

# Post-Exaの HPCシステムアーキテクチャ

2015年12月18日  
株式会社富士通研究所  
久門 耕一

# お断り

本資料に記載する内容は、私（久門）の現時点での個人的見解です。

富士通並びに富士通研究所の見解を示すものではなく、実現をお約束するものではありません。



- 汎用CPUとは、
  - 演算器を高速に動かし時分割で様々な用途に利用
  - 複数の演算間のデータ受け渡しをメモリを使って実現
  - 逐次実行セマンティクスを維持しながら細粒度並列実行するため、複雑な実行制御とデータ受け渡し
  - 常にRF（やキャッシュ）とALU/FPUとのデータ移動にエネルギーを使う
  - Moore則の進展で、（実はアーキテクチャ的改善はあんまり大きくな<sup>く</sup>）50年近く進歩してきたつもり
- データ移動や制御オーバヘッドの低減を目的に、基本演算に、FMAやSIMD命令を追加
- 逐次処理のセマンティクスの維持コストはかなり高い

# もう一つの観点(~~アクセラレータ~~万歳)

- やりたいことが決まっているなら、ASICを作ればよいのだが、

## アクセラレータの特徴と問題点

- ノイマン型に比べプログラミングに強い制約を受けるため、問題に依存し実装コストか実行効率が犠牲になるかもしれない
- 例えば、Graph500とTop500を比較
- 双方とも、実装レベルが同じと仮定。

システム名	Top500 PFLOPs	Graph500 PGTEPs	比率(TOP500 /Graph500)
京	11.3	38.62	3.42 (7.1)
TSUBAME2.5	2.8	1.34	0.48 (1)

- 京とTSUBAME2.5ではグラフ処理/LINPACKの比が7倍

# じゃ、FPGAにしよう、、、

- FPGAはHPC領域では、Gate Arrayと言うよりも、DSP要素+ゲートの Arrayと見るべき。

FPGAとは、

- 演算器間を可変制御ロジックと可変結線で繋ぐデバイス。
- 実行クロックはCPUに比べ低いが、演算器数で並列度を稼ぐ
- CPUが得意でない単純処理を高速にこなす
- クロック同期で動的待ちを生じずに処理を行う→低オーバヘッド

問題点

- 配線領域がチップの70%以上を占め使わない結線が多数
- ただし、ASICに比べると10倍以上の効率低下がある。
- そもそも、プログラミングに良い手法がない。  
ソフトで書けるなんて、大いなる幻想

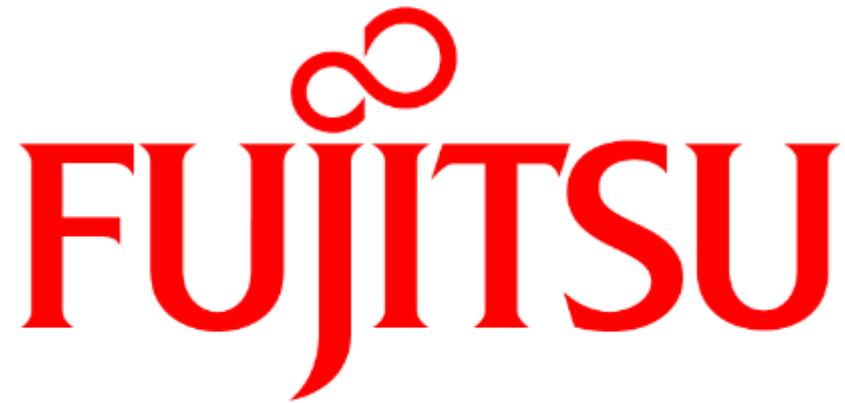
# 汎用CPUの柔軟性を持ちASICの効率を。。

- 2010年に出現したGraph500は、京にとって未知のアプリだったが、世界1位
- アプリが定まっているならASICでOK。
- アクセラレータを使う場合、アプリが限定されることを合意し、更にソフト開発手法の同時開発が必要

以下、勝手に妄想を書きます（未完）

- （SIMDのように）細粒度での動的待ち合わせは最小限
- データ移動距離を不必要に大きくしないため、FMAより高機能な演算を基礎演算とする
- 演算カーネルを高速化可能な小規模な可変要素を持たせる
- 遅延時間の隠蔽のため、マルチスレッド化

。。。。。

The logo features a red infinity symbol positioned above the word "FUJITSU". The word "FUJITSU" is rendered in a bold, red, serif typeface. The letter "J" is stylized with a long, sweeping tail that extends downwards and to the left.

**FUJITSU**

shaping tomorrow with you