GRAPE-DR 計画

国立天文台
理論研究部/天文シミュレーションプロジェクト
牧野淳一郎

今日の話の構成

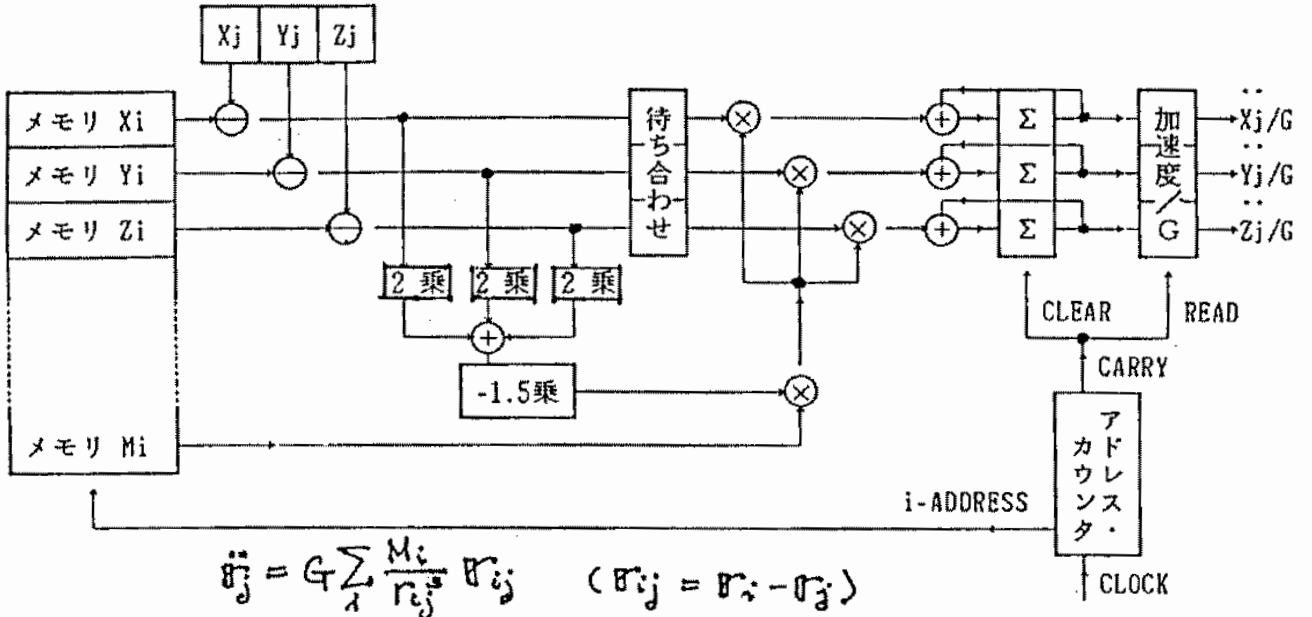
- これまでの GRAPE
- GRAPE-DR の考え方
- 開発状況
- まとめ

GRAPE の考え方

- 重力多体問題 (強結合プラズマ、分子動力学): 粒子間相互作用の計算が計算量のほとんど全部
- 効率のよい計算法 (Barnes-Hut tree, FMM, Particle-Mesh Ewald(PPPM) ...): 粒子間相互作用の計算を速くするだけでかなり加速できる
- そこだけ速くする電子回路を作る (「計算機」というようなものではない)

近田提案

1988年、天文・天体物理夏の学校



$$\ddot{\vec{r}}_j = G\sum_i \frac{M_i}{r_{ij}^3}\vec{r}_{ij} \qquad (\vec{r}_{ij} = \vec{r}_i - \vec{r}_j)$$

+，－，×，２乗は１operation，　-1.5乗は多項式近似でやるとして10operation 位に相当する．
総計24operation．
各operation の後にはレジスタがあって，全体がpipelineになっているものとする．
「待ち合わせ」は２乗してMと掛け算する間の時間ズレを補正するためのFIFO(First-In First-Out memory)．
「Σ」は足し込み用のレジスタ．Ｎ回足した後結果を右のレジスタに転送する．

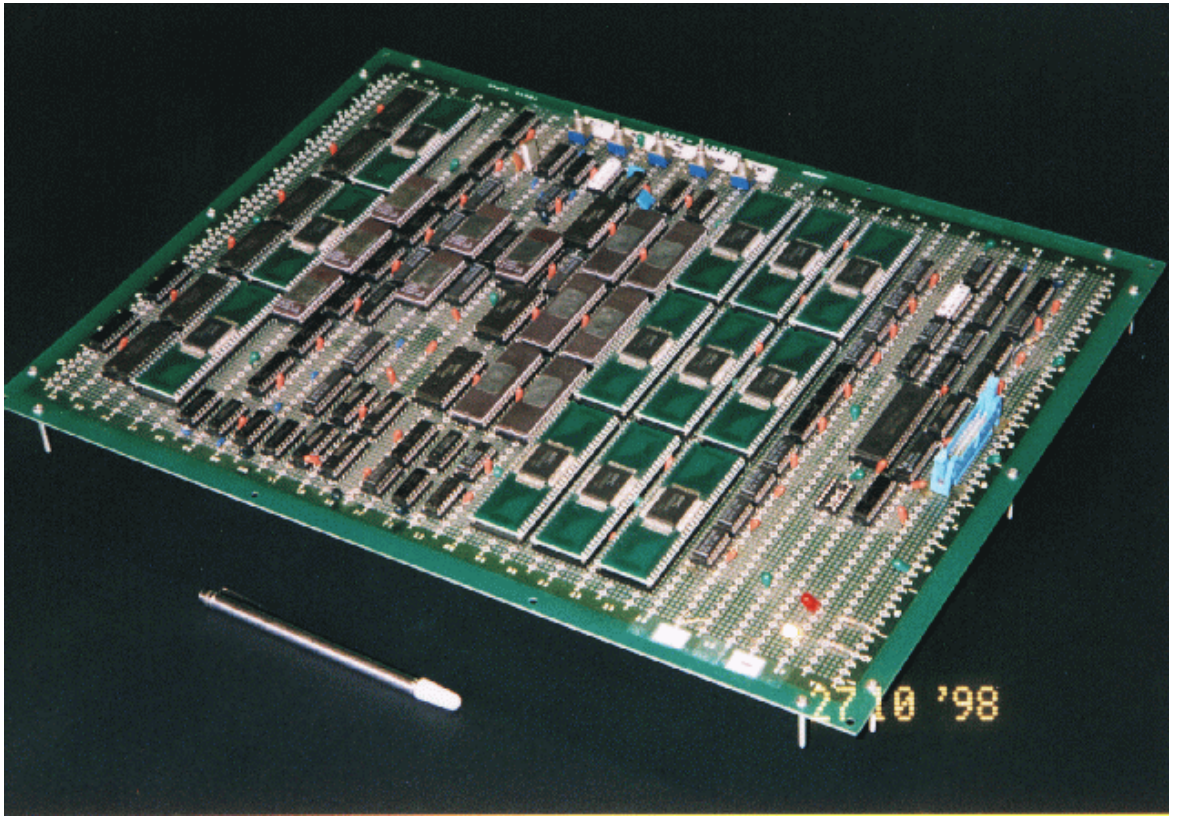
図２．Ｎ体問題のj-体に働く重力加速度を計算する回路の概念図．

近田さんによる見積もり

- 32 ビット固定小数点
- IC 200 個
- 体積 $0.1\mathrm{m}^3$
- コスト 400万

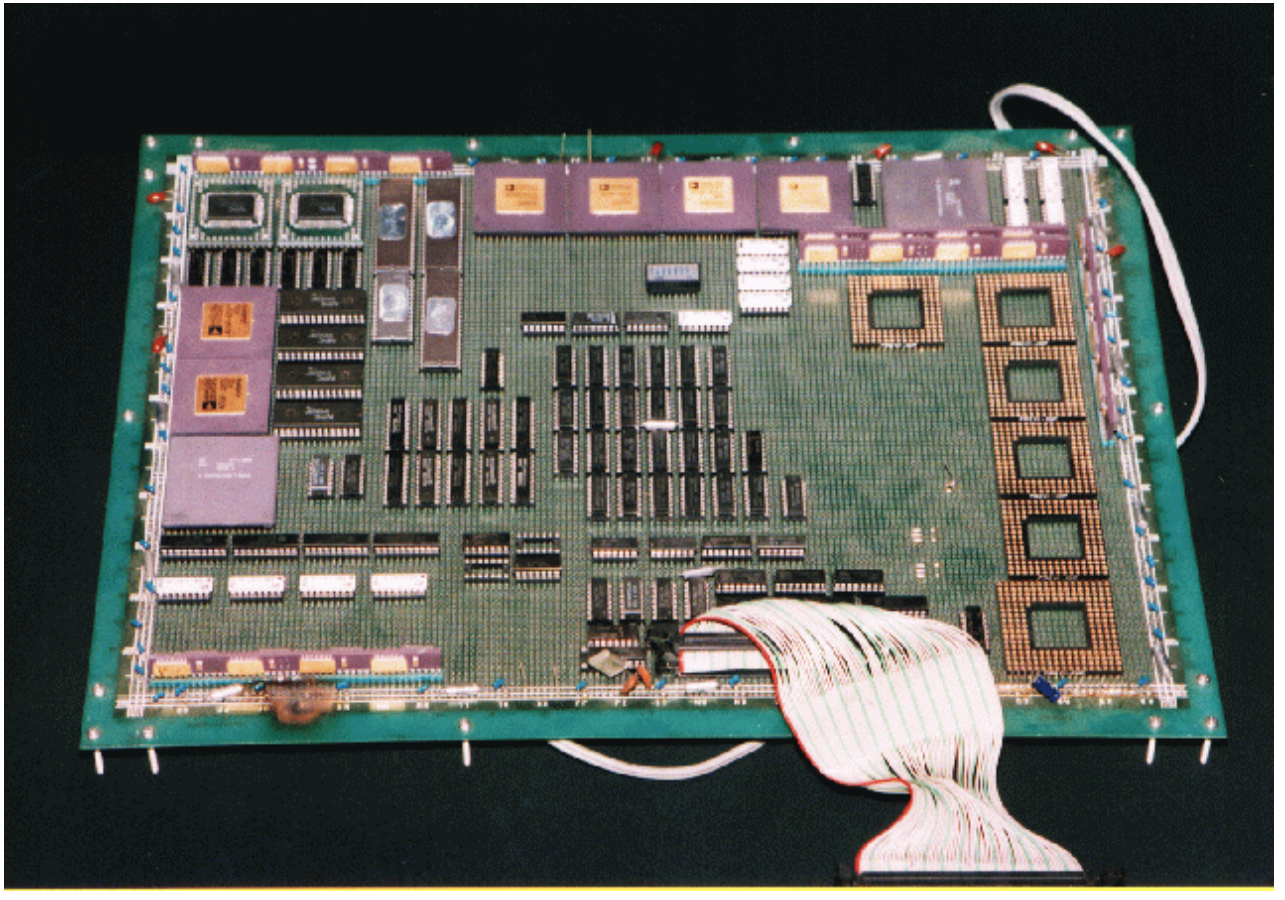
「但し、近田電子製作所の見積もりは甘いという声もあることを付け加えます」

GRAPE-1(1989)



演算毎に語長指定。固定 16–**対数** 8–**固定** 32–**固定** 48
240Mflops **相当**

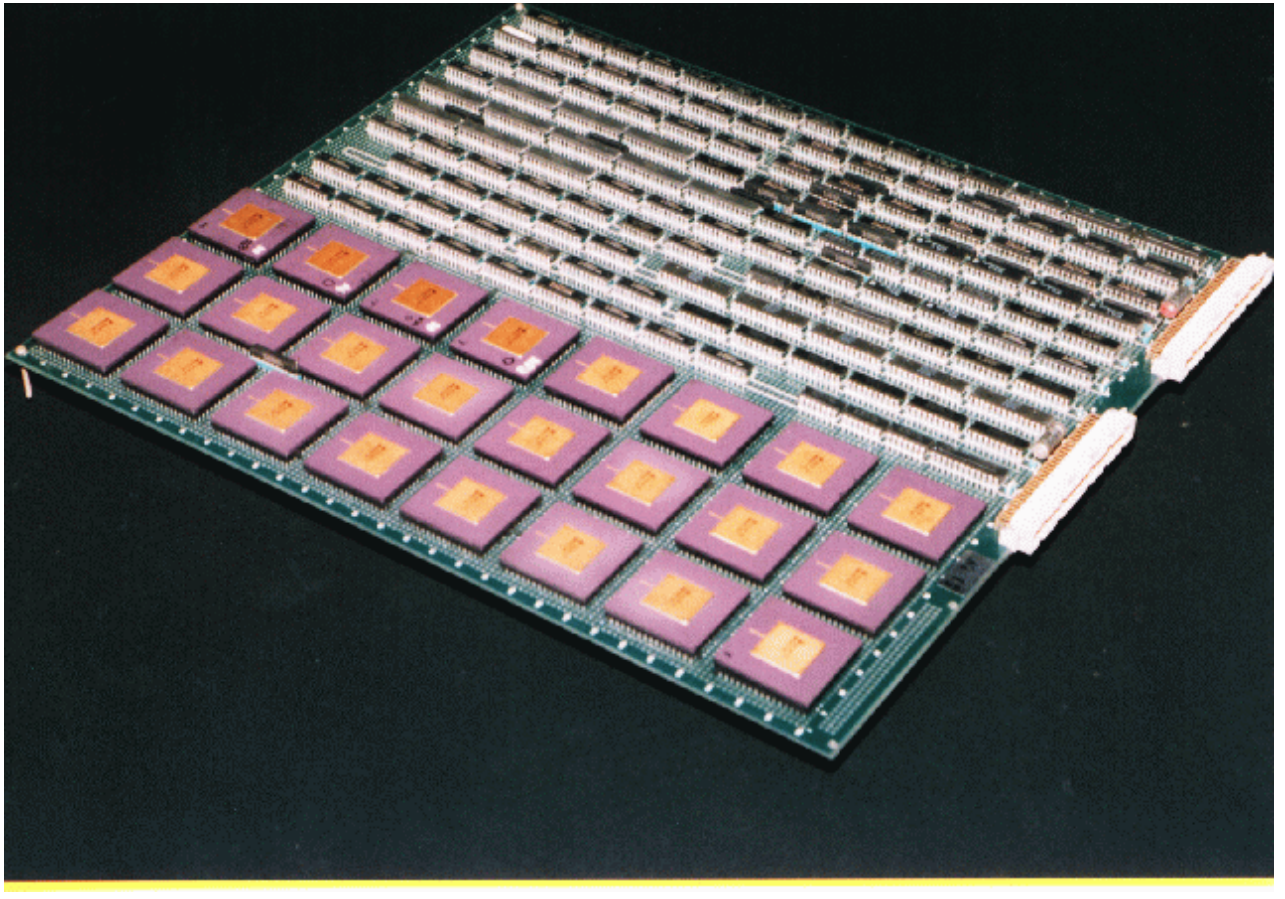
GRAPE-2(1990)



8ビット演算とかは止めて普通に浮動小数点演算 (倍精度は最初と最後だけ)

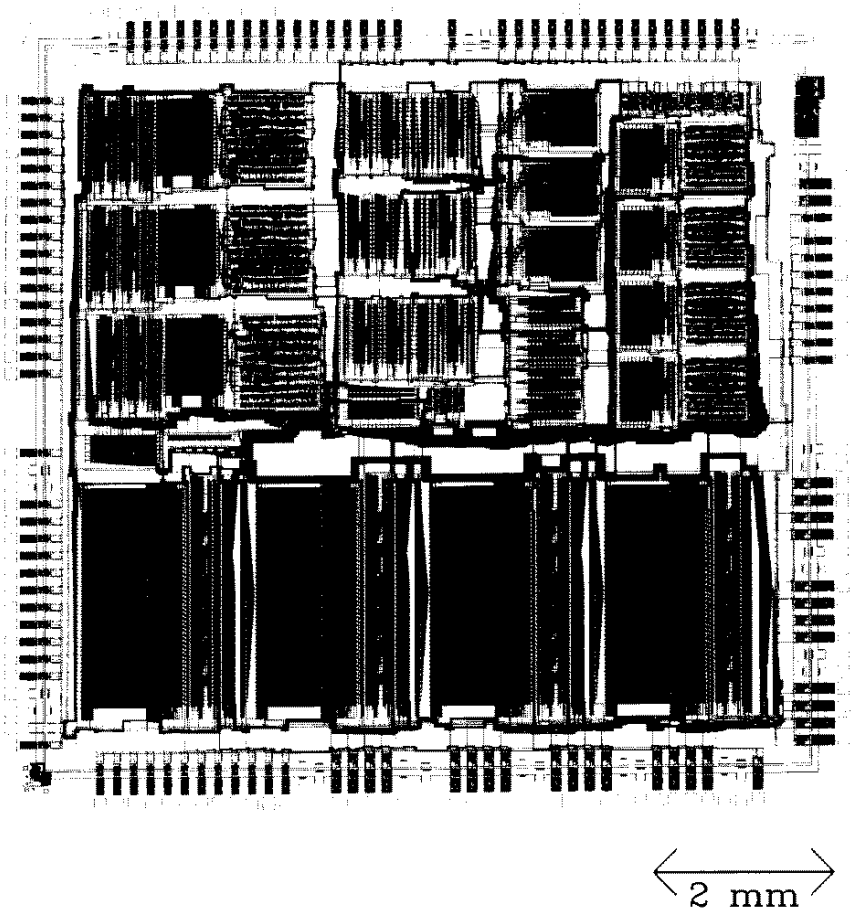
40Mflops

GRAPE-3(1991)



カスタムチップ 24個1ボード、 10MHz 動作、 7.2Gflops

GRAPE-3 チップ

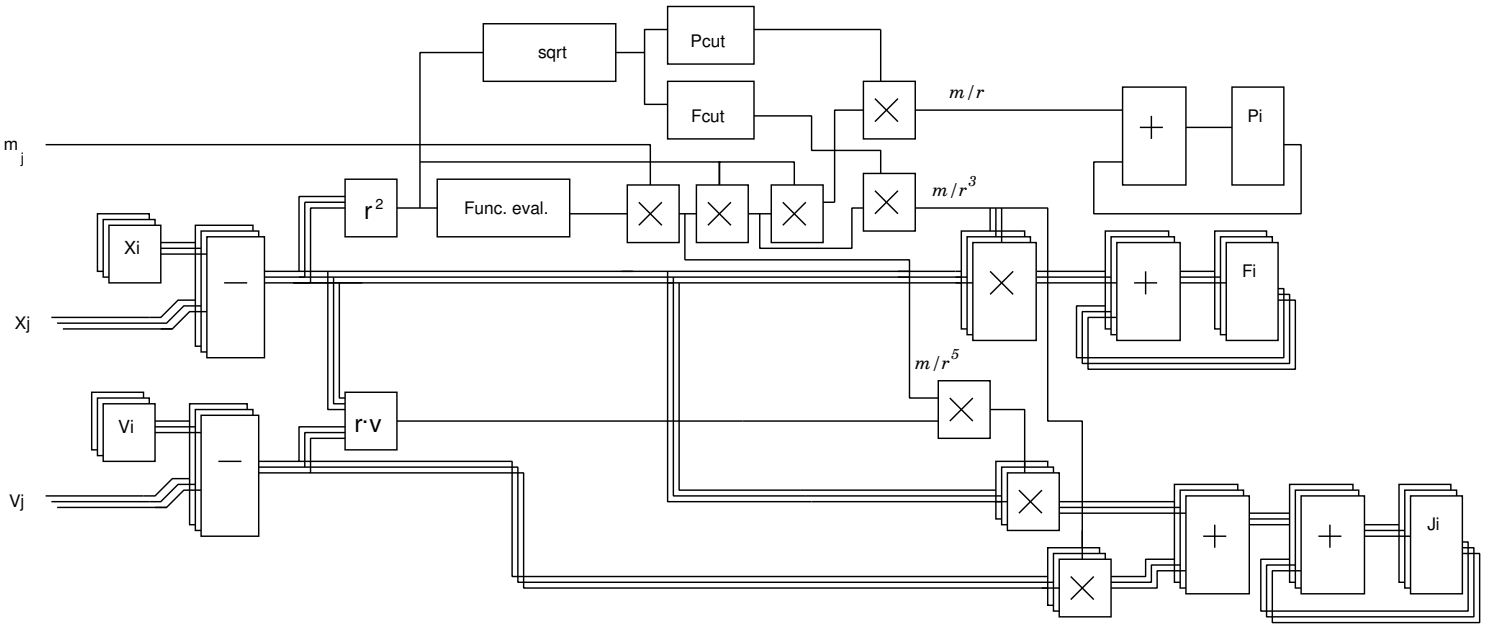


- 1μm プロセス
- 11万トランジスタ
- 20 MHz クロック動作
- 600 Mflops 相当

GRAPE-4(1995)

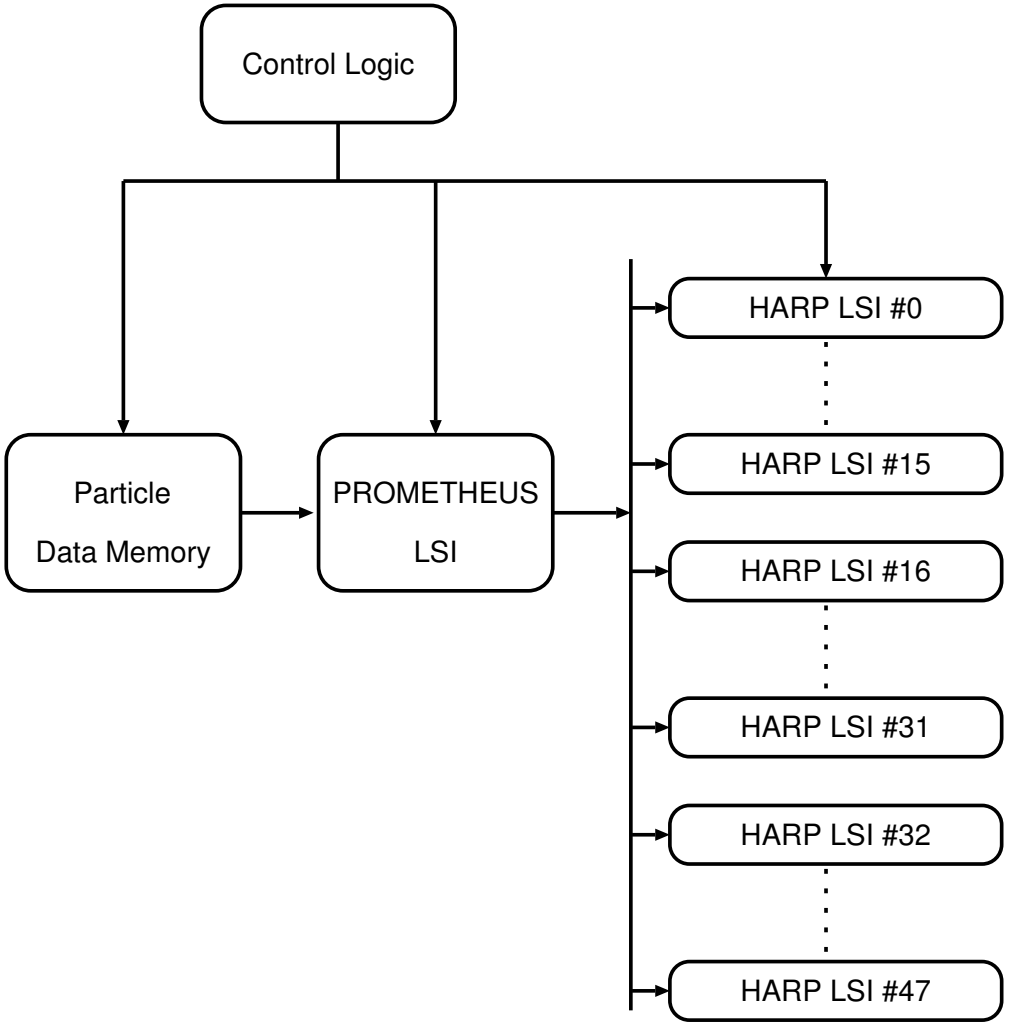
トータル 1792 チップ、 1.1 Tflops

GRAPE-4 パイプライン



1μm プロセス、10万ゲート(40万トランジスタ)、640Mflops

GRAPE-4 演算ボード

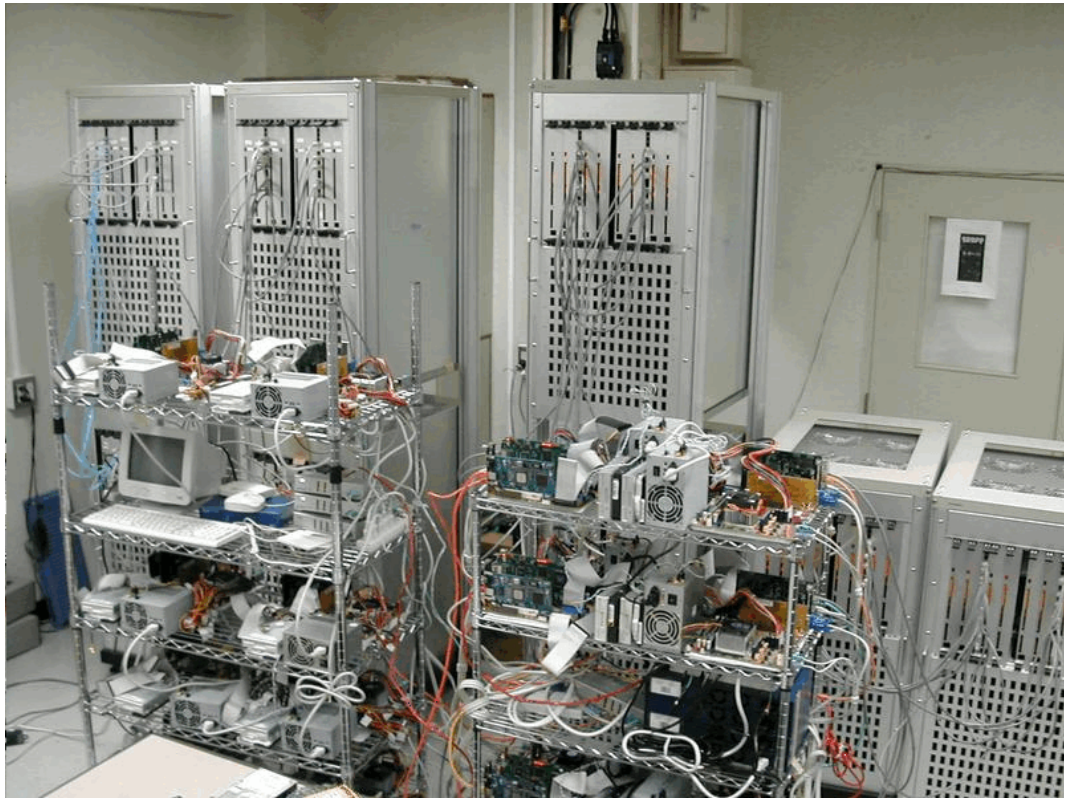


多数の パイプライン LSI がメモリユニットを共有

↓

ボードが単純になり、集積度をあげられる

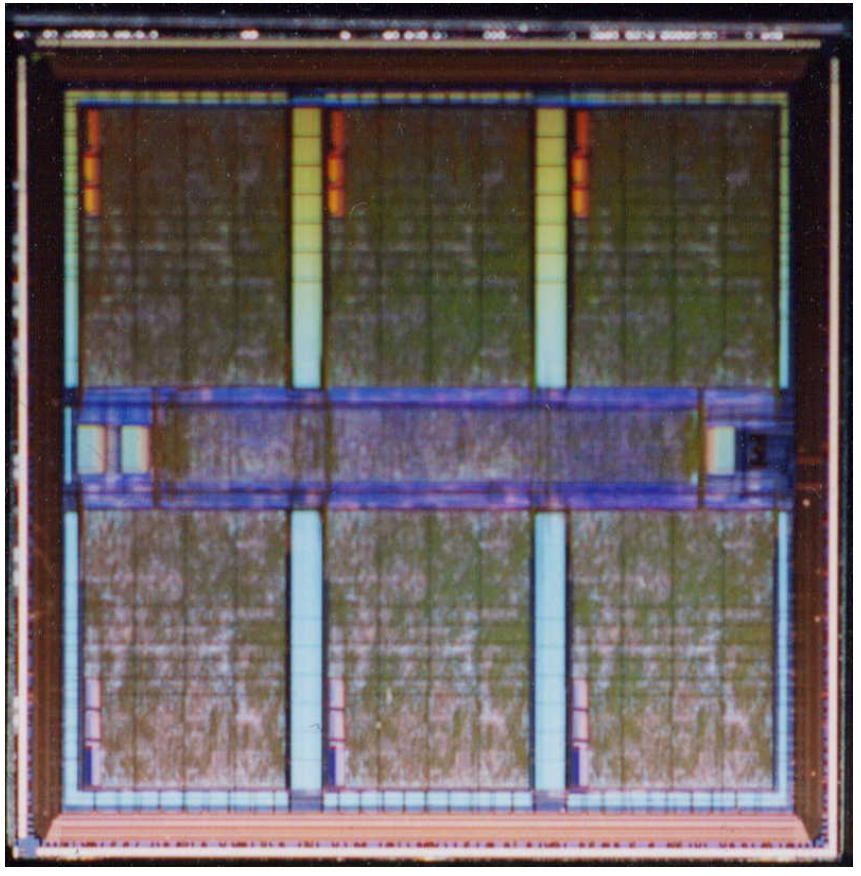
GRAPE-6(2002)



2002年現在の 64 Tflops システム

4 ブロック
16 ホスト
64 プロセッサボード

パイプライン LSI



- 0.25 μm ルール
  (東芝　TC-240，　1.8M ゲート)
- 90 MHz 動作
- 6 パイプラインを集積
- チップあたり 31 Gflops

2006 年のマイクロプロセッサと比べてみる

	GRAPE-6	Athlon FX-62
デザインルール	250nm	90nm
動作クロック	90MHz	2.8Gflops
ピーク性能	32.4Gflops	11.2Gflops
消費電力	10W	95W
1W あたり性能	3.24Gflops	0.12 Gflops

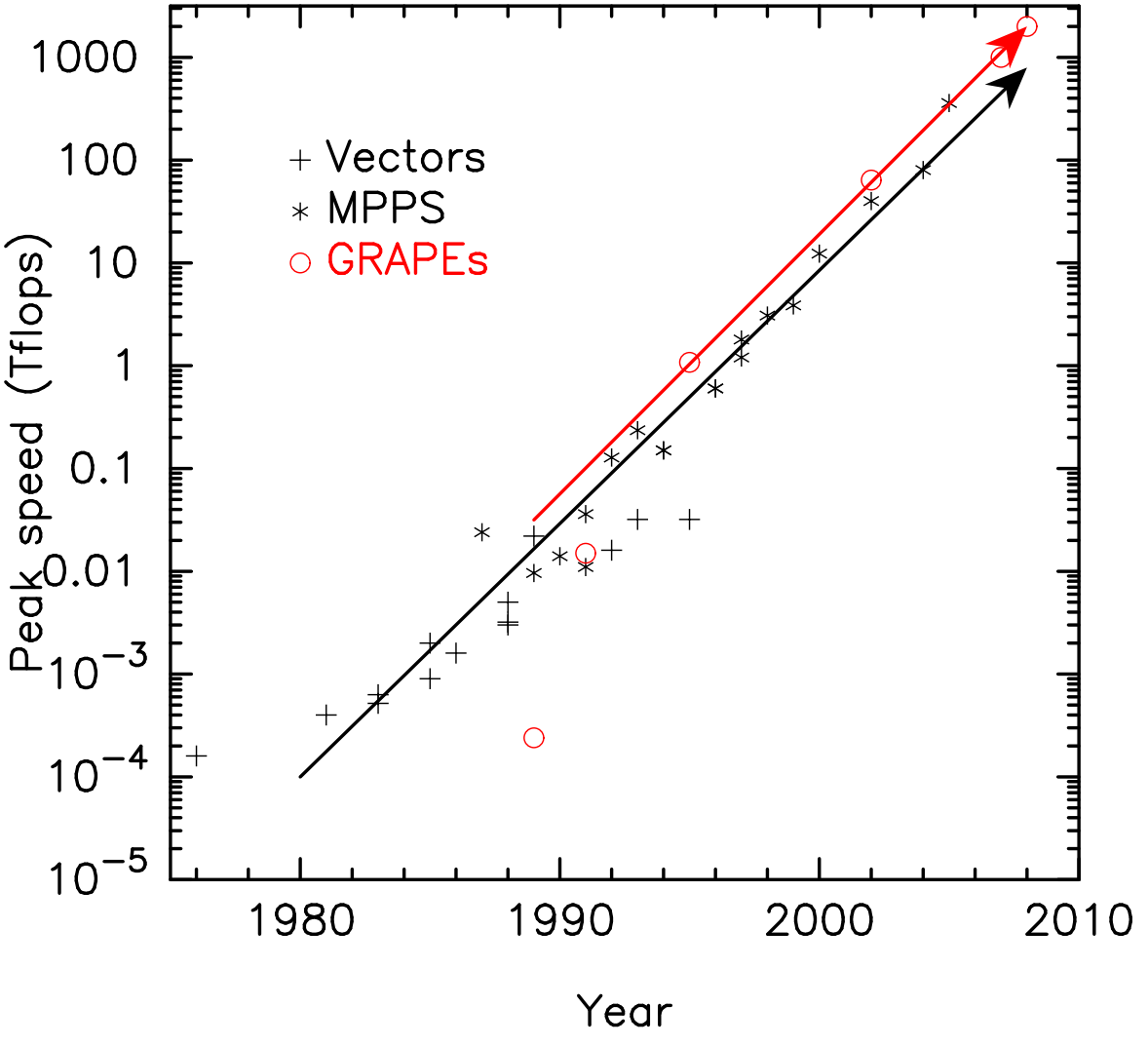
他の GRAPE 型機械

- 1991 GRAPE-1A : GRAPE-1 の細かい改良。設計: 福重
- 1992 GRAPE-2A : 最初の MD 用 GRAPE 設計:伊藤・福重
- 1992 HARP-1: 小久保君設計。 Hermite 公式用
- 1993 GRAPE-3A: 商業版
- 1996 MD-GRAPE: 最初の MD 用 GRAPE チップ チップ設計:泰地
- 2001 MDM (MDG2): 理研・戎崎グループによる。 75T
- 2006 PE (MDG3): 理研・泰地グループによる。 1P
- 1992- MD-Engine: 富士ゼロックス、大正製薬 (現在 NEC)

他の GRAPE 関係専用計算機

- 1991 DREAM: 大規模偏微分方程式計算「専用」計算機。
  ハードディスクを主記憶に、という発想。
  ディスク 1 台の試作程度まで。大野、牧野、戎崎。
- 1993 ZEBRA : Radiocity 法専用計算機。成見、戎崎
- 1995 General: LU 分解専用計算機。市販チップで構成。
  清木、戎崎、泰地、牧野。
- 2002? MACE: LU 分解専用計算機。泰地、戎崎

ピーク性能の進歩



GRAPE-4 以降、完成した時点で世界最高速を実現

商業版 GRAPE

- GRAPE-3 から商業版
- GRAPE-6 を購入した機関

American Museum of Natural History
Drexel University
Indiana University Rochester Institute of Technology
Rutgers University
Rochester Institute of Technology
University of Michigan
University of California
McMaster University
The University of Cambridge
University of Edinburgh
Observatoire Astronomique Marseille-Provence(OAMP)
Astronomisches Rechen-Institute (ARI)
Ludwing-Maximillans University
Max-Planck-Institute fur Astronomic

GRAPE-6 を購入した機関 (つづき)

University of Bonn
University of Mannheim
Holland University of Amsterdam
Nanjing Univesity
Citec Co., Ltd
Gunma Astronomical Observatory
Hokkai-Gakuen University
Kansai University
Kyoto University
National Institute for Fusion Science (NIFS)
National Astronomical Observatory of Japan
Osaka University
The University of Tokyo
Tokyo Institute of Technology
University of Tsukuba

約 30 機関、 60Tflops。MDGRAPE-2 も同様

GRAPE-6 の次は？

MDGRAPE-3 の次: MDGRAPE-4, 20Pflops@2010

そもそも MDGRAPE-3 にあたるものは? → GRAPE-DR

GRAPE-DR 計画とは何か？

「基本的には」次期 GRAPE 計画

- 2004 年度から 5 年計画
- 目標ピーク性能: 2 Petaflops
- チップ数 4096
- 単体チップ性能 0.5Tflops

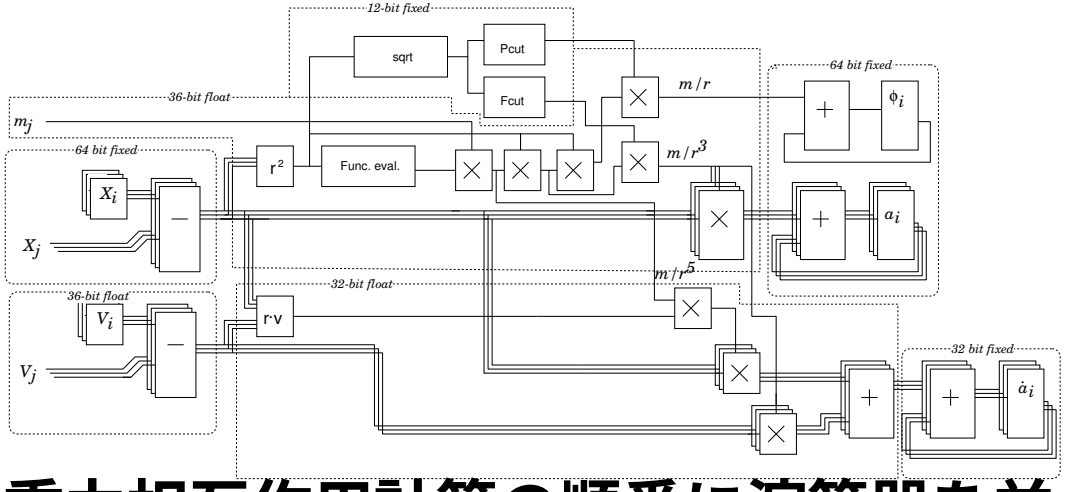
と、これだけなら今までの GRAPE が速くなっただけ。

実際のアーキテクチャ: 今までの GRAPE とは全然違う

- なぜ違うか
- それで何ができるか

GRAPE とはどんなものだったか？

プロセッサアーキテクチャ



重力相互作用計算の順番に演算器を並べたパイプライン

- シリコンの利用効率は極めて高い
- 動作クロックも上げやすい

- アプリケーション限られる。多種類作るのはリソースがかかり過ぎる

「次期 GRAPE」の実際的な問題

天文だけ (しかも理論だけ (しかも N 体だけ)) のの機械としてはチップ開発コストが大き過ぎる

チップ開発費

1990	1μm	1500 万円
1997	0.25μm	1 億円
2004	90nm	3 億円以上？
2006	65nm	10 億円以上

ある程度広い応用を持つものでないと予算獲得が難しい

ではどうするか

1. **やめる**

ではどうするか

1. やめる
2. 安くあげる方法を考える

ではどうするか

1. やめる
2. 安くあげる方法を考える
3. なんかお金を取る方法を考える

ではどうするか

1. やめる
2. 安くあげる方法を考える
3. なんかお金を取る方法を考える

GRAPE-DR では (3) を選択

基本的な考え

- 応用に特化し、多数の演算器を１チップに集積、並列動作させて高い性能を得た専用計算機の特徴を生かす
- しかし広い応用範囲を実現する

そんなことができるか？が問題

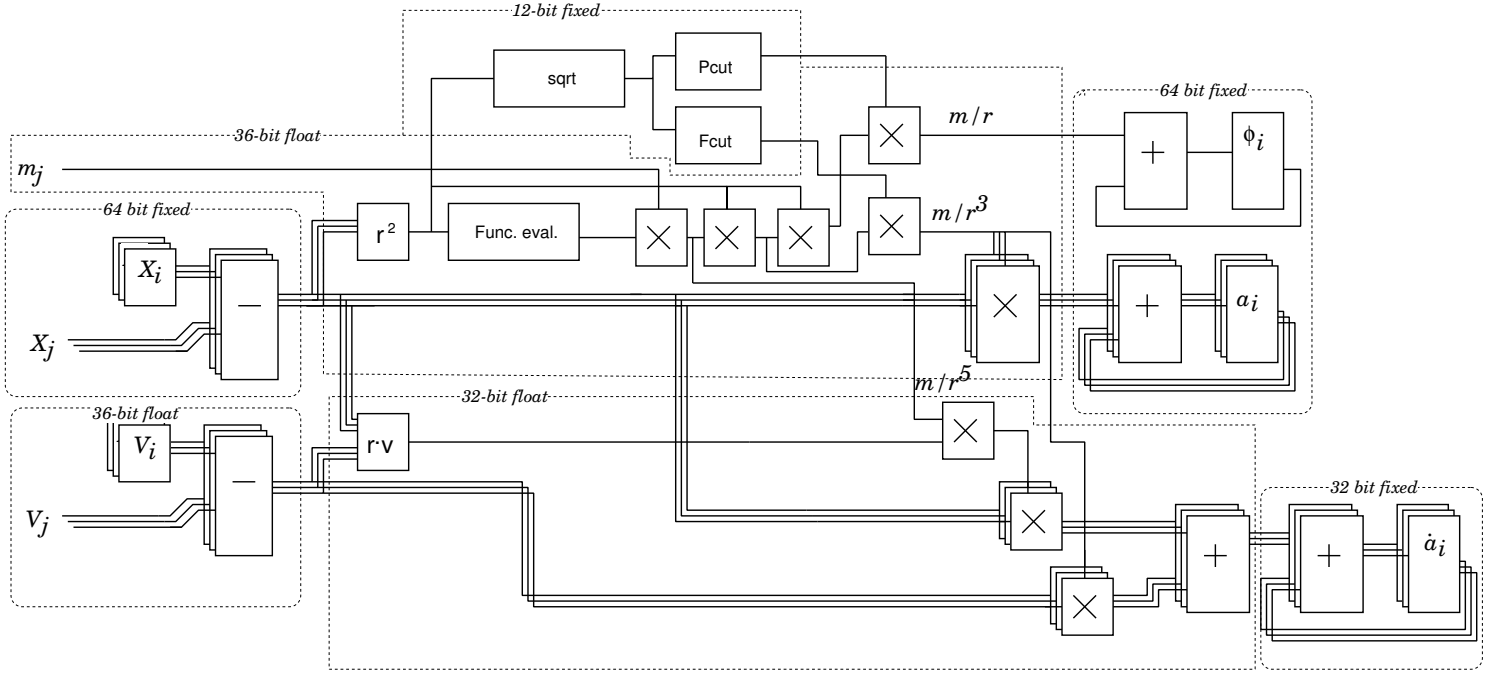
多数の演算器を詰め込む方法

境界条件: メモリバンド幅は増やしたくない (システムコストはほぼメモリバンド幅で決まる)
可能な方策

1. GRAPE 的専用パイプラインプロセッサ
2. 再構成可能プロセッサ
3. SIMD 並列プロセッサ

GRAPE的専用プロセッサ

これでよければ別に何も考えることはない。



再構成可能プロセッサ

FPGA ベース

- 任意のロジックを実現可能
- 集積度、速度は大きなペナルティがある
- 精度が低くてもいい応用には向く

いわゆる「動的再構成可能プロセッサ」
IPFlex DAP/DNA 等

- 8-32 ビットの単純な ALU を多数集積
- その間をプログラマブルな配線でつなぐ
- 集積度、速度はやはり大きなペナルティ

SIMD 並列処理

パイプラインプロセッサをやめにして、「プログラム可能なプロセッサ」を沢山載せる。

SIMD (Single Instruction Multiple Data): 全プロセッサが同じ命令を実行

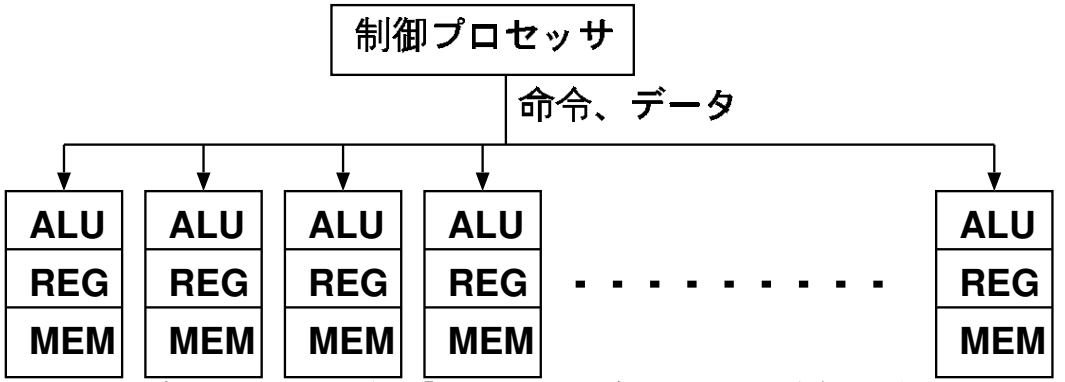
基本的には、全プロセッサがソフトウェアで GRAPE をエミュレーションする。

SIMD 並列処理って？

- 古典的 SIMD 並列計算機
- SSE、MMX とかの SIMD 拡張命令
- GRAPE-DR における SIMD

古典的SIMD並列計算機

Illiac IV, Goodyear MPP, ICL DAP, TMC CM-2, MASPAR MP-1

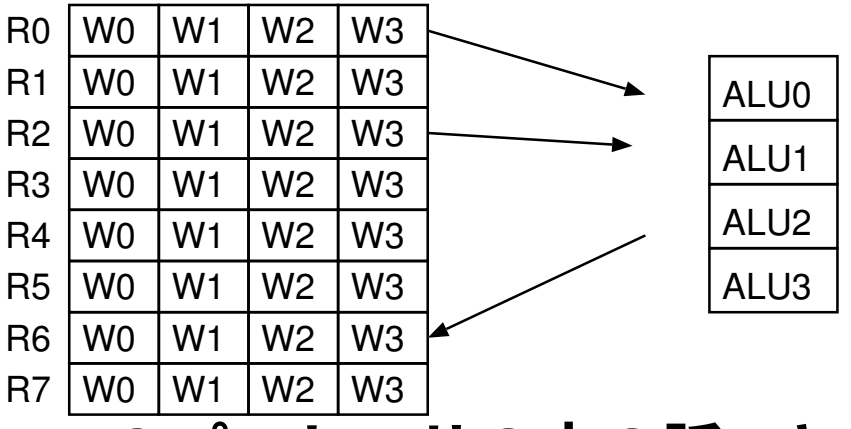


1960年代に発生、80年代に絶滅
半導体技術の向上に対応できないアーキテクチャ: 計算速度とメモリアクセス速度が比例する必要あり。
メモリ階層をつける: プロセッサが複雑になりすぎて SIMD の意味が無くなる。

SIMD 拡張命令

Pentium III, IV が有名

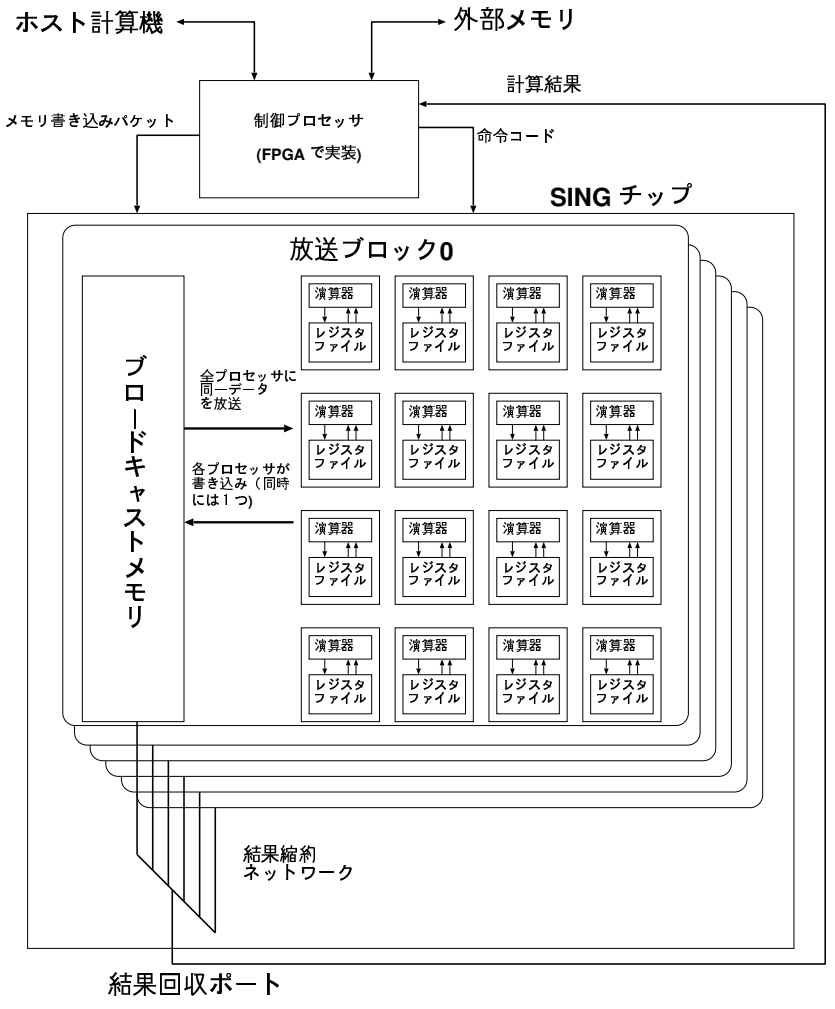
128 ビットなり 64 ビットのデータを 4 語に区切って、それぞれの要素に対する演算を 4 個の演算器で同時に処理



1 つのプロセッサの中の話: キャッシュとデータをやりとり

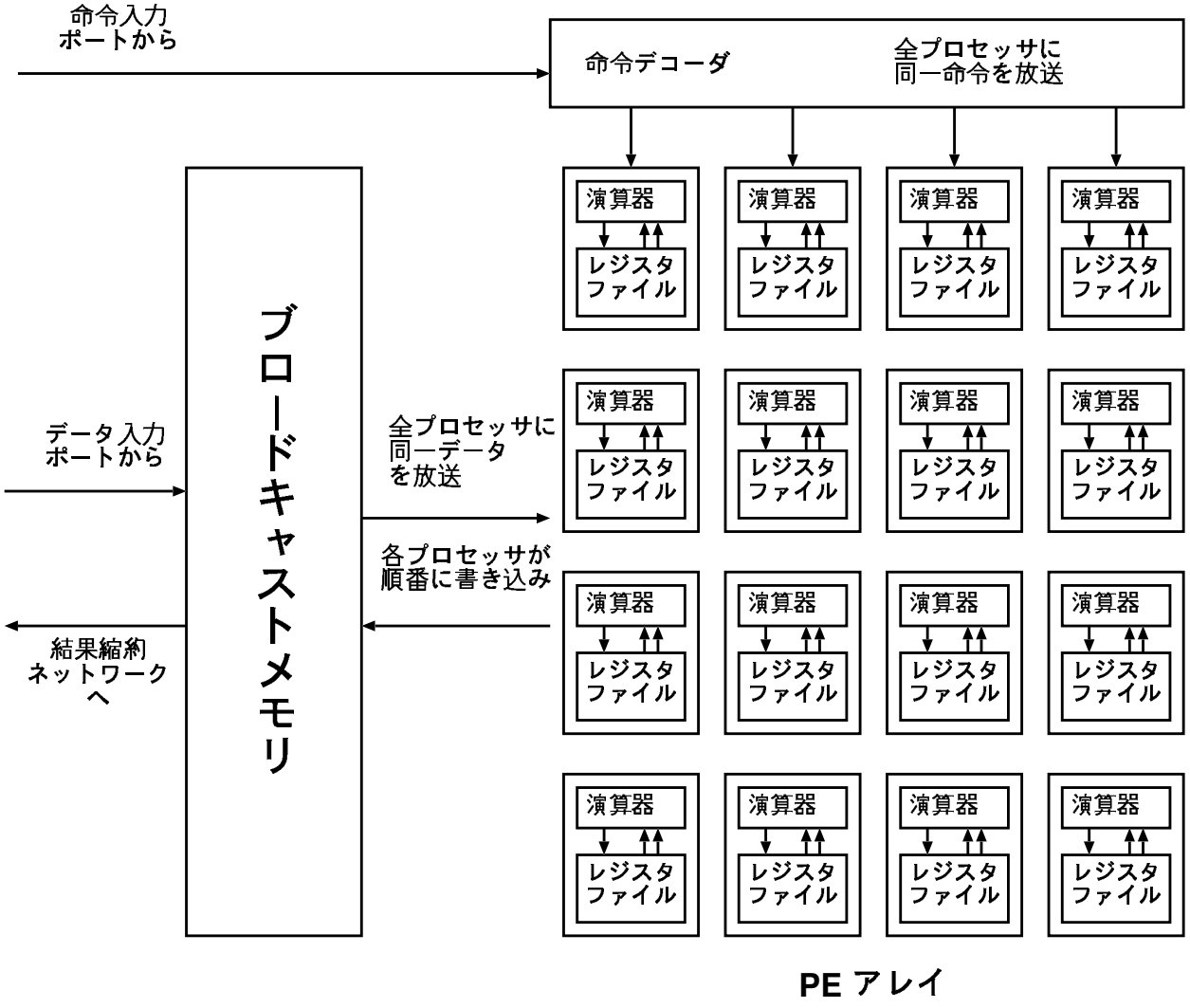
並列度 4 程度が限界？それ以上増やすとキャッシュの速度が追いつかなくなる。

GRAPE-DR における SIMD



- 非常に多数のプロセッサエレメント (PE) を 1 チップに集積
- PE = 演算器 + レジスタファイル (メモリをもたない)
- チップ内に小規模な共有メモリ (PE にデータをブロードキャスト)。これを共有する PE をブロードキャストブロック (BB) と呼ぶ。
- 制御プロセッサ、外部メモリへのインターフェースを持つ

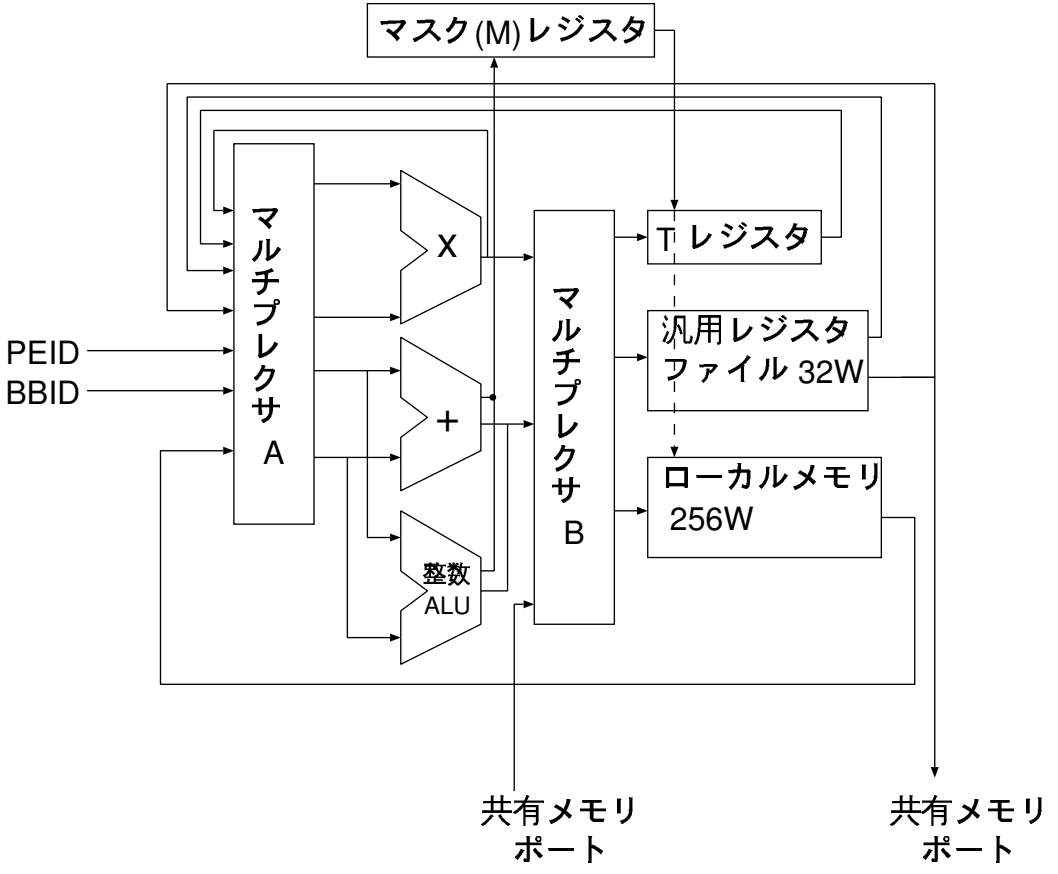
ブロードキャストブロック



各プロセッサは同じデータをブロードキャストメモリから受け取る

このブロックがチップ内には沢山ある。答はブロックにまたがって合計することもできる。

PE の構造



- 浮動小数点演算器
- 整数演算器
- レジスタ
- メモリ (256 語),K とか M ではない。

PEの詳細

データ形式

単精度浮動小数点: 36 ビット (符号1、指数11、仮数24)
倍精度浮動小数点: 72 ビット (符号1、指数11、仮数60)
36/72ビット固定小数点数

演算命令

乗算は単精度のみ (倍精度のための部分積をサポート) 倍精度乗算を 2 サイクルでするために 25×50 ビットの乗算器
整数演算、加減算は倍精度のみ (メモリ/レジスタからの読出し/格納時に単・倍変換ができる)
特殊な浮動小数点命令: 仮数を正規化しないまま演算を続ける。これにより、演算順序によらないで結果が同じになることを保証する (GRAPE-6 の積算と同様)
普通に正規化もできる (こっちがデフォルト)

PEの詳細(続き)

- パイプラインは8ステージ。
- 基本命令は4データに対するベクトル命令。4サイクルに1回しか命令ははいらない。
- T レジスタのみ直前の命令の実行結果を利用可能。
- T レジスタはアドレスレジスタになる(間接アクセス)

サポートする命令等は基本的には昔の SIMD 計算機、例えば CM-2, MasPar MP-1 なんかとあまり変わらない。但し、PE がはるかに強力になっている。

アプリケーションに対する考え方

- Memory Wall が問題にならないような
  アプリケーションのみを対象にする
- 3 つの型に特化
  - 散乱実験型
  - 粒子間相互作用型
  - 密行列型
- 可能ならばアプリケーションを書き換える

散乱実験型

- 多数の PE が、独立にイベントを計算
  - イベント間の相互作用はない、または非常に少ない
    - レイトレース計算：光学部品（レンズ、導光版）設計
    - 放射線伝播のモンテカルロ計算：検出器設計
    - 3 体問題:連星と単独星の遭遇、微惑星同士の遭遇
- “Embarassingly Parallel” とほぼ対応
- 古典的 SIMD 機と同様の振る舞い:
  - Goodyear MPP, ICL DAP, TMC CM-1/2, Maspar MP-1/2
  - 極端に少ないメモリ
  - PE 間通信が遅い
- 計算速度と通信速度の比:
  - 散乱実験の計算がどれだけ複雑かで決まる

粒子間相互作用型

$$f_i = \sum_j f(x_i, x_j)$$

- 他の「粒子」との「相互作用」を縮約。
  - 全ての相互作用を並列に計算可能
  - 同じ「粒子」のための計算結果を高速に縮約する必要
- 計算手順
  - PE に相互作用を受ける粒子をロード
  - 相互作用を及ぼす粒子をロード
  - 計算機終了したら結果を縮約しながら回収
  - 計算速度とチップ外への通信速度の比:
    相互作用を及ぼす粒子数に比例

密行列型

$$c_{ij} = \sum_{k} a_{ik} b_{kj}$$

- 計算手順
  - 行列が PE に収まるところまで分割。それから
  - 行列 A の部分行列を PE にロード
  - B の1列を分解して各グループにロード
  - 各 PE で B の部分列と A の部分行列の積を計算
  - 計算が終わったものから順次回収。グループ間で合計
- 計算速度・通信速度の比はチップ全体にロードできる行列のサイズに依存
  - メモリサイズの平方根に比例して通信速度を落とせる

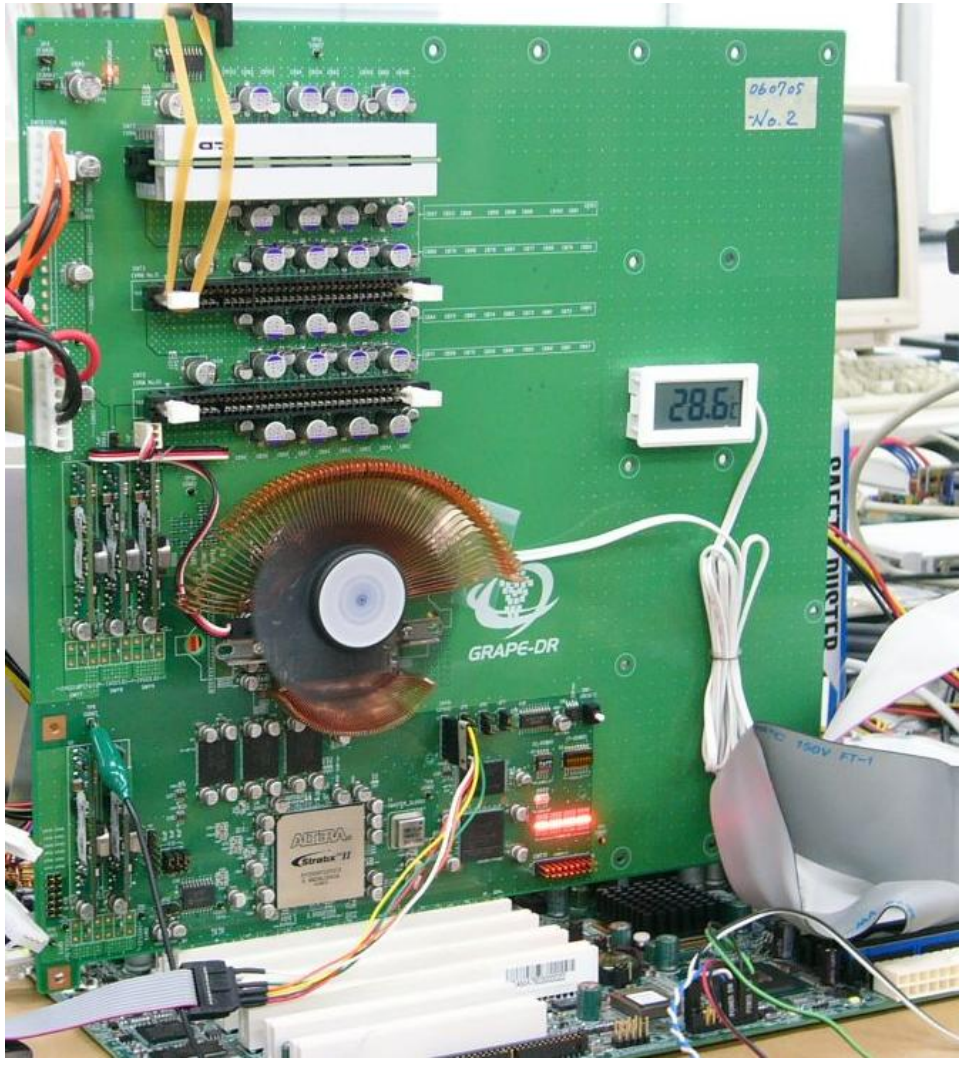
計算・通信比のまとめ

- 散乱実験型: アプリケーション依存
- 粒子間相互作用型: 粒子数依存
- 密行列型: オンチップメモリサイズ依存
- 設計におけるトレードオフ:
  - なるべくアプリケーション範囲を広く
    - メモリを多く、バンド幅を広く $\rightarrow$ コスト増
  - コストを圧迫しないようにバランスを考える必要あり
- 実際の設計では密行列型の要求がもっとも厳しい

GRAPE-DR の開発状況

シミュレーションデータと同じものを供給して同じ答がでるところまで確認。
(これとは別ボードで) 500MHz 動作も確認、消費電力 25-50W 程度。

GRAPE-DR 別ボード



- こっちが「プロジェクト公式」
- 中身は殆ど同じ
- 何故か大きい
- 500MHz 動作まで確認済

原始的なコンパイラ

(中里 2006)

```
/VARI  xi, yi, zi, e2;
/VARJ  xj, yj, zj, mj;
/VARF  fx, fy, fz;
dx = xi - xj;
dy = yi - yj;
dz = zi - zj;
r2 = dx*dx + dy*dy + dz*dz + e2;
r3i= powm32(r2);
ff = mj*r3i;
fx += ff*dx;
fy += ff*dy;
fz += ff*dz;
```

これから GRAPE 並のことをするアセンブラ、インターフェースライブラリを生成。
基本的なアイデアは PGR (FPGA を使ったPROGRAPE用コンパイラ、濱田D論2006) と同様

まとめ

- GRAPE では、重力相互作用計算に専用化したパイプラインプロセッサをフルカスタム LSI で実現することで、同時期のマイクロプロセッサの数十倍の性能を数分の一の商品電力で実現してきた。
- このアプローチは、 LSI 開発のコストが膨らみすぎたために困難になった。
- GRAPE-DR では、SIMD 方式でプログラム可能にすることで GRAPE より広い応用を実現する。コストは数倍あがるが我慢する。
- サンプルチップは完成し、シミュレーション通りの動作をすることが評価ボードで確認できた。
- 計画では 2008 年度にピーク性能 2Pflops を実現する。