筑波大学計算科学センターシンポジウム
「計算機アーキテクトが考える次世代スパコン」

2006年4月5日

村上和彰
九州大学
murakami@cc.kyushu-u.ac.jp

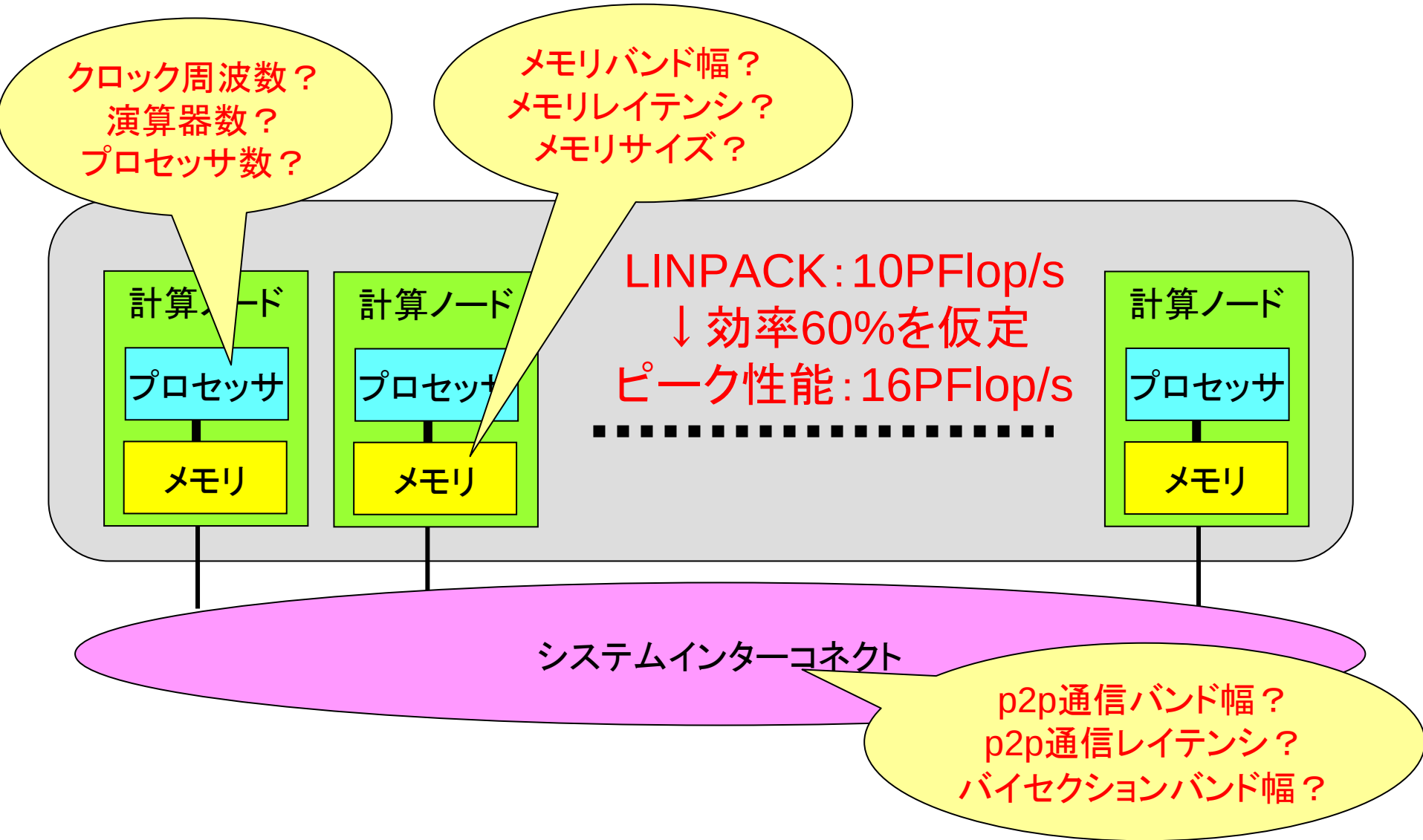
次世代スパコン
～達成目標と制約条件の整理～

- 達成目標
  - 性能目標（2011年）
    - LINPACK (HPL)：10PFlop/s
    - 実アプリケーション：1PFlop/s
  - 成果目標
    - （私見）科学技術計算能力の国際競争力の向上ならびに維持による我が国の科学技術力、産業力の発揚
- 制約条件

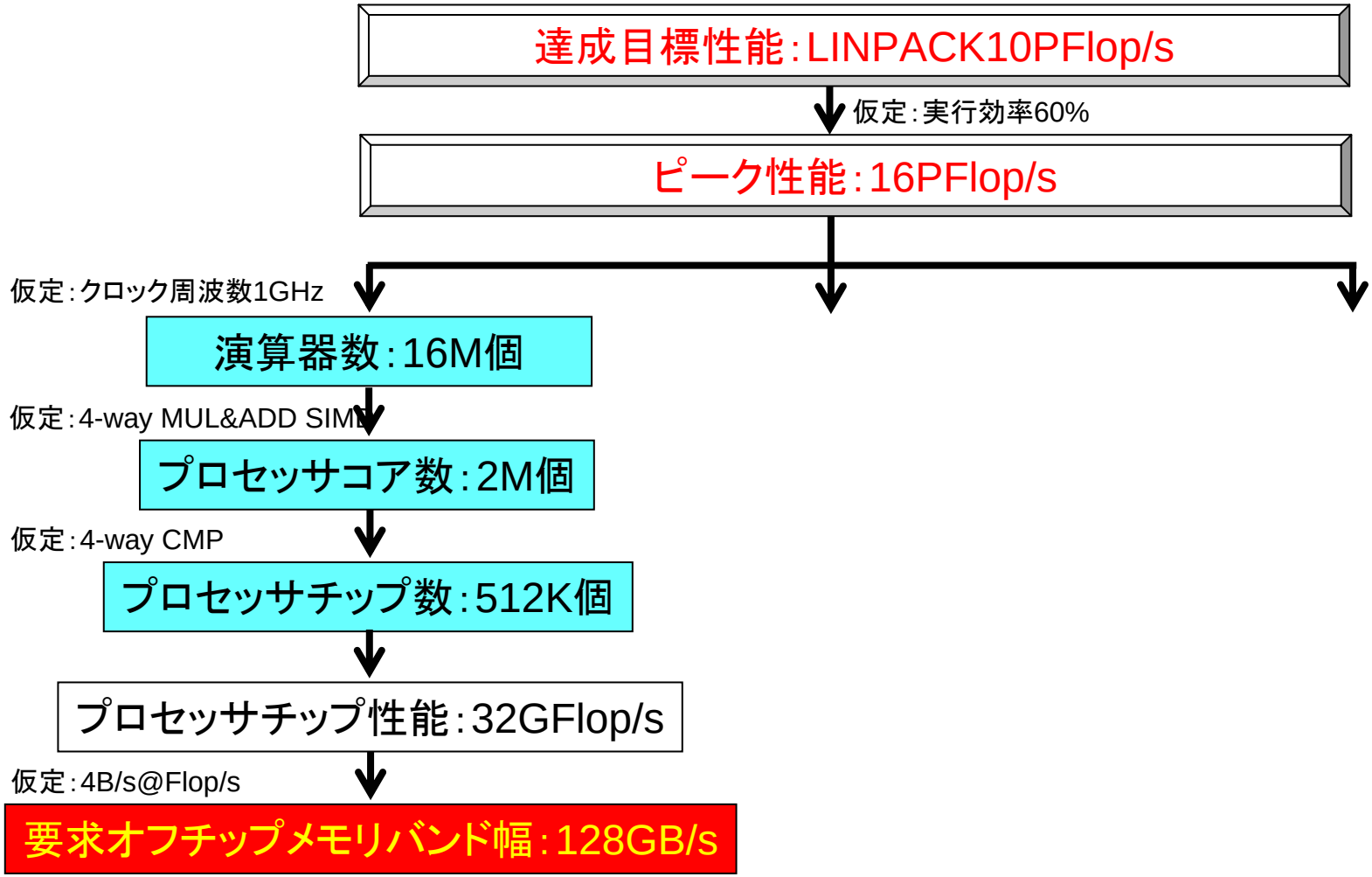
次世代スパコン
～達成目標と制約条件の整理～

- 達成目標
- 制約条件
  - Scalability: NLSとしての存在のみならず、NISへの下方展開が可能なこと
  - Productivity: ソフトウェアの生産性が高いこと
  - Compatibility: 過去のソフトウェア資産からの、ならびに、将来のソフトウェア開発への連続性、継続性を保証すること
  - Dependability: 長期間連続運転可能なこと
  - Serviceability: センター運用可能なこと
  - Ecology: 低消費電力
  - Economy: 将来にわたってビジネス的に持続的に開発可能であること

10PFlop/sのマシンをどう作るか？



10PFlop/sのマシンをどう作るか？

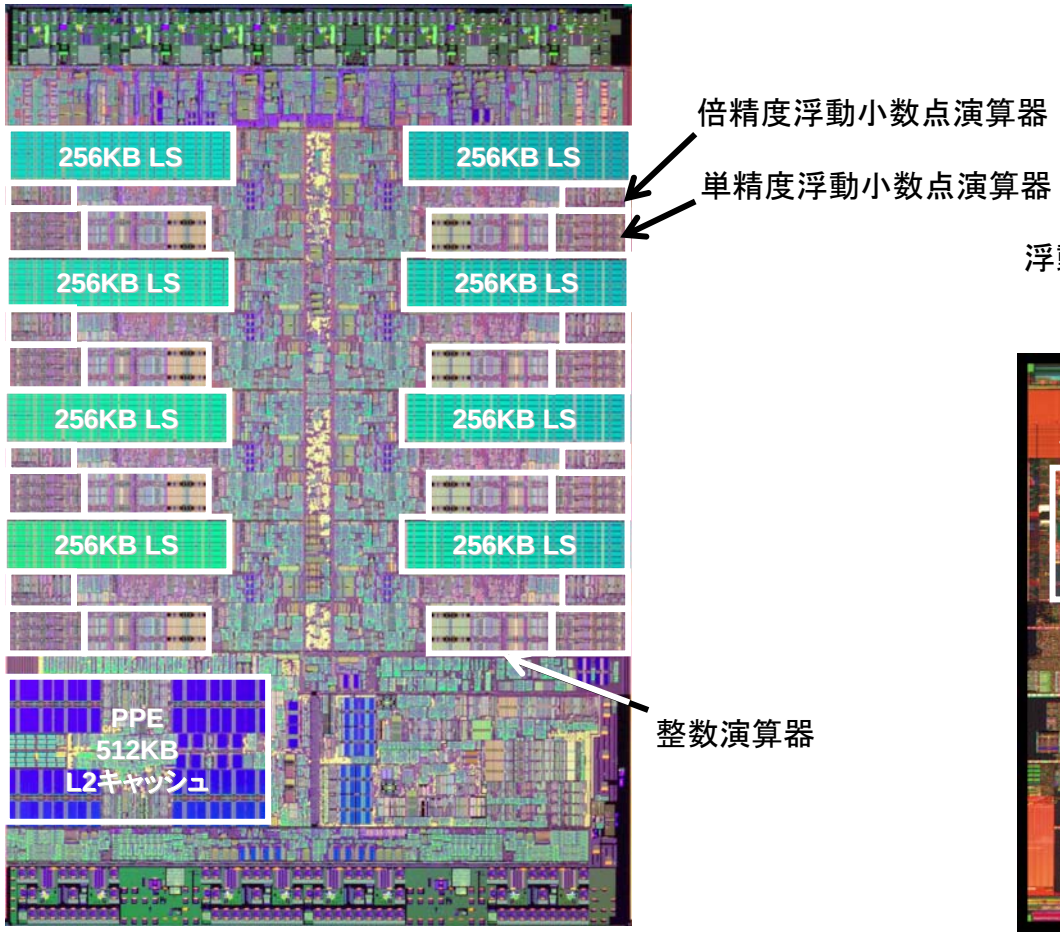


現在のプロセッサチップは押し並べて・・・

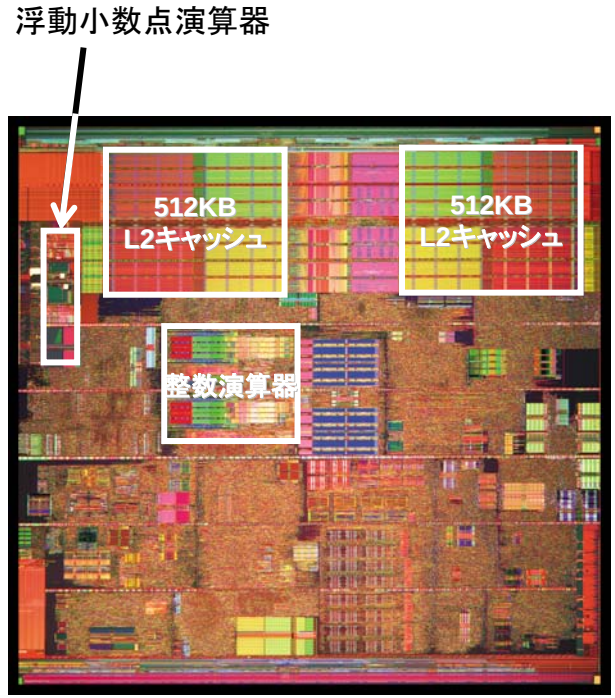
i-way superscalar processor with *j*-way SIMD × *k*-way CMP

	i	*j*	*k*	クロック周波数	プロセッサチップ性能	オフチップメモリバンド幅
Intel Pentium	3	4	2	4GHz	64GFlop/s	
IBM BG/L	2	4	2	800MHz	12.8GFlop/s	
Cell	2	4	8	4GHz	256GFlop/s	25.6GB/s
ClearSpeed	1	96	1	200MHz	38.4GFlop/s	

現在のプロセッサチップは押し並べて・・・



Cell (221mm^2 @90nm)



Pentium4 (112mm^2 @90nm)

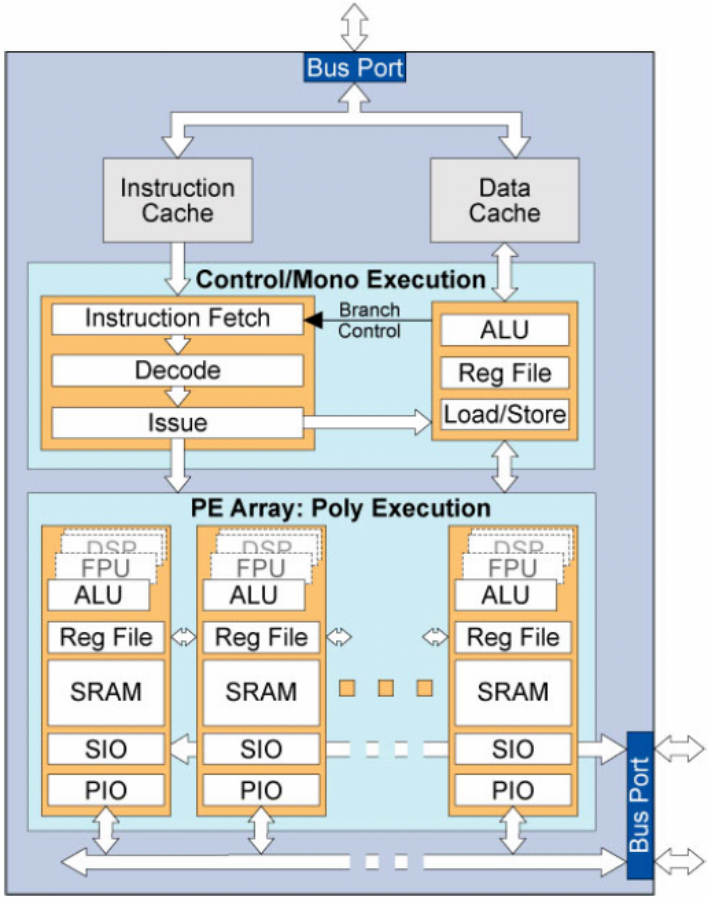
写真:SONY提供

写真:Intel提供

CellとPentium4の比較

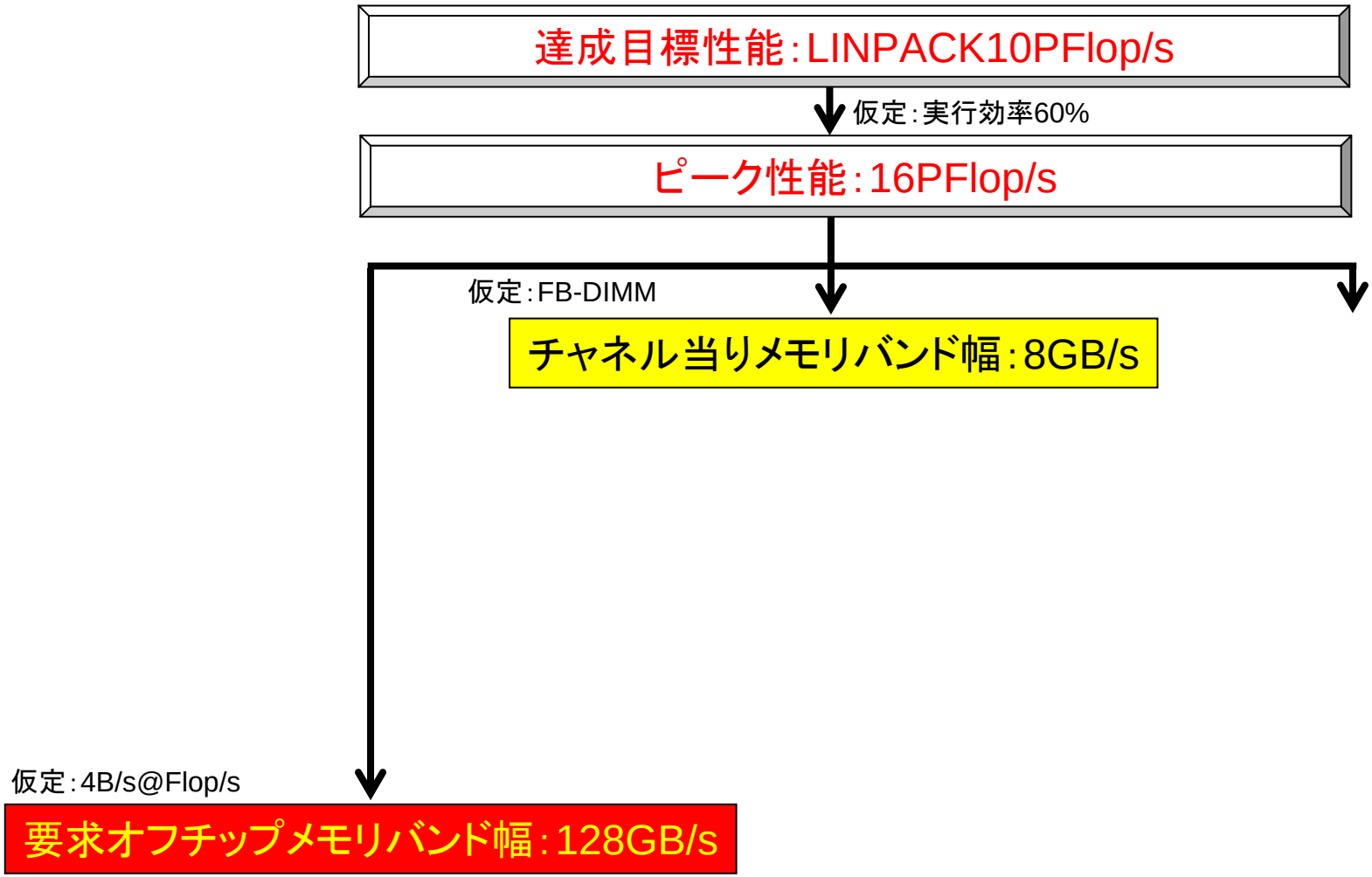
		CELL	Pentium4 (Prescott)
製造技術		90nm SOI	90nm
トランジスタ数		2億3400万	1億2500万
面積		221mm^2	112mm^2
動作周波数		4.0GHz（最大4.6GHz）	3.8GHz
ピーク性能		256GFlop/s（SPE×8個）	15.2GFlop/s
消費電力		48W（6W×8SPE）＋α	103W
プロセッサコア		9個	1個
	メインプロセッサ	PPE（Powerベース＋VMX）×1個	IA32（x86）ベース（＋SSE3）
	演算プロセッサ	SPE（SIMDプロセッサ）×8個	―
内部メモリ			
	メインプロセッサ	•L1命令キャッシュ：32KB •L1データキャッシュ：32KB •L2キャッシュ：512KB	•L1命令キャッシュ：12Kマイクロ命令 •L1データキャッシュ：16KB •L2キャッシュ：1MB
	演算プロセッサ	•各SPEのLS：256KB	―
チップ内インターコネクト（オンチップバス）		EIB：192Gバイト/秒（（128ビット＋64ビット）×2GHz×4リング）	―
外付けDRAMインタフェース		XIO：25.6Gバイト/秒（32ビット×3.2GHz×2チャネル）	FSB：6.4Gバイト/秒（64ビット×800MHz）
入出力インタフェース		FlexIO：76.8Gバイト/秒（6.4Gビット/秒×8ビット×12トランシーバ/レシーバ）	

Multi-Threaded Array Processing Architecture

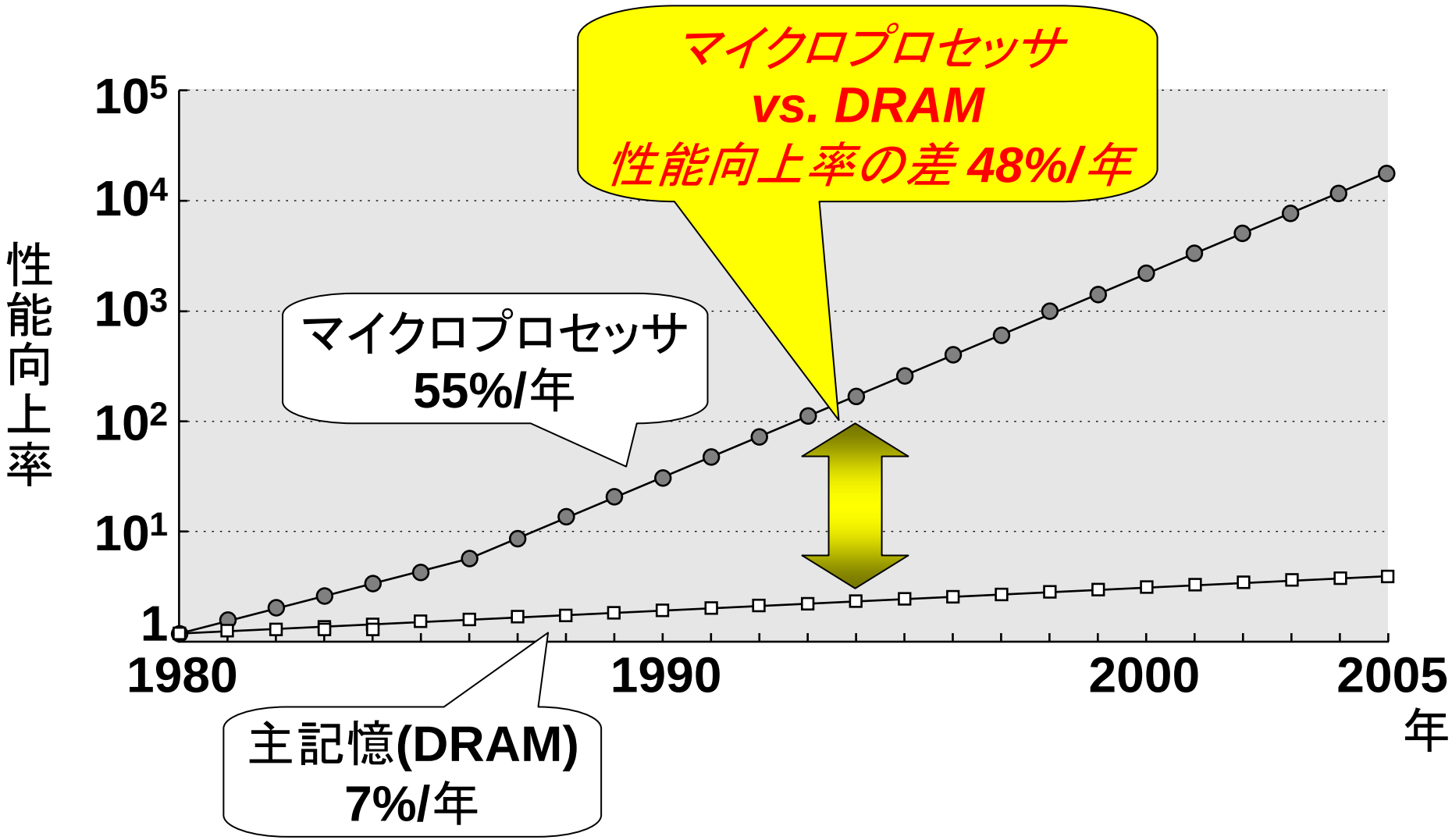


- Multi-Threaded Array Processor
  - Fully programmable in C
  - Hardware multi-threading
  - Extensible instruction set
- Scalable internal parallelism
  - Array of processing elements (PEs)
  - Compute, bandwidth scale together
  - From 10s to 1,000s of PEs
  - Built-in PE redundancy
- High performance, low power
  - ~10 GFLOPS/watt
- Multiple high-speed I/O channels

10PFlop/sのマシンをどう作るか？



「Memory Wall」問題



【出典】J.L. Hennessy, D.A. Patterson, *Computer Architecture: A Quantitative Approach 3rd Edition*, Fig. 5.2

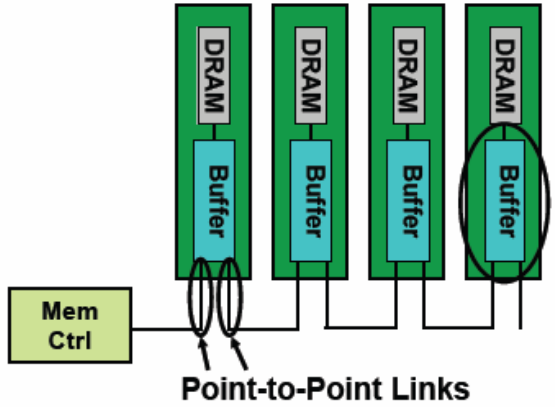
DDR/DDR2/DDR3 SDRAM

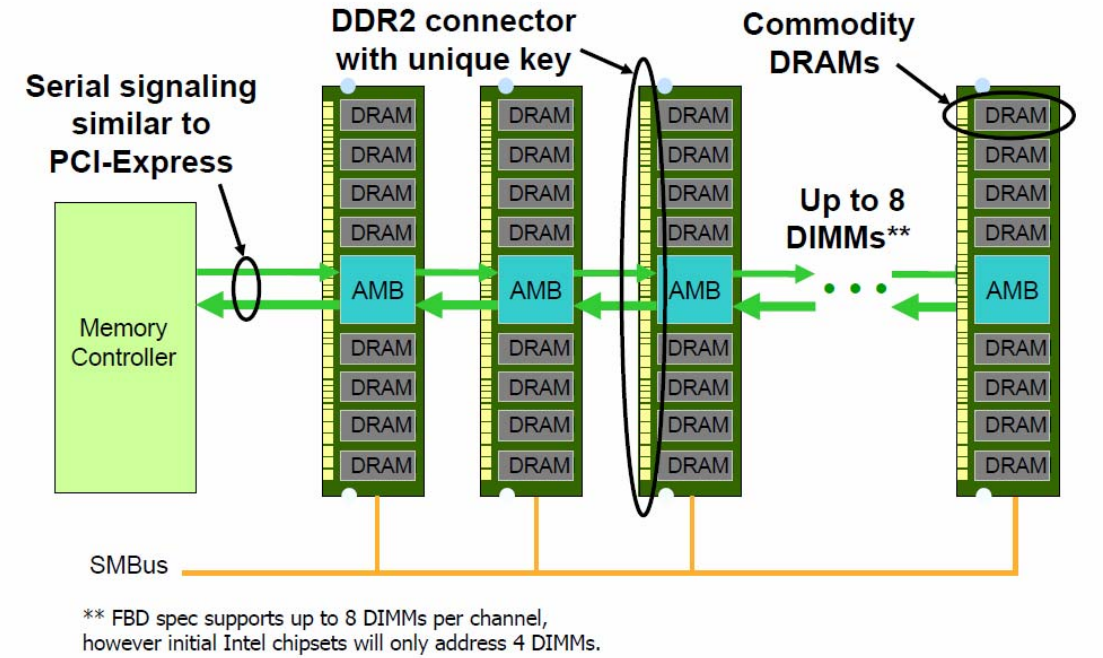
項　目	DDR3 SDRAM	DDR2 SDRAM	DDR SDRAM
クロック周波数	533/667MHz	200/266/333/400MHz	100/133/166/200MHz
データ転送速度	1066/1333Mbps	400/533/667/800Mbps	200/266/333/400Mbps
I/O幅	x4/x8/x16	x4/x8/x16	x4/x8/x16/x32
プリフェッチ	8ビット	4ビット	2ビット
クロック入力	ディファレンシャルクロック	ディファレンシャルクロック	ディファレンシャルクロック
バースト長	4, 8, 4（バーストチョップ）	4, 8	2, 4, 8
データストローブ形式	ディファレンシャルデータストローブ	ディファレンシャルデータストローブ	シングルデータストローブ
電源電圧	1.5V	1.8V	2.5V
I/Oインタフェース	SSTL_15	SSTL_18	SSTL_2
/CASレーテンシ（CL）	5, 6, 7, 8, 9, 10クロック	3, 4, 5クロック	2, 2.5, 3クロック
On die termination（ODT）	対　応	対　応	非対応
単体パッケージ	FBGA	FBGA	TSOP(II)/FBGA/LQFP
鉛フリー	対　応	対　応	対　応

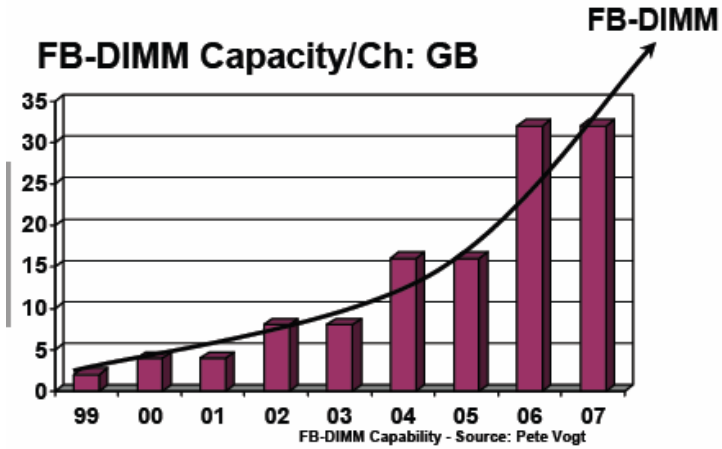
http://www.elpida.com/pdfs/J0876E10.pdf

FB-DIMM

- アドレスやデータ等のバッファリング
- P2P通信
- 総メモリ容量のスケーリングを可能に！（最大8DIMM）







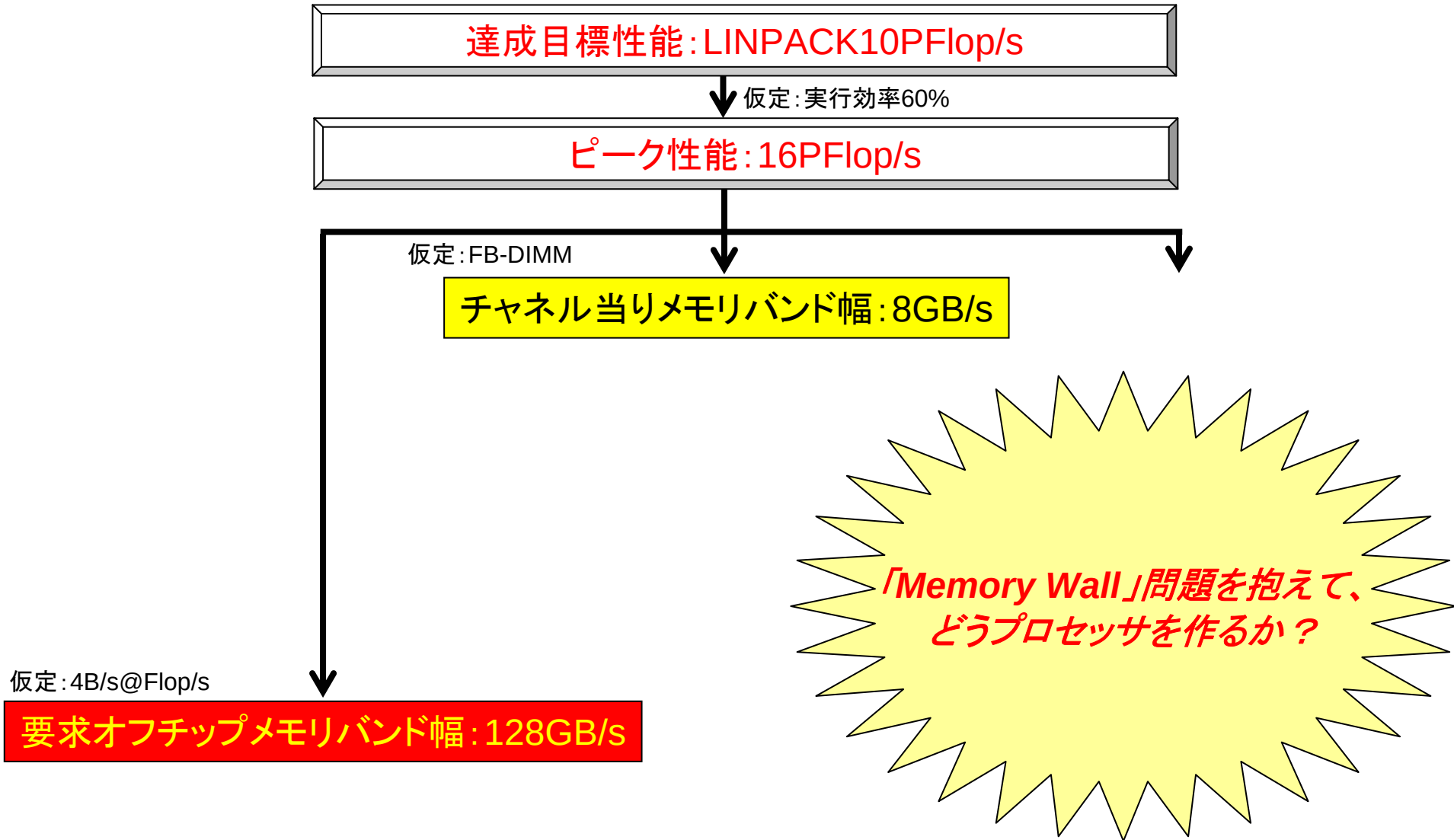
FB-DIMM製品例（エルピーダ）

- 2005年8月サンプル出荷

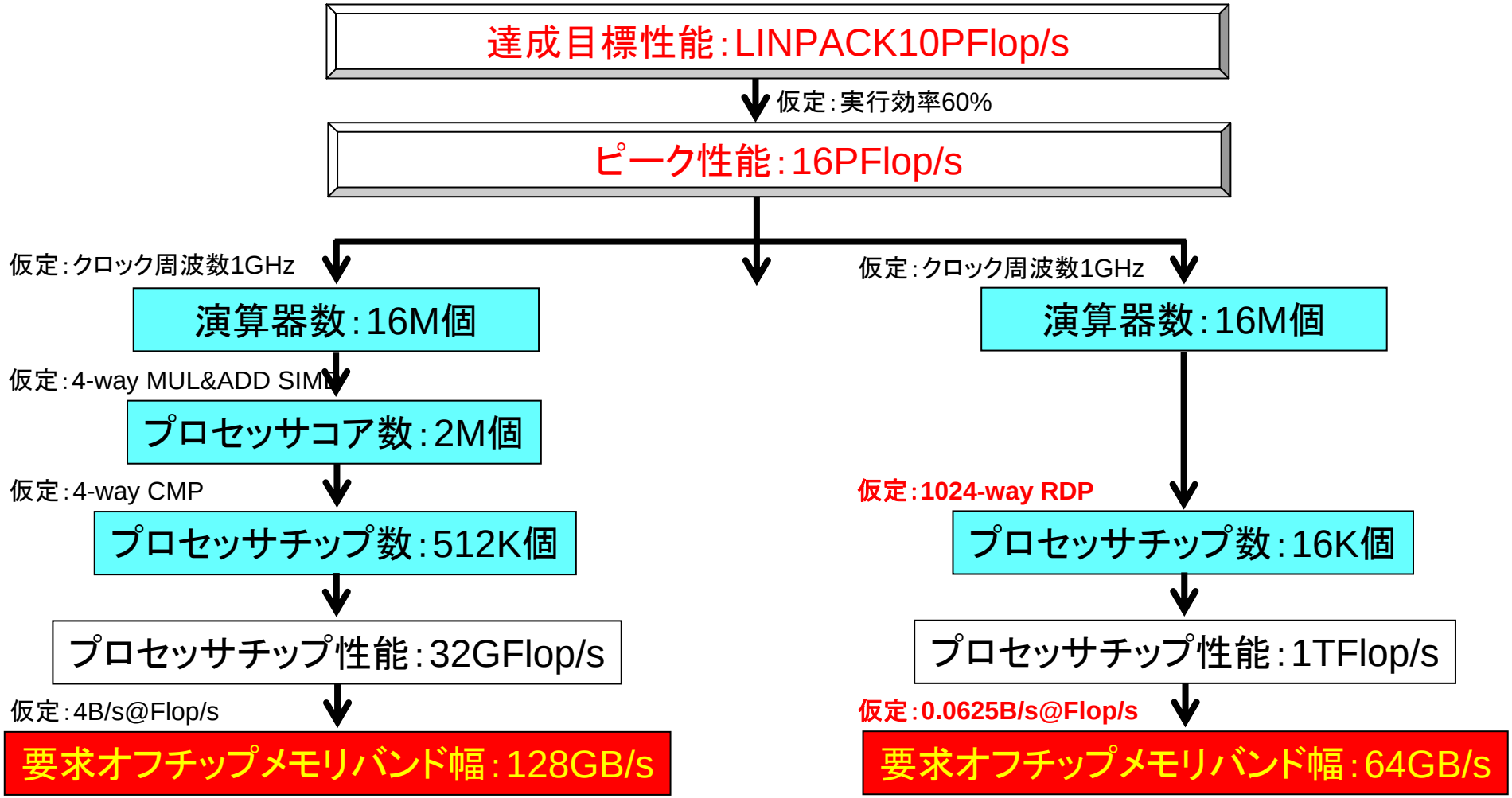
4GB FB-DIMM　主な仕様

Density	4GB	
Organization	256M words x 72-bits x 2 ranks	
Module Speed Grade	PC2-4200F	PC2-5300F
DRAM Speed Grade	DDR2-533 (533Mbps)	DDR2-667 (667Mbps)
FB-DIMM Channel Peak Throughput	6.4GB/s	8.0GB/s
Mounted Devices	1Gb DDR2 SDRAM x 36 pieces	
Burst Length	4 or 8	
Supply Voltage	1.8V±0.1V	
Package	240-pin Fully Buffered DIMM 133.35mm x 30.35mm Thin:6.7mm (max.)	

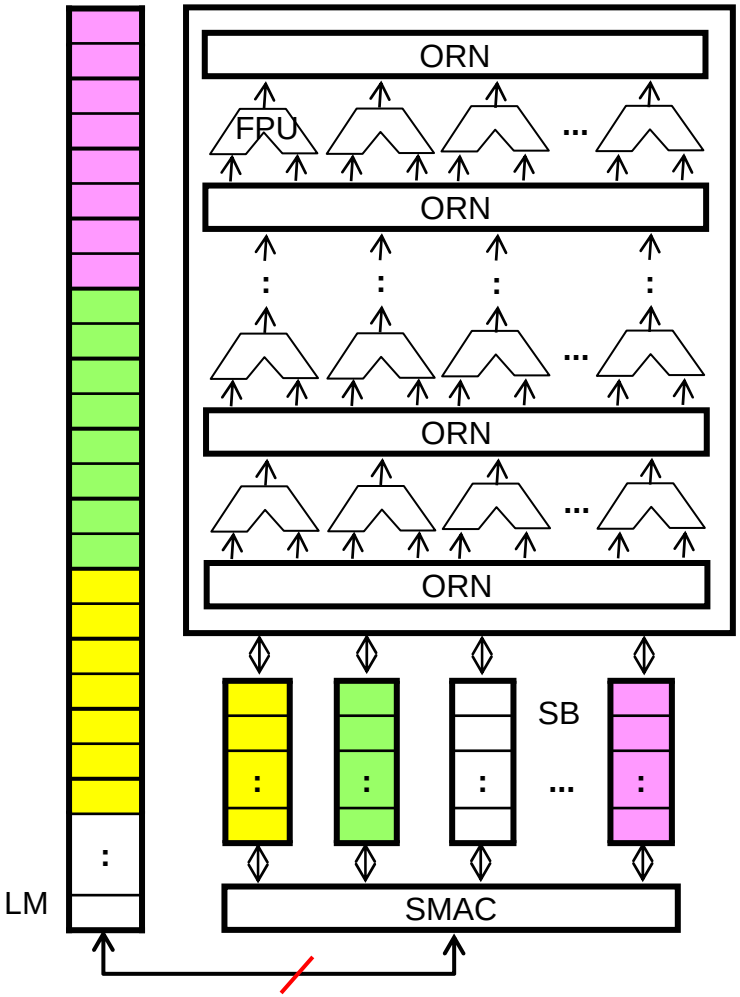
10PFlop/sのマシンをどう作るか？



10PFlop/sのマシンをどう作るか？



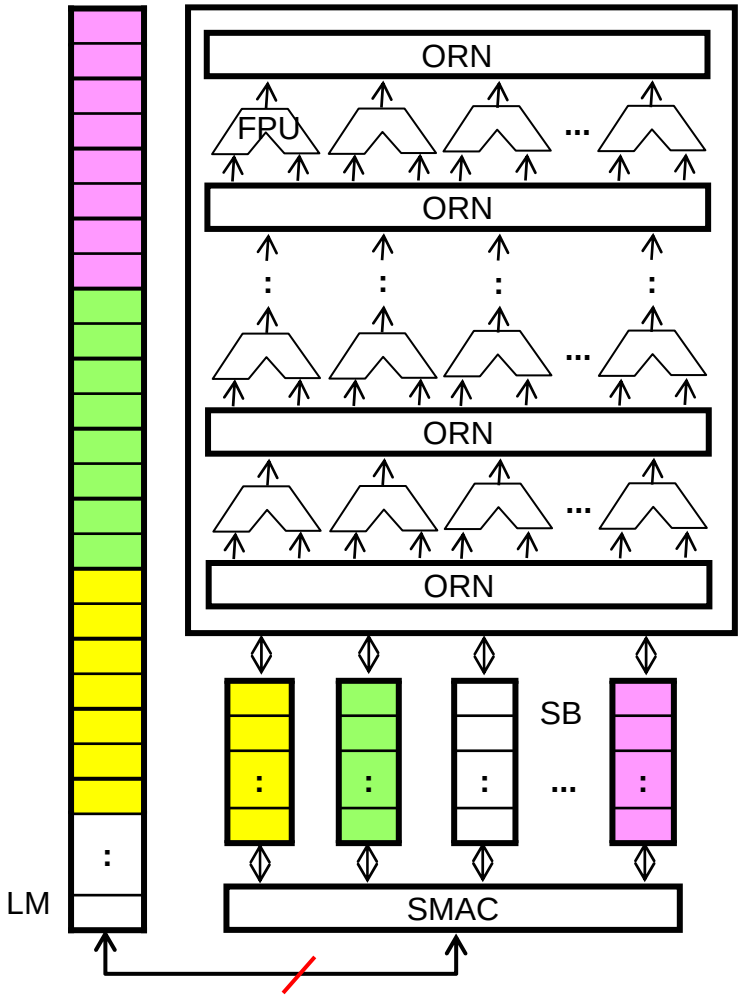
提案：再構成可能大規模データパス（RDP: Reconfigurable Datapath）



- RDPとは？
  - 多数の演算器（FPU: Floating-Point Unit）とそれらを相互接続する網（ORN: Operand Routing Network）を搭載し、
    - FPUで行う演算内容
    - ORN上のFPU間接続関係

    を動的に再構成可能としたデータパス

提案：再構成可能大規模データパス（RDP: Reconfigurable Datapath）

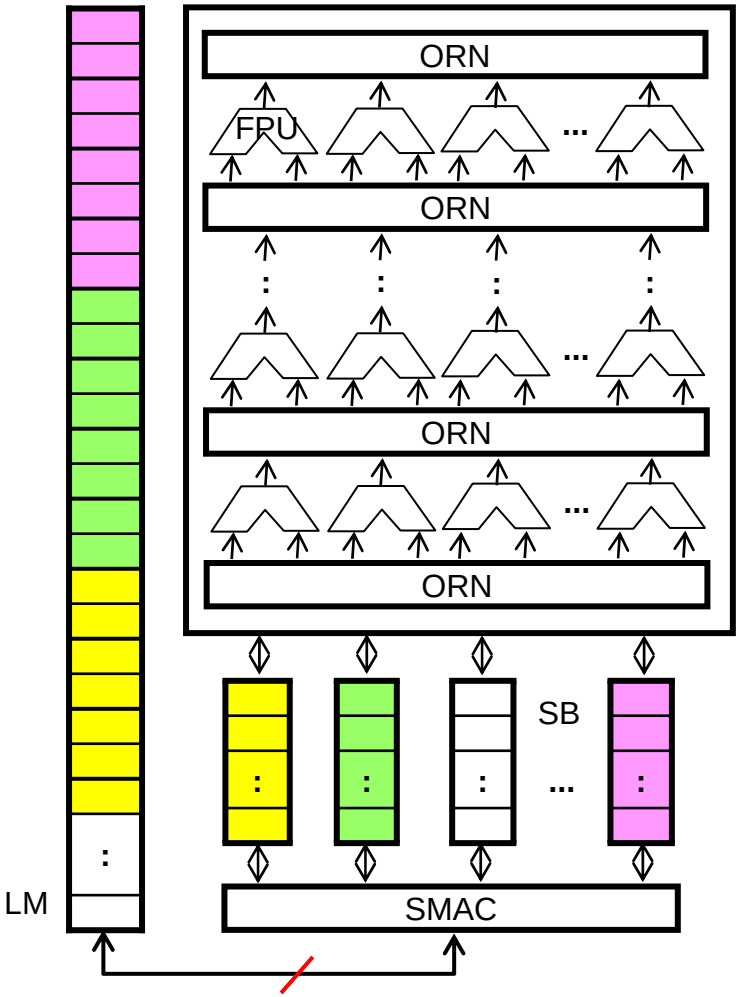


- 趣旨は？
  - トランジスタ資源を（従来のプロセッサのように）、
    - データ並列性の活用（SIMD、ベクトルプロセッサ）
    - 命令レベル並列性の活用（OOOスーパースカラ）
    - プロセスレベル並列性の活用（CMP）

    に投資するのではなく、
    - データ依存性の維持（ORN上でのデータ転送）
    - イタレーション間並列性の活用（複数のイタレーションをパイプライン処理）

    に投資することで実効性能を確保！

提案:再構成可能大規模データパス（RDP: Reconfigurable Datapath）



- 用途は？
  - 主プロセッサに対するコプロセッサ
  - コア計算部のループボディのデータフロー全体を直接マッピング
    - ORNでデータ依存関係を維持
    - 原理的には毎クロックサイクル、新しいイタレーションを実行開始可能→複数イタレーションのパイプライン処理

RDPの応用例

～分子軌道法における二電子積分計算 $(\mu\nu\|\lambda\sigma)$ の場合～

tei(4,4,4,4)=(((3+2*p*(4*PAx*PBx+PBx**2+PAx**2*(1+2*p*PBx**2)))*(3+2*q*(4*QCx*QDx+QDx**2+QCx**2*(1+2*q*QDx**2)))*f(0,t))/(p**2*q**2)+(4*(3+2*p*(4*PAx*PBx+PBx**2+PAx**2*(1+2*p*PBx**2)))*PQx*(QCx+QDx)*(3+2*q*QCx*QDx)*f(1,t))/(p*q*(p+q))(4*(PAx+PBx)*(3+2*p*PAx*PBx)*PQx*(3+2*q*(4*QCx*QDx+QDx**2+QCx**2*(1+2*q*QDx**2)))*f(1,t))/(p*q*(p+q))(8*(PAx+PBx)*(3+2*p*PAx*PBx)*(QCx+QDx)*(3+2*q*QCx*QDx)*(((p+q)*f(1,t))+2*p*PQx**2*q*f(2,t)))/(p*q*(p+q)**2)+(2*(3+2*p*(4*PAx*PBx+PBx**2+PAx**2*(1+2*p*PBx**2)))*(3+q*(QCx**2+4*QCx*QDx+QDx**2))*(((p+q)*f(1,t))+2*p*PQx**2*q*f(2,t)))/(p*q**2*(p+q)**2)+(2*(3+p*(PAx**2+4*PAx*PBx+PBx**2))*(3+2*q*(4*QCx*QDx+QDx**2+QCx**2*(1+2*q*QDx**2)))*(((p+q)*f(1,t))+2*p*PQx**2*q*f(2,t)))/(p**2*q*(p+q)**2)+(4*(3+2*p*(4*PAx*PBx+PBx**2+PAx**2*(1+2*p*PBx**2)))*PQx*(QCx+QDx)*(3*(p+q)*f(2,t)+2*p*PQx**2*q*f(3,t)))/(q*(p+q)**3)¥+(8*(3+p*(PAx**2+4*PAx*PBx+PBx**2))*PQx*(QCx+QDx)*(3+2*q*QCx*QDx)*(3*(p+q)*f(2,t)+2*p*PQx**2*q*f(3,t)))/(p*(p+q)**3)(8*(PAx+PBx)*(3+2*p*PAx*PBx)*PQx*(3+q*(QCx**2+4*QCx*QDx+QDx**2))*(3*(p+q)*f(2,t)+2*p*PQx**2*q*f(3,t)))/(q*(p+q)**3)(4*(PAx+PBx)*PQx*(3+2*q*(4*QCx*QDx+QDx**2+QCx**2*(1+2*q*QDx**2)))*(3*(p+q)*f(2,t)+2*p*PQx**2*q*f(3,t)))/(p*(p+q)**3)+((3+2*p*(4*PAx*PBx+PBx**2+PAx**2*(1+2*p*PBx**2)))*(3*(p+q)**2*f(2,t)+4*p*PQx**2*q*(3*(p+q)*f(3,t)+p*PQx**2*q*f(4,t)))))/(q**2*(p+q)**4)(8*(PAx+PBx)*(3+2*p*PAx*PBx)*(QCx+QDx)*(3*(p+q)**2*f(2,t)+4*p*PQx**2*q*(3*(p+q)*f(3,t)+p*PQx**2*q*f(4,t)))))/(q*(p+q)**4)(8*(PAx+PBx)*(QCx+QDx)*(3+2*q*QCx*QDx)*(3*(p+q)**2*f(2,t)+4*p*PQx**2*q*(3*(p+q)*f(3,t)+p*PQx**2*q*f(4,t)))))/(p*(p+q)**4)+(4*(3+p*(PAx**2+4*PAx*PBx+PBx**2))*(3+q*(QCx**2+4*QCx*QDx+QDx**2))*(3*(p+q)**2*f(2,t)+4*p*PQx**2*q*(3*(p+q)*f(3,t)+p*PQx**2*q*f(4,t)))))/(p*q*(p+q)**4)+((3+2*q*(4*QCx*QDx+QDx**2+QCx**2*(1+2*q*QDx**2)))*(3*(p+q)**2*f(2,t)+4*p*PQx**2*q*(3*(p+q)*f(3,t)+p*PQx**2*q*f(4,t)))))/(p**2*(p+q)**4)(4*p*(PAx+PBx)*(3+2*p*PAx*PBx)*PQx*(15*(p+q)**2*f(3,t)+4*p*PQx**2*q*(5*(p+q)*f(4,t)+p*PQx**2*q*f(5,t)))))/(q*(p+q)**5)+(8*(3+p*(PAx**2+4*PAx*PBx+PBx**2))*PQx*(QCx+QDx)*(15*(p+q)**2*f(3,t)+4*p*PQx**2*q*(5*(p+q)*f(4,t)+p*PQx**2*q*f(5,t)))))/(p+q)**5+(4*PQx*q*(QCx+QDx)*(3+2*q*QCx*QDx)*(15*(p+q)**2*f(3,t)+4*p*PQx**2*q*(5*(p+q)*f(4,t)+p*PQx**2*q*f(5,t)))))/(p*(p+q)**5)(8*(PAx+PBx)*PQx*(3+q*(QCx**2+4*QCx*QDx+QDx**2))*(15*(p+q)**2*f(3,t)+4*p*PQx**2*q*(5*(p+q)*f(4,t)+p*PQx**2*q*f(5,t)))))/(p+q)**5+(8*(PAx+PBx)*(QCx+QDx)*(15*(p+q)**3*f(3,t)+30*p*PQx**2*q*(p+q)*(3*(p+q)*f(4,t)+2*p*PQx**2*q*f(5,t))8*p**3*PQx**6*q**3*f(6,t)))/(p+q)**6+(2*(3+p*(PAx**2+4*PAx*PBx+PBx**2))*(15*(p+q)**3*f(3,t)30*p*PQx**2*q*(p+q)*(3*(p+q)*f(4,t)+2*p*PQx**2*q*f(5,t))+8*p**3*PQx**6*q**3*f(6,t)))/(q*(p+q)**6)+(2*(3+q*(QCx**2+4*QCx*QDx+QDx**2))*(15*(p+q)**3*f(3,t)30*p*PQx**2*q*(p+q)*(3*(p+q)*f(4,t)+2*p*PQx**2*q*f(5,t))+8*p**3*PQx**6*q**3*f(6,t)))/(p*(p+q)**6)

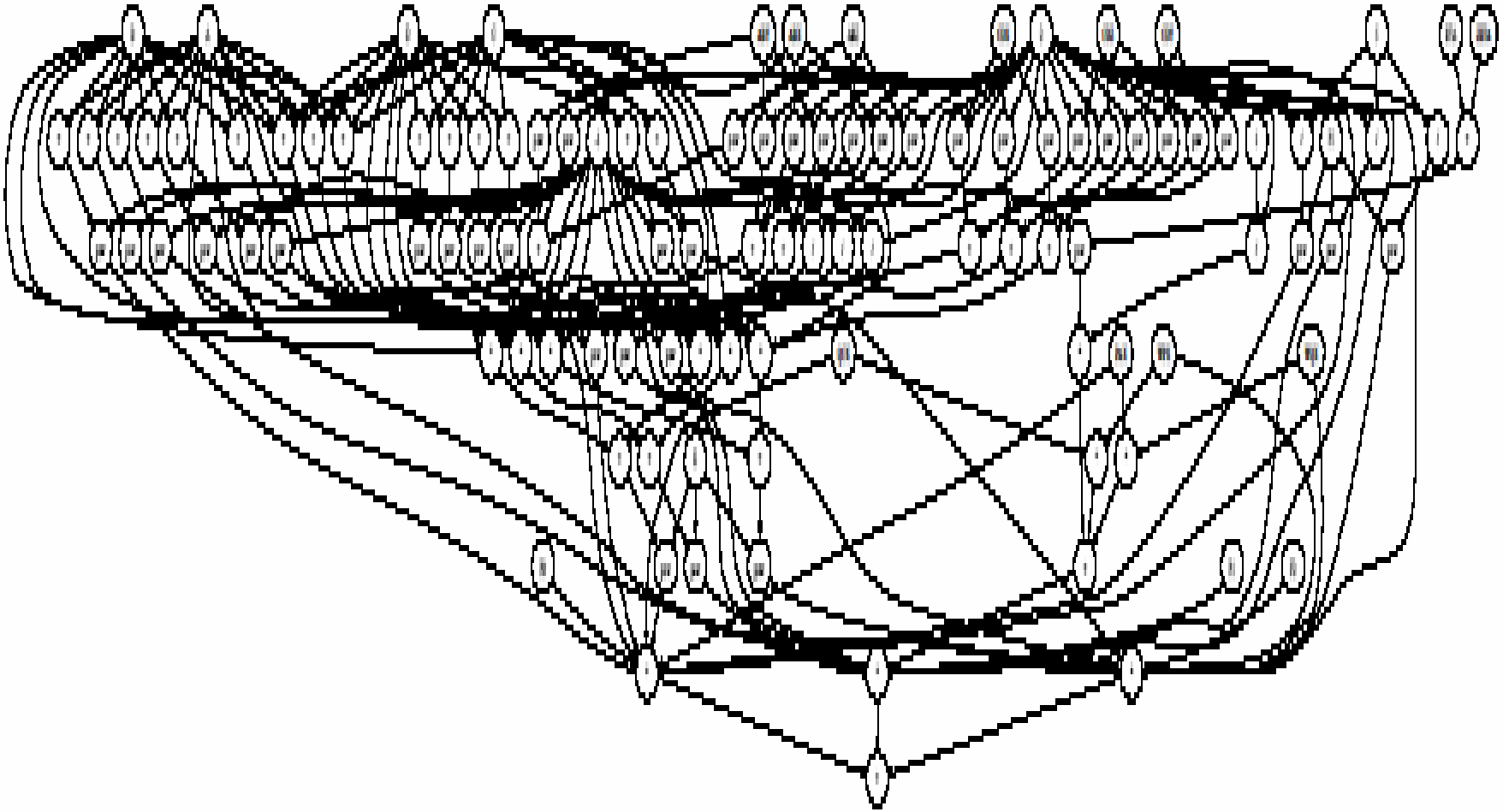
➔ **787 MUL, 261 ADD, 69 FUNC**

tei(3,1,1,1)=((PAy*(1+2*p*PAx*PBx)*(1+2*q*QCx*QDx)*f(0,t))/q+((((p+q)**4*((PAy+PQy)*q*(1+2*q*QCx*QDx)+p*(PAy+2*PAx*PAy*PQx*q+2*PAy*PBx*PQx*q+2*PAx*PBx*PQy*q2*PAx*PAy*q*QCx2*PAy*PBx*q*QCx2*PAy*PQx*q*QCx+2*q*((PAy*(PAx+PBx+PQx))+2*(PAy*(PAx+PBx)*PQx+PAx*PBx*PQy)*q*QCx)*QDx)2*p**2*PAx*PAy*PBx*(1+2*PQx*q*(QCx+QDx)))*f(1,t))/q)+(p+q)*((p+q)*((p+q)*(3*p*PAy+6*p**2*PAx*PAy*PQx+6*p**2*PAy*PBx*PQx+2*p**2*PAy*PQx**2+4*p**3*PAx*PAy*PBx*PQx**2+p*PQy+2*p**2*PAx*PBx*PQy+2*p*PAy*PQx**2*q+PQy*q+2*p*PAx*PQx*PQy*q

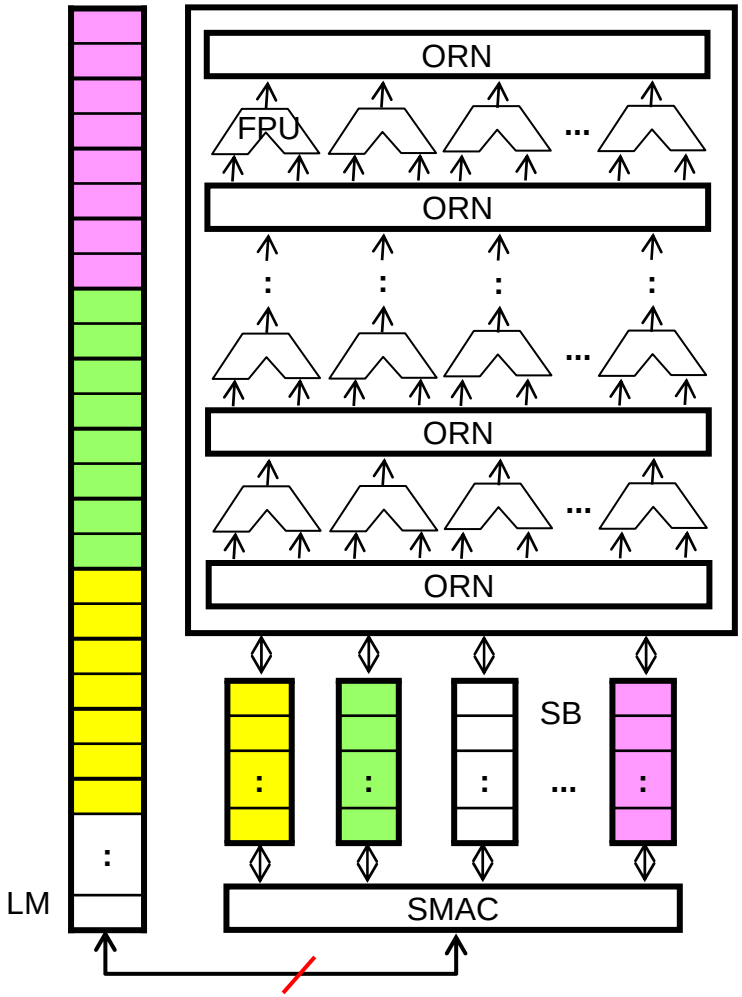
➔ **116 MUL, 31 ADD, 2 FUNC**

【出典：青柳 睦 教授（九大）】

RDPにマッピングするデータフローの例 ～(ps,ps) 型積分計算の場合～

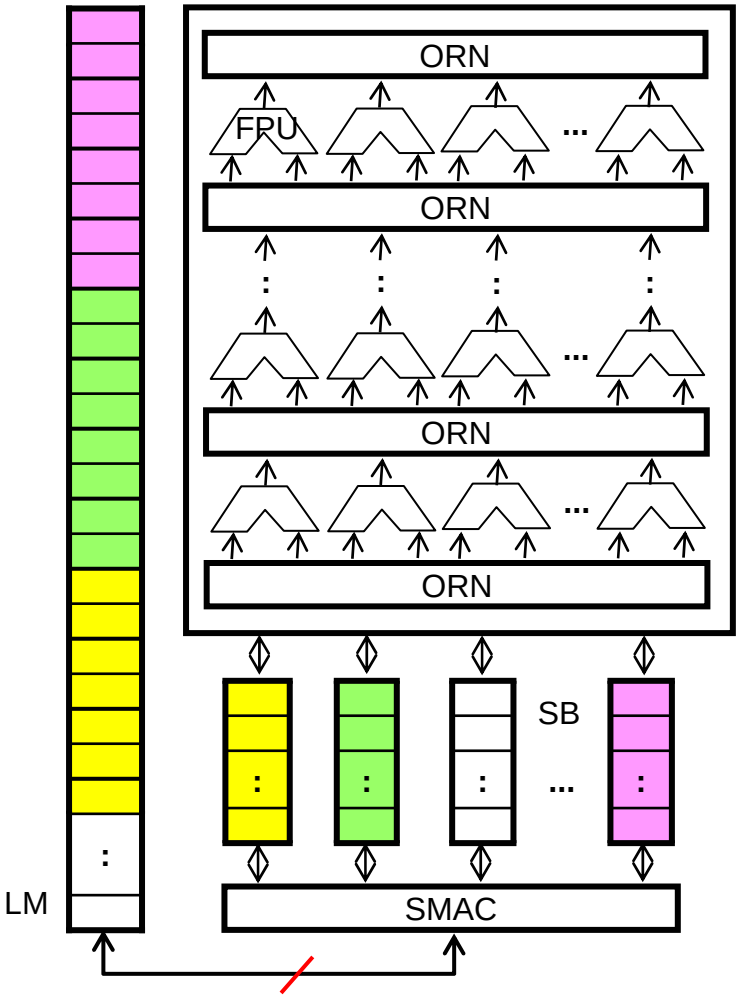


提案：再構成可能大規模データパス（RDP: Reconfigurable Datapath）



- 用途は？
  - 主プロセッサに対するコプロセッサ
  - コア計算部のループボディのデータフロー全体を直接マッピング
    - ORNでデータ依存関係を維持
    - 原理的には毎クロックサイクル、新しいイタレーションを実行開始可能
- 効能は？
  - ***「必要とするメモリアクセス回数」を大幅に削減！***
    - 従来プロセッサは、潜在的に「1演算につき3回のメモリアクセス」が必要

提案：再構成可能大規模データパス（RDP: Reconfigurable Datapath）



- 影響は？
  - プログラミングモデル：影響なし
  - コンパイラ最適化：
    - ループ・コラプシング（loop collapsing）：複数のループをまとめて、ループボディの大きなループを作る
- 継続性、経済的効果は？
  - 組込みシステム、SoC業界との間での技術移転が可能
    - 例）IP FLEX DAP/DNA
    - 例）Stretch

IP FLEX DAP/DNA2

DAP
RISCコア

DNA

DNA
ダイレクト
I/O

高速
I/O機器

高速バス･スイッチ･ユニット

PCI
インタフェース

SDRAM
インタフェース

GPIO、UART、
Interrupt Controller 、他

PCIバス

外部メモリ

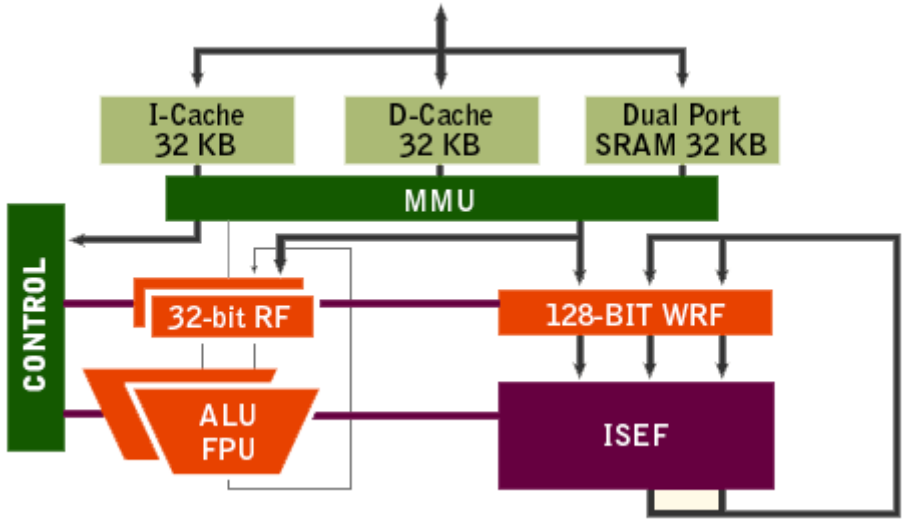
周辺機器

多彩な
32bit
演算ユニット

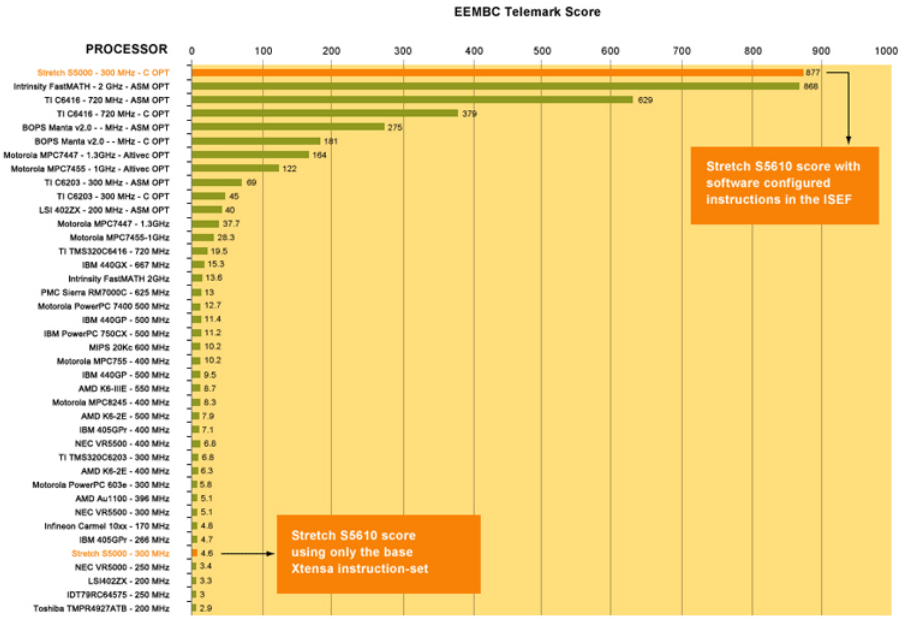
DNA-Matrix

PE

Stretch



- 300 MHz, 32-bit Xtensa-based processor
- 16- and 24-bit instructions
- FPU
- MMU with TLB
- Stretch Instruction Set Extension Fabric
  - Aligned load and store
    - 8, 16, 32, 64, and 128 bit
  - Unaligned load and store
    - Up to 16 bytes variable byte streaming I/O
    - Up to 32 bits variable bit streaming I/O
- User-defined extensions to the core ISA
  - Defined in C/C++
  - Fully pipelined and interlocked
- Low power consumption
- Support for standard operating systems



10PFlop/sのマシンをどう作るか？