al fine di realizzare i vari blocchi sono:

<i>INV – RVT – X1</i>	<i>NAND2 – RVT – X1</i>	<i>AOI21 – RVT – X1</i>	<i>AOI22 – RVT – X1</i>
-----------------------	-------------------------	-------------------------	-------------------------

3.1.2 Blocco XOR

Il blocco XOR viene impiegato sia in ingresso per consentire la differenza tra gli operandi sia per eseguire la somma in uscita.

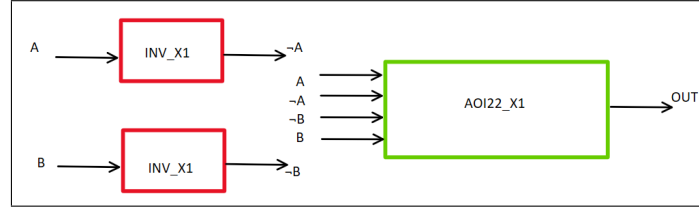


Figura 3: Blocco XOR.

3.1.3 Blocco "single bit"

Questo blocco logico è composto da due ingressi e due uscite. Esso ha il compito di ricevere in ingresso singoli bit dei due operandi campionati dai registri (A_i e B_i) e fornire in uscita i due segnali noti come "generate" (G_i) e "propagate" (P_i).

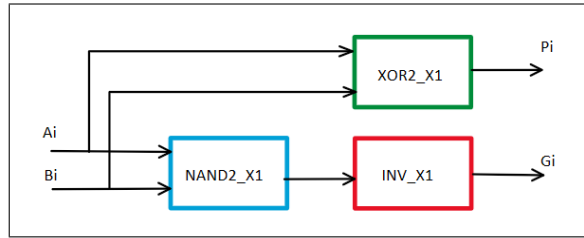


Figura 4: Blocco "single bit".

3.1.4 Blocco operatore "O"

L'operatore "O" rappresenta l'elemento principale della rete combinatoria e riceve in ingresso le due coppie di segnali *generate* e *propagate* provenienti dai blocchi precedenti definendo la propagazione del dato.

In fase di realizzazione tuttavia, è necessario tenere conto del fatto che non tutti i blocchi "O" hanno lo stesso ruolo, ed infatti tramite un'opportuna modifica è possibile ottenere dei blocchi "O" che hanno il compito di fornire unicamente il segnale *generate* e dei blocchi "O" utilizzati come buffer, affinché non si abbiano differenze rilevanti nella propagazione del segnale dal punto di vista temporale.

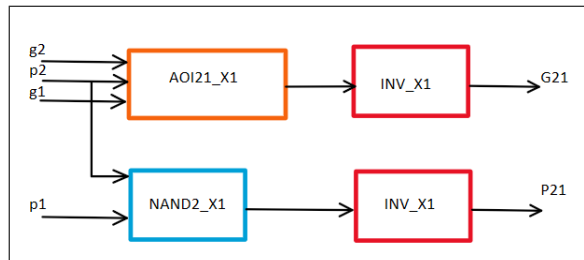


Figura 5: Blocco operatore "O"

3.2 Registri

I registri impiegati nel modulo digitale sono tre, due per campionare gli operandi A e B oltre ai bit di Cin e Sub, uno per campionare il risultato ed il bit di Cout.

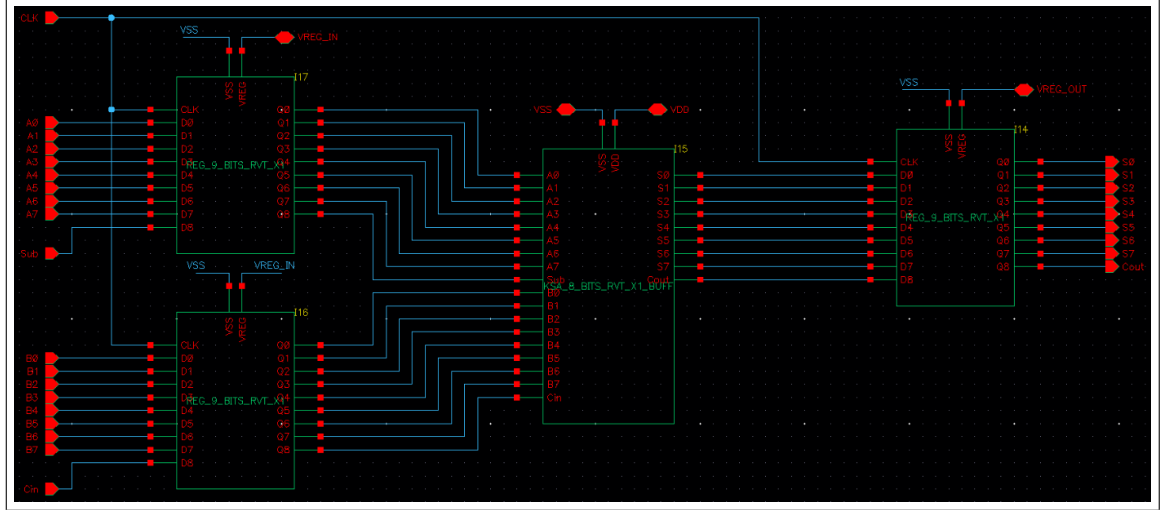


Figura 6: Datapath del sistema digitale che illustra i collegamenti tra registri e rete combinatoria.

3.2.1 Struttura interna

Ciascun registro è realizzato posizionando in parallelo registri statici da 1 bit ($FD1 - RVT - X1$), di cui è già stata fatta precedentemente la caratterizzazione.

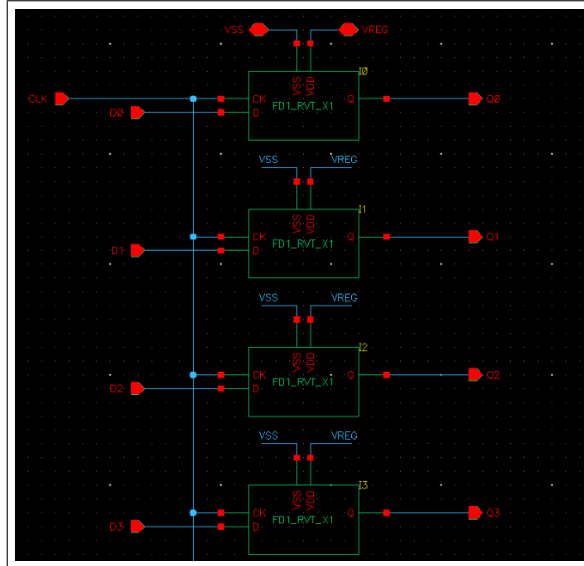


Figura 7: Vista della sezione interna del registro.

4 Operazioni di somma e sottrazione

L'intero sistema digitale, è concepito per effettuare la somma(A+B) o la sottrazione(A-B) tra due operandi. Quest'ultima operazione tuttavia, richiede l'ausilio sia di un blocco XOR (che riceve in ingresso il bit i-esimo corrispondente all'operando "B" e il segnale "Sub"), sia della condizione del segnale Cin = 1 (valore logico alto). Per la sottrazione quindi, ricorriamo al metodo del complemento a 2:

- [1] I numeri *positivi* sono indicati dal bit di segno '0' (MSB) seguito dalla rappresentazione binaria del valore;
- [2] I numeri *negativi* sono indicati con la rappresentazione in complemento a 2 del corrispondente numero positivo ($2^n - N$);

Quando il segnale Sub è pari al valore logico 1 (alto) viene abilitata la differenza(A-B), quando invece vale 0 si verifica la somma(A+B).

5 Caratterizzazione

5.1 Prestazioni dinamiche

5.2 Potenza statica media

I sommatore realizzati con struttura ad albero consentono di avere una latenza molto più bassa rispetto a strutture come il *Carry Bypass* o il *Carry Select*. Tuttavia, si ha inevitabilmente un consumo di potenza statica di gran lunga maggiore.

POTENZA STATICA	Stima [nW]	Simulazione [nW]
Rete combinatoria (RC)	13.64	12.33
Registri (REG)	7.26	////
Totale (RC+REG)	20.9	////

Tabella 1: Stime e simulazioni dei consumi di potenza statica.

5.3 Energia media per ciclo di clock

I dati seguenti corrispondono all'analisi fatta sull'energia media, facendo distinzione tra rete combinatoria e registri.

Questi valori corrispondono alla media aritmetica tra i 4 valori medi di energia simulati in 4 transizioni, in questo caso per le transizioni n.3 e n.4 sono state ripetute le configurazioni di ingresso delle transizioni n.1 e n.2.

Tali analisi vengono effettuate al variare del *coefficiente di attività* α , ovvero al variare della stimolazione dei bit associati agli operandi in ingresso.

Per ciascun valore del coefficiente α , l'energia media è stata calcolata su 4 transizioni.

5.3.1 Valori energetici relativi a rete combinatoria e registri.

α	$E_{media,RC}[fJ]$	$E_{media,REG}[fJ]$	$E_{TOT} = E_{RC} + E_{REG}[fJ]$
1	196.35	153.05	349.43
0.5	88.5	122.21	210.71
0	0.086	98.06	98.14

Tabella 2: Simulazioni dell'energia media considerando $\alpha = 1, 0.5, 0$.

Naturalmente con un coefficiente di attività pari a 0, gli unici moduli che effettivamente consumano energia sono i registri!

5.4 Area occupata

In un sommatore ad albero binario, l'area complessiva occupata dai vari blocchi cresce seguendo un andamento logaritmico, in base al numero N di bit:

$$Area \propto N * \log_2[N] \quad (3)$$

6 Considerazioni finali

Lo sviluppo di un circuito sommatore presenta diverse criticità. In primo luogo è la specifica di progetto che suggerisce il tipo di architettura. Nella fattispecie, ottimizzare la latenza significa avere un circuito sicuramente più articolato e dispendioso in termini di occupazione di area e consumo di potenza.

Considerando le altre architetture, è infatti possibile notare come cambiano le caratteristiche del circuito sommatore al variare del numero di bit(N).

Architettura	modello di ritardo	scelte	stima T_{adder} $N = 8$	Aumento area
Ripple carry	$N \cdot \tau$		$8 \cdot \tau$	
Carry-skip	$[2 \cdot M + N/M] \cdot \tau$	$M = 2$ $N/M = 4$	$8 \cdot \tau$	$N/M = 4$ mux a due vie e circuiti per calcolare BP
Carry select	$[M+2 + N/M] \cdot \tau$	$M = 2$ $N/M = 4$	$8 \cdot \tau$	N mux a 2 vie N blocchi di calcolo Cout
Albero binario	$[2 + \log_2 N] \cdot \tau$		$5 \cdot \tau$	Al posto della catena di propagazione del carry $N \cdot \log_2 N$ operatori o

Figura 8: Confronto tra sommatore con $N=8$ numero di bit impiegati.

Scegliendo $N=8$ bit, si può osservare che dal punto di vista delle stime della latenza non ci sono grandi differenze, fatta eccezione per l'albero binario.

Architettura	modello di ritardo	scelte	stima T_{adder} $N = 32$	Aumento area
Ripple carry	$N \cdot \tau$		$32 \cdot \tau$	
Carry-skip	$[2 \cdot M + N/M] \cdot \tau$	$M = 4$ $N/M = 8$	$16 \cdot \tau$	$N/M = 8$ mux a due vie e circuiti per calcolare BP
Carry select	$[M+2 + N/M] \cdot \tau$	$M = 6$ $N/M = 6$	$14 \cdot \tau$ (36 bit)	N mux a 2 vie N blocchi di calcolo Cout
Albero binario	$[2 + \log_2 N] \cdot \tau$		$7 \cdot \tau$	Al posto della catena di propagazione del carry $N \cdot \log_2 N$ operatori o

Figura 9: Confronto tra sommatore con $N=32$ numero di bit impiegati.

Al contrario, con $N=32$ la differenza tra le tre architetture di tipo carry-lookahead (carry-bypass, carry-select, albero binario) è sempre più marcata. Tale aspetto suggerisce particolare attenzione nello scegliere l'architettura più adatta a seconda della specifica.

Benché la richiesta in termini di bit per questo progetto sia $N=8$ è facile comprendere che anche con N pari a 32 (in generale $N \geq 8$) l'albero binario continui ad essere la scelta più indicata per ottimizzare la latenza.

Tuttavia, si ha un consumo di potenza statica (e di area occupata) notevolmente maggiore rispetto alle altre architetture.