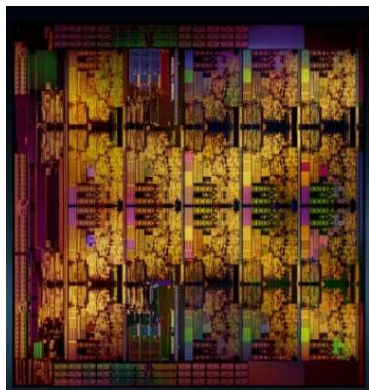
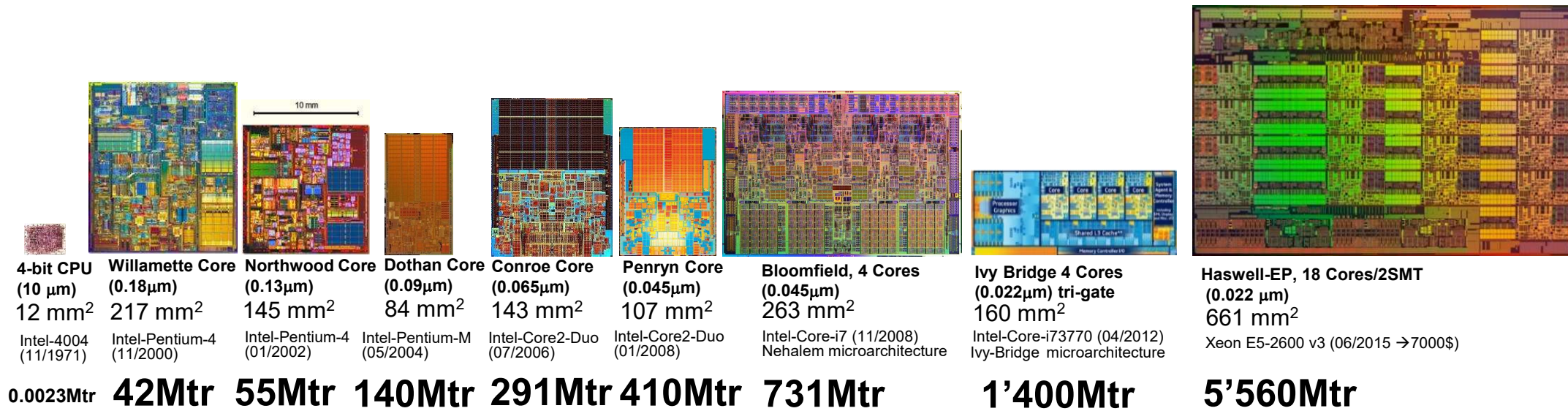
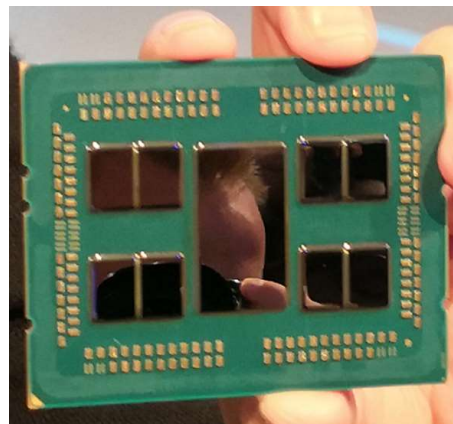


Architettura dei Calcolatori



Skylake-X, 18 Cores/2SMT
(0.014 µm)
473 mm²
Intel Core i9-7980XE (09/2017 → 1900\$)

??10'000Mtr



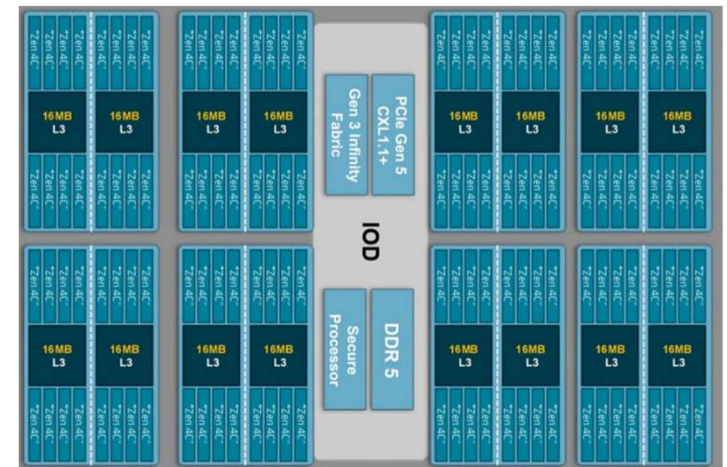
AMD Epyc Rome, 64 Cores/2SMT
(0.007 µm) → 9 chiplets x86, 1 chip I/O
1'008 mm² → CHIPLETS
AMD EPYC-7742 (08/2019 → 7000\$)

39'540Mtr



CERABRAS, 400'000 Cores
(0.016 µm)
46'225 mm² → WAFER-SCALE
CEREBRAS CS-1 (09/2019 → 2.5 M\$)

1'200'000Mtr



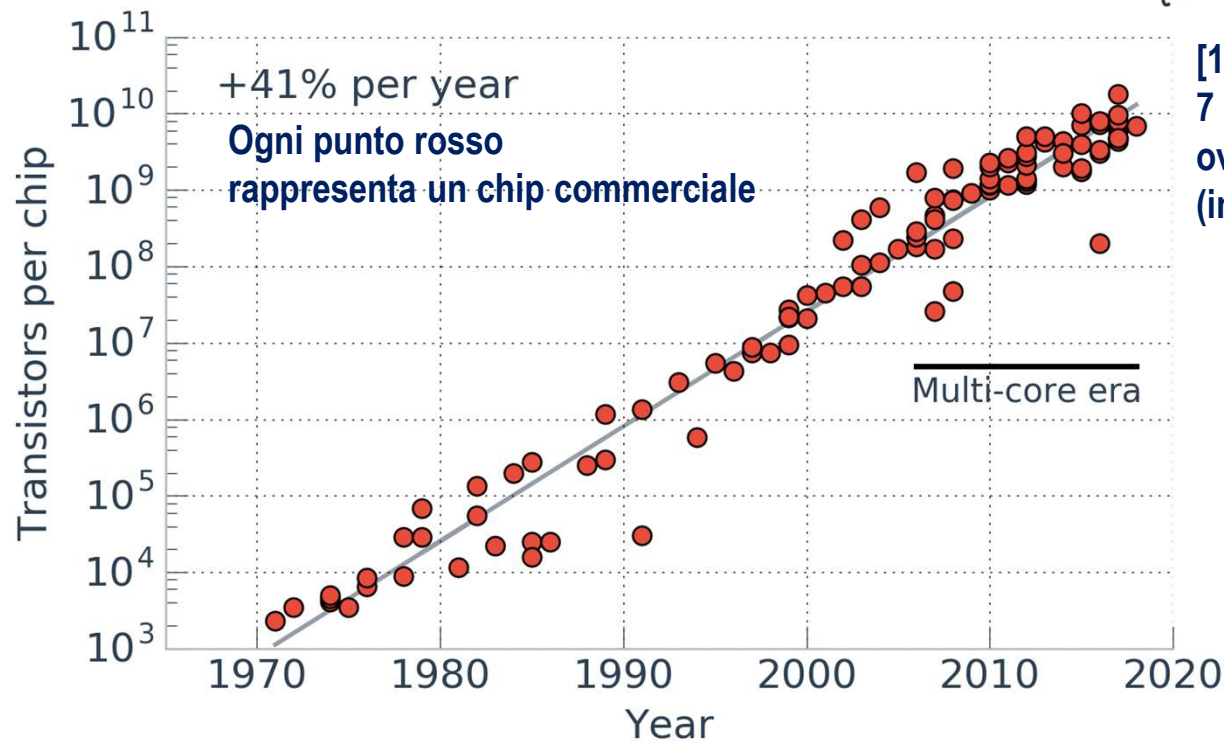
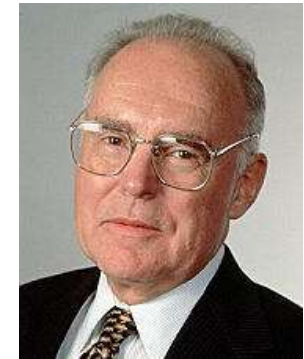
AMD Epyc Bergamo, 128 Cores/2SMT
(0.005 µm) → 9 chiplets x86, 1 chip I/O
700 mm² → CHIPLETS
AMD EPYC-9754 (06/2023 → 12'000\$) "Zen4c"

82'000Mtr

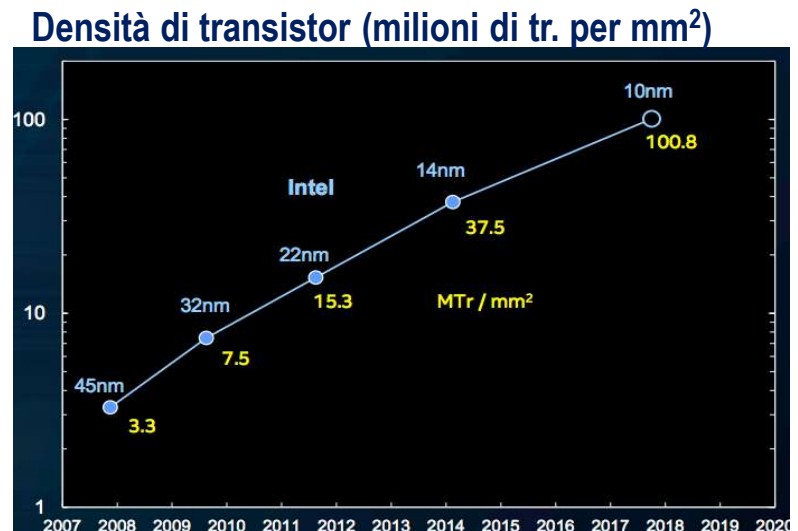
Vedi anche: Hennessy, J.L. and Patterson, D.A. 2019. **A new golden age for computer architecture.** Communications of the ACM. 62, 2 (Jan. 2019), 48–60. DOI: <https://doi.org/10.1145/3282307>

"Legge" di Moore *Uno dei Fondatori di INTEL*

- Il numero di transistor *integrati nel silicio* **RADDOPPIA** ogni **18 mesi** successivamente (modificato in "24 mesi") *si è ridotto un po'*
- Consentito sia da una maggiore densità che da chip di maggiori dimensioni



[1971-2016] $\rightarrow 46\text{anni} \cdot 12\text{mesi} = 552\text{mesi}$
7 decuplicazioni circa $\rightarrow$ decuplica ogni ~ 79 mesi
ovvero raddoppia ogni $79/\log_2(10) \approx 24$ mesi
(incremento annuo $10^{(7/46)} \approx 1.42 \rightarrow 42\%$ annuo)



Obiettivi del Corso e Come Raggiungerli

1) Capire l'architettura dei moderni calcolatori



Analisi dell'organizzazione interna dei principali elementi: processore, memoria, I/O, elementi di progettazione logica

2) Saper scegliere un calcolatore esaminando i parametri che ne influenzano le prestazioni



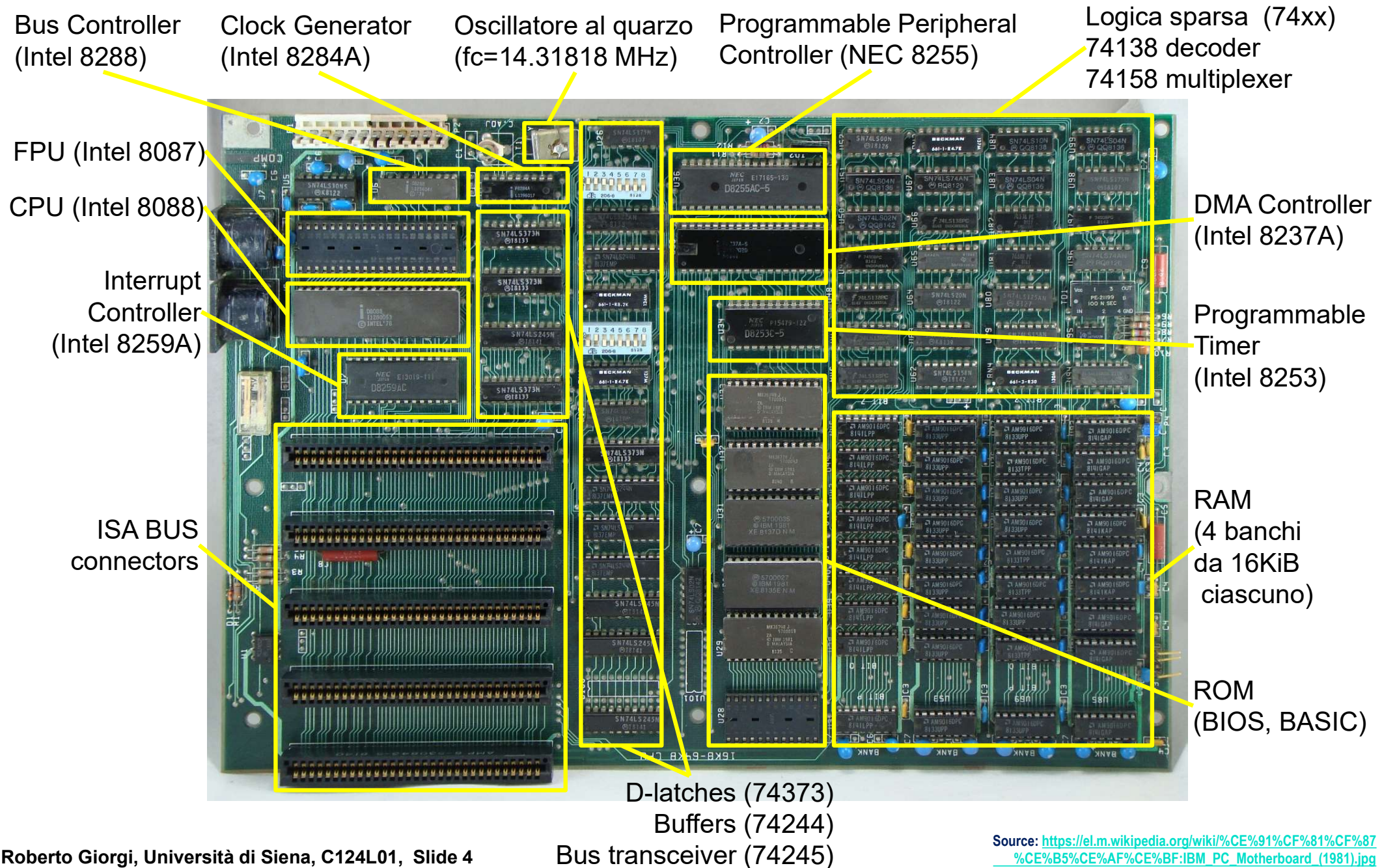
Analisi quantitativa dei fattori che influenzano le prestazioni e discussione su come le architetture incidono su tali fattori

3) Essere in grado di valutare l'efficacia dei meccanismi architetturali atti a migliorare le potenzialità dei calcolatori



Analisi di soluzioni architetturali non necessarie ma ormai universalmente presenti quali la memoria cache

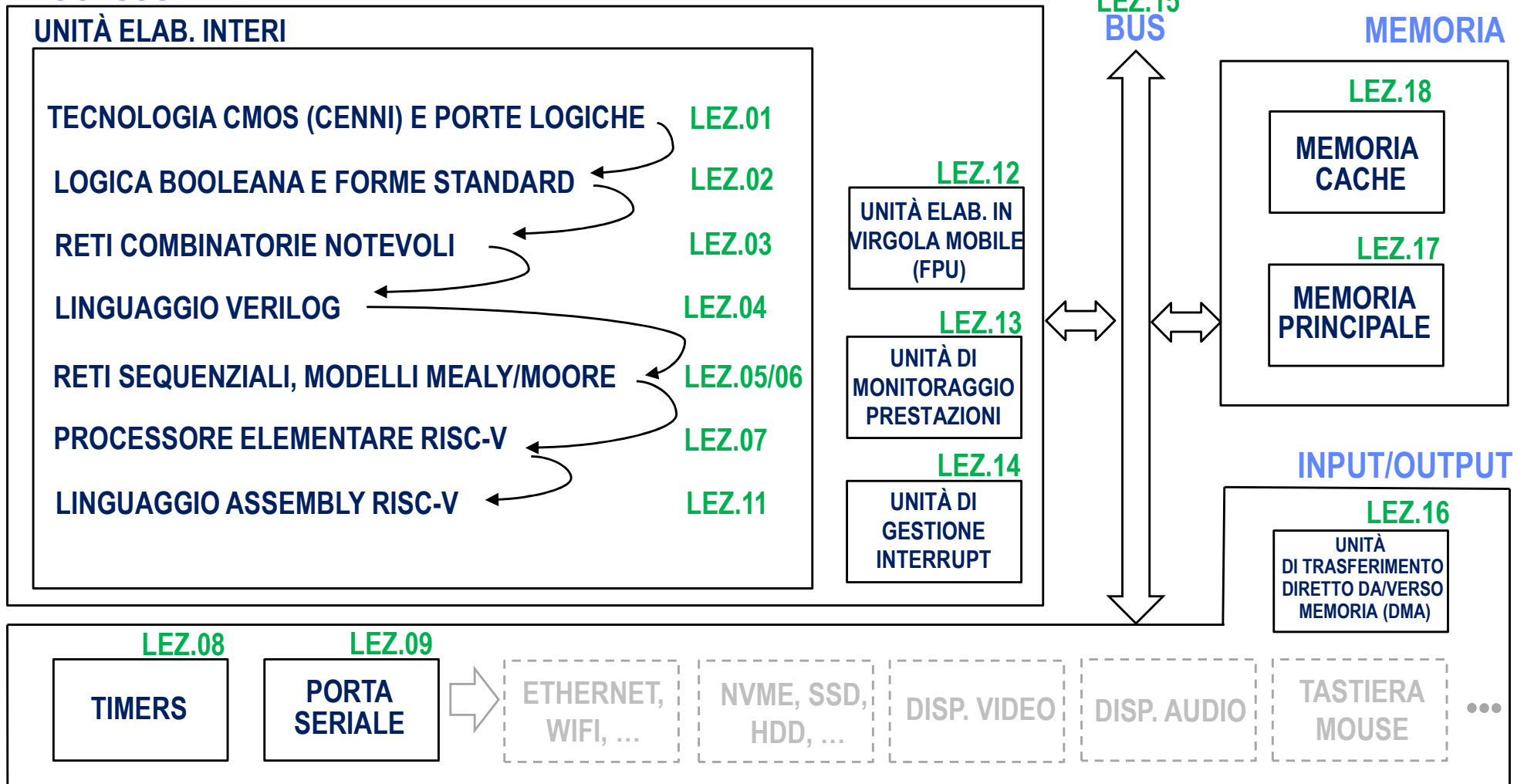
Il Personal Computer IBM-5150 (a.k.a. "PC" - Agosto 1981)



Analisi dell'architettura di un moderno calcolatore

- 4 elementi principali: processore, bus, memoria, input/output
 - Sempre gli stessi dal 1945: noti come «architettura di von Neumann»

PROCESSORE

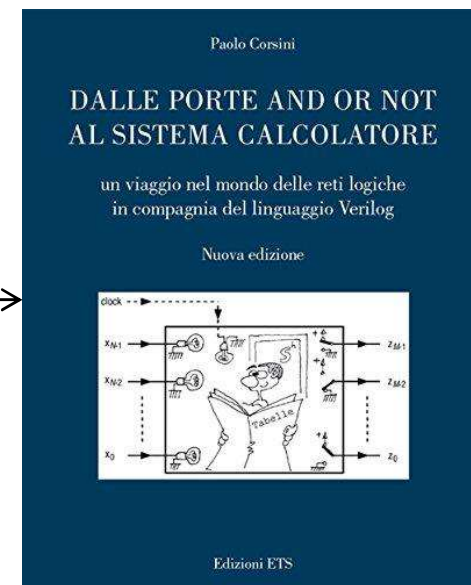


Conoscenze di base dai corsi precedenti

- Saper formalizzare un problema descritto in linguaggio naturale in modo che possa essere seguito dal calcolatore
- Saper leggere e saper scrivere semplici programmi C
- Struttura base della macchina
 - Modello di Esecuzione
ovvero principi di una macchina astratta in grado di eseguire programmi (ciclo fetch-execute, struttura di Von Neumann)
- Capire i passi che vengono compiuti per generare un programma:
 - compilazione, link, caricamento e esecuzione

Amministrazione del Corso

- Docente: Roberto Giorgi (giorgi@unisi.it) → NON ABUSARE)
- Telefono: 0577-23-5880
- Ricevimento: Lunedì 16:30/19:00
- Sito del corso: <https://arcal.diism.unisi.it>
- Testi di Riferimento:
 - Patterson and Hennessy, *Computer Organization and Design RISC-V Edition: The Hardware/Software Interface*, Morgan Kauffman, 2017
 - Edizione in Italiano (basata sulla precedente ed. in inglese):
Patterson e Hennessy, *Struttura e Progetto dei Calcolatori - Progettare con RISC-V*, Zanichelli, 2019
 - P. Corsini, "Dalle porte AND OR NOT al sistema calcolatore", Edizioni ETS, 2015 (o 2020)



Materiale Didattico (v. <https://arcal.diism.unisi.it>)

- Sono disponibili tutte le slide presentate durante le lezioni

N.B. LE SLIDE COSTITUISCONO UNA TRACCIA DELLA LEZIONE
CHE DEVE ESSERE INTEGRATA CON LA PARTECIPAZIONE ATTIVA ALLE LEZIONI,
CON LE ESERCITAZIONI E CON LO STUDIO PERSONALE

- Sono disponibili tutte le videolezioni dell'A.A. 2020-21
 - Il numero della lezione può variare rispetto all'A.A. attuale
- Per ogni lezione è indicato in quale parte dei testi di riferimento studiare l'argomento
- Sono disponibili moltissimi testi di compiti precedenti
 - molti dei quali svolti o con una traccia dello svolgimento e con molti programmi in assembly e Verilog da poter provare
- Sono disponibili tool didattici quali simulatori ed altro
 - per poter sperimentare personalmente quanto appreso
- Le informazioni sono aggiornate costantemente sul sito del corso

Formato dell'insegnamento

- **LEZIONI+ ESERCITAZIONI (totale 60 ore)**
 - Circa 30 ore di teoria, 30 ore di esercitazioni/prove_in_itinere
- Il programma segue molto da vicino i più elevati standard internazionali
 - <https://www.computer.org/cms/Computer.org/professional-education/curricula/ComputerEngineeringCurricula2016.pdf>
 - CE-CAO Computer Architecture and Design (circa 40 ore)
 - CE-DIG Digital Design (circa 20 ore)
- Useremo un approccio moderno per lo studio dei “Sistemi Digitali” con ampio riferimento al linguaggio Verilog
- Argomenti di Progettazione Architetture e Logica saranno intercalati durante lo svolgimento delle lezioni
- **In via SPERIMENTALE per il 2023-24** sono stati eliminati i seguenti argomenti (circa -20% rispetto dal 2022-23, -30% dal 2021-22):
 - Contatori
 - Memoria virtuale
 - Pipeline
 - Vari approfondimenti

Modalità di Esame - Syllabus Ufficiale

- Prova scritta seguita da orale
 - Sia la prova scritta che la prova orale consistono in domande aperte riguardanti gli argomenti svolti nelle lezioni frontali, documentati sia nei libri di testo, nelle diapositive illustrate e approfonditi nelle esercitazioni; la prova scritta viene svolta in laboratorio, ove ci si può avvalere del simulatore sia del linguaggio assembly che del linguaggio Verilog per una immediata verifica della correttezza dei corrispondenti esercizi; nella prova orale le domande sono del tutto analoghe a quelle dello scritto (e viceversa)
 - La prova scritta è superata con un voto non inferiore a 18/30. Il peso della prova scritta è il 50%, il peso della prova orale è il 50% del voto finale.
 - È caldamente consigliata la frequenza alle lezioni e esercitazioni; ove la frequenza non sia possibile, riferirsi al docente in caso di difficoltà
- Prove in itinere (novità dal 2020-21)
 - DUE prove in itinere
 - Il superamento delle prove scritte in itinere (media non inferiore a 18/30) permette l'esonero della prova scritta ai primi due appelli dell'anno accademico (la prova orale deve in ogni caso essere sostenuta)
- Progetto facoltativo
 - È possibile facoltativamente far valere un progetto di gruppo (max 3 persone) che contribuisce al voto complessivo fino a 5/30 in aggiunta al voto finale ottenuto con scritto e orale

Consigli per una miglior preparazione - Syllabus ufficiale

- Sia nelle prove scritte che orali (ma anche nei progetti), allo studente è principalmente richiesto di mostrare la conoscenza dettagliata dell'argomento, almeno al livello indicato dal docente durante la lezione. E' molto apprezzata la capacità di ragionamento sul problema, piuttosto che una meccanica (pedante) descrizione del tema
- Negli esercizi scritti, verrà principalmente considerata la correttezza della soluzione (in termini numerici) e una giustificazione molto breve del modo scelto per svolgere l'esercizio (lunghe formulazioni generali sono completamente inutili)
- Nell'interrogazione orale, l'argomento è in genere uno dei concetti illustrato durante la lezione. Elementi che sono richiesti sono, per esempio: la prova del concetto/teorema, schemi precisi del sistema, il comportamento e il funzionamento dettagliato, ragioni per le quali questa soluzione viene utilizzata nel mondo reale

Lezione 1

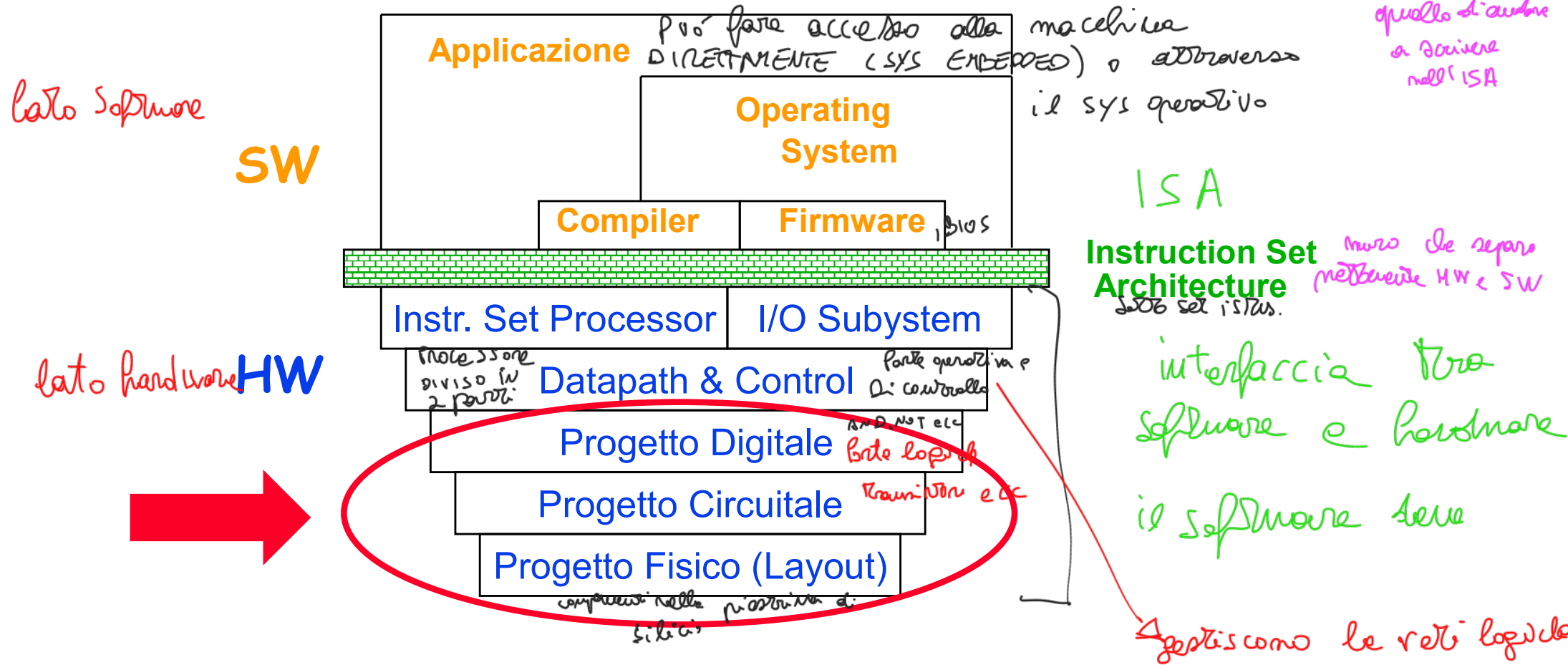
Dal transistor MOS alle Porte Logiche

<https://arcal.diism.unisi.it>

Dal livello fisico ai dispositivi logici

Calcolare strutture attraverso
vari livelli di astrazione

• Livelli di Astrazione di un Calcolatore



- Come sono correlati «progetto digitale» e «layout» ?
- Quali possibilità offre l'attuale tecnologia ?

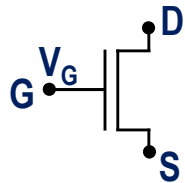
Il transistor MOS (Metal+Oxide+Silicon)

METALLO OSSIDO SILICIO

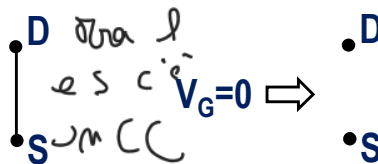
Transistor composto da 3 strati diversi

- Dettagli transistor MOS → insegnamento di Elettronica
 - Esistono di due tipi: NMOS e PMOS
- Possiamo assimilare il transistor MOS ad un interruttore
 - Quando la tensione di ingresso è alta l' NMOS **conduce**
 - Quando la tensione di ingresso è bassa l' NMOS è **interdetto**

MOS
Tipo N



se Tensione
su G = 1
 $V_G = 1 \Rightarrow$



tra l
e S c'è
 $V_G = 0 \Rightarrow$

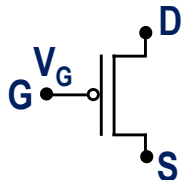


QUESTO E' CHIAMATO "NMOS"

Tensione su D e'
un interruttore

- Esiste anche il duale, chiamato PMOS
 - Quando la tensione di ingresso è alta il PMOS è **interdetto**
 - Quando la tensione di ingresso è bassa il PMOS **conduce**

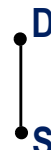
avviene l'opposto



$V_G = 1 \Rightarrow$



$V_G = 0 \Rightarrow$



QUESTO E' CHIAMATO "PMOS"

- Usando contemporaneamente transistor NMOS e dualmente PMOS si ottiene un circuito **CMOS (Complementary MOS)**

ANNI 60

Questi circuiti digitali degli anni 60

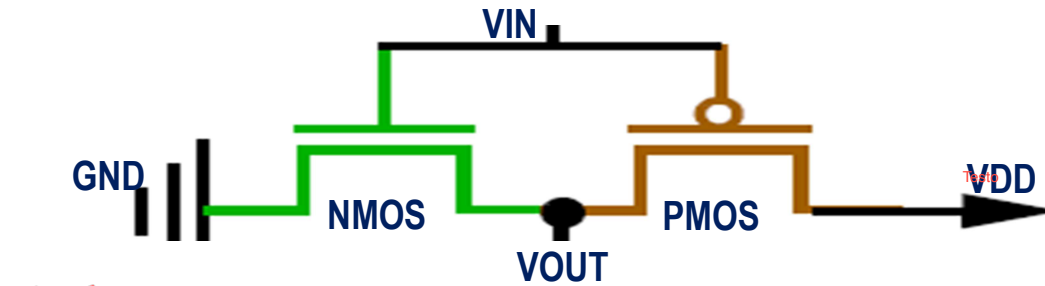
L'inverter CMOS

- Il circuito CMOS più semplice è l'inverter
 - Quando la tensione di ingresso è alta l'uscita è bassa
 - Quando la tensione di ingresso è bassa l'uscita è alta

N e P in serie

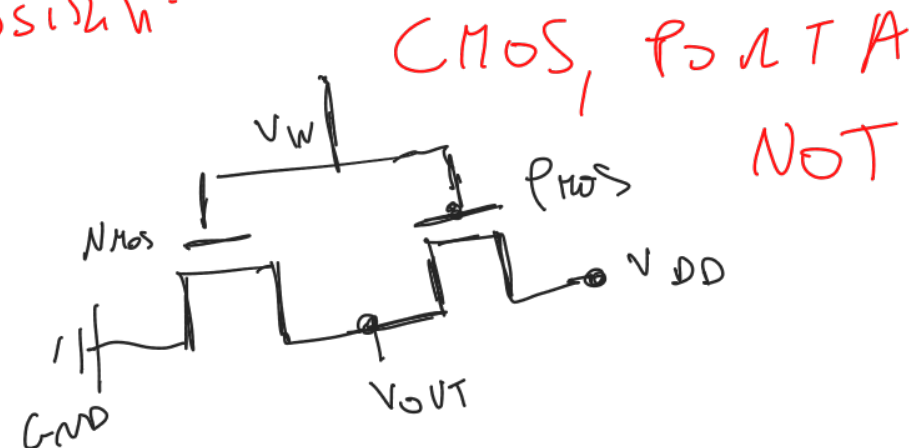
- Circuito elettrico:

e logico:



*Uscita presa tra
tra i due dispositivi*

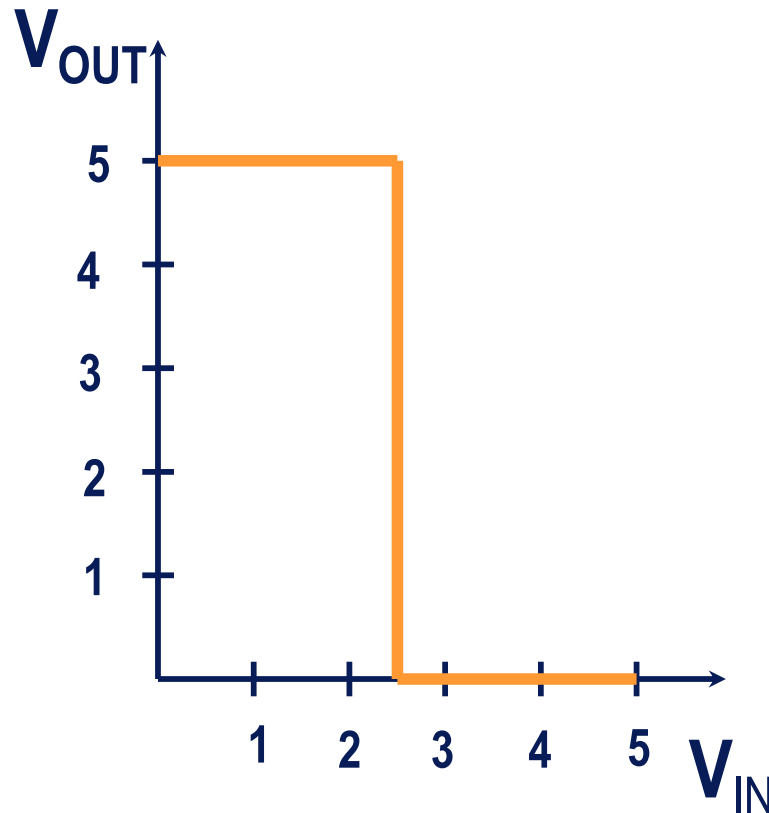
*$V_{IN} = 1, V_{OUT} = 0$
 $V_{IN} = 0, V_{OUT} = 1$*



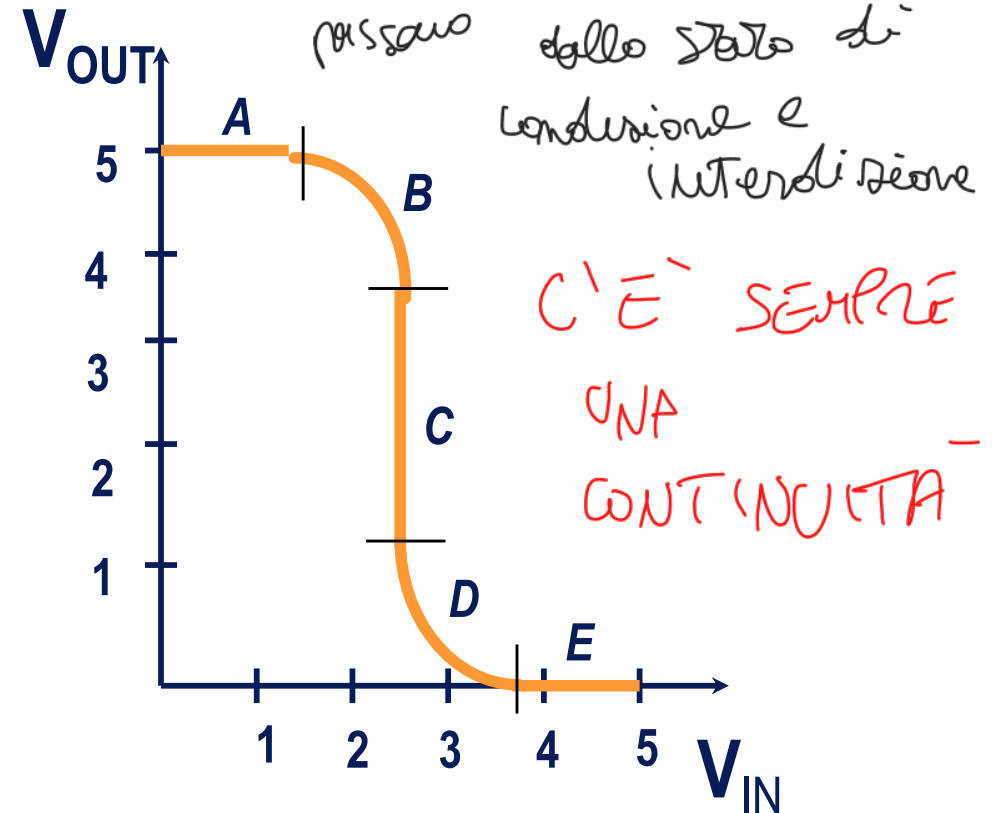
La transizione NEL TEMPO NON È A GRADINO, ha una ^{caratteristica} ~~ideale~~ ^{ideale}

CARATTERISTICA DI TRASFERIMENTO dell'INVERTER

Caratteristica IDEALE



Caratteristica REALE i due transist



Tratto di curva	Transistor NMOS	Transistor PMOS
A	Interdetto ($V_{GS} = V_{in} \rightarrow$ bassa ovvero $< V_{th,n}$)	Conduce ($I_D \propto V_{in}$) $\rightarrow V_{out}$ collegata a V_{DD} *
B	Saturo ($V_{DS} > V_{GS} - V_{th,n} \rightarrow I_D \cong$ costante)	Conduce ($I_D \propto V_{in}$) $\rightarrow V_{out}$ collegata a V_{DD} *
C	Saturo ($V_{DS} > V_{GS} - V_{th,n} \rightarrow I_D \cong$ costante)	Saturo ($V_{DS} < V_{GS} - V_{th,p} \rightarrow I_D \cong$ costante) **
D	Conduce ($I_D \propto V_{in}$ *) $\rightarrow V_{out}$ collegata a massa	Saturo ($V_{DS} < V_{GS} - V_{th,p} \rightarrow I_D \cong$ costante) **
E	Conduce ($I_D \propto V_{in}$ *) $\rightarrow V_{out}$ collegata a massa	Interdetto ($V_{GS} = V_{in} - V_{DD} \rightarrow$ bassa ovvero $> - V_{th,p} $)

Osservazioni

- La tecnologia CMOS consente di realizzare la funzione logica più semplice (inverter o porta NOT) in modo tale che la sua caratteristica di trasferimento sia vicina alla caratteristica ideale

- CI SONO ALTERNATIVE, TIPO SOLO NMOS*
Altre tecnologie (es. NMOS, BJT) hanno alcune limitazioni (NMOS: livelli logici non pieni, BJT: alto consumo) ma possono offrire alcuni vantaggi (NMOS: maggiore compattezza, BJT: maggiore velocità)

- Attualmente la tecnologia CMOS è la tecnologia di gran lunga più utilizzata per realizzare sistemi digitali soprattutto per la sua efficienza energetica

CON CMOS REALIZZO F. PIÙ COMPLESSE

- Per realizzare funzioni logiche più complesse ci viene in aiuto l'algebra booleana (v. lezioni successive)

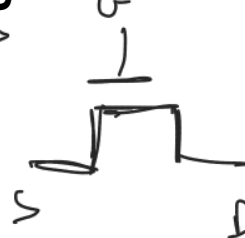
- In teoria oltre all'inverter ci servirebbero le funzioni elementari AND e OR
- In pratica, le porte più semplici da realizzare a partire dall'inverter sono le porte NAND e NOR (v. slide successive)

- Fortunatamente si può facilmente dimostrare che è possibile costruire qualsiasi funzione logica a partire anche dalle sole porte NAND (o NOR)

- È immediato altresì vedere che NAND seguito da NOT equivale a AND e NOR seguito da NOT equivale a OR

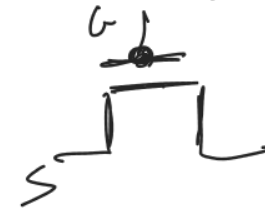
NMOS $G=1 \Rightarrow C.L.$
 $G=0 \Rightarrow G.A$

NMOS



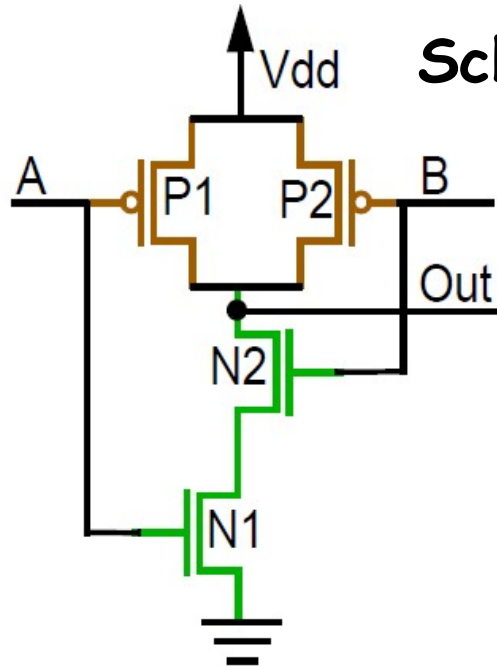
PMOS PMOS

$G=1, C.\theta$
 $G=0, C.C.$



Porta NAND CMOS

DA UNA NAND POSSIAMO COSTRUIRE
QUALSIASI F. LOGICA



Schema elettrico



Schema logico

NAND NOT AND

A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

- P in parallelo in ALTO
- N in SERIE

Tabella
di Verità

A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

Comportamento complesso
IN-OUT Formisue una tabella di
Verità del tipo

$$Q = \overline{A \cdot B}$$

Equazione booleana

$$Q = \sim(A \& B)$$

Verilog

USATA BASSA SÈ entrambi gli IN
sono alti, ciò posso scrivere in
EQUAZ. BOOL.

Porta NOR CMOS

2 Pin Serie, 2 N in Parallelo

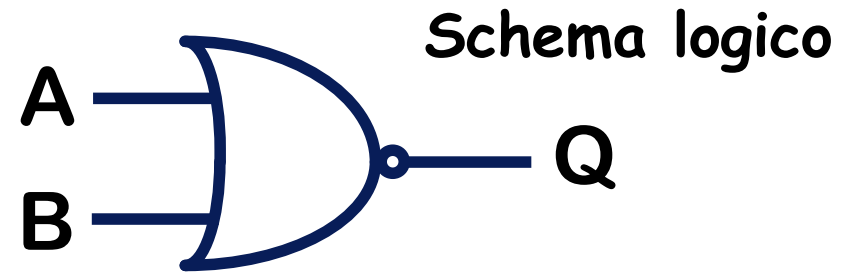
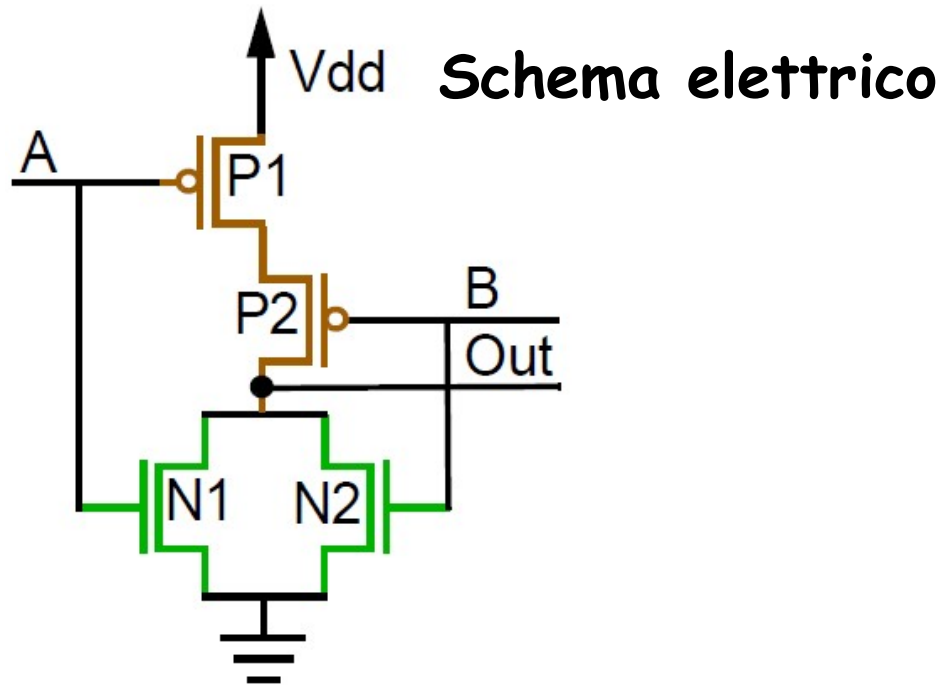


Tabella
di Verità

A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

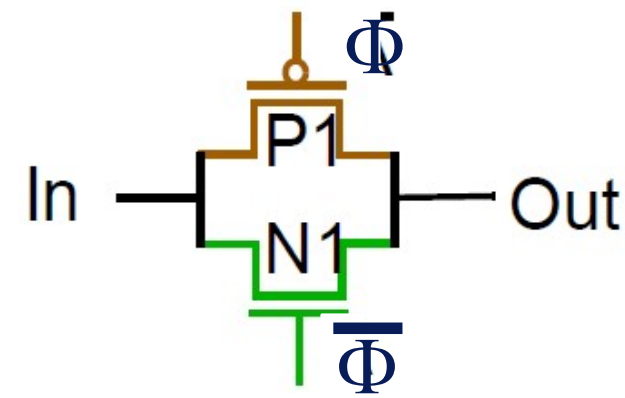
$$Q = \sim(A + B)$$

Equazione booleana

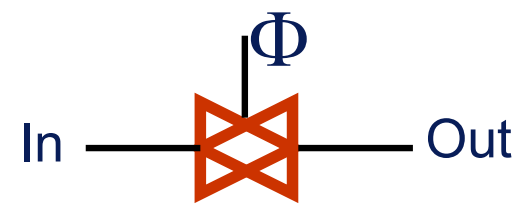
$$Q = \sim(A | B)$$

Verilog

Porta di passo/transito CMOS (transmission gate)



Schema elettrico



Simbolo

Nota: $\overline{\Phi}$ è sottinteso

Nota1: Φ e $\overline{\Phi}$ rappresentano le due fasi di un «clock a due fasi», ovvero di un'onda quadra Φ e della sua negata $\overline{\Phi}$ in cui però i due segnali non sono mai attivi contemporaneamente

ABILITO il camino tra in e out
e per farlo in modo efficiente
uso un clock a due parti

Tabella di Verità

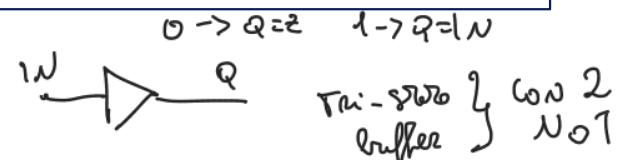
Φ	In	Out
0	X	Z ALTA IMPEDENZA
1	0	0
1	1	1

quando una Fase Φ è alta, l'altra è bassa. Non è un NOT perché entrambi sono sempre attivi

Nota2: non è un "tri-state" buffer in quanto quest'ultimo rigenera anche il segnale, mentre in questo caso il segnale si degrada un poco nel passaggio.

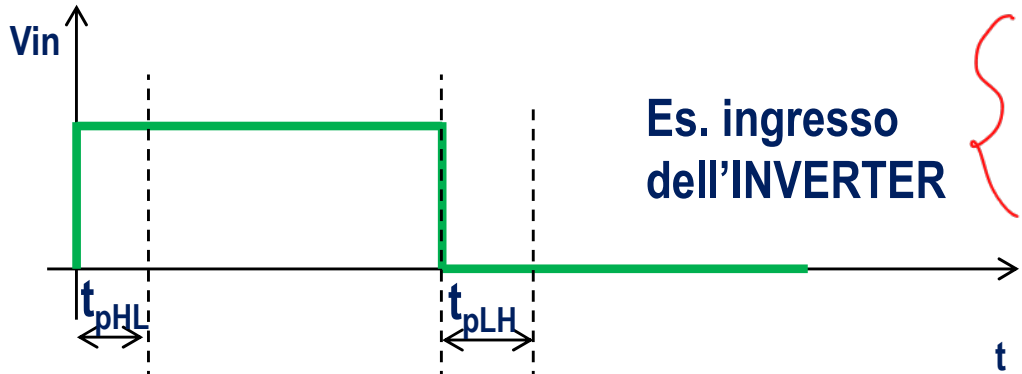
Nota2: viceversa si realizza un buffer mettendo due inverter in cascata

Se clock è alto, porta di passo conduce, ALTRIMENTI è INTERDETTO



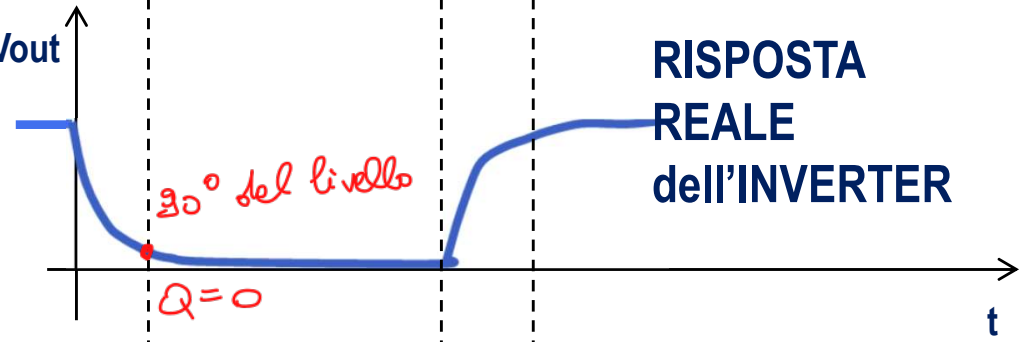
Ritardo di propagazione dell'inverter CMOS

- Le porte fondamentali e di trasmissione presentano un certo ritardo di propagazione t_p (t_{pHL} quando l'uscita va da High a Low e t_{pLH} viceversa)

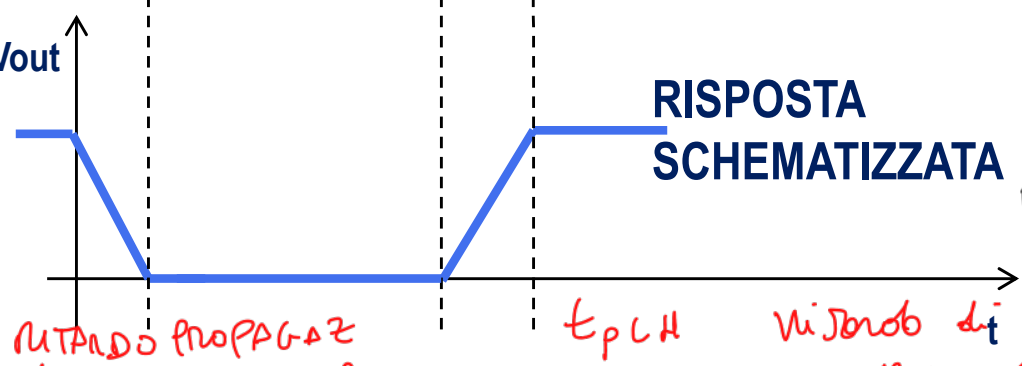


Es. ingresso dell'INVERTER

Applicando un segnale a gradino all'ingresso dell'inverter la forma d'onda di uscita raggiunge un livello accettabile (es. 90%) con un ritardo t_p . Il ritardo t_{pHL} necessario per la transizione dal livello alto a livello basso può essere diverso da quello t_{pLH} della transizione opposta
quando deciso se il segnale e' alto o basso?



RISPOSTA REALE dell'INVERTER



RISPOSTA SCHEMATIZZATA

Approfondimenti ulteriori possono essere fatti nei corsi di Elettronica. In questo contesto schematizzeremo il ritardo con una transizione lineare con durata finita
i tempi dipendono dai parametri fisici di costruzione

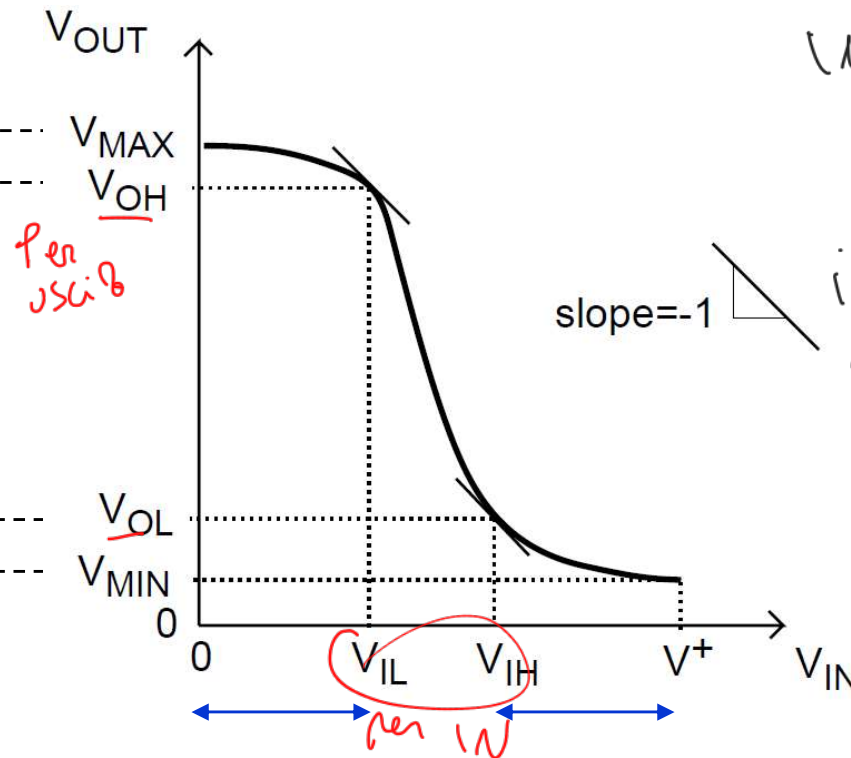
ritardo propag. da livello alto a basso
ritardo di propag. da livello basso a alto
 $V_{in}=0, Q=1$

Definizione dei livelli logici 1 e 0

risultando la caratteristica

IN uscita ha 3 casi

CIRCUITO
NON È
STABILE
influenza
caratteristica
dei circuiti successivi



IN OUT livello del NOT

individuare il punto in
cui pendenza curva diventa
 $= -1$ istante V_{IL} per IN
 V_{OH} per uscita

Intervallo di valori di ingresso
che produce un valido 1 logico

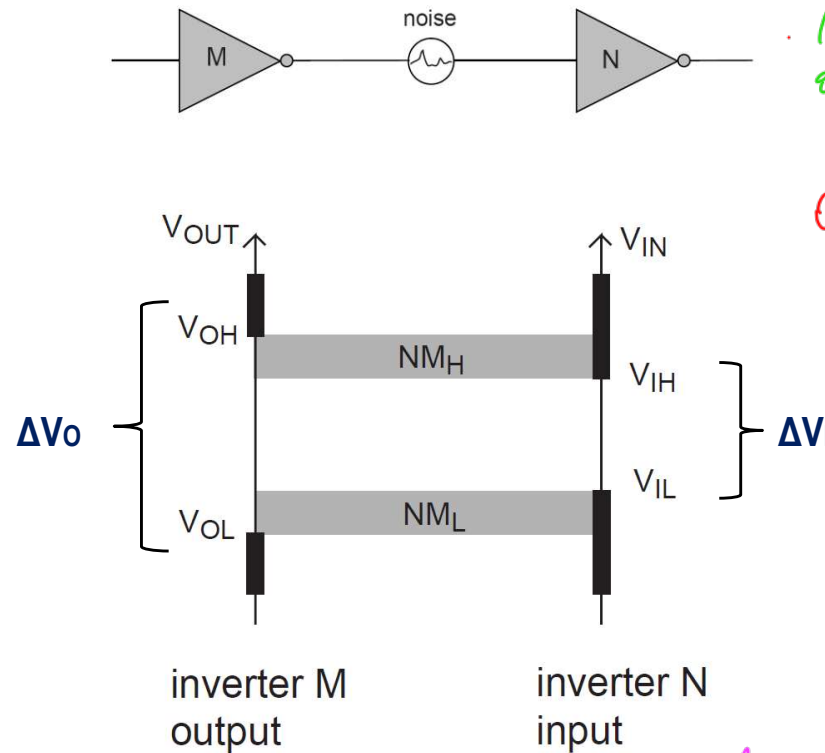
Intervallo di valori di ingresso
che produce un valido 0 logico

- **0 logico:** intervallo di valori di uscita compreso fra V_{MIN} e V_{OL}
 - V_{MIN} = tensione di uscita per la quale $V_{IN}=V^+$
 - V_{OL} = la più piccola tensione di uscita per la quale la pendenza è -1
- **1 logico:** intervallo di valori di uscita compreso fra V_{OH} e V_{MAX}
 - V_{OH} = la più grande tensione di uscita per la quale la pendenza $= -1$
 - V_{MAX} = tensione di uscita per la quale $V_{IN}=0$

fig circuiti e continuo bene,

Margine di rumore

- Se $(V_{OH}-V_{OL}) > (V_{IH}-V_{IL})$ l'inverter ha una certa **immunità al rumore**



NOISE MARGIN

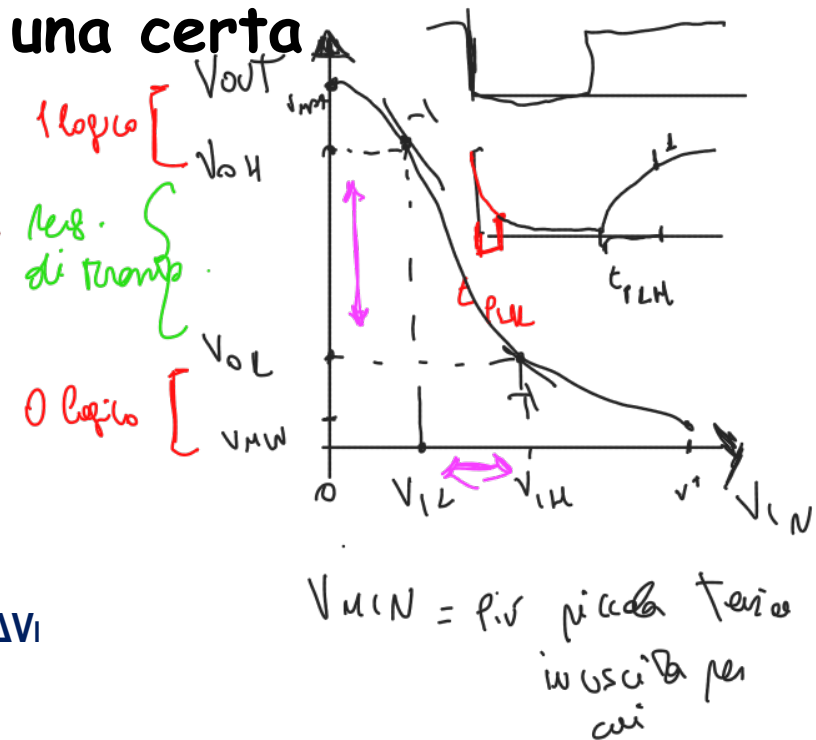
- $NM_H = (V_{OH}-V_{IH})$
- $NM_L = (V_{IL}-V_{OL})$

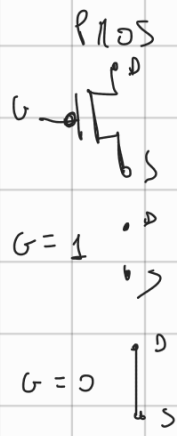
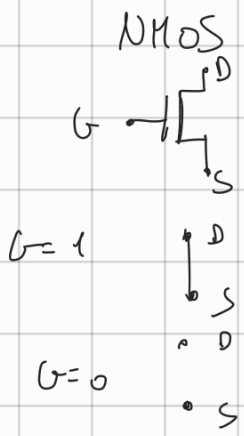
DIFFERENZA dei LIVELLI

margine di rumore sul livello alto

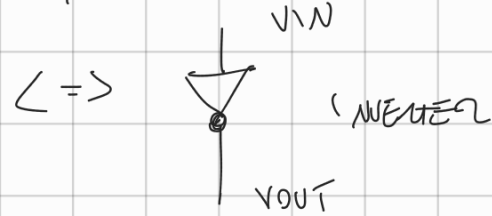
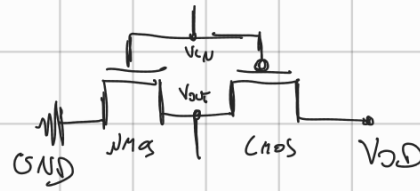
margine di rumore sul livello basso

più il margine è ampio





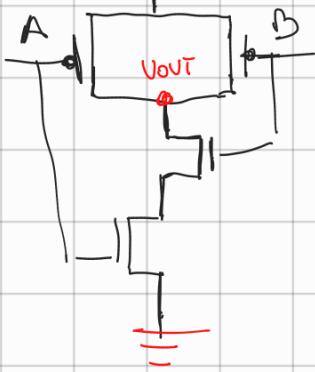
NMOS + PMOS = CMOS, INVERTER



$$V_{IN} = 1 \Rightarrow V_{OUT} = 0$$

$$V_{IN} = 0 \Rightarrow V_{OUT} = 1$$

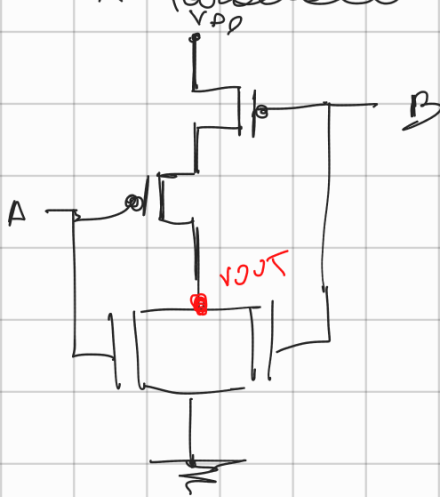
NAND, P parallel, N serie, com CMOS corutao poro logica



A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

NOR

N in serie
P in paralelo



A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

