

Calcolatori Elettronici

È proibita qualunque riproduzione di questo fascicolo, anche parziale, in libri,

pubblicazioni anche telematiche, cd, dvd, siti web e ogni altra forma di pubblicazione

senza il consenso scritto dell'autore.

In particolare, è proibita la vendita di questo fascicolo o di parti di esso in qualunque forma.

Calcolatori Elettronici

02/03/12

Lezione I

E-MAIL: torlone@dia.uniroma3.it

sito: <http://torlone.dia.uniroma3.it/calcolatori>

RICEVIMENTO: venerdì 11:00 - 13:00 II piano stanza 209

LIBRO: A.S Tanenbaum: Structured Computer Organization 5th ed

- EVOLUZIONE DEI CALCOLATORI -

MACCHINA DI BABBAGE:

- ↳ Macchine differenziali
 - algoritmo fisso
 - output su piastra di rame
- ↳ Macchine analitiche:
 - prima macchina programmabile
 - memoria: 1000 x 50 cifre decimali
 - mecano CPU
 - I/O schede perforate

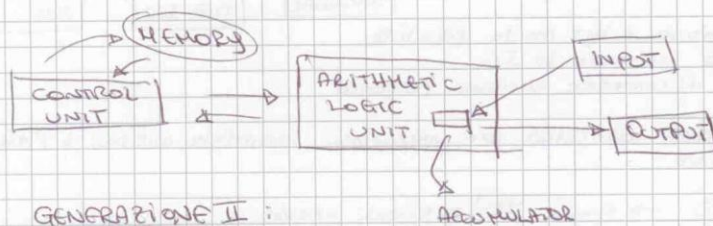
GENERAZIONE 0

- KRYAD ZUSE: macchina a relè
- JOHN ATANASOFF: aritmetica binaria
- HOWARD AIKEN: versione a relè della macchina di Babbage

GENERAZIONE I:

- COLOSSUS: decifratura del codice Enigma - progetto tenuto segreto
- UNIVAC
- ENIAC

MACCHINA DI VON NEUMANN



GENERAZIONE II:

- tecnologia a transistori
- TX0 e TX2 macchine sperimentali
- DEC → produce PDP-1 → memoria: 4K parole di 18 bit → tempo di ciclo 5 μsec → prod. di massa → 512 x 512

SUPER COMPUTER : → grosse capacità di calcolo → rapidità
• parallelismo all'interno della CPU • architettura molto sofisticate
• macchine dedicate al number crunching

GENERAZIONE III

- ARCHITETTURA MIN: processori dedicati alla gestione I/O; microprogrammazione, unità veloci floating-point
- SISTEMI OPERATIVI:
 - virtualizzazione delle risorse
 - multi programmazione → esecuzione in contemporanea
 - memoria virtuale → rimuove le limitazioni dovute alle dimensioni della memoria fisica

GENERAZIONE IV (PC)

- ARCHITETTURA A BUS
 - PAROLE A ISTRUZIONI A 16 bit
- 1980 nasa e' IBM

Bim acquisto PC dovuto al crollo dei costi ed enorme espansione dell'utente

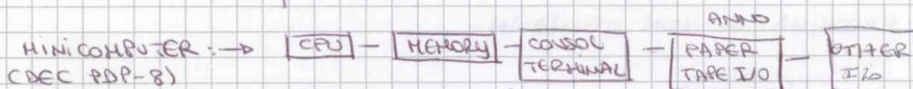
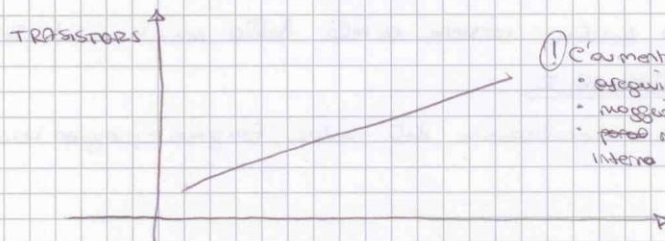
CAUSE ESPANSIONE:

1. aumento capacità CPU
2. discesa dei costi della memoria
3. discesa dei costi dei dischi

GENERAZIONE V (modello ubiquitous computing)

Introduzione da parte della APPLE del computer palmare (PDA)

- LEGGE DI MOORE → il numero di transistor su di un chip raddoppia ogni 18 mesi
Aumento della capacità dei chip di memoria - capacità della CPU



- ↳ interconnessione a bus molto flessibile
- ↳ architettura incentrata su I/O
- ↳ possibilità di connettere qualsiasi periferica

MAINFRAME : grossi calcolatori per applicazioni scientifiche, militari e pubblica amministrazione

- ↳ IBM 7090 → anni '60 per grandi aziende
- ↳ IBM 1401 : → piccole aziende

Calcolatori Elettronici

2/03/2012

Lezione II

- Cache: memoria vicina alla CPU
→ cache di primo livello → vicinissime alla CPU e non meno che
ci si allontana dalla CPU si parla di cache di secondo livello etc

LEGGI DI NATHAN

Il software è un gas: riempie sempre completamente qualsiasi contenitore
in cui esso si mette.

RFID (radio frequency identification)

- appartengono alla categoria usa e getta
- Memorizzano un numero di 128 bit
- dotati di un trasponder radio
- su chip
- è tipicamente senza batteria

se valore dei costi e dell'
aumento della memoria
disponibile, le dimensioni del
software sono sempre cresciute
in proporzione.

Nel caso di RFID attivi la distanza
di comunicazione con la stazione
è ampia

MICROCONTROLLORI

Piccoli computer incisi in vari dispositivi tipicamente connessi in
rete → si usano con modalità embedded

si trovano su:

- Elettrodomestici
- Telefoni
- Automobili
- periferiche
- dispositivi medici

Dotati di:

- una CPU
- una piccola memoria
- qualche dispositivo I/O

GAMES - COMPUTERS

1. COMPUTER WORKMAN: Effetti grafici speciali
Software di base esaurito
2. PLAYSTATION 3: Non estendibile
CPU IBM multicore a 128 bit a 3,2 GHz
256 MB di RAM
3. XBOX 360 chip grafico RIX da 256 MB a 550 MHz
CPU IBM Xenon: tri-core a 32 bit e 3,2 GHz
512 MB ram
chip grafico Xenon a 500 MHz

SMARTPHONE

Telefoni cellulari dotati di CPU (potenti): con sistema operativo: android,
ios, windows.

- Fotocamera
- Funzionalità estendibili
- CPU da 800 MHz

TABLET PC (computer normali con schermi ridotti)

- dotati di interfacce grafiche basate sul touch-screen
- CPU potenti > 1 GHz
- processore grafico
- memoria ridotta
- tastiere virtuali

TIPLOGIE DI COMPUTER TRADIZIONALI

↳ PERSONAL COMPUTER: desktop - laptop

↳ SERVER - WORKSTATION:

- su rete locale o web-server
- memoria di diversi GB
- diversi TB di disco rigido
- gestione di rete efficiente

↳ COW: (clusters of workstations)

- sistema multiprocessore ad accoppiamento L3/CO
- hardware di tipo standard: costi contenuti
- strutture di connessione veloci
- elevata affidabilità e capacità di elaborazione (SERVER FARM)

↳ MAINFRAME:

- diretti discendenti della serie 360
- gestione efficiente dell'I/O
- perfomene a dischi di molti Tbyte
- centinaia di terminali connessi

VERSIONE MODERNA:

server farm
Data center
Offrono soluzioni di cloud computing

FAMIGLIA INTEL

1. PEN PENTIUM IV: 2,4 - 3,6 GHz di frequenza di clock
 - capacità di parallelizzazione lavoro
 - oltre 40 milioni di transistori in un chip
 - Bus di sistema 800 MHz
 - potenziamento della memoria cache (512 KB - 1 MB - 2 MB)
 - dissipa 115 Watt di potenza
2. INTEL - CORE MICROARCHITECTURE: nome commerciale di una serie di microprocessori Intel di nuova generazione sempre x86
 - architettura multicore
 - " " a 64bit
 - 1,4 - 3,4 GHz di f di clock
 - fino a 8 MB cache
 - 500 milioni di transistori
 - potenza < 50 Watt
3. 2nd GENERATION INTEL CORE: microprocessori Intel di nuova generazione x86-64
 - ↳ Intel CORE i3, CORE i5, CORE i7
 - ↳ fino a 12 MB cache L3 condivisa
 - ↳ 4000 milioni di transistori
 - ↳ i7 "GULFTOWN" six core 3,6 GHz - 32nm 1170M (transistor)

FAMIGLIA (ITANIUM) : "nuova" architettura della intel → famiglia processore

- RISC HIGH-END
- Architettura a 64bit
 - 1 - 3,6 GHz clock
 - 1, 2, 4 core
 - bus di sistema 600 MHz
 - grosse memoria cache

Calcolatori Elettronici

05/03/12

Lezione III

MICROPROCESSORI FAMIGLIA SPARC: (Scalable Processor Architecture)

- ↳ proposta dalla SUN MICROSYSTEM
- ↳ dedicata a workstation Unix

Caratteristiche:

- Insieme ridotto di istruzioni
- Eseecuzione non interpretata
- Architettura a 32 bit

} asc

ULTRASPARC III

- Registri e indirizzi a 64 bit
- VIS (Visual Instruction Set)
- dedicata a sistemi high-end

ULTRASPARC IV è dual-core (1,5 GHz)

multithread → tanti processi
monothread → unico processo

[analogie MULTITASKING (SOFTWARE) - MULTITHREAD (HARDWARE)]

ULTRASPARKT2 - T2 - MULTICORE - MULTITHREAD

" " T4: MULTICORE fino a 16 e multithread 2.8 GHz
(800 milioni di transistor)

MICROPROCESSORE 8051 (EMBEDDED)

CPU - molto economica → costo bassissimo 8 bit - 60.000 transistor

100 MHz

- 4KB - 128 bytes RAM

- Programmata dal compilatore

- 32 linee I/O + 2 porte seriali

- timer (2)

→ Interfacciarsi con la rete

1 Byte = 8 bit

NUMERI e NUMERALI

NUMERO: entità astratta

NUMERALE: stringa di caratteri che rappresenta un numero in un dato sistema

→ NUMERO A PRECISIONE FINITA:

- numero finito di cifre
- si perdono alcune proprietà
 - chiusura operatori (+, -, x)
 - proprietà associativa, distributiva
 - errori di arrotondamento
 - non continuità

MECCANISMO DI BASE: sistema di posizione

- ciascuna cifra rappresenta il coefficiente di una potenza della base
- l'esponente è dato dalla posizione della cifra

b = base

$0 \leq a_i \leq b-1$

$a_n, a_{n-1}, \dots, a_0, a_{-1}, a_{-2}, \dots, a_{-k}$

□

→
$$N = \sum_{i=k}^s a_i b^i$$
 se la base è b ha b simboli

CONVERSIONE DECIMALE - BINARIO

- Si fanno divisioni ripetute per 2
- Il resto delle divisioni fornisce le cifre del numerale binario (o partire da quella meno significativa)

NUMERI NATURALI → Rappresenta interi positivi in notazione binaria con n bit si copre $[0, 2^n - 1]$
↳ si sfruttano tutte le 2^n disposizioni

ADDIZIONI: se addizioni si effettuano cifra a cifra portando il riporto alla cifra successiva
Se il numero di cifre non permette di rappresentarlo → si ha un trabocco nella prop. del riporto

MOLTIPLICAZIONI → "normali" + somme binarie

NUMERI IN VIRGOLA FISSA → si stabilisce il numero di bit
• viene fissata la pos. della virgola

INTERI POSITIVI E NEGATIVI

- Per rappresentare gli interi relativi, a parità di cifre si dimezza l'intervallo dei valori assoluti
- Si utilizzano varie rappresentazioni

→ MODULO E SEGNO

- un bit per segno 0: + 1: -
- $n-1$ bit per il modulo
- intervallo $[-2^{n-1} + 1; +2^{n-1} - 1]$

intervallo simmetrico
doppio rappresent.
dello zero

→ COMPLEMENTO A 1

- si aggiunge uno 0 a sinistra alla rapp. dei numeri positivi
- Per cambiare il segno si complementano i numerali bit a bit
- I numerali positivi iniziano per 0, i negativi per 1

Calcolatori Elettronici

06/03/12

Lezione IV

ESEMPIO : BASE BINARIA (VIRGOLA FISSA)

$$1010,01 \rightarrow 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} = 10,25$$

OPERAZIONI : ADDIZIONE (BINARI)

$0 + 0 = 0$
$0 + 1 = 1$
$1 + 0 = 1$
$1 + 1 = 0$ con riporto 1

NUMERI NATURALI \rightarrow interi positivi, in notazione binaria con n bit si
copre e' intervallo $[0, 2^n - 1]$

- \rightarrow sfrutta tutte le disposit.
- \rightarrow utilizzato anche gli 0 non significativi.

OPERAZIONI : MOLTIPLICAZIONE (BINARI)

	0	1
0	0	0
1	0	1

ESEMPIO :

$$\begin{array}{r} 1011 \times \\ \underline{101} \\ 1011 \\ 0000 \\ \underline{1011} \\ 11011 \end{array}$$

NUMERI IN VIRGOLA FISSA : estensione della rappresentazione dei numeri naturali

OPERAZIONE : MOLTIPLICAZIONE

- \rightarrow si opera come in decimale \rightarrow si tiene conto della virgola
- \rightarrow moltiplicare per 2^n equivale spostare il punto di n posti a destra
- \rightarrow moltiplicare per 2^{-n} " " " " a sinistra

ESEMPIO : ADDIZIONE

$$3,5 + 2,75 = 6,25$$

$$\begin{array}{r} 0011,10 + \\ \underline{0010,11} \\ 0110,01 \end{array}$$

COMPLEMENTI

COMPLEMENTO 1

- ↳ si aggiunge uno 0 a sinistra → numeri positivi / 1 → negativi
 - ↳ per cambiare il segno si complementa bit a bit
 - ↳ con n bit $[-2^{n-1} + 1, 2^{n-1} - 1]$
 - ↳ è una notazione puntuale
- ! complementare: significa cambiare segno
- doppia rappresentazione dello 0
 - 0000
 - 1111

INTERI POSITIVI E NEGATIVI

Per rappresentare gli interi relativi a posto di cifre si detta l'intervallo dei valori assoluti

- VARIE RAPPRESENTAZIONE -

MODULO E SEGNO:

- un bit per il segno 0: + 1: -
- n-1 bit per il modulo
- intervallo $[-2^{n-1} + 1, 2^{n-1} - 1]$

COMPLEMENTO A 2

- ↳ I positivi hanno la stessa rappresentazione che in complemento a 1
 - ↳ I negativi si ottengono sommando 1 alla loro rappresentazione in complemento a 1
 - ↳ Intervallo con n bit $[-2^{n-1}, 2^{n-1} - 1]$
- ! Per complementare cambia segno al numerale
 → Partendo da destra si cascano i numeri tutti i bit fino al primo uno compreso e poi si complementa bit a bit

- intervallo più esteso
- una sola rappresentazione dello 0
- complementare (a 2) = cambiare segno

! si complementa ad 1 o a 2 solo per rappresentare il numero negativo

ESEMPIO $(-347)_{10} \rightarrow [-2^{n-1}, +2^{n-1} - 1]$

2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0			
+347	→	512	256	128	64	32	16	8	4	2	1	
307	2		0	1	1	0	1	1	0	1	0	1
14	13	86	2	0	0	1	0	0	1	0	1	
7	1	0	43	2	2	0	1	0	1	0	1	
1	1	0	1	2	1	0	1	0	1	0	1	

Calcolatori Elettronici

06/03/12

RAPPRESENTAZIONE IN ECCESSO 2^{n-1}

Lezione III

- I numeri vengono rappresentati come somma fra il numero dato e una potenza di 2
- con n bit si rappresenta l'eccesso 2^{n-1}
- Intervallo come CP2 $[-2^{n-1}, +2^{n-1} - 1]$

I numeri si ottengono complementando il bit più significativo

ESEMPIO

n = 4 bit : eccesso 8, $[-8; +7]$

-3	$-3 + 8 = 5$	0101
+4	$+4 + 8 = 12$	1100

OPERAZIONI IN COMPLEMENTO : ADDIZIONI

- Nelle CP2 somme e sottrazioni tra numeri sono gestite nello stesso modo, ma si deve ignorare il trabocco
- Se i due operandi hanno segno diverso il risultato è sempre corretto
- Se i due operandi hanno lo stesso segno e il risultato ha segno diverso c'è un errore

SOTTRAZIONI : per fare la differenza si complementa il sottraendo e si somma

MOLTIPLICAZIONI : si fanno tra i valori assoluti e alla fine se necessario si complementa

definizione → NUMERALE : ① è solo una stringa di cifre ② un numerale rappresenta un numero solo se si specifica un sistema di numerazione ③ lo stesso numerale rappresenta diversi numeri in diverse notazioni

NOTAZIONE IN VIRGOLA MOBILE

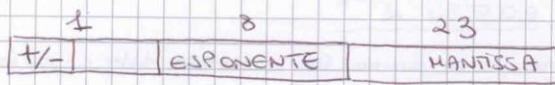
- Estende l'intervallo di numeri rappresentati a potenze di cifre, rispetto alla notazione in virgola fissa
- Numeri reali rappresentati tramite una coppia di numeri $\langle m, e \rangle$

$$n = m \cdot b^e$$

m : mantissa (normalizzata tra due potenze successive della base)
e : esponente intero con segno $b^{i-1} \leq |m| < b^i$

Se m che e hanno un numero finito di cifre : Intervalli limitati / Error di arrotondamento

STANDARD IEEE 754 a 32 bit



- Esponente: eccesso 127 $\rightarrow -126 \leq e \leq 127$
- Mantissa: rappresenta solo la parte frazionaria $1 \leq m < 2$
- Numeri NORMALIZZATI $[2^{-126}, 2^{128}]$
- Numeri DE-NORMALIZZATI (esp. 00000000)
 - m non normalizzata $0 < m < 1$
 - e vale convenzionalmente 2^{-126}

ALTRE CONFIGURAZIONI

- \rightarrow e ed m tutti a 0 rappresentano 0
- \rightarrow m tutti 0 ed e tutti 1 : overflow
- \rightarrow m \neq 0 ed e tutti 1 : Not a number

NATAZIONE IN BASE 16 : (ESADECIMALE)

- Per i numerali esadecimali occorrono 16 cifre $\{0, 1, \dots, 9, A, B, C, D, E, F\}$
- Conversione esadecimale-binario: si fa corrispondere a ciascuna cifra esadecimale il gruppo di 4 bit che ne rappresenta il valore
- Conversione binario-esadecimale: partendo da destra si fa corrispondere a ciascun gruppo di 4 o meno cifre binarie la cifra esadecimale che ne rappresenta il valore

ESempio: F 5 7 A 3 1
 1111 0101 0111 1010 0011 0001

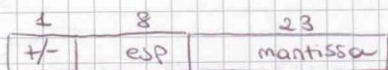
Calcolatori Elettronici

08/03/12

Lezione IV

STANDARD IEEE 754

formato non proprietario → ovvero non dipende dall'architettura



esponente : $[-127; 128]$ $-126 \leq e \leq 127$

mantissa: rappresenta la parte frazionaria

normalizzati : $1 \leq m < 2$
 $[2^{-126}; \sim 2^{128}]$

de-normalizzati : - m non normalizzata : $0 < m < 1$
 - e vale convenzionalmente 2^{-126}

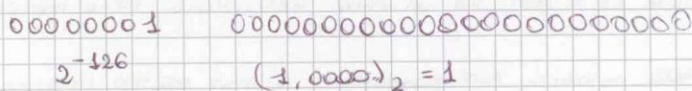
~~e ed m tutti 0 ed e tutti 1 : over~~

- e ed m tutti a 0 rappresentano 0
- m tutti 0 ed e tutti 1 : overflow
- m ≠ 0 ed e tutti 1 : not a number

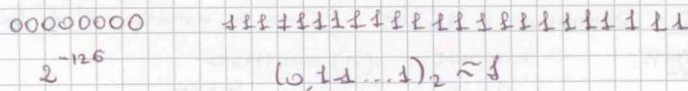
→ PIÙ GRANDE NORMALIZZATO $\sim 2^{128}$



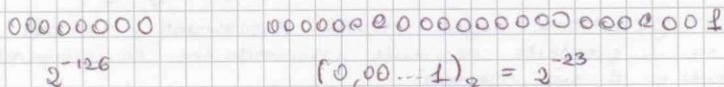
→ PIÙ PICCOLO NORMALIZZATO $\sim 2^{-126}$



→ PIÙ GRANDE DENORMALIZZATO $\sim 2^{-126}$



→ PIÙ PICCOLO DENORMALIZZATO



OPERAZIONI

- ADDIZIONE IN VIRGOLA MOBILE -

① occorre scalare le mantisse per eguagliare gli esponenti. L'addendo più piccolo perde cifre significative

ESEMPIO: $n_1 + n_2$

$$\begin{array}{r}
 n_1: 0 \ 10011001 \ 00010111011100101100111 + \\
 n_2: 0 \ 10101010 \ 11001100111000111000100 =
 \end{array}$$

} scala e' esponente e mantisse

OPERAZIONI: MOLTIPLICAZIONE

→ si moltiplicano le mantisse e si sommano algebricamente gli esponenti se necessario si scala la mantissa per normalizzarla e si aggiusta l'esponente

ESEMPIO: $n_3 = n_1 \times n_2$

$$\begin{array}{r}
 n_1: 0 \ 10011001 \ 10010111011100101100111 \ e_1 = (26)_{10} \\
 n_2: 1 \ 10101010 \ 100000000000000000000000 \ e_2 = (43)_{10}
 \end{array}$$

$e_1 + e_2 = (69)_{10} = 11000100$

$m_1 \times m_2 = 1001100110010111011101110101$

ERRORI

→ Rappresentando un numero reale n in una notazione floating point si commette un errore di approssimazione

→ In realtà viene rappresentato un numero razionale n' con un numero limitato di cifre

ERRORE ASSOLUTO: $e_A = n - n'$

ERRORE RELATIVO: $e_R = e_A / n = (n - n') / n$

① se la mantissa è normalizzata l'errore relativo massimo è costante su tutto l'intervallo rappresentato ed è pari ad un'unità sull'ultima cifra rappresentata



Nelle notazioni non normalizzate l'errore relativo massimo non è costante

ESERCIZIO: Rappresen. a 16 bit

→ si omette e' i davanti la mantissa

primo bit rappresenta segno +/-
1

esponent: 8 bit

(1) mantissa

7 bit

eccezione 128

$1 \leq m \leq 2$

1) Calcolare gli estremi degli intervalli rappresentati i numeri corrispondenti e l'ordine di grandezza decimale assumendo che le configurazioni con tutti 0 o con tutti 1 sia riservate

2) il non complemento a 2 = FF5AB9

3) COMPL. A 1 = 13B472

4) Errori

Calcolatori elettronici

08/03/12
 Esame II

\Rightarrow MAX 11111110 $11111111 \approx 2^{126} \times 2 = 2^{127}$
 MIN 00000001 $00000000 \approx 2^{-127} \times 1 = 2^{-127}$

osservazione $\frac{1111}{2^n - 1} = 15$

ORDINE DI GRANDEZZA

$x: \frac{127}{126} = 3:10$ $x = \frac{127 \times 3}{10} = 38,1$

ordine di grandezza del MAX $\approx 10^{38}$
 MIN $\approx 10^{-38}$

$2^{127} = 10^x$

CP2) CONVERSIONE DI FORMATO

F F 5 A B 9
 1111 1111 0101 1010 1011 1001

→ chiedersi se è positivo o negativo

se è positivo → borsale faccio la somma dei bit

$0000 \ 0000 \ 1010 \ 0101 \ 0100 \ 0111$
 $- (2^{15} + 2^{13} + 2^{10} + 2^8 + 2^6 + 2^2 + 2^1 + 2^0)$

$-2^{15} (1 + 2^{-2} + 2^{-5} + 2^{-7} + 2^{-9} + 2^{-13} + 2^{-14} + 2^{-15})$

$\underline{1} \quad \underline{10001111} \quad \underline{0100101}$

ERRORE ASSOLUTO $E_A = 2^6 + 2^2 + 2 + 2^0$

A parte non rappresentata

CP1) $2^{15} (1.010010101000111)$ → il numerale è costante

operazione per convertire in virgola mobile

1 3 B 4 7 2
 0001 0011 1011 0100 0111 0010

$2^{20} (1.0011101101000111)$

$$E_A = 2^{20} (2^{-8} + 2^{-10} + 2^{-14} + 2^{-15} + 2^{-19}) = 2^1 + 2^4 + 2^5 + 2^6 + 2^{10} + 2^{12}$$

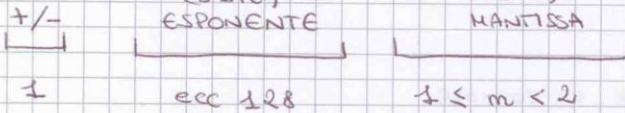
ESERCIZIO

09/03/12

Rapp. binaria a 16 bit
 (8 bit)

(7 bit)

Esazione VI



CP2) C 3 A 5

1 100 0011 1,0100101

moltiplica per 2^{-7} → $2^7 \cdot (1,0100101)$

→ (10100101)

co posto a 16 bit

→ (0000000010100101)

complemento a 2

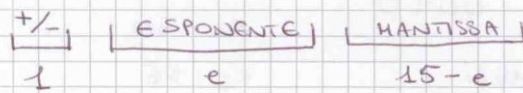
→ (1111111101011010)

Calcolatori Elettronici

09/03/12

ESERCIZIO

Lezione VI



minimo e

$$n = FF5AB9$$

1111 1111 0101 1010 1011 1001

0000 0000 1010 0101 0100 0111

comple.
a 2

$$-2^{15} (1,0100\dots)$$

5 bit n bit $[-2^{n-1}; +2^{n-1}]$

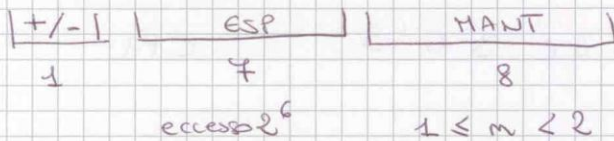
semberebbero bistore \rightarrow corretto a 6 bit poiché

15 \rightarrow 11111

Rappresentazione corretta 10 11111

eccesso 2^{e-1}

ESERCIZIO



$$FC53$$

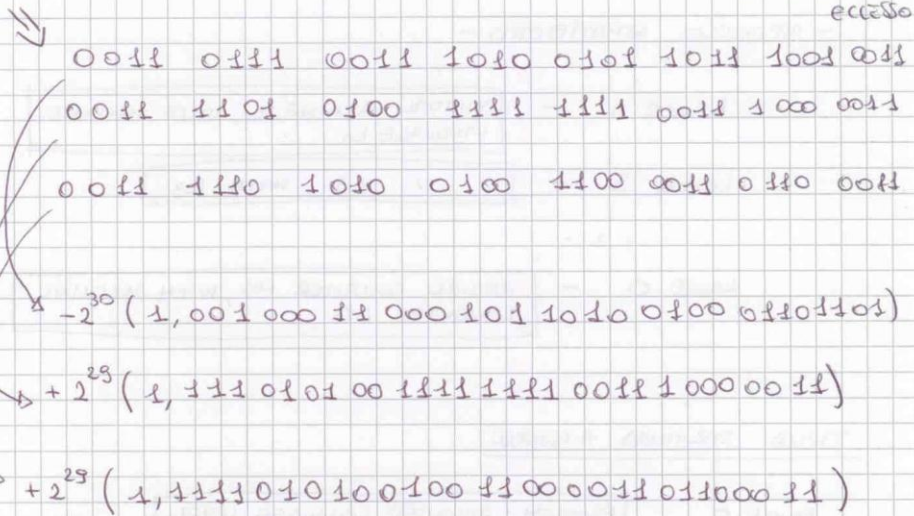
1 111 1100 0101 0011

Calcolatori Elettronici

09/03/12

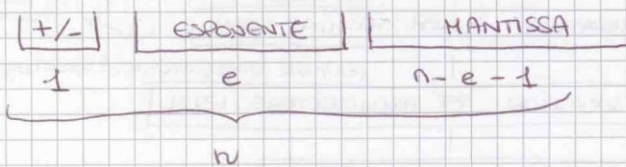
Lezione VII

esercizio

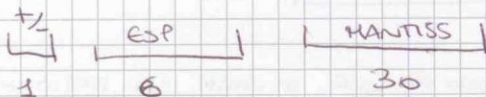


ESERCIZIO : numero minimo di bit senza commettere errore

↑ riferito ai numeri in alto



6 bit [-32, 31]



ORGANIZZAZIONE GENERALE DEL CALCOLATORE

12/03/12

Lezione VII

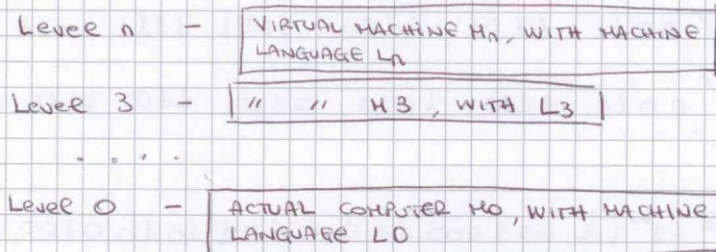
CALCOLATORE ELETTRONICO : macchina fatta di dispositivi elettronici che può risolvere problemi eseguendo istruzioni fornitegli

PROGRAMMA : sequenza di istruzioni in un linguaggio

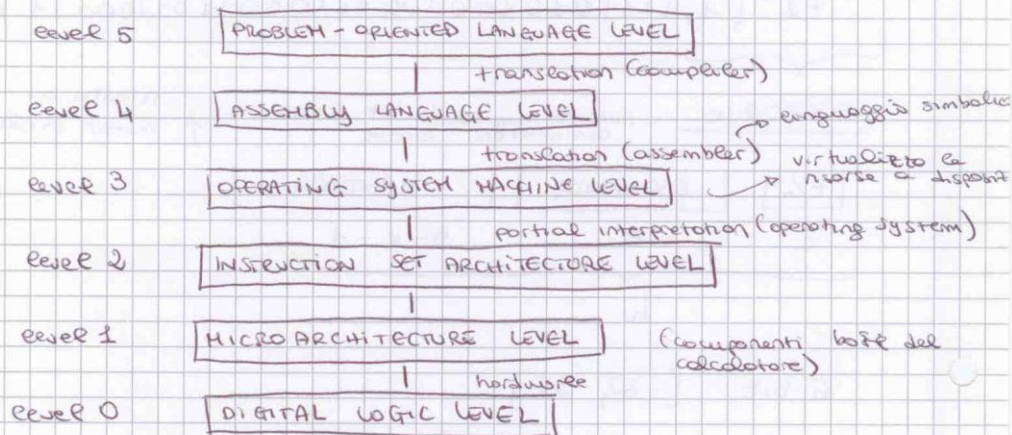
LINGUAGGIO MACCHINA : eseguibile direttamente da un calcolatore

→ trasformare un linguaggio di alto livello in un linguaggio macchina

- APPROCCIO STRATIFICATO -



TIPICA STRUTTURA A LIVELLI



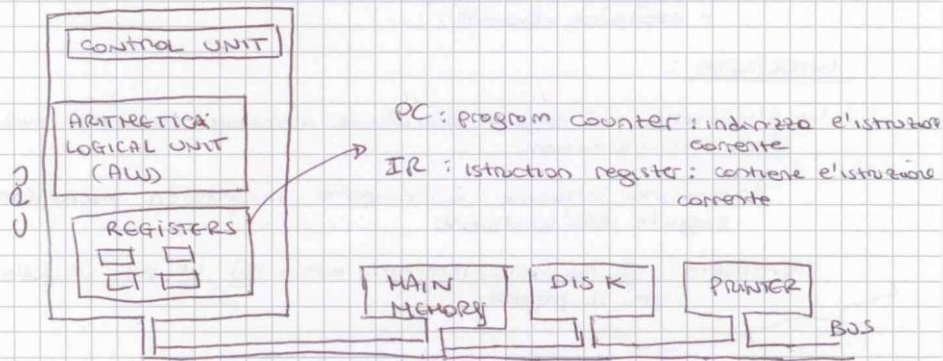
Il livello 2 è il più basso al quale un utente può programmare

Calcolatori Elettronici

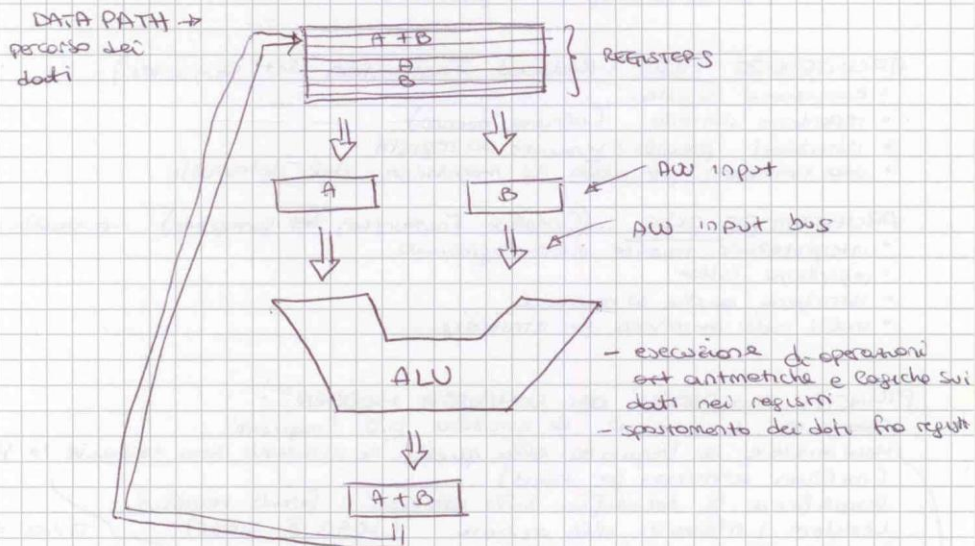
12/03/12

Lezione VII

STRUTTURA DEL COMPUTER



STRUTTURA CPU



l'esecuzione di ciascuna istruzione nella CPU richiede:

- ① carica l'istruzione da memoria in IR (FETCH)
 - ② incrementa (incrementa) il program counter
 - ③ decodifica l'istruzione
 - ④ se l'istruzione usa un dato in memoria calcolarne l'indirizzo
 - ⑤ carica l'operando in un registro
 - ⑥ esegui l'istruzione (EXECUTE)
 - ⑦ torna al passo 1 per l'istruzione successiva
- gli accessi alla memoria sono solo ①

FILOSOFIA DI ESECUZIONE (o una o l'altra)

ESECUZIONE DIRETTA

- ↳ le istruzioni possono venire eseguite direttamente dai circuiti hardware
- approccio complesso:
 - repertorio di istruzioni limitato
 - progettazione hardware complessa
 - esecuzione efficiente

INTERPRETE

- ↳ l'hardware può eseguire solo alcune operazioni elementari molto semplici dette micro istruzioni
- ↳ ciascuna istruzione è scomposta in ulteriori micro-istruzioni per eseguire dell'hardware

VANTAGGIO: ① Repertorio istruzioni esteso ② HW più compatto ③ flessibilità di progetto

MICROPROGRAMMAZIONE: l'unità di controllo della CPU esegue un microprogramma per effettuare l'interpretazione delle istruzioni macchina

- ↳ il microprogramma è contenuto in una ~~memoria~~ memoria ROM su chip del processore

ARCHITETTURE RISC (Reduced Instruction Set Computer)

- esecuzione diretta
- repertorio ristretto (alcune decine)
- istruzioni prevalentemente su registri
- una istruzione per ciclo di macchina del datapath

ARCHITETTURE CISC (Complex Instruction Set Computer) → quelle intel

- interpretazione tramite microprogramma
- repertorio esteso
- istruzioni anche su memoria
- molti cicli macchina per istruzione

PRINCIPI PROGETTUALI DEI COMPUTER MODERNI

Eseguire direttamente le istruzioni più frequenti

Massimizzare la frequenza alla quale le istruzioni sono eseguite in MIPS (Million instruction per second)

Semplificare la decodifica delle istruzioni: formati regolari

Limitare i riferimenti alla memoria (LOAD & STORE)

Compilare il numero di registri

Tipici della RISC ma anche CISC

Agli inizi del 2000 si è capito che questo strada non era percorribile perché c'era troppa d'efficienza dissipazione di energia

- ↳ si introduce il concetto di PARALLELISMO

Calcolatori Elettronici

13/03/12

VIII lezione

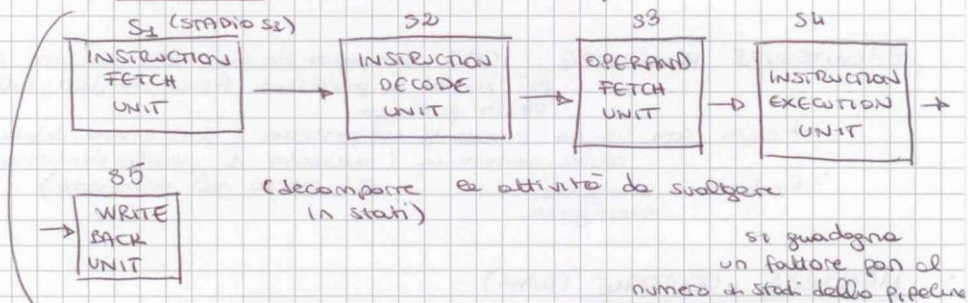
PARALLELISMO: è l'unica strada per aumentare le prestazioni
 Limite di un'esecuzione sequenziale

Due tipi di PARALLELISMO

- A) A LIVELLO DI ISTRUZIONI: diverse istruzioni eseguite insieme diverse fasi dello stesso istr. eseguite insieme
- B) A LIVELLO DI PROCESSORI: molti processori lavorano insieme allo stesso problema
 - Fattori di parallelismo molto elevati
 - diversi tipi di interconnessione e di cooperazione

COME REALIZZO IL PARALLELISMO?

PIPELINE (A LIVELLO DI ISTRUZIONI)

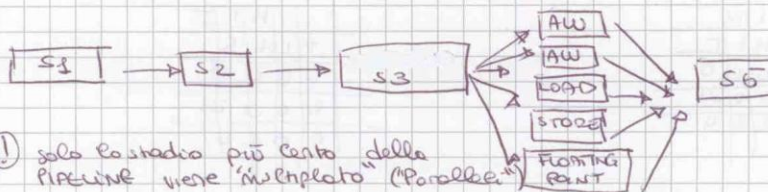


- ciascuna istruzione è divisa in fasi
- l'esecuzione avviene in una pipeline a più stadi
- più istruzioni in esecuzione contemporanea
- una istruzione completata per ogni ciclo

Inoltre consente un compromesso tra:
LATENZA: tempo per eseguire un'istruzione

AMPIEZZA DI BANDA: numero di istruzioni completate per unità di tempo MIPS (milioni di istr. al secondo)

ARCHITETTURE SUPER SCALARI: si ottengono più istruzioni (4-6) si aumento avendo più di una pipeline nel microprocessore
 → parallelismo fisico



!! solo lo stadio più lento della PIPELINE viene moltiplicato (Parallelismo)

(Prima forma di architettura super-scalare)

ARCHITETTURA MULTIPROCESSORE : duplica i processori

(PROCESSORS)
 → ARRAY COMPUTER : processori identici ed autonomi → eseguono le stesse istruzioni su dati diversi

→ VECTOR PROCESSORS : unità aritmetiche che operano su registri vettoriali (parallelizza i registri)

MULTIPROCESSORI : le CPU operano indipendentemente

SHARED MEMORY : il bus può diventare collo di bottiglia

PRIVATE MEMORY : contiene le codice e parte dei dati → scambio dati tramite la shared memory

MULTICORE : più CPU (core) sullo stesso chip

ARCHITETTURE MULTICORE : CPU composta da più core, ovvero da più nuclei di processori fisici montati sullo stesso package

→ ogni core : è un processore indipendente e può essere dotato di cache cache autonoma (memoria di piccola dimensione realizza all'interno del processore)

→ architettura omogenea
 " " eterogenea

MEMORIA CENTRALE (RAM) :

Contiene sia dati che i programmi

Memorizzazione binaria (bit)

Cella : unità indirizzabile

→ byte = 8 bit

word : insieme di k byte (k dip. dall'architettura)

Indirizzo : (della cella) tramite il quale la CPU accede al dato nella cella

Indirizzi binari a m bit : n° di celle a disposizione 2^m (spazio di indirizzamento)

① per le codice ASCII in carattere → un singolo byte

- ORDINAMENTO DEI BYTE -

BIG ENDIAN (IBM, SUN)

0	J	I	M
4	S	M	I
8	H	O	O
12	O	O	Z
16	O	O	4

LITTLE ENDIAN (INTEL)

		M	I	J
	T	I	M	S
	O	O	O	H
	O	O	O	Z
	O	O	O	4

Calcolatori elettronici

13/03/12

VIII lezione



CODICI A CORREZIONE DI ERRORE;

tecniche per garantire maggiore affidabilità nello registrazione
- trasmissione di informazioni binarie

↳ recupero di errori hardware tramite codifiche ridondante

CODIFICHE con $n = m + r$

n bit complessivi di codice

m bit di dati

r check bit (ridondanti)

DISTANZA DI HAMMING: (di un codice) h = distanza di Hamming

minima tra due codifiche valide del codice

↳ tra due codifiche: n° di bit diversi

Ex: 0101 e 1001 sono a distanza 2

(!) per RIEVARRE: errori di d bit che no

almeno $h = d + 1 \Rightarrow d \leq h - 1$

per CORREGGERE errori su d bit occorre che no almeno

$h = 2d + 1 \quad d \leq (h - 1) / 2$

ESEMPIO

↳ 0000 1111 → 1111 0111 → 4 errori

RIEVAZIONE DI ERRORE SINGOLO

↳ Nel caso più semplice si vogliono solo rilevare errori singoli
• bisogna aggiungere un solo check bit $r = 1 \quad n = m + 1$

BIT DI PARITÀ: scelto in modo che il numero complessivo di 1

nella codifica sia sempre pari

- errore rilevato da circuiti molto semplici
- memorie segnalano parity error quando un errore si manifesta

ES: 11011010 bit PARITÀ 1 → OK

01100101 bit parità 0 → 01101101 Error

- CORREZIONE DI ERRORI SINGOLI -

- m dato bit, r check bit, n bit totali
- 2^m codifiche valide
- n codifiche errate a distanza 1 da ciascuna delle valide
- ogni codifica valida ne richiede in tutto n+1

ogni codifica valida ne richiede n+1

$$(n+1)2^m \leq 2^n \Rightarrow (m+r+1) \leq 2^r$$

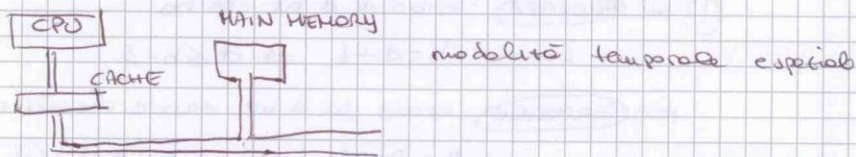
OVERHEAD: % di bit per controllo

MEMORIA CACHE:

- ↳ è sempre più lenta della CPU e tende a rallentarla
- ↳ memorie veloci sono disponibili ma per piccole dimensioni

La CACHE funziona alla velocità del processore e quindi ne scade la sua lentezza della memoria

- ↳ contiene le ultime porzioni di memoria acceduta
 - principio della LOCALITÀ SPAZIALE ovvero accade o a dati contigui oppure a dati molto recenti
 - funziona bene a causa della località degli accessi



se una parola viene letta k volte di seguito, k-1 volte sono trovate in cache

CACHE HIT RATIO $H = (k-1) / k$

CACHE MISS

↳ fallimento

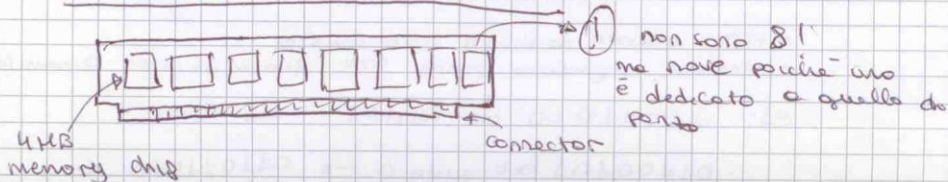
↳ successo nella ricerca di un processo

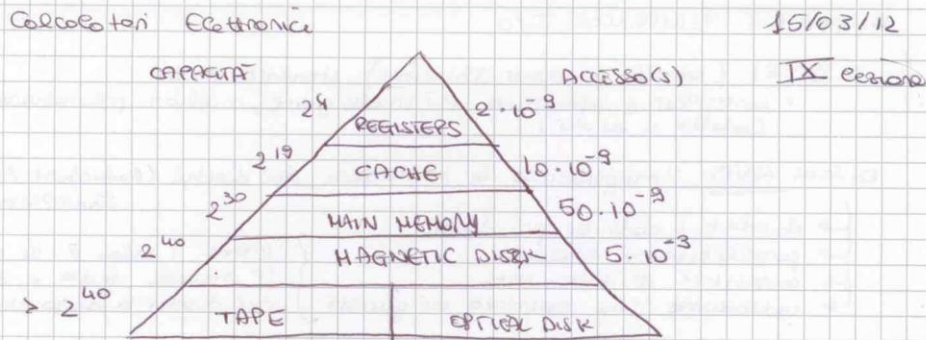
A (tempo medio di accesso alla memoria) = $C + (1-H)m$

m : tempo di accesso alla memoria
 C : tempo di accesso della cache

la memoria è organizzata in blocchi

TIPICI TIPI SCHEDE DI MEMORIA





- Scendendo nella gerarchia
- cresce il tempo di accesso
 - aumento della capacità
 - diminuisce il costo per bit

16/03/12 X lezione

DISCHI MAGNETICI (MEMORIA SECONDARIA)

- dischi di alluminio con registrazione su tracce concentriche hanno circa 5000-10000 tracce/cm
- Tracce divise in settori contenenti: dati un preambolo e un ECC (errore correcting code)
- Velocità di rotazione costante 5.400 - 10.800 RPM
- Velocità di trasferimento di 20 - 40 Mbit/sec
- Burst rate: velocità da quando la testina è su primo bit
- Sustained rate: velocità di transf. in un certo intervallo

- CIUNDRIO: insieme di tracce sulla stessa verticale
- TEMPO DI SEEK: spostamento della testina sul cilindro desiderato, dipende in parte dalla distanza ($\approx 5-10$ ms)
- TEMPO DI LATENCY: spostamento sul settore desiderato

TEMPO DI ACCESSO $\Rightarrow t_{acc} = t_{seek} + t_{lat}$

Un primo problema si genera quando si desidera a scivolare su tracce più vicine (con raggio minore)

Ⓜ la testina non deve toccare mai il disco \rightarrow altrimenti è hard disk si

- STANDARD -

- IDE (IBM) prevede a 16 testine, 63 settori transf ≈ 4 MB/sec
- EIDE estende lo standard mediante LBA (logical block addressing)
- ATA-3: 33 MB - 66 MB/sec a 1000 MB/sec \rightarrow 48 bit
- SATA prevede connettori con meno bit da 16 a 7 ma introduce velocità maggiori > 300 MB/sec

dispositivi periferici I/O

SCSI (small computer interface), standard ANSI

- controller e interfaccia più intelligente, migliori prestazioni (adatto a server)

DISCHI RAID : migliorano le prestazioni dei dischi (Redundant Array of Inexpensive)

- ↳ dividere i dati su più dischi
 - ↳ parallelizzare l'accesso
 - ↳ aumentare la data-rate
 - ↳ introdurre una resistenza ai guasti
- prende i file e li spezza in piccole fette e distribuisce nei dischi a disposizione

RAID LEVEL 0 e LEVEL 1 : su n dischi si può guadagnare un fattore n sia in lettura che in scrittura

⚠️ Lo MTBF (Mean Time Between Failures) peggiora

Inoltre ciascun disco è duplicato → SHADOWING

RAID 0 : → MTBF peggiore

→ non c'è ridondanza : non è un vero RAID

  (senza copie)

RAID 1 : → ~~ogni~~ ciascun disco è duplicato : shadowing

→ ottime prestazioni soprattutto in lettura

→ eccellente resistenza ai dati

RAID 2 : striping (e' affettamento dei dati) → è fatto a livello di bit. Si registrano ed i bit per ogni bit disco

- Resistenza a guasti semplici
- Guadagna un fattore 1/2 in read e write
- forte overhead (75%)

RAID 3 : Resiste a guasti semplici. Le bit di parità vengono scritte su un disco, consentendo la correzione

RAID 2 e 3 offrono una eccellente data-rate ma permettono di gestire solo una operazione su disco per volta perché ciascuna operazione coinvolge tutti i dischi

RAID 4 : a livello di blocco → drive non sincronizzati

→ la strip nell'ultimo disco contiene i bit di parità dell'insieme di bit omologhi di tutte le altre strip

→ se una sola strip è stata scritta occorre leggere tutte le altre per calcolare la parità → il disco di parità è il collo di bottiglia

RAID 5 : distribuisce le strip di parità

Calcolatori Elettronici

16/03/11

Lezione 8

Dischi ottici - CD-ROM

- Registrazione su pit e lands
- e' interferenza nello spazio di un pit e l'intensita' luminosa
- Rotazione variabile 200 - 520 RPM per avere flusso dati uniforme



ORGANIZZAZIONE DEI DATI

- Byte codificati in simboli di 4 bit
- Frame di 62 simboli, di cui 40 utili
- Settore di 98 frame, con 24 byte utili più 16 di preambolo e 288 di ECC
- contiene in tutto 650 MB utili
Drive 1x → 150 KB/sec
Drive 32x → 5 Mbit/sec

CD SCRIVIBILI (CD-R)

- laser a due potenze (scrivibile: CD-R)
- AUT: (Scrive) "brucia" delle piccole aree nello strato colorato
- BASSA: legge nei CD-ROM

CD-RW

- laser a tre potenze
- AUT: (Scrive) : cristallino → amorfo
- MEDIA: (cancella) amorfo → cristallino
- BASSE (legge): tipo CD-ROM

STANDARD per i CD-ROM

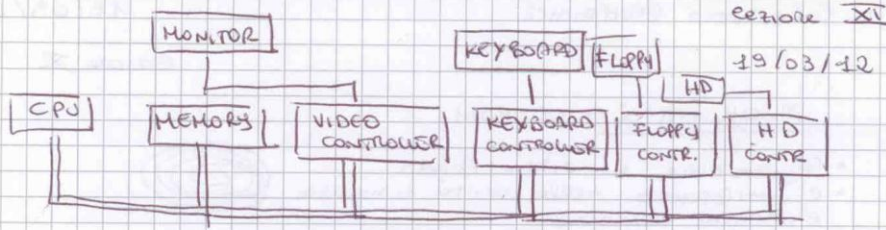
- RED BOOK: standard ISO per i CD audio simboli e frame
- YELLOW BOOK: standard ISO per CD-ROM dati
- GREEN BOOK: standard CD multimediali

DVD (Digital Versatile Disk)

- aumento della capacità
 - Layer $\lambda = 0,65 \mu$
 - Pit = $0,4 \mu$
 - Capacità: 4,7 GB
 - data-rate 1x
- si utilizzano due strati di registrazione e ricorda dell'inclinazione del laser
- singola faccia, layer doppio: 8,5 GB
- doppio faccia, layer doppio: 17 GB

BWE-RAY:

- laser (HeNe) $\lambda = 0,405 \mu$
- Singola faccia: 25 GB
- doppia faccia: 50 GB
- data rate: 45



i dispositivi I/O sono connessi al bus tramite controller

- ↳ i controller gestiscono autonomamente i trasferimenti da e per la memoria; DMA (Direct Memory Access)
- ↳ possono comunicare con la CPU tramite le interruzioni

STRUTTURA FISICA DEL PC

- scheda madre (Mother Board)
- CPU, Bus e vari controller
- Bus, costituito da una serie di piste su circuito stampato

BUS: bus dedicato → comunicazione diretta CPU memoria

ISA: Industry Standard Architecture

PCI: Peripheral Component Interconnect → ha più linee e maggiore frequenza di funzionamento

MONITOR CRT

sono gli stessi usati nei televisori → con prestazioni migliori

- DOT PITCH: dimensione pixel (mm)
- RISOLUZIONE: n° di pixel (1280 x 1024)
- REFRESH RATE: frequenza con cui sono i pixel

DISPLAY FLAT PANEL

ovvero a disposizione i cristalli liquidi → veri e propri cristalli con proprietà di disporsi in maniera diversa in base al campo elettrico/magnetico

↳ basati sulla condizione di luce polarizzata da parte dei cristalli liquidi

↳ DISPLAY: A MATRICE ATTIVA (TFT)

↳ si riesce a lavorare su singoli pixel

Calcolatori Elettronici

18/03/12

VIDEO RAM: memoria dedicata alla gestione video → situato nel controller video XI lezione

→ la dimensione dipende dalla dimensione risoluzione

① L'immagine è costituita da una matrice di punti (pixel)

Dimensione: $P \times N \times K$

P: n° di pixel

N: n° di byte x pixel

K: n° di matrici (frame)

$P \times N \times K$

TRASIBRE: dispositivo fisico che interviene rispetto al calcolatore ovvero interviene al di là di quello che succede nel calcolatore

↳ genera una interruzione dalla CPU
mapping dei tasti gestito dal software

MOUSE

Tecnologia:

- ↳ MECCANICA: movimento di rotelle dotate di potenziometri
- ↳ OPTO-MECCANICA: come quelli meccanici ma rotelle rilevate da infrarossi
- ↳ OTTICA: dotate di LED (Light Emitting Diode) fotorelettrici

① 3 byte inviati ogni volta che il mouse fa un certo spostamento

STAMPANTI

- A MATRICE
- A GETTO DI INCHIOSTRO
- LASER

STAMPANTI INKJET

STAMPANTI GRAFICHE

- Termiche: espansione e contrazione dell'inchiostro per impulsi elettrici nell'avvolgimento

↳ poca colore ma lente

PROBLEMI:

- trascinamento della carta
- intasamento dei capiletto
- generazione dell'img da parte del computer

STAMPANTI LASER

eccitazione elettrostatica del tamburo di Selenio con permiscela laser
→ Trasferimento del toner sulla carta a caldo e sua fusione

VANTAGGI: silenziosi, veloci e precisi

GESTIONE DELLE STAMPANTI LASER:

- ↳ generazione dell'img. viene fatta dalla stessa stampante
- ↳ stampante dotata di microproc. e parecchi MB di RAM e RAM
- ↳ set di caratteri contenuti nella ROM
- Il computer invia una descrizione dell'immagine in un linguaggio POSTSCRIPT → programma che descrive un documento
- ↳ linguaggio proprietario ADOBE per la descrizione di img

MODEM: informazioni binarie trasmesse su linee analogiche modulando una portante

- ↳ MODULAZIONE DI AMPIEZZA, FREQUENZA & FASE
- BITRATE: frequenza di invio dei bit
- BAUD RATE: freq. con cui viene il segnale

ADSL: tecnica appartenente alla famiglia Digital Subscriber Line
ASYMMETRIC DIGITAL SL

- ↳ parallelizza i vari segnali
- ↳ suddivisione dello banda in 256 canali indipendenti
- trasmissione DSL freq. 1.1 MHz
- POTS 3000 MHz

STANDARD COLORI -

- **CYMK** (Cyan, Yellow, Magenta, Black) : per img. a luce riflessa
- **RGB** (Red, Green, Blue)
- **CMYK**: set di colori gestiti

CAMERE DIGITALI: → utilizzano matrici di Charge-Coupled Devices (CCDs)
fotosensibili

- filtro per suddividere nei colori base
- macchina con 5 milioni di pixel

- **PRE-PROCESSING**: messa a fuoco, esposizione, bilanciamento
- **MEMORIZZAZIONE** dell'immagine in forma di matrice (RAM grande)
- **POST-PROCESSING**: bilanciamento, correzione, eliminazione oculare, compressione (JPEG)

Calcolatori Elettronici

19/03/12

XI lezione

CODICE ASCII : - Codice ASCII a 7 bit, poi esteso a 8 bit
 - escape sequences → per caratteri speciali

CODICE UNICODE

- Codice ASCII a 7 bit, poi esteso a 8
- Escape sequences: per caratteri speciali

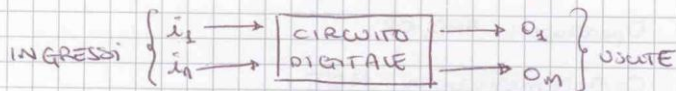
Codice UNICODE a 16 bit: nuova proposta di standard:

- 65.536 code points
- Semplifica la scrittura del SOFTWARE
- 336 code points: alfabeti latini
- 112 accenti e simboli diacritici
- Greco, Latino, Ebraico etc
- 21.000 ideogrammi cinesi
- 11.000 sillabe Coreane

20/03/12

LOGICA DIGITALE

XII lezione



- circuiti elettronici i cui ingressi e cui uscite assumano solo due livelli
- se circuito sono uscite associate a funzioni che calcolano le uscite a partire dai degli ingressi

$$o_1 = f_1(i_1, \dots, i_n)$$

$$o_m = f_m(i_1, \dots, i_n)$$

FUNZIONI LOGICHE

- $y = f(x_1, \dots, x_n)$ → variabili che possono assumere due soli valori

$\{0, 1\}$
 ↑
 FALSO VERO

$\{F, T\}$
 ↑
 FALSE TRUE

TAVOLA DELLA VERITA'

x_1	x_2	...	x_{n-1}	x_n	f
0	0		0	0	0
0	0		0	1	1
.
1	1		1	1	0

2^n combinazioni di ingresso \rightarrow 2^{2^n} funzioni distinte di n variabili

ESEMPLO $n=2 \rightarrow 4$ funzioni

x_1	f_0	f_1	f_2	f_3
0	0	0	1	1
1	0	1	0	1

f_2 NOT

$f_0 (0001)^T \rightarrow$ AND

$f_2 (0111)^T \rightarrow$ OR

ALGEBRA CIRCUITALE

Rappresentazione algebrica di funzioni booleane
Reticolo distributivo complementato

Insieme: $I = \{0, 1\}$

Operatore: AND, OR

Complementazione: NOT

NOTAZIONE \rightarrow x, y due variabili booleane

- L'AND di x e y si indica $x \cdot y$ (o xy)
- L'OR di x e y si indica $x + y$
- IL NOT di x si indica con \bar{x}

TEOREMA: ogni funzione booleana è algebrica, cioè rappresentabile con un'espressione algebrica

FORMA CANONICA
$$f = \sum_{j=1, \dots, m} \prod_{i=1, \dots, n} x_{ij}^*$$

x_{ij}^* vale x_i oppure \bar{x}_i

\rightarrow f è espressa come OR delle combinazioni per cui la funzione è vera

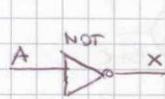
Calcolatori Elettronici

20/03/12

XII lezione

Funzione di MAGGIORANZA (M) = è vero solo se almeno due delle tre variabili sono vere

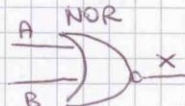
CIRCUITI LOGICI



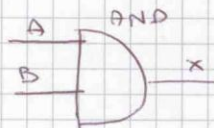
A	X
0	1
1	0



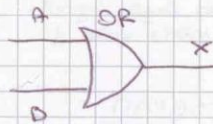
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0



A	B	X
0	0	1
0	1	0
1	0	0
1	1	0



A	B	X
0	0	0
0	1	0
1	0	0
1	1	1



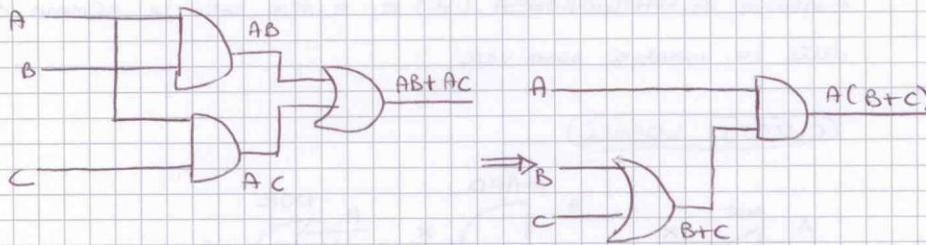
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

⚠️ PARTE LOGICHE: circuiti elementari che realizzano gli operatori dell'algebra

Funzione di DE-MORGAN → conversione dell'AND in OR

⚠️ È possibile simulare AND, OR, e NOT e quindi realizzare qualsiasi circuito, utilizzando solo NAND oppure solo NOR

ESEMPPIO



PORTE XOR



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

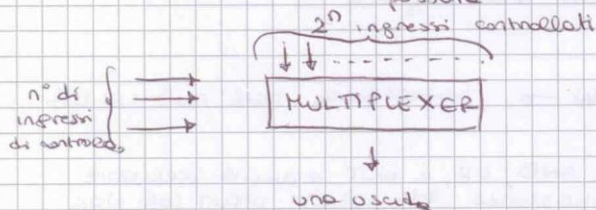
- Calcola la funzione OR - esclusiva: da uscita 1 quando uno solo degli ingressi (ma non entrambi) vale 1

CIRCUITI INTEGRATI

- Molte porte realizzate sulla stessa piastrina di silicio (chip)
- Contenitori a 14 e 68 pin - piedini
- Tempo di commutazione: tempo ingresso - uscita

↳ VARI LIVELLI DI INTEGRAZIONE → n° di porte

CIRCUITI COMBINATORI: circuiti in cui l'uscita dipende solo dagli ingressi e non dallo stato cioè dalla storia passata



① Gli ingressi di controllo selezionano quale degli ingressi controllati viene mandato in uscita

Calcolatori Elettronici

22/03/12

XIII lezione

un solo segnale in uscita ed
 n ingressi di controllo e 2^n ingressi
controllati

MULTIPLEXER: Con n multiplexer od n bit si può calcolare
qualsiasi funzione di n variabili

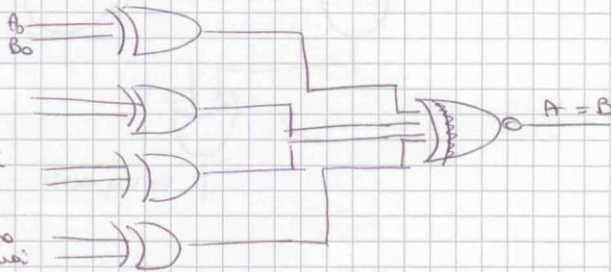
- gli ingressi controllati corrispondono ai mintermini
- utile nella conversione di un segnale parallelo seriale

DECODIFICATORE: $\rightarrow 5$ ingressi $\rightarrow 2^5$ uscite = 32

- Circuito a n ingressi e 2^n uscite
- Una ed una sola delle 2^n uscite assume valore vero in corrispondenza della configurazione di n bit in ingresso
- Utile nella conversione di un segnale seriale - parallelo

COMPARATORE:

- comparo i bit omologhi di due stringhe
- L'uscita vale 1 se e solo se $A_i = B_i \forall i$
- se $A_i = B_i$ allora $A_i \times \text{OR } B_i = 0$
- È NOR da ingresso 1 quando tutti i suoi ingressi valgono 0



- PLA: PROGRAMMABLE LOGIC ARRAY -

- Permette di realizzare una qualsiasi funzione
- Circuito configurabile tramite bruciatura (interconnessione) di connessioni
- Si possono calcolare più funzioni
 - \rightarrow generazione di un insieme di mintermini
 - \rightarrow selezione dei mintermini da inviare a ciascuna delle uscite

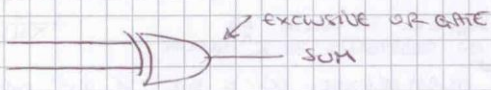
LIMITAZIONI: n° limitato di mintermini $\ll 2^n$

n° limitato di mintermini in ingresso a ciascuna degli OR di uscita

SHIFTER (SCALATURA)

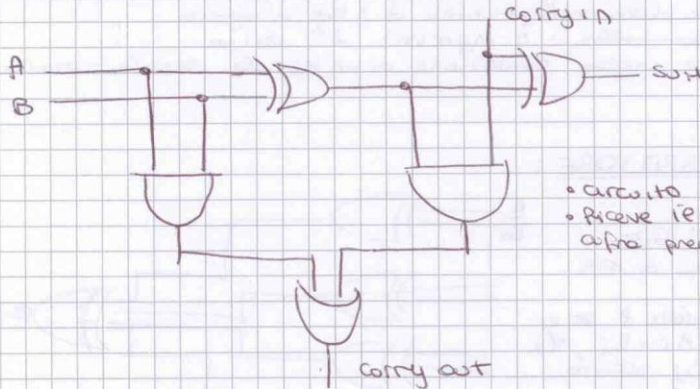
- il segnale c determina il verso dello shift

SEMI ADDIZIONATORE (HALF-HALF-ADDER)



- Circuito a 2 ingressi e due uscite: somma e riporto (Carry)
- Non può essere usato per la somma di numeri a più bit, dove occorre sommare anche il riporto della cifra precedente

ADDIZIONATORE COMPLETO (FULL-ADDER)



- Circuito a 3 ingressi e 2 uscite
- Riceve il riporto dalla cifra precedente

23/03/12

ALU AD N BIT :

XIV lezione

- Realizzato connettendo n ALU ad 1 bit
- Problema: ciascuno stato deve attendere il riporto dal precedente
- Tempo di addizione cresce con n

CIRCUITI SEQUENZIALI: Le uscite del circuito dipendono dagli ingressi e dallo stato passato → riassunto nello stato che è codificato nelle variabili di Stato → sono memorizzate in elementi di memoria binari

↳ circuiti combinatori calcolano le uscite e il nuovo valore dello stato

CLOCK: tutti i cambiamenti di stato vengono sincronizzati da un segnale di clock → da un segnale di clock primario vengono ricavati tutti gli altri per sfasatura

- Le transizioni di stato del circuito possono avvenire:
 - 1) In corrispondenza dei livelli
 - 2) In corrispondenza dei fronti

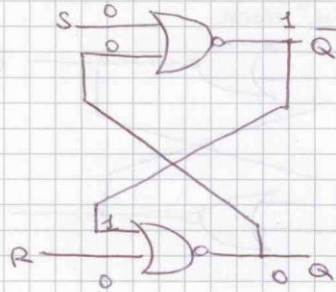
serve a temporizzare le operazioni → tutte le cose che fanno succedere hanno una cronologia

Calcolatori Elettronici

23/03/12

XIV lezione

LATCH :



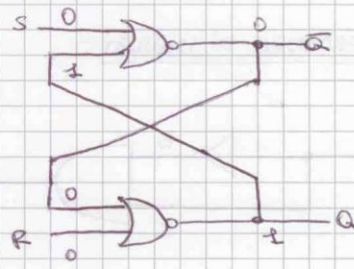
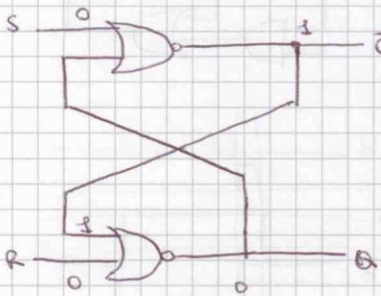
A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

Ⓛ dispositivo di memoria ELEMENTARE

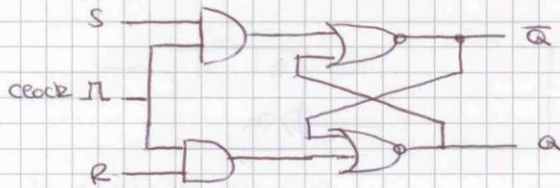
- due stati stabili $Q=0$ e $Q=1$
- S (SET) forza Q a 1
- R (RESET) forza Q a 0
- con $S=R=0$ il circuito mantiene lo stato

il latch si ricorda che è l'ultimo operatore che ha fatto il set

LO STATO DELLA LATCH può essere uno di questi due:

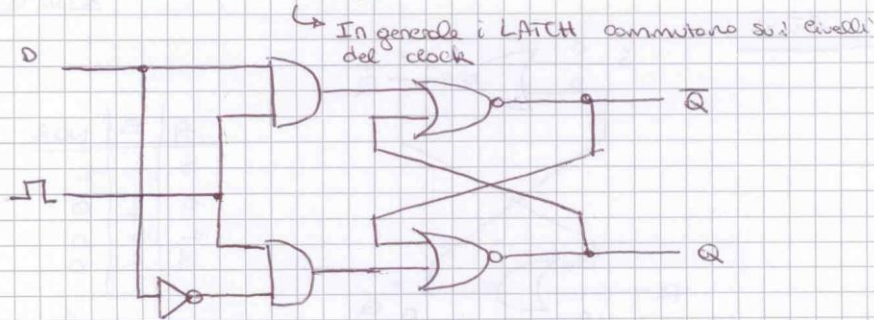


LATCH CON CLOCK :

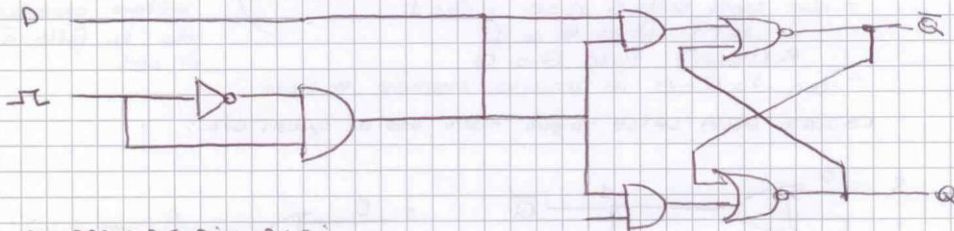


- R ed S vengono trasferiti sugli ingressi del latch solo quando il clock è ad 1
- Quando il clock è a 0 vengono ignorati

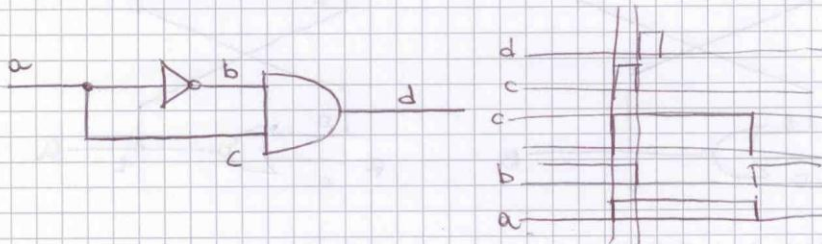
LATCH D (Delay) → quando il clock va ad il registro nello stato
 a il valore dell'ingresso D



FLIP-FLOP : È una variante del LATCH che commuta sui fronti del clock
 • sono gli elementi base di memorizzazione nel computer

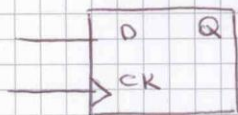


GENERATORE DI IMPULSI :

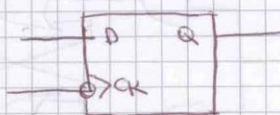


I FLIP-FLOP commutano sui fronti del clock :

I



Commuta sul fronte di salita



Commuta sul fronte di discesa

Molti Flip-Flop possono essere messi su un unico chip
 Occorrono in genere da 6 a 10 transistor per ogni Flip-Flop

Calcolatori Elettronici

26/03/12

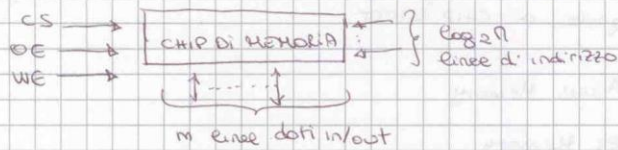
Lezione XV

DISPOSITIVI A 3 STATI :



- In base ad un segnale di controllo C si comporta :
 - C=1 come circuito chiuso
 - C=0 come circuito aperto
- Tempo di commutazione: pochi nsec.
- Consente di usare gli stessi pin sia per la lettura che per la scrittura → usato per connettere ai bus o linee bidirezionali

CHIP DI MEMORIA :



- Chip da $n \times m$ bit complessivi (n parole da m bit)
- m linee dati bidirezionali
- CS (chip select) ② OE (output enable) ③ WE (write enable)
 → ha un numero limitato di pin:

SEGNALI ASSERTITI E NEGATI : in alcuni casi un segnale provoca l'azione corrispondente quando la sua tensione è alta, in altri quando è bassa.

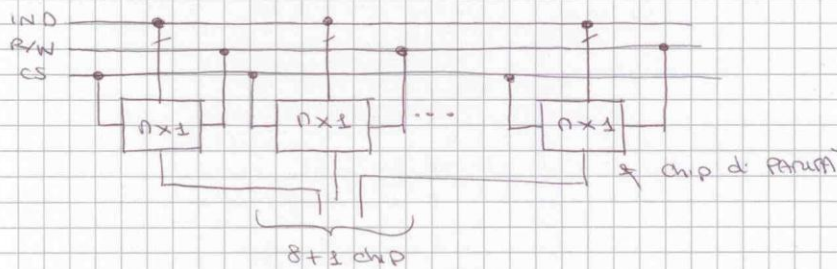
SEGNALE ASSERTITO : quando assume il valore che provoca l'azione

SEGNALE NEGATO : " non " " " "

S : segnale che è asserto alto

\bar{S} / S# : segnale che è asserto basso

SCHEDE DI MEMORIA :



- La scheda memorizza n parole di 1 byte
- Si usano 8+3 chip e si vuole il controllo di porta
- Bit di porta gestito dal controller della memoria

TIPLOGIE SCHEDE MEMORIA:

- SIMM (Single In-line Memory Module)
32 piedini, 32 bit, 8-16 chip, 128 MByte
- DIMM (Double In-line Memory Module)
168/184 piedini, 64 bit, 16 chip, 256 MByte
- SO-DIMM (Small Outline) → per i notebook

alcune hanno le
br di porta altre
no

Il controller può gestire più schede:

- ogni scheda informa il controller della sua dimensione
- Il controller determina al momento del boot il tipo di RAM
- Al boot, l'indirizzo e dalla configurazione il controller calcola a quale scheda mandare il segnale di CHIP SELECT

RAM : Random Access Memory

ROM : Read Only Memory

SRAM : Static RAM : a Flip-Flop, molto veloce

DRAM : Dynamic RAM : basata su capacità parassite, richiede refresh, alta densità, basso costo

→ EPROM : Selezione a matrice

→ EDO : Extended Data Output, lettura in pipeline, più banda

↳ SDRAM : (Synchronous DRAM)

PDR : Double Data Rate - Lettura/scrittura in pipeline → velocità 3,2 Gbps

Calcolatori Elettronici:

27/03/12

ESECUZIONE:

Lezione XVI

Esercizio 1 : circuito combinatorio che effettui un controllo di parità su tre linee digitali

- Realizz. il circuito mediante PORTE LOGICHE
- Indicare come bisogna trasformare il circuito ottenuto per ottenere un circuito equivalente contenente solo porte NAND
- Realizzare il circuito con un singolo MULTIPLEXER

Soluzione : trova la soluz. facendo la tabella

A	B	C	OUT
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

$$P = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

↳ per l'eqn. algebrica prendo in considerazione solo gli 1

↳ segue la legge del bit di parità ovvero 0 1 quando la somma degli 1 è pari

Esercizio 2: Realizzare un circuito combinatorio che ha in ingresso tre segnali

A, B, C e si comporta come segue:

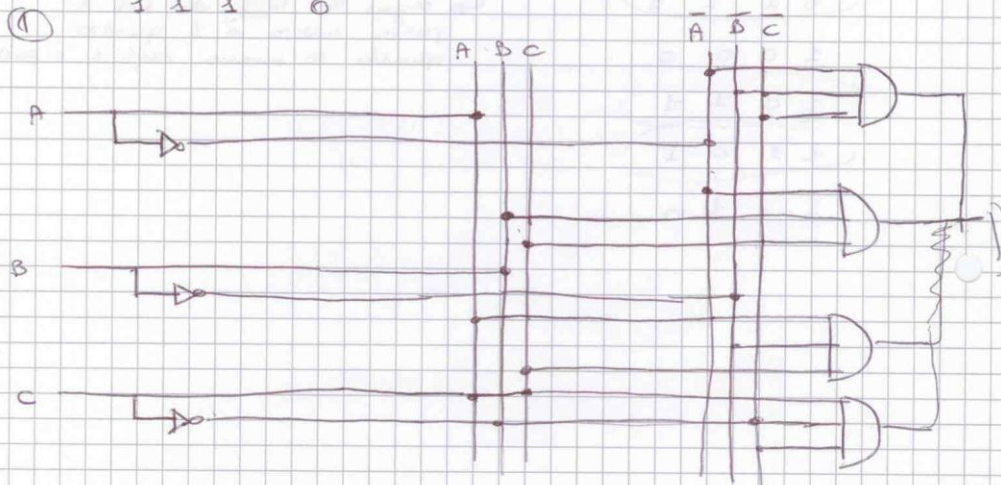
- a) quando $C=0$ fa un test di disuguaglianza ovvero restituisce 1 se A e B sono uguali e 0 altrimenti
- b) quando $C=1$ fa un test di disuguaglianza ovvero restituisce 1 se A e B sono diversi e 0 altrimenti

- ① Realizzare, mediante porte logiche qualunque
 - ② Realizzare il circuito utilizzando solo NOT e XOR
- Indicare come si può realizzare il circuito con una PLA (ovvero 3 ingressi, 5 porte AND e 2 porte OR)

Soluz.

$$P = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

A	B	C	OUT
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0



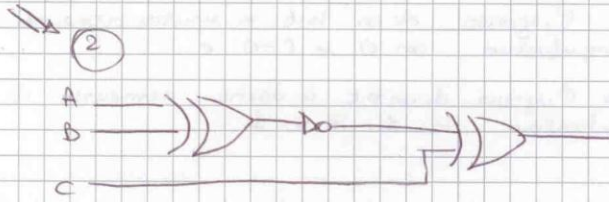
* TABELLA XOR

A	B	U
0	0	0
0	1	1
1	0	1
1	1	0

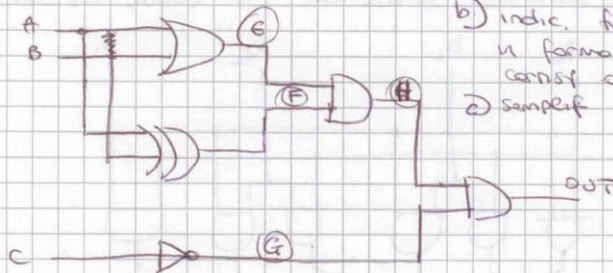
Calcolatori Elettronici

28/03/12

Esame XVI



ESERCIZIO 3



- a) Det. tav. verità?
- b) indic. funz. booleana
 in forma canonica
 consist. alla tabella
 e semplif.

OR

A	B	U_1
0	0	0
0	1	1
1	0	1
1	1	1

XOR

A	B	U_2
0	0	0
0	1	1
1	0	1
1	1	0

AND

U_1	U_2	U_3
0	0	0
1	1	1
1	1	1
1	0	0

U_3	C	U_4
0	1	0
1	1	1
1	0	0
0	0	0

Tab

ABC	EFG	H	OUT
000	001	0	0
001	000	0	0
010	111	1	1
011	110	1	0
100	111	1	1
101	110	1	0
110	101	0	0
111	100	0	0

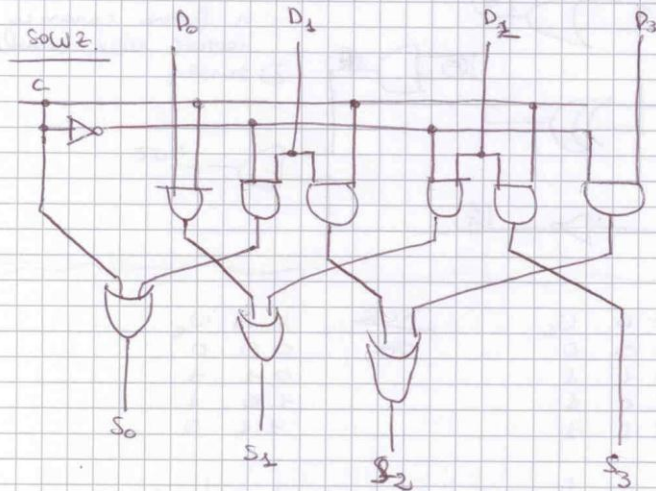
$$\overline{ABC} + A\overline{BC} = (\overline{AB} + A\overline{B})\overline{C} = (\overline{A \oplus B})\overline{C}$$

ESERCIZIO 4 : Fornire lo schema di uno SHIFTER a 4 ingressi e 4 uscite che sulla base di un segnale di controllo C :

- a) Sposta l'ingresso di un bit a sinistra rimpicciando i bit meno significativi con 0 se $C=0$ e
- b) sposta l'ingresso di un bit a destra rimpicciando i bit più significativi con 1 se $C=1$

(PROCEDIMENTO)

Mi faccio lo schema di uno SHIFTER normale poi lo modifico a seconda di quello che mi chiede



Collocazioni Elettronica

29/03/12

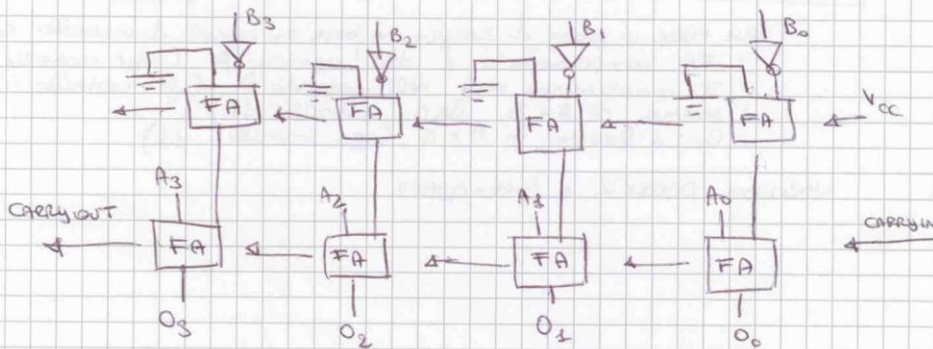
ESERCITAZIONE

Lezione XVII

Esercizio 1 : Fornire lo schema di un sottrattore a 4 bit per notazione in complemento a 2 realizzato con sommatore completi

- Illustrarne concisamente il funzionamento
- Specificare il valore di uscita di ciascuna componente quando A è un ingresso (minuendo) e B è il numero 0011 e nell'altro (sottraendo) il numero 0100

SOLUZIONE



Esercizio 2 : Schema di un circuito combinatorio a 4 bit → che calcola il valore assoluto di un numero, secondo il sistema di rappresentazione in complemento a 2. Tale circuito, ricevuto in ingresso un numero X a 4 bit deve restituire in uscita

- Lo stesso numero in ingresso, se X rappresenta un numero positivo
- Il numero in ingresso con segno invertito, se X rappresenta un numero negativo (per esempio, se $X = 3$ allora l'uscita vale $X = 3$ oppure se $X = -3$ allora l'uscita vale $X = 3$)

Si possono utilizzare HALF-ADDER / FULL-ADDER. Illustrare il funzionamento del circuito e specificare il valore di uscita di ciascuna componente quando l'ingresso si trova a 1010110

SOLUZIONE

- Utilizzo due porte AND di ABILITAZIONE poste alla fine

- 1) mettere sempre le porte di abilitazione sempre alla fine

ESERCIZIO 3 : Schema di una ALU in grado di effettuare :

- somma aritmetica
- complemento a 1
- complemento a 2
- shift circolare (cambio) di due bit di ingresso

SOLUZ. USATO SU4E

ESERCIZIO 4 - Progettare ALU due operandi in ingresso da 8 bit (A e B)

deve essere in grado di scegliere in base ai segnali di controllo e operazioni

- IF complemento a 1 dell'operando A (segn controllo: 00)
- IF complemento a 2 dell'operando B (segn controllo: 01)
- somma di A e B (segn controllo: 10)
- la differenza di A e B (segn controllo: 11)

Utilizzare DECODER e FULL-ADDER

Calcolatori Elettronici

12/04/12

PIN-OUT : PIEDINATURA

Lezione XVIII

INTERRUPT : consentire ai controllori esterni di inviare un segnale asincrono al processore

i bus sono di vari tipi \rightarrow quelli memoria - processore - scheda video ~~sono~~ utilizzano bus dedicati

\hookrightarrow in ^{ciclo} ~~ogni~~ tipo di BUS comunicano solo 2 elementi disponibili

- IL MASTER
- LO SLAVE

\hookrightarrow i dispositivi sono connessi al bus tramite un bus transceiver

LARGHEZZA DEL BUS = numero di linee

LINEE INDIRIZZO : dimensione dello spazio indirizzabile 2^n locazioni con n bit di indirizzo

LINEE DATI + VELOCITÀ DI TRASMISSIONE : banda di riferimento

BUS SKEW : problema dovuto alla differenza nella velocità di propagazione dei segnali su linee diverse

BUS SINCRONO :

ES : 300 MHz periodo 10 nsec

$T_{AD} = 4$

13/04/12

Lezione XIX

PROCESSORI P4 - ULTRASPARE III

16/04/12

Lezione XX

PROCESSORE 8051

incorporato EMBEDDED : 60.000 transistors
freq. < 100 MHz

→ possiede una RAM da 4-8K
Può gestire un max di 16 indirizzi } Ha una scheda con 40 pin

BUS ISA del PC/AT

↳ non viene più utilizzato (Industry Standard Architecture)
viene usata un'estensione fisica del BUS per la compatibilità

BUS PCI : (Peripheral Component Interconnect)

↳ molto utilizzato → BUS SINCRONO
→ introdotto da Intel per le applicazioni video
→ connessione tramite chip PCI bridge
→ BRIDGE ISA (include doppio controller IDE)
→ linee di indirizzo e dati condivise

Arbitraggio centralizzato (Bridge)

Diversi algoritmi di arbitraggio
• Round Robin: un po di tempo a tutti
• Pronto

master & slave → Initiator & Target

BUS PCI EXPRESS

Funzione completamente diversa → bypassare il bus PCI nel caso di periferiche veloci.
Usa slot più piccole → comunicazioni dedicate

↳ Soluzione: connessione punto-punto con una trasmissione seriale → spazio bit a velocità altissime, molto più avanti di quella in parallelo

↳ dimensioni molto ridotte

I dati vengono spezzati in pacchetti e inviati in uscita → i dati hanno un'interazione → un canale attivo (full duplex)

Banda: 8 Gbps

PCI EXPRESS PROTOCOL STACK

- Trasmissione basata su protocollo multibyte lungo coppie di corse
- Software layer garantisce: ① la gestione dei pacchetti ② Compatibilità con i sistemi
↳ il software viene eseguito dal controller e non dalla CPU

Calcolatori Elettronici

16/04/12

Lezione XX

↳ Affidabilità → la transazione deve essere garantita senza errori. Vi è un meccanismo acknowledgment che garantisce maggiore affidabilità.

CODIFICA: 16 byte 8 bit → diventano 10 bit
onde evitare segnali scomodi di tutti 0

BUS USB (Universal Serial Bus)

economico per la gestione di dispositivi I/O (novembre 1995)
Non obiettivi, tra cui evitare switch, installazione di tipo esterno
cavo di connessione unificato, alimentazione fornita dal cavo, fino a
127 dispositivi collegabili real-time, installazione a PC acceso,
Reboot non necessario.

- ↳ bus seriale composta da 4 linee: alimentazione, terra
+ 2 linee di segnale
- ↳ controllare ^{principale} ROOT HUB → è possibile collegare tra loro

17/04/12

tutti i dispositivi sono connessi
parallelamente al ROOT HUB

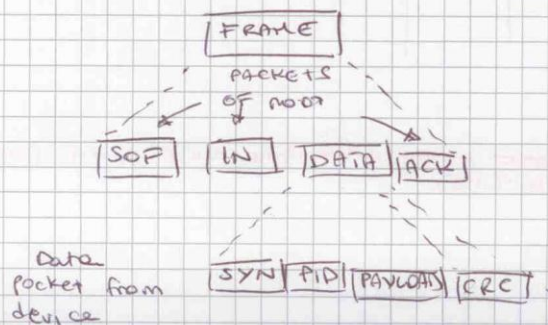
Lezione XXI

A diff. del BUS PCI EXPRESS (no velocità costante)
invece per il BUS USB la velocità è fissa → ogni microsecondo
vengono inviati dei bit (frame)

↳ OBIETTIVI: praticità ed economicità

SOF → pacchetto (frame) in cui non viene inviato nulla

↳ START OF FRAME → sincronizzazione



CHIP I/O : UART, USART e PIO



UART /SART

• usati in interfacce parallelo/serie

→ presenta 4 registri
 3 per le porte / 1 per lo stato

PIO : configurabile CPU

3 porte indipendenti da 8 bit

La CPU legge e scrive direttamente nella porte

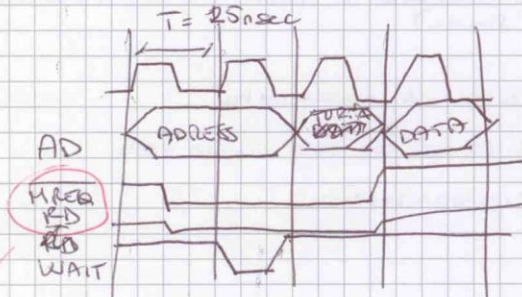
DECODIFICA DEGLI INDIRIZZI

MECCANISMO MEMORY-MAPPED → unico spazio non condiviso tra i vari dispositivi

ESERCIZIO

A) BUS SINCRO $\frac{1}{T} = 40 \text{ MHz}$ $\overline{MREQ}, \overline{RD}, \overline{WAIT}$ $t_m = 40 \text{ nsec}$

B) BUS ASINCRONO $\overline{MREQ}, \overline{RD}, \overline{MSYN}, \overline{SSYN}$ $t_m = 5 \text{ nsec}$

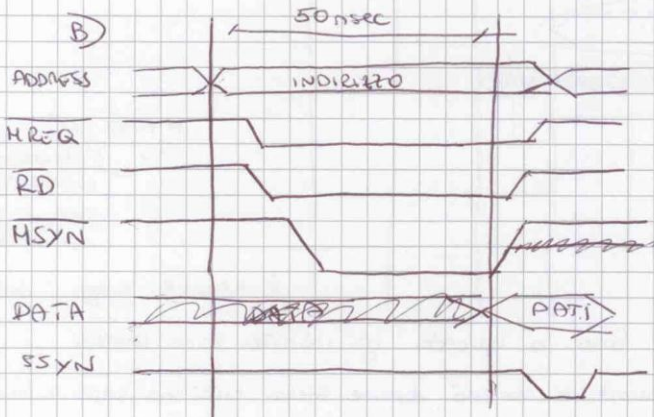


→ vengono assenti quando prima della fine delle attiv. prelievo dei dati e A-CAS

Coecobron Elettronici

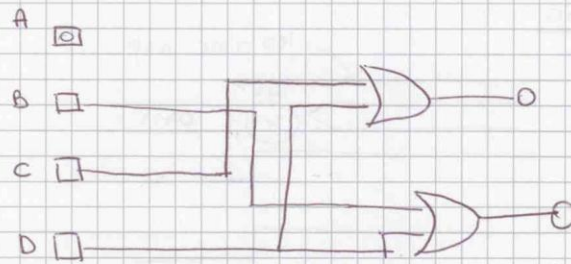
17/04/12

esone XXI



ESERCITAZIONE:

HOME 1



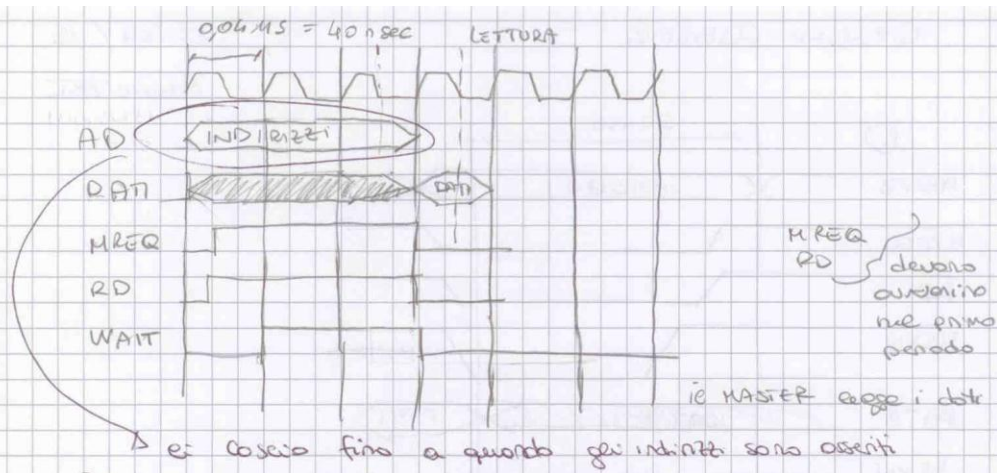
ESERCIZIO SIC BUS

19/04/12

BUS sincrono a 25 MHz con linee separate tra dati e indirizzi

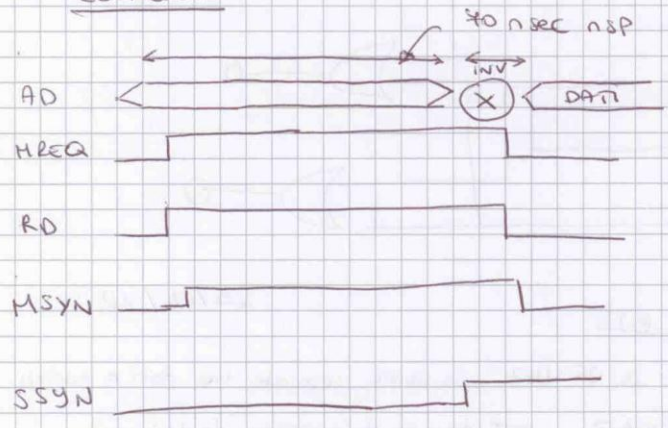
HREQ, RD e WAIT → tempo di risposta di 110 nsec

$$T = \frac{1}{25} = 4 \cdot 10^{-8}$$



- ⚠ - Il segnale di wait non può durare mezza ciclo ma bensì a metà ciclo
- MREQ, ED RD devono essere assenti prima della fine del primo periodo
- Gli indirizzi devono essere presenti fino a che i dati non diventano presenti nel bus

ESERCIZIO



Calcolatori Elettronici

20/04/12

Lezione XIII

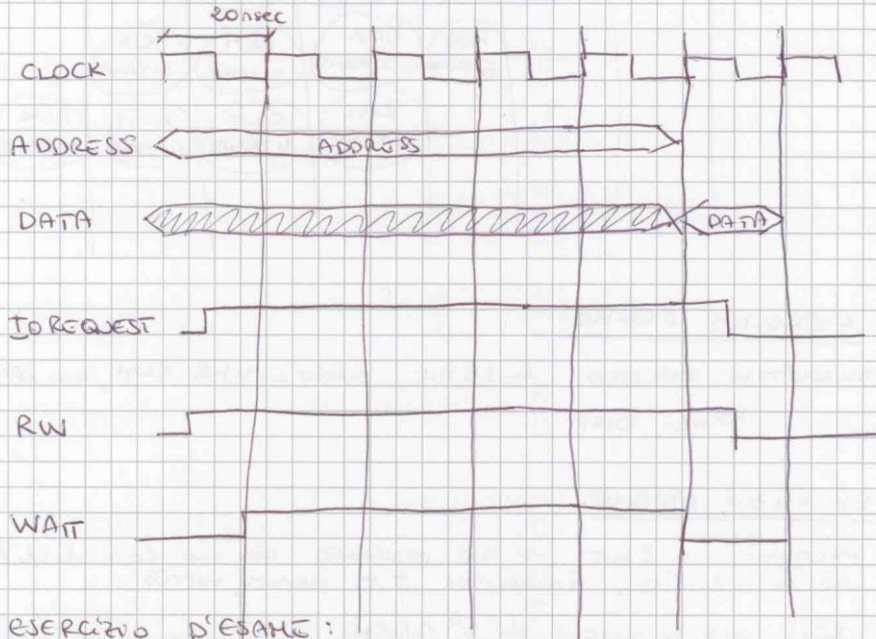
Esercizio d'Esame

con dispositivo I/O

Bus sincrono

→ 50MHz

$t_{risposta} = 100 \text{ nsec}$

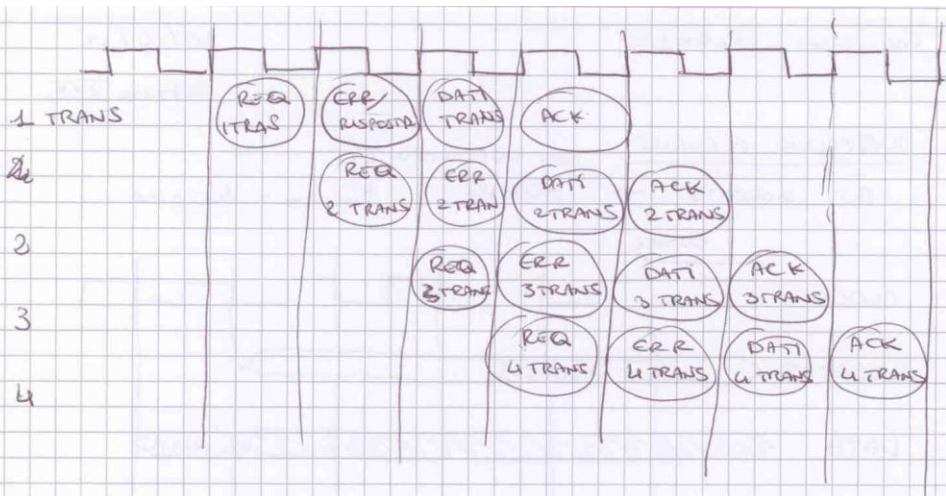


Esercizio d'Esame:

Bus di memoria per una CPU a 32 bit a 500MHz e
funziona a 10 Pipe Line

Individuare una decomposizione

- 32 bit (1° FASE = RICHIESTA
- 32 bit (2° FASE = ERRORI / RISPOSTA
- 32 bit (3° FASE = DATI
- 32 bit (4° FASE = ACKNOWLEDGMENT



ESERCIZIO D'ESAME

CALCOLATORE EMBEDDED a 16 bit dotato di: 4KB ROM, una RAM di 8KB
 SOWE. DSP

ESERCIZIO D'ESAME

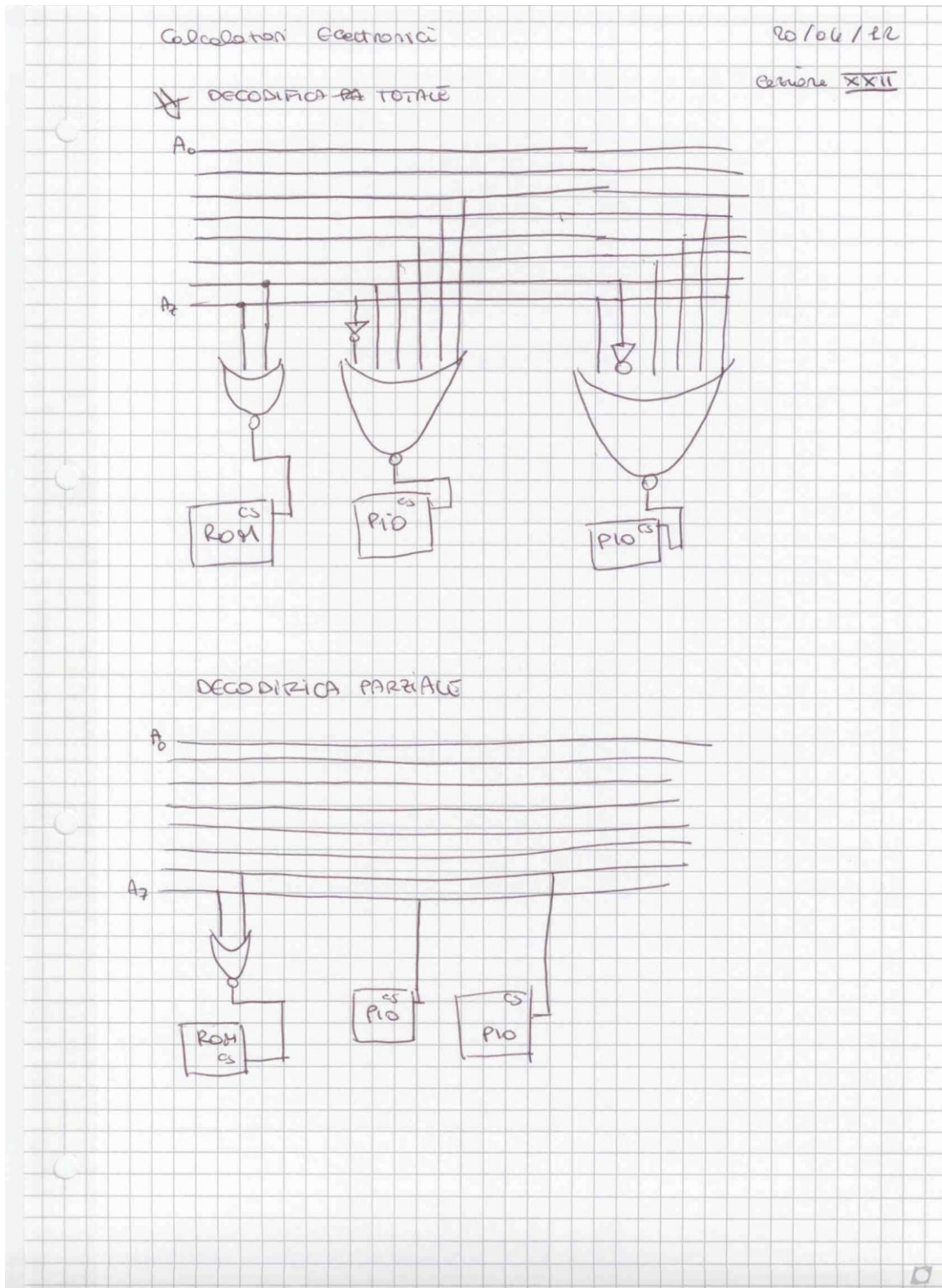
microproc. a 8 bit → bus parallelo con una ROM di 64KB e porta
 us di 2 PIO, 6 dispositivi I/O memory MAPPED

ROM → ha bisogno di 2⁶ indirizzi

ROM { 0000 0000
 0011 1111

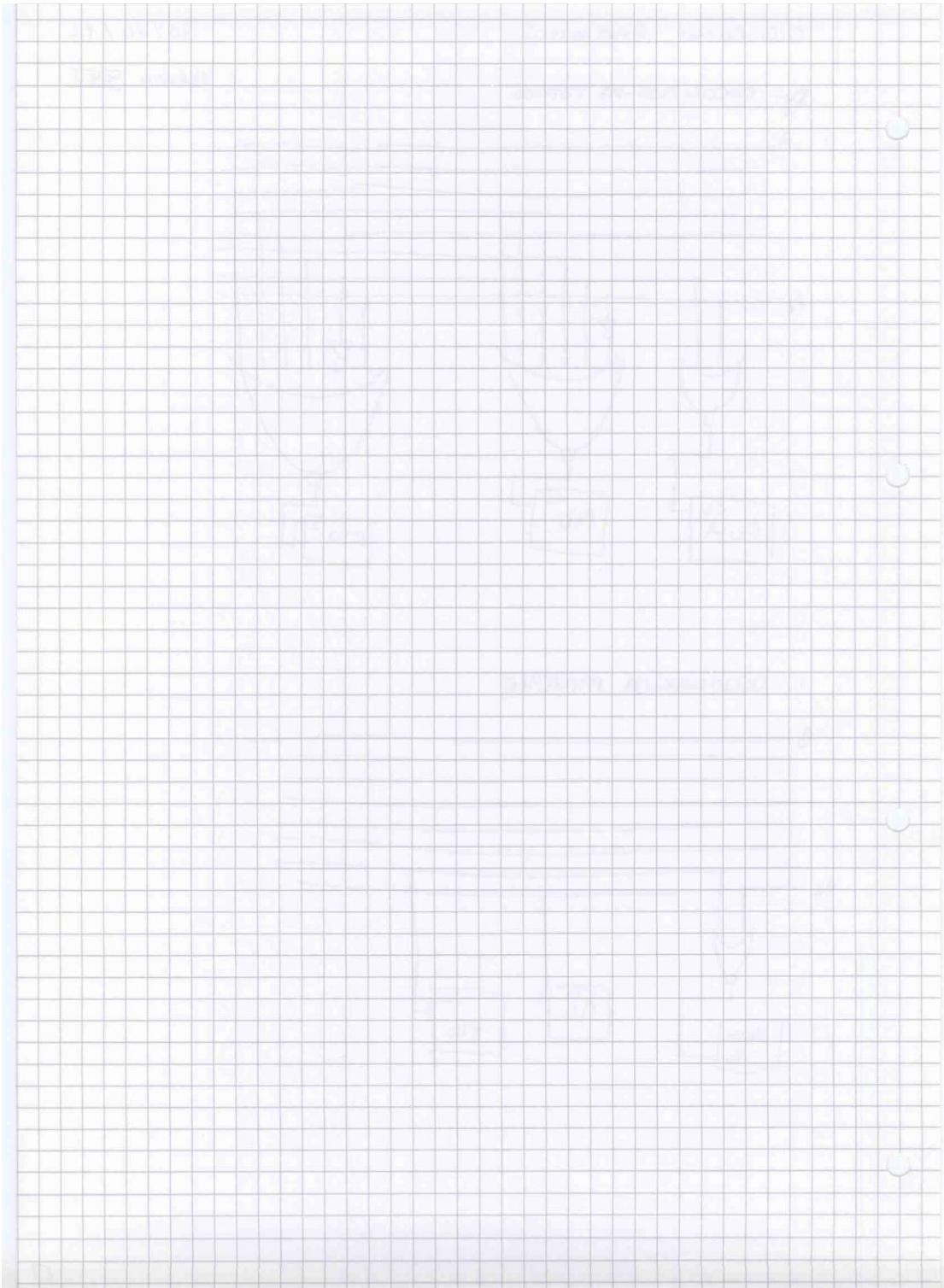
1° PIO { 1000 0000
 1000 0011

2° PIO { 0100 0000
 0100 0011



Sorgente : <http://hol.si.hasanaj.com/>
Autore : Holsi Hasanaj
Professore: Riccardo Torlone

Corso di Ingegneria Informatica Roma Tre
Anno di produzione: 2011-2012



Calcolatori Elettronici

23/04/12

ESERCITAZIONE

Lezione XXIV

Esercizio 1 16 bit / 1 sign 0=pos / 7bit esp in ecc 84
 8 bit m

m C E 3 5
 1100 1110 0011 0101
 1 100 1110 0011 0101
 ~~~~~  
 sign      esp      mant  
 ( 0001110 →  $-2^{14} (1,00110101)$

4      9      F      2  
 0100 1001 1111 0010  
 0 1001001 11110010  
 ~~~~~  
 sign esp mant
 (0001001 → $2^9 (1,11110010)$

a) m+n $2^{14} (0,00011110010)$
 - $2^{14} (1,00110101)$

 10100010

Le mantisse sono in valore positivo

bit di controllo per vedere se è pos o neg
 01, 00110101
 00, 00001111
 10, 11001011

 10, 11011010
 CP2 = cambio segno
 CP2

RESULTATO

1 1001110 00100110

B) $n \times m$

$$2^{14} \cdot 2^9 = 2^{23}$$

$$10 : 3 = 23 : x$$

$$[2 \cdot 10^{-6} \approx 10^{-6}]$$

$$\hookrightarrow 2^{24} \text{ max}$$

C) $r = m \cdot 2^{-5}$ in CPL a 16bit

$$- 2^{14} (1,00110101) \cdot 2^{-5}$$

$$\hookrightarrow - 2^9 (1,00110101)$$

$$\textcircled{\ast} 8+4+3$$

$$\text{CPL} \left(- (1001101010) \right)$$

$$\text{CPL} \left(\text{01100101010} \right)$$

$$- (00000001001101010)$$

$$\text{CPL} \left(111110110010110 \right)$$

$$\frac{13}{2}$$

$$\frac{13}{2}$$

D) COPPIE IN VIRGOLA NONIVE

$$\underbrace{1}$$

$$\underbrace{1001001}$$

$$\underbrace{00110101}$$

Calcolatori Elettronici

83/04/12
 come se ~~12~~ in esp a 64

ESERCIZIO 2 : 40 bit / 1 sign (0=pos) / 7 bit ^{exp} / 12 bit mant

denormalizzato = -63

A) Più piccolo normalizzato / Più grande denormalizzato

B) $n = FF323B$ (FP)

$$m = n \cdot 2^{-85}$$

C) ERRORI

⊕ e^e esponente con tutti 0000 → numeri denormalizzati

A) Più piccolo dei normalizzati

$$\frac{+/-}{\sqrt{\quad}} \frac{0000001}{\quad} \frac{00000000000000}{\quad} = 2^{-63}$$

PIÙ GRANDE DEI DENORMALIZZATI

$$\frac{+/-}{\sqrt{\quad}} \frac{0000000}{\quad} \frac{11111111111111}{\quad} = 2^{-63}$$

-63

B)

F F 3 2 3 B C
 1111 1111 0011 0011 0011 1101

decap2/

$$- (0000\ 0000\ 1100\ 1110\ 1100\ 0011)$$

$$= 2^{-15} (1,10011101100011)$$

$$= 2^{-15} \cdot 2^{-85} (1,10011101100011)$$

$$= 2^{-70} (1,10011101100011)$$

⊕ e'è pos/noma repr. come no denormalizzato $[2^{-63} \rightarrow 2^{-70}]$

$$= 2^{-63} \cdot 2^{-7} (1,100110110001101)$$

$$= 2^{-63} (0,00000110011011)$$

RAPPN

$$\frac{+/-}{\sqrt{\quad}} \frac{0000000}{\quad} \frac{00000110011}{\quad}$$

CERORE ASSIUTO
 CIFRA DA RAPPRE

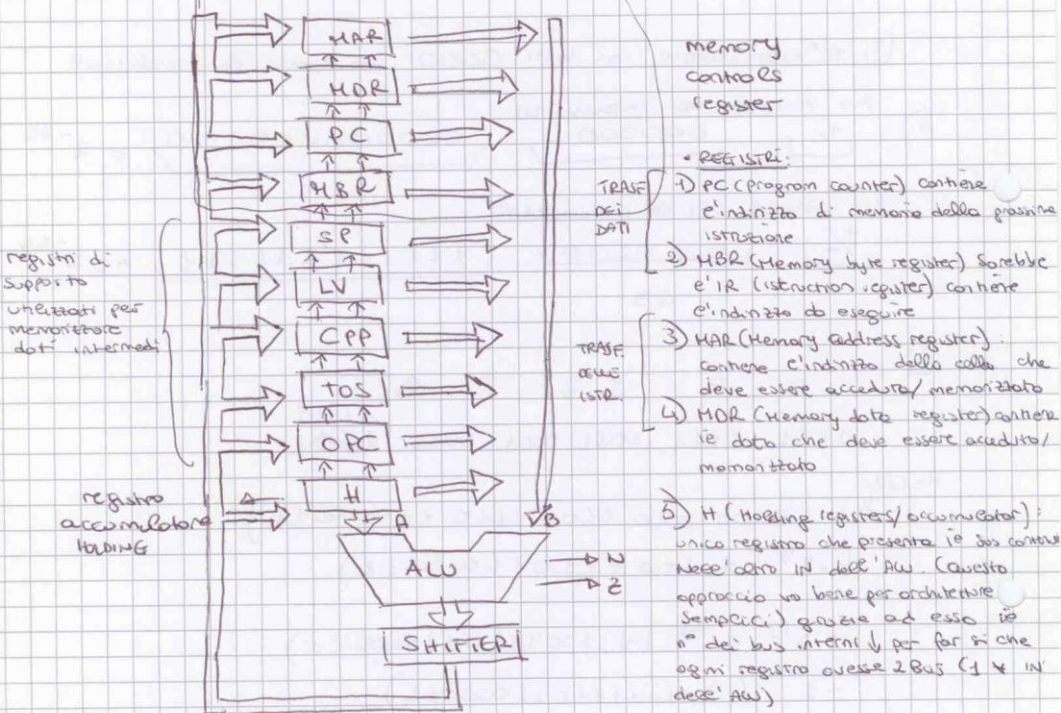
XXIV lezione
 3/05/12

LIVELLO DI MICROARCHITETTURA

In un architettura microprogrammata le istruzioni macchina non sono eseguite direttamente dall'hardware

M - ARCHITETTURA : Implementazione di un JVM con sole istruzioni di LALR

CAMMINO DATI JVM (DATA PATH)



SEGNALI DI CONTROLLO:

- 1) ↑ ABILITATORI : servono a determinare le porzioni dei dati sul Bus
 OSS : se 1 ↑ può essere assente contemporaneamente
- 2) ↑ ABILITATORI : servono a consentire la scrittura dei dati in cui dell'ALU sui registri
 OSS : può essere assente anche più d'una contemporaneamente

OSS - MBR è più piccola perché invece di 32 byte ne può contenere solo 1. Le 2 frecce blu scure : 1 serve per estendere i dati per poter permettergli di essere inviato, e' l'altra come semplice abilitazione

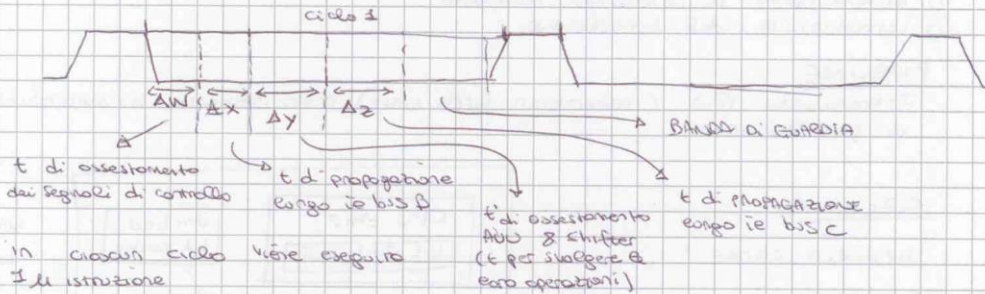
Calcolatori Elettronici

4/05/12

XXV lezione

TEMPORIZZAZIONE DEL CICLO BASE :

tutte le operazioni d'un ciclo devono avvenire in un ciclo di clock ω



In ciascun ciclo viene eseguita 1 istruzione

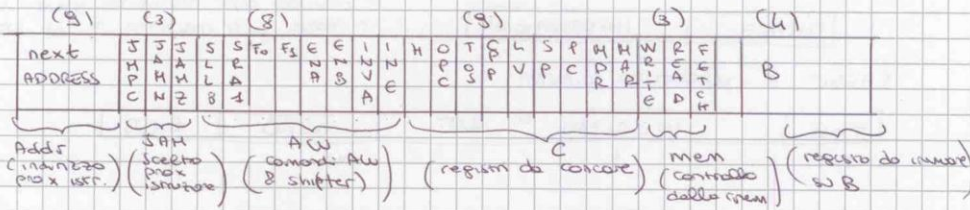
oss: nelle risc 3 istruzioni machine viene eseguita direttamente \rightarrow architettura non c'è

STRUTTURA DELLE μ -ISTRUZIONI: una micro istruzione contiene tutti i segnali di controllo

da inviare ai datapath durante il ciclo.

oss: dim. μ istruzioni $\uparrow \rightarrow$ no segnali di controllo \uparrow

In questo caso la DIM è 32



Il meccanismo mostrato in figura è quello che permette di trasformare un'istruzione in una micro istruzione

- Si mette l'istruzione nel MPC (micro-program-counter) registro che sceglie la locazione di una memoria interna, ovvero la control STORE, una memoria ROM (in questo caso $2^8 = 512$ celle ciascuna, con una locazione di 32 bit di m(MPC) = 32 bit \rightarrow memoria ha dim 29
- Addr con la nuova istruzione quando $A \neq 1$ allora verranno caricate le altre
- J: indica che a meno di imprevisti la prox istruzione da eseguire è nella Addr se è nastro è no \rightarrow potrebbe ad un'altra istruzione

oss: l'ordine di esecuzione delle micro istruzione non è strettamente sequenziale

JAHN \rightarrow cambio di stato 1°
 JAHZ bit se è nastro dell'AW è 0 o 1

JAHPC: indica che bisogna prelevare una nuova istruzione da MBR

Come fanno a migliorare prestazioni? \uparrow velocità? non solo ma max vel prezzo

- 1) Introdurre componenti dedicate che svolgono determinate operazioni (\downarrow cicli del datapath)
- 2) \uparrow ϕ ck (ormai ha perso importanza)
- 3) Introduzione parallelismo (pipelining/duplicazione componenti) \leftarrow più importante

velocità
prezzo

- 3 opp. fondamentali: - Componenti dedicate ①
 - Aumento f ②
 - Parallelismo ③

7/05/12
 XXVI lezione

Quanto alla velocità del processore

- 1) introduzione di componenti dedicate
- 2) introduzione del parallelismo

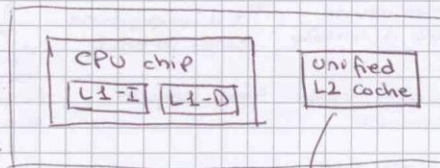
PIPELINE

Introducendo IFU (Instruction fetch unit) posso introdurre un meccanismo di parallelismo

CACHING

MEMORIA CACHE:

Scoperto dallo scop. cache: disaccopp. la velocità tra la memoria e la CPU



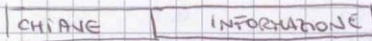
Unified L2 cache

ORGANIZ. DELLA MEMORIA IN PRESENZA DI CACHE:
 → trasferimento che avviene per blocchi

possono contenere più dati che L1 unit.

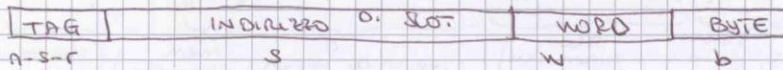
Ricerca di una locazione di memoria

MEMORIE ASSOCIATIVE



c'è accesso ad ogni elemento viene e (effettuata) in base all'indirizzo o al suo contenuto

CACHE A MAPPATURA DIRETTA



ESEMPIO:



Indirizzo a 8 bit ($n=8$)
 linee di cache a 8 byte ($r=3$)
 word di 2 byte
 cache di 8 slot

Calcolatori Elettronici

08/05/12

~~Esercitazione 1~~ ~~Calcolatori~~

XXVII lezione

CACHE ASSOCIATIVA AD INSIEMI: (o a più vie)

- Ogni slot è costituita da n elementi, ciascuno composto di bit valid, tag e blocco
- Un blocco può stare in un elemento qualsiasi dello slot che gli corrisponde.
- Allevia il problema della collisione di più blocchi sullo stesso slot

① aggiungo un TIME STEP che si mette nel momento in cui il blocco entra nella cache

LRU (least ^{recent} recently used)

GESTIONE DELLA CACHE

- La CPU deduce numero di slot e TAG del blocco a partire dall'indirizzo ~~viene fatto~~

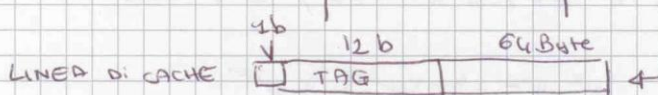
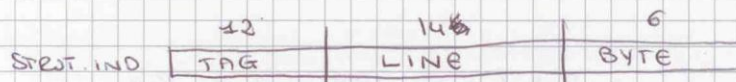
ESERCITAZIONE:

ESERCIZIO 1

$$16K \text{ slot} = 2^{14} \text{ slot}$$

$$64B \text{ cache line} = 2^6$$

32 bit indirizzo



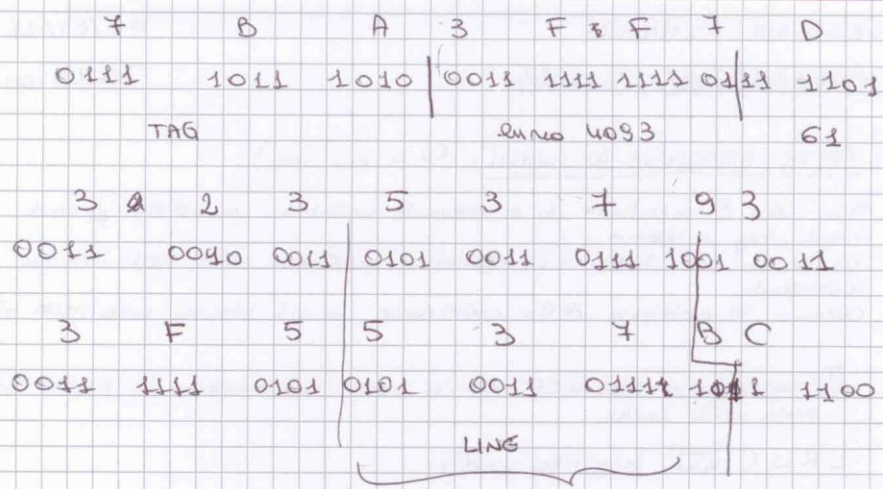
$$64 \cdot 2^6 \cdot 2^3 + 13 \text{ bit} = 525 \text{ bit}$$

$$\text{dimensione CACHE} = 525 \cdot 16 \cdot 2^{10} = 2^{23} \text{ bit}$$

- 4BA3FF4D
- verificare collisioni

32353793 e 3F5537BC



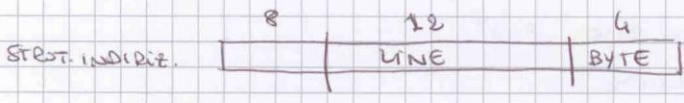


① Collisione = quando ho collisione LINE
 Calcolano perché la linea è collisione

ESERCIZIO 2 : → non importa la via

- 4 way
- 24 bit
- 16 B
- 4K SET
- INDIRIZZO
- CACHE LINE

Strutture indirizzi?
 Strutt. cache?
 dispense



18 16B / 18 16B / 18 16B / 18 16B

DIMENSIONE = $2^8 \cdot 2^{12} \cdot 2^4 = 2^{24}$
 $(8b + 16 \cdot 2^3 b + 4) \times 4 = 548b = 68,5B$
 DIM CACHE = $68,5B \cdot 2^{12} \cong 2^{18}B$

ESERCIZIO 3

Calcolatori Elettronici

9/05/12

ESERCITAZIONE:

XXVIII lezione

ESERCIZIO 1 : MEMORIA CACHE III

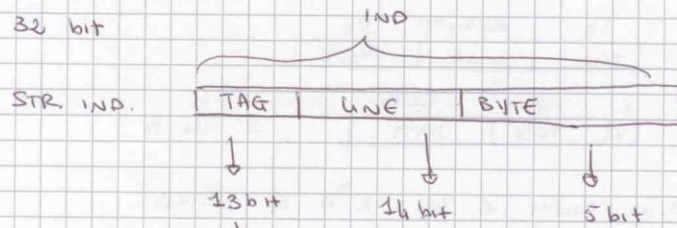
Cache Line 32 B

2^{13} blocchi di memoria

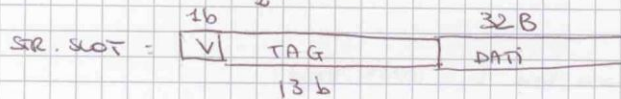
2 way

STRUTT. INDIRIZZO

32 bit



n° di collisioni
 2^{13}

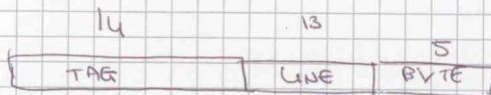


con una CACHE a più vie

n° collisioni A RAPPRESENTAZIONE DIRETTA = 2^{TAG}

TAG = 14

n° collisioni A PIÙ VIE = $\frac{2^{\text{TAG}}}{N\text{-VIE}} = 2^{13}$



ESERCIZIO 2 : MEMORIA CACHE III

32 bit → indirizzi

32 byte → line cache

4 MB → 2^{22} B

4 CACHE ≤ 32 KB

Calcolatori Elettronici

14/05/12
Lezione XXIX

IN ORDER - EXECUTION :

Registri SCRATCH → sono dei buffer nei quali si appoggia e inserita; dopo aver effettuato l'operazione posso ritirare la prossima istruzione

OUT-OF ORDER - EXECUTION : si rinuncia ad eseguire in ordine

le istruzioni → sfruttando maggiormente la pipe line

1) devo sempre vedere se dipendo anche da quelle in attesa. L'importante è che alla fine tutte le istr. vengono eseguite

Per ~~WAW~~ ~~WAR~~ me lo posso capire con i registri segreti meno per la RAW

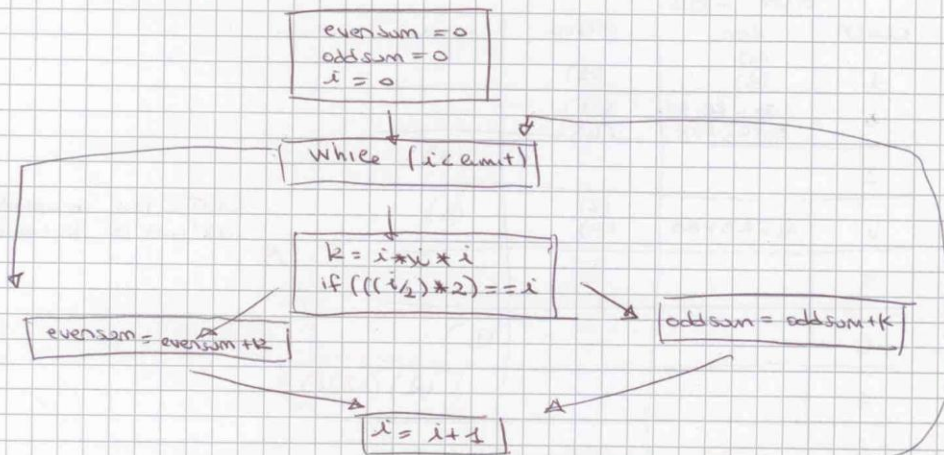
Per la RAW → devo rimanere in attesa

forte risarcimento in termini di prestazioni

ESEMPLO SPECULATIVE EXECUTION

```
even sum = 0;  
odd sum = 0;  
i = 0;  
while (i < limit) {  
    k = i * i * i;  
    if (((i/2) * 2) == i)  
        even sum = even sum + k;  
    else  
        odd sum = odd sum + k;  
    i = i + 1;  
}
```

Si preferisce eseguire entrambe le istruzioni anche se non verranno eseguite entrambe → per questo vengono dette speculative



ESERCIZIO 1

- (1) $R3 = R1 + R2$ ←
- (2) $R5 = R4 * R3$ (RAW)
- (3) $R2 = R6 - R4$ (WAR)
- (4) $R2 = R3 + R8$

10 registri ufficiali

2 archi di clock per computer: somme e differenze

3 archi di clock per computer: prodotti

AVVIO IN ORDINE ENTRO IN ORDINE

clock	decodifica	avvio	ritiro
1	(1) (2)	(1)	
2	(3) (4)		
3			(1)
4		(2) (3)	
5			
6			
7			(2) (3)
8		(4)	
9			
10			(4)

OUT - OUT

clock	dec	avvio	ritiro
1	(1) (2)	(1)	
2	$S2 = R6 - R7$	(3)	
3			(1)
4	$S3 = R3 + R8$	(2) (4)	(3)
5			
6			(4)
7			(2) (3) (4)
8			
9			
10			(4)

OUT - IN → uguale all'out in + nome per quello

Calcolatori Elettronici

15/05/12

ESERCIZIO - in ordine e fuori ordine

Versione ~~XXX~~

(1) $R_2 = R_2 * R_1$

(2) $R_5 = R_1 + R_3$

(3) $R_4 = R_6 - R_1$

(4) $R_5 = R_6 * R_3$

POLITICA IN - IN (outside in ordine interno in ordine)

	POLITICA	IN	- IN	ritiro
1	(1) (2)	(1) (2)		
2	(3) (4)		-	
3				
4				(1)(2)
5			(3) (4)	
6				
7				(3)
8				(4)

POLITICA OUT - OUT

		AVVIO	ritiro
1	(1) (2)	(1) (2)	
2	(3) (4)		-
3			(2)
4		(3) (4)	(1)
5			
6			(3)
7			(4)

	PUNTA		OUT-IN	
	AVVIO	FINITO	AVVIO	FINITO
1	(1)	(1)		
	(2)	(2)		
2	(3)	-		
	(4)			
3				
4			(4)	(2)
5		(3)		
		(4)		
6				
7			(3)	
8			(4)	

Esercizio

- (1) $R_2 = R_4 - R_1$
- (2) $R_3 = R_3 / R_2$
- (3) $R_3 = R_3 + R_1$
- (4) $R_5 = R_3 * R_3$

+/- = 3 celle
 */ = 4 celle

IN-IN è uguale ad OUT-OUT

	IN-IN		AVVIO		FINITO	
	AVVIO	FINITO	AVVIO	FINITO	AVVIO	FINITO
1	(1)	(2)	(1)	-		
2						
3						
4						(1)
5			(2)			
6						
7						
8						
9						(2)
10			(3)			
11						
12						
13						(3)
14			(4)			
15						
16						
17						
18						(4)

Calcolatori Elettronici

15/05/12

Lezione ~~XXX~~

CPU INTEL PENTIUM IV

ESTERNAMENTE

- cisc a 32 bit
- bus a 64 bit
- Set di istruzioni esteso e disordinato da 1 a 17 byte
- 8 registri

Internamente

- Nucleo RISC
- Lungo Pipeline
- Diverso da predecessori

MICROARCHITETTURA DEL PENTIUM IV

Sottosistema di memoria

Front-end

Controllo dell'esecuzione

CPU ULTRASPARC III

macchina RISC a 64 bit

bus a 128 bit

Set di istruzioni ridotto

17/05/12

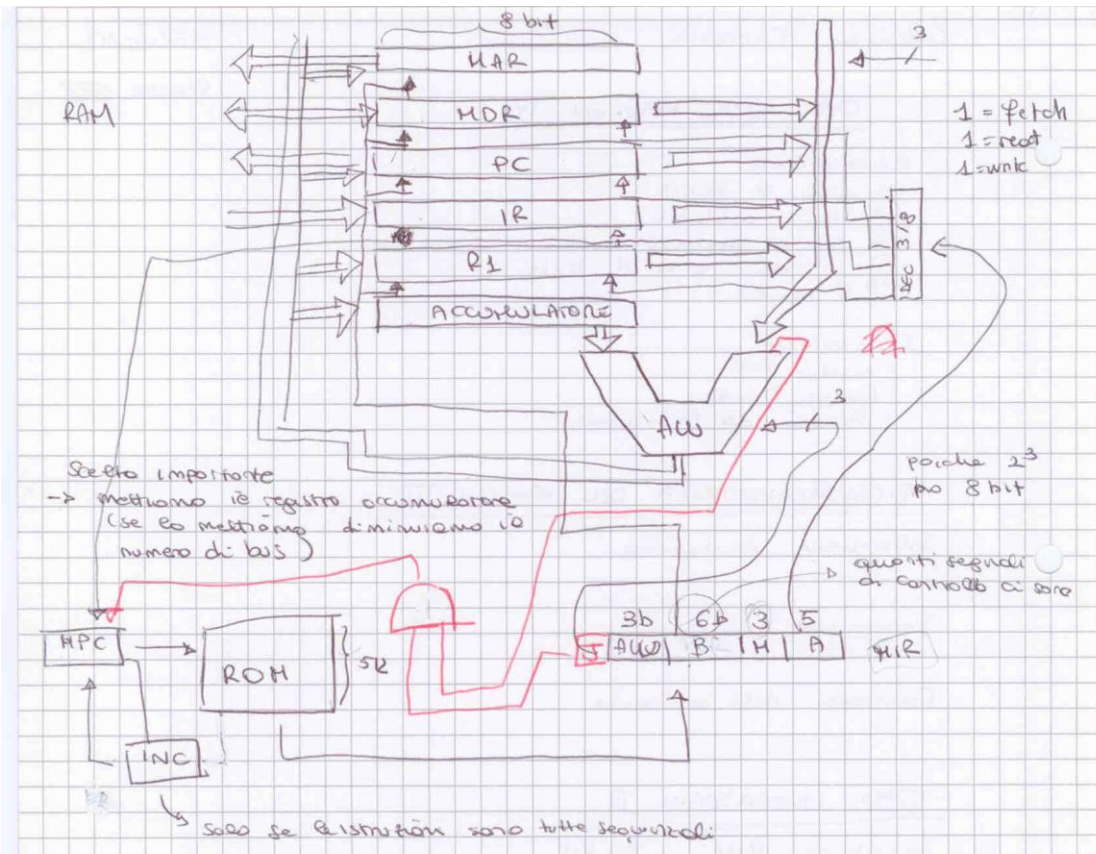
ESERCITAZIONE

ESERCIZIO SUE ARCHITETTURE DI CPU I

CPU CISC A 8 bit

- 1 registro general purpose
- 2 registri per le fetch di
- 2 registri per le transf. dati per la memoria
- ALU con 8 operazioni



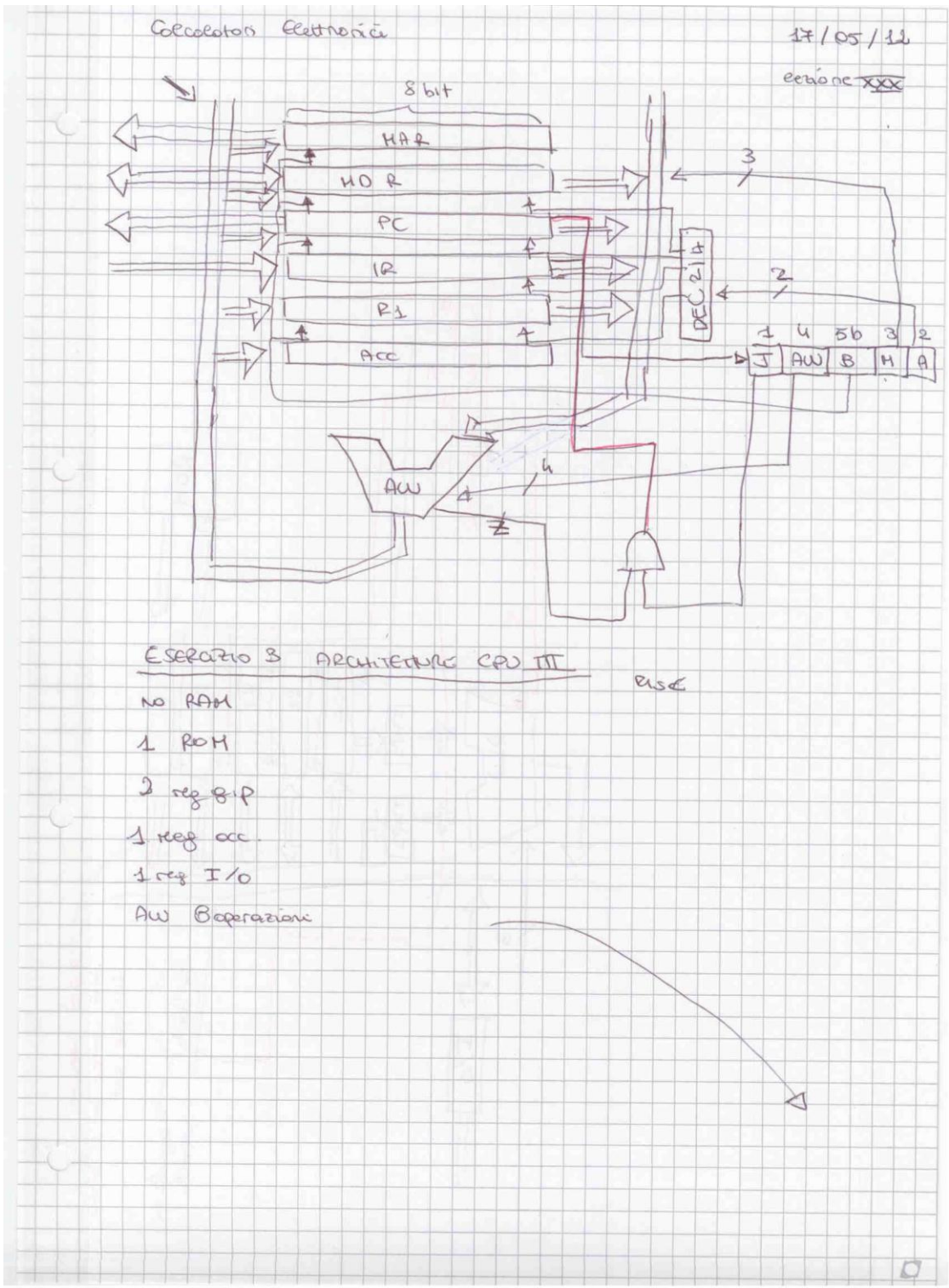


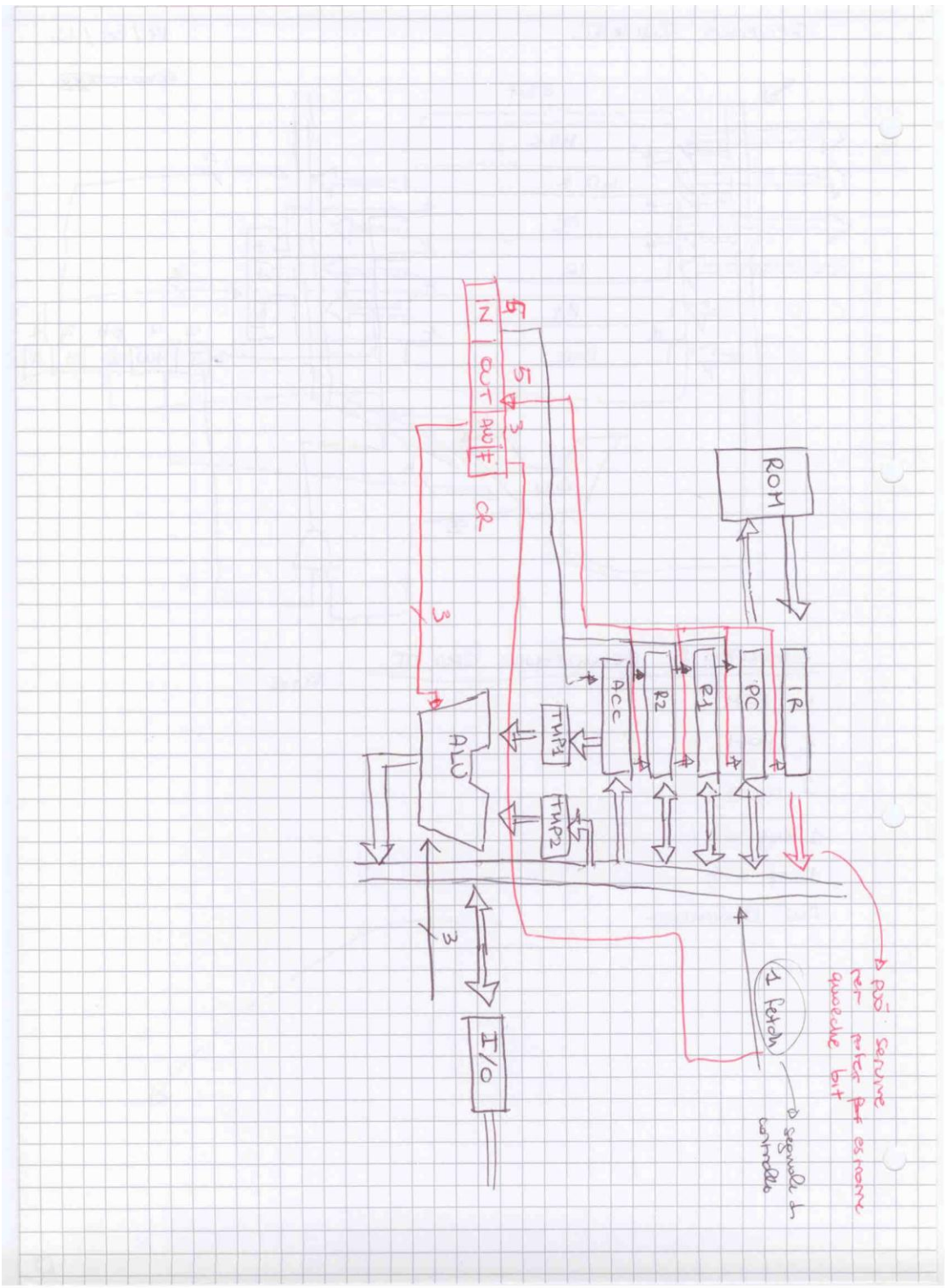
CON QUESTO SCHEMA NON POTREI IMPLEMENTARE I SALTI CONDIZIONALI

prevedo anche i salti condizionali

ESERCIZIO SULLE ARCHITETTURE DI CPU II

- archit. risc 8 bit
- 1 reg g.p
- 2 reg fetch instr
- 2 reg dati
- ALU con 16 op.





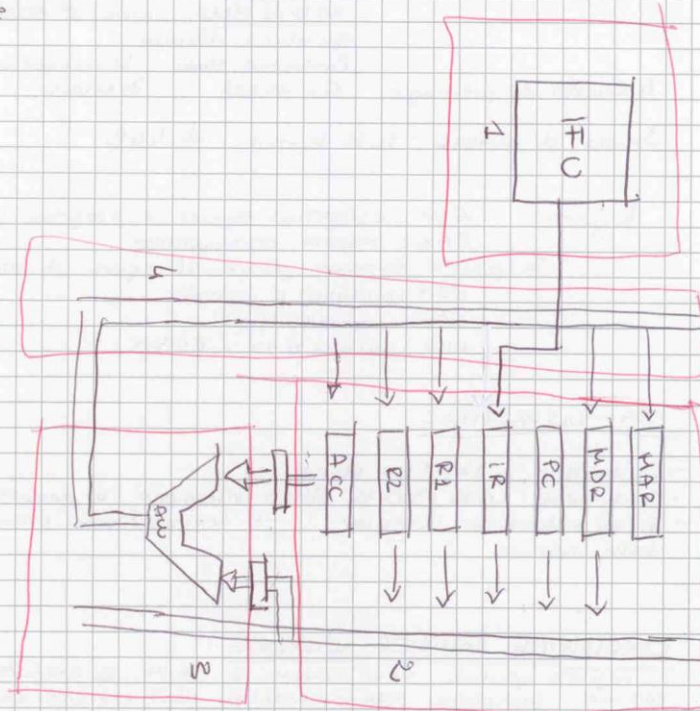
Calcolatori Elettronici

17/05/12

ESERCIZIO 4 di CPU IV

Lezione XXX

- CPU csc
- 3 reg. g.p
- 1 accumulatore
- 1 IFU



LIVELLO DELLE ISTRUZIONI MACCHINA

18/05/12

Lezione XXXI

INSTRUCTION SET ARCHITECTURE (ISA) e' interfaccia tra HW e SW

↳ e' il livello più basso a cui il processore è "programmabile"

LINGUAGGIO ASSEMBLATO a simboli
' ' BINARIO a bit
(MACCHINA)

REGISTRI UFFICIALI

- registri general-purpose
- special purpose purpose
- registri visibili solo in kernel mode
- PSW (Program Status Word) reg. che contiene una serie di flag relativi

invocazione di funzione: per ciascuna invocazione viene allocata in una porzione della memoria organizzata a pila (stack), una nuova area detta STACK FRAME

ISA - PENTIUM 4 : Architettura a 32 bit IA-32

modi di funzionamento: Real Mode: si comporta come 8088
Virtual Mode: come e' 8086 ma intercetta tutte le operazioni delicate
Protected Mode: si comporta come un P4
4 livelli di privilegio 0 = kernel 3 = Users

Spazio di memoria: 16K segmenti di 4GB

Registri: EIP: instruction pointer = program counter
EAX: registro accumulatore
si possono chiamare porzioni di registri diversi
EBX: puntatori a memoria
ECX: controlli cicli
EDX: estende EAX a 64 bit

ISA - ULTRASPARC

- Architettura a 32 bit o a 64 bit
- Architettura LOAD/STORE fanno riferimento a memoria
- Spende abbondanza di registri → organizzazione pensata per alleggerire il carico delle chiamate

OVERLAPPING REGISTER WINDOWS

i registri globali sono comuni a tutte le strutture
con una semplice denominazione dei registri posso gestire ^{una} l'elaborazione

ISA - 8051:

- 8 registri ufficiali da 8 bit

FORMAT DELLE ISTRUZIONI:

OP CODE

(a)

OP CODE | ADDRESS

(b)

OP CODE | ADDR1 | ADDR2

(c)

OP CODE | ADDR1 | ADDR2 | ADDR3

(d)

Calcolatori Elettronici

21/05/12

ESANSIONE dei CODICI OPERATIVI

lezione XXXII

- 1) La sezione di codice operativo non deve avere intersezioni
 - ↳ prima vede i 4 bit e se che i successivi saranno gli operandi

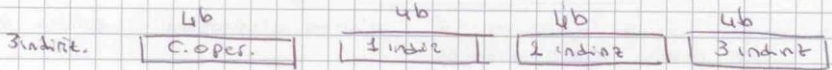
ESER C270

16 bit

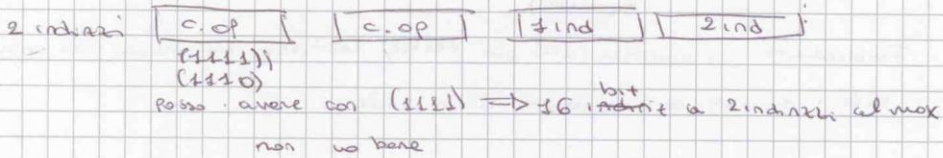
4 bit per indirizzare 4 bit

- 18 istruzioni a 2 indirizzi (2 operandi)
- 8 a 1 indirizzo
- 0 a 0 indirizzi

- 1) quante istr. a 3 è possibile avere al MAX?
- 2) schema istr.
- 3) 8 → 80



insero una codifica (1111) per gruppi successivi
 (1111: 15 istruzioni a 3 indirizzi)

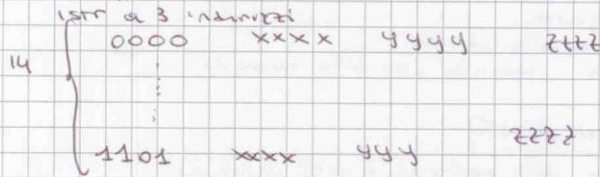


provo con 2 codifica (1111) ⇒ 32 bit indirizzi a 2 indirizzi al max
 (1110)

$32 - 18 = 14$ me ne overtolto

14 * 16 istruzioni a 1 indiriz

Schema di OGNI indirizzi



Istr a 2 indirizzi



2 $\left\{ \begin{array}{cccc} 1111 & 0000 & 4444 & 2222 \\ 1111 & 0001 & 4444 & 2222 \end{array} \right.$

Istruz a 1 indirizzo

8 istr $\left\{ \begin{array}{ccc|c} 1111 & 1111 & 0000 & 2222 \\ 1111 & 1111 & 0111 & 2222 \end{array} \right.$

3) istruzioni a 1 indirizzo max 14 * 16

FORMATO DELLE ISTR. DEL PENTIUM IV

prefisso \rightarrow per codice qualità e lunghezza istruzione (poiché le istr. possono essere variabili)

Displacement (offset) \rightarrow organizza la memoria a segmenti in cui posso specificare un indirizzo (indirizzo all'interno del segmento)

Immediate: e' operando sta più dentro nell'istr. immediata \Rightarrow costante

SIB: prevede la disponibilità di un indice

FORMATO DELLE ISTR. SPARC

di memoria standard \rightarrow 32 bit

gli operandi sono fissi a 5 bit e sono 3

FORMATO ISTR. 8051

- istr a 1, 2 o 3 byte
- Operandi 3 bit: registro
8 bit: costante o indirizzo
11 o 16 bit: indirizzi altri della memoria

MODI DI INDIRIZZAMENTO

- Immediato: il valore dell'operando è nell'istruzione
- Indiretto: l'istruzione contiene l'indirizzo di memoria completo dell'operando
- Indiretto a registro: l'indirizzo di memoria fornito contiene l'indirizzo dell'operando
- A registro: si specifica un registro da contenere l'operando
- Indirizzo a registro: il reg. specificato contiene l'indirizzo dell'operando
- Indirizzato: l'indirizzo è dato da una costante più il contenuto di un registro
- A registro base: viene sommato a tutti gli indirizzi il contenuto di un registro
- A stack: l'operando è sulla cima dello stack (o a discesa)

Calcolatori Elettronici

21/05/12
Lezione XXXII

→ INDIRIZZAMENTI VARI

MOV	R1	4
-----	----	---

MOV R1, #0 ← indica che il valore è costante indirizzamento immediato

MOV R2, #A - 100h

MOV R3, #A + 100h

LOOP: ADD R1, (R2)

ADD R2, #4

confronto tra R2/R3 → CMP R2, R3

BLT LOOP : // salto condizionale

salto l'istruzione quando è negativo

R less than

22/05/12
Lezione XXXIII

Indirizzamento a STACK (o PILA): è utilizzato per

- Gestire le chiamate di procedura
- Calcolare espressioni aritmetiche
- Opposizione risultati intermedi

↳ stack pointer SP punta all'elemento offuscato dello STACK

OPERAZIONI FONDAMENTALI:

PUSH: aggiunge un elemento alla cima dello stack

POP: preleva un elemento dalla cima dello stack

- operazioni aritmetiche

↳ NOTAZIONE POLACCA INVERSA:

Per calcolare un'espressione mette l'operatore alla fine

Es: A A + (e non (A+A)) ⇒ A+A

Permette di non mettere le parentesi → no. ambiguità

visita e' opera in POST-ORDINE prima albero sinistro, poi destro

COERENZA: un set di istruzioni è coerente:

- codice operativo
- Modalità di indirizzamento

Calcolatori Elettronici

28/05/12

AX: registro accumulatore

lezione XXXIV

BX: registro base, usato come accumulatore o come puntatore alla memoria

CX: registro contatore per contare i cicli

DX: registro dati, usato insieme ad AX per contenere parole a 32 bit

REGISTRI PUNTATORI

SP: puntatore alla cima della stack modificato con le operazioni PUSH e POP

BP: puntatore alla base della stack, assegnato alla procedura corrente

segmento: pezzo della memoria → istruzioni continue

OFFSET → spostamento

ADC somma ai due dati viene anche sommato il valore del bit di carry

29/05/12

ESEMPIO

lezione XXXIV

! calcolo di $(a+3) * b$

EXIT = 1

SECT.TEXT

start:

MOV AX, (a)

ADD AX, 3

MUL (b)

PUSH 0

PUSH EXIT

SVS

SECT.DATA

a: WORD 5

b: WORD 3

0508 I

t38 I

la memoria è contenuta
della memoria

! calcolo moltiplicazione: ripetute per 2

```
    -EXIT = 1
.sect .EXT
start:
    MOV AX, 258
    ADD B AH, AL
    MOV CX, (times)
    MOV BX, mword
    MOV AX, (BX)
    1:  MUL 2 (BX)
        LOOP 1b
        PUSH 0
        PUSH -EXIT
    sys
.sect .DATA
times: .WORD 5
mword: .WORD 1, 2
```

Somma di Byte

speto all'indirizzo b → back
in avanti f → forward

MovE SP, BP

31/05/12

4/05/12

metto nella pila in ordine inverso a come sono
messi nella pila

Q CALL inserisce in automatico l'indirizzo di ritorno
nella pila

! POP che lo toglie direttamente dalla pila ma lo lascia
nel registro

ESERCIZIO IV (ASSEMBLER)

SI → registro indice
BX → registro base

! le allocazioni della memoria sono sequenziali

8/05/12

JUMP PC → rimpicciolare una nuova istruzione
prende l'istruzione direttamente dal
program counter

Collocazioni Elettronici

05/06/12

ESERCITAZIONE

Lezione XXXV

ESERCIZIO VII

```
vect1 .word 1, 3, 5, 7
vect2 .word 2, 3, 5, 0
len .word 0
MOV end, vect1 - vect2
CMP end, vect1
JNE 1f
MOV cx, end ;
LOOP .
MOV si, 0
MOV bx, (vect)(si)
```

vect1 - vect2 → mi dice
la lunghezza
di 4

ESERCIZIO VIII

1. Fattorizzare

```
n .word 5
MOV AX, (n)
MOV CX, n+1
1 MOV CX, n
  LOOP 1b
  PRINTF - AX
```

Ⓛ con il LOOP bisogna mettere sempre
il salto all'indietro

Sorgente : <http://holisi.hasanaj.com/>
Autore : Holsi Hasanaj
Professore: Riccardo Torlone

Corso di Ingegneria Informatica Roma Tre
Anno di produzione: 2011-2012

