



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

<b>1</b>	<b>Laboratorio 1</b>	<b>2</b>
1.1	Calcolo di probabilità e attività: porte logiche elementari . . . . .	2
1.2	Calcolo di probabilità e attività: half adder e full adder . . . . .	4
1.3	MUX: generazione e propagazione di glitch . . . . .	5
1.4	Calcolo di probabilità e attività: contatore sincrono . . . . .	6

# Laboratorio 1

## 1.1 Calcolo di probabilità e attività: porte logiche elementari

Il primo esercizio consiste nel valutare le probabilità e attività 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:

$$A = 2 \cdot P1 \cdot (1 - P0)$$

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.

In particolare, è possibile ricavare la switching activity delle uscite dividendo il numero di commutazioni per il numero di cicli di clock simulati.

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

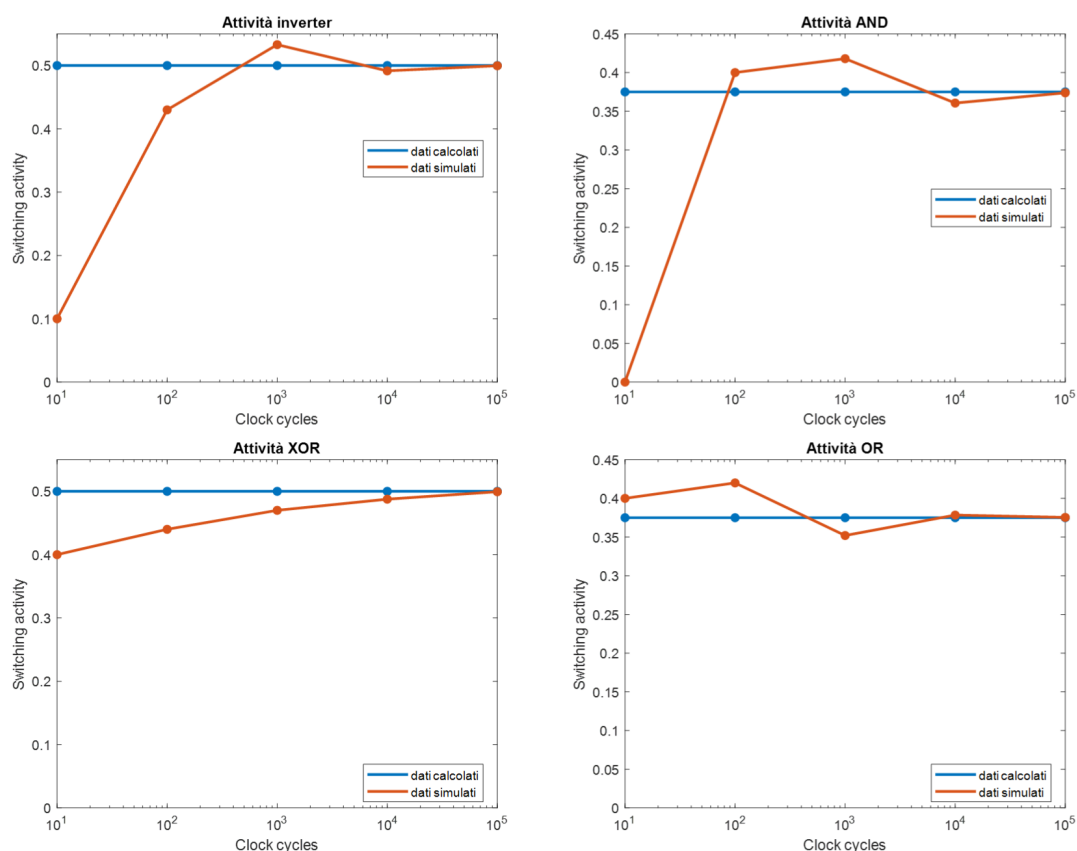
E' possibile stimare la switching activity dividendo il numero di commutazioni di un nodo per il numero di colpi di clock della relativa simulazione.

Dal momento che il parametro  $T_c$  si riferisce al numero totale di commutazioni, il numero di cicli di clock è ottenuto dividendo per due tale parametro.

I risultati dei calcoli sono riportati nella seguente tabella.

$T_c(CK)$	$T_c(INV)$	$T_c(AND)$	$T_c(OR)$	$T_c(XOR)$
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	0.3738	0.3754	0.4939

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 a man mano più accurata. In particolare, nel caso analizzato, si osserva che per un numero di cicli di clock superiore a 10000, i dati 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**

$$S = A \text{ XOR } B$$

$$C_{out} = A \text{ AND } B$$

- **Full adder**

$$S = A \text{ XOR } B \text{ XOR } C_{in}$$

$$C_{out} = A \text{ AND } B \text{ AND } C_{in}$$

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

- **Half adder**

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

$$P(C_{out} = 1) = P(A = 1) \cdot P(B = 1)$$

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

$$A(C_{out}) = 2 \cdot P(C_{out} = 1) \cdot (1 - P(C_{out} = 1))$$

- **Full adder**

$$P(S = 1) = P(A = 1) \cdot (1 - P(B = 1)) \cdot (1 - P(C_{in} = 1)) +$$

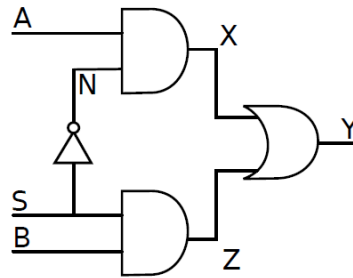
$$P(B = 1) \cdot (1 - P(A = 1)) \cdot (1 - P(C_{in} = 1)) +$$

$$P(C_{in} = 1) \cdot (1 - P(A = 1)) \cdot (1 - P(B = 1)) +$$

$$P(A = 1) \cdot P(B = 1) \cdot P(C_{in} = 1) \quad P(C_{out} = 1) = A(S) = A(C_{out}) =$$

### 1.3 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 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 di 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.

```

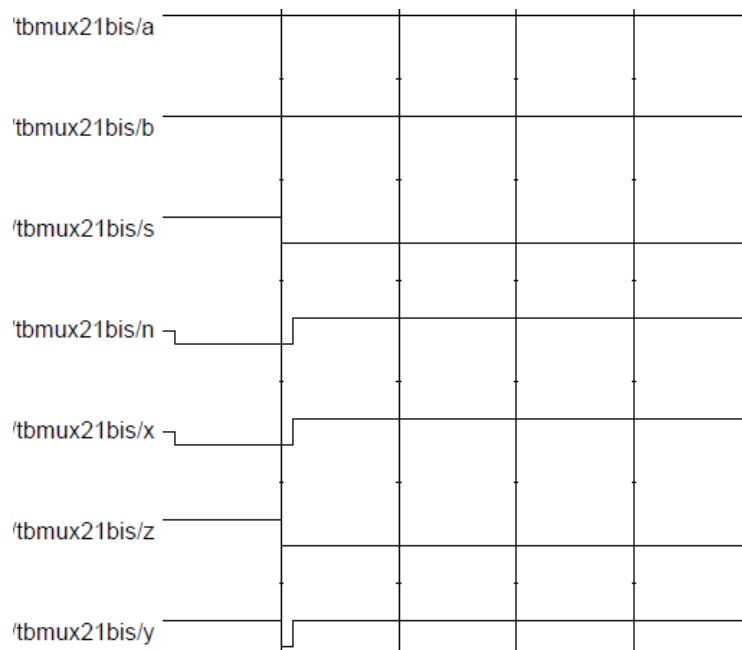
A <= '1';
B <= '1';
S <= '1', '0' after 1 ns;

```

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.

Si ipotizza che, a causa del ritardo di propagazione introdotto dall'inverter, vi è un intervallo temporale in cui i nodi interni X e Z assumono entrambi un valore logico basso, portando quindi l'uscita a 0.

Attraverso 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.



E' 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.

### MAPPA DI CARNO'

In particolare è possibile notare come, al fine di coprire la mappa con il circuito in figura X, siano stati coperti i due implicant rappresentati sulla mappa stessa. In particolare il glitch analizzato è dovuto ad una transizione degli ingressi che consegue in un passaggio tra un implicants e l'altro. Tale problema potrebbe essere risolto andando a coprire, in modo ridondante, con un terzo implicants per evitare la transizione precedentemente trattata.

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 alla potenza secondo la seguente equazione:

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

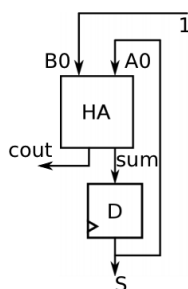
Dove C rappresenta la capacità di carico e V la tensione a cui viene caricata tale capacità.

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:

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

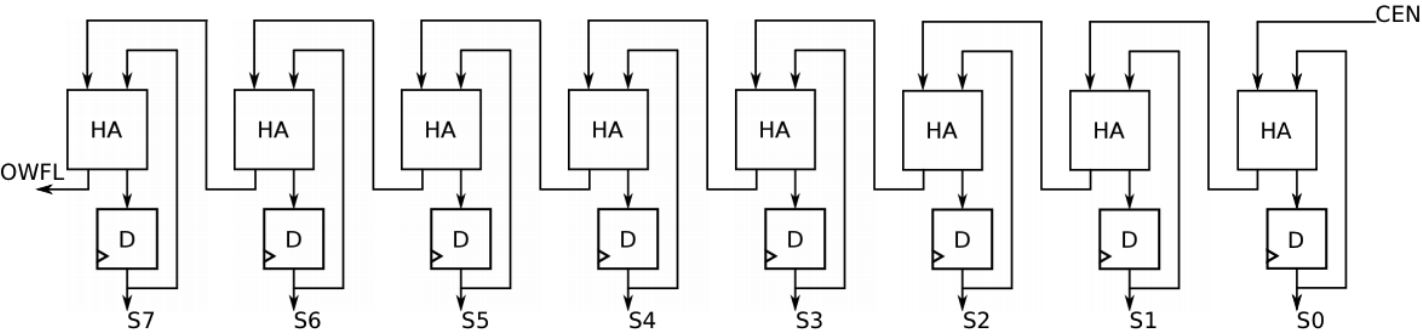
## 1.4 Calcolo di probabilità e attività: contatore sincrono

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 attivo alto, 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.

E' 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 quando il segnale di enable è attivo si ottiene il seguente timing.

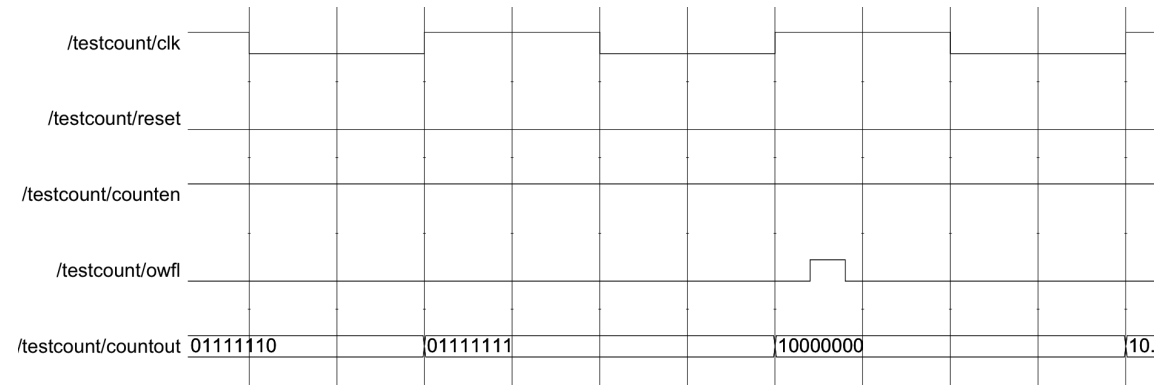
TIMING CONTATORE FATTO DA NOI

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
20	0.1	0
200	0.43	0.40
2000	0.533	0.418
20000	0.4916	0.3606
200000	0.4997	0.3738
200000	0.4997	0.3738
200000	0.4997	0.3738
200000	0.4997	0.3738
200000	0.4997	0.3738
200000	0.4997	0.3738
200000	0.4997	0.3738

Si può notare che 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.

Dalla simulazione è stato inoltre possibile osservare l'andamento delle waveforms delle uscite. E' stato possibile notare che, in un preciso intervallo temporale, il segnale OVFL ha assunto un valore logico alto durante il conteggio, in figura X, comportamento inatteso da parte di un contatore.





Si può notare che tale glitch avviene successivamente 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)	0.1	0	Cout(0)	0.1	0

E' 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.

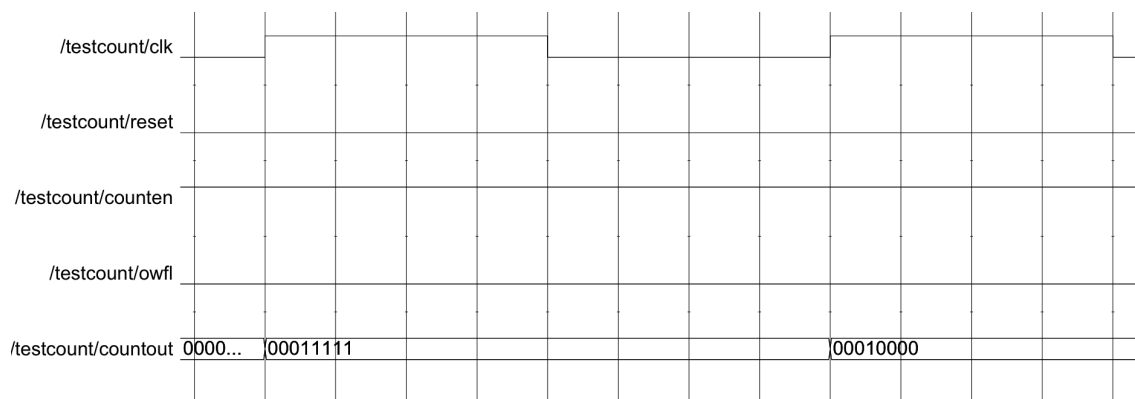
$$Periodo = 2ns$$

$$Ritardo_{somma} = 0.2ns$$

$$Ritardo_{carry} = 0.2ns$$

E' possibile notare 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.

## COSE DA FARE:

- Capire il toggle rate nel verbose (ESW)
- Capire l'incongruenza del verbose RCA
- Spiegare il worst case del sommatore
- Fare mappa di Karnaugh
- Plot segnali MUX (fatto da noi)
- Plot dei segnali del blocchetto contatore e dei primi bit del contatore (fatto da noi)
- Completare le tabelle
- Revisione grafica