

Lezione 17-09-08

Criteri per migliorare le prestazioni (vedi slide da 17 a 23)

- Sfruttare il parallelismo (mediante uso di pipeline e architetture superscalari)
- Disaccoppiare il fetch dall'esecuzione (mediante l'uso di una coda di prefetch o di una L1 Cache)

La fase di prelevamento istruzioni e la fase di esecuzione sono eseguite in tempi differenti tra i diversi processi; la coda di prefetch permette di sincronizzare due processi con lunghezza variabile (essenzialmente mediante un buffer). Non fa altro che precaricare l'istruzione successiva in una coda buffer mediante algoritmi di predizione molto complessi (nella famiglia Intel l'unità che si occupa della predizione è la Branch Target Buffer e ha ottimi risultati circa nel 90% dei casi).

L'altro modo per migliorare le prestazioni consiste nell'aumentare il parallelismo.

Per esempio, nelle macchine multicore si cerca di aumentare il parallelismo tra i thread. Macchine superscalari sfruttano invece il parallelismo a livello di istruzione. Quindi nel passaggio da sistemi a singolo core a sistemi multicore si nota un passaggio da ILP a TLP, cioè da instruction level parallelism a thread level parallelism.

A livello hardware, nei sistemi a multiprocessore serve una parte di controllo che permetta di coordinare i processi tra i singoli core. Il problema sta appunto nell'aggiungere questo componente che faccia da scheduler.

Architettura 80x86 (vedi slide 24-25)

Architettura composita: sulla stessa piattaforma posso coesistere architetture logiche diverse.

È un'architettura che utilizza istruzioni CISC cioè a lunghezza variabile (così si risparmia memoria, ma si aumenta la complessità progettuale).

Esistono tre tipi di architettura x86:

- x86-16 (IA-16): caratterizzata da indirizzamento su 16 bit, registri a 8 e 16 bit
- x86-32 (IA-32): caratterizzata da indirizzamento su 16 e 32 bit, registri a 8, 16 e 32 bit
- x86-64 (x64): caratterizzata da indirizzamento su 16, 32 e 64 bit, registri a 8, 16, 32 e 64 bit.

L'x86-64 non viene chiamato IA-64 perché questo nome identifica un'architettura differente, l'ITANIUM, basata su tecnologia VLIW (very long instruction word).

Architettura IA-32 e x86-64 (x64) (vedi slide 26)

La semantica di un'istruzione non è funzione del codice operativo, ma funzione del contesto in cui opera. Si hanno due modalità operative:

- Protected Mode: ha introdotto i concetti di protezione, indirizzo logico e memoria virtuale; nativa del Pentium; utilizzata in ambiente Windows e Linux
- Real Mode: realizza la modalità IA-16 dell'8086; senza né protezioni né memoria virtuale; non presenta paginazione e la gestione della memoria è segmentata con segmenti fisici; utilizzata in ambiente DOS, è nativa del BIOS.

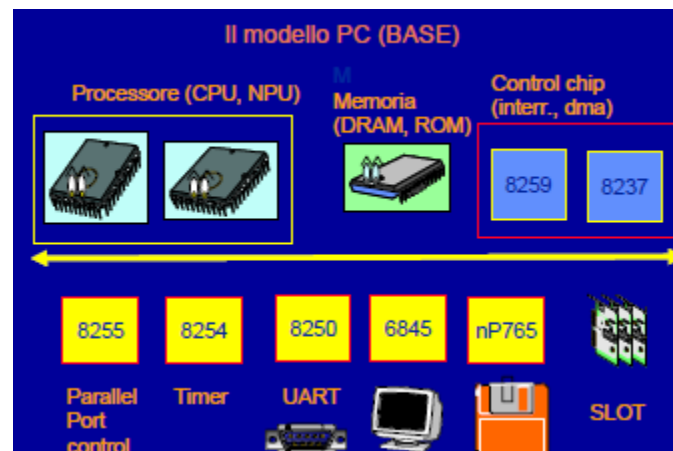
In modalità RM il codice è non rientrante nel senso che non può essere chiamato più volte da programmi diversi. Per esempio, può succedere che due processi vogliano chiamare entrambi il disco. Supponiamo di utilizzare modalità round robin per la scelta del processo. Il secondo processo per accedere alla risorsa deve attendere che il primo processo abbia finito l'esecuzione. Per evitare problemi con situazioni del genere alcuni sistemi sostituiscono le chiamate al BIOS con propri driver fisici.

Virtual 8086 Mode (vedi slide 27)

Consiste in un'emulazione in protected mode dell'ambiente real mode dell'IA-16.

Sostanzialmente si crea una macchina virtuale in cui lavorando in protected mode si emula un ambiente real mode, realizzato in modo hardware e software. Questa macchina virtuale solitamente utilizza 1 MB di RAM (che è la quantità di memoria usata in real mode).

Modello PC (Base) (vedi slide 31-32)



Valutazione performance (vedi slide da 33 a 45)

Per la valutazione delle performance si valuteranno questi elementi:

- Velocità del bus esterno
- Parallelismo
- L1 Cache

Verrà preso in esempio il caso della gestione della memoria video.

Ipotesi M24:

- macchina sequenziale e monobus
- 8086/88 8MHZ,
- bus pre - ISA (16 bit, 8 MHZ),
- tempo accesso memoria $4 \cdot T_{clk} = 4 \cdot 125ns = 500ns$

Calcolo del tempo medio di istruzione, T_m :

- Numero medio byte = 2,5 (da cui 2 accessi in memoria per fetch)
- $T_m = T_{\text{fetch}} + T_{\text{exec}} = 1000 + 500 = 1,5ms$ (0.7 MIPS)
- Trasfer rate bus = 4 MB/s

Facendo riferimento al processore ipotizzato con questi dati analizziamo il caso del refresh della memoria video.

Ipotesi:

- 600x400 pixel, true colour => 0,72 MB
- Frequenza refresh di 50 Hz, da cui transfer di 0,72MB ogni 20ms, per un totale di 36 MB/s

Per l'aggiornamento della memoria video mediante istruzioni è richiesto un tempo (min) pari a:

- $0,72 \cdot 1,5ms / 2$ (bus da 16 bit) = 540ms

Questo valore è largamente incompatibile con le caratteristiche di sistema.

Per migliorare queste prestazioni si può utilizzare un buffer di prefetch oppure un bus più veloce (per esempio, $2 \cdot T_{clk}$; evoluzione 0).

Nell'esempio precedente, se introduciamo questi due fattori si ha:

- tempo accesso memoria $2 \cdot T_{clk} = 2 \cdot 125ns = 250ns$
- $T_m = T_{fetch} + T_{exec} = 250ns$ (4 MIPS)
- Trasfer rate bus = 8 MB/s

Nel caso del refresh memory abbiamo però un'architettura ancora insufficiente. Infatti:

- $0,72 \cdot 250ns / 2$ (bus da 16 bit) = 90ms >> 20ms

Possiamo cambiare la velocità del bus.

Ipotesi 80x86 (Pentium I):

- bus (32 bit, 30 MHz),
- tempo accesso memoria $2 \cdot T_{clk} = 2 \cdot 33ns = 66ns$

Calcolo del tempo medio di istruzione, T_m :

- Numero medio byte = 2,5 (da cui 1 accesso in memoria per fetch)
- $T_m = T_{fetch} + T_{exec} = 66ns$ (se si usa prefetch)
- Trasfer rate bus = 60 MB/s

Adesso l'architettura è sufficiente. Infatti:

- $0,72 \cdot 66ns / 4$ (bus da 32 bit) = 12ms < 20ms

Consideriamo adesso l'evoluzione 1 dei processori, cioè l'utilizzo di multi level cache e multi level bus (AGP, PCI, ISA).

Calcolo del tempo medio di istruzione, T_m :

- $T_m = \text{media}(T_{cpu}, T_{mem}, T_{IO})$ dove:
- T_{cpu} = tempo istruzioni eseguite solo nella Cpu
- T_{mem} = tempo istruzioni di accesso a memoria
- T_{IO} = tempo istruzioni di I/O

Ipotesi 1:

- $T_{\text{cpu}} = 5ns$ (200 MIPS)
- $T_{\text{mem}}: T_{\text{cache}} = 10ns$ (100 MIPS)
- $T_{\text{dram}} = 60ns$ (16 MIPS)
- $T_{\text{IO}} = \text{bus ISA (10 MHz, } 2 \cdot t_{\text{clk}}) = 200ns$ (5 MIPS)
- $T_{\text{IO}} = \text{bus PCI (66 MHz, } 2 \cdot t_{\text{clk}}) = 33ns$ (30 MIPS)

Ipotesi 2:

- 25% istruzioni eseguite solo nella Cpu
- 70% istruzioni di accesso a memoria
- 5% istruzioni di I/O
- Hit/Miss 90% Hit

Ne derivano i seguenti tempi:

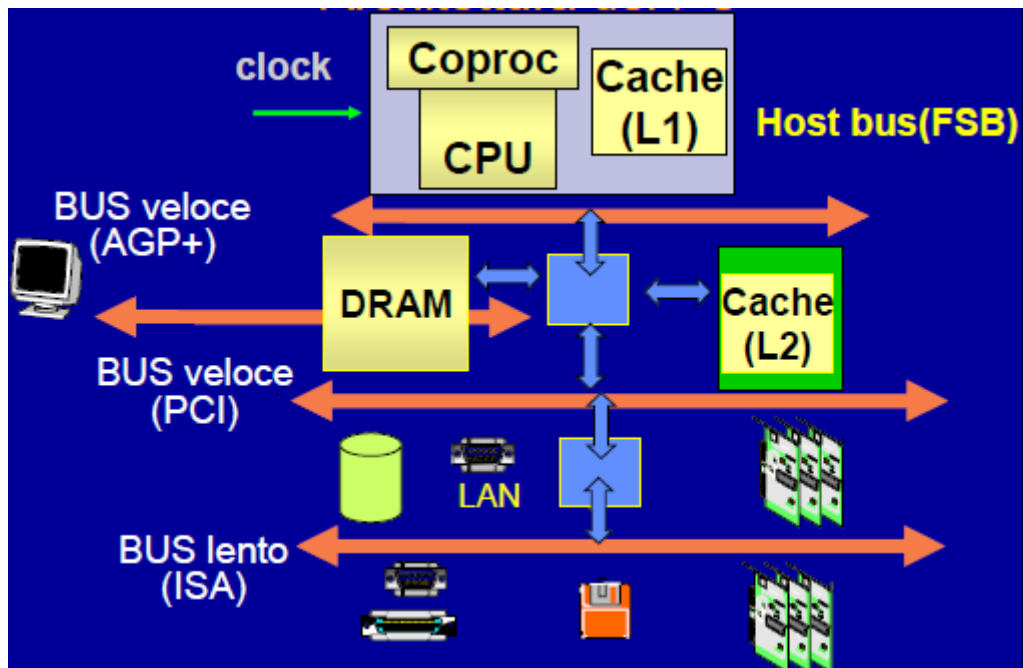
- $T_{\text{mem}} = 0,90 \cdot 10 + 0,10 \cdot 60 = 15ns$

Da cui si perviene al seguente valore del tempo medio di istruzione:

- $T_m = 0,25 \cdot 5 + 0,7 \cdot 15 + 0,05 \cdot 200 = 22ns$ (BUS ISA)
- $T_m = 0,25 \cdot 5 + 0,7 \cdot 15 + 0,05 \cdot 3 = 13,4ns$ (BUS PCI)

Si osservi come anche percentuali di incremento significativo della velocità interna (CPU) non necessariamente modifichino drasticamente le prestazioni. Ad esempio, una prestazione del 100% in più (metà tempo di T_{cpu}) riduce solo di circa il 5% il tempo totale, T_m , e quindi i MIPS. Invece, si nota come variando il bus di I/O variano le prestazioni anche se le istruzioni di I/O rappresentano solo il 5% del totale.

Architettura PC (vedi slide 46)

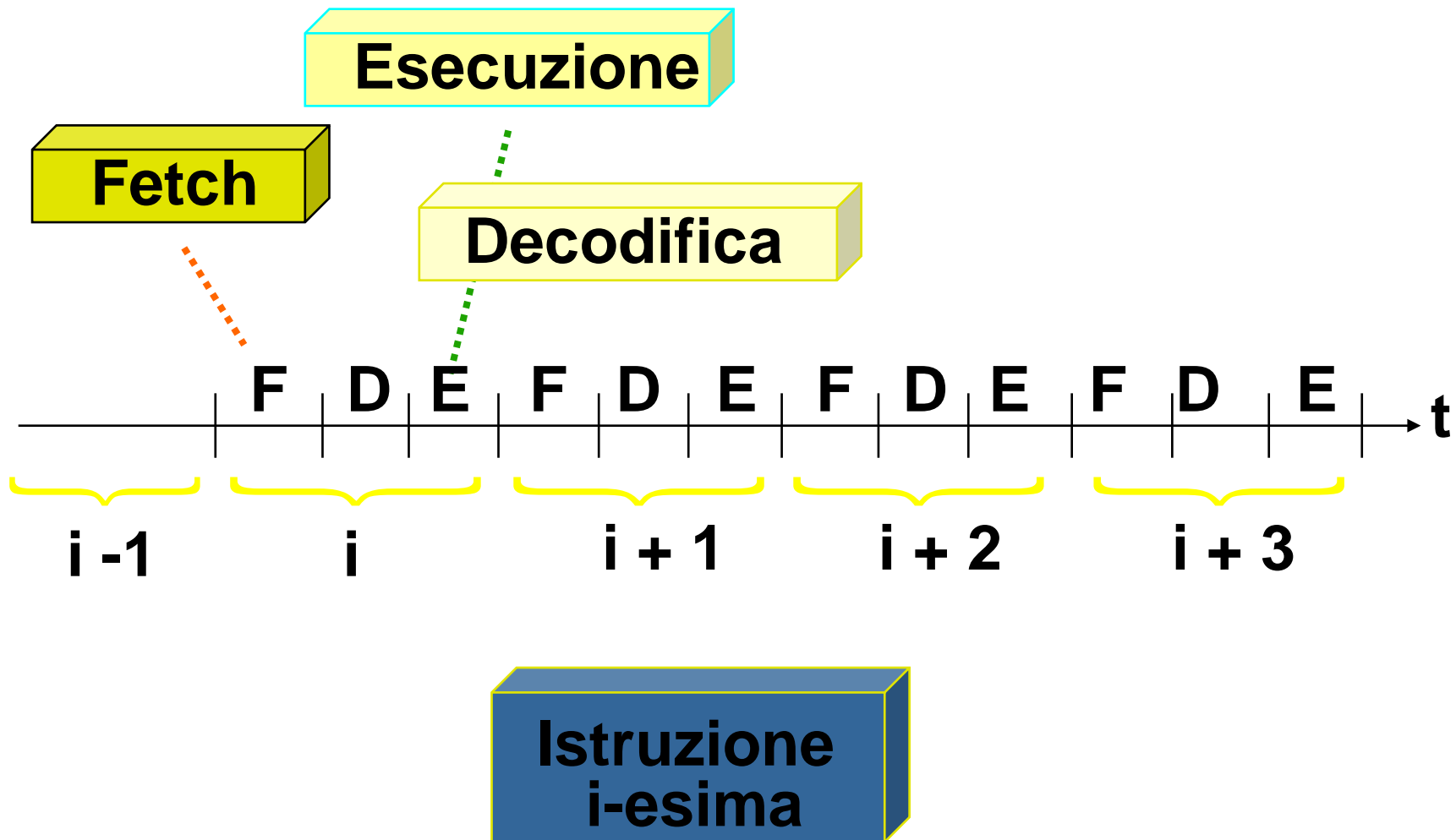


Nuova legge di Moore (vedi slide da 47 a 49)

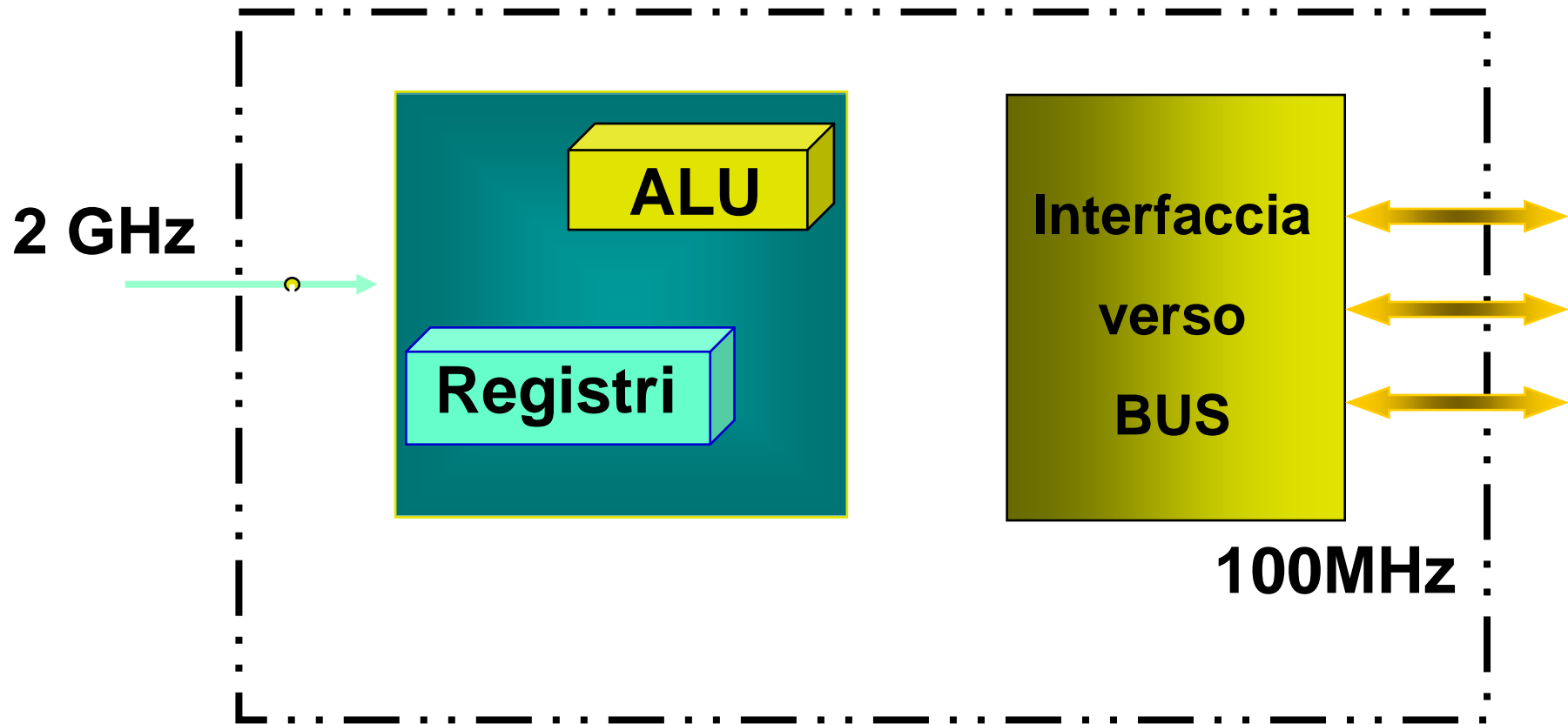
Si applica al numero di core e afferma che questo numero raddoppia ogni 18 mesi.

Viene citata perché l'evoluzione 2 dell'architettura dei processori prevede l'utilizzo di sistemi dual core o quad core con un dual instruction set (assembly level – machine level) e l'utilizzo di un parallelismo implicito a livello hardware o di compilatore.

LEZIONE 17 Settembre 2008



Clock interno e di sistema



**Migliorare
Le prestazioni!**



**Disaccoppiare i bus di sistema (BIU)
dalla cpu mediante buffer:**

- Coda di prefetch
- L1 cache

**Aumentare il parallelismo di
esecuzione:**

- Pipeline
- Architettura superscalare

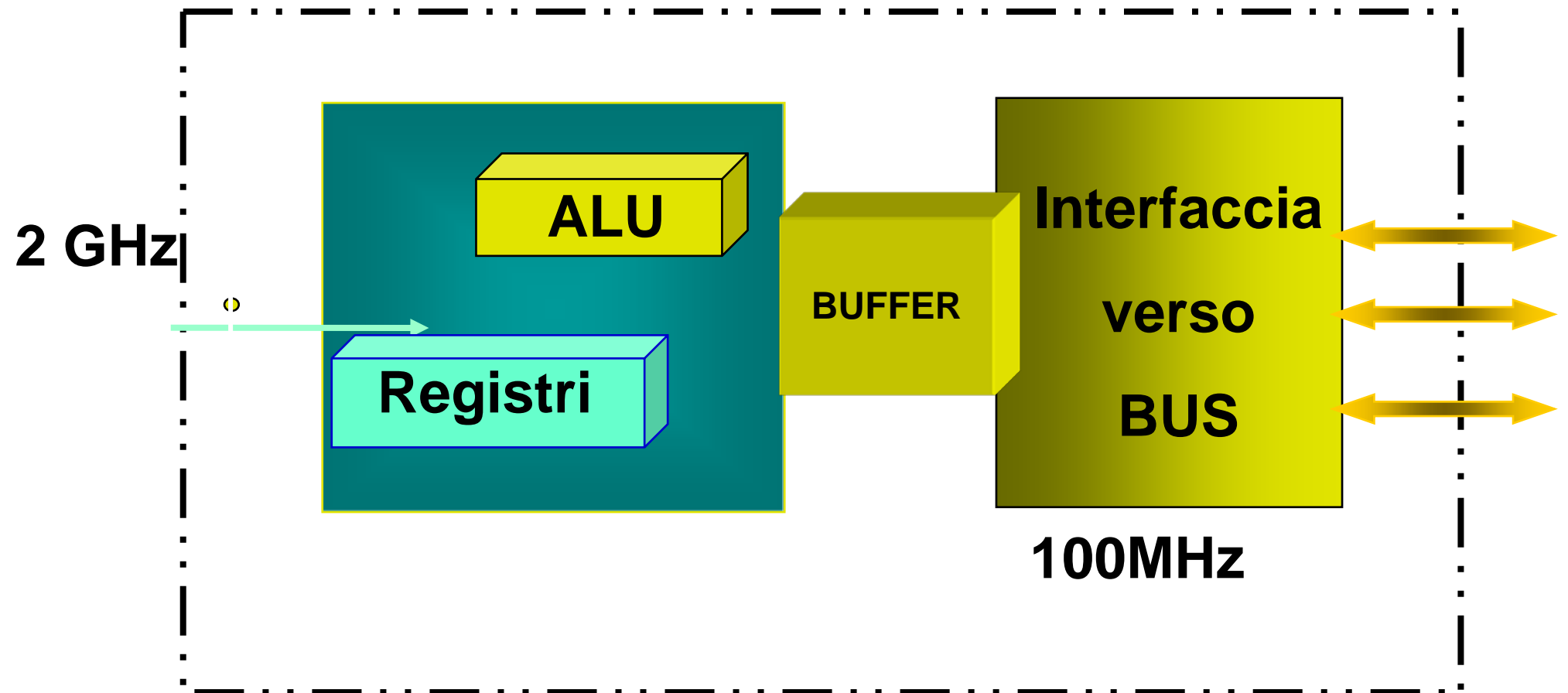
**Migliorare
Le prestazioni!**

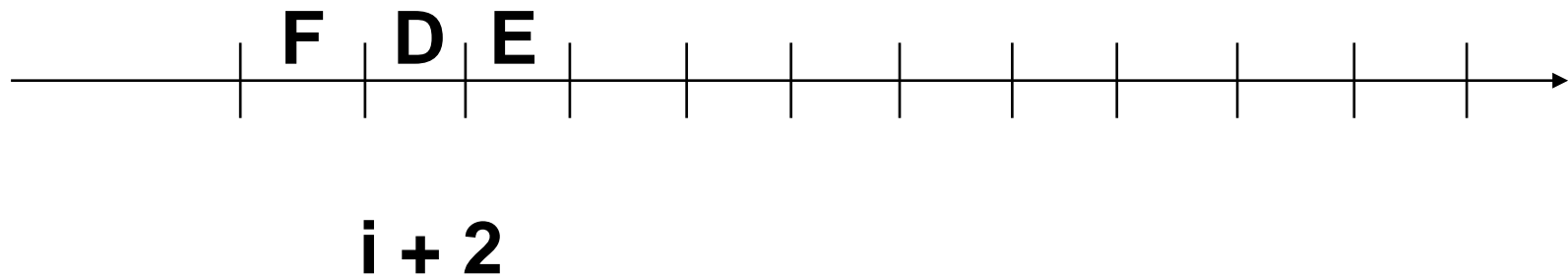
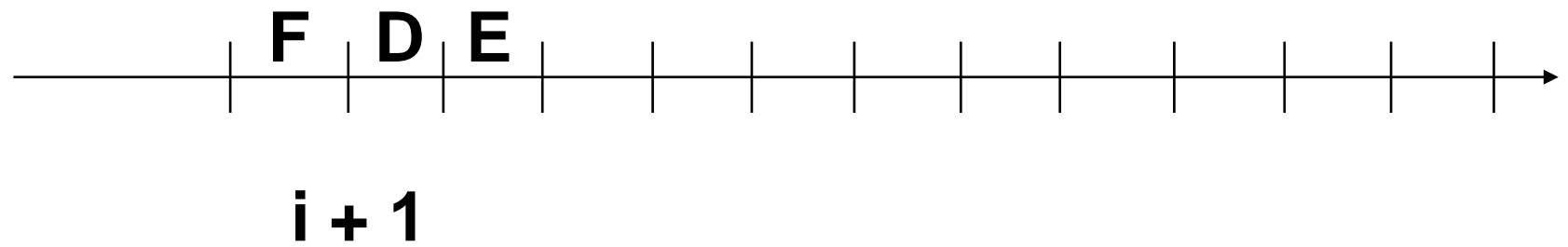
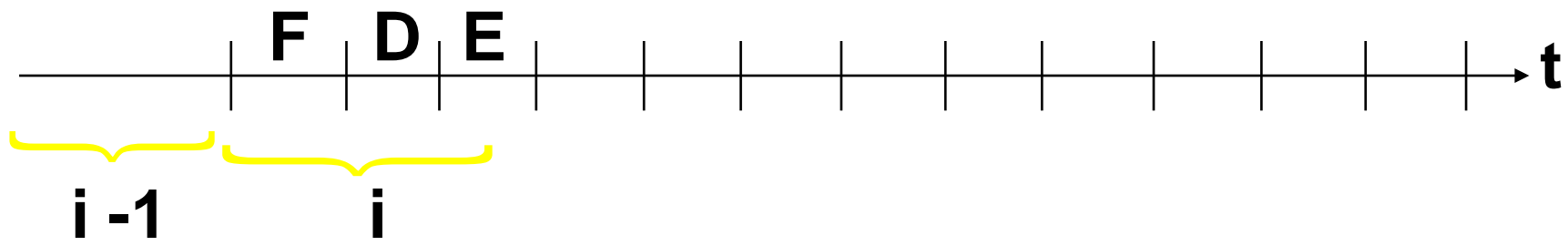


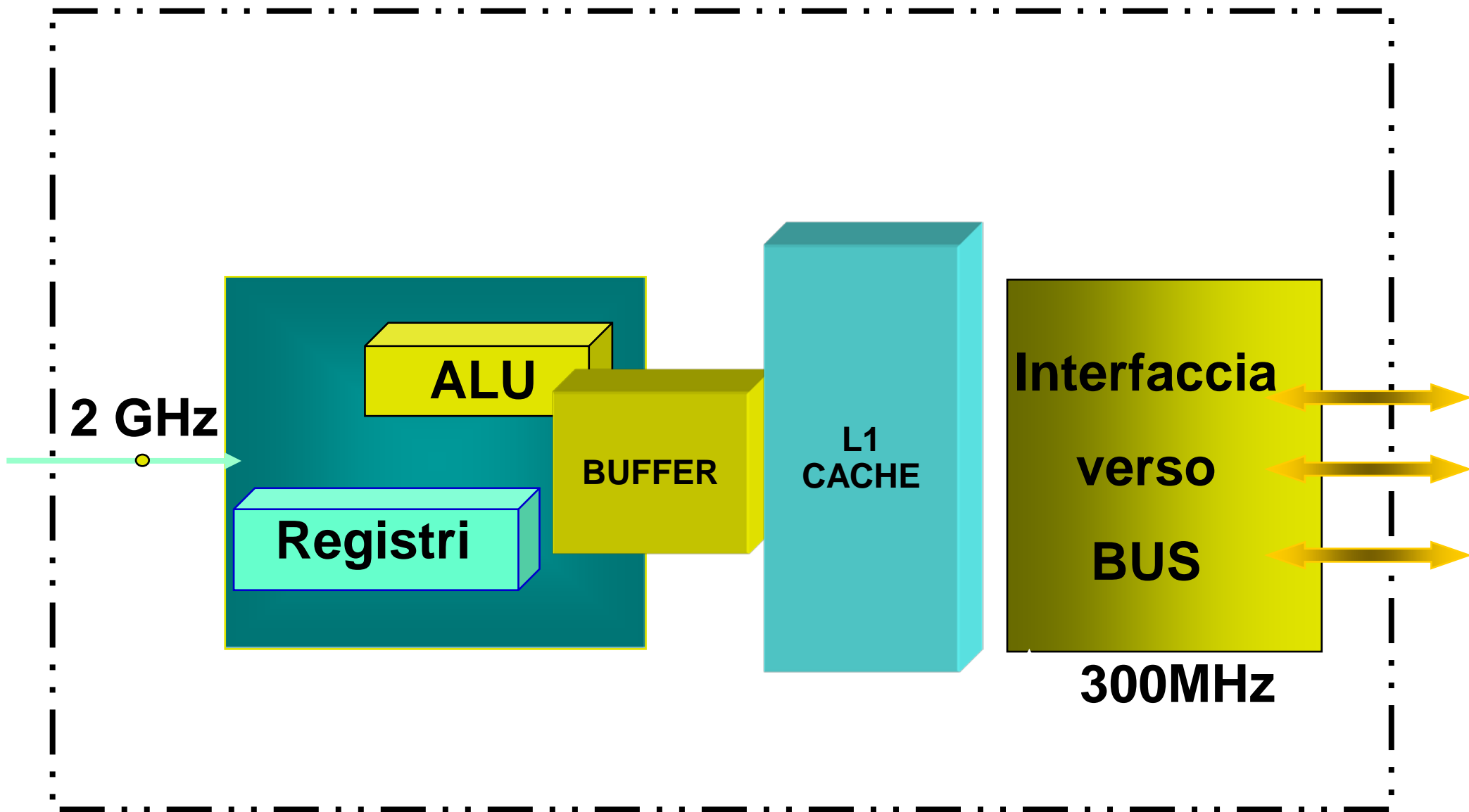
Nei sistemi multi core, aumentare il parallelismo tra thread

Si passa da strategie ILP (instruction level parallelism) a TLP (thread level parallelism)

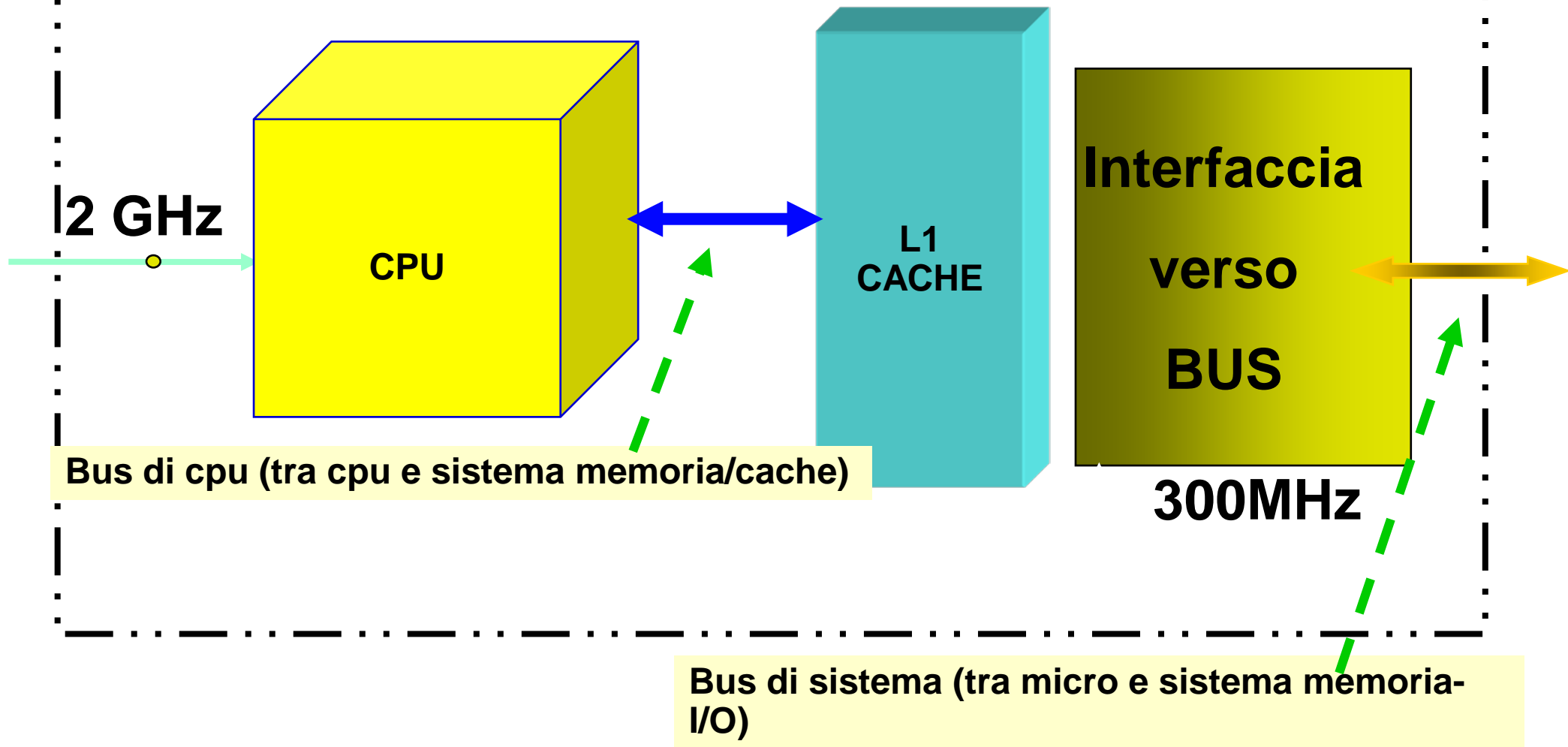
Questo oltre a apposito hw richiede nuovi criteri di programmazione







I due bus (interno-esterno) del microprocessore



Architettura 80x86

X86-16

- Indirizzamenti su 16 bit (offset)
- Registri 8, 16 bit
- Nativa su 8086 e 286

X86-32

- Indirizzamenti su 16, 32 bit (offset)
- Registri 8,16,32 bit
- Nativa su 386,486, pentium

X86-64

Indirizzamenti su 16, 32 bit, 64 (offset)
Registri 8,16,32, 64 bit
Nativa pentium 4

Architettura IA-16, IA-32, ...

x86 – 16 definita anche IA-16 (intel e Amd)

x86 – 32 definita anche IA – 32 (intel e Amd)

x86 – 64 (intel e Amd)

IA – 64 architettura totalmente differente (Itanium)
Basata su tecnologia VLIW (very long instruction)

Architettura IA-32 e x86-64 (x64)

PROTECTED MODE

È il modo nativo del pentium in cui tutte le funzionalità architetturali sono disponibili
Ambiente di Windows NT, 2000, xp, vista, Linux

REAL MODE

Realizza la modalità 8086 in modello IA – 16
Ambiente di MS-DOS
È attivo all'avvio del processore (boot S.O.)
Nativo BIOS

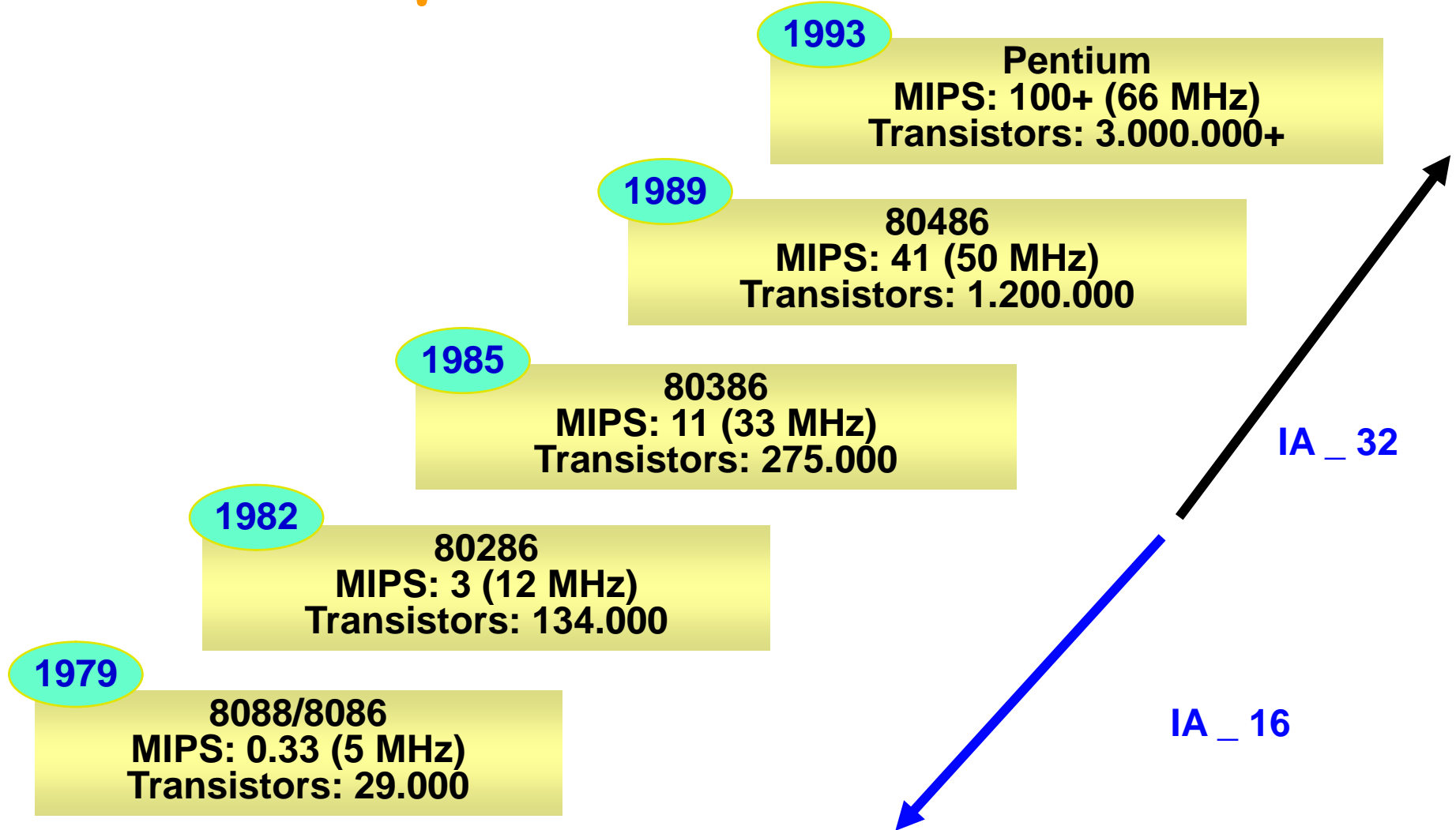
Architettura IA-32

VIRTUAL 8086 MODE

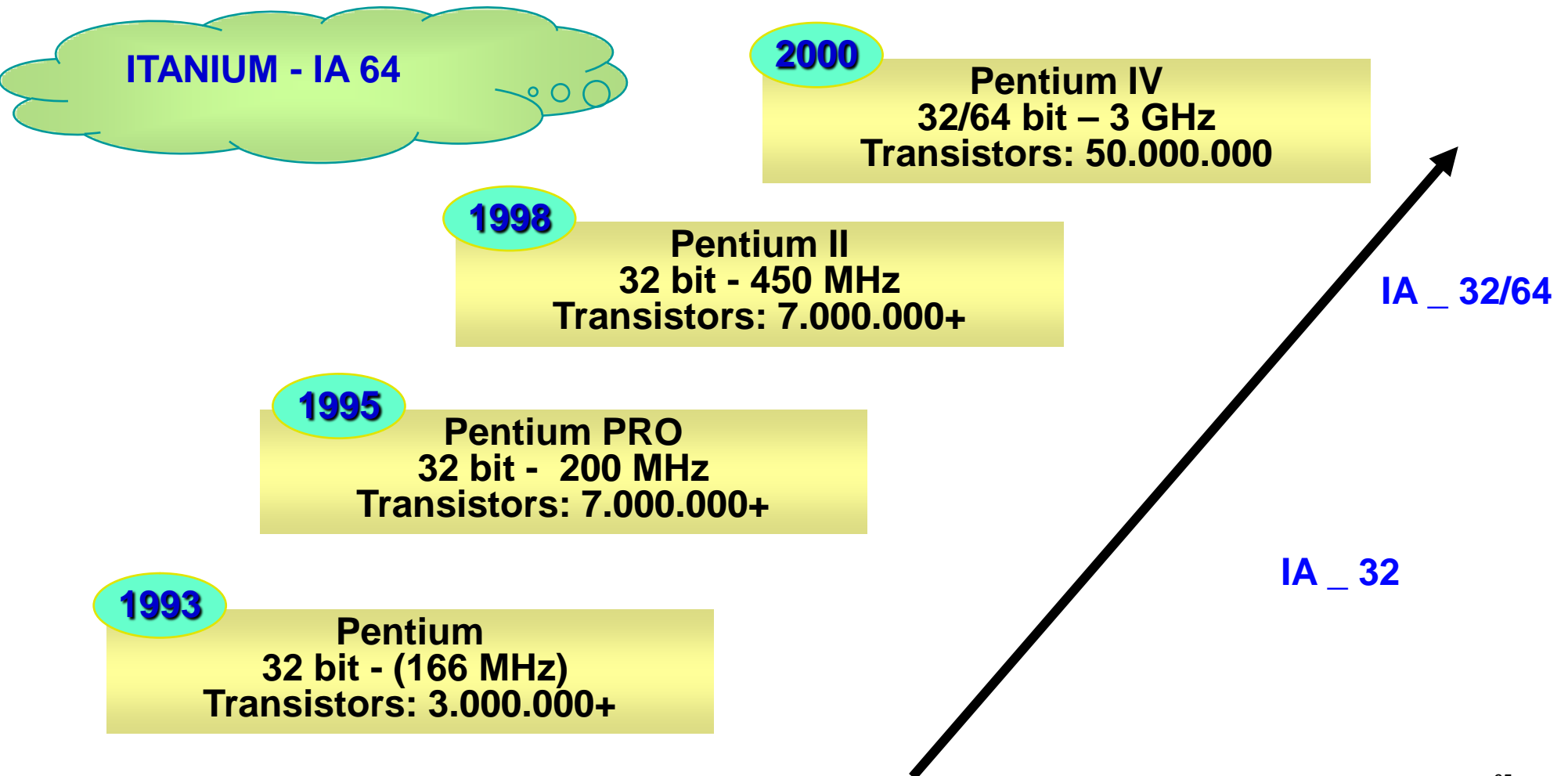
Emulazione in modo protetto dell'ambiente real IA-16

Impiegato in Win per emulare su più task il mondo DOS

Microprocessori INTEL 80x86



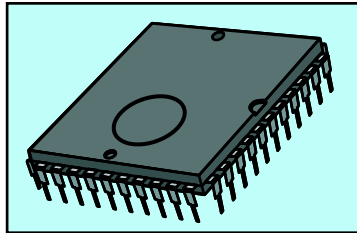
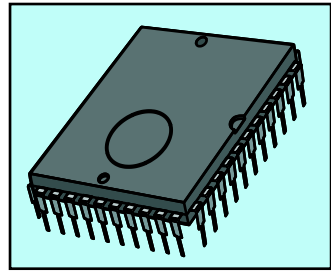
Microprocessori INTEL 80x86



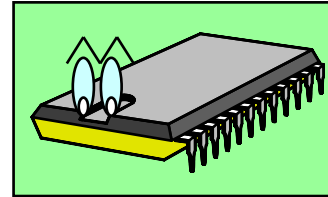
TYPE	YEAR	D/A bus	L1 cache	System CLK	CPU CLK
8086	1978	16/20		4.77 -8	4.77-8
386	1985	32/32		16-33	16-33
486	1992	32/32	8	25-50	50-100
Pentium	1993	64/32	8+8	60	200
Pentium pro	1995	64/32	8+8	66	200
Pentium II	1998	64/36	16+16	66/100	450
Pentium IV	2000	64/36	12+8+L2	300	2500

Il modello PC (BASE)

Processore (CPU, NPU)



M
Memoria
(DRAM, ROM)



Control chip
(interr., dma)

8259

8237



8255

8254

8250

6845

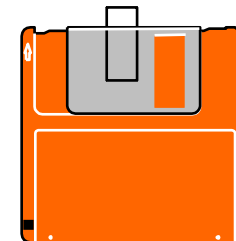
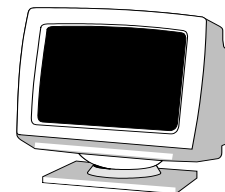
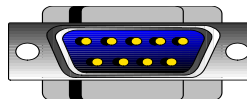
nP765



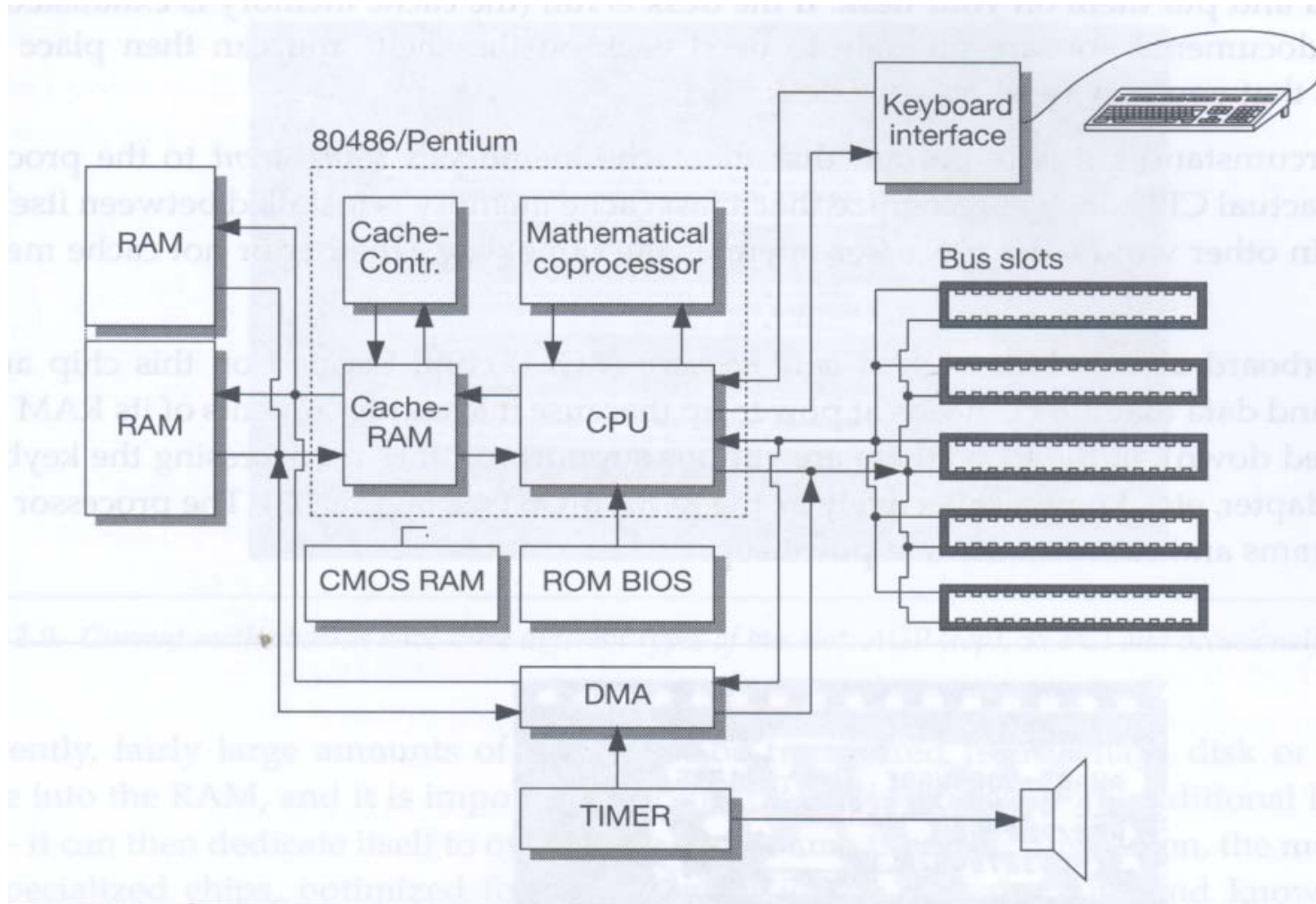
Parallel
Port
control

Timer

UART



SLOT



PERFORMANCE EVALUATION

Nel seguito vengono considerati, *il termini quantitativi*, i miglioramenti prestazionali introdotti da alcune evoluzioni tecnologiche-architetture sui microprocessori, In particolare si valuteranno i seguenti elementi:

- *Velocità del bus esterno*
- *Parallelismo(prefetch, ...),*
- *cache L1*

Il contesto è la gestione della memoria video

PERFORMANCE EVALUATION

Ipotesi M24:

8086/88 8MHz,

bus pre - ISA (16 bit, 8 Mhz),

tempo accesso memoria $4 \cdot t_{clk} = 4 \cdot 125 = 500 \text{ ns}$

Calcolo del tempo medio di istruzione, T_m

Numero medio byte = 2,5 (da cui 2 accessi in memoria x fetch)

$T_m = T_{\text{fetch}} + T_{\text{exec}} = 1000 + 500 = 1.5_{\text{ms}}$ (0.7 MIPS)

Trasfer rate bus = 4 MB/s

PERFORMANCE EVALUATION

Si faccia riferimento al processore ipotizzato, e cioè:

- $T_m = T_{\text{fetch}} + T_{\text{exec}} = 1000 + 500 = 1.5_{\text{ms}}$ (0.7 MIPS)
- Trasfer rate bus = 4 MB/s

CASO : Refresh video memory

- 600x400 pixel, true colour => 0.72 MB
- Frequenza refresh = 50 Hz, da cui transfer di 0.72MB ogni 20 ms, per un totale di 36 MB/s

PERFORMANCE EVALUATION

Per l'aggiornamento della memoria video mediante istruzioni è richiesto un tempo (min) pari a:

$$0.72\text{MB} * 1.5_{\text{ms}}/2 \text{ (bus 16 bit)} \Rightarrow 540 \text{ ms}$$

Largamente incompatibile con le caratteristiche di sistema!!

Architettura dei processori (0 evoluzione)

Prefetch buffer

Faster bus ($2 \cdot t_{clk}$)

PERFORMANCE EVALUATION

Ipotesi M24:

8086 8MHz,

bus ISA (16 bit, 8 Mhz),

tempo accesso memoria $2 \cdot t_{clk} = 2 \cdot 125 = 250 \text{ ns}$

Calcolo del tempo medio di istruzione, T_m

Numero medio byte = 2,5 (2 accessi in memoria x fetch)

$T_m = T_{\text{fetch}} + T_{\text{exec}} = 1000 + 250 = 250 \text{ ns}$ (4 MIPS)

Trasfer rate bus = 8 MB/s

PERFORMANCE EVALUATION

CASO : *Refresh video memory*

IPOTESI:

- 600 x 400 pixel, true colour =0.72 MB di memoria video
- Frequenza refresh = 50 Hz, pari a un trasferimento di 0.72MB ogni 20 ms, pari a 36 MB/s

Ma tenendo conto che il T_m è pari a 250 ns si ottiene:
 $0.72\text{MB} * 250\text{ns}/2 \text{ (bus 16 bit)} = 90 \text{ ms} \gg 20 \text{ ms!}$

L'architettura è ancora insufficiente!

PERFORMANCE EVALUATION

Ipotesi 80x86 (pentium I)
bus (32 bit, 30 Mhz),
tempo accesso memoria $2 \cdot t_{clk} = 2 \cdot 33 = 66 \text{ ns}$

Calcolo del tempo medio di istruzione, T_m

Numero medio byte = 2,5 (1 accessi in memoria x fetch)

$T_m = T_{\text{fetch}} + T_{\text{exec}} = 66 + 66 = 66$ (se prefetch)

Trasfer rate bus = 60 MB/s

PERFORMANCE EVALUATION

CASO : *Refresh video memory*

IPOTESI:

- 600 x 400 pixel, true colour = 0.72 MB di memoria video
- Frequenza refresh = 50 Hz, pari a un trasferimento di 0.72MB ogni 20 ms, pari a 36 MB/s

Ma tenendo conto che il T_m è pari a 250 ns si ottiene:
 $0.72\text{MB} * 66\text{ns}/4 \text{ (bus 32 bit)} = 12 \text{ ms} < 20 \text{ ms!}$

L'architettura è sufficiente!

Architettura dei processori (1 evoluzione)

Cache (multilevel)

Multi level Bus (AGP, PCI, ISA)

PERFORMANCE EVALUATION

Calcolo del tempo medio di istruzione, T_m

$T_i = \text{media}(T_{\text{cpu}}, T_{\text{mem}}, T_{\text{io}})$ dove:

T_{cpu} = tempo istruzioni eseguite solo nella Cpu

T_{mem} = tempo istruzioni di accesso a memoria

T_{io} = tempo istruzioni di I/O

PERFORMANCE EVALUATION

Ipotesi 1:

$T_{cpu} = 5ns$ (200 MIPS)

T_{mem} : $T_{cache} = 10ns$ (100 MIPS)
 $T_{dram} = 60ns$ (16 MIPS)

$T_{io} = \text{bus ISA (10 MHz, } 2 \cdot t_{clock}) = 200ns$ (5 MIPS)
 $\text{bus PCI (66 MHz, } 2 \cdot t_{clock}) = 33ns$ (30 MIPS)

Ipotesi 2

1

25% istruzioni eseguite solo nella Cpu
70% istruzioni di accesso a memoria
5% istruzioni di I/O

2

Hit/Miss 90% Hit

Ne derivano i seguenti tempi:

$$T_{\text{mem}} = 0.9 * 10 + 0.1 * 60 = 15\text{ns}$$

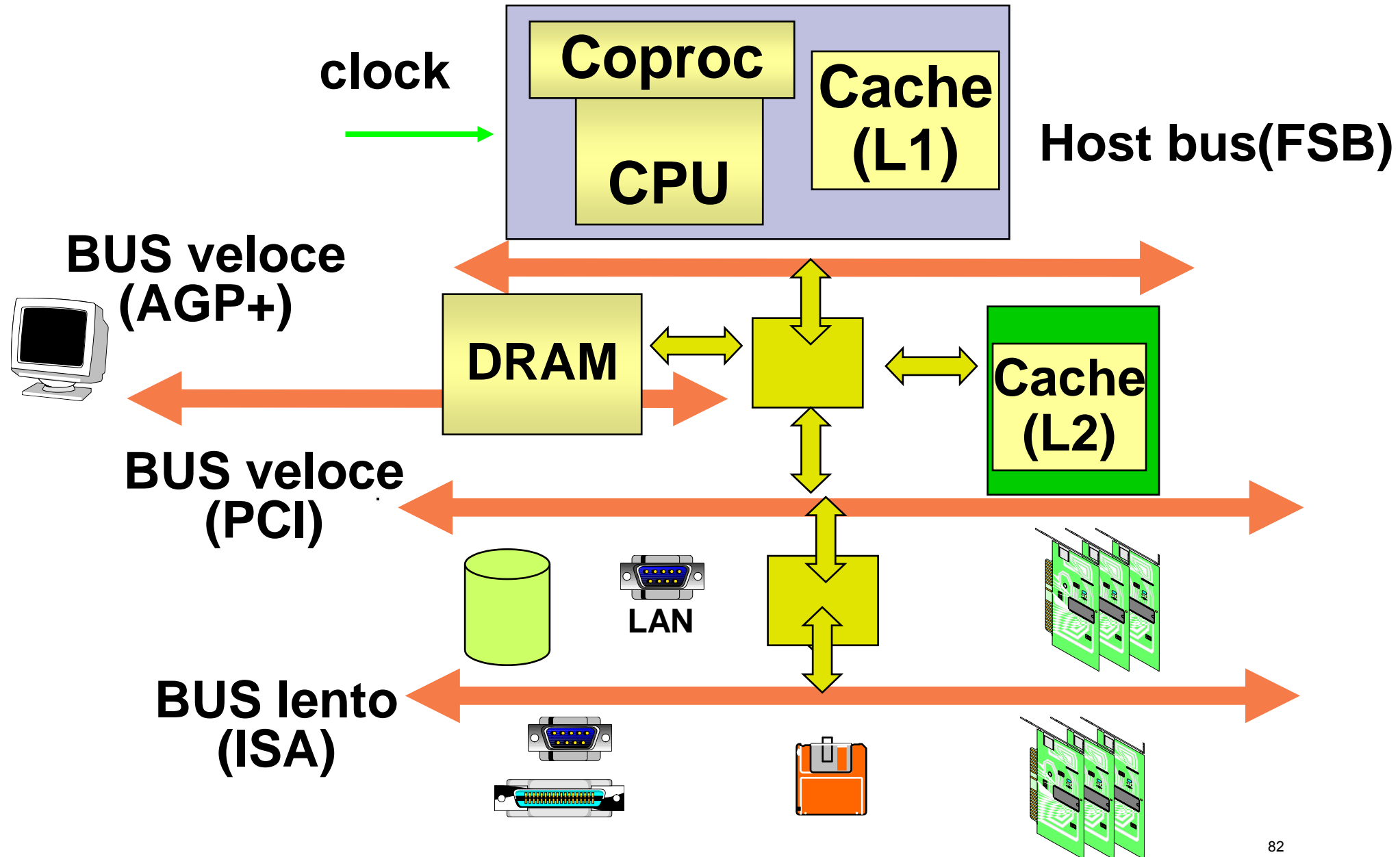
Da cui si perviene al seguente valore del tempo medio di istruzione:

$$T_i = 0.25 * 5 + 0.7 * 15 + 0.05 * 200 = 22 \text{ ns} \quad (\text{BUS Isa})$$

$$T_i = 0.25 * 5 + 0.7 * 15 + 0.05 * 33 = 13.4 \text{ ns} \quad (\text{BUS Pci})$$

Si osservi come anche percentuali di incremento significativo della velocità interna (cpu) non necessariamente modifichino drasticamente le prestazioni. Ad esempio una prestazione del 100% in più (metà tempo di T_{cpu}) riduce solo di circa il 5% il tempo totale, T_i , e quindi i MIPS.

Architettura del PC



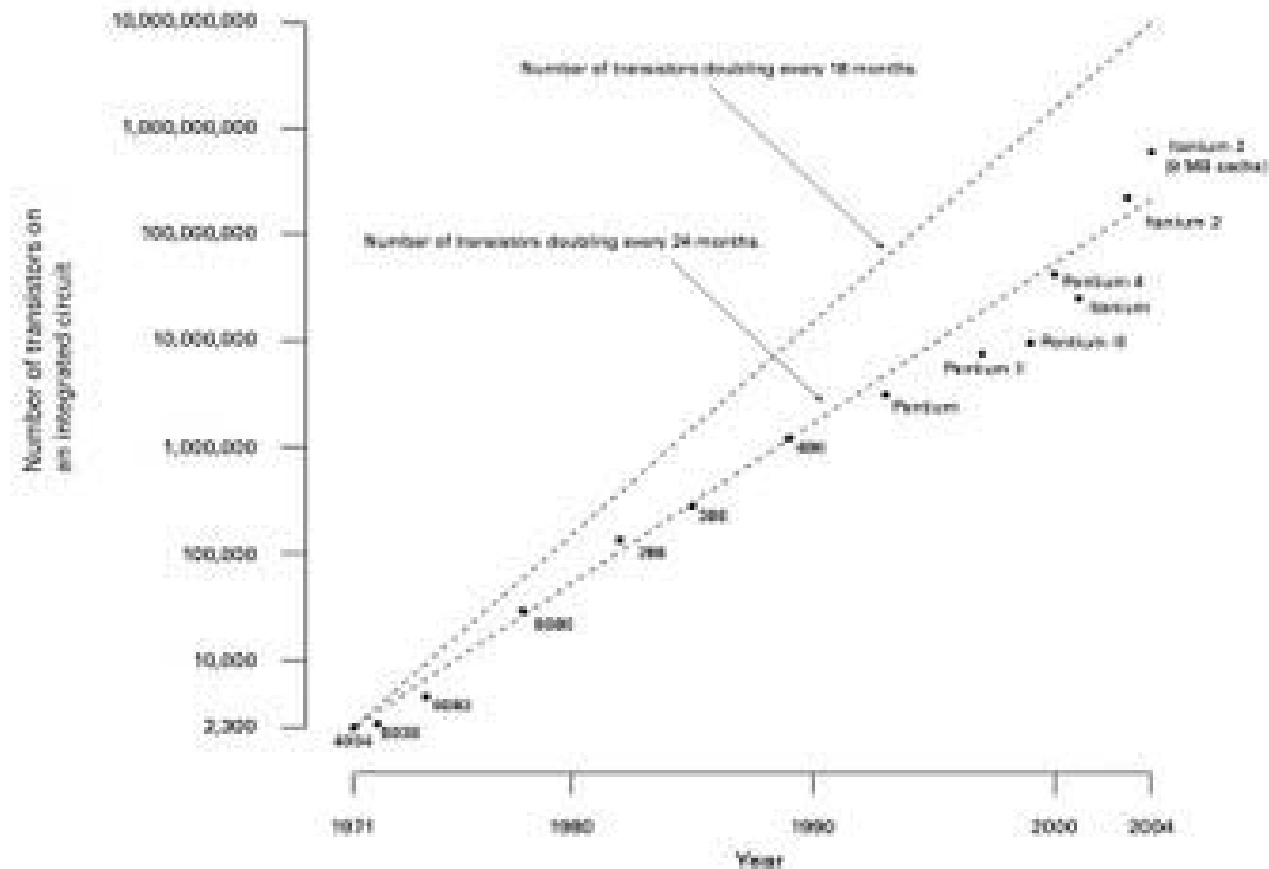
Architettura dei processori (2 evoluzione)

Dual/Four core architecture

Dual level instruction set (assembly level –machine o micro level)

Parallelismo implicito a livello hw (pentium) o compilatore(Itanium-EPIC)

Moore's Law



Growth of transistor counts for Intel processors (dots) and Moore's Law (upper line=18 months; lower line=24 months)

NEW Moore's Law

Growth of *core cpu's within a micro* doubles every 18 months;

