



UNIVERSITÀ DEGLI STUDI DI PADOVA

SCUOLA DI INGEGNERIA

Corso di Laurea Magistrale in Ingegneria Elettronica

**CONVERTITORE DC-DC A CAPACITA' COMMUTATE
MULTIFASE E AUTO-RICONFIGURABILE CON
RISOLUZIONE BINARIA E DI FIBONACCI**

Laureando

Alberto Bevilacqua

Relatore

Prof. Andrea Gerosa

ANNO ACCADEMICO 2016/2017

A tutti coloro che mi son stati accanto in questi anni.

Do not worry about your difficulties in Mathematics.
I can assure you mine are still greater. *Albert Einstein*

Indice

1	Introduzione	1
1.1	Energy Harvesting	2
1.2	Convertitore DC-DC	5
1.3	EXB - SFN	9
2	Struttura del chip	13
2.1	Switched Capacitor Converter - SCC	15
2.1.1	Convertitore 8/2	16
2.1.2	Convertitore 3/1	18
2.1.3	Convertitore 8/3	20
2.1.4	Convertitore 5/2	23
2.1.5	Convertitore 8/4	25
2.1.6	SCC completo e dimensionamento	27
2.1.7	Modello medio con perdite	35
2.1.8	Dimensionamento dei C_{fly}	39
2.1.9	Buffer SCC	41
2.2	Phase Divider e Logic Control Unit	43
2.3	Gear Control e Buffer	53
2.4	Hysteretic Control	55
2.5	Start-up	57
2.6	Power Good	61
3	Simulazioni	63
3.1	Verifica funzionamento convertitori	64
3.2	Verifica funzionamento Gear Control	114
3.3	Verifica funzionamento controllo isteretico	115
3.4	Efficienza	117
3.5	Simulazione start-up	119
4	Conclusioni	123

A	Ridondanza codice EXB	125
B	Codici MatLab	129
B.1	Resistenza dei MOSFET e dimensionamento switch SCC . . .	129
B.2	Resistenze del gear control	154
B.3	Modelli a regime dei convertitori	155
	Bibliografia	161

Capitolo 1

Introduzione

Negli ultimi decenni il mondo dell'elettronica integrata ha avuto un notevole sviluppo nel settore della potenza di calcolo, dell'efficienza, e del costo del singolo dispositivo. Tuttavia si sta facendo sempre più strada l'idea di un mondo dominato dall'*Internet of Things* - (*IoT*) cioè un mondo nel quale anche gli oggetti sono connessi in una rete di comunicazione e interagiscono tra di loro. Se dal lato filosofico l'IoT risulta essere molto interessante, dal lato elettronico questo pone qualche interrogativo. Sebbene ci siano già dispositivi in commercio in grado di svolgere i compiti richiesti (processori TI SitaraTM), resta il problema della fonte energetica per alimentare tali dispositivi in quei casi dove non è possibile utilizzare la semplice rete elettrica. È proprio su quest'ultimo punto che nasce l'idea di utilizzare l'energia proveniente dall'ambiente e convertirla in energia elettrica per alimentare tali dispositivi. In questo modo si potrebbe evitare l'utilizzo di pile sia per questioni ecologiche che per durata; dispositivi che compiono questa trasformazione vengono detti *energy harvester*. Tuttavia serve anche un dispositivo elettronico in grado di convertire la tensione e corrente proveniente dall'ambiente e adattarlo al microprocessore: un *convertitore dc-dc*.

1.1 Energy Harvesting

L'*energy harvesting* (o *energy scavenging*), come detto in precedenza, è quel processo per il quale viene estratta l'energia fornita da fonti energetiche alternative. Le sorgenti di energia alternative sono tutte disponibili dall'ambiente come:

- *sorgenti radio*: una possibile forma di energia dell'ambiente deriva proprio dalle onde elettromagnetiche che viaggiano nell'aria. Un tipico esempio di questo utilizzo di questa tecnologia sono gli identificatori a radiofrequenza (*RFID*). L'antenna che riceve l'onda elettromagnetica alimenta anche il circuito elettrico.

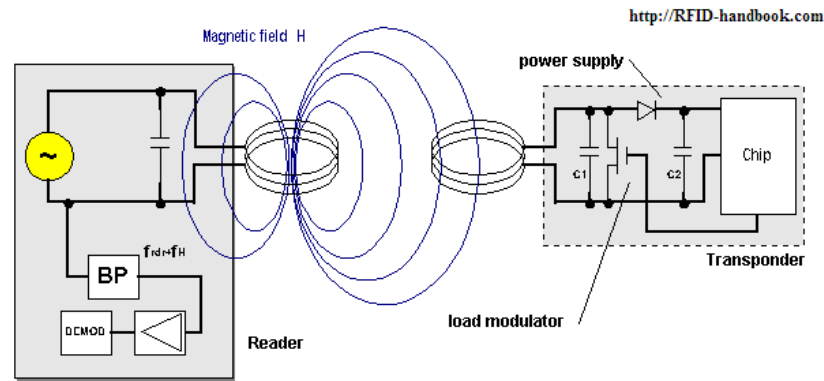


Figura 1.1: Schema di funzionamento dei chip RFID.

- *fotovoltaico*: capace di trasformare l'energia del fotone che impatta sulla cella fotovoltaica in energia elettrica, molto variabile in base alle condizioni climatiche.

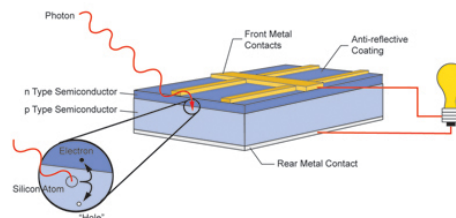


Figura 1.2: Schema di funzionamento della cella fotovoltaica.

- *cinetico*: grazie alla proprietà di alcuni cristalli piezoelettrici è possibile estrarre energia elettrica mediante stress meccanico del cristallo stesso. Tuttavia oltre che per stress è possibile estrapolare energia anche da movimenti. Una modalità è quella di utilizzare l'energia cinetica di un pendolo magnetico immerso in un avvolgimento elettromagnetico. Le forze che mettono in moto il pendolo possono derivare da vibrazioni (motori) o sollecitazioni periodiche (per esempio: battito cardiaco e movimento del piede durante una camminata). La figura 1.3 evidenzia la struttura interna di tale dispositivo e il suo funzionamento.

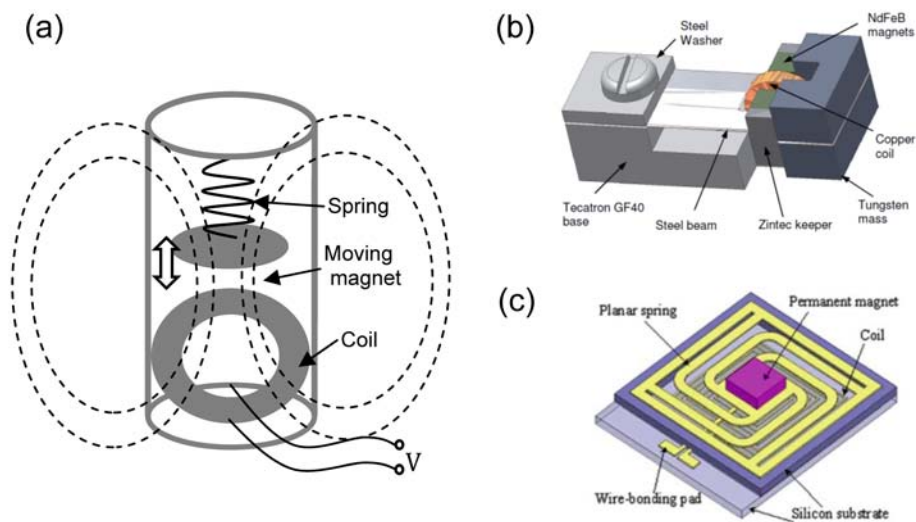


Figura 1.3: Schema di funzionamento di un energy harvester che utilizza la tecnica del pendolo.

- *termoelettrico*: sfrutta il gradiente termico per generare una tensione tipica di $0.1\text{-}0.2\text{mV/K}$. Mettendo in serie numerose di queste sorgenti si possono ottenere tensioni utili per mondo dell'elettronica. Le potenze in gioco si aggirano attorno ai milliWatt. Si può utilizzare anche il corpo umano o animale. L'*effetto Seebeck* è alla base di questa tecnologia. La figura 1.4 mostra la struttura della classica cella di Peltier, una tipologia di questo genere di sorgenti di energia.

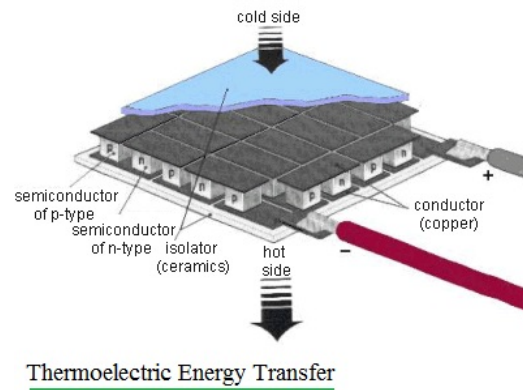


Figura 1.4: Schema di funzionamento di un energy harvester termoelettrico.

La tabella sottostante riassume i range di potenza estraibile in base alle diverse tipologie di energy harvester:

<i>Sorgente</i>	<i>Densità di potenza</i>
<i>Sorgenti radio</i>	
GSM	$0.1\mu W/cm^2$
WiFi	$0.001\mu W/cm^2$
<i>Fotovoltaico</i>	
interno	$10\mu W/cm^2$
esterno	$10mW/cm^2$
<i>Cinetico</i>	
umano	$4\mu W/cm^2$
industriale	$100\mu W/cm^2$
<i>Termoelettrico</i>	
umano	$25\mu W/cm^2$
industriale	$1 - 10mW/cm^2$

Tabella 1.1: Tabella riassuntiva delle densità di potenza erogate dalle varie fonti energetiche per unità di area.

1.2 Convertitore DC-DC

Il convertitore dc-dc è un dispositivo in grado di effettuare una conversione della tensione in ingresso in base ad un ben specifico rapporto di conversione:

$$V_{out} = V_{in} \cdot M$$

dove M può essere maggiore di uno (convertitore *step-up*) oppure minore di uno (convertitore *step-down*).

Esistono due famiglie di convertitori: *lineari* e *switching*. I primi si basano su un riferimento interno PVT (per esempio band-gap) e poi tramite una retroazione interna viene regolata la tensione di uscita. La seconda famiglia di convertitori a sua volta si divide in altre due categorie: *magnetici* dove si ha la presenza di un induttore e *capacitivi* dove non ci sono induttori, ma solo condensatori e transistor.

Nel mondo dei circuiti integrati l'induttore è un componente realizzabile, ma occupa molto spazio (poi varia a seconda del valore di induttanza) e provoca campi elettromagnetici (EMI) che potrebbero causare effetti negativi sul resto del circuito. I convertitori senza induttori vengono anche comunemente chiamati a *capacità commutate* (*switched capacitor converter - scc*) e si basano sul fatto che i condensatori vengono caricati in una fase e scaricati in un'altra. A seconda di come vengono caricati e scaricati si possono ottenere i diversi rapporti di conversione. Per il fatto che i condensatori non hanno una posizione fissa nello schematico vengono chiamati *condensatori flottanti* C_{fly} . Gli switch vengono comunemente implementati dai transistor MOSFET.

Dalla letteratura si sa che il modello medio di un convertitore è rappresentato da un generatore di tensione d'ingresso, un trasformatore con un preciso rapporto di trasformazione, una resistenza di uscita R_{out} data dai fenomeni parassiti e poi la resistenza di carico R_{load} . La figura 1.5 mostra il modello medio appena descritto:

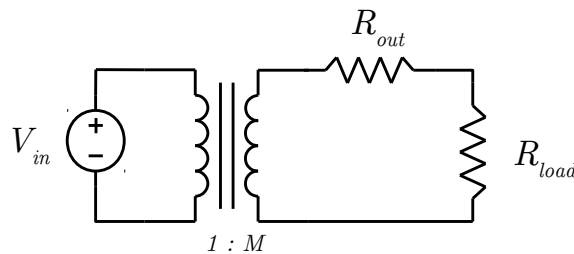


Figura 1.5: Modello medio del convertitore dc-dc switching.

Questo tipo di circuiti ha in generale due modi operativi: *regime di slow switching limit* - *SSL* ed il *regime di fast switching limit* - *FSL*. Il circuito lavora in uno dei due modi a seconda della relazione tra la costante di tempo RC delle fasi e la durata temporale di ciascuna fase:

$$R_{\Phi_i} \cdot C_{eq} < D_{\Phi_i} \cdot T \quad \rightarrow \quad SSL \quad (1.1)$$

$$R_{\Phi_i} \cdot C_{eq} \gg D_{\Phi_i} \cdot T \quad \rightarrow \quad FSL \quad (1.2)$$

La differenza sostanziale tra i due regimi operativi sta nel fatto che in *SSL* le resistenza degli interruttori, la resistenza serie dei condensatori e quella delle interconnessioni possono essere trascurate e la corrente ha andamenti impulsivi (poichè i C_{fly} hanno tutto il tempo di caricarsi e scaricarsi). Invece in *FSL* sono proprio le resistenze degli switch e tutte le altre resistenze parassite a dominare. In questo regime la corrente che scorre nel circuito è costante a tratti perchè i condensatori hanno pochissimo tempo per caricarsi e scaricarsi ($R_{\Phi_i} \cdot C_{eq} \gg D_{\Phi_i} \cdot T$). Sapendo che la corrente è la derivata temporale della tensione sul condensatore:

$$i_{C_{fly}} = C \frac{\partial V_{C_{fly}}}{\partial t}$$

allora la tensione varia linearmente. Questo risultato è confermato dal fatto che il classico andamento esponenziale di carica e scarica del condensatore può essere approssimato ad una retta per intervalli di tempo sufficientemente piccoli.

Analizzando i flussi di carica nelle varie fasi si può ricavare il vettore dei coefficienti di carica $\mathbf{a}$ (ovvero un vettore che tiene conto di quanta carica $q_i^{\Phi_j}$ scorre in un determinato ramo del circuito nelle varie fasi e normalizzato sulla carica di uscita $q_{out} = \sum_{i=1}^{\#phases} q_{out}^{\Phi_i}$). Con questo vettore è possibile ricavare le formule della resistenza di uscita R_{out} del convertitore nei due regimi operativi:

$$R_{SSL} = -\frac{v_{out}}{i_{out}} = \sum_{i \in caps} \sum_{j=1}^n \frac{(a_{c,i}^j)^2}{2C_{fly-i}f_{sw}} \quad (1.3)$$

$$R_{FSL} = -\frac{v_{out}}{i_{out}} = \sum_{i \in switches} \sum_{j=1}^n (a_{c,i}^j)^2 \frac{R_i}{D_j} \quad (1.4)$$

Come si può notare la differenza è sostanziale al variare della frequenza di lavoro: nella 1.3 la resistenza di uscita è inversamente proporzionale alla frequenza di lavoro, invece nella 1.4 la resistenza di uscita non varia al variare della f_{sw} . Quindi si potrebbe dire che le due formule calcolate rappresentano

i due asintoti (per bassa e alta frequenza) dell'andamento della resistenza di uscita. La figura 1.6 mostra l'andamento appena descritto:

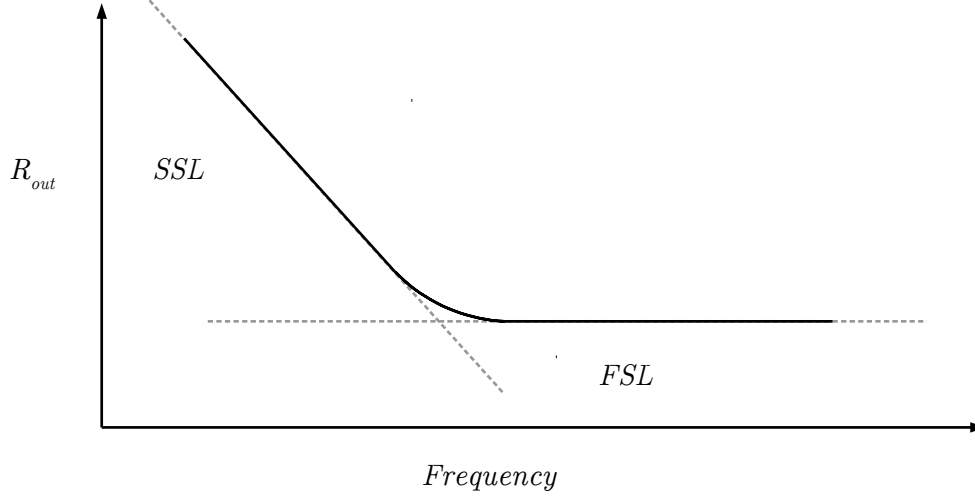


Figura 1.6: Andamento della resistenza di uscita del convertitore in funzione della frequenza di lavoro.

Le linee tratteggiate rappresentano i due asintoti descritti dalle equazioni, mentre la curva nera rappresenta il vero andamento della R_{out} . Come si può ben notare esiste una regione di raccordo tra i due tratti asintotici e in quella regione il regime operativo diventa ibrido, dove la costante di tempo RC è confrontabile con la durata della singola fase.

L'efficienza del sistema si può ricavare osservando la figura 1.5:

$$\eta = \frac{V_{out}}{M \cdot V_{in}} = \frac{M \cdot V_{in}}{M \cdot V_{in}} \cdot \frac{R_{load}}{R_{out} + R_{load}} = \frac{R_{load}}{R_{out} + R_{load}} \quad (1.5)$$

dall'equazione sovrastante risulta evidente che nel regime operativo di FSL (cioè quando si ha il minimo valore di R_{out}) l'efficienza risulta essere maggiore ed è per questo motivo che in questo progetto di tesi si è deciso di far lavorare il convertitore in FSL . Un'altra considerazione sull'efficienza è che se la tensione di uscita deve essere tenuta costante (da un eventuale regolatore) mentre la tensione di ingresso sale allora osservando la 1.5 si nota che l'efficienza cala. Se il range della tensione di ingresso è molto ampio allora conviene utilizzare diversi rapporti di conversione. Graficando l'andamento dell'efficienza in funzione della tensione di ingresso con diversi rapporti di conversione si ottiene una forma del tipo:

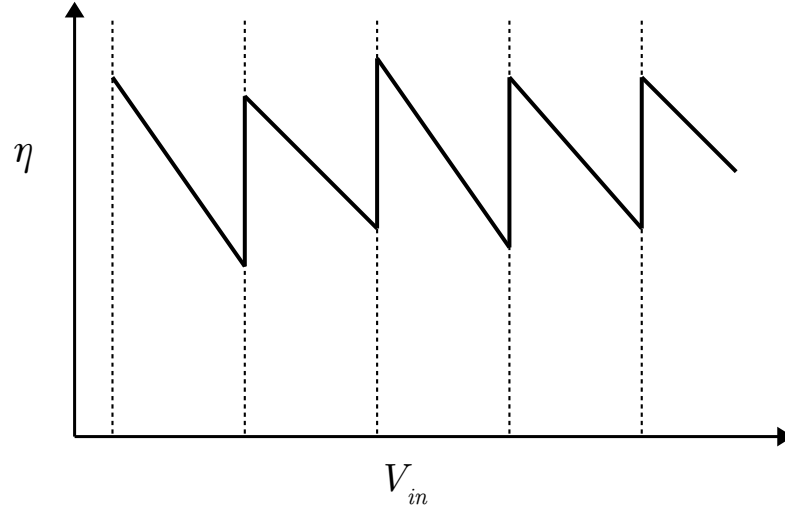


Figura 1.7: Andamento dell'efficienza in funzione della V_{in} con 5 rapporti di conversione.

Osservando la figura 1.7 si può intuire che minore è la distanza tra i rapporti di conversione, migliore sarà l'efficienza globale del sistema. Quindi integrare in uno stesso chip più rapporti di conversione comporterebbe un sistema molto efficiente, a patto di avere una logica di controllo adeguata e poco costosa.

Nella letteratura si trovano diverse topologie di convertitori utilizzabili: Ladder, Dickson, Serie-parallelo, Fibonacci. Tuttavia in questo progetto di tesi si è utilizzato un approccio diverso per la costruzione delle topologie delle fasi che si basa su un particolare tipo di algoritmo che verrà spiegato nella prossima sezione. L'idea di utilizzare una tecnica diversa nasce dalla motivazione espressa in precedenza di integrare il maggior numero possibile di rapporti di conversione nello stesso chip. Utilizzando questa topologia e avendo a disposizione gli stessi C_{fly} e stessi switch si posso realizzare diversi rapporti di conversione.

1.3 EXB - SFN

Extended Binary Representation

La *rappresentazione binaria estesa* (EXB) è una tecnica che permette di esprimere un qualunque numero M_n come un numero binario, ma con la differenza di poter aggiungere usare anche la cifra -1 :

$$M_n = A_0 + \sum_{j=1}^n A_j 2^{-j}$$

dove $A_0 \in \{0, 1\}$, invece $A_j \in \{-1, 0, 1\}$. Il parametro n imposta la *risoluzione*. Per esempio il numero $7/8$ nella rappresentazione EXB è $\{0111\}$. Infatti:

$$M_3 = 0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} = 0 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} = \frac{7}{8}$$

si noti che per il numero scelto come esempio la risoluzione deve essere $n = 3$. A differenza della versione binaria, in questa rappresentazione lo stesso numero M_n può avere diversi numeri EXB. L'algoritmo che permette di trovare gli stessi numeri EXB di un unico numero M_n consiste nel sommare 1 (somma binaria) ad una qualsiasi cifra 1 del codice originario. Visto che si è sommato 1 per mantenere lo stesso M_n si sottrae -1 nella stessa posizione in cui si è aggiunto 1. In breve ad un numero EXB si sostituisce $A_j = 1$ con $A_j = -1$ e si somma 1 ad A_{j-1} . Per esempio, partendo sempre dal numero $M_3 = 7/8$ che nella versione EXB vale $\{10 - 11\}$ e lavorando su A_3 :

				↓
	2^0	2^{-1}	2^{-2}	2^{-3}
	0	1	1	1
+	0	0	0	1
	1	0	0	0
+	0	0	0	-1
	1	0	0	-1

Iterando il procedimento si ottengono gli altri numeri equivalenti:

$$\begin{array}{cccc}
& & \downarrow & \\
& 2^0 & 2^{-1} & 2^{-2} \quad 2^{-3} \\
& \hline
& \mathbf{0} & \mathbf{1} & \mathbf{1} \quad \mathbf{1} \\
+ & \hline
& 0 & 0 & 1 \quad 0 \\
& \hline
& 1 & 0 & 0 \quad 1 \\
+ & \hline
& 0 & 0 & -1 \quad 0 \\
& \hline
& \mathbf{1} & \mathbf{0} & \mathbf{-1} \quad \mathbf{1}
\end{array}
\qquad
\begin{array}{cccc}
& & \downarrow & \\
& 2^0 & 2^{-1} & 2^{-2} \quad 2^{-3} \\
& \hline
& \mathbf{0} & \mathbf{1} & \mathbf{1} \quad \mathbf{1} \\
+ & \hline
& 0 & 1 & 0 \quad 0 \\
& \hline
& 1 & 0 & 1 \quad 1 \\
+ & \hline
& 0 & -1 & 0 \quad 0 \\
& \hline
& \mathbf{1} & \mathbf{-1} & \mathbf{1} \quad \mathbf{1}
\end{array}$$

Avendo eseguito il procedimento su tutti e tre gli $A_j = 1$ del numero iniziale e se si prova ad iterare il procedimento sui numeri finali ottenuti si ottengono risultati già noti allora questo significa che tutti i numeri EXB equivalenti son stati trovati e sono:

A_0	A_1	A_2	A_3
0	1	1	1
1	0	0	-1
1	0	-1	1
1	-1	1	1

Tabella 1.2: Tabella riassuntiva dei numeri EXB che rappresentano $M_3 = 7/8$.

Questi numeri si possono tradurre in circuiti elettrici imponendo un numero di condensatori flottanti pari al numero della risoluzione imposta e seguendo queste regole:

- Se $A_0 = 1$ il generatore V_{in} è connesso in modo tale da caricare l'uscita, altrimenti ($A_0 = 0$) non è connesso;
- Se $A_j = -1$ il condensatore C_{fly-j} è connesso in serie all'uscita con la stessa polarità;
- Se $A_j = 0$ il condensatore C_{fly-j} non è connesso;
- se $A_j = 1$ il condensatore C_{fly-j} è connesso in serie all'uscita, ma con polarità opposta.

Tenendo fissa la risoluzione $n = 3$ i numeri che si riescono a rappresentare con questa tecnica sono: $1/8, 2/8, 3/8, 4/8, 5/8, 6/8$ e $7/8$. Se si avesse bisogno di una risoluzione più fine bisogna imporre $n = 4$ o maggiore, ma questo richiederebbe un numero maggiore di condensatori C_{fly} e una logica di controllo più complessa. Tuttavia esiste un'altra strada per aumentare la risoluzione ma tenendo fisso il numero di condensatori flottanti: introdurre un'ulteriore rappresentazione numerica.

Signed Fibonacci Representation

La serie di Fibonacci si costruisce partendo da $F_1 = F_2 = 1$ e per $i \geq 3$ $F_i = F_{i-1} + F_{i-2}$. Il teorema di Zeckendorf dimostra che ogni numero può essere espresso univocamente come somma di soli numeri appartenenti alla serie di Fibonacci:

$$N_n = \sum_{j=0}^n A_j F_{n-j+2}$$

dove $A_j \in \{0, 1\}$ ed n ancora una volta rappresenta la risoluzione. Normalizzando N_n con F_{n+2} , imponendo che $A_j \in \{-1, 0, 1\}$ e $A_0 \in \{0, 1\}$ si ottiene la *signed Fibonacci representation*:

$$M_n = A_0 + \sum_{j=0}^n A_j \frac{F_{n-j+2}}{F_{n+2}}$$

la modifica che A_j possa valere anche -1 elimina l'univocità del teorema di Zeckendorf e consente di avere più numeri SFN che rappresentano lo stesso numero. L'algoritmo è lo stesso del paragrafo precedente e anche la sua traduzione in schemi elettrici. Analizzando il caso con risoluzione pari a $n = 3$ allora i numeri rappresentabili sono: $1/5, 2/5, 3/5$ e $4/5$.

Sarebbe possibile generalizzare la sequenza dei numeri di Fibonacci creando la *signed generalized Fibonacci representation* (SGF):

$$M_n = A_0 + \sum_{j=1}^n A_j \frac{F_{n-j+1}}{F_{n+1}}$$

questo permetterebbe di aggiungere ulteriori rapporti di conversione. Tenendo sempre come esempio la risoluzione $n = 3$ si avrebbe: $1/7, 2/7, 3/7, 4/7, 5/7$ e $6/7$.

La figura 1.8 mostra l'andamento dell'efficienza in funzione dei vari rapporti di conversione finora descritti; la linea blu descrive i rapporti di conversione trovati mediante EXB, quella verde rappresenta i rapporti di con-

versione calcolati mediante SFN e infine quella rossa mostra quelli calcolati mediante SGF.

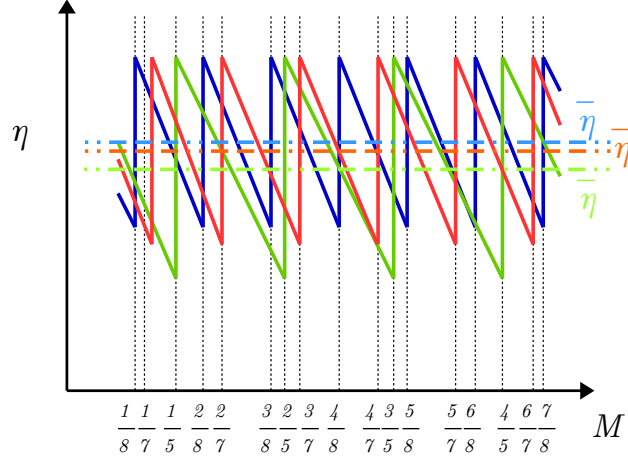


Figura 1.8: Andamento dell'efficienza in funzione dei vari rapporti di conversione con risoluzione $n = 3$.

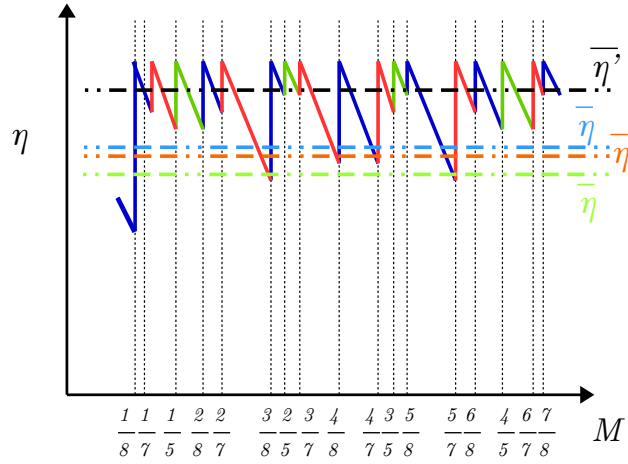


Figura 1.9: Andamento dell'efficienza in un ipotetic chip con tutti i rapporti di conversione dell'immagine 1.8.

Dall'immagine 1.9 si nota chiaramente come l'efficienza media $\bar{\eta}'$ sia maggiore rispetto alle singole efficienze medie rappresentate nella figura 1.8.

Capitolo 2

Struttura del chip

Il nucleo del progetto è il convertitore a capacità commutate, ma per pilotare gli switch sono necessari ulteriori blocchi che siano in grado di gestire le diverse fasi dei convertitori e i diversi rapporti di conversione oltre al fatto di saper regolare finemente la tensione di uscita. L'architettura proposta è illustrata nell'immagine seguente:

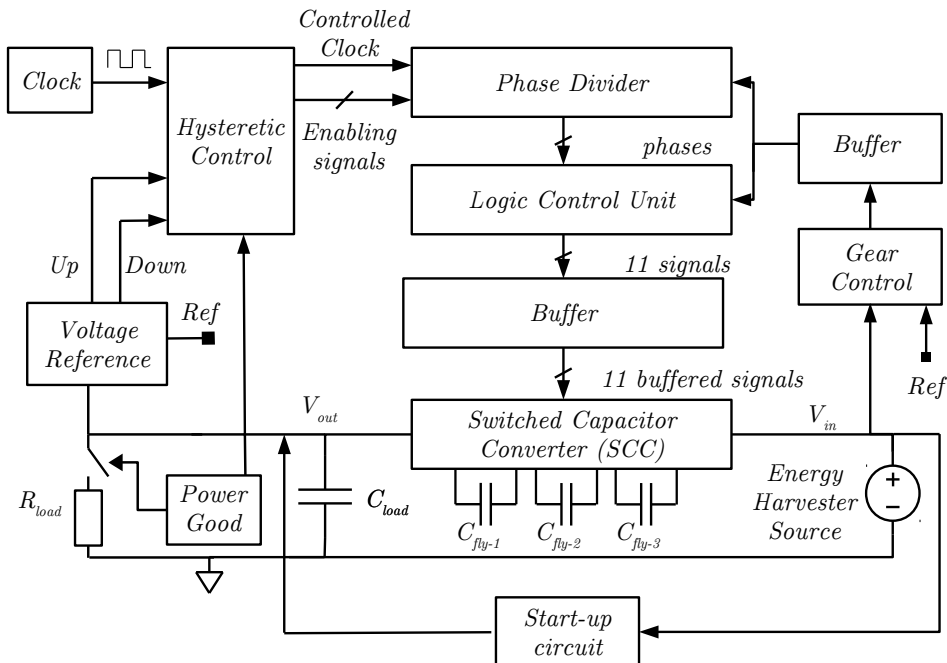


Figura 2.1: Struttura a livello di sistema del progetto.

Il progetto prevede un blocco chiamato *Start-up* che viene direttamente collegato alla sorgente di energia e al quale è affidato il compito di caricare il condensatore di uscita C_{load} ad una tensione tale che il resto del circuito riesca ad auto-alimentarsi e raggiungere il regime stazionario di funzionamento. Questo blocco è fondamentale per il fatto che non si hanno disposizione delle batterie nemmeno nella fase di accensione.

Una volta che il circuito si è avviato allora intervengono anche gli altri blocchi:

- *Gear Control*: confronta la tensione di ingresso V_{in} con diverse soglie e seleziona il rapporto di conversione per il convertitore.
- *Buffer*: prende il segnale dal *Gear Control* e lo rigenera attraverso un buffer CMOS in modo che riesca a pilotare carichi capacitivi maggiori.
- *Phase Divider*: blocco che ha il compito di generare un numero di fasi non sovrapposte adeguato al fattore di conversione selezionato. E' il primo elemento ad essere disabilitato quando il controllore lo richiede.
- *Logic Control Unit*: a seconda del convertitore e della fase corrente seleziona la giusta sequenza di interruttori da attivare nel convertitore.
- *Buffer*: rigenera i segnali provenienti dalla logica di controllo in modo che possano pilotare i grossi carichi capacitivi degli switch.
- *Switched Capacitor Converter*: nucleo centrale del progetto ed è colui che trasforma la tensione di ingresso V_{in} nella tensione di uscita V_{out} secondo un ben preciso fattore di conversione.
- *Hysteric Control*: fa un sensing della tensione di uscita e confrontandola con due soglie imposte va a generare un segnale di abilitazione per l'intera circuiteria.
- *Clock*: generatore di clock a frequenza costante.
- *Voltage Reference*: genera dei riferimenti di tensione insensibili alle variazioni di temperatura e di processo.
- *Power Good*: blocco che ha il compito di fare un sensing della tensione di uscita e genera un segnale di abilitazione per il carico quando il circuito è a regime.

In seguito verranno analizzati e spiegati in dettaglio i vari blocchi.

2.1 Switched Capacitor Converter - SCC

Come detto in precedenza questo blocco rappresenta il cuore dell'intero sistema in quanto è proprio questo scc che esegue lo step-up della tensione di uscita seguendo un preciso fattore di conversione.

Sapendo che i convertitori a capacità commutate eseguono dei rapporti di trasformazione discreti e che da specifica si aveva che il range della tensione di ingresso era: $300mV - 600mV$ con una tensione di uscita stabile di $1.2V$ allora risulta immediato che gli estremi dei fattori di conversione diventino:

$$Inferiore : \frac{V_{out}}{V_{in-low}} = \frac{1.2}{0.3} = 4 \quad Superiore : \frac{V_{out}}{V_{in-up}} = \frac{1.2}{0.6} = 2 \quad (2.1)$$

In base al procedimento descritto in precedenza per la costruzione del convertitore secondo *EXB* o *SNF* si è deciso di utilizzare una risoluzione $M=3$, con 3 condensatori flottanti C_{fly} . Tuttavia sapendo che con questo numero di condensatori si possono ottenere al massimo $2^{num_{C_{fly}}} - 1 = 2^3 - 1 = 7$ fattori di conversione si è comunque deciso di utilizzare solo 5 dei rapporti per semplificare la complessità della logica di controllo. Quindi i fattori di conversione utilizzati sono:

$$4 = \frac{8}{2}, \quad \frac{3}{1}, \quad \frac{8}{3}, \quad \frac{5}{2}, \quad \frac{8}{4} = 2.$$

In seguito verranno analizzati tutti e 5 i convertitori per poi arrivare ad un circuito comune.

La procedura seguita nella creazione e dimensionamento del scc è stata:

- creazione dei codici e la loro traduzione nei circuiti elettrici
- analisi dei flussi di carica e calcolo dei vettori dei coefficienti di carica
- analisi delle varie configurazioni mediante la *KVL* e calcolo delle tensioni medie ai capi dei vari condensatori flottanti C_{fly}
- dimensionamento per ogni singolo convertitore e dimensionamento comune
- verifica tramite simulazione.

La potenza richiesta in uscita è di $1mW$.

2.1.1 Convertitore 8/2

Utilizzando l'algoritmo descritto nell'introduzione e applicandolo al fattore di conversione step-down $\frac{2}{8}$ si ottiene:

A_0	A_1	A_2	A_3
1	-1	-1	0
0	1	-1	0
0	0	1	0

Tabella 2.1: Tabella riassuntiva dei codici EXB per il numero 2/8.

Quindi le fasi finali sono:

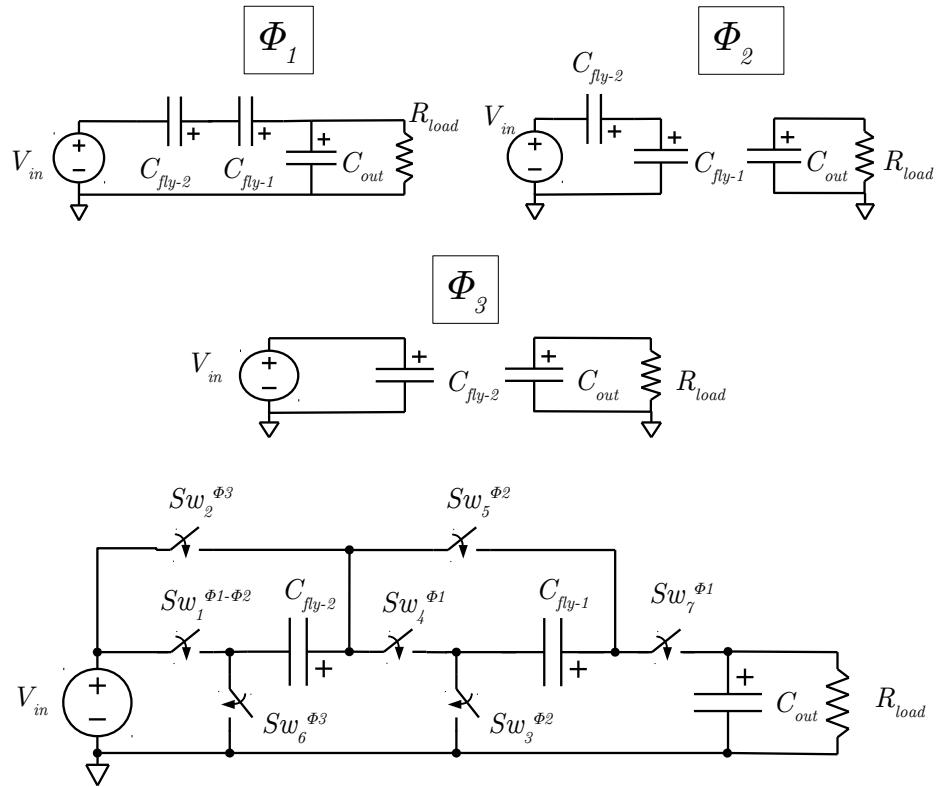


Figura 2.2: Fasi e circuito completo per il convertitore step-up 8/2.

Come già detto anche in precedenza prendendo le fasi del convertitore step-down e scambiando la porta d'ingresso con quella di uscita si ottiene la versione step-up 8/2.

Al fine di dimensionare in modo ottimale la dimensione degli switch, la capacità dei condensatori C_{fly} , la frequenza di lavoro e imporre che tutto sia in *FSL* risulta necessario calcolare i vettori dei coefficienti di carica. Questi vettori vengono estrapolati andando ad analizzare i flussi di carica nelle varie fasi:

$$\begin{aligned} \begin{cases} q_{out} = -q_{C1}^{\Phi_1} = -q_{C2}^{\Phi_1} \\ q_{C1}^{\Phi_1} = -q_{C1}^{\Phi_2} = q_{C2}^{\Phi_2} \\ q_{C2}^{\Phi_1} + q_{C2}^{\Phi_2} + q_{C2}^{\Phi_3} = 0 \end{cases} & \xrightarrow[\cdot \frac{1}{q_{out}}]{} \begin{cases} 1 = -a_{C1}^{\Phi_1} = -a_{C2}^{\Phi_1} \\ a_{C1}^{\Phi_1} = -a_{C1}^{\Phi_2} = a_{C2}^{\Phi_2} \\ a_{C2}^{\Phi_1} + a_{C2}^{\Phi_2} + a_{C2}^{\Phi_3} = 0 \end{cases} \\ & \rightarrow \begin{cases} a_{C1}^{\Phi_1} = a_{C2}^{\Phi_1} = -1 \\ a_{C1}^{\Phi_2} = -a_{C2}^{\Phi_2} = 1 \\ a_{C2}^{\Phi_3} = 2 \end{cases} \end{aligned} \quad (2.2)$$

Partendo da coefficienti appena ottenuti e analizzando il circuito completo in figura 2.2 si ottiene:

Φ	Sw_1	Sw_2	Sw_3	Sw_4	Sw_5	Sw_6	Sw_7
Φ_1	1	0	0	1	0	0	1
Φ_2	1	0	-1	0	1	0	0
Φ_3	0	2	0	0	0	-2	0

Tabella 2.2: Tabella riassuntiva dei coefficienti per gli switch.

Se si considera la legge di Kirchhoff alle maglie per ciascuna fase si trova la tensione teorica che dovrebbe cadere ai capi dei due condensatori C_{fly} :

$$\begin{cases} V_{out} = V_{in} + V_{C1} + V_{C2} \\ V_{in} + V_{C2} = V_{C1} \\ V_{in} = V_{C2} \end{cases} \rightarrow \begin{cases} V_{out} = 4 \cdot V_{in} \\ V_{C1} = 2 \cdot V_{in} \\ V_{C2} = V_{in} \end{cases} \quad (2.3)$$

2.1.2 Convertitore 3/1

Nel caso del convertitore step-down $\frac{1}{3}$ i codici ottenuti con l'algoritmo sono:

A_0	A_1	A_2
0	0	1
0	1	-1
1	-1	0

Tabella 2.3: Tabella riassuntiva dei codici EXB per il numero 1/3.

Quindi le fasi finali sono:

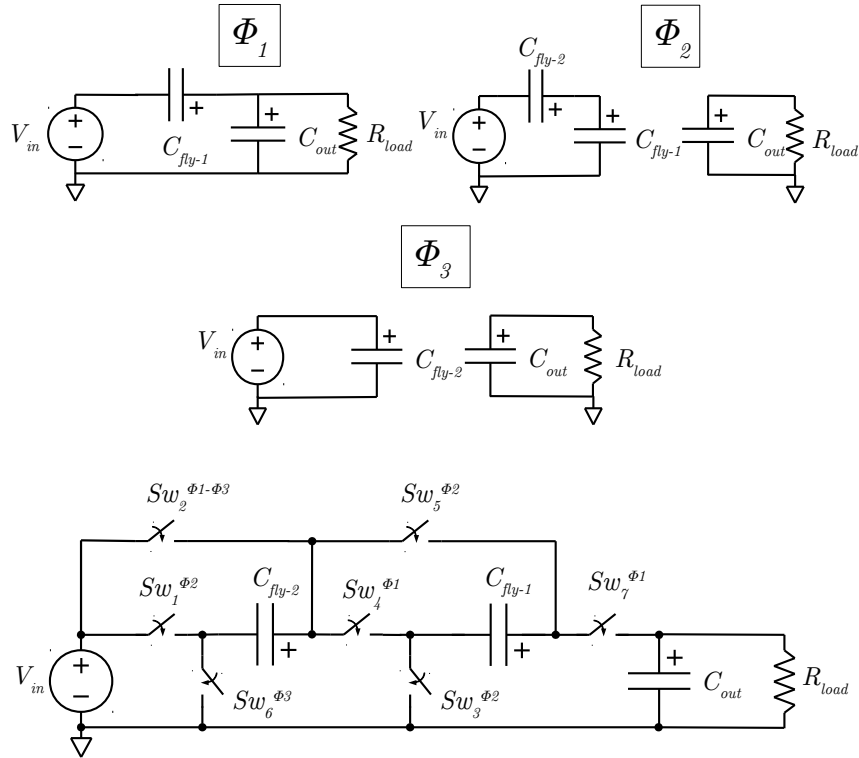


Figura 2.3: Fasi e circuito completo per il convertitore step-up 3/1.

Come già detto anche in precedenza se si invertono le porte di ingresso con quelle di uscita si ottiene la versione step-up e il rapporto di conversione non altro che il reciproco della versione step-down: in questo caso diventa $3/1$.

Per dimensionare gli switch, la capacità dei condensatori C_{fly} , la frequenza di lavoro e imporre che tutto sia in FSL , risulta necessario calcolare i vettori dei coefficienti di carica. Questi vettori vengono estrapolati andando ad analizzare i flussi di carica nelle varie fasi:

$$\begin{aligned} \begin{cases} q_{out} = -q_{C1}^{\Phi_1} = q_{C1}^{\Phi_2} \\ q_{C1}^{\Phi_2} = -q_{C2}^{\Phi_2} \\ q_{C2}^{\Phi_2} = -q_{C2}^{\Phi_3} \end{cases} & \xrightarrow[\cdot \frac{1}{q_{out}}]{} \begin{cases} 1 = -a_{C1}^{\Phi_1} = a_{C1}^{\Phi_2} \\ a_{C1}^{\Phi_2} = -a_{C2}^{\Phi_2} \\ a_{C2}^{\Phi_2} = -a_{C2}^{\Phi_3} \end{cases} \\ & \rightarrow \begin{cases} a_{C1}^{\Phi_1} = -a_{C1}^{\Phi_2} = -1 \\ a_{C1}^{\Phi_2} = -a_{C2}^{\Phi_2} = 1 \\ a_{C2}^{\Phi_2} = -a_{C2}^{\Phi_3} = 1 \end{cases} \end{aligned} \quad (2.4)$$

Considerando la sequenza di switch per ogni fase e i risultati ottenuti in 2.3 si trova che:

Φ	Sw_1	Sw_2	Sw_3	Sw_4	Sw_5	Sw_6	Sw_7
Φ_1	0	1	0	1	0	0	1
Φ_2	1	0	0	0	1	1	0
Φ_3	0	1	1	0	0	0	0

Tabella 2.4: Tabella riassuntiva dei coefficienti per gli switch.

Se per ogni singola fase si applica la KVL è possibile ricavare la tensione media ai capi dei condensatori flottanti:

$$\begin{cases} V_{out} = V_{in} + V_{C1} \\ V_{C1} = V_{in} + V_{C2} \\ V_{C2} = V_{in} \end{cases} \rightarrow \begin{cases} V_{out} = 3 \cdot V_{in} \\ V_{C1} = 2 \cdot V_{in} \\ V_{C2} = V_{in} \end{cases} \quad (2.5)$$

2.1.3 Convertitore 8/3

I codici per il convertitore $\frac{3}{8}$ estrapolati sempre mediante il medesimo algoritmo sono:

A_0	A_1	A_2	A_3
1	-1	-1	1
0	1	-1	1
1	-1	0	-1
0	1	0	-1
0	0	1	1

Tabella 2.5: Tabella riassuntiva dei codici EXB per il numero 3/8.

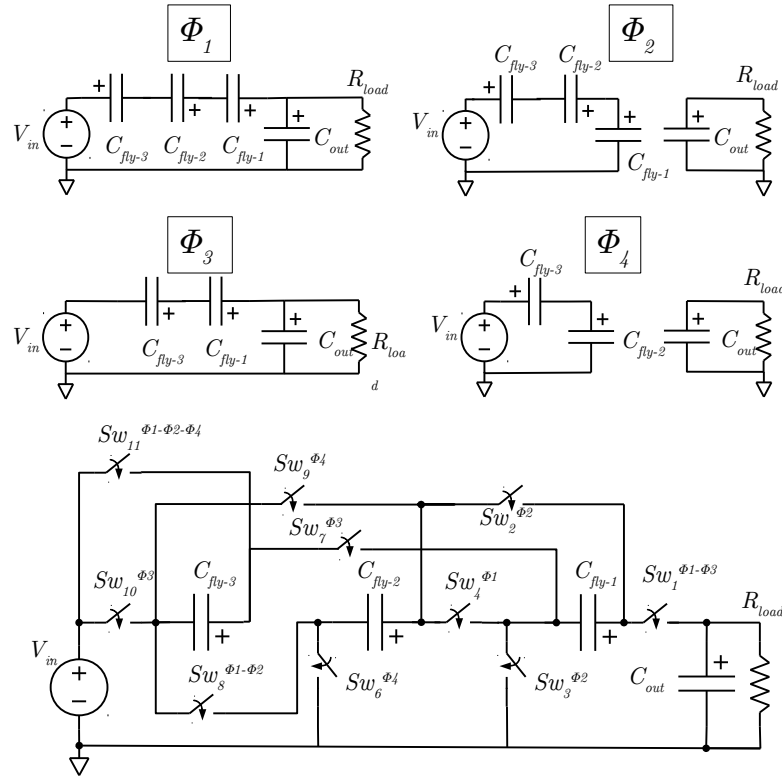


Figura 2.4: Fasi e circuito completo per il convertitore step-up 8/3.

Le fasi per il convertitore con rapporto di conversione pari a $8/3$ sono le medesime di quella rispettiva versione step-down solo che basta invertire il generatore di ingresso con il parallelo RC del condensatore e carico di uscita. Solo in questo convertitore però è possibile ridurre il numero delle fasi da 5 a 4 per il fatto che l'algoritmo produce, in alcuni casi, dei codici ridondanti. Nel caso di questo progetto si è deciso di togliere la fase numero 4 e sostituirla con quella che era la fase numero 5. Tutto questo è specificato nell'appendice A.

Come fatto in precedenza anche qui si son calcolati i vettori dei coefficienti di carica in modo tale da poter dimensionare gli switch, i condensatori flottanti e stabilire la frequenza di lavoro in modo tale che tutto il convertitore lavori nel regime operativo FSL.

$$\begin{aligned}
 & \begin{cases} q_{out}^{\Phi_1} + q_{out}^{\Phi_3} = q_{out} \\ q_{out}^{\Phi_1} = -q_{C1}^{\Phi_1} = -q_{C2}^{\Phi_1} = q_{C3}^{\Phi_1} \\ q_{C1}^{\Phi_2} = -q_{C2}^{\Phi_2} = q_{C3}^{\Phi_2} \\ q_{out}^{\Phi_3} = -q_{C1}^{\Phi_3} = -q_{C3}^{\Phi_3} \\ q_{C2}^{\Phi_4} = q_{C3}^{\Phi_4} \end{cases} \xrightarrow[\cdot \frac{1}{q_{out}}]{} \begin{cases} a_{out}^{\Phi_1} + a_{out}^{\Phi_3} = 1 \\ a_{out}^{\Phi_1} = -a_{C1}^{\Phi_1} = -a_{C2}^{\Phi_1} = a_{C3}^{\Phi_1} \\ a_{C1}^{\Phi_2} = -a_{C2}^{\Phi_2} = a_{C3}^{\Phi_2} \\ a_{out}^{\Phi_3} = -a_{C1}^{\Phi_3} = -a_{C3}^{\Phi_3} \\ a_{C2}^{\Phi_4} = a_{C3}^{\Phi_4} \end{cases} \\
 & \rightarrow \begin{cases} a_{C3}^{\Phi_1} - a_3^{\Phi_3} = 1 \\ -a_{C3}^{\Phi_1} + a_{C3}^{\Phi_2} + a_{C3}^{\Phi_3} = 0 \\ -a_{C3}^{\Phi_1} - a_{C3}^{\Phi_2} + a_{C3}^{\Phi_4} = 0 \\ a_{C3}^{\Phi_1} + a_{C3}^{\Phi_2} + a_{C3}^{\Phi_3} + a_{C3}^{\Phi_4} = 0 \end{cases} \rightarrow \begin{cases} a_{C3}^{\Phi_1} = 1 + a_3^{\Phi_3} = -\frac{1}{3} \\ a_{C2}^{\Phi_2} = 1 \\ a_{C3}^{\Phi_4} = 2 + a_{C3}^{\Phi_3} = -\frac{2}{3} \\ a_{C3}^{\Phi_3} = -\frac{4}{3} \end{cases} \quad (2.6)
 \end{aligned}$$

I coefficienti relativi ai condensatori C_2 e C_3 si ricavano immediatamente dal primo sistema.

Analizzando il circuito completo in figura 2.4 si ottengono i coefficienti relativi ai vari switch:

Φ	Sw_1	Sw_2	Sw_3	Sw_4	Sw_5	Sw_6	Sw_7	Sw_8	Sw_9	Sw_{10}	Sw_{11}
Φ_1	$\frac{1}{3}$	0	0	$\frac{1}{3}$	0	0	0	$\frac{1}{3}$	0	0	$\frac{1}{3}$
Φ_2	0	1	1	0	0	0	0	1	0	0	1
Φ_3	$\frac{4}{3}$	0	0	0	0	0	$\frac{4}{3}$	0	0	$\frac{4}{3}$	0
Φ_4	0	0	0	0	0	$\frac{2}{3}$	0	0	$\frac{2}{3}$	0	$\frac{2}{3}$

Tabella 2.6: Tabella riassuntiva dei coefficienti per gli switch.

Come già descritto in precedenza se si applica anche in questo la *KVL* per ogni circuito si ottiene che le tensioni ai capi dei condensatori dovrebbero essere:

$$\begin{cases} V_{out} = V_{in} - V_{C3} + V_{C2} + V_{C1} \\ V_{C1} = V_{in} - V_{C3} + V_{C2} \\ V_{out} = V_{in} + V_{C3} + V_{C1} \\ V_{C2} + V_{C3} = V_{in} \end{cases} \rightarrow \begin{cases} V_{out} = \frac{8}{3} \cdot V_{in} \\ V_{C1} = \frac{4}{3} \cdot V_{in} \\ V_{C2} = \frac{2}{3} \cdot V_{in} \\ V_{C3} = \frac{1}{3} \cdot V_{in} \end{cases} \quad (2.7)$$

2.1.4 Convertitore 5/2

I codici per il convertitore $\frac{2}{5}$ estrapolati sempre mediante il medesimo algoritmo sono:

A_0	A_1	A_2	A_3
0	0	1	0
0	1	-1	1
0	1	0	-1
1	-1	0	0

Tabella 2.7: Tabella riassuntiva dei codici EXB per il numero $2/5$.

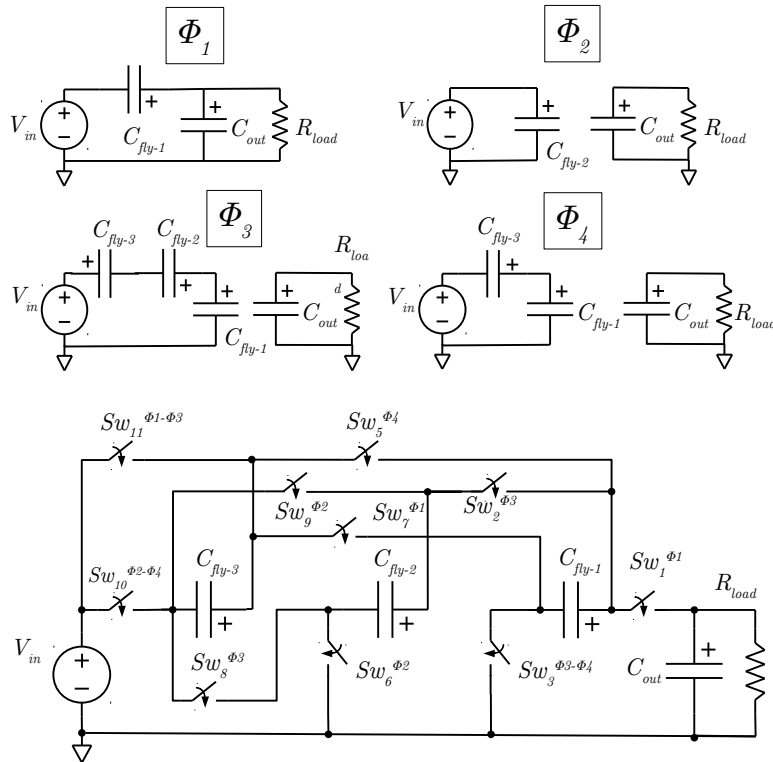


Figura 2.5: Fasi e circuito completo per il convertitore step-up 5/2.

Le fasi per il convertitore con rapporto di conversione pari a $5/2$ sono le medesime di quella rispettiva versione step-down solo che basta invertire il generatore di ingresso con il parallelo RC del condensatore e carico di uscita. Come fatto in precedenza anche qui si son calcolati i vettori dei coefficienti di carica in modo tale da poter dimensionare gli switch, i condensatori flottanti e stabilire la frequenza di lavoro in modo tale che tutto il convertitore lavori nel regime operativo FSL.

$$\begin{aligned}
 & \begin{cases} q_{out} = -q_{C1}^{\Phi_1} \\ q_{C1}^{\Phi_3} = -q_{C2}^{\Phi_3} = q_{C3}^{\Phi_3} \\ q_{C1}^{\Phi_4} = -q_{C3}^{\Phi_4} \end{cases} \xrightarrow[\cdot \frac{1}{q_{out}}]{} \begin{cases} 1 = -a_{C1}^{\Phi_1} \\ a_{C1}^{\Phi_3} = -a_{C2}^{\Phi_3} = a_{C3}^{\Phi_3} \\ a_{C1}^{\Phi_4} = -a_{C3}^{\Phi_4} \end{cases} \\
 & \rightarrow \begin{cases} a_{C1}^{\Phi_1} + a_{C1}^{\Phi_3} + a_{C1}^{\Phi_4} = 0 \\ a_{C2}^{\Phi_2} + a_{C2}^{\Phi_3} = 0 \\ a_{C3}^{\Phi_3} + a_{C3}^{\Phi_4} = 0 \end{cases} \rightarrow \begin{cases} a_{C1}^{\Phi_1} = -1 \\ a_{C1}^{\Phi_3} = \frac{1}{2} = a_{C1}^{\Phi_4} \\ a_{C2}^{\Phi_2} = -a_{C2}^{\Phi_3} = -a_{C1}^{\Phi_3} = -\frac{1}{2} \end{cases} \quad (2.8)
 \end{aligned}$$

Analizzando il circuito completo in figura 2.5 si ottengono i coefficienti relativi ai vari switch:

Φ	Sw_1	Sw_2	Sw_3	Sw_4	Sw_5	Sw_6	Sw_7	Sw_8	Sw_9	Sw_{10}	Sw_{11}
Φ_1	1	0	0	0	0	0	1	0	0	0	1
Φ_2	0	0	0	0	0	$\frac{1}{2}$	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0
Φ_3	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0	$\frac{1}{2}$	0	0	$\frac{1}{2}$
Φ_4	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	0	0	$\frac{1}{2}$	0

Tabella 2.8: Tabella riassuntiva dei coefficienti per gli switch.

La *KVL* applicata ad ogni circuito di ogni fase fornisce la tensione media che cade ai capi dei condensatori C_{fly} :

$$\begin{cases} V_{out} = V_{in} + V_{C1} \\ V_{C2} = V_{in} \\ V_{C1} = V_{in} + V_{C2} - V_{C3} \\ V_{C1} = V_{C3} + V_{in} \end{cases} \rightarrow \begin{cases} V_{out} = \frac{5}{2} \cdot V_{in} \\ V_{C1} = \frac{3}{2} \cdot V_{in} \\ V_{C2} = V_{in} \\ V_{C3} = \frac{1}{2} \cdot V_{in} \end{cases} \quad (2.9)$$

2.1.5 Convertitore 8/4

L'insieme di codici prodotti dall'algoritmo EXB per fattore di conversione step-down $\frac{4}{8}$ si ottiene:

A_0	A_1	A_2	A_3
1	-1	0	0
0	1	0	0

Tabella 2.9: Tabella riassuntiva dei codici EXB per il numero 4/8.

Come già detto anche in precedenza prendendo le fasi del convertitore step-down e scambiando la porta d'ingresso con quella di uscita si ottiene la versione step-up 8/4. Quindi le fasi finali sono:

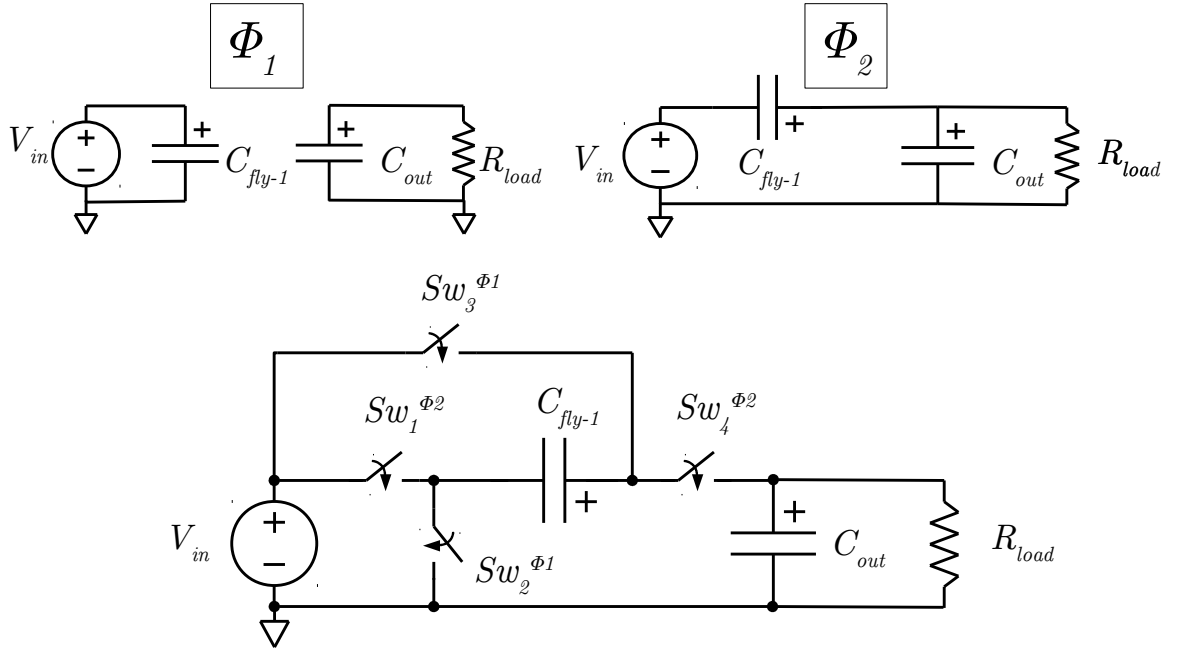


Figura 2.6: Fasi e circuito completo per il convertitore step-up 8/4.

Al fine di dimensionare in modo ottimale la W ed L degli switch, la capacità dei condensatori C_{fly} e la frequenza di lavoro affinché tutto sia in

FSL risulta necessario calcolare i vettori dei coefficienti di carica. Questi vettori vengono estrapolati andando ad analizzare i flussi di carica nelle varie fasi:

$$\begin{aligned} \begin{cases} q_{out} = -q_{C1}^{\Phi_2} \\ q_{C1}^{\Phi_1} = -q_{C1}^{\Phi_2} \end{cases} & \xrightarrow{\cdot \frac{1}{q_{out}}} \begin{cases} 1 = -a_{C1}^{\Phi_2} \\ a_{C1}^{\Phi_1} + a_{C1}^{\Phi_2} = 0 \end{cases} \\ \rightarrow & a_{C1}^{\Phi_1} = -a_{C1}^{\Phi_2} = 1 \end{aligned} \quad (2.10)$$

Partendo da questi coefficienti e andando ad analizzare il circuito completo in figura 2.6 si ottengono i coefficienti relativi ai vari switch:

Φ	Sw_1	Sw_2	Sw_3	Sw_4
Φ_1	1	0	0	1
Φ_2	0	1	1	0

Tabella 2.10: Tabella riassuntiva dei coefficienti per gli switch.

Applicando il medesimo procedimento della legge di Kichhoff alla maglia troviamo che le tensioni medie ai capi dei condensatori sono:

$$\begin{cases} V_{C1} = V_{in} \\ V_{out} = V_{in} + V_{C1} = 2 \cdot V_{in} \end{cases} \quad (2.11)$$

2.1.6 SCC completo e dimensionamento

Una volta analizzati tutti e cinque i convertitori sono stati uniti gli schematici per formare un unico circuito. La figura 2.7 rappresenta il circuito con switch ideale del convertitore finale. Spetterà poi alla logica di controllo selezionare la giusta sequenza di interruttori.

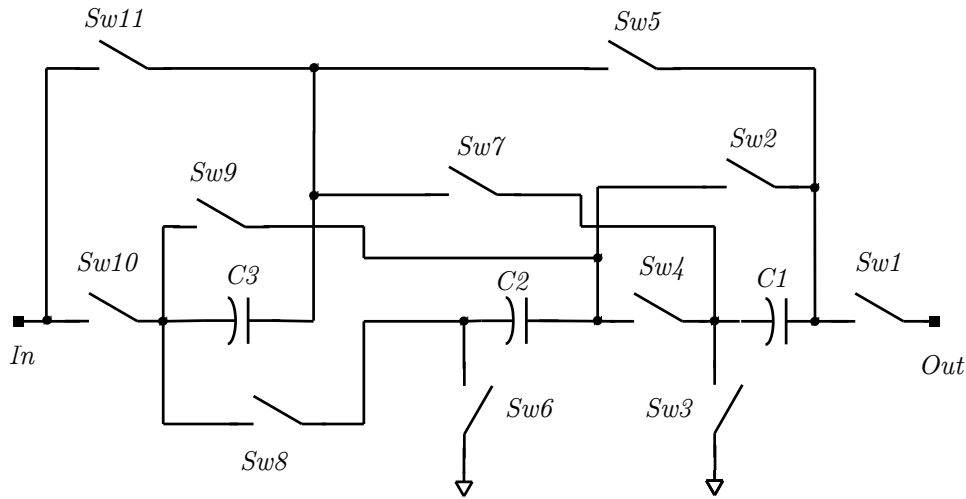


Figura 2.7: SCC completo con switch ideali.

Sapendo che gli switch ideali saranno sostituiti da transistor MOSFET e sapendo che la resistenza di tali dispositivi è influenzata anche dalla tensione che cade ai capi dei terminali di source e bulk, allora è stata fatta una simulazione del circuito (con switch ideali) per vedere la caduta di tensione ai capi di ogni switch per ogni convertitore.

Φ	Sw_1	Sw_3	Sw_5	Sw_7	Sw_{11}
—	$V_{out} - C_{1+}$	$C_{1-} - GND$	$C_{1-} - C_{2+}$	$C_{1+} - C_{3+}$	$C_{3+} - V_{in}$
Φ_1	1.2 – 1.2	0.6 – 0	1.2 – 0.6	0.6 – 0.6	0.6 – 0.6
Φ_2	1.2 – 0.6	0 – 0	0.6 – 0.6	0 – 0.6	0.6 – 0.6

Tabella 2.11: Tabella riassuntiva delle tensioni per il convertitore con rapporto di conversione 8/4 con $V_{in} = 0.6V$.

Φ	Sw_1	Sw_2	Sw_3	Sw_4	Sw_6	Sw_8	Sw_9	Sw_{10}
—	$V_{out} - C_{1+}$	$C_{1+} - C_{2+}$	$C_{1-} - GND$	$C_{1-} - C_{2+}$	$C_{2-} - GND$	$C_{2-} - C_{3-}$	$C_{2+} - C_{3-}$	$C_{3-} - V_{in}$
Φ_1	1.2 – 1.2	1.2 – 0.6	0.6 – 0	0.6 – 0.6	0.3 – 0	0.3 – 0.3	0.6 – 0.3	0.3 – 0.3
Φ_2	1.2 – 0.6	0.6 – 0.6	0 – 0	0 – 0.6	0.3 – 0	0.3 – 0.3	0.6 – 0.3	0.3 – 0.3
Φ_3	1.2 – 0.6	0.6 – 0.3	0 – 0	0 – 0.6	0 – 0	0 – 0.3	0.3 – 0.3	0.3 – 0.3

Tabella 2.12: Tabella riassuntiva delle tensioni per il convertitore con rapporto di conversione 8/2 con $V_{in} = 0.3V$.

Φ	Sw_1	Sw_2	Sw_3	Sw_6	Sw_7	Sw_8	Sw_9	Sw_{10}	Sw_{11}
—	$V_{out} - C_{1+}$	$C_{1+} - C_{2+}$	$C_{1-} - GND$	$C_{2-} - GND$	$C_{1-} - C_{3+}$	$C_{2-} - C_{3-}$	$C_{2+} - C_{3-}$	$C_{3-} - V_{in}$	$C_{3+} - V_{in}$
Φ_1	1.2 – 0.8	0.8 – 0.4	0 – 0	0 – 0	0 – 0.42	0 – 0.4	0.4 – 0.4	0.4 – 0.4	0.42 – 0.4
Φ_2	1.2 – 1.2	1.2 – 0.4	0.4 – 0	0.4 – 0	0.4 – 0.4	0 – 0.37	0.4 – 0.37	0.37 – 0.4	0.4 – 0.4
Φ_3	1.2 – 0.8	0.8 – 0.8	0 – 0	0 – 0	0 – 0.42	0.4 – 0.4	0.8 – 0.4	0.4 – 0.4	0.42 – 0.4

Tabella 2.13: Tabella riassuntiva delle tensioni per il convertitore con rapporto di conversione 3/1 con $V_{in} = 0.4V$.

Φ	Sw_1	Sw_2	Sw_3	Sw_4	Sw_5	Sw_6	Sw_7	Sw_8	Sw_9	Sw_{10}	Sw_{11}
—	$V_{out} - C_{1+}$	$C_{1+} - C_{2+}$	$C_{1-} - GND$	$C_{1-} - C_{2+}$	$C_{1+} - C_{3+}$	$C_{2-} - GND$	$C_{1-} - GND$	$C_{2-} - C_{3-}$	$C_{2+} - C_{3-}$	$C_{3-} - V_{in}$	$C_{3+} - V_{in}$
Φ_1	1.2 – 1.2	1.2 – 0.6	0.6 – 0	0.6 – 0.6	1.2 – 0.45	0.3 – 0	0.6 – 0.45	0.3 – 0.3	0.6 – 0.3	0.3 – 0.45	0.45 – 0.45
Φ_2	1.2 – 0.6	0.6 – 0.6	0 – 0	0 – 0.6	0.6 – 0.45	0.3 – 0	0 – 0.45	0.3 – 0.3	0.6 – 0.3	0.3 – 0.45	0.45 – 0.45
Φ_3	1.2 – 1.2	1.2 – 0.5	0.6 – 0	0.6 – 0.5	1.2 – 0.6	0.2 – 0	0.6 – 0.6	0.2 – 0.45	0.45 – 0.45	0.45 – 0.45	0.6 – 0.45
Φ_4	1.2 – 0.6	0.6 – 0.3	0 – 0	0 – 0.3	0.6 – 0.45	0 – 0	0 – 0.45	0 – 0.3	0.3 – 0.3	0.3 – 0.45	0.45 – 0.45

Tabella 2.14: Tabella riassuntiva delle tensioni per il convertitore con rapporto di conversione 8/3 con $V_{in} = 0.45V$.

Φ	Sw_1	Sw_2	Sw_3	Sw_5	Sw_6	Sw_7	Sw_8	Sw_9	Sw_{10}	Sw_{11}
—	$V_{out} - C_{1+}$	$C_{1+} - C_{2+}$	$C_{1-} - GND$	$C_{1+} - C_{3+}$	$C_{2-} - GND$	$C_{1-} - GND$	$C_{2-} - C_{3-}$	$C_{2+} - C_{3-}$	$C_{3-} - V_{in}$	$C_{3+} - V_{in}$
Φ_1	1.2 – 1.2	1.2 – 0.48	0.48 – 0	1.2 – 0.72	0 – 0	0.48 – 0.72	0 – 0.48	0.48 – 0.48	0.48 – 0.48	0.72 – 0.48
Φ_2	1.2 – 0.72	0.72 – 0.72	0 – 0	0.72 – 0.48	0.24 – 0	0 – 0.48	0.24 – 0.24	0.72 – 0.24	0.24 – 0.48	0.48 – 0.48
Φ_3	1.2 – 0.72	0.72 – 0.5	0 – 0	0.72 – 0.72	0 – 0	0 – 0.72	0 – 0.48	0.5 – 0.48	0.48 – 0.48	0.72 – 0.48
Φ_4	1.2 – 1.2	1.2 – 0.5	0.48 – 0	1.2 – 0.48	0 – 0	0.48 – 0.48	0 – 0.24	0.5 – 0.24	0.24 – 0.48	0.48 – 0.48

Tabella 2.15: Tabella riassuntiva delle tensioni per il convertitore con rapporto di conversione 5/2 con $V_{in} = 0.48V$.

Al fine di sapere come varia la resistenza serie del transistor in funzione della lunghezza di canale e soprattutto in funzione della tensione di source è stata fatta una simulazione parametrica del punto operativo dei seguenti circuiti e poi si è calcolata la resistenza equivalente facendo la divisione tra la tensione V_{DS} e la corrente di *drain* nel caso del transistor nMOS e quella di *source* nel caso del transistor pMOS.

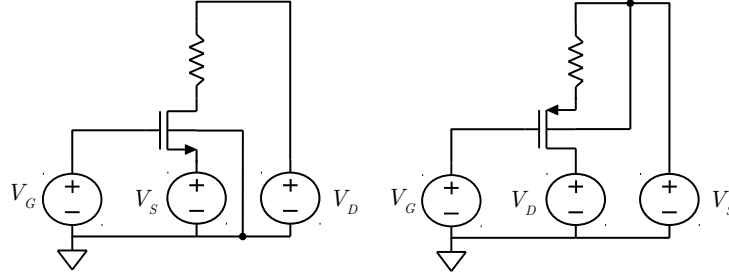


Figura 2.8: Schemi dei circuiti usati per estrarre le informazioni sulla dipendenza della resistenza dalla tensione di drain nel pMOS e source nel nMOS.

Con gli schemi di figura 2.8 si sono ottenute le seguenti curve.

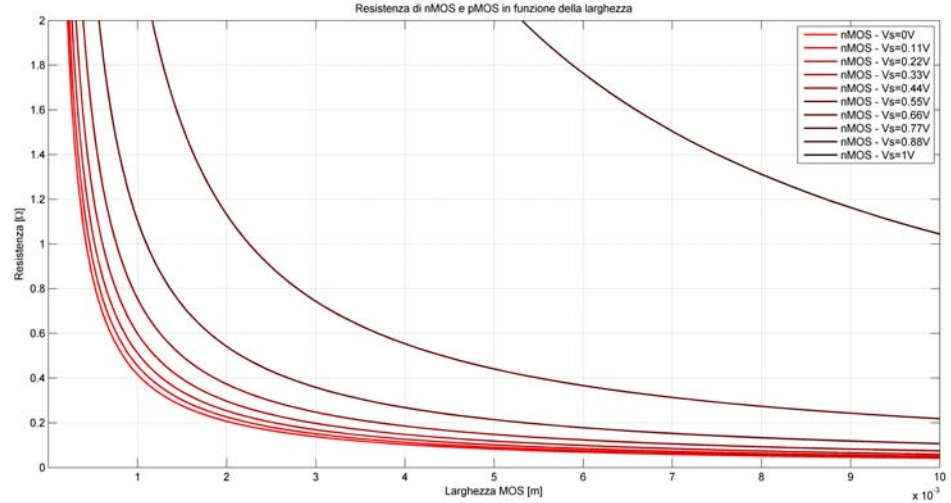


Figura 2.9: Resistenza del transistor nMOS in funzione della W e della tensione di source V_S .

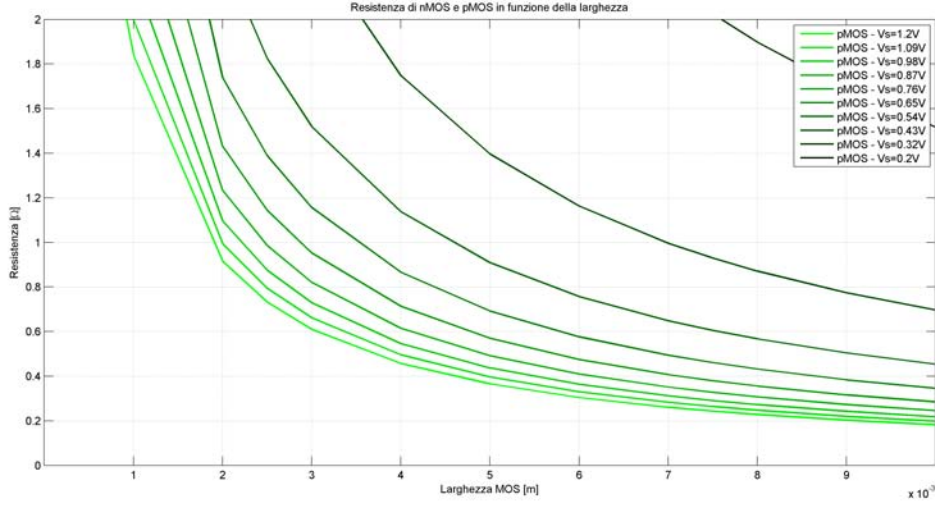


Figura 2.10: Resistenza del transistor pMOS in funzione della W e della tensione di source V_S .

Il calcolo della resistenza serie lo si fa considerando il transistor in zona lineare per il fatto che tale dispositivo, dovendo emulare l'interruttore ideale, la caduta di tensione tra il *source* ed il *drain* deve tendere a zero. La corrente di drain del transistor MOS in zona lineare si esprime come:

$$I_D = \mu C_{ox} \frac{W}{L} \left[(V_{GS} - V_{th}) V_{DS} - \frac{V_{DS}^2}{2} \right]$$

Derivando l'equazione sovrastante per V_{DS} si ottiene la conduttanza del transistor (il reciproco della resistenza):

$$G = \frac{\partial I_D}{\partial V_{DS}} = \mu C_{ox} \frac{W}{L} (V_{GS} - V_{th} - V_{DS}) \quad \rightarrow \quad R = \frac{1}{G} = k \cdot \frac{L}{W} \quad (2.12)$$

È immediato osservare che la resistenza del transistor MOS è inversamente proporzionale alla larghezza di canale W . Tramite lo script MatLab B.1 è stato possibile eseguire un fitting delle curve di simulazione estrapolando in questo modo il parametro k che rappresenta tutti i parametri tecnologici del transistor. Quindi dato un valore di resistenza, conoscendo k e impostando la minima lunghezza di canale $L_{min} = 120nm$ si ricava la larghezza di canale W .

La procedura di dimensionamento dei transistor segue quella descritta in *A Design Methodology for Switched-Capacitor DC-DC Converters*. Sapendo che l'efficienza del modello linearizzato del convertitore è:

$$\eta = \frac{R_{load}}{R_{load} + R_{FSL}} \rightarrow R_{FSL} = \frac{R_{load} \cdot (1 - \eta)}{\eta} \quad (2.13)$$

Sapendo che la conduttanza totale di tutti gli switch è:

$$G_{tot} = \left(\sum_i |a_{r,i}| \right)^2 \cdot \frac{2}{R_{FSL}} \quad (2.14)$$

dove $\sum_i |a_{r,i}|$ rappresenta la somma di tutti gli elementi dei vettori dei coefficienti di carica precedentemente calcolati e infine sapendo che la conduttanza del singolo transistor si può calcolare con vettori dei coefficienti di carica, allora si può calcolare la resistenza di ciascun switch:

$$G_{Sw_i} = \left(\sum_k^{#phases} a_{Sw_i}^{\Phi_k} \right) \cdot \frac{G_{tot}}{\sum_i |a_{r,i}|} \rightarrow R_{Sw_i} = \frac{1}{G_{Sw_i}} \quad (2.15)$$

Una volta ricavata anche la resistenza che dovrebbe avere il transistor, grazie alla 2.12 si ottiene la dimensione del MOSFET (sia nMOS che pMOS in base anche alla tensione che si ritrova ai suoi capi). Questo procedimento è stato ripetuto con tutti cinque i convertitori.

Dall'analisi delle tabelle 2.11 2.12 2.13 2.14 2.15 e dalle dimensioni dei transistor sia nMOS che pMOS si è deciso di utilizzare tutti transistor di tipo nMOS ad eccezione di Sw_1 che è stato scelto un pMOS.

Come si può osservare dal procedimento di dimensionamento tutto parte da un valore di efficienza ed infatti sono stati fatti diversi dimensionamenti in base a valori di efficienza che andavano dal 95% al 99%. La spiegazione di questa scelta progettuale risiede nel fatto che con valori di efficienza sempre maggiori anche le dimensioni degli switch aumentano. Avere nel chip switch troppo grandi (anche di qualche millimetro) significa avere capacità molto grandi, prima tra tutte la C_{GS} . Per pilotare carichi capacitivi molto grandi nella tecnologia *CMOS* si usano dei buffer costituiti da catene di invertitori CMOS scalati di un certo fattore l'uno rispetto all'altro. Tuttavia il consumo di potenza di questi buffer cresce linearmente con la frequenza e con la capacità di carico che devono pilotare. Quindi maggiore è la capacità C_{GS} del transistor maggiore sarà il consumo dei buffer. Perciò avere dispositivi molto grandi per raggiungere alti valori di efficienza potrebbe causare un crollo delle prestazioni del sistema totale a causa dell'elevato dispendio di energia causato dai relativi buffer. Risulta quindi più conveniente a livello di sistema

avere un scc meno efficiente ma con un'efficienza globale (scc ed il relativo buffer) migliore. Quindi il dimensionamento finale è stato trovato andando ad analizzare quale sistema (scc e relativo buffer) fosse più efficiente in base alla seguente relazione:

$$\eta_{scc+buffer} = \frac{P_{out}^{scc}}{P_{in}^{scc} + P_{alim}^{buffer}} = \frac{\int_0^T (V_{out} \cdot I_{load}) dt}{\int_0^T (V_{in} \cdot I_{in}) dt + \int_0^T (V_{out} \cdot I_{buffer}) dt} \quad (2.16)$$

La figura 2.11 mostra i valori di efficienza raggiunti in base alle diverse tensioni in ingresso:

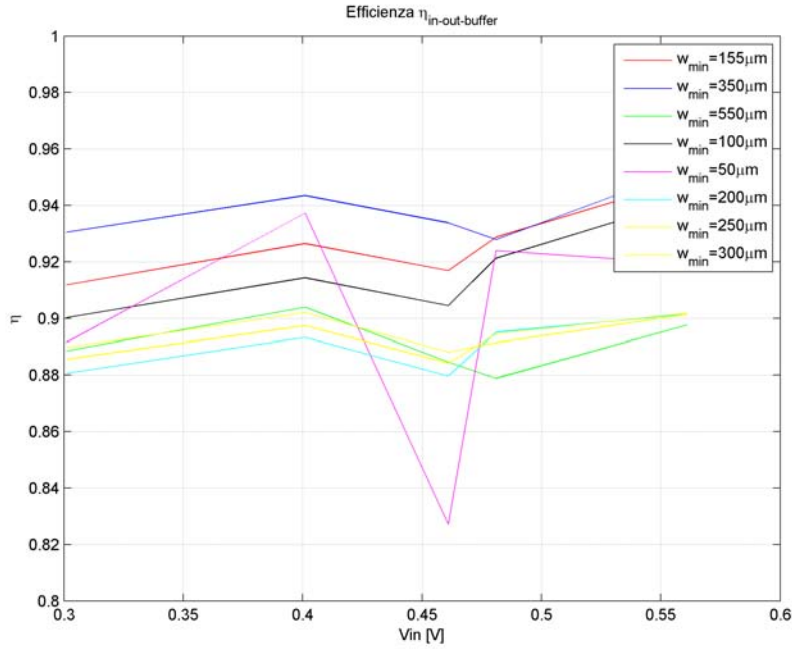


Figura 2.11: Simulazioni di efficienza in base a diversi valori di tensione in ingresso. La leggenda indica la dimensione del transistor con la minor larghezza di canale.

Dal grafico sovrastante si deduce che il dimensionamento con la miglior efficienza raggiunta è quello con $W_{min} = 350 \mu m$ e quindi:

Switch	Sw_1	Sw_2	Sw_3	Sw_4	Sw_5	Sw_6	Sw_7	Sw_8	Sw_9	Sw_{10}	Sw_{11}
Type	pMOS	nMOS	nMOS	nMOS	nMOS	nMOS	nMOS	nMOS	nMOS	nMOS	nMOS
W [μm]	2084	1230	350	1188	434	700	967	905	905	1812	1045

Il modello del transistor pMOS usato per le simulazioni è *P_12_HSL130E* con il terminale di *bulk* cortocircuitato al *source*.

I transistor nMOS utilizzati per le simulazioni sono quelli costruiti in *triple-well* (*N_BPW_12_HSL130E*). La scelta di utilizzare transistor costruiti su una triple-well è nata per due motivi: il primo è legato all'effetto body. Infatti sapendo che il terminale di bulk deve essere connesso al potenziale minore del circuito (in questo caso la massa) e non avendo la possibilità di cortocircuitare il source con il bulk (altrimenti non viene più rispettato lo schematico) allora si crea inevitabilmente una tensione V_{SB} che causa l'innalzamento della V_{th} per effetto body. Avendo inoltre la tensione di gate costante e pari a V_{out} (a livello logico alto), allora la $V_{GS} - V_{th}$ risulta minore e di conseguenza il transistor risulta più resistivo. Invece nei dispositivi costruiti nella triple-well il bulk può essere cortocircuitato con il source evitando in questo modo l'effetto body. Il secondo motivo deriva dal fatto che cortocircuitare tutti i bulk (e quindi source) significa rendere tutti i source equipotenziali, cosa che non va bene per il funzionamento del circuito. L'immagine 2.12 dimostra come sono costruiti i dispositivi triple-well.

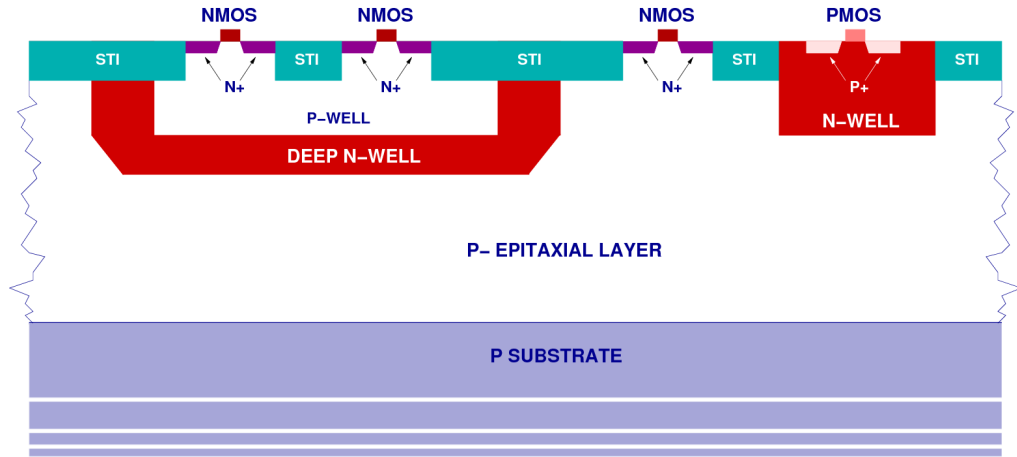


Figura 2.12: Dispositivi CMOS triple-well e normali a confronto.

Una piccola eccezione è stata fatta allo switch Sw_4 perchè in simulazione si è notato che durante il funzionamento dei convertitori con rapporto di conversione $5/2$ e $3/1$ il diodo *drain-bulk* di questo switch andava in conduzione e quindi si notava una grossa diminuzione di efficienza. La soluzione è stata quella di mettere il bulk a massa ed in questo modo il diodo non può più entrare in conduzione. Il lato negativo di questa soluzione è stato l'inevitabile effetto body.

La figura 2.13 rappresenta il convertitore a capacità commutate con i MOSFET al posto degli switch ideali. I transistor verdi sono gli nMOS mentre quello rosso è il pMOS.

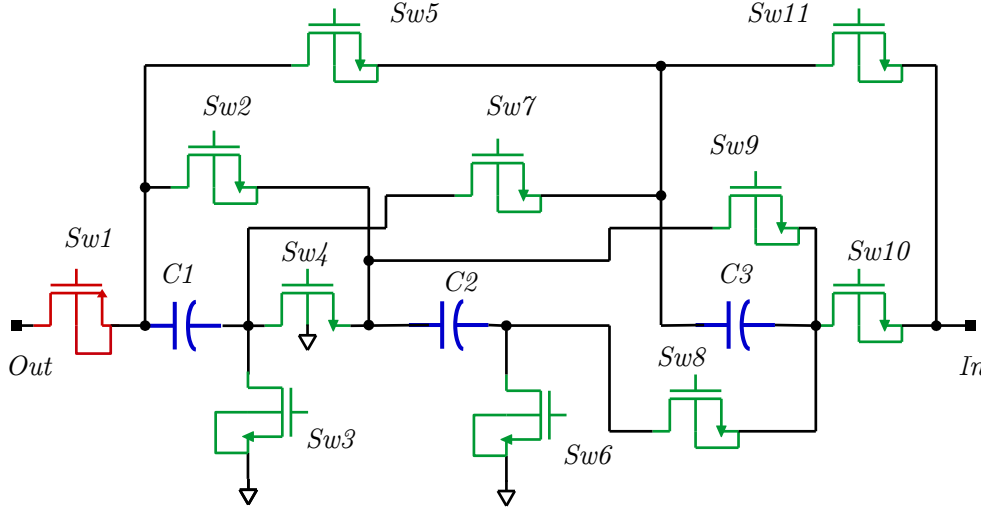


Figura 2.13: SCC completo con switch reali.

2.1.7 Modello medio con perdite

Nota l'intera topologia e note le dimensioni dei transistor (quindi la loro resistenza nei vari casi) è possibile costruire un modello medio dei convertitori in modo tale da verificare se il calcolo della R_{FSL} è corretto.

Di seguito verrà illustrato il procedimento per il calcolo del modello medio con perdite del convertitore 8/4 ma il procedimento lo si può estendere a tutti gli altri per ricavare i relativi modelli.

Modello del convertitore 8/4

Avendo imposto che il regime operativo è il FSL allora le curve esponenziali di carica e scarica dei condensatori nelle varie fasi possono essere approssimate con delle rette pertanto la tensione di uscita è approssimabile come:

$$V_{out}(T) = V_{out}(D_{\Phi_1}T) \exp\left(-\frac{D_{\Phi_2}T}{R_{load}C_{out}}\right) \simeq V_{out}(D_{\Phi_1}T) \left(1 - \frac{D_{\Phi_2}T}{R_{load}C_{out}}\right) \quad (2.17)$$

dove $D_{\Phi_i}T$ è la durata della fase i -esima. La variazione della tensione di uscita ΔV_{out} è esprimibile come:

$$\Delta V_{out} = V_{out}(T) - V_{out}(D_{\Phi_1}T) = V_{out}(D_{\Phi_1}T) \cdot \left(\frac{D_{\Phi_2}T}{R_{load}C_{out}} \right) \quad (2.18)$$

ma la stessa quantità può essere espressa come:

$$\Delta V_{out} = \frac{I_{C_{out}}D_{\Phi_2}T}{C_{out}} = \frac{I_{R_{load}}D_{\Phi_2}T}{C_{out}} = \frac{V_{out}D_{\Phi_2}T}{R_{load}C_{out}} \quad (2.19)$$

dove la relazione $I_{C_{out}} = I_{R_{load}}$ è vera nella fase due poichè carico e condensatore di uscita sono staccati da scc e sono in serie.

Andando invece ad osservare gli schemi di figura 2.6 ed inserendo le resistenze serie degli switch si possono facilmente ricavare le seguenti equazioni:

$$\begin{cases} V_{out} = V_{in} + V_{C1} - I_{\Phi_1}R_{sw-\Phi_1} \\ V_{C1} = V_{in} - I_{\Phi_2}R_{sw-\Phi_2} \end{cases} \quad (2.20)$$

dove $R_{sw-\Phi_1}$ è la somma delle resistenze serie degli switch che vengono chiusi nella fase uno quindi: $R_{sw-\Phi_1} = R_{Sw_1} + R_{Sw_7} + R_{Sw_{11}}$, analogamente $R_{sw-\Phi_2} = R_{Sw_3} + R_{Sw_5} + R_{Sw_{11}}$.

Imponendo che tutte le fasi abbiano la stessa durata: $D_{\Phi_1} = D_{\Phi_2}$, sapendo che vale il principio di conservazione della carica: $Q_{iniziale} = Q_{finale} \rightarrow I_{\Phi_1}D_{\Phi_1}T = I_{\Phi_2}D_{\Phi_2}T \rightarrow I_{\Phi_1} = I_{\Phi_2} = I$ e risolvendo il sistema sopra descritto si ottiene la seguente equazione:

$$V_{out} = 2V_{in} - I \cdot (R_{sw-\Phi_1} + R_{sw-\Phi_2}) \quad (2.21)$$

La 2.21 rappresenta già il modello medio del convertitore, ma non è ancora ben definito il valore della corrente I . Pensando al modello descritto in precedenza, la corrente I non è altro che la corrente che scorre nel carico e quella che serve per ricaricare il condensatore posto nella porta di uscita del convertitore:

$$I = I_{R_{load}} + I_{C_{out}} = \frac{V_{out}}{R_{load}} + \frac{C_{out}\Delta V_{out}}{D_{\Phi_1}T} = \frac{V_{out}}{R_{load}} \cdot \left(\frac{1}{D_{\Phi_1}} \right) \quad (2.22)$$

inserendo quest'ultima equazione della 2.21 e sapendo che $D_{\Phi_1} = D_{\Phi_2} = 0.5$ (poichè ci sono solo due fasi che si alternano):

$$V_{out} = 2V_{in} - \frac{V_{out}}{R_{load}} \cdot 2 \cdot (R_{sw-\Phi_1} + R_{sw-\Phi_2}) \quad (2.23)$$

L'equazione 2.23 è il modello medio finale e la si risolve mediante iterazione. Si può facilmente notare che la resistenza di uscita per questo convertitore è:

$$R_{out} = R_{FSL} = 2 \cdot (R_{sw-\Phi_1} + R_{sw-\Phi_2}) \quad (2.24)$$

Inoltre qui viene evidenziato il fatto che il rapporto di conversione non potrà mai raggiungere il valore teorico di $M_{8/4} = 8/4 = 2$ se non per resistenza degli switch prossima allo zero.

Le resistenze degli switch sono:

Switch	Sw_1	Sw_3	Sw_5	Sw_7	Sw_{11}
R [Ω]	0.897	1.21	2.50	1.21	1.04

Modello del convertitore 8/2

Seguendo un procedimento analogo ma impostando il sistema di partenza come:

$$\begin{cases} V_{out} = V_{in} + V_{C1} + V_{C2} - I_{\Phi_1} R_{sw-\Phi_1} \\ V_{C1} = V_{in} + V_{C2} - I_{\Phi_2} R_{sw-\Phi_2} \\ V_{C2} = V_{in} - I_{\Phi_3} R_{sw-\Phi_3} \end{cases} \quad (2.25)$$

dove $R_{sw-\Phi_1} = R_{Sw_1} + R_{Sw_4} + R_{Sw_8} + R_{Sw_{10}}$, $R_{sw-\Phi_2} = R_{Sw_2} + R_{Sw_3} + R_{Sw_8} + R_{Sw_{10}}$ e $R_{sw-\Phi_3} = R_{Sw_6} + R_{Sw_9} + R_{Sw_{10}}$. Imponendo che tutte le fasi abbiano la stessa durata: $D_{\Phi_1} = D_{\Phi_2} = D_{\Phi_3} = 0.33$ si ricava che:

$$V_{out} = 4V_{in} - \frac{V_{out}}{R_{load}} \cdot 3 \cdot (R_{sw-\Phi_1} + R_{sw-\Phi_2} + 4R_{sw-\Phi_3}) \quad (2.26)$$

con una resistenza di uscita:

$$R_{out} = R_{FSL} = 3 \cdot (R_{sw-\Phi_1} + R_{sw-\Phi_2} + 4R_{sw-\Phi_3}) \quad (2.27)$$

Le resistenze degli switch sono:

Switch	Sw_1	Sw_2	Sw_3	Sw_4	Sw_6	Sw_8	Sw_9	Sw_{10}
R [Ω]	0.897	0.881	1.21	1.21	0.605	0.605	0.605	0.302

Modello del convertitore 3/1

Seguendo lo stesso procedimento, ma applicandolo a questo nuovo convertitore si ha che il sistema diventa:

$$\begin{cases} V_{out} = V_{in} + V_{C1} - I_{\Phi_1} R_{sw-\Phi_1} \\ V_{C1} = V_{in} + V_{C2} - I_{\Phi_2} R_{sw-\Phi_2} \\ V_{C2} = V_{in} - I_{\Phi_3} R_{sw-\Phi_3} \end{cases} \quad (2.28)$$

dove $R_{sw-\Phi_1} = R_{Sw_1} + R_{Sw_7} + R_{Sw_{11}}$, $R_{sw-\Phi_2} = R_{Sw_2} + R_{Sw_3} + R_{Sw_8} + R_{Sw_{10}}$ e $R_{sw-\Phi_3} = R_{Sw_6} + R_{Sw_9} + R_{Sw_{10}}$. Imponendo che tutte le fasi abbiano la stessa durata: $D_{\Phi_1} = D_{\Phi_2} = D_{\Phi_3} = 0.33$ si ricava che:

$$V_{out} = 3V_{in} - \frac{V_{out}}{R_{load}} \cdot 3 \cdot (R_{sw-\Phi_1} + R_{sw-\Phi_2} + R_{sw-\Phi_3}) \quad (2.29)$$

con una resistenza di uscita:

$$R_{out} = R_{FSL} = 3 \cdot (R_{sw-\Phi_1} + R_{sw-\Phi_2} + R_{sw-\Phi_3}) \quad (2.30)$$

Le resistenze degli switch sono:

Switch	Sw_1	Sw_2	Sw_3	Sw_6	Sw_7	Sw_8	Sw_9	Sw_{10}	Sw_{11}
R [Ω]	0.897	4.26	1.21	0.605	0.656	0.7	0.7	0.35	0.607

Modello del convertitore 5/2

Impostando lo stesso identico procedimento si ottiene:

$$\begin{cases} V_{out} = V_{in} + V_{C1} - I_{\Phi_1} R_{sw-\Phi_1} \\ V_{C1} = V_{in} + V_{C3} - I_{\Phi_4} R_{sw-\Phi_4} \\ V_{C1} = V_{in} - V_{C3} + V_{C2} - I_{\Phi_3} R_{sw-\Phi_3} \\ V_{C2} = V_{in} - I_{\Phi_2} R_{sw-\Phi_2} \end{cases} \quad (2.31)$$

dove $R_{sw-\Phi_1} = R_{Sw_1} + R_{Sw_7} + R_{Sw_{11}}$, $R_{sw-\Phi_2} = R_{Sw_6} + R_{Sw_9} + R_{Sw_{10}}$, $R_{sw-\Phi_3} = R_{Sw_2} + R_{Sw_3} + R_{Sw_8} + R_{Sw_{11}}$ e $R_{sw-\Phi_4} = R_{Sw_3} + R_{Sw_5} + R_{Sw_{10}}$. Imponendo che tutte le fasi abbiano la stessa durata: $D_{\Phi_1} = D_{\Phi_2} = D_{\Phi_3} = D_{\Phi_4} = 0.25$ si ricava che:

$$V_{out} = \frac{5}{2}V_{in} - \frac{V_{out}}{R_{load}} \cdot (4R_{sw-\Phi_1} + R_{sw-\Phi_2} + R_{sw-\Phi_3} + R_{sw-\Phi_4}) \quad (2.32)$$

con una resistenza di uscita:

$$R_{out} = R_{FSL} = 4R_{sw-\Phi_1} + R_{sw-\Phi_2} + R_{sw-\Phi_3} + R_{sw-\Phi_4} \quad (2.33)$$

Le resistenze degli switch sono:

Switch	Sw_1	Sw_2	Sw_3	Sw_5	Sw_6	Sw_7	Sw_8	Sw_9	Sw_{10}	Sw_{11}
R [Ω]	0.897	1.495	1.21	4.237	0.605	0.808	0.605	0.862	0.431	0.748

Modello del convertitore 8/3

Partendo dal seguente sistema e seguendo il procedimento illustrato in precedenza si ottiene:

$$\begin{cases} V_{out} = V_{in} + V_{C1} + V_{C2} - V_{C3} - I_{\Phi_1} R_{sw-\Phi_1} \\ V_{C1} = V_{in} + V_{C2} - V_{C3} - I_{\Phi_2} R_{sw-\Phi_2} \\ V_{out} = V_{in} + V_{C1} + V_{C3} - I_{\Phi_3} R_{sw-\Phi_3} \\ V_{C2} = V_{in} - V_{C3} - I_{\Phi_4} R_{sw-\Phi_4} \end{cases} \quad (2.34)$$

dove $R_{sw-\Phi_1} = R_{Sw_1} + R_{Sw_4} + R_{Sw_8} + R_{Sw_{11}}$, $R_{sw-\Phi_2} = R_{Sw_2} + R_{Sw_3} + R_{Sw_8} + R_{Sw_{11}}$, $R_{sw-\Phi_3} = R_{Sw_1} + R_{Sw_7} + R_{Sw_{10}}$ e $R_{sw-\Phi_4} = R_{Sw_6} + R_{Sw_9} + R_{Sw_{11}}$. Imponendo che tutte le fasi abbiano la stessa durata: $D_{\Phi_1} = D_{\Phi_2} = D_{\Phi_3} = D_{\Phi_4} = 0.25$ si ricava che:

$$V_{out} = \frac{8}{3}V_{in} - \frac{V_{out}}{R_{load}} \cdot \frac{2}{9} \cdot (R_{sw-\Phi_1} + 7R_{sw-\Phi_2} + 16R_{sw-\Phi_3} + 6R_{sw-\Phi_4}) \quad (2.35)$$

con una resistenza di uscita:

$$R_{out} = R_{FSL} = \frac{2}{9} \cdot (R_{sw-\Phi_1} + 7R_{sw-\Phi_2} + 16R_{sw-\Phi_3} + 6R_{sw-\Phi_4}) \quad (2.36)$$

Le resistenze degli switch sono:

Switch	Sw_1	Sw_2	Sw_3	Sw_4	Sw_5	Sw_6	Sw_7	Sw_8	Sw_9	Sw_{10}	Sw_{11}
R [Ω]	0.897	1.49	1.21	1.21	2.497	0.605	1.12	0.605	0.862	0.35	0.748

2.1.8 Dimensionamento dei C_{fly}

Affinchè il circuito lavori in una condizione operativa di FSL bisogna che la costante di tempo data dai condensatori C_{fly} e la resistenza degli switch sia molto maggiore rispetto alla durata di ciascuna fase. Tuttavia al fine di

evitare consumi di potenza eccessivi da parte della circuiteria della logica si è cercato di usare una frequenza di lavoro non troppo alta in modo che la potenza dissipata non sia troppo elevata per il sistema. L'idea è quella di trovare la minor frequenza affinché il circuito lavori in FSL.

Considerando tutte le fasi si nota che il caso peggiore per quanto riguarda la costante di tempo è la fase 2 del convertitore 8/3 poichè mette in serie 3 condensatori C_{fly} . Imponendo la durata di ciascuna fase pari a $T = 500ns$ e calcolando la $R_{sw-\Phi_2} = 4.053\Omega$:

$$T \gg R_{sw-\Phi_2} \cdot \frac{C_{fly}}{3} \quad \rightarrow \quad C_{fly} = \frac{3T}{R_{sw-\Phi_2}} \cdot 10 = 3.7\mu F$$

Il valore dei condensatori ottenuto è accettabile se considerato come componente esterno al chip (poichè il valore di capacità risulta essere troppo elevato per essere integrabile). Tuttavia pensando ai condensatori della serie E24 (con tolleranza del 5%) il valore di capacità più vicino sarebbe il $3.9\mu F$. Invece il condensatore C_{out} può essere messo anche di valore maggiore rispetto ai condensatori C_{fly} , tuttavia maggiore è la sua capacità maggiore sarà il tempo necessario per portare a regime V_{out} .

2.1.9 Buffer SCC

Le grandi dimensioni dei transistor calcolati nel punto precedente hanno come diretta conseguenza la generazione di notevoli capacità. Il segnale responsabile dell'accensione e spegnimento di questi transistor deve essere in grado di pilotare tali capacità. Nella tecnologia CMOS per eseguire questo compito vengono usati dei buffer costituiti da una cascata di invertitori opportunamente scalati di un certo fattore. La figura 2.14 illustra lo schema di principio di un buffer CMOS.

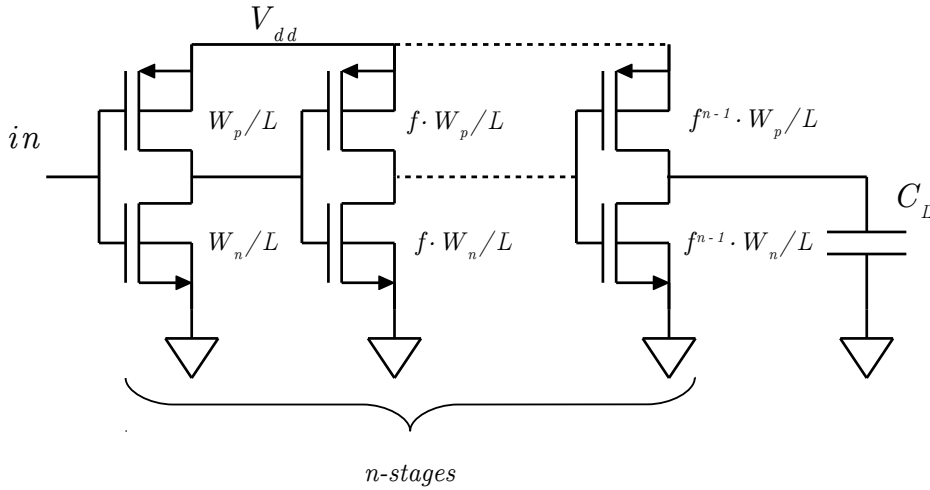


Figura 2.14: Schema generico di un buffer CMOS.

Viene definito *fan-out effettivo* (f) il rapporto tra la capacità di carico che vede ogni invertitore e la sua capacità d'ingresso. Siccome il tempo di ritardo di propagazione dell'invertitore è funzione di questo fattore e siccome si vuole minimizzare il tempo di ritardo dell'intero buffer allora si può dimostrare che questo obiettivo si ottiene facendo in modo che ciascun invertitore CMOS abbia lo stesso f . Si definisce invece *fan-out globale* (F) il rapporto tra la capacità di carico esterna (C_L) e quella di ingresso del primo invertitore. Quindi noto il valore della capacità di carico esterna (dato dalla simulazione del punto operativo del transistor da pilotare), nota la capacità di carico del primo invertitore (di solito coincide con l'invertitore minimo) e imposto un fan-out effettivo di 4 (scelta standard) si può calcolare il numero di stadi necessari:

$$f_i = \frac{C_{ext,i}}{C_{g,i}} = 4, \quad F = \frac{C_L}{C_{g,1}} \quad \rightarrow \quad f = \sqrt[N]{F} \quad \Rightarrow \quad N = \log_f F^1 \quad (2.37)$$

L'unico vincolo è che il numero di stadi deve essere pari per i segnali che pilotano gli switch di tipo nMOS e dispari per il segnale del dispositivo pMOS (poichè ha bisogno del segnale complementare).

Il numero finale di stadi utilizzato per ogni switch è illustrato nella seguente tabella ed è stato scelto anche in base al consumo energetico come spiegato nel paragrafo precedente:

Switch	Sw_1	Sw_2	Sw_3	Sw_4	Sw_5	Sw_6	Sw_7	Sw_8	Sw_9	Sw_{10}	Sw_{11}
Type	pMOS	nMOS	nMOS	nMOS	nMOS	nMOS	nMOS	nMOS	nMOS	nMOS	nMOS
N	7	6	6	6	6	6	6	6	6	6	6

¹Per applicare questa formula bisogna trascurare le capacità parassite di uscita dell'ultimo stadio.

2.2 Phase Divider e Logic Control Unit

Il *Phase Divider* prende in ingresso il segnale di clock (onda quadra con duty cycle del 50%) e a seconda del convertitore selezionato è in grado di fornire due, tre o quattro fasi non sovrapposte di uguale durata. In altre parole è come se fosse un divisore di frequenza programmabile. La durata di ciascuna fase è fissa ed è pari al periodo del clock.

Il principio di funzionamento è che l'impulso fornito dal clock viene campionato in un flip-flop di tipo D e questo bit viene trasmesso periodo dopo periodo agli altri flip-flop. La figura 2.15 mostra lo schema del principio di funzionamento del divisore di frequenza.

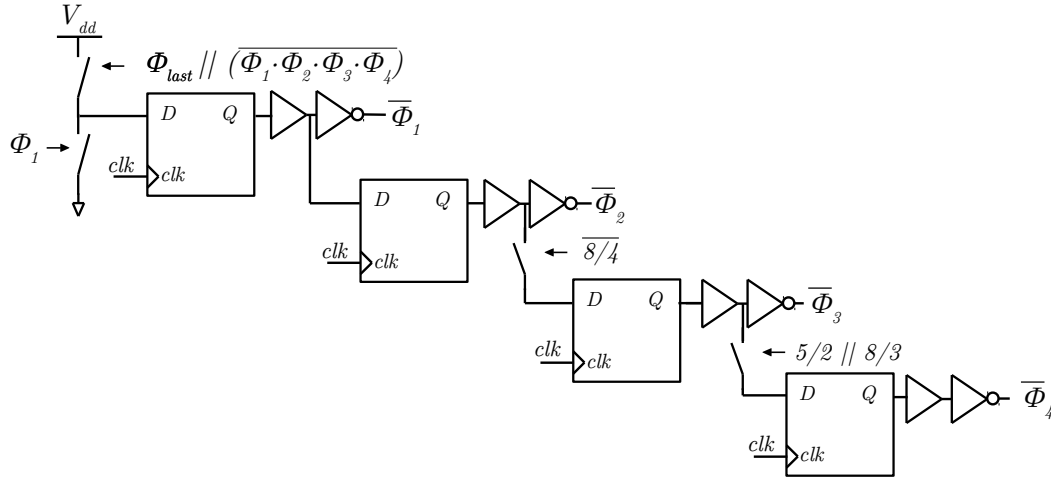


Figura 2.15: Schema logico del phase divider.

Come si può vedere dall'immagine sovrastante i segnali delle fasi sono negati rispetto a quelli normali e il motivo verrà spiegato quando verrà descritto il funzionamento della logic control unit.

La non sovrapposizione delle fasi si ottiene facendo passare il segnale attraverso una catena di ritardo di invertitori la quale funge anche da buffer CMOS per i vari carichi capacitivi delle porte logiche. Il buffer è stato dimensionato come nel caso precedente ovvero si è calcolato il fan out globale e imponendo un fan out effettivo pari a 4 si è trovato il numero corretto di stadi e poi lo si è approssimato al numero dispari successivo.

Il segnale Φ_{last} viene ottenuto facendo una *OR* logica a pass-transistor tra le fasi Φ_2 , Φ_3 e Φ_4 . Il segnale $\overline{\Phi_1 \cdot \Phi_2 \cdot \Phi_3 \cdot \Phi_4}$ serve solo nella condizione di

start-up ovvero quando tutte le uscite sono al valore logico basso. Il flip-flop di tipo D è stato realizzato usando gate di trasmissione ed invertitori.

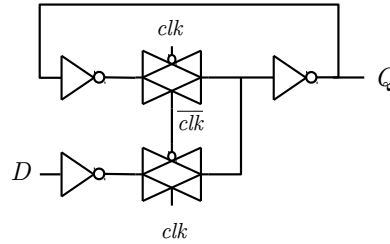


Figura 2.16: Schema circuitale del flip flop di tipo D.

Con il phase divider ora si è in grado di generare il numero di fasi appropriate a seconda del convertitore selezionato. Tutta via ogni switch del scc può essere utilizzato in fasi diverse e in convertitori diversi; la seguente tabella riassume in quali fasi vengono attivati gli switch del scc.

<i>Switch</i>	$\frac{8}{2}$	$\frac{3}{1}$	$\frac{8}{3}$	$\frac{5}{2}$	$\frac{8}{4}$
1	1	1	1, 3	1	1
2	2	2	2	3	—
3	2	2	2	3, 4	2
4	1	—	1	—	—
5	—	—	—	4	2
6	3	3	4	2	—
7	—	1	3	1	1
8	1, 2	2	1, 2	3	—
9	3	3	4	2	—
10	1, 2, 3	2, 3	3	2, 4	—
11	—	1	1, 2, 4	1, 3	1, 2

Tabella 2.16: Tabella riassuntiva delle fasi in cui ciascun switch viene utilizzato.

Dal momento che il segnale per ogni switch è solo uno allora nasce l'esigenza di un avere un blocco logico che esegua la *OR* tra i vari segnali possibili, ma che commuti le operazioni anche in base al convertitore selezionato. Il *logic control unit* è il blocco che ha questa funzione: ogni segnale di controllo per i transistor di scc è il risultato di una operazione logica tra le fasi che a sua volta sono debitamente controllate in base al convertitore selezionato. Per realizzare le porte logiche si è usata la logica statica complementare e per velocizzarne la commutazione sono state usate porte logiche *NAND* invece di *OR*. Tale sostituzione è stata possibile per la seguente uguaglianza:

$$A + B = \overline{\overline{A + B}} = \overline{\overline{A} \cdot \overline{B}} \quad (2.38)$$

L'equazione 2.38 non è altro che una manipolazione dell'operazione $A + B$ usando i teoremi di *De Morgan*. È da questo risultato che si spiega il motivo per cui le fasi in uscita dal phase divider devono essere negate. La porta *NAND* è stata realizzata con la logica statica complementare e si son usate tutte porte con quattro ingressi.

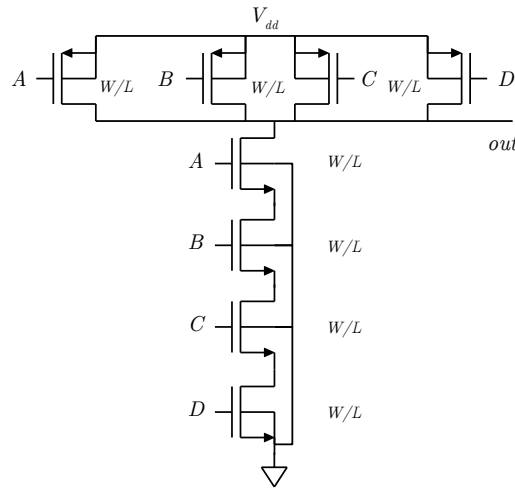


Figura 2.17: Schema circuitale della porta *NAND*.

Il dimensionamento della porta si basa sul fatto che la resistenza equivalente della rete di *pull-up* (PUN) deve essere pari a quella del *pull-down* (PDN) in modo da ottenere una soglia di commutazione pari a metà della tensione di alimentazione. La resistenza della PDN è costituita dalla serie dei quattro transistor nMOS, mentre quella del PUN è costituita da un solo transistor pMOS (questo perchè gli ingressi alla porta sono i segnali delle

fasi ed non essendoci sovrapposizione tra le fasi uno solo di questi si trova ad un valore logico basso tale da accendere il pMOS). Sapendo inoltre che la relazione tra le transconduttanze di processo è: $k_{pMOS} \simeq 4 \cdot k_{nMOS}$ (da simulazioni del punto operativo) allora si dimostra facilmente che i fattori di forma dei transistor sia pMOS che nMOS devono essere uguali.

La figura 2.18 mostra il controllo dello switch numero uno Sw_1 , in accordo con la tabella 2.16 si nota che solo la fase 1 e la 3 vede impegnato il transistor pMOS del scc.

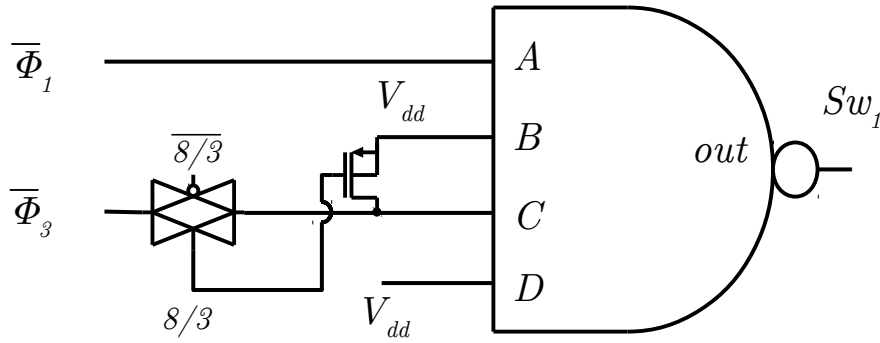


Figura 2.18: Schema circuitale del controllo per lo Sw_1 .

Il gate di trasmissione per la fase 3 impedisce alla porta di commutare quando non è attivo la configurazione 8/3.

Il transistor pMOS della figura 2.18 ha una funzione di *bleeder* ovvero tiene alto il valore logico all'ingresso C della porta *NAND* quando non è attiva la configurazione 8/3 del convertitore. La funzione del bleeder è stata applicata in molti altri punti poichè la corrente di leakage attraverso i gate dei transistor tende a scaricare i nodi fino al punto tale da accendere (anche in sottosoglia) i transistor della porta e quindi a comprometterne il funzionamento. Questa soluzione non introduce consumi di tipo statici poichè si disattiva quando il segnale 8/3 va al livello logico alto e il nodo segue l'andamento della fase 3. Il transistor utilizzato ha dimensioni le minime ($160nm/120nm$).

La figura 2.19 invece mostra il controllo per lo Sw_2 , il blocco con scritto *bleeder* ha la stessa funzione di quello spiegato in precedenza, ma non essendo direttamente controllabile da un solo segnale allora fa un sensing della tensione al nodo B, viene invertito da un inverter CMOS con la soglia sbilanciata verso l'alto e l'uscita dell'invertitore va a pilotare il solito pMOS. La figura 2.20 ne mostra lo schema circuitale.

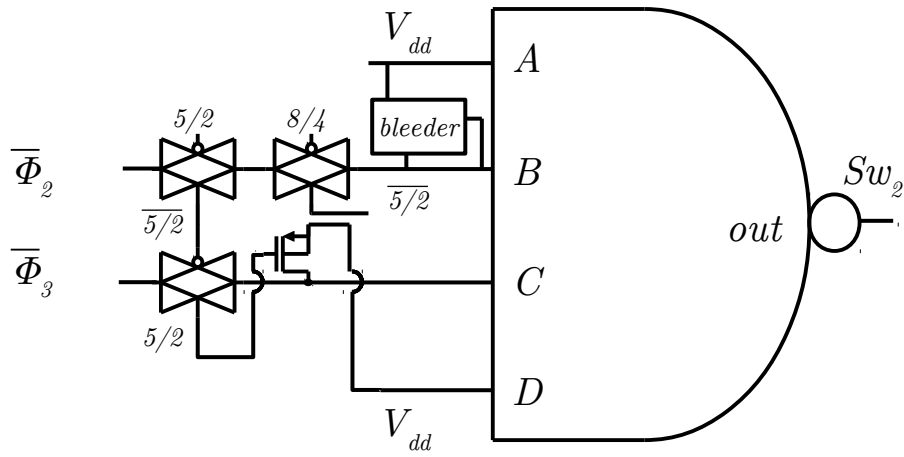
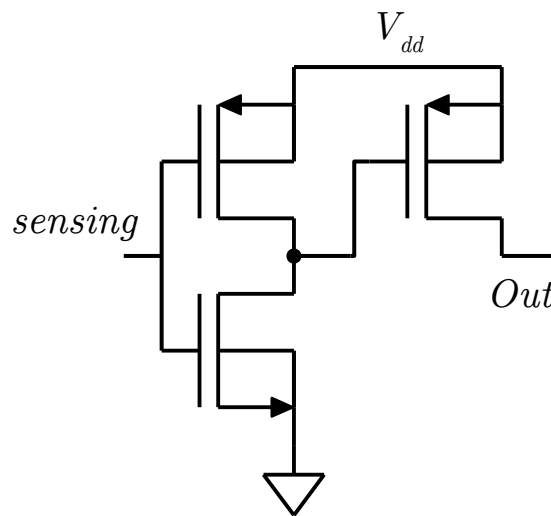
Figura 2.19: Schema circuitale del controllo per lo Sw_2 .

Figura 2.20: Schema circuitale del bleeder con l'inverter ed il nodo di sensing.

Le immagini seguenti rappresentano lo schema di controllo per il resto degli switch.

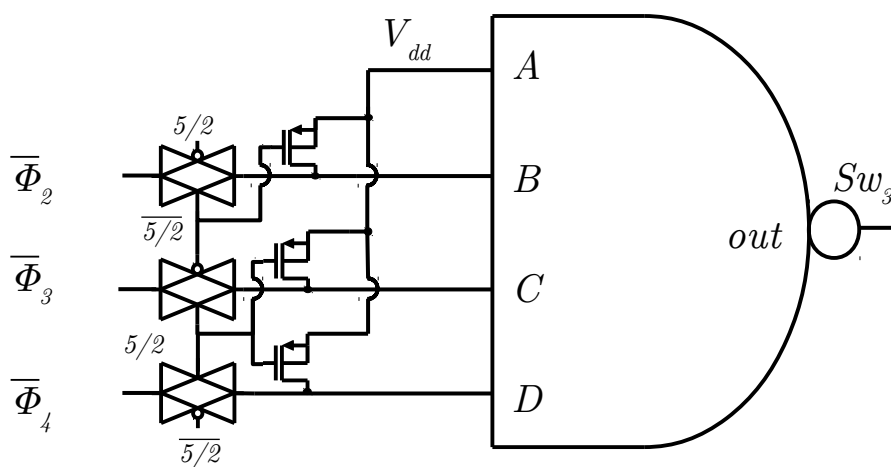


Figura 2.21: Schema circuitale del controllo per lo Sw_3 .

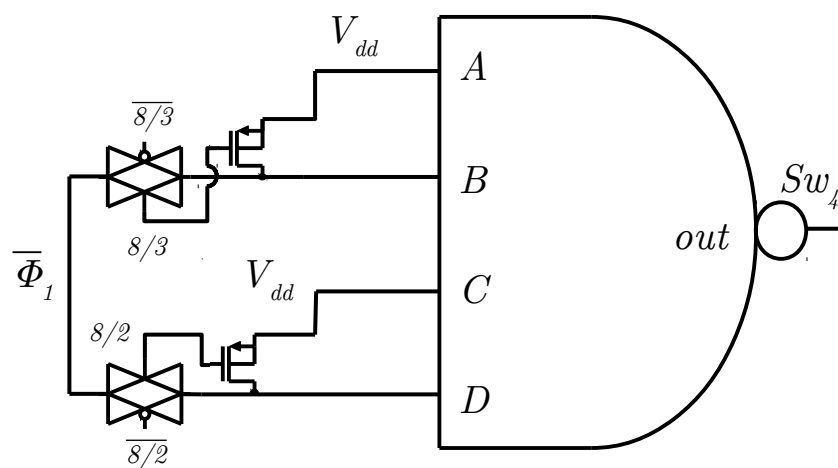
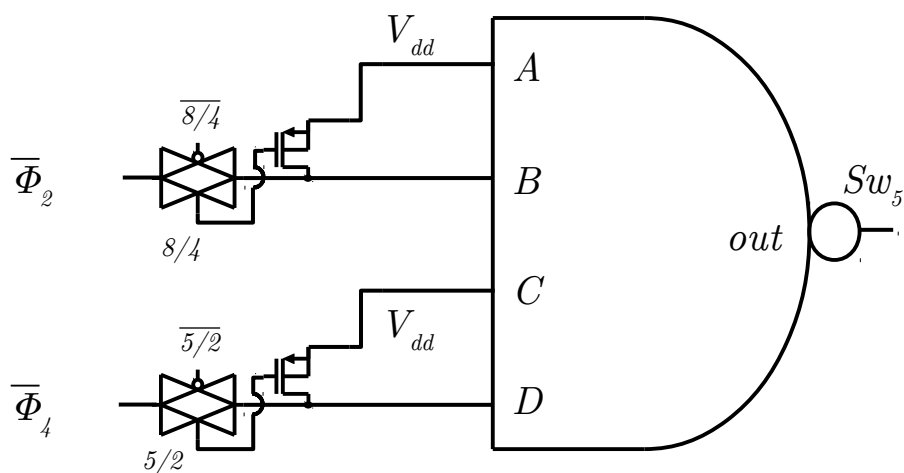
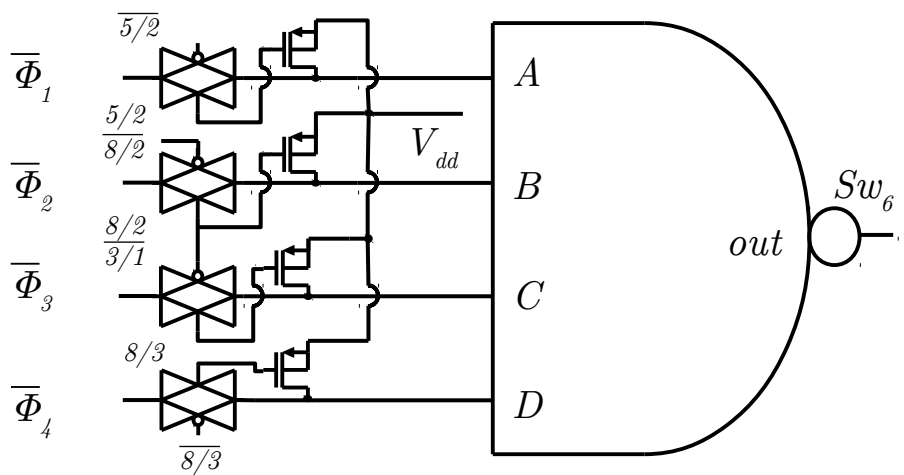


Figura 2.22: Schema circuitale del controllo per lo Sw_4 .

Figura 2.23: Schema circuitale del controllo per lo Sw_5 .Figura 2.24: Schema circuitale del controllo per lo Sw_6 .

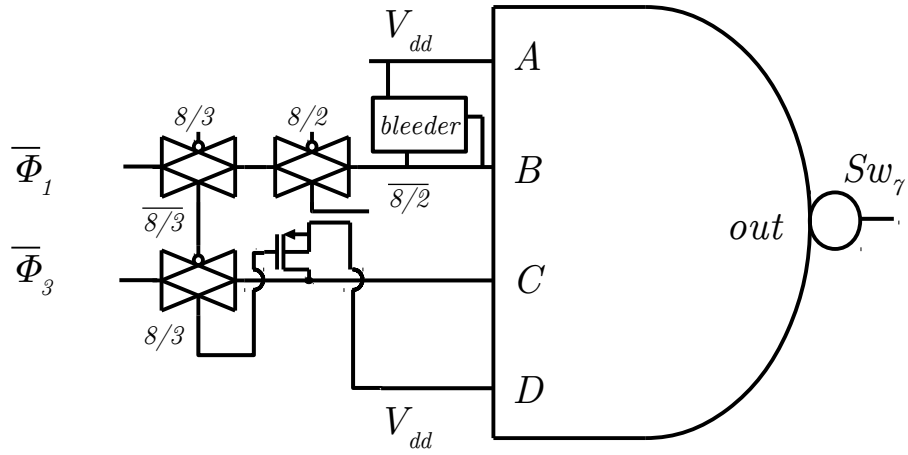


Figura 2.25: Schema circuitale del controllo per lo Sw_7 .

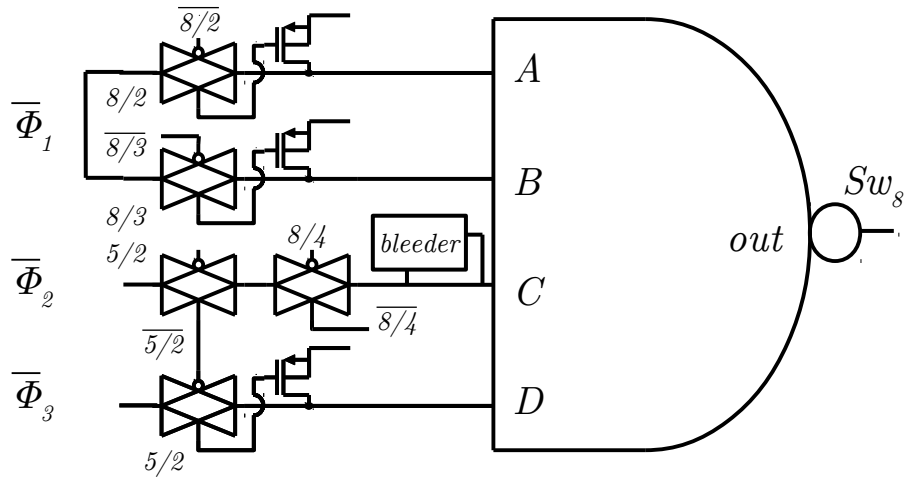
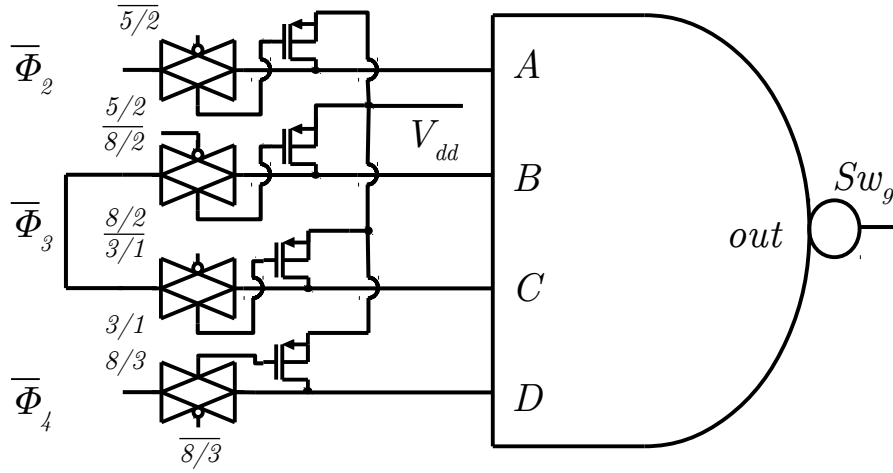
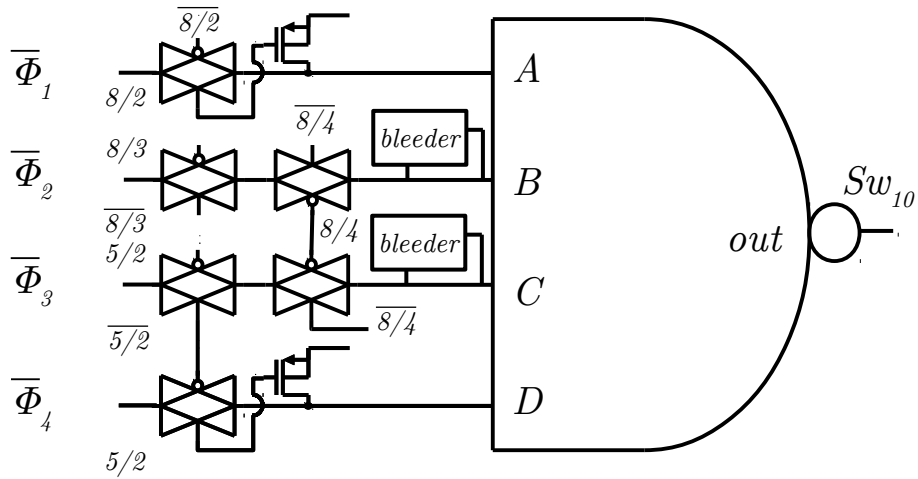


Figura 2.26: Schema circuitale del controllo per lo Sw_8 .

Figura 2.27: Schema circuitale del controllo per lo Sw_9 .Figura 2.28: Schema circuitale del controllo per lo Sw_{10} .

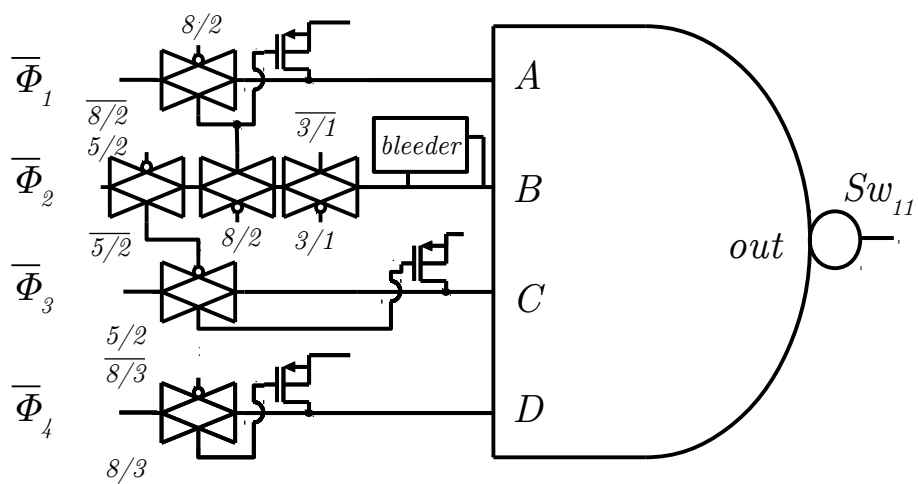


Figura 2.29: Schema circuitale del controllo per lo Sw_{11} .

2.3 Gear Control e Buffer

I blocchi precedentemente descritti e dimensionati assicurano il giusto funzionamento e una buona efficienza del sistema, ma manca ancora la descrizione e progettazione di un blocco in grado di selezionare il corretto convertitore in base alla tensione d'ingresso fornita dall'energy harvester. Il *gear control* ed il relativo *buffer* hanno proprio il compito di confrontare la tensione V_{in} con delle soglie di riferimento, selezionare il relativo convertitore e fornire questa informazione all'unità logica e al phase divider.

L'idea di base per la realizzazione del gear control è quella di utilizzare l'interfaccia di un convertitore *flash* e poi bufferizzare il risultati ottenuti. La figura 2.30 mostra lo schema circuitale con il quale è stato realizzato il gear control. Dalla figura si riconosce la struttura del convertitore flash con la stringa delle resistenze poste in serie una con l'altra per la creazione delle diverse soglie a partire da un riferimento di tensione ed i comparatori che confrontano la tensione d'ingresso con la relativa soglia. Il codice all'uscita di comparatori viene detto *codice termometrico* e bisogna trasformarlo nel codice *1-of-n*. Per fare ciò si esegue una *NAND* tra il segnale di uscita dal comparatore e la versione negata del segnale successivo. Il risultato così ottenuto viene invertito da un inverter CMOS di minime dimensioni e con la tensione di soglia pari a metà della tensione di alimentazione.

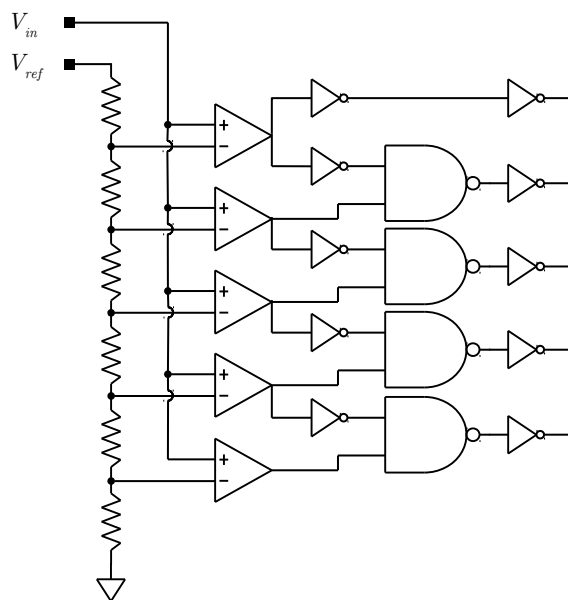


Figura 2.30: Schema circuitale del gear control.

Andando ad osservare i modelli matematici precedentemente descritti dei singoli convertitori e osservando il vero rapporto di conversione è possibile andare a calcolare le soglie di tensioni in modo tale che la tensione di uscita (regolata dal controllore o meno) sia sempre all'interno del massimo ripple di specifica. Una volta note le soglie e nota la tensione di riferimento disponibile si son calcolati i vari valori di resistenza in base anche al massimo carico per il generatore di riferimento.

La procedura per il dimensionamento delle resistenze è descritta nell'equazione 2.39 e il codice MatLab che la implementa è nell'appendice B.2:

$$\begin{cases} R_i = \frac{V_{th-i}}{I} - \sum_{k=1}^{i-1} R_k, & \text{con } i = 1, 2, 3, 4, 5 \\ R_6 = R_{sum} - \sum_{k=1}^5 R_k \end{cases} \quad (2.39)$$

dove V_{th-i} è la i -esima tensione di soglia, R_{sum} è la somma delle resistenze (resistenza vista dal generatore di riferimento). La corrente I è ricavata dalla potenza che deve essere dissipata nella somma delle resistenze:

$$I = \frac{V_{ref}}{R_{sum}} = \frac{V_{ref}}{\frac{V_{ref} \cdot V_{ref}}{P_{diss}}} = \frac{P_{diss}}{V_{ref}}$$

Usando una $V_{ref} = 0.6V$ e richiedendo un consumo da parte del generatore di riferimento pari a $P_{diss} = 5\mu W$ si son ottenuti i seguenti valori di resistenza: $R_1 = 36k\Omega$, $R_2 = 13.44k\Omega$, $R_3 = 6.36k\Omega$, $R_4 = 2.988k\Omega$, $R_5 = 14k\Omega$ e $R_6 = 100m\Omega$. I risultati ottenuti non sono facilmente integrabili.

Invece per quanto riguarda il buffer lo si è dimensionato sempre stimando la capacità di carico totale e, imponendo un fan-out effettivo di quattro, si è calcolato il numero di stadi usando la 2.37. Sono stati sottoposti a buffer anche la versione negata dei segnali poichè vedono lo stesso carico capacitivo.

2.4 Hysteretic Control

I convertitori dc-dc a capacità commutate hanno un rapporto di conversione fisso e al variare della V_{in} , la tensione d'uscita varia nello stesso modo ma moltiplicata per il rapporto di conversione in atto. Nasce quindi l'esigenza di un controllore che monitori la tensione di uscita e la faccia rimanere costante e vicina al valore di riferimento. Da specifica l'uscita dovrebbe rimanere attorno a 1.2V con un ripple massimo del 3%.

Uno dei controllori maggiormente usati in letteratura per i convertitori dc-dc a capacità commutate per l'energy harvesting è il controllo isteretico. L'idea di base di questo tipo di controllore è quella di un termostato per ambienti: se il valore da controllare risulta sotto la soglia minima allora viene attivato il sistema, quando il valore supera la soglia massima allora il controllore impone lo spegnimento del sistema. Questo controllore viene utilizzato in queste applicazioni per la sua semplicità e per il suo basso impatto energetico. La figura 2.31 descrive il principio di funzionamento del controllore isteretico. Il segnale da regolare V_{out} viene confrontato da due soglie una superiore e una inferiore, l'uscita dei due comparatori sono collegate rispettivamente al *reset* e *set* di un flip flop S-R. Quando l'uscita Q del flip-flop è a livello logico alto allora il buffer viene alimentato e il clock viene trasferito, viceversa quando l'uscita Q assume il valore logico 0 allora il buffer non viene alimentato e il clock non viene più trasmesso, nessuna fase viene generata e il circuito viene interdetto e il carico R_{load} viene alimentato solo da C_{out} .

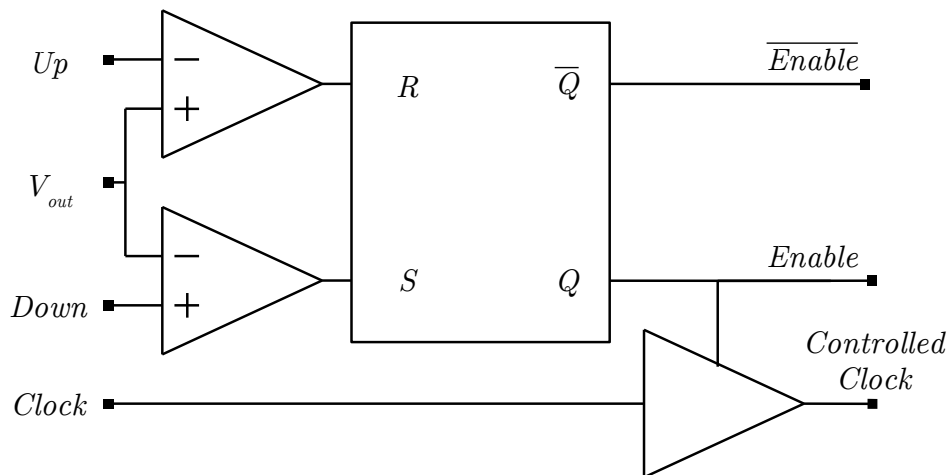


Figura 2.31: Principio di funzionamento del controllore isteretico.

La figura 2.32 invece mostra l'andamento della tensione di uscita del convertitore: i fronti di salita sono lineari a tratti per il fatto che scc sta lavorando, mentre quello di discesa no perchè la curva di scarica di un condensatore con costante di tempo pari a $\tau = R_{load} \cdot C_{out}$.

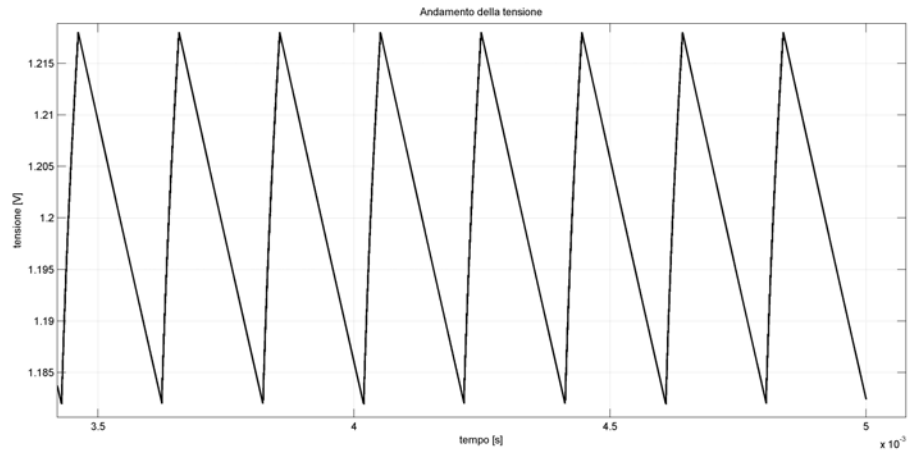


Figura 2.32: Andamento della tensione di uscita sotto controllo isteretico.

Il flip-flop SR è stato implementato come in figura 2.33

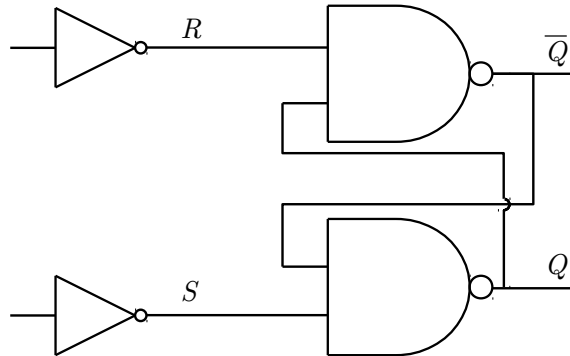


Figura 2.33: Schema dell'implementazione del flip flop SR.

2.5 Start-up

Senza un circuito di start-up il convertitore non sarebbe in grado di accendersi per il fatto che non ha la possibilità di usare fonti energetiche tipo batterie e per il fatto che è auto-alimentato. Serve quindi un circuito che sia in grado di caricare il condensatore di uscita (e quindi alzare la tensione di V_{out}) fino a che il convertitore principale non si accende assieme a tutta la circuiteria di controllo.

Lo schema logico di funzionamento, osservabile in figura 2.34, mostra che non appena viene fornita una $V_{in} > 0.3V$ un oscillatore CMOS costruito mediante un *oscillatore ad anello* comincia a produrre un'onda sinusoidale che viene trasformata in un'onda quadra dopo aver attraversato una cascata di inverter CMOS di minime dimensioni e con soglia centrata a metà della tensione di alimentazione (V_{in}). L'onda quadra che funge da clock del circuito di start-up viene trasformata in clock a due fasi non sovrapposte. Queste due fasi opportunamente bufferizzate attivano un convertitore dc-dc costituito da una pompa di carica di Dickson il quale a sua volta va a caricare direttamente il condensatore di uscita.

Si noti che per il fatto che le tensioni sono ancora molto basse sono stati impiegati tutti transistor a bassa tensione di soglia: *N_LV_12_HSL130E*, *P_LV_12_HSL130E*.

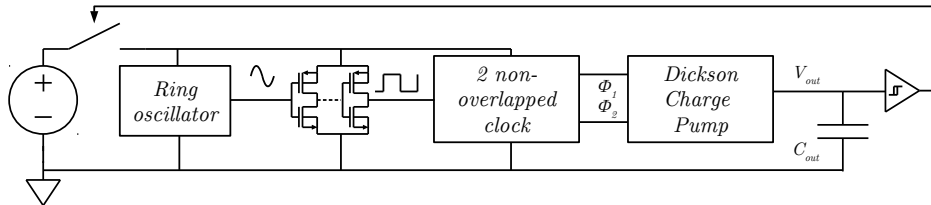


Figura 2.34: Schema logico di funzionamento del circuito di start-up.

L'oscillatore ad anello è stato implementato usando 3 inverter minimi con dei condensatori di carico (*MIMCAPS*: $w = l = 26\mu m \rightarrow 687.12pF$) per raggiungere una frequenza operativa di $f = 14.1MHz$.

Il circuito che genera le due fasi non sovrapposte segue lo schema di figura 2.35, dove una fase cambia il suo valore logico solo quando l'altra ha già fatto la commutazione opposta. La durata della non sovrapposizione tra le due fasi viene impostata e regolata dalla catena di ritardo costituita ancora una volta da inverter minimi posta all'uscita della porta *NAND*.

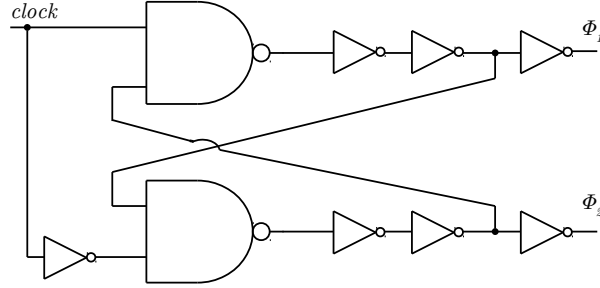


Figura 2.35: Schema di funzionamento logico del circuito che crea due fasi non sovrapposte a partire da un unico segnale di clock.

Da simulazioni parametriche fatte variando la condizione iniziale della tensione di alimentazione (quindi quella di uscita) si è dimostrato che la minima tensione in cui il convertitore principale riesce ad accendersi è di $V_{out} \simeq 0.55V$. Di conseguenza la pompa di Dickson deve essere in grado di raggiungere la tensione di $0.55V$ in uscita con la minima tensione di ingresso, che da specifica risulta: $0.3V$. La figura 2.36 mostra che gli switch sono stati implementati con dei transistor nMOS chiusi a diodo. Da osservare che per la realizzazione di questa pompa di carica sono stati utilizzati tutti transistor a bassa tensione di soglia e costruiti in un *triple-well* ($N_LVBPW_12_HSL130E$, da documentazione hanno una tensione di soglia tipica di $V_{th}^{LVT} = 0.21V$).

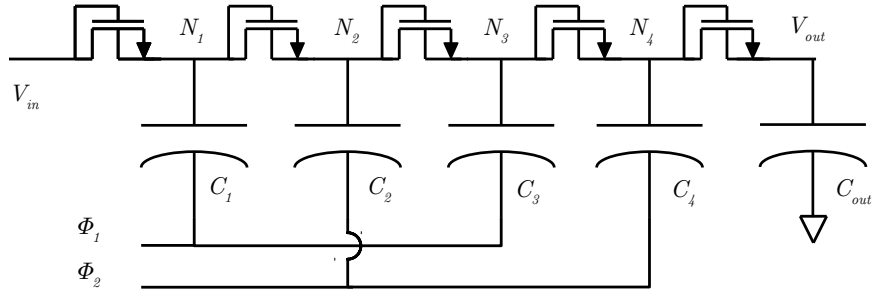


Figura 2.36: Schema circuitale della pompa di carica.

Andando ad analizzare il funzionamento della pompa di carica si vede che quando il primo diodo a sinistra va in conduzione (fase Φ_2) carica il condensatore C_1 a $V_1 = V_{in} - V_d$, poi quando la fase Φ_1 sale manda in inversa il primo diodo e accende il secondo diodo che a sua volta carica il secondo condensatore alla tensione $V_2 = V_1 + V_{in} - V_d = 2V_{in} - 2V_d$. Estendendo il procedimento si nota che il nodo N_3 si carica alla tensione di $V_3 = 3V_{in} - 3V_d$ ed il nodo N_4 si porta ad una tensione media di $V_4 = 4V_{in} - 4V_d$. Di conseguenza il nodo di uscita della pompa di carica che coincide con il nodo di uscita del scc si porta ad una tensione di $V_{out} = V_4 + V_{in} - V_d = 5V_{in} - 5V_d$. In presenza di carico esterno (circuiteria di controllo del circuito principale) la tensione all'uscita del scc si trasforma in:

$$V_{out} = 5V_{in} - 5 \left(V_d + \frac{I_{out}}{2\pi f C_{out}} \right)$$

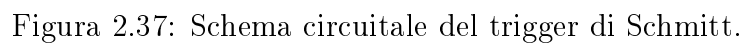
dove I_{out} è la corrente assorbita dal carico esterno, e f è la frequenza con la quale il circuito lavora.

Dato che da simulazioni di efficienza si è notato che la circuiteria di controllo assorbe al più $200\mu W$ di potenza allora si può esprimere la corrente del carico I_{out} in funzione della potenza e della tensione di uscita: $I_{out} = \frac{P_{out}}{V_{out}}$. Quindi si ottiene una formula risolvibile tramite ricorsione:

$$V_{out} = 5V_{in} - 5 \left(V_d + \frac{P_{out}}{2\pi f C_{out} V_{out}} \right) \quad (2.40)$$

Da simulazione la caduta di tensione V_d vale circa $0.15V$, inoltre noti il resto dei parametri, grazie all'equazione sovrastante si ricava il valore della tensione di uscita a regime della pompa di Dickson: $V_{out} = 750mV$. Tale valore risulta essere maggiore di $0.55V$ richiesti e quindi il resto del circuito dovrebbe riuscire ad accendersi. I transistor della pompa di carica hanno tutti una larghezza di canale di $W = 4\mu m$ e lunghezza minima; i condensatori della pompa invece hanno dimensione $w = l = 53\mu m$, multiplier $m = 10$ e quindi una capacità di $28.33pF$.

Il trigger di Schmitt invece è stato realizzato come in figura 2.37 e le dimensioni sono state scelte in modo da avere una soglia inferiore di $62mV$ circa e una soglia superiore di $150mV$ circa. Quando il segnale V_{out} partizionato da R_1 e R_2 raggiunge la soglia superiore allora il circuito di startup smette di funzionare e lascia che il resto del circuito si porti a regime. Al fine di garantire un certo margine di sicurezza per l'accensione si è impostato lo sgancio del circuito di start-up quando la tensione di uscita è $V_{out} = 0.75V$. Questo significa che $R_1 = 4 \cdot R_2$, nell'ottica di non creare troppo carico per la pompa di carica si è impostato $R_1 = 400k\Omega$ e $R_2 = 100k\Omega$ (ben sapendo che non sono facilmente integrabili).



2.6 Power Good

Il blocco di *power good* ha lo scopo di emulare la circuiteria di controllo dell'alimentazione di un eventuale microprocessore che deve essere alimentato dal convertitore dc-dc. Infatti un microprocessore non parte finchè la sua tensione di alimentazione non è stabile e pari al valore di riferimento. Quindi durante tutto il transitorio iniziale di accensione e fino al momento in cui la tensione V_{out} non ha raggiunto il valore stabilito, il carico esterno non viene connesso alla porta di uscita del convertitore.

La figure sottostante mostra lo schema semplificato e semi-ideale utilizzato per emulare questo circuito.

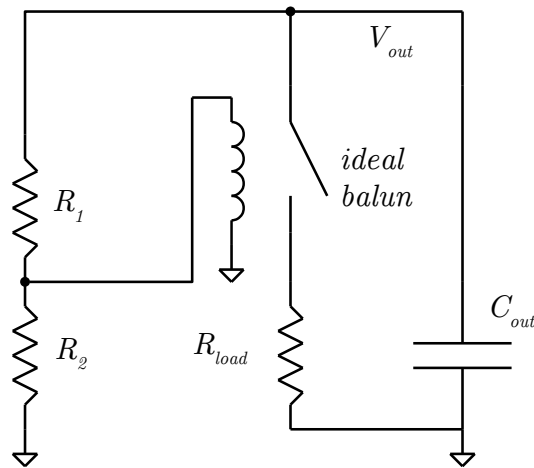


Figura 2.38: Schema per la realizzazione del blocco di power good.

L'idea di base è quella di fare un sensing della tensione di uscita opportunamente scalata dalle resistenze R_1 e R_2 e una volta raggiunta la tensione desiderata lo switch si chiude e il carico risulta connesso alla porta di uscita del convertitore. È stato possibile usare un ideal balun per il fatto che si deve emulare solo il comportamento di un circuito che risulterebbe esterno al convertitore.

Capitolo 3

Simulazioni

Il paragrafo che segue contiene le simulazioni dei convertitori disegnati e simulati con Cadence Virtuoso Design Framework II. Le simulazioni sono state fatte in modalità *conservative*, e l'unica condizione iniziale è stata per la tensione di uscita che è stata posta a $V_{out} = 1.2V$ solo per facilitare il raggiungimento del regime stazionario altrimenti la simulazione sarebbe durata molto di più. È stato imposto un parametro $cmin = 20fF$. I dati di simulazione sono stati convertiti nel formato *comma-separated values* (.csv) e poi rielaborati in MatLab.

Il carico è stato simulato con una resistenza del valore di 1440Ω in modo da assorbire una potenza di $1mW$ circa, la frequenza del generatore di clock è di $f = 2MHz$ ($T = 500ns$). Le tensioni di ingresso per i vari convertitori sono state scelte in modo tale che siano appena al di sopra delle soglie di commutazione del gear control (in modo tale che il controllore non intervenga).

3.1 Verifica funzionamento convertitori

Convertitore 8/2

Come prima simulazione si è verificato corretto funzionamento del sistema: le figure 3.1, 3.2 e 3.3 mostrano la corretta sequenza di accensione degli switch in base alla fase:

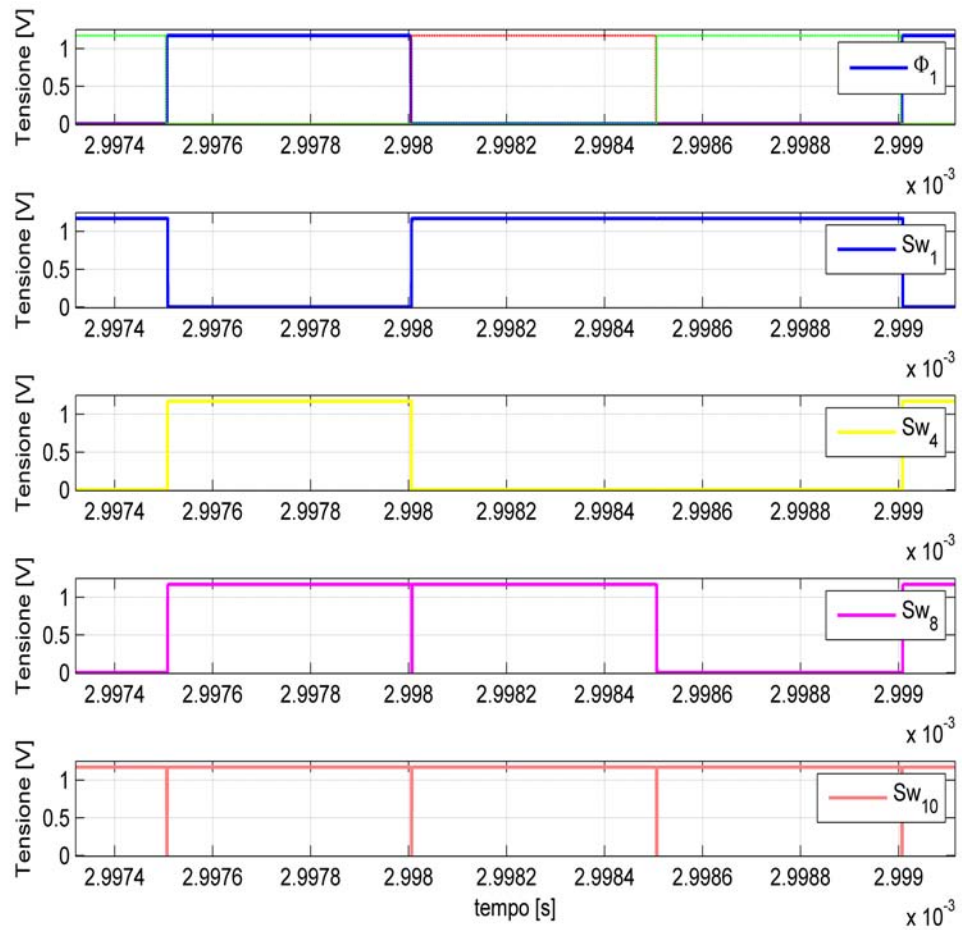


Figura 3.1: Ordine di accensione per la fase Φ_1 .

La tabella 2.16 mostra che nella fase Φ_1 vengono accesi gli switch: Sw_1 , Sw_4 , Sw_8 e Sw_{10} e la figura 3.1 lo conferma (si ricorda che il transistor Sw_1

essendo di tipo pMOS ha bisogno di un segnale complementare). La stessa tabella mostra inoltre che nella fase Φ_2 devo accendersi i transistor: Sw_2 , Sw_3 , Sw_8 e Sw_{10} e questo viene confermato dall'immagine 3.2.

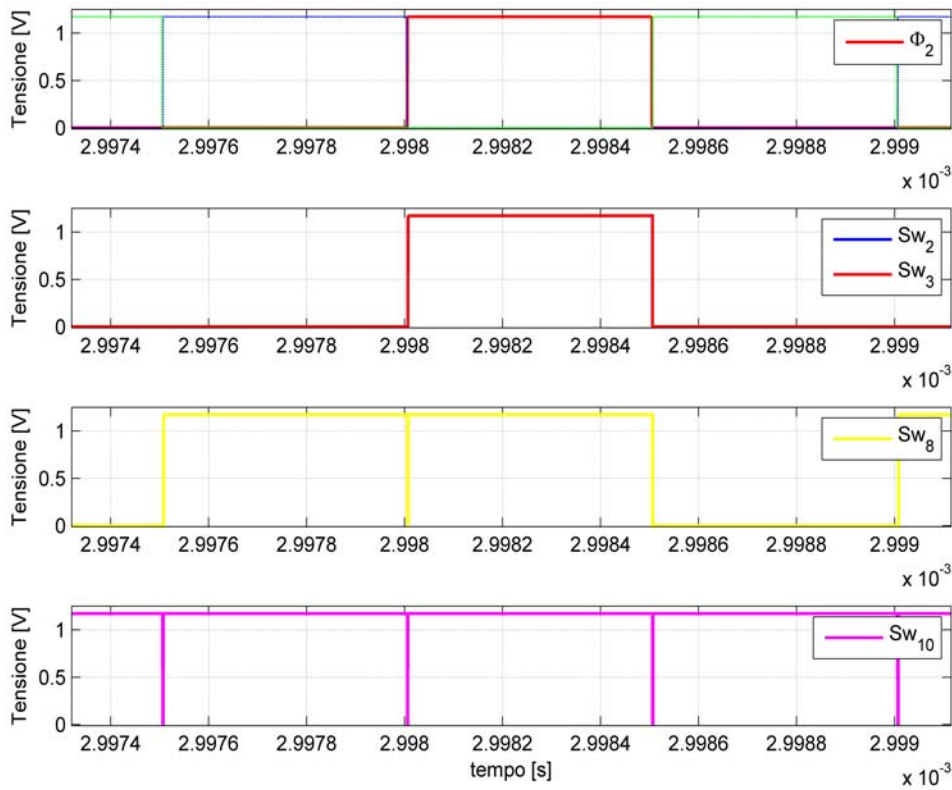


Figura 3.2: Ordine di accensione per la fase Φ_2 .

In queste immagini sono stati omessi i grafici degli altri switch per il semplice fatto che nella fase evidenziata risultavano spenti. Per quanto riguarda la fase Φ_3 , seguendo le configurazioni descritte in precedenza, dovrebbero accendersi i transistor: Sw_6 , Sw_9 e Sw_{10} . L'immagine sottostante conferma l'ordine di accensione anche per l'ultima fase.

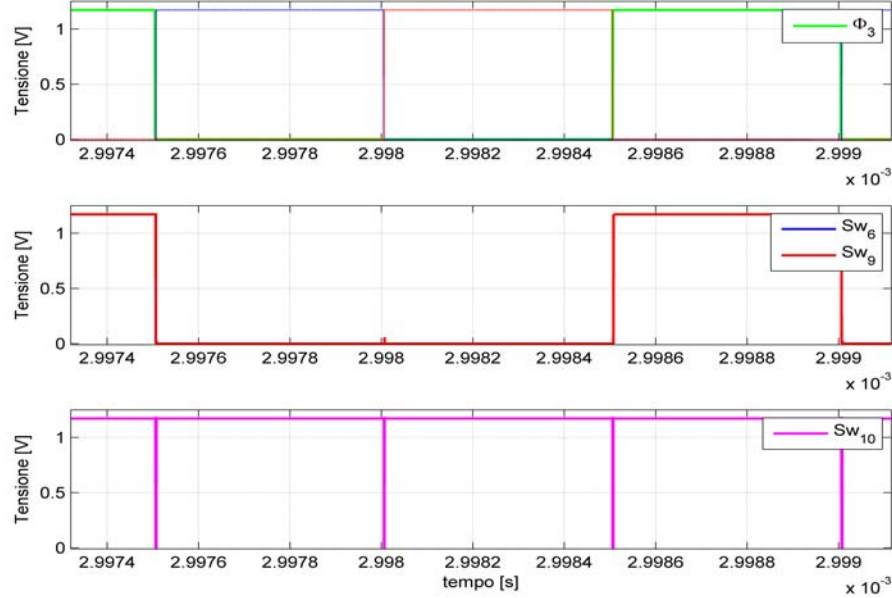


Figura 3.3: Ordine di accensione per la fase Φ_3 .

Con l'aiuto dei grafici ora è più facile vedere una correlazione tra il periodo di utilizzo del transistor e la sua resistenza serie. Infatti andando ad osservare i valori della resistenza serie che si è calcolata nella sezione 2.1.7 si nota che maggiore è il periodo in cui un transistor deve rimanere acceso, minore è la resistenza serie che esso deve avere. L'osservazione era intuibile anche con i soli dati teorici, ma l'ausilio dei grafici l'ha reso più facile.

La figura 3.4 invece mostra l'andamento della tensione di uscita: il tratto iniziale dove si vede l'azione del controllore è solo dovuto al fatto che è stata impostata la condizione iniziale sulla tensione di uscita. La cosa importante da notare è che l'andamento teorico (con le perdite resistive degli switch) prevedeva una tensione a regime di $V_{out,teorico} = 1.1713V$ (con una tensione di ingresso di $V_{in} = 0.301$), mentre i dati di simulazione mostrano che la tensione di uscita media a regime vale $\overline{V_{out,sim}} = 1.1705V$. Si è dovuto calcolare la tensione media a regime a causa del ripple intrinseco. I due valori sono simili al:

$$\frac{\overline{V_{out,sim}}}{V_{out,teorico}} = 99.93\%$$

ciò significa che il modello analizzato e simulato in MatLab è corretto.

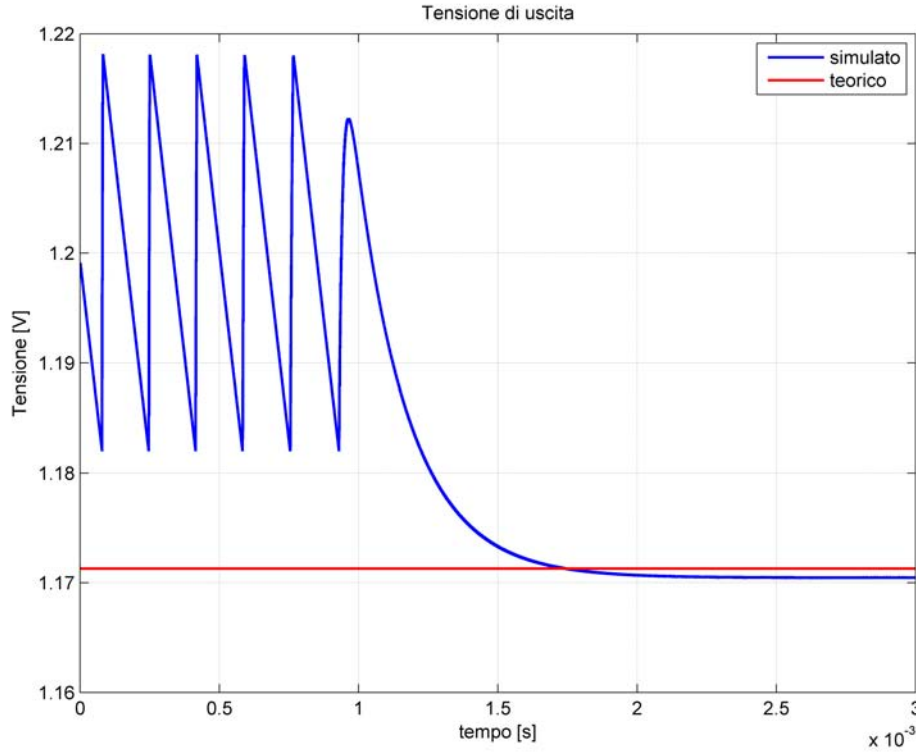


Figura 3.4: Andamento della tensione di uscita.

Il modello teorico che si era calcolato per la tensione di uscita è:

$$V_{out} = 4V_{in} - \frac{V_{out}}{R_{load}} \cdot 3 \cdot (R_{sw-\Phi_1} + R_{sw-\Phi_2} + 4R_{sw-\Phi_3})$$

Il rapporto di conversione è: $\frac{1.1705}{0.301} = 3.89$.

L'immagine 3.5 mostra l'andamento della tensione sul condensatore C_1 e si può ben osservare che il valore teorico e quello simulato quasi coincidono infatti essendo rispettivamente: $V_{C1,teorico} = 0.5857V$ e $\overline{V_{C1,sim}} = 0.5851V$. Il modello teorico usato in MatLab per il calcolo della tensione V_{C1} a regime è:

$$V_{C1} = V_{in} + V_{C2} - \frac{R_{sw-\Phi_2} \cdot I_{out}}{R_{load} \cdot D}$$

e si ricava facilmente dal sistema 2.25.

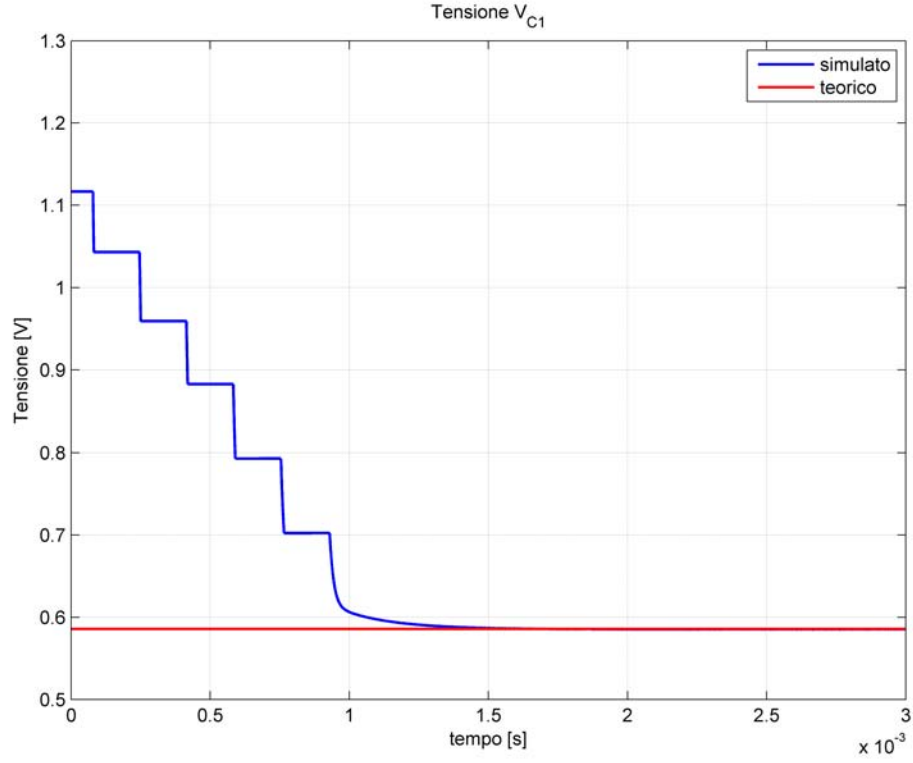


Figura 3.5: Andamento della tensione sul condensatore C_1 .

La figura 3.6 mostra come anche in questo caso il modello teorico predice con una buona precisione la simulazione. La tensione di uscita del modello matematico è $V_{C2,teorico} = 0.2928V$ mentre quella della simulazione è $\overline{V_{C2,sim}} = 0.2927V$. La corrispondenza tra i due valori è del:

$$\frac{\overline{V_{C2,sim}}}{V_{C2,teorico}} = 99.96\%$$

Il modello teorico usato in MatLab per il calcolo della tensione V_{C2} a regime è:

$$V_{C2} = V_{in} - 2 \cdot \frac{R_{sw-\Phi_3} \cdot V_{out}}{R_{load} \cdot D}$$

anche quest'ultima espressione è stata ricavata dal modello descritto in 2.1.7.

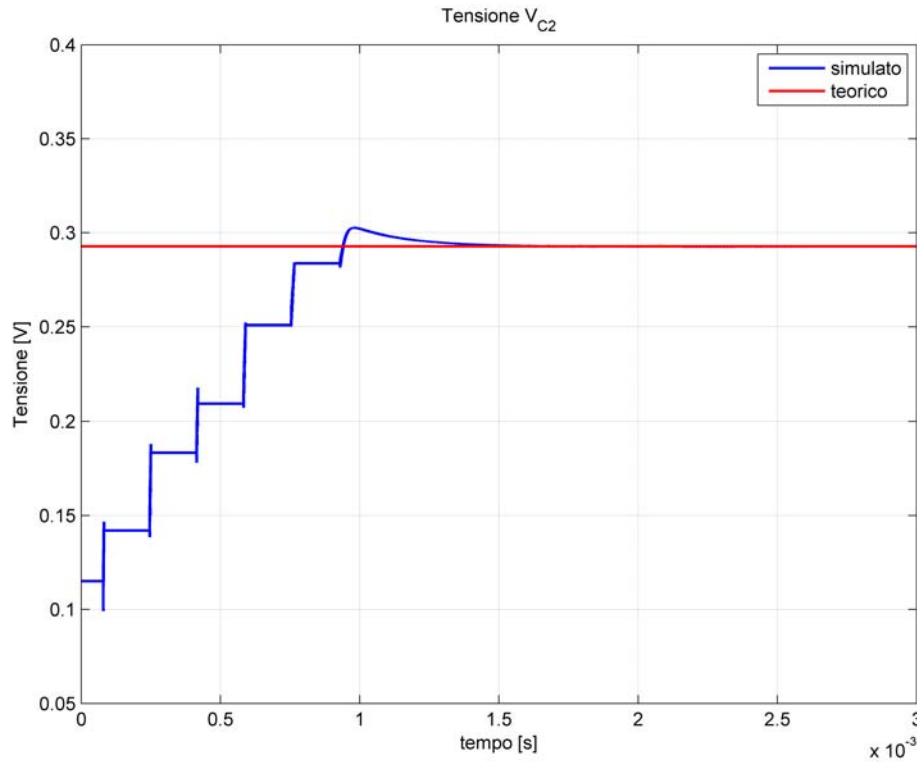


Figura 3.6: Andamento della tensione sul condensatore C_2 .

Una volta verificato che il funzionamento della parte logica è corretto e che il modello teorico con le relative perdite coincide con quello della simulazione si può andare ad osservare in dettaglio come varia la tensione ai capi dei condensatori durante le varie fasi. Ricordando la figura 2.2 ci si può aspettare il seguente comportamento:

- fase Φ_1 :
 - La tensione di uscita sale poichè il condensatore di uscita C_{out} si carica;
 - I condensatori C_1 e C_2 si scaricano;
- fase Φ_2 :
 - La tensione di uscita scende poichè il condensatore di uscita si sta scaricando a causa del carico R_{load} ;
 - Il condensatore C_1 si sta caricando quindi la sua tensione sale;

- Il condensatore C_2 si sta scaricando;
- fase Φ_3 :
 - La tensione di uscita scende poichè il condensatore di uscita si sta scaricando a causa del carico R_{load} ;
 - Il condensatore C_1 è lasciato flottante e quindi la sua tensione rimane costante;
 - Il condensatore C_2 si sta caricando quindi la sua tensione sale;

La figura 3.7 rispetta esattamente quanto appena specificato.

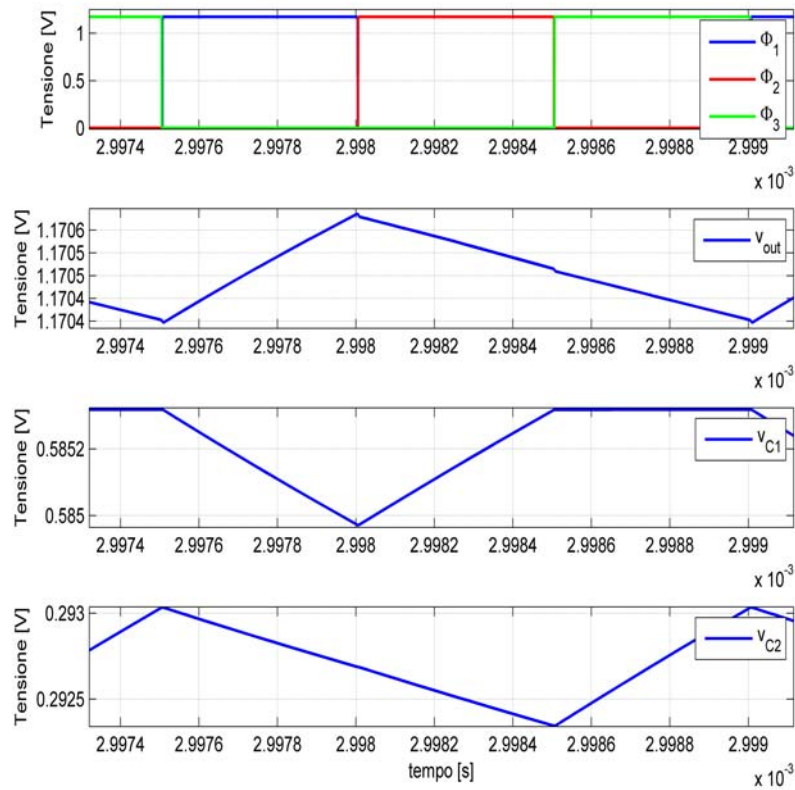


Figura 3.7: Andamento della tensioni.

Un piccolo dettaglio che si può osservare dalla figura sovrastante è che sono presenti dei piccoli picchi di tensione nel grafico della V_{out} . Tali picchi

sono da ricondurre al fenomeno del *charge injection*. Infatti, viste le notevoli dimensioni del transistor Sw_1 , la carica che il transistor richiama al momento dell'accensione o che libera al momento dello spegnimento causa rispettivamente un undershoot ed un overshoot della tensione.

L'ultimo controllo da effettuare per confermare la correttezza dello schematico e del suo funzionamento è stato verificare se, con il dimensionamento ottenuto (dei condensatori e dei transistor), il convertitore lavora effettivamente nella condizione di fast switching mode. Per verificarlo è bastato andare ad osservare un ingrandimento della tensione sul condensatore e della corrente. Secondo quanto detto in precedenza la tensione ai capi dei condensatori dovrebbe avere andamenti lineari (e non esponenziali tipici del condensatore in soft switching mode) e di conseguenza la corrente che fluisce sul condensatore dovrebbe essere costante. Le figure 3.8 e 3.9 mostrano proprio l'andamento descritto e quindi si può confermare che il convertitore lavora in fast switching mode.

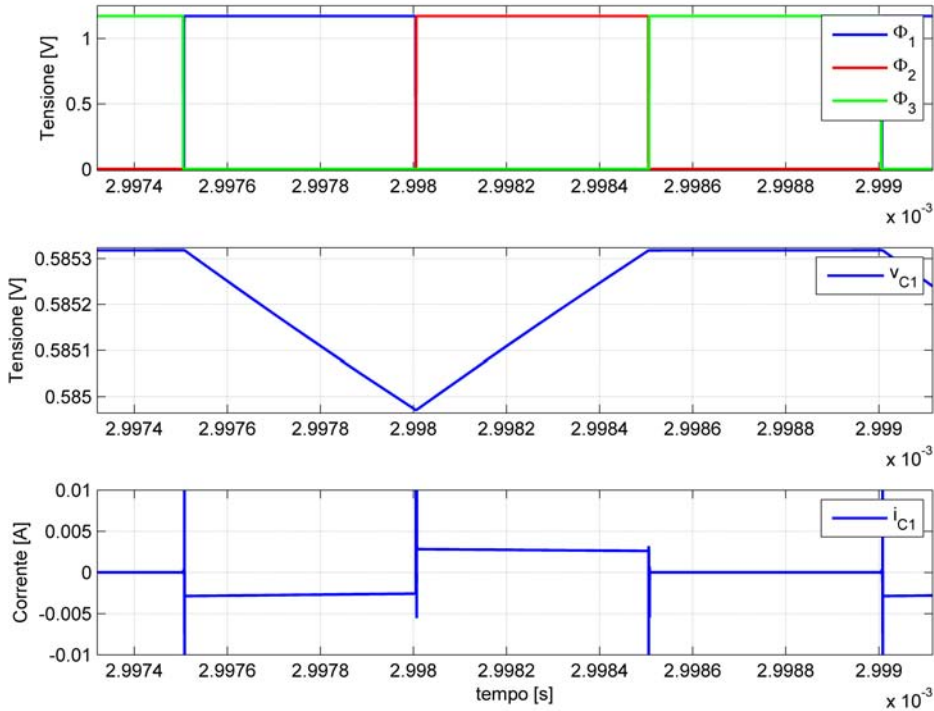


Figura 3.8: Verifica del funzionamento in FSL di C_1 .

Gli impulsi di corrente che si vedono in entrambe le immagini sono da attribuirsi ai transistori tra il cambio delle fasi.

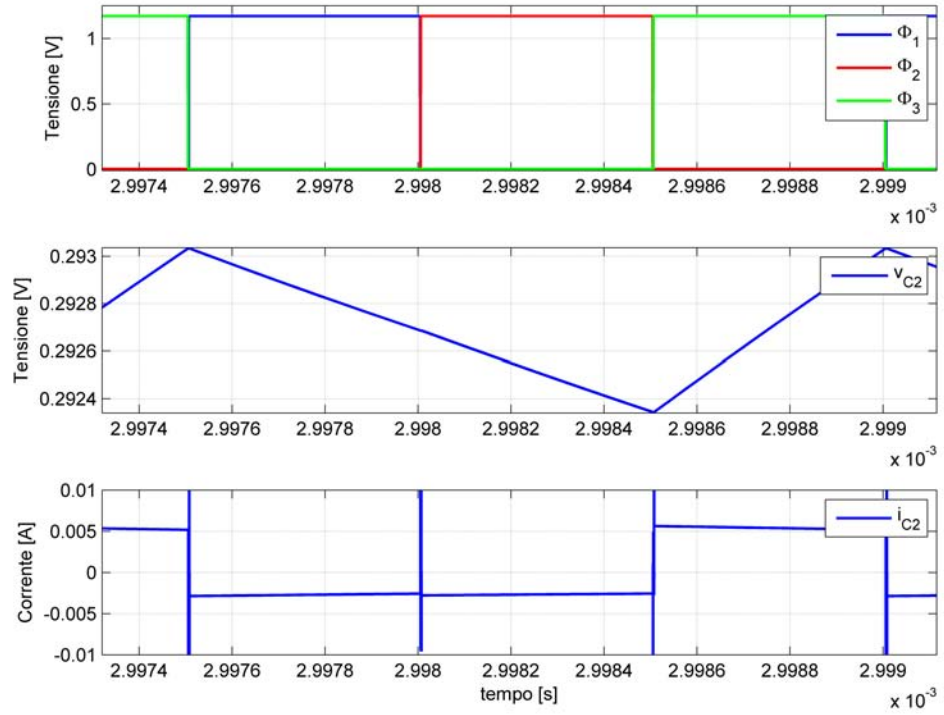


Figura 3.9: Verifica del funzionamento in FSL di C_2 .

Convertitore 3/1

Nella prima simulazione si è verificato corretto funzionamento del sistema: le figure 3.10, 3.11 e 3.12 mostrano la corretta sequenza di accensione degli switch in base alla fase:

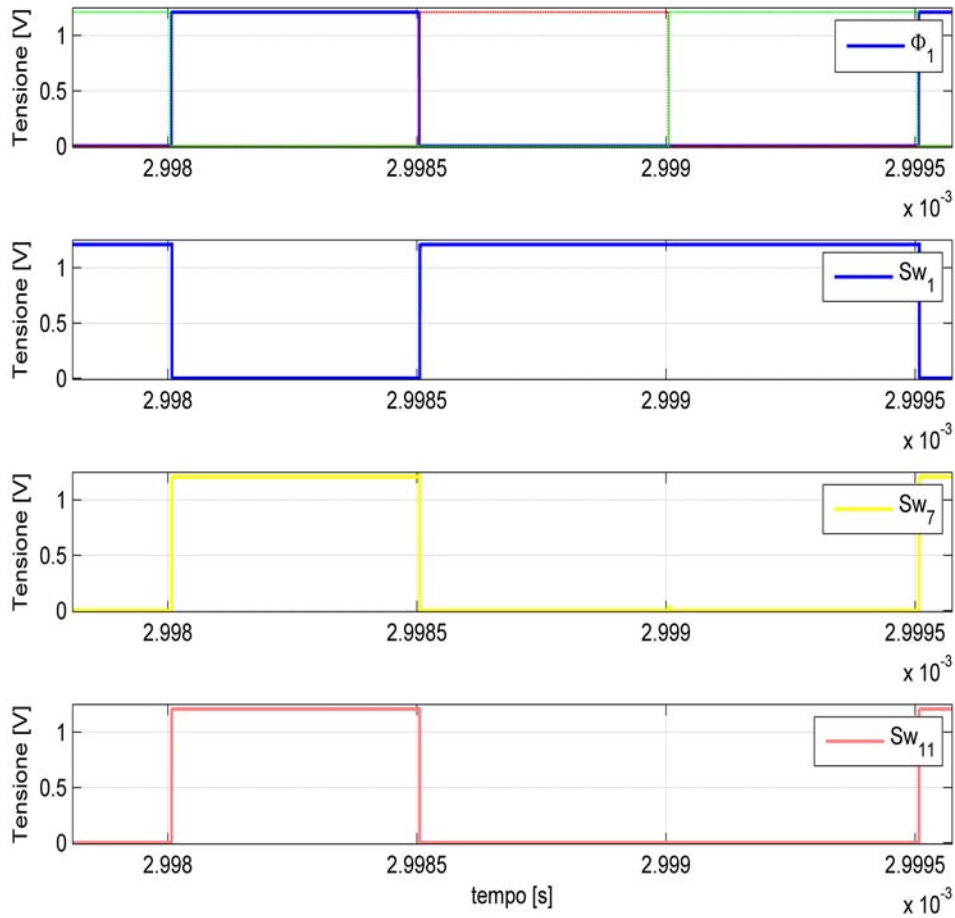


Figura 3.10: Ordine di accensione per la fase Φ_1 .

La tabella 2.16 mostra che nella fase Φ_1 vengono accesi gli switch: Sw_1 , Sw_7 e Sw_{11} e la figura 3.10 lo conferma (si ricorda che il transistor Sw_1 essendo di tipo pMOS ha bisogno di un segnale complementare). La stessa

tabella mostra inoltre che nella fase Φ_2 devo accendersi i transistor: Sw_2 , Sw_3 , Sw_8 e Sw_{10} e questo viene confermato dall'immagine 3.11.

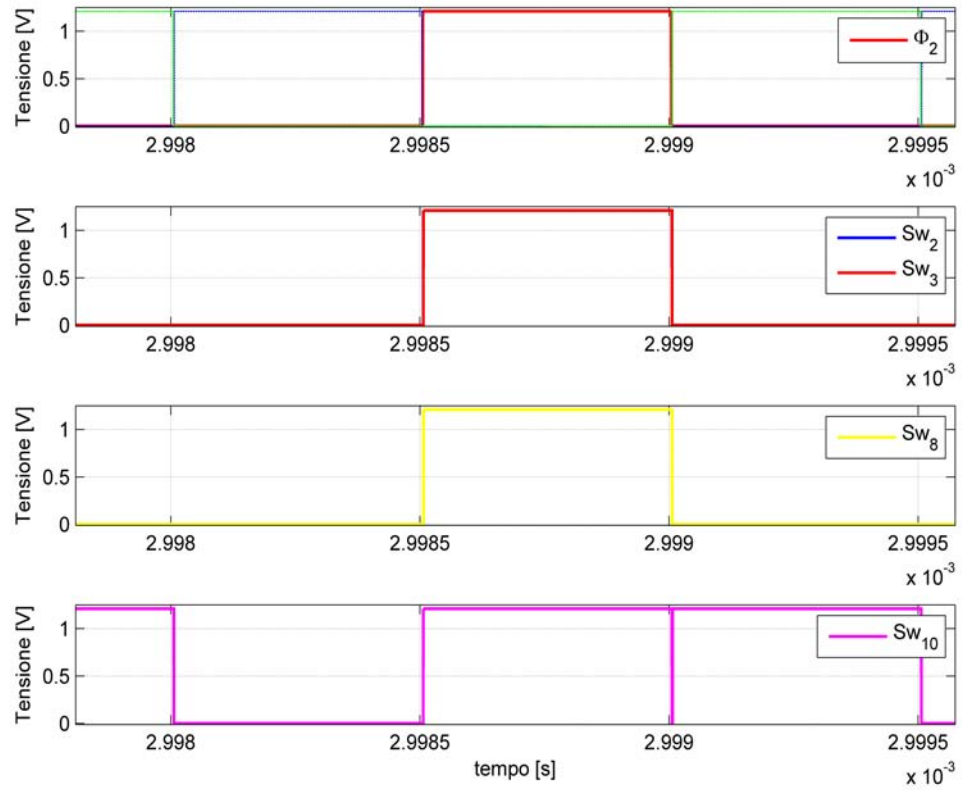


Figura 3.11: Ordine di accensione per la fase Φ_2 .

In queste immagini sono stati omessi i grafici degli altri switch per il semplice fatto che nella fase evidenziata risultavano spenti. Per quanto riguarda la fase Φ_3 , seguendo le configurazioni descritte in precedenza, dovrebbero accendersi i transistor: Sw_6 , Sw_9 e Sw_{10} . L'immagine sottostante conferma l'ordine di accensione anche per l'ultima fase.

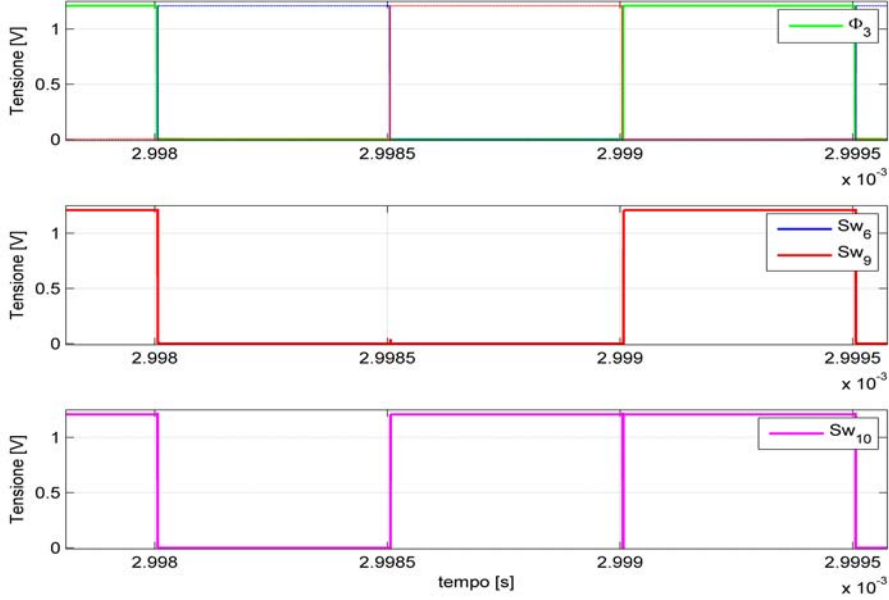


Figura 3.12: Ordine di accensione per la fase Φ_3 .

Anche in questo caso, con l'aiuto dei grafici, è più facile vedere la correlazione tra il periodo di utilizzo del transistor e la sua resistenza serie. Infatti andando ad osservare i valori della resistenza serie che si è calcolata nella sezione 2.1.7 si nota che maggiore è il periodo in cui un transistor deve rimanere acceso, minore è la resistenza serie che esso deve avere. L'osservazione era intuibile anche con i soli dati teorici, ma l'ausilio dei grafici l'ha reso più facile.

La figura 3.13 invece mostra l'andamento della tensione di uscita: il tratto iniziale dove si vede l'azione del controllore è solo dovuto al fatto che è stata impostata la condizione iniziale sulla tensione di uscita $V_{out} = 1.2V$. La cosa importante da notare è che l'andamento teorico (con le perdite resistive degli switch) prevedeva una tensione a regime di $V_{out,teorico} = 1.2101V$ (con una tensione di ingresso di $V_{in} = 0.413V$), mentre i dati di simulazione mostrano che la tensione di uscita media a regime vale $\overline{V_{out,sim}} = 1.2090V$. Si è dovuto calcolare la tensione media a regime a causa del ripple intrinseco. I due valori sono simili al:

$$\frac{\overline{V_{out,sim}}}{V_{out,teorico}} = 99.94\%$$

ciò significa che il modello analizzato e simulato in MatLab è corretto.

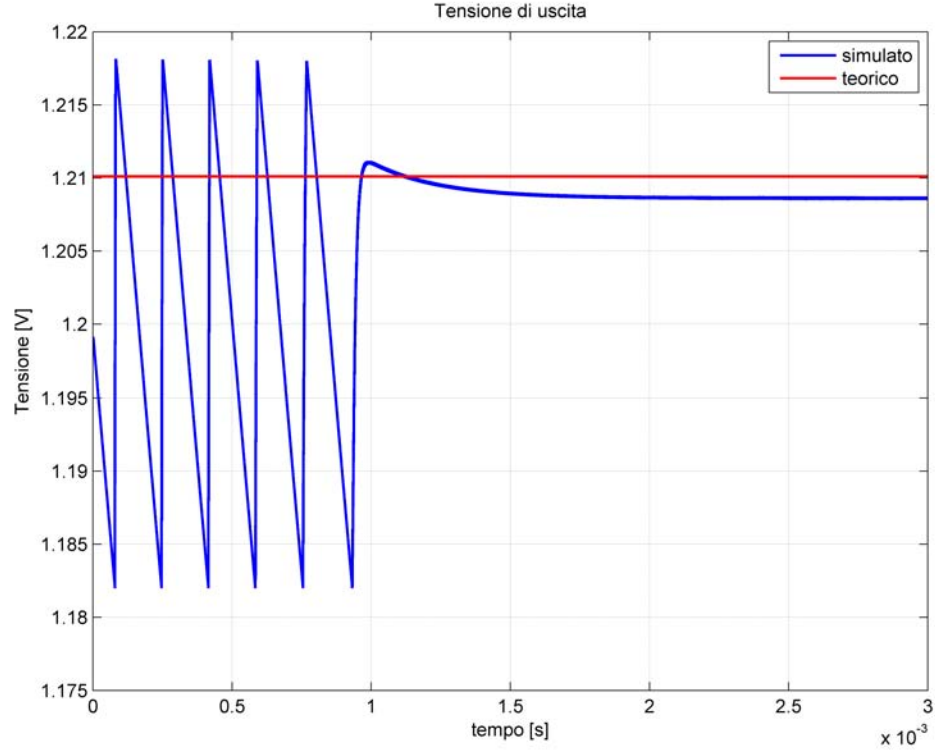


Figura 3.13: Andamento della tensione di uscita.

Il modello teorico che si era calcolato per la tensione di uscita è:

$$V_{out} = 3V_{in} - \frac{V_{out}}{R_{load}} \cdot (R_{sw-\Phi_1} + R_{sw-\Phi_2} + R_{sw-\Phi_3})$$

Il rapporto di conversione è: $\frac{1.2090}{0.413} = 2.93$.

L'immagine 3.14 mostra l'andamento della tensione sul condensatore C_1 e si può osservare che il valore teorico e quello simulato quasi coincidono infatti essendo rispettivamente: $V_{C1,teorico} = 0.8153V$ e $\overline{V_{C1,sim}} = 0.8017V$. Il modello teorico usato in MatLab per il calcolo della tensione V_{C1} a regime è:

$$V_{C1} = V_{in} + V_{C2} - \frac{R_{sw-\Phi_3} \cdot V_{out}}{R_{load} \cdot D}$$

e si ricava facilmente dal sistema 2.28.

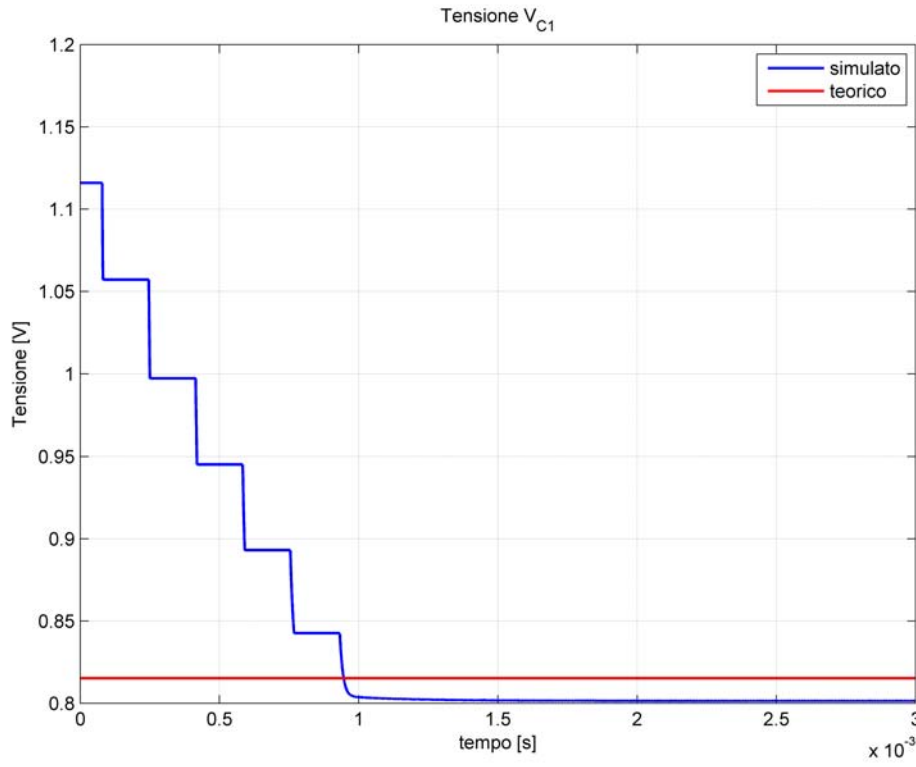


Figura 3.14: Andamento della tensione sul condensatore C_1 .

La figura 3.15 mostra come anche in questo caso il modello teorico predice con una buona precisione la simulazione. La tensione di uscita del modello matematico è $V_{C2,teorico} = 0.4069V$ mentre quella della simulazione è $\overline{V_{C2,sim}} = 0.4086V$. I due valori differiscono solo del 0.42% quindi tale da garantire una buona relazione tra il modello calcolato e quello simulato. Il modello teorico usato in MatLab per il calcolo della tensione V_{C2} a regime è:

$$V_{C2} = V_{in} - \frac{R_{sw-\Phi_1} \cdot V_{out}}{R_{load} \cdot D}$$

anche quest'ultima espressione è stata ricavata dal modello descritto in 2.1.7.

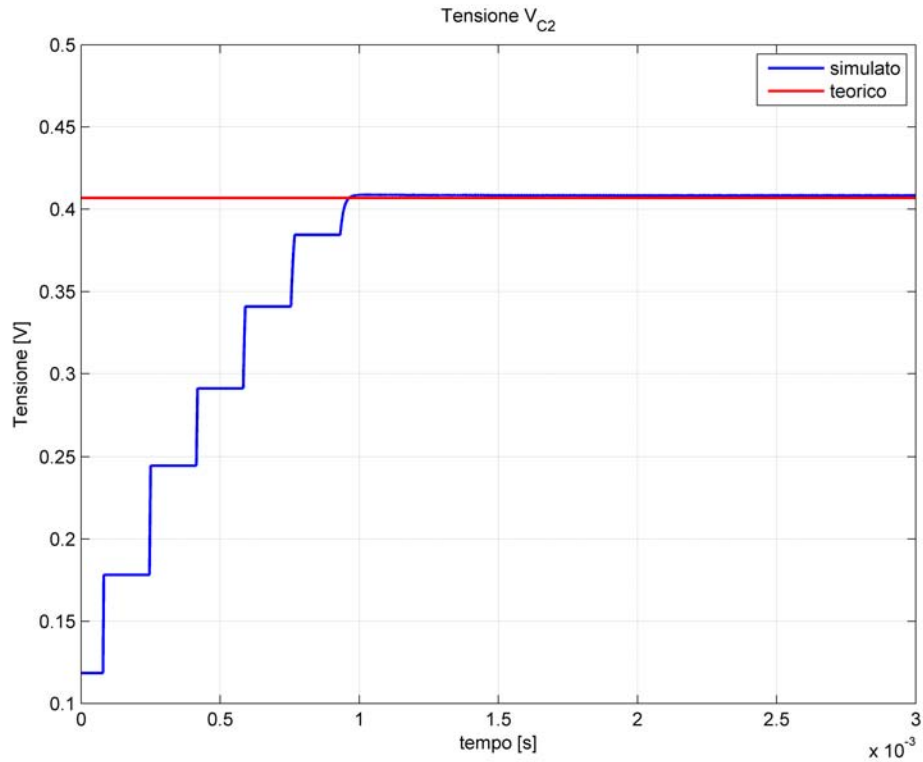


Figura 3.15: Andamento della tensione sul condensatore C_2 .

Una volta verificato che il funzionamento della parte logica è corretto e che il modello teorico con le relative perdite coincide con quello della simulazione si può andare ad osservare in dettaglio come varia la tensione ai capi dei condensatori durante le varie fasi. Ricordando la figura 2.3 ci si può aspettare il seguente comportamento:

- fase Φ_1 :
 - La tensione di uscita sale poichè il condensatore di uscita C_{out} si carica;
 - Il condensatore C_1 si sta scaricando in favore di C_{out} e quindi la sua tensione cala;
 - Il condensatore C_2 è sganciato dal resto del circuito e quindi la sua tensione resta costante;
- fase Φ_2 :

- La tensione di uscita scende poichè il condensatore di uscita si sta scaricando a causa del carico R_{load} ;
- Il condensatore C_1 si sta caricando quindi la sua tensione sale;
- Il condensatore C_2 si sta scaricando;
- fase Φ_3 :
 - La tensione di uscita scende poichè il condensatore di uscita si sta scaricando a causa del carico R_{load} ;
 - Il condensatore C_1 è lasciato flottante e quindi la sua tensione rimane costante;
 - Il condensatore C_2 si sta caricando quindi la sua tensione sale;

La figura 3.16 mostra esattamente quanto appena specificato.

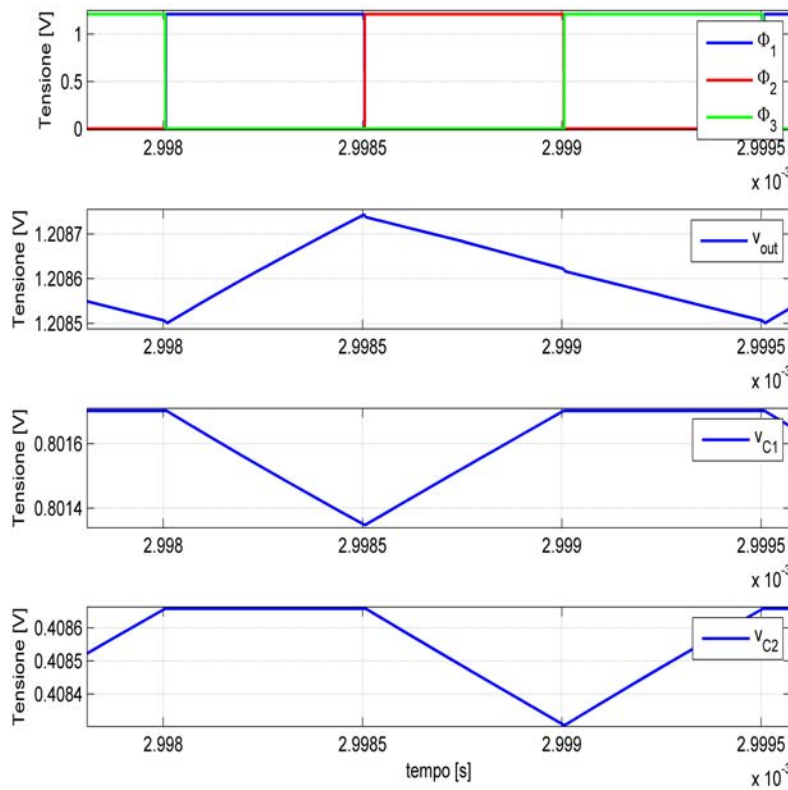


Figura 3.16: Andamento della tensioni.

Anche in questo caso, i picchi della tensione di uscita sono causati dal fenomeno del *charge injection* provocato dal transistor Sw_1 .

L'ultimo controllo da effettuare per confermare la correttezza dello schematico e del suo funzionamento è stato verificare se, con il dimensionamento ottenuto (dei condensatori e dei transistor), il convertitore lavora effettivamente nella condizione di fast switching mode. Come nel caso precedente, se la tensione ai capi del condensatore è lineare a tratti e se corrente è costante durante l'intera fase allora si può confermare il corretto regime di funzionamento. Le figure 3.17 e 3.18 mostrano proprio l'andamento descritto e quindi si può confermare che il convertitore lavora in fast switching mode.

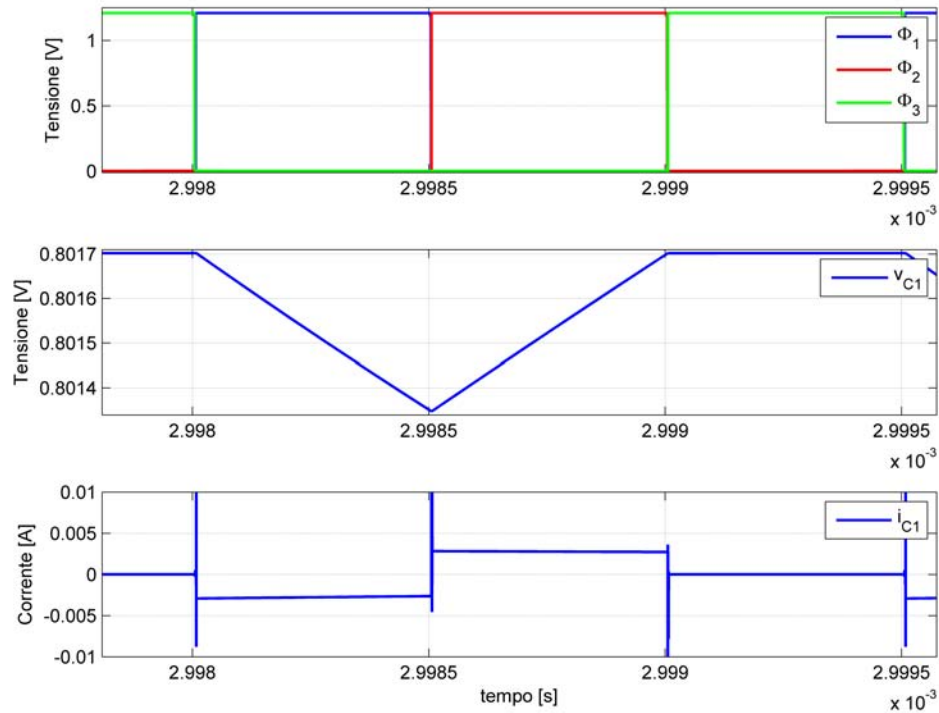


Figura 3.17: Verifica del funzionamento in FSL di C_1 .

Gli impulsi di corrente che si vedono in entrambe le immagini sono da attribuirsi ai transistori tra il cambio delle fasi.

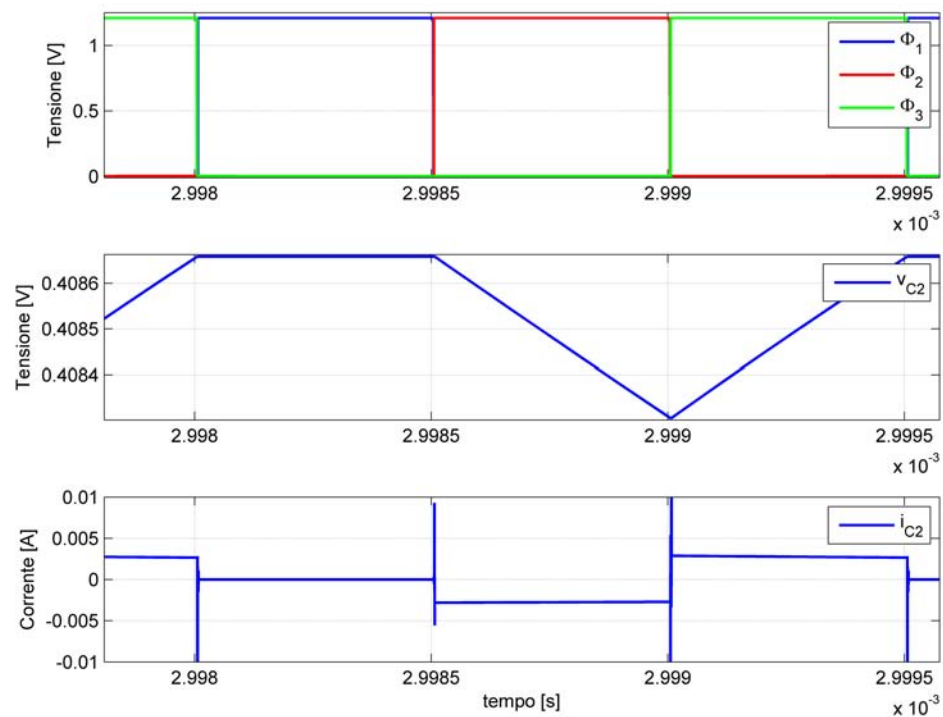


Figura 3.18: Verifica del funzionamento in FSL di C_2 .

Convertitore 8/3

Le figure 3.19, 3.20, 3.21 e 3.23 mostrano la sequenza di accensione degli switch in base alla fase:

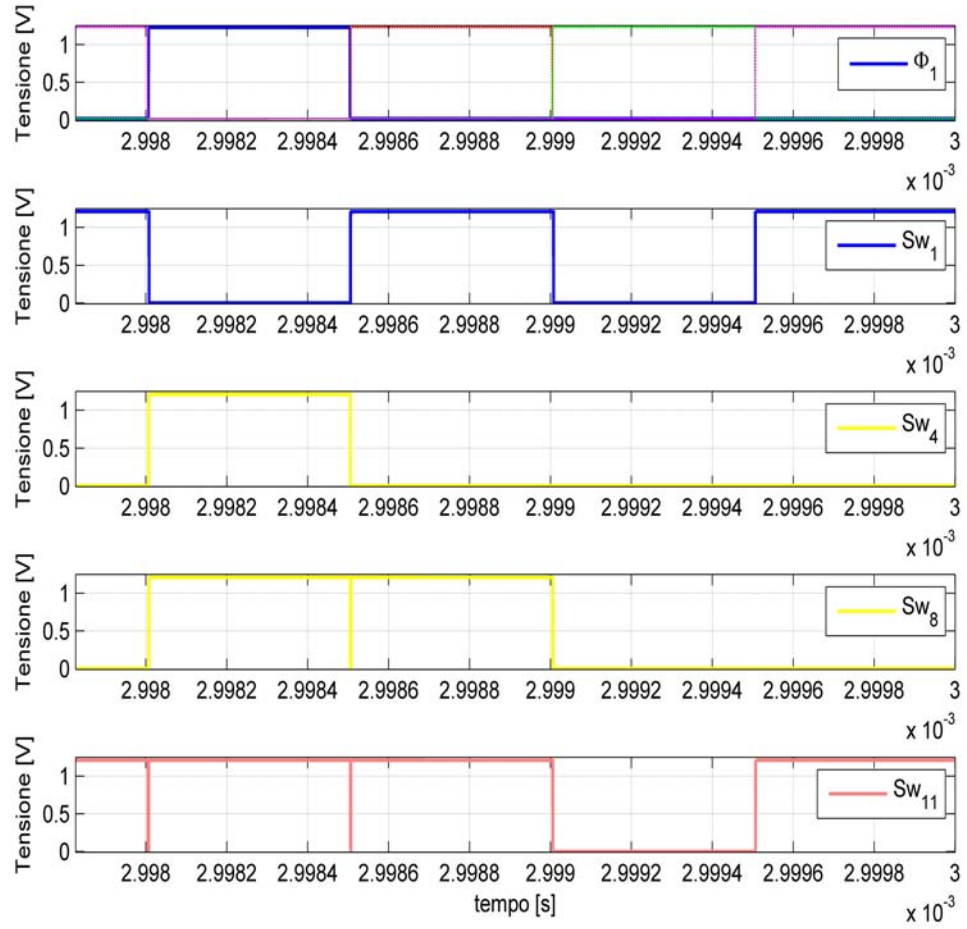


Figura 3.19: Ordine di accensione per la fase Φ_1 .

La tabella 2.16 mostra che nella fase Φ_1 vengono accesi gli switch: Sw_1 , Sw_4 , Sw_8 e Sw_{11} e la figura 3.19 lo conferma. La stessa tabella mostra inoltre che nella fase Φ_2 devo accendersi i transistor: Sw_2 , Sw_3 , Sw_8 e Sw_{11} e questo viene confermato dall'immagine 3.20.

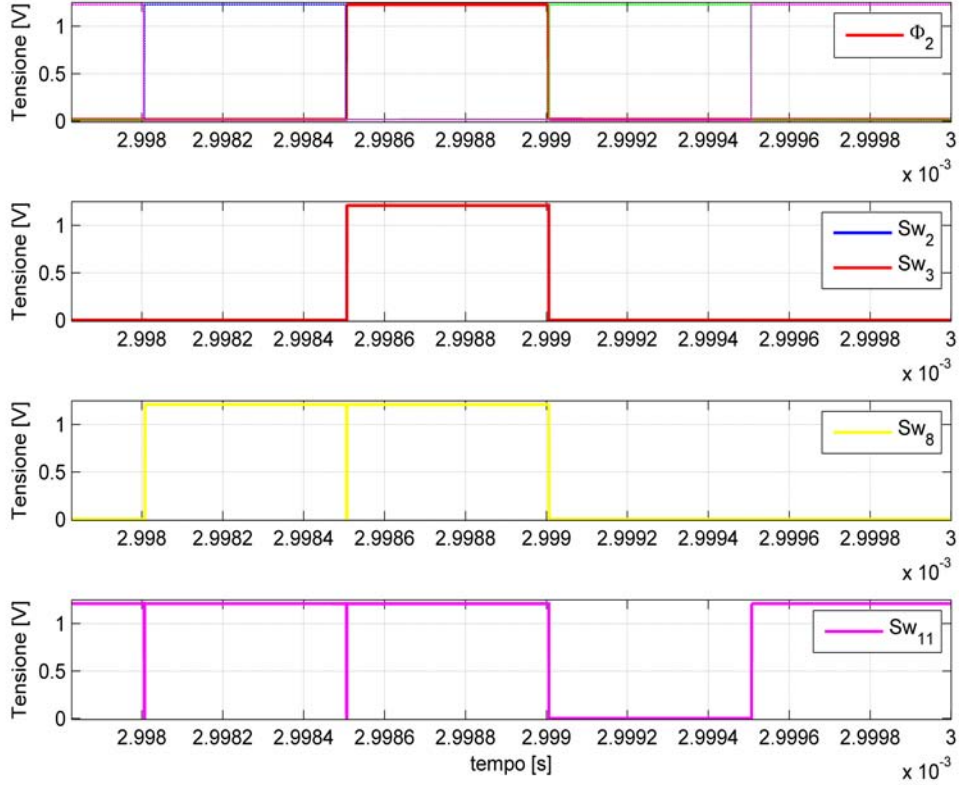


Figura 3.20: Ordine di accensione per la fase Φ_2 .

Gli switch non riportati in figura risultano spenti nella relativa fase. Per quanto riguarda la fase Φ_3 , l'ordine di accensione dei transistor dovrebbe essere: Sw_1 , Sw_7 e Sw_{10} . Nella fase Φ_4 invece dovrebbe veder coinvolti i transistor: Sw_6 , Sw_9 e Sw_{11} .

Nella parte introduttiva si era accennato al fatto che le fasi di qualsiasi convertitore non avrebbero dovuto sovrapporsi, altrimenti si corre il rischio che i transistor in fase di spegnimento e quelli in fase di accensione cortocircuitino i condensatori C_{fly} scaricandoli. La figura 3.22 è un ingrandimento tra la fase Φ_1 e la fase Φ_2 . Si nota molto bene che non sono sovrapposte e che l'intervallo tra le due fasi è di $1.1ns$ circa.

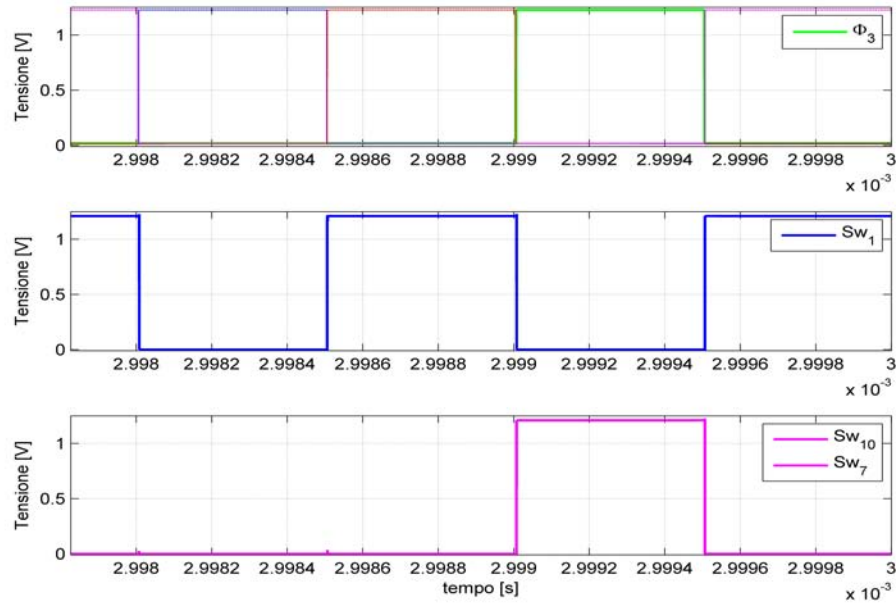


Figura 3.21: Ordine di accensione per la fase Φ_3 .

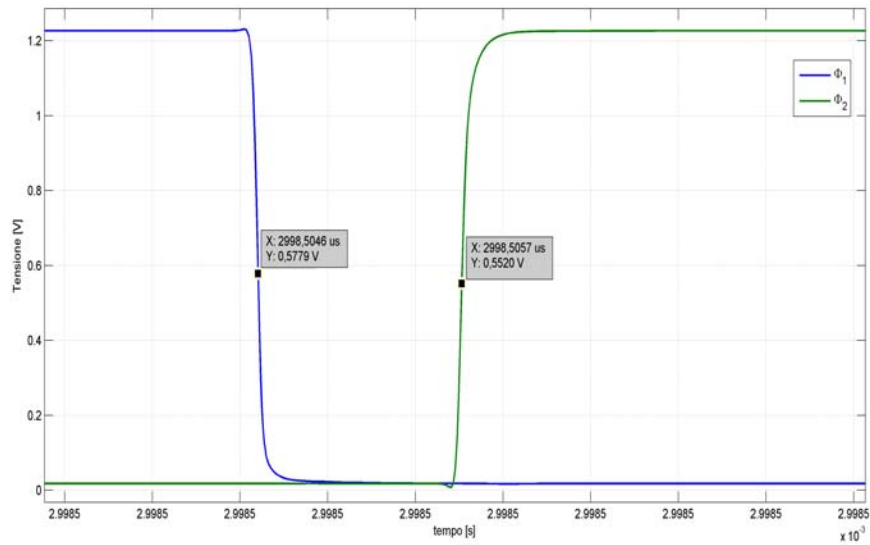


Figura 3.22: Zoom sulla non sovrapposizione delle fasi.

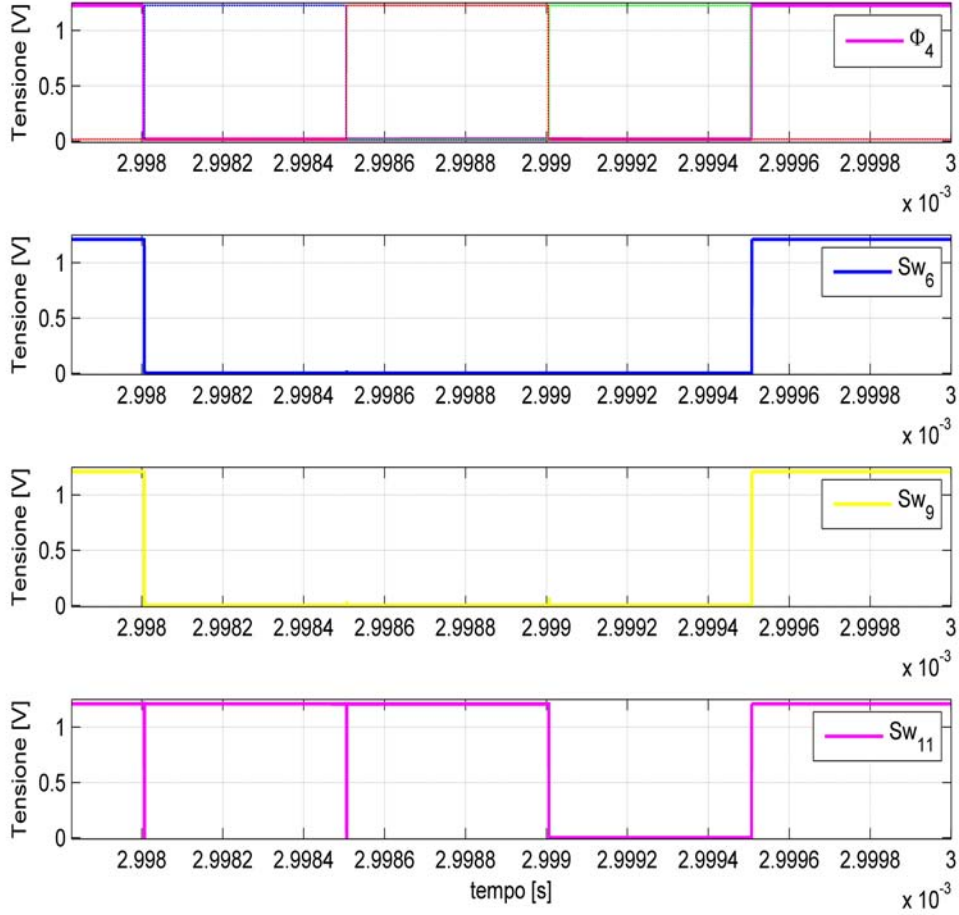


Figura 3.23: Ordine di accensione per la fase Φ_4 .

Anche in questo caso si può fare la stessa correlazione tra durata di accensione di un transistor e la sua resistenza serie calcolata in precedenza.

La figura 3.24 invece mostra l'andamento della tensione di uscita: il tratto iniziale è sempre dovuto all'azione del controllore. La cosa importante da notare è che l'andamento teorico (con le perdite resistive degli switch) prevedeva una tensione a regime di $V_{out,teorico} = 1.2270V$ (con una tensione di ingresso di $V_{in} = 0.466V$), mentre i dati di simulazione mostrano che la tensione di uscita media a regime vale $\overline{V_{out,sim}} = 1.2094V$. La previsione

teorica è corretta al:

$$\frac{\overline{V_{out,sim}}}{V_{out,teorico}} = 92.24\%$$

ciò significa che il modello analizzato e simulato in MatLab è abbastanza corretto, ma probabilmente questo convertitore è quello che ne ha risentito di più di tutti del dimensionamento comune. È per questo motivo il modello teorico risulta più distante da quello simulato rispetto agli altri quattro convertitori.

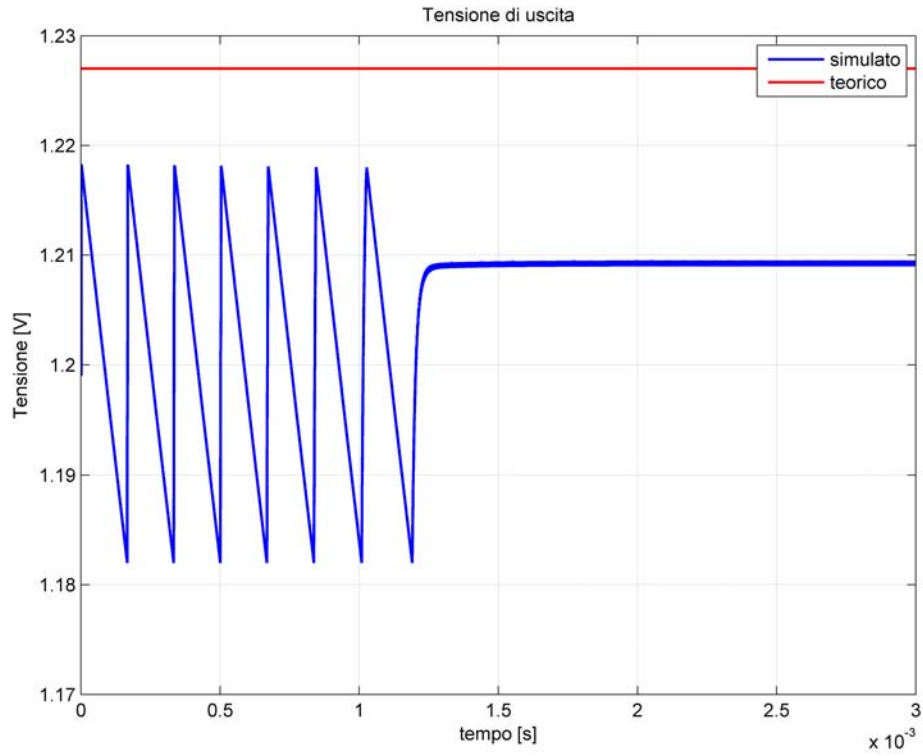


Figura 3.24: Andamento della tensione di uscita.

Il modello teorico che si era calcolato per la tensione di uscita è:

$$V_{out} = \frac{8}{3}V_{in} - 2 \cdot \frac{V_{out}}{9 \cdot R_{load}} \cdot (R_{sw-\Phi_1} + 7R_{sw-\Phi_2} + 16R_{sw-\Phi_3} + 8R_{sw-\Phi_4})$$

Il rapporto di conversione è: $\frac{1.2094}{0.466} = 2.595$.

L'immagine 3.25 mostra l'andamento della tensione sul condensatore C_1 e si

può osservare che il valore teorico e quello simulato sono abbastanza vicini come valori: infatti essendo rispettivamente: $V_{C1,teorico} = 0.6087V$ e $\overline{V_{C1,sim}} = 0.5962V$ si ha uno scarto che non supera il 2%. Il modello teorico usato in MatLab per il calcolo della tensione V_{C1} a regime è:

$$V_{C1} = \frac{4}{3}V_{in} + 2 \cdot \frac{V_{out}}{9 \cdot R_{load}} \cdot (2R_{sw-\Phi_1} + 9R_{sw-\Phi_2} + 8R_{sw-\Phi_3} + 2R_{sw-\Phi_4})$$

e si ricava facilmente dal modello della tensione di uscita e sostituendolo nel sistema di partenza.

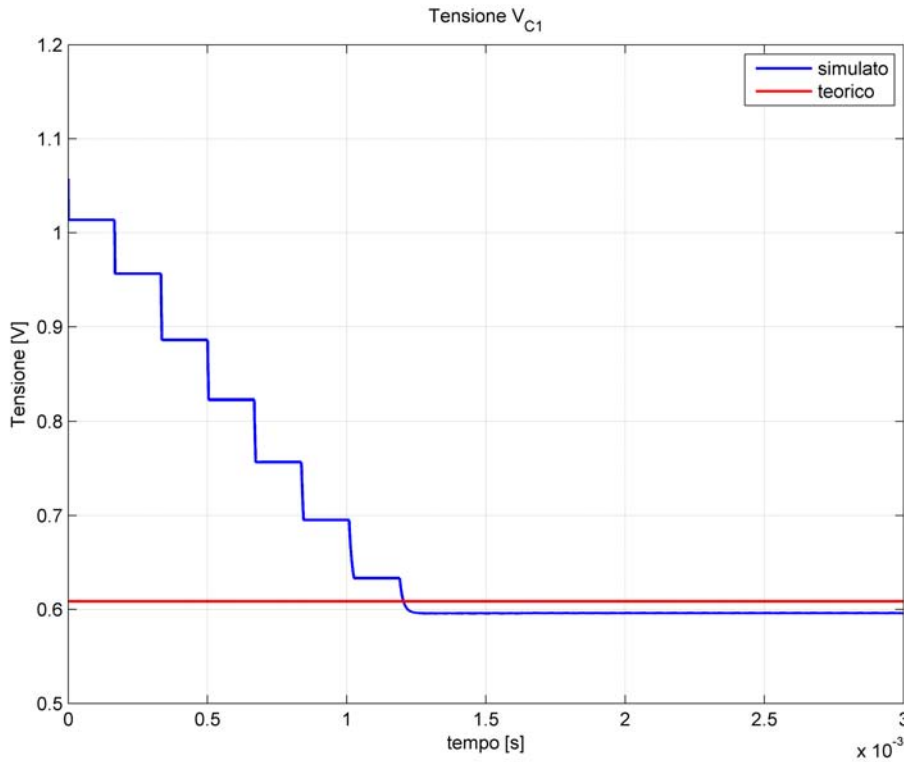


Figura 3.25: Andamento della tensione sul condensatore C_1 .

La figura 3.26 mostra come il modello teorico predice con una buona precisione la simulazione. La tensione di uscita del modello matematico è $V_{C2,teorico} = 0.3065V$ mentre quella della simulazione è $\overline{V_{C2,sim}} = 0.3022V$. I due valori differiscono solo coincidono al:

$$\frac{\overline{V_{C2,sim}}}{V_{C2,teorico}} = 98.59\%$$

quindi tale da garantire una buona relazione tra il modello calcolato e quello simulato. Il modello teorico usato in MatLab per il calcolo della tensione V_{C2} a regime è:

$$V_{C2} = \frac{2}{3}V_{in} + 2 \cdot \frac{V_{out}}{9 \cdot R_{load}} \cdot (R_{sw-\Phi_1} + 4R_{sw-\Phi_3} + 4R_{sw-\Phi_4})$$

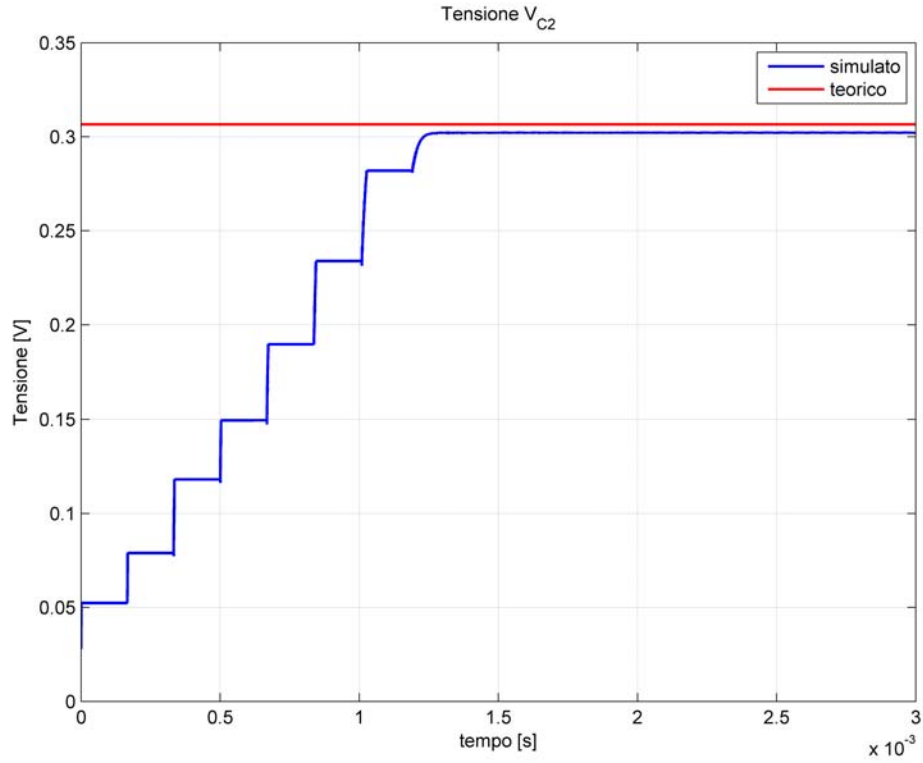


Figura 3.26: Andamento della tensione sul condensatore C_2 .

La figura 3.27 mostra come anche in questo caso il modello teorico predice con una buona precisione la simulazione. La tensione di uscita del modello matematico è $V_{C3,teorico} = 0.1562V$ mentre quella della simulazione è $\overline{V_{C3,sim}} = 0.1593V$. I due valori differiscono solo del 1.94% quindi tale da garantire una buona relazione tra il modello calcolato e quello simulato. Il modello teorico usato in MatLab per il calcolo della tensione V_{C3} a regime è:

$$V_{C3} = \frac{1}{3}V_{in} - 2 \cdot \frac{V_{out}}{9 \cdot R_{load}} \cdot (R_{sw-\Phi_1} - 2R_{sw-\Phi_2} + 4R_{sw-\Phi_3})$$

Come per tutte le altre espressioni delle tensioni dei condensatori, anche quest'ultima è stata ricavata a partire dal sistema del paragrafo 2.1.7.

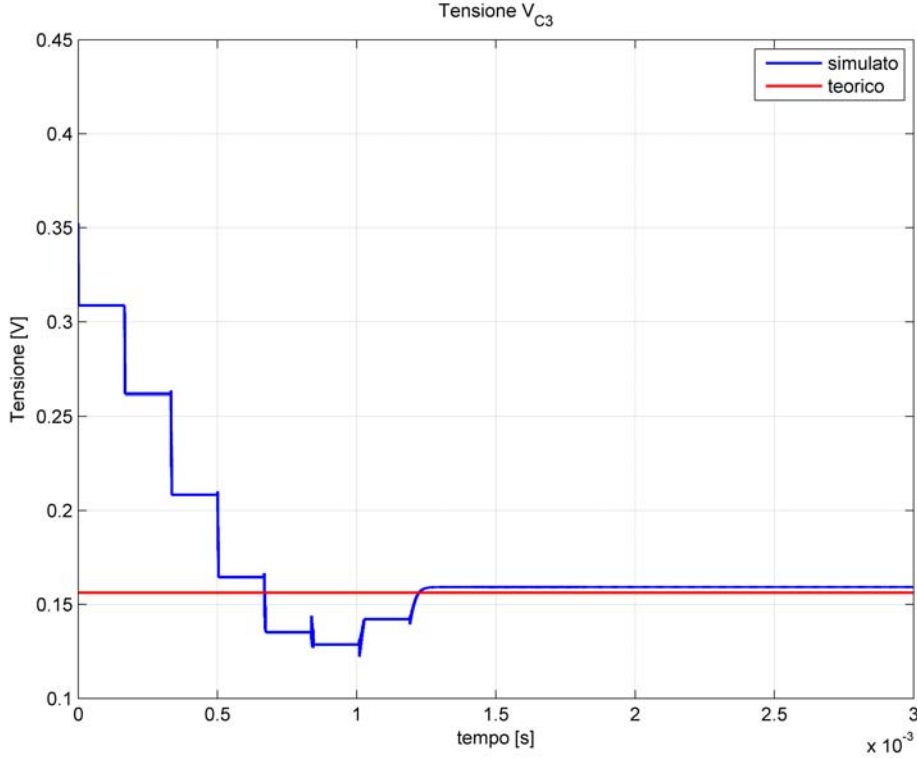


Figura 3.27: Andamento della tensione sul condensatore C_3 .

Una volta verificato che il funzionamento della parte logica è corretto e che il modello teorico con le relative perdite coincide con quello della simulazione si può andare ad osservare in dettaglio come varia la tensione ai capi dei condensatori durante le varie fasi. Ricordando la figura 2.4 ci si può aspettare il seguente comportamento:

- fase Φ_1 :
 - La tensione di uscita sale poichè il condensatore di uscita C_{out} si carica;

- Il condensatore C_1 si sta scaricando in favore di C_{out} e quindi la sua tensione cala;
 - Il condensatore C_2 si sta scaricando in favore di C_{out} e quindi la sua tensione cala;
 - Il condensatore C_3 si sta caricando e di conseguenza la sua tensione sale;
- fase Φ_2 :
 - La tensione di uscita scende poichè il condensatore di uscita si sta scaricando a causa del carico R_{load} ;
 - Il condensatore C_1 si sta caricando quindi la sua tensione sale;
 - Il condensatore C_2 si sta scaricando;
 - Il condensatore C_3 si sta caricando quindi la sua tensione sale;
- fase Φ_3 :
 - La tensione di uscita sale poichè il condensatore di uscita C_{out} si carica;
 - Il condensatore C_1 si sta scaricando in favore di C_{out} e quindi la sua tensione cala;
 - Il condensatore C_2 è sganciato dal resto del circuito, di conseguenza la tensione ai suoi capi resta costante;
 - Il condensatore C_3 si sta scaricando in favore di C_{out} e quindi la sua tensione cala;
- fase Φ_4 :
 - La tensione di uscita scende poichè il condensatore di uscita si sta scaricando a causa del carico R_{load} ;
 - Il condensatore C_1 è sganciato dal resto del circuito, di conseguenza la tensione ai suoi capi resta costante;
 - Il condensatore C_2 si sta caricando quindi la sua tensione sale;
 - Il condensatore C_3 si sta caricando quindi la sua tensione sale;

La figura 3.28 mostra esattamente quanto appena specificato.

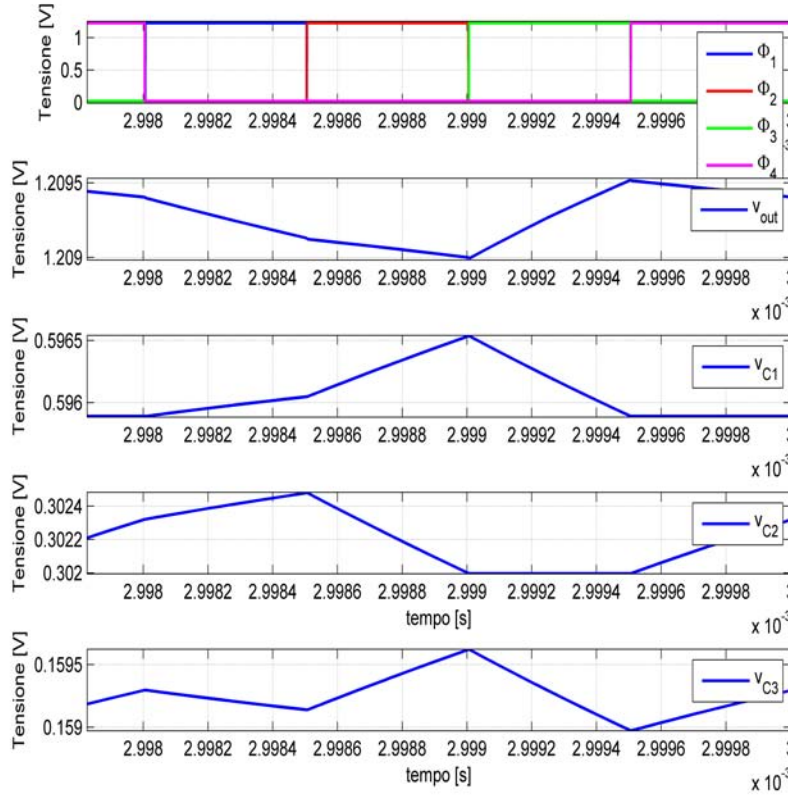


Figura 3.28: Andamento della tensioni.

Anche in questo caso, i picchi della tensione di uscita sono causati dal fenomeno del *charge injection* provocato dal transistor Sw_1 .

L'ultimo controllo da effettuare per confermare la correttezza dello schematico e del suo funzionamento è stato verificare se, con il dimensionamento ottenuto (dei condensatori e dei transistor), il convertitore lavora effettivamente nella condizione di fast switching mode. Come nel caso precedente, se la tensione ai capi del condensatore è lineare a tratti e se corrente è costante durante l'intera fase allora si può confermare il corretto regime di funzionamento. Le figure 3.29, 3.30 e 3.31 mostrano proprio l'andamento descritto e quindi si può confermare che il convertitore lavora in fast switching mode.

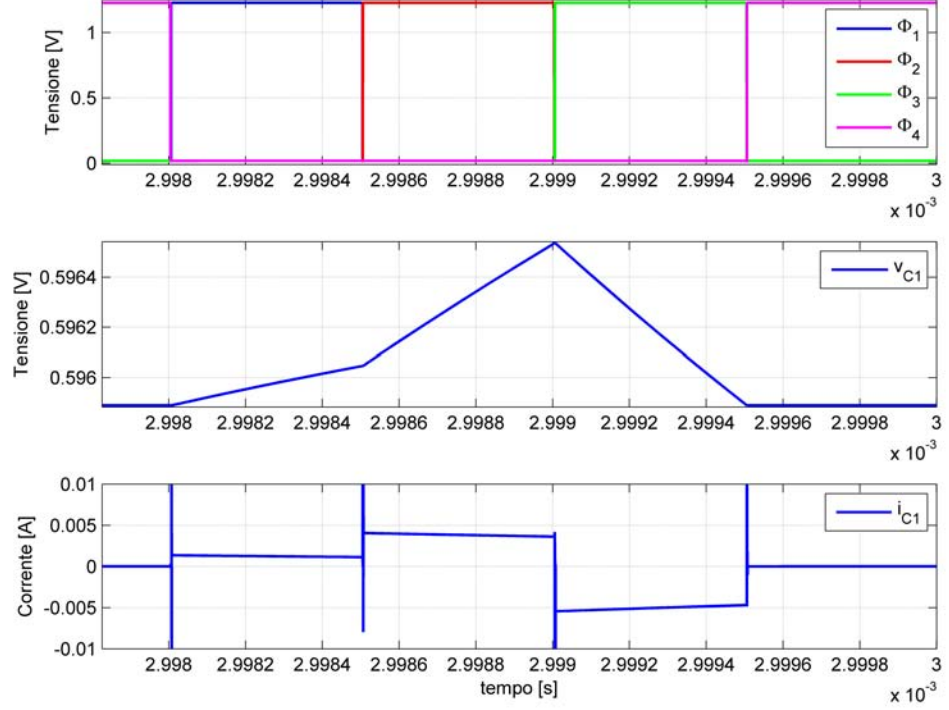


Figura 3.29: Verifica del funzionamento in FSL di C_1 .

Gli impulsi di corrente che si vedono in entrambe le immagini sono da attribuirsi ai transistori tra il cambio delle fasi.

Nonostante si sia osservato il comportamento in FSL, la corrente del condensatore C_2 in Φ_2 e la corrente del condensatore C_1 e C_3 nella fase Φ_3 mostrano una lieve pendenza. Questo sta a significare che in questa condizione operativa, il convertitore lavora appena dentro la regione di fast switching (subito dopo la curva di raccordo tra le rette di figura 1.6). Se si volesse togliere questo effetto basterebbe aumentare la capacità dei condensatori C_{fly} , oppure aumentare la frequenza del generatore di clock, ma questo effetto porterebbe ad un consumo maggiore di potenza da parte della circuiteria di controllo.

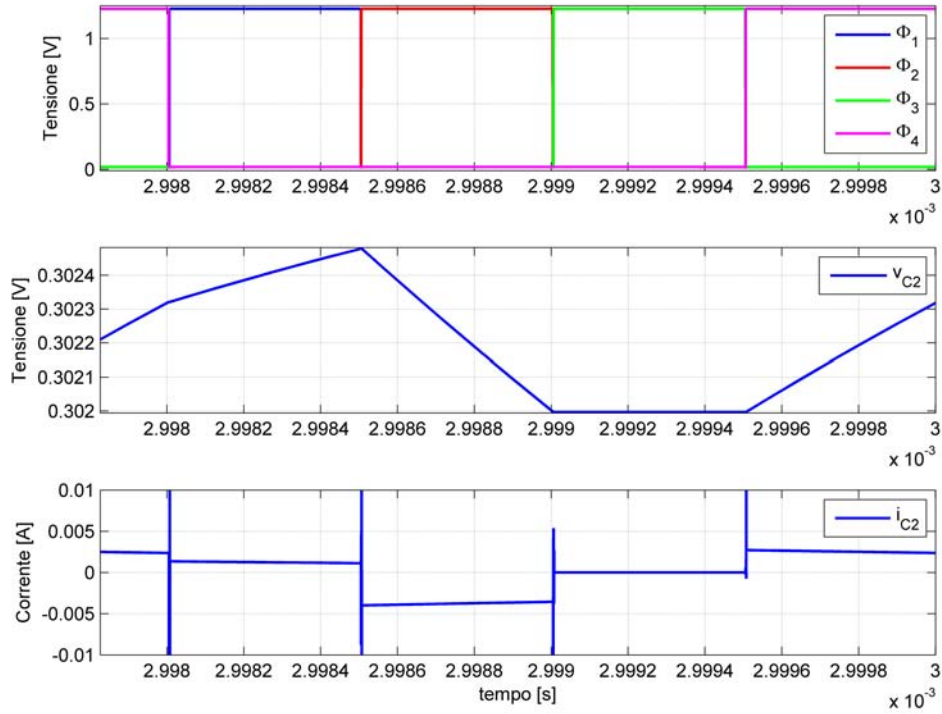


Figura 3.30: Verifica del funzionamento in FSL di C_2 .

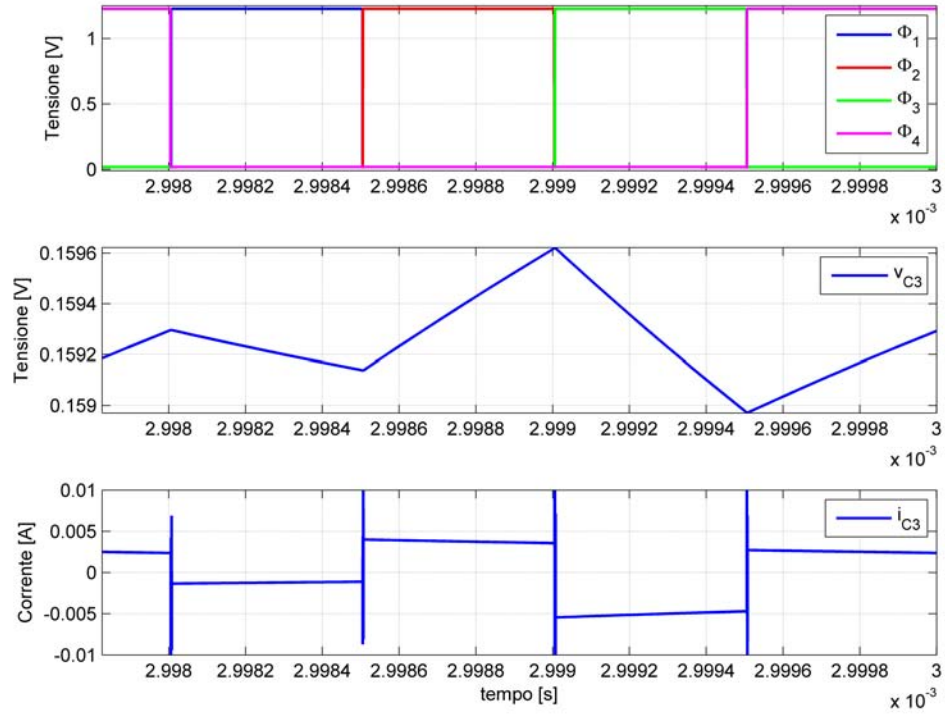


Figura 3.31: Verifica del funzionamento in FSL di C_3 .

Convertitore 5/2

Le figure 3.32, 3.33, 3.34 e 3.35 mostrano la sequenza di accensione degli switch durante le quattro fasi di questo convertitore:

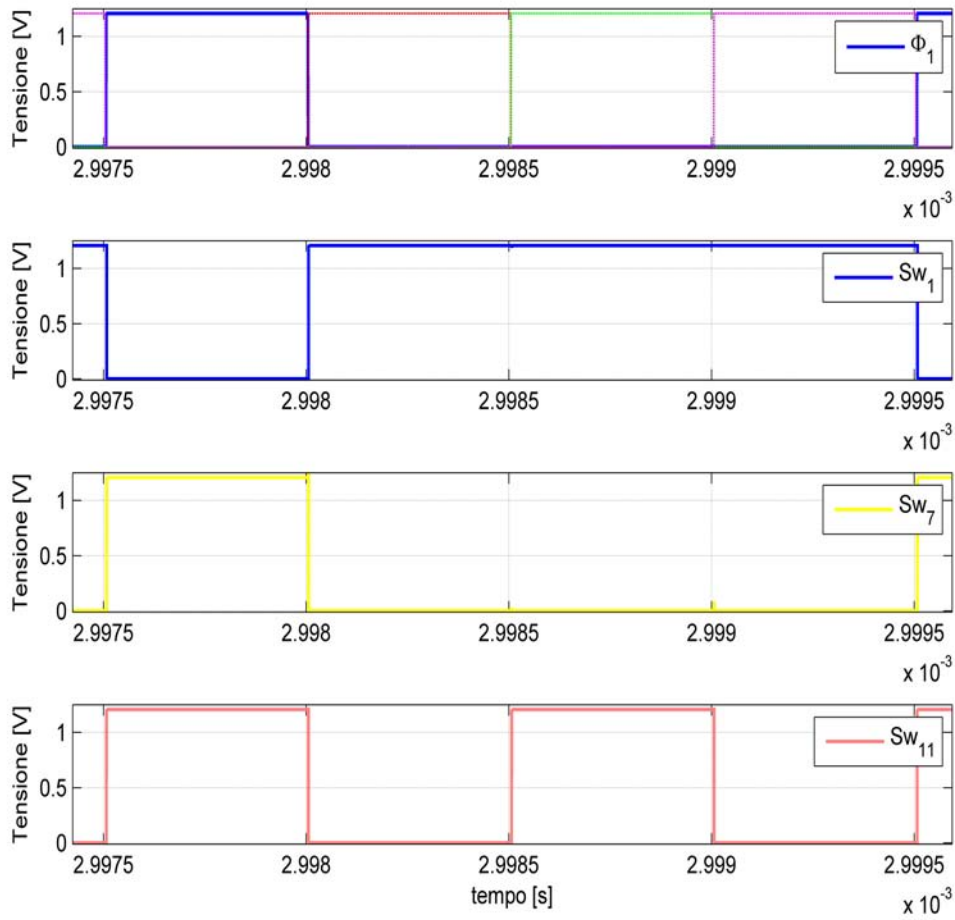


Figura 3.32: Ordine di accensione per la fase Φ_1 .

La tabella 2.16 mostra che nella fase Φ_1 vengono accesi gli switch: Sw_1 , Sw_7 e Sw_{11} e la figura 3.32 lo conferma. La stessa tabella mostra inoltre che nella fase Φ_2 devo accendersi i transistor: Sw_6 , Sw_9 e Sw_{10} e questo viene confermato dall'immagine 3.33.

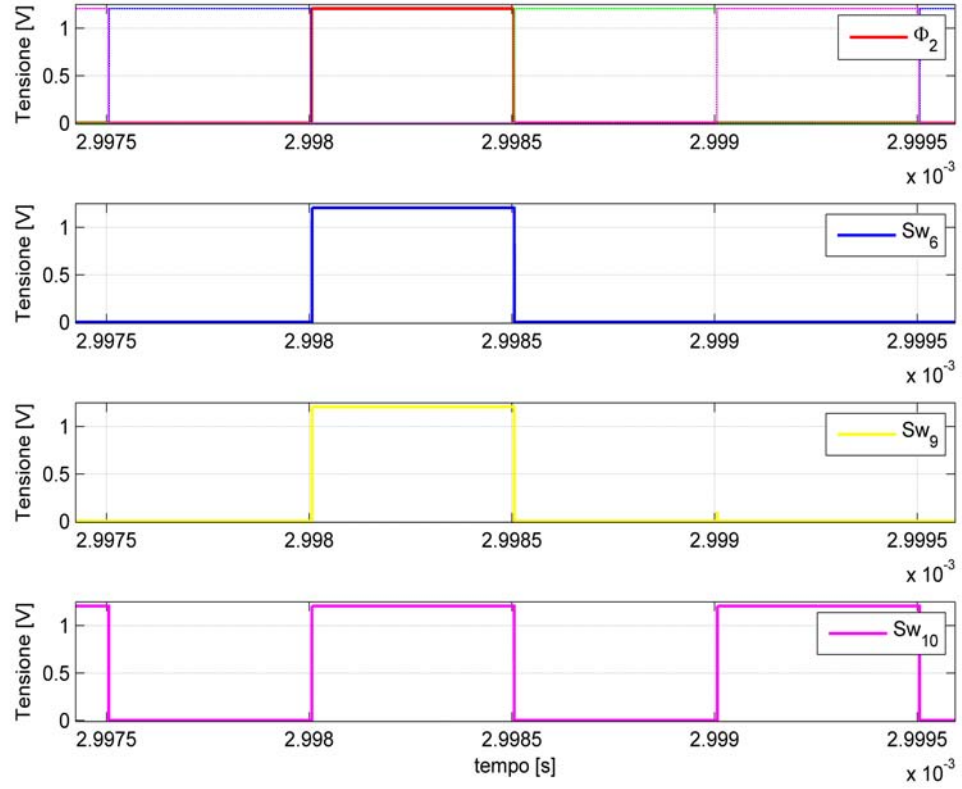
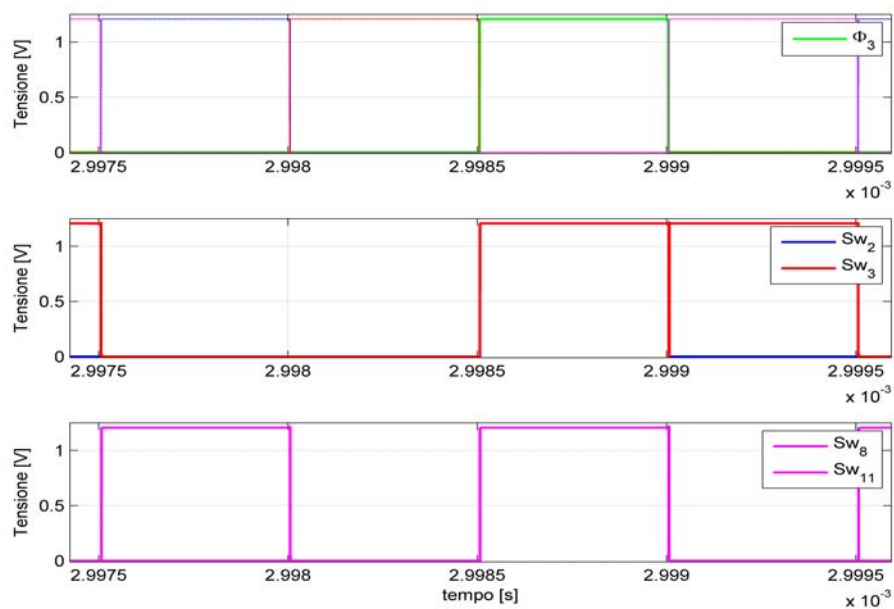


Figura 3.33: Ordine di accensione per la fase Φ_2 .

Gli switch non riportati in figura risultano spenti nella relativa fase. Per quanto riguarda la fase Φ_3 , l'ordine di accensione dei transistor dovrebbe essere: Sw_2 , Sw_3 , Sw_8 e Sw_{11} . Nella fase Φ_4 invece dovrebbe veder coinvolti i transistor: Sw_3 , Sw_5 e Sw_{10} .

Figura 3.34: Ordine di accensione per la fase Φ_3 .

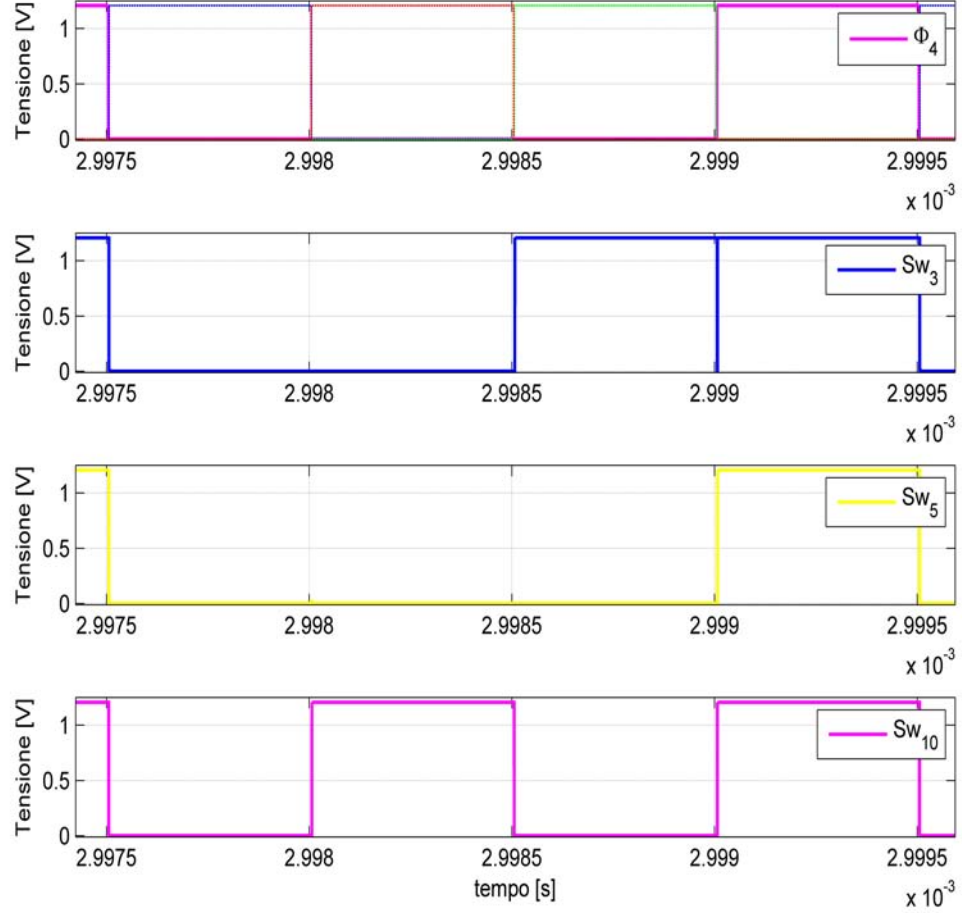


Figura 3.35: Ordine di accensione per la fase Φ_4 .

Anche in questo caso si può fare la stessa correlazione tra durata di accensione di un transistor e la sua resistenza serie calcolata in precedenza.

La figura 3.36 invece mostra l'andamento della tensione di uscita: il tratto iniziale è sempre dovuto all'azione del controllore. La cosa importante da notare è che l'andamento teorico (con le perdite resistive degli switch) prevedeva una tensione a regime di $V_{out,teorico} = 1.2070V$ (con una tensione di ingresso di $V_{in} = 0.491V$), mentre i dati di simulazione mostrano che la tensione di uscita media a regime vale $\overline{V_{out,sim}} = 1.2058V$. La previsione

teorica è corretta al:

$$\frac{\overline{V_{out,sim}}}{V_{out,teorico}} = 99.42\%$$

ciò significa che il modello analizzato e simulato in MatLab sono pressochè equivalenti.

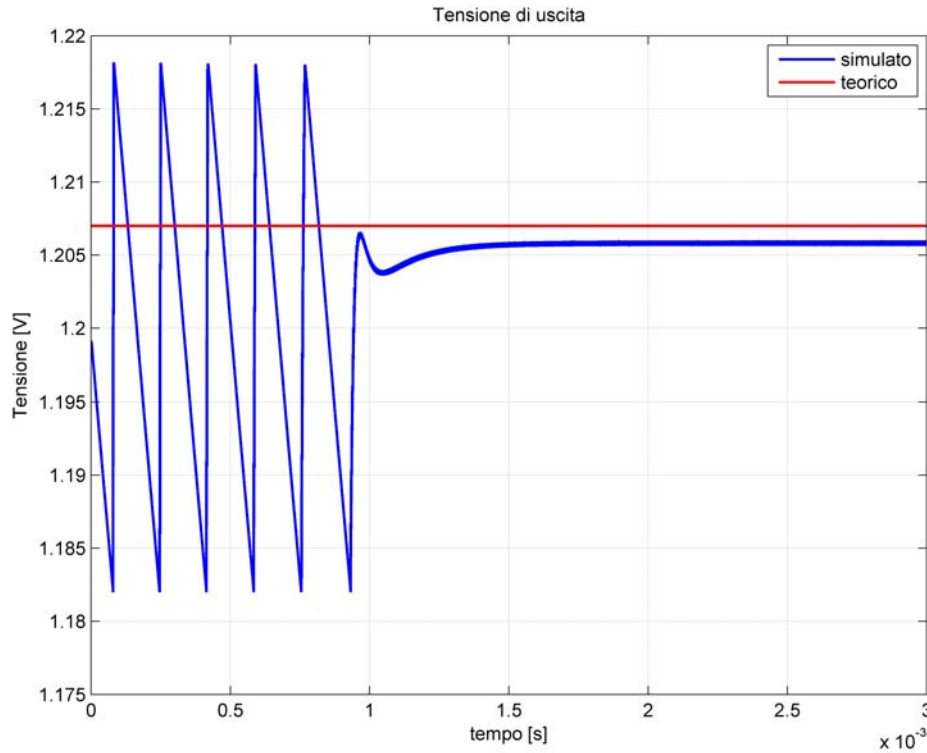


Figura 3.36: Andamento della tensione di uscita.

Il modello teorico che si era calcolato per la tensione di uscita è:

$$V_{out} = \frac{5}{2}V_{in} - \frac{V_{out}}{R_{load}} \cdot (R_{sw-\Phi_1} + R_{sw-\Phi_2} + R_{sw-\Phi_3} + 4R_{sw-\Phi_4})$$

Il rapporto di conversione è: $\frac{1.2058}{0.491} = 2.456$.

L'immagine 3.37 mostra l'andamento della tensione sul condensatore C_1 e si può osservare che il valore teorico e quello simulato sono abbastanza vicini come valori: infatti essendo rispettivamente: $V_{C1,teorico} = 0.7296V$ e $\overline{V_{C1,sim}} =$

0.7236V si ha uno scarto che non supera il 0.83%. Il modello teorico usato in MatLab per il calcolo della tensione V_{C1} a regime è:

$$V_{C1} = \frac{3}{2}V_{in} - 4 \cdot \frac{V_{out}}{R_{load}} \cdot (R_{sw-\Phi_1} + R_{sw-\Phi_2} + R_{sw-\Phi_3})$$

e si ricava facilmente dal modello della tensione di uscita e sostituendolo nel sistema di partenza.

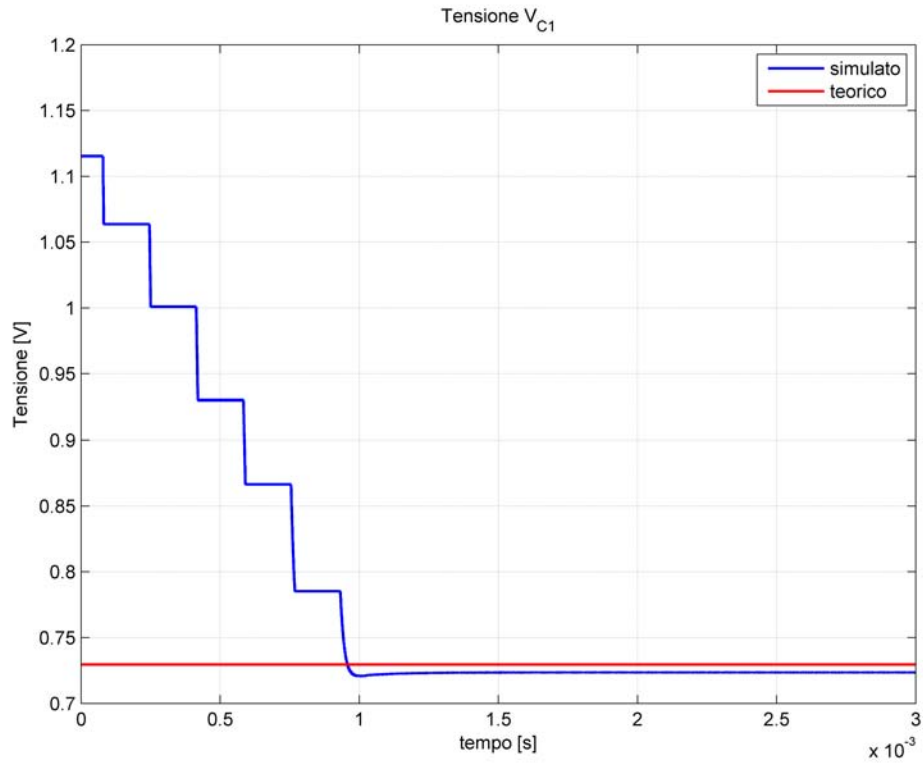


Figura 3.37: Andamento della tensione sul condensatore C_1 .

La figura 3.38 mostra come anche in questo caso il modello teorico predice con una buona precisione la simulazione. La tensione di uscita del modello matematico è $V_{C2,teorico} = 0.4818V$ mentre quella della simulazione è $\overline{V_{C2,sim}} = 0.4874V$. I due valori differiscono solo per lo 1.16% e quindi si può affermare con certezza che il modello teorico è in linea con quello simulato dal software. Il modello teorico usato in MatLab per il calcolo della tensione V_{C2} a regime è:

$$V_{C2} = V_{in} - 4 \cdot \frac{V_{out}}{R_{load}} \cdot R_{sw-\Phi_1}$$

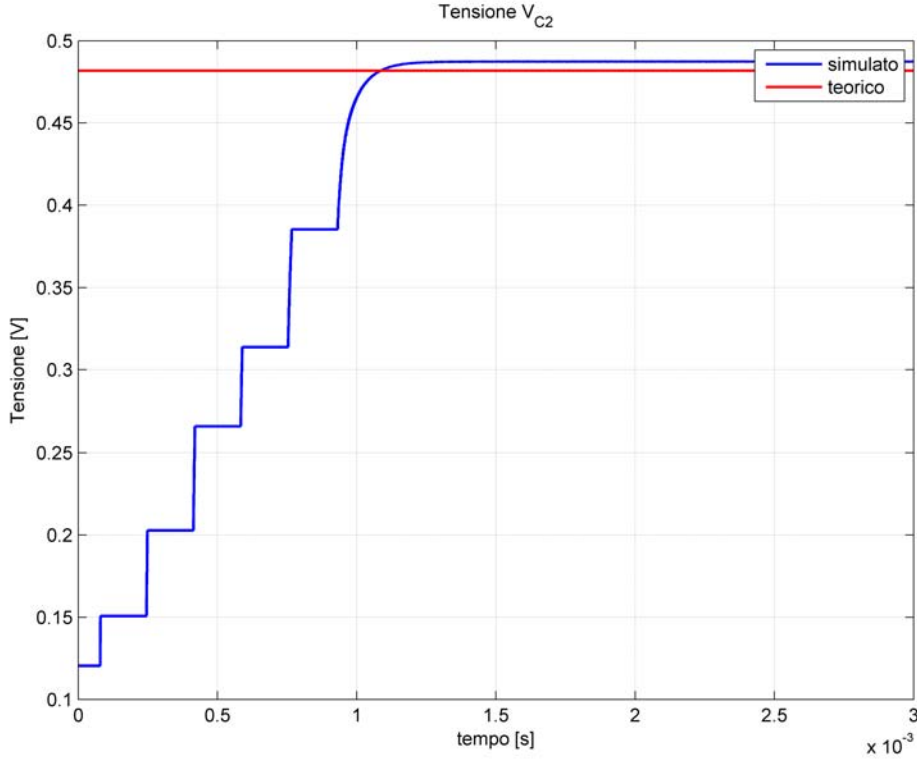


Figura 3.38: Andamento della tensione sul condensatore C_2 .

La figura 3.39 mostra come anche in questo caso il modello teorico predice con una buona precisione la simulazione. La tensione di uscita del modello matematico è $V_{C3,teorico} = 0.2447V$ mentre quella della simulazione è $V_{C3,sim} = 0.2462V$. I due valori differiscono solo del 0.6% quindi tale da garantire una buona relazione tra il modello calcolato e quello simulato. Il modello teorico usato in MatLab per il calcolo della tensione V_{C3} a regime è:

$$V_{C3} = V_{C1} - V_{in} + 2 \cdot \frac{V_{out}}{R_{load}} \cdot R_{sw-\Phi_3}$$

Come per tutte le altre espressioni delle tensioni dei condensatori, anche quest'ultima è stata ricavata a partire dal sistema del paragrafo 2.1.7.

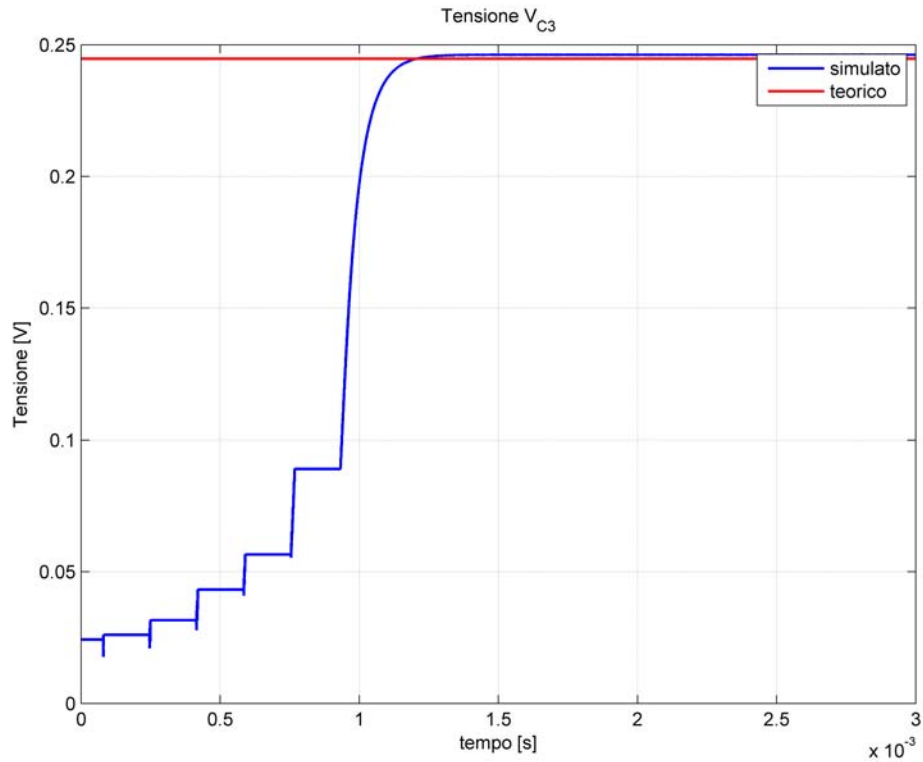


Figura 3.39: Andamento della tensione sul condensatore C_3 .

Una volta verificato che il funzionamento della parte logica è corretto e che il modello teorico con le relative perdite coincide con quello della simulazione si può andare ad osservare in dettaglio come varia la tensione ai capi dei condensatori durante le varie fasi. Ricordando la figura 2.5 ci si può aspettare il seguente comportamento:

- fase Φ_1 :
 - La tensione di uscita sale poichè il condensatore di uscita C_{out} si carica;
 - Il condensatore C_1 si sta scaricando in favore di C_{out} e quindi la sua tensione cala;
 - Il condensatore C_2 è sganciato dal resto del circuito, di conseguenza la tensione ai suoi capi resta costante;
 - Il condensatore C_3 è sganciato dal resto del circuito, di conseguenza la tensione ai suoi capi resta costante;

- fase Φ_2 :
 - La tensione di uscita scende poichè il condensatore di uscita si sta scaricando a causa del carico R_{load} ;
 - Il condensatore C_1 è sganciato dal resto del circuito, di conseguenza la tensione ai suoi capi resta costante;
 - Il condensatore C_2 si sta caricando e di conseguenza la sua tensione sale;
 - Il condensatore C_3 è sganciato dal resto del circuito, di conseguenza la tensione ai suoi capi resta costante;

- fase Φ_3 :
 - La tensione di uscita scende poichè il condensatore di uscita si sta scaricando a causa del carico R_{load} ;
 - Il condensatore C_1 si sta caricando quindi la sua tensione sale;
 - Il condensatore C_2 si sta scaricando in favore degli altri due condensatori;
 - Il condensatore C_3 si sta caricando quindi la sua tensione sale;

- fase Φ_4 :
 - La tensione di uscita scende poichè il condensatore di uscita si sta scaricando a causa del carico R_{load} ;
 - Il condensatore C_1 si sta caricando quindi la sua tensione sale;
 - Il condensatore C_2 è sganciato dal resto del circuito, di conseguenza la tensione ai suoi capi resta costante;
 - Il condensatore C_3 si sta scaricando in favore del condensatore C_1 ;

La figura 3.40 mostra esattamente quanto appena specificato.

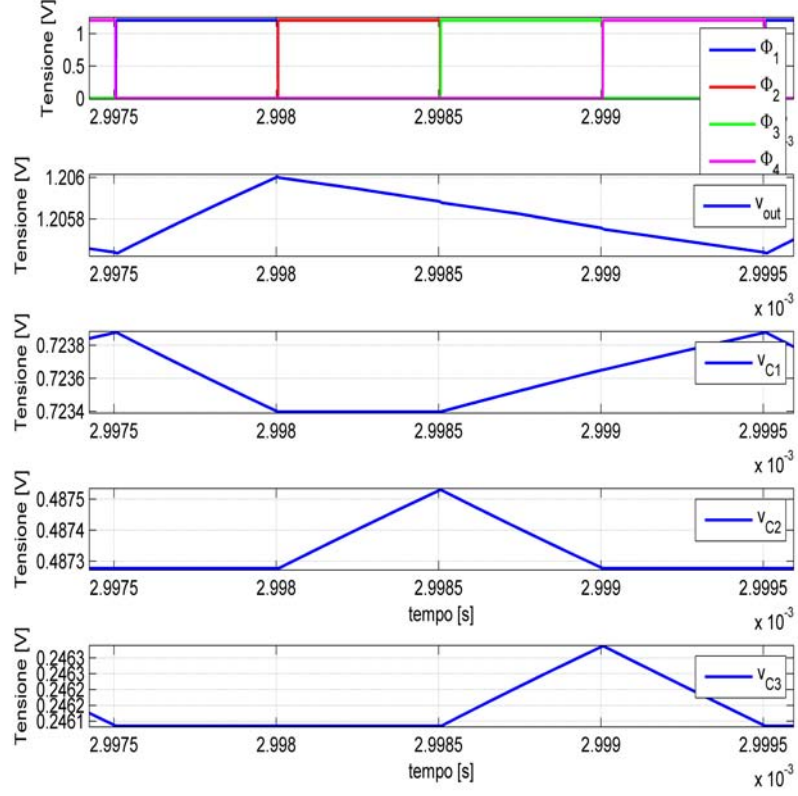


Figura 3.40: Andamento della tensioni.

Anche in questo caso, i picchi della tensione di uscita sono causati dal fenomeno del *charge injection* provocato dal transistor Sw_1 .

L'ultimo controllo da effettuare per confermare la correttezza dello schematico e del suo funzionamento è stato verificare se, con il dimensionamento ottenuto (dei condensatori e dei transistor), il convertitore lavora effettivamente nella condizione di fast switching mode. Come nel caso precedente, se la tensione ai capi del condensatore è lineare a tratti e se corrente è costante durante l'intera fase allora si può confermare il corretto regime di funzionamento. Le figure 3.41, 3.42 e 3.43 mostrano proprio l'andamento descritto e quindi si può confermare che il convertitore lavora in fast switching mode.

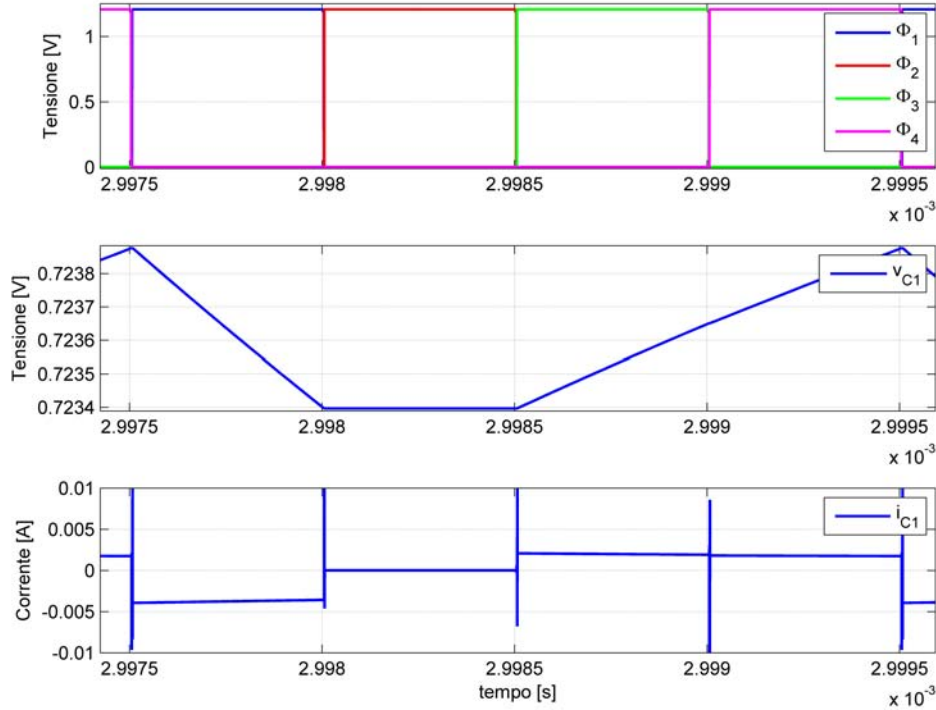


Figura 3.41: Verifica del funzionamento in FSL di C_1 .

Gli impulsi di corrente che si vedono in entrambe le immagini sono da attribuirsi ai transistori tra il cambio delle fasi.

Tuttavia anche in questo caso è possibile notare che il convertitore lavora al limite del fast switching mode. Infatti la corrente sul condensatore C_1 durante la prima fase risulta avere una leggera pendenza. Come spiegato in precedenza se si aumentano le dimensioni dei condensatori si riduce ancor di più questo effetto e non si va ad impattare sul consumo generale del sistema.

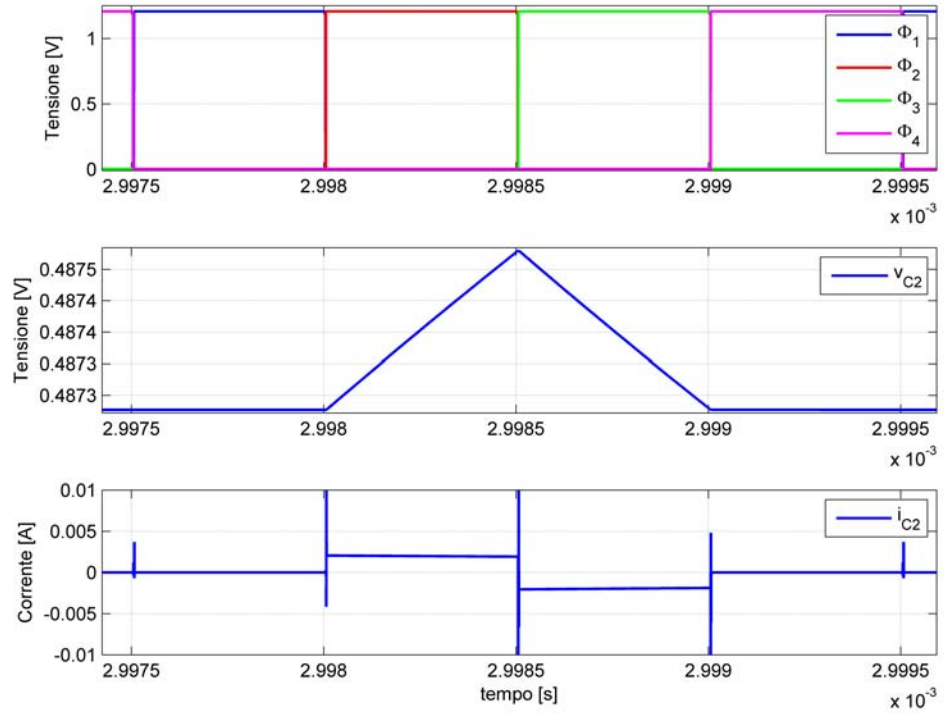


Figura 3.42: Verifica del funzionamento in FSL di C_2 .

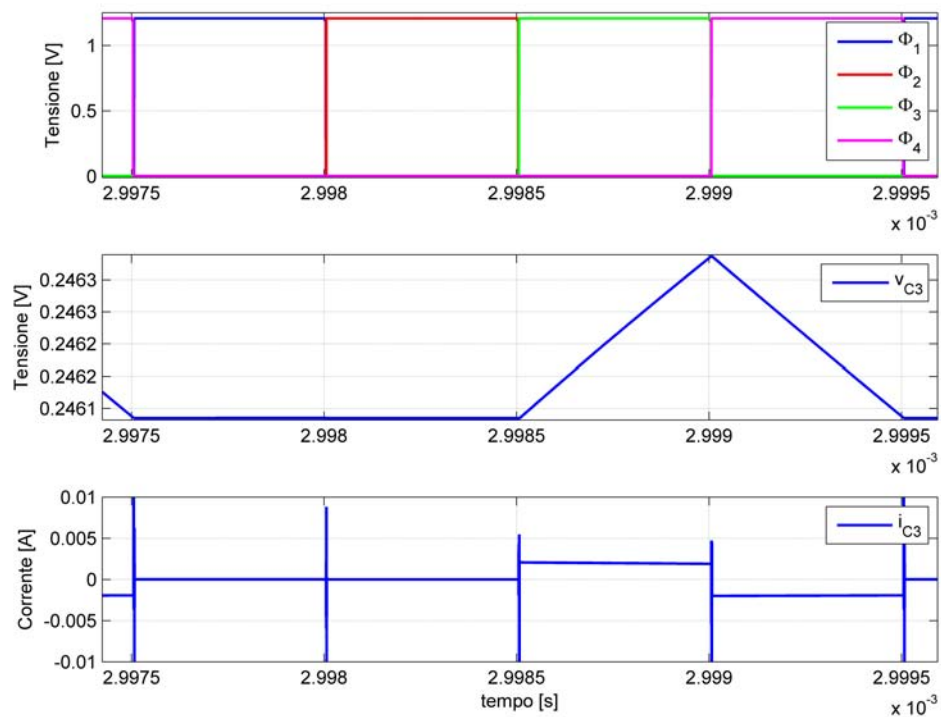


Figura 3.43: Verifica del funzionamento in FSL di C_3 .

Convertitore 8/4

Le figure 3.44, 3.45 mostrano la sequenza di accensione degli switch durante le quattro fasi di questo convertitore:

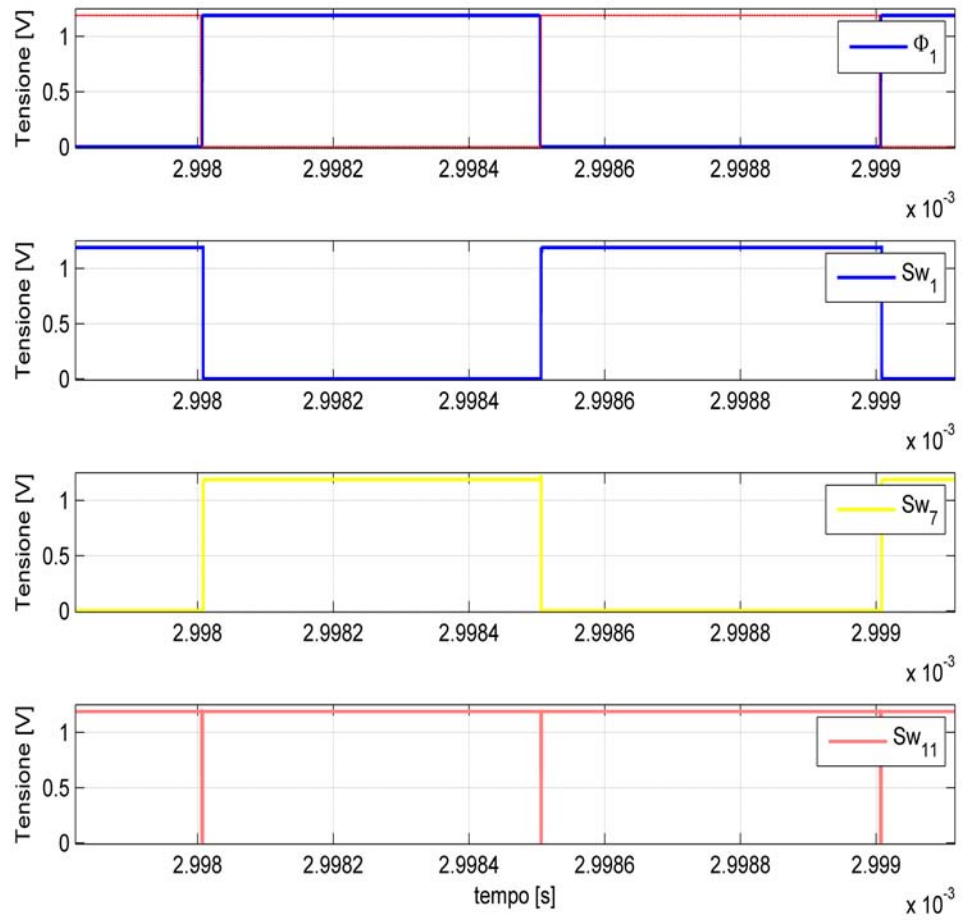


Figura 3.44: Ordine di accensione per la fase Φ_1 .

La tabella 2.16 mostra che nella fase Φ_1 vengono accesi gli switch: Sw_1 , Sw_7 e Sw_{11} e la figura 3.44 lo conferma. La stessa tabella mostra inoltre che nella fase Φ_2 devono accendersi i transistor: Sw_3 , Sw_5 e Sw_{11} e questo viene confermato dall'immagine 3.45.

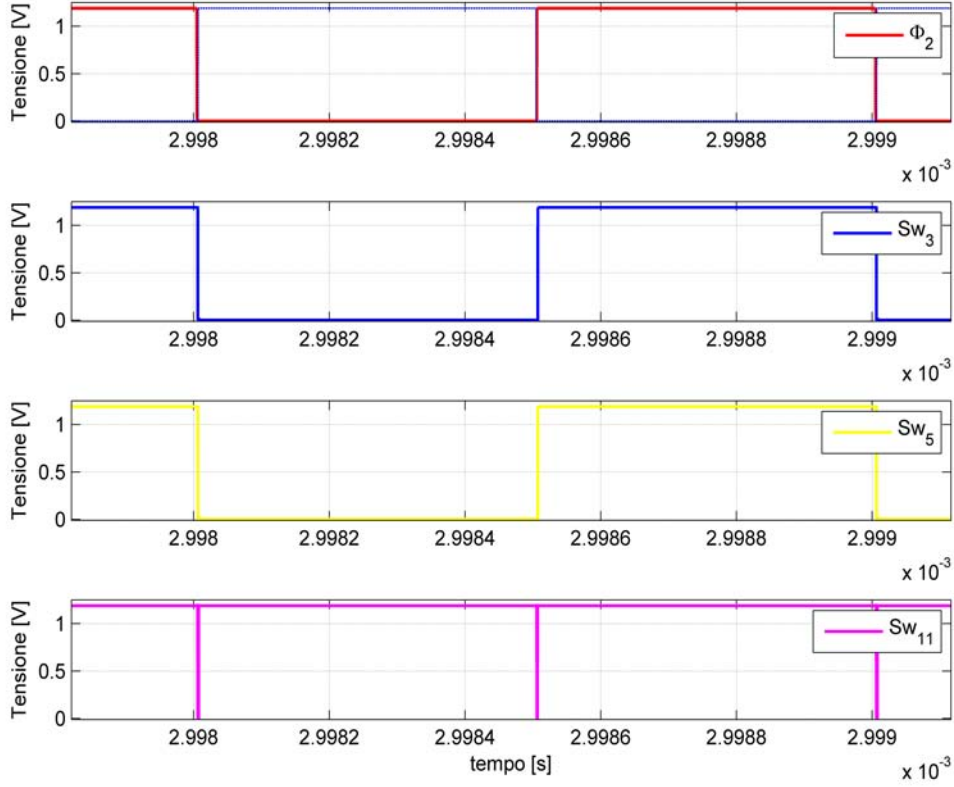


Figura 3.45: Ordine di accensione per la fase Φ_2 .

Anche in questo caso si può fare la stessa correlazione tra durata di accensione di un transistor e la sua resistenza serie calcolata in precedenza.

La figura 3.46 invece mostra l'andamento della tensione di uscita: il tratto iniziale è sempre dovuto all'azione del controllore. La cosa importante da notare è che l'andamento teorico (con le perdite resistive degli switch) prevedeva una tensione a regime di $V_{out,teorico} = 1.1890V$ (con una tensione di ingresso di $V_{in} = 0.601V$), mentre i dati di simulazione mostrano che la tensione di uscita media a regime vale $\overline{V_{out,sim}} = 1.1874V$. La previsione teorica è corretta al:

$$\frac{\overline{V_{out,sim}}}{V_{out,teorico}} = 99.15\%$$

ciò significa che il modello analizzato e simulato in MatLab sono pressoché equivalenti.

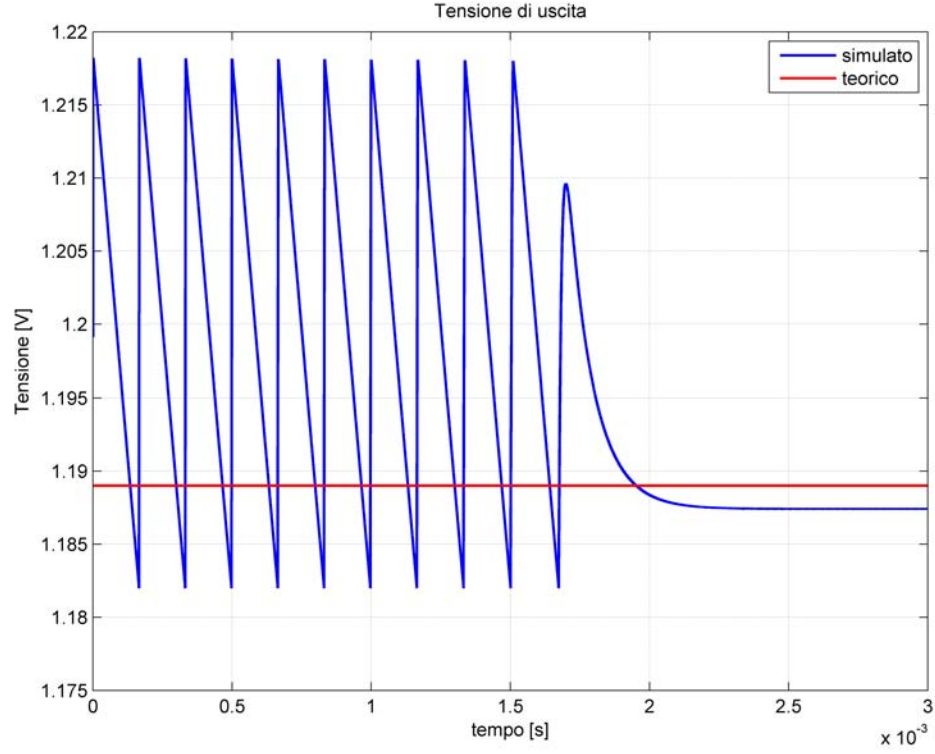


Figura 3.46: Andamento della tensione di uscita.

Il modello teorico che si era calcolato per la tensione di uscita è:

$$V_{out} = 2V_{in} - 2 \cdot \frac{V_{out}}{R_{load}} \cdot (R_{sw-\Phi_1} + R_{sw-\Phi_2})$$

Il rapporto di conversione è: $\frac{1.1874}{0.601} = 1.976$.

L'immagine 3.47 mostra l'andamento della tensione sul condensatore C_1 e si può osservare che il valore teorico e quello simulato sono abbastanza vicini come valori: infatti essendo rispettivamente: $V_{C1,teorico} = 0.5971V$ e $\overline{V_{C1,sim}} = 0.5921$ si ha uno scarto che non supera il 0.84%. Il modello teorico usato in MatLab per il calcolo della tensione V_{C1} a regime è:

$$V_{C1} = V_{in} - \frac{V_{out}}{R_{load}} R_{sw-\Phi_2}$$

e si ricava facilmente dal modello della tensione di uscita e sostituendolo nel sistema di partenza.

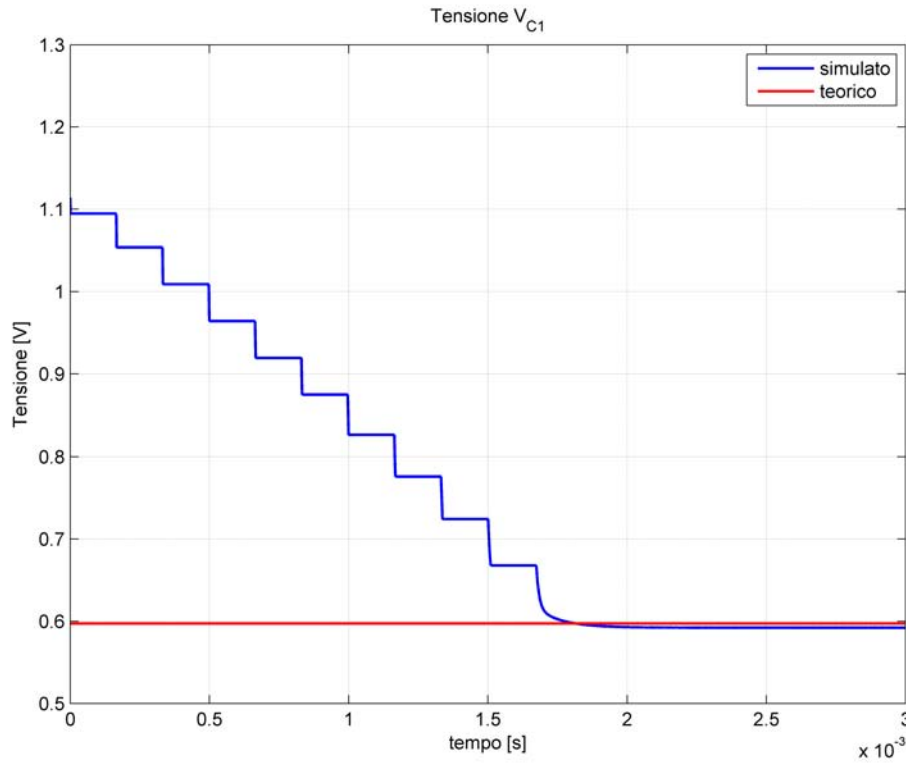


Figura 3.47: Andamento della tensione sul condensatore C_1 .

Una volta verificato che il funzionamento della parte logica è corretto e che il modello teorico con le relative perdite coincide con quello della simulazione si può andare ad osservare in dettaglio come varia la tensione ai capi dei condensatori durante le varie fasi. Ricordando la figura 2.6 ci si può aspetta il seguente comportamento:

- fase Φ_1 :
 - La tensione di uscita sale poichè il condensatore di uscita C_{out} si carica;
 - Il condensatore C_1 si sta scaricando in favore di C_{out} e quindi la sua tensione cala;
- fase Φ_2 :
 - La tensione di uscita scende poichè il condensatore di uscita si sta scaricando a causa del carico R_{load} ;

- Il condensatore C_1 si sta caricando e quindi la sua tensione sale.

La figura 3.48 mostra esattamente quanto appena specificato.

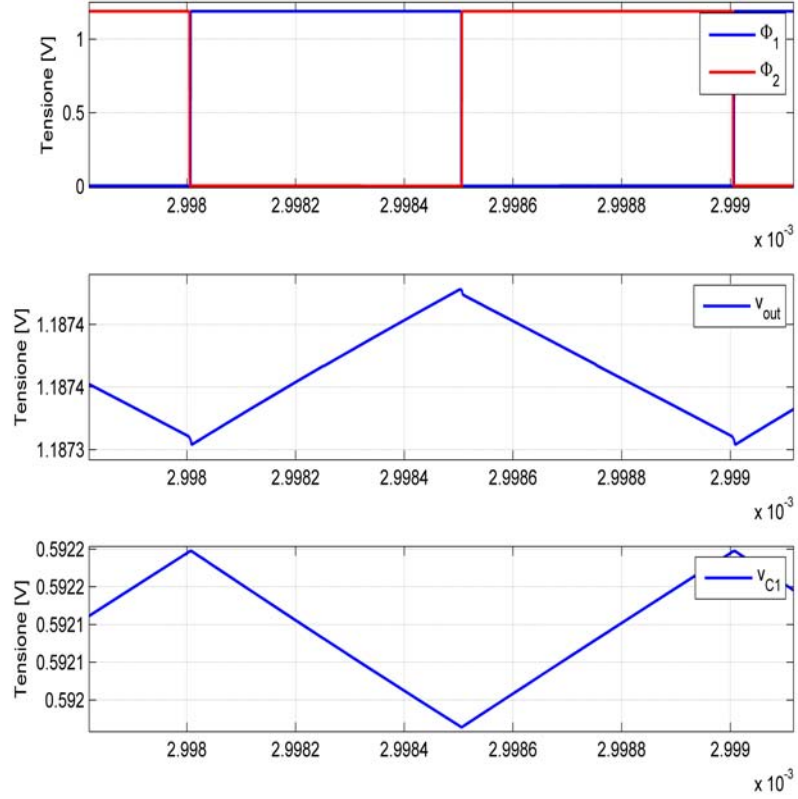


Figura 3.48: Andamento della tensioni.

Anche in questo caso, i picchi della tensione di uscita sono causati dal fenomeno del *charge injection* provocato dal transistor Sw_1 .

L'ultimo controllo da effettuare per confermare la correttezza dello schematico e del suo funzionamento è stato verificare se, con il dimensionamento ottenuto (dei condensatori e dei transistor), il convertitore lavora effettivamente nella condizione di fast switching mode. Come nel caso precedente, se la tensione ai capi del condensatore è lineare a tratti e se corrente è costante durante l'intera fase allora si può confermare il corretto regime di funzionamento. La figura 3.49 mostra proprio l'andamento descritto e quindi si può confermare che il convertitore lavora in fast switching mode.

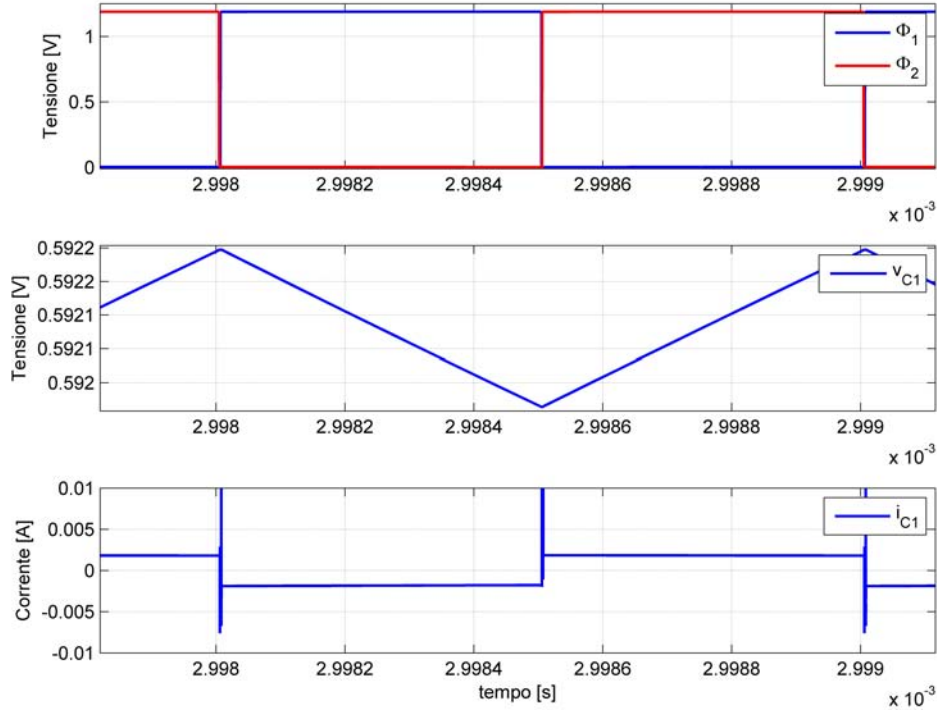


Figura 3.49: Verifica del funzionamento in FSL di C_1 .

Gli impulsi di corrente che si vedono in entrambe le immagini sono da attribuirsi ai transistori tra il cambio delle fasi.

Tuttavia anche in questo caso è possibile notare che il convertitore lavora al limite del fast switching mode. Infatti la corrente sul condensatore C_1 durante la prima fase risulta avere una leggera pendenza. Come spiegato in precedenza se si aumentano le dimensioni dei condensatori si riduce ancor di più questo effetto e non si va ad impattare sul consumo generale del sistema.

3.2 Verifica funzionamento Gear Control

Si è valutato anche il corretto del gear control: il test è stato effettuato ponendo come generatore di tensione di ingresso un *piecewise linear generator* dove la rampa della tensione era talmente lenta da lasciar commutare senza difficoltà il blocco in questione. La figura 3.50 mostra l'esatto funzionamento del blocco.

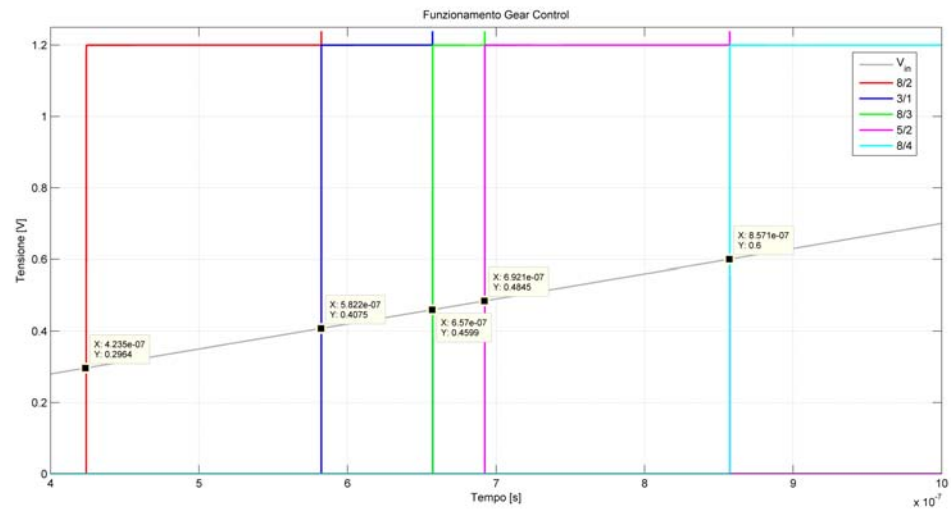


Figura 3.50: Verifica del funzionamento del gear control.

Le tensioni di soglia ottenute sono:

- $OFF \rightarrow 8/2$: 0.3V;
- $8/2 \rightarrow 3/1$: 0.410V;
- $3/1 \rightarrow 8/3$: 0.460V;
- $8/3 \rightarrow 5/2$: 0.486V;
- $5/2 \rightarrow 8/4$: 0.6V;

Si ricorda che in questo blocco i comparatori e la rete resistiva per creare i partitori di tensione sono ancora ideali. Il riferimento di tensione utilizzato è pari a 0.6V per il semplice fatto che anche se si fosse costruito un riferimento di tensione a band-gap, la tensione di alimentazione non sarebbe mai stata 1.2V per ogni tensione di ingresso.

3.3 Verifica funzionamento controllo isteretico

La figura 3.51 mostra il funzionamento del controllore isteretico. Quando la tensione di uscita (scalata di un fattore due per essere confrontabile con i riferimenti) scende fino a $V_{ref-down}$ (soglia inferiore) il segnale di set S fa un impulso e porta il segnale di $enable$ al valore logico 1. Quando invece tocca la soglia superiore allora è il segnale di reset R che porta a zero il segnale di enable; così facendo il clock non viene più trasmesso al phase divider il quale non genera più nessuna fase e tutti i transistor del convertitore risultano spenti.

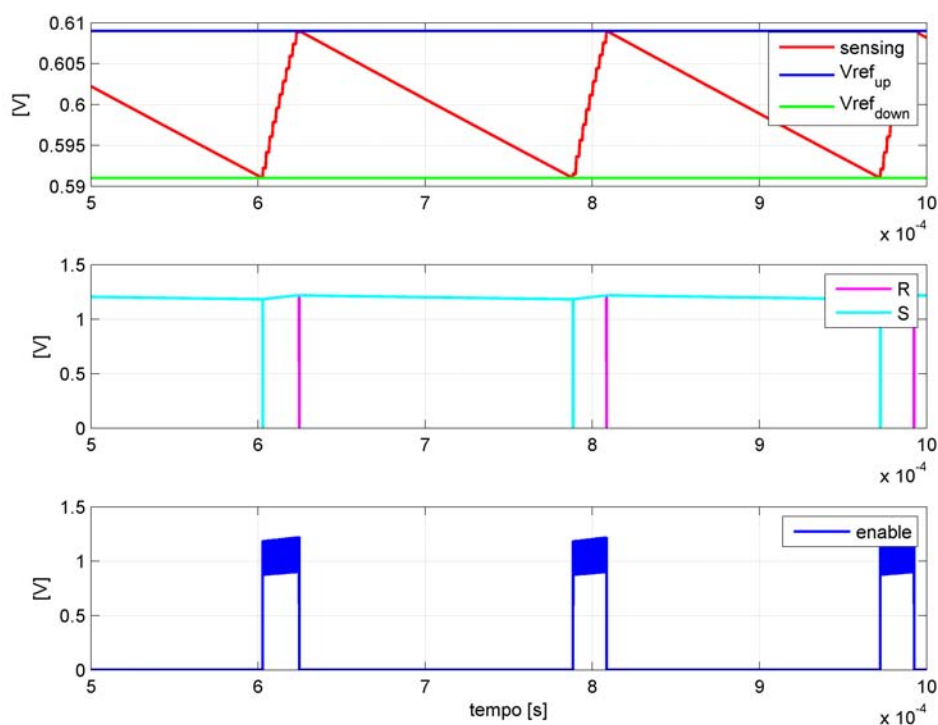


Figura 3.51: Verifica del funzionamento del controllo isteretico.

L'immagine 3.52 invece è uno zoom per evidenziare come il clock che gestisce il phase divider venga abilitato e disabilitato dal segnale $enable$.

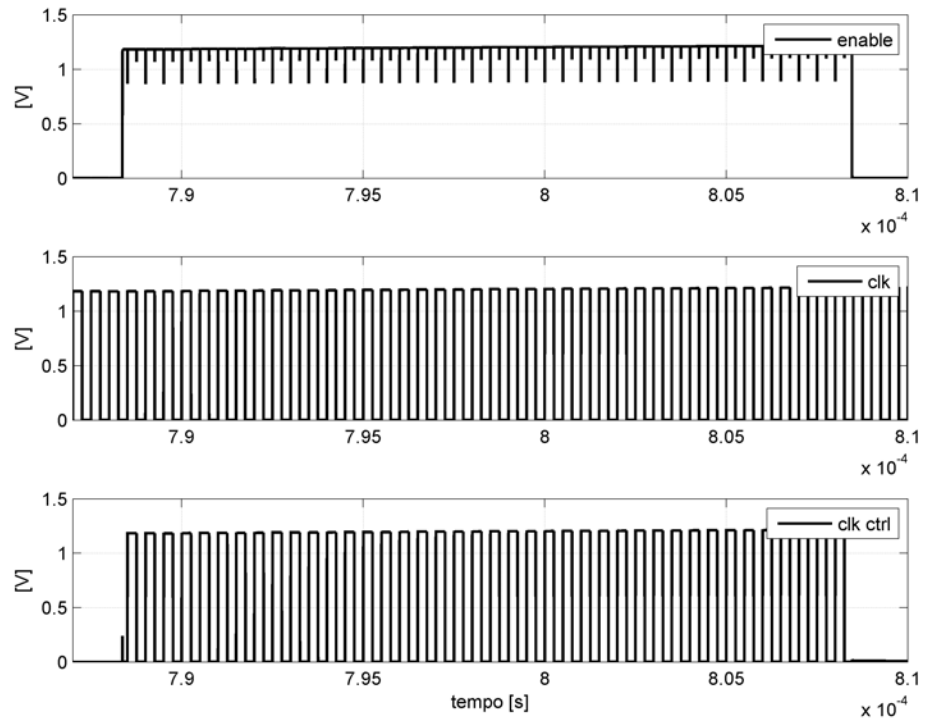


Figura 3.52: Zoom sul controllo del clock.

3.4 Efficienza

Le simulazioni per questo genere di calcolo sono state fatte impostando sempre una tensione iniziale $V_{out} = 1.2V$ e facendo un transitorio della durata di $5ms$ in modalità *liberal*. È stato possibile usare la modalità *liberal* per il fatto che non serviva un dettaglio di simulazione, ma un lungo periodo nel quale integrare ed eventuali piccoli errori vengono mediati ed annullati nell'operazione di integrazione.

L'efficienza dell'intero sistema è stata calcolata nel seguente modo:

$$\eta = \frac{P_{out}}{P_{in} + P_{supply}} = \frac{\int_t (V_{out} \cdot i_{out}) dt}{\int_t (V_{in} \cdot i_{in}) dt} \quad (3.1)$$

dove le correnti e le tensioni sono intese a regime.

Per quantificare l'efficienza per ogni tensione di ingresso è bastato fare un'analisi parametrica al variare della V_{in} . Per velocizzare la simulazione è stata impostata una condizione iniziale sulla tensione di uscita $V_{out} = 1.2V$ e poi si è atteso il raggiungimento del regime operativo (tempo di simulazione del transitorio $3ms$). La figura sottostante riporta l'andamento dell'efficienza al variare della tensione d'ingresso. La potenza erogata al carico è di $1mW$ circa ($R_{load} = 1440\Omega$).

Come si può notare la figura 3.53 rispecchia abbastanza la figura 1.9 e l'efficienza media risulta essere pari a: $\bar{\eta} = 0.828$. Tale valore risulta essere promettente se si considera il fatto che i convertitori dc-dc in commercio hanno un'efficienza media che si aggira tra il 0.6 e 0.7. Tuttavia bisogna anche ricordare che all'interno del circuito ci sono ancora dei componenti ideali quali: generatore di clock, comparatori e i generatori di tensioni di riferimento. La somma del loro consumo energetico non sarà completamente trascurabile e di conseguenza andrà ad abbassare il valore di $\bar{\eta}$.

La figura 3.54 rappresenta invece l'andamento dell'efficienza con potenze sul carico di $1mW$ e $2mW$. Si può ben notare che l'efficienza non è peggiorata, anzi dai conti risulta che $\bar{\eta} = 0.844$. Questo aumento di efficienza è probabilmente da considerarsi fuorviante e la causa è da attribuirsi ad un intervallo di integrazione troppo piccolo per i dati relativi alla potenza di $1mW$.

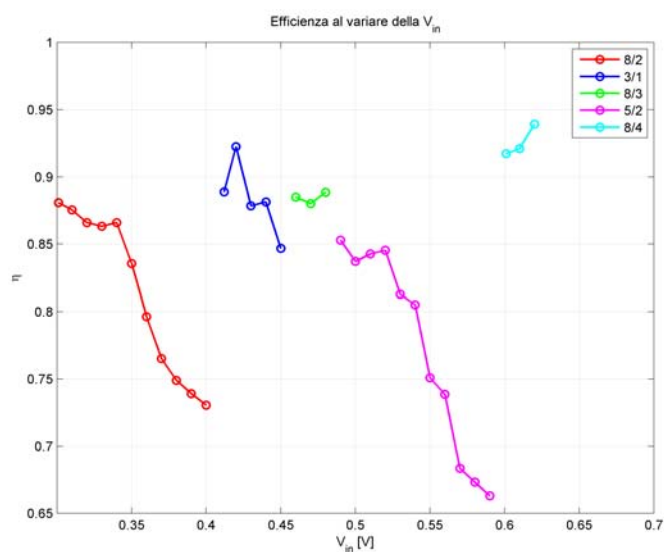


Figura 3.53: Andamento dell'efficienza con potenza di uscita di 1mW.

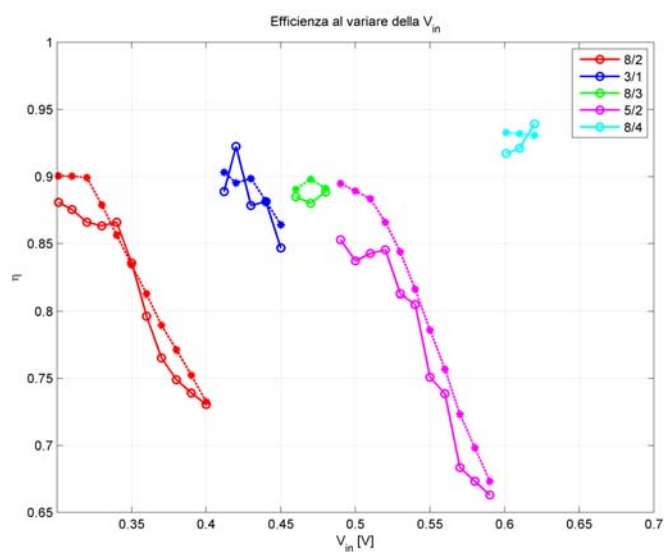


Figura 3.54: Andamento dell'efficienza con diverse potenza di uscita; il grafico con i pallini vuoti è relativo alla potenza di 1mW e l'altro con potenza di 2mW.

3.5 Simulazione start-up

La simulazione per verificare lo start-up del circuito è stata fatta imponendo un condensatore di uscita di $C_{out} = 500nF$ e tale sostituzione è solo dovuta al fatto che con valori maggiori la simulazione sarebbe durata troppo a lungo e producendo una mole di dati difficili da trattare. Tuttavia nulla cambia per il fatto che il carico R_{load} risulta sconnesso grazie al circuito di *power-good*. L'unico requisito sul valore della capacità del condensatore è dovuta al fatto che per valori minori si scarica troppo velocemente impedendo alla pompa di Dickson di raggiungere la tensione di $0.75V$. La tensione di ingresso è quella minima: $V_{in} = 0.3V$.

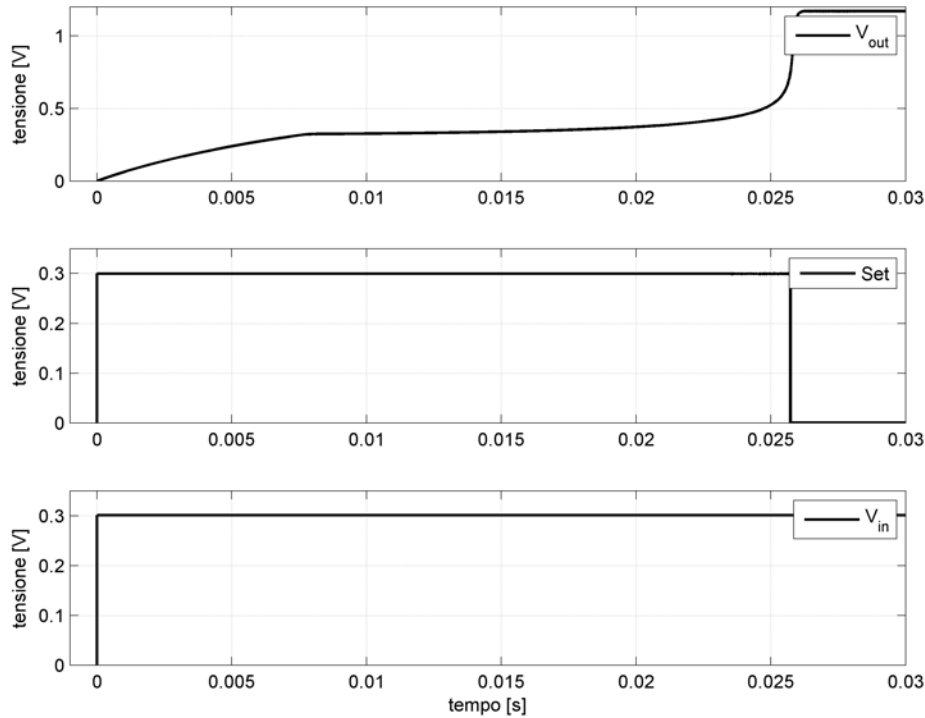


Figura 3.55: Zoom sulla tensione di uscita durante il funzionamento anche con la il resto del circuito.

La figura 3.55 riassume il funzionamento di accensione del circuito: una volta applicata una tensione di ingresso, lo startup si attiva e alza il segnale di set S . Il condensatore di uscita comincia a caricarsi, quando la tensione

V_{out} raggiunge i $0.32V$ circa (tensione di soglia dei transistor del resto del circuito) allora si verifica un flesso sull'andamento della V_{out} per il fatto che la logica di controllo comincia a fare un effetto di carico per lo start-up. In questa fase comincia un regime di funzionamento ibrido dove sia la pompa di carica di Dickson che il convertitore principale lavorano assieme. Mano a mano che la tensione sale, gli switch di scc si accendono sempre di più (aumenta la tensione al loro gate) e la pendenza di V_{out} aumenta sempre di più. In prossimità di $V_{out} = 0.75V$ (figura 3.56) il circuito si statup si sgancia ed il segnale di set va a zero e il convertitore principale raggiunge velocemente il suo regime operativo.

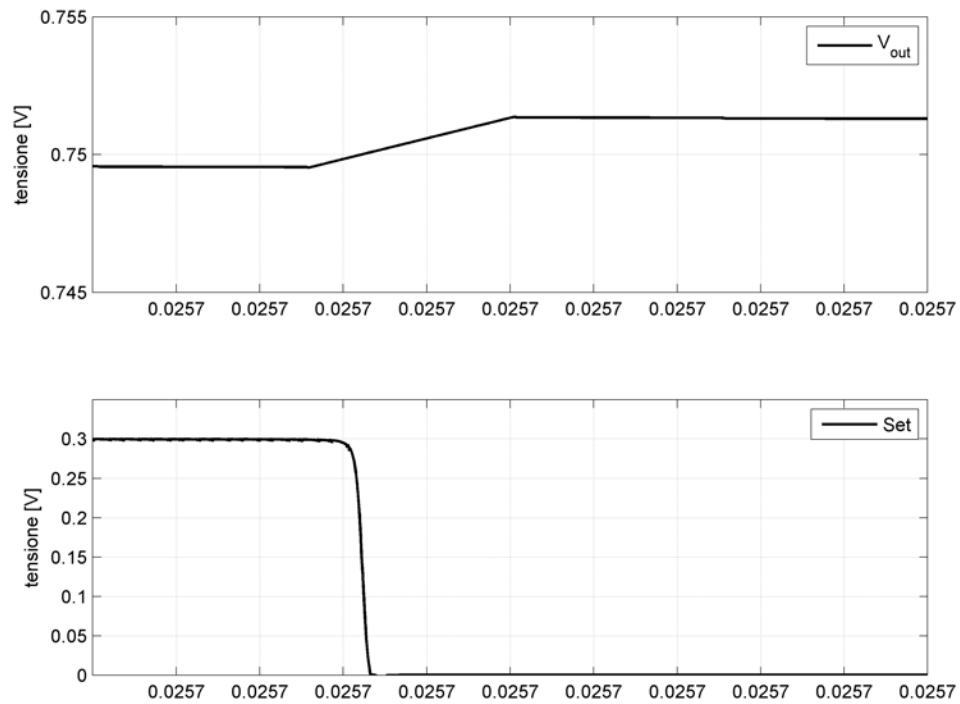


Figura 3.56: Zoom nell'istante di sgancio del circuito di startup.

La figura 3.57 mostra l'andamento della tensione di uscita prodotta dalla pompa di carica, si nota un funzionamento quasi in *FSL*. Invece la figura 3.58 mostra sempre la tensione di uscita, dove la pompa di Dickson lavora assieme al circuito principale.

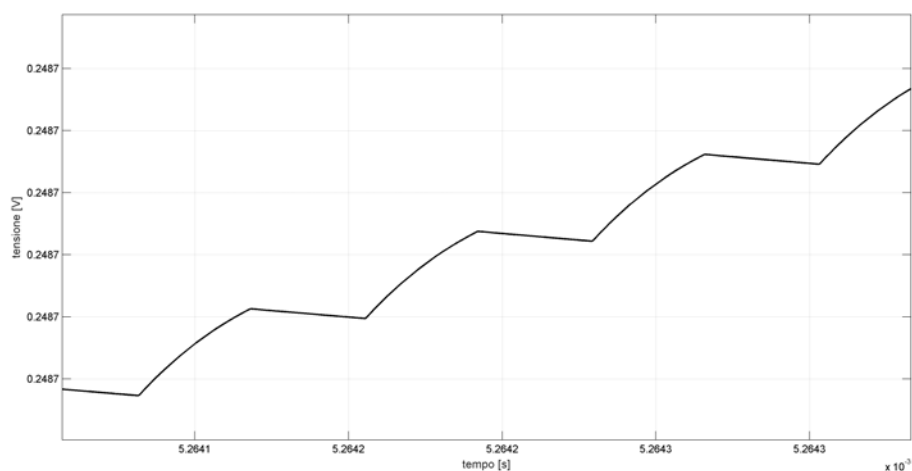


Figura 3.57: Zoom sulla tensione di uscita durante il funzionamento della sola pompa di carica.

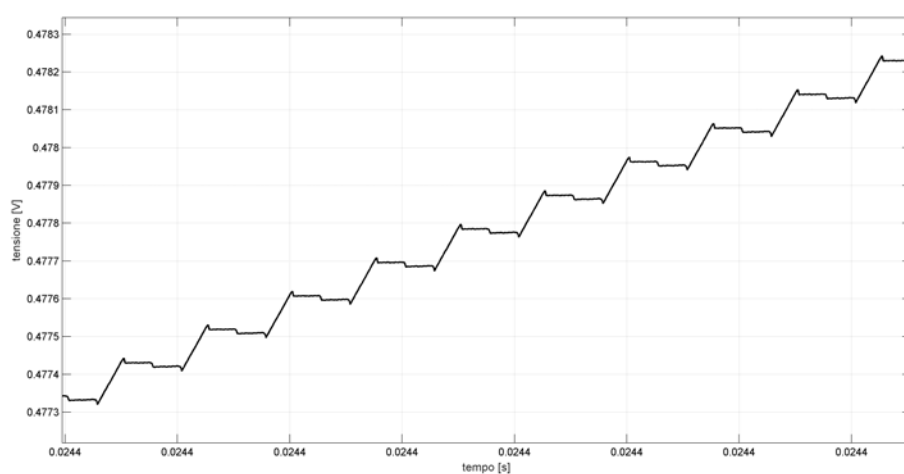


Figura 3.58: Zoom sulla tensione di uscita durante il funzionamento anche con la il resto del circuito. Sono ben visibili gli effetti di carico delle tre fasi del convertitore 8/2 e anche il periodo di non sovrapposizione delle fasi.

Capitolo 4

Conclusioni

I dati di simulazione hanno confermato che ogni blocco funziona correttamente: la parte di logica sceglie il giusto rapporto di conversione e sa creare la giusta sequenza di comandi per gli switch. Il blocco del controllo isteretico regola correttamente la tensione di uscita in base al ripple che viene specificato dalle soglie. Il blocco di start-up, con il dovuto tempo, riesce ad accendere il circuito e poi si sconnette automaticamente. Le specifiche sono state soddisfatte per quanto riguarda la tensione di uscita e la potenza erogata garantendo un livello di efficienza attorno all'80%. Tuttavia il convertitore sa erogare anche il doppio della potenza con un livello di efficienza ancora accettabile. Bisogna ricordare che gli unici componenti ideali presi dalla libreria *analogLib* di Cadence sono: il generatore di clock (*vpulse*), i generatori di tensione di riferimento (*vdc*), le resistenze per il gear control (*res*) e i comparatori (*vcvs*: voltage gain=1M, maximum output voltage=1.2V e minimum output voltage=0V). Per quanto riguarda la realizzazione del generatore di clock si potrebbe realizzarlo o con un oscillatore o con un ring oscillator CMOS. La scelta tra i due sarà fatta in base al minor consumo di potenza a parità di frequenza di lavoro. I generatori di riferimento possono essere realizzati impostando un unico generatore di corrente a band-gap che fa scorrere una corrente nota su una serie di resistenze che realizzano i vari riferimenti. Infine i comparatori possono essere realizzati con un semplice OTA a cinque transistor single ended e poi l'uscita bufferizzata da una serie di invertitori CMOS. Non è richiesta una banda elevata per OTA e questo è positivo per quanto riguarda il consumo di potenza.

Le questioni che restano aperte riguardano l'implementazione di ulteriori due fattori di conversione ($7/2$ e $7/3$) che dovrebbero aumentare l'efficienza media del sistema e sfruttare al 100% il convertitore. Cercare di studiare l'effetto delle capacità parassite dei transistor del scc e le resistenze parassite per la connessione dei condensatori C_{fly} (offchip), provare a migliorare la

logica di controllo provando con il linguaggio *VHDL*, oltre al fatto di eseguire delle simulazioni Montecarlo per vedere l'affidabilità del circuito con lo spread di processo.

Appendice A

Ridondanza codice EXB

In precedenza si è accennato al fatto che il procedimento per produrre i numeri EXB è un procedimento ridondante e nel caso specifico del convertitore $8/3$ è stato possibile togliere una fase (e quindi un numero EXB) non compromettendo il funzionamento del convertitore stesso. Prendendo proprio in esame il caso del convertitore $8/3$ si ricorda che il procedimento fornisce i seguenti numeri EXB (per il numero $3/8$):

A_0	A_1	A_2	A_3
1	-1	-1	1
0	1	-1	1
1	-1	0	-1
0	1	0	-1
0	0	1	1

Tabella A.1: Tabella riassuntiva dei codici EXB per il numero $3/8$.

In base alla tabella A.1 è possibile disegnare i 5 circuiti che rappresentano le 5 fasi:

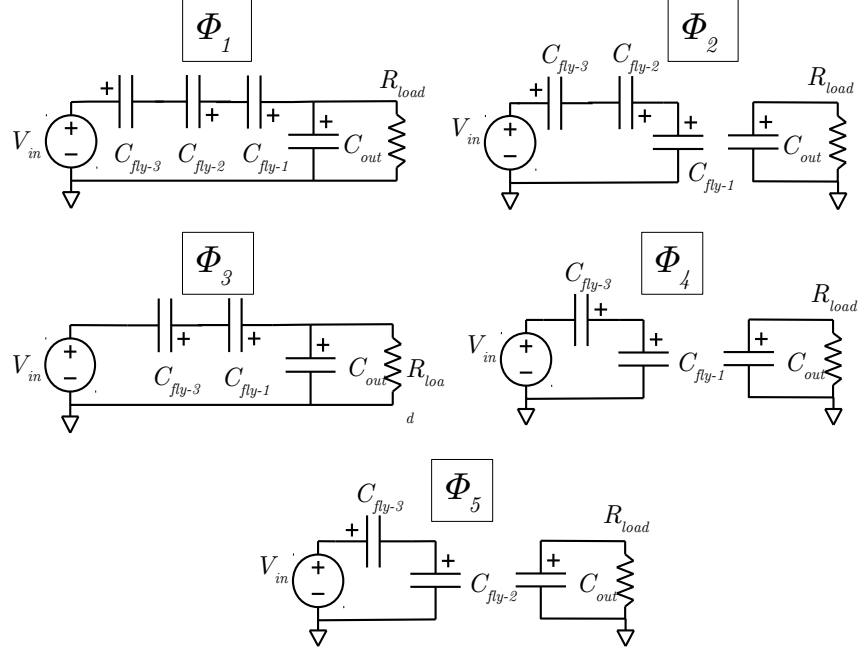


Figura A.1: Fasi per il convertitore step-up 8/3.

Se si scrive la legge di Kirchhoff (*KCL*) per ogni fase ed esprimendole in funzione di una sola corrente (per esempio corrente sul condensatore C_3) si ottiene la seguente relazione:

$$\begin{bmatrix} -1 & -1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 & 1 \\ -1 & -1 & 0 & 0 & 1 \\ -1 & 1 & 1 & -1 & 0 \\ 1 & 0 & -1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} a_{C_3}^1 \\ a_{C_3}^2 \\ a_{C_3}^3 \\ a_{C_3}^4 \\ a_{C_3}^5 \end{bmatrix} = \begin{bmatrix} \frac{8}{3} \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (\text{A.1})$$

Tale equazione non è risolvibile per il fatto che la matrice 5×5 è singolare, infatti il suo determinante è nullo ed il suo rango è 4. Questo significa che c'è una colonna (o riga) che è linearmente dipendente dalle altre e quindi è possibile toglierla. Togliere una di quelle colonne significa togliere una di quelle fasi e di conseguenza togliere un numero EXB.

La figura sottostante rappresenta la tensione di uscita a regime dopo aver tolto la fase numero 4 e rinominato la fase numero 5 con il numero 4, come si può ben notare il sistema raggiunge il valore a regime senza nessun problema. L'undershoot è solo dovuto ad un transitorio di simulazione.

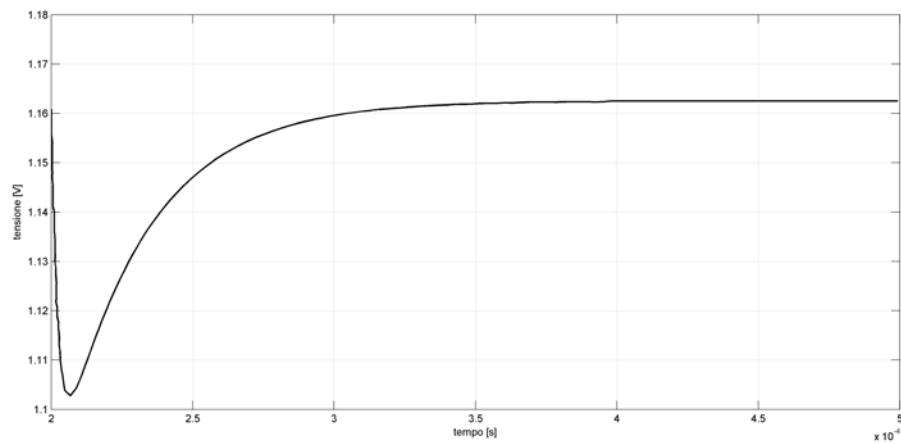


Figura A.2: Fasi per il convertitore step-up 8/3.

Appendice B

Codici MatLab

B.1 Resistenza dei MOSFET e dimensionamento switch SCC

```
%% Switch optimization
clear all; close all; clc;

load Rnmos.mat;
w=R_nMOSVsource0X(10:end);
R_v00=R_nMOSVsource0Y(10:end);
R_v01=R_nMOSVsource01Y(10:end);
R_v02=R_nMOSVsource02Y(10:end);
R_v03=R_nMOSVsource03Y(10:end);
R_v04=R_nMOSVsource04Y(10:end);
R_v05=R_nMOSVsource05Y(10:end);
R_v06=R_nMOSVsource06Y(10:end);
R_v07=R_nMOSVsource07Y(10:end);
R_v08=R_nMOSVsource08Y(10:end);
R_v09=R_nMOSVsource09Y(10:end);
R_v1=R_nMOSVsource1Y(10:end);
R_v11=R_nMOSVsource11Y(10:end);
R_v12=R_nMOSVsource12Y(10:end);

k_nmos=[0 0 0 0 0 0 0 0 0 0 0 0 0];
[xData, yData] = prepareCurveData( w, R_v00 );
% Set up fittype and options.
ft = fittype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off'; opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );
```

```

k_nmos(1)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult , xData , yData );
% legend( h , 'R_v00 vs. w', 'untitled fit 1', 'Location',
'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on

[xData , yData] = prepareCurveData( w , R_v01 );
% Set up fitype and options.
ft = fitype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off'; opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult , gof] = fit( xData , yData , ft , opts );
k_nmos(2)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult , xData , yData );
% legend( h , 'R_v00 vs. w', 'untitled fit 1', 'Location',
'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on

[xData , yData] = prepareCurveData( w , R_v02 );
% Set up fitype and options.
ft = fitype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off'; opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult , gof] = fit( xData , yData , ft , opts );
k_nmos(3)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult , xData , yData );
% legend( h , 'R_v00 vs. w', 'untitled fit 1', 'Location',
'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on

[xData , yData] = prepareCurveData( w , R_v03 );
% Set up fitype and options.
ft = fitype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );

```


B.1. RESISTENZA DEI MOSFET E DIMENSIONAMENTO SWITCH SCC131

```
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off';          opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );
k_nmos(4)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult, xData, yData );
% legend( h, 'R_v00 vs. w', 'untitled fit 1', 'Location',
'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on

[xData, yData] = prepareCurveData( w, R_v04 );
% Set up fittype and options.
ft = fittype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off';          opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );
k_nmos(5)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult, xData, yData );
% legend( h, 'R_v00 vs. w', 'untitled fit 1', 'Location',
'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on

[xData, yData] = prepareCurveData( w, R_v05 );
% Set up fittype and options.
ft = fittype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off';          opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );
k_nmos(6)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult, xData, yData );
% legend( h, 'R_v00 vs. w', 'untitled fit 1', 'Location',
'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on
```

```

[xData, yData] = prepareCurveData( w, R_v06 );
% Set up fittype and options.
ft = fitttype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off'; opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );
k_nmos(7)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult, xData, yData );
% legend( h, 'R_v00 vs. w', 'untitled fit 1', 'Location',
% 'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on

[xData, yData] = prepareCurveData( w, R_v07 );
% Set up fittype and options.
ft = fitttype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off'; opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );
k_nmos(8)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult, xData, yData );
% legend( h, 'R_v00 vs. w', 'untitled fit 1', 'Location',
% 'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on

[xData, yData] = prepareCurveData( w, R_v08 );
% Set up fittype and options.
ft = fitttype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off'; opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );
k_nmos(9)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );

```

B.1. RESISTENZA DEI MOSFET E DIMENSIONAMENTO SWITCH SCC133

```
% h = plot( fitresult , xData , yData );
% legend( h , 'R_v00 vs. w', 'untitled fit 1', 'Location',
'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on

[xData , yData] = prepareCurveData( w , R_v09 );
% Set up fittype and options.
ft = fittype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off'; opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult , gof] = fit( xData , yData , ft , opts );
k_nmos(10)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult , xData , yData );
% legend( h , 'R_v00 vs. w', 'untitled fit 1', 'Location',
'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on

[xData , yData] = prepareCurveData( w , R_v1 );
% Set up fittype and options.
ft = fittype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off'; opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult , gof] = fit( xData , yData , ft , opts );
k_nmos(11)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult , xData , yData );
% legend( h , 'R_v00 vs. w', 'untitled fit 1', 'Location',
'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on

[xData , yData] = prepareCurveData( w , R_v11 );
% Set up fittype and options.
ft = fittype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off'; opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
```

```

% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );
k_nmos(12)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult, xData, yData );
% legend( h, 'R_v00 vs. w', 'untitled fit 1', 'Location',
'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on

[xData, yData] = prepareCurveData( w, R_v12 );
% Set up fitype and options.
ft = fitype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off'; opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );
k_nmos(13)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult, xData, yData );
% legend( h, 'R_v00 vs. w', 'untitled fit 1', 'Location',
'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on

load Rnmosbulk.mat;
R_v00b=R_nMOSVsource0Y(10:end);
R_v01b=R_nMOSVsource01Y(10:end);
R_v02b=R_nMOSVsource02Y(10:end);
R_v03b=R_nMOSVsource03Y(10:end);
R_v04b=R_nMOSVsource04Y(10:end);
R_v05b=R_nMOSVsource05Y(10:end);
R_v06b=R_nMOSVsource06Y(10:end);
R_v07b=R_nMOSVsource07Y(10:end);
R_v08b=R_nMOSVsource08Y(10:end);
R_v09b=R_nMOSVsource09Y(10:end);
R_v1b=R_nMOSVsource1Y(10:end);
R_v11b=R_nMOSVsource11Y(10:end);
R_v12b=R_nMOSVsource12Y(10:end);
% nb -> nMos con il bulk a massa
k_nmosb=[0 0 0 0 0 0 0 0 0 0 0 0 0];
[xData, yData] = prepareCurveData( w, R_v00b );
% Set up fitype and options.
ft = fitype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );

```

```

opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off';          opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );
k_nmosb(1)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult, xData, yData );
% legend( h, 'R_v00 vs. w', 'untitled fit 1', 'Location',
'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on

[xData, yData] = prepareCurveData( w, R_v01b );
% Set up fittype and options.
ft = fittype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off';          opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );
k_nmosb(2)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult, xData, yData );
% legend( h, 'R_v00 vs. w', 'untitled fit 1', 'Location',
'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on

[xData, yData] = prepareCurveData( w, R_v02b );
% Set up fittype and options.
ft = fittype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off';          opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );
k_nmosb(3)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult, xData, yData );
% legend( h, 'R_v00 vs. w', 'untitled fit 1', 'Location',
'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on

```

```

[xData, yData] = prepareCurveData( w, R_v03b );
% Set up fittype and options.
ft = fittype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off'; opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );
k_nmosb(4)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult, xData, yData );
% legend( h, 'R_v00 vs. w', 'untitled fit 1', 'Location',
% 'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on

[xData, yData] = prepareCurveData( w, R_v04b );
% Set up fittype and options.
ft = fittype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off'; opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );
k_nmosb(5)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult, xData, yData );
% legend( h, 'R_v00 vs. w', 'untitled fit 1', 'Location',
% 'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on

[xData, yData] = prepareCurveData( w, R_v05b );
% Set up fittype and options.
ft = fittype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off'; opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );
k_nmosb(6)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );

```

B.1. RESISTENZA DEI MOSFET E DIMENSIONAMENTO SWITCH SCC137

```
% h = plot( fitresult , xData , yData );
% legend( h , 'R_v00 vs. w', 'untitled fit 1', 'Location',
'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on

[xData , yData] = prepareCurveData( w , R_v06b );
% Set up fittype and options.
ft = fittype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off'; opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult , gof] = fit( xData , yData , ft , opts );
k_nmosb(7)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult , xData , yData );
% legend( h , 'R_v00 vs. w', 'untitled fit 1', 'Location',
'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on

[xData , yData] = prepareCurveData( w , R_v07b );
% Set up fittype and options.
ft = fittype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off'; opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult , gof] = fit( xData , yData , ft , opts );
k_nmosb(8)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult , xData , yData );
% legend( h , 'R_v00 vs. w', 'untitled fit 1', 'Location',
'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on

[xData , yData] = prepareCurveData( w , R_v08b );
% Set up fittype and options.
ft = fittype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off'; opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
```

```

% Fit model to data.
[fitresult , gof] = fit( xData , yData , ft , opts );
k_nmosb(9)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult , xData , yData );
% legend( h , 'R_v00 vs. w', 'untitled fit 1', 'Location',
'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on

[xData , yData] = prepareCurveData( w , R_v09b );
% Set up fitype and options.
ft = fitype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off'; opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult , gof] = fit( xData , yData , ft , opts );
k_nmosb(10)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult , xData , yData );
% legend( h , 'R_v00 vs. w', 'untitled fit 1', 'Location',
'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on

[xData , yData] = prepareCurveData( w , R_v1b );
% Set up fitype and options.
ft = fitype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off'; opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult , gof] = fit( xData , yData , ft , opts );
k_nmosb(11)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult , xData , yData );
% legend( h , 'R_v00 vs. w', 'untitled fit 1', 'Location',
'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on

[xData , yData] = prepareCurveData( w , R_v11b );
% Set up fitype and options.

```



```

ft = fittype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off'; opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );
k_nmosb(12)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult, xData, yData );
% legend( h, 'R_v00 vs. w', 'untitled fit 1', 'Location',
'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on

[xData, yData] = prepareCurveData( w, R_v12b );
% Set up fittype and options.
ft = fittype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off'; opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );
k_nmosb(13)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult, xData, yData );
% legend( h, 'R_v00 vs. w', 'untitled fit 1', 'Location',
'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on

load Rpmos.mat;
Rp_v00=R_pMOSVsource0Y(10:end);
Rp_v01=R_pMOSVsource01Y(10:end);
Rp_v02=R_pMOSVsource02Y(10:end);
Rp_v03=R_pMOSVsource03Y(10:end);
Rp_v04=R_pMOSVsource04Y(10:end);
Rp_v05=R_pMOSVsource05Y(10:end);
Rp_v06=R_pMOSVsource06Y(10:end);
Rp_v07=R_pMOSVsource07Y(10:end);
Rp_v08=R_pMOSVsource08Y(10:end);
Rp_v09=R_pMOSVsource09Y(10:end);
Rp_v1=R_pMOSVsource1Y(10:end);
Rp_v11=R_pMOSVsource11Y(10:end);

k_pmos=[0 0 0 0 0 0 0 0 0 0 0 0 0];

```

```

[xData, yData] = prepareCurveData( w, Rp_v00 );
% Set up fitttype and options.
ft = fitttype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off';      opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );
k_pmos(1)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult, xData, yData );
% legend( h, 'R_v00 vs. w', 'untitled fit 1', 'Location',
%         'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on

[xData, yData] = prepareCurveData( w, Rp_v01 );
% Set up fitttype and options.
ft = fitttype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off';      opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );
k_pmos(2)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult, xData, yData );
% legend( h, 'R_v00 vs. w', 'untitled fit 1', 'Location',
%         'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on

[xData, yData] = prepareCurveData( w, Rp_v02 );
% Set up fitttype and options.
ft = fitttype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off';      opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );
k_pmos(3)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult, xData, yData );

```

B.1. RESISTENZA DEI MOSFET E DIMENSIONAMENTO SWITCH SCC141

```
% legend( h, 'R_v00 vs. w', 'untitled fit 1', 'Location',  
    'NorthEast' );  
%% Label axes  
% xlabel w; ylabel R_v00; grid on  
  
[xData, yData] = prepareCurveData( w, Rp_v03 );  
% Set up fittype and options.  
ft = fittype( 'k/x', 'independent', 'x', 'dependent', 'y' );  
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );  
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;  
opts.Display = 'Off'; opts.Robust = 'LAR';  
opts.StartPoint = 0.913375856139019;  
% Fit model to data.  
[fitresult, gof] = fit( xData, yData, ft, opts );  
k_pmos(4)=fitresult.k;  
%% Plot fit with data.  
% figure( 'Name', 'untitled fit 1' );  
% h = plot( fitresult, xData, yData );  
% legend( h, 'R_v00 vs. w', 'untitled fit 1', 'Location',  
    'NorthEast' );  
%% Label axes  
% xlabel w; ylabel R_v00; grid on  
  
[xData, yData] = prepareCurveData( w, Rp_v04 );  
% Set up fittype and options.  
ft = fittype( 'k/x', 'independent', 'x', 'dependent', 'y' );  
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );  
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;  
opts.Display = 'Off'; opts.Robust = 'LAR';  
opts.StartPoint = 0.913375856139019;  
% Fit model to data.  
[fitresult, gof] = fit( xData, yData, ft, opts );  
k_pmos(5)=fitresult.k;  
%% Plot fit with data.  
% figure( 'Name', 'untitled fit 1' );  
% h = plot( fitresult, xData, yData );  
% legend( h, 'R_v00 vs. w', 'untitled fit 1', 'Location',  
    'NorthEast' );  
%% Label axes  
% xlabel w; ylabel R_v00; grid on  
  
[xData, yData] = prepareCurveData( w, Rp_v05 );  
% Set up fittype and options.  
ft = fittype( 'k/x', 'independent', 'x', 'dependent', 'y' );  
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );  
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;  
opts.Display = 'Off'; opts.Robust = 'LAR';  
opts.StartPoint = 0.913375856139019;  
% Fit model to data.
```

```

[fitresult , gof] = fit( xData , yData , ft , opts );
k_pmos(6)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult , xData , yData );
% legend( h , 'R_v00 vs. w', 'untitled fit 1', 'Location',
'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on

[xData , yData] = prepareCurveData( w , Rp_v06 );
% Set up fitype and options.
ft = fitype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off'; opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult , gof] = fit( xData , yData , ft , opts );
k_pmos(7)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult , xData , yData );
% legend( h , 'R_v00 vs. w', 'untitled fit 1', 'Location',
'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on

[xData , yData] = prepareCurveData( w , Rp_v07 );
% Set up fitype and options.
ft = fitype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off'; opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult , gof] = fit( xData , yData , ft , opts );
k_pmos(8)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult , xData , yData );
% legend( h , 'R_v00 vs. w', 'untitled fit 1', 'Location',
'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on

[xData , yData] = prepareCurveData( w , Rp_v08 );
% Set up fitype and options.
ft = fitype( 'k/x', 'independent', 'x', 'dependent', 'y' );

```

```

opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off';      opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );
k_pmos(9)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult, xData, yData );
% legend( h, 'R_v00 vs. w', 'untitled fit 1', 'Location',
'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on

[xData, yData] = prepareCurveData( w, Rp_v09 );
% Set up fitype and options.
ft = fitype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off';      opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );
k_pmos(10)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult, xData, yData );
% legend( h, 'R_v00 vs. w', 'untitled fit 1', 'Location',
'NorthEast' );
%% Label axes
% xlabel w; ylabel R_v00; grid on

[xData, yData] = prepareCurveData( w, Rp_v1 );
% Set up fitype and options.
ft = fitype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off';      opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );
k_pmos(11)=fitresult.k;
%% Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult, xData, yData );
% legend( h, 'R_v00 vs. w', 'untitled fit 1', 'Location',
'NorthEast' );
%% Label axes

```

```

% xlabel w; ylabel R_v00; grid on

[xData, yData] = prepareCurveData( w, Rp_v11 );
% Set up fittype and options.
ft = fittype( 'k/x', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.DiffMaxChange = 0.001; opts.DiffMinChange = 1e-15;
opts.Display = 'Off'; opts.Robust = 'LAR';
opts.StartPoint = 0.913375856139019;
% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );
k_pmos(12)=fitresult.k;
% % Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult, xData, yData );
% legend( h, 'R_v00 vs. w', 'untitled fit 1', 'Location',
% 'NorthEast' );
% % Label axes
% xlabel w; ylabel R_v00; grid on

%%
eta=0.977; RL=1440;
%%
% 8/2 considerato caso peggiore per efficienza teorica
a1=[1 0 0 1 0 0 0 1 0 1 0]; %sw1 sw2 sw3 sw4 sw5 sw6 sw7 sw8
sw9 sw10 sw11
a2=[0 1 1 0 0 0 0 1 0 1 0]; %a2(3)=-1
a3=[0 0 0 0 0 2 0 0 2 2 0]; %a3(6)=-2
% v1=[0.001 0.6 0.6 0.001 0 0.3 0 0.001 0.3 0.001 0];
% v2=[0.6 0.001 0.001 0.6 0 0.3 0 0.001 0.3 0.001 0];
% v3=[0.6 0.3 0.001 0.3 0 0.001 0 0.3 0.001 0.001 0];
eta82=eta;
R_FSL= RL*(1-eta82)/eta82;
G=[0 0 0 0 0 0 0 0 0 0 0];
sum=0;
for i=1:11
    sum=sum+a1(i)+a2(i)+a3(i);
end
Gtot=sum.*sum.*2/R_FSL;
for i=1:11
    G(i)=(a1(i)+a2(i)+a3(i)).*Gtot./sum;
end
R82=1./G;
% Rsw1=R(1)+R(4)+R(8)+R(10);
% Rsw2=R(2)+R(3)+R(8)+R(10);
% Rsw3=R(6)+R(9)+R(10);
% y=(Rsw1+Rsw2+4*Rsw3)/0.333;
Wn=[0 0 0 0 0 0 0 0 0 0 0]'; Wnb=Wn; Wp=Wn;
k82=[12 7 1 7 20 1 20 4 4 4 20]'; %indici da usare in base

```

B.1. RESISTENZA DEI MOSFET E DIMENSIONAMENTO SWITCH SCC145

```
alle tensioni di % source: 20 indica che non viene usato
for i=1:11
    a=k82(i);
    if ((a1(i)+a2(i)+a3(i))~=0)
        Wn(i)=k_nmos(a)/R82(i);
        Wp(i)=k_pmos(13-a)/R82(i);
        Wnb(i)=k_nmosb(a)/R82(i);
    end
end
W82=zeros(11, 3);
for i=1:11
    if ((a1(i)+a2(i)+a3(i))~=0)
        if (i==1)
            W82(i, 1)=Wp(i);
        elseif (i==4)
            W82(i, 1)=Wnb(i);
        else
            W82(i, 1)=Wn(i);
        end
    end
end
for i=1:11
    if ((a1(i)+a2(i)+a3(i))~=0)
        if (i==4)
            W82(i, 2)=Wnb(i);
        else
            W82(i, 2)=Wp(i);
        end
    end
end
W82(1, 3)=0;
for i=2:11
    if (W82(i, 1)<W82(i, 2))
        W82(i, 3)=1;
    else
        W82(i, 3)=0;
    end
end
end

%%
% 3/1
a1=[1 0 0 0 0 0 1 0 0 0 1]; %sw1 sw2 sw3 sw4 sw5 sw6 sw7 sw8
sw9 sw10 sw11
a2=[0 1 1 0 0 0 0 1 0 1 0];
a3=[0 0 0 0 0 1 0 0 1 1 0];
% v1=[0.001 0.6 0.6 0.001 0 0.3 0 0.001 0.3 0.001 0];
% v2=[0.6 0.001 0.001 0.6 0 0.3 0 0.001 0.3 0.001 0];
% v3=[0.6 0.3 0.001 0.3 0 0.001 0 0.3 0.001 0.001 0];
eta31=eta;
```

```

R_FSL= RL*(1-eta31)/eta31;
G=[0 0 0 0 0 0 0 0 0 0 0];
sum=0;
for i=1:11
    sum=sum+a1(i)+a2(i)+a3(i);
end
Gtot=sum.*sum.*2/R_FSL;
for i=1:11
    G(i)=(a1(i)+a2(i)+a3(i)).*Gtot./sum;
end
R31=1./G;
Rsw1=R31(1)+R31(7)+R31(11);
Rsw2=R31(2)+R31(3)+R31(8)+R31(10);
Rsw3=R31(6)+R31(9)+R31(10);
y=(Rsw1+Rsw2+Rsw3)/0.333;
Wn=[0 0 0 0 0 0 0 0 0 0 0]'; Wnb=Wn; Wp=Wn;
k31=[12 9 1 20 20 1 5 5 5 5 5]'; %indici da usare in base
    alle tensioni di
% source: 20 indica che non viene usato
for i=1:11
    a=k31(i);
    if ((a1(i)+a2(i)+a3(i))~=0)
        Wn(i)=k_nmos(a)/R31(i);
        Wp(i)=k_pmos(13-a)/R31(i);
        Wnb(i)=k_nmosb(a)/R31(i);
    end
end
W31=zeros(11, 3);
for i=1:11
    if ((a1(i)+a2(i)+a3(i))~=0)
        if (i==1)
            W31(i, 1)=Wp(i);
        elseif (i==4)
            W31(i, 1)=Wnb(i);
        else
            W31(i, 1)=Wn(i);
        end
    end
end
for i=1:11
    if ((a1(i)+a2(i)+a3(i))~=0)
        if (i==4)
            W31(i, 2)=Wnb(i);
        else
            W31(i, 2)=Wp(i);
        end
    end
end
W31(1, 3)=0;

```


B.1. RESISTENZA DEI MOSFET E DIMENSIONAMENTO SWITCH SCC147

```
for i=2:11
    if (W31(i, 1)<W31(i, 2))
        W31(i, 3)=1;
    else
        W31(i, 3)=0;
    end
end

%%
% 8/4
a1=[1 0 0 0 0 0 1 0 0 0 1]; %sw1 sw2 sw3 sw4 sw5 sw6 sw7 sw8
sw9 sw10 sw11
a2=[0 0 1 0 1 0 0 0 0 0 1];
% v1=[0.001 0.6 0.6 0.001 0 0.3 0 0.001 0.3 0.001 0];
% v2=[0.6 0.001 0.001 0.6 0 0.3 0 0.001 0.3 0.001 0];
eta84=eta;
R_FSL= RL*(1-eta84)/eta84;
G=[0 0 0 0 0 0 0 0 0 0 0];
sum=0;
for i=1:11
    sum=sum+a1(i)+a2(i);
end
Gtot=sum.*sum.*2/R_FSL;
for i=1:11
    G(i)=(a1(i)+a2(i)).* Gtot./sum;
end
R84=1./G;
% Rsw1=R84(1)+R84(7)+R84(11);
% Rsw2=R84(3)+R84(5)+R84(11);
% y=(Rsw1+Rsw2)/0.5;
Wn=[0 0 0 0 0 0 0 0 0 0 0]'; Wnb=Wn; Wp=Wn;
k84=[12 20 1 20 7 20 7 20 20 20 7]'; %indici da usare in
base alle tensioni
% di source: 20 indica che non viene usato
for i=1:11
    a=k84(i);
    if ((a1(i)+a2(i))~=0)
        Wn(i)=k_nmos(a)/R84(i);
        Wp(i)=k_pmos(13-a)/R84(i);
        Wnb(i)=k_nmosb(a)/R84(i);
    end
end
W84=zeros(11, 3);
for i=1:11
    if ((a1(i)+a2(i))~=0)
        if (i==1)
            W84(i, 1)=Wp(i);
        elseif (i==4)
            W84(i, 1)=Wnb(i);
        end
    end
end
```

```

        else
            W84(i, 1)=Wn(i);
        end
    end
end
for i=1:11
    if ((a1(i)+a2(i))~=0)
        if (i==4)
            W84(i, 2)=Wnb(i);
        else
            W84(i, 2)=Wp(i);
        end
    end
end
W84(1, 3)=0;
for i=2:11
    if (W84(i, 1)<W84(i, 2))
        W84(i, 3)=1;
    else
        W84(i, 3)=0;
    end
end

%%
% 5/2
a1=[1 0 0 0 0 0 1 0 0 0 1]; %sw1 sw2 sw3 sw4 sw5 sw6 sw7 sw8
sw9 sw10 sw11
a2=0.5.*[0 0 0 0 0 1 0 0 1 1 0];
a3=0.5.*[0 1 1 0 0 0 0 1 0 0 1];
a4=0.5.*[0 0 1 0 1 0 0 0 0 1 0];

eta52=eta;
R_FSL= RL*(1-eta52)/eta52;
G=[0 0 0 0 0 0 0 0 0 0 0];
sum=0;
for i=1:11
    sum=sum+a1(i)+a2(i)+a3(i)+a4(i);
end
Gtot=sum.*sum.*2/R_FSL;
for i=1:11
    G(i)=(a1(i)+a2(i)+a3(i)+a4(i)).*Gtot./sum;
end
R52=1./G;
% Rsw1=R52(1)+R52(7)+R52(11);
% Rsw3=R52(2)+R52(3)+R52(8)+R52(10);
% Rsw2=R52(6)+R52(9)+R52(10);
% Rsw4=R52(3)+R52(5)+R52(10);
% y=(4*Rsw1+Rsw2+Rsw3+Rsw4)/(0.25*4);
Wn=[0 0 0 0 0 0 0 0 0 0 0]'; Wnb=Wn; Wp=Wn;

```

B.1. RESISTENZA DEI MOSFET E DIMENSIONAMENTO SWITCH SCC149

```
k52=[12 8 1 20 8 1 6 4 6 6 6]'; %indici da usare in base
alle tensioni di
% source: 20 indica che non viene usato
for i=1:11
    a=k52(i);
    if ((a1(i)+a2(i)+a3(i)+a4(i))~=0)
        Wn(i)=k_nmos(a)/R52(i);
        Wp(i)=k_pmos(13-a)/R52(i);
        Wnb(i)=k_nmosb(a)/R52(i);
    end
end
W52=zeros(11, 3);
for i=1:11
    if ((a1(i)+a2(i)+a3(i)+a4(i))~=0)
        if (i==1)
            W52(i, 1)=Wp(i);
        elseif (i==4)
            W52(i, 1)=Wnb(i);
        else
            W52(i, 1)=Wn(i);
        end
    end
end
for i=1:11
    if ((a1(i)+a2(i)+a3(i)+a4(i))~=0)
        if (i==4)
            W52(i, 2)=Wnb(i);
        else
            W52(i, 2)=Wp(i);
        end
    end
end
W52(1, 3)=0;
for i=2:11
    if (W52(i, 1)<W52(i, 2))
        W52(i, 3)=1;
    else
        W52(i, 3)=0;
    end
end
%%
% 8/3 -> tolta la vecchia fase 4
%   sw1 sw2 sw3 sw4 sw5 sw6 sw7 sw8 sw9 sw10 sw11
a1=[1/3 0 0 1/3 0 0 0 1/3 0 0 1/3];
a2=[0 1 1 0 0 0 0 1 0 0 1];
a3=[4/3 0 0 0 0 0 4/3 0 0 4/3 0];
a4=[0 0 0 0 0 2/3 0 0 2/3 0 2/3];

eta83=eta;
```

```

R_FSL= RL*(1-eta83)/eta83;
G=[0 0 0 0 0 0 0 0 0 0 0];
sum=0;
for i=1:11
    sum=sum+a1(i)+a2(i)+a3(i)+a4(i);
end
Gtot=sum.*sum.*2/R_FSL;
for i=1:11
    G(i)=(a1(i)+a2(i)+a3(i)+a4(i)).*Gtot./sum;
end
R83=1./G;
% Rsw1=R83(1)+R83(7)+R83(11);
% Rsw3=R83(2)+R83(3)+R83(8)+R83(10);
% Rsw2=R83(6)+R83(9)+R83(10);
% Rsw4=R83(3)+R83(5)+R83(10);
% y=(4*Rsw1+Rsw2+Rsw3+Rsw4)/(0.25*4);
Wn=[0 0 0 0 0 0 0 0 0 0 0]'; Wnb=Wn; Wp=Wn;
k83=[12 8 1 7 7 1 7 4 6 5 6]'; %indici da usare in base
alle tensioni di
% source: 20 indica che non viene usato
for i=1:11
    a=k83(i);
    if ((a1(i)+a2(i)+a3(i)+a4(i))~=0)
        Wn(i)=k_nmos(a)/R83(i);
        Wp(i)=k_pmos(13-a)/R83(i);
        Wnb(i)=k_nmosb(a)/R83(i);
    end
end
W83=zeros(11, 3);
for i=1:11
    if ((a1(i)+a2(i)+a3(i)+a4(i))~=0)
        if (i==1)
            W83(i, 1)=Wp(i);
        elseif (i==4)
            W83(i, 1)=Wnb(i);
        else
            W83(i, 1)=Wn(i);
        end
    end
end
for i=1:11
    if ((a1(i)+a2(i)+a3(i)+a4(i))~=0)
        if (i==4)
            W83(i, 2)=Wnb(i);
        else
            W83(i, 2)=Wp(i);
        end
    end
end
end

```

B.1. RESISTENZA DEI MOSFET E DIMENSIONAMENTO SWITCH SCC151

```
W83(1, 3)=0;
for i=2:11
    if (W83(i, 1)<W83(i, 2))
        W83(i, 3)=1;
    else
        W83(i, 3)=0;
    end
end
%%
% creo vettore finale
% Sw1 pMOS tutti gli altri li voglio nMOS
W=zeros(11,1);
k=zeros(5,1);
for i=1:11
    if (i==1)
        k(1,1)=W82(1,2); k(2,1)=W31(1,2); k(3,1)=W52(1,2);
        k(4,1)=W84(1,2);
        k(5,1)=W83(1,2);
    elseif (i==2)
        W31(i,1)=0; %nel caso del convertitore 3/1 sarebbe
        % usare un pMOS perche' nMOS troppo grande. Scarto
        % visto che per gli altri convertitori conviene un nMOS
        k(1,1)=W82(i,1); k(2,1)=W31(i,1); k(3,1)=W52(i,1);
        k(4,1)=W84(i,1);
        k(5,1)=W83(i,1);
    else
        k(1,1)=W82(i,1); k(2,1)=W31(i,1); k(3,1)=W52(i,1);
        k(4,1)=W84(i,1);
        k(5,1)=W83(i,1);
    end
    W(i)=max(k);
end
% estrapolo le proporzioni
w=min(W);
r=W./w;
% w=350e-6;
% W=r.*w;
%da ispezione
R=[R83(1) R83(2) R82(3) R82(4) R52(5) R82(6) R83(7) R82(8)
    R82(9) R82(10) R83(11)]';

%% Ricavo il valore di resistenza dei mos per ogni convertitore
% 8/2
R82_opt=zeros(11,1);
for i=1:11
    if (k82(i)~=20)
        a=k82(i);
```

```

        if (i==1)
            R82_opt(i)=k_pmos(13-a)/W(i);
        elseif (i==4)
            R82_opt(i)=k_nmosb(a)/W(i);
        else
            R82_opt(i)=k_nmos(a)/W(i);
        end
    else
        R82_opt(i)=1e9;
    end
end
% 3/1
R31_opt=zeros(11,1);
for i=1:11
    if (k31(i)~=20)
        a=k31(i);
        if (i==1)
            R31_opt(i)=k_pmos(13-a)/W(i);
        elseif (i==4)
            R31_opt(i)=k_nmosb(a)/W(i);
        else
            R31_opt(i)=k_nmos(a)/W(i);
        end
    else
        R31_opt(i)=1e9;
    end
end
% 8/3
R83_opt=zeros(11,1);
for i=1:11
    if (k83(i)~=20)
        a=k83(i);
        if (i==1)
            R83_opt(i)=k_pmos(13-a)/W(i);
        elseif (i==4)
            R83_opt(i)=k_nmosb(a)/W(i);
        else
            R83_opt(i)=k_nmos(a)/W(i);
        end
    else
        R83_opt(i)=1e9;
    end
end
% 5/2
R52_opt=zeros(11,1);
for i=1:11
    if (k52(i)~=20)
        a=k52(i);
        if (i==1)

```

B.1. RESISTENZA DEI MOSFET E DIMENSIONAMENTO SWITCH SCC153

```
        R52_opt(i)=k_pmos(13-a)/W(i);
    elseif (i==4)
        R52_opt(i)=k_nmosb(a)/W(i);
    else
        R52_opt(i)=k_nmos(a)/W(i);
    end
end
else
    R52_opt(i)=1e9;
end
end
% 8/4
R84_opt=zeros(11,1);
for i=1:11
    if (k84(i)~=20)
        a=k84(i);
        if (i==1)
            R84_opt(i)=k_pmos(13-a)/W(i);
        elseif (i==4)
            R84_opt(i)=k_nmosb(a)/W(i);
        else
            R84_opt(i)=k_nmos(a)/W(i);
        end
    else
        R84_opt(i)=1e9;
    end
end
end

% Salvataggio dei vettori resistenza e dimensione ottimale switch
save('R_optimized.mat','R82_opt','R31_opt','R83_opt',
'R52_opt','R84_opt','eta');
save('W_opt.mat','W','r','w','eta');
```

B.2 Resistenze del gear control

```
%% Dimensionamento partitore resistivo del Gear Control
clear all; clc; close all; format shortEng;

Vref=0.6; Pload=1e-3; Pdiss=0.005*Pload; %il sistema si
    autoalimenta
Rsum=Vref*Vref/Pdiss;
I=Vref/Rsum;

%soglia inferiore 0.3V
V=0.3; R1=V/I
%soglia 0.412V
V=0.412; R2=V/I-R1
%soglia 0.465V
V=0.465; R3=V/I-R1-R2
%soglia 0.4899V
V=0.4899; R4=V/I-R1-R2-R3
%soglia 0.6V
V=0.6; R5=V/I-R1-R2-R3-R4

R6=Rsum-(R1+R2+R3+R4+R5)
```


B.3 Modelli a regime dei convertitori

```

%% FSL

%% 8/4
clear all; close all; clc;
Vout=1.2; RL=1440; D84=0.5; % [0.05 0.1 0.15 0.2 0.25 0.3 0.35 0.4
0.45 0.5];
%D84=[0.025 0.1375 0.25 0.3625];
Rsw1=3.147; Rsw2=4.75; Vin=0.601; % 1.2*4/8;

for i=1:2000
    Vout=2*Vin-Vout./(RL.*D84.*(D84)).*(Rsw1+D84.*(Rsw2-Rsw1));
    i;
end
Rout84=(1.2-Vout).*RL./Vout;
% figure; plot(D, Vout); grid on;
% xlabel('Duty Cycle'); ylabel('Vout');
% title('Convertitore 8/4');

i1=Vout./RL; i2=i1;
VC1=Vin-i2*Rsw2;
a1=1
a2=-1
r=(a1*a1*3+a2*a2*3)./D84;

converter84=Vout./Vin;
Vin_v=0.3:0.001:0.6;
figure(2);
for i=1:length(converter84)
    Vout84=converter84(i).*Vin_v;
    plot(Vin_v, Vout84); hold on;
end
grid on;

Rout84=(1.2-Vout).*RL./Vout;
eta84=RL./(RL+Rout84);
figure(1); plot(converter84, eta84); hold on; grid on;
Vin_discreto84=1.2./converter84;
figure(3); plot(Vin_discreto84, eta84); hold on; grid on;

%% 8/2
clc;
Vout=1.2; RL=1440; D82=0.3; % [0.05 0.1 0.15 0.2 0.25 0.3 0.3333];
%D82=[0.016 0.05 0.083 0.116 0.15 0.183 0.25 0.283];
D2=D82; D3=D82;
Rsw1=3.014; Rsw2=2.998; Rsw3=1.512; Vin=0.301; % 1.2*2/8;

```

```

for i=1:2000
    Vout=4.*Vin-Vout./(RL.*D82).*(Rsw1+Rsw2.*D82./D2+4
    .*Rsw3.*D82./D3);
    i;
end
VC2=Vin-Rsw3.*2.*D82./D3.*Vout./(RL.*D82);
VC1=Vin+VC2-Rsw2.*Vout./(RL.*D82).*D82./D2;

% figure(1);
% plot3(D1, D2, Vout); grid on;
% xlabel('D_1'); ylabel('D_2'); zlabel('Vout');
% title('Convertitore 8/2'); hold on;

converter82=Vout./0.3;
Vin_v=0.3:0.001:0.6;
figure(2);
for i=1:length(converter82)
    Vout82=converter82(i).*Vin_v;
    plot(Vin_v, Vout82); hold on;
end
grid on;

Rout82=(1.2-Vout).*RL./Vout;
eta82=RL./(RL+Rout82);
figure(1); plot(converter82, eta82); hold on;
Vin_discreto82=1.2./converter82;
figure(3); plot(Vin_discreto82, eta82); hold on; grid on;
%% 3/1
clc;
Vout=1.2; RL=1440; D31=0.3; % [0.05 0.1 0.15 0.2 0.25 0.3 0.3333];
%D31=[0.0166 0.09166 0.166 0.2416];
D2=D31; D3=D31;
Rsw1=2.16; Rsw2=6.52; Rsw3=1.655; Vin=0.413; % 1.2*1/3;

for i=1:2000
    Vout=3*Vin-Vout./(RL.*D2).*(Rsw1.*D2./D31+Rsw2+Rsw3.*D2./D3);
end
i2=Vout./(RL.*D2);
VC2=Vin-Rsw1.*i2.*D2./D31;
VC1=Vin+VC2-Rsw3.*i2.*D2./D3;
Rout31=(1.2-Vout).*RL./Vout;
% figure;
% plot3(D1, D2, Vout); grid on;
% xlabel('D_1'); ylabel('D_2'); zlabel('Vout');
% title('Convertitore 3/1'); hold on;

converter31=Vout./0.4;
Vin_v=0.3:0.001:0.6;
figure(2);

```

```

for i=1:length(converter31)
    Vout31=converter31(i).*Vin_v;
    plot(Vin_v, Vout31); hold on;
end
grid on;

Rout31=(1.2-Vout).*RL./Vout;
eta31=RL./(RL+Rout31);
figure(1); plot(converter31, eta31);hold on;
Vin_discreto31=1.2./converter31;
figure(3); plot(Vin_discreto31, eta31);hold on; grid on;
%% 5/2
clc;
Vout=1.2; RL=1440; D52=0.25;%D52=[0.05 0.1 0.15 0.2 0.25];
%D52=[0.0125 0.06875 0.125 0.18125];
D2=(D52); D3=D52; D4=D52;
%Rsw1=3; Rsw2=4; Rsw3=3; Rsw4=3;
Rsw1=2.756;%R52_opt(1)+R52_opt(7)+R52_opt(11);
Rsw2=1.817;%R52_opt(6)+R52_opt(9)+R52_opt(10);
Rsw3=3.655;%R52_opt(2)+R52_opt(3)+R52_opt(8)+R52_opt(10);
Rsw4=4.057;%R52_opt(3)+R52_opt(5)+R52_opt(10);
Vin=0.491;%1.2*2/5;

for i=1:2000
    Vout=5/2*Vin - Vout./RL.*(Rsw1.*D4.*D2.*D3 + Rsw2.*D52.*
D3.*D4 + Rsw3.*D52.*D2.*D4 + Rsw4.*4.*D52.*D2.*D3)./(4.*D52.*
D2.*D3.*D4);
end
Rout52=(1.2-Vout).*RL./Vout;
i4=Vout/(RL*D52); i1=i4/2; i2=i1; i3=i1;
VC2=Vin-i4*Rsw1;
VC1=1.5*Vin-0.5*i1*Rsw1-0.5*i2*Rsw2-0.5*i3*Rsw3;
VC3=VC1-Vin+i3*Rsw3;
% figure;
% plot3(D1, D4, Vout); grid on;
% xlabel('D_1'); ylabel('D_4'); zlabel('Vout');
% title('Convertitore 5/2'); hold on;

converter52=Vout./0.48;
Vin_v=0.3:0.001:0.6;
figure(2);
for i=1:length(converter52)
    Vout52=converter52(i).*Vin_v;
    plot(Vin_v, Vout52); hold on;
end
grid on;

Rout52=(1.2-Vout).*RL./Vout;
eta52=RL./(RL+Rout52);

```

```

figure(1); plot(converter52 , eta52);hold on;
Vin_discreto52=1.2./converter52;
figure(3); plot(Vin_discreto52 , eta52);hold on; grid on;
%% 8/3
clc; clear all;
Vout=1.2; RL=1440; D83=0.25; %[0.05 0.1 0.15 0.2];
%D83=[0.01 0.055 0.1 0.145];
Vin=0.466; %1.2*3/8;
Rsw1=3.46; Rsw2=4.053; Rsw3=2.367; Rsw4=2.215;

for i=1:2000
    Vout=8*Vin/3-2*Vout*(Rsw1+16*Rsw3+6*Rsw4+7*Rsw2)/(9*RL);
end
Rout83=(1.2-Vout).*RL./Vout;
i1=-2*Vout/(3*RL); i2=-3*i1; i3=-4*i1; i4=-2*i1;
VC1=4/3*Vin+i1/3*(2*Rsw1+8*Rsw3+2*Rsw4+9*Rsw2);
VC2=2/3*Vin+i1/3*(Rsw1+4*Rsw3+4*Rsw4);
VC3=1/3*Vin-i1/3*(Rsw1+4*Rsw3-2*Rsw2);
% converter83=Vout./Vin;
% Vin_v=0.3:0.001:0.6;
% figure(2);
% for i=length(converter83)
%     Vout83=converter83(i).*Vin_v;
%     plot(Vin_v, Vout83); hold on;
% end
% axis([0.3 0.6 1 1.5]);
% xlabel('Vin [V]'); ylabel('Vout [V]');
% title('Effetto del duty cycle sul fattore di conversione');
%
% Rout83=(1.2-Vout).*RL./Vout;
% eta83=RL./(RL+Rout83);
% figure(1); plot(converter83 , eta83);hold on;
% xlabel('Rapporto di conversione'); ylabel('\eta');
% title('Efficienza in funzione del rapporto di conversione');
% Vin_discreto83=1.2./converter83;
% figure(3); plot(Vin_discreto83 , eta83);hold on; grid on;
% xlabel('Vin [V]'); ylabel('\eta');
% title('Efficienza in funzione della tensione di ingresso');

%% decisione range tensioni ingresso
clc;
%la media deve fare 1.2V
%cambio 8/2 <=> 3/1
Vchange82_31=2*1.2/(converter82(1)+converter31(end));

%cambio 3/1 <=> 8/3
Vchange31_83=2*1.2/(converter31(1)+converter83(end));

%cambio 8/3 <=> 5/2

```

```

Vchange83_52=2*1.2/(converter83(1)+converter52(end));

%cambio 5/2 <-> 8/4
Vchange52_84=2*1.2/(converter52(1)+converter84(end));

%% Riproduco la tensione di uscita
V_in=[Vin_discreto84 Vchange52_84 Vchange52_84 Vin_discreto52
Vchange83_52 Vchange83_52 Vin_discreto83 Vchange31_83
Vchange31_
83 Vin_discreto31 Vchange82_31 Vchange82_31 Vin_discreto82
0.3];
gear=[converter84 converter84(end) converter52(1) converter52
converter52(end) converter83(1) converter83 converter83(end)
converter31(1) converter31 converter31(end) converter82(1)
converter82 converter82(end)];
V_out=gear.*V_in;
figure(4); plot(V_in, V_out);
axis([0.3 0.6 1.1 1.3]);
xlabel('Vin [V]'); ylabel('Vout [V]'); grid on; title('Uscita
con ripple minimo');

```


Bibliografia

- [1] Sam Ben-Yaakov, Alexander Kushnerov, *Algebraic Foundation of Self Adjusting Switched Capacitors Converters*, in IEEE Energy Conversion Congress and Expo (ECCE), 2009, pp. 1582-1589.
- [2] Sam Ben-Yaakov, Alexander Kushnerov, *Unified Algebraic Synthesis of Generalized Fibonacci Switched Capacitor Converters*, 2012, pp. 774-778.
- [3] Sam Ben-Yaakov, Alexander Kushnerov, *The best of both worlds: Fibonacci and binary switched capacitor converters combined*, 2012, pp. 774-778.
- [4] Michael D. Seeman, *Analytical and Practical Analysis of Switched-Capacitor DC-DC Converters*, Master of Science in Electrical Engineering and Computer Science University of California at Berkeley.
- [5] Alexander Kushnerov, *High-efficiency self-adjusting switched capacitor DC-DC converter with binary resolution*, Master of Science in Electrical Engineering and Computer Science Ben-Gurion University of the Negev.
- [6] Sam Ben-Yaakov, Michael Evzelman, *Generic and Unified Model of Switched Capacitor Converters*, 2009, pp. 3501-3508.
- [7] Michael D. Seeman, *A Design Methodology for Switched-Capacitor DC-DC Converters*, Doctor of Philosophy in Electrical Engineering and Computer Sciences, University of California at Berkeley, 2009.
- [8] Weidong Liu, Xiaodong Jin, James Chen, Min-Chie Jeng, Zhihong Liu, Yuhua Cheng, Kai Chen, Mansun Chan, Kelvin Hui, Jianhui Huang, Robert Tu, Ping K. Ko and Chenming Hu *BSIM3v3.2.2 MOSFET Model, user's manual*, Department of Electrical Engineering and Computer Sciences University of California, Berkeley, CA 94720, 1999.
- [9] Sam Ben-Yaakov, Alexander Kushnerov, *Algebraic Synthesis of Fibonacci Switched Capacitor Converters*.

- [10] Sam Ben-Yaakov, Alexander Kushnerov, *A Least Squares Method Applied to Multiphase Switched Capacitor Converters*, 2015.
- [11] Sam Ben-Yaakov, <http://www.ee.bgu.ac.il/~pel/seminars/APEC09.pdf>, 2009.
- [12] Jan M. Rabaey, Anantha Chandrakasan, Bora Nikolić, *Circuiti Integrati Digitali, L'ottica del progettista*, seconda edizione, edizione italiana a cura di Andrea Cestere e Andrea Gerosa, Università di Padova, edizione Pearson Prentice Hall, 2005.

Ringraziamenti

Al termine di questo mio lavoro desidererei ringraziare i professori Andrea Gerosa, Andrea Neviani, Andrea Bevilacqua e Giorgio Spiazzi per il loro aiuto durante questi 7 mesi di tesi. Vorrei ringraziare anche tutti i ragazzi del laboratorio ICARUS per aver reso meno pesante il lavoro, ma vorrei citare anche i ragazzi del laboratorio BioDevices ed in particolare il dottorando Giulio Rosati per le sue immancabili pause caffè nel nostro laboratorio.

Un ringraziamento particolare va a Davide Dal Bianco, Filippo Dalla Valle e Davide Broccardo per aver condiviso con me le gioie e le difficoltà di design di questo piccolo chip.

Un lavoro di tesi però non si rinchiude nel quaderno degli appunti o nella schermata del simulatore, ma invade anche tutti gli aspetti della vita quotidiana e per questo devo ringraziare tutti coloro che mi hanno aiutato sotto questo punto di vista: gli amici di banda e di fanfara per le goliardiche serate, agli amici di sempre che sono la mia seconda famiglia e tutti i miei parenti, in particolare mio fratello, mia madre e mio padre.