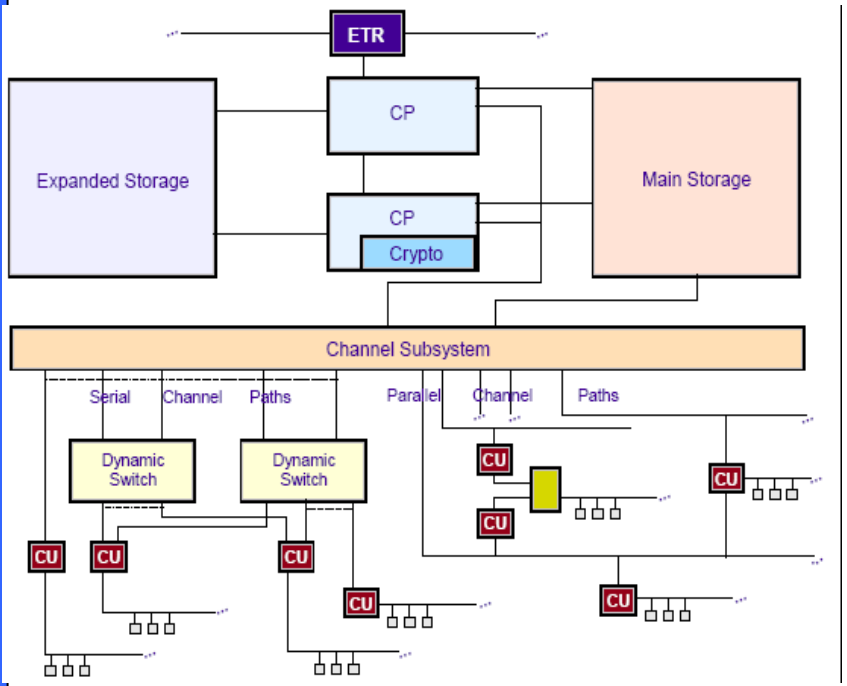
Architettura dei Sistemi Centrali (2 di 3)

z/Architecture

- **Elementi della z/Architecture sono:**
  - **L'organizzazione generale dei Calcolatori**
  - **La Program Status Word (PSW)**
  - **Il modo di indirizzamento della Memoria (Addressing Mode)**
  - **Il Set di Istruzioni**
  - **Le Modalita' di INPUT/OUTPUT e le relative istruzioni.**
  - **L'Organizzazione della Memoria Reale e Virtuale e dei Registri**
  - **Le Istruzioni di Assistenza alla crittografia.**
  - **Le 'Operator facilities'**
  - **Le modalita' di Controllo del Complesso Elaborativo (CEC)**
  - **Le Modalita' di esecuzione dei programmi**
  - **Le modalita' di 'interrupt' (IH)**
  - **Le modalita' di Gestione degli Errori (Machine-Check Handling).**

Definizione Architetturale del Mainframe – la z/Architecture



Tutte le caratteristche funzionali dei Sistemi Centrali IBM (detti z/Systems) sono pubbliche e contenute in un volume denominato '**Principles of Operation**' pubblicato nel 9/2005
Tali caratteristiche vengono indicate col nome di **z/Architecture**.

Elementi essenziali di essa sono:

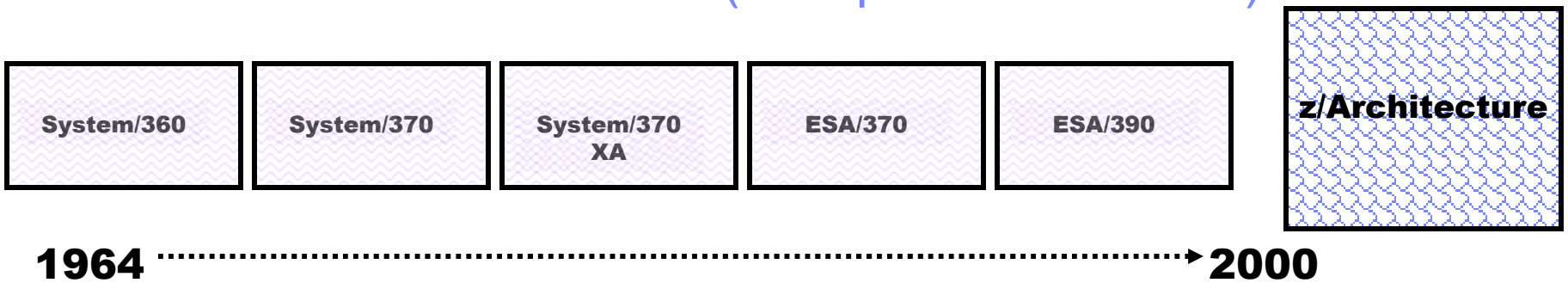
- L'organizzazione dei Sistemi
- La gestione della Memoria
- Le caratteristiche del Sottosistema I/O.
- IL Set di Istruzioni a 64 bit
- La Crittografia integrata

La z/Architecture e' compatibile con e contiene le precedenti Architetture denominate:

- ESA/390
- ESA/370
- S/370 XA
- S/370

Fonte: Z/Architecture Principles of Operation
SA22-7832-04 IBM Corporation September 2005

Elementi della z/Architecture (Compatibilità Binaria)



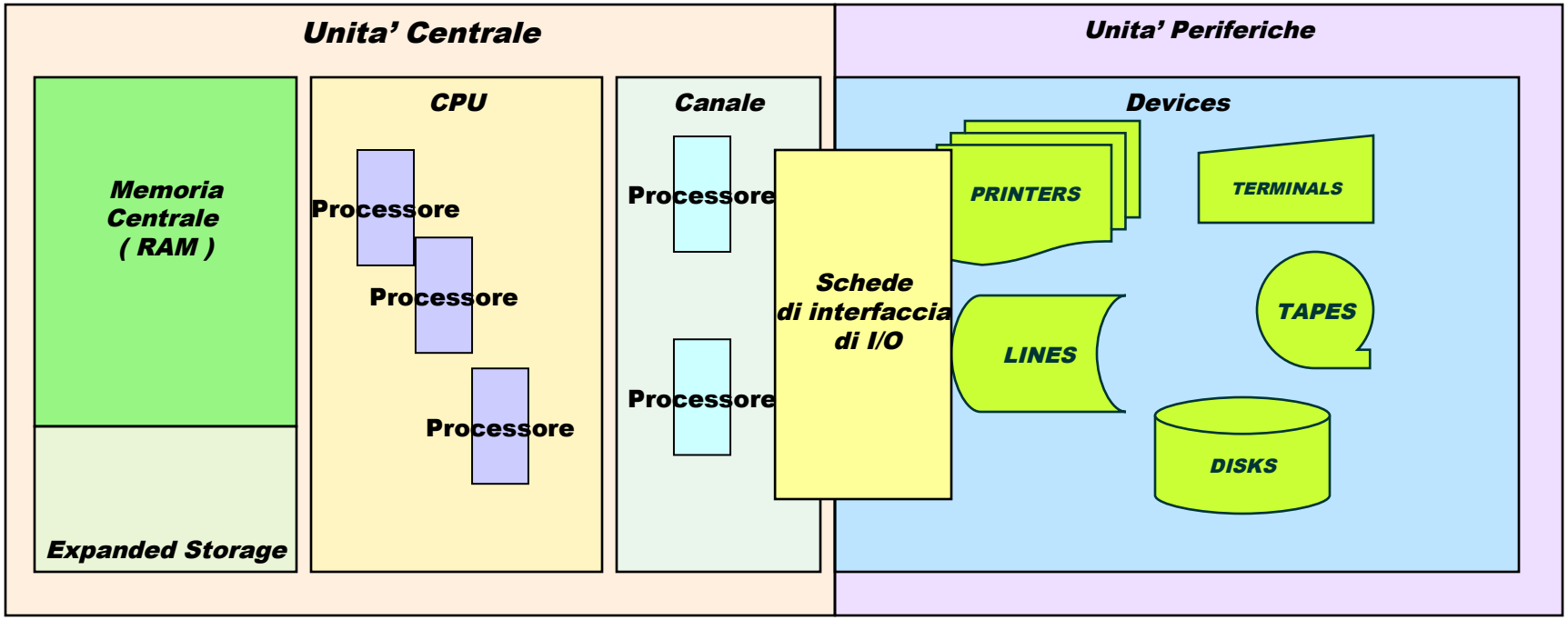
Per **compatibilita' binaria** all'interno di una **famiglia** di calcolatori si intende la abilita' che ha **qualunque programma** scritto secondo le regole della z/Architecture di poter essere eseguito su **qualunque calcolatore** che e' con essa compatibile **senza alcuna modifica** ne al codice sorgente, ne al programma direttamente eseguibile.

Tale caratteristica obbliga tutti **i Calcolatori compatibili** ad essere in grado di eseguire con lo **stesso risultato** tutte le istruzioni definite dall'Architettura, indipendentemente dalla implementazione Tecnologica del Processore**.**

La compatibilità sui mainframe è stata mantenuta nonostante il passaggio a metà degli anni '80 dalla tecnologia bipolare a quella Cmos, che nella versione Ibm portò la densità dei processori dai 5.996 circuiti per chip della prima ai 2 milioni di circuiti per chip della seconda.

Molti utenti dei Sistemi Centrali IBM eseguono con successo oggi, programmi che sono stati compilati nel 1964, senza averli mai modificati o rielaborati .

Architettura dei Mainframe

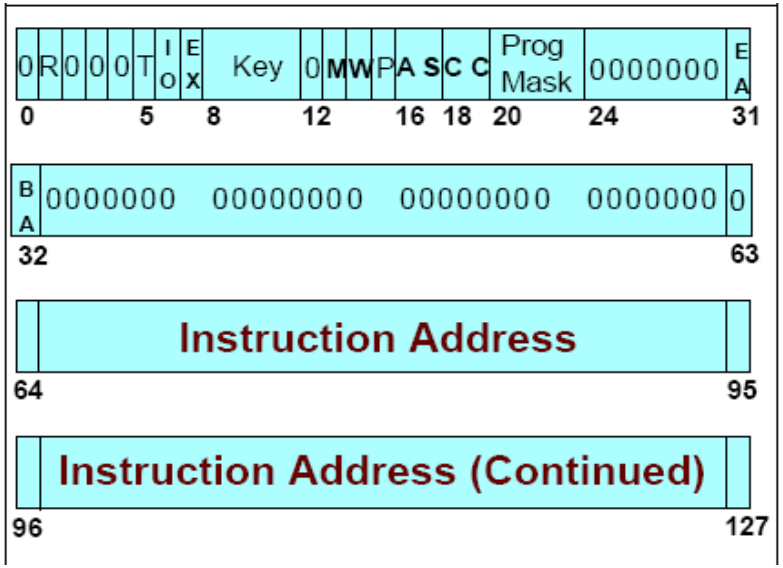


Program Status Word


Elementi della z/Architecture – La Program Status Word (PSW)

La Program Status Word (PSW) e' una struttura binaria che contiene in ogni istante l'indirizzo della istruzione in esecuzione, ed altre informazioni di controllo sullo stato della CPU.

La PSW attiva in ogni istante si chiama 'Current PSW'.



Le CPU della z/Architecture hanno la possibilita' di interrompere il ciclo di istruzioni in esecuzione (programma) e passare subito ad un altro quando ricevono un particolare segnale detto **interruption**.

In questo caso la **Current PSW** viene conservata in una apposita locazione di memoria (registro) e viene caricata una **nuova PSW** relativa al nuovo programma da eseguire.

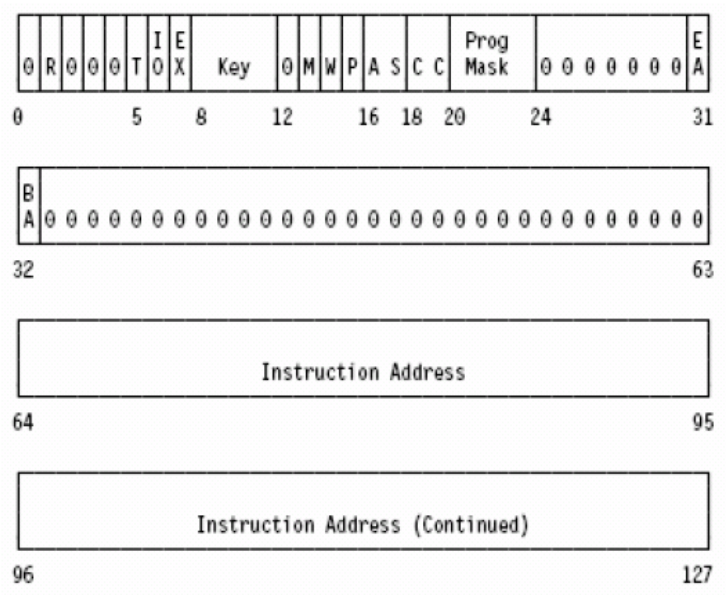
Esistono sei tipi possibili di Interruptions:

1. **External**
2. **I/O**
3. **Machine check**
4. **Program**
5. **Restart**
6. **Supervisor Call**

Elementi di z/architecture – Program Status Word

La PSW contiene le informazioni richieste per l'esecuzione del programma corrente:

- Indirizzo della istruzione
- Addressing Mode
- Condition Code
- Interrupt Mask
- Indicatore di problem/supervisor state (user/kernel mode)



Elementi di z/architecture – Program Status Word

Program Status Word – Fields

- R – enable program event recording (PER)
- T – enable dynamic address translation (virtual storage)
- I – enable I/O interrupts
- E – enable external interrupts
- Key – define storage protection key
- M – enable machine checks
- W – wait state
- P – problem state (0 = supervisor state)
- AS – address space (00 = primary space mode, 01 = secondary space mode, 10 = access register mode, 11 = home space mode)
- CC – condition code

Elementi di z/architecture – Program Status Word

Program Status Word – Fields (continued…)

- **Program mask**
  - Bit 20: enable fixed-point overflow exception
  - Bit 21: enable decimal overflow exception
  - Bit 22: enable hex floating-point exponent underflow exception
  - Bit 23: enable hex floating-point significance exception
- **EA – extended addressing (64-bit addressing mode, BA must also be 1)**
- **BA – basic addressing (31-bit addressing mode, 0 = 24-bit addressing mode)**
- **Instruction address**
  - It is stepped by the length of the current instruction

Problem State (User Mode) e Supervisor State (Kernel Mode)

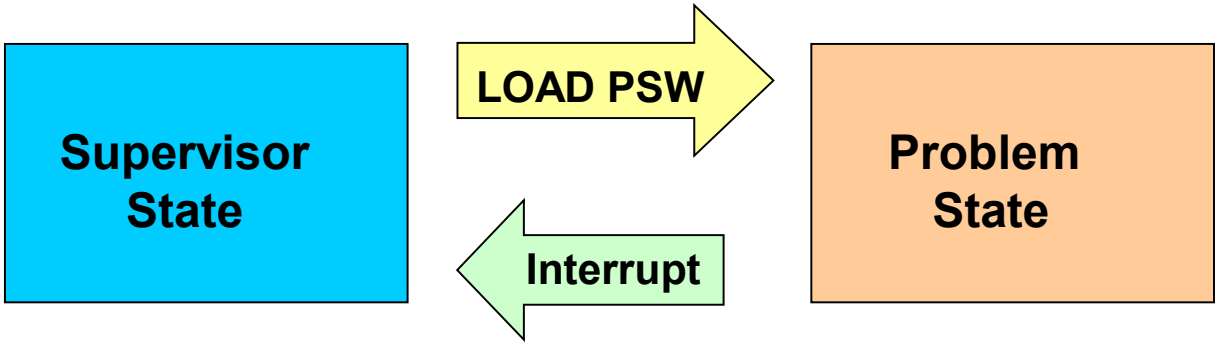
Problem State (User Mode)

- un programma non deve eseguire istruzioni privilegiate, ovvero non puo' accedere a quelle parti dell'architettura che sono vitali per il sistema
- Non accede ai Control Registers, ai timers, alle chiavi di memoria ; accede solo alla parte non critica della PSW.

Supervisor State (Kernel Mode)

- un programma puo' esercitare tutte le potenzialità dell'architettura

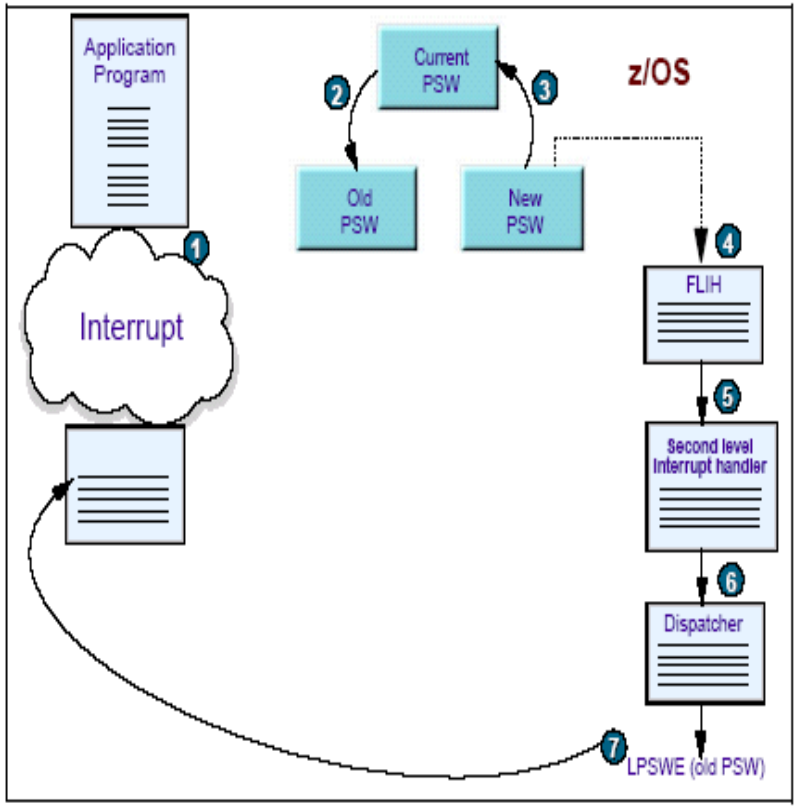
Transizione di stato



return

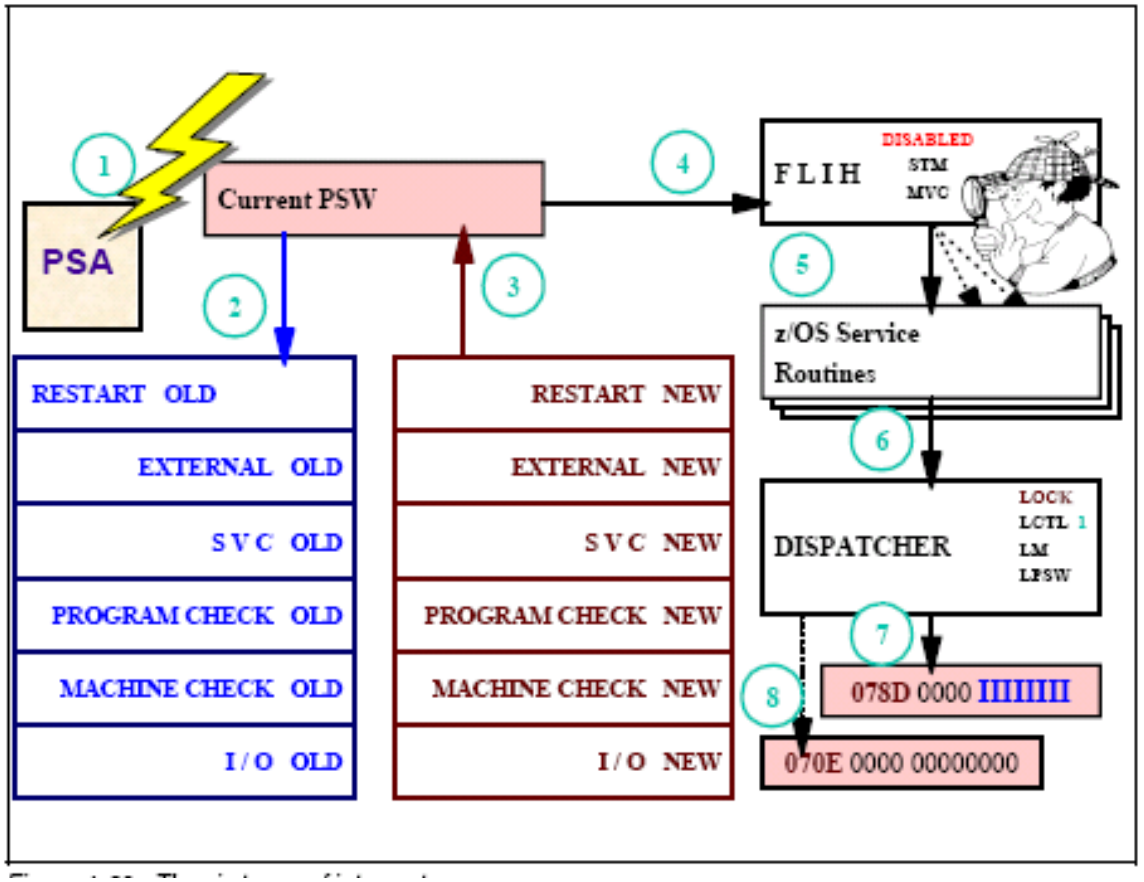
Elementi di z/architecture – Interruzioni (interrupts)

1. Mentre il programma utente sta eseguendo e' viene interrotto da un segnale di interruzione
2. salvando la PSW corrente nella locazione di memoria ad essa dedicata(*)
3. L'HW carica la New PSW il cui indirizzo punta al FLIH (first Level Interrrupt handler)
4. Lo z/OS passa il controllo al FLIH che analizza il tipo di Interrupt e individua l' Handler opportuno SLHI (second level interrupt handler)
5. Viene dato il controllo allo SLHI che decide cosa fare a seconda del tipo di interrupt occorso
6. IL controllo viene poi passato al Dispatcher che ricarica il vecchio contesto (LPSWE old PSW)



* I registri non sono salvati. Tale azione e' delegata al software

Elementi di z/architecture – Interruzioni (interrupts)



Elementi di z/architecture – PSW e locazioni assegnate di memoria Reale

Real addresses	Contents
0x120 – 0x012F	Restart old PSW
0x130 – 0x013F	External old PSW
0x140 – 0x014F	Supervisor-call old PSW
0x150 – 0x015F	Program old PSW
0x160 – 0x016F	Machine-check old PSW
0x170 – 0x017F	I/O old PSW
0x1A0 – 0x01AF	Restart new PSW
0x1B0 – 0x01BF	External new PSW
0x1C0 – 0x01CF	Supervisor-call new PSW
0x1D0 – 0x01DF	Program new PSW
0x1E0 – 0x01EF	Machine-check new PSW
0x1F0 – 0x01FF	I/O new PSW

Elementi di z/architecture – PSW e Interrupt Masking

Alcuni interrupts posso essere inibiti con un meccanismo di “Interrupt Masking” ovvero modificando opportunamente dei bits nella PSW corrente e/o nei Control Register

Possono essere mascherati:

- I/O Interrupts
- External Interrupts
- Machine check Interrupts per guasti non vitali

Non possono essere mascherati

- Program Interrupts (con qualche piccola eccezione)
- Restart Interrups
- Machine Check Interrupt critici
- Supervisor Call Interrupt

return

Elementi di z/architecture – PSW e Condition Code

Condition Code (CC):
Bytes 18 and 19 della PSW rappresentano il "condition code" . Il condition code puo' valere 0, 1, 2, or 3, a seconda del risultato ottenuto eseguendo talune istruzioni.

La maggior parte delle operazioni logiche e matematiche ed altre, danno un "condition code".

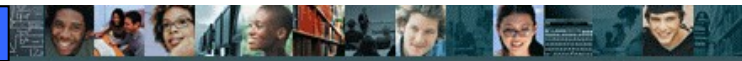
L'Istruzione BRANCH ON CONDITION può specificare un particolare valore di condition-code come criterio per effettuare un salto (branch) .

return

Instruction	Condition Code			
	0	1	2	3
ADD (gen) ADD (BFP) ADD DECIMAL ADD HALFWORD ADD HALFWORD IMMEDIATE	Zero Zero Zero Zero Zero	< zero < zero < zero < zero < zero	> zero > zero > zero > zero > zero	Overflow NaN Overflow Overflow Overflow
ADD LOGICAL ADD LOGICAL WITH CARRY ADD NORMALIZED ADD UNNORMALIZED AND	Zero, no carry Zero, no carry Zero Zero Zero	Not zero, no carry Not zero, no carry < zero < zero Not zero	Zero, carry Zero, carry > zero > zero --	Not zero, carry Not zero, carry -- -- --
CANCEL SUBCHANNEL CHECKSUM CIPHER MESSAGE CIPHER MESSAGE WITH CHAINING CLEAR SUBCHANNEL	Function initiated Checksum complete Normal completion Normal completion Function initiated	-- -- -- -- --	-- -- -- -- --	Not operational CPU-determined completion Partial completion Partial completion Not operational
COMPARE (gen, HFP) COMPARE (BFP) COMPARE AND FORM CODEWORD COMPARE AND SIGNAL COMPARE AND SWAP	Equal Equal Equal Equal Equal	Low Low OCB-0: low OCB-1: high Low Not equal	High High OCB-0: high OCB-1: low High --	-- Unordered -- Unordered --
COMPARE AND SWAP AND PURGE COMPARE DECIMAL COMPARE DOUBLE AND SWAP COMPARE HALFWORD COMPARE HALFWORD IMMEDIATE	Equal Equal Equal Equal Equal	Not equal Low Not equal Low Low	-- High -- High High	-- -- -- -- --

Figure C-1 (Part 1 of 6). Summary of Condition-Code Settings

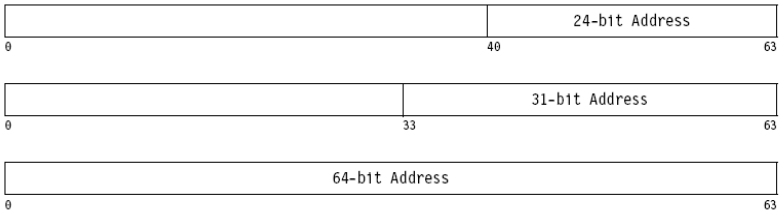
ADDRESSING MODE

Elementi della z/Architecture – Addressing Mode

Sui Sistemi della z/Architecture l'indirizzamento puo' avvenire con tre modalita' (TRIMODAL ADDRESSING):

1. *A 24 Bit. Puo' indirizzare 16 Megabytes (2^{24})di Memoria Reale o Virtuale. Viene mantenuto per compatibilita' con le precedenti architetture.*
2. *A 31 Bit. Puo' indirizzare 2Gigabytes (2^{31})di Memoria Reale o Virtuale . Viene mantenuto per compatibilta' con le precedenti architetture.*
3. *A 64 Bit . Metodo Standard puo' indirizzare 16 ExaBytes (2^{64}) .*



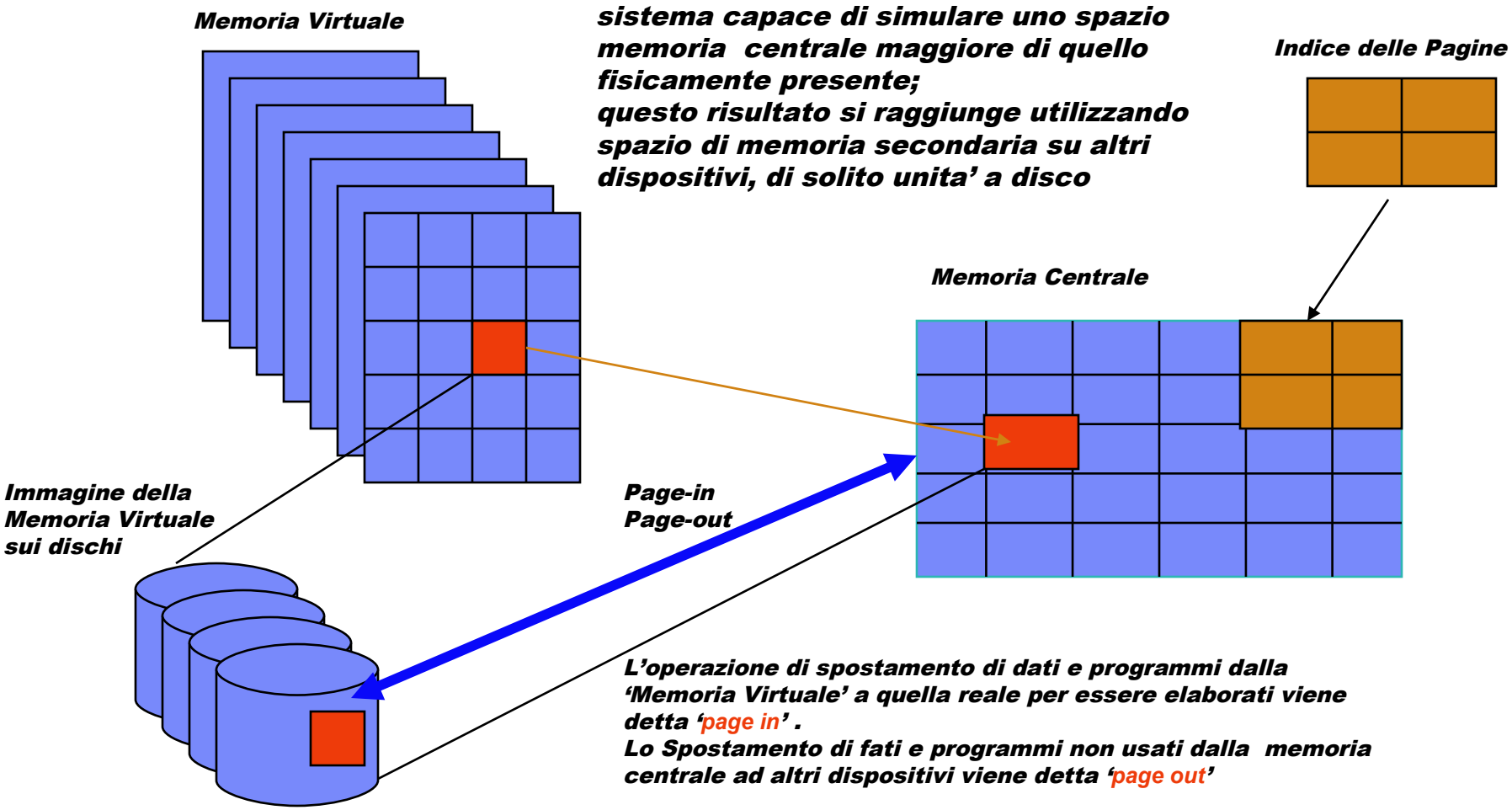
Return

Fonte: Z/Architecture Principles of Operation SA22-7832-04 IBM Corporation September 2005

MEMORIA


Elementi della z/Architecture –Memoria Virtuale

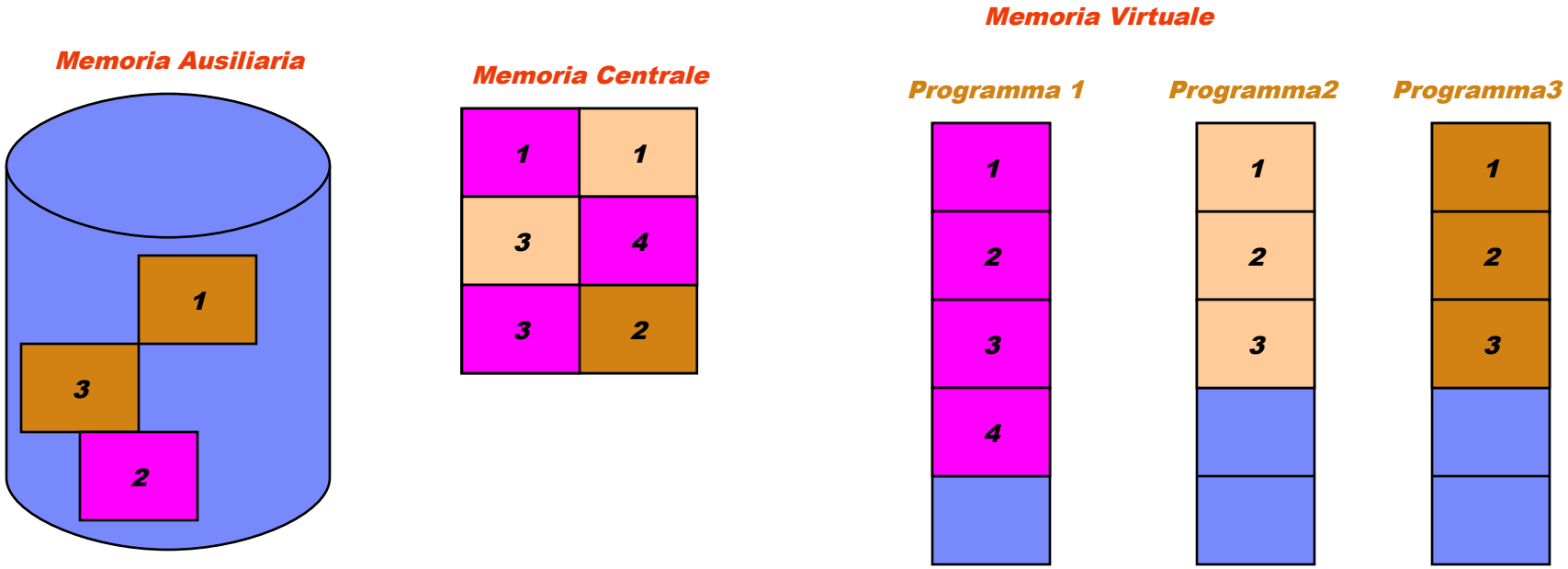
Memoria Virtuale è una architettura di sistema capace di simulare uno spazio memoria centrale maggiore di quello fisicamente presente;
questo risultato si raggiunge utilizzando spazio di memoria secondaria su altri dispositivi, di solito unita' a disco



L'operazione di spostamento di dati e programmi dalla 'Memoria Virtuale' a quella reale per essere elaborati viene detta 'page in' .
Lo Spostamento di fati e programmi non usati dalla memoria centrale ad altri dispositivi viene detta 'page out'

Elementi della z/Architecture – il dynamic address translator (DAT)

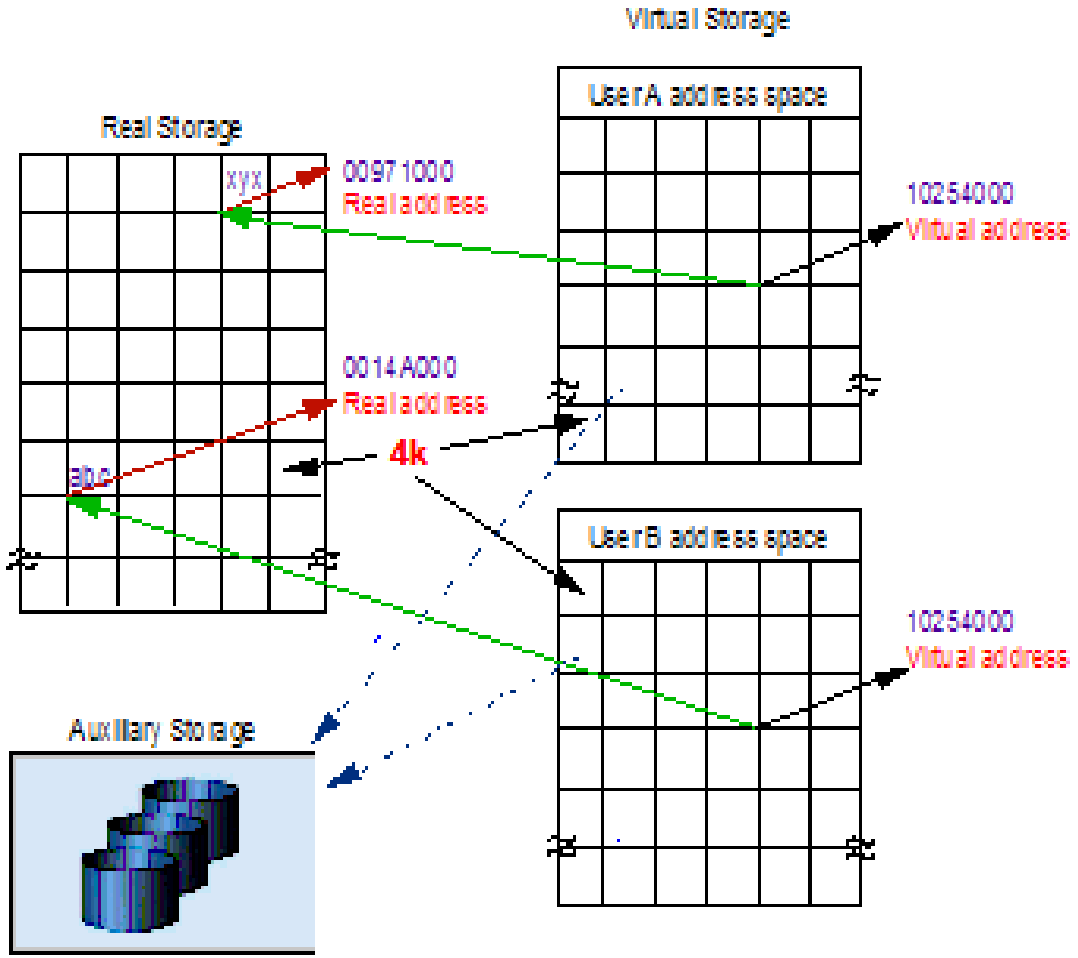
IL Dynamic Address transaltor (DAT) ha la capacita' di interrompere l'esecuzione di programmi per spostare i contenuti relativi dalla memoria centrale su una memoria ausiliaria (dischi) ed in un secondo momento restituire dati e programma alla memoria centrale ponendoli in una differente locazione.
Tale operazione viene definita 'Paginazione ' e rappresenta il passaggio dalla Memoria Virtuale a quella Reale.
Le operazioni del DAT sono assolutamente trasparenti al programma.



Dynamic Address translation (DAT)

Dynamic address translation (DAT) e' il processo che ,durante un riferimento in memoria, trasla un indirizzo virtuale nel corrispondente indirizzo reale.

Se l'indirizzo non e' in memoria si verifica un interrupt di "page fault" e il sistema operativo prende la pagina in memoria ausiliaria



Dynamic Address translation (DAT)

Dynamic Address Translation (DAT)

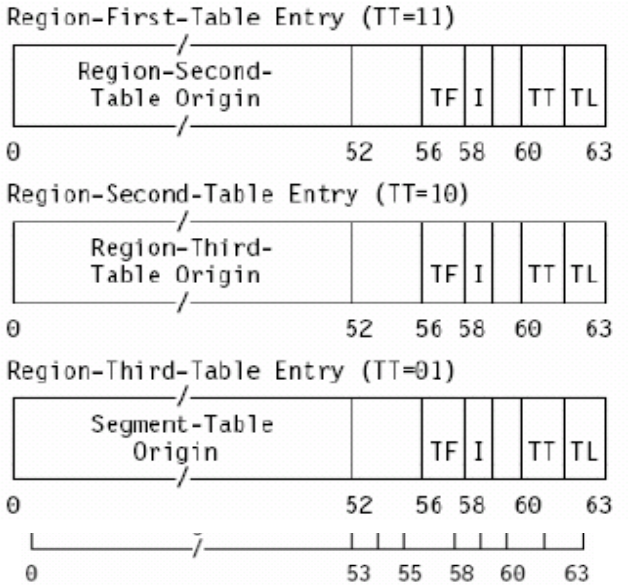
•e' il processo che ,durante un riferimento in memoria, trasla un indirizzo virtuale nel corrispondente indirizzo reale.

.

•DAT e' implementato via hardware e software attraverso l'uso di page tables, segment tables, region tables and translation lookaside buffers.

z/Architecture DAT Table Entries

- **The region-table entries all have the same format:**



- **And finally here are the page-table entries:**

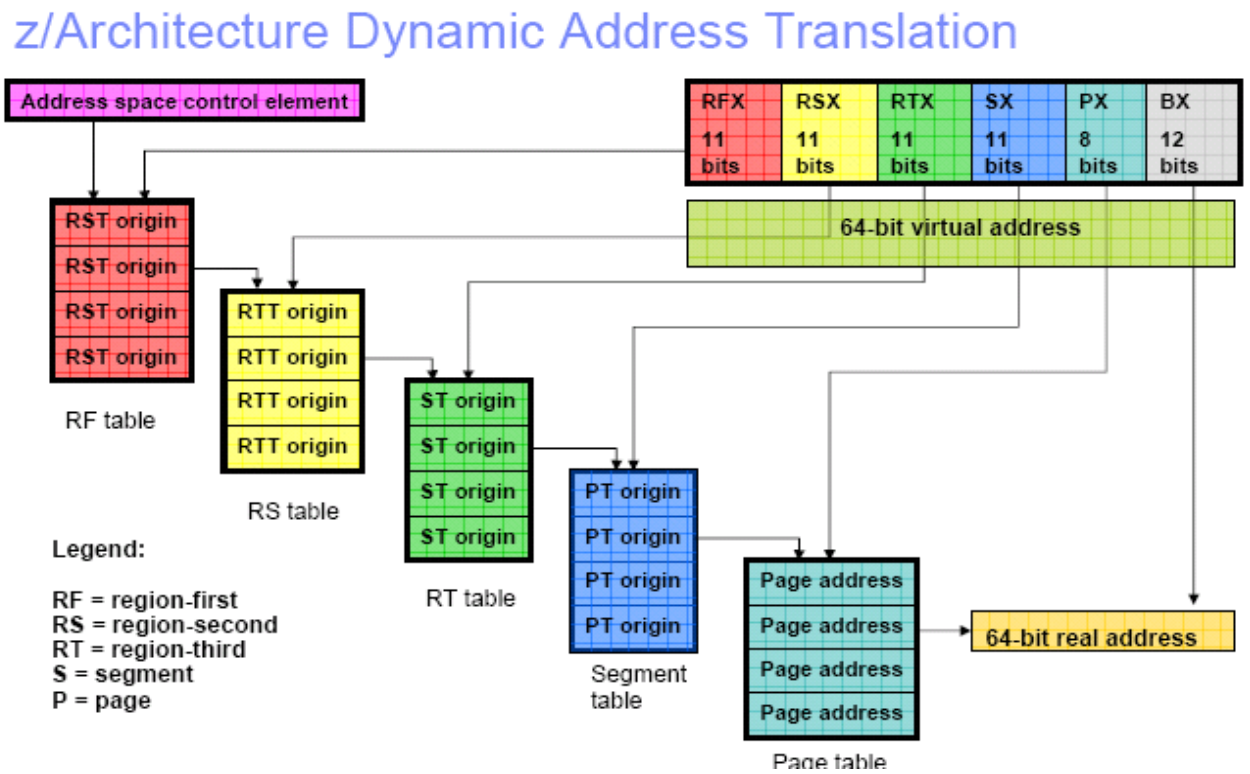
- DAT permette di condividere lo stesso codice/dato in sola lettura da differenti address space

- Infatti indirizzi virtuali in address space differenti possono essere tradotti dallo stesso frame di memoria reale

Se l'indirizzo e' in memoria il processo di DAT e' accelerato attraverso l'uso di un TLB (translation lookaside buffer) ovvero di una cache di CPU.

DAT - Tabelle

I puntatori a queste tabelle sono mantenuti nei control registers (CR1, CR7 e CR13) o sono negli Access Registers



Memoria Virtuale (Invalid Bit)

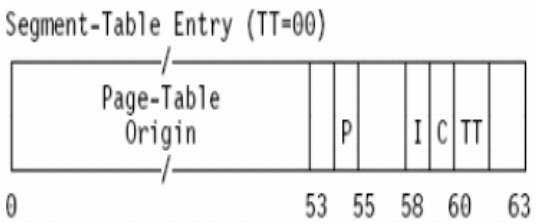
Invalid Bit

Ad ogni page table entry e' associato un "invalid bit" il cui significato indica:

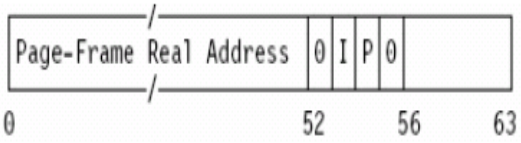
1 = pagina non in memoria
0 = pagina in memoria

Inizialmente il bit ha valore 1 in tutte le entrate.
Durante la traslazione se l'invalid bit e' a 1 si ha un "interrupt" di "Page fault"

- The segment-table entries look as follows:



- And finally here are the page-table entries:



L'istruzione e' annullata e la old PSW che verrà caricata a causa del program interrupt puntera' all'istruzione in corrente e non a quella successiva

Architettura del Mainframe - Memoria

Memoria

- Rappresentazione **BIG ENDIAN** (come IBM System p e diversamente da Intel)
- Gli indirizzi di memoria centrale sono **Byte Address**

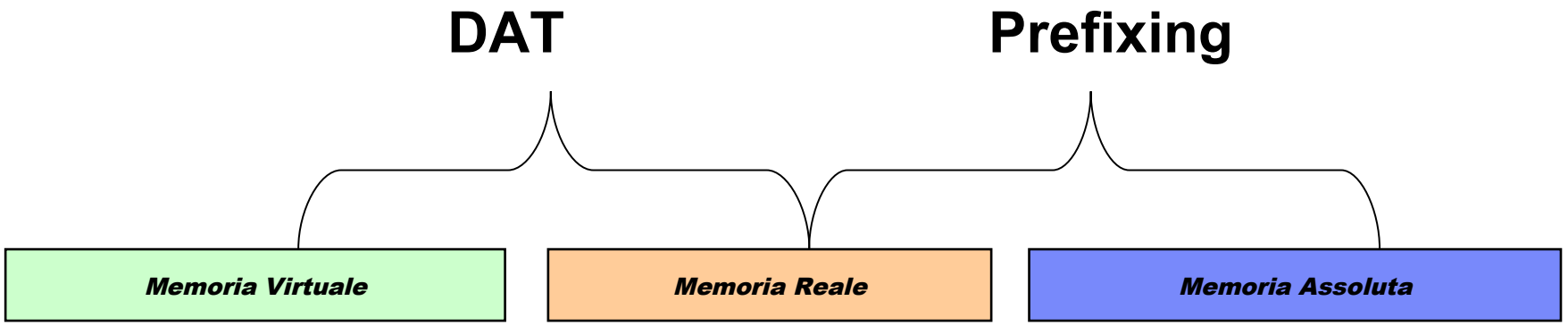
Tipi di Indirizzi sono:

- **Virtuale** traslato in Reale dal Dynamic Address Translation
- **Reale** traslato in assoluto usando un registro di prefixing
- **Assoluto** in Memoria fisica dopo aver applicato il registro di prefix
- **Logica** L'indirizzo visto dal programma (virtuale o reale)

Elementi della z/Architecture – MEMORIA

Al fine di indirizzare la memoria esistono tre tipi di indirizzi:

1. *Indirizzo Assoluto, ovvero la posizione esatta del dato/programma all'interno della memoria Centrale.*
2. *Indirizzo Reale : e' un indirizzo di memoria Centrale ottenuto dalla traslazione dell'indirizzo virtuale con il DAT.*
3. *Indirizzo Virtuale : Indica una posizione nella memoria Virtuale*



Multi Processing e Prefixing


Multiprocessing e prefixing

- **multiprocessing**, permette a più processori di condividere la memoria principale sotto il controllo di un unico sistema operativo.

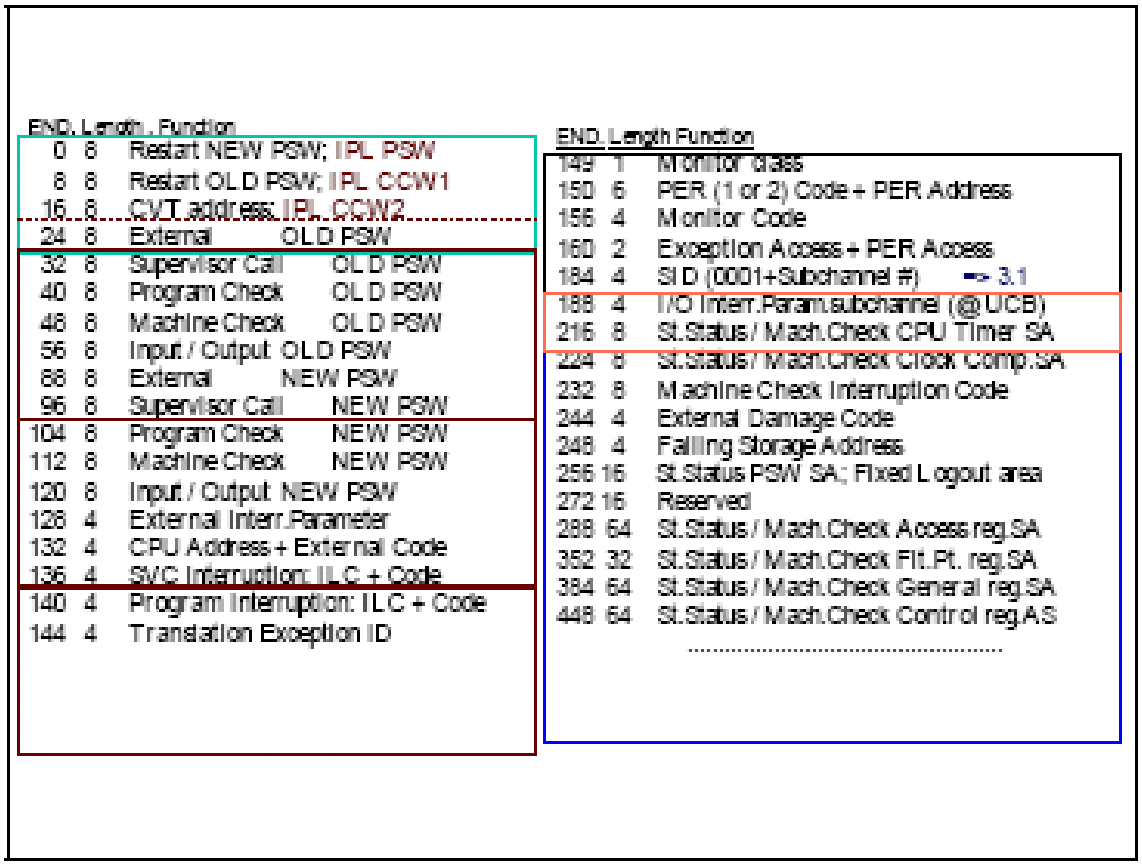
- **Il prefixing** evita conflitti nell'uso delle pagine (frame) di memoria reale iniziale (indirizzo 0- 8kB) chiamata "prefix area" . Infatti l'architettura mappa sui I primi 8kB di memoria reale le informazioni di controllo piu' importanti.

.

  - Il prefixing utilizza dei registri chiamati "prefix register" che puntano alle prefix area legata a ciascun processore.
  - Ogni riferimento a tale page frame 0 sarà quindi indirizzata a quest'area.
  - Al contrario ogni riferimento di un processore a un indirizzo reale ove il pageframe address coincide con il contenuto del prefix register del processore sara' diretto all'area di memoria reale page frame 0
  - Un indirizzo reale a cui e' stato applicato il prefixing e' un indirizzo assoluto

Prefixed Save Area

Prefixed save area (PSA)



END.	Length	Function
0	8	Restart NEW PSW; IPL PSW
8	8	Restart OLD PSW; IPL CCW1
16	8	CVT address; IPL CCW2
24	8	External OLD PSW
32	8	Supervisor Call OLD PSW
40	8	Program Check OLD PSW
48	8	Machine Check OLD PSW
56	8	Input / Output OLD PSW
88	8	External NEW PSW
96	8	Supervisor Call NEW PSW
104	8	Program Check NEW PSW
112	8	Machine Check NEW PSW
120	8	Input / Output NEW PSW
128	4	External Interr.Parameter
132	4	CPU Address + External Code
136	4	SVC Interruption: ILC + Code
140	4	Program Interruption: ILC + Code
144	4	Translation Exception ID

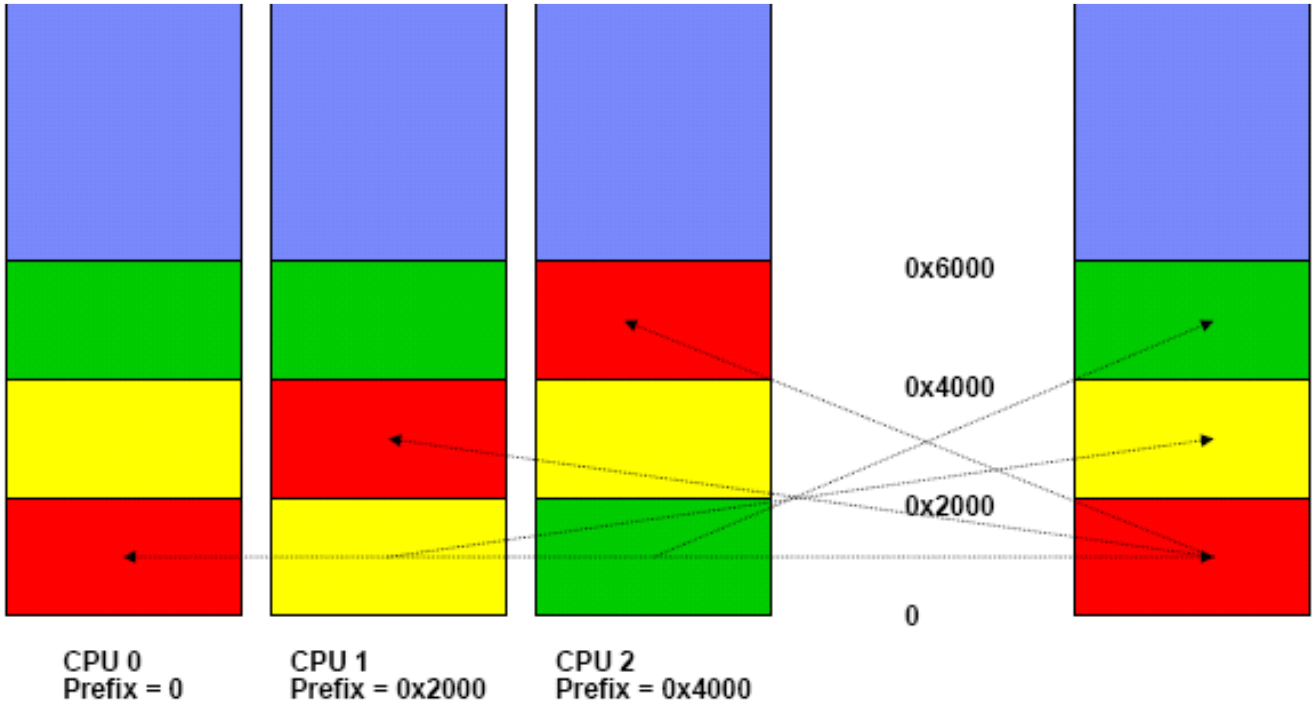
END.	Length	Function
149	1	Monitor class
150	6	PER (1 or 2) Code + PER Address
156	4	Monitor Code
160	2	Exception Access + PER Access
184	4	SID (0001+Subchannel #) → 3.1
188	4	I/O Interr.Param.subchannel (@ UCB)
216	8	St.Status / Mach.Check CPU Timer SA
224	8	St.Status / Mach.Check Clock Comp.SA
232	8	Machine Check Interruption Code
244	4	External Damage Code
248	4	Failing Storage Address
256	16	St.Status PSW SA; Fixed Logout area
272	16	Reserved
288	64	St.Status / Mach.Check Access reg.SA
352	32	St.Status / Mach.Check Flt.Pt. reg.SA
384	64	St.Status / Mach.Check General reg.SA
448	64	St.Status / Mach.Check Control reg.AS
		.....

Figure 1-15 Prefixed save area (PSA)

Prefixing

Mapping of Real to Absolute Storage (Prefixing)

Mappatura di Memoria Reale verso la Memoria Assoluta (Prefixing)



Le CPU e il channel subsystem usano gli stessi indirizzi assoluti. La memoria centrale è assegnata di solito con indirizzi assoluti a partire da 0 e gli indirizzi sono sempre assegnati in blocchi di 4k byte completi

La protezione degli indirizzi bassi di memoria serve a evitare la distruzione delle informazioni di main storage usata dalla CPU durante il processo di interrupt

Prefixing fornisce la capacità di assegnare l'intervallo di indirizzi reali 0-8k a un blocco differente in memoria assoluta per ogni CPU, permettendo a più di un CPU di condividere la memoria centrale per operare concorrentemente con un minima interferenza , specialmente nel processo di interrupt

Meccanismi di protezione della memoria

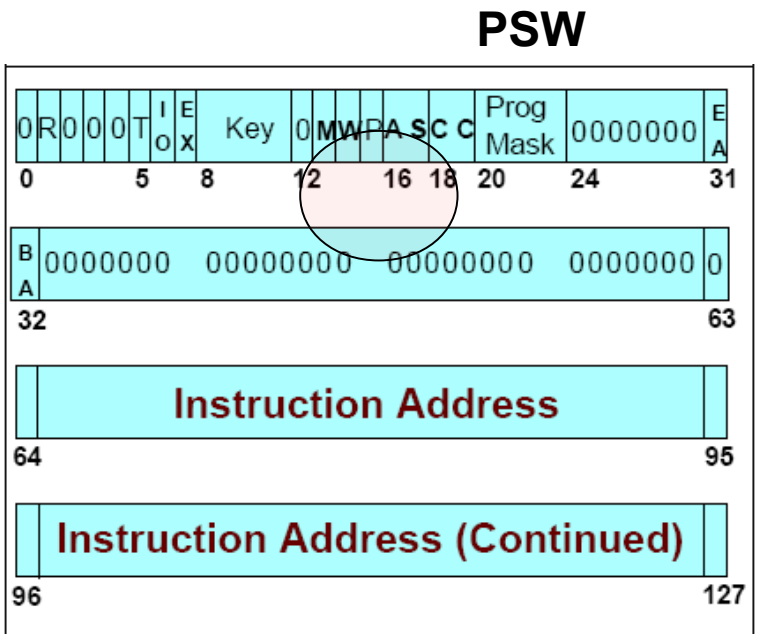
Protezione controllata da chiavi

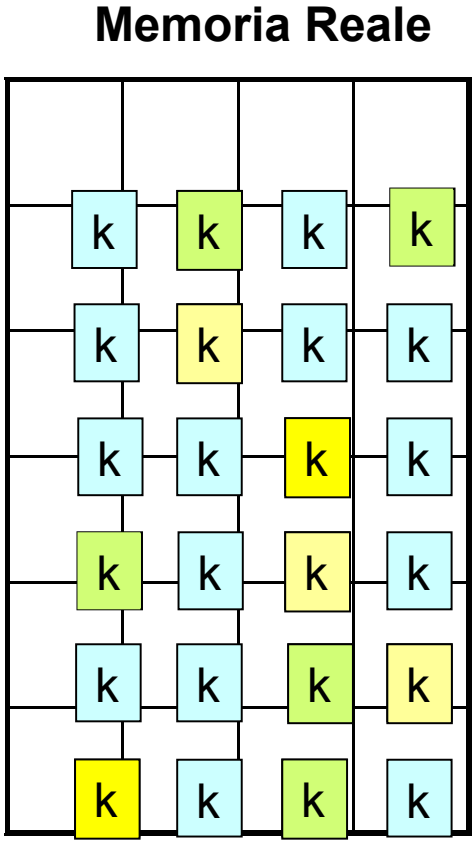
Protezione degli indirizzi di memoria “iniziale”

Protezione delle pagine

Protezione controllata via lista di accesso

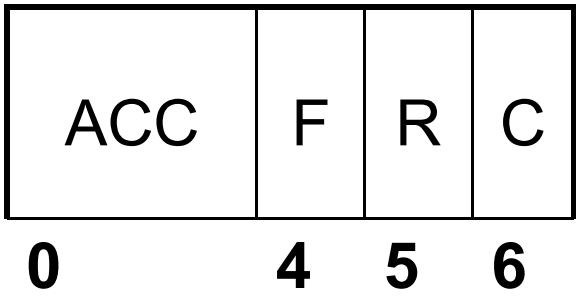
Storage Keys





Chiavi di memoria (Storage Keys)

Una chiave di memoria è associata con ogni blocco di di memoria reale di 4 Kbyte



- ACC= Access Control List
- F= fetch protection bit
- R= Reference Bit
- C= Change Bit

Condizione		E' permesso l'accesso in memoria?	
Fetch-protect bit	chiave in PSW (*) = chiave in memoria ?	Fetch	Store
0	**Si**	**si**	**si**
0	**No**	**si**	**no**
1	**Si**	**si**	**si**
1	**No**	**no**	**no**

(*) Se la chiave in PSW e uguale a 0 il programma e' autorizzato ad accedere in memoria sia in Fetch che in Store

Protezione di memoria “iniziale” e protezione di pagina

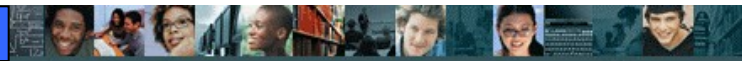
Protezione di Memoria Iniziale

- Previene la memorizzazione in indirizzi da 0-511 e 4096-4607
- Usata come precauzione addizionale per prevenire la corruzione delle informazioni vitali di sistema (ad es new e old PSW)
- Controllata dal Control Register 0 bit 35

Protezione di pagina

- Previene l’alterazione non voluta di una pagina
- Non previene il caricamento di quella pagina (fetch protection)
- Si ottiene con il fetch-protection bit in page table entry

REGISTRI

Elementi della z/Architecture : Principali Registri della z/Architecture

I **Registri** sono strutture della CPU designate a contenere informazioni di controllo e servizio ovvero i dati da elaborare

I Registri si dividono in:

•**General Registers** : (16 a 64 bit) – Sono i registri di base per il funzionamento della CPU e per l'esecuzione delle operazioni elementari (Accumulatori, Program Counter,etc...)

•**Floating Point Registers**: (16 a 32 o 64 bit)- Sono usati per le operazioni in virgola mobile a singola o doppia precisione.

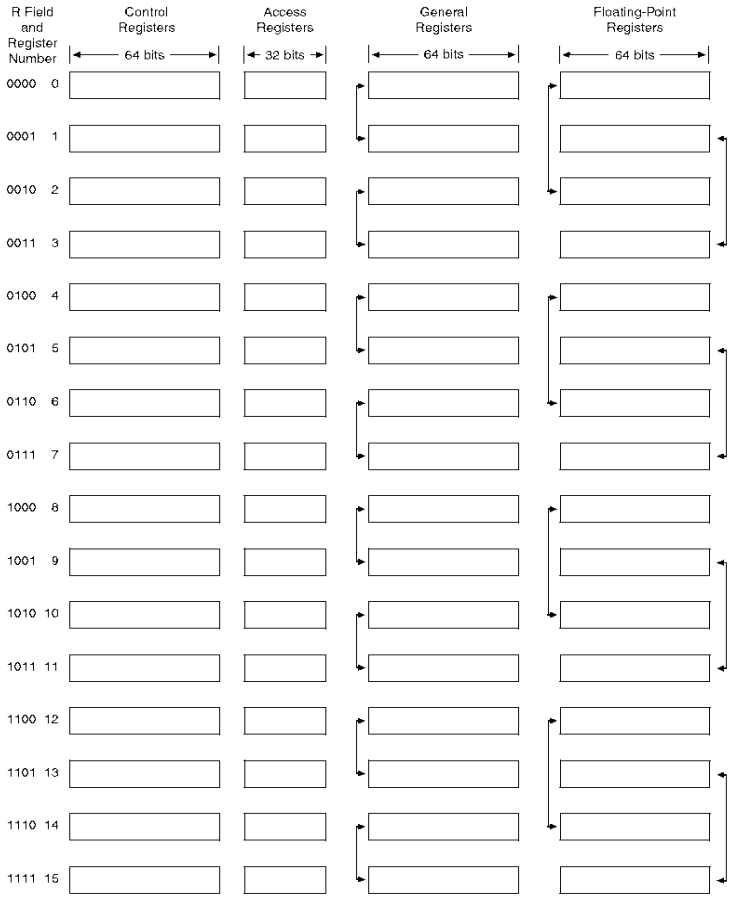
•**Floating point Control register** : un registro a 32 bit che contiene informazioni di controllo per la gestione delle operazioni in virgola mobile.

•**Control Registers:(**16 a 64 bit) Sono usati dalla CPU solo per funzioni di controllo e registrazione

•**Access Registers**: (16 a 32 bit) Servono a controllare la complessa struttura della memoria virtuale e sono utilizzati dal DAT

•**Prefix Register** (32 Bit) servono per definire gli indirizzi assoluti di memoria (prefixing)

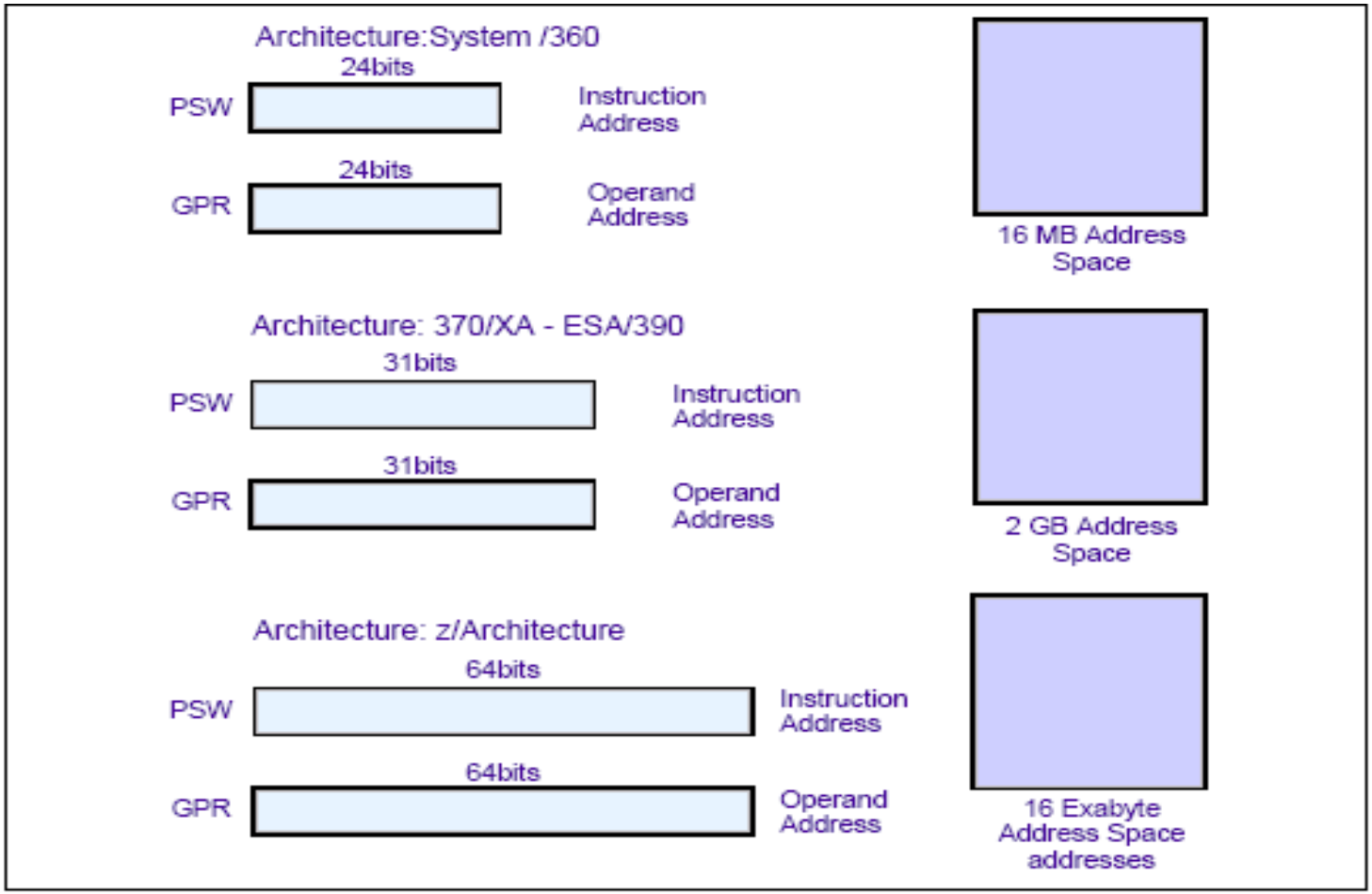
Fonte: Z/Architecture Principles of Operation SA22-7832-04 IBM Corporation September 2005



Note: The arrows indicate that the two registers may be coupled as a double-register pair, designated by specifying the lower-numbered register in the R field. For example, the floating-point register pair 13 and 15 is designated by 1101 binary in the R field.

Figure 2-2. Control, Access, General, and Floating-Point Registers

Elementi di z/Architecture – PSW ,Registri, Memoria

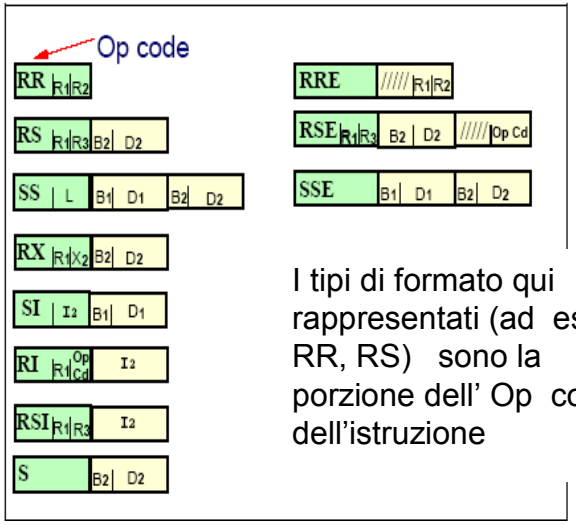


Elementi della z/Architecture – il Set di Istruzioni (Instruction set)

- 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**)



I tipi di formato qui rappresentati (ad es. RR, RS) sono la porzione dell' Op code dell'istruzione

Ogni istruzione e' costituita da due parti:

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

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.

Architettura z – Set di Istruzioni

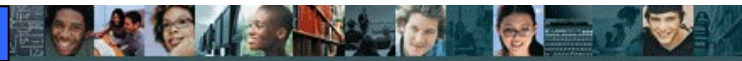
S/360 (Novembre 1970) aveva 143 istruzioni oggi 2007 System z9 ha 689 istruzioni

Le istruzioni si possono suddividere in

- General instruction
- Decimal Instrction
- Floating Point Instruction
  - General
  - Hexadecimal
  - Binary
  - Decimal
- Control Instruction (*)
- I/O Instruction (*)

** istruzioni privilegiate che agiscono sulle risorse vitali di sistema*

SET DI 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
  - .....

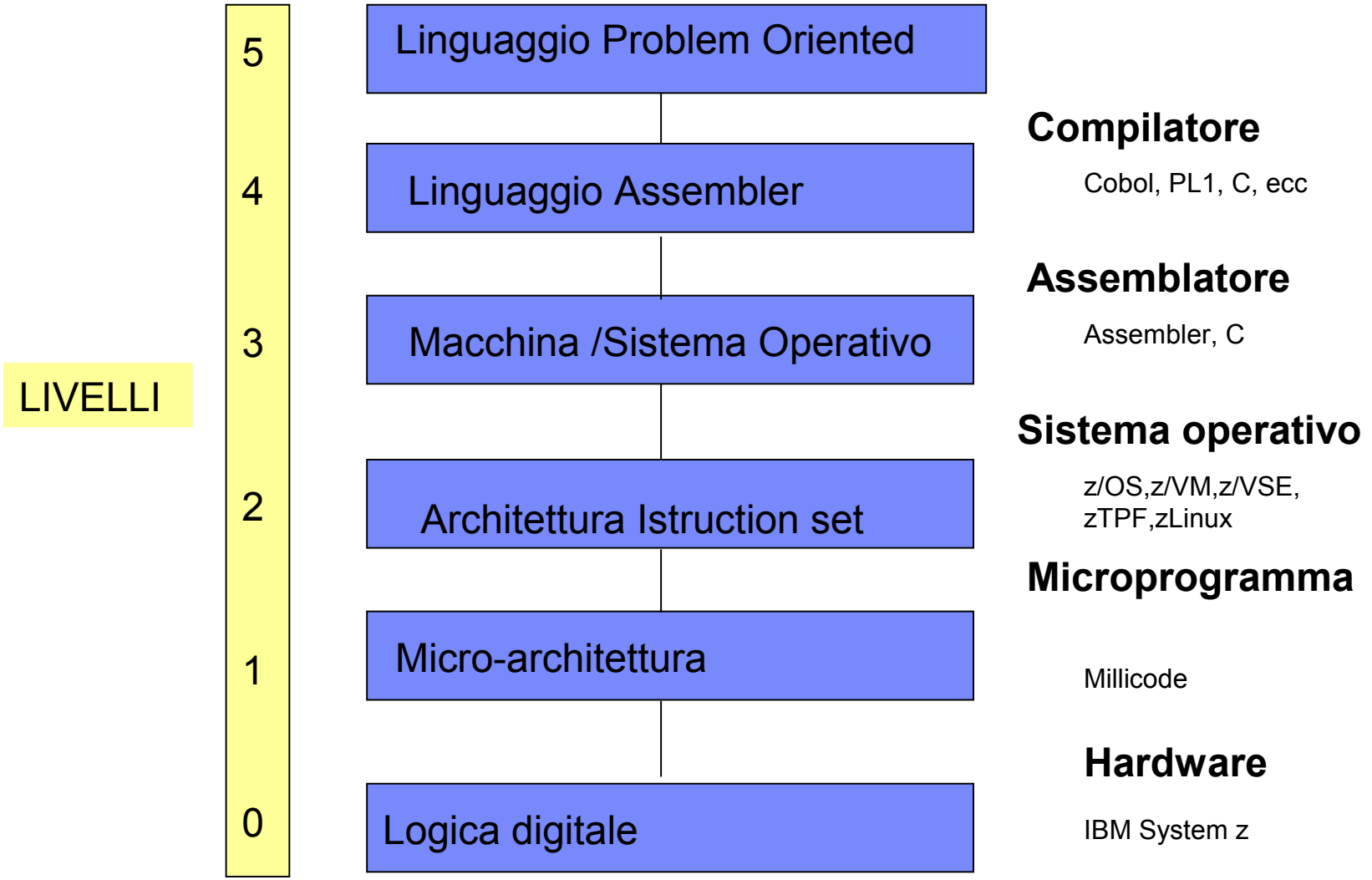
- Control Instructions:
  - COMPARE AND SWAP
  - DIAGNOSE
  - MOVE PAGE
  - LOAD PSW
  - SET CLOCK
  - SIGNAL PROCESSOR
  - PAGE IN
  - PAGE OUT
  - STORE CPU ID
  - .....
- Hexadecimal FP Istructions:
  - ADD NORMALIZED
  - CONVERT TO FIXED
  - MULTIPLY
  - SQUARE ROOT
  - LOAD AND TEST
  - .....
- Binary FP Instructions
  - ADD
  - COMPARE
  - LOAD FPC
  - MULTIPLY AND ADD
  - .....

Esecuzione dele Istruzioni

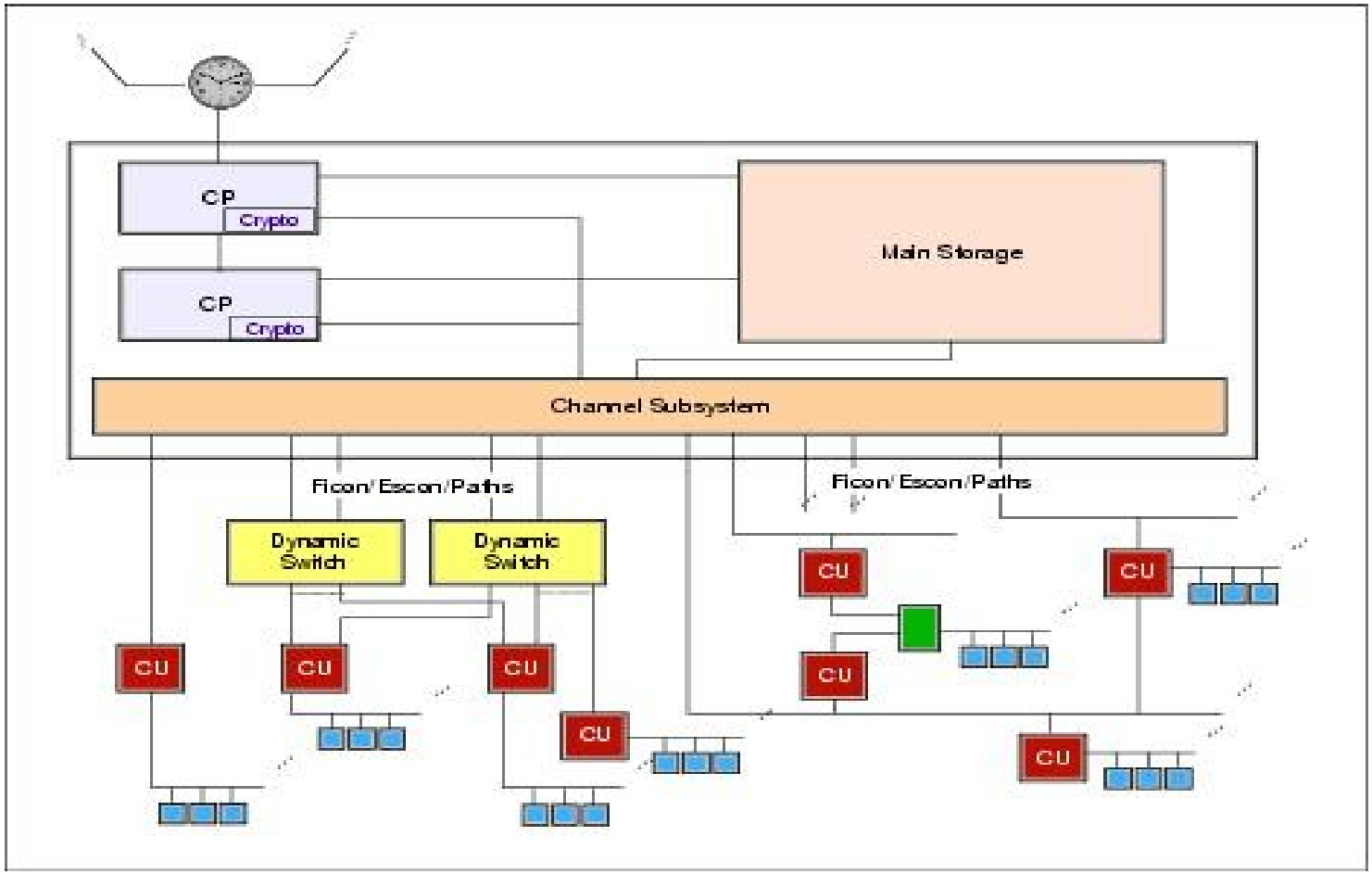
Le istruzioni vengono eseguite via

- **Hardware**
  - **Istruzioni semplici**
  - **Alta performance**
- **Millicode**
  - **Istruzioni complesse ,**
  - **Minore performance,**
  - **Funzioni precedentemente eseguite dal processore**

Architettura z Livelli di interpretazione Programma



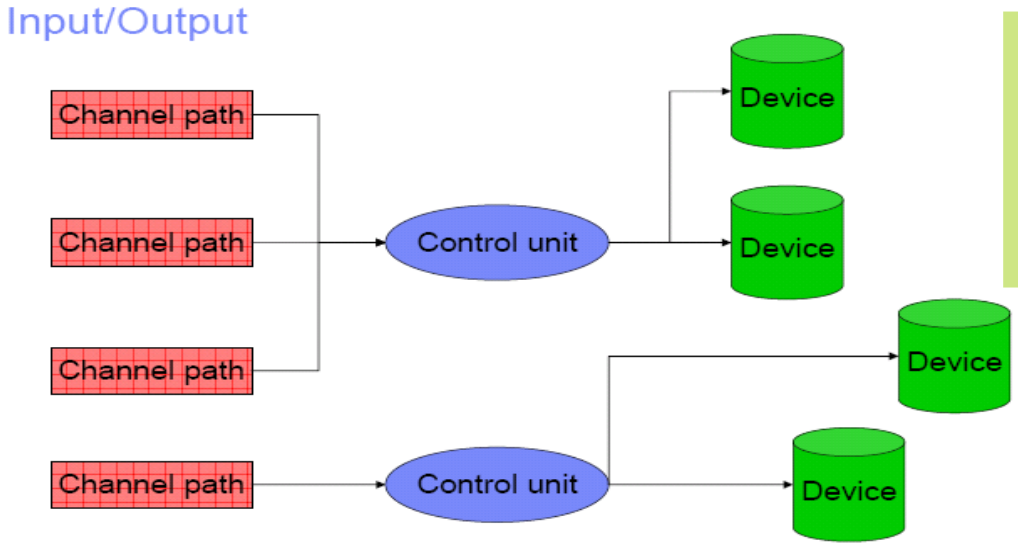
Architettura generale del mainframe



Architettura generale del Mainframe - Input/Output

Un Channel Path e' un processore separato che controlla il trasferimento dei dati tra memoria centrale e Dispositivi di I/O (devices)

- Il dato che e' letto/scritto su un mezzo esterno (Nastro Disco ,,,)
- Oltre ai dati si trasferiscono Informazioni di controllo



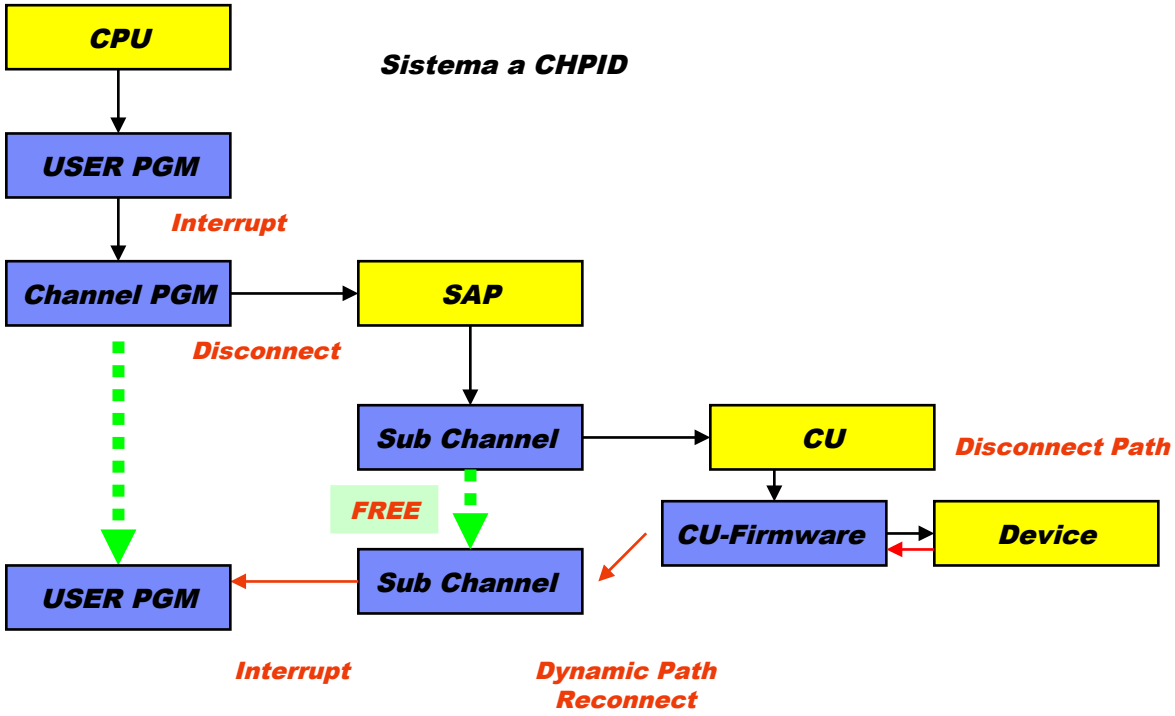
Un device è guidato da una Control Unit.

A ciascun device e' assegnato un numero dall'Amministratore del Sistema

Una Control Unit interpreta in dettaglio i comandi specifici per il device ad essa connesso es.

- Posizionamento della testina di lettura di un disco
- Il riavvolgimento di un nastro
- Il trascinamento di un pagina su stampante

Gestione I/O interrupt



L'Operazione di I/O viene gestita da diverse CPU indipendenti

La potenza dei Processori - Definizioni

- **Ciclo Base (Nanosecondi)** = Tempo medio per eseguire una istruzione Elementare di Macchina ovvero tempo minimo per passare tra due stati stabili del processore.
- **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.

Architettura dei Sistemi Centrali (2 di 3)