



Politecnico di Torino
III Facoltà di Ingegneria

Sistemi elettronici a basso consumo

Relazioni di laboratorio

Laurea Magistrale in Ingegneria Elettronica
Orientamento: Sistemi Elettronici

Gruppo n. 9

Autori:

Favero Simone, Micelli Federico, Spanna Francesca

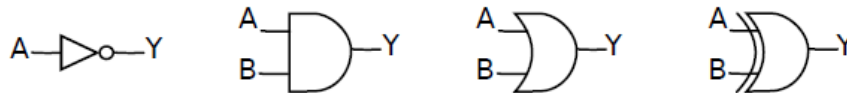
Contents

1	Laboratorio 1 - Power Estimation: probabilistic techniques	2
1.1	Calcolo di probabilità e attività: porte logiche elementari	2
1.2	Calcolo di probabilità e attività: half adder e full adder	5
1.3	Sintesi e analisi di potenza di un RCA	10
1.4	MUX: generazione e propagazione di glitch	14
1.5	Calcolo di probabilità e attività: contatore sincrono	17
2	Laboratorio 2 - FSM Assignment and VHDL Synthesis	22
2.1	FSM State Assignment	22
2.1.1	Progetto	22
2.1.2	Descrizione VHDL e simulazione Modelsim	25
2.2	Sintesi del VHDL	25

Laboratorio 1 - Power Estimation: probabilistic techniques

1.1 Calcolo di probabilità e attività: porte logiche elementari

Il primo esercizio consiste nel valutare le probabilità e attività dell'uscita di quattro porte logiche elementari: NOT, AND, OR e XOR.



Mentre la probabilità di uscita del gate è definita dalla funzione logica stessa, la switching activity è valutata allo stesso modo per tutti i casi, mediante la seguente formula:

$$A = 2 \cdot P1 \cdot (1 - P1)$$

dove $P1$ indica la probabilità che l'uscita assuma valore logico alto.

Di seguito è riportata l'analisi delle porte logiche richieste, considerando ingressi equiprobabili e scorrelati.

- **NOT**

$$P(Y = 1) = 1 - P(A = 1) = 0.5$$
$$A(Y) = 0.5$$

- **AND**

$$P(Y = 1) = P(A = 1) \cdot P(B = 1) = 0.25$$
$$A(Y) = 0.375$$

- **OR**

$$P(Y = 1) = 1 - ((1 - P(A = 1)) \cdot (1 - P(B = 1))) = 0.75$$
$$A(Y) = 0.375$$

- **XOR**

$$P(Y = 1) = P(A = 1) \cdot (1 - P(B = 1)) + P(B = 1) \cdot (1 - P(A = 1)) = 0.5$$
$$A(Y) = 0.5$$

Simulando il test bench fornito tramite ModelSim, è possibile ottenere un file riportante il numero di commutazioni di ogni segnale del circuito durante il tempo di simulazione.

Il testbench fornito sfrutta un generatore di numeri casuali per generare gli ingressi delle porte, rendendo questi ultimi equiprobabili e statisticamente indipendenti.

Sono riportati i seguenti valori:

Tc(CK)	Tc(INV)	Tc(AND)	Tc(OR)	Tc(XOR)
20	1	0	4	4
200	43	40	42	44
2000	533	418	352	470
20000	4916	3606	3784	4876
200000	49967	37834	37541	49939

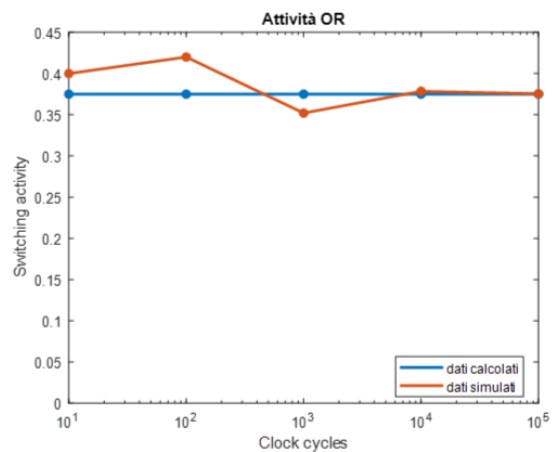
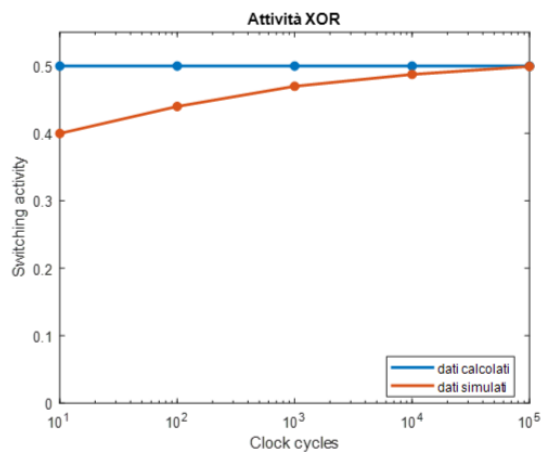
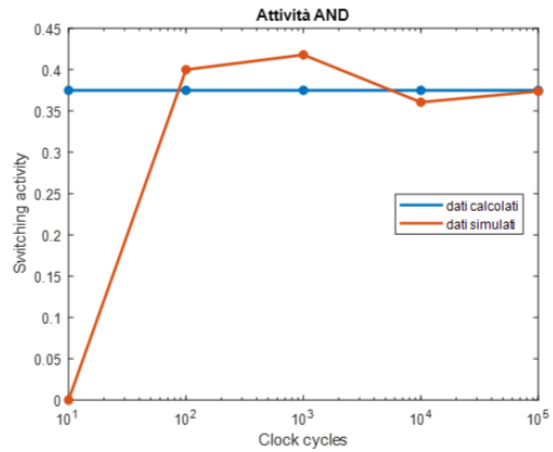
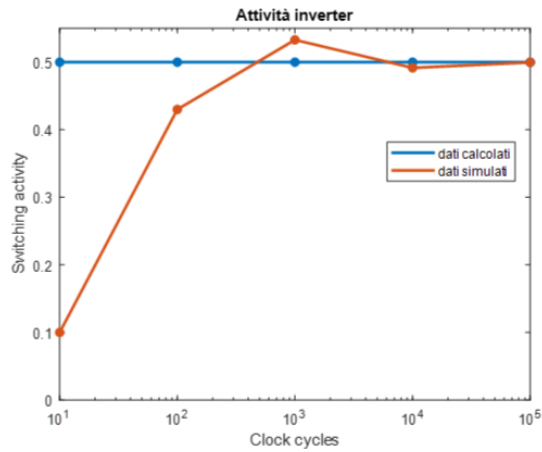
È possibile stimare la switching activity dividendo il numero di commutazioni di un nodo per il numero di colpi di clock della relativa simulazione.

Dal momento che il parametro Tc si riferisce al numero totale di commutazioni, il numero di cicli di clock è ottenuto dividendo per due il parametro Tc(CK).

I risultati dei calcoli sono riportati nella seguente tabella.

Tc(CK)	Esw(INV)	Esw(AND)	Esw(OR)	Esw(XOR)
20	0.1	0	0.4	0.4
200	0.43	0.40	0.42	0.44
2000	0.533	0.418	0.352	0.470
20000	0.4916	0.3606	0.3784	0.4876
200000	0.4997	0.3783	0.3754	0.4994

Per garantire una migliore visualizzazione dei dati ottenuti al variare del tempo di simulazione, sono stati realizzati i seguenti grafici.



Si osserva che, all'aumentare del tempo di simulazione, la stima dell'attività risulta sempre più accurata. In particolare, nel caso analizzato, si osserva che per un numero di cicli di clock superiore a 10000, i dati simulati sono confrontabili con quelli teorici.

1.2 Calcolo di probabilità e attività: half adder e full adder

Dalle tavole di verità di Half Adder e Full Adder si ottengono le seguenti funzioni:

- **Half adder**

$$\begin{aligned} S &= A \text{ XOR } B \\ \text{Cout} &= A \text{ AND } B \end{aligned}$$

- **Full adder**

$$\begin{aligned} S &= A \text{ XOR } B \text{ XOR } \text{Cin} \\ \text{Cout} &= A \text{ AND } B \text{ AND } \text{Cin} \end{aligned}$$

Partendo dalle funzioni logiche che descrivono le uscite è stato possibile ricavare le probabilità associate alle uscite e le relative attività.

- **Half adder**

$$P(S = 1) = P(A = 1) \cdot ((1 - P(B = 1)) + P(B = 1) \cdot (1 - P(A = 1)))$$

$$P(\text{Cout} = 1) = P(A = 1) \cdot P(B = 1)$$

$$A(S) = 2 \cdot P(S = 1) \cdot (1 - P(S = 1))$$

$$A(\text{Cout}) = 2 \cdot P(\text{Cout} = 1) \cdot (1 - P(\text{Cout} = 1))$$

- **Full adder**

$$\begin{aligned} P(S = 1) &= P(A = 1) \cdot (1 - P(B = 1)) \cdot (1 - P(\text{Cin} = 1)) + \\ &+ P(B = 1) \cdot (1 - P(A = 1)) \cdot (1 - P(\text{Cin} = 1)) + \\ &+ P(\text{Cin} = 1) \cdot (1 - P(A = 1)) \cdot (1 - P(B = 1)) + \\ &+ P(A = 1) \cdot P(B = 1) \cdot P(\text{Cin} = 1) \end{aligned}$$

$$\begin{aligned} P(\text{Cout} = 1) &= P(A = 1) \cdot P(B = 1) \cdot (1 - P(\text{Cin} = 1)) + \\ &+ P(\text{Cin} = 1) \cdot P(A = 1) \cdot (1 - P(B = 1)) + \\ &+ P(\text{Cin} = 1) \cdot P(B = 1) \cdot (1 - P(A = 1)) + \\ &+ P(A = 1) \cdot P(B = 1) \cdot P(\text{Cin} = 1) \end{aligned}$$

$$A(S) = 2 \cdot P(S = 1) \cdot (1 - P(S = 1))$$

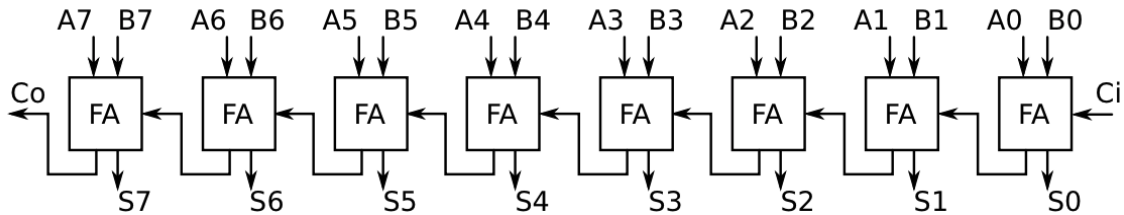
$$A(\text{Cout}) = 2 \cdot P(\text{Cout} = 1) \cdot (1 - P(\text{Cout} = 1))$$

I risultati ottenuti sono riportati nella seguente tabella.

$$P(A) = P(B) = 0.5$$

	P(S=1)	A(S)	P(Cout=1)	A(Cout)
HA	0.5	0.5	0.25	0.375
FA	0.5	0.5	0.5	0.5

Sfruttando i risultati ottenuti per il singolo full adder, è stato possibile ottenere le probabilità e le attività di un ripple carry adder avente parallelismo pari a 8 bit.



Per tale analisi sono stati considerati equiprobabili gli ingressi A e B mentre il carry in ingresso al primo full adder è stato fissato a 0.

I risultati ottenuti sono riportati nella seguente tabella.

$$P(A) = P(B) = 0.5$$

	S7	S6	S5	S4	S3	S2	S1	S0	Cout
P(Y=1)	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.4980
A(Y)	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5

Variando le probabilità associate agli ingressi, in particolare considerando $P(A=1) = 0.4$ e $P(B=1) = 0.6$, si ottengono i seguenti risultati.

$$P(A) = 0.4 \text{ e } P(B) = 0.6$$

	S7	S6	S5	S4	S3	S2	S1	S0	Cout
P(Y=1)	0.52	0.5104	0.5054	0.5028	0.5015	0.5008	0.5004	0.5002	0.4973
A(Y)	0.4992	0.4998	0.4999	0.5	0.5	0.5	0.5	0.5	0.5

È possibile notare come la probabilità delle varie uscite sia funzione delle probabilità degli ingressi. In particolare, la probabilità delle uscite relative alle somme aumenta. Si nota che ai bit relativi agli stadi iniziali sono associati valori che non si discostano molto dal caso di ingressi equiprobabili mentre i bit più significativi subiscono una variazione più marcata.

Al fine di avere un riscontro sui dati stimati, sono state effettuate due simulazioni tramite ModelSim: in un primo caso è stato considerato un ritardo limitato alle uscite relative alle

somme; successivamente è stato introdotto un ulteriore ritardo anche sul carry in uscita di ogni full adder. Come descritto precedentemente, il file fornito da Modelsim riporta il numero di commutazioni dei segnali, di conseguenza l'attività è stata calcolata in modo analogo all'esercizio precedente.

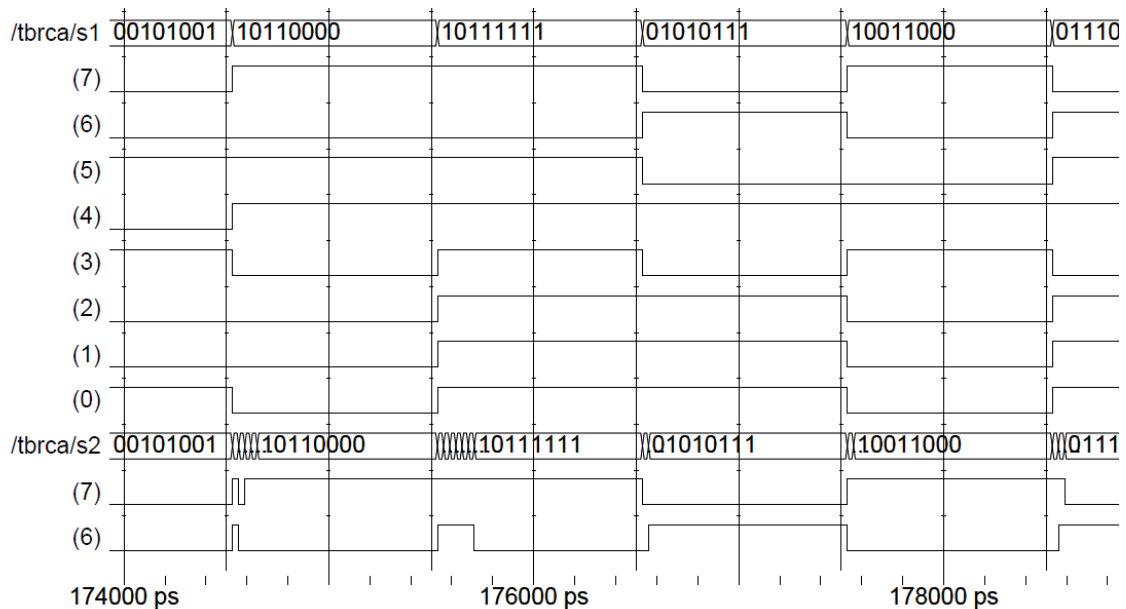
	A(S7)	A(S6)	A(S5)	A(S4)	A(S3)	A(S2)	A(S1)	A(S0)	A(Cout)
Sim. 1	0.42	0.51	0.525	0.425	0.49	0.505	0.51	0.46	0.615
Sim. 2	1.25	1.21	1.065	0.995	1.05	1.005	0.89	0.46	0.615

Si nota come la prima simulazione abbia fornito dati comparabili con quelli precedentemente stimati. Nella seconda simulazione, invece, la presenza di un ritardo sul carry di uscita di ogni full adder ha come conseguenza un generale aumento dell'attività delle uscite di somma. Tale ritardo causa infatti glitch, responsabili dell'aumento della E_{sw} .

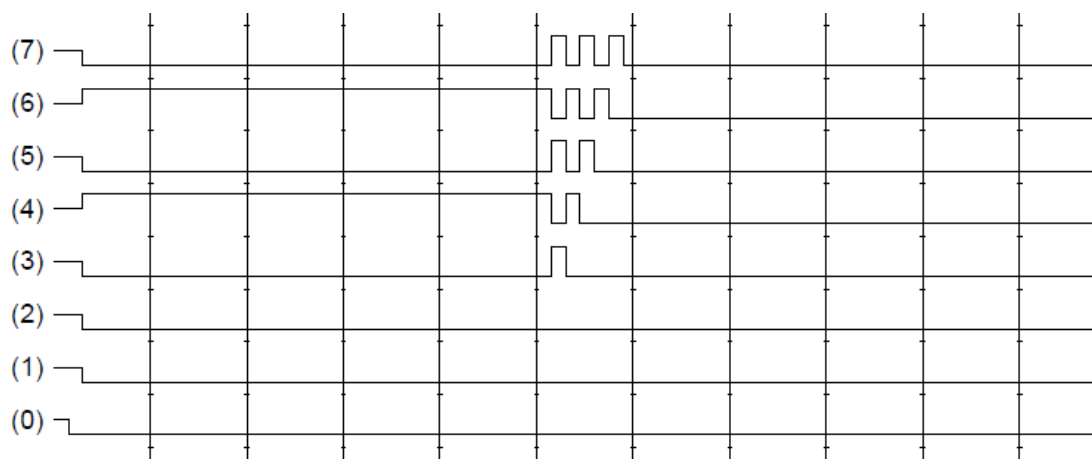
Tale risultato è particolarmente evidente considerando la E_{sw} totale dei due casi: si nota che la presenza del ritardo sui carry out raddoppia tale valore.

$$A(S) = \sum_{i=0}^{N-1} A(S_i)$$

	Sim. 1	Sim. 2
A(S)	3.845	7.925



È possibile notare, attraverso le waveforms, la presenza di commutazioni indesiderate nel secondo caso, rappresentato dal vettore di segnali s2.



In particolare, nell'immagine sopra riportata è possibile osservare come i bit relativi all'uscita di somma, a partire dal quarto, presentino una continua commutazione fino al raggiungimento del valore logico corretto. La causa di tale comportamento è da ricercare nelle transizioni a cui gli ingressi sono stati sottoposti nella simulazione. Le due somme, calcolate una di seguito all'altra, sono le seguenti:

	Somma 1	Somma 2
Addendo 1	10101000	00000100
Addendo 2	10101000	11111100
Risultato	01010000	00000000
Carry in	01010000	

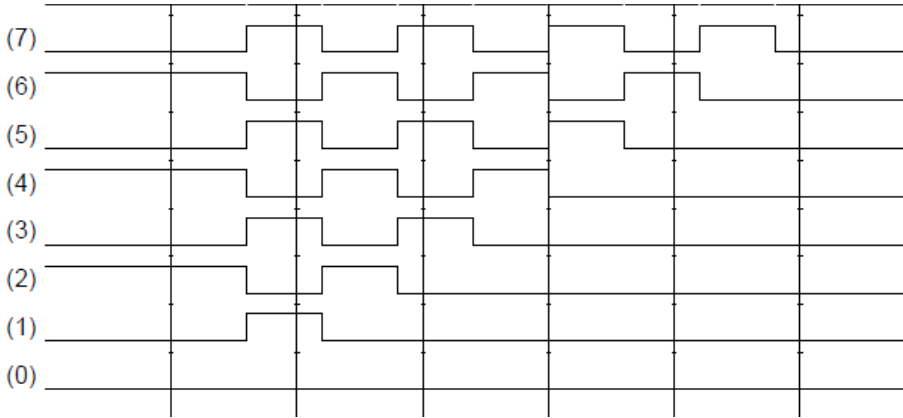
In particolare è possibile osservare che la somma 2 è soggetta ad una propagazione di carry dal terzo bit in poi, inoltre è importante tenere in considerazione i carry in ingresso agli stadi lasciati dal risultato della somma 1, in quanto questi saranno una potenziale fonte di glitch propagandosi anch'essi lungo gli stadi.

In questo caso, la combinazione di carry lasciati dal calcolo precedente e la nuova operazione richiesta (somma 2) fanno sì che il sommatore risulti in una condizione per cui, ad ogni colpo di clock, le uscite degli stadi successivi al secondo continueranno a commutare in modo alternato fino a che il carry introdotto dalla seconda operazione non raggiungerà il suddetto stadio, portando così il bit di somma ad un risultato finale.

Si può notare che i primi 3 bit della somma non sono affetti da tale fenomeno, questi infatti non sono soggetti alla propagazione di carry derivanti dalle due somme. È possibile ricostruire una situazione simile per questi primi bit, ponendo in ingresso un susseguirsi di due somme leggermente differenti dalle precedenti:

	Somma 1	Somma 2
Addendo 1	10101010	00000001
Addendo 2	10101010	11111111
Risultato	01010100	00000000
Carry in	01010100	

In tal modo il fenomeno di propagazione dei carry e delle commutazioni delle uscite di somma è esteso anche per i bit meno significativi. Di seguito è riportato un risultato della simulazione ottenuto ponendo tali valori in ingresso al sommatore.



1.3 Sintesi e analisi di potenza di un RCA

È possibile effettuare una stima di potenza avanzata attraverso la piattaforma Synopsys. Partendo infatti dalla descrizione VHDL del RCA fornito, il software è in grado di compiere una sintesi ottimizzata e valutare parametri del circuito basandosi su una libreria di celle caratterizzate dal punto di vista di area, consumi e ritardi.

Per poter stimare la potenza dinamica è necessario conoscere la frequenza di lavoro del circuito, ottenuta valutando il percorso combinatorio con ritardo maggiore: è possibile valutare tale parametro del circuito attraverso un timing report. Il ritardo massimo riportato è pari a 0.78ns, di conseguenza è stato scelto un periodo di clock pari a 1ns.

Successivamente è possibile procedere con l'analisi relativa alla potenza. Il comando power report consente di ottenere una generica stima della potenza dissipata dal circuito, distinguendo tre contributi:

- **leakage**: consumo statico dovuto alle correnti di leakage dei dispositivi utilizzati
- **internal**: contributo alla potenza dinamica totale dovuto alla corrente di corto circuito
- **switching**: contributo alla potenza dinamica dovuto alle commutazioni dei nodi e alle relative cariche/scariche delle capacità

Nella seguente tabella è riportata la suddivisione dei tre contributi.

Internal Power	Switching Power	Leakage Power	Total Power
16.7429 uW	9.7406 uW	953.4833 nW	27.4370 uW

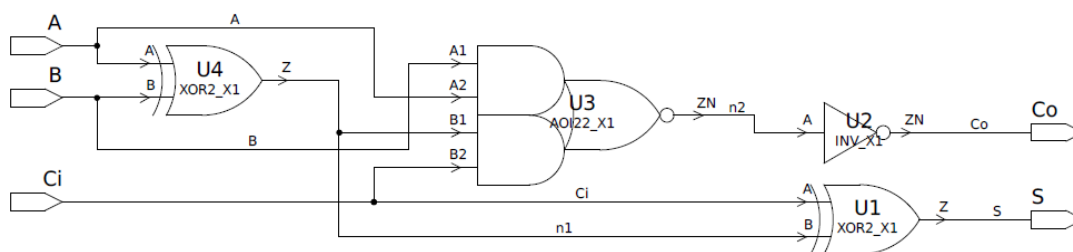
L'apporto maggiore al consumo totale è dato dalla potenza dinamica, in particolare dal consumo della potenza interna.

È interessante notare la suddivisione del consumo in potenza per i vari moduli interni (FA) dell'RCA. È possibile ottenere ciò tramite un power report di tipo gerarchico. La distribuzione dei consumi è riportata nella seguente tabella.

Blocco	Switching Power	Internal Power	Leakage Power	Total Power
FA(8)	0.848 uW	2.154 uW	119.291 nW	3.121 uW
FA(7)	1.293 uW	2.150 uW	118.962 nW	3.562 uW
FA(6)	1.332 uW	2.191 uW	119.340 nW	3.642 uW
FA(5)	1.328 uW	2.171 uW	119.110 nW	3.618 uW
FA(4)	1.278 uW	2.101 uW	119.294 nW	3.498 uW
FA(3)	1.263 uW	2.090 uW	118.879 nW	3.471 uW
FA(2)	1.229 uW	2.002 uW	119.508 nW	3.351 uW
FA(1)	1.171 uW	1.884 uW	119.999 nW	3.175 uW
RCA	9.741 uW	16.743 uW	953.483 nW	27.437 uW

Si nota come la suddivisione risulti simile per tutti i FA fatta eccezione per FA(8). In particolare si differenzia dagli altri per il consumo relativo alla potenza di switch. Si ipotizza che quest'ultima risulti inferiore rispetto agli altri full adder a causa del minor carico di uscita.

È inoltre possibile approfondire l'analisi di potenza considerando i contributi delle celle interne ai singoli moduli. In particolare, vengono messi in luce i contributi delle celle che compongono i singoli FA, sempre suddivisi nei tre contributi precedentemente descritti. Per comprendere l'origine del consumo inferiore del FA8, quest'ultimo è stato analizzato come cella ed è stato confrontato con FA1. Lo schema del circuito interno e i risultati ottenuti sono riportati di seguito.



FA(1)

Cella	Switching Power	Internal Power	Leakage Power	Total Dynamic Power
U4	0.5488 uW	0.5899 uW	36.1637 nW	1.138 uW
U3	0.1749 uW	0.3374 uW	32.5747 nW	0.512 uW
U2	0.3964 uW	0.1377 uW	14.2499 nW	0.534 uW
U1	0.0512 uW	0.8205 uW	36.0111 nW	0.872 uW

FA(8)

Cella	Switching Power	Internal Power	Leakage Power	Total Dynamic Power
U4	0.5365 uW	0.5757 uW	36.1637 nW	1.112 uW
U3	0.2155 uW	0.4278 uW	32.7466 nW	0.643 uW
U2	0.0332 uW	0.1774 uW	14.2174 nW	0.211 uW
U1	0.0624 uW	0.9733 uW	36.1631 nW	1.036 uW

La differenza nel consumo di potenza di switching è individuata nella cella U2, legata al carry in uscita dal singolo FA.

È possibile ottenere informazioni aggiuntive sul calcolo della switching power relativa ai nodi interni ad un modulo con un'analisi di tipo net verbose. In tale analisi sono riportati i contributi capacitivi necessari al calcolo di tale potenza, espressi nodo per nodo.

FA(1)

Nodo	Total Net Load	Static Probability	Toggle Rate	Switching Power
n1	4.694 fF	0.493	0.1932	0.5488 uW
n2	2.010 fF	0.488	0.1439	0.1749 uW
S	0.310 fF	0.507	0.2735	0.0512 uW
Cout	4.554 fF	0.512	0.1439	0.3964 uW

FA(8)

Nodo	Total Net Load	Static Probability	Toggle Rate	Switching Power
n1	4.694 fF	0.499	0.1889	0.5365 uW
n2	2.010 fF	0.484	0.1772	0.2155 uW
S	0.310 fF	0.497	0.3328	0.0624 uW
Cout	0.310 fF	0.516	0.1772	0.0332 uW

Si nota come il nodo legato al carry in uscita presenti una capacità inferiore per FA8. Ciò conferma l'ipotesi legata al carico inferiore.

Analizzando i vari FA con analisi di tipo net verbose si nota inoltre che i contributi di capacità relativi ai nodi della somma (S) siano trascurabili rispetto agli altri contributi. Di conseguenza si evince che il contributo di potenza relativo a tale nodo è di un ordine di grandezza inferiore rispetto agli altri.

Da un report di tipo net verbose è inoltre possibile confrontare i valori di static probability e toggle rate dei nodi di uscita di un FA derivanti dalla simulazione con le probabilità in uscita e le attività stimate precedentemente in modo teorico.

Si nota come i valori di probabilità dei nodi di uscita, S e Cout, siano distribuiti attorno al valore 0.5, dunque confrontabili con il caso precedentemente considerato di ingressi equiprobabili.

Il toggle rate rappresenta la frequenza media di commutazione associata al singolo nodo, che identifica, con il suo reciproco, il periodo medio con cui tale nodo presenta una commutazione: attraverso tale valore è possibile stimare la switching activity dei nodi di somma e dei carry in uscita. In tal caso, poichè il periodo di clock è pari a 1ns, il valore di toggle rate riportato coincide con la switching activity del segnale. E' possibile notare come i valori simulati si distanzino da quelli teorici, in quanto all'interno del metodo probabilistico non viene tenuta in considerazione la correlazione spaziale e temporale tra diversi segnali. Tale fatto mette in luce, di conseguenza, le limitazioni che l'utilizzo di un modello probabilistico comporta.

CONFRONTO VERBOSE RCA

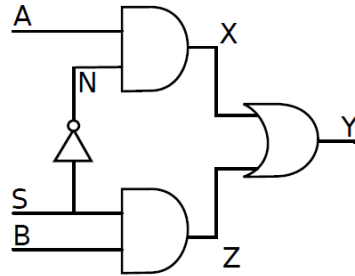
E' stato richiesto inoltre un power report di tipo net verbose relativo all'intero RCA. Si può osservare come i contributi capacitivi relativi ai nodi delle uscite di somma siano significativamente inferiori rispetto ai contributi dei nodi relativi ai carry tra uno stadio e quello successivo. Anche in questo caso la motivazione è dovuta al carico capacitivo inferiore delle uscite di somma, che non risultano collegate ad ulteriori blocchi. Tuttavia, osservando i contributi totali riportati dall'analisi di tipo verbose, è possibile osservare una differenza tra questi e quelli presenti nel power report precedentemente richiesto.

	Internal Power	Switching Power	Leakage Power	Total Power
Power report	16.7429 uW	9.7406 uW	953.4833 nW	27.4370 uW
Net Verbose	8.848 uW	3.766 uW	403.098 nW	12.614 uW

I contributi riportati dall'analisi di tipo net verbose risultano proporzionalmente inferiori. La possibile causa di ciò è da attribuirsi al tipo di analisi effettuata da Synopsys: in particolare la keyword -net utilizzata ha richiesto uno studio della potenza relativo ai nodi interni dell'architettura dell'RCA, presenti tra i vari FA connessi. In tale analisi i contributi dei nodi interni ai FA stessi sono stati tralasciati. Al contrario un'analisi di tipo power report fornisce un risultato completo comprendendo all'interno l'analisi di tutti i nodi dei blocchi che compongono l'architettura.

1.4 MUX: generazione e propagazione di glitch

L'obiettivo di questo esercizio è lo studio delle conseguenze introdotte dai ritardi delle porte, in particolare all'interno del multiplexer rappresentato nella seguente figura.



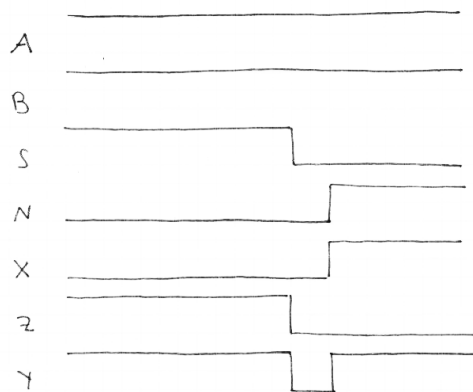
In questo caso particolare tutte le porte sono esenti da ritardi fatta eccezione per l'inverter, caratterizzato da un ritardo di propagazione pari a 0.1 ns.

All'interno del file *tb_mux21_glitch.vhd* è stato possibile identificare la combinazione dei segnali di ingresso con i quali il multiplexer è stato testato.

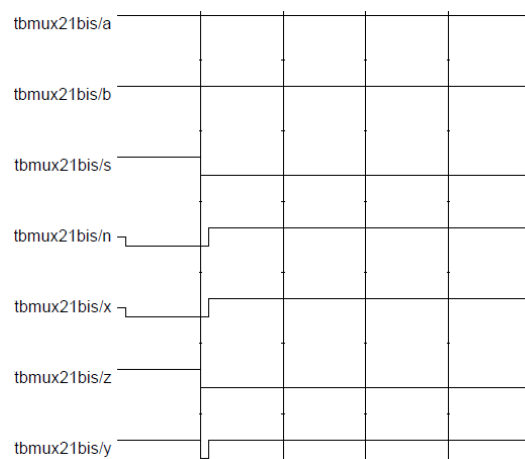
```
A <= '1';
B <= '1';
S <= '1', '0' after 1 ns;
```

In particolare, è possibile notare dal codice VHDL sopra riportato che inizialmente gli ingressi assumono tutti un valore logico alto. Dopo 1ns il segnale S commuta. L'uscita, in un caso ideale, non dovrebbe presentare commutazioni.

Tuttavia, si ipotizza che, a causa del ritardo di propagazione introdotto dall'inverter, vi sia un intervallo temporale in cui i nodi interni X e Z assumono entrambi un valore logico basso, portando quindi l'uscita Y a 0. Tale comportamento atteso è mostrato nella seguente figura.



Per mezzo di una simulazione ModelSim è stato possibile osservare attraverso le waveforms il comportamento reale dei segnali. Il risultato di tale simulazione è riportato nella seguente immagine.



È possibile visualizzare direttamente sulla waveform dell'uscita Y il glitch causato dall'inverter. Tale comportamento è in linea con quanto atteso.

Al fine di analizzare altre possibili combinazioni degli ingressi che possano generare un glitch in uscita è stata realizzata una mappa di Karnaugh rappresentante la funzione logica del multiplexer.

S \ AB	00	01	11	10
0	0	0	1	1
1	0	1	1	0

In particolare, è possibile notare che il circuito precedentemente riportato sia l'implementazione della funzione logica coperta da due implicanti, rappresentata nella figura precedente.

Il glitch analizzato è dovuto ad una transizione degli ingressi che consegue in un passaggio tra un implicante e l'altro. Tale problema potrebbe essere risolto inserendo, in modo ridondante, un terzo implicante per evitare la transizione precedentemente trattata.

S \ AB	00	01	11	10
0	0	0	1	1
1	0	1	1	0

Tale ragionamento porta alla conclusione che l'unica combinazione in grado di causare un glitch in uscita sia quella presa in analisi.

Essendo i glitch associati a commutazioni di nodi, questi portano un contributo aggiuntivo al consumo totale di potenza dinamica. In particolare l'energia sprecata durante queste commutazioni spurie è la somma dei consumi durante le due transizioni del segnale. Ciascuna delle due contribuisce all'energia secondo la seguente equazione:

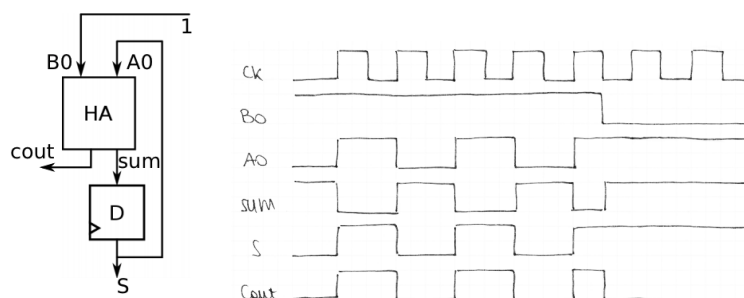
$$E = C \cdot V^2$$

dove C rappresenta la capacità di carico e V la tensione a cui viene caricata tale capacità. Metà di tale contributo di energia è usata per caricare o scaricare la capacità di carico associata all'uscita, mentre la restante parte viene dissipata. Di conseguenza il consumo di energia totale associato ad un glitch è dato da:

$$E = 2 \cdot C \cdot V^2$$

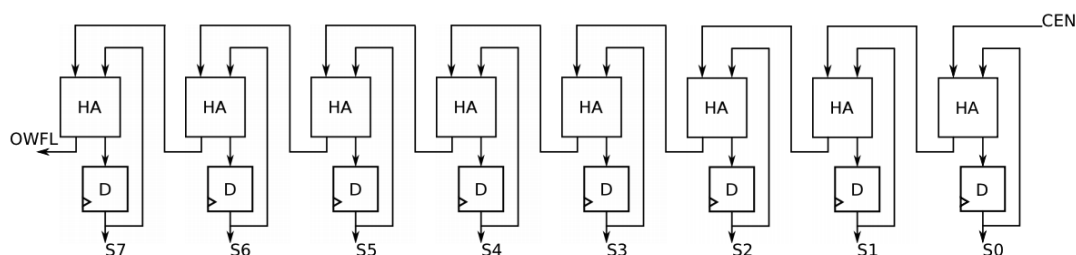
1.5 Calcolo di probabilità e attività: contatore sincrono

Nella prima parte di questo esercizio è stato analizzato il timing del blocco rappresentato nella figura seguente.

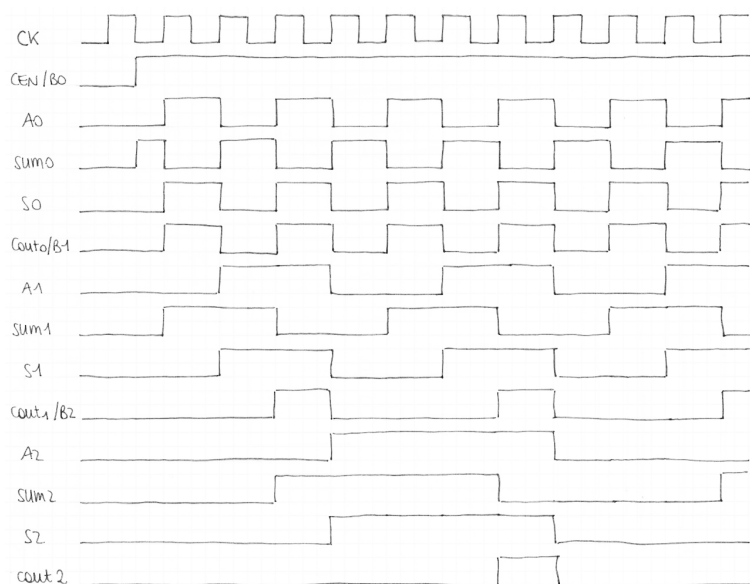


In particolare è stato possibile stabilire il suo comportamento atteso. Quando il segnale B0 è asserito, l'uscita S presenta una commutazione per ogni fronte sensibile del clock. Al contrario, quando B0 non è attivo, l'uscita non presenta alcuna commutazione.

È possibile utilizzare tali strutture per realizzare un contatore sincrono, come mostrato in figura. In particolare, i segnali di CEN e OVFL rappresentano rispettivamente l'enable del contatore e il terminal count di questo.



Considerando infatti il comportamento delle uscite S0, S1 e S2 quando il segnale di enable è attivo si ottiene il seguente timing.



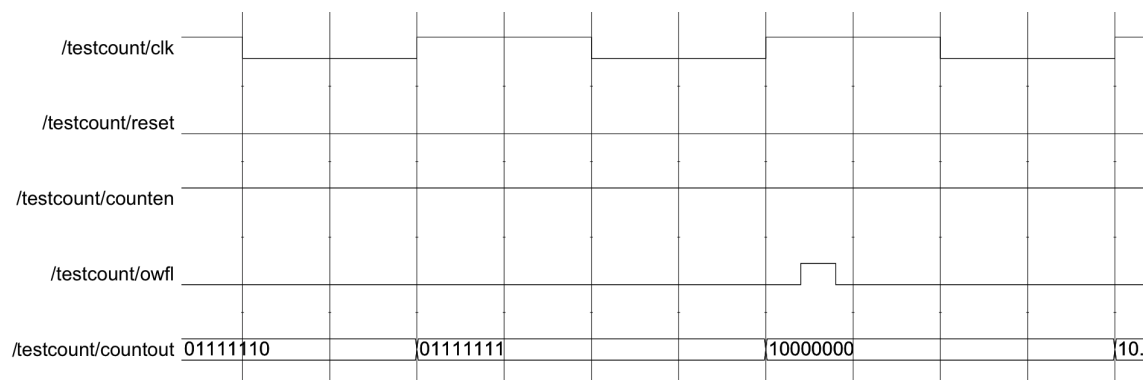
Dalle considerazioni fatte in precedenza è possibile stimare il numero di commutazioni delle 8 uscite e del terminal count durante un intero ciclo di conta. Inoltre, nella tabella seguente sono stati aggiunti i dati ottenuti tramite power report, ottenuto attraverso una simulazione ModelSim. Il clock utilizzato per la simulazione ha un periodo di 2ns.

Segnale	Transizioni stimate	Transizioni simulate
S0	255	257
S1	127	128
S2	63	64
S3	31	32
S4	15	16
S5	7	8
S6	3	4
S7	1	2
OVFL	1	16

Si può notare che, senza considerare il segnale OVFL, i risultati ottenuti in entrambi i casi siano confrontabili, tuttavia è presente una leggera differenza, pari a 1 o 2 colpi di clock, dovuta al fatto che è stato scelto un tempo di simulazione leggermente superiore al tempo di conta totale.

Riguardo al segnale di OVFL, il risultato simulato è distante da quello atteso, in quanto il segnale presenta un numero di commutazioni decisamente maggiore. Tale segnale, al contrario di quelli di uscita della somma (rappresentanti il conteggio), non presenta un elemento sequenziale che lo interfacci con l'esterno. Per tale motivo sono ben visibili tutte le commutazioni spurie del segnale stesso, dovute ai ritardi di propagazione definiti all'interno dei vari stati del contatore.

Dalla simulazione è stato inoltre possibile osservare l'andamento delle waveforms delle uscite, riportato nella figura sottostante.



Si nota che, in un preciso intervallo temporale, il segnale OVFL assume un valore logico alto durante il conteggio, comportamento inatteso da parte di un contatore. Tale glitch avviene in seguito alla commutazione del bit più significativo (S7); infatti, per un certo intervallo di tempo, in ingresso all'half adder del blocco 7 sono presenti due ingressi con valore logico alto: uno dovuto al carry in uscita dal blocco precedente che, a causa dei ritardi, non ha ancora commutato il suo valore, e l'altro relativo all'uscita attuale S7.

Successivamente, sempre attraverso un power report, è stato possibile analizzare l'attività dei segnali interni al contatore. I risultati sono riportati nella seguenti tabelle.

Segnale	Commutazioni	Attività	Segnale	Commutazioni	Attività
SUM(0)	258	0.992	Cout(0)	257	0.988
SUM(1)	385	1.377	Cout(1)	256	0.985
SUM(2)	320	1.231	Cout(2)	192	0.738
SUM(3)	224	0.862	Cout(3)	128	0.492
SUM(4)	144	0.554	Cout(4)	80	0.308
SUM(5)	88	0.338	Cout(5)	48	0.185
SUM(6)	52	0.200	Cout(6)	28	0.108
SUM(7)	30	0.115	Cout(7)	16	0.062

È possibile notare come l'attività delle somme in uscita dagli half adder sia differente rispetto alle uscite sincrone dei flip-flop. Infatti, a causa dei ritardi di propagazione dei carry in uscita dai singoli stadi, si generano una serie di glitch, che, propagandosi lungo la serie di half adder, causano commutazioni multiple dei nodi SUM.

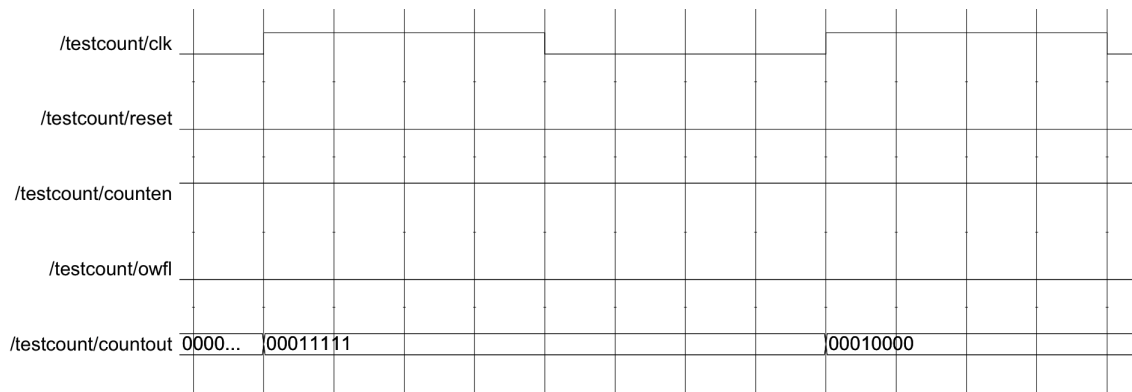
Similmente all'analisi effettuata per il blocco 7 è possibile ipotizzare comportamenti simili legati al carry in uscita anche per gli altri blocchi. Tale vettore di carry presenta infatti un'attività complessiva maggiore rispetto a quella stimata nel caso senza ritardi.

Nella simulazione precedentemente effettuata i ritardi sono stati modellizzati in modo tale da presentare un valore significativamente inferiore rispetto al periodo di clock.

$$Periodo = 2ns \quad Ritardo_somma = 0.2ns \quad Ritardo_carry = 0.2ns$$

Si nota che in questo caso il periodo di clock è sufficientemente elevato per garantire un corretto funzionamento del contatore: la propagazione del carry out può avvenire correttamente in tutti gli stadi.

Riducendo il periodo di clock a 0.8ns è possibile osservare un comportamento inatteso, rappresentato in figura, da parte del contatore.



L'immagine mette in luce una transizione da un conteggio all'altro:

$$00011111 \rightarrow 00010000$$

Tale errore è dovuto alla mancata completa propagazione del carry in tutti gli stadi, infatti il clock campiona le uscite dei flip-flops prima che tutti gli stadi abbiano raggiunto un valore stabile.

La presenza di glitch all'interno del contatore in analisi comporta un incremento dei consumi rispetto al caso ideale. Una possibile soluzione, che potrebbe essere adottata per garantire un numero inferiore di commutazioni spurie modificando il VHDL, è legata all'aumento del ritardo coinvolto nel calcolo della somma per i singoli HA. Se tale tempo fosse infatti maggiore o uguale del massimo tempo di propagazione del carry attraverso tutti gli stadi, le uscite SUM non presenterebbero più un elevato numero di glitch.

FORSE CI STAREBBE VERIFICARE CON UNA SIMULAZIONE E NEL CASO METTERE I RISULTATI

COSE DA FARE:

- Capire il toggle rate nel verbose (ESW) IPOTESI FATTE
- Capire l'incongruenza del verbose RCA FATTO
- Spiegare il worst case del sommatore FATTO
- Fare mappa di Karnaugh FATTO

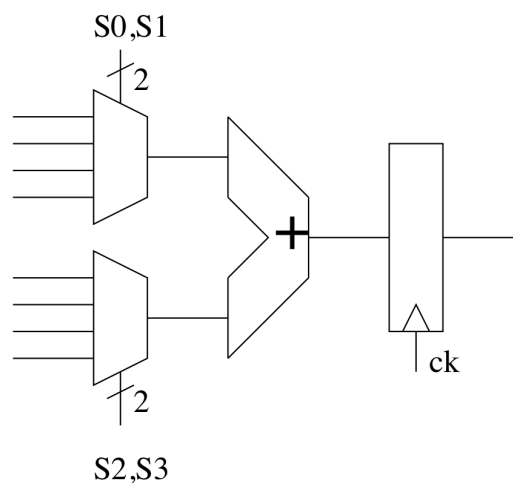
- Plot segnali MUX (fatto da noi) FATTO
- Plot dei segnali del blocchetto contatore e dei primi bit del contatore (fatto da noi) FATTO
- Completare le tabelle FATTO
- Revisione grafica FATTO

Laboratorio 2 - FSM Assignment and VHDL Synthesis

2.1 FSM State Assignment

2.1.1 Progetto

L'obiettivo del progetto è la definizione di una macchina a stati al fine di realizzare un algoritmo di somma tra 6 input, utilizzando il datapath in figura seguente e scegliendo l'ordine con cui collegare i diversi ingressi ai multiplexer.



In particolare, al fine di minimizzare i consumi della macchina a stati, è necessario ridurre al minimo il numero di commutazioni di:

- variabili di stato
- segnali di controllo

Essendo i segnali di controllo coincidenti con i selettori dei multiplexer, è stata innanzitutto stabilita una sequenza per la selezione dei diversi input tale per cui il numero di commutazioni durante l'esecuzione dell'intero algoritmo di somma fosse minimo.

Operazione	S3	S2	S1	S0
A+B	0	1	0	1
S+C	0	0	0	0
S+D	0	0	1	0
E+S	1	0	1	1
F+S	1	1	1	1

È possibile notare come sia necessario riportare il valore della somma, denominato S, in ingresso ad entrambi i multiplexer.

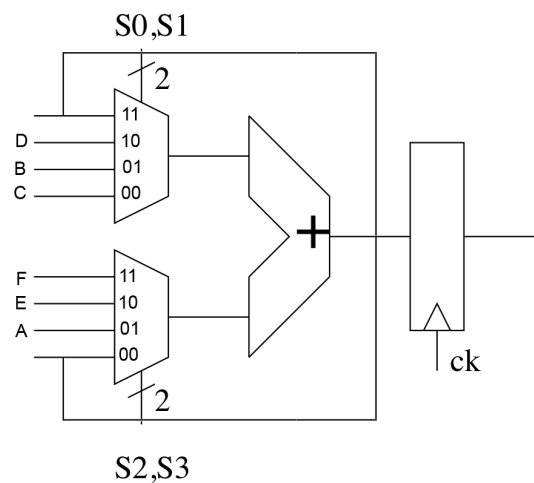
Nella transizione da uno stato all'altro il numero di commutazioni è pari alla variazione degli addendi. Ne consegue che il massimo numero di commutazioni durante una transizione è pari a 2.

Considerando un ciclo di esecuzione dell'algoritmo, è possibile stimare il numero totale di commutazione sui segnali di selezione e la loro switching activity.

	S3	S2	S1	S0
Commutazioni in un ciclo	2	2	2	2
Attività	0.4	0.4	0.4	0.4

Il numero di commutazioni totale risulta quindi pari a 8.

È stata scelta dunque la seguente configurazione degli ingressi



Il passo successivo è la scelta della codifica dei singoli stati. In tale contesto è importante tenere in considerazione che una generica FSM presenta una rete per la generazione degli output e una rete per il calcolo dello stato futuro.

Una codifica ottimale punta alla minimizzazione delle commutazioni in entrambe le reti. Per tale scopo è stato scelto di far coincidere le due reti, in modo che la logica utilizzata

per generare le uscite potesse essere utilizzata anche per la definizione dello stato futuro. In particolare, la codifica dello stato futuro si ottiene a partire dalle uscite di controllo dello stato attuale.

Operazione	Codifica dello stato	S3	S2	S1	S0
A+B	1 1 1	0	1	0	1
S+C	1 0 1	0	0	0	0
S+D	0 0 0	0	0	1	0
E+S	0 1 0	1	0	1	1
F+S	0 1 1	1	1	1	1

È possibile notare come tutte le transizioni presentino una distanza di Hamming pari a 1, fatta eccezione della transizione da S+C a S+D, nella quale tale distanza è pari a 2.

In tale contesto, la sola logica combinatoria necessaria è relativa alla valutazione delle uscite a partire dalla codifica dello stato attuale.

Si è notato che tali funzioni logiche presentano un'implementazione semplice con un numero limitato di porte. [Di seguito sono riportate le mappe di Karnaugh per le funzioni logiche sopra descritte](#)

S3				
C2,C1 C0	00	01	11	10
0	0	1	-	-
1	-	1	0	0

S2				
C2,C1 C0	00	01	11	10
0	0	0	-	-
1	-	1	1	0

S1				
C2,C1 C0	00	01	11	10
0	1	1	-	-
1	-	1	1	0

S0				
C2,C1 C0	00	01	11	10
0	0	1	-	-
1	-	1	1	0

È possibile notare che le funzioni logiche presentano implicantì in comune, semplificando

maggiormente la logica necessaria.

2.1.2 Descrizione VHDL e simulazione Modelsim

La macchina a stati precedentemente definita è stata descritta attraverso il VHDL, specificando in particolare la codifica degli stati progettata.

DOBBIAMO METTERE IL VHDL? fare eventuale appendice

Al fine della simulazione, datapath e FSM sono stati inclusi in un'unica entity per ottenere il Processing Element complessivo. È stato dunque realizzato un testbench per verificare il corretto susseguirsi degli stati e le relative transizioni dei segnali. In particolare, è stata effettuata una simulazione di 2000 colpi di clock, che ha fornito i seguenti risultati attraverso un power report.

Segnale	Tc	Activity
Clock	4000	2
S3	799	0.3995
S2	802	0.4010
S1	799	0.3995
S0	802	0.4010

Le attività ottenute sono comparabili con quelle stimate in fase di progetto.

2.2 Sintesi del VHDL

In tale sezione è stata effettuata la sintesi della top level entity che comprende sia FSM sia datapath.

Inizialmente i file sono stati analizzati nel corretto ordine ed è stata elaborata la struttura del Processing Element.

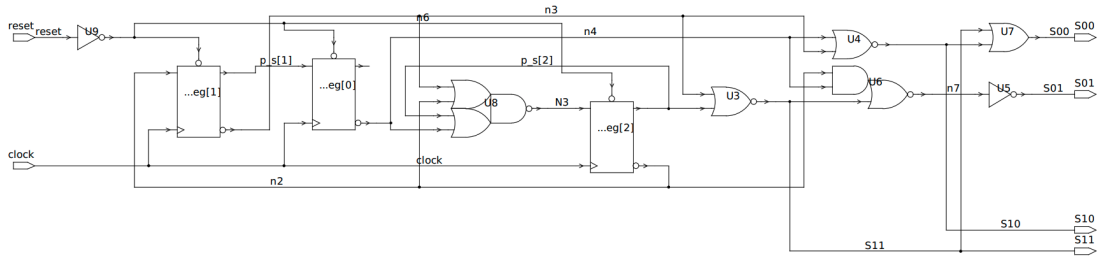
Successivamente è stato generato un clock di periodo pari a 10 ns.

create_clock -name "CLK" -period 10 CLK

È stata verificata la presenza del clock generato attraverso il comando **report_clock**.

Successivamente è stata effettuata la sintesi attraverso la direttiva **compile -exact map**.

In seguito è stato possibile visualizzare, in modo gerarchico, la mappatura del circuito con la libreria scelta. Nell'immagine seguente, in particolare, è riportato lo schematico della macchina a stati.



È possibile notare, come atteso, che il numero di porte logiche utilizzato risulta limitato. Sono stati effettuati una serie di report inizialmente sulla top level entity per poi analizzare in modo più approfondito la macchina a stati. In primo luogo è stata ottenuta una stima dell'area delle celle utilizzate nella sintesi, attraverso il comando **report_area**.

Total cell area = 295.53 um²

Successivamente, al fine di valutare lo slack, sono stati analizzati i 10 percorsi combinatori con ritardi maggiori.

report_timing -nworst 10

Per semplicità sono riportati i dati relativi al critical path.

Clock period	10 ns
Data required time	9.97 ns
Data arrival time	1.94 ns
Slack	8.02 ns

Come è possibile notare, lo slack risulta particolarmente elevato se confrontato con il periodo di clock. Nel report sono inoltre presenti i ritardi relativi agli elementi compresi tra lo start-point e l'end-point del percorso critico.

È possibile ottenere informazioni riguardo alla potenza dissipata attraverso il comando **report_power**. In particolare, si evidenziano i tre contributi relativi alla top level entity.

Internal Power	29.4711 uW
Switching Power	13.2924 uW
Leakage Power	5.4747 uW
Total Power	48.2382 uW

Sono stati ottenuti dati aggiuntivi relativi al consumo di potenza dei singoli moduli dell'architettura, tramite l'aggiunta della direttiva **-hier**.

FSM	4.504 uW	9.3%
Datapath	43.734 uW	90.7%

Il consumo della macchina a stati rappresenta una percentuale ridotta rispetto al valore totale.

Al fine di approfondire lo studio relativo ai nodi interni del circuito, è stata realizzata un'analisi di tipo **-net**. In particolare sono riportati i risultati relativi ai nodi dei segnali di selezione dei multiplexer.

Segnale	Net Load [fF]	Static Probability	Toggle Rate	Switching Power [uW]
sel11_wire	13.792	0.120	0.0246	0.2055
sel01_wire	10.222	0.253	0.0272	0.1682
sel10_wire	8.489	0.720	0.0301	0.1547
sel00_wire	7.058	0.720	0.0301	0.1286

Il periodo medio di variazione dei selettori, ottenibile dal toggle rate, è confrontabile con le variazioni medie che avvengono su tali segnali in un ciclo di esecuzione. È importante tenere in considerazione la possibile presenza di commutazioni non previste a causa dell'attivazione del segnale di reset durante la simulazione, il quale è caratterizzato da una probabilità non nulla di commutazione.

È possibile imporre dei vincoli sui vari parametri coinvolti nel processo di sintesi. In particolare, è stato generato un segnale di clock alla massima frequenza consentita dal circuito. La stima di tale frequenza è stata effettuata a partire dall'analisi del critical path precedentemente riportato.

Clock period	1.94 ns
Data required time	1.91 ns
Data arrival time	1.91 ns
Slack	0 ns

Come atteso, lo slack relativo al percorso critico risulta essere nullo, di conseguenza il circuito opera alla massima frequenza supportata. Dal momento che la potenza dissipata è funzione della frequenza di lavoro, è stato riscontrato un aumento dei consumi.

Internal Power	151.3501 uW
Switching Power	69.4127 uW
Leakage Power	5.5735 uW
Total Power	226.3363 uW

A seguito di un incremento della frequenza di un fattore di circa 5, la potenza è aumentata proporzionalmente.

È inoltre possibile imporre un vincolo sul contributo totale di potenza dinamica, attraverso la direttiva

set_max_dynamic_power

tale valore è stato fissato a 40 uW. Si nota che tale valore è leggermente inferiore rispetto alla potenza dinamica ottenuta nelle prime simulazioni.

In particolare, è stato considerato il clock di partenza di periodo pari a 10 ns. Di seguito sono riportati i risultati ottenuti.

	No constraints	Power constraints
Internal Power	29.4711 uW	33.8116 uW
Switching Power	13.2924 uW	13.3092 uW
Total Dynamic Power	42.76 uW	47.12 uW

I risultati ottenuti dalla sintesi non risultano in linea con quanto atteso, infatti è registrato un aumento del consumo di potenza dinamica.

Si ipotizza che il consumo di potenza dinamica ottenuto nella prima sintesi fosse già prossimo al minimo globale.

Successivamente è stata effettuata un'analisi più dettagliata della FSM accedendovi tramite la direttiva **current_instance FSM**.

Attraverso il comando **report_fsm** è stato possibile verificare la corretta implementazione della codifica progettata.

È possibile risalire alla suddivisione del consumo della potenza dinamica all'interno delle celle che costituiscono la FSM attraverso un'analisi di potenza di tipo **-cell**. In particolare, nella seguente tabella vengono riportati i tre contributi maggiori relativi al consumo di potenza dinamica.

ps_reg[1]	1.143 uW
ps_reg[2]	0.979 uW
ps_reg[0]	0.924 uW

Si nota che tali contributi sono relativi alle variabili di stato, che presentano numerose commutazioni durante l'esecuzione dell'algoritmo.

Uno studio più approfondito sulla switching power relativa ai nodi interni è ottenuta tramite un'analisi di tipo **-net**. In particolare, nella seguente tabella, sono riportati i dati relativi ai segnali di uscita della FSM, ovvero i segnali di selezione dei multiplexer.

Segnale	Net Load [fF]	Static Probability	Toggle Rate	Switching Power [uW]
S11	13.792	0.120	0.0246	0.2055
S01	10.222	0.253	0.0272	0.1682
S10	8.489	0.720	0.0301	0.1547
S00	7.058	0.720	0.0301	0.1286

È possibile notare che tali valori coincidono con quelli ottenuti precedentemente durante l'analisi -net della top level entity. Il contributo maggiore del consumo è dovuto alle capacità di carico dei nodi in analisi, che assumono valori significativi rispetto agli altri nodi. Questo è il principale motivo per cui tali segnali contribuiscono maggiormente, data la parità di toggle rate.

Per ragioni di praticità, tali sintesi sono state effettuate raggruppando i comandi e le direttive in uno script in modo darendere il processo automatico. Di seguito è riportato uno degli script utilizzati.

```
gui_start
analyze -library WORK -format vhd
{/home/lp20.9/Desktop/low-power-2020/Lab-2/codifica_3bit_bis/syn/dpadder.vhd
/home/lp20.9/Desktop/low-power-2020/Lab-2/codifica_3bit_bis/syn/fsm1.vhd
/home/lp20.9/Desktop/low-power-2020/Lab-2/codifica_3bit_bis/syn/FSM.Adder.vhd}
elaborate FSMADDER -architecture BEHAVIOR -library WORK
change_selection -name __slctBus18 -type {cell design} _sel426
create_clock -name "CLK" -period 10 {CLK}
compile -exact_map
report_timing -nworst 10 > report/simple_clk_10/report_clock.txt
report_area > report/simple_clk_10/report_area.txt
report_power > report/simple_clk_10/report_power.txt
report_power -hier > report/simple_clk_10/report_power_hier.txt
report_power -net > report/simple_clk_10/report_power_net.txt
current_instance FSM
report_power -hier > report/simple_clk_10/report_power_hier_FSM.txt
report_power -net > report/simple_clk_10/report_power_net_FSM.txt
report_fsm > report/simple_clk_10/report_fsm_encoding.txt
```