ポストエクサスケールの計算機アーキテクチャ

牧野淳一郎
神戸大学理学研究科惑星学専攻/惑星科学研究センター

第16回アクセラレーション技術発表討論会「高度計算科学の現状と未来」 2022/9/16

但し書き

本講演の内容は牧野個人の考えを表すものであり、8月に発足した「次世代計算基盤に係る調査研究―オリジナルアクセラレータアーキテクチャによる超低消費電力次世代計算基盤の評価」の方向性そのものを表すものではありません。

まとめ

- ポストエクサスケール=ポストムーアの高性能プロセッサ開発における課題を整理した。
- 基本的な問題は、「我々は高性能プロセッサを定量的に評価する適切な方法をもっていない」ということである。
- 原理的には、効率を「使っている半導体技術で構成した、その計算をするのに必要な理想的に無駄のない演算回路の組合せ論理の部分の消費電力と実際の消費電力の比」と定義できそう。
- 現状のプロセッサではこの比は実用アプリケーションで「すごく低い」。**1%** とかのレベル。
- **GRAPE-DR/MN-Core** 的なアプローチでこれを **10%**以上にはもっていける。多様なアプリケーションが本当にみんな動くのか？は検証が必要。

話の構成

- ポストエクサスケールにおける大規模数値計算の展望
  - 何が問題と普通考えているか？
  - 本当の問題はなにか？
- 「高性能」という言葉の意味
- 過去・現在のプロセッサの評価
- **MN-Core** の立ち位置
- まとめ

ポストエクサスケールにおける大規模数値計算の展望

- 何が問題と普通考えているか？
- 本当の問題はなにか？

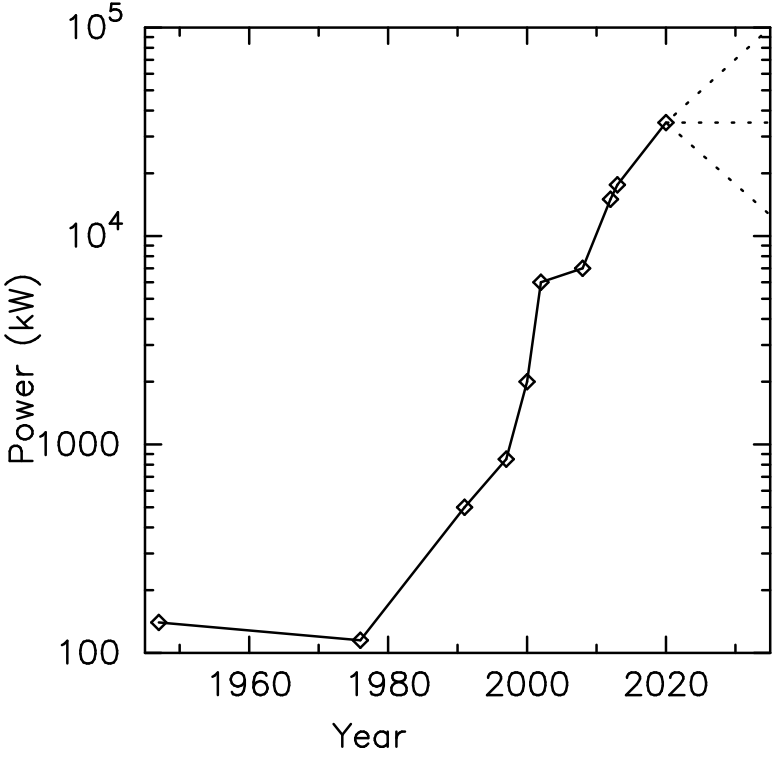
何が問題と普通考えているか？

- ムーアの法則の限界
  - 微細化の限界 → 動作速度の向上、トランジスタ密度の向上、消費電力の低下の限界
  - システムとしての消費電力の爆発的増加
- むしろムーアの法則の結果としての困難
  - (オンチップで集積できるトランジスタ数が増えたことに対応するための) アーキテクチャの複雑化 → 実行効率の低下、アプリケーション開発、チューニングの困難化

消費電力:ENIAC から「京」まで

ENIAC	**1947**	**140kW**
Cray-1	**1976**	**115kW**
Cray C90	**1991**	**500kW**
ASCI Red	**1997**	**850kW**
ASCI White	**2000**	**2MW**
ES	**2002**	**6MW**
ORNL XT5	**2008**	**7MW**
「京」	**2012**	**20MW**
富岳	**2020**	**35MW**

グラフにしてみると、、、



ENIAC から **Cray-1** まであまり変わらない

そのあと **20** 年間で **10** 倍

そのあと **10** 年間でさらに **10** 倍

そのあとさらに数倍

スパコン費用の大きな部分が電気代になってきた

古典的 **CMOS** スケーリングとの関係

古典的スケーリング
フィーチャーサイズ **1/2** → 面積当りキャパシタンス **2** 倍、電圧 **1/2**、クロック **2** 倍
→ 消費電力不変、性能 **8** 倍

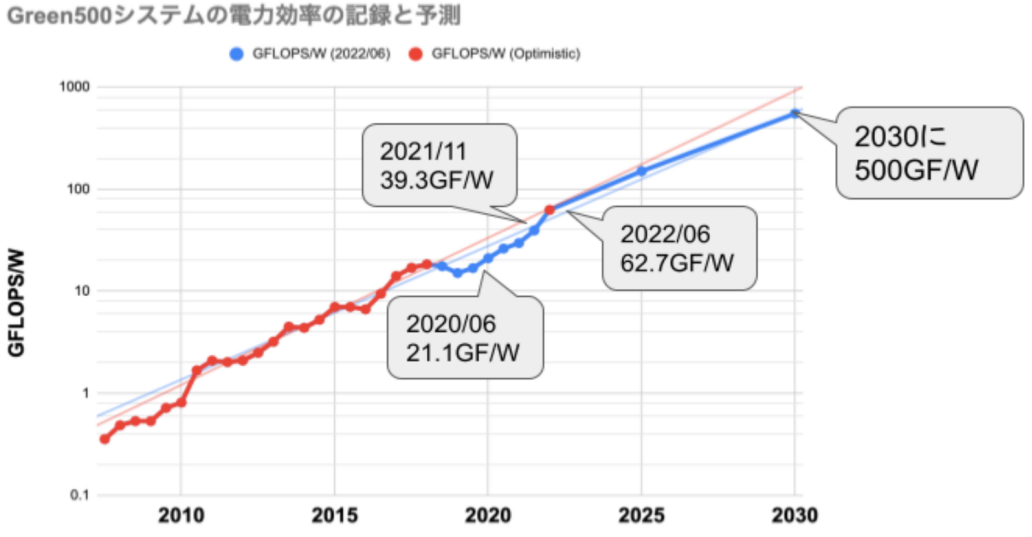
現実 **(**ムーアの法則の終わり**)**

- 電圧下げるのは限界 **(1V** 以下では動作速度急激に下がる。 **0.4V** 以下の動作は困難、メモリは **0.6V** くらいが限界**)**
- 微細化してもトランジスタ構造の複雑化のためキャパシタンス、つまりは消費電力が下がらない

消費電力一定 → クロック下げないといけない

最近のマイクロプロセッサ**:**
性能上がるけど電力もどんどん増える **(AMD EPYC** とかがそうなっている**)**

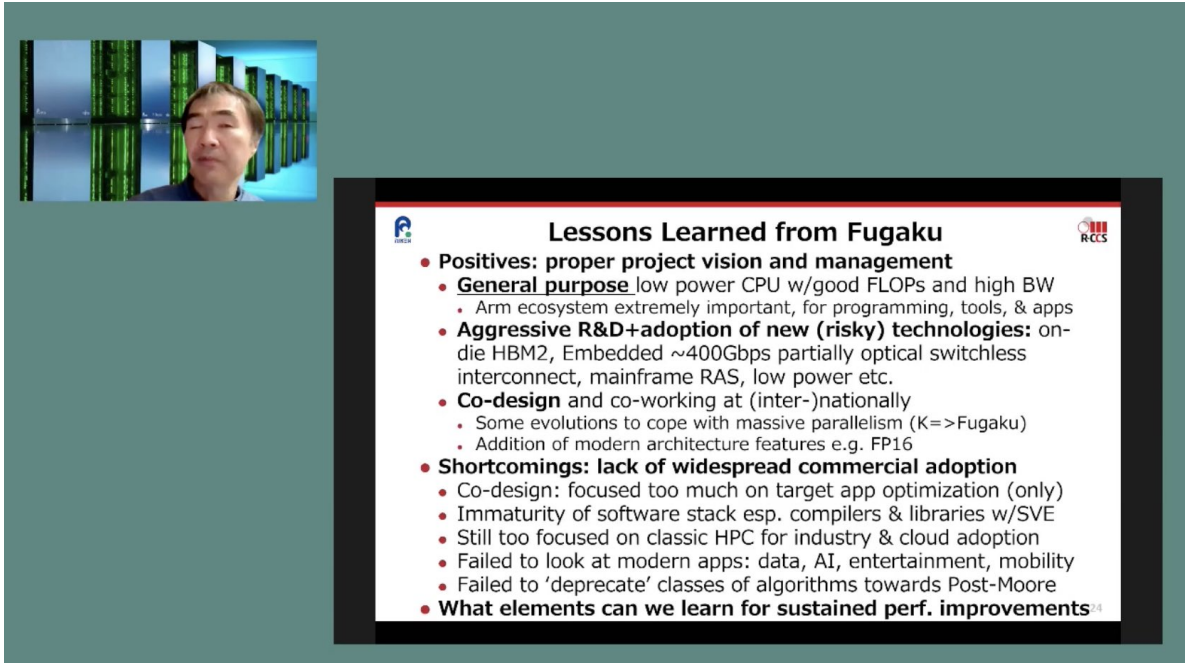
消費電力向上のトレンド



2013 年くらいから全部アクセラレータ **(2019/11** の **A64fx** を除く**)**
GPU 設計者はカード電力リミット **(300W** を大きくは超えないよう努力してきた

GPU 以外のアクセラレータ **(**国産**)** は同じ半導体テクノロジーなら **GPU** より若干ないしかなり良い数字をだしてきた
とはいえ**: 1TF/W** になってやっと **20MW** で **20EF**。半導体技術の進歩による電力性能向上があまり期待できない今後、**500GF/W** も容易ではない。

実行効率の問題 —A64fx の例



SIAMPP22

要するに: 「実用アプリケーションで性能でてないじゃないか」
原因:

- **A64fx** 固有の問題
- 一般的問題

A64fx 固有の問題

Skylake Xeon とのレイテンシ等の比較

	A64fx	**Skylake**
浮動小数点演算レイテンシ	**9**	**4**
L1 キャッシュレイテンシ	**8-11**	**4**
L2 キャッシュレイテンシ	**37-47**	**14**
L3 キャッシュレイテンシ	**-**	**50-70**
命令ウインドゥ	**128**	**224**
ロード/ストア	どちらか	同時実行

命令レイテンシが２ー３倍、命令ウインドゥが半分、バンド幅も半分くらい

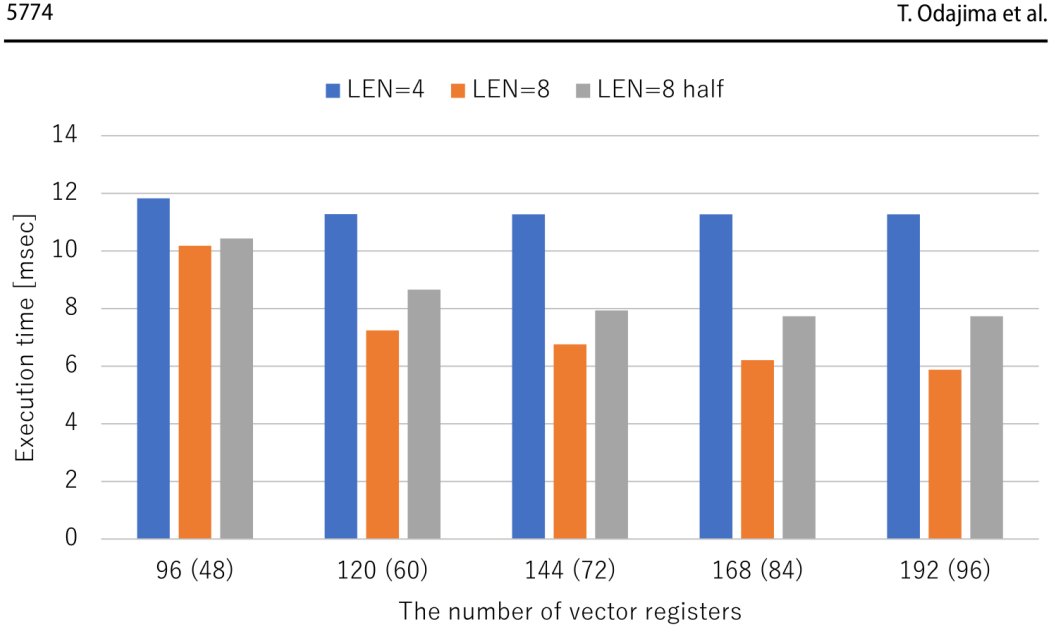
↓

Skylake では実行時にスケジューリングしてパイプラインを埋められるループでもリソース不足でパイプラインが埋まらない

レイテンシ大きくなった理由の推測**:** 低消費電力化のため低電圧で動作する低速・ドライブ能力の低いトランジスタを主に使ったので、動作周波数を保つためには深いパイプラインが必要になった。

それ作る前に気が付かなかったの？

公開資料から言える範囲で、、、



(演算器等のレイテンシは **A64fx** のよりずっと小さい)

Odajima, Kodama and Sato 2021, J Supercomput 77, 57575778 (2021).
元々は **Cool Chips 21 (2018/5)** で発表**=2017** 年の研究
SVE 命令を **1024** ビット幅にして **2** サイクルで処理するほうが「物理レジスタの面積が同じでも」性能がずっと上がる

理研は問題を認識していた

一般的問題

ワイド **SIMD** 命令セットをもつ共有キャッシュアーキテクチャメニーコア共通の問題

ほとんどのアプリケーションで実行効率が低い

(密行列乗算が演算カーネルでないとなかなか性能でない、、、**)**

理由は沢山の色々なものの組合せだが、

- メモリへのストライドアクセス、ランダムアクセスが相対的に非常に遅い
- キャッシュサイズが相対的に小さくなっていて、メモリアクセスを隠蔽できない
- メモリバンド幅が相対的に下がっている

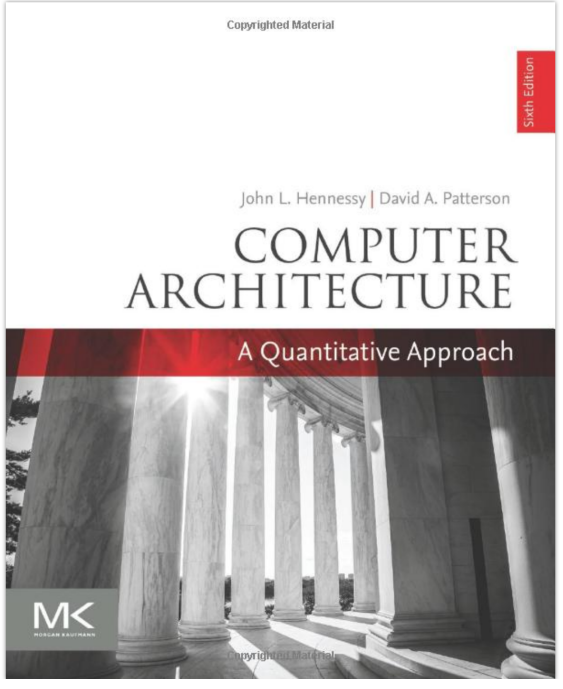
本当の問題はなにか？

ここまでの話はよく聞くような話なわけですが、よく考えるとそういう考え方でいいのか？という気が。

そもそも我々は(というか **I** とか **A** とか **F** とか **N1** とか **N2** とかは) どうやってアーキテクチャ決めてるのか？

- **x86** みていると言葉の正しい意味で「進化」に見える。
- とはいえランダムに変化させてるわけでもなく、色々なパラメータ (例えば **SIMD** 幅とかキャッシュサイズとか) は「増やしてよくなるなら採用」とかの局所最適化になっていると思われる。
- そうすると、進化のいきつく先は局所最適解で、大域的な最適解ではないのではないか？

本当の問題はなにか？



右側の本の序文から

「ヘネパタは、コンピュータの設計の指針として「定量的アプローチ」を掲げている。これによるとコンピュータの設計は、実用的なプログラムを実行した評価に基づき、コスト、性能、電力を定量的に測定することを通して行わなければならない。

この定量的アプローチは絶対的に正しい」

この主張自体はもちろん間違ってはいない

が、実際の **HPC** 応用を考えると非常に大きな問題がある。**(HPC** 応用以外も同じ**)**

現存の「実用プログラム」は現存の **(**というのは嘘で **10-20** 年前に存在した**)** プロセッサに最適化されている **(**まあ全然最適化されてないこともあるが**)**

↓

設計で可能な探索空間 **(**「最適解」が見つかるところを含めて**)** が、現存 **(**あるいは **10-20** 年前の**)** プロセッサの近傍になる。

つまり**:** 本当の最適解 **(**がどんなものかはさておき**)** 全然明後日の方向にプログラムとアーキテクチャが共進化してきた「かもしれない」

= 「本当の問題」

つまり問題は

- 我々は「このプロセッサはどれくらい良いか」を定量的に判断する規準をもっていない。
- 「実用的なプログラム」での「コスト、性能、電力」は局所的。大域的に有効ではない。

「高性能」という言葉は何を意味するべきか？

よくわかんないので他の業界を考える。

- エンジン: 燃料の熱エネルギーをなるべく沢山機械的運動エネルギーに変えられるのがよいエンジン。熱力学第二法則による限界がある。
- 飛行機: 主翼に対する摩擦抵抗以外は原理的には減らせるはず。誘導抗力、圧力抗力。
- (アルゴリズム: 演算数で比較はできる。)

「汎用」「高性能」は定義可能か？ ―現実的な「汎用」性

- まず、何を最小にするものと定義するか？
  - エンジンや飛行機: まずは必要エネルギー。それがコストの全部ではないが大きな割合を占める。
  - 計算機でも、エネルギー消費=電気代は大きな割合になっている。
- エネルギー消費最小は実際的な意味はありそう
  - でも現実的に定義できるか？

HPC における「エネルギー最小」の定義の問題

- エンジン**:** 「効率 **100%**」が定義可能。入力熱エネルギーから第二法則の限界まで機械的エネルギーに。飛行機も同様**:**圧力抗力、誘導抗力ゼロ。
- 計算機**:** 半導体技術、動作電圧、ターゲット周波数とかで同じ論理回路でも電力は全く変わる。そもそも、アプリケーションだって色々ある。基準にできるものはあるか？

半導体技術に依存しない定義は可能か？

- 一つの考え方: 半導体技術 (使うプロセスルール、動作電圧) を決める
- そうすると、あるアルゴリズムに従った計算 (計算性能や数値フォーマットも決めたとして) するのに「エネルギー最小」の回路は原理的には決まるはず。
- これは、各演算を必要な最小精度で実現し、演算の組合せ論理だけの消費電力を考えることに相当する。
- メモリアクセス、レジスタアクセス、パイプラインレジスタの電力消費、チップ内でのデータ移動に伴う電力消費はすべて無駄、つまり飛行機の場合の圧力抵抗や誘導抵抗みたいなものと考える。演算は摩擦抵抗。
- そうすると、結局、効率を、「全消費電力と、その半導体技術で構成した理想的な演算回路の組合せ論理部分の消費電力の比」と定義できたことになる。(これは半導体技術に依存しない定義になっている)

この定義には意味があるか？

- どうしてもメモリアクセスが多い
- どうしても演算毎にデータが物理的に遠くまで動く必要がある

とかでなければそんなに悪くないはず。実際のアプリケーションではどうかということに。
というわけで、アプリケーションのほうはどうか？

アプリケーションの「多様性」

アプリケーションにはどれくらい色々あるか?日、米、欧のエクサスケールプロジェクトでの対象アプリケーションを見ると:

分類	数
構造格子	**14**
非構造格子	**13**
粒子	**2**
ランダムグラフ	**2**
密行列	**3**
その他	**3**

「ランダムグラフ」は神経回路シミュレータ。**NEST**とかは実装は構造格子に近い。
「その他」の **1** つはゲノミクス。　日本の**GENOMON(**中身は **BWA-MEM**、、、**)**
今はもちろん深層学習が重要。これは結局 **(**低精度**)**密行列。

つまり

- 構造格子・非構造格子が多い
- 残りの多くが深層学習と量子化学・物性計算 (密行列)
- あと粒子法とゲノム (動的計画法)
- これくらいできれば実際上「汎用スパコン」
- 深層学習は専用計算機でいいかも (**AI** プロセッサ、、、) あるいは **AI** プロセッサがおまけで他のこともできるとか。(「現実的な **HPC** のエコシステム」)

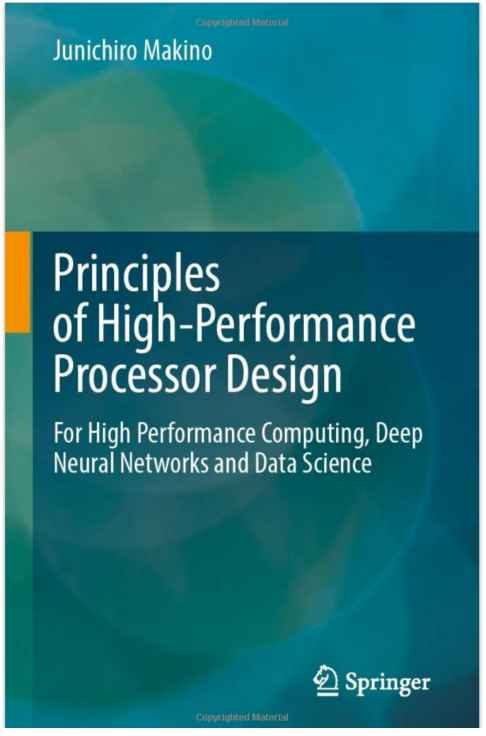
現実的汎用性からみた過去と現在のプロセッサ

- 現実のプロセッサについて、「アプリケーションでの演算部論理の消費電力の割合」をだすのはそんなに簡単ではない。
- 雑だが相関はありそうな指標:コアロジックの全トランジスタ数に対する、演算ロジックに使われていそうなトランジスタ数の割合。
- 過大評価: 演算器以外はフルに動いてるとは限らない。
- 過小評価: 効率 **100%**で演算器が動いてるわけではない
- ちょっと別の話: 特に **7nm** 以降、「トランジスタの価格は上昇」
- それなりに意味があるかも？

牧野の本から

	Single Core	MIMD	In core SIMD	Global SIMD	Coherent Cache	Non-Coherent Cache	Local memory	Transistor efficiency
CM-2	-	-	-	✓	-	-	✓	**30%**
Illiac IV	-	-	-	✓	-	-	✓	**7%**
Cray-1	✓	-	-	-	-	-	-	**6.5%**
Intel i860	✓	-	-	-	-	✓	-	**6.5%**
PEZY-SC	-	✓	-	-	-	✓	✓	**2.4%**
NVIDIA A100	-	✓	-	✓	-	✓	✓	**1.7%**
Fujitsu A64fx	-	✓	✓	-	✓	-	-	**1.3%**
Intel Skylake	-	✓	✓	-	✓	-	-	**0.8%**
Intel P4	✓	-	✓	-	-	✓	-	**0.3%**
Intel Core2	-	✓	✓	-	✓	-	-	**0.08%**
SW26010	-	✓	✓	-	-	-	✓	**-**

宣伝



Principles of High-Performance Processor Design: For High Performance Computing, Deep Neural Networks and Data Science
Springer 2021/8

Kindle もあります。

表見ると:

- **MIMD+In core SIMD+**コヒーレントキャッシュでは **1%** を超えるのは難しそう。
- **A64fx** は **1%** 超えていて数字は確かに良い。
- **A100** と **PEZY-SC** は **2%**前後で、さらに良い。
- 昔のベクトルプロセッサや **SIMD** 超並列マシンは非常によい。

要するに**:** マルチコア**+**コヒーレント階層キャッシュで高い効率のプロセッサを設計するのは「無理」
ある意味当たり前**:** キャッシュが大量の電力を制御しにくいやり方で消費する。

実行効率の観点

- コヒーレント階層キャッシュっておいしいの**?**
- コヒーレント階層キャッシュのままで頑張るとどうなるの**?**

この辺皆様思うところがあるのではないかみたいな。

「理想に近い」プロセッサの例

- 専用計算機ならもちろんそうできる。**GRAPE-3,4,5,6** とか。
  - **GRAPE-4**、**6** は演算器 **1** つ当たり **20k** トランジスタくらい **(FMA** ユニット換算**)**。普通の **FMA** ユニットは **100k** トランジスタくらい必要。
  - なんでも倍精度でやるのに比べて **5-10** 倍効率あげられる。
- 汎用プロセッサでは、、、手前味噌だけど **GRAPE-DR** の例を。
  - チップレベル **SIMD**
  - ローカルメモリ**+**レジスタ**+**階層的チップ内ネットワーク
  - チップ内ネットワークは放送/縮約をサポート
  - 演算器の割合 **14%**くらい。確かに高い。電力性能はチップレベルで **3GF/W** くらい **(90nm)**。
  - まあ「汎用」といえるか？という問題点はある。
  - トランジスタ効率では専用回路に比べると **50** 倍くらい悪い。

主要な問題点

- **GRAPE-DR** くらい極端な設計でもまだ専用回路に比べると数十倍損
- **GRAPE-DR** みたいな極端な設計で本当に色々使えるか？プログラムどうやって書くのか？

前者は、粒子系以外ではそこまで差がないかもだけど、逆にいうと研究開発の余地がまだ一杯あるということでもある。

後者はちゃんと考える必要あり。とはいえ **OpenCL** まがいのもの載せれば勇者 **(**人柱ともいう**)** は使うのではないか？

チップ内ネットワークとか外付けメモリアクセス方法とかは検討必要 **(**机上検討はさっきの本に大体書いたつもり**)**

まとめ

- ポストエクサスケール=ポストムーアの高性能プロセッサ開発における課題を整理した。
- 基本的な問題は、「我々は高性能プロセッサを定量的に評価する適切な方法をもっていない」ということである。
- 原理的には、効率を「使っている半導体技術で構成した、その計算をするのに必要な理想的に無駄のない演算回路の組合せ論理の部分の消費電力と実際の消費電力の比」と定義できそう。
- 現状のプロセッサではこの比は実用アプリケーションで「すごく低い」。**1%** とかのレベル。
- **GRAPE-DR/MN-Core** 的なアプローチでこれを **10%**以上にはもっていける。多様なアプリケーションが本当にみんな動くのか？は検証が必要。

議論

Q: メモリバンド幅は？
A: おそらく **DRAM** の **3** 次元実装 **(AMD V-Cache** の **DRAM** 版) がかなり安価に利用可能になる。 **0.x pJ/bit** くらいまではいけるかも。
Q: チップ内部ネットワークは？
A: 現在はツリー的なものだけ。さすがにちょっと厳しいかも。ポスト京 **FS** ではメッシュネットワークを検討した。
Q: チップ間ネットワークは？
A: MN-Core も実はチップ間ネットワークがある。おそらく何か考える。