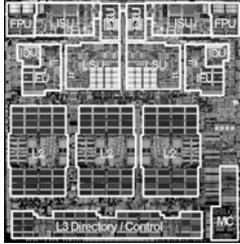




Università degli Studi
di Cassino



**Corso di
Calcolatori Elettronici II**

Introduzione

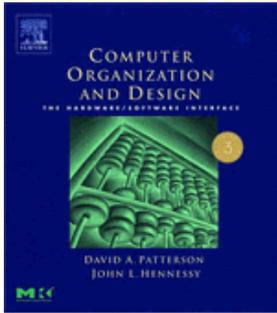
Anno Accademico 2004/2005
Francesco Tortorella

Contenuti del corso

- Architettura del processore
- Sistema di bus
- Gerarchia di memoria
- Sistema di I/O

Testo adottato:

Patterson, Hennessy
Computer Organization and Design
The hardware/software interface
3° edition
Morgan Kaufmann 2005



Calcolatori Elettronici II
Introduzione - 2

Versione italiana (?):

Patterson, Hennessy
Struttura, organizzazione e
progetto dei calcolatori
2a edizione
Jackson Libri

F. Tortorella © 2005
Università degli Studi
di Cassino

...chi siamo, da dove veniamo ?

- Fondamenti di Informatica I:
 - Strutture dati
 - Costrutti di programmazione
 - Algoritmi fondamentali
- Fondamenti di Informatica II:
 - Strutture dati complesse
 - Ricorsione
- Calcolatori Elettronici I:
 - Modello di programmazione del processore
 - Programmazione in linguaggio assembly
 - Rappresentazione dei dati
 - Elementi di progettazione logica dei circuiti

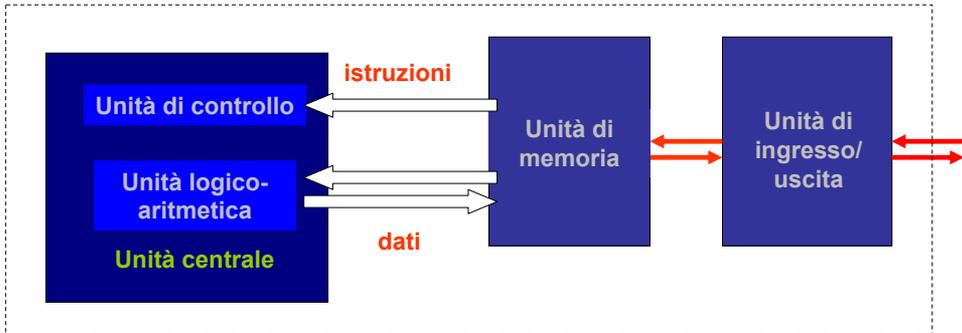
C / C++

Assembly

Calcolatori Elettronici II
Introduzione - 3

F. Tortorella © 2005
Università degli Studi
di Cassino

Organizzazione del calcolatore

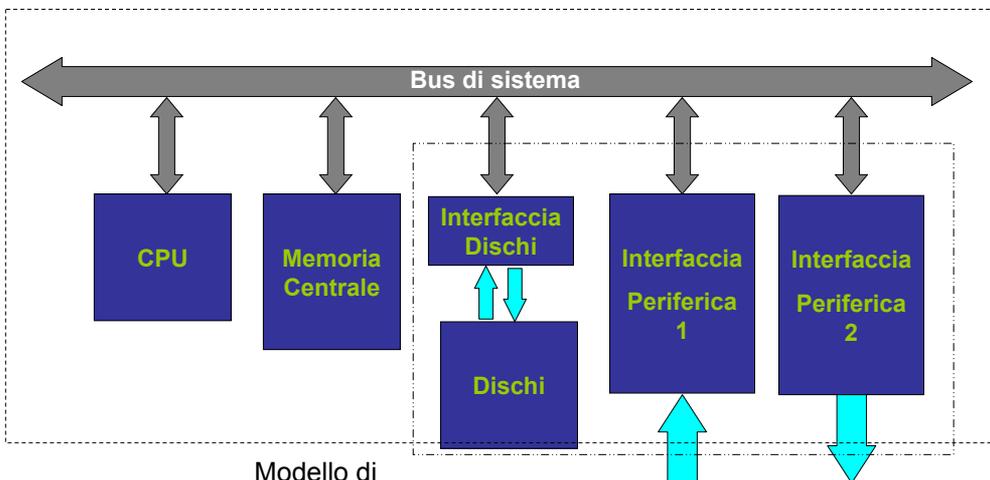


Modello logico

single componenti

flussi di dati e istruzioni

Modello di von Neumann



Modello di
implementazione

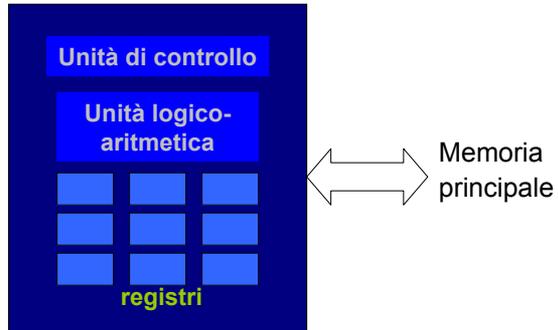
CPU (Central Processing Unit)

Funzione:

eseguire i programmi immagazzinati in memoria principale prelevando le istruzioni (e i dati relativi), interpretandole ed eseguendole una dopo l'altra

E' formata da:

- unità di controllo
- unità logico aritmetica
- registri



La CPU è inoltre caratterizzata dall'insieme delle istruzioni che può eseguire (instruction set)

Calcolatori Elettronici II
Introduzione - 6

F. Tortorella © 2005
Università degli Studi
di Cassino

Architettura del processore MIPS

Microprocessor without Interlocking Pipe Stages

- Architettura Load/Store con istruzioni aritmetiche registro-registro a 3 operandi
- Istruzioni di 32-bit - 3 Formati (R, I, J)
- 32 registri generali di 32 bit (R0 contiene 0, R31 riceve l'indirizzo di ritorno) (+ HI, LO)
- Modi d'indirizzamento: Register, Immediate, Base+Offset, PC-relative
- Immediati a 16-bit + istruzione LUI

Calcolatori Elettronici II
Introduzione - 7

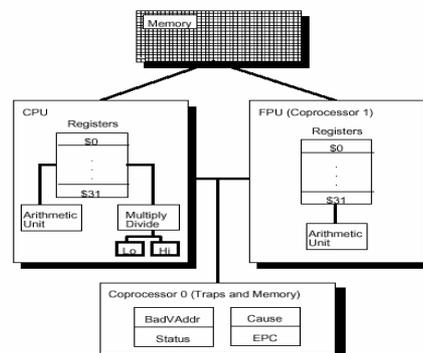
F. Tortorella © 2005
Università degli Studi
di Cassino

Architettura del processore MIPS

- Supporto per interi in complemento a 2 di 8 (byte), 16 (halfword) e 32 (word) bit e, con coprocessore opzionale, per numeri floating-point IEEE 754 singola e doppia precisione
- Branch semplici senza codici di condizione
- *Delayed branch* (l'istruzione dopo il salto viene comunque eseguita) e *Delayed load* (l'istruzione dopo una load non deve usare il registro caricato), senza interlock

Coprocessori

- Può supportare fino a 4 coprocessori, numerati da 0 a 3
- Il coprocessore di controllo del sistema (coprocessore 0) è integrato nel chip e gestisce la memoria e le eccezioni
- Il coprocessore floating-point (coprocessore 1) opzionale ha 32 registri di 32-bit (\$f0 - \$f31), di cui sono utilizzabili quelli di posto pari in semplice o doppia precisione



Registri del MIPS e convenzione di uso

32 registri generali da 32 bit



registri speciali

PC

HI
LO

Register name	Number	Usage
\$zero	0	constant 0
\$at	1	reserved for assembler
\$v0	2	expression evaluation and results of a function
\$v1	3	expression evaluation and results of a function
\$a0	4	argument 1
\$a1	5	argument 2
\$a2	6	argument 3
\$a3	7	argument 4
\$t0	8	temporary (not preserved across call)
\$t1	9	temporary (not preserved across call)
\$t2	10	temporary (not preserved across call)
\$t3	11	temporary (not preserved across call)
\$t4	12	temporary (not preserved across call)
\$t5	13	temporary (not preserved across call)
\$t6	14	temporary (not preserved across call)
\$t7	15	temporary (not preserved across call)
\$s0	16	saved temporary (preserved across call)
\$s1	17	saved temporary (preserved across call)
\$s2	18	saved temporary (preserved across call)
\$s3	19	saved temporary (preserved across call)
\$s4	20	saved temporary (preserved across call)
\$s5	21	saved temporary (preserved across call)
\$s6	22	saved temporary (preserved across call)
\$s7	23	saved temporary (preserved across call)
\$t8	24	temporary (not preserved across call)
\$t9	25	temporary (not preserved across call)
\$k0	26	reserved for OS kernel
\$k1	27	reserved for OS kernel
\$gp	28	pointer to global area
\$sp	29	stack pointer
\$fp	30	frame pointer
\$ra	31	return address (used by function call)

Gestione degli indirizzi di memoria

- Spazio di indirizzi di 2^{32} byte (4 Gigabyte, con i 2 superiori riservati al S.O.), ossia 2^{30} word
- L'indirizzamento è al byte (incremento di 4 per passare da una word alla successiva)
- L'indirizzo di una word è quello del suo primo byte (byte di indirizzo minore)
- Negli accessi, l'indirizzo di un dato di s byte deve essere allineato, ossia $A \bmod s = 0$ (esistono istruzioni per accedere a dati disallineati)
- L'ordinamento dei byte in una word può essere sia *big-endian* (il primo byte è quello più significativo) che *little-endian* (il primo byte è quello meno significativo), in dipendenza del valore logico su di un pin

L'Unità di controllo

E' l'unità che si occupa di dirigere e coordinare le attività interne alla CPU che portano all'esecuzione di una istruzione

Ciclo del processore

L'esecuzione di una istruzione avviene attraverso alcune fasi:

Fetch

L'istruzione da eseguire viene prelevata dalla memoria e trasferita all'interno della CPU

Decode

L'istruzione viene interpretata e vengono avviate le azioni interne necessarie per la sua esecuzione

Operand Assembly

Vengono prelevati dalla memoria i dati su cui eseguire l'operazione prevista dalla istruzione

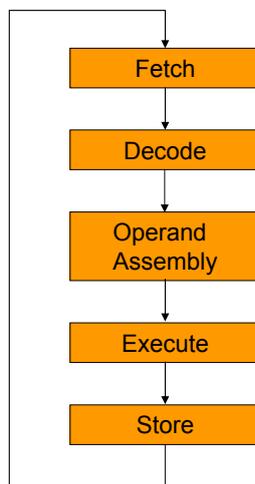
Execute

Viene portata a termine l'esecuzione dell'operazione prevista dalla istruzione

Store

Viene memorizzato il risultato dell'operazione prevista dalla istruzione

L'Unità di controllo



L'unità di controllo realizza in ciclo le fasi per eseguire la sequenza di istruzioni che costituiscono il programma

L'Unità Logico Aritmetica

E' l'unità che si occupa di realizzare le operazioni logiche ed aritmetiche eventualmente richieste per eseguire un'istruzione

Operazioni Aritmetiche

ADD

SUB

MUL

DIV

REM

SET

Operazioni Logiche

CMP

AND

OR

NOT