UNIVERSITA' DEGLI STUDI DI PISA

FACOLTA' DI SCIENZE MATEMATICHE, FISICHE E NATURALI
CORSO DI LAUREA IN FISICA

TESI DI LAUREA

LE UNITA` DI CONTROLLO DEL SUPERCALCOLATORE APE

CANDIDATO: Fabrizio Coppola

RELATORE: Chiar.mo Prof. Gabriele Torelli

CORRELATORE: Chiar.mo Dott. Raffaele Tripiccione

CONTRORELATORI: Chiar.mo Prof. Giuseppe Pierazzini

ANNO ACCADEMICO 1986/87

Tesine.

Applicazione del metodo di Rabi-Ramsey in misure ultraprecise.

Relatore: Chiar.mo Prof. Franco Strumia.

Produzione di coppie monocromatiche elettrone-positrone in collisioni fra ioni pesanti.

Relatore: Chiar.mo Prof. Luciano Bracci.

Misura della parte reale nell'ampiezza dello scattering elastico protone-protone e antiprotone-protone ad altissime energie.

Relatore: Chiar.mo Prof. Rino Castaldi.

Ad Angela e Vincenzo

Precisazioni successive (2011).

Dopo la presente tesi di laurea (Pisa, 1987) sono seguiti altri articoli accademici pubblicati con lo stesso gruppo APE dell'INFN. I più importanti sono:

P. Bacilieri, M. Bernaschi, S. Cabasino, N. Cabibbo, F. Coppola, et al. (1988).
"Order of the Deconfining Phase Transition in Pure-Gauge QCD".
Physical Review Letters, 61, 1545-1548

P. Bacilieri, S. Cabasino, N. Cabibbo, F. Coppola, G.Fiorentini, et al. (1988).
"The hadronic mass spectrum in quenched lattice QCD: results at beta = 5.7 and 6.0".
Physics Letters B, 214, 115-119.

P. Bacilieri, M. Bernaschi, S. Cabasino, N. Cabibbo; F. Coppola, et al. (1989).
"The hadronic mass spectrum in quenched lattice QCD: results at beta = 5.7".
Nuclear Physics B, 317, 509-525.

P. Bacilieri, M. Bernaschi, S. Cabasino, N. Cabibbo, F. Coppola, et al. (1989)
"The deconfining phase transition in lattice gauge SU(3)".
Nuclear Physics B, 318, 553-578.

P. Bacilieri, M. Bernaschi, S. Cabasino, N. Cabibbo, F. Coppola, et al. (1989)
"On the order of the deconfining phase transition in SU(3) LGT".
Nuclear Physics B - Proceedings Supplements, 9, 315-319.

Ed altri articoli minori nei "Proceedings" di conferenze accademiche (1988-1991)

Indice

Capitolo 0

Introduzione

La tesi è incentrata sulle due unità di controllo del supercalcolatore APE (Array Processor Experiment), alla realizzazione del quale partecipano alcune sezioni dell'Istituto Nazionale di Fisica Nucleare (INFN), tra cui quella di Pisa.

Il calcolatore è stato progettato per compiere calcoli di Cromodinamica Quantistica (QCD) su reticolo. Com'è noto, la Cromodinamica Quantistica è la teoria di gran lunga più accreditata per l'interpretazione delle interazioni forti; essa è una teoria di gauge non abeliana, in accordo con l'idea diffusa fra la maggior parte dei fisici che tutte le interazioni fondamentali si possano spiegare con teorie di questo tipo. La Cromodinamica Quantistica ha dato risultati conformi ai dati sperimentali, ma solo nel limite in cui possono essere impiegati metodi perturbativi. Volendo studiare le caratteristiche degli adroni, l'approssimazione perturbativa non è più valida, e d'altra parte una risoluzione analitica dei calcoli di QCD appare oggi impossibile, data la loro complessità. E` però possibile trattare il problema con metodi numerici, effettuando i calcoli di QCD su di un reticolo spazio-temporale ed avendo a disposizione un'enorme capacità di calcolo.

Data la natura parallela degli algoritmi risolutori, la macchina progettata per questo scopo, APE, è un supercalcolatore parallelo, che viene programmato in un linguaggio simile al fortran - creato appositamente per esso - e che è dedicato ma non specializzato per teorie di gauge su reticolo; può quindi essere impiegato in altri problemi in cui l'algoritmo risolutore sia parallelo e (preferibilmente) preveda aritmetica a numeri complessi. Da un anno sono funzionanti due prototipi ridotti di APE, chiamati "Apetto", che hanno ottenuto interessanti risultati sullo spettro di massa degli stati legati di soli gluoni (glueballs) e recentemente anche degli stati fermionici (adroni). La macchina completa ha iniziato a funzionare in questo periodo.

Il supercalcolatore APE comprende una parte scalare, preposta al controllo, ed una parte vettoriale. Quest'ultima prevede 16 schede FPU (Floating Point Unit) e 16 schede di memoria dinamica. Ciascuna FPU, ottimizzata per compiere l'operazione D = A*B+C fra numeri complessi, ha una capacità di calcolo di 64 Megaflop, e ciascuna memoria ha una capacità di 64 Megabyte. Pertanto le caratteristiche dell'intero calcolatore APE sono 1 Gigaflop di capacità di calcolo ed 1 Gigabyte di memoria, e si trovano quindi in linea con quelle dei maggiori supercalcolatori attualmente esistenti. Il prototipo ridotto, Apetto, comprende 4 FPU e 4 memorie da 16 MegaByte. La generica FPU i-esima può dialogare, attraverso la scheda swichnet, con la [(i+s) MOD 16]-esima scheda di memoria, dove s può essere scelto fra 0 e 15 ed assume il medesimo valore per tutte le 16 memorie.

Le due unità che costituiscono la parte scalare del calcolatore sono un 3081/E - che è un emulatore del calcolatore IBM 3081 realizzato da una collaborazione CERN/SLAC per analisi di dati in esperimenti di fisica di alte energie -, e dalla scheda "sequencer", il cui compito è l'espansione dei comandi del 3081/E in sequenze di

istruzioni per le FPU.

Nel lavoro di tesi mi sono occupato di sviluppi relativi a queste due unità.

La struttura della tesi è la seguente. Dopo che nel cap.1 è stato brevemente esaminato il problema della formulazione della QCD su reticolo, nel cap.2 viene analizzata la funzione dei supercalcolatori in fisica, ed è introdotto il supercalcolatore APE. I problemi di principio relativi alle unità di controllo nei calcolatori paralleli sono brevemente esaminati nel cap.3. Nel cap.4 sono descritti il 3081/E e la scheda sequencer. Parte del lavoro originale della mia tesi è consistito nella progettazione, realizzazione e test della versione definitiva del sequencer, attualmente funzionante a Roma sul prototipo completo di APE. Il sequencer controlla il funzionamento delle FPU, fornendo loro il microcodice ad ogni ciclo di clock, e controllando contemporaneamente la scheda switchnet. L'altro mio contributo originale alla tesi è contenuto nei capp.5 e 6, in cui vengono analizzate le efficienze in APE rispettivamente del 3081/E, e di due suoi possibili sostituti come controllori.

Il 3081/E svolge l'aritmetica a numeri interi, tra cui anche il calcolo degli indirizzi di memoria, e controlla sia le memorie dinamiche che il sequencer, il quale a sua volta controlla le FPU e lo switchnet. L'analisi effettuata su alcuni tipici programmi di QCD ha dimostrato che l'efficienza del 3081/E è molto alta, nonostante che esso non sia stato progettato espressamente per APE (ne è l'unica parte non originale): la perdita di prestazioni rispetto ad un controllore ideale - ovvero infinitamente efficiente - risulta di poco superiore al 10%. Ciononostante, la sua voluminosità (6 schede), la probabile interruzione della sua produzione da parte del CERN, ed altri motivi minori, ne consigliano la sostituzione con un nuovo controllore progettato appositamente per APE su una singola scheda, a condizione che esso offra prestazioni non inferiori al 3081/E.

I progetti che sono stati concepiti a tale scopo sono due. Uno è basato su di un microprocessore evoluto, il cui funzionamento asincrono richiede un'interfaccia elastica col resto di APE. L'altro è basato su due componenti sincroni complementari e prevede l'eliminazione del sequencer, o più propriamente l'assorbimento di questo nell'unico controllore, il che eliminerebbe la necessità di due codici separati in APE, sia pure sincronizzati. La simulazione di alcuni programmi di QCD sui due possibili nuovi controllori hanno dimostrato che essi, nonostante la diversità, sono pressocchè equivalenti riguardo alle prestazioni, le quali risultano leggermente migliori rispetto a quelle del 3081/E: la perdita di prestazioni rispetto al controllore ideale, valutata con criteri pessimistici, risulta pari all'8% circa.

Prescindendo dal primo capitolo, la prima parte della tesi - costituita dai capp. 2 e 3 e dai primi tre paragrafi del cap.4 - consiste in un'introduzione all'argomento dei supercalcolatori paralleli in generale, e ad APE in particolare, mentre la seconda parte della tesi è di stile più tecnico, in quanto descrive in dettaglio il lavoro da me svolto.

Desidero a questo punto ringraziare le persone che, oltre ai relatori, mi hanno aiutato nella realizzazione della presente tesi: Adriano Lai, che ha collaborato al mio lavoro; Giovanni Fiorentini, Luigi Fonti, Maria Paola Lombardo, Gaetano Salina ed Emilio Simeone, che mi hanno aiutato in diversi modi; e tutti i laureandi presso l'INFN di Pisa.

Capitolo 1

Cromodinamica Quantistica su reticolo

§ 1.0 - Introduzione.

In un solo secolo la ricerca dei costituenti elementari della materia ha compiuto dei progressi enormi. Nel secolo scorso si affermò la teoria atomica e fu accettata la tavola periodica degli elementi proposta da Mendeleev. Nel giro di qualche decennio si scoprì che i 92 elementi chimici componenti la tavola di Mendeleev potevano essere spiegati in termini di poche particelle elementari: il protone, l'elettrone ed il neutrone.

Gli esperimenti compiuti a partire dagli anni '30, compiuti essenzialmente al fine di studiare la forza che teneva uniti protoni e neutroni nel nucleo atomico, diedero però come inaspettato risultato la scoperta di decine di nuove particelle complicando così il quadro semplice che si era venuto a creare negli anni '30. Le particelle, come sappiamo, furono classificate in adroni ed in leptoni, a seconda che interagissero o meno per interazione forte.

Nel 1963, M.Gell-Mann propose una teoria che poteva rendere di nuovo semplice il quadro delle particelle elementari: basandosi su delle evidenti simmetrie

presentate dagli adroni, Gell-Mann propose che questi fossero composti di particelle più elementari, dette quark; soli 3 tipi (o "sapori") di quark potevano spiegare l'esistenza di tutte le particelle ad interazione forte allora conosciute; i barioni, per esempio, erano composti di 3 quark con simmetria SU(6) contemporanea rispetto al sapore ed allo spin (6=2*3, dove 2 è la molteplicità di spin e 3 è il numero dei sapori di quark).

Le difficoltà di tale teoria erano molte. Tra queste vi era il fatto che non si era mai osservato un quark libero (e non lo è stato osservato a tutt'oggi), ed il fatto che alcune particelle - come la Ω^- - prevedevano di essere composte da 3 quark identici nello stesso stato, contraddicendo così il principio di Pauli. Per ovviare a ciò, fu introdotto un nuovo numero quantico, detto "colore", che poteva assumere 3 valori (p.es rosso, verde e blù); i barioni pertanto dovevano avere simmetria SU(3) rispetto al colore, oltre alla simmetria introdotta in precedenza.

Tutto ciò sembrava un bellissimo artificio matematico, ma pochi fisici credevano che dentro un adrone si trovassero realmente i quark, finchè nel 1968 alcuni esperimenti analoghi a quello di Rutherford, che bombardando atomi di oro con particelle α scoprì l'esistenza di un piccolo nucleo dentro l'atomo, mostrarono che l'adrone si comportava come se fosse avesse natura composta: inviando leptoni ad alta energia su un adrone si scoprì infatti che l'interazione avveniva come se l'adrone fosse composto di particelle puntiformi, dette partoni. Questo era indubbiamente un punto a favore per la teoria dei quark.

L'esistenza del colore come numero quantico fu in seguito confermata dal calcolo di sezioni d'urto; per esempio, nell'annichilazione di coppie e^+e^- in mesoni, un calcolo immediato dal diagramma di Feyman corrispondente mostra che il rapporto fra la sezione d'urto totale relativa a tale processo elettromagnetico e la

sezione d'urto relativa al processo simile in cui le particelle finali sono coppie $\mu^+\mu^-$, è uguale alla somma dei quadrati delle cariche elettriche (in unità e) di tutti quark. Il valore osservato sperimentalmente risulta invece il triplo: ciò è spiegato dal'esistenza del colore.

Negli anni '70, inoltre, si è scoperto che il colore può essere interpretato come la "carica forte", in analogia con la carica elettrica nell'elettromagnetismo. L'analogia è molto limitata, poichè mentre la carica elettrica può essere positiva o negativa, il colore può essere di tre "segni", ovvero va rappresentata su un piano e non su una retta; come conseguenza di ciò, i bosoni responsabili dell'interazione (i gluoni) sono di 8 tipi, e portano colore, mentre il bosone elettromagnetico, cioè il fotone, è unico e non porta carica elettrica.

La teoria relativa all'interazione forte, oggi detta "di colore", è una teoria di gauge non abeliana detta Cromodinamica Quantistica (QCD, da Quantum Chromo Dynamics). Essa sembra in grado di spiegare il "confinamento dei quark", ovvero il fatto che i quark non possono essere osservati liberi: infatti la QCD prevede un andamento della costante di accoppiamento con la distanza del tipo

$$\alpha(r) = \frac{k}{\log(r_0/r)} \qquad (1.1)$$

con r_0 dell'ordine di 1 fermi; chiaramente α cresce al crescere di r.

La maggior parte dei fisici oggi ritiene che tutte le interazioni possano essere spiegate con teorie di gauge non abeliane. Ciò è dovuto al successo della teoria dell'interazione elettrodebole di Weinberg e Salam, che ha avuto la maggiore conferma sperimentale nella scoperta dei bosoni dell'interazione debole. Tale teoria è una teoria di gauge non abeliana che unifica l'interazione elettromagnetica con

l'interazione debole.

La conferma della validità della QCD, che è una teoria di gauge non abeliana, costituirebbe una base importantissima per l'unificazione dei tutte le interazioni note in un'unica interazione fondamentale - notiamo che la teoria della relatività generale, ovvero la teoria dell'interazione gravitazionale, pur non essendo ancora una teoria quantistica, può essere interpretata come una teoria di gauge non abeliana.

Purtroppo i calcoli di QCD sono estremamente complessi e si possono calcolare osservabili solo in regime perturbativo. Per esempio gli stati legati di quark pesanti, come le particelle J/ψ ed Y, possono essere valutati con notevole precisione.

§ 1.1 - La formulazione su reticolo della QCD.

Volendo studiare le caratteristiche degli adroni, la cui distanza tipica è dell'ordine del fermi, l'approssimazione perturbativa non è più valida. Infatti questa è accettabile solo per grandi valori di q, dove q è il momento trasferito, poichè in tal caso si hanno piccole distanze in gioco e quindi una costante di accoppiamento sufficientemente piccola da permettere una trattazione perturbativa.

La soluzione analitica del problema di QCD d'altra parte è improponibile, data l'enorme complessità dei calcoli coinvolti. Un approccio al problema, allora, può essere basato su metodi numerici.

Il metodo proposto da K.Wilson consiste nel calcolare le grandezze dinamiche sui punti di un reticolo spazio-temporale, per mezzo di metodi tipici della meccanica statistica, e di ottenere i risultati fisici reali effettuando il passaggio al limite continuo per il passo reticolare tendente a zero.

Un numero ragionevole di punti per dimensione del reticolo, per rendere

trascurabili gli effetti della discretizzazione, è valutato essere circa 30. Pertanto si può costruire un reticolo, dentro il quale è contenuto l'adrone, con passo reticolare dell'ordine del decimo di fermi, per cui il reticolo si espande per circa 3 fermi, ovvero circa 3 volte il diametro dell'adrone, come indicato in fig. 1.1. Notiamo che il reticolo si intende come un reticolo periodico, nel senso che è ripetuto infinite volte accanto a se stesso lungo tutte le dimensioni; i conseguenti effetti sui calcoli sono però trascurabili (l'interazione forte ha un raggio d'azione di circa 1 fermi).

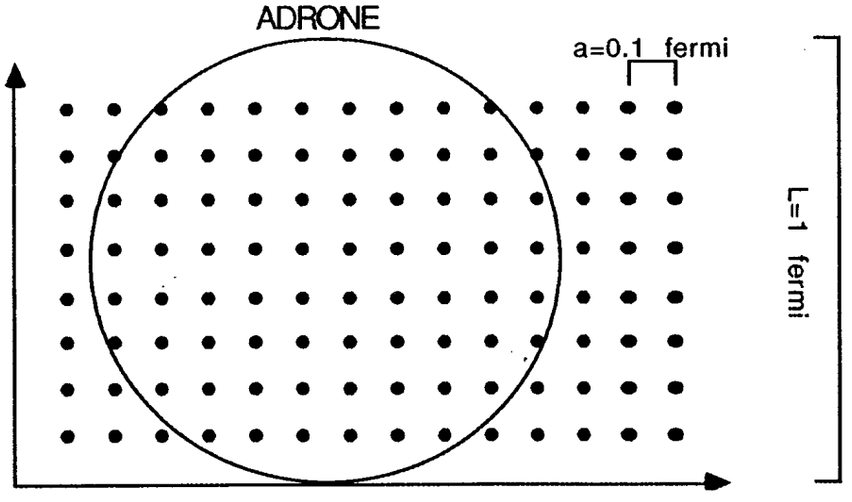

Fig. 1.1. Reticolo costruito su un adrone.

Consideriamo un reticolo spazio-temporale di N punti per dimensione, con una distanza fra punti adiacenti uguale ad a. In ogni punto del reticolo allora i campi gluonici saranno descritti da 4 matrici di SU(3), ognuna relativa al collegamento (link) fra tale punto ed i punti immediatamente vicini nelle 4 dimensioni, nel verso positivo degli assi coordinati. Avremo quindi $4N^4$ matrici di SU(3), poichè i punti del reticolo sono N^4. Allora ciascuna matrice U_{ij}, dove i rappresenta il generico punto del reticolo e varia da 1 ad N^4, mentre j rappresenta la generica dimensione

spazio-temporale e varia da 1 a 4, sarà definita come

$$U_{ij} = \exp\left(\sum_{a=1}^{8} i\, T^a A_{ij}^a \right) \qquad (1.2)$$

dove i T^a rappresentano le 8 matrici generatrici del gruppo SU(3) - corrispondenti ciascuna ad un tipo di gluone) e gli A_{ij}^a sono numeri reali che rappresentano una sorta di potenziale relativo a tale gluone; notiamo infatti che il potenziale elettromagnetico sul reticolo si scriverebbe come A_{ij} (senza indice a, ovvero un solo valore del potenziale); nell'equazione 1.2 non si confonda l'indice i con l'unità immaginaria).

Date le U_{ij}, una trasformazione di gauge è definita come

$$U_{ij} \dashrightarrow U'_{ij} = g(i)\, U_{ij}\, g^{-1}(i+1) \qquad (1.3)$$

dove le g(i) sono le matrici che definiscono la trasformazione di gauge.

L'azione S sarà una funzione delle matrici U_{ij}, che rappresentano le variabili dinamiche associate al campo di colore in ogni punto del reticolo. Poichè S deve essere invariante per rotazione locale di gauge, e poichè - come si ricava facilmente dalla 1.3 - una qualsiasi quantità calcolata su un qualsiasi circuito chiuso è invariante anch'essa, sarà sicuramente possibile esprimere l'azione S in funzione delle variabili U_{ij}.

Per trovare tale relazione, facciamo un'analogia con il campo elettromagnetico, che è descritto anch'esso da una teoria di gauge, però Abeliana, per cui le matrici U_{ij} si riducono a numeri complessi; la loro forma è la medesima dell'equazione 1.2, ma con l'indice a che assume un solo valore invece di 8, e $T^a=T^1=1$, per cui il calcolo risulterà molto più semplice.

L'invariante più semplice che sappiamo costruire è il cammino elementare costituito da 4 link giacenti su uno stesso piano coordinato, secondo la fig.1.2; il circuito così definito viene chiamato placchetta. Per ogni link avremo allora un valore di U_{ij}: chiamiamo U_1 il valore di U_{ij} fra il punto (n,m) ed il punto (n+1,m), e così via, in accordo alla figura 1.2. Se come invariante consideriamo la quantità $Tr(U_1U_2U_3U_4)$, avremo allora

$$Tr(U_1U_2U_3U_4)=Tr\ \{\exp[iA_0(n,m)]\ \exp[iA_1(n+1,m)]\ \exp[iA_0(n+1,m+1)]\ \exp[iA_1(n,m+1)]\} \tag{1.4}$$

dove A_0 rappresenta la componente del potenziale lungo le ascisse (nel disegno) ed A_1 lungo le coordinate. A questo punto, notando che $A_0(n+1,m+1)= -A_0(n,m+1)$ ed $A_1(n,m+1)= -A_1(n,m)$, sviluppando al primo ordine $A_1(n+1,m)= A_1(n,m) + a\,\partial_0 A_1(n,m)$ ed $A_0(n,m+1)= A_0(n,m) + a\,\partial_1 A_0(n,m)$ (il che è permesso se a è piccolo), e sviluppando anche l'esponenziale, in definitiva [sottintendendo che tutti i potenziali sono relativi al punto (n,m)] si ha

$$Tr(U_1U_2U_3U_4) = 1 + ia(\partial_0 A_1-\partial_1 A_0) + \frac{a^2}{4}(\partial_0 A_1-\partial_1 A_0)^2 + \ldots \tag{1.5}$$

(notiamo che avendo numeri complessi invece di matrici, l'operazione Tr è ininfluente). Sullo stesso piano vi sarà un'altra placchetta avente U_1 come link. Eseguendo lo stesso calcolo otteniamo un risultato simile alla 1.5, ma dove il termine al primo ordine ha il segno meno [cioè -ia(...)]. Sommando i due risultati, quindi, i termini lineari si elidono. La stessa cosa avviene sugli altri tre piani coordinati. Notando che l'azione S nel caso del campo elettromagnetico è data da

$$S = \int d^4x\ F_{\mu\nu}\ F_{\mu\nu} \tag{1.6}$$

con

$$F_{\mu\nu} = \partial_\mu A_\nu - \partial_\nu A_\mu \qquad (1.7)$$

è evidente che sommando i termini del tipo 1.4 (ovvero 1.5) su tutte le placchette aventi link a comune, otteniamo il valore dell'azione associata a tale link.

In SU(3) il calcolo è molto più complesso, ma il concetto è lo stesso. Un'altra osservazione da fare è che il calcolo numerico viene effettuato su un reticolo quadridimensionale a geometria euclidea, mentre lo spazio-tempo è uno spazio di Minkowski. I risultati ottenuti nello spazio euclideo comunque possono essere trasformati analiticamente e con esattezza nei risultati corretti, cioè relativi ad uno spazio di Minkowski.

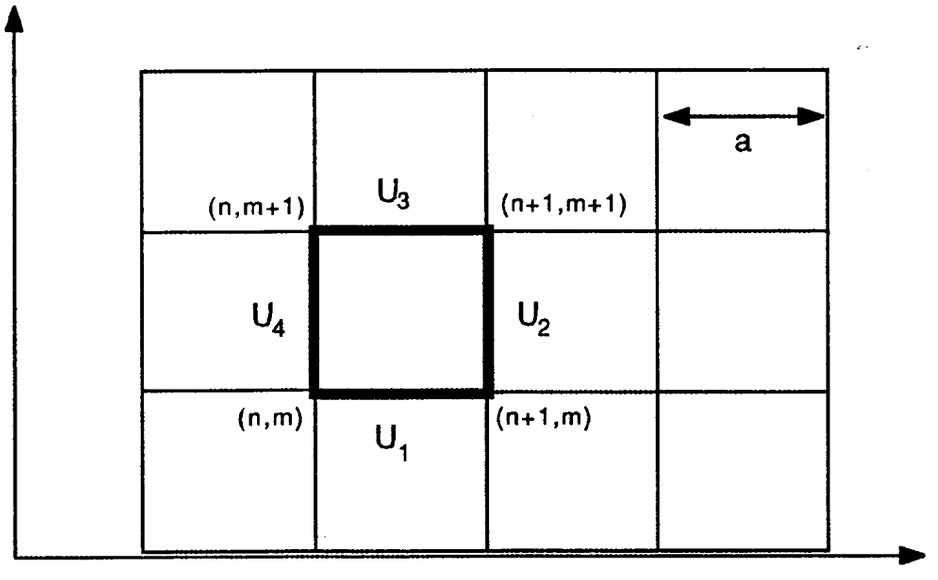

Fig. 1.2 - Placchetta o cammino elementare sul reticolo.

A questo punto sappiamo calcolare l'azione, e quindi, con metodi numerici,

partendo da una configurazione casuale delle variabili dinamiche (cioè delle varie matrici nei punti del reticolo), possiamo ottenere il valore minimo dell'azione, che corrisponderà al valore fisico effettivo delle variabili dinamiche, in accordo col principio della minima azione.

L'azione viene usata per calcolare il valor medio delle osservabili desiderate, in accordo con la definizione di grandezza fisica in un sistema statistico all'equilibrio termico:

$$<O> = \sum_n \frac{O(n)}{Z} \exp(-E_n/kt) \tag{1.8}$$

dove O è l'osservabile e Z è la funzione di ripartizione

$$Z = \sum_n \exp(-E_n/kt) \ . \tag{1.9}$$

Nel nostro caso avremo allora

$$<O> = \sum_{U_{ij}} \frac{O(U_{ij})}{Z} \exp(-S(U_{ij})) \tag{1.10}$$

con

$$Z = \sum_{U_{ij}} \exp(-S(U_{ij})) \ . \tag{1.11}$$

Il calcolo esplicito indicato dalle equazioni (1.10) ed (1.11) comportano un numero enorme di calcoli, che è praticamente impossibile completare.

§ 1.2 Metodo di Montecarlo.

Per ovviare al fatto che le equazioni 1.10 ed 1.11 risultano inutilizzabili all'atto pratico, si impiegano un metodo statistico detto metodo di Montecarlo.

Sappiamo dalla meccanica statistica che solo un piccolo insieme delle configurazioni possibili in un sistema statistico contribuiscono realmente al valor medio dell'osservabile nell'equazione 1.8: le configurazioni ad alte energie infatti danno un contributo trascurabile, a causa dell'esponenziale. Ciò è altrettanto vero per quanto riguarda la 1.10: solo le poche configurazioni con basso valore di S contribuiscono al valor medio che intendiamo calcolare.

Il metodo di Montecarlo consiste allora nel ricercare il minimo dell'azione e considerare le configurazioni in un intorno di tale minimo. In realtà vi sono più modi di applicazione del metodo di Montecarlo, ed i più significativi sono il metodo di Metropolis ed il bagno termico.

Per cercare la configurazione di equilibrio, ovvero di minimo S, verrà effettuato un certo numero di passaggi (updating), ovvero di variazioni della configurazione. Ad ogni passaggio viene associata una distribuzione di probabilità P(C,C'), corrispondente alla probabilità che alla configurazione C venga sostituita la configurazione C'. Ovviamente

$$P(C,C')\geq 0 \; ; \quad \sum_{C'} P(C,C') = 1 \tag{1.12}$$

Per trovare la configurazione di equilibrio si sostituisce alla generica matrice U_{ij} una matrice U'_{ij} tale che $P(U_{ij},U'_{ij}) = P(U'_{ij},U_{ij})$. Quindi si calcola la variazione dell'azione conseguente, ΔS. Se $\Delta S \leq 0$ la sostituzione viene accettata. In caso

contrario viene generato un numero casuale R fra 0 e 1, con distribuzione di probabilità uniforme, e la modifica viene allora accettata se $\exp(-\Delta S)>R$. Ciò serve per evitare di intrappolarsi in minimi locali, il che potrebbe succedere se le configurazioni con $\Delta S>0$ venissero sistematicamente rifiutate.

Il metodo di Metropolis consiste nell'effettuare questa operazione a turno su tutti i link, prima di passare ad effettuarla di nuovo su uno stesso link. Il metodo del bagno termico invece consiste nell'insistere sulla modifica di un link se la sostituzione in questione continua a dare configurazioni accettate; ciò significa che tale link era molto più lontano dal minimo rispetto ai link vicini, e l'operazione ripetuta più volte lo porta a livello di questi, ovvero lo pone in "equilibrio termico" con essi, da cui segue il termine di bagno termico. Quest'ultimo metodo è più veloce nel raggiungere l'equilibrio, ma la sua implementazione è più complessa rispetto al metodo di Metropolis.

§ 1.3 - Alcuni risultati di Cromodinamica Quantistica su reticolo ottenuti da APE [1].

Come citato nell'introduzione, il supercalcolatore APE, che sarà esaminato in dettaglio nei capitoli seguenti, è stato progettato per calcoli di Cromodinamica Quantistica su reticolo. Esaminiamo i risultati ottenuti da un prototipo ridotto di APE, chiamato Apetto, sulla determinazione delle masse delle glueballs.

Le glueballs sono stati legati di soli gluoni, la cui esistenza è prevista dalla teoria, ma sulle quali non esistono ancora evidenze sperimentali. Notiamo che i gluoni possono formare stati legati senza fermioni (quark), poichè portano anch'essi il colore.

Le masse sono state calcolate per gli stati 0^{++} e 2^{++}, dove il numero indica lo spin ed i due segni rispettivamente la parità e la coniugazione di carica; i risultati sono riportati in tabella 1.1, in unità 1/a ed in GeV, dove L indica il numero di punti per

dimensione spaziale nel reticolo. I punti lungo la dimensione temporale sono sempre 32.

L	m(0++)	m(2++)	m(0++)[Gev]
10	0.65(3)	0.6(3)	1.137 ± .060
12	0.76(4)	0.8(2)	1.330 ± .070
16	0.82(5)	1.0(2)	1.435 ± .090

Tabella 1.1

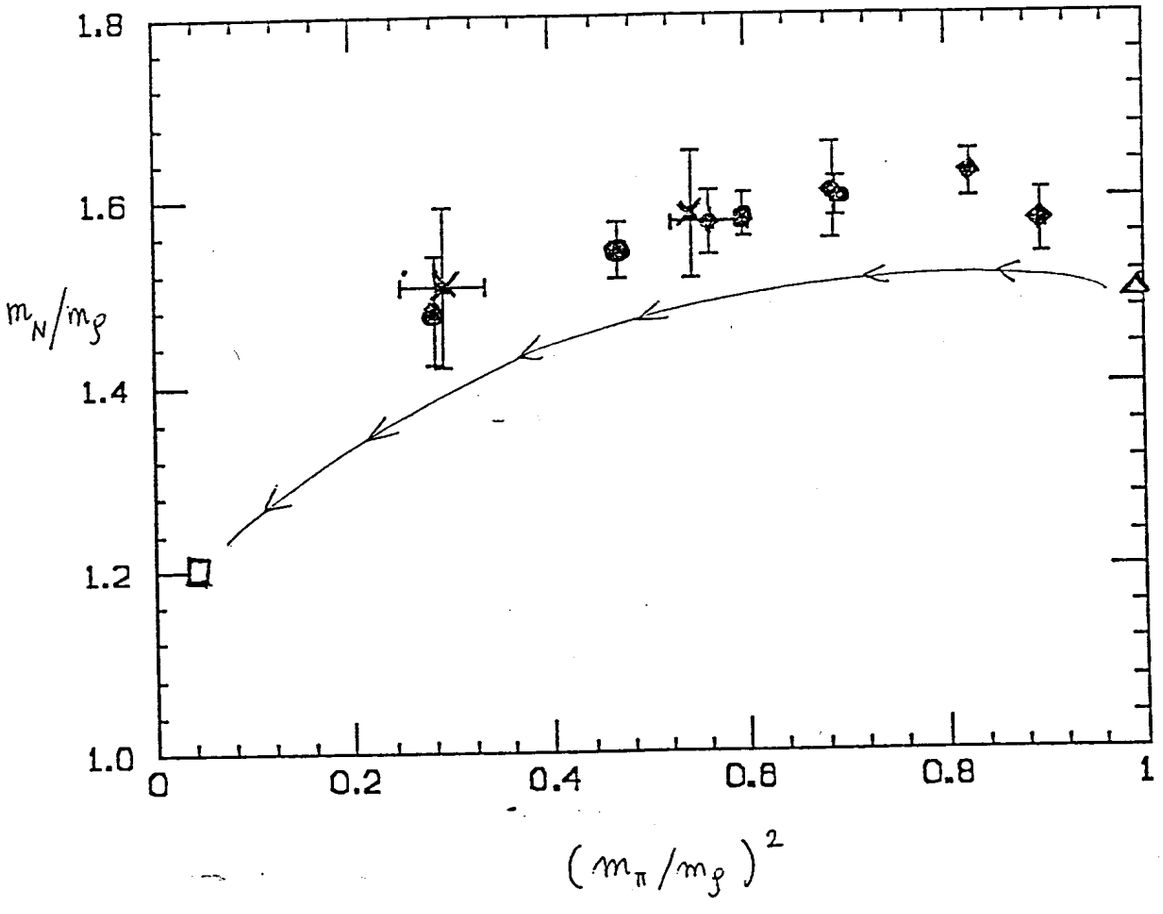

Fig. 1.3 - Primi risultati sugli stati fermionici.

Recentemente si è iniziato a studiare lo spettro di massa degli stati fermionici; i

risultati sono ancora preliminari, e citiamo solo il grafico di fig. 1.3. Esso riporta il rapporto (m_N / m_ρ) in funzione di $(m_\pi / m_\rho)^2$. I grossi punti e le crocette rappresentano i risultati ottenuti rispettivamente su reticoli $12^3 \cdot 24$ e $9^3 \cdot 24$ a $\beta = 5.9$ (β è definito come $6/g^2$ ed è quindi inversamente proporzionale ad α).

Capitolo 2

Calcolatori paralleli ed il supercalcolatore APE

§ 2.0 - Introduzione.

Il problema di Cromodinamica Quantistica su reticolo, esaminato nel precedente capitolo, non è l'unico problema fisico trattabile con metodi numerici. Nel paragrafo successivo esamineremo brevemente altri esempi di problemi fisici trattabili in modo analogo. La potenza di calcolo richiesta da tali problemi è molto elevata, e l'utilizzo di tipici calcolatori commerciali - come i familiari calcolatori della serie VAX, prodotti dalla Digital, o delle serie 43XX e 30XX, prodotti dalla IBM - è quasi improponibile. Infatti, stime attendibili effettuate sulla capacità di calcolo in operazioni in virgola mobile al secondo (flop/s, da floating point operations / second) su tali calcolatori attribuiscono loro una velocità dell'ordine di 1 Megaflop e di 10 Megaflop rispettivamente (le ditte sono restie a dichiarare i valori precisi).

Queste potenze risultano insufficienti per le applicazioni desiderate. Si pensi per esempio che una tipica simulazione di Montecarlo di QCD su un reticolo di 30 punti per dimensione richiede complessivamente circa 10^{15} flop, equivalenti a circa un miliardo di secondi (circa 30 anni) di calcolo continuato su un calcolatore da 1

Megaflop, senza considerare le operazioni di Input/Output, mentre lo stesso calcolo durerebbe solo 280 ore circa (meno di 12 giorni) su un calcolatore da 1 Gigaflop. Inoltre la memorizzazione delle configurazioni di gauge in tale problema richiede una memoria di qualche centinaia di Megabyte.

Il colossale sviluppo compiuto dalla microelettronica negli ultimi anni permette oggi di costruire calcolatori con una potenza di calcolo sull'ordine del Gigaflop impiegando componenti elettronici standard, purchè si modifichi radicalmente l'architettura del calcolatore. La nuova concezione architetturale consiste essenzialmente nell'abbandono di una macchina sequenziale, che esegue in successione temporale i calcoli desiderati, a vantaggio di una macchina parallela, in cui più unità equivalenti eseguono contemporaneamente e separatamente calcoli diversi: l'unità funzionale preposta all'esecuzione delle istruzioni diventa allora un "vettore" di unità separate. Affinchè si abbia un effettivo guadagno di prestazioni, è ovviamente opportuno che l'algoritmo di risoluzione dello specifico problema sia adatto ad una trattazione di questo tipo, ovvero abbia una intrinseca natura parallela. Gli algoritmi vettoriali sono molto adatti ad una trattazione parallela.

La possibilità offerta dalla attuale tecnologia elettronica di costruire supercalcolatori paralleli impiegando componenti standard o quasi-standard, è stata intuita ed apertamente dichiarata da molti fisici teorici, tra cui T.D.Lee e K.Wilson, ed in tale ottica si inserisce il progetto APE dell'INFN. Il progetto può apparire ambiziosissimo al profano, essendo difficile immaginarsi come un gruppo di pochi fisici possano riuscire a costruire una macchina 100-1000 volte più potente di quelle costruite da ditte come l'IBM; una possibile spiegazione in termini semplicissimi può essere data da una vaga analogia con quanto succede nella costruzione di prototipi di automobili per gare di Formula 1, in cui una piccola "scuderia" - ovvero un gruppo di

pochi ingegneri - riesce a costruire un'automobile molto più potente delle automobili comuni costruite dalle grandi case, ma certamente non altrettanto affidabile e ovviamente dedicata ad un uso specifico. Cionondimeno, esistono anche grandi case automobilistiche specializzate nella costruzione esclusiva di automobili potentissime, così come esistono ditte specializzate nella costruzione di supercalcolatori paralleli: l'esempio classico è la ditta Cray, pioniere in questo campo. Il costo di un supercalcolatore commerciale è però proibitivo (decine di miliardi di lire) se confrontato col costo totale di un supercalcolatore "artigianale" come APE (un miliardo di lire).

Notiamo che i calcolatori paralleli in generale costituiscono oggigiorno un argomento attualissimo e trovano vastissime applicazioni anche in altri settori di ricerca, come per esempio in campo militare, in campo industriale, in materia di image processing, o in materia di intelligenza artificiale.

§ 2.1 - Calcolo parallelo in fisica.

In questo paragrafo esamineremo velocemente alcuni esempi di problemi fisici che presentano caratteristiche intrinseche di parallelismo.

Il massimo grado di parallelismo si ha in analisi di dati in esperimenti di fisica delle particelle elementari; tali analisi vengono effettuate per mezzo di catene di calcolatori operanti simultaneamente su dati diversi ma utilizzando uno stesso software. Un esempio di tali catene, dette farm, si ha al CERN.

I calcolatori utilizzati in tale farm sono dei CERN/SLAC 3081/E; essi non sono dei veri e propri calcolatori ma dei semplici "emulatori" del calcolatore IBM 3081. Per "emulatore" si intende una macchina priva di periferiche e consistente

quasi esclusivamente nella nuda CPU, con struttura hardware non necessariamente simile al calcolatore emulato, ma capace di eseguire esattamente i programmi creati e compilati su quest'ultimo. L'efficienza di un 3081/E è circa 1/3 rispetto a quella di un "autentico" IBM 3081, ma il suo costo è inferiore di due ordini di grandezza o quasi. Come vedremo, l'unità principale di controllo del supercalcolatore APE consiste proprio in un 3081/E.

I vari emulatori costituiscono le varie unità funzionali separate operanti in parallelo, fra le quali vengono suddivisi i dati da elaborare; il programma di analisi è lo stesso per tutte le unità, ma ciascuna elabora dati diversi, per cui in caso di salti condizionati dipendenti dai dati stessi il programma può seguire strade diverse sulle varie unità. Il grado di parallelismo è massimo poichè (almeno in linea di principio) fra le varie unità non è necessaria alcuna comunicazione ed esse si comportano come entità separate.

Un esempio di problema fisico che comporta una comunicazione minima ma non nulla fra le diverse unità funzionali è il problema di teorie di gauge su reticolo descritto nel capitolo precedente. Esso è un problema locale in cui il calcolo in ciascun punto del reticolo dipende solo dai valori in quel punto e nei punti immediatamente vicini. Se distribuiamo i punti fra le varie unità, il calcolo su di un punto "di confine" necessiterà dei anche valori nel punto adiacente che si trova "al di là del confine", ovvero su un'unità adiacente. Pertanto in questo tipo di problema è necessaria una comunicazione "a livello di vicini immediati" fra le varie unità: la i-esima unità dovrà comunicare con la (i+1)-esima e con la (i-1)-esima unità. Dato che il numero di unità n è finito, in realtà la regola esatta per una "comunicazione circolare" prevede che la i-esima unità debba comunicare con la j-esima e la k-esima unità, dove

$$j = (i+1) \text{ MOD } n \tag{2.1}$$

e

$$k = (i-1) \text{ MOD } n \,. \tag{2.2}$$

Un problema in cui si ha minore località, e pertanto il livello di comunicazione necessario fra le varie unità funzionali può risultare maggiore, è l'integrazione di equazioni differenziali a derivate parziali. Tale problema è frequente in studi di fluidodinamica. Senza entrare nei dettagli dell'integrazione, notiamo che al crescere dell'ordine dell'equazione differenziale il calcolo dei valori in un certo punto evidentemente coinvolgeranno i valori di punti via via più distanti; avendo distribuito come al solito i vari punti del reticolo n-dimensionale (dove n è il numero di variabili coinvolte nell'equazione) fra le varie unità funzionali, potrà quindi presentarsi una necessità di comunicazione non solo "fra primi vicini", come nel caso del problema di QCD, ma anche "fra secondi vicini" o addirittura "fra k-esimi vicini", con $k > 2$. Comunque le equazioni differenziali a derivate parziali vengono spesso risolte non per integrazione diretta, ma per mezzo di metodi che utilizzano la trasformata di Fourier, scrivendo e risolvendo, cioè, le equazioni nello spazio reciproco.

Vediamo un altro esempio, stavolta più in dettaglio; si tratta di un problema ancora meno locale, e che necessita quindi di comunicazioni ancora maggiori fra le diverse unità funzionali.

Si voglia studiare l'evoluzione di una galassia a seguito della forza gravitazionale generata dalle sue stesse parti. Una galassia a spirale di dimensioni medio-grandi, come la nostra Via Lattea, è un disco di diametro dell'ordine dei 100000 anni luce e dello spessore massimo pari a circa 1/5 del diametro, contenente

circa 100 miliardi di stelle - oltre ad altra materia interstellare -. Il volume della galassia può essere diviso in N parti e si può studiare il moto di ciascuna parte calcolando la forza gravitazionale che agisce su di essa; tale forza è ovviamente generata dalle altre N-1 parti. Per avere una accettabile precisione il numero N non deve essere troppo piccolo (possiamo immaginare N>1000); d'altra parte, se vogliamo trattare la galassia come un sistema continuo, N non deve essere troppo grande (per esempio $N<10^8$, cosicchè il numero medio di stelle contenuto in ciascuna parte è >1000). Allora, partendo dalla configurazione iniziale, si può calcolare l'accelerazione $F_k(t)/m_k$ agente sulla k-esima parte ad intervalli di tempo Δt, con Δt opportunamente piccolo:

$$\mathbf{a}_k(t) = \sum_{i=1}^{N} \frac{G m_i m_k (\mathbf{r}_i - \mathbf{r}_k)}{|\mathbf{r}_i - \mathbf{r}_k|^3} \qquad (2.3)$$

dove il vettore $\mathbf{a}_k(t)$ è l'accelerazione sulla k-esima parte al tempo t, m_i è la massa della i-esima parte della galassia, ed i vettori posizione $\mathbf{r}_i$ ed $\mathbf{r}_k$ sono calcolati al tempo $t - |\mathbf{r}_i - \mathbf{r}_k|/c$ per tener conto del tempo di propagazione dell'interazione gravitazionale. Chiaramente questo calcolo non è locale: il calcolo della forza in un punto dipende dai valori di tutti gli altri punti. Se le N parti della galassia sono ripartite fra le varie unità funzionali, ciascuna unità dovrà comunicare con tutte le altre.

Notiamo che il passo successivo al calcolo della forza consiste nel calcolo dello spostamento dovuto all'accelerazione, che è l'integrazione di una derivata seconda totale rispetto al tempo ed è quindi un calcolo locale, in cui è probabile una comunicazione semplicemente fra primi vicini.

Questo è il modo più intuitivo di risolvere il problema, che può essere risolto

con un altro tipo di approccio.

Un altro problema tipicamente risolto con tecniche numeriche, e che ha caratteristiche non-locali, è la risoluzione di trasformate di Fourier. Esempi tipici in cui essa trova applicazione sono:

- analisi digitale di segnali elettrici;
- misure su strutture cristalline tramite diffrazione di raggi X;
- risoluzione numerica di equazioni differenziali a derivate parziali;

Se ci limitiamo ad uno spazio monodimensionale, il consueto reticolo spaziale si riduce ad una retta su cui sono considerati n punti, e la trasformata di Fourier nel punto k_i dello spazio reciproco diventa allora:

$$F(p_j) = \frac{1}{\sqrt{2\pi}} \sum_{k=1}^{N} f(t_k) \exp(-ip_j t_k) \; . \qquad (2.4)$$

In realtà questo algoritmo comporta un numero troppo grande di operazioni (dell'ordine di N^2), per cui abitualmente viene utilizzato un algoritmo più veloce, detto Fast Fourier Transform. In ogni caso, in più dimensioni il numero di operazioni cresce come N^{2D}, dove D è il numero di dimensioni.

Per concludere, è doveroso sottolineare che esistono molti altri problemi fisici trattabili con metodi di calcolo parallelo oltre quelli citati in questo paragrafo a titolo di esempio; fra i più importanti ricordiamo il calcolo del momento magnetico anomalo dell'elettrone, determinabile coi metodi perturbativi dell'Elettrodinamica Quantistica con l'ausilio dei diagrammi di Feynman, o i calcoli relativi alla meccanica statistica di sistemi magnetici ordinati; per quanto riguarda le tecniche usate, quest'ultimo è simile

al problema di teorie di gauge.

§ 2.2 - Architetture dei calcolatori e calcolatori paralleli.

Dato un qualsiasi calcolatore, la sua architettura viene solitamente individuata in una delle seguenti tre categorie di massima, secondo la classificazione proposta da M.J.Flynn nel 1966 sulla base della distribuzione delle istruzioni e dei dati fra le varie parti del calcolatore [2]:

- SISD (Single Instruction Single Data);
- SIMD (Single Instruction Multiple Data);
- MIMD (Multiple Instruction Multiple Data).

Un calcolatore con architettura SISD è semplicemente un convenzionale calcolatore sequenziale, in cui esiste un'unità di controllo che riceve e decodifica le istruzioni ed un'unità aritmetico-logica che le esegue, oltre ad un'unità di memoria ed un'unità di input-output; alcune istruzioni possono essere eseguite dall'unità di controllo stessa.

In un calcolatore con architettura SIMD, esistono più unità aritmetico-logiche che eseguono in maniera sincrona la medesima istruzione decodificata dall'unità di controllo, su dati differenti. Chiaramente l'architettura SIMD è un'architettura parallela, ed i calcolatori basati su di essa vengono chiamati "array processor". Essi consistono quindi di una parte "scalare", che consiste nell'unità di controllo - la qualepuò eseguire anche un sottoinsieme delle istruzioni -, ed una parte "vettoriale", che consiste nell'insieme delle diverse unità funzionali in parallelo.

In un calcolatore con architettura MIMD, infine, esistono più unità di controllo - a ciascuna delle quali è associata un'unità aritmetico-logica - le quali

operano in maniera asincrona su dati differenti, permettendo così l'esecuzione contemporanea di diverse parti del programma. Anch'essa chiaramente è un'architettura parallela, ed è indubbiamente l'architettura più sofisticata ed articolata, ma anche la più difficile da gestire.

Flynn individua in realtà una quarta categoria, chiamata MISD (Multiple Instruction Single Data); in un'architettura di questo tipo, più unità di controllo agirebbero sugli stessi dati. Non si conoscono esempi di calcolatori basati su questi questa architettura.

Come accennato, la classificazione di Flynn è una classificazione di massima e, specialmente col rapido sviluppo delle tecnologie e delle esigenze di calcolo, le architetture dei moderni calcolatori tendono a fuggire da una schematizzazione rigida ed a possedere spesso un'architettura composta.

Una caratteristica architetturale che non è contemplata dalla suddivisione di Flynn, ma che ha assunto un'importanza fondamentale nei calcolatori delle ultime generazioni, è la struttura in pipelining. Un'analogia immediata che descrive in termini semplici il principio del pipelining è il funzionamento di una catena di montaggio: diverse unità hardware preposte ad operare in successione su ciascuna istruzione, operano contemporaneamente su istruzioni successive. Analizziamo il funzionamento più in dettaglio.

Per poter essere eseguita, un'istruzione dovrà passare una procedura costituita di diverse fasi successive, le quali sono generalmente la fase di fetch (ovvero di lettura dalla memoria), la fase di decodifica ed infine la fase di esecuzione vera e propria - la quale peraltro può consistere a sua volta di varie fasi -, e solitamente ciascuna fase viene compiuta da una diversa unità hardware. In un tipo di funzionamento tradizionale, una nuova istruzione non può iniziare la procedura

finchè l'istruzione precedente non ha concluso la propria, cosicchè ad ogni istante risulta operante soltanto una delle N unità hardware corrispondenti alle N fasi della procedura; in tale tipo di funzionamento si nota un evidente spreco di prestazioni, giacchè ciascuna unità viene sfruttata per un tempo medio pari ad 1/N volte il tempo totale in cui essa è disponibile. Nel funzionamento in pipelining invece, quando un'istruzione passa dal k-esimo stadio al (k+1)-esimo, il k-esimo stadio non resta inattivo ma inizia ad operare immediatamente sull'istruzione successiva, proveniente dal (k-1)-esimo stadio. In tal modo tutte le unità hardware corrispondenti alle varie fasi della procedura sono sfruttate al massimo, e si dice che la "pipeline" è "piena". Il guadagno in prestazioni è uguale ad N solo nel caso in cui le varie fasi hanno tutte la stessa durata; in caso contrario il tempo di esecuzione totale (ovvero il tempo che intercorre fra l'uscita di due istruzioni successive dall'ultimo stadio della procedura) è pari alla durata della fase più lenta. Una conseguenza immediata è che il pipelining risulta efficiente purchè le durate delle varie fasi siano confrontabili, ovvero non vi sia un "collo di bottiglia" a rallentare l'intera procedura.

Nei moderni supercalcolatori paralleli, il parallelismo ed il pipelining sono generalmente sfruttati a più livelli: per esempio un array processor non solo avrà M unità funzionali parallele, ma i componenti all'interno di ciascuna unità funzionale saranno disposti a loro volta in una struttura parallela; inoltre sia l'unità centrale di controllo che le varie unità funzionali parallele opereranno in pipeline.

Vediamo adesso alcuni esempi di supercalcolatori attualmente esistenti. Il valore della capacità di calcolo dato per ciascun calcolatore va inteso come limite massimo teorico.

- Cray 1: costruito nel 1976, è il capostipite dei supercalcolatori; la sua

capacità di calcolo è di 160 Megaflop/s.

- Cray X/MP : basato su di un'architettura MIMD con fino a 4 CPU, ognuna con una capacità di calcolo di 200 Megaflop/s.

- Cray 2: basato anch'esso su di un'architettura MIMD, ha una capacità di calcolo di 1600 Megaflop/s.

- Cray 3: ancora in fase di realizzazione, avrà una capacità di calcolo di 10 Gigaflop/s.

- CDC Cyber 205 : costruito nel 1980, è basato su di un'architettura SIMD ed è capace di 800 Megaflop/s.

I supercalcolatori suddetti sono disponibili in commercio; su una scala più "artigianale", alcuni gruppi di ricerca hanno sviluppato dei supercalcolatori dedicati a simulazioni di teorie di gauge su reticolo, che sono i seguenti:

– ARRAY , costruito dalla Columbia University di New York: progettato nel 1984, è basato su di un'architettura MIMD. Nella sua versione finale dovrebbe disporre di una capacità di calcolo di 4 Gigaflop/s. Dal 1985 sta funzionando una versione ridotta a 256 Megaflop/s.

– GF11, costruito dallo Yorktown Research Center di New York: è basato su di un'architettura SIMD. E` ancora in fase di realizzazione, e la sua capacità di calcolo sarà di 11.52 Gigaflop/s.

- APE, costruito dall'INFN, il progetto cui ho collaborato per il lavoro di tesi: è basato su di un'architettura SIMD, ed ha una capacità di calcolo di 1 Gigaflop/s. La versione definitiva ha iniziato a funzionare in questo periodo, ma da oltre un anno sono funzionanti due versioni ridotte da 256 Megaflop/s, chiamate "Apetto".

§ 2.3 - Il supercalcolatore APE [3].

Al progetto APE (Array Processor Experiment) collaborano alcuni fisici teorici e sperimentali delle Università o delle sezioni dell'INFN di Roma, Pisa, Bologna e Cagliari, nonchè del CERN di Ginevra. Lo scopo del progetto APE è la realizzazione e l'utilizzazione di un supercalcolatore parallelo omonimo, dedicato ma non specializzato per calcoli di teorie di gauge su reticolo, e quindi in linea di principio utilizzabile in altri settori di ricerca fisica. Come già citato, APE rappresenta un esempio indicativo della nuova tendenza, che si sta delineando fra i fisici teorici, di progettare e costruire gli strumenti necessari alle loro ricerche, secondo un approccio da lungo tempo tipico dei fisici sperimentali.

Le caratteristiche fondamentali del supercalcolatore APE sono le seguenti:

- architettura SIMD (Single Instruction Multiple Data);
- aritmetica complessa, a 32 bit per la parte reale + 32 bit per la parte immaginaria (ovvero "precisione singola" su ogni numero floating point);
- ciclo di clock di 120 ns, pari ad una frequenza di clock di 8.333 MegaHz;
- capacità di calcolo di 1 Gigaflop/s;
- capacità di memoria di 1 Gigabyte (attualmente soltanto 256 Megabyte).

Notiamo che la frequenza di clock è relativamente bassa; la potenza di APE è dovuta al grande numero di operazioni compiute in un ciclo di clock, permesso dalla struttura parallela. Analizziamo tale struttura.

Essendo un array processor, APE consiste di una parte scalare, preposta al controllo, ed una parte vettoriale. Esaminiamo velocemente la parte vettoriale del calcolatore.

Le unità funzionali parallele di APE sono 16, e ciascuna è formata di due

schede: esse sono

- un'unità aritmetica a numeri in virgola mobile con una capacità di calcolo di 64 Megaflop/s, chiamata FPU (da Floating Point Unit);

- una memoria dinamica con capacità, nella versione finale, di 64 Megabyte - le memorie attualmente funzionanti hanno una capacità di 16 Megabyte.

Le 16 FPU pertanto effettuano in sincronia le medesime operazioni (Single Instruction), su dati differenti (Multiple Data), dal momento che ciascuna comunica (ovviamente sia in lettura che in scrittura) con la relativa memoria. In realtà l'accoppiamento fra le 16 memorie e le 16 FPU non è fisso: la generica memoria i-esima può essere accoppiata (ovvero può dialogare) con la j-esima scheda FPU, dove

$$j = (i+s) \text{ MOD } 16 . \qquad (2.5)$$

Il numero s, che può essere scelto fra 0 e 15, assume il medesimo valore per tutte le 16 memorie e rappresenta lo spostamento (shift), che è quindi rigido, nel senso che i 16 collegamenti si spostano uniformemente al cambiare di s. L'unità hardware preposta alla connessione consiste in una scheda chiamata switchnet (da switching network), o più semplicemente switchboard (da switchnet board).

Notiamo che l'equazione 2.5 è più generale delle equazioni 2.1 e 2.2 (con n=16), che avevamo scritto per le connessioni necessarie fra le varie unità funzionali nel caso del problema di teorie di gauge su reticolo. La 2.5 diventa la 2.1 per s=1 e diventa la 2.2 per s=15. Infatti è stata prevista una comunicazione fra i moduli paralleli maggiore di quella necessaria per lo specifico problema di teorie di gauge su reticolo, affinchè APE non risulti troppo specializzato e possa trovare applicazioni anche in problemi in cui sia ha necessità di comunicazione "fra k-esimi vicini", senza

limitazioni su k.

La parte scalare di APE consiste di due unità di controllo, che costituiscono il tema principale di questa tesi:

- un emulatore CERN/SLAC 3081/E, che ha la funzione di controllore principale; essa è l'unica parte di APE non originale;
- una scheda chiamata sequencer, che ha il compito di fornire alle FPU delle sequenze di microcodice ad esse relativo (il quale viene chiamato costituisce la maggior parte del codice memorizzato sul sequencer, detto nanocodice).

Il 3081/E, sul quale è memorizzato ed è eseguito il programma principale di APE, svolge l'aritmetica a numeri interi, tra cui anche il calcolo degli indirizzi di memoria, e controlla sia le memorie dinamiche che il sequencer inviando ad essi degli opportuni comandi. Il sequencer, sul quale è memorizzato il nanocodice per le FPU, invia a queste una sequenza di nanocodice ogni qual volta riceve il relativo comando (ed indirizzo di partenza) dal 3081/E; occorre sottolineare che il nanocodice è formato di due campi: uno a 64 bit, che è appunto il microcodice per le FPU, ed uno a 32 bit, che controlla il funzionamento interno del sequencer, e che ovviamente non viene inviato alle FPU. Il sequencer inoltre controlla il funzionamento dello switchboard, sulla base di dati opportuni forniti dal 3081/E.

Le memorie dinamiche ed il sequencer sono visti dal 3081/E come suoi moduli ausiliari. L'intero calcolatore, infatti, è basato sulla struttura modulare del 3081/E, essendo alloggiato su un unico rack da 19 pollici alto circa 2 metri, in cui sono montati due crate 3081/E. Ciascun crate 3081/E può contenere oltre 20 schede, e sul suo backplane viaggiano quattro bus a 32 bit per la comunicazione fra le schede:

- un bus di istruzioni, detto PMD (Program Memory Data); il codice portato da questo bus, detto opcode, si trova nella memoria programma del 3081/E ed è da esso gestito; è il codice che contiene i comandi sia per il 3081/E stesso che per il sequencer e le memorie dinamiche; ciascuna unità riconosce le proprie istruzioni ed ignora le altre;

- un bus per gli indirizzi, detto DMA (Data Memory Address); gli indirizzi portati da questo bus valgono sia per le memorie dinamiche, che per la memoria del 3081/E e (in fase di caricamento) del sequencer (che in tale fase viene visto dal 3081/E come parte della memoria programma; in fase di run gli indirizzi per il sequencer vengono dati su bit opportuni del bus PMD);

- due bus per i dati, detti ABUS e BBUS. Questi non sono i dati floating point da elaborare e servono soltanto in fase di caricamento della memoria del sequencer e delle memorie dinamiche, o per motivi ausiliari.

Nel crate superiore si trovano le sei schede del 3081/E, la scheda Sequencer e le schede di memoria. I bus del crate inferiore non sono collegati al 3081/E; in questo crate si trovano le FPU ed una scheda chiamata Transfer, che serve a trasferire sul backplane il nanocodice proveniente, tramite un cavo piatto a 32 bit (flat-cable), dal sequencer. Lo switchboard trova posto in mezzo ai due crate, in posizione esterna. Attraverso di esso passano vari flat-cable per la comunicazione fra le FPU e le memorie dinamiche; questi flat-cable costituiscono il Data-bus di APE, detto DBUS.

La definizione di "calcolatore" APE è in realtà impropria, e una definizione più esatta sarebbe quella di "processore": APE infatti non è dotato di sistemi di Input/Output e necessita quindi di un calcolatore ospite (host computer) che ne consenta l'uso dall'esterno. Tale sistema è attualmente un MicroVAX, collegato alla

rete INFNET dell'INFN. Il connessione fra APE ed il VAX procede da un'interfaccia del 3081/E attraverso un crate VME, collegato al QBUS (ovvero al bus principale del VAX). Sul VAX è siponibile un software chiamato HACK (Host Resident_APE Kernel Software) che permette l'impiego di APE ad alto livello col sistema operativo VMS del VAX. La compilazione dei programmi per APE viene effettuata sul VAX, dopo di che i codici per il 3081/E e per il sequencer (ovvero per le FPU) viene trasferito su APE. Per la programmazione di APE è stato creato un apposito linguaggio, familiarmente chiamato apese, che è simile al fortran ma che tiene conto fedelmente dell'hardware della macchina. Lo schema di massima del calcolatore APE è illustrato in fig. 2.1 (in cui per chiarezza solo 8 FPU e 8 memorie sono disegnate).

La versione completa di APE, la cui realizzazione è stata ultimata recentissimamente, si trova a Roma. Due versioni ridotte, chiamate Apetto, si trovano una a Roma ed una a Pisa. Apetto comprende soltanto 4 coppie FPU/memoria, per cui la sua capacità di calcolo è di 256 Megaflop/s. Per il 1989 si prevede che saranno funzionanti un Apetto a Bologna, un APE ed un Apetto a Pisa, ed a Roma un APE ed un APE21; come vedremo, APE21 è una versione di APE con capacità di calcolo pari a 2 Gigaflop, attualmente in corso di progettazione.

Nei prossimi paragrafi saranno analizzate più in dettaglio le singole parti di APE, tranne le due unità di controllo - il 3081/E ed il sequencer -, alla cui descrizione delle quali è dedicato un'intero capitolo, il capitolo 4.

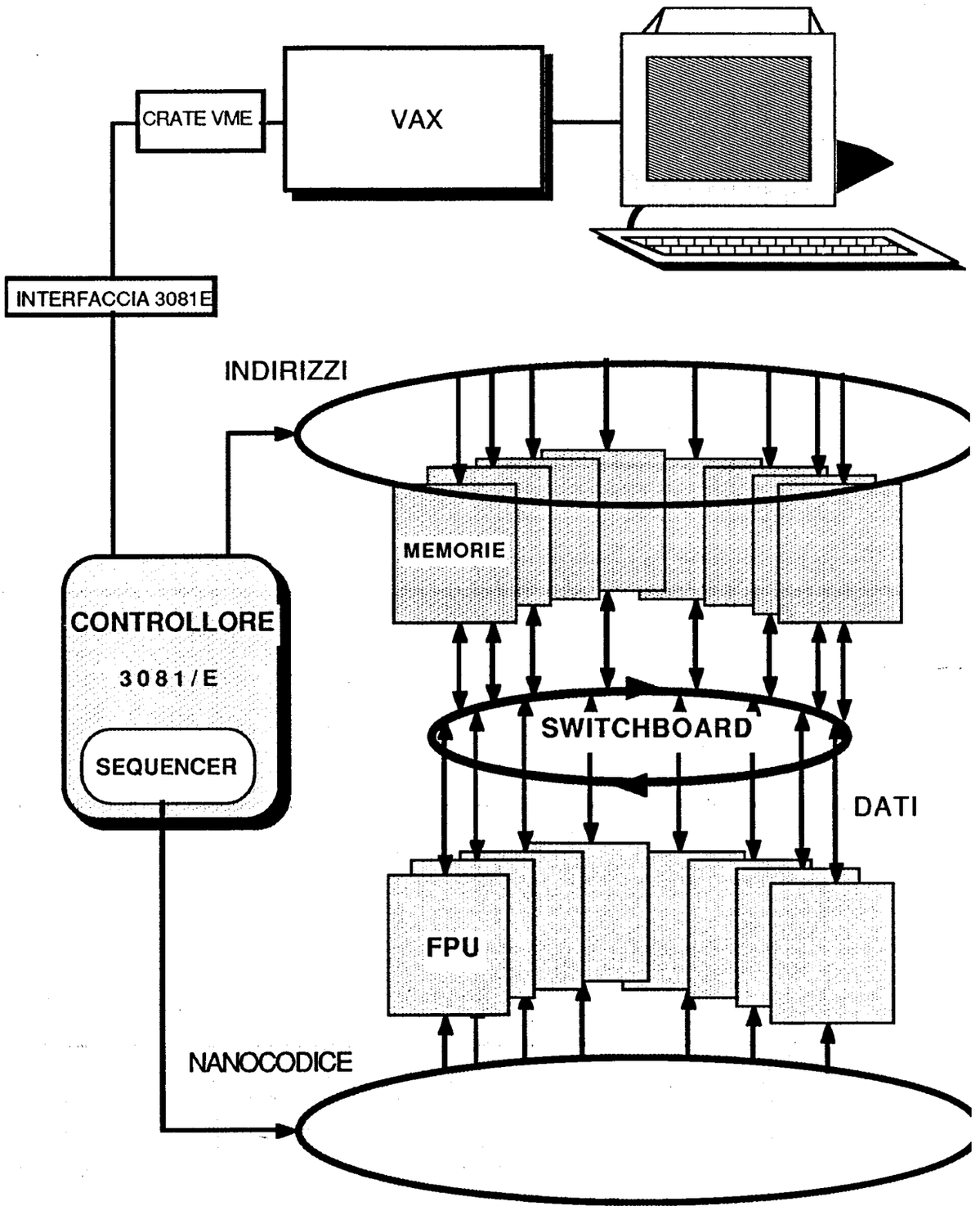

Fig. 2.1 - Schema di massima del supercalcolatore APE.

§ 2.4 - Le Unità Floating Point di APE.

Le Unità Floating Point di APE (FPU) sono state progettate tenendo presente il problema di simulazione di una teoria di gauge SU(3) su reticolo, che prevede un grande numero di moltiplicazioni fra matrici SU(3); tali moltiplicazioni si traducono in prodotti ed addizioni fra numeri complessi. Pertanto le FPU hanno una struttura ottimizzata per l'esecuzione dell'operazione A=B+C*D, fra numeri complessi. Ciascun numero complesso è rappresentato da 64 bit, di cui 32 per la parte reale e 32 per la parte immaginaria; si ha pertanto precisione singola su ciascun numero reale. Vediamo la struttura di una singola scheda FPU.

Il trasferimento dei dati da/a lo switchboard, ovvero da/a la memoria dinamica, avviene attraverso due registri - uno per la parte reale ed uno per la parte immaginaria -, ognuno dei quali può contenere 64 numeri floating point. Tali registri sono realizzati utilizzando 4 chip WTL1066 della Weitek, che sono chip VLSI (Very Large Integration System) ad alta prestazione. Per effettuare l'operazione fra numeri complessi A=B+C*D vengono impiegati 4 moltiplicatori e 4 addizionatori floating point, che sono rispettivamente i chip CMOS VLSI WTL1232 e WTL1233, anch'essi della Weitek. Tali chip funzionano in pipeline, cosicchè possono iniziare una nuova operazione in virgola mobile ogni ciclo di clock, ed il risultato di ogni singola operazione viene ottenuto dopo 5 cicli. Disponendo in un modo opportuno i 12 chip, "a regime" (cioè dopo che la pipeline si sia riempita) si ottiene un risultato dell'operazione completa ogni ciclo di clock. La struttura impiegata è illustrata in fig. 2.2. Gli stadi seguiti dagli operandi sono i seguenti:

- moltiplicazione, tramite i 4 moltiplicatori, delle parti reali ed immaginarie dei numeri complessi B e C, contenute nei quattro registri: ReB*ReC, ReB*ImC,

ImB*ReC, ImB*ImC;

- calcolo, tramite 2 dei 4 addizionatori, della parte reale e della parte immaginaria del prodotto B*C in funzione dei prodotti calcolati allo stadio precedente: Re(B*C)=ReB*ReC, Im(B*C)=ReB*ImC+ImB*ReC;

- calcolo, tramite i rimanenti 2 addizionatori, che vengono chiamati accumulatori, delle due parti del risultato, ReA ed ImA, utilizzando i risultati dello stadio precedente ed il valore di D preso dai registri: ReA=Re(B*C)+ReD, ImA=Im(A*B)+Im(D).

Il risultato viene portato ai registri per poter essere usato a sua volta, o trasferito alle memorie dinamiche. Il tempo totale dell'operazione è di 18 cicli, ma, come già notato, il funzionamento in pipelining permette di ottenere l'uscita di un risultato ogni ciclo di clock.

Avendo 4 addizionatori e 4 moltiplicatori, ovvero 8 chip, e poichè ciascuno di essi dà un risultato ogni ciclo di clock, il numero di operazioni effettuate in un secondo sarà pari ad 8 volte la frequenza di clock; pertanto ciascuna FPU ha una capacità di calcolo di 64 Megaflop/s. Naturalmente questa valore della capacità di calcolo è quello massimo teorico, e può essere raggiunto solo con aritmetica complessa; è comunque possibile modificare, sotto il controllo del programma, alcune connessioni fra i chip in modo da rendere piuttosto efficienti anche operazioni con aritmetica reale.

Oltre alla struttura dedicata alla moltiplicazione di matrici, ciascuna FPU contiene anche l'hardware necessario all'implementazione delle strutture logiche IF...THEN...ELSE, previste dall'Apese.

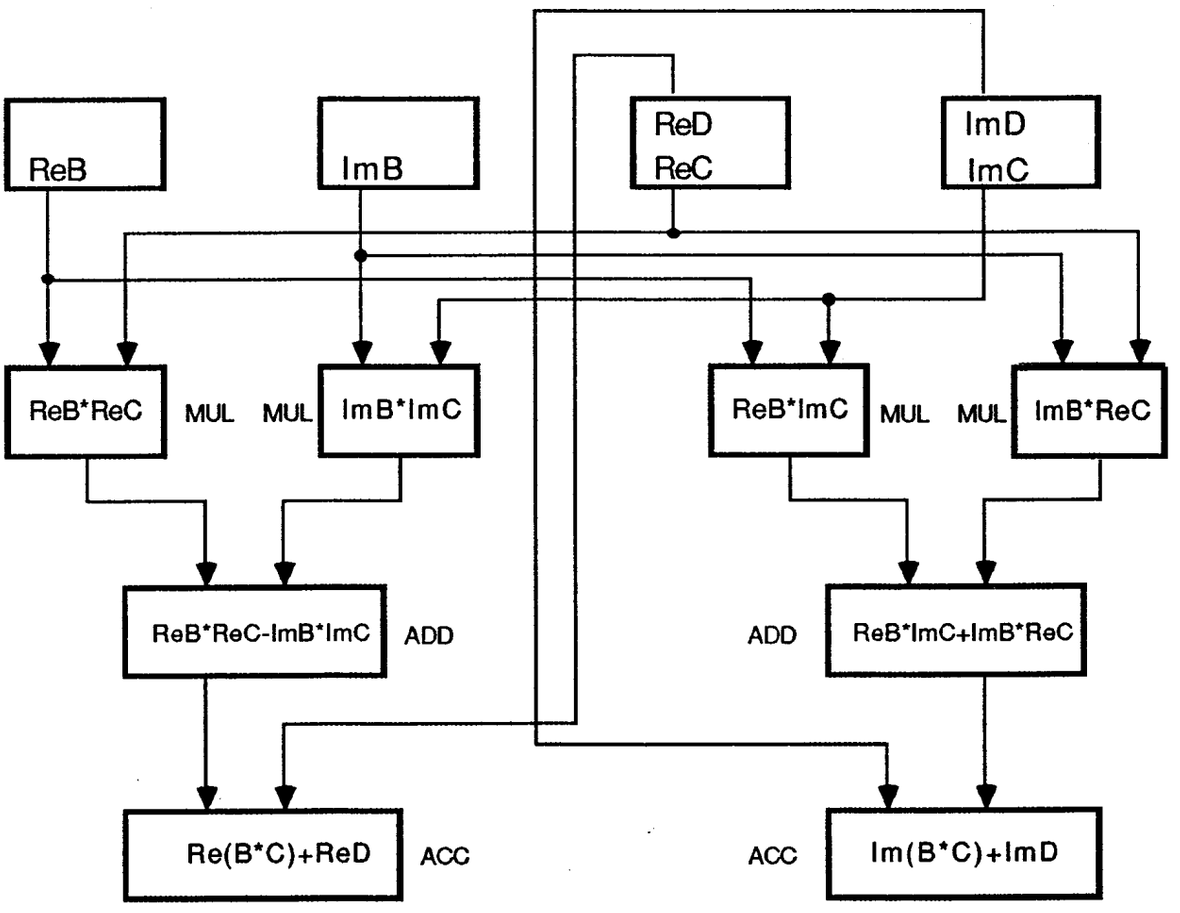

Fig. 2.2 - La struttura hardware per l'operazione complessa A=B*C+D nelle FPU di APE.

Esistono due tipi di IF su APE: un primo tipo, detto locale, riguarda condizioni che devono essere soddisfatte su valori contenuti in una singola scheda FPU e non procura alcun ritardo nell'esecuzione di un programma, comportando in ultima analisi una decisione sulla scrittura o non scrittura dei dati dalle FPU alla Memoria. Un secondo tipo, detto globale, riguarda invece condizioni che devono essere soddisfatte simultaneamente (AND logico); se questo accade, opportuni

segnali (FPUSTRE, FPUSTIM) vengono mandati attraverso il sequencer al 3081/E, e la condizione può modificare l'esecuzione del programma. L'IF globale attualmente non è ancora implementato, pertanto il 3081/E non può essere condizionato dalle FPU - per inciso, questa caratteristica ha permesso l'analisi sulle attuali prestazioni del 3081/E, esposta nel cap.5 -. In ambedue i tipi di IF, l'hardware permette fino a 4 livelli di annidamento.

Infine l'hardware della FPU permette di ottenere risultati approssimati delle funzioni esponenziale, logaritmo, radice quadrata ed inverso di un numero. Pertanto, pur essendo specializzate nell'esecuzione dell'operazione complessa A=B*C+D, le FPU possono essere utilizzate come unità logico-aritmetiche per scopi più generali.

§ 2.5 - Le memorie dinamiche di APE.

La necessità di disporre di una grande memoria per i dati floating point impone la scelta di memorie dinamiche, che sono più dense ed economiche delle memorie statiche; gli svantaggi delle memorie dinamiche rispetto a quelle statiche sono una maggiore lentezza e la necessità di un refresh periodico.

D'altra parte è strettamente necessario poter disporre di un tempo veloce di accesso alle memorie per non rendere inutile l'alta efficienza delle FPU: ad esempio, durante le moltiplicazioni fra matrici di SU(3) è richiesto in media il trasferimento di un numero complesso (64 bit = 8 bytes) ogni ciclo di clock, il che è equivalente ad avere una banda passante di 64 Megabyte/s. Per migliorare le prestazioni quindi anche nelle memorie dinamiche si è utilizzata una struttura basata sul principio del pipelining: la memoria è suddivisa in 8 banchi, individuati dai 3 bit meno significativi dell'indirizzo, e ad ogni ciclo di clock può essere richiesto un accesso ad un banco diverso; pertanto, se il tempo di accesso ad una parola è inferiore ad 8 cicli di clock,

in caso di accessi ad indirizzi sequenziali (in cui gli otto banchi vengono chiamati uno dietro l'altro) otterremo in totale un accesso alla memoria ogni ciclo di clock. In realtà il tempo di accesso ad una singola parola è di 3 cicli; quindi un trasferimento di n parole in successione richiede n+2 cicli di clock.

Poichè la memoria è piuttosto grande, la probabilità di che per motivi casuali il valore di un bit si modifichi è non trascurabile; pertanto sulla scheda vengono utilizzati dei chip EDAC (Error Detection And Correction) che calcolano, secondo un opportuno algoritmo, 7 bit ogni 32 da memorizzare; tali 7 bit vengono anch'essi memorizzati, ed in fase di lettura il chip consente la rivelazione e la correzione di un eventuale errore singolo, e la rivelazione semplice di un eventuale errore doppio. La probabilità di un errore triplo può essere assolutamente trascurabile.

Nella versione attuale delle memorie sono utilizzati chip di memoria DRAM (Dynamic RAM) da 256 Kilobit ciascuno, nel montaggio SIP (Single In Parallel), che consentono un'altissima densità. La capacità totale di memoria in una scheda è di 16 Megabyte. Sono attualmente in fase di realizzazione schede di memoria da 64 Megabyte.

Come già notato, l'indirizzamento delle memorie viene realizzato dal 3081/E sul bus DMA; esse possono essere indirizzate sia singolarmente, che globalmente, per mezzo di un bit di "broadcast" sul bus DMA. In fase di run, in virtù del parallelismo di APE, si ha ovviamente quest'ultimo caso.

§ 2.6 - Lo switchnet di APE.

Come già evidenziato, lo switchnet (o switchboard) è la scheda che permette la comunicazione delle generica i-esima scheda di memoria dinamica con la j-esima

scheda FPU, dove j è dato dall'equazione 2.5, che ricordiamo:

$$j = (i+s) \text{ MOD } 16 . \qquad (2.5)$$

Nel caso di Apetto, nell'equazione evidentemente risulterà un 4 al posto del 16.

Lo shift s è lo stesso per tutte le memorie dinamiche, e può essere cambiato ad ogni operazione di lettura/scrittura, in quanto il suo valore (a 4 bit) è affiancato nel DMA all'indirizzo (a 24 bit) relativo all'operazione. Tali 4 bit ed altri bit del DMA e del PMD necessari al controllo dello switchboard, vengono gestiti e tradotti in maniera opportuna, comprensibile allo switchboard, da un'unità hardqare che trova posto sulla scheda sequencer e che viene chiamata Switchboard Control Unit. Tale unità sarà esaminata dettagliatamente nel capitolo 4.

Le prime versioni della scheda switchboard sono state costruite per Apetto, e prevedono uno shift da 0 a 3. Esse sono state realizzate con l'ausilio di alcuni chip standard a logica programmabile detti PAL (Programmable Array Logic). Tali versioni di switchboard sono dette trasparenti, in quanto il ritardo che comportano sul DBUS è molto minore di 120 ns e non causano alcun ciclo di ritardo nel trasferimento.

Le ultime versioni, che sono state costruite per APE intero e prevedono quindi uno shift da 0 a 15, impiegano invece dei chip basati sulla tecnologia CMOS gate array, che sono stati progettati all'interno della collaborazione e costruiti dalla ditta Plessey Ltd (i chip commissionati in tale modo vengono chiamati "chip custom"). La realizzazione della logica di connessione sarebbe stata infatti problematica utilizzando chip PAL, che peraltro avrebbero comportato un consumo eccessivo e conseguenti problemi di dissipazione termica. Le nuove versioni dello

switchboard causano però un ciclo di ritardo alla comunicazione sul DBUS. Ciò comunque non costituisce un problema poichè lo switchboard opera in pipelining e quindi tale ciclo di ritardo non influenza la banda passante. Queste nuove versioni di switchboard vengono chiamate pipelined.

§ 2.7 - Il compilatore di APE.

Come già detto, la stesura e la compilazione dei programmi per APE viene effettuata sull'host computer. In questo paragrafo esamineremo brevemente la struttura del compilatore.

I compiti svolti dal compilatore sono i seguenti:

- generazione del microcodice eseguibile per il 3081/E (opcode);
- generazione del microcodice eseguibile per le FPU (nanocodice);
- ottimizzazione del nanocodice, per il migliore sfruttamento delle FPU;
- sincronizzazione dei due codici.

Per compiere tali oprazioni, il compilatore prevede vari stadi successivi di compilazione, ed ogni stadio genera diversi file; di questi, alcuni sono file ausiliari, altri sono file necessari agli stadi successivi del compilatore, ed il file .lex contiene il codice definitivo per APE. I nomi del file prendono la radice del nome del programma APE in compilazione, ed un'estensione che individua i vari tipi di file (p.es. il file esempio.lex conterrà il codice eseguibile del programma esempio); l'estensione del listato del programma in apese è .ape. Vediamo allora i vari passi della compilazione, ed i file da essi generati - soltanto i file più importanti verranno brevemente descritti. La compilazione varia leggermente a seconda che venga impiegato uno switchboard trasparente o uno switchboard pipelined. Il nome degli

stadi per il caso pipelined è posto fra parentesi.

Il primo stadio della compilazione è Cmpape (Cmpape-p). Tale stadio genera il codice per le FPU, che verrà poi migliorato e sincronizzato negli stadi successivi, e genera la successione delle macro-istruzioni per il 3081/E. Le macro-istruzioni, dette brevemente macro, sono sequenze di istruzioni 3081/E elementari, appositamente create per APE; esistono 26 tipi diversi di macro, sebbene soltanto una decina vengano utilizzate abitualmente. Partendo dal file .ape, i file generati da Cmpape sono i file .smb, .lex, .obe e .par. Il file .lex è la prima versione del codice per le FPU, che verrà aggiornata negli stadi successivi. Il file .obe contiene le informazioni fondamentali relative a tutte le macro del programma. Il file .par contiene altre informazioni relative al codice per le FPU.

Dopo Cmpape, la compilazione si divide in due strade parallele: una è Mcrape (Mcrape-p) che ha in input il file .obe; l'altra è Triape (Triape) che ha in input il file .par.

Lo stadio Mcrape, (Mcrape_p), detto anche macroespansore, genera i file .mle e .mcd. Il primo contiene informazioni relative alla lunghezza delle macro, ed il secondo contiene il microcodice quasi definitivo per il 3081/E.

Lo stadio Triape tende a riunire, per quanto possibile, le operazioni FPU in operazioni del tipo A=B*C+D, nella cui esecuzione le FPU sono estremamente efficienti. Il file generato, contenente le relative informazioni, è il file .tri.

Lo stadio successivo, che si trova sul "ramo" di Triape, è Optape (Opt-pipe), detto anche Ottimizzatore. Ha in input i file .lex, .tri e ,mle (quest'ultimo necessario alla sincronizzazione col codice 3081/E) e la sua funzione consiste nell'ottimizzare il codice FPU. I file generati sono i file .lab, .tim, .hex, .mas ed una versione

aggiornata del file .lex.

Il file .lab contiene le informazioni relative alle label, che sono le istruzioni di partenza delle sequenze di nanocodice sul sequencer.

Il file .tim contiene informazioni relative alla sincronizzazione fra i codici per le FPU e per il 3081/E.

Il file .mas è ottenibile solo con un'opzione nella richiesta del comando DCL @optape (@opt-pipe): occorre cioè chiedere @optape/mas (@opt-pipe/mas); tale file contiene informazioni dettagliate sul microcodice per le FPU.

Si ha quindi lo stadio Lnkape (Lnkape-p) che corrisponde alla fase di linking. I file in input sono i file .mcd, .lab, .tim e .lex. I file generati sono il file .lmc, che contiene il codice definitvo per il 3081/E, ed la versione quasi definitiva del file .lex.

L'ultimo stadio è New-astot (New-astot), che riunisce i due file suddetti nel file .lex, in una forma che ne permette il caricamento su APE.

Ciascuno stadio della compilazione può essere effettuato separatamente per mezzo dell'opportuno comando DCL (p.es. @cmpape nomedelprogramma). La compilazione completa si ottiene col comando Apec (Apec-p). Dopo la compilazione, il caricamento si ottiene col comando Apel, ed il debugger per l'esecuzione del programma e il trattamento dei dati viene chiamato col comando Hack.

Nel capitolo 5, che riguarda l'analisi delle prestazioni del 3081/E in APE, i file .obe, .mle, .tim, .mas saranno analizzati in maggior dettaglio.

§ 2.8 - Futuri sviluppi del progetto APE.

Gli sviluppi previsti per il progetto APE sono due, uno a breve scadenza, ed

uno a lunga scadenza.

Il primo riguarda la costruzione di APE21, che prevede l'accoppiamento di 2 FPU per ciascuna memoria; il numero totale di FPU salirebbe in tal caso a 32, e la capacità di calcolo di APE21 sarebbe di 2 Gigaflop/s. I problemi di principio legati alla progettazione di APE21 sono limitati, poichè la struttura di APE rimarrebbe essenzialmente immutata. Si prevede che APE21 sarà funzionante nel 1989.

Il secondo riguarda la realizzazione di SuperAPE. Il progetto, molto ambizioso, prevede una capacità di calcolo di 100 Gigaflop. Poichè appare impossibile un aumento significativo della potenza delle FPU, si sta progettando la loro miniaturizzazione in chip dedicati (chip custom), la cui costruzione sarebbe affidata ad una ditta specializzata. Tale miniaturizzazione prevede la presenza di quattro nodi su una singola scheda, dove un nodo è costituito da una FPU con relativa memoria e circuito di switching (analogo allo switchnet ma con shift più limitato). Si prevede che SuperAPE, detto anche APE100, possa funzionare nei primi anni '90.

Nota successiva (2011).

Il testo della tesi prosegue fino a pag. 160 circa, dove iniziano le Appendici, che comprendono i dettagli hardware e software, ovvero:

- gli schemi elettrici del sequencer e dei possibili sostituti delle unità di controllo;

- i programmi per le statistiche, scritti in linguaggio C;

- le simulazioni dei possibili sostituti, scritti nei linguaggi macchina del processore Fairchild Clipper e dei processori Weitek WTL 7136 e 7137.

In questo documento tali parti non vengono riportate.

Referenze.

M.Albanese et al., Phys. Lett. 192 B (1987) 163

M.Albanese et al., Phys. Lett. 197 B (1987) 400

P.Bacilieri et al., Preprint ROM2F/88/001 in corso di stampa su Phys. Lett .B.

M.J.Flynn, Very high speed computing systems, Proc.IEEE 54 (Dec.1966).

M.Albanese et al., The APE computer: an array processor optimized for lattice gauge theories simulations, Comp. Phys. Comm. 45 (1987) 345.

P.Bacilieri et al., APE: a fast array processor for physics simulations - in corso di Pubblicazione su Proc. of the 3rd International Conference on Supercomputers - Boston '88.

P.M. Farran et al., Proc. of the conference on computing in high energy physics, D.Hertzbeng e W.Hooglang editori (North Holland, Amsterdam 1986).

E.Simeone, The switchnet of the APE processor, Preprint APE/1987/3.

F.Coppola and A.Lai, The sequencer of the APE processor, Preprint APE/1987/2.

Fairchild C. (1986), Clipper: User's manual (Prentice Hall)

Weitek C. (1986), WTL 7136 32-bit CMOS Sequencer, Advance data.

Weitek C. (1986), WTL 7137 32-bit CMOS Integer Processor, Preliminary data.

Bibliografia.

E.Close (1979), An Introduction to Quarks and Partons (Academic Press).

Coppola and A.Lai, Performance of the 3081/E in the APE Processor, Preprint APE/1987/4.

Coppola and A.Lai, Performance of two possible new controllers for the APE Processor, Preprint APE/1987/5.

I.Creutz. (1983), Quarks, gluons and lattices (Cambridge University Press).

M. Creutz, L. Jacobs, C. Rebbi (1983), Phys. Rep., 95, 201.

G. De Vincentiis (1987), Tesi di Laurea, Università di Pisa.

FAIRCHILD C. (1986), CLIPPER: Product Description (Prentice Hall).

F. Flore, (1987), Tesi di Laurea, Università di Pisa.

R.W. Hockney and C.R. Jesshope (1981), Parallel Computers, A.Hilger Ltd., Bristol.

A. Lai (1987),Tesi di Laurea, Universita` di Cagliari.

G. Salina (1985). Tesi di Laurea, Università "La Sapienza", Roma.

Errata Corrige

Lungo le 160 pagine del testo sono disseminati circa 40 errori, per la maggior parte insignificanti; qui sono elencati quelli che potrebbero ostacolare una comprensione immediata.

Pag.24, riga 10: in luogo di "punto k_i" si legga "punto p_j".

Pag.31, riga 7: in luogo di "il quale viene chiamato costituisce" si legga "il quale costituisce".

Pag.33, riga 3: in luogo di "Sul VAX è siponibile" si legga "Sul VAX è disponibile".

Pag.36, riga 4: in luogo di "Re(B*C)=ReB*ReC" si legga "Re(B*C)=ReB*ReC-ImB*ImC".

Pag.72, fig.4.3: in luogo dei numeri 11 e 12 si leggano i numeri 7 ed 8 rispettivamente.

Pag.79, prima riga dopo la tabella: in luogo di "2, 4, 8 o 16" si legga "2, 4 o 8".

Pag.99, equazione 5.4: l'indice nella sommatoria è chiaramente j e non i.

Pag.80, riga 18: in luogo di "fissati scelti per mezo di un dip-switch, mai bit..." si legga "scelti per mezzo di un dip-switch, ma i bit...".

Pag.107, righe 5,...,10: da ignorare (sono la duplicazione di righe di una pagina precedente).

Pag.107, riga 12: in luogo di "comunque norevole" si legga "comunque notevole".

Pag.121, righe 1,2,3 (fino a "Supervisor"): da ignorare (sono la duplicazione di righe di una pagina precedente).

Pag.128, righe 1,...,4: idem.

Pag.138, riga 6: in luogo di "entre" si legga "mentre".

Pag.147, riga 21: in luogo di "macro-espansoreatore" si legga "macro-espansore".