Il set di istruzioni a 32 bit

Il set di istruzioni dei processori Intel x86 a 32 bit (IA-32) è un'evoluzione del set di istruzioni dell'8086.

Si possono distinguere due ambiti principali di programmazione:

- Programmazione applicativa
- Programmazione di sistema

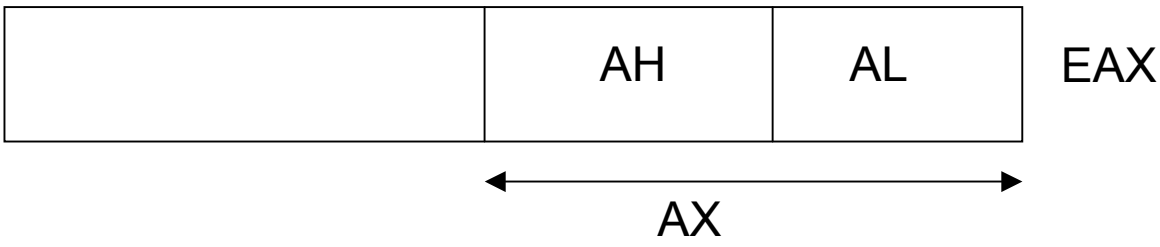
Per gli scopi del corso il primo settore è adeguato, e il sottoinsieme di istruzioni che lo riguarda è molto simile a quello dell'8086.

I registri generali

I registri generali sono un'estensione a 32 bit di quelli dell'8086:

EAX EBX ECX EDX EBP ESP ESI EDI

I 16 bit più bassi si riferiscono con gli stessi nomi di queli dell'8086, e allo stesso modo il primo e secondo byte:



Formato delle istruzioni

Prefissi	**Opcode**	ModR/M	SIB	Spiazzamento	Immediato

- Prefissi (1-4 byte): ai prefissi dell'8086 (ripetizione – REP - e override di segmento) sono aggiunti l'override di taglia di dato e di indirizzo: ciò consente che a livello di singola istruzione si possano usare operandi di taglia diversa da quella di default del segmento (16 o 32 bit)
- Opcode (1-2 byte): il codice operativo
- ModR/M (1 byte): specifica gli operandi registro e la modalità di indirizzamento (registro – memoria)
- SIB (1 byte): specifica il registro indice, base e il fattore di scala (2, 4, 8) (scale, index, base)
- Spiazzamento (0-4 byte): componente di indirizzo
- Immediato (0-4 byte): operando costante

Modalità di indirizzamento

- Simili a quelle dell'8086, ma con maggiore flessibilità:

base + indice * scala + costante

decido la taglia dei salti di indirizzo nelle indicizzazioni (buono per vettori e matrici)

base: uno qualsiasi tra i registri generali

indice: uno qualsiasi tra i registri generali, tranne ESP

costante: fino a 32 bit

- Come per l'8086, se ESP o EBP sono usati come base, il registro di segmento di default è CS; altrimenti, è DS

Esempi di istruzioni

MOV destination, source

- Operandi a 8, 16, 32 bit
- Un operando in memoria al più
- Registri di segmento con altri registri o memoria, ma non costanti

Come la MOV, si estendono intuitivamente a 32 bit istruzioni quali ADD, SUB, AND, OR, etc…

MUL source

- Operandi a 8, 16, 32 bit
- Nel caso di operandi a 32 bit:

  EDX:EAX = EAX * source
  Quindi risultato in 64 bit (16+16).

J*cc* relative_target

- Come per l'8086, salta se la condizione *cc* è verificata a un indirizzo che è ottenuto sommando relative_target a EIP (puntatore all'istruzione).
- Differenza: relative_target non è limitato a 8 bit, ma può essere di 16 o 32.

PUSH source

- source può essere una qualsiasi quantità (anche una costante, a differenza dell'8086) di 1, 2, o 4 byte.

SHL operand, count

- operand può essere un registro o una locazione di memoria di 1, 2, o 4 byte.
- count non è più limitato a 1 o a CL, ma può essere una costante di 8 bit.

IN AL/AX/EAX, DX e OUT DX, AL/AX/EAX

- legge o scrive una quantità di 1, 2, 4 byte dall'indirizzo di I/O specificato in DX