



Consiglio Nazionale delle Ricerche

**ISTITUTO DI ELABORAZIONE
DELLA INFORMAZIONE**

PISA

RICOGNIZIONE DI SISTEMI DISTRIBUITI IN UNA
CLASSIFICAZIONE GENERALE DI SISTEMI DI ELA-
BORAZIONE

F. Rabitti

Nota Interna B79-29

Dicembre 1979

RICOGNIZIONE DI SISTEMI DISTRIBUITI IN UNA
CLASSIFICAZIONE GENERALE DI SISTEMI DI
ELABORAZIONE

Dr. FAUSTO RABITTI

Consiglio Nazionale delle Ricerche

Istituto di Elaborazione dell'Informazione

Via S. Maria, 46 - 56100 PISA

RIASSUNTO:

Per prima cosa, si tenta di dare una definizione generale delle caratteristiche principali che un sistema di elaborazione deve possedere per potersi definire distribuito.

Poi si passa ad una rassegna delle motivazioni di ordine tecnico-economico, soprattutto in conseguenza dello affermarsi della tecnologia LSI, e di ordine strutturale, soprattutto riguardo alla flessibilità dell'organizzazione interna risultante.

Quindi si passa in rassegna una classificazione generale delle strutture architettoniche dei sistemi di elaborazione realizzati, oppure soltanto pronostici sperimentalmente, per reperire in essa le tendenze effettive di realizzazione dei sistemi a struttura distribuita (nei diversi tipi riconosciuti: sistemi con elevato parallelismo interno, sistemi multi-processors, sistemi a struttura interconnessa, a rete locale o geograficamente distribuita, ecc.).

Infine si tenta un bilancio sulla possibilità di trovare una prospettiva unificante per i sistemi distribuiti, alla luce sia dei risultati della rassegna delle motivazioni dei sistemi distribuiti, sia dei risultati della rassegna delle realizzazioni di sistemi ad architettura distribuita, nell'ambito dei sistemi di elaborazione conosciuti.

TITOLO:

RICOGNIZIONE DI "SISTEMI DISTRIBUITI" IN UNA CLASSIFICAZIONE
GENERALE DI SISTEMI DI ELABORAZIONE

° Attualità del dibattito sui "Sistemi Distribuiti"

La tendenza verso i Sistemi Distribuiti é ormai largamente accettata e condivisa, nel campo dei Sistemi di Elaborazione, per svariatissimi settori di applicazione. Inoltre l'affermarsi dei Sistemi Distribuiti non ha prodotto solo ripercussioni sul piano commerciale, ma si può dire abbia portato, e porterà ancora nel futuro, radicali cambiamenti nel modo stesso di concepire e di gestire i Sistemi di Elaborazione, e di organizzare le relative applicazioni.

° Definizione informale di Sistemi Distribuiti

Il termine di "Sistemi Distribuiti" sta più ad indicare una linea di tendenza, un orientamento nelle concezioni dei "Sistemi di Elaborazione", che una classe ben determinata e definita di architetture di sistemi.

L'attributo di "distribuito" ai sistemi si contrappone a quello, considerato tradizionale, di "centralizzato"; tuttavia non è possibile indicare una precisa frontiera di separazione tra questi due tipi generali di sistemi. Non é possibile precisare con chiarezza che cos'è un Sistema Distribuito, definendo caratteristiche che un Sistema di Elaborazione deve avere per poter essere considerato "distribuito" anziché centralizzato. Infatti queste due tendenze si contrappongono e si affrontano ai diversi livelli di strutturazione dei Sistemi di Elaborazione, nei momenti diversi della loro concezione e nelle diverse sezioni della loro composizione.

Si può tuttavia puntualizzare una importante caratteristica di principio che distingue i Sistemi Distribuiti: essa consiste nella autonomia di molteplici elementi (unità) funzionali e nella natura

fortemente cooperativa della loro comunicazione per la funzionalità complessiva del sistema integrato. (5)

° Si può procedere ulteriormente nella caratterizzazione dei Sistemi Distribuiti? - Cosa si può dire di più a questo proposito?

Si possono, a questo proposito, porre importanti quesiti riguardo la possibilità e la significatività di una visione complessiva dei Sistemi Distribuiti.

Ci si può porre il problema di trovare una caratterizzazione comune, più profonda e più specifica, su aspetti ben delineati, dei Sistemi Distribuiti; d'altra parte ci si può porre il problema di riportare ad una problematica unificante le diverse realizzazioni genericamente ricondotte al campo dei Sistemi Distribuiti. In fase preliminare ci si può chiedere se queste due serie di problematiche possono avere risposte positive.

Obiettivo di questo lavoro consiste nel cercare risposte a questi quesiti.

A questo scopo saranno esaminate prima le ragioni che portano alla affermazione dei Sistemi Distribuiti; successivamente saranno esaminate le realizzazioni concrete di Sistemi Distribuiti, nell'ambito di una più vasta panoramica delle Architetture dei Sistemi di Elaborazione.

PARTE PRIMA

° Cause del successo della tendenza verso i Sistemi Distribuiti

Le ragioni dell'affermarsi della tendenza alla distribuzione nei Sistemi di Elaborazione sono soprattutto di due ordini: economico e funzionale. Inoltre si può dire che questi due ordini di ragioni, come si vedrà, risultano strettamente correlati.

° Cause di ordine funzionale (dal punto di vista interno al sistema)

Sul piano funzionale si può applicare, nel campo dei Sistemi di Elaborazione, la legge di Parkinson relativa alle grandi organizzazioni centralizzate: "oltre certe dimensioni, l'organizzazione impiega risorse sempre maggiori, in un rapporto che si può rivelare antieconomico, per controllare o gestire se stessa, invece di adempiere al proprio compito istituzionale!" (1)

Ribortando questo principio al campo dei computers, si può dire che, oltre certe dimensioni del sistema centralizzato, il sistema operativo che lo governa diventa sempre meno efficiente.

"Si sono infatti segnalati casi in cui addirittura il 95% del tempo dell'unità centrale veniva destinato al sistema operativo (attività gestionale interna), cioè a controllare ed a governare le operazioni delle varie unità hardware e software, sicchè rimaneva solo il 5% del tempo-macchina per i calcoli specifici richiesti dai programmi in esecuzione (attività operativa)". (1)

Viceversa, in un Sistema Distribuito si può frazionare il peso della gestione totale del Sistema di Elaborazione, ripartendola fra le diverse Unità Funzionali, ciascuna dimensionata in modo da sostenere il sistema operativo necessario alle risorse della singola unità ed alle richieste di comunicazione e cooperazione reciproca, per la funzionalità globale del sistema.

° Cause di ordine funzionale (dal punto di vista dell'utente)

Esistono poi esigenze di razionalizzazione dei servizi sempre più complessi richiesti dall'utente, che portano alla distribuzione delle funzioni dei sistemi, intesi globalmente.

Infatti la pressione degli utenti per ottenere sempre più risorse di calcolo, nella forma di una maggiore varietà di servizi e di accresciute possibilità di accesso ai dati, può condurre, ad esempio, alla interconnessione di singoli sistemi di calcolo in una Rete di Elaboratori (Computer Network), che può essere vista come un unico sistema globale (esempio di Sistema Distribuito) i cui singoli componenti sono spazialmente distribuiti.

La necessità di rendere più razionale la distribuzione delle risorse di calcolo in un sistema distribuito globale, per aumentare le prestazioni rispetto ai costi, conduce a localizzare presso l'utente Unità di Elaborazione capaci di interagire con altre Unità lontane, mediante trasmissione di dati a distanza, e quindi loro elaborazione (teleprocessing).

I singoli componenti di tali sistemi distribuiti, secondo l'esigenza dell'utente, ... (1) ... "dispongono di prestazioni diversificate e complementari: essi sono organizzati in modo autonomo per applicazioni dedicate a specifici settori e nello stesso tempo sono in grado di rendere le varie risorse mutualmente disponibili per raggiungere obiettivi più generali (general purpose) da ricercare non tanto in singole localizzazioni ma nella totalità della stessa Rete (sistema distribuito globale)".

Tuttavia si deve rilevare, a questo punto, che le esigenze, sopra motivate, di ordine funzionale (sia dal punto di vista interno al sistema che dal punto di vista dell'utente) di riorganizzazione su base distribuita dei Sistemi di Elaborazione è stata resa effettiva (cioè conveniente, e quindi realizzabile in pratica, dal punto di vista dell'ottimizzazione del rapporto performance/costo) dagli enormi cambiamenti portati dalla tecnologia LSI in campo economico.

° Cause di ordine economico

Le cause economiche costituiscono l'altro ordine di ragioni dell'affermarsi della tendenza ai Sistemi Distribuiti.

Sul piano economico un'influenza determinante, nella direzione dell'affermarsi di questa tendenza, hanno avuto i recenti sviluppi dei componenti LSI, integrati a larga scala, che permettono la produzione di massa di "moduli di elaboratore" (computer modules), formati da un processor più una memoria di dimensioni moderate, integrati su di un singolo "chip". Tali componenti, pur di complessità crescente, sono prodotti a prezzi fortemente decrescenti.

Questi sviluppi tecnologici hanno cambiato profondamente il modo di concepire le architetture dei Sistemi di Elaborazione: infatti questi "moduli di elaboratore" sono usati come blocchi costitutivi (building blocks) altamente complessi per sistemi a cui gli utenti richiedono una funzionalità via via più avanzata e sofisticata.

Come conseguenza di queste innovazioni tecnologiche si può affermare che ora ... (C. G. Bell, R. Chen, S. Rege: "Effect of technology on near term computer structure" - Computer, 1972)... "non è più vera l'affermazione secondo cui al crescere della dimensione di un sistema la potenza di calcolo aumenta più rapidamente del costo solo quando il Sistema di Elaborazione è centralizzato".

°Esame di una serie di ragioni tecniche che hanno portato a specifiche realizzazioni nell'ambito dei sistemi distribuiti

Si sono esaminate precedentemente le ragioni di ordine funzionale ed economico, su di un piano generale, che giustificano la tendenza complessiva verso una strutturazione distribuita dei sistemi di elaborazione.

Prenderemo ora in esame una serie di ragioni, a livello meno generale ma tecnicamente più specifico, che hanno portato alla realizzazione ed allo sviluppo di una importante classe di sistemi appartenente all'ambito dei sistemi distribuiti: le reti di elaboratori (Computer Networks).

Tali ragioni possono essere facilmente generalizzate ed adattate a sistemi di elaborazione a struttura distribuita, al di là degli aspetti specifici riguardanti le reti di elaboratori, che costituiscono una classe particolare dei sistemi distribuiti.

Le reti di comunicazione di calcolatori (Computer Communication Networks) sono composte da un certo numero di calcolatori ospiti (hosts), che forniscono i servizi all'utente, e da una sottorete di comunicazione (subnetwork), che provvede alle comunicazioni a distanza tra i vari calcolatori ospiti. (9)

L'insieme risultante può essere anche considerato globalmente come un unico sistema complessivo, le cui unità funzionali di elaborazione sono spazialmente distribuite, avente lo scopo di permettere la condivisione di risorse-hardware, risorse-software e risorse-dati. Alcune di queste risorse possono essere uniche all'interno del sistema distribuito globale. Tutto questo prescinde dal fatto che i singoli calcolatori ospiti siano di tipi, caratteristiche e destinazioni diverse.

Esponiamo ora la serie di ragioni specifiche relative particolarmente ai sistemi "Computer Network". (2)

- 1) Suddivisione e bilanciamento del carico di lavoro complessivo (Load sharing).

Un problema (programmi e dati) iniziato in un computer ospite che è

temporaneamente sovraccarico è inviato ad un altro per l'elaborazione. Chiaramente il costo del trasbordo deve essere minore del costo del ritardo prevedibile per poter effettuare l'elaborazione nello stesso ospite. Inoltre questo procedimento implica caratteristiche altamente simili tra gli ospiti della rete interessati.

2) Condivisione dei dati (Data sharing).

Un programma è inviato per l'esecuzione a quell'ospite della rete ove può aver accesso ad un vasto "data base", specializzato per l'applicazione richiesta. Naturalmente deve essere meno costoso, in questo caso, trasferire il programma ai dati piuttosto che trasferire i dati al programma.

3) Condivisione dei programmi (Program sharing).

I dati sono inviati ad un ospite che ha programmi specializzati per la applicazione richiesta. Questo può accadere a causa della dimensione del programma (per cui è meno costoso trasferire i dati al programma piuttosto che il programma ai dati) ma anche a causa del fatto che il patrimonio di conoscenze necessarie all'esecuzione corretta del programma (ad esempio per ciò che concerne le procedure di inizializzazione e di trattamento degli errori) è disponibile presso un'installazione di un'ospite che presso un'altra.

4) Dispositivi specializzati (Specialized facilities).

Nell'ambito della rete occorre esistano, presso alcuni ospiti, solamente pochi dispositivi di tipi altamente specializzati e raramente usati, quali memorie ad accesso casuale molto vaste, speciali apparecchi di display, processors con memorie associative o "special purpose array processors" (processors strutturati per il trattamento parallelo di dati in forma opportuna).

5) Scambio di messaggi (Message switching).

Il compito delle comunicazioni reciproche nell'ambito della rete assume una tale importanza da giustificare lo sviluppo dei protocolli più sofis-

sticati di commutazione e controllo messaggi.

6) Affidabilità (Reliability).

Nel caso di guasto di alcune componenti della rete (sia computer ospiti che sottoinsieme della sottorete di comunicazione), altre componenti possono essere usate al loro posto, permettendo così una degradazione parziale, controllata ed incrementale (Graceful degradation) del sistema complessivo.

A tale proposito la sottorete di comunicazione può essere disegnata senza nessuna componente centrale critica; ciò permette alle componenti non interessate dal guasto, di determinare e diagnosticare la natura del guasto stesso, escludendo poi le parti non più funzionali dalla configurazione del sistema complessivo. Viene così recuperata una situazione di corretta funzionalità del sistema globale riconfigurato, con solamente una diminuzione della potenza di calcolo totale del sistema.

7) Potenza di picco del sistema (Peak computer power).

Vaste sezioni del sistema complessivo possono essere dedicate, per brevi periodi, ad un singolo compito se, per la esecuzione di questo occorre soddisfare importanti adempimenti in tempo reale.

Ciò dipende dalla possibilità di frazionare quel compito in sottocompiti paralleli.

8) "Multiplexing" delle comunicazioni (Communication multiplexing).

Un uso efficiente delle possibilità di comunicazione è ottenuto attraverso il multiplexing di un certo numero di attività che richiedono uno scambio di dati a bassa velocità.

9) Miglior livello qualitativo delle comunicazioni (Better communication).

Una comunità di utenti (ad esempio una comunità scientifica che lavora ad un progetto comune), attraverso un uso concorrente, anche a distanza, degli stessi programmi, delle stesse basi di dati e attraverso una comu-

nicazione diretta riguardante il progetto comune (non solo tramite lo scambio, attraverso messaggi sulla rete, di rapporti informativi, ma nel contesto delle possibilità di uso mutuale di strumenti comuni), può diventare una comunità più produttiva, con minore duplicazione del lavoro ed una più pronta comunicazione dei risultati.

10) Migliore distribuzione del carico di lavoro al sistema tramite la pre-elaborazione (Better load distribution through preprocessing).

Alcuni lavori richiedono una comunicazione di dati ad alta velocità con un particolare computer ospite della rete. Mediante una pre-elaborazione del lavoro in un più piccolo computer, ospite della rete, è possibile ridurre la portata e la velocità del flusso di informazioni che devono essere inviate per l'elaborazione conclusiva a quel particolare più potente computer della rete.

Le Reti di Elaboratori non sono certamente la sola classe di sistemi ad architettura o conformazione complessiva distribuita di cui si contano numerosi esempi attualmente realizzati ed a cui fanno capo studi e ricerche specializzati. Esistono altre classi molto importanti, quali ad esempio i "Multiprocessors", che si distinguono dalle reti per il fatto che, nei primi, due o più unità di processo, sotto controllo integrato, condividono una memoria logicamente comune, mentre nelle seconde le intercomunicazioni tra le unità di processo sono effettuate tramite messaggi a protocollo prestabilito.

In ogni modo, un esame delle varie motivazioni tecniche, analogo a quello effettuato per il caso delle "Computer Networks", condotto anche per le altre classi di sistemi distribuiti, quali ad esempio i "Multiprocessors", porterebbe a giustificare una importante osservazione: e cioè che, al di là di accentuazioni e terminologie specifiche, dovute al particolare ambito cui sono applicate, le ragioni tecniche che motivano le diverse e specifiche realizzazioni di sistemi distribuiti possono ricondursi a tipologie comuni o comunque simili.

° Verifica del mutamento di prospettive, per i sistemi distribuiti, prodotto dalle innovazioni tecnologiche.

Si è affermato, nei paragrafi precedenti, che l'approccio distribuito alla strutturazione dei Sistemi di Elaborazione dell'Informazione è diventato efficace e conveniente in seguito alle innovazioni introdotte nel settore tecnologico negli ultimissimi anni.

Situazione fino al 1973 per i sistemi distribuiti.

A conferma del fatto che, solo pochi anni fa, la distribuzione dell'elaborazione fosse non conveniente nella grande maggioranza dei casi, si può citare uno studio del 1973. (18)

Le ipotesi su cui si fonda questo studio riflettono i presupposti tecnologici su cui si basava, in quel periodo, la realizzazione dei sistemi di elaborazione.

Le due ipotesi principali sono: (7)

- 1) la memoria centrale deve avere alti coefficienti di utilizzazione;
- 2) l'ottimizzazione del rapporto prestazioni-costi si ottiene con la saturazione delle unità di calcolo (CPU) a scapito di una bassa utilizzazione dei sottosistemi di comunicazione; questo in base alla considerazione che il rapporto tra i costi di elaborazione ed i costi di comunicazione è di 9:1.

Sotto queste ipotesi è stato realizzato un modello il cui scopo è di valutare gli approcci "centralizzato" (C), "semi-centralizzato" (SC) e "distribuito" (D) per applicazioni miste (AM), applicazioni commerciali (AC), piccoli lavori scientifici (PLS) e grossi lavori scientifici (GLS). (7)

I risultati del modello possono essere così riassunti: (18)

- nel campo delle applicazioni commerciali (AC) il rapporto tra l'efficienza del sistema a struttura "semi-centralizzata" e l'efficienza del sistema a struttura "distribuita" è di 1.125:

$$E(SC)/E(D)=1.125$$

mentre il rapporto tra l'efficienza del sistema a struttura "centralizzata" e l'efficienza del sistema a struttura "distribuita" è di 0.625:

$$E(C)/E(D)=0.625$$

- analogamente, nel campo delle applicazioni miste (AM) si ha:

$$E(SC)/E(D)=2.25$$

$$E(C)/E(D)=2$$

- nel campo dei piccoli lavori scientifici (PLS) si ha:

$$E(SC)/E(D)=2.28$$

$$E(C)/E(D)=2.5$$

- nel campo dei grossi lavori scientifici (GLS) si ha:

$$E(SC)/E(D)=4.8$$

$$E(C)/E(D)=5$$

Questi risultati sono riassunti nel diagramma di Fig. 1.

Da questi risultati si può facilmente dedurre come, nelle ipotesi conseguenti alla situazione tecnologica nel campo dei componenti di sistemi fino al 1973, l'approccio architetturale-organizzativo "distribuito" sia penalizzato all'approccio "semi-centralizzato" e "centralizzato" in tutti i campi eccetto quello delle applicazioni commerciali, in un rapporto sfavorevole che varia da 1:2 fino a 1:5. Viceversa, in quest'ultimo campo (AC), l'efficienza dell'approccio "distribuito" è di poco inferiore a quella dell'approccio "semi-centralizzato" (1:1.125) mentre è già superiore a quella dell'approccio "centralizzato" (1.6:1).

Origini, motivazioni, implicazioni e prospettive della tecnologia LSI

Abbiamo ora un inciso sulle origini, le motivazioni, le implicazioni e le prospettive della tecnologia LSI. (13) (28) (30)

In seguito, alla luce di questi risultati, saranno riviste le ipotesi principali su cui si basava lo studio precedente (13), e sarà ipotizzato anche il modo in cui le stesse conclusioni dovrebbero cambiare, soprattutto considerando le linee di sviluppo consolidate nel settore. (25) (26)

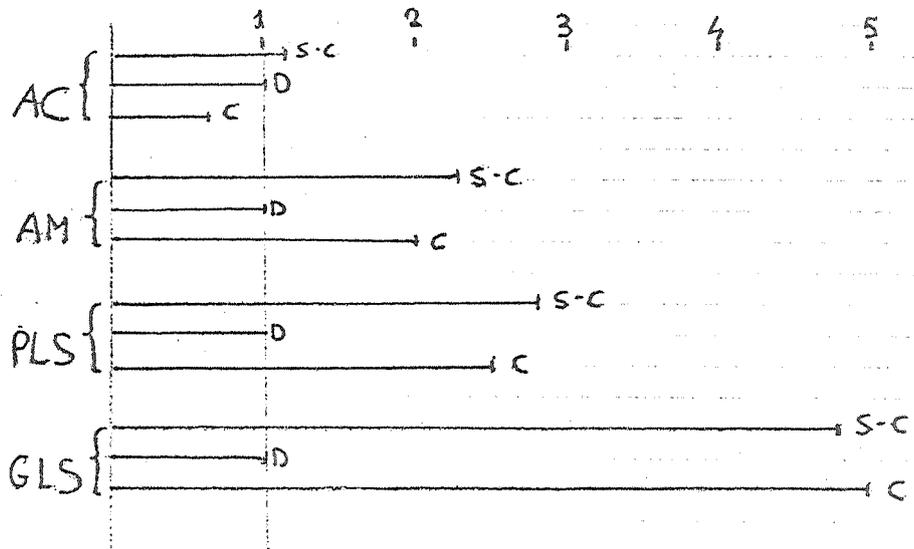


FIGURA 1

Rapporto tra l'approccio "centralizzato"(C), "semi-centra-
lizzato"(SC) rispetto l'approccio "distribuito"(D), per
applicazioni miste(AM), piccoli lavori scientifici(PLS),
grossi lavori scientifici(GLS) ed applicazioni commerciali(AM).

TECNOLOGIA	COMPONENTI ELEMENTARI	CIRCUITI	SOTTOSISTEMI	SISTEMI
COMPONENTI DISCRETI				
CIRCUITI INTEGRATI				
DISPOSITIVI FUNZIONALI LS				

FABBRICANTI DI SISTEMI

FABBRICANTI DI SEMICONDUTTORI

FIGURA 2

Livello logico del progetto di sistemi col progredire della tecnologia.

L'integrazione circuitale su larga scala (LSI = Large Scale Integration) ha origine dalle esigenze di miniaturizzazione per applicazioni specifiche, specialmente in campo spaziale. Tuttavia le implicazioni risultanti da queste tecniche su altri importanti fattori, quali la velocità, la dissipazione, l'affidabilità ed il costo, costituiscono ora motivazioni più significative di quella originaria. (34)

(13)-(23)-(27)-(30):

Una delle principali implicazioni della microelettronica è l'aumento della velocità, dovuto alla drastica riduzione dei percorsi geometrici ed alla minimizzazione delle impedenze parassite. Parallelamente alla disponibilità di elementi costitutivi più veloci, si procede verso una miniaturizzazione globale a livello di sistema.

Un altro fattore di cui si ottiene il miglioramento (cioè una diminuzione specifica) è la dissipazione di energia, in conseguenza della riduzione delle dimensioni e della ottimizzazione dei carichi.

Enormi progressi si sono ottenuti con la microelettronica nel campo della affidabilità (Reliability), tanto che alcune delle più significative realizzazioni tecniche dei nostri tempi (quali le applicazioni spaziali, i grandi elaboratori commerciali, ecc.) non sarebbero state ^{possibili} senza il salto di affidabilità introdotto dalla produzione di componenti integrati.

Tale miglioramento di affidabilità è dovuto principalmente al processo di fabbricazione "per insiemi" (Batch Processing) degli elementi LSI, il quale si svolge in condizioni ambientali rigorosamente controllabili e col minimo intervento umano. Altre ragioni sono costituite dalla riduzione degli effetti di degradazione dei dispositivi a semiconduttori, dovuti alla minore dissipazione specifica, e dalla drastica riduzione del numero delle saldature.

Oltre agli aspetti funzionali, il fattore decisivo per l'affermazione delle tecniche microelettroniche è quello economico.

Gli effetti della microelettronica sui costi sono essenzialmente di due ordini: uno riguarda la parte di fabbricazione dei componenti, l'altro la parte dell'applicazione della loro attività funzionale.

Per il primo aspetto, l'economia deriva fondamentalmente dal principio di

fabbricazione per insiemi su cui si basa la microelettronica. Esso comporta la realizzazione contemporanea non soltanto di grandi quantità di circuiti ma anche delle loro interconnessioni. Si noti a questo proposito che gli elementi circuitali costituiscono una parte non preponderante (dell'ordine del 20%) del costo totale dell'apparecchiatura; una buona parte del costo rimanente è dovuto alle sezioni di interconnessione degli elementi circuitali tra loro, mediante supporti stampati, connettori, cablaggi, ecc..

Per il secondo aspetto, basti considerare le conseguenze che l'aumento di affidabilità ha sui tempi di disponibilità delle macchine e sui costi di manutenzione.

L'importanza dei fattori precedentemente esposti ha fatto sì che le tecniche microelettroniche nel loro insieme rappresentino ora un orientamento fondamentale, anche per il futuro, nella tecnologia dell'elaboratore, con implicazione sulla logica e l'architettura dei sistemi. (30)

Lo schema di Fig. 2 (34) mostra infatti come l'ingegnere di sistemi abbia potuto innalzare il livello logico del proprio progetto col progredire della tecnologia dei semiconduttori. Quando la tecnologia era al livello dei componenti discreti (porte logiche, registri di memoria, ecc.) l'ingegnere di sistemi aveva a disposizione componenti elementari per la realizzazione dei circuiti di base, con cui si realizzavano poi le strutture funzionali più complesse. Con la tecnologia a livello dei primi circuiti integrati SSI (Small Scale Integration) e MSI (Medium Scale Integration), aveva già a disposizione i circuiti di base con cui poteva realizzare sottosistemi funzionali specifici. Con la tecnologia a livello degli attuali circuiti integrati LSI, l'ingegnere di sistemi ha ora sovente a disposizione (e questo si realizzerà e si sfrutterà più compiutamente in futuro) dei veri e propri dispositivi funzionali, su di un singolo "chip", già specializzati oppure specializzabili, che costituiscono effettivi sottosistemi funzionali con cui può realizzare i sistemi nel loro complesso.

Occorre tuttavia prendere con una certa cautela i risultati precedentemente esposti (30), nel senso che alcuni di essi sono piuttosto proiettati in -

una prospettiva futura che realizzati nella situazione presente: cioè non è detto che la strada verso una maggiore integrazione dei dispositivi semiconduttori conduca ogni volta a prodotti che sono contemporaneamente più veloci, più affidabili, meno costosi, con minore dissinazione di energia e maggiormente integrabili.

Per esempio, attualmente i circuiti logici unipolari MOS (Metallo-Ossido-Silicio) sono più altamente integrabili rispetto a quelli bipolari, mentre sono svantaggiati riguardo le possibili velocità di funzionamento: i più veloci tra i circuiti MOS, quelli di tipo C-MOS, hanno tempi di commutazione superiori a 15 ns., mentre i più veloci tra i circuiti bipolari, quelli di tipo TTL, hanno tempi di commutazione fino a 1 ns.!

Concludendo si può dire che i risultati tecnologici, esposti in questo inciso sugli effetti della microelettronica, vengono a ricercarsi principalmente sugli indici di costo e di prestazione delle unità di memoria e di processo.

Per quantificare questa influenza viene preso in considerazione un indice, riferito a quanto di meglio è prodotto in ogni singola epoca, che si ottiene, per quanto riguarda i dispositivi di memoria, moltiplicando il prezzo di un singolo bit per il suo tempo di accesso; mentre, per quanto riguarda i dispositivi logici, si ottiene moltiplicando il prezzo di una porta logica per il suo ritardo (le unità di misura assunte sono il dollaro USA, per i prezzi, e il nanosecondo, per i tempi). (34) (30)

Il diagramma di Fig. 3, unitamente alla tabella seguente, indica come tale indice sia diminuito di più di un ordine di grandezza (quasi 2 per quanto concerne i dispositivi logici) nel periodo 1965-1970, e abbia poi continuato a diminuire, di quasi un ordine di grandezza, nel periodo 1970-1975. (34)

Valori indice

Anni considerati:	1965	1970	1975
Dispositivi di memoria:	117.5	4.26	0.499
Dispositivi logici:	23.17	0.34	0.04

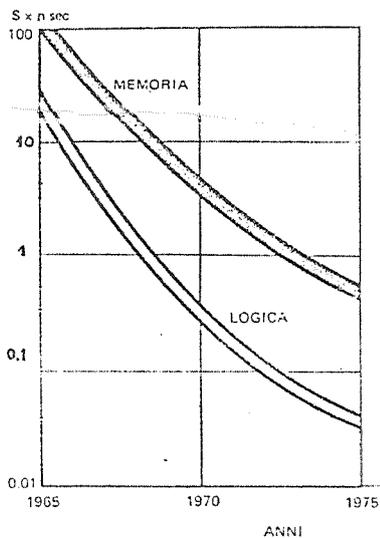


FIGURA 3

Andamento dell'indice prezzoXprestazioni nel periodo 1965-1975 per dispositivi di memoria e dispositivi logici.

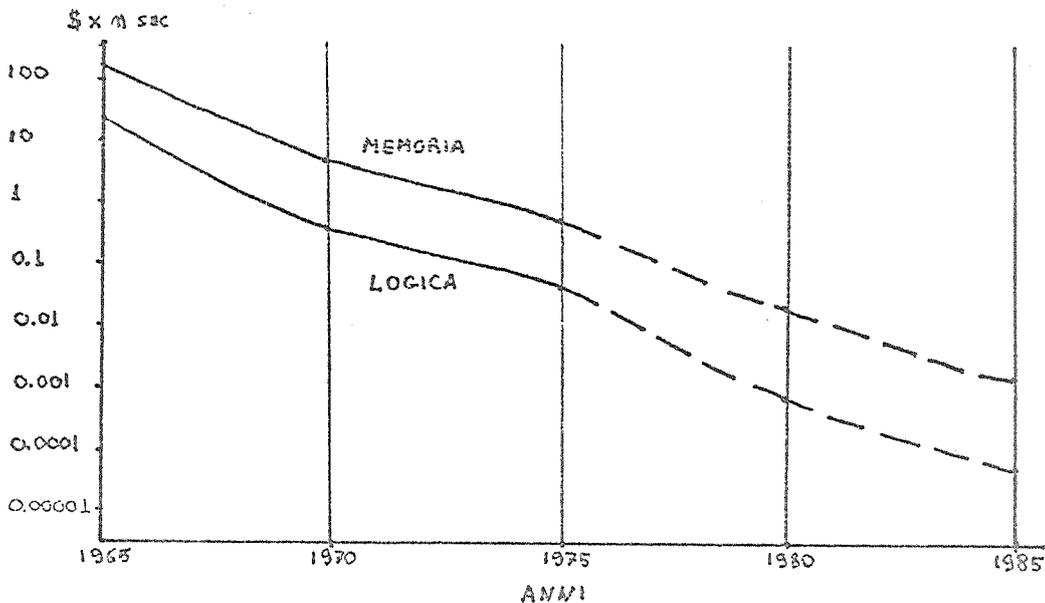


FIGURA 4

Evoluzione dell'indice prezzoXprestazioni realizzata nel periodo 1965-1975 ed ipotizzata per il periodo 1975-1985 per i dispositivi logici e di memoria.

Possiamo quindi cercare di individuare l'andamento futuro di questi due indici (indice riferito ai dispositivi di memoria e indice riferito ai dispositivi logici) per verificare se la tendenza rilevata nei periodi 1965-1970 e 1970-1975 trovi o meno conferma nei periodi 1975-1980 e 1980-1985. (28)

Per quanto riguarda le memorie integrate MOS (in particolare MOS dinamiche), nella tabella seguente viene mostrata l'evoluzione del costo per bit (in centesimi di dollaro USA per bit) e del tempo di accesso (in nanosecondi, $1 \text{ ns.} = 10^{-9} \text{ sec.}$) e quindi dell'indice risultante, a partire dalla situazione esistente per il 1975 e poi secondo le previsioni per il 1980 e il 1985.

Epoca:	Costo(¢/bit):	Tempo accesso(ns.):	Valore indice:
1975	0.2	250	0.5
1980	0.085	80	0.02
1985	0.003	50	0.0015

Si noti che il valore dell'indice per il 1975 in questa tabella coincide con quello espresso nella tabella precedente.

Analogamente, per quanto riguarda i circuiti integrati MOS, in particolare C-MOS (Complementary-MOS), nella tabella seguente viene mostrata l'evoluzione del costo per porta logica (in centesimi di dollaro USA per porta) e del ritardo (in nanosecondi) e quindi dell'indice risultante, a partire dalla situazione esistente per il 1975 e poi secondo le previsioni per il 1980 e il 1985.

Epoca:	Costo(¢/porta):	Ritardo(ns.):	Valore indice:
1975	0.13	22	0.04
1980	0.024	3	0.00072
1985	0.005	1	0.00005

* Si noti, anche qui, come il valore dell'indice per il 1975 coincida con quello della tabella precedente, per i dispositivi logici.

Si possono quindi compattare questi risultati in un unico diagramma complessivo (Fig. 4) che riassume, per gli indici dei dispositivi logici e di memoria, l'evoluzione realizzata nel periodo 1965-1975, e l'evoluzione ipotizzata nel periodo 1975-1985.

Si noti che la scelta dei circuiti MOS è quella che massimizza i valori dell'indice, relativamente ai vari tipi di circuiti esistenti; quindi i risultati di Fig. 3 possono essere considerati omogenei coi risultati complessivi di Fig. 4 (si ricordi che, in riferimento alla Fig. 3, si era affermato che gli indici erano riferiti a quanto di meglio, in rapporto alla massimizzazione di tali indici, era possibile ottenere dalle varie tecniche circuitali, sia MOS che binolari).

Dal diagramma di Fig. 4, si trae la conferma che, anche in proiezione futura, gli indici dovrebbero continuare a diminuire, perfino accentuando questa tendenza: infatti sia nel periodo 1975-1980 che nel periodo 1980-1985 si prevede che tali indici diminuiscano di più di un ordine di grandezza (quasi due, per quel che concerne i dispositivi logici, nel periodo 1975-1980).

Occorre cautelarsi, tuttavia, nella interpretazione rigorosa di questi dati, soprattutto in proiezione futura, nel senso che può succedere che certe tendenze accentuate di riduzione di costi siano rese meno pronunciate da complesse questioni di politica di mercato e di genere commerciale.

Comunque questi risultati riassumono in chiaro, nella loro essenza, le conclusioni che si possono trarre da tutti i discorsi precedenti sulle conseguenze e le prospettive conseguenti allo sviluppo della tecnologia LSI.

Situazione attuale e prospettive future per i sistemi distribuiti.

Alla luce di questi risultati, si può riprendere in esame lo studio precedente sulla convenienza della distribuzione dell'elaborazione, per contestarne le relative conclusioni, rapportate al momento attuale ed al futuro. (31)

In particolare possono essere praticamente rovesciate le due ipotesi principali. (13)

La prima ipotesi, consistente nella necessità di avere alti coefficienti di utilizzazione della memoria centrale, si basava sul fatto che nei sistemi di calcolo si poteva contare di avere a disposizione una memoria centrale costosa, relativamente al costo totale dell'hardware del sistema, e quindi di capacità oscillante da 250 K-bytes, fino a non più di 1 M-bytes ($1K=1024$; $1M=1024K=1048576$).

Dagli studi precedenti, si può constatare che ora, allo stesso costo, è possibile avere una memoria integrata LSI (di tipo MOS), con prestazioni circa equivalenti e con dimensioni circa dieci volte maggiori.

Sempre dagli studi di previsione, compendiate nel diagramma di Fig. 4, è possibile prevedere un'analogia, e forse più accentuata, tendenza anche per il futuro: in teoria si potrebbe avere a disposizione, entro i prossimi cinque anni, una memoria centrale di capacità ancora dieci volte maggiore (oppure con prestazioni, in teoria, dieci volte maggiori, per sistemi particolari) a costi analoghi.

Chiaramente, una effettiva utilizzazione di memorie centrali di tali dimensioni presupporrebbe una diversa organizzazione del sistema complessivo.

Infatti è impensabile una gestione efficiente di tali memorie secondo schemi tradizionali: un'unica grande memoria centrale, direttamente accessibile da un'unica potente unità centrale di elaborazione, comporta, da una parte, problemi per trovare un tipo di gestione tale da poter trarre vantaggio effettivo dalla dimensione della memoria disponibile; d'altra parte, comporta problemi di costi per i circuiti di pilotamento di una tale memoria.

In ogni modo, come conseguenza del nuovo ordine di dimensioni delle memorie centrali disponibili, soprattutto in prospettiva, si ha che non è più una condizione stringente avere alti coefficienti di utilizzazione di tali memorie centrali, rovesciando così la prima ipotesi del lavoro precedente. (16)

Anche la seconda ipotesi, consistente nella necessità di saturazione delle unità di calcolo, in rapporto ai sottosistemi di comunicazione, in base al rapporto 3:1 tra i costi di elaborazione ed i costi di trasmissione, deve essere rivista in base a considerazioni sulle unità di processo analoghe a quel-

le esposte riguardo le unità di memoria.

Dai diagrammi illustrati precedentemente si ricava una precisa tendenza nella produzione dei circuiti logici: oggi si possono avere, allo stesso costo precedente, unità di processo con circuiti integrati LSI, di potenza logica dieci volte maggiore e con velocità simile (oppure con velocità dieci volte maggiore e con potenza simile, per applicazioni particolari); e per il prossimo futuro è previsto (si veda il diagramma di Fig. 4) un accentuarsi di questa tendenza, con un fattore di incremento decisamente superiore a dieci, (incremento in potenza logica oppure in velocità, a costi essenzialmente costanti).

D'altra parte, non essendo stato finora possibile e non essendo prevedibile per il futuro un analogo decremento dei costi dei sottosistemi di comunicazione, oppure un corrispondente incremento di prestazioni (anzi, dati certi fattori costrittivi, la situazione è rimasta e rimarrà nel complesso piuttosto costante), il rapporto tra i costi di computazione ed i costi di trasmissione risulta completamente alterato, rispetto all'ipotesi precedente di 9:1.

Ovvi tale rapporto si può considerare all'incirca equilibrato (1:1), mentre per il futuro la situazione tenderà a ribaltarsi completamente (verso il rapporto 1:10 tra costi di elaborazione e costi di trasmissione nel prossimo quinquennio).

Si ha così un rovesciamento di posizioni anche per quanto riguarda la seconda ipotesi su cui si basava il modello per lo studio del '73, sulla convenienza della distribuzione dell'elaborazione. (16)

Dato che un'analisi delle tendenze evolutive tende a rendere definitive ed a sottolineare il rovesciamento di queste ipotesi, si può affermare che anche le conclusioni di questo studio (16), e cioè che l'approccio architetturale-organizzativo "distribuito" risulta notevolmente meno efficiente in quasi tutti i campi (eccetto quello commerciale) rispetto all'approccio "semi-centralizzato" e "centralizzato", si possono considerare ormai non più valide.

Anzi, pur non disponendo di risultati di studi attuali analoghi a quello

precedentemente esaminato (16), contenenti la presentazione di modelli opportuni per valutare, alla luce delle nuove prospettive, l'efficienza e la convenienza, per i vari campi applicativi, di una strutturazione distribuita dei sistemi di elaborazione, si può a ragione ipotizzare di poter completamente ribaltare, a favore dell'approccio "distribuito" dell'elaborazione, le conclusioni di quello studio. (16)

Questa affermazione è infatti coerente con tutto quanto sopra affermato, in questo paragrafo, nell'analisi delle cause e conseguenze del profondo mutamento di prospettive prodotto dalle innovazioni introdotte nel settore tecnologico.

° Delineazione di alcune caratteristiche di una nuova organizzazione distribuita dei sistemi di elaborazione, con lo scopo di realizzare i vantaggi precedentemente prospettati.

Per poter rendere effettivi i vantaggi, prospettati nel paragrafo precedente, derivati dalle innovazioni tecnologiche precedentemente discusse, realizzando gli incrementi, dei diversi tipi, nei rapporti teorici sopra esposti, è necessaria una opportuna organizzazione del sistema di elaborazione complessivo, sia dal punto di vista dell'architettura che dal punto di vista del sistema operativo.

Per cominciare, è possibile risparmiare sui costi di comunicazione, che si avviano a diventare preponderanti rispetto ai costi di elaborazione, nell'economia del sistema.

Ciò sarebbe realizzabile mediante un dimensionamento inferiore inferiore dei flussi massimi, cioè capacità, di trasferimento di informazioni sulle varie linee di comunicazione.

Infatti, avendo a disposizione elementi computazionali (Computing Elements) di grande potenza logica a costi decrescenti, si può applicare il princi-

pio di decentrare le funzioni del sistema dove devono essere effettivamente svolte, piuttosto che trasferirle per l'elaborazione a unità di processo centralizzate, da cui devono poi essere ritrasferiti i risultati.

Questo porta ad avere varie unità funzionali di processo, più o meno specializzate, distribuite nelle varie componenti del sistema, che diventa così un sistema distribuito, in modo da essere localizzate ove deve essere eseguita la corrispondente elaborazione.

Si risparmia così, ove possibile e conveniente, il doppio trasferimento, che si ripercuote sui costi di comunicazione, delle informazioni per le elaborazioni alle CPUs (Unità di Elaborazione Centrale) centralizzate, e dei risultati, o comunque dei segnali di controllo intesi in senso estensivo, di ritorno da tali CPUs.

Inoltre si può rendere, in questo modo, chiaro e pulito il disegno complessivo del sistema, semplificando e rendendo più organico il lavoro di quelle unità di processo con funzioni e capacità generali (le CPUs), che venivano altrimenti intasate, in una organizzazione centralizzata del sistema, con i più svariati tipi di incombenze, anche di tipo gestionale interno del sistema.

Una tale organizzazione distribuita può corrispondere anche alla richiesta, già esposta nel paragrafo precedente, di una opportuna strutturazione di sistemi per realizzare i vantaggi inerente alla disponibilità di nuove memorie centrali, aventi dimensioni di un superiore ordine di grandezza.

Tali memorie possono diventare effettivamente utilizzabili e gestibili, e quindi convenienti, se distribuite in un sistema complessivo ad organizzazione distribuita: infatti esse, in una tale struttura, saranno accessibili parallelamente dalle varie unità funzionali di processo, le quali potranno così cooperare, lavorando su queste memorie distribuite in maniera concorrente. Si può poi pensare di localizzare porzioni di tali memorie presso quelle unità di processo che necessitano di avere insieme i dati di lavoro più direttamente accessibili; si proceda così anche sulla strada della disaggre-

gazione della memoria centrale, per corrispondere alla topologia ed alla architettura più articolate di un sistema distribuito, abbandonando in tal modo il concetto di memoria centrale come supporto fisico unitario e ben isolabile nel sistema.

I processi evolutivi delle tecnologie hardware hanno influenzato non solo alcuni aspetti specifici dell'architettura dei sistemi, quali la distribuzione delle unità di processo e delle unità di memoria, ma pure l'organizzazione del "sistema integrato", inteso come il complesso dell'hardware (macchina nuda) con il software di sistema per la sua gestione (sistema operativo).

Infatti gli attuali orientamenti nel processo di disegno di sistemi hanno portato ad un cambiamento radicale nella strutturazione dei sistemi, intesi nella precedente accezione; tale cambiamento si pone nella direzione della distribuzione delle funzioni e del controllo, anche all'interno di un sistema di calcolo "general purpose" localizzato, concepito in maniera tradizionale. (14)

Per tali sistemi, che abbiamo detto "tradizionali", la strutturazione a livelli del sistema operativo, che ingloba la macchina nuda nel suo livello più interno, risale essenzialmente a Dijkstra, per il sistema T.H.E. (E. W. Dijkstra: "The structure of the T.H.E. multiprogrammed system" - A.C.M. Communications, Vol. 11, Maggio 1968).

La struttura complessiva veniva qui vista, secondo una prospettiva centralizzata, come una successione concentrica di macchine virtuali (livelli), dove i servizi dei livelli più interni sono disponibili ai processi in esecuzione a livello più esterno.

La distribuzione delle funzioni di sistema ai vari livelli è, a grandi linee, la seguente: (si veda la Fig. 5)

Livello 0: Virtualizzazione della CPU o processor computazionale (time slicing, dispatching della CPU, ecc.).

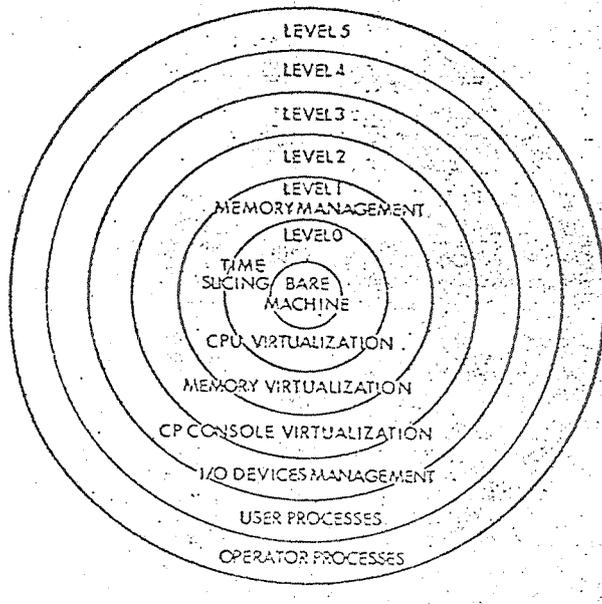


FIGURA 5
Struttura del Sistema THE di Dijkstra.

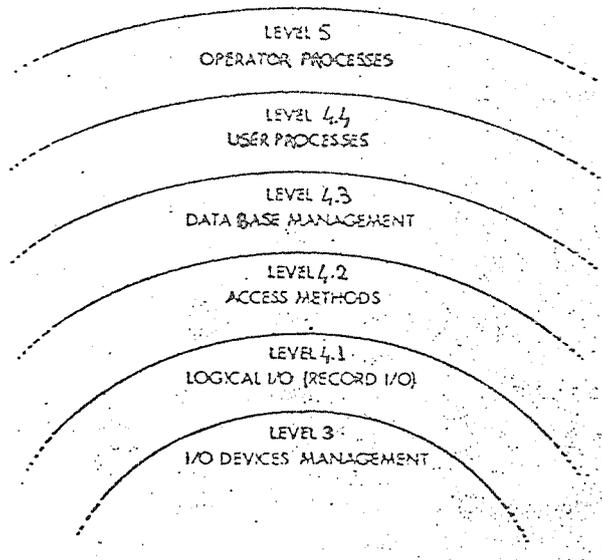


FIGURA 6
Struttura dei sistemi attuali (estensione del THE).

Livello 1: Virtualizzazione della memoria (gestione di pagine e segmenti residenti o non residenti in memoria eseguibile, ecc.).

Livello 2: Virtualizzazione della console d'operatore.

Livello 3: Gestione delle apparecchiature di Input/Output.

Livello 4: Ambiente di esecuzione dei processi utente.

Livello 5: Ambiente di esecuzione dei processi d'operatore.

Nella pratica di realizzazione di sistemi, successivi al sistema T.H.E. ed operazionalmente più avanzati e complessi, si è articolato l'importante Livello 4, dove si devono trovare anche i processi di utilità per la gestione di files e databases, con l'aggiunta di tre nuovi livelli: (si veda la Fig. 6) (14)

Livello 4.1: Funzioni per l'Input/Output logico (agganciamento dei records).

Livello 4.2: Metodi di accesso (diversi schemi di indirizzamento dei files).

Livello 4.3: Gestione di basi di dati (arbitrari reticoli di records, databases integrati).

Livello 4.4: Ambiente di esecuzione dei processi utente, a cui sono forniti in aggiunta i servizi dei tre nuovi livelli precedenti.

Soprattutto in quest'area di gestione dei dati si sono avuti notevoli cambiamenti che hanno mutato il modo stesso di distribuire le corrispondenti funzioni nella struttura, sopra esposta, del "sistema integrato".

Infatti attualmente le primitive di gestione di database sono in genere costruite in cima alla vecchia architettura di gestione files, come si vede nella struttura precedente. (14)

Quindi ne risulta che le funzioni di gestione dati (files e databases) sono interamente al di fuori del sistema basico; così non sono strettamente correlate alla gestione della memoria, e sono isolate dalle necessarie funzioni hardware da molti strati di software.

Questo non vale per il database del sistema operativo (blocchi per il controllo dei processori, delle altre apparecchiature fisiche, dei processi e i loro spazi di memoria, dei files, ecc.) che ogni volta viene agganciato ai Livelli

0 e 1; ne consegue che i tempi di accesso sono grandemente differenti, cioè di tre o quattro ordini di grandezza inferiori. (14)

Viceversa, per le basi di dati applicative, si hanno overheads computazionali dell'ordine di diversi millisecondi nell'esecuzione delle primitive per la gestione di tali dati.

Tali overheads sono tollerabili nella tecnologia di oggi, in cui i tempi di accesso delle apparecchiature delle memorie di massa sono dell'ordine di grandezza di decine di millisecondi.

Tutta questa organizzazione viene rimessa in questione con le tecnologie, a sviluppo futuro, delle cosiddette "memorie di massa elettroniche" (quali memorie a bolle magnetiche, "Charge Coupled Devices" o memorie a trasferimento di carica, ecc.), con tempi di accesso di ordine di grandezza inferiore al millisecondo.

Data l'alta densità di impaccamento ed il basso costo, si potranno avere "memorie di massa elettroniche" di larghissime dimensioni; in aggiunta con le memorie integrate, più veloci, destinate a costituire il livello superiore di memoria eseguibile, e pure con possibilità di raggiungere dimensioni via via crescenti, si potranno avere memorie centrali gerarchiche con dimensioni effettive di miliardi di bytes.

Come prima conseguenza di questo, diventa possibile avere porzioni significative delle basi di dati applicative residenti in memoria principale. (14)

Ciò significa che l'approccio corrente della bufferizzazione dei records nello spazio di indirizzabilità dei processi è destinato ad essere abbandonato e rimpiazzato dalla procedura di agganciare direttamente ("linking") un'intera sezione del database allo spazio di indirizzabilità del processo applicativo.

Precursore di questo modo di procedere è stato il sistema operativo MULTICS (R. C. Daley e J. B. Dennis: "Virtual memory, processes, sharing in MULTICS" A. C. U. Communications, Vol. 11, Maggio 1968) il cui "file system" è una collezione di segmenti che possono essere resi conosciuti, e quindi agganciati, ai processi individuali con opportune modifiche alla tabella dei

segmenti del processo.

Una volta, poi, che le basi di dati applicative possono essere direttamente agganciate ai processi che le usano, con tempi di accesso, anche per le porzioni di database più frequentemente usate, di ordini inferiori al millisecondo, vi potrebbe essere un incentivo molto scarso a trattare i database di sistema ed applicativi in maniere differenti.

Il risultato di questo nuovo approccio è essenzialmente l'unificazione della gestione della memoria e dei dati; ciò però non è possibile senza un cambiamento piuttosto drastico dell'architettura del "sistema integrato".

Infatti la strategia attuale, per cui il processor centrale, in esecuzione al Livello 3 di software, costruisce le procedure (programmi di canale) da essere eseguite da parte dei processor di I/O (Input/Output Channels), noi schedula ed inizializza l'attività di questi processor di I/O, è inerentemente lenta, richiedendo anche diversi millisecondi per ogni connessione di I/O.

Si ha quindi che l'architettura del "sistema integrato", che voglia raggiungere le alte prestazioni derivanti dalla gestione unificata, sopra esposta, della memoria e dei dati in un ambiente a tecnologia molto avanzata, dovrà richiedere che il processor principale ("master") sia non più il processor computazionale, ma il processor che gestisce la gerarchia di memoria. (

Da questa conclusione principale, e da altre considerazioni funzionali, che qui non esponiamo, sulle altre attività di un sistema di elaborazione nel senso "tradizionale" sopra esposto, viene ricavato il modello presentato da U. Gagliardi (14), di una nuova strutturazione dell'architettura di un "sistema integrato", che sia consistente, in prospettiva, con i processi evolutivi delle attuali tecnologie più avanzate.

Non si vuole qui fare una presentazione completa di questo modello, che tra l'altro si mantiene a livello di proposta per linee piuttosto generali, ma si vuole solo mettere in risalto come in esso il sottosistema di memoria sia il vero sottosistema principale ("master"), in posizione centrale nella struttura del "sistema integrato", mentre gli altri sottosistemi, cioè quel-

lo computazionale, quello per la comunicazione e quello delle unità-record (predisposto per le attività di "spooling" o conversione tra supporti diversi (compresi trasferimenti a e da memoria centrale estesa) di informazione), siano sottosistemi secondari ("slave"), come livello funzionale, ed inoltre siano in posizione periferica, rispetto al sottosistema di memoria, nella struttura complessiva del "sistema integrato". (si veda la Fig. 7).

Il sottosistema di memoria include, in questo modello, tutti i supporti fisici della memoria gerarchica ed un processor specializzato, chiamato DB-processor (processor dei databases).

Inoltre, poiché il sistema operativo esegue le sue funzioni di gestione delle risorse di sistema manipolando il database di sistema, ne risulta che tutto il software di gestione delle risorse di sistema è eseguito dal DB-processor; così esso diventa il gestore delle risorse dell'intero sistema di elaborazione.

Il processor computazionale (o il sistema di processors computazionali) è destinato ad eseguire esclusivamente processi utenti, così che il software d'applicazione ed il software di sistema risultano chiaramente separati, con il primo codificato per una macchina che include tutte le primitive di alto livello per la gestione integrata della memoria e di dati strutturati.

Ne risulta che la memoria, dal punto di vista del processor computazionale, apparirà come una memoria strutturata, con una ricca varietà di tipi di dati di alto livello, quali records, liste, pile, code, vettori, anelli, ecc.. (14)

La strutturazione completa, che risulta dal modello presentato, del "sistema integrato", è illustrata nella Fig. 7: in essa si nota come il sottosistema di memoria sia al centro di tale struttura, anche per le funzioni di controllo generale che possiede.

Viceversa gli altri tre sottosistemi, il sottosistema computazionale, di comunicazione, di "spooling" generalizzato, si pongono in posizione esterna, rispetto al sottosistema principale di memoria, ma, tra loro, si pongono in

STORAGE SUBSYSTEM :

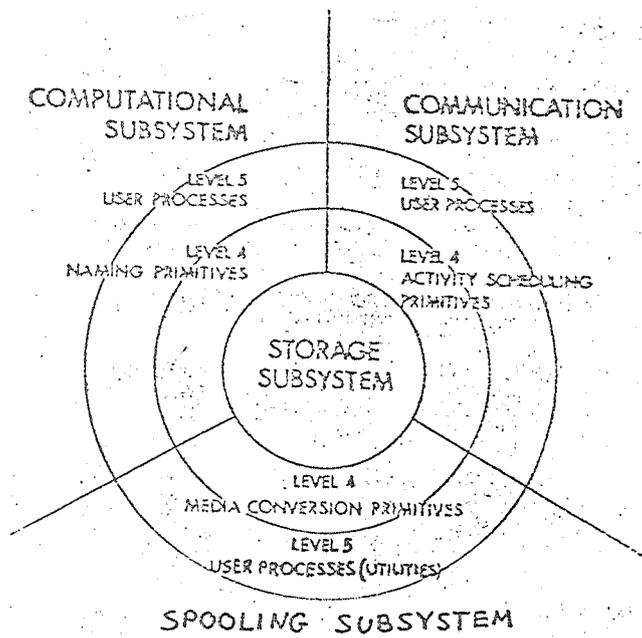
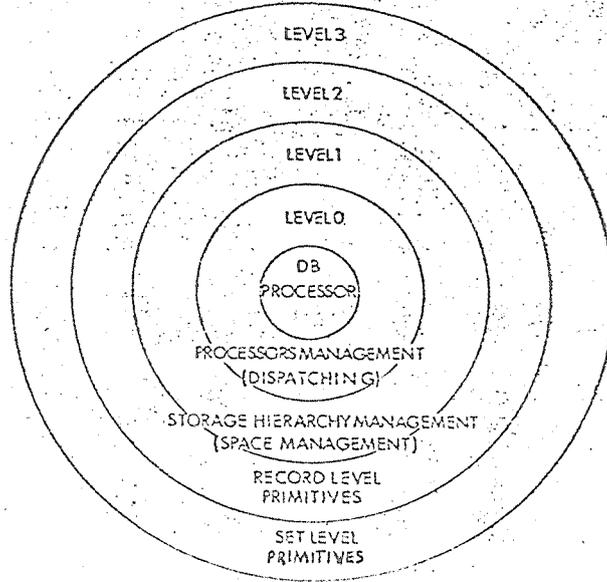


FIGURA 7

Struttura del modello di sistema futuro esposto da Gagliardi (14).

posizione parallela (questo rispecchia la natura parallela delle loro funzioni), senza nessuna gerarchia reciproca.

Tali tre sottosistemi "slave" vengono a costituire come tre settori paralleli, composti dai Livelli 4 e 5 della struttura complessiva, che circondano il sottosistema "master", costituito dai Livelli che vanno da 0 a 3.

L'attribuzione funzionale dei vari livelli, nei diversi sottosistemi, è specificata nella Fig. 7.

La considerazione più importante sull'architettura di questo modello di sviluppo futuro di Gagliardi (14), per i "sistemi tradizionali", è che il risultato consiste in una organizzazione funzionalmente distribuita della struttura presentata.

Nella struttura di questo modello si possono riconoscere unità funzionali distinte e cooperanti sotto controllo integrato (secondo la definizione generale di sistema distribuito). Infatti nei tre sottosistemi esterni si possono riconoscere unità funzionali (o sistemi di unità funzionali), con processors opportunamente specializzati, dedicate, nell'ambito del sistema di elaborazione, alle attività computazionali, alle attività di comunicazione (con altri sistemi o con utenti remoti), alle attività di conversione di records di informazione tra supporti diversi. Inoltre le attività di queste unità funzionali avvengono con il supporto del sottosistema centrale di memoria, col suo DB-processor, e sotto il controllo generale del sistema operativo in esecuzione in questo sottosistema, che può essere considerato l'unità funzionale principale.

Questa importante conclusione è consistente con quanto già affermato precedentemente, in un discorso più generale sui sistemi di elaborazione non ristretti ad un ambito "tradizionale" locale, come in questo modello. Infatti si ricava nuovamente che una organizzazione funzionalmente distribuita del sistema risulta più adatta a sfruttare i vantaggi dell'evoluzione, anche futura, delle tecnologie realizzative.

° Bilancio sulla possibilità di trovare una prospettiva unificante per i Sistemi Distribuiti: prima parte (dal punto di vista delle motivazioni e ragioni dei Sistemi Distribuiti).

Ci si era posti l'obbiettivo di rivedere la materia riguardante i Sistemi di Elaborazione ad architettura distribuita, sia dal punto di vista delle motivazioni generali del loro affermarsi, sia dal punto di vista delle varie realizzazioni effettive, con lo scopo di trovare una prospettiva unificante, se possibile e fin dove possibile, nell'esame di tali sistemi.

Riguardo la prima parte, che si è appena finito di esaminare, si può trarre un bilancio nel complesso positivo, nel senso che le varie considerazioni separate nell'approfondire le ragioni, sia di ordine funzionale che economico, si possono estendere a tutto il campo dei sistemi che applicano il principio della distribuzione nell'organizzare l'elaborazione, piuttosto che applicarsi a particolari classi di sistemi realizzate (che vanno dai multiprocessors alle reti di computers, ecc.), con differenze fondamentali di accenti.

Anche la delineaazione di aspetti di un sistema distribuito in senso generalizzato, esposta nel paragrafo precedente, come risultato dell'analisi delle motivazioni ai sistemi distribuiti, si mantiene su di un piano così generale e non legato a particolari aspetti implementativi, in modo tale da potersi concretizzare in svariate possibilità realizzative, senza legarsi a particolari classi realizzate di sistemi.

Si può, in conclusione, ribadire un bilancio positivo, per questa prima parte, nel senso che si è potuto sviluppare un discorso unico e comune per i sistemi distribuiti in generale, anche se gli aspetti di un modello risultante, che si sono potuti noi ricavare a grandi linee, rimangono ad un livello di abbozzo generico e con scarse possibilità di ulteriori specifiche in un'ottica.

PARTE SECONDA

CLASSIFICAZIONE GENERALE DI ARCHITETTURE
DI SISTEMI DI ELABORAZIONE DELLE INFORMAZIONI
CON DISCUSSIONE DI EVENTUALI CARATTERISTICHE
DISTRIBUITE IN ESSE REPERIBILI.

° Premessa: scopo di questa parte del lavoro.

Lo scopo di questa Seconda Parte del lavoro, come precedentemente anticipato, è di cercare di trovare una prospettiva unificante, fin dove possibile, nella visione dei sistemi distribuiti, dal punto di vista della realizzazione di tali sistemi (a differenza della Prima Parte, in cui ci si era posti dal punto di vista delle motivazioni ai sistemi distribuiti).

Per fare questo, si costruisce e si esamina una classificazione, il più completa possibile, anche se a livello necessariamente generale, di architetture di sistemi di elaborazione, in base soprattutto alla loro struttura fondamentale. In questa classificazione generale si sono presi in considerazione sia sistemi di elaborazione compiutamente realizzati, e prodotti su scala industriale, sia sistemi realizzati sperimentalmente per scopi di ricerca; ed infine, per completezza della classificazione, si sono considerati, a livello di semplice accenno a volte, i più importanti modelli di studio di sistemi non convenzionali, che si giudica presentino un particolare interesse o possano avere le maggiori possibilità di sviluppi realizzativi.

Nell'ambito di questa classificazione si vogliono reperire le reali tendenze realizzative, consolidate od emergenti, di sistemi distribuiti, cioè di quei sistemi la cui struttura possa essere riconosciuta effettivamente distribuita in base alla definizione informale di sistema distri-

huito esposta all'inizio del lavoro (tale definizione sarà riferita, da qui in avanti, come "definizione generale di sistema distribuito" o DGSD).

A proposito del significato di questa ricerca, occorre notare che, in base alla definizione riferita di sistemi distribuiti (DGSD), non si deve intendere il termine "sistema distribuito" nel senso superficiale, ma spesso largamente diffuso, per cui ci si riferisce essenzialmente alla distribuzione spaziale o geografica di unità componenti, ma nel senso più completo per cui la distribuzione viene intesa in senso funzionale.

Occorre tener presente con chiarezza la differenza fondamentale di questi due diversi modi di intendere la distribuzione dell'elaborazione per poter affrontare in modo corretto e senza malintesi l'esame generale dei sistemi di elaborazione.

A questo proposito si può riproporre un discorso basilare sul tipo di approccio ai sistemi distribuiti (7):

"La distribuzione dell'elaborazione è ottenibile sia mediante la distribuzione geografica delle risorse, sia tramite la distribuzione logica del controllo, ... Dal punto di vista strutturale, la distribuzione geografica è ottenibile mediante reti di calcolatori, mentre la distribuzione logica del controllo implica la realizzazione di un sistema distribuito nei vari sottoinsiemi connessi tra loro, più o meno lascamente, secondo criteri di organizzazione dei multiprocessors e delle reti di calcolatori".

E' chiaro che il punto di vista della nostra ricerca è quello di discutere ed identificare sistemi con distribuzione logica del controllo (oppure con controllo logico integrato tra unità funzionali distinte); naturalmente questo non esclude di considerare i sistemi con componenti geograficamente distribuite, anzi essi costituiscono una classe particolare dei sistemi distribuiti, secondo questo punto di vista.

° Modalità e problemi di costruzione della "Classificazione Generale dei Sistemi".

In letteratura si possono trovare varie classificazioni di sistemi di elaborazione di informazioni. Alcune di esse hanno una visione globale, ad alto livello, dei sistemi. Queste classificazioni cercano di coprire tutto lo spettro dei sistemi di elaborazione, suddividendoli in categorie piuttosto generali, in base a caratteristiche altrettanto generali sulla struttura dei sistemi e sul loro modo di operare.

Altre classificazioni si pongono a livello più basso, cioè prendono in visione un campo particolare di definizione di sistemi, ed operano una suddivisione, possibilmente a più livelli successivi di specifica, dei sistemi appartenenti a questo campo. In questo caso si procede in genere ad un esame più approfondito delle caratteristiche di questi sistemi, per poterli classificare in categorie più caratterizzate e particolari.

Per procedere alla costruzione di questa classificazione generale si è proceduto secondo questa semplice strategia: si è partiti da una classificazione globale, ad alto livello, per una prima suddivisione in categorie generali. Queste sono poi state specificate secondo classificazioni, a livello inferiore, riferentesi a quei campi particolari di sistemi. In quei settori, poi, che non risultavano sufficientemente specificati, si sono inserite ulteriori classificazioni, a livello ancora inferiore, per articolare in modo più completo i campi presi in esame.

Con questa strategia si è proceduto per vari livelli di profondità, fino a raggiungere l'obbiettivo di ottenere una struttura, il più possibile completa e dettagliata, di classificazione che potesse coprire tutte le aree praticate nella realizzazione e nello sviluppo dei sistemi di elaborazione. Si è fatto in modo che la struttura risultante sia così completa da servire da cornice ("main frame") ove poter inserire i diversi sistemi realizzati o studiati.

Occorre notare che in questo modo di procedere, usando classificazioni a livello decrescente di generalità, e quindi crescente in dettaglio, per arrivare ad una struttura finale soddisfacente, ci si è arrestati a diversi livelli di profondità, a seconda delle differenti direzioni via via esplorate nell'esame dei diversi settori. Si trova, infatti, che certi campi di realizzazione di sistemi sono articolati in maniera più complessa di altri, ed in questi campi si è proceduto a livelli più profondi di specifica, perchè sono stati interessati da un maggiore sviluppo e da una maggiore ricerca specifica rispetto ad altri.

Nella scelta delle possibili classificazioni, reperibili nella letteratura, da utilizzare in questa classificazione generale, si sono affrontati due tipi di problemi: problemi di consistenza temporale delle varie classificazioni che si vogliono usare, e problemi di consistenza dei loro rispettivi domini.

Il primo problema deriva dalla natura di questa materia, riguardante i sistemi di elaborazione, che è soggetta a rapide trasformazioni col passare del tempo. Può succedere così che uno stesso campo di realizzazione di sistemi sia esaminato e catalogato secondo visuali diverse, e possibilmente discordanti, da classificazioni diverse effettuate in tempi diversi.

Il secondo problema deriva dal fatto che classificazioni particolari, ai vari livelli di generalità, che si possono reperire nella letteratura, guardano a settori di realizzazione dei sistemi non sempre complementari e ben distinti, ma spesso tra loro intersecantesi a diversi livelli di profondità.

Ne risulta che la scelta e l'impiego di classificazioni di vari autori, da inglobare nella classificazione generale, è condizionata da questi due problemi. Infatti l'esigenza di consistenza temporale ha portato a scegliere le classificazioni il più possibile aggiornate, oppure quelle che si possono almeno ritenere ancora valide allo stato attuale dell'arte. L'esigenza di consistenza dei domini delle varie classificazioni usate,

ha portato alla scelta, dalla letteratura, di quelle combinazioni di classificazioni che permettano di evitare sovrapposizioni reciproche e che risultino complementari tra loro; si è cercato di rispettare questa esigenza anche a costo di intervenire su queste classificazioni, tagliandole parzialmente in caso di conflitti.

Da tutte queste premesse si ricava la struttura della classificazione generale di sistemi di elaborazione.

Essa è di natura essenzialmente gerarchica, articolata a vari livelli di profondità. Livelli superiori di classificazione corrispondono a categorie più generali, mentre livelli inferiori corrispondono a categorie più dettagliate. In pratica si ottiene una struttura tipo albero gerarchico: seguendo un suo ramo dall'alto al basso si ottengono successive categorie, specificate sempre più in dettaglio, fino ad arrivare alla classe finale (foglia) di quel ramo; seguendo viceversa un suo ramo dal basso all'alto, si ottengono le categorie, via via più generali, a cui appartiene la classe di partenza (foglia) di quel ramo.

In questa classificazione saranno dette categorie di livello K le classi di sistemi specificate al livello K di profondità. Una categoria di livello K è identificata da una successione di K numeri, in questo modo:

Categoria $/X_1, X_2, \dots, X_K/$
dove $X_1 \in \{1, 2, \dots, N_1\}$ con $N_1 =$ numero di categorie specificate a livello 1.
.....
 $X_i \in \{1, 2, \dots, N_i\}$ con $N_i =$ numero di categorie specificate a livello i.
.....
 $X_K \in \{1, 2, \dots, N_K\}$ con $N_K =$ numero di categorie specificate a livello K.

Può succedere poi, ad esempio, che la Categoria $/X_1, X_2, \dots, X_K/$ si possa articolare, nell'ambito di questa classificazione generale, in R successive categorie, che saranno di livello $K+1$. Si ha così la seguente classificazione di livello $K+1$:

Classificazione $/X_1, X_2, \dots, X_K, X/$

composta dalle seguenti R categorie:

Categorie $/X_1, X_2, \dots, X_K, X_{K+1}/$ con $X_{K+1} \in \{1, 2, \dots, R\}$

Può succedere, in questa classificazione generale, che un particolare sistema di elaborazione possa essere inserito in categorie diverse; questo può succedere per quei sistemi cosiddetti "misti", cioè non facilmente classificabili univocamente perchè aventi caratteristiche differenti e possibilmente contrastanti, in funzione di una rigida classificazione. Nella successiva spiegazione della classificazione generale, questi sistemi saranno specificamente indicati.

° Struttura della Classificazione Generale di Architetture di Sistemi di Elaborazione.

- /1./ - SISD: Single Instruction stream, Single Data stream Systems.
 - Sistemi SISD, a flusso singolo di istruzioni ed a flusso singolo di dati.

 - /2./ - MISD: Multiple Instruction stream, Single Data stream Systems.
 - Sistemi MISD, a flusso multiplo di istruzioni ed a flusso singolo di dati.

 - /3./ - SIMD: Single Instruction stream, Multiple Data stream Systems.
 - Sistemi SIMD, a flusso singolo di istruzioni ed a flusso multiplo di dati.

 - /4./ - MIMD: Multiple Instruction stream, Multiple Data stream Systems.
 - Sistemi MIMD, a flusso multiplo di istruzioni ed a flusso multiplo di dati.
-
- /3,1./ - Parallel processor Systems.
 - Sistemi a processor parallelo.

 - /3,2./ - Ensemble processor Systems.
 - Sistemi a insieme di processors.

 - /3,3./ - Associative processor Systems.
 - Sistemi a processor associativo.

 - /3,4./ - Unconventional distributed SIMD Systems.
 - Sistemi SIMD distribuiti, non convenzionali.

/3,1,1./ - Array processor Systems.

- Sistemi a processor a struttura matriciale.

/3,1,2./ - Orthogonal processor Systems.

- Sistemi a processor ortogonale.

/3,1,3./ - Highly parallel experimental Systems.

- Sistemi sperimentali a struttura altamente parallela.

/3,3,1./ - Fully parallel associative processor Systems.

- Sistemi a processor associativo, completamente parallelo.

/3,3,2./ - Word parallel associative processor Systems.

- Sistemi a processor associativo, parallelo a parola.

/3,3,3./ - Word serial associative processor Systems.

- Sistemi a processor associativo, seriale a parola.

/3,3,4./ - Block oriented associative processor Systems.

- Sistemi a processor associativo, orientato a blocchi.

/3,3,5./ - Highly parallel associative processor Systems.

- Sistemi a processor associativo, altamente parallelo.

/3,3,1,1./ - Fully parallel, word organized associative processor Systems.
- Sistemi a processor associativo, completamente parallelo, organizzato a parola.

/3,3,1,2./ - Fully parallel, distributed logic associative processor Systems.
- Sistemi a processor associativo, completamente parallelo, a logica distribuita.

/3,3,2,1./ - Bit serial, word parallel associative processor Systems.
- Sistemi a processor associativo, seriale a bit e parallelo a parola.

/3,3,2,2./ - Byte serial, word parallel associative processor Systems.
- Sistemi a processor associativo, seriale a byte e parallelo a parola.

/3,4,1./ - Microprogrammed modular Systems.
- Sistemi modulari microprogrammati.

/3,4,2./ - Configurable Systems (Data-flow Systems).
- Sistemi configurabili (Sistemi data-flow).

/3,4,1,1./ - Hierarchical cooperation modular Systems.
- Sistemi modulari a cooperazione gerarchica.

/3,4,1,2./ - Anonymous cooperation modular Systems.
- Sistemi modulari a cooperazione anonima.

/3,4,2,1./ - Interconnection mode data-flow Systems.
- Sistemi data-flow a modalità di interconnessione.

/3,4,2,2./ - Search mode data-flow Systems.
- Sistemi data-flow a modalità di ricerca.

/4,1./ } - D-MIMD: Direct message transfer strategy MIMD Systems.
/4,1,1./ } - Sistemi MIMD a strategia diretta di trasferimento messaggi
(D-MIMD).

/4,2./ - I-MIMD: Indirect message transfer strategy MIMD Systems.
- Sistemi MIMD a strategia indiretta di trasferimento messaggi
(I-MIMD).

/4,2,1./ - IC-MIMD: Centralized transfer control method I-MIMD Systems.
- Sistemi MIMD a metodo centralizzato di controllo trasferimento
(IC-MIMD).

/4,2,2./ - ID-MIMD: Decentralized transfer control method I-MIMD Systems.
- Sistemi I-MIMD a metodo decentralizzato di controllo trasferimento
(ID-MIMD).

/4,1,1,1./ - DD-MIMD: Dedicated transfer path D-MIMD Systems.
- Sistemi D-MIMD a cammini di trasferimento dedicati (DD-MIMD).

/4,1,1,2./ - DS-MIMD: Shared transfer path D-MIMD Systems.
- Sistemi D-MIMD a cammini di trasferimento condivisi (DS-MIMD)

/4,2,1,1./ - ICD-MIMD: Dedicated transfer path IC-MIMD Systems.
- Sistemi IC-MIMD a cammini di trasferimento dedicati (ICD-MIMD).

/4,2,1,2./ - ICS-MIMD: Shared transfer path IC-MIMD Systems.
- Sistemi IC-MIMD a cammini di trasferimento condivisi (ICS-MIMD)

/4,2,2,1./ - IDD-MIMD: Dedicated transfer path ID-MIMD Systems.
- Sistemi ID-MIMD a cammini di trasferimento dedicati (IDD-MIMD).

/4,2,2,2./ - IDS-MIMD: Shared transfer path ID-MIMD Systems.
- Sistemi ID-MIMD a cammini di trasferimento condivisi (IDS-MIMD)

/4,1,1,1,1./ - DDL-MIMD: Loop architecture DD-MIMD Systems.
- Sistemi DD-MIMD ad architettura ad anello (DDL-MIMD).

/4,1,1,1,2./ - DDC-MIMD: Complete interconnection architecture DD-MIMD Systems.
- Sistemi DD-MIMD ad architettura ad interconnessione completa (DDC-MIMD).

/4,1,1,2,1./ - DSM-MIMD: Central memory arcitecture DS-MIMD Systems (Multiprocessor architecture Systems).
- Sistemi DS-MIMD ad architettura a memoria centrale (DSB-MIMD) (Sistemi ad architettura multiprocessor).

/4,1,1,2,2./ - DSB-MIMD: Global bus architecture DS-MIMD Systems.
- Sistemi DS-MIMD ad architettura a bus globale (DSB-MIMD).

/4,2,1,1,1./ - ICDS-MIMD: Star architecture ICD-MIMD Systems.
- Sistemi ICD-MIMD ad architettura a stella (ICDS-MIMD).

/4,2,1,1,2./ - ICDL-MIMD: Loop with central switch architecture ICD-MIMD Systems.
- Sistemi ICD-MIMD ad architettura ad anello con elemento di commutazione centrale (ICDL-MIMD).

/4,2,1,2,1./ - ICS-MIMD: Bus with central switch architecture ICS-MIMD Systems.
- Sistemi ICS-MIMD ad architettura a bus con elemento di commutazione centrale (ICS-MIMD).

/4,2,2,1,1./ - IDDR-MIMD: Regular network architecture IDD-MIMD Systems.
- Sistemi IDD-MIMD ad architettura a rete regolare (IDDR-MIMD).

/4,2,2,1,2./ - IDDI-MIMD: Irregular network architecture IDD-MIMD Systems (Geographically distributed computer network Systems).
- Sistemi IDD-MIMD ad architettura a rete irregolare (IDDI-MIMD) (Sistemi a rete di calcolatori geograficamente distribuite).

/4,2,2,2,1./ - IDS-MIMD: Bus window architecture IDS-MIMD Systems.
- Sistemi IDS-MIMD ad architettura a finestra su bus (IDS-MIMD).

- /4,1,1,2,1,1./ - Time-shared common bus multiprocessor architecture (DSM-MIMD) Systems.
- Sistemi ad architettura multiprocessor (DSM-MIMD) a bus comune a condivisione di tempo.
- /4,1,1,2,1,2./ - Crossbar switch multiprocessor architecture (DSM-MIMD) Systems.
- Sistemi ad architettura multiprocessor (DSM-MIMD) ad elemento di commutazione tipo crossbar.
- /4,1,1,2,1,3./ - Multiport memory multiprocessor architecture (DSM-MIMD) Systems.
- Sistemi ad architettura multiprocessor (DSM-MIMD) a memoria multiporto.
-

- /4,2,2,1,2,1./ - Broadcast communication media, computer network (IDDI-MIMD) Systems.
- Sistemi rete di calcolatori (IDDI-MIMD) a mezzi di comunicazione a diffusione.
- /4,2,2,1,2,2./ - Point-to-point communication media, computer network (IDDI-MIMD) Systems.
- Sistemi rete di calcolatori (IDDI-MIMD) a mezzi di comunicazione punto-a-punto.

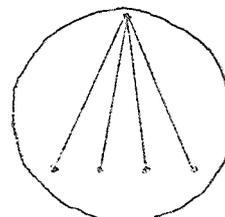
- /4,2,2,1,2,2,1./ - Circuit switching, point-to-point communication media, computer network (IDDI-MIMD) Systems.
- Sistemi rete di calcolatori (IDDI-MIMD) a mezzi di comunicazione punto-a-punto, a commutazione di circuito.
- /4,2,2,1,2,2,2./ - Message switching, point-to-point communication media, computer network (IDDI-MIMD) Systems.
- Sistemi rete di calcolatori (IDDI-MIMD) a mezzi di comunicazione punto-a-punto, a commutazione di messaggio.
- /4,2,2,1,2,2,3./ - Packet switching, point-to-point communication media, computer network (IDDI-MIMD) Systems.
- Sistemi rete di calcolatori (IDDI-MIMD) a mezzi di comunicazione punto-a-punto, a commutazione di pacchetto.
-
-

° Spiegazione della Classificazione Generale di Architetture di Sistemi di Elaborazione.

CLASSIFICAZIONE GLOBALE /X/

Primo livello

4 Categorie /X./, $X \in \{1, 2, 3, 4\}$



Inizialmente, si è dovuto operare una scelta, nella letteratura specializzata, di una classificazione che abbia una visione globale, al più alto livello, dei sistemi e che copra tutto lo spettro delle possibilità esistenti. Questa scelta è molto importante, perchè su di essa va costruita questa classificazione generale, innestandovi altre classificazioni più specifiche, per i vari livelli inferiori nella struttura.

Si è scelto, a questo scopo, la famosa classificazione globale di M. J. Flynn (17), poichè, oltre alla sua intrinseca chiarezza, ha il merito di essere universalmente conosciuta e seguita, e quindi può essere usata in maniera effettiva per i nostri scopi, essendo consistente con la gran parte di altre classificazioni, in letteratura, a livelli più dettagliati.

Naturalmente questa classificazione di Flynn non è l'unica classificazione globale presente in letteratura, anche se è stata scelta per questi scopi. Perciò alla fine di questa sezione si accennerà ad altre classificazioni alternative alla classificazione di Flynn e ad altre classificazioni più dettagliate usate in questa classificazione generale, che costituiscano casi di particolare interesse.

Nella classificazione globale proposta da Flynn, per considerare le strutture architettoniche da un punto di vista macroscopico, ci si rifà al concetto di flusso (17). Flusso, in questo contesto, significa

semplicemente sequenza di entità che comandino l'esecuzione di una Unità di Elaborazione (istruzioni), o su cui operi una Unità di Elaborazione (dati).

Da questa visuale si ha che le organizzazioni di sistemi sono differenziate dall'entità (o spaziale o a suddivisione di tempo) delle interazioni dei loro flussi di istruzioni e di dati.

La prima immediata distinzione deriva dal constatare se un sistema ha un singolo flusso (sia nel caso di istruzioni che di dati) oppure ha molteplici flussi. Dalla unicità o dalle molteplicità del flusso di istruzioni e del flusso di dati (cioè dalle quattro combinazioni risultanti dai due rispettivi casi) ne deriva immediatamente la classificazione globale delle possibili organizzazioni macchina nelle seguenti quattro categorie:

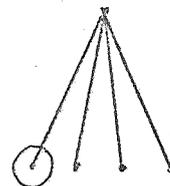
- 1) Sistemi SISD
- 2) Sistemi MISD
- 3) Sistemi SIMD
- 4) Sistemi MIMD

In realtà questa classificazione di Flynn viene riferita a sistemi considerati come macchine singole (anche in conseguenza all'epoca della stesura) e non vengono discusse in essa sistemi a struttura geograficamente distribuita. Tuttavia anche a tali sistemi si può applicare questa classificazione poichè anche per essi si possono fare le stesse considerazioni sui flussi di istruzioni e di dati.

CATEGORIA /1./ - Livello 1

Sistemi SISD

Questa è la categoria dei sistemi con organizzazione a singolo flusso di istruzioni ed a singolo flusso di dati.



Ad essa appartiene la gran parte dei calcolatori convenzionali disponibili oggi a livello commerciale. Infatti a questa categoria appartengono i sistemi a struttura centralizzata uniprocessor (10), cioè quella struttura che più direttamente deriva dall'organizzazione interna tradizionale alla Von Neuman.

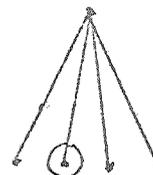
Naturalmente in questa categoria si può riscontrare una gran varietà di sistemi, i quali, appunto perchè così vastamente prodotti ed usati, si possono grandemente differenziare sia per le loro dimensioni e prestazioni (si passa, ad esempio, dai minicalcolatori, come il DEC PDP-11, ai calcolatori medi, come l'UNIVAC 90/30, ed ai calcolatori grandi, come l'IBM 370, ecc.), sia per il grado di avanzamento tecnologico dei loro componenti, sia per la diversa raffinatezza e complessità dell'organizzazione interna e del modo di lavorare delle loro singole unità fondamentali costituenti (la CPU può, ad esempio, essere o meno un processor micronprogrammato, l'Input/Output può essere affidato a Canali di I/O con differenti capacità di funzionamento autonomo, la memoria può essere gerarchizzata in livelli, dal "cache" alla memoria di massa, con differenti tempi di accesso, ecc.).

Tuttavia è importante notare che questa diversità di caratteristiche, anche notevoli, non risulta influente ai fini di questa classificazione per strutture architettoniche fondamentali.

Tali sistemi sono così accomunati in questa categoria che non viene qui ulteriormente differenziata, anche perchè in questo lavoro si vuole approfondire l'indagine su sistemi più originali o comunque con un più spiccato interesse architettonico (anche nell'ottica dei sistemi distribuiti).

CATEGORIA /2./ - Livello 1

Sistemi MISD



Questa è la categoria dei sistemi con organizzazione a flusso multiplo di istruzioni ed a flusso singolo di dati.

A proposito di questa categoria, Flynn parla di possibili organizzazioni specializzate, e, come unico esempio di una forma degenerata di struttura MISD, parla di macchine collegate a spina, di un'epoca ormai passata, in cui i flussi di istruzioni consistevano in singole istruzioni (nelle macchine collegate in successione) e in cui un dato derivato (SD) era passato dal passo programma i al passo programma $i+1$ (cioè dalla macchina i alla macchina $i+1$ - MD). (17)

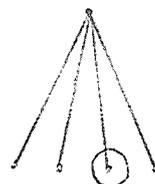
Viceversa Thurber (10) include in questa categoria i processors "pipeline", come il CDC STAR. Tuttavia questa scelta sembra inopportuna poiché è vero che, in linea di principio, nel sistema pipeline i successivi stadi di elaborazione operano su di un flusso comune di dati (ogni stadio S_i riceve gli operandi dallo stadio S_{i-1} e fornisce i risultati allo stadio S_{i+1}) e sono comandati, essendo unità specializzate, da diversi flussi di comandi, ma è anche vero il fatto che questo tipo di organizzazione riguarda il modo interno di operare del processor. Infatti esso riguarda, in genere, l'organizzazione particolare in sottounità specializzate di una unità esecutiva (35), cioè di esecuzione di istruzioni, all'interno di processors generalmente ad organizzazione di unità istruzioni Look-Ahead, come ad esempio l'IBM 360/195, il CDC 6600 e 7600 e il CDC STAR/100. Esso non riguarda, invece, strettamente l'organizzazione strutturale del sistema complessivo: in effetti si ha che i flussi multipli che pilotano le sottounità esecutive organizzate in "pipeline", non sono costituiti che dalle microistruzioni (di tipo orizzontale) derivate da un flusso singolo di istruzioni, ed è tale flusso singolo che, a livello macroscopico della struttura, induce a classificare i sistemi "pipeline" come

sistemi a singolo flusso di istruzioni (SI) ed a singolo o multiplo flusso di dati (SD o MD, a seconda dei casi).

In effetti, si può ritenere che questa categoria sia stata inserita più per completezza della classificazione globale che per effettiva rispondenza a possibili strutture organizzative di sistemi di elaborazione effettivi.

CATEGORIA /3./ - Livello 1

Sistemi SIND



Questa è la categoria dei sistemi con organizzazione a flusso singolo di istruzioni ed a multiplo flusso di dati.

A questa categoria appartengono sistemi con una singola unità di controllo globale che pilota unità di processo multiple (con una propria memoria locale, generalmente), ognuna delle quali può, nella gran parte delle realizzazioni, eseguire oppure ignorare l'istruzione corrente (10).

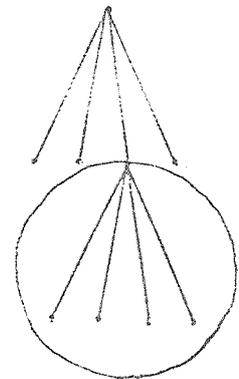
In questa organizzazione, l'unità di controllo globale è preposta alla esecuzione del flusso singolo di istruzioni, mentre ciascuna unità di processo è preposta all'elaborazione, secondo quanto richiesto dall'unità centrale di controllo, dei vari flussi di dati paralleli.

Questo tipo di organizzazione è concettualmente orientata verso un parallelismo funzionale interno che sfrutti gli aspetti concorrenti insiti nella natura di certe elaborazioni più o meno specifiche (12).

Ricadono nell'ambito di questa categoria anche sistemi "special purpose" destinati all'elaborazione di problemi specifici caratterizzati da un

inerente parallelismo nella loro esecuzione: esempi tipici sono problemi di elaborazione di segnali radar e problemi di "pattern recognition" (riconoscimento di modelli) (12). In questi sistemi l'organizzazione interna rispecchia la struttura del problema specifico; questo è il caso di problemi, ad esempio, che si risolvono in una elaborazione parallela di grandi quantità di dati a struttura matriciale (problemi di fisica nucleare, di idrodinamica, di meteorologia, ecc.) (12).

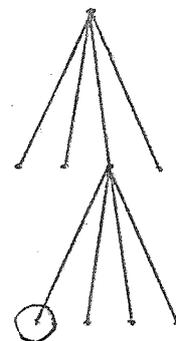
CLASSIFICAZIONE /3,X/ - Livello 2
4 Categorie /3,X./, $X \in \{1,2,3,4\}$



Questa classificazione dei sistemi SIMD (Categoria /3./) è presa essenzialmente da Thurber (10) per quel che concerne le prime tre categorie (cioè le Categorie /3,1./, /3,2./ e /3,3./); mentre si è ritenuto opportuno aggiungere ad esse la Categoria /3,4./, che raccoglie quei sistemi SIMD a struttura e concezione non convenzionali, con conformazione funzionale distribuita, i quali non rientrano nelle tre categorie precedenti.

CATEGORIA /3,1./ - Livello 2

Sistemi a processor parallelo



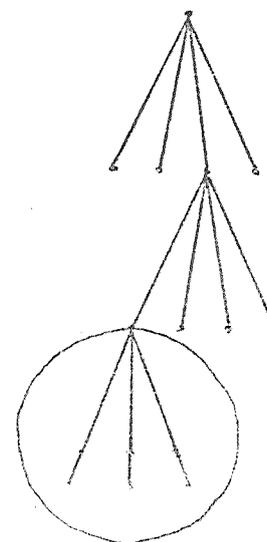
A questa categoria appartengono le macchine SIMD in cui gli elementi di processo (cioè gli elementi computazionali operanti sotto il controllo integrato dell'unità di controllo globale) hanno singolarmente una capacità limitata di elaborazione ma sono caratterizzate da un alto livello di interconnessione tra i singole PE (elementi di processo) (10).

Oltre a questa caratteristica strutturale, i sistemi appartenenti a questa categoria sono accomunati dai tipi di applicazioni che appaiono effettive dal punto di vista del costo. Ad esempio si possono citare applicazioni a problemi di inseguimento di tracce di obiettivi (36), problemi di elaborazione di segnali (37), problemi di riconoscimento di modelli (38).

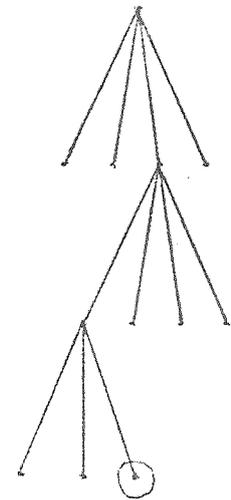
CLASSIFICAZIONE /3,1,X/ - Livello 3

3 Categorie /3,1,X./, $X \in \{1,2,3\}$

Questa classificazione è stata essenzialmente ricavata dall'articolo di Thurber (10), anche se la Categoria /3,1,3./ non vi appare esplicitamente, ma è stata aggiunta per completezza di classificazione.



CATEGORIA /3,1,3./ - Livello 3
Sistemi sperimentali a struttura
altamente parallela



Questa categoria raccoglie sistemi che, in questa classificazione, si sono denominati come "sperimentali a struttura altamente parallela". Tale categoria è stata aggiunta nella Classificazione /3,1,X./ con lo scopo di raccogliere esempi di macchine sperimentali, o anche solo modelli, di interesse di studio, che si possono inserire nella Categoria /3,1./, ma sono caratterizzati dall'alto livello di parallelismo interno.

A questa categoria (/3,1,3./) appartengono rilevanti modelli di studio di "Macchine a struttura matriciale di celle" (Cells array machines). Questi modelli sono costituiti da matrici generalizzate di celle (rappresentanti automi o unità con semplici capacità elaborative), ciascuna collegata con le quattro celle più vicine nella struttura.

Tali esempi di modelli di studio di macchine SIMD includono gli Automi cellulari di Von Neuman (39) e la Macchina di Holland (40). In questo tipo di struttura non vi è un meccanismo di controllo centrale che gestisca la matrice generalizzata degli automi, ma il controllo complessivo deriva dalla interazione tra l'insieme degli automi.

E' da notare che questi modelli di macchine appartengono ad un'epoca ormai passata (fine anni '50 - inizio anni '60) e rappresentano storicamente i primi tentativi di esplorare strutture di macchine ad organizzazione altamente parallela. Alla fine di questo tipo di ricerche l'interesse si è spostato su sistemi ad organizzazione matriciale, operanti in maniera SUP, cioè con funzione di controllo centralizzata ("Array processor") (10). E' questo il motivo per cui questi modelli sono, in letteratura, compresi tra le macchine SIMD, anche se ciò non sarebbe del

tutto proprio, dal punto di vista strutturale.

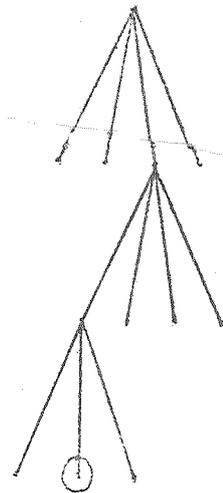
La Categoria /3,1,3./ presenta, poi, una realizzazione sperimentale rilevante, cioè la Macchina di Unger (41), macchina orientata all'elaborazione di linee. Anche questa macchina, come i modelli a cui si è precedentemente accennato, si può collocare nell'area degli studi storici (fine anni '50) che hanno portato alle successive realizzazioni di sistemi a processor parallelo.

La Macchina di Unger possedeva le tipiche funzioni per eseguire il riconoscimento di modelli, come il riconoscimento e l'assottigliamento di linee, la loro estensione, la determinazione del centro. Le unità funzionali principali del complesso consistevano in un computer di controllo centrale e in una matrice di Elementi di Processo (PE). I PE erano connessi ai loro rispettivi quattro più prossimi vicini. Però, a differenza dei successivi Array Processors (Categoria /3,1,1./), i PE non avevano un bit di attività, per escluderli selettivamente dall'esecuzione di certi comandi globali. Tuttavia la matrice di PE poteva abilitare salti condizionali nel computer di controllo principale, per mezzo di un meccanismo che interrogava la condizione di tutti zero in un bit designato nei PE della matrice.

CATEGORIA /3,1,2./ - Livello 3

Sistemi a processor ortogonale

Il concetto di computer ortogonale è stato proposto da Shooman nel 1960 (42). La struttura consiste in una unità aritmetica orizzontale (HAU) ed in una unità aritmetica verticale (VAU) che condividono l'accesso ad una memoria ortogonale (OM), tale cioè che può essere vista in modo diverso nelle due direzioni, orizzontale e verticale. La OM è una apparecchiatura di memoria che fornisce l'accesso sia alle unità parola (word slice - HAU access) orizzontalmente, sia alle unità bit (bit slice - VAU access) verticalmente. Ne risulta che con l'accesso orizzontale (HAU) si può avere una elaborazione seriale a parola e parallela a bit (cioè tradizionale) della OM, mentre con l'accesso verticale (VAU) si può avere una elaborazione seriale a bit e parallela a parola della OM (vedi Fig. 8).



L'accesso verticale alla OM è quindi di tipo associativo, ragione per cui i Processors Ortogonali possono anche essere visti come speciali Processors Associativi (Categoria /3,3./) e quindi essere inseriti precisamente nella Categoria /3,3,2,1./, come si vedrà a proposito.

L'implementazione principale di sistema ad architettura ortogonale è l'OMEN (Ortogonal Mini EmbedmeNt), sistema di elaborazione disegnato dalla Sanders Associates (43). Esso utilizza come OM una memoria ad accesso associativo byte-slice, come HAU un PDP-11 (minicalcolatore della Digital Equipment Co.), ed infine possiede 64 Elementi di Processo nella VAU. Si ha così che l'accesso casuale alla OM nella direzione verticale è byte-slice nell'OMEN, invece di bit-slice, come nella proposta originaria dei sistemi ortogonali (42).

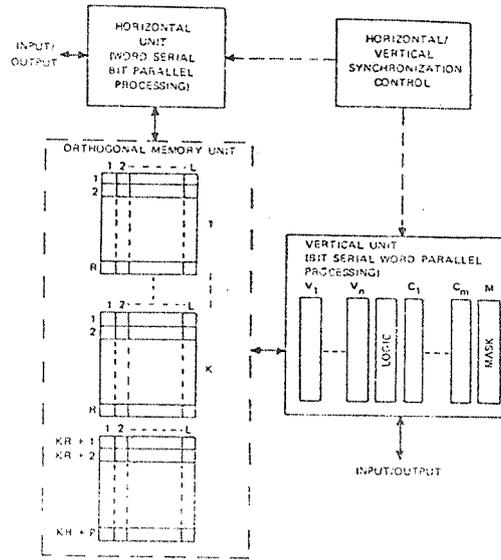


FIGURA 8
Diagramma a blocchi di un Computer Ortogonale.

CATEGORIA /3,1,1./ - Livello 3
Sistemi a processor a struttura
matriciale (Array processor)

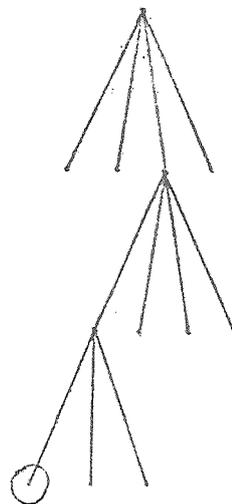
I sistemi Array Processor costituiscono la classe più importante nell'ambito della Categoria /3,1./, infatti gli Array Processors rappresentano le realizzazioni più usate e diffuse di Parallel Processors.

I sistemi Array Processor (44) possiedono una matrice, possibilmente generalizzata, di Elementi di Processo (PE) con connessione del tipo ai quattro vicini più prossimi (in genere in una topologia "chiusa"). Ciascun PE ha conformazione interna uguale ed uguali capacità elaborative, con una propria memoria di capacità limitata, per contenere i dati su cui lavora la sottounità logico-aritmetica del PE.

Tali PE sono controllati e pilotati dall'unità di controllo (CU), che provvede all'esecuzione delle istruzioni globali. Si ha quindi che i PE hanno un particolare insieme di istruzioni, differente dall'insieme di istruzioni della CU; ogni istruzione globale della CU comporta, in genere, l'esecuzione di un particolare sottoprogramma a livello delle PE, nel loro particolare linguaggio. Si può avere un'esecuzione differenziata di un'istruzione globale della CU sui vari dati paralleli nei PE, tramite un bit di esclusione presente in ogni PE della matrice. Tali bits di controllo possono essere utilizzati per salti condizionati a livello di CU.

Gli esempi più importanti di Array Processor sono il SOLOMON I, il SOLOMON II e l'ILLIAC IV della Burroughs. Tali realizzazioni indicano anche una evoluzione nei sistemi di questa categoria.

Il SOLOMON I (45) era un processor seriale a bit. Ciascun PE conteneva un accumulatore seriale e poteva eseguire istruzioni. Un diagramma



a blocchi generale del sistema SOLOMON I è illustrato in Fig. 9; tale diagramma indica, in generale, la struttura complessiva di un Array Processor.

Nel SOLOMON I particolari registri erano usati per ottenere un Input/Output parallelo della matrice dei PE; essi potevano leggere dalla cima, dal fondo, dal lato destro o sinistro della matrice di PE. Era anche disponibile un I/O seriale attraverso la CU.

Pochi importanti cambiamenti al SOLOMON I hanno portato al SOLOMON II. Il concetto di aritmetica serale, sebbene del tutto flessibile, era troppo lento per le applicazioni proposte. Quindi le unità aritmetiche del SOLOMON II (nei singoli PE) furono cambiate a unità floating-point (a virgola mobile) a 24 bits.

Ulteriori importanti sviluppi ai processors SOLOMON hanno portato al sistema ILLIAC IV (46), il più grande Parallel Processor (Categoria /3,1./) correntemente operante. In questo sistema, le unità aritmetiche in virgola mobile, nei PE, sono state ulteriormente modificate fino ad una lunghezza di parola di 32 bits. Inoltre la configurazione matriciale è cambiata a quattro quadranti 8X8 di PE. Lo scopo della configurazione a quadranti multipli era di permettere l'elaborazione di programmi particolari ad alto parallelismo.

Infine occorre notare che questi tipi di macchine Array Processor sono stati specificatamente designati a lavorare su problemi involventi equazioni differenziali, manipolazioni di matrici, elaborazioni di dati meteorologici ed algebra lineare (15).

Come molti altri sistemi SIMD (Categoria /3./), questi sistemi sono, nella loro struttura architettonica, concepiti per lavorare su problemi che implicano attività, nel sistema, ad alto parallelismo interno. Per questa ragione sono spesso denominati come "special-purpose" (cioè con sconi specializzati).

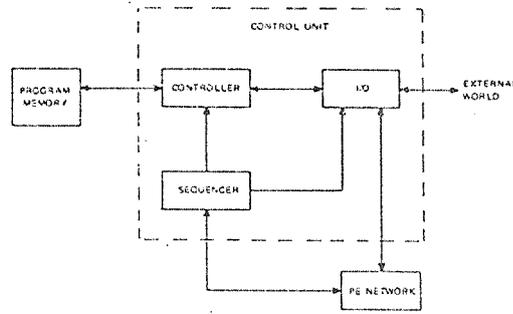


FIGURA 9
Struttura complessiva di un Array Processor (in particolare riferita al sistema SOLOMON I).

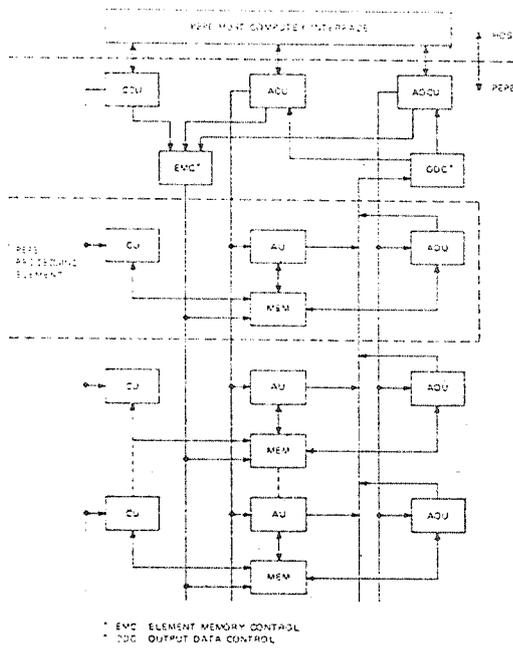
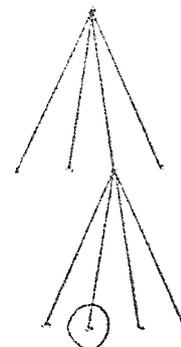


FIGURA 10
Struttura logica del sistema PEPE.

Dal punto di vista della loro efficacia elaborativa in generale, cioè della valutazione delle loro prestazioni rispetto ai costi, in rapporto agli usuali sistemi SISD Uniprocessor (Categoria /1./), si può dire che la concezione architeturale dei sistemi SIMD è controversa. Essa ha attualmente detrattori (19) e fautori (47) nella letteratura.

Il punto chiave, però, è che i sistemi SIMD (Categoria /3./), e Parallel Processor in particolare (Categoria /3,1./), sono in grado di avere prestazioni notevolissime per le applicazioni per cui sono correttamente disegnati, ma non sono effettivi, cioè sufficientemente efficaci, quando sono applicati a problemi che non sono stati designati a risolvere (10).

CATEGORIA /3,2./ - Livello 2
Sistemi a insieme di processors
(Ensemble processor)



A questa categoria appartengono le macchine SIMD in cui gli elementi di processo (cioè gli elementi computazionali operanti sotto il controllo integrato dell'unità di controllo globale) hanno un livello di reciproca interconnessione molto basso od inconsistente.

La più importante, e forse unica rilevante, realizzazione di Ensemble Processor è il PEPE (Parallel Element Processing Ensemble) (48).

Questo sistema è stato disegnato per elaborazioni riguardanti la difesa radar da missili balistici. La configurazione della struttura logica del sistema è mostrata in Fig. 10.

Questo sistema può effettuare simultaneamente tre azioni principali:

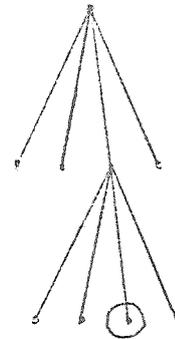
l'immissione di dati nel complesso CCU/CU; l'aggiornamento delle tracce da seguire, nel complesso ACU/AU; l'emissione di comandi di controllo radar, attraverso il complesso AOCU/AOU.

E' da notare che, poichè in questo sistema è possibile l'esame simultaneo di elementi di memoria nei vari PE, si può ottenere un accesso di tipo associativo alla memoria; ragione per cui il PEPE è considerato anche un sistema a processor associativo (Categoria /3,3./), e, più precisamente, viene inserito nella Categoria /3,3,1,2./ (Fully Parallel Distributed Logic Associative Processor). Nell'esame di tale categoria di Associative Processor verrà data una spiegazione più dettagliata del sistema PEPE.

CATEGORIA /3,3./ - Livello 2

Sistemi a processor associativo

A questa categoria appartengono le macchine SIMD in cui le unità di processo, che costituiscono specificatamente unità di processo di memoria, sono indirizzate, dall'unità di controllo centrale che esegue il flusso principale di istruzioni, in base a proprietà dei dati contenuti piuttosto che in base al loro indirizzo.



Da un punto di vista logico-funzionale, si definisce Associative Processor una macchina che possiede le due seguenti proprietà (8):

- 1) I dati memorizzati possono essere ritrovati usando il loro contenuto, o parte di esso, invece del loro indirizzo.
- 2) Le operazioni di trasformazione di dati, sia aritmetiche che logiche, possono essere eseguite sopra diversi insiemi di argomenti con una singola istruzione.

L'architettura complessiva di un sistema a processor associativo è illustrata, nella struttura essenziale, in Fig. 11.

Dal punto di vista hardware (22) (49), in ordine a poter ritrovare l'unità atomica di dati (data item) attraverso questa modalità, si deve essere in grado di accedere le parole di memoria facendo corrispondere (matching) il loro contenuto, o una determinata parte di esso, con una data chiave di ricerca delle parole (search key).

L'elemento basico della memoria associativa è la cella-bit (da un punto di vista logico). Essa ha la proprietà che l'informazione-bit può essere scritta, letta e comparata all'informazione di interrogazione. Le operazioni di ricerca, che consistono in mascheramento e comparazione, sono eseguite in una maniera che dipende dall'organizzazione della memoria associativa. La parola chiave di ricerca può essere comparata a tutte le parole in memoria, attraverso le linee pilotate di interrogazione a bit e la circuiteria della logica di comparazione.

La possibilità di far combaciare parole multiple ad una parola chiave di ricerca, richiede che la memoria associativa possieda qualche metodo per contrassegnare (tag) tutte le parole combacianti. La funzione di contrassegno e l'indicazione di combaciamento di parola sono eseguite dalla cosiddetta rete di contrassegno delle parole combacianti. Lo scopo fondamentale consiste nella possibilità di accedere a queste parole, segnalate combacianti, con una singola istruzione.

E' da notare che la memoria può avere la proprietà associativa eseguendo o una comparazione a parallelismo a bit (parallela a parola o seriale a parola) oppure seriale a bit.

In genere, nelle realizzazioni di memorie associative, sono consentite le seguenti operazioni di comparazione (22) (49):

- | | |
|-------------------|-----------------|
| - uguale | - diverso |
| - minore di | - maggiore di |
| - non maggiore di | - non minore di |
| - valore massimo | - valore minimo |

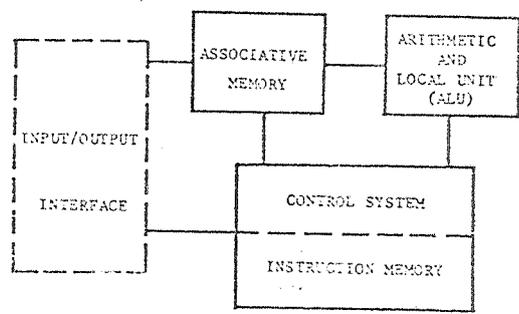


FIGURA 11

Diagramma a blocchi dell'architettura complessiva di un Processor Associativo.

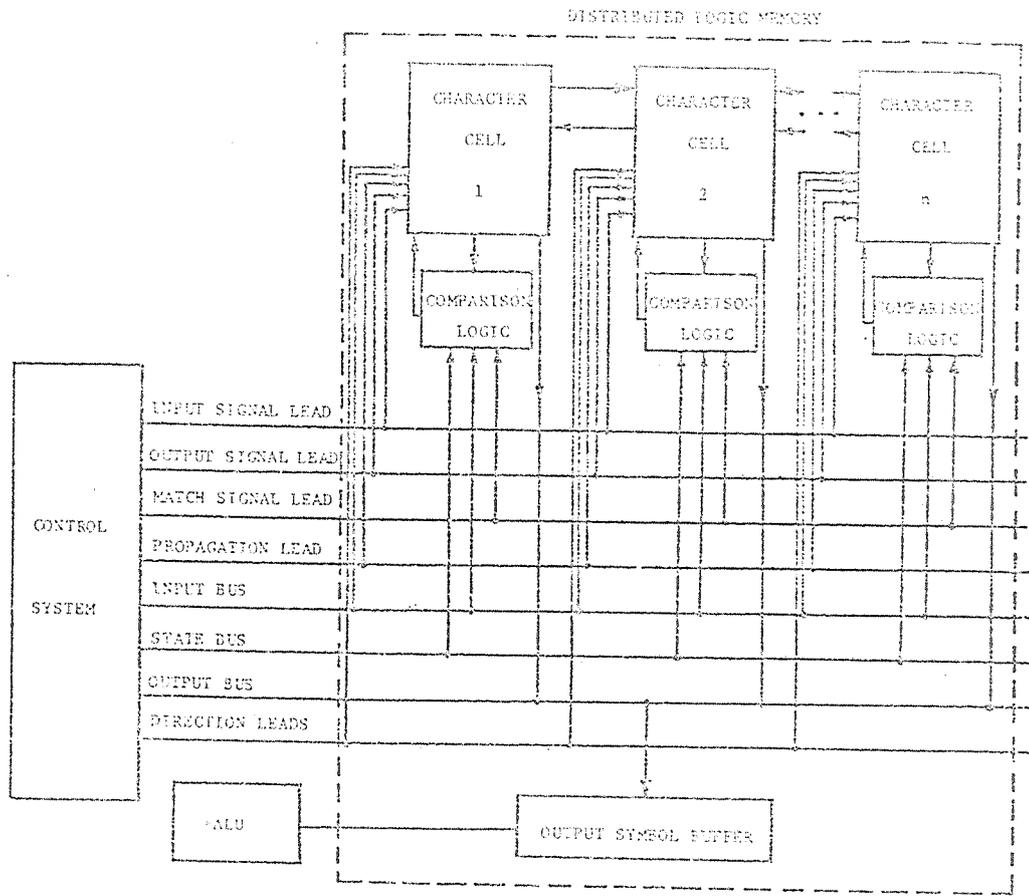


FIGURA 12

Architettura del Processor Associativo completamente parallelo a logica distribuita di Lee.

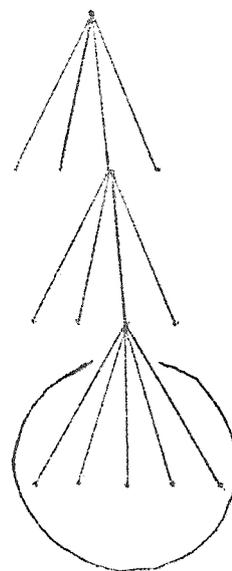
- tra i limiti
- non tra i limiti
- prossimo maggiore
- prossimo minore

Dal punto di vista operativo, un Processor Associativo può eseguire altre complicate operazioni di trasformazione di dati in aggiunta alle operazioni di comparazione eseguite dalla propria memoria associativa. Per esempio, le parole combacianti nella memoria associativa possono essere prese serialmente dall'Unità Aritmetico-Logica (ALU) centrale attraverso il circuito di uscita della memoria associativa, sotto la supervisione del sistema di controllo. La ALU esegue le operazioni di trasformazione di dati, specificate nel flusso principale di istruzioni, ed il risultato viene quindi memorizzato nella memoria associativa, se necessario (3).

CLASSIFICAZIONE /3,3,X/ - Livello 3
5 Categorie /3,3,X./, $X \in \{1,2,3,4,5\}$

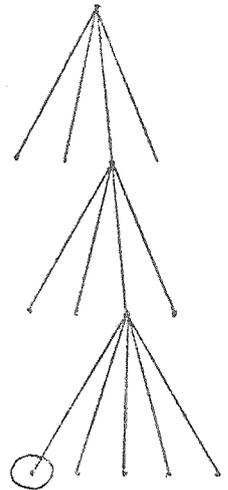
Questa classificazione dei sistemi a processore associativo è presa, per le prime quattro categorie, da Yau e Fung (8). Essi differenziano queste categorie in base ai diversi processi di combaciamento (nel senso precedentemente espresso) delle memorie associative dei sistemi ad esse appartenenti.

L'ultima categoria (Categoria /3,3,5./) è stata aggiunta, in accordo a quanto affermano Thurber e Wald (10), per differenziare speciali sistemi Associative Processor ad alto parallelismo interno.



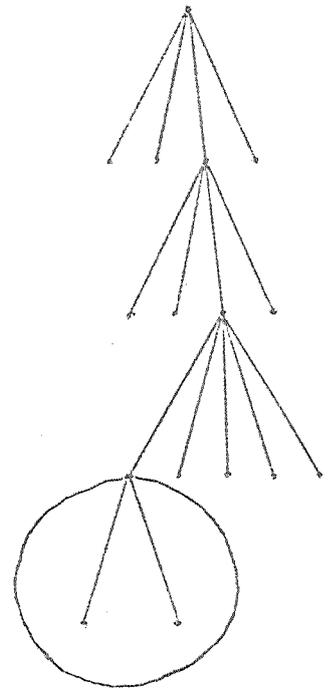
CATEGORIA /3,3,1./ - Livello 3
Sistemi a processor associativo,
completamente parallelo

Questa categoria è composta da sistemi a processor associativo caratterizzati da completo parallelismo della logica di comparazione della memoria associativa (Fully parallel associative processor)



CLASSIFICAZIONE /3,3,1,X/ - Livello 4
2 Categorie /3,3,1,X./, $X \in \{1,2\}$

I sistemi a processor associativo completamente parallelo si dividono in due categorie (3): nella prima, la logica di comparazione della memoria associativa è di tipo ad organizzazione a parola, nella seconda, essa è di tipo ad organizzazione a logica distribuita.

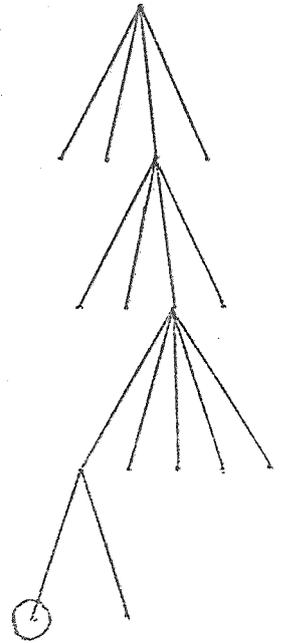


CATEGORIA /3,3,1,1./ - Livello 4
Sistemi a processor associativo,
completamente parallelo, organizzato
a parola

I sistemi di questa categoria sono caratterizzati dal fatto che la logica di comparazione è associata con ciascuna cella-bit di ciascuna parola della loro memoria associativa. La decisione logica è disponibile all'uscita di ogni parola. Si ha così che il loro processo di comparazione è eseguito in una maniera parallela a parola e parallela a bit.

Sebbene le operazioni nei sistemi di questa categoria sono le più semplici e le più veloci, comparate agli altri tipi di processors associativi, il loro hardware è anche il più complesso poiché ciascuna cella-bit deve contenere la logica di comparazione.

A causa della complessità del loro hardware, questa categoria di Associative Processors è stata sviluppata solo nelle prime fasi di studio e per modelli sperimentali, quali sistemi usanti, per la memoria associativa, componenti criogenici (Cryogenic associative memory) (50) (51) (52) e a logica cellulare tipo "cutpoint" (53).



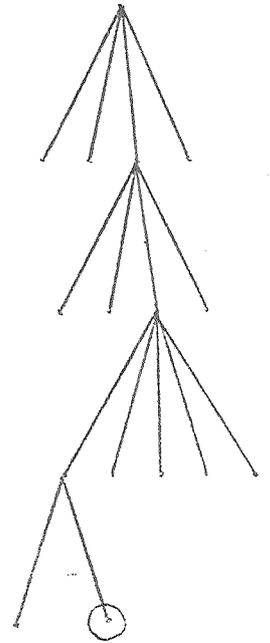
CATEGORIA /3,3,1,2./ - Livello 4
Sistemi a processor associativo,
completamente parallelo, a logica
distribuita

I sistemi di questa categoria sono caratterizzati dal fatto che la logica di comparazione è associata con ciascuna cella-carattere (per un numero fissato di bits) o con un gruppo di celle-carattere, della memoria associativa.

Il primo processor associativo di questo tipo è stato proposto da Lee, e prende il nome di Processor Associativo a Logica Distribuita di Lee (54); nella Fig. 12 è illustrata l'architettura di questo sistema.

Ciascuna cella-carattere ha un singolo elemento di stato-cella (parte stato) S, che può essere o in stato attivo o di riposo, e inoltre ha un certo numero di elementi simbolo-cella (parti simbolo) X_1, \dots, X_n , dipendentemente dalla dimensione dell'alfabeto di simboli. Ogni cella-carattere memorizza un simbolo-carattere di informazione e può comunicare con le sue due celle-carattere adiacenti, così come col sistema di controllo. Ogni stringa di informazione (blocco di dati) è quindi memorizzata in una corrispondente stringa di celle-carattere.

Logicamente, ogni blocco dati consiste di una stringa nome ed un arbitrario numero di stringhe parametri, ciascuno di questi elementi preceduto da un apposito contrassegno. Quando la chiave di ricerca di entrata è una stringa nome, la memoria a logica distribuita completamente parallela è attesa dare in uscita tutte le stringhe parametri associate con la stringa nome. Questo è il così detto ritrovamento diretto. D'altra parte, quando la chiave di ricerca d'entrata è una stringa parametro, la memoria a logica distribuita completamente parallela è attesa dare in uscita tutte le stringhe nome associate con questa stringa parametro.



Questo è chiamato ritrovamento incrociato.

Un sistema di Lee di questo tipo, consistente in 72 celle carattere di 8 bits, è stato costruito sperimentalmente usando componenti criogenici (55).

Il più importante esempio di sistema di questa categoria (Categoria /3,3,1,2./) è il sistema PEPE (Parallel Element Processing Ensemble), che costituisce anche uno dei due processors associativi (Categoria /3,3./) di grandi dimensioni (l'altro è lo STARAN, appartenente alla Categoria /3,3,2./) realizzati fino ad ora.

Il suo concetto architetturale basico è stato derivato dal processor associativo a logica distribuita di Lee ed è stato originariamente sviluppato dai Bell Laboratories per l'U.S. Army Advanced Ballistic Missile Defence Agency (48) (56).

Occorre notare che il sistema PEPE costituisce, per la sua struttura architettonica, anche il più importante esempio di Ensemble Processor (Categoria /3,2./), e, nell'ambito di questa categoria, è già stato presentato. In effetti non c'è contraddizione in questo fatto, essendo il PEPE un sistema articolato in maniera tale (poichè le funzioni dei suoi elementi di processo sono proprie di elementi di memoria associativa a logica distribuita, e contemporaneamente essi sono strutturati, nel complesso, come PE di un Ensemble Processor) da poter essere contemporaneamente classificato come appartenente alla Categoria /3,3,1,2./ ed alla Categoria /3,2./ (48).

Il sistema PEPE è composto, a livello della parte di controllo globale del sistema, dai seguenti sottosistemi funzionali (48):

- Controllo degli elementi di uscita (ODC);
- Controllo degli elementi di memoria (EMC);
- Unità di controllo aritmetica (ACU);
- Unità di controllo correlazioni (CCU);
- Unità di controllo di uscita associativa (AOCU);

- Sistema di controllo; composto a sua volta dai sottosistemi:
 - Logica di controllo sequenziale;
 - Coda di istruzioni parallele;
 - Unità di controllo istruzioni parallele.

Vi è poi nel sistema, un insieme di elementi di processo (PE), organizzati come unità a virgola mobile con parallelismo a 32 bits, ciascuno composto dalle seguenti sottounità:

- Unità aritmetica (AU);
- Unità di correlazione (CU);
- Unità di uscita associativa (AOU);
- Elemento di memoria (MEM) di 1024X32 bits.

In aggiunta, vi sono sottoinsiemi per convertire ed instradare segnali di controllo e di dati tra le varie unità funzionali.

Nella Fig. 10 è illustrata la struttura logica del sistema PEPE (come Ensemble Processor), mentre nella Fig. 13 è illustrata l'architettura di questo sistema.

Nel sistema PEPE gli elementi di processo costituiscono la componente computazionale principale. Nel funzionamento del sistema, porzioni selezionate del carico di elaborazione dati vengono caricate dal computer ospite (un CDC 7600) ai PE. Il processo di selezione di questo carico è determinato dal parallelismo inerente del task e dall'abilità della particolare architettura del PEPE di manipolare il task più efficacemente del computer ospite. A ciascun PE è delegata la responsabilità di un oggetto sotto osservazione da parte del sistema radar (infatti il PEPE è designato ad elaborazioni riguardanti la difesa radar da missili balistici); ciascun PE mantiene un file di dati, per specifici oggetti, dentro la sua memoria, ed usa le sue capacità aritmetiche per aggiornare continuamente il suo rispettivo file.

Un sistema PEPE con 288 PE, organizzati in 8 banchi (element bays), è presentato da Evensen e Troy (57).

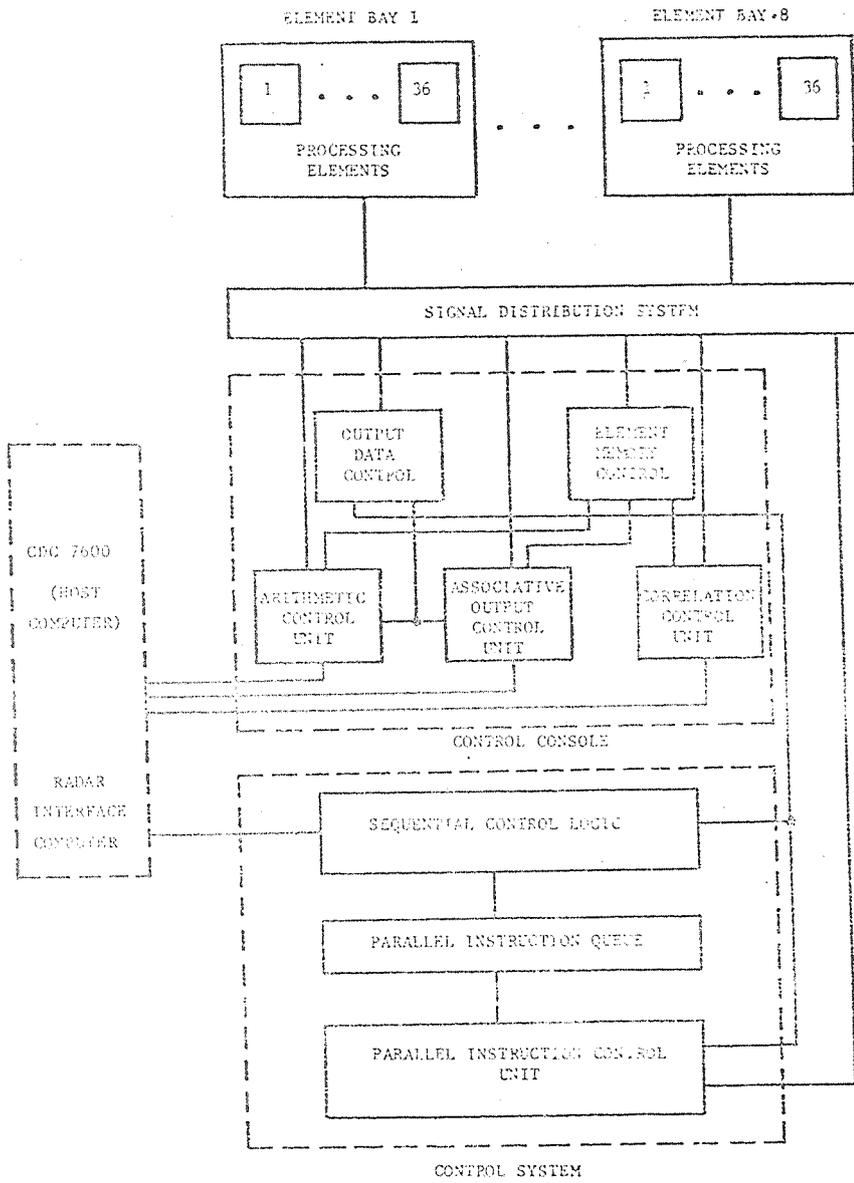
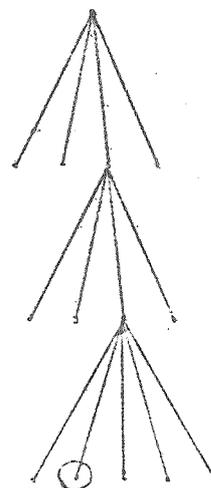


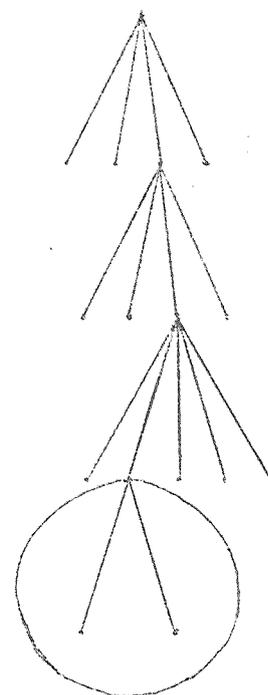
FIGURA 13
Architettura del sistema PEPE.

CATEGORIA /3,3,2./ - Livello 3
Sistemi a processor associativo,
parallelo a parola



A questa categoria appartengono i sistemi a processor associativo che usano il concetto dell'elaborazione parallela di dati verticali: una colonna-elementi di ogni parola è elaborata allo stesso tempo. I sistemi di questa categoria operano, quindi, serialmente per questi elementi, e parallelamente a parola.

CLASSIFICAZIONE /3,3,2,X/ - Livello 4
2 Categorie /3,3,2,X./, $X \in \{1,2\}$



I processors associativi con parallelismo a parola (Categoria /3,3,2./) si suddividono in due ulteriori classi, a seconda che gli elementi, costituenti ogni parola, che vengono elaborati serialmente per colonne a struttura verticale, siano costituiti da un solo bit oppure da un byte. Si ha così che, con questa classificazione (8), i sistemi appartenenti alla Categoria /3,3,2./ vengono suddivisi in processors associativi seriali a bit e paralleli a parola (Categoria /3,3,2,1./), e processors associativi seriali a byte e paralle-

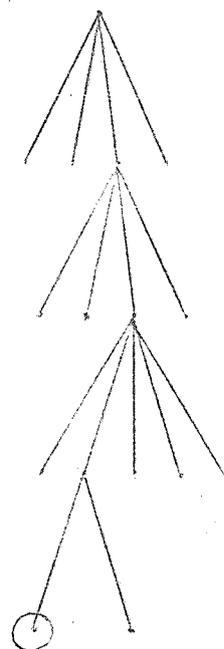
li a parola (Categoria /3,3,2,2./).

Di queste due ulteriori categorie, la prima, quella cioè dei sistemi a processor associativo seriale a bit e parallelo a parola, è di gran lunga la più importante.

CATEGORIA /3,3,2,1./ - Livello 4
Sistemi a processor associativo,
seriale a bit e parallelo a parola

A questa categoria appartengono i processors associativi in cui una colonna-bit di ogni parola è elaborata contemporaneamente. Per questa ragione, questi sistemi sono detti seriali a bit e paralleli a parola.

È opportuno notare, a questo punto, che una parte importante dei processors associativi seriali a bit, cioè appartenenti a questa categoria, è al rete di permutazione (cioè la "permutation network", detta anche "flin network"), che è il blocco funzionale per le comunicazioni tra i moduli di memoria e i moduli degli elementi di processo. Questa rete di permutazione non è usata solo nei sistemi di questa categoria, ma anche negli Array Processors (Categoria /3,1,1./), per preparare gli operandi appropriati per l'esecuzione. Infatti in molte applicazioni, quali sistemi di controllo del traffico aereo, calcoli matriciali, previsioni meteorologiche, ecc., risulta molto utile il preparare operandi appropriati usando la rete di permutazione.



Feng (61) ha proposto una rete di permutazione generalizzata, il manipolatore di dati ("data manipulator"). Vi sono due tipi di manipolatori di dati: il manipolatore di linee ed il manipolatore di pagine. Il primo è designato ad operare in maniera seriale a bit. Il secondo opera in maniera completamente parallela; esso è simile al manipolatore di linee, eccetto per il fatto che è costituito da N circuiti manipolatori di linee interconnessi insieme, dove N è l'ampiezza della parola. Il costo di implementazione di un manipolatore di pagine è ovviamente maggiore di quello di un manipolatore di linee, ma il suo uso permette ad un processor associativo seriale a bit (cioè appartenente a questa categoria), di operare come un processor associativo completamente parallelo (cioè appartenente alla Categoria /3,3,1./) per un costo molto inferiore rispetto a quello di un effettivo processor associativo completamente parallelo.

Il più importante esempio di processor associativo, parallelo a parola e seriale a bit, è lo STARAN (58) (59) (60) della Goodyear Aerospace Corporation. Tale sistema costituisce una delle più importanti realizzazioni (l'altra è il PEPE, appartenente alla Categoria /3,3,1,2./) di processors associativi (Categoria /3,3./) a larga scala.

La struttura basica dello STARAN è illustrata in Fig. 14. Lo STARAN consiste in un sistema di controllo ed in un certo numero (fino a 32) di moduli matrice associativa (59). Ciascun modulo matrice associativa contiene (si veda la Fig. 15) una memoria ad accesso multidimensionale di 256 parole di 256 bits, 256 semplici elementi di processo, una rete di permutazione (flip network) ed un selettore. Vi è un semplice elemento di processo per ciascuna delle 256 parole di memoria, e ciascun semplice elemento di processo opera serialmente per bit sui dati nella parola di memoria. Usando la rete di permutazione, si può accedere ai dati memorizzati nella memoria ad accesso multidimensionale, attraverso il

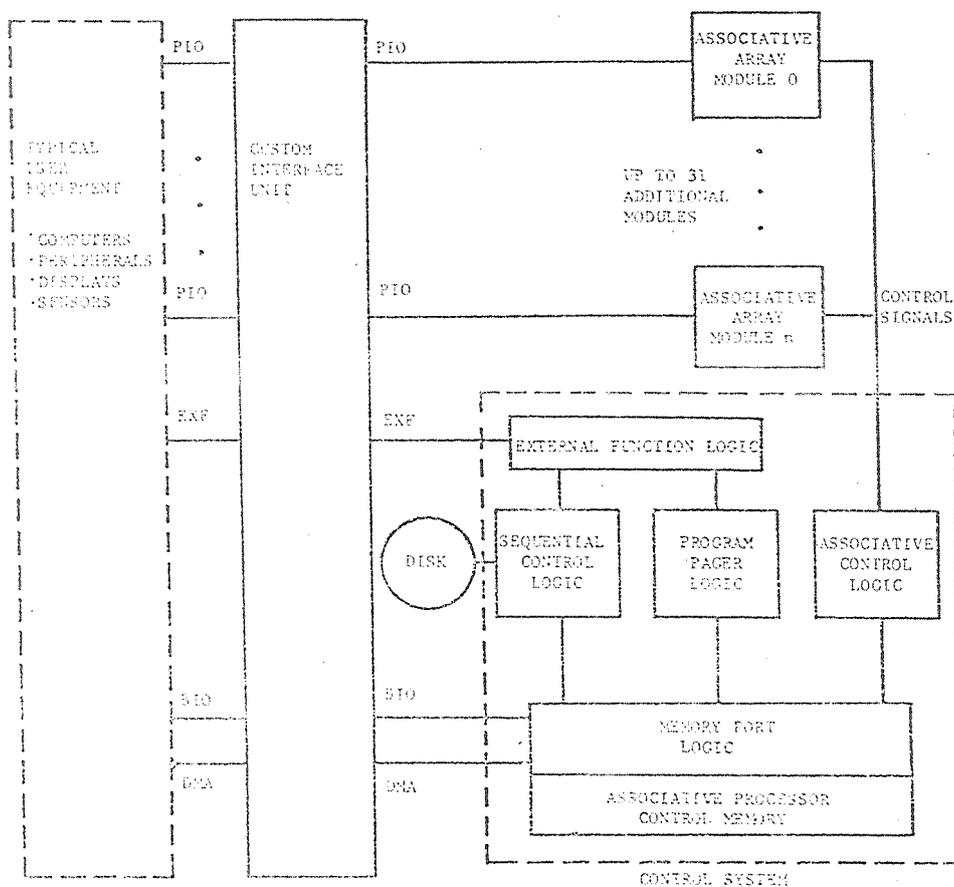


FIGURA 14
Architettura complessiva del sistema STARAN.

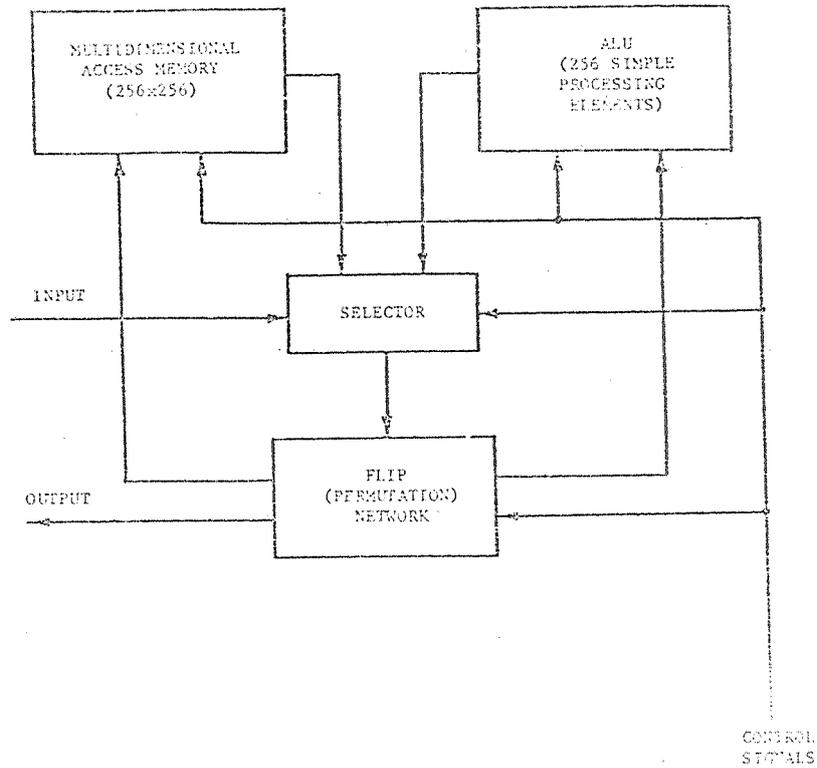


FIGURA 15

Architettura di un modulo-matrice associativa del sistema STARAN.

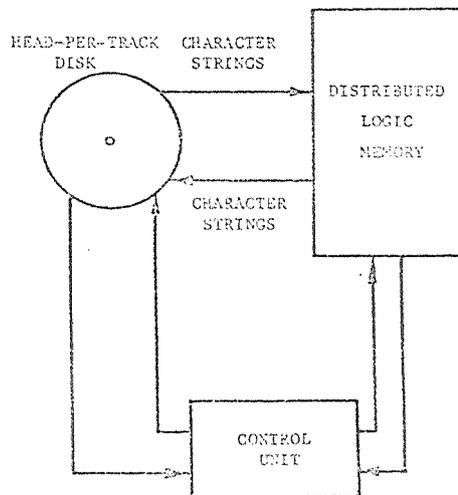


FIGURA 16

Struttura della memoria associativa orientata a blocchi usata nel sistema RAPID.

canale di I/O, in direzione bit-slice (un bit per volta di tutte le parole), o in direzione parola (una parola alla volta), oppure in una combinazione di esse. La rete di permutazione è anche usata per la traslazione ed il riordinamento di dati in un modulo matrice associativa, così che la ricerca parallela, le operazioni aritmetiche o logiche, possono essere eseguite tra parole della memoria ad accesso multidimensionale (60).

Per localizzare un particolare dato, lo STARAN inizia una ricerca richiedendo un combaciamento con un dato di entrata. Tutte le parole nelle memorie di tutti i moduli, che soddisfano il criterio di ricerca, sono identificate da una singola istruzione. I semplici elementi di processo eseguono simultaneamente operazioni, come specificato dalla logica di controllo associativa. Quindi, nell'esecuzione di una istruzione, i dati, in tutte le memorie selezionate di tutti i moduli, sono elaborati simultaneamente dai semplici elementi di processo ad ogni parola (59).

Oltre allo STARAN, diversi altri processors associativi, seriali a bit e paralleli a parola, sono stati sviluppati. Nel seguito, vengono brevemente presentati gli esemplari più importanti.

Il computer OMEN (43) costituisce un esempio di processor ortogonale, cioè appartenente alla Categoria /3,1,2./, ma, nello stesso tempo, può essere inserito in questa categoria (come già accennato nella discussione della Categoria /3,1,2./) per il suo funzionamento da processor associativo, seriale a bit, per quanto riguarda l'accesso di tipo verticale alla memoria ortogonale (si veda la Fig. 8).

Il processor associativo ibrido (Hybrid Associative Processor) di Love (53), sviluppato dalla Hughes Aircraft Company, contiene 10 memorie associative seriali a bit, e una memoria di massa MOS a registri di traslamento.

Il Raytheon Associative/Array Processor, o RAP (64), contiene una matrice di elementi di processo, così come un canale di accesso diretto

alla matrice, che facilita il trasferimento di dati di massa da e per la matrice di elementi di processo.

Il sistema ALAP (Associative Linear Array Processor) può essere considerato come un processor associativo seriale a bit ed a logica distribuita (65). Le celle parola formano, in esso, una linea di elementi di processo più memoria.

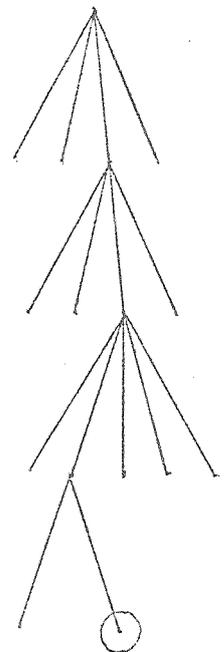
Infine l'ECAM (Extended Content Addressed Memory) della Honeywell (66) è stato sviluppato sulla loro tecnica "superchip" (a grande integrazione su singoli elementi semiconduttori) per ottenere matrici di vaste memorie associative di dimensioni fino ad un miliardo di bits, per sistemi di gestione di dati ad alta velocità.

CATEGORIA /3,3,2,2./ - Livello 4
Sistemi a processor associativo,
seriale a byte e parallelo a parola

A questa categoria appartengono i processors associativi in cui una colonna byte di ogni parola è elaborata contemporaneamente. Per questa ragione questi sistemi sono detti seriali a byte e paralleli a parola.

Questa concezione di funzionamento di sistemi associativi si pone tra quella seriale a bit e quella completamente parallela. Infatti la parola logica consiste di una logica di operazione byte piuttosto che di una logica di operazione bit, come nei sistemi a processor associativo seriale a bit (Categoria /3,3,2,1./).

Per ragioni di efficienza ad un costo ragionevole, un processor asso=



ciativo seriale a byte e parallelo a parola, chiamato Associative Processor Computer System (APCS), è stato proposto dalla System Development Corporation (62). Il sistema APCD contiene due unità di processo associative ed un canale parallelo di I/O.

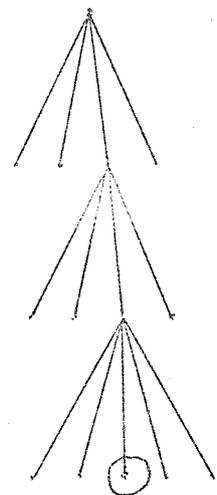
CATEGORIA /3,3,3./ - Livello 3
Sistemi a processor associativo,
seriale a parola

I processors associativi di questa categoria rappresentano essenzialmente una implementazione hardware di un semplice ciclo programma per la ricerca di una parola (3).

Il fattore importante che contribuisce alla relativa efficienza di questo approccio, in confronto alla ricerca a livello programma in un processor standard sequenziale, è che viene ridotto il tempo di decodifica delle istruzioni, poiché solo una singola istruzione è richiesta per eseguire una operazione di ricerca nel processor associativo seriale a parola.

A causa della bassa velocità delle memorie associative seriali a parola, sono stati sviluppati solo modelli sperimentali di questo tipo di sistemi.

Ad esempio, nel 1966 Crofut e Sottile (67) hanno presentato un processor associativo basato su di una memoria associativa che usava N linee di ritardo digitali ultrasoniche, dove N è il numero di bits per una parola.



Nel 1969 Rux (68) ha presentato una memoria associativa seriale a parola, con 35 linee di ritardo in cristallo, con capacità di memorizzazione di 2046 bits per linea (a 20.48 MHz), che era connessa ad un computer "general purpose" di medie capacità, chiamato NEBULA.

CATEGORIA /3,3,4./ - Livello 3
Sistemi a processor associativo,
orientato a blocchi

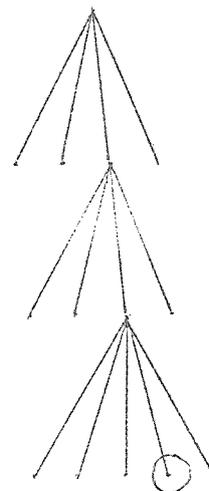
I processors associativi di questa categoria forniscono un compromesso tra l'alto costo di un processor associativo seriale a bit (Categoria /3,3,2,1./) e la bassa velocità di un processor associativo seriale a parola (Categoria /3,3,3./).

Un processor associativo orientato a blocchi usa una apparecchiatura rotante di memoria di massa, come un disco od un tamburo, per provvedere un grado limitato di capacità associative.

Un certo numero di sistemi a processor associativo orientato a blocchi è stato sviluppato e realizzato.

Parker ha presentato (69) il concetto di apparecchiature a logica-per-traccia che consistono in memorie a disco magnetico con testina-per-traccia (cioè con una testina di lettura/scrittura per ogni traccia fisica) che possiedono una certa quantità di capacità logiche associate con ciascuna traccia.

Basandosi su questo concetto e su quello di Lee (54) sulla memoria a logica distribuita (proprio della Categoria /3,3,1,2./), per applicazioni

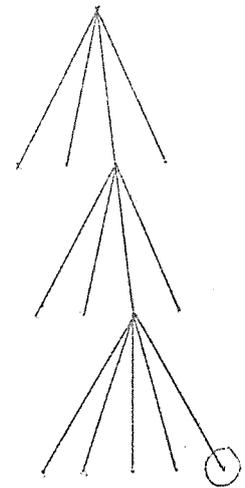


di memorizzazione e ritrovamento di informazioni, Parahmi (70) ha presentato un processor associativo, orientato a blocchi, chiamato RAPID (Rotating Associative Processor for Information Dissemination), la cui struttura è illustrata in Fig. 16. Poichè la velocità di trasferimento di dati tra dischi con testina-per-traccia e memoria a logica distribuita è piuttosto alta, il sistema RAPID è adatto per applicazioni che richiedono una grande capacità di memoria, le quali al presente soffrono del, relativamente, alto costo di memorie ad accesso casuale o della degradazione di prestazioni dovuta al frequente trasferimento tra memorie primarie e secondarie.

CATEGORIA /3,3,5./ - Livello 3
Sistemi a processor associativo,
altamente parallelo

Questa categoria viene inserita da Thurber e Wald (10) nell'ambito dei sistemi a processor associativo (Categoria /3,3./), per raccogliere, e distinguere dagli altri, speciali sistemi a processor associativo caratterizzati da alto parallelismo nel controllo interno. Tali sistemi, essenzialmente sperimentali, sono caratterizzati da processors che sono associativi ma che hanno o un controllo estensivo costruito dentro alla matrice di memoria, o unità di controllo multiple, anche se in genere dipendenti da un unico flusso di istruzioni (10).

Mantz (71) ha disegnato un processor associativo nel quale esistono possibilità aritmetiche estensive in ciascuna localizzazione di cella di



memoria associativa. Questa macchina è chiamata ACAM (Augmented Content-Addressed Memory). Le celle di questa matrice di memoria associativa hanno una complessità logica circa quattro volte superiore dell'ordine di complessità di una tipica cella di memoria di un processor associativo.

In questo modello sperimentale ciascun bit è memorizzato in una singola cella, ciascuna pilotata da tre ingressi (a,b,c), che rappresentano un valore programmabile determinante il funzionamento della matrice. I valori "c" esistono su base colonna e possono essere usati come maschere bit così che solo le colonne selezionate sono incluse nelle operazioni. I valori "a", "b" sono accoppiati su base parola, cioè esiste un valore "a-b" per ciascuna parola nella matrice. Essi servono per selezionare l'operazione da eseguire su quella particolare parola. L'intersezione dei valori "c", "a-b", ad una cella, specificano completamente la funzione che la cella deve eseguire.

Le funzioni della cella dell'ACAM di Kautz sono illustrate in Fig. 17. Una cosa importante è che varie parole della matrice possono eseguire contemporaneamente differenti operazioni; per esempio diversi test sulla parola di entrata possono avvenire simultaneamente. Il prezzo pagato per questo sta nella complessità realizzativa risultante.

In direzione orizzontale, la matrice può realizzare operazioni quali traslazione, entrata di colonna, uscita di colonna, ricerca per eguaglianza ed inclusione. In direzione verticale, la matrice permette traslazione di parola, I/O di parola, mascheramento di parola, complementazione ed altre operazioni. Funzioni quali staks, code, ecc. possono essere sintetizzate usando le colonne più a destra od a sinistra come registri di contrassegno.

Un particolare processor associativo, disegnato per il filtraggio di massa di dati radar, è stato descritto da Schmitz (72). Questo processor,

CELL FUNCTIONS			Z	P	C	X
c	b	a	$\bar{c} \oplus \bar{p} \bar{c} \bar{a} \oplus \bar{y}$	Z		R
0	0	0	$\bar{y} \oplus \bar{y}$	$\bar{z} \oplus z(x \oplus \bar{y})$		$\bar{z} \oplus y$
0	0	1	$y \oplus \bar{y}$	$\bar{z} \oplus z(x \oplus y)$		$\bar{z} \oplus y$
0	1	0	$\bar{y} \oplus \bar{y}$	$\bar{z} \oplus \bar{z}z \oplus yx$		$\bar{z} \oplus \bar{y}x \oplus yz$
0	1	1	$\bar{y} \oplus \bar{y}$	$\bar{z} \oplus z$		$\bar{z} \oplus x \oplus yz$
1	0	0	$\bar{y} \oplus \bar{z}y \oplus zx$	$\bar{z} \oplus z$		$\bar{z} \oplus x \oplus yz$
1	0	1	$\bar{y} \oplus \bar{z}y \oplus zx$	$\bar{z} \oplus z$		$\bar{z} \oplus y$
1	1	0	$\bar{y} \oplus \bar{x}y \oplus xz$	$\bar{z} \oplus \bar{z}z \oplus xy$		$\bar{z} \oplus x$
1	1	1	$y \oplus \bar{y} \oplus x \oplus yz$	$\bar{z} \oplus \bar{m} \bar{x}(x, y, z)$		$\bar{z} \oplus x$

FIGURA 17
Funzioni della cella dell'ACAI di Kautz.

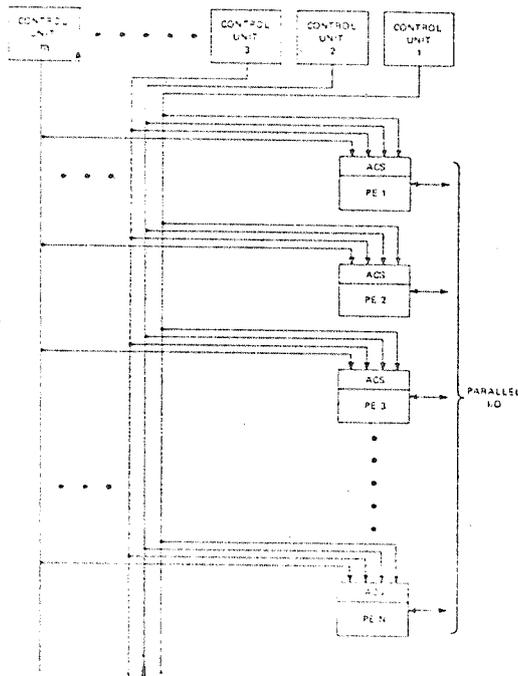


FIGURA 18
Architettura dell'Associative Control Switch di Schmitz.

illustrato in Fig. 13, possiede M unità di controllo (CU) e N elementi di processo (PE). L'elemento chiave di questa architettura è l'ACS (Associative Control Switch) che permette ad una delle CU di essere connessa a ciascuno dei PE. Così ciascun PE può essere connesso a qualsiasi delle M CU o può essere inattivo. Questo permette la suddivisione in sottinsiemi degli N PE in M processors paralleli e associativi. Questa connessione è perseguita associativamente, quindi questa macchina è un processor associativo.

Questa architettura tende a risolvere uno dei problemi di un processor associativo convenzionale, consistente nel fatto che mentre alcuni PE sono operanti, altri PE sono inattivi e così non contribuiscono alla soluzione del problema.

In questo tipo di processor associativo diverse porzioni del sistema possono eseguire operazioni concorrenti derivanti dallo stesso flusso di istruzioni, sfruttando maggiormente il sistema globale. Un'altra possibilità, nell'ambito di questo tipo di architettura, è quello di designare diverse porzioni del sistema ad eseguire operazioni concorrenti, relative a molteplici flussi di dati, orientandosi così in direzione dei sistemi MIMD (Categoria /4./), anche se ciò farebbe sorgere seri problemi di utilizzazione dell' hardware.

Seeber e Lindquist (73) hanno proposto un'architettura simile, in cui i PE erano processors associativi.



CATEGORIA /3,4./ (Livello 2)

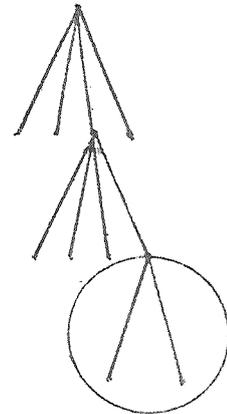
Sistemi SIMD distribuiti, non convenzionali

Questa categoria comprende quei sistemi SIMD a concezione non convenzionale, poiché spesso basati su idee che si discostano dalla tradizionale visione della macchina di Von Neuman, ed a struttura caratterizzata da accentuata decentralizzazione delle funzioni di controllo, e diversi possibili livelli operativi.

Tali sistemi, che si differenziano dagli altri sistemi SIMD in questi termini, sono stati raccolti in questa Categoria /3,4./ che viene così aggiunta alle altre 3 categorie della classificazione di Thurman (10).

CLASSIFICAZIONE /3,4,X/ (Livello 3)

2 Categoria /3,4,X./ , $X \in \{1,2\}$



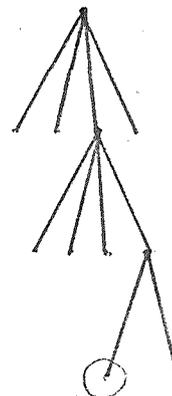
I sistemi S/D appartenenti alla Categoria /3,4./ si suddividono in due filoni di ricerca e sviluppo nettamente differenziati.

Il primo comprende i sistemi modulari microprogrammati (Categoria /3,4,1./), orientati ad una modularità di struttura che sfrutti la possibilità di disporre di componenti integrati (es. microprocessori) opportunamente microprogrammabili come blocchi elementari componenti il sistema.

Il secondo comprende i sistemi configurabili, detti anche sistemi data-flow (Categoria /3,4,2./), in cui il controllo è indipendente dal flusso di dati.

CATEGORIA /3,4,1./ (Livello 3)

Sistemi modulari micronrogrammati.



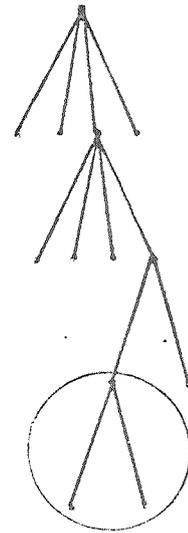
A questa categoria appartengono sistemi le cui funzioni sono decentralizzate in moduli, indipendenti e cooperanti, interconnessi in modo da permettere un elevato grado di espandibilità e modificabilità. Tali moduli sono principalmente costituiti da Unità Funzionali, ognuna specializzata o specializzabile, tramite forme di micronprogrammazione dinamica, verso ben determinati compiti nel sistema.

Ad esempio si possono avere, accanto ad unità funzionali di tipo aritmetico-logico, unità di controllo di I/O controllori di memoria, moduli che implementano subroutine di libreria, funzioni di sistema operativo, ecc..

I moduli, nell'ambito del sistema complessivo, per la funzionalità globale del sistema, interagiscono principalmente, oltre che indirettamente nella competizione di certe risorse, direttamente tramite scambio di messaggi.

CLASSIFICAZIONE /3,4,1,X/ (Livello 4)

2 Categorie /3,4,1,X./ , $X \in \{1,2\}$

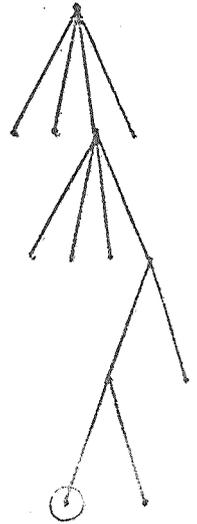


I sistemi modulari micropro-grammati si suddividono in due ulteriori categorie a seconda della strutturazione interna del controllo tra le varie unità funzionali.

Infatti i moduli costituenti tali sistemi possono cooperare in modo gerarchico (Categoria /3,4,1,1./), quando è possibile riconosce- re più di un livello di controllo, oppure in modo esecuto (Categoria /3,4,1,0./), quando non vi sono più livelli di controllo.

CATEGORIA /3,4,1,1./ (Livello 4)

Sistemi modulari microprogrammati a cooperazione gerarchica



A questa categoria appartengono Sistemi modulari microprogrammati in cui vi sono all'interno diversi livelli di controllo disposti gerarchicamente.

Nell'ambito di questa categoria ha particolare risalto il modello di organizzazione di sistema denominato "Macchina Logica", proposto da Kehl e Torode (76).

Lo schema strutturale è illustrato nella Fig.19. Questa struttura è composta essenzialmente da un Processor di Controllo (CP), e da un insieme di unità funzionali UF_1, UF_2, \dots, UF_n interconnesse attraverso un bus bidirezionale.

E' importante notare come una generica unità funzionale UF_1 può essere a sua volta realizzata come una Macchina Logica, e così via a più livelli.

Il PC agisce da schedulatore dei compiti affidati alle Unità Funzionali. Il suo scopo è quello di segnalare alle varie UF_i dipendenti quando devono iniziare a funzionare e su quali dati.

Questi dati vengono di regola trasmessi ad ogni Unità Funzionale da altre Unità Funzionali oppure, in certi casi, dallo stesso CP.

Quasi tutto, quindi, si preoccupa solo di innescare l'elaborazione delle varie UF_i , le quali poi proseguono l'elaborazione indipendentemente.

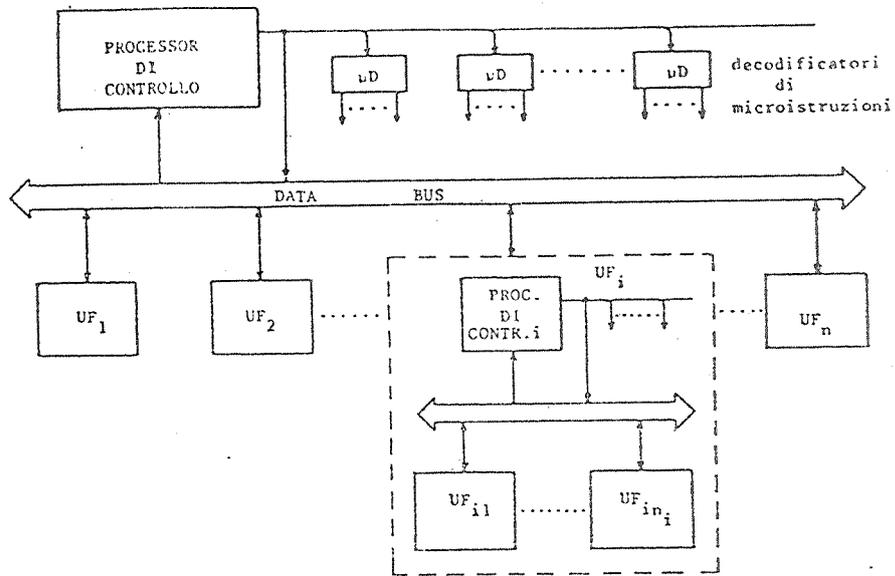


FIGURA 19
Schema strutturale della Macchina Logica di Kehl.

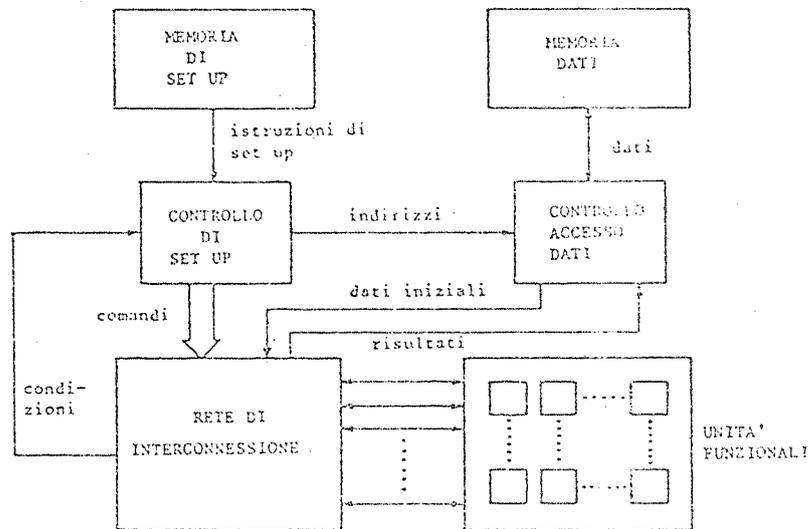


FIGURA 20
Schema strutturale di un sistema Data-flow di tipo "Interconnection Mode".

in generale in ogni istante vi saranno più Unità Funzionali che, sebbene innescate sequenzialmente dal Processor di Controllo, sono in fase di elaborazione contemporaneamente.

Il PC come le varie UF_i sono microprogrammabili; il PC è realizzato con microprogrammazione verticale mentre le UF_i sono realizzate con microprogrammazione orizzontale.

Questo modello organizzativo di sistema è stato usato per realizzazione del sistema, a scala minicomputer, chiamato LM² (Logical Machine Minicomputer) (77), il quale usa come Unità Funzionali una unità logica aritmetica di tipo generale, ed una serie di controllori di memoria principale e di dispositivi di I/O. La cooperazione tra Unità Funzionali e Processor di Controllo avviene attraverso un meccanismo di coroutines.

CATEGORIA /3,4,1,2./ (Livello 4)

Sistemi modulari micronprogrammati a cooperazione anonima



A questa categoria appartengono sistemi modulari micronprogrammati in cui non vi sono, all'interno, più livelli gerarchici di controllo, e quindi non esiste nessun modulo che agisca da "master" (controllore principale) nei confronti degli altri.

Un esempio di sistema appartenente a questa categoria è costituito dai "Macromoduli" di Clark (74).

In questo sistema ogni modulo segnala ad un altro, cui è connesso direttamente, di aver completato una funzione e che questo può servirsi dei dati in uscita dal primo per elaborarvi una nuova funzione.

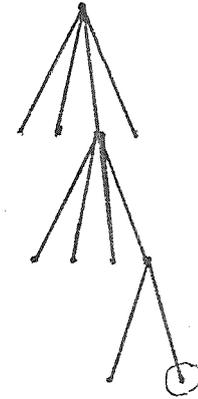
Un altro esempio di sistema modulare micronprogrammato, a cooperazione anonima, è costituito dai "Register Transfer Modules" (75), (Moduli a trasferimento di registri). In questo sistema vengono distinti moduli (unità funzionali) di elaborazione e moduli di controllo connessi da un bus bidirezionale; la programmazione del sistema è ottenuta sequenzializzando il funzionamento dei moduli di controllo, ognuno dei quali comanda l'operazione di un modulo di elaborazione.

Occorre però notare che queste architetture di sistemi si pongono ad un livello elementare di strutture macchina di dimensioni e capacità limitate. In entrambe queste proposte, poi, le funzioni di controllo sono realizzate direttamente in hardware, per cui una loro eventuale modifica implica un cambiamento della struttura di interconnessione (nei Macromoduli) o dei moduli di controllo (nei Register Transfer Modules).

CATEGORIA 3,4,2 (Livello 3)

Sistemi Configurabili(o Sistemi Data-Flow)

Tutte le classi di sistemi esistenti in generale conservano una caratteristica essenziale, cioè quella di essere calcolatori a programma memorizzato. Questo implica che i programmi devono essere opportunamente adattati, dal programmatore e dal software di base, alla struttura ed alla organizzazione del sistema a cui vengono affidati per l'esecuzione.



Questo fatto può provocare alcuni inconvenienti, come inefficienza e mancanza di flessibilità, derivati dalle costrizioni imposte dall'architettura e dall'organizzazione del sistema che possono non bene adattarsi alle modalità di funzionamento di un programma.

Ovviamente il massimo di prestazioni per un particolare programma sarebbe ottenibile con un sistema "special purpose" la cui architettura sia orientata alla struttura funzionale di quel programma.

Naturalmente un tale sistema non sarebbe assolutamente adatto per computazioni di tipo diverso.

Da queste considerazioni nasce l'idea di una macchina la cui struttura sia capace di modificarsi dinamicamente per adattarsi alla struttura della computazione da effettuare. Questo è il principio dei calcolatori configurabili o sistemi con controllo dipendente dal flusso di dati (Sistemi Data-Flow), appartenenti a questa categoria; tale principio è esattamente contrario a quello dell'organizzazione tradizionale tipo Von Neuman (73).

Infatti se supponiamo di avere la possibilità di stabilire tutte le possibili interconnessioni tra le unità funzionali a disposizione, il sistema assume un carattere di generalità tale da essere in

grado di effettuare una qualunque computazione che faccia uso di tali unità funzionali. Questa generalità è ottenuta senza rinunciare ai vantaggi di elevata velocità di calcolo tipica di una implementazione "special-purpose" concepita per un particolare problema; questa velocità è inoltre, in teoria, la massima possibile se dalla computazione sorgente è esplicitate il massimo parallelismo.

Inoltre, essendo l'esplicitazione del massimo parallelismo di una computazione dipendente solo da relazioni sui dati, e conseguentemente dipendendo l'abilitazione di un generico task (nel senso di attività funzionale interna per un singolo problema) solo dalla disponibilità dei dati, la più efficiente possibilità di implementare una computazione è quella in cui il controllo dell'abilitazione del task è dipendente solo dal flusso dei dati. (78).

Si possono ora schematizzare le caratteristiche principali per il funzionamento di un sistema configurabile. Per prima cosa, risulta la necessità di traduzione del programma sorgente in un linguaggio macchina tipo data-flow, in cui cioè le dipendenze tra le istruzioni siano funzione solo delle relazioni di interferenza sui dati di entrata e uscita delle istruzioni data-flow.

Questo significa che una istruzione è abilitata, e dunque può essere eseguita, non appena gli operandi che richiede sono prodotti da eventuali precedenti istruzioni.

Risulta poi la necessità di organizzare il sistema attorno ad un insieme di unità funzionali capaci di eseguire tutte le istruzioni del linguaggio macchina, e di dotarlo di opportune unità di controllo capaci di provocare dinamicamente l'interconnessione delle unità funzionali, secondo una struttura che riproduca esattamente quella del programma espressa in forma data-flow.

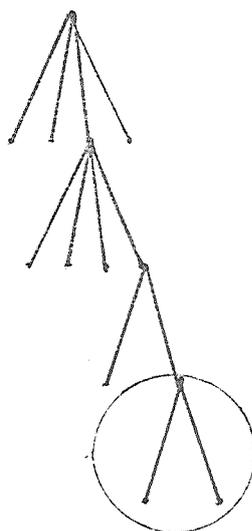
L'architettura teorica che ne risulta è dunque per definizione capace di esplicitare il massimo parallelismo insito nella computazione.

Vi sono state varie proposte di architetture di sistemi data-flow (79) (80), anche se per ora si rimane ad uno stato sperimentale di ricerca, in questa nuova direzione ancora da esplorare a fondo.

CLASSIFICAZIONE /3,4,2,X/ (Livello 4)

2 Categorie /3,4,2,X./ $X \in \{1,2\}$

Miller e Cocke hanno proposto(78) due modelli basici di architetture per sistemi configurabili o sistemi data-flow. I sistemi di questa categoria (Categoria /3,4,2./) vengono così suddivisi, in base a questa distinzione, in due ulteriori categorie : i sistemi data-flow di tipo "Interconnection Node" (Categoria /3,4,2,1./) ed i sistemi data-flow di tipo "Search Node" (Categoria /3,4,2,2./)

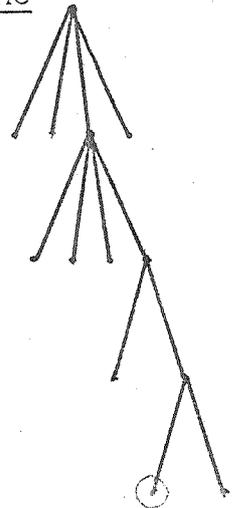


CATEGORIA 3,4,2,1. (Livello 4)

Sistemi data-flow a modalità di intercomunicazione

(Interconnection Mode)

Appartengono a questa categoria i sistemi data-flow in cui le unità funzionali sono connesse fisicamente attraverso una grossa rete di interconnessione (78) (79) (80). Lo schema strutturale della architettura dei sistemi di questo tipo è illustrata nella fig. 20.



In questi sistemi una Rete di Interconnessione Generale realizzata secondo una struttura tipo crossbar o multibus, ha lo scopo di stabilire tutti i possibili collegamenti tra le uscite e gli ingressi delle unità funzionali in modo tale da poter riprodurre la struttura della rappresentazione data-flow del programma.

In pratica, non essendo ovviamente possibile configurare un intero programma tutto in una volta, questo andrà decomposto in blocchi di dimensioni limitate, così che l'esecuzione del programma procederà per passi successivi, configurando ed eseguendo un blocco alla volta.

Tale decomposizione andrà effettuata in un modo che consenta di sfruttare al massimo le unità funzionali a disposizione e di minimizzare il numero di istanze di configurazione (ad esempio tentando di inserire interi loops in singoli blocchi). Questo processo sarà effettuato in fase di compilazione del programma sorgente.

Ogni blocco verrà codificato mediante una istruzione, detta istruzione di "set-up", eventualmente di lunghezza variabile, che dovrà specificare:

- una lista degli indirizzi dei dati in ingresso ed in uscita al blocco;
- le unità funzionali interessate e le loro interconnessioni;
- i blocchi che sono possibili successori diretti di quello in oggetto e le condizioni logiche in base alle quali deve essere selezionato un blocco successore.

Il programma in forma data-flow è dunque rappresentato da un insieme di istruzioni di Set Up, che vengono memorizzate nella Memoria Dati, che è in genere distinta dalla Memoria di Set Up.

L'interpretazione delle istruzioni di Set Up è demandata ad una unità di controllo detta Controllo di Set Up, la quale ha il compito di chiamare dalla Memoria di Set Up l'istruzione corrispondente al blocco da eseguire, decodificarla e stabilire le appropriate interconnessioni tra le unità funzionali inviando una serie di comandi alla Rete di Interconnessione. Essa inoltre invia gli indirizzi dei dati di ingresso e di uscita ad una unità di Controllo Accesso Dati, la quale si preoccupa di chiamare gli operandi e di inviarli alle unità funzionali tramite le vie stabilite nella Rete di Interconnessione.

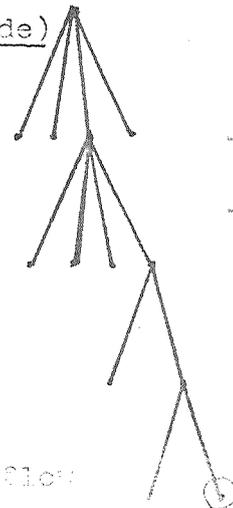
Alla fine dell'esecuzione del blocco, i risultati saranno inviati al Controllo Accesso Dati il quale provvederà alla loro memorizzazione; inoltre certe condizioni di terminazione saranno inviate al Controllo di Set Up, il quale potrà scegliere il successivo blocco da mandare in esecuzione. Per ragioni di efficienza il Controllo di Set Up potrà nel frattempo aver già letto le istruzioni di Set Up corrispondenti a tutti i possibili blocchi successivi.

È da notare come, una volta configurato il blocco e chiamati gli operandi iniziali, l'esecuzione proceda indipendentemente all'interno del complesso delle unità funzionali impiegando un tempo che dipende solo dalla velocità delle unità funzionali interessate e dai ritardi associati ai collegamenti stabiliti tra queste.

Si consegue così il vantaggio che il sistema si comporta come uno "special-purpose" durante l'esecuzione di ogni blocco, pur conservando le caratteristiche di un general-purpose. Un ulteriore vantaggio è che gli accessi in memoria sono ridotti al minimo, essendo eliminati sia per tutti i dati temporanei, sia per le istruzioni, tranne quelle di Set Up.

CATEGORIA/3,4,2,2./ (Livello 4)

Sistemi data-flow a modalità di ricerca (Search Mode)



Appartengono a questa categoria i sistemi data-flow in cui le unità funzionali sono connesse solo logicamente attraverso l'uso di uno speciale formato delle istruzioni in linguaggio macchina data-flow (78) (79) (80).

Lo schema strutturale di principio di un sistema data-flow di questo tipo è illustrato nella Fig.21 a.

In questi sistemi il controllo della computazione è effettuato da una unità di Controllo Abilitazione Istruzioni o "Search Unit" (SU) la quale ha il compito di cercare nella memoria le istruzioni abilitate. Ogni volta che essa trova una istruzione abilitata, la invia ad una unità funzionale in grado di eseguire l'operazione corrispondente. Quando una unità funzionale completa una istruzione, essa provvede, eventualmente tramite la SU, a scrivere il risultato in memoria; in questo modo viene prodotto un operando che può essere usato da una altra istruzione.

Nella fig.21 b è mostrato un tipico formato di una istruzione data-flow, con riferimento ad una operazione, specificata dal codice operativo, con due operandi e capace di produrre come risultato un dato per un'altra istruzione. I campi Operando 1 e Operando 2 sono designati a contenere il valore dei due operandi, mentre il campo Indirizzo di Risultato definisce la locazione in cui deve essere memorizzato il risultato; questa locazione sarà in generale quella di un campo operando di un'altra istruzione.

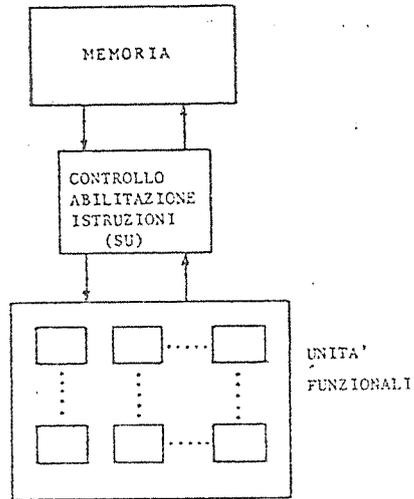


FIGURA 21 a
Schema strutturale di principio di un sistema Data-flow di tipo "Search Mode".

CODICE OPERATIVO	FLAGS 1	OPERANDO 1	FLAGS 2	OPERANDO 2	INDIRIZZO RISULTATO
------------------	---------	------------	---------	------------	---------------------

FIGURA 21 b
Formato di una istruzione Data-flow.

I campi Flag (contrassegno) contengono informazioni di controllo per specificare se l'operando associato è pronto e se può essere utilizzato. Se i campi Flag associati a tutti gli operandi indicano il verificarsi di queste condizioni, allora l'istruzione è abilitata e può essere inviata in esecuzione.

Quando questo avviene, i campi Flag vengono azzerati, indicando così che i nuovi valori di operandi dovranno essere prodotti affinché l'istruzione stessa possa essere ancora eseguita. In generale le istruzioni risultano essere a formato variabile.

Da un punto di vista architetturale, le prestazioni di un calcolatore data-flow tipo "Search Node" dipendono essenzialmente dalla realizzazione della unità di controllo SU.

È essenziale che questa unità sia strutturata in modo da consentire un alto parallelismo funzionale interno.

Una soluzione prospettata (80) consiste nell'organizzare questa unità di controllo mediante tre noduli (SU_1, SU_2, SU_3) indipendenti e cooperanti.

Un'altra soluzione (79) porta alle estreme conseguenze la decentralizzazione della SU, poiché ad ogni insieme di celle di memoria contenenti una istruzione è associato un piccolo controllo per la verifica dell'abilitazione dell'istruzione stessa e per il suo invio alle unità funzionali.

CATEGORIA /4./ (Livello 1)

Sistemi MIMD



Questa è la categoria dei sistemi con organizzazione a flussi multipli di istruzioni e a flussi multipli di dati.

A questa categoria appartengono sistemi in cui vi siano unità hardware distinte di elaborazione e in cui differenti processi possano essere in esecuzione (11).

Questi processi si possono concettualmente considerare come attività interne al sistema, corrispondenti ai flussi multipli paralleli di istruzioni da eseguire nel sistema.

Non si deve però pensare ad una corrispondenza biunivoca od almeno immediata tra flussi di istruzioni ed unità di elaborazione distinte nel sistema (nel senso che ognuna di queste è preposta all'elaborazione di un singolo flusso di istruzioni); in genere si ha una situazione dinamica per cui i flussi esterni di istruzioni generano attività interne al sistema, cioè processi che vengono poi eseguiti dalle varie unità interne di elaborazione secondo criteri specifici di efficienza globale.

Si ha così che i flussi paralleli effettivi di istruzioni interni al sistema sono costituiti dai passi successivi di esecuzione dei vari processi attivi sulle diverse unità di elaborazione. Analogamente i flussi paralleli effettivi di dati, interni al sistema, sono costituiti dagli insiemi di dati su cui tali processi in esecuzione operano.

In questo modo i sistemi MIMD si differenziano dai sistemi Uni-processori multiprogrammati. Infatti la situazione vista dall'esterno può considerarsi analoga: anche in quest'ultimo caso si ha una molte-

plicità di flussi esterni di istruzioni in relazione con una molteplicità di flussi di dati da elaborare, cioè si ha una situazione di flusso multiplo di lavori (multiple job streams).

Tuttavia la situazione si differenzia nettamente all'interno del sistema: nei sistemi Uniprocessor a differenze dei sistemi MIMD, ci può essere solo un processo in esecuzione in ogni momento (anche se con tecniche di multiprogrammazione e "time slice" nei sistemi operativi si vuole dare l'impressione agli utenti di una situazione di concorrenza tra i processi), per cui si ha effettivamente un solo flusso di istruzioni, interno al sistema, in esecuzione sull'unica Unità di Elaborazione (Central Processing Unit), operante su un unico flusso di dati interno.

Occorre poi far notare la differenza fondamentale tra i sistemi SIMD (Categoria /3./) ed i sistemi MIMD (Categoria /4./). Mentre nei sistemi SIMD si aveva una unica Unità di Controllo, globale per il sistema, destinata all'esecuzione dell'unico flusso di istruzioni, che pilotava le diverse Unità di Processo, che operavano su multipli flussi paralleli di dati, nei sistemi MIMD si hanno multiple Unità di Elaborazione, ciascuna con una sua Sottounità di Controllo, destinata al controllo dell'esecuzione dei flussi multipli paralleli interni di istruzioni, e con una sua Sottounità di Processo o Esecuzione, destinata ad operare sui corrispondenti flussi multipli paralleli interni di dati.

PRESENTAZIONE DELLA CLASSIFICAZIONE ADOTTATA PER I

SISTEMI MIMD

Si sono precedentemente caratterizzati i sistemi appartenenti alla Categoria /4./, cioè i sistemi MIMD, con un principio funzionale di ordine generale: un sistema per definirsi di tipo MIMD deve possedere più unità di elaborazione, che chiameremo (come per i sistemi SIMD) Processing Element (PE), ed ogni PE deve possedere una sua Parte di Controllo ed una sua Parte Operativa o di esecuzione (e questo li differenzia dai PE dei sistemi SIMD, infatti quei PE agivano sotto comando integrato ed erano perciò essenzialmente costituiti di una Parte Operativa) in modo da poter eseguire un particolare flusso di istruzioni sul rispettivo flusso di dati; questo significa che ogni PE è in grado di eseguire un particolare processo.

Questa è una definizione funzionale di principio che non discrimina in particolare sulla natura di questi PE, cioè sulla loro struttura e sulle loro capacità elaborative. Alla Categoria /4./ appartengono infatti, per esempio, sia sistemi multimicroprocessor in cui i PE sono microprocessor che operano ad un livello elementare di flussi di istruzioni/dati, sia reti di grossi elaboratori (in questo caso il sistema MIMD è la rete di computers vista nel suo complesso) in cui i singoli PE sono i vari computers della rete, quindi con capacità elaborative notevoli e con complessi flussi singoli di istruzioni/dati.

Tuttavia sistemi così diversi per dimensioni, prestazioni ed orientamenti di applicazioni si basano sul comune principio funzionale, sopra esposto, dei sistemi MIMD.

Non appare però conveniente basare una classificazione dei sistemi MIMD su distinzioni di carattere quantitativo, ad esempio basate su capacità o complessità dei singoli PE.

Il problema chiave per poter ben capire il vasto e differenziato panorama dei sistemi MIMD, è di considerare la funzionalità globale di questi sistemi come risultante complessiva della funzionalità delle singole unità di elaborazione. Infatti i vari PE, indipendentemente dal loro livello di capacità, cooperano ed interagiscono, cioè comunicano, in differenti modi, per poter ottenere la funzionalità del sistema complessivo.

Quindi si ha che l'approccio qualitativo più rispondente alla necessità di caratterizzare e classificare l'area dei sistemi MIMD è quello di esaminarli dal punto di vista delle modalità con cui le varie unità funzionali comunicano, e quindi interagiscono, all'interno del sistema.

Questo discorso si esplica, dal punto di vista architetturale, nell'esaminare le strutture di interconnessione dei PE in questi sistemi MIMD.

Per fare questo, ci si è basati su di un lavoro di Anderson e Jensen (11), nel quale si compie un esame completo delle varie strutture di interconnessione sperimentate e realizzate per sistemi di questo tipo (Categoria /4./).

Il metodo usato per identificare la struttura di interconnessione di un sistema MIMD consiste nell'isolare le maggiori unità, a livello hardware, coinvolte nel trasferimento di informazioni tra i processi sulle varie Unità di Elaborazione (cioè i PE). Questo trasferimento di informazione sarà chiamato "Trasmissione di messaggi", ed a questo scopo non si farà distinzione sulla natura di messaggi quali blocchi di dati, richieste di servizi, semafori, ecc..

Nella struttura di interconnessione propria si distingueranno due entità funzionali: i cammini di trasferimento (transfer paths) e gli elementi di computazione (switching elements).

Un cammino è il mezzo attraverso il quale un messaggio è trasferito tra i vari elementi del sistema.

Alcuni esempi di cammini sono cavi o bus, ponti radio, possibilità di trasmissione dati di reti pubbliche (ad esempio reti telefoniche), e memoria. La trasmissione di un messaggio su di un cammino implica che il messaggio sia teoricamente alterato.
^{non}

Un elemento di commutazione può essere visto come un'entità intelligente che interviene tra il mandante ed il ricevente di un messaggio. Un elemento di commutazione influisce sulla destinazione di un messaggio: alterando il messaggio (ad esempio cambiando il suo indirizzo di destinazione), instradandolo in uno dei diversi possibili cammini alternativi, o compiendo entrambe le azioni precedenti.

Occorre sottolineare il livello basico di questo approccio all'esame di strutture di interconnessioni, delle percorsi di messaggio, cammino e commutatore. Si ha così che nella strutturazione dei sistemi MMD, saranno considerate configurazioni di tre archetipi a livello di hardware: PE, cammini e elementi di commutazione (11).

Questo numero limitato di elementi fondamentali conduce a diverse semplificazioni che servono da un lato a rendere chiaro l'esame strutturale dei sistemi MMD, e dall'altro di non approfondire intenzionalmente gli aspetti del disegno di questi sistemi che non riguardano la struttura di interconnessione.

Per esempio non si farà distinzione tra un computer e la sua interfaccia con il resto del sistema, infatti entrambi sono parte del PE. Neppure si farà distinzione tra la commutazione di circuito e la commutazione di messaggio, infatti entrambe sono effettuate da una entità di commutazione.

La classificazione presentata da Anderson e Jensen (11) per i sistemi MMD è una classificazione gerarchica ad albero di quattro livelli, in cui i diversi tipi alternativi di architetture di sistemi, individuati da questa classificazione, sono rappresentati dalle foglie di questo albero.

Lo schema di questa classificazione a quattro livelli di Anderson e Jensen, è illustrato nella fig.22.

I livelli successivi di questo albero possono essere visti come livelli successivi di decisioni che si devono prendere nella specifica progressiva del disegno di un sistema ad interconnessione di moduli PE distinti.

La radice di questo albero (1° livello) è costituita dalla decisione di interconnettere un certo numero di Unità di Elaborazione con lo scopo di avere una completa intercomunicazione per poter costituire nel complesso un sistema di tipo MFD.

Sotto stanno i livelli di decisione che rappresentano la scelta della strategia di trasferimento messaggi (2° livello), la scelta del metodo per il controllo di tali trasferimenti (3° livello), e la scelta del tipo di cammino su cui il trasferimento deve essere fatto (4° livello).

I primi due livelli concernono decisioni di ordine strategico, cioè di politica di scelta strutturale del sistema, mentre il terzo ed il quarto livello concernono decisioni di ordine tattico, cioè di implementazione del sistema.

Con la scelta iniziale, al primo livello, ci si pone nella Categoria /4./ della nostra Classificazione Generale. In effetti la classificazione ad albero a quattro livelli di Anderson e Jensen (Fig.22) è stata completamente inserita nella Classificazione Generale.

Al primo livello si fa corrispondere la categoria che raccoglie le architetture di sistema di tipo MFD (Cat./4./), i livelli successivi, fino al quarto, corrispondono alle successive articolazioni dei sistemi MFD, classificati in base alle loro caratteristiche di interconnessione delle Unità di Elaborazione. All'ultimo livello della classificazione della Fig.22, il quinto, vi sono le distinte categorie in cui si riscontra siano suddivisi i tipi di strutture di sistemi MFD.

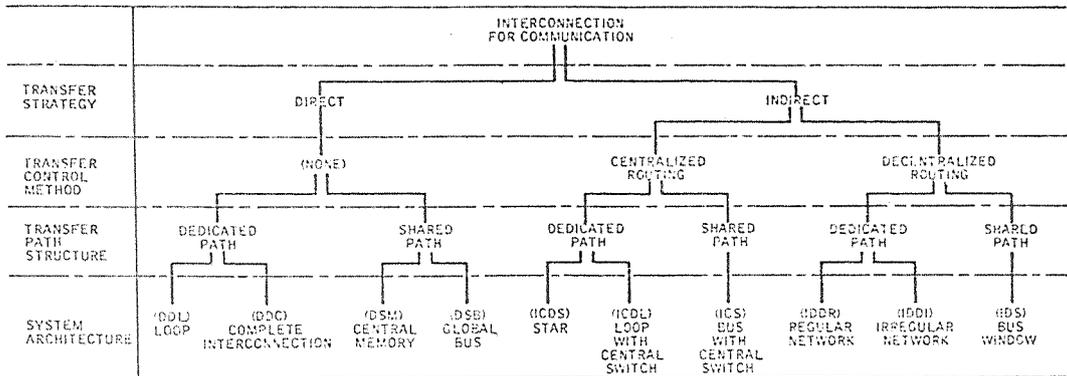


FIGURA 22

Schema della classificazione a quattro livelli di Anderson e Jensen applicata ai sistemi MIMD.

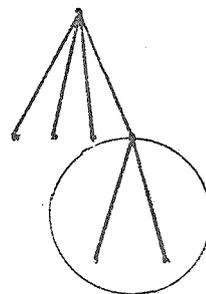
Infine due categorie di livello cinque, corrispondenti ai due tipi più importanti di sistemi MIMD, cioè i sistemi Multiprocessors e i sistemi a Rete di Elaboratori, sono state ulteriormente suddivise con, rispettivamente, un altro livello, per la prima categoria, e due altri livelli, per la seconda, ^{di} specifica in altre particolari categorie.

Nell'ambito della nostra classificazione Generale si ha così che, per la Categoria /4./, le classificazioni di livello due, tre e quattro servono ad articolare, per successivi livelli di decisione nel disegno delle architetture, le strutture di interconnessione nei sistemi MIMD in categorie intermedie successive (livelli due, tre e quattro) che, in effetti, hanno valore di passi intermedi nella delimitazione specifica dei tipi di sistemi MIMD, studiati e realizzati, che si ritroveranno solo nelle categorie del livello cinque.

Per questa ragione, nella presentazione successiva della Classificazione Generale di architetture non si darà una spiegazione approfondita per le categorie dei livelli due, tre e quattro, ma ci si limiterà alla presentazione della loro funzione di articolazioni successive di strutture di tipo MIMD.

CLASSIFICAZIONE /4,X/(Livello 2)

2 Categorie /4,X./ X ∈ {1,2}



La prima scelta strategica è tra la trasmissione diretta di messaggi dalla sorgente alla destinazione, e la loro trasmissione indiretta in cui una operazione di intervento è richiesta.

Per i nostri propositi di classificazione, il criterio per distinguere tra queste due alternative è l'esistenza di una o più entità di commutazione che prendono decisioni per ogni messaggio.

Così, circuiti di intervento con funzione ripetitiva o elementi di memoria sono semplicemente generi differenti di cammini e non influenzano sulla natura diretta della comunicazione. Ma una entità di intervento che alteri i messaggi, ad esempio con una trasformazione di indirizzo, o che instradi i messaggi su di un particolare cammino di uscita, tra i numerosi cammini alternativi presenti, agisce ad un livello di comunicazione indiretta.

Un altro modo di fare questa distinzione è di determinare se l'informazione di controllo è posseduta dall'entità di intervento oppure è inviata a tale entità (tale informazione di controllo può essere costituita da tavole di trasformazione di indirizzi).

Decisioni prese dal mandante del messaggio (per esempio, riguardanti su quale porto trasmettere) e decisioni prese dal ricevente il messaggio (ad esempio se accettare un dato messaggio) non influenzano la natura diretta della comunicazione.

In conclusione questa Classificazione /4,X/ suddivide la Categoria /4./ dei sistemi MFD nella Categoria /4,1./ dei sistemi D-MFD, in cui la strategia di trasferimento messaggi è di natura diretta, e nella Categoria /4,2./ dei sistemi I-MFD, in cui la strategia di trasferimento messaggi è di natura indiretta.

CATEGORIA /4,1./ (Livello 2)

Sistemi D-MIND



CATEGORIA /4,1,1./ (Livello 3)

Sistemi D-MIND



Sia la Categoria /4,1./ (livello 2) che la Categoria /4,1,1./ (livello 3) sono composte dai Sistemi MIND a Strategia Diretta di Trasferimento Messaggi (Sistemi D-MIND).

Il fatto che entrambe queste categorie, di cui una padre della altra, siano designate a contenere lo stesso insieme di sistemi, e siano quindi coincidenti dal punto di vista del contenuto, si spiega per ragioni di completezza di classificazione (sono distinte solo per il diverso livello nella Classificazione Generale).

Infatti si ha che la Categoria /4,1./ era stata preposta dalla Classificazione /4,X/, a livello due, a contenere i sistemi D-MIND, distinguendola dalla Categoria /4,2./, destinata a contenere i sistemi I-MIND, in base alla diversa strategia di trasferimento messaggi (di natura diretta per i sistemi D-MIND, di natura indiretta per i sistemi I-MIND).

Poi, a livello tre, nella classificazione di Anderson e Jenson (11) adottata, si prevede una articolazione ulteriore delle strutture di interconnessione dei sistemi MIND, in base al metodo di controllo

di trasferimento di messaggi. Quindi, in questo ambito, una classificazione di livello tre, tra cui la Classificazione /4,1,X/ , dovrebbe operare distinzioni con considerazioni di questo tipo.

Tuttavia si ha che nei sistemi della Categoria /4,1./ , essendo sistemi P-MND, i messaggi sono inviati in maniera diretta dalla sorgente alla destinazione, senza che, per loro definizione, possa venire adottato nessun tipo di metodo di controllo del trasferimento di questi messaggi.

Da ciò risulta che per la Categoria /4,1./ non ci può essere nessuna distinzione in sottocategorie tramite una classificazione: tuttavia, si è arrivata la Classificazione /4,1,X/ , che definisce solo la categoria /4,1,1./ la quale contenga ancora i sistemi P-MND, come la Categoria /4,1./ sopra.

CATEGORIA /4,2./ (Livello 2)

Sistemi I-MIND

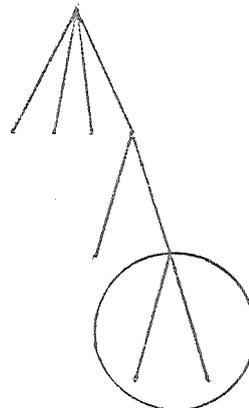
A questa categoria appartengono i sistemi MIND a Strategia Indiretta di Trasferimento Messaggi.



CLASSIFICAZIONE /4,2,X/ (Livello 3)

2 Categorie /4,2,X/ $X \in \{1,2\}$

Nel caso sia scelta a livello due una comunicazione tra Unità di Elaborazione, di tipo indiretto (Categoria /4,2./ dei sistemi I-MIND), deve essere presa una ulteriore decisione riguardo il metodo di trasferimento dei messaggi.



Si ha così che al terzo livello si attua una articolazione delle strutture di interconnessione dei sistemi I-MIND in base al metodo di controllo di trasferimento dei messaggi.

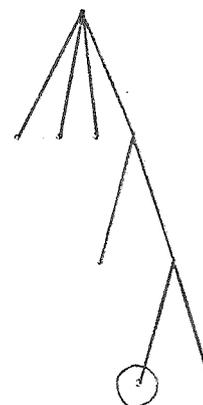
Le alternative sono due: la centralizzazione, in cui una singola entità presiede alla commutazione di tutti i messaggi, e la decentralizzazione, in cui interviene un certo numero di entità di commutazione.

Questa decisione corrisponde, al livello tre, la Classificazione /4,2,X/ della Categoria /4,2./, dei sistemi I-MIND, in due categorie: la Categoria /4,2,1/ dei sistemi IC-MIND, in cui l'instradamento dei messaggi è di natura centralizzata, e la Categoria /4,2,2./, dei sistemi DC-MIND, in cui tale instradamento è di natura decentralizzata.

CATEGORIA /4,2,1./ (Livello 3)

Sistemi IC-MIND

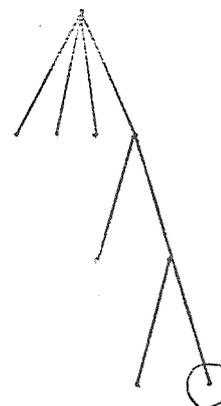
A questa categoria appartengono sistemi MIND a Strategia Indiretta di Trasferimento Messaggi (I-MIND) ed a Metodo Centralizzato di Controllo Trasferimento (IC-MIND, cioè a commutazione centralizzata di messaggi).



CATEGORIA /4,2,2./ (Livello 3)

Sistemi ID-MIND

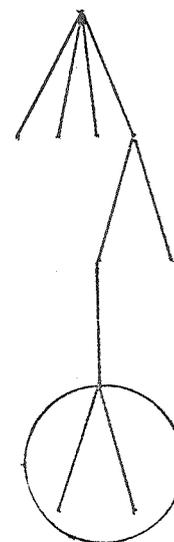
A questa categoria appartengono sistemi MIND a Strategia Indiretta di Trasferimento Messaggi (I-MIND) ed a Metodo Decentralizzato di Controllo di Trasferimento (ID-MIND, cioè a commutazione decentralizzata di messaggi).



CLASSIFICAZIONE /4,1,1,X/ (Livello 4)

2 Categorie /4,1,1,X/ , X ∈ {1,2}

Il quarto livello riguarda la scelta del tipo di struttura dei canali di trasferimento dei messaggi.



Tale scelta è compiuta tra due possibili alternative, cioè cammini di trasferimento dedicati e cammini di trasferimento condivisi (shared).

Si definisce un cammino condiviso come un cammino che è accessibile da più di due punti. In realtà, vi sono almeno tre alternative che possono essere distinte: cammini che sono unidirezionali punto-a-punto, cammini che sono bidirezionali punto-a-punto, cammini che sono bidirezionali e visitano più di due punti.

Nel primo caso non può avvenire nessun tipo di contesa per la risorsa; ma nel secondo esiste una rudimentale specie di condivisione, per cui può avvenire un certo grado di contesa per la risorsa; comunque nel terzo caso, la contesa della risorsa diventa la considerazione maggiore, per cui noi definiamo questo caso come caso di cammino condiviso e gli altri due come caso di connessioni dedicate.

E' opportuno ripetere che la nozione di cammino (path) non implica una particolare implementazione, e sia bus che memorie possono essere usati appropriatamente come cammini di trasferimento di messaggi. Inoltre va notato che cammini che sono ridondanti per tolleranza a guasti ('fault tolerance') o per ragioni di ampiezza di banda di trasmissione (bandwidth) sono qui considerati singoli, da un punto di vista logico.

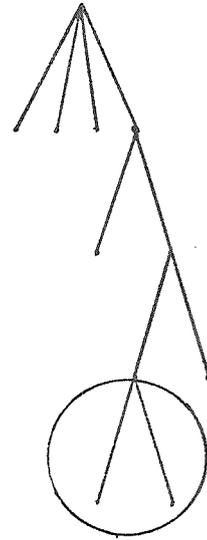
E' importante sottolineare che queste considerazioni sono comuni a tutte le classificazioni, a livello quattro dei sistemi MIMD, e quindi si intendono riferite a ciascuna di esse.

La scelta del tipo di struttura dei cammini di trasferimento dei messaggi determina, al livello quattro, la Classificazione /4,1,1,X/ della Categoria /4,1,1./, dei sistemi D-MIMD, in due categorie: la Categoria /4,1,1,1./, dei sistemi DD-MIMD, in cui i cammini di trasferimento dei messaggi sono di tipo dedicato, e la Categoria /4,1,1,2/ dei sistemi DD-MIMD, in cui i cammini di trasferimento dei messaggi sono di tipo condiviso.

CLASSIFICAZIONE /4,2,1,X/ (Livello 4)

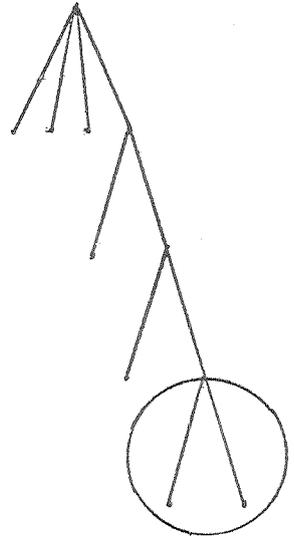
2 Categorie /4,2,1,X./, $X \in \{1,2\}$

La scelta del tipo, se dedicato o condiviso, di struttura dei cammini di trasferimento dei messaggi determina, al livello quattro, la Classificazione /4,2,1,X/ della Categoria /4,2,1./, dei sistemi IC-MIND, in due Categorie: la Categoria /4,2,1,1./, dei sistemi ICD-MIND, in cui i cammini di trasferimento dei messaggi sono di tipo dedicato, e la Categoria /4,2,1,2./, dei sistemi ICS-MIND, in cui i cammini di trasferimento dei messaggi sono di tipo condiviso.



CLASSIFICAZIONE /4,2,2,X./ (Livello 4)

2 Categorie /4,2,2,X./ , $X \in \{1,2\}$



La scelta del tipo (se dedicato o condiviso) dei cammini di trasferimento dei messaggi determina, al livello quattro, la Classificazione /4,2,2,X/ della Categoria /4,2,2./ dei sistemi IDS-NIDS in due categorie: la Categoria /4,2,2,1./ , in cui i cammini di trasferimento dei messaggi sono di tipo dedicato, e la Categoria /4,2,2,2./ , dei sistemi IDS-NIDS, in cui i cammini di trasferimento dei messaggi sono di tipo condiviso.

CATEGORIA /4,1,1,1./ (Livello 4)

Sistemi DD-MIMD

A questa categoria appartengono i sistemi MIMD a Strategia Diretta di Trasferimento Messaggi ed a Cammini di Trasferimento Dedicati.



CATEGORIA /4,1,1,2./ (Livello 4)

Sistemi DS-MIMD

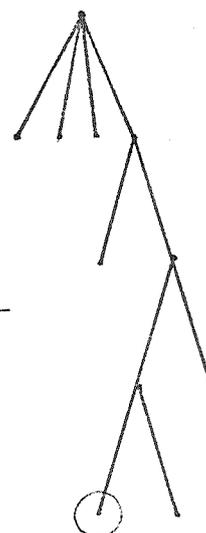
A questa categoria appartengono i sistemi MIMD a Strategia Diretta di Trasferimento Messaggi ed a Cammini di Trasferimento Condivisi.



CATEGORIA /4,2,1,1./ (Livello 4)

Sistemi ICD-MIMD

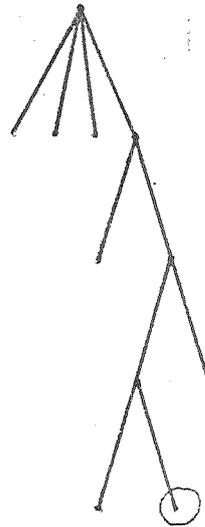
A questa categoria appartengono i sistemi MIMD a Strategia Indiretta di Trasferimento Messaggi, a Metodo Centralizzato di Controllo Trasferimento e a Cammini di Trasferimento Dedicati.



CATEGORIA /4,2,1,2./ (Livello 4)

Sistemi ICS-MIND

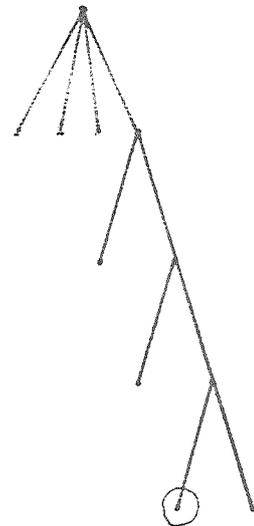
A questa categoria appartengono i sistemi MIND a Strategia Indiretta di Trasferimento Messaggi, a Metodo Centralizzato di Controllo Trasferimento e a Cammini di Trasferimento Condivisi.



CATEGORIA /4,2,2,1./ (Livello 4)

Sistemi IDD-MIND

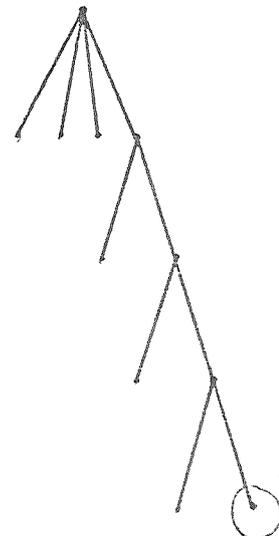
A questa categoria appartengono i sistemi MIND a Strategia Indiretta di Trasferimento Messaggi, a Metodo Decentralizzato di Controllo Trasferimento ed a Cammini di Trasferimento Dedicati.



CATEGORIA /4,2,2,2./ (Livello 4)

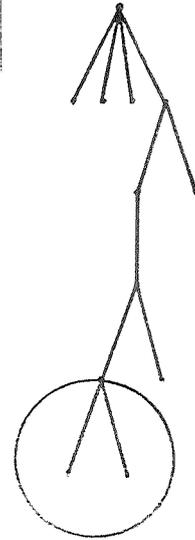
Sistemi IDS-MIND

A questa categoria appartengono i Sistemi MIND a Strategia Indiretta di Trasferimento Messaggi, a Metodo Decentralizzato di Controllo Trasferimento ed a Cammini di Trasferimento Condivisi.



CLASSIFICAZIONE /4,1,1,1,X/ (Livello 5)

2 Categorie /4,1,1,1,X./, X ∈ {1,2}



Il quinto livello, l'ultimo della classificazione ad albero di Anderson e Jensen (11), prevede la suddivisione delle categorie di livello quattro, che sono le categorie risultanti dall'articolazione dei sistemi MFD dal punto di vista della struttura di interconnessione interna (a livello due per la strategia di trasferimento messaggi, a livello tre per il metodo di controllo di tale trasferimento, a livello quattro per la struttura dei cammini usati per tale trasferimento), in categorie, di livello cinque, che contengono i tipi fondamentali, studiati e realizzati, di sistemi MFD.

Le categorie strutturali, identificate fino a questo punto, di livello quattro, vengono così classificate, con queste classificazioni di livello cinque, in categorie realizzative di livello cinque, le quali costituiscono le foglie dell'albero di classificazione di Anderson e Jensen.

Nei paragrafi seguenti, nell'esame di queste categorie realizzative di quinto livello, saranno discussi i vari sistemi MFD, sperimentati e applicati nella realtà.

Questa rassegna si ritiene riferita a tutte le classificazioni di livello cinque, riferite ai sistemi MFD.

La classificazione /4,1,1,1,X/ della categoria strutturale /4,1,1,1./, dei sistemi DD-MIMO, identifica due categorie realizzative di livello cinque: la Categoria /4,1,1,1,1./, dei sistemi DDL-MIMO, cioè dei sistemi DD-MIMO ad Architettura ad Anello, e la Categoria /4,1,1,1,2./, ^{dei sistemi DDC-MIMO, cioè} dei sistemi DD-MIMO ad Architettura ad Interconnessione Completa.

CLASSIFICAZIONE /4,1,1,2,X/ (Livello 5)

2 Categorie /4,1,1,2,X./, X ∈ {1,2}

Questa classificazione della Categoria strutturale /4,1,1,2./, dei sistemi DS-MIMD, identifica due categorie realizzative di livello cinque:

La Categoria /4,1,1,2,1./, dei sistemi DNS-MIMD, cioè dei sistemi DS-MIMD ad Architettura a Memoria Centrale, e la Categoria

/4,1,1,2,2./, dei sistemi DSR-MIMD, cioè dei sistemi DS-MIMD ad Architettura a Memoria Centrale.



CLASSIFICAZIONE /4,2,1,1,X/ (Livello 5)

2 Categorie /4,2,1,1,X./, X ∈ {1,2}

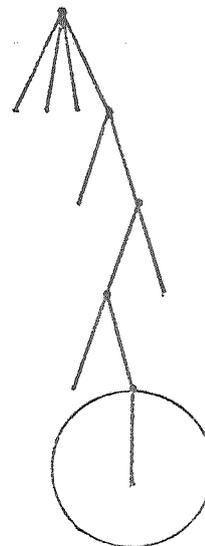
Questa classificazione della Categoria strutturale /4,2,1,1./, dei sistemi ICD-MIMD, identifica due categorie realizzative di livello cinque: la Categoria /4,2,1,1,1./, dei sistemi ICD-MIMD ad Architettura a Sella, e la Categoria /4,2,1,1,2./, dei sistemi ICDI-MIMD, cioè dei sistemi ICD-MIMD ad Architettura ad Anello con Elemento di Computazione Centrale.



CLASSIFICAZIONE /4,2,1,2,X/ (Livello 5)

1 Categoria /4,2,1,2,1./

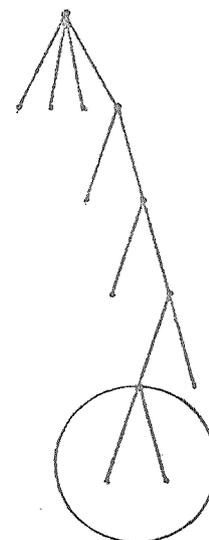
Questa classificazione della Categoria strutturale /4,2,1,2./, dei sistemi ICS-MIND, identifica una sola categoria realizzativa di livello cinque: la Categoria /4,2,1,2,1./, dei sistemi ICS-MIND ad Architettura a Bus con Elemento di Computazione Centrale.



CLASSIFICAZIONE /4,2,2,1,X/ (Livello 5)

2 Categorie /4,2,2,1,X./ X ∈ {1,2}

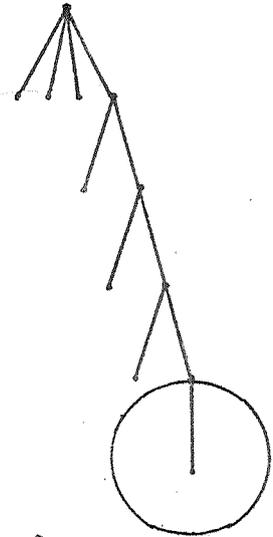
Questa classificazione della Categoria strutturale /4,2,2,1./, dei sistemi IDD-MIND, identifica due categorie realizzative di livello cinque: la Categoria /4,2,2,1,1./, dei sistemi ^{IDD-MIND, cioè dei sistemi} IDD-MIND ad Architettura a Rete Regolare, e la Categoria /4,2,2,1,2./, dei sistemi IDDI-MIND, cioè dei sistemi IDD-MIND ad Architettura a Rete Irregolare.



CLASSIFICAZIONE /4,2,2,2,X/ (Livello 5)

1 Categoria /4,2,2,2,1./

Questa classificazione della Categoria strutturale /4,2,2,2./ dei sistemi IDS-MIND, identifica una sola categoria realizzativa di livello cinque: la Categoria /4,2,2,2,1./ dei sistemi IDS-MIND ad Architettura a Finestra Bus.



CATEGORIA /4,1,1,1,1 ./ (Livello 5)

Sistemi DDL-MIND

Questa categoria contiene i sistemi DDL-MIND, cioè i sistemi DD-MIND ad Architettura ad Anello (Loop Architecture). Lo schema strutturale dell'architettura DDL-MIND è illustrata in Fig. 23.



L'architettura ad anello DDL-MIND, consiste in un certo numero di Unità di Elaborazione individuali (PE), ciascuna delle quali è connessa alle due unità vicine. Il traffico di messaggi in un tale anello potrebbe, in linea di principio, viaggiare in entrambe le direzioni.

In pratica, la complessità del traffico bidirezionale ha fatto sì che praticamente tutti gli anelli realizzati siano a traffico unidirezionale.

In un anello unidirezionale, un vicino di un PE può essere visto come il vicino-sorgente e l'altro come il vicino-destinazione: i messaggi circolano nell'anello dalla sorgente alla destinazione, coi PE

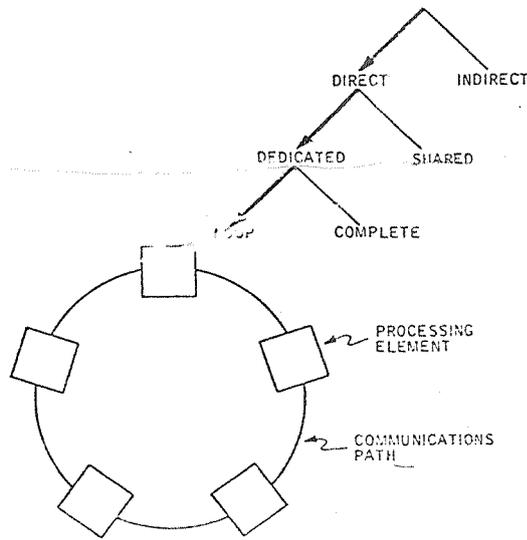


FIGURA 23
Schema strutturale dell'architettura DDL-MIND.

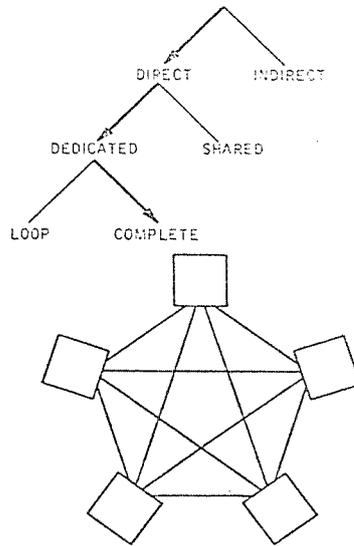


FIGURA 24
Schema strutturale dell'architettura DDC-MIND.

intermedi che agiscono come ritrasmettitori (relay) e unità bruffer.

I sistemi DDI-MMD possono ammettere uno (31), o più messaggi (32), circolare simultaneamente nell'anello, e i messaggi possono essere di lunghezza fissa o variabile (33).

In letteratura si trovano riferiti genericamente come sistemi ad anello, specifici sistemi ad anello che contengono una funzione di commutazione centralizzata, i quali in questa classificazione appartengono alla Categoria /4,2,1,1,2./ dei sistemi ICD-MMD, e sistemi costituiti da anelli accoppiati con controllo decentralizzato, che sono qui classificati nella Categoria /4,2,2,1,2./ dei sistemi IDDI-MMD.

Occorre a questo punto introdurre alcuni criteri per la valutazione delle architetture di queste classi realizzative di sistemi MMD. La modularità di costo (cost-modularity) è essenzialmente la misura del costo incrementale per aggiungere un elemento, quale un processore.

La modularità locale (place-modularity) è la misura del grado in cui si possono considerare localizzate, nella architettura complessiva, la posizione e la funzione di un elemento incrementale.

La flessibilità di connessione (connection flexibility), caratteristica affine alla modularità, misura il grado di complessità in una architettura, risultante dal metodo di connessione, per aggiungere un nuovo elemento, quale un processore, alla struttura.

L'effetto guasto (failure-effect) è la misura dell'effetto di un guasto in una certa architettura; esso serve a dare una misura di quanto il malfunzionamento di un elemento può compromettere la funzionalità globale del sistema. Questo parametro viene anche chiamato tolleranza al malfunzionamento (fault tolerance).

La riconfigurazione dopo il guasto (failure reconfiguration) misura il grado di complessità del metodo per cui un sistema si può ricon-

figurare per nascondere i funzionamenti nei processors o nei cammini di intercomunicazione, per salvaguardare in certa misura la funzionalità globale del sistema.

Una misura delle limitazioni di prestazione, e del costo per superarle, inerenti alla struttura di un sistema, è dato dalla identificazione dei colli di bottiglia (bottlenecks) di una architettura, dovuti o al flusso non uniforme di comunicazione all'interno del sistema, o alla saturazione di certe risorse condivise.

La complessità logica (logical complexity) di un sistema MIMD è la misura del numero di decisioni che devono essere prese per effettuare comunicazioni all'interno della struttura.

Questa si riferisce alla totalità delle decisioni prese durante le comunicazioni, dai processi sorgente e destinazione o dalle entità di computazione.

I criteri sopra esposti saranno presi come riferimento nella valutazione delle architetture dei sistemi MIMD appartenenti alle varie classificazioni del quinto livello.

Per quel che concerne la categoria ora in esame, quella dei sistemi DDI-MIMD, sia la modularità di costo che la modularità locale sono molto buone. Infatti in PE addizionale può essere inserito ovunque nell'anello con l'aggiunta di un singolo cammino di comunicazione, e il flusso di messaggi non è significativamente influenzato dalla sua presenza.

Al contrario, le caratteristiche di tolleranza al malfunzionamento o riconfigurazione al guasto sono povere. Un singolo guasto in un cammino o in una interfaccia di PE causa l'arresto dell'intercomunicazione.

Se è necessaria la riconfigurazione per mascherare il guasto, ci deve essere una struttura di cammini completamente ridondante ed un qualche tipo di commutazione nelle interfacce di PE per escludere

le unità guaste. La riconfigurazione da DDL-MIMD ad un'altra struttura non è una facile operazione poiché i cammini sono unidirezionali e le interfacce relativamente semplici. La complessità logica di comunicazione in un sistema DDL-MIMD è decisamente bassa: un PE deve solo ricevere messaggi, originare messaggi e trasmetterli a una singola destinazione, riconoscere messaggi destinati a se stesso.

L'ampiezza della banda di trasmissione dell'anello è un collo di bottiglia potenziale in relazione all'aumento dei carichi di comunicazione. In aggiunta, alcune discipline di accesso all'anello presentano l'inconveniente che un singolo utente, al limite intenzionalmente, potrebbe saturare l'intera ampiezza di banda disponibile.

Le architetture DDL-MIMD implementate usano quasi tutte collegamenti di tipo seriale a bit come cammini di comunicazione tra i PE.

Così, insieme col ritardo concernente la ritrasmissione dei messaggi, ne risulta un aumento significativo nei tempi di transito dei messaggi attorno all'anello.

In generale questi sistemi sono designati ad applicazioni dove affidabilità e prestazioni non sono esigenze stringenti. L'obiettivo primario di moltissimi progetti di architetture di questo tipo è stato l'interconnessione di sistemi minicomputer geograficamente distribuiti per permettere la condivisione di files di dati e risorse varie.

In questi casi la riconfigurazione dopo un guasto non è in genere eseguita, e neppure il ritardo di un messaggio è un problema stringente.

L'esempio più conosciuto di sistema DDL-MIMD è il "Distributed Computer System" (DCS-Loop) della University of California di Irvine(34). Questo sistema è stato originariamente sviluppato per un preciso interesse nel campo delle comunicazioni di dati e solo più tardi è stato orientato verso il campo dell'elaborazione distribuita.

Il DCS-loop attualmente consiste di cinque minicomputers ed un certo numero di apparecchiature periferiche distribuite nel campus di Irvine e collegate ad anello. L'anello, chiamato "ring", è di tipo seriale a bit ed opera alla velocità di 2.3 Mbs.

Una moltitudine di messaggi di lunghezza variabile può circolare simultaneamente. La tolleranza al malfunzionamento è fornita da un anello ridondante e da commutatori di esclusione dei PE (bypass switches). La già buona modularità locale inerente a queste strutture ad anello è stata incrementata nel Distributed Computer System dalla incorporazione dell'indirizzamento di messaggi di tipo associativo (con una memoria associativa dedicata a questo scopo) ed a livello software.

Invece che inviare un messaggio ad un processor fisico, lo si manda ad un processo logico. La "Ring Interface", corrispondente al processor su cui il processo destinatario risiede correntemente, riconosce l'indirizzo ed accetta i messaggi. Questo permette che la comunicazione sia indipendente dal numero di processors nel sistema, e quindi dagli assegnamenti processo/processor.

Questa idea è stata incorporata anche in altre architetture distribuite.

Un altro esempio di architettura DDL-MIMO è la versione ad anello del sistema data-multiplexing CAMAC (85) che è stata definita per trasferimento sia di tipo seriale a bit che seriale a byte.

CATEGORIA /4, 1, 1, 1, 2./ (Livello 5)

Sistemi DDC-MIND

Questa categoria contiene i sistemi DDC-MIND, cioè i sistemi DD-MIND ad Architettura di Interconnessione Completa (Complete Interconnection Architecture). Lo schema strutturale della architettura DDC-MIND è illustrato in fig.24.



L'architettura DDC-MIND è forse il tipo di disegno concettualmente semplice tra i tipi di strutture di interconnessione MIND. In essa ciascun processor (PE) è connesso da un cammino dedicato con altro nel sistema, ed i messaggi tra processors sono trasferiti solo attraverso il cammino che li connette.

Il PE sorgente deve scegliere il cammino per il PE destinazione tra i cammini alternativi disponibili, e tutti i PE devono essere equipaggiati per accogliere messaggi provenienti da una molteplicità di cammini.

La più significativa caratteristica dell'architettura DDC-MIND è la sua povera modularità di costo.

L'aggiunta dell' n -esimo PE richiede non solo l'aggiunta di $n-1$ cammini tra di esso e gli altri PE, ma anche che tutti i PE del sistema debbano avere le opportune facilità di accettare il PE incrementale come sorgente di dati. Così, le loro interfacce devono avere al minimo $n-1$ porti, dove n è la dimensione massima del sistema (numero massimo di PE).

Alternativamente deve essere possibile per tutti i PE accettare connessioni esterne quando il numero dei PE deve essere incrementato al di là del numero dei porti disponibili. La modularità locale dei sistemi DDC-MIND è buona, come pure le caratteristiche di effetto-quasto e di riconfigurazione al guasto.

L'architettura DDC-MIND è tale che può essere facilmente degradata, nel caso di un guasto, senza il bisogno di cambiare la classe di interconnessione (il processor guasto, o uno dei due processors agli estremi del cammino guasto, può essere semplicemente disconnesso dal sistema).

In aggiunta, la riconfigurazione ad una struttura I-MIND potrebbe essere effettuata nel caso di malfunzionamento di un cammino se il costo del software relativo e l'aumento del tempo di transito dei messaggi fossero accettabili.

I sistemi DDC-MIND non hanno colli di bottiglia strutturali e la loro complessità logica è relativamente bassa. Tuttavia va notato che questa architettura obbliga ad una politica di indirizzamento a locazione (e non logico) nella comunicazione tra processi, poichè la commutazione all'interno di un processo e le eventuali attività di trasmissione porrebbero la struttura risultante nelle categorie dei sistemi I-MIND e IIR-MIND.

I sistemi DDC-MIND possono essere geograficamente localizzati o distribuiti, sebbene vi sono pochi esempi di entrambi i casi. L'esempio più conosciuto di architettura DDC-MIND locale è la versione pienamente connessa dell'IBM Attached Support Processor System (86) in cui fino a quattro sistemi /360 o /370 possono essere collegati attraverso accoppiatori di canali di I/O.

Virtualmente tutti gli esempi esistenti di sistemi DDC-MIND geograficamente dispersi sono di piccole dimensioni, cioè con al massimo tre PE, e sembrano essere interconnessioni ad hoc di installazioni di computers precedentemente esistenti, come esemplificato dalla configurazione pienamente connessa del sistema IERIT (87).

Esso consiste di due IBM 360/57 e un CDC 6500, situati in diversi campus dell'Università del Michigan e connessi da linee di rete pubblica.

Il disegno del sistema Merit ha la potenzialità di configurazioni inferiori alla piena interconnessione, come la IDDI-MIND, per ridurre i costi delle comunicazioni.

CATEGORIA /4.1.1.2.2./ (Livello 5)

Sistemi DSB-MIMD



Questa categoria contiene i sistemi DSB-MIMD ad Architettura a Bus globale (Global Bus Architecture).

Lo schema strutturale dell'architettura DSB-MIMD è illustrato in fig.25.

L'architettura DSB-MIMD è costituita da un certo numero di PE interconnessi da un bus comune, o globale. L'accesso a questo bus è condiviso tra i processori in base ad un certo schema di allocazione, ed i messaggi sono inviati direttamente dal PE sorgente sul bus, per essere riconosciuti ed accettati dalla destinazione (o da più destinazioni) appropriata.

Sia la modularità di costo che locale dei sistemi DSB-MIMD sono buone rispetto ai PE. Dipendentemente dalla scelta dello schema di allocazione del bus, può essere possibile aggiungere un PE al sistema in ogni posizione con un piccolo effetto, o nessuno, sugli altri PE.

Tuttavia, la modularità di costo e locale del cammino di comunicazione è povera. Non è possibile aumentare facilmente l'ampiezza della banda di trasmissione come occorrerebbe, neppure è possibile spesso aumentare le prestazioni solamente dove occorrerebbe.

Per accrescere le prestazioni è usualmente necessario cambiare l'implementazione dell'intero bus o replicarlo, entrambe alternative che hanno un significativo impatto di disegno sulle interfacce sul bus di tutti i PE del sistema.

Stilmente, le caratteristiche di effetto-custo e riconfigurazione al costo dei sistemi DSB-MIMD sono molto buone rispetto ai

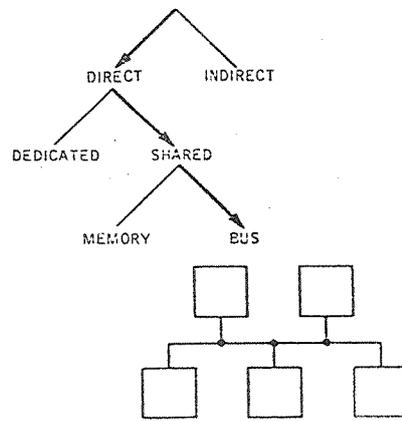


FIGURA 25
Schema strutturale dell'architettura DSB-MIMD.

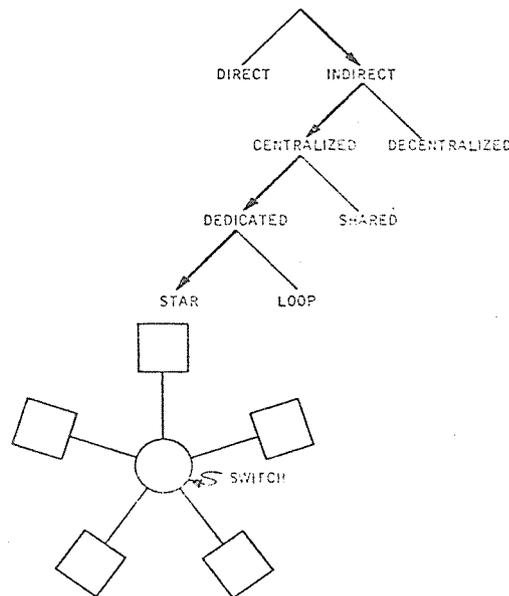


FIGURA 26
Schema strutturale dell'architettura ICDS-MIMD.

processors, ma novere rispetto al bus. Per i malfunzionamenti di PE, questa architettura non richiede nessuna speciale attività hardware di riconfigurazione per continuare le operazioni come sistema DSB-MIND.

Supposta una ragionevole cura nel disegno delle interfacce bus, i malfunzionamenti di processors hanno un piccolo effetto sulle operazioni del sistema. Guasti sul bus, tuttavia, sono inevitabilmente catastrofici e la replicazione è richiesta se l'architettura DSB-MIND deve essere conservata dopo la riconfigurazione.

Il bus globale è naturalmente, nell'ampiezza di banda, un potenziale collo di bottiglia del sistema complessivo.

Volto dell'interesse corrente nella categoria dei sistemi DSB-MIND è occorso nell'ambito aerospaziale, dove sono usati cammini sia seriali che paralleli.

In queste applicazioni, le caratteristiche di modularità locale e di costo di questi sistemi sono particolarmente vantaggiose, e permettono una grande flessibilità nel configurare sistemi per specifiche applicazioni.

Anche le caratteristiche di riconfigurazione al gusto dei sistemi DSB-MIND sono utili per queste applicazioni poiché moltissime applicazioni aerospaziali sono, per loro natura, in tempo reale e, quindi, eccessivi ritardi di riconfigurazione devono essere evitati.

L'uso di bus seriale per il cammino di comunicazione sembra costituire l'approccio più popolare, in primo luogo perché le applicazioni dei sistemi tendono ad avere alti costi associati ai collegamenti fisici tramite cavi, necessità questa tanto più impellente per la distribuzione fisica su distanze notevoli, dal punto di vista elettrico, in secondo luogo perché sono richieste velocità di trasmissione di dati piuttosto basse.

La replicazione del cammino di comunicazione è, in questo ambito, la tecnica predominante per mitigare sia le restrizioni di ampiezza di banda, sia la vulnerabilità ai guasti del bus condiviso.

Molte architetture per applicazioni aerospaziali di tipo DSB-MIMD, sono in effetti degli ibridi rispetto questa classificazione (33).

Tuttavia i servizi militari stanno cominciando a stabilire standards per il multiplexing di dati (34) che richiedono l'uso di approcci di tipo DSB-MIMD nei futuri sistemi aerospaziali.

Una tipica architettura DSB-MIMD per uso della US. Navy, che utilizza bus di ampiezza-parola, è presentata da Jensen (35).

L'approccio DSB-MIMD è pure una popolare disciplina nella condizionale di linea nelle applicazioni di automazione industriale e di laboratorio (36), come è anche esemplificato dal sistema CANAC (37).

CATEGORIA /4,2,1,1,1./ (Livello 5)

Sistemi ICDS-MIMD



Questa categoria contiene i sistemi ICDS-MIMD, cioè i sistemi ICDS-MIMD ad Architettura a Stella (Star Architecture).

Lo schema strutturale dell'architettura ICDS-MIMD è illustrata in fig.26.

L'architettura ICDS-MIMD consiste in una risorsa centrale di computazione alla quale sono connessi un certo numero di processori (PE), ciascuno tramite un cammino bidirezionale, funzionalmente unico.

I messaggi sono scambiati tra i PE usando il commutatore centrale come intermediario; essa è l'apparente destinazione o l'apparente sorgente di tutti i messaggi.

La funzione della risorsa di computazione è vista usualmente come quella di "isolare" i processi operanti su un dato PE dalla conoscenza fisica del sistema, e di proteggerli l'uno dall'altro.

La architettura ICDS-MIMD ha molte caratteristiche in comune con le architetture DS-MIMD poiché entrambe hanno facilità di trasferimento messaggi di tipo condiviso.

La modularità di costo e locale, in questa categoria, è buona rispetto ai PE e povera rispetto alle risorse centrale. Similmente, le caratteristiche di effetto guasto e di riconfigurazione al guasto sono buone per i PE e povere rispetto al commutatore.

L'insorgenza di un collo di bottiglia nel commutatore è un problema potenziale. La flessibilità di connessione dei sistemi ICDS-MIMD è povera poiché il PE incrementale deve sempre essere

provvisto di un cammino individuale al commutatore centrale. La complessità logica di questa architettura è moderata. La risorsa di commutazione deve possedere, al suo interno, informazioni sufficienti (ad esempio, tabelle di instradamento) per consentire che le comunicazioni abbiano luogo, ma il fatto che vi sia una sola copia di queste informazioni, usualmente dinamiche, semplifica il loro agganciamento durante la riconfigurazione.

La novità delle caratteristiche di effetto guasto e riconfigurazione al guasto della risorsa centrale è comune anche a uno dei PE nel sistema se le informazioni di commutazione sono mantenute al di fuori di tale risorsa centrale; questo è il caso in cui il commutatore usa la traduzione di indirizzi per compiere l'instradamento dei messaggi, e i registri di "mapping" sono sotto il controllo di uno dei PE.

Le architetture ICSS-MFD sono piuttosto comuni. Un esempio è costituito dal sistema Network/340, dell'IBM (93), nel quale nodi utente, costituiti da Sistemi/360 remoti, sono connessi su linee affittate a un controllore centrale 360/31.

CATEGORIA /4,2,1,1,2./ (Livello 5)

Sistemi ICPL-MIND



Questa categoria contiene i sistemi ICPL-MIND, cioè i sistemi ICD-MIND ad Architettura ad Anello con Elemento di Commutazione Centrale (Loop with Central Switch Architecture).

Lo schema strutturale dell'architettura ICPL-MIND è illustrato in fig. 27.

In maniera analoga ai sistemi DD-MIND, le connessioni dirette richieste per le architetture ICD-MIND possono essere implementate o nel modello a stella ICDS-MIND o in un anello, nel caso della struttura ICPL-MIND.

In un tale sistema i messaggi sono messi sull'anello dal mandante, rimossi per un'operazione di traduzione a mappa (mapping) degli indirizzi da un elemento centrale di commutazione, quindi rimessi sull'anello con l'indirizzo effettivo (cioè fisico) per la loro voluta destinazione.

I sistemi ICPL-MIND condividono caratteristiche con le organizzazioni ICDS-MIND e DDL-MIND. Le caratteristiche di risposta ai malfunzionamenti sono quelle dei sistemi DDL-MIND, rispetto ai cammini di comunicazione dati, e quelle dei sistemi ICDS-MIND, rispetto alla risorsa centrale.

La flessibilità di connessione è superiore rispetto la struttura a stella ICDS-MIND nel fatto che un PE incrementale deve essere collegato solo ai suoi due vicini a livello fisico, e non ad un elemento di controllo centrale possibilmente remoto.

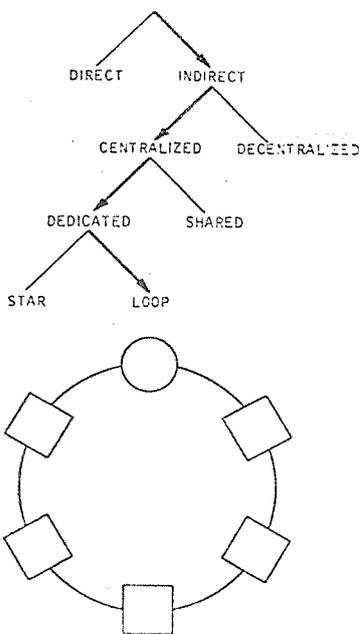


FIGURA 27
Schema strutturale dell'architettura ICDL-MIMD.

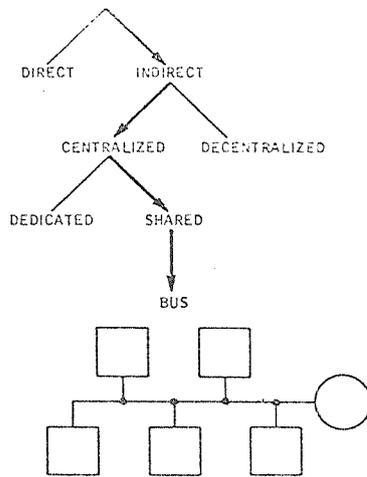


FIGURA 28
Schema strutturale dell'architettura ICS-MIMD.

Similmente alla struttura DDL-MIMD, colli di bottiglia e vulnerabilità nei confronti di utenti maliziosi sono problemi potenziali, ma poiché questi sono rischi inerenti alla commutazione centralizzata delle strutture IC-MIMD, essi non costituiscono necessariamente elementi di debolezza rispetto alla struttura ICDS-MIMD.

La complessità logica dei sistemi ICOL-MIMD è moderata, sebbene sia leggermente superiore rispetto ai sistemi ICDS-MIMD a causa delle richieste di funzioni addizionali che la disciplina ad anello pone alle interfacce dei PT.

Un esempio di architettura ICOL-MIMD è il sistema SPIDER(94), un sistema sperimentale per comunicazioni di dati che interconnette undici computer ai Bell Laboratories a Murray Hill.

In questo sistema una moltitudine di messaggi di dati di lunghezza fissa circola su linee di rete pubblica di capacità di 1.544 MBS.

Il computer centrale dell'anello è costituito da un minicomputer. Ciascun computer nell'anello, eccetto il computer centrale, può essere commutato in stato di sola ricezione in caso di malfunzionamento.

CATEGORIA /4,2,1,2,1./ (Livello 5)

Sistemi ICS-MIND

Questa categoria contiene i sistemi ICS-MIND ad Architettura a Bus con Elemento di Commutazione Centrale (Bus with Central Switch Architecture).

Lo schema strutturale dell'architettura ICS - MIND è illustrato in fig. 22.

L'architettura ICS-MIND è funzionalmente equivalente a quella ICDS-MIND, con la maggiore eccezione che i processors non sono individualmente connessi alla risorsa di commutazione ma condividono un cammino tramite il quale accedono ad essa, così, quando un PE vuole trasmettere un messaggio, deve prima acquisire il bus, quindi trasmettere il messaggio al commutatore.

De esse, il messaggio è ritrasmesso, sullo stesso bus, almeno dal punto di vista funzionale, alla destinazione appropriata. Questa trasmissione è la caratteristica che differenzia i sistemi ICS-MIND dalla organizzazione DSR-MIND e dai sistemi DSI-MIND che usano un singolo bus per la memoria.

Come ci si aspetterebbe, le caratteristiche dell'architettura ICS-MIND sono simili a quelle dei sistemi ICDS-MIND. Dal punto di vista della misura dell'effetto questo, essa è più povera poiché il cammino di accesso all' elemento di commutazione centrale non è, in questo caso, replicato per ciascun PE.

Però l'esistenza di un unico (funzionalmente) cammino condiviso al commutatore non risulta contribuire significativamente a costituire un collo di bottiglia, poiché è del tutto agevole bilanciare le prestazioni con quelle del commutatore.



Così, fin tanto che entrambi si saturano contemporaneamente, oppure il commutatore si satura per primo, il rischio di occorrenze di colli di bottiglia non è superiore rispetto all'organizzazione ICDS-MIND.

La modularità di costo dell'architettura ICS-MIND è influenzata positivamente dal fatto che un processor incrementale occorre sia connesso solo al bus condiviso, e non direttamente al, possibilmente remoto, commutatore.

Come già accennato nella discussione dei sistemi a bus condiviso ma a trasferimento diretto (DSR-MIND), certi disegni di bus rendono questo costo di connessione molto basso.

La scarsa popolarità dei sistemi ICS-MIND può derivare dal fatto che non vi è nessun nome, comunemente usato fino ad ora, che li identifichi. I disegni esistenti, comunque, usano un processor come risorsa di commutazione, ed hanno i PE distribuiti su distanze geograficamente brevi ma elettricamente lunghe.

Sebbene nulla di intrinseco alla architettura escluda l'uso di hardware specializzato, come un crossbar, per il commutatore, implicazioni di velocità sul bus, se si vuole raggiungere il bilanciamento nel senso sopra esposto, probabilmente precludono il suo uso a livello pratico.

Inoltre, come già visto, dato che l'uso di un bus condiviso non è normalmente messo a disposizione dalle reti di comunicazione pubbliche o commerciali, i sistemi ICS-MIND risultano geograficamente ristretti.

Un sistema ICS-MIND è stato disegnato dalla Hughes per la US Navy (25).

La risorsa di commutazione centrale è costituita da un processor "special purpose" chiamato "Network Manager", che esegue funzioni di interfacciamento con un certo numero di nodi funzionali su uno o più bus condivisi.

Oltre a gestire le comunicazioni tra i processors, il Network Manager fornisce vari altri servizi esecutivi ai nodi. Poichè il sistema è fisicamente localizzato, il bus (o i busses) al Network Manager è parallelo a bit.

Il sistema ALOHA (93) è una particolare rete di computers con una struttura ICS-IIIID, che usa principalmente un "bus" radio. Un canale radio a 24K-band di tipo full duplex connette un computer IBM 360/65, che funge da elemento di commutazione centrale del sistema, a terminali remoti, che includono un certo numero di minicomputers.

CATEGORIA /4,2,2,1,1./ (Livello 5)

Sistemi IDDR-MIND

Questa categoria contiene i sistemi IDDR-MIND, cioè i sistemi IDDR-MIND ad Architettura a Rete Regolare (Regular Network Architecture).

Lo schema strutturale dell'architettura IDDR-MIND è illustrato in fig.28.

L'organizzazione IDDR-MIND è caratterizzata dal fatto che i PE sono interconnessi con carmini dedicati, ed hanno identiche relazioni coi vicini PE, eccetto, al limite, sulla frontiera del sistema. Nella struttura esemplificativa di fig.28, ciascun PE ha vicini "a sinistra", "a destra", "sopra" e "sotto", sebbene altre geometrie siano state proposte.

I messaggi sono instradati attraverso la rete dalla sorgente alla destinazione, con ciascun PE intervenente che determina quale dei suoi alternativi vicini dovrebbe essere il successivo PE che riceve il messaggio.

Si può notare che l'architettura DDL-MIND è un caso speciale dell'architettura IDDR-MIND, in cui ciascun PE ha due vicini e non vi è nessuna decisione di commutazione da essere presa.

Le caratteristiche di modularità e di risposta al guasto dei sistemi IDDR-MIND sono significativamente e negativamente influenzate dall'esigenza di assoluta regolarità. Non è possibile aggiungere anche un singolo PE o un singolo cammino; piuttosto la dimensione dell'incremento possibile dipende dal numero di PE del sistema, dal numero di vicini per PE e dal modello di interconnessione.



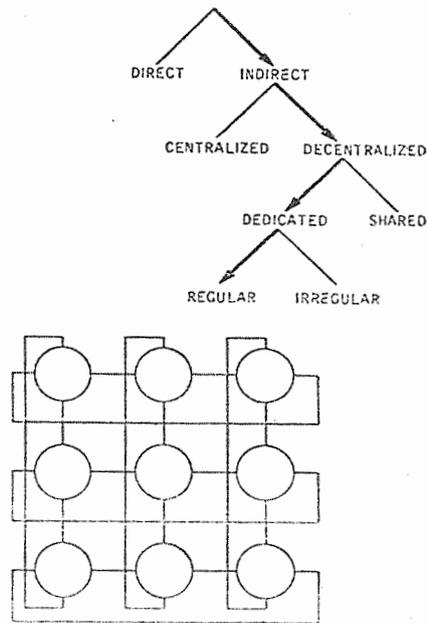


FIGURA 29
Schema strutturale dell'architettura IDDR-MIMD.

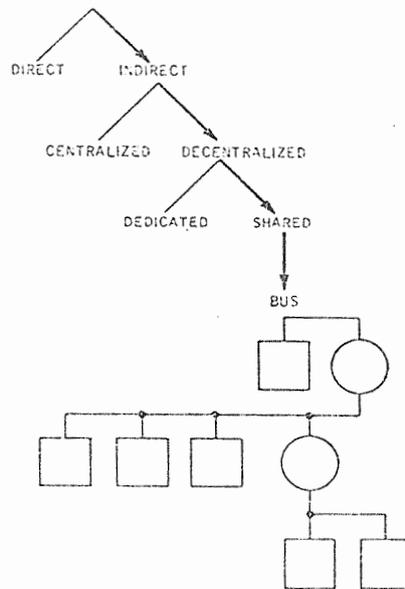


FIGURA 30
Schema strutturale dell'architettura IDS-MIMD.

Così, sia la modularità di costo che locale sono estremamente rovere. Inoltre non c'è nessuna flessibilità di connessione. La complessità logica del disegno è invece moderata: sebbene esistano molti commutatori, la regolarità della connessione semplifica l'instradamento.

La misura dell'effetto-guasto di sistemi IDDR-MIND può essere moderata o buona, dipendentemente dal metodo con cui l'instradamento dei messaggi è eseguito. Un guasto di un singolo PE o cammino non ferma completamente la comunicazione in nessun caso, e neppure è necessario fermare nessun messaggio di cui l'elemento fuori uso non sia interessato (anche in qualità di intermediario).

La riconfigurazione al guasto dei sistemi IDDR-MIND è estremamente rovera, data la richiesta di completa duplicazione di PE e cammini, se la struttura basica regolare debba rimanere inalterata dopo la riconfigurazione.

Invece, se si prevede accettabile la riconfigurazione ad una struttura irregolare (IDDI-MIND), questo inconveniente viene eliminato. L'eleganza delle strutture IDDR-MIND ha provocato un significativo interesse accademico ma i loro inconvenienti pratici hanno impedito un loro successo a livello di implementazioni concrete.

Progetti, a livello di studio e proposta, si sono avuti per sistemi a struttura rettangolare (quattro vicini per ogni PE), con o busse o memorie proposte come cammini.

Un sistema IDDR-MIND non di tipo rettangolare è il TREE (97). Questa macchina comprende un certo numero di PE connessi ad albero.

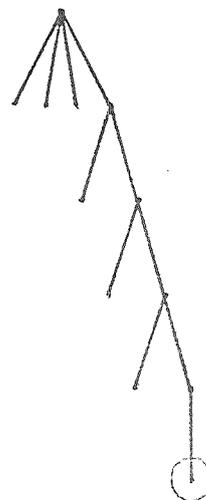
Un PE, può comunicare con i suoi superiori o con ognuno dei suoi subordinati nella gerarchia; poiché vi è solo un cammino tra ogni coppia di nodi in un albero, la complessità logica dell'instradamento dei messaggi non è elevata.

Il sistema TREE è stato concepito primariamente come fisicamente centralizzato, ma potrebbe invece essere implementato con linee di rete pubblica per una più vasta distribuzione.

CATEGORIA /4,2,2,2,1./ (Livello 5)

Sistemi IDS-MIMD

Questa categoria contiene i sistemi IDS-MIMD ad Architettura a Finestra su Bus (Bus Window Architecture). Lo schema strutturale dell'architettura IDS-MIMD è illustrato in fig.30 (esso costituisce solo un esempio di struttura di questo tipo).



Nelle architetture IDS-MIMD l'accesso alle risorse di computazione avviene via un cammino condiviso da più PE. La computazione è eseguita da più di una risorsa, e i messaggi possono essere ritrasmessi sul cammino da cui erano stati ricevuti oppure da un altro.

Le caratteristiche di modularità delle architetture IDS-MIMD sono simili a quelle della struttura IDDI-MIMD. Le caratteristiche di tolleranza al malfunzionamento e di riconfigurazione al guasto sono nuove, poiché più PE e commutatori possono essere isolati dal guasto di un singolo cammino.

Inoltre, sistemi di questo tipo non sono facilmente distribuibili geograficamente a causa dell'uso di busse condivisi (come precedentemente spiegato).

La Digital Equipment Corporation fabbrica un'apparecchiatura chiamata DA11-P Unibus Window per facilitare l'implementazione di architetture IDS-MIMD (98). Un precedente simile è stato disegnato alla Carnegie-Mellon University per il loro Computer Mobile System (99) e alla IBM per il loro OLMPUS MIP (100), che sarà di cui considerare un esempio relativamente a questa classificazione generale.

Tutte queste unità forniscono un cammino bidirezionale tra i buses di due minicomputers. Blocchi o segmenti nello spazio di indirizzabilità sorgente sono tradotti nello spazio di indirizzabilità di destinazione. Un certo numero di queste interfacce può essere usato per costruire gerarchie di elementi di processo.

Una appropriata simulazione ed esperienze di programmazione da parte di utenti di tutti e tre questi meccanismi indicano che la complessità logica di questo approccio aumenta rapidamente col numero dei livelli di traduzione e a mano a mano che il legame di traduzione diventa più dinamico.

Questo tipo di interconnessione è anche soggetto a deadlock, a meno che non si ponga una grande cura nel suo disegno e nel suo uso.

CATEGORIA /4,1,1,2,1./ (Livello 5)

Sistemi DSM-MIMD



Questa categoria contiene i sistemi DSM-MIMD, cioè i sistemi DS-MIMS ad Architettura a Memoria Centrale (Central Memory Architecture).

I sistemi appartenenti a questa categoria sono comunemente denominati come "Sistemi ad Architettura Multiprocessor".

Questa categoria, coi sistemi multiprocessor, e la Categoria /4,2,-2,1,2./ dei sistemi IDDI-MIMD, fra cui sono compresi i sistemi a rete geografica di calcolatori, costituiscono le due più importanti classi realizzative (livello 5) dei sistemi MIMD.

Lo schema strutturale dell'architettura DSM-MIMD è illustrato in fig.31.

Lo schema strutturale a memoria centrale (DSM-MIMD), detto anche multiprocessor, costituisce certamente il modo più comune di interconnettere multiple unità di elaborazione, cioè PE.

In questo tipo di architettura, due o più processors comunicano lasciando messaggi, l'uno per l'altro, in una memoria ad accesso comune.

La caratteristica chiave dell'architettura DSM-MIMD è che la memoria è, o può essere, usata come un cammino piuttosto che semplicemente come unità per la conservazione (cioè memorizzazione) di dati di varia natura.

La modularità locale dei sistemi DSM-MIMD è molto buona: è possibile aggiungere processors arbitrariamente (poichè i processors non sono topologicamente distinti); è pure possibile aumentare la capacità

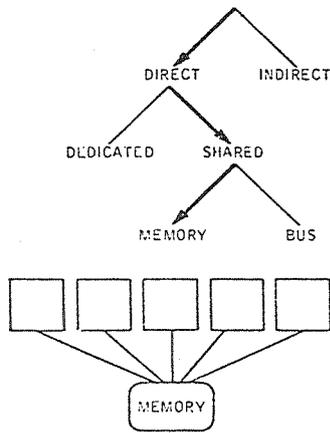


FIGURA 31
Schema strutturale dell'architettura DSM-MIMD.

di transito del cammino semplicemente aumentando la dimensione della memoria comune.

La modularità di costo di questi sistemi dipende quasi esclusivamente dalla struttura del cammino attraverso cui i processors possono accedere al sistema di memoria comune (dei vari tipi di queste strutture se ne parlerà nel seguito). Se ciascun PE è provvisto di un cammino diretto, la modularità di costo può essere povera, poiché un processor incrementale potrebbe richiedere un porto di memoria eventualmente non più disponibile.

Alternativamente, se l'accesso alla memoria avviene attraverso un bus singolo, con un opportuno meccanismo di allocazione, la modularità di costo può essere molto buona.

Un sistema DSM-MIND è del tutto vulnerabile al problema del collo di bottiglia consistente nell'eventualità che l'ampiezza della banda di utilizzo della memoria comune diventi una restrizione sui flussi di comunicazione.

In questo caso la modularità di costo è ancora più povera, poiché è costoso accrescere l'ampiezza di utilizzo della memoria oppure del cammino di accesso.

La complessità logica dei sistemi DSM-MIND è del tutto bassa. Le caratteristiche di tolleranza e di riconfigurazione al guasto, in questi sistemi, sono buone in caso di malfunzionamento di PE, ma povere in caso di malfunzionamento dell'unità di memoria centrale, o dell'eventuale bus condiviso di accesso ad essa.

Vi è inoltre un problema di effetto-guasto di natura software dovuto al fatto che i processors normalmente non hanno restrizioni di accesso alla memoria centrale; in questo modo il software malfun-

zionante o malizioso di un singolo processo può danneggiare e compromettere anche le transizioni di messaggi che non riguardano direttamente tale processo.

Nella realtà si ha che quasi ogni implementazione di un sistema DSM-MIND è avvenuta perchè nel disegno del sistema si voleva che la memoria fosse condivisa, ma risultasse egualmente come luogo di memorizzazione di programmi e dati, mentre l'uso di essa come cammino di comunicazione tra i PE risultava quasi come effetto collaterale.

Nell'implementare questa condivisione, con molteplici scopi, della memoria centrale, si è trovato che le prestazioni dei sistemi realizzati sono cresciute in misura inferiore rispetto alla crescita del numero dei PE e che, in generale, i sistemi consistenti in più di quattro processors, non sono stati effettivi dal punto di vista del costo (11).

La ragione di ciò è stato il livello, estremamente alto, di contesa per l'accesso alla memoria centrale, superando così l'ampiezza massima del suo utilizzo, quando una memoria funzionalmente singola doveva servire per tutti gli scopi.

Viceversa, l'ampiezza della banda richiesta per le sole comunicazioni tra PE è molto difficilmente causa di formazione di un collo di bottiglia, nell'utilizzo di una memoria comune dedicata solo a queste funzioni di intercomunicazione.

Un esempio di questa situazione è costituito dal sistema C.mmp, sistema multiprocessor della Carnegie Mellon (101), che consente ad un massimo di 16 processors di condividere fino a 16 moduli di memoria, attraverso un "crossbar switch", con una trascurabile interferenza reciproca (102).

Procediamo ora ad un esame più approfondito dei sistemi multiprocessor, cioè della categoria DSH-MIMD, e ad una ulteriore articolazione di questa categoria (3).

Nella delineaazione funzionale di un sistema multiprocessor, occorre mettere in risalto il controllo integrato sotto cui opera l'insieme dei PE del sistema, il che significa che un multiprocessor deve avere un singolo sistema operativo integrato; inoltre vanno sottolineati i concetti di condivisione delle risorse globali del sistema, e di interazione dei vari PE nella struttura complessiva del sistema, sui quali si basa la funzionalità di un tale sistema.

Dal punto di vista hardware, il sistema multiprocessor deve avere la capacità di condivisione della memoria principale da parte di tutti i processors e di condivisione delle facilità di Input/Output (Canali, Unità di Controllo, e apparecchiature di I/O) da parte di tutte le combinazioni di memoria e processors (si veda la fig.32).

Un importante aspetto dell'interazione è il livello a cui essa avviene. In questi sistemi tale interazione non si deve limitare allo scambio di files o data sets completi, come in certi sistemi multicomputer (lascamente connessi) in cui l'interscambio avviene a livello I/O, ma deve essere più flessibile.

Infatti nei sistemi multiprocessor deve essere possibile scendere ai più bassi livelli fisici di interazione: dai files e i data sets fino agli elementi di dati; analogamente, dal punto di vista operativo, deve essere possibile l'interazione dal livello di jobs e tasks fino al livello di passi singoli di esecuzione.

Il sistema intero deve essere controllato da un solo sistema operativo che provveda le facilità di interazione dei PE e di condivi-

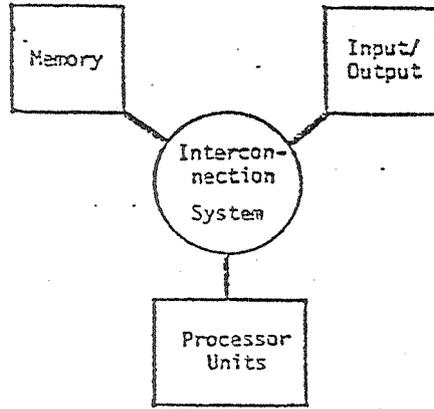


FIGURA 32

Schema funzionale, dal punto di vista hardware, di un sistema multiprocessor.

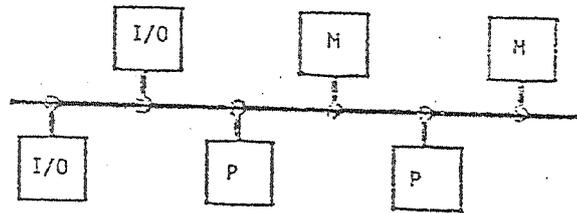


FIGURA 33

Schema strutturale di un sistema multiprocessor a bus condiviso temporalmente, con organizzazione a singolo bus.

sione delle risorse. In un ambito multiprocessor l'efficienza di un sistema operativo diventa anche più importante che nei sistemi convenzionali: scarse prestazioni di un sistema operativo per un multiprocessor ^{possono} distruggere i vantaggi che un tale sistema può presentare, da un punto di vista di rapporto costo prestazioni, rispetto un sistema uniprocessor.

Vi sono essenzialmente tre organizzazioni che sono state utilizzate nel disegno di sistemi operativi per sistemi multiprocessors. Esse sono : schema padrone - schiavo (master - slave), schema a sezione esecutiva per ciascun processor (separate executive for each processor); schema a trattamento simmetrico o anonimo per tutti i processors (symmetric or anonymous treatment of all processors).

È molto importante notare che questi concetti sui sistemi operativi per multiprocessor (sistemi DSM-MMD, Categoria /4, 1, 1, 2, 4./) si possono in genere estendere per ipotesi di sistemi operativi globali di sistemi MMD (Categoria /4./); per questo motivo le principali organizzazioni di tali sistemi operativi vengono spiegate nel seguito.

Sistema Operativo a "Schema padrone-schiavo"

Questo schema di sistema operativo è certamente il tipo più semplice da implementare e può spesso essere prodotto apportando estensioni relativamente semplici ad un sistema operativo per uniprocessor che includa piene capacità di multiprogrammazione.

Questo tipo di sistema operativo presenta sicuramente il vantaggio della semplicità, ma è usualmente del tutto inefficiente nel suo controllo ed utilizzazione delle risorse totali del sistema.

Tuttora non è chiaro quale delle altre due organizzazioni di sistema operativo (per sistema multiprocessor o comunque distribuito) è la migliore dal punto di vista delle prestazioni; comunque appare evidente che entrambe sono decisamente superiori allo schema master-slave.

Elenchiamo ora le caratteristiche principali di un sistema operativo operante in modo padrone-schiavo.

La procedura esecutiva (nel senso di "executive") viene sempre eseguita dallo stesso processor (padrone). Se un altro processor (schiavo) ha bisogno di un servizio che deve essere fornito dal modulo esecutivo del sistema operativo, deve fare la richiesta di tale servizio, e quindi aspettare finché l'attività corrente sul processor padrone sia interrotta e la procedura esecutiva sia attivata.

Le procedure del modulo esecutivo non è necessario siano rientranti, in questo caso, poiché vi è un solo processor (il padrone) che le usa.

Un vantaggio consiste nel fatto che, avendo un singolo processor che esegue la procedura esecutiva, non si hanno conflitti nell'accesso delle tabelle di controllo del sistema operativo, e quindi neppure problemi di "lock-out" (garanzia di accesso esclusivo) e di "dead-lock"

(attesa indefinita) conseguenti, per accessi a sezioni critiche di tali tabelle.

Viceversa altri svantaggi sono costituiti dalla scarsissima flessibilità del sistema complessivo: l'intero sistema è soggetto ad un blocco totale di funzionamento, che richiede l'intervento esterno dell'operatore ed il "restart" (riavviamento) del sistema, quando il processor designato come padrone è soggetto a malfunzionamento o ad un errore non recuperabile.

Inoltre il tempo di inattività nei processors schiavi può crescere considerevolmente e diventare piuttosto significativo se il processor padrone non è in grado di eseguire le procedure di "dispatching" (schedulazione di attività) abbastanza velocemente da mantenere occupati i processors schiavi.

A vantaggio di questo tipo di sistema operativo è la sua esigenza di strumenti software ed hardware relativamente semplici; inoltre esso appare più effettivo per speciali applicazioni in cui il carico di lavoro è ben definito, oppure per sistemi asimmetrici in cui i processors schiavi hanno capacità inferiori rispetto al processor padrone.

Sistema Operativo a "sezione esecutiva per ciascun processor"

Questo schema di sistema operativo prevede che sia operante in ciascun processor una copia separata della sezione delle procedure di sistema operativo chiamata esecutiva.

È quindi necessario che la parte del codice del sistema operativo relativo a questa attività di supervisione ("supervisory"), detta sezione esecutiva, sia di natura ricorrente oppure sia replicata per ogni processor, in entrambi i casi garantendo così una copia separata, o condivisa,

di
Queste procedure, per ogni processor.

Dal punto di vista funzionale ne risulta che, in questa organizzazione di sistema, ciascun processor è preposto a servire le proprie esigenze. Ciascun processor, mediante il proprio modulo esecutivo, ha il suo insieme privato di tabelle, o di altre opportune strutture dati, di gestione del sistema, sebbene alcune di esse dovranno essere in comune all'intero sistema, cioè ad uso condiviso da tutti i processors (tramite i loro moduli "executive") del sistema.

In questo caso, a differenza dello schema padrone-schiavo, sorgono dei problemi di regolamentazione all'accesso a queste tabelle condivise; si tratterà di gestirle come regioni critiche, quindi ad accesso controllato tramite tecniche di "lock-out" (accesso esclusivo), garantendosi però da eventuali problemi di "dead-lock" (attesa indefinita).

Inoltre ciascun processor, in questo schema gestionale, possiede un proprio insieme di apparecchiature di I/O, attraverso la propria sezione esecutiva del sistema operativo; e quindi esso solo potrà avere accesso diretto ai files di dati presenti su tali apparecchiature di I/O. Quindi, uno svantaggio di questa organizzazione consiste nel fatto che il malfunzionamento di un processor, sebbene non pregiudichi totalmente la funzionalità del sistema complessivo (come per l'organizzazione padrone-schiavo), compromette l'accesso a tutte le apparecchiature di I/O gestite da quel processor, e quindi ai files di dati da esse contenuti (a meno di interventi di operatore e di commutazione manuale dei collegamenti di tali apparecchiature di I/O).

Sistema Operativo a "Trattamento Simmetrico, od anonimo, di tutti i processors"

Questa organizzazione di sistema operativo consiste nel trattare tutti i processors come tutte le altre risorse, cioè simmetricamente, vale a dire come un insieme anonimo di risorse.

Questo schema costituisce certamente il modo più difficile di operare nella gestione del sistema; comunque i vantaggi risultanti possono essere superiori alle difficoltà e complicazioni che possono sorgere. Infatti questa organizzazione di sistemi operativi appare la più adatta a sfruttare le potenzialità di un sistema di elaborazione a struttura distribuita, sia esso un multiprocessor che un sistema MIMD in generale.

In questa organizzazione si ha che la funzione di processor "padrone", nel senso visto per lo schema padrone-schiavo, vaga da un processor ad un altro; si ha inoltre che diversi processors possono essere in esecuzione, contemporaneamente, di procedure di servizi di supervisione, cioè di procedure esecutive.

Ne consegue che gran parte del codice delle procedure delle funzioni di supervisione del sistema operativo deve essere di natura rientrante (dato che più processors possono eseguire simultaneamente tale codice).

In questo tipo di sistema si può raggiungere il migliore bilanciamento di carico di lavoro su tutti i tipi di risorse, infatti le richieste di servizi possono essere servite, nell'ambito del sistema, ove esistono risorse adatte più disponibili.

Un pericolo si ha che conflitti nelle richieste di servizi possono essere risolti attraverso l'uso di opportune discipline a priorità, le quali possono essere definite staticamente oppure possono essere sotto

controllo gerarchico.

L'inconveniente maggiore è rappresentato dai conflitti di accesso a tabelle di gestione (o comunque strutture dati) del sistema operativo globale, e dai possibili ritardi derivanti dal "lock-out" (accesso esclusivo) a queste tabelle; tuttavia non c'è nessun modo di evitare questo tipo di problema in esecuzioni di natura multipla concorrente.

E' opportuno sottolineare che, ^{dato che} in un'organizzazione di questo tipo la gran parte delle strutture dati per la gestione, a livello software, del sistema debbono essere condivise dai processors che ne fanno richiesta (eseguendo procedure di supervisione), l'accesso a tali strutture dati deve essere correttamente ed efficientemente controllato per mantenere l'integrità del sistema operativo e le sue prestazioni, evitando ad esempio situazioni di "dead-lock".

I potenziali vantaggi che possono derivare dall'operare in questo tipo di sistema operativo si possono riassumere nei seguenti punti: questa organizzazione può fornire l'uso più efficiente delle risorse disponibili nel sistema; inoltre può consentire una degradazione parziale progressiva e controllata ("graceful degradation") delle funzioni globali del sistema in caso di guasti di alcune sue componenti, provvedendo, allo stesso tempo, una migliore disponibilità del sistema a capacità ridotta.

Alla fine della presentazione di questi tre tipi di organizzazione di sistemi operativi, occorre sottolineare che molti sistemi operativi realmente implementati per sistemi multiprocessor o per sistemi MIMD in generale (nel caso essi posseggano un effettivo sistema operativo globale al sistema complessivo) non sono "puri" esempi di nessuno di questi tre tipi di organizzazione, ma presentano caratteristiche diverse, specie che di, possibilmente, più di uno di questi tipi.

CATEGORIA /4,2,2,1,2./ - Livello 5
Sistemi IDDI-MIMD

Questa categoria contiene i sistemi IDDI-MIMD, cioè i sistemi IDD-MIMD ad Architettura a Rete Irregolare (Irregular Network Architecture).

Una grandissima parte dei sistemi comunemente chiamati Rete di Calcolatori Geograficamente Distribuita (Geographically Distributed Computer Networks) fanno parte di questa categoria, per cui essa, assieme alla Categoria /4,1,1,2,1./ dei sistemi Multiprocessor, costituisce una delle più importanti classi realizzative (livello 5) dei sistemi MIMD.

Lo schema strutturale dell'architettura IDDI-MIMD è illustrato in Fig. 40 (esso costituisce solo un esempio di struttura di questo tipo).

La differenza fondamentale tra l'architettura di tipo IDDR-MIMD e IDDI-MIMD consiste nel fatto che in questo secondo caso non sono richieste relazioni di connessione regolari tra i PE vicini. Così, un dato PE può avere da uno fino ad un numero arbitrario di vicini con i quali comunica direttamente.

Molte delle caratteristiche del sistema variano col grado di regolarità e con la topologia di interconnessione. La modularità globale dei sistemi IDDI-MIMD tende ad essere estremamente buona, poichè sia processors che cammini possono essere aggiunti come e dove occorrono.

La modularità di costo è similmente buona, poichè ai PE incrementali si richiede architetturalmente di avere anche solo una (o due; negli approcci orientati ad anello) connessione col resto del sistema.

La flessibilità di connessione è un altro vantaggio, con connessioni



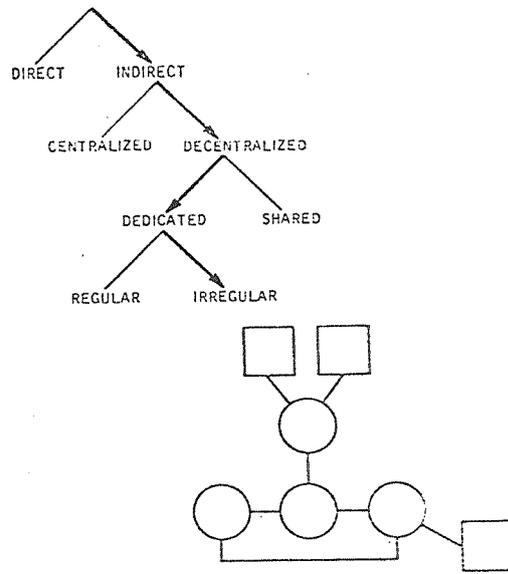


FIGURA 40
Schema strutturale dell'architettura IDDI-MIMD.

ammesse con ogni PE nel sistema. Più è irregolare il modello di interconnessione, più la tolleranza al malfunzionamento e la capacità di riconfigurazione al guasto possono essere prese notevoli dislocando molteplici cammini potenziali tra i PE. Analogamente, l'irregolarità aumenta l'estensione possibile della riconfigurazione che può essere eseguita senza abbandonare la classe strutturale IDDI-MIMD.

La complessità logica di questi sistemi è tuttavia molto alta: presso ciascun commutatore le decisioni di instradamento dei messaggi ("routing") devono essere basate sulla conoscenza della topologia complessiva del sistema (anche se possibilmente parziale).

A causa della buona modularità locale di queste architetture, il problema dei colli di bottiglia non è probabile e può essere eventualmente sormontato con facilità.

L'applicazione corrente dominante della struttura di interconnessione IDDI-MIMD consiste nelle reti di calcolatori geograficamente distribuite. In tali sistemi i cammini sono forniti da reti di comunicazione pubbliche o commerciali e le attività di commutazione sono eseguite da processors dedicati a questa funzione.

Gli algoritmi di instradamento dei messaggi, ai vari punti di commutazione, sono usualmente poco eleganti, specialmente per i sistemi meno regolari, e sembrano più che altro soluzioni ad hoc del problema.

Data noi la complessità delle decisioni che devono essere prese dagli elementi di commutazione in questo tipo di struttura, si rende necessario l'uso di processors di potenze adeguate per queste funzioni; il loro costo risulta quindi rilevante nell'ambito dell'architettura di interconnessione del sistema, anche se, come accennato, la loro relativa potenza può essere sfruttata per una più efficiente definizione e gestione delle linee di trasmissione dati (elementi molto costosi) costituenti i cammini di trasferimento messaggi (ad esempio riducendo all'essenzia-

Le le connessioni topologiche nella rete e sfruttando al massimo la capacità delle linee con tecniche di multiplexing statistico, cioè di concentrazione).

Quindi il costo della complessità logica dei sistemi rete di calcolatori, rappresentato dai processors di commutazione, risulta piuttosto notevole.

Un tipo di sistema IDDI-MIMD con un certo grado di regolarità è rappresentato dagli anelli accoppiati di Pierce (103), sistema concepito per le telecomunicazioni. In esso un certo numero di strutture locali ad anello, di tipo DDL-MIMD, controllate indipendentemente, sono accoppiate assieme direttamente oppure da "tronchi ad anello" (a loro volta di struttura tipo DDL-MIMD). Inoltre è proposto un certo numero di anelli alternati, ridondanti oppure commutati per l'esclusione, con lo scopo di fornire protezione al guasto.

Tuttavia i sistemi caratterizzanti la Categoria /4,2,2,1,2./ delle architetture IDDI-MIMD sono, come già anticipato, i sistemi a rete di elaboratori geograficamente distribuita.

Quindi un ulteriore e più approfondito esame di questa categoria, ed una sua successiva articolazione, saranno fatti nell'ottica delle reti di calcolatori ("Computer Networks") (9).

Una rete di comunicazione di calcolatori si definisce consistente di questi elementi (9):

- 1) un insieme di ospiti (hosts), cioè computers che forniscono servizi all'utente, che può essere un utente finale oppure un altro computer ospite;
- 2) una sottorete che fornisce le comunicazioni tra i computers ospiti, gli utenti, oppure entrambi.

BIBLIOGRAFIA

Bibliografia per la Parte I e per aspetti generali di classificazione
nella Parte II.

- (1) U. Pellegrini: "Evoluzione dei sistemi di Informatica negli anni '70:
dai sistemi centralizzati ai sistemi distribuiti"
Automatica e Strumentazione - Ott. 1975
- (2) D. Siewiorek: "The multiple processor design space"
AFIPS - Conference Proceedings, Vol. 46 - 1977
- (3) S. P. Dennis, J. G. Smith: "LSI implications for future design and
architecture"
SJCS Proc., Vol. 33 - 1970
- (4) S. S. Reddi, E. A. Faustel: "A conceptual framework for computer ar=
chitecture"
ACM - Computing Surveys - Giugno 1973
- (5) R. Levin, J. McQuillan, R. Shantz: "Distributed Systems"
ACM - Operating Systems Review - Gen. 1976
- (6) P. W. Enslow: "Multiprocessor architecture: a survey"
Sagamore Computer Conference on Parallel Processing - 1975
- (7) G. P. Casaglia, W. Lijtmaer: "Strutturazione uniforme di sistemi di
calcolo distribuiti realizzati tramite micro e mini calcolatori"
Tecnologie dei sistemi di informatica Olivetti, Vol. 8 - 1973
- (8) S. C. Yau, H. G. Funz: "Associative processor Architecture: a survey"
ACM - Computing Surveys - Marzo 1977

- (9) S. R. Kimbleton, G. W. Schneider: "Computer communication networks: approaches, objectives and performance considerations"
ACM - Computing Surveys - Sett. 1975
- (10) K. J. Thurber, L. D. Wald: "Associative and parallel processors"
ACM - Computing Surveys - Dic. 1975
- (11) G. A. Anderson, E. D. Jensen: "Computer interconnection structure: taxonomy, characteristics and examples"
ACM - Computing Surveys - Dic. 1975
- (12) L. G. Hobbs, D. J. Theis: "Survey of parallel processor approaches and techniques"
"Parallel processor systems, technologies and applications"
Ediz. Spartan Books - 1970
- (13) E. D. Jensen: "The influence of microprocessors on computer architecture: distributed processing"
ACM Annual Conference - 1975
- (14) U. O. Gagliardi: "Trends in computing-system architecture"
Proceedings of the IEEE, Vol. 63 - Giugno 1975
- (15) D. J. Kuck: "A survey of parallel machine organization and programming"
ACM - Computing Surveys - Marzo 1977
- (16) G. W. Cady, G. Luther: "Trade-off studies in computer network"
Proceedings of the IEEE, Computer conference - 1973
- (17) W. J. Flynn: "More computer organization and their effectiveness"
IEEE Transactions on Computing - Sett. 1972

- (18) J. C. Murtha, R. L. Readles: "Survey of highly parallel information processing systems"
Office of Naval Research, Rep. 4755 - Nov. 1964
- (19) J. E. Shore: "Second thoughts on parallel processing"
Computer and Electrical Engineering, Vol. 1 - 1973
- (20) L. C. Higbie: "Supercomputer architecture"
Computer - Dic. 1973
- (21) G. L. Hollander: "Architecture for large computing systems"
Proceedings AFIPS Spring Joint Computer Conference - 1967
- (22) B. Farhani: "Associative memories and processors: an overview and selected bibliography"
Proceedings of the IFIP - Giugno 1970
- (23) E. J. Thurber: "Large scale architecture: associative and parallel processors"
Ediz. Hayden - 1973
- (24) C. C. Foster: "Computer architecture"
Ediz. Van Nostran - 1970
- (25) G. C. Tessera: "Orientamenti dell'architettura degli elaboratori elettronici"
Quaderni di informatica, n° 3 - 1975
- (26) I. J. Lubal: "Architectural questions of the seventies"
Communication - Marz. 1973

- (27) L. Altman: "Single-chip microprocessors open up a new world of applications"
Electronics - Aprile 1974

- (28) J. C. Nelson: "The economic implication of microprocessor on future computer technology and systems"
Proceedings of National Computer Conference - 1975

- (29) T. C. Chen: "Distributed intelligence for user oriented computing"
Proceedings of Fall Joint Computer Conference - 1972

- (30) R. Rice: "LSI and computer system architecture"
Computer design - Dic. 1970

- (31) F. Turn: "Computers in 1980s"
Columbia University Press - 1974

- (32) P. Enslow: "Multiprocessors and parallel processing"
Ediz. Wiley - 1974

- (33) C. C. Foster: "A view of computer architecture"
Communications of ACM - Luglio 1972

- (34) F. Filippazzi: "Tecnologia dell'elaboratore elettronico: situazione attuale e prospettive"
Ediz. Franco Angeli - 1976

Bibliografia particolare di Sistemi nell'ambito
della Classificazione Generale di Architetture

- (35) T. G. Hallin, M. J. Flynn: "Pipelining of arithmetic functions"
IEEE - Trans. on Computers - Agosto 1972
- (36) E. Joseph, A. Karlan: "Target track correlation with a search memory"
Proc. 6th National MIL-E-CON - Giugno 1972
- (37) M. C. Pease: "An adaptation of the fast Fourier transform for parallel processing"
J. ACM - Aprile 1968
- (38) R. Mc Cormick, J. Divilbiss: "Tentative logical realization of a pattern recognition computer"
Rep. N° 4031, Digital Computer Lab., University of Illinois - 1969
- (39) J. Von Neuman: "A system of 29 states with a general transition rule"
Theory of self-reproducing automata - A. W. Burks Ed., University of Illinois Press - 1966
- (40) J. H. Holland: "A universal computer capable of executing an arbitrary number of sub-programs simultaneously"
Proc. AFIPS Fall Jt. Computer Conf. - 1959
- (41) S. H. Unger: "A computer oriented toward spatial problems"
Proc. IRE - Ott. 1958
- (42) W. Shoeman: "Parallel computing with vertical data"
1960 Eastern Jt. Computer Conf. - 1960

- (43) I. C. Higbie: "The OMEN computers: associative array processors"
COMPCON '72 - 1972

- (44) D. L. Slotnick: "Unconventional systems"
Proc. AFIPS Spring Jt. Computer Conf. - 1967

- (45) D. L. Slotnick et al. "The SOLOMON computer"
Proc. AFIPS Fall Jt. Computetr Conf. - 1962

- (46) G. Barnes, R. Brown, M. Kato, D. Kuck, D. Slotnick, R. Stokes: "The ILLIAC IV computer"
IEEE Trans. Computers - Agosto 1968

- (47) K. J. Thurber, P. C. Patton: "The future of parallel processing"
IEEE Trans. Computers - Dic. 1973

- (48) J. A. Cornell: "Parallel processing of ballistic missile defense radar data with PEPE"
COMPCON '72 - Sett. 1972

- (49) A. A. Hanlon: "Content-addressable and associative memory systems: a survey"
IEEE Trans. Computers - Agosto 1966

- (50) A. Slade, U. Mac Mahon: "A cryoton catalog memory system"
Proc. 1956 Eastern Jt. Computer Conf. - 1956

- (51) R. F. Rosin: "An organization of an associative cryogenic computer"
Proc. AFIPS 1962 Spring Jt. Computer Conf. - 1962

- (52) A. Slade, C. Smallman: "The film cryotron catalog memory"
Automatic Control - Agosto 1960
- (53) C. Yang, S. Yau: "A cutpoint cellular associative memory"
IEEE Trans. Computers - Agosto 1966
- (54) C. Y. Lee: "Intercommunicating cells, basis for a distributed logic computer"
Proc. AFIPS 1962 Fall Jt. Computer Conf. - 1962
- (56) B. Crane, H. Gilmartin, J. Huttenhoff, P. Rux, R. Shively: "PEPE computer architecture"
IEEE COMPCON 1972 - 1972
- (55) B. Crane, R. Laane: "A cryoelectronic distributed logic memory"
Proc. AFIPS 1967 Spring Jt. Computer Conf. - 1967
- (57) A. Evensen, J. Troy: "Introduction to the architecture of a 288-element PEPE"
Proc. 1973 Sagamore Computer Conf. on Parallel Processing - 1973
- (58) K. Batcher: "Flexible parallel processing and STARAN"
1972 WESCON Tech. Papers, Parallel Processing Systems - Sett. 1972
- (59) J. Rudolph: "A production implementation of an associative array processor: STARAN"
Proc. AFIPS 1972 Fall Jt. Computer Conf. - 1972
- (60) K. Batcher: "STARAN parallel processor system hardware"
Proc. AFIPS 1974 National Computer Conf. - 1974

- (61) T. Feng: "Data manipulating functions in parallel processors and their implementations"
IEEE Trans. Computers - Marzo 1974
- (62) R. Linde, R. Gates, T. Peng: "Associative processor applications to real-time data management"
Proc. AFIPS 1973 National Computer Conf. - 1973
- (63) H. Love: "An efficient associative processor using bulk storage"
Proc. 1973 Sagamore Computer Conf. on Parallel Processing - 1973
- (64) G. Courantz, M. Gerhardt, C. Young: "Programmable radar signal processing using RAP"
Proc. Sagamore Computer Conf. on Parallel Processing - 1974
- (65) C. Finnila: "The associative linear array processor"
IEEE Trans. on Computers - Febbraio 1977
- (66) G. Anderson, R. Kain: "A content-addressed memory design for data base applications"
Proc. 1976 Internat. Conf. on Parallel Processing - IEEE - 1976
- (67) W. Crofut, U. Sottile: "Design techniques of a delay-line content-addressed memory"
IEEE Trans. Computers - Agosto 1966
- (68) J. Roles, R. Rux, W. Weingarten: "NEBULA: a digital computer using a 20 mc glass delay line memory"
Comm. ACM - Luglio 1966

- (69) J. Parker: "A logic per track retrieval system"
Proc. IFIP 1971 Congress - 1971
- (70) B. Parahmi: "A highly parallel computing system for information retrieval"
Proc. AFIPS 1972 Fall Jt. Computer Conf. - 1972
- (71) W. Kautz: "An augmented content-addressed memory array for implementation with large scale integration"
J. ACM - Gen. 1971
- (72) W. Schmitz et al. "ABMDA prototype bulk filter development concept definition phase"
Final Report Contract N° DA460-72-C-0050 - National Technical Information Service - Aprile 1972
- (73) R. Seeber, A. Lindquist: "Associative logic for highly parallel systems"
Proc. AFIPS Fall Jt. Computer Conf. - 1968
- (74) W. A. Clark: "Macromodular computer systems"
AFIPS Conf. Proc. 1967 SJCC - 1967
- (75) C. Bell, J. Grason: "The register transfer module design concept"
Computer Design - Maggio 1971
- (76) J. Torode, T. Kehl: "The logic machine: a modular computer design system"
IEEE Trans. on Computers - Nov. 1974
- (77) T. Kehl, C. Hoss, L. Dunkel: "LM² - a logic machine minicomputer"
Computer - Nov. 1975

- (78) R. Miller, J. Cooke: "Configurable computers: a new class of general purpose machines"
Lecture notes in Computer Science, N° 5 - 1974
- (79) J. B. Dennis, D. Misunas: "A preliminary architecture for a basic data-flow processor"
Proc. 2nd Annual Symposium on Computer Architecture - Genn. 1975
- (80) M. Vanneschi: "Models and architectures for data-flow computing systems"
Tecnologie dei Sistemi di Informatica Olivetti -Febbraio 1977
- (81) W. Farmer, E. Newhall: "An experimental distributed switching system to handle bursty computer traffic"
Proc. ACM Symposium on Problems in the Optimization of Data Communications - Ott. 1969
- (82) C. Reames, M. Lin: "A loop network for simultaneous transmission of variable-length messages"
Proc. Symposium on Computer Architecture, IEEE - Genn. 1975
- (83) L. West: "Loop-transmission control structures"
IEEE Trans. on Communications - Giugno 1972
- (84) D. Farber, K. Larson: "The system architecture of the Distributed Computer System - the communications system"
Symposium on distributed computing systems: micros, minis and networks
Pisa - Dic. 1975
- (85) ARC/NEH e ESOM: "CAIAC, serial system organization"
National Tech. Information Service, TID-26483 - Dic. 1973

- (86) IBM: "IBM System/360 and System/370 Attached Support Processor, Version 3, asymmetrical multiprocessor system: general information manual" - GH20-1173

- (87) W. Becher, E. Aubnerle: "The communication computer hardware of the HERIT computer network"
IEEE Trans. on Communications - Giugno 1972

- (88) G. Anderson: "Interconnecting a distributed processor system for avionics"
Proc. Symposium on Computer Architecture, IEEE - Dic. 1973

- (89) US Air Force: "MIL-STD-1553: military standard aircraft internal time division multiplex data bus" - Agosto 1973

- (90) D. Jensen: "A distributed function computer for real-time control"
Proc. Symposium on Computer architecture, IEEE - Genn. 1975

- (91) R. Aronson: "Line-sharing systems for plant monitoring and control"
Control Engineering - Genn. 1971

- (92) L. Costrell: "CAMAC: a modular instrumentation system for data handling; revisited description and specification"
National Tech. Information Service, TID-25875 - Luglio 1972

- (93) D. Mc Kay, D. Karn: "IBM Computer Network/440"
Computers Networks - Proc. Courant Institute Symposium - Nov. 1970

- (94) A. Fraser: "SPIDER - An experimental data communications system"
Proc. Internat. Conf. on Communications, IEEE - Giugno 1974

- (95) J. Rowan, D. Smith, M. Swensen: "Toward the design of a network manager for a distributed computer network"
Proc. Sagamore Conf. on Parallel Processors - Agosto 1974
- (96) N. Abramson, F. Kuo: "Computer-communication networks"
Prentice-Hall - 1973
- (97) R. Goodwin: "A design for distributed-control multiple-processor computer system"
National Tech. Information Service, AD-772 883 - Dic. 1973
- (98) B. Fritzsche: "Standard interfaces promote new minicomputer networks"
Electronics - Sett. 1973
- (99) S. Fuller, D. Siewiorek, R. Swan: "Computer modules: an architecture for large digital modules"
Proc. Symposium on Computer Architecture, IEEE - Dic. 1973
- (100) F. Heart, S. Ornstein, W. Crowther, W. Baker: "A new minicomputer/multi-processor for the ARPA network"
Proc. AFIPS 1973 National Computer Conf. - 1973
- (101) W. Wulf, G. Bell: "C.mmo: a multi-mini processor"
Proc. AFIPS 1972 Fall Jt. Computer Conference - 1972
- (102) W. Wulf, R. Levin, C. Pierson: "Overview of the HYDRA operating system development"
Symposium on distributed computing systems: micros, minis and networks, Pisa - Dic. 1975