

Architettura x86 – 32

	IA-16 (x86-16)	IA-32 (x86-32)	X64 (x86-64)
Indirizzamenti	16	16,32	16,32,64
Registri	8,16	8,16,32	8,16,32,64
Natività	8086 e 286	386,486,Pentium	Pentium

L'IA-64 non è presente nella precedente tabella in quanto è una architettura Intel completamente differente, chiamata Itanium, basata su tecnologia VLIW (Very Long Instruction Word). Le VLIW sono macroistruzioni già ottimizzate per essere eseguite con il massimo parallelismo.

L'architettura IA-32 ha 3 diverse modalità di funzionamento. Ogni modalità determina quali istruzioni e features del processore sono utilizzabili.

Esse sono:

- *Real Mode (Real Address Mode)*

E' attiva al momento del Power-On o Reset. In questa modalità si ha uno spazio di allocazione di memoria a 20bit, accesso diretto alle routine del BIOS.

La quantità massima di memoria indirizzabile è 1MB (220 Byte).

Non è presente (o attivo) alcun sistema di protezione memoria e multitasking a livello hardware.

- *Protected Mode (Protected Virtual Address Mode)*

E' lo stato nativo del processore. Modo base a 32bit (Windows e Linux).

La quantità massima di memoria indirizzabile è 4GB (232 Byte).

In questa modalità è possibile utilizzare memoria virtuale, paging, multitasking (safe).

Si può passare in questa modalità modificando alcuni registri di controllo ed abilitando opportuni flag.

- *VM86 (Virtual 8086 Mode)*

La VM86 (Virtual Mode) permette l'esecuzione di software Real Mode che non possono essere eseguite all'interno di ambienti in Protected Mode (ad esempio, applicazioni DOS su O.S. recenti).

Essa usa una segmentazione identica alla Real Mode.

Vi è comunque il meccanismo di paginazione, ma è trasparente al programmatore. E' attiva la protezione della memoria e l'isolamento dello spazio indirizzi.

Nei processori attuali è in realtà una sotto-modalità della Protected Mode.

Esempio:

Richiesta di Accesso a disco

→ Viene richiamato l'O.S (Protected Mode)

→ Viene coinvolto il Filesystem (Protected Mode)

→ Viene richiamato il Driver Logico (BIOS) (Protected Mode)

→ INT xx (Real Mode, non rientrante)

Solitamente, in questa fase, l'O.S. bypassa il BIOS.

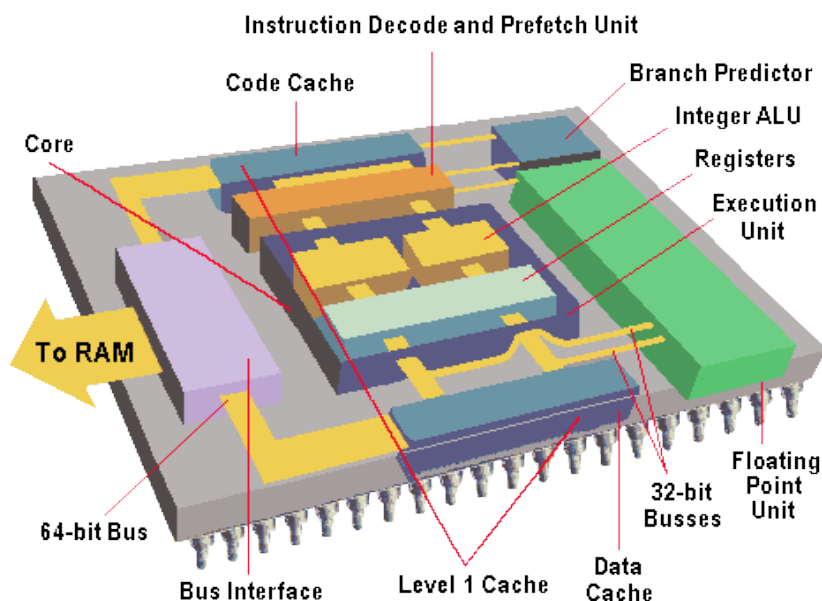
Registri e Modes

A seconda della modalità in cui si opera, si ha una diversa visibilità di registri e modello di memoria.

VM86			Real Mode			Protected Mode		
Registro	Descrizione	Bit	Registro	Descrizione	Bit	Registro	Descrizione	Bit
IP	Instruction Pointer	16	EIP	Instruction Pointer	32	EIP	Instruction Pointer	32
SI	Source Index	16	ESI	Source Index	32	ESI	Source Index	32
DI	Destination Index	16	EDI	Destination Index	32	EDI	Destination Index	32
FLAG	Status Flags	16	EFLAG	Status Flags	32	EFLAG	Status Flags	32
AX	Accumulator	16	EAX	Accumulator	32	EAX	Accumulator	32
BX	Base	16	EBX	Base	32	EBX	Base	32
CX	Counter	16	ECX	Counter	32	ECX	Counter	32
DX	Data	16	EDX	Data	32	EDX	Data	32
SP	Stack Pointer	16	ESP	Stack Pointer	32	ESP	Stack Pointer	32
BP	Base Pointer	16	EBP	Base Pointer	32	EBP	Base Pointer	32
CS	Code Segment	16	ECS	Code Segment	16	ECS	Code Segment	16
SS	Stack Segment	16	SS	Stack Segment	16	SS	Stack Segment	16
DS	Data Segment	16	DS	Data Segment	16	DS	Data Segment	16
ES	Extra Segment	16	ES	Extra Segment	16	ES	Extra Segment	16
			FS	Extra Segment	16	FS	Extra Segment	16
			GS	Extra Segment	16	GS	Extra Segment	16
						TR	Task	16 32 16
						LDTR	Local Desc. Table	16 32 16
						IDTR	Interrupt Desc. Table	32 16
						GDTR	Global Desc. Table	32 16
						CR3	Control	32
						CR2	Control	32
						CR1	Control	32
						CR0	Control	32
			TR7	Test	32	TR7	Test	32
			TR6	Test	32	TR6	Test	32
						DR7	Debug	32
						DR6	Debug	32
						DR5	Debug	32
						DR4	Debug	32
						DR3	Debug	32
						DR2	Debug	32
						DR1	Debug	32
						DR0	Debug	32

Componenti architettura 80x86

- Core
- Bus interface
- Cache L1
(Data cache & Code cache)
- Instruction Decode and Prefetch Unit
- Alu
- Registri
- Execution Unit
- Floating point unit
- Branch predictor



Registri, segmenti e principali funzioni

I registri possono essere classificati a seconda del loro utilizzo.

Data Registers

Sono utilizzati per memorizzare operandi ed i risultati delle operazioni.

Registro	Descrizione	Utilizzato anche	Note per l'Assembler
EAX / AX / AH\AL	Accumulator Register		Moltiplicazione, divisione, I/O, Shift Veloce
EBX / BX / BH\BL	Base Register	Per calcolo indirizzi	
ECX / CX / CH\CL	Count Register	Contatore	Cicli, Rotazioni, Dhift
EDX / DX / DH\DL	Data Register	Operazioni I/O come puntatore	Moltiplicazione, Divisione, I/O

Pointer Registers

Sono utilizzati come puntatori.

Registro	Descrizione	Utilizzato anche	Note per l'Assembler
EIP / IP	Instruction pointer	Punta alla successiva istruzione da eseguire	Non modificabile dall'utente
ESI / SI	Source index	Registro indice per la memoria	Da trattare come operando di tipo <i>mem</i>
EDI / DI	Destination index	Registro indice per la memoria	Da trattare come operando di tipo <i>mem</i>

Stack Registers

Sono utilizzati per l'implementazione dello stack.

Registro	Descrizione	Utilizzato anche	Note per l'Assembler
ESP / SP	Stack Pointer	Punta alla testa dello stack	L'indirizzo viene decrementato ad ogni <i>push</i> e incrementato ad ogni <i>pop</i>
EBP / BP	Base Pointer	Base address dello stack	

Test Registers

Registri utilizzati nella fase di Self-Test

Registro	Descrizione
TR7	Test Data
TR6	Test Command

Segment Registers

Registro	Registri offset validi	Descrizione	Funzione	Note per l'Assembler
DS,ES	EBX / ESI / EDI / BX / SI / DI	Data segment	Indirizzo base del segmento dati	
CS	EIP / IP	Code segment	Indirizzo base del segmento codice	Per modificare tale registro è necessaria una chiamata a procedura far, oppure una far jump oppure un interrupt. In Protected Mode viene verificato se il nuovo segmento può essere utilizzato dal task
SS	ESP / EBP / SP / BP	Stack segment	Indirizzo del segmento stack	

Process Status Word (PSW)

È composta da 16 bit, ma solo 9 di questi sono usati. Ogni bit corrisponde ad un flag.

I flag si dividono in:

- flag di condizione
- flag di controllo.

Flag di condizione

Vengono automaticamente scritti al termine di varie operazioni:

- SF (Sign Flag): coincide con il MSB del risultato dopo una operazione aritmetica
- ZF (Zero Flag): vale 1 se il risultato è nullo, 0 altrimenti
- PF (Parity Flag): vale 1 se il numero di 1 negli 8 bit meno significativi del risultato è pari, 0 altrimenti
- CF (Carry Flag): dopo le operazioni aritmetiche vale 1 se c'è stato riporto (somma) o prestito (sottrazione); altre istruzioni ne fanno un uso particolare
- AF (Auxiliary Carry Flag): usato nell'aritmetica BCD; vale 1 se c'è stato riporto (somma) o prestito (sottrazione) dal bit 3
- OF (Overflow Flag): vale 1 se l'ultima istruzione ha prodotto un overflow.

Flag di controllo

Possono venire scritti e manipolati da apposite istruzioni e servono a regolare il funzionamento di talune funzioni del processore:

- DF (Direction Flag): utilizzato dalle istruzioni per la manipolazione delle stringhe; se vale 0 le stringhe vengono manipolate partendo dai caratteri all'indirizzo minore, se vale 1 a partire dall'indirizzo maggiore
- IF (Interrupt Flag): se vale 1, i segnali di Interrupt mascherabili vengono percepiti dalla CPU, altrimenti questi vengono ignorati
- TF (Trap Flag): se vale 1, viene eseguita una trap al termine di ogni istruzione.