

Note sull' architettura del calcolatore

Informatica

- Scienza che studia la *rappresentazione* e l' *elaborazione* dell' informazione

Algoritmo

- **Problema**

- Preparazione di una torta
- Programmazione del VCR
- MCD tra due numeri

- **Algoritmo**

- **Sequenza ordinata e finita** di **passi (istruzioni)** che risolve il problema specifico
- PASSO 1: Fai qualcosa
- PASSO 2: Fai qualcosa
- ...
- PASSO n: Fermati, hai finito

Problema: consumo medio

PASSO 1: **ACQUISISCI** i valori per *litri utilizzati*, *km alla partenza* e *km all'arrivo*

PASSO 2: **PONI** il valore di *distanza percorsa* a (*km all'arrivo* – *km alla partenza*)

PASSO 3: **PONI** il valore di *km al litro* a (*distanza percorsa* / *litri utilizzati*)

PASSO 4: **STAMPA** il valore di *km al litro*

PASSO 5: **FERMATI**

Problema: $ax^2+bx+c=0$

- PASSO 1: **ACQUISISCI** i valori per a , b e c
PASSO 2: **PONI** il valore di Δ a b^2-4ac
PASSO 3: **SE** il valore di Δ è minore di zero allora
PASSO 4: **STAMPA** "Non esistono soluzioni reali"
ALTRIMENTI
PASSO 5: **SE** il valore di Δ è uguale a zero allora
PASSO 6: **STAMPA** "Due soluzioni reali coincidenti"
PASSO 7: **PONI** il valore di x a $(-b/2a)$
PASSO 8: **STAMPA** il valore di x
ALTRIMENTI
PASSO 9: **STAMPA** "Due soluzioni reali distinte"
PASSO 10: **PONI** il valore di x_1 a $(-b-\sqrt{\Delta})/2a$
PASSO 11: **PONI** il valore di x_2 a $(-b+\sqrt{\Delta})/2a$
PASSO 12: **STAMPA** il valore di x_1 e di x_2
PASSO 13: **FERMATI**

23 December 2015

FixL

5

Problema: n è primo?

- PASSO 1: **ACQUISISCI** il valore per n
PASSO 2: **PONI** il valore di $primo$ a 1
PASSO 3: **PONI** il valore di d a 2
PASSO 4: **RIPETI** i passi da 5 a 7 fintanto che $primo$ è uguale a 1 e d è minore di n
PASSO 5: **SE** n è divisibile per d
PASSO 6: **PONI** il valore di $primo$ a 0
ALTRIMENTI
PASSO 7: **PONI** il valore di d a $(d+1)$
PASSO 8: **SE** il valore di $primo$ è uguale a 1
PASSO 9: **STAMPA** "È primo!"
ALTRIMENTI
PASSO 10: **STAMPA** "Non è primo!"
PASSO 11: **FERMATI**

23 December 2015

FixL

6

Istruzioni

- Istruzioni di elaborazione
 - PONI il valore della *variabile* a *espressione aritmetica*
- Istruzioni di trasferimento
 - Ingresso:
 - ACQUISISCI il valore della *variabile*, ...
 - Uscita:
 - STAMPA il valore della *variabile*, ...
 - STAMPA il “messaggio”

Istruzioni (cont.)

- Istruzioni di controllo
 - Istruzioni condizionali (alternative)
 - SE una condizione è vera allora
sequenza di istruzioni
 - ALTRIMENTI
sequenza di istruzioni
 - Istruzioni ripetitive (cicliche)
 - RIPETI finché una *condizione* è vera
sequenza di istruzioni

Automazione della soluzione

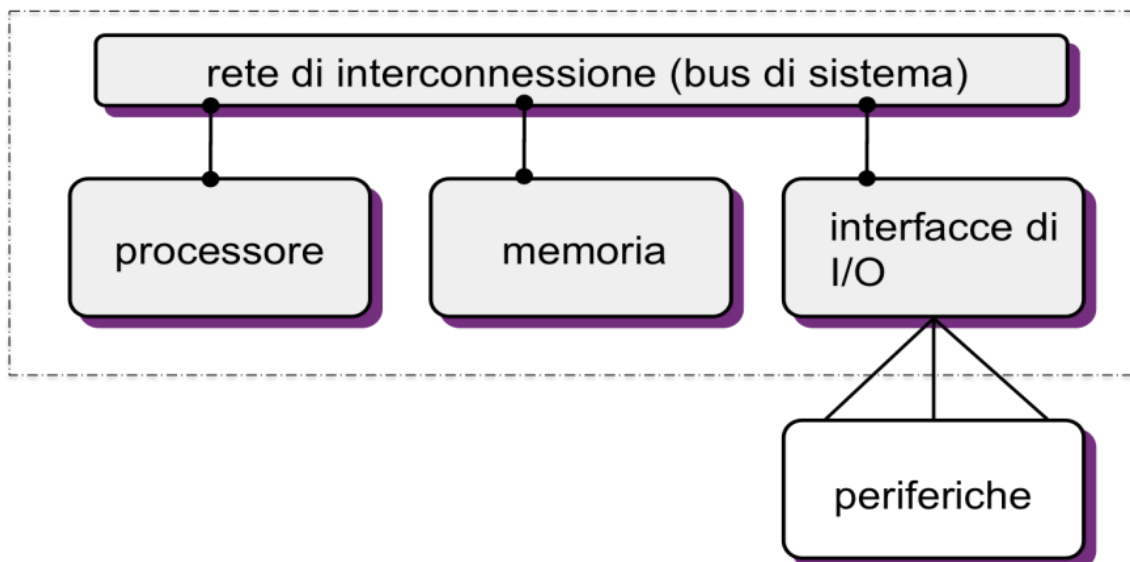
- Se siamo capaci di specificare un algoritmo per risolvere un problema, allora siamo in grado di automatizzare la soluzione
- Per fare ciò bisogna saper costruire una macchina capace di eseguire i passi/istruzioni dell'algoritmo

Il calcolatore come esecutore di algoritmi

- Calcolatore
 - Macchina calcolatrice che accetta in ingresso informazioni codificate in forma digitale, le elabora attraverso un programma memorizzato e produce informazioni in uscita
- Un calcolatore è un esecutore di algoritmi in cui
 - un algoritmo viene descritto per mezzo di un **programma** ed
 - il programma è una sequenza di istruzioni espresse in un linguaggio “comprensibile” al calcolatore (**linguaggio di programmazione**)

Struttura logica di un calcolatore

Architettura di von Neumann (1946)

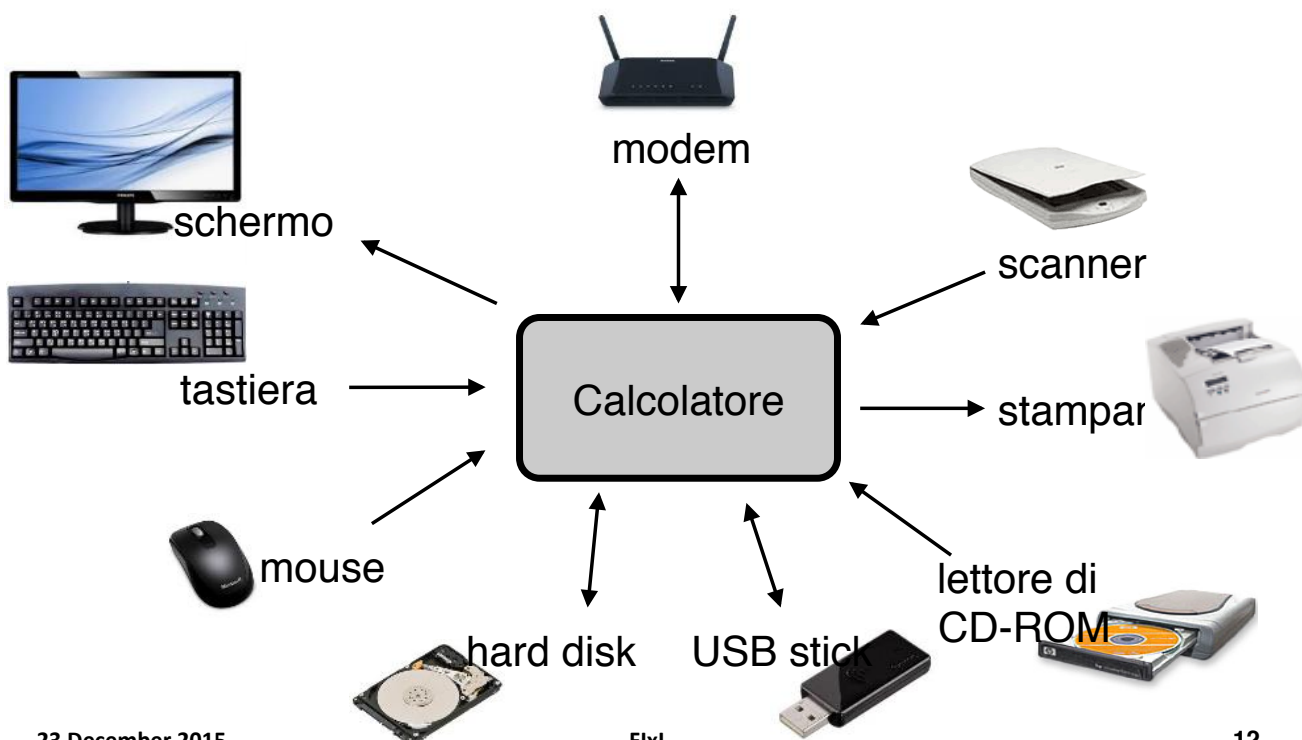


23 December 2015

FixL

11

Calcolatore e periferiche



23 December 2015

FixL

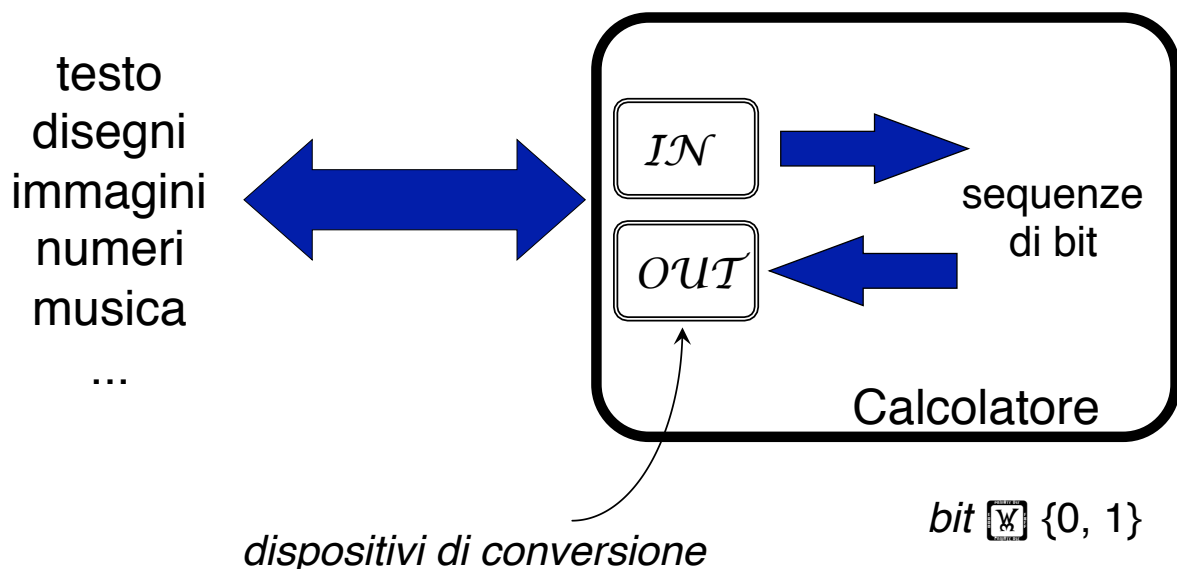
12

Descrizione funzionale

- La memoria contiene dati ed i programmi (istruzioni)
- Il processore ripete all'infinito le seguenti azioni:
 - preleva una nuova istruzione dalla memoria
 - la decodifica (capisce cosa deve fare!)
 - la esegue (lo fa!)
- L'esecuzione di un'istruzione può comportare
 - Calcolo di un'espressione e/o
 - Trasferimenti memoria-CPU e CPU-I/O)
- Le periferiche permettono al calcolatore di interagire con il mondo esterno (trasduttori)

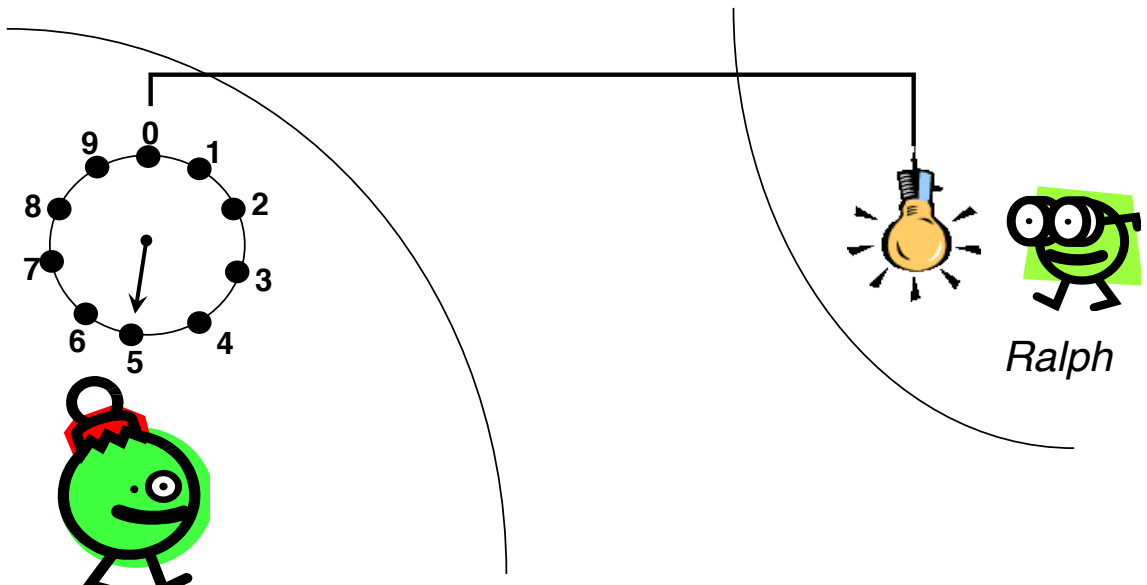
Calcolatore e Informazione

In un calcolatore i dati e le istruzioni sono codificati in forma **binaria**



Perché i bit (I)

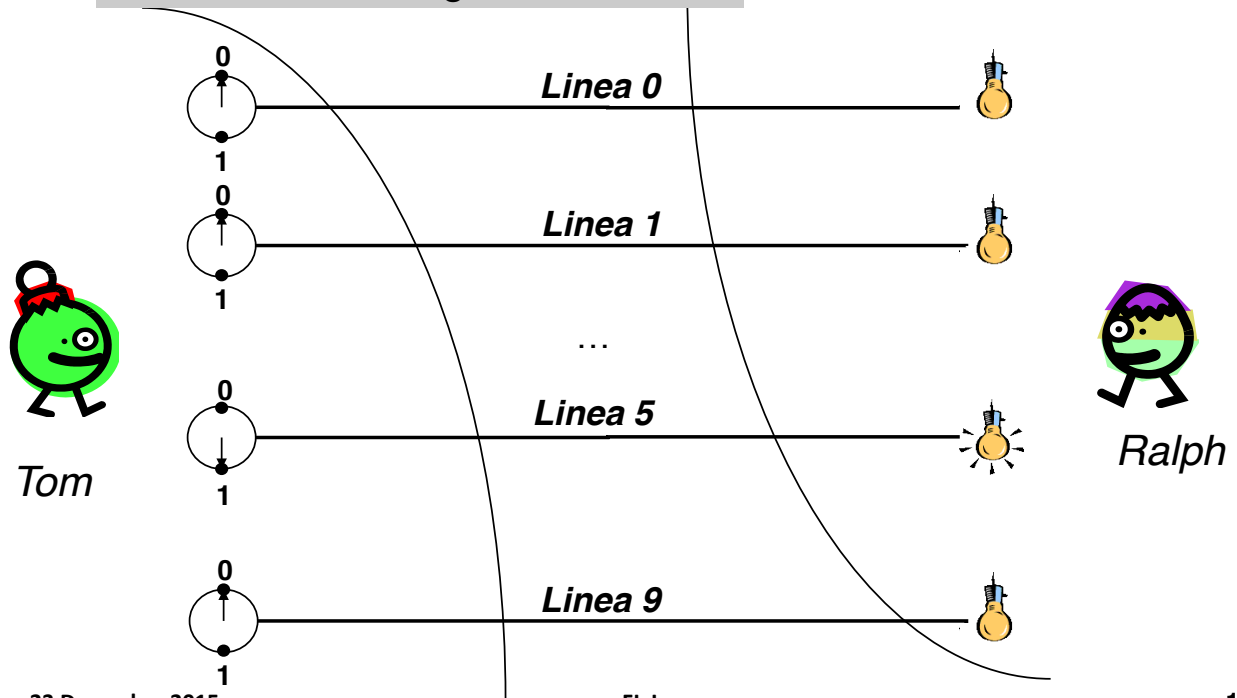
Tom vuole trasmettere a Ralph un'informazione numerica {0..9}



Tom e Ralph lavorano in base dieci

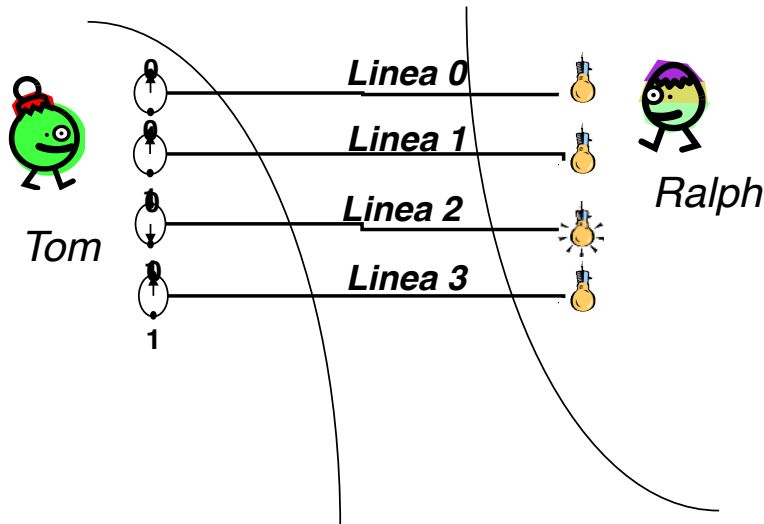
Perché i bit (II)

Una soluzione migliore



Perché i bit (III)

Una soluzione migliore



Cifra	Linea			
	3	2	1	0
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				

23 December 2015

FixL

17

Rappresentazione Naturali (I)

BASE DIECI

- Cifre: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Rappresentazione posizionale
 $(123)_{dieci}$ significa $1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$

BASE DUE

- Cifre: 0, 1
- Rappresentazione posizionale
 $(11001)_{due}$ significa
 $1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 (= 25)_{dieci}$

23 December 2015

FixL

18

Rappresentazione Naturali (II)

ESPRESSIONE DI UN NUMERO NATURALE IN BASE DUE

Algoritmo del mod e div

Esempio: $n = 47$

inizio

	QUOZIENTE	RESTO
	47	1
div 2	23	1
div 2	11	1
div 2	5	1
div 2	2	0
div 2	1	1
div 2	0	—

fine

$(101111)_{due}$

23 December 2015

FixL

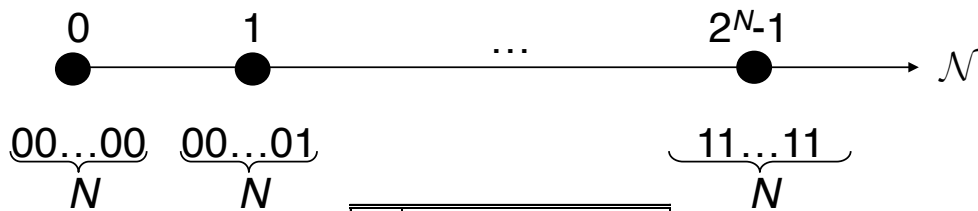
19

Esempi (base due)

- $N = 2$ bit
 - 4 (2^2) possibili configurazioni
 - 4 (2^2) possibili valori
 - 00 (0)
 - 01 (1)
 - 10 (2)
 - 11 (3)
- $N = 3$ bit
 - 8 (2^3) possibili configurazioni
 - 8 (2^3) possibili valori
 - 000 (0)
 - 001 (1)
 - 010 (2)
 - 011 (3)
 - 100 (4)
 - 101 (5)
 - 110 (6)
 - 111 (7)

Proprietà

Intervallo di rappresentabilità con N bit



N	Intervallo
8	[0, 255]
16	[0, 65535]
32	[0, 4294967295]

Potenza della Base: $2^N = \underbrace{(100\dots00)}_N \text{ due}$

Il problema della somma

- Il calcolatore lavora con un numero finito di bit
- Esempio $N = 16$ bit
 - Intervallo [0, 65535]
 - $A = 13521$ (0011010011010001) è rappresentabile
 - $B = 27428$ (0110101100100100) è rappresentabile
 - $S = A + B = 40949$ (1001111111110101) è rappresentabile

 - $A = 30421$ (0111011011010101) è rappresentabile
 - $B = 43127$ (1010100001110111) è rappresentabile
 - $S = A + B > 2^{16}-1(65535)$ quindi **S non è rappresentabile su p bit**
 - Si dice che la somma ha dato luogo ad un riporto (*carry*)
- In generale, ci vogliono $N+1$ bit per rappresentare la somma di due numeri di N bit

Somma tra numeri naturali

- La somma è data dalla somma degli addendi, cifra per cifra, da destra a sinistra, tenendo conto dei riporti.

In base dieci

0010	+	236	=	457	=	693	riporto	addendo	addendo	somma

In base due

0010	+	101	=	001	=	110	riporto	addendo	addendo	somma

Come mi accordo se/quando la somma fallisce?

N = 4

- A = 0101 (5)
B = 0110 (6)
S = A + B = ?

Riporto	0	1	0	0	0
A		0	1	0	1 +
B		0	1	1	0 =
S		1	0	1	1

- A = 1101 (13)
B = 0110 (6)
S = A + B = ?

Riporto	1	1	0	0	0
A		1	1	0	1 +
B		0	1	1	0 =
S		0	0	1	1

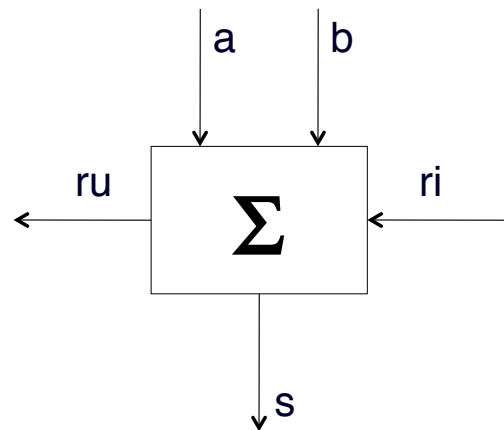
La somma fallisce quando il riporto uscente è uguale a 1

Somma ad un bit

- **a**: addendo
- **b**: addendo
- **s**: somma
- **ri**: riporto entrante
- **ru**: riporto uscente

a	b	ri	s	ru
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

Tabella di verità

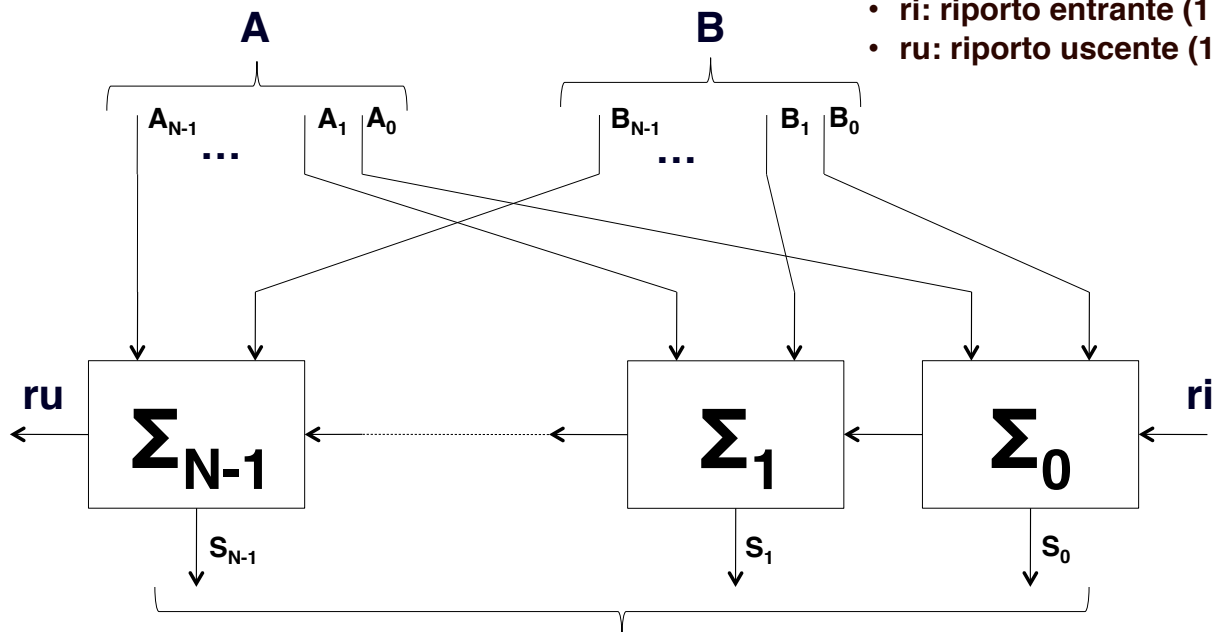


Sommatore ad un bit

Realizza la tabella di verità

Somma a N bit

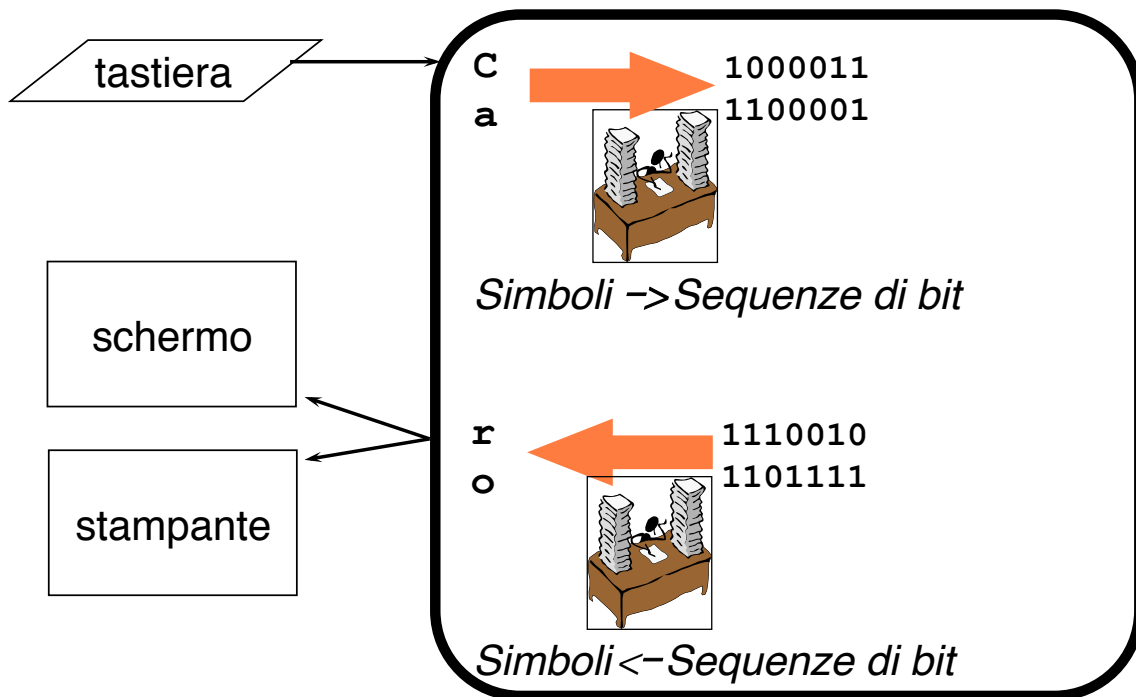
- **A**: addendo (N bit)
- **B**: addendo (N bit)
- **S**: somma (N bit)
- **ri**: riporto entrante (1 bit)
- **ru**: riporto uscente (1 bit)



S

N sommatore ad 1-bit sono montati in cascata

Rappresentazione del testo



23 December 2015

FixL

27

La codifica ASCII

American Serial Code
for Information
Interchange

Byte	Cod.	Char	Byte	Cod.	Char	Byte	Cod.	Char	Byte	Cod.	Char
00000000	0	Null	00100000	32	Spc	01000000	64	@	01100000	96	`
00000001	1	Start of heading	00100001	33	!	01000001	65	A	01100001	97	a
00000010	2	Start of text	00100010	34	"	01000010	66	B	01100010	98	b
00000011	3	End of text	00100011	35	#	01000011	67	C	01100011	99	c
00000100	4	End of transmit	00100100	36	\$	01000100	68	D	01100100	100	d
00000101	5	Enquiry	00100101	37	%	01000101	69	E	01100101	101	e
00000110	6	Acknowledge	00100110	38	&	01000110	70	F	01100110	102	f
00000111	7	Audible bell	00100111	39	'	01000111	71	G	01100111	103	g
00010000	8	Backspace	00101000	40	(01001000	72	H	01101000	104	h
00010001	9	Horizontal tab	00101001	41)	01001001	73	I	01101001	105	i
00010010	10	Line feed	00101010	42	*	01001010	74	J	01101010	106	j
00010011	11	Vertical tab	00101011	43	+	01001011	75	K	01101011	107	k
00011000	12	Form Feed	00101100	44	,	01001100	76	L	01101100	108	l
00011001	13	Carriage return	00101101	45	-	01001101	77	M	01101101	109	m
00011010	14	Shift out	00101110	46	.	01001110	78	N	01101110	110	n
00011011	15	Shift in	00101111	47	/	01001111	79	O	01101111	111	o
00010000	16	Data link escape	00110000	48	0	01010000	80	P	01110000	112	p
00010001	17	Device control 1	00110001	49	1	01010001	81	Q	01110001	113	q
00010010	18	Device control 2	00110010	50	2	01010010	82	R	01110010	114	r
00010011	19	Device control 3	00110011	51	3	01010011	83	S	01110011	115	s
00010100	20	Device control 4	00110100	52	4	01010100	84	T	01110100	116	t
00010101	21	Neg. acknowledge	00110101	53	5	01010101	85	U	01110101	117	u
00010110	22	Synchronous idle	00110110	54	6	01010110	86	V	01110110	118	v
00010111	23	End trans. block	00110111	55	7	01010111	87	W	01110111	119	w
00011000	24	Cancel	00111000	56	8	01011000	88	X	01111000	120	x
00011001	25	End of medium	00111001	57	9	01011001	89	Y	01111001	121	y
00011010	26	Substitution	00111010	58	:	01011010	90	Z	01111010	122	z
00011011	27	Escape	00111011	59	;	01011011	91	[01111011	123	{
00011100	28	File separator	00111100	60	<	01011100	92	\	01111100	124	
00011101	29	Group separator	00111101	61	=	01011101	93]	01111101	125	}
00011110	30	Record Separator	00111110	62	>	01011110	94	^	01111110	126	~
00011111	31	Unit separator	00111111	63	?	01011111	95	_	01111111	127	Del

23 December 2015

FixL

28

Rappresentazione Testo (ASCII)

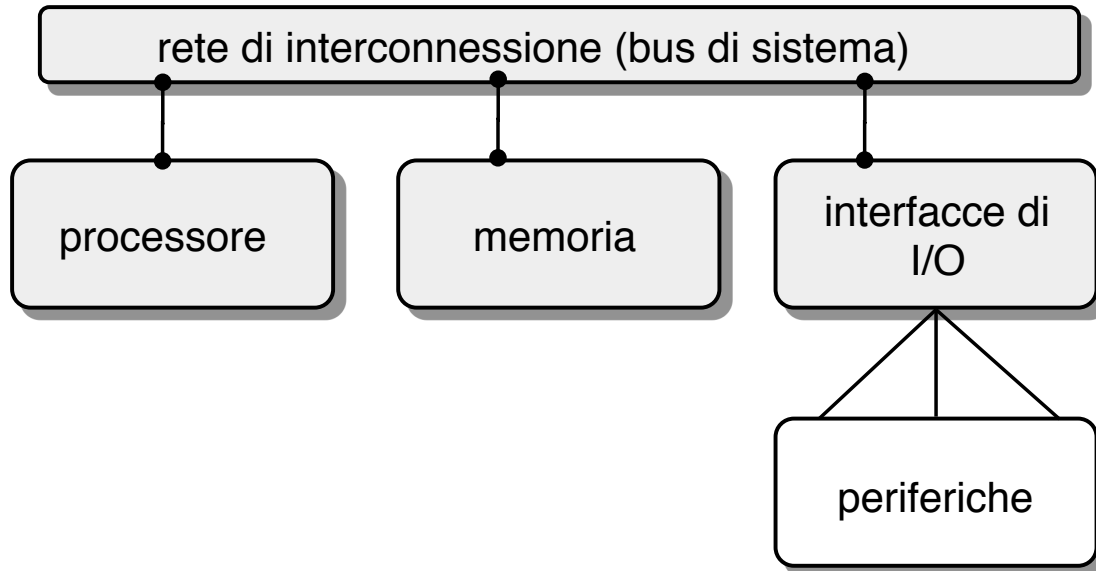
Byte	Cod	Char	Byte	Cod	Char	Byte	Cod	Char	Byte	Cod	Char
00000000	0	Null	00100000	32	Spc	01000000	64	@	01100000	96	
00000001	1	Start of heading	00100001	33	!	01000001	65	A	01100001	97	a
00000010	2	Start of text	00100010	34	"	01000010	66	B	01100010	98	b
00000011	3	End of text	00100011	35	#	01000011	67	C	01100011	99	c
00000100	4	End of transmit	00100100	36	\$	01000100	68	D	01100100	100	d
00000101	5	Enquiry	00100101	37	%	01000101	69	E	01100101	101	e
00000110	6	Acknowledge	00100110	38	&	01000110	70	F	01100110	102	f
00000111	7	Audible bell	00100111	39	'	01000111	71	G	01100111	103	g
00001000	8	Backspace	00101000	40	(01001000	72	H	01101000	104	h
00001001	9	Horizontal tab	00101001	41)	01001001	73	I	01101001	105	i
00001010	10	Line feed	00101010	42	*	01001010	74	J	01101010	106	j
00001011	11	Vertical tab	00101011	43	+	01001011	75	K	01101011	107	k
00001100	12	Form Feed	00101100	44	,	01001100	76	L	01101100	108	l
00001101	13	Carriage return	00101101	45	-	01001101	77	M	01101101	109	m
00001110	14	Shift out	00101110	46	.	01001110	78	N	01101110	110	n
00001111	15	Shift in	00101111	47	/	01001111	79	O	01101111	111	o
00010000	16	Data link escape	00110000	48	0	01010000	80	P	01110000	112	p
00010001	17	Device control 1	00110001	49	1	01010001	81	Q	01110001	113	q
00010010	18	Device control 2	00110010	50	2	01010010	82	R	01110010	114	r
00010011	19	Device control 3	00110011	51	3	01010011	83	S	01110011	115	s
00010100	20	Device control 4	00110100	52	4	01010100	84	T	01110100	116	t
00010101	21	Neg. acknowledge	00110101	53	5	01010101	85	U	01110101	117	u
00010110	22	Synchronous idle	00110110	54	6	01010110	86	V	01110110	118	v
00010111	23	End trans. block	00110111	55	7	01010111	87	W	01110111	119	w
00011000	24	Cancel	00111000	56	8	01011000	88	X	01111000	120	x
00011001	25	End of medium	00111001	57	9	01011001	89	Y	01111001	121	y
00011010	26	Substitution	00111010	58	:	01011010	90	Z	01111010	122	z
00011011	27	Escape	00111011	59	;	01011011	91	[01111011	123	{
00011100	28	File separator	00111100	60	<	01011100	92	\	01111100	124	
00011101	29	Group separator	00111101	61	=	01011101	93]	01111101	125	}
00011110	30	Record Separator	00111110	62	>	01011110	94	^	01111110	126	~
00011111	31	Unit separator	00111111	63	?	01011111	95	_	01111111	127	Del

→ 1000011
 1100001
 1110010
 1101111
 → 0100000
 Caro amico,
 1100001
 1101110
 1101001
 1100011
 1101111
 → 0101101

La codifica ASCII estesa (Italiano)

Byte	Cod	Char	Byte	Cod	Char	Byte	Cod	Char	Byte	Cod	Char
10000000	128	Ç	10100000	160	à	11000000	192	+	11100000	224	Ó
10000001	129	ü	10100001	161	í	11000001	193	-	11100001	225	Ô
10000010	130	ë	10100010	162	ó	11000010	194	-	11100010	226	Ö
10000011	131	â	10100011	163	ú	11000011	195	-	11100011	227	Û
10000100	132	ã	10100100	164	û	11000100	196	-	11100100	228	ö
10000101	133	ä	10100101	165	Ñ	11000101	197	+	11100101	229	Û
10000110	134	å	10100110	166	ª	11000110	198	+	11100110	230	ü
10000111	135	ç	10100111	167	•	11000111	199	+	11100111	231	µ
10001000	136	ê	10101000	168	¿	11001000	200	+	11101000	232	Þ
10001001	137	ë	10101001	169	@	11001001	201	+	11101001	233	Û
10001010	138	è	10101010	170	¬	11001010	202	-	11101010	234	Û
10001011	139	í	10101011	171	½	11001011	203	-	11101011	235	Û
10001100	140	î	10101100	172	¾	11001100	204	-	11101100	236	Û
10001101	141	ï	10101101	173	ì	11001101	205	-	11101101	237	Û
10001110	142	Ä	10101110	174	«	11001110	206	+	11101110	238	Û
10001111	143	Å	10101111	175	»	11001111	207	+	11101111	239	Û
10010000	144	È	10110000	176	-	11010000	208	+	11110000	240	-
10010001	145	É	10110001	177	-	11010001	209	+	11110001	241	±
10010010	146	Æ	10110010	178	-	11010010	210	+	11110010	242	-
10010011	147	ø	10110011	179	-	11010011	211	+	11110011	243	¼
10010100	148	õ	10110100	180	-	11010100	212	+	11110100	244	½
10010101	149	ö	10110101	181	À	11010101	213	+	11110101	245	¾
10010110	150	ù	10110110	182	Á	11010110	214	+	11110110	246	+
10010111	151	ú	10110111	183	Â	11010111	215	+	11110111	247	+
10011000	152	ÿ	10111000	184	©	11011000	216	+	11111000	248	°
10011001	153	Û	10111001	185	-	11011001	217	+	11111001	249	-
10011010	154	Ü	10111010	186	-	11011010	218	+	11111010	250	-
10011011	155	ß	10111011	187	+	11011011	219	-	11111011	251	±
10011100	156	£	10111100	188	+	11011100	220	-	11111100	252	±
10011101	157	ø	10111101	189	+	11011101	221	-	11111101	253	±
10011110	158	×	10111110	190	¥	11011110	222	-	11111110	254	-
10011111	159	f	10111111	191	+	11011111	223	-	11111111	255	-

Architettura di von Neumann (ripresa)



23 December 2015

FixL

31

Trasferimento dell'informazione

Ricorda
Ralph e Tom

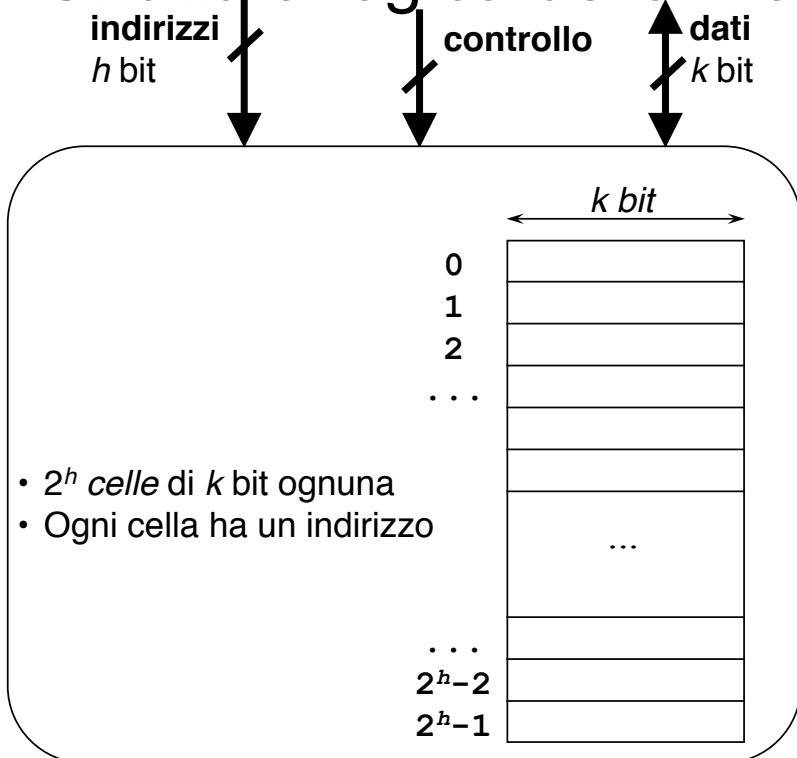
- Il **bus** è un insieme di *fili conduttori*, su ciascuno dei quali può essere impostata una *tensione*
- Su ciascun filo del bus, le tensioni "lecite" sono soltanto due
 - Tensione "alta" (5V, 3.3V)
 - Tensione "bassa" (0V)
- A ciascuna di queste tensioni è associato un *valore binario* (ad esempio: 5V=1, 0V=0)
- Possiamo quindi dire che, in un dato istante, su ciascuno dei fili del bus è presente un *bit*

23 December 2015

FixL

32

Struttura logica della memoria (I)

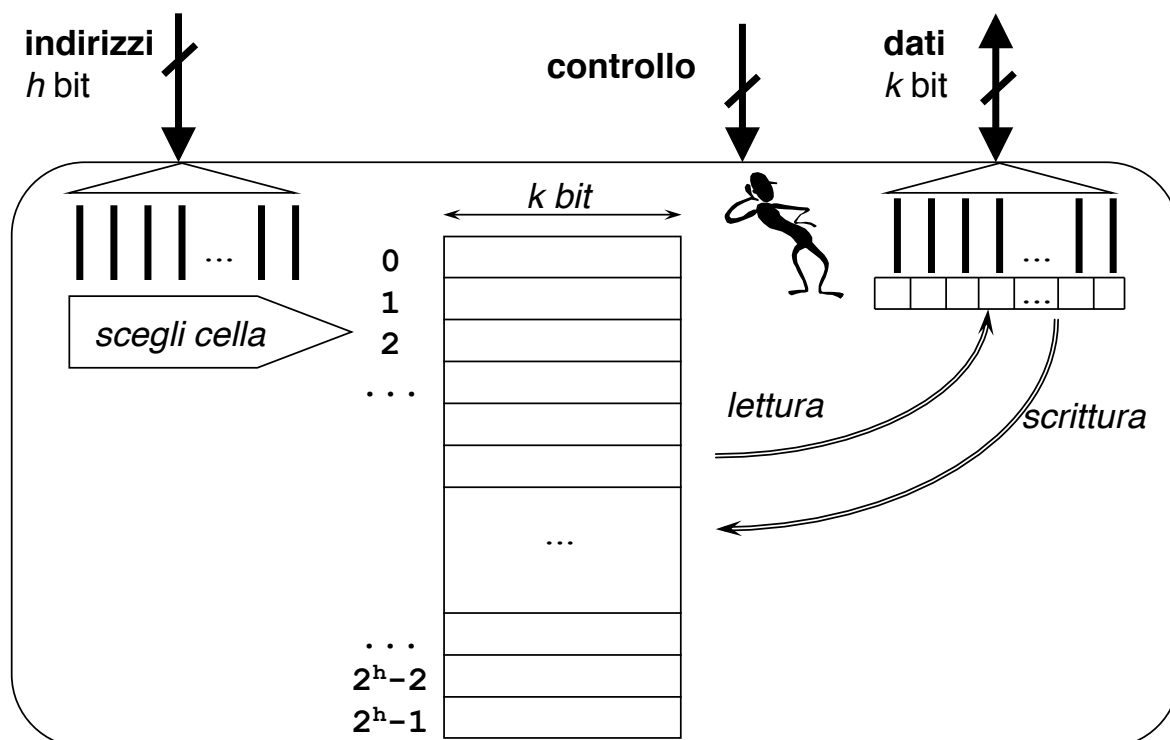


Sinonimi: *Cella, word, parola, locazione di memoria*

OPERAZIONI

1. LETTURA di UNA cella
2. SCRITTURA di UNA cella

Struttura logica della memoria (II)



Dimensione massima della memoria

- **Larghezza di una cella**

- Valori tipici per k (in bit): 8 (byte), 16, 32, 64

- **Numero di celle**

- h bit per gli indirizzi → 2^h celle

h	Numero di celle (2^h)	
16	65 536	64K
20	1 048 576	1M
22	4 194 304	4M
24	16 777 216	16M
32	4 294 967 296	4G

$$2^{10} = 1024 = 1K \text{ (kilo)}$$

$$2^{20} = 1\,048\,576 = 1\text{ M (mega)}$$

$$2^{30} = 1\,073\,741\,824 = 1G \text{ (giga)}$$

$$2^{40} = 1\,099\,511\,627\,776 = 1T \text{ (tera)}$$

...tanto per fissare le idee

Dimensione memoria	Ordine di grandezza	Informazione testuale
1 byte	10^0	un carattere
1 kilobyte	10^3	una pagina scritta
1 megabyte	10^6	due o tre racconti
1 gigabyte	10^9	una biblioteca dipartimentale
1 terabyte	10^{12}	la biblioteca di una università

RAM e ROM

- Random Access Memory (RAM)
 - Ogni cella può essere acceduta indipendentemente dalle altre
 - Il tempo di accesso è costante (30-70 ns)
 - Volatile
 - Costi: ~0.2 euro per Mbyte
- Read-Only Memory (ROM): unità RAM su cui si può *solo* leggere
 - Memoria protetta e permanente
 - EROM, PROM, EPROM

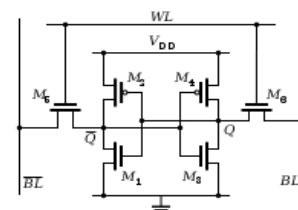
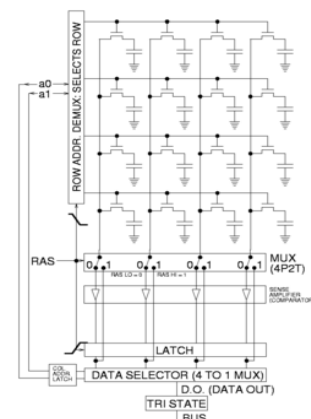
23 December 2015

FixL

37

Note tecnologiche

- **DRAM (Dynamic RAM)**
 - **Obiettivo principale: capacità**
 - Un transistor + 1 capacitore per ogni bit
 - Usate per main memory
 - Semplicità, Alta densità, basso costo
 - Necessità di refresh (~decine di ms)
- **SRAM (Static RAM)**
 - **Obiettivo principale: velocità**
 - Usate per cache
 - SRAM usa 6 transistor per bit
 - Non c'è bisogno di refresh
 - Costose



23 December 2015

FixL

38

Note tecnologiche

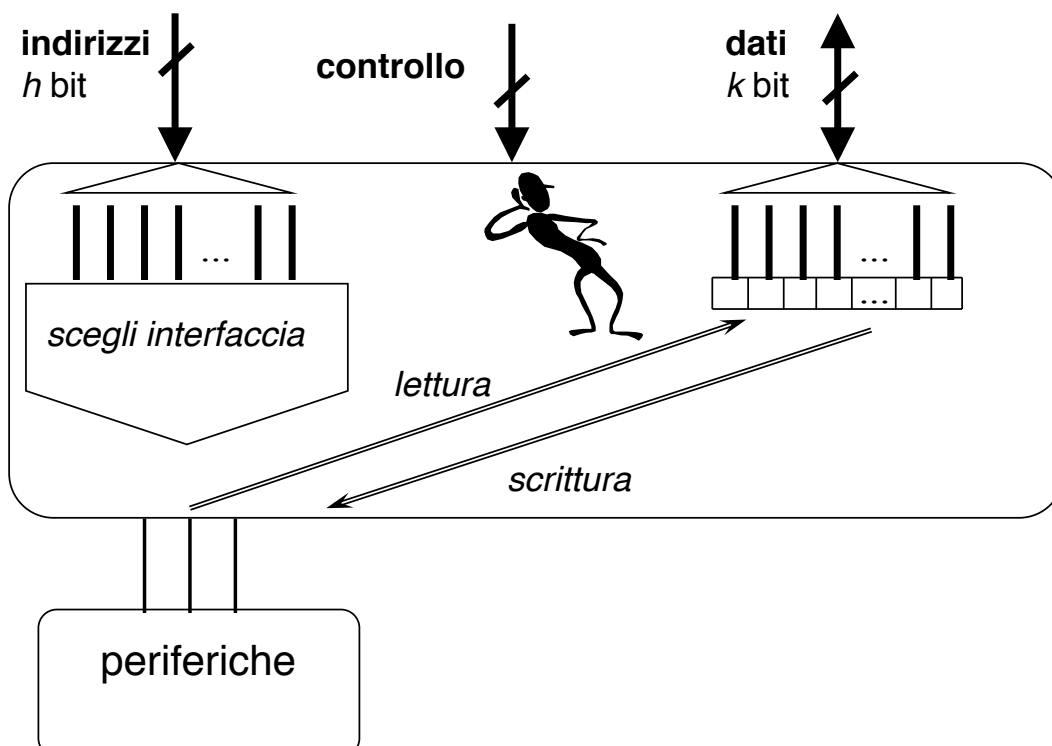
- Per memorie progettate con tecnologie comparabili
 - Capacità: DRAM/SRAM = 4-8
 - Velocità: SRAM/DRAM = 8-16
 - Costo: SRAM/DRAM = 8-16

23 December 2015

FixL

39

Struttura logica dello spazio di I/O

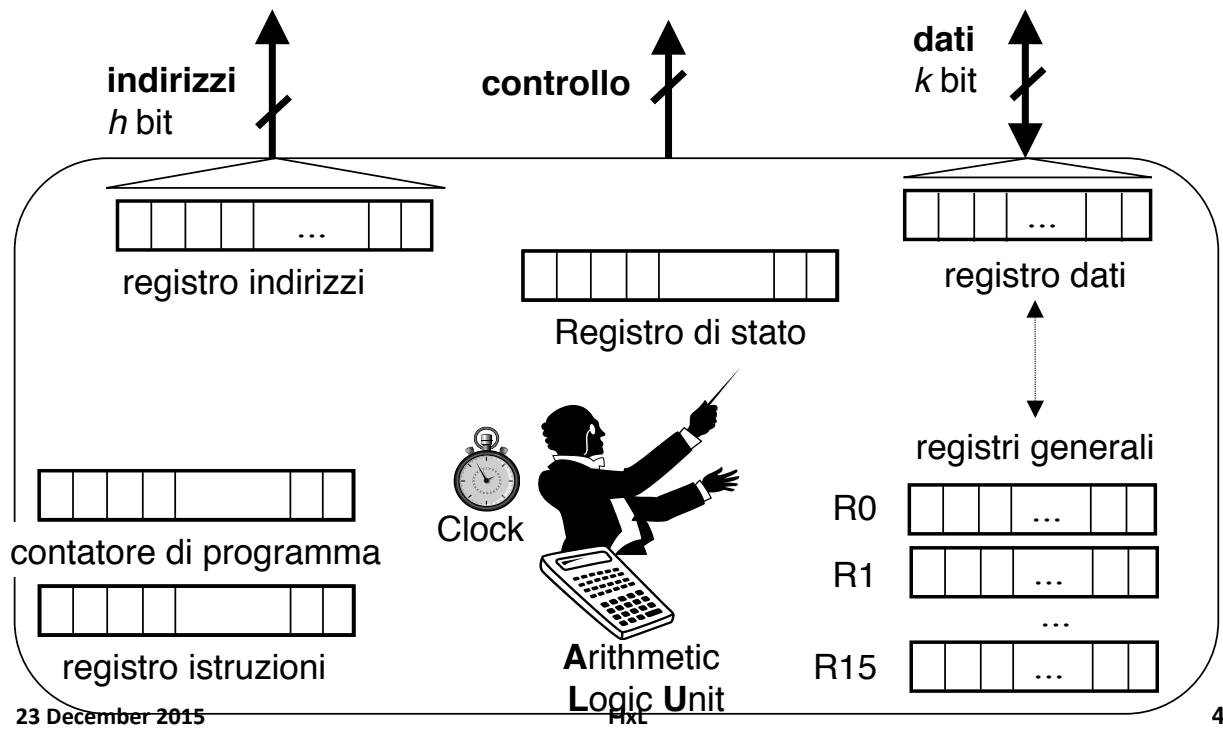


23 December 2015

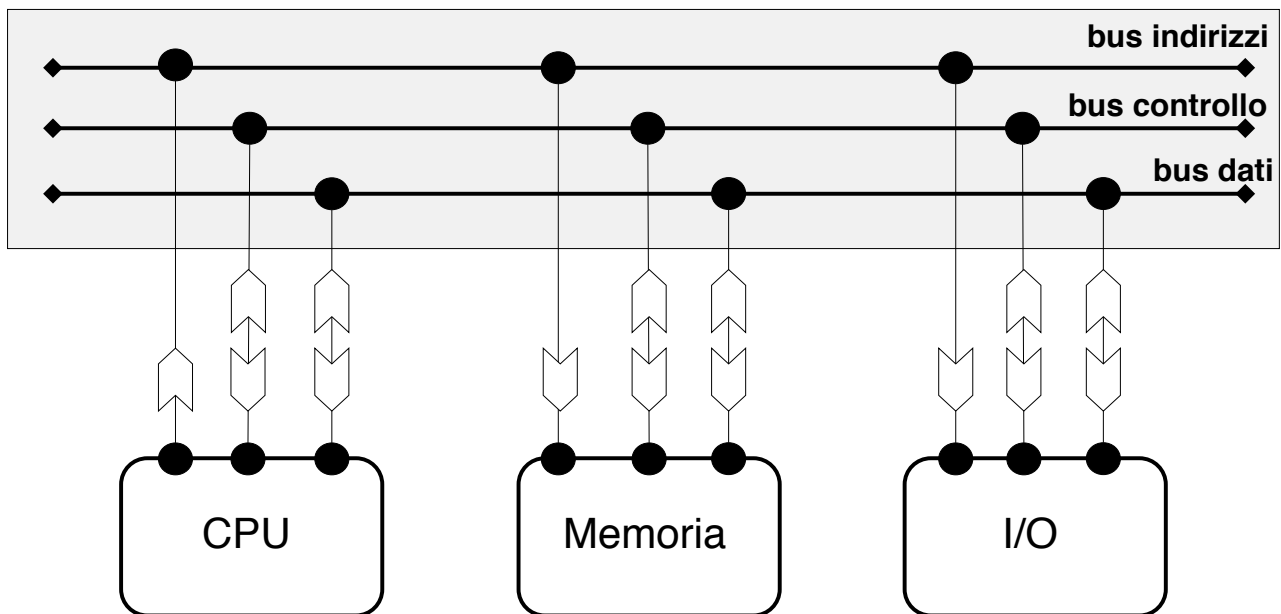
FixL

40

Struttura logica del processore



Struttura logica del Bus



Le linee del bus

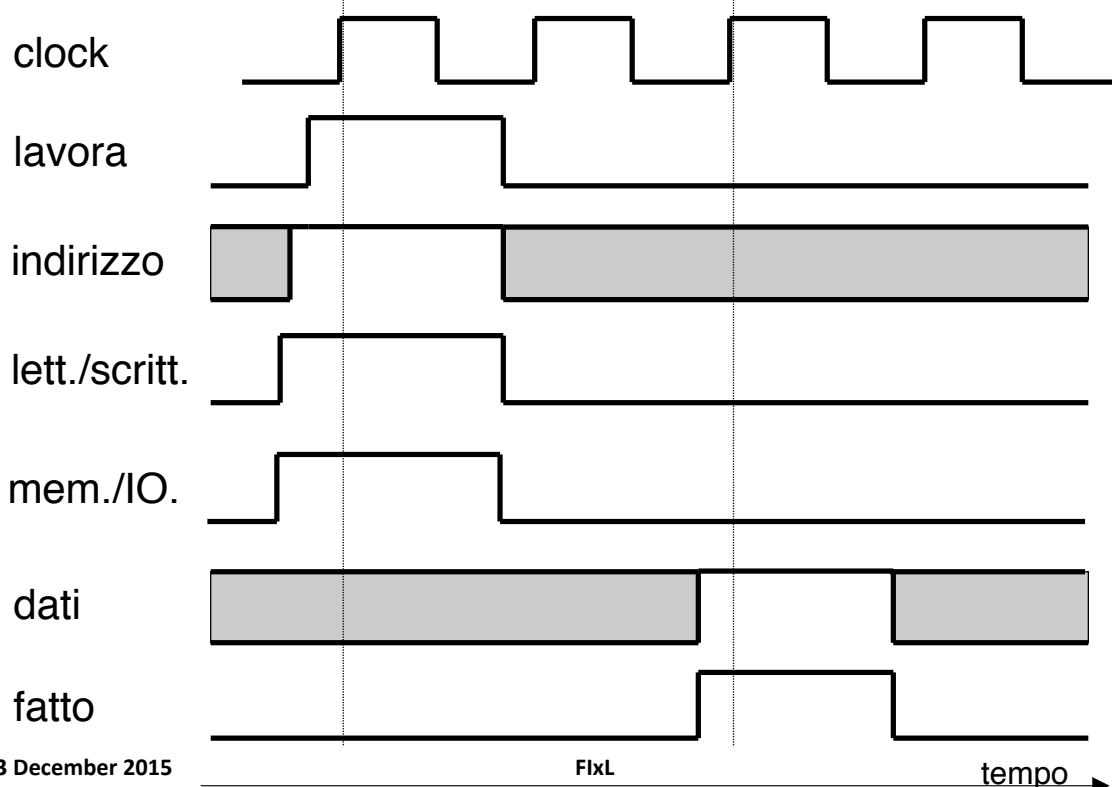
- **BUS DATI** - fili su cui vengono trasferiti i dati tra il processore e le altre componenti
- **BUS INDIRIZZI** - fili che specificano l'indirizzo della sorgente o della destinazione dei dati che si trovano sul Bus Dati.
- **BUS CONTROLLO** - fili che specificano l'accesso e l'uso del Bus Dati e del Bus Indirizzi:
 - **Lavora**: filo che indica che bisogna lavorare
 - **Memoria - I/O**: filo che indica "chi" deve lavorare
 - **Leggi - Scrivi**: filo che indica "cosa" bisogna fare
 - **Fatto**: filo che indica che il lavoro è stato fatto
- **Clock, Alimentazione...**

23 December 2015

FixL

43

Esempio di Temporizzazione (lettura in memoria)



23 December 2015

FixL

tempo

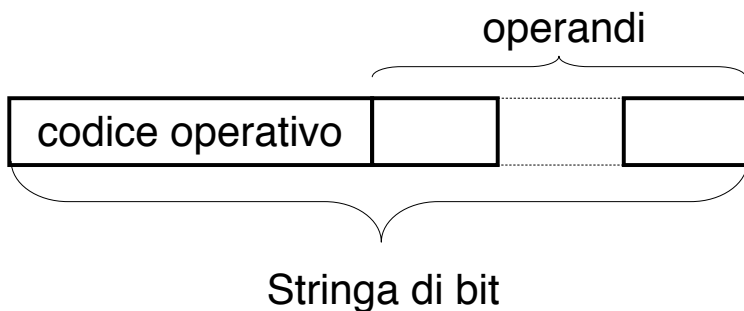
44

Programmi ed Istruzioni

La funzione principale di un calcolatore consiste nell' esecuzione da parte del processore di un programma memorizzato in memoria

Un **programma** è una sequenza di istruzioni

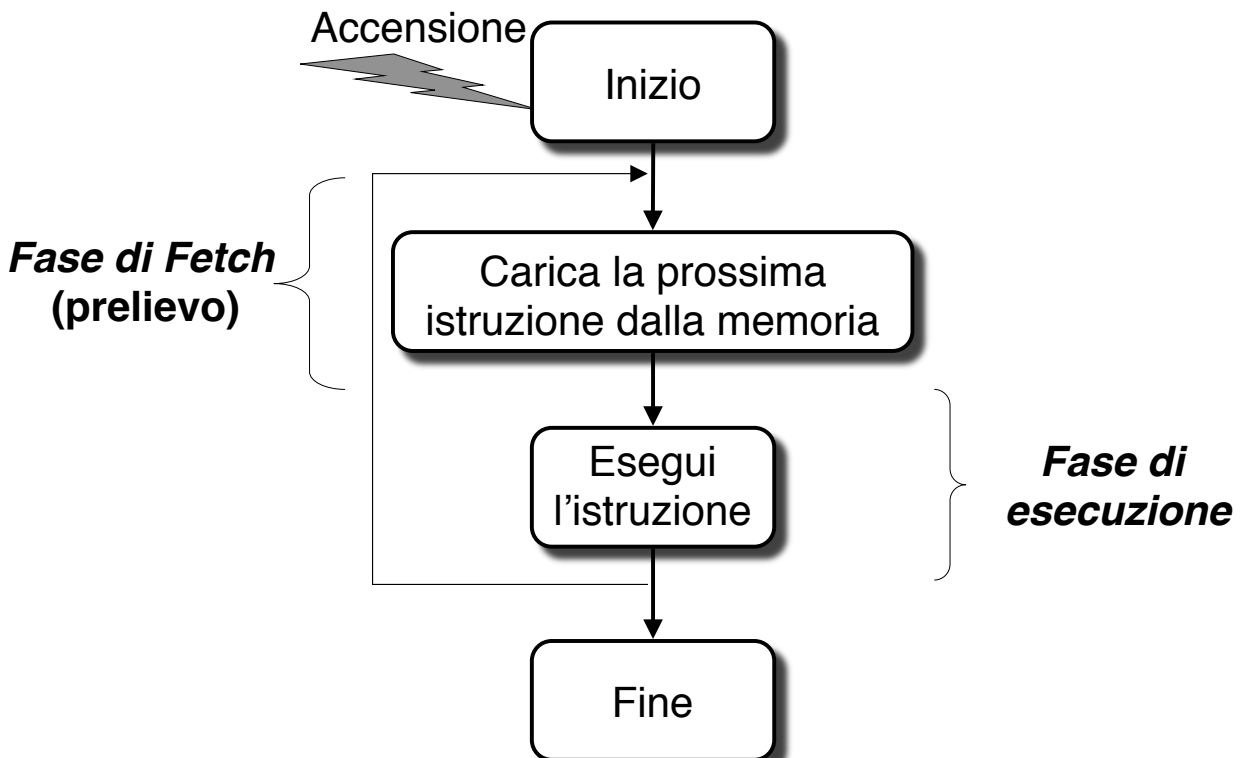
Un **istruzione** è un' informazione codificata che specifica "cosa fare" e "dove" farlo



TIPI DI ISTRUZIONE

- Trasferimento
- Elaborazione
- Controllo

Ciclo della CPU



Fase di Esecuzione

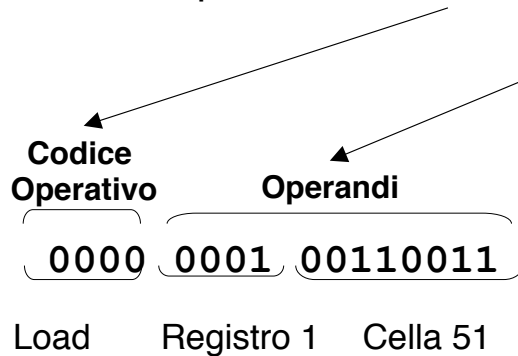
- La **fase di Esecuzione** consiste in due sottofasi:
 - A. Per prima cosa la CPU **decodifica** l'istruzione, cioè capisce cosa deve fare e dove deve farlo.
 - B. Poi la CPU **esegue** l'istruzione, cioè fa quello che ha capito al passo precedente.
- Il passaggio dalla Fase di Esecuzione alla Fase Fine viene effettuato eseguendo un'istruzione particolare.

Processore di esempio

- Quello visto nei precedenti lucidi, con 16 registri generali R0, R1, ..., R15
- Il processore sa eseguire **16 operazioni diverse**
- Inserito in un sistema in cui c'è una memoria RAM di **256 celle**, ciascuna di **16 bit**
 - **h=8, k=16**
- Ogni istruzione (che sta in una cella di memoria) è una **sequenza di 16 bit**

Esempio di Istruzione

Un'istruzione specifica: **cosa** fare e **dove** farlo



Carica (load) il contenuto della cella di indirizzo 51 nel registro R1

Tipi di Istruzioni

16 Istruzioni; 16 Registri; 256 Parole di memoria

TRASFERIMENTO

LOAD Reg Mem ($M \rightarrow CPU$)

STORE Reg Mem ($M \rightarrow CPU$)

IN Reg IO ($I/O \rightarrow CPU$)

OUT Reg IO ($I/O \rightarrow CPU$)

...

ELABORAZIONE

ADD Reg_dest Reg_sorg

SUB Reg_dest Reg_sorg

SET Reg_dest Numero

...

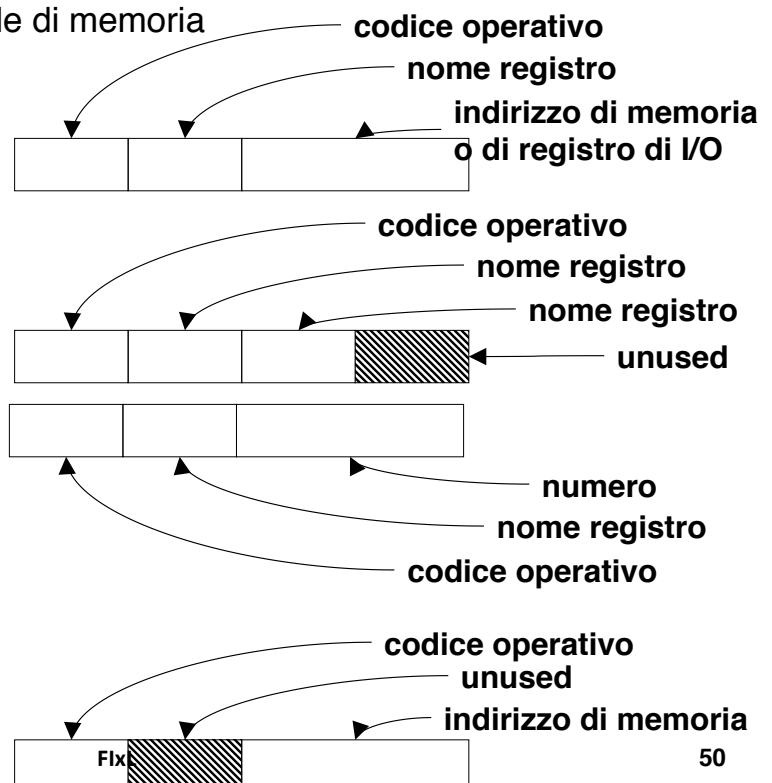
CONTROLLO

JMP Mem

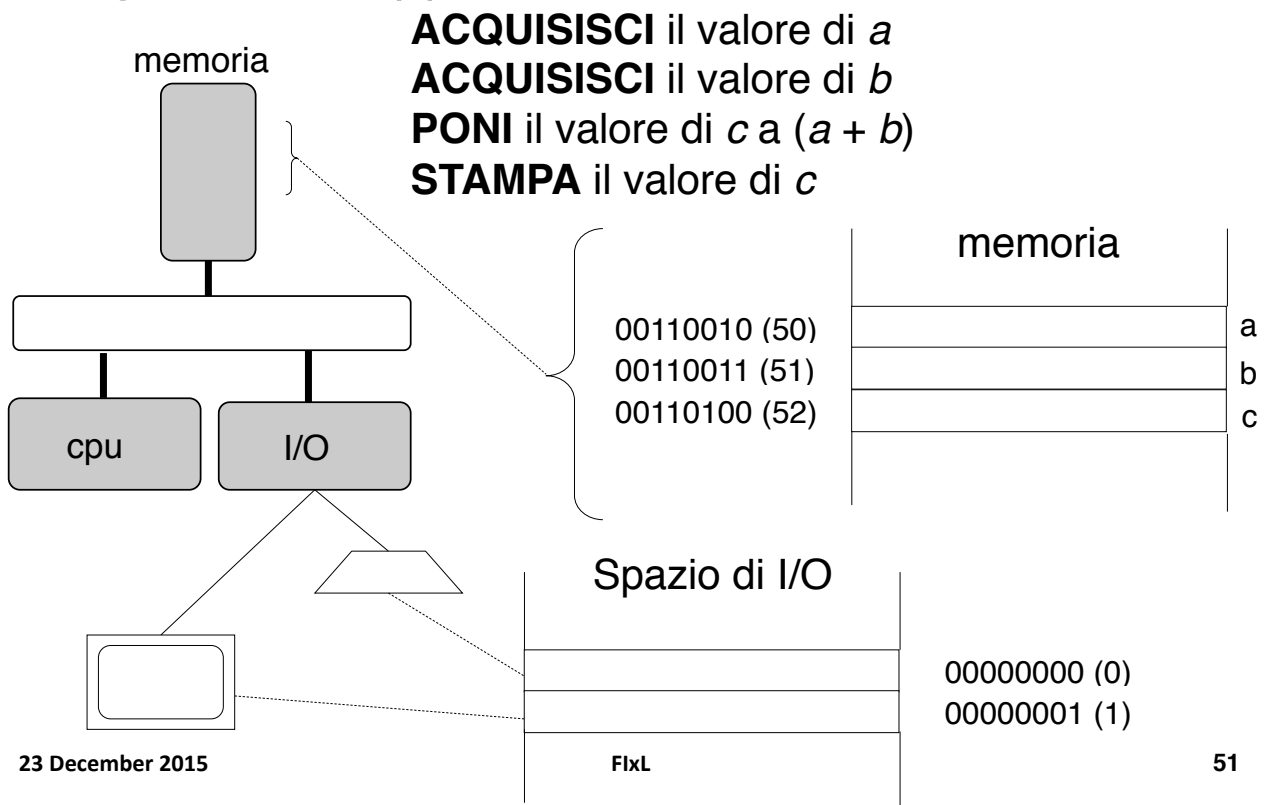
JZ Mem

JC Mem

... 23 December 2015



Programma (I)

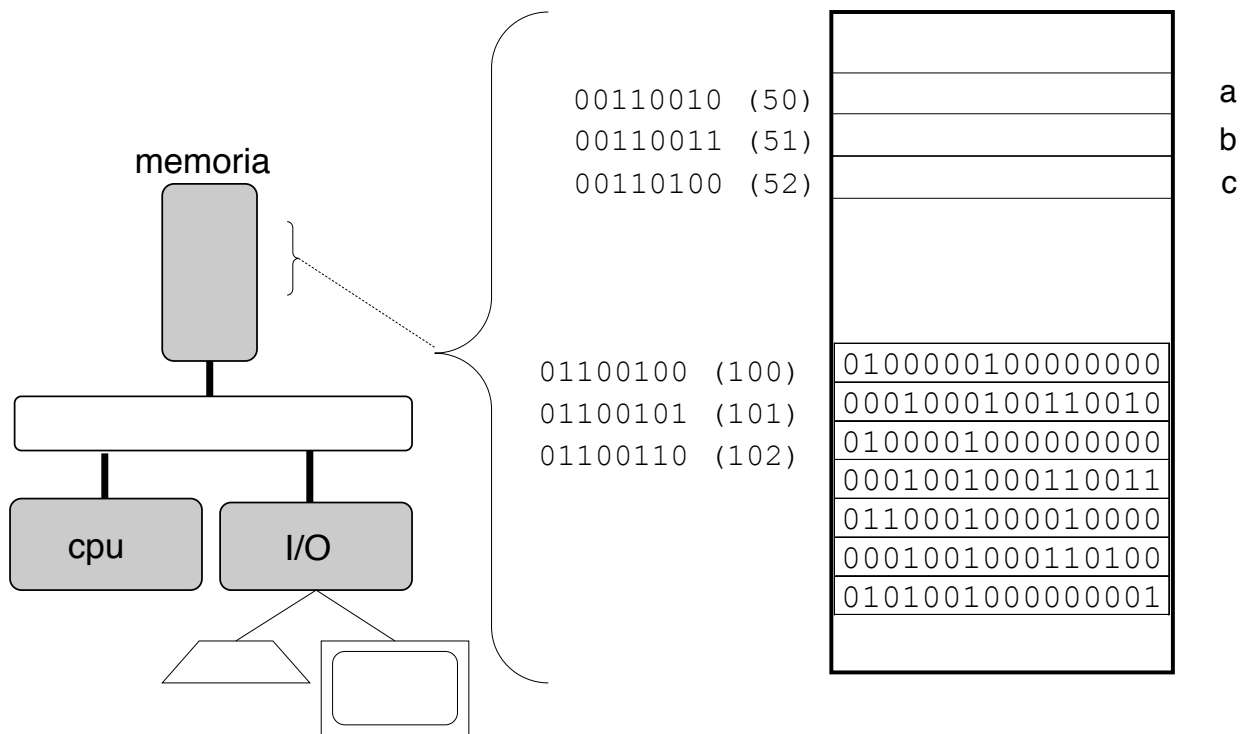


Programma (II)

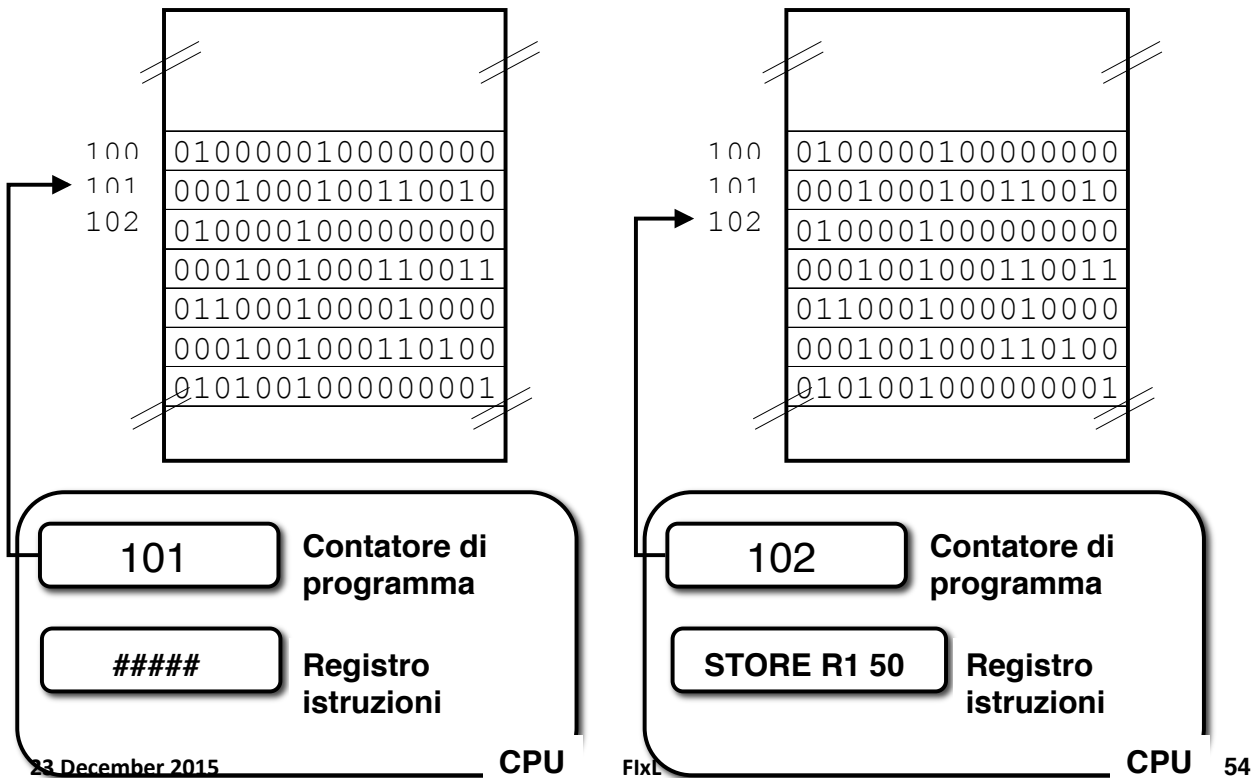
ACQUISISCI il valore di *a*
ACQUISISCI il valore di *b*
PONI il valore di *c* a (*a + b*)
STAMPA il valore di *c*

IN MEMORIA	MNEMONICO
0100 0001 00000000	IN R1 0
0001 0001 00110010	STORE R1 50
0100 0010 00000000	IN R2 0
0001 0010 00110011	STORE R2 51
0110 0010 0001xxxx	ADD R2 R1
0001 0010 00110100	STORE R2 52
0101 0010 00000001	OUT R2 1

Programma (III)



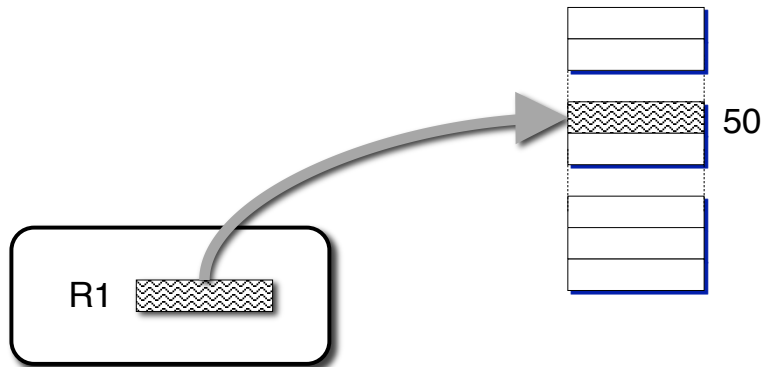
Fase di fetch



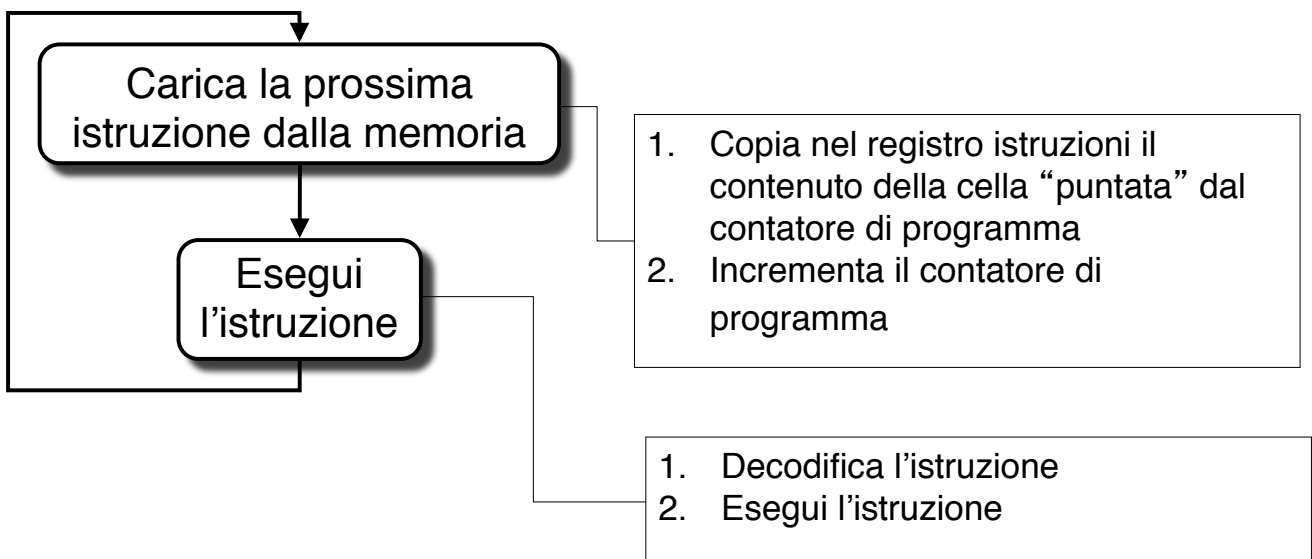
Fase di esecuzione

Esecuzione di **STORE R1 50**

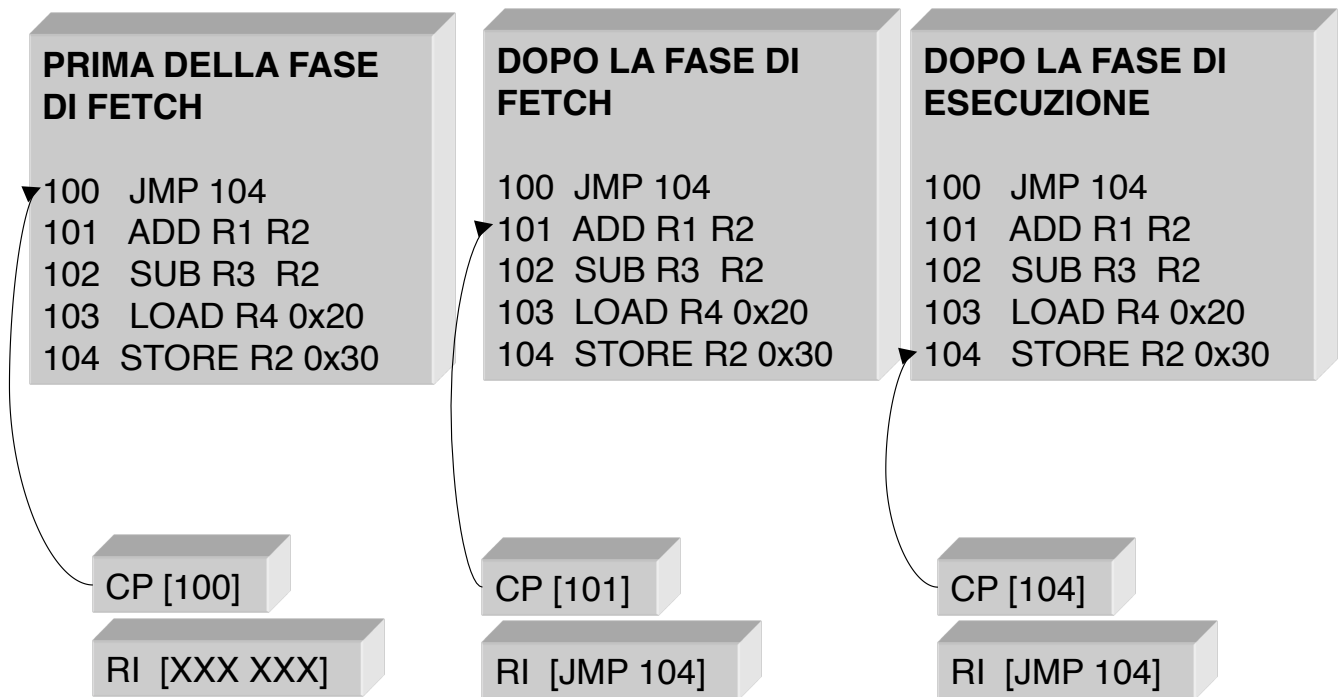
1. Il processore esegue un'operazione di scrittura in memoria all'indirizzo 50 (0011 0010)
2. I contenuti scritti nella cella di memoria 50 sono letti dal registro generale R1



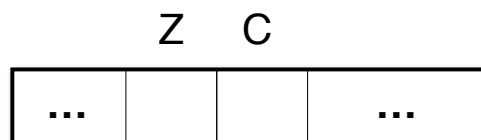
Ciclo della CPU (ripreso)



Istruzioni di controllo



Registro di stato



Alcune istruzioni di elaborazione modificano i bit del Registro di stato

- Se l'ultima istruzione di elaborazione che è stata eseguita ha prodotto un valore pari a zero, allora il bit Z vale 1; altrimenti vale 0
- Il bit C riporta il valore del carry (riporto/prestito) dell'ultima istruzione di elaborazione che è stata eseguita
- Alcune istruzioni di controllo (salti condizionati) eseguono il salto se i bit del registro di controllo assumono certi valori
- JZ = salta se il bit Z vale 1
- JC = salta se il bit C vale 1

Salto: Esempio I

(Supponiamo per ipotesi $a > 0$)

SE b è divisibile per a

PONI il valore di c a 1

ALTRIMENTI

PONI il valore di c a 0

Variabile **Indirizzo**

a 48

b 49

c 50

INDIRIZZO	ISTRUZIONE
100	LOAD R0 48
101	LOAD R1 49
102	SUB R1 R0
103	JC 108
104	JZ 106
105	JMP 102
106	SET R2 #1
107	JMP 109
108	SET R2 #0
109	STORE R2 50

Salto: Esempio II

(Supponiamo per ipotesi $b > 0$)

1. ACQUISISCI *il valore di a*

2. ACQUISISCI *il valore di b*

3. RIPETI i passi 4-5 *finché* $b > 0$

4. PONI *il valore di a ad* $(a + 1)$

5. PONI *il valore di b a* $(b - 1)$

6. PONI *il valore di c ad* a

Variabile **Indirizzo**

a 48

b 49

c 50

INDIRIZZO	ISTRUZIONE
100	IN R1 0
101	STORE R1 48
102	IN R2 0
103	STORE R2 49
104	SET R3 #1
105	ADD R1 R3
106	SUB R2 R3
107	JZ 109
108	JMP 105
109	STORE R1 50
110	OUT R1 1

Linguaggi ad alto livello

Esempio di programma in linguaggio C++: test di primalità

```
#include <iostream.h>

void main() {
    int n; // numero da testare
    int primo = 1; // indica se n è primo
    int d = 2; // divisore corrente

    cin >> n; // ACQUISISCI

    while ( (d < n) && // RIPETI
           ( primo == 1) )
        if ( n % d == 0 ) // SE
            primo = 0; // PONI
        else // ALTRIMENTI
            d = d+1;
    if ( primo == 1)
        cout << "E' PRIMO"; // STAMPA
    else
        cout << "NON E' PRIMO!";
}
```

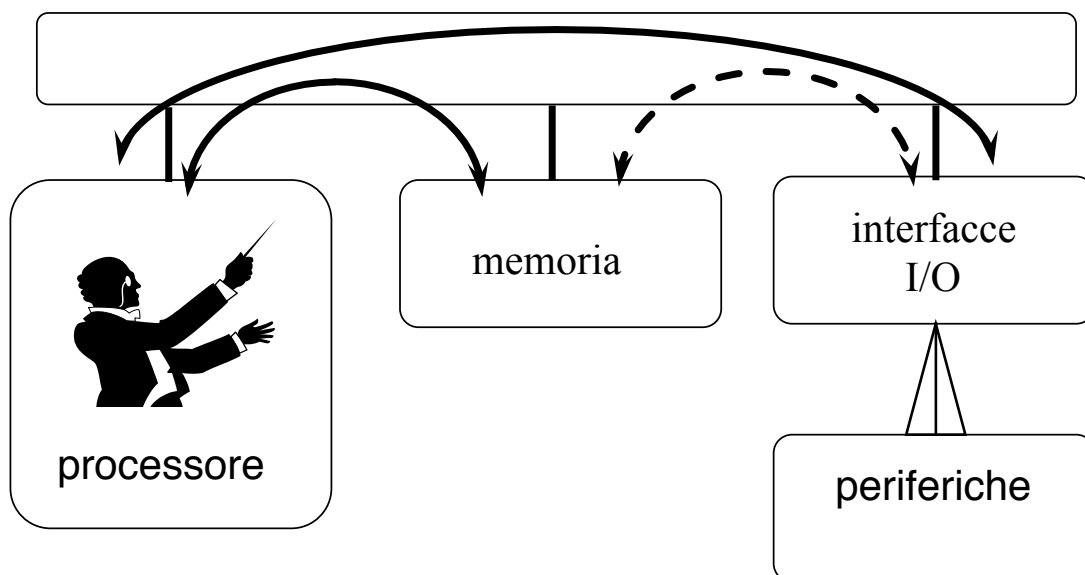


23 December 2015

FixL

61

Trasferimento dell'informazione



23 December 2015

FixL

62

Gerarchie di bus (I)

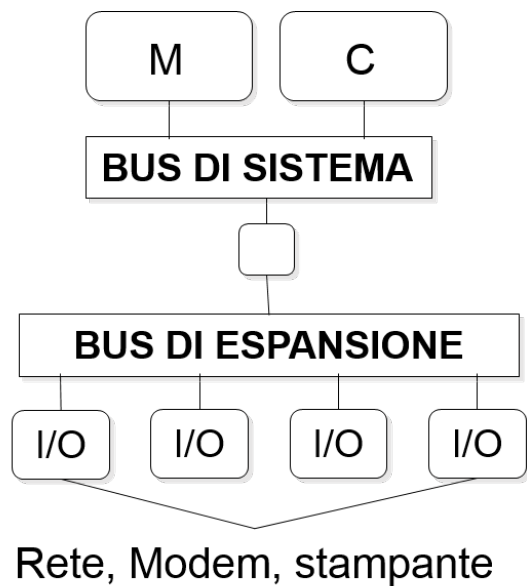
- Se un bus viene collegato a molti dispositivi si avrà un calo delle prestazioni
 - **Aumenta la lunghezza del bus** e quindi aumenta il tempo di propagazione delle informazioni lungo il bus
 - **La domanda totale dei dati** da trasferire tende ad avvicinarsi alla capacità del bus
 - **I dispositivi sono asimmetrici**
 - Un dispositivo “lento” (periferica) può rallentare un dispositivo veloce (memoria)
- Più linee, più informazione ma maggiori costi

Gerarchie di Bus (I)

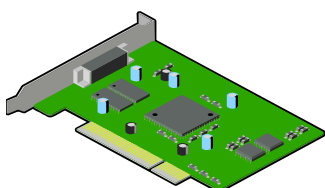
- Se un bus viene collegato a molti dispositivi si avrà un calo delle prestazioni a causa dei seguenti motivi:
 - **Aumenta la lunghezza del bus** e quindi aumenta il tempo di propagazione delle informazioni lungo il bus
 - **La domanda totale dei dati** da trasferire tende ad avvicinarsi alla capacità del bus

Gerarchie di Bus (II)

- La CPU interagisce più frequentemente con la memoria
- **Bus di sistema** è più corto e più veloce
- I dispositivi di I/O sono lenti perciò possono essere collegati al **bus di espansione**

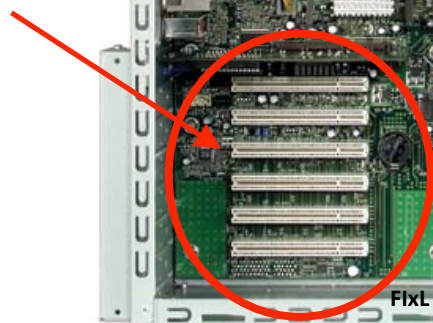


Bus di espansione e slot

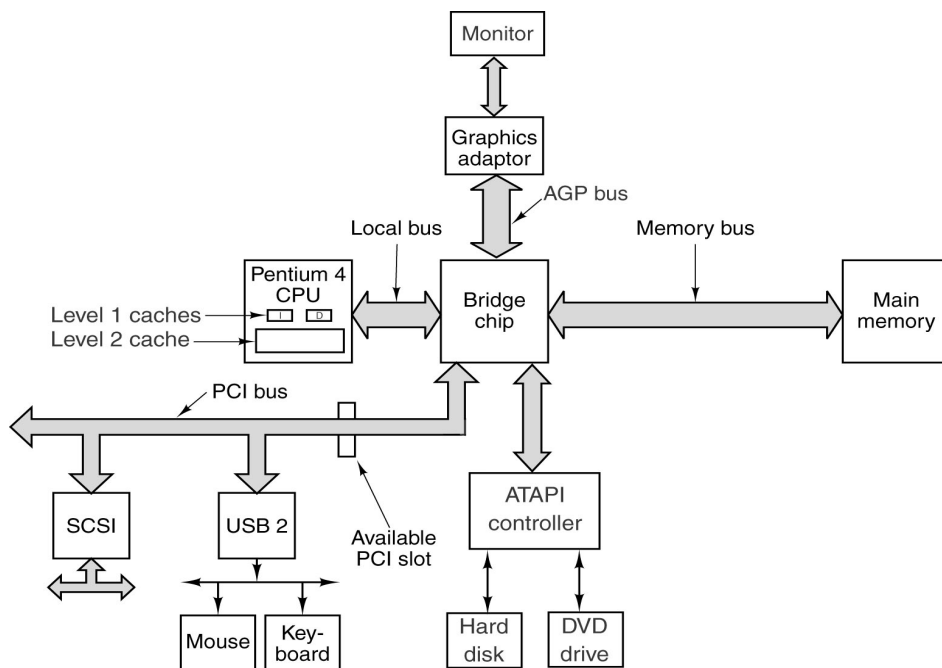


Interfaccia (scheda)

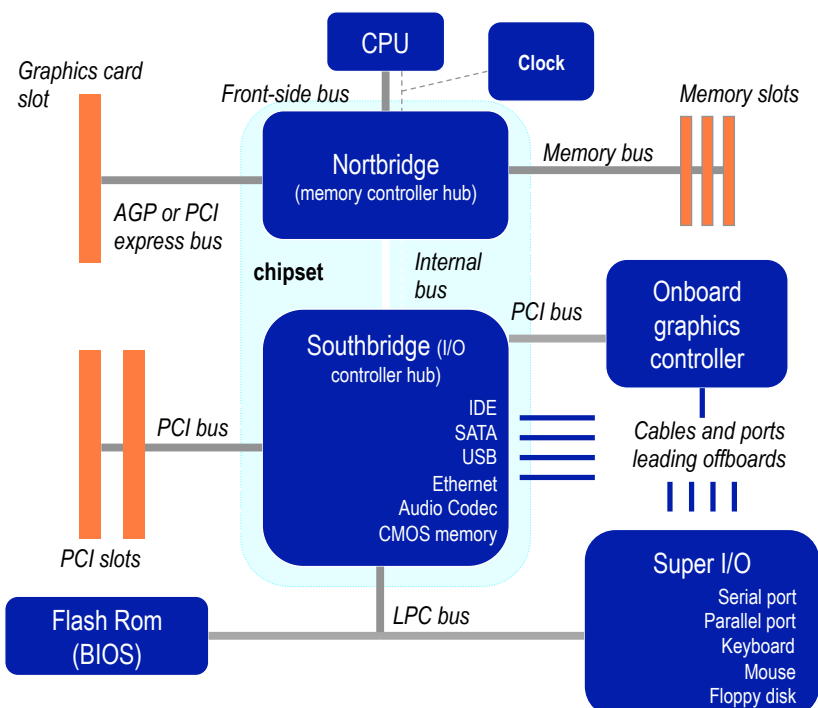
slot



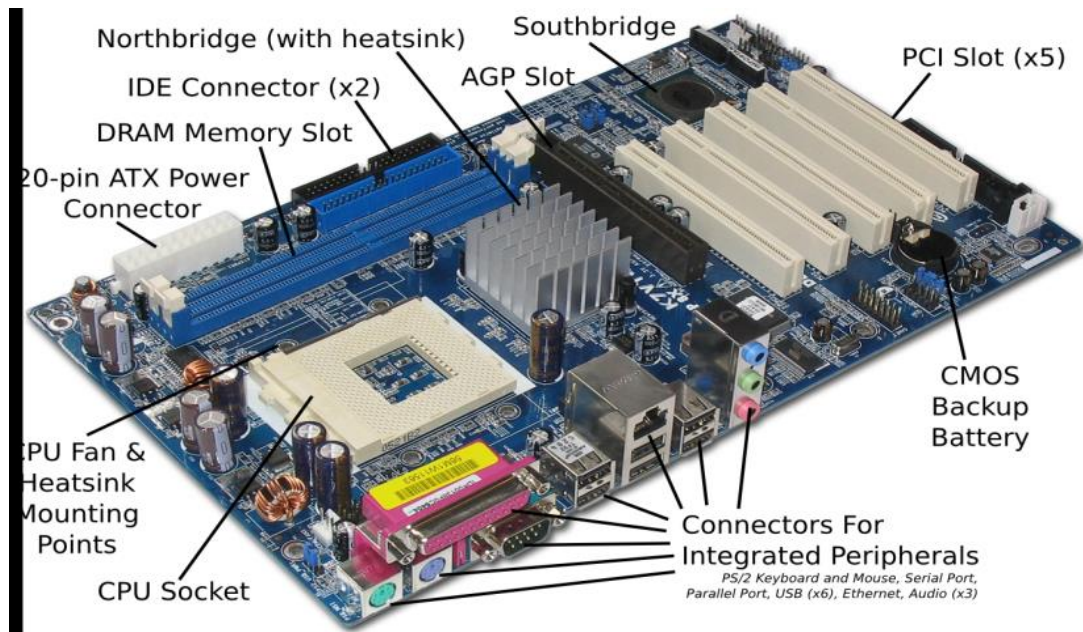
Struttura logica: una vista più accurata



Struttura del PC



Struttura del PC



23 December 2015

FixL

69

Aumentare le prestazioni del sistema

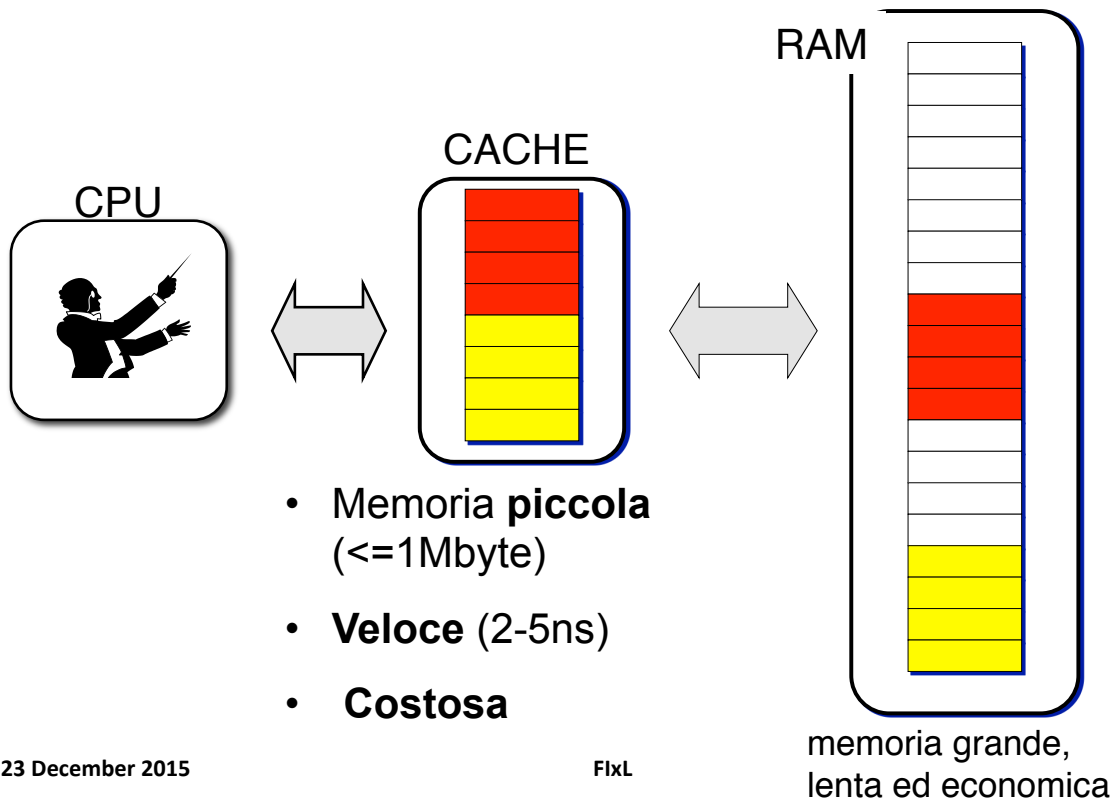
- Il tempo di accesso alla RAM è di $\sim 30-70$ ns
- Un processore con clock ad 1Ghz esegue *in media* 1 istruzione ogni ns
- Perciò quando un processore accede alla memoria, si blocca (aspetta) per molto tempo in attesa dei dati
 - La velocità *effettiva* del sistema computer è quella della memoria, non quella della CPU
 - La memoria è un “collo di bottiglia” per le prestazioni del sistema computer
- Migliorare la velocità di accesso alla memoria significa migliorare notevolmente le prestazioni d’insieme

23 December 2015

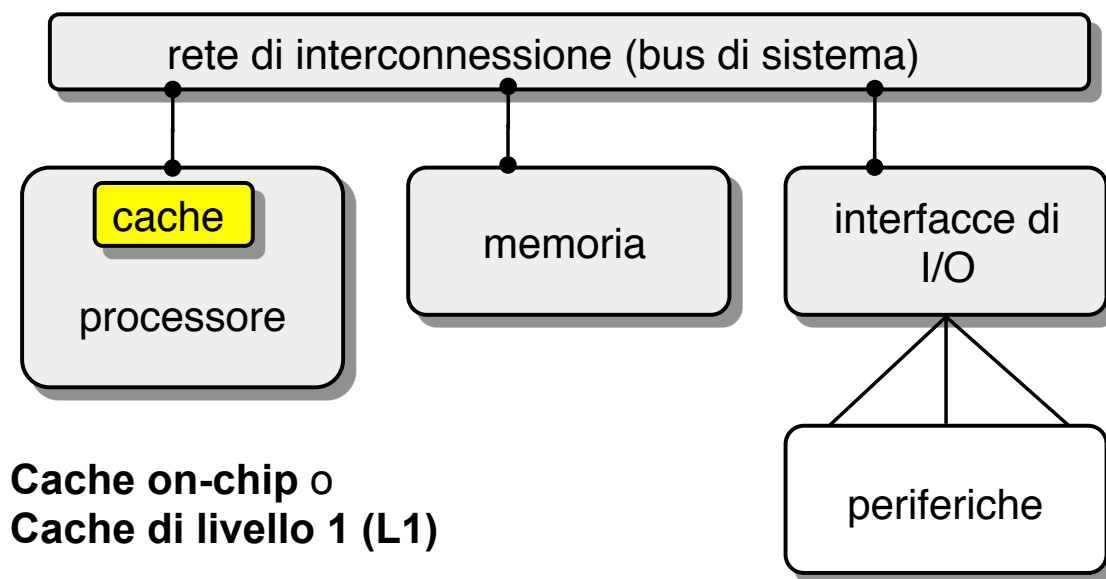
FixL

70

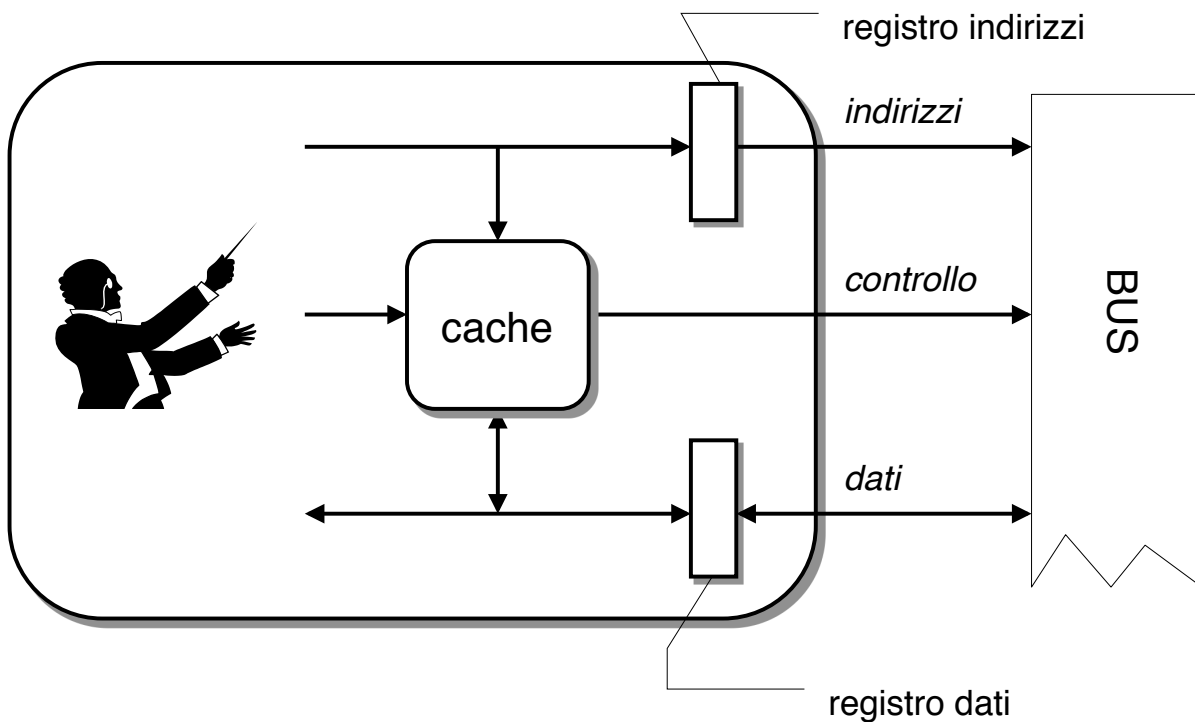
Memoria Cache



Cache



Esempio di organizzazione



23 December 2015

FixL

73

Importanza della cache

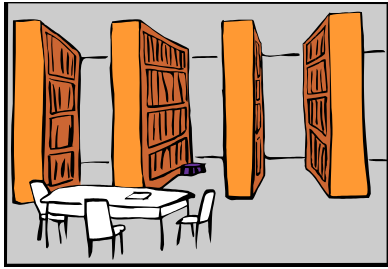
- La presenza di una memoria cache efficiente (che, cioè, risponde “al posto della RAM” nel 95-98% dei casi) è *fondamentale* per le prestazioni del sistema computer
- La presenza della cache complica notevolmente l’architettura di un sistema di elaborazione
 - Problema del rimpiazzo
 - Problema della scrittura (coerenza dei dati)
- La cache ha una dimensione limitata
 - Il 50% del consumo di potenza di un processore è dovuto alla cache

23 December 2015

FixL

74

Principio di località (I)



PRINCIPIO DI LOCALITÀ

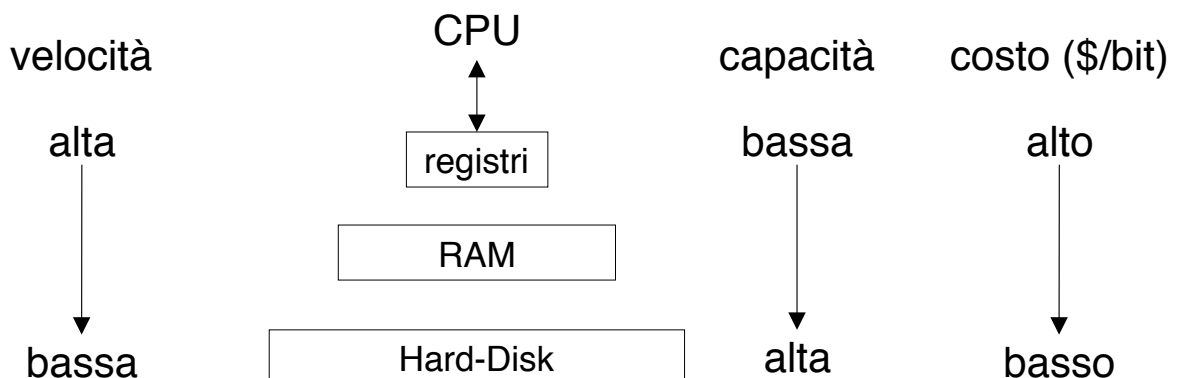
Ricerca sugli autori italiani del '900

- **Località temporale.** Se lo studente ha portato un libro sulla scrivania per guardarlo, probabilmente ne avrà nuovamente bisogno non molto tempo dopo.
- **Località spaziale.** Se uno studente ha preso un testo su Pasolini, è probabile che gli servirà un testo su Pavese (che nello scaffale si trova vicino al precedente)

Gerarchie di memoria

PRINCIPIO DI LOCALITÀ

- **Località temporale.** È probabile che un oggetto (dato, istruzione) a cui si è fatto riferimento venga nuovamente richiesto in tempi brevi.
- **Località spaziale.** È probabile che oggetti che si trovano vicini ad un oggetto a cui si è fatto riferimento vengano richiesti in tempi brevi.



Tecnologie

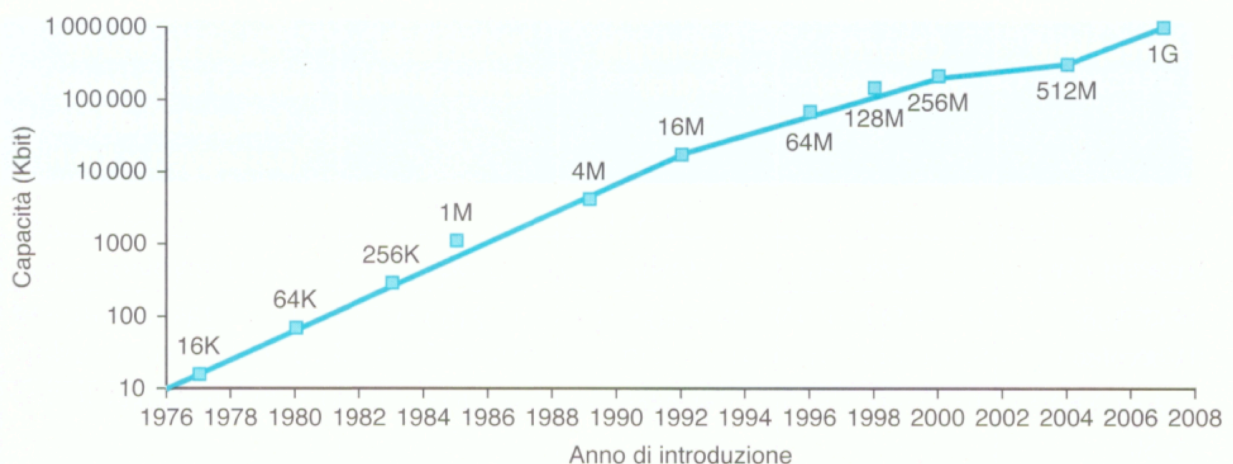
(*) relative per unità di costo

Anno	Tecnologia	Prestazioni(*)
1951	Tubo a vuoto (valvola)	1
1965	Transistor	35
1975	Circuito integrato	900
1995	Circuito integrato a grandissima scala di integrazione (VLSI)	2 400 000
2005	Circuito integrato a scala di integrazione estremamente grande (ULSI)	6 200 000 000

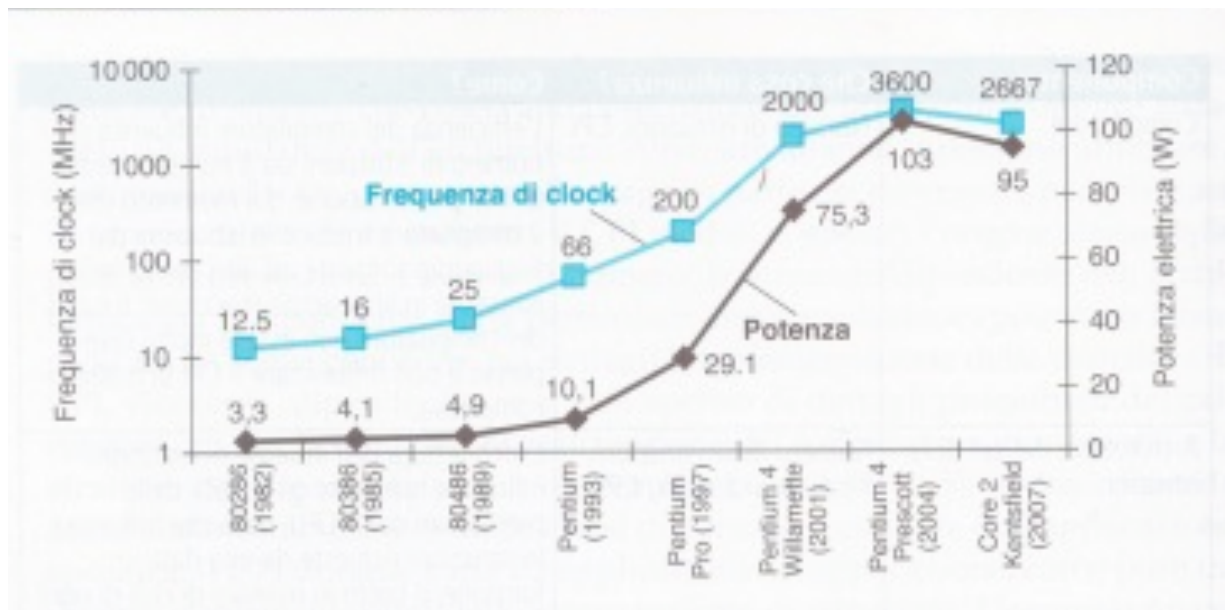
Legge di Moore: la capacità dei circuiti integrati in termini di transistor raddoppia ogni 18-24 mesi

Prestazioni memoria

Capacità di un chip di DRAM in funzione del tempo



La barriera dell'energia (I)



23 December 2015

FixL

79

La barriera dell'energia

- La potenza e la frequenza sono correlate: $W = C \times V^2 \times f$
- Come è stato possibile aumentare f di un fattore 1000 mentre W di un fattore 30?
 - Riducendo la tensione (da 5 V a 1V)
- Sono stati raggiunti due limiti
 - Quello della potenza dissipabile dai sistemi di raffreddamento montati sui microprocessori
 - Quello della tensione minima
- Lo sviluppo del Pentium 4 è stato abbandonato a partire dal modello Prescott (2004) e si è passati ai **multi-core**

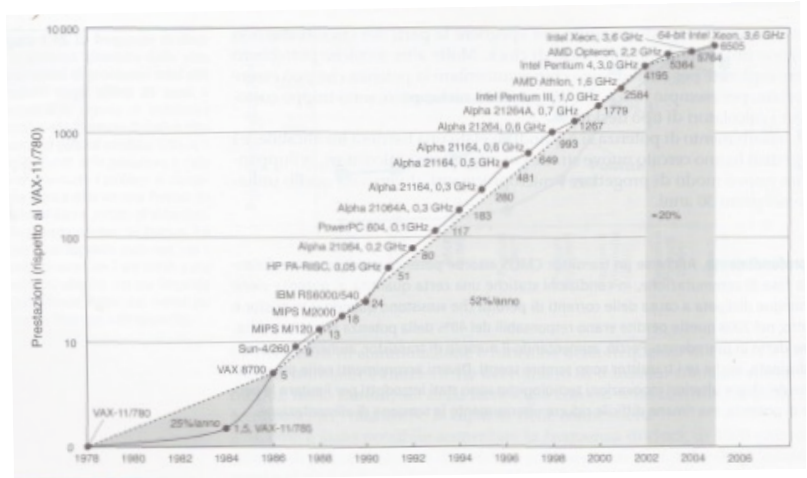
23 December 2015

FixL

80

Il passaggio da mono- a multi-processore

- L'assorbimento di potenza ha ridotto i miglioramenti
- Dal 2002 il miglioramento è sceso da 1.5 a 1.2 per anno
- Dal **2006** tutti i produttori di desktop e server hanno iniziato a distribuire microprocessori con più processori (**multicore**)



Multi-core

- **Potenza assorbita**
- I produttori hanno pianificato di raddoppiare il numero di core ad ogni generazione
- Svantaggio
 - Ad ogni generazione il sw va messo a punto per sfruttare i nuovi core
 - La programmazione parallela è difficile
 - Il programma deve essere **corretto** e **veloce**
 - **Bilanciare il carico** e **ridurre le comunicazioni** tra i core

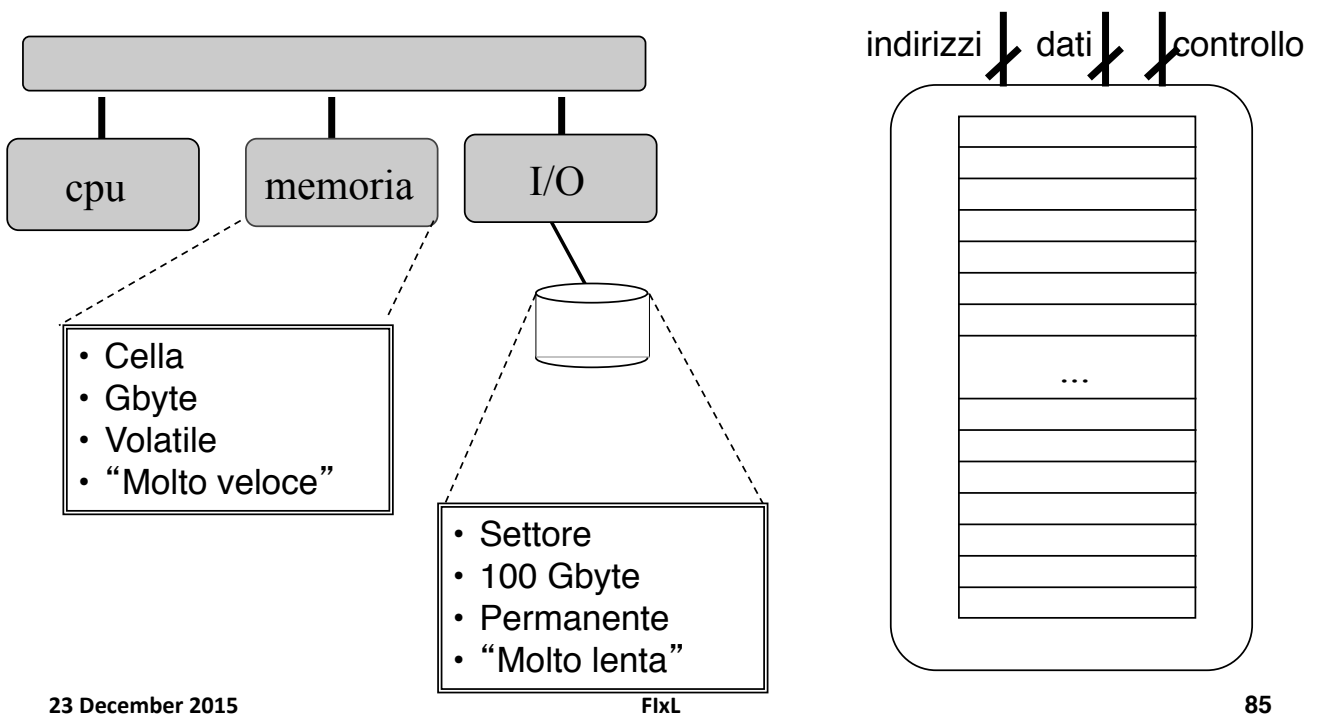
	AMD Opteron x4 (Barcelona)	Intel Nehalem	IBM Power 6	Sun Ultra SAPRC T2 (Niagara 2)
Core per chip	4	4	2	8
Frequenza	2.5 GHz	2.5 GHz	4.7 GHz	1.4 GHz
Potenza	120W	100W	100W	94W

DISPOSITIVI DI I/O

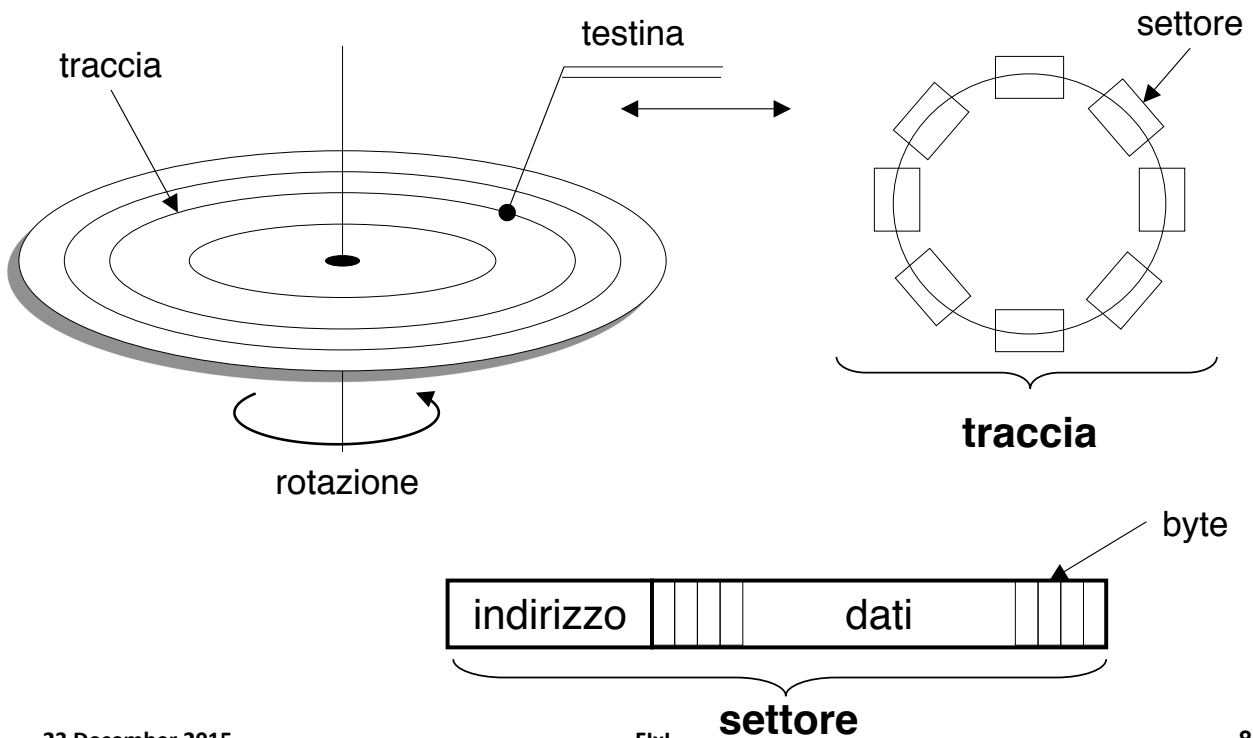
Volatilità della memoria

- La memoria RAM immagazzina le informazioni sotto forma di *carica di un condensatore*
- In assenza di tensione, le informazioni vanno perse
 - *Volatilità della memoria*
- Sono necessari dispositivi che mantengano l'informazione anche in assenza di tensione (memorie *di massa, permanenti, non-volatili*)
 - Il fenomeno fisico con il quale l'informazione viene memorizzata sarà necessariamente diverso da quello della RAM
 - Floppy disk, hard disk: *dischi magnetici*
 - CD-ROM, DVD *dischi ottici*
 - Compact Flash *memorie a stato solido*

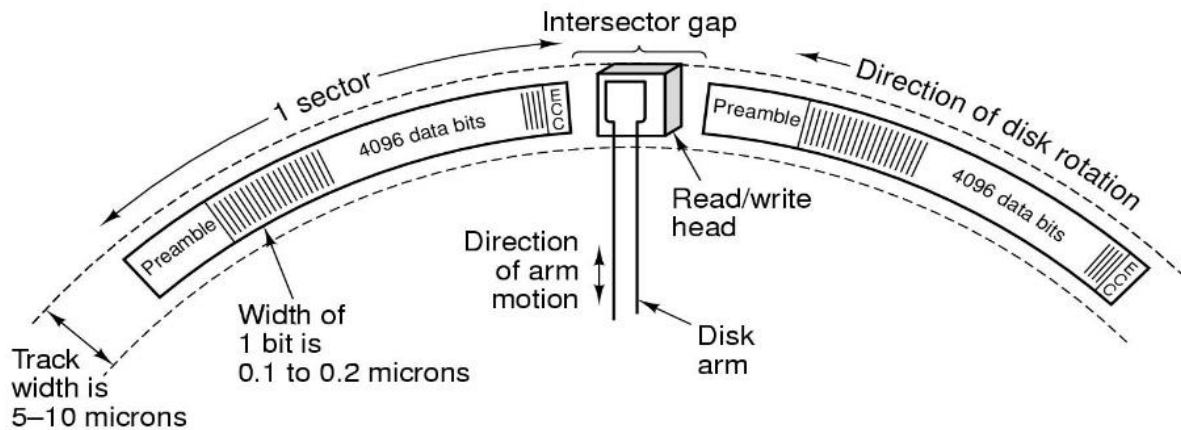
Struttura logica del disco



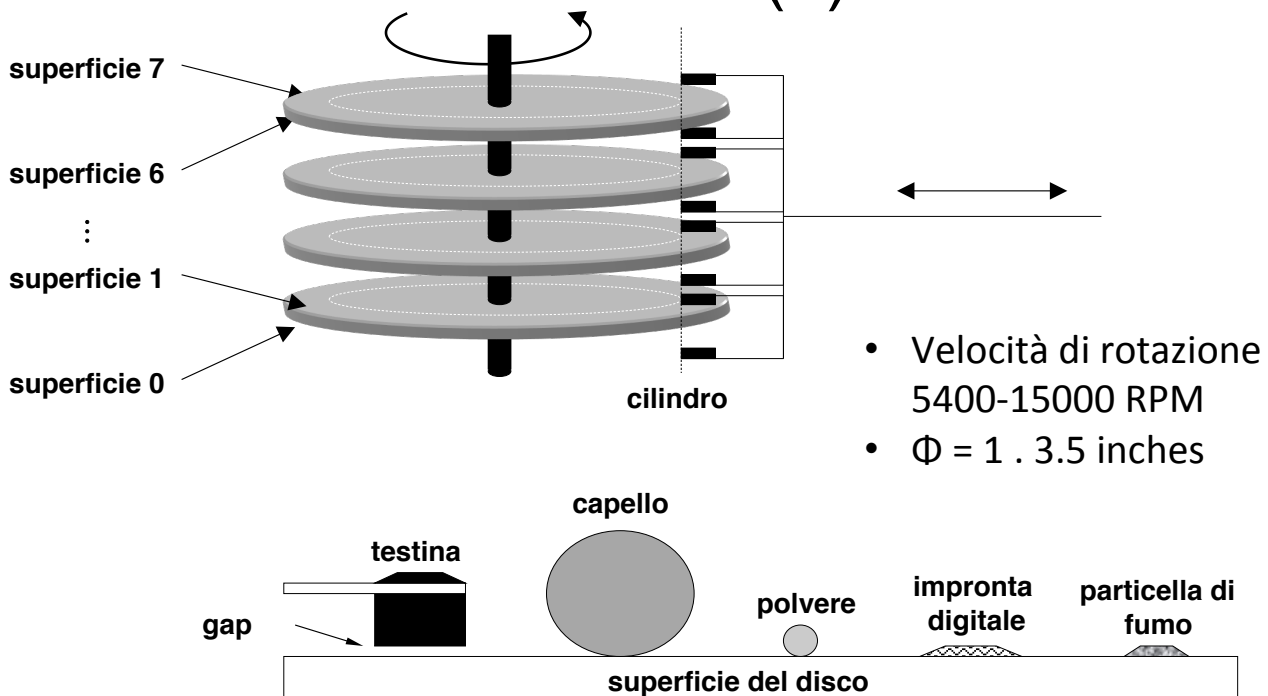
Struttura fisica del disco (I)



Struttura fisica: una vista piu' accurata



Struttura fisica del disco (II)



Tempo di accesso al disco

- Il **tempo di accesso** ha tre componenti
 - **tempo di ricerca**: tempo per posizionare la testina sulla traccia (**variabile**)
 - **latenza**: tempo necessario affinché il settore cominci a ruotare sotto la testina (**variabile**)
 - **tempo di trasferimento**: tempo necessario affinché l'intero settore passi sotto la testina (**fisso**)
- Il tempo di accesso è **variabile** e dipende dal settore scelto e dallo stato corrente del disco

Tempo di accesso al disco: Esempio

- **Velocità di rotazione** = 7200 giri/min \rightarrow 8.33 ms/giro
- **Tempo di spostamento** (del braccio sulla traccia adiacente) = 0.02 ms
- **Numero tracce** = 1000
- **Settori/traccia** = 50
- **Byte/settore** = 512

	Tempo di accesso		
	Migliore (ms)	Peggiora (ms)	Medio (ms)
<i>Tempo di ricerca</i>	0	$999 \times 0.02 = 19.98$	$500 \times 0.02 = 10$
<i>Latenza</i>	0	8.33	$8.33/2 = 4.17$
<i>Trasferimento</i>	$8.33/50 = 0.17$	$8.33/50 = 0.17$	$8.33/50 = 0.17$
<i>Totale</i>	0.17	28.48	14.34

Memoria RAM vs. Hard Disk

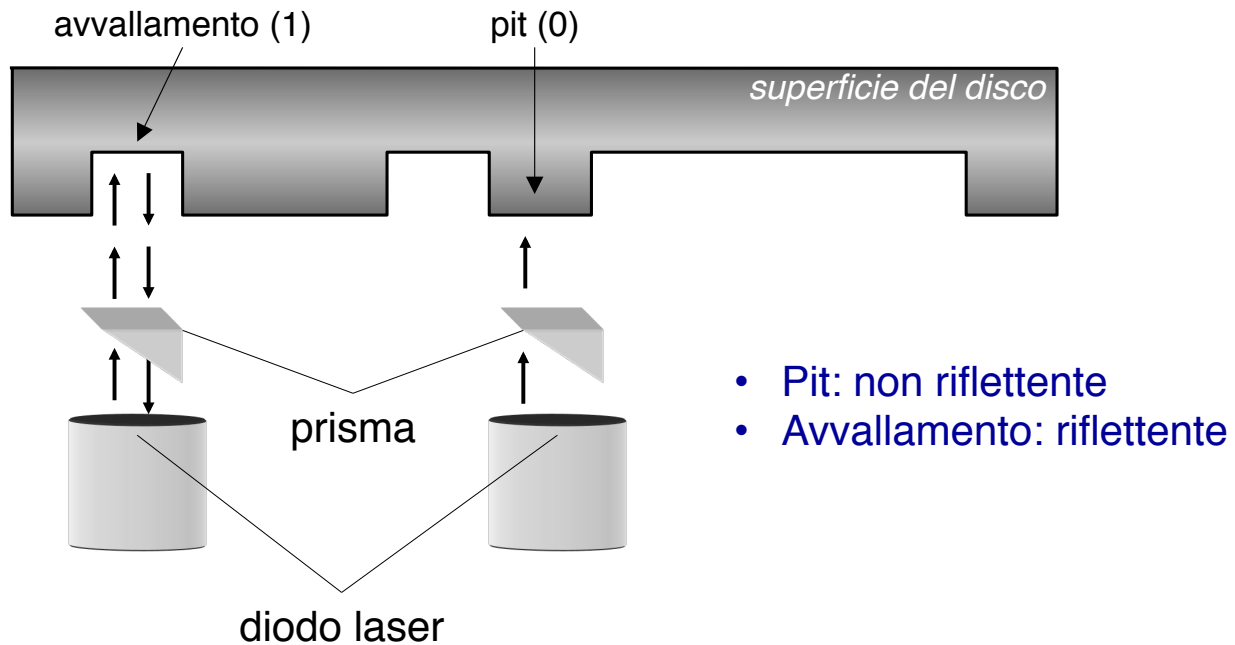
	MEMORIA RAM	DISCO
<i>Persistente/volatile</i>	Volatile	Persistente
<i>Unità di accesso</i>	cella (8, 16, 32 bit)	settore (512 byte)
<i>Capacità*</i>	alcuni Gbyte	Centinaia di Gbyte
<i>Tempo di accesso</i>	30 - 70 ns	0.2 - 20 ms
<i>Costo</i>	150 - 200 €/Gbyte	0.6 - 1.5 €/Gbyte

* In un PC

RAM vs Hard Disk - confronto

- La RAM è volatile mentre l'HD è persistente
- Il tempo di accesso basso (decine di ns)
- Nelle RAM il costo

CDROM



23 December 2015

FixL

93

CDROM e DVD

- L'informazione è memorizzata sfruttando il medesimo principio
- Rispetto al CDROM, un DVD ha tracce più vicine ed avvallamenti/pit di dimensione minore
 - Maggior capacità a parità di superficie
 - Necessaria maggior precisione nella lettura
- CDROM: 650/800 Mbyte (74-90m di musica)
- DVD: 4.7/9.6 Gbyte (1,5h di filmato hi-def)

23 December 2015

FixL

94

Memorie Compact Flash

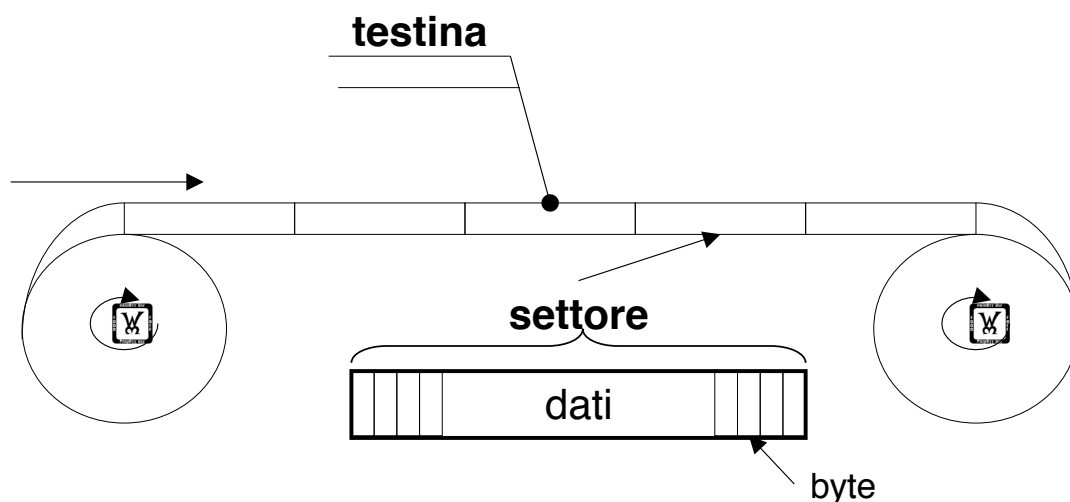
- Strutturalmente simili a memorie ROM, con possibilità di *cancellare* blocchi di memoria (Flash-EPROM)
- Connesse tramite bus plug-and-play USB (Universal Serial Bus)
- Capacita' : 128MB - 4GB (oggi)
- Tempi di accesso paragonabili a quelli di un HD
- Costo contenuto (25 euro/GB)



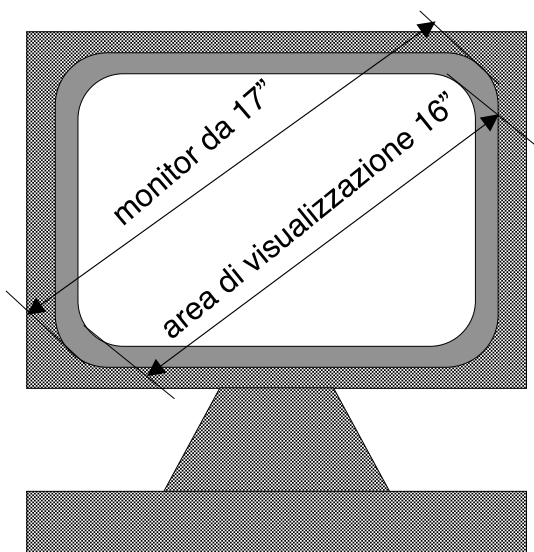
Dispositivi ad Accesso Sequenziale

Un settore non è più identificato da un indirizzo

Lettura/scrittura: i settori sono acceduti in sequenza



Schermo



Dimensioni monitor (pollici)	Area di visualizzazione (pollici)
15	14
17	16
21	20

PARAMETRI

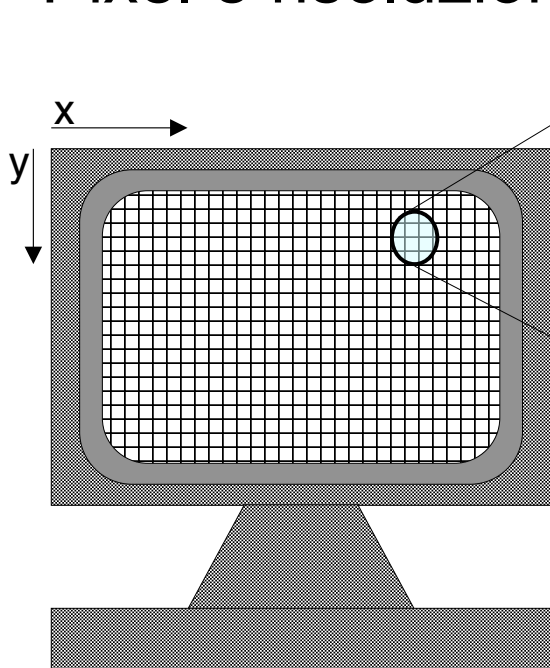
- **Nitidezza** (dimensione dei punti, risoluzione, velocità di aggiornamento)
- **Tipo di schermo** (CRT, LCD)
- **Potenzialità di visualizzazione** (SVGA, XVGA)

23 December 2015

FixL

97

Pixel e risoluzione



Dimensione dei punti (DOT-PITCH)

Distanza tra i centri di due pixel adiacenti

Esempio: 0,28: 28 centesimi di mm

alternativamente

N. di pixel per unità di misura lineare (72-100 dpi, dots per inch) (risoluzione)

Quantità di punti che possono essere visualizzati sullo schermo del monitor

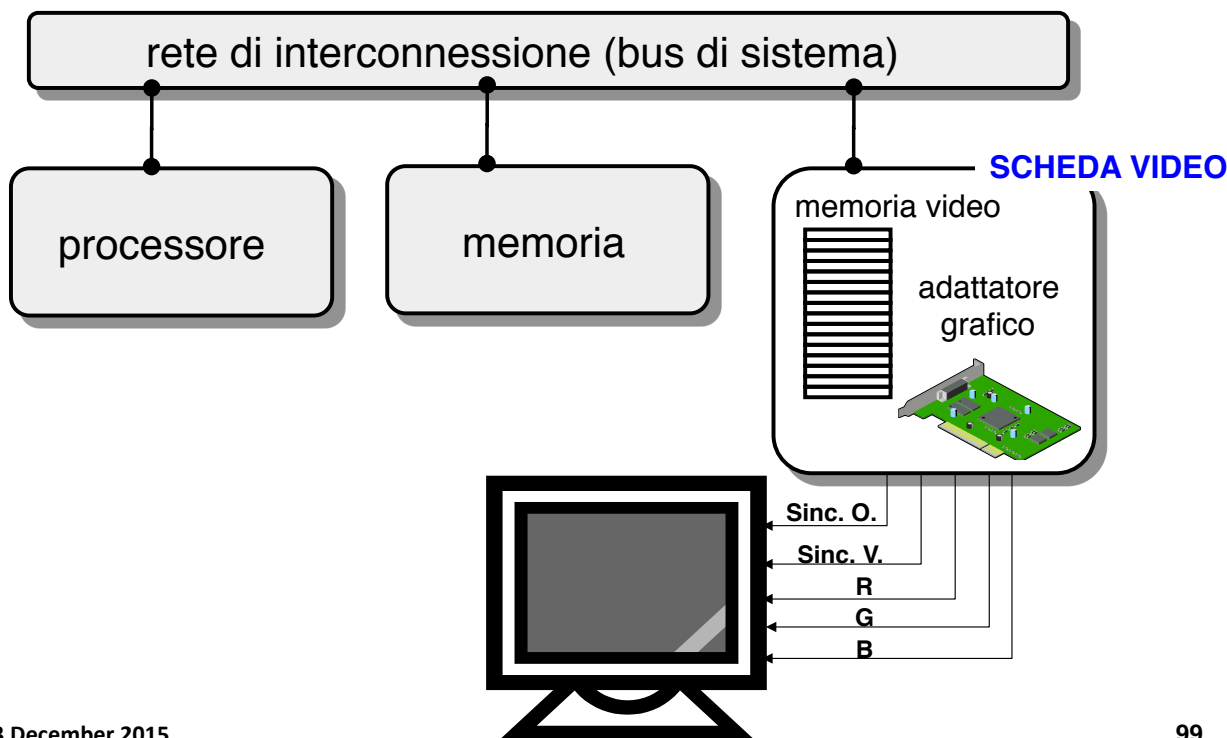
Esempio: 640x480 (VGA), 800x600 (SVGA), 1024x768 pixel (XGA)

23 December 2015

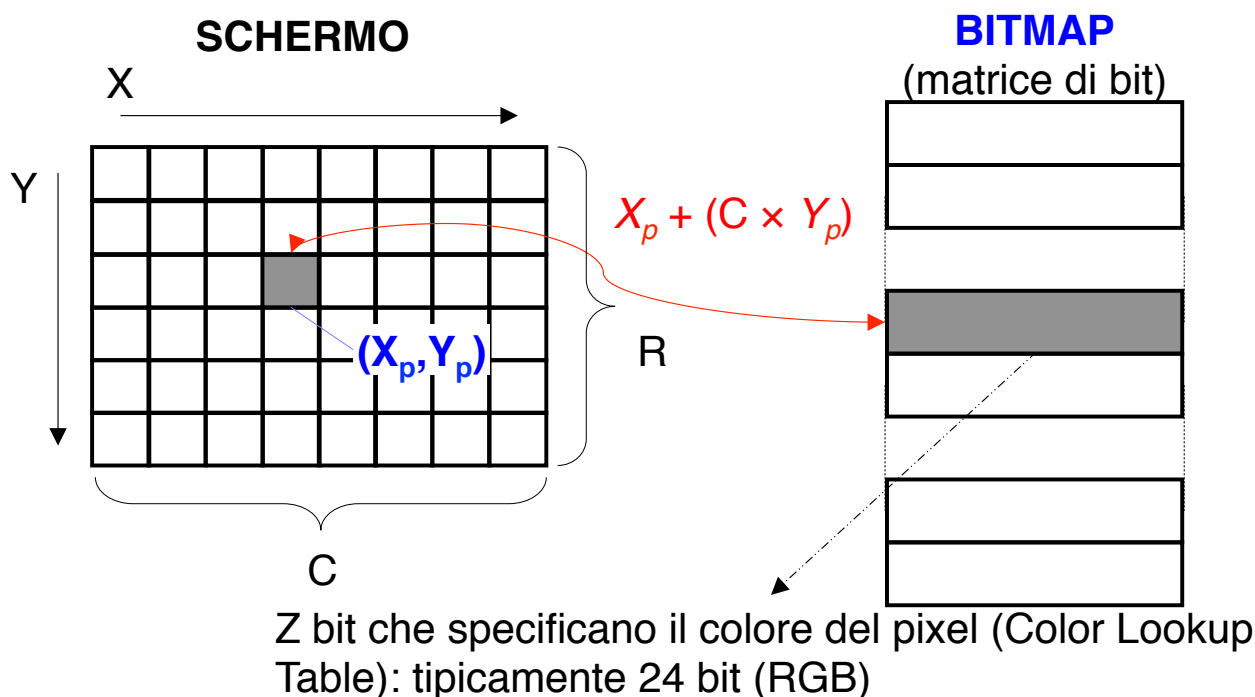
FixL

98

Scheda Video (semplificata)



Memoria Video (semplificata)



Esempio: dimensioni di una bitmap

C = Numero dei colori

P = Numero di pixel

N = bit di indirizzo per la Memoria Video

$$N = \lceil \log_2 P \rceil$$

Z = dimensione di una parola di Memoria Video

$$Z = \lceil \log_2 C \rceil$$

L = Numero locazioni della Memoria Video = 2^N ($L \geq P$)

ESEMPIO

Modalità Grafica: 640 × 480; COLORI: 256 →

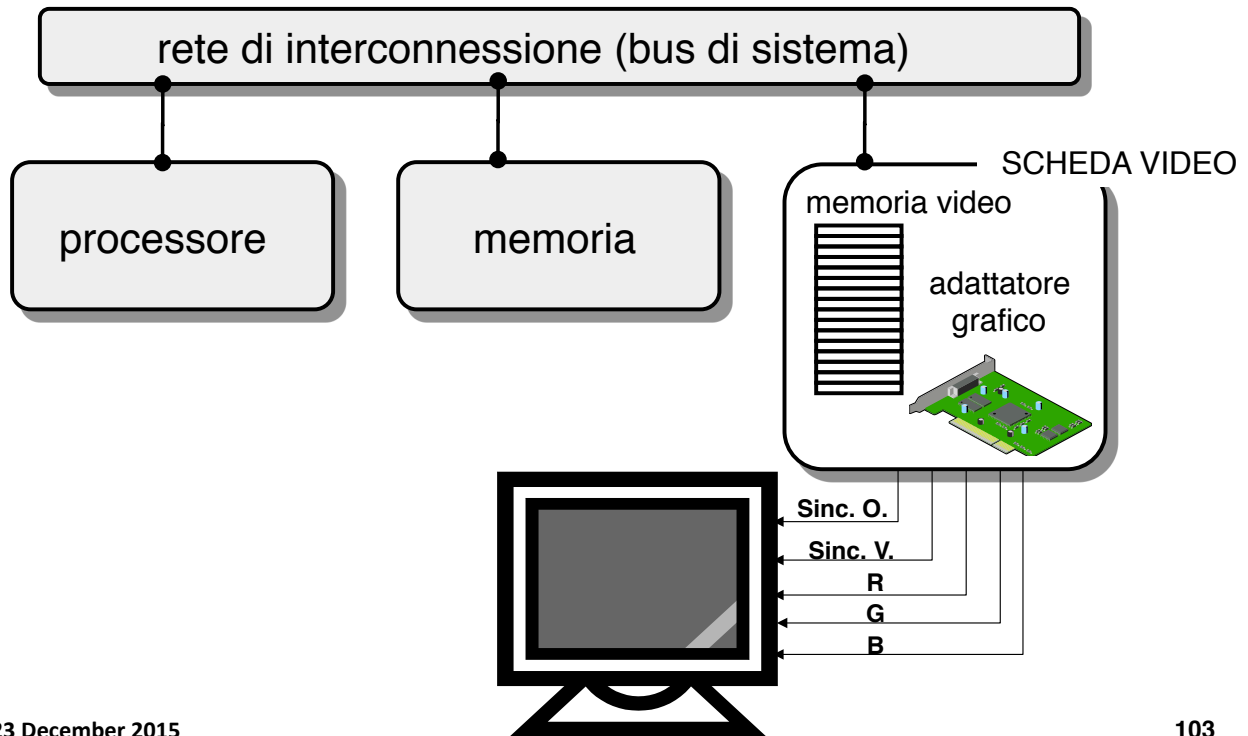
$P = 307200$

$N = 19$; $Z = 8$ → Bitmap di 512 Kbyte

Adattatore Grafico (semplificato)

- L'AG legge ad intervalli di tempo regolari la bitmap e la invia al terminale alla frequenza di aggiornamento stabilita
 - Una volta che ha letto il contenuto di una locazione associata ad un certo pixel, l'AG compie le seguenti azioni:
 - Considera i bit letti come la codifica del colore che dovrà caratterizzare il pixel
 - Genera tre segnali corrispondenti ai tre colori fondamentali (R,G,B)
 - Trasmette tali segnali al monitor tramite i fili R, G e B
- La bitmap è contenuta nel **frame buffer**
- La difficoltà nella realizzazione dei sistemi grafici sta nel fatto che l'occhio umano è molto sensibile anche a minimi cambiamenti sullo schermo

Pixel logici e pixel fisici

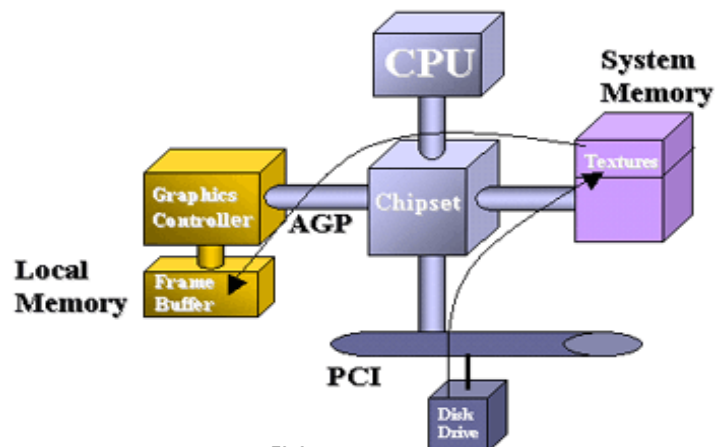


23 December 2015

103

Schede Video AGP

- L'architettura **AGP** (*Accelerated Graphics Port*) permette uno scambio diretto delle informazioni tra CPU e memoria centrale e scheda grafica
- In questa maniera il bus standard (in questo caso il PCI) non entra nella comunicazione e può essere usato in parallelo
- Incremento della velocità di comunicazione tra chipset e scheda grafica; il limite di comunicazione del bus PCI è 132 MB/s, lo standard AGP 3.0 8x arriva a 2.1 GB/s di picco

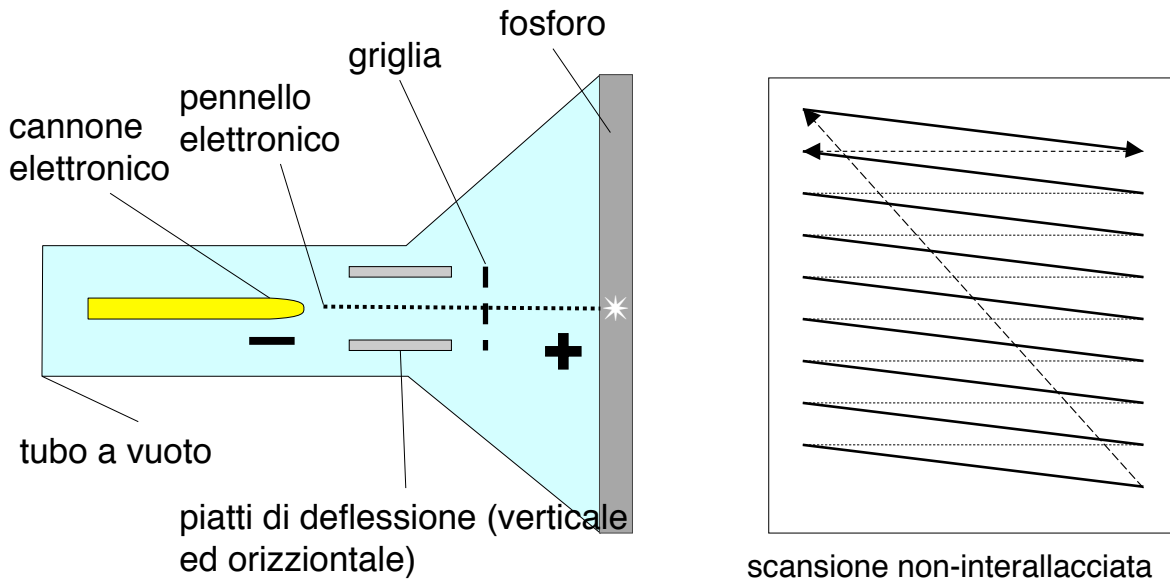


23 December 2015

FixL

104

CRT



Frequenza di aggiornamento (verticale): numero di volte al secondo che i pixel vengono aggiornati per conservarne la luminosità (es.: 50-100 volte il secondo)

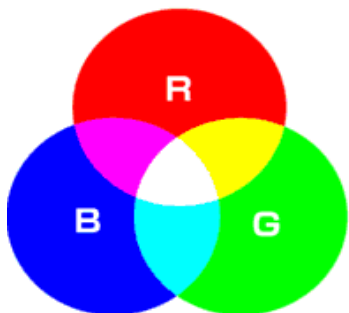
23 December 2015

FixL

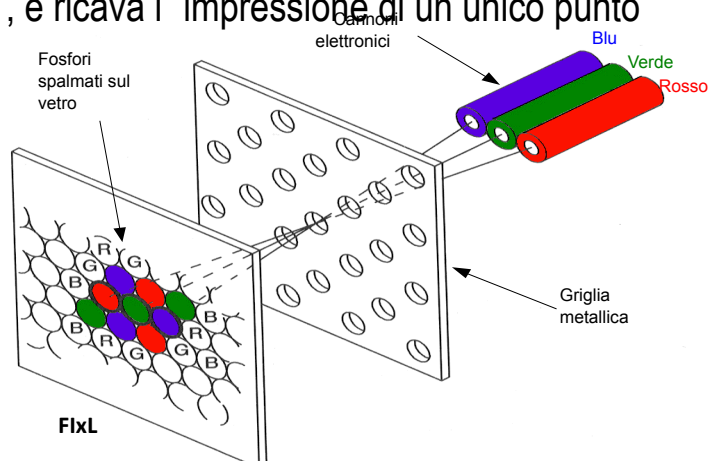
105

CRT a colori

- Tre cannoni luminosi, uno per ognuno dei colori fondamentali (R,G,B)
- In realtà, un pixel è un agglomerato di 3 puntini molto vicini dei tre colori fondamentali
 - L'occhio umano "sintetizza", e ricava l'impressione di un unico punto colorato



23 December 2015



FixL

Schermo

Frequenza di aggiornamento/refresh (verticale): numero di volte al secondo che i pixel vengono aggiornati per conservarne la luminosità (50 – 120 Hz)

Frequenza orizzontale: numero di righe orizzontali che possono essere dipinte in un secondo (30-100 KHz)

Banda video: il numero di pixel contigui che possono essere colorati con colori diversi in un secondo (ovvero la frequenza con cui è possibile cambiare l'intensità del pennello). Valore tipico 200 MHz

Liquid Cristal Display

- Display a schermo piatto
- Svareti ambiti di applicazioni
 - TV, Laptop, monitor
- Dimensioni e risoluzione
 - decine – centinaia di pollici
 - 640 x 480 – 1920 x 1080
- Vantaggi
 - basso consumo
 - Retroilluminazione
 - Luce riflessa per strumentazione portatile
 - Tempo di vita: 50000 ore

LCD

▪ Storia

- Cristalli liquidi scoperti nel 1888 da un botanico austriaco, Friedrich Reinitzer
- Nel 1889, il fisico tedesco Otto Lehmann studio il fenomeno
- Primi display nel 1965-1968 nel Centro di Ricerche Sarnoff, RCA, USA
- Attualmente hanno sostituito i CRT

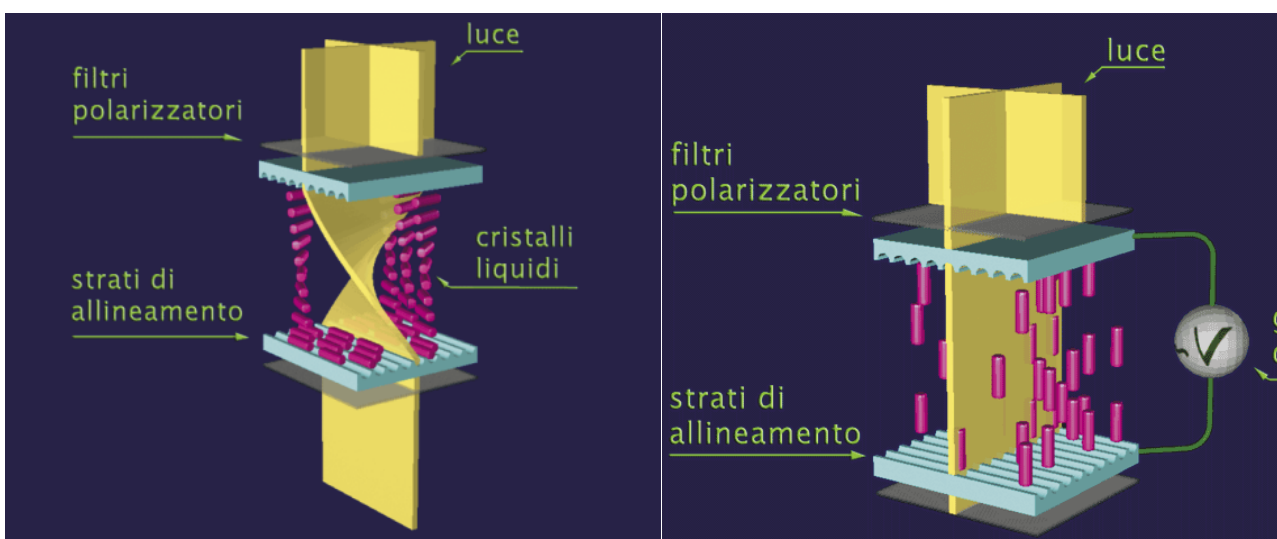
23 December 2015

FixL

109

Liquid Crystal Display (LCD)

- LCD funziona come una valvola di luce



23 December 2015

FixL

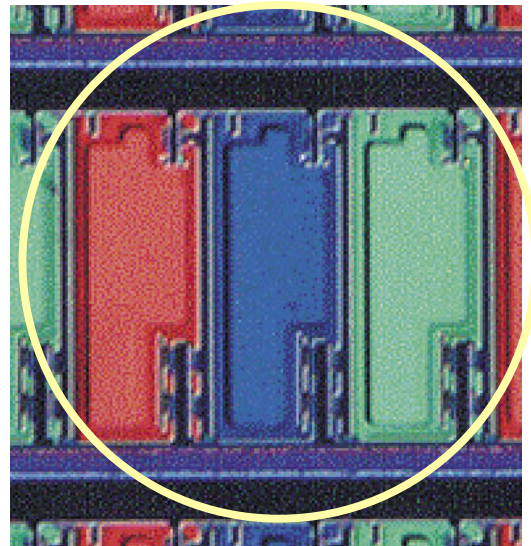
110

LCD – tipi di schermi

- Retroilluminati
 - Alta luminosità ma alti consumi
 - Leggibilità buona in condizioni di scarsa luce ambientale ma cattiva in condizioni di forte luce ambientale → adatti per interni
- Schermi riflessivi
 - Usano la luce ambientale (schermo posto sul retro)
 - Minore contrasto (la luce passa due volte per il filtro)
 - Bassi consumi (alimentazione con batteria fotovoltaica)
 - Buona leggibilità con forte illuminazione ambientale → adatto per esterni

LCD a colori

- Si aggiungono 3 filtri colorati a ciascun pixel
 - La luminanza di ciascun pixel è ottenuta variando la tensione
 - 64 livelli per ciascun colore (6 bit)
- Applicazioni multimediali
 - Fedeltà cromatica
 - Fino a 256 livelli (8 bit) per ciascun colore primario
 - Equivalente a True Color (24 bit) di CRT



LCD

- **LCD a matrice passiva**
 - Sistema di pilotaggio lento
 - Problema del ghosting
- **LCD a matrice attiva**
 - Sistema di pilotaggio veloce
 - Elimina il problema del ghosting
 - Leggeri
 - Più costosi da produrre
 - Maggior numero di transistor
 - Transistor “guasti”

Dispositivi di I/O

- Memorie di massa (dischi, CD-ROM, cassette)
 - *rappresentano le informazioni in forma binaria*
 - accesso diretto
 - hard-disk, floppy-disk, CD-ROM
 - accesso sequenziale
 - cassette
- Tastiera, mouse (in), schermi, stampanti (out)
 - *Convertono le informazioni da una forma ad un'altra*
 - Utente umano -> computer (ingresso)
 - Computer -> utente umano (uscita)

Tastiera (semplificata)

- Sotto i tasti c'è una *matrice* di contatti elettrici
- La pressione di un tasto viene ricondotta ad una coppia (riga,colonna)
- Un *programma* (parte del sistema operativo) si occupa di tradurre l'informazione (riga, colonna) in un codice ASCII esteso
- Tale programma può essere configurato per interagire con tastiere *diverse* (italiana, USA, etc.)

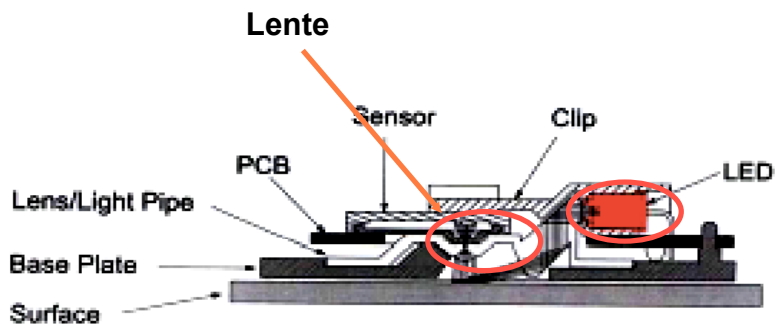
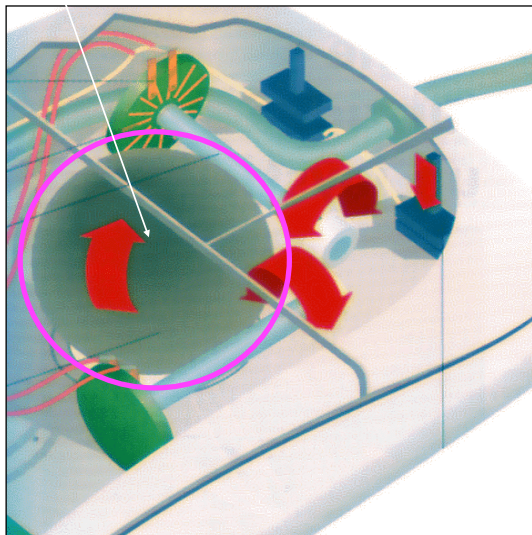


Mouse

- Consente di associare il movimento di un oggetto su un piano al movimento di un puntatore sullo schermo
 - Utile in ambienti grafici
- Il Mouse converte in impulsi elettrici
 - Il **moto** di una sfera (mouse **meccanici**)
 - La differenza di immagini successive (mouse **ottici**)

Mouse meccanico ed ottico

Pallina di gomma



23 December 2015

FixL

117

Stampanti

Differiscono per:

- Tecnologia di stampa
 - Getto di inchiostro
 - Laser
- Capacità cromatica
 - Bianco e nero
 - Colore
- Velocità di stampa
 - N. di pagine al minuto
- Risoluzione
 - 600-4800 dpi
- Capacità di eseguire compiti complessi
 - Stampanti "PostScript"



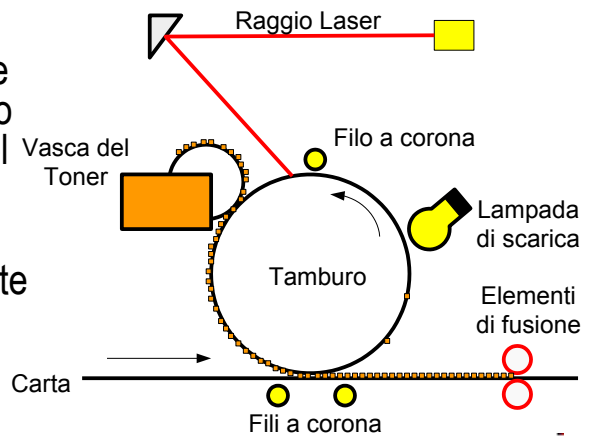
23 December 2015

FixL

118

Stampanti laser

- Un raggio laser, tramite specchi che lo deflettono, scandisce per righe successive un tamburo e ne carica elettricamente solo i punti che dovranno corrispondere ai pixel da stampare
- Contemporaneamente un foglio di carta passa vicino ad un filo percorso da corrente e si carica elettrostaticamente
- Il tamburo, ruotando, *pesca* toner da una vaschetta
- Il toner, a carica opposta al tamburo, si fissa ad esso solo nei punti caricati dal raggio laser
- Quando la carta passa a contatto con il tamburo, dato che la carta è molto più carica, il toner viene attirato dalla carta



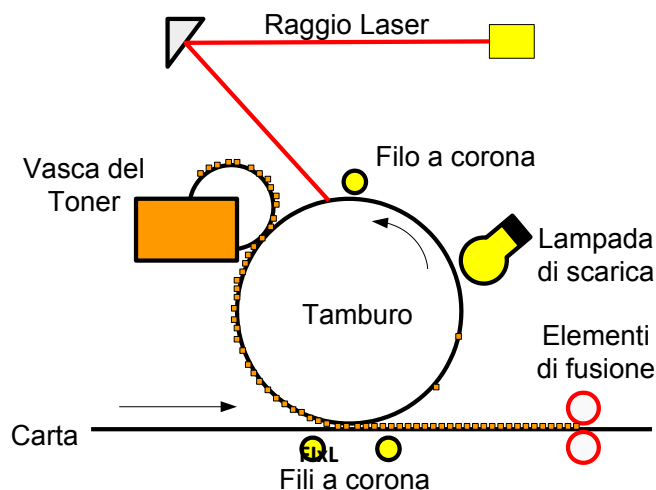
23 December 2015

FixL

119

Stampanti laser

- Il tamburo, continuando a ruotare, passa vicino alla *lampada di scarica* che lo scarica completamente
- La carta passa invece da sistema di fusione che, con una combinazione di calore e pressione, fissa in permanenza, fondendolo, il toner alla carta



23 December 2015

120

Stampanti Inkjet

- Al posto del toner, ci sono *cartucce* di inchiostro
- Le cartucce sono montate su una testina ancorata ad un braccio mobile che scandisce orizzontalmente il foglio
- Viene rilasciato inchiostro su ogni pixel da annerire
- Il foglio viene fatto avanzare per stampare nuove righe



23 December 2015

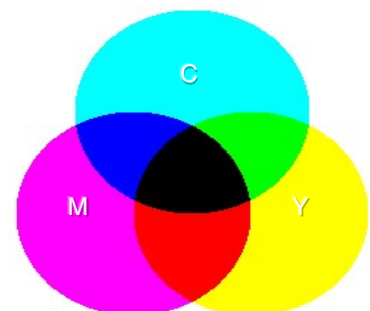
FixL



121

Stampanti a colori

- Nelle stampanti a colori, ogni colore e' rappresentato mescolando quattro colori
 - **Cyan, Magenta, Yellow, Black (CMYK)**
- In realta', basterebbero i primi tre
 - Nero usato per convenienza e precisione
- Il processo appena descritto viene ripetuto 4 volte, ogni volta con un inchiostro di colore differente
- Vale sia per stampanti laser che per stampanti inkjet



23 December 2015

FixL

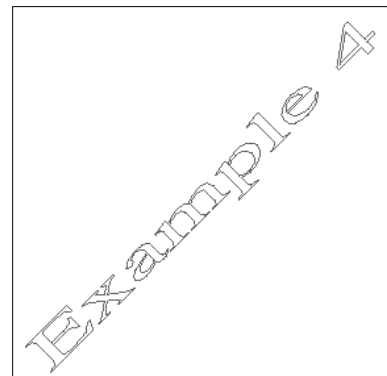
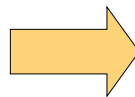
Il processo di stampa

- Come si passa da una *pagina* (ad esempio) di testo *all'insieme di operazioni* che devono essere effettuate sulla stampante per ottenere la pagina stampata?
 - Il software che elabora il documento (e.g. Word) interagisce con il **driver** della stampante
 - Tale interazione avviene secondo criteri *standard*, che **non** dipendono dal modello di stampante
 - Il driver invia alla stampante i comandi nella maniera ad essa comprensibile
 - Tale interazione **dipende** dal modello di stampante. Infatti stampanti *diverse* hanno driver *diversi*

Stampanti PostScript

- Hanno dei processori (RIP, Raster Image Processor) che eseguono comandi in linguaggio PostScript

```
/Times-Roman findfont
32 scalefont
setfont
100 200 translate
45 rotate
2 1 scale
newpath
0 0 moveto
(Example 4) true charpath
0.5 setlinewidth
0.4 setgray
stroke
```

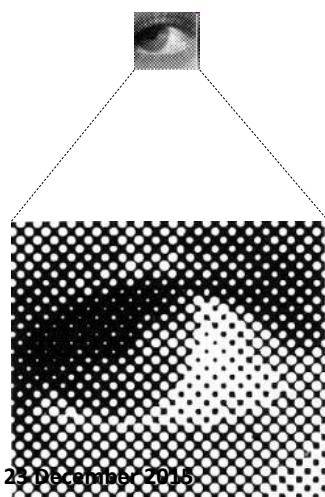


- Per inviare una pagina di testo, invece che convertirla in una matrice di punti posso darle una descrizione molto più sintetica in PS e lasciare che il RIP la interpreti

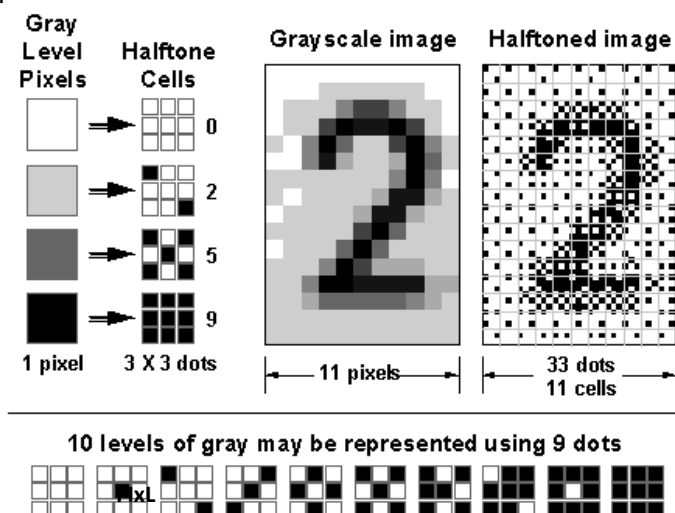
- L'immagine viene creata **all'interno della stampante**

Retinatura (Halftoning)

- Una stampante b/n (a colori) ha inchiostri di un colore (4 colori)
- Per realizzare (e.g.) il grigio o il magenta chiaro, si usa la *retinatura*
 - Non tutti i pixel adiacenti vengono colorati
 - Il cervello integra nello spazio punti bianchi e colorati vicini



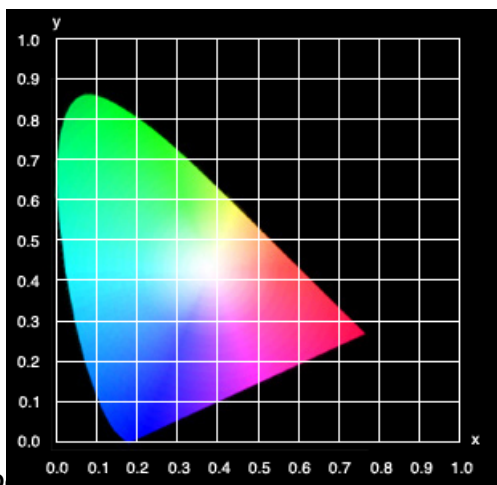
23 Dicembre 2015



125

Gamut

- Ogni dispositivo e' in grado di rappresentare un sottoinsieme *finito* dei colori visibili, detto *gamut*
- Dispositivi diversi hanno gamut diversi



23 D

Spettro di luce visibile



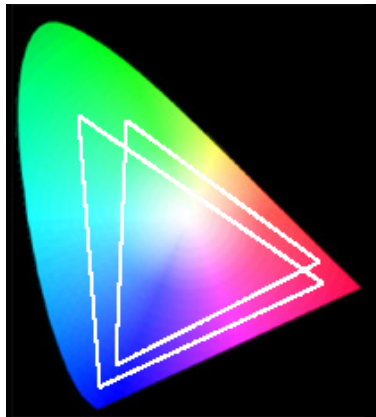
Diagramma ufficiale della CIE (Commission Internationale de l'Eclairage), 1931, che rappresenta I colori visibili in coordinate assolute (xyY)

FIXL

126

Gamut (2)

- Il gamut di dispositivi a sintesi *additiva* (monitor) e sottrattiva (stampanti) e' notevolmente differente
 - Non tutti i colori visibili sono stampabili e viceversa
 - Monitor diversi in cui i tre fosfori sono illuminati con una data intensita' mostrano colori diversi



23 December 2015



FixL

127