

SISTEMI OPERATIVI

Processori intel












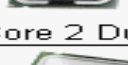
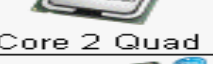
Prof. Viglietti Francesco

Classe 4 B info

A.S. 2010-11



Evoluzione processori intel

	N transistor	Frequenza	Processo produttivo [μm]	bit	N core
 4004	2.300	108 kHz	10,00	4	1
 8008	3.500	800 kHz	10,00	8	1
 8086	29.000	4.7 - 10 MHz	3,00	16	1
 386	134.000	16 - 40 MHz	1,50 - 1,00	32	1
 486	275.000	16 - 133 MHz	0,80	32	1
 Pentium	3.100.000	60 - 300 MHz	0,80 - 0,25	32	1
 Pentium II	7.500.000	233 - 450 MHz	0,35 - 0,25	32	1
 Pentium III	9.5 - 46 mil	450 - 1400 MHz	0,25 - 0,13	32	1
 Pentium IV	42 - 178 mil	1.3 - 3.8 GHz	0,18 - 0,065	32 - 64	1
 Pentium D	230 - 376 mil	2.66 - 3.6 GHz	0,09 - 0,065	64	2
 Core 2 Duo	167 - 410 mil	1.06 - 3.33 GHz	0,065 - 0,045	64	2
 Core 2 Quad	582 - 820 mil	2.4 - 3 GHz	0,065 - 0,045	64	4
 Core i7	731 mil	2.66 - 3.2 GHz	0,045	64	4

Caratteristiche IA-32 1

IA-32 → sono processori CISC

Pentium → pipelining

BTB → tabella che gestisce i salti nel codice, è una tabella che statisticamente (con buona probabilità) ne predice il salto ...

Pipeline superscalare → la fase di elaborazione di numeri fixed point viene eseguita mediante due ALU che possono lavorare in parallelo... i numeri floated point sono eseguiti da una terza ALU organizzata a pipelining con unità dedicate a somma, moltiplicazione e divisione.

Pentium pro, pentium II e pentium III → P6

Superscalare a tre vie con pipeline, esegue circa 3 istruzioni per ciclo di clock, usando una pipelining a 12 livelli, ed un'esecuzione dinamica → (branch prediction, dynamic data-flow analysis, speculative execution)

Caratteristiche IA-32 2

Pentium 4, Core Duo → Netburst

Usa una pipeline a 20 livelli ed un'esecuzione dinamica

Core 2 Duo → Core

Usa un'esecuzione dinamica ampia, ciascun core della CPU esegue fino a 4 istruzioni per ciclo di clock usando una pipeline a 14 livelli, con 3 ALU e 4 decoder, usa un branch prediction avanzato.

Hyper-Threading → più processi o parti (thread) possono essere eseguiti contemporaneamente. Al OS appare come un doppio processore “logico”.

Multi-core sulla stessa CPU ci sono 2 o più core, ogni core, può anche essere hyper-threading

Caratteristiche IA-32 3

MMX → (Multimedia extension) istruzioni introdotte dal pentium II in poi per migliorare la multimedialità. Si basano su operazioni SIMD.

SSE → (Streaming SIMD Enhancement) istruzioni introdotte dal pentium III in poi per il miglioramento della multimedialità.

Evoluzioni sono le SSE2, le SSE3 e le SSSE3

Cache

Il pentium usa una cache associativa a 2 vie con 128 gruppi. Ogni linea è costituita da 32byte → $(128 \times 2 \times 32 = 8\text{KB})$ 20bit di TAG, 7bit di SET e 5bit di OFFSET

Nell'architettura P6 la cache è associativa a 4 vie
 $(128 \times 4 \times 32 = 16\text{KB})$

Nell'architettura Netburst la cache è associativa a 8 vie
 $(128 \times 8 \times 64 = 64\text{KB})$

Modalità operative

Due modalità di funzionamento + una modalità per il risparmio energetico:

Reale

Protetta

SMM

La CPU si avvia in modalità reale e passa poi alla protetta. In modalità reale (8086), applicazioni possono avere max 1MB di memoria, i registri sono a 16 bit. In modalità protetta i registri hanno 32 bit, e max 4GB di memoria, si privilegia il multitasking.

Ad ogni programma viene associato un livello di privilegio differente (livello 0..3). Livello 0 è senza limitazione, il livello 3 è quello delle applicazioni.

I registri degli applicativi

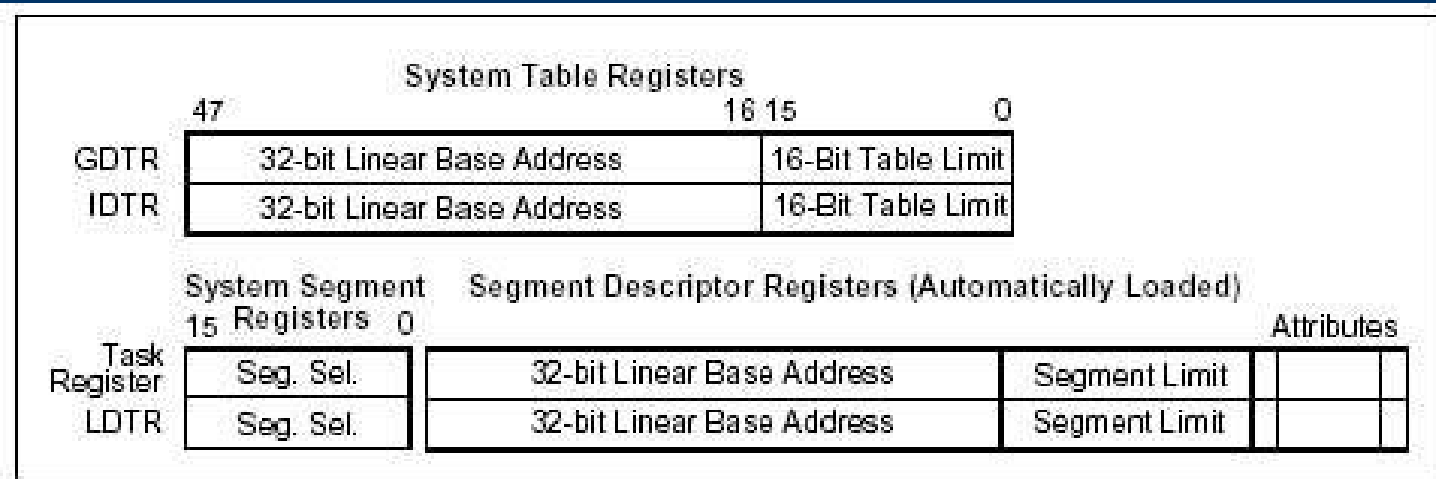
- 8 registri generali a 32bit: EAX (accumulatore) per operazioni aritmetiche, EBX (base) puntatore all'indirizzo base del segmento dati, ECX (contattore) indice di conteggio per i cicli, EDX (data) per operazioni aritmetiche, ESP (stack pointer), EBP (base pointer) indirizzo base del segmento dello stack, ESI (source index) indice per la stringa sorgente, EDI (destination index) indice per la stringa destinazione.
 - 6 registri selettori a 16bit: CS, DS, SS, ES, FS, GS individuano i segmenti dei dati (max 3), il segmento di codice e di stack.
 - Il registro a 32bit EFLAGS dove vengono i risultati notevoli come branch, carry, overflow...
 - Il registro program counter a 32 bit: EIP
 - 8 registri FPU a 80 bit per operazioni a virgola mobile
 - 8 registri MMX a 64 bit
 - 8 registri XMM a 128 bit
-
-

I registri di sistema

- 5 Registri di controllo a 32 bit:
 - CR0: contiene flag importanti bit 0 per l'abilitazione della modalità reale o protetta
 - CR1: riservato
 - CR2: usato per la gestione della memoria virtuale
 - CR3: contiene la tabella principale usata per la paginazione
 - CR4: controlla estensioni rispetto al 386
 - 4 registri a 32 bit per la gestione della memoria
 - Control Register MSR
 - 8 Debug Register
-
-

Registri gestione della memoria

- GDTR contiene il base address a 32 bit e il limite di 16 bit per la GDT (tabella globale usata da tutti i programmi). All'accensione il base address è settato al valore 0, il limite a FFFFh.
- LDTR contiene il selettore 16 bit, il base address a 32 bit, il limite del segmento di 16 bit e gli attributi del descrittore per la LDT (tabella usata per descrivere segmenti di memoria specifici).
- IDTR contiene il base address a 32 bit e il limite di 16 bit per la IDT (tabella usata per le routine di gestione degli interrupt).
- TR contiene un selector a 16bit, il base address di 32bit, il limite a 16bit e gli attributi del descrittore per il TSS (Task State Segment per il supporto del multitasking) del task corrente.



Gestione della memoria modalità reale

La memoria è organizzata in segmenti, i quali, possono essere suddivisi in pagine di memoria virtuale. I segmenti rappresentano sempre il punto di riferimento principale e vengono specificati attraverso l'aiuto di registri di segmento.

Nella modalità reale si possono indirizzare al massimo 1MB di memoria (20 bit).

I programmi sono divisi in segmenti fissi da 64KB.

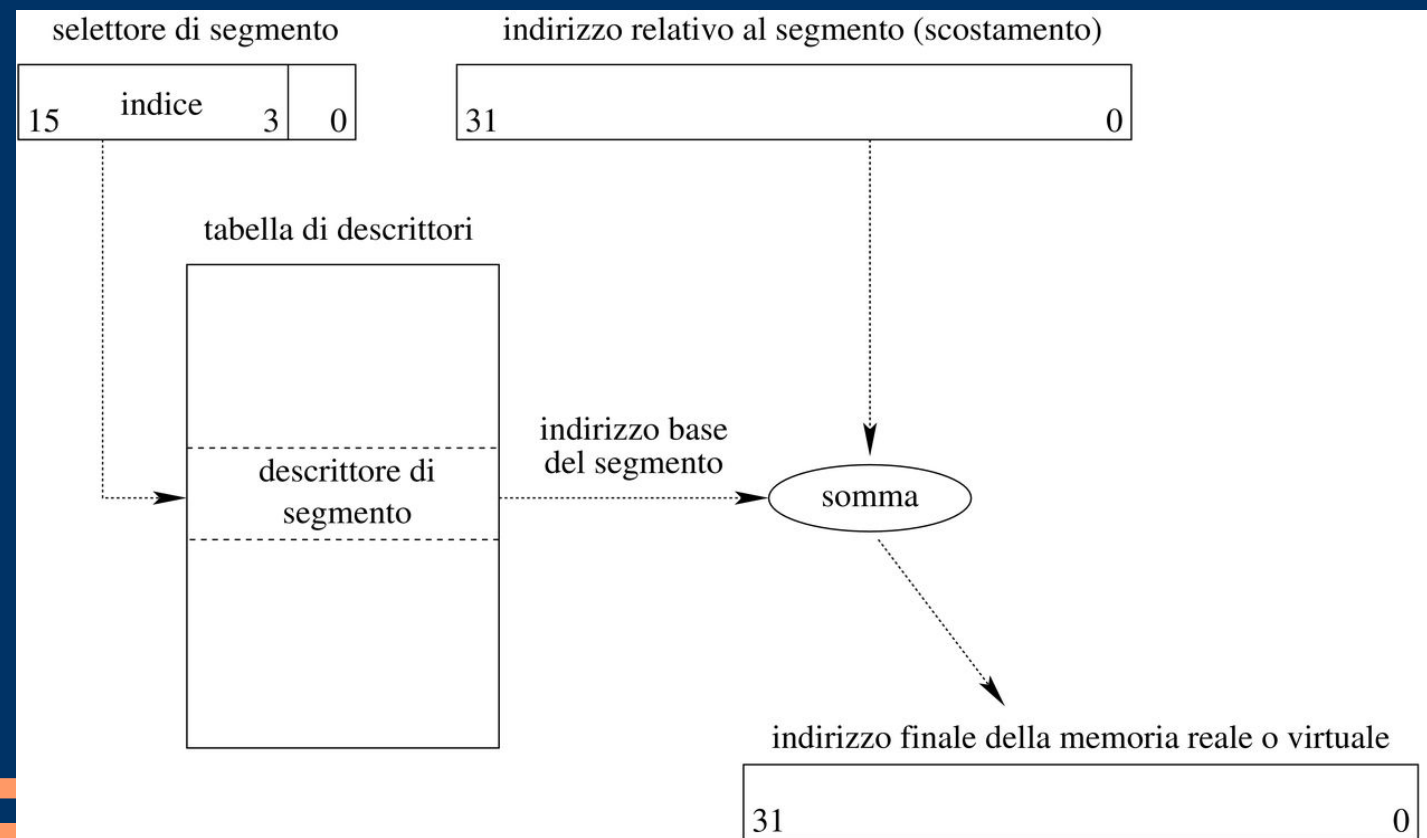
I registri di segmento (16 bit) individuano le parti di programma in memoria. Per individuare la locazione nel segmento serve l'offset, quindi l'indirizzo sarà nella forma:

Indirizzo di segmento:offset → *indirizzo fisico*

Es: 3A7F:120C → 3A7F0+120C=3B9FC

Gestione della memoria modalità protetta

Per individuare un indirizzo di memoria (reale o virtuale), si parte da un selettore, contenuto in un registro di segmento appropriato al contesto, dal quale si ottiene un indice per selezionare una riga dalla tabella dei descrittori. Viene individuato il descrittore di segmento, dal quale si ottiene l'indirizzo iniziale (indirizzo base del segmento). A questo indirizzo iniziale va aggiunto uno scostamento che rappresenta l'indirizzo relativo all'interno del segmento.



La tabella dei descrittori

La tabella dei descrittori contiene i descrittori ai segmenti:

- Indirizzo iniziale del segmento: da 16bit fino a 32bit (12,11,1)
- Dimensione del segmento: 16bit fino a 20bit (13,6)
- Tipo di segmento (dati/codice tipo di accesso): 4bit (10)
- Tipo di descrittore (sistema/applicazione): 1bit (9)
- Livello di privilegio : 2bit (8)
- Presenza in memoria: 1bit (7)
- Vari flag

1	2	3	4	5	6	7	8	9	10	11	12	13
---	---	---	---	---	---	---	---	---	----	----	----	----

Gestione degli interrupt 1

L'esecuzione del programma può essere interrotta tramite interrupt o exceptions.

Le interruzioni possono essere HW (esterne o asincrone) o SW (interne o sincrone).

Le eccezioni sono eventi sincroni generati durante l'esecuzione.

La CPU associa un numero ad ogni interrupt o exception in modo da eseguirne la giusta routine di gestione.

Nella modalità reale le interruzioni saranno suddivise in:

Predefinite → associate a particolari eventi, riconoscibili dalla CPU

HW → generate da dispositivi esterni

SW → richieste di programma

Quando arriva un interrupt la CPU interrompe l'esecuzione del programma corrente, salvando nello STACK i registri dei flag, il valore di IP e CS, disabilitando le interruzioni, individua la locazione della routine di gestione dalla IVT; esegue la routine dell'interrupt (ISR), poi riprende l'esecuzione del programma.

Gestione degli interrupt 2

In modalità protetta le eccezioni si suddividono in:

Fault → eccezioni che possono essere corrette

Trap → eccezioni che permettono l'esecuzione del programma dall'istruzione successiva a quella che ha generato l'istruzione.

Abort → eccezioni gravi, che interrompono l'esecuzione del programma

