**Dipartimento di** [**Ingegneria dell'Energia Elettrica e dell'Informazione "Guglielmo Marconi"- DEI**](https://dei.unibo.it/it)

**Corso di Laurea magistrale in**

**ingegneria elettronica**

**studio e progettazione di un sottrattore di corrente per calcolo analogico in memoria**

**Tesi di laurea magistrale in Elettronica Digitale M**

**Relatore**

**Prof. Eleonora Franchi Scarselli**

**Correlatori**

**Dott. Ing. Lorenzo Greco**

**Dott. Ing. Alessio Antolini**

**Dott. Ing. Andrea Lico**

**Presentata da**

**Fatima Gahem**

**Sessione marzo 2025**

**Anno Accademico 2023/2024**

**INDICE**

[ABSTRACT 8](#_Toc193116743)

[Capitolo 1 9](#_Toc193116744)

[Introduzione 9](#_Toc193116745)

[1.1 Panoramica 9](#_Toc193116746)

[1.2 Le memorie PCM 11](#_Toc193116747)

[1.3 Realizzazione della MVM 15](#_Toc193116748)

[1.4 Convertitori 19](#_Toc193116749)

[1.5 Applicazioni 22](#_Toc193116750)

[1.6 Organizzazione della tesi 23](#_Toc193116751)

[Capitolo 2 24](#_Toc193116752)

[Scopo del sottrattore 24](#_Toc193116753)

[2.1 Introduzione 24](#_Toc193116754)

[2.2 Considerazioni iniziali 27](#_Toc193116755)

[Capitolo 3 29](#_Toc193116756)

[Sottrattore ideale single-ended 29](#_Toc193116757)

[3.1 Schema circuitale 29](#_Toc193116758)

[3.2 Studio su LTspice 31](#_Toc193116759)

[Capitolo 4 36](#_Toc193116760)

[Soluzione fully-differential 36](#_Toc193116761)

[4.1 Schema circuitale 36](#_Toc193116762)

[4.2 Studio e simulazione dello schema 37](#_Toc193116763)

[Capitolo 5 41](#_Toc193116764)

[Confronto fra le soluzioni not e common source 41](#_Toc193116765)

[5.1 Soluzione con invertitore 41](#_Toc193116766)

[5.2 Soluzione con Common Source 49](#_Toc193116767)

[Capitolo 6 51](#_Toc193116768)

[Risultati simulazione transitorio 51](#_Toc193116769)

[6.1 Schema con invertitore 52](#_Toc193116770)

[6.2 Schema con source comune 54](#_Toc193116771)

[6.3 Schema con pmos 56](#_Toc193116772)

[Capitolo 7 63](#_Toc193116773)

[Considerazioni sull’ADC 63](#_Toc193116774)

[7.1 Panoramica 63](#_Toc193116775)

[7.2 Architetture di tensione 64](#_Toc193116776)

[7.3 Architettura basata su CCO 67](#_Toc193116777)

[7.4 Caso studiato 70](#_Toc193116778)

[7.5 Considerazioni finali 74](#_Toc193116779)

[8. Conclusioni 75](#_Toc193116780)

[Riferimenti bibliografici 76](#_Toc193116781)

[Ringraziamenti 78](#_Toc193116782)

**Elenco delle figure**

[Figura 1: confronto fra le varie soluzioni in termini di velocità e consumo [1] 10](#_Toc193115269)

[Figura 2: Schema della cella PCM nel caso di Reset e Set dello stato [3] 12](#_Toc193115270)

[Figura 3: impulsi di reset e set applicati nel tempo [4] 13](#_Toc193115271)

[Figura 4: parametri regolabili degli impulsi di set e reset [6] 14](#_Toc193115272)

[Figura 5: vista complessiva di una matrice di memorie resistive [1] 15](#_Toc193115273)

[Figura 6: Crossbar con la reference column [1] 16](#_Toc193115274)

[Figura 7: Crossbar con due memorie resistive per peso sinaptico [1] 17](#_Toc193115275)

[Figura 8: matrice delle conduttanze con relativa circuiteria annessa [1] 17](#_Toc193115276)

[Figura 9: schema dell'ADC basato su CCO 20](#_Toc193115277)

[Figura 10: non linearità della caratteristica 20](#_Toc193115278)

[Figura 11: schema attualmente usato 25](#_Toc193115279)

[Figura 12: schema proposto 26](#_Toc193115280)

[Figura 13: implementazione circuitale del sottrattore 29](#_Toc193115281)

[Figura 14: caratteristica ingresso-uscita delle correnti 30](#_Toc193115282)

[Figura 15: circuito single-ended ideale 31](#_Toc193115283)

[Figura 16: correnti di ingresso e di uscita 32](#_Toc193115284)

[Figura 17: Circuito single-ended reale 33](#_Toc193115285)

[Figura 18: correnti di ingresso e di uscita 33](#_Toc193115286)

[Figura 19: Circuito single-ended reale modificato 34](#_Toc193115287)

[Figura 20: Sottrattore Fully-differential 36](#_Toc193115288)

[Figura 21: circuito fully-differential con uscite specchiate 37](#_Toc193115289)

[Figura 22: grafico delle uscite al variare di I1 38](#_Toc193115290)

[Figura 23: andamento delle correnti nel divisore 39](#_Toc193115291)

[Figura 24: andamento delle tensioni di drain di M9, M10, M11 ed M12 39](#_Toc193115292)

[Figura 25: prima modifica alla soluzione fully-differential 42](#_Toc193115293)

[Figura 26: uscite ottenute dopo la modifica, al variare di I1 43](#_Toc193115294)

[Figura 27: correnti sui rami dei divisori 43](#_Toc193115295)

[Figura 28: Consumo associato all'invertitore introdotto 43](#_Toc193115296)

[Figura 29: ingresso V(-i/2) e uscita V(n002) dell'invertitore 44](#_Toc193115297)

[Figura 30: implementazione con invertitore limitato in corrente 45](#_Toc193115298)

[Figura 31: uscite ottenute in presenza della limitazione in corrente 46](#_Toc193115299)

[Figura 32: consumo associato agli invertitori 46](#_Toc193115300)

[Figura 33: schema con generatori reali 47](#_Toc193115301)

[Figura 34: consumo e uscite associati allo schema reale 48](#_Toc193115302)

[Figura 35: soluzione con controllo a common source 49](#_Toc193115303)

[Figura 36: risultati relativi all'implementazione a source comune 50](#_Toc193115304)

[Figura 37: andamento delle correnti di bitline nel tempo 51](#_Toc193115305)

[Figura 38: andamento delle correnti con un consumo pari a 30 µA 52](#_Toc193115306)

[Figura 39: andamento delle correnti con un consumo pari a 20 µA 53](#_Toc193115307)

[Figura 40: andamento delle correnti con un consumo pari a 10 µA 53](#_Toc193115308)

[Figura 41: andamento delle correnti con un consumo pari a 30 µA 54](#_Toc193115309)

[Figura 42: andamento delle correnti con un consumo pari a 20 µA 55](#_Toc193115310)

[Figura 43: andamento delle correnti con un consumo pari a 10 µA 55](#_Toc193115311)

[Figura 44: implementazione con soli pmos 56](#_Toc193115312)

[Figura 45: grafici associati allo schema di figura 44 57](#_Toc193115313)

[Figura 46: implementazione con pmos M16 ed M24 a soglia più alta 58](#_Toc193115314)

[Figura 47: risultati riferiti allo schema di figura 46 59](#_Toc193115315)

[Figura 48: uscite e ingressi del sottrattore 60](#_Toc193115316)

[Figura 49: uscite e ingressi del sottrattore in una nuova configurazione degli ingressi 61](#_Toc193115317)

[Figura 50: uscite per nuova configurazione 62](#_Toc193115318)

[Figura 51: vincoli nel progetto di un ADC [8] 63](#_Toc193115319)

[Figura 52: implementazioni dell'ADC (a) Flash (b) SAR (c) IS [8] 64](#_Toc193115320)

[Figura 53: Schematico dell'ADC all'interno del core HERMES. (a) Stadio di regolazione, (b) CCO, (c) Ripple Counter [9] 68](#_Toc193115321)

[Figura 54: schema di connessione fra le uscite del sottrattore e l'ADC 70](#_Toc193115322)

[Figura 55: schematico di figura 46 con modifica aggiunta 71](#_Toc193115323)

[Figura 56: andamento delle correnti e delle tensioni nel tempo 72](#_Toc193115324)

[Figura 57: logica introdotta per controllare le uscite 73](#_Toc193115325)

# ABSTRACT

L’AIMC è un tema attualmente rinomato, perché fornisce alle applicazioni emergenti un’alternativa in grado di superare il collo di bottiglia posto dall’architettura di Von Neumann. Nella nuova implementazione, archiviazione e computazione sono svolte nella stessa unità, realizzata da una matrice di memorie resistive, le quali sono capaci di permanere in uno stato, fra tanti, per diverso tempo. Le celle di memoria sono realizzate da conduttanze, i cui valori derivano dai pesi delle matrici sinaptiche, che possiedono valori sia positivi sia negativi, presenti all’intersezione di ogni riga e colonna. Sulle *Wordline*, le righe della matrice, si forniscono gli ingressi in tensione attraverso un DAC, sulle *Bitline*, le colonne della matrice, si raccolgono le correnti, ottenute mediante la legge di Ohm V = R·I, attraverso un ADC. Il blocco complessivo possiede un consumo importante associato a DAC e ADC; pertanto, è necessario prestare grande attenzione a tali componenti. Il consumo peggiora con l’aumento della loro risoluzione, per ridurre quest’ultima è stata introdotta la tecnica di lettura in due fasi temporali, nella prima si forniscono gli ingressi alle conduttanze positive, nella seconda a quelle negative. In questo modo si evita di dimensionare tali componenti per un range esteso da valori negativi a positivi, ma solo per positivi. Al termine della conversione analogico-digitale, si effettua la sottrazione fra i due valori di Bitline e si ottiene il valore desiderato.

In questa tesi si propone un’alternativa all’utilizzo delle due fasi, data dall’aggiunta di un sottrattore analogico in grado di fornire la differenza fra le Bitline all’ingresso dell’ADC. L’introduzione di tale circuito permetterebbe l’utilizzo di una singola fase, raddoppiando così il throughput, senza dimezzarne, però, il consumo, poiché è richiesta la duplicazione dell’intera matrice. Sono studiate due soluzioni circuitali differenti, la prima single-ended e la seconda fully-differential, e sono mostrate le simulazioni associate che ne dimostrano la validità.

# Capitolo 1

## Introduzione

### Panoramica

La parallelizzazione di operazioni per accelerare la computazione non è più solo un obiettivo, ma è diventata una necessità: le recenti applicazioni hanno rilevato grossi svantaggi nelle strutture attualmente usate, comportando così un’esigenza nel realizzare nuove architetture.

Le reti neurali (Artifical Neural Networks ANNs) sono strutture formate da molteplici nodi interconnessi fra di loro, che permettono l’implementazione di operazioni matematiche, e sono definite in questo modo perché si ispirano al modello del cervello umano. I sistemi usati fino ad ora sono basati sull’architettura di Von Neumann, per cui sono formati da una CPU e da una memoria: nella CPU vengono elaborati sia i dati sia le istruzioni contenuti in memoria e salvati nella medesima. Ciò comporta un frequente accesso in memoria, che diventa limitante quando è implicata una notevole quantità di dati, perché gran parte dell’energia e del tempo vengono spesi in questo scambio.

Per far fronte a questa importante problematica, sono state adottate diverse soluzioni:

1. **GPU**, Graphic Processor Unit, le quali forniscono un’accelerazione dei tempi di computazione, perché permettono di svolgere molteplici operazioni in parallelo. Purtroppo, esse risultano essere dispendiose in termini di consumo, richiedendo sistemi di calcolo così grandi, tali da non poter essere integrabili in dispositivi mobili.
2. **FPGA**, field programmable gate arrays, che consumano meno delle GPU e si pongono in una posizione intermedia fra CPU e GPU.
3. **ASIC**, application specific integrated circuits, acceleratori dedicati basati sulla tecnologia CMOS e sono capaci di calcolare e salvare informazioni nella stessa unità. Questi circuiti vengono suddivisi in due categorie:
   1. **Processori dataflow**, che sono customizzati per l’inference e il training della rete neurale, e contengono in modo esplicito il numero di computazioni e accessi in memoria.
   2. **Acceleratori** PIM, processor in memory, che integrano l’elaborazione direttamente in memoria.

Le implementazioni descritte finora sono fully-CMOS e richiederebbero molteplici dispositivi per simulare ogni singola sinapsi; quindi, sono poco efficienti dal punto di vista dell’area e del consumo. [1]

Immagine che contiene linea, schermata, Diagramma, diagramma

Il contenuto generato dall'IA potrebbe non essere corretto.

Figura 1: confronto fra le varie soluzioni in termini di velocità e consumo [1]

È stata studiata la tecnica IMC e una sua particolare realizzazione, su cui si concentra questo lavoro di tesi, è l’AIMC, che sfrutta il calcolo analogico e utilizza memorie non volatili di tipo resistivo. La soluzione si basa sull’idea di raggruppare elaborazione e memorizzazione in un’unica unità, che consente di evitare il consumo e la latenza associati al traffico dei dati. Questa architettura è costituita da dispositivi a memoria resistiva, ovvero sistemi di memoria con materiali che presentano una resistenza elettrica programmabile a due o più stati stabili, generalmente attraverso l’applicazione di stress elettrici. Questi dispositivi possiedono principalmente tre vantaggi:

1. È possibile programmare molteplici stati non volatili,
2. Il consumo di energia per la programmazione della memoria è piuttosto basso,
3. È possibile ottenere una strutturata scalabile appropriata per l’integrazione matriciale. [1]

### Le memorie PCM

Fra le diverse memorie resistive, sono rilevanti le memorie PCM (Phase-Change Memory), le quali si basano sull’utilizzo di un materiale a cambiamento di fase, sfruttando la combinazione delle sue notevoli proprietà, come la grande differenza di resistività fra gli stati amorfo e policristallino, l'elevata velocità di cristallizzazione della fase amorfa e la stabilità dei due stati programmati durante il tempo. La cella viene letta e programmata usando impulsi elettrici, che generano calore e inducono una trasformazione di fase nel materiale della cella di memoria. Il riscaldamento Joule nella cella PCM è responsabile delle diverse operazioni di programmazione e lettura. Il materiale inizialmente sfruttato per la realizzazione di tali celle è il composto stechiometrico Ge2Sb2Te5, detto GST, che possiede un elevato contrasto elettrico e una capacità di memorizzazione di molteplici livelli di informazioni in un singolo elemento di memoria. Tuttavia, questi dispositivi non sono in grado di raggiungere la stabilità termica desiderata per applicazioni ad alta temperatura di funzionamento e per quelle ad alte prestazioni, che richiedono un'elevata velocità con una bassa corrente di programmazione. Per far fronte a questi aspetti, sono state effettuate ottimizzazioni delle prestazioni del dispositivo in varie direzioni, grazie all’ingegneria dei materiali, consentendo di ottenere una riduzione della corrente di programmazione, un aumento della velocità di programmazione o un miglioramento della stabilità termica degli stati programmati. [2]

Un array PCM è una matrice di celle di memoria in cui ad ognuna di esse corrisponde un elemento di archiviazione posto in serie con un elemento selettore, per esempio un transistor. L'elemento di memorizzazione è un resistore variabile costituito da un piccolo volume di materiale a cambiamento di fase inserito tra due elettrodi. L'informazione binaria è codificata attraverso la fase del materiale, quest'ultimo è amorfo o cristallino. L'operazione di lettura si basa sulla differenza nella resistività elettrica presentata dai due stati, la fase cristallina è tipicamente 103 volte più conduttiva della fase amorfa. Per quanto riguarda l'operazione di scrittura, un impulso di programmazione viene applicato all'elemento di accumulo, in modo che una parte del materiale a cambiamento di fase venga riscaldata dall'effetto Joule. Le caratteristiche associate all’impulso, come l’intensità e la durata, consentono di passare dallo stato amorfo a quello cristallino nel seguente modo:

* per lo stato amorfo ad alta resistenza noto come stato RESET, l'applicazione di un impulso di programmazione di breve durata e alta intensità riscalda il materiale a cambiamento di fase fino ad una temperatura superiore a quella di fusione. La geometria del dispositivo e le proprietà termiche dei materiali che circondano l'area attiva sono ottimizzate in modo tale che la zona fusa venga rapidamente solidificata dopo l'impulso di scrittura, determinando così lo stato amorfo al termine dell'operazione di scrittura;
* per ottenere lo stato cristallino a bassa resistenza, detto stato SET, viene applicato un impulso di programmazione di durata maggiore ma di intensità inferiore, riscaldando così il materiale a cambiamento di fase fino ad una temperatura compresa fra quella di transizione vetrosa e quella di fusione. Ciò consente al materiale di trasformarsi in uno stato termodinamicamente stabile o metastabile, che è la fase cristallina. La durata degli impulsi di programmazione necessari per l'amorfizzazione e per la cristallizzazione è dell'ordine di alcune decine di nanosecondi, essendo queste durate, rispettivamente, governate dall'inerzia termica della cella che controlla il tempo necessario per raggiungere la fusione del materiale a cambiamento di fase e dal tempo necessario per i riarrangiamenti atomici necessari per la cristallizzazione.

Queste due trasformazioni avvengono indipendentemente dallo stato iniziale del materiale, consentendo così un'operazione di programmazione diretta, senza alcuna fase di inizializzazione che richiederebbe molto tempo. [2]

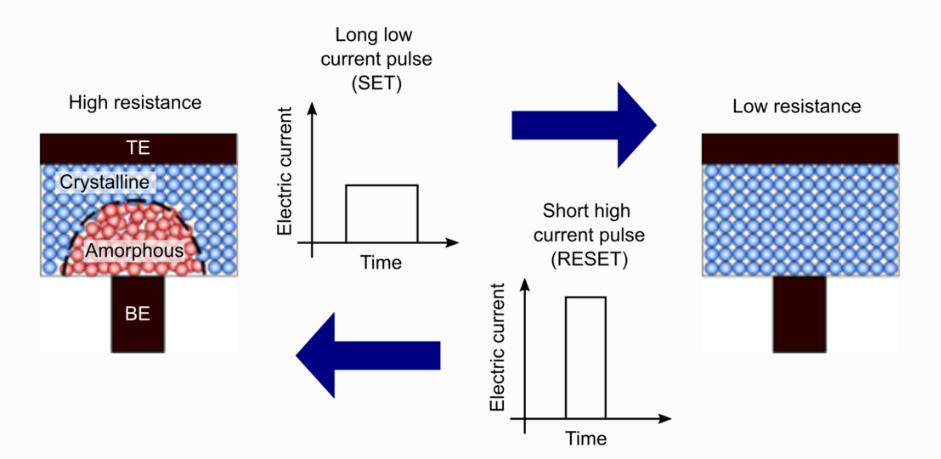


Figura 2: Schema della cella PCM nel caso di Reset e Set dello stato [3]

Immagine che contiene testo, linea, schermata, diagramma

Il contenuto generato dall'IA potrebbe non essere corretto.

Figura 3: impulsi di reset e set applicati nel tempo [4]

Una caratteristica importante della PCM, che la rende molto interessante, è la capacità di archiviazione dei dati su più livelli, e non soltanto su due. Il grande contrasto di resistività tra le fasi cristallina e amorfa permette l'introduzione di stati resistivi intermedi per la memorizzazione delle informazioni, oltre a SET e RESET. La possibilità di impostare la cella in un ampio range di stati intermedi di conduttanza si ottiene attraverso un adeguato controllo delle diverse configurazioni delle fasi cristallina e amorfa all'interno del volume attivo: in altri termini, la resistenza della cella dipende dalla forma e dal volume delle due fasi. Controllando attentamente i parametri dell’impulso, è possibile regolare con precisione la resistenza analogica della PCM, permettendo così l’implementazione dell’archiviazione multilivello. I parametri su cui è possibile agire sono i seguenti:

* per l'impulso di SET si possono effettuare una modulazione in ampiezza AS, in larghezza della parte piatta TON,S, e in pendenza decrescente ΔI/ΔT;
* per l'impulso di RESET si possono modulare l’ampiezza AR e la larghezza TON,R. [5] [6]

Immagine che contiene diagramma, linea, Diagramma, Parallelo

Il contenuto generato dall'IA potrebbe non essere corretto.

Figura 4: parametri regolabili degli impulsi di set e reset [6]

Nonostante i notevoli vantaggi permessi da queste memorie, esistono diverse problematiche associate, responsabili di una precisione limitata del risultato che impatta sulle prestazioni a livello applicativo:

1. la non linearità della caratteristica I-V, in genere affrontata codificando i valori di ingresso nel tempo come ampiezza di un segnale a tensione costante applicata su ciascuna cella dell'array,
2. la deriva della conduttanza nel tempo, solitamente compensata da tecniche di post-elaborazione o dall'impiego di soluzioni di addestramento compatibili con HW,
3. la variabilità degli stati delle celle programmati: è possibile utilizzare algoritmi iterativi di programmazione e verifica all'avanguardia per ridurre, ma non rimuovere completamente, tale variabilità in modo identico. Pertanto, la variabilità da dispositivo a dispositivo, dovuta sia alla deriva che allo stato di programmazione, è ancora un argomento ampiamente studiato che limita l'accuratezza dei sistemi basati su PCM e che non può essere correttamente compensato. [7]

### Realizzazione della MVM

In una ANN vengono svolte diverse operazioni matematiche, fra queste domina la MVM, vector matrix multiplication, sull’intero carico di lavoro. Tale operazione deve essere accelerata, essendo ricorrente, e può essere implementata attraverso un crossbar array di elementi di memoria, come la PCM. Essa contiene celle in grado di realizzare l’operazione di moltiplicazione attraverso la legge di Ohm, I = V x G, con V la tensione applicata in ingresso e G la conduttanza di ogni memoria. [1]

Immagine che contiene diagramma, schermata, Piano

Il contenuto generato dall'IA potrebbe non essere corretto.

Figura 5: vista complessiva di una matrice di memorie resistive [1]

In questo modo, l’area e il consumo sono estremamente ridotti. Un’opzione adottata per controllare una cella è data dall’aggiunta di un selettore in serie alla resistenza, che abilita il flusso di corrente. Per effettuare la MVM è necessario aggiungere altra circuiteria per convertire i segnali elettrici, come DAC e ADC, e amplificatori a transimpedenza per fare il sensing delle correnti di uscita. Le tensioni generate da ogni DAC sono applicate agli input (le righe) della crossbar che sono le *Wordline* mentre le colonne di uscita, le *Bitline*, sono connesse a massa attraverso l’amplificatore. La corrente che fluisce su ogni memoria verrà data dalla tensione applicata e dalla conduttanza di ogni memoria:

La matrice delle conduttanze si definisce a partire dalla matrice dei pesi sinaptici, WM, che è formata da numeri reali, sia positivi che negativi, di molteplici valori poiché l’archiviazione multilivello è resa possibile. I valori negativi non possono essere direttamente rappresentati con le conduttanze, quindi si adottano delle strategie:

1. Viene aggiunta una colonna detta reference column, dove tutte le sue conduttanze sono settate a 0.5·GLRS e la corrente alla j-esima uscita è ottenuta sottraendo alla corrente generata dalla reference column la corrente generata dalla j-esima colonna:

La gref rappresenta 0.5·GLRS mentre g’i,j viene calcolata in modo tale che dispositivi con una conduttanza maggiore di 0.5·GLRS produca pesi sinaptici positivi, altrimenti negativi. Questa strategia, purtroppo, presenta due svantaggi:

* Il range dei pesi viene ridotto,
* Il routing della reference column al resto delle colonne non è banale. [1]

Immagine che contiene testo, schermata, diagramma, linea

Il contenuto generato dall'IA potrebbe non essere corretto.

Figura 6: Crossbar con la reference column [1]

1. Si usano due memorie resistive per peso sinaptico, generando così due crossbar. La corrente diventa:

In questo modo le conduttanze positive e negative sono codificate da una coppia di memorie resistive adiacenti, e , ottenendo così un raddoppio del range dei livelli di conduttanza che rende la crossbar meno sensibile al rumore e alla variabilità. [1]

Immagine che contiene testo, schermata, diagramma, Carattere

Il contenuto generato dall'IA potrebbe non essere corretto.

Figura 7: Crossbar con due memorie resistive per peso sinaptico [1]

Alla corrente d’uscita viene fatto un sensing, che può essere implementato attraverso una resistenza, un amplificatore a transimpedenza oppure una capacità, e la scelta dipende dalla dimensione della corrente da misurare.

Immagine che contiene testo, diagramma, linea, Carattere

Il contenuto generato dall'IA potrebbe non essere corretto.

Figura 8: matrice delle conduttanze con relativa circuiteria annessa [1]

La corrente d’uscita di ogni coppia di bitline è una somma lineare pesata di tutte le wordline connesse alla colonna considerata. Per replicare una funzione complessa non lineare attraverso una ANN, è necessario applicare una funzione di attivazione ad ogni colonna per introdurre non linearità. Questa può essere implementata via software o via hardware, la prima risulta essere la più usata attualmente perché la sua eventuale modifica richiede una modifica al codice, mentre su un circuito fabbricato non si può più cambiare.

### Convertitori

La parte più grande e dispendiosa in termini di consumo è associata all’ADC: la sua scelta viene fatta sulla base della risoluzione del DAC, del numero di righe della crossbar e della risoluzione delle conduttanze. Le prestazioni di un ADC cambiano al variare della risoluzione e della frequenza di campionamento, presentando un trade-off fra velocità, area e consumo. Aumentando la risoluzione si ottiene una migliore precisione del risultato, però il convertitore risulta più lento e consuma di più. Per questi motivi l’ADC risulta essere il componente più critico all’interno della rete neurale a memoria resistiva, arrivando a consumare fino all’80-88% dell’energia totale e occupando fino al 70-90% dell’area totale. Per ottimizzare questi aspetti è possibile:

1. Approssimare la computazione oppure ridurre la precisione,
2. Condividere un unico ADC fra alcune o tutte le colonne, che richiede l’aggiunta di multiplexer e circuiti di sample&hold che introducono ulteriore latenza.

Esistono diverse tipologie di ADC che possono essere implementate, a seconda dell’applicazione considerata e delle richieste da soddisfare:

* Per risoluzioni alte, si utilizzano gli ADC SAR (Registro ad Approssimazioni Successive) oppure gli ADC delta-sigma.
* Per frequenze alte, invece, si adottano gli ADC Flash. [1]

Tipicamente, per raggiungere il miglior compromesso fra le due specifiche, si utilizzano ADC a integrazione di carica, che integrano una corrente nel tempo, generando una tensione proporzionale che viene poi digitalizzata. Un esempio specifico è quello dell’ADC basato sul CCO, Current Controlled Oscillator, che genera una frequenza proporzionale alla corrente che riceve in ingresso. Al CCO arrivano più valori di corrente, quindi vengono generati più impulsi a frequenze diverse e questi vengono convertiti in un valore digitale attraverso un contatore [9], come mostrato nella seguente figura:

Immagine che contiene testo, diagramma, schermata, Carattere

Il contenuto generato dall'IA potrebbe non essere corretto.

Figura 9: schema dell'ADC basato su CCO

Con questa implementazione si riduce la complessità circuitale richiesta e non si ha un alto consumo; tuttavia, si presenta un problema di non linearità della caratteristica uscita-ingresso, dominante per alte correnti, che comporta riduzione dell’accuratezza:

Immagine che contiene oscurità, notte, spazio

Il contenuto generato dall'IA potrebbe non essere corretto.

Figura 10: non linearità della caratteristica

Nel capitolo 7 verrà ampiamente approfondita la tematica riguardante gli ADC, sia a livello dei vantaggi sia degli svantaggi.

Precedentemente è stata discussa la possibilità di avere vettori di ingresso con elementi positivi e negativi. Questi ultimi generano la necessità di raddoppiare le uscite del DAC, aumentando così la sua risoluzione di 1 bit. Ciò si riproduce ugualmente sull’ADC, e si ottiene un aumento globale del consumo di potenza e una maggiore sensibilità agli errori. Per evitare questo consumo aggiuntivo, si applicano gli input positivi e negativi in due fasi di lettura separate con tensioni unipolari e poi si applica la sottrazione fra le uscite risultanti dall’ADC. Questa soluzione comporta, però, una riduzione del throughput poiché sono richieste due fasi di lettura per completare una singola VMM, e non più solo una. [1]

### Applicazioni

Le reti neurali sono contenute in diversi dispositivi che vengono utilizzati quotidianamente:

* Nei telefoni, per il riconoscimento dell’impronta o facciale,
* Nelle videocamere di sicurezza per il riconoscimento di immagini,
* Nei motori di conversione da audio a testo,
* Chat-GPT.

Sono di notevole importanza in diversi campi, tra cui quello medico e di sicurezza, e quindi l’esigenza di sviluppare nuove architetture capaci di soddisfare i requisiti richiesti è molto alta. [1]

### 

### Organizzazione della tesi

Nel capitolo 1 si tratta la tematica dell’AIMC su un piano generale. A partire dai capitoli successivi, si analizza l’argomento di tesi, il sottrattore analogico, con la seguente organizzazione:

* Nel capitolo 2 si definiscono le motivazioni per cui viene studiata tale proposta.
* Nel capitolo 3 si analizza e si simula una prima implementazione circuitale, single-ended, mostrandone i difetti.
* Nel capitolo 4 si analizza e si simula una seconda implementazione circuitale, fully-differential, introdotta per risolvere le problematiche della prima soluzione. In questa versione, però, è richiesta una modifica, attraverso l’aggiunta di una logica di controllo.
* Nel capitolo 5 si analizzano due possibili schemi per la logica di controllo.
* Nel capitolo 6 si mostrano i risultati ottenuti dalla simulazione transitorio per le due tipologie di logica di controllo e si introduce la logica di controllo basata su pMOS.
* Nel capitolo 7 si effettua una digressione sui possibili schemi di ADC utilizzati nell’ambito di IMC, focalizzando l’attenzione sulla soluzione a base di CCO e introducendo una modalità per realizzare l’interfacciamento fra sottrattore e ADC.
* Nel capitolo 8 si forniscono le conclusioni relative al lavoro di tesi, presentando i vari risultati ottenuti ed effettuando considerazioni su possibili realizzazioni future.

Tutte le analisi a livello circuitale sono state fatte utilizzando il simulatore LTspice e i modelli open-source di un processo PDK 45 nm.

# Capitolo 2

## Scopo del sottrattore

### 2.1 Introduzione

È stata accennata la tematica sull’utilizzo di due fasi di lettura per l’applicazione di ingressi sia positivi sia negativi, senza introdurre ulteriore consumo. L’ingresso è rappresentato da un impulso temporale, la cui durata è proporzionale all’input digitale per la MVM considerato. Nella prima fase si applicano gli impulsi associati alle conduttanze positive, mentre nella seconda si applicano alle negative. Tale procedura, com’è evidente, riduce il throughput perché rende indispensabile l’uso di due fasi ai fini della riduzione del consumo.

Questa tesi si focalizza sullo studio di una possibile soluzione, in grado di raggiungere un compromesso fra throughput e consumo: al termine delle due fasi, viene eseguita una sottrazione fra le correnti d’uscita, pertanto risulta lecito testare un nuovo schema a monte dell’ADC in grado di fornire in uscita la differenza fra le correnti di bitline, attraverso un circuito che svolge tale operazione, dopo aver raddoppiato l’intera crossbar: le conduttanze, positive e negative, non sono più comandate da un’unica wordline, ma vengono raddoppiate, scambiate e comandate da due wordline diverse, una per ogni coppia di conduttanze positiva-negativa. Tra le due bitline viene posto il sottrattore e la sua uscita viene collegata all’ADC.

Lo schema generale, mostrato in figura 12, è piuttosto semplice e non implica introduzione di complessità circuitale, rispetto allo schema attualmente usato di figura 11.

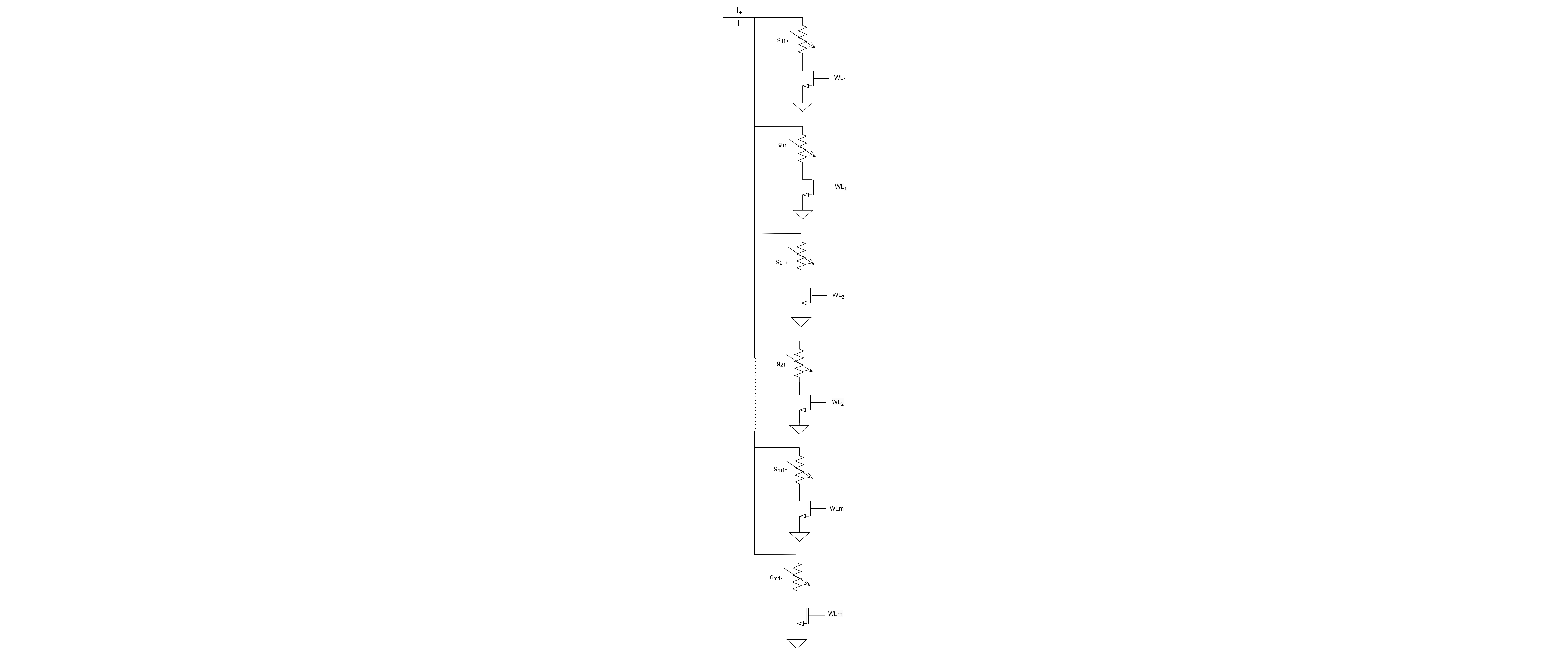


Figura 11: schema attualmente usato

Immagine che contiene schermata, bianco e nero, nero, monocromatico

Il contenuto generato dall'IA potrebbe non essere corretto.

Figura 12: schema proposto

### 2.2 Considerazioni iniziali

Si effettua, di seguito, un’analisi preliminare delle prestazioni associate alla nuova implementazione:

|  |  |
| --- | --- |
| Throughput | *x(2 – ritardo associato al sottrattore)* |
| Consumo di potenza | *x(2 + sottrattore)* |
| Energia | *Maggiore o invariata* |
| Area | *x(2 + sottrattore)* |
| Throughput/Consumo | *Minore o invariato* |
| Throughput/Area | *Minore o invariato* |

In prima approssimazione, si osserva come sia necessario implementare un sottrattore capace di svolgere il calcolo con minor ritardo e consumo possibili, in modo tale da rendere vantaggiosa tale soluzione. Il throughput raddoppia con conseguente raddoppio dell’area occupata e del consumo, come da teoria, però permette di ottenere altri importanti vantaggi:

1. Non è più la bitline ad essere connessa all’ingresso dell’ADC, ma è l’uscita del sottrattore ad esserlo, e ciò risulta vantaggioso in quanto il rumore dovuto alla capacità di integrazione del CCO non si ripercuote più sulla bitline,
2. L’ADC non deve più leggere la corrente che scorre su una bitline, ma la differenza fra le correnti che scorrono su due bitline diverse; pertanto, il range di valori in ingresso viene ridotto e ciò permette all’ADC di lavorare nella zona più lineare della caratteristica ingresso-uscita.

Da un punto di vista matematico, non si hanno cambiamenti in termini di conversione dell’ADC, perché prima veniva svolta la seguente operazione sull’output z:

Mentre ora:

Dove le correnti I1 e I2 rappresentano le correnti di bitline positiva e negativa.

Pertanto, si ottengono gli stessi risultati e il funzionamento atteso non è affetto da errori concettuali.

Il sottrattore può essere realizzato in diversi modi:

1. Con un amplificatore operazionale in configurazione differenziale,
2. Con uno specchio di corrente con mosfet.

Le due proposte differiscono principalmente per il consumo, ma anche per il tipo di alimentazione adottato.

Nella tesi verrà studiata un’implementazione a specchio, si individueranno i suoi svantaggi e verrà proposto uno schema alternativo per risolverli.

# Capitolo 3

## Sottrattore ideale single-ended

### 3.1 Schema circuitale

Per il calcolo della differenza, è stata dapprima adottata la seguente implementazione:

Immagine che contiene luna, oscurità, nero, Oggetto astronomico

Il contenuto generato dall'IA potrebbe non essere corretto.

Figura 13: implementazione circuitale del sottrattore

Essa risulta formata da quattro mosfet di tipo p, connessi in una configurazione a specchio high-swing e da due generatori di corrente IB. In ingresso possiede le correnti I1 e I2 che, nel caso di studio considerato, sono rappresentate dalle correnti di bitline. La corrente di uscita IOUT possiede il seguente andamento:

Immagine che contiene nero, oscurità

Il contenuto generato dall'IA potrebbe non essere corretto.

Figura 14: caratteristica ingresso-uscita delle correnti

Per I1 – I2 ≥ -IB, si ha l’andamento lineare desiderato, altrimenti si ottiene una costante pari a -IB. In una fase preliminare, si osserva che il consumo è interamente dato dai generatori di corrente IB, quindi è necessario ridurlo il più possibile. Questo valore non può essere diminuito in maniera arbitraria, per via della relazione I1 – I2 > -IB: nel caso in cui una delle due correnti di bitline, ad esempio I1, sia pari a zero e l’altra, I2, sia uguale al valore massimo che può assumere, si ottiene che – I2, max ≥ -IB; pertanto, la IB deve essere posta alla massima corrente di bitline per garantire il corretto funzionamento.

### 3.2 Studio su LTspice

Il circuito appena descritto è stato studiato su LTspice, in una tecnologia CMOS a 45 nm, in cui i transistor sono descritti da un modello di livello 14 (BSIM 4). Lo schema di partenza è completamente ideale:

Immagine che contiene testo, diagramma, Piano, linea

Il contenuto generato dall'IA potrebbe non essere corretto.

Figura 15: circuito single-ended ideale

L’uscita del circuito è connessa ad un carico resistivo assunto molto piccolo, in modo tale da simulare un cortocircuito, perché si vuole misurare una corrente senza alterare il suo valore. Nella realtà, l’uscita è connessa al condensatore che effettua il sensing e poi all’ADC. Con questa scelta, l’uscita risulta essere sensibile al carico e i quattro pmos sono stati dimensionati con W = 100 µm e L = 0.8 µm per ridurre al minimo tale dipendenza, come osservato nel seguente grafico:

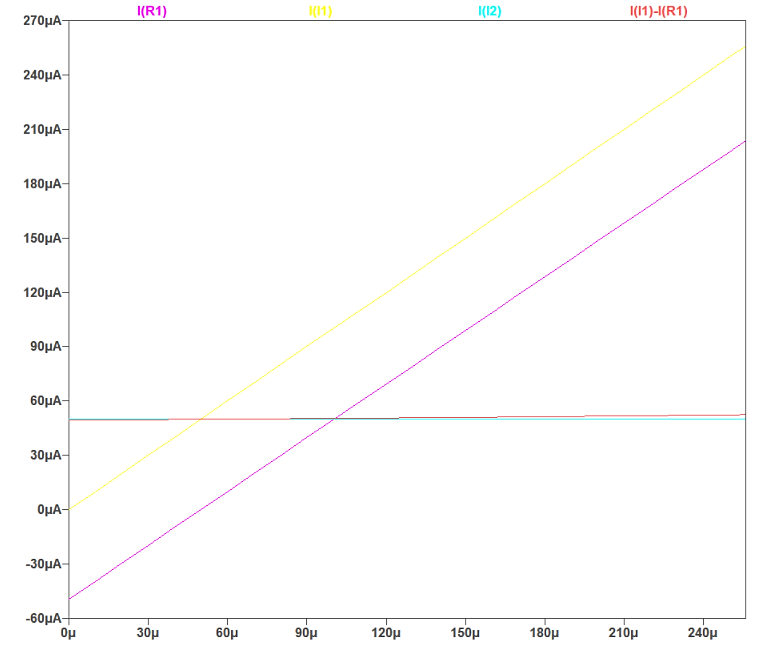


Figura 16: correnti di ingresso e di uscita

Le correnti di ingresso, rappresentate da generatori di corrente, sono state settate nel seguente modo:

* I1 variabile da 0 a 256 µA (traccia blu),
* I2 costante a 50 µA (traccia rossa).

Idealmente in uscita si ottiene Iout = I(R1) = I1 - I2, che è variabile visto che la I1 lo è, però la differenza fra Iout e I1 si deve mantenere costante al valore della I2, ovvero 50 µA. Questo viene verificato nel grafico, per cui le forme d’onda rossa e azzurra sono quasi ovunque coincidenti.

Lo schema appena studiato è da modificare perché è ancora completamente ideale. Si procede sostituendo ai generatori I3 e I4 degli nmos, che svolgono il ruolo di generatore di corrente costante, connessi a specchio ad un altro nmos in serie ad un generatore di corrente di 10 µA, che si trova in libreria:

Immagine che contiene diagramma, Piano, testo, Disegno tecnico

Il contenuto generato dall'IA potrebbe non essere corretto.

Figura 17: Circuito single-ended reale

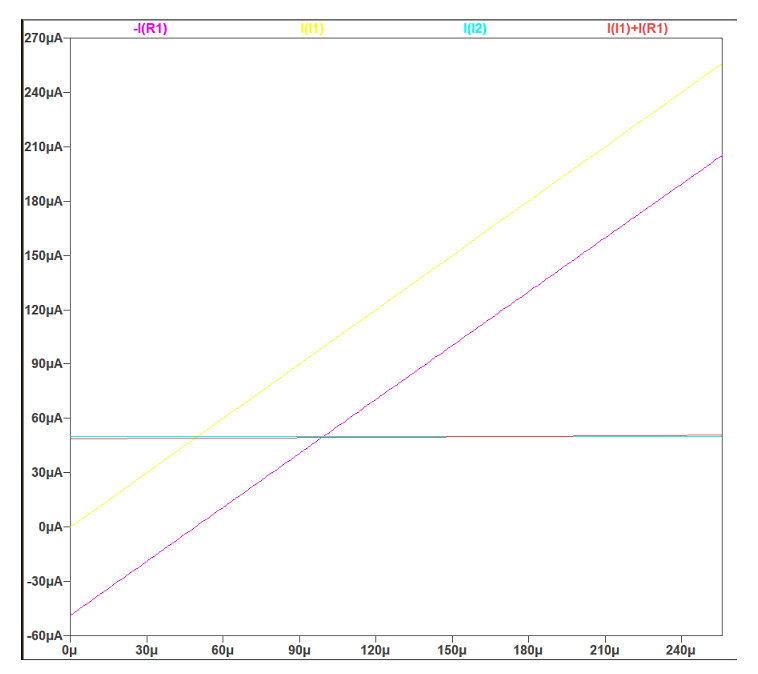


Figura 18: correnti di ingresso e di uscita

Con la seguente modifica, è stato necessario cambiare la configurazione del carico perché altrimenti correnti negative in uscita non sono possibili. Si osserva dal grafico di figura 18 la coerenza dei risultati con il caso completamente ideale e l’ottima precisione ottenuta. È di seguito proposta un’altra implementazione analoga a quella appena vista:

Immagine che contiene diagramma, Piano, linea, Disegno tecnico

Il contenuto generato dall'IA potrebbe non essere corretto.

Figura 19: Circuito single-ended reale modificato

In questo schema, il carico non influenza più l’uscita, in quanto essa viene specchiata attraverso M14 su M15, però permane la non possibilità di ottenere correnti negative.

Il carico non costituisce una problematica effettiva poiché, come precedentemente accennato, nella realtà si presenta un condensatore in parallelo. L’aspetto su cui invece è necessario agire è associato al consumo dei mosfet M5 ed M6: su ciascuno di essi scorre costantemente una corrente pari a 256 µA, per un totale di 512 µA, che è un valore molto alto e non accettabile per via delle considerazioni inizialmente fatte riguardo al consumo. Pertanto, risulta necessario agire nella direzione di riduzione di tale consumo: si propone uno schema alternativo al sottrattore single-ended, che mira a ridurre al minimo il consumo presente finora.

# Capitolo 4

## Soluzione fully-differential

### 4.1 Schema circuitale

Immagine che contiene diagramma, Piano, Disegno tecnico, schematico

Il contenuto generato dall'IA potrebbe non essere corretto.

Figura 20: Sottrattore Fully-differential

Lo schema di figura 20 mostra la soluzione proposta per ridurre il consumo: il circuito di partenza viene reso completamente differenziale e in ingresso ai due sottrattori si hanno le correnti di bitline dimezzate. Il sottrattore a sinistra fornisce in uscita

quando questa differenza è positiva, ovvero per i casi in cui la I1 è maggiore della I2. L’altro sottrattore svolge la differenza opposta, cioè

per i casi in cui la I2 è maggiore della I1. In questo modo, si eliminano risultati negativi, permettendo così una notevole riduzione del consumo: i generatori di corrente di ogni sottrattore non sono più settati alla massima corrente di bitline ma a 1 µA.

### 4.2 Studio e simulazione dello schema

Per valutare la correttezza delle uscite, lo schema proposto di figura 20 è stato perfezionato con l’aggiunta di due specchi sulle uscite e di due resistenze per leggere la corrente:

Immagine che contiene diagramma, linea, Piano

Il contenuto generato dall'IA potrebbe non essere corretto.

Figura 21: circuito fully-differential con uscite specchiate

Il comportamento ottenuto in uscita è il seguente:



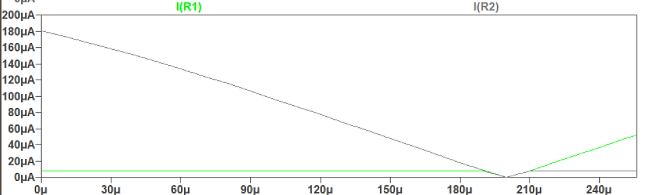


Figura 22: grafico delle uscite al variare di I1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **I1** | **I2** | **I(R1) teorico** | **I(R1) reale** | **I(R2) teorico** | **I(R2) reale** |
| 0 | 200 µ | -100 µ | 7.5 µ | 100 µ | 180.5 µ |
| 30 µ | 200 µ | -85 µ | 7.5 µ | 85 µ | 158.2 µ |
| 60 µ | 200 µ | -70 µ | 7.5 µ | 70 µ | 133.3 µ |
| 90 µ | 200 µ | -55 µ | 7.5 µ | 55 µ | 106 µ |
| 120 µ | 200 µ | -40 µ | 7.5 µ | 40 µ | 77.3 µ |
| 150 µ | 200 µ | -25 µ | 7.5 µ | 25 µ | 47.9 µ |
| 200 µ | 200 µ | 0 | 0 | 0 | 0 |
| 210 µ | 200 µ | 5 µ | 7.8 µ | -5 µ | 7.5 µ |
| 240 µ | 200 µ | 20 µ | 36.9 µ | -20 µ | 7.5 µ |

Come mostrato nella tabella sopra, si ottengono risultati corretti quando le due correnti si annullano, altrimenti si hanno risultati errati. Le correnti negative dovrebbero annullarsi, e non mostrare altri tipi di valori, in questo caso si ha un valore costante pari a 7.5 µA. Perciò, i profili delle correnti risultano corretti ma non i valori associati. Il motivo si attribuisce allo sbilanciamento dei due divisori, che non forniscono sempre la metà della corrente di bitline ricevuta in ingresso:



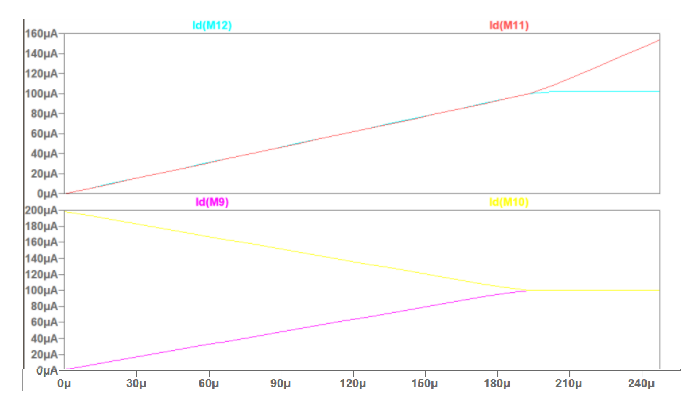


Figura 23: andamento delle correnti nel divisore

Sono state considerate I1 variabile da 0 a 256 µA e I2 fissa a 200 µA: le correnti associate al divisore formato dai mosfet M11 ed M12 devono andare da 0 a 128 µA e quelle associate al divisore formato dai mosfet M9 ed M10 devono essere costanti al valore di 100 µA. Come osservato nei grafici di figura 23, ciò non si verifica e i divisori non si comportano come dovrebbero. L’andamento ottenuto è associato alle tensioni sui nodi di drain di M9 ed M12:

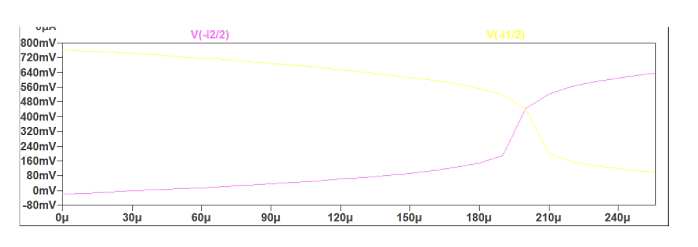




Figura 24: andamento delle tensioni di drain di M9, M10, M11 ed M12

Inizialmente, la tensione di drain di M12, , è alta mentre quella di M9, , è bassa e le correnti di ingresso sono significativamente diverse tra di loro, perché la I1 è bassa mentre la I2 è alta. Le tensioni di drain M10 ed M11 sono circa costanti. Il drain di M9 va ad un valore di tensione molto basso poiché M9 richiede una corrente che valga ma M2 ne eroga una del valore di 1µA + < (in questa configurazione i2 >> i1). Poiché il drain di M9 è a una tensione molto bassa il transistor tende a spegnersi e in tal caso è M10 che deve fornire tutta i2 alla bitline. Questo implica Id(M9) ≠ Id(M10) anziché Id(M9) = Id(M10) = , condizione necessaria al funzionamento del sottrattore. È pertanto necessario che la tensione sul drain di M9 non si abbassi troppo anche nel caso i2 > i1 oppure se il ramo di M9 non è in grado di erogare una corrente pari a ci sia un altro ramo che si accenda e lo faccia, in modo che M10 non debba erogare una corrente diversa da .

# Capitolo 5

## Confronto fra le soluzioni not e common source

### 5.1 Soluzione con invertitore

Come prima soluzione si introduce uno schema formato da un invertitore, usato come stadio amplificatore, che pilota un nmos, il quale è connesso sul drain alla tensione di alimentazione in modo tale che quando la tensione sul nodo di drain del divisore si abbassa, il not fornisce in uscita una tensione alta che accenda l’nmos e si crea il percorso fra alimentazione e divisore. Lo stesso discorso si applica al divisore in ingresso al secondo sottrattore; pertanto, si introducono complessivamente due invertitori e due nmos.

Con questo cambiamento, si verifica l’andamento atteso delle uscite, con un errore massimo pari a 1 µA, e anche le correnti del divisore risultano corrette, come mostrato nelle figure 26 e 27. L’inserimento di un invertitore comporta, però, un consumo aggiuntivo che deve essere limitato, poiché l’intero schema è costruito sull’idea di ottenere il minor consumo possibile. È chiaro che si tratta di funzionalità del circuito, esso deve fornire correttamente le uscite attese, pertanto è importante tale modifica. Nella figura 28 si mostra il consumo associato agli invertitori, che risulta essere complessivamente costante al valore di 30 µA, e la ragione di tale consumo è visibile nel grafico di figura 29, che mostra che gli invertitori lavorano circa alla soglia. La loro presenza è necessaria, o perlomeno si ha l’esigenza di inserire un controllo; pertanto, risulta necessario determinare una strategia tale da permettere la riduzione del consumo associato.

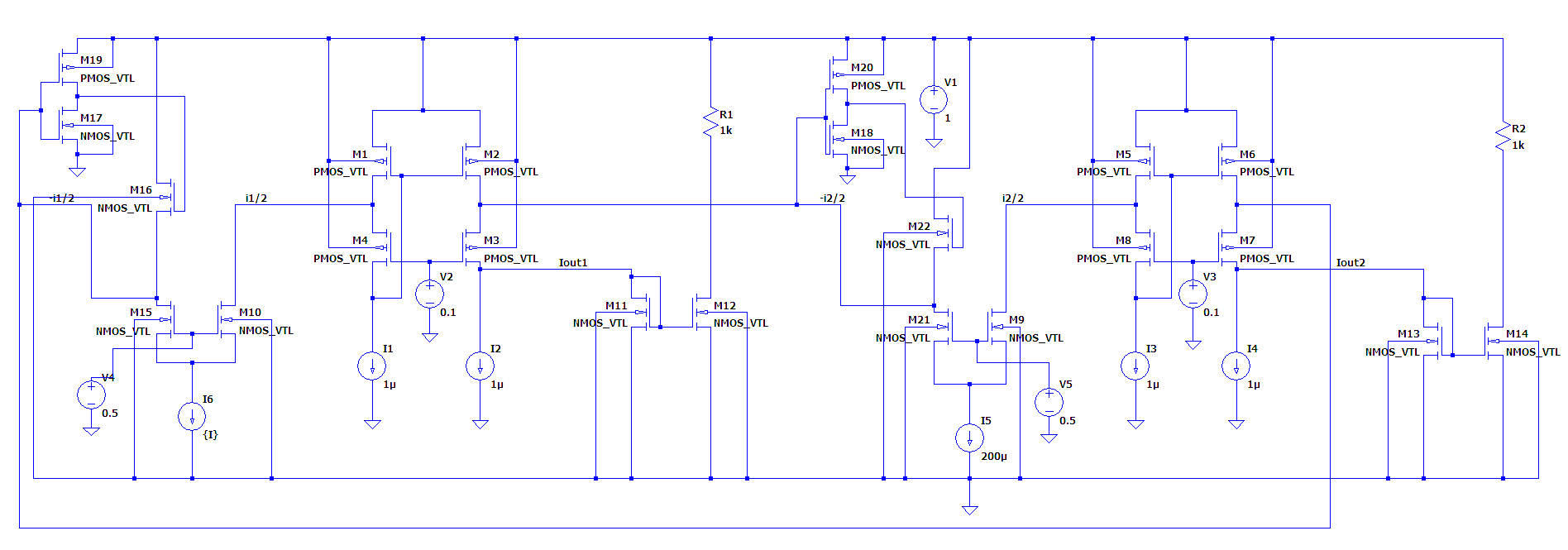


Figura 25: prima modifica alla soluzione fully-differential



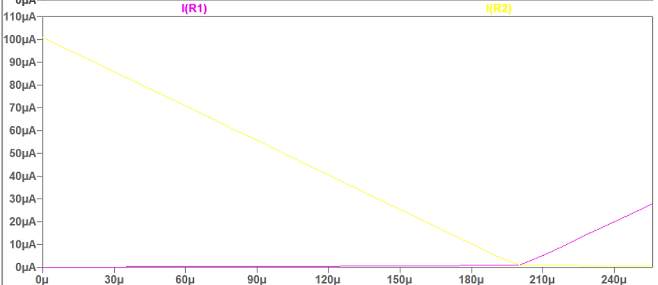




Figura 26: uscite ottenute dopo la modifica, al variare di I1

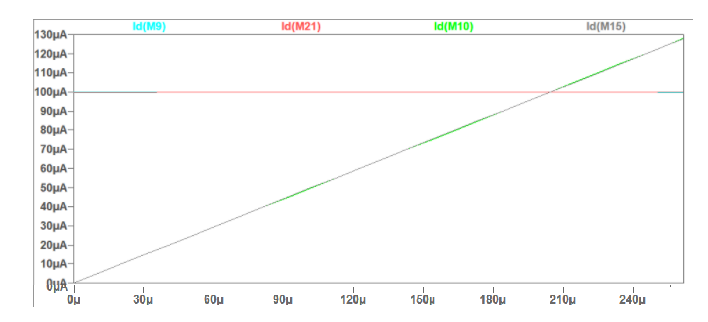




Figura 27: correnti sui rami dei divisori

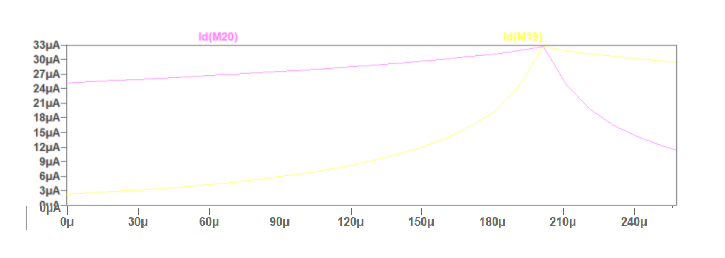




Figura 28: Consumo associato all'invertitore introdotto

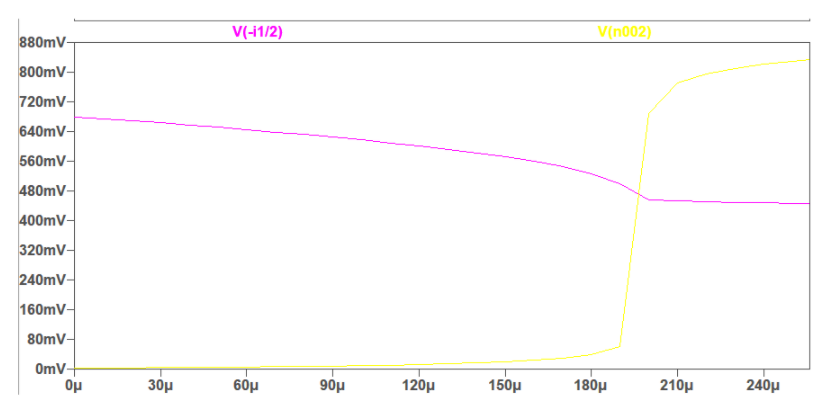


Figura 29: ingresso V(-i/2) e uscita V(n002) dell'invertitore

Come prima soluzione, si introduce la possibilità di limitare i due invertitori in corrente attraverso un generatore di corrente:

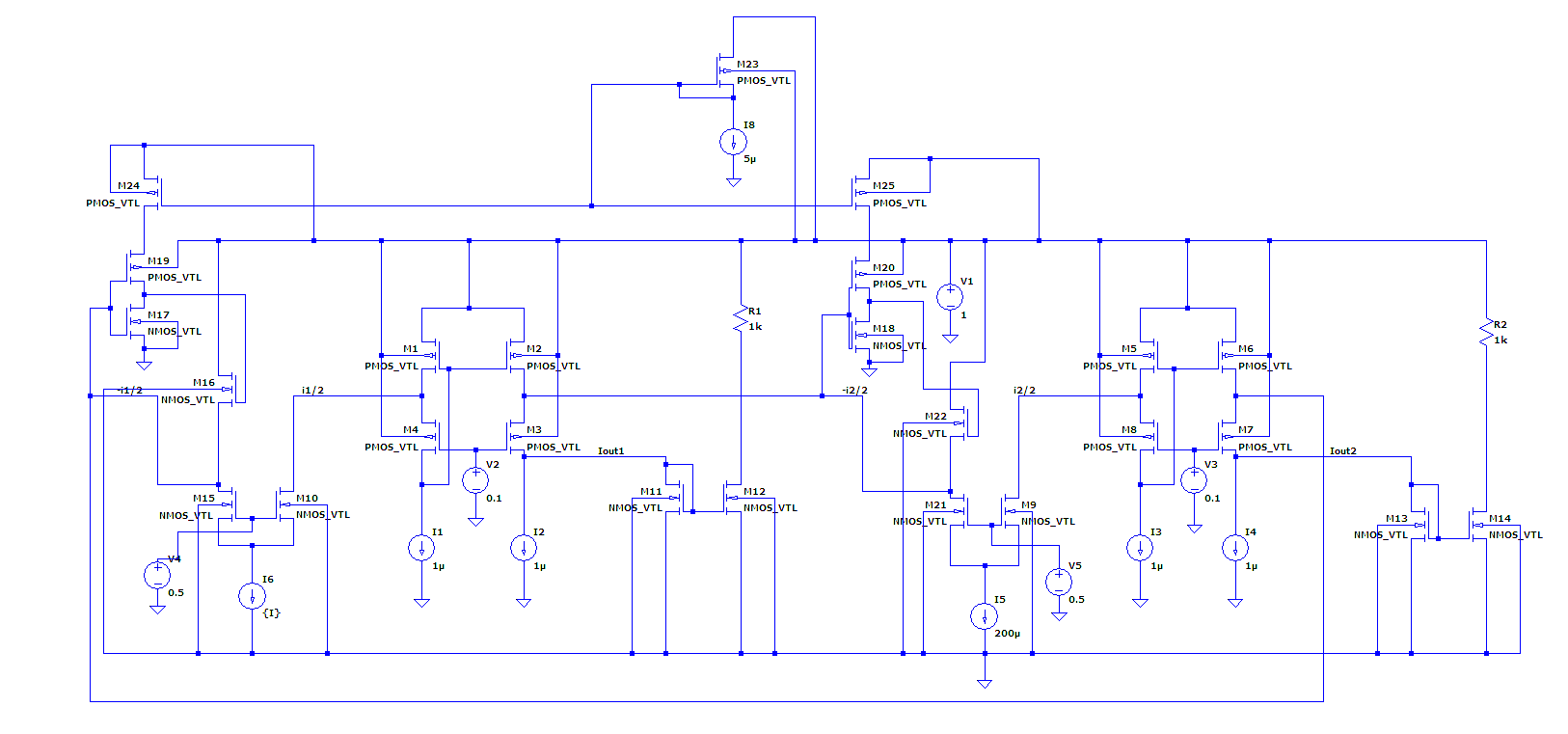


Figura 30: implementazione con invertitore limitato in corrente

La funzionalità è verificata, con un po’ di errore introdotto:

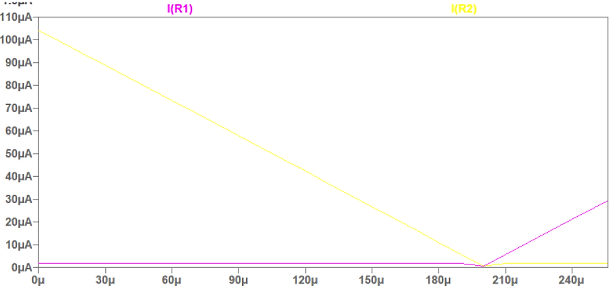


Figura 31: uscite ottenute in presenza della limitazione in corrente

A differenza di prima, però, il consumo è stato drasticamente ridotto (figura 32). Idealmente, quindi, è possibile fissare il consumo al valore desiderato, attraverso il valore del generatore.

Nel capitolo 6 verrà analizzata a fondo questa proposta, perché si intuisce che un vantaggio così notevole implica sicuramente uno svantaggio che attualmente non è visibile.

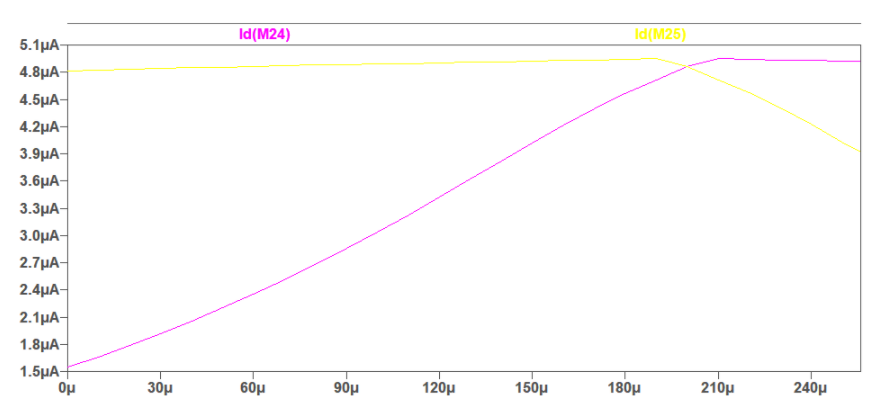


Figura 32: consumo associato agli invertitori

Si procede con la sostituzione dei generatori di corrente ideali con quelli reali:

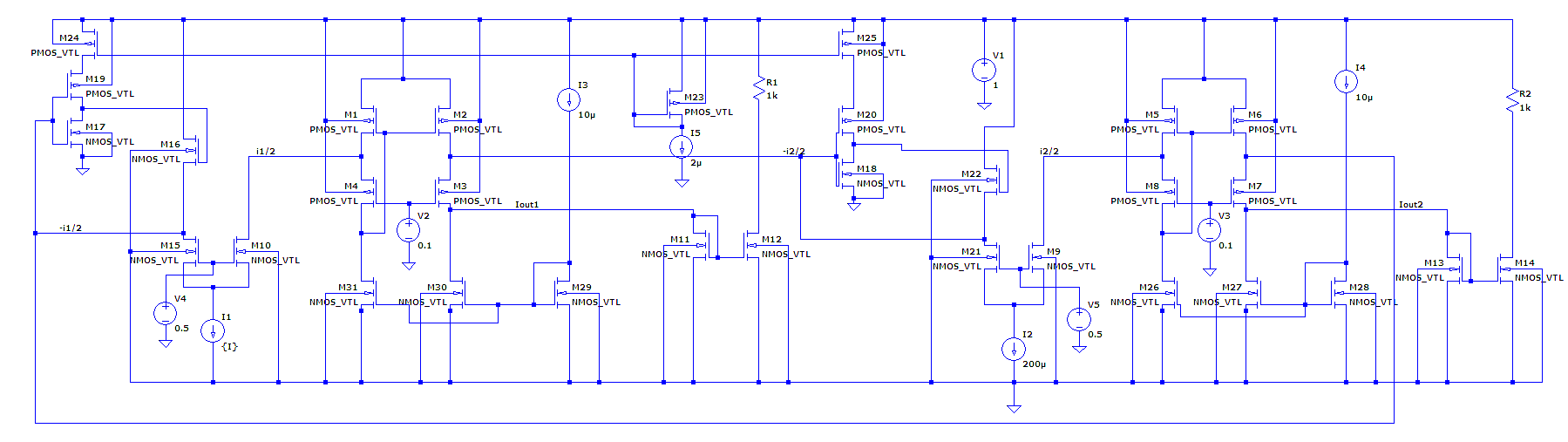


Figura 33: schema con generatori reali

I risultati in uscita sono corretti (l’errore prima presente è stato eliminato) e, variando il valore della corrente impostata nel limitatore, continuano ad esserlo:

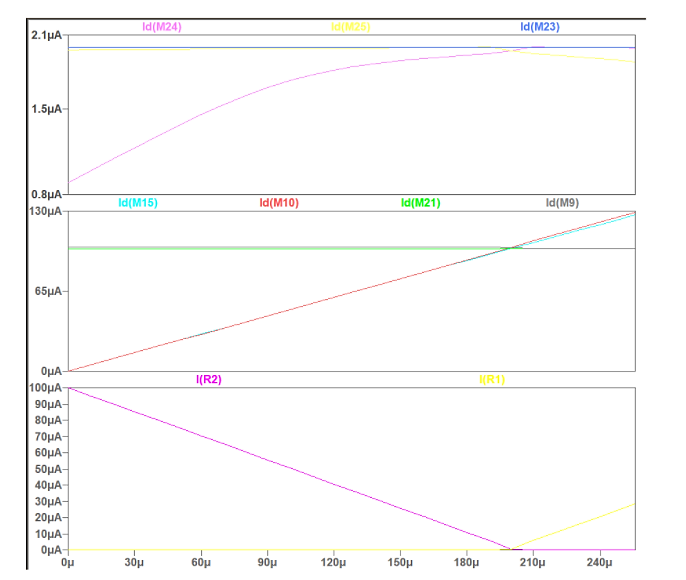


Figura 34: consumo e uscite associati allo schema reale

### 5.2 Soluzione con Common Source

È possibile raggiungere gli stessi risultati adottando una soluzione a source comune, invece dell’invertitore. L’implementazione usata è la seguente:

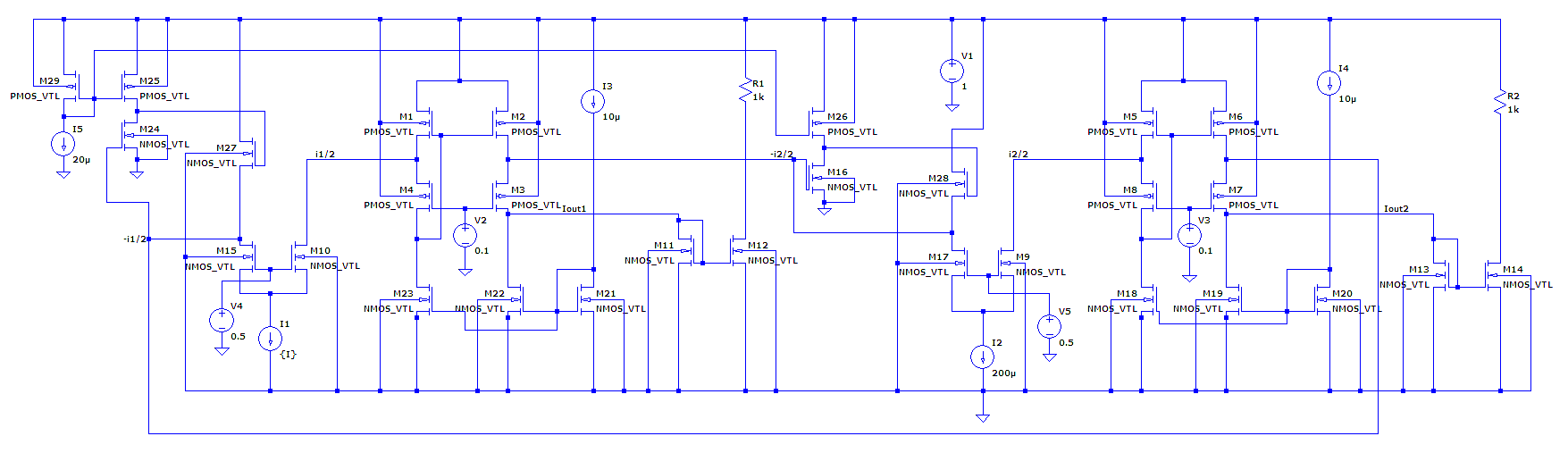


Figura 35: soluzione con controllo a common source

L’implementazione a source comune fornisce un’uscita amplificata quando l’ingresso è basso, la quale permette l’accensione dell’nmos e il conseguente collegamento fra alimentazione e divisore.

In figura 36 si osservano i risultati ottenuti, la funzionalità del circuito è preservata e il consumo, come nel caso dell’invertitore, è fissato dal ramo che polarizza il source comune.

Esistono, quindi, due possibili soluzioni in grado di realizzare il corretto funzionamento del circuito con lo stesso consumo e la scelta di uno schema o dell’altro risulterebbe essere arbitraria.

Per introdurre un criterio nella decisione, le proposte studiate verranno studiate nel capitolo successivo in termini di transitorio, per valutare i ritardi introdotti da ciascuna di esse.

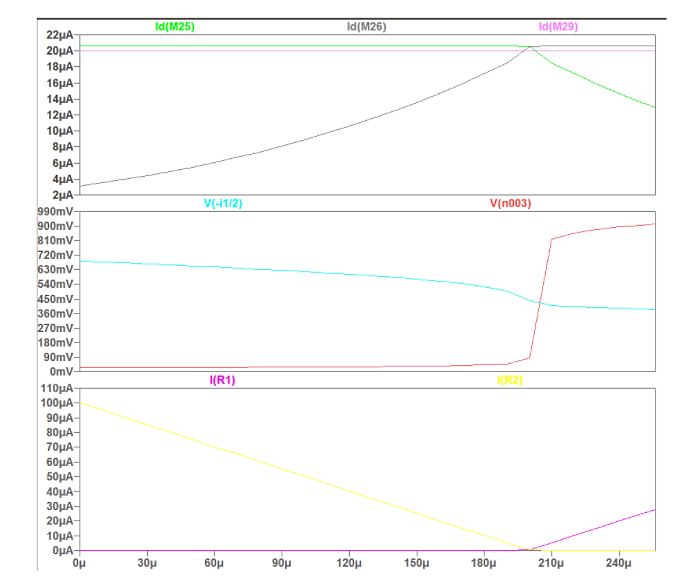


Figura 36: risultati relativi all'implementazione a source comune

# Capitolo 6

## Risultati simulazione transitorio

Nelle simulazioni di questo capitolo sono state considerate correnti di bitline I1 e I2 variabili nel tempo, con il seguente andamento:

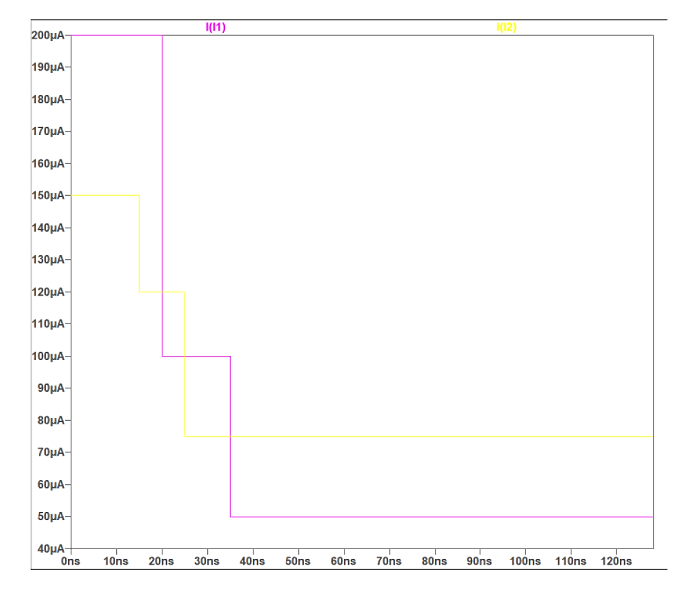


Figura 37: andamento delle correnti di bitline nel tempo

I valori scelti sono casuali, ma l’andamento descritto nel tempo è tipicamente realistico, poiché si ha un profilo decrescente.

### 6.1 Schema con invertitore

L’implementazione inizialmente considerata prevede una limitazione in corrente dell’invertitore pari a 30 µA:



Figura 38: andamento delle correnti con un consumo pari a 30 µA

Le uscite attese sono le forme d’onda azzurra e rossa, nei due grafici di figura 38, che per valori negativi devono essere pari a zero. Si osserva, pertanto, la correttezza delle uscite, con un ritardo associato pari a massimo 7 ns. Esso non è accettabile, in quanto a volte l’uscita si stabilizza al valore corretto quando l’uscita ideale è già cambiata.

Nei seguenti grafici si mostrerà l’andamento delle correnti con una maggiore limitazione in corrente, prima a 20 µA e poi a 10 µA:



Figura 39: andamento delle correnti con un consumo pari a 20 µA

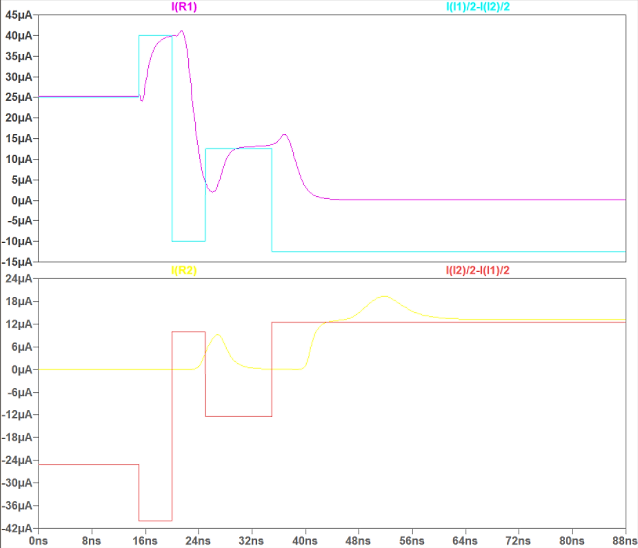


Figura 40: andamento delle correnti con un consumo pari a 10 µA

Con la riduzione del consumo, il ritardo non viene influenzato ed esso rimane invariato, però si osserva la presenza di spike con un’intensità crescente, che sono responsabili di errori nell’uscita.

### 6.2 Schema con source comune

Inizialmente si considera una polarizzazione di 30 µA:

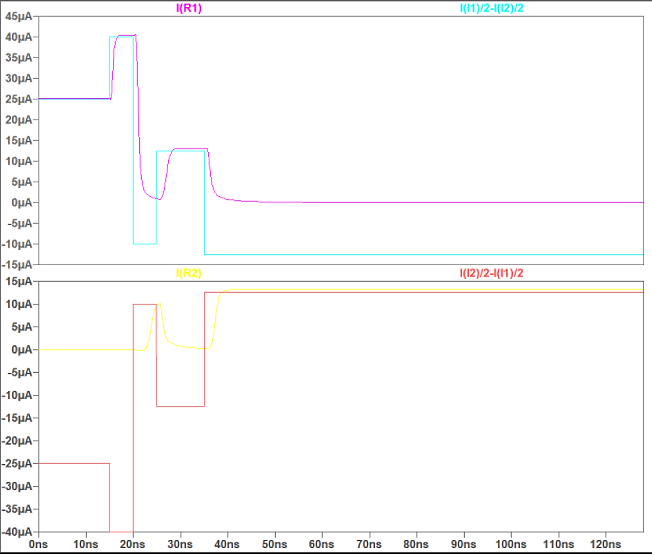


Figura 41: andamento delle correnti con un consumo pari a 30 µA

L’andamento ottenuto è decisamente migliore di quello visto per l’invertitore: il ritardo risulta essere più contenuto, con un massimo di 3 ns, e le forme d’onda sono più precise.

Si procede, dunque, con la riduzione del consumo:

* A 20 µA la situazione si mantiene invariata rispetto al caso di 30 µA,
* A 10 µA si ha l’introduzione di uno spike che causa errori nel valore d’uscita.

Complessivamente, risulta più opportuno introdurre la variante a common source come controllo perché il ritardo e la precisione del risultato sono migliori rispetto all’implementazione con l’invertitore.



Figura 42: andamento delle correnti con un consumo pari a 20 µA



Figura 43: andamento delle correnti con un consumo pari a 10 µA

### 6.3 Schema con pmos

È stata studiata un’ulteriore implementazione, basata sull’utilizzo di un unico pmos connesso a diodo come controllo: nel momento in cui la tensione del ramo del divisore è bassa, esso si accende e fornisce un percorso diretto alla tensione di alimentazione. Con questo schema, si elimina completamente il consumo dovuto al controllo, perché il pmos è connesso in serie al ramo del divisore.

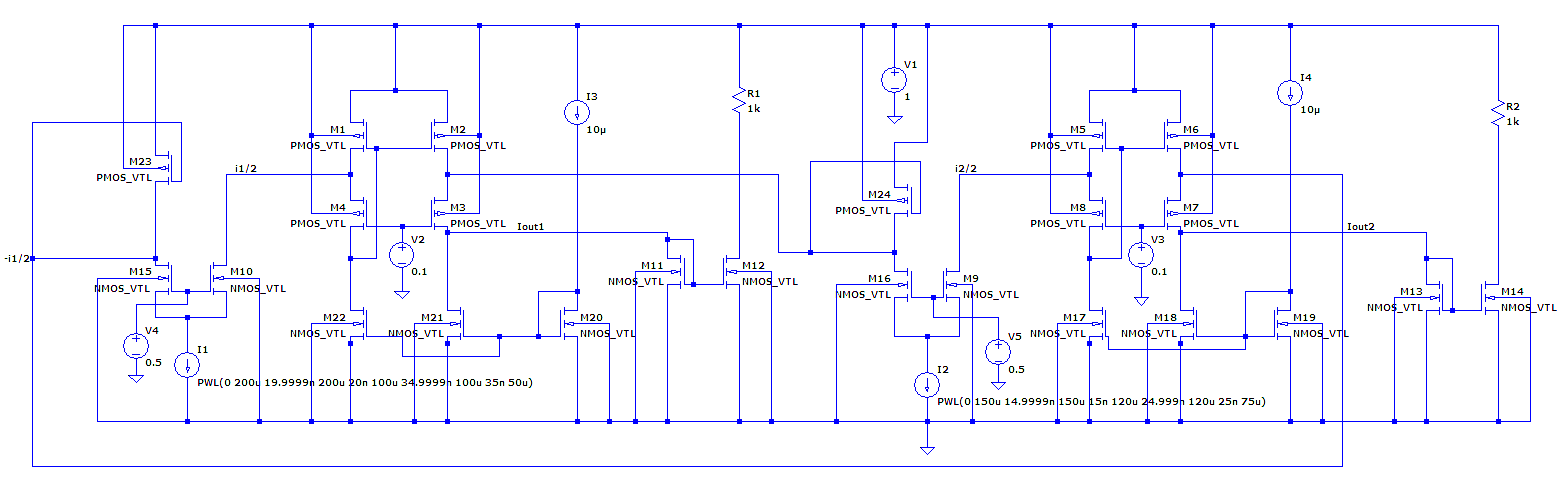


Figura 44: implementazione con soli pmos

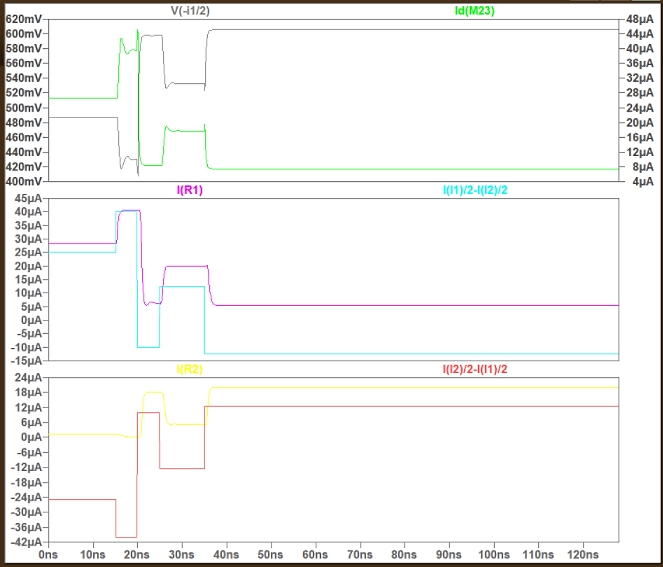


Figura 45: grafici associati allo schema di figura 44

Il profilo delle correnti di uscita risulta corretto, ma i valori specifici no, come osservato nei grafici in basso e centrale di figura 45. La ragione di tali errori si ritrova nel grafico in alto: sono mostrate la tensione in ingresso al pmos e la corrente del pmos, quest’ultima è diversa da zero quando esso dovrebbe essere spento: infatti, il contributo di Ioff, per i MOS di questo processo, è molto importante.

Per far fronte a questo inconveniente, si introduce un pmos con una tensione di soglia più alta, 0.5 V rispetto all’attuale 0.3 V, che permette di ridurre il contributo indesiderato:

Immagine che contiene diagramma, linea, Piano, schematico

Il contenuto generato dall'IA potrebbe non essere corretto.

Figura 46: implementazione con pmos M16 ed M24 a soglia più alta

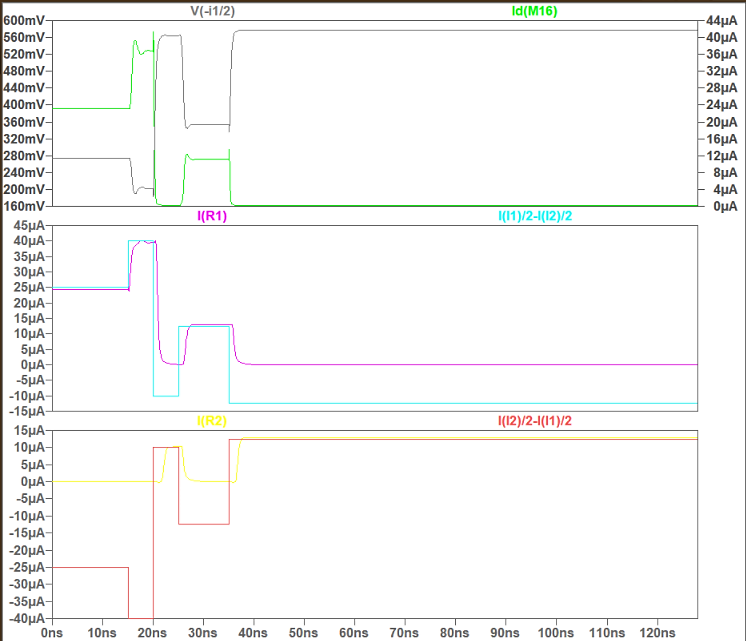


Figura 47: risultati riferiti allo schema di figura 46

Si osservano, in figura 47, le uscite e si effettuano le seguenti considerazioni:

1. Si ottiene un’alta precisione del risultato,
2. Il ritardo massimo associato è di 2 ns,
3. Il consumo aggiuntivo dovuto al controllo è azzerato.

La soluzione con i pmos risulta essere la più vantaggiosa rispetto alle proposte trattate, permettendo di ottenere un consumo totale dovuto solamente ai generatori di corrente presenti nel sottrattore.

Nel capitolo 2 si è parlato di correttezza del risultato della conversione attuata dall’ADC, in quanto la differenza fra le cariche è equivalente alla carica ottenuta mediante differenza. È possibile verificarlo nel seguente grafico:



Figura 48: uscite e ingressi del sottrattore

LTspice fornisce la possibilità di calcolare il valore medio di una grandezza in un range temporale a scelta: si sceglie l’intervallo complessivo di 70 ns, e si calcolano i valori medi riportati nella seguente tabella:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | I(R1) | I(R2) |  |  |
| Average | 10.275 µA | 6.7449 µA | 51.786 µA | 48.75 µA |
| Q | 0.71925 pC | 0.472143 pC | 3.625 pC | 3.4125 pC |

Effettuando la differenza di carica fra e si ottiene (3.625 - 3.4125) pC = 0.2125 pC che è circa uguale alla carica di I(R1) – I(R2) = (0.71925 – 0.472143) pC = 0.247107 pC. La carica attesa è leggermente più alta poiché le uscite non sono perfettamente uguali a quelle ideali, ma si ha un errore massimo pari a 0.5 µA. L’errore assoluto è pari a 0.0346 pC, quello relativo al 16.28 %.

Si verifica il medesimo aspetto con un’altra configurazione degli ingressi:

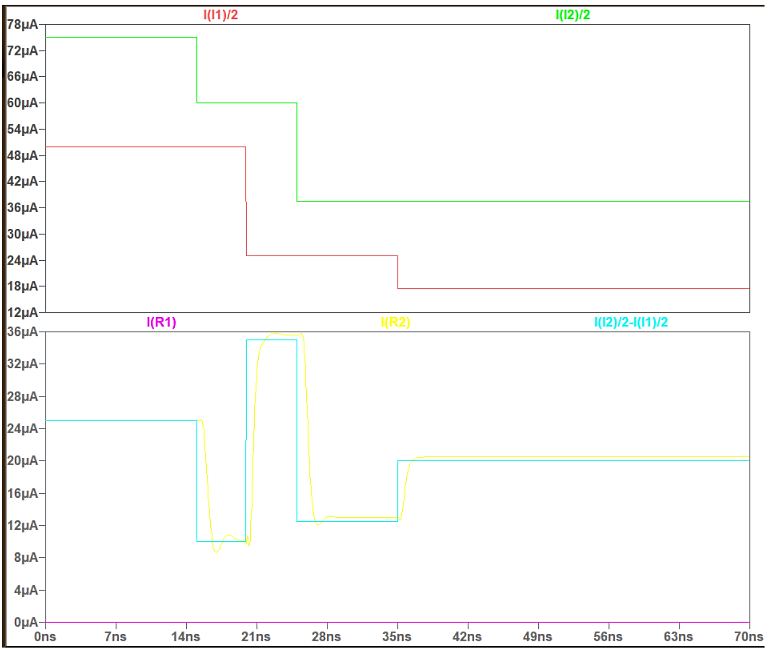


Figura 49: uscite e ingressi del sottrattore in una nuova configurazione degli ingressi

|  |  |  |  |
| --- | --- | --- | --- |
|  | I(R1) | I(R2) | - |
| Average | 0 | 20.931 µA | 20.357 µA |
| Q | 0 | 1.46517 pC | 1.42499 pC |

Si ottiene un errore assoluto di 0.0402 pC, ovvero un errore relativo del 2.82 %. Si simula nuovamente con un’ultima configurazione:

Immagine che contiene testo, schermata, diagramma, linea

Il contenuto generato dall'IA potrebbe non essere corretto.

Figura 50: uscite per nuova configurazione

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | I(R1) | I(R2) |  |  |
| Average | 3.7604 µA | 12.568 µA | 28.393 µA | 36.785 µA |
| Q | 0.26323 pC | 0.87976 pC | 1.98751 pC | 2.57495 pC |

La differenza di carica fra e è pari a (2.57495 – 1.98751) pC = 0.58744 pC che è circa uguale alla carica di I(R2) – I(R1) = (0.87976 – 0.26323) pC = 0.61653 pC. L’errore assoluto è pari a 0.0291 pC, l’errore relativo al 4.95 %.

L’errore ottenuto risulta compreso nel range [0.03 pC, 0.04 pC].

# Capitolo 7

## Considerazioni sull’ADC

### 7.1 Panoramica

Nella realizzazione di un sistema IMC è necessario prestare particolare attenzione ai circuiti periferici, nello specifico ai convertitori che interfacciano la crossbar con il mondo digitale, poiché responsabili della maggior parte del consumo di energia, con latenza e area totali associati fortemente dominati da questi. Ponendo il focus sull’uscita, ovvero sull’ADC, la soluzione più semplice risulta quella con ingresso in tensione, che richiede una conversione da corrente a tensione, impiegando un condensatore di grandi dimensioni per l'integrazione, che genera ulteriori problematiche. [9]

Le *Bitline* rappresentano gli ingressi dell’ADC, il quale quantizza i valori ricevuti per poi digitalizzarli. Con una precisione multi-bit dell'output, l'integrazione degli ADC nell'acceleratore di memoria è complessa e devono essere presi in considerazione diversi vincoli di progettazione, come mostrato in figura 51, affinché l’architettura basata su memorie resistive rimanga più vantaggiosa delle altre. [8]

Immagine che contiene testo, diagramma, Carattere, schermata

Il contenuto generato dall'IA potrebbe non essere corretto.

Figura 51: vincoli nel progetto di un ADC [8]

### 7.2 Architetture di tensione

In letteratura, le tre principali architetture di ADC impiegate negli acceleratori sono i convertitori Flash, SAR (Registro ad Approssimazioni Successive) e a Integrazione Sequenziale (IS). Questi ADC offrono così velocità di campionamento adeguate, consumi energetici contenuti e un’efficiente occupazione dell’area di silicio. Per via di queste prestazioni, essi risultano idonei per le implementazioni AIMC. [8]

Immagine che contiene diagramma, linea, Piano, Disegno tecnico

Il contenuto generato dall'IA potrebbe non essere corretto.

Figura 52: implementazioni dell'ADC (a) Flash (b) SAR (c) IS [8]

Il risultato dell’operazione di MVM si presenta come un segnale di tensione o di corrente, a seconda dell’architettura utilizzata. Nel caso di rilevamento di corrente, è possibile inserire un integratore ad anello chiuso all'ingresso dell'ADC, il quale agisce da morsetto sulla linea di sommatoria e realizza la conversione corrente-tensione. In alternativa, per ottimizzare sia il consumo energetico sia l’occupazione dell’area di silicio, il segnale sulla linea di sommatoria può essere trasformato da corrente a carica mediante un condensatore di campionamento.

Affinché le dimensioni dell’ADC si adattino alla struttura della memoria e si eviti un consumo superfluo di area di silicio, la sua larghezza, così come quella dell’integratore se presente, dovrebbe rientrare nel passo della cella di memoria. Tuttavia, considerando che in letteratura le dimensioni delle celle di memoria risultano già inferiori al micrometro, questo vincolo impatta significativamente sulla progettazione dell’ADC. Se la larghezza dell'interfaccia analogica non può essere ridotta fino al passo della memoria, quest'ultima deve essere automaticamente aumentata.

L'accuratezza della DNN durante l'inferenza dipende dalla risoluzione degli operandi, aspetto che influisce direttamente sulla progettazione dell'ADC. Nei convertitori Flash, il risultato dell'MVM viene confrontato simultaneamente con 2n-1 riferimenti, pertanto l'area occupata aumenta esponenzialmente con il numero di bit di uscita n. Una relazione analoga si osserva anche nei convertitori SAR, dove l'area del DAC capacitivo è determinata dal numero di bit.

L'occupazione di area dell'ADC è influenzata anche da altri aspetti di progettazione: ad esempio, nei convertitori Flash, vengono utilizzati condensatori che occupano una grande area per il campionamento dell'offset e la sottrazione in ciascun comparatore, contribuendo a ridurre l'errore di non linearità (INL). Alcune soluzioni alternative per minimizzare l'area occupata includono la tecnica del doppio campionamento correlato, che prevede l'inversione degli ingressi del comparatore, a scapito di una riduzione della velocità di conversione, e la calibrazione dell'offset, in cui la polarizzazione del body dei dispositivi di ingresso dei comparatori viene regolata all'accensione.

Un'altra strategia per ottimizzare l'uso dell'area consiste nel condividere un singolo ADC tra più colonne tramite multiplexing temporale. Tuttavia, affinché il throughput della memoria rimanga invariato, la velocità di campionamento dell'ADC deve superare 1 GS/s, limitando la condivisione del convertitore a poche decine di colonne. Tra le architetture disponibili, il convertitore Flash risulta l'opzione più adatta a questa configurazione.

A differenza dei convertitori Flash e SAR, negli ADC IS l'area non dipende dalla risoluzione in bit dell'uscita ed è significativamente ridotta grazie a una tecnica di condivisione della carica basata su un condensatore unitario CREF ​e un condensatore di campionamento CSC, come mostrati in figura 48. La conversione si basa sul confronto tra VRBL e una tensione di riferimento intermedia VC per determinare l’MSB. Successivamente, un gradino di tensione costante, corrispondente al bit meno significativo LSB, viene aggiunto o sottratto alla tensione campionata VIN​ fino a raggiungere il valore medio. Il tempo di conversione dell'ADC IS dipende dalla distanza di VY, ricevuta in ingresso,​ rispetto alla tensione di riferimento, e se l’acceleratore richiede la sincronizzazione di tutti gli ADC, il throughput complessivo è vincolato dal convertitore con il tempo di conversione più lungo.

Anche la distribuzione statistica del segnale VY​ incide sulle prestazioni dell’acceleratore. Nei modelli DNN, l’output della MVM segue spesso una distribuzione normale. Se il segnale VY presenta una deviazione standard ridotta, l'ADC IS risulta vantaggioso per le implementazioni AIMC, poiché garantisce una velocità di conversione superiore a quella degli ADC SAR, oltre a un minore consumo di energia e una minore occupazione di area. Tuttavia, quando VY​ presenta una grande deviazione standard, la conversione A/D richiede numerosi passaggi, rendendo questo ADC sequenziale un potenziale collo di bottiglia per le prestazioni dell'acceleratore. [8]

### 7.3 Architettura basata su CCO

Gli ADC di tensione appena trattati non rappresentano l’unica possibile implementazione alle esigenze richieste, ma esistono anche convertitori di corrente basati sul tempo, che offrono vantaggi in termini di efficienza. Essi eliminano la necessità di avere cicli di conversione aggiuntivi e permettono di ottenere un maggior bilanciamento tra precisione e latenza. Inoltre, grazie all'assenza di condensatori di grande dimensione per l’integrazione della corrente e all'impiego prevalente di circuiti digitali, questi convertitori possono essere integrati con un’architettura che prevede un ADC per ciascuna colonna della matrice, permettendo di avere una latenza complessiva minima e rimuovendo l’esigenza di condividere una risorsa tra colonne adiacenti. [9]

Una possibile implementazione è quella dell’ADC basato su CCO, i cui diversi stadi sono stati mostrati in figura 9, nel capitolo 1. Il primo stadio è rappresentato dal CCO, oscillatore controllato in corrente, mentre il secondo stadio è associato al ripple counter, contatore asincrono. Una possibile realizzazione si osserva in figura 53, implementazione adottata all’interno del core HERMES, dove è presente uno stadio aggiuntivo precedente all’oscillatore, che regola la tensione sulle Bitline e cattura l’informazione sulla polarità della corrente nel segnale D. In seguito a questo passo, il CCO genera un segnale proporzionale alla corrente di Bitline, codificato nel tempo. L'oscillazionedel segnale generato viene controllata attraverso due condensatori, C1 e C2, che vengonocaricati o scaricati alternativamente, finché una delle loro tensioni VC1 o VC2 raggiunge la tensione di soglia Vth dellacoppia di inverter ad accoppiamento incrociato collegata. La sua commutazione attiva anche il segnale di stato di latch Ae quindi digitalizza, in ultima analisi, il flusso di una quantità fissa di Qunit di carica nel circuito. In base al segnale A, viene attivata la prima o la seconda delle due parti simmetriche nell'oscillatore, permettendo al lato corretto di scattare quando necessario. Inoltre, in base alla polarità della corrente in ingresso HBL, il segnale Dcontrolla se la scarica dei condensatori di integrazione C1 e C2 raggiungela soglia dei pMOS P1 o P2, oppure se la carica avviene fino a quandogli nMOS N1 o N2 si attivano*.* Infine, il segnale oscillante Aviene inoltrato all'ultimo stadio dell'ADC, che è il ripple counter a 12 bit, che funge da integratore per le informazioni in corrente codificate nel tempo. Pertanto, la frequenza fCCO del segnale Aviene catturata dalle uscite positiva e negativa del contatore, ADCP e ADCN, che vengono incrementate secondo una velocità proporzionale alla corrente iHBL. [9]

Immagine che contiene testo, diagramma, mappa, Piano

Il contenuto generato dall'IA potrebbe non essere corretto.

Figura 53: Schematico dell'ADC all'interno del core HERMES. (a) Stadio di regolazione, (b) CCO, (c) Ripple Counter [9]

Come precedentemente accennato, i vantaggi introdotti sono notevoli, tuttavia è presente una problematica legata alla linearità della frequenza di uscita fCCO, cherisulta essere limitata per alte correnti di ingresso. Tale aspetto è dovuto al ritardo di gate costante, tdelay, pari alla distanza temporale fra l'istante in cui il latch si sposta verso l'alto e l'istante in cui l'integrazione di corrente procede sul secondo condensatore. Il tempo tdelay viene aggiunto all'inverso della frequenza di uscita dell'oscillatore*,* che rappresenta il periodo di tempo TCCO, interferendo così con la relazione lineare tra la frequenza fCCO e la corrente iHBL [9]:

Come soluzioni proposte in letteratura, si hanno la semplice restrizione al funzionamento a bassa frequenza, dove il ritardo è di minor impatto, l'estesa post-elaborazione digitale attraverso l’utilizzo di tabelle di ricerca, l’uso di filtri digitali o di altre strutture di feedback. All’interno del sistema IMC, queste soluzioni non sarebbero ideali, in quanto comportano limiti significativi per area o latenza. Inoltre, le operazioni di post-elaborazione, volte a compensare tale non linearità, funzionano solo per le misure di corrente continua e fallirebbero nel momento in cui correnti variabili nel tempo, che sono integrate in un periodo di tempo, venissero prese in considerazione.

Si osserva una possibile soluzione nella stessa figura 53, tracciata in blu: si implementa uno specchio di corrente che permette al latch di commutare con un anticipo, che può essere regolato in modo tale da compensare il ritardo introdotto.

Nell'ultimo stadio dell'ADC, il segnale oscillante Aè integrato nel dominio digitale utilizzando un ripple counter. Con la selezione dell'apposito flip-flop D che riceve A, è possibile rendere variabile la dimensione dell'incremento del contatore. Ciò consente l'esecuzione di operazioni di shift e somma all'interno dell'ADC con un overhead minimo, evitando l’introduzione di sommatori multi-bit dedicati. [9]

### 7.4 Caso studiato

È stata descritta l’implementazione dell’ADC basata su CCO, che rappresenta il miglior compromesso fra i requisiti desiderati. La proposta trattata in questa tesi, ovvero l’introduzione di un sottrattore fra due correnti di Bitline per dimezzare la durata complessiva di lettura, richiede di effettuare nuove considerazioni sull’ADC da implementare:

* la versione ultimata del sottrattore prevede due uscite, I(R1) e I(R2), rispettivamente pari a e , le quali arrivano in ingresso all’ADC e vengono connesse nella seguente modalità:

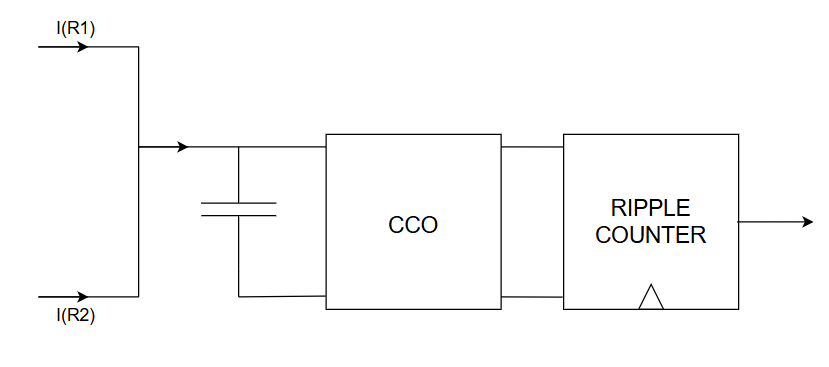


Figura 54: schema di connessione fra le uscite del sottrattore e l'ADC

Nella pratica solo un percorso di corrente è attivo, poiché quando la prima uscita è positiva, l’altra è nulla e viceversa.

* L’attivazione di un percorso o dell’altro deve essere tracciata: il ripple counter a valle deve essere capace di contare in maniera crescente per I(R1) positiva e in maniera decrescente per I(R2) positiva. È necessario, quindi, trovare un segnale in grado di fornire tale indicazione.
* Il segnale di partenza individuato è la tensione sul ramo in cui scorre :

Immagine che contiene testo, diagramma, Piano, linea

Il contenuto generato dall'IA potrebbe non essere corretto.

Figura 55: schematico di figura 46 con modifica aggiunta

Tale tensione viene portata in ingresso ad un invertitore logico (M25-M26), in modo tale da ottenere l’andamento di figura 56: per intervalli in cui l’uscita I(R1) è positiva, la tensione è bassa ma non nulla, altrimenti, per I(R2) positiva, è alta ma non pari a 1. Questa tensione viene pertanto invertita, in modo tale da avere come unici valori di uscita 0 e 1. Attraverso V(out), è possibile ora comandare il ripple counter, affinché conti verso l’alto o verso il basso.



Figura 56: andamento delle correnti e delle tensioni nel tempo

Si ottengono, quindi, V(out) = 1 per I(R1) positiva e V(out) = 0 per I(R2) positiva.

* Grazie al sottrattore, si hanno valori di corrente dimezzati in uscita, pertanto si permette all’ADC di rimanere nella regione più lineare della caratteristica ingresso-uscita. In questo modo, gli errori associati all’accuratezza del risultato sono ridotti,
* Attraverso il segnale V(out) si realizza anche un controllo sulle uscite I(R1) e I(R2): quando una corrente è nulla, lo è solo idealmente perché si ha un piccolo contributo associato e bisogna permettere unicamente il flusso all’altra corrente che non è nulla. Per realizzare tale aspetto si introduce una logica di controllo alle uscite, nel seguente modo:

Immagine che contiene schermata, nero, design

Il contenuto generato dall'IA potrebbe non essere corretto.

Figura 57: logica introdotta per controllare le uscite

Il segnale V(out) è in ingresso ad un pmos, con il source connesso al gate di M11, e ad un nmos, con il drain connesso al gate di M13.

Se V(out) = 1, la I(R1) è positiva e in ingresso alla capacità di integrazione si deve ricevere solo tale corrente. In questa configurazione, il pmos controllato da V(out) risulta spento, mentre l’nmos controllato dalla stessa risulta acceso: la corrente I(R1) viene specchiata e raggiunge l’ingresso dell’ADC, mentre lo specchio che riguarda I(R2) viene spento poiché l’nmos che si accende va a massa.

Se V(out) = 0, la I(R2) è positiva e in ingresso alla capacità di integrazione si deve ricevere solo tale corrente. In questa configurazione, il pmos controllato da V(out) risulta acceso, mentre l’nmos controllato dalla stessa risulta spento: la corrente I(R2) viene specchiata e raggiunge l’ingresso dell’ADC, mentre lo specchio che riguarda I(R1) viene spento poiché il pmos che si accende va a massa.

Con questa modalità è possibile attuare un controllo diretto sul flusso o meno di una corrente, garantendo così la digitalizzazione della sola carica richiesta.

### 7.5 Considerazioni finali

Nel capitolo 2 sono state effettuate considerazioni preliminari sull’introduzione del sottrattore, in termini di alcuni parametri:

|  |  |
| --- | --- |
| Throughput | *x(2 – ritardo associato al sottrattore)* |
| Consumo di potenza | *x(2 + sottrattore)* |
| Energia | *Maggiore o invariata* |
| Area | *x(2 + sottrattore)* |
| Throughput/Consumo | *Minore o invariato* |
| Throughput/Area | *Minore o invariato* |

Raggiunto il termine dello studio, è possibile affermare che l’introduzione del sottrattore fully-differential, con logica di controllo basata su un pMOS a tensione di soglia più elevata rispetto ai MOS del restante circuito, permette di ottenere un consumo associato pari a 4 µA e un ritardo complessivo pari a 2 ns, e rappresenta la miglior soluzione trattata all’interno di questo lavoro.

Lo schema studiato è pensato per essere introdotto a monte dell’ADC, in modo tale da raggiungere il valore desiderato alla fine di una fase di lettura. La sua presenza non influenza in maniera importante il circuito complessivo, poiché consumo e ritardo introdotti sono molto contenuti.

# 8. Conclusioni

In questa tesi è stata esplorata l’architettura di un sottrattore analogico, capace di eseguire la sottrazione fra le correnti di Bitline, che permetterebbe l’utilizzo di una singola fase di lettura, migliorando la performance in termini di Throughput. Lo scopo di questo progetto è stato analizzare l’implementazione circuitale del sottrattore in termini di funzionalità, consumo e ritardo, prestando particolare attenzione al consumo generato. Con l’introduzione di tale modifica architetturale, è stato necessario raddoppiare l’intera crossbar, portando al raddoppio del consumo complessivo, con il vantaggio di ottenere un raddoppio del throughput. Ulteriori consumi sono, quindi, da evitare per rendere tale soluzione più vantaggiosa rispetto a quella attualmente usata; pertanto, sono state effettuate modifiche in questa direzione. Nell’evoluzione di questo lavoro, infatti, la soluzione single-ended è stata sostituita da quella fully-differential, permettendo una riduzione del consumo del 99 %, generando però la necessità di introdurre una logica di controllo realizzata prima da un nmos comandato da un invertitore, poi da un nmos comandato da uno stadio a source comune e infine da un pmos a soglia più alta. Ai fini della riduzione del consumo, sono stati tenuti in considerazione sia il ritardo associato alla generazione del risultato, sia la precisione del risultato. È emerso un compromesso fra i tre parametri, sfavorendo il raggiungimento di una soluzione a consumo quasi nullo. Con le modifiche apportate, il ritardo massimo ottenuto è di 2 ns, che potrebbe essere eliminato se il convertitore a valle iniziasse a funzionare con lo stesso ritardo. È stato fornito anche un possibile interfacciamento fra le uscite e il convertitore analogico-digitale, che comprende un controllo sulle uscite e un controllo sul ripple counter.

In conclusione, lo studio di questa tesi ha permesso di raggiungere buoni risultati, creando la possibilità di intraprendere questa strada come valida soluzione al limite temporale imposto dall’attuale architettura. Inoltre, questa alternativa può essere applicata a qualunque tipologia di memoria resistiva, perché non sono presenti vincoli fra i due aspetti.

Per questi motivi, credo che sia possibile approfondire lo studio trattato in questa tesi, i vantaggi emersi sono notevoli, a fronte di svantaggi poco impattanti, che permettono di proseguire l’esplorazione in questa direzione.

# Riferimenti bibliografici

[1] Aguirre, F., Sebastian, A., Le Gallo, M. *et al.* Hardware implementation of memristor-based artificial neural networks. *Nat Commun* **15**, 1974 (2024). <https://doi.org/10.1038/s41467-024-45670-9>.

[2] Zambelli, Cristian & Navarro, Gabriele & Sousa, Veronique & Prejbeanu, Ioan & Perniola, Luca. (2017). Phase Change and Magnetic Memories for Solid-State Drive Applications. Proceedings of the IEEE. PP. 1-22. 10.1109/JPROC.2017.2710217.

[3] A. Sebastian, M. Le Gallo, G. W. Burr, S. Kim, M. Brightsky, and E. Eleftheriou, “Tutorial: Brain-inspired computing using phase-change memory devices”, Journal of Applied Physics, vol. 124, no. 11, 2018, ISSN: 10897550. DOI: 10.1063/1.5042413.

[4] S. W. Fong, C. M. Neumann and H. . -S. P. Wong, "Phase-Change Memory—Towards a Storage-Class Memory," in *IEEE Transactions on Electron Devices*, vol. 64, no. 11, pp. 4374-4385, Nov. 2017, doi: 10.1109/TED.2017.2746342.

[5] A. Athmanathan, M. Stanisavljevic, N. Papandreou, H. Pozidis and E. Eleftheriou, "Multilevel-Cell Phase-Change Memory: A Viable Technology," in *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 6, no. 1, pp. 87-100, March 2016, doi: 10.1109/JETCAS.2016.2528598.

[6] Antolini A, Franchi Scarselli E, Gnudi A, Carissimi M, Pasotti M, Romele P, Canegallo R. Characterization and Programming Algorithm of Phase Change Memory Cells for Analog In-Memory Computing. *Materials*. 2021; 14(7):1624. https://doi.org/10.3390/ma14071624.

[7] A. Antolini *et al*., "Combined HW/SW Drift and Variability Mitigation for PCM-Based Analog In-Memory Computing for Neural Network Applications," in *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 13, no. 1, pp. 395-407, March 2023, doi: 10.1109/JETCAS.2023.3241750.

[8] M. Caselli, P. Debacker and A. Boni, "Memory Devices and A/D Interfaces: Design Tradeoffs in Mixed-Signal Accelerators for Machine Learning Applications," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 7, pp. 3084-3089, July 2022, doi: 10.1109/TCSII.2022.3174622.

[9] R. Khaddam-Aljameh *et al*., "HERMES-Core—A 1.59-TOPS/mm2 PCM on 14-nm CMOS In-Memory Compute Core Using 300-ps/LSB Linearized CCO-Based ADCs," in *IEEE Journal of Solid-State Circuits*, vol. 57, no. 4, pp. 1027-1038, April 2022, doi: 10.1109/JSSC.2022.3140414.

# Ringraziamenti

L’Università è stata la parte più importante di questi ultimi anni e raggiungere la fine di questo percorso impegnativo lascia, al contempo, felicità e tristezza.

La parte finale è stata caratterizzata dall’attività di tesi, resa possibile dalla professoressa Eleonora Franchi Scarselli, che ringrazio per la sua costante disponibilità e per il suo incredibile supporto.

Questa attività mi ha permesso di sviluppare nuove conoscenze e nuove attitudini, e ciò non sarebbe stato possibile senza il continuo supporto ricevuto da Lorenzo, Alessio e Andrea, che ringrazio di cuore per tutto il tempo speso nei miei riguardi e per tutta la pazienza.

Raggiungere questa parte finale è stato difficile e per alcuni momenti quasi impossibile, ma nel complesso è risultato un percorso estremamente piacevole e, senza Cristina, Veronica, Kevin, Youness e Ivan, non penserei lo stesso. La loro costante presenza e il loro costante supporto mi ha permesso di affrontare il percorso con una leggerezza incredibile, che solo loro sanno fare, e per questo ringrazio infinitamente ognuno di loro.

Ringrazio Giacomo per il suo incredibile supporto e per tutti i momenti trascorsi assieme.

Ringrazio Fabiana che, nonostante la lontananza, è stata sempre presente e disponibile nelle difficoltà.

Ringrazio Inva per la sua costante disponibilità e il suo continuo supporto, ricevuto anche da lontano, per questi cinque anni.

Ringrazio Sofia per il prezioso supporto ricevuto in tutti questi anni, che mi ha permesso di credere in me stessa anche nei momenti di difficoltà.

Ringrazio mio fratello che, nonostante la differenza di età, è riuscito a comprendere e sopportare le mie emozioni contrastanti, fornendomi un continuo supporto.

Ringrazio, infine, i miei genitori, senza i quali nulla di tutto ciò sarebbe stato possibile, tutti i miei traguardi lo sono grazie a loro, e questa esperienza si conclude solo grazie al loro continuo supporto.