

1222-2022
800 ANNI



UNIVERSITÀ
DEGLI STUDI
DI PADOVA



Consapevolezza Digitale

Codifica binaria delle informazioni; operatori logici, porte logiche e diagramma temporale

Andrea Gerosa – andrea.gerosa@unipd.it

Numeri binari e codifica binaria

- **Tipi di informazione**
 - **Numerica**
 - Dobbiamo elaborare un certo intervallo di valori
 - Vogliamo rappresentare tali dati in un formato tale per cui sia facile ed efficiente realizzare le operazioni aritmetiche
 - Usiamo numeri binari (con qualche modifica)
 - **Non numerica**
 - Possiamo gestire qualsiasi tipo di informazione, assegnando a ogni dato una combinazione di bit diversa (codice)
 - Possiamo quindi sfruttare comunque l'elaborazione digitale di queste informazioni

Es.: codifica binaria dei colori

Color	Binary Number
Red	000
Orange	001
Yellow	010
Green	011
Blue	101
Indigo	110
Violet	111

Numero di bit

$$2^n \geq M > 2^{(n-1)}$$

$n = \log_2 M$, con x il minimo intero maggiore o uguale a x .

◻ ◻ ◻

Quanti bit di codice sono necessari per rappresentare le cifre decimali da 0 a 9?

ASCII Character Codes

- American Standard Code for Information Interchange
- 7-bits:
 - 94 caratteri grafici
 - 34 caratteri non grafici

ASCII Character Codes

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	Ø	96	60	140	`	~
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	:	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source: www.LookupTables.com

Logica binaria e Porte logiche

- Variabili binarie
- Operatori logici
- **Operatori logici fondamentali: sono le 3 funzioni logiche**
 - AND, OR e NOT.
- Porte logiche
- Algebra di Boole

Variabili binarie

- **Valori**
 - True/False
 - On/Off
 - Yes/No
 - 1/0
- **Esempi di variabili:**
 - A, B, y, z, o X_1
 - RESET, START_IT, o ADD1

Operatori Logici fondamentali

- AND (\cdot)
- OR (+)
- NOT ($\bar{}$), () o (\sim)

AND

$$0 \cdot 0 = 0$$

$$0 \cdot 1 = 0$$

$$1 \cdot 0 = 0$$

$$1 \cdot 1 = 1$$

OR

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 1$$

NOT

$$\bar{0} = 1$$

$$\bar{1} = 0$$

Tabella di verità

- Elenco dei valori di uscita di una funzione logica per ogni possibile combinazione degli ingressi binari
 - Funzione di N variabili binarie $\rightarrow 2^N$ righe

AND		
X	Y	Z = X · Y
0	0	0
0	1	0
1	0	0
1	1	1

OR		
X	Y	Z = X + Y
0	0	0
0	1	1
1	0	1
1	1	1

NOT	
X	Z = \bar{X}
0	1
1	0

Regole di precedenza

- Precedenza degli operatori fondamentali in ordine decrescente:
 1. Parentesi
 2. NOT
 3. AND
 4. OR
- Esempio: $F = A(B + C)(C + \overline{D})$

$$F = A(B + C)(C + D)$$

$$A \ 1 \quad F = 1(0+1)(1+0)$$

B 0

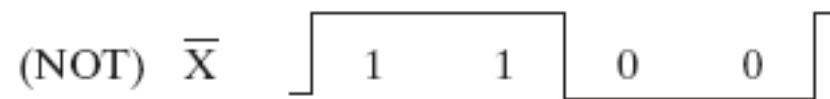
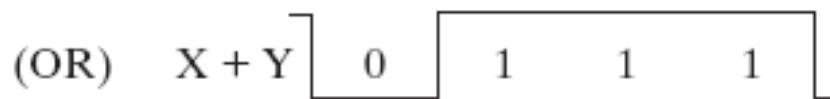
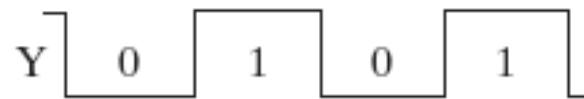
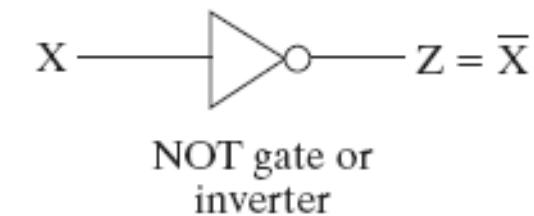
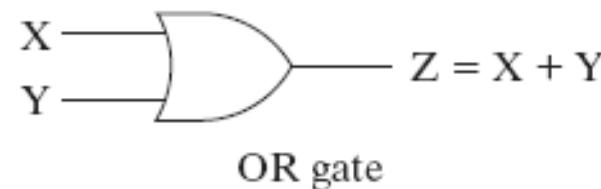
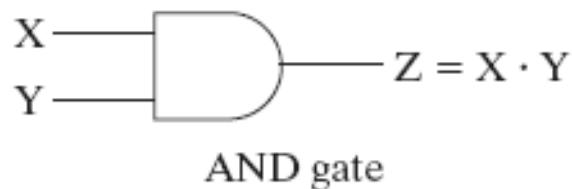
C 1

D 0

$$1(1)(1) = 1$$

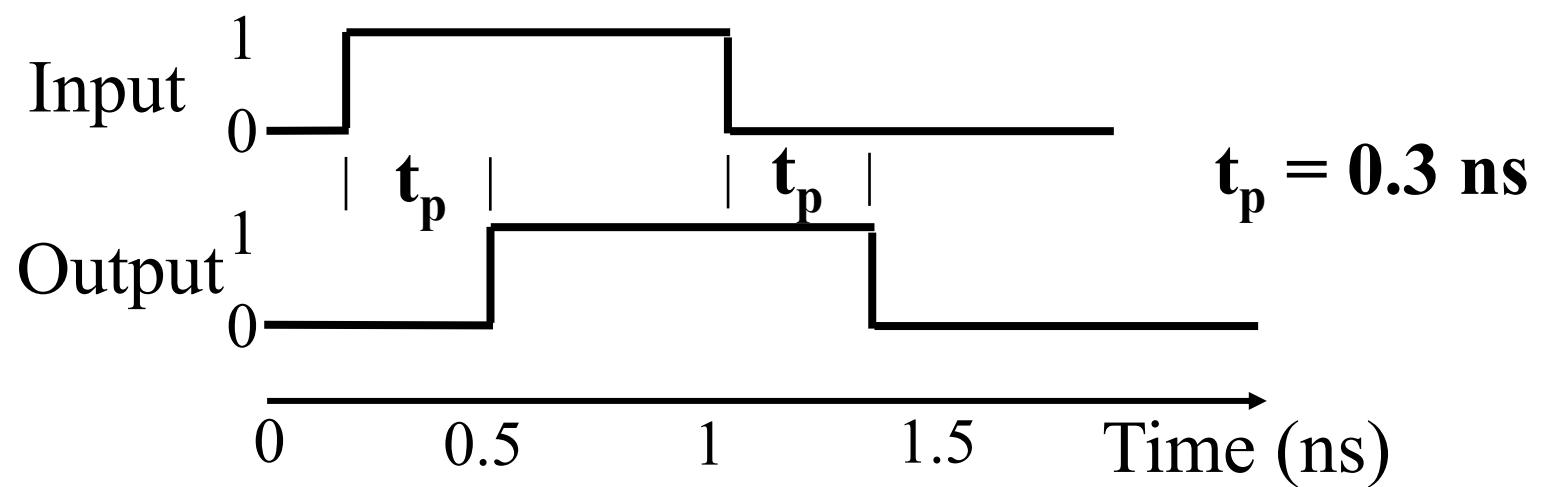
Porte Logiche

- Rappresentazione simbolica del circuito che realizza una funzione logica:



Tempo o ritardo di propagazione

- La commutazione dell'uscita di una porta non può essere istantanea.
- Il ritardo tra la commutazione dell'ingresso(i) e la commutazione dell'uscita è il **tempo o ritardo di propagazione**) t_p :



Circuiti logici

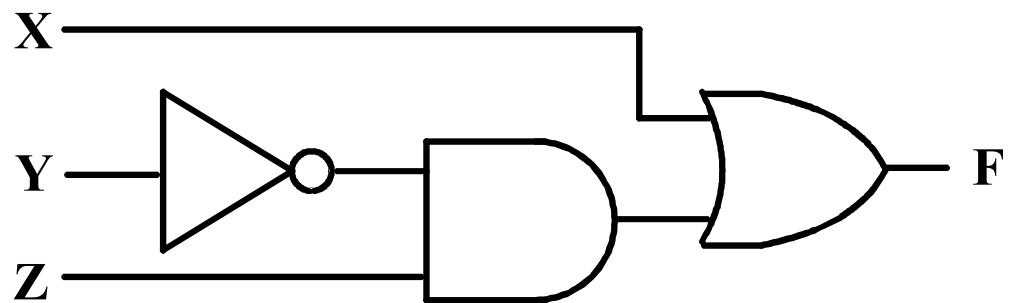
Tabella di verità

X Y Z	$F = X + \bar{Y} \cdot Z$
0 0 0	0
0 0 1	1
0 1 0	0
0 1 1	0
1 0 0	1
1 0 1	1
1 1 0	1
1 1 1	1

Equazione logica

$$F = X + \bar{Y} Z$$

Circuito logico



$$F = \bar{X}\bar{Y}Z + X\bar{Y}\bar{Z} + X\bar{Y}Z + XY\bar{Z} + XYZ$$

Terms of Use

- All (or portions) of this material © 2008 by Pearson Education, Inc.
- Permission is given to incorporate this material or adaptations thereof into classroom presentations and handouts to instructors in courses adopting the latest edition of Logic and Computer Design Fundamentals as the course textbook.
- These materials or adaptations thereof are not to be sold or otherwise offered for consideration.
- This Terms of Use slide or page is to be included within the original materials or any adaptations thereof.

1222-2022
800 ANNI



UNIVERSITÀ
DEGLI STUDI
DI PADOVA



Consapevolezza Digitale

Forme canoniche per le funzioni logiche

Andrea Gerosa – andrea.gerosa@unipd.it

Forme canoniche

- Tra le possibili espressioni logiche di una data funzione, ne identifichiamo 2 particolari:
 - Somma di mintermini – SOM (*Sum of Minterms*)
 - Prodotto di maxtermini – POM (*Product of Maxterms*)

Mintermini - definizioni

- Data una funzione logica a N variabili:
- Termine prodotto: qualsiasi prodotto tra letterali
- Mintermine: termine prodotto che contiene tutte le N variabili (dirette o negate).
- Per ogni funzione di N variabili, esistono 2^N mintermini.

$$\begin{array}{c} XY \\ X\bar{Y} \\ \bar{X}Y \\ \bar{X}\bar{Y} \end{array}$$

Maxtermini - definizioni

- Data una funzione logica a N variabili:
- Termine somma: qualsiasi somma tra letterali
- Maxtermine: termine somma che contiene tutte le N variabili (dirette o negate).
- Per ogni funzione di N variabili, **esistono 2^N maxtermini.**

$$X + Y$$

$$X + \overline{Y}$$

$$\overline{X} + Y$$

$$\overline{X} + \overline{Y}$$

Maxtermini e Mintermini

- Indici

i	Mint., m_i	Maxt., M_i
0	$\bar{x} \bar{y}$	$x + y$
1	$\bar{x} y$	$x + \bar{y}$
2	$x \bar{y}$	$\bar{x} + y$
3	$x y$	$\bar{x} + \bar{y}$

- L'indice i corrisponde alla riga della TdV a cui associamo m_i e M_i

Mintermini e TdV

#	X	Y	Z	Product Term	Symbol
0	0	0	0	$\overline{X}\overline{Y}\overline{Z}$	m_0
1	0	0	1	$\overline{X}\overline{Y}Z$	m_1
2	0	1	0	$\overline{X}YZ$	m_2
3	0	1	1	$\overline{X}Y\overline{Z}$	m_3
4	1	0	0	$X\overline{Y}\overline{Z}$	m_4
5	1	0	1	$X\overline{Y}Z$	m_5
6	1	1	0	$XY\overline{Z}$	m_6
7	1	1	1	XYZ	m_7

- Variabile = 0
 - appare negata nel mint.
- Variabile = 1
 - Appare diretta nel mint.

Mintermini e TdV

X	Y	Z	Product Term	Symbol	m ₀	m ₁	m ₂	m ₃	m ₄	m ₅	m ₆	m ₇
0	0	0	$\overline{X}\overline{Y}\overline{Z}$	m ₀	1	0	0	0	0	0	0	0
0	0	1	$\overline{X}\overline{Y}Z$	m ₁	0	1	0	0	0	0	0	0
0	1	0	$\overline{X}YZ$	m ₂	0	0	1	0	0	0	0	0
0	1	1	$\overline{X}Y\overline{Z}$	m ₃	0	0	0	1	0	0	0	0
1	0	0	$X\overline{Y}\overline{Z}$	m ₄	0	0	0	0	1	0	0	0
1	0	1	$X\overline{Y}Z$	m ₅	0	0	0	0	0	1	0	0
1	1	0	$X\overline{Y}\overline{Z}$	m ₆	0	0	0	0	0	0	1	0
1	1	1	XYZ	m ₇	0	0	0	0	0	0	0	1

- Un solo mint. vale “1” per ogni riga
- Una funzione logica può essere espressa come OR tra mintermini

Somma di Mintermini

- $F_1 = m_1 + m_4 + m_7$
- $F_1 = \overline{x} \ \overline{y} \ z + x \ \overline{y} \ \overline{z} + x \ y \ z$

x y z	index	m_1	+	m_4	+	m_7	= F_1
0 0 0	0	0	+	0	+	0	= 0
0 0 1	1	1	+	0	+	0	= 1
0 1 0	2	0	+	0	+	0	= 0
0 1 1	3	0	+	0	+	0	= 0
1 0 0	4	0	+	1	+	0	= 1
1 0 1	5	0	+	0	+	0	= 0
1 1 0	6	0	+	0	+	0	= 0
1 1 1	7	0	+	0	+	1	= 1

Maxtermi e TdV

X	Y	Z	Sum Term	Symbol
0	0	0	$X + Y + Z$	M ₀
0	0	1	$X + Y + \bar{Z}$	M ₁
0	1	0	$X + \bar{Y} + Z$	M ₂
0	1	1	$X + \bar{Y} + \bar{Z}$	M ₃
1	0	0	$\bar{X} + Y + Z$	M ₄
1	0	1	$\bar{X} + Y + \bar{Z}$	M ₅
1	1	0	$\bar{X} + \bar{Y} + Z$	M ₆
1	1	1	$\bar{X} + \bar{Y} + \bar{Z}$	M ₇

- Variabile = 0
 - appare diretta nel maxt.
- Variabile = 1
 - Appare negata nel maxt.

Maxtermini e TdV

X	Y	Z	Sum Term	Symbol	M ₀	M ₁	M ₂	M ₃	M ₄	M ₅	M ₆	M ₇
0	0	0	$X+Y+Z$	M ₀	0	1	1	1	1	1	1	1
0	0	1	$X+Y+\bar{Z}$	M ₁	1	0	1	1	1	1	1	1
0	1	0	$X+\bar{Y}+Z$	M ₂	1	1	0	1	1	1	1	1
0	1	1	$X+\bar{Y}+\bar{Z}$	M ₃	1	1	1	0	1	1	1	1
1	0	0	$\bar{X}+Y+Z$	M ₄	1	1	1	1	0	1	1	1
1	0	1	$\bar{X}+Y+\bar{Z}$	M ₅	1	1	1	1	1	0	1	1
1	1	0	$\bar{X}+\bar{Y}+Z$	M ₆	1	1	1	1	1	1	0	1
1	1	1	$\bar{X}+\bar{Y}+\bar{Z}$	M ₇	1	1	1	1	1	1	1	0

- Un solo maxt. vale “0” per ogni riga
- Una funzione logica può essere espressa come AND tra maxtermini

Prodotto di Maxtermini

- $F_1 = M_0 \cdot M_2 \cdot M_3 \cdot M_5 \cdot M_6$

$$F_1 = (x + y + z) \cdot (x + \bar{y} + z) \cdot (x + \bar{y} + \bar{z}) \\ \cdot (\bar{x} + y + \bar{z}) \cdot (\bar{x} + \bar{y} + z)$$

x y z	i	$M_0 \cdot M_2 \cdot M_3 \cdot M_5 \cdot M_6 = F_1$
0 0 0	0	0 · 1 · 1 · 1 · 1 = 0
0 0 1	1	1 · 1 · 1 · 1 · 1 = 1
0 1 0	2	1 · 0 · 1 · 1 · 1 = 0
0 1 1	3	1 · 1 · 0 · 1 · 1 = 0
1 0 0	4	1 · 1 · 1 · 1 · 1 = 1
1 0 1	5	1 · 1 · 1 · 1 · 0 = 0
1 1 0	6	1 · 1 · 1 · 1 · 1 = 1
1 1 1	7	1 · 1 · 1 · 1 · 1 = 1

Somma canonica

- Qualsiasi funzione logica può essere espressa come Somma di Mintermini.

$$f = x + \bar{x} \bar{y}$$

$$f = x(y + \bar{y}) + \bar{x} \bar{y}$$

$$f = xy + x\bar{y} + \bar{x}\bar{y}$$

Abbreviazioni

$$F = m_1 + m_4 + m_5 + m_6 + m_7$$

$$F(A, B, C) = \Sigma_m(1, 4, 5, 6, 7)$$

$$F(A, B, C) : On-set = \{m_1, m_4, m_5, m_6, m_7\}$$

Prodotto canonico

- Qualsiasi funzione logica può essere espressa come Prodotto di Maxtermini.

$$f(x, y, z) = x + \bar{x} \bar{y}$$

$$x + \bar{x} \bar{y} = (x + \bar{x})(x + \bar{y}) = 1 \cdot (x + \bar{y}) = x + \bar{y}$$

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

Altre forme standard

- Somma di prodotti (SOP)
- Prodotto di somme (POS)

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

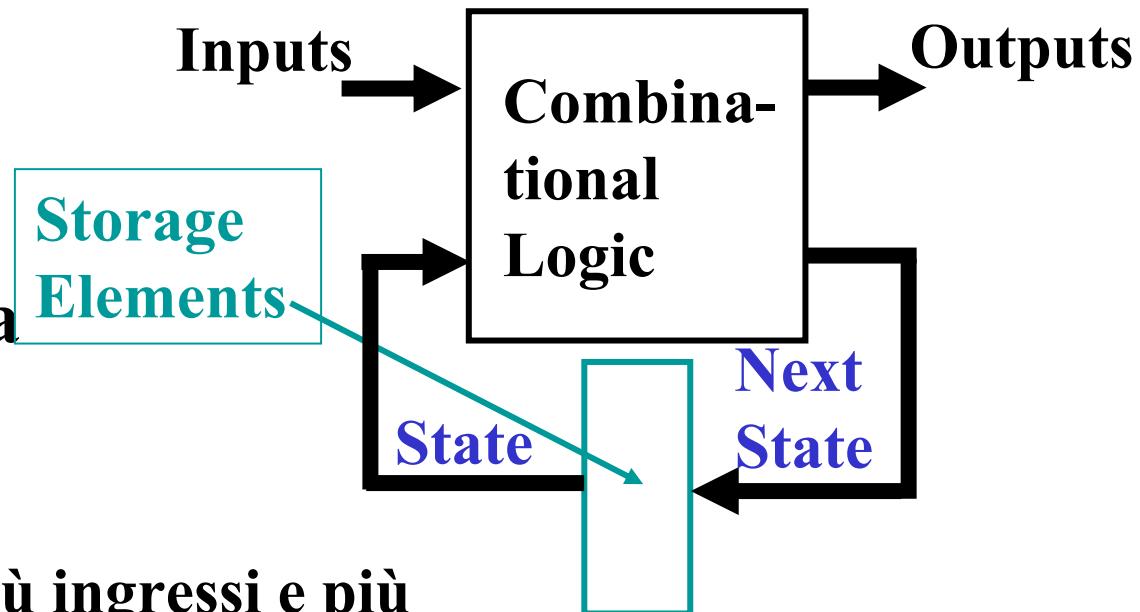
$$(A + B) \cdot (A + \overline{B} + \overline{C}) \cdot C$$

Terms of Use

- All (or portions) of this material © 2008 by Pearson Education, Inc.
- Permission is given to incorporate this material or adaptations thereof into classroom presentations and handouts to instructors in courses adopting the latest edition of Logic and Computer Design Fundamentals as the course textbook.
- These materials or adaptations thereof are not to be sold or otherwise offered for consideration.
- This Terms of Use slide or page is to be included within the original materials or any adaptations thereof.

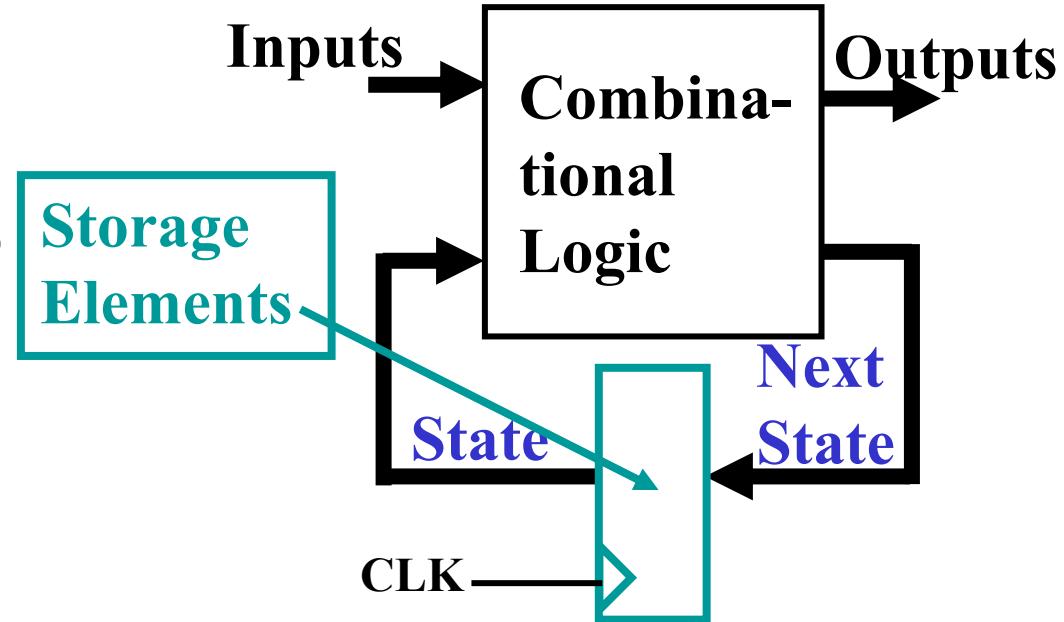
Circuiti Sequenziali

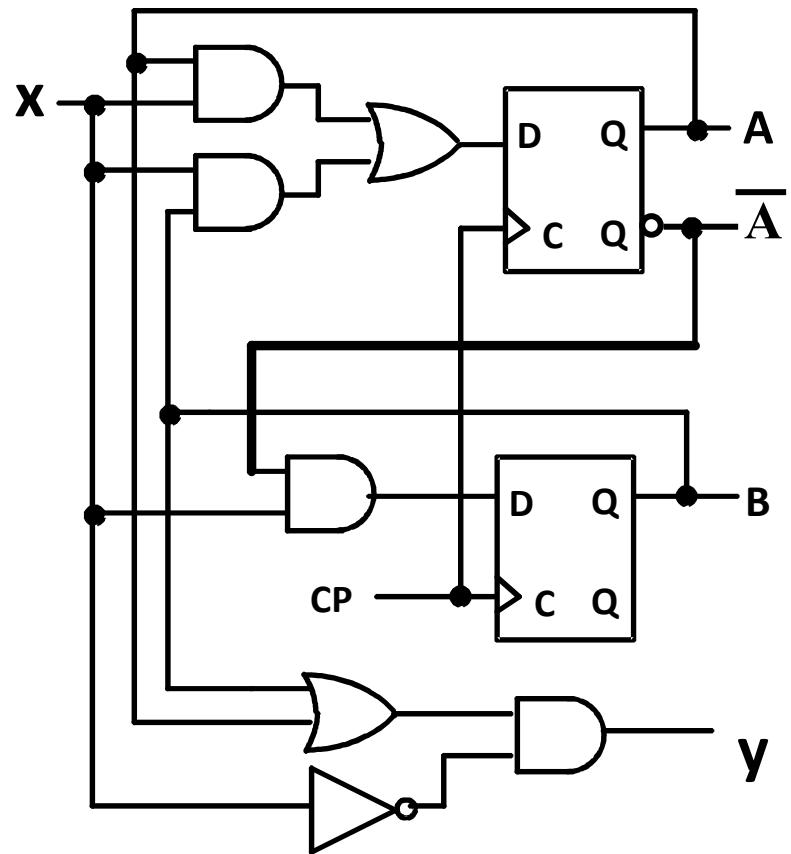
- Elementi di memoria
 - Stato del sistema
- Logic combinatoria
 - Funzioni logiche a più ingressi e più uscite
 - Ingressi: segnali dall'esterno
 - Uscite segnali verso l'esterno
 - Altri ingressi: Stato Presente
 - Altre uscite: Stato Futuro



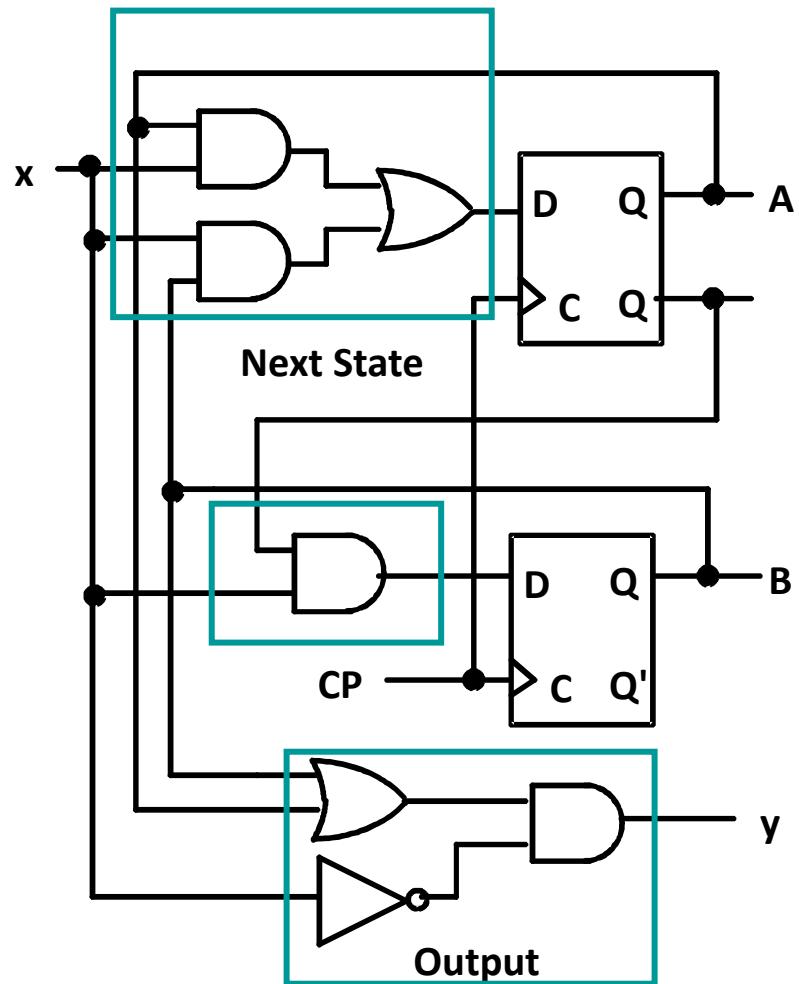
Circuiti sequenziali sincroni

- Lo stato corrente al tempo t è memorizzato nei registri
- Lo stato futuro al tempo $t+1$ è una funzione logica dello stato presente e dell'ingresso al tempo t
- Le uscite al tempo t sono funzioni logiche dello stato al tempo t e degli ingressi al tempo t .





- $A(t+1) = A(t)x(t)$
+ $B(t)x(t)$
- $B(t+1) = \bar{A}(t)x(t)$
- $y(t) = \bar{x}(t)(B(t) + A(t))$



RESET

CK

x

A

B

$$A^* = A \cdot x + B \cdot \bar{x}$$

$$B^* = \bar{A} \cdot x$$

$$y = \bar{x} \cdot (A + B)$$

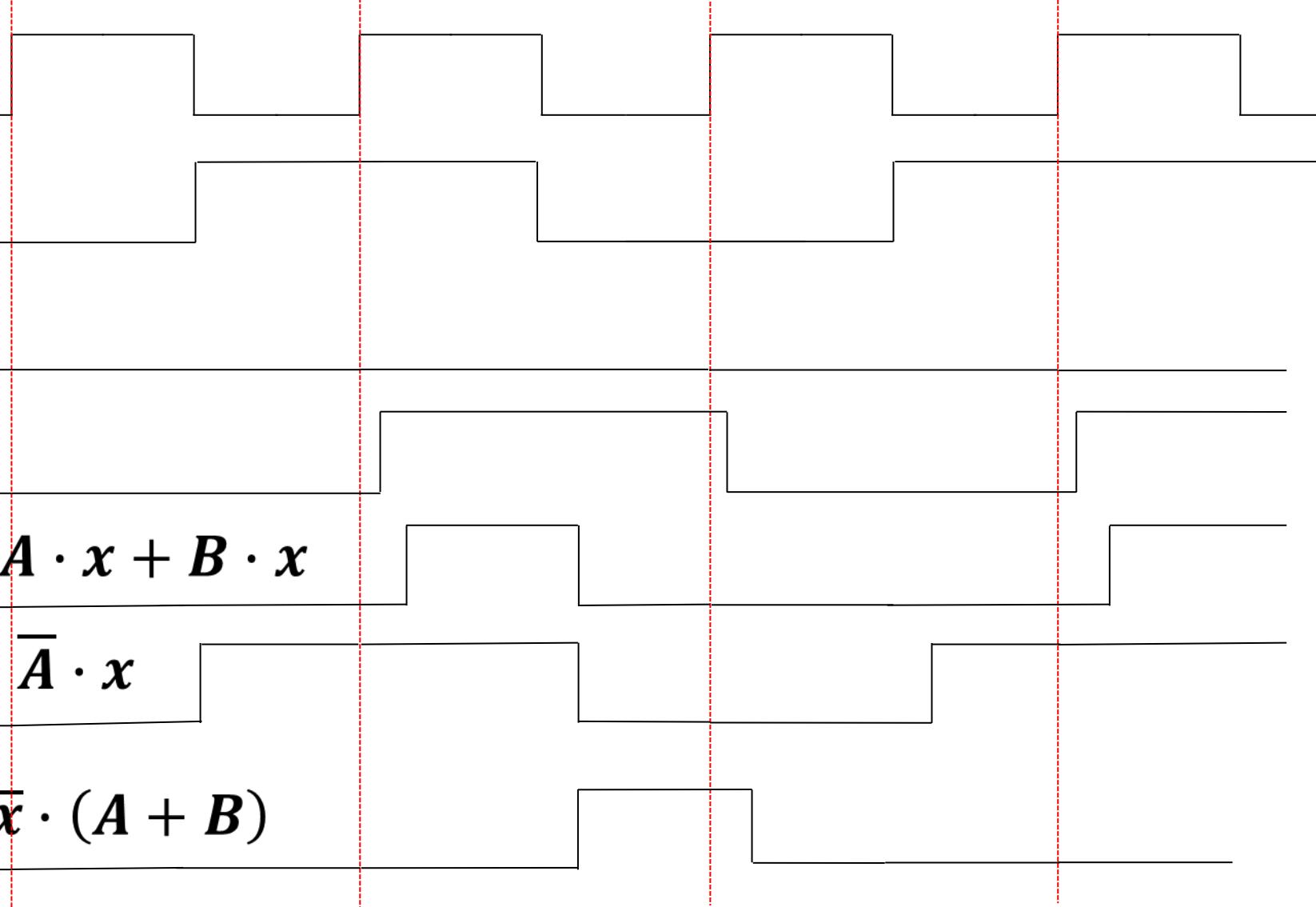


Tabella di Stato

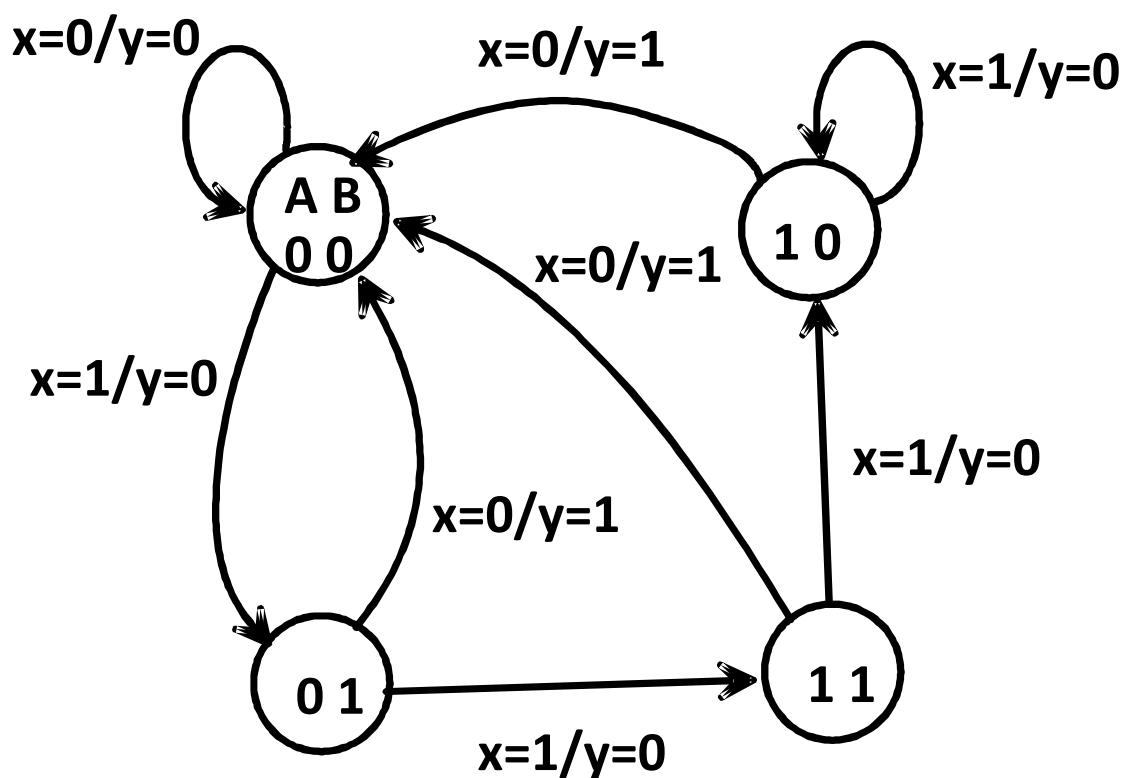
- *Tabella a più ingressi:*
 - *Stato presente*
 - *Ingressi presenti*
- *Tabella a più uscite*
 - *Stato futuro*
 - *Uscite*

- $A(t+1) = A(t)x(t) + B(t)x(t)$
- $B(t+1) = \bar{A}(t)x(t)$
- $y(t) = \bar{x}(t)(B(t) + A(t))$

Present State $A(t)$ $B(t)$	Next State		Output	
	$x(t)=0$	$x(t)=1$	$x(t)=0$	$x(t)=1$
0 0	0 0	0 1	0	0
0 1	0 0	1 1	1	0
1 0	0 0	1 0	1	0
1 1	0 0	1 0	1	0

Diagrammi di stato

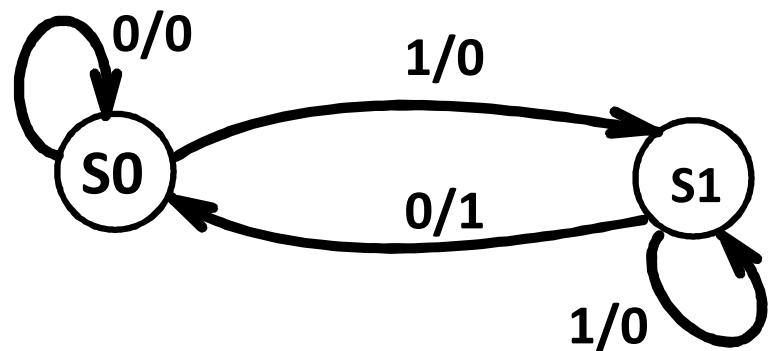
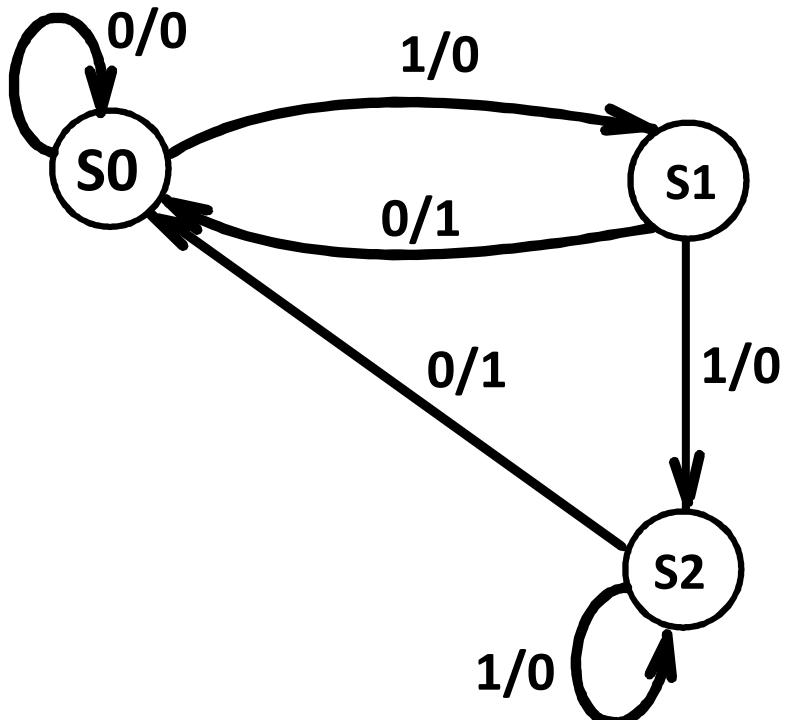
- **Rappresentazione grafica dell'evoluzione dello stato**
 - **Nodo per ogni stato**
 - **Arco direzionale per ogni possibile evoluzione dello stato**
 - **Un etichetta per ogni arco per indicare la configurazione degli ingressi corrispondente alla transizione di stato e per indicare il corrispondente valore dell'uscita**



Stati Equivalenti

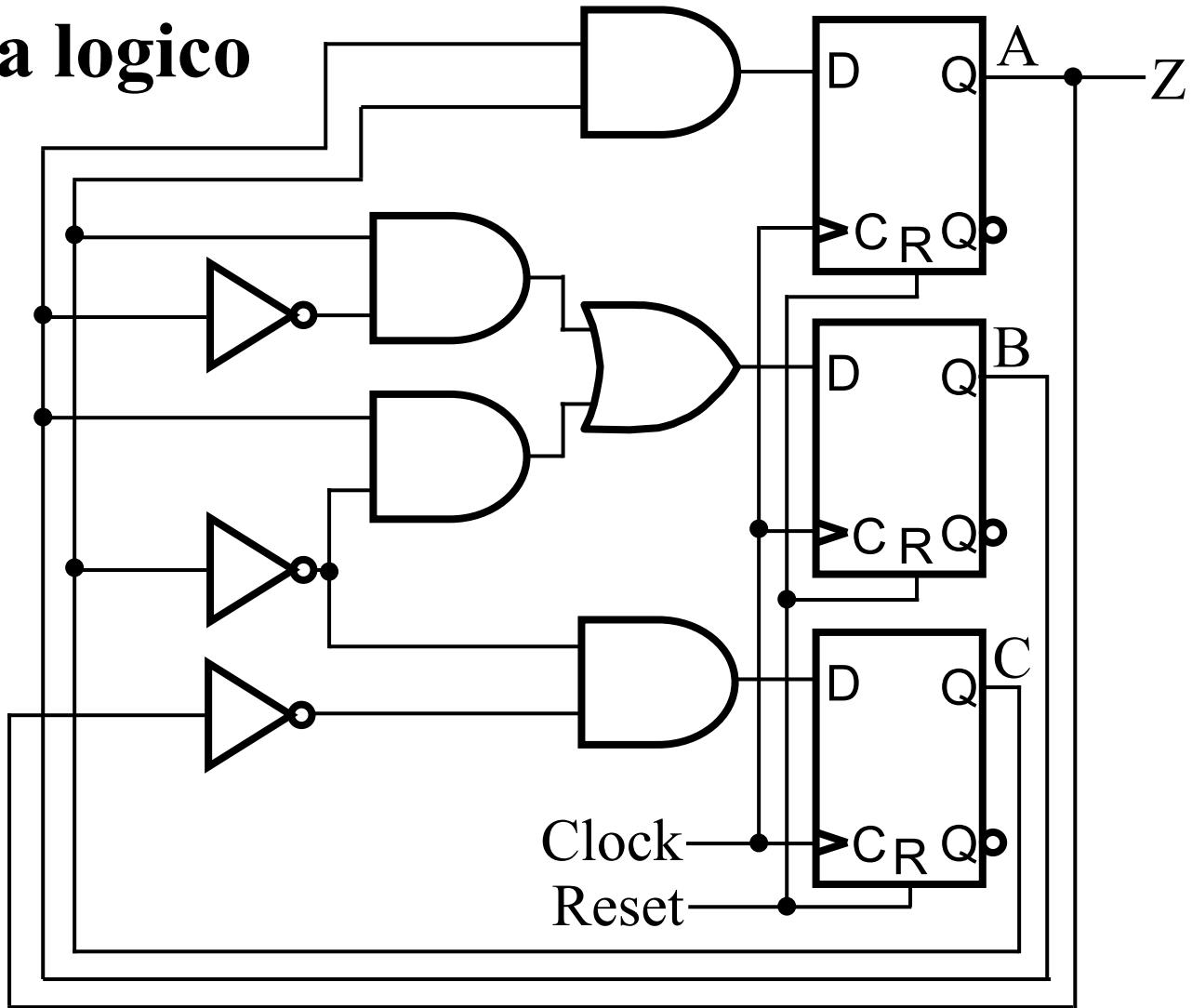
- Due stati sono detti *equivalenti* se per ogni possibile sequenza di ingresso:
 - le corrispondenti evoluzioni dello stato sono equivalenti
 - le corrispondenti sequenze di uscita sono identiche

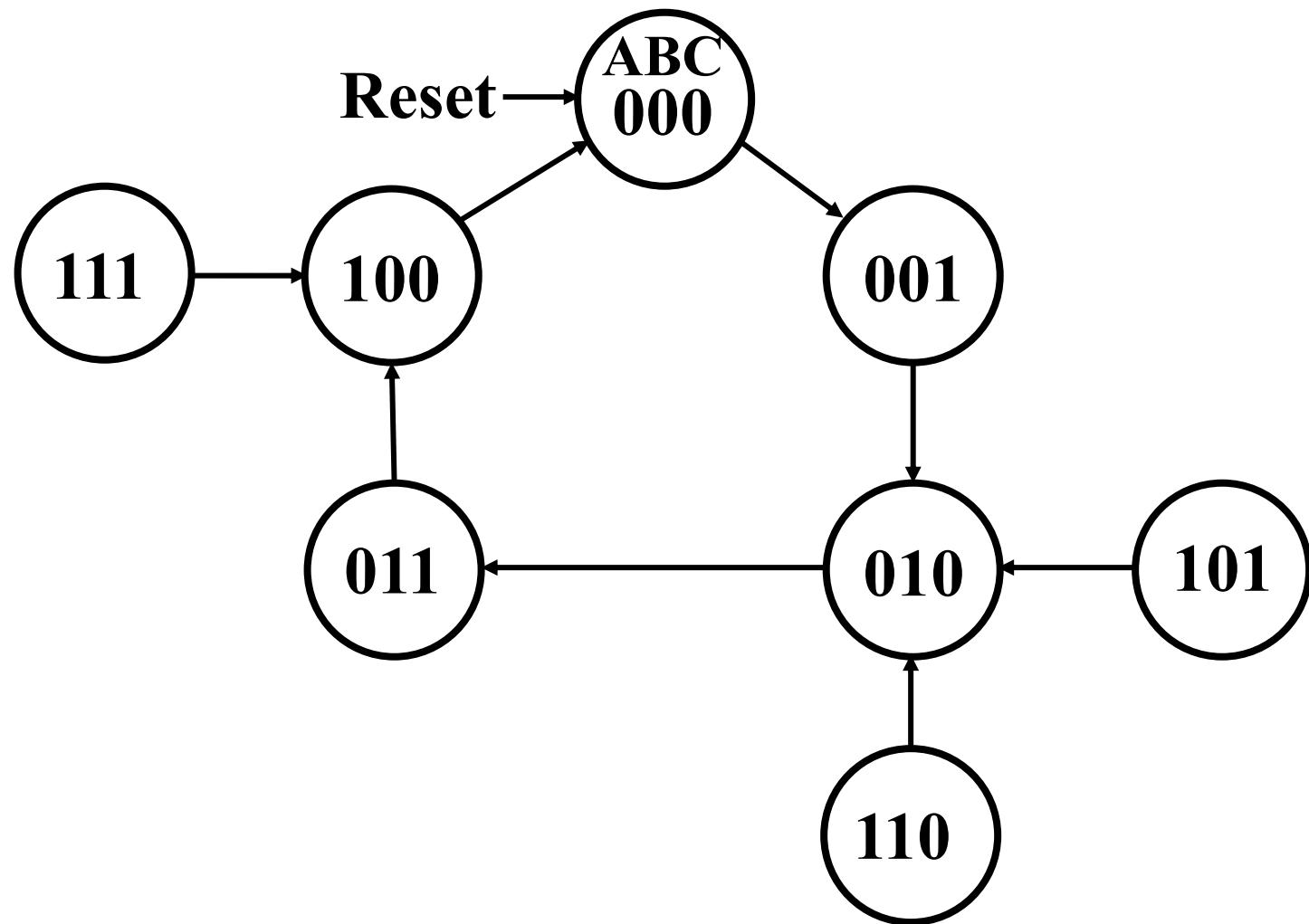
Semplificazione degli stati



Esempio 2

- Diagramma logico





Terms of Use

- All (or portions) of this material © 2008 by Pearson Education, Inc.
- Permission is given to incorporate this material or adaptations thereof into classroom presentations and handouts to instructors in courses adopting the latest edition of Logic and Computer Design Fundamentals as the course textbook.
- These materials or adaptations thereof are not to be sold or otherwise offered for consideration.
- This Terms of Use slide or page is to be included within the original materials or any adaptations thereof.



DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

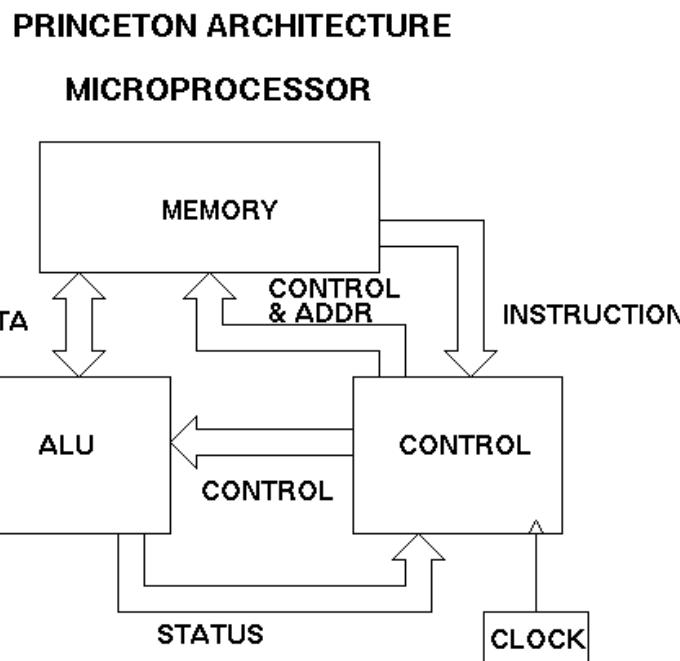
Architettura di una memoria RAM

corso di Consapevolezza Digitale

Andrea Gerosa

gerosa@dei.unipd.it – 049.827.7728

Un semplice microprocessore



- Unità di controllo
 - elabora le istruzioni macchina
 - abilita gli altri blocchi e funzioni
- ALU
 - elaborazione aritmetica
- Memoria
- BUS
 - unidirezionali o bidirezionali



Struttura della memoria RAM

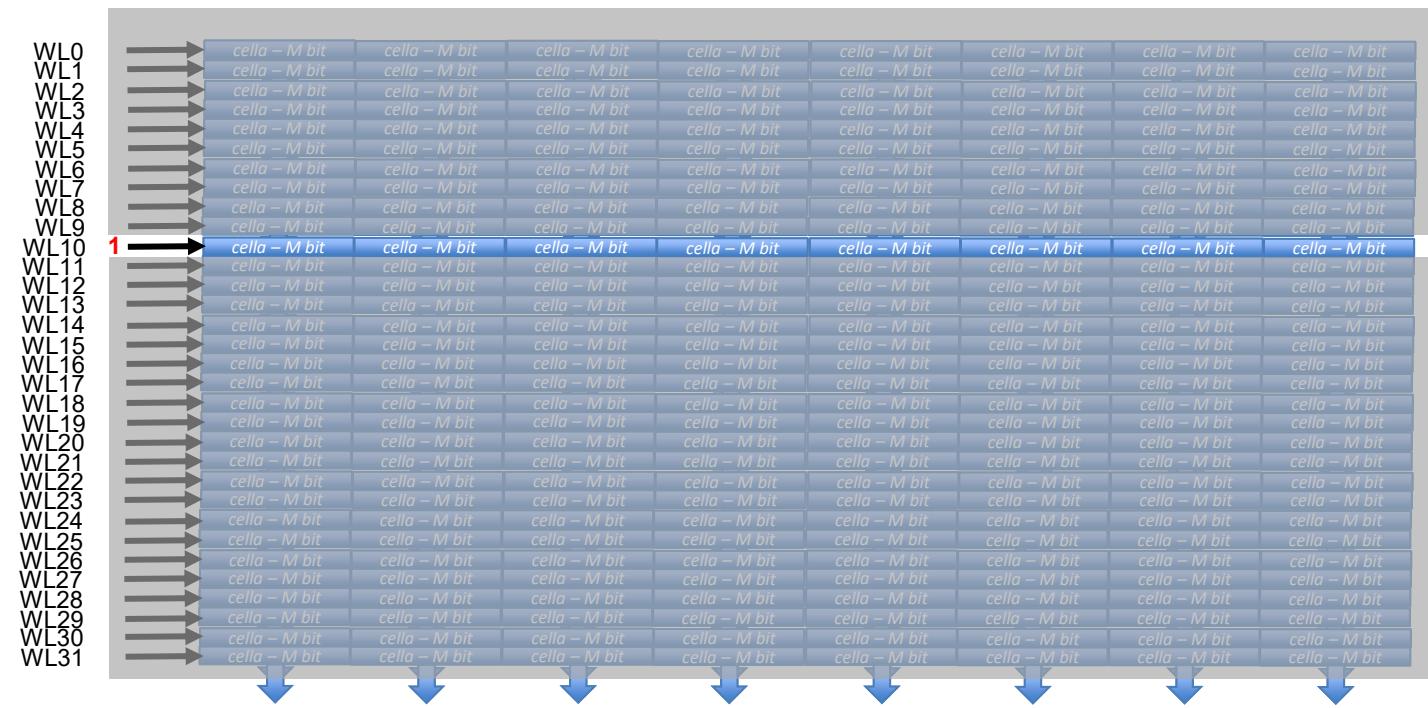
8 colonne (x 8 bit)

$$8 \times 32 = 256 \text{ byte} = 2 \text{kbit}$$

- La RAM è formata da celle di memoria che memorizzano un byte (tipicamente 8 bit) o word
 - Le celle sono organizzate in una matrice

Indirizzamento della memoria

- Passo 1
 - individuo quale riga voglio usare
 - ogni riga ha un proprio segnale di attivazione (WL)



Indirizzamento della memoria

□ Passo 2

- individuo quale colonna va mandata in uscita
- ogni colonna ha un proprio segnale di attivazione (BL)

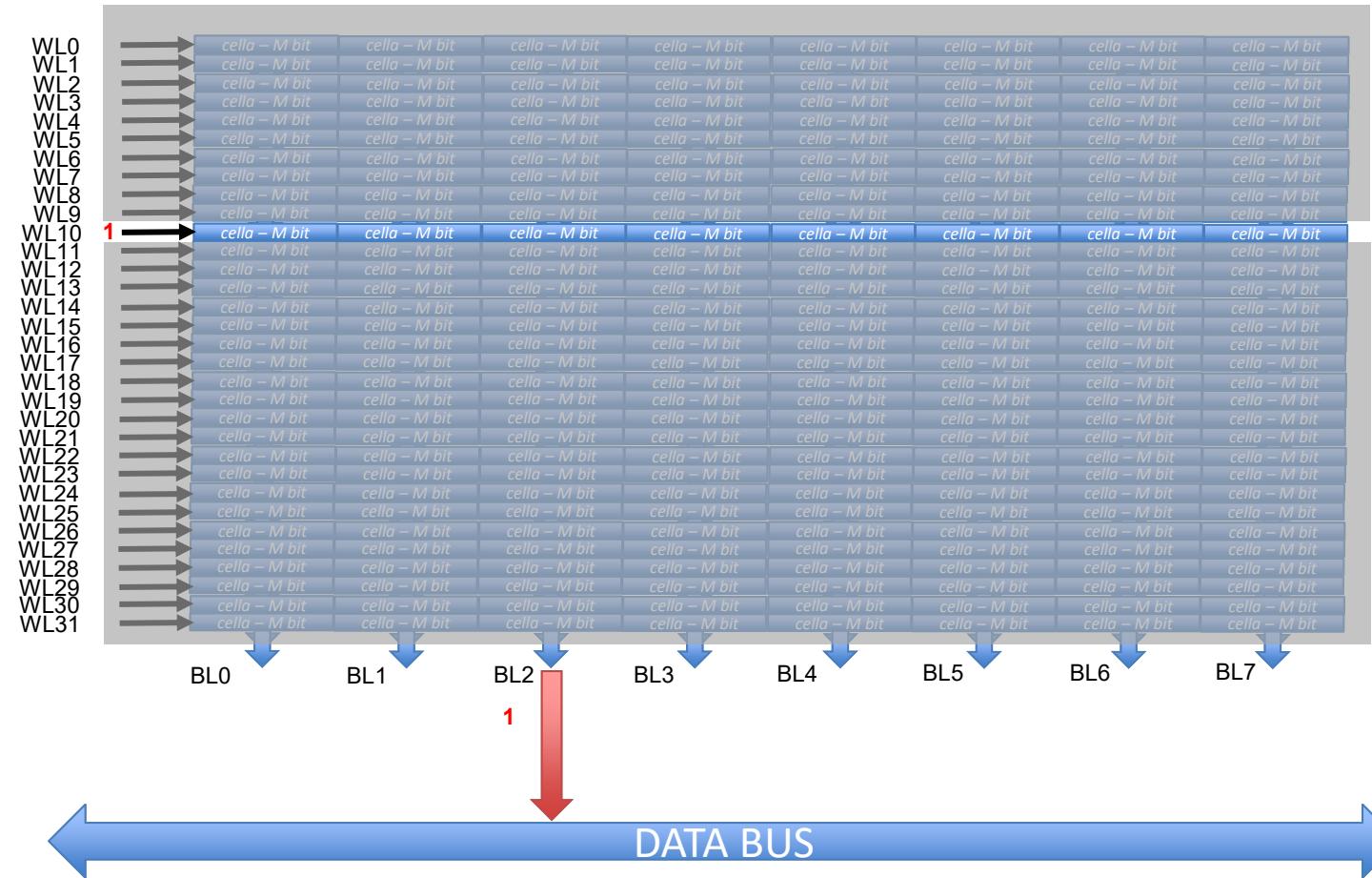
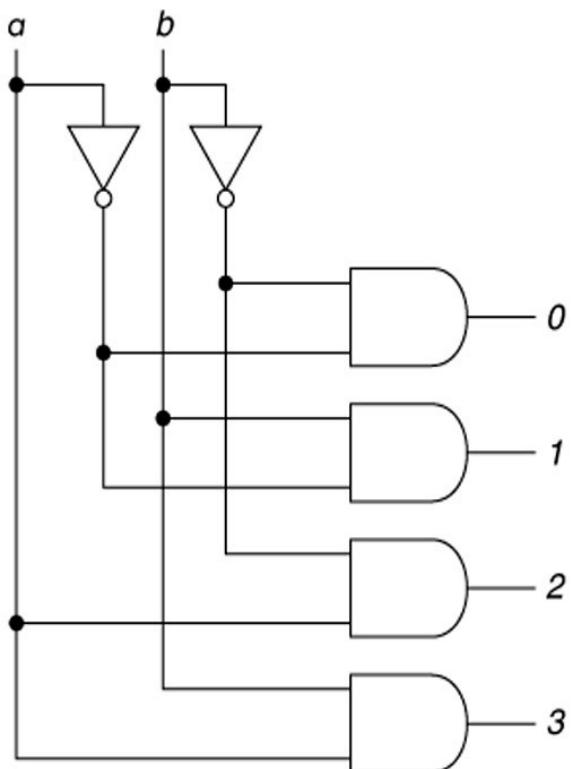


Table 5.2aAn active
high decoder.

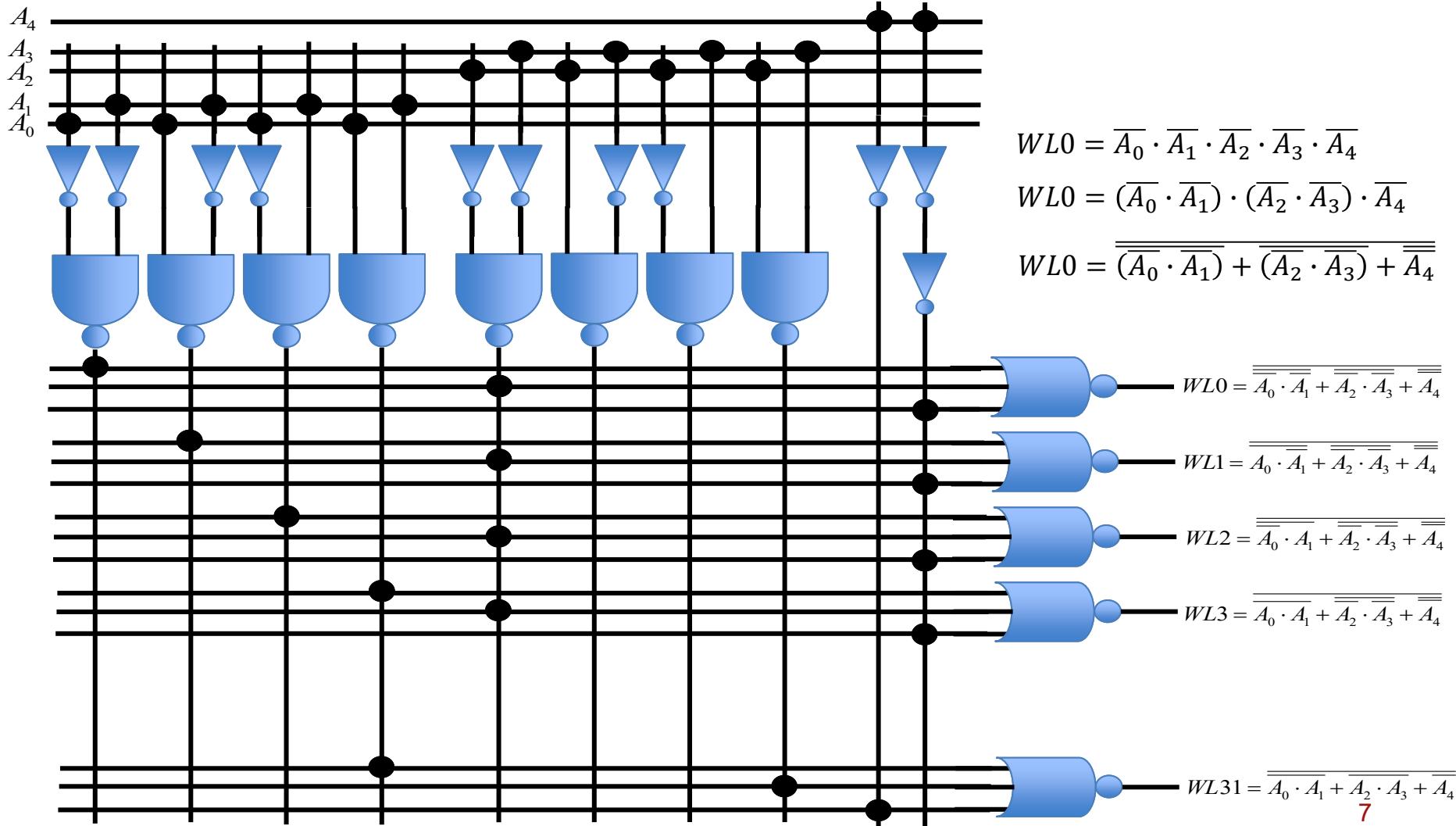
<i>a</i>	<i>b</i>	0	1	2	3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



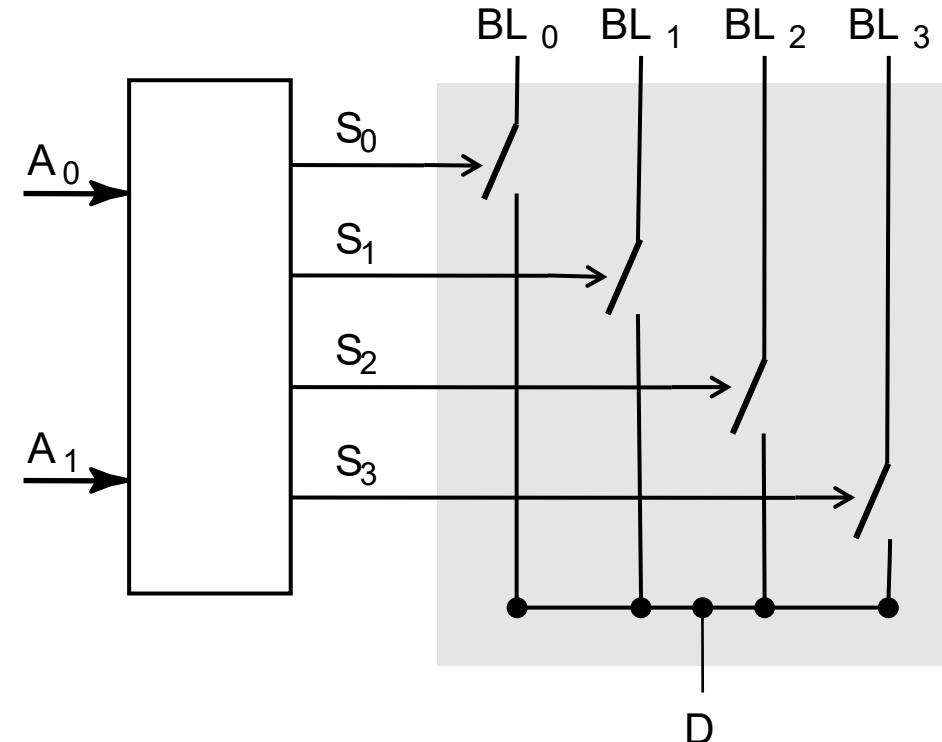
Decoder di riga

- Uso un numero binario a N bit per identificare quale WL deve essere attivata
 - posso distinguere tra 2^N WL
- Decoder binario
- 32 WL \Rightarrow 5 ingressi
- 16k WL \Rightarrow 14 ingressi!

Decoder multistadio

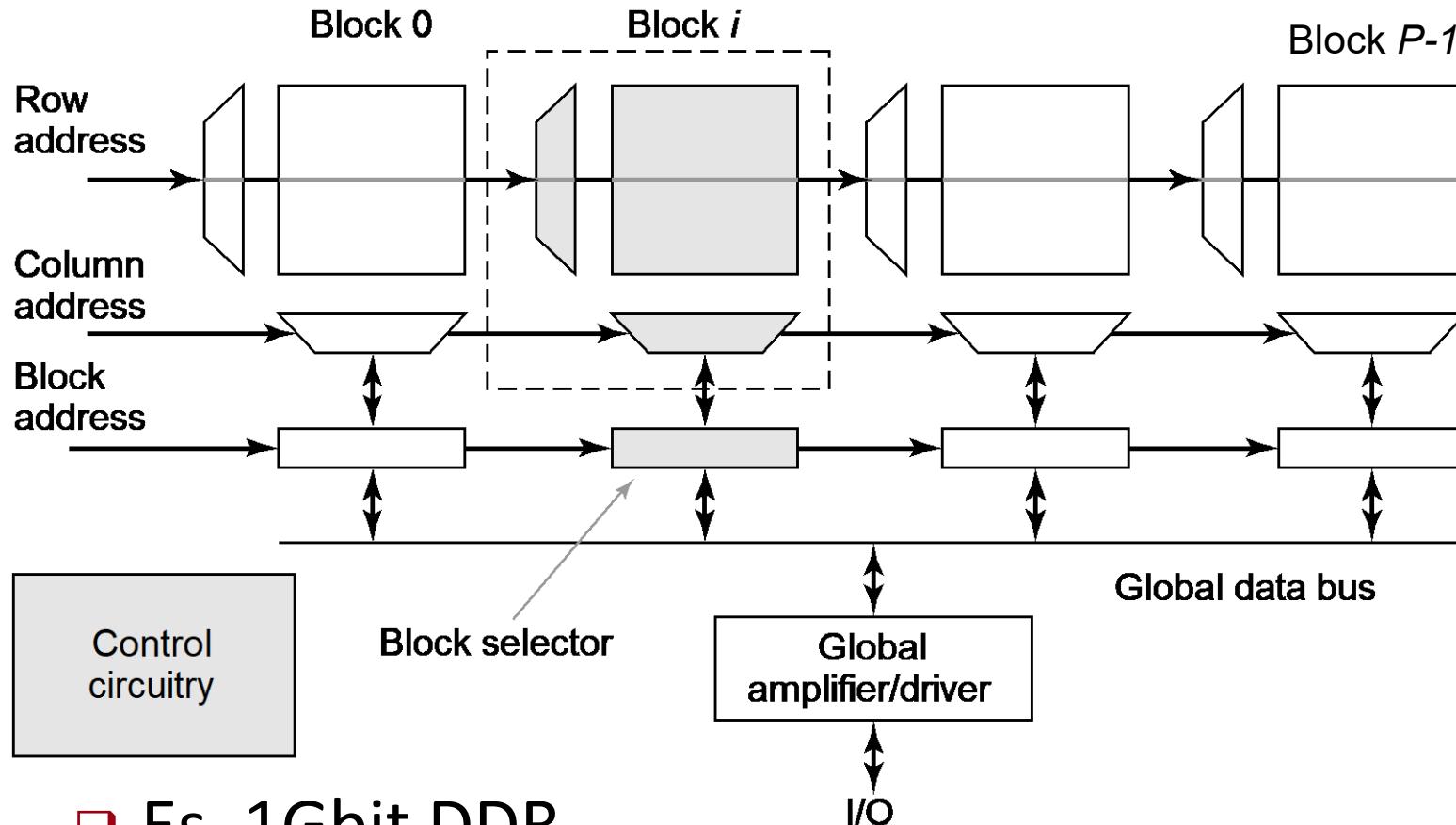


Decoder di colonna



- Uso un numero binario a K bit per identificare quale BL deve essere collegata in uscita
 - posso distinguere tra 2^K BL
- Decoder + Multiplexer
- 8 BL \Rightarrow 192 interruttori
- 2e3 BL \Rightarrow 61.400 interruttori

Struttura gerarchica

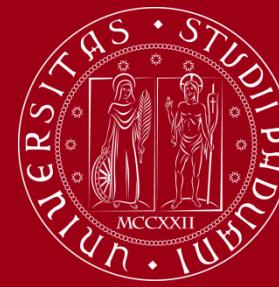


□ Es. 1Gbit DDR

- 8 moduli (128MB)
- $128\text{MB} = 2^{14} \text{WL} \times 2^{11} \text{BL} \times 2^2 \text{Blocchi}$



DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

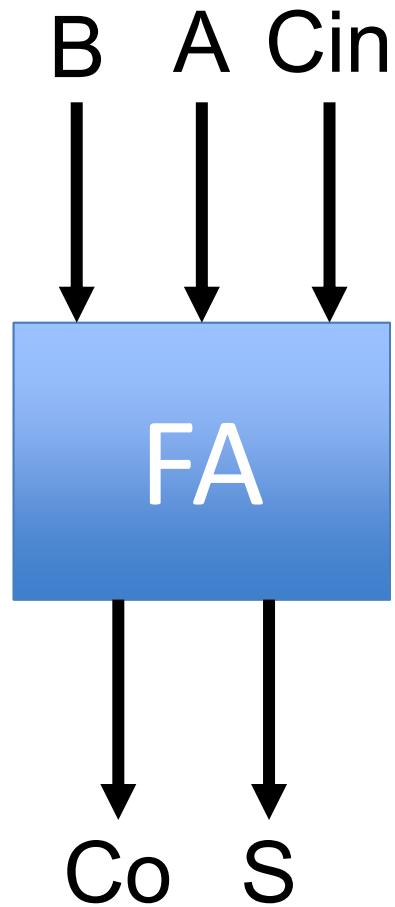
Sommatori

corso di Consapevolezza digitale

Andrea Gerosa

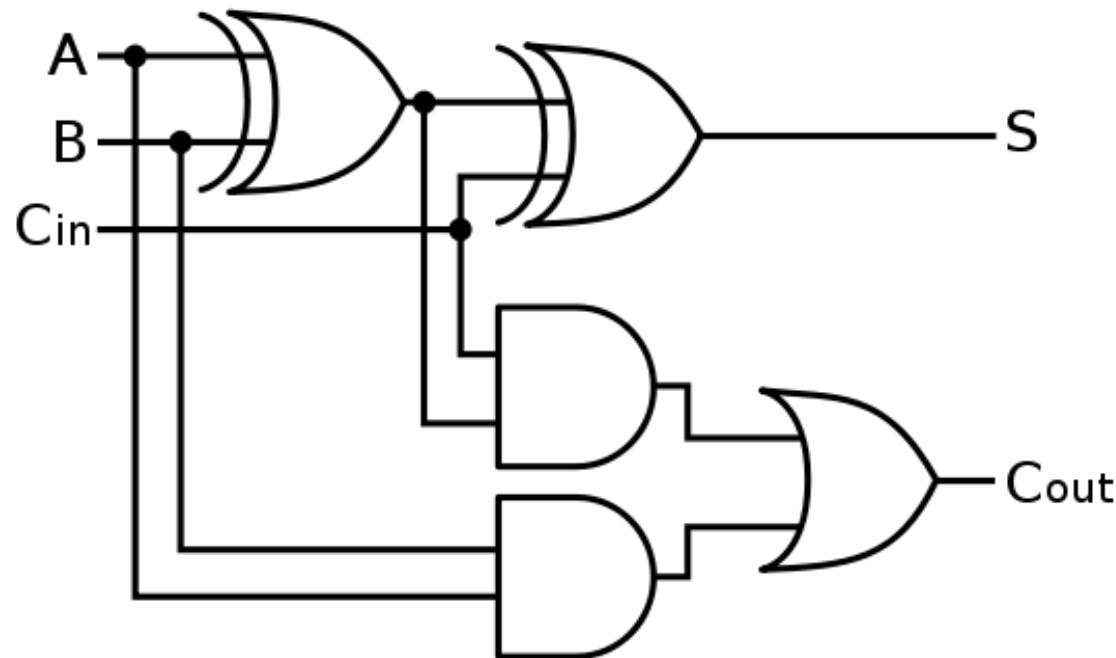
gerosa@dei.unipd.it – 049.827.7728

Full Adder



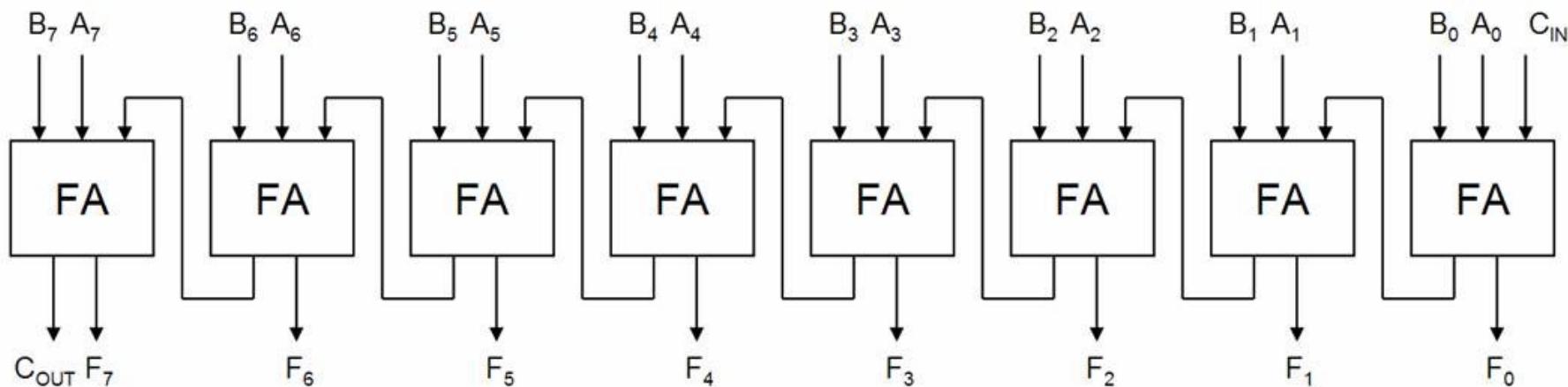
A	B	Ci	S	Co
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Full adder



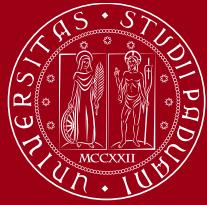
- Porte logiche fondamentali

Sommatore



- 8 bit ripple-carry adder
 - 8 blocchi full-adder in cascata

8¹²²²⁻²⁰²²
000 ANNI



UNIVERSITÀ
DEGLI STUDI
DI PADOVA



Consapevolezza Digitale

Sistemi di numerazione e operazioni aritmetiche con i
numeri binari

Andrea Gerosa – andrea.gerosa@unipd.it

SISTEMI DI NUMERAZIONE

- **Sistemi posizionali**

- La posizione della cifra ha un significato specifico

- **Esempio: 326,5**

- $$\bullet 326,5 = 3 * 100 + 2 * 10 + 6 * 1 + 5 * 0,1$$

- $$= 3 * 10^2 + 2 * 10^1 + 6 * 10^0 + 5 * 10^{-1}$$

- **Radice o base r (r=10 nel nostro esempio)**

- **Il numero è una serie (somma) di potenze della radice**

- Le cifre sono un coefficiente moltiplicativo delle potenze

Conversione da binario a decimale

- $11010_2 \rightarrow N_{10}$:

$$0 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 + 1 \cdot 2^4$$

$$0 \cdot 1 + 1 \cdot 2 + 0 \cdot 4 + 1 \cdot 8 + 1 \cdot 16 = 26_{10}$$

Conversione da decimale a binario

- $26_{10} = 1 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 0 \cdot 1$
- $13_{10} = (1 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 0 \cdot 1) / 2$
- $13_{10} = 1 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1$ resto = 0
- $(13_{10}) / 2 = (1 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1) / 2$
- $= 1 \cdot 4 + 1 \cdot 2 + 0 \cdot 1 = 6_{10}$ resto = 1

$$26 / 2 = 13 \text{ resto } 0 \rightarrow d_0 = 0$$

$$13 / 2 = 6 \text{ resto } 1 \rightarrow d_1 = 1$$

$$6 / 2 = 3 \text{ resto } 0 \rightarrow d_2 = 0$$

$$3 / 2 = 1 \text{ resto } 1 \rightarrow d_3 = 1$$

$$1 / 2 = 0 \text{ resto } 1 \rightarrow d_4 = 1$$

Basi

Nome	Radice	Cifre
Binaria	2	0,1
Ottale	8	0,1,2,3,4,5,6,7
Decimale	10	0,1,2,3,4,5,6,7,8,9
Esadecimale	16	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Numeri in basi diverse

Decimal (Base 10)	Binary (Base 2)	Octal (Base 8)	Hexadecimal (Base 16)
00	00000	00	00
01	00001	01	01
02	00010	02	02
03	00011	03	03
04	00100	04	04
05	00101	05	05
06	00110	06	06
07	00111	07	07
08	01000	10	08
09	01001	11	09
10	01010	12	0A
11	01011	13	0B
12	01100	14	0C
13	01101	15	0D
14	01110	16	0E
15	01111	17	0F
16	10000	20	10

Da ottale (esadecimale) a binario e viceversa

- Sostituire a ogni cifra il corrispondente valore in base 2, usando 3 (4) bit per cifra:

- $14_8 \rightarrow 001\ 100_2$

$$14_{16} \rightarrow 0001\ 0100_2$$

- Raggruppare i bit a gruppi di 3 (4)
- Sostituire ogni bit con il codice ottale (esadecimale)

- $001\ 100_2 \rightarrow 14_8$

$$0000\ 1100_2 \rightarrow 0C_{16}$$

Addizione di bit singoli: riporti

$$\begin{array}{r} X \quad 0 \quad 0 \quad 1 \quad 1 \\ + Y \quad + 0 \quad + 1 \quad + 0 \quad + 1 \\ \hline CS \quad 0 0 \quad 0 1 \quad 0 1 \quad 1 0 \end{array}$$

$$\begin{array}{r} Z \quad 1 \quad 1 \quad 1 \quad 1 \\ X \quad 0 \quad 0 \quad 1 \quad 1 \\ + Y \quad + 0 \quad + 1 \quad + 0 \quad + 1 \\ \hline CS \quad 0 1 \quad 1 0 \quad 1 0 \quad 1 1 \end{array}$$

Addizione tra numeri

$$\begin{array}{r} \underline{0} & \textcolor{red}{11} & \underline{0} \\ 01100 & 10110 \\ +10001 & +10111 \\ \hline 11101 & \textcolor{red}{101101} \end{array}$$

Sottrazione a singolo bit

Z	0	0	0	0
X	0	0	1	1
<u>-Y</u>	<u>-0</u>	<u>-1</u>	<u>-0</u>	<u>-1</u>
BS	0 0	1 1	0 1	0 0
Z	1	1	1	1
X	0	0	1	1
<u>-Y</u>	<u>-0</u>	<u>-1</u>	<u>-0</u>	<u>-1</u>
BS	11	1 0	0 0	1 1

Moltiplicazione binaria

$$0 * 0 = 0 \quad | \quad 1 * 0 = 0 \quad | \quad 0 * 1 = 0 \quad | \quad 1 * 1 = 1$$

$$\begin{array}{r} 1011 \\ \times 101 \\ \hline 1011 \\ 0000 - \\ \hline 1011 - - \\ \hline 110111 \end{array}$$

Terms of Use

- All (or portions) of this material © 2008 by Pearson Education, Inc.
- Permission is given to incorporate this material or adaptations thereof into classroom presentations and handouts to instructors in courses adopting the latest edition of Logic and Computer Design Fundamentals as the course textbook.
- These materials or adaptations thereof are not to be sold or otherwise offered for consideration.
- This Terms of Use slide or page is to be included within the original materials or any adaptations thereof.