

POLITECNICO DI TORINO

Corso di Laurea in Ingegneria Elettronica

Tesi di Laurea Magistrale

**Analisi di un riferimento di tensione
virtuale basato su un oscillatore
controllato in tensione in tecnologia
CMOS 65nm**



Relatore

prof. Paolo Stefano Crovetto

Corelatore:

ing. Pedro Filipe Leite Correia de Toledo

Studente

Anthony DI LABBIO

matricola: 255367

Aprile 2021

Abstract

Although geometrical scaling has enabled an enhancement in terms of cost, performance, silicon area and power management in digital circuits, the same is not true for analog ICs. For this reason, there is an increasing research interest in new solutions that exploit both digital design-flow and digital techniques to implement analog functions to take advantage of geometrical scaling. In this scenario, a voltage-controlled-oscillator (VCO)-based virtual voltage reference, based on digital processing of the information, is analyzed in this thesis. For this purpose, an integrated VCO in a 65nm technology has been designed and simulated in the Cadence environment, and has been tested over different PVT (Power supply, Temperature and Process) conditions and simulation data have been processed in the MATLAB tool to estimate the performance of a virtual reference structure. Based on this analysis, the main performance benchmarks of a VCO-based virtual reference have been estimated, also discussing the impact of the main design parameters on the virtual reference performance.

Ringraziamenti

Vorrei ringraziare il prof. Paolo S. Croveti per la sua professionalità e per la disponibilità mostrata durante tutto il lavoro di tesi. Un ringraziamento è rivolto anche all'ing. Pedro de Toledo per i suoi consigli e il suo supporto. Un grazie anche al dott. Roberto Rubino per le molteplici risposte alle mie domande e al dott. Andrea Malacrino per aver condiviso parte del viaggio con me.

Un grazie speciale va ai miei genitori per i sacrifici che mi hanno permesso di raggiungere questo traguardo. Un grazie a Chiara e a tutta la mia famiglia.

Ringrazio il mio amico e coinquilino Francesco per la condivisione di gioie e dolori durante tutto il percorso universitario.

Ultimo, ma non per ordine di importanza, un grazie alla mia fidanzata Gabriella per il suo sorriso che ha reso meno bui quei momenti in cui non bastava accendere la luce.

Indice

1	Introduzione	5
1.1	Scopo della tesi	8
2	Riferimenti di tensione	11
2.1	Definizione e applicazioni	11
2.2	Figure di merito	14
2.3	Pseudo-riferimenti di tensione	15
2.3.1	Partitore resistivo di tensione	16
2.3.2	Divisore di tensione MOS	17
2.4	Classificazione dei riferimenti di tensione	20
2.5	Riferimenti basati sulle proprietà della giunzione pn	20
2.5.1	Riferimenti Bandgap	22
2.5.2	Bandgap alternativi	25
2.5.3	Riferimento bandgap a bassa tensione	27
2.6	Riferimento di tensione per sistemi ultra-low power	28
2.6.1	Riferimento di tensione 2T	28
3	Riferimento di tensione virtuale	33
3.1	Riferimento bandgap di Kuijk	33
3.2	Struttura hardware del riferimento di tensione virtuale	37
3.3	Procedura di compensazione	39
3.4	Riferimento virtuale	41
3.5	Riferimento virtuale digitale generalizzato	42
3.6	Indici di prestazione sperimentali	43
4	Riferimento di tensione virtuale VCO-based	45
4.1	Struttura	45
4.2	CSVCO	47
4.2.1	Progettazione	49
4.2.2	Simulazioni	50
4.3	VCO-based ADC	55
4.3.1	Sottosistemi del riferimento virtuale VCO-based	55
4.4	Analisi della struttura	57
4.4.1	Riferimento virtuale	58
4.5	Procedura di compensazione	59

4.6	Organizzazione delle simulazioni	62
4.6.1	Matrice delle frequenze	63
4.6.2	Flow-Chart e algoritmo	65
5	Risultati e stima delle performance	67
5.1	Variazioni della tensione di alimentazione	67
5.2	Variazione della temperatura	72
5.3	Variazioni di processo	80
5.4	Studio della risoluzione del DAC e del periodo dell'oscillatore locale	82
5.4.1	DAC	82
5.4.2	Oscillatore locale	86
5.4.3	Combinazione del DAC e dell'oscillatore locale	88
5.5	Consumo di potenza	89
6	Conclusioni	91
A	Codice MATLAB	93
B	Codice MATLAB con compensazione empirica del numero A	109

Capitolo 1

Introduzione

Per Internet of Things (IoT), o Internet delle Cose, si intende quella rete in cui ogni oggetto della vita quotidiana è connesso con gli altri oggetti ed acquisisce un'identità digitale. Di fatto l'IoT si pone come anello di congiunzione tra il mondo dell'Internet, attraverso la sua rete pervasiva, e il mondo fisico [1]. Molto spesso, l'IoT viene associata a quelle applicazioni in cui oggetti, come le automobili, i droni, gli smartphone, gli apparecchi intelligenti riescono a comunicare tra di loro, ossia scambiare dati utili, tramite una rete wireless di sensori (WSN, Wireless Sensor Network) che permettono di misurare, elaborare e capire le informazioni provenienti dal mondo fisico [2].

In particolare, la suddetta rete di sensori coincide con la rete dei nodi IoT, a loro volta formati da circuiti integrati (ICs) miniaturizzati. Alle stringenti specifiche di progetto che riguardano i nodi IoT in termini di volume fisico ed autonomia energetica, corrispondono ulteriori vincoli riguardanti gli ICs. L'intero nodo IoT, infatti, si basa su sistemi Ultra-Low-Power (ULP), in cui la tecnologia dominante risulta essere quella CMOS. Risulta particolarmente arduo progettare interfacce analogiche, sempre necessarie per scambiare le informazioni, in quei nodi IoT in cui sono richiesti volumi fino alla scala millimetrica, consumi di potenza prossima ai nW e dal costo ridotto. Il motivo principale risiede nell'incapacità, da parte dei circuiti analogici, di trarre beneficio dallo scaling geometrico, tipico della tecnologia CMOS [3]. Nonostante la riduzione della lunghezza di canale del singolo transistor comporti un aumento della velocità intrinseca del dispositivo (f_T), il guadagno intrinseco risulta fortemente degradato. A titolo di esempio, passando dal nodo 500 nm al nodo 22 nm, il guadagno intrinseco del transistor diminuisce da 180 V/V a 6V/V, ossia di 30 volte [4]. Oltre alla degradazione della caratteristica di matching tra i dispositivi con dimensioni minime, che comporta delle performance analogiche fortemente dipendenti dal layout, la riduzione della tensione di alimentazione, che ha seguito negli anni la miniaturizzazione della lunghezza minima di canale (L_{min}) e che è risultata indispensabile per controbilanciare l'aumento della densità di potenza nei circuiti digitali, ha prodotto una riduzione del rapporto segnale-rumore (SNR), rendendo molto arduo il compito di progettare circuiti analogici ad alta linearità e bassa distorsione [5]. Di fatto, i vantaggi introdotti dallo scaling geometrico, tra cui la riduzione dei parassiti, l'area occupata sul silicio, le performance e l'effort di progettazione, vengono controbilanciati da un deterioramento delle performance analogiche che rappresentano un fattore limitante nello sviluppo dei circuiti integrati. Infatti, se da un lato l'area del blocco digitale SRAM ha visto ridurre

le proprie dimensioni, lo stesso trend non è stato riscontrato nei classici blocchi analogici come l'amplificatore operazionale di transconduttanza (OTA) e il riferimento di tensione Bandgap [3]. Nel corso degli anni, come si nota nella figura 1.3, si è sviluppato un filone di ricerca che ha l'obiettivo di implementare i blocchi analogici tradizionali attraverso l'utilizzo di circuiti digitali [3].

Le tecniche automatiche di design digitale, basate sull'utilizzo di porte logiche, permettono sia di restringere i tempi di progettazione, garantendo un time-to-market minore, sia di sintetizzare e validare facilmente blocchi analogici in cui l'informazione viene trasferita dal dominio dell'ampiezza a quello del tempo. L'idea principale consiste in una codifica temporale del segnale analogico, rappresentato da un'onda quadra modulata, in cui l'informazione risiede nei fronti del segnale e non nell'ampiezza [6]. Questo tipo di segnale, caratteristico dei circuiti digitali, gode della robustezza nei confronti del rumore e della distorsione. Un ulteriore vantaggio di un circuito analogico, realizzato e basato su tecniche digitali, consiste nell'incremento delle performance dovuto al netto decremento dei ritardi delle porte logiche all'avanzare dello scaling geometrico.

Un generico blocco analogico può essere trasformato in un equivalente digitale, come mostrato nella figura 1.1:

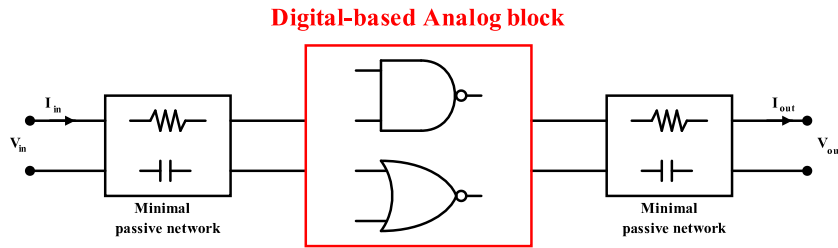


Figura 1.1. Diagramma a blocchi di un sistema analogico implementato con circuiti digitali [3]

I circuiti digitali, che implementano le funzioni analogiche tramite le porte e tecniche digitali, si interfacciano con l'esterno tramite reti passive minime, e non sempre necessarie, che raccolgono informazioni da segnali, ossia da tensioni e correnti a banda limitata e ad ampiezza finita, e che forniscono in uscita segnali con le stesse caratteristiche e dotati di un grado di accuratezza prefissato [3].

Un esempio è rappresentato in figura 1.2, in cui si mostra un convertitore D/A a rilassamento che sfrutta una rete RC di uscita che riceve un flusso di dati digitali da un registro a scorrimento, composto da porte logiche e opportunamente controllato da un sistema digitale.

Tra i blocchi analogici presi in esame dalla ricerca, citandone alcuni, si trovano PLL completamente digitali, convertitori A/D, come gli ADC basati su oscillatori controllati in tensione (Voltage Controlled Oscillator, VCO), oscillatori, filtri, regolatori e riferimenti di tensione [3].

A titolo di esempio, nonostante siano stati progettati diversi riferimenti di tensione analogici, in tecnologia CMOS, operanti con tensioni di alimentazione ridotte, ad esempio [8] e [9], la loro accuratezza dipende essenzialmente da parametri relativi al processo tecnologico e da stringenti vincoli di matching tra i diversi componenti, che si possono soddisfare con

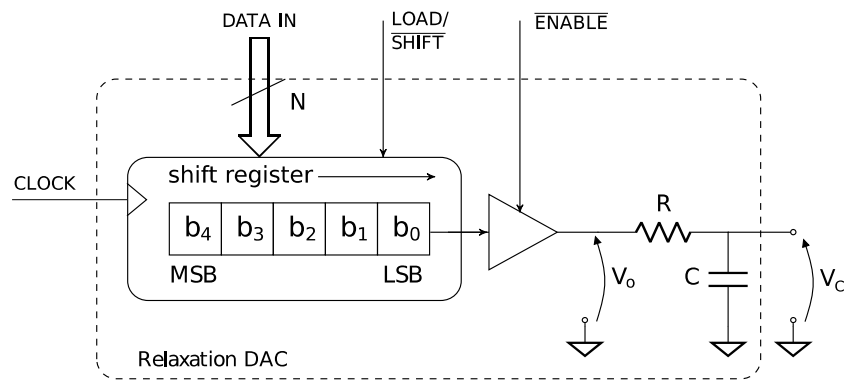


Figura 1.2. Relaxation Digital to Analog Converter [7]

costose tecniche di trimming e attraverso dispositivi con dimensioni di diversi ordini di grandezza superiori al minimo imposto dalla litografia [10]. Nonostante i circuiti analogici risultino sempre necessari, almeno nei sistemi di interfaccia con il mondo fisico analogico [11] e nonostante il fatto che la circuiteria analogica risulti, in determinate applicazioni, meno complessa, meno costosa e più performante di un sistema digitale [12], i vantaggi introdotti dallo scaling della tecnologia, in termini di costi, integrazione su silicio e di performance, generalmente, rappresentano un motivo fondamentale che sta spingendo la ricerca a progettare i circuiti integrati utilizzando un approccio di tipo digitale.

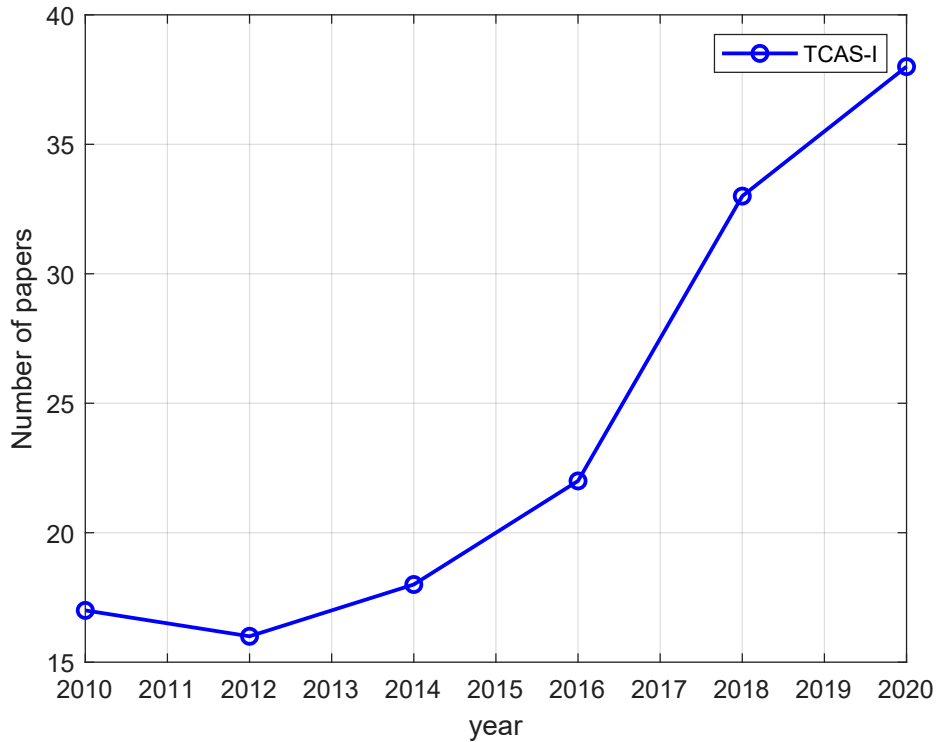


Figura 1.3. Andamento del numero di documenti di ricerca TCAS-I su sistemi analogici realizzati con tecniche digitali [3]

1.1 Scopo della tesi

L'obiettivo della tesi consiste nell'analisi del comportamento di un riferimento di tensione virtuale basato su un oscillatore controllato in tensione (ingl. Voltage Controlled Oscillator, VCO), basato su tecniche digitali e ideato per essere inglobato in un sistema IoT. Il lavoro svolto comprende la progettazione di un oscillatore CSVCO integrato, implementato in ambiente Cadence in tecnologia CMOS 65 nm, le simulazioni al variare della tensione di alimentazione, della temperatura e dei parametri di processo e l'elaborazione dei dati,

effettuata dal software di calcolo MATLAB, che simula il comportamento del riferimento di tensione virtuale. Di seguito viene presentata la struttura della tesi.

Nel secondo capitolo viene fornita la definizione di riferimento di tensione, vengono discusse le principali figure di merito e vengono riportati e descritti, dopo una breve classificazione, alcuni tipici esempi di riferimenti integrati, evidenziandone le caratteristiche. Al termine del capitolo, inoltre, viene presentata una soluzione circuitale adatta per applicazioni ULP. Nel terzo capitolo, a partire dall'analisi del riferimento di tensione bandgap di Kuijk, viene descritta e analizzata la struttura hardware del riferimento di tensione virtuale e la procedura software di compensazione attraverso la quale viene calcolato il riferimento virtuale. Infine, vengono menzionati i principali indici di prestazione, tra cui il coefficiente di temperatura e la line sensitivity, al variare delle condizioni PVT, al fine di descrivere le performance della struttura.

Nel quarto capitolo, a partire dal riferimento di tensione virtuale, presentato nel terzo capitolo, viene mostrata la struttura del riferimento di tensione virtuale VCO-based. Lo studio teorico del sistema segue la sezione in cui vengono mostrate le scelte progettuali del CSVCO, implementato, a livello transistor, in tecnologia CMOS a 65 nm, mediante l'utilizzo del Process Design Kit di STMicroelectronics e le relative simulazioni. Viene fornita un'analisi dell'intera struttura, per evidenziarne il funzionamento, e viene presentata la procedura algoritmica di compensazione per il calcolo del riferimento virtuale. Infine, viene presentata l'organizzazione sia delle simulazioni in ambiente Cadence sia del software implementato in ambiente MATLAB, attraverso la presentazione del rispettivo flow-chart. Nel quinto capitolo vengono riportati i risultati che indicano le performance della struttura al variare delle condizioni PVT e al variare delle caratteristiche dei sottosistemi, ossia viene anche riportato uno studio del comportamento del riferimento di tensione virtuale VCO-based al variare della risoluzione del DAC e del periodo di un oscillatore locale.

Nel capitolo delle conclusioni vengono passati in rassegna i principali risultati ottenuti, definite le condizioni minime che consentono di ottenere prestazioni accettabili da parte della struttura, ponendo le basi per una futura completa implementazione hardware.

Capitolo 2

Riferimenti di tensione

In questo capitolo è riportata la definizione generale di riferimento di tensione e vengono presentate alcune applicazioni. In seguito sono descritte le principali figure di merito dei riferimenti di tensione e, inoltre, vengono illustrate alcune soluzioni circuitali, ponendo le basi per la spiegazione del riferimento di tensione virtuale nel capitolo successivo.

2.1 Definizione e applicazioni

Di fondamentale importanza, in generale, in tutti i sistemi elettronici, è la presenza di circuiti di riferimento capaci di generare tensioni e/o correnti in continua (DC) che esibiscono una scarsa dipendenza dalle variazioni della tensione di alimentazione e dalle tolleranze di fabbricazione dei processi tecnologici dei circuiti integrati e una dipendenza ben definita nei confronti delle variazioni della temperatura [13]. Il termine *riferimento* è infatti usato per indicare che la tensione/corrente possiede una precisione e una stabilità maggiore rispetto a quella di una comune sorgente [14]. In particolare il riferimento di tensione è un dispositivo presente in tutte quelle architetture che richiedono una tensione continua immune, nel caso ideale, alle variazioni di temperatura, della tensione di alimentazione e del processo (PVT-independent). A titolo di esempio, nei convertitori A/D e D/A è richiesto un riferimento di tensione per definire il range di fondo scala di ingresso o di uscita [13]. La seguente equazione, infatti, che riporta la relazione ingresso-uscita ideale di un ADC operante su N bit, con riferimento alla figura 2.1, dimostra che il numero digitale in uscita $Code$ si ottiene attraverso il rapporto tra la tensione di ingresso V_{in} e la tensione di riferimento V_{ref} :

$$Code = \frac{V_{in} \cdot 2^N}{V_{ref}} \quad (2.1)$$

Risulta di facile comprensione come una modifica del numero digitale di uscita, dovuta ad una alterazione indesiderata della tensione di riferimento, provochi degli errori di valutazione nel sistema digitale posto a valle dell'ADC. Dunque, al fine di ottenere misurazioni accurate durante il processo di conversione dei segnali, è necessario che il riferimento di tensione sia molto stabile [15].

Inoltre, come si può notare dalla figura 2.2, nei sistemi elettronici di potenza, normalmente, un regolatore (controllore), in anello chiuso, confronta l'uscita del sistema di conversione

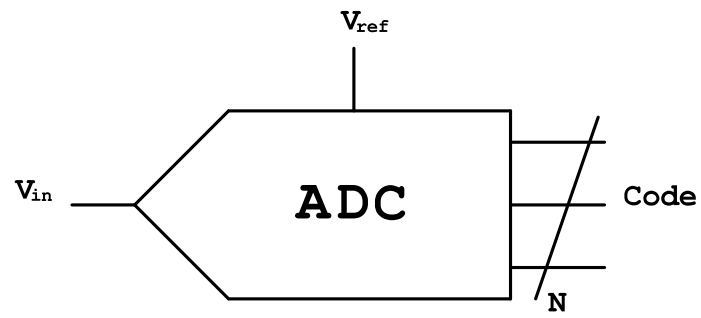


Figura 2.1. ADC e riferimento di tensione

con un valore desiderato (riferimento) e l'errore tra i due è minimizzato dal controllore stesso [16].

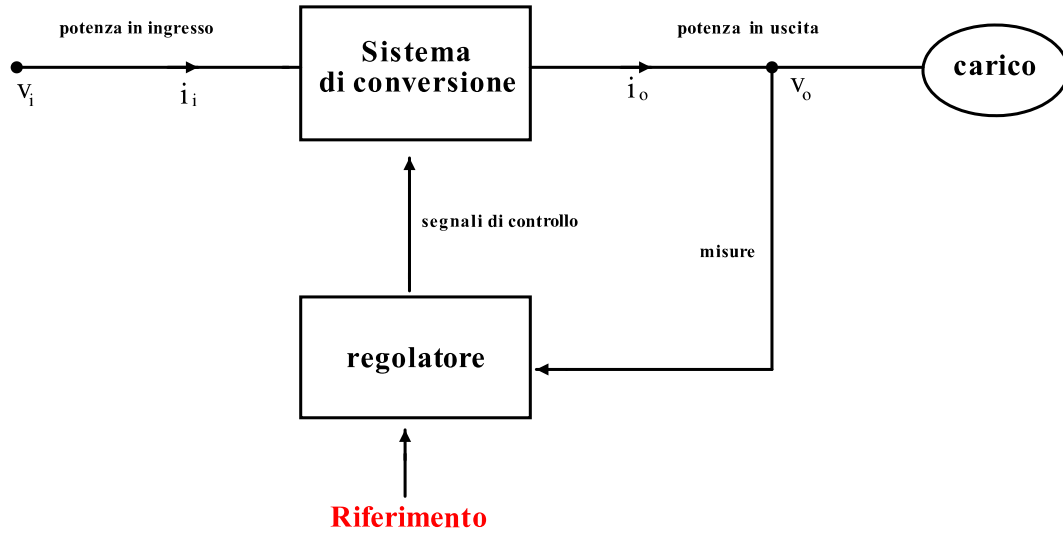


Figura 2.2. Generico sistema di potenza che impiega un riferimento [16]

Un'altra applicazione che richiede l'utilizzo di un riferimento di tensione è il regolatore di tensione [17], mostrato nella figura 2.3. Esso ha l'obiettivo di fissare la tensione di uscita V_{out} al valore di una tensione di riferimento V_{ref} attraverso un coefficiente di proporzionalità espresso come un rapporto di valori di resistenza.

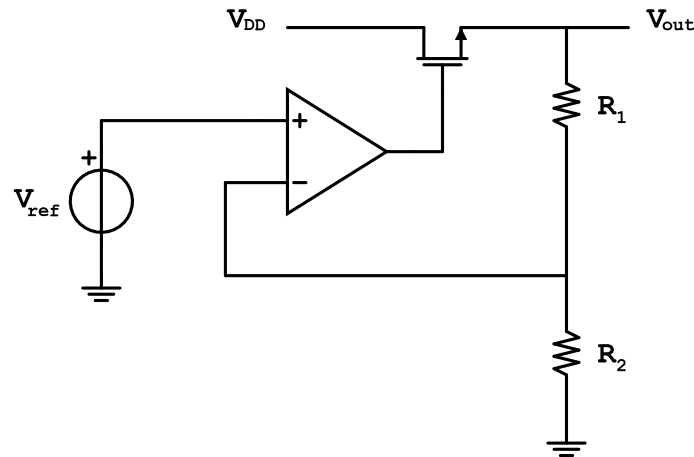


Figura 2.3. Regolatore di tensione [17]

La tensione di uscita può essere espressa infatti come:

$$V_{out} = V_{ref} \frac{R_1 + R_2}{R_2} \quad (2.2)$$

Si tratta di un circuito formato da un amplificatore operazionale, che presenta sul morsetto positivo la tensione di riferimento, seguito da un transistor in configurazione source-follower che fornisce la corrente al carico. L'accuratezza della tensione di uscita dipende dalle rispettive accuratezze del rapporto delle resistenze e del riferimento di tensione. Poiché, attraverso un' opportuna selezione dei resistori, si riesce ad ottenere un rapporto di resistenze molto accurato, la precisione finale dipende dalla stabilità della tensione di uscita del riferimento di tensione posto all'ingresso dell'operazionale [17].

2.2 Figure di merito

Di seguito vengono elencate e contestualmente discusse alcune principali figure di merito che giustificano la scelta di un particolare riferimento di tensione rispetto ad altre configurazioni.

- La regolazione di linea, o line sensitivity (LS), è quel parametro che specifica, per una data temperatura, la variazione relativa della tensione di riferimento V_{ref} rispetto alla variazione assoluta della tensione d'ingresso, identificabile con la tensione di alimentazione V_{dd} :

$$LS = \frac{V_{ref,max} - V_{ref,min}}{V_{ref,nom}} \frac{1}{V_{dd,max} - V_{dd,min}} \cdot 100 \quad [\%/V] \quad (2.3)$$

Un ottimo riferimento di tensione usato in applicazioni in cui la tensione di alimentazione varia nella sua componente continua (DC) deve presentare una LS molto ridotta, idealmente nulla. Il range operativo della tensione di alimentazione in cui si misura la line sensitivity deve essere specificato al fine di poter confrontare le diverse soluzioni circuitali tenendo conto dei valori assoluti della V_{dd} .

- Un altro parametro rilevante è il PSRR (Power Supply Rejection Ratio), ossia il rapporto di reiezione dell'alimentazione, che indica la capacità, da parte del riferimento, di sopprimere la componente alternata della tensione di alimentazione dovuta all'effetto del rumore nella rete di alimentazione, ad una determinata frequenza:

$$PSRR = 20 \times \log \frac{V_{ref,AC}(f)}{V_{dd,AC}(f)} \quad [dB] \quad (2.4)$$

Nella scelta tra due diversi riferimenti, la soluzione migliore consiste nel selezionare quello con un PSRR più negativo in dB.

- Il coefficiente di temperatura TC (temperature coefficient) indica la variazione della tensione di riferimento V_{ref} , per un certo valore nominale di V_{dd} , rispetto ad un range operativo di temperatura:

$$TC = \frac{V_{ref,max} - V_{ref,min}}{V_{ref,nom}} \frac{1}{T_{max} - T_{min}} \cdot 100 \quad [\%/^{\circ}C] \quad (2.5)$$

Come nel caso della line sensitivity, anche per il coefficiente di temperatura si ottengono ottime prestazioni se per un range esteso di valori di temperatura la variazione della tensione di riferimento risulta minima, idealmente nulla. Inoltre, alcune applicazioni, come i sensori di temperatura, richiedono una dipendenza dalla temperatura non nulla, ma ad esempio lineare (PTAT, Proportional to Absolute Temperature).

- La corrente a riposo (quiescent current) I_q che proviene dall'alimentazione rappresenta la corrente richiesta dal riferimento a regime. Il consumo di potenza P a regime si calcola come di seguito mostrato:

$$P = I_{q,nom} \cdot V_{dd,nom} \quad [W] \quad (2.6)$$

in cui $V_{dd,nom}$ rappresenta la tensione di alimentazione in condizioni nominali. Un riferimento di tensione con una contenuta corrente di riposo è desiderabile in quanto implica un consumo di potenza ridotto e dunque un tempo di vita più esteso, preferibile per le applicazioni con un limitato budget energetico.

- Il rumore, che rappresenta un fattore limitante in un qualsiasi circuito elettronico, viene calcolato e modellato come una tensione V_{RMS} , ottenuta a partire dalla densità spettrale di potenza di rumore $S^2(f)$ (espressa in V^2/Hz), come di seguito mostrato:

$$V_{RMS} = \sqrt{\int_{f_L}^{f_H} S^2(f) \cdot df} \quad [V] \quad (2.7)$$

in cui f_H-f_L rappresenta l'intervallo delle frequenze di interesse. Il rumore viene classificato in tre principali tipi differenti. Il rumore termico, dovuto al movimento casuale degli elettroni a causa della loro energia termica, è direttamente proporzionale alla temperatura e inversamente proporzionale alla radice quadrata della corrente a riposo, oltre che alla dimensione fisica totale del componente [18]. Il rumore shot, che nei MOSFET è dovuto al campo elettrico esistente tra il gate ed il canale del dispositivo, diventa importante per applicazioni che presentano correnti inferiori ai pA. Il rumore $1/f$ (flicker noise), inoltre, risulta inversamente proporzionale all'area del singolo dispositivo e il suo impatto cresce a frequenze via via più contenute[18].

- L'area occupata sul silicio rappresenta un'ulteriore figura di merito. Si preferisce progettare circuiti di dimensioni ridotte per motivi legati al costo finale dell'intero sistema che ingloba il riferimento di tensione.

2.3 Pseudo-riferimenti di tensione

I seguenti circuiti rappresentano alcuni esempi di pseudo riferimenti di tensione e sono riportati con lo scopo di illustrare, attraverso l'utilizzo di equazioni matematiche, le problematiche relative alla loro semplicità strutturale. Si tratta infatti di soluzioni non adatte ad applicazioni che richiedono elevate accuratezze.

2.3.1 Partitore resistivo di tensione

Nella figura 2.4 sono riportati due resistori in serie R_1 ed R_2 alimentati dalla tensione di alimentazione V_{dd} .

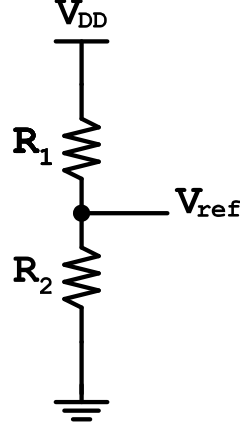


Figura 2.4. Voltage Divider

La pseudo tensione di riferimento può essere scritta nel seguente modo:

$$V_{ref} = V_{dd} \cdot \frac{R_2}{R_1 + R_2} \quad (2.8)$$

È evidente la diretta proporzionalità tra la tensione di riferimento e la tensione di alimentazione V_{dd} . Se quest'ultima risultasse affetta da fluttuazioni, inevitabilmente anche la V_{ref} erediterebbe le variazioni, rendendo il circuito non utile da un punto di vista applicativo. Di fatto questa semplice struttura può essere utilizzata solo se si dispone di un generatore di tensione PVT-independent dato che, una volta garantita l'indipendenza della V_{dd} , il rapporto delle resistenze nell'equazione assicura un'insensibilità nei confronti della temperatura [19]. Al fine di dimostrare l'indipendenza dalla temperatura è utile riportare, per un processo CMOS, la relazione esistente tra la resistenza R del resistore e la temperatura T :

$$R(T) = R(T_0) \cdot [1 + TCR1 \cdot (T - T_0) + TCR2 \cdot (T - T_0)^2] \quad (2.9)$$

Nell'equazione $R(T_0)$ rappresenta il valore di resistenza per un valore di temperatura fissato, ad esempio a temperatura ambiente, mentre $TCR1$ e $TCR2$ sono i coefficienti di temperatura, rispettivamente del primo ordine e del secondo ordine, in cui:

$$TCR1 = \frac{1}{R} \cdot \frac{\partial R}{\partial T} \quad (2.10)$$

Se si assume $TCR2 = 0$ allora l'equazione può essere riscritta secondo la seguente espressione:

$$\begin{aligned}
 V_{ref} &= V_{dd} \cdot \frac{R_2(T)}{R_1(T) + R_2(T)} = \\
 &= V_{dd} \cdot \frac{R_2(T_0) \cdot [1 + TCR1 \cdot (T - T_0)]}{(R_1(T_0) + R_2(T_0)) \cdot [1 + TCR1 \cdot (T - T_0)]} = V_{dd} \cdot \frac{R_2(T_0)}{R_1(T_0) + R_2(T_0)}
 \end{aligned} \tag{2.11}$$

L'espressione (2.11) dimostra il vantaggio introdotto dall'operazione matematica di divisione che permette di compensare l'effetto della temperatura, ma sottolinea lo svantaggio principale, ossia la dipendenza dalla tensione di alimentazione V_{dd} , come spiegato in precedenza.

2.3.2 Divisore di tensione MOS

Un altro esempio di pseudo-riferimento è rappresentato nella figura 2.5 in cui un n-MOS (M1) ed un p-MOS (M2), condividendo i terminali di gate e drain, operano nella regione di funzionamento satura, infatti, considerando il MOS di tipo n, analogamente al p-MOS, la condizione $V_{ds} > V_{gs} - V_{th}$ è automaticamente verificata, essendo:

$$V_{dg} = 0 \tag{2.12}$$

$$V_{ds} = V_{gs} + V_{dg} = V_{gs} \tag{2.13}$$

Le due correnti di drain nella regione di funzionamento di saturazione [20], trascurando la modulazione della lunghezza di canale, sono riportate nelle seguenti equazioni:

$$I_{d1} = \frac{\beta_1}{2} (V_{ref} - V_{th1})^2 \tag{2.14}$$

$$I_{d2} = \frac{\beta_2}{2} (V_{dd} - V_{ref} - |V_{th2}|)^2 \tag{2.15}$$

in cui il parametro β è descritto dalla seguente espressione:

$$\beta = \mu C_{ox} \frac{W}{L} \tag{2.16}$$

in cui μ è la mobilità dei portatori (elettroni o lacune), W indica la larghezza del canale, L la lunghezza del canale e il termine C_{ox} è la capacità dell'ossido per unità di area data da:

$$C_{ox} = \frac{\epsilon_{ox}}{t_{ox}} \tag{2.17}$$

dove t_{ox} è lo spessore dell'ossido e ϵ_{ox} è la permittività elettrica dell'ossido.

Dato che la corrente in continua, ossia a frequenza molto bassa, entrante nel gate di un MOSFET può essere considerata pressochè nulla, a causa della elevata impedenza di ingresso che il terminale stesso di gate presenta, è lecito scrivere la seguenti uguaglianze tra le correnti di drain dei due transistori:

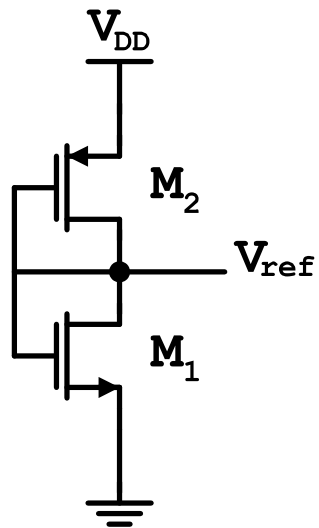


Figura 2.5. Divisore di tensione MOSFET

$$I_{d1} = I_{d2} \quad (2.18)$$

$$\frac{\beta_1}{2}(V_{ref} - V_{th1})^2 = \frac{\beta_2}{2}(V_{dd} - V_{ref} - |V_{th2}|)^2 \quad (2.19)$$

Di conseguenza la tensione di riferimento può essere espressa nel modo seguente:

$$V_{ref} = \frac{V_{dd} - |V_{th2}| + \sqrt{\frac{\beta_1}{\beta_2}} V_{th1}}{\sqrt{\frac{\beta_1}{\beta_2}} + 1} \quad (2.20)$$

Anche questa soluzione circuitale presenta lo stesso problema riscontrato nel precedente partitore di tensione, ossia la totale dipendenza della tensione di riferimento dalle possibili variazioni della tensione di alimentazione V_{dd} . Introducendo il concetto di sensitivity S nella seguente espressione:

$$S_{V_{dd}}^{V_{ref}} = \frac{\frac{\partial V_{ref}}{V_{ref}}}{\frac{\partial V_{dd}}{V_{dd}}} = \frac{V_{dd}}{V_{ref}} \frac{\partial V_{ref}}{\partial V_{dd}} \quad (2.21)$$

si ricava:

$$S_{V_{dd}}^{V_{ref}} = \frac{V_{dd}}{V_{dd} - |V_{th2}| + \sqrt{\frac{\beta_1}{\beta_2}} V_{th1}} \quad (2.22)$$

Inoltre, se si considera $\beta_1 \approx \beta_2$ e $|V_{th2}| \approx V_{th1}$ allora:

$$S_{V_{dd}}^{V_{ref}} \approx 1 \quad (2.23)$$

Il risultato ottenuto indica che, ad esempio, una variazione del 10% della tensione di alimentazione comporta una variazione del 10% della tensione di riferimento. Si tratta di un pessimo comportamento, non tollerabile in sistemi che richiedono elevate prestazioni in termini di accuratezza. Ciononostante, un corretto dimensionamento dei due transistori permette di ottenere un eccellente comportamento nei confronti delle variazioni di temperatura. Il coefficiente di temperatura della tensione di riferimento TCV_{ref} viene riportato di seguito:

$$TCV_{ref} = \frac{1}{V_{ref}} \frac{\partial V_{ref}}{\partial T} = \frac{1}{V_{ref}} \frac{1}{1 + \sqrt{\frac{\beta_1}{\beta_2}}} \left[\frac{\partial(-|V_{th2}|)}{\partial T} + \sqrt{\frac{\beta_1}{\beta_2}} \frac{\partial V_{th1}}{\partial T} \right] \quad (2.24)$$

Se si pone

$$TCV_{ref} = 0 \quad (2.25)$$

si ottiene la seguente espressione:

$$\frac{\partial(-|V_{th2}|)}{\partial T} + \sqrt{\frac{\beta_1}{\beta_2}} \frac{\partial V_{th1}}{\partial T} = 0 \quad (2.26)$$

Siccome $\frac{\partial(-|V_{th2}|)}{\partial T} > 0$ e $\frac{\partial V_{th1}}{\partial T} < 0$, ossia presentano rispettivamente un comportamento CTAT (complementary to absolute temperature) e PTAT (proportional to absolute temperature), si può affermare che si riesce ad ottenere una dipendenza nei confronti della temperatura pressochè nulla, dimensionando opportunamente le lunghezze e le larghezze dei transistori [19].

2.4 Classificazione dei riferimenti di tensione

Esistono diversi approcci che sono stati usati per realizzare i riferimenti di tensione nei circuiti integrati [21], di seguito elencati:

- Si sfrutta un diodo Zener che presenta una certa tensione di breakdown quando viene fatto lavorare nella regione di funzionamento inversa.
- Viene sfruttata la differenza tra le tensioni di soglia tra un transistor ad arricchimento ed un transistor a svuotamento.
- Il comportamento CTAT (complementary to absolute temperature) della tensione ai capi di un diodo polarizzato direttamente viene compensato dal comportamento PTAT da parte di altri elementi del circuito.

Il primo approccio non è più molto usato in quanto, nei moderni circuiti integrati, non sempre si dispone dello Zener voluto nel processo. Inoltre, il consumo di potenza di un riferimento basato su uno Zener risulta essere abbastanza elevato e il comportamento in termini di rumore risulta degradato. Il secondo approccio non può essere sfruttato quando i transistori a svuotamento non sono disponibili, come spesso accade. Il terzo approccio, impiegato in circuiti che presentano sia la tecnologia bipolare che quella CMOS, risulta essere il più utilizzato [21].

2.5 Riferimenti basati sulle proprietà della giunzione pn

I circuiti di seguito presentati prevedono l'utilizzo di un diodo, ossia di una giunzione p-n, ottenibile in un processo CMOS sfruttando il contatto p+ e la n-well, in un processo n-well, in cui è presente il transistor bipolare parassita tra il substrato di tipo p, l'impianto p+ e la n-well come mostrato nella figura 2.6 [19].

La corrente che scorre in una giunzione pn polarizzata direttamente è descritta dalla seguente espressione:

$$I_d = I_s \cdot \left(\exp^{\frac{V_d}{nV_T}} - 1 \right) \quad (2.27)$$

in cui V_d rappresenta la tensione ai capi della giunzione, V_T è l'equivalente in tensione della temperatura ($\frac{kT}{q}$), n è il coefficiente di emissione, ed infine I_s è la corrente inversa di saturazione (scale current). Imponendo un valore di corrente che scorre nel diodo è possibile stimare come la tensione ai capi del diodo varia al variare della temperatura. Da risultati sperimentali si osserva che:

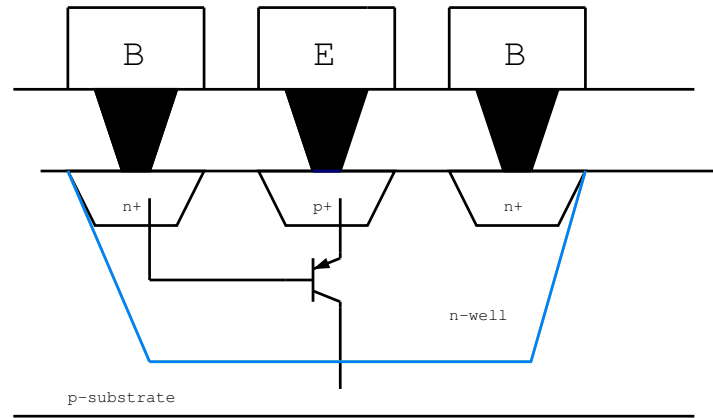


Figura 2.6. Transistore pnp bipolare parassita verticale

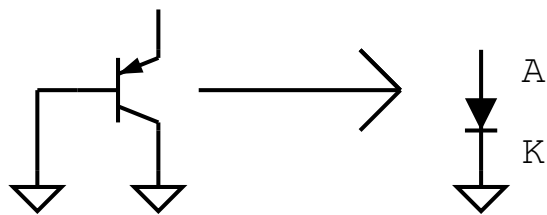


Figura 2.7. Transistore bipolare in configurazione diodo

$$\frac{dV_d}{dT} = -(1.5 : 2.5) \text{ [mV/K]} \quad (2.28)$$

in cui il valore esatto dipende dal valore della corrente iniettata nel diodo. Il risultato dell'espressione permette di affermare che la tensione ai capi del diodo diminuisce all'aumentare della temperatura, ossia assume un comportamento CTAT (complementary to absolute temperature). Il motivo alla base del comportamento appena descritto è legato alla crescita esponenziale della concentrazione dei portatori minoritari al crescere della temperatura. Inoltre, la debole dipendenza della bandgap energy E_g del silicio dalla temperatura, riportata nella seguente relazione [19], rappresenta un fenomeno del secondo ordine:

$$E_g(T) = 1.16 - (702 \cdot 10^{-6}) \frac{T^2}{T + 1108} \text{ [eV]} \quad (2.29)$$

in cui la temperatura T è espressa in Kelvin e $1\text{eV} = 1.61 \cdot 10^{-19} \text{ J}$. A temperatura ambiente il bandgap del silicio vale circa 1.1 eV.

2.5.1 Riferimenti Bandgap

Il circuito in figura 2.8 sfrutta il comportamento CTAT (Complementary to absolute temperature) della tensione ai capi di un diodo [19], insieme al comportamento PTAT (proportional to absolute temperature) della corrente che scorre nel ramo di uscita, per ottenere una tensione di riferimento, di valore paragonabile alla tensione di bandgap del silicio, indipendente dalle variazioni della temperatura.

È necessaria la presenza di un circuito di start-up che consenta l'accensione dell'intero sistema, non appena viene applicata una tensione di alimentazione V_{dd} , per permettere lo scorrimento di una corrente non nulla nel ramo di uscita.

Considerando la maglia comprendente i diodi D1 e D2, ottenuti cortocircuitando la base e il collettore dei transistori bipolari corrispondenti, e il resistore di resistenza R , si applica la legge di Kirchhoff delle tensioni (KVL), supponendo che i due transistori nella maglia presentino la stessa V_{gs} :

$$V_{d1} = V_{d2} + I_{d2} \cdot R \quad (2.30)$$

L'equazione (2.30) è soddisfatta solo se $V_{d2} < V_{d1}$ nonostante $I_{d1} = I_{d2}$ per via dello specchio di corrente formato dai transistori della parte superiore del circuito. A tal fine il BJT corrispondente al diodo D2 viene dimensionato con larghezza K volte superiore rispetto a quella del BJT D1. Di conseguenza:

$$I_{s2} = K \cdot I_{s1} \quad (2.31)$$

Richiamando l'equazione (2.27), essendo D1 e D2 due diodi polarizzati direttamente, sono vere le seguenti espressioni:

$$I_{d1} \approx I_{s1} \cdot e^{\frac{V_{d1}}{nV_T}} \quad (2.32)$$

$$I_{d2} \approx K \cdot I_{s1} \cdot e^{\frac{V_{d2}}{nV_T}} \quad (2.33)$$

dalle quali si possono ricavare le tensioni sui diodi D1 e D2:

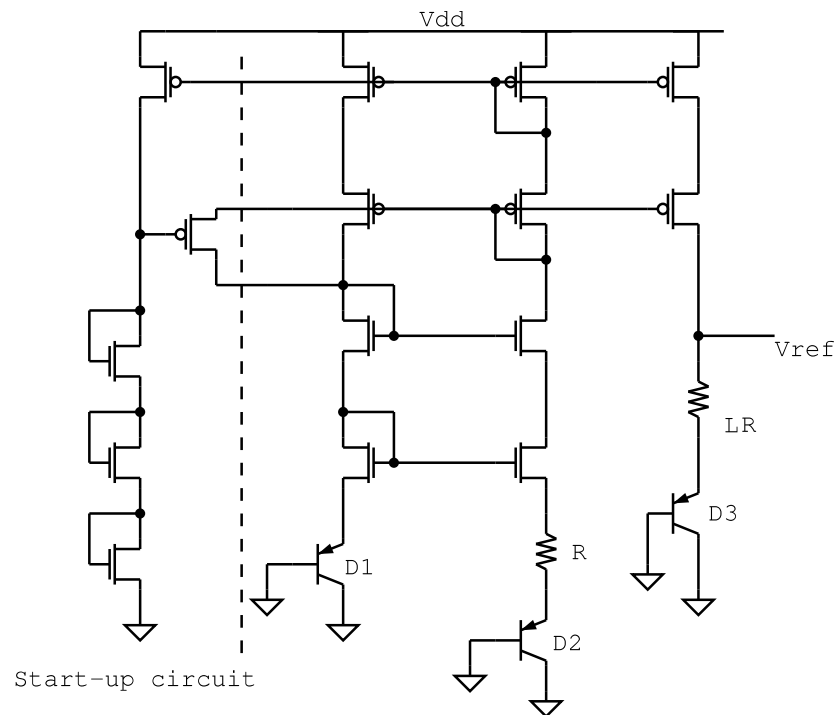


Figura 2.8. Riferimento bandgap con transistori a canale lungo

$$V_{d1} = n \cdot V_T \cdot \log\left(\frac{I_{d1}}{I_{s1}}\right) \quad (2.34)$$

$$V_{d2} = n \cdot V_T \cdot \log\left(\frac{I_{d1}}{K \cdot I_{s1}}\right) \quad (2.35)$$

Siccome si impone, attraverso la rete superiore di specchio di corrente, $I_{d1}=I_{d2}=I$ allora si ricava il valore del resistore e di conseguenza il valore della corrente che scorre nei diodi:

$$R = \frac{n \cdot V_T \cdot \log(K)}{I} \quad (2.36)$$

$$I = \frac{n \cdot k \cdot \log(K)}{qR} \cdot T \quad (2.37)$$

Dunque si nota la diretta dipendenza della corrente dalla temperatura T: all'aumentare della temperatura la corrente che scorre nei diodi D1 e D2 aumenta. Questa è la stessa corrente che scorre nel ramo di uscita e nel diodo D3, essendo specchiata dalla rete di p-MOS.

La tensione di riferimento V_{ref} si ricava come somma della tensione sul diodo D3, di dimensioni uguali a quelle del diodo D2, e della caduta di tensione sul resistore di resistenza LR :

$$V_{ref} = V_{d3} + I \cdot LR = V_{d3} + L \cdot n \cdot \log(K) \cdot V_T \quad (2.38)$$

Per mettere meglio in evidenza i contributi principali che costituiscono la tensione di riferimento si mostra l'equazione come:

$$V_{ref} = V_{d3} + \gamma \cdot V_T \quad (2.39)$$

in cui $\gamma = L \cdot n \cdot \ln K$. L'espressione di fatto esprime la tensione di riferimento come la somma di un contributo CTAT (V_{d3} , come mostrato nell'espressione 2.18) e di uno PTAT ($V_T = \frac{kT}{q}$), e inoltre non mostra una diretta dipendenza nei confronti della tensione di alimentazione V_{dd} . La variazione della tensione di riferimento dovuta ad una variazione della temperatura, con riferimento all'equazione 2.28, è data da:

$$\frac{\partial V_{ref}}{\partial T} = \frac{\partial V_{d3}}{\partial T} + L \cdot n \cdot \log(K) \cdot \frac{\partial V_T}{\partial T} \quad (2.40)$$

in cui $\frac{\partial V_{d3}}{\partial T} < 0$ (espressione 2.27) e $\frac{\partial V_T}{\partial T} > 0$. Se si pone l'espressione 2.39 pari a zero si riesce a trovare quel valore L da moltiplicare al valore di resistenza del resistore di uscita che garantisce $\frac{\partial V_{ref}}{\partial T} = 0$:

$$L = \frac{-\frac{\partial V_{d3}}{\partial T}}{n \cdot \log(K) \cdot \frac{\partial V_T}{\partial T}} \quad (2.41)$$

A livello circuitale si nota la necessità di utilizzare una $V_{dd} > 1V$, in quanto sono sfruttati stacks di transistori n-MOS e p-MOS nella configurazione cascode a specchio di corrente per ottenere una minore sensibilità alle variazioni della tensione di alimentazione [19].

2.5.2 Bandgap alternativi

La figura 2.9 rappresenta un'alternativa al bandgap reference presentato in precedenza e viene sfruttato nel caso in cui si voglia ridurre la potenza dissipata e l'area occupata, nonostante necessiti di una minima tensione di alimentazione V_{dd} maggiore di quella precedente a causa della presenza di una resistenza in più (LR) sui due rami [19].

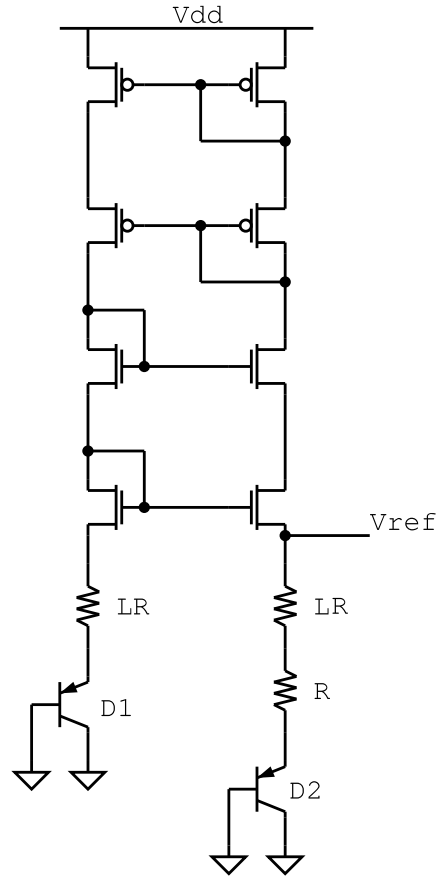


Figura 2.9. Esempio di bandgap alternativo

In questo caso la corrente che scorre nei due rami è ancora PTAT e vale:

$$I_{ref} = \frac{V_T \cdot \log(K)}{R} = \frac{k \cdot \log(K)}{qR} \cdot T \quad (2.42)$$

e di conseguenza

$$V_{ref} = V_{d2} + L \cdot \log(K) \cdot V_T \quad (2.43)$$

La tensione di riferimento presenta come in precedenza un elemento CTAT (V_{d2}) e PTAT (V_T) utili per compensare le variazioni della temperatura.

La figura 2.10 mostra una topologia di bandgap in cui viene sfruttato un amplificatore

differenziale che fissa la tensione sui gate di due p-MOS, i quali erogano la stessa corrente nei due rami del circuito, condividendo la stessa V_{sg} [19].

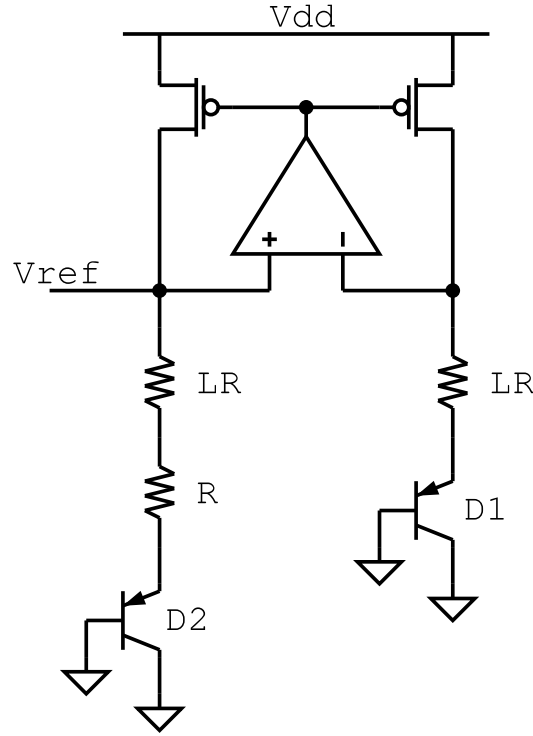


Figura 2.10. BGR with amplifier

Dato che i due p-MOS in figura rappresentano degli stadi invertenti, essendo stadi in configurazione common-source, e affinché il sistema risulti stabile, è necessario che l'ingresso non invertente dell'amplificatore sia collegato al nodo in cui si ha maggiore impedenza, visto che nei due rami scorre la stessa corrente. In questo modo si garantisce che il segnale proveniente dalla rete di feedback e sottratto all'ingresso dell'amplificatore sia maggiore del segnale che viene sommato all'ingresso, assicurando la stabilità. Il beneficio principale della suddetta struttura consiste nel poter usare V_{dd} minori e nel poter garantire una maggior indipendenza della V_{ref} nei confronti delle variazioni della tensione di alimentazione [19]. L'introduzione dell'amplificatore differenziale dotato di guadagno elevato permette, infatti, di ridurre notevolmente le variazioni della tensione di riferimento. Una variazione della tensione di alimentazione si traduce in una variazione della corrente che scorre nei due rami, comportando di conseguenza un'iniziale variazione delle tensioni all'ingresso dell'amplificatore. L'uscita di quest'ultimo agirà sulla tensione V_{sg} dei p-MOS, causando una variazione della corrente che si oppone all'iniziale variazione della tensione V_{dd} .

2.5.3 Riferimento bandgap a bassa tensione

Un esempio di riferimento bandgap usato con tensioni di alimentazione $V_{dd} \leq 1V$ è mostrato nella figura 2.11 [22]. La maglia costituita dal resistore R e dai transistori bipolari connessi a diodo $D1$ e $D2$, in cui $D2$ è K volte più largo di $D1$, fornisce una corrente direttamente proporzionale alla temperatura, come spiegato in precedenza:

$$I_{PTAT} = \frac{n \cdot V_T \cdot \log(K)}{R} \quad (2.44)$$

Inoltre se si considera la maglia formata dal diodo $D1$ e dal resistore di resistenza LR , si può affermare che la corrente attraverso il resistore è di tipo CTAT, essendo la tensione sul diodo $D1$ (polarizzato direttamente) di tipo CTAT, come osservato in precedenza:

$$I_{CTAT} = \frac{V_{d1}}{LR} \quad (2.45)$$

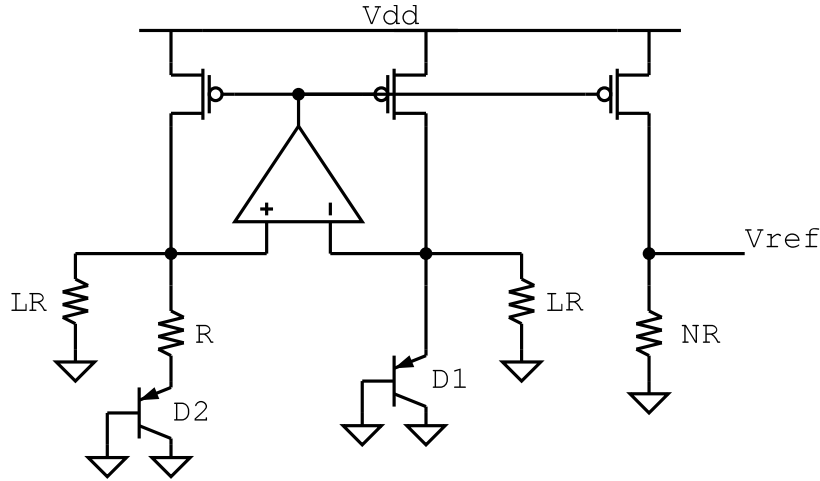


Figura 2.11. Bandgap a bassa tensione [22]

Le due correnti si sommano al nodo negativo dell'amplificatore operazionale e forniscono così una corrente totale I , specchiata nel ramo di uscita grazie alla presenza dello specchio di corrente formato dai p-mos, che, scorrendo nel resistore di resistenza NR , determina la tensione di riferimento V_{ref} :

$$V_{ref} = n \cdot V_T \cdot N \cdot \log(K) + \frac{N}{L} \cdot V_{d1} \quad (2.46)$$

Dunque V_{ref} contiene un parametro CTAT e uno PTAT, come negli esempi precedenti, infatti il comportamento in temperatura della tensione di riferimento è:

$$\frac{\partial V_{ref}}{\partial T} = \frac{N}{L} \cdot \frac{\partial V_{d1}}{\partial T} + N \cdot n \cdot \log(K) \cdot \frac{\partial V_T}{\partial T} \quad (2.47)$$

Per ottenere $\frac{\partial V_{ref}}{\partial T} = 0$ si sceglie:

$$L = \frac{-\frac{\partial V_{d1}}{\partial T}}{n \cdot \log(K) \cdot \frac{\partial V_T}{\partial T}} \quad (2.48)$$

Per scegliere il valore di riferimento di tensione in uscita bisogna dimensionare opportunamente la sezione del diodo D2, ossia il parametro K, ed in seguito, stabilire il valore N del resistore di uscita sul quale viene prelevata la tensione V_{ref} . Questa configurazione permette di ottenere tensioni di riferimento prossime a 0.5V e dunque minori del valore del bandgap del silicio.

2.6 Riferimento di tensione per sistemi ultra-low power

Per tutti quei sistemi elettronici integrati che sono sfruttati in applicazioni ultra-low power, sono necessari dei riferimenti di tensione con caratteristiche diverse rispetto a quelle dei classici bandgap presentati precedentemente, a causa di un consumo relativamente elevato e non particolarmente adatti a funzionare a tensioni di alimentazione ridotte (<1.2V per un riferimento bandgap tradizionale). In particolare, oltre a dover svolgere la funzione di riferimento, ossia, oltre a dover garantire ottime prestazioni nei confronti delle variazioni PVT, essi devono operare con tensioni di alimentazione minime, su aree molto ridotte e dissipare una frazione della potenza imposta dagli stretti vincoli energetici propri sistema che li ospita. Nei bandgap reference, ad esempio, anche se la presenza degli amplificatori operazionali fornisce una buona insensibilità alle variazioni di temperatura e della tensione di alimentazione, la potenza associata (centinaia di μW) e l'area occupata (centesimi di mm^2) rappresentano due fattori limitanti e significativi [8]. Di seguito viene descritta una soluzione circuitale strutturalmente molto semplice che permette di soddisfare i requisiti di minima tensione di alimentazione e di minima potenza dissipata.

2.6.1 Riferimento di tensione 2T

La figura 2.12 rappresenta un riferimento di tensione [8] composto da due MOSFET. Il transistor M1, a differenza del transistor M2, è un MOS nativo, ossia presenta una tensione di soglia prossima allo zero. Al fine di dissipare la minore potenza possibile, i due transistori lavorano nella regione di funzionamento sottosoglia (weak inversion), ossia nella condizione in cui [20]:

$$V_{gs} < V_{th} \quad (2.49)$$

Anche se si tratta, soprattutto nei sistemi puramente digitali, di una condizione in cui il MOS viene considerato spento, in realtà esso conduce una corrente di perdita (leakage), che provoca delle dissipazioni parassite di potenza [23]. D'altra parte, invece, nelle applicazioni low-power la conduzione sotto soglia risulta particolarmente importante quando il MOS è

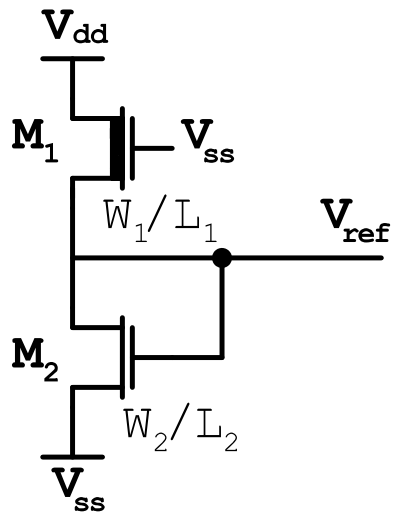


Figura 2.12. 2T voltage reference [8]

sfruttato come un dispositivo a bassa tensione e bassa potenza [24].
L'espressione della corrente sottosoglia è riportata di seguito [8]:

$$I_{sub} = \mu C_{ox} \frac{W}{L} (m-1) V_T^2 \exp\left(\frac{V_{gs} - V_{th}}{m V_T}\right) \left(1 - \exp\left(\frac{-V_{ds}}{V_T}\right)\right) \quad (2.50)$$

in cui μ è la mobilità degli elettroni, C_{ox} è la capacità dell'ossido di gate per unità di area, W e L sono rispettivamente la larghezza e la lunghezza del transistor, m è il subthreshold slope factor, V_T è l'equivalente in tensione della temperatura $\frac{kT}{q}$, in cui k è la costante di Boltzmann, T è la temperatura, q è la carica dell'elettrone, V_{th} è la tensione di soglia del transistor.

Se si suppone che la tensione tra il drain e il source del singolo transistor è tale per cui $V_{ds} > 5 - 6V_T$, allora il termine $(1 - \exp(\frac{-V_{ds}}{V_T}))$ può essere considerato trascurabile. Di conseguenza, siccome la stessa corrente scorre nei due transistori, è valida la seguente uguaglianza:

$$I = \mu_1 C_{ox1} \frac{W_1}{L_1} (m_1-1) V_T^2 \exp\left(\frac{0 - V_{ref} - V_{th1}}{m_1 V_T}\right) = \mu_2 C_{ox2} \frac{W_2}{L_2} (m_2-1) V_T^2 \exp\left(\frac{V_{ref} - V_{th2}}{m_2 V_T}\right) \quad (2.51)$$

Dall'uguaglianza delle correnti è possibile ricavare l'espressione della tensione di riferimento:

$$V_{ref} = \frac{m_1 m_2}{m_1 + m_2} (V_{th2} - V_{th1}) + \frac{m_1 m_2}{m_1 + m_2} V_T \log\left(\frac{\mu_1 C_{ox1} W_1 L_2}{\mu_2 C_{ox2} W_2 L_1}\right) \quad (2.52)$$

Si può notare che nessun termine nella formula dipende al primo ordine dalla tensione di alimentazione. Se si utilizzano transistori a canale lungo ($L > 1\mu m$), infatti, si riesce ad aumentare l'insensibilità della V_{ref} rispetto alle variazioni della tensione di alimentazione, in quanto gli effetti di canale corto, come l'effetto DIBL (drain induced barrier lowering) risultano trascurabili. Per ottenere l'indipendenza di V_{ref} dalle variazioni della temperatura, ossia, al fine di garantire che $\frac{\partial V_{ref}}{\partial T} = 0$, risulta necessario dimensionare opportunamente la larghezza e la lunghezza dei singoli transistori, tenendo in considerazione che la tensione di soglia V_{th} esibisce un comportamento CTAT, invece l'equivalente in tensione della temperatura V_T , come definito in precedenza, è un parametro PTAT. Ponendo infatti $\frac{\partial V_{ref}}{\partial T} = 0$ si ottiene il seguente rapporto ottimale tra le larghezze dei due transistori:

$$\left(\frac{W_1}{W_2}\right)_{opt} = \frac{\mu_2 C_{ox2} L_2}{\mu_1 C_{ox1} L_1} \exp\left[\frac{q}{k} (TCV_{th2} - TCV_{th1})\right] \quad (2.53)$$

in cui TCV_{th1} e TCV_{th2} rappresentano i coefficienti di temperatura del primo ordine delle tensioni di soglia dei transistori M_1 ed M_2 . Il coefficiente di temperatura del secondo ordine è stato ignorato, senza perdite significative di accuratezza, essendo due ordini di grandezza inferiore rispetto al coefficiente di temperatura del primo ordine.

Per quanto riguarda la minima tensione di alimentazione applicabile, che rappresenta uno dei punti di forza di questa struttura circuitale, bisogna garantire che la tensione drain-source del transistor M_2 , che coincide con la tensione di riferimento, sia superiore a $5-6V_T$ per poter considerare valida l'espressione (2.52) della V_{ref} . Quest'ultimo vincolo impone, inoltre, una accurata scelta della minima differenza tra le due tensioni di soglia ΔV_{th} : se si approssima il termine logaritmico al primo ordine nella formula della V_{ref} , se si assume che tipicamente $m_1 = m_2 = 1.5$, allora:

$$V_{ref} = 0.75\Delta V_{th} \quad (2.54)$$

Di conseguenza la minima ΔV_{th} corrisponde al range 6.6-8 V_T , ossia 170-200mV. Utilizzando una tecnologia CMOS standard 0.13 μ m, si riescono a raggiungere ottime performance in termini di coefficiente termico TC e di line sensitivity LS [8].

Si ottiene un ottimo comportamento al variare della temperatura, infatti il coefficiente termico risulta prossimo a 20 ppm/°C anche con la minima tensione di alimentazione applicabile, ossia 0.5V. Complessivamente si tratta di un riferimento scarsamente sensibile alle variazioni di temperatura per diversi valori di V_{dd} compreso tra 0.5V e 3.5V. Si ottengono risultati significativi anche per quanto riguarda la line sensitivity (LS \approx 0.033%/V a 20°C).

Inoltre, se si adotta una tensione di alimentazione V_{dd} =0.5V, il consumo di potenza statica è di circa 2 pW, a 27°C. Nel caso peggiore, invece, alla temperatura di 80°C e V_{dd} =3.3V, il consumo di potenza equivale a 243 pW [8]. Si tratta di un risultato eccellente, che rende questo riferimento di tensione pienamente adatto ai sistemi che lavorano sia in modalità attiva che in modalità standby, in cui il consumo di potenza deve essere minimizzato all'estremo.

Il rapporto tra le larghezze dei due transistori (2.53), come sottolineato in precedenza, risulta critico nello stabilire il coefficiente termico TC e la tensione di uscita. Nonostante si possa determinare il rapporto ottimale delle larghezze, le variazioni introdotte dalle tolleranze di fabbricazione dei processi tecnologici determinano uno scostamento dai risultati ritenuti ottimali nella fase di progettazione. Come accade per i riferimenti tradizionali, la calibrazione post-fabbricazione consente di migliorare le prestazioni, nonostante risulti dispendioso in termini di tempo e di costi.

Capitolo 3

Riferimento di tensione virtuale

In questo capitolo viene presentato il concetto di riferimento di tensione virtuale (virtual voltage reference) [10], inteso come un riferimento di tensione basato su un algoritmo software di tipo digitale, utile per rimpiazzare i tradizionali riferimenti analogici negli attuali sistemi elettronici digitali integrati che sfruttano lo scaling della tecnologia e la conseguente diminuzione delle tensioni utilizzate.

In principio viene mostrata la derivazione dell'algoritmo a partire dalla presentazione di un riferimento bandgap tradizionale. In seguito viene presentata lo schema a blocchi del riferimento virtuale, lo studio teorico e i risultati che mostrano la validità in termini di performance della struttura.

Per introdurre il principio del riferimento virtuale di tensione, viene illustrato come sia possibile tradurre in digitale il funzionamento di un riferimento analogico, quale il bandgap di Kuijk.

3.1 Riferimento bandgap di Kuijk

Nel riferimento di tensione di Kuijk [25], mostrato nella figura 3.1, in maniera simile ai riferimenti bandgap appena discussi, sono presenti due transistori bipolari pnp integrati in configurazione diodo (Q1-Q2), che presentano due aree di emettitore adeguate, tre resistori e un amplificatore operazionale.

I due diodi polarizzati direttamente sono percorsi dalla stessa corrente i_0 , ottenuta a partire dalla tensione di uscita dell'amplificatore operazionale che incide sui due resistori R_0 . Di fatto essi sono percorsi dalla stessa corrente in quanto le tensioni ai loro capi vengono mantenute costanti grazie alla reazione negativa. Considerando trascurabili le correnti entranti nell'amplificatore operazionale, la corrente i_0 , oltre ad attraversare i resistori, scorre anche nei diodi e può essere espressa come di seguito:

$$i_0 = \frac{V_{REF} - V_{D1}}{R_0} \quad (3.1)$$

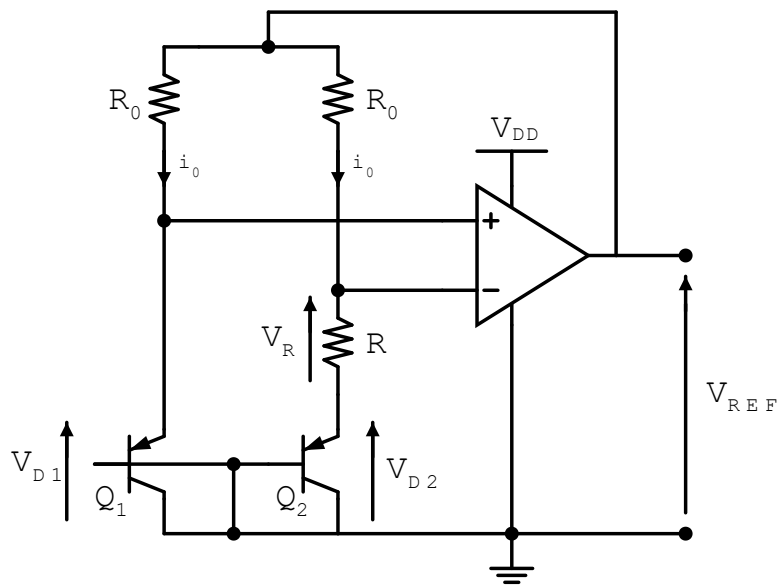


Figura 3.1. Circuito di Kuijk [25]

Inoltre la tensione di riferimento V_{REF} , applicando la legge di Kirchoff delle tensioni sulle due maglie comprendenti gli ingressi dell'operazionale, assume le seguenti uguaglianze:

$$V_{REF} = V_{D1} + R_0 i_0 \quad (3.2)$$

$$V_{REF} = V_{D2} + i_0(R_0 + R) \quad (3.3)$$

Ponendo l'uguaglianza delle due espressioni, si ottiene:

$$V_{D1} = V_{D2} + \frac{R}{R_0}(V_{REF} - V_{D1}) \quad (3.4)$$

Di conseguenza la tensione di riferimento V_{REF} ottiene la seguente espressione chiave

$$V_{REF} = V_{D1} + \frac{R_0}{R} \Delta V_D \quad (3.5)$$

in cui $\Delta V_D = V_{D1} - V_{D2}$. Si riesce ad ottenere un riferimento di tensione indipendente dalle variazioni della temperatura notando che la tensione di riferimento viene espressa come la somma di una componente CTAT e di una PTAT. Come già osservato precedentemente, la tensione ai capi di un diodo polarizzato direttamente assume un comportamento CTAT, per cui $\frac{dV_{D1}}{dT} < 0$. Inoltre la differenza delle due tensioni di polarizzazione dei due diodi Q1 e Q2 è esprimibile come di seguito:

$$\Delta V_D = \frac{kT}{q} \log\left(\frac{J_1}{J_2}\right) \quad (3.6)$$

in cui $J_1/J_2 = h^*$ rappresenta il rapporto tra le densità di corrente che scorrono nel diodo, k è la costante di Boltzmann e q è la carica dell'elettrone. Si intuisce facilmente che ΔV_D ha un comportamento PTAT, essendo direttamente proporzionale alla temperatura. Analogamente ai bandgap reference presentati precedentemente, in cui la compensazione in temperatura viene raggiunta per una V_{REF} prossima a 1.2V, ossia pari al bandgap del silicio, è possibile ottenere una $\frac{\partial V_{REF}}{\partial T} = 0$ scegliendo opportunamente il rapporto tra le resistenze R_0 ed R .

Come mostrato nella figura 3.2, il circuito in questione può essere visto come un insieme di blocchi funzionali all'interno di un unico sistema che fornisce in uscita la tensione di riferimento V_{REF} [10]:

- I due diodi Q1 e Q2 sono usati come standard fisici, ricevono la corrente i_0 dai resistori R_0 e dall'amplificatore operazionale e forniscono in uscita le tensioni V_{D1} e ΔV_D .
- Considerando trascurabile la finita reiezione della tensione di alimentazione da parte dell'amplificatore operazionale e considerando che i resistori R_0 soddisfano i requisiti di matching, questi ultimi e l'opamp posso essere considerati come circuiti di bias che forniscono la corrente i_0 agli standard fisici. Dall'espressione (3.1) infatti non si nota una dipendenza esplicita della i_0 nei confronti della V_{dd} .

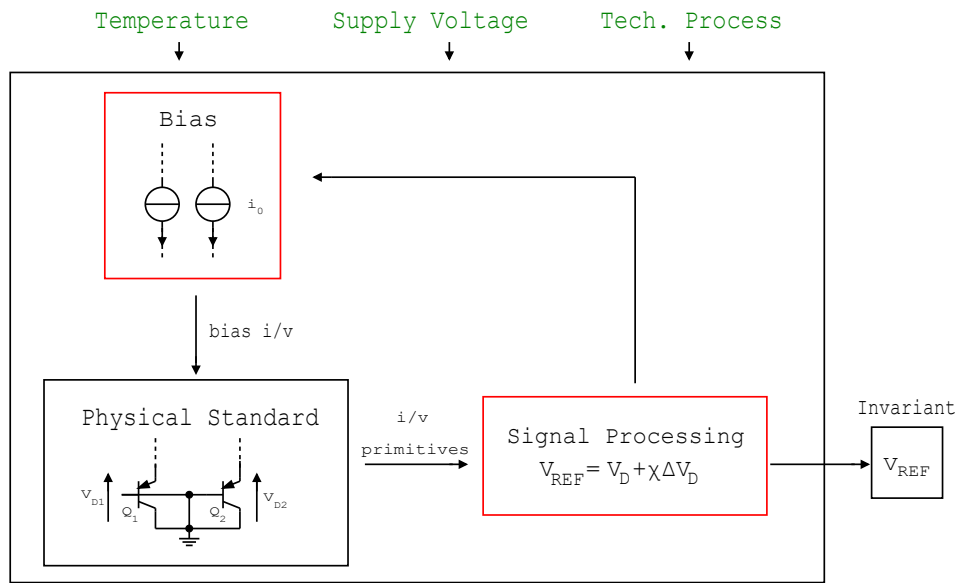


Figura 3.2. Diagramma a blocchi del bandgap di Kuijk [10]

- L'amplificatore operazionale funge anche da blocco di signal processing in quanto effettua la somma pesata delle tensioni V_{D1} e ΔV_D grazie alla presenza di un feedback positivo (R0, Q1) e di uno negativo (R0, R, Q2) e fornisce in uscita la tensione di riferimento V_{REF} .

Lo sviluppo del riferimento di tensione virtuale parte dalla volontà di trasferire nel dominio digitale i blocchi di polarizzazione e di signal processing, mantenendo lo stesso tipo di standard fisico, ossia un diodo.

3.2 Struttura hardware del riferimento di tensione virtuale

L'architettura hardware, mostrata in figura 3.3, rappresenta un SoC (System on Chip) prevalentemente digitale che si compone di un'unità di elaborazione numerica, ossia un processore, di un resistore di resistenza R , di un diodo e presenta un DAC (convertitore digitale-analogico) e un ADC (convertitore analogico-digitale) riferiti alla stessa tensione V_0 [10]. Essa è ottenuta a partire da uno pseudo-riferimento di tensione, come il partitore resistivo di tensione descritto nel capitolo 2, collegato ad un circuito di sample and hold (S/H), che ha lo scopo di mantenere (hold) la tensione V_0 costante con alta accuratezza e per un periodo di tempo piuttosto esteso (decine o centinaia di secondi). Il sistema infatti richiede che la tensione V_0 resti costante entro 1 LSB (Least Significant Bit) per un opportuno periodo di tempo [10].

Per quanto riguarda il funzionamento dell'ADC, dato un valore di ingresso analogico V_{ADC} tale per cui $0 < V_{ADC} < V_0$, dopo un tempo $T_{A/D}$ comprendente il tempo di conversione dell'ADC e ulteriori ritardi nella catena di acquisizione, viene fornito al processore il numero digitale n , su N bit, tale per cui:

$$V_{ADC} = \frac{n}{2^N} V_0 + V_\epsilon \quad (3.7)$$

in cui V_ϵ rappresenta il termine di errore che comprende tutte le non idealità dell'ADC e in particolare si assume che sia dominato dall'errore di quantizzazione tale per cui

$$|V_\epsilon| < \frac{V_0}{2^{N+1}} = (1/2)LSB \quad (3.8)$$

Il DAC riceve dal processore il numero digitale m rappresentato su N bit e lo trasforma, dopo un tempo $T_{D/A}$ in una tensione costante V_{DAC} :

$$V_{DAC} = \frac{m}{2^N} V_0 + V_\epsilon \quad (3.9)$$

anche in questo caso si suppone che V_ϵ sia minore di $1/2$ LSB. In questa trattazione ci si limita a considerare le relazioni ingresso-uscita dei due convertitori appena presentati, ossia non vengono riportate specifiche architetture del DAC e dell'ADC, nonostante esistano implementazioni digitali [26] [7] [27], a bassa tensione e bassa potenza e dall'area ridotta, che potrebbero essere impiegate nello schema. Inoltre nel resto della trattazione non vengono considerati i termini di errore V_ϵ .

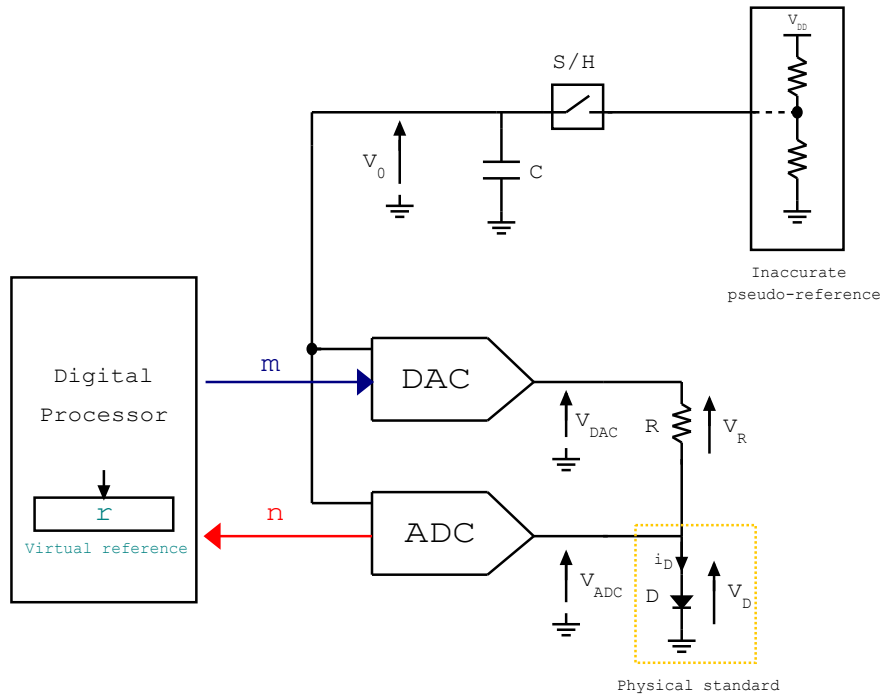


Figura 3.3. SoC per implementazione del riferimento di tensione virtuale basato su tecniche digitali [10]

3.3 Procedura di compensazione

Di seguito viene descritta la procedura di compensazione [10] che permette di ricavare una tensione di riferimento indipendente dalle variazioni della temperatura. Durante la spiegazione della procedura si fa riferimento alle figure 3.3 e 3.4.

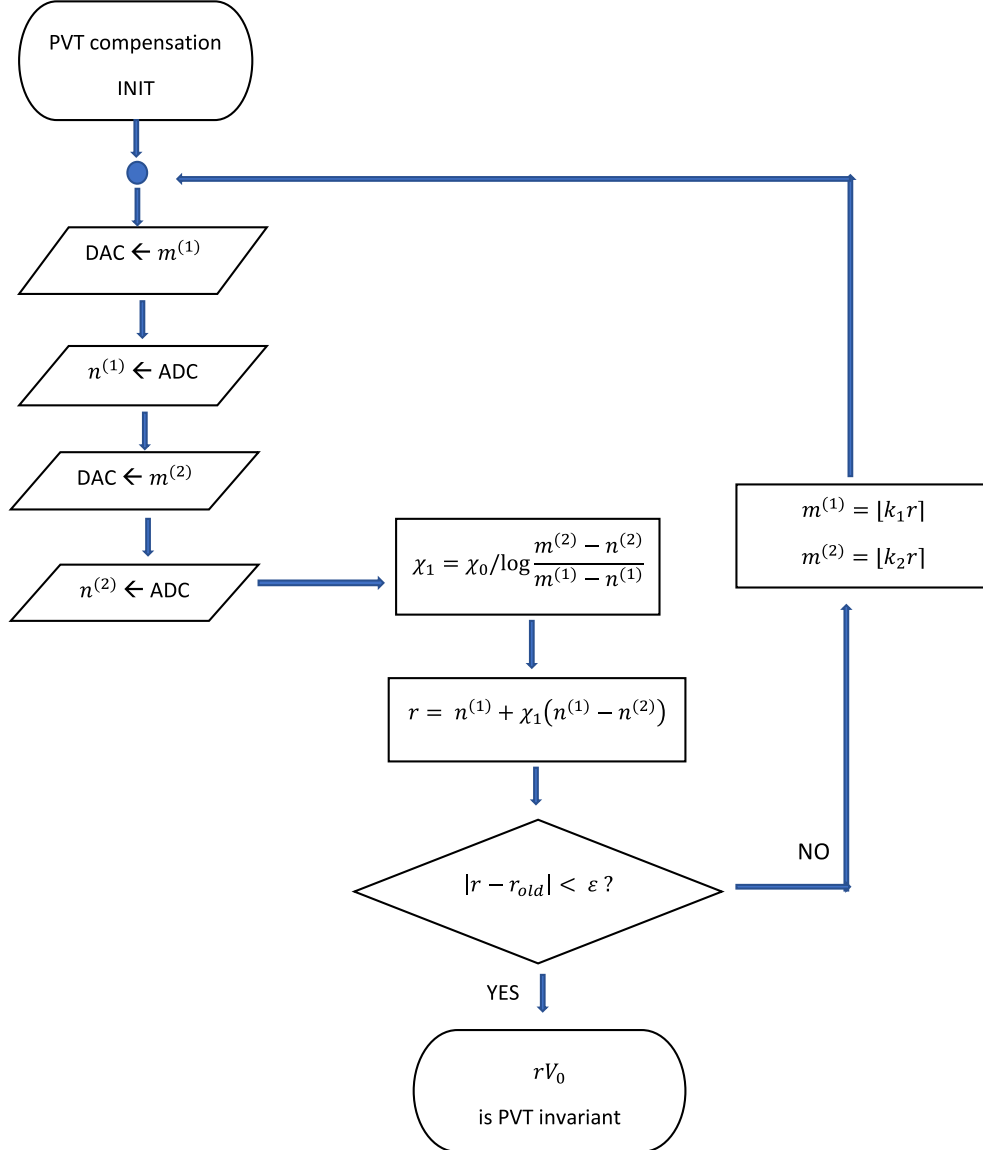


Figura 3.4. Procedura software utilizzata per valutare il riferimento virtuale PVT-independent [10]

Inizialmente, come mostrato nella figura 3.4, vengono generati all'interno del processore due numeri interi digitali $m^{(1)} = \lfloor k_1 r_0 \rfloor$ ed $m^{(2)} = \lfloor k_2 r_0 \rfloor$, in cui $\lfloor \bullet \rfloor$ rappresenta l'operatore

di arrotondamento, r_0 è il valore iniziale di un parametro che verrà presentato nel seguito e k_1 e k_2 sono delle costanti razionali che definiscono due diversi punti di bias.

I due numeri $m^{(1)}$ ed $m^{(2)}$ vengono inviati uno alla volta al DAC che, di conseguenza, genera una tensione analogica $V_{DAC}^{(i)}$ tale per cui:

$$V_{DAC}^{(i)} = \frac{m^{(i)}}{2^N} V_0, \quad i = 1, 2 \quad (3.10)$$

La tensione generata dal DAC, come mostrato nella figura 3.3, viene applicata al ramo contenente il resistore R e il diodo D , creando una corrente di bias $i_D^{(i)}$, che si traduce in una tensione di polarizzazione diretta sul diodo $V_D^{(i)}$ all'ingresso dell'ADC di valore:

$$V_D^{(i)} = \frac{n^{(i)}}{2^N} V_0, \quad i = 1, 2 \quad (3.11)$$

All'uscita del convertitore analogico-digitale dunque viene fornito il numero digitale $n^{(i)}$ su N bit e viene inviato al processore. Per capire la derivazione dei termini χ_1 e χ_0 nell'algoritmo di compensazione, in figura 3.4, è utile esprimere la corrente che scorre nel diodo usato come standard fisico. La corrente di polarizzazione attraverso il diodo $i_D^{(i)}$ può essere espressa come:

$$i_D^{(i)} = \frac{V_{DAC}^{(i)} - V_D^{(i)}}{R} = (m^{(i)} - n^{(i)}) \frac{V_0}{2^N R}, \quad i = 1, 2 \quad (3.12)$$

Di conseguenza è possibile anche definire il rapporto h tra le due densità di corrente J_1 e J_2 , corrispondenti alle due correnti $i_D^{(1)}$ ed $i_D^{(2)}$ che attraversano il diodo:

$$h = \frac{J_1}{J_2} = \frac{i_D^{(1)}}{i_D^{(2)}} = \frac{V_{DAC}^{(1)} - V_D^{(1)}}{V_{DAC}^{(2)} - V_D^{(2)}} = \frac{m^{(1)} - n^{(1)}}{m^{(2)} - n^{(2)}} \quad (3.13)$$

Con riferimento alla formula (3.6), è possibile esprimere la differenza tra le due tensioni di polarizzazione diretta sul diodo ΔV_D , che rappresentava una delle uscite dello standard fisico del bandgap di Kuijk, in funzione del parametro h e delle tensioni $V_D^{(1)}$ e $V_D^{(2)}$, come di seguito mostrato:

$$\Delta V_D = \frac{\log(h^*)}{\log(h)} (V_D^{(1)} - V_D^{(2)}) \quad (3.14)$$

in cui il rapporto dei logaritmi si ricava numericamente.

Di conseguenza, al fine di ricavare l'espressione della tensione di riferimento e del parametro r della procedura di compensazione, in analogia con la formula (3.5), se si considera $V_D = V_D^{(1)}$ e l'espressione (3.11), allora

$$V_{REF} = V_D^{(1)} + \chi \frac{\log(h^*)}{\log(h)} (V_D^{(1)} - V_D^{(2)}) = \frac{V_0}{2^N} [n^{(1)} + \frac{\chi_0}{\log(h)} (n^{(1)} - n^{(2)})] = \frac{V_0}{2^N} r \quad (3.15)$$

in cui $\chi_0 = \chi \log(h^*)$.

Dalla (3.15) si nota che il parametro r è un numero digitale che viene calcolato dal processore, una volta noti i numeri $m^{(1)}$, $m^{(2)}$, $n^{(1)}$ ed $n^{(2)}$. Come si nota dalla procedura in

figura 3.4, per il calcolo definitivo di r sono necessari ulteriori cicli, in cui vengono utilizzati nuovi numeri digitali $m^{(1)}$ ed $m^{(2)}$, riferiti al numero r tramite due costanti k_1 e k_2 , utili per generare nuovi numeri $n^{(1)}$ ed $n^{(2)}$. Nel momento in cui la procedura fornisce un r , tale per cui $|r - r_{old}| < \epsilon$, allora esso viene definito come riferimento virtuale. Nell'espressione (3.15), siccome la tensione V_{REF} , per costruzione, risulta indipendente al primo ordine dalla temperatura allora, data l'uguaglianza, anche il termine a destra risulta esserlo, nonostante V_0 possa essere soggetto a variazioni. Questo vuol dire anche che, ad ogni variazione di temperatura introdotta dalla tensione V_0 , corrisponde una variazione del riferimento virtuale r , tale per cui il prodotto rV_0 resta costante.

Di seguito viene riportata una stima approssimata del tempo richiesto per l'esecuzione dell'algoritmo di compensazione T_{PVT} [10]:

$$T_{PVT} = 2 \cdot T_{D/A} + 2 \cdot T_{A/D} + T_{proc} \quad (3.16)$$

in cui $T_{D/A}$ e $T_{A/D}$ sono i tempi, presentati in precedenza, di conversione dei segnali da parte del DAC e dell'ADC, mentre T_{proc} rappresenta il tempo impiegato dal processore per l'elaborazione dei dati. Risulta che il termine dominante nell'espressione del tempo di esecuzione sia $2T_{A/D}$ [10], che varia da centinaia di nanosecondi fino alle decine di millisecondi, che in generale dipende dall'architettura dell'ADC e dalla risoluzione utilizzata.

Il tempo di esecuzione, inoltre, deve risultare sufficientemente ridotto in modo tale che ogni tipo di variazione PVT introdotta nel sistema durante l'esecuzione risulti minore di 1 LSB. Siccome il riferimento virtuale r risulta valido fino a quando le variazioni di V_0 non superano 1 LSB, allora la procedura di compensazione deve essere ripetuta con un tempo di refresh pari a T_{REFR} :

$$T_{REFR} = \frac{C \cdot V_0}{I_{leak} \cdot 2^N} \quad (3.17)$$

dove C è la capacità del condensatore di hold del circuito di S/H e I_{leak} rappresenta la corrente media di leakage che scorre in esso. Per particolari tipi di S/H a bassa perdita, sono richiesti dei tempi di refresh di decine di secondi per $N=14$ e $C=2.5\text{pF}$ [10].

3.4 Riferimento virtuale

Con riferimento alla figura 3.3, in generale, come osservato in precedenza, risulta lecito affermare che ad ogni tensione V_{ADC} all'ingresso dell'ADC, riferito alla tensione V_0 soggetta a variazioni PVT, corrisponde in uscita un numero digitale k tale per cui:

$$V_{ADC} = \frac{k}{2^N} V_0 \quad (3.18)$$

Sfruttando l'espressione (3.15), si riesce ad esprimere la stessa tensione V_{ADC} in funzione della tensione di riferimento indipendente dalla temperatura V_{REF} :

$$V_{ADC} = \frac{k}{r} V_{REF} = \frac{k 2^N}{r} V_{REF} \quad (3.19)$$

Di fatto il numero digitale $2^N k/r$ rappresenta l'uscita dell'ADC quando questo presenta come tensione di riferimento V_{REF} . Inoltre, analogamente a quanto appena mostrato per

l'ADC, sostituendo l'espressione (3.15) nella (3.10), anche la tensione fornita in uscita dal DAC può essere espressa in funzione della tensione V_{REF} e del riferimento virtuale r :

$$V_{DAC} = \frac{m}{2^N} V_0 = \frac{m}{r} V_{REF} = \frac{m}{2^N} \left(\frac{2^N}{r} V_{REF} \right) \quad (3.20)$$

Considerando le espressioni (3.19) e (3.20), si afferma che il numero digitale r , calcolato nell'espressione (3.15), può essere considerato come un riferimento di tensione virtuale in grado di rimpiazzare un riferimento indipendente dalle variazioni della temperatura utile per i convertitori A/D e D/A. Inoltre, volendo fornire all'uscita fisica del sistema una tensione di riferimento proporzionale a V_{REF} , occorre semplicemente inviare al DAC un numero digitale m che risulti essere proporzionale al riferimento virtuale r . Infatti, volendo convertire un valore $\frac{x}{2^N} V_{REF}$, imponendo

$$\frac{m}{r} V_{REF} = \frac{x}{2^N} V_{REF} \quad (3.21)$$

risulta sufficiente porre in ingresso al DAC il numero digitale $x \cdot r / 2^N$.

3.5 Riferimento virtuale digitale generalizzato

Di seguito viene presentato un metodo che porta alla formulazione generale del concetto di riferimento virtuale [10]:

- La tensione di riferimento dell'espressione (3.5), in generale, può essere associata ad una funzione invariante f di un set v di tensioni primitive PVT-dipendenti provenienti da diversi standard fisici, soggetti a diverse condizioni di bias, progettati in modo che, entro un limite di tolleranza ϵ , la funzione f sia a valore costante in un preciso range di variazioni PVT, tale per cui:

$$\partial f / \partial v_i \neq 0 \quad 1 \leq i \leq N \quad (3.22)$$

$$|f(v) - f_c| < \epsilon \quad \forall v \in V \quad (3.23)$$

in cui $f_c = f(v_n)$ è il valore della funzione f corrispondente alle tensioni v_n sotto le nominali condizioni PVT. L'insieme V rappresenta il set di valori assunti dalle primitive sotto le diverse condizioni PVT comprese nel range operativo.

- Assumendo $\epsilon=0$ ed esprimendo la funzione f in termini di tensioni adimensionali, ossia riferite ad un valore fissato V_u , l'espressione (3.23) si riscrive come:

$$f\left(\frac{v}{V_u}\right) - f_c = 0 \quad \forall v \in V \quad (3.24)$$

- Trascurando l'errore di quantizzazione, siccome i valori delle tensioni sono convertite da un ADC riferito ad un riferimento V_0 , anche PVT-dipendente, introducendo il vettore n dei campioni acquisiti, l'espressione (3.24) viene modificata come di seguito:

$$f\left(\frac{V_0}{2^N V_u} n\right) - f_c = f(\alpha n) - f_c = 0 \quad (3.25)$$

- Per un preciso vettore $n = n^*$, l'equazione appena formulata viene considerata come un'equazione, generalmente non lineare, nella quantità scalare α , proporzionale allo pseudo-riferimento V_0 . Assumendo che un'unica soluzione α^* esista, si può scrivere che:

$$\alpha^* = \frac{V_0}{V_u 2^N} \quad (3.26)$$

$$V_u = \frac{V_0}{2^N \alpha^*} \quad (3.27)$$

La quantità $r = 1/\alpha^*$, in analogia con l'espressione (3.15), può essere considerata come il riferimento virtuale di tensione generalizzata per le conversioni A/D e D/A.

3.6 Indici di prestazione sperimentali

Il concetto di riferimento virtuale di tensione è stato anche implementato, sotto forma di riferimento VKB, che mima il funzionamento di un riferimento bandgap di Kuijk, su un prototipo basato su un'unità a microcontrollore [10]. Il riferimento virtuale, alla fine della procedura di compensazione, viene convertito dal DAC in un'uscita analogica su un pin di uscita sul quale sono stati effettuati i diversi test rispetto alla temperatura (nel range da -10°C a 100°C) e alla tensione di alimentazione da 2.3 V a 2.7 V [10]. Per ulteriori dettagli sulla configurazione si rimanda il lettore a [10]. Di seguito vengono riportati i risultati principali:

- Nei confronti della temperatura, il riferimento esibisce un valore nominale di 1.157 V e un drift termico TC=16 ppm/°C [10].
- Si ottiene una line sensitivity dello 0.15%/V al variare della tensione di alimentazione [10].
- Il consumo di potenza $P = P_{PVT} \cdot \delta$, esprimibile come il prodotto tra la potenza media dissipata dalla procedura di compensazione P_{PVT} dell'ordine del mW e il fattore di attività $\delta = T_{PVT}/T_{REFR}$ dell'ordine di 10^{-4} , raggiunge valori prossimi alle centinaia di nW (120 nW) [10].

Capitolo 4

Riferimento di tensione virtuale VCO-based

Di seguito viene analizzato e discusso un riferimento di tensione concepito a partire dal riferimento di tensione virtuale illustrato nel capitolo precedente. Questo capitolo rappresenta l'inizio dell'effettivo lavoro di tesi, in cui si progetta un Current Starved VCO e si analizza il funzionamento dell'intero riferimento di tensione virtuale al variare delle caratteristiche interne. In particolare, vengono descritti la struttura a blocchi circuitale del riferimento virtuale ed il design del CSVCO, l'algoritmo usato per ricavare il riferimento virtuale, e il flow chart usato nel codice MATLAB, presente in appendice.

4.1 Struttura

Con riferimento e analogia alla struttura del riferimento di tensione virtuale della figura 3.3, viene di seguito descritta la struttura a blocchi del riferimento VCO-based.

La funzione svolta precedentemente dall'ADC viene ora eseguita dalla combinazione di diversi blocchi: un oscillatore CSVCO (Current Starved Voltage Controlled Oscillator), la cui frequenza viene confrontata con quella di un altro oscillatore tramite una porta logica AND, è collegato ad un contatore digitale. Non viene utilizzato nessun resistore e la funzione di standard fisico viene eseguita da un transistor n-MOS all'interno del CSVCO. Viene sfruttato un DAC per convertire i numeri digitali, provenienti da un processore digitale, in tensioni analogiche e la tensione di riferimento del DAC coincide con la tensione V_{dd} , applicata anche al sistema ADC.

Dalla figura 4.1, riportante lo schema a blocchi dell'architettura, si nota che la tensione V_{dac} , in uscita dal DAC, che converte il numero digitale m proveniente dal processore, pilota il CSVCO. Esso fornisce in uscita un segnale, oscillante ad una data frequenza, che viene trasformato in un'onda quadra dal buffer di uscita. La frequenza del segnale in uscita dal buffer viene confrontata con quella di un oscillatore locale tramite una porta logica AND che pilota il segnale di clock all'ingresso del contatore. Lo scopo è quello di contare, ossia di fornire al processore, sotto forma di numero digitale n , il numero che indica la frequenza del segnale in uscita dal CSVCO e dal buffer rispetto alla frequenza dell'oscillatore locale. Il processore digitale anche in questa configurazione funge da blocco di signal processing.

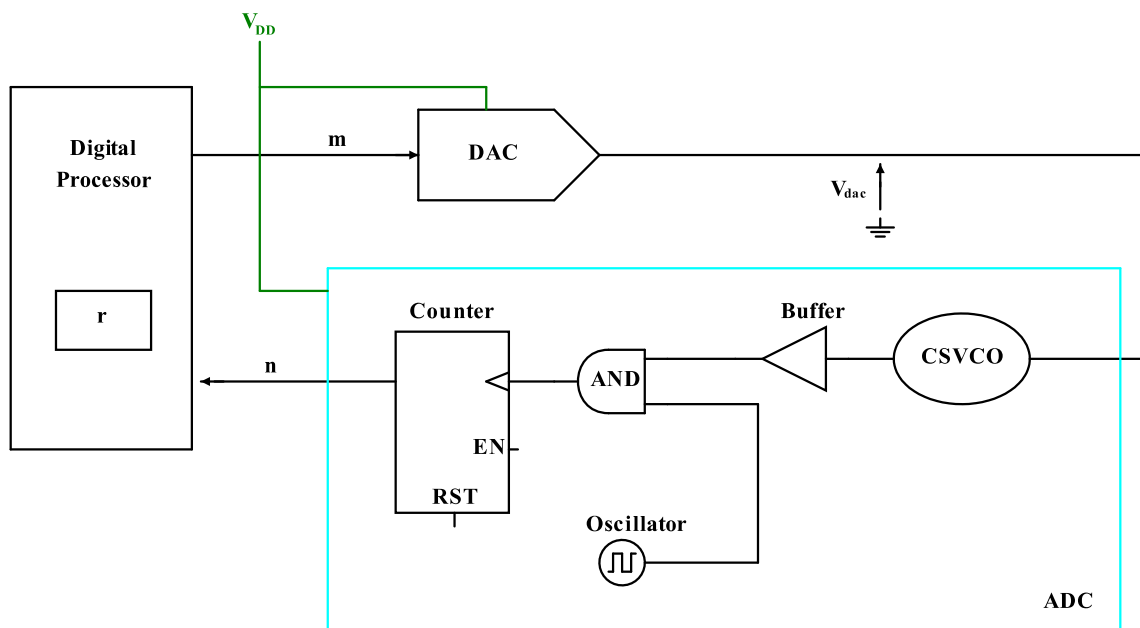


Figura 4.1. Sistema di riferimento

4.2 CSVCO

Il Current-Starved VCO, raffigurato nelle figura 4.2, è un particolare tipo di oscillatore, appartenente alla categoria degli oscillatori non lineari, che sfrutta i transistori M2-M3 e gli analoghi transistori in figura in configurazione inverter, mentre i transistori M1 ed M4 operano come erogatore ed assorbitore di corrente. La catena di inverter viene alimentata dalla corrente, generata dal ramo comprendente i transistori M6-M5, che viene specchiata dalla rete di pMOS ed nMOS nella parte superiore e inferiore del circuito. Di fatto, ogni stadio di inverter, a differenza di un comune oscillatore ad anello, riceve una corrente controllata dalla tensione di ingresso V_{in} , nominalmente uguale a tutti gli nMOS che compongono lo specchio di corrente della parte inferiore del circuito.

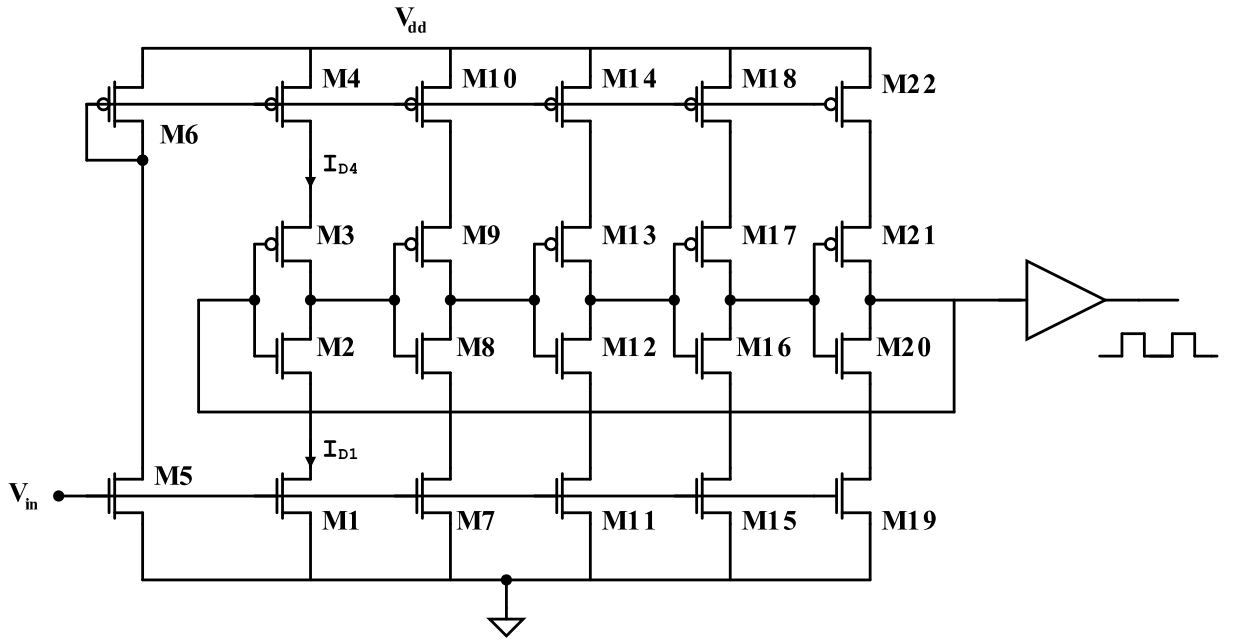


Figura 4.2. CSVCO con 5 stage dotato di un buffer in uscita

Per studiare il funzionamento del CSVCO si può fare riferimento alla figura 4.3.

La capacità totale C_{tot} sul drain dei due generici transistori, che formano l'inverter, coincide con la somma delle capacità di ingresso e di uscita dell'inverter. Si definisce V_{SP} la tensione di inversione (switching point) in ingresso (o uscita) dell'inverter, ossia il punto sulla caratteristica ingresso-uscita dell'inverter tale per cui la tensione d'ingresso è pari alla tensione di uscita:

$$V_{SP} = \frac{\sqrt{\frac{\beta_n}{\beta_p}} \cdot V_{THn} + (V_{DD} - V_{THp})}{1 + \sqrt{\frac{\beta_n}{\beta_p}}} \quad (4.1)$$

Il periodo di tempo richiesto per caricare C_{tot} dalla tensione nulla a V_{SP} a corrente costante I_{D4} , viene indicato con t_1 e mostrato di seguito:

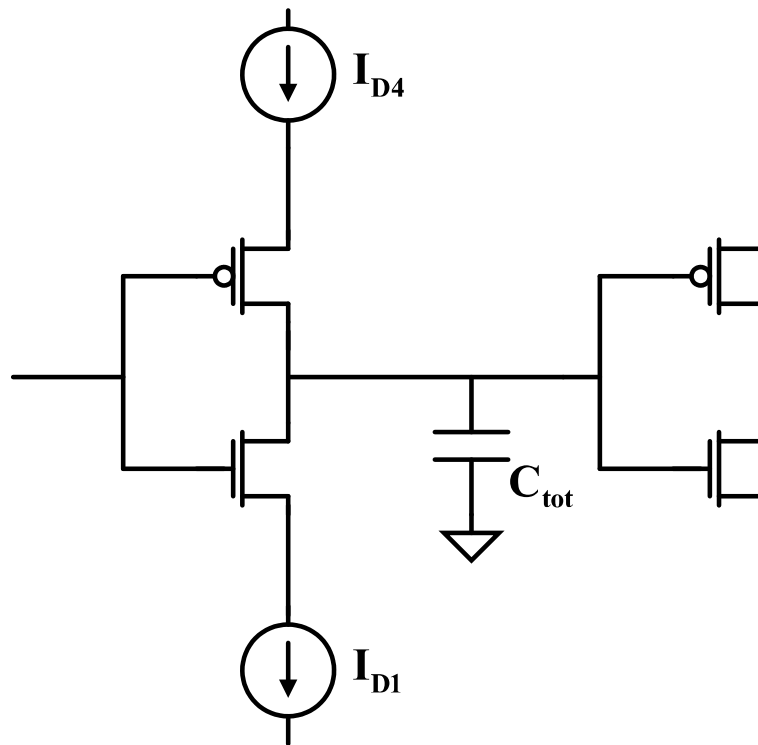


Figura 4.3. Vista semplificata del singolo stadio del CSVCO

$$t_1 = C_{tot} \cdot \frac{V_{SP}}{I_{D4}} \quad (4.2)$$

Il tempo richiesto per scaricare il condensatore di capacità C_{tot} da V_{dd} a V_{SP} è dato da t_2 :

$$t_2 = C_{tot} \cdot \frac{V_{dd} - V_{SP}}{I_{D1}} \quad (4.3)$$

Settando $I_{D4}=I_{D1}=I_D$ allora la somma dei due tempi sopracitati vale:

$$t_1 + t_2 = \frac{C_{tot} \cdot V_{dd}}{I_D} \quad (4.4)$$

Indicando con N il numero dispari di stage contenenti il singolo inverter e i due generatori di corrente, si ricava la frequenza di oscillazione del CSVCO f_{osc} :

$$f_{osc} = \frac{1}{N \cdot (t_1 + t_2)} = \frac{I_D}{N \cdot C_{tot} \cdot V_{dd}} \quad (4.5)$$

Dalla formula si nota che la frequenza di oscillazione dipende in modo direttamente proporzionale dalla corrente I_D che scorre negli inverter. A sua volta la corrente I_D è pilotata dalla tensione di ingresso V_{in} . Di conseguenza, come il nome stesso suggerisce, la frequenza in uscita dal CSVCO dipende dalla tensione di ingresso applicata.

4.2.1 Progettazione

La progettazione del CSVCO e le successive simulazioni sono state effettuate con il supporto dell'ambiente Cadence utilizzando una tecnologia CMOS a 65 nm. Le principali specifiche sono le seguenti:

- Minima potenza dissipata
- Minima area occupata
- Tensioni di alimentazioni ridotte per processi a canale corto

Con lo scopo di ridurre la potenza dissipata si è scelto di fare operare i transistori in regione di funzionamento sotto soglia e con tensioni di alimentazione V_{dd} che variano tra 0.5V e 1V. La tensione di ingresso dell'oscillatore, infatti, coincidente con la tensione V_{gs} dei transistori nMOS dello specchio di corrente, è controllata in modo da non superare la tensione di soglia di circa 0.5V.

Con riferimento alla figura 4.2, nella quale non sono mostrati i terminali di body degli nMOS a ground e dei pMOS alla tensione di alimentazione, sono stati utilizzati transistori SVT (Standard Threshold Voltage), tutti dotati di una lunghezza di canale L pari a 65 nm, al fine di occupare la minima area sul silicio.

Per quanto riguarda il dimensionamento dei singoli transistori degli inverter, al fine di compensare il rapporto tra le mobilità degli nMOS e dei pMOS, per ottenere gli stessi tempi di salita e di discesa sull'uscita, i pMOS sono stati dimensionati con una larghezza W doppia rispetto a quella degli nMOS. Si è scelto di dimensionare i transistori che operano

da generatori di corrente con delle larghezze doppie rispetto a quelle dei transistori che compongono gli inverter, in modo tale da assicurare una minore resistenza di ingresso. Come risultato, la corrente fornita attraverso i 5 stage risulta più robusta indipendentemente dalle condizioni di pilotaggio [28].

La tabella 4.1 riporta i valori delle larghezze di tutti i transistori.

Transistor	W
M6,M4,M10,M14,M18,M22	$0.540\mu\text{m}$
M3,M9,M13,M17,M21	$0.270\mu\text{m}$
M2,M8,M12,M16,M20	$0.135\mu\text{m}$
M5,M1,M7,M11,M15,M19	$0.270\mu\text{m}$

Tabella 4.1. Larghezze dei transistori

Inoltre, per meglio apprezzare la forma d'onda quadra in uscita dal CSVCO, sono stati posti due inverter in cascata, non visibili nella figura 4.2, di dimensioni uguali a quelle degli inverter dell'oscillatore stesso. Di fatto non è stato eseguito nessun progetto del buffer in uscita dal CSVCO in quanto, come spiegato nel seguito, gli altri blocchi del sistema in figura 4.1 sono stati assunti ideali.

4.2.2 Simulazioni

Di seguito vengono presentate alcune figure che riportano, oltre alla forma d'onda in uscita, il comportamento della frequenza di uscita dell'oscillatore al variare della tensione di alimentazione e della tensione di ingresso e della temperatura.

La figura 4.4 riporta la relazione ingresso-uscita dell'oscillatore alla tensione di alimentazione di 0.75V. Si nota la sua natura non lineare e il range esteso di frequenze che varia da pochi MHz a centinaia di MHz.

Risulta interessante riportare anche il comportamento della frequenza di uscita f_{osc} al variare della tensione di alimentazione, considerando costante l'ingresso V_{in} .

Il risultato della figura 4.5 dimostra che, nonostante nell'espressione 4.5 la f_{osc} sia inversamente proporzionale alla tensione di alimentazione, un aumento della V_{dd} comporta un ulteriore aumento della corrente I_D , dovuto all'effetto DIBL (Drain-Induced Barrier Lowering), che fa crescere la frequenza di oscillazione. Il comportamento della frequenza di oscillazione rispetto alle variazioni della temperatura nel range tra -20°C e 120°C viene mostrato nella figura 4.6. Si osserva che la frequenza aumenta con la temperatura, in quanto i generatori di corrente MOS dell'oscillatore CSVCO sono in regione sub-threshold, in cui la conduzione aumenta all'aumentare della temperatura [20].

Inoltre, dalla figura 4.7 si apprezza la forma d'onda in uscita dall'oscillatore in funzione del tempo nelle condizioni $V_{dd}=0.75\text{V}$ e $V_{in}=0.3\text{V}$ e a 27°C . In questo caso la frequenza di uscita f_{osc} risulta essere di circa 11 MHz.

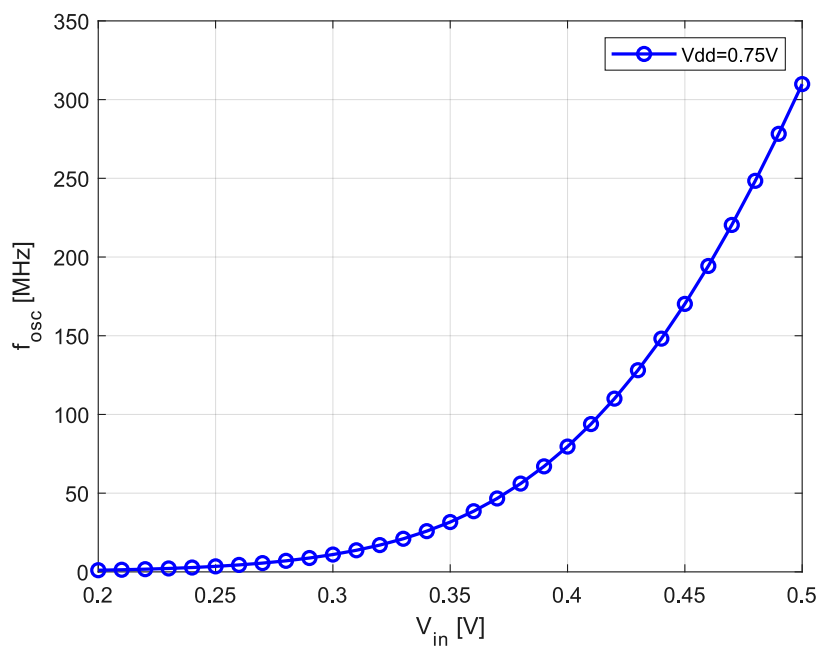


Figura 4.4. Frequenza di uscita dell'oscillatore in funzione della tensione di ingresso con $V_{dd}=0.75V$

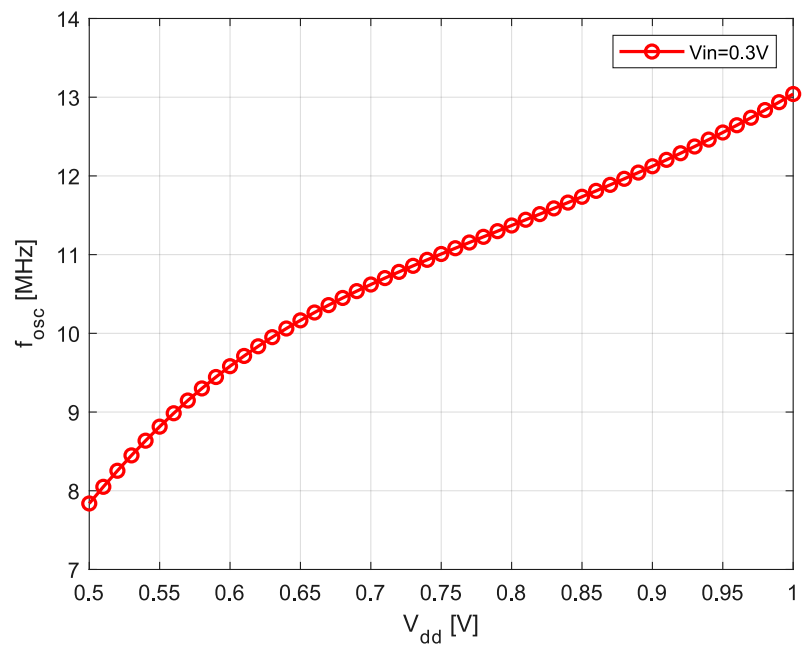


Figura 4.5. Frequenza di uscita dell'oscillatore in funzione della tensione di alimentazione con $V_{in}=0.3V$

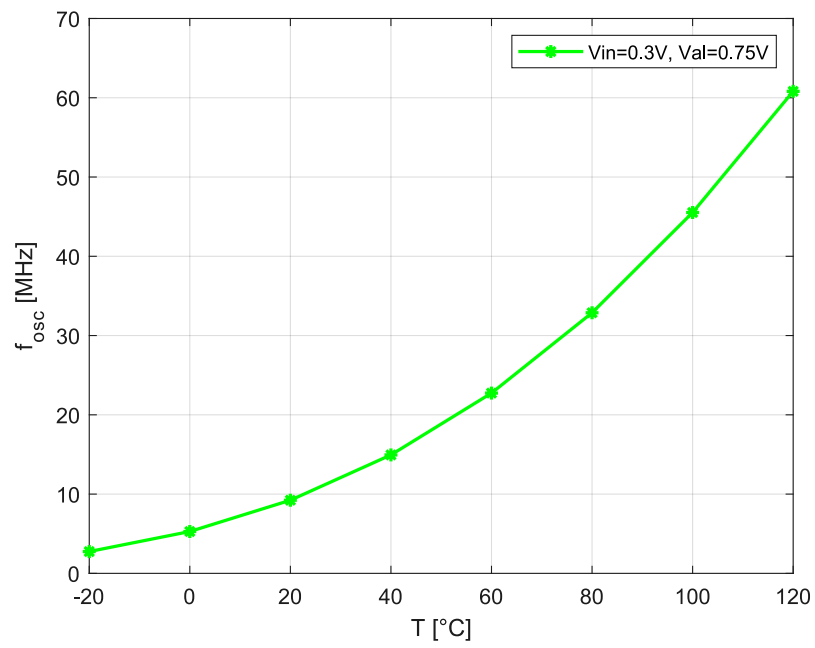


Figura 4.6. Frequenza di uscita dell'oscillatore in funzione della temperatura con $V_{dd}=0.75V$ e $V_{in}=0.3V$

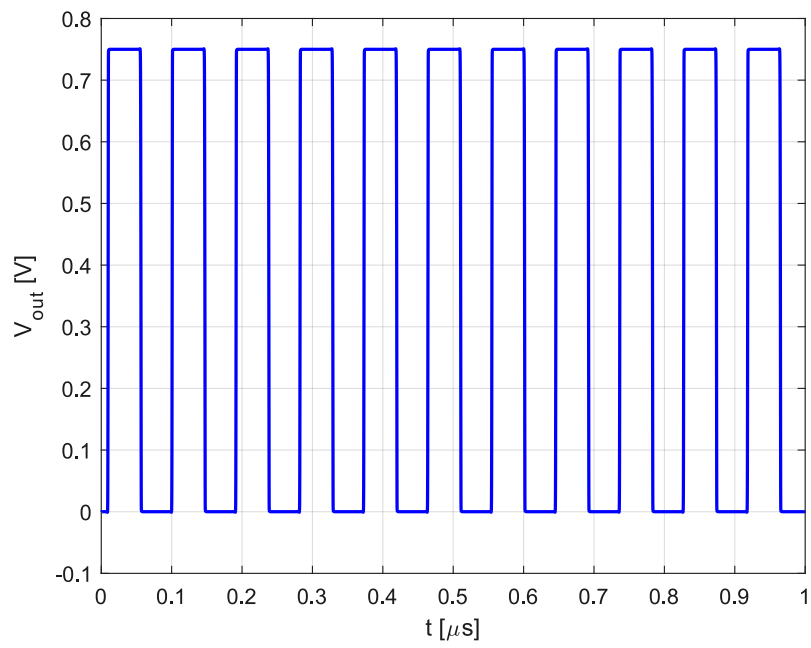


Figura 4.7. Tensione di uscita dell'oscillatore in funzione della tempo con $V_{dd}=0.75\text{V}$ e $V_{in}=0.3\text{V}$

4.3 VCO-based ADC

Come già anticipato in precedenza, a differenza della struttura hardware del riferimento di tensione virtuale della figura 3.3, la funzione svolta dall'ADC, che consiste nel convertire il segnale analogico, proveniente dal DAC, in un numero digitale n da inviare al processore, viene ora eseguita essenzialmente dalla combinazione del CSVCO e di un contatore digitale. Con riferimento alla figura 4.1, si nota che la struttura del sistema ADC ricorda la composizione di un VCO-based quantizer [29].

Lo scopo del CSVCO è quello di convertire il segnale analogico di ingresso proveniente dal DAC, che a sua volta riceve il numero digitale m dal processore, in un segnale di uscita a frequenza f_{osc} proporzionale al valore in continua dell'ingresso. Prendendo in considerazione un oscillatore locale che produce un segnale, di periodo T , maggiore rispetto a quello fornito dal CSVCO, una porta AND presenta ai suoi due ingressi l'uscita del CSVCO e dell'oscillatore locale. Come mostrato nella figura 4.8, il compito della porta AND consiste nel fornire un segnale di uscita contenente i fronti di salita del segnale proveniente dal CSVCO contenuti in un periodo del segnale prodotto dall'oscillatore locale.

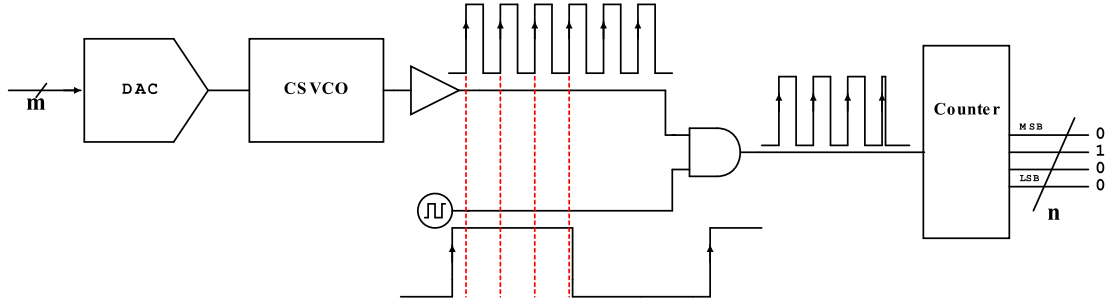


Figura 4.8. Schema a blocchi del sistema, escluso il processore

Il compito del contatore digitale, opportunamente controllato, consiste nel contare i fronti di salita del segnale di ingresso, che funge da clock, e fornire in uscita al processore il numero n . Nell'esempio della figura 4.8 ai 4 fronti di salita corrisponde in uscita dal contatore il numero digitale 0100, ossia 4. Siccome l'ADC VCO-based è formato prevalentemente da componenti digitali, la sua performance trae vantaggio dallo scaling della tecnologia, a differenza degli ADC tradizionali.

4.3.1 Sottosistemi del riferimento virtuale VCO-based

Ai fini dell'obiettivo di questo lavoro di tesi, non è stato eseguito nessun progetto dei blocchi all'interno del riferimento di tensione, ad esclusione del CSVCO.

- Il DAC, sebbene possa essere implementato attraverso una soluzione completamente digitale [30], viene considerato nelle simulazioni come un blocco ideale, tale per cui ad ogni numero digitale di ingresso m corrisponde un'uscita analogica V_{dac} :

$$V_{dac} = \frac{m}{2^N} V_{dd} + V_{\epsilon} \quad (4.6)$$

in cui V_{dd} è la tensione di riferimento, N rappresenta la risoluzione del DAC e V_e rappresenta il termine di errore che comprende tutte le non idealità del DAC. Si assume di utilizzare un DAC con una risoluzione effettiva tale per cui il termine V_e risulta trascurabile.

- Tutti i blocchi dell'ADC VCO-based, escluso il CSVCO, sono stati considerati ideali. Ai fini delle simulazioni il parametro preso in considerazione risulta essere la frequenza f_{osc} in uscita dal CSVCO. Inoltre, solamente la risoluzione del contatore $N_{counter}$, il periodo T e la risoluzione N del DAC rappresentano due ulteriori parametri usati per effettuare una stima di massima delle prestazioni del sistema, al fine di porre le basi per una futura completa implementazione della struttura.

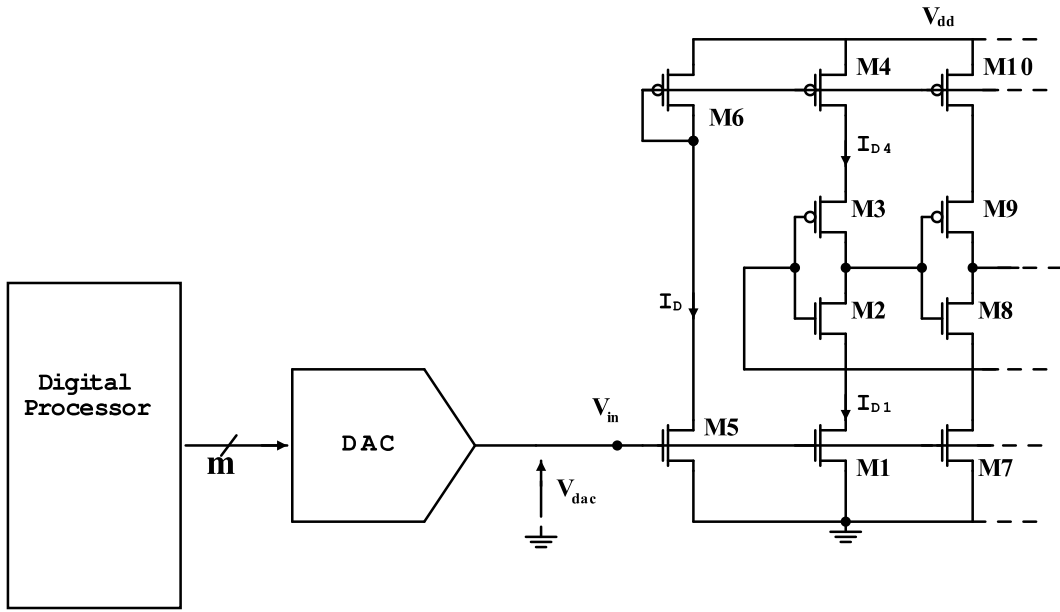


Figura 4.9. L'ingresso del CSVCO coincide con l'uscita del DAC

4.4 Analisi della struttura

Come anticipato nella sezione della progettazione del CSVCO, i transistori che formano l'oscillatore vengono fatti lavorare in regione di funzionamento sotto soglia al fine di dissipare la minima energia. Sotto questa condizione risulta utile riportare il modello matematico che approssima il comportamento della corrente sotto soglia I_{sub} [8], trascurando l'effetto DIBL:

$$I_{sub} = \mu C_{ox} \frac{W}{L} (\eta - 1) V_T^2 \cdot \exp\left(\frac{V_{gs} - V_{th}}{\eta V_T}\right) \left(1 - \exp\left(\frac{-V_{ds}}{V_T}\right)\right) \quad (4.7)$$

in cui μ è la mobilità degli elettroni, C_{ox} è la capacità dell'ossido di gate per unità di area, W e L sono rispettivamente la larghezza e la lunghezza del transistor, η è il subthreshold slope factor, V_T è l'equivalente in tensione della temperatura pari a $\frac{kT}{q}$, in cui k è la costante di Boltzmann, T è la temperatura, q è la carica dell'elettrone, V_{th} è la tensione di soglia del transistor.

Dall'espressione (4.5) si ricava la seguente relazione tra la corrente I_D che scorre nel generico transistor e la frequenza di oscillazione in uscita dal CSVCO:

$$I_D = f_{osc} \cdot N \cdot C_{tot} \cdot V_{dd} \quad (4.8)$$

In particolare, presa in considerazione la figura 4.9, si nota come la tensione d'ingresso del CSVCO (V_{in}) coincida sia con la tensione in uscita dal DAC (V_{dac}), esprimibile come nell'espressione (4.6), sia con la tensione V_{gs} dei transistori nMOS al di sotto degli stadi di inverter.

Di conseguenza è lecito affermare la seguente uguaglianza tra le correnti delle espressioni (4.8) e (4.7):

$$f_{osc} \cdot N \cdot C_{tot} \cdot V_{dd} = I_0 \cdot \exp\left(\frac{V_{gs}}{\eta V_T}\right) \quad (4.9)$$

in cui il termine $I_0 = \mu C_{ox} \frac{W}{L} (\eta - 1) V_T^2 \exp(\frac{-V_{th}}{\eta V_T})$ e il termine $(1 - \exp(\frac{-V_{ds}}{V_T}))$ è stato considerato trascurabile, adottando delle tensioni $V_{in} = V_{gs} > 4 V_T$.

Un ulteriore step consente di esprimere, in generale, la frequenza di oscillazione del CSVCO in funzione del numero digitale m in ingresso al DAC e proveniente dal processore:

$$f_{osc} \cdot N \cdot C_{tot} \cdot V_{dd} = I_0 \cdot \exp\left(\frac{m \cdot V_{dd}}{\eta V_T}\right) \quad (4.10)$$

Considerando il blocco contatore e la porta AND ideali, il numero digitale n in uscita dal contatore si può esprimere come:

$$n = \frac{f_{osc}}{f_R} = f_{osc} \cdot T \quad (4.11)$$

in cui f_R è la frequenza del segnale prodotto dall'oscillatore locale. Di conseguenza applicati due diversi numeri digitali $m^{(1)}$ ed $m^{(2)}$ in ingresso al DAC, sono valide le seguenti relazioni:

$$\begin{cases} n^{(1)} \cdot f_R \cdot N \cdot C_{tot} \cdot V_{dd} = I_0 \cdot e^{\left(\frac{m^{(1)} \cdot V_{dd}}{\eta V_T \cdot 2^N}\right)} \\ n^{(2)} \cdot f_R \cdot N \cdot C_{tot} \cdot V_{dd} = I_0 \cdot e^{\left(\frac{m^{(2)} \cdot V_{dd}}{\eta V_T \cdot 2^N}\right)} \end{cases} \quad (4.12)$$

Eseguendo il rapporto tra le due espressioni si ottiene il rapporto tra i due numeri digitali $n^{(1)}$ ed $n^{(2)}$:

$$\frac{n^{(1)}}{n^{(2)}} = e^{\frac{V_{dd}}{\eta V_T \cdot 2^N} (m^{(1)} - m^{(2)})} \quad (4.13)$$

Eseguendo l'operazione di logaritmo da entrambi i membri dell'espressione (4.13), si ottiene:

$$\log \left(\frac{n^{(1)}}{n^{(2)}} \right) = \frac{V_{dd}}{\eta V_T \cdot 2^N} (m^{(1)} - m^{(2)}) \quad (4.14)$$

L'espressione appena ricavata si può riscrivere come di seguito:

$$V_{dd} = \log \left(\frac{n^{(1)}}{n^{(2)}} \right) \frac{2^N}{(m^{(1)} - m^{(2)})} \eta V_T \quad (4.15)$$

Si nota che la tensione di alimentazione V_{dd} dipende dalla tensione ηV_T tramite un coefficiente che dipende solo dai numeri digitali $m^{(1)}$, $m^{(2)}$, $n^{(1)}$, $n^{(2)}$, che possono essere elaborati dal processore. Dunque, data un'opportuna coppia di numeri digitali ($m^{(1)}$, $m^{(2)}$), la tensione di alimentazione può essere espressa in funzione di una tensione indipendente dall'alimentazione stessa.

Nel seguito la relazione (4.15) viene utilizzata nella seguente forma compatta:

$$V_{dd} = A \cdot V_0 \quad (4.16)$$

in cui $A = \log \left(\frac{n^{(1)}}{n^{(2)}} \right) \frac{2^N}{(m^{(1)} - m^{(2)})}$ e $V_0 = \eta V_T$.

4.4.1 Riferimento virtuale

Al fine di ricavare un'espressione per il riferimento virtuale r per questa struttura, basata sull'oscillatore CSVCO, si riporta la relazione fondamentale (3.15), adeguatamente aggiornata per tale trattazione:

$$V_{REF} = \frac{r}{2^N} V_{dd} \quad (4.17)$$

in cui V_{REF} indica quella tensione PVT-invariante che si ricava a partire dallo schema a blocchi del riferimento bandgap di Kuijk.

In analogia con la (4.16), si nota che la tensione V_0 svolge un ruolo analogo a quello della tensione V_{REF} , essendo entrambe note e proporzionali alla tensione di alimentazione tramite un numero digitale.

Inizialmente, l'obiettivo consiste nel definire un riferimento proporzionale alla tensione V_0 dell'espressione (4.16), dunque direttamente proporzionale alla temperatura. Inoltre, sfruttando l'approccio algoritmico su cui si basa il funzionamento del riferimento di tensione

virtuale [10], mostrato nel capitolo precedente, si potrà estendere in futuro questo approccio per generare un riferimento virtuale indipendente dalla temperatura.

Nel seguito, dunque, imponendo l'uguaglianza $V_{REF}=V_0$, si ottiene:

$$V_{REF} = \frac{r}{2^N} V_{dd} = V_0 = \frac{V_{dd}}{A} \quad (4.18)$$

Si considera, di fatto, una V_{REF} non solo proporzionale, ma anche coincidente alla tensione $V_0=\eta V_T$. Proseguendo nell'analisi, dalla (4.18), il riferimento virtuale r , per la struttura basata sul CSVCO, può essere espresso come:

$$r = \frac{2^N}{A} \quad (4.19)$$

in cui $A=\log\left(\frac{n^{(1)}}{n^{(2)}}\right)\frac{2^N}{(m^{(1)}-m^{(2)})}$. Si nota che il riferimento virtuale dipende solo da numeri digitali processati dal processore.

Riutilizzando la relazione (4.17), la tensione di riferimento si esprime come:

$$V_{REF} = \frac{\frac{2^N}{A}}{2^N} V_{dd} = \frac{V_{dd}}{A} \quad (4.20)$$

In analogia con la (3.20), una volta ricavato il riferimento virtuale r , grazie ad opportuni valori di $m^{(1)}$, $m^{(2)}$ usati nell'algoritmo di compensazione spiegato di seguito, è possibile calcolare il codice m da convertire con il DAC in modo da ottenere in uscita una tensione proporzionale alla tensione di riferimento voluta, indipendentemente dalle variazioni di V_{dd} . Infatti, noto a priori il valore della tensione di riferimento V_{REF} , la tensione in uscita dal DAC si esprime come di seguito:

$$V_{dac} = \frac{m \cdot A \cdot 2^N}{2^N} V_{REF} = \frac{m}{r} V_{REF} \quad (4.21)$$

Dall'espressione (4.21), infatti, si nota che per ricavare una tensione in uscita dal DAC riferita a V_{REF} , dopo che il riferimento virtuale è stato ricavato, basta che il processore invii un numero digitale m tale per cui

$$m = \alpha \cdot r \quad (4.22)$$

dove $\alpha = V_{dac}/V_{REF}$.

4.5 Procedura di compensazione

Di seguito viene presentata, con riferimento alla figura 4.11, la procedura di compensazione che permette di ricavare il riferimento virtuale r . In analogia con la procedura della figura 3.4, inizialmente vengono generati all'interno del processore due numeri interi digitali $m^{(1)}$ ed $m^{(2)}$. In seguito il numero $m^{(1)}$ viene inviato al DAC, producendo in uscita una tensione V_{dac} che pilota la frequenza di oscillazione f_{osc} del CSVCO. In uscita dal contatore, e dunque dal sistema ADC, è presente il numero digitale $n^{(1)}$ che viene inviato al processore, che mantiene salvato in un registro interno sia il numero $m^{(1)}$ che il numero $n^{(1)}$. Viene ripetuta la stessa procedura anche per il numero digitale $m^{(2)}$, e dunque all'interno del

processore viene salvato anche il numero $n^{(2)}$.

Successivamente il processore calcola il numero digitale $A = \log\left(\frac{n^{(1)}}{n^{(2)}}\right) \frac{2^N}{(m^{(1)} - m^{(2)})}$, e contestualmente si ottiene il primo riferimento virtuale $r = \frac{2^N}{A}$, in cui N è il numero di bit del DAC. La procedura viene rieseguita con altri due numeri digitali $m^{(1)} = \lfloor k_1 r \rfloor$ ed $m^{(2)} = \lfloor k_2 r \rfloor$, riferiti al riferimento virtuale r appena discusso attraverso i numeri k_1 e k_2 , che risultano utili per ottenere nuovi numeri $n^{(1)}$ e $n^{(2)}$ sufficientemente diversi, limitando gli errori legati alla risoluzione del conteggio. Risulta opportuno che i numeri $m^{(1)}$ e $m^{(2)}$ risultino proporzionali al riferimento virtuale r in modo tale da fissare la tensione di ingresso del CSVCO a valori rientranti nel range di validità del modello, indipendentemente dalle variazioni della tensione di alimentazione.

Viene ricalcolato un nuovo riferimento virtuale r e viene confrontato con quello calcolato nella fase precedente r_{old} . Se la differenza $|r - r_{old}| < \epsilon$ allora il riferimento virtuale risulta essere stato trovato e il prodotto rV_{dd} rimane costante. Al contrario verrà utilizzata una nuova coppia $m^{(1)} = \lfloor k_1 r \rfloor$ ed $m^{(2)} = \lfloor k_2 r \rfloor$ finché non viene raggiunta la convergenza. La figura 4.10 mostra un esempio di derivazione del riferimento virtuale r , attraverso cicli successivi, considerando una tensione di alimentazione di 0.5V. Si nota come occorrono sei cicli nel processo di compensazione per la definizione del riferimento virtuale $r = 1329$.

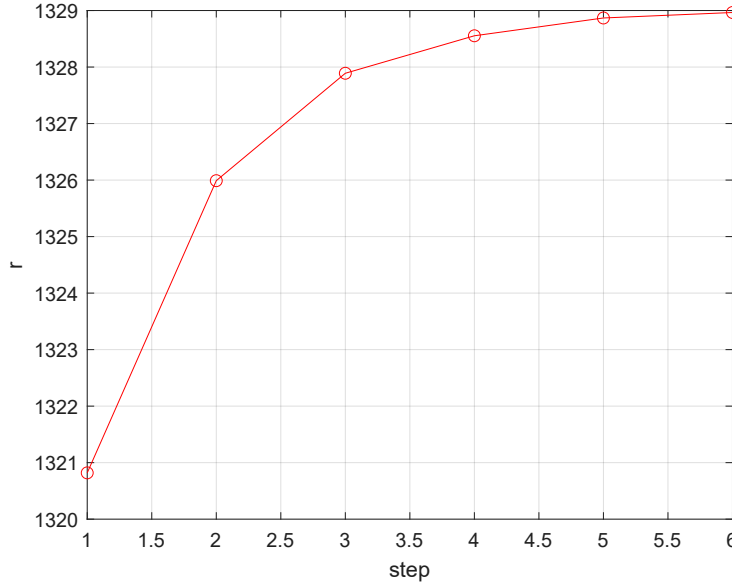
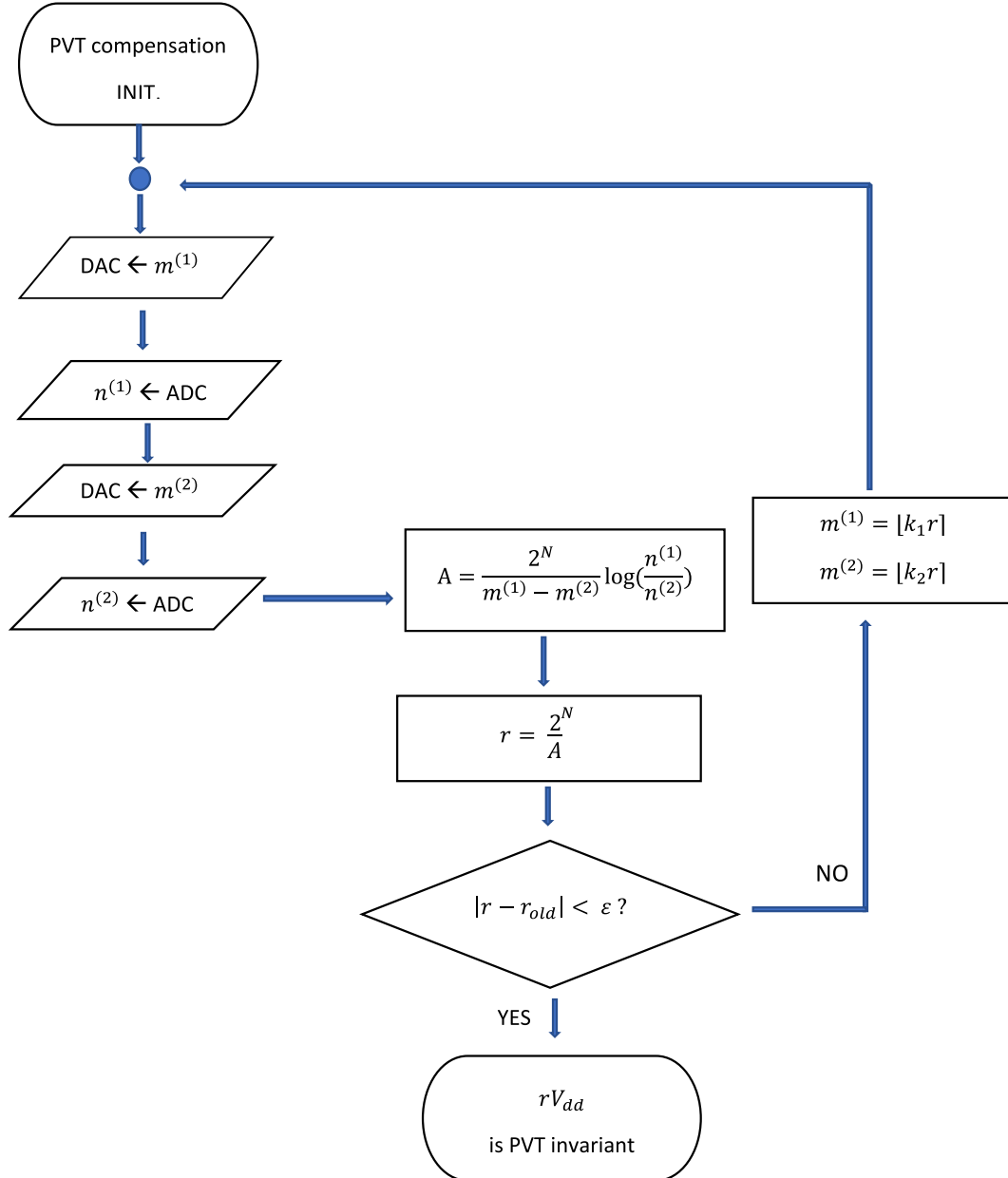


Figura 4.10. Esempio di convergenza nella definizione del riferimento virtuale r

Figura 4.11. Algoritmo per la ricerca del riferimento virtuale r in un riferimento VCO-based

4.6 Organizzazione delle simulazioni

L'obiettivo principale del lavoro di tesi consiste nell'effettuare delle simulazioni finalizzate a testare le performance della struttura VCO-based sotto le diverse condizioni PVT. A questo scopo è stato sfruttato l'ambiente Cadence Virtuoso per la raccolta dei dati delle simulazioni, in seguito esportati ed organizzati in un database all'interno del programma MATLAB, in cui si effettua il processing delle informazioni.

Nota

Sebbene sia stato riportato lo schema a blocchi della struttura del riferimento VCO-based della figura 4.1, le simulazioni sono state effettuate unicamente sulla struttura del CSVCO riportato in 4.12, supponendo gli altri blocchi, tra cui il contatore, il DAC, e l'oscillatore locale ideali. Uno studio successivo è stato effettuato sulla risoluzione del DAC N e sul numero di bit $N_{counter}$ del contatore, in relazione al periodo T dell'oscillatore locale.

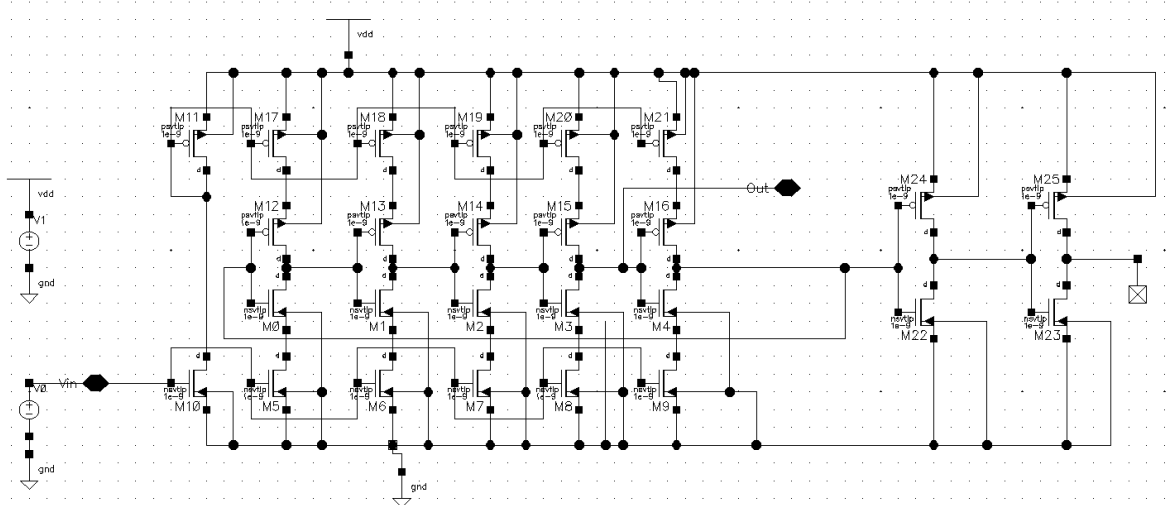


Figura 4.12. Struttura simulata

Il DAC, ad esempio, è stato considerato come un generatore ideale di tensione che produce la tensione V_{dac} :

$$V_{dac} = \frac{m}{2^N} V_{dd} \quad (4.23)$$

posta all'ingresso del CSVCO. Per garantire che i transistori lavorino nella regione di funzionamento sotto soglia, siccome la tensione di soglia degli nMOS è di circa 0.55V e dato che la massima tensione di alimentazione applicata risulta essere di 1 V, si è considerato un range di valori di m che spaziano tra 3200 e 8192, supponendo di utilizzare un DAC a 14 bit. Inoltre, nel caso delle simulazioni, con il numero n si intende la frequenza di uscita f_{osc} del CSVCO rispetto al periodo $T=1s$, che coincide con la finestra temporale in cui è stato contato il numero di periodi del segnale di uscita del CSVCO.

4.6.1 Matrice delle frequenze

Nell'ambiente ADE XL di Cadence Virtuoso sono state effettuate diverse simulazioni parametriche riguardanti, essenzialmente, il calcolo della frequenza f_{osc} al variare della tensione di alimentazione V_{dd} tra 0.5V e 1V, alla temperatura $T = 27^\circ\text{C}$. Sono state eseguite anche simulazioni a temperature differenti e, inoltre, sono state considerate le variazioni dei parametri di processo. Principalmente lo scopo riguarda il calcolo del riferimento virtuale r per verificare la procedura di compensazione illustrata nella figura 4.11. Al termine delle simulazioni i dati sono stati raccolti secondo una logica ben precisa: per ogni singola tensione di alimentazione sono stati importati in MATLAB diversi vettori contenenti le frequenze di uscita del CSVCO, ossia i diversi numeri n , al variare dei numeri digitali m in ingresso al DAC. In seguito, tutti i vettori sono stati ordinati in un'unica matrice. Essa è composta da 4993 righe e 51 colonne per un totale di 254643 frequenze. Su ogni riga viene considerato uno dei 4993 numeri digitali m in ingresso al DAC, mentre su ogni colonna viene considerata un'unica tensione di alimentazione, come mostrato nella figura 4.13. Nelle simulazioni, infatti, è stato considerato un passo di 0.01V della tensione di alimentazione e un passo di 1 LSB per i numeri digitali m .

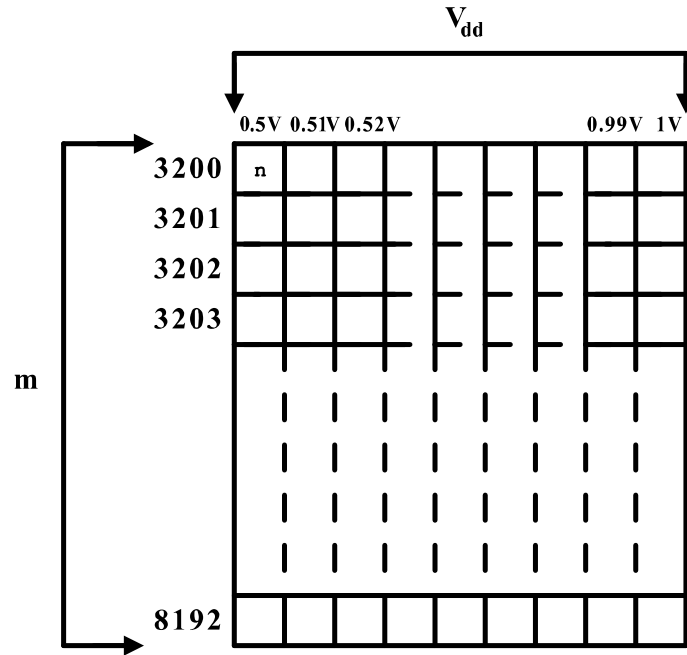


Figura 4.13. Tabella delle frequenze simulate al variare di V_{dd}

Ad esempio, il numero corrispondente alla prima riga e alla prima colonna della matrice corrisponde alla frequenza di oscillazione f_{osc} , che si ottiene quando all'ingresso del DAC viene posto il numero 3200 e la $V_{dd} = 0.5\text{V}$.

Di seguito viene riportato il flow-chart, riferito allo script MATLAB in appendice A, utilizzato per il calcolo del riferimento virtuale e della tensione di riferimento V_{REF} . Si tratta di un algoritmo applicato in seguito a tutte le altre simulazioni.

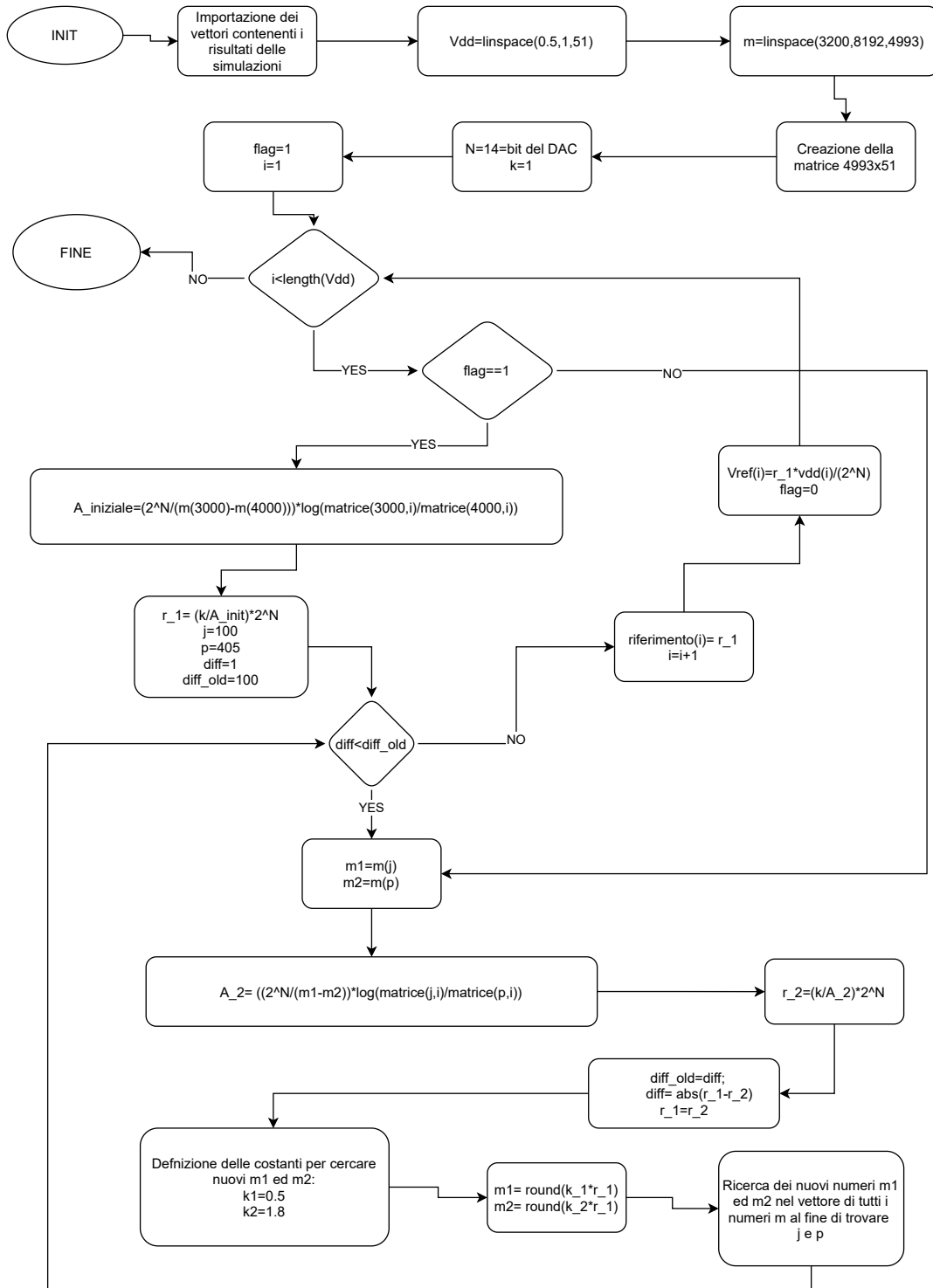


Figura 4.14. Flow-chart della procedura di compensazione in ambiente MATLAB

4.6.2 Flow-Chart e algoritmo

Con riferimento al codice MATLAB presente in appendice A viene di seguito discusso l'algoritmo utilizzato per testare le performance della struttura. Dopo la fase di importazione dei risultati delle simulazioni effettuate in Cadence, tutti i vettori vengono organizzati nella matrice 4.13 discussa in precedenza. Vengono creati i vettori relativi ai numeri digitali m e il vettore delle tensioni di alimentazione V_{dd} . Inizialmente è stato supposto che il DAC lavorasse con una risoluzione $N=14$ bit e che il contatore contasse idealmente la frequenza stessa del segnale in uscita dal CSVCO. Si inizializza la variabile i , usata come indice per scorrere il vettore delle tensioni di alimentazione, pari ad 1 e viene creata la variabile *flag* che viene sfruttata per indicare di essere all'inizio dell'algoritmo.

Viene impiegato un ciclo *for* per eseguire l'algoritmo su ogni tensione di alimentazione. A partire dalla tensione di 0.5V, all'inizio l'algoritmo seleziona due numeri pseudo-casuali presi dal vettore dei numeri m . Come si nota dal flow-chart 4.14, viene calcolato il numero digitale $A_{iniziale}$ a partire dai numeri $m(3000)$ ed $m(4000)$ insieme alle rispettive frequenze calcolate nelle condizioni in cui $V_{dd}=0.5V$ e $m^{(1)}=m(3000)$ e $m^{(2)}=m(4000)$. Dunque le frequenze vengono selezionate dalla matrice come $matrice(3000,i)$ e $matrice(4000,i)$.

In seguito viene calcolato il possibile riferimento r_1 e vengono decisi gli indici (j e p) utili per la selezione della prossima coppia di numeri $m^{(1)}$ $m^{(2)}$. Vengono inoltre inizializzati i parametri *diff* e *diff_{old}* al fine del calcolo del riferimento virtuale r .

Attraverso un ciclo *while* viene ricalcolato il numero A , attraverso la definizione di due nuovi numeri m a partire dal riferimento trovato nel ciclo precedente moltiplicato per i fattori k_1 e k_2 , e il rispettivo riferimento virtuale r fino a quando i riferimenti virtuali di due cicli successivi non raggiungono la convergenza. Essa viene raggiunta nel momento in cui la differenza dei riferimenti virtuali di due cicli successivi non diminuisce ulteriormente. Infine, ricavato il riferimento virtuale, si aggiorna l'indice i per simulare il passaggio ad un'altra tensione di alimentazione. Contemporaneamente vengono salvati in un vettore sia il riferimento virtuale che il calcolo di V_{REF} . Risulta importante mettere in evidenza, ai fini della comprensione, che il calcolo del riferimento virtuale r , per ogni singolo ciclo di operazioni che conduce alla convergenza, viene effettuato per un unico valore di V_{dd} . In altre parole, l'indice i , che simula la variazione della tensione di alimentazione, viene aggiornato solamente nel momento in cui il valore r raggiunge la convergenza. Alla fine di tutto l'algoritmo viene eseguito il *plot* del riferimento virtuale r e della tensione V_{REF} , ossia della tensione che si ottiene convertendo il riferimento virtuale con il DAC, la quale, in base alla teoria presentata nelle sezioni precedenti, risulta indipendente dalle variazioni della tensione di alimentazione V_{dd} .

Capitolo 5

Risultati e stima delle performance

In questo capitolo vengono presentati i risultati delle simulazioni effettuate in ambiente MATLAB. Le simulazioni riguardano lo studio del comportamento del riferimento di tensione virtuale al variare della tensione di alimentazione, della temperatura e dei parametri di processo (PVT). Viene inoltre effettuato uno studio della struttura al variare delle risoluzioni del DAC e del periodo dell'oscillatore locale al fine di porre le basi per una futura implementazione hardware della struttura.

5.1 Variazioni della tensione di alimentazione

I seguenti risultati riportano il comportamento sia del riferimento virtuale r che della tensione di riferimento V_{REF} , ricavato come il prodotto tra la singola r e la rispettiva V_{dd} . Vengono presentate di seguito due figure, delle quali la prima sfrutta la definizione del numero digitale A , tale per cui:

$$A = \log\left(\frac{n^{(1)}}{n^{(2)}}\right) \frac{2^N}{(m^{(1)} - m^{(2)})} \quad (5.1)$$

Si tratta dell'espressione ricavata con il modello (4.7), utilizzato nella trattazione precedente per esprimere il rapporto tra la V_{dd} e la V_0 . Si tratta di un modello approssimato, sul quale sono state considerate ulteriori approssimazioni. Di fatto, ad esempio, il termine $(1 - e(\frac{-V_{ds}}{V_T}))$ dell'espressione (4.7) è stato considerato trascurabile. Inoltre, non sono noti a priori i modelli utilizzati dall'ambiente Cadence per il calcolo della corrente sotto soglia, per cui, nell'espressione ricavata di A potrebbero mancare elementi legati alle non idealità. La figura 5.1 mostra il comportamento del riferimento virtuale alla temperatura nominale di 27°C in un range di tensione di alimentazione che varia tra 0.5V e 1V. Si nota come al crescere della V_{dd} , il riferimento r diminuisca. Sebbene si tratti, a prima vista, dell'andamento sperato, in quanto idealmente il prodotto $r \cdot V_{dd}$ deve risultare costante, la tensione V_{REF} calcolata presenta un comportamento pessimo. Essa viene riportata nella figura 5.2. Si nota, complessivamente, una line sensitivity (LS), nei confronti della tensione di alimentazione, pari a circa il 13%, non accettabile in un comune riferimento di

tensione. Inoltre, per tensioni di alimentazione superiori a 0.7V si nota un comportamento abbastanza lineare nell'andamento delle V_{REF} . In più, è possibile notare come sia stato imposto $V_{REF} = V_0 \approx \eta V_T \approx 40\text{mV}$ a 27°C , considerando il parametro $\eta \approx 1.5$.

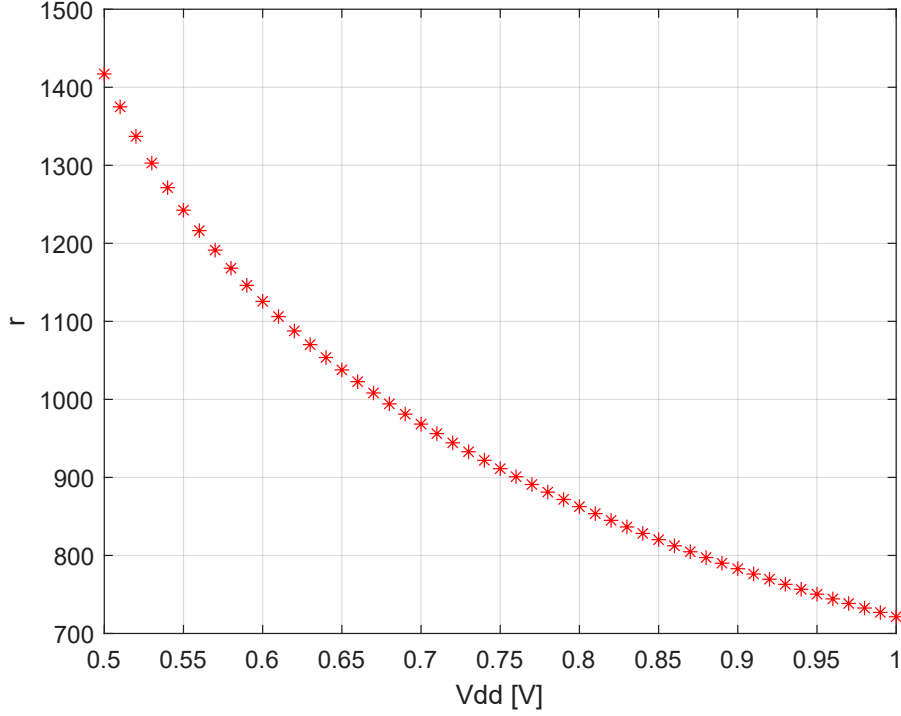


Figura 5.1. Riferimento virtuale r

Per tenere in considerazione quanto illustrato in precedenza, è stata effettuata una compensazione empirica di tipo software del numero digitale A . Invece di considerare nelle prossime simulazioni l'espressione (5.1), si è scelto di modificarla come di seguito :

$$A_2 = A - \alpha \cdot m^{(2)} = \log\left(\frac{n^{(1)}}{n^{(2)}}\right) \frac{2^N}{(m^{(1)} - m^{(2)})} - \alpha \cdot m^{(2)} \quad (5.2)$$

$$A_{new} = \frac{A_2}{1 - \beta \cdot A_2^2} \quad (5.3)$$

in cui $\alpha = 10^{-4}$ e $\beta = 2 \cdot 10^{-4}$. Ancora una volta il numero A dipende esclusivamente dai numeri digitali $m^{(1)}$, $m^{(2)}$, $n^{(1)}$, $n^{(2)}$, che vengono elaborati dal processore. Di seguito vengono riproposti i due plot del riferimento virtuale e della V_{REF} , nel caso della compensazione empirica del numero A , ottenibile dal codice MATLAB presente in appendice B.

La figura 5.4 mostra che la tensione di riferimento resta pressoché costante, presentando una riduzione significativa della line sensitivity fino allo 0.65% sullo stesso range di V_{dd} .

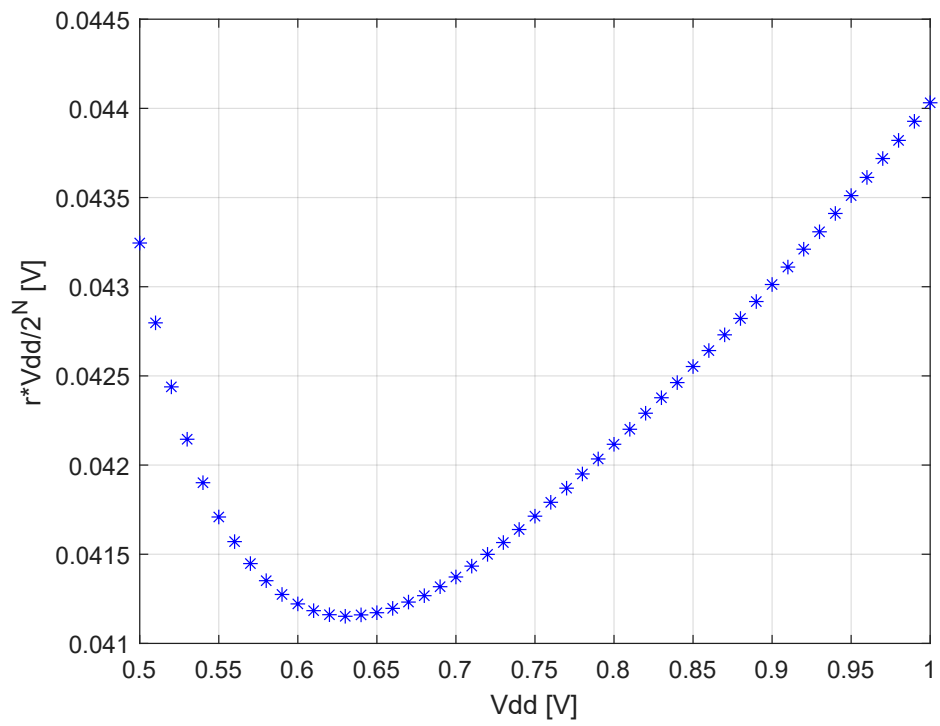


Figura 5.2. Tensione di riferimento V_{REF}

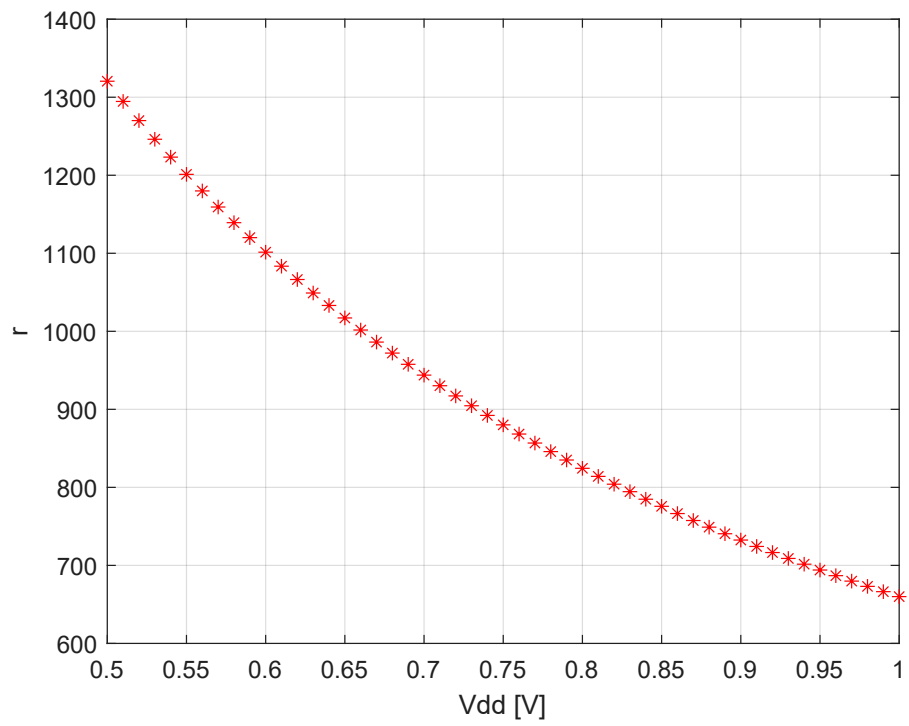


Figura 5.3. Riferimento virtuale r ottenuto con la compensazione empirica del numero A

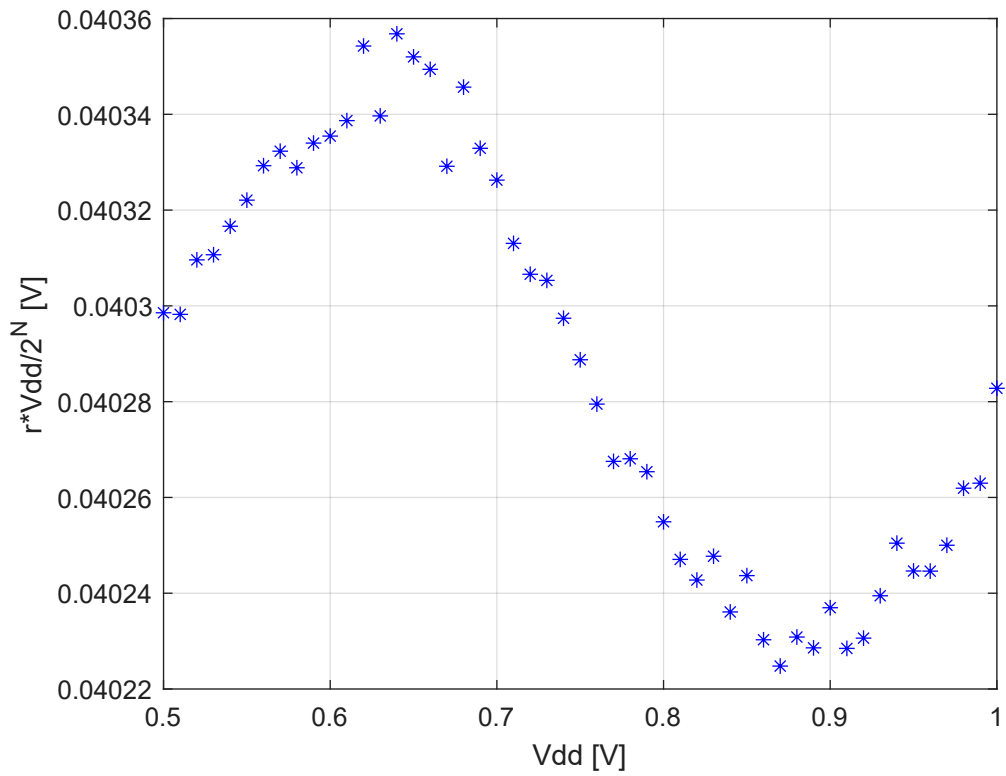


Figura 5.4. Tensione di riferimento V_{REF} ottenuto con la compensazione empirica del numero A

5.2 Variazione della temperatura

Con lo scopo di studiare il comportamento del riferimento virtuale al variare della temperatura, sono state rieseguite una serie di simulazioni in ambiente Cadence in cui si considera la variazione della tensione di alimentazione tra 0.5V e 1V con dei valori di temperatura che variano tra -40°C e 120°C , con un passo di 20°C . In seguito i dati sono stati importati in MATLAB ed è stato rieseguito l'algoritmo di compensazione per il calcolo del riferimento virtuale. Si è riutilizzato lo stesso algoritmo presente in appendice B, per cui non vengono mostrati i nuovi codici.

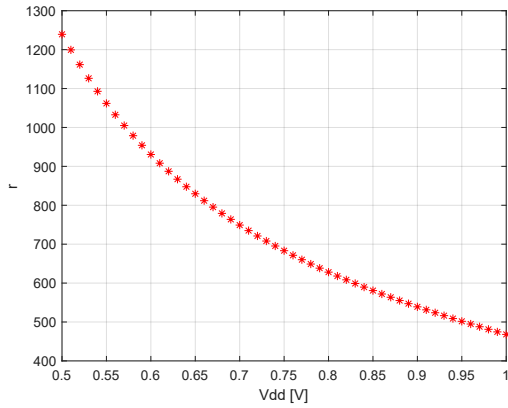


Figura 5.5. Riferimento virtuale a -40°C

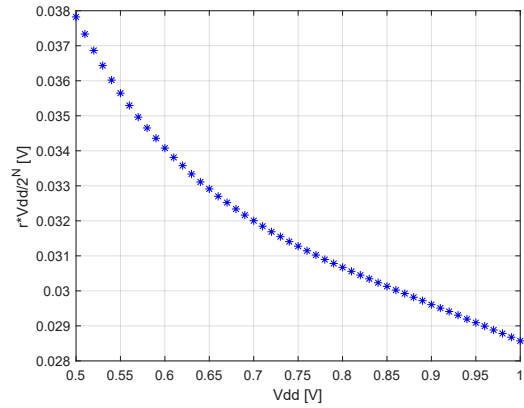


Figura 5.6. Tensione di riferimento a -40°C

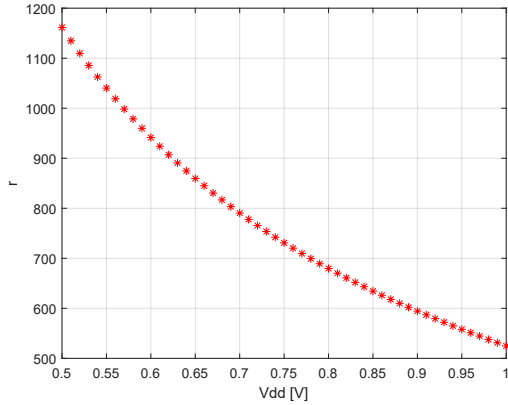


Figura 5.7. Riferimento virtuale a -20°C

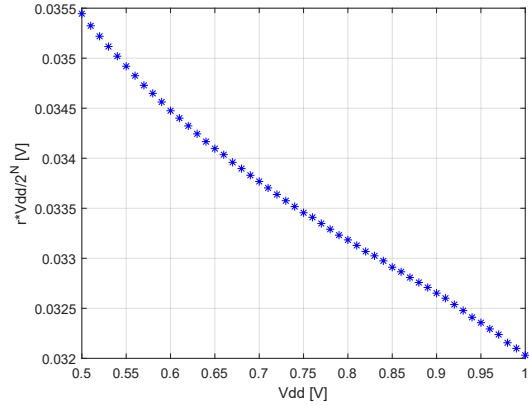


Figura 5.8. Tensione di riferimento a -20°C

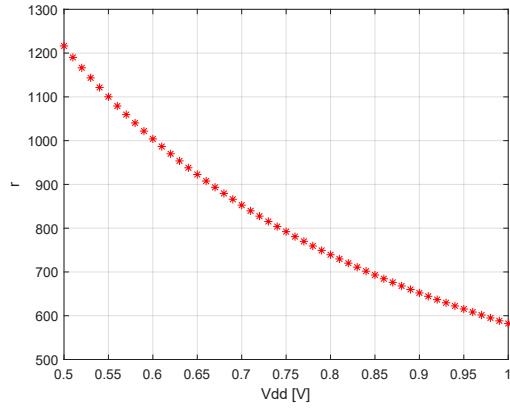


Figura 5.9. Riferimento virtuale a 0°C

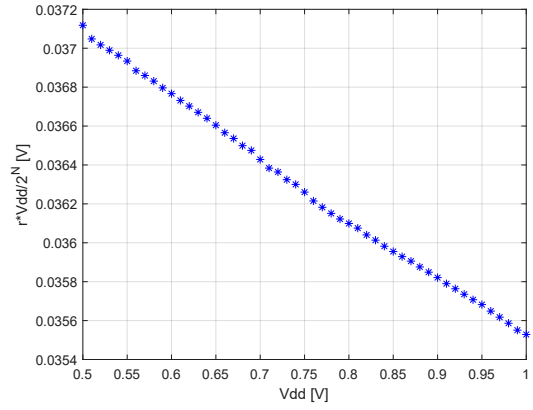


Figura 5.10. Tensione di riferimento a 0°C

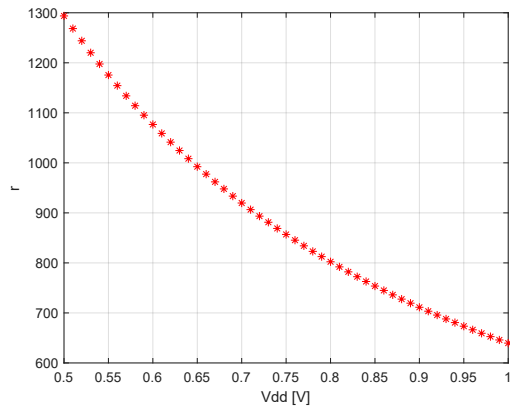


Figura 5.11. Riferimento virtuale a 20°C

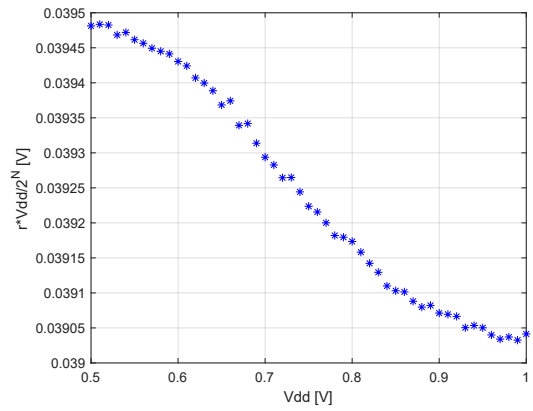


Figura 5.12. Tensione di riferimento a 20°C

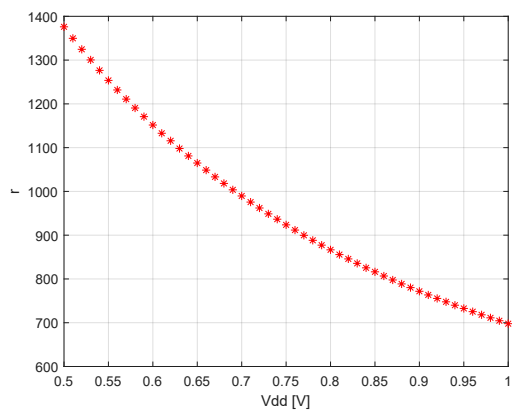


Figura 5.13. Riferimento virtuale a 40°C

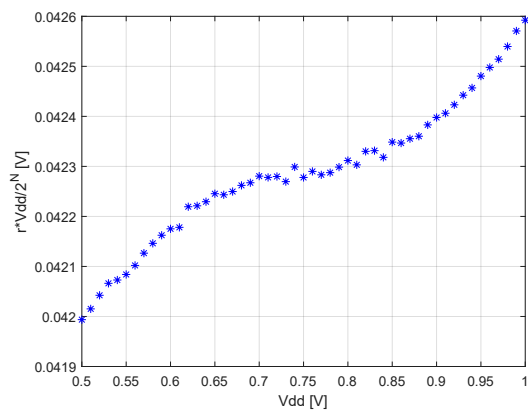


Figura 5.14. Tensione di riferimento a 40°C

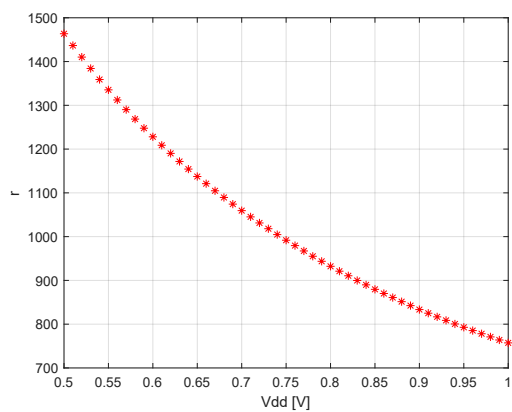


Figura 5.15. Riferimento virtuale a 60°C

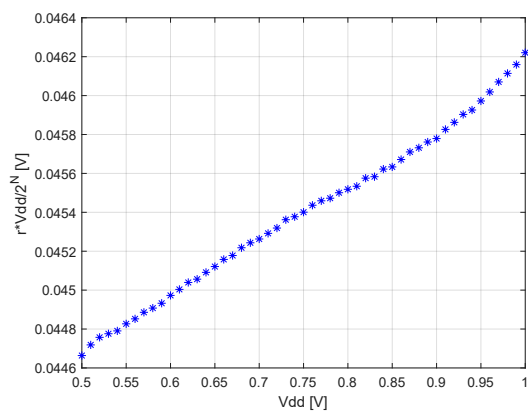


Figura 5.16. Tensione di riferimento a 60°C

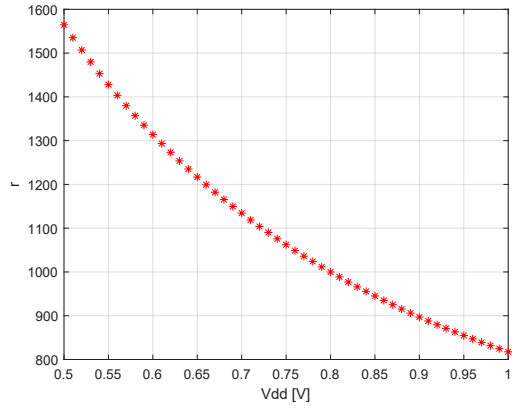


Figura 5.17. Riferimento virtuale a 80°C

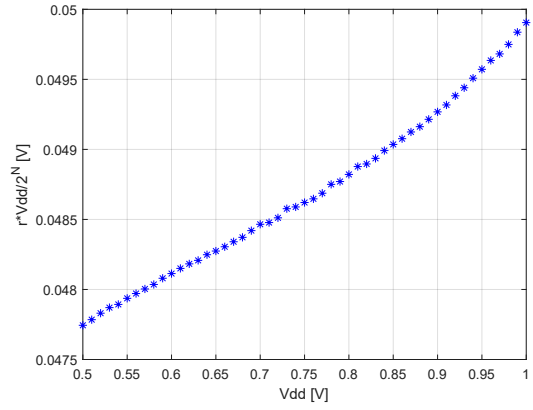


Figura 5.18. Tensione di riferimento a 80°C

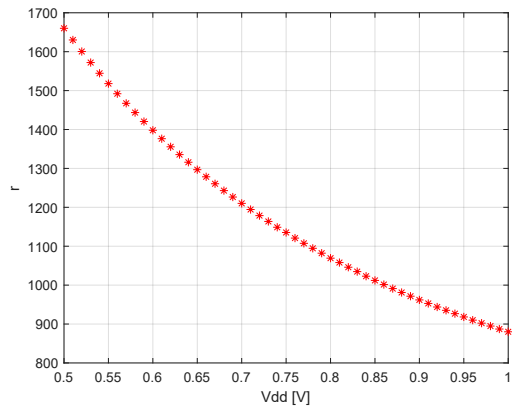


Figura 5.19. Riferimento virtuale a 100°C

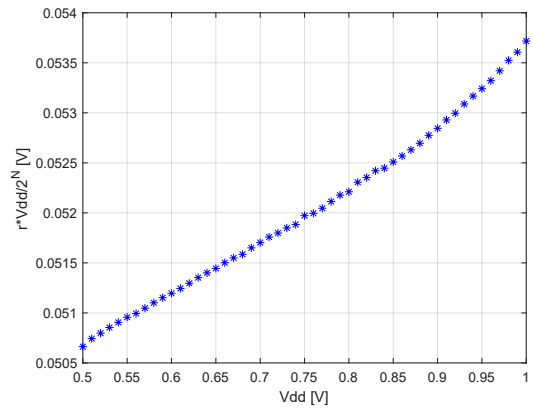


Figura 5.20. Tensione di riferimento a 100°C

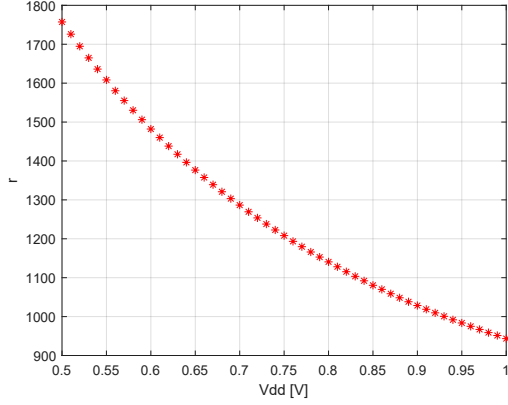


Figura 5.21. Riferimento virtuale a 120°C

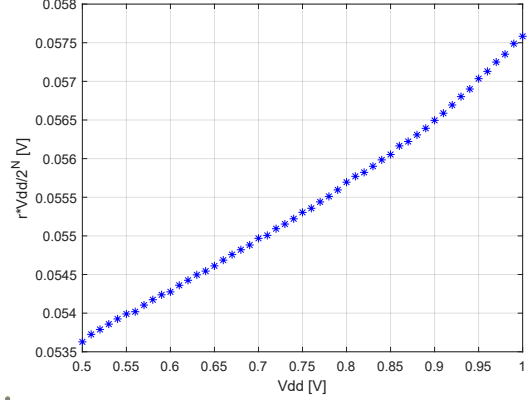


Figura 5.22. Tensione di riferimento a 120°C

La figura 5.24 riporta l'andamento della line sensitivity al variare della temperatura. In particolare si nota un pessimo comportamento a temperature estreme e il range in cui si assiste a performance accettabili coincide con l'intervallo 20-40 °C.

La figura 5.23 mostra nel dettaglio il trend della V_{REF} , al variare della tensione di alimentazione, a diverse temperature, mettendo in luce le diverse line sensitivity, riportate nella figura 5.24.

Risulta interessante studiare il comportamento della tensione V_{REF} al variare della temperatura per un valore fissato della tensione di alimentazione V_{dd} . A tal proposito, viene considerata una temperatura che varia tra -20° e 120° e delle tensioni di alimentazione che variano tra 0.5V a 1V con un passo di 0.1V. La figura 5.25 mostra che il coefficiente termico TC cresce al crescere della tensione di alimentazione. Alla tensione di 0.5V corrisponde un $TC = 0.33\%/^{\circ}C$, mentre alla tensione di 1V si ottiene $TC = 0.47\%/^{\circ}C$.

Dunque il riferimento ottenuto si comporta come un riferimento di tensione PTAT, in quanto la tensione di riferimento aumenta al crescere della temperatura. Si tratta di un trend prevedibile in quanto, nella trattazione, si impone che $V_{REF} = V_0 \approx \eta V_T = \eta kT/q$.

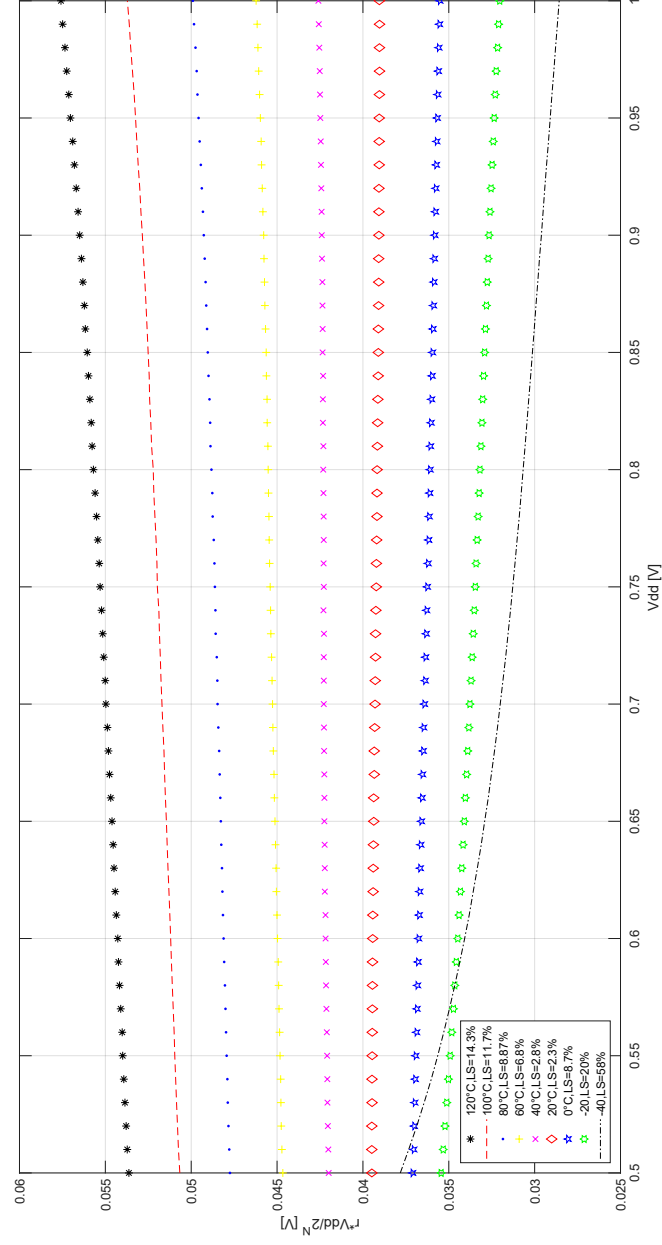


Figura 5.23. Tensione di riferimento V_{REF} ottenuto a diverse temperature al variare della tensione V_{dd}

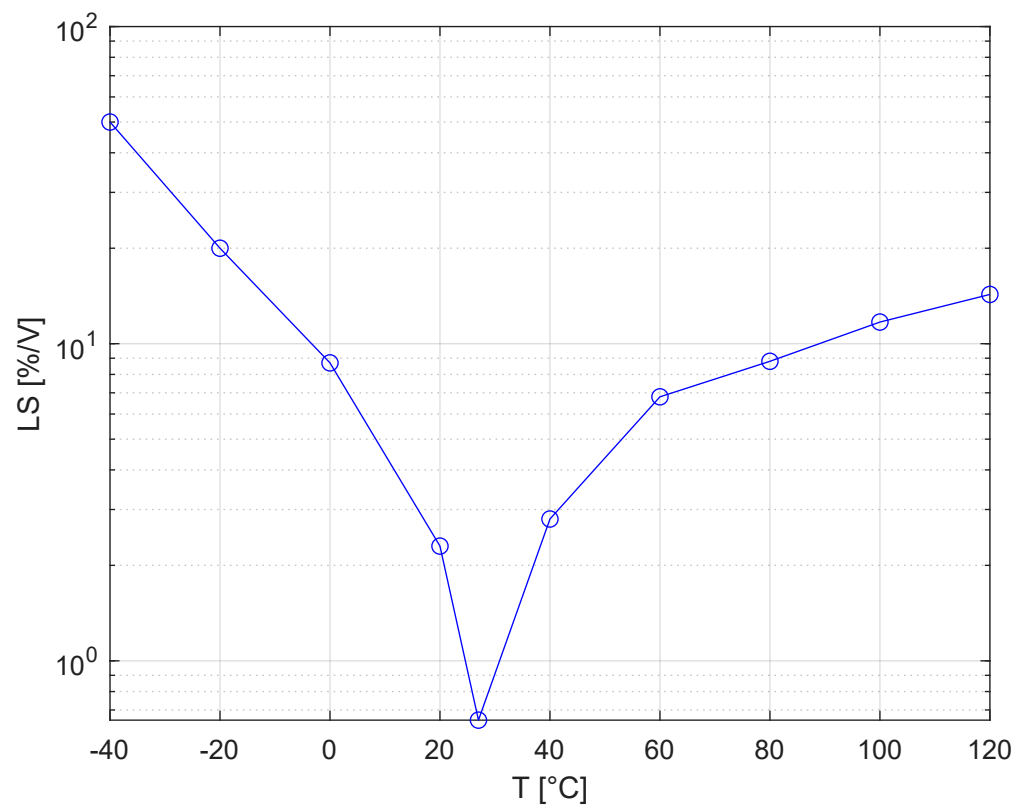


Figura 5.24. Line sensitivity (LS) al variare della temperatura (T)

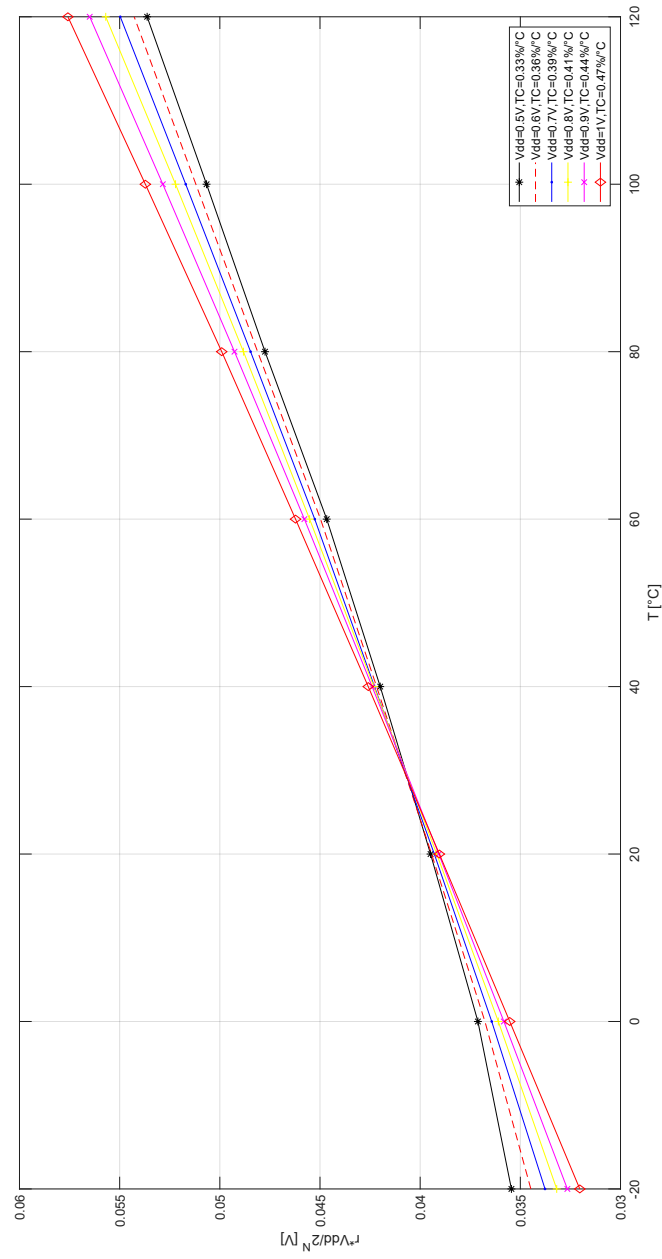


Figura 5.25. Tensione di riferimento V_{REF} ottenuto a diverse tensioni di alimentazione al variare della temperatura

5.3 Variazioni di processo

Per stimare la robustezza del riferimento di tensione virtuale rispetto alle variazioni di processo sono state eseguite delle simulazioni Monte Carlo, che tengono in considerazione anche le variazioni dovute al mismatch tra i diversi componenti. Per mismatch si intende la differenza nel valore di un parametro del singolo dispositivo rispetto ai valori dello stesso parametro di altri dispositivi identici. L'impatto delle variazioni casuali dei parametri dei dispositivi sul comportamento del circuito vengono elaborate dalla simulazione Monte Carlo, analizzando molte istanze dei dispositivi dotati di parametri con valori diversi [31]. Sono stati considerati 100 campioni (samples) e sono state calcolate le diverse frequenze di oscillazione del CSVCO, per diversi numeri digitali m in ingresso al DAC. In seguito a diverse simulazioni sono stati individuati i 2 campioni che presentano il comportamento peggiore in frequenza, ossia sono stati selezionati quei 2 campioni che presentano le massime differenze di frequenza rispetto alla frequenza media. Nella figura 5.26 viene riportato un esempio di simulazione nelle condizioni in cui $V_{dd}=0.75V$ e $m=5000$.

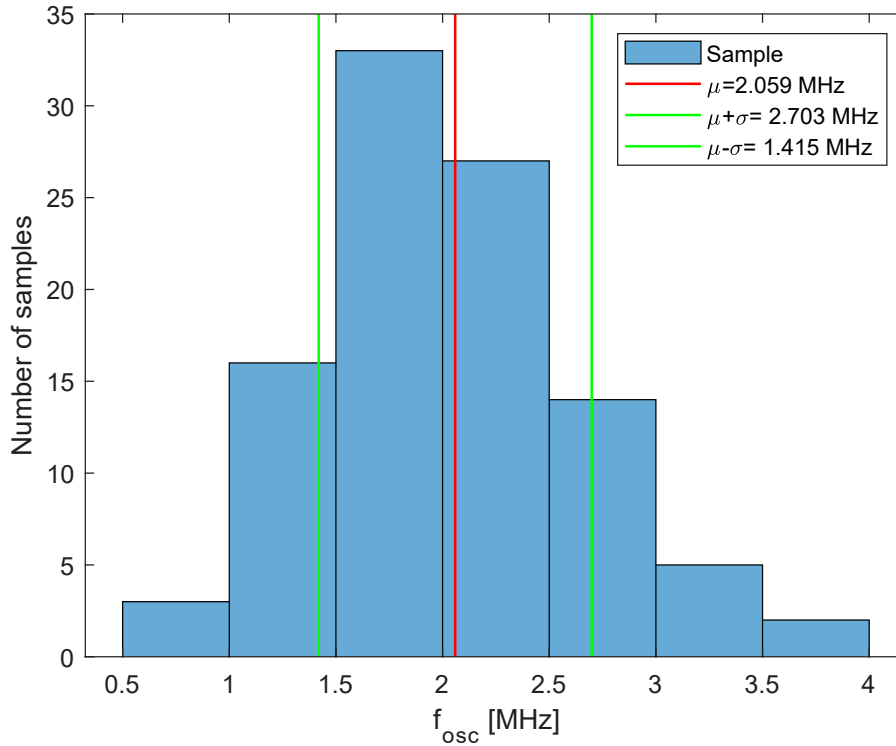


Figura 5.26. Simulazione Monte Carlo a 100 campioni di f_{osc} per mismatch e processo

Dopo aver selezionato i due campioni peggiori, sono state rieseguite le simulazioni nei confronti delle variazioni della sola tensione di alimentazione. In seguito i dati sono stati importati in MATLAB ed è stato rieseguito l'algoritmo di compensazione per il calcolo del

riferimento virtuale. Vengono dunque considerati gli stessi numeri digitali m da 3200 a 8192 e le tensioni di alimentazione tra 0.5V e 1V a 27°C.

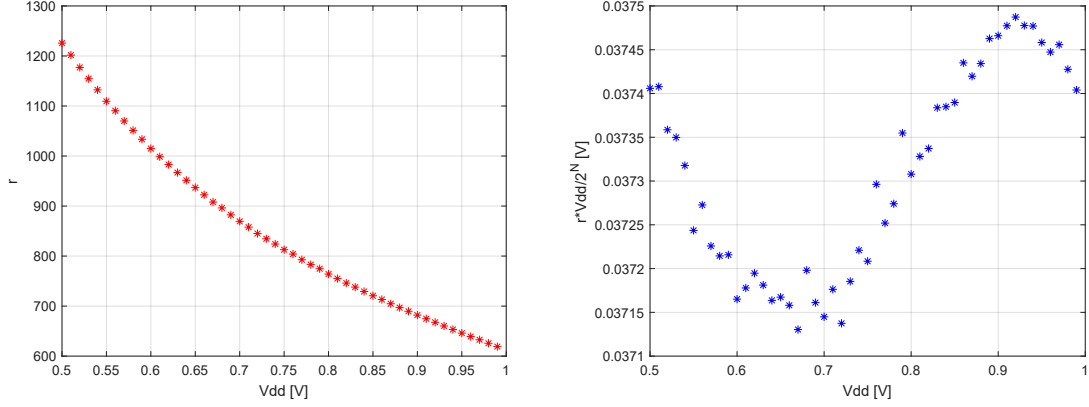


Figura 5.27. Riferimento virtuale ottenuto da Figura 5.28. Tensione di riferimento ottenuto da uno dei sample peggiori

Il primo dei due campioni peggiori, il cui comportamento viene mostrato nella figura 5.28 presenta una line sensitivity pari all' 1.9%/V sul range di V_{dd} tra 0.5V e 1V.

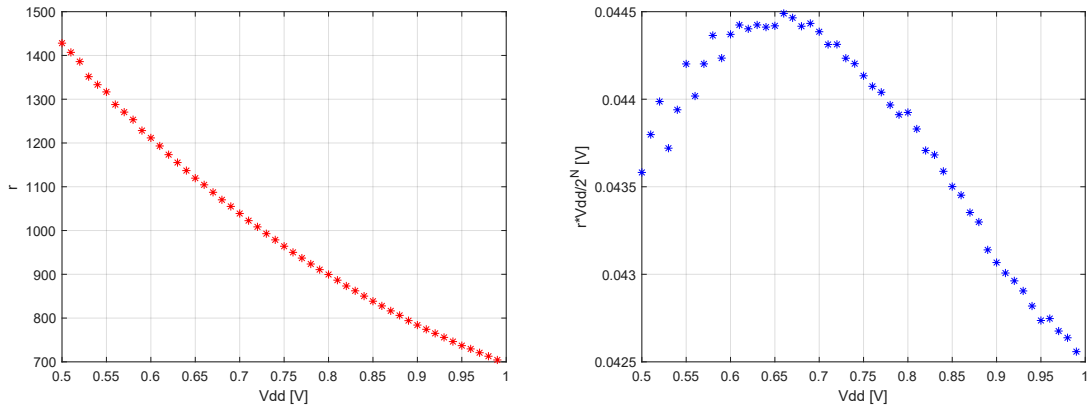


Figura 5.29. Riferimento virtuale ottenuto dal Figura 5.30. Tensione di riferimento ottenuto dal secondo dei sample peggiori

Nel caso del secondo sample 5.30 invece il comportamento risulta peggiore, presentando una LS= 8.8%/V. In generale, essendo stati considerati i due sample con variazione di frequenza maggiore, si afferma che il riferimento in questione presenta un comportamento accettabile nei confronti delle variazioni dei parametri di processo.

5.4 Studio della risoluzione del DAC e del periodo dell'oscillatore locale

Come anticipato in precedenza, il DAC, inteso come componente ideale, opera con un numero di bit pari a 14, mentre il contatore inizialmente opera con una risoluzione massima, considerando $T=1s$. Uno degli obiettivi del lavoro di tesi consiste nella determinazione delle condizioni architetturali minime che permettono di ottenere risultati soddisfacenti, ossia vengono indagate la risoluzione minima del DAC e il periodo T minimo dell'oscillatore locale, che impone la risoluzione del contatore, che permettono di ottenere buone performance da parte del riferimento. A questo scopo sono stati rieseguiti i codici in ambiente MATLAB, riguardanti le variazioni della tensione di alimentazione, facendo variare il numero di bit del DAC e il periodo dell'oscillatore locale T .

5.4.1 DAC

La soluzione adottata per simulare il comportamento del riferimento virtuale al variare dei bit N del DAC consiste nel selezionare, all'interno del codice, ogni numero digitale m in ingresso al DAC e applicare la seguente operazione:

$$m_{(14-E)bit} = \text{floor}\left(\frac{m_{14bit}}{2^E}\right) \cdot 2^E \quad (5.4)$$

in cui E rappresenta il numero di bit che si vuole sottrarre ai 14 bit iniziali. Ad esempio, volendo considerare 13 bit, il numero E vale 1, ossia l'LSB del numero su 14 bit viene posto pari a 0. L'operatore *floor* viene impiegato per approssimare il numero reale al numero intero inferiore. Di seguito vengono mostrati i risultati delle simulazioni effettuate al variare unicamente del numero di bit del DAC.

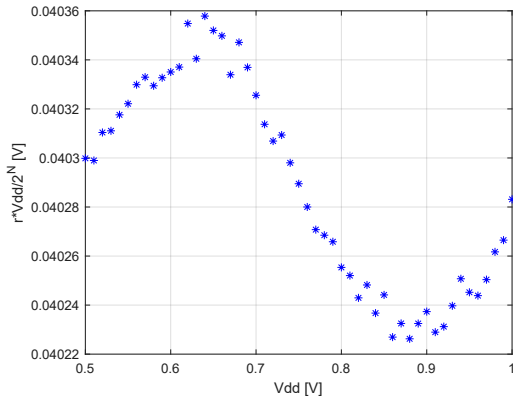


Figura 5.31. Tensione di riferimento ottenuta con DAC a 13 bit

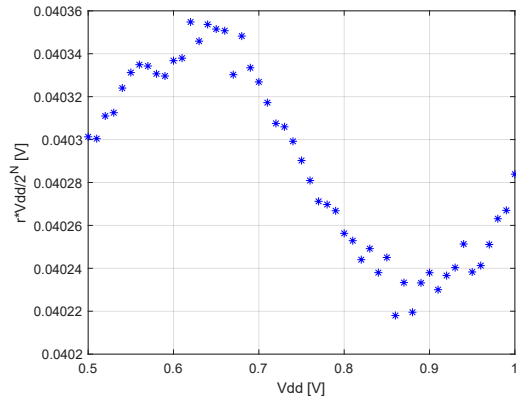


Figura 5.32. Tensione di riferimento ottenuta con DAC a 12 bit

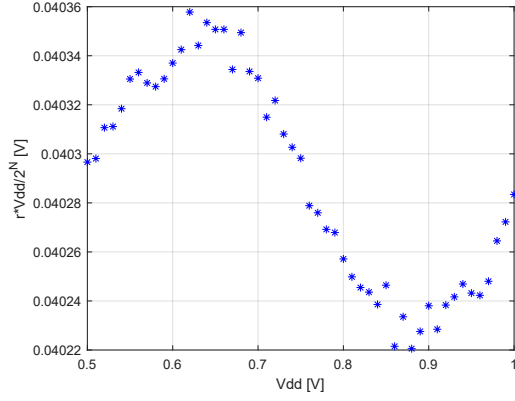


Figura 5.33. Tensione di riferimento ottenuta con DAC a 11 bit

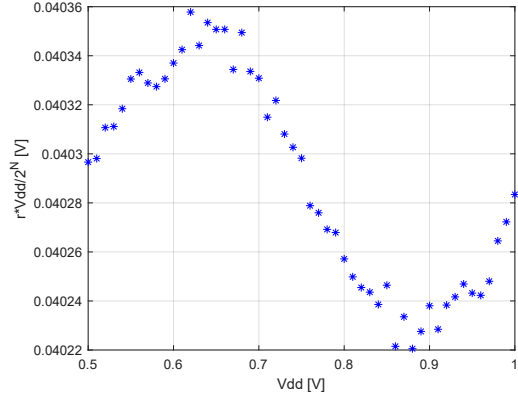


Figura 5.34. Tensione di riferimento ottenuta con DAC a 10 bit

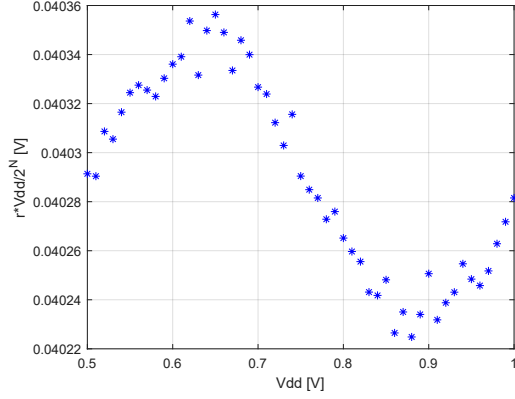


Figura 5.35. Tensione di riferimento ottenuta con DAC a 9 bit

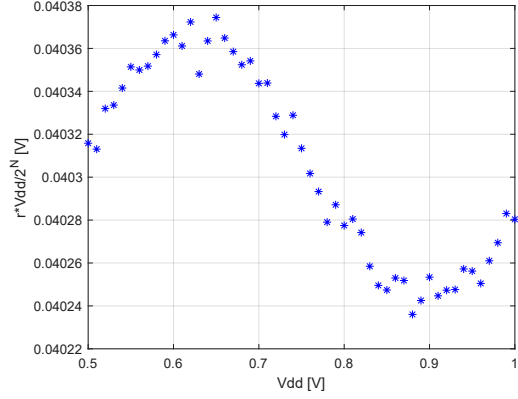


Figura 5.36. Tensione di riferimento ottenuta con DAC a 8 bit

La figura 5.38 riporta le diverse line sensitivity al variare del numero di bit del DAC. Si nota che le performance risultano soddisfacenti anche se si utilizzasse un DAC con soli 7 bit. Nel caso si utilizzassero risoluzioni inferiori, come 6 o 5 bit, non riportati nella figura, il riferimento smetterebbe di funzionare correttamente.

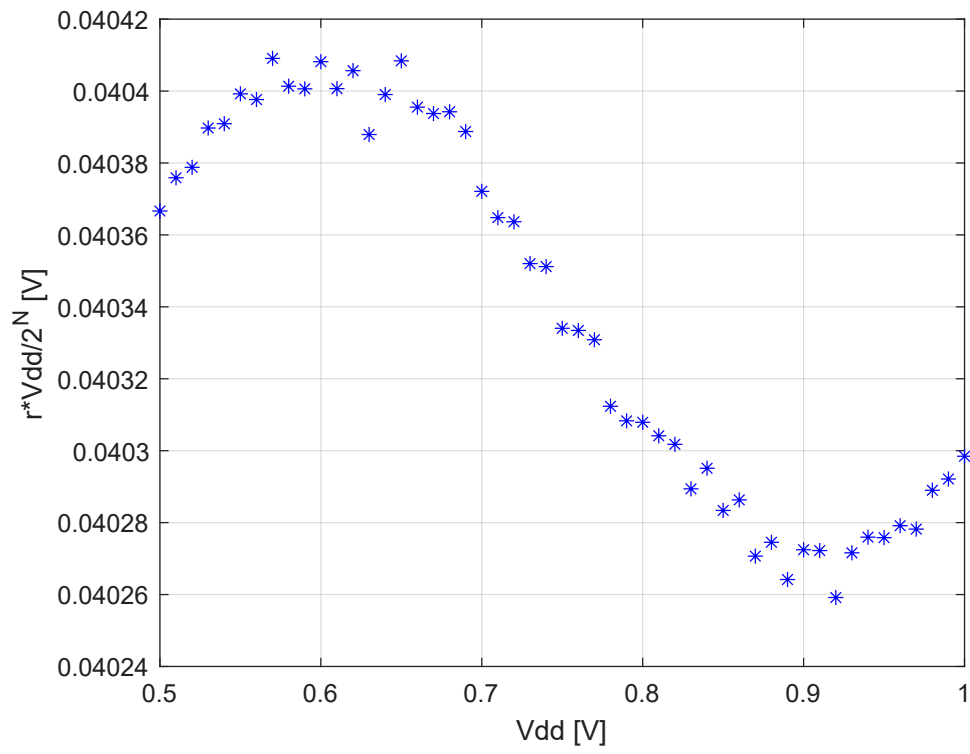


Figura 5.37. Tensione di riferimento ottenuta con DAC a 7 bit

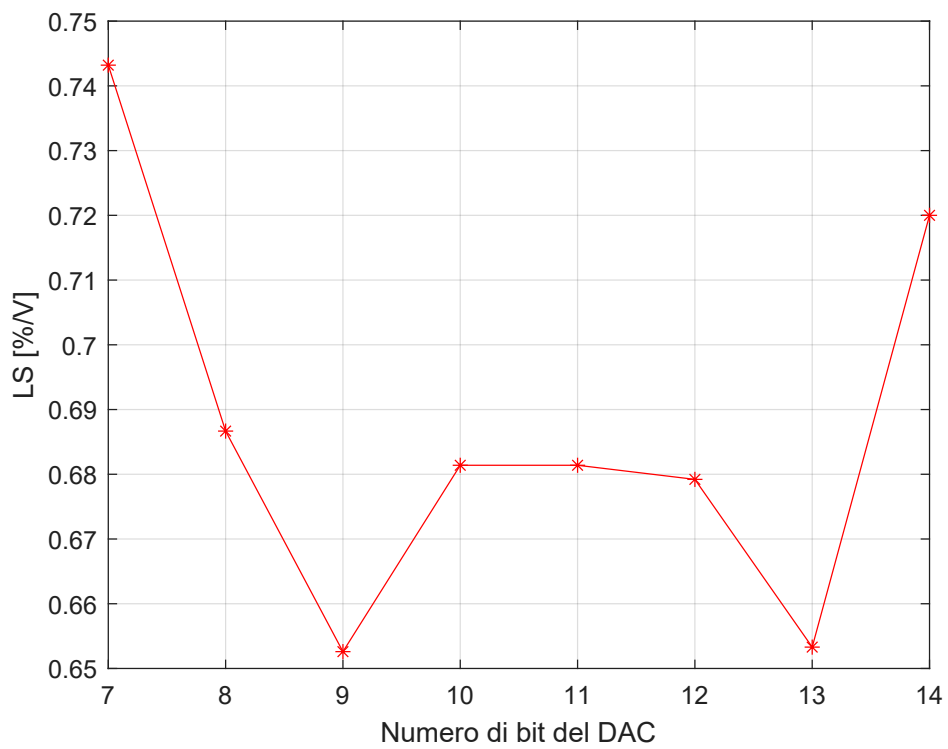


Figura 5.38. Line sensitivity al variare del numero di bit del DAC a 27°C

5.4.2 Oscillatore locale

In maniera analoga a quanto fatto per il DAC, si effettua uno studio del comportamento del riferimento al variare del periodo T dell'onda generata dall'oscillatore locale, mostrato in figura 5.39, che impone il numero di bit del contatore $N_{counter}$. Nelle simulazioni finora riportate sono state considerate delle frequenze con la risoluzione massima fornita dal simulatore in ambiente Cadence. Dunque la generica frequenza appartenente alla tabella 4.13 viene rappresentato da un numero floating point con risoluzione massima.

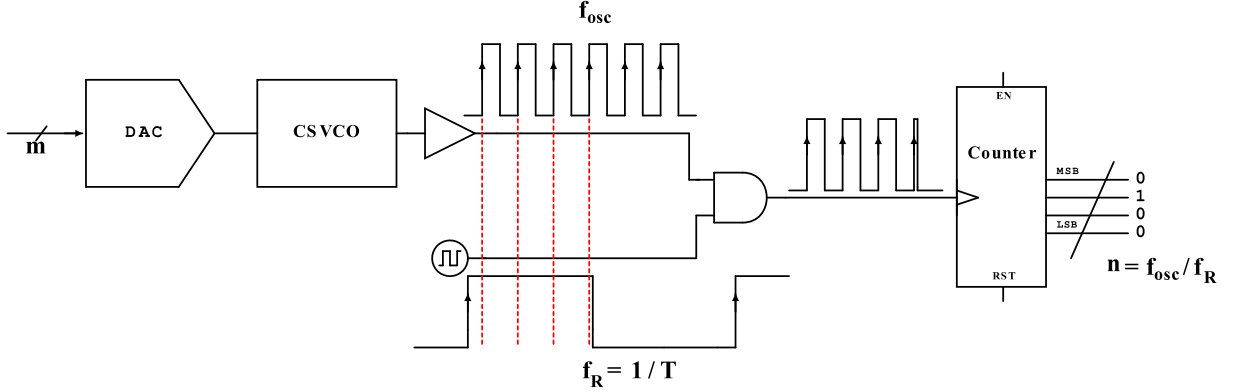


Figura 5.39. Struttura dell'ADC VCO-based che ingloba il CSVCO, l'oscillatore locale e il contatore

Inizialmente si individua il range di valori di frequenza che il codice MATLAB seleziona nella matrice delle frequenze per ricavare il riferimento virtuale nelle condizioni in cui V_{dd} varia tra 0.5V e 1V, alla temperatura di 27°C. In seguito viene effettuata l'analisi delle prestazioni del riferimento virtuale, in termini di line sensitivity, al variare del periodo T di osservazione, ossia si studia come il comportamento del sistema varia al variare del numero di fronti di salita contenuti in un intervallo T via via minore. Dopo questa analisi viene individuato il periodo T minimo che consente di ottenere performance accettabili e, di conseguenza, si calcola il numero di bit del contatore $N_{counter} = \log_2(T_{min} \cdot f_{max})$ al fine di poter rappresentare il rapporto di frequenze su un numero di bit opportuno. In questo caso, nel codice MATLAB, la singola frequenza f contenuta nella matrice 4.13, viene trasformata come di seguito:

$$f = \frac{\text{round}(f_{matrice} \cdot T)}{T} \quad (5.5)$$

Dall'analisi del codice e del suo funzionamento si ricavano la massima e la minima frequenza utilizzata dall'algoritmo per ricavare il riferimento virtuale r :

$$f_{max} = 2.15MHz \quad (5.6)$$

$$f_{min} = 81.69kHz \quad (5.7)$$

Sono state eseguite le simulazioni al variare della tensione di alimentazione, a 27°C e con un numero di bit del DAC pari a 14, al variare del periodo T . I risultati delle simulazioni sono illustrati in figura 5.40. In questo caso, per ottenere una $LS < 2\%$ e attuando una scelta conservativa, si sceglie un periodo $T = 4 \text{ ms}$ e, di conseguenza, si calcola $N_{counter}$ come di seguito:

$$N_{counter} = \log_2(T_{min} \cdot f_{max}) = \log_2(4ms \cdot 2.15MHz) = 14 \quad (5.8)$$

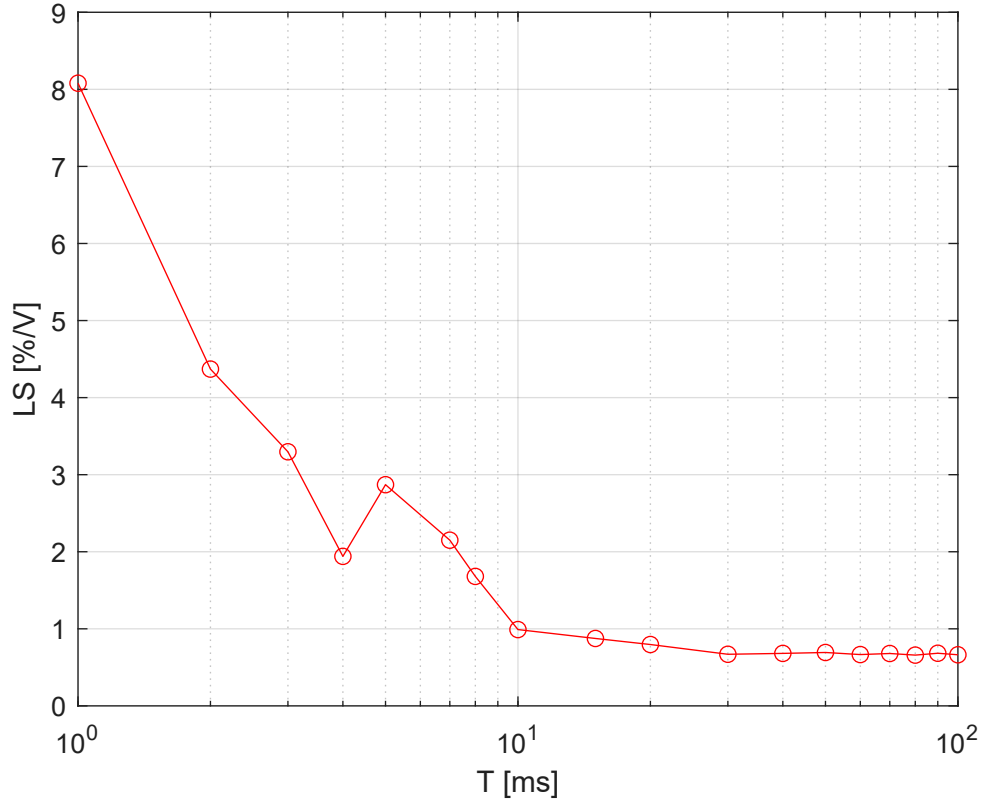


Figura 5.40. Line sensitivity al variare del periodo T dell'oscillatore locale, a 27°C

5.4.3 Combinazione del DAC e dell'oscillatore locale

Nel seguito vengono mostrati i risultati ottenuti eseguendo il codice MATLAB, opportunamente modificato, che include i risultati dello studio sia della risoluzione del DAC che del periodo dell'oscillatore locale. L'obiettivo principale consiste nel fornire una stima delle prestazioni nel caso in cui si considerano i parametri N_{DAC} e T minimi. Considerando, infatti, $N_{DAC}=7$ bit e il periodo di osservazione $T=4\text{ms}$, si ottiene una line sensitivity pari al $2.64\%/V$, con riferimento alla figura 5.41. Scegliendo $N_{DAC}=8$ bit e considerando $T=4\text{ms}$ si riottiene di nuovo una $LS=2.64\%/V$, dunque il periodo di osservazione rappresenta un fattore decisivo che impone le prestazioni della struttura. Per dimostrare quanto appena affermato, si rieseguono le stesse simulazioni con $T=7\text{ms}$. Si ottiene che, in entrambi i casi in cui $N_{DAC}=7$ bit e $N_{DAC}=8$ bit, la $LS=1.55\%/V$, come mostrato nelle figure 5.42 e 5.43.

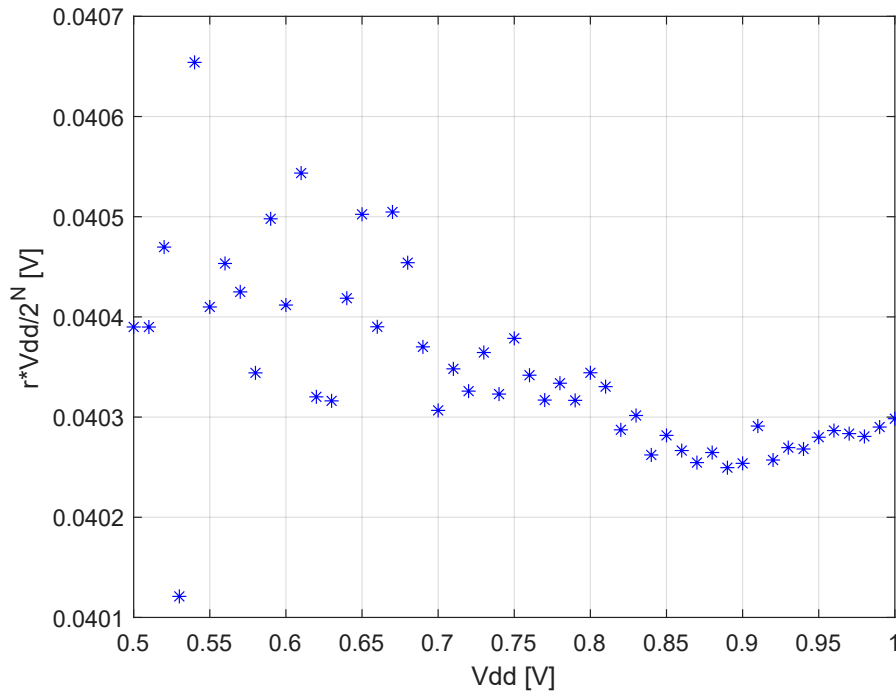


Figura 5.41. Tensione di riferimento con $T=4\text{ms}$ e $N_{DAC}=7$ a 27°C

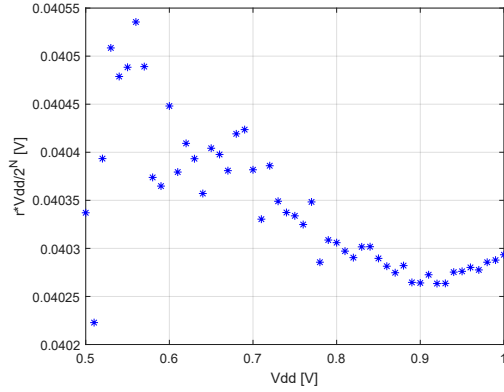


Figura 5.42. Tensione di riferimento $T=7\text{ms}$ e $N_{DAC}=7$ a 27°C

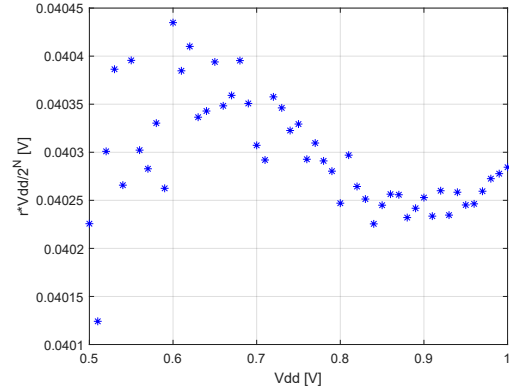


Figura 5.43. Tensione di riferimento $T=7\text{ms}$ e $N_{DAC}=8$ a 27°C

5.5 Consumo di potenza

Per valutare il consumo medio di potenza del CSVCO, che rappresenta di fatto l'unico blocco progettato, è stato considerato il caso particolare in cui l'ingresso digitale m sia costante e pari al numero 5000. La potenza viene calcolata a partire dalle simulazioni della corrente RMS (Root Mean Square), prelevata dal generatore di alimentazione V_{dd} . Dunque, la potenza media viene calcolata come di seguito:

$$P = I_{rms} \cdot V_{dd} \quad (5.9)$$

Essendo l'ingresso del DAC, considerato come un generatore ideale di tensione, pari a 5000, al variare della tensione di alimentazione, di fatto, cambia la tensione analogica all'ingresso del CSVCO. La figura 5.44 mostra l'andamento, su scala logaritmica, della potenza media in funzione della tensione di alimentazione V_{dd} .

Si nota che le potenze in gioco variano tra poche decine di nW fino al singolo μW . Inoltre, l'andamento della potenza risulta essere esponenziale, dato che la corrente richiesta dal circuito assume un comportamento esponenziale, dovuto alla regione di funzionamento sotto soglia dei transistori.

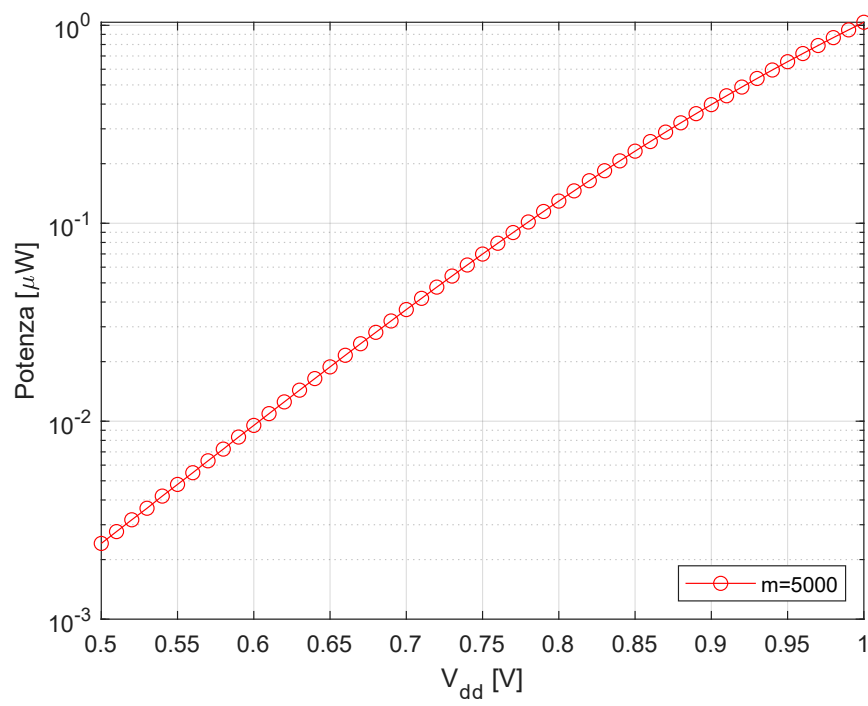


Figura 5.44. Potenza media dissipata dal CSVCO al variare della tensione di alimentazione

Capitolo 6

Conclusioni

Dopo aver presentato alcuni esempi di riferimenti di tensione integrati analogici, avendo descritto le figure di merito di un generale riferimento di tensione, a partire da un riferimento bandgap di Kuijk, è stato presentato il concetto di riferimento di tensione virtuale, basato su tecniche digitali, la cui struttura consiste in una valida alternativa ai riferimenti analogici tradizionali che non godono dei vantaggi introdotti dallo scaling geometrico, tipico della tecnologia CMOS.

A partire dal riferimento di tensione virtuale, è stato studiato ed analizzato un riferimento di tensione virtuale VCO-based, che rappresenta il principale argomento del lavoro di tesi. In particolare, è stata descritta la fase di progettazione di un Current Starved Voltage Controlled Oscillator (CSVCO), effettuata in ambiente Cadence, riportando i rapporti di aspetto dei singoli transistori e le relative simulazioni inerenti il comportamento della frequenza di uscita dell'oscillatore al variare della tensione di alimentazione e della temperatura.

L'intera struttura del riferimento di tensione virtuale è stata analizzata nel dettaglio. Esso, oltre a contenere un Digital-to-Analog Converter (DAC) ideale di N bit, è composto dal CSVCO, da un contatore con $N_{counter}$ bit e da un oscillatore locale, generante un segnale di periodo T , i quali compongono la struttura di un ADC VCO-based. Inoltre, i blocchi DAC e ADC VCO-based, si interfacciano con un processore, scambiando numeri digitali, che occorrono per il funzionamento dell'algoritmo su cui la struttura è basata. Il processing dei numeri digitali, infatti, permette ad una procedura algoritmica di compensazione di ricavare il riferimento virtuale r , grazie al quale si riesce ad ottenere un riferimento di tensione, all'uscita del DAC, scarsamente dipendente dalle variazioni PVT. L'intera procedura di compensazione è stata riprodotta e testata all'interno del software MATLAB, disponendo di una matrice delle frequenze del segnale in uscita dal CSVCO, sotto le diverse condizioni PVT, ricavate dalle simulazioni effettuate in Cadence. Il sistema simulato permette di ottenere una tensione di uscita che presenta una line sensitivity (LS) (%/V) dello 0.65%/V a fronte di una variazione della tensione di alimentazione V_{dd} tra 0.5V e 1V, a 27°C. Questa performance vede deteriorarsi al variare della temperatura, anche se nell'intervallo tra i 20°C e i 40°C, la line sensitivity si attesta intorno al 2%/V. Inoltre, in termini di coefficiente di temperatura (%/°C), nell'intervallo di tensione di alimentazione tra 0.5V e 1V, si ottiene una crescita dallo 0.33%/°C allo 0.47%/°C, presentando un comportamento PTAT (Proportional-To-Absolute-Temperature). Per quanto riguarda le variazioni dei parametri

di processo, dalla selezione dei due campioni (su 100) con comportamento peggiore in termini di line sensitivity, a 27°C, nel range tra 0.5V e 1V, si ottengono risultati dell'ordine del 1.9%/V e dell'8.8%/V. Un ulteriore studio effettuato sul periodo fornito dall'oscillatore locale, che impone il numero di bit del contatore, mostra che si ottiene una $LS = 1.55\%/V$, considerando un DAC a 7 bit e un oscillatore con $T = 7ms$.

I possibili sviluppi futuri della struttura prevedono l'integrazione di un DAC completamente digitale, dell'implementazione di un contatore digitale e dell'oscillatore locale, al fine di ottenere una struttura completamente digitale, tale da trarre beneficio dallo scaling geometrico in termini di costi, riduzione della potenza dissipata e dell'integrazione su silicio, in modo da essere impiegato all'interno di un nodo IoT.

Appendice A

Codice MATLAB

```
close all
clear all
format long

% T=27°C!

m= linspace(3200,8192 ,4993); % numeri di ingresso al DAC
vdd= linspace(0.5,1,51); % vdd varia tra 0.5V e 1V a passo di 0.01V
N=14; % numero di bit del DAC

% INIZIO IMPORTAZIONE DEI DATI

freq_n_vdd_050_1= importdata('m_da_3200_a_3299_vdd_050_a_1_passo_001.csv');
n_vdd_050_1 = freq_n_vdd_050_1.data(:,2);

freq_n_vdd_051_1= importdata('m_da_3200_a_3299_vdd_051_a_1_passo_001.csv');
n_vdd_051_1 = freq_n_vdd_051_1.data(:,2);

freq_n_vdd_052_1= importdata('m_da_3200_a_3299_vdd_052_a_1_passo_001.csv');
n_vdd_052_1 = freq_n_vdd_052_1.data(:,2);

freq_n_vdd_053_1= importdata('m_da_3200_a_3299_vdd_053_a_1_passo_001.csv');
n_vdd_053_1 = freq_n_vdd_053_1.data(:,2);

freq_n_vdd_054_1= importdata('m_da_3200_a_3299_vdd_054_a_1_passo_001.csv');
n_vdd_054_1 = freq_n_vdd_054_1.data(:,2);

freq_n_vdd_055_1= importdata('m_da_3200_a_3299_vdd_055_a_1_passo_001.csv');
n_vdd_055_1 = freq_n_vdd_055_1.data(:,2);

freq_n_vdd_056_1= importdata('m_da_3200_a_3299_vdd_056_a_1_passo_001.csv');
n_vdd_056_1 = freq_n_vdd_056_1.data(:,2);

freq_n_vdd_057_1= importdata('m_da_3200_a_3299_vdd_057_a_1_passo_001.csv');
n_vdd_057_1 = freq_n_vdd_057_1.data(:,2);

freq_n_vdd_058_1= importdata('m_da_3200_a_3299_vdd_058_a_1_passo_001.csv');
```

```

n_vdd_058_1 = freq_n_vdd_058_1.data(:,2);

freq_n_vdd_059_1= importdata('m_da_3200_a_3299_vdd_059_a_1_passo_001.csv');
n_vdd_059_1 = freq_n_vdd_059_1.data(:,2);

freq_n_vdd_060_1= importdata('m_da_3200_a_3299_vdd_060_a_1_passo_001.csv');
n_vdd_060_1 = freq_n_vdd_060_1.data(:,2);

freq_n_vdd_061_1= importdata('m_da_3200_a_3299_vdd_061_a_1_passo_001.csv');
n_vdd_061_1 = freq_n_vdd_061_1.data(:,2);

freq_n_vdd_062_1= importdata('m_da_3200_a_3299_vdd_062_a_1_passo_001.csv');
n_vdd_062_1 = freq_n_vdd_062_1.data(:,2);

freq_n_vdd_063_1= importdata('m_da_3200_a_3299_vdd_063_a_1_passo_001.csv');
n_vdd_063_1 = freq_n_vdd_063_1.data(:,2);

freq_n_vdd_064_1= importdata('m_da_3200_a_3299_vdd_064_a_1_passo_001.csv');
n_vdd_064_1 = freq_n_vdd_064_1.data(:,2);

freq_n_vdd_065_1= importdata('m_da_3200_a_3299_vdd_065_a_1_passo_001.csv');
n_vdd_065_1 = freq_n_vdd_065_1.data(:,2);

freq_n_vdd_066_1= importdata('m_da_3200_a_3299_vdd_066_a_1_passo_001.csv');
n_vdd_066_1 = freq_n_vdd_066_1.data(:,2);

freq_n_vdd_067_1= importdata('m_da_3200_a_3299_vdd_067_a_1_passo_001.csv');
n_vdd_067_1 = freq_n_vdd_067_1.data(:,2);

freq_n_vdd_068_1= importdata('m_da_3200_a_3299_vdd_068_a_1_passo_001.csv');
n_vdd_068_1 = freq_n_vdd_068_1.data(:,2);

freq_n_vdd_069_1= importdata('m_da_3200_a_3299_vdd_069_a_1_passo_001.csv');
n_vdd_069_1 = freq_n_vdd_069_1.data(:,2);

freq_n_vdd_070_1= importdata('m_da_3200_a_3299_vdd_070_a_1_passo_001.csv');
n_vdd_070_1 = freq_n_vdd_070_1.data(:,2);

freq_n_vdd_071_1= importdata('m_da_3200_a_3299_vdd_071_a_1_passo_001.csv');
n_vdd_071_1 = freq_n_vdd_071_1.data(:,2);

freq_n_vdd_072_1= importdata('m_da_3200_a_3299_vdd_072_a_1_passo_001.csv');
n_vdd_072_1 = freq_n_vdd_072_1.data(:,2);

freq_n_vdd_073_1= importdata('m_da_3200_a_3299_vdd_073_a_1_passo_001.csv');
n_vdd_073_1 = freq_n_vdd_073_1.data(:,2);

freq_n_vdd_074_1= importdata('m_da_3200_a_3299_vdd_074_a_1_passo_001.csv');
n_vdd_074_1 = freq_n_vdd_074_1.data(:,2);

freq_n_vdd_075_1= importdata('m_da_3200_a_3299_vdd_075_a_1_passo_001.csv');
n_vdd_075_1 = freq_n_vdd_075_1.data(:,2);

freq_n_vdd_076_1= importdata('m_da_3200_a_3299_vdd_076_a_1_passo_001.csv');
n_vdd_076_1 = freq_n_vdd_076_1.data(:,2);

```

```

freq_n_vdd_077_1= importdata('m_da_3200_a_3299_vdd_077_a_1_passo_001.csv');
n_vdd_077_1 = freq_n_vdd_077_1.data(:,2);

freq_n_vdd_078_1= importdata('m_da_3200_a_3299_vdd_078_a_1_passo_001.csv');
n_vdd_078_1 = freq_n_vdd_078_1.data(:,2);

freq_n_vdd_079_1= importdata('m_da_3200_a_3299_vdd_079_a_1_passo_001.csv');
n_vdd_079_1 = freq_n_vdd_079_1.data(:,2);

freq_n_vdd_080_1= importdata('m_da_3200_a_3299_vdd_080_a_1_passo_001.csv');
n_vdd_080_1 = freq_n_vdd_080_1.data(:,2);

freq_n_vdd_081_1= importdata('m_da_3200_a_3299_vdd_081_a_1_passo_001.csv');
n_vdd_081_1 = freq_n_vdd_081_1.data(:,2);

freq_n_vdd_082_1= importdata('m_da_3200_a_3299_vdd_082_a_1_passo_001.csv');
n_vdd_082_1 = freq_n_vdd_082_1.data(:,2);

freq_n_vdd_083_1= importdata('m_da_3200_a_3299_vdd_083_a_1_passo_001.csv');
n_vdd_083_1 = freq_n_vdd_083_1.data(:,2);

freq_n_vdd_084_1= importdata('m_da_3200_a_3299_vdd_084_a_1_passo_001.csv');
n_vdd_084_1 = freq_n_vdd_084_1.data(:,2);

freq_n_vdd_085_1= importdata('m_da_3200_a_3299_vdd_085_a_1_passo_001.csv');
n_vdd_085_1 = freq_n_vdd_085_1.data(:,2);

freq_n_vdd_086_1= importdata('m_da_3200_a_3299_vdd_086_a_1_passo_001.csv');
n_vdd_086_1 = freq_n_vdd_086_1.data(:,2);

freq_n_vdd_087_1= importdata('m_da_3200_a_3299_vdd_087_a_1_passo_001.csv');
n_vdd_087_1 = freq_n_vdd_087_1.data(:,2);

freq_n_vdd_088_1= importdata('m_da_3200_a_3299_vdd_088_a_1_passo_001.csv');
n_vdd_088_1 = freq_n_vdd_088_1.data(:,2);

freq_n_vdd_089_1= importdata('m_da_3200_a_3299_vdd_089_a_1_passo_001.csv');
n_vdd_089_1 = freq_n_vdd_089_1.data(:,2);

freq_n_vdd_090_1= importdata('m_da_3200_a_3299_vdd_090_a_1_passo_001.csv');
n_vdd_090_1 = freq_n_vdd_090_1.data(:,2);

freq_n_vdd_091_1= importdata('m_da_3200_a_3299_vdd_091_a_1_passo_001.csv');
n_vdd_091_1 = freq_n_vdd_091_1.data(:,2);

freq_n_vdd_092_1= importdata('m_da_3200_a_3299_vdd_092_a_1_passo_001.csv');
n_vdd_092_1 = freq_n_vdd_092_1.data(:,2);

freq_n_vdd_093_1= importdata('m_da_3200_a_3299_vdd_093_a_1_passo_001.csv');
n_vdd_093_1 = freq_n_vdd_093_1.data(:,2);

freq_n_vdd_094_1= importdata('m_da_3200_a_3299_vdd_094_a_1_passo_001.csv');
n_vdd_094_1 = freq_n_vdd_094_1.data(:,2);

```



```

freq_n_vdd_095_1= importdata('m_da_3200_a_3299_vdd_095_a_1_passo_001.csv');
n_vdd_095_1 = freq_n_vdd_095_1.data(:,2);

freq_n_vdd_096_1= importdata('m_da_3200_a_3299_vdd_096_a_1_passo_001.csv');
n_vdd_096_1 = freq_n_vdd_096_1.data(:,2);

freq_n_vdd_097_1= importdata('m_da_3200_a_3299_vdd_097_a_1_passo_001.csv');
n_vdd_097_1 = freq_n_vdd_097_1.data(:,2);

freq_n_vdd_098_1= importdata('m_da_3200_a_3299_vdd_098_a_1_passo_001.csv');
n_vdd_098_1 = freq_n_vdd_098_1.data(:,2);

freq_n_vdd_099_1= importdata('m_da_3200_a_3299_vdd_099_a_1_passo_001.csv');
n_vdd_099_1 = freq_n_vdd_099_1.data(:,2);

freq_n_vdd_01_1= importdata('m_da_3200_a_3299_vdd_01_a_1_passo_001.csv');
n_vdd_01_1 = freq_n_vdd_01_1.data(:,2);


freq_n_vdd_050_2= importdata('m_da_3300_a_3499_vdd_050_a_1_passo_001.csv');
n_vdd_050_2 = freq_n_vdd_050_2.data(:,2);

freq_n_vdd_051_2= importdata('m_da_3300_a_3499_vdd_051_a_1_passo_001.csv');
n_vdd_051_2 = freq_n_vdd_051_2.data(:,2);

freq_n_vdd_052_2= importdata('m_da_3300_a_3499_vdd_052_a_1_passo_001.csv');
n_vdd_052_2 = freq_n_vdd_052_2.data(:,2);

freq_n_vdd_053_2= importdata('m_da_3300_a_3499_vdd_053_a_1_passo_001.csv');
n_vdd_053_2 = freq_n_vdd_053_2.data(:,2);

freq_n_vdd_054_2= importdata('m_da_3300_a_3499_vdd_054_a_1_passo_001.csv');
n_vdd_054_2 = freq_n_vdd_054_2.data(:,2);

freq_n_vdd_055_2= importdata('m_da_3300_a_3499_vdd_055_a_1_passo_001.csv');
n_vdd_055_2 = freq_n_vdd_055_2.data(:,2);

freq_n_vdd_056_2= importdata('m_da_3300_a_3499_vdd_056_a_1_passo_001.csv');
n_vdd_056_2 = freq_n_vdd_056_2.data(:,2);

freq_n_vdd_057_2= importdata('m_da_3300_a_3499_vdd_057_a_1_passo_001.csv');
n_vdd_057_2 = freq_n_vdd_057_2.data(:,2);

freq_n_vdd_058_2= importdata('m_da_3300_a_3499_vdd_058_a_1_passo_001.csv');
n_vdd_058_2 = freq_n_vdd_058_2.data(:,2);

freq_n_vdd_059_2= importdata('m_da_3300_a_3499_vdd_059_a_1_passo_001.csv');
n_vdd_059_2 = freq_n_vdd_059_2.data(:,2);

```

```
freq_n_vdd_060_2= importdata('m_da_3300_a_3499_vdd_060_a_1_passo_001.csv');
n_vdd_060_2 = freq_n_vdd_060_2.data(:,2);

freq_n_vdd_061_2= importdata('m_da_3300_a_3499_vdd_061_a_1_passo_001.csv');
n_vdd_061_2 = freq_n_vdd_061_2.data(:,2);

freq_n_vdd_062_2= importdata('m_da_3300_a_3499_vdd_062_a_1_passo_001.csv');
n_vdd_062_2 = freq_n_vdd_062_2.data(:,2);

freq_n_vdd_063_2= importdata('m_da_3300_a_3499_vdd_063_a_1_passo_001.csv');
n_vdd_063_2 = freq_n_vdd_063_2.data(:,2);

freq_n_vdd_064_2= importdata('m_da_3300_a_3499_vdd_064_a_1_passo_001.csv');
n_vdd_064_2 = freq_n_vdd_064_2.data(:,2);

freq_n_vdd_065_2= importdata('m_da_3300_a_3499_vdd_065_a_1_passo_001.csv');
n_vdd_065_2 = freq_n_vdd_065_2.data(:,2);

freq_n_vdd_066_2= importdata('m_da_3300_a_3499_vdd_066_a_1_passo_001.csv');
n_vdd_066_2 = freq_n_vdd_066_2.data(:,2);

freq_n_vdd_067_2= importdata('m_da_3300_a_3499_vdd_067_a_1_passo_001.csv');
n_vdd_067_2 = freq_n_vdd_067_2.data(:,2);

freq_n_vdd_068_2= importdata('m_da_3300_a_3499_vdd_068_a_1_passo_001.csv');
n_vdd_068_2 = freq_n_vdd_068_2.data(:,2);

freq_n_vdd_069_2= importdata('m_da_3300_a_3499_vdd_069_a_1_passo_001.csv');
n_vdd_069_2 = freq_n_vdd_069_2.data(:,2);

freq_n_vdd_070_2= importdata('m_da_3300_a_3499_vdd_070_a_1_passo_001.csv');
n_vdd_070_2 = freq_n_vdd_070_2.data(:,2);

freq_n_vdd_071_2= importdata('m_da_3300_a_3499_vdd_071_a_1_passo_001.csv');
n_vdd_071_2 = freq_n_vdd_071_2.data(:,2);

freq_n_vdd_072_2= importdata('m_da_3300_a_3499_vdd_072_a_1_passo_001.csv');
n_vdd_072_2 = freq_n_vdd_072_2.data(:,2);

freq_n_vdd_073_2= importdata('m_da_3300_a_3499_vdd_073_a_1_passo_001.csv');
n_vdd_073_2 = freq_n_vdd_073_2.data(:,2);

freq_n_vdd_074_2= importdata('m_da_3300_a_3499_vdd_074_a_1_passo_001.csv');
n_vdd_074_2 = freq_n_vdd_074_2.data(:,2);

freq_n_vdd_075_2= importdata('m_da_3300_a_3499_vdd_075_a_1_passo_001.csv');
n_vdd_075_2 = freq_n_vdd_075_2.data(:,2);

freq_n_vdd_076_2= importdata('m_da_3300_a_3499_vdd_076_a_1_passo_001.csv');
n_vdd_076_2 = freq_n_vdd_076_2.data(:,2);

freq_n_vdd_077_2= importdata('m_da_3300_a_3499_vdd_077_a_1_passo_001.csv');
n_vdd_077_2 = freq_n_vdd_077_2.data(:,2);

freq_n_vdd_078_2= importdata('m_da_3300_a_3499_vdd_078_a_1_passo_001.csv');
```

```
n_vdd_078_2 = freq_n_vdd_078_2.data(:,2);

freq_n_vdd_079_2= importdata('m_da_3300_a_3499_vdd_079_a_1_passo_001.csv');
n_vdd_079_2 = freq_n_vdd_079_2.data(:,2);

freq_n_vdd_080_2= importdata('m_da_3300_a_3499_vdd_080_a_1_passo_001.csv');
n_vdd_080_2 = freq_n_vdd_080_2.data(:,2);

freq_n_vdd_081_2= importdata('m_da_3300_a_3499_vdd_081_a_1_passo_001.csv');
n_vdd_081_2 = freq_n_vdd_081_2.data(:,2);

freq_n_vdd_082_2= importdata('m_da_3300_a_3499_vdd_082_a_1_passo_001.csv');
n_vdd_082_2 = freq_n_vdd_082_2.data(:,2);

freq_n_vdd_083_2= importdata('m_da_3300_a_3499_vdd_083_a_1_passo_001.csv');
n_vdd_083_2 = freq_n_vdd_083_2.data(:,2);

freq_n_vdd_084_2= importdata('m_da_3300_a_3499_vdd_084_a_1_passo_001.csv');
n_vdd_084_2 = freq_n_vdd_084_2.data(:,2);

freq_n_vdd_085_2= importdata('m_da_3300_a_3499_vdd_085_a_1_passo_001.csv');
n_vdd_085_2 = freq_n_vdd_085_2.data(:,2);

freq_n_vdd_086_2= importdata('m_da_3300_a_3499_vdd_086_a_1_passo_001.csv');
n_vdd_086_2 = freq_n_vdd_086_2.data(:,2);

freq_n_vdd_087_2= importdata('m_da_3300_a_3499_vdd_087_a_1_passo_001.csv');
n_vdd_087_2 = freq_n_vdd_087_2.data(:,2);

freq_n_vdd_088_2= importdata('m_da_3300_a_3499_vdd_088_a_1_passo_001.csv');
n_vdd_088_2 = freq_n_vdd_088_2.data(:,2);

freq_n_vdd_089_2= importdata('m_da_3300_a_3499_vdd_089_a_1_passo_001.csv');
n_vdd_089_2 = freq_n_vdd_089_2.data(:,2);

freq_n_vdd_090_2= importdata('m_da_3300_a_3499_vdd_090_a_1_passo_001.csv');
n_vdd_090_2 = freq_n_vdd_090_2.data(:,2);

freq_n_vdd_091_2= importdata('m_da_3300_a_3499_vdd_091_a_1_passo_001.csv');
n_vdd_091_2 = freq_n_vdd_091_2.data(:,2);

freq_n_vdd_092_2= importdata('m_da_3300_a_3499_vdd_092_a_1_passo_001.csv');
n_vdd_092_2 = freq_n_vdd_092_2.data(:,2);

freq_n_vdd_093_2= importdata('m_da_3300_a_3499_vdd_093_a_1_passo_001.csv');
n_vdd_093_2 = freq_n_vdd_093_2.data(:,2);

freq_n_vdd_094_2= importdata('m_da_3300_a_3499_vdd_094_a_1_passo_001.csv');
n_vdd_094_2 = freq_n_vdd_094_2.data(:,2);

freq_n_vdd_095_2= importdata('m_da_3300_a_3499_vdd_095_a_1_passo_001.csv');
n_vdd_095_2 = freq_n_vdd_095_2.data(:,2);

freq_n_vdd_096_2= importdata('m_da_3300_a_3499_vdd_096_a_1_passo_001.csv');
n_vdd_096_2 = freq_n_vdd_096_2.data(:,2);
```

```

freq_n_vdd_097_2= importdata('m_da_3300_a_3499_vdd_097_a_1_passo_001.csv');
n_vdd_097_2 = freq_n_vdd_097_2.data(:,2);

freq_n_vdd_098_2= importdata('m_da_3300_a_3499_vdd_098_a_1_passo_001.csv');
n_vdd_098_2 = freq_n_vdd_098_2.data(:,2);

freq_n_vdd_099_2= importdata('m_da_3300_a_3499_vdd_099_a_1_passo_001.csv');
n_vdd_099_2 = freq_n_vdd_099_2.data(:,2);

freq_n_vdd_01_2= importdata('m_da_3300_a_3499_vdd_01_a_1_passo_001.csv');
n_vdd_01_2 = freq_n_vdd_01_2.data(:,2);


freq_n_vdd_050_3= importdata('m_da_3500_a_3699_vdd_050_a_1_passo_001.csv');
n_vdd_050_3 = freq_n_vdd_050_3.data(:,2);

freq_n_vdd_051_3= importdata('m_da_3500_a_3699_vdd_051_a_1_passo_001.csv');
n_vdd_051_3 = freq_n_vdd_051_3.data(:,2);

freq_n_vdd_052_3= importdata('m_da_3500_a_3699_vdd_052_a_1_passo_001.csv');
n_vdd_052_3 = freq_n_vdd_052_3.data(:,2);

freq_n_vdd_053_3= importdata('m_da_3500_a_3699_vdd_053_a_1_passo_001.csv');
n_vdd_053_3 = freq_n_vdd_053_3.data(:,2);

freq_n_vdd_054_3= importdata('m_da_3500_a_3699_vdd_054_a_1_passo_001.csv');
n_vdd_054_3 = freq_n_vdd_054_3.data(:,2);

freq_n_vdd_055_3= importdata('m_da_3500_a_3699_vdd_055_a_1_passo_001.csv');
n_vdd_055_3 = freq_n_vdd_055_3.data(:,2);

freq_n_vdd_056_3= importdata('m_da_3500_a_3699_vdd_056_a_1_passo_001.csv');
n_vdd_056_3 = freq_n_vdd_056_3.data(:,2);

freq_n_vdd_057_3= importdata('m_da_3500_a_3699_vdd_057_a_1_passo_001.csv');
n_vdd_057_3 = freq_n_vdd_057_3.data(:,2);

freq_n_vdd_058_3= importdata('m_da_3500_a_3699_vdd_058_a_1_passo_001.csv');
n_vdd_058_3 = freq_n_vdd_058_3.data(:,2);

freq_n_vdd_059_3= importdata('m_da_3500_a_3699_vdd_059_a_1_passo_001.csv');
n_vdd_059_3 = freq_n_vdd_059_3.data(:,2);

freq_n_vdd_060_3= importdata('m_da_3500_a_3699_vdd_060_a_1_passo_001.csv');
n_vdd_060_3 = freq_n_vdd_060_3.data(:,2);

freq_n_vdd_061_3= importdata('m_da_3500_a_3699_vdd_061_a_1_passo_001.csv');
n_vdd_061_3 = freq_n_vdd_061_3.data(:,2);

```

```
freq_n_vdd_062_3= importdata('m_da_3500_a_3699_vdd_062_a_1_passo_001.csv');
n_vdd_062_3 = freq_n_vdd_062_3.data(:,2);

freq_n_vdd_063_3= importdata('m_da_3500_a_3699_vdd_063_a_1_passo_001.csv');
n_vdd_063_3 = freq_n_vdd_063_3.data(:,2);

freq_n_vdd_064_3= importdata('m_da_3500_a_3699_vdd_064_a_1_passo_001.csv');
n_vdd_064_3 = freq_n_vdd_064_3.data(:,2);

freq_n_vdd_065_3= importdata('m_da_3500_a_3699_vdd_065_a_1_passo_001.csv');
n_vdd_065_3 = freq_n_vdd_065_3.data(:,2);

freq_n_vdd_066_3= importdata('m_da_3500_a_3699_vdd_066_a_1_passo_001.csv');
n_vdd_066_3 = freq_n_vdd_066_3.data(:,2);

freq_n_vdd_067_3= importdata('m_da_3500_a_3699_vdd_067_a_1_passo_001.csv');
n_vdd_067_3 = freq_n_vdd_067_3.data(:,2);

freq_n_vdd_068_3= importdata('m_da_3500_a_3699_vdd_068_a_1_passo_001.csv');
n_vdd_068_3 = freq_n_vdd_068_3.data(:,2);

freq_n_vdd_069_3= importdata('m_da_3500_a_3699_vdd_069_a_1_passo_001.csv');
n_vdd_069_3 = freq_n_vdd_069_3.data(:,2);

freq_n_vdd_070_3= importdata('m_da_3500_a_3699_vdd_070_a_1_passo_001.csv');
n_vdd_070_3 = freq_n_vdd_070_3.data(:,2);

freq_n_vdd_071_3= importdata('m_da_3500_a_3699_vdd_071_a_1_passo_001.csv');
n_vdd_071_3 = freq_n_vdd_071_3.data(:,2);

freq_n_vdd_072_3= importdata('m_da_3500_a_3699_vdd_072_a_1_passo_001.csv');
n_vdd_072_3 = freq_n_vdd_072_3.data(:,2);

freq_n_vdd_073_3= importdata('m_da_3500_a_3699_vdd_073_a_1_passo_001.csv');
n_vdd_073_3 = freq_n_vdd_073_3.data(:,2);

freq_n_vdd_074_3= importdata('m_da_3500_a_3699_vdd_074_a_1_passo_001.csv');
n_vdd_074_3 = freq_n_vdd_074_3.data(:,2);

freq_n_vdd_075_3= importdata('m_da_3500_a_3699_vdd_075_a_1_passo_001.csv');
n_vdd_075_3 = freq_n_vdd_075_3.data(:,2);

freq_n_vdd_076_3= importdata('m_da_3500_a_3699_vdd_076_a_1_passo_001.csv');
n_vdd_076_3 = freq_n_vdd_076_3.data(:,2);

freq_n_vdd_077_3= importdata('m_da_3500_a_3699_vdd_077_a_1_passo_001.csv');
n_vdd_077_3 = freq_n_vdd_077_3.data(:,2);

freq_n_vdd_078_3= importdata('m_da_3500_a_3699_vdd_078_a_1_passo_001.csv');
n_vdd_078_3 = freq_n_vdd_078_3.data(:,2);

freq_n_vdd_079_3= importdata('m_da_3500_a_3699_vdd_079_a_1_passo_001.csv');
n_vdd_079_3 = freq_n_vdd_079_3.data(:,2);

freq_n_vdd_080_3= importdata('m_da_3500_a_3699_vdd_080_a_1_passo_001.csv');
```

```

n_vdd_080_3 = freq_n_vdd_080_3.data(:,2);

freq_n_vdd_081_3= importdata('m_da_3500_a_3699_vdd_081_a_1_passo_001.csv');
n_vdd_081_3 = freq_n_vdd_081_3.data(:,2);

freq_n_vdd_082_3= importdata('m_da_3500_a_3699_vdd_082_a_1_passo_001.csv');
n_vdd_082_3 = freq_n_vdd_082_3.data(:,2);

freq_n_vdd_083_3= importdata('m_da_3500_a_3699_vdd_083_a_1_passo_001.csv');
n_vdd_083_3 = freq_n_vdd_083_3.data(:,2);

freq_n_vdd_084_3= importdata('m_da_3500_a_3699_vdd_084_a_1_passo_001.csv');
n_vdd_084_3 = freq_n_vdd_084_3.data(:,2);

freq_n_vdd_085_3= importdata('m_da_3500_a_3699_vdd_085_a_1_passo_001.csv');
n_vdd_085_3 = freq_n_vdd_085_3.data(:,2);

freq_n_vdd_086_3= importdata('m_da_3500_a_3699_vdd_086_a_1_passo_001.csv');
n_vdd_086_3 = freq_n_vdd_086_3.data(:,2);

freq_n_vdd_087_3= importdata('m_da_3500_a_3699_vdd_087_a_1_passo_001.csv');
n_vdd_087_3 = freq_n_vdd_087_3.data(:,2);

freq_n_vdd_088_3= importdata('m_da_3500_a_3699_vdd_088_a_1_passo_001.csv');
n_vdd_088_3 = freq_n_vdd_088_3.data(:,2);

freq_n_vdd_089_3= importdata('m_da_3500_a_3699_vdd_089_a_1_passo_001.csv');
n_vdd_089_3 = freq_n_vdd_089_3.data(:,2);

freq_n_vdd_090_3= importdata('m_da_3500_a_3699_vdd_090_a_1_passo_001.csv');
n_vdd_090_3 = freq_n_vdd_090_3.data(:,2);

freq_n_vdd_091_3= importdata('m_da_3500_a_3699_vdd_091_a_1_passo_001.csv');
n_vdd_091_3 = freq_n_vdd_091_3.data(:,2);

freq_n_vdd_092_3= importdata('m_da_3500_a_3699_vdd_092_a_1_passo_001.csv');
n_vdd_092_3 = freq_n_vdd_092_3.data(:,2);

freq_n_vdd_093_3= importdata('m_da_3500_a_3699_vdd_093_a_1_passo_001.csv');
n_vdd_093_3 = freq_n_vdd_093_3.data(:,2);

freq_n_vdd_094_3= importdata('m_da_3500_a_3699_vdd_094_a_1_passo_001.csv');
n_vdd_094_3 = freq_n_vdd_094_3.data(:,2);

freq_n_vdd_095_3= importdata('m_da_3500_a_3699_vdd_095_a_1_passo_001.csv');
n_vdd_095_3 = freq_n_vdd_095_3.data(:,2);

freq_n_vdd_096_3= importdata('m_da_3500_a_3699_vdd_096_a_1_passo_001.csv');
n_vdd_096_3 = freq_n_vdd_096_3.data(:,2);

freq_n_vdd_097_3= importdata('m_da_3500_a_3699_vdd_097_a_1_passo_001.csv');
n_vdd_097_3 = freq_n_vdd_097_3.data(:,2);

freq_n_vdd_098_3= importdata('m_da_3500_a_3699_vdd_098_a_1_passo_001.csv');
n_vdd_098_3 = freq_n_vdd_098_3.data(:,2);

```

```
freq_n_vdd_099_3= importdata('m_da_3500_a_3699_vdd_099_a_1_passo_001.csv');
n_vdd_099_3 = freq_n_vdd_099_3.data(:,2);

freq_n_vdd_01_3= importdata('m_da_3500_a_3699_vdd_01_a_1_passo_001.csv');
n_vdd_01_3 = freq_n_vdd_01_3.data(:,2);

% NON VENGONO RIPORTATI TUTTE LE RIGHE DURANTE LA FASE DI IMPORTAZIONE
% DEI circa 1600 VETTORI CONTENENTI I DATI DELLE SIMULAZIONI SU CADENCE

freq_n_vdd_050_30= importdata('m_da_8051_a_8192_vdd_050_a_1_passo_001.csv');
n_vdd_050_30 = freq_n_vdd_050_30.data(:,2);

freq_n_vdd_051_30= importdata('m_da_8051_a_8192_vdd_051_a_1_passo_001.csv');
n_vdd_051_30 = freq_n_vdd_051_30.data(:,2);

freq_n_vdd_052_30= importdata('m_da_8051_a_8192_vdd_052_a_1_passo_001.csv');
n_vdd_052_30 = freq_n_vdd_052_30.data(:,2);

freq_n_vdd_053_30= importdata('m_da_8051_a_8192_vdd_053_a_1_passo_001.csv');
n_vdd_053_30 = freq_n_vdd_053_30.data(:,2);

freq_n_vdd_054_30= importdata('m_da_8051_a_8192_vdd_054_a_1_passo_001.csv');
n_vdd_054_30 = freq_n_vdd_054_30.data(:,2);

freq_n_vdd_055_30= importdata('m_da_8051_a_8192_vdd_055_a_1_passo_001.csv');
n_vdd_055_30 = freq_n_vdd_055_30.data(:,2);

freq_n_vdd_056_30= importdata('m_da_8051_a_8192_vdd_056_a_1_passo_001.csv');
n_vdd_056_30 = freq_n_vdd_056_30.data(:,2);

freq_n_vdd_057_30= importdata('m_da_8051_a_8192_vdd_057_a_1_passo_001.csv');
n_vdd_057_30 = freq_n_vdd_057_30.data(:,2);

freq_n_vdd_058_30= importdata('m_da_8051_a_8192_vdd_058_a_1_passo_001.csv');
n_vdd_058_30 = freq_n_vdd_058_30.data(:,2);

freq_n_vdd_059_30= importdata('m_da_8051_a_8192_vdd_059_a_1_passo_001.csv');
n_vdd_059_30 = freq_n_vdd_059_30.data(:,2);

freq_n_vdd_060_30= importdata('m_da_8051_a_8192_vdd_060_a_1_passo_001.csv');
n_vdd_060_30 = freq_n_vdd_060_30.data(:,2);

freq_n_vdd_061_30= importdata('m_da_8051_a_8192_vdd_061_a_1_passo_001.csv');
n_vdd_061_30 = freq_n_vdd_061_30.data(:,2);

freq_n_vdd_062_30= importdata('m_da_8051_a_8192_vdd_062_a_1_passo_001.csv');
n_vdd_062_30 = freq_n_vdd_062_30.data(:,2);

freq_n_vdd_063_30= importdata('m_da_8051_a_8192_vdd_063_a_1_passo_001.csv');
n_vdd_063_30 = freq_n_vdd_063_30.data(:,2);

freq_n_vdd_064_30= importdata('m_da_8051_a_8192_vdd_064_a_1_passo_001.csv');
n_vdd_064_30 = freq_n_vdd_064_30.data(:,2);
```

```

freq_n_vdd_065_30= importdata('m_da_8051_a_8192_vdd_065_a_1_passo_001.csv');
n_vdd_065_30 = freq_n_vdd_065_30.data(:,2);

freq_n_vdd_066_30= importdata('m_da_8051_a_8192_vdd_066_a_1_passo_001.csv');
n_vdd_066_30 = freq_n_vdd_066_30.data(:,2);

freq_n_vdd_067_30= importdata('m_da_8051_a_8192_vdd_067_a_1_passo_001.csv');
n_vdd_067_30 = freq_n_vdd_067_30.data(:,2);

freq_n_vdd_068_30= importdata('m_da_8051_a_8192_vdd_068_a_1_passo_001.csv');
n_vdd_068_30 = freq_n_vdd_068_30.data(:,2);

freq_n_vdd_069_30= importdata('m_da_8051_a_8192_vdd_069_a_1_passo_001.csv');
n_vdd_069_30 = freq_n_vdd_069_30.data(:,2);

freq_n_vdd_070_30= importdata('m_da_8051_a_8192_vdd_070_a_1_passo_001.csv');
n_vdd_070_30 = freq_n_vdd_070_30.data(:,2);

freq_n_vdd_071_30= importdata('m_da_8051_a_8192_vdd_071_a_1_passo_001.csv');
n_vdd_071_30 = freq_n_vdd_071_30.data(:,2);

freq_n_vdd_072_30= importdata('m_da_8051_a_8192_vdd_072_a_1_passo_001.csv');
n_vdd_072_30 = freq_n_vdd_072_30.data(:,2);

freq_n_vdd_073_30= importdata('m_da_8051_a_8192_vdd_073_a_1_passo_001.csv');
n_vdd_073_30 = freq_n_vdd_073_30.data(:,2);

freq_n_vdd_074_30= importdata('m_da_8051_a_8192_vdd_074_a_1_passo_001.csv');
n_vdd_074_30 = freq_n_vdd_074_30.data(:,2);

freq_n_vdd_075_30= importdata('m_da_8051_a_8192_vdd_075_a_1_passo_001.csv');
n_vdd_075_30 = freq_n_vdd_075_30.data(:,2);

freq_n_vdd_076_30= importdata('m_da_8051_a_8192_vdd_076_a_1_passo_001.csv');
n_vdd_076_30 = freq_n_vdd_076_30.data(:,2);

freq_n_vdd_077_30= importdata('m_da_8051_a_8192_vdd_077_a_1_passo_001.csv');
n_vdd_077_30 = freq_n_vdd_077_30.data(:,2);

freq_n_vdd_078_30= importdata('m_da_8051_a_8192_vdd_078_a_1_passo_001.csv');
n_vdd_078_30 = freq_n_vdd_078_30.data(:,2);

freq_n_vdd_079_30= importdata('m_da_8051_a_8192_vdd_079_a_1_passo_001.csv');
n_vdd_079_30 = freq_n_vdd_079_30.data(:,2);

freq_n_vdd_080_30= importdata('m_da_8051_a_8192_vdd_080_a_1_passo_001.csv');
n_vdd_080_30 = freq_n_vdd_080_30.data(:,2);

freq_n_vdd_081_30= importdata('m_da_8051_a_8192_vdd_081_a_1_passo_001.csv');
n_vdd_081_30 = freq_n_vdd_081_30.data(:,2);

freq_n_vdd_082_30= importdata('m_da_8051_a_8192_vdd_082_a_1_passo_001.csv');
n_vdd_082_30 = freq_n_vdd_082_30.data(:,2);

```



```
freq_n_vdd_083_30= importdata('m_da_8051_a_8192_vdd_083_a_1_passo_001.csv');
n_vdd_083_30 = freq_n_vdd_083_30.data(:,2);

freq_n_vdd_084_30= importdata('m_da_8051_a_8192_vdd_084_a_1_passo_001.csv');
n_vdd_084_30 = freq_n_vdd_084_30.data(:,2);

freq_n_vdd_085_30= importdata('m_da_8051_a_8192_vdd_085_a_1_passo_001.csv');
n_vdd_085_30 = freq_n_vdd_085_30.data(:,2);

freq_n_vdd_086_30= importdata('m_da_8051_a_8192_vdd_086_a_1_passo_001.csv');
n_vdd_086_30 = freq_n_vdd_086_30.data(:,2);

freq_n_vdd_087_30= importdata('m_da_8051_a_8192_vdd_087_a_1_passo_001.csv');
n_vdd_087_30 = freq_n_vdd_087_30.data(:,2);

freq_n_vdd_088_30= importdata('m_da_8051_a_8192_vdd_088_a_1_passo_001.csv');
n_vdd_088_30 = freq_n_vdd_088_30.data(:,2);

freq_n_vdd_089_30= importdata('m_da_8051_a_8192_vdd_089_a_1_passo_001.csv');
n_vdd_089_30 = freq_n_vdd_089_30.data(:,2);

freq_n_vdd_090_30= importdata('m_da_8051_a_8192_vdd_090_a_1_passo_001.csv');
n_vdd_090_30 = freq_n_vdd_090_30.data(:,2);

freq_n_vdd_091_30= importdata('m_da_8051_a_8192_vdd_091_a_1_passo_001.csv');
n_vdd_091_30 = freq_n_vdd_091_30.data(:,2);

freq_n_vdd_092_30= importdata('m_da_8051_a_8192_vdd_092_a_1_passo_001.csv');
n_vdd_092_30 = freq_n_vdd_092_30.data(:,2);

freq_n_vdd_093_30= importdata('m_da_8051_a_8192_vdd_093_a_1_passo_001.csv');
n_vdd_093_30 = freq_n_vdd_093_30.data(:,2);

freq_n_vdd_094_30= importdata('m_da_8051_a_8192_vdd_094_a_1_passo_001.csv');
n_vdd_094_30 = freq_n_vdd_094_30.data(:,2);

freq_n_vdd_095_30= importdata('m_da_8051_a_8192_vdd_095_a_1_passo_001.csv');
n_vdd_095_30 = freq_n_vdd_095_30.data(:,2);

freq_n_vdd_096_30= importdata('m_da_8051_a_8192_vdd_096_a_1_passo_001.csv');
n_vdd_096_30 = freq_n_vdd_096_30.data(:,2);

freq_n_vdd_097_30= importdata('m_da_8051_a_8192_vdd_097_a_1_passo_001.csv');
n_vdd_097_30 = freq_n_vdd_097_30.data(:,2);

freq_n_vdd_098_30= importdata('m_da_8051_a_8192_vdd_098_a_1_passo_001.csv');
n_vdd_098_30 = freq_n_vdd_098_30.data(:,2);

freq_n_vdd_099_30= importdata('m_da_8051_a_8192_vdd_099_a_1_passo_001.csv');
n_vdd_099_30 = freq_n_vdd_099_30.data(:,2);

freq_n_vdd_01_30= importdata('m_da_8051_a_8192_vdd_01_a_1_passo_001.csv');
n_vdd_01_30 = freq_n_vdd_01_30.data(:,2);
```

% FINE IMPORTAZIONE DEI DATI

```
% matrice contiene tutte le frequenze ottenute dalle simulazioni
matrice= zeros(length(m),length(vdd));
```

```
% % concateno tutti i vettori colonna importati
```

```
n_vdd_050_mix= cat(1,n_vdd_050_1,n_vdd_050_2, n_vdd_050_3,... ,n_vdd_050_30);
```

```
n_vdd_051_mix= cat(1,n_vdd_051_1,n_vdd_051_2 , n_vdd_051_3, ... , n_vdd_051_30);
n_vdd_052_mix= cat(1,n_vdd_052_1,n_vdd_052_2, n_vdd_052_3, ... , n_vdd_052_30);
n_vdd_053_mix= cat(1,n_vdd_053_1,n_vdd_053_2, n_vdd_053_3, ... , n_vdd_053_30);
n_vdd_054_mix= cat(1,n_vdd_054_1,n_vdd_054_2, n_vdd_054_3, ... , n_vdd_054_30);
n_vdd_055_mix= cat(1,n_vdd_055_1,n_vdd_055_2 , n_vdd_055_3, ... , n_vdd_055_30);
n_vdd_056_mix= cat(1,n_vdd_056_1,n_vdd_056_2, n_vdd_056_3, ... , n_vdd_056_30);
n_vdd_057_mix= cat(1,n_vdd_057_1,n_vdd_057_2 , n_vdd_057_3, ... , n_vdd_057_30);
n_vdd_058_mix= cat(1,n_vdd_058_1,n_vdd_058_2 , n_vdd_058_3, ... , n_vdd_058_30);
n_vdd_059_mix= cat(1,n_vdd_059_1,n_vdd_059_2 , n_vdd_059_3, ... , n_vdd_059_30);
n_vdd_060_mix= cat(1,n_vdd_060_1,n_vdd_060_2 , n_vdd_060_3, ... , n_vdd_060_30);
n_vdd_061_mix= cat(1,n_vdd_061_1,n_vdd_061_2 , n_vdd_061_3, ... , n_vdd_061_30);
n_vdd_062_mix= cat(1,n_vdd_062_1,n_vdd_062_2 , n_vdd_062_3, ... , n_vdd_062_30);
n_vdd_063_mix= cat(1,n_vdd_063_1,n_vdd_063_2 , n_vdd_063_3, ... , n_vdd_063_30);
n_vdd_064_mix= cat(1,n_vdd_064_1,n_vdd_064_2 , n_vdd_064_3, ... , n_vdd_064_30);
n_vdd_065_mix= cat(1,n_vdd_065_1,n_vdd_065_2 , n_vdd_065_3, ... , n_vdd_065_30);
n_vdd_066_mix= cat(1,n_vdd_066_1,n_vdd_066_2 , n_vdd_066_3, ... , n_vdd_066_30);
n_vdd_067_mix= cat(1,n_vdd_067_1,n_vdd_067_2 , n_vdd_067_3, ... , n_vdd_067_30);
n_vdd_068_mix= cat(1,n_vdd_068_1,n_vdd_068_2 , n_vdd_068_3, ... , n_vdd_068_30);
n_vdd_069_mix= cat(1,n_vdd_069_1,n_vdd_069_2 , n_vdd_069_3, ... , n_vdd_069_30);
n_vdd_070_mix= cat(1,n_vdd_070_1,n_vdd_070_2 , n_vdd_070_3, ... , n_vdd_070_30);
n_vdd_071_mix= cat(1,n_vdd_071_1,n_vdd_071_2 , n_vdd_071_3, ... , n_vdd_071_30);
n_vdd_072_mix= cat(1,n_vdd_072_1,n_vdd_072_2 , n_vdd_072_3, ... , n_vdd_072_30);
n_vdd_073_mix= cat(1,n_vdd_073_1,n_vdd_073_2 , n_vdd_073_3, ... , n_vdd_073_30);
n_vdd_074_mix= cat(1,n_vdd_074_1,n_vdd_074_2 , n_vdd_074_3, ... , n_vdd_074_30);
n_vdd_075_mix= cat(1,n_vdd_075_1,n_vdd_075_2 , n_vdd_075_3, ... , n_vdd_075_30);
n_vdd_076_mix= cat(1,n_vdd_076_1,n_vdd_076_2 , n_vdd_076_3, ... , n_vdd_076_30);
n_vdd_077_mix= cat(1,n_vdd_077_1,n_vdd_077_2 , n_vdd_077_3, ... , n_vdd_077_30);
n_vdd_078_mix= cat(1,n_vdd_078_1,n_vdd_078_2 , n_vdd_078_3, ... , n_vdd_078_30);
n_vdd_079_mix= cat(1,n_vdd_079_1,n_vdd_079_2 , n_vdd_079_3, ... , n_vdd_079_30);
n_vdd_080_mix= cat(1,n_vdd_080_1,n_vdd_080_2 , n_vdd_080_3, ... , n_vdd_080_30);
n_vdd_081_mix= cat(1,n_vdd_081_1,n_vdd_081_2 , n_vdd_081_3, ... , n_vdd_081_30);
n_vdd_082_mix= cat(1,n_vdd_082_1,n_vdd_082_2 , n_vdd_082_3, ... , n_vdd_082_30);
n_vdd_083_mix= cat(1,n_vdd_083_1,n_vdd_083_2 , n_vdd_083_3, ... , n_vdd_083_30);
n_vdd_084_mix= cat(1,n_vdd_084_1,n_vdd_084_2 , n_vdd_084_3, ... , n_vdd_084_30);
n_vdd_085_mix= cat(1,n_vdd_085_1,n_vdd_085_2 , n_vdd_085_3, ... , n_vdd_085_30);
n_vdd_086_mix= cat(1,n_vdd_086_1,n_vdd_086_2 , n_vdd_086_3, ... , n_vdd_086_30);
n_vdd_087_mix= cat(1,n_vdd_087_1,n_vdd_087_2 , n_vdd_087_3, ... , n_vdd_087_30);
n_vdd_088_mix= cat(1,n_vdd_088_1,n_vdd_088_2 , n_vdd_088_3, ... , n_vdd_088_30);
n_vdd_089_mix= cat(1,n_vdd_089_1,n_vdd_089_2 , n_vdd_089_3, ... , n_vdd_089_30);
n_vdd_090_mix= cat(1,n_vdd_090_1,n_vdd_090_2 , n_vdd_090_3, ... , n_vdd_090_30);
n_vdd_091_mix= cat(1,n_vdd_091_1,n_vdd_091_2 , n_vdd_091_3, ... , n_vdd_091_30);
n_vdd_092_mix= cat(1,n_vdd_092_1,n_vdd_092_2 , n_vdd_092_3, ... , n_vdd_092_30);
n_vdd_093_mix= cat(1,n_vdd_093_1,n_vdd_093_2 , n_vdd_093_3, ... , n_vdd_093_30);
n_vdd_094_mix= cat(1,n_vdd_094_1,n_vdd_094_2 , n_vdd_094_3, ... , n_vdd_094_30);
n_vdd_095_mix= cat(1,n_vdd_095_1,n_vdd_095_2, n_vdd_095_3, ... , n_vdd_095_30);
n_vdd_096_mix= cat(1,n_vdd_096_1,n_vdd_096_2 , n_vdd_096_3, ... , n_vdd_096_30);
n_vdd_097_mix= cat(1,n_vdd_097_1,n_vdd_097_2 , n_vdd_097_3, ... , n_vdd_097_30);
n_vdd_098_mix= cat(1,n_vdd_098_1,n_vdd_098_2 , n_vdd_098_3, ... , n_vdd_098_30);
```

```
n_vdd_099_mix= cat(1,n_vdd_099_1,n_vdd_099_2 , n_vdd_099_3, ... , n_vdd_099_30);
n_vdd_01_mix= cat(1,n_vdd_01_1,n_vdd_01_2 , n_vdd_01_3, ... , n_vdd_01_30);

% generazione della matrice
% ad esempio matrice(1,1) è quella frequenza relativa al numero 3200 con
% Vdd= 0.5V
% matrice(2,51) è quella frequenza relativa al numero 3201 con vdd=1V
matrice(:,1)=n_vdd_050_mix;
matrice(:,2)=n_vdd_051_mix;
matrice(:,3)=n_vdd_052_mix;
matrice(:,4)=n_vdd_053_mix;
matrice(:,5)=n_vdd_054_mix;
matrice(:,6)=n_vdd_055_mix;
matrice(:,7)=n_vdd_056_mix;
matrice(:,8)=n_vdd_057_mix;
matrice(:,9)=n_vdd_058_mix;
matrice(:,10)=n_vdd_059_mix;
matrice(:,11)=n_vdd_060_mix;
matrice(:,12)=n_vdd_061_mix;
matrice(:,13)=n_vdd_062_mix;
matrice(:,14)=n_vdd_063_mix;
matrice(:,15)=n_vdd_064_mix;
matrice(:,16)=n_vdd_065_mix;
matrice(:,17)=n_vdd_066_mix;
matrice(:,18)=n_vdd_067_mix;
matrice(:,19)=n_vdd_068_mix;
matrice(:,20)=n_vdd_069_mix;
matrice(:,21)=n_vdd_070_mix;
matrice(:,22)=n_vdd_071_mix;
matrice(:,23)=n_vdd_072_mix;
matrice(:,24)=n_vdd_073_mix;
matrice(:,25)=n_vdd_074_mix;
matrice(:,26)=n_vdd_075_mix;
matrice(:,27)=n_vdd_076_mix;
matrice(:,28)=n_vdd_077_mix;
matrice(:,29)=n_vdd_078_mix;
matrice(:,30)=n_vdd_079_mix;
matrice(:,31)=n_vdd_080_mix;
matrice(:,32)=n_vdd_081_mix;
matrice(:,33)=n_vdd_082_mix;
matrice(:,34)=n_vdd_083_mix;
matrice(:,35)=n_vdd_084_mix;
matrice(:,36)=n_vdd_085_mix;
matrice(:,37)=n_vdd_086_mix;
matrice(:,38)=n_vdd_087_mix;
matrice(:,39)=n_vdd_088_mix;
matrice(:,40)=n_vdd_089_mix;
matrice(:,41)=n_vdd_090_mix;
matrice(:,42)=n_vdd_091_mix;
matrice(:,43)=n_vdd_092_mix;
matrice(:,44)=n_vdd_093_mix;
matrice(:,45)=n_vdd_094_mix;
matrice(:,46)=n_vdd_095_mix;
matrice(:,47)=n_vdd_096_mix;
```

```
matrice(:,48)=n_vdd_097_mix;
matrice(:,49)=n_vdd_098_mix;
matrice(:,50)=n_vdd_099_mix;
matrice(:,51)=n_vdd_01_mix;

save matrice matrice

load matrice

k=1; % k=V_REF/V0

flag=1; % flag utile per capire quando passare al secondo giro dell'algoritmo

for i=1:length(vdd) % processo ripetuto per ogni vdd

    if flag==1 % vuol dire 'se sono al primo giro'

        % A= (2^N/(m1-m2))*log(n1/n2)
        A_init= (2^N/(m(3000)-m(4000)))*log(matrice(3000,i)/matrice(4000,i));

        r_1= (k/A_init)*2^N;

        %vengono definiti gli indici del vettore m per ciclo successivo
        j=100;
        p=405;
    end

    diff=1; % inizializzo r-r_old=1 per entrare nel ciclo while
    diff_old=100;

    while diff < diff_old % abs(r-r_old)<eps?

        m1=m(j);
        m2=m(p);

        A_2= (2^N/(m1-m2))*log(matrice(j,i)/matrice(p,i));

        r_2= (k/A_2)*2^N % calcolo del nuovo r
        diff_old=diff;
        diff= abs(r_1-r_2)
        r_1=r_2 % utile per giro successivo

        k_1= .5;
        k_2= 1.8;

        % definisco dei nuovi m1 ed m2 per il prossimo giro riferiti ad r_1
        m1= round(k_1*r_1);
        m2= round(k_2*r_1);
```

```
% i seguenti 2 cicli for servono per individuare in quale posizione
% si trovano m1 e m2 nel vettore m : per esempio se m1=4010
% quanto vale w di m(w)?
for ell=1:length(m)
    if m1==m(ell)
        j=ell;
    end
end

for ell=1:length(m)
    if m2==m(ell)
        p=ell;
    end
end

end
r_1

riferimento(i)=r_1; % salvo i diversi r_1 per ogni vdd
Vref(i)=r_1*vdd(i)/(2^N); %calcolo di Vref
    flag=0; % non sono più nel primo giro

end

plot(vdd, Vref, '*b')
xlabel('Vdd [V]')
ylabel('r*Vdd/2^N [V]')
grid on
figure(2)
plot(vdd, riferimento, '*r')
xlabel('Vdd [V]')
ylabel('r')
grid on
line_sensitivity= ((max(Vref)-min(Vref))/mean(Vref))*100*2
```

Appendice B

Codice MATLAB con compensazione empirica del numero A

```
close all
clear all
format long
```

```
% T=27°C!
```

```
m= linspace(3200,8192 ,4993); % numeri di ingresso al DAC
vdd= linspace(0.5,1,51); % vdd varia tra 0.5V e 1V a passo di 0.01V
N=14; % numero di bit del DAC
```

```
% INIZIO IMPORTAZIONE DEI DATI
```

```
freq_n_vdd_051_1= importdata('m_da_3200_a_3299_vdd_051_a_1_passo_001.csv');
n_vdd_051_1 = freq_n_vdd_051_1.data(:,2);
```

```
freq_n_vdd_052_1= importdata('m_da_3200_a_3299_vdd_052_a_1_passo_001.csv');
n_vdd_052_1 = freq_n_vdd_052_1.data(:,2);
```

```
freq_n_vdd_053_1= importdata('m_da_3200_a_3299_vdd_053_a_1_passo_001.csv');
n_vdd_053_1 = freq_n_vdd_053_1.data(:,2);
```

```
freq_n_vdd_054_1= importdata('m_da_3200_a_3299_vdd_054_a_1_passo_001.csv');
n_vdd_054_1 = freq_n_vdd_054_1.data(:,2);
```

```
freq_n_vdd_055_1= importdata('m_da_3200_a_3299_vdd_055_a_1_passo_001.csv');
n_vdd_055_1 = freq_n_vdd_055_1.data(:,2);
```

```
freq_n_vdd_056_1= importdata('m_da_3200_a_3299_vdd_056_a_1_passo_001.csv');
n_vdd_056_1 = freq_n_vdd_056_1.data(:,2);
```

```
freq_n_vdd_057_1= importdata('m_da_3200_a_3299_vdd_057_a_1_passo_001.csv');
n_vdd_057_1 = freq_n_vdd_057_1.data(:,2);

freq_n_vdd_058_1= importdata('m_da_3200_a_3299_vdd_058_a_1_passo_001.csv');
n_vdd_058_1 = freq_n_vdd_058_1.data(:,2);

freq_n_vdd_059_1= importdata('m_da_3200_a_3299_vdd_059_a_1_passo_001.csv');
n_vdd_059_1 = freq_n_vdd_059_1.data(:,2);

freq_n_vdd_060_1= importdata('m_da_3200_a_3299_vdd_060_a_1_passo_001.csv');
n_vdd_060_1 = freq_n_vdd_060_1.data(:,2);

freq_n_vdd_061_1= importdata('m_da_3200_a_3299_vdd_061_a_1_passo_001.csv');
n_vdd_061_1 = freq_n_vdd_061_1.data(:,2);

freq_n_vdd_062_1= importdata('m_da_3200_a_3299_vdd_062_a_1_passo_001.csv');
n_vdd_062_1 = freq_n_vdd_062_1.data(:,2);

freq_n_vdd_063_1= importdata('m_da_3200_a_3299_vdd_063_a_1_passo_001.csv');
n_vdd_063_1 = freq_n_vdd_063_1.data(:,2);

freq_n_vdd_064_1= importdata('m_da_3200_a_3299_vdd_064_a_1_passo_001.csv');
n_vdd_064_1 = freq_n_vdd_064_1.data(:,2);

freq_n_vdd_065_1= importdata('m_da_3200_a_3299_vdd_065_a_1_passo_001.csv');
n_vdd_065_1 = freq_n_vdd_065_1.data(:,2);

freq_n_vdd_066_1= importdata('m_da_3200_a_3299_vdd_066_a_1_passo_001.csv');
n_vdd_066_1 = freq_n_vdd_066_1.data(:,2);

freq_n_vdd_067_1= importdata('m_da_3200_a_3299_vdd_067_a_1_passo_001.csv');
n_vdd_067_1 = freq_n_vdd_067_1.data(:,2);

freq_n_vdd_068_1= importdata('m_da_3200_a_3299_vdd_068_a_1_passo_001.csv');
n_vdd_068_1 = freq_n_vdd_068_1.data(:,2);

freq_n_vdd_069_1= importdata('m_da_3200_a_3299_vdd_069_a_1_passo_001.csv');
n_vdd_069_1 = freq_n_vdd_069_1.data(:,2);

freq_n_vdd_070_1= importdata('m_da_3200_a_3299_vdd_070_a_1_passo_001.csv');
n_vdd_070_1 = freq_n_vdd_070_1.data(:,2);

freq_n_vdd_071_1= importdata('m_da_3200_a_3299_vdd_071_a_1_passo_001.csv');
n_vdd_071_1 = freq_n_vdd_071_1.data(:,2);

freq_n_vdd_072_1= importdata('m_da_3200_a_3299_vdd_072_a_1_passo_001.csv');
n_vdd_072_1 = freq_n_vdd_072_1.data(:,2);

freq_n_vdd_073_1= importdata('m_da_3200_a_3299_vdd_073_a_1_passo_001.csv');
n_vdd_073_1 = freq_n_vdd_073_1.data(:,2);

freq_n_vdd_074_1= importdata('m_da_3200_a_3299_vdd_074_a_1_passo_001.csv');
n_vdd_074_1 = freq_n_vdd_074_1.data(:,2);
```

```
freq_n_vdd_075_1= importdata('m_da_3200_a_3299_vdd_075_a_1_passo_001.csv');
n_vdd_075_1 = freq_n_vdd_075_1.data(:,2);

freq_n_vdd_076_1= importdata('m_da_3200_a_3299_vdd_076_a_1_passo_001.csv');
n_vdd_076_1 = freq_n_vdd_076_1.data(:,2);

freq_n_vdd_077_1= importdata('m_da_3200_a_3299_vdd_077_a_1_passo_001.csv');
n_vdd_077_1 = freq_n_vdd_077_1.data(:,2);

freq_n_vdd_078_1= importdata('m_da_3200_a_3299_vdd_078_a_1_passo_001.csv');
n_vdd_078_1 = freq_n_vdd_078_1.data(:,2);

freq_n_vdd_079_1= importdata('m_da_3200_a_3299_vdd_079_a_1_passo_001.csv');
n_vdd_079_1 = freq_n_vdd_079_1.data(:,2);

freq_n_vdd_080_1= importdata('m_da_3200_a_3299_vdd_080_a_1_passo_001.csv');
n_vdd_080_1 = freq_n_vdd_080_1.data(:,2);

freq_n_vdd_081_1= importdata('m_da_3200_a_3299_vdd_081_a_1_passo_001.csv');
n_vdd_081_1 = freq_n_vdd_081_1.data(:,2);

freq_n_vdd_082_1= importdata('m_da_3200_a_3299_vdd_082_a_1_passo_001.csv');
n_vdd_082_1 = freq_n_vdd_082_1.data(:,2);

freq_n_vdd_083_1= importdata('m_da_3200_a_3299_vdd_083_a_1_passo_001.csv');
n_vdd_083_1 = freq_n_vdd_083_1.data(:,2);

freq_n_vdd_084_1= importdata('m_da_3200_a_3299_vdd_084_a_1_passo_001.csv');
n_vdd_084_1 = freq_n_vdd_084_1.data(:,2);

freq_n_vdd_085_1= importdata('m_da_3200_a_3299_vdd_085_a_1_passo_001.csv');
n_vdd_085_1 = freq_n_vdd_085_1.data(:,2);

freq_n_vdd_086_1= importdata('m_da_3200_a_3299_vdd_086_a_1_passo_001.csv');
n_vdd_086_1 = freq_n_vdd_086_1.data(:,2);

freq_n_vdd_087_1= importdata('m_da_3200_a_3299_vdd_087_a_1_passo_001.csv');
n_vdd_087_1 = freq_n_vdd_087_1.data(:,2);

freq_n_vdd_088_1= importdata('m_da_3200_a_3299_vdd_088_a_1_passo_001.csv');
n_vdd_088_1 = freq_n_vdd_088_1.data(:,2);

freq_n_vdd_089_1= importdata('m_da_3200_a_3299_vdd_089_a_1_passo_001.csv');
n_vdd_089_1 = freq_n_vdd_089_1.data(:,2);

freq_n_vdd_090_1= importdata('m_da_3200_a_3299_vdd_090_a_1_passo_001.csv');
n_vdd_090_1 = freq_n_vdd_090_1.data(:,2);

freq_n_vdd_091_1= importdata('m_da_3200_a_3299_vdd_091_a_1_passo_001.csv');
n_vdd_091_1 = freq_n_vdd_091_1.data(:,2);

freq_n_vdd_092_1= importdata('m_da_3200_a_3299_vdd_092_a_1_passo_001.csv');
n_vdd_092_1 = freq_n_vdd_092_1.data(:,2);

freq_n_vdd_093_1= importdata('m_da_3200_a_3299_vdd_093_a_1_passo_001.csv');
```



```
n_vdd_093_1 = freq_n_vdd_093_1.data(:,2);

freq_n_vdd_094_1= importdata('m_da_3200_a_3299_vdd_094_a_1_passo_001.csv');
n_vdd_094_1 = freq_n_vdd_094_1.data(:,2);

freq_n_vdd_095_1= importdata('m_da_3200_a_3299_vdd_095_a_1_passo_001.csv');
n_vdd_095_1 = freq_n_vdd_095_1.data(:,2);

freq_n_vdd_096_1= importdata('m_da_3200_a_3299_vdd_096_a_1_passo_001.csv');
n_vdd_096_1 = freq_n_vdd_096_1.data(:,2);

freq_n_vdd_097_1= importdata('m_da_3200_a_3299_vdd_097_a_1_passo_001.csv');
n_vdd_097_1 = freq_n_vdd_097_1.data(:,2);

freq_n_vdd_098_1= importdata('m_da_3200_a_3299_vdd_098_a_1_passo_001.csv');
n_vdd_098_1 = freq_n_vdd_098_1.data(:,2);

freq_n_vdd_099_1= importdata('m_da_3200_a_3299_vdd_099_a_1_passo_001.csv');
n_vdd_099_1 = freq_n_vdd_099_1.data(:,2);

freq_n_vdd_01_1= importdata('m_da_3200_a_3299_vdd_01_a_1_passo_001.csv');
n_vdd_01_1 = freq_n_vdd_01_1.data(:,2);


freq_n_vdd_050_2= importdata('m_da_3300_a_3499_vdd_050_a_1_passo_001.csv');
n_vdd_050_2 = freq_n_vdd_050_2.data(:,2);

freq_n_vdd_051_2= importdata('m_da_3300_a_3499_vdd_051_a_1_passo_001.csv');
n_vdd_051_2 = freq_n_vdd_051_2.data(:,2);

freq_n_vdd_052_2= importdata('m_da_3300_a_3499_vdd_052_a_1_passo_001.csv');
n_vdd_052_2 = freq_n_vdd_052_2.data(:,2);

freq_n_vdd_053_2= importdata('m_da_3300_a_3499_vdd_053_a_1_passo_001.csv');
n_vdd_053_2 = freq_n_vdd_053_2.data(:,2);

freq_n_vdd_054_2= importdata('m_da_3300_a_3499_vdd_054_a_1_passo_001.csv');
n_vdd_054_2 = freq_n_vdd_054_2.data(:,2);

freq_n_vdd_055_2= importdata('m_da_3300_a_3499_vdd_055_a_1_passo_001.csv');
n_vdd_055_2 = freq_n_vdd_055_2.data(:,2);

freq_n_vdd_056_2= importdata('m_da_3300_a_3499_vdd_056_a_1_passo_001.csv');
n_vdd_056_2 = freq_n_vdd_056_2.data(:,2);

freq_n_vdd_057_2= importdata('m_da_3300_a_3499_vdd_057_a_1_passo_001.csv');
n_vdd_057_2 = freq_n_vdd_057_2.data(:,2);

freq_n_vdd_058_2= importdata('m_da_3300_a_3499_vdd_058_a_1_passo_001.csv');
```

```
n_vdd_058_2 = freq_n_vdd_058_2.data(:,2);

freq_n_vdd_059_2= importdata('m_da_3300_a_3499_vdd_059_a_1_passo_001.csv');
n_vdd_059_2 = freq_n_vdd_059_2.data(:,2);

freq_n_vdd_060_2= importdata('m_da_3300_a_3499_vdd_060_a_1_passo_001.csv');
n_vdd_060_2 = freq_n_vdd_060_2.data(:,2);

freq_n_vdd_061_2= importdata('m_da_3300_a_3499_vdd_061_a_1_passo_001.csv');
n_vdd_061_2 = freq_n_vdd_061_2.data(:,2);

freq_n_vdd_062_2= importdata('m_da_3300_a_3499_vdd_062_a_1_passo_001.csv');
n_vdd_062_2 = freq_n_vdd_062_2.data(:,2);

freq_n_vdd_063_2= importdata('m_da_3300_a_3499_vdd_063_a_1_passo_001.csv');
n_vdd_063_2 = freq_n_vdd_063_2.data(:,2);

freq_n_vdd_064_2= importdata('m_da_3300_a_3499_vdd_064_a_1_passo_001.csv');
n_vdd_064_2 = freq_n_vdd_064_2.data(:,2);

freq_n_vdd_065_2= importdata('m_da_3300_a_3499_vdd_065_a_1_passo_001.csv');
n_vdd_065_2 = freq_n_vdd_065_2.data(:,2);

freq_n_vdd_066_2= importdata('m_da_3300_a_3499_vdd_066_a_1_passo_001.csv');
n_vdd_066_2 = freq_n_vdd_066_2.data(:,2);

freq_n_vdd_067_2= importdata('m_da_3300_a_3499_vdd_067_a_1_passo_001.csv');
n_vdd_067_2 = freq_n_vdd_067_2.data(:,2);

freq_n_vdd_068_2= importdata('m_da_3300_a_3499_vdd_068_a_1_passo_001.csv');
n_vdd_068_2 = freq_n_vdd_068_2.data(:,2);

freq_n_vdd_069_2= importdata('m_da_3300_a_3499_vdd_069_a_1_passo_001.csv');
n_vdd_069_2 = freq_n_vdd_069_2.data(:,2);

freq_n_vdd_070_2= importdata('m_da_3300_a_3499_vdd_070_a_1_passo_001.csv');
n_vdd_070_2 = freq_n_vdd_070_2.data(:,2);

freq_n_vdd_071_2= importdata('m_da_3300_a_3499_vdd_071_a_1_passo_001.csv');
n_vdd_071_2 = freq_n_vdd_071_2.data(:,2);

freq_n_vdd_072_2= importdata('m_da_3300_a_3499_vdd_072_a_1_passo_001.csv');
n_vdd_072_2 = freq_n_vdd_072_2.data(:,2);

freq_n_vdd_073_2= importdata('m_da_3300_a_3499_vdd_073_a_1_passo_001.csv');
n_vdd_073_2 = freq_n_vdd_073_2.data(:,2);

freq_n_vdd_074_2= importdata('m_da_3300_a_3499_vdd_074_a_1_passo_001.csv');
n_vdd_074_2 = freq_n_vdd_074_2.data(:,2);

freq_n_vdd_075_2= importdata('m_da_3300_a_3499_vdd_075_a_1_passo_001.csv');
n_vdd_075_2 = freq_n_vdd_075_2.data(:,2);

freq_n_vdd_076_2= importdata('m_da_3300_a_3499_vdd_076_a_1_passo_001.csv');
n_vdd_076_2 = freq_n_vdd_076_2.data(:,2);
```

```
freq_n_vdd_077_2= importdata('m_da_3300_a_3499_vdd_077_a_1_passo_001.csv');
n_vdd_077_2 = freq_n_vdd_077_2.data(:,2);

freq_n_vdd_078_2= importdata('m_da_3300_a_3499_vdd_078_a_1_passo_001.csv');
n_vdd_078_2 = freq_n_vdd_078_2.data(:,2);

freq_n_vdd_079_2= importdata('m_da_3300_a_3499_vdd_079_a_1_passo_001.csv');
n_vdd_079_2 = freq_n_vdd_079_2.data(:,2);

freq_n_vdd_080_2= importdata('m_da_3300_a_3499_vdd_080_a_1_passo_001.csv');
n_vdd_080_2 = freq_n_vdd_080_2.data(:,2);

freq_n_vdd_081_2= importdata('m_da_3300_a_3499_vdd_081_a_1_passo_001.csv');
n_vdd_081_2 = freq_n_vdd_081_2.data(:,2);

freq_n_vdd_082_2= importdata('m_da_3300_a_3499_vdd_082_a_1_passo_001.csv');
n_vdd_082_2 = freq_n_vdd_082_2.data(:,2);

freq_n_vdd_083_2= importdata('m_da_3300_a_3499_vdd_083_a_1_passo_001.csv');
n_vdd_083_2 = freq_n_vdd_083_2.data(:,2);

freq_n_vdd_084_2= importdata('m_da_3300_a_3499_vdd_084_a_1_passo_001.csv');
n_vdd_084_2 = freq_n_vdd_084_2.data(:,2);

freq_n_vdd_085_2= importdata('m_da_3300_a_3499_vdd_085_a_1_passo_001.csv');
n_vdd_085_2 = freq_n_vdd_085_2.data(:,2);

freq_n_vdd_086_2= importdata('m_da_3300_a_3499_vdd_086_a_1_passo_001.csv');
n_vdd_086_2 = freq_n_vdd_086_2.data(:,2);

freq_n_vdd_087_2= importdata('m_da_3300_a_3499_vdd_087_a_1_passo_001.csv');
n_vdd_087_2 = freq_n_vdd_087_2.data(:,2);

freq_n_vdd_088_2= importdata('m_da_3300_a_3499_vdd_088_a_1_passo_001.csv');
n_vdd_088_2 = freq_n_vdd_088_2.data(:,2);

freq_n_vdd_089_2= importdata('m_da_3300_a_3499_vdd_089_a_1_passo_001.csv');
n_vdd_089_2 = freq_n_vdd_089_2.data(:,2);

freq_n_vdd_090_2= importdata('m_da_3300_a_3499_vdd_090_a_1_passo_001.csv');
n_vdd_090_2 = freq_n_vdd_090_2.data(:,2);

freq_n_vdd_091_2= importdata('m_da_3300_a_3499_vdd_091_a_1_passo_001.csv');
n_vdd_091_2 = freq_n_vdd_091_2.data(:,2);

freq_n_vdd_092_2= importdata('m_da_3300_a_3499_vdd_092_a_1_passo_001.csv');
n_vdd_092_2 = freq_n_vdd_092_2.data(:,2);

freq_n_vdd_093_2= importdata('m_da_3300_a_3499_vdd_093_a_1_passo_001.csv');
n_vdd_093_2 = freq_n_vdd_093_2.data(:,2);

freq_n_vdd_094_2= importdata('m_da_3300_a_3499_vdd_094_a_1_passo_001.csv');
n_vdd_094_2 = freq_n_vdd_094_2.data(:,2);
```

```
freq_n_vdd_095_2= importdata('m_da_3300_a_3499_vdd_095_a_1_passo_001.csv');
n_vdd_095_2 = freq_n_vdd_095_2.data(:,2);

freq_n_vdd_096_2= importdata('m_da_3300_a_3499_vdd_096_a_1_passo_001.csv');
n_vdd_096_2 = freq_n_vdd_096_2.data(:,2);

freq_n_vdd_097_2= importdata('m_da_3300_a_3499_vdd_097_a_1_passo_001.csv');
n_vdd_097_2 = freq_n_vdd_097_2.data(:,2);

freq_n_vdd_098_2= importdata('m_da_3300_a_3499_vdd_098_a_1_passo_001.csv');
n_vdd_098_2 = freq_n_vdd_098_2.data(:,2);

freq_n_vdd_099_2= importdata('m_da_3300_a_3499_vdd_099_a_1_passo_001.csv');
n_vdd_099_2 = freq_n_vdd_099_2.data(:,2);

freq_n_vdd_01_2= importdata('m_da_3300_a_3499_vdd_01_a_1_passo_001.csv');
n_vdd_01_2 = freq_n_vdd_01_2.data(:,2);


freq_n_vdd_050_3= importdata('m_da_3500_a_3699_vdd_050_a_1_passo_001.csv');
n_vdd_050_3 = freq_n_vdd_050_3.data(:,2);

freq_n_vdd_051_3= importdata('m_da_3500_a_3699_vdd_051_a_1_passo_001.csv');
n_vdd_051_3 = freq_n_vdd_051_3.data(:,2);

freq_n_vdd_052_3= importdata('m_da_3500_a_3699_vdd_052_a_1_passo_001.csv');
n_vdd_052_3 = freq_n_vdd_052_3.data(:,2);

freq_n_vdd_053_3= importdata('m_da_3500_a_3699_vdd_053_a_1_passo_001.csv');
n_vdd_053_3 = freq_n_vdd_053_3.data(:,2);

freq_n_vdd_054_3= importdata('m_da_3500_a_3699_vdd_054_a_1_passo_001.csv');
n_vdd_054_3 = freq_n_vdd_054_3.data(:,2);

freq_n_vdd_055_3= importdata('m_da_3500_a_3699_vdd_055_a_1_passo_001.csv');
n_vdd_055_3 = freq_n_vdd_055_3.data(:,2);

freq_n_vdd_056_3= importdata('m_da_3500_a_3699_vdd_056_a_1_passo_001.csv');
n_vdd_056_3 = freq_n_vdd_056_3.data(:,2);

freq_n_vdd_057_3= importdata('m_da_3500_a_3699_vdd_057_a_1_passo_001.csv');
n_vdd_057_3 = freq_n_vdd_057_3.data(:,2);

freq_n_vdd_058_3= importdata('m_da_3500_a_3699_vdd_058_a_1_passo_001.csv');
n_vdd_058_3 = freq_n_vdd_058_3.data(:,2);

freq_n_vdd_059_3= importdata('m_da_3500_a_3699_vdd_059_a_1_passo_001.csv');
n_vdd_059_3 = freq_n_vdd_059_3.data(:,2);

freq_n_vdd_060_3= importdata('m_da_3500_a_3699_vdd_060_a_1_passo_001.csv');
```

```
n_vdd_060_3 = freq_n_vdd_060_3.data(:,2);

freq_n_vdd_061_3= importdata('m_da_3500_a_3699_vdd_061_a_1_passo_001.csv');
n_vdd_061_3 = freq_n_vdd_061_3.data(:,2);

freq_n_vdd_062_3= importdata('m_da_3500_a_3699_vdd_062_a_1_passo_001.csv');
n_vdd_062_3 = freq_n_vdd_062_3.data(:,2);

freq_n_vdd_063_3= importdata('m_da_3500_a_3699_vdd_063_a_1_passo_001.csv');
n_vdd_063_3 = freq_n_vdd_063_3.data(:,2);

freq_n_vdd_064_3= importdata('m_da_3500_a_3699_vdd_064_a_1_passo_001.csv');
n_vdd_064_3 = freq_n_vdd_064_3.data(:,2);

freq_n_vdd_065_3= importdata('m_da_3500_a_3699_vdd_065_a_1_passo_001.csv');
n_vdd_065_3 = freq_n_vdd_065_3.data(:,2);

freq_n_vdd_066_3= importdata('m_da_3500_a_3699_vdd_066_a_1_passo_001.csv');
n_vdd_066_3 = freq_n_vdd_066_3.data(:,2);

freq_n_vdd_067_3= importdata('m_da_3500_a_3699_vdd_067_a_1_passo_001.csv');
n_vdd_067_3 = freq_n_vdd_067_3.data(:,2);

freq_n_vdd_068_3= importdata('m_da_3500_a_3699_vdd_068_a_1_passo_001.csv');
n_vdd_068_3 = freq_n_vdd_068_3.data(:,2);

freq_n_vdd_069_3= importdata('m_da_3500_a_3699_vdd_069_a_1_passo_001.csv');
n_vdd_069_3 = freq_n_vdd_069_3.data(:,2);

freq_n_vdd_070_3= importdata('m_da_3500_a_3699_vdd_070_a_1_passo_001.csv');
n_vdd_070_3 = freq_n_vdd_070_3.data(:,2);

freq_n_vdd_071_3= importdata('m_da_3500_a_3699_vdd_071_a_1_passo_001.csv');
n_vdd_071_3 = freq_n_vdd_071_3.data(:,2);

freq_n_vdd_072_3= importdata('m_da_3500_a_3699_vdd_072_a_1_passo_001.csv');
n_vdd_072_3 = freq_n_vdd_072_3.data(:,2);

freq_n_vdd_073_3= importdata('m_da_3500_a_3699_vdd_073_a_1_passo_001.csv');
n_vdd_073_3 = freq_n_vdd_073_3.data(:,2);

freq_n_vdd_074_3= importdata('m_da_3500_a_3699_vdd_074_a_1_passo_001.csv');
n_vdd_074_3 = freq_n_vdd_074_3.data(:,2);

freq_n_vdd_075_3= importdata('m_da_3500_a_3699_vdd_075_a_1_passo_001.csv');
n_vdd_075_3 = freq_n_vdd_075_3.data(:,2);

freq_n_vdd_076_3= importdata('m_da_3500_a_3699_vdd_076_a_1_passo_001.csv');
n_vdd_076_3 = freq_n_vdd_076_3.data(:,2);

freq_n_vdd_077_3= importdata('m_da_3500_a_3699_vdd_077_a_1_passo_001.csv');
n_vdd_077_3 = freq_n_vdd_077_3.data(:,2);

freq_n_vdd_078_3= importdata('m_da_3500_a_3699_vdd_078_a_1_passo_001.csv');
n_vdd_078_3 = freq_n_vdd_078_3.data(:,2);
```

```
freq_n_vdd_079_3= importdata('m_da_3500_a_3699_vdd_079_a_1_passo_001.csv');
n_vdd_079_3 = freq_n_vdd_079_3.data(:,2);

freq_n_vdd_080_3= importdata('m_da_3500_a_3699_vdd_080_a_1_passo_001.csv');
n_vdd_080_3 = freq_n_vdd_080_3.data(:,2);

freq_n_vdd_081_3= importdata('m_da_3500_a_3699_vdd_081_a_1_passo_001.csv');
n_vdd_081_3 = freq_n_vdd_081_3.data(:,2);

freq_n_vdd_082_3= importdata('m_da_3500_a_3699_vdd_082_a_1_passo_001.csv');
n_vdd_082_3 = freq_n_vdd_082_3.data(:,2);

freq_n_vdd_083_3= importdata('m_da_3500_a_3699_vdd_083_a_1_passo_001.csv');
n_vdd_083_3 = freq_n_vdd_083_3.data(:,2);

freq_n_vdd_084_3= importdata('m_da_3500_a_3699_vdd_084_a_1_passo_001.csv');
n_vdd_084_3 = freq_n_vdd_084_3.data(:,2);

freq_n_vdd_085_3= importdata('m_da_3500_a_3699_vdd_085_a_1_passo_001.csv');
n_vdd_085_3 = freq_n_vdd_085_3.data(:,2);

freq_n_vdd_086_3= importdata('m_da_3500_a_3699_vdd_086_a_1_passo_001.csv');
n_vdd_086_3 = freq_n_vdd_086_3.data(:,2);

freq_n_vdd_087_3= importdata('m_da_3500_a_3699_vdd_087_a_1_passo_001.csv');
n_vdd_087_3 = freq_n_vdd_087_3.data(:,2);

freq_n_vdd_088_3= importdata('m_da_3500_a_3699_vdd_088_a_1_passo_001.csv');
n_vdd_088_3 = freq_n_vdd_088_3.data(:,2);

freq_n_vdd_089_3= importdata('m_da_3500_a_3699_vdd_089_a_1_passo_001.csv');
n_vdd_089_3 = freq_n_vdd_089_3.data(:,2);

freq_n_vdd_090_3= importdata('m_da_3500_a_3699_vdd_090_a_1_passo_001.csv');
n_vdd_090_3 = freq_n_vdd_090_3.data(:,2);

freq_n_vdd_091_3= importdata('m_da_3500_a_3699_vdd_091_a_1_passo_001.csv');
n_vdd_091_3 = freq_n_vdd_091_3.data(:,2);

freq_n_vdd_092_3= importdata('m_da_3500_a_3699_vdd_092_a_1_passo_001.csv');
n_vdd_092_3 = freq_n_vdd_092_3.data(:,2);

freq_n_vdd_093_3= importdata('m_da_3500_a_3699_vdd_093_a_1_passo_001.csv');
n_vdd_093_3 = freq_n_vdd_093_3.data(:,2);

freq_n_vdd_094_3= importdata('m_da_3500_a_3699_vdd_094_a_1_passo_001.csv');
n_vdd_094_3 = freq_n_vdd_094_3.data(:,2);

freq_n_vdd_095_3= importdata('m_da_3500_a_3699_vdd_095_a_1_passo_001.csv');
n_vdd_095_3 = freq_n_vdd_095_3.data(:,2);

freq_n_vdd_096_3= importdata('m_da_3500_a_3699_vdd_096_a_1_passo_001.csv');
n_vdd_096_3 = freq_n_vdd_096_3.data(:,2);
```

```

freq_n_vdd_097_3= importdata('m_da_3500_a_3699_vdd_097_a_1_passo_001.csv');
n_vdd_097_3 = freq_n_vdd_097_3.data(:,2);

freq_n_vdd_098_3= importdata('m_da_3500_a_3699_vdd_098_a_1_passo_001.csv');
n_vdd_098_3 = freq_n_vdd_098_3.data(:,2);

freq_n_vdd_099_3= importdata('m_da_3500_a_3699_vdd_099_a_1_passo_001.csv');
n_vdd_099_3 = freq_n_vdd_099_3.data(:,2);

freq_n_vdd_01_3= importdata('m_da_3500_a_3699_vdd_01_a_1_passo_001.csv');
n_vdd_01_3 = freq_n_vdd_01_3.data(:,2);

% NON VENGONO RIPORTATI TUTTE LE RIGHE DURANTE LA FASE DI IMPORTAZIONE
% DEI circa 1600 VETTORI CONTENENTI I DATI DELLE SIMULAZIONI SU CADENCE

freq_n_vdd_050_30= importdata('m_da_8051_a_8192_vdd_050_a_1_passo_001.csv');
n_vdd_050_30 = freq_n_vdd_050_30.data(:,2);

freq_n_vdd_051_30= importdata('m_da_8051_a_8192_vdd_051_a_1_passo_001.csv');
n_vdd_051_30 = freq_n_vdd_051_30.data(:,2);

freq_n_vdd_052_30= importdata('m_da_8051_a_8192_vdd_052_a_1_passo_001.csv');
n_vdd_052_30 = freq_n_vdd_052_30.data(:,2);

freq_n_vdd_053_30= importdata('m_da_8051_a_8192_vdd_053_a_1_passo_001.csv');
n_vdd_053_30 = freq_n_vdd_053_30.data(:,2);

freq_n_vdd_054_30= importdata('m_da_8051_a_8192_vdd_054_a_1_passo_001.csv');
n_vdd_054_30 = freq_n_vdd_054_30.data(:,2);

freq_n_vdd_055_30= importdata('m_da_8051_a_8192_vdd_055_a_1_passo_001.csv');
n_vdd_055_30 = freq_n_vdd_055_30.data(:,2);

freq_n_vdd_056_30= importdata('m_da_8051_a_8192_vdd_056_a_1_passo_001.csv');
n_vdd_056_30 = freq_n_vdd_056_30.data(:,2);

freq_n_vdd_057_30= importdata('m_da_8051_a_8192_vdd_057_a_1_passo_001.csv');
n_vdd_057_30 = freq_n_vdd_057_30.data(:,2);

freq_n_vdd_058_30= importdata('m_da_8051_a_8192_vdd_058_a_1_passo_001.csv');
n_vdd_058_30 = freq_n_vdd_058_30.data(:,2);

freq_n_vdd_059_30= importdata('m_da_8051_a_8192_vdd_059_a_1_passo_001.csv');
n_vdd_059_30 = freq_n_vdd_059_30.data(:,2);

freq_n_vdd_060_30= importdata('m_da_8051_a_8192_vdd_060_a_1_passo_001.csv');
n_vdd_060_30 = freq_n_vdd_060_30.data(:,2);

freq_n_vdd_061_30= importdata('m_da_8051_a_8192_vdd_061_a_1_passo_001.csv');
n_vdd_061_30 = freq_n_vdd_061_30.data(:,2);

freq_n_vdd_062_30= importdata('m_da_8051_a_8192_vdd_062_a_1_passo_001.csv');
n_vdd_062_30 = freq_n_vdd_062_30.data(:,2);

```

```
freq_n_vdd_063_30= importdata('m_da_8051_a_8192_vdd_063_a_1_passo_001.csv');
n_vdd_063_30 = freq_n_vdd_063_30.data(:,2);

freq_n_vdd_064_30= importdata('m_da_8051_a_8192_vdd_064_a_1_passo_001.csv');
n_vdd_064_30 = freq_n_vdd_064_30.data(:,2);

freq_n_vdd_065_30= importdata('m_da_8051_a_8192_vdd_065_a_1_passo_001.csv');
n_vdd_065_30 = freq_n_vdd_065_30.data(:,2);

freq_n_vdd_066_30= importdata('m_da_8051_a_8192_vdd_066_a_1_passo_001.csv');
n_vdd_066_30 = freq_n_vdd_066_30.data(:,2);

freq_n_vdd_067_30= importdata('m_da_8051_a_8192_vdd_067_a_1_passo_001.csv');
n_vdd_067_30 = freq_n_vdd_067_30.data(:,2);

freq_n_vdd_068_30= importdata('m_da_8051_a_8192_vdd_068_a_1_passo_001.csv');
n_vdd_068_30 = freq_n_vdd_068_30.data(:,2);

freq_n_vdd_069_30= importdata('m_da_8051_a_8192_vdd_069_a_1_passo_001.csv');
n_vdd_069_30 = freq_n_vdd_069_30.data(:,2);

freq_n_vdd_070_30= importdata('m_da_8051_a_8192_vdd_070_a_1_passo_001.csv');
n_vdd_070_30 = freq_n_vdd_070_30.data(:,2);

freq_n_vdd_071_30= importdata('m_da_8051_a_8192_vdd_071_a_1_passo_001.csv');
n_vdd_071_30 = freq_n_vdd_071_30.data(:,2);

freq_n_vdd_072_30= importdata('m_da_8051_a_8192_vdd_072_a_1_passo_001.csv');
n_vdd_072_30 = freq_n_vdd_072_30.data(:,2);

freq_n_vdd_073_30= importdata('m_da_8051_a_8192_vdd_073_a_1_passo_001.csv');
n_vdd_073_30 = freq_n_vdd_073_30.data(:,2);

freq_n_vdd_074_30= importdata('m_da_8051_a_8192_vdd_074_a_1_passo_001.csv');
n_vdd_074_30 = freq_n_vdd_074_30.data(:,2);

freq_n_vdd_075_30= importdata('m_da_8051_a_8192_vdd_075_a_1_passo_001.csv');
n_vdd_075_30 = freq_n_vdd_075_30.data(:,2);

freq_n_vdd_076_30= importdata('m_da_8051_a_8192_vdd_076_a_1_passo_001.csv');
n_vdd_076_30 = freq_n_vdd_076_30.data(:,2);

freq_n_vdd_077_30= importdata('m_da_8051_a_8192_vdd_077_a_1_passo_001.csv');
n_vdd_077_30 = freq_n_vdd_077_30.data(:,2);

freq_n_vdd_078_30= importdata('m_da_8051_a_8192_vdd_078_a_1_passo_001.csv');
n_vdd_078_30 = freq_n_vdd_078_30.data(:,2);

freq_n_vdd_079_30= importdata('m_da_8051_a_8192_vdd_079_a_1_passo_001.csv');
n_vdd_079_30 = freq_n_vdd_079_30.data(:,2);

freq_n_vdd_080_30= importdata('m_da_8051_a_8192_vdd_080_a_1_passo_001.csv');
n_vdd_080_30 = freq_n_vdd_080_30.data(:,2);

freq_n_vdd_081_30= importdata('m_da_8051_a_8192_vdd_081_a_1_passo_001.csv');
```



```
n_vdd_081_30 = freq_n_vdd_081_30.data(:,2);

freq_n_vdd_082_30= importdata('m_da_8051_a_8192_vdd_082_a_1_passo_001.csv');
n_vdd_082_30 = freq_n_vdd_082_30.data(:,2);

freq_n_vdd_083_30= importdata('m_da_8051_a_8192_vdd_083_a_1_passo_001.csv');
n_vdd_083_30 = freq_n_vdd_083_30.data(:,2);

freq_n_vdd_084_30= importdata('m_da_8051_a_8192_vdd_084_a_1_passo_001.csv');
n_vdd_084_30 = freq_n_vdd_084_30.data(:,2);

freq_n_vdd_085_30= importdata('m_da_8051_a_8192_vdd_085_a_1_passo_001.csv');
n_vdd_085_30 = freq_n_vdd_085_30.data(:,2);

freq_n_vdd_086_30= importdata('m_da_8051_a_8192_vdd_086_a_1_passo_001.csv');
n_vdd_086_30 = freq_n_vdd_086_30.data(:,2);

freq_n_vdd_087_30= importdata('m_da_8051_a_8192_vdd_087_a_1_passo_001.csv');
n_vdd_087_30 = freq_n_vdd_087_30.data(:,2);

freq_n_vdd_088_30= importdata('m_da_8051_a_8192_vdd_088_a_1_passo_001.csv');
n_vdd_088_30 = freq_n_vdd_088_30.data(:,2);

freq_n_vdd_089_30= importdata('m_da_8051_a_8192_vdd_089_a_1_passo_001.csv');
n_vdd_089_30 = freq_n_vdd_089_30.data(:,2);

freq_n_vdd_090_30= importdata('m_da_8051_a_8192_vdd_090_a_1_passo_001.csv');
n_vdd_090_30 = freq_n_vdd_090_30.data(:,2);

freq_n_vdd_091_30= importdata('m_da_8051_a_8192_vdd_091_a_1_passo_001.csv');
n_vdd_091_30 = freq_n_vdd_091_30.data(:,2);

freq_n_vdd_092_30= importdata('m_da_8051_a_8192_vdd_092_a_1_passo_001.csv');
n_vdd_092_30 = freq_n_vdd_092_30.data(:,2);

freq_n_vdd_093_30= importdata('m_da_8051_a_8192_vdd_093_a_1_passo_001.csv');
n_vdd_093_30 = freq_n_vdd_093_30.data(:,2);

freq_n_vdd_094_30= importdata('m_da_8051_a_8192_vdd_094_a_1_passo_001.csv');
n_vdd_094_30 = freq_n_vdd_094_30.data(:,2);

freq_n_vdd_095_30= importdata('m_da_8051_a_8192_vdd_095_a_1_passo_001.csv');
n_vdd_095_30 = freq_n_vdd_095_30.data(:,2);

freq_n_vdd_096_30= importdata('m_da_8051_a_8192_vdd_096_a_1_passo_001.csv');
n_vdd_096_30 = freq_n_vdd_096_30.data(:,2);

freq_n_vdd_097_30= importdata('m_da_8051_a_8192_vdd_097_a_1_passo_001.csv');
n_vdd_097_30 = freq_n_vdd_097_30.data(:,2);

freq_n_vdd_098_30= importdata('m_da_8051_a_8192_vdd_098_a_1_passo_001.csv');
n_vdd_098_30 = freq_n_vdd_098_30.data(:,2);

freq_n_vdd_099_30= importdata('m_da_8051_a_8192_vdd_099_a_1_passo_001.csv');
n_vdd_099_30 = freq_n_vdd_099_30.data(:,2);
```

```
freq_n_vdd_01_30= importdata('m_da_8051_a_8192_vdd_01_a_1_passo_001.csv');
n_vdd_01_30 = freq_n_vdd_01_30.data(:,2);
```

```
% FINE IMPORTAZIONE DEI DATI
```

```
% matrice contiene tutte le frequenze ottenute dalle simulazioni
matrice= zeros(length(m),length(vdd));
```

```
% % concatenano tutti i vettori colonna importati
```

```
n_vdd_050_mix= cat(1,n_vdd_050_1,n_vdd_050_2, n_vdd_050_3,... ,n_vdd_050_30);
```

```
n_vdd_051_mix= cat(1,n_vdd_051_1,n_vdd_051_2 , n_vdd_051_3, ... , n_vdd_051_30);
n_vdd_052_mix= cat(1,n_vdd_052_1,n_vdd_052_2, n_vdd_052_3, ... , n_vdd_052_30);
n_vdd_053_mix= cat(1,n_vdd_053_1,n_vdd_053_2, n_vdd_053_3, ... , n_vdd_053_30);
n_vdd_054_mix= cat(1,n_vdd_054_1,n_vdd_054_2, n_vdd_054_3, ... , n_vdd_054_30);
n_vdd_055_mix= cat(1,n_vdd_055_1,n_vdd_055_2 , n_vdd_055_3, ... , n_vdd_055_30);
n_vdd_056_mix= cat(1,n_vdd_056_1,n_vdd_056_2, n_vdd_056_3, ... , n_vdd_056_30);
n_vdd_057_mix= cat(1,n_vdd_057_1,n_vdd_057_2 , n_vdd_057_3, ... , n_vdd_057_30);
n_vdd_058_mix= cat(1,n_vdd_058_1,n_vdd_058_2 , n_vdd_058_3, ... , n_vdd_058_30);
n_vdd_059_mix= cat(1,n_vdd_059_1,n_vdd_059_2 , n_vdd_059_3, ... , n_vdd_059_30);
n_vdd_060_mix= cat(1,n_vdd_060_1,n_vdd_060_2 , n_vdd_060_3, ... , n_vdd_060_30);
n_vdd_061_mix= cat(1,n_vdd_061_1,n_vdd_061_2 , n_vdd_061_3, ... , n_vdd_061_30);
n_vdd_062_mix= cat(1,n_vdd_062_1,n_vdd_062_2 , n_vdd_062_3, ... , n_vdd_062_30);
n_vdd_063_mix= cat(1,n_vdd_063_1,n_vdd_063_2 , n_vdd_063_3, ... , n_vdd_063_30);
n_vdd_064_mix= cat(1,n_vdd_064_1,n_vdd_064_2 , n_vdd_064_3, ... , n_vdd_064_30);
n_vdd_065_mix= cat(1,n_vdd_065_1,n_vdd_065_2 , n_vdd_065_3, ... , n_vdd_065_30);
n_vdd_066_mix= cat(1,n_vdd_066_1,n_vdd_066_2 , n_vdd_066_3, ... , n_vdd_066_30);
n_vdd_067_mix= cat(1,n_vdd_067_1,n_vdd_067_2 , n_vdd_067_3, ... , n_vdd_067_30);
n_vdd_068_mix= cat(1,n_vdd_068_1,n_vdd_068_2 , n_vdd_068_3, ... , n_vdd_068_30);
n_vdd_069_mix= cat(1,n_vdd_069_1,n_vdd_069_2 , n_vdd_069_3, ... , n_vdd_069_30);
n_vdd_070_mix= cat(1,n_vdd_070_1,n_vdd_070_2 , n_vdd_070_3, ... , n_vdd_070_30);
n_vdd_071_mix= cat(1,n_vdd_071_1,n_vdd_071_2 , n_vdd_071_3, ... , n_vdd_071_30);
n_vdd_072_mix= cat(1,n_vdd_072_1,n_vdd_072_2 , n_vdd_072_3, ... , n_vdd_072_30);
n_vdd_073_mix= cat(1,n_vdd_073_1,n_vdd_073_2 , n_vdd_073_3, ... , n_vdd_073_30);
n_vdd_074_mix= cat(1,n_vdd_074_1,n_vdd_074_2 , n_vdd_074_3, ... , n_vdd_074_30);
n_vdd_075_mix= cat(1,n_vdd_075_1,n_vdd_075_2 , n_vdd_075_3, ... , n_vdd_075_30);
n_vdd_076_mix= cat(1,n_vdd_076_1,n_vdd_076_2 , n_vdd_076_3, ... , n_vdd_076_30);
n_vdd_077_mix= cat(1,n_vdd_077_1,n_vdd_077_2 , n_vdd_077_3, ... , n_vdd_077_30);
n_vdd_078_mix= cat(1,n_vdd_078_1,n_vdd_078_2 , n_vdd_078_3, ... , n_vdd_078_30);
n_vdd_079_mix= cat(1,n_vdd_079_1,n_vdd_079_2 , n_vdd_079_3, ... , n_vdd_079_30);
n_vdd_080_mix= cat(1,n_vdd_080_1,n_vdd_080_2 , n_vdd_080_3, ... , n_vdd_080_30);
n_vdd_081_mix= cat(1,n_vdd_081_1,n_vdd_081_2 , n_vdd_081_3, ... , n_vdd_081_30);
n_vdd_082_mix= cat(1,n_vdd_082_1,n_vdd_082_2 , n_vdd_082_3, ... , n_vdd_082_30);
n_vdd_083_mix= cat(1,n_vdd_083_1,n_vdd_083_2 , n_vdd_083_3, ... , n_vdd_083_30);
n_vdd_084_mix= cat(1,n_vdd_084_1,n_vdd_084_2 , n_vdd_084_3, ... , n_vdd_084_30);
n_vdd_085_mix= cat(1,n_vdd_085_1,n_vdd_085_2 , n_vdd_085_3, ... , n_vdd_085_30);
n_vdd_086_mix= cat(1,n_vdd_086_1,n_vdd_086_2 , n_vdd_086_3, ... , n_vdd_086_30);
n_vdd_087_mix= cat(1,n_vdd_087_1,n_vdd_087_2 , n_vdd_087_3, ... , n_vdd_087_30);
n_vdd_088_mix= cat(1,n_vdd_088_1,n_vdd_088_2 , n_vdd_088_3, ... , n_vdd_088_30);
n_vdd_089_mix= cat(1,n_vdd_089_1,n_vdd_089_2 , n_vdd_089_3, ... , n_vdd_089_30);
n_vdd_090_mix= cat(1,n_vdd_090_1,n_vdd_090_2 , n_vdd_090_3, ... , n_vdd_090_30);
n_vdd_091_mix= cat(1,n_vdd_091_1,n_vdd_091_2 , n_vdd_091_3, ... , n_vdd_091_30);
n_vdd_092_mix= cat(1,n_vdd_092_1,n_vdd_092_2 , n_vdd_092_3, ... , n_vdd_092_30);
n_vdd_093_mix= cat(1,n_vdd_093_1,n_vdd_093_2 , n_vdd_093_3, ... , n_vdd_093_30);
```

```
n_vdd_094_mix= cat(1,n_vdd_094_1,n_vdd_094_2 , n_vdd_094_3, ... , n_vdd_094_30);
n_vdd_095_mix= cat(1,n_vdd_095_1,n_vdd_095_2, n_vdd_095_3, ... , n_vdd_095_30);
n_vdd_096_mix= cat(1,n_vdd_096_1,n_vdd_096_2 , n_vdd_096_3, ... , n_vdd_096_30);
n_vdd_097_mix= cat(1,n_vdd_097_1,n_vdd_097_2 , n_vdd_097_3, ... , n_vdd_097_30);
n_vdd_098_mix= cat(1,n_vdd_098_1,n_vdd_098_2 , n_vdd_098_3, ... , n_vdd_098_30);
n_vdd_099_mix= cat(1,n_vdd_099_1,n_vdd_099_2 , n_vdd_099_3, ... , n_vdd_099_30);
n_vdd_01_mix= cat(1,n_vdd_01_1,n_vdd_01_2 , n_vdd_01_3, ... , n_vdd_01_30);
```

```
% generazione della matrice
```

```
% ad esempio matrice(1,1) è quella frequenza relativa al numero 3200 con
```

```
% Vdd= 0.5V
```

```
% matrice(2,51) è quella frequenza relativa al numero 3201 con vdd=1V
```

```
matrice(:,1)=n_vdd_050_mix;
matrice(:,2)=n_vdd_051_mix;
matrice(:,3)=n_vdd_052_mix;
matrice(:,4)=n_vdd_053_mix;
matrice(:,5)=n_vdd_054_mix;
matrice(:,6)=n_vdd_055_mix;
matrice(:,7)=n_vdd_056_mix;
matrice(:,8)=n_vdd_057_mix;
matrice(:,9)=n_vdd_058_mix;
matrice(:,10)=n_vdd_059_mix;
matrice(:,11)=n_vdd_060_mix;
matrice(:,12)=n_vdd_061_mix;
matrice(:,13)=n_vdd_062_mix;
matrice(:,14)=n_vdd_063_mix;
matrice(:,15)=n_vdd_064_mix;
matrice(:,16)=n_vdd_065_mix;
matrice(:,17)=n_vdd_066_mix;
matrice(:,18)=n_vdd_067_mix;
matrice(:,19)=n_vdd_068_mix;
matrice(:,20)=n_vdd_069_mix;
matrice(:,21)=n_vdd_070_mix;
matrice(:,22)=n_vdd_071_mix;
matrice(:,23)=n_vdd_072_mix;
matrice(:,24)=n_vdd_073_mix;
matrice(:,25)=n_vdd_074_mix;
matrice(:,26)=n_vdd_075_mix;
matrice(:,27)=n_vdd_076_mix;
matrice(:,28)=n_vdd_077_mix;
matrice(:,29)=n_vdd_078_mix;
matrice(:,30)=n_vdd_079_mix;
matrice(:,31)=n_vdd_080_mix;
matrice(:,32)=n_vdd_081_mix;
matrice(:,33)=n_vdd_082_mix;
matrice(:,34)=n_vdd_083_mix;
matrice(:,35)=n_vdd_084_mix;
matrice(:,36)=n_vdd_085_mix;
matrice(:,37)=n_vdd_086_mix;
matrice(:,38)=n_vdd_087_mix;
matrice(:,39)=n_vdd_088_mix;
matrice(:,40)=n_vdd_089_mix;
matrice(:,41)=n_vdd_090_mix;
matrice(:,42)=n_vdd_091_mix;
```

```
matrice(:,43)=n_vdd_092_mix;
matrice(:,44)=n_vdd_093_mix;
matrice(:,45)=n_vdd_094_mix;
matrice(:,46)=n_vdd_095_mix;
matrice(:,47)=n_vdd_096_mix;
matrice(:,48)=n_vdd_097_mix;
matrice(:,49)=n_vdd_098_mix;
matrice(:,50)=n_vdd_099_mix;
matrice(:,51)=n_vdd_01_mix;

save matrice matrice

load matrice

k=1; % k=V_REF/V0

flag=1; % flag utile per capire quando passare al secondo giro dell'algoritmo

for i=1:length(vdd) % processo ripetuto per ogni vdd

    if flag==1 % vuol dire 'se sono al primo giro'

        % A= (2^N/(m1-m2))*log(n1/n2)
        A_init= (2^N/(m(3000)-m(4000)))*log(matrice(3000,i)/matrice(4000,i));

        r_1= (k/A_init)*2^N;

        %vengono definiti gli indici del vettore m per ciclo successivo
        j=100;
        p=405;
    end

    diff=1; % inizializzo r-r_old=1 per entrare nel ciclo while
    diff_old=100;

    while diff < diff_old % abs(r-r_old)<eps?

        m1=m(j);
        m2=m(p);

        A_2= ((2^N/(m1-m2))*log(matrice(j,i)/matrice(p,i))-0.1e-3*m2);
        A_2= A_2/(1-2e-4*A_2^2);
        r_2= (k/A_2)*2^N % calcolo del nuovo r
        diff_old=diff;
        diff= abs(r_1-r_2)
        r_1=r_2 % utile per giro successivo

        % f/(1-beta*f^2)

        k_1= .5;
        k_2= 1.8;
```

```
% definisco dei nuovi m1 ed m2 per il prossimo giro riferiti ad r_1
m1= round(k_1*r_1);
m2= round(k_2*r_1);

% i seguenti 2 cicli for servono per individuare in quale posizione
% si trovano m1 e m2 nel vettore m : per esempio se m1=4010
% quanto vale w di m(w)?
for ell=1:length(m)
    if m1==m(ell)
        j=ell;
    end
end

for ell=1:length(m)
    if m2==m(ell)
        p=ell;
    end
end

end
r_1

riferimento(i)=r_1; % salvo i diversi r_1 per ogni vdd
Vref(i)=r_1*vdd(i)/(2^N); %calcolo di Vref
    flag=0; % non sono più nel primo giro

end

plot(vdd, Vref, '*b')
xlabel('Vdd□[V]')
ylabel('r*Vdd/2^N□[V]')
grid on
figure(2)
plot(vdd, riferimento, '*r')
xlabel('Vdd□[V]')
ylabel('r')
grid on
line_sensitivity= ((max(Vref)-min(Vref))/mean(Vref))*100*2
```

Bibliografia

- [1] Massimo Alioto. *Enabling the Internet of Things: from Integrated Circuits to Integrated Systems*. Springer, 2017, pp.1-45.
- [2] S. Vashi et al. “Internet of Things (IoT): A vision, architectural elements, and security issues,” in: *International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, (pp 492-496).
- [3] P. Toledo et al. “Re-thinking Analog Integrated Circuits in Digital Terms: A New Design Concept for the IoT Era”. In: *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 3 (2021), 816–822, doi: 10.1109/TCSII.2021.3049680.
- [4] B. Xu et al. “A scaling compatible, synthesis friendly VCO-based delta-sigma ADC design and synthesis methodology,” in: *54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, (2017), pp. 1–6.
- [5] P.R. Kinget. “Scaling analog circuits into deep nanoscale CMOS: Obstacles and ways to overcome them,” in: *2015 IEEE Custom Integrated Circuits Conference (CICC)*, San Jose, CA, USA (2015), pp. 1–8.
- [6] G. G. E. Gielen, L. Hernandez e P. Rombouts. “Time-Encoding Analog-to-Digital Converters: Bridging the Analog Gap to Advanced Digital CMOS-Part 1: Basic Principles,” in: *IEEE Solid-State Circuits Magazine*, vol. 12, no. 2 (pp 47-55).
- [7] P. S. Crovetto, R. Rubino e F. Musolino. “Relaxation Digital-to-Analogue Converter,” in: *Electronics Letters*, vol. 55, no. 12 (2019), pp. 685–688.
- [8] M. Seok et al. “A Portable 2-Transistor Picowatt Temperature-Compensated Voltage Reference Operating at 0.5 V,” in: *IEEE Journal of Solid-State Circuits*, vol. 47, no. 10 (2012), pp. 2534–2545.
- [9] Domenico Albano et al. “A Sub-kT /q Voltage Reference Operating at 150 mV,” in: *IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS*, vol. 23, no. 8 (2015), pp. 1547–1551.
- [10] P. S. Crovetto. “A Digital-Based Virtual Voltage Reference,” in: *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 5 (2015), pp. 1315–1324.
- [11] P.S. Crovetto et al. “breaking the boundaries between analogue and digital,” in: *ELECTRONICS LETTERS*, (pp 672-673).
- [12] “<http://www.itrs2.net/2013-itrs.html>”. In: *Chapter-Design*, pp 42.
- [13] Behzad Razavi. *Design of Analog CMOS Integrated Circuits*, 2nd Edition: McGraw-Hill, pp. 509-538.

- [14] Phillip E. Allen e Douglas R. Holberg. *CMOS Analog Circuit Design*, 3rd Edition: Oxford University Press, pp. 147-172.
- [15] *Tips and tricks for designing with voltage references*, Texas Instruments, 2017, pp. 10-11.
- [16] Ned Mohan, Tore M. Undeland e William P. Robbins. *Elettronica di potenza: convertitori e applicazioni*, 3rd Edition: HOEPLI, pp. 1-2.
- [17] Willy M. C. Sansen. *Analog Design essentials*, Springer, pp. 457-458.
- [18] Chi Wah Kok e Wing Shan Tam. *CMOS Voltage References: An Analytical and Practical Perspective*, First Edition: IEEE, WILEY, pp. 49-62.
- [19] R. Jacob Baker. *CMOS Circuit Design, Layout, and Simulation*, 3rd edition: Wiley, IEEE PRESS, pp. 745-770.
- [20] Yannis Tsividis e Colin McAndrew. *Operation and modeling of the MOS transistor*, 3rd Edition: Oxford University Press, pp. 151-230.
- [21] Tony Chan Carusone, David A. Johns e Kennet W. Martin. *Analog Integrated Circuit Design*, 2nd Edition: WILEY, pp. 310-311.
- [22] H. Banba, H. Shiga e et al. "A CMOS bandgap reference circuit with sub-1-V operation," in: *IEEE Journal of Solid-State Circuits*, vol. 34, no. 5 (1999), pp. 670-674.
- [23] P. van der Meer e et al. *Low-Power Deep Sub-Micron CMOS Logic*. Springer, 2004, pp.77-91.
- [24] Simon Sze. *Dispositivi a semiconduttore*. Milano, Italia: Hoepli, 1991, pp.251-252.
- [25] K. E. Kuijk. "A precision reference voltage source," in: *IEEE Journal of Solid-State Circuits*, vol. 8, no. 3 (1973), pp. 222-226.
- [26] O.Aiello, P.Crovetti e M.Alioto. "Standard Cell-Based Ultra-Compact DACs in 40-nm CMOS," in: *IEEE Access*, vol. 7 (2019), pp. 126479-126488.
- [27] P. S. Croveti, R.Rubino e F.Musolino. "Relaxation Digital-to-Analog Converter with Foreground Digital Self-Calibration," in: *2020 IEEE International Symposium on Circuits and Systems (ISCAS), Seville, Spain, (2020)*, pp. 1-5.
- [28] Christian Rivera et al. "A 3.2 nW Current-Starved Voltage Controlled Ring Oscillator for Low Power Applications," in: (2018).
- [29] M. Z. Straayer e M. H. Perrott. "A 12-Bit, 10-MHz Bandwidth, Continuous-Time $\Sigma\Delta$ ADC With a 5-Bit, 950-MS/s VCO-Based Quantizer," in: *IEEE Journal of Solid-State Circuits*, vol. 43, no. 4 (2008), pp. 805-814.
- [30] P. S. Croveti. "All-Digital High Resolution D/A Conversion by Dyadic Digital Pulse Modulation," in: *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 3 (2017), pp. 573-584.
- [31] Hector Hung e Vladislav Adzic. "Monte Carlo Simulation of Device Variations and Mismatch in Analog Integrated Circuits," in: *Department of Electrical Engineering Columbia University*, (2006).