

NOTE SULLA ARCHITETTURA XA E SULLA VECTOR FACILITY
DEL SISTEMA IBM 3090.

GIUSEPPE SEVERINO

Rapporto interno C87-20

PISA - Settembre 1987
By CNUCE CNR

Queste brevi note non hanno la pretesa di essere un corso, tanto meno di essere esaustive. Si vuole solo indicare le principali modifiche apportate dall'architettura estesa (XA) alla precedente architettura 370. Per piu' approfonditi studi rimandiamo alle pubblicazioni IBM.

**ARCHITETTURA XA
CENNI**

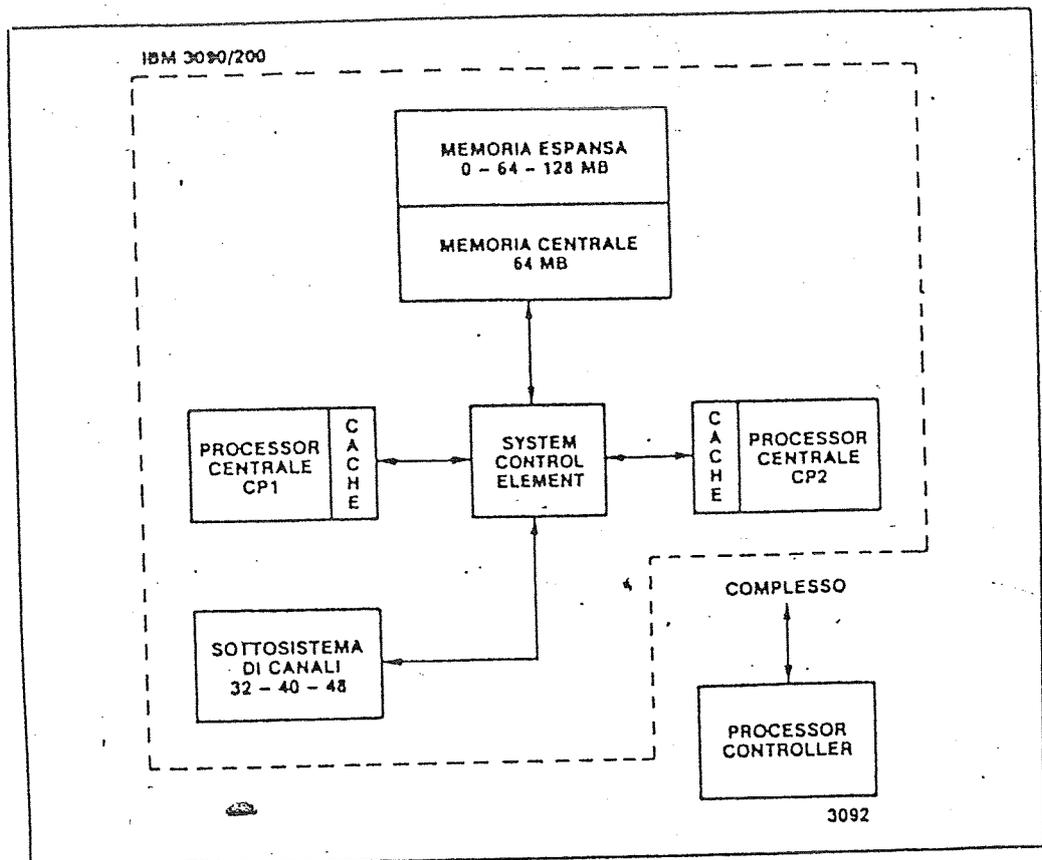
Generalita'

Il sistema IBM 3090 annunciato ed immesso sul mercato agli inizi del 1985 si presenta in sette modelli:

- 120E;
- 150E;
- 180E;
- 200E;
- 300E;
- 400E;
- 600E.

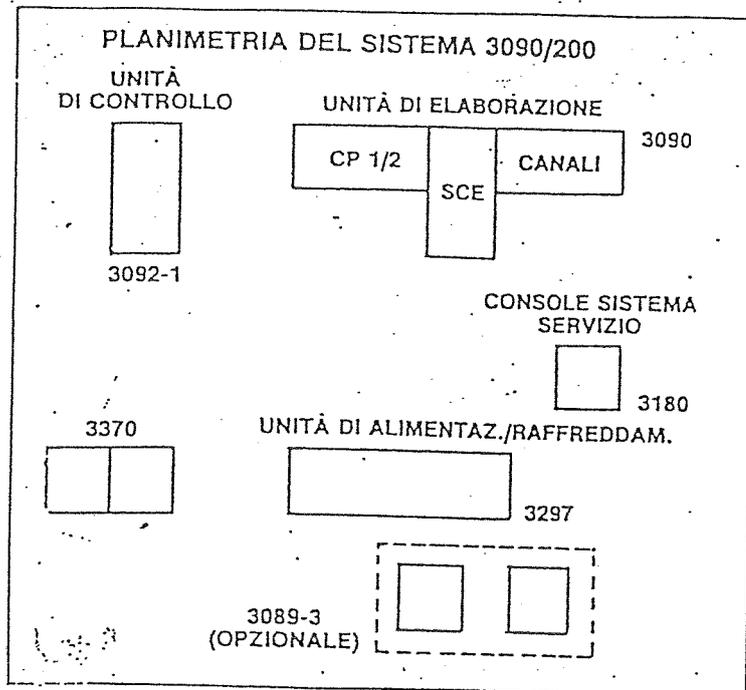
Noi parleremo del mod. 180E con alcuni riferimenti ai modelli 200E e 400E. Tralascieremo volutamente il modello 150E perche' non offre l'unita' di memoria di secondo livello o memoria espansa, e il numero dei canali e' limitato a 24.

figura 1



Il sistema IBM 3090 si presenta con la seguente planimetria:

figura 2



Legenda:

- 3090 Unità di elaborazione;
- 3180 Consoles di sistema;
- 3297 Unità di cooling & power;
- 3092 Unità elaborative (due per ass. tec.);
- 3370 Dischi in linea per 3092;
- 3420 Disponibilita' nastri 3420, o comp.;
- 3864 Modem per RSF.

Il sistema 3090 e' una evoluzione tecnologica/architetturale dei 308X, e per comprenderne l'effetto e' opportuno accennare al disegno dei 308X. La tecnologia che ha caratterizzato il 308X e' la large scale integration (L.S.I.), che si ritrova nel progetto del 3090 sia per il raffreddamento dei circuiti elettronici sia per la loro alimentazione. Infatti la tecnica di montaggio di un gran numero di chips -insieme di circuiti elettronici elementari- su una stessa board, sebbene fattibile, impone delle restrizioni sul tipo dei circuiti impiegati dovute ai problemi che si manifesterebbero sia al momento della distribuzione dell'alimentazione elettrica, sia quando e' richiesto il loro raffreddamento. I due problemi sono stati risolti utilizzando il Thermal Conductor Module (TCM). Grazie al TCM siamo oggi in grado di alloggiare 133 chips in un'area di 80 cmq., riducendo, in maniera notevole, le distanze che i segnali debbono percorrere; e gli stessi TCM garantiscono al progettista un efficiente raffreddamento dei circuiti elettronici installati. E' opportuno notare come la densita' circuitale dei TCM sia dello stesso ordine di grandezza della Wafer Scale Integration (WSI).

I problemi tecnici che derivano dall'impaccamento dei circuiti (densita') sono stati risolti tramite il supporto in ceramica dei TCM.

Le connessioni elettriche tra chip e chip sono realizzate con fili di molibdeno. Sebbene tale metallo presenti un'alta resistivita', la sua scelta e' dovuta all'alta refrattarieta' dei substrati di ceramica sui quali sono fissati ed impaccati i chips. Tale caratteristica termica del supporto non consente una rapida dissipazione del calore prodotto; quindi il metallo utilizzato deve avere precise caratteristiche termiche, tali da garantire il collegamento elettrico nonostante le alte temperature che si hanno all'interno di un TCM.

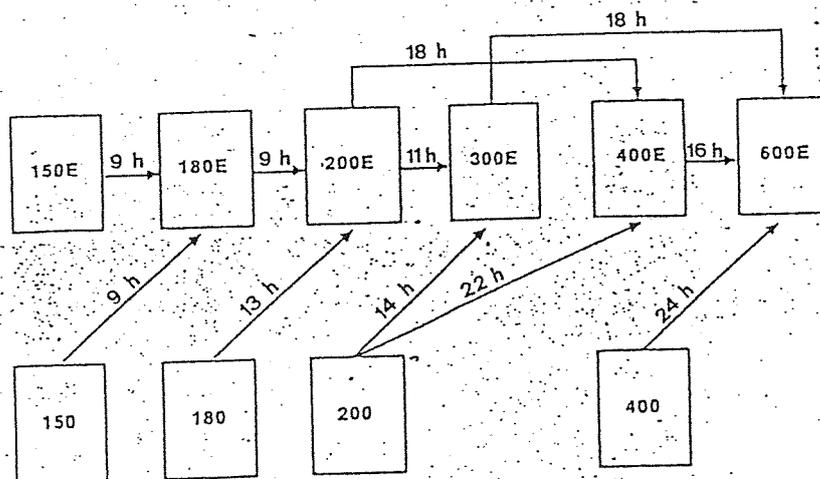
Il progetto del 308X, si oriento' su circuiti TTL. L'impiego di una 'logica bipolare' come la TTL consente una notevole riduzione dei consumi energetici del sistema. Infatti, la soluzione 'logica ECL', piu' veloce, necessita di una maggiore quantita' di corrente (3033 e 3090). Il problema della lentezza della logica TTL rispetto a quella ECL e' risolto dal minimo percorso che i segnali debbono compiere quando tali circuiti sono montati sul TCM.

figura 3

CONFIGURAZIONI IBM 3090

MODELLO	UNITA' DI ELABORAZIONE	MEMORIA MEGA-BYTES		CANALI	VECTOR FACILITY
		CENTRALE	ESPANSA		
150E	1	32 - 64	0 - 128	16 - 24	0 - 1
180E	1	32 - 64	0 - 256	16 - 32	0 - 1
200E	2	64 - 128	0 - 512	32 - 64	0 - 2
300E	3	64 - 128	0 - 512	32 - 64	0 - 3
400E	4	128 - 256	0 - 1024	64 - 128	0 - 4
600E	6	128 - 256	0 - 1024	64 - 128	0 - 6

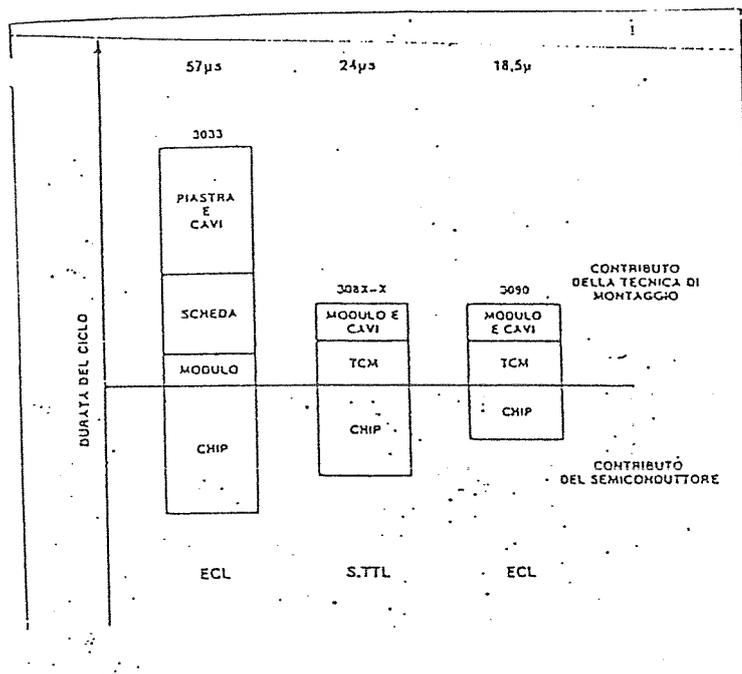
OPZIONI DI MIGRAZIONE



Sistema 3090: Tecnologia e Disegno.

E' possibile a questo punto riportare nel diagramma seguente i cicli dei sistemi 3033 308X e 3090 in funzione degli apporti che l'evoluzione tecnologica e il disegno hanno dato alla durata del ciclo macchina:

figura 4



Riportare in produzione la tecnologia ECL ha modificato la tecnica del TCM che rimanendo delle stesse dimensioni del 308X permette di alloggiare tanti ECL quanti erano i TTL con maggiore necessita' di alimentazione e conseguente impegno nella dissipazione di calore. Il fatto che i TCM fossero gia' stati collaudati per tecnologie 'calde' ha reso possibile tale realizzazione. Il risultato e' un ciclo base migliorato del 30% circa.

Il raffronto tra i due computers non e' giusto limitarlo al solo ciclo base perche' un'analisi piu' completa si ottiene considerando anche il numero medio di cicli base per istruzione. Questo parametro e' strettamente dipendente dall'organizzazione del computer.

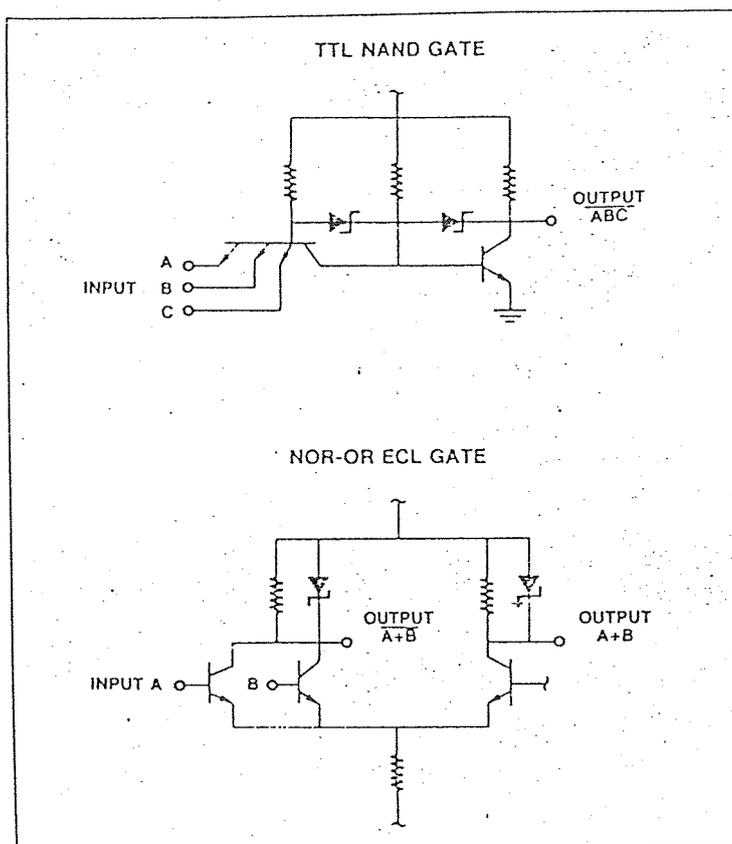
Il sistema 3090 presenta un maggior parallelismo cioe' l'insieme delle attivita' che possono essere eseguite in un ciclo macchina sono superiori a quelle del 308X. Questa maggiore attivita' comporta un numero superiore di circuiti disponibili contemporaneamente.

Essendo tutto questo realizzato sul 3090 e' ovvio che tale macchina ha un numero superiore di chips e TCM. Contarli e tramite tale numero stabilire quale sia il migliore e' anch'esso un aspetto fuorviante.

Per comprendere meglio come sia possibile aver sfruttato la tecnica di montaggio del 308X inserendola in un disegno piu' complesso e' bene ricordare come la IBM abbia da un ventennio adottato una tecnica di disegno automatizzato dei chips noto come 'gate array' o 'masterslice'.

Tale tecnica consiste di inserire strutture predeterminate di elementi circuitali in un'area del chip detta cella. Questa struttura viene quindi ripetuta in modo che ogni chip sia composto da tante celle identiche. La personalizzazione del chip avviene durante la metallizzazione, ossia l'aggiunta di strati isolanti e connessioni sopra le diverse strutture di celle. Pur avendo una standardizzazione decisamente economica si lascia la possibilita' di realizzare molte variazioni. Dato che la tecnologia ECL dispone di un numero maggiore di segnali in uscita rispetto alla tecnologia TTL i progettisti hanno maggiori varietas' di funzioni realizzabili lasciando inalterato il numero di chips. In altre parole oltre a migliorare il ciclo base, il 3090 con il disegno ha ridotto anche il numero medio di cicli per istruzione.

figura 5



Central Processor

Il Central Processor (CP) e' composto da un Instruction Preprocessor Element (I) ed un Execution Element (E). Il CP e' connesso ad una Cache. I componenti I ed E hanno delle features con disegno di parallelismo di esecuzione (pipelining) e sovrapposizione di esecuzione (overlap).

L'elemento I prefeatch (cioe' trasferisce le istruzioni dalla memoria in una unita' di governo) le istruzioni di doppia parola (double word) in uno dei tre buffers di istruzioni composto da quattro doppie parole. Queste istruzioni sono portate in un 'instruction register' dove sono decodificate (una volta decodificate, le istru-

zioni vengono trasferite in un elemento di coda (buffer) uguale a quello che precede l'istruzione register). La decodifica e la generazione degli indirizzi sono eseguite contemporaneamente e l'indirizzo viene inviato alla cache con una alimentazione di operandi. Tutto questo in un solo ciclo macchina. Decodificata con un indirizzo di starting control store, l'istruzione e' posta dentro un buffer di istruzione di quattro elementi, chiamato 'di coda pronta', per essere eseguita dall'elemento E.

Se ci sono ritardi nell'elemento I, la macchina puo' continuare l'elaborazione attraverso E prelevando altre istruzioni dalla 'buffer di coda'. Questo puo' avvenire quando, per esempio l'elemento I aspetta di caricare un'istruzione alimentata da un ritorno di una condizione di branch.

In un'istruzione di branch, l'elemento I puo' continuare l'alimentazione delle istruzioni nella linea principale ed anche far partire un altro caricamento di istruzioni sotto il comando di branch utilizzando un altro instruction buffer.

Creata questa condizione sia che il salto avvenga oppure no, la macchina ha gia' pronte le istruzioni e switchera' allo stream di alimentazione a seconda della condizione di esecuzione del branch.

L'elemento I puo' alimentare una doppia parola per ciclo. La gestione dei tre instruction-stream avviene su prioritaa'.

L'elemento E contiene un operand buffer di doppie parole che sono alimentate dall'I element e ritornano alla cache direttamente da E.

Ogniqualvolta un'istruzione valida e' presente nella coda essa viene eseguita e rimossa.

L'E element e' in grado di eseguire un certo numero di istruzioni in un solo ciclo macchina (istruzioni semplici).

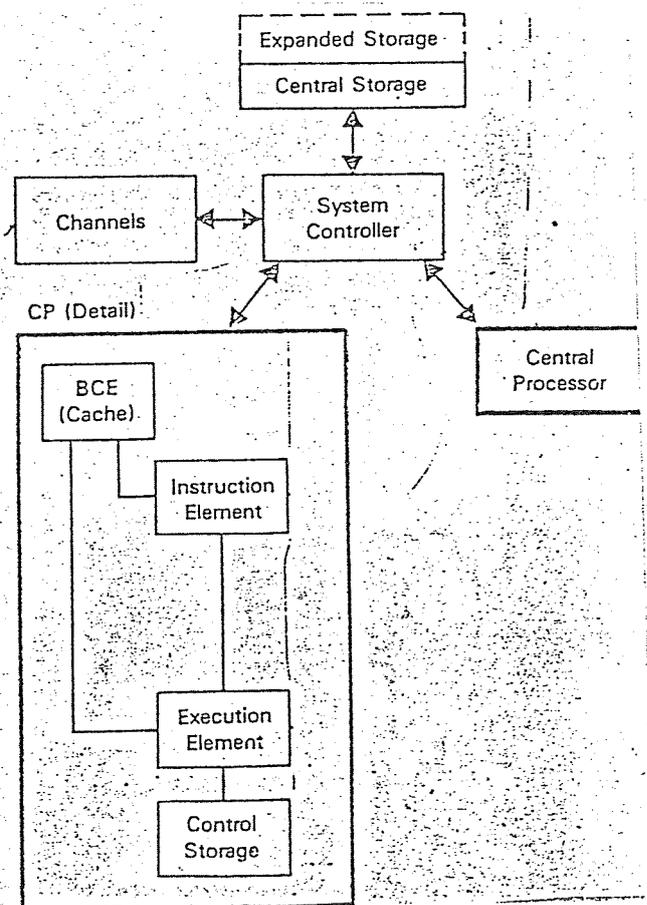
L'elemento E ha un microcodice (Horizontal Microcode) scritto con una lunga microparola (oltre 100 bits) che controlla piu' campi.

Questo permette, durante un singolo ciclo macchina, un uso simultaneo parallel adder, shifter, serial adder, scrivere e leggere general register e, allo stesso tempo, aggiornare contatori.

Tutti questi risultati sono possibili in una organizzazione pipeline.

figura 6

Central Processor (CP)



Memoria Centrale

La memoria centrale del sistema 3090 comprende una cache per ciascun processore centrale, un elemento di controllo sistema (S.C.E), il central storage arrays, e l'expanded storage feature.

La " cache"

A partire dal sistema 360/85 tutte le macchine IBM hanno avuto una cache. La cache e' una (relativamente) piccola memoria con un rapido tempo di accesso che e' usata per contenere porzioni della memoria centrale. La gestione e' organizzata in modo da mantenere nella cache le pagine (porzioni di memoria centrale) piu' recentemente referenziate.

Nel caso del 3090, la cache (HSB) ha una taglia di 64k-byte con pipeline e due cicli di accesso.

Nel progetto di una cache si hanno piu' soluzioni. Il 3033, per esempio, utilizzava un disegno denominato 'store-through'. Questa relativa semplice organizzazione, chiamata store-through, e' tale che, tutte le memorizzazioni vanno non solo alla cache ma anche alla central storage array. In questa organizzazione, la central storage array contiene l'ultima copia dei dati memorizzati. Tale organizzazione penalizza ovviamente il sistema perche' ogni comando di memorizzazione effettua due store: uno in memoria centrale l'altro nella cache. Cosi' sul 3090, come sul 308X, viene usata la soluzione 'store-in-cache'. Nel progetto store-in-cache, l'operazione di store si limita a memorizzare il dato nella cache senza inviarlo immediatamente in memoria. Il dato rimane soltanto nella cache finche' un altro processor richiede quella linea o fino a quando la locazione della cache e' richiesta da un'altra linea. Questa gestione e' una complicazione per il system control element che ad ogni richiesta di memorizzazione deve determinare dove l'ultima copia di una linea di memorizzazione risiede: nel central storage array (memoria centrale) o in quale cache.

La cache e' suddivisa in quattro sets di 2k doppie parole ciascuno.

La cache puo' essere acceduta per i seguenti 4 motivi:

1. Processor fetcher (prelievo dati);
2. Processor stores (memorizzazione dati);
3. Cache line (trasferisce dal sistem control element (SCE));
4. Cache line (trasferisce verso il SCE).

In tutti questi casi il trasferimento puo' essere fatto attraverso una doppia parola (8 bytes) per ciclo.

System Control Element

Fisicamente il SCE serve per connettere il central processor, il canale subsystem, e le altre parti del sistema al central storage array.

Logicamente esso consente l'accesso alla memoria centrale del central processor e del channel subsystem.

Il fatto che il SCE sia a conoscenza che l'ultima copia risieda, in una cache, nel central register array si assicura sempre una singola ultima copia di ciascuna linea e può essere abilitato a trasferirla a ciascuno dei richiedenti. I richiedenti di un SCE sono i due central processor, i channel subsystem, e i remoti SCE. Il SCE ha due registri richiesta per ciascun richiedente, uno per fetches e uno per store. Questi registri sono connessi attraverso richieste indipendenti e indirizzi occupati, e possono essere settati da un richiedente ad ogni ciclo in cui sono liberi. In ciascun ciclo, il SCE può esaminare tutti gli otto registri richiesta e selezionarne uno con priorità di base.

The Expanded Storage

La memoria espansa è una nuova memoria disegnata per il trasferimento di pagine (4096 Bytes).

Essa non è gestita in maniera asincrona come un normale device di i/o, e' semplicemente una parte addizionale della memoria centrale; infatti la memoria espansa gira in sincronia con il Central Processor che richiede i trasferimenti.

Il trasferimento delle pagine tra se e la memoria centrale avviene sotto il controllo del sistema operativo.

La decisione di utilizzare la memoria espansa in maniera sincrona deriva dal fatto che si evitano tutta una serie di controlli necessari ad una apparecchiatura che opera in maniera asincrona.

I block-data vengono trasferiti direttamente tra l'expanded storage e la central storage, con una piccola interferenza con il resto della memoria del sistema.

Reliability features of the storage

Il 3090 storage system ha implementato degli algoritmi ad hardware che consentono la correzione degli errori che si manifestano sia nella memoria principale sia nella expanded storage (memoria estesa).

La tecnica è basata su un concetto chiamato complement/recomplement o double complement.

Gli errori che si manifestano possono appartenere a due classi:

Hard errors;

Soft errors.

Gli hard errors sono quelli persistenti; i soft errors sono random perciò non localizzabili. L'algoritmo di cui sopra consente, nella memoria principale una correzione di un errore hardware ed un errore software che si manifestino, anche contemporaneamente, su un suo segmento (8 bytes). Per quanto si riferisce alla external memory la reliability feature è in grado di correggere due errori hardware ed uno software che si manifestino, anche contemporaneamente, su un suo segmento (16 bytes).

Channels

Le strutture dei canali del 3090 e' uguale a quella del 308X ma ne aumenta il numero (doppio) e la velocita' (4.5 MB).

La gestione dei canali tramite l'architettura XA e' ottenuta da un sistema dedicato XSDE o DCS.

Il sottosistema di canale DCS rappresenta una macchina a se stante che gestisce le risorse I/O dell'intero sistema sganciandosi dai vecchi legami che volevano le riconessioni procedere sulla stessa strada delle connessioni. In questo modo si fornisce una indicazione dinamica al percorso dei dati scegliendo di volta in volta le vie libere, consentendo una riconnessione indipendente dai busy di canale e/o di control unit.

figura 7

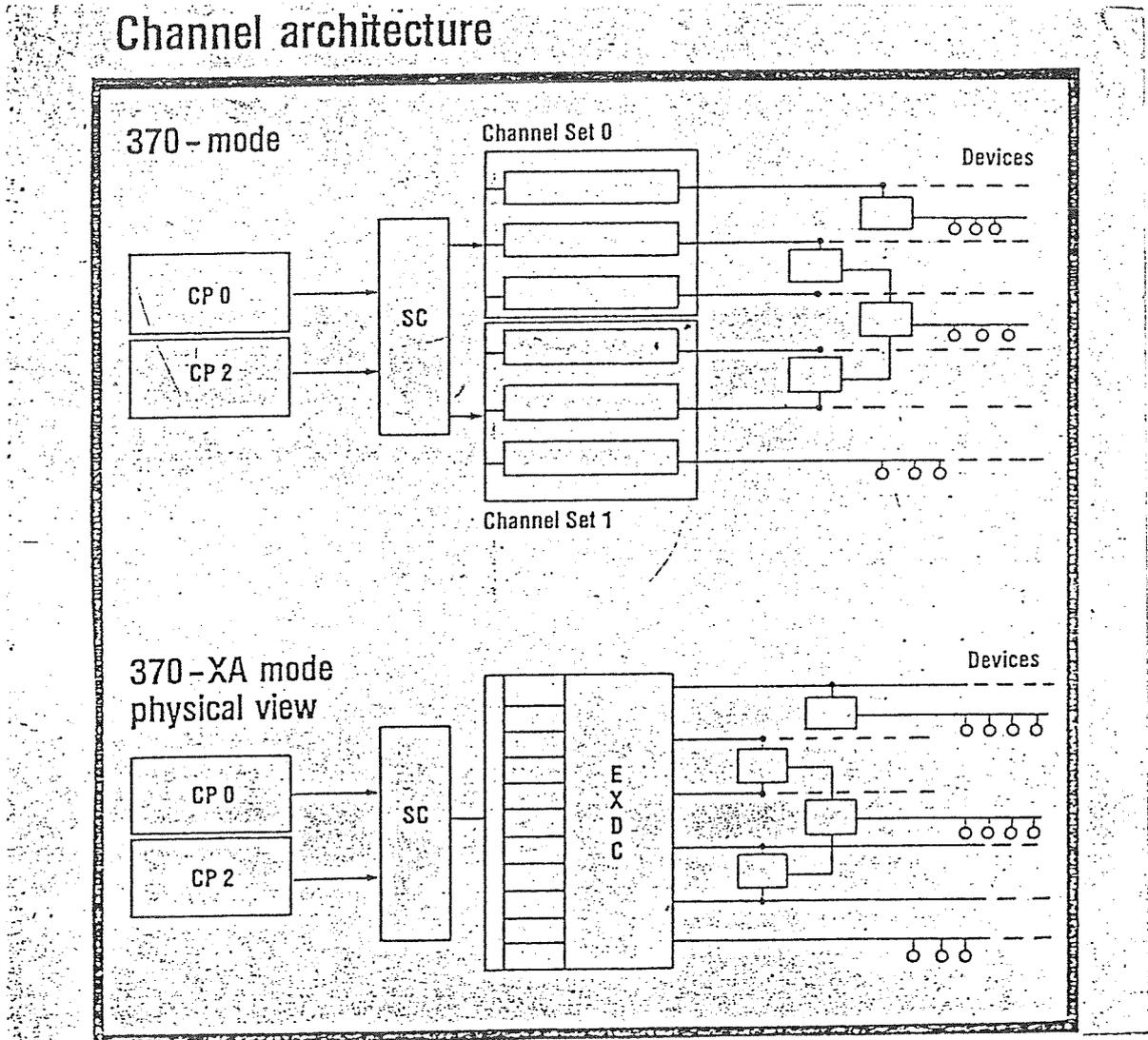
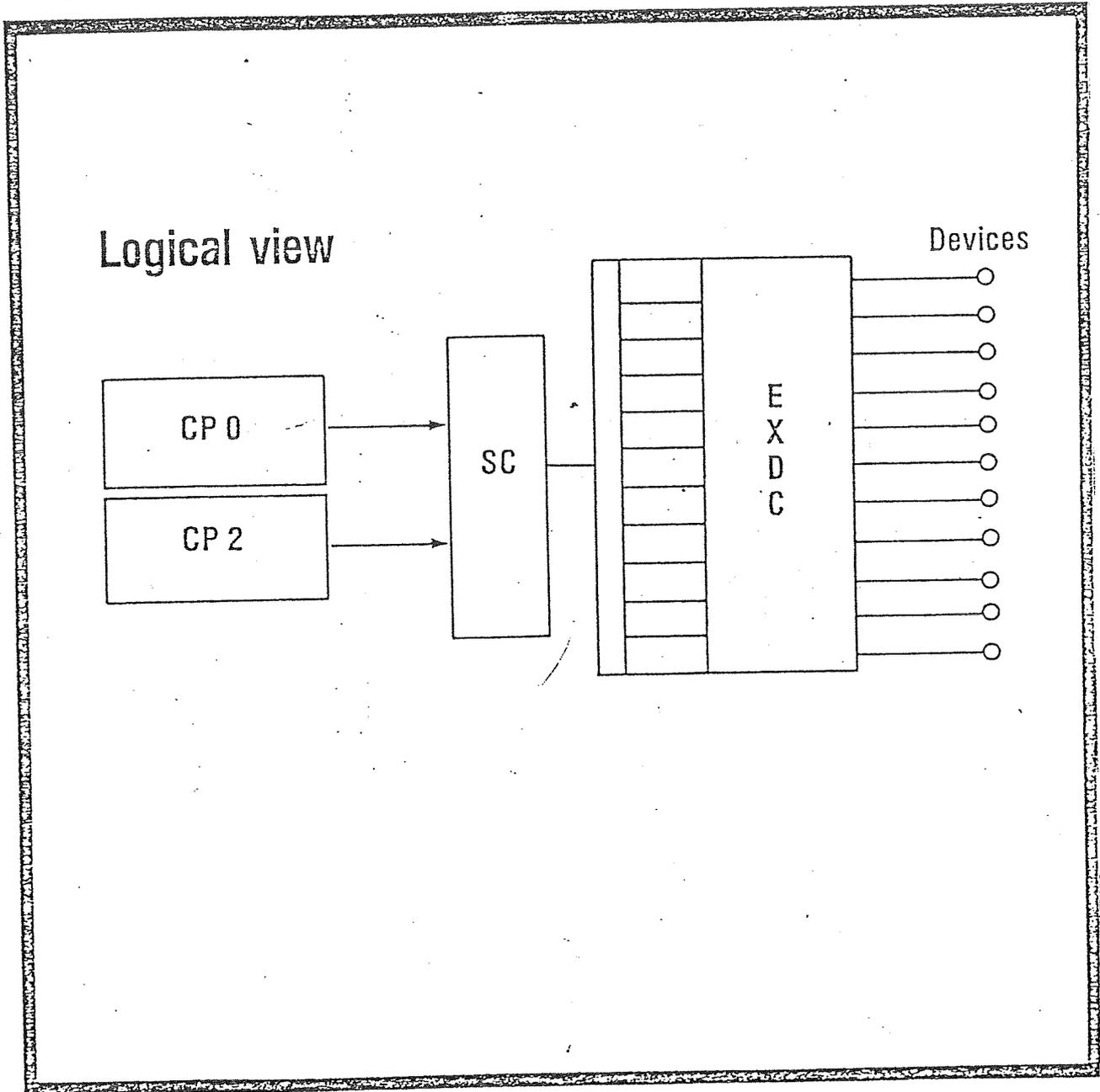


figura 8

Dynamic Channel Subsystem



Conclusioni

Nel 3090 sono state potenziate le seguenti funzioni:

- l'insieme delle istruzioni tecnico scientifiche;
- moltiplicazione ad alta velocità (3 c.m.);
- somma e sottrazione ad alta velocità;
- gestione dei loops ad alta velocità;
- trasferimento dei dati a 64 bit;
- sovrapposizione istruzione/esecuzione (overlap).

La gestione delle funzioni di moltiplicazione ad alta velocità e' stata resa possibile dalla realizzazione di particolari circuiti, mentre per le operazioni di somma e sottrazione sono state realizzate nuove tecniche di microprogrammazione che hanno consentito di ridurre sostanzialmente il numero di cicli necessari. Per la gestione dei loop sono stati realizzati opportuni circuiti e sono state sovrapposte alcune fasi dell'intera attività.

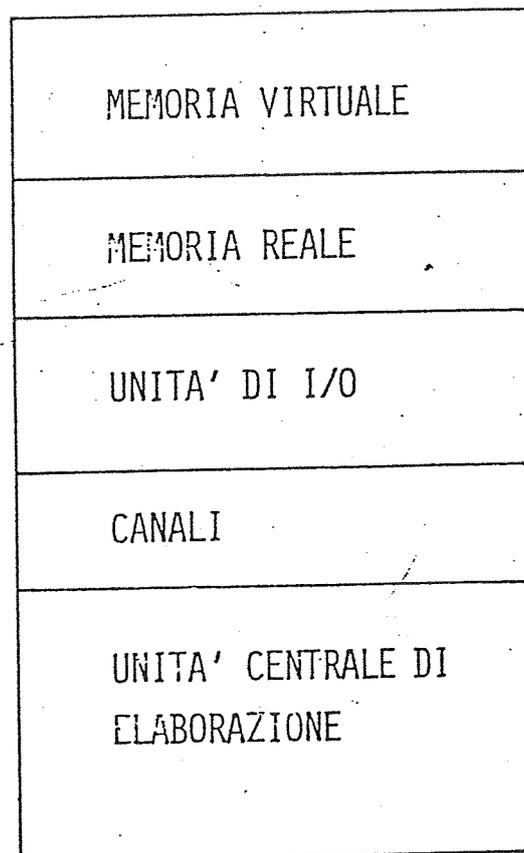
MVS/XA
CENNI

Architettura XA in MVS

Le risorse di un elaboratore sono rappresentate da:
Memoria virtuale, Memoria reale, Unita' di i/o , Canali, Unita' centrale di elaborazione.

figura 9

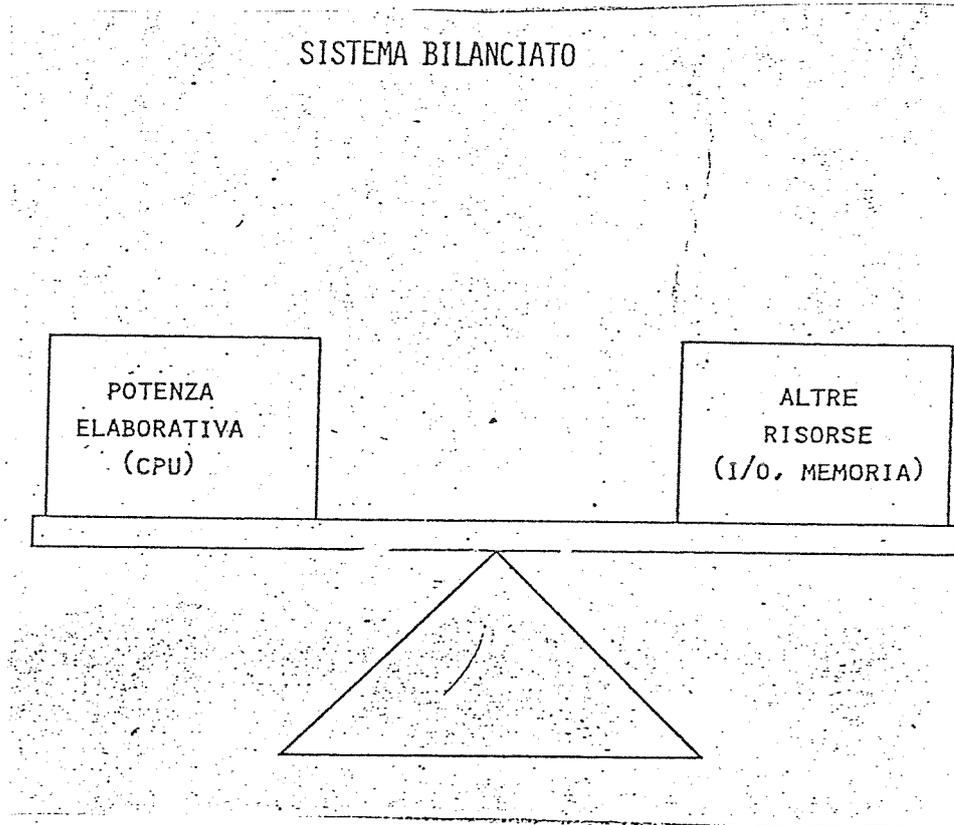
RISORSE DI UN ELABORATORE



POTENZA ELABORATIVA
IN RAPIDO AUMENTO
←

Le case costruttrici ed i progettisti sono tesi a migliorare sempre piu' la potenza elaborativa delle macchine e/o dei sistemi. In ogni caso lo sfruttamento ottimale della potenza di calcolo che viene messa a disposizione dell'utenza dipende dal bilanciamento tra la potenza disponibile della CPU e le risorse ad essa collegate (memoria, I/O).

figura 10



L'architettura 370 poneva dei limiti all'utilizzo dei grossi e potenti elaboratori:

- Max 16 MB d'indirizzamento di memoria virtuale;
- Max 32 MB di memoria reale;
- Max 1917 unita' periferiche;
- Max 16 canali per CPU;
- Affinita' CPU/CHANNELS.

I limiti elencati dovevano in qualche modo essere rimossi e l'introduzione di una architettura estesa che non rivangasse il passato ma ne rappresentasse la naturale espansione dava luogo alla architettura XA.

- L'architettura XA consente le seguenti espansioni:

1. Indirizzamento memoria virtuale a 24/31 bit;
2. Indirizzamento memoria reale a 31 bit;
3. Sottosistema per la gestione dei canali;
4. Protezione a livello di pagina;
5. Nuove istruzioni.

Utilizzo della memoria estesa

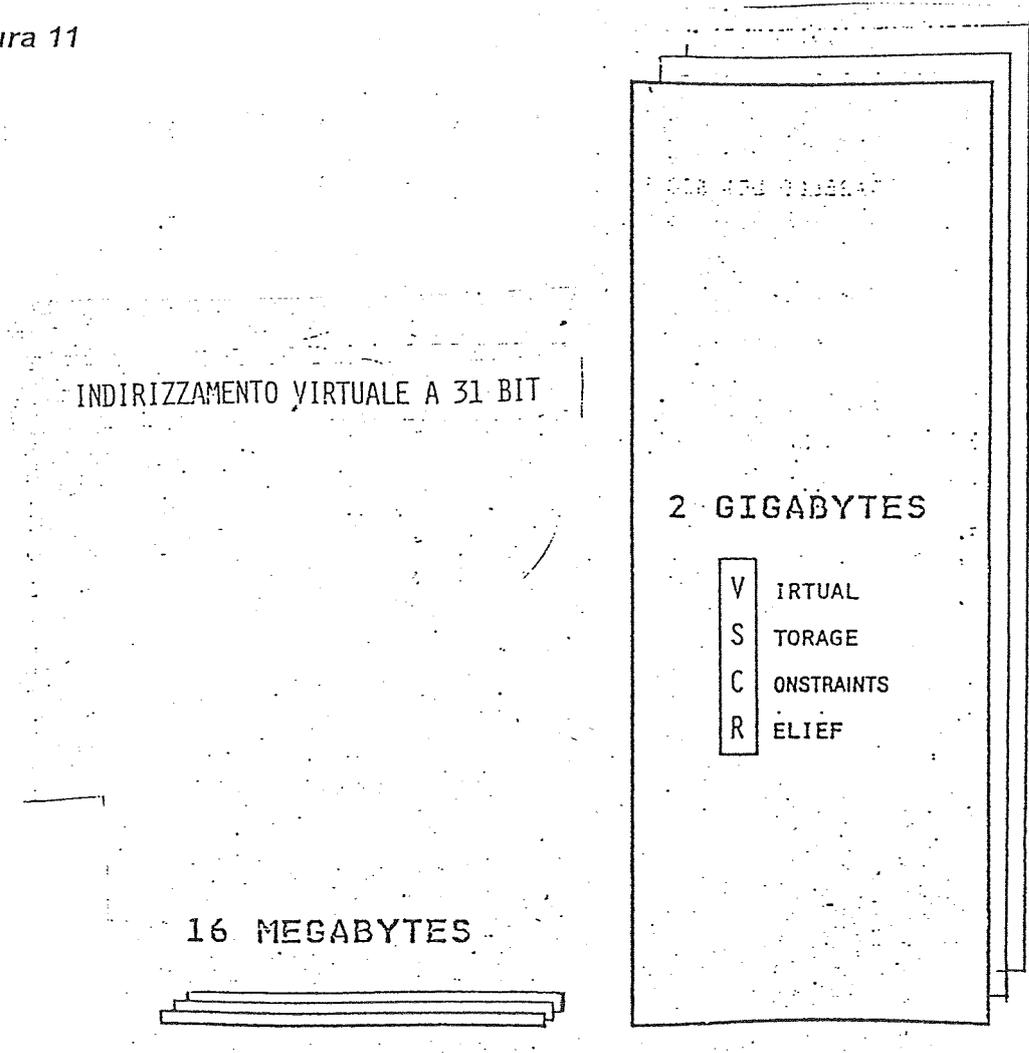
L'architettura XA permette due tipi di indirizzamento:

1. Virtuale esteso
2. Reale esteso

Indirizzamento virtuale esteso

Struttura della memoria virtuale segmento e pagine.

figura 11



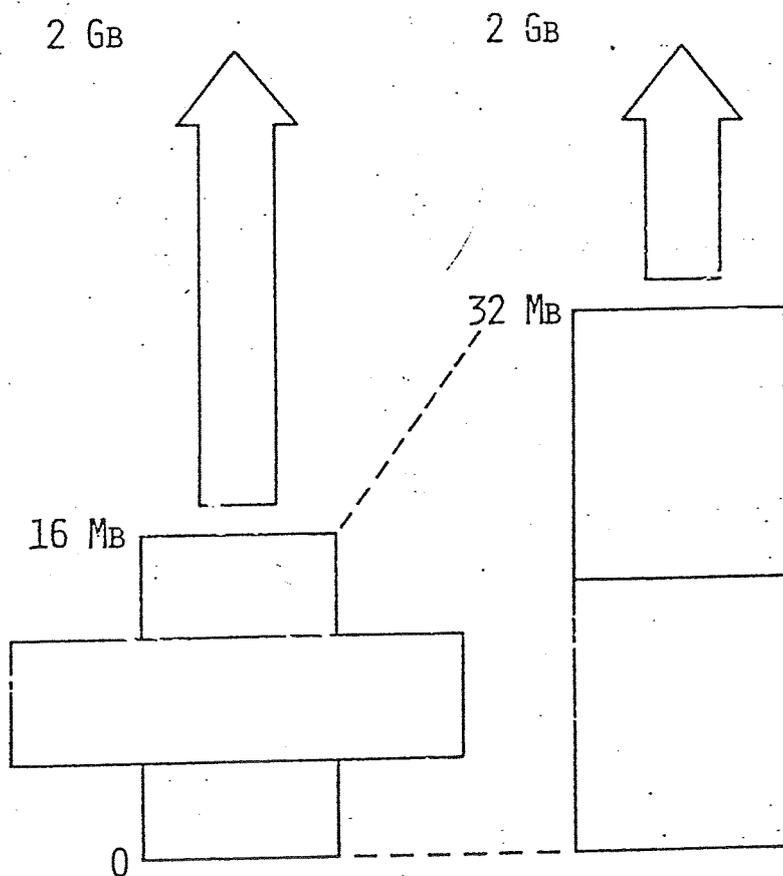
Indirizzamento reale esteso

L'indirizzamento reale esteso comporta modifiche a:

- Tabelle dei segmenti;
- Tabelle delle pagine;
- Prefix register;
- Istruzione L.R.A. (Load Real Register).
- Channel Control Word (C.C.W.).

figura 12

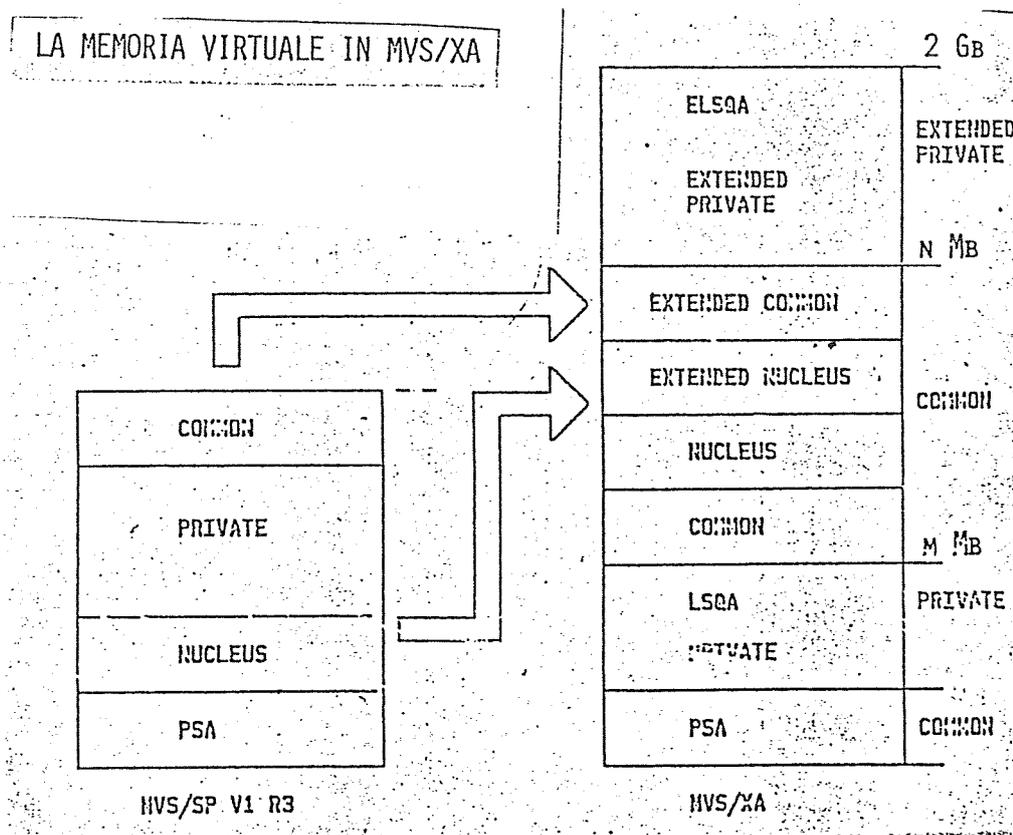
INDIRIZZAMENTO ESTESO DELLA MEMORIA



La memoria virtuale in MVS/XA

- Common Area necessaria per il passaggio di informazioni tra un address space ed un altro (indirizzo mem.virtuale)
- Private Area privata (istr. programma)
- Nucleus operating system

figura 13



La memoria virtuale in MVS/XA consente i seguenti vantaggi:

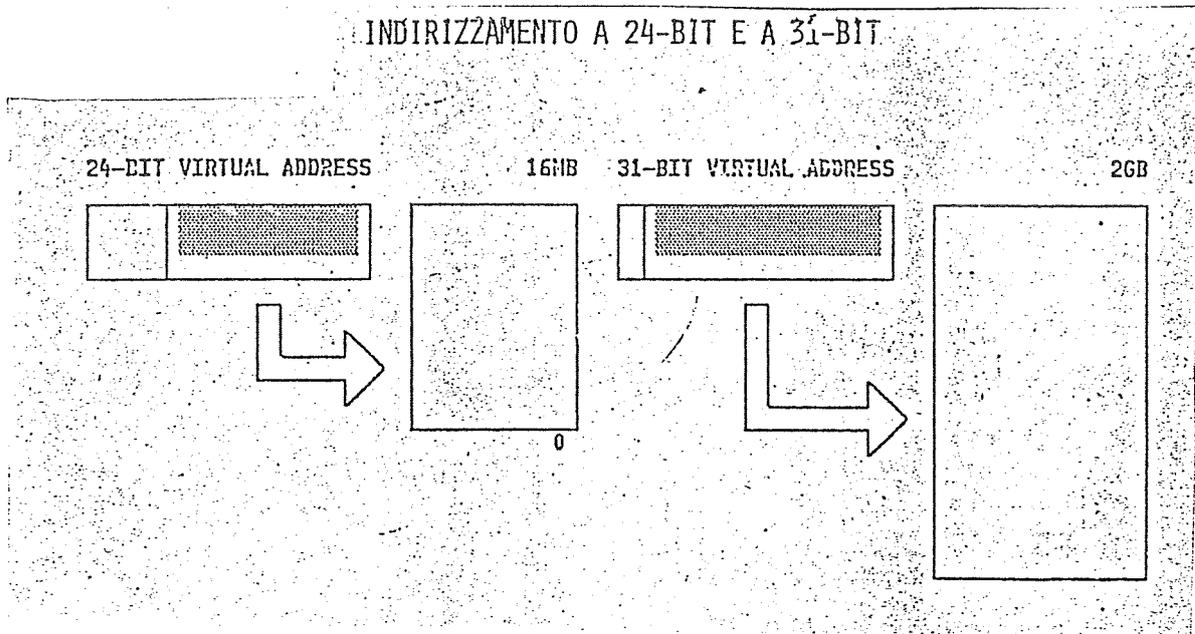
- Indirizzamento 2 Gbytes
- Indirizzi di separazione tra le aree private e area comune arrotondati al Mb (segmento)
- Compatibilita' con i programmi aventi indirizzamento a 24 bit.

Inoltre lo spostamento del nucleo a cavallo del segmento 16 e la gran parte dell'area comune mossa sopra i 16 Mbytes consentono una maggiore disponibilita' di memoria virtuale sotto i 16 Mbytes.

Indirizzamento a 24 bit e a 31 bit

L'architettura XA consente l'utilizzo di entrambi gli indirizzamenti che viene chiamato indirizzamento bi-modal. A tale proposito sono state inserite nuove istruzioni per lo switch dinamico tra i due modi. L'indirizzamento BI-MODAL (24 o 31 bit) si ottiene tramite lo stato del bit 32 della PSW:

figura 14

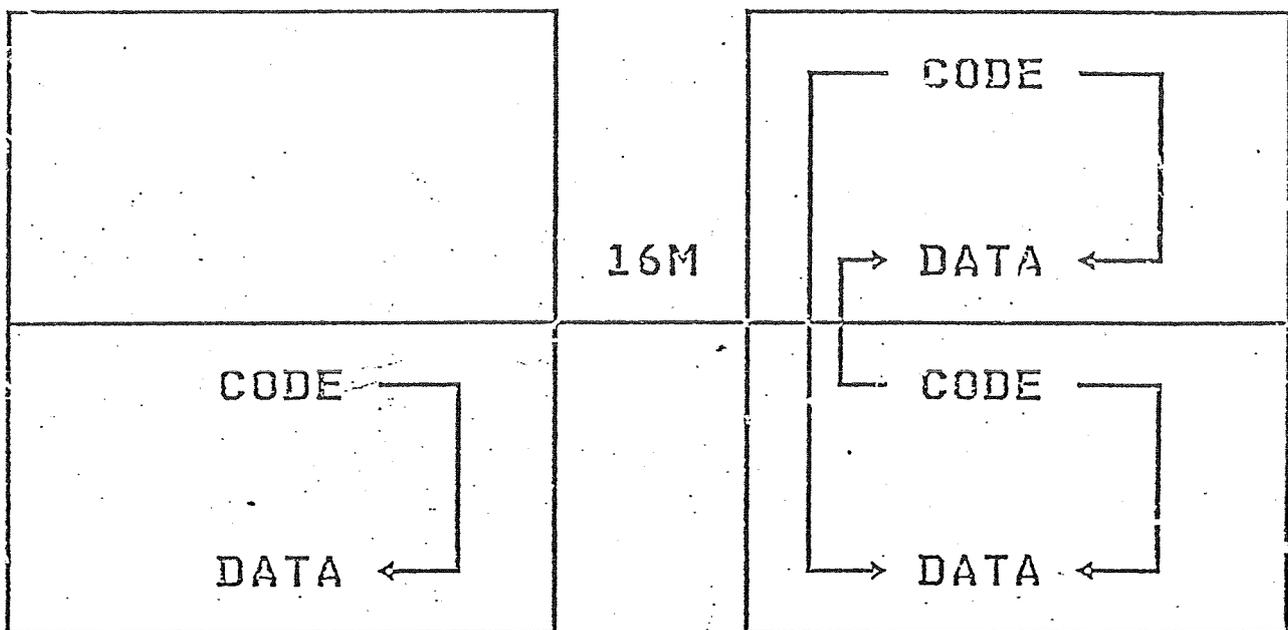


Il tipo di indirizzamento definisce sia il modo in cui il programma prende il controllo, sia il modo in cui un programma esegue.

Esempio di indirizzamento BI-MODAL

figura 15

INDIRIZZAMENTO BI-MODAL (24-BIT E 31-BIT)



Attributi di programma : AMODE e RMODE

Sono comandi che definiscono che tipo di indirizzamento si usa (24 o 31 bit) e dove il programma deve risiedere.

L'AMODE quindi definisce il tipo di indirizzamento col quale il programma prende il controllo, mentre l'RMODE definisce se un modulo deve risiedere sotto i 16MB o ovunque nella memoria virtuale.

Il sistema operativo assicura il caricamento di un modulo nell'appropriata locazione e il richiamo di un programma con relativo indirizzamento.

Durante l'esecuzione del programma vi e' la possibilita' di effettuare agganci tra i moduli con diverso AMODE.

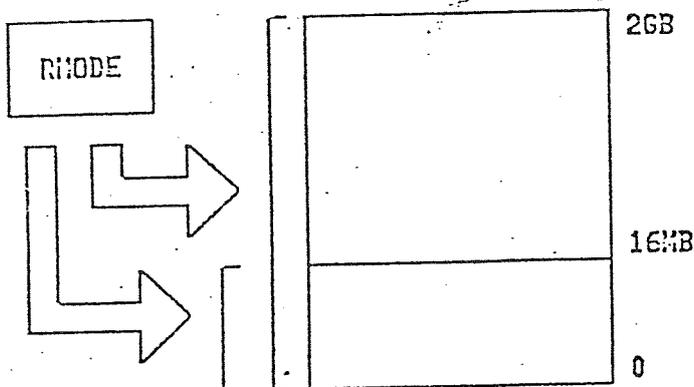
Per ottenere cio' occorrera' rispettare le seguenti convenzioni:

- Salvataggio/ripristino AMODE del chiamante
- Set del proprio AMODE
- Indirizzi passati come parametri devono essere indirizzabili da ambedue moduli
- High order bytes degli indirizzi dovrebbero contenere zeri.
- Indirizzi passati a programmi con AMODE 24 devono essere a 16MB.

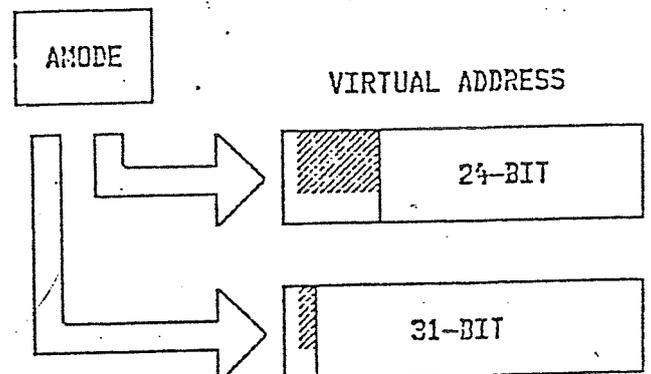
figura 16

ATTRIBUTI DI PROGRAMMA: AMODE E RMODE

RESIDENCY MODE



ADDRESSING MODE



A tale scopo nell'assembler XA sono state inserite due nuove istruzioni BASSM e BSM che gestiscono gli indirizzi nel passaggio da 24 a 31 e da 31 a 24 bit di indirizzamento.

Vi sono inoltre routines di assistenza che permettono la modifica dei richiami.

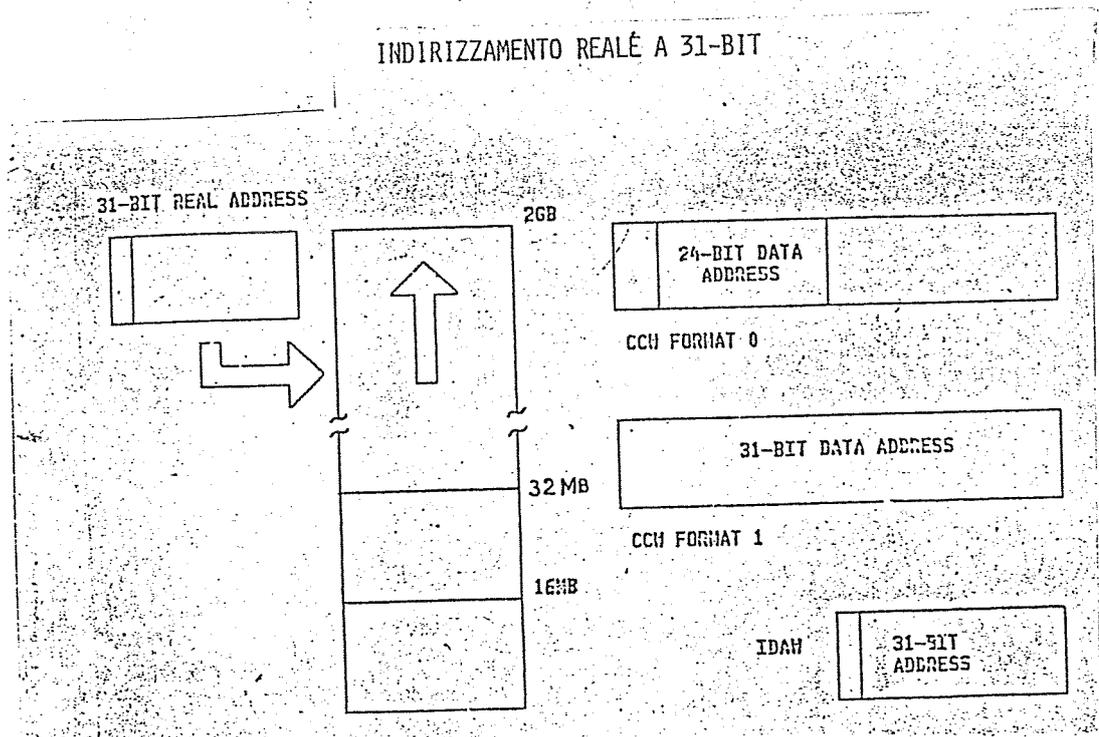
Indirizzamento Reale a bit-bit

L'indirizzamento Reale a 31 bit ha richiesto un certo numero di modifiche architettoniche:

- Nuovo formato del segment table origin;
- Nuovo formato per le entrate delle tabelle dei segmenti;
- Modificati il formato e la lunghezza delle entrate delle tabelle delle pagine (bit per protezione pagina, bit per pagina prefissata);
- Ampiezza dei segmenti;
- Nuovo formato del prefix register;
- Una differente definizione dell'istruzione L.R.A. (load real address);
- Nuovo formato di C.C.W.

L'indirizzamento reali a 31 bit consente di supportare memoria reale molto estesa perche' a questo punto non esiste alcuna restrizione all'uso di memoria reale sopra i 16Mb. Tutto questo rimane trasparente ai programmi.

figura 17



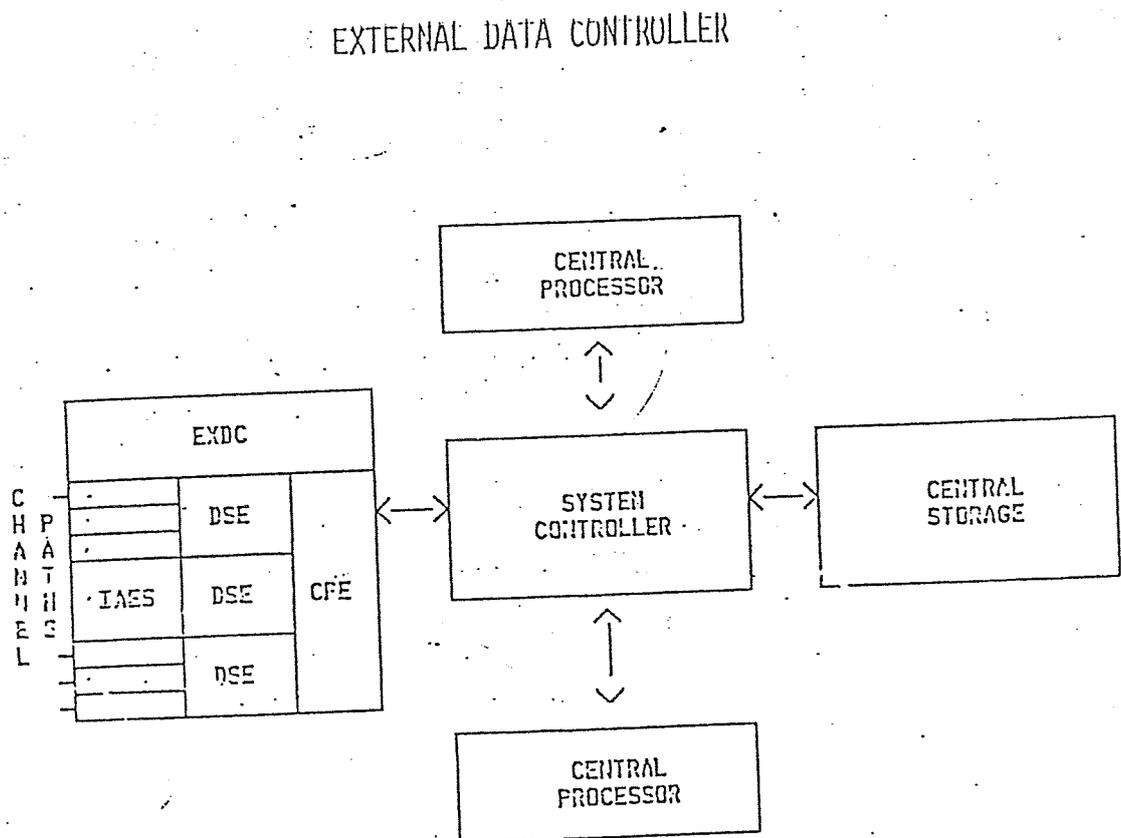
Dynamic Channel Subsystem

Il Dynamic Channel Subsystem e' il gestore delle operazioni di I/O. E' un elaboratore integrato nel sistema ed e' connesso al System Control del sistema. Ha il compito di eseguire una parte delle istruzioni di I/O sfruttando i sentieri dei dati (paths) tra la memoria e le control units. Evidenzia e controlla il trasferimento dei dati.

Il Dynamic Channel Subsystem (DCS) o External Data Controller (EXDC) e' composto da:

1. Channel Processor Element (C.P.E.);
2. Data Server Elements (D.S.E.) max 3;
3. Interface aDapter Elements (I.D.E.) max. 24.

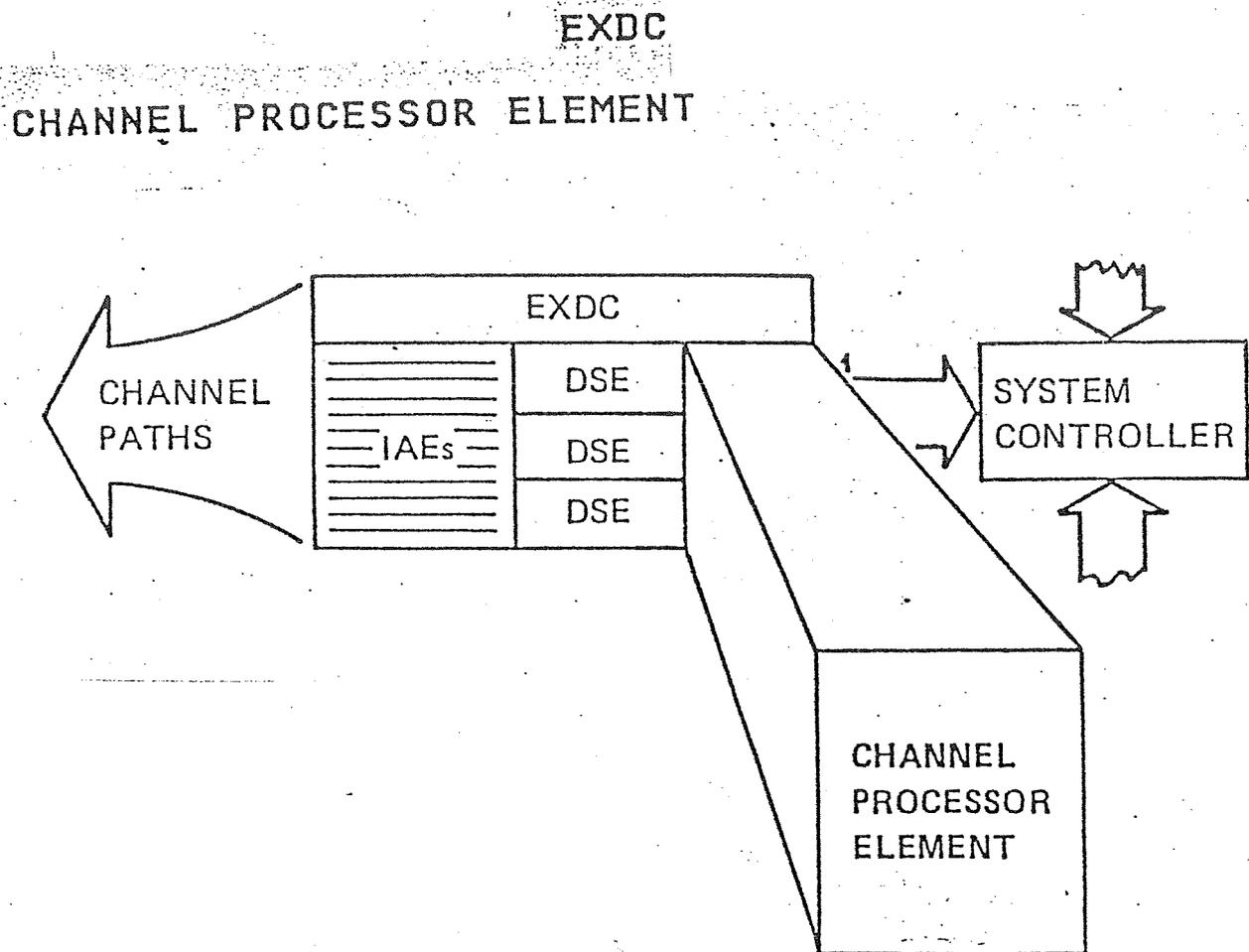
figura 18



Channel Processor Element (C.P.E.)

Il Channel Processor Element rappresenta la cpu dell'EXDC, esso riceve le istruzioni di I/O dal System Controller, provvede agli accodamenti, passa i comandi ai D.S.E. e riceve dai D.S.E. gli ending status.

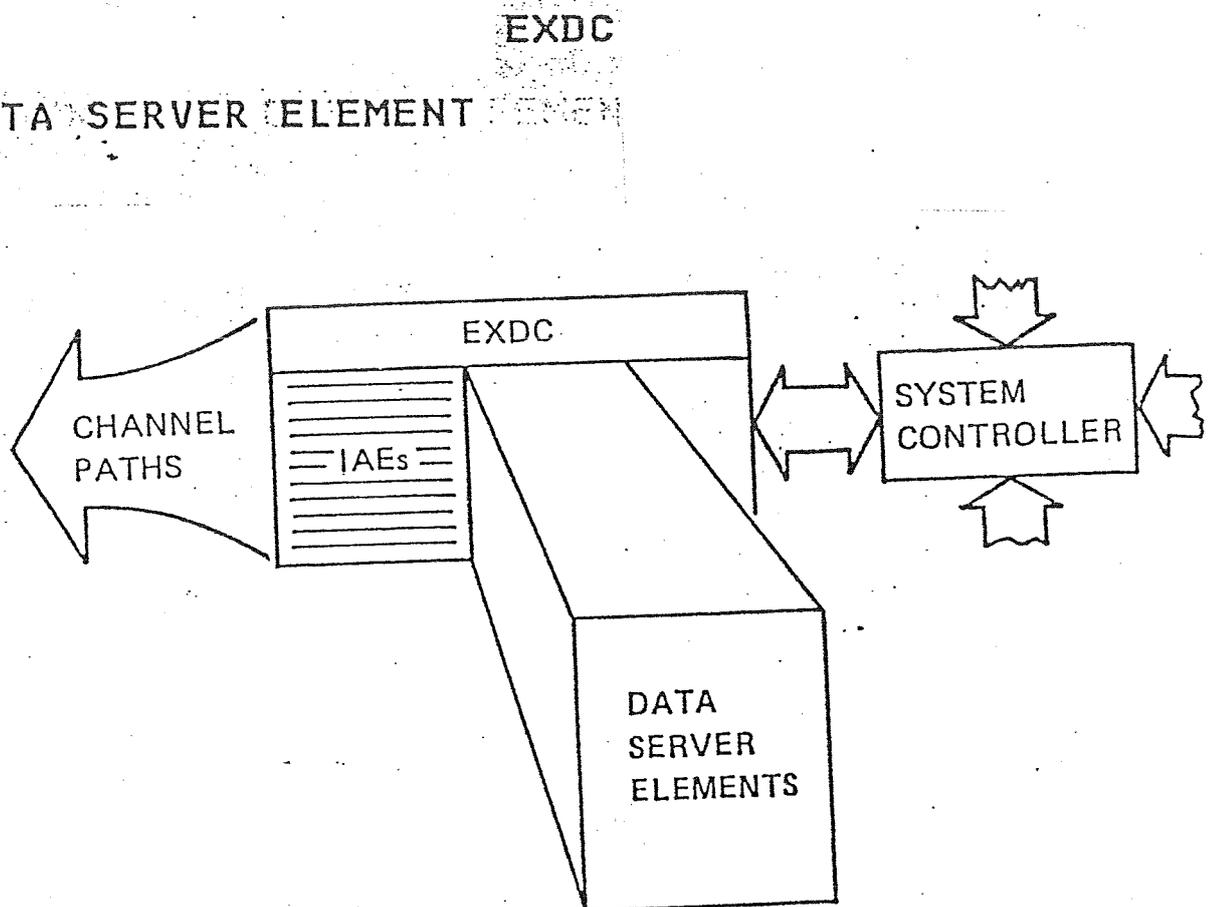
figura 19



Data Server Element (D.S.E.)

Accetta i comandi dal Control Program Element formando un buffer su i data transfer (trasferimento dati) e analizza lo stato di fine (endline status).

figura 20

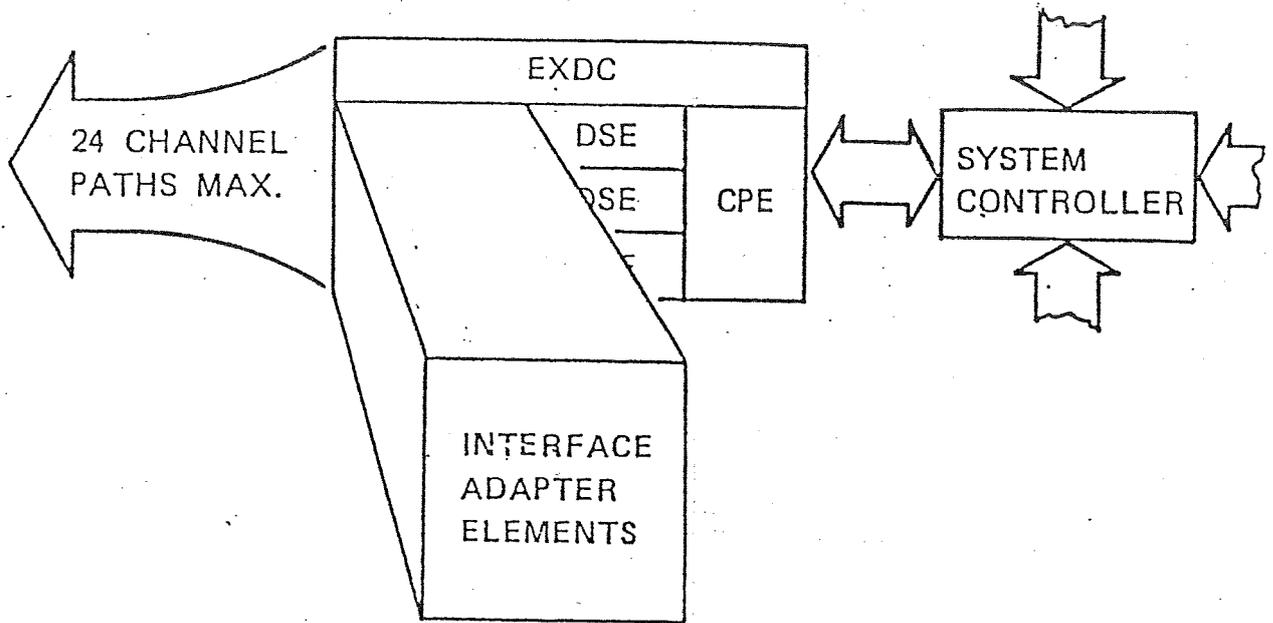


Interface Adapter Element (I.D.E.)

Interfaccia i canali fisici e bufferizza i dati durante le operazioni di trasferimento.

figura 21

EXDC INTERFACE ADAPTER ELEMENT

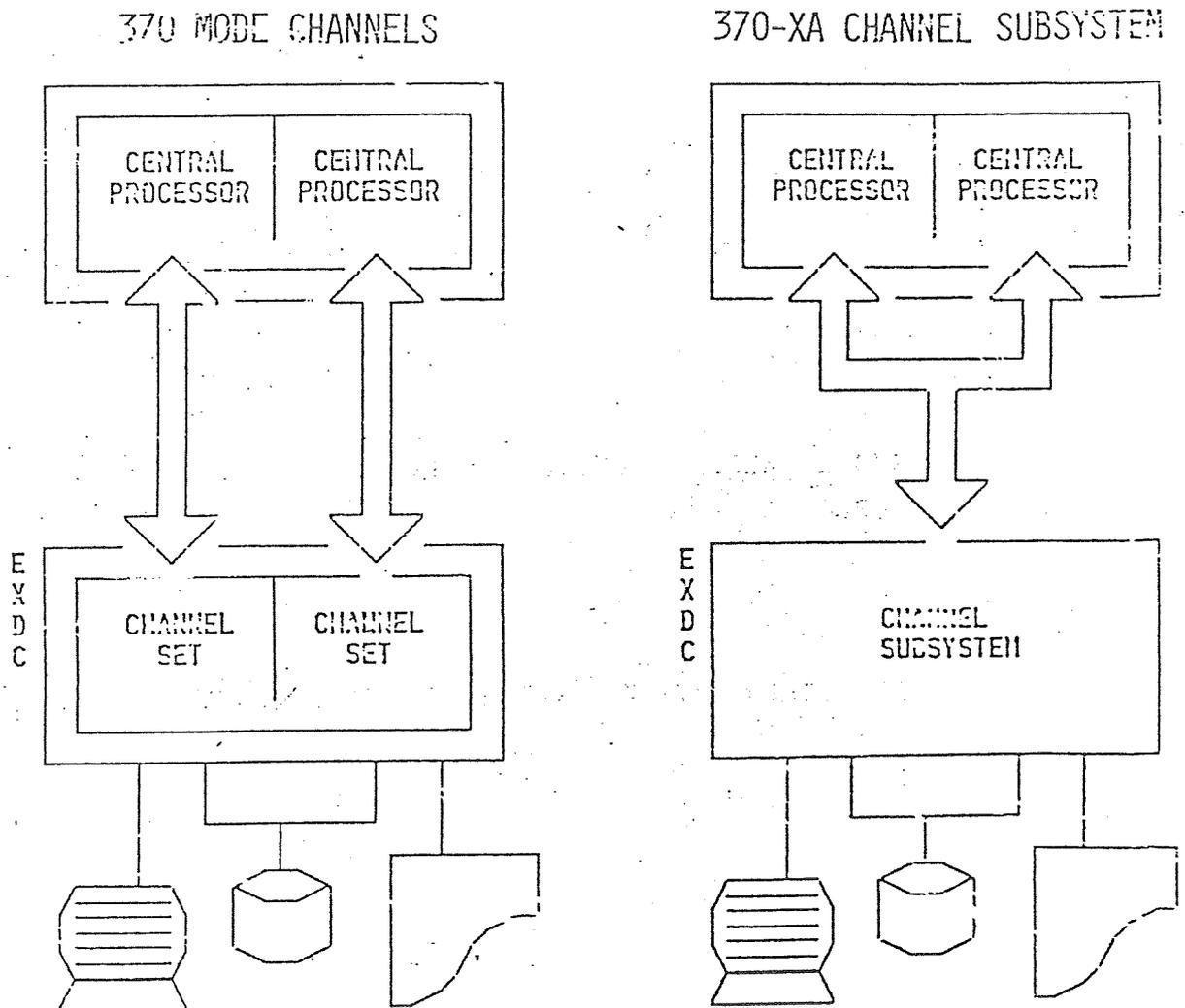


Dynamic Channel Subsystem(D.C.S.)

Il DCS offre tutta una serie di vantaggi, rispetto alla concezione di rigidita' fisica di collegamento sulla quale correvano le richieste agli i/o devices e le risposte conseguenti, che possono essere cosi' riassunti:

- Migliorato controllo dell'I/O;
- Riconnessione dinamica;
- Piu' ampie configurazioni di I/O;
- No affinita' tra CPU e CHANNEL;
- Miglioramento nei meccanismi di interrupt;
- Maggior monitoraggio.

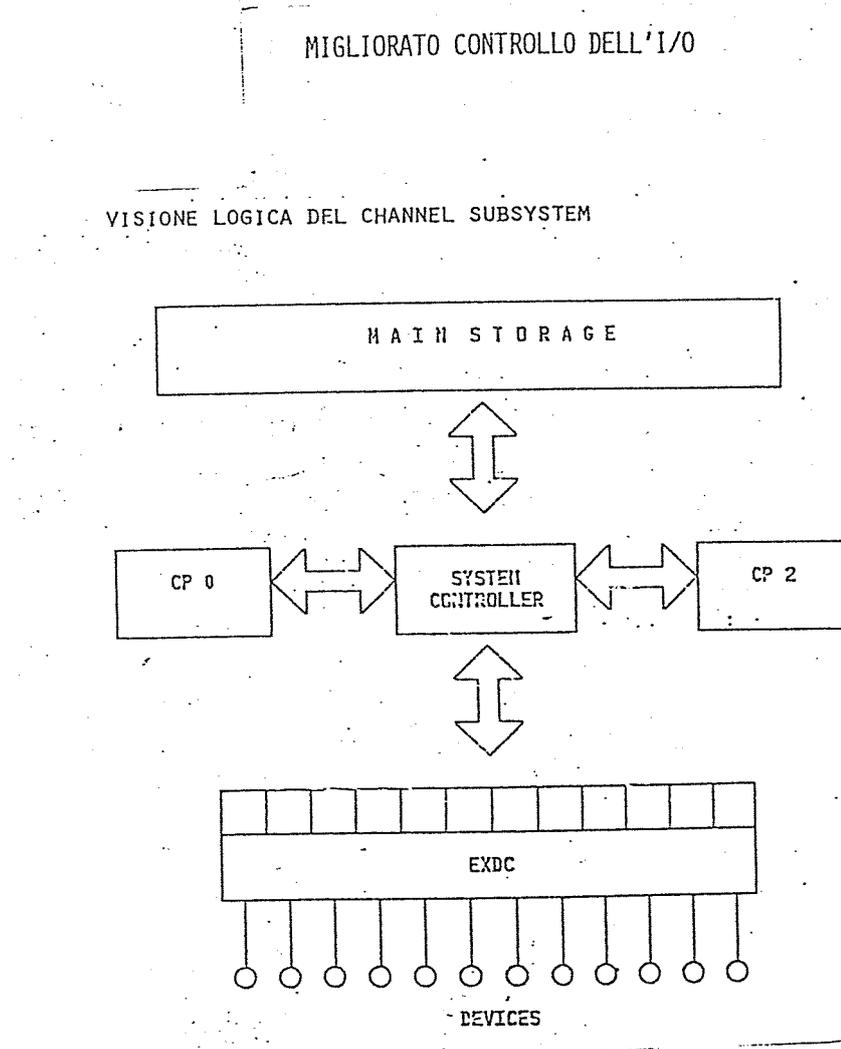
figura 22



Migliorato controllo dell'I/O

Il Channel Subsystem conosce lo stato dei paths e quindi seleziona il channel path libero (miglior utilizzo di canale), fa partire l'operazione di I/O e interrompe il sistema operativo non appena l'I/O e' completato. Misura la path utilization.

figura 23



Scelta del path

E' compito del path management nel Channel Subsystem eseguire tutta una serie di attivita':

- Test circa la disponibilita' del path;
- Elaborare la scelta del path;
- Gestione path, control unit, device busy;
- Elaborazione della scelta del path alternato (8 per device).

Il Channel Subsystem e' a conoscenza dello stato dei path e puo' quindi selezionare il channel path libero. Questo comporta un migliore utilizzo del canale in quanto la scelta della strada libera consente lo scarto dei canali occupati ed accede ai devices attraverso altri path in quel momento liberi (free). Tutto questo determina un bilanciamento automatico dell'utilizzo dei canali. Infatti il sistema non si trovera' mai nella condizione di avere un canale sottoutilizzato ed un altro sovrautilizzato: inconveniente abbastanza frequente nell'architettura 370.

figura 24

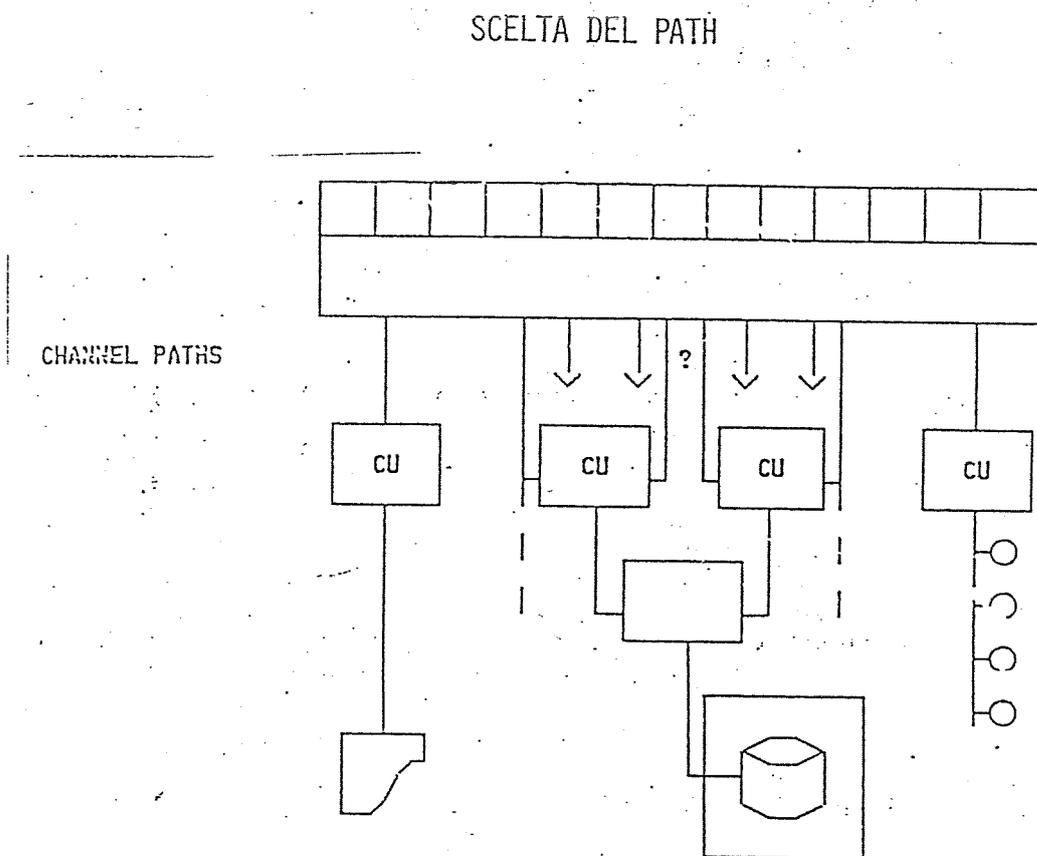
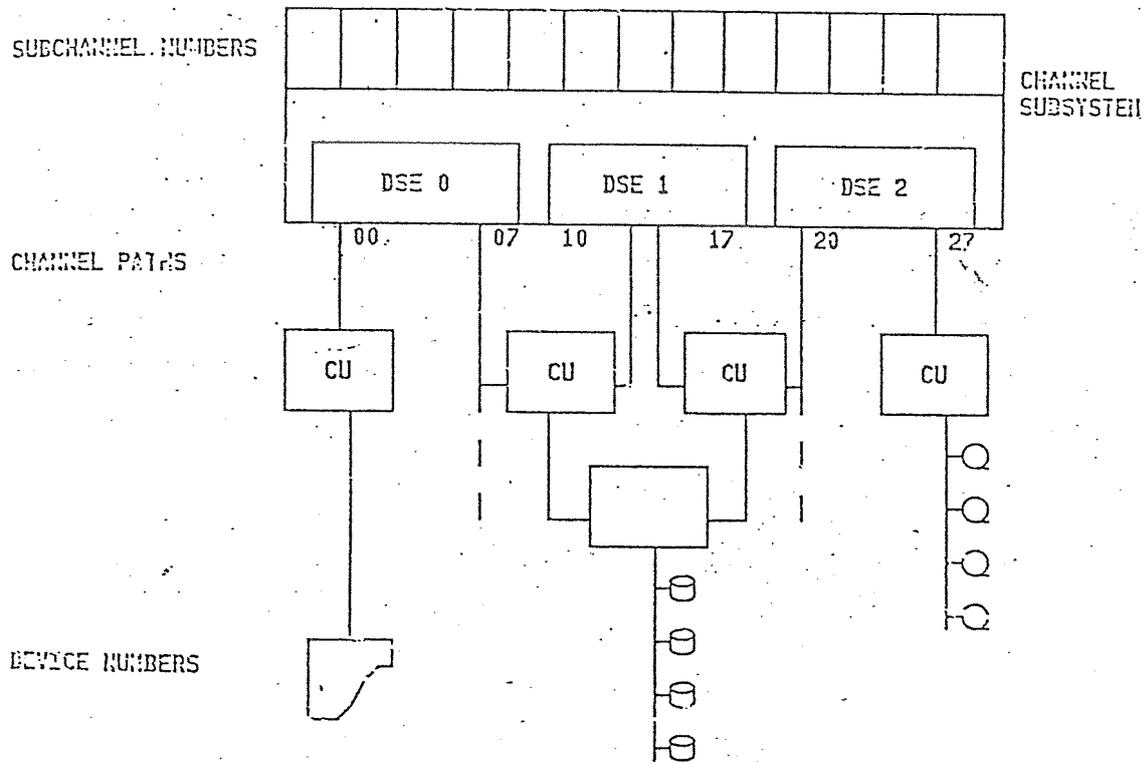


figura 25

CHANNEL PATHS



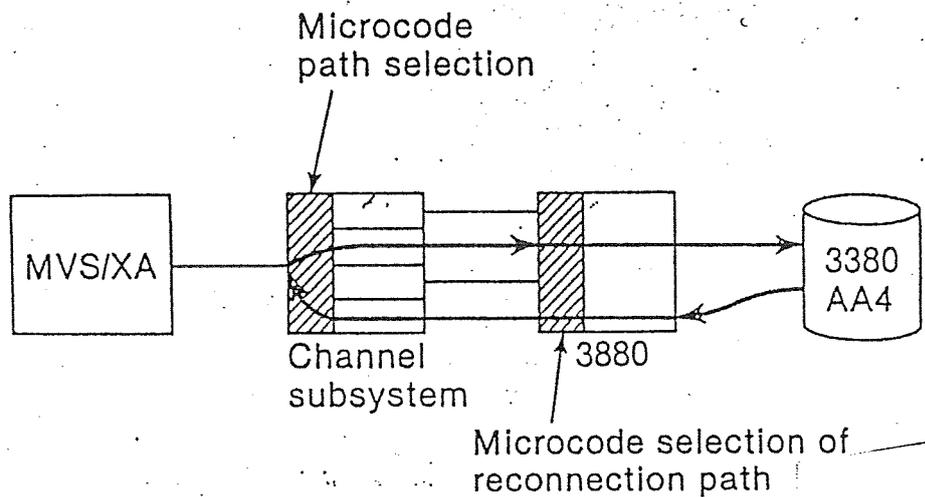
Riconnessione dinamica

Una Control Unit puo' connettersi a qualunque channel path. Cio' permette una maggiore probabilita' di connessione non appena il device e' pronto. Questa maggiore probabilita' garantisce una maggiore velocita' nel riconnettersi. L'effetto finale e' un miglioramento de:

- il tempo di risposta;
- il troughput del sistema.

figura 26

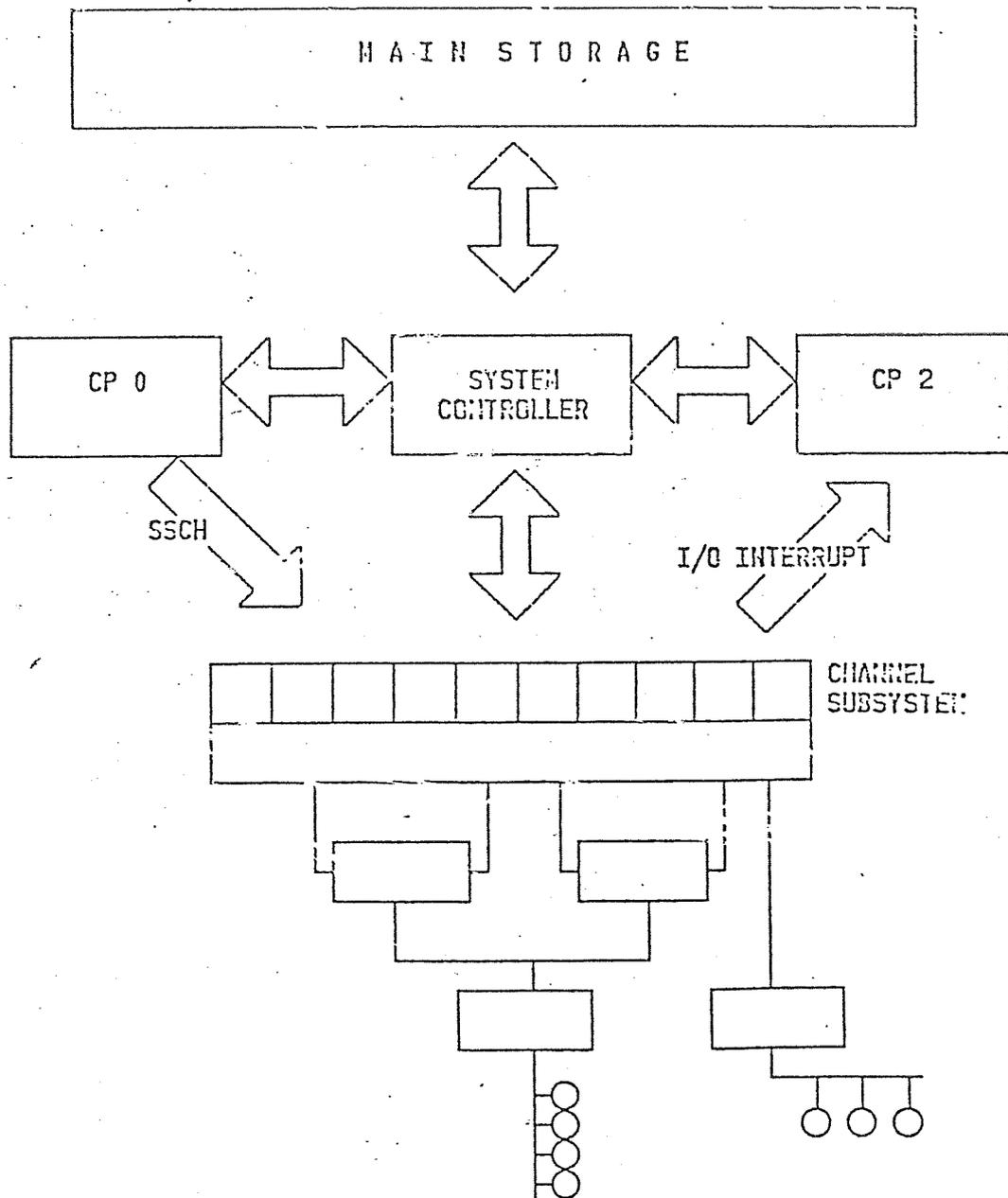
DYNAMIC PATH RECONNECTION



Nessuna affinita' tra C.P.U./Channel

Ciascun processor (mod. 200E o mod. 400E) puo' accedere a qualunque device e accettare qualunque I/O interrupt. Questa situazione si verifica quando concorrono due diverse C.P.U. sul medesimo I/O. Un I/O interrupt puo' interessare anche una C.P.U. che non aveva richiesto dati e se li vede tornare. Sara' cura del system controller provvedere e garantire l'esatto indirizzamento dei dati in arrivo.

figura 27



Channel Subsystem Monitor

Il EXDC presentandosi come un sistema a se stante consente una memorizzazione di dati che permettono una analisi sul funzionamento del channel subsystem attraverso il 'device timing' ed il 'mesurement data'.

Il device timing e' un dato che riporta l'elapsed time richiesto per eseguire le varie funzioni di I/O.

Il mesurement data colleziona statistiche di I/O per ogni operazione eseguita:

- Channel path utilization;
- Device connect time.

Il channel path utilization fornisce dati sull'uso dei path, mentre il device connect time fornisce informazioni sui tempi di connessione necessari a trasferire i dati.

**VECTOR FACILITY
CENNI**

Vector Facility(VF)

La Vector Facility dei sistemi 3090 e' una pipeline arithmetic unit (opzionale), aggiunta a ciascuno dei processori centrali che provvede ad un incremento sostanziale della performance per piu' applicazioni scientifiche e ingegneristiche. Ad oggi due tipi di Vector Facility potevano essere considerate. Una prevedeva l'integrazione dell'unita' vettoriale con il processore centrale; l'altra prevedeva una unita' vettoriale indipendente con memoria centrale sharing con il processore centrale.

L'implementazione che accedesse tutti gli operandi direttamente dalla memoria centrale era considerata cattiva e respinta in favore di una architettura basata su un set di registri vettore.

La 3090 Vector Facility e' integrata nella architettura XA che prevede 171 nuove istruzioni vettoriali che sono comprese nell'Instruction stream di ciascun processore centrale (CP) con vector facility. La VF e' separata nel senso che e' implementata usando un set di registri vettoriali e pipelined arithmetic unit su una board separata. La board puo' contenere due vector facility, una per ciascun Central Processor in un modello 200. Cio' significa che la Vector Facility puo' essere offerta come upgrade.

L'istruzione vettoriale esegue una singola istruzione per specificare che la stessa operazione aritmetica puo' essere calcolata su corrispondenti elementi del vettore operandi. Per esempio l'addizione di vettori A1, A2, A3, ..., An e B1, B2, B3, ..., Bn implica formazioni di somma $A1 + B1, A2 + B2, A3 + B3, \dots, An + Bn$.

I registri vettoriali (VRs) sono 16 ciascuno con 128 elementi di 32 bits cadauno. I VRs possono anche essere utilizzati in forma 8 registri ciascuno di 128 elementi con 64 bits (doppia precisione).

Un Vector Facility e' connessa sia per le istruzioni che per le esecuzioni al Central Processor e non presenta connessione diretta con la memoria centrale. L'Instruction Element preleva e decodifica tutte le istruzioni incluse quelle vettoriali dall'Instruction stream.

figura 28

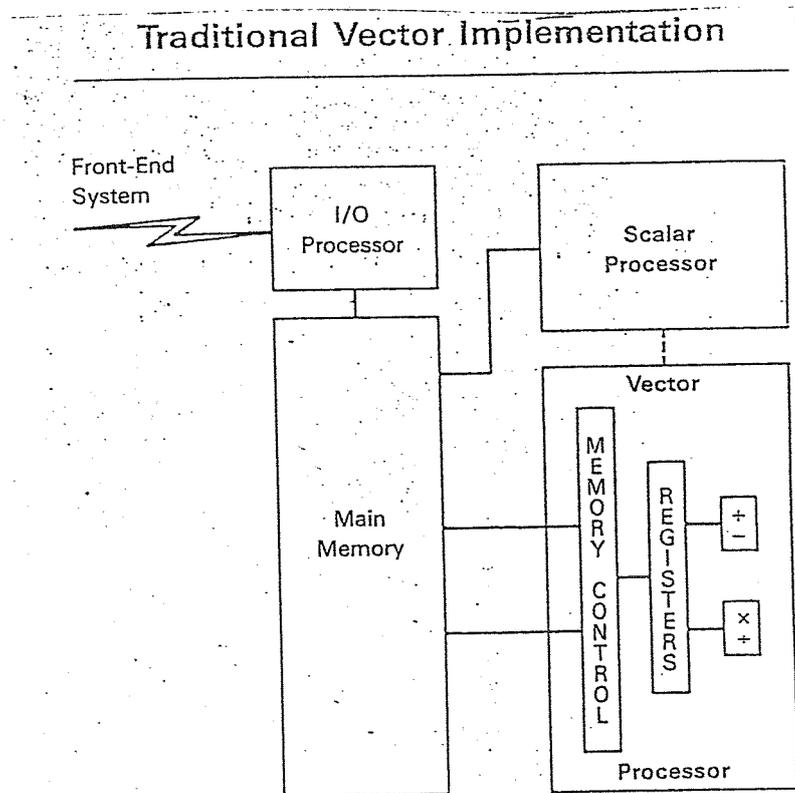
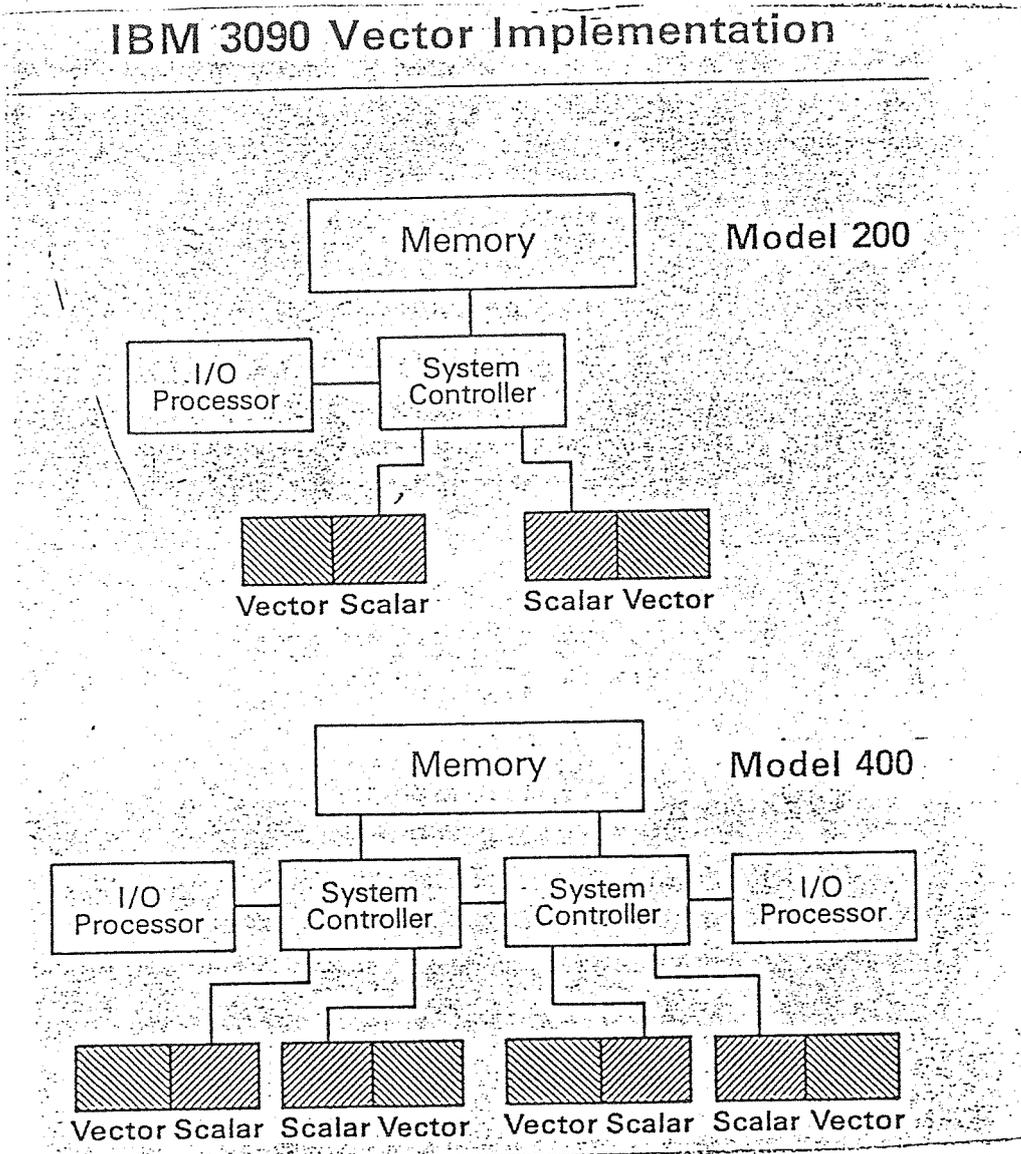


figura 29



Dettagli tecnici:

- Estensione opzionale di ciascun processore.
- Disponibile su ogni singola unita' di elaborazione.
- Aggiunge 171 istruzioni vettoriali.
- Dispone di 16 registri vettoriali (128 el. di 32 bit).
- Indirizzamento contiguo, non contiguo, random.
- Operazioni composte: $x \& +$ e $x \& -$ ad ogni ciclo.
- Prestazioni migliori fino a circa 7 volte quelle scalari.

