

九州大学を中心としたアーキテクチャ研究の現状： SoCから次世代スパコンまで

井上, 弘士
九州大学大学院システム情報科学研究院

<https://hdl.handle.net/2324/9133>

出版情報：SLRC プレゼンテーション, 2006-07-19. 九州大学システムLSI研究センター
バージョン：
権利関係：

九州大学を中心としたアーキテクチャ
研究の現状
～SoCから次世代スパコンまで～

九州大学大学院システム情報科学研究所

井上弘士(いのうえこうじ)

inoue@i.kyushu-u.ac.jp

福岡と言えば・・・アーキテクチャ研究

福岡システムLSI総合開発センター



新キャンパス(伊都キャンパス)



研究メンバー：総勢47名

● 九州大学

- 村上・井上研究室 (<http://www.c.csce.kyushu-u.ac.jp/SOC/index.html>)
 - 教員：村上教授，井上助教授，馬場助手
 - 学生：D10人(3人は社会人)，M12人，B5人
- システムLSI研究センター (<http://www.slrc.kyushu-u.ac.jp/index.html>)
 - 教員：佐藤教授，林田助手
- 情報基盤センター (<http://www.cc.kyushu-u.ac.jp/>)
 - 教員：青柳教授，南里助教授
 - PSIラボ：本田研究員，稲富研究員，于研究員，Gu研究員

● 福岡県産業・科学技術振興財団 (<http://www.ist.or.jp/>)

- 薄田研究員
- 福岡知的クラスター研究所 (<http://www.fleets.jp/>)
 - 吉松研究員，曾我研究員，吉田研究員，ゴーチェ研究員，フェヘイラ研究員，神戸研究員

● 九州システム情報技術研究所 (<http://www.isit.or.jp/>)

- 杉原研究員，柴村研究員

2006年度の研究テーマ

()内の数字は研究紹介番号

次世代スーパー・コンピュータ/科学技術計算専用計算機		共同研究
次世代スパコン要素技術開発(*1)	PetaFlops級スパコン用インターコネクト技術を開発！(2)	富士通
科学技術計算専用計算機の開発(*2)	高速な二電子積分計算専用計算機を開発！	セイコーエプソン
科学技術計算向けCMPアーキテクチャ	HPC向けCMPアーキテクチャを探求！	某企業
再構成可能大規模データパス	大量の演算器を使ってメモリ・ウォール問題を解決！(10)	理化学研究所
汎用プロセッサ・アーキテクチャ/性能評価		共同研究
高性能/低消費電力プロセッサの開発	メモリ・ウォール問題を解決！(7)	某企業
	CMP向けアーキテクチャを探求！(1)	某企業
	キャッシュメモリの消費電力を削減！(8)	某企業
	動的最適化により性能向上と消費電力削減を達成！	
ディペンダブル・プロセッサ	耐故障性や安全性を向上！(1, 6)	
性能評価環境の構築	高速かつ高精度な性能予測を可能に！(9)	某企業
SoCプラットフォーム/性能評価		共同研究
システム動的最適化技術の開発(*3)	プログラム実行時に自動でシステムを最適化！(3)	某企業
動的再構成可能プロセッサの開発(*3)	超高機能命令を実行可能なプロセッサを実現！(3)	某企業
SoC設計技術	次世代SoC向けテスト/製造/見積り技術を開発！(4)	某企業
次世代アプリケーション		共同研究
ホモロジーサーチ	近似文字列照合を高速に実行！(5)	

(*1) 文部科学省「次世代IT基盤構築のための研究開発」(H17-19)

(*2) 文部科学省「科学技術振興調整費」(H12-16)

(*3) 文部科学省「知的クラスター創成事業」(H14-18)

Challenges!!

A Venn diagram with three overlapping ovals. The top oval is red and labeled 'High Performance'. The bottom-left oval is blue and labeled 'Low Power/Energy'. The bottom-right oval is green and labeled 'Dependable'. The three ovals overlap in a central region.

High Performance

Low Power/Energy

Dependable

A horizontal bar with a gradient from light green on the left to yellow on the right, containing the text 'Design Environment'.

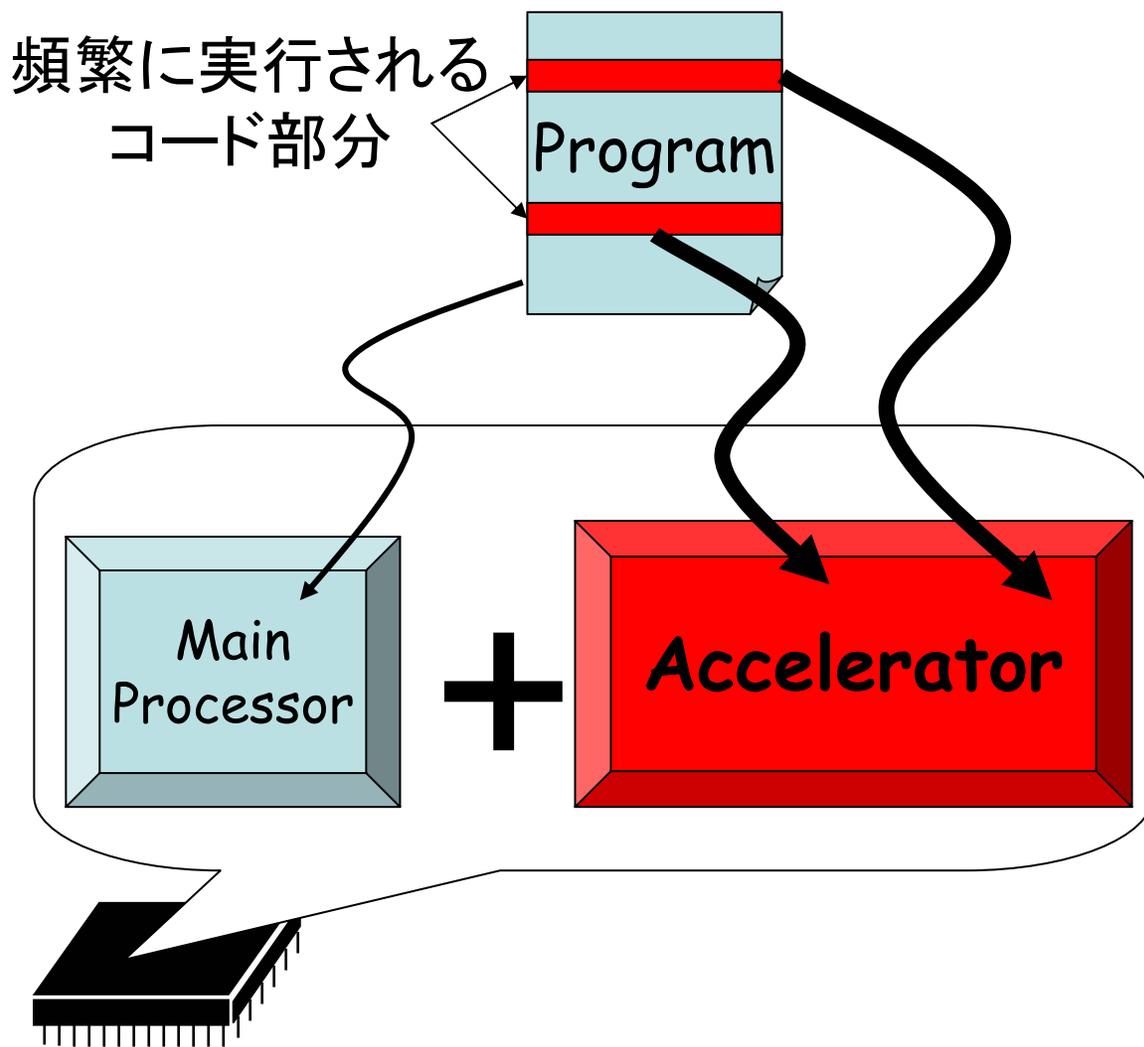
Design Environment

Challenges!!

High Performance

- チップ内並列処理で性能を向上する！
- プログラム実行時に自動チューニング！
- メモリ・ウォールを打ち破る！

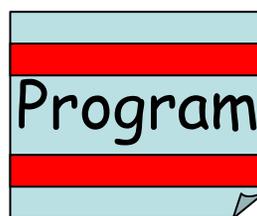
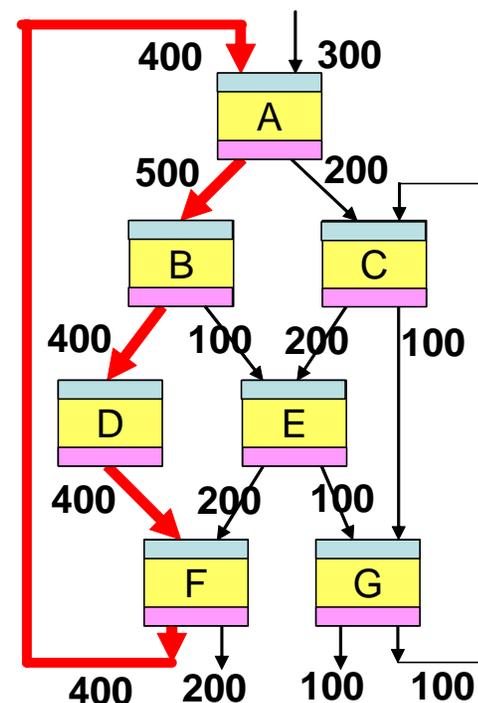
プロセッサ性能を向上する！ ～プログラム実行中に自己チューニング～



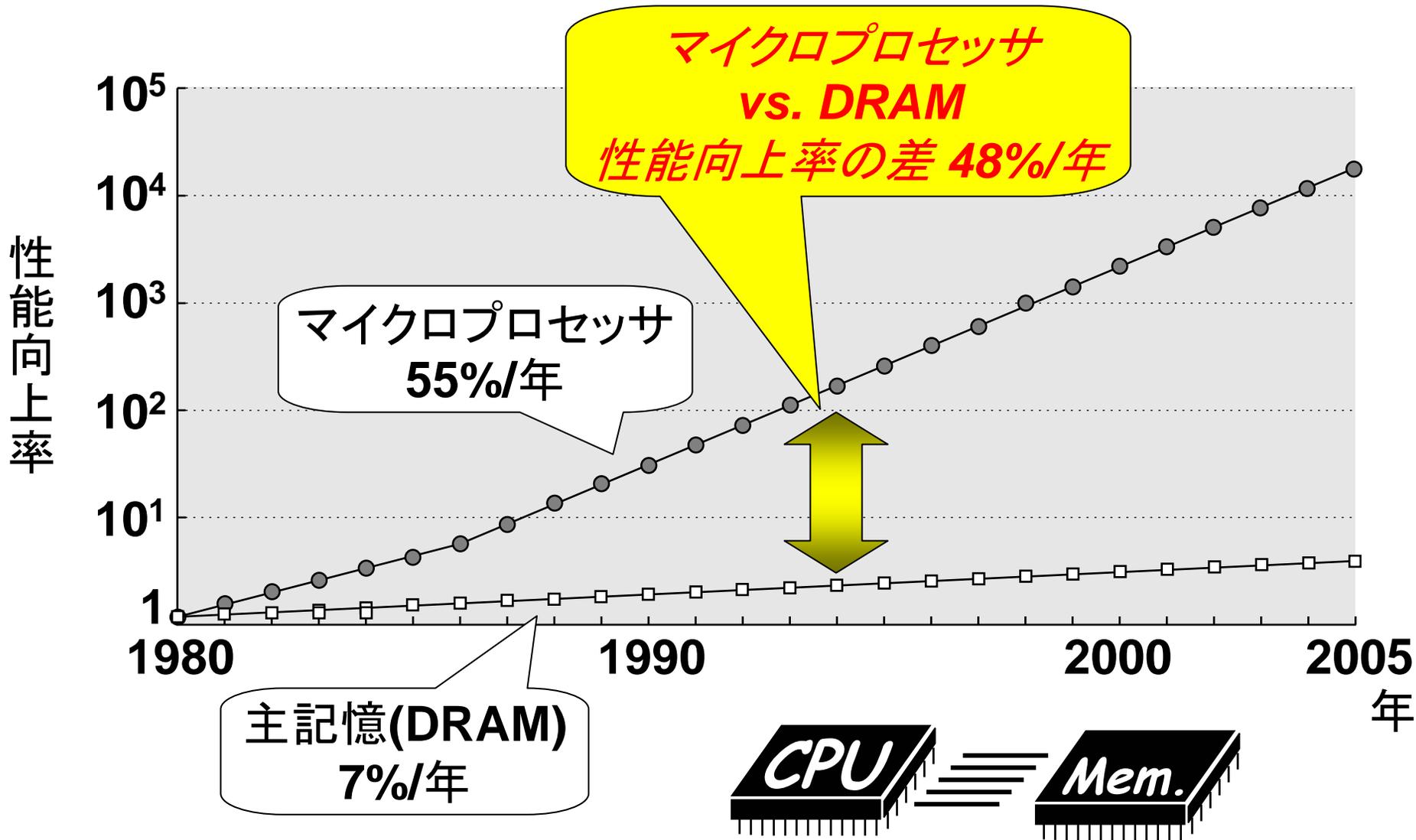
- 高頻度実行部分は存在するのか？
- それを動的に検出できるのか？
- それを加速実行できるのか？

高頻度実行部を検出し加速実行する

- 高頻度実行部分は存在する！
 - 多くのプログラムはループ構造を持つ
 - コードの「(たった)1%」の部分が全実行命令数の「50%」を占める！
- 高頻度実行部分は検出できる！
 - 分岐命令の実行履歴で検出可能
- 高頻度実行部分は加速実行できる！
 - 再構成可能機能ユニットの活用
 - 命令レベル並列性の活用
 - など



メモリの壁を打ち破る！



戦略その1

「ミス頻発ロード命令」をやっつける！

- 20個程度の静的な(コード中に現れる)ロード命令が、全L2キャッシュ・ミスの90%を占める場合もある！
- キャッシュ・ミスを引き起こす「ロード対象データ」を生成したストア命令にも偏りがある！

ミス頻発ロード命令による
キャッシュ・ミスを回避



実行時間を70%も削減！

```
Load a
Load b
Add c, a, b
Store c
⋮
Load c
Load x
Add z, x, c
Store z
```

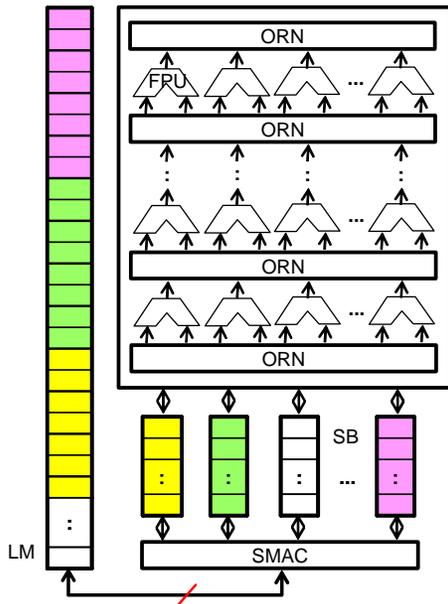
Cache
Miss!

戦略その2

中間結果はすぐ使え（保存しない）！

● 「メモリアクセス」そのものを減らそう！

- 不必要なメモリアクセスがあるのでは？
 - スピルコード（中間結果の一時退避）
- 中間結果の一時保存の必要性を無くす！
- 依存関係のある大量の命令を「一度に実行」する！



LSRDP (Large Scale Reconfigurable Data Path)

- 1個のチップに1,000個程度のALUを搭載
- ALU間をオンチップ・ネットワークで接続
- データフロー・グラフを直接マッピング

メモリアクセス回数を1/5に！

どのような場合に効果があるのか？

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

$$\text{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

$$\text{tei}(3,1,1,1)=((PAy*(1+2*p*PAx*PBx)*(1+2*q*QCx*QDx)*f(0,t))/q+(((p+q)**4*((PAy+PQy)*q*(1+2*q*QCx*QDx)+p*(PAy+2*PAx*PAy*PQx*q+2*PAy*PBx*PQx*q+2*PAx*PBx*PQy*q^2*PAx*PAy*q*QCx^2*PAy*PBx*q*QCx^2*PAy*PQx*q*QCx+2*q*((PAy*(PAx+PBx+PQx))+2*(PAy*(PAx+PBx)*PQx+PAx*PBx*PQy)*q*QCx)*QDx))^2*p**2*PAx*PAy*PBx*(1+2*PQx*q*(QCx+QDx)))*f(1,t))/q+(p+q)*((p+q)*((p+q)*(3*p*PAy+6*p**2*PAx*PAy*PQx+6*p**2*PAy*PBx*PQx+2*p**2*PAy*PQx**2+4*p**3*PAx*PAy*PBx*PQx**2+p*PQy+2*p**2*PAx*PBx*PQy+2*p*PAy*PQx**2*q+PQy*q+2*p*PAx*PQx*PQy*q$$

➔ 116 MUL, 31 ADD, 2 FUNC

Challenges!!

High performance

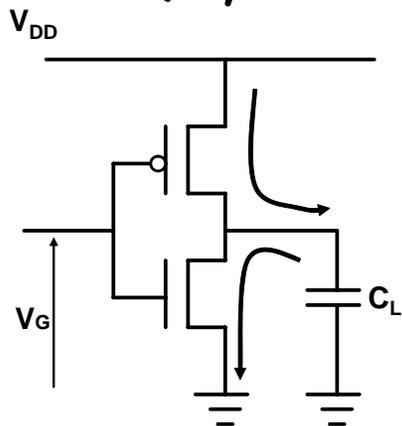
Low Power/Energy

- 高速かつ低消費電力なメモリを実現する！
- 高速かつ低消費電力なプロセッサを実現する！

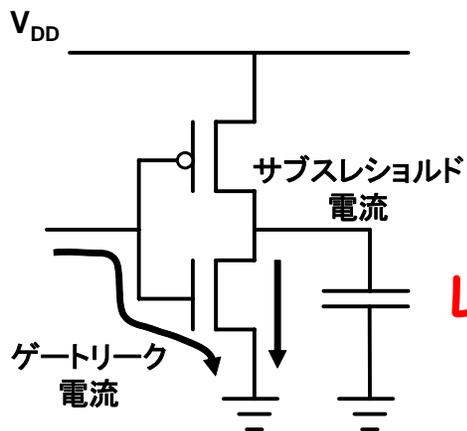
Design Environment

増加するリーク消費電力

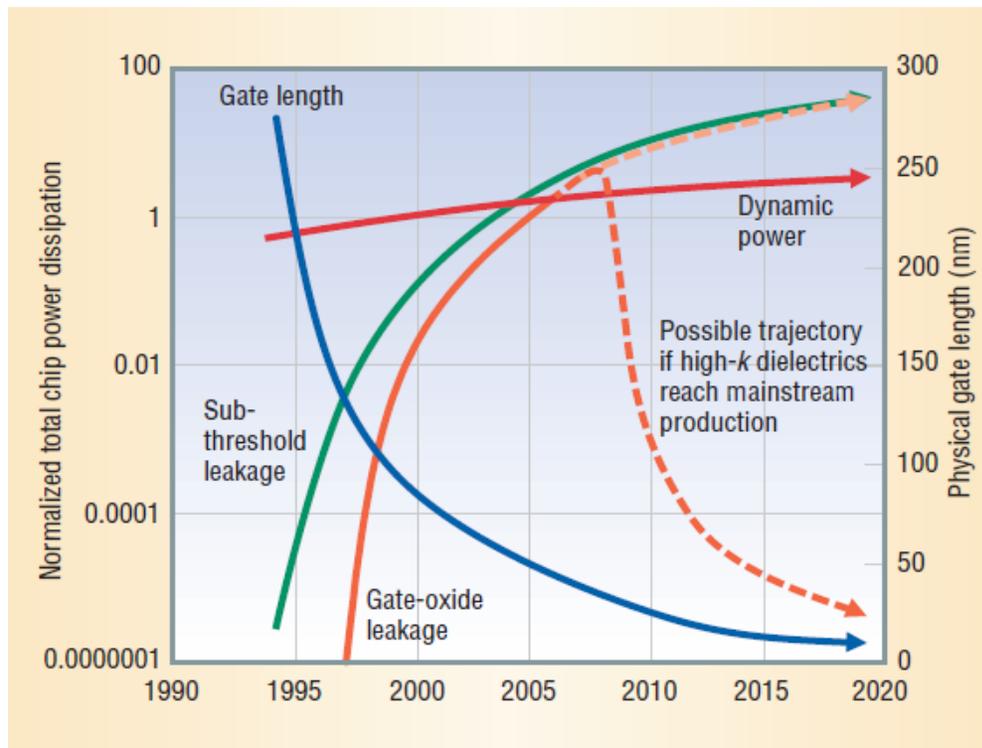
Capacitive (Dynamic) Power



Leakage (Static) Power



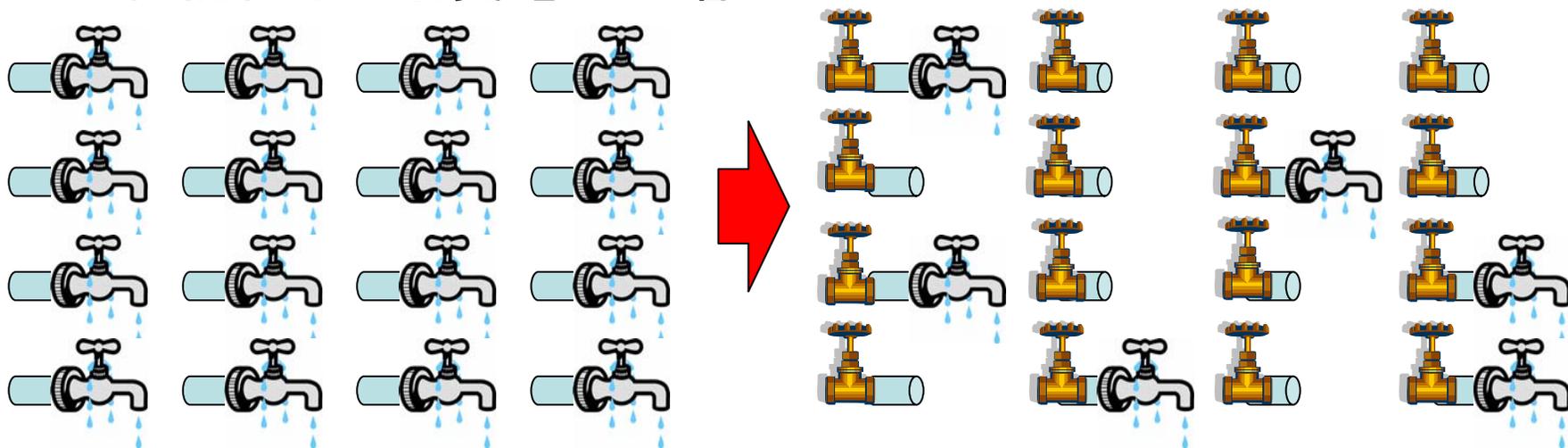
Leakage!



Source: N. S. Kim, T. Austin, D. Blaauw, T. Mudge, K. Flautner, J. S. Hu, M. J. Irwin, M. Kandemir, and V. Narayanan, "Leakage Current: Moore's Law Meets Static Power," IEEE Computer, Vol. 36, No. 12, pp.68-75, Dec. 2003.

性能を犠牲にしてキャッシュのリーク 消費電力を削減 (Cache Decay[Kaxiras01])

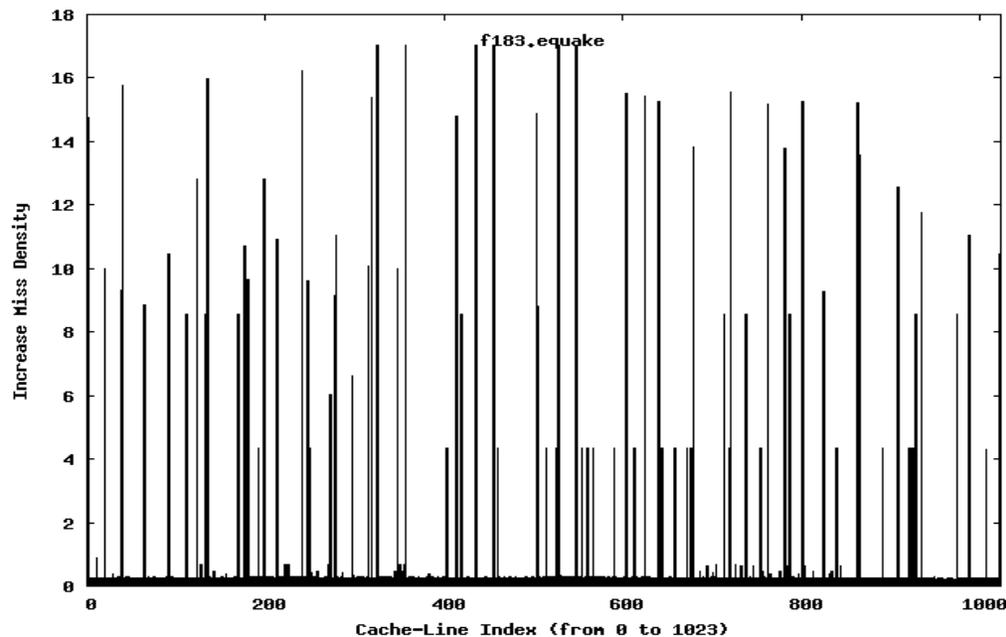
- 衰退ライン(使用済みライン)への電源供給を停止
- しばらくアクセスが無ければ、「将来もアクセスされないだろう」と予測
 - 予測が外れた場合はキャッシュ・ミスが発生 (Extra-Miss)
 - オフチップ・アクセス回数の増加 (10倍以上になることも)
 - 性能低下と消費電力の増加



性能を犠牲にする事なくキャッシュの リーク消費電力を削減！

- 停止モード・ラインへのアクセスの90%以上は「3%のキャッシュ領域」で発生している！
- この3%の領域を動的に検出して常にアクティブ・モードで動作させる！

- 性能オーバーヘッドを1/5に！
- 消費エネルギーを20%削減！



Challenges!!

- 耐故障性を高める！
- 製造ばらつきの影響を小さくする！
- 安全性を高める！

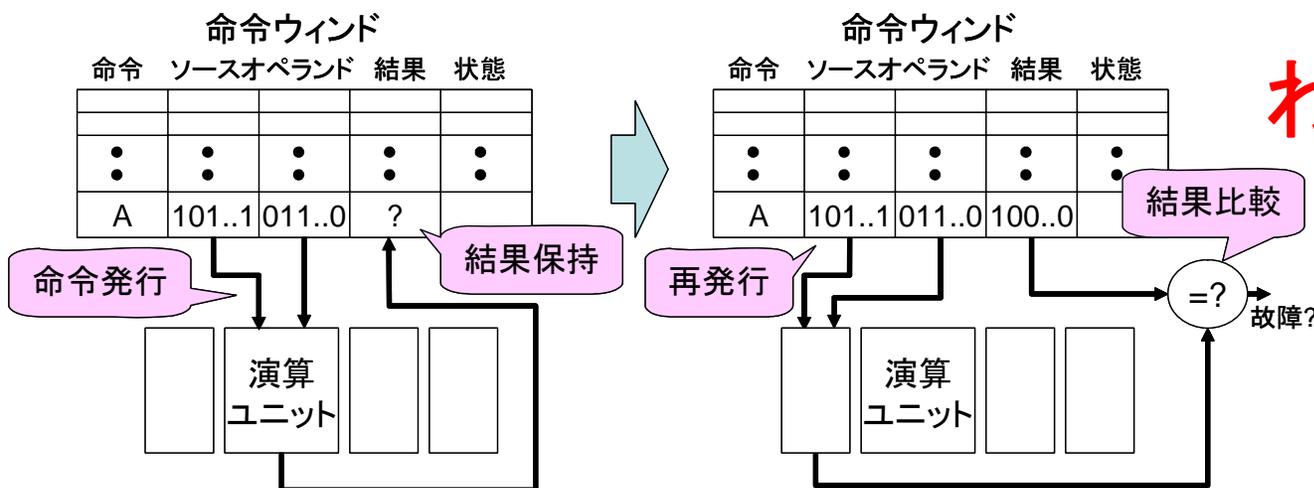
Dependable

Design Environment

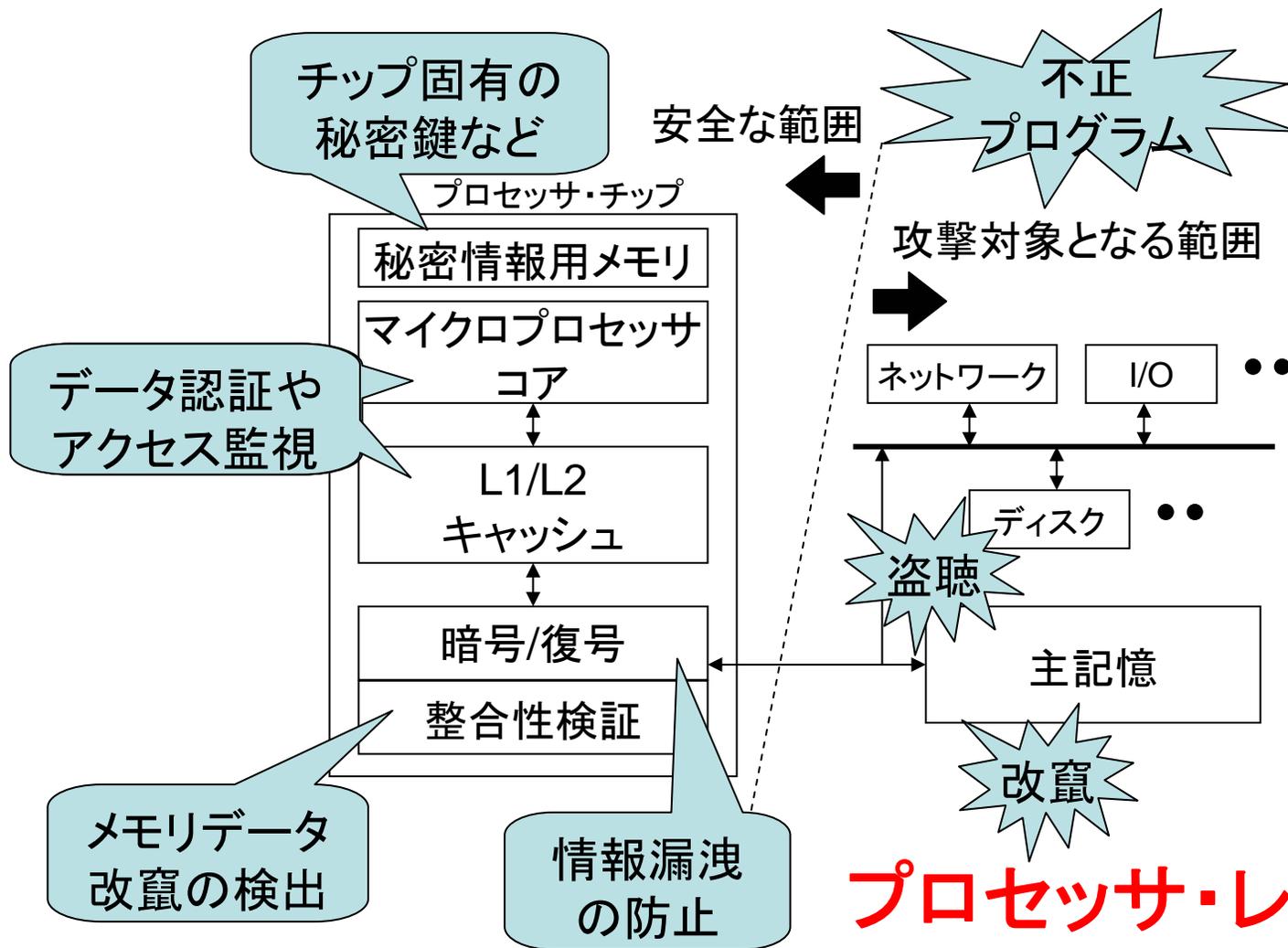
耐故障性を考慮したマイクロプロセッサ

- ソフトエラー
 - 外部放射線粒子によって引き起こされる回路中の一過性エラー
 - 原因はアルファ線や中性子線など
 - 回路内部で「アップセット・イベント」を引き起こす(値の反転)
- 対策: プロセッサ内部で**命令を複数回実行する!**
 - 様々な性能オーバーヘッド抑制技術

性能低下は
わずか15%程度!



安全性を考慮したマイクロプロセッサ



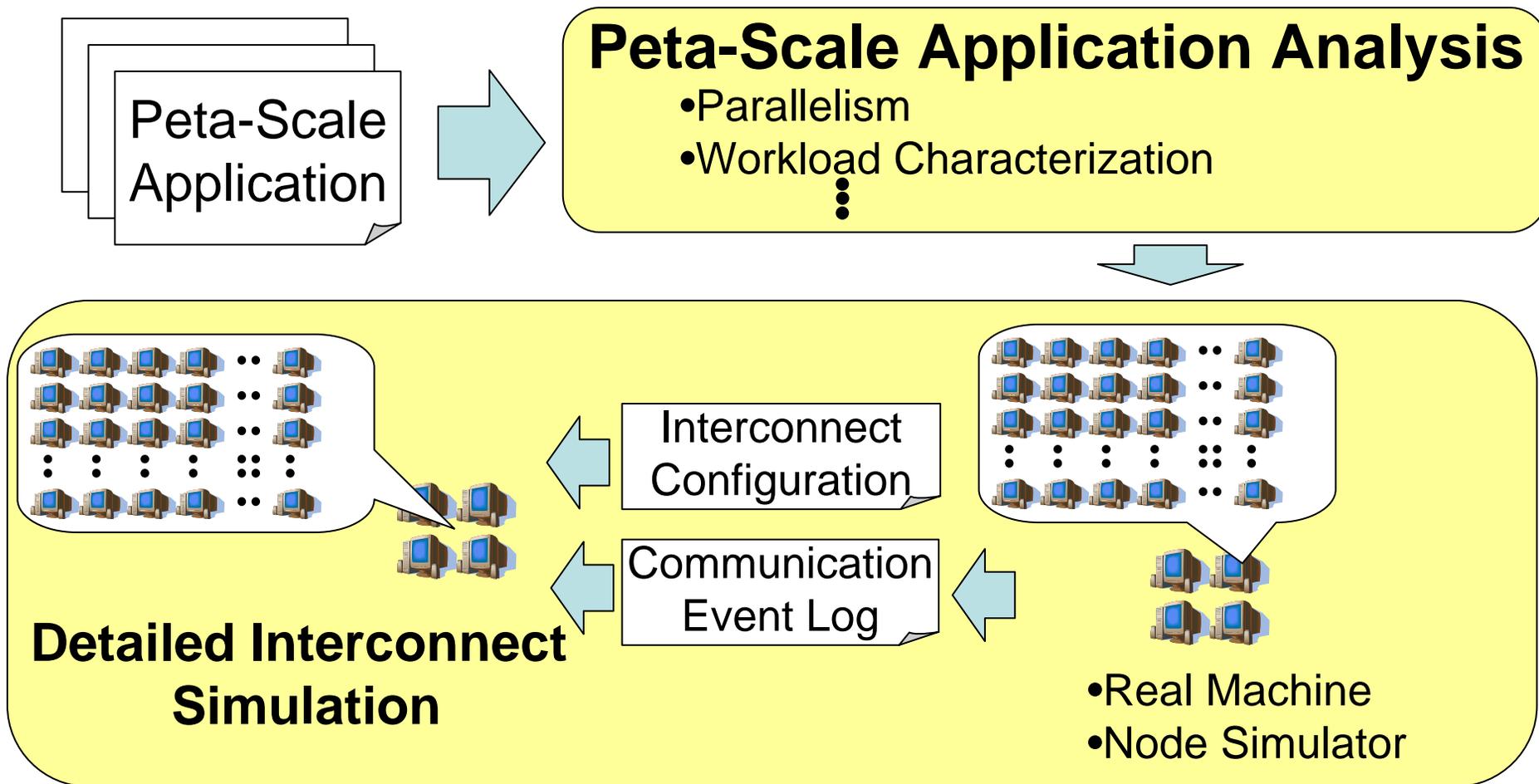
**プロセッサ・レベルで
ウィルス実行を検知!**

Challenges!!

- 次世代スパコンの性能を予測する！
- メモリシステムの性能評価時間を短縮する！
- プログラムから専用プロセッサを生成する！
- SoCの開発効率を向上する！

Design Environment

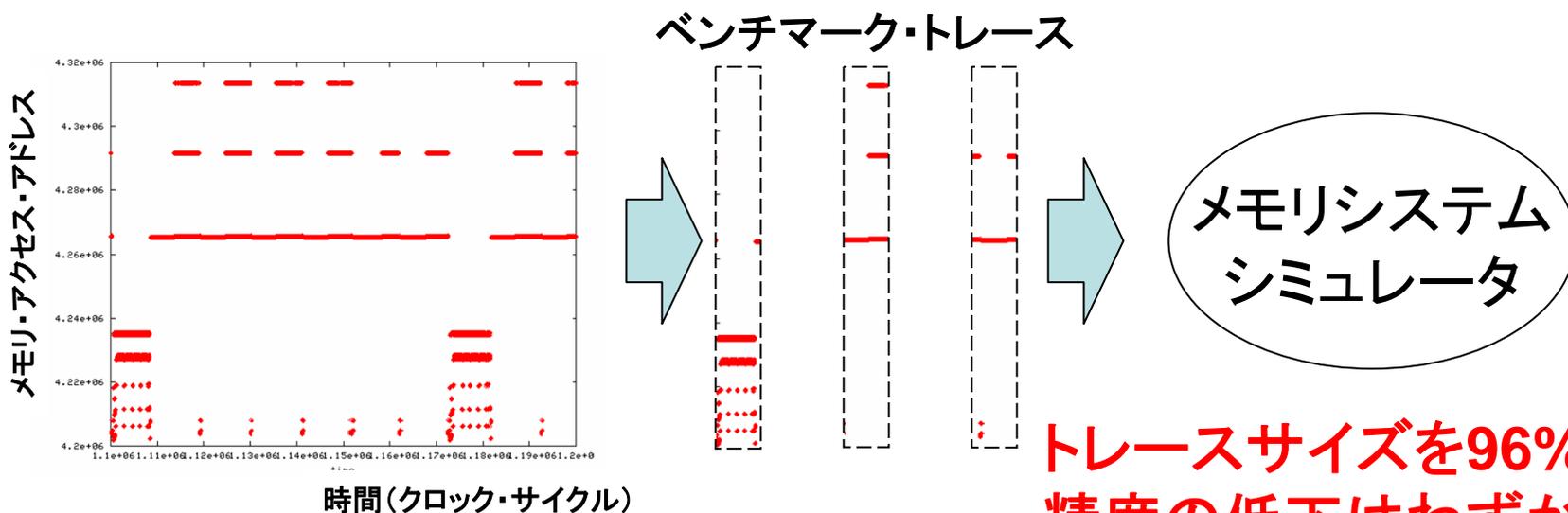
設計空間を探索する！ ～次世代スパコン開発の場合～



設計空間を探索する！

～SoC開発(特にメモリシステム)の場合～

- メモリアクセス・トレースを用いたシミュレーション
 - シミュレーション時間はトレースサイズに比例
- 精度の低下を招くことなくシミュレーション時間を短縮！
 - メモリアクセスの特長抽出
 - ベンチマーク・トレースの生成



トレースサイズを96%削減！
精度の低下はわずか2%！

Challenges!!

A Venn diagram with three overlapping ovals. The top oval is red and labeled 'High Performance'. The bottom-left oval is blue and labeled 'Low Power/Energy'. The bottom-right oval is green and labeled 'Dependable'. The three ovals overlap in a central region.

High Performance

Low Power/Energy

Dependable

A horizontal bar with a gradient from light green on the left to yellow on the right, containing the text 'Design Environment'.

Design Environment

SoCからスパコンまで！

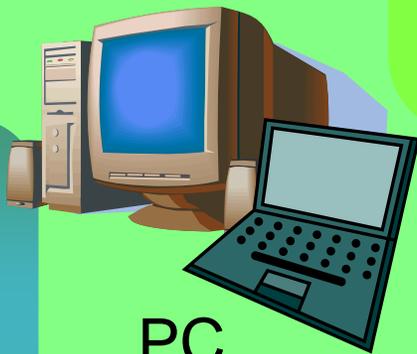
- 科学技術計算専用計算機の開発
- 次世代スパコン要素技術開発
- 科学技術計算向けCMPアーキテクチャ
- 再構成可能大規模データパス



スーパーコンピュータ



デスクサイド
高性能計算機



PC



組み込みシステム

- システム動的最適化技術の開発
- 動的再構成可能プロセッサの開発
- SoC設計技術

- 高性能/低消費電力プロセッサの開発
- ディペンダブル・プロセッサ
- 性能評価環境の構築