

## Architettura dei Sistemi Centrali (3 di 3)



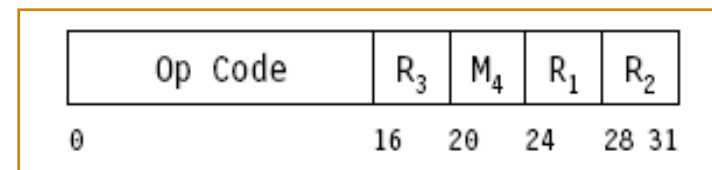
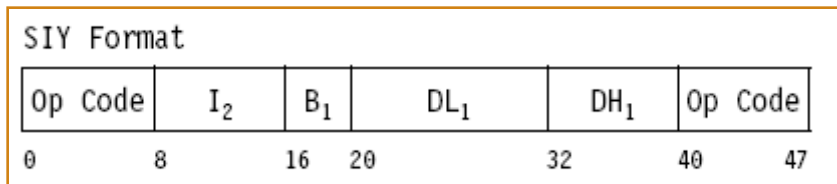
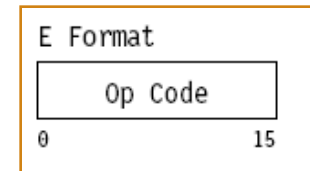
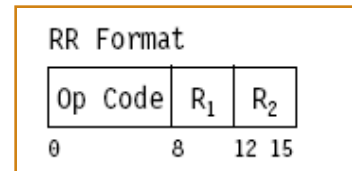
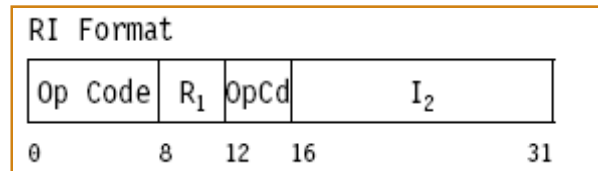
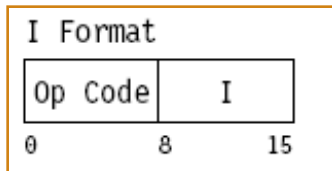
# Elementi della z/Architecture – il Set di Istruzioni CISC (Instruction set)

- **Le operazioni della CPU sono controllate da una serie di istruzioni, in memoria che, eseguite in maniera sequenziale, ed una per volta, rappresentano un programma.**
- **Un cambiamento nella esecuzione sequenziale delle istruzioni puo' essere determinata:**
  1. **Da una istruzione di salto (branch)**
  2. **Da una Interruzione (interrupt) con richiesta di LOAD PSW)**
  3. **Da un intervento esterno**
  4. **Da un segnale proveniente da un'altra CPU dello stesso CEC (Signal Processor)**

**Ogni istruzione e' costituita da due parti:**

- **Operational Code, che specifica quale operazione deve essere eseguita**
- **Operando(s) Indirizzo del dato(i) che si deve elaborare**

**Le istruzioni possono avere quindi lunghezza e formati variabili (i formati previsti dall'architettura sono 21)  
L'architettura ad oggi e' costituita da circa 640 Istruzioni.**



# Alcune Istruzioni della z/Architecture

## General Instructions:

- ADD
- SUBTRACT
- BRANCH
- COMPARE
- DIVIDE
- LOAD
- MOVE
- MOVE STRING
- STORE CHARACTER
- STORE CLOCK
- TRANSLATE
- SUPERVISOR CALL

## Decimal Instructions:

- EDIT
- ADD DECIMAL
- DIVIDE DECIMAL
- MULTIPLY DECIMAL
- .....

## Floating point Instructions:

- CONVERTE BFP to HFP
- STORE
- LOAD ZERO
- .....

## Control Instructions:

- COMPARE AND SWAP
- DIAGNOSE
- MOVE PAGE
- LOAD PSW
- SET CLOCK
- SIGNAL PROCESSOR
- PAGE IN
- PAGE OUT
- STORE CPU ID
- .....

## Hexadecimal FP Istruzioni:

- ADD NORMALIZED
- CONVERT TO FIXED
- MULTIPLY
- SQUARE ROOT
- LOAD AND TEST
- .....

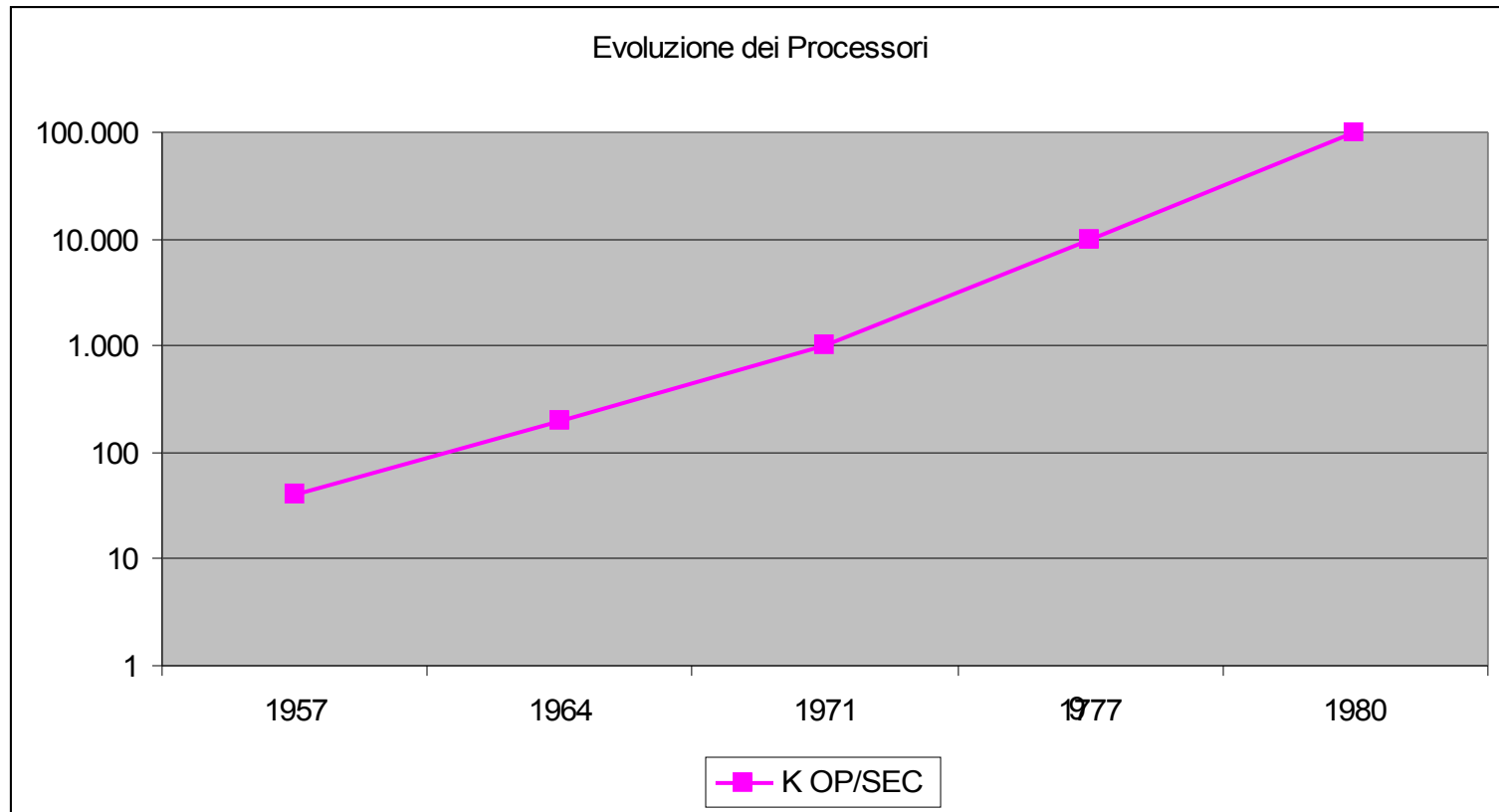
## Binary FP Instructions

- ADD
- COMPARE
- LOAD FPC
- MULTIPLY AND ADD
- .....

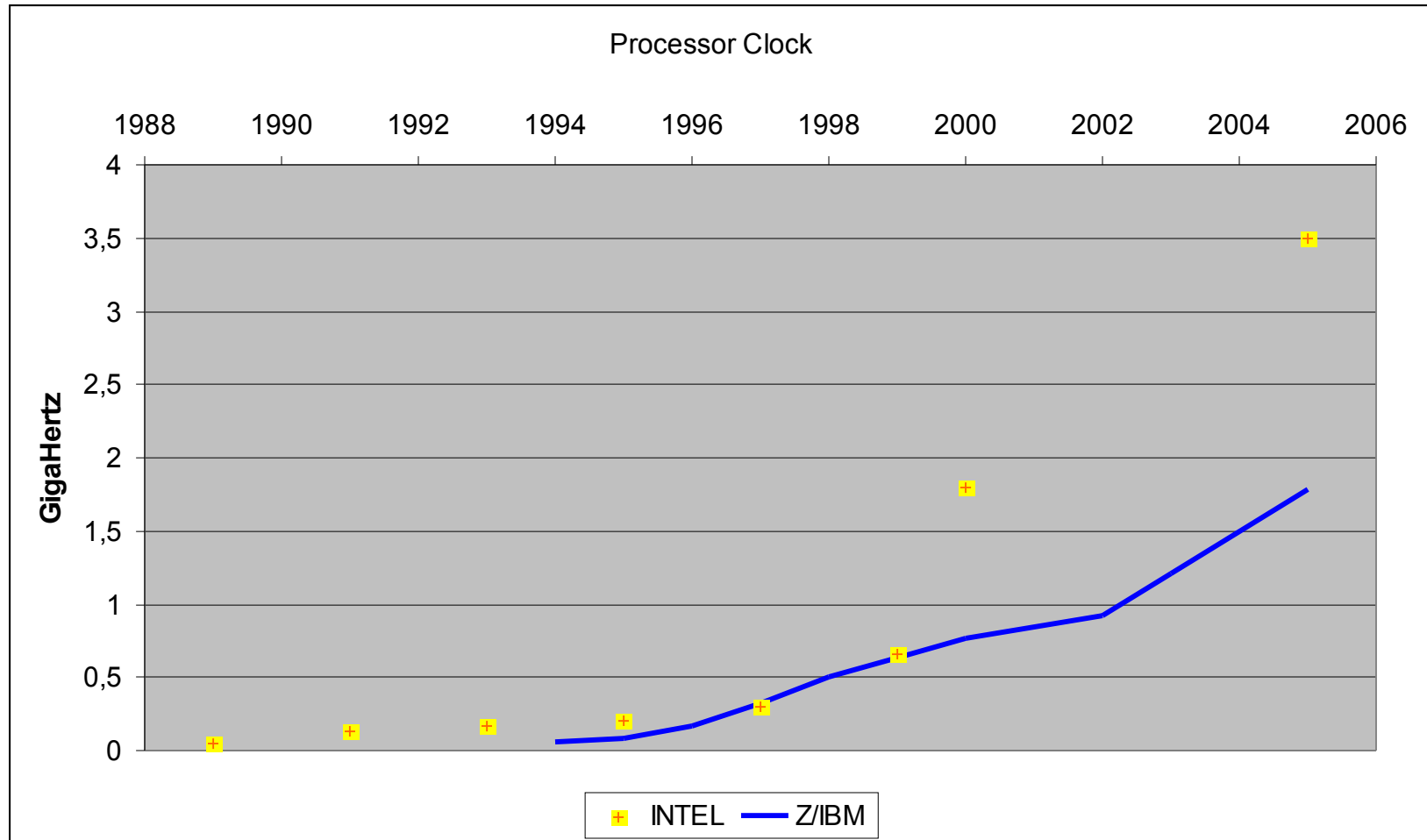
## La potenza dei Processori - Definizioni

- Ciclo Base (Nanosecondi) = Tempo medio per eseguire una istruzione Elementare di Macchina.
- Frequenza (GigaHertz) = Inverso del Ciclo Base, ovvero numero di istruzioni elementari al secondo.
- MIPS (Millions Instructions per Second) = Numero di Istruzioni CISC eseguite al secondo (in milioni).
- Cicli per Istruzione = Numero medio di istruzioni macchina per istruzione CISC si ottiene da Frequenza/MIPS
- MIPS UNI = Milioni di istruzioni CISC eseguite da una macchina con un solo processore.
- MIPS Tot = Milioni di istruzioni CISC eseguite da una macchina con il massimo di processori attivi
- MSU (Millions Service Units) = Unita' di misura tipica del Sistema Operativo z/OS, OS/390, MVS.

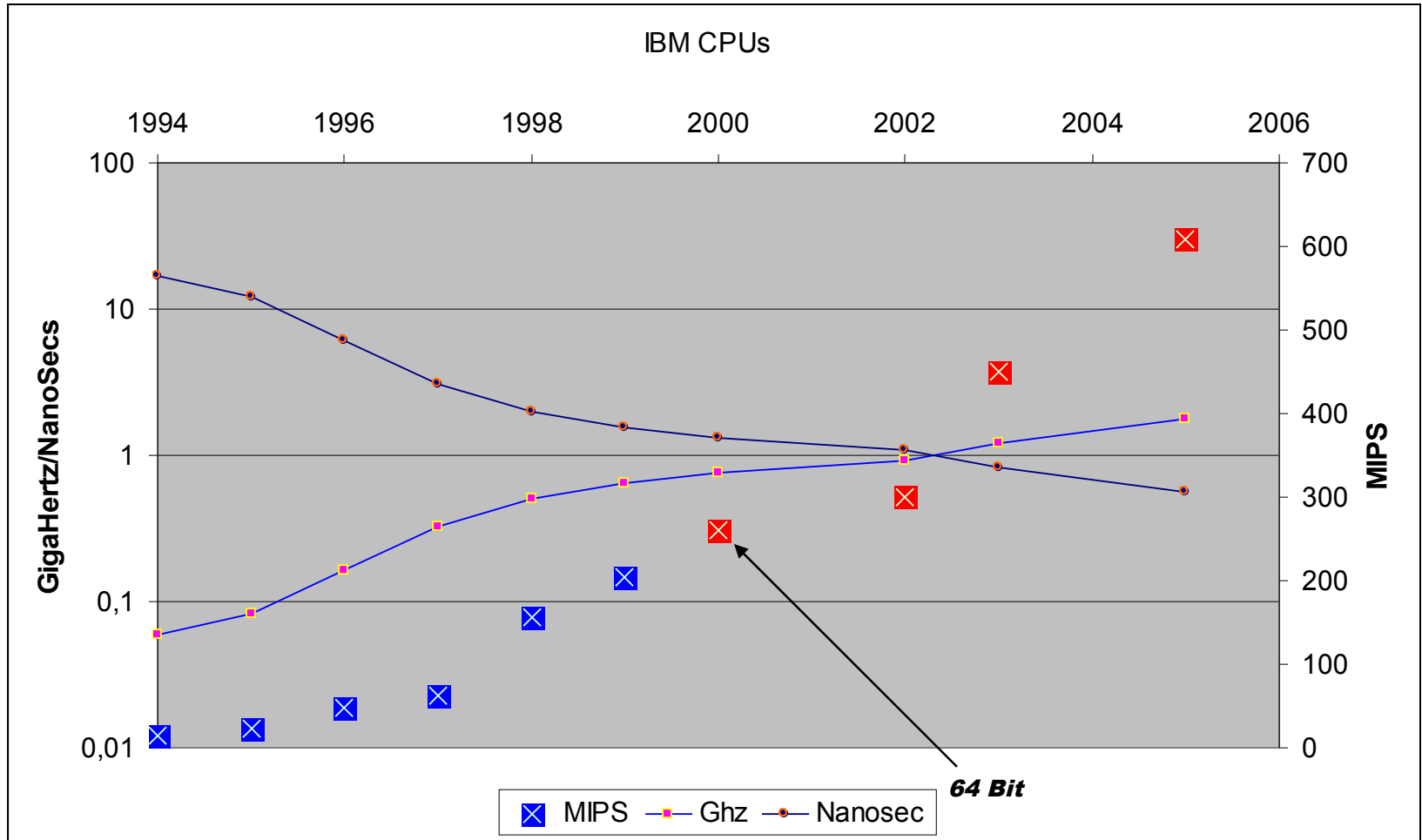
## Evoluzione dei Processori – Operazioni al Secondo



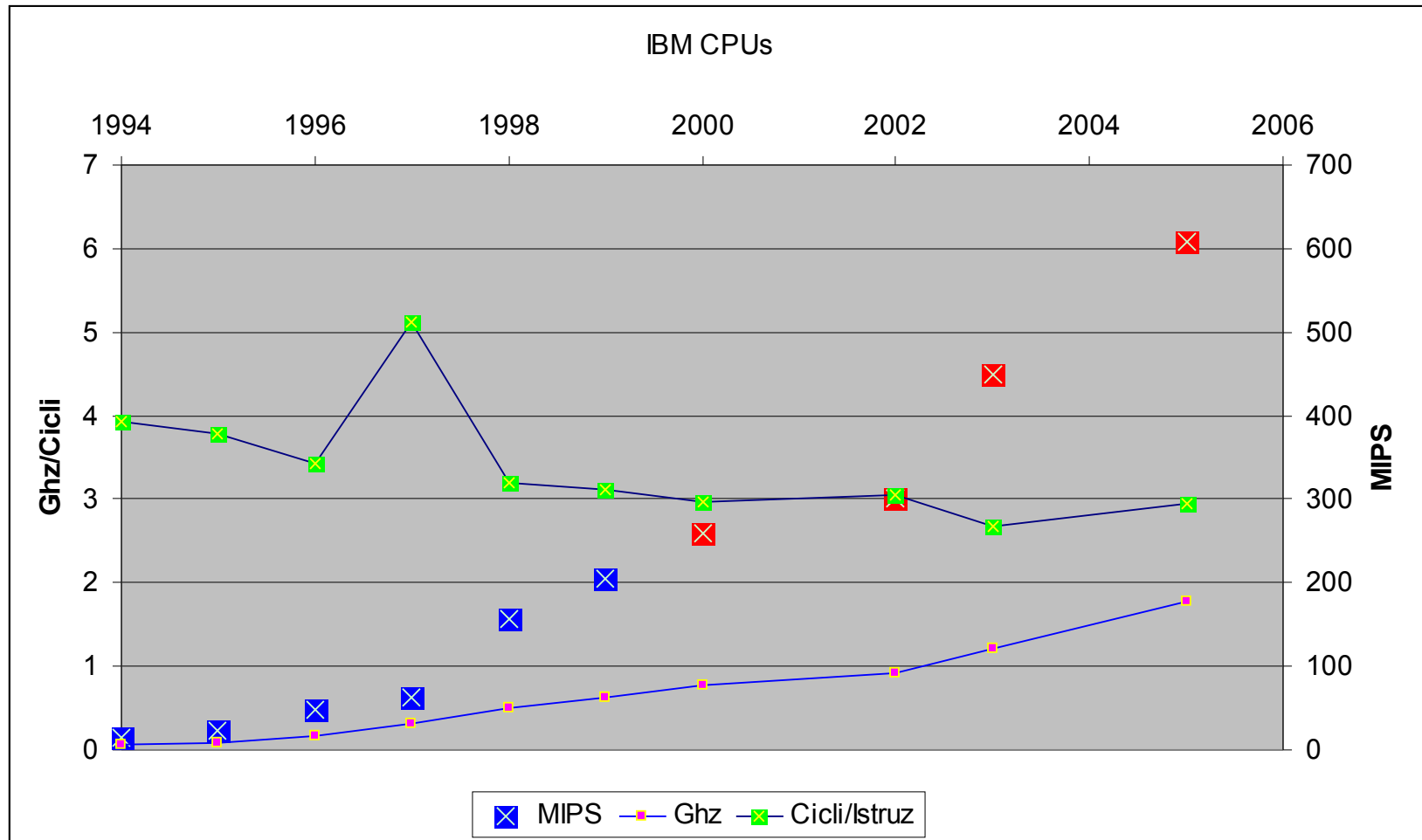
## Evoluzione dei Processori – Processor Clock



# Evoluzione dei Processori – Metriche dei Processori IBM S/390 & Z

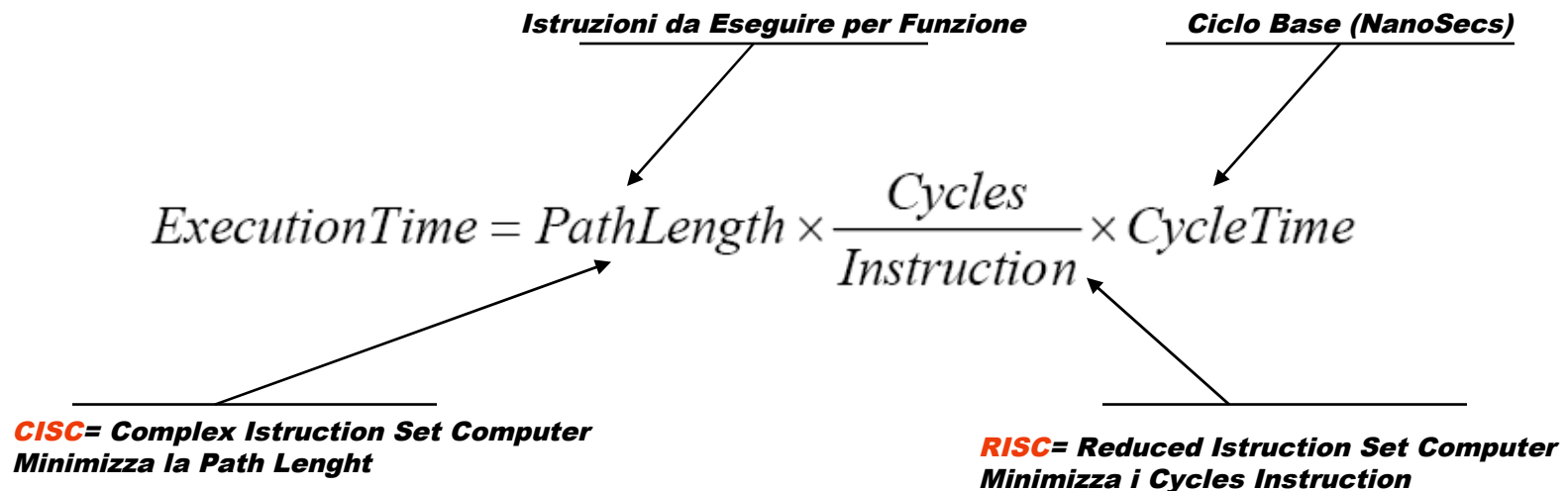


# Evoluzione dei Processori IBM per i Mainframe





## Differenze tra architetture - CISC e RISC



*La frequenza di Clock (inverso del Ciclo Base) non e' l'unico elemento per misurare le prestazioni di un calcolatore*

*Entrambe le implementazioni sono **Super Scalari** ovvero eseguono piu' di una ISTRUZIONE contemporaneamente*

## Contenuto del Processore z9

Dentro il chip del processore ci sono fondamentalmente 4 unità

1. La L1 cache che contiene strutture dati (arrays,directory ) TLB e address translation logic
2. La I-Unit che gestisce il caricamento dell'istruzione la decodifica e la generazione di indirizzi e contiene la coda di istruzioni in attesa di esecuzione
3. La E-Unit che contiene le varie unità di esecuzione e la copia locale di lavoro dei registri GR FPR AR
4. La R-unit ovvero la unita' di recupero(recovery) che mantiene la copia dello stato "architected"del processore

## Contenuto del Processore z9

La R-Unit per ogni ciclo di clock nel quale la E-unit produce risultato questo viene scritto anche nella R-Unit

La R-Unit verifica se il risultato è corretto e poi genera un ECC su quel risultato

Il risultato “fotografato” è scritto sui registri della R-Unit insieme all'ECC

Il contenuto dei registri della R-Unit rappresentano l'esatta fotografia dello stato del processore ad ogni ciclo di clock utilizzabile per ripristinare in caso di errore hardware

Millicode e' usato per implementare istruzioni complesse e poco frequenti

Il millicode a completo accesso R/W ai registri R-Unit

Millicode svolge molte funzioni

## Contenuto del Processore z9

Ad ogni ciclo di clock i segnali provenienti da I-unit e E-Unit inclusi i risultati delle istruzioni sono comparati nella R-Unit e nella L1 cache

Se i segnali non corrispondono si va a fare un ripristino HW

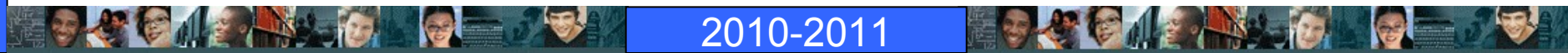
Tutti gli arrays in L1 –cache sono protetti con parità eccetto i store buffers protetti via ECC

Se R-unit o L1-cache intercettano un errore il processore entra automaticamente in stato di ripristino

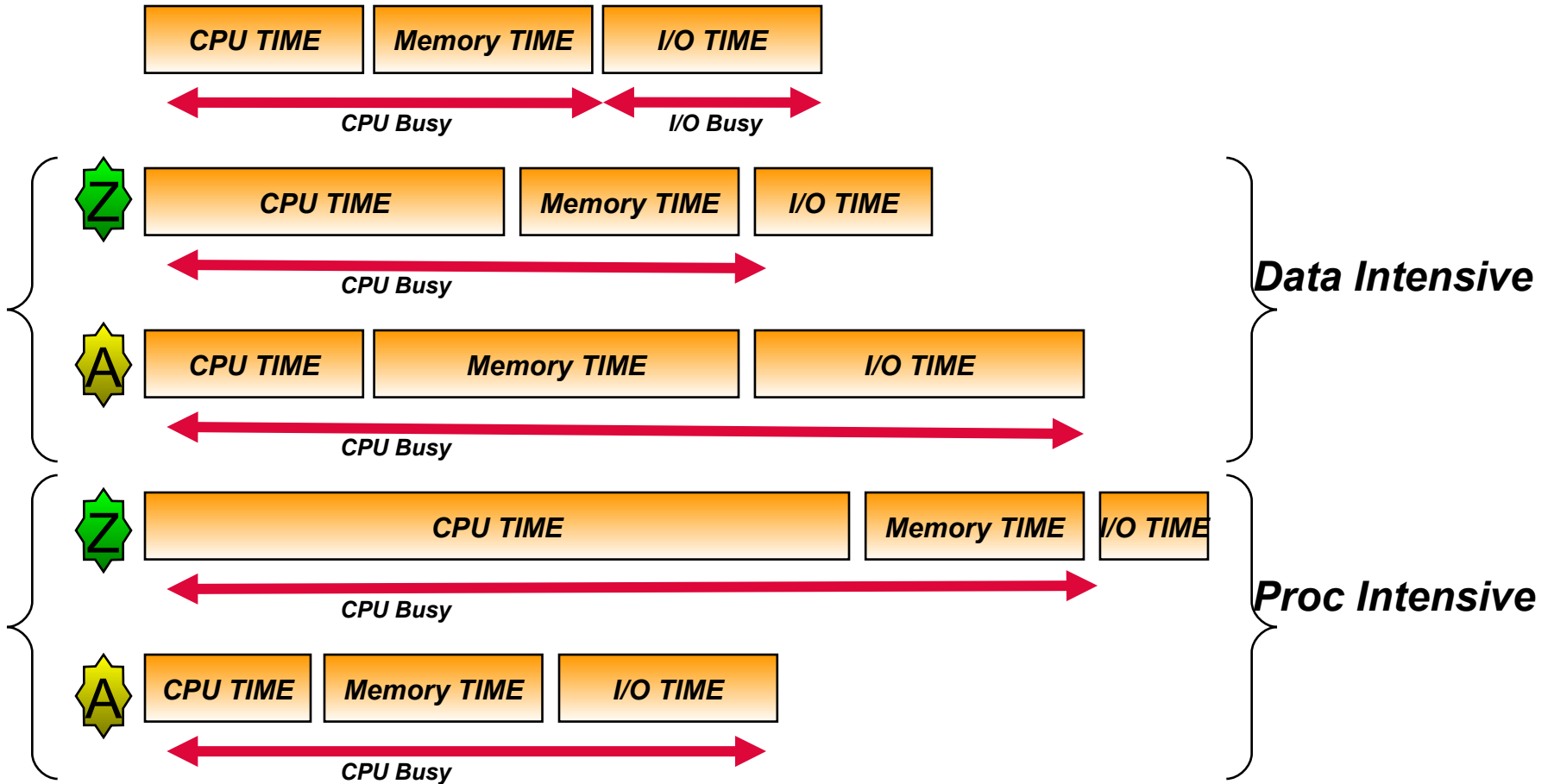
La R-Unit congela lo stato e non permette a interruzioni pendenti di modificarlo

La L1 Cache invia in L2 ogni dato memorizzato per istruzioni già congelate

Tutti gli arrays in L1 sono azzerati



# Differenze tra architetture - Performances



# Architetture - CISC e RISC

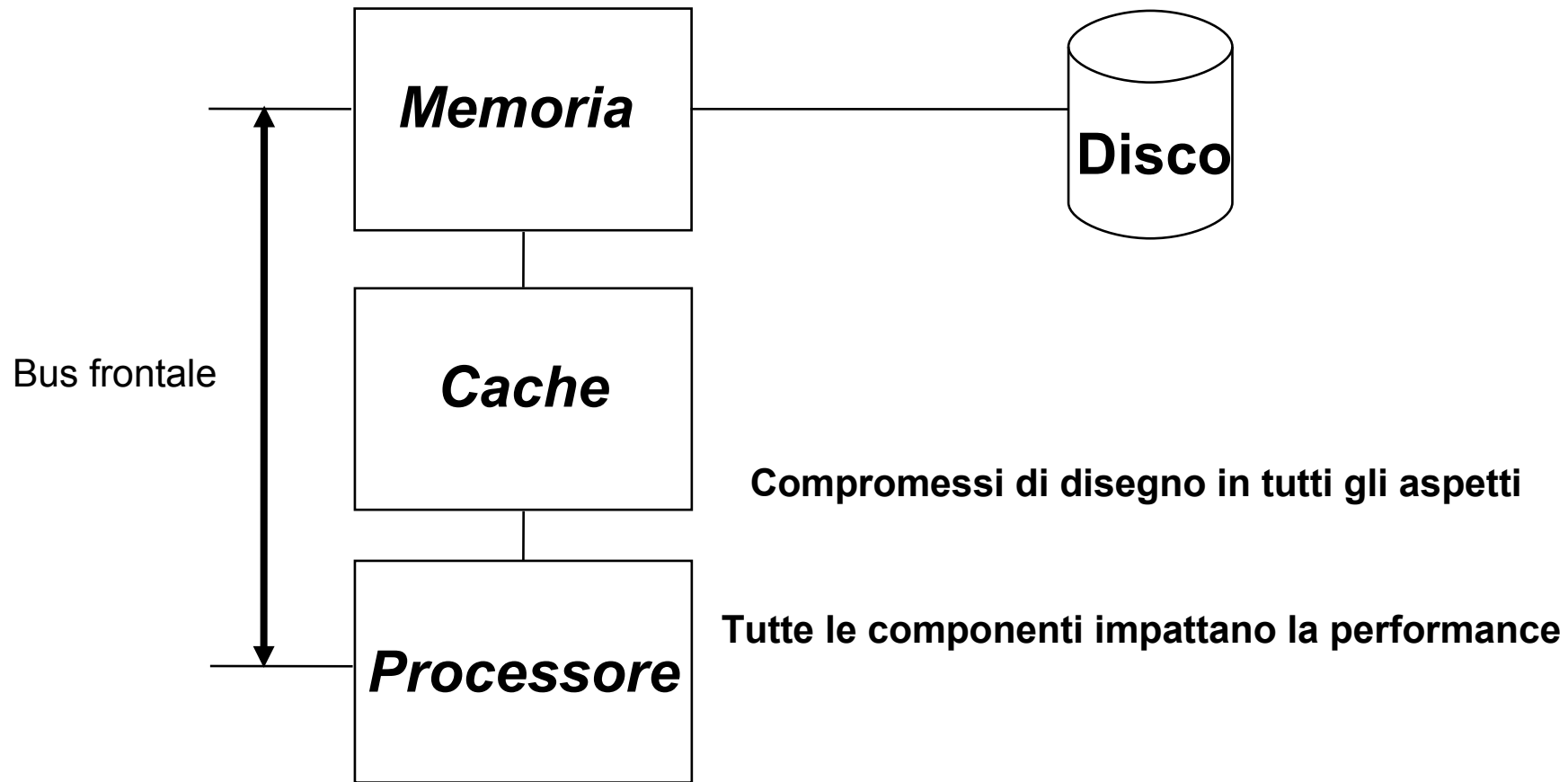
## **CISC (Complex Instruction Set Computer)**

- Tipica di z/Architecture & INTEL
- Molte Centinaia di Istruzioni
- Istruzioni molto complesse che fanno riferimento anche a diversi operandi in memoria e diversi registri.
- Pochi Registri nella CPU
- Ogni istruzione puo' richiedere anche molti cicli macchina.
- Oggi e' usuale implementare un SET di istruzioni CISC su processori che eseguono istruzioni RISC (Pentium , z9).

## **RISC (Reduced Instruction Set Computer)**

- Tipica dei sistemi UNIX
- Poche Centinaia di Istruzioni
- Istruzioni molto semplici che operano sempre su una posizione di memoria ed un registro (Load & Store) e tra due registri.
- Molti Registri nella CPU
- Ogni ciclo macchina viene conclusa una istruzione (Pipeline).
- Negli ultimi anni tendono a divenire piu' complesse.

# Architettura Elaborativa semplificata

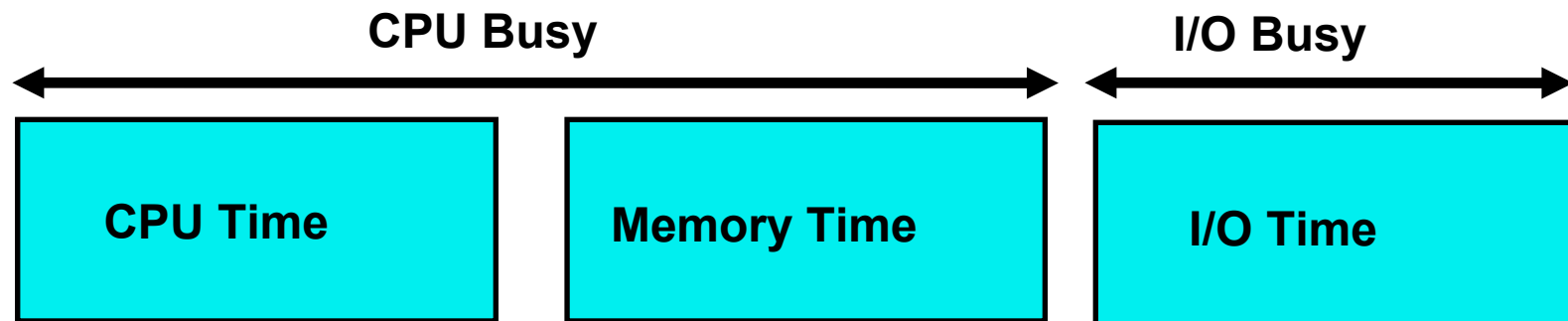




# Cosa determina la capacità del Sistema

***La performance non dipende solo dalla potenza del processore***

- **La capacità del sistema singolo dipende da:**
  - Velocità del Processore
  - Gerarchia di memoria
  - Struttura di I/O



***I tempi di processore, memoria, and I/O dipendono da applicazione e tipo di macchina***

# Considerazioni

## ■ Velocità del processore (CPU Time)

- La tecnologia del processore e' fondamentale
- Un numero limitato di “Quantita di silicio” risulta in compromessi per altre caratteristiche

## ■ Efficacia del Processore (Memory Time)

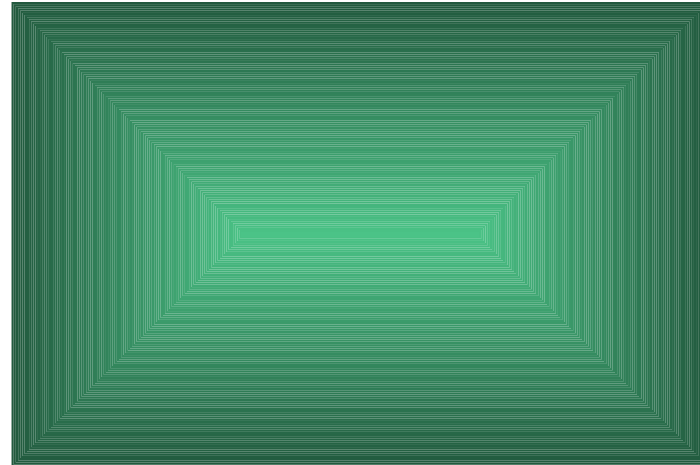
- Velocità di Caches, bus
- Capacità di I/O
- Caratteristiche del carico

## ■ Utilizzazione

- Modelli e versioni hardware e software
- Capacità di Partizionamento
- response time desiderato

## Questioni Ereditate

Di fronte ad un foglio bianco di silicone un progettista deve fare alcune scelte fondamentali...

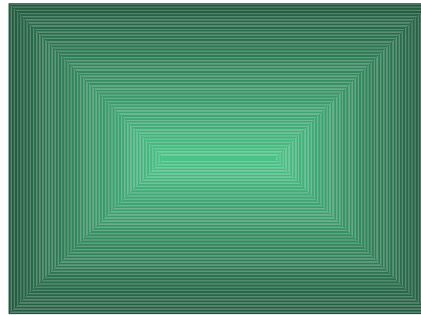


Questioni di Architettura!

# Ci sono due fondamentali architetture nel mercato attuale

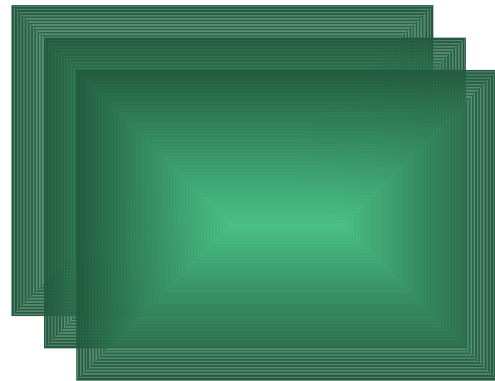
## ***RISC –Dedicated Workloads***

Dedicati / Commodity

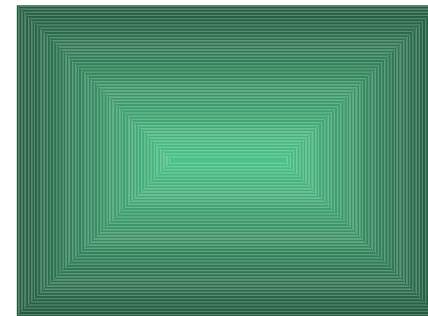


## ***CISC – Commercial Mixed Workloads***

Condivisi / Ottimizzati



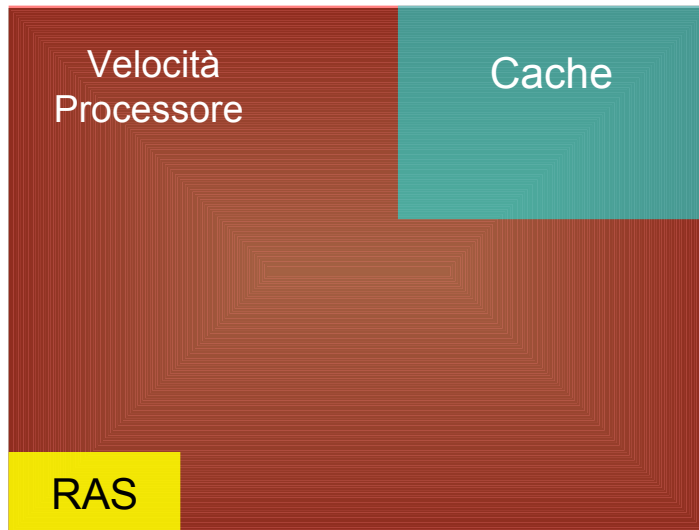
Replicati



Consolidati

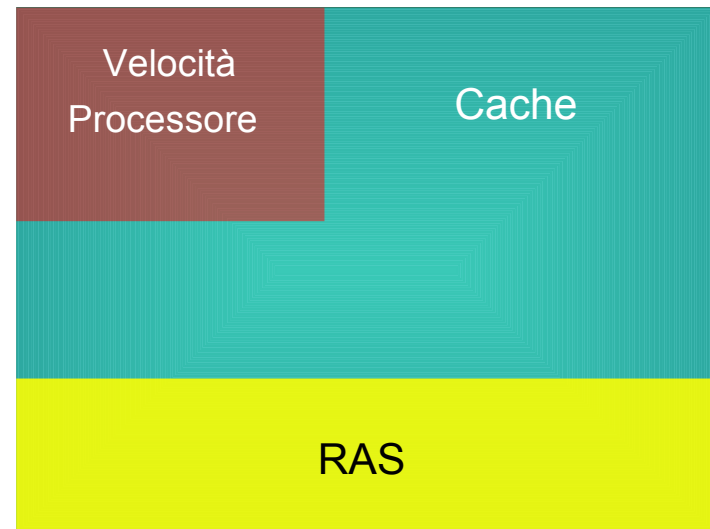
# Queste strategie portano a dei compromessi...

Piu' silicio impiego per la velocità , meno silicio e' a disposizione di Cache and RAS (e viceversa)



- Piccoli Workload – piccoli Working Set
- RAS attraverso replicazione
- Path di esecuzione più lunghi
  - Cultura MIPS economica

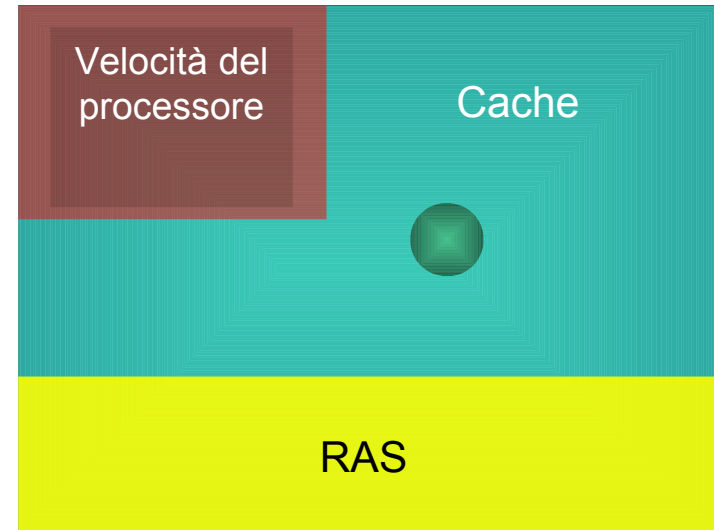
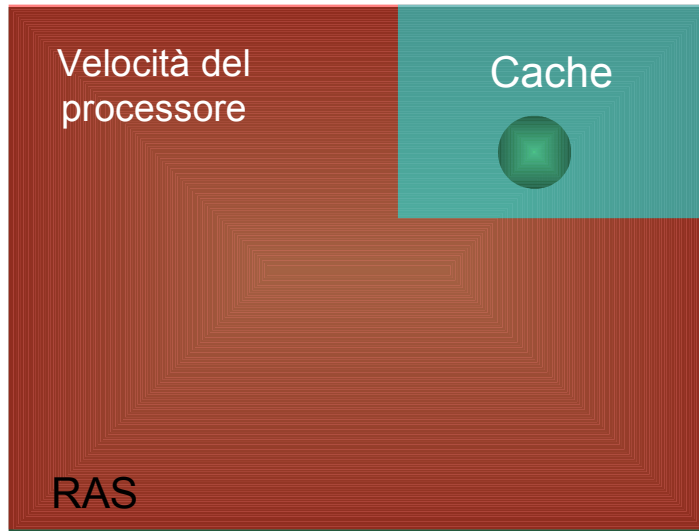
Replicati



- Mixed Workload –Working Set Larghi
- RAS intrinseca è richiesta
- workloads ottimizzati
  - Cultura MIPS di valore

Consolidati

# Le scelte di disegno interessano la performance dei “benchmark” Es.SPECint :



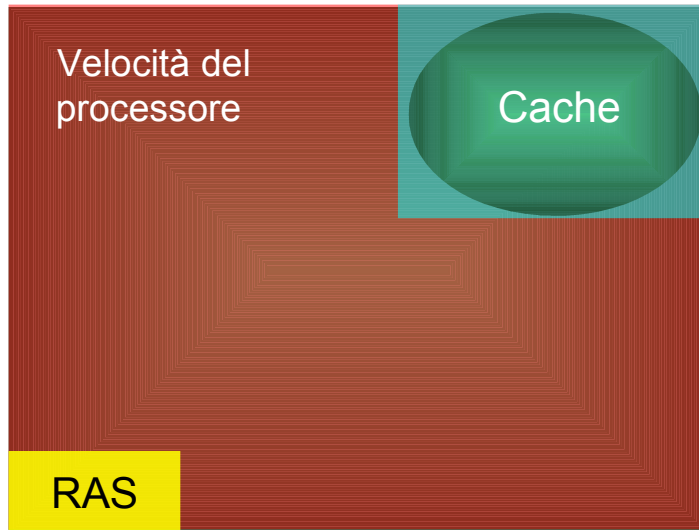
- Working set piccoli si adattano facilmente in cache
- La “leva” della velocità del processore e’ favorita
- RAS non e’ misurata
- RAS al minimo non e’ penalizzata

Replicato

- Cache grande e’ sfruttata male
- IL compromesso tra Velocità del processore e cache e’ penalizzata
- La qualità RAS non e’ valorizzata
- RAS e Cache massimizzate sono penalizzate

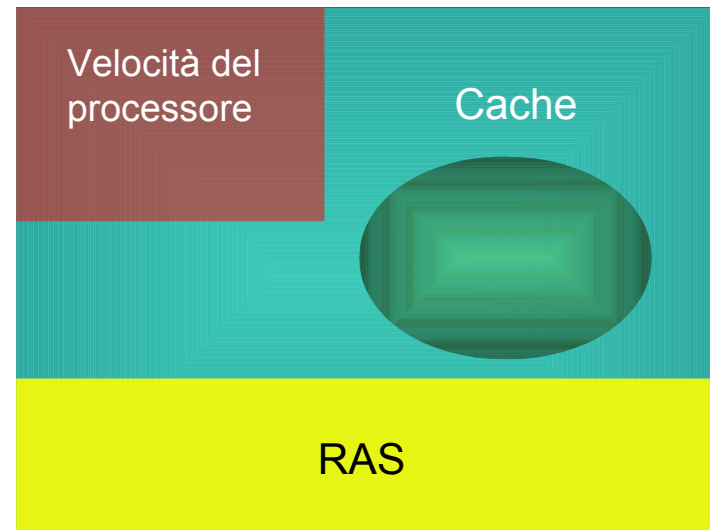
Consolidato

# Le scelte di disegno interessano la performance dei “benchmark” Es.TPC-C



- La cache e' ottimizzata per questo working set
- La “leva” della velocità del processore e' favorita
- RAS non e' misurata
- RAS al minimo non e' penalizzata

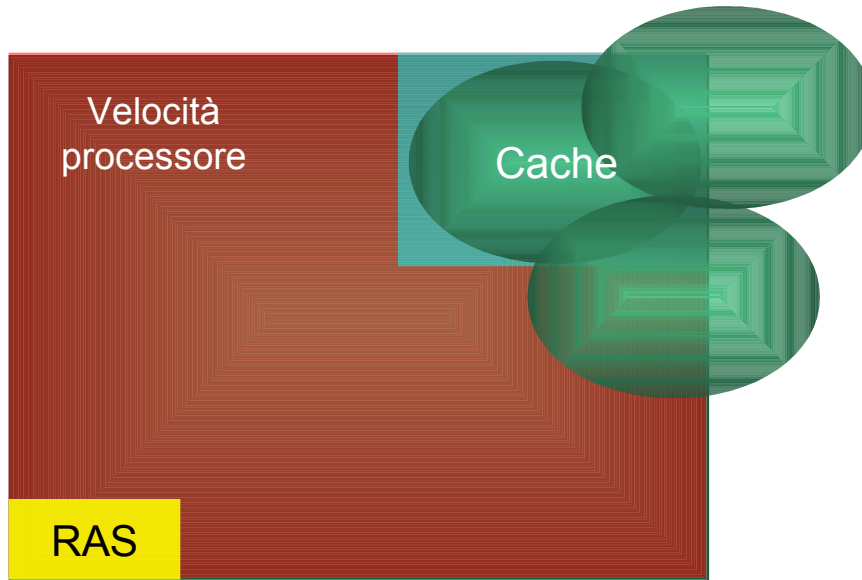
Replicato



- Cache grande e' sfruttata male
- Il compromesso tra Velocità del processore e cache e' penalizzata
- La qualità RAS non e' valorizzata
- RAS e Cache massimizzate sono penalizzate

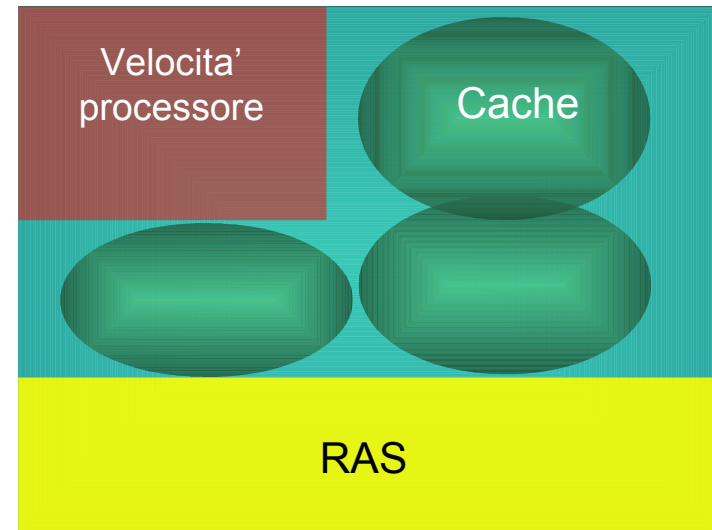
Consolidato

# Le scelte di disegno interessano la performance dei "benchmark" Mixed/WLM/Virtualization



- Working set(s) troppo larghi per la cache
- La velocità del processore massimizzata e' penalizzata
- RAS e' valutata nei carichi misti
- RAS minimizzata e' penalizzata
- Processore veloce e' penalizzato

**Replicato**



- Cache contiene working sets multipli
- La velocità del processore è ottimizzata dalla cache
- La RAS ha valore
- Tutte le caratteristiche del processore lento sono sfruttate

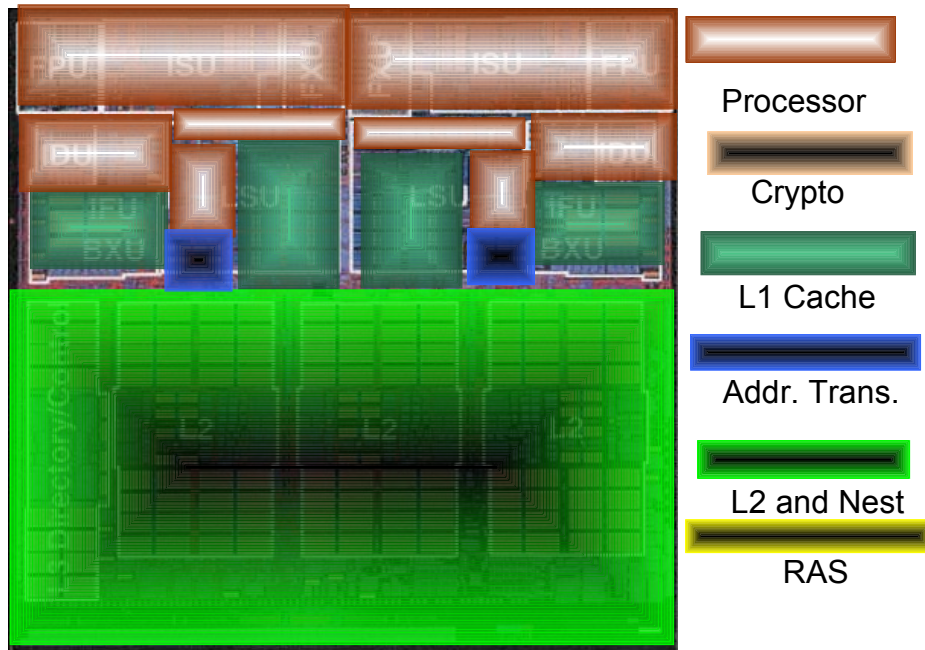
**Consolidato**



# Es. Microfotografie pSeries (RISC) and zSeries (CISC)

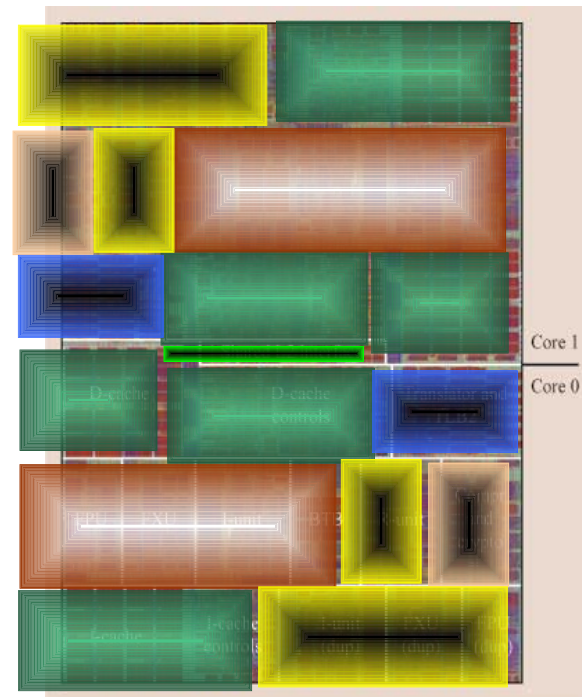
## pSeries

Core 0 | Core 1



**Figure 1**  
POWER4 chip photograph showing the principal functional units in the microprocessor core and in the memory subsystem.

## zSeries



**Figure 8**  
Micrograph of z990 processor chip.

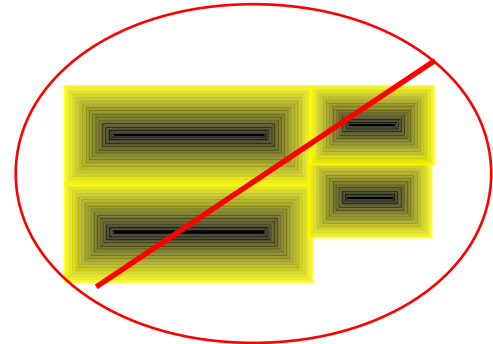
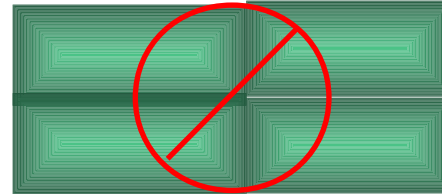
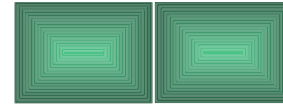
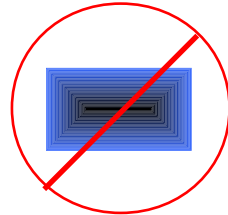
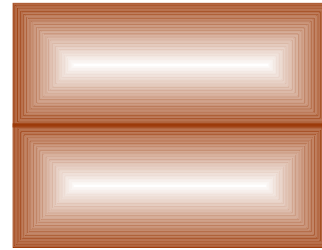
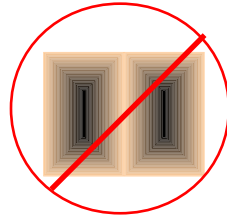
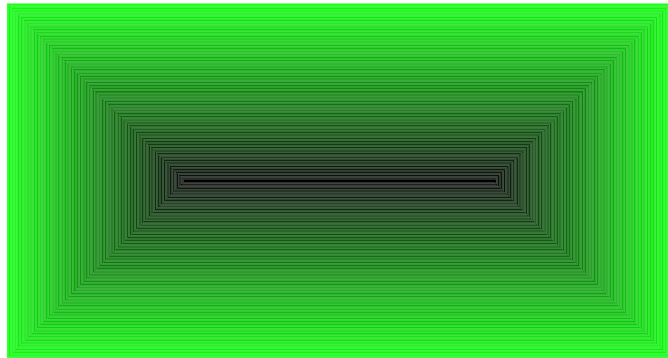
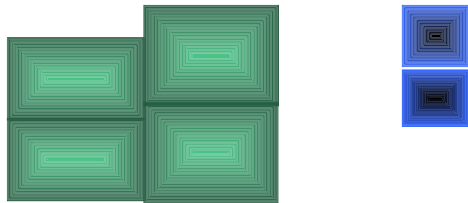
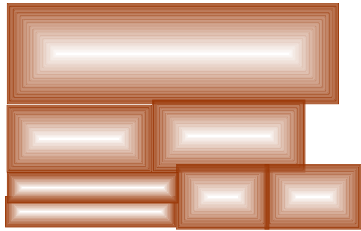
Foto su scale differenti

zSeries L2 e' 32 MB / book

# pSeries(RISC) and zSeries(CISC) sono su scale differenti



Does not contribute to benchmarked performance



Core 0 | Core 1

- Processor
- Crypto
- L1 Cache
- Addr. Trans.
- L2 and Nest
- RAS

## Architettura dei Sistemi Centrali (3 di 3)

