POLITECNICO
DI TORINO

Laurea Magistrale in Ingegneria Elettronica

Anno Accademico 2018/2019

Sistemi Elettronici a basso consumo

prof. Maurizio Zamboni, prof. Mariagrazia Graziano

Relazione Laboratori

Raffaele Tuzzo 263722

Giulio Pecoraro 266391

Luigi Massari 265396

Contents

1. Laboratorio 1: Power Estimation: probabilistic techniques

1.1 Probability and Activity Calculation: Simple Logic Gates

Durante la prima parte dell'esercitazione è stata calcolata la probabilità di avere '1' logico in uscita di alcuni gate elementari, con la relativa Switching Activity.

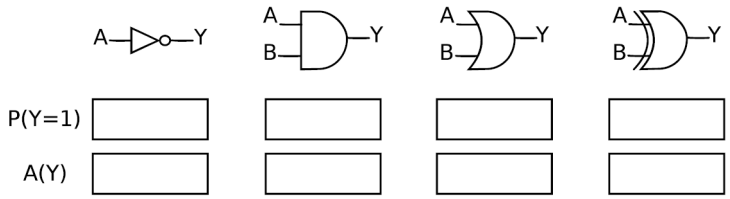

Figure 1.1: *Probabilità e Switching Activity stimati manualmente*

La probabilità di '1' logico è stata stimata semplicemente andando a valutare il rapporto fra il numero di possibili combinazioni con '1' logico diviso il numero di combinazioni totali. Invece per il calcolo della Switching Activity è stata utilizzata la formula vista a lezione:

$$A = P_1P_0 + P_1P_0 = 2P_1(1 - P_1)$$

Dove P_1 e P_0 sono le probabilità di avere '1' e '0' logici in uscita dalla mia

porta.
In seguito, tramite il programma *ModelSim* è stato analizzato il numero di toogle delle varie porte utilizzando un testbench sviluppato appositamente dai docenti. Si è andato a variare il numero di colpi di clock, come richiesto dalla traccia ed in seguito si sono comparati i valori ottenuti dalla simulazione con ciò che si era calcolato manualmente.
Tramite appositi comandi di Modelsim (**-power report**), sono stati stilati dei report relativi ad una stima delle commutazioni delle varie porte, della quale se ne riporta un esempio in Figura 1.2. Questi report consentono di stimare l'attività delle porte come verrà descritto in seguito.

```
Power Report                          Node     Tc     Ti    Time At
1    Time At 0    Time At X
-------------------------------------------------------------------
----------------------------
                                   /tbprob/clk  20000      0   5000000
ps   5000000 ps         0 ps
                                     /tbprob/reset      1      0      1000
ps   9999000 ps         0 ps
                                          /tbprob/ld      1      0      2000
ps   9998000 ps         0 ps
                                    /tbprob/dout(15)   4924      0   4929000
ps   5071000 ps         0 ps
                                    /tbprob/dout(14)   4923      0   4928500
ps   5071500 ps         0 ps
                                    /tbprob/dout(13)   4922      0   4928000
ps   5072000 ps         0 ps
                                    /tbprob/dout(12)   4922      0   4928000
ps   5072000 ps         0 ps
```

Figure 1.2: *Probabilità e Switching Activity stimati manualmente*

Si riportano nella tabella 1.1, i risultati ottenuti dalle varie simulazioni.

Tc(CK)	**Tc(INV)**	**Tc(AND)**	**Tc(OR)**	**Tc(XOR)**
20	1	?	4	4
200	43	40	42	44
2000	533	418	352	470
20000	4916	3606	3784	4876

Table 1.1: *Risultati simulazione*

Dai seguenti valori è facile ricavare i valori di Switching Activity simualte, in quanto si possono stimare da:

$$A = \frac{Tc(PORT)}{T_{CLK}}$$

Come ci si aspettava, essendo la Switching Activity il numero di toogle avvenuti in un periodo, i valori delle simulazioni vengono molto simili ai

valori calcolati analiticamente. Aumentando il tempo di simulazione, i valori di Switching Activity diventano sempre più precisi, arrivando ad avere un errore tra 0,01-0,5.

1.2 Probability and Activity Calculation: Half and full adder

Per prima cosa sono state calcolate le probabilità di avere un '1' logico sull'uscita sia dell'half adder e sia del full adder e le probabilità di avere un '1' logico come carry out degli stessi blocchi. Una volta eseguiti questi calcoli sono state calcolate anche le corrispettive switching activity.

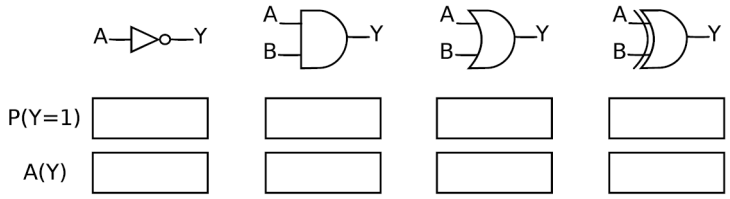

Figure 1.3: *Probabilità e Switching Activity stimati manualmente*

La probabilità di '1' logico è stata stimata semplicemente andando a valutare il rapporto fra il numero di possibili combinazioni con '1' logico diviso il numero di combinazioni totali. Invece per il calcolo della Switching Activity è stata utilizzata la formula vista a lezione:

$$A = P_1P_0 + P_1P_0 = 2P_1(1 - P_1)$$

α Dove P_1 e P_0 sono le probabilità di avere '1' e '0' logici in uscita dalla mia porta.
In seguito, si sono calcolate sempre manualmente le probabilità di uscita con le rispettive switching activity del Ripple carry adder, valutando la probabilità per ogni singolo Half adder come riportato in figura. Per questo calcolo iniziale gli ingressi sono stati considerati scorrelati ed equiprobabili.

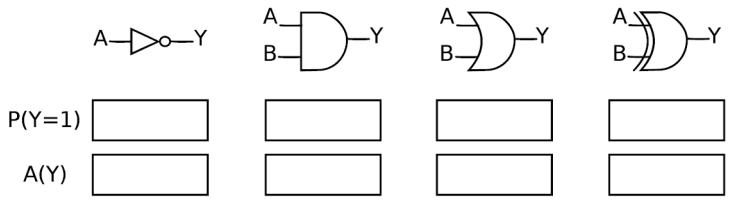

Figure 1.4: *Probabilità e Switching Activity stimati manualmente con ingressi equiprobabili*

Nel secondo caso, invece, gli ingressi sono stati considerati sempre scorrelati ma con probabilità diverse, infatti si ha che:

$$P(A =' 1') = 0.4 e P(B =' 1') = 0.6$$

I risultati ottenuti risultano uguali ai precedenti, per quanto riguarda la probabilità dell'uscita e del carry out. In seguito, tramite il programma *ModelSim* è stato simulato il Ripple carry adder (giusto?) utilizzando un testbench sviluppato appositamente dai docenti.Il testbench è stato costruito appositamente per assegnare ritardi diversi al bit di somma, DRCAS, e al bit di carry, DRCAC. Inoltre per garantire una simulazione generale si è utilizzato l'LFSR per generare ingressi randomici.Per una corretta visualizzazione dei risultati si è impostata una risoluzione di 1ps.Dopo aver visualizzato il power report, come fatto già in precedenza, si sono comparati i valori ottenuti dalla simulazione con ciò che si era calcolato manualmente, seguendo lo stesso ragionamento del punto precedente.

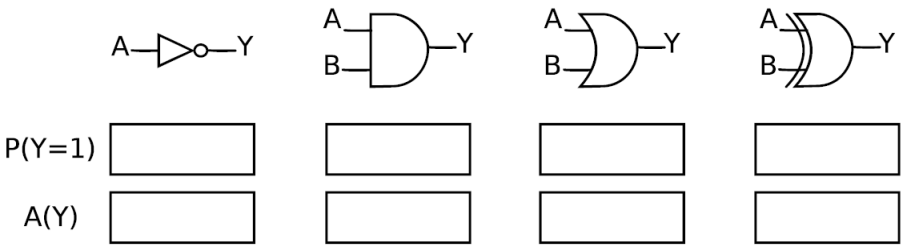

Figure 1.5: *Power report*

(cosa noto?) e bisogna mettere le waveforms?
¡- ATTENZIONE Si è poi simulato il caso in cui i due ritardi riguardanti il bit di somma e il bit di carry fossero uguali, e anche per questo è stato visualizzato il power report.

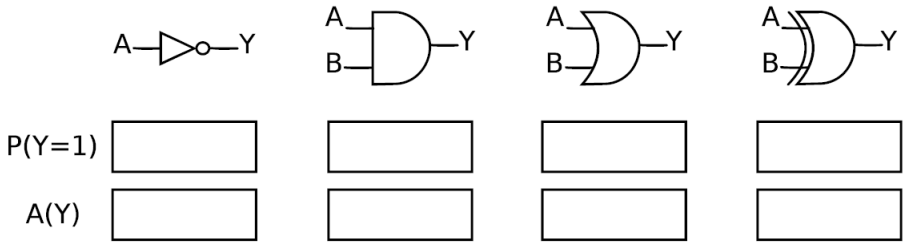

Figure 1.6: *Power report con ritardi uguali*

Da un analisi e confronto tra i risultati ottenuti manualmente e quelli ottenuti con le simulazioni, si nota come questi coincidano dato che avendo uguali ritardi, non riesco a simulare la presenza di eventuali glitch.
In seguito si è calcolata la switching activity totale dei due sommatori, utilizzanso la seguente formula:

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

What is the overhead computation of the second adder? <- cioè?
Come ultima cosa è stato simulato il secondo testbench che ci è stato fornito, dove si simulava sempre un Ripple carry adder, ma questa volta in maniera puramente combinatoria.Analizzando i risultati, si può concludere come non avendo un segnale di temporizzazione, lavoro alla massima velocità ma ho la presenza di glitch.

1.3 RCA synthesis and power analysis

Nella seguente sezione dell'esercitazione è stata analizzata la potenza del sommatore RCA già analizzato in precedenza, tramite il software *Synopsys*. Dopo aver analizzato ed elaborato i file che descrivono la struttura dell'RCA, il tutto è stato sintetizzato e sono stati raccolti i vari report relativi alla

potenza. Un primo report di potenza, riportato in Figura 1.7, descrive i contributi di potenza relativi alle 8 istanze dei Full-Adder che compongono il somamtore RCA.

```
Design        Wire Load Model            Library
------------------------------------------------
RCA                   5K_hvratio_1_1    NangateOpenCellLibrary


Global Operating Voltage = 1.1
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000ff
    Time Units = 1ns
    Dynamic Power Units = 1uW    (derived from V,C,T units)
    Leakage Power Units = 1nW


--------------------------------------------------------------------
---------------
                                    Switch   Int      Leak
Total
Hierarchy                           Power    Power    Power
Power    %
--------------------------------------------------------------------
---------------
RCA                                   9.741   16.743  953.483
27.437 100.0
  FAI_8 (FA_1)                        0.848    2.154  119.291
3.121  11.4
  FAI_7 (FA_2)                        1.293    2.150  118.962
3.562  13.0
  FAI_6 (FA_3)                        1.332    2.191  119.340
3.642  13.3
  FAI_5 (FA_4)                        1.328    2.171  119.110
3.618  13.2
  FAI_4 (FA_5)                        1.278    2.101  119.294
3.498  12.7
  FAI_3 (FA_6)                        1.263    2.090  118.979
3.471  12.7
  FAI_2 (FA_7)                        1.229    2.002  119.508
3.351  12.2
  FAI_1 (FA_0)                        1.171    1.884  118.999
3.175  11.6
1
```

Figure 1.7: *Power report*

Come ci si aspettava, i contributi dei vari Full Adder sono tutti simili tra di loro, ad eccezione dell'istanza *FAI_8*: il motivo consiste nel fatto che il Carry Out dell'ultimo Full Adder non è connesso a nessun altra porta, dunque il carico da pilotare è decisamente minore.
Diventa ora interessante andare ad analizzare la singola istanza, per andare a valutare l'origine dei singoli contributi di potenza. Tramite il comando *current_instance FAI_1* si va ad analizzare l'istanza relativa al primo Full Adder. Viene riportato il report in Figura 1.8.

Come ci si aspettava i valori di potenza risultano assolutamente identici al report trovato in precendenza e riportato in Figura 1.7. Il passo successivo è comprendere come avvenga la stima della potenza dinamica (*switching power*) dei singoli nodi del FA, dato che la potenza interna e quella di leak-

```
                          Cell      Driven Net  Tot Dynamic
Cell
                          Internal  Switching   Power
Leakage
Cell                      Power     Power       (% Cell/Tot)
Power      Attrs
-------------------------------------------------------------------
---------------
U1                           0.8205    0.0512      0.872 (94%)
36.0111
U4                           0.5889    0.5488      1.138 (52%)
36.1637
U3                           0.3374    0.1749      0.512 (66%)
32.5747
U2                           0.1377    0.3964      0.534 (26%)
14.2499
-------------------------------------------------------------------
---------------
Totals (4 cells)             1.884uW   1.171uW     3.056uW (62%)
118.999nW
```

Figure 1.8: *Power report*

age dipendono solo dal gate e non dal circuito. Si utilizza a questo scopo il comando *report_power -net -verbose* e si riporta il report risultante in Figura 1.9. Si può notare come la potenza dinamica consumata dal nodo di uscita

```
  Attributes
  ----------
      a  -  Switching activity information annotated on net
      d  -  Default switching activity information on net

                            Total      Static   Toggle   Switching
Net                         Net Load   Prob.    Rate     Power
Attrs
-------------------------------------------------------------------------
-------
n1                            4.694    0.493    0.1932      0.5488
Co                            4.554    0.512    0.1439      0.3964
n2                            2.010    0.488    0.1439      0.1749
S                             0.310    0.507    0.2735      0.0512
-------------------------------------------------------------------------
-------
Total (4 nets)                                               1.1714 uW
```

Figure 1.9: *Power report*

S sia praticamente nulla, in quanto essa non deve pilotare alcun carico, a meno della potenza che il software considera per le varie capacità parassite. Con questa analisi, il software si preoccupa anche di fare un'analisi probabilistica delle varie attività dei nodi. Dal report si può notare come siano assolutamenti concordi con i valori teorici trovati in precedenza.
Si ritorna ora al circuito complessivo, salendo di livello logico e si analizzano i nodi anche in questa situazione utilizzando lo stesso comando utilizzato in

precedenza. Il report viene raffigurato in Figura 1.10.

```
  Attributes
  ----------
      a  -  Switching activity information annotated on net
      d  -  Default switching activity information on net

                              Total      Static   Toggle   Switching
Net                           Net Load   Prob.    Rate     Power
Attrs
-------------------------------------------------------------------------------
-------
n1                              4.694    0.493    0.1932      0.5488
Co                              4.554    0.512    0.1439      0.3964
n2                              2.010    0.488    0.1439      0.1749
S                               0.310    0.507    0.2735      0.0512
-------------------------------------------------------------------------------
-------
Total (4 nets)                                                1.1714 uW
```

Figure 1.10: *Power report*

1.4 A simple MUX: glitch generation and propagation

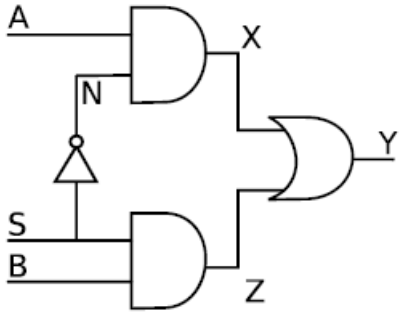

Figure 1.11: *Power report*

Durante questa sezione, viene chiesto di analizzare il comportamento di un Multiplexer, riportato in Figura 1.11, dove si assume che i ritardi delle varie porte elementari siano nulli, ad eccezione dell'Inverter che presenta un ritardo pari a 0,1 ns. Inizialmente si simula il circuito utilizzando *ModelSim* e si riportano le one in Figura 1.12. Si può notare benissimo come l'uscita Y abbia un glitch tra 1000 ps e 1100 ps, in quanto effettua una transizione

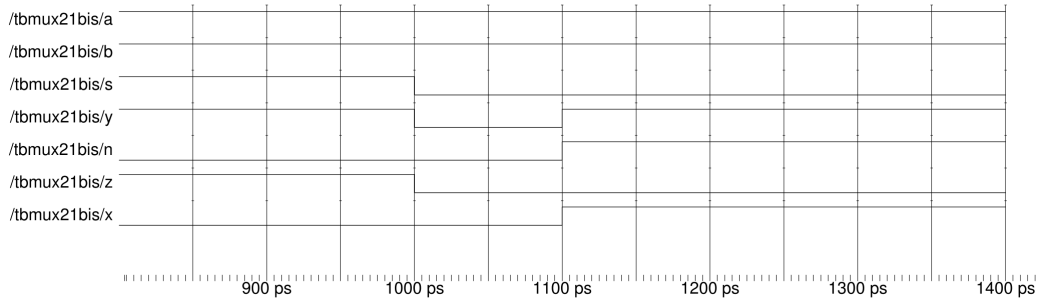

Figure 1.12: *Power report*

0-¿1 non voluta. Il tutto è causato dal ritardo introdotto dall'Inverter, che si può notare dal segnale 'n', che commuta al suo valore corretto dopo un ritardo di 0,1 ns, rispetto all'istante in cui cambiano tutti i restanti gate del circuito, ossia 1000 ps. Si verifica dunque un intervallo di tempo, ossia quello compreso tra 1000 ps e 1100 ps in cui la porta OR ha entrambi gli ingressi bassi e dunque produce un'uscita Y errata.
Questo produce quindi un glitch in uscita: in generale i glitch sono problematici a livello di potenza, in quanto si tratta di commutazioni spurie all'interno del mio circuito, che causano uno spreco di potenza. L'energia consumata per ogni doppia commutazione indesiderata è

$$E = C_L V_{DD}^2$$

considerando C_L come il carico che viene pilotato e V_{DD} la tensione di alimentazione alla quale si lavora.

1.5 Probability and Activity Calculation: Syncronous Counter

Si analizza in questa sezione un *Syncronous Counter* ad 1 bit, realizzato con una Half Adder e un D-FlipFlop, riportato in Figura 1.13. Il primo ingresso *A0* dell'Half Adder è l'uscita del D-FlipFlop, mentre il secondo ingresso *B0* viene collegato fisso ad 1. Si riporta il timing diagram del circuito in Figura **??**. Si può dedurre dal timing, che il segnale S ha una

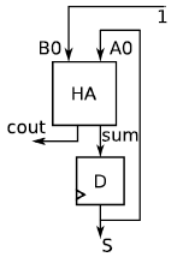

Figure 1.13: *Power report*

transizione per ogni colpo di clock, di conseguenza la sua Switching Activity sarà metà dell'attività del segnale di clock. Nel caso in cui il segnale B0 vada a 0, il valore del nodo S dipenderà esclusivamente dall'ultimo valore presente all'uscita del Flip-Flop, in quanto l'Half Adder sommerà l'ultimo valore presente nel Flip Flop con uno '0'. Quindi non si verificheranno più transizioni sul nodo S e di conseguenza neanche sul nodo C_{OUT}. In seguito

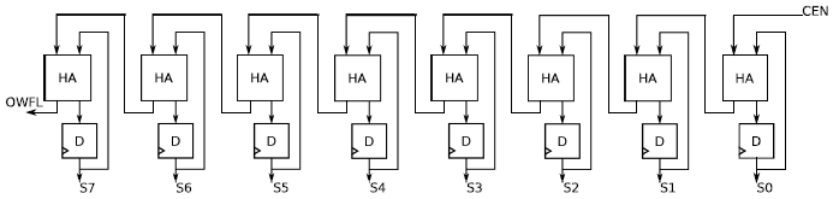

Figure 1.14: *Power report*

viene chiesto di analizzare la struttura presente in Figura 1.14. Questo circuito permette di ottenere un contatore sincrono ad 8 bit, utilizzando in cascata 8 celle del circuito precedente. Anche in questo si riporta, in Figura **??**, il timing diagram del funzionamento del circuito. In Tabella 1.2, vengono ora riportati il numero di commutazioni di tutti i segnali di uscita ricavati in modo analitico, considerando un intervallo di tempo compreso da quando il circuito parte dal valore '00000000', fino a quando non arriva al valore '11111111'.

Il segnale *CEN*, rappresenta l'enable del contatore, mentre il segnale

Signal	Number of Transitions
Clock	511
S0	255
S1	127
S2	63
S3	31
S4	15
S5	7
S6	3
S7	1

Table 1.2: *Risultati simulazione*

OWFL va ad 1 in presenza di overflow, dunque quando si supera la massima dinamica del contatore. Il segnale CEN è l'ingresso B0 del primo Half-Adder, mentre il segnale OWFL è il carry out dell'ultimo Half-Adder.
Il Segnale OWFL farà solo una transizione per ogni ciclo di conta, dunque andrà ad '1', solo quando le uscite S7-S0 andranno dal valore '11111111' al valore '00000000'. Di conseguenza avrò che la Switching Activity sarà:

$$E_{SW} = \frac{1}{2^8-1} =$$

Si vuole ora confrontare i risultati analitici ricavati in precedenza ed i risultati ottenuti sfruttando le simulazione di *Synopsis*. In Figura 1.15, viene riportata la simulazione ottenuta.

Viene chiesto di confrontare i valori analitici calcolati e riportati nella Tabella 1.2, con quelli derivanti da una simulazione *Synopsis*. Si simula il comportamento del circuito utilizzando un $T_{CLK} = 2ns$, per un periodo pari a 260 colpi di clock. Il report risultante è presente in Figura 1.15. Andando a studiare i file di testbench riportati, si può notare come vengano introdotti dei ritardi di 0.2 ns nelle uscite dell'Half Adder: questo provoca dei glitch che si andranno a propagare lungo la rete ed avranno effetti sul segnale OWFL, che invece di andare ad '1' solo una volta a fine conteggio, andrà ad '1' per tre volte (DA VERIFICARE!!!).
Tramite il power report si può notare come, in generale, il numero di commutazioni della simulazione sia superiore a quello ottenuto analiticamente: si può notare questo fenomeno nei segnali *stmp(n)*, ossia le uscite non sincrone dei Flip Flop, e nei segnali di Carry Out. Il tutto è dovuto ai glitch

Power Report Interval
520000

Power Report	Node	Tc	Ti	Time At 1	Time At 0	Time At X
/testcount/ucounter1/a	1	0	514000	6000	0	
/testcount/ucounter1/ck	520	0	260000	260000	0	
/testcount/ucounter1/reset	1	0	2000	518000	0	
/testcount/ucounter1/s(7)	2	0	256000	264000	0	
/testcount/ucounter1/s(6)	4	0	256000	264000	0	
/testcount/ucounter1/s(5)	8	0	256000	264000	0	
/testcount/ucounter1/s(4)	16	0	256000	264000	0	
/testcount/ucounter1/s(3)	32	0	256000	264000	0	
/testcount/ucounter1/s(2)	64	0	256000	264000	0	
/testcount/ucounter1/s(1)	128	0	256000	264000	0	
/testcount/ucounter1/s(0)	257	0	257000	263000	0	
/testcount/ucounter1/co	16	0	2000	517800	200	
/testcount/ucounter1/stmp(7)	30	0	256000	263600	400	
/testcount/ucounter1/stmp(6)	52	0	256000	263600	400	
/testcount/ucounter1/stmp(5)	88	0	256000	263600	400	
/testcount/ucounter1/stmp(4)	144	0	256000	263600	400	
/testcount/ucounter1/stmp(3)	224	0	256000	263600	400	
/testcount/ucounter1/stmp(2)	320	0	256000	263600	400	
/testcount/ucounter1/stmp(1)	385	0	256600	263000	400	
/testcount/ucounter1/stmp(0)	258	0	257000	262800	200	
/testcount/ucounter1/ctmp(8)	16	0	2000	517800	200	
/testcount/ucounter1/ctmp(7)	28	0	4000	515800	200	
/testcount/ucounter1/ctmp(6)	48	0	8000	511800	200	
/testcount/ucounter1/ctmp(5)	80	0	16000	503800	200	
/testcount/ucounter1/ctmp(4)	128	0	32000	487800	200	
/testcount/ucounter1/ctmp(3)	192	0	64000	455800	200	
/testcount/ucounter1/ctmp(2)	256	0	128000	391800	200	
/testcount/ucounter1/ctmp(1)	257	0	256800	263000	200	
/testcount/ucounter1/ctmp(0)	1	0	514000	6000	0	
/testcount/ucounter1/stmpsync(7)	2	0	256000	264000	0	
/testcount/ucounter1/stmpsync(6)	4	0	256000	264000	0	
/testcount/ucounter1/stmpsync(5)	8	0	256000	264000	0	
/testcount/ucounter1/stmpsync(4)	16	0	256000	264000	0	
/testcount/ucounter1/stmpsync(3)	32	0	256000	264000	0	
/testcount/ucounter1/stmpsync(2)	64	0	256000	264000	0	
/testcount/ucounter1/stmpsync(1)	128	0	256000	264000	0	
/testcount/ucounter1/stmpsync(0)	257	0	257000	263000	0	

Figure 1.15: *Power report*

che si propagano interamente al contatore a causa del ritardo di 0,2 ns di ciascun Half Adder.

Fortunatamente, questi glitch vengono filtrati dai Flip-FLop, in quanto il T_{CLK} è superiore al ritardo introdotto dalla rete combinatoria.

2. Laboratorio 2: FSM State Assignment and VHDL Synthesis

2.1 FSM State Assignment

Durante la prima parte dell'esercitazione di laboratorio, viene richiesto di implementare un circuito per sommare 6 numeri

$$s = a + b + c + d + e + f$$

utilizzando un unico sommatore, due multiplexer e un registro. Viene richiesto di valutare e minimizzare il consumo di potenza, andando a modificare la connessione degli input A-H, considerando esclusivamente l'attività della FSM e i bit di selezione del MUX S0-S3.
Il circuito completo è riportato in Figura 2.4, mentre la FSM è presente in Figura **??**.

Dopo varie ottimizzazioni, si è arrivati ad avere un'attività totale pari a 8 per il multiplexer e 6 per la State transition della macchina a stati, andando a considerare che la macchina a stati e il multiplexer ricomincino le operazioni una volta terminate. Nella tabella 2.1 viene riportata la configurazione degli stati e dei bit del multiplexer scelta:

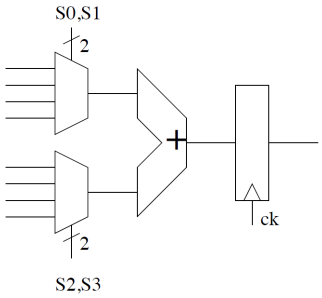

Figure 2.1: *Probabilità e Switching Activity stimati manualmente*

STATI	$S_3S_2S_1S_0$
000	0000
001	0101
011	0111
010	1110
110	1010

Table 2.1: *Risultati simulazione*

2.2 VHDL synthesis

Il secondo punto del laboratorio prevede di sintetizzare l'FSM tramite synopsys e studiarne le caratteristiche in termini di area, potenza e timing in modo da ricercare possibili ottimizzazioni. Si è utilizzata la libreria a 45 nm, definito un segnale di clock di periodo corrispondente a 10 ns, si è verificato il corretto inserimento tramite il comando *report_clock* e si è sintetizzato il circuito.

clock	**period**	**waveform**
CLK	10.00 ns	{0 5} V

Di seguito è riportato lo schema generato da synopsys:

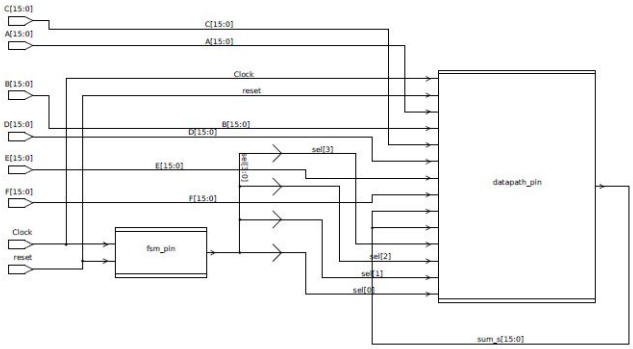

Figure 2.2: *Schematico del circuito sintetizzato*

Dal report sull area si sono ottenute informazioni riguardanti la quantità di componenti, le connessioni, l'area relativa occupata dalla logica combinatoria, circa il doppio rispetto a quella non combinatoria e quindi dell'area totale.

type	**number**
ports	114
nets	118
cells	2
references	2

area type	**value**
combinational	195.244003
noncombinational	101.08003
total cell	296.324006

Successivamente, dopo aver verificato la corretta codifica degli stati della FSM si è analizzato il timing del circuito, dal quale si sono ottenute importanti informazioni riguardo ai ritardi delle varie porte e allo Slack time nel caso del percorso peggiore che è di 8.03 ns, parametro che consente di ottimizzare frequenza di funzionamento del circuito, visto che la condizione necessaria è che lo slack time sia positivo. Inoltre si è eseguito il timing per i peggiori 10 percorsi e si sono ricavati i valori di slack.

Slack (MET)	value [ns]
1	8.04
2	8.04
3	8.04
4	8.04
5	8.04
6	8.05
7	8.05
8	8.05
9	8.05
10	8.05

Non si evidenziano rilevanti differenze tra i diversi slack, ciò è dovuto alla simmetria dei percorsi critici che presentano la stessa struttura. Ecco come sono distribuiti i peggiori slack:

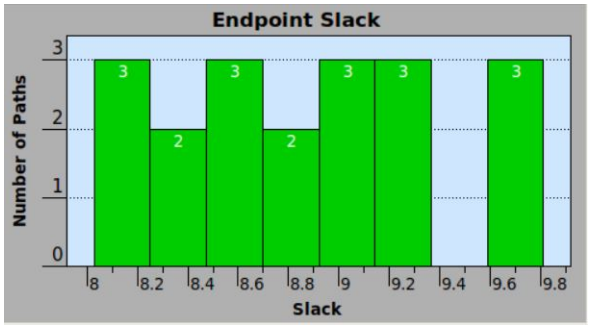

Figure 2.3: *Grafico distribuzione slacks*

Dunque si è analizzata la potenza dissipata sia dall'intera logica che da ogni singola cella. Il report sulla potenza distingue la potenza dissipata dinamicamente, staticamente e quella dovuta alle correnti di leakage, fornendo la percentuale rispetto alla potenza dissipata totale e analizzando il circuito a livello gerarchico. In questo modo è possibile capire quanto influiscono i diversi contributi di potenza dissipata e se necessario intervenire opportunamente per consumare meno.

hierarchy	switch	int	leak	tot	%
m_adder	11.943	28.443	5.46e+3	45.844	100
datapath_adder	10.930	25.532	5.03e+3	41.495	90.5
add_78	1.765	4.921	1.19e+3	7.879	17.2
fsm	1.012	2.911	425.171	4.349	9.5

Inoltre si è analizzata l'attività delle singole celle in modo da studiare i consumi di ogni singola cella. Da quest'ultimo report si è ricavato che le celle corrispondenti ai registri hanno dei consumi più elevati di potenza statica e consumano più del doppio della corrente di leakage rispetto alle altre celle del circuito.

cell	cell internal	driven net switching	tot dynamic [% cell/tot]	cell leakage
REG[0]	1.0163	0.0584	1.075 (95%)	87.1072
REG[1]	0.8660	0.1134	0.979 (88%)	81.4649
REG[2]	0.7468	0.1083	0.855 (87%)	84.7325
U8	0.0465	0.0297	7.62e-2 (61%)	33.6813
U9	0.0386	0.1368	0.175 (22%)	31.6341
U6	0.0375	0.1356	0.173 (22%)	18.0848
U5	0.0314	0.0193	5.07e-2 (62%)	19.3118
U4	0.0304	0.0174	4.77e-2 (64%)	17.9767
U10	0.0285	0.0746	0.103 (28%)	12.9020
U7	0.0233	0.1282	0.151 (15%)	15.8344
U3	0.0190	0.1236	0.143 (13%)	17.0242
U11	9.993e-3	0.0392	4.92e-2 (20%)	14.3532
tot (12 cells)	2.894 uW	984.349 nW	3.878 uW (75%)	434.107 nW

E' utile inoltre valutare il numero di commutazioni delle singole uscite valutando sia la capacità del carico che il tasso di commutazioni. In accordo con i risultati precedenti si denota un'attività più intensa per i registri per hanno un carico capacitivo molto più alto delle altre uscite anche se il toggle rate è praticamente equivalente per tutte le uscite e la probabilità statistica è comunque minore.

hierarchy	switch	int	leak	tot	%
m_adder	11.943	28.443	5.46e+3	45.844	100
datapath_adder	10.930	25.532	5.03e+3	41.495	90.5
add_78	1.765	4.921	1.19e+3	7.879	17.2
fsm	1.012	2.911	425.171	4.349	9.5

Inoltre si è analizzata l'attività delle singole celle in modo da studiare i consumi di ogni singola cella. Da quest'ultimo report si è ricavato che le celle corrispondenti ai registri hanno dei consumi più elevati di potenza statica e consumano più del doppio della corrente di leakage rispetto alle altre celle del circuito.

net	**total net load**	**static prob.**	**toggle rate**	**switching power**
S[2]	11.104	0.326	0.0244	0.1641
S[0]	9.304	0.228	0.0244	0.1375
S[1]	10.166	0.295	0.0221	0.1360
S[3]	9.541	0.186	0.0200	0.1153
n21	10.518	0.088	0.0181	0.1153
n5	6.169	0.814	0.0200	0.0746
n6	3.949	0.772	0.0244	0.0584
n8	4.078	0.706	0.0222	0.0546
n25	3.843	0.500	0.0221	0.0514
n28	6.482	0.772	0.0100	0.0392
n27	1.980	0.706	0.0244	0.0293
N8	1.438	0.098	0.0221	0.0193
n7	1.438	0.0392	0.0200	0.0174
tot (13 nets)				1.0125 uW

Si è adesso focalizzata l'attenzione sui consumi della Macchina a Stati. La FSM è caratterizzata da una leakage current di 418 nW a fronte dei 434 nW totali e anche per gli altri contributi di consumo di potenza dinamica i dati tendono ad evidenziare il ruolo preponderante del componente sul totale consumo del circuito.

Cell	Cell Internal Power	Driven Net Switching Power	Tot Dynamic Power (% Cell/Tot)	Cell Leakage Power	Attrs
CURRENTSTATE_reg[0]	0.8986	0.0979	0.997 (90%)	89.1728	
CURRENTSTATE_reg[1]	0.5562	0.0461	0.602 (92%)	81.9647	
CURRENTSTATE_reg[2]	0.3711	2.719e-03	0.374 (99%)	79.5076	
U11	0.0999	0.3922	0.492 (20%)	14.3532	
U6	0.0989	0.0453	0.144 (69%)	30.0052	
U5	0.0644	0.0595	0.124 (52%)	17.0430	
U7	0.0384	0.0491	8.75e-02 (44%)	11.6295	
U4	0.0193	1.813e-03	2.12e-02 (91%)	34.6953	
U10	0.0109	8.310e-03	1.92e-02 (57%)	20.1295	
U9	2.905e-03	1.593e-03	4.50e-03 (65%)	16.3885	
U8	2.689e-03	6.834e-03	9.52e-03 (28%)	5.3388	
U3	1.734e-03	3.431e-04	2.08e-03 (83%)	18.5247	
Totals (12 cells)	2.165uW	711.655nW	2.877uW (75%)	418.753nW	

Figure 2.4: *Potenza dissipata dalla FSM*

Dopodiché si è provato a variare la frequenza di lavoro del circuito, provando a sintetizzare il circuito in modo da lavorare alla massima frequenza di funzionamento consentita dal percorso critico. Dall'analisi sul timing si è trovato lo slack peggiore di circa 8.02 ns lavorando a 10 ns di periodo di clock. Si può allora decrementare il periodo di clock fino a 10-8.02=1.98 ns. Difatti si è scelto un periodo di clock di 2 ns.

cell internal power	141.4894 uW (71%)
net switching power	58.8064 uW (29%)
total dynamic power	200.2958 uW (100%)
cell leakage power	5.5273 uW

Si nota come sia aumentata la Total Dynamic Power, questo poiché aumentando la frequenza operativa aumentano anche il numero di commutazioni interne e quindi viene dissipata maggiore potenza dinamica. Resta invece invariata la corrente di leakage che infatti dipende solo dalla tecnologia usata.

Infine è stato posto al sintetizzatore un ulteriore vincolo sulla massima potenza dinamica dissipabile a 200 uW, considerando che nell'ultimo report la potenza totale dissipata ammonta a 200.2958 uW.

cell internal power	140.3547 uW (70%)
net switching power	59.0397 uW (30%)
total dynamic power	199.3944 uW (100%)
cell leakage power	5.5832 uW

Adesso la potenza dinamica totale dissipata è di 199.3944 uW e rispetta il vincolo. Inoltre, si nota che stavolta il parametro della leakage power è leggermente variato, questo poichè stavolta per rispettare il vincolo sul consumo di potenza è stata variata la topologia del circuito usando differenti porte logiche.

3. Laboratorio 3: Clock gating, pipelining and parallelizing

Durante questa esperienza di laboratorio verranno analizzate una serie di tecniche per ridurre i consumi mediante l'ottimizzazione dell'architettura dei circuiti.

3.1 A first approach to clock gating

Un prima tecnica utilizzata per ridurre i consumi andando a lavorare sull'architettura è il *Clock Gating*. Questa tecnica permette di "staccare" il clock ad un determinato blocco del mio circuito, quando questo non deve lavorare. Un circuito di massima è riportato in Figura 3.1.

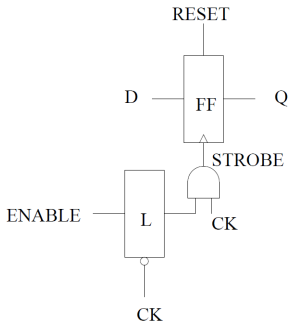

Figure 3.1: *Schema implementativo della tecnica del Clock Gating*

Nella prima parte dell'esperienza viene chiesto di analizzare il file *ckgbug.vhd* che contiene la descrizione VHDL di una struttura composta da due registri in cascata, denominati L1 ed L2. La tecnica del clock gating viene applicata al secondo registro, mediante una AND tra il clock e un segnale di ENABLE. Bisogna prestare attenzione al fatto che i segnali di ingresso dei due registri (D1 e D2) siano rispettivamente *std_logic_vector (7 downto 0)* e *std_logic_vector (0 to 7*.
In una prima simulazione, si forza il segnale D1 al valore '01111111' e, attivando il segnale di ENABLE, ci si aspetterebbe che D2, al colpo di clock successivo, vada al valore '111111101' e che l'uscita di L2, denominata D3, vada al valore '01111111' al colpo di clock ancora successivo. In realtà la simulazione porta al risultato in Figura 3.2. Si può ben notare come l'uscita

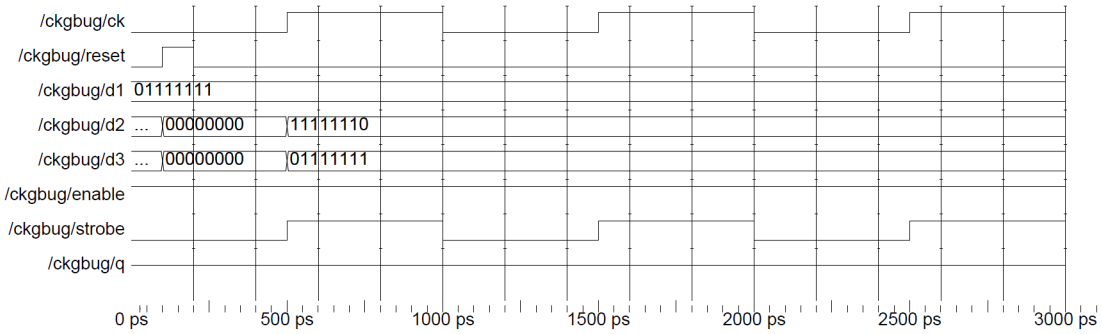

Figure 3.2: *Schema implementativo della tecnica del Clock Gating*

D3 dopi esattamente D1 dopo un solo colpo di clock. Questo è dovuto al fatto che il clock gated arriva a L2 un "passo di simulazione" dopo L1, perchè il simulatore programma il calcolo dell'uscita AND dopo l'assegnazione del clock. Quindi accade come se l'AND avesse un ritardo interno.
La traccia suggerisce che si può risolvere questo inconveniente andando ad aggiungere un ritardo *Clock-to-Output* pari a 0.1 ps all'uscita del generico Flip-Flop. Andando a risimulare il file, si ottiene il comportamento desiderato, che viene riportato in Figura 3.3.

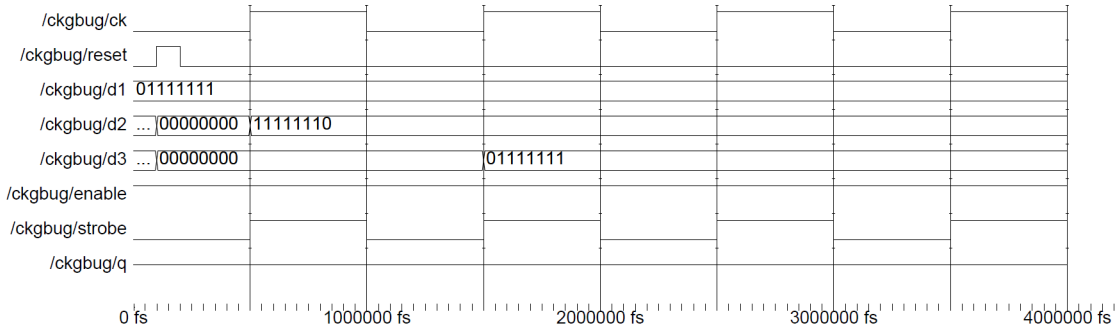

Figure 3.3: *Schema implementativo della tecnica del Clock Gating*

Infine si aggiunge un ulteriore ritardo alla porta AND pari a 0.2 ps, ossia un tempo superiore a quello *Clock-to-Output* inserito in precendeza. In questo modo si va a violtare il t_{hold} e dunque il circuito ritorna nella situazione precendete con un funzionamento non corretto. Il risultato della simulazione è riportato in Figura 3.4.

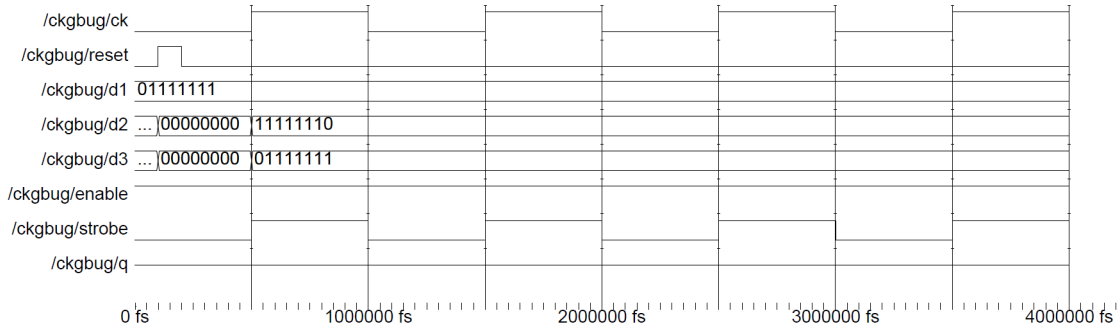

Figure 3.4: *Schema implementativo della tecnica del Clock Gating*

4. Laboratorio 4: Bus Encoding

Durante questa esperienza di laboratorio, viene chiesto di analizzare e di valutare alcune tecniche di bus encoding. Nella prima parte dell'esperienza viene chiesto di valutare le performance delle varie tecniche in termini di transizioni e poi ———

4.1 Simulation

Durante la prima parte dell'esperienza viene chiesto di valutare, in termini di commutazioni, diverse tecniche di bus-encoding. Il tutto viene simulato grazie ad un testbench fornite e tramite i *power report* generati da *ModelSim*.

4.1.1 Non-encoded

Inizialmente si è simulato il caso in cui utilizzi un bus non codificato, andando a vedere l'evoluzione dei toogle in 10000 colpi di clock in modo tale da avere un punto di riferimento nel confronto con le altre varie tecniche. All'interno del testbench da simulare sono distinti due processi:

- processo per gli indirizzi
- processo per i dati

i quali prendono come input i dati contenuti nel file *rndin.txt*, che, come sottolineato nella traccia, contiene delle stringe ad 8 bit con un a probabilità bassa di 1 logico. Questo fa già prevedere che la tecnica *Transition-Based*

risulterà la meno performante, in quanto questa tipologia di codifica porta ad riduzione delle commutazioni, solo nel caso in cui le probabilità di '1' logico e di '0' logico siano equiprobabili. risulterà meno efficiente. Nelle Figure 4.2 e 4.1, sono riportati i *power report* rispettivamente nel caso di dati e di indirizzi.Infine, nelle Tabelle 4.1 e 4.2 si riportano le switching activity rispettivamente nel caso di indirizzi e dati.

Power Report	Node	Tc	Ti	Time At 1	Time At 0	Time At X
	/testbench/ABUSNORM(7)	4974	0	50030000	49970000	0
	/testbench/ABUSNORM(6)	5022	0	49460000	50540000	0
	/testbench/ABUSNORM(5)	4987	0	49384900	50615100	0
	/testbench/ABUSNORM(4)	4972	0	49970000	50030000	0
	/testbench/ABUSNORM(3)	4960	0	51250000	48750000	0
	/testbench/ABUSNORM(2)	4965	0	49894900	50105100	0
	/testbench/ABUSNORM(1)	5024	0	50000000	50000000	0
	/testbench/ABUSNORM(0)	5060	0	49940000	50060000	0
	/testbench/COUNTBUSNORM(7)	4974	0	50030000	49969900	100
	/testbench/COUNTBUSNORM(6)	5021	0	49455000	50544900	100
	/testbench/COUNTBUSNORM(5)	4987	0	49375000	50624900	100
	/testbench/COUNTBUSNORM(4)	4972	0	49970000	50029900	100
	/testbench/COUNTBUSNORM(3)	4959	0	51245000	48754900	100
	/testbench/COUNTBUSNORM(2)	4965	0	49885000	50114900	100
	/testbench/COUNTBUSNORM(1)	5023	0	49995000	50004900	100
	/testbench/COUNTBUSNORM(0)	5060	0	49940000	50059900	100
	/testbench/CBUSNORM(7)	4973	0	50025000	49974900	100
	/testbench/CBUSNORM(6)	5021	0	49445000	50554900	100
	/testbench/CBUSNORM(5)	4987	0	49365000	50634900	100
	/testbench/CBUSNORM(4)	4972	0	49970000	50029900	100
	/testbench/CBUSNORM(3)	4958	0	51240000	48759900	100
	/testbench/CBUSNORM(2)	4964	0	49880000	50119900	100
	/testbench/CBUSNORM(1)	5023	0	49985000	50014900	100
	/testbench/CBUSNORM(0)	5060	0	49940000	50059900	100
	/testbench/CK	20000	0	50000000	50000000	0
	/testbench/RST	2	0	200	99999800	0
	/testbench/A(7)	4974	0	50030000	49970000	0
	/testbench/A(6)	5022	0	49460000	50540000	0
	/testbench/A(5)	4987	0	49384900	50615100	0
	/testbench/A(4)	4972	0	49970000	50030000	0
	/testbench/A(3)	4960	0	51250000	48750000	0
	/testbench/A(2)	4965	0	49894900	50105100	0
	/testbench/A(1)	5024	0	50000000	50000000	0
	/testbench/A(0)	5060	0	49940000	50060000	0

Figure 4.1: *Power Report, tecnica no-encoding, caso 'data'*

Nodo	E_{SW}
countbusnorm(7)	0.0097
countbusnorm(6)	0.0118
countbusnorm(5)	0.0312
countbusnorm(4)	0.0624
countbusnorm(3)	0.1249
countbusnorm(2)	0.2499
countbusnorm(1)	0.4999
countbusnorm(0)	0.9999

Table 4.1: *Switching Activity, tecnica no-encoding, caso 'data'*

Power Report	Node	Tc	Ti	Time At 1	Time At 0	Time At X
	/testbench/ABUSNORM(7)	97	0	48164900	51835100	0
	/testbench/ABUSNORM(6)	118	0	37760000	62240000	0
	/testbench/ABUSNORM(5)	312	0	49927000	50073000	0
	/testbench/ABUSNORM(4)	625	0	49931900	50068100	0
	/testbench/ABUSNORM(3)	1250	0	50007000	49993000	0
	/testbench/ABUSNORM(2)	2500	0	50007000	49993000	0
	/testbench/ABUSNORM(1)	5000	0	50007000	49993000	0
	/testbench/ABUSNORM(0)	10000	0	50007000	49993000	0
	/testbench/COUNTBUSNORM(7)	97	0	48155000	51844900	100
	/testbench/COUNTBUSNORM(6)	118	0	37760000	62239900	100
	/testbench/COUNTBUSNORM(5)	312	0	49920000	50079900	100
	/testbench/COUNTBUSNORM(4)	624	0	49920000	50079900	100
	/testbench/COUNTBUSNORM(3)	1249	0	49995000	50004900	100
	/testbench/COUNTBUSNORM(2)	2499	0	49995000	50004900	100
	/testbench/COUNTBUSNORM(1)	4999	0	49995000	50004900	100
	/testbench/COUNTBUSNORM(0)	9999	0	49995000	50004900	100
	/testbench/CBUSNORM(7)	97	0	48145000	51854900	100
	/testbench/CBUSNORM(6)	118	0	37760000	62239900	100
	/testbench/CBUSNORM(5)	312	0	49920000	50079900	100
	/testbench/CBUSNORM(4)	624	0	49920000	50079900	100
	/testbench/CBUSNORM(3)	1249	0	49985000	50014900	100
	/testbench/CBUSNORM(2)	2499	0	49985000	50014900	100
	/testbench/CBUSNORM(1)	4999	0	49985000	50014900	100
	/testbench/CBUSNORM(0)	9998	0	49990000	50009900	100
	/testbench/CK	20000	0	50000000	50000000	0
	/testbench/RST	2	0	200	99999800	0
	/testbench/A(7)	97	0	48164900	51835100	0
	/testbench/A(6)	118	0	37760000	62240000	0
	/testbench/A(5)	312	0	49927000	50073000	0
	/testbench/A(4)	625	0	49931900	50068100	0
	/testbench/A(3)	1250	0	50007000	49993000	0
	/testbench/A(2)	2500	0	50007000	49993000	0
	/testbench/A(1)	5000	0	50007000	49993000	0
	/testbench/A(0)	10000	0	50007000	49993000	0

Figure 4.2: *Power Report, tecnica no-encoding, caso 'address'*

Nodo	E_{SW}
countbusnorm(7)	0.4974
countbusnorm(6)	0.5021
countbusnorm(5)	0.4987
countbusnorm(4)	0.4972
countbusnorm(3)	0.4959
countbusnorm(2)	0.4965
countbusnorm(1)	0.5023
countbusnorm(0)	0.506

Table 4.2: *Switching Activity, tecnica no-encoding, caso 'address'*

4.1.2 Bus-invert technique, Transition based technique, Gray technique

La codifica del Bus-invert consiste nel verificare che il dato successivo comporti un numero di commutazioni superiori di $\frac{N}{2}$, dove N è il numero di

linee che compone il bus. In caso, conviene mandare il dato complementato e forzare ad '1' un bit denominato *inv*, per segnalare che il dato in arrivo è in realtà complementato. Nelle Tabelle 4.3 e 4.4, si riportano le Switching Activity rispettivamente nel caso di indirizzi e dati.

Nodo	E_{SW}
countbusinv(8)	0.0624
countbusinv(7)	0.0527
countbusinv(6)	0.0506
countbusinv(5)	0.0312
countbusinv(4)	0
countbusinv(3)	0.0625
countbusinv(2)	0.1875
countbusinv(1)	0.4375
countbusinv(0)	0.9375

Table 4.3: *Switching Activity, tecnica bus-invert, caso 'address'*

Nodo	E_{SW}
countbusinv(8)	0.365
countbusinv(7)	0.3576
countbusinv(6)	0.3611
countbusinv(5)	0.3617
countbusinv(4)	0.364
countbusinv(3)	0.3613
countbusinv(2)	0.3601
countbusinv(1)	0.3639
countbusinv(0)	0.3722

Table 4.4: *Switching Activity, tecnica bus-invert, caso 'data'*

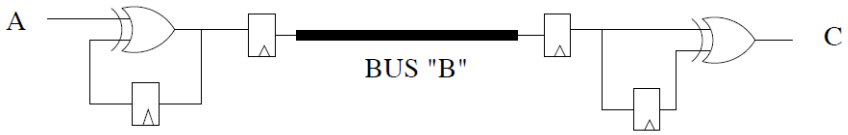

Figure 4.3: *Schema tecnica 'Transition Based'*

Si esegue la medesima analisi per la tecnica *Transition Based*, riportata in Figura 4.3. Questa tecnica consiste nell'inviare uno '0' logico ogni qual volta la linea corrisponde alla precedente, mentre un '1' logico viene inteso come

complemento del valore precedente della linea. I risultati delle simulazioni sono riportati per il caso dati in Tabella 4.5 e per il caso indirizzi in Tabella 4.6.

Nodo	E_{SW}
countbustran(7)	0.5003
countbustran(6)	0.4946
countbustran(5)	0.4938
countbustran(4)	0.4997
countbustran(3)	0.5125
countbustran(2)	0.4989
countbustran(1)	0.500
countbustran(0)	0.4994

Table 4.5: *Switching Activity, tecnica Transition-Based, caso 'data'*

Nodo	E_{SW}
countbustran(7)	0.4816
countbustran(6)	0.3776
countbustran(5)	0.4992
countbustran(4)	0.4992
countbustran(3)	0.500
countbustran(2)	0.500
countbustran(1)	0.500
countbustran(0)	0.500

Table 4.6: *Switching Activity, tecnica Transition-Based, caso 'address'*

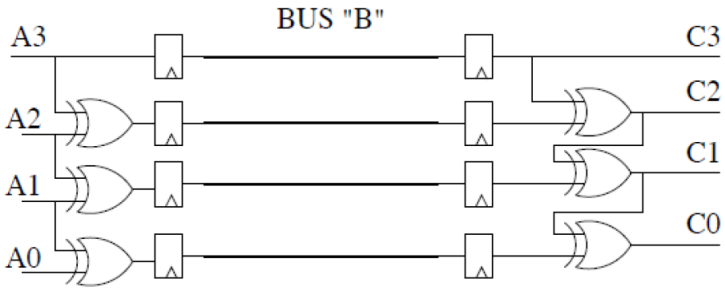

Figure 4.4: *Schema tecnica 'Codifica di Gray'*

La tecnica *Codifica di Gray*, riportata in Figura 4.4: questa tecnica viene sfruttata specialmente nelle trasmissioni sequenziali, come ad esempio nel caso di indirizzi, in modo tale cambio esclusivamente un solo bit alla volta. I risultati delle Switching Activity sono riportati per i dati nella Tabella 4.7 e nella Tabella 4.8 per gli indirizzi.

Nodo	E_{SW}
countbusgray(7)	0.4974
countbusgray(6)	0.5087
countbusgray(5)	0.4952
countbusgray(4)	0.4981
countbusgray(3)	0.4937
countbusgray(2)	0.5056
countbusgray(1)	0.4962
countbusgray(0)	0.5075

Table 4.7: *Switching Activity, tecnica Codifica di Gray, caso 'data'*

Nodo	E_{SW}
countbusgray(7)	0.097
countbusgray(6)	0.055
countbusgray(5)	0.194
countbusgray(4)	0.312
countbusgray(3)	0.625
countbusgray(2)	0.125
countbusgray(1)	0.250
countbusgray(0)	0.500

Table 4.8: *Switching Activity, tecnica Transition Based, caso 'address'*

Si riportano per completezza i *Power Report* delle simulazioni svolte con *ModelSim*. In Figura 4.5 si riporta il Report nel caso di indirizzi, mentre in Figura 4.6 il caso di dati.

Power Report Interval
100000000

Power Report	Node	Tc	Ti	Time At 1	Time At 0	Time At X
	/testbench/A(7)	97	0	48164900	51835100	0
	/testbench/A(6)	118	0	37760000	62240000	0
	/testbench/A(5)	312	0	49927000	50073000	0
	/testbench/A(4)	625	0	49931900	50068100	0
	/testbench/A(3)	1250	0	50007000	49993000	0
	/testbench/A(2)	2500	0	50007000	49993000	0
	/testbench/A(1)	5000	0	50007000	49993000	0
	/testbench/A(0)	10000	0	50007000	49993000	0
	/testbench/COUNTBUSNORM(7)	97	0	48155000	51844900	100
	/testbench/COUNTBUSNORM(6)	118	0	37760000	62239900	100
	/testbench/COUNTBUSNORM(5)	312	0	49920000	50079900	100
	/testbench/COUNTBUSNORM(4)	624	0	49920000	50079900	100
	/testbench/COUNTBUSNORM(3)	1249	0	49995000	50004900	100
	/testbench/COUNTBUSNORM(2)	2499	0	49995000	50004900	100
	/testbench/COUNTBUSNORM(1)	4999	0	49995000	50004900	100
	/testbench/COUNTBUSNORM(0)	9999	0	49995000	50004900	100
	/testbench/COUNTBUSTRAN(7)	4816	0	24080000	75919900	100
	/testbench/COUNTBUSTRAN(6)	3776	0	18880000	81119900	100
	/testbench/COUNTBUSTRAN(5)	4992	0	24960000	75039900	100
	/testbench/COUNTBUSTRAN(4)	4992	0	24960000	75039900	100
	/testbench/COUNTBUSTRAN(3)	5000	0	25000000	74999900	100
	/testbench/COUNTBUSTRAN(2)	5000	0	25000000	74999900	100
	/testbench/COUNTBUSTRAN(1)	5000	0	25000000	74999900	100
	/testbench/COUNTBUSTRAN(0)	5000	0	50000000	49999900	100
	/testbench/COUNTBUSGRAY(7)	97	0	48155000	51844900	100
	/testbench/COUNTBUSGRAY(6)	55	0	24475000	75524900	100
	/testbench/COUNTBUSGRAY(5)	194	0	49920000	50079900	100
	/testbench/COUNTBUSGRAY(4)	312	0	49920000	50079900	100
	/testbench/COUNTBUSGRAY(3)	625	0	49995000	50004900	100
	/testbench/COUNTBUSGRAY(2)	1250	0	50000000	49999900	100
	/testbench/COUNTBUSGRAY(1)	2500	0	50000000	49999900	100
	/testbench/COUNTBUSGRAY(0)	5000	0	50000000	49999900	100
	/testbench/COUNTBUSINV(8)	624	0	49920000	50079900	100
	/testbench/COUNTBUSINV(7)	527	0	50075000	49924900	100
	/testbench/COUNTBUSINV(6)	506	0	49920000	50079900	100
	/testbench/COUNTBUSINV(5)	312	0	49920000	50079900	100
	/testbench/COUNTBUSINV(3)	625	0	49995000	50004900	100
	/testbench/COUNTBUSINV(2)	1875	0	49995000	50004900	100
	/testbench/COUNTBUSINV(1)	4375	0	49995000	50004900	100
	/testbench/COUNTBUSINV(0)	9375	0	49995000	50004900	100

Figure 4.5: *Power Report complessivo, caso 'address'*

```
Power Report Interval
100000000

Power Report                          Node     Tc     Ti    Time At 1    Time At 0    Time At X
-------------------------------------------------------------------------------------------------
                        /testbench/A(7)   4974      0     50030000     49970000            0
                        /testbench/A(6)   5022      0     49460000     50540000            0
                        /testbench/A(5)   4987      0     49384900     50615100            0
                        /testbench/A(4)   4972      0     49970000     50030000            0
                        /testbench/A(3)   4960      0     51250000     48750000            0
                        /testbench/A(2)   4965      0     49894900     50105100            0
                        /testbench/A(1)   5024      0     50000000     50000000            0
                        /testbench/A(0)   5060      0     49940000     50060000            0
              /testbench/COUNTBUSNORM(7)  4974      0     50030000     49969900          100
              /testbench/COUNTBUSNORM(6)  5021      0     49455000     50544900          100
              /testbench/COUNTBUSNORM(5)  4987      0     49375000     50624900          100
              /testbench/COUNTBUSNORM(4)  4972      0     49970000     50029900          100
              /testbench/COUNTBUSNORM(3)  4959      0     51245000     48754900          100
              /testbench/COUNTBUSNORM(2)  4965      0     49885000     50114900          100
              /testbench/COUNTBUSNORM(1)  5023      0     49995000     50004900          100
              /testbench/COUNTBUSNORM(0)  5060      0     49940000     50059900          100
              /testbench/COUNTBUSTRAN(7)  5003      0     49825000     50174900          100
              /testbench/COUNTBUSTRAN(6)  4946      0     49830000     50169900          100
              /testbench/COUNTBUSTRAN(5)  4938      0     50480000     49519900          100
              /testbench/COUNTBUSTRAN(4)  4997      0     49585000     50414900          100
              /testbench/COUNTBUSTRAN(3)  5125      0     49945000     50054900          100
              /testbench/COUNTBUSTRAN(2)  4989      0     50245000     49754900          100
              /testbench/COUNTBUSTRAN(1)  5000      0     49890000     50109900          100
              /testbench/COUNTBUSTRAN(0)  4994      0     49540000     50459900          100
              /testbench/COUNTBUSGRAY(7)  4974      0     50030000     49969900          100
              /testbench/COUNTBUSGRAY(6)  5087      0     50105000     49894900          100
              /testbench/COUNTBUSGRAY(5)  4952      0     50100000     49899900          100
              /testbench/COUNTBUSGRAY(4)  4981      0     50505000     49494900          100
              /testbench/COUNTBUSGRAY(3)  4937      0     49415000     50584900          100
              /testbench/COUNTBUSGRAY(2)  5056      0     50320000     49679900          100
              /testbench/COUNTBUSGRAY(1)  4962      0     50710000     49289900          100
              /testbench/COUNTBUSGRAY(0)  5075      0     49715000     50284900          100
               /testbench/COUNTBUSINV(8)  3650      0     49260000     50739900          100
               /testbench/COUNTBUSINV(7)  3576      0     48810000     51189900          100
               /testbench/COUNTBUSINV(6)  3611      0     49595000     50404900          100
               /testbench/COUNTBUSINV(5)  3617      0     52235000     47764900          100
               /testbench/COUNTBUSINV(4)  3640      0     50190000     49809900          100
               /testbench/COUNTBUSINV(3)  3613      0     49685000     50314900          100
               /testbench/COUNTBUSINV(2)  3601      0     50405000     49594900          100
               /testbench/COUNTBUSINV(1)  3639      0     50295000     49704900          100
               /testbench/COUNTBUSINV(0)  3722      0     51500000     48499900          100
```

Figure 4.6: *Power Report complessivo, caso 'data'*

4.1.3 T0 techinque

L'ultima tecnica richiesta è di implementare un codice per realizzare la tecnica *T0 Encoding*. Nella traccia viene riportata la formula richiesta da implementare, riportata anche in Figura 4.7.

$$\left(B^{(t)}, INC^{(t)}\right) = \begin{cases} \left(B^{(t-1),1}\right) & if \quad b^{(t)} = b^{(t-1)} + 1 \\ \left(b^{(t)}, 0\right) & otherwise \end{cases}$$

Figure 4.7: *Formula implementata, tecnica 'T0'*

La codifica T0 consiste nell'inserire un bit in più, denominato *INC*. Ogni volta che si va in sequenza, si forza questo segnale a '0' in modo tale che il ricevitore capisca di dover andare semplicemente in sequenza e si preoccupi lui di incrementare il dato. In caso di brach, salito o subroutine, forzo il segnale ad '1' e trasmetto sul bus il nuovo dato. Si riporta in seguito il *Codice VHDL* dell'implementazione della codifica.

4.1.4 Confronto tra le tecniche

Confrontando i risultati tra le varie tecniche, si può notare come l'unica tecnica che porta ad una riduzione dei consumi nel caso dei dati, risulta essere la **Bus-Invert** con la quale si ottiene un risparmio sulle commutazioni pari a circa il 20%.
Per quanto concerne la tramissione di indirizzi, le codifiche più perfomanti risultano la **Codifica di Gray**, che porta ad un risparmio sulle commutazioni del 50%, e la **Codifica T0**, che porta ad un risparmio del XX%.
Si può notare come la tecnica che risulta meno performante è la **Transition Based**, che porta ad un incremento del 93% nel caso di dati e ad un incremento circa del 0,1% nel caso di indirizzi. Ovviamente questo è concorde con quanto ci si aspettava, in quanto, come già detto prima, questa tipologia di decofica funziona efficentemente solo nel caso in cui le probabilità di '1' e di '0' logico non risultino sbilanciate.

4.2 Synthesis