

Facoltà di Ingegneria, Università degli Studi di Firenze



Calcolatori Elettronici

CDL in Ingegneria Elettronica

Nuovo Ordinamento
Slide del corso

Prof. Paolo Nesi
<http://www.dsi.unifi.it/~nesi>, nesi@dsi.unifi.it
AA 2005-2006
Ver:5.1

1

Calcolatori Elettronici, CDL Nu.ord.

- Parte 0: Introduzione
- Parte 1: L'algebra di Boole e i sistemi di numerazione
- Parte 2: Logica Combinatoria
- Parte 3: Logica Sequenziale
- Parte 4: Le architetture degli elaboratori
- Parte 5: L'architettura software e la scelta
- Parte 6: La CPU 8086
- Parte 7: Il Sistema di I/O
- Parte 8: Altri Aspetti ed evoluzione
- Parte 8bis: La programmazione
- Parte 9: La programmazione in Assembly, le istruzioni
- Parte 10: Procedure, Stack ed Interruzioni
- Parte 11: Esempi di Programmazione



Paolo Nesi, Univ. Firenze, Italy, 2003-06

2

Suggerimenti

- Le dispense/slide sono la traccia della lezione ma non sono il libro, si studia sul libro, e anche su altro materiale
- Potete accedere ad un sito web sul quale trovate tutto il materiale: <http://www.dsi.unifi.it/~nesi>
 - Regole per l'esame e compiti, etc.
 - Dispense, slide, etc.
 - Riferimenti bibliografici: lista libri consigliati, etc.
 - Strumenti software: simulatore di logica, etc.
 - Esercizi di Assembly, Assemblatore, etc.
 - Etc.
- Le slide saranno aggiornate e messe sul sito web in formato PDF via via che saranno aggiornate



Paolo Nesi, Univ. Firenze, Italy, 2003-06

3

Modalità di esame

- L'esame e' solo scritto, un compito
 - Risposte aperte e chiuse
- Compitino a meta' corso
 - circa la meta' dei contenuti, inciderà per circa la meta' del voto finale, una stima precisa sarà fatta in base al testo dello stesso.
 - Tutti possono partecipare: studenti in corso e/o degli anni passati e/o fuori corso, etc.
 - Supera il compitino solo chi prende un voto $\geq 18/30$
- A fine corso ci sarà la possibilità di fare:
 - il secondo compitino sulla seconda parte del corso solo, per chi ha superato il primo (il voto viene stimato come media pesata dei voti dei compitini), oppure
 - Il compito completo su tutto il programma del corso (il voto dipende solo da come viene svolto il compito completo)



Paolo Nesi, Univ. Firenze, Italy, 2003-06

4

Principale Libro di Testo



- G. Bucci
- McGrawHill



Paolo Nesi, Univ. Firenze, Italy, 2003-06

5

Calcolatori Elettronici

**CDL in Ingegneria Elettronica
Facoltà di Ingegneria,
Università degli Studi di Firenze**

Nuovo Ordinamento

Parte 0, Introduzione

Prof. Paolo Nesi

<http://www.dsi.unifi.it/~nesi>

nesi@dsi.unifi.it

2006



Paolo Nesi, Univ. Firenze, Italy, 2003-06

6

Cenni Storici

- Greci e Romani avevano già delle prime realizzazioni di macchine per automatizzare il calcolo
- 1642 appare la macchina di Biagio Pascal
- 1671 Leibniz presenta la prima macchina per le moltiplicazioni tramite addizioni successive
- 1823 Charles Babbage (1792-1871) presenta la prima macchina da calcolo di utilizzo generale senza però poterla completare. La meccanica del periodo non era sufficientemente sofisticata per lo scopo.
- 1946 ENIAC, la prima macchina da calcolo a valvole
- 1948 IBM 604, calcolatore a valvole/tubi
- 1951 UNIVAC, calcolatore a valvole/tubi
- 1971 INTEL 4004, il primo *Calcolatore Elettronico*
 - capace di lavorare con *numeri di 4 bit* ma cosa sono i bit ?

(a questo punto e' necessario andare a vedere cosa e' un calcolatore)



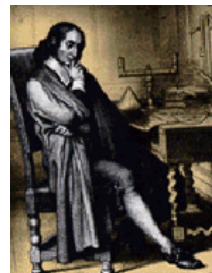
Paolo Nesi, Univ. Firenze, Italy, 2003-06

7

Cenni Storici

1642 -- Macchina calcolatrice di Pascal

- Dimensioni di una scatola per scarpe
- interazione con delle "rotelline" sulle quali erano riportati i numeri da 1 a 9.
- La novità rivoluzionaria: introduzione del principio del "riporto automatico".
- L'unico limite dell'invenzione di Pascal era che la macchina permetteva soltanto di eseguire addizioni e sottrazioni.



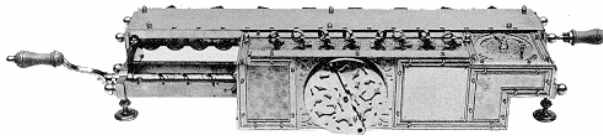
Paolo Nesi, Univ. Firenze, Italy, 2003-06

8

Cenni Storici(2)

1671 -- Modifiche apportate alla macchina calcolatrice da Leibniz

- Perfezionò la “calcolatrice” di Pascal inventando una macchina in grado di eseguire anche moltiplicazioni e divisioni.
- Erano macchine non programmabili ma dotate di programmi definiti dalla meccanica stessa della macchina.
- Fondatore del **sistema di numerazione binario** su cui si basa il funzionamento di tutti i computer moderni.



Paolo Nesi, Univ. Firenze, Italy, 2003-06

9

Cenni Storici (3)

1801 -- Jacquart inventa la scheda perforata

Invenzione di un telaio per la tessitura automatica che funzionava per mezzo di una serie di schede perforate dove, la posizione dei fori guidava il filo a formare un certo disegno nell'ordito.

Le schede erano la prima forma di programmazione di una macchina automatica.



Paolo Nesi, Univ. Firenze, Italy, 2003-06

10

Cenni Storici (4)

1823 -- Macchina di Babbage

Sfruttava il principio sia della macchina di Pascal sia del telaio di Jacquard.

Due serie di schede perforate: una serie costituiva il **programma** (le istruzioni), l'altra serie di schede rappresentavano i **dati** (i valori).

Viene considerato il "Padre dei computer", tanto che le sue idee sono alla base della moderna programmazione.



Introduzione alla programmazione

- Il calcolatore elettronico è uno strumento in grado di eseguire insiemi di **azioni** ("*mosse*") **elementari**
- le azioni vengono **eseguite** su oggetti (**dati**) per produrre altri oggetti (**risultati**)
- l'esecuzione di azioni viene richiesta all'elaboratore attraverso **frasi** scritte in qualche **linguaggio** (**istruzioni**)

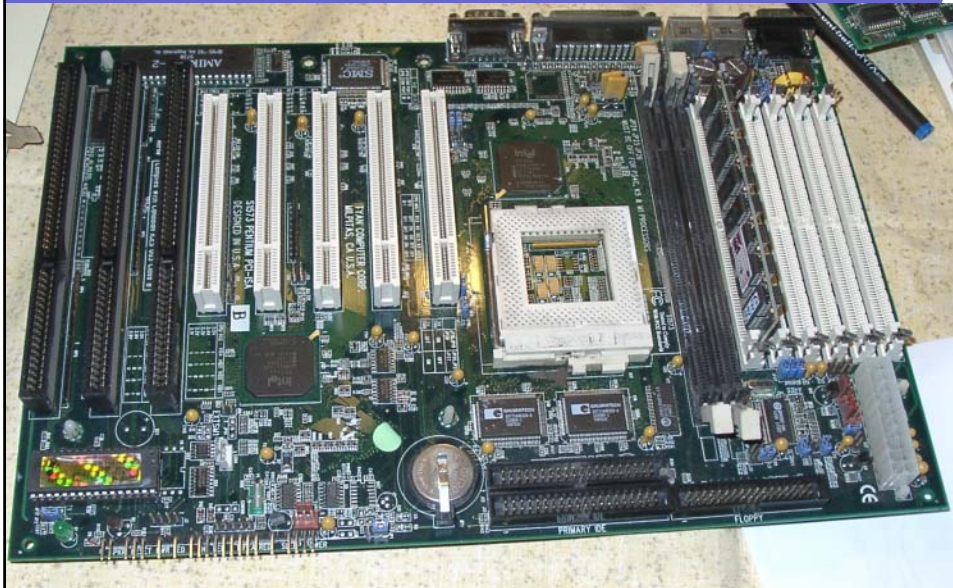


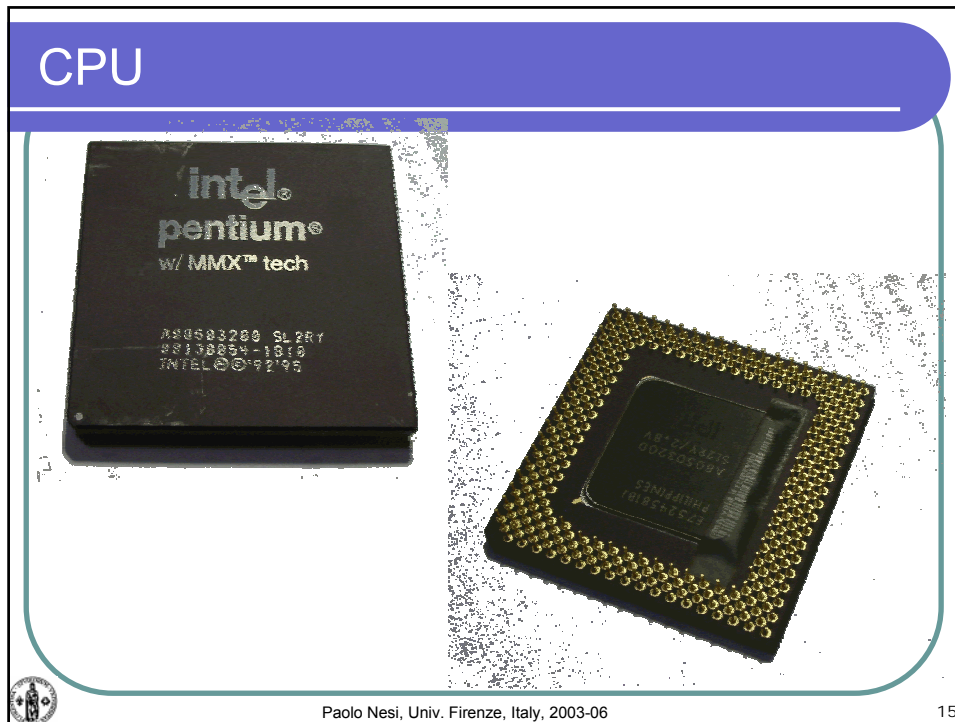
Programmazione

- È l'attività con cui si predispone l'elaboratore a eseguire un **particolare insieme di azioni su particolari dati**, allo scopo di *risolvere un problema*.



Scheda Madre

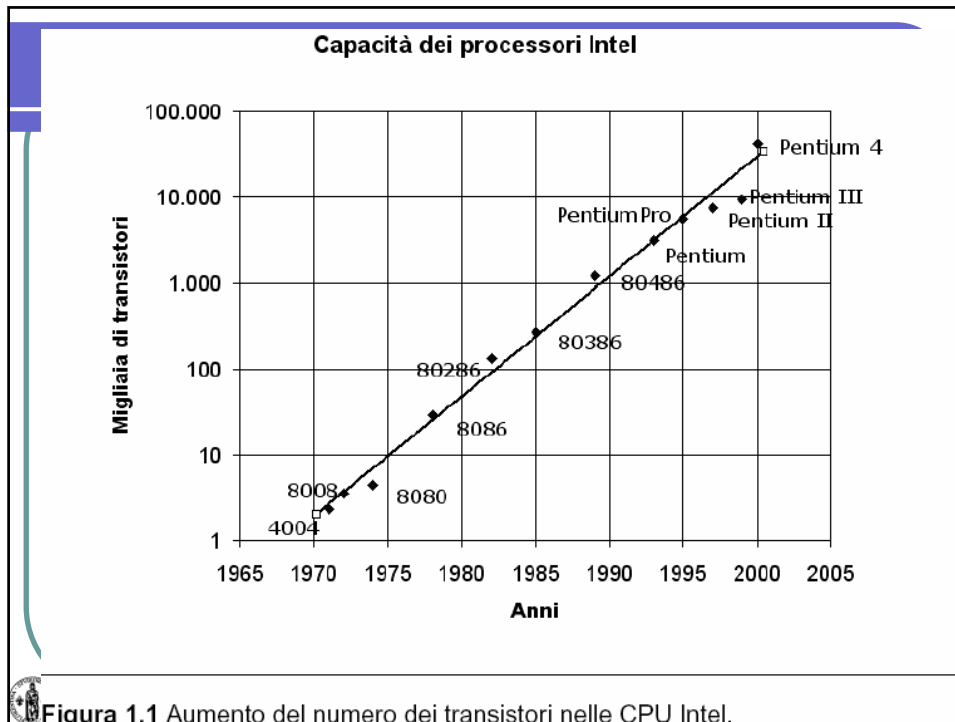




Evoluzione dei calcolatori

Data di introduzione	Nome del chip	N. di transistori (/1000)	Tecnologia (μm)	Frequenza (MHz)
Novembre 1971	4004	2,3	10	0,108
Aprile 1972	8008	3,5	10	0,500
Aprile 1974	8080	4,5	6	2
Giugno 1978	8086	29	3	5
Febbraio 1982	80286	134	1,5	8
Ottobre 1985	80386	275	1,5	16
Aprile 1989	80486	1.200	1	25
Marzo 1993	Pentium	3.100	0,8	60
Novembre 1995	PentiumPro	5.500	0,6	150
Maggio 1997	Pentium II	7.500	0,35	233
Febbraio 1999	Pentium III	9.500	0,25	450
Novembre 2000	Pentium 4	42.000	0,18	1400

Tabella 1.2 Aumento del numero di transistori delle CPU Intel. I dati riportati si riferiscono al modello di introduzione. Per i modelli introdotti in più versioni, la tabella riporta i dati relativi alla versione di più bassa capacità. Per esempio, il PentiumPro è stato introdotto in ben quattro versioni, di cui la meno potente (quella riportata) era tecnologia a $0,6 \mu\text{m}$ e frequenza pari a 150 MHz, mentre la più avanzata era in tecnologia a $0,35 \mu\text{m}$ e frequenza pari a 200 MHz.



Calcolatori Elettronici

CDL in Ingegneria Elettronica
Facoltà di Ingegneria,
Università degli Studi di Firenze

Nuovo Ordinamento
Parte 1a, L'Algebra di Boole

Prof. Paolo Nesi
<http://www.dsi.unifi.it/~nesi>
nesi@dsi.unifi.it
2006

Paolo Nesi, Univ. Firenze, Italy, 2003-06

ALGEBRA di BOOLE

- George Boole matematico inglese (XIX secolo)
- La sua algebra viene utilizzata solo dall'inizio del XX secolo (primi sistemi di calcolo)
- Si basa su due soli valori:
 - acceso (ON, vero, alto, true..)
 - spento (OFF, falso, basso, false, ...)



Variabili e Proposizioni

- Le variabili possono assumere solo due valori:
 - 0 L F
 - 1 H T
- Si chiamano *Variabili logiche* o *Booleane*
- Sono *proposizioni* le espressioni dell'algebra Booleana con certi operatori:

A or B and C



Costanti Booleane

- Oltre alle variabili vi sono anche le costanti
- Essendo l'Algebra Booleana definita su due soli simboli, esistono solo due possibili costanti:
 - 0 Falso basso....
 - 1 Vero alto....



Funzioni Booleane

- Usando le variabili Booleane, si possono costruire le funzioni Booleane

$$F(x,y,z)$$

- Funzioni che possono assumere solo due valori: 0,1
- Variabili Booleane e non:
 - $F(x,y,z) = X \text{ and } Y \text{ or } Z$
 - $F(x,y,z) = X > (Y + Z)$



Operatori

- Esistono due tipi di operatori, in relazione al numero di variabili che utilizzano:
 - Monadici, detti anche unari
 - Diadici, detti anche binari
- Questi possono essere
 - Logici
 - Confronto



Operatori di Confronto

- Maggiore: $>$
- Maggiore uguale: $>=$, \geq ,
- Minore: $<$
- Minore uguale: $<=$, \leq ,
- Diverso: \neq , $<>$, \neq , ...
- Uguale: $==$, $=$,

Il loro risultato e' un numero Booleano



Operatori Booleani

- Congiunzione: AND, e, &, \wedge , &&,
- Disgiunzione: OR, oppure, |, \vee , ||, +,
- Negazione o complemento:
 - NOT, not, no, !, -, *, \sim , \neg ,
 - Negato (A negato): \bar{A} , not A, *A,
- Implica: \rightarrow
- Co-implica: \leftrightarrow

25

L'operatore AND

- Il risultato è *vero* solo se sono *vere* entrambe le variabili

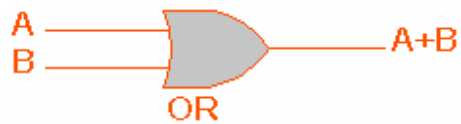
A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

26


L'operatore OR

- Il risultato è *vero* solo se è *vera almeno una* delle variabili

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1



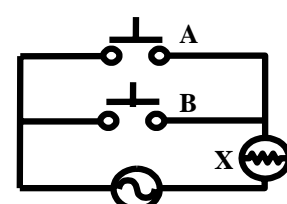
OR



Paolo Nesi, Univ. Firenze, Italy, 2003-06

27

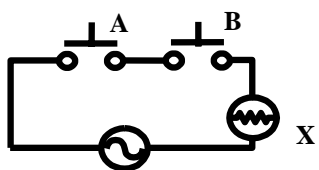
Similitudine con i circuiti elettrici



OR

A
B


A+B



AND

A
B

AB



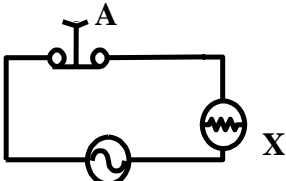
Paolo Nesi, Univ. Firenze, Italy, 2003-06

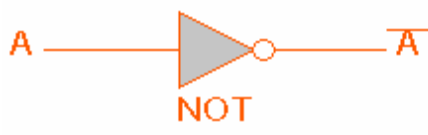
28


L'operatore NOT

- Il risultato è il complemento

A	NOT A
0	1
1	0






 Paolo Nesi, Univ. Firenze, Italy, 2003-06
29

Operatore Implica

- $A \rightarrow B$ che si legge A IMPLICA B
- Il termine **implica** non B e' vero (o falso) come potrebbe lasciare intuire il significato comune del termine implica.

A	B	-A	-A or B
0	0	1	1
0	1	1	1
1	0	0	0
1	1	0	1

 Paolo Nesi, Univ. Firenze, Italy, 2003-06
30

L'operatore Coimplica

- $A \leftrightarrow B$
- che si legge A COIMPLICA B e' equivalente a $A \rightarrow B$ and $B \rightarrow A$

A	B	-A	-B	-A or B	-B or A	-AorB and -BorA
0	0	1	1	1	1	1
0	1	1	0	1	0	0
1	0	0	1	0	1	0
1	1	0	0	1	1	1



Proprieta' degli operatori

OR	AND
$X \vee 1 = 1$	$X \wedge 1 = X$
$X \vee 0 = X$	$X \wedge 0 = 0$
$X \vee X = X$	$X \wedge X = X$
$X \vee -X = 1$	$X \wedge -X = 0$



Precedenze tra operatori

- Le precedenze sono simili alla somma '+' e al prodotto 'x' dell'algebra numerica consueta:
 - priorità alta, maggiore \times
 - priorità bassa, minore $+$
 - Si possono inoltre utilizzare tipicamente le parentesi tonde, (), per forzare la precedenza



Proprietà dell'algebra Booleana

$X \cdot 0 = 0$	$X + 1 = 1$	
$X \cdot 1 = X$	$X + 0 = X$	
$X \cdot X = X$	$X + X = X$	idempotenza
$X \cdot \overline{X} = 0$	$X + \overline{X} = 1$	complementazione
$X \cdot Y = Y \cdot X$	$X + Y = Y + X$	commutativa
$X \cdot (X + Y) = X$	$X + (X \cdot Y) = X$	assorbimento
$X \cdot (\overline{X} + Y) = X \cdot Y$	$X + (\overline{X} \cdot Y) = X + Y$	assorbimento
$X \cdot (Y + Z) = X \cdot Y + X \cdot Z$		
$X + (Y \cdot Z) = (X + Y) \cdot (X + Z)$		distributiva




Proprietà dell'algebra Booleana

$X \cdot (Y \cdot Z) = (X \cdot Y) \cdot Z = X \cdot Y \cdot Z$
 $X + (Y + Z) = (X + Y) + Z = X + Y + Z$
associativa

$\overline{(\overline{X})} = X$

$\overline{X \cdot Y} = \overline{X} + \overline{Y}$ $\overline{X + Y} = \overline{X} \cdot \overline{Y}$
De Morgan




Paolo Nesi, Univ. Firenze, Italy, 2003-06


35

L'operatore XOR

- Il risultato è *vero* solo se è *vera* solo una delle due variabili

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0





Paolo Nesi, Univ. Firenze, Italy, 2003-06

36

Espressioni logiche

- Un insieme di variabili e/o costanti Booleane a cui siano applicati gli operatori logici si dice *espressione Booleana o logica*
- Una espressione logica rappresenta una funzione logica, ad esempio:

$$T = a \cdot \bar{b} + \bar{a} \cdot b$$



Funzioni Booleane

- **Funzioni completamente specificate:**
se per tutte le combinazioni delle variabili il suo valore è determinato
- Esempio: puoi fare la frittata se hai
 - le uova,
 - una padella,
 - olio
 - sale



Esercizi - Semplificare le seguenti espressioni logiche

- 1) $WXYZ' + WXY'Z' + WX'YZ' + WX'Y'Z'$
 (WZ')
- 2) $(A + B'C)'$
 $(A'(B + C'))$
- 3) $ABC + AB'C + ABC'$
 $(A(C + B))$
- 4) $(X + Y' + Z)(X' + Y + Z')(X' + Y' + Z')$
 $(XZ' + X'Y' + Y'Z' + X'Z)$

$A' = \bar{A}$

Paolo Nesi, Univ. Firenze, Italy, 2003-06

4) $(X + Y' + Z)(X' + Y + Z')(X' + Y' + Z')$ $\times \begin{matrix} x & y \\ x & y \end{matrix}$

$(x\bar{x} + xy + x\bar{z} + \bar{y}\bar{x} + \bar{y}y + \bar{y}\bar{z} + z\bar{x} + zy + z\bar{z})(\bar{x} + \bar{y} + \bar{z})$
 $x\bar{y}\bar{x} + x\bar{y}y + x\bar{y}\bar{z} \quad x\bar{z}\bar{x} + x\bar{z}y + x\bar{z}\bar{z}$
 $\bar{y}\bar{x}\bar{x} + \bar{y}\bar{x}y + \bar{y}\bar{x}\bar{z} \quad \bar{y}\bar{z}\bar{x} + \bar{y}\bar{z}y + \bar{y}\bar{z}\bar{z}$
 $z\bar{x}\bar{x} + z\bar{x}y + z\bar{x}\bar{z} \quad z\bar{y}\bar{x} + z\bar{y}y + z\bar{y}\bar{z}$
 $x\bar{z} + \bar{x}\bar{y} + \bar{x}\bar{y}\bar{z} + \bar{y}\bar{z} + z\bar{x}$

Paolo Nesi, Univ. Firenze, Italy, 2003-06

Handwritten Karnaugh map for a 3-variable function. The variables are x , y , and z . The map shows the following minterms circled in red:

- $\bar{x}\bar{y}z$ (minterm 1)
- $\bar{x}y\bar{z}$ (minterm 2)
- $x\bar{y}\bar{z}$ (minterm 3)
- $xy\bar{z}$ (minterm 4)

Paolo Nesi, Univ. Firenze, Italy, 2003-06 41

Handwritten Boolean algebra derivation and logic circuit diagram:

$$\bar{x}\bar{y}z + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xy\bar{z}$$

$$\bar{z}(\bar{x} + y) + \bar{z}(x + \bar{y})$$

The circuit diagram shows three inputs x , y , and z connected to three AND gates. The outputs of these AND gates are connected to an OR gate, which produces the final output. The number 3 is written below the circuit diagram.

Paolo Nesi, Univ. Firenze, Italy, 2003-06 42


Esercizi - Semplificare le seguenti espressioni logiche

5) $A'B'C' + A'B'C + A'BC' + AB'C' + ABC'$ $A' = \bar{A}$
 $(A'B' + C')$

6) $A'B'C'D' + A'BC'D' + A'BCD + ABC'D' + ABC'D$
 $+ ABCD + ABCD' + AB'C'D'$
 $(C'D' + BCD + AB)$

7) $ABC + C(AB(D + A'(C'D + B) + B'D) + AB') +$
 $AB'C'$
 $(AC + AB')$


8) $((AB') (CD))' ((A'BC) (BD))' ((BC') (A + B + C))'$
 $(A'B' + ABC + CD' + B'C' + B'D')$ $A' = \bar{A}$



Paolo Nesi, Univ. Firenze, Italy, 2003-06 43

Funzioni Booleane

- **Funzioni non completamente specificate:** se a una o più combinazioni delle sue variabili non corrisponde alcun valore della funzione (dont care)
- Esempio: puoi fare la frittata se hai
 - le uova,
 - una padella,
 - Olio
 - ---



Paolo Nesi, Univ. Firenze, Italy, 2003-06 44

Tabella della verità

- Ogni funzione Booleana è caratterizzata dalla propria *tabella della verità*

x	y	z	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Paolo Nesi, Univ. Firenze, Italy, 2003-06 45

Operatori NAND/NOR

- Con gli operatori NOT, OR, AND, XOR si possono costruire tutte le funzioni Booleane
- Esistono due operatori (NAND, NOR) che permettono la sintesi di *qualsiasi funzione*, utilizzando un unico tipo di operatori

Paolo Nesi, Univ. Firenze, Italy, 2003-06 46

L'operatore NAND

- Il risultato è *vero* solo se è *falso* l'AND tra le due variabili

A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

Paolo Nesi, Univ. Firenze, Italy, 2003-06

47

L'operatore NOR

- Il risultato è *vero* solo se è *falso* l'OR tra le due variabili

A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

Paolo Nesi, Univ. Firenze, Italy, 2003-06

48

Tutto con NAND (NOR)

$A + B \equiv \text{NAND}(\text{NAND}(A, B), \text{NAND}(A, B))$

Paolo Nesi, Univ. Firenze, Italy, 2003-06 49

Tutto con i NAND a 2

$\text{Not } A = \text{NAND}(A, 1)$

$A B = \text{NAND}(\text{NAND}(A, B), \text{NAND}(A, B))$

$A + B = \text{not}(\text{not}A \text{ not}B) = \text{NAND}(\text{NAND}(A, B), \text{NAND}(A, B))$

Paolo Nesi, Univ. Firenze, Italy, 2003-06 50

Logica Positiva e Negativa

Voltage Levels

A	B	F
low	low	low
low	high	low
high	low	low
high	high	high

Positive Logic Levels

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

Negative Logic Levels

A	B	F
1	1	1
1	0	1
0	1	1
0	0	0

Physical AND gate

$F = A B$

$F = A + B$

Voltage Levels

A	B	F
low	low	high
low	high	high
high	low	high
high	high	low

Positive Logic Levels

A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

Negative Logic Levels

A	B	F
1	1	0
1	0	0
0	1	0
0	0	1

Physical NAND gate

$F = \overline{A B}$

$F = \overline{A + B}$

Proprietà operatori NAND/NOR

NOR (\uparrow)

$X \uparrow 1 = 0$

$X \uparrow 0 = \bar{X}$

$X \uparrow X = \bar{X}$

$\bar{X} \uparrow \bar{Y} = X \cdot Y$

$X \uparrow Y = X + Y$

NAND (\downarrow)

$X \downarrow 0 = 1$

$X \downarrow 1 = \bar{X}$

$X \downarrow X = \bar{X}$

$\bar{X} \downarrow \bar{Y} = X + Y$

$X \downarrow Y = X \cdot Y$

Data sheet: NAND x 2

SN7400 QUADRUPLE 2-INPUT POSITIVE-NAND GATES

description
These devices contain four independent 2-input NAND gates.

function table (each gate)

INPUTS		OUTPUT
A	B	Y
H	H	L
L	X	H
X	L	H

package (top view)

schematic (each gate)

absolute maximum ratings
 Supply voltage, VCC: 7 V
 Input voltage: 5.5 V
 Operating free-air temperature range: 0 °C to 70 °C
 Storage temperature range: - 65 °C to 150 °C

logic diagram (positive logic)

recommended operating conditions

	MIN	NOM	MAX	UNIT
V _{CC} Supply voltage	4.75	5	5.25	V
V _{IH} High-level input voltage	2			V
V _{IL} Low-level input voltage			0.8	V
I _{OH} High-level output current			- 0.4	mA
I _{OL} Low-level output current			16	mA
T _A Operating free-air temperature	0		70	°C

53

Forme Canoniche

- Prodotti di Somme
- Somme di Prodotti

a	b	c	f(a, b, c)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Prima Forma Canonica
 $a'b'c + a'bc + ab'c' + ab'c + abc$

(a+b+c) * (a+b'+c) * (a'+b'+c)

Seconda forma Canonica

Paolo Nesi, Univ. Firenze, Italy, 2003-06

54

Sintesi Somme di Prodotti

- Si identifica i termini che danno 1
 - A=0 allora not A
 - A=1 allora A
- $Y = \text{not } A \text{ and not } B + A \text{ and } B$

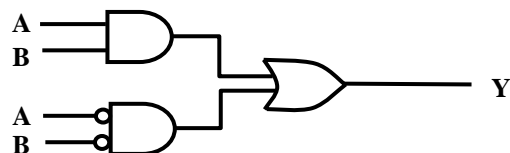
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1



Circuito: somma di prodotti

- $Y = \text{not } A \text{ and not } B + A \text{ and } B$
- Produrre la rete

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1



Sintesi Prodotti di Somme

- Si identifica i termini che danno 0, logica negata

- A=0 allora A
- A=1 allora not A

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

- $Y = (\text{not } A + B) \text{ and } (A + \text{not } B) =$

- $(\text{not } A + B) \text{ and } (A + \text{not } B) = \text{not } A \text{ and } A + \text{not } A \text{ and not } B + B \text{ and } A + B \text{ and not } B$

- in base alle proprietà di invarianza:

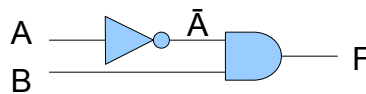
$\text{not } A \text{ and } A = 0, B \text{ and not } B = 0$, pertanto:

- $(\text{not } A + B) \text{ and } (A + \text{not } B) = \text{not } A \text{ and not } B + A \text{ and } B$



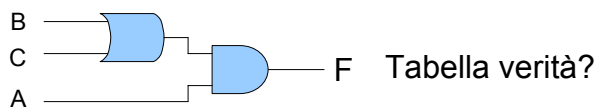
Sintesi circuitale di espressione logiche

- $F = \bar{A}B$



A	B	F
0	0	0
0	1	1
1	0	0
1	1	0

- $F = A(C+B)$



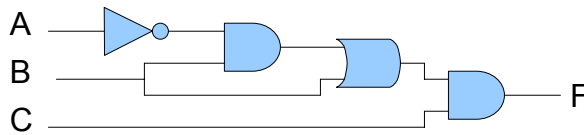
- $F = (A'B' + ABC + CD' + B'C' + B'D')$? ...

Circuito + Tabella verità?



Sintesi circuitale di espressione logiche

- F= ? Tabella della verità?

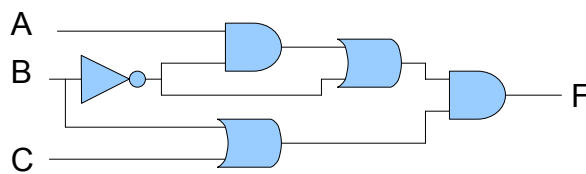


A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

$$F = C(B + (B\bar{A}))$$



Esercizi




- Calcolare F e scrivere la tabella della verità.



Operazioni Binarie su sequenze di bit

- AND sui bit
 - 101101 and
 - 010110 =
 - 000100
- OR sui bit
 - ...
- NOT sui bit
 -



Paolo Nesi, Univ. Firenze, Italy, 2003-06

62

Operazioni logiche sui bit

- AND sui bit

AND

10101010	
01011101	
00001000	

↑

- XOR sui bit

10101010	
01011101	
11101111	

↑


- OR sui bit

OR

10101010	
01011101	
11111111	

- NOT sui bit

10101010 ⇒ 01010101



Paolo Nesi, Univ. Firenze, Italy, 2003-06

63

Mascheratura sui bit, Operatore AND

ABCDEFG
 0100011 ← 1^a
 0001000

 000D000
 0

AND ↓

Mask

01001

01000

01000

↑

Paolo Nesi, Univ. Firenze, Italy, 2003-06 64

Combinazione sui bit, operatore OR

ABC

010

DEF

011

ABC ~~xxx~~
 000DEF

010000

000011

010011

ABC DEF

OR

Paolo Nesi, Univ. Firenze, Italy, 2003-06 65

Nibble, byte, word ...

Un bit rappresenta una cifra binaria.

Il bit però è un'unità di informazione troppo piccola per poter essere elaborata in modo efficiente.

I bit pertanto sono trattati secondo i seguenti gruppi:

1 nibble = 4 bit

1 byte = 8 bit

1 word = 16 bit

1 doubleword = 32 bit

1 Kilobyte = 2^{10} byte = 1024 byte = 8196 bit

1 Megabyte = 2^{20} byte = 1048576 byte ~ 8 milioni di bit

1 Gigabyte = 2^{30} byte ~ 1 miliardo di byte ~ 8 miliardi di bit

1 Terabyte = 2^{40} byte ~ 10^{12} byte ~ 2^{43} bit

1 Exabyte = 2^{50} byte



Calcolatori Elettronici

CDL in Ingegneria Elettronica
Facoltà di Ingegneria,
Università degli Studi di Firenze

Nuovo Ordinamento

Parte 1b, I Sistemi di Numerazione

Prof. Paolo Nesi

<http://www.dsi.unifi.it/~nesi>

nesi@dsi.unifi.it

2006



Rappresentazione dell'Informazione

In un calcolatore si possono rappresentare vari tipi di informazioni:

- Numeri reali
- Numeri interi
- Testi
- Grafici
-
- Disegni
- Fotografie
- Filmati
- Suoni
- ...



Rappresentazione dell'Informazione

L'informazione può essere rappresentata in due forme:

➤ **Analogica**

➤ la grandezza è rappresentata in modo continuo.

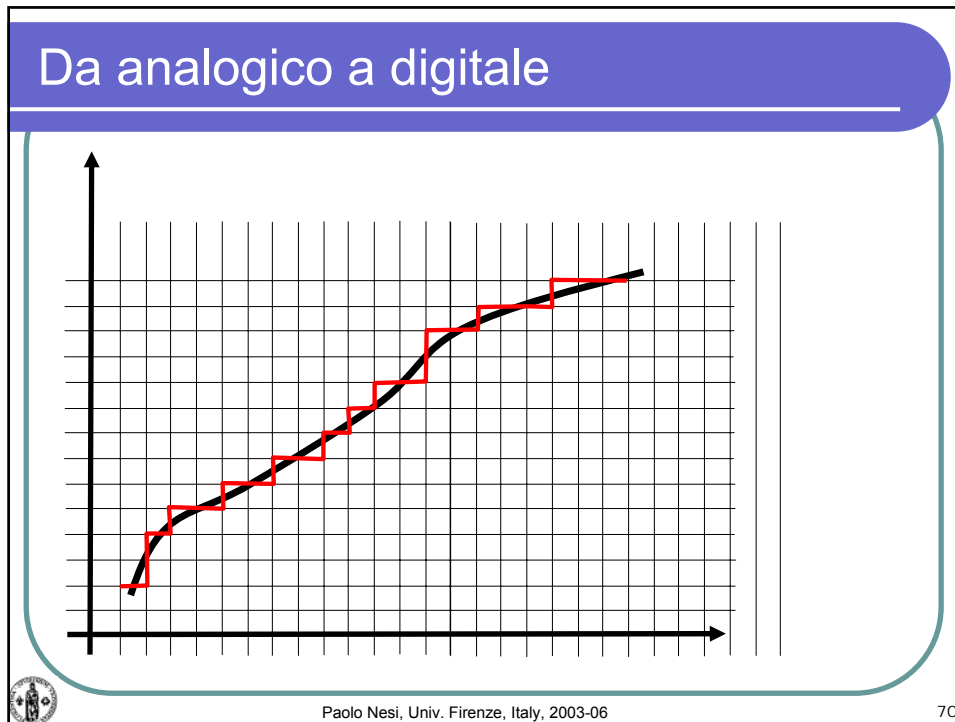
➤ **Digitale**

➤ una grandezza è rappresentata in modo discreto.

• Gran parte delle grandezze fisiche sono di tipo continuo (ad esempio un segnale acustico).

• Tuttavia alcuni tipi di informazioni "artificiali" sono di tipo discreto (ad esempio un testo scritto).





- ### Rappresentazione dell'Informazione
- Per elaborare delle grandezze di tipo continuo con un calcolatore, bisogna utilizzare/trasformarle in una rappresentazione digitale, discreta.
 - La rappresentazione digitale è una approssimazione della rappresentazione analogica.
 - L'errore di approssimazione dipende dalla precisione della rappresentazione digitale utilizzata per la codifica interna nel calcolatore.
- Paolo Nesi, Univ. Firenze, Italy, 2003-06
- 71

Rappresentazione dell'Informazione

- Internamente ad un elaboratore ogni informazione è rappresentata da una sequenza di bit (cifre binarie)
- Una sequenza di bit può rappresentare entità diverse. Il modello di rappresentazione da la chiave di lettura.
- Ad esempio la sequenza di cifre binarie 01000001 può rappresentare:
 - l'intero 65
 - il carattere A
 - il valore di un segnale musicale
 - il codice del colore di un punto sullo schermo
 - etc.



Codici

- Un codice è un sistema di simboli atto a rappresentare una informazione di qualsiasi genere (caratteri, numeri, etc.).
- Ogni simbolo è messo in corrispondenza biunivoca con una entità che si vuole rappresentare
- Un codice binario usa come simboli le cifre binarie "0" e "1"



Sistemi di Numerazione Posizionali

- di una base, b
- di un insieme ordinato di cifre $d = \{\dots\dots\dots\}$, distinte l'una dall'altra con dimensione pari a quella della base: $\dim(d) = b$, L'insieme contiene simboli che hanno una certa posizione nella base ordinata.
- di un codice di interpretazione cioè di un insieme di regole che permettono di determinare quale sia il numero rappresentato da un gruppo di cifre,
- di un insieme di regole e di algoritmi per definire le operazioni fondamentali.



Paolo Nesi, Univ. Firenze, Italy, 2003-06

74

Sistemi di Numerazione Posizionali

- Un numero N può essere rappresentato come una sequenza di cifre:

$$N = d_{n-1}d_{n-2}\dots\dots d_1d_0 \cdot d_{-1}d_{-2}\dots\dots d_{-m}$$

corrispondente a (forma polinomica):

$$N = \underbrace{d_{n-1} \cdot b^{n-1} + d_{n-2} \cdot b^{n-2} + \dots + d_0 \cdot b^0}_{\text{PARTE INTERA}} + \underbrace{d_{-1} \cdot b^{-1} + \dots + d_{-m} \cdot b^{-m}}_{\text{PARTE FRAZIONARIA}}$$

dove:

- d = cifra
- n = numero cifre parte intera
- b = base
- m = numero cifre parte decimale



Paolo Nesi, Univ. Firenze, Italy, 2003-06

75

Sistema di Numerazione decimale

Sistema Decimale

- Indicato con "10", "d", "D"
- $b=10$, $d=\{0,1,2,3,4,5,6,7,8,9\}$

$$N=d_{n-1} \cdot 10^{n-1} + d_{n-2} \cdot 10^{n-2} + \dots + d_0 + d_{-1} \cdot 10^{-1} + d_{-m} \cdot 10^{-m}$$

- Esempio:

$$123.45 = 1 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0 + 4 \cdot 10^{-1} + 5 \cdot 10^{-2}$$



Sistemi di Numerazione Binario

- Indicato con "b", "2"
- $b=2$, $d=\{0,1\}$

$$N=d_{n-1} \cdot 2^{n-1} + d_{n-2} \cdot 2^{n-2} + \dots + d_0 \cdot 2^0 + d_{-1} \cdot 2^{-1} + d_{-m} \cdot 2^{-m}$$

- Esempio:

$$101.01_2 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} = 5.25_{10}$$

- La cifra binaria è detta "bit" (binary digit – pezzo, pezzetto: cioè l'unità più piccola di informazione).
- E' il sistema usato nei calcolatori.



2 alla n	N	2 alla - n
1	0	1.0
2	1	0.5
4	2	0.25
8	3	0.125
16	4	0.0625
32	5	0.03125
64	6	0.015625
128	7	0.0078125
256	8	0.00390625
512	9	0.001953125
1024	10	0.0009765625
2048	11	0.00048828125
4096	12	0.000244140625
8192	13	0.0001220703125
16384	14	0.00006103515625
32768	15	0.000030517578125
65536	16	0.0000152587890625
131072	17	0.00000762939453125
262144	18	0.000003814697265625
524288	19	0.0000019073486328125
1048576	20	0.00000095367431640525

78

Sistema di Numerazione ottale

- Indicato con “o”, oppure con il numero “8”
- $b=8$, $d=\{0,1,2,3,4,5,6,7\}$

$$N = d_{n-1} \cdot 8^{n-1} + d_{n-2} \cdot 8^{n-2} + \dots + d_0 \cdot 8^0 + d_{-1} \cdot 8^{-1} + d_{-m} \cdot 8^{-m}$$

- Esempio:

$$127_8 = 1 \cdot 8^2 + 2 \cdot 8^1 + 7 \cdot 8^0 = 87_{10}$$



Sistema di Numerazione esadecimale

(Base 16, indicato anche con H, e, 16)

- $b=16$, $d=\{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F\}$

$$N=d_{n-1} \cdot 16^{n-1} + d_{n-2} \cdot 16^{n-2} + \dots + d_0 \cdot 16^0 + d_{-1} \cdot 16^{-1} + d_{-m} \cdot 16^{-m}$$

- Esempio:

$$A1_H = A1_{16} = 10 \cdot 16^1 + 1 \cdot 16^0 = 161_{10}$$

- Per avere 16 simboli distinti bisogna aggiungere:

$$A_H=10 \quad C_H=12 \quad E_H=14$$

$$B_H=11 \quad D_H=13 \quad F_H=15$$



Conversioni di Base

Conversione dalla base 10 ad una base qualsiasi: parte intera

$$N=d_0 \cdot b^0 + d_1 \cdot b^1 + d_2 \cdot b^2 + \dots + d_{n-1} \cdot b^{n-1}$$

$$N=d_0 + b(d_1 + b(d_2(\dots + d_{n-1})))$$

Quindi dividendo N per la base r si ottiene come quoziente $d_1 + b(d_2(\dots + d_{n-1}))$ e come (resto d_0 la cifra meno significativa).

Dividendo ancora il quoziente per la base si ottiene la cifra di peso 1 e così via fino ad avere un quoziente 0



Conversioni di Base

Esempio di conversione dalla base 10 alla base 2: parte intera

The diagram illustrates the conversion of the decimal number 115 to binary using the division-by-2 method. The process is as follows:

Step	Quotient	Remainder (d _i)
1	57	1 (d ₀)
2	28	1 (d ₁)
3	14	0 (d ₂)
4	7	0 (d ₃)
5	3	1 (d ₄)
6	1	1 (d ₅)
7	0	1 (d ₆)

Reading the remainders from bottom to top (d₆ to d₀), the binary representation is 1110011₂. The final result is shown as $115_{10} = 1110011_2$.

82

Conversioni di Base

Conversione dalla base 10 alla base 2: parte frazionaria

$$N = d_{-1} \cdot 2^{-1} + d_{-2} \cdot 2^{-2} + \dots + d_{-m} \cdot 2^{-m}$$

Se si moltiplica N per la base 2 si ottiene per la parte intera d₋₁.

Moltiplicando ancora per 2 la parte frazionaria si ottiene come parte intera d₋₂ etc.

Il risultato termina quando il risultato della moltiplicazione è esattamente 1 o quando si è raggiunta la precisione voluta (una rappresentazione può essere finita in una base e infinita in un'altra).


83

Conversioni di Base

Esempio di conversione dalla base 10 alla base 2: parte frazionaria

	0.625×2	
	<hr/>	
d_1 →	1.250×2	0.250×2
	<hr/>	
d_2 →	0.500×2	
	<hr/>	
d_3 →	1.000×2	0.000×2

Ci si ferma → $0.625_{10} = 0.101_2$



Paolo Nesi, Univ. Firenze, Italy, 2003-06

84

Conversioni di Base

Esempio di conversione dalla base 10 alla base 2: parte intera e frazionaria


Per convertire un numero che ha parte intera e parte frazionaria si effettuano le conversioni separatamente.

Esempio: sia dato il numero decimale 115.625

$$115_{10} = 1110011_2$$

$$0.625_{10} = 0.101_2$$

$$115.625_{10} = 1110011.101_2$$



Paolo Nesi, Univ. Firenze, Italy, 2003-06

85

Conversioni di Base

Conversione fra le basi 8 o 16 e la base 2:

Le notazioni in base 8 e 16 possono essere pensate come delle abbreviazioni della notazione in base 2.

Ogni cifra ottale corrisponde a 3 cifre binarie:

$$101\ 110\ 001_2 = 561_8$$

Ogni cifra esadecimale corrisponde a 4 cifre binarie:

$$1\ 0111\ 0001_2 = 171_{16} \quad (0001\ 0111\ 0001_2 = 171_{16})$$



Tabella Dec ed Esad

- Tabella da binario a decimale a esadecimale

Esadecimale	Binario	decimale
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15



Tabella Decimale a Ottale

- Tabella da binario a ottale a esadecimale

<u>Ottale</u>	<u>Binario</u>	<u>decimale</u>
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7



Rappresentazione di numeri naturali

- Quante cifre sono necessarie per rappresentare in numero X in base 2?
 - Se si usano k bit si hanno 2^k possibili configurazioni, da 0000...a 1111...
 - Si possono pertanto rappresentare i numeri naturali compresi fra 0 e (2^k-1)



Rappresentazione di numeri naturali

- Tutti i numeri x con $2^{k-1} - 1 < x = 2^k - 1$ richiedono k bit per essere rappresentati
- Anche $2^{k-1} < x + 1 = 2^k$
- Quindi per rappresentare il numero x occorrono k bit con

$$k = \lceil \log_2(x+1) \rceil$$

Cioè l'intero immediatamente superiore a $\log_2(x+1)$



Rappresentazione di numeri naturali

- Il risultato vale anche per una generica base b
 $k = \lceil \log_b(x+1) \rceil$
- Se B è il numero di cifre binarie che serve a rappresentare il numero x e D è il numero di cifre necessarie sempre per rappresentare x

$$B/D = \lceil \log_2(x+1) \rceil / \lceil \log_{10}(x+1) \rceil = \lceil \log_{10}(x+1) / \log_{10}(2) \rceil / \lceil \log_{10}(x+1) \rceil \sim 3.3$$

- Occorrono all'incirca 10 cifre binarie per rappresentare 3 cifre decimali.
- Si usano queste abbreviazioni:
 - Kilo (K) = $2^{10} = 1024 \sim 1000$
 - Mega (M) = $2^{20} = 1.048.576 \sim 1.000.000$
 - Giga (G) = $2^{30} = \dots \sim 1.000.000.000$
 - Tera (T) = $2^{40} = \dots \sim 1.000.000.000.000$



Dinamica

- Per numeri interi con K cifre binarie
- Dinamica pari a 2 alla $k - 1$, $2^{(k-1)}$
- Numero di valori rappresentati pari a 2 alla k



Numeri reali, Rappresentazione in virgola fissa

- Supponendo di utilizzare $n=5$ bit per la parte intera e $m=3$ bit per la parte frazionaria:

XXXXX.XXX

Per esempio il numero

11000.101

corrisponde a:

$$1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^{-1} + 1 \cdot 2^{-3} = 24.625$$



Numeri reali, Rappresentazione in virgola fissa

- La rappresentazione in virgola fissa limita fortemente l'intervallo numerico utilizzabile. In tale caso è possibile rappresentare numeri da $2^{-3}=0.125$ a $2^5=32$

- Il numero di cifre significative dipende dal suo valore assoluto:

$$10011.101 = 19.625 \text{ ha 8 cifre significative}$$

- Anche

$$1001.1101 = 9.81125 \text{ ha 8 cifre significative}$$

Ma nella rappresentazione in virgola fissa diventa (5:3):

$$01001.110 = 9.75$$

con notevole perdita di precisione.



Precisione, P

- $E_a = V_v - V_r$ Errore assoluto
- $E_r = E_a / V_v$ Errore relativo
- $E_r\% = E_r \cdot 100$ Errore relativo percentuale

- Per m cifre dopo la virgola
- 2 alla $-m$, 2^{-m} , per la parte frazionaria
- Confronto della precisione P con un esempio reale, verifica che la $P \geq E_a$



Operazioni sui numeri binari


- Come per i numeri decimali ma con le seguenti tabelle:

Somma:

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	10

Prodotto:

A	B	AB
0	0	0
0	1	0
1	0	0
1	1	1



Paolo Nesi, Univ. Firenze, Italy, 2003-06

97

Operazioni sui numeri binari


- **Somma:**

0+0=0
 0+1=1
 1+0=1
 1+1=0 con riporto 1 (2_{10})
 1+1+1=1 con riporto 1 (3_{10})

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	10

```

111      riporto
 10110+
  11101
  -----
 110011
        
```



Paolo Nesi, Univ. Firenze, Italy, 2003-06

98

Operazioni sui numeri binari

- **Sottrazione:**
 $0-0=0$
 $1-0=1$
 $1-1=0$
 $0-1=1$
 con un prestito dal bit più a sinistra.


A	B	A-B
0	0	0
0	1	1(1)
1	0	1
1	1	0

prestito

$$\begin{array}{r} 1110- \\ 0101 \\ \hline 1001 \end{array}$$

prestito

$$\begin{array}{r} 1100 \\ 1001 \\ \hline 0011 \end{array}$$



Paolo Nesi, Univ. Firenze, Italy, 2003-06


99

Operazioni sui numeri binari

- **Prodotto:**

A	B	AB
0	0	0
0	1	0
1	0	0
1	1	1

$$\begin{array}{r} 1011 \\ 1101 \\ \hline 1011+ \\ 0000- \\ \hline 01011+ \\ 1011-- \\ \hline 110111+ \\ 1011--- \\ \hline 10001111 \end{array}$$




Paolo Nesi, Univ. Firenze, Italy, 2003-06

100

Operazioni sui numeri binari

- **Divisione:**

101101	11
00	01111
<hr style="width: 100%;"/>	
101	
011	
<hr style="width: 100%;"/>	
101	
11	
<hr style="width: 100%;"/>	
100	
11	
<hr style="width: 100%;"/>	
011	
11	
<hr style="width: 100%;"/>	
00	



Paolo Nesi, Univ. Firenze, Italy, 2003-06

101


Operazioni di Shift (Scorrimento)

0	0	1	1	0	0	1	1
---	---	---	---	---	---	---	---

- **Shift a sinistra:**
moltiplica per 2

0	1	1	0	0	1	1	0
---	---	---	---	---	---	---	---
- **Shift a destra:**
divide per 2

0	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---



Paolo Nesi, Univ. Firenze, Italy, 2003-06


102

Complemento

- Complemento a b (base):
 - Dato un numero x con base b , e k cifre è definito come

$$C_b(x) = b^k - x$$
- Complemento a $b-1$:

$$C_{b-1}(x) = b^k - 1 - x = C_b(x) - 1$$




Paolo Nesi, Univ. Firenze, Italy, 2003-06

103

Complemento, proprietà

- $C_b(x) = C_{b-1}(x) + 1$
- $C_b(C_b(x)) = x$
- $C_{b-1}(C_{b-1}(x)) = x$

$C_2(x) = C_1(x) + 1$



Paolo Nesi, Univ. Firenze, Italy, 2003-06


104

Complemento

- Complemento a 2: $2^k - x$
 - Es: $x = 01010110$ $k=8$

$$\begin{array}{r} 100000000 \\ - \\ \underline{01010110} \\ 010101010 \end{array}$$
- Complemento a 1: $2^k - 1 - x$
 - Es: $x = 01010110$ $k=8$

$$\begin{array}{r} 11111111 \\ - \\ \underline{01010110} \\ 10101001 \end{array}$$



Paolo Nesi, Univ. Firenze, Italy, 2003-06


105

Complemento

- Complemento a 1: $2^k - 1 - x$
 - Lo si ottiene semplicemente scambiando 0 con 1 e 1 con 0
 - Es: $x = 01010110$ $k=8$

$$\begin{array}{r} 01010110 \quad x \\ 10101001 \quad \text{complemento di } x \text{ (= not } x \text{)} \end{array}$$
- Complemento a 2: $2^k - x$
 - Si fa il complemento ad 1 di x e si somma 1 (complemento veloce)
 - Es: $x = 01010110$ $k=8$

$$\begin{array}{r} 01010110 \quad x \\ 10101001 \quad + \quad \text{complemento a 1} \\ \underline{\quad\quad\quad 1} \quad = \quad \text{sommo 1} \\ 10101010 \quad \text{complemento a 2} \end{array}$$



Paolo Nesi, Univ. Firenze, Italy, 2003-06

106

Calcolatori Elettronici

CDL in Ingegneria Elettronica

Facoltà di Ingegneria,
Università degli Studi di Firenze

Nuovo Ordinamento
Parte 1c, Rappresentazioni

Prof. Paolo Nesi
<http://www.dsi.unifi.it/~nesi>
nesi@dsi.unifi.it
2006




Paolo Nesi, Univ. Firenze, Italy, 2003-06 107

Rappresentazione dell'Informazione

In un calcolatore si possono rappresentare vari tipi di informazioni:

- Numeri naturali (visto)
- Numeri reali in virgola fissa (visto)
- Numeri interi (da vedere in questa sessione)
 - In varie forme
- Rappresentazione ASCII
- ...
- ...



Paolo Nesi, Univ. Firenze, Italy, 2003-06 108

Rappresentazione dei numeri nei calcolatori

- La memoria è organizzata in celle (“parole”) con un numero fisso di bit (ad esempio 8 = 1 byte).

0	1	0	1	1	0	0	1
---	---	---	---	---	---	---	---

- Corrisponde a $1 \cdot 64 + 1 \cdot 16 + 1 \cdot 8 + 1$
- Usando 8 cifre bit si possono rappresentare numeri compresi fra 0 e $2^8 - 1 = 255$



Rappresentazione dei numeri nei calcolatori

L'uso un numero finito di cifre porta ad una aritmetica modulare, che ritorna in forma chiusa oltre la dinamica al numero piu' basso.

Incrementi unitari.

00000000 (0)
 00000001 (1)
 00000010 (2)

 11111111 (255)
 (1)00000000 (256 -> 0)
 00000001 (1)

• Quando si raggiunge il numero 255 (tutti 1) non potendo rappresentare il 256 che richiederebbe 9 bit si torna allo 0.

• Si ha quindi una aritmetica modulo 256.

• In generale se si usano parole di k bit si ha una aritmetica modulo 2^k .



Rappresentazione degli interi

- Disponendo di k bit si possono avere 2^k configurazioni diverse.
- Metà possono essere usate per i numeri positivi, l'altra metà per quelli negativi.
- Ci sono due possibili rappresentazioni:
 - in modulo e segno
 - in complemento a 2



Paolo Nesi, Univ. Firenze, Italy, 2003-06

111

Rappresentazione degli interi in modulo e segno

- Il bit più significativo viene usato per rappresentare il segno:
 - 0 per i numeri positivi
 - 1 per quelli negativi
- Le cifre restanti rappresentano il modulo
 $+5_{10} \rightarrow 00000101_2$
 $-10_{10} \rightarrow 10001010_2$

↑ modulo segno
- Lo zero ha due rappresentazioni:
(+0) 00000000 e (-0) 10000000




Paolo Nesi, Univ. Firenze, Italy, 2003-06

112

Rappresentazione degli interi in modulo e segno

01111111	+127	● Se si aggiunge 1 a 127 si ha -0
01111110	+126	
00000010	+2	● Se si "aggiunge" 1 a -127 si ha 0 con un traboccamento
00000001	+1	
00000000	+0	
10000000	-0	
10000001	-1	
10000010	-2	
11111110	-126	
11111111	-127	




Paolo Nesi, Univ. Firenze, Italy, 2003-06

113

Rappresentazione degli interi relativi in modulo e segno

- Algoritmo di **somma**:
 - Confrontare i bit di segno dei due numeri
 - Se sono uguali:
 - Somma i moduli
 - Assegna come bit di segno del risultato il bit di segno degli *operandi*
 - Altrimenti:
 - Confronta i valori assoluti dei due numeri
 - Assegna come bit di segno del risultato quello dell'operando con modulo maggiore
 - Operare la sottrazione fra i moduli nell'ordine giusto
- Macchinoso!! (Analogo la sottrazione).



Paolo Nesi, Univ. Firenze, Italy, 2003-06

114

Rappresentazione dei numeri interi in complemento a 2

- Se hanno numeri di k bit
- I numeri positivi sono rappresentati dal loro modulo e hanno il bit più significativo (segno)= 0
- I numeri negativi sono rappresentati facendo il complemento a due del corrispondente numero positivo. Hanno il bit più significativo (segno) = 1
- Si rappresentano come positivi i numeri da 0 a $2^{k-1}-1$
- Si rappresentano come negativi i numero che vanno da -2^{k-1} a -1



Rappresentazione dei numeri interi in complemento a 2

01111111	+127
01111110	+126

00000010	+2
00000001	+1
00000000	0
11111111	-1
11111110	-2

10000001	-127
10000000	-128

Supponendo di avere una cella di 8 bit
($k=8$, $2^k=256$)

$$2^{k-1}-1 = 127$$

$$-2^{k-1} = -128$$

Rappresentazione ciclica:

127+1 sconfinava a -128



Rappresentazione dei numeri in complemento a 2

- Se si desidera calcolare $A-B$ con numeri di k bit.
- È possibile farlo utilizzando il complemento a 2 di B come numero $-B$, Infatti:

$$A - B = A - B + 2^k - 2^k = A + 2^k - B - 2^k$$
 poiché $C_2(B) = 2^k - B$ si ha:

$$A - B = A + C_2(B) - 2^k$$
- Per fare una sottrazione si somma il complemento a 2 del secondo operando
- 2^k è un 1 seguito da k zeri, non rappresentabile con k bit e permette di trascurare i traboccamenti.



Rappresentazione dei numeri in complemento a 2

- interpretare un numero in complemento a due?
- Se il bit più significativo (segno) =0 allora il numero è positivo e le sue cifre ci danno il modulo
 - Es: 00110011 = + 51 (32+16+2+1)
- Se il bit più significativo (segno) =1 allora il numero è negativo e per avere il modulo:

Es: 10110011 negativo

- 01001100+ complemento a 1
- 1
- 01001101 complemento a 2 = 77 (64+8+4+1)
- il numero originale era -77




Rappresentazione dei numeri in complemento a 2

- Esempi (se $k=5$, è possibile rappresentare i numeri da -16 a + 15):

01001+	+9	
<u>00100=</u>	<u>+4</u>	
01101	+13	

→

01001+	+9	(si noti che si somma per sottrarre)
<u>11100=</u>	<u>-4</u>	
<u>1</u> 00101	+5	(traboccamento sul sesto bit eliminato grazie al 2^k che in questo caso è 100000)



Paolo Nesi, Univ. Firenze, Italy, 2003-06


119

Rappresentazione dei numeri in complemento a 2

- Esempi (se $k=5$, numeri da -16 a + 15):

10111+	-9	
<u>00100=</u>	<u>+4</u>	
11011	-5	

10111+	-9	
<u>11100=</u>	<u>-4</u>	
<u>1</u> 10011	-13	(sesto bit eliminato)



Paolo Nesi, Univ. Firenze, Italy, 2003-06

120

Rappresentazione dei numeri in complemento a 2

- Esempi (k=5, numeri da -16 a + 15):

$\begin{array}{r} \textcircled{1} \\ 01001+ \\ 01000= \\ \hline 10001 \end{array}$	$\begin{array}{r} +9 \\ +8 \\ \hline +17 \end{array}$	<p>OVERFLOW + 17 non è rappresentabile con 5 bit (riporto sul bit di segno)</p>
$\begin{array}{r} 10111+ \\ 10111= \\ \hline 101110 \end{array}$	$\begin{array}{r} -9 \\ -9 \\ \hline -18 \end{array}$	<p>(sesto bit eliminato) OVERFLOW -18 non è rappresentabile con 5 bit (riporto fuori dal bit di segno)</p>

Paolo Nesi, Univ. Firenze, Italy, 2003-06 121

Rappresentazione dei numeri in complemento a 2

OVERFLOW (Superamento della dinamica)

- Si ha quando il risultato non è rappresentabile con il numero di bit disponibili.
- Si ha solo sommando due numeri entrambi positivi o entrambi negativi.
- In questo caso il segno del risultato risulta opposto a quello degli operandi.




Paolo Nesi, Univ. Firenze, Italy, 2003-06 122

Esercizi

- 1) Calcolare $A+B$ e $A \text{ AND } B$ con $A=(10111011)_2$ e $B=(00110110)_2$
- 2) Calcolare $A-B$ e $A \text{ OR } B$ con $A=(10000001)_2$ e $B=(00000101)_2$
- 3) Calcolare $A \times B$ con $A=(10001010)_2$ e $B=(1001)_2$
- 4) Calcolare A/B con $A=(1100)_2$ e $B=(101)_2$
- 5) Si converta da decimale ad ottale il numero 426
- 6) A quale numero ottale corrisponde $(101110110)_2$?
- 7) Si convertano i seguenti numeri esadecimali in binari e decimali:
(a) $(AF5)_{16}$ (b) $(FE8)_{16}$
- 8) Convertire in binario il numero -1238 utilizzando 16 bit per la rappresentazione.



Codici

- Codice: sistema di simboli che permette la rappresentazione dell'informazione
- Esempi:   
- Decodifica agevole vs codici compressi



Definizioni

- **SIMBOLO**: entità di cui non si da qui una definizione formale
- **STRINGA**: sequenza finita di simboli giustapposti (lunghezza della stringa, stringa vuota)
- **ALFABETO**: insieme finito di simboli
- **LINGUAGGIO**: insieme di stringhe di simboli tratti da un alfabeto



Esempi di alfabeti

- Alfabeto italiano:
 $\{A, B, C, D, \dots Z\}$
- Alfabeto greco:
 $\{\alpha, \beta, \gamma, \delta, \dots \omega\}$
- Alfabeto binario:
 $\{0, 1\}$



Alfabeto usato dal calcolatore

- Interruttore (aperto/chiuso)
- Foro su scheda (aperto/chiuso)
- Transistor (in conduzione/spento)
- Tensione (alta/bassa)
- Dominio di magnetizzazione (\uparrow/\downarrow)
-



Alfabeto usato dal calcolatore

- Gli elaboratori utilizzano una logica e un'aritmetica **binaria**
- Ai due stati di un dispositivo vengono associati i due simboli **0** e **1**



Codice ASCII

- American Standard Code for Information Interchange
- 7 bit quindi 128 simboli diversi
- ASCII esteso (8bit)
 - diverse estensioni in dipendenza dal paese
 - oppure aggiunge la parità



Paolo Nesi, Univ. Firenze, Italy, 2003-06

129

Tabella Ascii

ASCII Codes

Chr	Dec	Hex	Char	Code	Dec	Hex	Char	Code	Dec	Hex	Char	Code	Dec	Hex	Char	Code	Dec	Hex	Char	Code											
@	0	00	NUL		32	20	!		64	40	Q		96	60	s		128	80	Ç		160	A0	à		192	C0	Ì		224	F0	œ
A	1	01	SOH		33	21	"		65	41	R		97	61	a		129	81	ü		161	A1	á		193	C1	Í		225	F1	ø
B	2	02	STX		34	22	#		66	42	A	B	98	62	b		130	82	ÿ		162	A2	â		194	C2	Î		226	F2	¹
C	3	03	ETX		35	23	\$		67	43	C		99	63	c		131	83	ÿ		163	A3	ã		195	C3	Ï		227	F3	º
D	4	04	ENQ		36	24	%		68	44	D		100	64	d		132	84	ÿ		164	A4	ä		196	C4	Ï		228	F4	»
E	5	05	ENQ		37	25	Z		69	45	E		101	65	e		133	85	ÿ		165	A5	å		197	C5	Ï		229	F5	¼
F	6	06	ACK		38	26	[70	46	F		102	66	f		134	86	ÿ		166	A6	æ		198	C6	Ï		230	F6	½
G	7	07	BEL		39	27	\		71	47	G		103	67	g		135	87	ÿ		167	A7	ç		199	C7	Ï		232	F7	¾
H	8	08	BS		40	28	(72	48	H		104	68	h		136	88	ÿ		168	A8	è		200	C8	Ï		232	F8	¸
I	9	09	HT		41	29)		73	49	I		105	69	i		137	89	ÿ		169	A9	é		201	C9	Ï		233	F9	¹
J	10	0A	LF		42	2A	*		74	4A	J		106	6A	j		138	8A	ÿ		170	AA	ê		202	CA	Ï		234	FA	º
K	11	0B	VT		43	2B	+		75	4B	K		107	6B	k		139	8B	ÿ		171	AB	ë		203	CB	Ï		235	FB	»
L	12	0C	FF		44	2C	,		76	4C	L		108	6C	l		140	8C	ÿ		172	AC	ì		204	CC	Ï		236	FB	¼
M	13	0D	CR		45	2D	-		77	4D	M		109	6D	m		141	8D	ÿ		173	AD	í		205	CD	Ï		237	FD	½
N	14	0E	SO		46	2E	.		78	4E	N		110	6E	n		142	8E	ÿ		174	AE	î		206	CE	Ï		238	FE	¾
O	15	0F	SI		47	2F	/		79	4F	O		111	6F	o		143	8F	ÿ		175	AF	ï		207	CF	Ï		239	FF	¸
P	16	10	DLE		48	30	0		80	50	P		112	70	p		144	90	ÿ		176	BO	ÿ		208	DO	Ï		240	FO	¹
Q	17	11	DC1		49	31	1		81	51	Q		113	71	q		145	91	ÿ		177	B1	ÿ		209	D1	Ï		241	F1	º
R	18	12	DC2		50	32	2		82	52	R		114	72	r		146	92	ÿ		178	B2	ÿ		210	D2	Ï		242	F2	»
S	19	13	DC3		51	33	3		83	53	S		115	73	s		147	93	ÿ		179	B3	ÿ		211	D3	Ï		243	F3	¼
T	20	14	DC4		52	34	4		84	54	T		116	74	t		148	94	ÿ		180	B4	ÿ		212	D4	Ï		244	F4	½
U	21	15	NAK		53	35	5		85	55	U		117	75	u		149	95	ÿ		181	B5	ÿ		213	D5	Ï		245	F5	¾
V	22	16	SYN		54	36	6		86	56	V		118	76	v		150	96	ÿ		182	B6	ÿ		214	D6	Ï		246	F6	¸
W	23	17	ETB		55	37	7		87	57	W		119	77	w		151	97	ÿ		183	B7	ÿ		215	D7	Ï		247	F7	¹
X	24	18	CAN		56	38	8		88	58	X		120	78	x		152	98	ÿ		184	B8	ÿ		216	D8	Ï		248	F8	º
Y	25	19	EM		57	39	9		89	59	Y		121	79	y		153	99	ÿ		185	B9	ÿ		217	D9	Ï		249	F9	»
Z	26	1A	SUB		58	3A	:		90	5A	Z		122	7A	z		154	9A	ÿ		186	BA	ÿ		218	DA	Ï		250	FA	¼
[27	1B	ESC		59	3B	;		91	5B	[123	7B	{		155	9B	ÿ		187	BB	ÿ		219	DB	Ï		251	FB	½
\	28	1C	FS		60	3C	<		92	5C	\		124	7C			156	9C	ÿ		188	BC	ÿ		220	DC	Ï		252	FB	¾
]	29	1D	GS		61	3D	=		93	5D]		125	7D	}		157	9D	ÿ		189	BD	ÿ		221	DD	Ï		253	FD	¸
^	30	1E	RS		62	3E	>		94	5E	^		126	7E	~		158	9E	ÿ		190	BE	ÿ		222	DE	Ï		254	FE	¹
_	31	1F	US		63	3F	?		95	5F	_		127	7F	ÿ		159	9F	ÿ		191	BF	ÿ		223	DF	Ï		255	FF	º



130

Tabella Ascii

Ctrl	Dec	Hex	Char	Code	Dec	Hex	Char	Dec	Hex	Char
	0	00		NUL	32	20		64	40	@
~	1	01	A	SOH	33	21	!	65	41	A
~	2	02	B	STX	34	22	"	66	42	B
~	3	03	C	ETX	35	23	#	67	43	C
~	4	04	D	EOT	36	24	\$	68	44	D
~	5	05	E	ENQ	37	25	%	69	45	E
~	6	06	F	ACK	38	26	&	70	46	F
~	7	07	G	BEL	39	27	'	71	47	G
~	8	08	H	BS	40	28	(72	48	H
~	9	09	I	HT	41	29)	73	49	I
~	10	0A	J	LF	42	2A	*	74	4A	J
~	11	0B	K	VT	43	2B	+	75	4B	K
~	12	0C	L	FF	44	2C	,	76	4C	L
~	13	0D	M	CR	45	2D	-	77	4D	M
~	14	0E	N	SO	46	2E	.	78	4E	N
~	15	0F	O	SI	47	2F	/	79	4F	O

Paolo Nesi, Univ. Firenze, Italy, 2003-06

131

Codice ASCII

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	##32;	Space	64	40	100	##64;	@
1	1	001	SOH (start of heading)	33	21	041	##33;	!	65	41	101	##65;	A
2	2	002	STX (start of text)	34	22	042	##34;	"	66	42	102	##66;	B
3	3	003	ETX (end of text)	35	23	043	##35;	#	67	43	103	##67;	C
4	4	004	EOT (end of transmission)	36	24	044	##36;	\$	68	44	104	##68;	D
5	5	005	ENQ (enquiry)	37	25	045	##37;	%	69	45	105	##69;	E
6	6	006	ACK (acknowledge)	38	26	046	##38;	&	70	46	106	##70;	F
7	7	007	BEL (bell)	39	27	047	##39;	'	71	47	107	##71;	G
8	8	010	BS (backspace)	40	28	050	##40;	(72	48	110	##72;	H
9	9	011	TAB (horizontal tab)	41	29	051	##41;)	73	49	111	##73;	I
10	A	012	LF (NL line feed, new line)	42	2A	052	##42;	*	74	4A	112	##74;	J
11	B	013	VT (vertical tab)	43	2B	053	##43;	+	75	4B	113	##75;	K
12	C	014	FF (NP form feed, new page)	44	2C	054	##44;	,	76	4C	114	##76;	L
13	D	015	CR (carriage return)	45	2D	055	##45;	-	77	4D	115	##77;	M
14	E	016	SO (shift out)	46	2E	056	##46;	.	78	4E	116	##78;	N
15	F	017	SI (shift in)	47	2F	057	##47;	/	79	4F	117	##79;	O
16	10	020	DLE (data link escape)	48	30	060	##48;	0	80	50	120	##80;	P
17	11	021	DC1 (device control 1)	49	31	061	##49;	1	81	51	121	##81;	Q
18	12	022	DC2 (device control 2)	50	32	062	##50;	2	82	52	122	##82;	R
19	13	023	DC3 (device control 3)	51	33	063	##51;	3	83	53	123	##83;	S
20	14	024	DC4 (device control 4)	52	34	064	##52;	4	84	54	124	##84;	T
21	15	025	NAK (negative acknowledge)	53	35	065	##53;	5	85	55	125	##85;	U
22	16	026	SYN (synchronous idle)	54	36	066	##54;	6	86	56	126	##86;	V
23	17	027	ETB (end of trans. block)	55	37	067	##55;	7	87	57	127	##87;	W
24	18	030	CAN (cancel)	56	38	070	##56;	8	88	58	130	##88;	X
25	19	031	EM (end of medium)	57	39	071	##57;	9	89	59	131	##89;	Y
26	1A	032	SUB (substitute)	58	3A	072	##58;	:	90	5A	132	##90;	Z
27	1B	033	ESC (escape)	59	3B	073	##59;	;	91	5B	133	##91;	[
28	1C	034	FS (file separator)	60	3C	074	##60;	<	92	5C	134	##92;	\
29	1D	035	GS (group separator)	61	3D	075	##61;	=	93	5D	135	##93;]
30	1E	036	RS (record separator)	62	3E	076	##62;	>	94	5E	136	##94;	^
31	1F	037	US (unit separator)	63	3F	077	##63;	?	95	5F	137	##95;	_

Source: www.asciitable.com

Codice ASCII - Esteso

128	Ç	144	É	160	á	176	☐	193	⊥	209	ƒ	225	ß	241	±
129	ù	145	æ	161	í	177	☐	194	⊥	210	π	226	Γ	242	≥
130	é	146	Æ	162	ó	178	☐	195	⊥	211	⊥	227	π	243	≤
131	â	147	ô	163	ú	179		196	-	212	⊥	228	Σ	244	ƒ
132	ä	148	ö	164	ÿ	180	†	197	+	213	ƒ	229	σ	245	√
133	à	149	ò	165	Ñ	181	†	198	†	214	ƒ	230	μ	246	+
134	â	150	û	166	ª	182	‡	199	‡	215	‡	231	τ	247	≈
135	ç	151	ù	167	º	183	π	200	⊥	216	+	232	Φ	248	°
136	ê	152	-	168	¸	184	¶	201	ƒ	217	√	233	⊙	249	.
137	ë	153	Ö	169	-	185	‡	202	⊥	218	ƒ	234	Ω	250	.
138	è	154	Û	170	¬	186	‡	203	ƒ	219	■	235	δ	251	√
139	ï	156	£	171	½	187	¶	204	‡	220	■	236	∞	252	-
140	î	157	¥	172	¾	188	¶	205	=	221	■	237	φ	253	z
141	í	158	-	173	ı	189	¶	206	‡	222	■	238	ε	254	■
142	Ï	159	ƒ	174	«	190	‡	207	⊥	223	■	239	∩	255	
143	Ä	192	Ł	175	»	191	¶	208	⊥	224	α	240	≡		

Source: www.asciitable.com

Paolo Nesi, Univ. Firenze, Italy, 2003-06

133

Conversioni con Esadecimali

Hexadecimal-Binary-Decimal Conversion

Hex Number	Binary Number	Decimal Digit 000X	Decimal Digit 00X0	Decimal Digit 0X00	Decimal Digit X000
0	0000	0	0	0	0
1	0001	1	16	256	4,096
2	0010	2	32	512	8,192
3	0011	3	48	768	12,288
4	0100	4	64	1,024	16,384
5	0101	5	80	1,280	20,480
6	0110	6	96	1,536	24,576
7	0111	7	112	1,792	28,672
8	1000	8	128	2,048	32,768
9	1001	9	144	2,304	36,864
A	1010	10	160	2,560	40,960
B	1011	11	176	2,816	45,056
C	1100	12	192	3,072	49,152
D	1101	13	208	3,328	53,248
E	1110	14	224	3,584	57,344
F	1111	15	240	3,840	61,440

134

Unicode Character Code

Unicode is a 16-bit code.

0000	NUL	0020	SP	0040	@	0060	`	0080	Ctrl	00A0	NBS	00C0	À	00E0	à
0001	SOH	0021	!	0041	A	0061	a	0081	Ctrl	00A1	¡	00C1	Á	00E1	á
0002	STX	0022	"	0042	B	0062	b	0082	Ctrl	00A2	¢	00C2	Â	00E2	â
0003	ETX	0023	#	0043	C	0063	c	0083	Ctrl	00A3	£	00C3	Ã	00E3	ã
0004	EOT	0024	\$	0044	D	0064	d	0084	Ctrl	00A4	¤	00C4	Ä	00E4	ä
0005	ENQ	0025	%	0045	E	0065	e	0085	Ctrl	00A5	¥	00C5	Å	00E5	å
0006	ACK	0026	&	0046	F	0066	f	0086	Ctrl	00A6	¦	00C6	Æ	00E6	æ
0007	BEL	0027	'	0047	G	0067	g	0087	Ctrl	00A7	§	00C7	Ç	00E7	ç
0008	BS	0028	(0048	H	0068	h	0088	Ctrl	00A8	¨	00C8	È	00E8	è
0009	HT	0029)	0049	I	0069	i	0089	Ctrl	00A9	©	00C9	É	00E9	é
000A	LF	002A	*	004A	J	006A	j	008A	Ctrl	00AA	ª	00CA	Ê	00EA	ê
000B	VT	002B	+	004B	K	006B	k	008B	Ctrl	00AB	«	00CB	Ë	00EB	ë
000C	FF	002C	,	004C	L	006C	l	008C	Ctrl	00AC	¬	00CC	Ì	00EC	ì
000D	CR	002D	-	004D	M	006D	m	008D	Ctrl	00AD	–	00CD	Í	00ED	í
000E	SO	002E	.	004E	N	006E	n	008E	Ctrl	00AE	®	00CE	Î	00EE	î
000F	SI	002F	/	004F	O	006F	o	008F	Ctrl	00AF	¯	00CF	Ï	00EF	ï
0010	DLE	0030	0	0050	P	0070	p	0090	Ctrl	00B0	°	00D0	Ð	00F0	ð
0011	DC1	0031	1	0051	Q	0071	q	0091	Ctrl	00B1	±	00D1	Ñ	00F1	ñ
0012	DC2	0032	2	0052	R	0072	r	0092	Ctrl	00B2	²	00D2	Ò	00F2	ò
0013	DC3	0033	3	0053	S	0073	s	0093	Ctrl	00B3	³	00D3	Ó	00F3	ó
0014	DC4	0034	4	0054	T	0074	t	0094	Ctrl	00B4	´	00D4	Ô	00F4	ô
0015	NAK	0035	5	0055	U	0075	u	0095	Ctrl	00B5	µ	00D5	Õ	00F5	õ
0016	SYN	0036	6	0056	V	0076	v	0096	Ctrl	00B6	¶	00D6	Ö	00F6	ö
0017	ETB	0037	7	0057	W	0077	w	0097	Ctrl	00B7	·	00D7	×	00F7	÷
0018	CAN	0038	8	0058	X	0078	x	0098	Ctrl	00B8	¸	00D8	Ø	00F8	ø
0019	EM	0039	9	0059	Y	0079	y	0099	Ctrl	00B9	¹	00D9	Ù	00F9	ù
001A	SUB	003A	:	005A	Z	007A	z	009A	Ctrl	00BA	º	00DA	Ú	00FA	ú
001B	ESC	003B	;	005B	[007B	{	009B	Ctrl	00BB	»	00DB	Û	00FB	û
001C	FS	003C	<	005C	\	007C		009C	Ctrl	00BC	¼	00DC	Ü	00FC	ü
001D	GS	003D	=	005D]	007D	}	009D	Ctrl	00BD	½	00DD	Ý	00FD	ý
001E	RS	003E	>	005E	^	007E	~	009E	Ctrl	00BE	¾	00DE	ÿ	00FE	ÿ
001F	US	003F	?	005F	_	007F	DEL	009F	Ctrl	00BF	¿	00DF	Ş	00FF	ş

NUL	Null	SOH	Start of heading	CAN	Cancel	SP	Space
STX	Start of text	EOT	End of transmission	EM	End of medium	DEL	Delete
ETX	End of text	DC1	Device control 1	SUB	Substitute	Ctrl	Control
ENQ	Enquiry	DC2	Device control 2	ESC	Escape	FF	Form feed
ACK	Acknowledge	DC3	Device control 3	FS	File separator	CR	Carriage return
BEL	Bell	DC4	Device control 4	GS	Group separator	SO	Shift out
BS	Backspace	NAK	Negative acknowledge	RS	Record separator	SI	Shift in
HT	Horizontal tab	NBS	Non-breaking space	US	Unit separator	DLE	Data link escape
LF	Line feed	ETB	End of transmission block	SYN	Synchronous idle	VT	Vertical tab

Prefissi Scientifici

- For computer memory, $1K = 2^{10} = 1024$. For everything else, like clock speeds, $1K = 1000$, and likewise for $1M$, $1G$, etc.

Prefix	Abbrev.	Quantity	Prefix	Abbrev.	Quantity
milli	m	10^{-3}	Kilo	K	10^3
micro	µ	10^{-6}	Mega	M	10^6
nano	n	10^{-9}	Giga	G	10^9
pico	p	10^{-12}	Tera	T	10^{12}
femto	f	10^{-15}	Peta	P	10^{15}
atto	a	10^{-18}	Exa	E	10^{18}



Calcolatori Elettronici

CDL in Ingegneria Elettronica

Facoltà di Ingegneria,
Università degli Studi di Firenze

Nuovo Ordinamento
Parte 2, Logica Combinatoria


Prof. Paolo Nesi
<http://www.dsi.unifi.it/~nesi>
nesi@dsi.unifi.it
2003-2006



Paolo Nesi, Univ. Firenze, Italy, 2003-06 138

Logica Combinatoria

- Un circuito logico combinatorio produce il valore delle sue uscite solo in base al valore degli ingressi in quel “momento”
- La stima dell’uscita non tiene conto dell’evoluzione temporale degli ingressi
- “Momento” significa che in circuiti reali si deve dare il tempo al circuito di raggiungere un valore stabile



Paolo Nesi, Univ. Firenze, Italy, 2003-06 139

Logica Combinatoria

- In seguito vedremo anche la logica sequenziale

Paolo Nesi, Univ. Firenze, Italy, 2003-06

140

Altre Porte oltre a: and, or, not

A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

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

NAND

A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

$F = \overline{A + B}$

NOR

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

$F = A \oplus B$

Exclusive-OR (XOR)

A	B	F
0	0	1
0	1	0
1	0	0
1	1	1

$F = A \odot B$

Exclusive-NOR (XNOR)

Paolo Nesi, Univ. Firenze, Italy, 2003-06

141

Teorema di De Morgan

A	B	$\overline{A B} = \overline{A} + \overline{B}$		$\overline{A + B} = \overline{A} \overline{B}$	
0	0	1	1	1	1
0	1	1	1	0	0
1	0	1	1	0	0
1	1	0	0	0	0

$$A + B = \overline{\overline{A + B}} = \overline{\overline{A} \overline{B}}$$

Paolo Nesi, Univ. Firenze, Italy, 2003-06

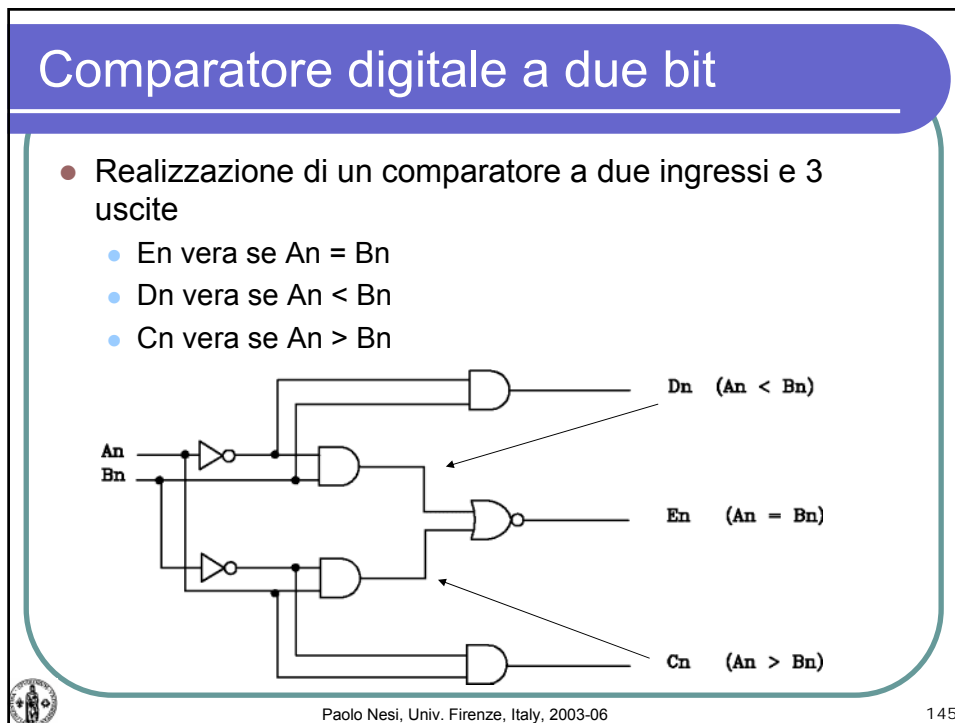
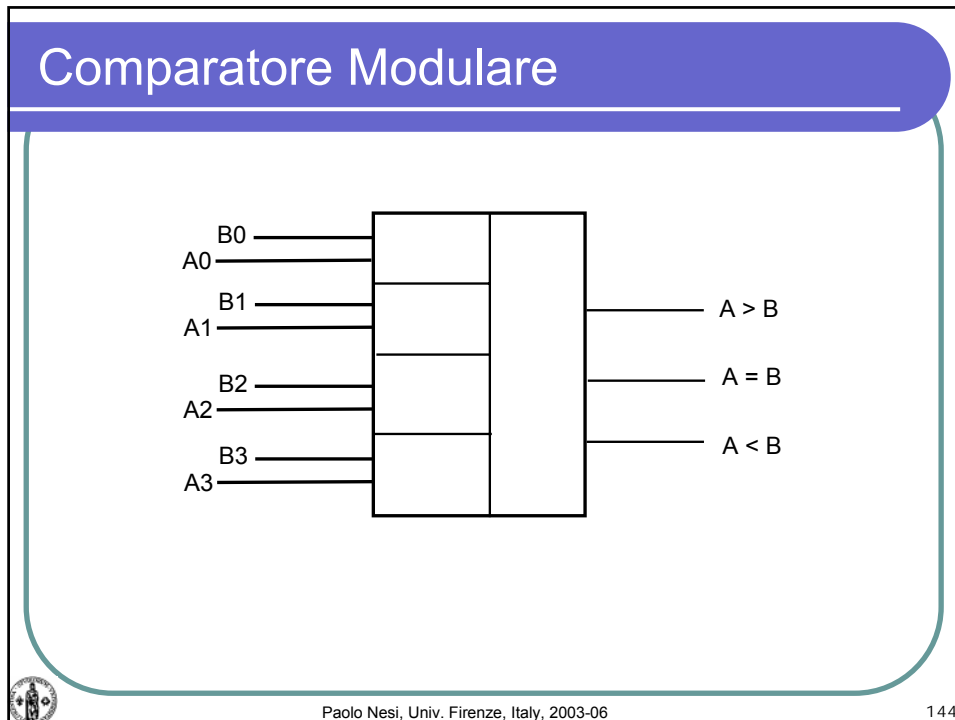
142

Comparatore a 4 bit

- A3-A2-A1-A0 che si confronta con B3-B2-B1-B0
- A=B
 - A3=B3, A2=B2, A1=B1, A0=B0
 - E=E3 E2 E1 E0
- A>B
 - A3>B3 => A3 not B3
 - A3=B3, A2>B2 => E3 A2 not B2
 - A3=B3, A2=B2, A1>B1 => E3 E2 A1 not B1
 - A3=B3, A2=B3, A1=B1, A0>B0 => E3 E2 E1 A0 not B0
 - G = C1 or C2 or C3 or C0
- A<B
 -
 -
 - ...
- Molto complessa come soluzione, e' meglio studiare una soluzione modulare....

Paolo Nesi, Univ. Firenze, Italy, 2003-06

143



Comparatore Digitale

- $D_n = \text{not } A B$ vero se $A_n < B_n$
- $C_n = \text{not } B A$ vero se $B_n > A_n$
- $E_n = A B + \text{not } A \text{ not } B = \text{not } (A \text{ xor } B)$

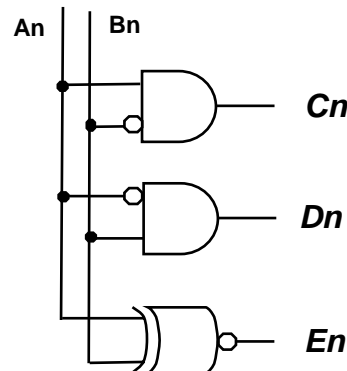
A_n	B_n	< D_n	E_n	> C_n
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0



Paolo Nesi, Univ. Firenze, Italy, 2003-06

146

Singolo Modulo Comparatore



- $D_n = \text{not } A B$
- $C_n = \text{not } B A$
- $E_n = A B + \text{not } A \text{ not } B = \text{not } (A \text{ xor } B)$



Paolo Nesi, Univ. Firenze, Italy, 2003-06

147

Se fosse a 2 bit !

>	>	<	<	>	<
C1	C0	D1	D0	C	D
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	0	0
0	1	1	0	0	1
0	1	1	1	0	1
1	0	0	0	1	0
1	0	0	1	1	0
1	0	1	0	0	0
1	0	1	1	0	1
1	1	0	0	1	0
1	1	0	1	1	0
1	1	1	0	1	0
1	1	1	1	0	0

- $C = \text{not } C1 \text{ C0 not } D1 \text{ not } D0 +$
 $C1 \text{ not } C0 \text{ not } D1 \text{ not } D0 +$
 $C1 \text{ not } C0 \text{ not } D1 D0 +$
 $C1 C0 \text{ not } D1 \text{ not } D0 +$
 $C1 C0 \text{ not } D1 D0 +$
 $C1 C0 D1 \text{ not } D0 =$
- $= C1 \text{ not } D1 + C0 \text{ not } D1 \text{ not } D0 +$
 $C1 C0 \text{ not } D0$



Comparatore

	P0	P1	P2	P3	
A	0	0	1	1	
B	0	1	1	0	
C	0	0	0	1	A>B **
D	0	1	0	0	B>A
E	1	0	1	0	A=B

- $E = E3 E2 E1 E0$



Sommatore, Half Adder

Ai	Bi	Si	Ri
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

A B

$S_i = A_i \oplus B_i$ $R_i = A_i \cdot B_i$

HA

- Verifica della scrittura delle equazioni per Si e Ri
- Si = somma, Sum
- Ri = riporto, Rest
- Non tiene conto del resto del bit precedente

Paolo Nesi, Univ. Firenze, Italy, 2003-06
150

Full Adder, Sommatore Completo

Ai	Bi	R _{i-1}	Si	Ri
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

A B R_{i-1}

$S_i = A_i \oplus B_i \oplus R_{i-1}$

$R_i = (A_i \cdot B_i) \vee (A_i \oplus B_i) \cdot R_{i-1}$

HA

HA


FA

- $S_i = \text{not } A \text{ not } B R + \text{not } A B \text{ not } R + A \text{ not } B \text{ not } R + A B R$
- $R_i = \text{not } A B R + A \text{ not } B R + A B \text{ not } R + A B R$
- $R_i = R(\text{not } AB + \text{not } BA) + AB$

Paolo Nesi, Univ. Firenze, Italy, 2003-06
151

Full Adder, Sommatore Completo

- $S = \text{not } A \text{ not } B R + \text{not } A B \text{ not } R + A \text{ not } B \text{ not } R + A B R$
 - Poiche': $D \text{ xor } F = \text{not } DF + D \text{ not } F$
 - Poiche': $\text{not } (D \text{ xor } F) = \text{not } DF + D \text{ not } F$
- $Si = \text{not } R (A \text{ xor } B) + R (\text{not } A \text{ not } B + AB)$
- $Si = \text{not } R (A \text{ xor } B) + R \text{ not } (A \text{ xor } B)$
- $Ri = R(\text{not } AB + \text{not } BA) + AB$



Paolo Nesi, Univ. Firenze, Italy, 2003-06

152

Full Adder, Sommatore Completo

R: 1

D

S:

A: D

P:

D

G:


B: D

R:

D

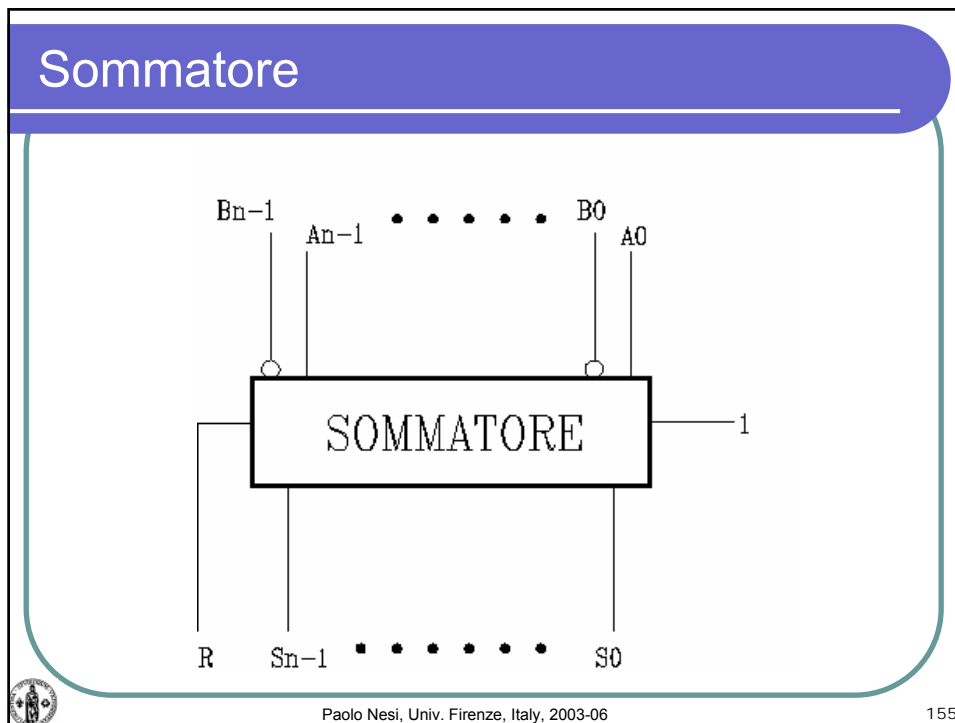
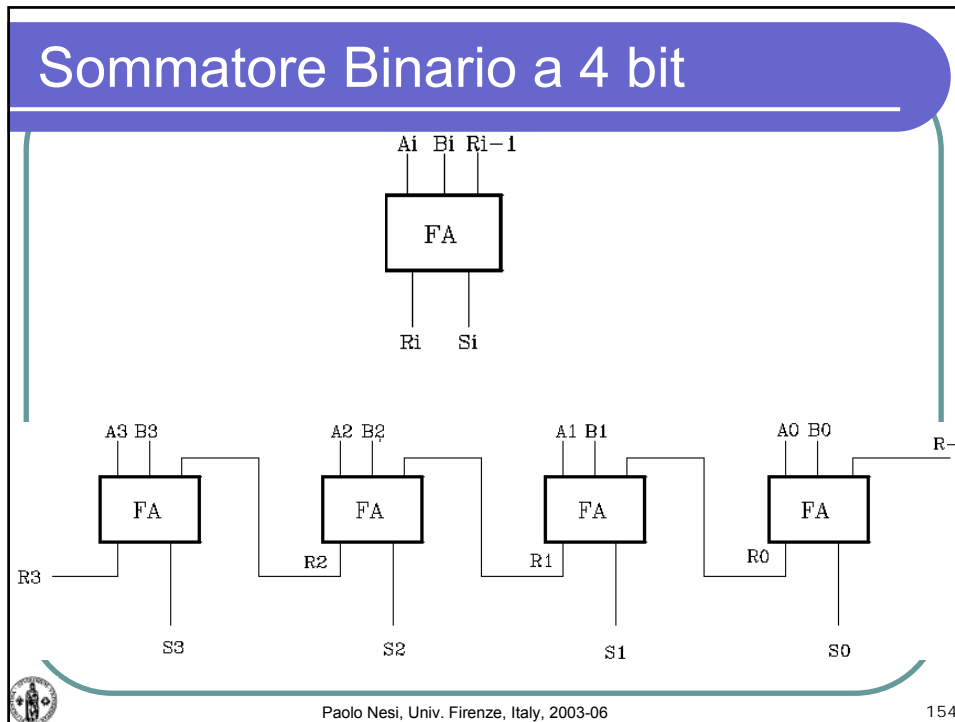
G:

- $Si = \text{not } R (A \text{ xor } B) + R \text{ not } (A \text{ xor } B)$
- $Ri = R (A \text{ xor } B) + AB$
- $Pi =$
- $Gi =$
- Estrarre equazioni ?



Paolo Nesi, Univ. Firenze, Italy, 2003-06

153



Sommatore controllato

a) Controllo del trabocco

b) Schematizzazione usuale

- Controllo del traboccamento, due termini, due casi
- $B_{n-1} A_{n-1} \text{ not } S_{n-1}$ or $\text{not } B_{n-1} \text{ not } A_{n-1} S_{n-1}$
- Riporto sul bit di segno, oppure riporto fuori dal bit di segno

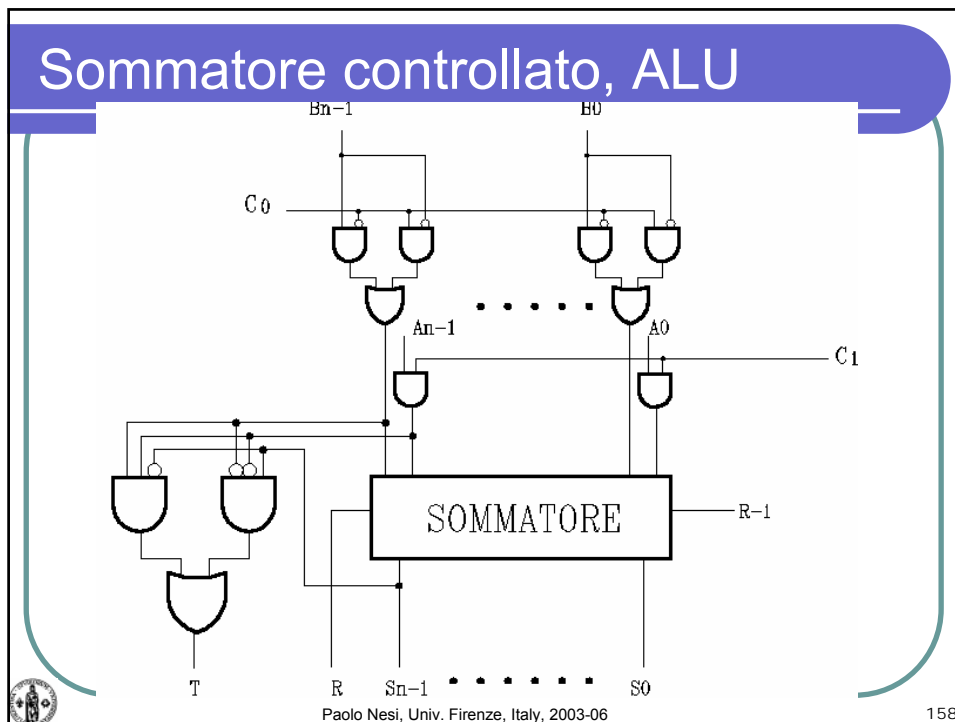
Paolo Nesi, Univ. Firenze, Italy, 2003-06
156

Traboccamento

- $B_{n-1} A_{n-1} \text{ not } S_{n-1}$ or $\text{not } B_{n-1} \text{ not } A_{n-1} S_{n-1} = T$
- Riporto sul bit di segno, oppure riporto fuori dal bit di segno

- Sono le uniche combinazioni critiche delle 8 che vengono fuori dai bit $B_{n-1} A_{n-1} S_{n-1}$

Paolo Nesi, Univ. Firenze, Italy, 2003-06
157



158

La ALU, controlli e risultati

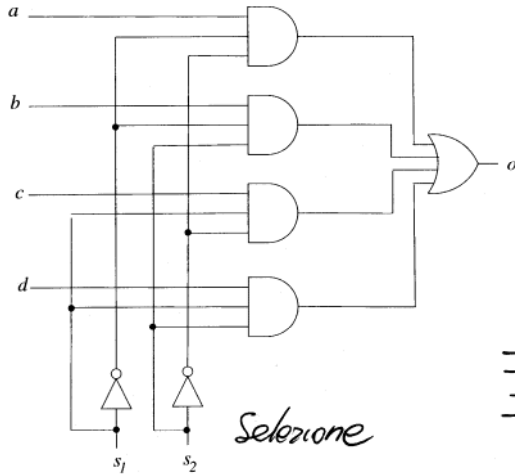
Q	C ₀	R ₁	S
0	0	0	B
0	0	1	B+1 <i>incremento</i>
0	1	0	B <i>comp. = 1</i>
0	1	1	B+1 = +B <i>CHS C₂(B)</i>
1	0	0	A+B
1	0	1	A+B+1
1	1	0	A+B̄ = A-B-1
1	1	1	A+B̄+1 = A-B <i>diff. C₂(B)</i>

Paolo Nesi, Univ. Firenze, Italy, 2003-06

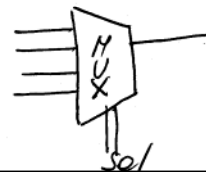
159

Multiplexer o Selettore

- Lo scopo e' selezionare uno degli ingressi e riportarlo in uscita.
- 2^n segnali di ingresso vengono selezionati in base al valore degli N segnali di selezione



s_1	s_2	o
0	0	a
0	1	b
1	0	c
1	1	d



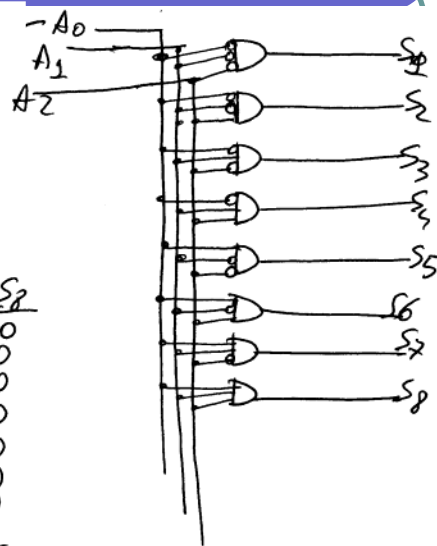
Selezione

160

Decoder or Demultiplexer

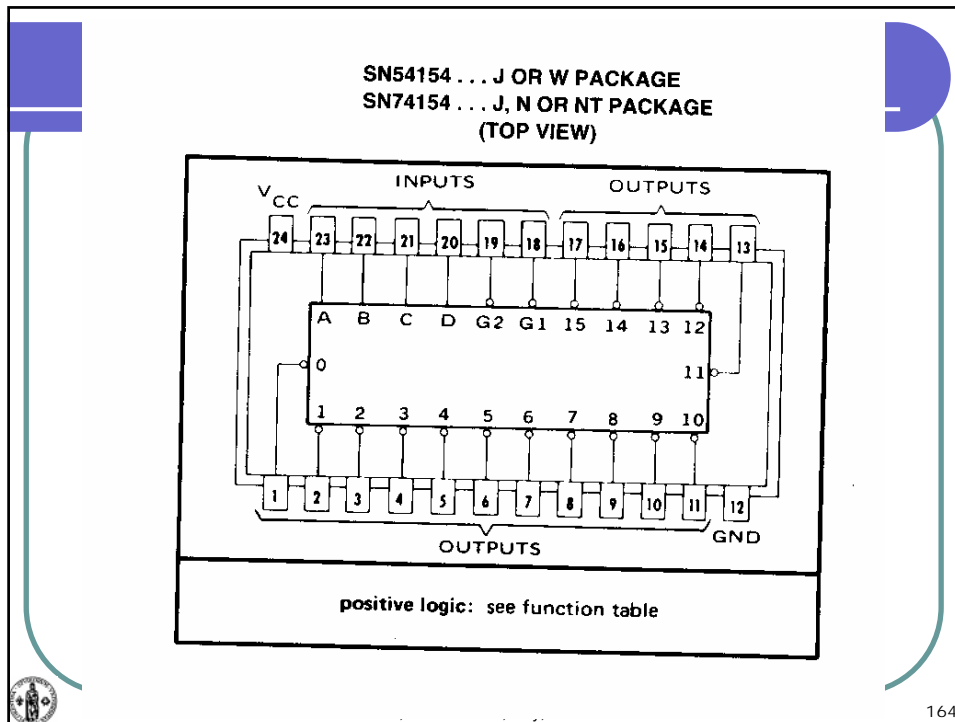


A_0	A_1	A_2	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1



Paolo Nesi, Univ. Firenze, Italy, 2003-06

161



164

logic

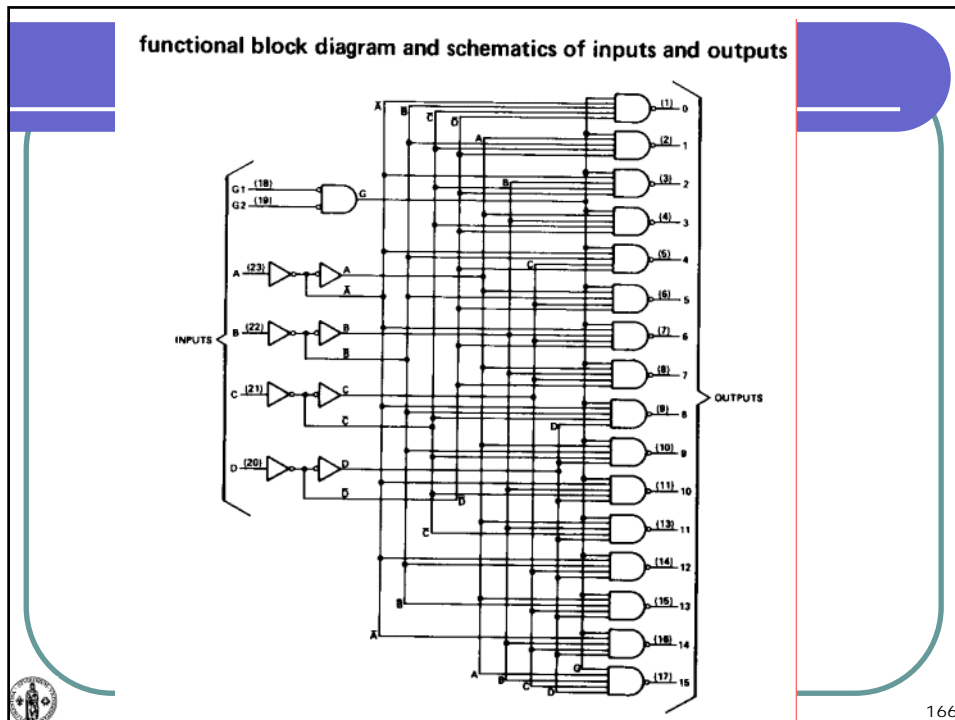
FUNCTION TABLE

INPUTS		OUTPUTS																				
G1	G2	D	C	B	A	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
L	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	L	L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	L	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	L	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H
L	L	H	L	L	L	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H
L	L	H	L	H	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H
L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H
L	L	H	H	L	L	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H
L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H
L	L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H
L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H
L	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
H	L	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
H	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H

H = high level, L = low level, X = irrelevant

Paolo Nesi, Univ. Firenze, Italy, 2003-06

165

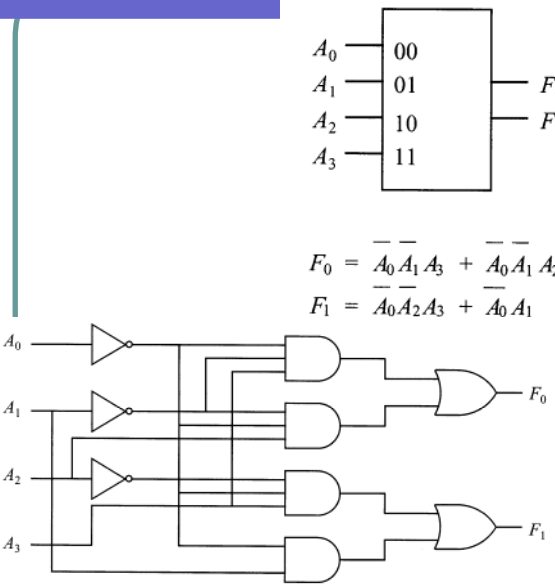


Simulatore

- Sulla pagina WEB potete scaricare un simulatore di reti logiche, per fare esercizio:
 - Realizzare delle reti
 - Presentare delle sequenze di dati in ingresso
 - Verificare la sequenza di uscita in base alle equazioni algebriche e alla tabella della verita'
 - Etc..
 - Etc.



Encoder



A_0 — 00
 A_1 — 01
 A_2 — 10
 A_3 — 11

$F_0 = \overline{A_0}A_1A_3 + \overline{A_0}A_1A_2$
 $F_1 = \overline{A_0}A_2A_3 + \overline{A_0}A_1$

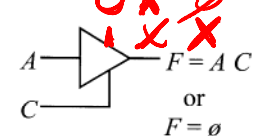
A_0	A_1	A_2	A_3	F_0	F_1
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	1	0
0	0	1	1	1	0
0	1	0	0	0	1
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	0	1
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	0	0
1	0	1	1	0	0
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0

Paolo Nesi, Univ. Firenze, Italy, 2003-06 168

Buffer tri-state

- \emptyset = Terzo stato, alta impedenza, isolato

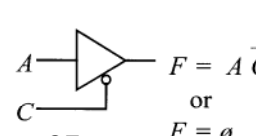
C	A	F
0	0	\emptyset
0	1	\emptyset
1	0	0
1	1	1



$F = AC$
 or
 $F = \emptyset$

C oppure OE
Tri-state buffer

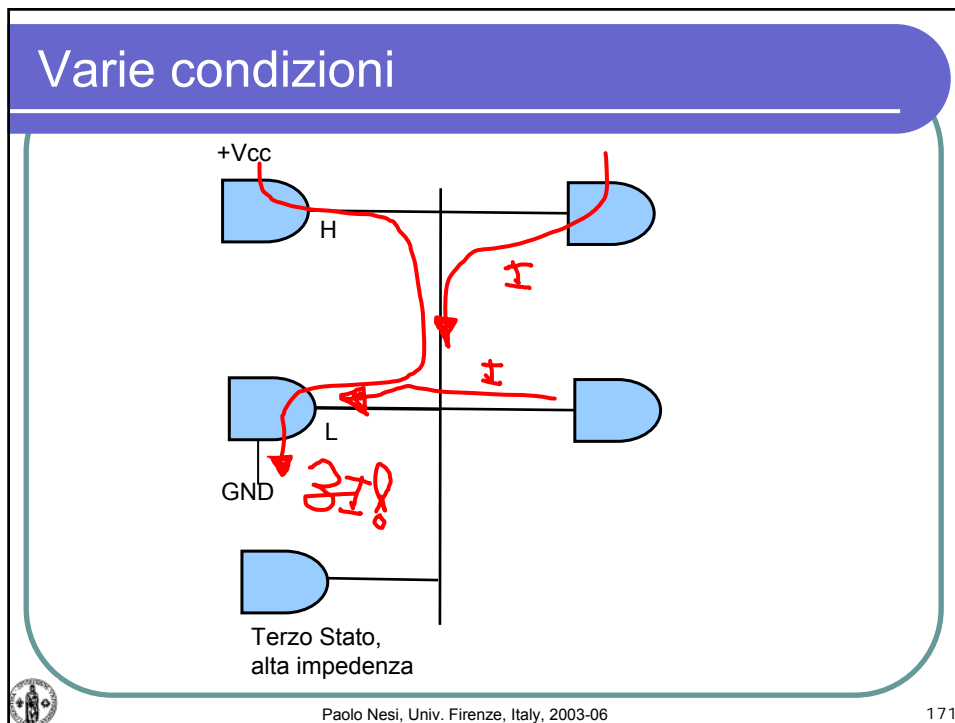
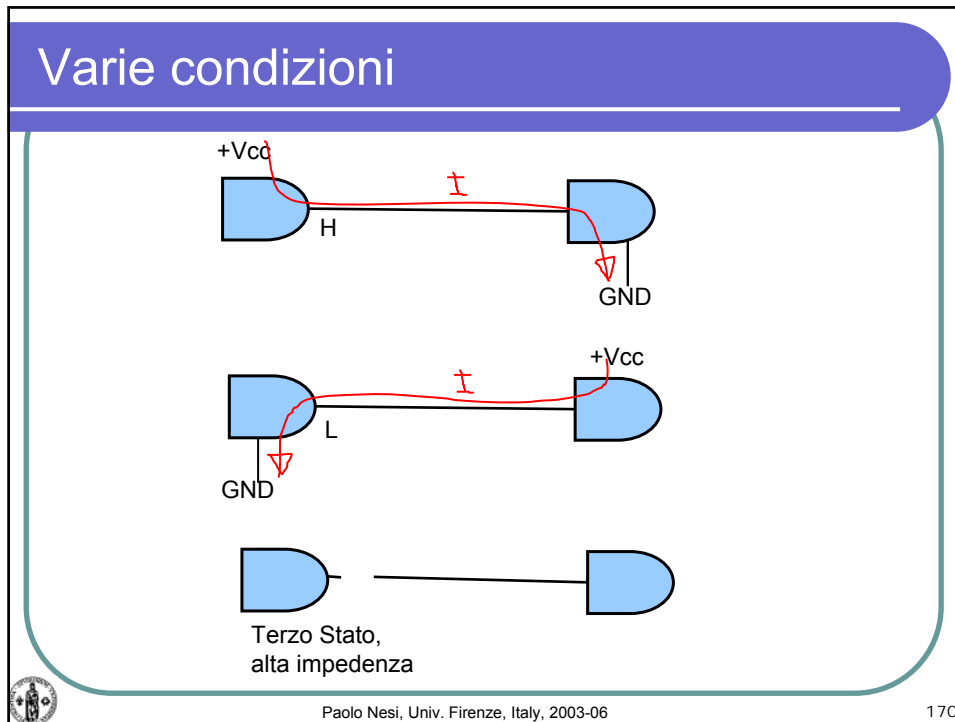
C	A	F
0	0	0
0	1	1
1	0	\emptyset
1	1	\emptyset



$F = A\overline{C}$
 or
 $F = \emptyset$

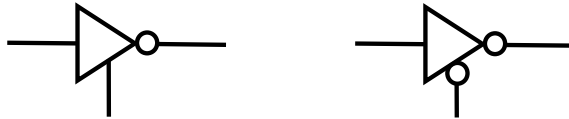
C oppure OE
Tri-state buffer, inverted control

Paolo Nesi, Univ. Firenze, Italy, 2003-06 169



Buffer tri-state negati

- \emptyset = Terzo stato, alta impedenza, isolato

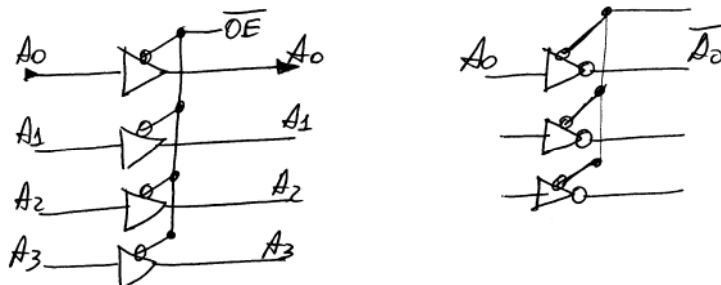


Paolo Nesi, Univ. Firenze, Italy, 2003-06

172

Buffer tristate per bus e gruppi di bit

- Può essere negato o meno e ha un Output Enable che può essere negato o meno
- Si veda il 240 e 241



Paolo Nesi, Univ. Firenze, Italy, 2003-06

173

TYPES SN54LS240, SN54LS241, SN54LS244, SN54S240, SN54S241, SN74LS240, SN74LS241, SN74LS244, SN74S240, SN74S241 OCTAL BUFFERS AND LINE DRIVERS WITH 3-STATE OUTPUTS

Typical Sink Current	Typical Source Current	Typical Propagation Delay Times	Typical Power Dissipation (each)
SN54LS240, SN74LS240: 12 mA	SN54LS240, SN74LS240: 12 mA	SN54LS240, SN74LS240: 17 ns	SN54LS240, SN74LS240: 138 mW
SN54LS241, SN74LS241: 12 mA	SN54LS241, SN74LS241: 12 mA	SN54LS241, SN74LS241: 17 ns	SN54LS241, SN74LS241: 138 mW
SN54LS244, SN74LS244: 48 mA	SN54LS244, SN74LS244: 48 mA	SN54LS244, SN74LS244: 9 ns	SN54LS244, SN74LS244: 538 mW
SN54S240, SN74S240: 84 mA	SN54S240, SN74S240: 84 mA	SN54S240, SN74S240: 6 ns	SN54S240, SN74S240: 538 mW

description

These octal buffers and line drivers are designed specifically to improve both the performance and density of three-state memory address drivers, clock drivers, and bus-oriented receivers and transmitters. The designer has a choice of selected combinations of inverting and noninverting outputs, symmetrical G (active-low output control) inputs, and complementary G and \bar{G} inputs. These devices feature high fan-out, improved fan-in, and 400-mV noise-margin. The SN74LS' and SN74S' can be used to drive terminated lines down to 133 ohms.

schematics of inputs and outputs

'LS240, 'LS241, 'LS244

EQUIVALENT OF EACH INPUT

INPUT

TYPICAL OF ALL OUTPUTS

'LS240, 'LS241, 'LS244; R = 50 Ω NOM

'S240, 'S241; R = 25 Ω NOM

TEXAS INSTRUMENTS

7-381

Paolo Nesi, Univ. Firenze, Italy, 2003-06

174

mentary G and \bar{G} inputs. These devices feature high fan-out, improved fan-in, and 400-mV noise-margin. The SN74LS' and SN74S' can be used to drive terminated lines down to 133 ohms.

schematics of inputs and outputs

'LS240, 'LS241, 'LS244

EQUIVALENT OF EACH INPUT

INPUT

TYPICAL OF ALL OUTPUTS

'LS240, 'LS241, 'LS244; R = 50 Ω NOM

'S240, 'S241; R = 25 Ω NOM

SN54LS241, SN54S241 ... J
SN74LS241, SN74S241 ... J OR N
(TOP VIEW)

SN54LS244 ... J
SN74LS244 ... J OR N
(TOP VIEW)

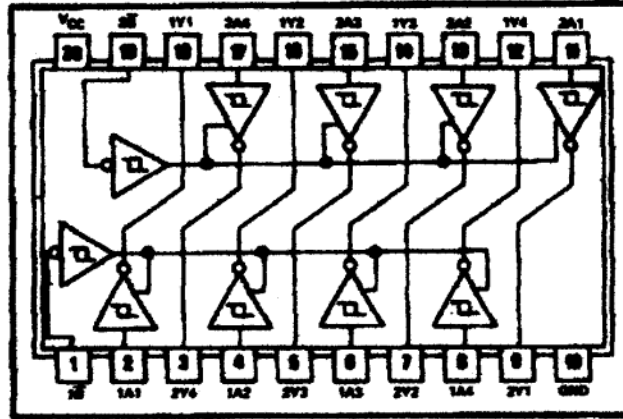
Paolo Nesi, Univ. Firenze, Italy, 2003-06

Paolo Nesi, Univ. Firenze, Italy, 2003-06

175

SN54LS240

SN54LS240, SN54S240 ... J
SN74LS240, SN74S240 ... J OR N
(TOP VIEW)

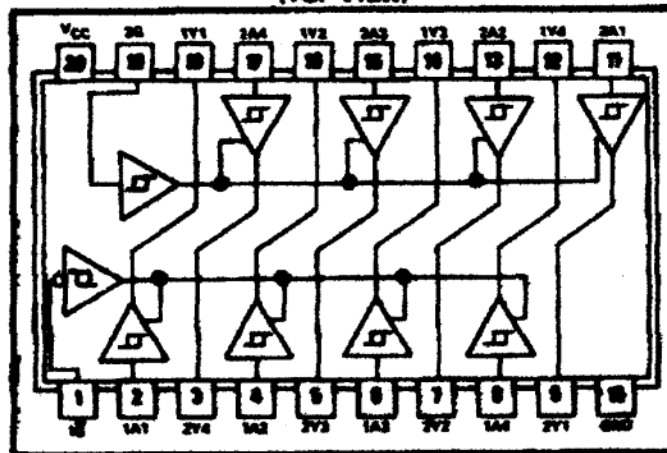


Paolo Nesi, Univ. Firenze, Italy, 2003-06

176

SN54LS241

SN54LS241, SN54S241 ... J
SN74LS241, SN74S241 ... J OR N
(TOP VIEW)



Paolo Nesi, Univ. Firenze, Italy, 2003-06

177

IL BUS con Tri State

The diagram shows a horizontal line labeled "linea del bus" with a handwritten red note "3 F.I.". Three tri-state buffers, T_1 , T_2 , and T_3 , are connected to the bus. Their control inputs are C_{T1} , C_{T2} , and C_{T3} respectively. Three AND gates, R_1 , R_2 , and R_3 , are also connected to the bus. Their control inputs are C_{R1} , C_{R2} , and C_{R3} respectively.

- 3 Buffer tri-state come ingressi al bus che devono essere attivi solo uno alla volta
- 3 porte AND che possono leggere contemporaneamente la stessa linea di BUS

179

Ricetrasmisione dati, Transceiver

Disposizione dei pin

8286

A ₀	1	20	V _{CC}
A ₁	2	19	B ₀
A ₂	3	18	B ₁
A ₃	4	17	B ₂
A ₄	5	16	B ₃
A ₅	6	15	B ₄
A ₆	7	14	B ₅
A ₇	8	13	B ₆
OE	9	12	B ₇
GND	10	11	T

8287

A ₀	1	20	V _{CC}
A ₁	2	19	B ₀
A ₂	3	18	B ₁
A ₃	4	17	B ₂
A ₄	5	16	B ₃
A ₅	6	15	B ₄
A ₆	7	14	B ₅
A ₇	8	13	B ₆
OE	9	12	B ₇
GND	10	11	T

Diagrammi logici

The 8286 diagram shows a bidirectional bus where local data (A₀-A₇) is sent to system data (B₀-B₇) and vice versa. The OE pin controls the output enable. The 8287 diagram shows a similar setup but with a transmission pin (T) for system-to-local data transfer.

Nomi dei pin

A ₀ -A ₇	Bus dati locale
B ₀ -B ₇	Bus dati di sistema
OE	Abilitazione dell'output
T	Trasmisione

180

Ricetrasmisione dati, Transceiver

Nomi dei pin

A ₀ -A ₇	Bus dati locale
B ₀ -B ₇	Bus dati di sistema
OE	Abilitazione dell'output
T	Trasmissione

OE'	T	Operation
0	1	A to B
0	0	B to A
1	X	Alta impedenza

Paolo Nesi, Univ. Firenze, Italy, 2003-06
181

TTL MSI

TYPES SN54LS245, SN74LS245

OCTAL BUS TRANSCEIVERS WITH 3-STATE OUTPUTS

- Bi-directional Bus Transceiver in a High-Density 20-Pin Package
- 3-State Outputs Drive Bus Lines Directly
- P-N-P Inputs Reduce D-C Loading on Bus Lines
- Hysteresis at Bus Inputs Improve Noise Margins
- Typical Propagation Delay Times, Port-to-Port . . . 8 ns
- Typical Enable/Disable Times . . . 17 ns

TYPE	I _{OL} (SINK CURRENT)	I _{OH} (SOURCE CURRENT)
SN54LS245	17 mA	-12 mA
SN74LS245	24 mA	-15 mA

description

These octal bus transceivers are designed for asynchronous two-way communication between data buses. The control function implementation minimizes external timing requirements.

The device allows data transmission from the A bus to the B bus or from the B bus to the A bus depending upon the logic level at the direction control (DIR) input. The enable input (G) can be used to disable the device so that the buses are effectively isolated.

The SN54LS245 is characterized for operation over the full military temperature range of -55°C to 125°C. The SN74LS245 is characterized for operation from 0°C to 70°C.

schematics of inputs and outputs

FUNCTION TABLE

ENABLE CONTROL G	DIRECTION DIR	OPERATION
L	L	B data to A bus
L	H	A data to B bus
H	X	Isolation

H = high level, L = low level, X = irrelevant

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1) 5 V

Input voltage 5 V

Operating free-air temperature range: SN54LS245 -55°C to 125°C

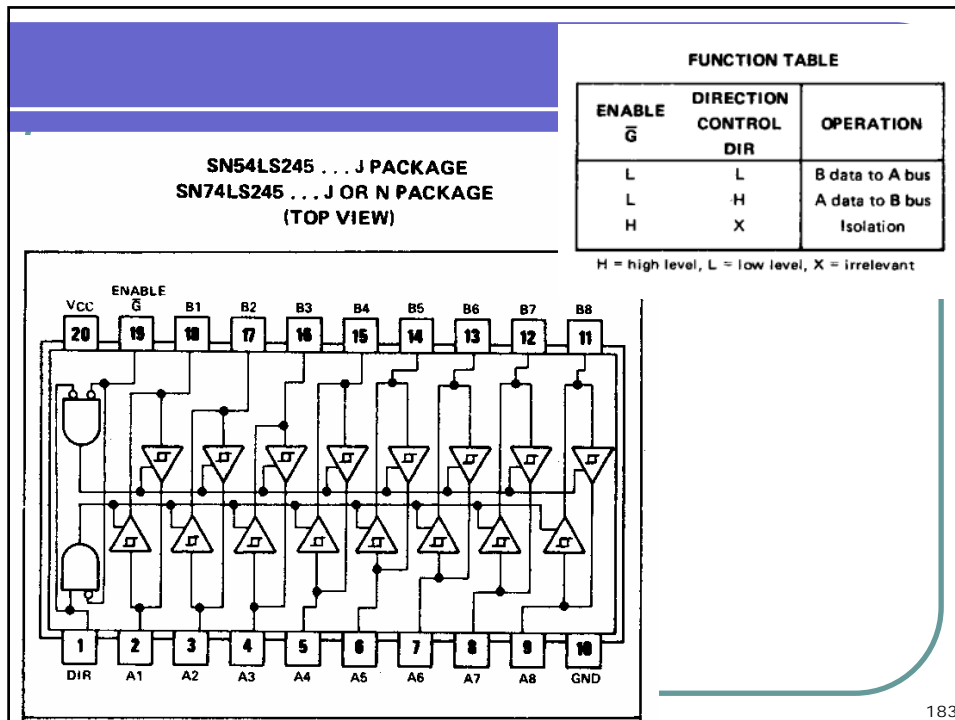
SN74LS245 0°C to 70°C

Storage temperature range -65°C to 150°C

Off state output voltage 5.5V

NOTE 1: Voltage values are with respect to network ground terminal.

Paolo Nesi, Univ. Firenze, Italy, 2003-06
182



183

Calcolatori Elettronici

CDL in Ingegneria Elettronica
Facoltà di Ingegneria,
Università degli Studi di Firenze

Nuovo Ordinamento

Parte 3, Logica Sequenziale

Prof. Paolo Nesi

<http://www.dsi.unifi.it/~nesi>

nesi@dsi.unifi.it

2006



Logica Sequenziale

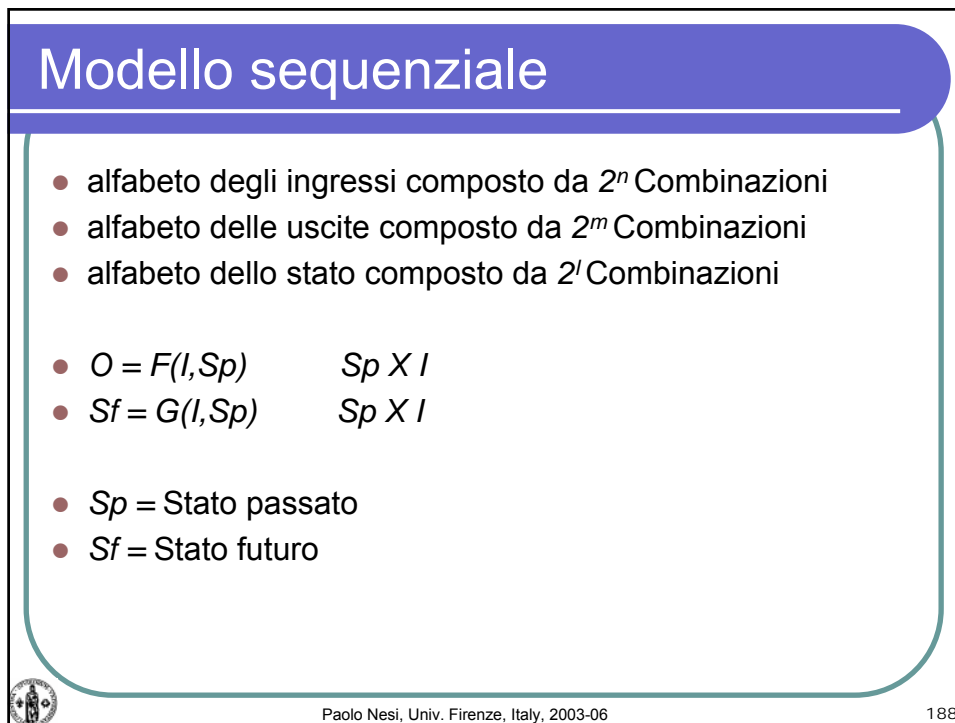
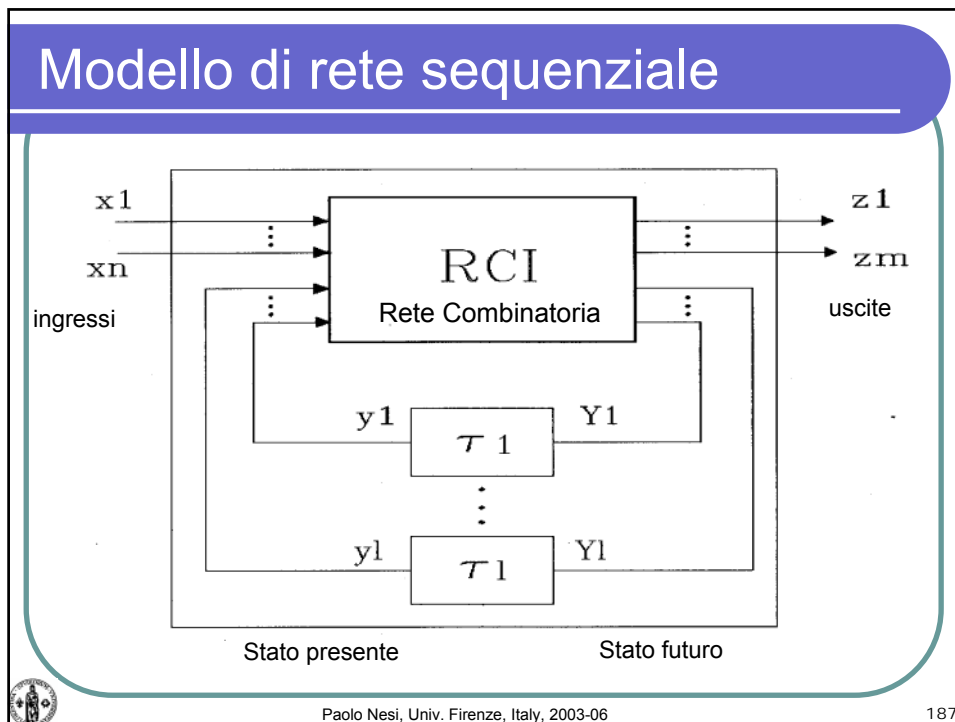
- La logica sequenziale si distingue da quella combinatoria poiché il valore delle uscite dipende dall'evoluzione temporale degli ingressi che si concretizza nello stato del circuito, come se questo avesse una memoria.
- Circuiti sequenziali si possono descrivere con macchine a stati



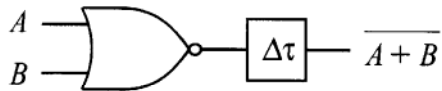
Reti Sequenziali

- Nelle **reti combinatorie** l'uscita è funzione solo degli ingressi. Per una certa configurazione degli ingressi $I = \{x_1, x_2, \dots, x_n\}$, risultano definiti dalla funzione di trasferimento i segnali di uscita
$$O = \{z_1, z_2, \dots, z_m\} = f(I)$$
- Nelle **reti sequenziali** l'uscita viene calcolata anche in base allo stato S della rete
$$S = \{y_1, y_2, \dots, y_l\}.$$
- Quindi si ha $O = f(I, S)$,
- mentre lo stato stesso $S = g(I, S \text{ passato})$

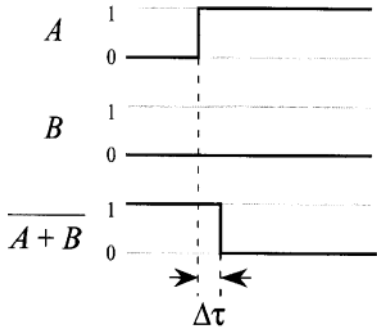





Si sfrutta il ritardo delle porte



Porta non piu' completamente ideale ma che presenta un ritardo di trasmissione



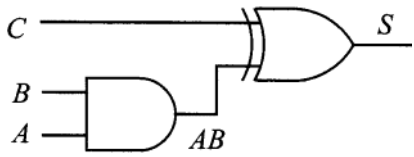
Timing Behavior

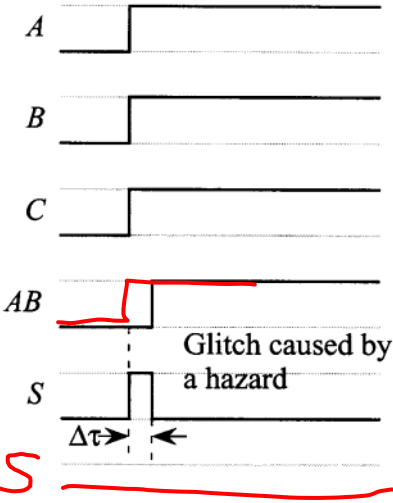


Paolo Nesi, Univ. Firenze, Italy, 2003-06


189

Effetti indesiderati dei ritardi





Glitch caused by a hazard

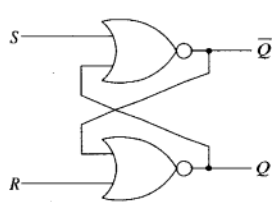


Paolo Nesi, Univ. Firenze, Italy, 2003-06


190

Latch di NOR

- S : Set (chiudo)
- R : Reset (apro)
- $Q_{n+1} = S + \text{not } R Q_n$
- Se S,R hanno entrambi valore 1 si ha un uscita non determinabile pertanto viene detto non consentito
- Latch: chiavistello

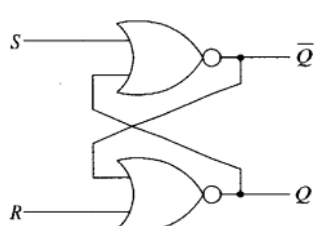


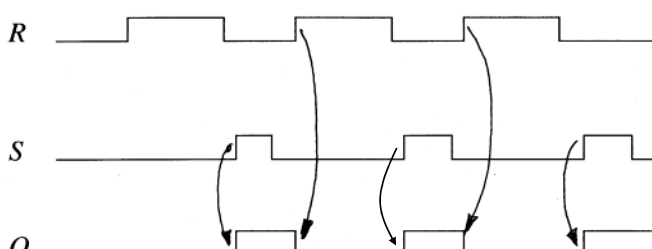
Q_t	S_t	R_t	Q_{t+1}	
0	0	0	0	← non cambia
0	0	1	0	← pone a 0
0	1	0	1	← pone a 1
0	1	1	non consentito	
1	0	0	1	← non cambia
1	0	1	0	← pone a 0
1	1	0	1	← pone a 1
1	1	1	non consentito	



Paolo Nesi, Univ. Firenze, Italy, 2003-06
191

Latch di NOR, rete asincrona

S	R	Q
0	0	non cambia
0	1	0
1	0	1
1	1	non ammesso






Paolo Nesi, Univ. Firenze, Italy, 2003-06
192

La versione reale, Flip Flop S-R

Timing Behavior

S	R	Q
0	0	non cambia
0	1	0
1	0	1
1	1	non ammesso

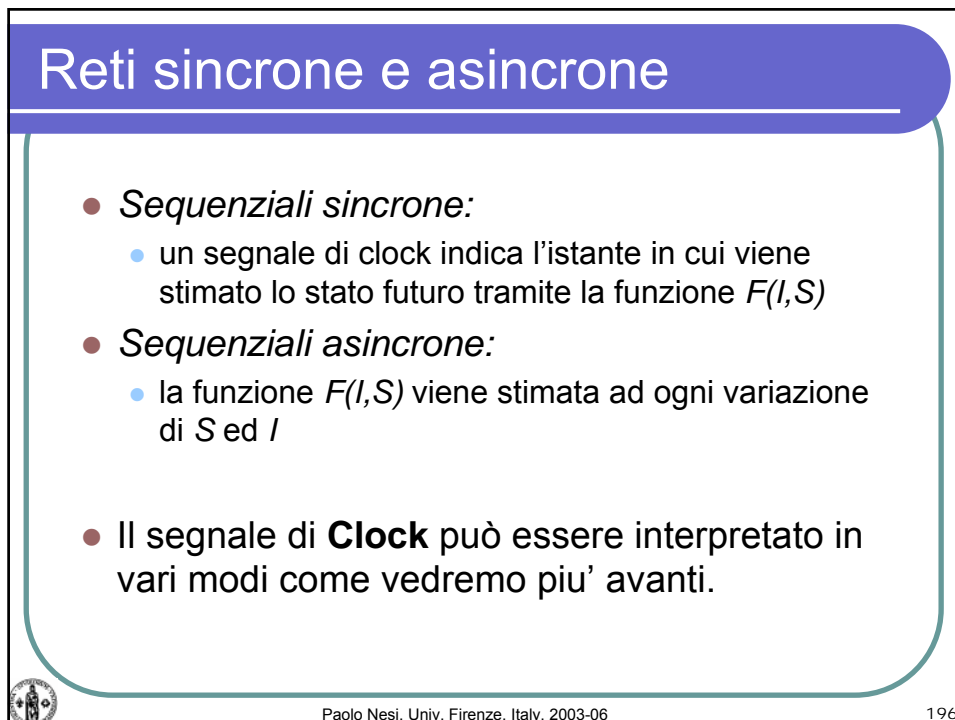
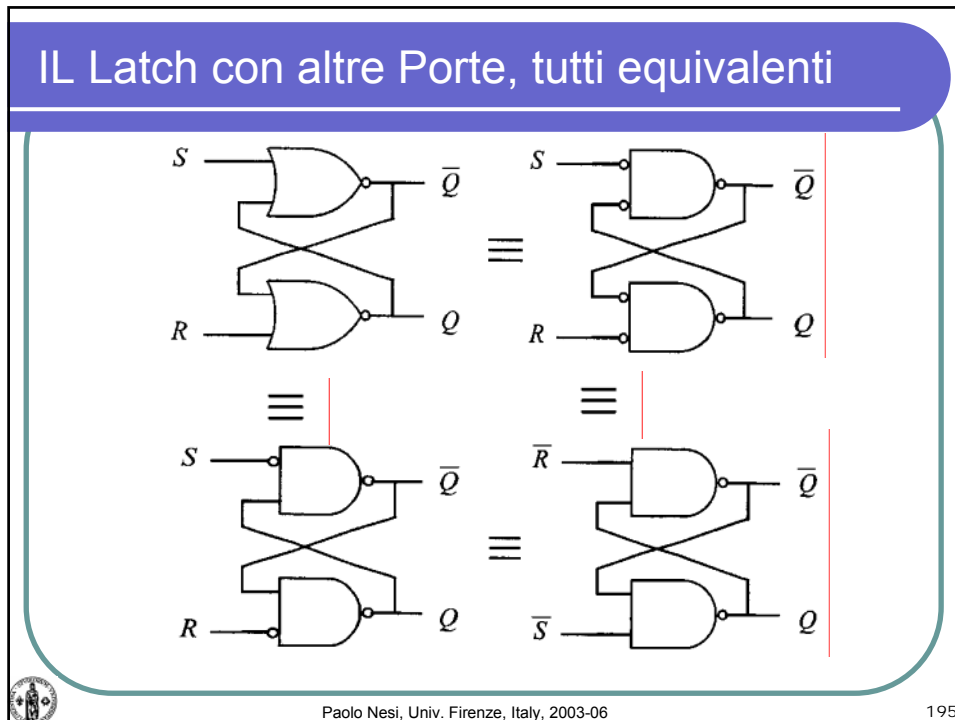
Paolo Nesi, Univ. Firenze, Italy, 2003-06
193

Problemi con valori di ingresso

non Ammissibile !!

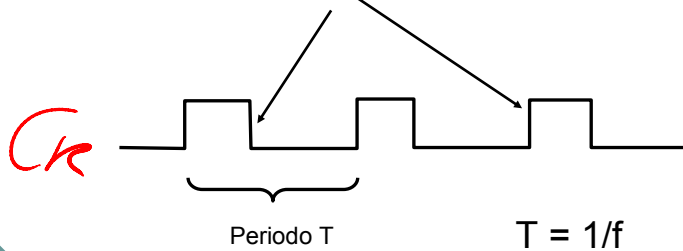
da evitare

Paolo Nesi, Univ. Firenze, Italy, 2003-06
194



IL Segnale di Clock

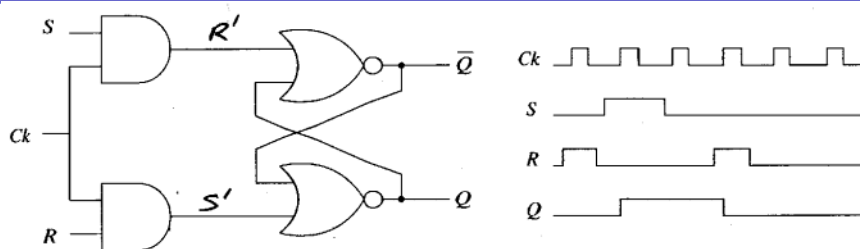
- T = Periodo di Clock = $1/F$
- F = Frequenza di Clock
- 1Mhz di freq. 1 microsecondo di T
- Duty Cycle, rapporto fra il tempo ad 1 e il tempo di T
- Fronte di salita
- Fronte di discesa



Paolo Nesi, Univ. Firenze, Italy, 2003-06

197

Flip Flop SR (sincrono)



- Nei Flip Flop sincroni il Latch acquisisce gli ingressi e pertanto valuta l'uscita **solo quando lo abilita il segnale di clock**
- Vi possono essere sempre problemi dovuti alla propagazione del segnale e alla realizzazione non ideale delle porte



Paolo Nesi, Univ. Firenze, Italy, 2003-06

198

Flip Flop SR (sincrono)

Si attiva sul fronte di salita come specificato dal triangolo

$Q(n+1) = S + \text{not } R Q(n)$

Paolo Nesi, Univ. Firenze, Italy, 2003-06
199

Flip Flop di tipo JK

- $Q(t+1) = \text{not } Q_t J + Q_t \text{not } K$

J	K	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	not Q(t)

Paolo Nesi, Univ. Firenze, Italy, 2003-06
200

Flip Flop di tipo D, Delay

- Si ottiene collegando S con un not ad R
- Sincronizza/allinea il segnale in ingresso a quello del clock
- Realizza un ritardo massimo pari al periodo di clock, $Q = D$

S	R	Q(t+1)
0	0	---
0	1	0
1	0	1
1	1	---

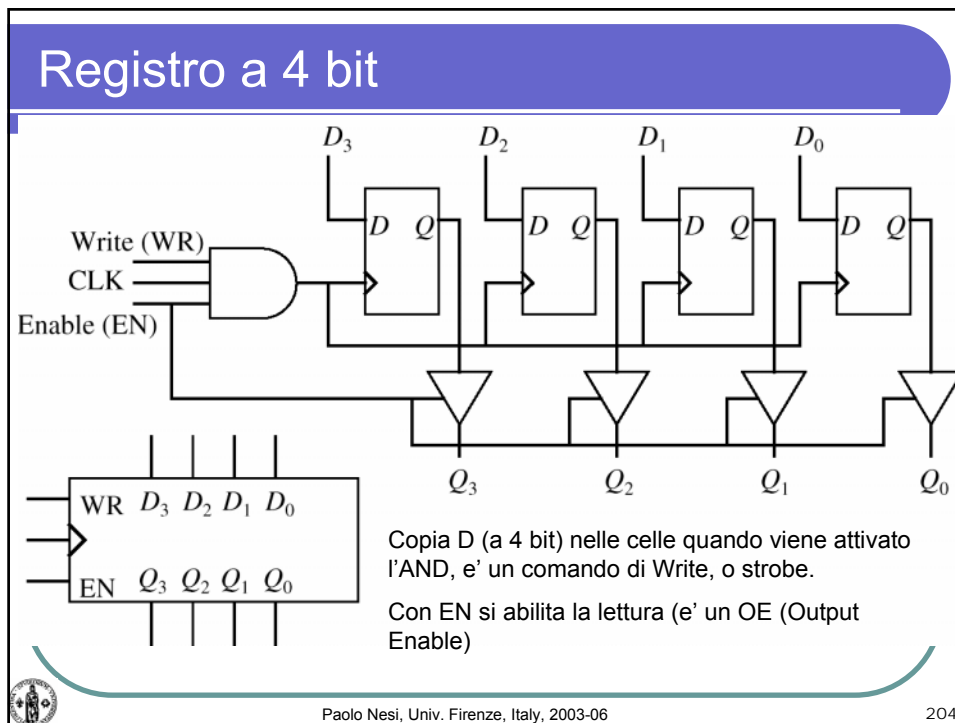
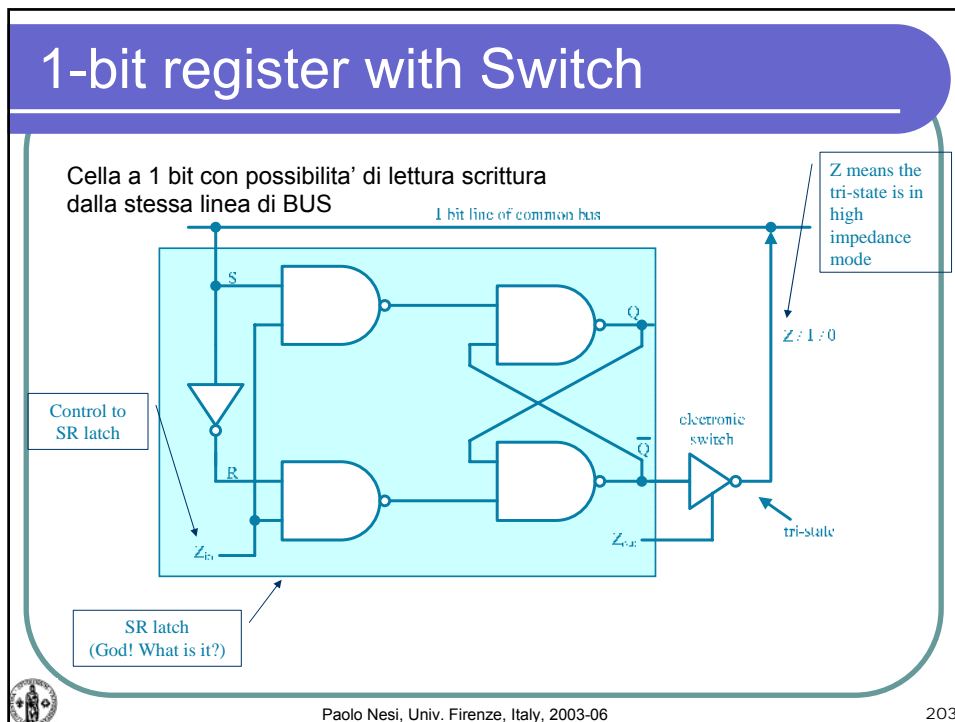
Paolo Nesi, Univ. Firenze, Italy, 2003-06
201

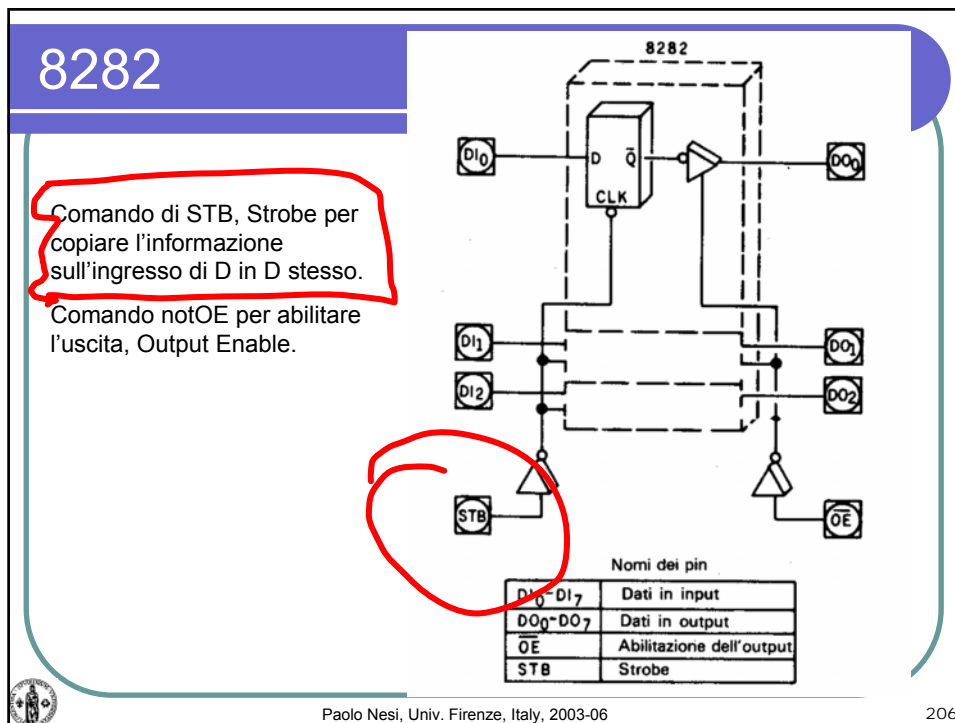
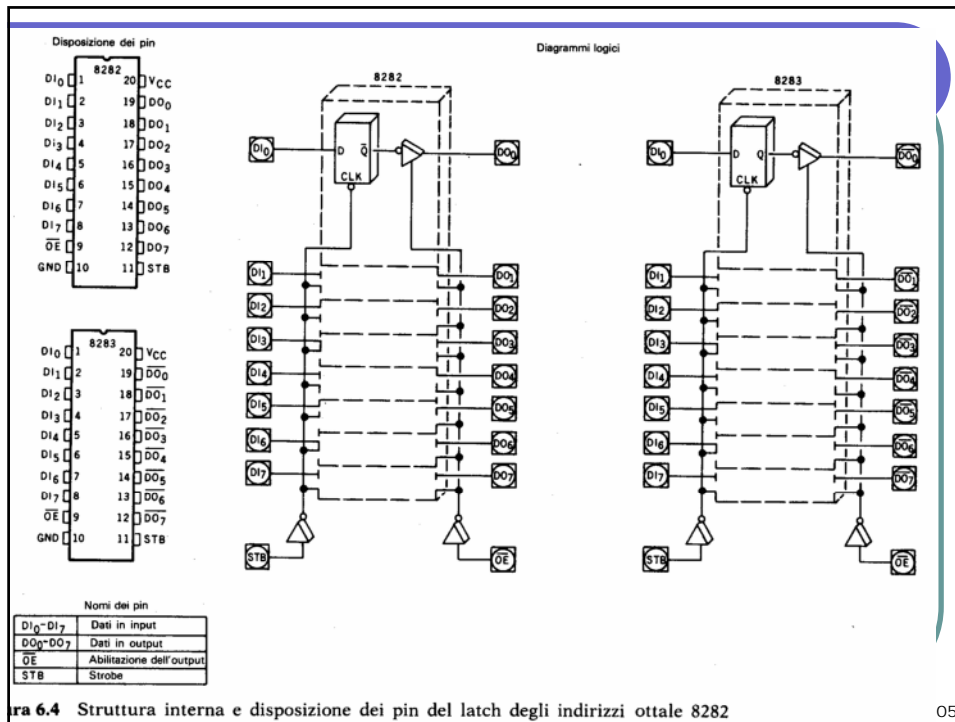
Flip Flop di tipo T, Toggle

- Commutazione, $Q(t+1) = T Q(t)$
- Si ottiene collegando J con K in un JK

J	K	Q(t+1)
0	0	Q(t)
0	1	---
1	0	---
1	1	not Q(t)

Paolo Nesi, Univ. Firenze, Italy, 2003-06
202





Shift Register, Registro a Scorrimento

Q_0 Q_1 Q_2 Q_3

00010
0001 *ad ogni colpo di ck*

Il registro scorre la sequenza che arriva in IN, ad ogni colpo si scorre di un bit.

207

Control	Function
$c_1 \ c_0$	No change
0 0	Shift left
0 1	Shift right
1 0	Parallel load
1 1	

208

Contatore con flip flop T

Timing Diagram:

CK: Clock signal (square wave)

Q0: Output of the least significant bit, toggles on every clock edge.

Q1: Output of the second bit, toggles on every second clock edge.

Q2: Output of the third bit, toggles on every fourth clock edge.

Q3: Output of the most significant bit, toggles on every eighth clock edge.

Count sequence: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

Characteristics:

- T**: Scattano sul fronte di discesa (Trigger on falling edge)
- 1/2 F**: Contatore sui bit da 0000 a 1111
- 1/4 F**: Divisore in frequenza
- 1/8 F**: Divisore in frequenza
- 1/16 F**: Multiplicatore in periodo

Paolo Nesi, Univ. Firenze, Italy, 2003-06 209

Contatore con JK(T) modulo 8

Block Diagram:

ENABLE (EN) and RESET inputs to a MOD(8) COUNTER block.

Outputs: Q₂, Q₁, Q₀

Timing Diagram:

CLK: Clock signal

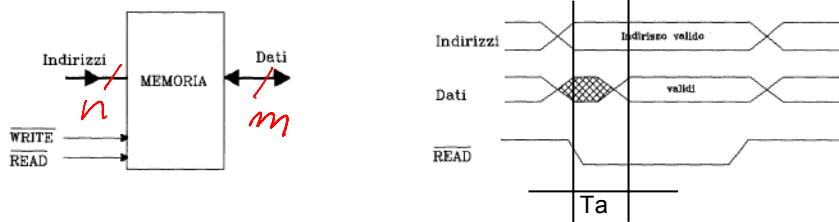
Q0: Output of the least significant bit

Q1: Output of the second bit

Q2: Output of the most significant bit

La Memoria, modello generale

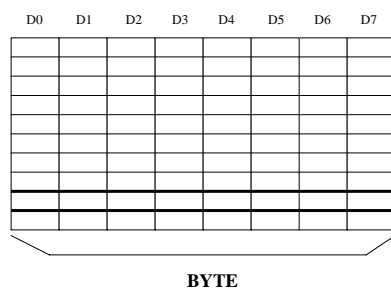
- Bit di Indirizzamento, Bus Indirizzi
- Bit dati, Bus dati
- Primo concetto di BUS
- Segnali di Controllo, Read, Write, Select...., Bus Controlli
- Tempo di lettura



Paolo Nesi, Univ. Firenze, Italy, 2003-06

212

Rappresentazione logica della Memoria



Vettore composto da celle in cui vengono immagazzinate informazioni.

Ogni cella ha un indice che costituisce il suo "indirizzo" all'interno della memoria.

Dati e informazioni contenuti sono in forma binaria.

La dimensione di una cella è data dal numero di bit che essa contiene: può essere di un bit, ma anche di 8, 16, 32, 64.



Paolo Nesi, Univ. Firenze, Italy, 2003-06

213

Tipi di Memoria

- Memorie a stato solido
 - RAM, Random Access Memory
 - SRAM, DRAM, etc.
 - ROM, Read Only Memory
 - PROM, Programmable ROM
 - EPROM, Erasable PROM
 - EEPROM, Electrically Erasable PROM
 - Flash.....
- Esistono anche Memorie su supporto magnetico
 - Dischi, nastri, HD, FD, etc.

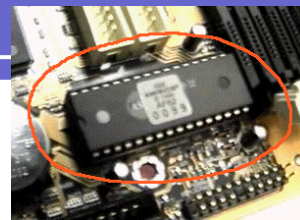


Paolo Nesi, Univ. Firenze, Italy, 2003-06

214

La Memoria ROM

ROM, Read Only Memory
(memoria di sola lettura)



I dati immagazzinati nella memoria ROM permangono sempre anche se viene a mancare l'alimentazione.

Contengono le istruzioni di base (**firmware, BIOS**) per consentire l'avvio: l'inizializzazione delle periferiche del calcolatore, il collegamento della stessa con i terminali di ingresso /uscita.



Paolo Nesi, Univ. Firenze, Italy, 2003-06

215

RAM, Random Access Memory

memoria di lettura scrittura.

conserva i dati intermedi delle elaborazioni, i risultati dei calcoli, ecc.

volatile, si perde l'informazione se viene a mancare l'alimentazione.

Si chiama ad accesso casuale poiché è possibile accedere ad ogni singola cella in modo diretto e pertanto la modalità casuale è possibile.

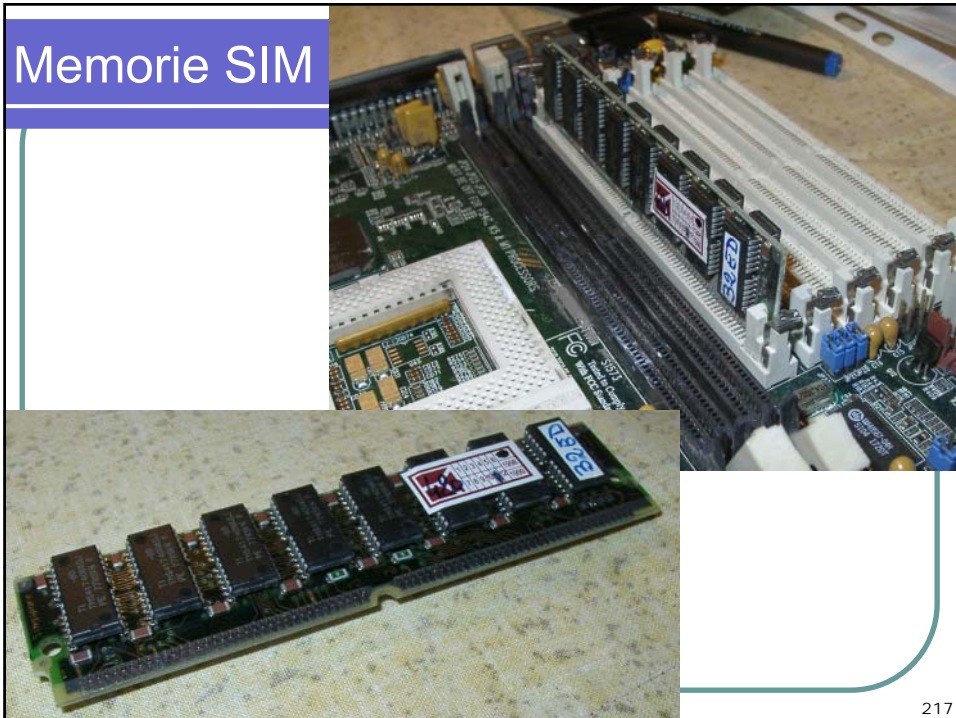
Questa modalità è da vedere contrapposta alla modalità sequenziale per la quale per accedere ad una cella si deve prima avere fatto accesso alla precedente, etc. Partendo dalla prima cella della memoria.



Paolo Nesi, Univ. Firenze, Italy, 2003-06

216

Memorie SIM



217

Altri tipi di Memoria

PROM (Programmable ROM)

Può essere programmata dal produttore una sola volta.

La programmazione di queste memorie avviene tramite un dispositivo elettronico detto programmatore che produce particolari tensioni di alimentazione e sequenze di operazioni al fine di provocare delle "lesioni" permanenti nella matrice che rappresenta la memoria.

EPROM (Electrically PROM)

Questa memoria può essere scritta e cancellata, ma solo globalmente sottoponendola a raggi UV per un certo numero di minuti. Dopo la cancellazione può essere riscritta.

Questa operazione può essere ripetuta un numero limitato di volte. Si noti che per la cancellazione è necessario smontare la EPROM dal circuito sul quale è installata.



Altri tipi di Memoria

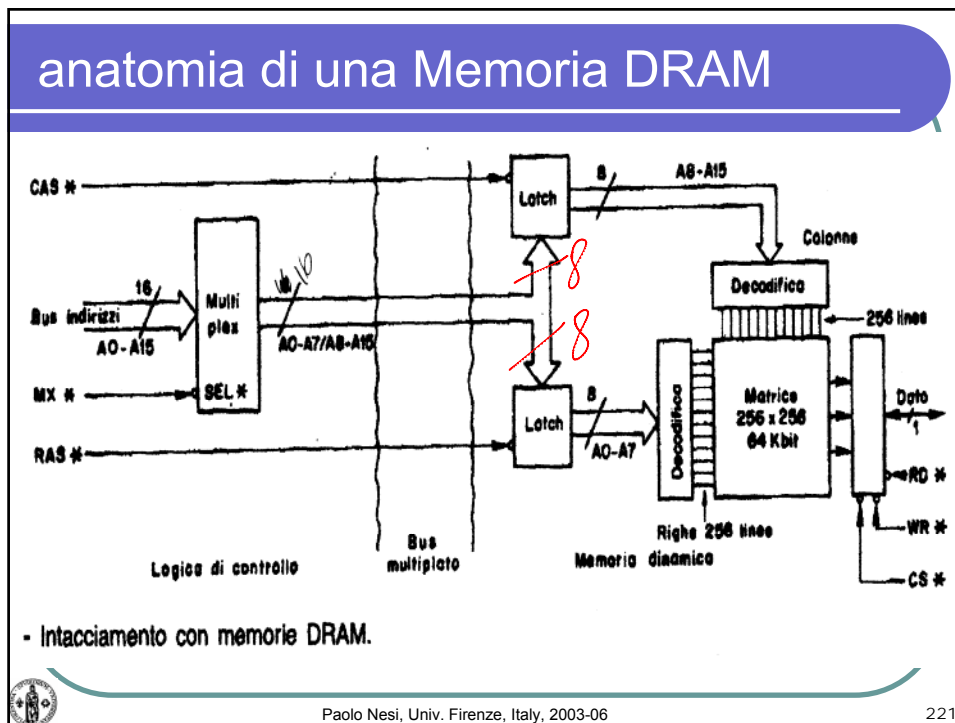
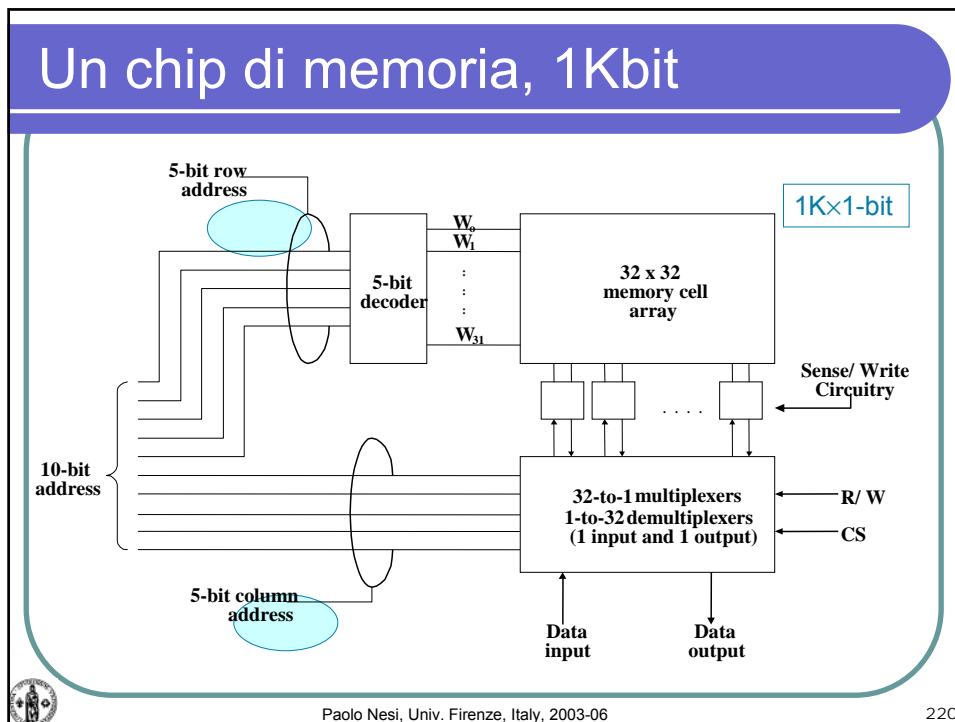
EEPROM (Electrically Erasable PROM)

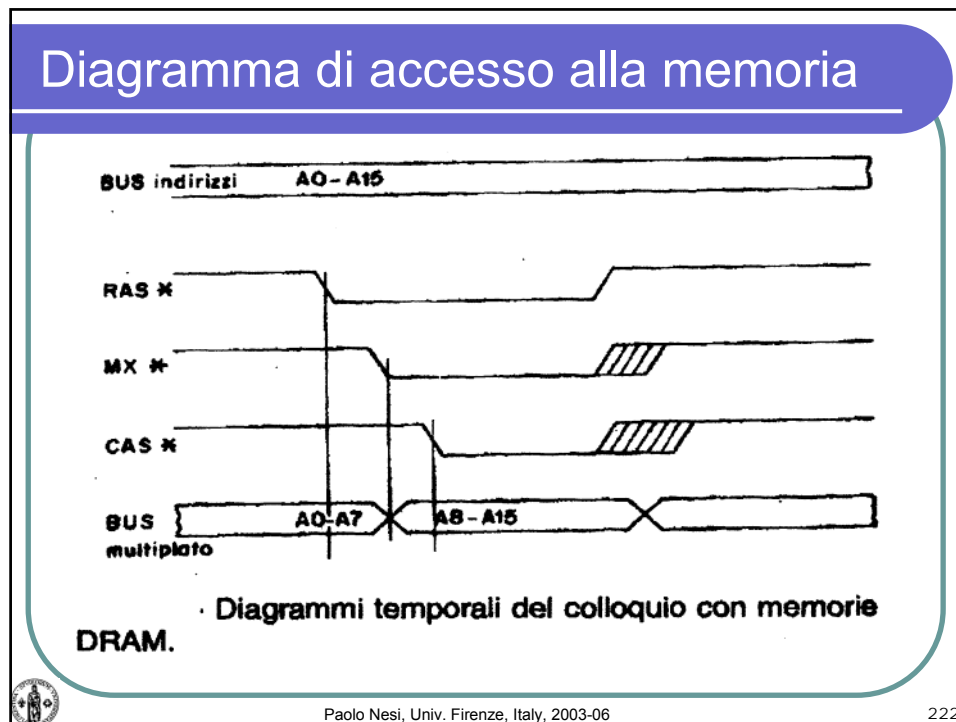
Come la EPROM ma le operazioni di cancellazione possono essere più selettive e vengono effettuate tramite segnali elettrici. Non è necessario rimuovere la EEPROM dal circuito per effettuare le cancellazioni. Dopo la cancellazione può essere riscritta. Questa operazione può essere effettuata un numero limitato di volte.

RAM, ROM, EPROM, PROM, EEPROM sono memorie cui si accede direttamente in base all'indirizzo della cella

Nei calcolatori vi è di solito la memoria RAM e EPROM; quest'ultima può essere utilizzata come memoria ROM (es: BIOS).







Memorie

- Con N segnali di indirizzamento si ha la possibilità di indirizzare 2^N diverse gruppi di celle
- Ogni gruppo di celle puo' essere di M bit, Per esempio 8, 16, o 32 (le taglie più probabili)
- Le memorie si possono comporre utilizzando il Cs (chip select) per raggiungere dimensioni maggiori
- Sui tempi di accesso e la composizione ritorneremo quando le vedremo dentro l'elaboratore.

Paolo Nesi, Univ. Firenze, Italy, 2003-06 223