

Appunti di Fondamenti di elettronica

Giacomo Simonetto

Primo semestre 2025-26

Sommario

Appunti del corso di Fondamenti di elettronica della facoltà di Ingegneria Informatica dell'Università di Padova.

Indice

1 Introduzione	5
1.1 Definizioni fondamentali	5
1.2 Settori dell'elettronica	5
1.3 Richiamo di teoria dei circuiti	6
1.4 Reti in regime transitorio	8
2 Semiconduttori	10
2.1 Classificazione e proprietà elettriche dei semiconduttori	10
2.2 Drogaggio dei semiconduttori	11
2.3 Corrente elettrica e conducibilità nei semiconduttori	13
2.4 Semiconduttori in equilibrio	14
3 Giunzione PN e diodi	16
3.1 Giunzione pn all'equilibrio	16
3.2 Giunzione pn polarizzata	19
3.3 Giunzione pn polarizzata vista come diodo	20
3.4 Applicazioni speciali dei diodi	21
4 Condensatore MOS o CMOS	23
4.1 Struttura e funzionamento	23
4.2 Condensatore mos con substrato di tipo p	23
4.3 Analisi del p-cmos in condizioni di svuotamento/inversione	24
4.4 cmos con substrato di tipo n e differenze rispetto al p-cmos	25
5 Transistor MOSFET	26
5.1 Struttura generale e classificazione dei mosfet	26
5.2 Struttura di un n-mosfet e vincoli sui potenziali	27
5.3 Aree di lavoro di un n-mosfet	28
5.4 Curve caratteristiche di corrente-tensione di un n-mosfet	30
5.5 Modello a canale corto di un n-mosfet	31
5.6 Struttura di un p-mosfet e vincoli sui potenziali	32
5.7 Aree di lavoro di un p-mosfet	33
5.8 Curve caratteristiche di corrente-tensione di un p-mosfet	33
5.9 Modello a canale corto di un p-mosfet	34
5.10 Simbologia e rappresentazione circuitale dei mosfet	35
5.11 Struttura reale del mosfet	35
5.12 Effetto Body e variazione della tensione di soglia	36
5.13 Corrente di sottosoglia	36
5.14 Capacità parassite dei mosfet	37
6 Circuiti con i mosfet	38
6.1 Mosfet in serie a una resistenza	38
6.2 Mosfet connesso a diodo	39
6.3 Mosfet come generatore di corrente	40
6.4 Mosfet usato come interruttore	42
6.5 Scarica di un condensatore con nmos	42
6.6 Carica di un condensatore con pmos	44
6.7 Scarica di un condensatore con un pmos	46
6.8 Carica di un condensatore con un nmos	48
6.9 Confronto dei transistori per carica e scarica con nmos e pmos	50
6.10 Resistenza equivalente del mosfet come interruttore	51
6.11 Reti di mosfet e resistenza equivalente complessiva	52

7 Breve cenno ai segnali digitali	53
7.1 Segnali binari logici ed elettrici	53
7.2 Circuiti e porte logiche	53
7.3 Parametri dinamici dei circuiti logici	54
8 Invertitore CMOS	55
8.1 Schema circuitale	55
8.2 Caratteristica di trasferimento ingresso-uscita	56
8.3 Soglia di commutazione logica	57
8.4 Tolleranza al rumore	57
8.5 Invertitore ideale	58
8.6 Tempo di ritardo	59
8.7 Consumo statico	61
8.8 Consumo dinamico	61
8.9 Oscillatore ad anello	62
8.10 Buffer cmos	63
9 Logica statica complementare	65
9.1 Reti duali	65
9.2 Implementazione di porte logiche elementari	66
9.3 Funzioni invertenti e porte logiche non invertenti	67
9.4 Tempi di ritardo	67
9.5 Problema dei nodi interni	69
9.6 Consumo di potenza	70
10 Logica a Pass Transistor	71
10.1 Struttura base	71
10.2 Trasmissione dei valori logici cattivi	72
10.3 Ottimizzazione dei valori logici - level restorer e transmission gate	73
10.4 Tempi di propagazione e ottimizzazioni	74
10.5 Consumo dinamico	75
10.6 Costruzioni di porte logiche	75
11 Fabbricazione dei circuiti integrati	77
11.1 Fasi di progettazione e costruzione	77
11.2 Fabbricazione del Wafer con metodo di Czochralski	77
11.3 Processo selettivo di costruzione del circuito integrato per litografia	78
11.4 Selezione delle regioni attive	79
11.5 Selezione del tipo di substrato	79
11.6 Ossido di gate ed elettrodo di gate	80
11.7 Diffusioni N+ e P+	80
11.8 Contatti e interconnessioni	81
11.9 Packaging	81
11.10 Layout, regole di layout e sviluppo delle maschere	82
12 Blocchi logico aritmetici	83
12.1 Struttura di una ALU	83
12.2 Operatori logici elementari	83
12.3 Blocchi di traslazione e rotazione	83
12.4 Sommatore ripple carry	84
12.5 Varianti del sommatore	84
12.6 Moltiplicatore	85
12.7 Sommatore (e sottrattore) floating point	86
12.8 Moltiplicatore floating point	87

13 Ottimizzazione del sommatore	88
13.1 Ottimizzazione del full adder con generate e propagate	88
13.2 Architettura Ripple Carry con FA ottimizzato	89
13.3 Architettura Carry Bypass - CBP	89
13.4 Architettura Linear Carry Select - LCS	90
13.5 Architettura Square Root Carry Select - SRCS	90
13.6 Architettura Carry Look Ahead - CLA	91
13.7 Architetture a confronto	92
13.8 Moltiplicatore con sommatore ottimizzato in architettura CLA	93
14 Latch e Flip Flop	94
14.1 Circuiti logici combinatori e sequenziali	94
14.2 Latch	94
14.3 Flip flop	95
14.4 Registri di flip flop	97
14.5 Disallineamento del clock	97
15 Circuiti sequenziali	98
15.1 Architettura pipeline	98
15.2 Macchine a stati	98
15.3 Esempi di macchine a stati - registri a scorrimento	99
15.4 Esempi di macchine a stati	99
16 Clock e sincronizzazione	100
16.1 Distribuzione del clock	100
16.2 Generazione del clock e Phase Locked Loop - PLL	101
16.3 Struttura della rete di distribuzione del clock	102
17 Memorie a semiconduttore	103
17.1 Introduzione alle memorie a semiconduttore	103
17.2 Matrice di celle per memorie ROM	103
17.3 Matrice di celle per memorie non volatili e scrivibili	104
17.4 Matrice di celle per memorie RAM	106
17.5 Tempi di accesso	107
17.6 Circuiti periferici	108
18 Scaling e consumo	109
18.1 Storia e trend di evoluzione	109
18.2 Strategie di scaling	110
18.3 Compromesso tra velocità e consumo	111
19 Circuiti logici programmabili	113
19.1 Introduzione ai circuiti logici programmabili	113
19.2 Logica programmabile a matrice	114
19.3 Logica programmabile a celle - FPGA	114
20 Hardware per alto sforzo computazionale - GPU, ASIC, TPU	117
20.1 Esempi di elaborazione dati ad alto sforzo computazionale	117
20.2 GPU - Graphical Processing Unit	117
20.3 FPGA - Field Programmable Gate Array	118
20.4 ASIC - Application Specific Integrated Circuit	118
20.5 TPU - Tensor Processing Unit	118

1 Introduzione

1.1 Definizioni fondamentali

- **elettronica:** studia e realizza sistemi elettronici;
- **sistema elettronico:** è un insieme di componenti elettronici (sensori, circuiti e attuatori) che raccolgono informazioni dal mondo reale attraverso sensori, le elaborano attraverso circuiti elettronici e prendono decisioni o comandano azioni con degli attuatori;
- **segnaile:** supporto fisico di natura qualunque (elettrica, acustica, ottica) a cui si associa un'informazione allo scopo di poterla trasferire da una sorgente ad un utilizzatore, può essere digitale (ampiezza e tempo discreti) o analogico (ampiezza e tempo continui);
- **sensore:** dispositivo che converte un segnale esterno (come temperatura, pressione, luce, suono) in una grandezza elettrica (come corrente o tensione);
- **circuito elettronico:** rete di componenti elettrici passivi (R, L, C) e attivi (diodi, transistor) che elaborano segnali elettrici (tensione e corrente). In base al tipo di segnale elaborato si distingue in:
 - **circuito analogico:** elabora segnali analogici;
 - **circuito digitale:** elabora segnali digitali;
 - **circuito misto:** opera in entrambi i domini del segnale.

Siccome i segnali provenienti dal mondo reale sono sempre analogici, in generale non esiste un sistema completamente digitale. Ogni sistema digitale, infatti, comprende un ADC (Analog-to-Digital Converter) in ingresso e un DAC (Digital-to-Analog Converter) in uscita.

In base alla realizzazione fisica, un circuito elettronico si distingue in:

- **circuito a elementi discreti:** realizzato con componenti costruiti separatamente che poi vengono montati su un supporto (breadboard, PCB) e collegati tra loro tramite fili o piste conduttrive;
- **circuito integrato (IC):** tutti i componenti sono miniaturizzati e vengono montati su un unico chip di silicio (es. microchip).

Un sistema elettronico completo è formato da circuiti integrati e componenti discreti montati in una scheda in cui sono realizzate le interconnessioni metalliche tra i terminali dei componenti

1.2 Settori dell'elettronica

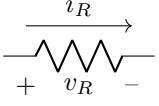
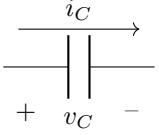
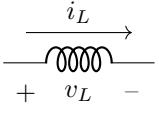
- **elettronica analogica:** progettazione e analisi di circuiti che elaborano segnali analogici;
- **elettronica digitale:** progettazione e analisi di circuiti che elaborano segnali digitali;
- **elettronica di consumo:** dispositivi elettronici per l'uso personale e domestico (computer, telefoni cellulari, televisori, elettrodomestici);
- **microelettronica:** progettazione e fabbricazione di componenti elettronici e circuiti integrati;
- **elettronica di potenza:** conversione e gestione dell'energia elettrica a diversi livelli (dal riscaldamento agli alimentatori per pc, cellulari o altri strumenti);
- **elettronica industriale:** sistemi elettronici per processi produttivi automatizzati;
- **telecomunicazioni:** sistemi per la trasmissione di dati (voce, video, file) attraverso dispositivi mobili o fissi;
- **biomedica:** sviluppo di apparecchiature elettroniche per la diagnostica, la cura e il monitoraggio della salute;
- **automotive:** sistemi per il controllo dei veicoli (dallo specchietto fino alla guida autonoma);
- **informatica:** dispositivi e sistemi elettronici per la gestione dei dati.

1.3 Richiamo di teoria dei circuiti

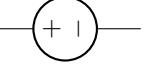
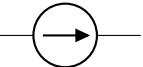
Leggi di Kirchhoff

- **Legge delle correnti (LKC):** la somma delle correnti entranti in un nodo è uguale alla somma delle correnti uscenti.
- **Legge delle tensioni (LKT):** la somma delle tensioni lungo una maglia è uguale a zero.

Elementi passivi

resistore (R):		$v_R(t) = R \cdot i_R(t)$	$i_R(t) = \frac{v_R(t)}{R}$
condensatore (C):		$i_C(t) = C \frac{dv_C(t)}{dt}$	$v_C(t) = \frac{1}{C} \int_0^t i_C(t) dt + \frac{I_0}{C}$
induttore (L):		$v_L(t) = L \frac{di_L(t)}{dt}$	$i_L(t) = \frac{1}{L} \int_0^t v_L(t) dt + \frac{V_0}{L}$

Elementi attivi

generatore ideale di tensione (GIT):		fornisce una tensione costante indipendentemente dalla corrente che lo attraversa
generatore ideale di corrente (GIC):		fornisce una corrente costante indipendentemente dalla tensione ai suoi capi
diodi e transistor:	...	componenti non lineari che verranno studiati successivamente.

Principi di analisi dei circuiti

- **partitore di tensione:** due resistori in serie dividono la tensione in ingresso V_{in} in due tensioni V_1 e V_2 direttamente proporzionali alle resistenze (inversamente proporzionali alle conduttanze):

$$V_1 = V_{in} \cdot \frac{R_1}{R_1 + R_2} \quad V_2 = V_{in} \cdot \frac{R_2}{R_1 + R_2}$$

- **partitore di corrente:** due resistori in parallelo dividono la corrente in ingresso I_{in} in due correnti I_1 e I_2 inversamente proporzionali alle resistenze (direttamente proporzionali alle conduttanze):

$$I_1 = I_{in} \cdot \frac{R_2}{R_1 + R_2} \quad I_2 = I_{in} \cdot \frac{R_1}{R_1 + R_2}$$

- **sovraposizione degli effetti:** dato un sistema lineare con C_1 , C_2 possibili ingressi ed E_1 , E_2 effetti prodotti in uscita dai due ingressi, se il sistema viene perturbato con un ingresso dato dalla composizione lineare dei due ingressi $C = p_1C_1 + p_2C_2$ con p_1 e p_2 pesi dei due ingressi, l'effetto risultante in uscita sarà la composizione lineare dei due effetti $E = p_1E_1 + p_2E_2$.

In particolare in un circuito lineare con più generatori, la risposta (tensione o corrente) in un componente è uguale alla somma algebrica delle risposte dovute a ciascun generatore preso singolarmente, con gli altri generatori sostituiti dai loro rispettivi cortocircuiti (generatore di tensione ideale) o circuiti aperti (generatore di corrente ideale).

Potenziali, tensioni e nodi di riferimento

- Il potenziale elettrico è definito a meno di una costante, per cui anche la soluzione di una rete elettrica (data dai potenziali ai vari nodi) non è univoca, ma è definita a meno di una costante.
- Per rendere univoca la soluzione, si sceglie un nodo di riferimento a cui si assegna potenziale nullo e si calcolano i potenziali degli altri nodi rispetto a tale nodo.
- Le tensioni, invece, sono sempre definite univocamente come differenze di potenziale tra due nodi.

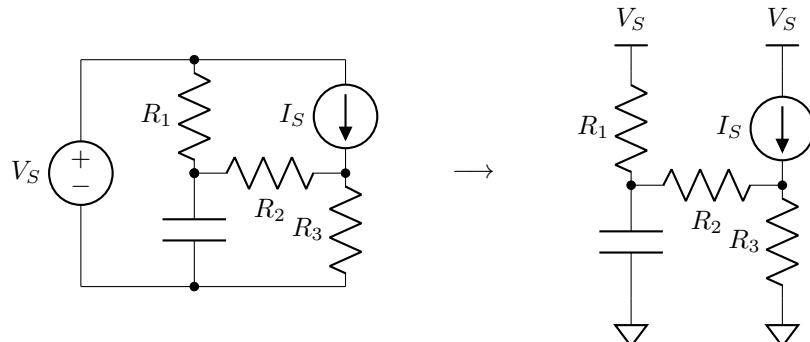
Esistono tre tipi di nodi di riferimento comunemente usati:

nodo con potenziale di riferimento:		nodo con il potenziale di riferimento scelto arbitrariamente a 0V
nodo di massa:		nodo collegato al telaio metallico del dispositivo elettronico
nodo di terra:		nodo collegato fisicamente alla terra tramite un conduttore metallico per motivi di sicurezza; di solito coincide con il nodo di massa; il potenziale di terra è molto stabile e indipendentemente dalle correnti che gli elettrodomestici prelevano o immettono in esso

Rappresentazione elettronica di un circuito

Un circuito elettronico può essere rappresentato in due modi equivalenti:

- **notazione a maglie:** rappresentazione del circuito in maglie e nodi
- **notazione elettronica:** scelto il nodo di riferimento, tutti i terminali collegati a tale nodo sono marcati con il simbolo del nodo di riferimento, inoltre i nodi di cui si conosce già il potenziale (ad esempio quelli collegati a generatori di tensione ideali) sono marcati con il loro valore di potenziale.



Potenza ed energia

Per definire la potenza e l'energia consumata da un componente, si definiscono:

- **convenzione degli utilizzatori:** la corrente entra nel terminale positivo della tensione, se la potenza o l'energia è positiva, il componente assorbe energia.
- **convenzione dei produttori:** la corrente entra nel terminale negativo della tensione, se la potenza o l'energia erogata è positiva, il componente fornisce energia.

Si definiscono quindi:

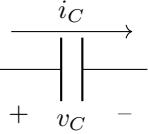
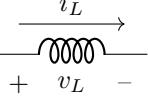
- **potenza istantanea:** $p(t) = v(t) \cdot i(t)$ misurata in Watt [W] = [J/sec]
- **energia:** $E = \int_{t_1}^{t_2} p(t) dt = \int_{t_1}^{t_2} v(t) \cdot i(t) dt$ misurata in Joule [J] = [W · sec]

1.4 Reti in regime transitorio

Introduzione

- una rete si dice in regime transitorio quando le variabili elettriche (tensione e corrente) variano nel tempo passando da uno stato iniziale a uno stato finale di equilibrio
- un esempio di reti in regime transitorio sono i circuiti in cui sono presenti componenti reattivi (condensatori e induttori) e interruttori che modificano la configurazione del circuito
- il transitorio è l'intervallo di tempo che impiegano le variabili elettriche per passare dallo stato iniziale allo stato finale di equilibrio

Componenti reattivi

componente	schema	in regime stazionario	in regime transitorio
condensatore:		circuito aperto $i_C = 0$	$i_C(t) = C \frac{dv_C(t)}{dt}$
induttore (L):		cortocircuito $v_L = 0$	$v_L(t) = L \frac{di_L(t)}{dt}$

Carica di un condensatore

- condizioni iniziali ($t < 0$): interruttore inizialmente aperto $\rightarrow v_C(0) = 0, i_C(0) = 0$
- nel transitorio ($t \geq 0$): $i_C(t) = C \frac{dv_C(t)}{dt}, i_R = \frac{v_R(t)}{R}, V_A = v_R(t) + v_C(t), i_R(t) = i_C(t)$
- dalla legge delle correnti si ottiene un'equazione differenziale del primo ordine:

$$C \frac{dv_C(t)}{dt} = \frac{v_R(t)}{R} = \frac{V_A - v_C(t)}{R} \quad \rightarrow \quad \frac{dv_C(t)}{dt} = -\frac{v_C(t)}{RC} + \frac{V_A}{RC} \quad \rightarrow \quad v_C(t) = A \cdot e^{-\frac{t}{RC}} + B$$

- si sostituisce la soluzione generale nell'equazione differenziale e si impongono le condizioni iniziali:

$$-\frac{1}{RC}A \cdot e^{-\frac{t}{RC}} = -\frac{1}{RC}A \cdot e^{-\frac{t}{RC}} - \frac{1}{RC}B + \frac{V_A}{RC} \quad \rightarrow \quad B = V_A$$

$$v_C(0) = 0 \quad \rightarrow \quad A + B = 0 \quad \rightarrow \quad A = -B = -V_A$$

- si ottengono quindi le espressioni delle variabili elettriche durante il transitorio:

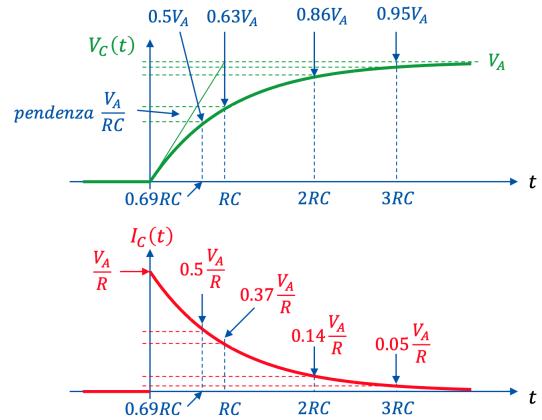
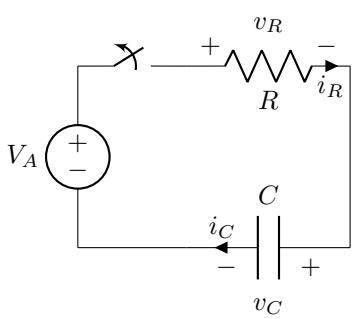
$$v_C(t) = V_A - V_A \cdot e^{-\frac{t}{RC}} \quad v_R(t) = V_A e^{-\frac{t}{RC}} \quad i_C(t) = i_R(t) = \frac{V_A}{R} e^{-\frac{t}{RC}}$$

- l'istante in cui la tensione sul condensatore raggiunge metà del suo valore di regime è:

$$\frac{V_A}{2} = V_A - V_A \cdot e^{-\frac{t_{1/2}}{RC}} \quad \rightarrow \quad t_{1/2} = \ln(2) \cdot RC \approx 0.69RC$$

- analizzando il bilancio energetico del circuito si ottiene che metà dell'energia fornita dal generatore viene immagazzinata nel condensatore e metà viene dissipata dalla resistenza come calore:

$$E_{V_A} = \int_0^\infty V_A \cdot i(t) dt = C \cdot V_A^2 \quad E_R = \int_0^\infty R \cdot i^2(t) dt = \frac{C \cdot V_A^2}{2} \quad E_C = \int_0^\infty v_C(t) \cdot i(t) dt = \frac{C \cdot V_A^2}{2}$$



Scarica di un condensatore

- condizioni iniziali ($t < 0$): interruttore inizialmente aperto $\rightarrow v_C(0) = V_A, i_C(0) = 0$
- nel transitorio ($t \geq 0$): $i_C(t) = C \frac{dv_C(t)}{dt}, i_R = \frac{v_R(t)}{R}, v_R(t) = v_C(t), i_R(t) + i_C(t) = 0$
- dalla legge delle correnti si ottiene un'equazione differenziale del primo ordine:

$$C \frac{dv_C(t)}{dt} = \frac{v_R(t)}{R} \rightarrow \frac{dv_C(t)}{dt} = -\frac{v_C(t)}{RC} \rightarrow v_C(t) = A \cdot e^{-\frac{t}{RC}}$$

- siccome l'equazione è omogenea, è sufficiente imporre le condizioni iniziali:

$$v_C(0) = V_A \rightarrow A = V_A$$

- si ottengono quindi le espressioni delle variabili elettriche durante il transitorio:

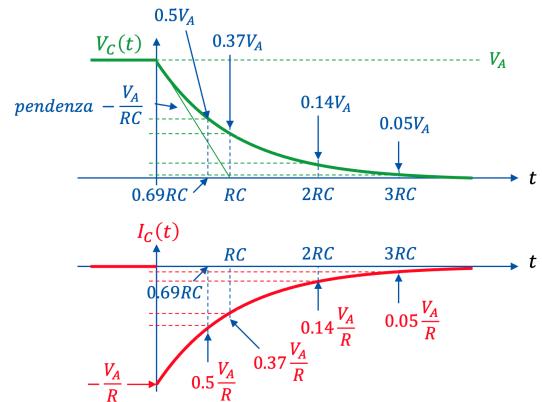
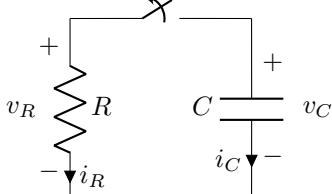
$$v_C(t) = v_R(t) = V_A \cdot e^{-\frac{t}{RC}} \quad i_C(t) = -\frac{V_A}{R} e^{-\frac{t}{RC}} \quad i_R(t) = \frac{V_A}{R} e^{-\frac{t}{RC}}$$

- l'istante in cui la tensione sul condensatore raggiunge metà del suo valore di regime è:

$$\frac{V_A}{2} = V_A - V_A \cdot e^{-\frac{t_{1/2}}{RC}} \rightarrow t_{1/2} = \ln(2) \cdot RC \approx 0.69RC$$

- analizzando il bilancio energetico del circuito si ottiene tutta l'energia immagazzinata nel condensatore viene dissipata dalla resistenza come calore e il condensatore rimane scarico alla fine del transitorio:

$$E_R = \int_0^\infty R \cdot i_R^2(t) dt = \frac{C \cdot V_A^2}{2} \quad E_C = \int_0^\infty v_C(t) \cdot i_C(t) dt = -\frac{C \cdot V_A^2}{2}$$



2 Semiconduttori

2.1 Classificazione e proprietà elettriche dei semiconduttori

Classificazione dei materiali

In base alla propria resistività elettrica ρ , ogni materiale si classifica in:

- conduttori: $\rho < 10^{-3} \Omega \cdot \text{m}$
- semiconduttori: $10^{-3} \Omega \cdot \text{m} < \rho < 10^5 \Omega \cdot \text{m}$
- isolanti: $\rho > 10^5 \Omega \cdot \text{m}$

Classificazione chimica dei semiconduttori

I semiconduttori sono composti dagli elementi chimici di transizione, come il silicio (Si) e il germanio (Ge). In base alla loro composizione chimica si classificano in:

- **semiconduttori a elemento singolo**: se sono formati da un solo elemento chimico, come silicio (Si) e germanio (Ge)
- **semiconduttori composti**: se sono formati da più elementi, come arseniuro di gallio (GaAs), fosfuro di indio (InP), nitruro di gallio (GaN), tellururo di cadmio (CdTe), ...
- **semiconduttori intrinseci**: se sono puri, cioè non contengono impurità

Struttura del silicio

Il silicio (Si) ha 14 protoni, 14 neutroni e 14 elettroni. Possiede 4 elettroni nel guscio più esterno, detti elettroni di valenza. Tali elettroni sono quelli che partecipano alla formazione dei legami chimici e di conseguenza il silicio è in grado di formare 4 legami covalenti con altri atomi. In un cristallo di silicio, ogni atomo di silicio condivide i suoi 4 elettroni di valenza con 4 atomi di silicio vicini, formando così una struttura cristallina tetraedrica e periodica (simile a quella del carbonio).

Conducibilità elettrica nei metalli, negli isolanti e nei semiconduttori

Un materiale conduce corrente elettrica se possiede elettroni liberi in grado di muoversi all'interno del reticolo cristallino. La differenza tra metalli, isolanti e semiconduttori risiede nel comportamento degli elettroni di valenza:

- nei metalli, gli elettroni di legame sono condivisi tra più atomi e formano una nube di elettroni che si muovono liberamente all'interno del reticolo metallico, permettendo così la conduzione elettrica
- negli isolanti, gli elettroni di valenza sono fortemente vincolati ai loro atomi e non possono muoversi liberamente all'interno del reticolo cristallino, per cui sono scarsi conduttori di elettricità
- nei semiconduttori si hanno comportamenti simili agli isolanti, però i legami tra gli atomi sono più deboli e con poca energia è possibile rompere tali legami liberando gli elettroni di valenza, che possono così condurre corrente elettrica

L'energia necessaria a liberare un elettrone di valenza è detta **energy gap** (o band gap) e varia a seconda del materiale. Di seguito una tabella con i valori di energy gap per alcuni materiali comuni:

Materiale	Energy gap	Materiale	Energy gap
Silicio (Si)	1.124 eV	Germanio (Ge)	0.66 eV
Arseniuro di gallio (GaAs)	1.42 eV	Nitruro di gallio (GaN)	3.4 eV
Fosfuro di indio (InP)	1.35 eV	Seleniuro di cadmio (CdSe)	1.74 eV
Stagno (Sn)	0.082 eV	Rame (Cu)	"0" eV

Elettroni liberi e lacune nei semiconduttori

Quando un elettrone di valenza acquisisce sufficiente energia per liberarsi dal legame con il suo atomo, esso diventa un elettrone libero in grado di muoversi liberamente all'interno del reticolo cristallino. Il legame incompleto per la mancanza di un elettrone è detto lacuna (o hole in inglese). La lacuna può essere colmata da un elettrone di valenza di un atomo vicino, che a sua volta lascia una nuova lacuna.

In questo modo, la lacuna sembra muoversi all'interno del reticolo cristallino, creando così una corrente elettrica. Gli elettroni liberi sono detti portatori di carica negativa, mentre le lacune sono dette portatori di carica positiva.

Concentrazione di elettroni e lacune in un semiconduttore intrinseco

In un semiconduttore intrinseco, la concentrazione di elettroni liberi n e la concentrazione di lacune p sono uguali e per definizione si indicano con n_i . La concentrazione di portatori di carica in un semiconduttore intrinseco dipende dalla temperatura T e dall'energy gap E_g del materiale, secondo la formula a destra, dove B è una costante dipendente dal materiale e k_B è la costante di Boltzmann.

$$n = p = n_i \quad n \cdot p = n_i^2 \quad n_i = BT^{3/2} e^{-\frac{E_g}{2k_B T}}$$

Si osserva che la concentrazione di portatori di carica aumenta all'aumentare della temperatura. Nel silicio a temperatura ambiente (300 K), la concentrazione di portatori di carica è circa $n_i = 1.45 \times 10^{10} \text{ cm}^{-3}$.

2.2 Drogaggio dei semiconduttori

Il drogaggio di un semiconduttore consiste nell'aggiunta di impurità al semiconduttore intrinseco per modificarne le proprietà elettriche, senza alterare la struttura del reticolo. Le impurità sono atomi di elementi chimici con un numero di elettroni di valenza diverso da quello del semiconduttore intrinseco che produrranno un eccesso di elettroni liberi o di lacune, migliorando la conducibilità elettrica del materiale.

Drogaggio di tipo n

Il drogaggio di tipo n si ottiene aggiungendo al silicio intrinseco atomi di un elemento chimico con 5 elettroni di valenza, come il fosforo (P), l'arsenico (As) o l'antimonio (Sb). Questi atomi, facendo 4 legami covalenti con gli atomi di silicio vicini, hanno un elettrone in più che non può essere utilizzato per il legame e diventa un elettrone libero. Per questo motivo sono detti donatori.

Drogaggio di tipo p

Il drogaggio di tipo p si ottiene aggiungendo al silicio intrinseco atomi di un elemento chimico con 3 elettroni di valenza, come il boro (B), l'alluminio (Al) o il gallio (Ga). Questi atomi fanno sempre 4 legami covalenti con gli atomi di silicio vicini, ma avendo un elettrone in meno, creano una lacuna. Per questo motivo sono detti accettori.

Generazione e ricombinazione di coppie elettrone-lacuna

In un semiconduttore si hanno due processi opposti che avvengono contemporaneamente:

- la **generazione** di coppie elettrone-lacuna, per cui si formano un elettrone libero e una lacuna dovuti alla rottura di un legame covalente; si indica con $G = f_1(T)$ il tasso di generazione, ovvero il numero di coppie elettrone-lacuna generate per unità di volume e di tempo ($\text{[cm}^{-3}\text{s}^{-1}\text{]}$), che dipende dalla temperatura T
- la **ricombinazione** di coppie elettrone-lacuna, in cui un elettrone libero si ricombina con una lacuna; si indica con $R = n p f_2(T)$ il tasso di ricombinazione, ovvero il numero di coppie elettrone-lacuna che si ricombinano per unità di volume e di tempo ($\text{[cm}^{-3}\text{s}^{-1}\text{]}$) e dipende dalle concentrazioni di elettroni n e di lacune p e dalla temperatura T

Equilibrio termodinamico e legge di azione di massa

Quando il semiconduttore si trova a temperatura costante e senza sollecitazioni esterne, raggiunge uno stato di **equilibrio termodinamico** in cui:

- la concentrazione di elettroni n e la concentrazione di lacune p rimangono costanti nel tempo
- la velocità di generazione di coppie elettrone-lacuna è uguale alla velocità di ricombinazione

Imponendo l'equilibrio tra generazione e ricombinazione si ottiene la **legge di azione di massa**:

$$G = R \rightarrow f_1(T) = n \cdot p \cdot f_2(T) \rightarrow n \cdot p = \frac{f_1(T)}{f_2(T)} \rightarrow n_i^2 = n \cdot p$$

Concentrazione dei portatori in un semiconduttore drogato di tipo n

In un semiconduttore drogato esclusivamente di tipo n si hanno le seguenti particelle cariche:

- elettroni liberi n (intrinseci e donati dagli atomi di impurità)
- lacune p (intrinseche)
- ioni donatori N_D^+ (atomi di impurità che hanno ceduto un elettrone libero)

Siccome il drogaggio non altera la carica complessiva del semiconduttore, si ha:

$$-qn + qp + qN_D = 0 \rightarrow n - p - N_D = 0 \rightarrow n - n_i^2/n - N_D = 0 \rightarrow n^2 - nN_D - n_i^2 = 0$$

$$n = \frac{N_D \pm \sqrt{N_D^2 + 4n_i^2}}{2} = N_D \frac{1 + \sqrt{1 + 4n_i^2/N_D^2}}{2} \xrightarrow{N_D \gg n_i} n \approx N_D, \quad p = \frac{n_i^2}{N_D}$$

La concentrazione di elettroni n è maggiore della concentrazione di lacune p , per cui gli elettroni si definiscono **portatori maggioritari**, mentre le lacune si definiscono **portatori minoritari**.

Concentrazione dei portatori in un semiconduttore drogato di tipo p

In un semiconduttore drogato esclusivamente di tipo p si hanno le seguenti particelle cariche:

- elettroni liberi n (intrinseci)
- lacune p (intrinseche e dovute agli atomi di impurità)
- ioni accettori N_A^- (atomi di impurità che hanno accettato un elettrone libero)

Siccome il drogaggio non altera la carica complessiva del semiconduttore, si ha:

$$-qn + qp - qN_A = 0 \rightarrow p - n - N_A = 0 \rightarrow p - n_i^2/p - N_A = 0 \rightarrow p^2 - pN_A - n_i^2 = 0$$

$$p = \frac{N_A \pm \sqrt{N_A^2 + 4n_i^2}}{2} = N_A \frac{1 + \sqrt{1 + 4n_i^2/N_A^2}}{2} \xrightarrow{N_A \gg n_i} p \approx N_A, \quad n = \frac{n_i^2}{N_A}$$

La concentrazione di lacune p è maggiore della concentrazione di elettroni n , per cui le lacune si definiscono **portatori maggioritari**, mentre gli elettroni si definiscono **portatori minoritari**.

Concentrazione dei portatori in un semiconduttore drogato sia di tipo n che di tipo p

In un semiconduttore drogato sia di tipo n che di tipo p si hanno le seguenti particelle cariche:

- elettroni liberi n (intrinseci e donati dagli atomi di impurità di tipo n)
- lacune p (intrinseche e dovute agli atomi di impurità di tipo p)
- ioni donatori N_D^+ (atomi di impurità di tipo n che hanno ceduto un elettrone libero)
- ioni accettori N_A^- (atomi di impurità di tipo p che hanno accettato un elettrone libero)

Siccome il drogaggio non altera la carica complessiva del semiconduttore, si ha:

$$-qn + qp + qN_D - qN_A = 0 \rightarrow n - p - N_D + N_A = 0$$

Se $N_D > N_A$ si comporta come un semiconduttore di tipo n con drogaggio netto $N'_D = N_D - N_A$:

$$n^2 - n(N_D - N_A) - n_i^2 = 0 \xrightarrow{N_D - N_A \gg n_i} n \approx N_D - N_A, \quad p = \frac{n_i^2}{N_D - N_A}$$

Viceversa se $N_A > N_D$ si comporta come un semiconduttore di tipo p con drogaggio netto $N'_A = N_A - N_D$:

$$p^2 - p(N_A - N_D) - n_i^2 = 0 \xrightarrow{N_A - N_D \gg n_i} p \approx N_A - N_D, \quad n = \frac{n_i^2}{N_A - N_D}$$

Scelta di N_D e N_A nel drogaggio dei semiconduttori

Affinché si abbia una buona riuscita del drogaggio, è necessario che la concentrazione dei droganti sia maggiore dei portatori intrinseci e maggiore della concentrazione di impurità ($> 10^{10} - 12 \text{ cm}^{-3}$) altrimenti il drogaggio non avrebbe effetto. Inoltre deve essere minore degli atomi di silicio nel reticolo cristallino ($< 10^{22} \text{ cm}^{-3}$). In genere si sceglie un drogaggio compreso tra 10^{14} cm^{-3} e 10^{19} cm^{-3} .

2.3 Corrente elettrica e conducibilità nei semiconduttori

Cause del moto dei portatori di carica

Lo spostamento dei portatori di carica nei semiconduttori è influenzata da tre meccanismi fisici:

- **temperatura:** l'aumento della temperatura provoca un aumento di energia interna (e di conseguenza di energia cinetica) dei portatori di carica, che si muovono più velocemente all'interno del reticolo cristallino
- **campo elettrico:** un campo elettrico esercita una forza sui portatori di carica che di conseguenza induce un movimento ordinato dei portatori di carica, essendo questi ultimi carichi elettricamente
- **gradiente di concentrazione:** una differenza di concentrazione di portatori di carica in due regioni del semiconduttore provoca un flusso di portatori dalla regione a concentrazione maggiore verso la regione a concentrazione minore, questo fenomeno è detto principio di diffusione

Moto dei portatori di carica dovuto alla sola energia cinetica / agitazione termica

In assenza di campo elettrico e di gradiente di concentrazione, i portatori di carica si muovono casualmente all'interno del reticolo cristallino a causa della loro energia cinetica. Questo moto casuale produce uno spostamento medio nullo, poiché i portatori di carica si muovono in tutte le direzioni con uguale probabilità. Per questo la temperatura viene definita come contributo caotico.

Moto dei portatori di carica dovuto al campo elettrico

Quando si applica un campo elettrico E al semiconduttore, i portatori di carica subiscono una forza F_E dovuta al campo elettrico che induce un movimento ordinato dei portatori di carica:

- gli elettroni liberi, essendo cariche negative, si muovono in direzione opposta al campo elettrico
- le lacune, essendo cariche positive, si muovono in direzione concorde al campo elettrico

Il moto ordinato dei portatori di carica dovuto al campo elettrico si sovrappone al moto casuale dovuto all'agitazione termica, producendo uno spostamento medio non nullo dei portatori di carica nella direzione del campo elettrico (per le lacune) o in direzione opposta al campo elettrico (per gli elettroni).

La velocità di deriva dei portatori dipende linearmente al campo elettrico (approssimando per campi non troppo elevati):

$$\begin{cases} v_n = -\mu_n \cdot E & \text{per gli elettroni} \rightarrow \text{moto opposto a } E \\ v_p = \mu_p \cdot E & \text{per le lacune} \rightarrow \text{moto concorde a } E \end{cases}$$

con v_n e v_p velocità di deriva degli elettroni e delle lacune ([cm/s]), E il campo elettrico ([V/cm]) e μ_n e μ_p le mobilità degli elettroni e delle lacune rispettivamente ([cm²/Vs]). In generale $\mu_n/\mu_p \approx 3$ siccome gli elettroni si muovono più facilmente delle lacune. Negli esercizi si assume $\mu_n = 1000 \text{ cm}^2/\text{Vs}$ e $\mu_p = 300 \text{ cm}^2/\text{Vs}$ per il silicio intrinseco a temperatura ambiente.

Corrente di deriva

Analizzando la quantità di carica che attraversa una sezione di area A in un intervallo di tempo dt , si ottiene la corrente di deriva:

$$\begin{aligned} \# \text{elettroni} &= n \cdot A \cdot dx \quad \rightarrow \quad I_n = \frac{-q \cdot \# \text{elettr.}}{dt} = \frac{-q \cdot n \cdot A \cdot dx}{dt} = -q \cdot n \cdot A \cdot v_n \quad \rightarrow \quad j_n^{\text{drift}} = \frac{I_n}{A} = -q \cdot n \cdot v_n \\ \# \text{lacune} &= p \cdot A \cdot dx \quad \rightarrow \quad I_p = \frac{q \cdot \# \text{lacune}}{dt} = \frac{q \cdot p \cdot A \cdot dx}{dt} = q \cdot p \cdot A \cdot v_p \quad \rightarrow \quad j_p^{\text{drift}} = \frac{I_p}{A} = q \cdot p \cdot v_p \\ J_{\text{tot}}^{\text{drift}} &= J_n + J_p = -q \cdot n \cdot v_n + q \cdot p \cdot v_p = q(n\mu_n + p\mu_p)E \end{aligned}$$

Si nota quindi che la densità di corrente di deriva totale $J_{\text{drift,tot}}$ è proporzionale e concorde al campo elettrico E , per una costante di proporzionalità detta resistività elettrica ρ :

$$j_{\text{tot}}^{\text{drift}} = \frac{E}{\rho}, \quad \rho = \frac{1}{q(n\mu_n + p\mu_p)} \quad \begin{cases} j_{\text{tot}}^{\text{drift}} = qN_D\mu_n E, & \rho = 1/qN_D\mu_n \quad \text{se drogato di tipo n} \\ j_{\text{tot}}^{\text{drift}} = qN_A\mu_p E, & \rho = 1/qN_A\mu_p \quad \text{se drogato di tipo p} \end{cases}$$

Moto dei portatori di carica dovuto al gradiente di concentrazione

Quando si mettono in contatto due regioni di un semiconduttore con diversa concentrazione di portatori di carica, si crea un gradiente di concentrazione che induce un flusso di portatori dalla regione a concentrazione maggiore verso la regione a concentrazione minore. Questo fenomeno è detto principio di diffusione e si verifica in natura per tutte le particelle libere di muoversi.

Per analizzare il moto dei portatori di carica dovuto al gradiente di concentrazione, si definisce il flusso per unità di area in un intervallo di tempo dt , misurato in $[\text{cm}^{-2}\text{s}^{-1}]$:

$$\phi(x) = -D \frac{dC(x)}{dx} \quad \phi(x)_n = -D_n \frac{dn(x)}{dx} \quad \phi(x)_p = -D_p \frac{dp(x)}{dx}$$

con $C(x)$ la concentrazione di particelle in funzione della posizione x ($[\text{cm}^{-3}]$) e D il coefficiente di diffusione ($[\text{cm}^2/\text{s}]$). Il segno “-” indica che il flusso avviene in direzione opposta al gradiente.

Corrente di diffusione

Analizzando la quantità di carica associata al flusso dei portatori si ottiene la corrente di diffusione:

$$j_n^{\text{diff}} = -q\phi(x)_n = +qD_n \frac{dn(x)}{dx} \quad j_p^{\text{diff}} = +q\phi(x)_p = -qD_p \frac{dp(x)}{dx} \quad [\text{C cm}^{-2}\text{s}^{-1}] = [\text{A cm}^{-2}]$$

Si osserva che il verso della densità di corrente di diffusione dipende dal tipo di portatore:

- la corrente degli elettroni ha lo stesso verso della concentrazione (cariche negative)
- la corrente delle lacune ha verso opposto alla concentrazione (cariche positive)

Corrente totale in un semiconduttore

La corrente totale in un semiconduttore è data dalla somma della corrente di deriva e della corrente di diffusione degli elettroni e delle lacune:

$$j_n = j_n^{\text{drift}} + j_n^{\text{diff}} = qn\mu_n E + qD_n \frac{dn(x)}{dx} \quad j_p = j_p^{\text{drift}} + j_p^{\text{diff}} = qp\mu_p E - qD_p \frac{dp(x)}{dx}$$

Le costanti D_n , D_p , μ_n , μ_p sono correlate tra loro dalla relazione di Einstein:

$$\frac{D_n}{\mu_n} = \frac{D_p}{\mu_p} = \frac{k_B T}{q} = V_T$$

dove k_B è la costante di Boltzmann, T la temperatura assoluta, q la carica elementare e V_T è il potenziale termico. Si ottiene quindi la seguente espressione per la corrente totale:

$$j_n = q\mu_n \left(nE + V_T \frac{dn(x)}{dx} \right) \quad j_p = qp\mu_p \left(pE - V_T \frac{dp(x)}{dx} \right) = q\mu_p \left(pE + pV_T \frac{dp(x)}{dx} \right)$$

2.4 Semiconduttori in equilibrio

Correnti in un semiconduttore in equilibrio

Quando un semiconduttore si trova in equilibrio termodinamico, la somma delle correnti di deriva e di diffusione per ciascun tipo di portatore di carica è nulla (che implica corrente totale nulla):

$$j_n = j_n^{\text{drift}} + j_n^{\text{diff}} = 0 \quad j_p = j_p^{\text{drift}} + j_p^{\text{diff}} = 0 \quad \Rightarrow \quad j_{\text{tot}} = j_n + j_p = 0$$

Gradienti di potenziale e concentrazione

Se in un semiconduttore in equilibrio si ha un gradiente di concentrazione di portatori di carica ($n_2 - n_1$ per gli elettroni o $p_2 - p_1$ per le lacune), si deve necessariamente avere un gradiente di potenziale $v_2 - v_1$ tale da bilanciare la corrente di diffusione con la corrente di deriva, in modo che la corrente totale sia nulla. Si ottiene quindi la seguente relazione:

$$\frac{n_2}{n_1} = \frac{p_1}{p_2} = e^{\frac{(v_2 - v_1)}{V_T}}$$

Viceversa se in un semiconduttore in equilibrio si ha un gradiente di potenziale $v_2 - v_1$, si deve necessariamente avere un gradiente di concentrazione di portatori di carica tale da bilanciare la corrente di deriva con la corrente di diffusione:

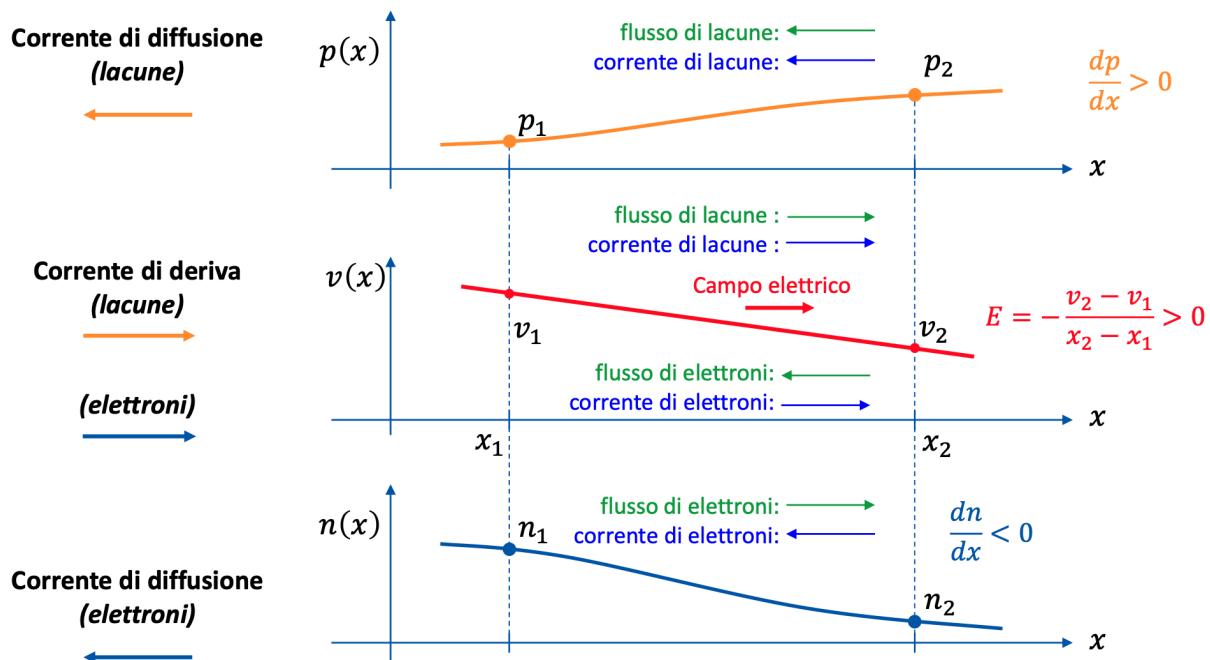
$$\frac{v_2 - v_1}{V_T} = \ln\left(\frac{n_2}{n_1}\right) = \ln\left(\frac{p_1}{p_2}\right)$$

Le due relazioni precedenti sono equivalenti e si ottengono imponendo la condizione di equilibrio $j_n = 0$ e $j_p = 0$, utilizzando $E = -dV/dx$ e risolvendo le due equazioni differenziali ottenute.

Considerazioni sulle correnti, gradienti e potenziali in equilibrio

In un semiconduttore in equilibrio termodinamico in cui è presente un gradiente di concentrazione di portatori di carica e un gradiente di potenziale, valgono le seguenti considerazioni:

1. - j_n^{drift} ha verso opposto al gradiente di potenziale e concorde con il campo elettrico
 - j_p^{drift} ha verso opposto al gradiente di potenziale e concorde con il campo elettrico
 - il gradiente di potenziale è lo stesso sia per gli elettroni che per le lacune, per cui j^{drift} ha complessivamente verso opposto al gradiente di potenziale e concorde con il campo elettrico
2. - j_n^{diff} ha verso concorde con il gradiente di concentrazione degli elettronni
 - j_p^{diff} ha verso opposto al gradiente di concentrazione delle lacune
 - i gradienti di concentrazione degli elettroni e delle lacune sono opposti tra loro (per la legge di azione di massa), per cui j^{diff} ha complessivamente verso concorde al gradiente di concentrazione degli elettroni e verso opposto al gradiente di concentrazione delle lacune
3. - siccome la corrente totale deve essere nulla, la corrente di deriva e la corrente di diffusione devono avere verso opposto da cui si conclude che:
 - il gradiente di potenziale e il gradiente di concentrazione degli elettroni hanno lo stesso verso
 - il gradiente di potenziale e il gradiente di concentrazione delle lacune hanno verso opposto



3 Giunzione PN e diodi

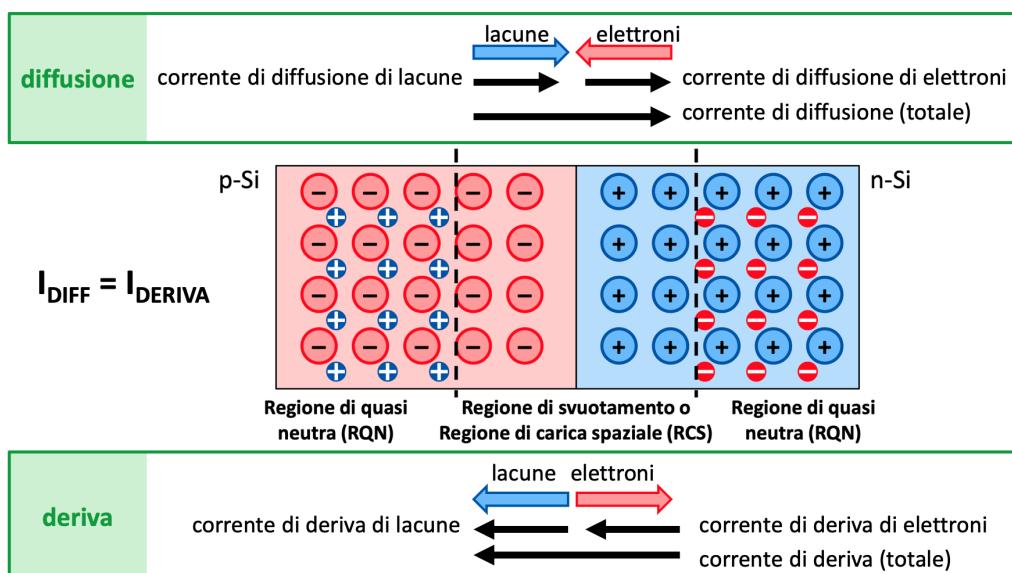
3.1 Giunzione pn all'equilibrio

Struttura base

Una giunzione pn si ottiene unendo due regioni di semiconduttore drogati in modo diverso: una regione di tipo p (con eccesso di lacune) e una regione di tipo n (con eccesso di elettroni).

Equilibrio tra diffusione e potenziale

- Quando le due regioni si uniscono, si forma un **gradiente di concentrazione** dei portatori di carica che induce uno spostamento di elettroni dalla regione n alla regione p e uno spostamento di lacune dalla regione p alla regione n; si forma in questo modo una corrente di diffusione dalla regione p alla regione n.
- Lo spostamento dei portatori induce la formazione di ioni fissi costituiti dagli atomi dei droganti: i donatori perdono il loro elettrone spaiato e diventano ioni con carica positiva nella regione n, mentre gli accettori catturano l'elettrone che gli mancava e diventano ioni con carica negativa nella regione p. Questi ioni fissi generano un campo elettrico e un **potenziale di giunzione**; si forma in questo modo anche una corrente di deriva dalla regione n alla regione p che si oppone alla corrente di diffusione.
- All'equilibrio le due correnti si bilanciano, ma rimane una regione in prossimità della giunzione sono presenti solo ioni fissi dei droganti per l'assenza di portatori di carica.
- Si formano in questo modo tre regioni:
 1. **regione di svuotamento o regione di carica spaziale (RCS)**: zona in prossimità della giunzione priva di portatori di carica liberi (svuotamento) in cui sono presenti solo ioni fissi (carica spaziale);
 2. **regione quasi neutra (RQN) di tipo p**: zona lontana dalla giunzione che non risente della giunzione pn e mantiene le caratteristiche di un semiconduttore di tipo p;
 3. **regione quasi neutra (RQN) di tipo n**: zona lontana dalla giunzione che non risente della giunzione pn e mantiene le caratteristiche di un semiconduttore di tipo n.



NOTA: i seguenti calcoli si riferiscono alle grandezze per unità di superficie di giunzione Σ_j . Per cui per ottenere ad esempio la densità di carica elettrica effettiva è necessario moltiplicare la densità di carica elettrica per unità di superficie utilizzata nei calcoli per la superficie di giunzione: $\rho_{\text{eff}} = \rho \cdot \Sigma_j$.

Carica elettrica all'equilibrio

La carica elettrica nelle regioni quasi neutre è nulla, siccome non vengono alterate le concentrazioni di portatori di carica liberi (e il drogaggio non modifica la carica complessiva). Nella regione di svuotamento, invece, la carica elettrica è data dalla somma delle cariche degli ioni fissi e dipende dalle concentrazioni di drogaggio N_A e N_D :

$$\rho(x) = \begin{cases} -qN_A & -x_p \leq x \leq 0 \quad (\text{regione p}) \\ +qN_D & 0 < x \leq x_n \quad (\text{regione n}) \end{cases} \rightarrow Q_p = -qN_A x_p, \quad Q_n = +qN_D x_n$$

Siccome non ci sono stati scambi di cariche con l'esterno, la carica totale deve rimanere nulla:

$$Q_p + Q_n = 0 \Rightarrow N_A x_p = N_D x_n$$

Campo elettrico all'equilibrio

Il campo elettrico nella regione di carica spaziale si calcola:

$$\frac{dE(x)}{dx} = -\frac{\rho(x)}{\varepsilon} = \begin{cases} -qN_A/\varepsilon & -x_p \leq x \leq 0 \\ +qN_D/\varepsilon & 0 < x \leq x_n \end{cases} \Rightarrow E(x) = \begin{cases} -qN_A(x + x_p)/\varepsilon & -x_p \leq x \leq 0 \\ +qN_D(x_n - x)/\varepsilon & 0 < x \leq x_n \end{cases}$$

Il campo elettrico è nullo nelle regioni quasi neutre e raggiunge il valore massimo in $x = 0$:

$$E_{max} = E(0) = -\frac{qN_A x_p}{\varepsilon} = -\frac{qN_D x_n}{\varepsilon}$$

Potenziale elettrico all'equilibrio e potenziale di contatto

La differenza di potenziale si calcola integrando il campo elettrico, per la relazione $dV(x)/dx = -E(x)$. In particolare si definiscono i potenziali nelle due regioni quasi neutre V_1 in $x = -x_p$ e V_2 in $x = x_n$ e si calcola il potenziale intrinseco di giunzione o potenziale di contatto $V_0 = V_2 - V_1$ tra le due estremità della regione di svuotamento ponendo $V_1 = 0$:

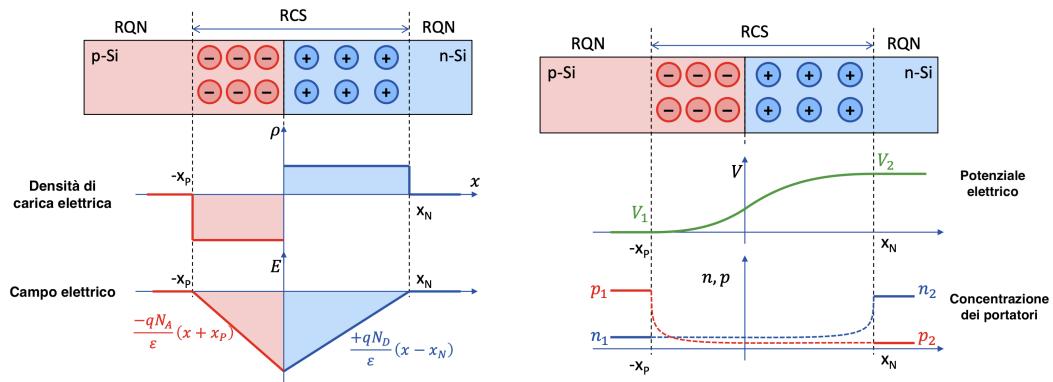
$$V_0 = \int_{-x_p}^{x_n} E(x) dx = \frac{-E(0) \cdot (x_n + x_p)}{2}$$

Potenziale di contatto e concentrazioni di drogaggio

Il potenziale di contatto può essere espresso in funzione delle concentrazioni di drogaggio N_A e N_D e delle concentrazioni intrinseche di portatori di carica n_i :

$$V_0 = V_2 - V_1 = V_T \ln \left(\frac{n_2}{n_1} \right) = V_T \ln \left(\frac{N_A N_D}{n_i^2} \right)$$

Schema riassuntivi per una giunzione pn in equilibrio



Aampiezza della regione di svuotamento all'equilibrio

L'ampiezza della regione di svuotamento $W = x_n + x_p$ dipende dalle concentrazioni dei drogaggi e dal potenziale di contatto:

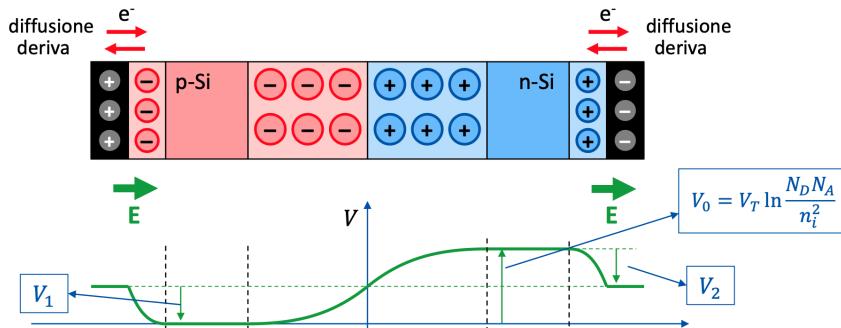
$$V_0 = \frac{-E(0) \cdot W}{2}, \quad x_n N_D = x_p N_A \quad \rightarrow \quad W = \frac{2V_0}{-E(0)} = \frac{2\varepsilon V_0}{q N_A x_p} = \sqrt{\frac{2\varepsilon V_0}{q} \left(\frac{1}{N_A} + \frac{1}{N_D} \right)}$$

$$x_n = W \frac{N_A}{N_A + N_D} \quad x_p = W \frac{N_D}{N_A + N_D}$$

Si osserva che l'ampiezza della regione di svuotamento è inversamente proporzionale alle concentrazioni di droggaggio, per cui aumentando i drogaggi diminuisce l'ampiezza della regione di svuotamento. Inoltre la regione di svuotamento si ripartisce in maniera inversamente proporzionale ai drogaggi, ovvero si allarga maggiormente nella regione meno drogata.

Regione pn con elettrodi metallici

Una giunzione pn, per essere utilizzabile in un circuito, deve essere collegata alle due estremità a due elettrodi metallici. Gli elettrodi, essendo buoni conduttori, inducono una locale ridistribuzione dei portatori di carica (spostamento di lacune dalla regione p all'elettrodo e di elettroni dalla regione n all'elettrodo). Questo effetto ha le stesse dinamiche di una giunzione pn, in particolare si formano due regioni di carica spaziale che inducono una differenza di potenziale tra gli elettrodi (con potenziali V_1 e V_2) e la giunzione che controbilanciano il potenziale (o meglio tensione) di contatto V_0 . All'equilibrio si ha $V_1 + V_0 + V_2 = 0$, per cui la differenza di potenziale degli estremi di una giunzione è nulla.



3.2 Giunzione pn polarizzata

Polarizzazione diretta e inversa

Si collega una giunzione pn ad un generatore di tensione V_A con polo positivo connesso alla regione p e polo negativo alla regione n. In questo modo si ha una polarizzazione della giunzione che può essere:

- **polarizzazione diretta** se $V_A > 0$
- **polarizzazione inversa** se $V_A < 0$

Inoltre si definisce il potenziale di riferimento come il potenziale della regione n ($V_n = 0V$). In questo modo la regione p ha un potenziale $V_p = -V_0$, uguale ed opposto al potenziale intrinseco o di contatto.

Giunzione in polarizzazione diretta

In polarizzazione diretta la tensione applicata $V_A > 0$ riduce la differenza di potenziale tra le due regioni quasi neutre, ottenendo: $V_p - V_n = -V_0 + V_A$. Di conseguenza il campo elettrico nella regione di svuotamento diminuisce in modulo e l'ampiezza della regione di svuotamento si riduce:

$$W(V_0 - V_A) = \sqrt{\frac{2\epsilon(V_0 - V_A)}{q} \left(\frac{1}{N_A} + \frac{1}{N_D} \right)} < W(V_0)$$

Siccome il campo elettrico e il potenziale si riducono, la corrente di deriva diminuisce e prevale il fenomeno di diffusione che induce un flusso di elettroni dalla regione n alla regione p e di lacune dalla regione p alla regione n. Siccome la regione n è ricca di elettroni e la regione p è ricca di lacune, questo flusso è detto flusso dei portatori maggioritari. Si induce in questo modo una **elevata corrente di diffusione dei maggioritari** che attraversa la giunzione pn dalla regione p alla regione n.

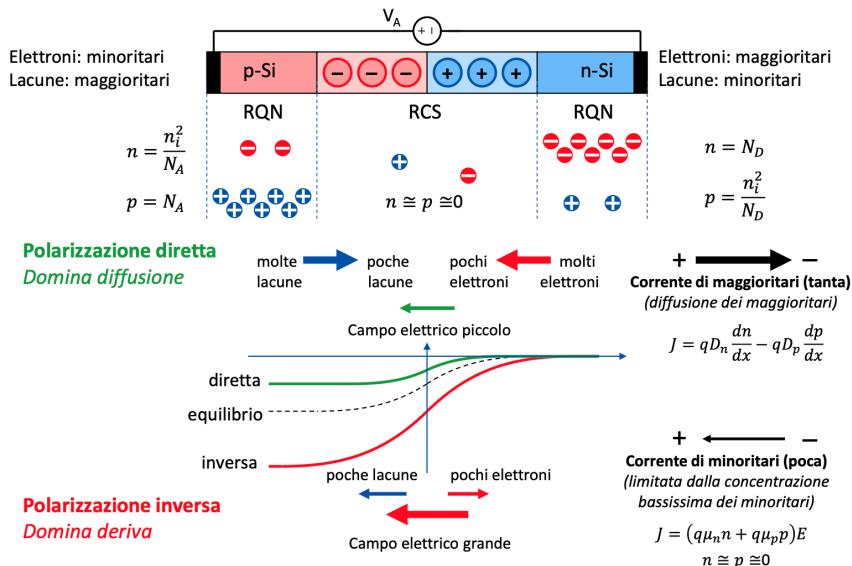
Giunzione in polarizzazione inversa

In polarizzazione inversa la tensione applicata $V_A < 0$ aumenta la differenza di potenziale tra le due regioni quasi neutre, ottenendo: $V_p - V_n = -V_0 - |V_A|$. Di conseguenza il campo elettrico nella regione di svuotamento aumenta in modulo e l'ampiezza della regione di svuotamento si allarga:

$$W(V_0 + |V_A|) = \sqrt{\frac{2\epsilon(V_0 + |V_A|)}{q} \left(\frac{1}{N_A} + \frac{1}{N_D} \right)} > W(V_0)$$

Siccome il campo elettrico e il potenziale aumentano, la corrente di deriva prevale sulla corrente di diffusione. Si ha in questo modo un flusso di elettroni dalla regione p alla regione n e di lacune dalla regione n alla regione p. Siccome la regione p è povera di elettroni e la regione n è povera di lacune, questo flusso è detto flusso dei portatori minoritari. Si induce in questo modo una **debole corrente di deriva dei minoritari** che attraversa la giunzione pn dalla regione n alla regione p.

$$J = (\mu_n n + \mu_p p) qE \quad \text{con } n \approx p \approx 0$$

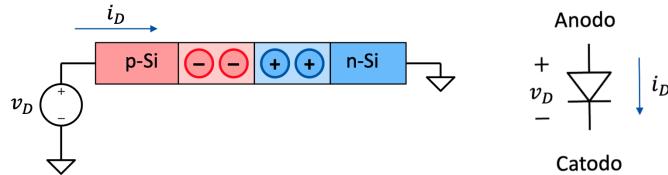


3.3 Giunzione pn polarizzata vista come diodo

Relazione tensione-corrente in un diodo

Si assegnano dei riferimenti ai due terminali della giunzione pn in modo da renderla schematizzabile e utilizzabile come diodo in un circuito elettrico:

- **anodo**: terminale positivo, collegato alla regione p, dove si assorbono gli elettroni
- **catodo**: terminale negativo, collegato alla regione n, dove si immettono gli elettroni

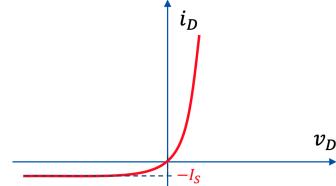


Definiti i riferimenti di tensione e corrente in un diodo (o giunzione pn polarizzata), si può definire la relazione tensione-corrente che lega la tensione applicata V_A alla corrente I che attraversa il diodo:

$$i_D = I_S \left(e^{\frac{v_D}{\eta V_T}} - 1 \right)$$

- i_D e v_D sono la corrente e la tensione nel diodo, con riferimento positivo dall'anodo al catodo;
- I_S è la corrente di saturazione inversa, ovvero la debole corrente che attraversa il diodo quando è collegato in polarizzazione inversa (tipicamente dell'ordine di qualche nA);
- η è il coefficiente di idealità del diodo, che dipende dal materiale e dal processo di fabbricazione (tipicamente compreso tra 1 e 2);
- $V_T = k_B T / q$ è il potenziale termico (del valore di circa 25mV a temperatura ambiente).

Analizzando la curva caratteristica del diodo si osserva che in polarizzazione diretta ($v_D > 0$) la corrente cresce esponenzialmente con la tensione applicata, mentre in polarizzazione inversa ($v_D < 0$) la corrente si stabilizza ad un valore negativo pari a $-I_S$.



Capacità della giunzione in polarizzazione inversa

Analizzando la carica elettrica (per unità di superficie) presente nella regione di carica spaziale di una giunzione pn in polarizzazione inversa si osserva che:

$$Q_n = qN_D x_n = qN_D \frac{N_A}{N_A + N_D} W = \sqrt{2q\varepsilon(V_0 - V_A) \frac{N_A N_D}{N_A + N_D}} \quad Q_n = -Q_p$$

$$Q_p = qN_A x_p = qN_A \frac{N_D}{N_A + N_D} W = \sqrt{2q\varepsilon(V_0 - V_A) \frac{N_A N_D}{N_A + N_D}}$$

Si osserva quindi che la carica dipende dalla tensione applicata V_A . La giunzione pn in polarizzazione inversa equivale ad un condensatore con capacità per unità di area non lineare data da:

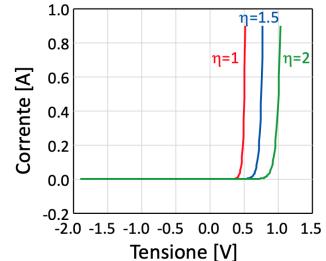
$$C_j = \frac{dQ}{dV_A} = \sqrt{\frac{q\varepsilon}{2(V_0 - V_A)} \frac{N_A N_D}{N_A + N_D}} = \frac{\varepsilon}{W}$$

NOTA: per ottenere la reale capacità del diodo bisogna moltiplicare la capacità per unità di area C_j per la sezione della giunzione Σ_j :

$$C = C_j \cdot \Sigma_j = \frac{\varepsilon}{W} \Sigma_j$$

Coefficiente di idealità

Facendo variare il coefficiente di idealità η tra 1 e 2 si osserva che avviene una traslazione orizzontale della curva caratteristica del diodo. Minore è il valore di η , più la curva sale rapidamente in polarizzazione diretta. In genere si utilizza $\eta = 1$ per correnti basse e $\eta = 2$ per correnti elevate.

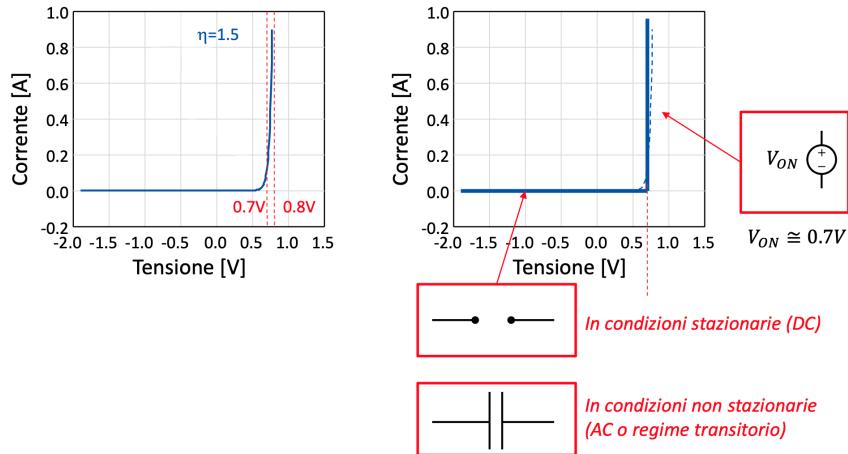


Modello semplificato del diodo

Si osserva che la curva caratteristica del diodo può essere approssimata con un modello semplificato definito in funzione della tensione applicata v_D e di conseguenza della polarizzazione del diodo:

- per $v_D < V_{ON} \rightarrow i_D = 0$ il diodo è in interdizione e si comporta come un circuito aperto in condizioni stazionarie oppure come condensatore non lineare in condizioni non stazionarie;
- per $v_D = V_{ON} \rightarrow i_D > 0$ il diodo è in conduzione e si comporta come un generatore ideale di tensione con tensione V_{ON} .

La tensione V_{ON} è detta tensione di soglia del diodo e divide le due regioni di funzionamento. Tipicamente per un diodo al silicio si assume $V_{ON} = 0.7V$, mentre per un diodo al germanio si assume $V_{ON} = 0.3V$.



3.4 Applicazioni speciali dei diodi

Applicazioni generali

I diodi sono componenti fondamentali in molti circuiti elettronici e trovano applicazione in diversi ambiti:

- **raddrizzatori**: i diodi vengono utilizzati nei circuiti raddrizzatori per convertire la corrente alternata (AC) in corrente continua (DC), permettendo il funzionamento di dispositivi elettronici alimentati a corrente continua;
- **protezione da inversioni di polarità**: i diodi proteggono i circuiti elettronici da danni causati da inversioni accidentali di polarità della tensione di alimentazione;
- **limitatori di tensione**: i diodi limitano la tensione in un circuito, proteggendo i componenti sensibili da sovrattensioni;
- **LED (Light Emitting Diode)**: i diodi LED emettono luce quando attraversati da corrente elettrica, trovando applicazione in display, indicatori luminosi e illuminazione;
- **fotorilevatore**: i diodi a semiconduttore possono essere utilizzati come sensori di luce, convertendo l'energia luminosa in corrente elettrica.

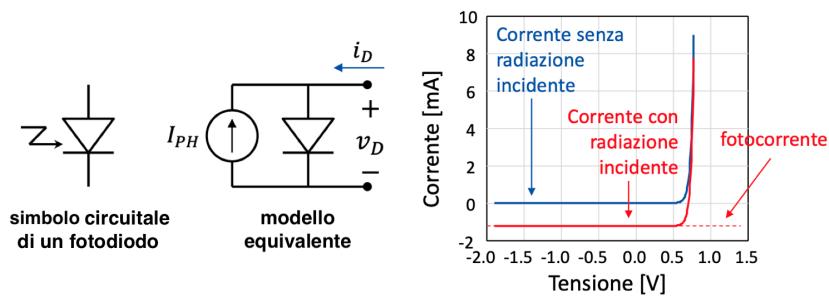
Fotodiodo

Il fotodiodo è una giunzione pn collegata in polarizzazione inversa con l'area di svuotamento esposta alla luce. Quando la luce colpisce la regione di svuotamento, eccita gli elettroni che si liberano dai legami covalenti, generando coppie elettrone-lacuna (fotogenerazione). I due portatori vengono separati dal campo elettrico presente nella regione, generando una corrente detta photocorrente. La photocorrente è proporzionale all'intensità della luce incidente. Il fotodiodo viene utilizzato in applicazioni come sensori di luce e telecomunicazioni ottiche.

$$i_D = I_S \left(e^{\frac{v_D}{nV_T}} - 1 \right) - I_{PH} \quad I_{PH} = R \cdot P_0$$

I_{PH} : photocorrente (A)
 P_0 : potenza ottica incidente (W)
 R : responsività (A/W)

La curva caratteristica risulta spostata verso il basso di un valore pari alla photocorrente I_{PH} . È possibile schematizzare un fotodiodo come un diodo ideale in parallelo ad una sorgente di corrente pari a I_{PH} .



LED (Light Emitting Diode) o diodi a emissione luminosa

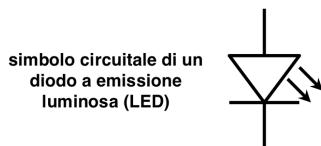
I LED sono giunzioni pn in polarizzazione diretta con l'area di svuotamento “scoperta”. Quando una corrente attraversa il LED, le lacune dalla regione p si ricombinano con gli elettroni dalla regione n nella regione di svuotamento, rilasciando energia sotto forma di fotoni (emissione di luce).

La lunghezza d'onda della luce emessa dipende dalla differenza di energia tra la banda di conduzione e la banda di valenza (energy gap) propria di ogni semiconduttore. Si utilizzano, infatti, semiconduttori diversi per ottenere colori diversi.

Un parametro importante è la tensione di accensione V_{ON} dei LED, ovvero la tensione a cui il LED inizia a emettere luce. In genere è superiore alla tensione di soglia in quanto non basta fornire energia per permettere il passaggio della corrente, ma è necessario fornire energia sufficiente per permettere l'emissione dei fotoni.

$V_{ON} > \frac{hc}{\lambda q}$	h : costante di Planck (6.626×10^{-34} Js)	$\lambda = 620$ nm (rosso) $V_{ON} \approx 2.0 - 2.2$ V
	c : velocità della luce (3.0×10^8 m/s)	$\lambda = 520$ nm (giallo) $V_{ON} \approx 2.1 - 2.2$ V
	λ : lunghezza d'onda della luce emessa	$\lambda = 510$ nm (verde) $V_{ON} \approx 2.5 - 3.3$ V
	q : carica dell'elettrone (1.6×10^{-19} C)	$\lambda = 470$ nm (blu) $V_{ON} \approx 3.2 - 3.3$ V

Non esiste nessun semiconduttore che emetta luce bianca: per creare un “LED bianco” si utilizza un LED blu con un rivestimento di fosforo (giallo) che converte parte della luce blu in luce gialla. Dalla combinazione delle due luci si ottiene la luce bianca percepita dall'occhio umano. In base alla quantità di fosforo utilizzata si possono ottenere diverse tonalità di bianco (caldo, neutro, freddo).



4 Condensatore MOS o CMOS

4.1 Struttura e funzionamento

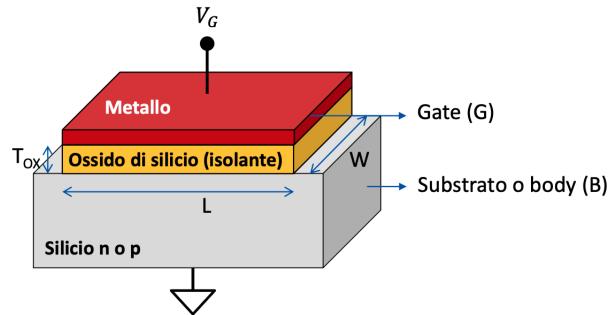
Struttura base

Un condensatore MOS (Metal-Oxide-Semiconductor) è costituito da tre strati principali:

- un metallo (Metal) che funge da elettrodo superiore detto **gate** (G), generalmente in polisilicio;
- un ossido (Oxide) che funge da dielettrico o isolante, di solito in biossido di silicio, SiO_2 ;
- un semiconduttore (Semiconductor) che funge da elettrodo inferiore detto **substrato o body** (B), generalmente in silicio drogato di tipo p o n.

Funzionamento e proprietà

- Si identificano le dimensioni del dielettrico con L lunghezza, W larghezza e T_{ox} spessore.
- Si assume di collegare il substrato a massa (0 V) e applicare una tensione variabile al gate V_G .
- La capacità del condensatore mos è data da: $C_{ox} = \varepsilon \frac{W \cdot L}{T_{ox}}$



4.2 Condensatore mos con substrato di tipo p

Tensione di gate negativa ($V_G < 0$)

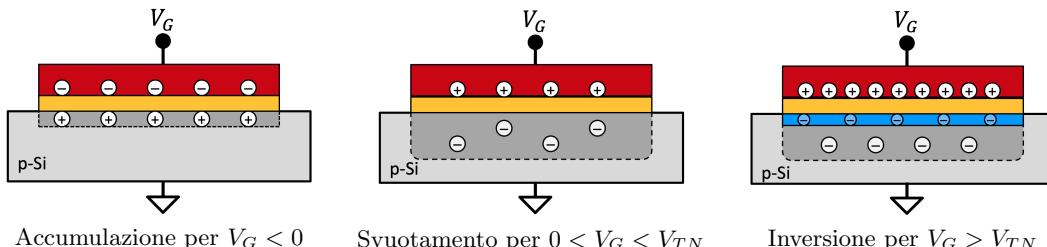
Se $V_G < 0$, il gate si carica negativamente, attirando le cariche positive (lacune) verso la superficie del semiconduttore adiacente all'ossido, creando una **regione di accumulazione** di lacune.

Tensione di gate inferiore alla tensione di soglia ($0 < V_G < V_{TN}$)

Se $0 < V_G < V_{TN}$, il gate si carica positivamente, creando una **regione di svuotamento** di lacune vicino alla superficie del semiconduttore, lasciando dietro di sé ioni negativi fissi (atomi droganti). Si forma così una zona di carica spaziale negativa, priva di portatori mobili.

Tensione di gate superiore alla tensione di soglia ($V_G > V_{TN}$)

Se $V_G > V_{TN}$, il gate si carica ancora più positivamente, attirando elettroni verso la superficie del semiconduttore adiacente all'ossido. Si crea una **regione di inversione** dove la concentrazione di elettroni supera quella delle lacune. Si forma così un canale conduttivo di tipo n. La dimensione della regione di svuotamento rimane quasi costante, dopo aver raggiunto il massimo per $V_G = V_{TN}$, mentre la concentrazione di elettroni nella regione di inversione aumenta con V_G .



4.3 Analisi del p-cmos in condizioni di svuotamento/inversione

Come per la giunzione pn, le grandezze analizzate sono per unità di area del condensatore mos, per cui per ottenere i valori effettivi è necessario moltiplicarle per l'area $\Sigma_C = W \cdot L$ del condensatore mos.

Densità di carica

Analizzando la densità di carica $\rho(x)$, il campo elettrico $E(x)$ si ottengono le seguenti relazioni:

$$\rho(x) = \begin{cases} -qN_A & \text{per } -x_D < x < 0 \\ 0 & \text{altrimenti} \end{cases}$$

Campo elettrico

Dalla densità di carica si ricava il campo elettrico $E(x)$ nella regione di svuotamento e nell'ossido, si noti che c'è una discontinuità del campo elettrico all'interfaccia semiconduttore-ossido dovuta alla differenza di permittività tra i due materiali, inoltre il campo elettrico nell'ossido è costante:

$$E(x) = \begin{cases} -qN_A(x + x_D)/\varepsilon_S & \text{per } -x_D < x < 0 \\ -qN_A x_D / \varepsilon_{OX} & \text{per } 0 < x < t_{OX} \\ 0 & \text{altrimenti} \end{cases} \quad \begin{aligned} E(0^-) &= -\frac{qN_A x_D}{\varepsilon_S} \\ E(0^+) &= -\frac{qN_A x_D}{\varepsilon_{OX}} \\ E_{OX} &= E(0^+) = E(0^-) \frac{\varepsilon_S}{\varepsilon_{OX}} \end{aligned}$$

Potenziale elettrico

Si ottiene il potenziale nel substrato V_B (potenziale di riferimento), il potenziale all'interfaccia (nella giunzione tra semiconduttore e ossido) $V(0)$ e il potenziale al gate V_G :

$$V_B = V(-x_D) = 0 \quad V(0) = \frac{qN_A}{2\varepsilon_S} x_D^2 \quad V_G = V(t_{OX}) = \frac{qN_A}{2\varepsilon_S} x_D^2 + \frac{qN_A x_D}{\varepsilon_{OX}} t_{OX}$$

Concentrazioni dei portatori

Si ricavano le concentrazioni dei portatori p_1 e n_1 nella regione neutra, lontano dall'interfaccia, e le concentrazioni all'interfaccia p_2 , n_2 . All'interfaccia le concentrazioni variano esponenzialmente con $V(0)$:

$$\begin{aligned} p_1 &= N_A & p_2 &= p_1 e^{-\frac{V(0)}{V_T}} & \frac{n_2}{n_1} &= \frac{p_1}{p_2} = e^{\frac{v_2 - v_1}{V_T}} & \frac{v_2 - v_1}{V_T} &= \ln \frac{n_2}{n_1} = \ln \frac{p_1}{p_2} \\ n_1 &= n_i^2 / N_A & n_2 &= n_1 e^{\frac{V(0)}{V_T}} \end{aligned}$$

Tensione di soglia

Si definisce la **tensione di soglia** V_{TN} come differenza di potenziale tra gate e substrato $V_G - V_B$ (pari a V_G) per cui la concentrazione di elettroni all'interfaccia è uguale al numero di lacune nella regione neutra, ovvero quando c'è inversione totale con $n_2 = N_A$:

$$V(0) = \frac{qN_A}{2\varepsilon_S} x_D^2 = 2V_T \ln \frac{N_A}{n_i}, \quad x_D = \sqrt{\frac{4\varepsilon_S V_T}{qN_A} \ln \frac{N_A}{n_i}} \rightarrow V_{TN} = 2V_T \ln \frac{N_A}{n_i} + \frac{t_{OX}}{\varepsilon_{OX}} \sqrt{4\varepsilon_S qN_A V_T \ln \frac{N_A}{n_i}}$$

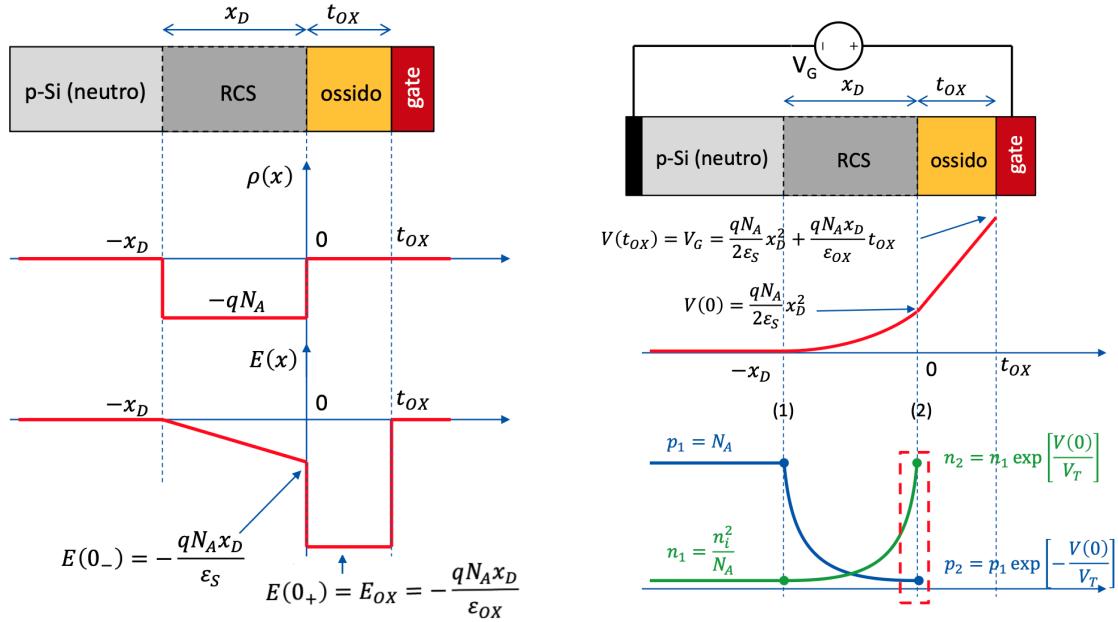
La tensione di soglia dipende, quindi, dallo spessore dell'ossido t_{OX} , dalla concentrazione di drogaggio del substrato N_A e dai materiali usati (tramite ε_{OX} e ε_S).

Carica elettrica e capacità

La carica elettrica per unità di area immagazzinata nel condensatore mos è data dalla somma degli ioni fissi nella regione di svuotamento e degli elettroni nella regione di inversione:

$$Q_{TOT} = Q_{RCS} + Q_n = Q \cdot V_G \quad Q_{RCS} = C \cdot V_{TN} \quad Q_n = C \cdot (V_G - V_{TN})$$

Rappresentazione grafica del comportamento del p-cmos

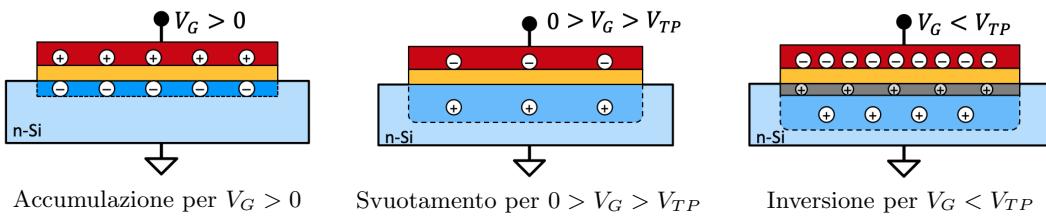


4.4 cmos con substrato di tipo n e differenze rispetto al p-cmos

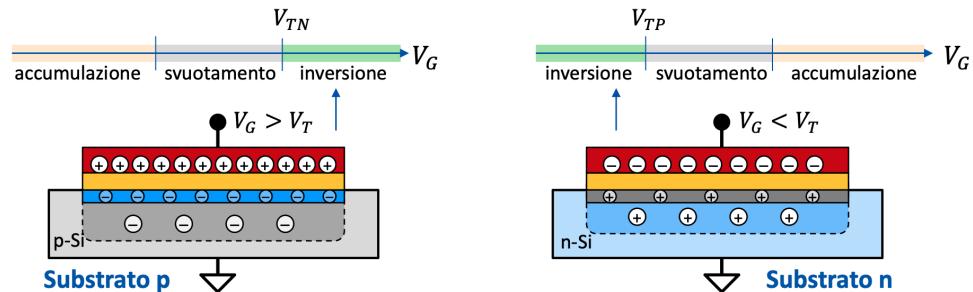
Comportamento del cmos con substrato di tipo p

Il funzionamento è analogo a quello del cmos con substrato di tipo p, ma con le polarità invertite:

- per $V_G > 0$, si crea una regione di accumulazione di elettroni.
- per $0 > V_G > V_{TP}$, si crea una regione di svuotamento di elettroni.
- per $V_G < V_{TP}$, si crea una regione di inversione con un canale conduttivo di tipo p.



Rappresentazione aree di lavoro dei cmos di tipo p e n

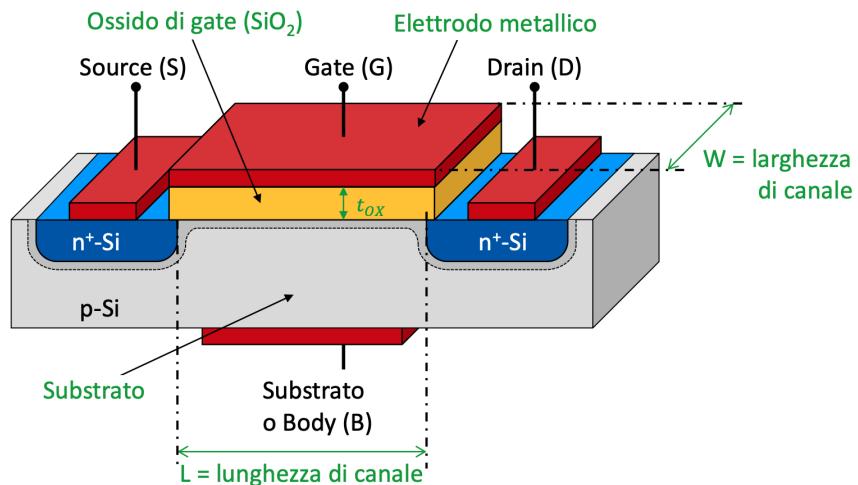


5 Transistor MOSFET

5.1 Struttura generale e classificazione dei mosfet

Introduzione

Un transistor MOSFET (Metal-Oxide-Semiconductor Field-Effect Transistor) è un dispositivo a quattro terminali che sfrutta un condensatore mos per controllare il flusso di corrente tra due terminali detti **source** (S) e **drain** (D) tramite un potenziale applicato ad un terzo terminale detto **gate** (G). Il quarto terminale è il substrato o **body** (B) e viene generalmente collegato al source o ad un potenziale di riferimento (massa o V_{DD}). Di seguito una rappresentazione schematica di un n-mosfet:



Struttura fisica

A livello fisico, un mosfet è costituito da un condensatore mos con gate e substrato affiancato da due regioni pesantemente drogata di tipo opposto al substrato, dette source e drain, che fungono da terminali di ingresso e uscita. Il dielettrico del condensatore mos è generalmente in diossido di silicio (SiO_2).

Ad ogni terminale è associato un potenziale elettrico e per ogni coppia di terminali si definisce la tensione e la corrente tra i due nodi:

- | | |
|------------------------------------|--|
| - V_G : potenziale del gate | - $V_{XY} = V_X - V_Y$: tensione tra i nodi X e Y |
| - V_S : potenziale del source | es. $V_{GS} = V_G - V_S$ tensione tra gate e source |
| - V_D : potenziale del drain | - I_{XY} : corrente che entra nel nodo X e esce dal nodo Y |
| - V_B : potenziale del substrato | es. I_{DS} corrente che entra nel drain e esce dal source |

Si definiscono inoltre le dimensioni fisiche del mosfet:

- L : lunghezza del canale tra source e drain
- W : larghezza del canale tra source e drain
- t_{OX} : spessore dell'ossido isolante tra gate e substrato

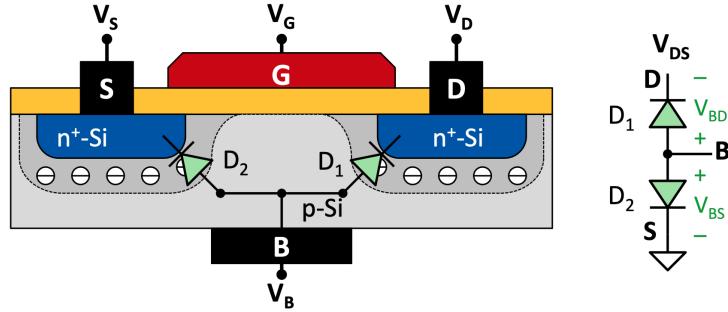
Classificazione

In base al tipo di canale (e di conseguenza in base al tipo del substrato), i mosfet si classificano in:

- **n-mosfet o mosfet a canale n:**
substrato di tipo p, source e drain di tipo n⁺, canale di tipo n con gli elettroni come portatori principali, il substrato p collegato al potenziale minore del circuito (massa o al source) e il source ha potenziale minore del drain
- **p-mosfet o mosfet a canale p:**
substrato di tipo n, source e drain di tipo p⁺, canale di tipo p con le lacune come portatori principali, il substrato n collegato al potenziale maggiore del circuito (V_{DD} o al source) e il source ha potenziale maggiore del drain

5.2 Struttura di un n-mosfet e vincoli sui potenziali

Si assume per convenzione che il terminale di source ha potenziale minore di quello di drain: $V_S < V_D$ e, di conseguenza, che la corrente scorra dal drain al source: $I_{DS} > 0$.



Condizioni all'equilibrio (nessuna tensione applicata)

In assenza di tensioni, tutti i potenziali sono nulli e non c'è corrente tra i terminali. In particolare si hanno due giunzioni pn in equilibrio tra il substrato p e le regioni n⁺ del source e del drain. Siccome le regioni n⁺ sono pesantemente drogate, la regione di svuotamento si estende quasi totalmente nel substrato p.

Vincoli di polarizzazione dei diodi e potenziale di substrato

I due diodi con catodi collegati ai nodi source e drain e con anodo in comune nel substrato, devono rimanere in interdizione per il corretto funzionamento del mosfet, si ottengono le seguenti condizioni:

- $V_{BS} \leq 0 \rightarrow V_S \geq V_B$ (diodo source-substrato in interdizione)
- $V_{BD} \leq 0 \rightarrow V_D \geq V_B$ (diodo drain-substrato in interdizione)

Si ottiene che il substrato V_B deve essere il nodo a potenziale più basso $V_D \geq V_S \geq V_B$. Si solito si collega il substrato al potenziale minore dell'intero circuito (massa) $V_B = 0$ oppure al source $V_B = V_S$.

NOTA: Non è possibile collegare il substrato al drain perché si violerebbe la condizione di interdizione del diodo source-substrato in quanto $V_S < V_D$ e quindi $V_{BS} = V_B - V_S = V_D - V_S > 0$.

Potenziale e tensioni di gate

Il potenziale di gate V_G controlla la tensione tra le armature del condensatore mos che si forma tra il gate e il substrato e di conseguenza identifica l'area di lavoro del p-mosfet. La tensione tra le due armature è variabile lungo la lunghezza del canale ed è compresa tra le tensioni di gate-source e gate-drain ai margini del canale:

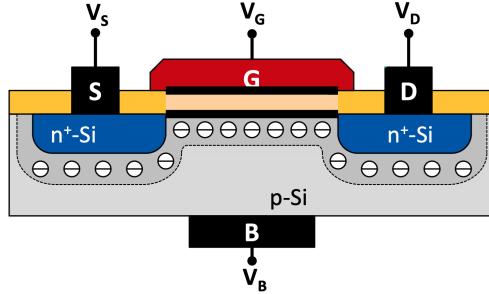
- in prossimità del source $V_{C,source} = V_{GS} = V_G - V_S$
- in prossimità del drain $V_{C,drain} = V_{GD} = V_G - V_D$

NOTA: siccome $V_D \geq V_S$, si ha $V_{GS} \geq V_{GD}$

5.3 Aree di lavoro di un n-mosfet

Transistor n-mosfet spento o in interdizione per $V_{GS} < V_{TN}$

La tensione tra le armature del condensatore mos è inferiore alla tensione di soglia V_{TN} sia in prossimità del source ($V_{GS} < V_{TN}$) che in prossimità del drain ($V_{GD} < V_{GS} < V_{TN}$). Il condensatore, quindi, è in regime di svuotamento ed è presente nel substrato, in prossimità dell'ossido, un'area di svuotamento (senza portatori di carica) che separa source e drain. Non essendoci cariche libere per condurre corrente tra drain e source, si ha corrente nulla $I_{DS} = 0$ e il transistor è spento o in interdizione.



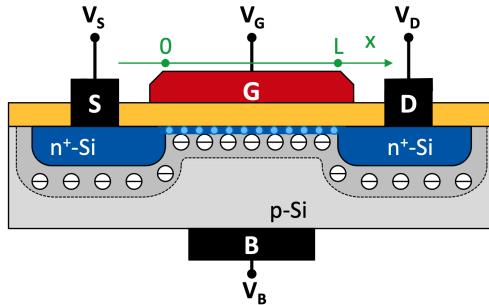
$$V_{GS} < V_{TN} \quad \text{source in interdizione}$$

$$V_{GD} < V_{GS} < V_{TN} \quad \text{drain in interdizione}$$

$$I_{DS} = 0$$

Transistor n-mosfet in conduzione lineare (o triodo) per $V_{DS} < V_{GS} - V_{TN}$

La tensione delle armature del condensatore mos è superiore alla tensione di soglia V_{TN} sia in prossimità del drain ($V_{GD} > V_{TN}$) che in prossimità del source ($V_{GS} > V_{GD} > V_{TN}$). Il condensatore si trova in regime di inversione e si forma un canale conduttivo di tipo n (di elettroni) che congiunge source e drain. Gli elettroni che costituiscono il canale si muovono liberamente dal source al drain, permettendo il passaggio di corrente $I_{DS} > 0$ dal drain al source. Il transistor si comporta come una resistenza (dipendente da V_{GS}) tra drain e source, lineare per V_{DS} piccoli o parabolica per V_{DS} grandi.



$$V_{GD} > V_{TN} \quad \text{drain in conduzione}$$

$$V_{GS} = V_{GD} + V_{DS}$$

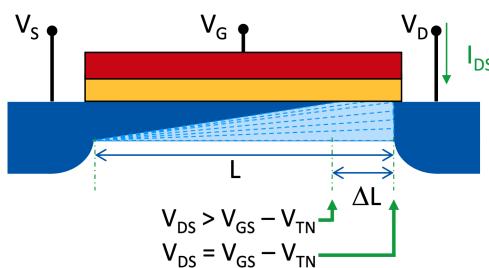
$$V_{GS} > V_{TN} + V_{DS} \quad \text{source in conduzione}$$

$$I_{DS} = k_n V_{DS} \left(V_{GS} - V_{TN} - \frac{V_{DS}}{2} \right)$$

$$k_n = k'_n \cdot Z_n \quad k'_n = \mu_n \varepsilon / T_{OX} \quad Z_n = W_n / L_n$$

Transistor n-mosfet in saturazione per pinchoff per $V_{DS} > V_{GS} - V_{TN}$

La tensione delle armature del condensatore mos è superiore alla tensione di soglia V_{TN} in prossimità del source ($V_{GS} > V_{TN}$), ma inferiore alla tensione di soglia in prossimità del drain ($V_{GD} < V_{TN}$). Il condensatore si trova in regime misto: inversione in prossimità del source e svuotamento in prossimità del drain e si forma un canale conduttivo di tipo n che congiunge solo parzialmente source e drain. Nonostante ciò i portatori riescono a fluire lo stesso nella RCS spinti dal campo elettrico tra source e drain. In questo modo si ha ugualmente il passaggio di una corrente $I_{DS} \neq 0$ costante rispetto a V_{DS} (a meno della modulazione di lunghezza di canale).



$$V_{GS} > V_{TN} \quad \text{source in conduzione}$$

$$V_{GD} < V_{TN} \quad \text{drain in interdizione}$$

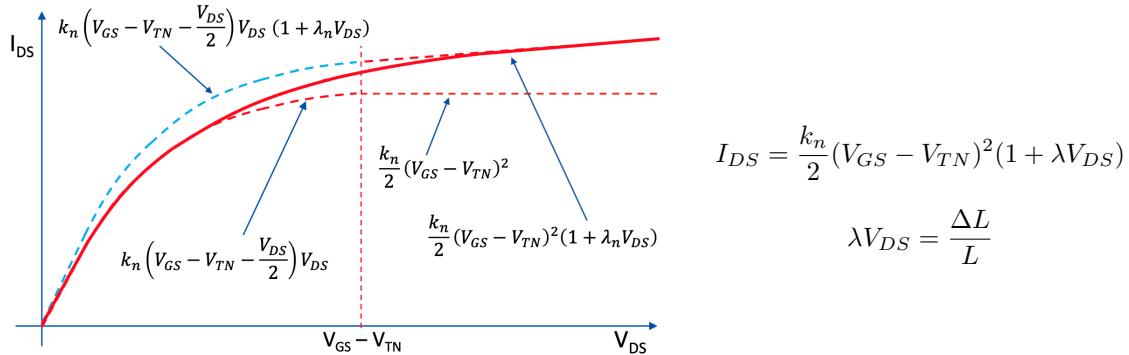
$$\begin{aligned} V_{GD} < V_{TN} \rightarrow & V_{GS} - V_{DS} < V_{TN} \\ \rightarrow & V_{DS} > V_{GS} - V_{TN} \end{aligned}$$

$$I_{DS} = \frac{k_n}{2} (V_{GS} - V_{TN})^2$$

$$k_n = k'_n \cdot Z_n \quad k'_n = \mu_n \varepsilon / T_{OX} \quad Z_n = W_n / L_n$$

Modulazione di lunghezza di canale con n-mosfet in saturazione per pinchoff

Quando il mosfet entra in saturazione per pinchoff, la regione di strozzamento si sposta verso il source, riducendo la lunghezza L del canale conduttivo. Questo fenomeno, chiamato modulazione di lunghezza di canale, provoca un lieve aumento della corrente I_{DS} con l'aumentare di V_{DS} anche in saturazione. Si può modellare questo effetto aggiungendo alla corrente in saturazione un termine correttivo che dipende dal coefficiente di modulazione di lunghezza λ determinato dalle caratteristiche fisiche del mosfet:



Analizzando la continuità della corrente tra la regione lineare e la regione di saturazione con modulazione di lunghezza di canale si osserva che matematicamente c'è una discontinuità che in natura non esiste. Per ovviare a questo problema si utilizza la correzione di lunghezza di canale anche nella regione lineare (tratteggio blu nella figura superiore):

$$I_{DS,\text{lin}} = k_n V_{DS} \left(V_{GS} - V_{TN} - \frac{V_{DS}}{2} \right) (1 + \lambda V_{DS}) \quad \text{per } V_{DS} \ll V_{GS} - V_{TN}$$

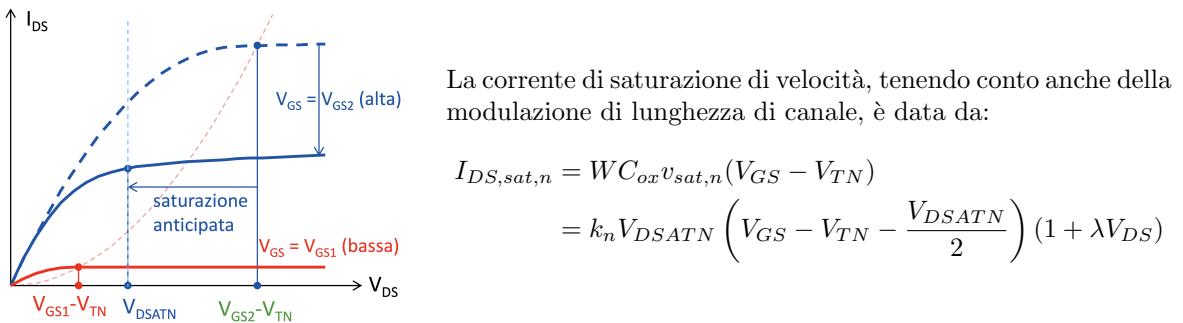
$$I_{DS,\text{sat}} = \frac{k_n}{2} (V_{GS} - V_{TN})^2 (1 + \lambda V_{DS}) \quad \text{per } V_{DS} \gg V_{GS} - V_{TN}$$

Transistor n-mosfet in saturazione di velocità per $V_{DS} > V_{DSATN}$

I portatori di carica (in questo caso elettroni) si muovono nel canale spinti dal campo elettrico $E = V_{DS}/L$ con velocità $v_n = \mu_n E = \mu_n V_{DS}/L$. La velocità è, quindi, proporzionale a V_{DS} . Tale velocità raggiunge un valore massimo costante detto velocità di saturazione $v_{sat,n} = 8 \cdot 10^6 \text{ cm/s}$ per un certo campo elettrico critico $E_{crit,n}$ e una certa tensione critica $V_{DS,sat,n}$:

$$E_{\text{crit},n} = \frac{v_{sat,n}}{\mu_n} = 1.3 \frac{\text{V}}{\mu\text{m}} \quad V_{DSATN} = E_{\text{crit},n} \cdot L = \frac{v_{sat,n}}{\mu_n} L$$

Per $V_{DS} > V_{DSATN}$, si osserva che la velocità degli elettroni e di conseguenza anche la corrente I_{DS} rimangono costanti. Si ha un fenomeno di saturazione anticipata, detto saturazione di velocità, quando ci si aspetterebbe che il mosfet lavori in regime di conduzione lineare.



La corrente di saturazione di velocità, tenendo conto anche della modulazione di lunghezza di canale, è data da:

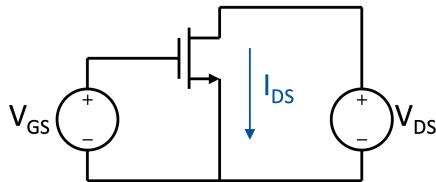
$$I_{DS,sat,n} = WC_{ox} v_{sat,n} (V_{GS} - V_{TN})$$

$$= k_n V_{DSATN} \left(V_{GS} - V_{TN} - \frac{V_{DSATN}}{2} \right) (1 + \lambda V_{DS})$$

5.4 Curve caratteristiche di corrente-tensione di un n-mosfet

La corrente I_{DS} dipende da due tensioni indipendenti V_{GS} e V_{DS} . Si identificano due curve caratteristiche:

- **caratteristica di uscita** con I_{DS} in funzione di V_{DS} per valori costanti di V_{GS}
- **caratteristica di trasferimento o transcaratteristica** con I_{DS} in funzione di V_{GS} per valori costanti di V_{DS}

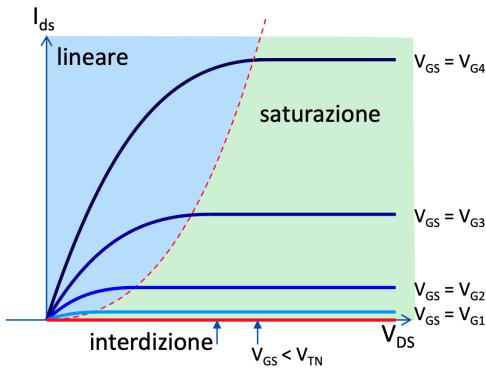


Per analizzare le curve caratteristiche si collega il mosfet ad un circuito di test con due generatori di tensione V_{GS} e V_{DS} e si misura la corrente I_{DS} che scorre tra drain e source. Per la caratteristica di uscita si mantiene V_{GS} costante e si varia V_{DS} , mentre per la transcaratteristica si mantiene V_{DS} costante e si varia V_{GS} .

Caratteristica di uscita $I_{DS} - V_{DS}$

Le tre aree di funzionamento del n-mosfet si riflettono nella caratteristica di uscita $I_{DS} - V_{DS}$:

- **interdizione** per $V_{GS} < V_{TN}$: linea orizzontale coincidente con l'asse delle ascisse
- **conduzione lineare** per $V_{DS} < V_{GS} - V_{TN}$: a sinistra della linea di saturazione
- **saturazione per pinchoff** per $V_{DS} > V_{GS} - V_{TN}$: a destra della linea di saturazione
- **curva di saturazione**: separa la regione di funzionamento lineare e quella di saturazione, è costituita dai punti (V_{DS}, I_{DS}) che soddisfano l'equazione:



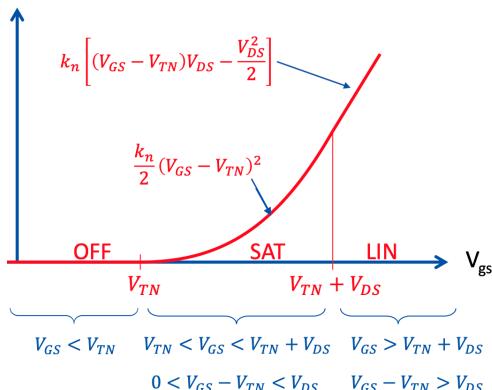
$$I_{DS,sat} = \frac{k_n}{2} V_{DS}^2 \quad \text{con } V_{DS} = V_{GS} - V_{TN}$$

- aumentando V_{DS} , il transistor passa da regime lineare a regime di saturazione per pinchoff
- aumentando V_{GS} ($V_{G4} > V_{G3} > V_{G2} > V_{G1} > V_{TN}$) la corrente I_{DS} aumenta

Transcaratteristica $I_{DS} - V_{GS}$

Le tre aree di funzionamento del n-mosfet si riflettono nella caratteristica di uscita $I_{DS} - V_{GS}$:

- **interdizione** per $V_{GS} < V_{TN}$: linea orizzontale coincidente con l'asse delle ascisse
- **saturazione per pinchoff** per $V_{TN} < V_{GS} < V_{DS} + V_{TN}$ crescita quadratica
- **conduzione lineare** per $V_{DS} + V_{TN} < V_{GS}$ crescita lineare



- aumentando V_{GS} , il transistor passa da regime di interdizione a regime di saturazione per pinchoff e successivamente a regime lineare
- aumentando V_{DS} il confine tra regime di saturazione e regime lineare si sposta verso destra, inoltre aumenta la pendenza della retta in regime lineare

5.5 Modello a canale corto di un n-mosfet

Equazione generale per I_{DS}

Il modello a canale corto tiene conto di tutti i fenomeni fisici che avvengono in un mosfet reale, tra cui la modulazione di lunghezza di canale e la saturazione di velocità. Tutte le aree di funzionamento del n-mosfet si possono descrivere con un'unica equazione per la corrente I_{DS} :

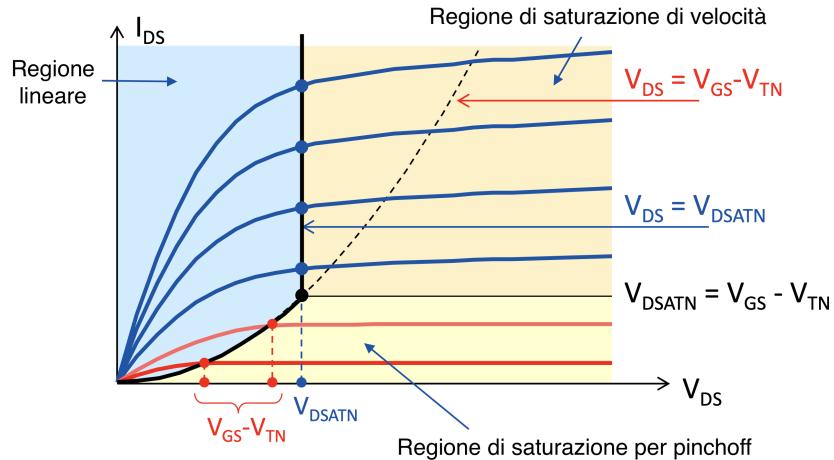
- per $V_{GS} < V_{TN}$ mosfet in interdizione e $I_{DS} = 0$
- per $V_{GS} > V_{TN}$ mosfet in conduzione con I_{DS} che dipende da V_{MIN} :

$$I_{DS} = k'_n Z_n V_{MIN} \left(V_{GS} - V_{TN} - \frac{V_{MIN}}{2} \right) (1 + \lambda_n V_{DS}) \quad k'_n = \frac{\mu_n \epsilon}{T_{OX}} \quad Z_n = \frac{W_n}{L_n}$$

$$V_{MIN} = \min \left\{ \begin{array}{ll} V_{DS} & \text{regime lineare} \\ V_{GS} - V_{TN} & \text{saturazione per pinchoff} \\ V_{DSATN} & \text{saturazione per velocità} \end{array} \right\}$$

Caratteristica di uscita del modello a canale corto

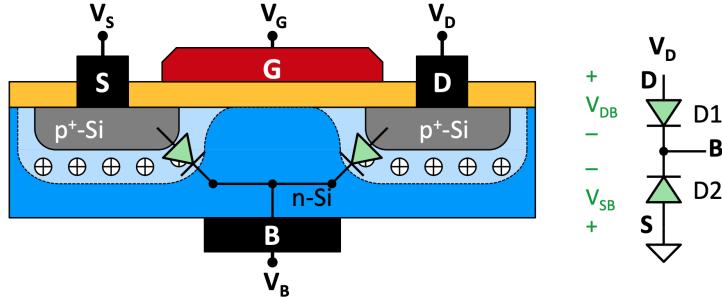
Includendo anche gli effetti della saturazione di velocità, la caratteristica di uscita complessiva del modello a canale corto del n-mosfet risulta come segue:



- la regione di conduzione lineare e la regione di saturazione per pinchoff sono separate dalla curva di saturazione per pinchoff data dalla parabola $I_{DS} = k_n/2 \cdot V_{DS}^2$ con $V_{DS} = V_{GS} - V_{TN}$
- la regione di conduzione lineare e la regione di saturazione per velocità sono separate dalla linea verticale $V_{DS} = V_{DSATN}$
- la regione di saturazione per pinchoff e la regione di saturazione per velocità sono separate dalla linea orizzontale $I_{DS} = k_n/2 \cdot V_{DS}^2$ con $V_{DS} = V_{DSATN}$

5.6 Struttura di un p-mosfet e vincoli sui potenziali

Il funzionamento di un mosfet a canale p (p-mosfet) con substrato di tipo n è analogo a quello del n-mosfet, ma con le polarità invertite. Tutte le tensioni, infatti, sono negative. Si assume per convenzione che il terminale di source ha potenziale maggiore di quello di drain ($V_S > V_D$) e, di conseguenza, che la corrente scorre dal source al drain rimanendo sempre positiva ($I_{DS} > 0$).



Condizioni all'equilibrio

In analogia al n-mosfet, all'equilibrio le due giunzioni pn tra substrato n e le regioni p^+ del source e del drain sono in equilibrio e la regione di svuotamento si estende quasi totalmente nel substrato n.

Vincoli di polarizzazione dei diodi e potenziale di substrato

I due diodi con anodi collegati ai nodi source e drain e con catodo in comune nel substrato, devono rimanere in interdizione per il corretto funzionamento del mosfet, si ottengono le seguenti condizioni:

- $V_{BS} \geq 0 \rightarrow V_S \leq V_B$ (diodo source-substrato in interdizione)
- $V_{BD} \geq 0 \rightarrow V_D \leq V_B$ (diodo drain-substrato in interdizione)

Si ottiene che il substrato V_B deve essere il nodo a potenziale più alto $V_D \leq V_S \leq V_B$. Di solito si collega il substrato al potenziale maggiore dell'intero circuito (alimentazione) $V_B = V_{DD}$ oppure al source $V_B = V_S$.

NOTA: Non è possibile collegare il substrato al drain perché si violerebbe la condizione di interdizione del diodo source-substrato in quanto $V_S > V_D$ e quindi $V_{BS} = V_B - V_S = V_D - V_S < 0$.

Potenziale e tensioni di gate

Analogamente al n-mosfet, il potenziale di gate V_G controlla l'area di lavoro del p-mosfet. La tensione tra le due armature è variabile lungo la lunghezza del canale ed è compresa tra le tensioni di gate-source e gate-drain ai margini del canale:

- in prossimità del source $V_{C,source} = V_{GS} = V_G - V_S$
- in prossimità del drain $V_{C,drain} = V_{GD} = V_G - V_D$

NOTA: siccome $0 > V_S \geq V_D$, si ha $V_{GS} \leq V_{GD}$ (oppure $|V_{GS}| \geq |V_{GD}|$)

5.7 Aree di lavoro di un p-mosfet

Le aree di lavoro del p-mosfet sono analoghe a quelle del n-mosfet, ma con le polarità invertite:

- **p-mosfet spento o in interdizione** per $0 > V_{GS} > V_{TP}$:
il condensatore è in regime di svuotamento e non c'è corrente tra drain e source $I_{DS} = 0$
- **p-mosfet in conduzione lineare** per $V_{DS} > V_{GS} - V_{TP} \Leftrightarrow 0 > V_{TP} > V_{GD} > V_{GS}$:
si ha una corrente I_{DS} tra source e drain data dal movimento delle lacune nel canale di tipo p

$$I_{DS} = k_p V_{DS} \left(V_{GS} - V_{TP} - \frac{V_{DS}}{2} \right) \quad k_p = k'_p \cdot Z_p \quad k'_p = \mu_p \varepsilon / T_{OX} \quad Z_p = W_p / Z_p$$

- **p-mosfet in saturazione per pinchoff** per $V_{DS} < V_{GS} - V_{TP} \Leftrightarrow 0 > V_{GD} > V_{TP} > V_{GS}$:
il condensatore è in regime misto e si ha una corrente I_{DS} costante rispetto a V_{DS} :

$$I_{DS} = \frac{k_p}{2} (V_{GS} - V_{TP})^2$$

- **Modulazione di lunghezza di canale**:

Analogamente a quella dell'n-mosfet, applicando il fattore correttivo si ottiene:

$$I_{DS} = \frac{k_p}{2} (V_{GS} - V_{TP})^2 (1 + \lambda_p V_{DS}) \quad \text{con } V_{DS} < 0, \lambda_p < 0$$

- **p-mosfet in saturazione di velocità** per $V_{DS} < V_{DSATP}$:

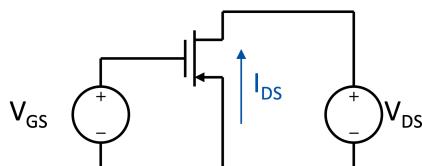
Analogamente a quello dell'n-mosfet, si ha una corrente I_{DS} costante rispetto a V_{DS} :

$$I_{DS} = k_p V_{DSATP} \left(V_{GS} - V_{TP} - \frac{V_{DSATP}}{2} \right) (1 + \lambda_p V_{DS}) \quad \text{con } V_{DSATP} < 0$$

(alcuni valori tipici per p-mosfet: $\mu_p = 200 \text{ cm}^2/\text{Vs}$ $v_{sat,p} \approx 4 \cdot 10^6 \text{ cm/s}$ $E_C \approx 2 \text{ V/cm}$)

5.8 Curve caratteristiche di corrente-tensione di un p-mosfet

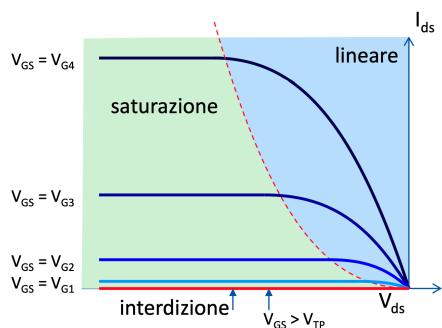
Alllo stesso modo dell'n-mosfet, si definiscono le curve caratteristiche di un p-mosfet, studiate attraverso un circuito simile:



- **caratteristica di uscita** con I_{DS} in funzione di V_{DS} per valori costanti di V_{GS}
- **caratteristica di trasferimento o transcaratteristica** con I_{DS} in funzione di V_{GS} per valori costanti di V_{DS}

Caratteristica di uscita $I_{DS} - V_{DS}$

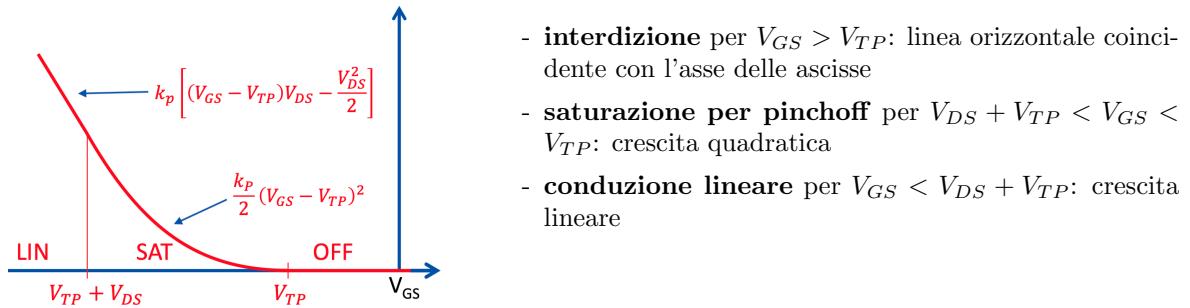
Rispetto al n-mosfet, la caratteristica di uscita $I_{DS} - V_{DS}$ del p-mosfet è speculare rispetto all'asse delle ordinate, mantenendo invariate le forme delle curve e il posizionamento delle aree di funzionamento:



- **interdizione** per $V_{GS} > 0$: linea orizzontale coincidente con l'asse delle ascisse
- **conduzione lineare** per $V_{DS} > V_{GS} - V_{TP}$: a destra della linea di saturazione
- **saturazione per pinchoff** per $V_{DS} < V_{GS} - V_{TP}$: a sinistra della linea di saturazione
- **curva di saturazione**: separa la regione di funzionamento lineare e quella di saturazione, è costituita dai punti in cui $V_{DS} = V_{GS} - V_{TP}$

Transcaratteristica I_{DS} - V_{GS}

Rispetto al n-mosfet, la transcaratteristica I_{DS} - V_{GS} del p-mosfet è speculare rispetto all'asse delle ordinate, mantenendo invariate le forme delle curve e il posizionamento delle aree di funzionamento:



5.9 Modello a canale corto di un p-mosfet

Equazione generale per I_{DS}

Anche per il p-mosfet si può utilizzare un'unica equazione per la corrente I_{DS} che tiene conto di tutti i fenomeni fisici:

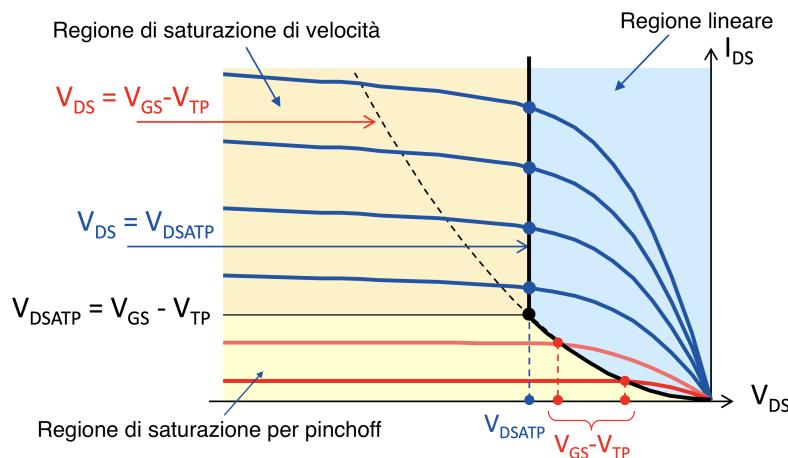
- per $V_{GS} > V_{TP}$ mosfet in interdizione e $I_{DS} = 0$
- per $V_{GS} < V_{TP}$ mosfet in conduzione con I_{DS} che dipende da V_{MAX} :

$$I_{DS} = k'_p Z_p V_{MAX} \left(V_{GS} - V_{TP} - \frac{V_{MAX}}{2} \right) (1 + \lambda_p V_{DS}) \quad k'_p = \frac{\mu_p \varepsilon}{T_{OX}} \quad Z_p = \frac{W_p}{L_p}$$

$$V_{MAX} = \max \left\{ \begin{array}{ll} V_{DS} & \text{regime lineare} \\ V_{GS} - V_{TP} & \text{saturazione per pinchoff} \\ V_{DSATP} & \text{saturazione per velocità} \end{array} \right\}$$

Caratteristica di uscita del modello a canale corto

La caratteristica di uscita complessiva del modello a canale corto del p-mosfet risulta come segue:

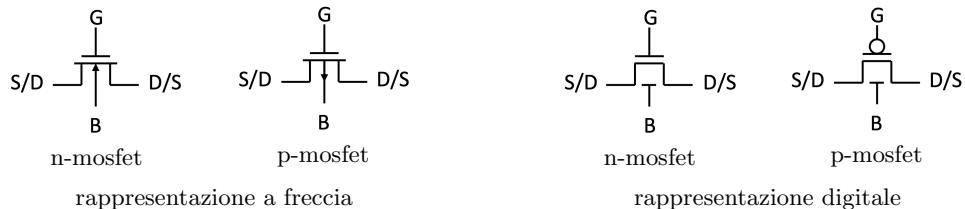


- la regione di conduzione lineare e la regione di saturazione per pinchoff sono separate dalla curva di saturazione per pinchoff data dalla parabola $I_{DS} = k_p/2 \cdot V_{DS}^2$ con $V_{DS} = V_{GS} - V_{TP}$
- la regione di conduzione lineare e la regione di saturazione per velocità sono separate dalla linea verticale $V_{DS} = V_{DSATP}$
- la regione di saturazione per pinchoff e la regione di saturazione per velocità sono separate dalla linea orizzontale $I_{DS} = k_p/2 \cdot V_{DS}^2$ con $V_{DS} = V_{DSATP}$

5.10 Simbologia e rappresentazione circuitale dei mosfet

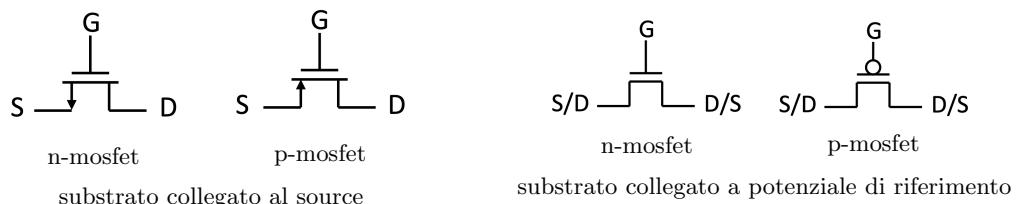
Simbologia a 4 terminali - con substrato

Nelle simbologie a 4 terminali non è possibile identificare univocamente il drain e il source in quanto sono perfettamente identici. Nelle rappresentazioni a freccia, la freccia sul terminale del substrato indica il tipo di mosfet ed è concorde con il flusso di corrente nel diodo formato tra substrato e source/drain.



Simbologia a 3 terminali - senza substrato

Nelle simbologie a 3 terminali, il substrato viene omesso in quanto collegato al source o ad un potenziale di riferimento. Quando il substrato è collegato al source, il terminale source è identificato con la freccia concorde al flusso di corrente tra source e drain. Quando il substrato è collegato ad un potenziale di riferimento, il terminale source non è identificabile univocamente in quanto non ci sono frecce.

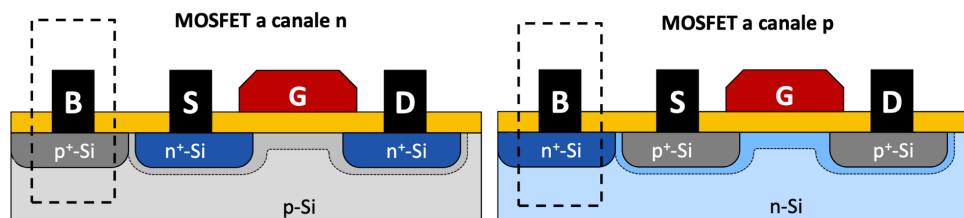


5.11 Struttura reale del mosfet

Per ottimizzare le prestazioni, lo spazio e il processo produttivo vengono apportate alcune modifiche al nodo di substrato (B):

- al posto di trovarsi sotto, il substrato viene realizzato sulla parte superiore del transistor a lato del source o del drain per facilitare il processo di fabbricazione
- in prossimità dell'elettrodo, il substrato viene ulteriormente drogato dello stesso tipo del substrato (p^+ in n-mosfet o n^+ in p-mosfet) per ridurre la resistenza di contatto

Di seguito sono riportate le illustrazioni delle strutture reali di un n-mosfet e di un p-mosfet:



5.12 Effetto Body e variazione della tensione di soglia

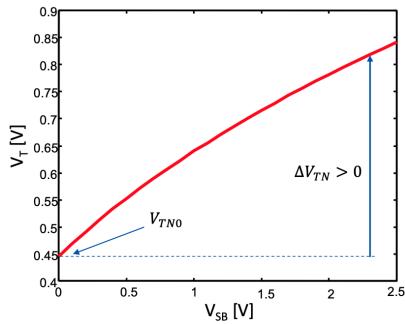
Quando il substrato non è collegato al source, ma ad un potenziale di riferimento (massa o alimentazione), la tensione tra substrato e source V_{BS} può essere diversa da zero. Questa tensione fa variare la tensione di soglia del mosfet secondo le relazioni (in base al tipo di mosfet):

$$V_{TN} = V_{TN0} + \gamma_n \left(\sqrt{V_{SB} + 2\phi_n} - \sqrt{2\phi_n} \right) \quad \gamma_n = \frac{\sqrt{2qN_D\varepsilon_{Si}}}{C_{OX}} \quad \phi_n = \frac{k_B T}{q} \ln \left(\frac{N_A}{n_i} \right)$$

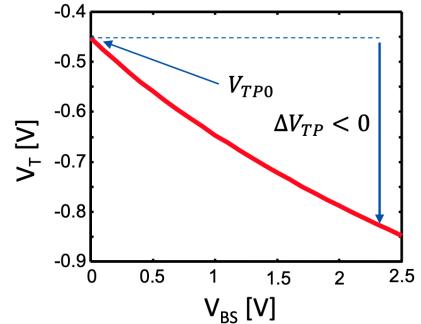
$$V_{TP} = V_{TP0} - \gamma_p \left(\sqrt{V_{BS} + 2\phi_p} - \sqrt{2\phi_p} \right) \quad \gamma_p = \frac{\sqrt{2qN_A\varepsilon_{Si}}}{C_{OX}} \quad \phi_p = \frac{k_B T}{q} \ln \left(\frac{N_D}{n_i} \right)$$

- V_{TN0} e V_{TP0} : tensione di soglia per $V_{SB} = 0$
- γ e ϕ : parametri legati al droggaggio e allo spessore dell'ossido
- C_{OX} : capacità per unità di area dell'ossido

Rappresentando graficamente la variazione della tensione di soglia in funzione di V_{SB} , si ottiene:



Variazione di V_{TN} per n-mosfet



Variazione di V_{TP} per p-mosfet

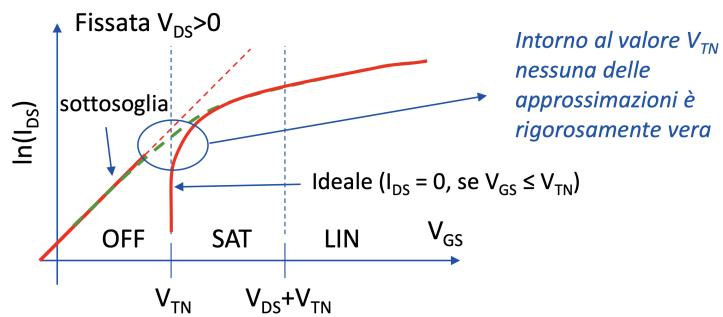
5.13 Corrente di sottosoglia

La corrente di sottosoglia è una piccola corrente che scorre tra drain e source anche quando il mosfet è in interdizione ($V_{GS} < V_T$). Questa corrente è molto debole e vale:

$$I_{DS} = I_{0n} e^{\frac{V_{GS}-V_{TN}}{nV_T}} \left(1 - e^{-\frac{V_{DS}}{nV_T}} \right) \quad \text{per n-mosfet}$$

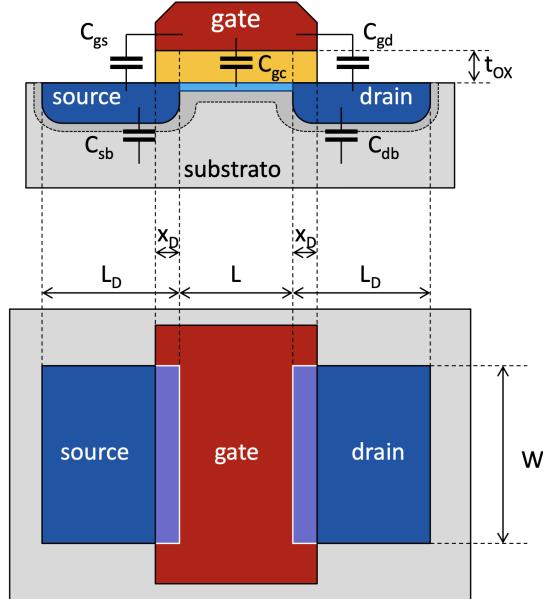
$$I_{DS} = I_{0p} e^{-\frac{V_{GS}-V_{TP}}{nV_T}} \left(1 - e^{\frac{V_{DS}}{nV_T}} \right) \quad \text{per p-mosfet}$$

Analizzando la transcaratteristica per un n-mosfet si osserva che la corrente di sottosoglia decresce esponenzialmente per $V_{GS} < V_{TN}$ (siccome le ordinate sono in scala logaritmica, la curva appare lineare):



5.14 Capacità parassite dei mosfet

Avendo numerose giunzioni pn che si vengono a formare, i mosfet presentano delle capacità parassite che influenzano il loro funzionamento alle alte frequenze, come evidenziato in figura. Le capacità parassite sono indipendenti dal tipo di mosfet (n-mosfet o p-mosfet), per cui le conclusioni ottenute valgono sia per un n-mosfet che per un p-mosfet.



Le capacità parassite per ogni coppia di terminali sono:

- C_{sb} : capacità source-substrato $C_{sb} \approx C_{j0} \cdot L_D \cdot W$
- C_{db} : capacità drain-substrato $C_{db} \approx C_{j0} \cdot L_D \cdot W$
- C_{gs} : capacità gate-source $C_{gs} \approx C_{gs0} \cdot W$
- C_{gd} : capacità gate-drain $C_{gd} \approx C_{gd0} \cdot W$
- C_{gc} : capacità gate-substrato (del condensatore mos) $C_{gc} = C_{OX} \cdot L \cdot W$

Le capacità parassite complessive di ogni nodo rispetto al substrato si calcolano come capacità equivalente della rete tra il nodo in esame e il substrato, ottenuta cortocircuitando tutti gli altri nodi al substrato. Di seguito sono riportate le espressioni delle capacità parassite totali:

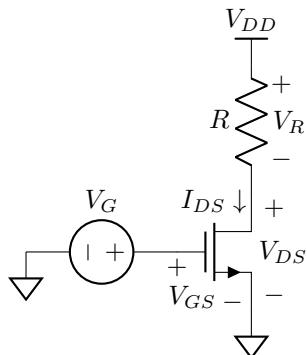
$$\begin{aligned} C_{source} &= C_{sb} + C_{gs} = (C_{j0}L_D + C_{gs0}) \cdot W = C_{s0} \cdot W && \text{con } C_{s0} = C_{j0}L_D + C_{gs0} \\ C_{drain} &= C_{db} + C_{gd} = (C_{j0}L_D + C_{gd0}) \cdot W = C_{d0} \cdot W && \text{con } C_{d0} = C_{j0}L_D + C_{gd0} \\ C_{gate} &= C_{gs} + C_{gd} + C_{gc} = (C_{gs0} + C_{gd0} + C_{OX}L) \cdot W \approx C_{g0} \cdot WL && \text{per } C_{gs} + C_{gd} \ll C_{OX}L \end{aligned}$$

NOTA: Dall'analisi dei circuiti RC in cui si hanno due condensatori C_1 e C_2 con un terminale in comune e l'altro collegato rispettivamente a massa e a V_{DD} , si osserva che la capacità equivalente tra i due nodi è data dalla somma delle due capacità $C_{eq} = C_1 + C_2$. Questo avviene anche se effettivamente i due condensatori non sono collegati in parallelo, in quanto la corrente che arriva al nodo comune si divide tra i due condensatori in funzione della loro capacità ed è come se fossero collegati in parallelo.

6 Circuiti con i mosfet

6.1 Mosfet in serie a una resistenza

Circuito n-mosfet e resistenza



Si costruisce un circuito con un n-mosfet e una resistenza come illustrato di lato, in cui:

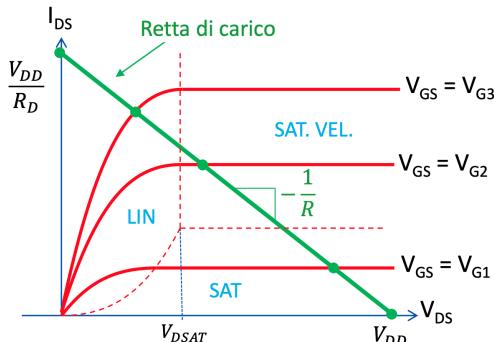
- il terminale di drain è collegato a una tensione di alimentazione V_{DD} tramite una resistenza di carico R ;
- il terminale di source è collegato a massa;
- il terminale di gate è collegato a un generatore di tensione V_G .

L'utilizzo di un n-mosfet al posto di un p-mosfet è perfettamente arbitrario e non cambia nulla nel procedimento di soluzione del circuito.

Soluzione del circuito per via grafica

La risoluzione del circuito consiste nel trovare la corrente I_{DS} e la tensione V_{DS} che soddisfano sia la caratteristica del mosfet sia la caratteristica della resistenza R . Dalle LKC si impone $I_{DS} = I_R$ e si risolve l'equazione in funzione di V_{DS} .

Un'alternativa alla risoluzione analitica è la risoluzione grafica, che consiste nel tracciare sullo stesso grafico la caratteristica di uscita del mosfet (I_{DS} in funzione di V_{DS}) e la caratteristica della resistenza R sempre in funzione di V_{DS} e trovare il punto di intersezione tra le due curve.



Un'alternativa alla risoluzione analitica è la risoluzione grafica, che consiste nel tracciare sullo stesso grafico la caratteristica di uscita del mosfet (I_{DS} in funzione di V_{DS}) e la caratteristica della resistenza R sempre in funzione di V_{DS} e trovare il punto di intersezione tra le due curve.

Modello a canale lungo

Il modello a canale lungo del mosfet prevede che il transistor operi in una delle tre regioni (interdizione, lineare o saturazione per pinchoff) senza considerare gli effetti di modulazione di lunghezza di canale e saturazione di velocità. Per risolvere il circuito si procede come segue:

1. si determina se il mosfet è “acceso” o “spento” confrontando V_{GS} con la tensione di soglia V_{TN} :
 - se $V_{GS} < V_{TN}$, il mosfet è in interdizione e $I_{DS} = 0$
 - se $V_{GS} \geq V_{TN}$, il mosfet è acceso e si procede al passo successivo
2. si ipotizza che il mosfet sia in uno dei due regimi di conduzione (lineare o saturazione) e si risolve la rete ponendo $I_{DS} = I_R$ (per le LKC), utilizzando la formula I_{DS} corrispondente all'ipotesi fatta e si risolve per V_{DS} :
 - se si ipotizza il regime lineare: $k_n V_{DS} \left(V_{GS} - V_{TN} - \frac{V_{DS}}{2} \right) = \frac{V_{DD} - V_{DS}}{R}$
 - se si ipotizza il regime di saturazione: $\frac{k_n}{2} (V_{GS} - V_{TN})^2 = \frac{V_{DD} - V_{DS}}{R}$
3. si verifica se l'ipotesi fatta sul regime di funzionamento è corretta:
 - se si è ipotizzato il regime lineare, si verifica che $V_{DS} < V_{GS} - V_{TN}$;
 - se si è ipotizzato il regime di saturazione, si verifica che $V_{DS} \geq V_{GS} - V_{TN}$.
4. se l'ipotesi è corretta, si è trovata la soluzione; altrimenti si ripete il passo 2 con l'altra ipotesi.

Modello a canale corto

La soluzione del circuito con il modello a canale corto è analoga a quella con il modello a canale lungo, con la differenza che si prevede anche l'effetto di modulazione di lunghezza di canale e la possibilità che il mosfet lavori in saturazione di velocità. In particolare si procede come segue:

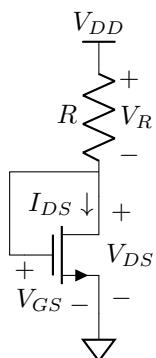
1. si determina se il mosfet è “acceso” o “spento” come nel modello a canale lungo;
2. si calcolano, se possibile, i tre valori di V_{MIN} (V_{DS} , $V_{GS} - V_{TN}$, V_{DSATN}) e se si conoscono già due dei valori (esempio $V_{GS} - V_{TN}$ e V_{DSATN}), si può escludere a priori quello maggiore (ovvero quello che sicuramente non sarà il minimo), riducendo così il numero di ipotesi da fare;
3. si ipotizza che il mosfet sia in uno dei tre regimi di conduzione (lineare, saturazione per pinchoff o saturazione di velocità) escludendone, se possibile, uno come spiegato al passo precedente, e si risolve la rete come nel modello a canale lungo:

- in regime lineare: $k_n V_{DS} \left(V_{GS} - V_{TN} - \frac{V_{DS}}{2} \right) (1 + \lambda V_{DS}) = \frac{V_{DD} - V_{DS}}{R}$
- in saturazione per pinchoff: $\frac{k_n}{2} (V_{GS} - V_{TN})^2 (1 + \lambda V_{DS}) = \frac{V_{DD} - V_{DS}}{R}$
- in saturazione di velocità: $k_n V_{DSATN} \left(V_{GS} - V_{TN} - \frac{V_{DSATN}}{2} \right) (1 + \lambda V_{DS}) = \frac{V_{DD} - V_{DS}}{R}$

4. si verifica se l'ipotesi fatta sul regime di funzionamento è corretta:
 - se si è ipotizzato il regime lineare: $V_{DS} = \min\{V_{GS} - V_{TN}, V_{DSATN}, V_{DS}\}$;
 - se si è ipotizzato il regime di sat. per pinchoff: $V_{GS} - V_{TN} = \min\{V_{GS} - V_{TN}, V_{DSATN}, V_{DS}\}$;
 - se si è ipotizzato il regime di sat. di velocità: $V_{DSATN} = \min\{V_{GS} - V_{TN}, V_{DSATN}, V_{DS}\}$;
5. se l'ipotesi è corretta, si è trovata la soluzione; altrimenti si ripete il passo 3 con un'altra ipotesi.

6.2 Mosfet connesso a diodo

Circuito di un n-mosfet connesso a diodo



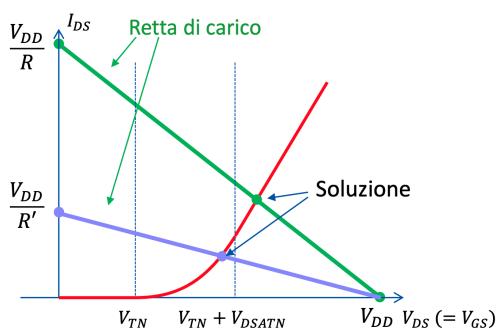
Si costruisce un circuito con un n-mosfet e una resistenza come illustrato di lato, in cui:

- il terminale di source è collegato a massa;
- il terminale di drain è collegato a una tensione di alimentazione V_{DD} tramite una resistenza di carico R ;
- il terminale di gate è cortocircuitato al terminale di drain e $V_{GS} = V_{DS}$

L'utilizzo di un n-mosfet al posto di un p-mosfet è perfettamente arbitrario e non cambia nulla nel procedimento di soluzione del circuito.

Considerazioni sul regime di funzionamento e risoluzione grafica

Si osserva che il mosfet connesso in questo modo, se acceso, lavora in saturazione per pinchoff o in saturazione di velocità, per cui si comporta in modo simile a un diodo ideale con soglia V_{TN} :



- se $V_{GS} < V_{TN}$, il mosfet è spento con $I_{DS} = 0$;
- se $V_{GS} \geq V_{TN}$, il mosfet è acceso e lavora in saturazione per pinchoff $V_{DS} = V_{GS} > V_{GS} - V_{TN}$ oppure in sat. di velocità se $V_{DS} = V_{GS} > V_{DSATN}$

Si nota che la caratteristica di uscita del mosfet in questo caso è simile a quella di un diodo con soglia V_{TN} . Come nel caso precedente, la risoluzione grafica consiste nel trovare le intersezioni delle due caratteristiche di uscita del mosfet e della resistenza R .

Analisi del circuito

Per risolvere il circuito si procede come segue:

1. si determina se il mosfet è “acceso” o “spento” confrontando V_{GS} con la tensione di soglia V_{TN} :
 - se $V_{GS} < V_{TN}$, il mosfet è in interdizione e $I_{DS} = 0$
 - se $V_{GS} \geq V_{TN}$, il mosfet è acceso e si procede al passo successivo
2. se si utilizza il modello a canale lungo, si utilizza I_{DS} in saturazione per pinchoff e si impone la condizione $I_{DS} = I_R \rightarrow I_{DS} = \frac{k_n}{2}(V_{GS} - V_{TN})^2 = \frac{V_{DD} - V_{DS}}{R}$; non serve verificare il regime di funzionamento, in quanto il mosfet connesso in questo modo lavora sempre in saturazione
3. se si utilizza il modello a canale corto, bisogna ipotizzare il regime di saturazione (pinchoff o velocità) e risolvere il circuito utilizzando la formula corrispondente:
 - in saturazione per pinchoff: $\frac{k_n}{2}(V_{GS} - V_{TN})^2(1 + \lambda V_{DS}) = \frac{V_{DD} - V_{DS}}{R}$
 - in saturazione di velocità: $k_n V_{DSATN} \left(V_{GS} - V_{TN} - \frac{V_{DSATN}}{2} \right) (1 + \lambda V_{DS}) = \frac{V_{DD} - V_{DS}}{R}$
4. si verifica se l’ipotesi fatta sul regime di funzionamento è corretta:
 - se si è ipotizzato il regime di sat. per pinchoff: $V_{GS} - V_{TN} = \min\{V_{GS} - V_{TN}, V_{DSATN}, V_{DS}\}$;
 - se si è ipotizzato il regime di sat. di velocità: $V_{DSATN} = \min\{V_{GS} - V_{TN}, V_{DSATN}, V_{DS}\}$;
5. se l’ipotesi è corretta, si è trovata la soluzione; altrimenti si ripete il passo 3 con un’altra ipotesi

6.3 Mosfet come generatore di corrente

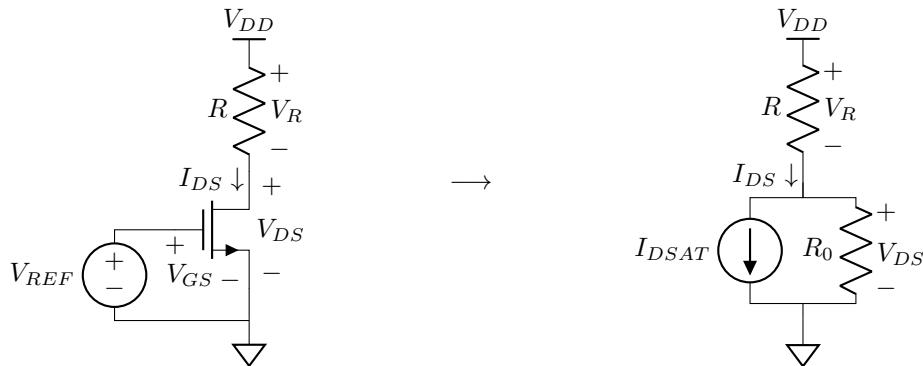
Circuito di un n-mosfet connesso come generatore di corrente

Si costruisce un circuito con un n-mosfet e una resistenza come illustrato sotto, in cui:

- il terminale di source è collegato a massa;
- il terminale di drain è collegato a una tensione di alimentazione V_{DD} tramite una resistenza di carico R ;
- il terminale di gate è cortocircuitato al terminale di source e $V_{GS} = V_{REF}$

Si suppone, inoltre, che il valore di V_{REF} sia tale da mantenere il mosfet sempre acceso e in regime di saturazione per pinchoff o in saturazione di velocità, ovvero che sia soddisfatta la condizione:

$$V_{DS} > \min\{V_{REF} - V_{TN}, V_{DSATN}\}$$



L’utilizzo di un n-mosfet al posto di un p-mosfet è perfettamente arbitrario e non cambia nulla nel procedimento di soluzione del circuito.

Considerazioni sul regime di funzionamento e risoluzione grafica

Il mosfet in saturazione, connesso in questo modo, si può modellare con un generatore lineare di corrente secondo il teorema di Norton, come illustrato sopra. In saturazione, infatti, la corrente I_{DS} è direttamente proporzionale a V_{DS} tramite il parametro λ_n :

$$I_{DS} = I_{DSAT}(1 + \lambda_n V_{DS}) = I_{DSAT} + \lambda_n I_{DSAT} V_{DS} = I_{DSAT} + R_0 V_{DS} \quad \text{con } R_0 = \frac{1}{\lambda_n I_{DSAT}}$$

Si osserva che la corrente I_{DS} ha due contributi:

- una corrente costante I_{DSAT} di saturazione del mosfet, corrisponde alla corrente erogata dal generatore ideale di corrente per Norton;
- una corrente variabile $R_0 V_{DS}$ che dipende linearmente dalla tensione V_{DS} e che può essere modellata con una resistenza R_0 in parallelo con il generatore ideale di corrente.

Si nota che se $\lambda = 0 \rightarrow I_{DS} = I_{DSAT} \rightarrow R_0 = \infty$, ovvero se non si considera l'effetto di modulazione di lunghezza di canale, il mosfet si comporta come un generatore ideale di corrente.

Il generatore lineare di corrente di Norton ha dei vincoli sul suo funzionamento, in quanto deve essere soddisfatta la condizione che il mosfet rimanga in saturazione per pinchoff o in saturazione di velocità:

$$V_{DS} > \min\{V_{REF} - V_{TN} \text{ sat. per pinchoff}, V_{DSATN} \text{ sat. di velocità}\}$$

Se questa condizione non è soddisfatta, il mosfet esce dal regime di saturazione e il modello con il generatore lineare di corrente non è più valido.

Analisi del circuito

Per risolvere il circuito utilizzando il modello a canale corto si procede come segue:

1. si impone che il mosfet sia acceso e che sia in saturazione (per pinchoff o di velocità) per cui si deve avere: $V_{REF} > V_{TN}$ e $V_{DS} > \min\{V_{REF} - V_{TN}, V_{DSATN}\}$
2. si ipotizza il regime di saturazione (pinchoff o velocità) e si calcola I_{DSAT} e R_0 :
 - in regime di saturazione per pinchoff: $I_{DSAT} = \frac{k_n}{2}(V_{REF} - V_{TN})^2$
 - in regime di saturazione di velocità: $I_{DSAT} = k_n V_{DSATN} \left(V_{REF} - V_{TN} - \frac{V_{DSATN}}{2} \right)$
 - a prescindere dal regime di saturazione $R_0 = 1/(\lambda_n I_{DSAT})$
3. si applica il teorema di Norton, ovvero si sostituisce il mosfet con il generatore lineare di corrente e si risolve il circuito (es. per sovrapposizione degli effetti) in funzione di V_{DS} :

$$V_{DS} = V_{DD} \frac{R_0}{R + R_0} - I_{DSAT} R_0 \frac{R}{R + R_0}$$

4. si verifica l'ipotesi sul regime di saturazione sia soddisfatta:
 - in regime di saturazione per pinchoff: $V_{DS} > V_{REF} - V_{TN}$
 - in regime di saturazione di velocità: $V_{DS} > V_{DSATN}$
5. se la condizione è soddisfatta, si è trovata la soluzione; altrimenti si deve ripetere il procedimento per l'altro regime di saturazione

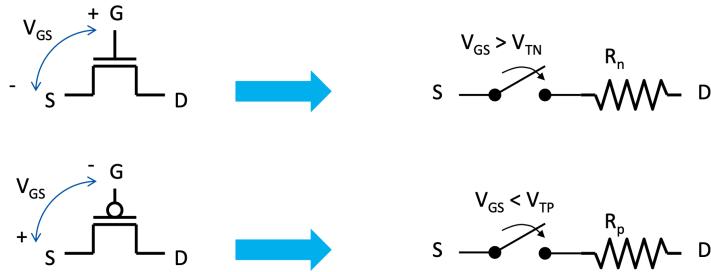
Il modello a canale lungo è un caso particolare del modello a canale corto con $\lambda_n = 0$ e $R_0 = \infty$, per cui il mosfet si può modellare come un generatore ideale di corrente con corrente I_{DSAT} data dal regime di saturazione per pinchoff. Non prevedendo l'effetto di modulazione di lunghezza di canale e il fenomeno di saturazione di velocità, alla fine è sufficiente verificare solo la condizione $V_{DS} > V_{REF} - V_{TN}$.

6.4 Mosfet usato come interruttore

Modellizzazione del mosfet come interruttore

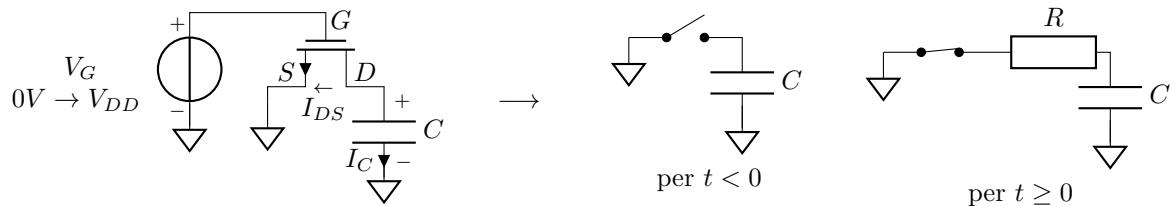
Nell'elettronica digitale i mosfet sono utilizzati come interruttori ON/OFF. È possibile, infatti, modellare un mosfet come un interruttore ideale pilotato dalla tensione di gate V_{GS} in serie ad una resistenza R_n per gli nmos o R_p per i pmos. Si considerano, quindi, soltanto due regimi di funzionamento:

- mosfet "spento": interruttore aperto, $I_{DS} = 0$, per $V_{GS} < V_{TN}$ (nmos) o $V_{GS} > V_{TP}$ (pmos);
- mosfet "acceso": resistore di resistenza R_n (nmos) o R_p (pmos).



6.5 Scarica di un condensatore con nmos

Circuito di scarica



Equazioni del circuito per le leggi di Kirchhoff e la struttura della rete

$$V_G(t) = \begin{cases} 0V & t < 0 \\ V_{DD} & t \geq 0 \end{cases} \quad V_C(t \leq 0) = V_{DD} \quad V_{GS}(t) = V_G(t) \quad I_{DS}(t) + I_C(t) = 0$$

Analisi delle fasi del transitorio

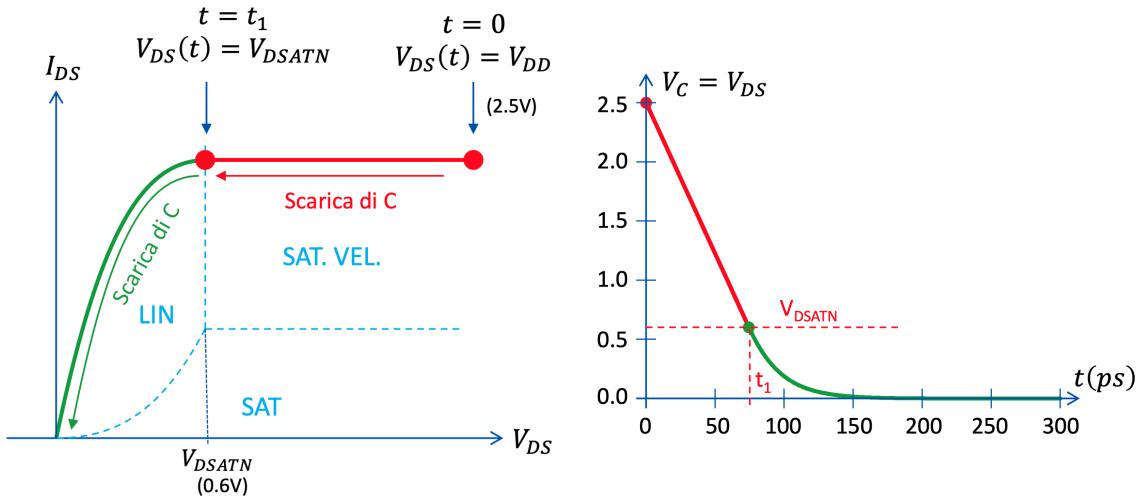
Si analizzano le variabili del circuito per $t < 0$, $t = 0$, $t > 0$ e $t \rightarrow \infty$:

- per $t < 0$: l'nmos è spento poiché $V_{GS} = 0V < V_{TN}$ e il condensatore è carico a V_{DD} ;
- per $t = 0$: l'nmos si accende poiché $V_{GS} = V_{DD} > V_{TN}$;
- per $t > 0$: l'nmos rimane acceso poiché $V_{GS} = V_{DD} > V_{TN}$ e il condensatore si scarica;
- per $t \rightarrow \infty$: a regime l'nmos rimane acceso e il condensatore si scarica completamente.

Analisi grafica delle fasi del transitorio

Si suppone, sotto certi dati, che il mosfet lavori soltanto in saturazione di velocità o in regime lineare. Si osserva che la scarica del condensatore avviene in due fasi (con un punto intermedio $t = t_1$):

1. per $0 < t < t_1$, $V_{DS} > V_{DSATN}$ il mosfet lavora in regime di saturazione di velocità e la corrente di scarica è costante pari a I_{DSATN} :
2. per $t = t_1$, $V_{DS} = V_{DSATN}$ il mosfet passa dal regime di saturazione di velocità a regime lineare;
3. per $t > t_1$, $V_{DS} < V_{DSATN}$ il mosfet lavora in regime lineare e la corrente di scarica dipende dalla tensione $V_{DS}(t) = V_C(t)$ e quindi dalla tensione del condensatore



Soluzione analitica del circuito

Si analizza la prima fase di scarica del condensatore per $0 < t < t_1$ (saturazione di velocità):

$$I_C(t) = C \frac{dV_C}{dt} \quad I_{DSATN} = k'_n Z_n V_{DSATN} \left(V_{DD} - V_{TN} - \frac{V_{DSATN}}{2} \right)$$

$$I_C(t) = -I_{DSATN} \quad \rightarrow \quad V_C(t) = V_{DD} - \frac{I_{DSATN}}{C} t$$

Si analizza il passaggio dalla prima alla seconda fase di per $t = t_1$:

$$V_{GS}(t_1) = V_{DSATN} \quad \rightarrow \quad V_C(t_1) = V_{DSATN} \quad \rightarrow \quad t_1 = \frac{C}{I_{DSATN}} (V_{DD} - V_{DSATN})$$

Si analizza la seconda fase di scarica del condensatore per $t > t_1$ (regime lineare):

$$I_C(t) = C \frac{dV_C}{dt} \quad I_{DS}(t) = k'_n Z_n V_C(t) \left(V_{DD} - V_{TN} - \frac{V_C(t)}{2} \right)$$

$$I_C(t) = -I_{DS}(t) \quad \rightarrow \quad V_C(t) = \frac{1}{A e^{\alpha(t-t_1)} + B}$$

$$\alpha = \frac{k'_n Z_n}{C} (V_{DD} - V_{TN}) \quad A = \frac{1}{V_{DSATN}} - \frac{1}{2(V_{DD} - V_{TN})} \quad B = \frac{1}{2(V_{DD} - V_{TN})}$$

Carica del condensatore a regime

Per $t \rightarrow \infty$, la tensione sul condensatore si riduce a zero e il condensatore si scarica completamente:

$$\lim_{t \rightarrow \infty} V_C(t) = 0V$$

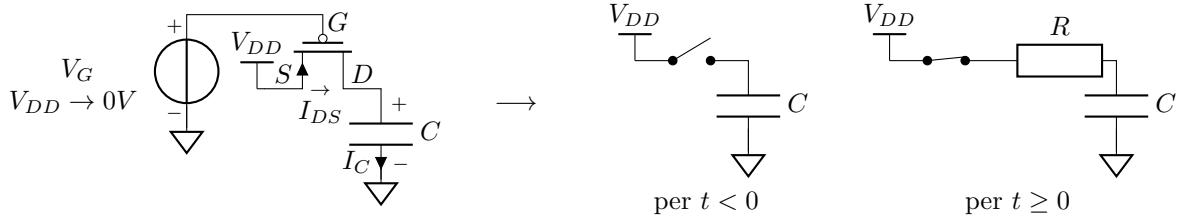
Tempo di dimezzamento di $V_C(t)$

Il tempo di dimezzamento $t_{1/2}$ è il tempo necessario affinché la tensione sul condensatore si riduca alla metà del suo valore iniziale V_{DD} , ovvero quando $V_C(t_{1/2}) = V_{DD}/2$. Si osserva che tale valore viene assunto durante la prima fase di scarica, in cui il mosfet lavora in saturazione di velocità. Si calcola, quindi, $t_{1/2}$ come segue:

$$V_C(t_{1/2}) = \frac{V_{DD}}{2} \quad \rightarrow \quad t_{1/2} = \frac{CV_{DD}}{2I_{DSATN}}$$

6.6 Carica di un condensatore con pmos

Circuito di carica



Equazioni del circuito per le leggi di Kirchhoff e la struttura della rete

$$V_G(t) = \begin{cases} V_{DD} & t < 0 \\ 0V & t \geq 0 \end{cases} \quad V_C(t \leq 0) = 0V \quad V_{GS}(t) = V_G(t) - V_{DD} \quad V_{DS}(t) = V_C(t) - V_{DD} \quad I_{DS} = I_C(t)$$

Analisi delle fasi del transitorio

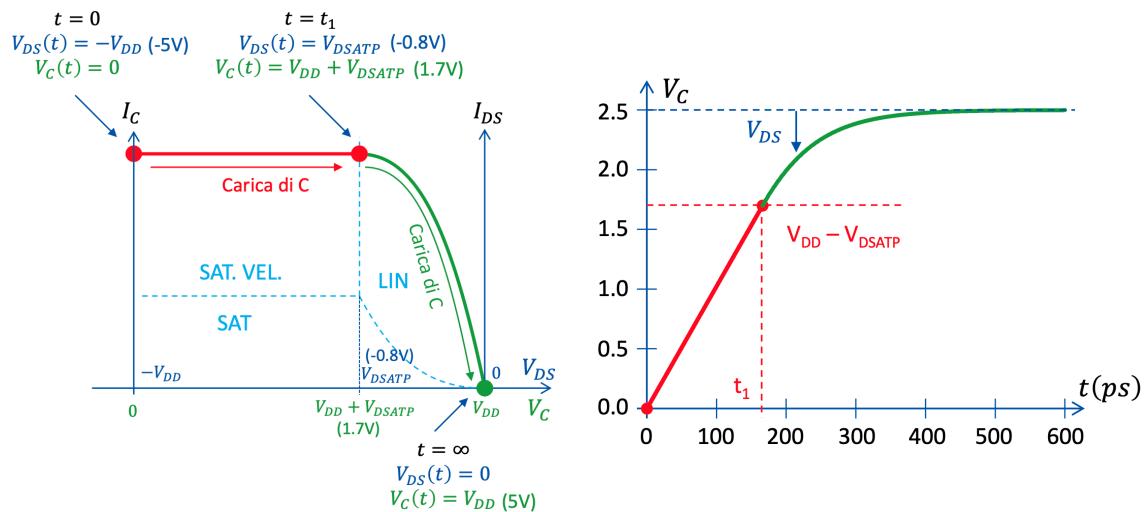
Si analizzano le variabili del circuito per $t < 0$, $t = 0$, $t > 0$ e $t \rightarrow \infty$:

- per $t < 0$: il pmos è spento poiché $V_{GS} = 0V > V_{TP}$ e il condensatore è scarico con $V_C = 0V$;
- per $t = 0$: il pmos si accende poiché $V_{GS} = -V_{DD} < V_{TP}$;
- per $t > 0$: il pmos rimane acceso poiché $V_{GS} = -V_{DD} < V_{TP}$ e il condensatore si carica;
- per $t \rightarrow \infty$: a regime il pmos rimane acceso e il condensatore si carica completamente a V_{DD} .

Analisi grafica delle fasi del transitorio

Si suppone, sotto certi dati, che il mosfet lavori soltanto in saturazione di velocità o in regime lineare. Si osserva che la scarica del condensatore avviene in due fasi (con un punto intermedio $t = t_1$):

1. per $0 < t < t_1$, $V_{DS} < V_{DSATP}$ il mosfet lavora in regime di saturazione di velocità e la corrente di scarica è costante pari a I_{DSATP} :
2. per $t = t_1$, $V_{DS} = V_{DSATP}$ il mosfet passa dal regime di saturazione di velocità a regime lineare:
3. per $t > t_1$, $V_{DS} > V_{DSATP}$ il mosfet lavora in regime lineare e la corrente di scarica dipende dalla tensione $V_{DS}(t) = V_C(t) - V_{DD}$ e quindi dalla tensione del condensatore



Soluzione analitica del circuito

Si analizza la prima fase di scarica del condensatore per $0 < t < t_1$ (saturazione di velocità):

$$I_C(t) = C \frac{dV_C}{dt} \quad I_{DSATP} = k'_p Z_p V_{DSATP} \left(-V_{DD} - V_{TP} - \frac{V_{DSATP}}{2} \right)$$

$$I_C(t) = I_{DSATP} \quad \rightarrow \quad V_C(t) = \frac{I_{DSATP}}{C} t$$

Si analizza il passaggio dalla prima alla seconda fase di per $t = t_1$:

$$V_{DS}(t_1) = V_{DSATP} \quad \rightarrow \quad V_C(t_1) - V_{DD} = V_{DSATP} \quad \rightarrow \quad t_1 = \frac{C}{I_{DSATP}} (V_{DD} + V_{DSATP})$$

Si analizza la seconda fase di scarica del condensatore per $t > t_1$ (regime lineare):

$$I_C(t) = C \frac{dV_C}{dt} \quad I_{DS}(t) = k'_p Z_p (V_C(t) - V_{DD}) \left(-V_{DD} - V_{TP} - \frac{V_C(t) - V_{DD}}{2} \right)$$

$$I_C(t) = I_{DS}(t) \quad \rightarrow \quad V_C(t) = \frac{1}{A e^{\alpha(t-t_1)} + B} + V_{DD}$$

$$\alpha = \frac{k'_p Z_p}{C} (V_{DD} + V_{TP}) \quad A = \frac{1}{V_{DSATP}} + \frac{1}{2(V_{DD} + V_{TP})} \quad B = -\frac{1}{2(V_{DD} + V_{TP})}$$

Carica del condensatore a regime

Per $t \rightarrow \infty$, la tensione sul condensatore raggiunge il valore V_{DD} e il condensatore si carica completamente:

$$\lim_{t \rightarrow \infty} V_C(t) = V_{DD}$$

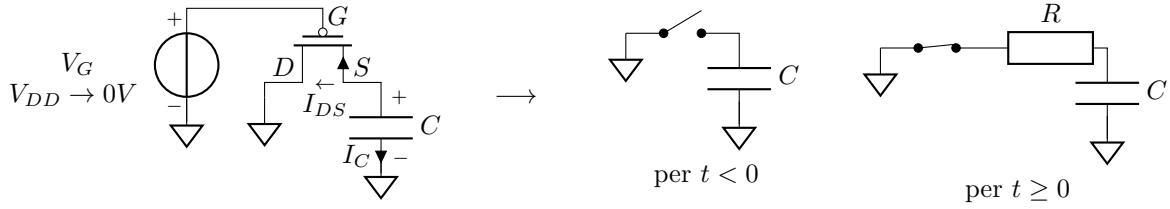
Tempo di dimezzamento di $V_C(t)$

Si analizza il tempo di dimezzamento $t_{1/2}$ osservando che tale valore viene assunto durante la prima fase di carica, in cui il mosfet lavora in saturazione di velocità:

$$V_C(t_{1/2}) = \frac{V_{DD}}{2} \quad \rightarrow \quad t_{1/2} = \frac{CV_{DD}}{2I_{DSATP}}$$

6.7 Scarica di un condensatore con un pmos

Circuito di scarica



Equazioni del circuito per le leggi di Kirchhoff e la struttura della rete

$$V_G(t) = \begin{cases} V_{DD} & t < 0 \\ 0V & t \geq 0 \end{cases} \quad V_C(t \leq 0) = V_{DD} \quad V_{GS}(t) = V_G(t) - V_C(t) \quad I_{DS} + I_C(t) = 0 \\ V_{DS}(t) = -V_C(t)$$

Analisi delle fasi del transitorio

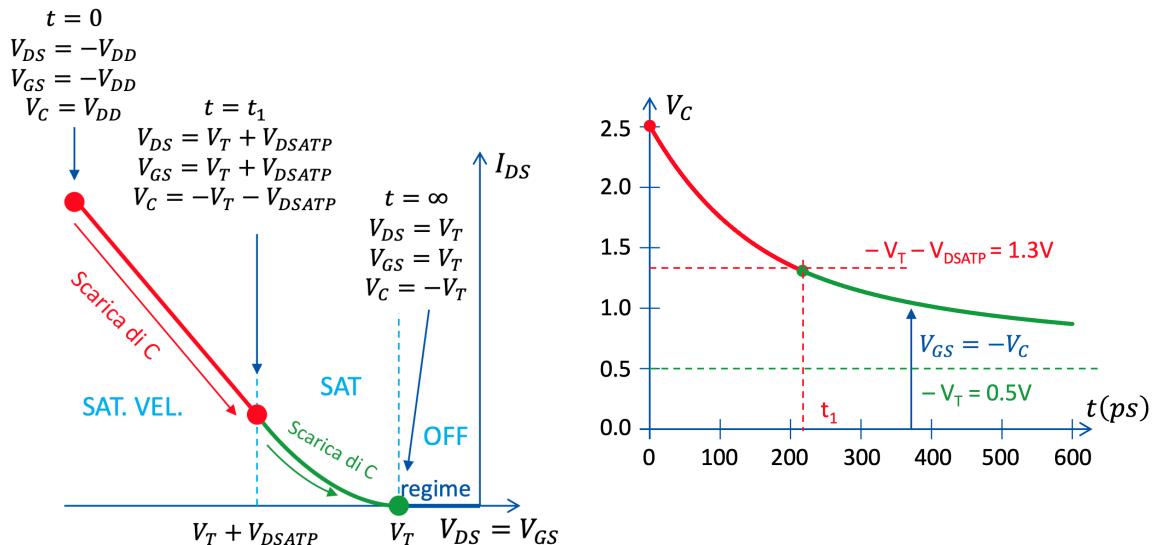
Si analizzano le variabili del circuito per $t < 0$, $t = 0$, $t > 0$ e $t \rightarrow \infty$:

- per $t < 0$: il pmos è spento poiché $V_{GS}(t) = 0V > V_{TP}$ e il condensatore è carico a V_{DD} ;
- per $t = 0$: il pmos si accende poiché $V_{GS}(t) = -V_{DD} < V_{TP}$;
- per $t > 0$: il pmos è acceso finché $V_{GS}(t) < V_{TP} \rightarrow V_C(t) > -V_{TP}$
- per $t \rightarrow \infty$: a regime il pmos si spegne per $V_{GS}(t) = V_{TP} \rightarrow V_C(t) = -V_{TP}$ lasciando il condensatore carico con tensione finale $V_C(t) = -V_{TP}$

Analisi grafica delle fasi del transitorio

Il pmos è connesso a diodo (gate e drain entrambi a massa per $t \geq 0$), quindi lavora solo in saturazione. Si osserva che la scarica del condensatore avviene in due fasi (con un punto intermedio $t = t_1$):

1. per $0 < t < t_1$, $V_{GS} - V_{TP} < V_{DSATP}$ il mosfet lavora in regime di saturazione di velocità e la corrente di scarica è costante pari a I_{DSATP} :
2. per $t = t_1$, $V_{GS} - V_{TP} = V_{DSATP}$ il mosfet passa dal regime di saturazione per velocità a quello di saturazione per pinchoff:
3. per $t > t_1$, $V_{GS} - V_{TP} > V_{DSATP}$ il mosfet lavora in saturazione per pinchoff e la corrente di scarica dipende da $V_{GS}(t) - V_{TP} = -V_C(t) - V_{TP}$ e quindi anche dalla tensione del condensatore $V_C(t)$



Soluzione analitica del circuito

Si analizza la prima fase di scarica del condensatore per $0 < t < t_1$ (saturazione di velocità):

$$I_C(t) = C \frac{dV_C}{dt} \quad I_{DSATP}(t) = k'_p Z_p V_{DSATP} \left(-V_C(t) - V_{TP} - \frac{V_{DSATP}}{2} \right)$$

$$I_C(t) = I_{DSATP}(t) \rightarrow V_C(t) = V_{DD} - \left(V_{TP} + \frac{V_{DSATP}}{2} \right) (1 - e^{-\alpha t}) \quad \text{con } \alpha = -\frac{k'_p Z_p V_{DSATP}}{C}$$

Si analizza il passaggio dalla prima alla seconda fase di per $t = t_1$:

$$V_{GS}(t_1) - V_{TP} = V_{DSATP} \rightarrow V_C(t_1) = -V_{TP} - V_{DSATP} \rightarrow t_1 = \frac{1}{\alpha} \ln \left(\frac{2V_{DD} + 2V_{TP} + V_{DSATP}}{-V_{DSATP}} \right)$$

Si analizza la seconda fase di scarica del condensatore per $t > t_1$ (saturazione per pinchoff):

$$I_C(t) = C \frac{dV_C}{dt} \quad I_{DS}(t) = \frac{k'_p Z_p}{2} (V_{GS}(t) - V_{TP})^2 = \frac{k'_p Z_p}{2} (-V_C(t) - V_{TP})^2$$

$$I_C(t) = I_{DS}(t) \rightarrow V_C(t) = -V_{TP} + \frac{2CV_{DSATP}}{k'_p Z_p V_{DSATP}(t - t_1) - 2C}$$

Valore di regime

Si analizza il valore di regime della tensione sul condensatore $V_C(t)$ per $t \rightarrow \infty$:

$$\lim_{t \rightarrow \infty} V_C(t) = -V_{TP}$$

Si ha, quindi, che il condensatore si scarica fino a raggiungere la tensione di soglia del pmos.

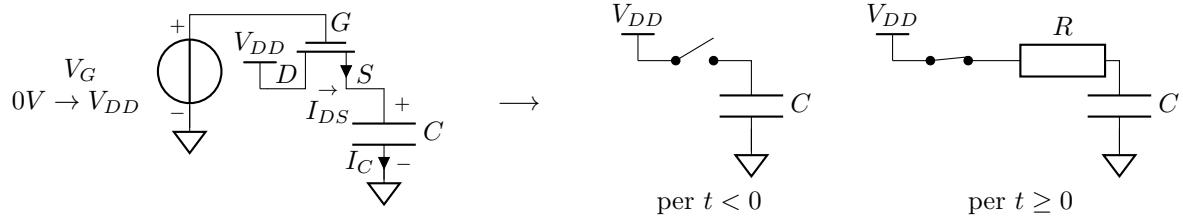
Tempo di dimezzamento di $V_C(t)$

Si analizza il tempo di dimezzamento $t_{1/2}$, osservando che tale valore viene assunto durante la seconda fase di scarica, in cui il mosfet lavora in saturazione per pinchoff:

$$V_C(t_{1/2}) = \frac{V_{DD}}{2} \rightarrow t_{1/2} = t_1 - 2C \frac{V_{DD} + 2V_{TP} + 2V_{DSATP}}{k'_p Z_p V_{DSATP}(V_{DD} + 2V_{TP})}$$

6.8 Carica di un condensatore con un nmos

Circuito di carica



Equazioni del circuito per le leggi di Kirchhoff e la struttura della rete

$$V_G(t) = \begin{cases} 0V & t < 0 \\ V_{DD} & t \geq 0 \end{cases} \quad V_C(t \leq 0) = 0V \quad \begin{aligned} V_{GS}(t) &= V_G(t) - V_C(t) \\ V_{DS}(t) &= V_{DD} - V_C(t) \end{aligned} \quad I_{DS} = I_C(t)$$

Analisi delle fasi del transitorio

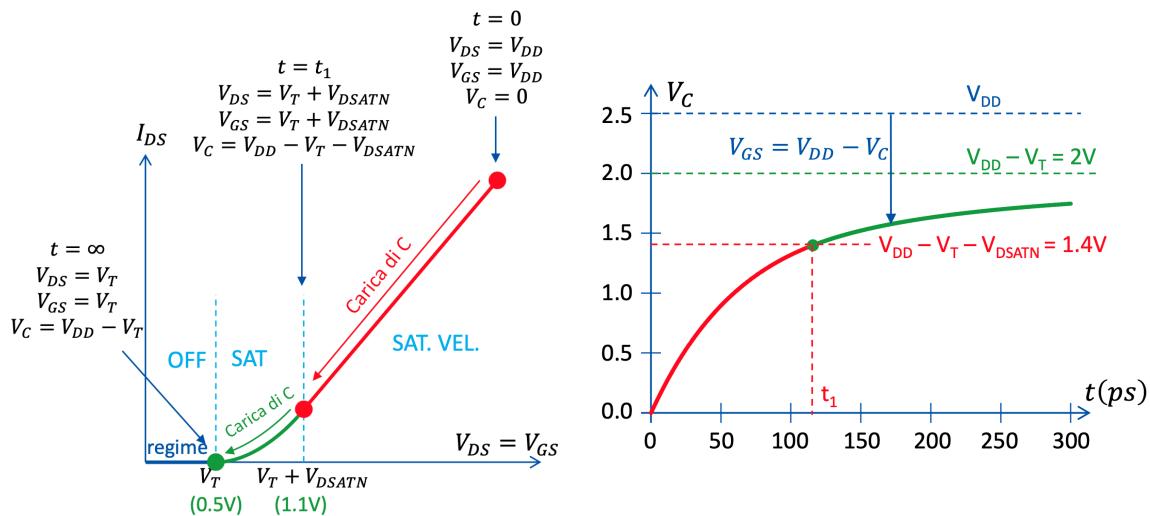
Si analizzano le variabili del circuito per $t < 0$, $t = 0$, $t > 0$ e $t \rightarrow \infty$:

- per $t < 0$: l'nmox è spento poiché $V_{GS}(t) = 0V < V_{TN}$;
- per $t = 0$: l'nmox si accende poiché $V_{GS}(t) = V_{DD} > V_{TN}$;
- per $t > 0$: l'nmox è acceso finché $V_{GS}(t) > V_{TN} \rightarrow V_C(t) < V_{DD} - V_{TN}$
- per $t \rightarrow \infty$: a regime l'nmox si spegne quando $V_{GS}(t) = V_{TN} \rightarrow V_C(t) = V_{DD} - V_{TN}$ lasciando il condensatore carico con tensione finale $V_C(t) = V_{DD} - V_{TN}$

Analisi grafica delle fasi del transitorio

Il pmos è connesso a diodo (gate e drain entrambi a V_{DD} per $t \geq 0$), quindi lavora solo in saturazione. Si osserva che la carica del condensatore avviene in due fasi (con un punto intermedio $t = t_1$):

1. per $0 < t < t_1$, $V_{GS} - V_{TN} > V_{DSATN}$ il mosfet lavora in regime di saturazione di velocità e la corrente di carica è costante pari a I_{DSATN} :
2. per $t = t_1$, $V_{GS} - V_{TN} = V_{DSATN}$ il mosfet passa dal regime di saturazione per velocità a quello di saturazione per pinchoff:
3. per $t > t_1$, $V_{GS} - V_{TN} < V_{DSATN}$ il mosfet lavora in saturazione per pinchoff e la corrente di scarica dipende da $V_{GS}(t) - V_{TN} = V_{DD} - V_C(t) - V_{TN}$, quindi anche dalla tensione del condensatore $V_C(t)$



Soluzione analitica del circuito

Si analizza la prima fase di carica del condensatore per $0 < t < t_1$ (saturazione di velocità):

$$I_C(t) = C \frac{dV_C}{dt} \quad I_{DSATN}(t) = k'_p Z_p V_{DSATN} \left(V_{DD} - V_C(t) - V_{TN} - \frac{V_{DSATN}}{2} \right)$$

$$I_C(t) = I_{DSATN}(t) \rightarrow V_C(t) = \left(V_{DD} - V_{TN} - \frac{V_{DSATN}}{2} \right) (1 - e^{-\alpha t}) \quad \text{con } \alpha = -\frac{k'_p Z_p V_{DSATN}}{C}$$

Si analizza il passaggio dalla prima alla seconda fase di per $t = t_1$:

$$V_{GS}(t_1) - V_{TN} = V_{DSATN} \rightarrow V_C(t_1) = V_{DD} - V_{TN} - V_{DSATN} \rightarrow t_1 = \frac{1}{\alpha} \ln \left(\frac{V_{DD} - V_{TN}}{V_{DSATN}} - 1 \right)$$

Si analizza la seconda fase di carica del condensatore per $t > t_1$ (saturazione per pinchoff):

$$I_C(t) = C \frac{dV_C}{dt} \quad I_{DS}(t) = \frac{k'_p Z_p}{2} (V_{GS}(t) - V_{TN})^2 = \frac{k'_p Z_p}{2} (V_{DD} - V_C(t) - V_{TN})^2$$

$$I_C(t) = I_{DS}(t) \rightarrow V_C(t) = V_{DD} - V_{TN} - \frac{2CV_{DSATN}}{k'_p Z_p V_{DSATN}(t - t_1) + 2C}$$

Valore di regime

Si analizza il valore di regime della tensione sul condensatore $V_C(t)$ per $t \rightarrow \infty$:

$$\lim_{t \rightarrow \infty} V_C(t) = V_{DD} - V_{TN}$$

Si ha, quindi, che il condensatore si carica fino a raggiungere la tensione finale $V_C(t) = V_{DD} - V_{TN}$.

Tempo di dimezzamento di $V_C(t)$

Si analizza il tempo di dimezzamento $t_{1/2}$, osservando che tale valore viene assunto durante la prima fase di carica, in cui il mosfet lavora in saturazione per velocità:

$$V_C(t_{1/2}) = \frac{V_{DD}}{2} \rightarrow t_{1/2} = \frac{1}{\alpha} \ln \left(\frac{2V_{DD} - 2V_{TN} - V_{DSATN}}{V_{DD} - 2V_{TN} - V_{DSATN}} \right)$$

6.9 Confronto dei transitori per carica e scarica con nmos e pmos

Dati numerici per il calcolo dei tempi caratteristici

Si scelgono i seguenti dati per il confronto dei transitori di carica e scarica con nmos e pmos:

circuito	nmos		pmos	
$C = 10fF$	$k'_n = 125\mu A/V^2$	$Z_n = 2$	$k'_p = 40\mu A/V^2$	$Z_p = 2$
$V_{DD} = 2.5V$	$V_{TN} = 0.5V$	$V_{DSATN} = 0.6V$	$V_{TP} = -0.5V$	$V_{DSATP} = -0.8V$

Confronto dei tempi caratteristici

circuito	t_1	$t_{1/2}$	$V_C(t \rightarrow \infty)$
scarica con nmos	74.5 ps	49 ps	0V
carica con nmos	115.6 ps	88.6 ps	$V_{DD} - V_{TN} < V_{DD}$
carica con pmos	151.7 ps	122 ps	V_{DD}
scarica con pmos	216.6 ps	237 ps	$-V_{TP} > 0V$

Si nota che:

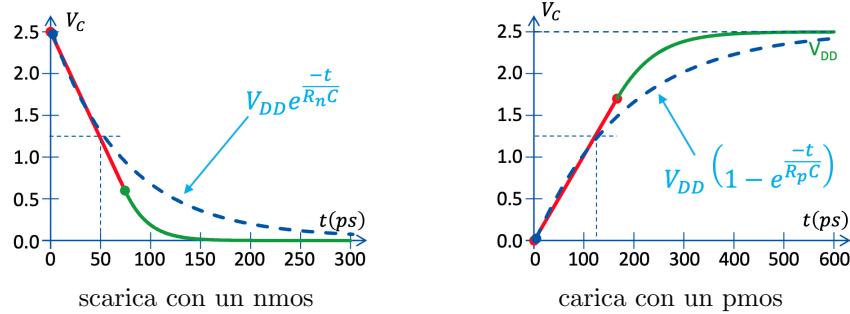
- l'nmos scarica totalmente il condensatore, ma lo carica parzialmente fino ad un valore inferiore a V_{DD} , si dice che **l'nmos trasmette bene il valore logico basso (0V)**;
- il pmos carica totalmente il condensatore, ma lo scarica parzialmente fino ad un valore superiore a 0V, si dice che **il pmos trasmette bene il valore logico alto (V_{DD})**;
- i tempi di dimezzamento per trasmettere il valore logico “cattivo” (carica con nmos e scarica con pmos) sono circa il doppio rispetto a quelli per trasmettere il valore logico “buono” (scarica con nmos e carica con pmos);
- il tempo di carica con pmos è circa il doppio rispetto al tempo di scarica con nmos, a parità del fattore di forma $Z_n = Z_p$, differenza dovuta alla minore mobilità delle lacune rispetto agli elettroni.

Effetto body nei circuiti di carica e scarica dei condensatori

Spesso, nei circuiti digitali, i body degli nmos sono collegati a massa e quelli dei pmos a V_{DD} , non necessariamente al loro source. Nell'nmos in scarica e nel pmos in carica ciò non comporta variazioni, mentre nell'nmos in carica e nel pmos in scarica si ha un aumento in modulo della soglia di tensione V_{TN} e V_{TP} rispettivamente, a causa dell'effetto body. Ciò comporta un aumento dei tempi di carica e scarica dei condensatori e un **peggioramento ulteriore nella trasmissione dei valori logici “cattivi”**.

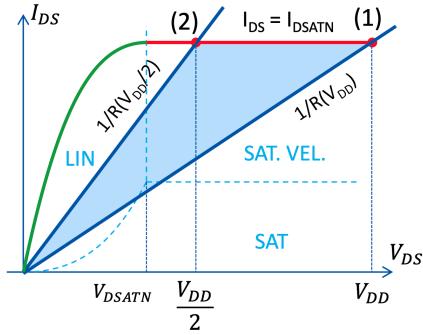
6.10 Resistenza equivalente del mosfet come interruttore

Per facilitare l'analisi dei circuiti con mosfet come interruttori, si può approssimare il comportamento del mosfet acceso con una resistenza equivalente R_{eq} . In questo modo si linearizza il comportamento non lineare del mosfet e si possono utilizzare le tecniche di analisi dei circuiti lineari. In particolare le curve di scarica e carica diventano esponenziali con costante di tempo $\tau = R_{eq}C$.



Definizione di resistenza equivalente R_{eq}

Si definisce la resistenza equivalente R_{eq} del mosfet come interruttore acceso come la media della resistenza istantanea agli estremi dell'intervallo di interesse, ovvero a $t = 0$ e a $t = t_{1/2}$, quando il mosfet trasmette il valore logico "buono".



$$\text{nmos: } R_n(0) = \frac{V_{DD}}{I_{DSATN}}, \quad R_n(t_{1/2}) = \frac{V_{DD}/2}{I_{DSATN}} \rightarrow$$

$$\rightarrow R_n = \frac{R_n(0) + R_n(t_{1/2})}{2} = \frac{3}{4} \frac{V_{DD}}{I_{DSATN}}$$

$$\text{pmos: } R_p(0) = \frac{V_{DD}}{I_{DSATP}}, \quad R_p(t_{1/2}) = \frac{V_{DD}/2}{I_{DSATP}} \rightarrow$$

$$\rightarrow R_p = \frac{R_p(0) + R_p(t_{1/2})}{2} = \frac{3}{4} \frac{V_{DD}}{I_{DSATP}}$$

NOTE:

- nelle elaborazioni delle reti logiche digitali, si considera solo la prima parte del transitorio, ovvero fino al tempo di dimezzamento $t_{1/2}$, in quanto è il momento in cui la tensione è vicina alla tensione di soglia logica, ovvero al valore in cui il segnale digitale cambia stato logico (da 0 a 1 o da 1 a 0);
- si calcola la resistenza solo nei casi in cui il mosfet trasmette il valore logico "buono", siccome è il comportamento più ricercato ed utilizzato nei circuiti digitali; più avanti viene anche approfondita la resistenza equivalente per valori logici "cattivi".

Fattore di forma e resistenza equivalente

La corrente di saturazione I_{DSAT} dipende dal fattore di forma Z del mosfet e può esser riscritta come:

$$I_{DSATN} = Z_n I_{DSATN0} \quad I_{DSATP} = Z_p I_{DSATP0}$$

Per cui la resistenza equivalente può essere riscritta in funzione delle costanti R_{n0} e R_{p0} , ovvero le resistenze equivalenti per fattore di forma unitario $Z_n = Z_p = 1$.

$$R_n = \frac{R_{n0}}{Z_n} \text{ con } R_{n0} = \frac{3}{4} \frac{V_{DD}}{I_{DSATN0}} \quad R_p = \frac{R_{p0}}{Z_p} \text{ con } R_{p0} = \frac{3}{4} \frac{V_{DD}}{I_{DSATP0}}$$

Si nota che la resistenza equivalente R_{eq} è inversamente proporzionale al fattore di forma Z del mosfet, per cui aumentando il fattore di forma Z del mosfet, si riduce la resistenza equivalente R_{eq} e quindi si riduce anche la costante di tempo $\tau = R_{eq}C$ del circuito, migliorando le prestazioni del circuito.

Analizzando i valori tipici delle tecnologie CMOS si osserva che:

$$k'_n \approx 3k'_p, \quad V_{TN} \approx -V_{TP}, \quad V_{DSATN} \approx -\frac{2}{3}V_{DSATP} \Rightarrow R_{p0} \approx 2R_{n0}$$

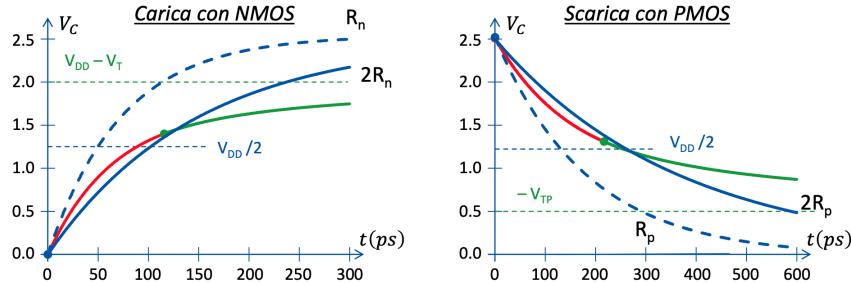
$$\frac{R_{p0}}{R_{n0}} = \frac{k'_n V_{DSATN} (V_{DD} - V_{TN} - \frac{V_{DSATN}}{2})}{k'_p V_{DSATP} (-V_{DD} - V_{TP} - \frac{V_{DSATP}}{2})} \approx \frac{k'_n V_{DSATN}}{-k'_p V_{DSATP}} \approx 3 \cdot \frac{2}{3} = 2$$

Resistenze equivalenti per valori logici “cattivi”

Si osserva che per approssimare al meglio il transitorio dei mosfet nella trasmissione di valori logici “cattivi”, è necessario raddoppiare le resistenze equivalenti calcolate in precedenza:

nmos (carica): $R_{n,cattivo} = 2R_{n,buono}$

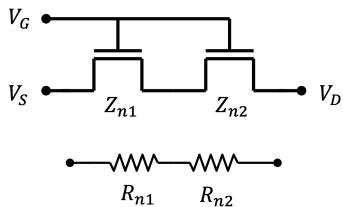
pmos (scarica): $R_{p,cattivo} = 2R_{p,buono}$



6.11 Reti di mosfet e resistenza equivalente complessiva

Serie di mosfet

La serie di mosfet equivale ad un mosfet equivalente con resistenza equivalente pari alla somma delle resistenze equivalenti dei singoli mosfet e con fattore di forma pari al reciproco della somma dei reciproci dei fattori di forma dei singoli mosfet:

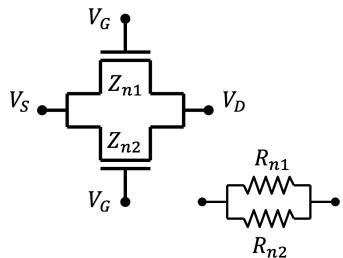


$$R_{eq} = R_1 + R_2$$

$$\frac{1}{Z_{eq}} = \frac{1}{Z_1} + \frac{1}{Z_2} \rightarrow Z_{eq} = \frac{Z_1 Z_2}{Z_1 + Z_2}$$

Parallelo di mosfet

La parallelo di mosfet equivale ad un mosfet equivalente con resistenza equivalente pari al reciproco della somma dei reciproci delle resistenze equivalenti dei singoli mosfet e con fattore di forma pari alla somma dei fattori di forma dei singoli mosfet:



$$\frac{1}{R_{eq}} = \frac{1}{R_1} + \frac{1}{R_2} \rightarrow R_{eq} = \frac{R_1 R_2}{R_1 + R_2}$$

$$Z_{eq} = Z_1 + Z_2$$

7 Breve cenno ai segnali digitali

7.1 Segnali binari logici ed elettrici

- **segnale digitale:** segnale ad ampiezza discreta e a tempo discreto. Viene detto anche segnale numerico perché tempo e ampiezza possono essere rappresentati con numeri
- **segnale binario:** segnale digitale che può assumere solo due valori distinti, uno il negato dell'altro. In base al dominio può essere rappresentato in modi diversi:
 - nel **dominio logico** è rappresentato con una variabile booleana che può assumere i valori logici *True* (1) e *False* (0).
 - nel **dominio elettrico** è rappresentato con due livelli di tensione distinti, uno alto V_H (HIGH) e uno basso V_L (LOW), che corrispondono al potenziale più alto e più basso del circuito.

7.2 Circuiti e porte logiche

Un **circuito logico** è un componente elettronico che elabora segnali digitali binari ed è formato da più transistor. È costituito da uno o più ingressi e da una o più uscite che possono assumere i due valori di tensione V_H e V_L . In particolare il valore dell'uscita è esprimibile come funzione logica degli ingressi.

Esistono due famiglie di circuiti logici:

- **circuiti combinatori:** l'uscita dipende solo dagli ingressi in un dato istante di tempo
- **circuiti sequenziali:** l'uscita dipende dagli ingressi in un dato istante di tempo e anche dall'uscita dell'istante precedente, vengono anche detti circuiti con memoria.

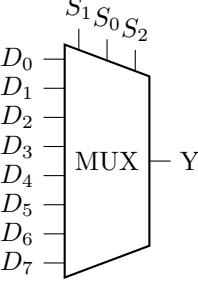
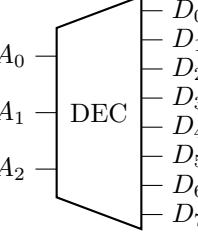
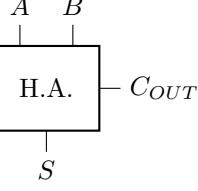
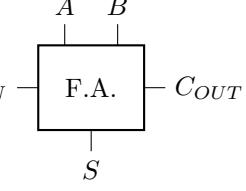
Per semplificare la notazione si assumono le seguenti convenzioni sui livelli di tensione:

- il livello logico alto è rappresentato con $V_H = V_{DD} > 0$, ovvero il potenziale maggiore del circuito;
- il livello logico basso è rappresentato con $V_L = 0V$, ovvero il potenziale minore del circuito;
- si nominano i nodi delle reti logiche con variabili booleane A, B, C, \dots per indicare che i valori di potenziale che possono assumere sono $V_L = 0V$ o $V_H = V_{DD}$.

Porte logiche e circuiti logici fondamentali

Una **porta logica** è un circuito logico combinatorio che implementa una specifica funzione logica booleana. Di seguito sono riportate le porte logiche e i circuiti logici fondamentali.

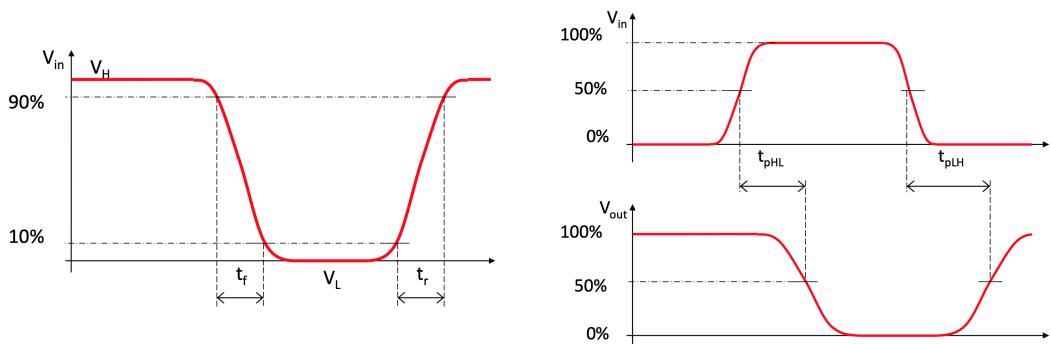
circuito	Funzione identità	Porta AND	Porta OR	Porta XOR
simbolo				
funzione	$Y = A$	$Y = A \cdot B$	$Y = A + B$	$Y = A \oplus B$
tab. di verità	$\begin{array}{c c} A & Y \\ \hline 0 & 0 \\ 1 & 1 \end{array}$	$\begin{array}{c c c} & A & B & Y \\ \hline 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{array}$	$\begin{array}{c c c} & A & B & Y \\ \hline 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{array}$	$\begin{array}{c c c} & A & B & Y \\ \hline 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{array}$
circuito	Invertitore NOT	Porta NAND	Porta NOR	Porta XNOR
simbolo				
funzione	$Y = \bar{A}$	$Y = \overline{A \cdot B}$	$Y = \overline{A + B}$	$Y = \overline{A \oplus B}$
tab. di verità	$\begin{array}{c c} A & Y \\ \hline 0 & 1 \\ 1 & 0 \end{array}$	$\begin{array}{c c c} & A & B & Y \\ \hline 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{array}$	$\begin{array}{c c c} & A & B & Y \\ \hline 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{array}$	$\begin{array}{c c c} & A & B & Y \\ \hline 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{array}$

circuito	simbolo	descrizione
Multiplexer (MUX)		Dispositivo che seleziona quale tra 2^n segnali di ingresso $D_0, D_1, \dots, D_{2^n-1}$ trasmettere alla porta di uscita Y , tramite n segnali di controllo S_0, S_1, \dots, S_{n-1} .
Decoder (DEC)		Dispositivo che decodifica un segnale di ingresso A di n bit in 2^n segnali di uscita $D_0, D_1, \dots, D_{2^n-1}$, attivando solo l'uscita corrispondente al valore binario di A .
Half Adder (HA)		Circuito che somma due bit A e B , producendo una somma S e una riporto C : $C_{OUT} = A \cdot B \quad S = A \oplus B$
Full Adder (FA)		Circuito che somma due bit A e B con un riporto in ingresso C_{IN} , producendo una somma S e un riporto in uscita C_{OUT} : $C_{OUT} = A \cdot B + C_{IN} \cdot (A + B) \quad S = A \oplus B \oplus C_{IN}$

7.3 Parametri dinamici dei circuiti logici

Per misurare le prestazioni di un circuito logico si usano i seguenti parametri dinamici:

- **tempo di commutazione**: tempo impiegato dall'uscita di un circuito logico per cambiare stato (grafico a sinistra). Si distinguono due tempi di commutazione:
 - t_f : tempo di discesa, dal 90% (alto) al 10% (basso) dell'escursione totale
 - t_r : tempo di salita, dal 10% (basso) al 90% (alto) dell'escursione totale
- **tempo di propagazione t_p** : tempo medio di propagazione del segnale dall'ingresso all'uscita del circuito logico (grafico a destra). Si distinguono due tempi di propagazione:
 - t_{PLH} : tempo di propagazione con l'uscita che passa da livello logico basso a livello logico alto
 - t_{PHL} : tempo di propagazione con l'uscita che passa da livello logico alto a livello logico basso

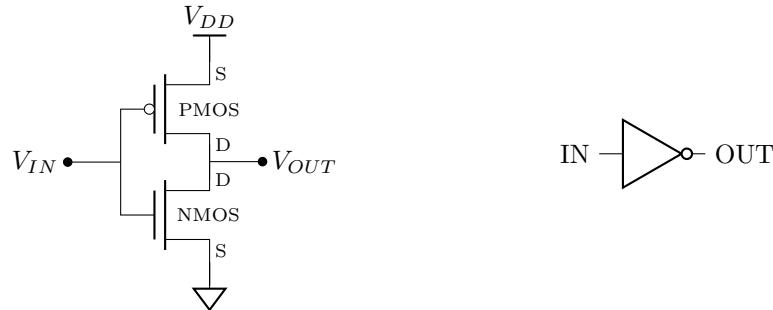


8 Invertitore CMOS

8.1 Schema circuitale

Schema circuitale e simbolo logico

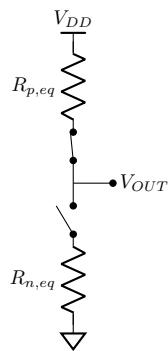
L'invertitore CMOS (detto anche Complementary-MOS) è il circuito logico più semplice realizzabile con la tecnologia cmos. Implementa la funzione logica NOT, ovvero l'operazione di negazione booleana. Lo schema circuitale dell'invertitore cmos è riportato in figura (a sinistra), insieme al simbolo logico corrispondente (a destra).



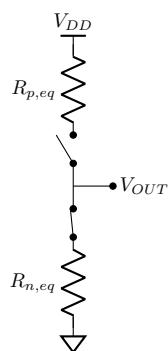
L'invertitore cmos è costituito da un transistor nmos e un transistor pmos collegati in serie tra V_{DD} e massa. L'ingresso del circuito è collegato ai gate di entrambi i transistor, mentre l'uscita è prelevata dal nodo di connessione tra i due transistor.

Funzionamento

Per comprendere meglio il funzionamento dell'invertitore cmos si sostituiscono i transistor (usati come interruttori) con la serie di interruttore e resistenza equivalente, come mostrato in figura. Si distinguono due casi principali in base al valore logico dell'ingresso V_{IN} :



Quando $V_{IN} = 0V$, il pmos è acceso, mentre l'nmos è spento. L'uscita V_{OUT} è, quindi, collegata a V_{DD} tramite la resistenza equivalente R_p del pmos e si ha $V_{OUT} = V_{DD}$. Il pmos porta l'uscita al livello logico alto (valore logico “buono” per pmos) ed è chiamato transistor di pull-up.



Quando $V_{IN} = V_{DD}$, il pmos è spento, mentre l'nmos è acceso. L'uscita V_{OUT} è, quindi, collegata a massa tramite la resistenza equivalente R_n dell'nmos e si ha $V_{OUT} = 0$. L'nmos porta l'uscita al livello logico basso (valore logico “buono” per nmos) ed è chiamato transistor di pull-down.

In entrambi i casi l'uscita V_{OUT} assume il valore logico opposto rispetto all'ingresso V_{IN} , come previsto dalla funzione logica NOT.

8.2 Caratteristica di trasferimento ingresso-uscita

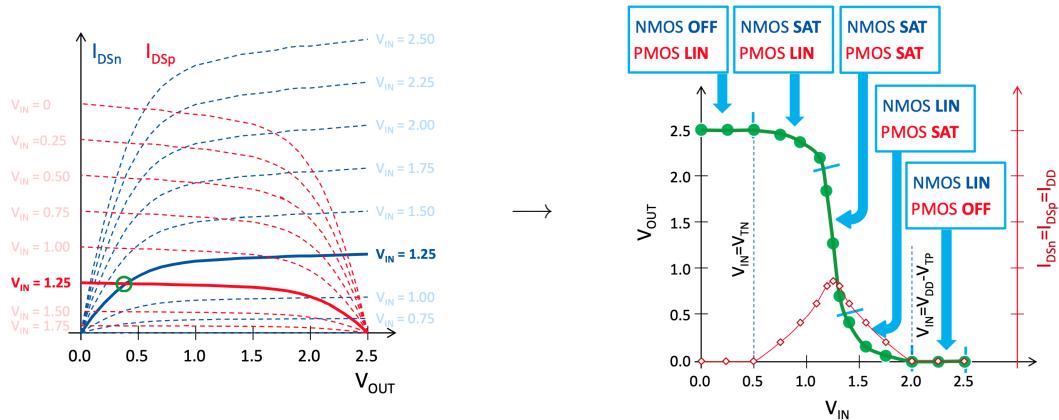
La caratteristica statica di trasferimento, o caratteristica di trasferimento ingresso-uscita (VTC, voltage transfer characteristic) è la curva che descrive il valore di V_{OUT} in funzione di V_{IN} .

Calcolo grafico della VTC

Per ottenerla, si uguagliano le correnti attraverso i due transistor secondo la legge di Kirchhoff delle correnti (LKC).

$$I_{DS,n}(V_{IN}, V_{OUT}) = I_{DS,p}(V_{IN}, V_{OUT})$$

Per evitare di fare conti, si risolve l'equazione graficamente, ovvero si tracciano le curve delle due correnti in funzione di V_{OUT} per diversi valori di V_{IN} . I punti di intersezione delle due curve rappresentano le soluzioni dell'equazione, ovvero i valori di V_{OUT} corrispondenti a ciascun valore di V_{IN} .



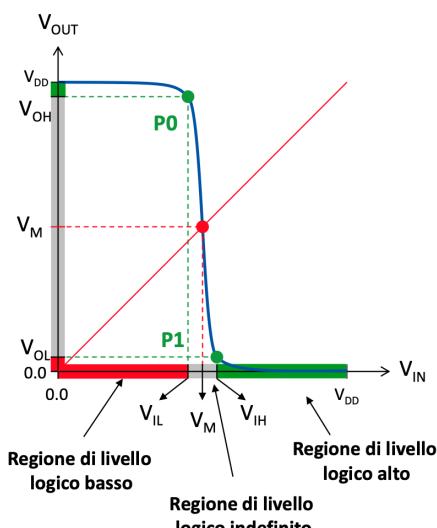
Regioni di funzionamento dei transistor

Si osserva che per $V_{IN} = 0V$ o $V_{IN} = V_{DD}$ un solo transistor è acceso in regime lineare, mentre l'altro è spento. Per valori intermedi si ha che:

- il pmos passa da lineare a saturazione e infine si spegne al crescere di V_{IN}
- l'nmos da interdizione, passa da saturazione e infine lineare al crescere di V_{IN}

Punti importanti

La pendenza della VTC (ovvero la derivata di V_{OUT} in funzione di V_{IN}) corrisponde al guadagno in tensione dell'invertitore. Si definiscono i seguenti punti importanti della VTC:



P_0 e P_1 corrispondono ai punti con guadagno unitario. Tali punti delimitano la regione di livello logico indefinito, con guadagno > 1 e le regioni di livello logico definito (a sinistra e a destra), con guadagno < 1 .

Si osserva che in corrispondenza delle regioni di livello logico definito la pendenza della curva è molto bassa (guadagno < 1), per cui piccole variazioni di V_{IN} producono variazioni trascurabili di V_{OUT} (proprietà di rigenerazione del segnale).

Il pallino rosso in cui $V_{OUT} = V_{IN}$ è il punto di soglia di commutazione logica dell'invertitore. La tensione di ingresso associata a questo punto è detta soglia di commutazione logica V_M .

8.3 Soglia di commutazione logica

Calcolo della tensione di commutazione logica

Per calcolare la soglia di commutazione logica V_M si impone la condizione $V_{OUT} = V_{IN} = V_M$, ovvero si cortocircuitano l'ingresso e l'uscita dell'invertitore. Si ottiene, quindi, la seguente equazione che dipende solo da V_M :

$$I_{DS,n}(V_M) = I_{DS,p}(V_M)$$

Si ottiene V_M risolvendo l'equazione:

$$V_M = \frac{V_{TN} + \frac{V_{DSATN}}{2} + r \left(V_{DD} + V_{TP} + \frac{V_{DSATP}}{2} \right)}{1 + r} \quad \text{con } r = -\frac{Z_p}{Z_n} \frac{k'_p V_{DSATP}}{k'_n V_{DSATN}}$$

Relazione tra tensione di commutazione logica e rapporto dei fattori di forma

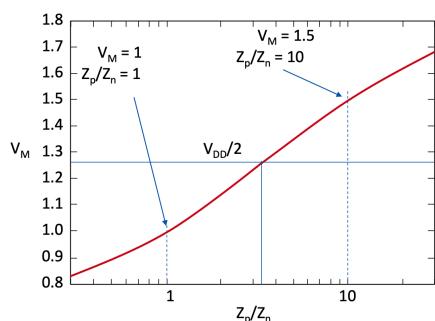
Si osserva che V_M oltre a dipendere dai parametri fisici dei transistor e dal valore di V_{DD} , dipende anche dal rapporto tra i fattori di forma dei due transistor Z_p/Z_n . Modificando tale rapporto è possibile regolare il valore di V_M . In particolare per avere $V_M = V_{DD}/2$ si ottiene:

$$V_M = \frac{V_{DD}}{2} \rightarrow r = \frac{\frac{V_{DD}}{2} - V_{TN} - \frac{V_{DSATN}}{2}}{\frac{V_{DD}}{2} + V_{TP} + \frac{V_{DSATP}}{2}} \rightarrow \frac{Z_p}{Z_n} = -r \frac{k'_n V_{DSATN}}{k'_p V_{DSATP}} \approx 3 - 3.5$$

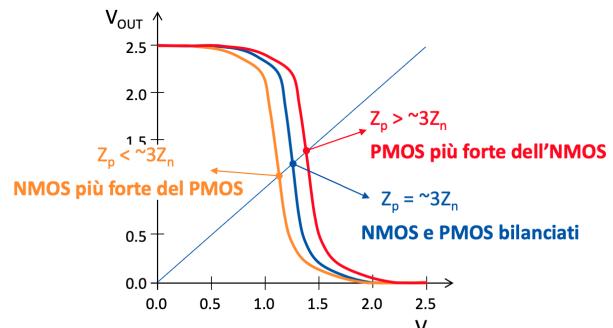
Analizzando graficamente come varia V_M al variare del rapporto Z_p/Z_n , si osserva che:

- per $Z_p > 3Z_n$ aumenta V_M , prevale il pmos e la VTC si sposta verso destra (il valore logico alto si mantiene più a lungo)
- per $Z_p < 3Z_n$ diminuisce V_M , prevale l'nmos e la VTC si sposta verso sinistra (il valore logico basso si mantiene più a lungo)
- per $Z_p = 3Z_n$ si ha $V_M \approx V_{DD}/2$ e i due transistor si bilanciano e la VTC è simmetrica

Infine si nota che la curva è molto piatta in corrispondenza della soglia di commutazione logica, il che implica che V_M è poco sensibile alle variazioni dei fattori di forma dei transistor.



V_M al variare di Z_p/Z_n , in scala logaritmica



VTC al variare di Z_p/Z_n

8.4 Tolleranza al rumore

Tipi di disturbi in un circuito elettrico

In un circuito elettrico possono essere presenti diversi tipi di disturbi che possono alterare il corretto funzionamento del circuito stesso. I principali tipi di disturbi sono:

- **accoppiamenti induttivi**: disturbi causati dalla mutua induzione di due conduttori vicini dovuta all'induttanza parassita delle piste di collegamento
- **accoppiamenti capacitivi**: disturbi causati dalla capacità parassita tra due conduttori vicini
- **rumore dell'alimentazione**: variazioni indesiderate della tensione di alimentazione del circuito dovuta ad esempio dalla caduta di tensione per le resistenze lungo le linee di alimentazione

Margine di immunità al rumore

Il margine di immunità al rumore o NM (noise margin) rappresenta l'ampiezza massima del disturbo che è possibile avere durante la trasmissione del segnale tra due porte logiche, senza che si verifichino errori di interpretazione del segnale logico. In base al livello logico si distinguono due margini di immunità:

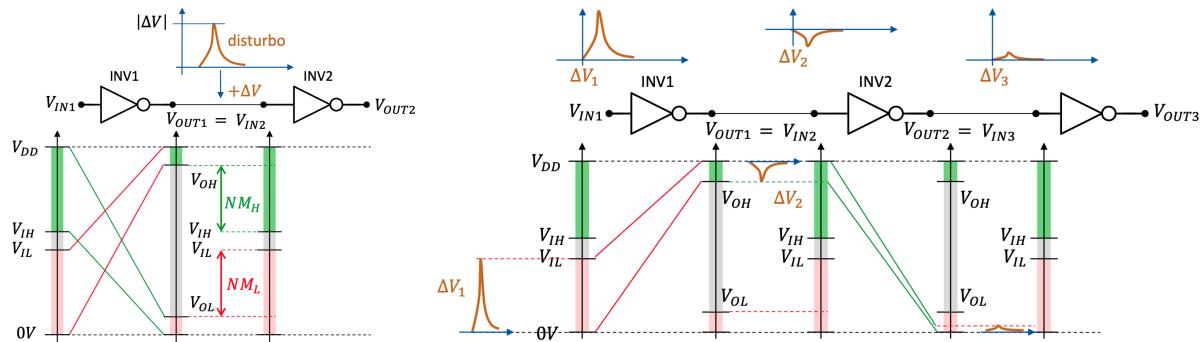
- NM per il livello logico alto: $NM_H = V_{OH} - V_{IH}$ ovvero la differenza tra la minima tensione di uscita di un segnale a livello logico alto e la minima tensione di ingresso per riconoscere un segnale come livello logico alto;
- NM per il livello logico basso: $NM_L = V_{IL} - V_{OL}$ ovvero la differenza tra la massima tensione di ingresso per riconoscere un segnale come livello logico basso e la massima tensione di uscita di un segnale a livello logico basso.

Il margine di immunità complessivo, ovvero il massimo disturbo tollerabile, corrisponde al margine di immunità più piccolo tra i due livelli logici $NM = \min(NM_H, NM_L)$.

Rigenerazione del segnale

La rigenerazione del segnale è la capacità di un circuito logico di ripristinare i livelli logici di uscita a valori vicini ai valori ideali V_{OH} e V_{OL} , attenuando gli effetti dei disturbi e del rumore sul segnale.

Affinché una porta logica sia in grado di rigenerare il segnale, è necessario che il guadagno della VTC sia minore di 1 nelle regioni di livello logico definito (VTC tendente all'orizzontale) e maggiore di 1 nella regione di livello logico indefinito (VTC tendente al verticale).



Ruolo della tensione di commutazione logica nella rigenerazione del segnale

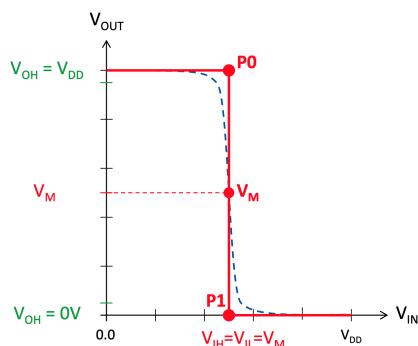
I margini di immunità al rumore per un invertitore cmos possono essere approssimati come:

$$NM_H \approx V_{DD} - V_M \quad NM_L \approx V_M \quad NM \approx \min(NM_H, NM_L)$$

Si osserva che per massimizzare il margine di immunità complessivo NM è necessario bilanciare i due margini di immunità, ovvero impostare la soglia di commutazione logica V_M a metà della tensione di alimentazione $V_{DD}/2$. Si ottiene così che il dimensionamento ottimo dei transistor per massimizzare la tolleranza al rumore è $Z_p \approx 3Z_n$.

8.5 Invertitore ideale

Un invertitore ideale è un invertitore logico che interpreta $V_{IN} < V_M$ come livello logico basso e $V_{IN} > V_M$ come livello logico alto, inoltre rigenera l'uscita esattamente ai valori ideali $V_{OH} = V_{DD}$ e $V_{OL} = 0V$.



La curva VTI di un invertitore ideale è una curva a gradino che passa per il punto di soglia di commutazione logica e la funzione di trasferimento è una funzione definita a tratti:

$$V_{OUT} = \begin{cases} V_{DD} & \text{se } V_{IN} < V_M \\ V_M & \text{se } V_{IN} = V_M \\ 0V & \text{se } V_{IN} > V_M \end{cases}$$

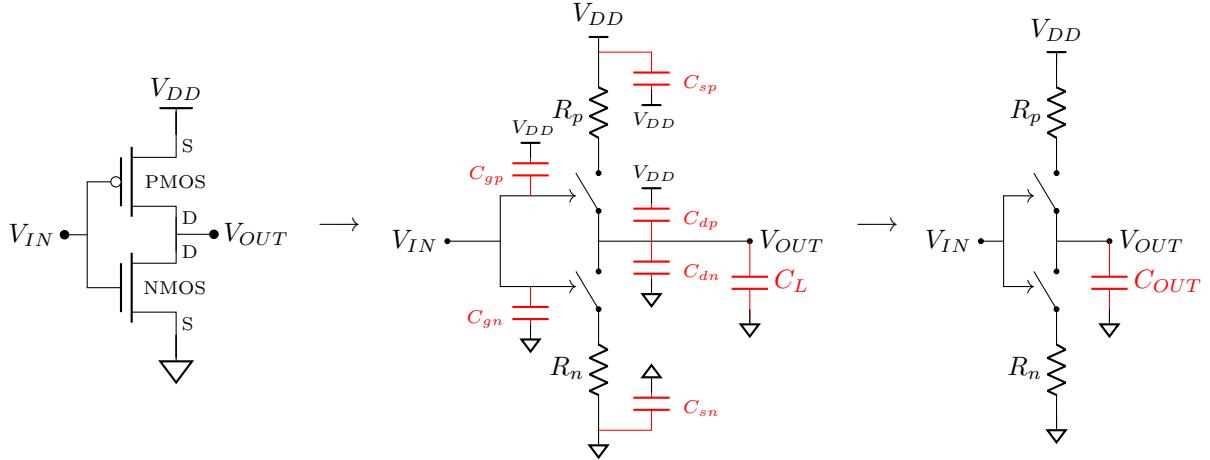
I margini di immunità al rumore diventano esattamente

$$NM_H = V_{DD} - V_M \quad NM_L = V_M$$

8.6 Tempo di ritardo

Modello RC dell'invertitore

Ad ogni terminale dei transistor sono associate delle capacità parassite. Queste capacità vanno caricate e scaricate durante la commutazione del segnale, causando un ritardo temporale tra l'ingresso e l'uscita dell'invertitore. Si rappresenta, quindi, l'invertitore con un modello RC equivalente in cui le capacità parassite sono evidenziate in rosso ed è stata aggiunta la capacità di carico C_L collegata all'uscita del circuito:



Analizzando le resistenze e le capacità del modello RC equivalente per i nodi di ingresso ed uscita si ha:

$$R_{IN} = \infty \quad C_{IN} = C_{gn} + C_{dp} \quad R_{OUT} = \frac{R_n + R_p}{2} \quad C_{OUT} = C_{gp} + C_{dn} + C_L$$

NOTA: le capacità dei source sono cortocircuitate per cui sono ininfluenti, inoltre le capacità del pmos possono essere considerate collegate tutte a massa invece che a V_{DD} , come visto alla fine del paragrafo delle capacità parassite dei mosfet.

Calcolo dei tempi di ritardo intrinseci

Il tempo di ritardo è il tempo che impiega la porta di uscita a raggiungere il 50% del valore finale dopo un cambio di stato dell'ingresso. Il tempo di ritardo si dice intrinseco se il carico esterno è nullo ($C_L = 0$). Si definiscono due tempi di ritardo intrinseci:

- tempo di ritardo di salita t_{pLH0} : tempo che impiega l'uscita a salire dal 50% del livello logico basso al 50% del livello logico alto dopo un fronte di salita dell'ingresso

$$V_{OUT} = \frac{V_{DD}}{2} \rightarrow V_{DD} - V_{DD} e^{-\frac{t_{pLH0}}{R_p C_{OUT}}} = \frac{V_{DD}}{2} \rightarrow t_{pLH0} = \ln(2) R_p C_{OUT} \approx 0.69 R_p C_{OUT}$$

- tempo di ritardo di discesa t_{pHL0} : tempo che impiega l'uscita a scendere dal 50% del livello logico alto al 50% del livello logico basso dopo un fronte di discesa dell'ingresso

$$V_{OUT} = \frac{V_{DD}}{2} \rightarrow V_{DD} e^{-\frac{t_{pHL0}}{R_n C_{OUT}}} = \frac{V_{DD}}{2} \rightarrow t_{pHL0} = \ln(2) R_n C_{OUT} \approx 0.69 R_n C_{OUT}$$

Si calcola il tempo di propagazione intrinseco medio come:

$$t_{p0} = \frac{t_{pLH0} + t_{pHL0}}{2} = \ln(2) \frac{R_n + R_p}{2} C_{OUT} \approx 0.69 \frac{R_n + R_p}{2} C_{OUT} = 0.69 R_{OUT} C_{OUT}$$

Carico esterno e fanout

Quando l'invertitore guida un carico esterno C_L , i tempi di ritardo aumentano in quanto la capacità totale da caricare o scaricare è maggiore. Si definiscono il fanout f e il coefficiente di carico γ :

$$f = \frac{C_L}{C_{IN}} \quad \gamma = \frac{C_{OUT}}{C_{IN}}$$

Si ottengono, quindi, i tempi di ritardo con carico esterno:

$$t_{pLH} = 0.69 R_p (C_{OUT} + C_L) = 0.69 R_p C_{OUT} + 0.69 R_p C_L = t_{pLH0} + 0.69 R_p C_L$$

$$t_{pHL} = 0.69 R_n (C_{OUT} + C_L) = 0.69 R_n C_{OUT} + 0.69 R_n C_L = t_{pHL0} + 0.69 R_n C_L$$

$$t_p = \frac{t_{pLH} + t_{pHL}}{2} = 0.69 R_{OUT} (C_{OUT} + C_L) = t_{p0} + 0.69 R_{OUT} C_L = t_{p0} \left(1 + \frac{f}{\gamma} \right)$$

Ottimizzazione del tempo di ritardo intrinseco

Si osserva che il tempo di ritardo dipende da una serie di parametri tecnologici (specifici dei materiali) e da altri parametri di progettazione che possono essere scelti liberamente. In particolare si può agire su:

- il rapporto tra i fattori di forma dei transistor Z_p/Z_n
- il fattore di carico f che dipende dall'architettura del circuito logico
- la tensione di alimentazione V_{DD} , difficile da modificare in quanto spesso imposta da vincoli esterni

Il tempo di ritardo può essere ottimizzato secondo i seguenti criteri:

- ottimizzare il tempo massimo di ritardo $\max(t_{pHL}, t_{pLH})$
- ottimizzare il tempo di ritardo medio t_p

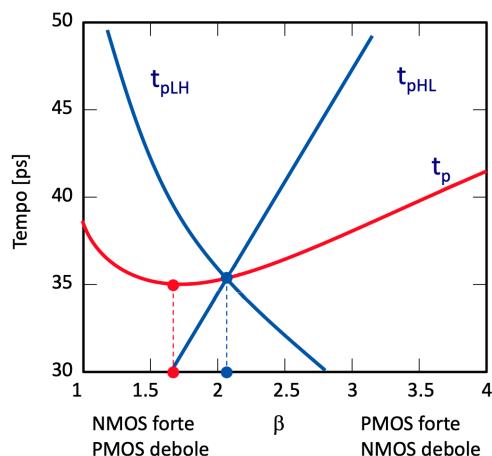
Si definiscono quindi i coefficienti β e ρ e si procede all'ottimizzazione scegliendo il valore ottimale di β in base al criterio scelto:

$$\beta = \frac{Z_p}{Z_n} = \frac{W_p}{W_n} \quad \rho = \frac{R_{p0}}{R_{n0}}$$

$$t_{pHL0} = 0.69 R_n C_{OUT} = 0.69 \frac{R_{n0}}{Z_n} C_{d0} W_n (1 + \beta) = 0.69 R_{n0} C_{d0} L (1 + \beta)$$

$$t_{pLH0} = 0.69 R_p C_{OUT} = 0.69 \frac{\rho R_{n0}}{\beta Z_n} C_{d0} W_n (1 + \beta) = 0.69 R_{n0} C_{d0} L \left(\frac{\rho}{\beta} + \rho \right)$$

$$t_{p0} = 0.69 \frac{R_{n0} C_{d0} L}{2} \left(1 + \beta + \rho + \frac{\rho}{\beta} \right)$$



Ottimizzando il tempo massimo di ritardo (ovvero imponendo $t_{pHL0} = t_{pLH0}$, punto blu in figura):

$$\begin{aligned} t_{pHL0} = t_{pLH0} &\rightarrow 1 + \beta = \frac{\rho}{\beta} + \rho \\ &\rightarrow \beta = \rho \approx 2 - 2.5 \end{aligned}$$

Ottimizzando il tempo medio di ritardo (ovvero trovando il minimo di t_{p0} , punto rosso in figura):

$$\begin{aligned} \frac{dt_{p0}}{d\beta} = 0 &\rightarrow 1 - \frac{\rho}{\beta^2} = 0 \\ &\rightarrow \beta = \sqrt{\rho} \approx 1.4 - 1.6 \end{aligned}$$

Si nota che il valore di β ottimizza anche l'affidabilità del circuito in quanto agisce sulla soglia di commutazione logica V_M e quindi sui margini di immunità al rumore. Per avere affidabilità massima (ovvero $V_M = V_{DD}/2$) bisogna scegliere $\beta \approx 3$, ma siccome V_M reagisce molto poco alle variazioni di β , si predilige scegliere β in modo da ottimizzare il tempo di ritardo.

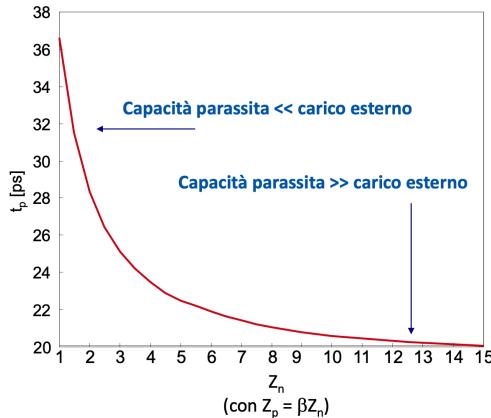
Dimensionamento dell'invertitore con carico esterno

Fissato un certo β si calcola come ottimizzare il tempo di ritardo in funzione della capacità di carico esterna C_L andando a dimensionare opportunamente i transistor. Si ottiene:

$$t_p = 0.69 R_{OUT} C_{OUT} \left(1 + \frac{C_L}{C_{OUT}} \right) = t_{p0} \left(1 + \frac{C_L}{C_{OUT}} \right) \quad \text{con } t_{p0} \text{ indipendente da } C_L$$

$$1 + \frac{C_L}{C_{OUT}} = 1 + \frac{C_L}{C_{d0} W_n (1 + \beta)} \approx 1 + C_L \frac{\text{costante}}{W_n} \quad R_{OUT} = \frac{1}{2} \frac{R_{n0}}{Z_n} \left(1 + \frac{\rho}{\beta} \right) \approx \frac{\text{costante}}{Z_n}$$

Si osserva che per minimizzare il tempo di ritardo si può aumentare la larghezza W_n (e proporzionalmente anche $W_p = \beta W_n$ e $Z_n = \beta Z_p$) in modo da ridurre sia R_{OUT} che il termine $\left(1 + \frac{C_L}{C_{OUT}} \right)$. Facendo così, però, si aumenta l'area occupata dal circuito e la capacità parassita di uscita C_{OUT} .



Osservando il grafico di t_p in funzione di $Z_n = W_n/L$ si osserva che:

- diminuendo Z_n il tempo di ritardo aumenta a causa dell'aumento di R_{OUT} che limita la corrente di carica/scarica di $C_{OUT} + C_L$
- aumentando Z_n il tempo di ritardo diminuisce inizialmente a causa della diminuzione di R_{OUT} , poi si stabilizza attorno ad un valore minimo dovuto al fatto che l'invertitore deve anche "auto-caricarsi" la capacità parassita C_{OUT} che è ormai diventata significativa rispetto al carico C_L

8.7 Consumo statico

Il consumo statico è il consumo di potenza dell'invertitore quando l'ingresso è mantenuto costante ad un livello logico definito (alto o basso). In questo caso uno dei due transistor è sempre spento per cui non c'è corrente di drenaggio tra i due terminali di alimentazione V_{DD} e massa. L'unico contributo al consumo statico è dovuto alle correnti di sottosoglia dei due transistor, alle correnti di perdita dell'ossido di gate e alla corrente inversa del diodo drain-substrato. Questi contributi sono però trascurabili, per cui si può considerare che il consumo statico di un invertitore cmos sia praticamente nullo.

8.8 Consumo dinamico

Analisi del consumo dinamico per una porta logica generale

Il consumo dinamico è il consumo di potenza di una generica porta logica durante la commutazione del segnale di uscita. La potenza assorbita dalla porta logica viene utilizzata per la carica e scarica delle capacità parassite e del carico esterno. Si analizza il consumo dinamico per entrambe le fasi di commutazione dell'uscita (LH e HL):

- commutazione dell'uscita LH:

$$\begin{aligned} \text{energia assorbita} \quad E_{V_{DD}} &= \int_0^T V_{DD} i_{DD}(t) dt = V_{DD} \int_{V_{OL}}^{V_{OH}} C \cdot dV = CV_{DD}(V_{OH} - V_{OL}) \\ \text{energia immagazzinata} \quad E_C &= \int_{V_{OL}}^{V_{OH}} V \cdot C \cdot dV = \frac{1}{2}CV_{OH}^2 - \frac{1}{2}CV_{OL}^2 \\ \text{energia dissipata} \quad E_{diss} &= E_{V_{DD}} - E_C \end{aligned}$$

- commutazione dell'uscita HL:

$$\begin{aligned} \text{energia assorbita} \quad E_{V_{DD}} &= 0 \quad \text{il pmos è spento e non viene assorbita corrente da } V_{DD} \\ \text{energia immagazzinata} \quad E_C &= \int_{V_{OH}}^{V_{OL}} V \cdot C \cdot dV = \frac{1}{2}CV_{OL}^2 - \frac{1}{2}CV_{OH}^2 \\ \text{energia dissipata} \quad E_{diss} &= -E_C \end{aligned}$$

Complessivamente si ottiene che l'energia totale assorbita dal generatore V_{DD} durante un ciclo di commutazione completo (LH + HL) è pari all'energia dissipata e vale:

$$E_{V_{DD},tot} = E_{diss,tot} = C \cdot V_{DD}(V_{OH} - V_{OL})$$

Definita la frequenza di commutazione f (numero di cicli di commutazione al secondo), si ottiene la potenza dinamica dissipata dalla porta logica:

$$P_{DYN} = E_{diss,tot} \cdot f = C \cdot V_{DD}(V_{OH} - V_{OL}) \cdot f$$

Consumo dinamico dell'invertitore

Applicando la formula generale del consumo dinamico all'invertitore cmos si ottiene:

$$P_{DYN,invertitore} = C_{OUT} \cdot V_{DD}^2 \cdot f$$

Cammino diretto

Si osserva che durante la commutazione del segnale di uscita, per un breve intervallo di tempo, entrambi i transistor possono essere contemporaneamente accesi, creando un cammino diretto tra V_{DD} e massa. Ciò si verifica quando la tensione di ingresso attiva entrambi i mosfet per legge $V_{TN} < V_{IN} < V_{DD} - V_{TP}$ e il tempo di durata del cammino diretto si indica con t_{cc} . In questo intervallo di tempo si ha una corrente di cortocircuito I_{CC} che causa una dissipazione di potenza addizionale e sottrae corrente alla carica/scarica delle capacità parassite e del carico esterno, allungando i tempi di ritardo. Per questo motivo si vuole minimizzare la durata del cammino diretto durante la progettazione dell'invertitore.

8.9 Oscillatore ad anello

L'oscillatore ad anello è un circuito costituito da un numero dispari di invertitori collegati in cascata, in cui l'uscita dell'ultimo invertitore è collegata all'ingresso del primo. In questo modo si crea un circuito ad anello chiuso che genera un segnale che commuta periodicamente tra i livelli logici alto e basso ad una data frequenza.

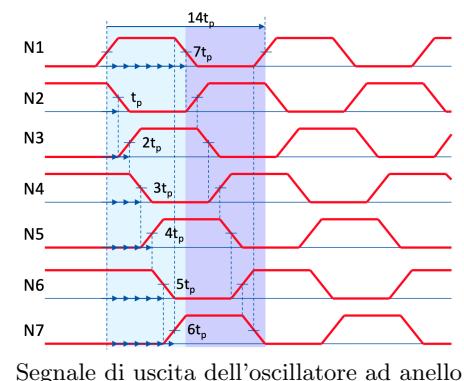
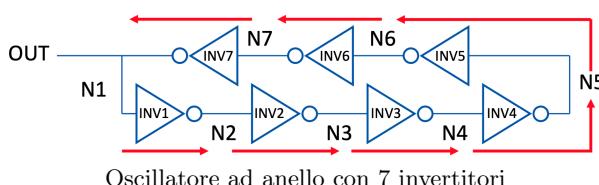
NOTA1: per il corretto funzionamento dell'oscillatore è necessario che il numero di invertitori sia dispari e maggiore di 3: con un invertitore si ha un circuito stabile alla soglia logica, con un numero pari di invertitori si ha un circuito che si stabilizza ad uno dei due livelli logici senza commutare mai.

NOTA2: è necessario, inoltre, che il tempo di salita e discesa del segnale attraverso un singolo invertitore sia molto inferiore al tempo di propagazione del segnale su tutto l'anello (metà del periodo di oscillazione), altrimenti il circuito non riesce a commutare correttamente e si comporta come un singolo invertitore stabilizzato alla soglia logica.

Analizzando periodo, frequenza di oscillazione e consumo dinamico si ottiene:

$$T = 2Nt_p \quad f = \frac{1}{T} = \frac{1}{2Nt_p} \quad P_{DYN} = N \cdot C_X V_{DD}^2 f \quad \text{per } N > t_{r,f}/t_p$$

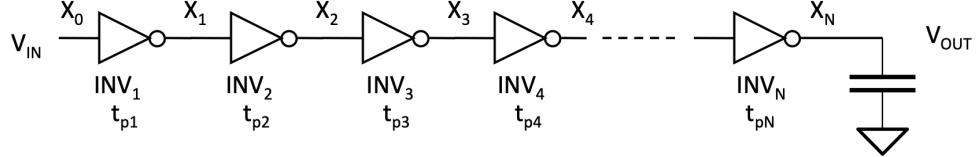
dove N è il numero di invertitori nell'anello, t_p è il tempo di propagazione di un singolo invertitore e C_X è la capacità di un nodo interno all'anello.



8.10 Buffer cmos

Struttura generale

Il buffer cmos è un circuito costituito da una serie di invertitori collegati in cascata, utilizzato per aumentare la capacità di pilotaggio del segnale di uscita e migliorare le prestazioni del circuito logico.



Si suppone per ipotesi che L , C_{d0} e C_{g0} siano uguali per tutti i transistor (pmos e nmos) di ogni invertitore. Di conseguenza anche il parametro γ risulta essere uguale per tutti gli invertitori del buffer e indipendente dalle dimensioni dei transistori.

$$\gamma_i = \frac{C_{OUT}}{C_{IN}} = \frac{C_{d0}}{C_{g0}L} \cdot \frac{W_p + W_n}{W_p + W_n} = \frac{C_{d0}}{C_{g0}L} = \text{costante per ogni invertitore}$$

Il tempo di ritardo complessivo è dato dalla somma dei tempi di ritardo di ogni singolo invertitore:

$$t_{p,tot} = \sum_{i=1}^N t_{p,i} \quad \text{con } t_{p,i} = t_{p0,i} \left(1 + \frac{f_i}{\gamma_i} \right)$$

Le possibili ottimizzazioni per minimizzare il tempo di ritardo del buffer sono:

- ottimizzare il tempo medio di ritardo intrinseco t_{p0}
- ottimizzare il numero di stadi N
- ottimizzare il fattore di carico f tra uno stadio e l'altro

Ottimizzazione di t_{p0} - dimensionamento del singolo invertitore

Negli invertitori il segnale viene ripetutamente invertito e il tempo di ritardo complessivo è dato dalla somma dei tempi di salita t_{pLH} e discesa t_{pHL} di ogni invertitore. Il modo migliore per ottimizzare il tempo complessivo è minimizzare la somma dei tempi di ritardo t_{pLH} e t_{pHL} , ovvero minimizzare il tempo di ritardo medio t_{p0} di ogni stadio. Di conseguenza (per come è stato visto in precedenza) si deve scegliere il rapporto tra i fattori di forma dei transistori come $\beta = \sqrt{\rho}$. In questo modo tutti gli invertitori hanno lo stesso tempo di ritardo intrinseco minimo:

$$t_{p0} = 0.69 \frac{R_{n0}C_{d0}L}{2} \left(1 + \beta + \rho + \frac{\rho}{\beta} \right) \quad R_{n0}, C_{d0}, L, \beta = \sqrt{\rho} \text{ costanti per ogni invertitore}$$

Ottimizzazione di f - dimensionamento progressivo degli stadi

Il fanout di un singolo invertitore dipende dal rapporto tra il fattore di forma dei suoi transistori e quello dell'invertitore successivo.

$$f_i = \frac{C_{IN,i+1}}{C_{IN,i}} = \frac{C_{g0}L(W_{p,i+1} + W_{n,i+1})}{C_{g0}L(W_{p,i} + W_{n,i})} = \frac{Z_{p,i+1} + Z_{n,i+1}}{Z_{p,i} + Z_{n,i}} = \frac{\beta Z_{n,i+1} + Z_{n,i+1}}{\beta Z_{n,i} + Z_{n,i}} = \frac{Z_{n,i+1}}{Z_{n,i}}$$

Si assume che tale rapporto sia costante per ogni stadio ottenendo così un dimensionamento progressivo dei transistori nei vari invertitori tale per cui ogni invertitore ha dimensioni f volte maggiori del precedente e ogni nodo ha capacità f volte maggiore del precedente:

$$\begin{aligned} Z_{n,i+1} &= f Z_{n,i} = f^i Z_{n,1} & W_{n,i+1} &= f W_{n,i} = f^i W_{n,1} & C_{X,j} &= f C_{X,j-1} = f^i C_{X,1} \\ Z_{p,i+1} &= f Z_{p,i} = f^i Z_{p,1} & W_{p,i+1} &= f W_{p,i} = f^i W_{p,1} \end{aligned}$$

Calcolo del valore di f ottimale

Si osserva che tutti gli stadi hanno lo stesso tempo di ritardo $t_{p,i}$, essendo t_{p0} , γ e f costanti per ogni invertitore, per cui è possibile calcolare il tempo di ritardo complessivo del buffer come:

$$t_{p,tot} = N \cdot t_p = N \cdot t_{p0} \left(1 + \frac{f}{\gamma}\right) \quad \text{per } t_p = t_{p0} \left(1 + \frac{f}{\gamma}\right)$$

Calcolando il fanout totale del buffer si ottiene il numero di stadi necessari per raggiungere il carico esterno C_L :

$$F = \frac{C_L}{C_{IN}} = \frac{C_{in,2}}{C_{in,1}} \frac{C_{in,3}}{C_{in,2}} \dots \frac{C_{in,N}}{C_{in,N-1}} = f_1 f_2 f_3 \dots f_N = f^N \rightarrow N = \frac{\ln(F)}{\ln(f)}$$

Sostituendo il valore di N con l'espressione appena trovata, si ottiene un'espressione per il tempo di ritardo complessivo con l'unica incognita f . Per determinare il valore ottimale di f si cerca il tempo di ritardo è minimo, analizzando la derivata rispetto a f :

$$t_{p,tot} = \frac{\ln F}{\ln f} t_{p0} \left(1 + \frac{f}{\gamma}\right) \quad \frac{\partial t_{p,tot}}{\partial f} \rightarrow f = e^{1+\frac{\gamma}{f}}$$

L'equazione precedente non ammette una soluzione analitica, per cui si ricorre a metodi numerici per trovare il valore ottimale di f . Per $\gamma = 1$ risulta $f = 3.6$.

Calcolo del valore di N , buffer invertente e non invertente

Una volta determinato il valore di f e conoscendo il fanout totale F si può calcolare il numero di stadi necessari per realizzare il buffer arrotondando il valore ottenuto all'intero più vicino:

$$N = \frac{\ln(F)}{\ln(f)}$$

Si osserva che se si vuole realizzare un buffer non invertente, è necessario avere un numero pari di stadi, mentre se si vuole realizzare un buffer invertente, è necessario avere un numero dispari di stadi.

Consumo dinamico del buffer

L'energia necessaria a far commutare l'uscita del buffer è usata per caricare e scaricare le capacità parassite di ogni invertitore e la capacità di carico esterna C_L . Per cui nella formula generale del consumo dinamico, al posto di C_{OUT} si deve usare la somma delle capacità dei nodi interni più la capacità del nodo di uscita:

$$C_{tot} = \sum_{i=1}^{N-1} C_{X,i} + C_F \quad \begin{aligned} C_{X,i} &= C_{OUT,i} + C_{IN,i+1} && \text{capacità nodo interno i-esimo} \\ C_F &= C_{OUT,N} + C_L && \text{capacità nodo di uscita} \end{aligned}$$

$$P_{DYN} = C_{tot} V_{DD}^2 f_{\text{frequenza}}$$

Impieghi

Il buffer cmos viene utilizzato per ridurre il tempo di ritardo quando si deve pilotare un carico esterno elevato. Alcuni esempi di impiego del buffer sono:

- pilotaggio di bus di dati o passaggio di segnali per piste lunghe con elevata capacità parassita
- pilotaggio di ingressi di circuiti logici con elevata capacità di ingresso
- pilotaggio da parte del clock di tutti i registri del computer

Siccome il buffer è costituito da più invertitori in cascata, il consumo dinamico è maggiore rispetto ad un singolo invertitore ed inoltre occupa più area sul chip. Per questo motivo si cerca di utilizzare il buffer solo quando strettamente necessario.

9 Logica statica complementare

9.1 Reti duali

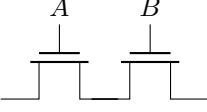
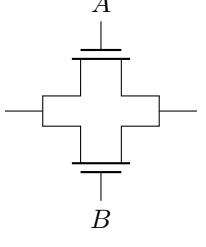
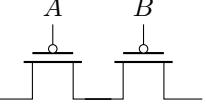
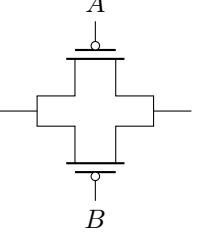
Rete di pull-up e rete di pull-down

Il principio alla base della logica statica complementare è quello di basarsi sulla struttura dell'invertitore cmos e generalizzarla per realizzare qualsiasi funzione logica combinatoria, in modo da sfruttare i vantaggi dell'invertitore:

- resistenza di ingresso infinita;
- consumo di potenza praticamente nullo in stato stazionario;
- elevata immunità al rumore, proprietà rigenerativa del segnale;
- la soglia logica e il tempo di propagazione intrinseci dipendono solo da Z_p/Z_n .

Per fare ciò, si sfrutta la presenza di due reti complementari: la rete di pull-up (PUN), costituita da mosfet di tipo p, e la rete di pull-down (PDN), costituita da mosfet di tipo n che si attivano in modo complementare per caricare o scaricare il nodo di uscita.

Serie e paralleli di mosfet

tipo di circuito	serie di nmos	parallelo di nmos	serie di pmos	parallelo di pmos																																																												
schema circuitale																																																																
funzione logica	AND $A \cdot B$	OR $A + B$	NOR $\overline{A + B}$	NAND $\overline{A \cdot B}$																																																												
tabella di verità	<table border="1"> <thead> <tr> <th>A</th><th>B</th><th>V_{out}</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>0</td></tr> <tr> <td>0</td><td>1</td><td>0</td></tr> <tr> <td>1</td><td>0</td><td>0</td></tr> <tr> <td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	V_{out}	0	0	0	0	1	0	1	0	0	1	1	1	<table border="1"> <thead> <tr> <th>A</th><th>B</th><th>V_{out}</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>0</td></tr> <tr> <td>0</td><td>1</td><td>1</td></tr> <tr> <td>1</td><td>0</td><td>1</td></tr> <tr> <td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	V_{out}	0	0	0	0	1	1	1	0	1	1	1	1	<table border="1"> <thead> <tr> <th>A</th><th>B</th><th>V_{out}</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>1</td></tr> <tr> <td>0</td><td>1</td><td>0</td></tr> <tr> <td>1</td><td>0</td><td>0</td></tr> <tr> <td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	A	B	V_{out}	0	0	1	0	1	0	1	0	0	1	1	0	<table border="1"> <thead> <tr> <th>A</th><th>B</th><th>V_{out}</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>1</td></tr> <tr> <td>0</td><td>1</td><td>1</td></tr> <tr> <td>1</td><td>0</td><td>1</td></tr> <tr> <td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	A	B	V_{out}	0	0	1	0	1	1	1	0	1	1	1	0
A	B	V_{out}																																																														
0	0	0																																																														
0	1	0																																																														
1	0	0																																																														
1	1	1																																																														
A	B	V_{out}																																																														
0	0	0																																																														
0	1	1																																																														
1	0	1																																																														
1	1	1																																																														
A	B	V_{out}																																																														
0	0	1																																																														
0	1	0																																																														
1	0	0																																																														
1	1	0																																																														
A	B	V_{out}																																																														
0	0	1																																																														
0	1	1																																																														
1	0	1																																																														
1	1	0																																																														

Interruttore duale o complementare

Due interruttori pilotati dallo stesso segnale sono duali o complementari se fissato un qualunque valore del segnale uno e uno solo dei due è acceso. Un esempio di interruttori duali sono un nmos e un pmos pilotati dallo stesso segnale, come avviene in un invertitore cmos.

Rete duale

Una rete duale è l'estensione del concetto di interruttori duali ad una rete di interruttori. Due reti pilotate dagli stessi ingressi sono duali se e solo se per ogni combinazione di ingressi una e una sola delle due reti è attiva. Il principio delle reti duali è alla base delle reti di pull-up e pull-down nella logica statica complementare. Da una rete di nmos si può ottenere la rete duale di pmos sostituendo ogni serie di nmos con un parallelo di pmos e viceversa.

9.2 Implementazione di porte logiche elementari

Principio generale

Data una funzione logica con N ingressi $F(A, B, C, \dots)$, si osserva che:

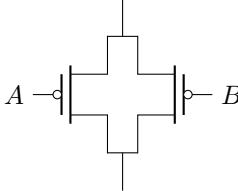
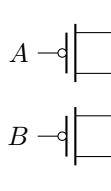
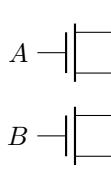
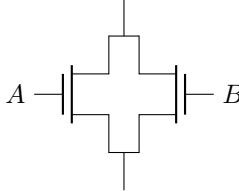
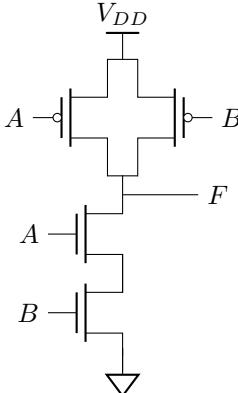
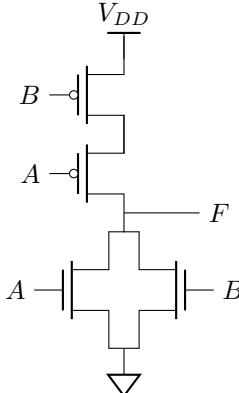
- per ogni combinazione di ingressi per cui $F(\dots) = 0$, l'uscita della porta logica deve essere $0V$, ovvero la rete di pull-down (PDN) deve essere attiva e collegare l'uscita a massa;
- per ogni combinazione di ingressi per cui $F(\dots) = 1$, l'uscita della porta logica deve essere V_{DD} , ovvero la rete di pull-up (PUN) deve essere attiva e collegare l'uscita a V_{DD} .

Per implementare una porta logica con logica statica complementare si segue il seguente procedimento:

1. si costruisce la rete di pull-down (PDN), che implementa la funzione logica $X = \overline{F}$
2. si costruisce la rete di pull-up (PUN), duale della PDN che implementa la funzione logica F
3. infine si collega la PUN tra V_{DD} e l'uscita, e la PDN tra l'uscita e massa

La caratteristica statica delle porte logiche realizzate con logica statica complementare è uguale a quella dell'invertitore cmos. Inoltre vale la proprietà rigenerativa del segnale.

Esempio di porte logiche elementari invertenti

tipo di circuito	porta NAND	porta NOR
funzione logica	$F = \overline{A \cdot B}$ $X = A \cdot B$	$F = \overline{A + B}$ $X = A + B$
simbolo logico		
rete di pull-up		
rete di pull-down		
porta logica completa		

9.3 Funzioni invertenti e porte logiche non invertenti

Funzioni invertenti e non invertenti

- Una funzione logica $F(X_1, X_2, \dots, X_N)$ è invertente se facendo variare un ingresso X_i da 0 a 1, dopo aver fissato gli altri ingressi, l'uscita F commuta nel verso opposto di X_i o rimane costante. Un esempio di funzioni invertenti sono le porte NAND e NOR.
- Una funzione logica $F(X_1, X_2, \dots, X_N)$ è non invertente se facendo variare un ingresso X_i da 0 a 1, dopo aver fissato gli altri ingressi, l'uscita F commuta nello stesso verso di X_i o rimane costante. Un esempio di funzioni non invertenti sono le porte AND e OR.
- Una funzione può non essere né invertente né non invertente, ad esempio la funzione XOR.

Limiti della logica statica complementare

Con la logica statica complementare è possibile implementare direttamente solo funzioni invertenti, come NAND e NOR (illustrate sopra). Per implementare funzioni non invertenti, come AND e OR, è necessario aggiungere un ulteriore stadio di inversione all'uscita oppure agli ingressi della porta logica invertente.

Esempi di porte logiche non invertenti

- AND non invertente: $F = A \cdot B = \overline{\overline{A} + \overline{B}} = \overline{\overline{A} \cdot \overline{B}}$ è implementabile come una porta NOR con ingressi invertiti, oppure come una porta NAND seguita da un invertitore.
- OR non invertente: $F = A + B = \overline{\overline{A} \cdot \overline{B}} = \overline{\overline{A} + \overline{B}}$ è implementabile come una porta NAND con ingressi invertiti, oppure come una porta NOR seguita da un invertitore.

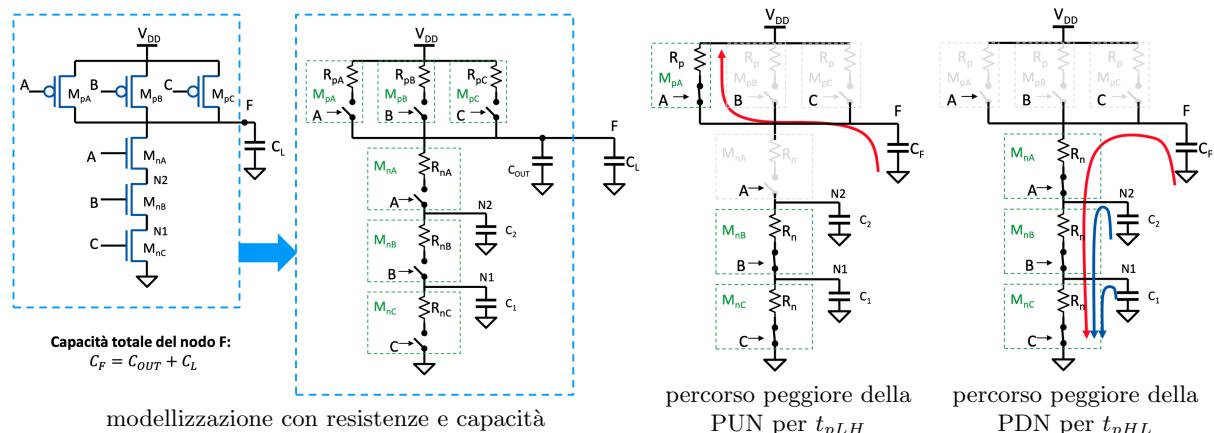
9.4 Tempi di ritardo

Ricerca del percorso peggiore e modello di Elmore

L'analisi dei tempi di ritardo di una porta logica a più ingressi consiste nell'analizzare la carica e scarica del nodo di uscita attraverso i percorsi di carica (nella PUN) e scarica (nella PDN) con resistenza equivalente massima, ovvero lungo il percorso peggiore. In genere il percorso peggiore è quello con maggior numero di mosfet in serie e minor numero di mosfet in parallelo.

Una volta individuato il percorso peggiore, si modella il circuito come una rete di Elmore, in cui ogni mosfet viene rappresentato come una resistenza (R_n o R_p) collegata a massa attraverso una capacità complessiva del nodo interno. Una volta tracciata la rete di Elmore, è possibile calcolare il tempo di ritardo t_{pHL} usando la rete di scarica (PDN) e il tempo di ritardo t_{pLH} usando la rete di carica (PUN), secondo le formule del modello di Elmore.

Di seguito l'esempio dell'individuazione dei percorsi peggiori nella PUN e PDN per il calcolo dei due tempi di ritardo t_{pLH} e t_{pHL} nel caso di una nand a 3 ingressi.



Per il caso sopra, i tempi di ritardo risultano:

$$t_{pLH} = 0.69 R_p (C_{OUT} + C_F) \quad t_{pHL} = 0.69 (R_n C_1 + 2R_n C_2 + 3R_n (C_{OUT} + C_L))$$

$$\text{con } C_1 = C_{dn,C} + C_{sn,B}, \quad C_2 = C_{dn,B} + C_{sn,A} \quad C_{OUT} = C_{dn,a} + C_{dp,A} + C_{dp,B} + C_{dp,C}$$

Tempo di ritardo medio

Il tempo di ritardo medio t_p di una porta logica in statica complementare risulta:

$$t_{p0} = \frac{t_{pHLO} + t_{pLHO}}{2} = 0.69 \frac{R_n(C_1 + 2C_2 + 3C_{OUT}) + R_p C_{OUT}}{2}$$

$$t_p = t_{p0} + 0.69 \frac{3R_n + R_p}{2} C_L = 0.69 R_{OUT} C_L \quad R_{OUT} = \frac{R_{PUN} + R_{PDN}}{2}$$

Dipendenza dal numero di ingressi

Dalle formule del modello di Elmore per una porta logica a N ingressi (N mosfet in parallelo e altrettanti in serie), il tempo di ritardo intrinseco dipende quadraticamente dal numero di ingressi N . Questo effetto non è desiderabile, in quanto porta ad un aumento significativo del tempo di ritardo al crescere del numero di ingressi della porta logica.

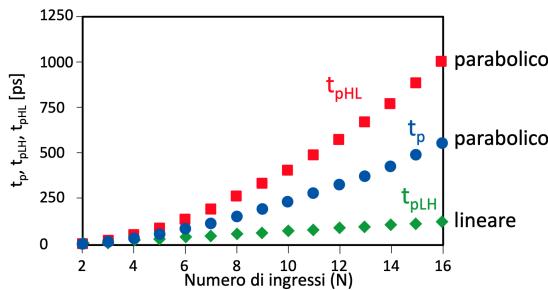
$$t_{p0} = 0.69 \frac{R_n(C_1 + C_2 + \dots + (N-1)C_{N-1} + NC_{OUT}) + R_p C_{OUT}}{2} \quad \text{con } C_{OUT} = C_{dn} + NC_{dp}$$

$$= 0.69 \frac{R_n(C_1 + C_2 + \dots + (N-1)C_{N-1} + N(C_{dn} + NC_{dp})) + R_p(C_{dn} + NC_{dp})}{2}$$

$$= 0.69 \frac{R_n(C_1 + C_2 + \dots + (N-1)C_{N-1} + NC_{dn} + \mathbf{N^2 C_{dp}}} + R_p(C_{dn} + NC_{dp})$$

Il contributo del carico esterno, invece, cresce linearmente con N :

$$t_p = t_{p0} + 0.69 \frac{\mathbf{NR_n} + R_p}{2} C_L$$



Analizzando il grafico del tempo di ritardo in funzione del numero di ingressi per una porta logica NAND, t_p cresce parabolicamente con N .

Bilanciamento PUN e PDN

Per uniformare il caso peggiore tra t_{pLH} e t_{pHL} , si può agire sul rapporto Z_p/Z_n dei mosfet, per bilanciare le resistenze equivalenti della PUN e PDN al caso peggiore.

$$R_{PU} = \alpha \frac{R_{p0}}{Z_p} \quad R_{PD} = \beta \frac{R_{n0}}{Z_n} \quad R_{PU} = R_{PD} \rightarrow Z_p = \frac{\alpha R_{p0}}{\beta R_{n0}} Z_n$$

Si osserva che per ottimizzare il tempo di ritardo è ulteriormente possibile avere Z_p e Z_n minori per i mosfet che sono collegati direttamente al nodo di uscita (coinvolti nel termine quadratico) e maggiori per i mosfet più lontani dall'uscita, senza però alterare il bilanciamento tra PUN e PDN.

Tempo di ritardo di più porte logiche in cascata

Quando si collegano in cascata più porte logiche (ad esempio per realizzare funzioni non invertenti come AND = NAND + NOT e OR = NOR + NOT), il tempo di ritardo complessivo è dato dalla somma dei tempi di ritardo delle singole porte logiche. In particolare il pedice del tempo di ritardo t_{pHL} o t_{pLH} complessivo è determinato dall'uscita dell'ultima porta logica della catena. Ad esempio per una AND realizzata come NAND + NOT, il tempo di ritardo complessivo risulta:

$$t_{pHL,AND} = t_{pLH,NAND} + t_{pHL,NOT} \quad t_{pLH,AND} = t_{pHL,NAND} + t_{pLH,NOT}$$

9.5 Problema dei nodi interni

Problema dei nodi interni

Il problema dei nodi interni si verifica quando, a causa della presenza di nodi interni tra mosfet in serie, il tempo di ritardo aumenta in modo significativo siccome, insieme alla capacità di uscita, devono essere caricate/scaricate anche tutte le capacità parassite dei nodi interni. Il numero di nodi interni è legato al numero di ingressi della porta logica, e quindi il problema si aggrava al crescere del numero di ingressi.

Soluzione 1: riordino degli ingressi

Si suppone che gli N ingressi di una porta logica provengano da altre reti combinatorie con tempi di ritardo diversi. Per cercare di contenere il problema dei nodi interni, si può riordinare gli ingressi in modo che la rete combinatoria con tempo di ritardo minore venga collegata al mosfet più vicino a massa (o a V_{DD} nella PUN) e la rete con tempo di ritardo maggiore venga collegata al mosfet più lontano dall'uscita. In questo modo, quando la rete più lenta commuta, le reti più veloci hanno già precaricato/scaricato i nodi interni e l'unica capacità da caricare/scaricare è quella di uscita.

Soluzione 2: suddivisione in più stadi

Quando il numero di ingressi resta comunque elevato, è possibile risolvere la questione dei nodi interni suddividendo la porta logica in più stadi di porte logiche con meno ingressi ciascuna. In questo modo si riduce il numero di nodi interni per ogni stadio, e si riduce il tempo di ritardo complessivo. Si osserva che il valore ottimo di ingressi per stadio si aggira tra 4 e 5. Ovviamente suddividendo il numero di ingressi in più stadi si aumenta il numero totale di mosfet necessari per realizzare la stessa funzione logica e quindi l'area occupata sul chip.

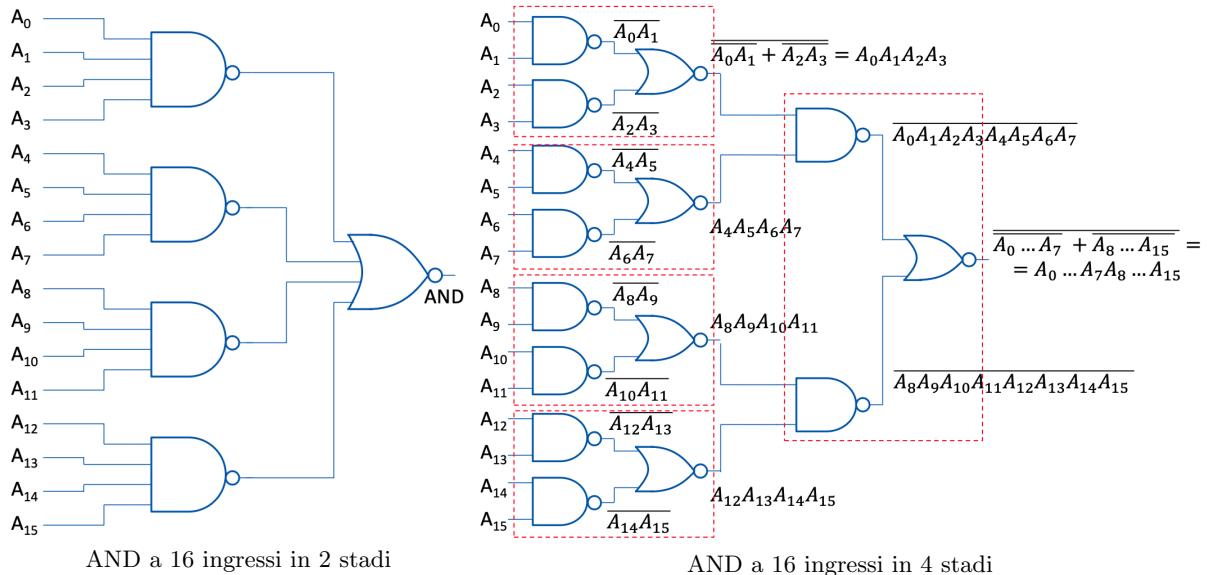
Per suddividere una porta logica in più stadi si possono utilizzare le leggi di De Morgan:

$$A + B + C + D \text{ (OR a 4 ingressi)} = \overline{\overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D}} \text{ (NOR + NAND a 2 ingressi)}$$

$$A \cdot B \cdot C \cdot D \text{ (AND a 4 ingressi)} = \overline{\overline{A} \cdot \overline{B} + \overline{C} \cdot \overline{D}} \text{ (NAND + NOR a 2 ingressi)}$$

Di seguito un confronto delle prestazioni di una porta AND a 16 ingressi realizzata con numero differente di stadi:

numero di stadi	tempo di ritardo intrinseco	numero di mosfet
1 stadio da 16 ingressi	$t_{p0} \approx 0.69RCN^2 = 0.69\ 256\ RC$	32 mosfet
2 stadi da 4 ingressi ciascuno	$t_{p0} \approx 2 \cdot 0.69RCN^2 = 0.69\ 32\ RC$	40 mosfet
4 stadi da 2 ingressi ciascuno	$t_{p0} \approx 4 \cdot 0.69RCN^2 = 0.69\ 16\ RC$	60 mosfet



9.6 Consumo di potenza

Consumo dinamico

Il consumo dinamico di una porta logica si basa sulla formula generale definita in precedenza, solo che essendoci più ingressi la frequenza media di commutazione f viene sostituita dalla frequenza effettiva di commutazione dell'uscita calcolata come prodotto tra la frequenza di commutazione degli ingressi f e il fattore di attività α_F dell'uscita.

$$P_{DYN} = C_F \cdot V_{DD}(V_H - V_L) \cdot f \cdot \alpha_F \quad f \cdot \alpha_F = \text{frequenza effettiva di commutazione dell'uscita}$$

Fattore di attività

Il fattore di attività α_F di una porta logica è definito come la probabilità che l'uscita della porta logica commuti da 0 a 1 in un dato intervallo di tempo. Definito p_F la probabilità che l'uscita della porta logica sia 1 e supponendo che l'uscita di una porta logica non dipenda dal suo stato precedente (tempo invariante), il fattore di attività risulta:

$$\alpha_F = P(0 \rightarrow 1) = P(F = 0 \text{ per } t = i - 1) \cdot P(F = 1 \text{ per } t = i) = p_F \cdot (1 - p_F)$$

La probabilità p_F che l'uscita della porta logica sia a 1 dipende dalla funzione logica implementata e dalle probabilità degli ingressi p_{X_i} (che si suppone siano tutte indipendenti e tempo invarianti). Per calcolare p_F si può utilizzare:

- la tabella di verità della porta logica, calcolando la somma delle probabilità delle combinazioni di ingressi che portano all'uscita a 1
- l'espressione logica della funzione F , convertendo le operazioni logiche tra ingressi in operazioni algebriche sulle probabilità degli ingressi secondo le seguenti regole e facendo attenzione ad eliminare le intersezioni (sulla OR):

$$\text{NOT} \rightarrow p_{\bar{X}} = 1 - p_X \quad \text{AND} \rightarrow p_{A \cdot B} = p_A \cdot p_B \quad \text{OR} \rightarrow p_{A+B} = p_A + p_B - p_A \cdot p_B$$

- le mappe di Karnaugh per semplificare l'espressione logica in somme esclusive di prodotti, in modo da evitare le intersezioni e facilitare il calcolo delle probabilità

Di seguito le probabilità p_F e i fattori di attività α_F per le porte logiche elementari a 2 ingressi:

Funzione logica	P_F	Fattore di attività α_F
AND	$p_A p_B$	$p_A p_B \cdot (1 - p_A p_B)$
NAND	$1 - p_A p_B$	$p_A p_B \cdot (1 - p_A p_B)$
OR	$p_A + (1 - p_A)p_B$	$(p_A + (1 - p_A)p_B) \cdot (1 - (p_A + (1 - p_A)p_B))$
NOR	$(1 - p_A)(1 - p_B)$	$(p_A + (1 - p_A)p_B) \cdot (1 - (p_A + (1 - p_A)p_B))$
XOR	$(p_A + p_B - 2p_A p_B)$	$(p_A + p_B - 2p_A p_B) \cdot (1 - p_A - p_B + 2p_A p_B)$
XNOR	$1 - p_A - p_B + 2p_A p_B$	$(p_A + p_B - 2p_A p_B) \cdot (1 - p_A - p_B + 2p_A p_B)$

Capacità logica

Si nota che nella formula del consumo dinamico compare la capacità C_F che rappresenta la capacità totale del nodo di uscita della porta logica. È possibile definire la capacità logica come il prodotto tra la capacità C_F e il fattore di attività α_F .

$$C_{\text{logica}} = C_F \cdot \alpha_F$$

In questo modo la formula del consumo dinamico può essere interpretata come il consumo dinamico per caricare e scaricare ad una frequenza f una capacità effettiva pari alla capacità logica.

$$P_{DYN} = C_{\text{logica}} \cdot V_{DD}(V_H - V_L) \cdot f$$

Il consumo dinamico totale è dato dalla somma dei consumi dinamici di ogni nodo interno. In alternativa si può calcolare il consumo dinamico totale usando la capacità logica equivalente dell'intera rete data dalla somma delle capacità di ogni singolo nodo.

10 Logica a Pass Transistor

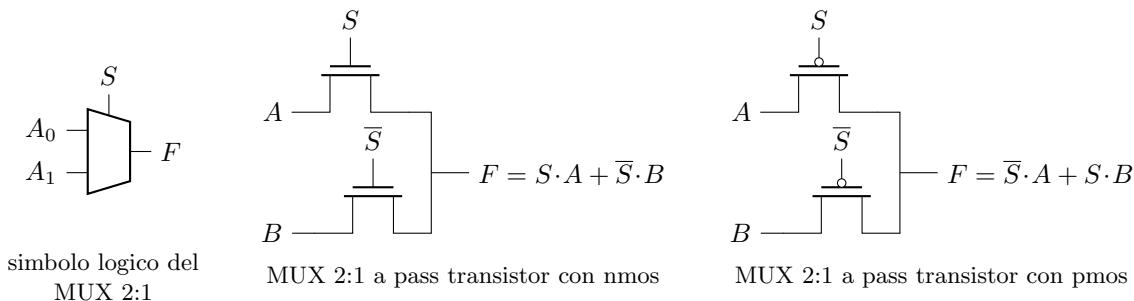
10.1 Struttura base

Principio di funzionamento

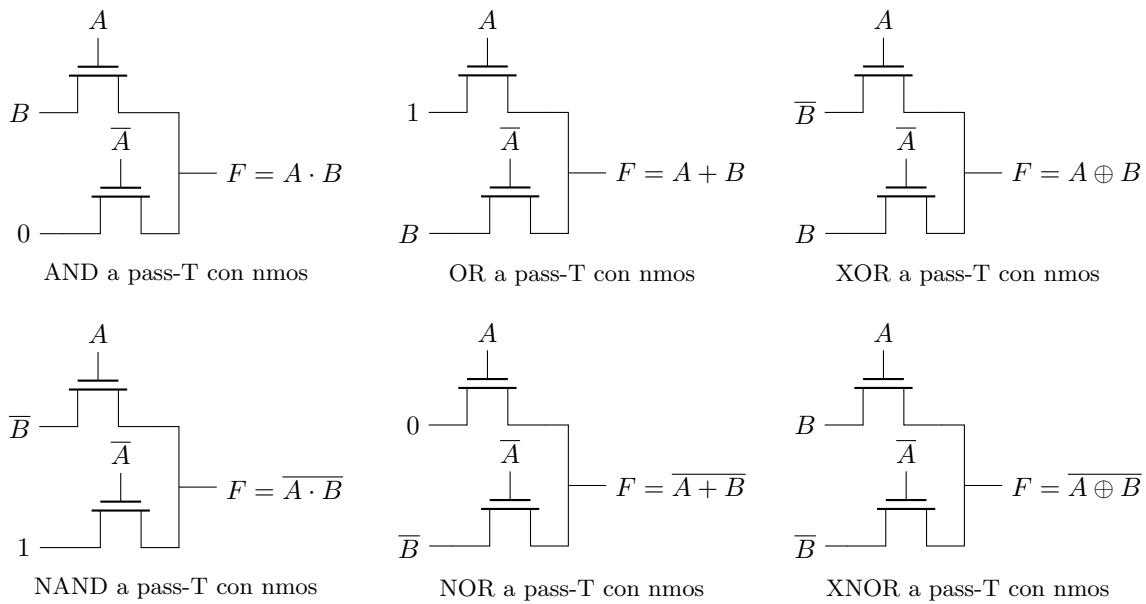
I pass transistor sono transistor usati come interruttori, controllati da un segnale di controllo (gate) che permettono o bloccano il passaggio di un certo segnale in ingresso verso l'uscita. A differenza delle porte logiche complementari, i pass transistor non sono collegati necessariamente a V_{DD} o a massa, ma possono trasmettere direttamente i segnali logici di ingresso. Inoltre le porte logiche a pass transistor possono essere costruite sia con nmos che con pmos e non necessitano di reti complementari.

Multiplexer

Il multiplexer è un circuito che seleziona uno tra più segnali di ingresso in base al valore di uno o più segnali di selezione. La configurazione base di un multiplexer prevede due ingressi e un segnale di selezione. Sono facilmente realizzabili con pass transistor.



Porte elementari (AND, OR, XOR, NAND, NOR, XNOR) a pass transistor



Teorema di Shannon e implementazione di funzioni complesse

Ogni funzione logica di N variabili $F(A, B, C \dots)$ può essere espressa nel seguente modo:

$$F = A \cdot F(1, B, C \dots) + \bar{A} \cdot F(0, B, C \dots) = A \cdot F_1(B, C \dots) + \bar{A} \cdot F_0(B, C \dots)$$

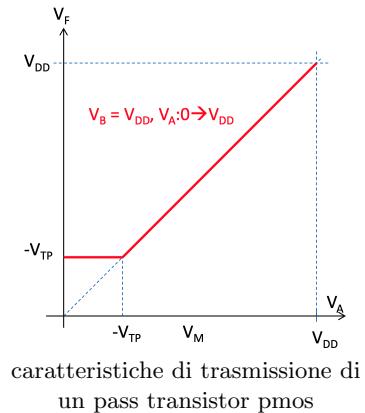
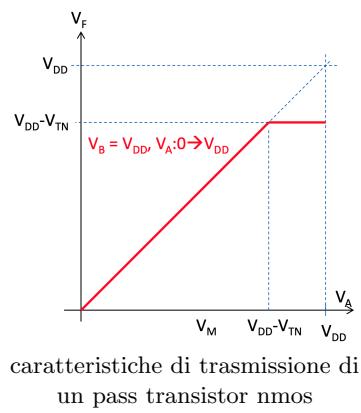
Questa espressione permette di implementare qualsiasi funzione logica usando multiplexer a pass transistor, dove la variabile A viene usata come segnale di selezione e le uscite delle due funzioni $F(0, B, C \dots)$ e $F(1, B, C \dots)$ vengono collegate agli ingressi del multiplexer. Il teorema può essere applicato in modo ricorsivo per ridurre ulteriormente le funzioni F_0 e F_1 fino ad ottenere solo porte logiche elementari.

10.2 Trasmissione dei valori logici cattivi

Trasmissione del valore logico alto con nmos e basso con pmos

La trasmissione di un valore logico alto tramite un pass transistor nmos avviene correttamente solo fino a $V_{DD} - V_{TN}$. Quando la tensione in ingresso raggiunge questo valore, la tensione tra source e gate uguaglia la tensione di soglia $V_G - V_S = V_{DD} - (V_{DD} - V_{TN}) = V_{TN}$ e il transistor entra in interdizione.

In analogo al valore alto per nmos, la trasmissione di un valore logico basso tramite un pass transistor pmos avviene correttamente solo fino a $-V_{TP}$. Quando la tensione in ingresso raggiunge questo valore, la tensione tra source e gate uguaglia la tensione di soglia $V_S - V_G = 0 - (-V_{TP}) = V_{TP}$ e il transistor entra in interdizione.

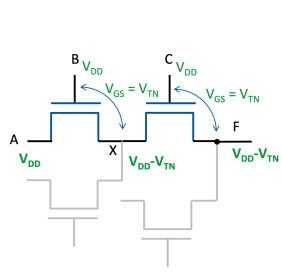


Dai grafici si osserva che i pass transistor hanno guadagno statico unitario (la pendenza della caratteristica è 1), per cui non hanno proprietà rigenerativa del segnale e il rumore viene trasmesso senza attenuazione. Inoltre l'effetto body peggiora ulteriormente la trasmissione dei valori logici "cattivi", aumentando in modulo la tensione di soglia V_{TN} e V_{TP} .

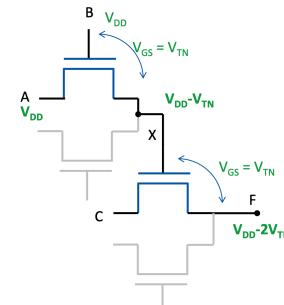
Propagazione ed effetti dei valori logici cattivi per pass transistor nmos

L'uscita di un pass transistor nmos che trasmette un valore logico alto risulta degradata a $V_{DD} - V_{TN}$ può essere collegata in tre modi principali:

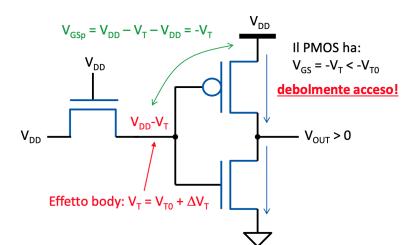
- al source/drain di un altro nmos: la trasmissione del valore logico alto degradato dal primo nmos non viene ulteriormente peggiorata dal secondo nmos, in quanto il secondo riesce a trasmettere correttamente il valore logico degradato $V_{DD} - V_{TN}$
- al gate di un altro nmos: in questo caso il valore logico alto $V_{DD} - V_{TN}$ limita la tensione di accensione del secondo nmos, che a sua volta peggiorerà ulteriormente il valore logico alto in uscita a $V_{DD} - 2V_{TN}$
- all'ingresso di una porta logica complementare: in questo caso il valore logico alto degradato $V_{DD} - V_{TN}$ può causare la parziale accensione della pull-up-network, causando un aumento del consumo statico della porta logica



trasmissione del valore degradato attraverso due pass transistor nmos in serie



trasmissione del valore degradato verso il gate di un pass transistor nmos



trasmissione del valore degradato verso una porta logica complementare (invertitore)

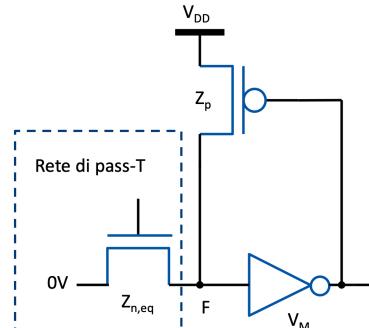
10.3 Ottimizzazione dei valori logici - level restorer e transmission gate

Level Restorer

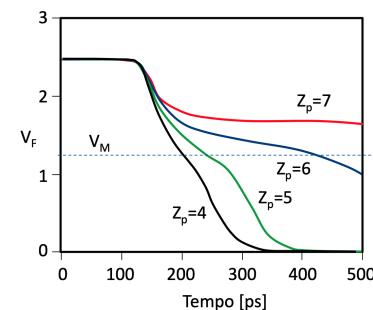
Per ripristinare i valori logici degradati in uscita da un pass transistor nmos o pmos, si può usare un circuito chiamato level restorer, costituito da un invertitore e un transistor di pull-up o pull-down (in base al valore logico da ripristinare). Analizzando il funzionamento di un level restorer usato per ripristinare il valore logico alto in uscita:

- quando il segnale in ingresso al level restorer è basso, l'uscita dell'invertitore è alta e il transistor di pull-up (pmos) è spento, per cui il circuito di level restorer non entra in funzione
- quando invece il segnale in ingresso è alto, l'uscita dell'invertitore è bassa e il transistor di pull-up è acceso, forzando l'uscita della rete pass transistor F al valore logico alto V_{DD} .

Quando l'uscita del nodo F inizialmente a V_{DD} (con il transistor di pull-up è acceso), deve essere portata al valore logico basso, è necessario che raggiunga la tensione V_M in modo da far commutare l'invertitore e spegnere il pmos, altrimenti questo continuerà a forzare l'uscita al valore logico alto. Per fare in modo che ciò avvenga, è necessario che il pmos abbia un fattore di forma Z_p inferiore al fattore di forma $Z_{n,eq}$ della rete pass transistor usata per generare il segnale F . Se questo non avviene, il transistor di pull-up potrebbe causare un ritardo nella discesa del segnale in uscita o addirittura la mancata commutazione del segnale.



circuito di un level restorer per pass transistor nmos

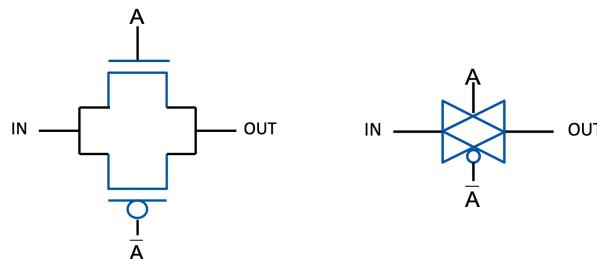


curva della commutazione HL del nodo F al variare di Z_p

Porta di trasmissione o transmission gate - TG

Per prevenire la degradazione dei valori logici (in alternativa al level restorer) è possibile usare le porte di trasmissione o transmission gate (TG) al posto dei semplici pass transistor nmos o pmos. Le porte TG sono costituite da un transistor nmos e un transistor pmos collegati in parallelo, controllati da segnali di gate complementari. In questo modo si accendono e si spengono insieme e si compensano a vicenda i difetti di trasmissione dei valori logici:

- il transistor nmos trasmette correttamente il valore logico basso fino a 0V
- il transistor pmos trasmette correttamente il valore logico alto fino a V_{DD}



schema circuitale di una porta di trasmissione (transmission gate)
e relativo simbolo di abbreviazione

10.4 Tempi di propagazione e ottimizzazioni

Analisi dei tempi di propagazione

Per calcolare il tempo di propagazione di un circuito a pass transistor si procede similmente a quanto visto per le reti di pull-up e pull-down delle porte logiche complementari:

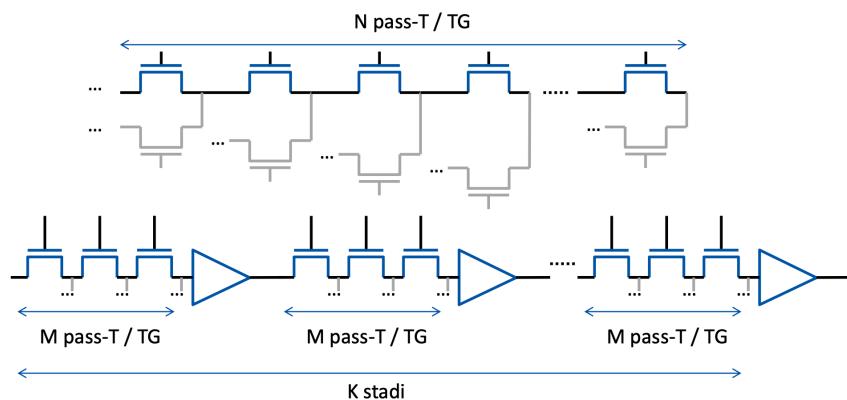
1. si individua il percorso peggiore (serie più lunga di pass transistor)
2. si modella il percorso peggiore secondo la rete di Elmore
3. si calcolano i tempi di propagazione usando le formule viste per le reti di Elmore.

Alcune osservazioni importanti riguardo al calcolo dei tempi di propagazione nei circuiti a pass transistor:

- in base al tipo di commutazione (LH o HL) la resistenza equivalente del pass transistor raddoppia se si sta trasmettendo un valore logico “cattivo”
- se si usano insieme pmos e nmos, il percorso peggiore può variare in base al tipo di commutazione (LH o HL) e non necessariamente coincide con il percorso con la serie di mosfet più lunga, specialmente se $R_n \neq R_p$
- quando si usano le porte TG, la resistenza equivalente del TG è data dal parallelo formato dalle due resistenze R_n e R_p degli nmos e pmos, di cui necessariamente una e una sola raddoppiata siccome trasmette un valore logico cattivo
- i tempi di propagazione intrinseci delle reti a pass transistor dipendono quadraticamente dal numero di pass transistor in serie e tale dipendenza può provocare ritardi molto elevati in circuiti complessi

Ottimizzazione dei tempi di propagazione - buffering

Per ottimizzare i tempi di propagazione dei circuiti a pass transistor si può usare la tecnica del buffering, che consiste nel suddividere il percorso di propagazione del segnale in più stadi, inserendo, tra uno stadio e l’altro, dei buffer costituiti da invertitori cmos in logica statica complementare. Questi hanno la funzione di interrompere la catena di pass transistor e deviare le correnti di carica/scarica a massa o V_{DD} . In questo modo si riduce la lunghezza del percorso peggiore e di conseguenza il tempo di propagazione complessivo del circuito.



Per calcolare il numero ottimale di stadi K e il numero di pass transistor per stadio M bisogna:

1. calcolare il tempo di propagazione per un singolo stadio $t_{p,stadio} = t_{buffer} + t_{p,pass}$ secondo la rete di Elmore, facendo attenzione che il nodo finale di ogni stadio ha come carico la capacità di ingresso dell'invertitore successivo e un source/drain in meno
2. calcolare il tempo di propagazione complessivo come $t_{p,tot} = K \cdot t_{p,stadio}$ ed effettuare le opportune sostituzioni $K = N/M$ per avere $t_{p,tot}$ in funzione di M solamente
3. derivare $t_{p,tot}$ rispetto a M e porre la derivata uguale a zero per trovare il valore ottimale di M che minimizza il tempo di propagazione complessivo (tipicamente $3 \leq M \leq 5$)
4. una volta trovato M intero, si può calcolare $K = N/M$

NOTA: i buffer possono essere invertenti o non invertenti, se sono invertenti è opportuno fare attenzione ad eventuali negazioni durante la elaborazione logica del segnale.

10.5 Consumo dinamico

Il consumo dinamico dei circuiti a pass transistor si basa sempre sulla formula generale del consumo dinamico, con le stesse variabili e considerazioni viste per le porte logiche complementari.

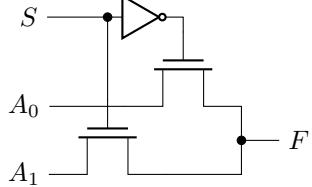
$$P_{DYN} = C_F \cdot V_{DD}(V_H - V_L) \cdot f \cdot \alpha_F$$

L'unica cosa di cui fare attenzione è l'escursione tra i valori logici alto e basso, che varia in base alla configurazione usata (nmos, pmos o TG) e alla presenza di level restorer.

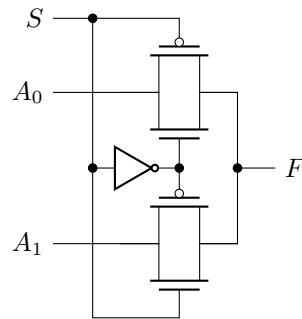
Per avere il consumo dinamico totale va fatta la somma dei consumi di ogni singolo nodo con la propria C_X e α_X . In alternativa si può calcolare il consumo dinamico totale usando la capacità logica equivalente dell'intera rete data dalla somma delle capacità di ogni singolo nodo.

10.6 Costruzioni di porte logiche

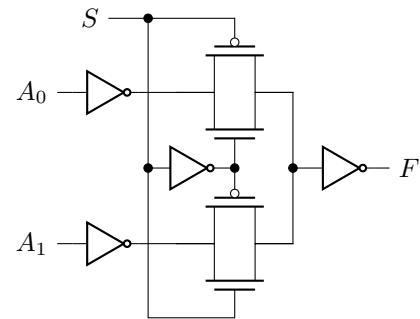
MUX a pass-T e a TG



MUX 2:1 a pass transistor con nmos, non rigenerativo, bidirezionale, con escursione logica limitata



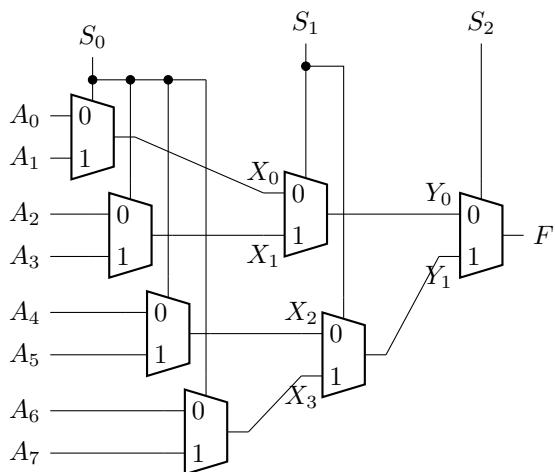
MUX 2:1 a transmission gate, non rigenerativo, bidirezionale, con escursione logica completa



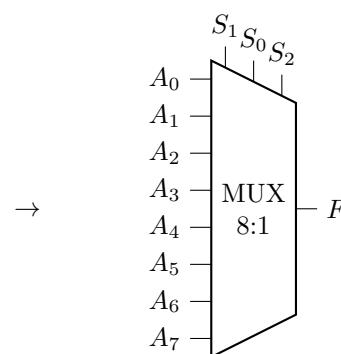
MUX 2:1 a transmission gate con invertitori, rigenerativo, unidirezionale con escursione logica completa

MUX a più stadi - struttura ad albero

Di seguito una rappresentazione di un MUX 8:1 a più stadi organizzati secondo una struttura ad albero, realizzato con MUX 2:1. I singoli MUX 2:1 possono essere realizzati con transmission gate per garantire un'escursione logica completa. Inoltre è possibile aggiungere degli invertitori per assicurare un funzionamento rigenerativo e unidirezionale, con l'accorgimento di non ripeterli per i nodi intermedi $X_0, X_1, X_2, X_3, Y_0, Y_1$ e lasciare solo quelli sugli ingressi e all'uscita e ad ogni M stadi, come visto nella sezione precedente sul buffering.



MUX 8:1 a più stadi con struttura ad albero realizzato con MUX 2:1

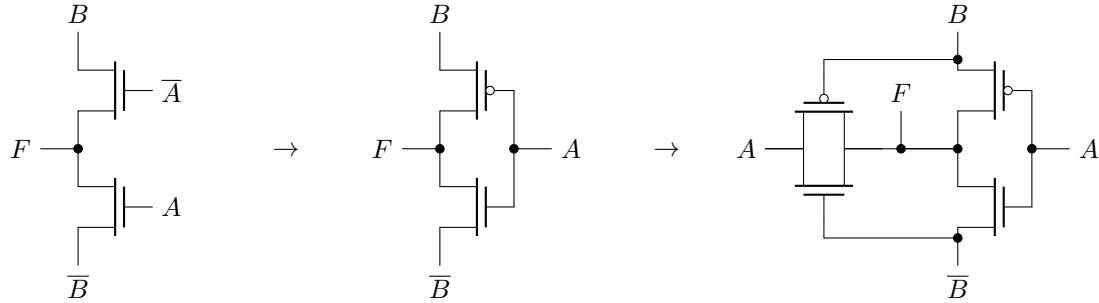


simbolo di abbreviazione del MUX 8:1

Porta XOR ibrida a pass-T / TG

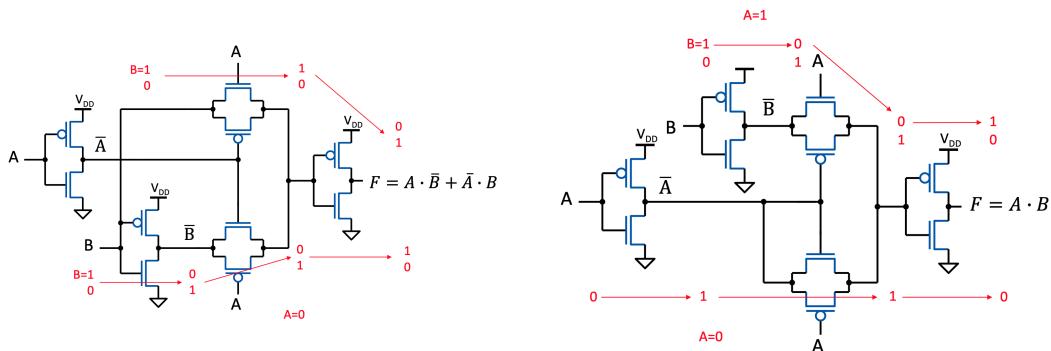
Di seguito un esempio di implementazioni di XOR usando pass transistor nmos, pmos e un'implementazione ibrida con transmission gate e pass transistor per correggere la trasmissione di valori logici degradati secondo le seguenti osservazioni:

- la prima implementazione usa solo pass-T nmos, per cui il valore logico alto in uscita sarà degradato
- la seconda implementazione usa solo pass-T pmos, per cui se $B = 0$ il valore logico A sarà degradato
- la terza implementazione ibrida usa un transmission gate aggiuntivo per trasmettere il valore logico di A corretto quando $B = 0$



Sommatore a 1 bit - Half Adder

Un half adder (sommatore a 1 bit senza riporto) è costituito da due porte logiche: una XOR per il calcolo della somma e una AND per il calcolo del riporto. Di seguito le implementazioni delle due funzioni logiche usando transmission gate secondo la struttura a MUX.

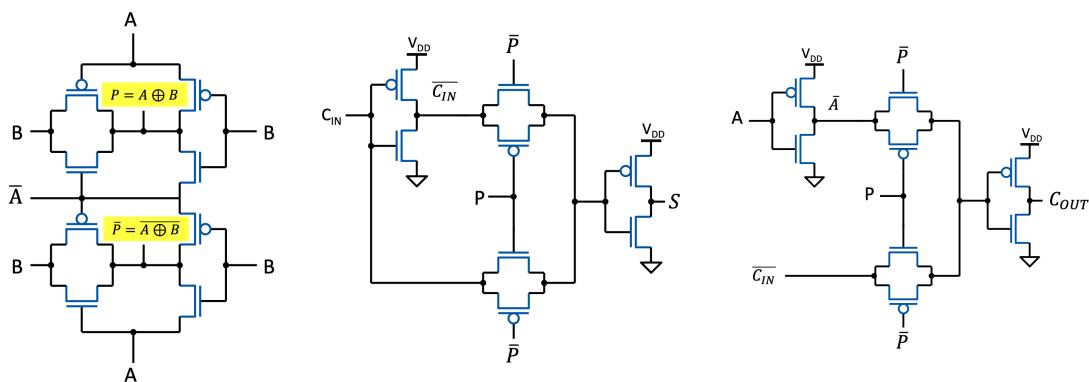


Sommatore a 1 bit con riporto - Full Adder

Un full adder (sommatore a 1 bit con riporto) è costituito, invece, da due funzioni più complesse:

$$S = A \oplus B \oplus C_{in} = P \oplus C_{in} \quad C_{out} = A \cdot (\overline{A} \oplus \overline{B}) + (A \oplus B) \cdot C_{in} = A \cdot \overline{P} + P \cdot C_{in} \quad P = A \oplus B$$

Si nota, quindi, che è possibile impiegare una XOR e una XNOR per calcolare P e \overline{P} (1° stadio), una XOR per calcolare la somma S (2° stadio) e un MUX 2:1 per calcolare il riporto C_{out} (3° stadio).



11 Fabbricazione dei circuiti integrati

11.1 Fasi di progettazione e costruzione

Fasi di progettazione

La fase di progettazione è svolta dal progettista del circuito integrato che ha conoscenze approfondite di elettronica e del funzionamento dei componenti da realizzare. Le fasi di progettazione sono:

1. definizione delle specifiche del circuito integrato
2. progettazione dello schema a blocchi
3. progettazione dello schema circuitale
4. progettazione del layout

Fasi di costruzione

Le fasi di costruzione sono svolte in un impianto di fabbricazione da tecnici specializzati che non necessariamente hanno conoscenze approfondite in elettronica. Le fasi di costruzione sono:

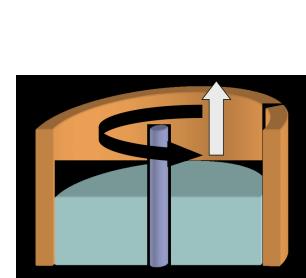
- | | |
|--|------------------------------|
| 1. fabbricazione del wafer (metodo di Czochralski) | |
| 2. selezione delle regioni attive | Front End of the Line - FEOL |
| 3. selezione del tipo di substrato | |
| 4. ossido di gate ed elettrodo di gate | |
| 5. diffusioni N+ e P+ | |
| 6. contatti e interconnessioni | Back End of the Line - BEOL |
| 7. packaging | |

Nelle spiegazioni successive si analizzeranno le varie fasi di produzione prendendo come modello la costruzione di un invertitore CMOS.

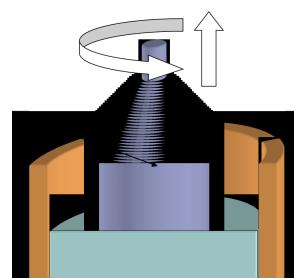
11.2 Fabbricazione del Wafer con metodo di Czochralski

Attraverso il metodo di Czochralski si ottiene un lingotto di silicio monocristallino che viene successivamente tagliato in fette sottili (wafer). Il processo prevede i seguenti passaggi:

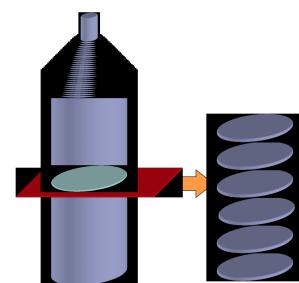
1. si fonde del silicio ad alta purezza in un crogolio di quarzo, è possibile aggiungere elementi chimici (ad esempio arsenico o fosforo) in maniera controllata per il droggaggio di base
2. si immmerge di un seme di silicio monocristallino (barra di silicio puro) nella massa fusa e viene messo in rotazione attorno al suo asse verticale
3. il seme viene estratto lentamente in modo che gli atomi di silicio fuso si solidificano e si dispongono naturalmente secondo la struttura cristallina del seme (per una proprietà propria del silicio), la velocità di estrazione controlla il diametro del lingotto
4. il lingotto viene tagliato in fette sottili (wafer) tramite una sega a filo abrasivo di diamante
5. i wafer vengono sottoposti a processi di lappatura e molatura per ottenere una superficie perfettamente piana che viene ricoperta con uno strato sottile di ossido di silicio per proteggerla



inserimento del seme nel crogolio



estrazione e formazione del lingotto



taglio del lingotto in wafer

11.3 Processo selettivo di costruzione del circuito integrato per litografia

Divisione di un wafer in die

Ogni wafer viene suddiviso a scacchiera in tante aree quadrate chiamate die (o chip). Ogni die compone un circuito integrato completo che alla fine della lavorazione verrà tagliato e confezionato singolarmente. La dimensione dei die dipende dallo spazio occupato dal circuito integrato e dal numero di difetti critici presenti nel wafer.

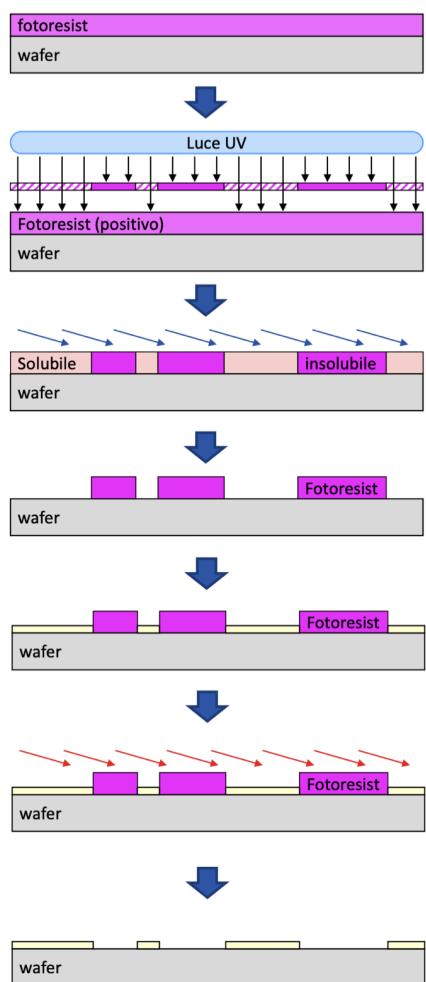
Un difetto critico è un difetto che rende inutilizzabile il circuito integrato. La probabilità di avere un difetto critico aumenta con l'aumentare della superficie del die. Per questo motivo, per aumentare la resa di produzione, si tende a ridurre la dimensione dei die.

Maschera

Per identificare le aree del wafer che devono essere lavorate in ogni fase del processo di costruzione, si utilizza una maschera. La maschera è una lastra di quarzo trasparente su cui sono incisi i disegni delle aree che devono essere lavorate.

Processo selettivo per litografia

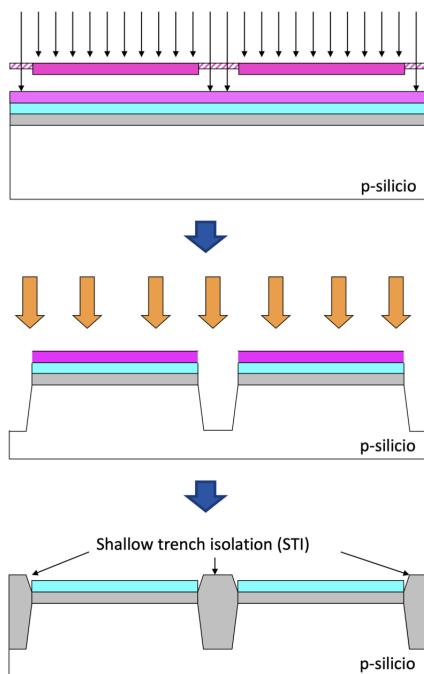
Per selezionare effettivamente le aree del wafer da lavorare e proteggere le altre si utilizza un processo selettivo per litografia che prevede i seguenti passaggi:



1. si ricopre il wafer con uno strato sottile di materiale fotosensibile chiamato fotoresist
2. si espone il wafer alla luce ultravioletta attraverso la maschera, le aree del fotoresist esposte alla luce cambiano le loro proprietà chimiche e diventano solubili in un apposito solvente
3. si sciolgono le aree del fotoresist diventate solubili, lasciando scoperte le aree del wafer che devono essere lavorate
4. si ottiene così il pattern desiderato sul wafer in cui le aree scoperte possono essere lavorate e il resto della superficie è protetta dal fotoresist
5. si esegue la lavorazione desiderata (ad esempio l'ossidazione o l'impiantazione ionica)
6. si rimuove il fotoresist rimanente con un altro solvente apposito
7. si ottiene così il wafer con le aree lavorate secondo il pattern desiderato

11.4 Selezione delle regioni attive

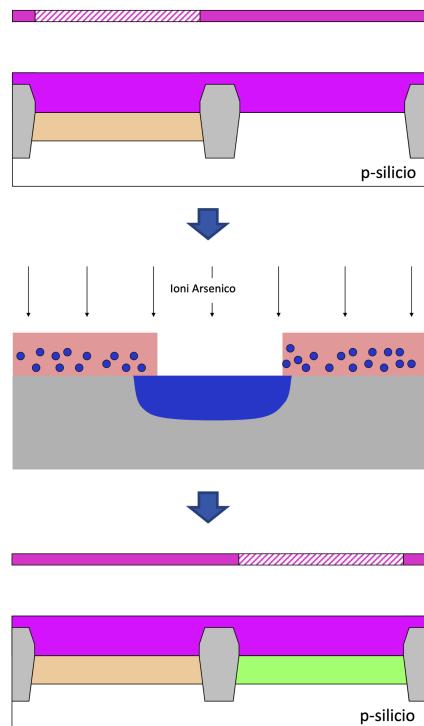
Il processo di selezione delle regioni attive consiste nel definire le regioni del wafer in cui verranno realizzati i singoli mosfet e inserire delle barriere di ossido per isolare elettricamente le varie regioni. Il processo prevede i seguenti passaggi:



1. si deposita uno strato di nitruro di silicio (Si_3N_4) e si utilizza il processo di litografia selettiva per selezionare le aree in cui vanno inserite le barriere di ossido isolante
2. si rimuove il nitruro di silicio, l'ossido di silicio e una parte del substrato di silicio nelle aree selezionate tramite un attacco chimico con acido; per scavare verticalmente (e non anche lateralmente) le cavità, il solvente viene ionizzato e nebulizzato così, sotto l'azione di un campo elettrico verticale, gli ioni vengono direzionali e colpiscono il wafer solo verticalmente senza intaccare le pareti laterali; questo processo è chiamato Reactive Ion Etching (RIE)
3. si rimuove il fotoresist rimanente e si deposita l'ossido di isolamento (di bassa qualità siccome deve solo fungere da isolante) nelle regioni scoperte dal nitruro (ovvero nelle cavità scavate con il RIE) formando le Shallow Trench Isolation (STI)
4. si rimuove il nitruro di silicio e lo strato di ossido di silicio rimanenti, in modo da liberare la superficie di silicio puro del wafer

11.5 Selezione del tipo di substrato

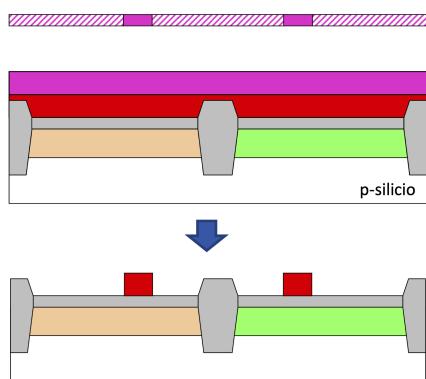
Il processo di selezione del tipo di substrato consiste nel creare un substrato di tipo P o N a seconda del tipo di mosfet che si vuole realizzare all'interno delle aree attive definite nel punto precedente. Il processo prevede i seguenti passaggi:



1. si utilizza il processo di litografia selettiva per selezionare le regioni attive in cui si vuole creare un substrato di tipo P (per la realizzazione di mosfet NMOS) o di tipo N (per la realizzazione di mosfet PMOS)
2. si bombardano le regioni scoperte con ioni droganti (ad esempio arsenico per il substrato di tipo P o fosforo per il substrato di tipo N) che penetrano nel silicio; questo processo è chiamato impiantazione ionica
3. si riscalda il wafer (annealing) per distribuire il drogante e farlo disporre correttamente nella struttura cristallina del silicio in modo da formare il substrato di tipo desiderato; questo è chiamato attivazione del droggaggio
4. si rimuove il fotoresist rimanente e si ripete tutto il processo di impiantazione ionica e attivazione del droggaggio con annealing per creare le aree di substrato del tipo opposto necessarie per la realizzazione dei mosfet complementari

11.6 Ossido di gate ed elettrodo di gate

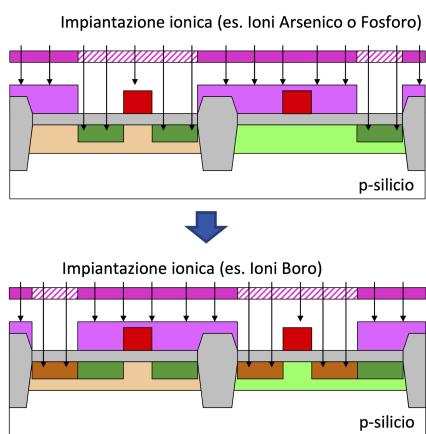
La fase successiva nella realizzazione dei mosfet è deposizione dell'ossido di gate e dell'elettrodo di gate in polisilicio. Il processo prevede i seguenti passaggi:



1. si deposita uno strato sottile di ossido di silicio (SiO_2) di alta qualità (costituirà il dielettrico dei condensatori dei gate dei mosfet) su tutta la superficie del wafer, non richiede un processo selettivo in quanto lo strato di ossido ha spessore trascurabile rispetto alle STI
2. si deposita uno strato di polisilicio (silicio policristallino con proprietà metalliche) sopra l'ossido di silicio per formare l'elettrodo di gate dei mosfet
3. si utilizza il processo di litografia selettiva per selezionare le aree in cui si vogliono formare i gate dei mosfet e si elimina il polisilicio nelle aree scoperte perché non selezionate tramite un attacco chimico

11.7 Diffusioni N+ e P+

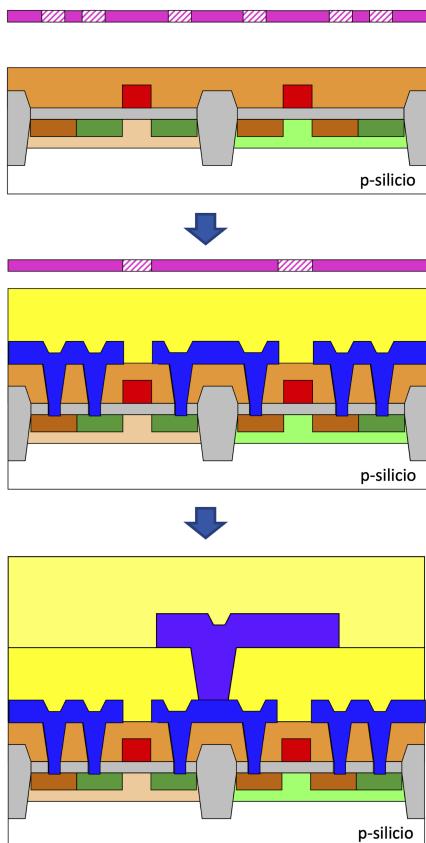
Le diffusioni N+ e P+ servono a formare le regioni di source, drain e body dei mosfet. Il processo prevede i seguenti passaggi:



1. si selezionano le aree in cui eseguire le diffusioni N+ (source e drain degli NMOS, body dei PMOS) tramite il processo di litografia selettiva
2. si bombarda il silicio con ioni di arsenico o fosforo attraverso il processo di impiantazione ionica con attivazione del drogaggio per annealing utilizzando per creare le regioni N+ nelle aree scoperte, infine si rimuove il fotoresist rimanente
3. si ripete il processo analogo di selezione delle aree, impiantazione ionica e attivazione del drogaggio per creare le regioni P+ (source e drain dei PMOS, body degli NMOS)

11.8 Contatti e interconnessioni

Dopo aver creato i mosfet, è necessario creare i terminali elettrici e le interconnessioni tra i vari componenti del circuito integrato. Il processo prevede i seguenti passaggi:

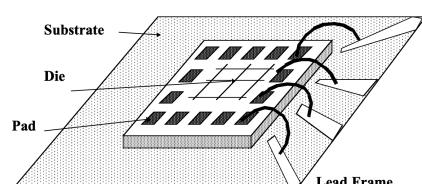


1. si deposita uno strato di ossido isolante (arancione) sopra tutta la superficie del wafer per isolare la struttura dei mosfet dalle piste metalliche di interconnessione costruite sopra
2. si utilizza un processo di litografia selettiva per forare l'ossido e raggiungere le aree di silicio dei terminali dei mosfet (source, drain, body e gate) dove vanno creati i contatti;
3. si deposita uno strato di metallo (ad esempio alluminio o rame) che riempie i fori sempre attraverso un processo di litografia selettiva in modo da formare i terminali del mosfet
4. si riveste l'intera superficie con un altro strato di ossido isolante (giallo) per separare i vari livelli di interconnessione
5. si riesegue l'intero processo di foratura e deposizione del metallo e deposizione del dielettrico più volte per creare i vari livelli di interconnessione necessari a collegare tra loro i vari componenti del circuito

11.9 Packaging

Il processo di packaging consiste nel preparare il die per l'utilizzo esterno al fine di poterlo collegare ad altri circuiti (ad esempio su una scheda madre) e proteggerlo da agenti esterni. Il processo prevede i seguenti passaggi:

1. si taglia il wafer in singoli die tramite una sega a filo abrasivo di diamante
 2. si monta ogni die in un contenitore protettivo (package) che può essere di plastica o metallo
 3. si collegano i terminali del die (detti pad) ai terminali esterni del package tramite fili sottili di oro
- Alcuni esempi di package sono illustrati in figura a destra



interconnessioni tra i pad del die e i terminali del package



Through-Hole Surface-Mount Ball Grid Array

11.10 Layout, regole di layout e sviluppo delle maschere

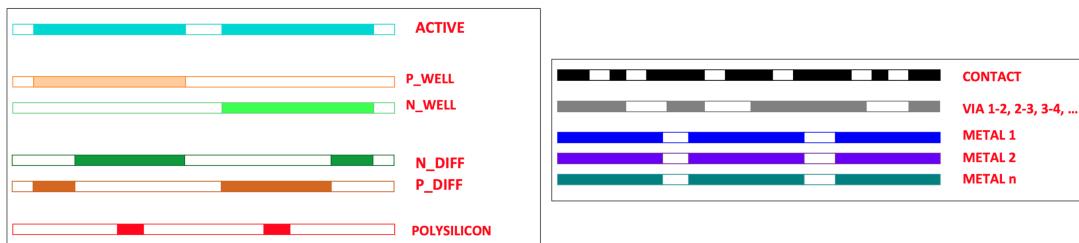
Definizione del layout

Il layout è la rappresentazione grafica in scala del circuito integrato che mostra la disposizione spaziale dei vari componenti con le loro dimensioni e le interconnessioni tra di essi. Il layout è composto da una serie di maschere che verranno utilizzate nel processo di costruzione del circuito integrato.

Il layout viene realizzato dal progettista del circuito integrato e viene poi usato dal tecnico di fabbricazione nelle varie fasi di costruzione del circuito integrato. Funge da linguaggio comune tra progettista e tecnico di fabbricazione.

Ad ogni maschera è associata una fase del processo di costruzione del circuito integrato:

- active: definizione delle regioni attive
- p-well/n-well: selezione del tipo di substrato
- n-diff/p-diff: diffusioni N+ e P+
- polysilicon: ossido di gate ed elettrodo di gate
- contact: deposizione del metallo per i terminali dei mosfet
- metal1, metal2, ... : deposizione del metallo per i vari livelli di interconnessione
- via 1-2, via 2-3, ... : foratura del dielettrico tra i vari livelli di interconnessione



esempio di layout (maschere) per la realizzazione di un inverter CMOS, come illustrato nelle fasi di fabbricazione precedenti, a sinistra le FEOL e a destra le BEOL

Regole di layout

Affinché il layout possa essere effettivamente utilizzato per la costruzione del circuito integrato, deve rispettare una serie di regole per garantire la corretta realizzazione fisica del circuito integrato. Le regole di layout sono dovute a:

- minima risoluzione e tolleranza della fotolitografia
- inevitabile disallineamento delle maschere
- imprecisioni del processo
- buon senso del progettista

Le regole di layout si dividono in:

- **intra-layer**: regole che riguardano una singola maschera, ovvero le dimensioni e le distanze minime tra gli elementi di una singola maschera (per rispettare la risoluzione e le tolleranze del processo)
- **inter-layer**: regole che riguardano più maschere, ovvero le distanze minime che gli elementi di maschere diverse devono rispettare tra di loro (per contenere problemi di disallineamento)

12 Blocchi logico aritmetici

12.1 Struttura di una ALU

Una alu è composta da vari blocchi semplici tra cui:

- blocchi di traslazione e rotazione
- sommatore/comparatore
- moltiplicatore
- operatori logici
- logica di controllo (registri)

12.2 Operatori logici elementari

Gli operatori logici elementari svolgono una funzione logica bit a bit tra due operandi di n bit ciascuno. Le funzioni logiche elementari sono ad esempio AND, OR, XOR, NOT.

12.3 Blocchi di traslazione e rotazione

Traslazione

La traslazione è un'operazione che sposta tutti i bit a destra o a sinistra di un certo numero di posizioni:

- lo spostamento a sinistra equivale a moltiplicare per 2, le posizioni vuote vengono riempite con 0
- lo spostamento a destra equivale a dividere per 2, le posizioni vuote vengono riempite con 0 (divisione senza estensione di segno) o con il bit di segno (divisione con estensione di segno)

Rotatore

La rotazione è un'operazione che sposta tutti i bit a destra o a sinistra di un certo numero di posizioni, ma i bit che escono da un lato rientrano dall'altro.

Rotatore/traslatore con MUX a struttura logaritmica ottimizzato a TG

Un rototraslatore con spostamento di un generico numero di bit n viene implementato con una serie di stadi in numero logaritmico rispetto a n . Ogni stadio effettua uno spostamento di 2^k bit (con k indice dello stadio da 0 a $\log_2(n) - 1$). Siccome ogni numero intero può essere scritto come somma di potenze di 2, è possibile combinare gli stadi opportunamente per ottenere lo spostamento di un generico numero intero di posizioni.

Ogni stadio è implementato con un parallelo di MUX 2:1 controllati da un segnale di controllo S_i che seleziona se lasciare passare il bit in ingresso (senza spostamento) oppure se trasmettere il bit spostato di 2^i posizioni. Ogni MUX 2:1 è implementato a porte TG ed eventualmente si aggiungono due invertitori CMOS all'ingresso e all'uscita per non degradare il segnale e mantenere i tempi di propagazione bassi.

Stima dei MOS impiegati

traslatore a n bit	n bit	8 bit	16 bit	32 bit	64 bit
# di stadi	$\log_2 n$	3	4	5	6
# di TG	$2n \cdot \log_2 n$	48	128	320	768
# di NOT per input/output	$2n$	16	32	64	128
# di NOT segnali di controllo	$\log_2 n$	3	4	5	6
totale MOS	$2 \cdot (2n \log_2 n + 2n + \log_2 n)$	134	328	778	1804

12.4 Sommatore ripple carry

Half Adder

Il blocco sommatore half adder somma due bit A e B e produce in uscita una somma $S = A \oplus B$ e un riporto $C = A \cdot B$. Esistono due tipi di implementazione:

- implementazione a MUX con TG (con invertitori CMOS in uscita):
 - le funzioni logiche XNOR e NAND (XNOR + NOT in uscita = NOR e NAND + NOT in uscita = AND) sono implementate a MUX con porte TG
 - il numero di transistor necessari è $2_{\text{MOS/NOT}} \cdot 4_{\text{NOT}} + 4_{\text{MOS/MUX}} \cdot 2_{\text{MUX}} = 16$ i tempi di ritardo delle due linee sono pressoché identici $t_p = t_{p,\text{NOT}} + t_{p,\text{TG}} + t_{p,\text{NOT}}$
- implementazione a porte logiche CMOS:
 - la funzione logica AND è implementata con 6 MOS (NAND + NOT), la funzione logica XOR è implementata con 8 MOS (NOR tra NOR e AND), per un totale di 14 MOS
 - i tempi di ritardo delle due linee sono diversi in quanto per calcolare la somma è necessario prima avere calcolato il riporto $t_{p,S} = t_{p,\text{NAND}} + t_{p,\text{NOT}} + t_{p,\text{NOR}} > t_{p,C} = t_{p,\text{NAND}} + t_{p,\text{NOT}}$

Full Adder

Il blocco sommatore full adder somma due bit A e B con un riporto in ingresso C_{in} e produce in uscita una somma $S = A \oplus B \oplus C_{in}$ e un riporto in uscita $C_{out} = A \cdot (\overline{A} \oplus B) + C_{in} \cdot (A \oplus B)$. Si osserva che è possibile riscrivere le equazioni in funzione delle variabili P e \overline{P} ottenendo:

$$S = P \oplus C_{in} \quad C_{out} = A \cdot \overline{P} + C_{in} \cdot P \quad \text{con } P = A \oplus B \quad \overline{P} = \overline{A \oplus B}$$

Si implementano quindi la XOR e la XNOR per P e \overline{P} con architettura ibrida per un totale 4 mos ciascuna, la somma con una XNOR a MUX con TG e una NOT di uscita (8 mos) e il riporto con un MUX con TG e una NOT di uscita (8 mos) per un totale di 24 mos.

Sommatore a n bit in Ripple Carry

Per implementare un generico sommatore tra operandi a n bit (in struttura ripple carry) si collegano in cascata n FA in cui il riporto in uscita di ogni FA viene collegato al riporto in ingresso del FA successivo. Il primo FA riceve come riporto in ingresso il valore 0 (o altri valori se il sommatore viene usato per altri scopi). Il tempo di propagazione cresce linearmente con il numero di bit degli operandi in quanto l'ultimo FA deve aspettare il riporto dall'FA precedente:

$$t_{tot,sum} = (n - 1) \cdot t_{carry} + t_s \quad t_{tot,carry} = n \cdot t_c$$

12.5 Varianti del sommatore

Sottrattore

Si osserva che la sottrazione tra due numeri può essere vista come la somma del primo numero con il complemento a 2 del secondo numero: $A - B = A + \overline{B} + 1$. È possibile quindi riciclare il blocco sommatore aggiungendo uno stadio di invertitori CMOS controllati a MUX con TG per eventualmente invertire i bit del secondo operando B e impostando il riporto in ingresso del sommatore a 1 per il “+1”. Ogni blocco invertitore è implementato con un MUX a porte TG, una NOT di uscita e una NOT per invertire un segnale di ingresso (8 mos ciascuno), in più è necessario una NOT per il segnale di controllo (2 mos). In totale si aggiungono $8n + 2$ mos al sommatore.

Comparatore

Il comparatore generico confronta due numeri A e B e ha due possibili implementazioni:

1. struttura ad albero con XOR/XNOR (uguaglianza):
 - si osserva che $A = B$ corrisponde ad una XNOR bit a bit, per cui è possibile implementare il comparatore come prodotto di XNOR oppure come somma di XOR (con struttura ad albero per AND e OR a tanti ingressi); in questo modo si ottiene un segnale di uscita EQ ad 1 se i due numeri sono uguali, 0 altrimenti.

2. struttura basata sul sommatore (maggior/minore/uguaglianza):

si osserva che facendo la sottrazione $A - B$ si ottiene un risultato positivo se $A > B$, negativo se $A < B$ e zero se $A = B$. Si può quindi implementare il comparatore utilizzando un sommatore in cui il bit più significativo del risultato è chiamato CMP o COMPARE (0 se $A > B$, 1 se $A < B$) e un blocco NAND (ad albero) per il segnale EQ (0 se $A \neq B$, 1 se $A = B$).

Stima dei MOS impiegati per un sommatore + sottrattore + comparatore

add/sub/cmp a n bit	n bit	8 bit	16 bit	32 bit	64 bit
# di MOS ripple carry	24n	192	384	768	1536
# di MOS sottrattore	$8n + 2$	66	130	258	514
# di MOS comparatore	$\approx 2.5n$	28	42	84	128
totale MOS	$\approx 34.5n$	286	556	1110	2178

12.6 Moltiplicatore

Analizzando il processo di moltiplicazione di due numeri (come viene insegnato alle elementari) si osserva che è possibile scomporre la moltiplicazione in due fasi:

1. calcolo dei prodotti parziali $B_0A_0, B_0A_1, B_0A_2 \dots B_0A_n, B_1A_0, B_1A_1, \dots B_1A_n, \dots B_nA_n$; per tale processo sono richieste n^2 AND bit a bit implementate con logica statica complementare per un totale di $6n^2$ MOS
2. somma dei prodotti parziali delle $n-1$ colonne (la prima non ha bisogno di somme) opportunamente shiftati attarverso una matrice di HA e FA strutturata a cascata:
 - la prima riga è composta solo da HA in quanto non ci sono riporti in ingresso delle colonne a sinistra e richiede n HA per un totale di $16n$ MOS
 - le righe successive sono composte da FA e terminano con un HA in quanto l'ultimo bit è il primo che inizia la somma dei prodotti parziali; di conseguenza si hanno $n-2$ righe con $n-1$ FA e un HA ciascuna per un totale di $(n-2) \cdot (24(n-1) + 16) = 24n^2 - 56n + 16$ MOS
 - l'ultima riga serve per sommare i riporti ed equivale ad un sommatore senza riporto d'ingresso composto da $n-2$ FA e 2 HA per un totale di $24(n-2) + 2 \cdot 16 = 24n - 16$ MOS

Stima del numero di MOS impiegati

Si osserva che il numero totale di MOS impiegati cresce quadraticamente con il numero di bit degli operandi, l'elevato numero di mos è dovuto principalmente alla matrice di HA e FA necessaria per sommare i vari prodotti parziali. Nei primi processori, siccome si avevano limitazioni tecniche sul numero massimo di transistor per singolo chip, i blocchi moltiplicatori non venivano implementati via hardware.

moltiplicatore a n bit	n bit	8 bit	16 bit	32 bit	64 bit
# di MOS prodotti parziali	$6n^2$	384	1536	6144	24576
# di MOS somma prodotti parziali	$24n^2 - 16n$	1408	5888	24064	97280
totale MOS	$30n^2 - 16n$	1792	7424	30208	121856

NOTA: il prof. nelle slides ha usato il blocco sommatore finale con n FA, mentre l'architettura prevede un sommatore ottimizzato con $n-2$ FA e 2 HA, di conseguenza il numero di MOS è inferiore di 16 unità rispetto alle slides del prof. Tale differenza ovviamente è irrilevante e non cambia l'ordine di grandezza.

Considerazioni sul tempo di ritardo

Il tempo di ritardo del moltiplicatore è dominato dal tempo di ritardo della somma dei prodotti parziali, che cresce linearmente con il numero di bit degli operandi (dipende dal numero di righe della matrice ovvero a $n-1$), infatti ogni riga deve aspettare le somme e i riporti della riga precedente:

$$t_{tot,mul} = t_{and} + (n-1) \max(t_c, t_s) + t_{sum} \quad t_{sum} = t_{\text{sommatore ripple carry}} = (n-1)t_c + t_s$$

12.7 Sommatore (e sottrattore) floating point

Accenni alla rappresentazione floating point (senza casi particolari)

$$(-1)^S \cdot 2^{E-K} \cdot (1.M)$$

- singola precisione (32 bit): 1 bit segno S , 8 bit esponente E , 23 bit mantissa M , $k = 127$
- doppia precisione (64 bit): 1 bit segno S , 11 bit esponente E , 52 bit mantissa M , $k = 1023$

Algoritmo di somma di numeri floating point

1. **costruzione della mantissa:** si costruisce la mantissa completa aggiungendo il bit隐含的 1 all'inizio
2. **classificazione degli operandi:** si confrontano gli esponenti e si classificano i due operandi in OP1 (con esponente maggiore) e OP2 (con esponente minore)
3. **allineamento della mantissa:** si effettua uno shift a destra della mantissa dell'OP2 di un numero di posizioni pari al modulo della differenza tra i due esponenti, in modo da allineare le mantisse allo stesso esponente
4. **somma/sottrazione delle mantisse:** si calcola la somma o la sottrazione delle due mantisse, scelta dal blocco di gestione del segno, in caso di sottrazione si effettua il complemento a 2 della mantissa dell'OP2
5. **normalizzazione del risultato:** se il risultato è negativo se ne calcola il complemento a 2, infine si normalizza il risultato shiftando a destra o a sinistra la mantissa per avere il primo bit a 1 e si modifica l'esponente di conseguenza; il calcolo del numero di posizioni da shiftare (numero di zeri iniziali) è eseguito dal *leading zero counter* o LZC, mentre il segno del risultato è dato dall'opposto del riporto in uscita dalla somma/sottrazione delle mantisse
6. **calcolo dell'esponente:** si parte dall'esponente massimo (quello dell'OP1), comune alle due mantisse allineate, e si somma/sottrae il numero di posizioni di cui la mantissa viene shiftata nella fase di normalizzazione, calcolato dal LZC
7. **calcolo del segno:** il segno del risultato è calcolato dal blocco di gestione del segno in base ai segni degli operandi e all'operazione richiesta (somma o sottrazione)

Gestione del segno

Si osserva che:

- le mantisse sono sempre considerate numeri interi positivi unsigned, infatti il segno è gestito a parte nell'apposito bit di segno
- l'operazione di somma tra unsigned (positivi) restituisce un unsigned (positivo) e il riporto in uscita dal sommatore indica se è avvenuto overflow ($C_{out} = 1$) oppure no ($C_{out} = 0$)
- l'operazione di sottrazione tra unsigned richiede di convertire il sottraendo nel suo complemento a 2, il riporto in uscita indica se il risultato è positivo ($C_{out} = 1$) oppure negativo ($C_{out} = 0$)

Si possono quindi riassumere le regole per la gestione del segno nella tabella seguente:

add - $OP = 1$	S_A	S_B	S_C	S/\bar{A}	diff - $OP = 0$	S_A	S_B	S_C	S/\bar{A}
$A + B$	0	0	0	0 - add	$A - B$	0	0	$\overline{C_{out}}$	1 - sub
$A + (-B)$	0	1	$\overline{C_{out}}$	1 - sub	$A - (-B)$	0	1	0	0 - add
$(-A) + B$	1	0	$\overline{C_{out}}$	1 - sub	$(-A) - B$	1	0	1	0 - add
$(-A) + (-B)$	1	1	1	0 - add	$(-A) - (-B)$	1	1	$\overline{C_{out}}$	1 - sub

Dalla tabella di verità sopra si ottengono le seguenti espressioni logiche:

$$S_C = S_A \cdot (OP \oplus S_B) + C_{out} \cdot (OP \oplus S_A \oplus S_B) \quad S/\bar{A} = OP \oplus S_A \oplus S_B$$

Il costo in MOS del blocco di gestione del segno secondo le funzioni logiche sopra è di 44 MOS.

Modulo di un numero intero

Il modulo di un numero intero a n bit equivale a calcolare il complemento a 2 di tale numero. Per fare ciò è necessario negare il numero utilizzando n invertitori CMOS (2 MOS ciascuno) e una serie di n HA (16 MOS ciascuno) per sommare 1 al numero negato e propagare il riporto per tutti i numeri. Per controllare meglio tutto ciò si aggiungono un parallelo di n MUX 2:1 con invertitore in uscita (6 MOS ciascuno) per abilitare o disabilitare gli invertitori. Il costo in MOS del modulo è quindi di $n \cdot (2 + 16 + 6) = 24n$ MOS.

Leading Zero Counter - LZC

Il leading zero counter (LZC) è un blocco che riceve in ingresso 4 bit x_3, x_2, x_1, x_0 e produce in uscita un numero binario a due cifre q_1, q_0 che rappresenta l'indice del primo bit a 1 (contando da sinistra a destra) e un bit a che vale 1 se tutti i bit in ingresso sono 0. Analizzando la tabella di verità si ottengono le seguenti espressioni logiche con un costo complessivo del 4-LZC di 22 MOS.

$$q_1 = \overline{x_3 + x_2} \quad (4 \text{ MOS}) \quad q_0 = \overline{x_3 + \overline{x_2} \cdot x_1} \quad (2 + 6 \text{ MOS}) \quad a = x_3 + x_2 + x_1 + x_0 \quad (8 + 2 \text{ MOS})$$

Siccome la mantissa di un numero floating point è più lunga di 4 bit, vengono combinati più LZC a 4 bit le cui coppie di uscite q_1, q_0 vanno in un mux controllato dalle uscite q_0, q_1, \dots di altri 4-LZC che ricevono in input i segnali a dei LZC precedenti per selezionare il primo LZC che ha trovato un bit a 1.

Stima dei MOS per un sommatore floating point

Sommando i vari contributi di tutti i componenti si ottengono approssimativamente i seguenti valori, osservando che la metà dei MOS sono impiegati nei sommatori (comparatore esponenti, sommatore esponenti, sommatore/sottrattore mantisse e moduli) e un terzo nei traslatori (allineamento mantisse e normalizzazione risultato).

componente	singola precisione	doppia precisione
totale	4819 mos	9394 mos
di cui in sommatori:	2329 mos	4206 mos
di cui in traslatori:	1556 mos	3160 mos

12.8 Moltiplicatore floating point

Il moltiplicatore floating point segue un algoritmo più semplice rispetto al sommatore in quanto la notazione floating point segno-mantissa-esponente è già una moltiplicazione:

$$A = (-1)^{S_A} \cdot 2^{E_A-K} \cdot (1.M_A) \quad B = (-1)^{S_B} \cdot 2^{E_B-K} \cdot (1.M_B) \quad C = (-1)^{S_C} \cdot 2^{E_C-K} \cdot (1.M_C)$$

$$C = A \cdot B = (-1)^{S_A \oplus S_B} \cdot 2^{E_A+E_B-K} \cdot (1.M_A \times 1.M_B)$$

L'algoritmo di moltiplicazione floating point prevede quindi i seguenti passi:

1. **costruzione della mantissa:** come nella somma, si aggiunge il bit 1 all'inizio
2. **calcolo del segno:** si calcola il segno del risultato come XOR tra i segni degli operandi
3. **somma degli esponenti:** si sommano gli esponenti e si sottrae la costante K
4. **moltiplicazione delle mantisse:** si moltiplicano le mantisse degli operandi con un moltiplicatore
5. **normalizzazione del risultato:** si normalizza il risultato shiftando a destra o a sinistra la mantissa per avere il primo bit a 1 e si modifica l'esponente di conseguenza; il calcolo del numero di posizioni da shiftare (numero di zeri iniziali) è eseguito dal *leading zero counter* o LZC

Stima del numero di transistor

Si osserva che il numero totale di MOS impiegati (nel moltiplicatore a doppia precisione) è dovuto in grandissima parte al moltiplicatore delle mantisse (più specificatamente alla matrice delle somme parziali):

totale doppia precisione:	85474 mos
nel moltiplicatore:	83438 mos
nella matrice di somme parziali:	82166 mos

13 Ottimizzazione del sommatore

13.1 Ottimizzazione del full adder con generate e propagate

Propagate, Generate-1 e Generate-0 o Delete

Analizzando la propagazione del riporto in un sommatore ripple carry si osservano tre casi:

1. **Propagate (P):**

quando la somma dei bit $A_i + B_i = 1$ (ovvero $A_i = 0, B_i = 1$ oppure $A_i = 1, B_i = 0$) il riporto in ingresso C_i viene propagato in uscita C_{i+1} (ovvero $C_{i+1} = C_i$)

2. **Generate-1 (G1):**

quando $A_i = B_i = 1$ il riporto in uscita C_{i+1} è forzatamente 1 indipendentemente dal valore di C_i del riporto in ingresso (ovvero $C_{i+1} = 1$)

3. **Generate-0 o Delete (G0):**

quando $A_i = B_i = 0$ il riporto in uscita C_{i+1} è forzatamente 0 indipendentemente dal valore di C_i del riporto in ingresso (ovvero $C_{i+1} = 0$)

Si generano quindi 2 segnali di controllo per ogni bit i (si igora $G0$):

$$P = A_i \oplus B_i \quad G = A_i \cdot B_i$$

Struttura ottimizzata del full adder - tempi di ritardo, numero di MOS e capacità logica

È possibile riscrivere le equazioni logiche delle uscite full adder in funzione dei segnali P e G :

$$S = P \oplus C_{\text{in}} \quad C_{\text{out}} = G + P \cdot C_{\text{in}}$$

Si suddivide quindi il full adder in tre sezioni:

1. **setup:** generazione dei segnali di P e G con una XOR e una AND

2. **CGP (Carry Generation and Propagation):** calcolo del riporto C_{out} con una AND e una OR

3. **somma:** calcolo della somma S con una XOR

Si osserva che per il calcolo del riporto e della somma, l'ultimo segnale ad essere disponibile (detto segnale critico) è il segnale del riporto in ingresso, infatti i segnali P e G sono già pronti all'arrivo del riporto del FA precedente. I tempi di ritardo di questi due blocchi non dipendono quindi dal tempo di ritardo del setup. Si ha quindi:

$$t_{\text{setup}} = \max(t_{\text{xor}}, t_{\text{and}}) \quad t_c = t_{\text{or}} + t_{\text{and}} \quad t_s = t_{\text{xor}}$$

Implementando il full adder con XOR ottimizzate e utilizzando porte TG, si utilizzano 10 MOS per il setup, 8 MOS per il CGP e 6 MOS per la somma, per un totale di 24 MOS e una capacità logica complessiva di $7.75C$, dove C è la capacità di gate/drain/source di una coppia di NMOS e PMOS.

13.2 Architettura Ripple Carry con FA ottimizzato

Struttura

La struttura di un sommatore ripple carry con FA ottimizzato è identica a quella del sommatore ripple carry classico, con la differenza che ogni FA è diviso nelle tre sezioni e il percorso del segnale critico del carry è accorciato.

Tempi di ritardo

Il tempo di ritardo quando il riporto deve propagarsi per tutti i FA del sommatore è:

$$t_{\text{carry,rc}} = t_{\text{setup}} + n \cdot t_c \quad t_{\text{sum,rc}} = t_{\text{setup}} + (n - 1) \cdot t_c + t_s$$

Rispetto ad un sommatore con struttura tradizionale con tempo di ritardo $t_{\text{sum,rc}} = (n - 1) \cdot t_c + t_s$, il vantaggio è che il tempo t_c , che domina il comportamento lineare, è minore nella architettura ottimizzata in quanto il calcolo del riporto è stato semplificato e velocizzato precalcolando i segnali P e G in anticipo.

Il tempo di ritardo calcolato per una specifica somma in questione è dato dalla catena di propagazione più lunga, ovvero quando tutti i FA della catena devono attendere il riporto del FA precedente.

Numero di MOS

Utilizzando FA ottimizzati con 24 MOS ciascuno, il numero totale di MOS diventa:

$$N_{\text{MOS,rc}} = n \cdot 24$$

Consumo/capacità logica

La capacità logica del sommatore ripple carry con FA ottimizzato (approssimando) è:

$$C_{\text{L,rc}} = n \cdot 8C$$

13.3 Architettura Carry Bypass - CBP

Struttura

L'architettura Carry Bypass raggruppa i FA in k gruppi di m FA ciascuno (con $n = k \cdot m$), e si aggiunge un multiplexer che permette di bypassare il calcolo del riporto all'interno del gruppo se tutti i FA del gruppo sono in condizione di propagate (ovvero $P_i = 1$ per ogni FA i del gruppo). In questo modo il riporto in ingresso del gruppo può essere propagato in un t_{mux} direttamente al blocco successivo. Il numero ottimale di FA per gruppo m è dato dalla seguente formula:

$$m_{\text{opt}} = \sqrt{n \cdot \frac{t_{\text{mux}}}{2t_c}} \quad \text{per } t_c = t_{\text{mux}} \rightarrow m_{\text{opt}} = \sqrt{n/2} \quad \begin{array}{ll} n = 8 \rightarrow m = 2 & n = 16 \rightarrow m = 3, 4 \\ n = 32 \rightarrow m = 4 & n = 64 \rightarrow m = 5, 6 \end{array}$$

Tempi di ritardo

Il tempo di ritardo quando il riporto deve propagarsi per tutti i FA del sommatore è:

$$\begin{aligned} t_{\text{carry,cbp}} &= t_{\text{setup}} + m \cdot t_c + k \cdot t_{\text{mux}} & t_{\text{sum,cbp}} &= t_{\text{setup}} + m \cdot t_c + (k - 1) \cdot t_{\text{mux}} + (m - 1) \cdot t_c + t_s \\ & & &= t_{\text{setup}} + (2m - 1) \cdot t_c + (k - 1) \cdot t_{\text{mux}} + t_s \end{aligned}$$

Si osserva che conviene usare un sommatore CBP con $m = 4$ rispetto ad un RC quando i numeri da sommare sono più grandi di 8 bit.

Numero di MOS

Rispetto ad un RC, si aggiungono k MUX 2:1 con invertitori in ingresso e uscita (10 MOS) e una porta NAND+NOT per calcolare la condizione di propagate del gruppo (10 MOS) per un totale di 20 MOS aggiuntivi per blocco. Il numero totale di MOS (per gruppi da 4 FA) diventa quindi:

$$N_{\text{MOS,cbp}} = n \cdot 24 + k \cdot 20 = n \cdot (24 + 20/m) = n \cdot 29$$

Consumo/capacità logica

Rispetto ad un RC, si aggiunge la capacità logica dei k MUX 2:1 ($2.5C$ ciascuno), mentre si può trascurare quella della porta NAND+NOT essendo molto piccola. Si ha quindi (per gruppi da 4 FA):

$$C_{L,cbp} = n \cdot 8C + k \cdot 2.5C = n \cdot (8 + 2.5/m)C = n \cdot 8.625C$$

13.4 Architettura Linear Carry Select - LCS

Struttura

L'idea alla base del sommatore Carry Select è di calcolare in parallelo due somme per ogni blocco di FA, una assumendo che il riporto in ingresso sia 0 e l'altra assumendo che sia 1, e poi selezionare il risultato corretto con un MUX 2:1 in base al riporto effettivo in ingresso al blocco. Anche in questo caso si suddivide il sommatore in k blocchi di m FA ciascuno (con $n = k \cdot m$). La scelta ottimale di m è simile a quella del CBP:

$$m_{\text{opt}} = \sqrt{n \cdot \frac{t_{\text{mux}}}{t_s}} \quad \text{per } t_s = t_{\text{mux}} \rightarrow m_{\text{opt}} = \sqrt{n} \quad \begin{array}{ll} n = 8 \rightarrow m = 2, 3 & n = 16 \rightarrow m = 4 \\ n = 32 \rightarrow m = 5, 6 & n = 64 \rightarrow m = 8 \end{array}$$

Tempi di ritardo

Il tempo di ritardo si divide in tre fasi: il precalcolo dei riporti, la propagazione del riporto e il calcolo della somma. Nel caso in cui il riporto deve propagarsi per tutti i FA del sommatore, il tempo di ritardo è:

$$t_{\text{carry,lcs}} = (t_{\text{setup}} + m \cdot t_c)_{\text{precalcolo}} + (k \cdot t_{\text{mux}})_{\text{propagazione}} \quad t_{\text{sum,lcs}} = t_{\text{setup}} + m \cdot t_c + k \cdot t_{\text{mux}} + t_s$$

Si osserva che conviene usare un sommatore LCS con $m = 4$ rispetto ad un RC per numeri più grandi di 8 bit e con $m = 8$ rispetto ad un CBP per numeri più grandi di 16 bit.

Numero di MOS

Rispetto ad un RC, si raddoppia lo stadio di CGP ($8n$) e si aggiungono n MUX 2:1 ($10n$) per arrivare a:

$$N_{\text{MOS,lcs}} = 24n + 8n + 10n = 42n$$

Consumo/capacità logica

Anche per la capacità logica si raddoppia lo stadio di CGP ($2.5nC$) e si aggiungono n MUX 2:1 ($2.5nC$) per un totale di:

$$C_{L,lcs} = 8nC + 2.5nC + 2.5nC = 13nC$$

13.5 Architettura Square Root Carry Select - SRCS

Struttura

Si osserva che nel LCS descritto sopra, i riporti per ogni blocco vengono calcolati in parallelo e sono pronti tutti assieme, ma il segnale del riporto, prima di arrivare all'ultimo blocco, deve attraversare tutti i MUX precedenti. L'ultimo blocco, quindi, rimane in attesa finché il segnale del riporto attraversi tutti i MUX. Per ridurre il tempo perso, si è pensato di incrementare progressivamente di un FA dimensione dei blocchi di FA in modo che, mentre il segnale si propaga nel MUX, il blocco successivo (al posto di rimanere in attesa) calcola la somma anche per il FA extra. In questo modo, si riesce a ridurre il numero di blocchi e di conseguenza il tempo di ritardo totale (anche se non è sempre detto).

Lo SRCS si basa proprio su questa idea, suddividendo il sommatore in k blocchi di FA di dimensione crescente partendo da un m piccolo (anche 2) fino a $m+k-1$. Il numero totale di bit sommabili è quindi:

$$n = k \cdot m + \frac{k(k-1)}{2} = \frac{k^2}{2} + k \left(m - \frac{1}{2} \right) \quad \text{per } k \gg m \quad k \approx \sqrt{2n}$$

Tempi di ritardo

Il tempo di ritardo riprende le formule del LCS, ma con $k \approx \sqrt{2n}$:

$$t_{\text{carry,srcs}} = t_{\text{setup}} + m \cdot t_c + \sqrt{2n} \cdot t_{\text{mux}} \quad t_{\text{sum,srcs}} = t_{\text{setup}} + m \cdot t_c + \sqrt{2n} \cdot t_{\text{mux}} + t_s$$

Si osserva che conviene usare un sommatore SRCS con $m = 2$ rispetto a qualsiasi altro sommatore per numeri più grandi di 8 bit.

Numero di MOS

Il numero di MOS non cambia rispetto al LCS, in quanto le architetture sono molto simili e rimane $42n$.

Consumo/capacità logica

Anche la capacità logica non cambia rispetto al LCS, rimanendo $13nC$.

13.6 Architettura Carry Look Ahead - CLA

Formula completa del calcolo del riporto

L'architettura Carry Look Ahead si basa sull'idea di calcolare in anticipo tutti i riporti necessari per la somma, in modo da non doverli propagare attraverso i FA. Si osserva che il riporto in uscita di ogni FA dipende dai segnali P e G dei FA precedenti e dal riporto in ingresso C_0 :

$$\begin{aligned} C_0 &= G_0 + P_0 C_{\text{in}} \\ C_1 &= G_1 + P_1 C_0 = G_1 + P_1(G_0 + P_0 C_{\text{in}}) = G_1 + P_1 G_0 + P_1 P_0 C_{\text{in}} \\ C_2 &= G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_{\text{in}} = G_{0:2} + P_{0:2} C_{\text{in}} \\ C_3 &= G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_{\text{in}} = G_{0:3} + P_{0:3} C_{\text{in}} \\ C_n &= G_{0:n} + P_{0:n} C_{\text{in}} \end{aligned}$$

Carry Lookahead Unit e segnali P e G di blocco

Si osserva che sarebbe possibile calcolare tutti i riporti attraverso una rete combinatoria in singolo stadio, ma dall'elevato numero di porte necessarie (soprattutto per numeri grandi) si avrebbe un enorme tempo di ritardo e si preferisce quindi suddividere il calcolo in più stadi gerarchici. Si sviluppa un componente chiamato CLU (Carry Lookahead Unit) che riceve in ingresso i segnali $P_{m:k}$, $G_{m:k}$ e $P_{k+1:n}$, $G_{k+1:m}$ di due blocchi $[m : k]$ e $[k + 1 : n]$ e calcola in output i segnali $P_{m:n}$ e $G_{m:n}$ del blocco unito $[m : n]$ secondo la seguente formula:

$$G_{m:n} = G_{k+1:n} + G_{m:k} \cdot P_{k+1:n} \quad P_{m:n} = P_{m:k} \cdot P_{k+1:n}$$

La prima espressione sopra equivale a dire che il generate del blocco $[m : n]$ $G_{m:n}$ equivale ad avere un generate nel blocco $[k + 1 : n]$ $G_{k+1:n}$, oppure ad avere un generate nel blocco $[m : k]$ $G_{m:k}$ e che il riporto venga propagato attraverso il blocco $[k + 1 : n]$ $P_{k+1:n}$.

La CLU è implementata in logica statica complementare per ridurre i tempi di ritardo (altrimenti si avrebbero reti di Elmore molto lunghe) con 14 MOS per unità.

Albero di Kogge-Stone per il calcolo dei riporti

Basandosi sulla struttura dell'albero di Kogge-Stone con le CLU per ogni nodo, si è in grado di calcolare tutti i segnali P e G con un numero di stadi logaritmico rispetto al numero di bit sommabili n . In questo modo si calcolano tutti i riporti in parallelo in quanto dipendono solo dai segnali P e G e dal riporto in ingresso C_{in} , senza doverli propagare attraverso i FA.

Struttura

La struttura di un CLA si compone di 4 “piani”:

1. calcolo di setup: calcolo dei segnali P_i e G_i per ogni coppia di bit A_i e B_i come nei FA ottimizzati
2. albero di CLU: calcolo dei segnali $P_{0:i}$ e $G_{0:i}$ per ogni blocco di bit, attraverso le CLU disposte ad albero di Kogge-Stone con profondità di $\log_2 n$
3. calcolo dei riporti: avviene in parallelo per ogni bit secondo la formula $C_n = G_{0:n} + P_{0:n}C_{in}$
4. calcolo delle somme: avviene in parallelo per ogni bit secondo la formula $S_i = P_i \oplus C_i$

Per contenere i tempi di ritardo, i vari moduli (setup, CLU, riporti e somme) sono implementati in logica statica complementare in modo da non avere lunghe reti di Elmore. Il numero di MOS per ogni modulo di setup, CLU, riporto e somma è rispettivamente 14, 14, 8, 6 MOS.

Tempi di ritardo

I tempi di ritardo al caso peggiore equivalgono al tempo impiegato a calcolare il riporto in uscita e la somma per l’ultimo bit più significativo:

$$t_{sum,cla} = t_{setup} + \log_2 n \cdot t_{clu} + t_c \quad t_{sum,cla} = t_{setup} + \log_2 n \cdot t_{clu} + t_c + t_s$$

Numero di MOS

Il numero totale di MOS impiegati in un sommatore CLA è dato dalla somma dei MOS dei vari moduli:

$$N_{MOS,cla} = n \cdot 14_{setup} + (n \log_2 n - n + 1) \cdot 14_{clu} + n \cdot 8_{riporti} + n \cdot 6_{somme} = 14 \cdot (n \log_2 n + n + 1)$$

Consumo/capacità logica

La capacità logica complessiva del sommatore CLA è sicuramente la più elevata dato il numero di MOS impiegati.

13.7 Architetture a confronto

Tempi di ritardo:

architettura	tempo di ritardo - carry	tempo di ritardo - somma
Ripple Carry	$t_{setup} + n \cdot t_c$	$t_{setup} + (n - 1) \cdot t_c + t_s$
Carry Bypass	$t_{setup} + m \cdot t_c + k \cdot t_{mux}$	$t_{setup} + (2m - 1) \cdot t_c + (k - 1) \cdot t_{mux} + t_s$
Linear Carry Select	$t_{setup} + m \cdot t_c + k \cdot t_{mux}$	$t_{setup} + m \cdot t_c + k \cdot t_{mux} + t_s$
Square Root Carry Select	$t_{setup} + m \cdot t_c + \sqrt{2n} \cdot t_{mux}$	$t_{setup} + m \cdot t_c + \sqrt{2n} \cdot t_{mux} + t_s$
Carry Look Ahead	$t_{setup} + \log_2 n \cdot t_{clu} + t_c$	$t_{setup} + \log_2 n \cdot t_{clu} + t_c + t_s$

Numero di MOS

architettura	n bit	8 bit	16 bit	32 bit	64 bit
Ripple Carry	$24n$	192	384	768	1536
Carry Bypass	$29n$	232	464	928	1856
Linear Carry Select	$42n$	336	672	1344	2688
Square Root Carry Select	$42n$	336	672	1344	2688
Carry Look Ahead	$14n \log_2 n + 14n + 14$	462	1134	2702	6286

Consumo/capacità logica

architettura	n bit	8 bit	16 bit	32 bit	64 bit
Ripple Carry	$8nC$	62C	124C	248C	486C
Carry Bypass	$8.625nC$	67C	134C	268C	526C
Linear Carry Select	$13nC$	102C	204C	408C	816C
Square Root Carry Select	$13nC$	102C	204C	408C	816C
Carry Look Ahead	-	101C	223C	486C	1069C

13.8 Moltiplicatore con sommatore ottimizzato in architettura CLA

Un esempio di applicazione di un sommatore ottimizzato con architettura CLA è nell'ultima fase di somma dei riporti parziali in un moltiplicatore binario. In questo modo il tempo di ritardo complessivo, rimane sempre lineare (somme dei prodotti parziali), ma la pendenza della retta diminuisce:

$$t_{\text{mult,rc}} = t_{\text{and}} + (N - 1) \cdot \max(t_c, t_s) + n \cdot t_c + t_s \approx (2n + 1)T$$

$$t_{\text{mult,cla}} = t_{\text{and}} + (N - 1) \cdot \max(t_c, t_s) + t_{\text{setup}} + \log_2 n \cdot t_{\text{clu}} + t_c + t_s \approx (n + \log_2 n + 3)T$$

14 Latch e Flip Flop

14.1 Circuiti logici combinatori e sequenziali

Un circuito logico può essere:

- combinatorio: se l'uscita dipende solo dal valore attuale degli ingressi
- sequenziale: se l'uscita dipende anche dallo stato precedente del circuito (circuito con memoria)

I circuiti sequenziali possono essere:

- asincroni: se lo stato cambia in funzione degli ingressi in qualsiasi istante di tempo
- sincroni: se lo stato cambia solo in corrispondenza di un segnale di clock

14.2 Latch

Struttura generale

Un latch è un circuito sequenziale asincrono che memorizza un bit di informazione. L'informazione memorizzata è sempre disponibile all'uscita in forma diretta Q e negata \bar{Q} . L'informazione viene modificata in funzione di un segnale di abilitazione o clock EN/CLK e di un ingresso dati D .

Sincronizzazione dei latch

Il latch è uno strumento di memoria sensibile ai livelli del segnale di clock CLK (o di abilitazione EN). Un latch può essere di due tipi:

- **latch attivo alto**: quando il segnale di abilitazione è alto $EN = 1$, l'uscita Q segue l'ingresso dati D e il latch si dice trasparente. Quando il segnale di abilitazione è basso $EN = 0$, l'uscita Q mantiene il valore precedente e il latch è in modalità memoria.
- **latch attivo basso**: quando il segnale di abilitazione è basso $EN = 0$, l'uscita Q segue l'ingresso dati D e il latch si dice trasparente. Quando il segnale di abilitazione è alto $EN = 1$, l'uscita Q mantiene il valore precedente e il latch è in modalità memoria.

Per distinguere i due tipi di latch, si inserisce un pallino (di negazione) sul segnale di abilitazione del latch attivo basso.

Implementazione statica dei latch

L'implementazione statica dei latch si basa sul **principio di bistabilità** propria di un anello con un numero pari di invertitori CMOS. Tale circuito possiede tre punti di equilibrio:

- due punti di equilibrio stabili (bi-stabilità), corrispondenti ai due stati logici $Q = 0$ e $Q = 1$
- un punto di equilibrio instabile, corrispondente al valore logico intermedio tra i due stati logici, a cui basta una piccola perturbazione per far sì che il circuito si porti in uno dei due stati stabili

Data l'elevato guadagno degli invertitori CMOS, un grande disturbo attorno ai punti di equilibrio stabili altera di poco il valore di uscita, che tende a ritornare rapidamente al punto di equilibrio stabile, mentre un piccolo disturbo attorno al punto di equilibrio instabile fa sì che il circuito si porti rapidamente in uno dei due stati stabili. Tale proprietà consegue dalla proprietà rigenerativa degli stadi invertitori CMOS e l'immunità ai disturbi è: $NM = \min(V_M, V_{DD} - V_M)$.

Per facilitare la scrittura dei dati nel latch (aggirando la proprietà rigenerativa), si utilizza l'**architettura multiplexer** con due porte TG, una posizionata tra l'ingresso del dato e l'anello e l'altra posta tra i due invertitori. Sono controllate in maniera complementare dal segnale di abilitazione EN :

- quando $EN = 1$, la porta TG dell'ingresso è attiva e permette il passaggio del dato D verso l'anello, mentre la porta TG tra i due invertitori è inattiva, disabilitando l'azione rigenerativa dell'anello e permettendo così la scrittura del dato
- quando $EN = 0$, la porta TG dell'ingresso è inattiva, bloccando il passaggio del dato D verso l'anello, mentre la porta TG tra i due invertitori è attiva, abilitando l'azione rigenerativa dell'anello e permettendo così la memorizzazione del dato

Il **tempo di memorizzazione** di un latch è il tempo impiegato dal segnale in ingresso D per propagarsi attraverso la porta TG, il primo invertitore e il secondo invertitore fino all'uscita Q . In questo modo, all'inversione del segnale di abilitazione EN , l'anello si trova già nel giusto stato logico e non ci sono fluttuazioni indesiderate nell'uscita Q .

$$t_{\text{mem}} = t_p^{TG} + t_p^{INV1} + t_p^{INV2}$$

Al latch è possibile aggiungere un **segnale di reset asincrono** RST che forza l'uscita Q a zero indipendentemente dallo stato del segnale di abilitazione EN e dell'ingresso dati D . Viene implementato sostituendo il primo invertitore con una porta NAND (sempre rigenerativa) con due ingressi: uno collegato alle TG (ingresso e anello) e l'altro collegato al segnale di reset RST . In questo modo se $RST = 0$, l'uscita della porta NAND è forzata a 1 e tale valore viene propagato attraverso il secondo invertitore all'uscita Q portandola a 0. Quando $RST = 1$, la porta NAND si comporta come un invertitore normale. Per indicare la presenza del segnale di reset asincrono attivo basso, si inserisce un pallino (di negazione) sul segnale di reset RST . Per rendere il RST attivo alto, si utilizza una porta NOR al posto della porta NAND.

Al latch è possibile aggiungere un **segnale di set asincrono** SET che forza l'uscita Q a uno indipendentemente dallo stato del segnale di abilitazione EN e dell'ingresso dati D . Viene implementato allo stesso modo del segnale di reset RST , sostituendo il secondo invertitore con una porta NAND. Quando $SET = 0$, l'uscita della porta NAND è forzata a 1 e tale nodo coincide con l'uscita Q che risulta a 1. Quando $SET = 1$, la porta NAND si comporta come un invertitore normale. Per indicare la presenza del segnale di set asincrono attivo basso, si inserisce un pallino (di negazione) sul segnale di set SET . Per rendere il SET attivo alto, si utilizza una porta NOR al posto della porta NAND.

Il numero di transistor per un latch base è di 8 transistor (2 porte TG e 2 invertitori CMOS). Con l'aggiunta del segnale di reset asincrono RST o del segnale di set asincrono SET , il numero di transistor sale a 10. Con l'aggiunta di entrambi i segnali RST e SET , il numero di transistor sale a 12.

Implementazione dinamica dei latch

L'implementazione dinamica dei latch si basa sul principio di **carica e scarica di un condensatore** C_L pilotata da un invertitore CMOS controllato attraverso una porta TG controllata dal segnale di abilitazione EN . Quando il segnale di abilitazione è alto $EN = 1$, la porta TG è attiva e il condensatore C_L si scarica, portando l'uscita Q al valore logico opposto a quello dell'ingresso dati D (a causa dell'inversione dell'invertitore). Quando il segnale di abilitazione è basso $EN = 0$, la porta TG è inattiva e il condensatore C_L mantiene la carica, mantenendo stabile l'uscita Q .

Dato che il valore memorizzato nel condensatore C_L tende a scaricarsi nel tempo a causa delle correnti di perdita, è necessario un **refresh periodico** del dato memorizzato, che consiste nel riattivare periodicamente il segnale di abilitazione EN per ricaricare il condensatore C_L . Inoltre non gode della proprietà rigenerativa degli anelli con invertitori CMOS, per cui il valore memorizzato nel condensatore C_L è soggetto a disturbi e rumore.

Il vantaggio principale dell'implementazione dinamica dei latch è la riduzione del numero di transistor necessari per la realizzazione del circuito. Un latch dinamico richiede solo 4 transistor (1 porta TG e 1 invertitore CMOS) rispetto agli 8 transistor necessari per un latch statico.

Il **tempo di memorizzazione** di un latch dinamico è il tempo impiegato dal segnale per propagarsi attraverso la prima porta TG, in quanto non ci sono retroazioni da vincere e l'invertitore può agire anche dopo aver chiuso la porta TG.

$$t_{\text{mem}} = t_p^{TG}$$

14.3 Flip flop

Struttura generale

Un flip flop è un circuito sequenziale sincrono che memorizza un bit di informazione ed ha la stessa struttura di un latch, solo che il segnale di abilitazione EN viene sostituito da un segnale di clock CLK che determina quando l'informazione deve essere aggiornata.

Sincronizzazione dei flip flop

Il flip flop è uno strumento di memoria sensibile ai fronti del segnale di clock CLK . Un flip flop può essere di due tipi:

- **flip flop al fronte di salita** o Positive Edge Triggered (PET): l'uscita Q viene aggiornata al fronte di salita del segnale di clock CLK e solo in prossimità di tale fronte il FF è trasparente all'ingresso.
- **flip flop al fronte di discesa** o Negative Edge Triggered (NET): l'uscita Q viene aggiornata al fronte di discesa del segnale di clock CLK e solo in prossimità di tale fronte il FF è trasparente all'ingresso.

Per distinguere i due tipi di flip flop, si inserisce un pallino (di negazione) sul segnale di clock del flip flop al fronte di discesa.

Il flip flop necessita di un tempo di setup t_{setup} prima del fronte di clock e di un tempo di hold t_{hold} dopo il fronte di clock, durante i quali l'ingresso dati D deve rimanere stabile per garantire la corretta memorizzazione del dato.

Implementazione in configurazione master-slave

Un flip flop può essere implementato utilizzando due latch in configurazione master-slave, controllati dallo stesso segnale di clock CLK ma con segnali di abilitazione complementari:

- per avere un PET-FF, il latch master è attivo basso e il latch slave è attivo alto, quando il clock passa da 0 a 1, il latch master si disabilita rendendo il dato memorizzato disponibile all'ingresso del latch slave, che si abilita e diventa trasparente, aggiornando l'uscita Q con il dato memorizzato nel latch master, mentre quando il clock passa da 1 a 0, il latch slave si disabilita mantenendo l'uscita Q stabile e il latch master si abilita, diventando trasparente e permettendo la scrittura di un nuovo dato
- per avere un NET-FF, il latch master è attivo alto e il latch slave è attivo basso, quando il clock passa da 1 a 0, il latch master si disabilita rendendo il dato memorizzato disponibile all'ingresso del latch slave, che si abilita e diventa trasparente, aggiornando l'uscita Q con il dato memorizzato nel latch master, mentre quando il clock passa da 0 a 1, il latch slave si disabilita mantenendo l'uscita Q stabile e il latch master si abilita, diventando trasparente e permettendo la scrittura di un nuovo dato

Il **tempo di memorizzazione** di un flip flop implementato in configurazione master-slave è composto da due fasi:

- **tempo di setup**: necessario al latch master per propagare il dato in ingresso fino alla porta TG dell'anello prima del fronte di clock
- **tempo di hold**: siccome quando il clock cambia stato il latch master si disabilita rendendo stabile il dato memorizzato per lo slave, non è necessario alcun tempo di hold

$$t_{\text{setup}} = t_p^{TG1} + t_p^{INV1} + t_p^{INV2} \quad t_{\text{hold}} = 0$$

Il tempo di propagazione t_p^{FF} di un flip flop implementato in configurazione master-slave è il tempo impiegato dal FF a rendere disponibile il dato in uscita Q dopo il fronte di clock. Tale tempo è dato dal tempo di propagazione del dato attraverso il solo latch slave e dipende dal tipo di uscita (Q o \bar{Q}). Si nota che la porta TG2 forma una rete di Elmore con l'invertitore precedente INV1.

$$t_p^{CLK \rightarrow Q} = t_p^{TG3} + t_p^{INV3} \quad t_p^{CLK \rightarrow \bar{Q}} = t_p^{TG3} + t_p^{INV3} + t_p^{INV4}$$

Implementazione con clock a impulsi

Il flip flop controllato con clock a impulsi consiste in un latch (attivo alto) il cui segnale di abilitazione EN è pilotato da un circuito esterno che genera un impulso di breve durata in corrispondenza di uno dei fronti del segnale di clock CLK . In questo modo, il flip flop è trasparente solo per la durata dell'impulso generato, permettendo così la memorizzazione del dato in ingresso D in corrispondenza del fronte del clock.

Il circuito di generazione di impulsi consiste ad esempio di una porta logica a due ingressi: uno collegato al segnale di clock CLK e l'altro collegato ad un ritardo del segnale di clock CLK_{delay} (anche invertito) attraverso una serie di invertitori CMOS. Quando entrambi i segnali hanno lo stesso livello, la porta logica produce un'uscita bassa. Quando il segnale di clock CLK cambia stato, l'uscita della porta logica diventa alta per la durata del ritardo introdotto dal circuito di ritardo, generando così un impulso di breve durata. È possibile realizzare circuiti sensibili sia al fronte di salita che al fronte di discesa che ad entrambi i fronti del segnale di clock CLK .

In base a come viene realizzato il circuito di generazione di impulsi, cambiano i tempi di setup e di hold del flip flop. In generale l'impulso generato C (e il suo negato \bar{C}) vengono generati in ritardo rispetto al fronte del clock e devono essere stabili per un certo intervallo di tempo pari almeno al tempo di memorizzazione del latch t_{mem} . Il tempo di setup è il tempo in cui il segnale di dati deve rimanere stabile prima del fronte del clock e siccome l'impulso C viene generato dopo un certo ritardo di solito è nullo. Il tempo di hold è il tempo tra il fronte di clock e la fine dell'impulso C (e \bar{C}) ovvero quando si invertono le porte TG del latch, generalmente è maggiore del tempo di memorizzazione del latch in quanto include i ritardi introdotti nella generazione del segnale C . Il tempo di propagazione è pari al tempo tra il fronte di clock e il momento in cui il dato è reso disponibile all'uscita Q e generalmente è minore del tempo di hold in quanto il dato viene reso disponibile prima della chiusura delle porte TG del latch ed è maggiore del tempo di memorizzazione del latch in quanto vengono inclusi anche i ritardi introdotti nella generazione del segnale C .

Implementazione dinamica

Un flip flop può essere implementato utilizzando due latch dinamici complementari in configurazione master-slave con lo stesso principio descritto precedentemente. È inoltre possibile implementare un flip flop dinamico con clock a impulsi utilizzando un singolo latch dinamico come descritto sopra. Utilizzando latch dinamici, si riduce il numero di transistor necessari, ma si devono considerare i problemi legati alla perdita di carica del condensatore e alla vulnerabilità ai disturbi.

Il tempo di setup è il tempo impiegato dal dato in ingresso D per propagarsi attraverso la porta TG, non serve che si propaghi anche per l'invertitore, siccome (a differenza del flip flop statico) non c'è nessuna retroazione da vincere. Il tempo di hold è sempre zero, siccome quando il clock cambia stato il latch si disabilita rendendo stabile il dato memorizzato. Il tempo di propagazione del flip flop dinamico è analogo al tempo di propagazione del solo latch dinamico, ovvero è il tempo che il dato ci impiega a propagarsi dal primo invertitore all'uscita, si nota che la porta TG2 forma una rete di Elmore con l'invertitore precedente.

$$t_{\text{setup}} = t_p^{TG} \quad t_{\text{hold}} = 0 \quad t_p^{FF} = t_p^{INV1} + t_p^{TG2} + t_p^{INV2}$$

14.4 Registri di flip flop

I registri sono un esempio di circuito sequenziale sincrono costituito da più flip flop in parallelo dello stesso tipo, controllati dallo stesso segnale di clock, che permettono di memorizzare più bit di informazione contemporaneamente. Come i flip flop, i registri possono essere Positive Edge Triggered (PET) o Negative Edge Triggered (NET).

14.5 Disallineamento del clock

Siccome i latch e i flip flop contengono porte TG, per pilotarle è necessario avere sia il segnale di clock CLK che il suo complementare \bar{CLK} . Tuttavia, non è detto che i fronti (opposti) dei due segnali coincidano perfettamente a causa di ritardi di propagazione nei circuiti di generazione del segnale di clock complementare. Questo fenomeno è noto come **disallineamento del clock**.

Nei latch (e FF) con implementazione statica, il disallineamento può provocare brevi periodi in cui entrambe le porte TG sono attive, ma con il fenomeno della rigenerazione, il circuito tende a stabilizzarsi in autonomia senza creare grossi danni. Nei latch (e FF) con implementazione dinamica, l'apertura di entrambe le porte TG può causare un collegamento diretto tra ingresso e uscita del latch creando un fenomeno di race non voluto.

Per risolvere il problema si utilizzano segnali di clock non sovrapposti, ovvero sdoppia un segnale di clock CLK in due segnali $CLK1$ e $CLK2$ tali per cui quando uno è attivo l'altro è sempre inattivo. Questi segnali piloteranno ciascuno una delle due porte TG del latch (o FF) e garantiranno che non siano mai attive contemporaneamente.

Il circuito di generazione del segnale di clock non sovrapposti può essere implementato con due porte NOR o NAND con le uscite retroazionate agli ingressi reciproci e controllate dai segnali di clock CLK e \bar{CLK} . Nel caso della NOR l'uscita ad 1 è vincolata dal fatto che l'altra uscita deve rimanere a 0 e in questo modo i segnali non si sovrappongono mai a 1. Viceversa nel caso della NAND l'uscita a 0 è vincolata dal fatto che l'altra uscita deve rimanere a 1 e in questo modo i segnali non si sovrappongono mai a 0. Il tempo di non sovrapposizione è pari al tempo di propagazione delle porte NOR e NAND .

15 Circuiti sequenziali

15.1 Architettura pipeline

Idea implementativa

In un processo di elaborazione si potrebbero avere più stadi che lavorano in serie uno dopo l'altro. Sono preceduti da un registro di input e un registro di output. Il tempo di elaborazione totale è dato dalla somma dei tempi di ogni stadio per cui nel caso di reti molto lunghe il tempo di elaborazione risulta essere molto elevato.

$$t_{p,RC} = t_{p,RC1} + t_{p,RC2} + t_{p,RC3} \quad t_{tot} = t_{p,FF} + t_{p,RC} + t_{setup}$$

Per ovviare a questo problema si è pensato di dividere il processo in più stadi che lavorano in parallelo separati da registri di memoria intermedi. In un'architettura di questo tipo, chiamata **pipeline**, tutti e tre gli stadi lavorano in parallelo, permettendo di aumentare le informazioni elaborate per unità di tempo. Il tempo di propagazione è dato dalla rete più lenta ed è minore del caso precedente. Si nota però che la latenza, ovvero il tempo che intercorre tra l'input e l'output, aumenta in quanto ogni informazione deve attraversare più blocchi (ci sono anche i registri intermedi).

Gestione temporale nelle pipeline

Per descrivere temporalmente una rete pipeline si indicano:

- **tempo di contaminazione**: tempo impiegato dalla rete più veloce ad elaborare l'informazione e renderla disponibile al registro successivo, tale tempo deve essere maggiore del tempo di hold del registro, ovvero il dato deve rimanere stabile e non aggiornarsi prima che il registro lo acquisisca
- **tempo di propagazione**: tempo impiegato dalla rete più lenta ad elaborare l'informazione, tale tempo deve essere minore del tempo di clock meno il tempo di setup del registro successivo, ovvero il dato deve essere calcolato e stabile un t_{setup} prima del fronte di clock
- **tempo di clock**: deve essere maggiore del tempo di propagazione e del tempo di setup del registro successivo, ovvero deve essere garantito che il dato sia calcolato e stabile prima del fronte di clock (istante di campionamento del dato)
- **tempo di latenza**: numero di cicli di clock necessari affinché un'informazione attraversi tutta la pipeline, per una rete non-pipeline è 0, per una rete pipeline è pari al numero di stadi di registri della pipeline

$$\begin{aligned} t_{cont} &= t_{p,FF} + \min\{t_{p,RC1}, t_{p,RC2}, \dots\} & t_{cont} > t_{hold} \\ t_{prop} &= t_{p,FF} + \max\{t_{p,RC1}, t_{p,RC2}, \dots\} & t_{CLK} > t_{prop} + t_{setup} \\ t_{CLK} &= \min(t_{prop} + t_{setup}) & t_{latenza} = N_{\text{stadi di registri}} \end{aligned}$$

15.2 Macchine a stati

Struttura

Le macchine a stati sono circuiti sequenziali composti da una rete combinatoria e una memoria che memorizza lo stato attuale della macchina. La rete combinatoria elabora gli ingressi e lo stato attuale per produrre dati in uscita e il nuovo stato della macchina. (Il numero di stati 2^k è dato dal numero k di bit usati per rappresentare lo stato stesso.)

Temporizzazione delle macchine a stati

Come per le pipeline, si definiscono i 4 tempi principali osservando che il calcolo del tempo di propagazione massimo e minimo non è delle reti combinatorie, ma dei bit in uscita Y_i e dello stato successivo Q_i . Inoltre il tempo di latenza è sempre 2 in quanto non ci sono registri intermedi.

$$\begin{aligned} t_{cont} &= t_{p,FF} + \min\{t_{p,Y_0}, t_{p,Q_0}, \dots\} & t_{cont} > t_{hold} & t_{CLK} = \min(t_{prop} + t_{setup}) \\ t_{prop} &= t_{p,FF} + \max\{t_{p,Y_0}, t_{p,Q_0}, \dots\} & t_{CLK} > t_{prop} + t_{setup} & t_{latenza} = N_{\text{stadi di registri}} = 2 \end{aligned}$$

15.3 Esempi di macchine a stati - registri a scorrimento

Struttura

Il registro a scorrimento è una serie di flip-flop collegati in cascata e controllati dallo stesso segnale di clock. Ad ogni fronte di clock, il dato presente in ogni flip-flop viene trasferito al flip-flop successivo nella catena, mentre il primo flip-flop riceve un nuovo dato in ingresso. È possibile aggiungere altra logica di controllo per:

- resettare tutti i flip-flop a 0 attraverso un segnale di reset asincrono
- caricare (sincronizzati con il clk) i dati in parallelo in tutti i flip-flop attraverso un segnale di load che pilota un mux 2:1 che seleziona se il dato in ingresso al flip-flop è quello del flip-flop precedente o quello esterno
- uscite in parallelo dai flip-flop per leggere il contenuto di tutti i FF contemporaneamente

Temporizzazione

Il tempo di clock minimo è dato soltanto dai tempi dei flip-flop, non essendoci reti combinatorie tra un flip-flop e l'altro. Siccome il tempo di hold dei FF è nullo, non ci sono vincoli sul tempo di contaminazione.

$$t_{\text{CLK}} = t_{\text{p,FF}} + t_{\text{setup}} \quad \text{con MUX diventa } t_{\text{CLK}} = t_{\text{p,FF}} + t_{\text{mux}} + t_{\text{setup}}$$

15.4 Esempi di macchine a stati

Contatori

I contatori sono circuiti sequenziali il cui stato rappresenta un numero binario che viene incrementato di una unità ad ogni ciclo di clock. Un semplice contatore per potenze di 2 può essere realizzato con un sommatore binario che incrementa di uno il valore dei registri, tale sommatore può essere una cascata di HA o un sommatore ottimizzato CLA. È possibile costuire un contatore con modulo arbitrario aggiungendo un comparatore di uguaglianza che resetta il contatore quando viene raggiunto il valore desiderato salvato su un registro esterno.

Il tempo di ritardo della rete combinatoria è data dal tempo t_{sum} del sommatore e dipende dall'architettura usata. Per un RC con HA vale $t_{\text{sum}} = (N - 1)t_{\text{and}} + t_{\text{xor}}$.

Timer/counter/generatore di funzioni di microcontrollori

Il blocco in questione è composto da un comparatore di modulo arbitrario che pilota il segnale di reset di un latch ed è presente un altro comparatore che pilota il segnale di enable del latch con l'entrata sempre a 1. In questo modo il secondo comparatore fa salire l'uscita a 1 quando il contatore raggiunge un certo valore, mentre il primo comparatore fa resettare il latch a 0 quando il contatore raggiunge il valore massimo. L'uscita del latch è quindi un'onda quadra con fronte di salita e discesa programmabili.

Divisore di frequenza

Un divisore di frequenza è un circuito che prende in ingresso un'onda quadra con frequenza f_{in} e produce in uscita un'onda quadra con frequenza $f_{\text{out}} = \frac{f_{\text{in}}}{N}$. In base al valore di N si possono usare diverse implementazioni:

- per $N = 2^k$ potenza di 2 si può usare un contatore binario a k bit e prendere in uscita il bit k -esimo bit, se lo si vuole rendere programmabile, basta aggiungere un MUX che seleziona quale bit di uscita prendere
- per N arbitrario (multiplo di 2) si può usare un contatore con reset programmabile collegato al clock di un latch con l'uscita negata retroazionata all'ingresso, in modo che ogni impulso di reset del contatore faccia cambiare stato al latch

16 Clock e sincronizzazione

16.1 Distribuzione del clock

Struttura della rete di distribuzione

Il circuito di distribuzione del clock è costituito da un circuito di generazione del segnale di clock, una rete di distribuzione globale ai vari moduli del circuito e una rete di distribuzione locale all'interno dei singoli moduli.

Durante la distribuzione possono provocarsi ritardi e sfasamenti, l'importante è che la fase relativa del segnale di clock che arriva a tutti i moduli sia perfettamente sincronizzata, non importa il ritardo assoluto dall'origine.

Ritardo delle piste in interconnessione

Per calcolare il ritardo delle piste di interconnessione si modella la pista come una serie di celle con una certa resistenza (data dalla legge di Ohm) e una certa capacità parassita (data dalle capacità di area e dalle capacità di bordo). Calcolando il tempo di ritardo della rete di Elmore così costruita si osserva che il modello RC equivale ad una lunga pista con tutta la capacità condensata al centro della pista stessa.

$$R_{W,\text{pista}} = \rho \frac{L_{\text{lunghezza}}}{h_{\text{altezza}} \cdot W_{\text{spessore}}} = R_q \frac{L}{W} \quad \text{con } R_q = \text{resistenza per quadro} [\Omega/q]$$

$$C_{W,\text{pista}} = C_{WA} \cdot W \cdot L + C_{WB} \cdot 2L \quad \begin{aligned} &\text{con } C_{WA} = \text{capacità di area} = [fF/\mu m^2] \\ &C_{WB} = \text{capacità di bordo} = [fF/\mu m] \end{aligned}$$

$$\begin{aligned} t_{\text{pista}} &= 0.69 \cdot R_{out} \cdot C_{out} && \text{circuito che pilotala pista} \\ &+ 0.69 \cdot (R_{out} + R_w/2) \cdot C_{W,\text{pista}} && \text{prima metà della pista con capacità al centro} \\ &+ 0.69 \cdot (R_{out} + R_w) \cdot C_L && \text{seconda metà pista e carico pilotato} \end{aligned}$$

Errori nella propagazione del clock

Le sorgenti di errori del clock possono essere:

- **errori sistematici:** errori prevedibili e corretti in fase di progettazione, come carichi capacitivi diversi per ogni percorso di clock o lunghezze diverse delle piste di clock
- **errori casuali:** randomici e non prevedibili come sbavature dei processi di produzione

Gli errori si manifestano come:

- **skew:** sfasamento costante nel tempo (e talvolta calcolabili e compensabili) tra due segnali di clock generati dalla stessa sorgente, dovuto a ritardi diversi nei percorsi di distribuzione
- **jitter:** variazione completamente casuale e che variano da ciclo a ciclo di clock, con media nulla e non prevedibili, per questo si definisce il valore di jitter massimo (in modulo)

Effetti del ritardo di propagazione del Clock

Analizzando due segnali sfasati di uno skew t_{skew} , lo sfasamento reale può variare nel range di $t_{\text{skew}} \pm 2t_{\text{jitter}}$ ad esempio se il segnale in anticipo viene anticipato di un jitter e il segnale in ritardo viene ritardato di un jitter o viceversa. Il massimo e il minimo sfasamento tra due segnali si indicano con:

$$\delta_{min} = t_{\text{skew}} - 2t_{\text{jitter}} \quad \delta_{max} = t_{\text{skew}} + 2t_{\text{jitter}}$$

Tale sfasamento può violare i vincoli di temporizzazione dei circuiti sequenziali e la sincronizzazione tra moduli diversi: in particolare uno sfasamento positivo nel clock potrebbe violare le condizioni su t_{hold} , mentre uno sfasamento negativo potrebbe non rispettare il t_{setup} .

$$\begin{aligned} t_{\text{prop,reale}} &= t_{\text{prop,ideale}} - t_{\text{skew}} + 2t_{\text{jitter}} & t_{\text{skew}} > 0 & \text{se ritarda progressivamente} \\ t_{\text{cont,reale}} &= t_{\text{cont,ideale}} - t_{\text{skew}} - 2t_{\text{jitter}} & t_{\text{skew}} < 0 & \text{se anticipa progressivamente} \end{aligned}$$

16.2 Generazione del clock e Phase Locked Loop - PLL

Struttura del PLL

Un PLL (o Anello ad Aggancio di Fase) è un circuito con retroazione che genera un segnale di clock ad una data frequenza con elevata precisione. È composto da PFD, CP, VCO e due divisori di frequenza.

Voltage Controlled Oscillator (VCO)

Il VCO è un oscillatore che genera un'onda quadra la cui frequenza è regolata da una tensione analogica in ingresso V_C . È costituito da un anello di N “invertitori” (raddrizzati) controllati in corrente in modo da variare il tempo t_p di propagazione e quindi la frequenza di oscillazione complessiva: $f_{\text{out}} = 1 / 2Nt_p$.

Un invertitore controllato in corrente è costituito da un invertitore CMOS collegato a V_{DD} e GND attraverso un PMOS e un NMOS rispettivamente che funzionano da generatori ideali di corrente. I due MOS extra sono collegati gate-gate ad altri due mos identici che formano un circuito di corrente speculare in cui il gate degli nmos è pilotato dalla tensione di controllo V_C e il pmos ha drain e gate ponticellati per lavorare in saturazione a diodo. Si hanno le seguenti relazioni:

$$I = k_n' Z_n \frac{(V_C - V_{TN})^2}{2} \quad \text{oppure} \quad I = k_n' Z_n V_{DSATN} \left(V_C - V_{TN} - \frac{V_{DSATN}}{2} \right)$$

$$\Delta T_{\text{ritardo invertitore}} = \frac{C \cdot V_{DD}/2}{I} \quad T_{\text{periodo anello}} = 2N \Delta T_{\text{ritardo invertitore}}$$

Si osserva quindi che il tempo di ritardo t_p del singolo invertitore dipende dalla corrente I e di conseguenza da V_C : aumentando V_C aumenta I e diminuisce t_p , aumentando la frequenza di oscillazione.

Siccome gli invertitori controllati in corrente non godono di proprietà rigenerative, inoltre per ridurre il consumo statico di correnti “analogiche” che pilotano gate, si inserisce un invertitore CMOS classico dopo ogni invertitore controllato in corrente. Si aggiunge, infine, un altro invertitore CMOS classico da solo per rendere l’anello invertente.

Phase Frequency Detector (PFD)

Il PFD è un circuito sequenziale asincrono che riceve in input due segnali di clock REF e IN e produce in uscita due segnali di controllo UP e $DOWN$ che indicano se il segnale IN è in anticipo o in ritardo rispetto a REF .

- se IN è in anticipo rispetto a REF in uscita si ha $UP = 0$ e $DOWN = 1$
- se IN è in ritardo rispetto a REF in uscita si ha $UP = 1$ e $DOWN = 0$
- se i due segnali sono sincronizzati in uscita si ha $UP = 0$ e $DOWN = 0$

Il circuito è implementato da due flip flop con ingressi fissi a 1, controllati dai due segnali di clock in ingresso, inoltre possiedono un segnale di reset asincrono che azzera entrambi i flip flop quando entrambi sono settati a 1. In questo modo quando arriva il fronte di uno dei due segnali di clock, il flip flop corrispondente si setta a 1 e l’altro rimane a 0. Quando arriva anche l’altro fronte di clock, il secondo flip flop si setta a 1 e il segnale di reset azzera entrambi i flip flop.

Charge Pump (CP)

La Charge Pump è un circuito che riceve in ingresso i segnali UP e $DOWN$ dal PFD e produce in uscita una tensione di controllo V_C per il VCO. Il circuito è costituito da un invertitore CMOS che pilota una capacità C . Quando arriva un impulso sul segnale UP , l’invertitore porta il nodo di uscita a V_{DD} e la capacità si carica, mentre quando arriva un impulso sul segnale $DOWN$, l’invertitore porta il nodo di uscita a GND e la capacità si scarica.

Per rendere il sistema stabile e limitare la velocità di caricamento/scaricamento della capacità, si utilizzano due mos in regime di generatori ideali di corrente (come nel VCO) per limitare la corrente di carica e scarica della capacità. Le tensioni che pilotano tali mos sono fissate in fase di progettazione.

Divisori di frequenza

Nel PLL sono presenti anche due divisori di frequenza, uno in ingresso al PFD che divide la frequenza del segnale f_{REF} di un fattore N e uno che collega il VCO al PFD (retroazione) che divide la frequenza del segnale f_{IN} di un fattore M . La PLL quindi lavora per mantenere la condizione:

$$\frac{f_{IN}}{M} = \frac{f_{REF}}{N} \quad \Rightarrow \quad f_{IN} = \frac{M}{N} f_{REF}$$

Si osserva che M e N possono essere solo multipli di 2, ma questo non ci interessa perché alla fine il rapporto semplifica tale fattore 2 e si può ottenere qualsiasi rapporto di frequenze razionale.

16.3 Struttura della rete di distribuzione del clock

Struttura globale ad albero a H

Per minimizzare lo skew tra i vari moduli dei circuiti si utilizza la distribuzione del clock ad albero ad H, ovvero una struttura ad albero in cui i rami originati da ogni biforcazione hanno la stessa lunghezza e di conseguenza lo stesso tempo di ritardo RC. In prossimità dei moduli si aggiungono dei buffer (“dinamici” in funzione del carico) per ripristinare il segnale di clock e adattarlo al carico.

Struttura locale a griglia

Per distribuire il clock all'interno dei moduli si utilizza una rete a griglia che forma una scacchiera di piste orizzontali e verticali polarizzate contemporaneamente al segnale di clock, a cui sono collegati i vari elementi sincroni del modulo. In questo modo si cerca di contenere lo skew locale evitando di avere numerose piste di clock con lunghezze diverse.

Origine di Skew e Jitter

Le principali origini di skew e jitter nella distribuzione del clock sono:

- variazioni del processo di fabbricazione:
ad esempio imprecisioni sulle dimensioni W ed L , sugli spessori dell'ossido e sui livelli di droggaggio che influenzano k_n e V_{TN} nei circuiti PLL e nei buffer di distribuzione del clock; tali errori casuali sono costanti nel tempo (localmente ad un singolo chip) e di conseguenza si definiscono skew
- variazioni delle interconnessioni:
ad esempio imprecisioni nello spessore del dielettrico, nella geometria delle piste e planarizzazione del chip che generano skew in quanto costanti nel tempo (localmente ad un singolo chip)
- variazioni della temperatura:
un uso non omogeneo del chip può generare gradienti di temperatura che provocano variazioni nelle tensioni di soglia e nella mobilità dei portatori, tali variazioni cambiano nel tempo, ma per tempi infinitesimi sono costanti e generano skew
- variazioni della tensione di alimentazione:
le cadute di tensione sulle linee di alimentazione possono variare nel tempo in funzione del carico dei moduli vicini generando skew (variazioni lente), oppure possono essere causate da rumore di commutazioni logiche che generano jitter, per ridurre tali jitter si aggiungono capacità di decoupling (disaccoppiamento) tra le linee di alimentazione in prossimità dei buffer del clock

Riduzione dello sfasamento nella distribuzione locale e DLL

Per bilanciare il carico locale dei singoli moduli e ridurre lo skew dovuto ai carichi diversi sui buffer, si utilizzano circuiti chiamati Delay Locked Loop (DLL). Sono simili ai PLL, ma non possiedono divisori di frequenza in quanto si vuole avere in uscita lo stesso segnale di clock in ingresso. Inoltre il VCO è sostituito da una catena di ritardo VCDL (Voltage Controlled Delay Line) che riceve in ingresso (oltre alla tensione V_C) la frequenza originale e produce in uscita lo stesso segnale ritardato di un tempo regolabile in base al carico del modulo. In questo modo si è in grado di compensare le variazioni di carico locale e ridurre lo skew.

17 Memorie a semiconduttore

17.1 Introduzione alle memorie a semiconduttore

Clasificazione delle memorie

Le memorie a semiconduttore si classificano in:

- memorie volatili perdono il contenuto memorizzato in assenza di alimentazione
 - SRAM (Static RAM) e DRAM (Dynamic RAM)
- memorie non volatili mantengono il contenuto memorizzato anche in assenza di alimentazione
 - memorie di sola lettura (ROM) programmabili una sola volta in fase di produzione
 - memorie riscrivibili EEPROM, Flash-EEPROM

Registro come elemento fondamentale delle memorie

L'elemento fondamentale di una memoria è il registro, che memorizza un bit di informazione. Un registro può essere realizzato con un flip-flop D, soltanto che si avrebbe un enorme spreco di spazio. Per questo motivo si usano delle celle di memoria compatte con una stessa linea per lettura e scrittura e un segnale di abilitazione. La gestione di lettura e scrittura avviene a livello di matrice di celle di memoria.

Architettura tipica delle memorie

Le singole celle di memoria sono organizzate in matrici di righe e colonne. Attraverso un decoder di riga si attiva una word line (orizzontale) che abilita la lettura/scrittura di tutte le celle di una riga. Le celle di ogni colonna sono collegate assieme tramite una bit line (verticale) che termina in un MUX (decoder di colonna) che seleziona su quale colonna agire. Ogni colonna possiede un sense amplifier per ottimizzare la lettura del dato.

Ogni matrice è detta banco di memoria, in un chip possono esserci più banchi di memoria indipendenti selezionabili tramite un selettore di blocco.

Ogni cella possiede un indirizzo univoco e, se si utilizzano potenze di 2 per il numero di banchi, righe e colonne, l'indirizzo può essere spezzato in tre parti: indirizzo di banco, indirizzo di riga, indirizzo di colonna.

17.2 Matrice di celle per memorie ROM

La memoria ROM (Read Only Memory) è una memoria non volatile programmabile una sola volta in fase di produzione che permette l'immagazzinamento di informazioni binarie fisse e non modificabili. L'informazione viene memorizzata a livello hardware per cui non è possibile modificarla in seguito.

ROM NOR

La ROM NOR è costituita da una matrice di celle in cui ogni cella è formata da un transistor collegato tra la bit line e il ground. La word line attiva il gate del transistor portando la bit line a massa che altrimenti rimane a V_{DD} tramite un pmos di pull-up. La cella memorizza uno 0 se il transistor è collegato alla bit line, altrimenti memorizza un 1 se non c'è nessun collegamento.

ROM NAND

La ROM NAND è costituita da una matrice di celle in cui una colonna è formata da una serie di transistor (uno per cella) collegati dall'estremità inferiore a massa e dall'estremità superiore alla bit line e a un pmos di pull-up. Ci sono inoltre le word line che attivano i gate dei transistor di una stessa riga. Per la lettura, si portano tutte le word line a 1 tranne quella della riga da leggere che viene portata a 0. Se per la riga selezionata il transistor non è bypassato, agisce come interruttore aperto e la bit line rimane a V_{DD} , leggendo un 1. Altrimenti se c'è un bypass source-drain, la bit line rimane a massa, (tutti gli altri transistor sono attivi) e si legge uno 0.

In alternativa ai collegamenti di bypass per i transistor, si droga il substrato del nmos in modo da avere una V_{TN} minore di 0. I transistor così ottenuti sono detti a svuotamento e sono sempre attivi. A livello di fabbricazione, si aggiunge una maschera ACTIVE per i drogaggi del substrato e si risparmia molto spazio evitando di fare i contatti metal per i collegamenti di bypass.

Pro e contro

Le ROM NOR hanno tempi di accesso più veloci rispetto alle ROM NAND in quanto queste ultime devono scaricare a massa la bit line attraverso una serie di nmos in cascata. Si ha infatti una lunga rete di Elmore con elevate resistenze.

Le ROM NAND, invece, sono più compatte in quanto non hanno bisogno dei contatti metallici per collegare i vari nmos della ROM NOR a massa e, avendo i mos in cascata, è possibile ridurre lo spazio tra le word line

17.3 Matrice di celle per memorie non volatili e scrivibili

Mosfet a gate flottante

I mosfet a gate flottante sono dei mosfet particolari in cui è possibile programmare la tensione di soglia in modo da avere transistor “normali”, sempre attivi (a svuotamento) o sempre spenti (a arricchimento). Per fare ciò si aggiunge un ulteriore strato di ossido tra il canale e l'elettrodo di gate, detto floating gate, senza collegamenti elettrici. Si forma in questo modo un doppio condensatore cmos tra canale e floating gate e tra floating gate e gate di controllo.

In base a come è caricato il floating gate, si modifica la tensione di soglia del mosfet:

- se il floating gate è neutro, si ha la V_{TN} standard del mosfet
- se il floating gate è carico positivamente il gate di controllo deve applicare una tensione minore per creare il canale in quanto la carica positiva nel FG attrae gli elettroni nel canale, si ha quindi una $V_{TN} < 0$ che fa funzionare il mosfet come un transistor a svuotamento sempre attivo
- se il floating gate è carico negativamente, il gate di controllo deve applicare una tensione maggiore per vincere le cariche negative nel FG che respingono gli elettroni nel canale, si ha quindi una $V_{TN} > V_{DD}$ che fa funzionare il mosfet come un transistor a arricchimento sempre spento

Per caricare/scaricare il floating gate ci si basa sul fenomeno quantistico dell'effetto tunnel, ovvero se si applica una tensione molto elevata tra canale e gate di controllo, gli elettroni possono attraversare lo strato di ossido e andare a caricare/scaricare il floating gate. Lo strato di ossido tra canale e FG è contenuto per permettere tale effetto di tunneling mentre quello tra FG e gate di controllo è più spesso per evitare perdite di carica. Si hanno quindi due operazioni fondamentali:

- **programmazione:** iniezione di elettroni nel FG rendendo V_T alta e il mosfet sempre spento, avviene tramite l'applicazione di un potenziale positivo molto alto al gate di controllo mentre il canale è a massa (il gate attira elettroni dal canale verso il FG)
- **cancellazione:** rimozione di elettroni dal FG rendendo V_T bassa e il mosfet sempre attivo, avviene tramite l'applicazione di un potenziale positivo molto alto al canale mentre il gate di controllo è a massa (il canale attira elettroni dal FG verso il canale)

Le operazioni di programmazione e cancellazione possono avvenire in maniera uniforme se source e drain hanno lo stesso potenziale oppure in maniera laterale se il potenziale si applica solo a uno dei due, lasciando l'altro libero.

Si osservano le seguenti considerazioni:

- per ritornare allo stato neutro del FG è necessario dosare opportunamente le tensioni di programmazione e cancellazione per non sovraccaricare o scaricare eccessivamente il FG ottenendo l'effetto contrario
- lo strato di ossido tra canale e FG va fatto sottile per permettere l'effetto tunnel, ma non troppo sottile per evitare perdite di carica nel tempo
- le tensioni di programmazione e cancellazione sono molto elevate rispetto alle normali tensioni di funzionamento in modo che nel normale funzionamento non avvenga alcun effetto tunnel

Memorie EEPROM (NOR)

Le memorie EEPROM (Electrically Erasable and Programmable Read Only Memory) sono memorie non volatili riscrivibili che utilizzano mosfet a gate flottante per memorizzare i dati. Le EEPROM hanno struttura simile alle ROM NOR in cui ogni cella è formata da nmos a gate flottante che collega la massa alla bit line attraverso un nmos di selezione. Gli nmos di selezione sono controllati da una word line e il fg-nmos funge da interruttore programmabile per collegare o meno l'nmos della cella a massa. È presente

inoltre una control line che collega tutti i gate dei fg-nmos di una stessa riga. Le operazioni di lettura, programmazione e cancellazione possono essere fatte a livello di singola cella come segue:

- l'operazione di **lettura** avviene attivando la word line della cella da leggere e verificando lo stato della bit line come in una ROM NOR, la control line non viene usata
- l'operazione di **programmazione** (nmos sempre spento, bit 1) avviene attivando la word line della cella da programmare e applicando una tensione elevata tra la control line (potenziale positivo) e la bit line (potenziale negativo) in modo da iniettare elettroni nel floating gate tramite effetto tunnel
- l'operazione di **cancellazione** (nmos sempre attivo, bit 0) avviene attivando la word line della cella da cancellare e applicando una tensione elevata tra la bit line (potenziale positivo) e la control line (potenziale negativo) in modo da rimuovere elettroni dal floating gate tramite effetto tunnel

Memorie Flash-EEPROM (NOR e NAND)

A differenza delle EEPROM, le memorie Flash-EEPROM non permettono operazioni di scrittura a livello di singola cella, ma solo a livello di intere pagine (gruppi di righe). Tutti i fg-nmos che condividono la stessa word line sono programmati assieme. Manca inoltre il transistor di selezione tra bit line e fg-nmos e di conseguenza le memorie Flash-EEPROM sono più compatte delle EEPROM classiche. È possibile implementarle sia con struttura NOR che NAND. L'operazione di programmazione deve sempre essere preceduta da una fase di cancellazione della pagina da sovrascrivere.

- Flash-EEPROM NOR:

i fg-nmos di coppie di righe non sono collegati direttamente a massa, ma ad una source line, le operazioni di lettura, programmazione e cancellazione avvengono come segue:

- **lettura:** avviene come in una normale ROM NOR, le source line sono portate a massa
- **cancellazione:** le source line vengono portate ad un potenziale positivo elevato, mentre le word line della pagina da cancellare a massa, in modo da rimuovere elettroni e avere fg-nmos sempre attivi (bit 0)
- **programmazione:** tutte le source line e le word line sono portate a massa, tranne la word line della pagina da programmare che viene portata ad un potenziale positivo elevato. Le bit line vengono portate ad un potenziale basso per le celle da programmare (fg-nmos sempre spenti, bit 1) e vengono lasciate aperte per le altre celle (fg-nmos sempre attivi, bit 0)

- Flash-EEPROM NAND:

la struttura è simile a quella delle ROM NAND, si aggiungono due nmos in testa e in coda alla serie di fg-nmos che vengono controllati da due linee di selezione. Il collegamento a massa avviene tramite la source line usata anche nelle operazioni di scrittura. Le operazioni di lettura, programmazione e cancellazione avvengono come segue:

- **lettura:** avviene come in una normale ROM NAND, la source line è portata a massa e i due nmos di selezione vengono abilitati
- **cancellazione:** la source line e le bit line sono portate ad un potenziale positivo elevato, mentre le word line della pagina da cancellare sono portate a massa, in modo da rimuovere elettroni e avere fg-nmos sempre attivi (bit 0), i due nmos di selezione sono abilitati
- **programmazione:** vengono abilitati solo i mos di selezione che collegano la serie alla bit line, lasciando isolata la source line. Le word line della pagina da programmare sono portate ad un potenziale positivo elevato, mentre le bit line sono portate ad un potenziale basso per le celle da programmare (fg-nmos sempre spenti, bit 1) o tenute ad un potenziale alto per le altre celle (fg-nmos sempre attivi, bit 0)

Il vantaggi NAND-NOR sono analoghi a quelli delle ROM NAND-NOR: le Flash-EEPROM NAND sono più compatte ma più lente viceversa le Flash-EEPROM NOR sono più veloci ma più ingombranti.

Memorie Flash multilivello (MLC)

Per aumentare la densità di memorizzazione delle memorie Flash-EEPROM, è possibile utilizzare più livelli di carica nel floating gate per memorizzare più di un bit per cella. Usando $N = 2^k$ livelli di carica, è possibile memorizzare k bit per cella. Si richiede però una circuiteria di lettura e scrittura più complessa e precisa per gestire i vari livelli di carica.

17.4 Matrice di celle per memorie RAM

Memoria RAM statica

La memoria RAM statica (SRAM) è costituita da celle di memoria formate da due invertitori cross-coupled in cui l'uscita di un invertitore controlla l'ingresso dell'altro e viceversa (come in un flip-flop senza TG). Alle uscite dei due invertitori sono collegati due nmos di selezione che collegano le uscite (una dritta, l'altra negata) a due bit line complementari. Il costo in termini di area è di 6 mos per cella.

Alle bit line è collegato anche un modulo detto sense amplifier con circuiteria analoga alla cella, soltanto che le uscite degli invertitori sono collegate direttamente alle bit line e ci sono due mos che controllano l'alimentazione del sense amplifier tramite un segnale di abilitazione (attivo alto). È presente anche un pmos controllato da un segnale *EQ* che collega le due bit line per bilanciarle prima della lettura (attivo basso).

La SRAM vista da fuori ha i seguenti ingressi/uscite:

- n bit dei segnali di indirizzo in ingresso
- m bit di dati in ingresso/uscita (segnali bidirezionale)
- CS segnale di chip select in ingresso attivo basso
- WE segnale di write enable in ingresso attivo basso
- OE segnale di output enable in ingresso attivo basso

La lettura e scrittura avvengono come segue:

- **lettura:** dall'esterno si rendono disponibili i bit di indirizzo, si porta CS a basso per abilitare il chip, si porta WE ad alto per abilitare la lettura, si porta OE a basso per abilitare l'uscita dei dati. Dall'interno si attiva il segnale *EQ* per bilanciare le bit line, si disattiva il segnale *EQ*, si attiva la word line della riga da leggere e la cella inizia a polarizzare le bit line in base al dato memorizzato, si attiva il sense amplifier per accelerare la polarizzazione delle bit line e di conseguenza del dato nella cella letta e si leggono i dati dalle bit line in uscita
- **scrittura:** dall'esterno si rendono disponibili i bit di indirizzo e i dati da scrivere, si porta CS a basso per abilitare il chip, si porta WE a basso per abilitare la scrittura, si porta OE ad alto per disabilitare l'uscita dei dati. Dall'interno si polarizzano le bit line con i dati da scrivere, si attiva la word line della riga da scrivere e la cella viene forzata a memorizzare il dato presente sulle bit line

Si nota che la lettura non è distruttiva perché la cella in autonomia è in grado di mantenere il dato memorizzato al valore logico corretto e agire per limitare i disturbi per la proprietà di bistabilità degli invertitori cross-coupled.

Si nota che gli invertitori della cella non sono in grado di polarizzare velocemente le bit line a causa dell'elevata capacità parassita delle bit line e della resistenza dei mos di selezione. Per questo entra in azione il sense amplifier che accelera la polarizzazione delle bit line e di conseguenza del dato nella cella letta.

Memoria RAM dinamica

Nelle memorie RAM dinamiche (DRAM) ogni cella di memoria è costituita da un mos di selezione che collega una bit line ad una capacità di storage che immagazzina la carica elettrica rappresentante il dato logico (1 o 0). La capacità di storage è soggetta a correnti di perdita che scaricano la carica nel tempo, per questo motivo è necessario un circuito di refresh che periodicamente legge e riscrive il dato in ogni cella per mantenere il contenuto della memoria.

La SRAM vista da fuori ha i seguenti ingressi/uscite:

- n bit dei segnali di indirizzo (ed eventualmente di selezione del banco) in ingresso
- m bit di dati in ingresso/uscita (segnali bidirezionale)
- CS segnale di chip select in ingresso attivo basso
- WE segnale di write enable in ingresso attivo basso
- RAS segnale di row address strobe di sincronismo per l'indirizzo di riga
- CAS segnale di column address strobe di sincronismo per l'indirizzo di colonna
- DQM segnale di data mask analogo allo OE delle SRAM, attivo basso

La lettura e scrittura avvengono come segue:

- **lettura:** dall'esterno si rendono disponibili i bit di indirizzo di riga e banco, si porta CS a basso per abilitare il chip, si porta WE ad alto per abilitare la lettura, si porta DQM a basso per abilitare l'uscita dei dati. Si porta RAS a basso per attivare il decoder di riga e selezionare la word line della riga da leggere, infine si rendono disponibili i bit di indirizzo di colonna e si porta CAS a basso per attivare il decoder di colonna e selezionare la bit line della colonna da leggere. Dall'interno si attiva il sense amplifier per precaricare la bit line ad un valore intermedio, si attiva la cella selezionata che polarizza la bit line in base al dato memorizzato (se 1 la bit line aumenta lievemente di potenziale, se 0 diminuisce lievemente di potenziale), si attiva il sense amplifier che ripristina la bit line al valore logico corretto (riscrivendo il dato nella cella), infine si leggono i dati dalle bit line.
- **scrittura:** dall'esterno si rendono disponibili i bit di indirizzo di riga e banco, si porta CS a basso per abilitare il chip, si porta DQM ad alto per disabilitare l'uscita dei dati. Si porta RAS a basso per attivare il decoder di riga e selezionare la word line della riga da scrivere, infine si rendono disponibili i bit di indirizzo di colonna e si porta CAS a basso per attivare il decoder di colonna e selezionare la bit line della colonna da scrivere. Infine di abilita il segnale WE a basso per attivare la scrittura. Dall'interno si polarizzano le bit line con i dati da scrivere, si attiva la word line della riga da scrivere e si carica la capacità di storage con il dato presente sulla bit line.

Si nota che la lettura è distruttiva perché la cella non è in grado di mantenere il dato memorizzato autonomamente a causa della perdita di carica nella capacità di storage. Durante la lettura, il dato letto viene ripristinato nella cella tramite il sense amplifier mentre polarizza la bit line al valore logico corretto.

Il ciclo di refresh per manterenere valido il contenuto delle singole celle consiste in una lettura integrale della memoria eseguita in parallelo su tutte le bitline e su tutti i banchi di memoria. Durante tale operazione il decoder di colonna è disabilitato e il segnale delle bit line non viene portato all'uscita.

Le capacità di storage possono essere realizzate verticalmente per risparmiare spazio in diversi modi:

- trench: si scava un buco profondo nel substrato e lo si riempie di materiale metallico per formare un condensatore a gusci cilindrici, il problema è che l'elevata profondità crea una rete di Elmore all'interno del metallo ritardando le operazioni di carica e scarica
- stack: si costruisce un condensatore verso l'alto sopra il substrato, il problema è che tutte le piste di interconnessione si trovano sopra il condensatore, distanti dal substrato e i contatti metallici avranno una rete di Elmore che rallenterà la propagazione dei segnali

17.5 Tempi di accesso

Circuiti equivalenti di WL e BL

Per analizzare i tempi di accesso delle memorie, si modellano le word line e le bit line con circuiti equivalenti RC risolvibili con le reti di Elmore.

Per le word line, rispetto al modello per le interconnessioni, bisogna aggiungere le capacità parassite dei gate dei mos delle celle collegate alla word line (per le SRAM va raddoppiata perché ci sono due mos per cella collegati alla WL).

$$R_{WL} = R_q \frac{D_{\text{lunghezza orizzontale celle}}}{L_{WL} \text{ spessore word line}} \quad C_{WL} = C_A^{WL} \cdot D \cdot L_{WL} + C_B^{WL} \cdot 2D + C_{g_{\text{gate dei mos}}} \\ t_{p,WL} = 0.69 \cdot R_0 \cdot C_{WL} + 0.69 \cdot R_{WL} \cdot C_{WL}/2 \quad \text{con } R_0 = \text{dec. di riga}$$

Per le bit line, rispetto al modello per le interconnessioni, bisogna aggiungere le capacità parassite dei drain/source dei mos delle celle collegate alla bit line.

$$R_{BL} = R_q \frac{H_{\text{altezza verticale celle}}}{L_{BL} \text{ spessore bit line}} \quad C_{BL} = C_A^{BL} \cdot H \cdot L_{BL} + C_B^{BL} \cdot 2H + C_{d_{\text{drain dei mos}}} \\ t_{p,BL} = 0.69 \cdot R_0 \cdot C_{BL} + 0.69 \cdot R_{BL} \cdot C_{BL}/2 \quad \text{con } R_0 = \text{dec. di colonna}$$

Per le bit line delle NAND, bisogna considerare la resistenza equivalente della serie di mos collegati alla bit line:

$$t_{p,BL} = 0.69 \cdot R_n \cdot 2C_{\text{drain}} \cdot \frac{M(M+1)}{2} + 0.69 \cdot M \cdot R_n \cdot C_{BL} \quad \text{con } N_{\text{mos}} = \text{numero di mos in serie}$$

17.6 Circuiti periferici

Decoder di riga

Il decoder di riga prende in ingresso un indirizzo binario di n bit e attiva una sola delle 2^n uscite, ovvero la word line corrispondente all'indirizzo in ingresso. Può essere realizzato anche in logica negata in cui l'uscita attivata è portata a 0 mentre tutte le altre sono portate a V_{DD} .

Una prima implementazione potrebbe essere quella di dedicare una porta AND (o NOR) per ogni uscita del decoder che riceve in ingresso gli n bit dell'indirizzo (diretti o negati a seconda del valore logico necessario per attivare l'uscita). Il tempo di ritardo cresce esponenzialmente con n a causa del carico degli invertitori sugli ingressi, per cui si usa una struttura a stadi. In logica negata si usano NAND al posto delle NOR.

Si suddividono gli n bit di ingresso in k gruppi di m bit ciascuno, con $n = k \cdot m$. Si costruisce un predecoder con k gruppi di 2^m porte NAND che prende in ingresso gli m bit di ogni gruppo e produce 2^m uscite parziali. Tali uscite parziali vengono poi combinate in un secondo stadio con porte NOR per ottenere le 2^n uscite finali del decoder. In questo modo si rallenta la crescita del tempo di ritardo complessivo. In logica negata si usano NOR nel predecoder e NAND nel secondo stadio.

Decoder di colonna

Il decoder di colonna svolge il ruolo di un MUX, selezionando una sola delle 2^n bit line in base all'indirizzo di colonna in ingresso, ma viene realizzato con tante porte TG per ogni bit line che vengono attivate una alla volta, pilotate dalle uscite di un decoder di colonna vero e proprio. Le porte TG permettono un collegamento bidirezionale tra la bit line selezionata e il bus di I/O della memoria.

Sense amplifier

Il sense amplifier viene posizionato uno per colonna, generalmente indicato tra la matrice di celle e il decoder di colonna. Un caso particolare di utilizzo del sense amplifier è nelle memorie DRAM dove è presente solo una bit line per colonna: si divide la matrice di celle in due metà speculari e si posizionano i sense amplifier al centro. Si aggiungono inoltre due celle fittizie (reference cell) per ogni colonna per equilibrare il carico capacitivo dei due rami delle bit line. Durante la lettura, prima si ha una precarica delle bit line ad un valore intermedio, poi si attiva la word line della riga da leggere e la cella di riferimento nella metà opposta. La cella da leggere altera la bit line in base al dato memorizzato e viene attivato il sense amplifier rileva la differenza di potenziale tra le due bit line e ripristina il valore logico corretto.

Buffer bidirezionale di I/O - invertitore tri-state

Per alcune applicazioni è utile avere un invertitore che, oltre allo stato logico alte e basso, possa assumere uno stato di alta impedenza (tri-state), ovvero con resistenza di uscita molto elevata che isola il circuito a valle. Per fare ciò si utilizza un invertitore cmos a cui vengono aggiunti un nmos e un pmos tra le reti di pull-up e pull-down e l'uscita dell'invertitore. I due mos aggiuntivi sono controllati da un segnale di abilitazione: quando è attivo, i mos sono spenti e l'uscita è in alta impedenza, quando è inattivo, i mos sono accesi e l'invertitore funziona normalmente. Il segnale di abilitazione può essere diretto o negato (con pallino) a seconda della logica desiderata.

In questo modo è possibile realizzare un buffer bidirezionale di I/O per le memorie RAM utilizzando due invertitori tri-state collegati in parallelo e in modo opposto e controllati da due segnali di abilitazione diversi. Un invertitore sarà orientato per l'ingresso dei dati nella memoria (scrittura) e verrà abilitato quando si vuole scrivere, l'altro invertitore sarà orientato per l'uscita dei dati dalla memoria (lettura) e verrà abilitato quando si vuole leggere. Quando nessuno dei due invertitori è abilitato, il bus di I/O è in alta impedenza e non interferisce con altri dispositivi collegati.

18 Scaling e consumo

18.1 Storia e trend di evoluzione

Breve storia

- 1925: prima idea del transistor a effetto di campo (FET)
- 1949: realizzazione del primo transistor bipolare
- anni 60: prime porte logiche, primi mos e prime idee di circuiti integrati con mos
- anni 70: ic nmos 4004 e 8008 di Intel (il 4004 aveva 2300 transistor e clock <1MHz)
- anni 80: ic cmos 8086 di Intel (30000 transistor, 5MHz, $3\mu m$, 16 bit, 5V)
- anni 2000: microprocessori Pentium 4 di Intel (42 milioni di transistor, 1GHz)
- anni 2005: processori dual core (Core 2 Duo di Intel)

Legge di Gordon Moore

"Le prestazioni dei processori e il numero di transistor ad esso relativo raddoppiano ogni 18 mesi"

— Gordon Moore, 1965

Evoluzione delle specifiche dei processori

Per mantenere tale trend di crescita (specialmente il numero di transistor), è stato necessario agire su diversi fattori in tempi diversi in modo che, quando si raggiungono i limiti fisici di un parametro, si possa intervenire su un altro. I principali fattori sono:

- frequenza di clock (saturato intorno al 2005)
- consumo di potenza (saturato nei primi 2000)
- numero di core logici (in aumento)
- prestazioni single thread (quasi in saturazione)
- evoluzione delle tecnologie di fabbricazione, es. dimensione mos (prossimo a saturare)

Consumo di potenza

Il consumo di potenza è diventato un fattore limitante per l'evoluzione dei processori a partire dai primi anni 2000, in quanto con la diffusione dei dispositivi mobili, era necessario mantenere il consumo energetico entro certi limiti per garantire una buona autonomia della batteria. Inoltre un elevato consumo di potenza comporta un aumento del calore emesso e si richiede dissipatori sempre più efficienti.

Dimensioni dei transistor

L'evoluzione delle tecnologie di fabbricazione, in particolare la riduzione delle dimensioni dei transistor, ha permesso di aumentare il numero di transistor per unità di area, di ridurre le capacità parassite riducendo ritardi e consumi e di ridurre le tensioni di alimentazione riducendo i consumi.

Integrazione 3D e moduli multichip

Per mantenere in crescita il numero di transistor si adottano tecniche di integrazione 3D o moduli multichip, rispettivamente impilare più wafer interconnessi verticalmente o affiancare più chip su un unico package.

18.2 Strategie di scaling

Scaling a V_{DD} costante

Si mantiene costante la tensione di alimentazione V_{DD} e si riducono le dimensioni W , L e t_{OX} dei mos di un fattore α . Si ottengono le seguenti variazioni:

parametro	scaling	parametro	scaling
$W, L t_{OX}$	$1/\alpha$	t_p	$1/\alpha^2$
V_{DD}	1	P_{dyn}	α
Area	$1/\alpha^2$	P_{dyn}/Area	α^3
Capacità	$1/\alpha$	$E \approx V_{DD}/L, V_{DD}/t_{OX}$	α
I_{DSAT}	α	$C V_{DD}^2$	$1/\alpha$

Le criticità di questo tipo di scaling sono l'aumento dei consumi per unità di area che inducono ad un surriscaldamento elevato del chip e alla necessità di dissipatori più efficienti.

Scaling a campo E costante o full scaling o full scaling

Si mantiene costante il campo elettrico E e si riducono le dimensioni W , L e t_{OX} dei mos di un fattore α . Si adotta il modello a canale lungo. Si ottengono le seguenti variazioni:

parametro	scaling	parametro	scaling
$W, L t_{OX}$	$1/\alpha$	t_p	$1/\alpha$
V_{DD}	$1/\alpha$	P_{dyn}	$1/\alpha^2$
Area	$1/\alpha^2$	P_{dyn}/Area	1
Capacità	$1/\alpha$	$E \approx V_{DD}/L, V_{DD}/t_{OX}$	1
I_{DSAT}	$1/\alpha$	$C V_{DD}^2$	$1/\alpha^3$

La criticità di questo tipo di scaling è l'aumento della tensione di soglia V_T dei transistor che porta ad un aumento esponenziale delle correnti di perdita aumentando il consumo statico. Le porte non risulterebbero totalmente spente.

Inoltre le dimensioni di V_{DD} aumentano e inducono ad un aumento della resistenza equivalente dei mos, aumentando i tempi di ritardo dei circuiti.

General scaling

Nello general scaling si sceglie di ridurre la tensione di alimentazione V_{DD} di un fattore k/α per evitare di avere aumenti nella tensione di soglia V_T e nella resistenza equivalente dei mos. Si adotta il modello a canale corto in quanto, per le dimensioni del canale, tanti effetti non si possono più trascurare. Si ottengono le seguenti variazioni:

parametro	scaling	parametro	scaling
$W, L t_{OX}$	$1/\alpha$	t_p	$1/\alpha$
V_{DD}	k/α	P_{dyn}	k^2/α^2
Area	$1/\alpha^2$	P_{dyn}/Area	k^2
Capacità	$1/\alpha$	$E \approx V_{DD}/L, V_{DD}/t_{OX}$	k
I_{DSAT}	k/α	$C V_{DD}^2$	k^2/α^3

In generale si adotta $k = \sqrt{\alpha}$. Le criticità di questo tipo di scaling sono il ritorno dell'aumento lineare del consumo per unità di area con α e di conseguenza la necessità di avere dissipatori più grandi ed efficienti.

18.3 Compromesso tra velocità e consumo

Parametri e tecniche di riduzione del consumo

In generale i parametri che si vuole ottimizzare sono:

- la latenza: velocità di elaborazione o tempo di risposta, utile per applicazioni real-time che richiedono risposte veloci e immediate
- il flusso: quantità di dati elaborati per unità di tempo, utile per applicazioni che richiedono l'elaborazione di grandi quantità di dati in tempi lunghi (streaming, elaborazione batch, ecc.)

Inoltre si cerca sempre di minimizzare il consumo energetico agendo sul consumo dinamico e statico agendo su:

- V_{DD} : riduzione di V_{DD} riduce il consumo dinamico, ma diminuisce la velocità
- V_{TN} : aumentando di V_{TN} riduce il consumo statico, ma diminuisce la velocità
- architettura: il parallelismo e pipeline aumentano il flusso e il consumo in proporzione

Esistono diverse tecniche intelligenti per ottimizzare i vari parametri sopra in riferimento al consumo:

tecnica impiegata	consumo dinamico	consumo statico
riduzione di V_{DD} impiego di V_{DD} multiple riduzione dinamica di V_{DD}	molto efficiente (agisce quadraticamente)	impatto limitato
riduzione di V_{TN} impiego di V_{TN} multiple variazione dinamica di V_{TN}	nessun impatto	molto efficiente
riorganizzazione logica e strutturale	molto efficiente	impatto limitato
modalità stand-by	poco efficiente	molto efficiente
clock gating	efficiente	molto efficiente

Riduzione globale di V_{DD} e compensazione con parallelismo e pipeline

La riduzione di V_{DD} provoca una riduzione della frequenza del clock. Per mantenere invariate le prestazioni servono misure compensative come l'ottimizzazione logica e strutturale (es. sommatore cla al posto di rc) o l'adozione di architetture con pipeline e parallelismo.

Un esempio di parallelismo può essere il raddoppio della rete logica in due reti logiche identiche, ma con registri che lavorano a fronti di clock opposti e l'uscita è selezionata da un multiplexer. In questo modo si può dimezzare la frequenza del clock e ridurre V_{DD} mantenendo invariate le prestazioni (il flusso). Il nuovo consumo è compreso tra il 30% e il 40% del consumo originale, ma serve il doppio dell'area.

Un esempio di pipeline può essere l'inserimento di registri tra le varie fasi di un datapath in modo che, mantenendo sempre la stessa frequenza di clock, si può ridurre la velocità di ogni fase agendo su V_{DD} . Il consumo risulta compreso tra il 50% e il 60% del consumo originale, ma serve più area per i registri aggiuntivi e si aumenta la latenza.

Tensione di alimentazione multiple e circuiti traslatori di livello

Una tecnica intelligente è ridurre V_{DD} in maniera selettiva solo per i blocchi che finiscono in anticipo l'operazione e che quindi rimangono inattivi in attesa degli altri blocchi. In questo modo si riduce il consumo dinamico senza penalizzare le prestazioni complessive del sistema. Tra due blocchi che lavorano a tensioni di alimentazione diverse serve aggiungere un circuito traslatore di livello che converte i segnali logici alti tra i due livelli di tensione.

Il circuito traslatore di livello da V_{DD1} a V_{DD2} è composto da 2 nmos e 2 pmos. Tutti i mos devono sopportare la tensione $\max\{V_{DD1}, V_{DD2}\}$ e almeno i due nmos devono avere $V_{TN} < V_{DD1}$. In pratica si hanno due nmos complementari controllati da V_{DD1} che controllano se portare l'uscita a GND e disattivare i pmos, oppure se attivare i pmos per portare l'uscita a V_{DD2} .

È possibile utilizzare tensioni di alimentazione multiple anche all'interno dello stesso blocco logico, ad esempio utilizzando una tensione più alta per le parti critiche del circuito che richiedono maggiore

velocità e una tensione più bassa per le parti meno critiche, in modo da ottimizzare il consumo energetico complessivo del blocco ed avere tutti i segnali pronti nello stesso istante (si ritardano quelli più veloci).

A livello pratico si è osservato che non ha senso utilizzare più di 2 tensioni di alimentazione diverse in un sistema, in quanto più linee di alimentazione richiedono maggiori interconnessioni e complessità di progettazione.

Riduzione dinamica di V_{DD} in funzione del carico computazionale - DVS, DTS

Il Dynamic Supply Voltage Scaling (DVS) consiste nel ridurre dinamicamente V_{DD} (e in proporzione la frequenza) quando il carico computazionale diminuisce e il processore non necessita di alte prestazioni. Il circuito di gestione della tensione è composto da un modulo simile alla CP + VCO per generare V_{DD} controllato da un comparatore tra un registro scritto dal SO e un oscillatore ad anello (che trasforma la tensione analogica in frequenza). Il comportamento di tale circuito non è lineare, ma è monotono crescente e ciò è sufficiente per il controllo. Il registro di controllo è gestito dal SO/BIOS o programmatore che imposta il livello di prestazioni richiesto in modo da garantire la minima frequenza per tutti i programmi in esecuzione. Nel caso di multicore, ogni core può avere il proprio circuito DVS indipendente.

Il Dynamic Threshold Voltage Scaling (DTS) consiste nel variare dinamicamente la tensione di soglia V_{TN} dei mos in funzione del carico computazionale. Per fare ciò si agisce sulla tensione di substrato V_{BS} dei mos, in modo da modificare V_{TN} in base alla formula dell'effetto body. Aumentando V_{BS} si aumenta V_{TN} e si riduce il consumo statico, mentre diminuendo V_{BS} si diminuisce V_{TN} e si aumenta la velocità. Per controllare V_{BS} si utilizzano due generatori di tensione per V_{BSn} (nmos) e V_{BSp} (pmos) per ogni modulo. Per poter far variare il potenziale di substrato degli nmos in maniera indipendente è necessario isolare il substrato p degli nmos circondandoli da una barriera di ossido n (substrato dei pmos). Tale processo produttivo è chiamato triple well. Per garantire il corretto funzionamento delle giunzioni pn tra i substrati p (nmos), n (involtucro) e p (wafer), è necessario che il potenziale del substrato p degli nmos $V_{BSn} \leq 0V$, che il potenziale del substrato n degli nmos (involtucro) $V_{BSp} \geq 0V$ ponendo $V_B = 0$ il potenziale dell'ossido p del wafer.

Modalità Stand-by e clock gating

La modalità stand-by consiste nello isolare completamente un blocco logico dall'alimentazione quando non è in uso, in modo da eliminare completamente il consumo statico e dinamico. Tale modalità è utile per blocchi che rimangono inattivi per lunghi periodi di tempo. Per fare ciò si utilizzano due transistor che collegano/isolano le piste di alimentazione V_{DD} e GND generali dalle piste di alimentazione virtuale del blocco logico. Si nota che se sono circuiti di memoria, il contenuto dei registri va perso.

Il clock gating consiste nel disabilitare il clock di un blocco logico quando non è in uso, in modo da evitare di commutare i suoi flip-flop e alzare/abbassare tutte le piste di distribuzione del clock, riducendo il consumo. Per disabilitare il clock si usano porte TG combinate opportunamente con dei PLL in quanto le TG creano ritardi e cambi di carico capacitivo sulla linea di clock principale.

19 Circuiti logici programmabili

19.1 Introduzione ai circuiti logici programmabili

Tipi di programmazione (hardware/software/circuitale)

- **microprocessori:** si programma un microprocessore (hardware generico) attraverso un linguaggio di programmazione software, per risolvere un determinato problema, il vantaggio è la facilità di programmazione, la possibilità di modificare il programma e adattarlo/migiorarlo nel tempo, lo svantaggio è la lentezza di esecuzione e l'elevato consumo dovuto alla presenza di un'architettura generale e non ottimizzata per il problema specifico
- **circuiti programmabili o configurabili:** circuiti logici che possono essere programmati a livello hardware dopo il processo di fabbricazione, il vantaggio è la velocità di esecuzione e il consumo ridotto rispetto ai microprocessori, lo svantaggio è la complessità di programmazione e la minore flessibilità, esempi sono i microcontrollori le matrici di porte logiche (PLA, PLS, PAL, PROM, GAL, CPLD) e le FPGA
- **circuiti dedicati (ASIC):** circuiti logici progettati e fabbricati per eseguire una funzione specifica e la programmazione avviene in fase di progettazione del circuito stesso, il vantaggio è la massima velocità di esecuzione e il minimo consumo dati dalla completa ottimizzazione del circuito per il problema specifico, lo svantaggio è l'elevato costo di progettazione e la scarsa flessibilità e aggiornabilità

Classificazione dei circuiti logici

- **CPU o microprocessori:** circuiti logici generici programmabili via software in grado di eseguire un'enorme varietà di funzioni logiche attraverso l'esecuzione di un programma memorizzato in memoria
- **circuiti logici programmabili:**
 - **microcontrollori:** microprocessori con periferiche integrate totalmente autonomi che non richiedono circuiteria esterna, si programmano via software caricando il firmware in memoria flash, si utilizzano principalmente in sistemi embedded dati il costo, le dimensioni e la potenza computazionale ridotti
 - **logica programmabile a matrice:** matrici di porte logiche con interconnessioni programmabili, possono essere programmati in fase di progettazione attraverso un'opportuna maschera di fabbricazione (mask-programmed, non riprogrammabili, veloci ed efficienti), oppure sul campo dall'utente (field-programmable, riprogrammabili, più lenti ed energivori); esempi sono PLA, PLS, PROM, PAL, GAL, CPLD
 - **logica programmabile a celle:** matrici di blocchi logici programmabili e interconnessi tramite una rete di interconnessioni programmabili, esempi sono le FPGA (riprogrammabili sul campo con linguaggi tipo VHDL/Verilog)
- **circuiti dedicati (ASIC):** circuiti in cui l'algoritmo che andrà ad eseguire il circuito è programmato a livello hardware e totalmente ottimizzato per massimizzare efficienza e velocità, esempi: Tensor Processing Unit (TPU)

Implementazioni di funzioni logiche

1. **logica cmos:** si calcola la funzione logica in forma minimale e si implementano le reti di pull-up e pull-down secondo i principi della logica statica complementare, sono molto veloci, efficienti e affidabili
2. **porte logiche elementari:** si sfruttano le porte logiche di base (NAND, NOR, AND, OR, NOT) già pronte per combinare gli ingressi e ottenere l'uscita desiderata, sono più lente e meno efficienti della logica cmos, utilizzano generalmente più transistor ma sono più semplici da progettare
3. **memoria ROM:** si memorizzano in una memoria di sola lettura (ROM) le uscite corrispondenti a tutte le possibili combinazioni di ingressi, sono molto lente e poco efficienti ma estremamente semplici da progettare (es. generatori di funzioni con decoder di riga per rapido accesso)

- realizzazione come somma di prodotti: si implementa la funzione logica come somma di prodotti (SOP) o prodotto di somme (POS) utilizzando una matrice di porte AND e OR (PLA, PAL) programmando opportunamente le interconnessioni desiderate, sono più lente e meno efficienti della logica cmos ma più semplici da progettare e realizzare in quanto esistono hardware già pronti per questo scopo

19.2 Logica programmabile a matrice

Introduzione e struttura base

Sono circuiti logici programmabili con la stessa struttura di base costituita da una matrice di interconnessioni programmabili che collegano un array di ingressi agli ingressi di una serie di porte logiche (AND/OR/NAND/NOR) che producono le uscite desiderate che a loro volta possono essere collegate agli ingressi di altre porte logiche (OR/AND/NOR/NAND) per implementare somme di prodotti o prodotti di somme.

PLA - Programmable Logic Array

Le PLA sono costituite da una matrice chiamata piano delle AND per configurare gli ingressi del modulo agli ingressi delle AND e un piano delle OR per configurare le uscite delle AND agli ingressi delle OR. Serve per implementare funzioni logiche in forma di somma di prodotti (SOP).

PLS - Programmable Logic Sequencer

Le PLS sono come le PLA, ma alcune uscite delle OR sono retroazionate nel piano delle AND (temporizzate da un flip-flop) per implementare circuiti sequenziali.

PAL - Programmable Array Logic

Le PAL sono come le PLA, ma solo il piano delle AND è programmabile, mentre il piano delle OR è fisso.

PROM - Programmable Read Only Memory

Le PROM sono come le PLA, ma solo il piano delle OR è programmabile, mentre il piano delle AND è fisso. Si chiama P-ROM perché il piano delle OR richiama la struttura di una memoria ROM-NOR.

CPLD - Complex Programmable Logic Device

Le CPLD sono circuiti logici programmabili complessi costituiti da più blocchi PLA/PAL all'interno di un unico chip. Ogni blocco può essere programmato indipendentemente e ha un bus di I/O strettamente personale per comunicare con l'esterno e un altro bus di dati condiviso con gli altri blocchi per comunicare tra di loro.

19.3 Logica programmabile a celle - FPGA

Struttura di un FPGA

Gli FPGA sono circuiti programmabili sul campo e sono utilizzate per implementare hardware configurabile. Sono più grandi, più complesse e più flessibili rispetto ai circuiti programmabili a matrice. Sono composte da:

- blocchi logici programmabili (CLB)**: blocchi di celle programmabili formate da porte logiche e flip-flop per realizzare funzioni logiche arbitrarie mediante un set di connessioni locali programmabili
- circuiti di I/O**: blocchi per la gestione delle comunicazioni tra l'interno dell'FPGA e l'esterno, hanno appositi buffer di ingresso/uscita per pilotare carichi capacitivi elevati
- interconnessioni programmabili**: rete di interconnessioni programmabili che collegano tra di loro i CLB e i circuiti di I/O organizzate con gerarchia di più livelli, inoltre comprendono anche le linee di distribuzione del clock
- memoria di configurazione**: memoria interna che memorizza la configurazione dell'FPGA, dei singoli blocchi

CLB (Configurable Logic Block) e LUT (Look Up Table)

Esistono tre tipi principali di CLB:

- **CLB basati su LUT** (Look Up Table):

le look up table sono memorie SRAM che fungono da tabelle di verità per implementare funzioni logiche combinatorie per ogni possibile combinazione di ingressi. Per accedere al valore corretto si usa un MUX controllato dai segnali in ingresso alla LUT. Un CLB basato su LUT è costituito da una o più LUT, da un FA (che riceve propagate, generate e il riporto) e un flip-flop per temporizzare l'uscita. Si nota che le LUT possono essere usate per implementare le funzioni di generate e propagate in modo da poter implementare un sommatore binario ottimizzato all'interno del CLB. Altimenti se viene ignorato il FA, le LUT possono implementare qualsiasi funzione logica combinatoria.

- **CLB con blocchi programmabili a MUX:**

in questo tipo di CLB sono presenti più mux a cascata controllati da segnali esterni per implementare funzioni logiche combinatorie a MUX come avviene quando si implementa una funzione logica con TG. L'uscita è sempre temporizzata con un flip-flop.

- **CLB con porte logiche programmabili:**

questi tipi di CLB si rifanno alla struttura delle PLA/PAL/... viste in precedenza

- **blocchi logici dedidati:**

alcuni FPGA includono blocchi logici dedicati per operazioni specifiche già implementate come sommatori, moltiplicatori, blocchi di generazione del clock, shift register, ram, ...

Blocchi di I/O

I blocchi di I/O servono per connettere l'interno dell'FPGA con l'esterno. Sono costituiti da buffer di ingresso/uscita bidirezionale (programmabile) e da un circuito traslatore di tensione per lavorare a tensioni diverse. Il funzionamento è molto simile a quello dei buffer di I/O visti per le memorie con invertitori three-state, in più si aggiungono dei flip-flop per temporizzare i dati in ingresso/uscita e un debole pmos e nmos di pull-up/pull-down per impostare uno stato definito all'ingresso quando la linea è flottante. Inoltre i blocchi di I/O possono essere configurati per agire come linea di ritardo programmabile per sfasare il segnale in ingresso attraverso dei buffer di ritardo programmabili.

Interconnessioni programmabili e di vario tipo

Esistono due tipi principali di interconnessioni:

- **interconnessioni locali:** collegano i CLB adiacenti tra di loro, permettono comunicazioni ad alta velocità tra CLB vicine, non impiegano interruttori o ripetitori programmabili
- **interconnessioni globali:** linee dedicate a collegare CLB distanti tra loro e le CLB con i circuiti di I/O, introducono ritardi e buffer ripetitori, attraversano le matrici di interruttori programmabili
- **interconnessioni speciali:** linee destinate alla distribuzione del clock (organizzate in modo gerarchico ad albero ad H) e ad altri segnali di controllo come interconnessioni con blocchi di memoria, sommatori, shift register, ..., possiedono dei buffer ripetitori unidirezionali o bidirezionali per pilotare carichi capacitivi elevati (es. linee di clock)

Nelle interconnessioni sono inclusi dei buffer di rigenerazione del segnale e un sistema di interruttori programmabili per collegare o scollegare le varie linee di interconnessione in modo da formare il percorso desiderato tra i vari blocchi logici.

L'organizzazione delle interconnessioni globali può essere:

- **gerarchica, ad albero:** le interconnessioni sono organizzate su più livelli, come la rete ad albero H di distribuzione del clock, ad ogni cambio di livello (o diramazione) è presente una matrice di interruttori programmabili
- **a isole:** i canali di interconnessione sono distribuiti attorno ad ogni CLB verticalmente e orizzontalmente, le intersezioni tra canali verticali e orizzontali sono implementate a matrici di interruttori programmabili

Gli interruttori programmabili possono essere implementati con un flip-flop o cella di memoria SRAM che controlla un nmos o una porta tg per collegare o scollegare due linee di interconnessione. I bit di

configurazione degli interruttori vengono memorizzati nella memoria di configurazione dell'FPGA. Per evitare degradazione del segnale alto trasmesso dagli nmos, si alimentano i transistor degli interruttori programmabili con tensione superiore a V_{DD} per ridurre tale problema.

Siccome usare pass-T potrebbe introdurre ritardi per l'elevata resistenza introdotta dai mos, per le piste lunghe si utilizzano interruttori con ripetitori (buffer) per rigenerare il segnale.

In qualsiasi caso le interconnessioni programmabili introducono ritardi e carichi capacitivi sulle linee di segnale (oltre al ritardo presente di base per le piste lunghe), che in una interconnessione fissa (hard-wired) sarebbero praticamente assenti.

Le matrici di interconnessione dividono in due tipi:

- **PSM - Programmable Switch Matrix:** matrici di interruttori programmabili che collegano più linee di interconnessione orizzontali e verticali tra di loro, implementano le connessioni con mux e buffer ripetitori
- **CB - Connection Block:** matrici di interruttori programmabili che collegano le linee di interconnessione a i pin di I/O e ai blocchi logici (CLB), implementano interconnessioni con pass-T o TG

Inoltre per ottimizzare i tempi di ritardo su lunghe distanze, si ha una gerarchia di interconnessioni globali che classifica le linee in base alla loro lunghezza:

- **a corto raggio:** linee di interconnessione che collegano CLB adiacenti e collegano due CB e PSM vicini, per ridurre i ritardi alcune piste saltano i CLB o PSM intermedi e si classificano in funzione della lunghezza del segmento senza interconnessioni (lunghezza 1: collega CLB/PSM adiacenti, lunghezza 2: salta un CLB/PSM, ...) e sono organizzate con un certo offset per permettere a ciascun CLB/PSM di avere accesso a più linee di corto raggio con lunghezze diverse
- **a lungo raggio:** linee di interconnessione che corrono per tutta la lunghezza del chip e vengono pilotate da appositi driver bidirezionali three-state, servono per trasmettere segnali globali o collegare CLB distanti con il minimo tempo di ritardo possibile

Configurazione delle FPGA

I bit di configurazione da memorizzare nelle LUT, nei flip-flop delle interconnessioni e nei blocchi di I/O vengono salvati in una memoria che generalmente può essere:

- **SRAM:** volatile, leggermente modificata in modo che l'uscita sia direttamente collegata alla parte di circuito che controlla, permette riconfigurazione dinamica (in esecuzione) e l'intero chip FPGA può essere prodotto usando le normali tecniche di fabbricazione CMOS, lo svantaggio è che la configurazione va persa quando si spegne il dispositivo e va ricaricata da una memoria esterna non volatile (es. EEPROM, Flash)
- **EEPROM/Flash:** non volatile, permette di memorizzare la configurazione anche a dispositivo spento e l'occupazione di area è minore (meno transistor), lo svantaggio è che la tecnologia di fabbricazione è più complessa e costosa per realizzare i floating gate

20 Hardware per alto sforzo computazionale - GPU, ASIC, TPU

20.1 Esempi di elaborazione dati ad alto sforzo computazionale

Elaborazione di immagini

L'elaborazione di immagini consiste in operazioni di convoluzione (prodotto di matrici) per applicazione di filtri. Equivale a calcolare matrici di somme di prodotti dati dalla sovrapposizione del kernel (filtro) sull'immagine originale.

Simulazione numerica

La simulazione numerica consiste nel trovare le soluzioni approssimate ad esempio di equazioni differenziali utilizzando metodi matematici iterativi. Anche in questo caso ci si riconduce sempre a prodotti di matrici.

Intelligenza artificiale

L'intelligenza artificiale (AI) si basa su reti neurali che simulano il funzionamento del cervello umano. L'elemento base di una rete neurale è il neurone artificiale, che riceve in ingresso un certo numero di segnali, effettua la somma pesata di tali segnali e restituisce in uscita un segnale ottenuto applicando una funzione di attivazione alla somma pesata. Tali operazioni di combinazione lineare degli ingressi con i relativi pesi possono essere rappresentate sempre come prodotti di matrici, ovvero somme di prodotti. La scelta dei pesi viene effettuata durante la fase di addestramento della rete neurale.

Il machine learning si occupa di sviluppare algoritmi di apprendimento automatico per l'addestramento delle reti neurali. Tale addestramento può avvenire in tre modi: supervisionato (con dati etichettati), non supervisionato (senza dati etichettati) e per rinforzo (con premi e punizioni).

Un esempio di applicazione della rete neurale è la regressione lineare, ovvero trovare i coefficienti di una funzione polinomiale di grado n che meglio approssima un insieme di punti sul piano cartesiano. Risolvendo il problema ci si riconduce a calcolare il prodotto di matrici.

Quando si dispone di un modello di rete neurale addestrato, si può procedere alla fase di inferenza, ovvero utilizzare la rete per effettuare previsioni su nuovi dati in ingresso. Il processo di inferenza è anch'esso basato su prodotti di matrici. Molte applicazioni richiedono tempi di inferenza molto rapidi, come ad esempio il riconoscimento vocale in tempo reale o la guida autonoma.

Ruolo dell'elettronica

L'elettronica si occupa di progettare e realizzare dispositivi hardware in grado di eseguire in modo efficiente e rapido le operazioni di elaborazione richieste dalle nuove frontiere dell'intelligenza artificiale, come ad esempio i prodotti di matrici. I dispositivi sono ad esempio sensori, processori e hardware dedicati, memorie e interfacce di comunicazione.

Date le elevate richieste di risposte sempre più rapide, si è passati da un modello cloud computing in cui i dati venivano inviati a server remoti per l'elaborazione, ad un modello edge computing in cui l'elaborazione avviene localmente sui dispositivi stessi, saltando il passaggio ai server remoti e riducendo così la latenza. È sempre più necessario disporre di hardware specializzato per l'elaborazione locale dei dati, come ASIC e TPU, realizzabili grazie alle moderne tecnologie di fabbricazione dei circuiti integrati.

20.2 GPU - Graphical Processing Unit

Architettura della GPU

La GPU è un processore ottimizzato per avere un altissimo flusso di dati ottenuto attraverso un elevato grado di parallelismo. Le GPU sono nate originariamente per l'elaborazione grafica (prodotto di matrici), ma si sono rivelate molto adatte anche per altri compiti come simulazione numerica, elaborazione di immagini mediche e intelligenza artificiale.

Il numero di core di una GPU può arrivare a diverse migliaia (alto grado di parallelismo), a differenza delle CPU (ottimizzate invece per bassa latenza) che ne hanno al massimo poche decine.

Un singolo dispositivo GPU è costituito da più Processor Cluster (PC), ognuno dei quali contiene più Streaming Multiprocessor (SM). Ogni SM contiene più core di elaborazione (ALU), raggruppati in blocchi, ciascuno con una propria cache SRAM e una propria Control Unit (CU) per la gestione del flusso di dati.

Questo perché CU decodifica una stessa istruzione che viene poi eseguita in parallelo da tutte le ALU del blocco. Le SM condividono una memoria condivisa (shared memory) ad alta velocità.

Consumi energetici delle GPU

Le GPU sono ottimizzate per le alte prestazioni per cui hanno consumi energetici elevati, che possono arrivare a centinaia di watt. Ogni GPU genera internamente il proprio clock di funzionamento, che può essere molto elevato per garantire alte prestazioni. Le GPU richiedono sistemi di raffreddamento dedicati per dissipare il calore generato.

HBM - High Bandwidth Memory

Per gestire l'elevato flusso di dati è necessario affiancare alle GPU delle memorie ad alta velocità e larghezza di banda, dette HBM (High Bandwidth Memory). Le HBM sono memorie DRAM 3D impilate verticalmente (fino a 8 livelli) per aumentare la densità di memoria ed è situata sullo stesso package della GPU per ridurre la lunghezza delle piste dati e indirizzi, aumentando così la velocità di trasferimento dei dati. I collegamenti tra GPU e HBM avvengono tramite due controller HBM situati sulla GPU e sulla DRAM connessi tramite un bus dati molto largo (fino a 1024 bit) che si trova su un interposer di silicio per ridurre ulteriormente la lunghezza delle piste.

20.3 FPGA - Field Programmable Gate Array

Vantaggi e svantaggi delle FPGA

I dispositivi FPGA sono già stati discussi in precedenza, implementano gli algoritmi in hardware e sono un compromesso tra alte prestazioni (migliori di GPU e CPU) e riconfigurabilità sul campo. Inoltre hanno consumi energetici inferiori rispetto alle GPU e costi più contenuti che li rendono adatti per edge computing. Il principale svantaggio è la complessità di progettazione e programmazione, che richiede competenze specifiche in hardware design per sfruttare appieno le potenzialità delle FPGA.

20.4 ASIC - Application Specific Integrated Circuit

Vantaggi e svantaggi degli ASIC

Gli ASIC sono circuiti integrati progettati per eseguire una specifica funzione o un insieme di funzioni in modo ottimizzato. Le funzioni e gli algoritmi sono implementati a livello hardware e non sono riconfigurabili o aggiornabili. Gli ASIC offrono le migliori prestazioni in termini di velocità e consumi energetici rispetto a CPU, GPU e FPGA, poiché sono progettati specificamente per l'applicazione desiderata. Tuttavia hanno costi di sviluppo elevati e tempi di progettazione lunghi, che li rendono adatti solo per settori di nicchia con grandi volumi di produzione.

20.5 TPU - Tensor Processing Unit

Introduzione alle TPU

Un esempio di ASIC è la TPU (Tensor Processing Unit) sviluppata da Google dal 2016 per l'addestramento e l'inferenza di reti neurali utilizzate nell'intelligenza artificiale. Le TPU sono ottimizzate per eseguire operazioni di prodotto di matrici di dimensione 128×128 di cui la matrice dei pesi si trova già precaricata nella rete combinatoria.

Sono hardware specializzato con elaborazione near-memory o in-memory (bassa latenza con la memoria), con elevata banda di memoria (attualmente 4096 bit) e basso consumo grazie all'uso di dati a bassa precisione (8 bit). La scelta di operare a bassa precisione è giustificata dal fatto che le reti neurali sono tolleranti agli errori e possono funzionare correttamente anche con dati approssimati.

La TPU è stata pensata per essere estremamente specializzata per l'AI e alcune scelte progettuali (come la bassa precisione) non la rendono adatta per altri tipi di elaborazioni generiche come simulazione numerica o elaborazione di immagini.

Struttura generale delle TPU

Una TPU è organizzata in Tensor Core (TC), ognuno dei quali contiene una Scalar Unit (SU) per il calcolo scalare (flussi di controllo, calcolo di indirizzi di memoria, ...), una Vector Unit (VU) per il calcolo vettoriale e quattro Matrix Multiplication Unit (MXU) per prodotto di matrici 128×128 attraverso array sistolici ottimizzati. Ogni TC si intrefaccia con una memoria di tipo HBM (High Bandwidth Memory) per garantire un elevato flusso di dati.

Architettura generale per il prodotto di matrici

Si suppone la moltiplicazione di matrici quadrate di dimensione $N \times N$. Il risultato della moltiplicazione è una matrice sempre $N \times N$ in cui ogni elemento è dato dalla somma di N prodotti. Per calcolare le somme di prodotti si utilizza un Processing Element (PE) costituito da un moltiplicatore, un sommatore e un registro di accumulo per memorizzare il risultato parziale. Si dispongono i PE in una struttura a matrice $N \times N$ in cui ogni PE calcola una determinata entrata della matrice risultato. Attraverso il calcolo in parallelo, in N cicli di clock si riesce ad avere il risultato.

Ci sono però delle criticità legate alla gestione del flusso di dati in ingresso e in uscita dai PE: tutti i PE di una stessa colonna (o riga) ricevono il ingresso il dato proveniente da uno stesso registro di ingresso che deve pilotare una pista molto lunga che attraversa tutta la matrice di PE. Inoltre, dopo aver calcolato il risultato, ogni PE deve inviare il risultato ad un registro di uscita, anch'esso collegato tramite una pista molto lunga. Queste lunghe piste generano ritardi non indifferenti ed un elevato consumo energetico. Per ovviare a questi problemi si utilizza una struttura ad array sistolico.

Array sistolico e architettura della MXU

Un array sistolico è una struttura di calcolo ideata alla fine degli anni '70 per ottimizzare il flusso di dati in operazioni di calcoli paralleli e che è passata in secondo piano con l'avvento delle GPU e degli FPGA. È stata riscoperta recentemente per l'implementazione efficiente del prodotto di matrici nelle TPU a seguito dell'aumento di interesse per l'intelligenza artificiale.

In un array sistolico, i PE sono sempre organizzati in una matrice $N \times N$, ma i dati in ingresso ad ogni PE non provengono da registri di ingresso esterni tramite lunghe piste, bensì dai PE adiacenti. In questo modo si riducono notevolmente le lunghezze delle piste di collegamento, i tempi di propagazione e i consumi energetici. I dati in ingresso alle righe e alle colonne della matrice di PE provengono da registri a scorrimento (shift register) che forniscono i dati in modo seriale, un dato alla volta, ad ogni ciclo di clock. I dati si propagano attraverso la matrice di PE in modo simile al flusso sanguigno (il cuore con le sistole "spinge" il sangue nelle arterie facendolo progredire un po' per volta ad ogni battito). In questo modo, dopo $3N - 2$ cicli di clock, i dati hanno attraversato ogni PE e gli accumulatori dei PE contengono il risultato voluto che è necessario leggere con una pista di uscita di lunghezza critica.

Si osserva che nelle operazioni di prodotto di matrici di reti neurali, la matrice dei pesi rimane fissa e costante per cui si può precaricarla già all'interno dei PE, eliminando così la necessità di fornire in ingresso tale matrice e farla transitare lungo le colonne. Al suo posto si fa transitare il risultato parziale calcolato da ogni PE lungo le colonne così da evitare di avere le piste lunghe per la lettura del risultato finale. In questo modo dopo sempre $3N - 2$ cicli di clock, si ottengono in uscita dalle colonne i risultati finali in sequenza. Non essendoci più la necessità di accumulare i risultati parziali nei PE, si può eliminare la retroazione del registro di accumulo trasformandolo in un semplice registro di output. I PE diventano da reti combinatorie a semplici reti sequenziali composte da un moltiplicatore, un sommatore e un registro di output.

Temporizzazione della TPU

Siccome i dati si propagano attraverso l'array sistolico "diagonalmente", le colonne più vicine all'ingresso ricevono i dati prima delle colonne più lontane e producono il risultato in output prima delle colonne più lontane. Per sincronizzare il flusso in uscita è necessario introdurre dei ritardi (buffering) nelle colonne più vicine all'ingresso in modo che tutte le colonne producano il risultato finale nello stesso ciclo di clock e possano essere lette e salvate in maniera più semplice.

Si osserva che ogni risultato che si propaga verticalmente lungo l'array sistolico è indipendente dagli altri risultati e dai valori precedentemente calcolati dai singoli PE. Infatti i PE sono semplici reti sequenziali e non hanno memoria da resettare come nel caso degli accumulatori in array sistolici tradizionali.

Quando è stato inviato l'ultimo dato in ingresso di una matrice, è quindi possibile iniziare subito ad inviare i dati della matrice successiva aumentando così il throughput dell'array sistolico (principio delle pipeline).

Numero dei componenti della TPU

La versione iniziale di matrice non sistolica per il calcolo del prodotto di matrici quadrate richiede numeri di transistor molto elevati per matrici grandi e numeri ad alta precisione. Le architetture sarebbero quindi molto grandi, complesse e costose, ma (a differenza delle TPU) si ha il vantaggio di una maggiore flessibilità di utilizzo per diversi tipi di calcoli.

dimensione matrice	int 8bit	int 16bit	float 64bit
64×64	$11 \cdot 10^6$	$44 \cdot 10^6$	$176 \cdot 10^6$
128×128	$40 \cdot 10^6$	$158 \cdot 10^6$	$632 \cdot 10^6$
256×256	$391 \cdot 10^6$	$1564 \cdot 10^6$	$6259 \cdot 10^6$

Nelle TPU, grazie all'architettura sistolica e alla riduzione della precisione a 8 bit, è sufficiente un numero di transistor molto inferiore, che consente di realizzare chip di dimensioni contenute e con costi di produzione più bassi. Indicativamente, una singola MXU contiene **dai 32 ai 41 milioni di transistor** a seconda delle ottimizzazioni adottate all'interno di moltiplicatori, sommatori o registri.