

高並列計算機 QCDPAX と数値シミュレーション

筑波大学 電子情報工学系 小柳義夫 (Yoshio Oyanagi)

0. はじめに

本研究は、科学研究費補助金（特別推進研究）「格子ゲージ理論によるハドロンの質量およびクォーク・グルーオン相構造の解明」（研究代表者岩崎洋一、分担者星野 力、白川友紀、小柳義夫、吉江友照、川合敏雄）に基づく。

1. 研究目的

我々の研究目的は、「クォーク、グルーオンの基礎理論と目される格子ゲージ理論に基づき、超並列型計算機を用いた数値計算によって、陽子、パイ中間子などのハドロンの質量、グルーボールの質量、未知の H 粒子の存在性、存在する場合その質量等を計算する。また、クォーク・プラズマ相への相転移の次数、相転移点を求める」ことである。以下に、これらの研究の重要性、並列計算機を用いる必然性を述べ、最後に研究目的を更に具体的に詳しく述べる。

素粒子のうち、陽子、中性子、パイ中間子など強い相互作用をする粒子はハドロンと総称される。ハドロンはクォーク、グルーオンから構成される。このクォーク、グルーオンの基礎法則と目されるのが量子色力学 (QCD) である。しかし、いまだに量子色力学の定量的検証はない。その困難さは、量子電磁気 (QED) において用いられた摂動論を適用できないことに由来する。量子色力学を非摂動的に厳密に定義した唯一の理論が格子ゲージ理論である。

物質の基本法則と目される格子ゲージ理論を検証し、物理的予言を行うことは素粒子物理における最も重要な問題の一つである。上に述べた理論の非摂動的性質のために、解析的方法はあまり役立たず、計算機を用いた数値的方法が有用であることが明らかになってきた。量子色力学の2つの基本的な性質、漸近的自由とクォークの閉じ込めとが、理論内に共存することがほぼ確かになったのも、数値計算の結果である。

格子ゲージ理論がクォーク・グルーオンの基礎法則であるからには、原理的にはハドロンのすべての物理量を、理論が含むパラメータ、クォークの質量とゲージ結合定数のみを与えることにより、計算できるはずである。とくに、陽子、パイ中間子などの質量を正しく導出することが、理論の正しさを証明する鍵である。また、素粒子の質量を第一原理から導くことは、それ自体として非常に魅力ある問題である。また単なる「クォーク模型」と量子色力学の違いとして、グルーボールの存在が挙げられる。グルーボールの存在の確認と、その質量の予言と実験との比較は、格子ゲージ理論の試金石となる。更に、宇宙初期における高温、高密度状態におけるクォーク・グルーオン状態の性質を明らかにすることも重要な課題である。このような状態にあっては、クォーク・グルーオンはハドロン状態でなく、クォーク・グルーオンのプラズマ状態になっていると考えられている。この2つの状態の間の相転移点、相転移の次数を明らかにすることは、実験物理においては不可能なことであり、いわゆる数値実験にして初めて可能となるのである。これらの結果は、宇宙物理、重イオン反応などへ極めて重要な情報を提供する。

以上のように、種々の重要な問題が格子ゲージ理論の数値的研究によって初めて解決されるはずである。ところが、現状では、それらの問題に対する数多くの研究にもかかわらず最終的解決には到っていない。その主なる理由は、現在存在するコンピュータの性能が、近年のコンピュータの急速な進歩にもかかわらず、素粒子のこれらの問題を解くのに十分とはいえないからである。素粒子の問題を扱うには、相対論的でなければならず、必然的に4次元系の物理の問題となる。このために、2次元、3次元の問題に比べて、格段の計算の速さと、メモリーの大きさが要求される。

上に述べた困難のため、現有のコンピュータを用いて、各人がそれぞれ種々の工夫をして、格子ゲージ理論を研究している。大きく分けて2つの流れがあり、一つ

は、クォーク・ループの寄与を無視するクエンチ近似中で正しい答を得ようという考えで、他方は、クォーク・ループの寄与も入れる予備的な計算を行うという考えである。後者では、格子として十分大きなサイズの格子の上で計算できないなどの理由により、数値として得られる答に系統的な誤差が含まれるのは避けられないが、種々のアルゴリズムを開発し、第0次近似の答を出そうとしている。もちろん本当にやりたいのは、両者を合わせたもの、すなわち十分大きなサイズの格子上でクォーク・ループの寄与を入れた計算である。

現在の計算機の約10～100倍の速さの計算機として、2つの可能性がある。一つは汎用の計算機で、その速さのものが出来るのを待つことであり、もう一つは、並列型計算機を設計し製作を依頼することである。ところで格子ゲージ理論は近接相互作用であり、一様であり、境界条件も通常周期境界条件、または反周期境界条件を用い、並列計算機に非常に適した問題である。コスト・パフォーマンスの点から考えれば、後者の方がずっと優れている（同じ性能の物が汎用機の約1/10～1/100の価格でできる）。また、本研究の更に先のこと、他の研究分野に対する波及効果などを考えると、並列型が唯一の選択と考えてよい。例えば、現在のパイプライン型のスーパー・コンピュータの1000倍の性能のものを将来製作しようとしたら、並列型しかありえない。

並列型の問題点としては、実際にハードの面で製作可能か、どのようなアーキテクチャを採用するかという点と、それに適したソフトウェアを開発できるかの2点である。この点に関しては、星野等による今までの実績がある。すなわち、世界に先駆け1977年より、マイクロコンピュータをMIMD方式隣接通信方式で結合するPACS計算システム（後にPAXと改名：実験機PACS-9, PACS-32, PAX-128）を開発し、量子色力学のひな型であるU(1)ゲージ模型、ハイゼンベルク・スピン系をはじめ、連立一次方程式、FFTの並列計算など多数の代表的計算を行い、50%～95%に及ぶ高い並列効率を実測した。更に、現在稼働中のPAX-64Jにより、格子ゲージ理論の代表的計算を行い、高い並列効率を実測している。これにより、PAXのアーキテクチャが格子ゲージ理論に適していることが明らかになった。このプロジェクトでは、この基本的アーキテクチャに基づき、1ボードあたりの計算速度と、ボード間の通信速度を早くして、パイ

ブライン型のいわゆるスーパーコンピュータより高速の計算機を製作し、最初に述べた研究を行う。

2. 現状

この研究計画は、大きく分けて二つのステップからなる。第一段階として高並列計算機を製作し、第二段階としてその計算機を用いて1.で述べた研究を行う。62年度、63年度の計画としては、主として第一段階のみで、第二段階は64年度以降となる。従って研究進捗状況としては、主として高並列計算機の製作（ハード、ソフト）に関するもののみとなる。

基本的な考え方としては、与えられた予算、時間、人的資源を考慮して、過去のPAXの研究成果を基に、目的とするQCDモデルのシミュレーションに最適な、QC DPAXの基本設計を行った。更に、その基本設計に基づき、並列計算を分担する演算要素（PU）のアーキテクチャ、HPI（HOST計算機とPUとのインターフェイス）、HOST計算機システム、ソフトウェア（コンパイラ、アセンブラ）の設計を行った。

62年度には、4台のPUからなるQC DPAX-4を製作し、63年度には、288台からなるQC DPAX-288の製作に取り掛かっている。これらの作業と並行して、QCDのシミュレーションをPAX64Jを用いて行い、PAXのアーキテクチャによる並列計算機の上でQCDの計算を効率よく行えることの確認もした。これらのこと及び関連することを、以下に詳しく述べる。

3. QC DPAXの基本設計

1.で述べたように、PAXのアーキテクチャはQCDのモデルによく適合している。モデルのもつ特徴、すなわち、局所性、近接性、境界条件が、PAXのMIMD、近接通信、トーラス構造に対応している。また4次元を2次元写像することは何の問題もなく、ハードウェアの困難さ、PU間の通信のオーバーヘッドなど総合的に判断して、従来と同じ二次元トーラス構造を採用することにした。目的とする大規模なモデルのシミュレーションを行うには、全体の速度が10GFLOPS程度で

あること、メモリ容量がモデルの大きさに見合ったものであることが必要である。

したがって

- (a) 全体の速度 = 演算ユニット (PU) の速度 * 演算ユニットの台数
 全体の容量 = 演算ユニット (PU) の容量 * 演算ユニットの台数
 ができるだけ大きいこと。
- (b) 隣接PU間の通信が演算速度と同程度の速度を持つこと。

が必要となる。

(a)のためには、現在利用できるECL技術によるスーパーコンピュータのプロセッサを採用することは不適當である。なぜなら、それらのプロセッサは最も高速(数10ないし数100MFLOPS)であるが、サイズ、価格、発熱が大きく、限られた予算の範囲内では結果的に実装できる台数が減少し、全体の速度性能が低くなるからである。従って現在のスーパーコンピュータを複数台並列に結合することは、最適ではない。

また、CMOSではあるがマイクロプロセッサ(MPU)の副プロセッサとして市販されている0.1MFLOPS程度の汎用FPUを用いることも、低速すぎて最適ではない。

全体の速度を最高にするのは、CMOS技術による専用浮動小数点演算用プロセッサ(FPU)とメモリを用いたワンボードコンピュータとしてPUを設計することである。現在市販されている専用FPU(実際にはL64133、33.3MFLOPSを採用した)とメモリ(256KbSRAMと1MbDRAM)によりワンボードPUを作成すれば、数10MFLOPSの速度と数MBの容量を実現でき、このPUを数100台(当初計画では576台)結合することにより、最高速度19GFLOPSが実現でき、QCDモデルのシミュレーションの要求に十分答えられる。

ただし、単純に専用FPUとDRAM/SRAMメモリ間を結合すると、制御情報とデータ転送はMPU経由となり、これがネックとなって高速性が実現しない。これを解決するのは、MPUを経ないで制御とデータの転送をすることである。直接メモリ参照方式(DMA:Direct Memory Access)がこれであり、データは一括してFPUへDRAM/SRAMメモリから流される。これはユーザからみれば、データは一塊のベクトルとして処理をうけることになり、ワンボードPUは必然的にベクトルコンピュータとなる。

Fermionのシミュレーションのためには頻繁に大規模な疎行列一次連立方程式を解く必要がある。このアルゴリズムとして有効な前処理付き反復法の並列処理は、各PU中の処理が一次漸化式になり、このベクトル処理はいわゆる演算パイプライン処理では特別な工夫が必要でソフトウェアが複雑になり、スピードも出ない。すなわちFPUはスカラープロセッサであることが望ましい。採用したL64133は現在市場に現れている最高速のスカラーFPUである。

FPUの計算速度にあわせてデータをメモリから供給しなくてはならない。そのためのメモリは高速メモリを用いるがこれは高価であるので、通常のスーパーコンピュータでは、低速のDRAM中に大量のデータを蓄えておき、ベクトル処理を必要とするデータのみをその都度高速のSRAMによるバッファメモリへ移し、それからパイプライン演算器によって処理を行っている。このためこのバッファメモリを複雑に制御する、実行形式プログラムを生成するコンパイラが必要となる。

このコンパイラの開発は、最終的なユーザのプログラミングの能率を向上させるために、必須のものであるが、その開発は容易ではない。全体のシステム設計は、ハードウェアのみならずこのソフトウェアの作成も考慮して最適に行わなければならない。ここでは、すべてのデータを単一バンクのSRAMの高速メモリにおき、バッファメモリの制御を不必要にした。それはコンパイラの開発を容易にし、開発期間の短縮とソフトウェア開発費の低減をもたらしたが、一方メモリコストの増大を結果した。

(b)は、並列計算機のオーバヘッドとして最も深刻なPU間のデータ転送を低く保つために、転送速度を計算速度と同程度にすることである。QC DPAXでは、内部の32ビットバスをそのままPU間のデータ転送バスとして延長している。PAXの基本的アーキテクチャがMIMDであることとあいまって、データ転送やそれにとともなうPU間同期が全体の性能のネックにならない。

4. 各論

4.1 P U (演算装置)

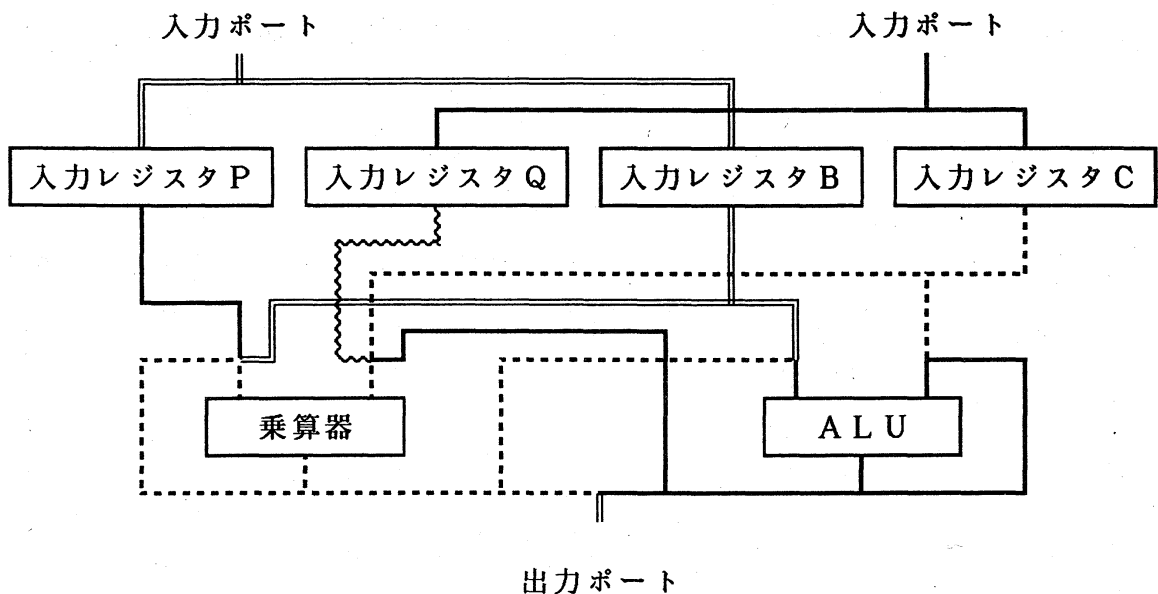
P Uに必要な機能の主なものは、高速の浮動小数点演算と近接通信である。また、必要なだけのデータとプログラムを蓄えるメモリが必要である。Q C D計算を行うため最低限必要なものとして、P Uは主として以下のものから構成する。

- C P U : M C 6 8 0 2 0 (2 5 . 0 0 M H z)
P U内の制御、近接通信を行う。
- F P U : L 6 4 1 3 2 B (1 2 . 5 0 M H z) (Q C D P A X - 2 8 8
ではL 6 4 1 3 3 (1 6 . 6 7 M H z) を使用予定)
浮動小数点演算を行う。
- F P U C : F P Uコントローラ。
- D R A M : 4 M B、1 0 0 n s、プログラム格納用。
- S R A M : 2 M B、3 5 n s、演算データ格納用。
- C M : 2 P o r t R A M 8 K B × 3、7 0 n s、
近接通信用2個、ホスト-P U間通信用1個。
- W C S : S R A M、8 K B、2 5 n s、F P U C用プログラム格納用。
- L U T : R O M、5 1 2 B × 2、逆数、平方根関数の初期値格納用。
- タイマー : 8 0 n s単位、4 0 b i t

高速の浮動小数点演算を行うため、浮動小数点演算L S I (F P U) を採用することにした。浮動小数点演算には、個々のデータに対し異なる演算を行う場合(以下スカラ演算と呼ぶ)と、ベクトル、行列などのデータの塊に対しほぼ同じ演算を行う場合(以下ベクトル演算と呼ぶ)がある。これら2つの場合について、F P Uの高速性をできるだけ活かすよう、専用のF P Uコントローラ(F P U C)を開発した。以下、F P U、F P U C、近接通信について述べる。

(a) FPU

現時点で市販されている最も高速の浮動小数点演算LSIとしてLSIロジック社のL64133を採用することにした。L64133は、4つの入力レジスタ、乗算器、加減算器、2つの出力レジスタをもっており、60nsで乗算と加減算の2つの演算を同時に行うことが出来る。



L64132B (L64133) 構成図

(b) FPUC (FPUコントローラ)

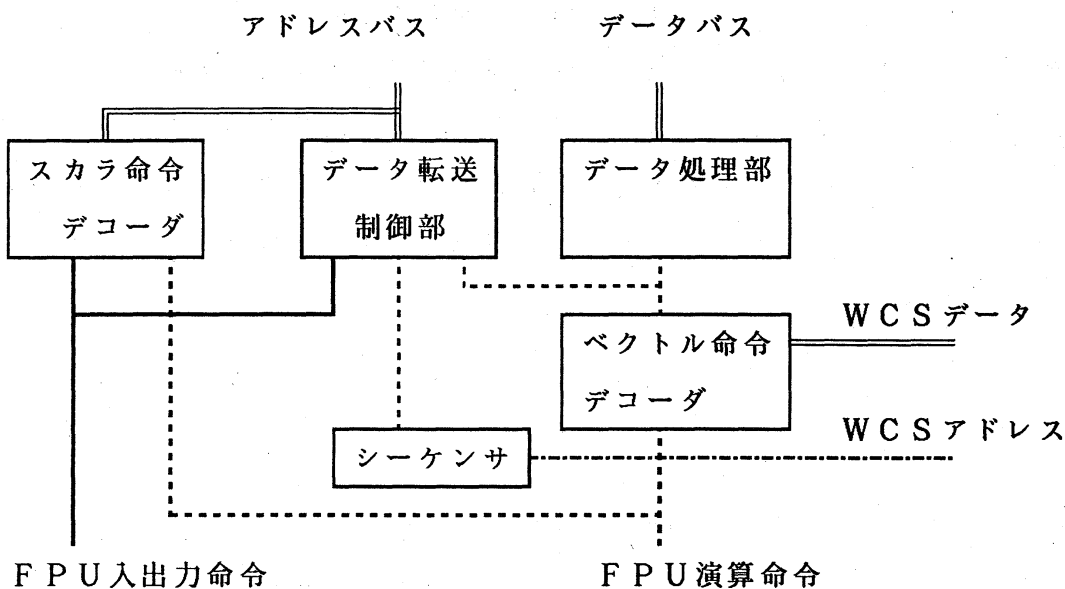
FPUは単独ではデータの入出力もできず、また、加減乗算程度の命令しか持っていない。CPUの命令に応じて、データの入出力や、除算、基本関数演算及び条件分岐が可能なベクトル演算処理を可能とするため、ゲートアレイにより、FPUのコントローラを開発することにした。ゲートアレイを用いて作成することにより小型高速化も行える。

FPUCは、スカラ演算命令デコーダ、データ転送制御部、シーケンサ、データ処理部から成る。

スカラ演算デコーダは、CPUの制御下でスカラ演算を行うときに、CPUが1ステップで出力した命令を、FPUの2ステップ分の命令に分けて入力する。CP

Uの1命令の実行時間 ($> 120 \text{ ns}$) 中にFPUの2ステップの命令 (データの入/出力 60 ns と演算 60 ns) を行うことができる。

データ転送制御部、シーケンサ、データ処理部は、除算、基本関数演算及び条件分岐が可能なベクトル演算処理をおこなうためのものである。データ転送制御部は、メモリ中のデータのアドレスを計算しつつ必要なデータをメモリとFPUの間で転送する。シーケンサは、条件分岐、サブルーチンの制御をおこなう。データ処理部ではFPUでは出来ないがQCD計算に必要なデータの正負判定、零か否かの判定、指数部の設定、仮数部の取り出し、排他的論理和の計算を行う。FPUCにより実行されるサブルーチンは、FPUCによって制御されるベクトル演算からも、CPUによって実行されるスカラ演算からも呼び出すことができる。



F P U C 構成図

(c) 近接通信

隣接するPU間の通信は、2ポートRAMにCPUがデータを書き込み、隣接するPUのCPUがこれを読み出すことにより行う。近接通信を高速に行うには、MC68020の高速性を利用することとし、DMAコントローラは用いなかった。高速性を保つため、データのバス幅は、32bitとした。PUのサイズを大きくしないため、容量はひとつの通信メモリあたり、8KBとした。

FPUによりベクトル演算が行われている間にCPUによりデータの転送が行えるが、これをやりやすくするため、隣接PUからの割り込みを可能とした。

4.2 HPI

HPIは、ホスト計算機とPUアレイ間でデータやプログラムを転送するため、バスを接続するインタフェースである。

PUにおける並列処理の制御は、基本的にはホスト計算機のユーザプログラムによっておこなわれる。しかし、ホスト計算機では、プログラム開発など他のジョブも行われるので、並列処理の制御の一切をホスト計算機が行うと負荷が過重になる。そこで、並列処理の制御の内、負荷は大きいが手続きは定型的であるものを、HPIをインテリジェント化して、HPIに行わせることにした。

HPIでは、PUの同期の処理、並列処理の終了、中断などの検出、PUからの画像データのディスプレイへの出力を行う。また、これらの内でも、PUの同期の処理、並列処理の終了、中断の検出は負荷が大きく、従来のPAXの経験から処理が定型化しているので、専用の同期処理回路をHPI上に作製し、これを用いて行うことにした。

4.3 ホスト計算機

ホスト計算機としては、SUN3/260を用いる。

HPIは、このホスト計算機のVMEバスに直接装着する。

SUN3/260の基本構成では、ディスク容量は260MBであるが、QCD計算を中断する際、途中結果を保存し、続きから計算を再開できるよう1GBを増設する。計算結果、データを保存するため、磁気テープ装置を持つ。

4.4 ソフトウェア

(a) コンパイラ

プログラム開発を効率よく行うには、コンパイラが不可欠である。QC DPAX

では、並列処理をおこない、また、FPUを用いて浮動小数点演算を行うので、既存のMC68020用のコンパイラは使用できない。そこで、新しくコンパイラを作成する。コンパイラの仕様は、PAX-64Jにおける方法に少し改良を加え、次のようにする。

1. 言語の名前はpscとする。
2. ノード記述（個々のPUの動作を記述する）言語とする。
3. ホスト計算機とPU間の通信、PU間の通信には共有変数を用いる。
4. 共有変数の宣言のため、領域指定子を設ける。
5. 同期などの並列処理のための関数を設ける。
6. 出力は、MC68020のアセンブリ言語、および浮動小数点演算機構アセンブリ言語とする。
7. 変数の型のひとつとして複素数を設ける。
8. ベクトル演算の記述のため、vfor文を設ける。

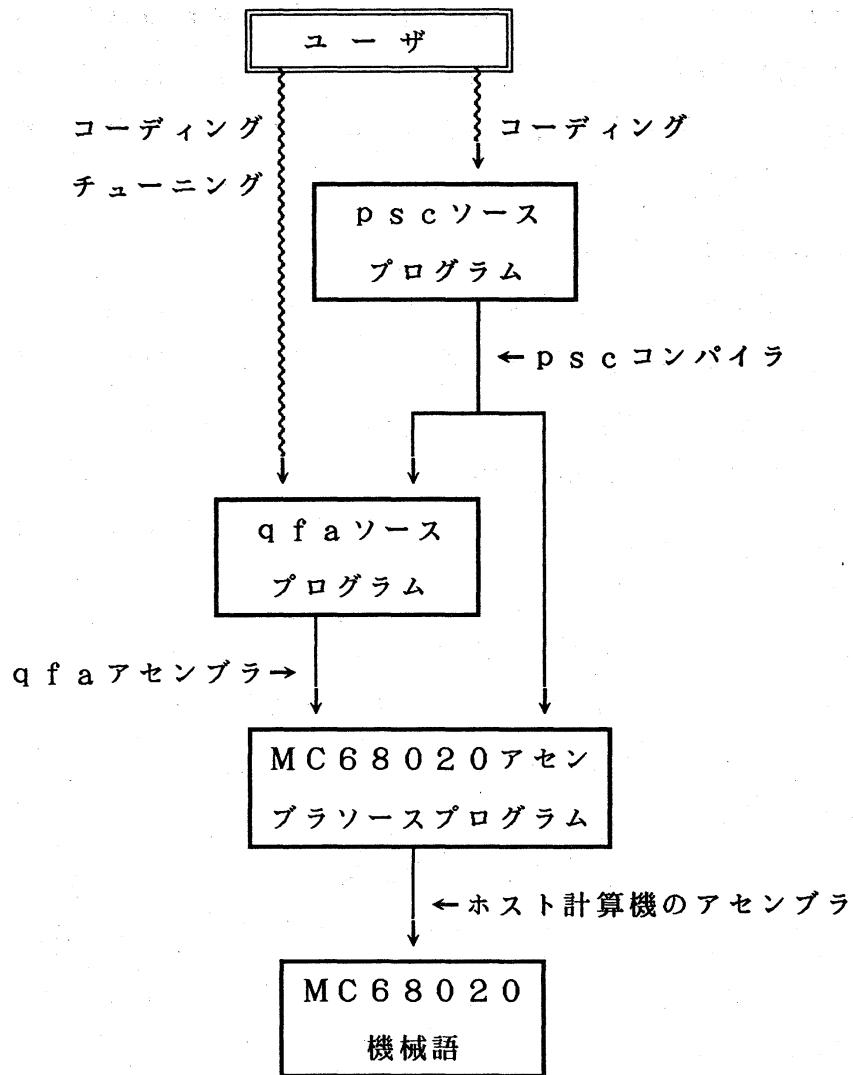
(b) 浮動小数点演算機構アセンブラ

従来、スーパーコンピュータなどでは高速計算を行うため、チューニングと称する、高級言語（FORTRANなど）で作成したプログラムをベクトル演算向きに改良したり、コンパイル後更にアセンブリ言語のプログラムを改良することが行われている。

このチューニングは、非常に効果的な場合があり、10倍、20倍の速度向上が得られることがある。本研究においては、積極的にチューニングを行うことにし、特にユーザがチューニングし易いよう、プログラムしやすいアセンブリ言語を開発する。

(c) PU用プログラム開発の手順

pscコンパイラ、qfaアセンブラを用いて、PU用プログラムを開発する手順を次ページの図に示す。



PU用プログラムの作成手順

5. QCDPAX-4の製作と評価

昨年度は、まず試験的に4台のPUとHPI、ならびにホスト計算機からPUやHPIを制御するソフトウェアを作製した。さらに今年度は、この4台のプロトタイプを使って、コンパイラ、アセンブラを開発しながらハードウェアの性能を測定して、回路方式、実装など、ハードウェアの設計の正しさを確認した。

pscコンパイラ、qfaアセンブラ、プリプロセッサ、その他システムプログ

ラムは、ホスト計算機上で開発され、QCDPAX-4で試用してデバッグを行っているが、QCDPAX-288にもそのまま使用できる。

5.1 PUの性能評価（QCDPAX-4で測定）

(a) 浮動小数点演算速度

PU上で演算を行い、演算速度を実測した。

スカラ演算では、

$a = b + c$ の計算時間は、 $1.26 \mu s$ 、

$a[i] = b[i] + c[i]$ 1要素の計算時間は、 $4.42 \mu s$

であった。

$a[i] = b[i] + c[i]$ のベクトル演算では、おおむね、

1000要素の計算時間は、 $249120 ns$ 、

4000要素の計算時間は、 $969120 ns$ 、

8000要素の計算時間は、 $1929120 ns$ 、であった。

従って、ベクトル長が無限大のときの速度は、約 $4.17 MFLOPS$ （ベクトル長が1000のとき約 $4 MFLOPS$ ）、半性能長は38となる。

また、以下に挙げる基本関数をマイクロプログラムにより作成した。その速度を示す。

基本関数計算時間（実測値）

$1/x$	$0.88 \mu sec$
$1/\sqrt{x}$	$1.20 \mu sec$
$\sin(x)$	$2.12 \mu sec$
$\cos(x)$	$2.04 \mu sec$
$\arctan(x)$	$2.44 \mu sec$
$\exp(x)$	$1.40 \mu sec$
$\ln(x)$	$2.08 \mu sec$ （予測値）
rnd	$6.25 \text{個}/\mu sec$ （予測値）

QCDPAX-4では、80nsのFPUを用いている。QCDPAX-288では60nsのFPUを用いる予定であるので、浮動小数点演算の速度は、上記の実測値より若干向上する予定である。

(b) PU間のデータ転送速度

PU間のデータ転送速度は、ひとつのPUがSRAM上の2000個のデータを隣接通信用メモリに書き、さらに隣のPUが隣接通信用メモリからSRAM上にコピーする、というプログラムを、MC68020のアセンブラにより作成し実測した。転送速度は、約2.7MB/secであった。また、一度に送るデータの個数の単位を10個にすると約4MB/secまで上がることを確かめた。

なお、QCDPAX-4のPUの回路設計には、アンリツ株式会社の協力を得た。PUのプリント基板の設計、QCDPAX-4の製作はアンリツ株式会社により行われた。

6. 並列計算機ための自動化ソフトウェアの研究

並列計算機の実用化上の最大の技術的課題は、ソフトウェアである。ソフトウェアは、ユーザが意識し表現しようとするモデルを、プログラミング言語とオペレーティングシステムの助けをかりて、ハードウェア（計算機アーキテクチャとして与えられた面に限るが）上で、実現し計算を実行するものである。もし計算機アーキテクチャとユーザモデルの構造が相当違っているときは、ソフトウェアすなわちプログラムの負担は、非常に大きくなる。

PAXの構造は、空間的に分布している場や連続体をそのまま空間に直接射像したようになっており、プログラムの意識しているモデルの構造と計算機の構造は相似である。この点、最近出現している種々雑多の構造をもつ並列計算機に比べて、プログラミングは著しく容易である。

しかし、PUのアレイの次元（2次元）よりモデルの空間の次元（QCDモデルでは4次元）が大きいときや、PU台数より空間の格子点数が多いとき、プログラミングは煩雑になり、誤りを生じやすく、プログラミングとデバッグの作業能率が低下する。

このため、PAXでは以前より数学的な表現で目的とするモデルを入力すれば、PUにロードすべきプログラムを生成する、自動プログラミングの研究が行われてきたが、これらのソフトウェアの研究は、成果がすぐには得られないものであり、当面の目的とするQCDモデルのシミュレーションには、人間による並列処理のプログラミングを行わざるを得ない。

しかし、モデルの特徴やPUの構造を特定することにより、限定した仕様のプログラミングの自動化は可能である。このDISTRANと呼ばれるソフトウェアの開発はそれを目指している。

このDISTRANでは、自然現象の連続表現を離散表現に変換するDISTRAN Translatorと、その離散表現からPUとHOSTのプログラムを生成するDISTRAN Generatorよりなる。ユーザの書く連続表現（微分方程式）の記述法の改良、数値解と自然の遊離の原因とその対策、解の表示方法などのユーザインターフェイスなどが研究課題である。

とくにQCDのモデルの並列プログラミングのためには、このGeneratorにより、煩雑な4次元空間の離散表現を、2次元PUアレイの空間に射像したときのPUのプログラムの自動生成を行うことが重要である。

すでに2次元および3次元空間に対するDISTRANの原型はできあがっており、現在これを4次元空間への拡張が進行中である。〔参考文献、上田將也、窪田誠、川合敏雄、情報処理学会昭和63年春年会3M-9〕

上記の研究課題を取り込んで、実用版の完成を来年度に目指している。

7. 世界の情勢

QCD PAXと似たプロジェクトは、アメリカではコロンビア大、IBM、FN

AL (Fermi National Accelerator Laboratory), ヨーロッパでは、イタリアのApeグループ、イギリスのEdinburgh大で行われている。各々のプロジェクトの概略は、次の通り。

1) コロンビア大: N. Christ教授を中心に、T. D. Lee教授のサポートによって推進されている。現在二号機まで完成しており、一号機はpeak speedで約200Mflops, 二号機は約1Gflopsである。一号機、二号機を用いてクエンチ近似での有限温度転移の研究を主として行ってきた。素粒子物理専用の並列計算機として、最初に有意義な物理結果を導いた。現在三号機を製作中である(来年春完成予定?)。PUの台数は256台でpeak speedは16Gflopsである。この計算機は規模、速度、構成法などすべての点において、QCDPAXとよく似ている。異なる点のうち、1つの重要な点は、QCDPAXではPU間のdata転送はasynchronousに行えるが、コロンビア大の計算機ではsynchronousにしかできない点である。またコロンビア大ではコンパイラを開発しなかったので、プログラムを書くのに、非常に手間取る。このやり方は失敗だったと思われる。

2) IBM: GF11と呼ばれ、素粒子物理専用並列計算機の計画としては一番古く(1985)発表された。しかしまだ完成していない。(今年中に完成の予定?) その理由の1つはアーキテクチャがSIMDで、巨大なswitch boardを用いるが、それを製作するのに時間がかかったと思われる。D. Weingartenを中心にソフトウェアを製作しており、良いコンパイラを作ったという。彼は有能な研究者なので、機械が完成した暁には、良い結果をだしてくると予想される。

3) イタリアのApe: 4 boardsの約250Mflopsの機械は2台稼動中。INFN (Istituto Nazionale di Fisica Nucleare)の会長のCabibbo, 物性物理から素粒子物理まで幅広い活躍をしているParisiなどがApeのグループに含まれている。有意義な物理結果を出しつつある。コロンビア大よりだいぶ遅れてスタートしたのに、す

で物理的結果において、コロンビア大に匹敵し、またはそれ以上のものを出していることは、1つはコンパイラなどソフトウェアの開発の重要性を示唆している（Apeはoptimize付きのコンパイラをParisi等を中心にして開発した）。現在16ボードの約1Gflopsの機械を完成中。

4) FNAL: ACPMAPS (Advanced Computer Program Multi-Array Processor System)。最初実験のデータ解析用に開発したACPを理論計算機に改造したもの。Peak speedは5Gflops。(今年中に完成? メモリチップの入手困難のため遅れている)。FNALのACPからかなり大勢の人が参加しているために、hardware, softwareとも完成度は高そうである。ただしアーキテクチャがcluster構造になっており、cluster間の通信の為どの程度オーバーヘッドがあるかなど問題点がある。

5) Edinburgh: T800 transputerを用いてpeak speedは3Gflopsの予定。完成時期などは不詳。

以上から分かるように、上の5つのプロジェクトがQCDPAXと同程度か、やや小規模な計画であり、今年から来年にかけて完成をめざしている。hardwareのアーキテクチャ及びsoftwareの開発共に重要な因子であることが上の例から分かる。この点に関して、QCDPAXはPAXシリーズの5代目にあたり、1代目は上の5グループよりずっと早く完成しており、PAXの評価は国際的に高い。

ところで科学計算用の高並列計算機がすべてを素粒子研究を主たる目的としているのはなぜだろうか? 1つは素粒子物理研究者の要求度が高いからであろう。しかしこれは他の分野の研究者でも同じはずではないだろうか? 素粒子物理の場合、最終目標は、基本法則の確立、基本粒子の確立、未知の粒子の予言など、量だけの問題でなく、質の問題である。この目標が、コンピューターの能力があるレベルを越えれば達成できる為に、要求度が一段と高い為であろう。

8. 日本の将来計画

イタリアのApeグループは将来計画として1989年に始まる5年計画、100 Gflopsを超すApe100を発表した。INFNでプロジェクトの調査、または準備は認められたそうである。これからは、加速器が素粒子実験物理に不可欠だったように、高並列計算機は素粒子計算物理学に不可欠となっていくであろう。素粒子理論にはビッグス粒子は”素”なのか否か、その質量は？ また、クォークの世代の問題など、摂動論では論ぜられない重要な問題がまだまだある。これらの問題の解決の為に、斬新な理論的ideaが必要なのはいうまでもないが、そのideaを確かめるには計算物理学に頼らざるをえない。次の日本の計画を早急にたてる必要がある。