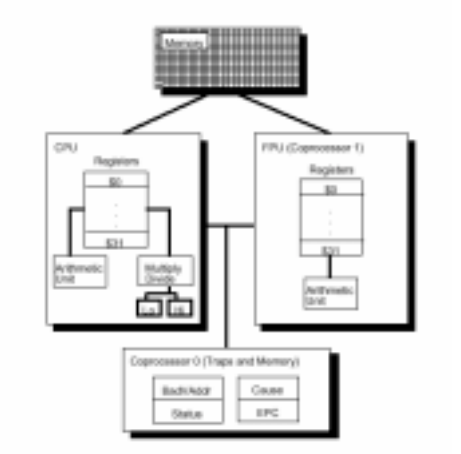
Architettura del processore MIPS
Microprocessor without Interlocking Pipe Stages

- Architettura Load/Store con istruzioni aritmetiche registro-registro a 3 indirizzi
- 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

- Supporto per interi in complemento a 2 di 8 (byte), 16 (halfword) e 32 (word) bit e, con coprocesore 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

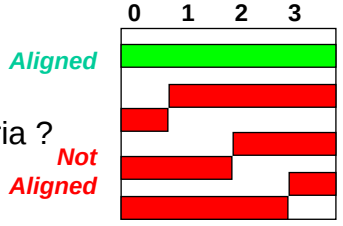
PC

HI
LO

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 mod 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

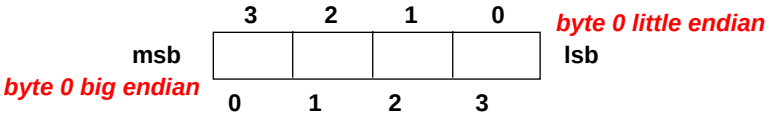
Scelte implementative nella gestione degli indirizzi



➔ può una word essere memorizzata in qualunque indirizzo della memoria ?

➔ come si distribuiscono gli indirizzi dei byte appartenenti alla word ?

Big Endian (IBM 360/370, Motorola 68k, Sparc, HP PA)
Little Endian (Intel 80x86, DEC Vax, DEC Alpha)



```
int i = 0x12345678;

struct s_point{
       short x;
       short y;
} p =
{0x1234,0x5678};

char str[]="ABC";
```