Università degli studi di Padova

DIPARTIMENTO DI TECNICA E GESTIONE DEI SISTEMI INDUSTRIALI

Corso di Laurea in Ingegneria Meccanica e Meccatronica

Tesi di Laurea

Impatto delle variazioni PVT nelle architetture di calcolatori digitali e analisi di soluzioni circuitali

Relatore: prof. Paolo Magnone

Laureando: Manzati Davide

Anno Accademico: 2015/2016

Indice

Introduzione:

Ai nostri tempi i microcontrollori risultano essere le architetture digitali più diffuse al mondo tanto che, si possono trovare in: computer, mouse, telefoni, elettrodomestici, auto, videogiochi, calcolatrici, orologi e in tantissimi altri oggetti. Fecero la loro comparsa negli anni settanta. La Texas Instruments, seguita successivamente da Intel, rilasciò sul mercato i primi microcontrollori stravolgendo le tecniche utilizzate fino a prima per l'implementazione di algoritmi. Infatti, i microcontrollori sono System on Chip (SoC) che a differenza dei microprocessori al loro interno hanno, oltre alla CPU, delle memorie per contenere il programma e i dati e, sono in grado di comandare qualsiasi tipo di periferica esterna svolgendo appositi programmi che ne permettano la gestione. Questo è stato il loro vantaggio principale: la flessibilità, ovvero la capacità di poter essere utilizzati in tutti i campi senza dover modificare il loro circuito interno.

Lo scopo di questa tesi è fornire al lettore una descrizione della composizione interna dei microprocessori, la componente principale dei microcontrollori, focalizzandosi sullo studio delle principali fonti di errore e degrado delle prestazioni rispetto alle caratteristiche nominali. Viene analizzato perciò come queste problematiche vengano affrontate dai produttori e, quindi, quali soluzioni sono state implementate sul chip per la loro rilevazione e gestione, in particolare quelle a livello di architettura. Tutto ciò viene fatto con lo scopo di poter rendere il dispositivo affidabile, trovare un giusto compromesso tra consumo e prestazioni e poter innalzare la percentuale di dispositivi che dopo la produzione soddisfano le specifiche di progetto.

La scelta di trattare questo argomento nasce principalmente da ciò che è lo studio dell'applicazione dei microcontrollori nel settore della telefonia mobile come mio interesse personale, arricchito ulteriormente da un'esperienza di programmazione alla quale ho partecipato: la Freescale Cup. Nel corso della competizione, che prevedeva la programmazione di un microcontrollore con un linguaggio scelto da me che fosse il più vicino possibile a quello macchina (ovvero il C), più volte mi sono interrogato su come il software venisse eseguito dall'hardware. Attraverso la trattazione e la stesura di questa tesi, non solo ho potuto trovare risposta al mio quesito, ma ho avuto l'opportunità di modificare il mio punto di vista: se prima, in mio giudizio, i microcontrollori erano scatole chiuse infallibili, attraverso lo studio di questi ho potuto apprendere la loro insita sensibilità alle variazioni esterne.

L'obiettivo finale di questa tesi è poter guardare con criticità ai microprocessori, potersi rendere conto a quanti problemi sono sottoposti e sottolineare la necessità che le operazioni che andranno a svolgere siano giustamente ottimizzate. Per farlo, verrà

illustrato il funzionamento generale di un microprocessore, verranno poi trattate le principali cause di variazione che colpiscono i transistori, infine verranno spiegate le tecnologie che mettono i processori nelle condizioni di evitare errori dovuti a problemi di hardware.

CAPITOLO 1

Il microprocessore

Il processore, o microprocessore, è un'architettura elettronica digitale, realizzato con la tecnologia dei circuiti integrati VLSI (Very Large Scale Integration), in grado di effettuare in modo autonomo operazioni aritmetiche e logiche secondo una successione preordinata di istruzioni, costituenti il programma esecutivo del microprocessore stesso.

1.1.ORGANIZZAZIONE CIRCUITALE DI UN PROCESSORE

L'organizzazione circuitale di un processore fa riferimento alle unità operative e alle loro interconnessioni che realizzano le specifiche architetturali.

Esempi di specifiche architetturali sono il repertorio delle istruzioni, il numero di bit per rappresentare i vari tipi di dati, i meccanismi di I/O e le tecniche di indirizzamento della memoria.

L'organizzazione dei processori è tra le più incisive: infatti, in genere, le case produttrici offrono famiglie di modelli che hanno in comune la stessa architettura, ma hanno differenti organizzazioni (si pensi ad esempio la famiglia di processori Intel composta da i3, i5, i7).

Mentre il processore i7 è il performante, l'i3 lo è in misura minore. Tuttavia, tutti e tre hanno la caratteristica di avere lo stesso Instruction Set (di cui si parlerà più avanti), per cui un programma scritto per l'i3 può essere eseguito anche su i5 e i7, ma con prestazioni diverse dovute a un diverso numero di core o a una memoria cache più capiente o a soluzioni quali l'Hyper Threading (una tecnologia sviluppata da Intel che è in grado di "raddoppiare" il numero di core in maniera "virtuale") (http://tomsblog.it/andreaferrario/2015/09/18/le-differenze-tra-core-i7-i5-i3-e-pentium-in-parole-semplici/),6 Agosto 2016 .

Nonostante il primo microprocessore fece la sua comparsa nel 1971 (Intel 4004), già nei primi anni 40 del 1900 sono state concepite le due principali filosofie che ancora ad oggi caratterizzano l'organizzazione intera di un processore:

- Organizzazione di von Neumann
- Organizzazione di Harvard

Successivamente fece la sua comparsa una delle organizzazioni ad oggi più utilizzata, la Pipeline.

Di seguito verranno illustrate le tre tipologie.

1.1.1.Organizzazione von Neumann

Questo tipo di organizzazione prende il suo nome da John von Neumann (1903-1957) [21].

E' un tipo di organizzazione per computer digitali programmabili a programma memorizzato dove le istruzioni del programma da eseguire e i dati del programma sono nello stesso spazio di memoria (https://it.wikipedia.org/wiki/Architettura_di_von_Neumann), 17 Agosto 2016.

La struttura è del tipo in Figura 1 e, da come si può notare, vi sono 2 blocchi principali: la CPU (central process unity) e la memoria (dove si hanno sia dati che istruzioni).

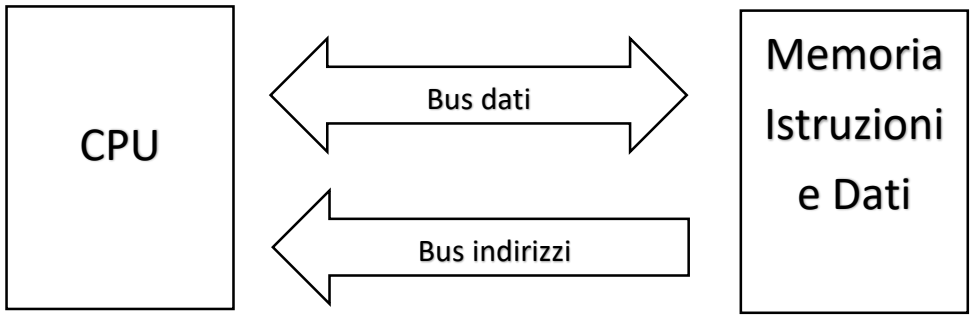

Figura 1: organizzazione di tipo von Neumann

I due blocchi sono tra loro connessi tramite i "Bus", dei quali in questa architettura se ne hanno due:

- Bus dati: Bidirezionale, cioè un dato contenuto in memoria può essere sia letto che scritto
- Bus indirizzi: Unidirezionale, dalla CPU alla memoria e permette di selezionare la locazione di memoria sul quale si vuole operare

Vi è in realtà un terzo bus non rappresentato, il bus controlli, che è un insieme di collegamenti il cui scopo è coordinare le attività del sistema

Il vantaggio principale di questo tipo di organizzazione è la semplicità circuitale e ne consegue un costo di fabbricazione minore.

Lo svantaggio principale è la sua lentezza perché istruzioni e dati condividono lo stesso canale ed in questo modo per eseguire un'istruzione che necessiti la lettura di un dato, prima si leggerà l'istruzione poi si leggerà il dato.

1.1.2.Organizzazione Harvard

L'organizzazione Harvard prende il nome dall'università che l'ha ideata [22].

Si distingue da quella di von Neumann poiché, a differenza di questa, i dati e le istruzioni risiedono in memorie differenti: la memoria dati e la memoria istruzioni.

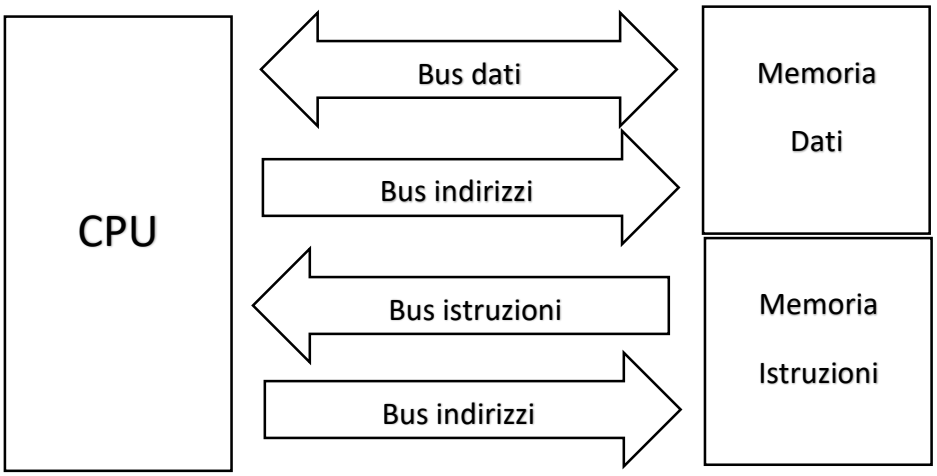

Figura 2: organizzazione di tipo Harvard

Da come si può notare, nella Figura 2, vi sono quattro bus:

- Bus dati, bidirezionale su cui transitano i dati che verranno scritti o letti dalla memoria
- Due Bus indirizzi, unidirezionali che permettono la scelta dell'indirizzo di memoria in cui sono collocati dati o le istruzioni
- Bus istruzioni, unidirezionale dove la CPU legge l'istruzione da eseguire

Anche in questo caso, non viene rappresentato il Bus controlli che ha la stessa funzione dell'organizzazione di von Neumann.

E' semplice notare come l'organizzazione Harvard sia più complessa dal punto di vista circuitale: la sua implementazione quindi risulta più costosa. Tuttavia, questa organizzazione introduce un minimo grado di parallelismo quando viene eseguito il codice, permettendo il raggiungimento di velocità più elevate nella sua esecuzione: questo ne ha permesso l'utilizzo in processori ad alte prestazioni.

1.1.3.Organizzazione con Pipeline

La Pipeline è un'organizzazione orientata al raggiungimento di elevate prestazioni. La sua idea di base è, a differenza delle altre, quella di svolgere simultaneamente le fasi di prelievo, decodifica e esecuzione di più istruzioni. Internamente al processore quello che avviene è rendere autonome le varie componenti così che ogni unità possa lavorare simultaneamente ad un'altra su un'altra istruzione.
Si può intuire subito che questa frammentazione delle unità porta con sé anche una complessità interna non indifferente che inevitabilmente va ad incidere sul costo del processore e questo è il motivo principale per cui l'implementazione di questo tipo di organizzazione ha avuto luogo solo in periodi recenti, quando la continua diminuzione dei costi di produzione l'ha permesso. Attualmente, i processori dei personal computer utilizzano tutti questa organizzazione: in particolare nelle architetture RISC, di cui si parlerà in seguito, ha trovato facile implementazione.

Nei casi più semplici l'organizzazione con pipeline suddivide lo svolgimento di una istruzione in fase di Fetch (F) e Decode and Execution (DE): nella fase di Fetch l'istruzione viene letta dalla memoria; nella fase di Decode l'istruzione appena letta viene convertita tramite l'unità di controllo in segnali di controllo, mentre vengono letti gli operandi contenuti nei registri (se l'istruzione prevede il loro utilizzo); infine nella fase di Execution viene eseguita l'istruzione dal processore.

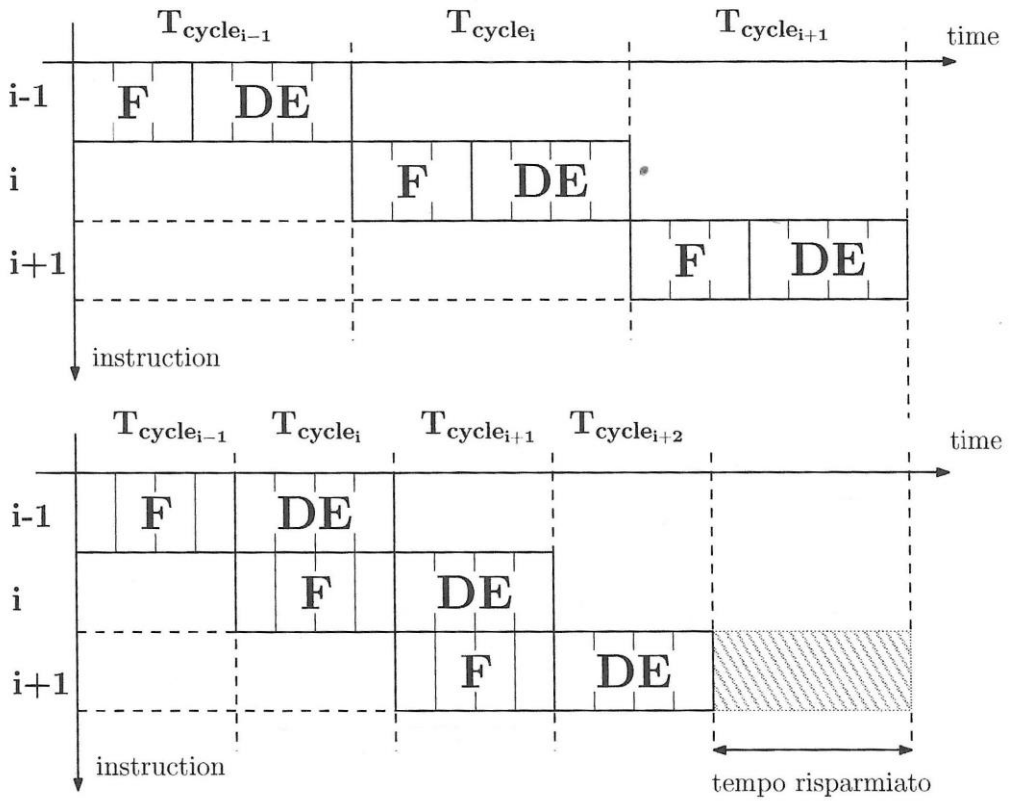

Figura 3: Temporizzazione di una sequenza di tre istruzioni in una organizzazione senza pipeline (sopra) e con una pipeline a 2 stadi (sotto) [22]

Nell'immagine sopra (Figura 3) è possibile osservare come l'organizzazione a pipeline velocizzi l'esecuzione di un codice: infatti, se nello schema in alto il processore per eseguire 3 istruzioni impiega 3 cicli macchina, nello schema sotto (organizzazione con pipeline) il processore impiega un tempo minore rispetto al primo, di 5/7 di ciclo macchina. Occorre però osservare come la fase di Fetch nel secondo caso sia stata allungata affinché ci fosse sincronismo tra le due fasi. In particolare, se escludiamo il primo e l'ultimo ciclo, durante il generico i-esimo ciclo viene eseguita la Fetch dell'i-esima istruzione in parallelo alla fase di Decode and Execution dell'istruzione i-1.

Come già detto, quella appena rappresentata è l'organizzazione con pipeline più semplice che si possa realizzare: tuttavia, è possibile ottenere una maggiore velocità con le Pipeline a molti stadi (cioè Pipeline che frammentano ulteriormente le unità della CPU per ottenere un maggior numero di fasi).
A titolo di esempio vi sono Pipeline a tre fasi (Fetch, Decode, Execution), dove le fasi di decodifica ed esecuzione sono separate (utilizzata nei processori ARM), e Pipeline a cinque fasi (Fetch, Decode, Read, Execution, Write), dove le fasi di lettura e scrittura dei registri sono slegate rispettivamente dalla fase di Decode e di Execution.

1.2.ARCHITETTURA: REPERTORIO DI ISTRUZIONI

L'architettura di un processore fa riferimento agli attributi di un sistema visibili al programmatore, nello specifico quelli che hanno un impatto diretto sull'esecuzione logica di un programma.
Esempi di attributi che caratterizzano l'architettura sono il repertorio istruzioni, il numero di bit necessari per rappresentare un tipo di dato, i meccanismi di I/O e le tecniche di indirizzamento della memoria.

In questo capitolo parleremo del repertorio istruzioni, che rappresenta una delle caratteristiche principali che va a influenzare l'efficienza del processore tale da rendere la sua programmazione semplice o ottenere delle alte prestazioni.

I principali aspetti che si devono valutare per poter discutere dell'efficacia di una architettura sono [22]:

- Struttura delle istruzioni
- Modi di indirizzamento dei dati
- Funzioni realizzate dalle istruzioni

Per quello che riguarda il repertorio istruzioni, le caratteristiche di maggior interesse sono:

- Simmetria
- Ortogonalità
- Compattezza
- Flessibilità
- Velocità di esecuzione
- Facilità di debugging
- Costo

Per **simmetria** si indica il grado di uniformità dei modi di indirizzamento tra le diverse istruzioni: più un repertorio istruzioni è simmetrico e meno differenze vi sono nei modi di indirizzamento applicati. Il vantaggio che ne trae il programmatore è la possibilità di applicare a tutti i tipi di dati più operazioni, oltre il fatto che le istruzioni, rappresentate tramite codici binari, sono uniformi.

Per **ortogonalità** si indica la regolarità nella struttura della parola che rappresenta un'istruzione (detto anche Instruction Word , IW). Un repertorio istruzioni ortogonale ha IW della stessa lunghezza, in cui le dimensioni dei campi associati alle specifiche funzioni sono sempre le stesse. Si ottiene un formato delle istruzioni unico e costante.

La **compattezza** del repertorio istruzioni indica la dimensione del codice macchina che viene generate nella fase di assemblaggio di un programma dato.

La **flessibilità** indica la facilità con la quale è possibile modificare l'insieme delle istruzioni: ad esempio, se si desidera aggiungere nuove istruzioni in una successiva versione del processore.

La **velocità di esecuzione** è un parametro fondamentale che è in stretta relazione con il **costo** del processore, dove con costo si intende il costo del processore che utilizza quella architettura, e questo dipende dalla tecnologia impiegata e dalla complessità dei circuiti, quindi sull'area di silicio richiesta e il volume di produzione atteso.

La **facilità di debugging** indica la leggibilità del codice scritto o compilato. È associata alla maggiore o minore somiglianza del codice macchina con un linguaggio di alto livello: infatti, alcune architetture forniscono, ad esempio, istruzioni come cicli for, mentre altre li possono fornire solamente ordinando il codice con più istruzioni che lo andranno a comporre.

Attualmente, sulla base dei parametri spiegati sopra, si è soliti suddividere le architetture in due parti:

- Architetture RISC
- Architetture CISC

L'architettura **RISC** è l'acronimo di Reduced Instrucion Set Computer: in questo tipo di architetture le istruzioni sono solitamente poche e il più semplici possibile. Sono dette istruzioni semplici per il formato e le funzionalità delle stesse. Si tratta di istruzioni che hanno tutte le stesse dimensioni e la stessa architettura.

L'architettura **CISC** è l'acronimo di Complexed Instruction Set Computer: in questo tipo di architetture le istruzioni sono molte (anche centinaia) e in genere piuttosto complesse. L'esecuzione delle istruzioni richiede quasi sempre più cicli macchina.

Il vantaggio principale di questa architettura è la riduzione del "gap" che si ha tra il linguaggio macchina e il linguaggio ad alto livello, con una conseguenza diretta sulla pesantezza del codice.
Sembrerebbe facile dire che l'architettura CISC sia migliore di quella RISC: tuttavia, la semplicità di quest'ultima ha portato alla possibilità di poter "modificare" più facilmente l'organizzazione interna dei processori, conducendo, ad esempio, alla pipeline. Attualmente, quest'ultima è l'organizzazione maggiormente utilizzata.

1.3.COMPONENTI DI UNA CPU

La CPU è composta al suo interno da diverse componenti, le principali sono [22]:

- L'unità di controllo
- La ALU (Arithmetic Logic Unit)
- Registri

In Figura 4 si riporta un esempio di processore a 8 bit semplificato. Osservando la figura, è possibile contare 9 blocchetti che possono comunicare tra loro attraverso il Bus. È facile individuare le due componenti che sono l'unità di controllo e la ALU, i rimanenti sono tutti registri.

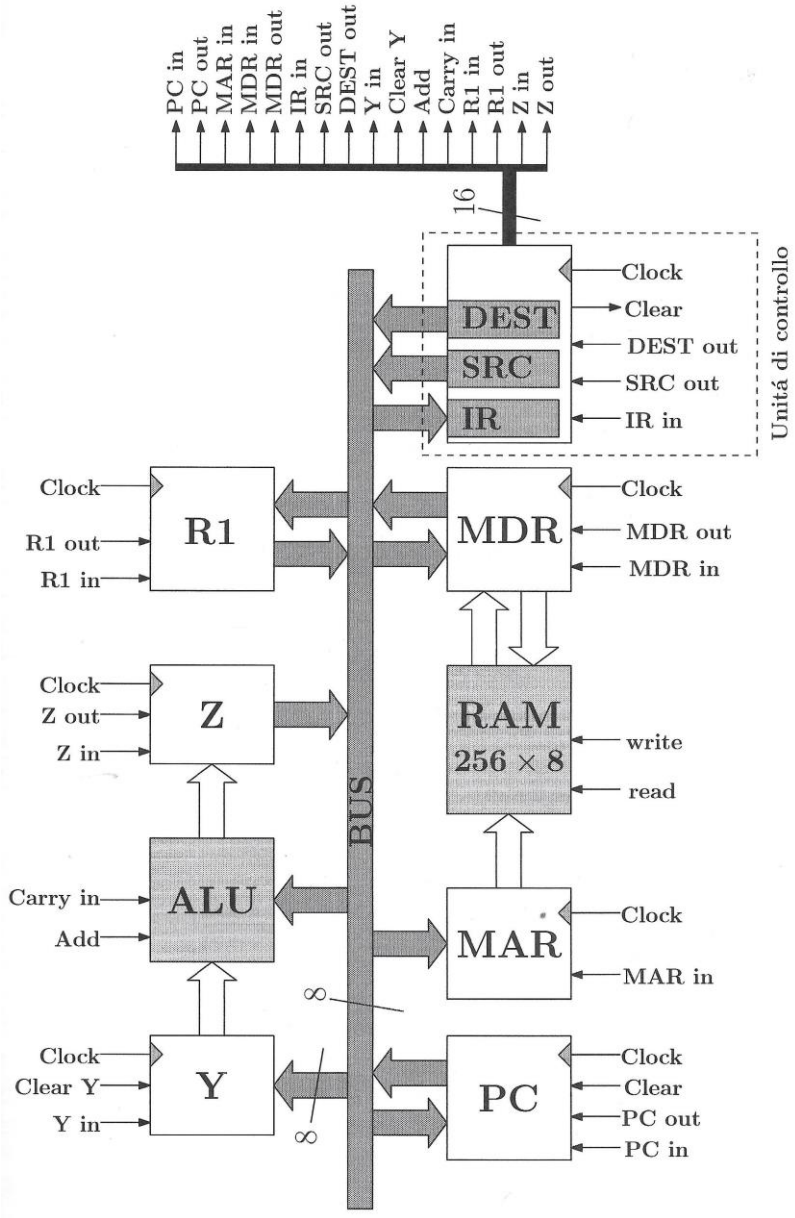

Figura 4: Un ipotetico processore a 8 bit [22]

1.3.1.L'unità di controllo

E' il blocco più complesso della CPU e genera i segnali di controllo che permettono di eseguire le istruzioni, fornisce il ciclo di clock e ha il compito di decodificare il codice che viene scritto dall'utente in linguaggio macchina, cosicché possa essere eseguito.

Come è possibile osservare dalla figura, l'unità di controllo al suo interno comprende altri tre registri che sono:

- IR o Instruction Register, ovvero il registro che contiene il codice binario che identifica un'istruzione (chiamato anche "Operation Code" o OpCode), cioè la serie di '1' e '0' che verranno inviati nel bus e andranno ad attivare o disattivare le altre componenti della CPU per eseguire quell'istruzione.
- SRC è un registro che può essere usato o per accogliere un valore numerico che non è soggetto a variazioni durante l'esecuzione del codice, oppure per contenere un valore numerico che fa riferimento ad un indirizzo di memoria.
- DEST è usato per accogliere un indirizzo o il codice di un registro nel quale depositare il risultato di un'operazione.

Il circuito di controllo ha il compito, dall'analisi dell'OpCode, di ricavare la sequenza di segnali di controllo necessaria per due cose:

- Svolgimento della fase di Fetch e aggiornamento del PC
- Esecuzione dell'istruzione corrente

Per ottenere ciò vi sono due approcci diversi [22]:

- Controllo microprogrammato
- Controllo cablato

I prodotti più recenti utilizzano il controllo cablato al posto del microprogrammato.

Il **controllo microprogrammato** (Figura 5) viene eseguito da una unità di controllo che, strutturalmente, si compone nel suo interno di un'ulteriore CPU, ma più semplice, quindi con una sua memoria una sua ALU e un suo PC: la CPU in questione viene indicata come "microcode engine". Il programmatore ha a disposizione delle istruzioni chiamate "macro-istruzioni" ed ognuna di queste corrisponde ad un microprogramma o microcodice contenuto nella memoria ROM.

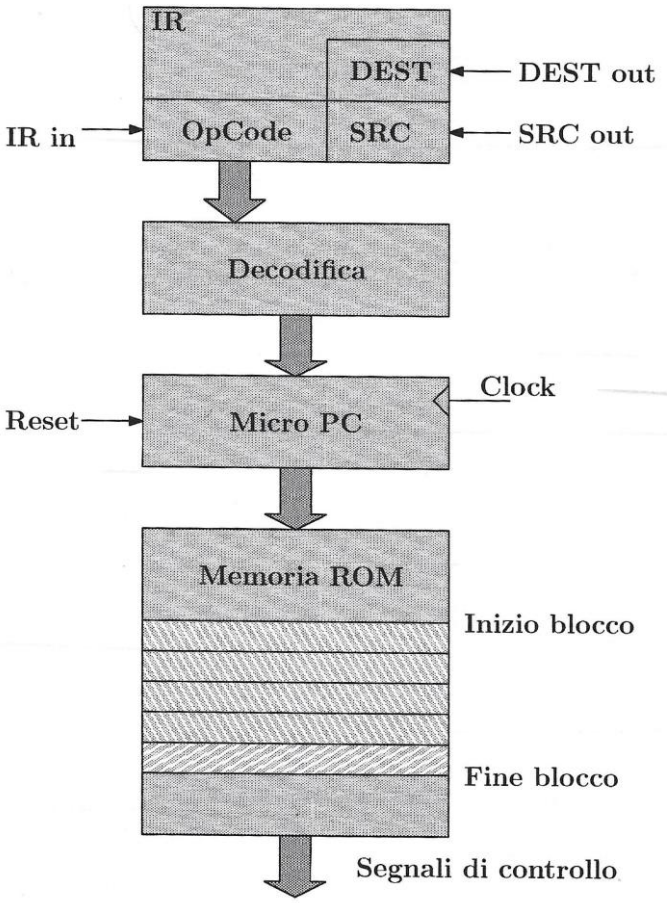

Figura 5: Schema a blocchi di una unità di controllo microprogrammata [22]

L'organizzazione dell'unità di controllo può essere di tipo verticale o orizzontale: la più semplice tra le due è quella orizzontale, anche se è quella che richiede una memoria ROM più estesa.

Nella microprogrammazione orizzontale, l'unità di controllo esegue il microcodice in modo sequenziale a partire da un indirizzo della memoria ROM a cui è associato l'OpCode della macro-istruzione. In questa organizzazione, ogni microcodice inizia prima con la fase di Fetch della macroistruzione successiva e si conclude con l'aggiornamento del "micro Program Counter": da qui deriva la necessità di una memoria ROM più estesa, poiché più microcodici possono iniziare e terminare con microistruzioni uguali.

Nella microprogrammazione verticale il microcodice non è eseguito solo sequenzialmente, ma si possono verificare anche "salti" dell'indirizzo della ROM: questo permette di evitare le ripetizioni di microcodici. Tuttavia, ne consegue che l'unità di controllo aumenta in complessità.

Le unità di controllo basate su **controllo cablato** sostituiscono il “microcode engine” con un circuito logico combinatorio che genera direttamente i segnali di controllo a partire dall’OpCode dell’istruzione corrente.

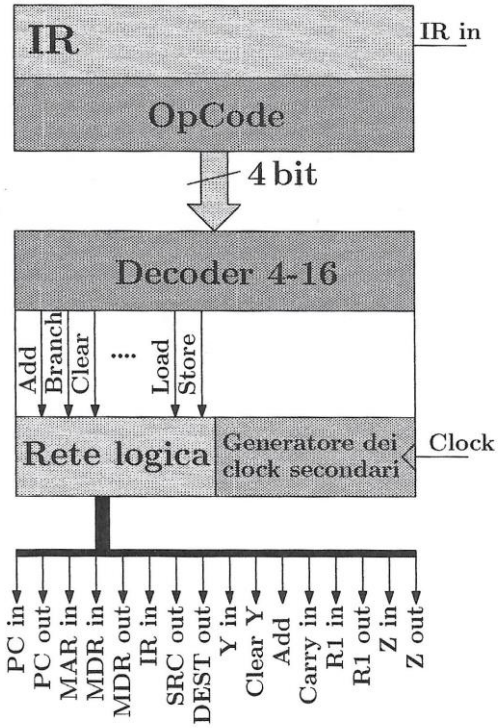

Figura 6: Schema a blocchi di una unità di controllo cablata [22]

Da come si può osservare in Figura 6 , l’OpCode è passato ad un decoder, il quale consiste in un demultiplexer che, successivamente, tramite una rete logica va ad impostare i segnali che vanno a gestire le altre componenti del processore.
Il controllo cablato occupa un’area di silicio nettamente minore di quello micorprogrammato: tuttavia, ha l’inconveniente di essere “rigido”, ovvero, una volta predisposto il circuito, non è possibile modificarlo per aggiungere una nuova istruzione, anche se la loro velocità e compattezza ha portato alla loro adozione in tutti i processori di recente progettazione.

1.3.2.La ALU

L'unità aritmetico logica (ALU) è un insieme di circuiti digitali che permetto a un processore di eseguire alcune operazioni fondamentali di tipo logico e aritmetico sui dati binari contenuti nei registri.
Ci sono delle operazioni sempre disponibili:

- somma e differenza di due registri
- operazioni logiche fondamentali, sia a due o più operandi, come AND e OR, che unarie, come la NOT
- scorrimento e/o rotazione a destra o sinistra di un registro

Alcuni processori hanno ALU in grado di fare operazioni complesse come moltiplicazioni o divisioni.

1.3.3.I registri

I registri sono le unità di memoria di una CPU e contengono gli operandi delle istruzioni del repertorio.
Si può osservare come nell'immagine del processore a 8 bit vi siano rappresentati i registri:

- PC
- MAR
- MDR
- R1
- Z
- Y

Il Program Counter (PC) è il registro che contiene l'indirizzo della prossima istruzione (OpCode) da eseguire nella memoria programma, il suo scopo è tenere conto a che punto dell'esecuzione del codice si è arrivati.

Il Memory Address Resgister (MAR) e il Memory Data Register (MDR) sono registri che gestiscono la memoria.

R1 è il registro di lavoro.

Y e Z sono registri che servono da supporto della ALU dove vengono scritti rispettivamente i dati su cui eseguire le operazioni e il risultato delle operazioni.

CAPITOLO 2

Principali fonti di variazione:

Da anni nella fabbricazione di dispositivi digitali vengono utilizzati i transistori MOSFET. Da quando avevano fatto la loro prima comparsa, Moore aveva ipotizzato una legge che fino ad ora è stata rispettata.

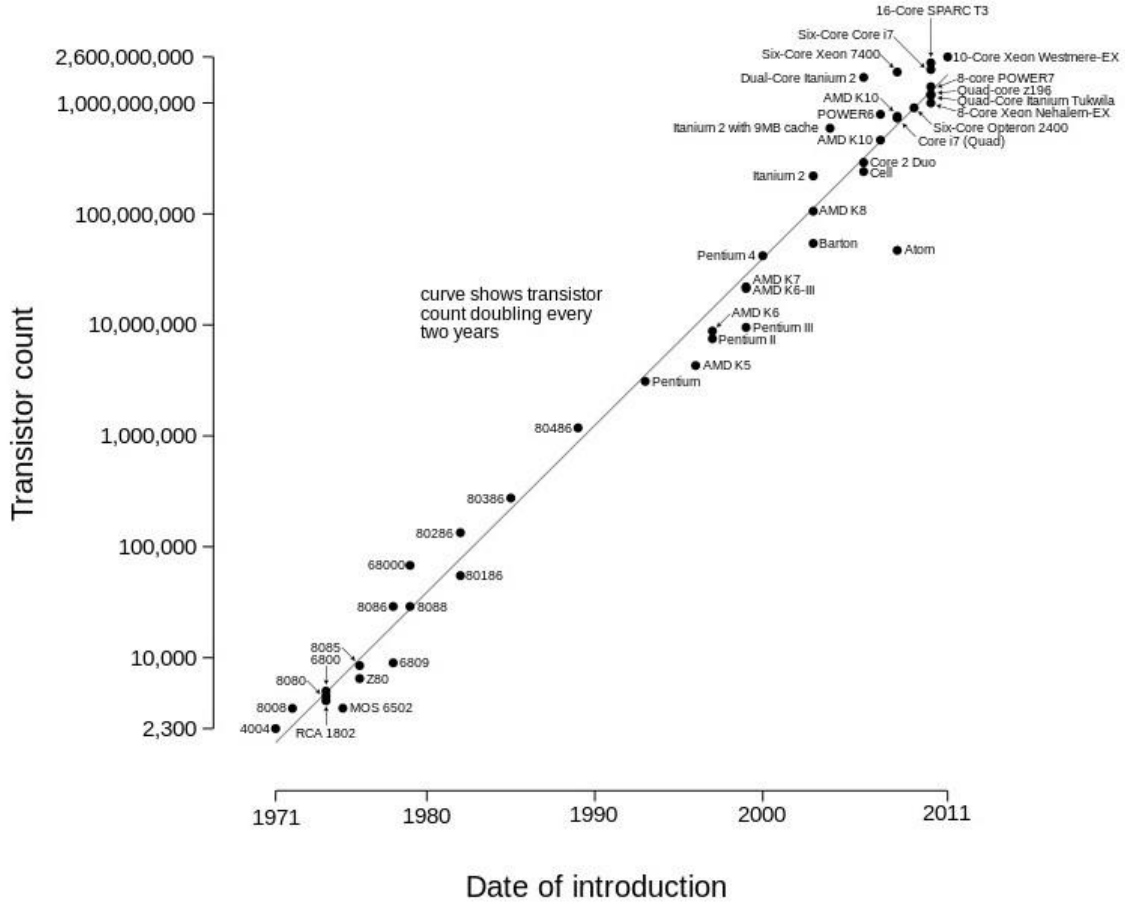

Figura 7: Grafico che mostra l'andamento della densità di transistori nei chip

La legge di Moore prevede che il numero di transistori presenti su un circuito integrato, raddoppia ogni 18 mesi. Questo è possibile grazie alla miniaturizzazione dei transistori che lo compongono. Questo di per sé rappresenta già un vantaggio, in quanto è possibile implementare una analoga architettura su un'area di silicio minore (costi minori), oppure si può pensare di implementare architetture più complesse sulla stessa quantità di area. Inoltre, le continue miniaturizzazioni non portano ad un vantaggio solo in termini di area, ma consentono anche di migliorare le prestazioni dei dispositivi MOSFET: in genere una riduzione delle dimensioni in modo opportuno porta a consumi e tempi di commutazioni minori. Questo avverrebbe se si considerassero i processi di produzione ideali: tuttavia, nel gran numero di processi da eseguire sul Silicio vi sono

sempre delle non idealità che portano ad una variazione delle caratteristiche del transistore rispetto alle specifiche di progetto. Queste differenze dalle specifiche comporta un distaccamento dalle specifiche del processore che è stato progettato. Per raggruppare velocemente tutte le principali cause di variazioni viene utilizzato l'acronimo PVT (talvolta PVTA) che sta per Process Voltage Temperature (PVT Aging), ovvero tutte quelle variazioni che possono essere imputate a variazioni nel Processo produttivo e fluttuazioni della tensione di alimentazione e della temperatura del chip durante il suo funzionamento e invecchiamento.

Nel primo paragrafo viene introdotto il transistore MOSFET e viene spiegato come le sue dimensioni fisiche influiscano direttamente sulle sue caratteristiche; nel secondo paragrafo vengono introdotte le variazioni di Processo che in genere sono ritenute costanti nel tempo; infine nel terzo, quarto e quinto paragrafo vengono discusse, rispettivamente, le variazioni di tensione, temperatura e quelle dovute ai fenomeni di invecchiamento che in genere vengono considerate casuali perché possono variare nel tempo.

2.1.MOSFET

In elettronica si possono individuare due categorie di transistori [17]:

- Transistori ad effetto di campo, a cui appartiene il MOSFET (Metal Oxide Semiconductor Fiel Effect Transistor)
- Transistori ad effetto giunzione, a cui appartiene il BJT (Bipolar Junction Transistor)

Le differenze principali tra i due transistori risiedono nel fatto che i MOSFET hanno consumi minori e un'alta impedenza di ingresso, mentre i BJT hanno alte frequenze di lavoro e sopportano una maggiore corrente. Questo, fa preferire l'utilizzo dei MOSFET nei circuiti digitali.

Nell'ambito dell'elettronica digitale la scelta non è ricaduta sui MOSFET solo per i loro consumi ridotti, ma anche per la possibilità di implementare una tecnologia CMOS. Essa si realizza con un numero uguale di p-MOSFET e n-MOSFET: i primi sono necessari per generare i casi in cui lo stato della rete deve essere a '1', mentre i secondi servono quando lo stato deve essere a '0'.

In Figura 8 è rappresentato un inverter CMOS.

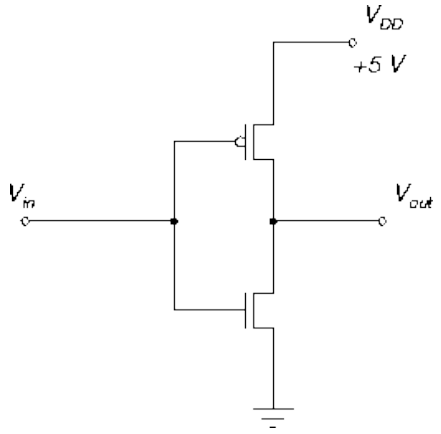

Figura 8: Inverter CMOS

L'inverter CMOS corrisponde all'operatore Booleano NOT, il transistore superiore è un PMOS, quello inferiore di tipo NMOS. Quando V_{in} ha come stato logico '1', l'NMOS diventa un corto-circuito e lo stato logico di V_{out} diventa '0', al contrario V_{in} = '0' fa diventare un corto-circuito il transistore PMOS quindi V_{out} = '1'.

Uno dei processi principali che portano alla creazione del MOSFET è il Drogaggio: vengono iniettati, all'interno di un blocco di silicio (Si) puro, degli atomi detti dopanti. Lo scopo del drogaggio del materiale è creare una sovrabbondanza di cariche in certe zone. Si dice silicio di 'tipo p' quando si ha un eccesso di lacune, silicio di 'tipo n' quando si ha un eccesso di elettroni.

La struttura del MOSFET è quella rappresentata in figura.

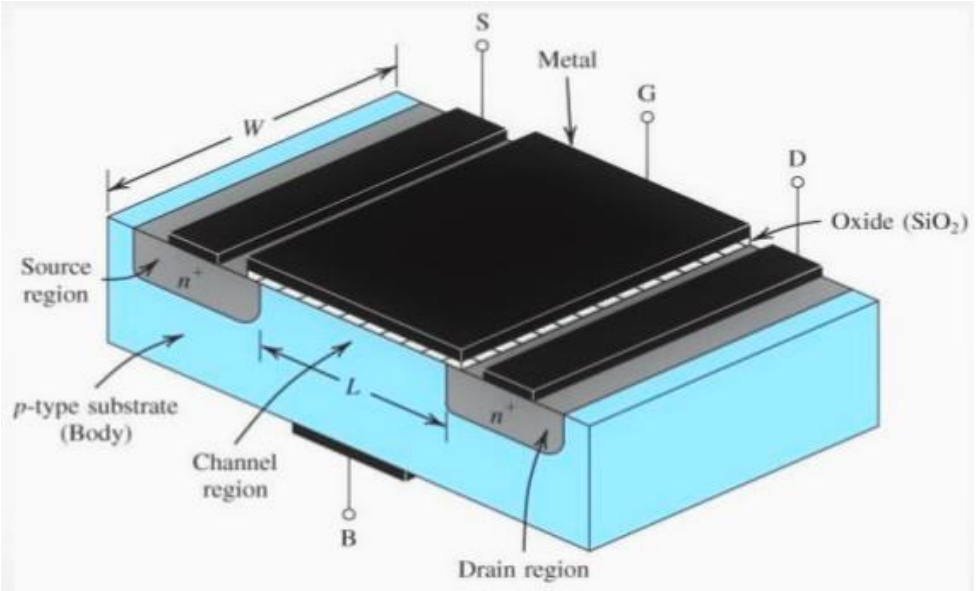

Figura 9: Transistore di tipo n-MOSFET [17]

Si noti che il transistore è composto da quattro terminali:

- Source (S)
- Gate (G)
- Drain (D)
- Body (B)

Al di sotto del terminale di Source e Drain il silicio è stato drogato in modo da ottenere il 'tipo n', mentre in tutto il resto (substrato) è stato drogato in modo da ottenere il 'tipo p'. Questo MOSFET prenderà il nome di p-MOSFET, mentre verranno chiamati n-MOSFET, i transistori che hanno il substrato di 'tipo n'.

Si noti che sotto al terminale di Gate si ha la regione di canale con dimensioni W e L, dove, appunto, L è la lunghezza di canale e W la larghezza di canale. Tra il terminale di Gate e la regione di canale (dello stesso tipo del substrato) vi è un materiale isolante che li separa; per isolare solitamente viene utilizzato l'Ossido di Silicio (SiO_2), poiché facilmente implementabile sul transistore.

Le tensioni e le correnti ai terminali dell'n-MOSFET sono rappresentati nella figura seguente:

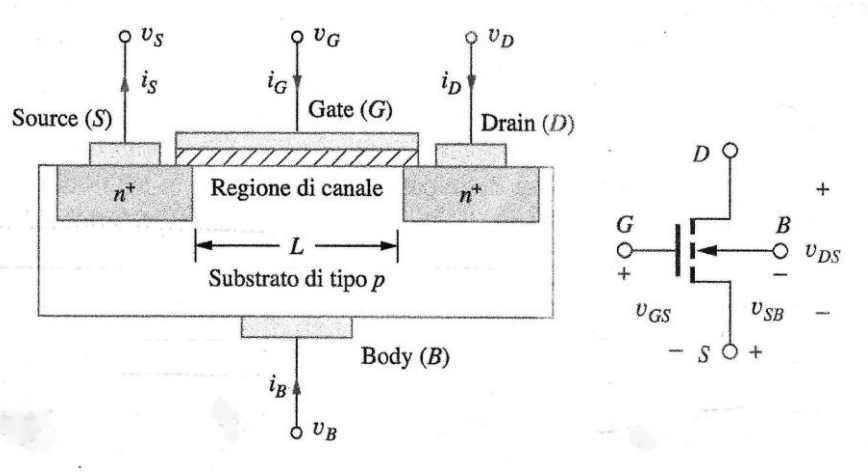

Figura 10: vista in sezione di un NMOS (sinistra), simbolo circuitale NMOS (destra) [17]

Agendo sulle tensioni dei terminali è possibile far lavorare il transistore n-MOSFET in tre maniere differenti, tenendo sempre presente che si considerano i_G e i_B nulle e quindi che per kirckhoff si deve avere i_D=i_S :

- Regione di triodo (o regione lineare): si ottiene ponendo V_{GS} - V_{th} >= V_{DS} >= 0, ne risulta che la corrente di drain dipende linearmente dalla tensione di gate;
- Regione di saturazione: si ottiene quando V_{DS} >= V_{GS} – V_{th} >= 0 si verifica la strozzatura (pinch off) del canale. In questo caso la corrente di drain dipende

quadraticamente dalla tensione di gate, mentre non dipende dalla tensione di drain (se si trascura l'effetto di modulazione della lunghezza di canale). Il MOSFET viene polarizzato in questo modo quando si utilizza come amplificatore;

- Regione di svuotamento: si ottiene per una tensione $V_{GS} < V_{th}$, ne risulta che il transistore può essere considerato come un circuito aperto, anche se scorre una leggera corrente di sottosoglia;

In applicazioni digitali, il MOSFET è utilizzato come un interruttore. Idealmente un interruttore presenta due sole modalità di funzionamento:

- "spento", quando si comporta da circuito aperto, ovvero nel caso in cui la corrente che scorre è nulla.
- "acceso", quando si comporta da corto-circuito, ovvero con caduta di tensione nulla ai suoi capi;

Per approssimare questo tipo di comportamento, è necessario che sia previsto per il MOSFET la seguente modalità di polarizzazione:

- $V_{GS} < V_{th}$ quando si desidera interdire il dispositivo. A meno di piccole correnti di perdita, il comportamento da interruttore spento è ben approssimato;
- $V_{GS} > V_{th}$, quando si desidera far condurre il dispositivo. Generalmente il MOSFET opera in regione triodo, esibendo quindi una resistenza equivalente tra i terminali di drain e source. Minore è questa resistenza, minore sarà la caduta di tensione ai suoi capi, migliore sarà il comportamento da interruttore "acceso"

È da notare l'importanza delle caratteristiche fisiche del transistore: infatti, quando il transistore lavora nella regione di triodo si ha che la corrente che circola in esso è pari a:

$$i_D = \left[C_{ox} W \left(v_{GS} - V_{th} - \frac{v_{DS}}{2}\right)\right]\left(\mu_n \frac{v_{DS}}{L}\right) \qquad (2.1)$$

Inoltre, sempre in regione di triodo, si ha che la resistenza di conduzione del transistore R_{on} è:

$$R_{on} = \frac{T_{ox} L}{\mu_n \varepsilon_{ox} W (V_{GS} - V_{th})} \qquad (2.2)$$

Si noti che fattori principali che vanno ad agire sia sul valore della corrente i_D che sulla resistenza R_{on} sono:

- W (larghezza di canale)
- L (lunghezza di canale)
- V_{th} (tensione di soglia del transistore)
- Mobilità delle cariche (μ_n)
- T_{ox} (spessore del dielettrico o ossido)
- Permettività dell'ossido (ε_{ox})

La scelta di ridurre le dimensioni dei transistori è guidata sia da un fattore di integrazione dei dispositivi all'interno del singolo die, sia da un fattore legato ai consumi di potenza L'utilizzo di dimensioni minori comporta una riduzione delle tensioni e correnti necessarie al loro funzionamento, ma comporta pure una maggiore difficoltà nei processi produttivi e una sensibilità alle variazioni non indifferenti.

Si è notato soprattutto che le variazioni PVT non sono più trascurabili da quando si è arrivati a processi produttivi a 90nm (ricordando che il valore del processo produttivo indica una media della lunghezza di canale L del transistore creati tramite quel processo).

In ciò che segue vengono trattate le fonti di variazioni principali che portano ad un degrado delle prestazioni del transistore e quindi dei chip.

2.2.Processo

In questa sezione ci si focalizza sulle principali variazioni delle caratteristiche del transistore che sono causate dal processo di produzione.
L'insieme di tutti processi che vengono eseguiti per la produzione di un chip portano inevitabilmente ad avere delle deviazioni delle proprietà elettriche dei transistori rispetto ai valori di progetto.

Nell'analizzare le principali fonti di variazione nell'ambito del processo tecnologico è opportuno innanzitutto distinguere tra effetti spaziali e temporali.
La suddivisione spaziale ha lo scopo di evidenziare quali variazioni sono maggiormente correlate con lo spazio. Si dividono in [13]:

- Variazioni globali (inter-die)
- Variazioni locali (intra-die)

Le variazioni globali sono quelle che mostrano una maggior correlazione con lo spazio. In questa categoria ricadono le variazioni L2L (Lot-to-Lot) e W2W (Wafer to Wafer), dove parametri del transistore cambiano in maniera uguale per tutti i transistori appartenenti al Wafer o al Lotto prodotto. Tali variazioni sono perciò legate a problematiche di uniformità del processo tecnologico lungo il wafer.

Le variazioni locali sono quelle che non mostrano correlazione con lo spazio, cambiano da transistore a transistore e in questa categoria vi ricado le variazioni WID (With-In-Die) e D2D (Die-2-Die). Le variazioni intra-die sono quindi legate a fluttuazioni stocastiche, e perciò non prevedibili, delle grandezze fisiche e geometriche del transistore. Per cui, se anche considerassimo due transistori sufficientemente vicini tra di loro, in modo tale da poter considerare il processo tecnologico uniforme, saremmo comunque di fronte a dispositivi con caratteristiche elettriche leggermente diverse.
In qualsiasi tecnologia MOS, la varianza della V_{th} tra transistori adiacenti dipenda dalle dimensioni dei transistori W e L e dalla superficie del gate. La legge di Pelgrom [1] mostra la dipendenza matematica della varianza a tali parametri:

$$\sigma_{VT}{}^2 = \frac{A_{VT}{}^2}{2WL} \qquad (2.3)$$

La suddivisione temporale ha lo scopo di far notare in che modo le variazioni cambiano nel tempo. Esse, si dividono in:

- Statiche
- Dinamiche

Le variazioni statiche rimangono costanti dopo il processo di produzione, oppure cambiano molto lentamente a causa di effetti di invecchiamento.
Le variazioni dinamiche colpiscono il processore durante il suo funzionamento e possono essere lente (se hanno costanti di tempo dell'ordine dei kHz), oppure veloci (se hanno costanti di tempo minori di un ciclo di clock).

Si può inoltre parlare di variazioni bilanciate o non bilanciate. Nel primo caso i transistori NMOS e PMOS hanno variazioni uguali, mentre nel secondo caso i transistori subiscono variazioni differenti.

Quando si hanno chip affetti maggiormente da variazioni bilanciate, questi vengono utilizzati per lo studio dei ritardi. Questi componenti possono essere etichettati con FF, TT o SS che stanno ad indicare la velocità dei transistori (Fast, Typical e Slow, vengono utilizzate 2 lettere per dire se si tratta di NMOS o PMOS) [5].

Quando si hanno chip affetti maggiormente da variazioni non bilanciate, questi vengono utilizzati per studiare i casi limite per la durata degli impulsi o del duty cycle, vengono suddivisi in SF o FS.

Entrando nel dettaglio, le principali fonti di variazione di processo che verranno spiegate nei successivi sotto paragrafi sono le seguenti:

- Variazioni delle dimensioni critiche (CD), Line-Edge Roughness (LER), Line-Width Roughness (LWR)
- Fluttuazione casuale dei dopanti (RDF)
- Spessore dell'ossido (T_{ox}).

2.2.1.Variazioni CD, LER e LWR

Per variazioni CD si intendono le variazioni delle dimensioni critiche, che sono L e W [13]. Si è visto come questi due parametri siano molto rilevanti per le caratteristiche del transistore, in particolare per la corrente e per il valore della resistenza. Inoltre, la lunghezza di canale gioca un ruolo importante sulla V_{th} (effetto di roll-of), che incide sulla velocità di commutazione del transistore.

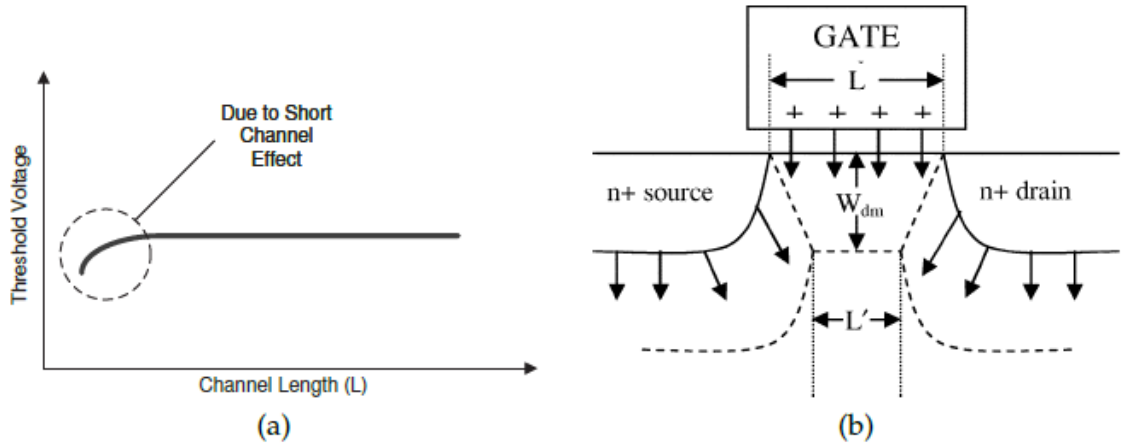

Figura 11: Effetto della lunghezza di canale sulla tensione di soglia [9]

La diminuzione della V_{th} è dovuta alla penetrazione maggiore delle regioni di svuotamento, che si creano all'interfaccia tra source e substrato e tra drain e substrato, sotto il gate (ovvero nel canale) ne risulta che la tensione necessaria per ottenere l'inversione del canale è minore perché la regione di canale è svuotata già in parte [9].

La variazione di L e W è principalmente causata dalla fase litografica nel processo di produzione: la lunghezza d'onda della luce utilizzata, infatti, è maggiore della dimensione dei transistori che devono essere implementati [13].

Per ottenere dimensioni minori si utilizzano delle lenti che riducono il fascio, il che porta alla manifestazione del fenomeno della diffrazione che conduce a “sbavature” che vanno a modificare sia le dimensioni, sia i bordi alle interfacce che corrispondono alle variazioni LER (Line-Edge Roughness) e LWR (Line-Width Roughness).
Questo tipo di variazioni ha acquisito un’importanza sempre più evidente perché, anche se le variazioni sono molto leggere, su dimensioni così ridotte come quelle attuali conducono a deviazioni molto importanti.

2.2.2.RDF

Come spiegato prima durante la fase del processo di produzione vengono iniettati atomi, accettori o donatori, che vanno a drogare il Si in modo da ottenere eccessi di cariche positive o negative. A questa fase è correlata una variazione che prende il nome di Random Dopant Fluctation (RDF) che, tiene conto del fatto che la presenza di atomi droganti nel Silicio segue una distribuzione di tipo Poisson.
La diminuzione delle dimensioni geometriche con l’avanzamento tecnologico ha fatto si che il numero di droganti nel canale sia notevolmente ridotto, tanto che, quando il processo produttivo era a 1 μm, nel canale vi erano migliaia di atomi dopanti mentre, con un processo produttivo a 32 nm, mediamente si hanno meno di 100 atomi dopanti nella regione di canale. È evidente che una variazione anche di poche unità nel numero di dopanti porta a variazioni significative nella concentrazione effettiva.

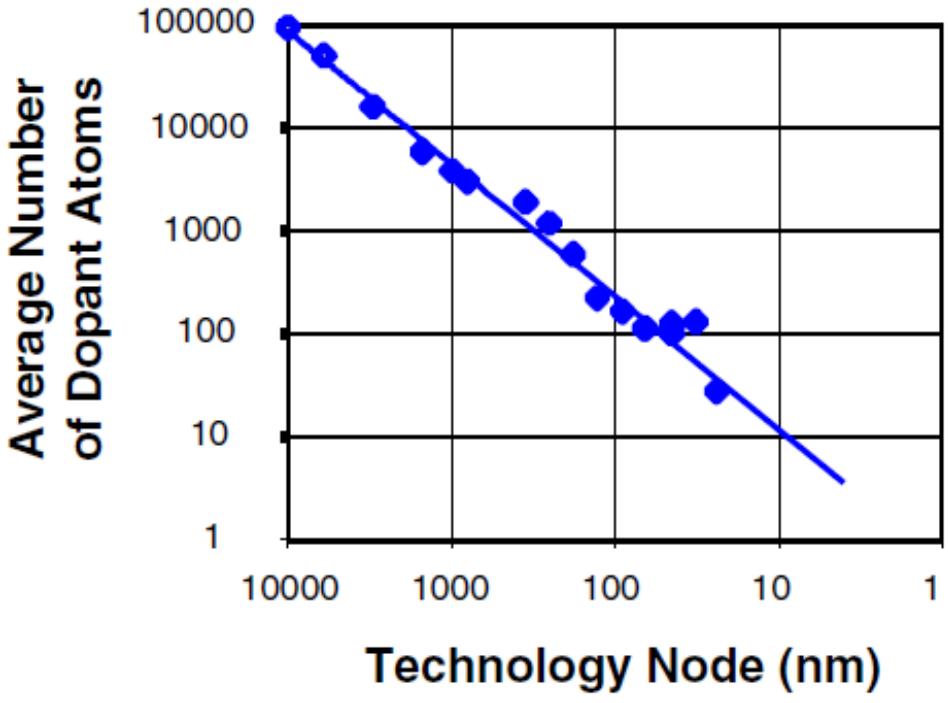

Figura 12: Grafico che mette in relazione il processo tecnologico con il numero di atomi dopanti contenuti nel transitore [12]

La variazione del numero di atomi dopanti attualmente è un problema critico, agisce sia sul valore della tensione di soglia del transistore, sia sulla corrente che vi può circolare, poiché va a modificare la mobilità delle cariche maggioritarie.

2.2.3.Spessore dell'ossido

Lo spessore dell'ossido di gate è un parametro fondamentale che influenza direttamente la capacità di gate.

$$C_{ox} = \frac{\varepsilon_{ox}}{T_{ox}} \qquad (2.4)$$

Esso agisce sul valore della tensione di soglia, ma soprattutto sulla corrente che può circolare nel transistore. In generale una diminuzione di questo spessore, porta ad un aumento della corrente di drain. Tuttavia, in termini di affidabilità, ossidi più sottili portano a maggiori correnti di perdita attraverso il gate e ad una maggiore probabilità di breakdown (per via dell'elevato campo elettrico).

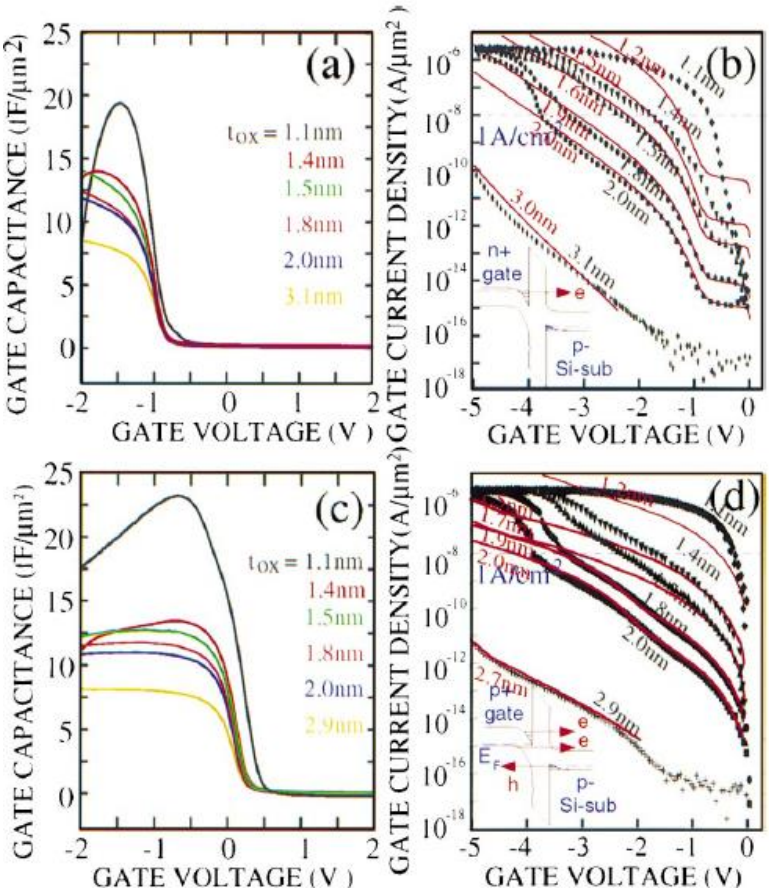

Figura 13: (a) grafico che mette in relazione la tensione di gate con la capacità di gate per un NMOS con terminale di gate policristallino n+ in funzione dello spessore dell'ossido. (b) grafico che mette in relazione la tensione di gate con la corrente dispersa per un NMOS con terminale di gate policristallino n+ in funzione dello spessore dell'ossido. (c) grafico che mostra la relazione tra tensione di gate e la capacità di gate per un NMOS con terminale di gate policristallino p+ in funzione dello spessore dell'ossido. (d) grafico che mette in relazione la tensione di gate con la corrente dispersa per un NMOS con terminale di gate policristallino p+ in funzione dello spessore dell'ossido. [8]

Nell'immagine viene mostrato chiaramente come mai la diminuzione dello spessore dell'ossido sia così ricercata. Esso va ad agire sul valore della capacità di gate: ovvero, incrementando il suo valore, in parallelo aumenta il valore della corrente che può circolare nel transistore (perché $C_{ox} = \varepsilon_{ox} / T_{ox}$). Tuttavia, essendo la potenza dinamica dissipata, nella tecnologia CMOS complementare, pari a:

$$P = \frac{1}{2} C V^2 f \qquad (2.5)$$

Ne risulta che un aumento della capacità di gate, tramite la riduzione dello spessore del dielettrico, porta ad un consumo di energia maggiore.

2.3.Tensione

Idealmente, si presume che la tensione di alimentazione sia uguale e costante su tutti i componenti del die. Tuttavia vi sono delle variazioni e le principali cause sono:

- cadute di tensione resistive
- cadute di tensione a causa di correnti indotte => V=L (di/dt);

Questi sono effetti vengono chiamati "power noise" e tipicamente hanno costanti di tempo comprese tra i ns e i µs; essi non causano solamente cadute di tensione, ma possono essere l'origine anche di overshoot della stessa.

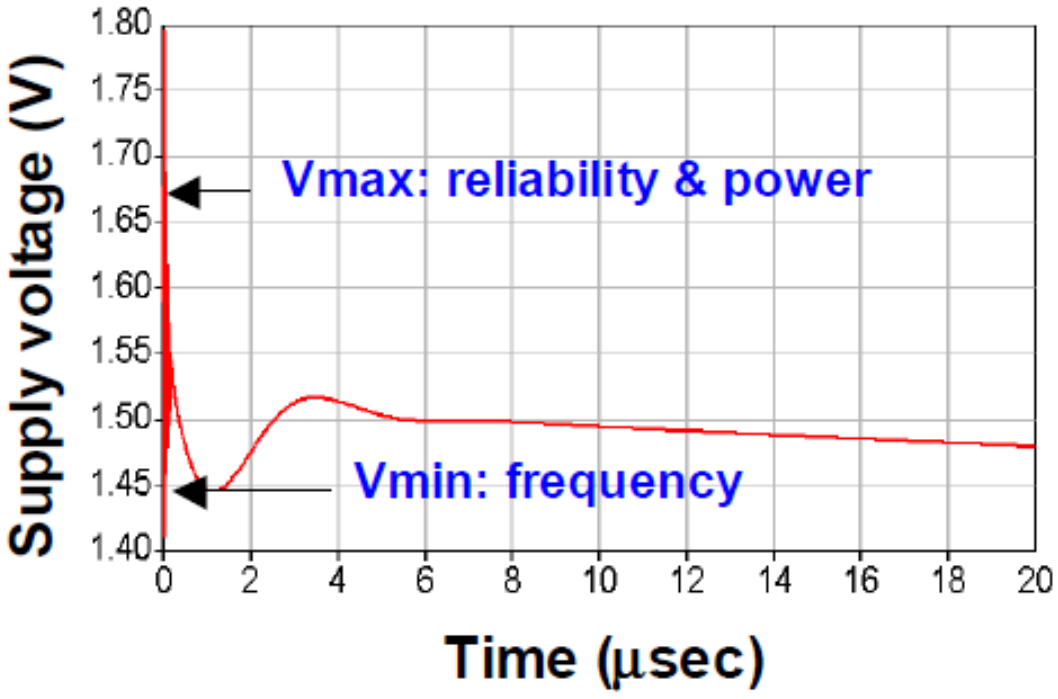

Figura 14: Esempio di variazione di tensione [18]

2.4.Temperatura

Le variazioni di temperatura sono globali se causate dalla variazione della temperatura ambientale, mentre vengono dette locali (hot spot) se causate da carichi di lavoro eccessivi in una certa zona. In quest'ultimo caso si parla di autoriscaldamento del dispositivo.

Entrambe portano ad un rallentamento del circuito a causa dell'aumento della resistenza delle interconnessioni e la riduzione della mobilità delle cariche. Normalmente queste variazioni hanno una costante di tempo che si trova tra i millisecondi e i secondi. In Figura 15 viene mostrato come può variare la temperatura all'interno di un processore.

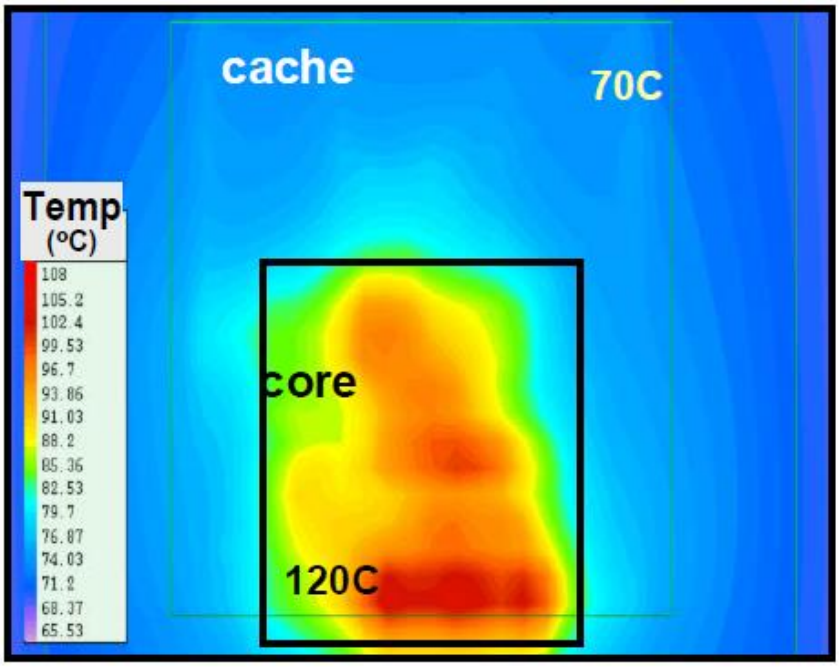

Figura 15: Variazioni di temperatura all'interno di un die [18]

2.5.Invecchiamento

Ultime variazioni e non meno importanti delle altre sono le variazioni dovuti a fenomeni di invecchiamento. Si manifestano durante la vita del componente quindi sono fortemente dipendenti dal loro tempo di utilizzo. Le principali fonti che portano ad una degradazione delle caratteristiche elettriche dei dispositivi sono:

- NBTI e PBTI;
- HCI;
- TDDB;

2.5.1.NBTI e PBTI

Con NBTI e PBTI si fa riferimento al fenomeno di Bias Temperature Instability che, può essere Positive nel caso di n-MOSFET e Negative nel caso di p-MOSFET. Esso porta ad un incremento della tensione di soglia (V_{th}) e una riduzione della velocità di commutazione [13].
Tra i due fenomeni è sempre stato considerato di rilevanza maggiore l'NBTI anche se, con l'introduzione di dielettrici ad elevate costanti dielettriche nelle tecnologie sotto i 45nm, PBTI è diventato un problema di uguale rilevanza [19].

In particolare l'NBTI è comunemente attribuito [13] alla creazione di "trappole" nell'interfaccia tra il dielettrico e il canale del transistore che, catturano le cariche circolanti nel canale del transistore durante il suo funzionamento e vengono successivamente rilasciate solo quando il transistore è a riposo.
Questo non sarebbe un problema se il tempo di cattura e di emissione fosse uguale, tuttavia la differenza tra loro è molta e questo fa in modo che una piccola variazione della V_{th} può essere osservata già dopo pochi microsecondi di utilizzo anche se un aumento considerevole della V_{th} si ha solo dopo giorni, settimane o addirittura anni.

Occorre precisare infine che [6] l'NBTI dipende dalla temperatura, dal campo elettrico che si ha sull'ossido e in particolare dal tempo e si verifica quando il transistore è adoperato nella regione di triodo. Inoltre la variazione della tensione di soglia può essere parzialmente recuperata quando il dispositivo è spento.

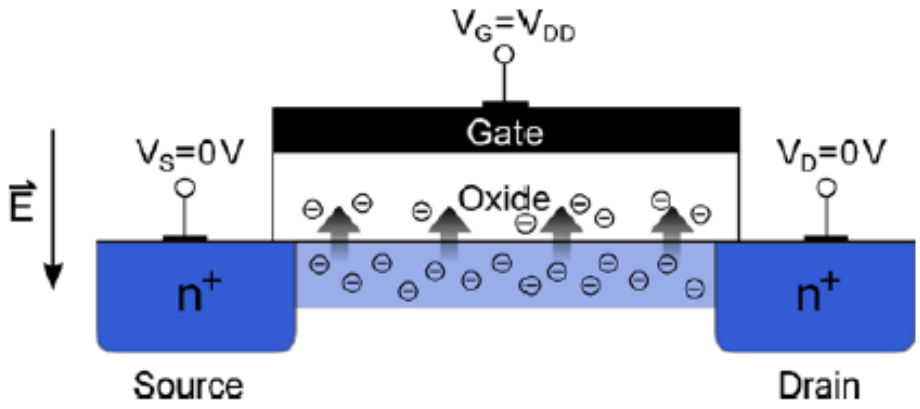

Figura 16:fenomeno di Bias Temperature Instability per un n-MOSFET (NBTI) [13]

2.5.2.HCI

Per HCI si intende Hot Carrier Injection, è un fenomeno che si manifesta maggiormente quando si applicano elevati campi elettrici longitudinali.
Il fenomeno comporta un aumento della tensione di soglia del transistore e una riduzione della corrente che vi può scorrere all'interno.
Storicamente, HCI è stato più significativo negli nMOSFET perché gli elettroni hanno maggiore mobilità delle lacune, e quindi, possono guadagnare una maggiore energia dal campo elettrico che si ha nel canale [19]. A seguito di un urto, l'energia cinetica può essere ceduta ad una coppia elettrone-lacuna. Se l'elettrone così generato ha una componente di velocità trasversale sufficiente, sarà iniettato attraverso l'ossido di Gate.

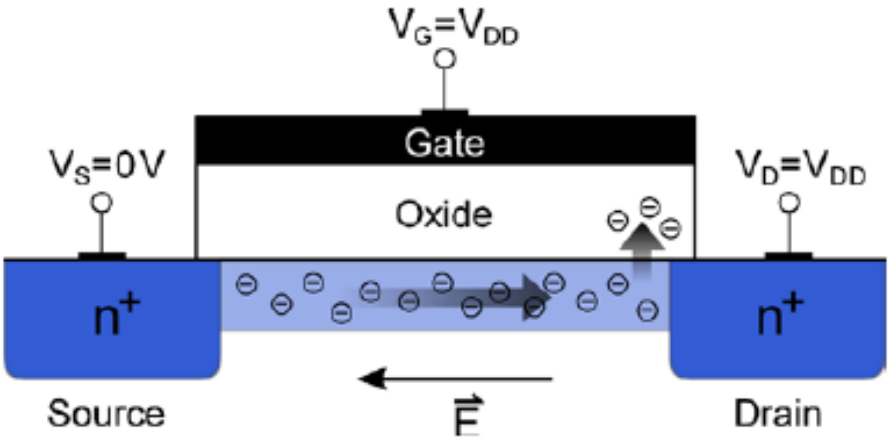

Figura 17: Fenomeno dell' Hot Carrier Injection per un n-MOSFET (HCI) [13]

2.5.3.TDDB

TDDB è l'acronimo di Time Dependent Dielectric Breakdown ed è altresì conosciuto come oxide breakdown. Quando un campo elettrico sufficientemente alto è applicato sul dielettrico del transistore, il continuo degrado del materiale produce dei percorsi conduttivi, i quali possono creare un cortocircuito tra anodo e catodo.

CAPITOLO 3

Soluzioni per la tolleranza delle variazioni

Nel capitolo precedente sono state evidenziate le principali variazioni che degradano le prestazioni dei processori, così da non rispettarne le specifiche. In numerosi articoli viene ribadito come oramai una importante sfida in questo settore sia rappresentata dal trovare delle soluzioni che permettano la riduzione o la compensazione di queste variazioni.
Con la riduzione del processo tecnologico da 350nm a 90nm, la resa dei chip si è ridotta da quasi il 90% a meno del 50% e, con 45nm, uno studio ha riportato che la resa è attorno al 30% [23].
Dallo studio di Michael Liu et al. (2004, p.2) risulta che, ad una tecnologia di 90nm, le perdite di potenza possono arrivare fino al 42% di tutta la potenza fornita.

I principali effetti delle variazioni sono divisibili in due categorie [19]:

- Impatto delle variazioni di processo
- Impatto delle degradazioni temporali

Come già detto le **variazioni di processo** vanno a modificare i principali parametri che caratterizzano i transistori prodotti (L, W, Tox, ecc.) e si manifestano: incrementando i ritardi e la loro propagazione, variando la velocità tra due chip differenti, diminuendo la resa nei progetti con pipeline (poiché in una pipeline sincrona la velocità è limitata dalla fase di pipeline più lenta), incrementando le perdite di potenza, aumentando la temperatura.

Le **variazioni temporali** sono quelle di tensione, temperatura e dovute ai fenomeni di invecchiamento. Un aumento di temperatura riduce complessivamente la velocità mentre; le fluttuazioni di tensione modificano i ritardi e rovinano la robustezza (un aumento di tensione aumenta la velocità, una diminuzione di tensione la diminuisce); infine i fenomeni di invecchiamento affliggono la velocità del circuito durante il suo utilizzo.

Vi sono due diverse categorie principali di tecniche utilizzabili nel progetto dei processori che permettano di tollerare queste variazioni:

- Tecniche a livello circuitale
- Tecniche a livello di architettura

Verranno ora spiegate prima le tecniche a livello circuitale e successivamente quelle a livello di architettura.

3.1.Tecniche a livello circuitale

Le tecniche a livello circuitale sono le seguenti:

- Progetto conservativo
- Progetto statistico
- CRISTA
- Tecniche run-time

3.1.1.Progetto conservativo

Il **progetto conservativo** è basato sul considerare dei margini associati alla tecnologia considerata: le variazioni di processo, le fluttuazioni di temperatura e tensione e il degrado temporale. Una possibile soluzione è dimensionare il circuito per raggiungere la frequenza scelta, dopo aver considerato tutti i margini o innalzando la tensione di alimentazione.

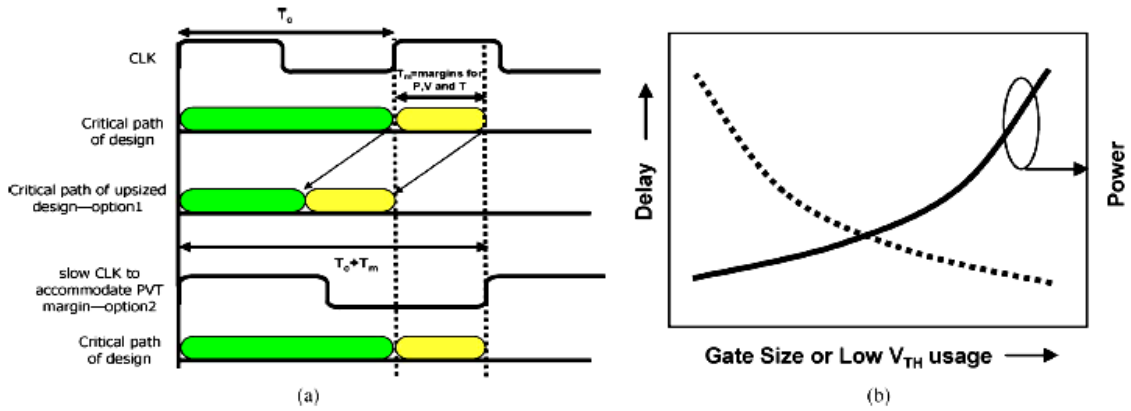

Figura 18:(a) due possibili opzioni per la tolleranza delle variazioni, ridimensionamento del progetto o aumento del periodo di clock. (b) grafico che mostra il compromesso che si ha adottando il progetto conservativo. [19]

In Figura 18(b) viene mostrato come l'approccio conservativo giochi a discapito del consumo di potenza e dell'area. Infatti aumentando le dimensioni e la tensione di alimentazione, al fine di aumentare la velocità delle porte logiche, accresce il consumo di potenza. Tale soluzione consente però di lavorare a frequenze di clock elevate.
D'altro canto, Figura18(a), un'altra possibile soluzione consiste nel ridurre la frequenza per adattare i ritardi del progetto originale.
Il progetto conservativo però non è più accettabile poiché sia la frequenza che il consumo di potenza sono diventati dei fattori predominanti nella progettazione dei processori.

3.1.2.Progetto statistico

Il **progetto statistico** può avvenire in tre modi diversi:

- Dimensionamento logico: prevede la scelta delle dimensioni da attuare nel progetto tramite strumenti statistici che permettano di valutare i ritardi in modo da minimizzare le dimensioni di progetto. Un esempio è lo studio condotto da Seung Hoon Choi et al. (2004, pp.455,456) dove vengono valutati i ritardi dovuti alle variazioni e successivamente vengono calcolate le dimensioni ottimali impostando una certa resa da raggiungere. Di seguito è rappresentato l'algoritmo utilizzato.

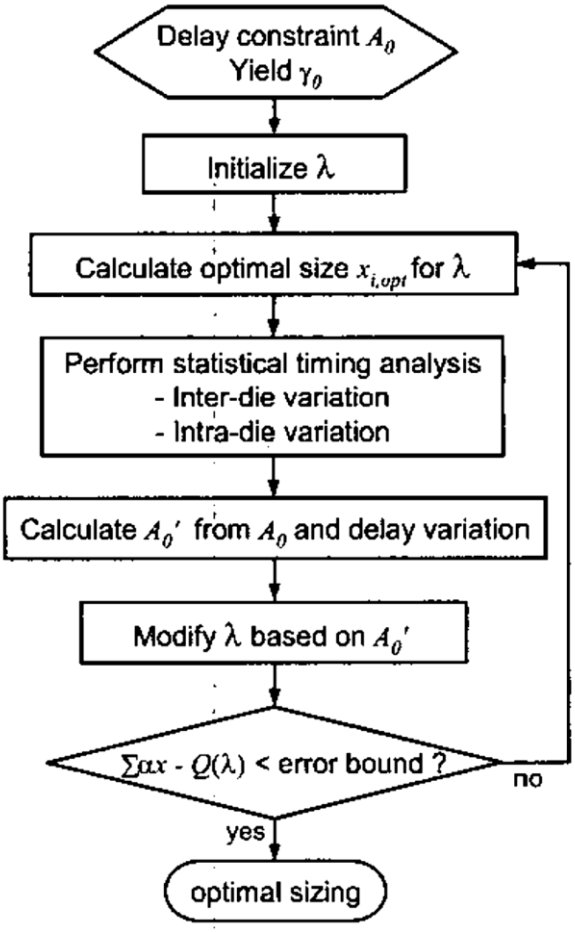

Figura 19: Diagramma di flusso dell'algoritmo per la scelta ottimale delle dimensioni del progetto. [21]

Occorre precisare che A_0 e γ_0 sono rispettivamente il tempo necessario per l'ottenimento dell'output nel "worst case" e la resa che si vuole ottenere. Con questo metodo viene mostrato che l'area risparmiata può essere al massimo del 19% confrontato con il progetto worst case.

- Dimensionamento a V_{th} doppia: la tensione di soglia del transistore (V_{th}) è un altro parametro importante che può essere modificato per ottenere un giusto compromesso tra perdite di potenza e velocità. Nell'articolo Michael Liu et al. (2004, p.2) vengono stimati i ritardi e le perdite di Potenza con lo scopo di trovare due tensione di soglia per il circuito in modo da minimizzare la potenza dissipata, imponendo un limite sul tempo, nei punti in cui la sua densità di probabilità è alta. È opportuno sottolineare la presenza di transistori con due tensioni di soglia (possono averne anche multiple) che sono high-V_{th} e low-V_{th} e servono per poter gestire meglio consumi e velocità dei transistori: una tensione di soglia elevata consente di ridurre i consumi, mentre una più bassa porta a velocità maggiori (a discapito dei consumi).
- Sbilanciamento pipeline: Nella progettazione dei processori risulta importante scegliere il numero di stadi della pipeline tenendo conto che: ridurre la profondità della pipeline (cioè ridurre il numero di stadi) porta a variazioni di tipo WID (With-In Die) quindi, aumentarne il numero permette di raggiungere una resa più alta innalzando però i consumi di potenza e l'area occupata.
  Sbilanciare la pipeline consiste nel rallentare certi stadi rimpicciolendoli e velocizzarne altri ingrandendoli per raggiungere un'ottimizzazione.
  Ad esempio, in un progetto con pipeline a 3 stadi la resa della pipeline (che chiameremo y_p) è pari alla moltiplicazione di tutte le rese di ogni singolo stadio della pipeline, ovvero:

  $$y_p = (y_0 y_1 y_2)$$

  Dove y_0, y_1 e y_2 sono le rese singole di ogni singolo stadio della pipeline.
  Nel caso della pipeline bilanciata si ha $y = y_0 = y_1 = y_2$, in quella sbilanciata si ha che $y_0 \neq y_1 \neq y_2$. Si può ottenere allora un miglioramento, sbilanciando la pipeline, quando la resa della pipeline bilanciata è minore di quella sbilanciata, in formule:

  $$(y_0 y_1 y_2) > (y)^3$$

3.1.3.CRISTA

CRISTA [20] è un nuovo modello per la progettazione di circuiti a basso consumo tolleranti alle variazioni. Il suo nome sta per Critical path ISolation for Timing Adaptiveness e permette uno scaling di tensione aggressivo e un aumento della robustezza del circuito. Il suo principio di funzionamento si divide in tre fasi:

- Isolare e predire la serie di percorsi che possono diventare critici sotto le variazioni di processo
- Assicurare che vengano attivati raramente
- Evitare possibili fallimenti, a causa di ritardi nei percorsi critici, passando dinamicamente a operazione a due cicli (assumendo che esse siano a ciclo singolo) quando sono attivi quei percorsi

Queste fasi permettono al circuito di poter lavorare a tensioni di alimentazioni minori, perdendo in velocità a causa delle operazioni a due cicli.

In Figura 20(a) è possibile notare come la seconda istruzione necessiti di un ciclo di clock aggiuntivo per il suo completamento: modificando il ciclo di clock si permette lo svolgimento delle istruzioni a una tensione minore.
In Figura 20(b) si ha uno schema rappresentativo di un'applicazione di CRISTA per la gestione dinamica della temperatura in un processore con pipeline a 5 stadi, in particolare CRISTA permette di raffreddare lo stadio che normalmente diventa più caldo, ovvero quello dedicato alla fase EX. I percorsi critici vengono predetti per mezzo della decodifica degli input attraverso dei predecodificatori: quando gli input che vanno ad EX corrispondono a quelli che passano per i percorsi critici, i predecodificatori li individuano portando "freeze" allo stato logico 0, bloccando quindi il segnale di clock per eseguire l'istruzione in tempi giusti.

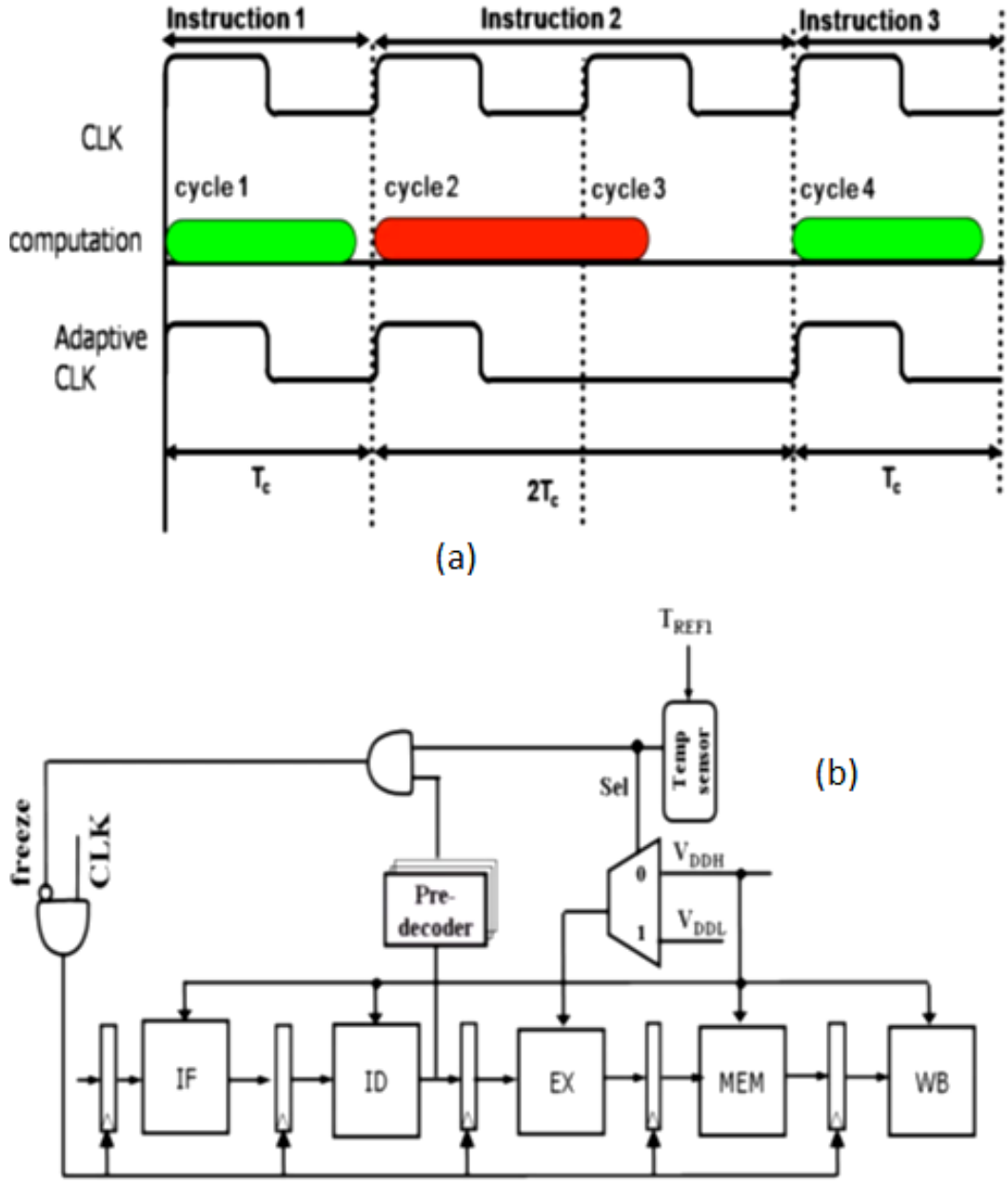

Figura 20: (a) diagramma temporale di CRISTA che mostra il comportamento del ciclo di clock con operazioni che richiedono più di un ciclo di clock. (b) CRISTA a livello microarchitetturale per una gestione adattiva della temperatura. [19]

A temperature nominali i predecodificatori sono disattivati mentre ad elevate temperature viene abbassata la tensione di alimentazione e attivati i predecodificatori in modo da assicurare il raffreddamento della zona e garantendo l'esecuzione di tutte le istruzioni tramite le operazioni a ciclo doppio. Nel caso in cui la temperatura continui ad aumentare allora viene eseguita la fase di "throttling": cioè per evitare che venga danneggiato fisicamente viene non solo abbassata la tensione di lavoro ma pure la frequenza tramite il DVFS (Dynamic Voltage Freqeuncy Scaling).

3.1.4.Tecniche run-time

Le **tecniche run-time** sono utilizzate per migliorare alcuni aspetti del processore durante la sua attività, gli esempi principali sono:

- Adaptive Body Bias
- Adaptive Voltage Scaling
- Dynamic Frequency Scaling
- Sensor-based design

L'Adaptive Body Bias (ABB) è una tecnica di controllo che prende i vantaggi di altre due tecniche di controllo che sono la Forward Body Bias (FBB) e la Reverse Body Bias (RBB). È stato ormai ribadito come abbassare la tensione di soglia V_{th} permetta di migliorare le prestazioni dei componenti ma porti ad un aumento della corrente di perdita. La tensione di soglia V_{th} è una funzione del potenziale che vi è tra i terminali Body e Source del transistore ($V_{th}=f(V_{BS})$).
La V_{th} può essere modulata dinamicamente, a discapito di una potenza dispersa maggiore, per ottenere prestazioni più alte, in componenti che non rispettano le specifiche, fornendo una tensione al Body per rendere V_{BS} diversa da 0.
La formula che lega la tensione di soglia alla tensione tra Body e Source è la seguente:

$$V_{th} = V_{th0} + \gamma\left(\sqrt{v_{SB} + 2\Phi_F} - \sqrt{2\Phi_F}\right) \qquad (3.1)$$

Dove V_{th0} = valore della tensione di soglia per v_{BS} = 0, γ = parametro relativo all'effetto body e $2\varphi_F$ = parametro relativo al potenziale superficiale.

Di seguito viene riportato un grafico, Figura 21, che mostra l'andamento della tensione di soglia per un transistore NMOS

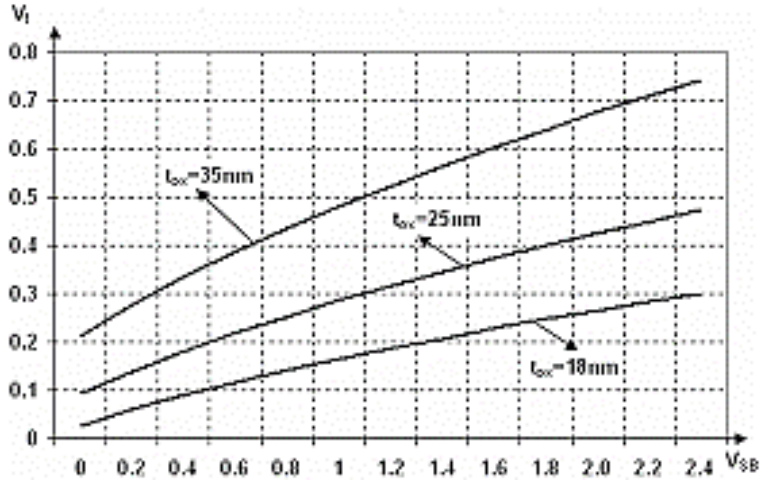

Figura 21: Grafico che mostra un esempio di variazione della tensione di soglia in funzione della tensione tra Source e Body e per diversi valori dello spessore dell'ossido di gate.

Nella Figura 22 viene invece mostrato un grafico che lega la percentuale di guadagno in frequenza che si ottiene con l'FBB.

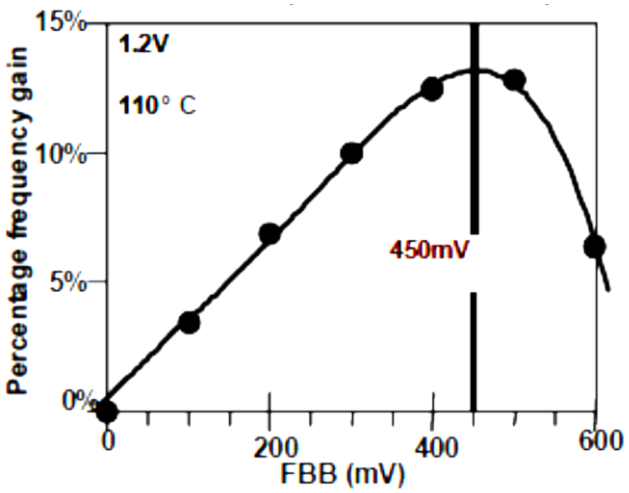

Figura 22: Variazione della percentuale di guadagno in frequenza applicando il Forward Body Biasing (FBB) [18]

Si noti come a temperature elevate (110° C) per tecnologie sotto i 90nm il valore ottimale di tensione sia di 450mV con un guadagno superiore al 10%.
Occorre specificare che la FBB è una tecnica che si utilizza solo sui transistori PMOS (infatti sui transistori NMOS si ottiene una V_{th} maggiore).
Dall'altra parte come già accennato vi è la tecnica di Reverse Body Bias che prevede l'applicazione di una tensione inversa su V_{BS} in modo da innalzare la V_{th}, ottenendo una corrente di dispersione minore e quindi una riduzione delle perdite di potenza.
L'Adaptive Body Bias è quindi una tecnica di controllo che utilizza sia l'RBB che l'FBB per aumentare la resa dei die prodotti, infatti, a causa delle variazioni di processo vi sono die che non raggiungono la massima frequenza e altri che la superano ma hanno troppe perdite. In Figura 23 è possibile osservare il vantaggio che porta l'ABB. La "campana" blu rappresenta la distribuzione dei die prodotti attorno all'obbiettivo di frequenza fissato.

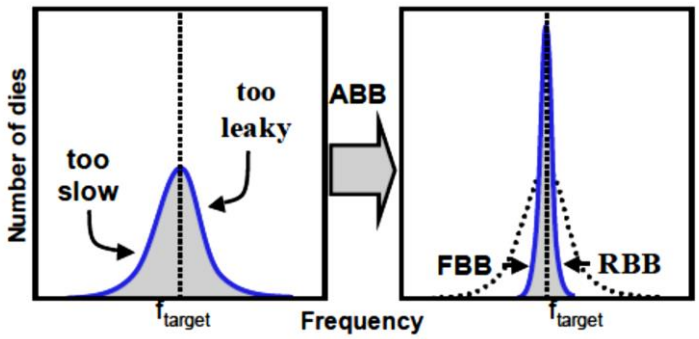

Figura 23: Vantaggio che si ottiene con l'applicazione della Adaptive Body Bias (ABB) [18]

L’Adaptive Voltage Scaling [7] è una tecnica di controllo che prevede di modificare la tensione di alimentazione dei processori: aumentandola nel caso di processori con V_{th} elevata, quindi processori lenti, e diminuendola nel caso di processori con V_{th} bassa, quindi più veloci ma più energivori.
Il vantaggio di questa tecnica di controllo si ha nella resa, come mostrato nella figura sottostante, Figura 24.

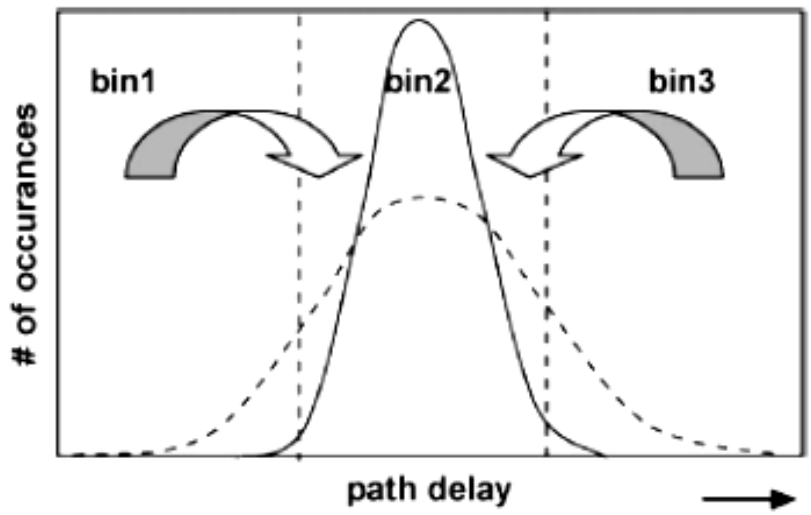

Figura 24: Risultato che si ottiene applicando l'Adaptive Voltage Scaling (AVS) [19]

Si noti che nell’ascissa del grafico è stato messo il path delay quindi I processori che stanno in bin3 sono quelli più lenti.

La sensor-based design è una tecnica di controllo che prevede di inserire nel progetto delle CPU dei sensori che possano stimare le variazioni in modo da poter applicare l’azione corretta per evitare dei possibili fallimenti nel processore. Un esempio di progetto basato su questa tecnica è quello fatto da K. Kang et al. (2007, p.934) dove viene detto come le tecniche come l’ABB e il DVS abbiano dei limiti:

- I circuiti che utilizzano i sensori aumentano l’area e la potenza dissipata
- Possono soffrire delle variazioni di temperatura sul chip soprattutto durante operazioni “pesanti”
- Non sono adatti alla a degradazioni temporali quali i fenomeni di aging

Considerando questo, propongono una tecnica che utilizza un circuito PLL (Phase Locked Loop, un circuito normalmente utilizzato nei processori per generare il clock) come sensore perché è in grado di catturare ogni variazione e generare un ciclo di clock stabile, al suo interno è strutturato in questo modo:

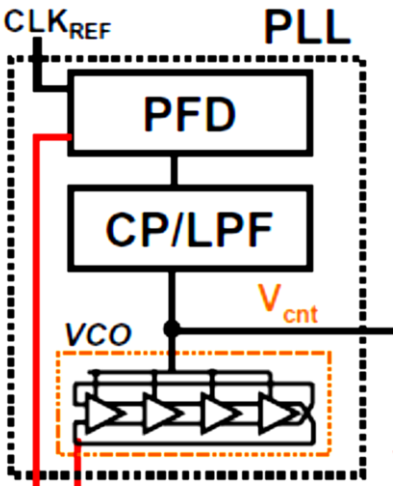

Figura 25: Schema a blocchi di un circuito Phase Locked Loop (PLL) [11]

Il PLL è composto da: un primo blocco che è il PFD (Phase Frequency Detector) che restituisce un riferimento di tensione in base alla fase del segnale in ingresso, un secondo blocco composto da un Low Pass Filter; infine un terzo blocco, il VCO (Voltage Controlled Oscillator) che restituisce un clock in base alla tensione di ingresso.
Nel testo viene utilizzata la tensione generata a monte del VCO per attuare l'ABB in maniera ottimale e viene mostrato come applicandolo ad un RCA (Ripple Carry Adder) si ottenga come l'area e la potenza dissipata possano essere ridotte rispettivamente del 42% e del 43% mantenendo una resa del 99%.

3.2.Tecniche a livello di architettura

Le tecniche a livello di architettura, a differenza delle precedenti, vanno ad inserirsi all'interno dei vari stadi della pipeline.
Esempi di questi sono:

- RAZOR
- ReVIVaL
- Trifecta

3.2.1.RAZOR

RAZOR [5] è una tecnica ibrida, ideata da ARM, per la rilevazione e correzione dinamica degli errori di temporizzazione, risulta molto importante poiché permette di ottenere un sistema robusto e l'eliminazione dei margini. Nell'articolo di riferimento gli autori hanno applicato questo circuito ad un lotto di 87 processori ARM a 32 bit con pipeline bilanciata, fabbricati con un processo produttivo a 65nm, e una frequenza di funzionamento da progetto di 1GHz ma di 724MHz risultante dall'analisi del caso peggiore.

Come detto poco sopra RAZOR è in grado di rilevare gli errori, questo è possibile tramite un circuito che è stato chiamato "Transition-Detector", in italiano Rilevatore di Transizione, che sarà abbreviato con TD. Il circuito è osservabile nell'immagine sotto, Figura 26.

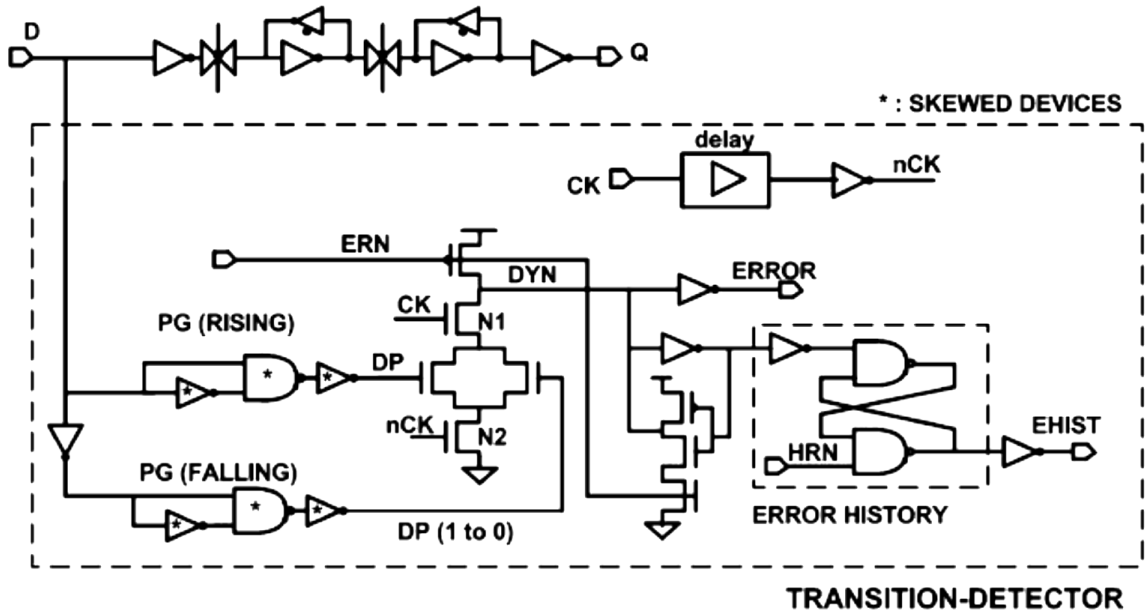

Figura 26:Circuito schematico del Transition Detector (TD) [5]

Il TD in questo caso è applicato a un Flip-Flop di tipo Master-Slave, attivo sul fronte di salita, dove per semplicità vengono rappresentati solo l'ingresso D e l'uscita Q.

Si può notare che all'interno del TD, in alto a destra, vi è un circuito di ritardo in serie ad una porta NOT che genera un segnale di clock ritardato e negato (in seguito verrà spiegata la sua funzione).

Il segnale D del flip-flop viene inviato a due generatori di impulsi (Pulse Generator, PG). La loro funzione è generare un impulso sia per ogni fronte di salita e discesa di D, rispettivamente da PG (RISING) e PG (FALLING). In parole semplici permettono di capire quando si verifica un cambiamento dell'ingresso nel FFMS (Flip-Flop Master Slave).

I Pulse Generator sfruttano il ritardo introdotto dalla porta NOT, inserita in uno dei due ingressi, in modo tale da ottenere per un piccolo istante, se D passa, ad esempio, dallo stato logico ‘0’ allo stato logico ‘1’, che gli ingressi della porta NAND siano entrambi a ‘1’. Si ottiene che DP diventa ‘1’ per pochi istanti di tempo quando si ha una variazione dello stato logico di D.

L’uscita DP viene di fatto posta in NAND con i segnali CLK e nCLK. Ciò significa che l’uscita DYN sarà posta a zero, quando tutti e tre i segnali (DP, CLK e nCLK) sono alti. Da un punto di vista concettuale, l’aver posto in AND (CLK e nCLK) significa abilitare il percorso nel circuito per un breve periodo a valle del fronte di salita. Se anche il segnale DP è alto, questo equivale a dire che il fronte (di salita o di discesa) del segnale D è vicino al fronte di salita del clock. In questo caso è necessario la generazione di un errore poiché l’uscita Q del FF diventa metastabile, cioè non è possibile sapere quale stato logico otterrà. Il segnale di errore (ERROR) si ottiene come semplice negazione di DYN.

Il circuito che genera un clock negato e ritardato serve per creare una finestra di tempo all’interno della quale si capisce se avverrà o meno un errore dovuto alla transizione dello stato logico di D. Nell’immagine seguente, Figura 27, viene mostrato il comportamento dei segnali del TD in corrispondenza di possibili transizioni di D che possono generare un errore.

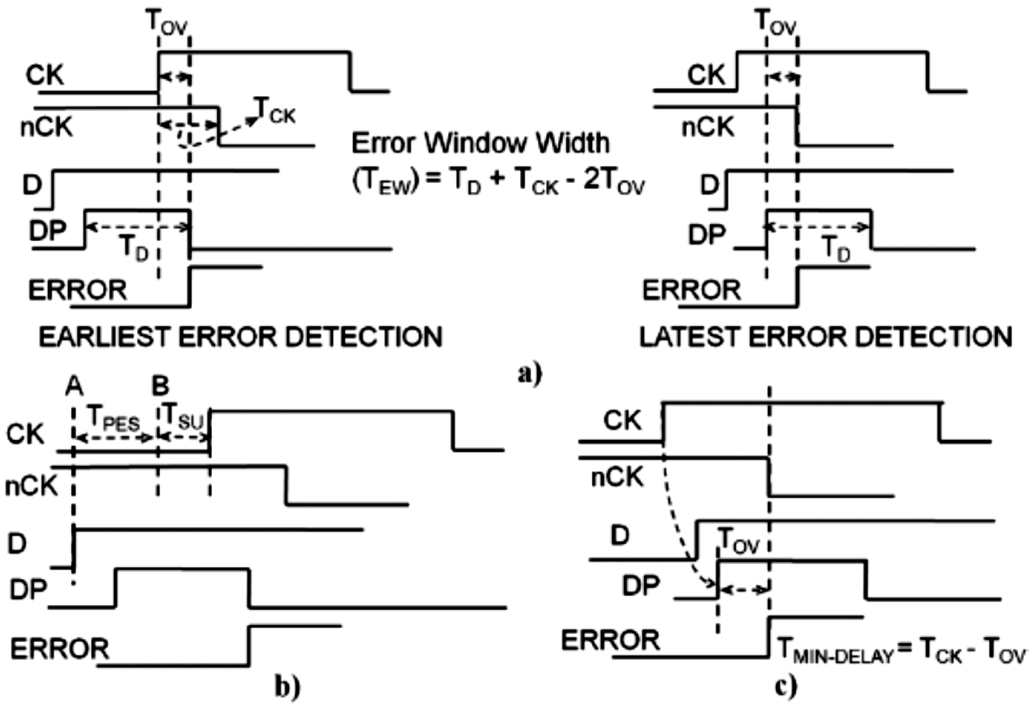

Figura 27: Diagramma temporale che illustra le operazioni del Transition Detector [5]

Si noti che viene specificata la durata della finestra di tempo, che è pari a:

$T_{EW} = T_D + T_{CK} - 2T_{OV}$

Dove T_D è la durata degli impulsi generati dalle transizioni di D, T_{CK} è pari al ritardo che si ha tra CK e nCK, infine T_{OV} è il tempo necessario per riuscire a valutare lo stato logico di ERROR.
Nell'immagine sotto viene rappresentato invece l'organizzazione interna del processore utilizzato per lo studio, Figura 28.

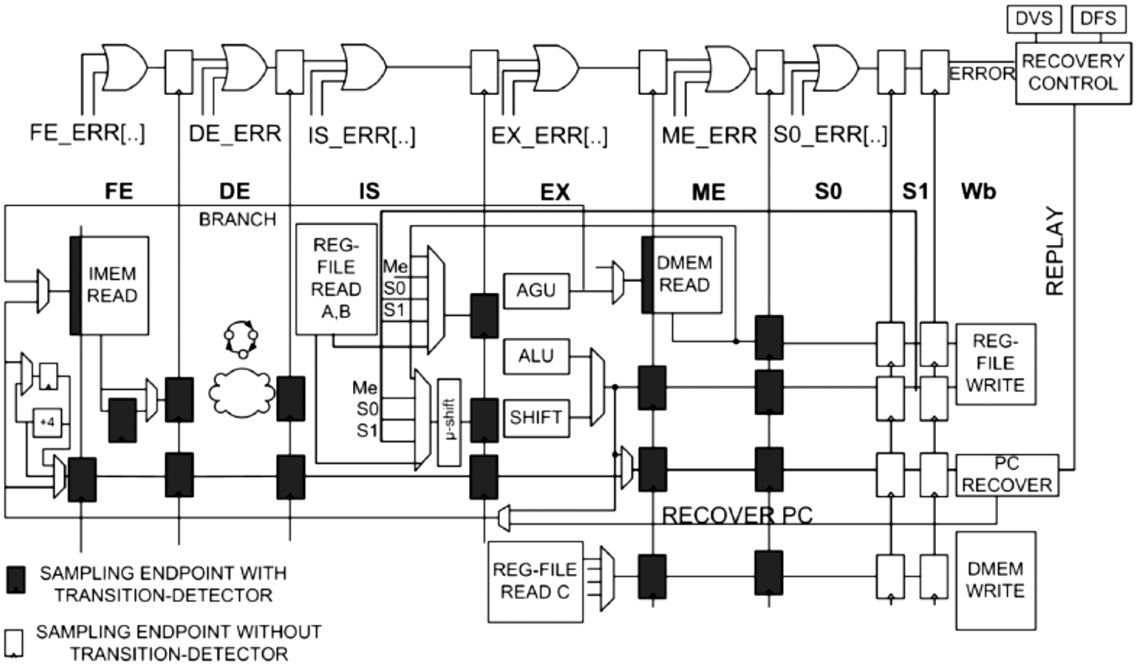

Figura 28: Diagramma della pipeline del processore ARM utilizzato che mostra i Transition Detector e il sistema di recovery [5]

Si può osservare che il processore implementa una pipeline a 6 stadi:

- Fetch
- Decode
- Issue
- Execute
- Memory
- Write-back

Tutti gli stadi della pipeline sono bilanciati in modo che tutti gli stadi abbiano dei percorsi critici con ritardi simili. Si può notare come tutti i segnali di errore all'interno di ogni stadio siano messi in OR tra loro, cosicché basti un solo un segnale a notificare l'errore in uno stadio della pipeline. Successivamente l'uscita di ogni OR degli errori di uno stadio è messo in OR con quelli dello stadio successivo. Il segnale risultante viene mandato al blocco Recovery Control. Quest'ultimo blocco quando viene segnalato un errore ha il compito di attuare il meccanismo di recovery durante la quale: la pipeline viene svuotata e l'istruzione e ripetuta, onde evitare l'incorrere di un altro errore viene ripetuta esattamente a metà del ciclo di clock.
Come ultima cosa si noti come il blocco di Recovery sia collegato ad altri due blocchi che sono: il Dynamic Voltage Scaling (DVS) e il Dynamic Frequency Scaling (DFS) di cui si spiegherà la loro utilità successivamente.

Nella Figura 29 viene riportato un grafico che mostra il comportamento di uno degli 87 processori presi in considerazione.

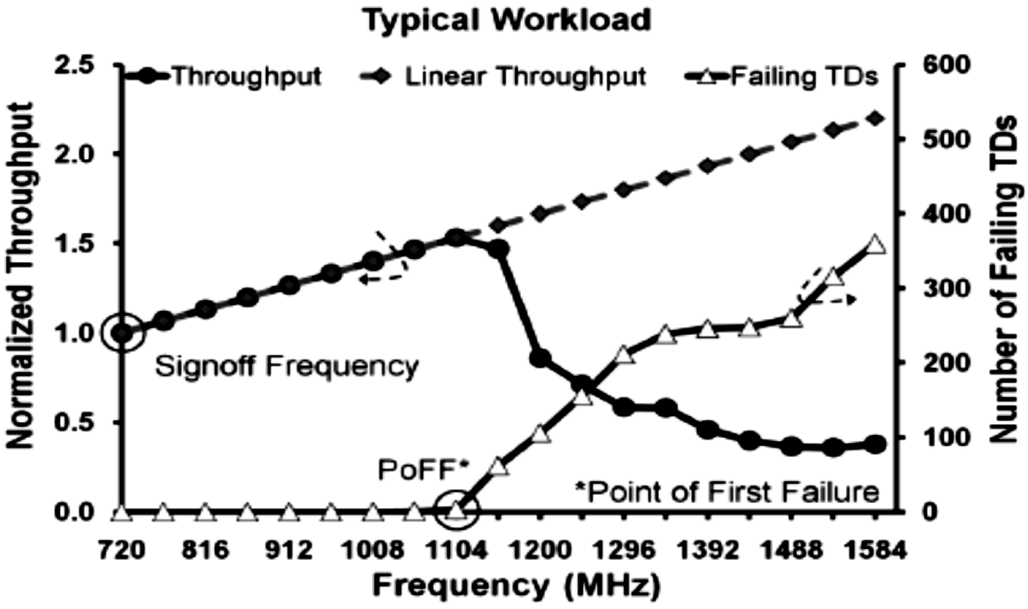

Figura 29: Caratteristica che lega il throughput con la frequenza, cerchi. Caratteristica che lega la frequenza e il numero di Transition Detector che segnalano errori, tringoli [5]

Si noti che sull'asse delle ascisse vi è la frequenza, sull'ordinata di sinistra il throughput normalizzato (ovvero il numero di volte che il processore ha terminato il codice a quella frequenza) e in quella di destra il numero di TD che hanno notificato un errore.
Il grafico contenente i triangoli lega l'ascissa con l'ordinata di destra, quello a pallini con l'ordinata di sinistra.

Si può notare come per questo processore il throughput normalizzato è aumentato linearmente con la frequenza fino a circa 1,104GHz oltre la quale si trova il PoFF (Point of First Failure, ovvero il punto a cui si verifica la prima segnalazione di un errore da parte del transition detector) e il numero di fallimenti del processore aumenta a tal punto da peggiorarne le prestazioni a frequenze superiori.

L'Adaptive Frequency Scaling (AFC) modifica la frequenza di lavoro del processore in base al numero di fallimenti dei TD, viene mostrato come questo lega la frequenza di lavoro anche al tipo di codice che viene eseguito. Qui sotto vi sono tre immagini.

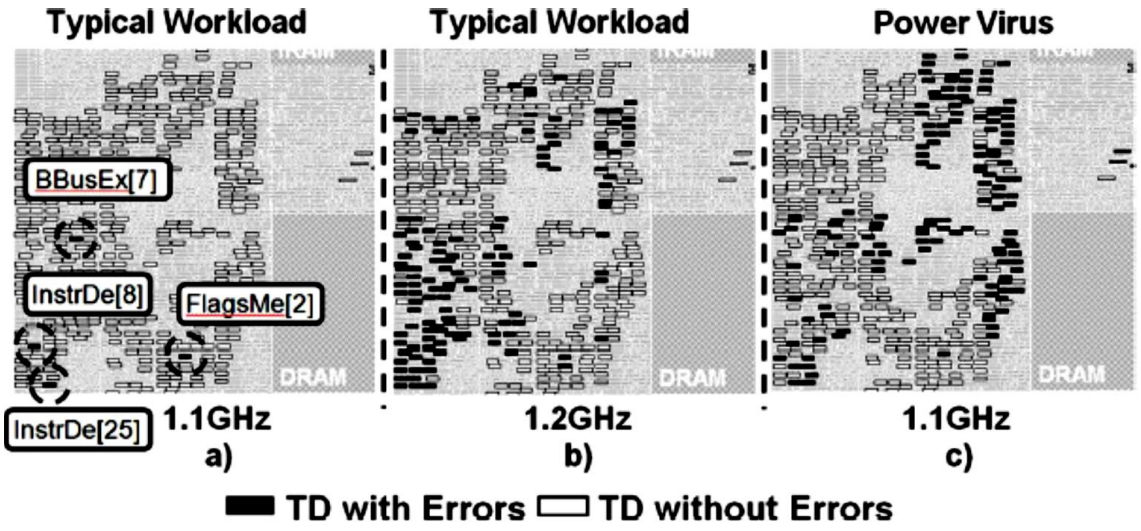

Figura 30: Mappa dei Transition Detector che segnalano errori sul chip TT a 1V. (a) Errore segnalato da 4 Transition Detector con carico di lavoro tipico ad una frequenza di 1,1GHz. (b) Errore segnalato da 122 Transition Detector con carico di lavoro tipico ad una frequenza di 1,2GHz. (c) (b) Errore segnalato da 249 Transition Detector con il Power Virus ad una frequenza di 1,1GHz. [5]

Le prime due mostrano il numero di TD che falliscono per un tipico carico di lavoro a frequenze diverse; nella terza viene mostrato come risponde il processore all'esecuzione di un Power Virus, ovvero un programma appositamente scritto per stressare pesantemente il processore, ad una frequenza operativa uguale alla prima immagine. Si capisce da qua che i TD sono legati anche al tipo di carico che il processore deve sostenere.

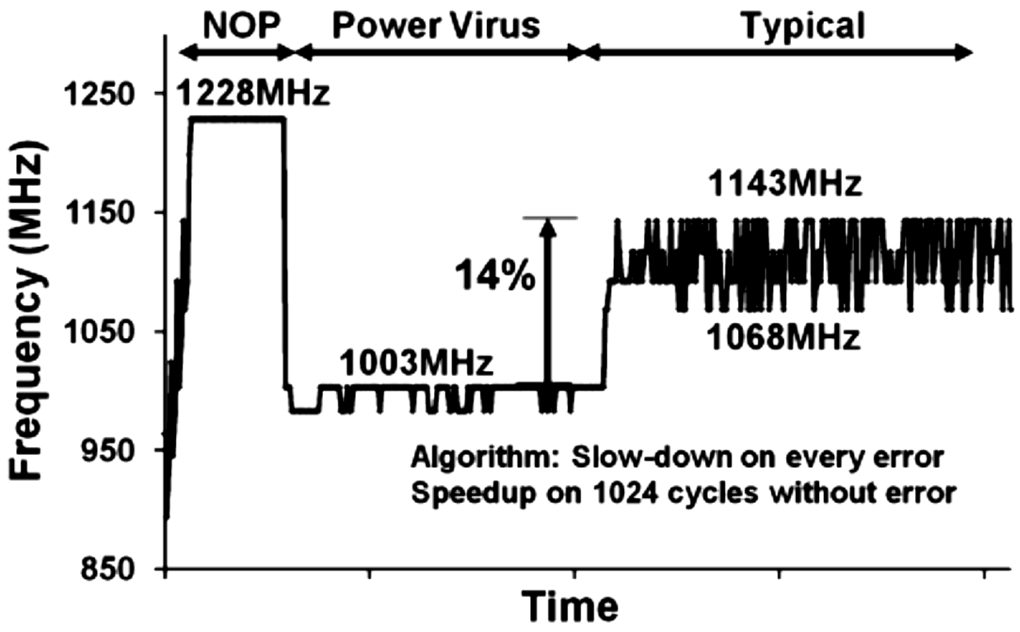

Figura 31: Risposta del blocco Dynamic Frequency Scaling(DFS) per tre fasi di carico diverse eseguite tutte a 1V. L'incremento o la riduzione di frequenza è di 24MHz alla volta. [5]

Viene infine riportata quest'immagine (sopra) che mostra il comportamento dell'AFC per tre diversi tipi di codici: il Typical Workload, il Power Virus e l'assenza di operazioni (NOP, No OPeration).

In aggiunta all'AFC è stato inserito anche un Dynamic Voltage Scaling (DVS) gestito tramite software da un processore a parte, il Razor Processor. La decisione del valore di tensione da utilizzare è basato sulla misurazione del rapporto d'errore fatto su 100 campioni di cui è tenuta traccia tramite un registro.
Viene riportata un'immagine (Figura 32) del comportamento del DVS, con frequenza costante a 1GHz, per tre diversi processori, dove l'SS è il più lento e FF è il più veloce.

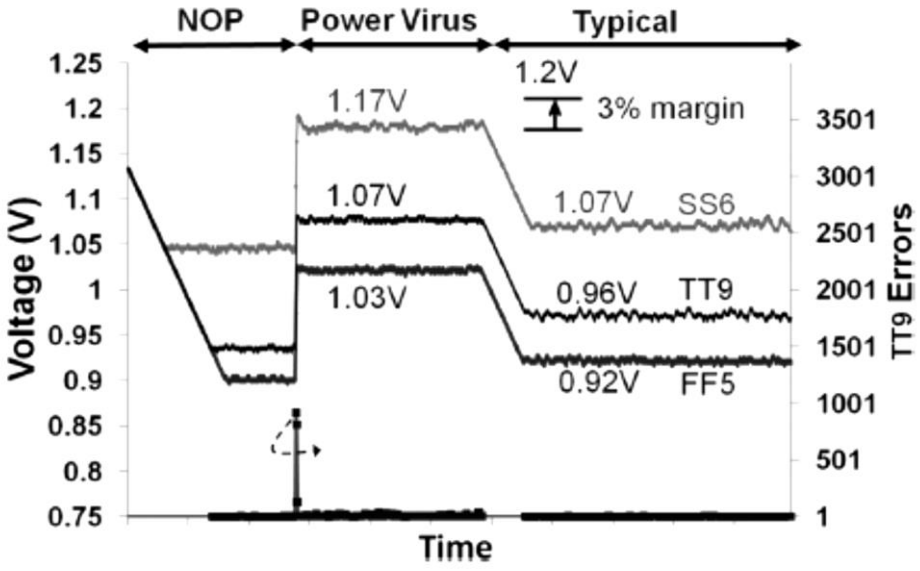

Figura 32: Impatto delle variazioni di processo sul controllore di tensione di Razor per tre diversi chip che lavorano tutti ad una frequenza di 1GHz. [5]

Considerato che da progetto il processore avrebbe dovuto lavorare ad una tensione di 1V e che, per un carico di lavoro tipico anche il componente più lento si avvicina a quella tensione si vede subito l'efficacia dell'implementazione del TD.

Nella seguente immagine viene riportato il consumo di potenza per i tre componenti testati, si noti come i risultati ottenuti da Razor siano ottimi confrontati con l'utilizzo a tensione costante.

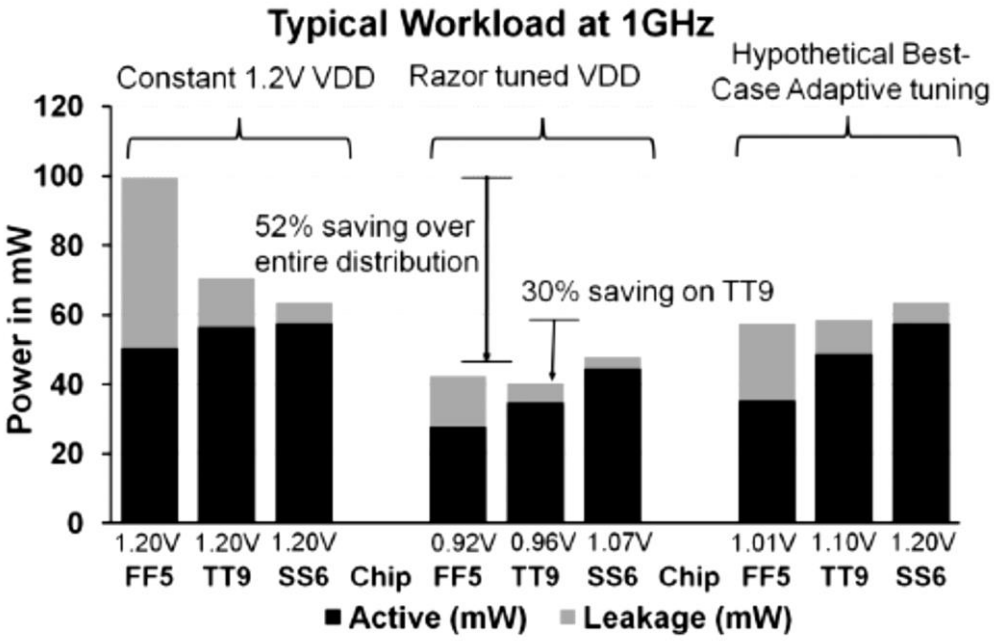

Figura 33: Consumo di potenza di tre diversi chip che lavorano tutti a 1GHz ma con tecniche di gestione della tensione differti.[5]

Viene dichiarato dagli autori dell'articolo che utilizzando Razor hanno ottenuto un risparmio di energia pari al 52% su una distribuzione di 87 componenti e di questi 87, la resa ottenuta è del 44% con AVS mentre con Razor si ottiene una resa pari al 65%.

3.2.2.ReVIVaL

ReVIVaL [24] è una tecnica che combina due tecniche di modifica dopo la fabbricazione:

- Variable Latency (VL)
- Voltage Interpolation (VI)

L'idea di base della Variable Latency, o Latenza Variabile, è di rendere il periodo di attesa del risultato di una specifica unità, della microarchitettura, modificabile mantenendo inalterata la frequenza globale del sistema. Se alcune unità mostrano delle perdite in velocità a causa delle variazioni, VL può aumentarne il tempo necessario al completamento dell'operazione. Il vantaggio di VL sta nel mantenimento del sincronismo dell'intero sistema a una sola frequenza. Nell'immagine riportata sotto viene mostrato il concetto base di VL per una pipeline con stadi multipli come una Floating Point Unit (FPU).

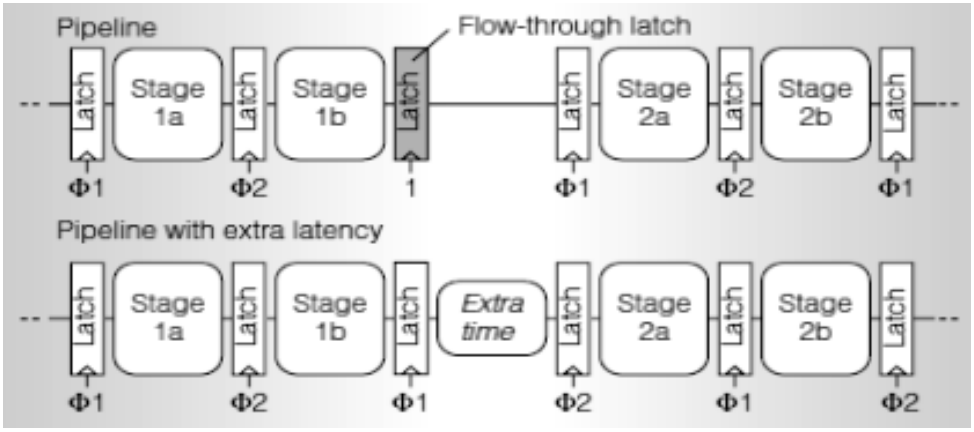

Figura 34: 2 esempi di pipeline con (sotto) e senza (sopra) uno stadio extra di pipeline [25]

La pipeline superiore a differenza di quella inferiore ha una latenza preimpostata, e quindi non modificabile. Nella seconda pipeline vi è uno stadio aggiuntivo che permette all'unità precedente di terminare la sua operazione in più cicli di clock (di solito un ciclo in più è sufficiente).

La Voltage Interpolation è secondaria, infatti da sola non può risolvere i problemi delle variazioni. Si basa sul rendere disponibile per ogni stadio della pipeline due diverse tensioni VDDH e VDDL, rispettivamente una più alta e una più bassa. Ogni stadio della pipeline così può utilizzare o una o l'altra tensione a seconda del risultato che si vuole

ottenere: usare una tensione più alta significa rendere il circuito più veloce e quindi introdurre meno ritardi, una tensione più bassa significa introdurre più ritardi e quindi rallentare lo stadio. L'immagine a seguire mostra come sia possibile modificare la tensione effettiva che utilizza l'intera unità.
Nell'articolo preso in considerazione vengono applicati entrambi i metodi visti ad una FPU a precisione singola, prodotta con un processo produttivo a 130nm. La pipeline di questa unità e composta da sei stadi, aggiungendone uno extra si è arrivato a sette, e ogni stadio della pipeline può selezionare due diverse tensioni.
Nell'immagine sotto è riportato un grafico che rappresenta i risultati ottenuti.

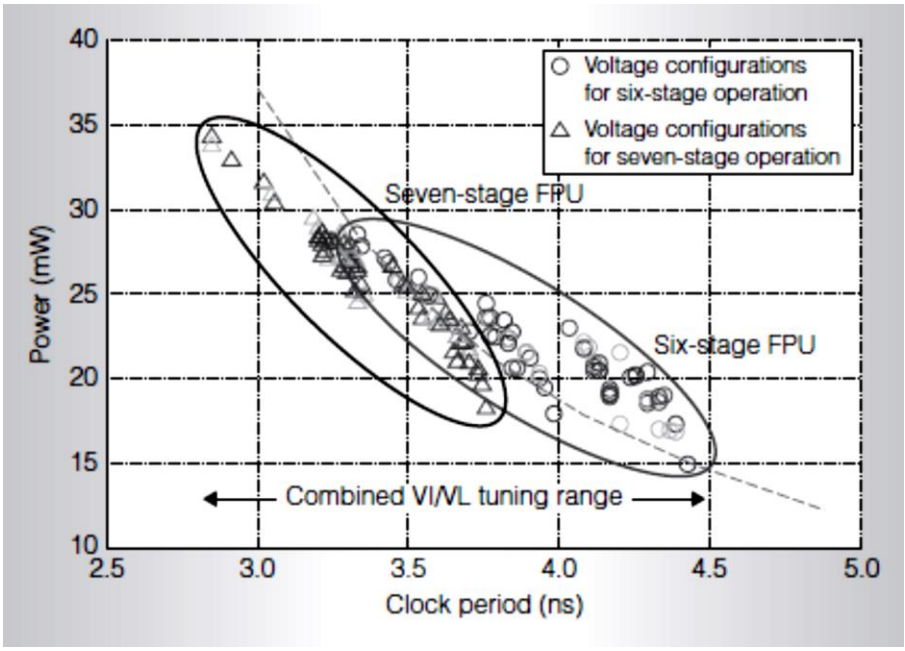

Figura 35: Risultati ottenuti applicando ReVIVaL ad una FPU a sei stadi [25]

La FPU a 7 stadi ha un consumo di potenza maggiore ma quella a 6 stadi lavora con periodi di clock più ampi, cioè a frequenze minori. Nonostante il consumo di potenza sia una cosa fondamentale è altresì vero che anche la velocità di lavoro lo è, soprattutto se si parla di unità che devono fare molti calcoli, quali la FPU o la ALU.

3.2.3.Trifecta

Trifecta [15] è una tecnica che partiziona gli input per ogni stadio in due insiemi in modo che un insieme di input possano terminare le operazioni in un ciclo mentre gli altri le terminano in due. Questa tecnica è normalmente applicata in quegli stadi della pipeline più critici, cioè quelli che determinano la frequenza di lavoro, infatti il periodo di clock del processore è normalmente più piccolo del ritardo che si ha nei percorsi critici. Le operazioni a doppio ciclo vengono attivate solamente quando si ha una combinazione in ingresso che andrebbe a utilizzare i percorsi con un ritardo superiore al periodo di clock. Quando devono essere eseguite le operazioni a doppio ciclo, ci si deve assicurare che:

- Gli input non cambino dopo un ciclo di clock
- Il risultato uscente dopo un ciclo di clock non sia utilizzato dallo stadio successivo

Invece di migliorare i componenti in termini di potenza e prestazioni, questa tecnica permette di aumentare la resa mantenendo inalterati gli altri due aspetti.
Gli autori del testo mostrano come applicando questa tecnica ad un processore la perdita che si ottiene nella velocità di esecuzione di un benchmark sia molto piccola (del 2% per calcoli in virgola mobile e del 5% per quelli di tipo intero) mentre si ottenga un aumento del 20% della resa.

Conclusione

Nei capitoli precedenti abbiamo visto un insieme delle principali fonti di variazioni che hanno lentamente spostato l'attenzione dal problema della miniaturizzazione al trovare delle soluzioni che permettano la riduzione o la compensazione di queste variazioni. Questa non sarebbe la prima volta che il settore della progettazione di circuiti elettronici digitali (in particolare dei processori) subisce un cambiamento degli obbiettivi da raggiungere. In anni precedenti si è passati dalla corsa ai GHz (ovvero lo studio di organizzazioni che permettessero di raggiungere frequenze elevate) alla corsa ai Core (ovvero lo studio di organizzazioni che permettessero di raggiungere un'efficienza maggiore). Nell'articolo riportato sulla rivista Nature dal titolo "The chips are down for Moore's law" (http://www.nature.com/news/the-chips-are-down-for-moore-s-law-1.19338), 8 Settembre 2016, viene spiegato come presto la legge di Moore diventerà solo una linea guida del passato poiché, anche se la miniaturizzazione del transistor potesse spingersi a processi produttivi a 2-3 nm, a quel livello il comportamento degli atomi sarà dominato da incertezze di natura quantistica (perché si arriverà ad avere componenti delle dimensioni di circa 10 atomi). Nello stesso articolo e nel sito web di TSMC (azienda leader nel settore della produzione di semiconduttori e circuiti integrati) (http://www.tsmc.com/english/dedicatedFoundry/technology/mtm.htm), 10 Settembre 2016, viene riportato che ci si sta spostando dalla legge di Moore all'approccio "More than Moore" (MtM): ovvero, anziché progettare chip migliori e lasciare che le applicazioni si adattino per poterli sfruttare, si inizierà dalle applicazioni e a ritroso si capirà quali chip sono necessari per sostenerle.

È facilmente pensabile che questo capitolo di storia, dopo anni di continua innovazione, stia giungendo ad un capolinea in termini di stabilità, così permettendo di focalizzarsi sullo sviluppo di nuove tecnologie, che possano innovare ulteriormente il settore. D'altra parte, però, la mancanza di tecnologie alternative che possano prendere il posto di quella attualmente utilizzata in tempi brevi rende le ricerche attualmente in corso (ovvero quelle sulle tecniche di gestione delle variazioni) ancora necessarie.

Bibliografia

[1] A. Sheikholeslami, "Process Variation and Pelgrom's Law [Circuit Intuitions]," in *IEEE Solid-State Circuits Magazine*, vol. 7, no. 1, pp. 8-9, winter 2015.

[2] Anantha Chandrakasan, William J. Bowhill, Frank Fox, 2000, *"DESIGN OF HIGH-PERFORMANCE MICROPROCESSOR CIRCUITS"*, Wiley-IEEE Press.

[3] B. Nikolic and L. t. Pang, "Measurements and analysis of process variability in 90nm CMOS," *2006 8th International Conference on Solid-State and Integrated Circuit Technology Proceedings*, Shanghai, 2006, pp. 505-508.

[4] C. Shin, X. Sun and T. J. K. Liu, "Study of Random-Dopant-Fluctuation (RDF) Effects for the Trigate Bulk MOSFET," in *IEEE Transactions on Electron Devices*, vol. 56, no. 7, pp. 1538-1542, July 2009.

[5] D. Bull, S. Das, K. Shivashankar, G. S. Dasika, K. Flautner and D. Blaauw, "A Power-Efficient 32 bit ARM Processor Using Timing-Error Detection and Correction for Transient-Error Tolerance and Adaptation to PVT Variation," in *IEEE Journal of Solid-State Circuits*, vol. 46, no. 1, pp. 18-31, Jan. 2011.

[6] Dieter K. Schroder, "Negative bias temperature instability: What do we understand?", *Microelectronics Reliability 47,* 2007, pp. 841-852.

[7] Frank Dehlmet, "Adaptive (Dynamic) Voltage (Frequency) Scaling – Motivation and Implementation", *SLVA646*, March 2014, pp.1-10.

[8] G. Timp et al., "The relentless march of the MOSFET gate oxide thickness to zero", *Microelectronics Reliability 40,* 2000, pp.557-562.

[9] Giovanni Bruni, <<*Dissipazione di Potenza nei Circuiti CMOS: Origini e Tecniche per la Riduzione*>>, Facoltà di ingegneria, Università di Padova, 2011.

[10] K. Bowman *et al*., "Circuit techniques for dynamic variation tolerance," *Design Automation Conference, 2009. DAC '09. 46th ACM/IEEE*, San Francisco, CA, 2009, pp. 4-7.

[11] K. Kang, K. Kim and K. Roy, "Variation Resilient Low-Power Circuit Design Methodology using On-Chip Phase Locked Loop," *2007 44th ACM/IEEE Design Automation Conference*, San Diego, CA, 2007, pp. 934-939.

[12] Kelin Kuhn et al., "Managing Process Variation in Intel's 45nm CMOS Technology", *Intel Technology Journal*, May2008, Vol. 12 Issue 2, p93-109. 17p.

[13] Martin Wirnshofer, 2013, *Variation-Aware Adaptive Voltage Scaling for Digital CMOS Circuits*, Springer Netherlands.

[14] Michael Liu et al., "Leakage Power Reduction by Dual-Vth Designs Under Probabilistic Analysis of Vth Variation," *Low Power Electronics and Design, 2004.*

ISLPED '04. Proceedings of the 2004 International Symposium on, Newport Beach, CA, USA, 2004, pp. 2-7.

[15] P. Ndai, N. Rafique, M. Thottethodi, S. Ghosh, S. Bhunia and K. Roy, "Trifecta: A Nonspeculative Scheme to Exploit Common, Data-Dependent Subcritical Paths," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 1, pp. 53-65, Jan. 2010.

[16] Q. A. Khan, G. K. Siddhartha, D. Tripathi, S. Kumar Wadhwa and K. Misri, "Techniques for on-chip process voltage and temperature detection and compensation," *19th International Conference on VLSI Design held jointly with 5th International Conference on Embedded Systems Design (VLSID'06)*, 2006, pp. 6 pp.-.

[17] Richard C. Jaeger, Travis N. Blalock, *Microelectronic Circuit Design, Fourth Edition*, McGraw-Hill.

[18] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi and V. De, "Parameter variations and impact on circuits and microarchitecture," *Design Automation Conference, 2003. Proceedings*, Anaheim, CA, 2003, pp. 338-342.

[19] S. Ghosh and K. Roy, "Parameter Variation Tolerance and Error Resiliency: New Design Paradigm for the Nanoscale Era," in *Proceedings of the IEEE*, vol. 98, no. 10, pp. 1718-1751, Oct. 2010.

[20] S. Ghosh, S. Bhunia and K. Roy, "CRISTA: A New Paradigm for Low-Power, Variation-Tolerant, and Adaptive Circuit Synthesis Using Critical Path Isolation," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 11, pp. 1947-1956, Nov. 2007.

[21] Seung Hoon Choi, B. C. Paul and K. Roy, "Novel sizing algorithm for yield improvement under process variation in nanometer technology," *Design Automation Conference, 2004. Proceedings. 41st*, San Diego, CA, USA, 2004, pp. 454-459.

[22] Simone Buso, 2015, *Introduzione alle applicazioni industriali di MICROCONTROLLORI E DSP,* Italia, Bologna: Esculapio.

[23] Sparsh Mittal. 2016. “A Survey of Architectural Techniques for Managing Process Variation. “, ACM *Comput. Surv.* 48, 4, Article 54 (February 2016), 29 pages.

[24] X. Jiang, R. Wang, T. Yu, J. Chen and R. Huang, "Investigations on Line-Edge Roughness (LER) and Line-Width Roughness (LWR) in Nanoscale CMOS Technology: Part I–Modeling and Simulation Method," in *IEEE Transactions on Electron Devices*, vol. 60, no. 11, pp. 3669-3675, Nov. 2013.

[25] X. Liang, G. Y. Wei and D. Brooks, "Revival: A Variation-Tolerant Architecture Using Voltage Interpolation and Variable Latency," in *IEEE Micro*, vol. 29, no. 1, pp. 127-138, Jan.-Feb. 2009.

Ringraziamenti

Desidero ringraziare tutti quelli che mi hanno aiutato nella realizzazione della tesi.

Ringrazio il prof. Paolo Magnone per la disponibilità, la cordialità dimostrata e l'aiuto fornito.
Ringrazio la mia famiglia che in questi anni mi ha sempre permesso di dedicarmi allo studio.
Ringrazio i miei compagni di università che mi hanno accompagnato e hanno condiviso con me questi anni.
Ringrazio Luca per avermi spronato a impegnarmi.

Ringrazio i miei amici, in particolare: Paolo, Francesco, Matteo, Simone e Nicola che in questi anni mi hanno aiutato a crescere.

Ringrazio i miei colleghi di lavoro che hanno reso le giornate più leggere.

Ringrazio Dio che mi aiuta e mi sostiene sempre.

Infine ringrazio Alice per: la sua pazienza, la sua disponibilità, la sua energia, la sua dolcezza e la sua serietà. Senza di lei non avrei mai raggiunto questo traguardo.

Tutte le persone citate in questa pagina hanno svolto un ruolo fondamentale nella stesura della tesi, ma desidero precisare che ogni errore o imprecisione è imputabile soltanto a me