

ダーク・シリコン時代の プロセッサの省電力技術

三輪 忍
(東京大学)



本日の内容

- プロセッサの省電力技術とダーク・シリコン
- ダーク・シリコン時代の省電力プロセッサ
- まとめと今後の展望

本日の内容

- プロセッサの省電力技術とダーク・シリコン
- ダーク・シリコン時代の省電力プロセッサ
- まとめと今後の展望

スマートタブレットの消費電力

- スマートタブレット
 - 2010年 iPad 発売
 - その後 急速に普及
 - 2015年度には出荷台数が「タブレット>ノートPC」になる予定 (ICT総研調べ)

著作権上の問題
により削除

- スマートタブレットの消費電力
 - アプリケーションプロセッサが半分近い？
 - 他の携帯情報端末よりもヘビーユーザが多い？

プロセッサの省電力化が重要

従来のプロセッサの省電力化

● 基本戦略

- 性能への影響が少ない HW（機能）を削る
- HW を削った分，消費電力が減る

● 削るのはいつ？

- 設計段階で
 - コア数/キャッシュ容量を減らす
 - Out-of-order をやめる
- 設計後に
 - パワー・ゲーティング：使っていない HW を停止
 - DVFS：ゆっくり動いていい HW をゆっくり動かす

今後のプロセッサの省電力化

- 「HW を増やして電力を減らす」にシフト
- ダーク・シリコンの誕生が原因

ダーク・シリコンとは

- ダーク・シリコン
 - プロセッサ・チップ上に存在する同時には稼働できないトランジスタ (Tr) のこと
- なぜ稼働できないのか？
 - 消費電力の大きなチップを作っても売れないから
 - 100W のチップを買う携帯機器メーカーはない！
 - ダイナミック電力は稼働トランジスタ数に比例
 - 消費電力を抑制 → 稼働トランジスタ数を抑制

ダーク・シリコンの実際

- 既にチップ上に多数存在†
 - 82.4% のシリコンがダーク
(TSMC 90nm, 300 mm² のチップを 80 W で稼働)
 - ITRS 32nm では 96.7% がダーク
- 今後 LSI の微細化が進むにつれて増加

† G. Venkatesh, et al., Conservation cores: reducing the energy of mature computations, ASPLOS, 2010.

なぜダーク・シリコンは生まれたか？

● LSI のスケールリング則の変化

- 以前 (Dennard Scaling[†])
 - 微細化とともに電源電圧も低下
 - Tr の消費電力も低下

Parameters	Dennard scaling
Voltage	1/s
Saturation current	1/s
Oxide thickness	1/s
Gate length	1/s
Gate width	1/s
Area	1/s ²
Power	1/s ²
Power density	1

[†] R. H. Dennard, et al., Design of ion-implanted MOSFETs with very small physical dimensions, JSSC, 1974.

[‡] Computer Community Consortium, 21st Century Computer Architecture, white paper, 2012.

なぜダーク・シリコンは生まれたか？

● LSI のスケーリング則の変化

- 以前 (Dennard Scaling[†])
 - 微細化とともに電源電圧も低下
 - T_r の消費電力も低下
- 現在 (Post Dennard Scaling[‡])
 - 微細化しても電源電圧は変わらない
 - リーク電流が原因
 - T_r の消費電力は一定

Parameters	Dennard scaling	Post Dennard scaling
Voltage	1/s	1
Saturation current	1/s	1
Oxide thickness	1/s	1/s
Gate length	1/s	1/s
Gate width	1/s	1/s
Area	1/s ²	1/s ²
Power	1/s ²	1
Power density	1	S ²

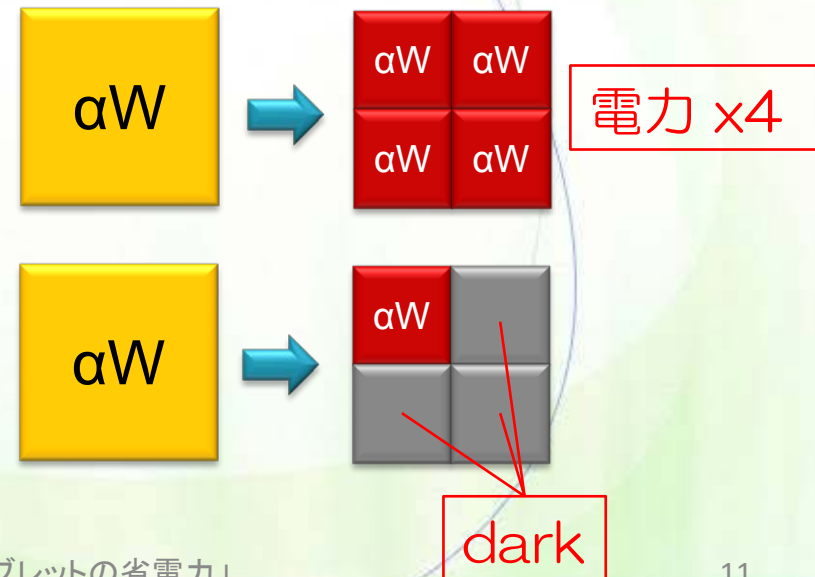
● T_r サイズはこれまで通り縮小. すると...

[†] R. H. Dennard, et al., Design of ion-implanted MOSFETs with very small physical dimensions, JSSC, 1974.
[‡] Computer Community Consortium, 21st Century Computer Architecture, white paper, 2012.

なぜダーク・シリコンは生まれたか？

- 微細化によってチップ上の Tr 数は増加
(ムーアの法則)
 - 一世代ごとに Tr サイズは 70% 縮小
 - 前世代と同じダイ面積なら Tr 数は 4 倍
- Post Dennard Scaling ではシリコンがダークになる！

- 増えた Tr を稼働
 - ➡ チップ電力の増加 😞
- 稼働 Tr 数を制限
 - ➡ チップ電力を維持 😊



省電力設計のパラダイム・シフト

- これまで：Dennard Scaling 時代
 - 電力制約 \doteq 回路資源制約
 - 「電力は減らしたいけど資源を増やしたくない」
例) 資源を1%増やして電力を10%減らす
- これから：Post Dennard Scaling 時代
 - 電力制約 \gg 回路資源制約
 - 電力制約のせいで使えない T_r は微細化とともに増加
 - 「電力を減らすために資源を贅沢に使う」
例) 資源を倍にして電力を半分にする

本日の内容

- プロセッサの省電力技術とダーク・シリコン
- **ダーク・シリコン時代の省電力プロセッサ**
- まとめと今後の展望

ダーク・シリコン時代の省電力プロセッサ

- ① big.LITTLE
- ② Composite Core
- ③ Single ISA ヘテロジニアスCMP
- ④ Conservation Core
- ⑤ 回路資源の投入により電力効率を改善するプロセッサ

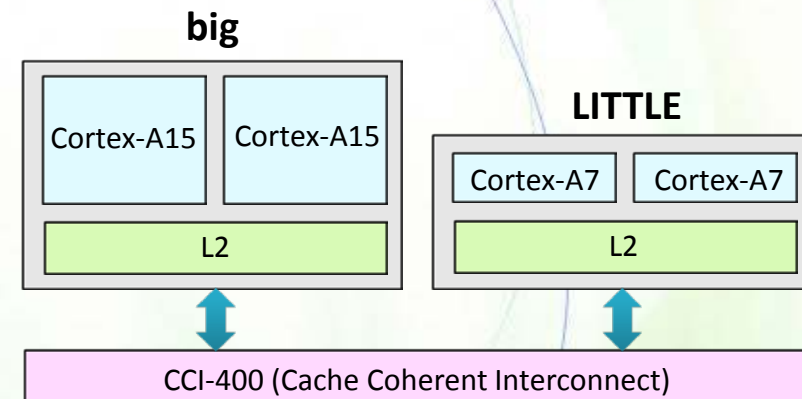
① big.LITTLE†

● 構成

- 電力性能特性の異なる2種類の汎用コアを搭載
 - big コア (e.g. Cortex-A15)
 - out-of-order 実行
 - 高性能かつ高消費電力
 - LITTLE コア (e.g. Cortex-A7)
 - Inorder 実行
 - 低性能かつ低消費電力
- 共有キャッシュはない
 - プライベートL2間をコヒーレンス制御

著作権上の問題
により削除

[big/LITTLE コアの電力性能特性]
(<http://www.ubergizmo.com/> より)



[big.LITTLE アーキテクチャ]

† P. Greenhalgh, big.LITTLE Processing with ARM Cortex-A15 & Cortex-A7, white paper, 2011.

① big.LITTLE†

● 使用方法

- アプリに応じて実行に使用するコアを変更
 - 性能が必要なアプリ
→ big で実行
 - 性能がいらぬアプリ
→ LITTLE で実行
- 使わないコアは電源OFF

● 効果

- 最大 76% の電力削減
- 回路規模は増加
- コアだけで17%増

著作権上の問題
により削除

[big/LITTLE コアの電力性能特性]
(<http://www.ubergizmo.com/> より)

著作権上の問題
により削除

[big.LITTLE の効果 (ARM社のHPより)]

† P. Greenhalgh, big.LITTLE Processing with ARM Cortex-A15 & Cortex-A7, white paper, 2011.

① big.LITTLE の問題点

1. コアは汎用のみ
 - 汎用な分、電力削減量は小さい
2. コアの種類が 2 つと少ない
 - 命令列によってはもっと省電力なコアがある
3. コア切り替えの時間粒度が粗い
 - アプリ単位ではフェーズの変化に対応できない
4. コア切り替えのオーバーヘッドが大きい
 - コアを切り替えた後はキャッシュ等が空の状態
 - 頻繁に切り替えると性能の劣化が大きい

② Composite Core†

- 狙い
 - コアの切り替えオーバーヘッドを削減
 - フェーズに応じたコア選択を可能にする

† A. Lukefahr, et al., Composite Cores: Pushing Heterogeneity Into a Core, MICRO, 2012.

② Composite Core[†]

● 構成

- 2種類の汎用実行系を搭載
 - big μ -エンジン：
out-of-order 実行の実行系
 - little μ -エンジン：
inorder 実行の実行系
- μ -エンジン間でキャッシュ
や命令フェッチ回路を共有
- オーバヘッドはほぼレジスタ値を転送するコストのみ

● 効果

- 1K 命令ごとの実行系の切り替えを実現
- big のみで実行した時と比べて平均18%のエネルギー減
- コア面積は約 20% 増加

著作権上の問題
により削除

[Composite Core のアーキテクチャ[†]]

[†] A. Lukefahr, et al., Composite Cores: Pushing Heterogeneity Into a Core, MICRO, 2012.

① big.LITTLE の問題点

1. コアは汎用のみ

- 汎用な分、電力削減量は小さい

2. コアの種類が 2 つと少ない

- 命令列によってはもっと省電力なコアがある

3. コア切り替えの時間粒度が粗い

- アプリ単位ではフェーズの変化に対応できない

4. コア切り替えのオーバーヘッドが大きい

- コアを切り替えた後はキャッシュ等が空の状態
- 頻繁に切り替えると性能の劣化が大きい

② Composite Core の問題点

1. コアは汎用のみ
 - 汎用な分，電力削減量は小さい
2. コアの種類が 2 つと少ない
 - 命令列によってはもっと省電力なコアがある
3. コア切り替えの時間粒度が粗い
 - アプリ単位ではフェーズの変化に対応できない
4. コア切り替えのオーバーヘッドが大きい
 - コアを切り替えた後はキャッシュ等が空の状態
 - 頻繁に切り替えると性能の劣化が大きい

③ Single ISA ヘテロジニアスCMP†

● 狙い

- コアの種類を増やす
- 各フェーズをより省電力なコアで実行できるようにする

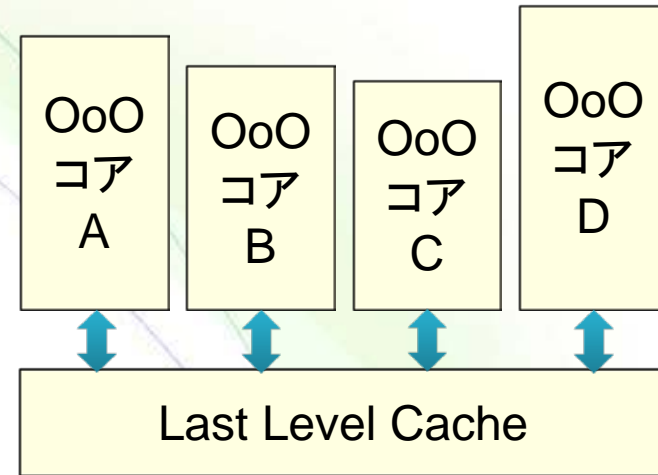
† S. Navada, et al., Unified View of Non-monotonic Core Selection and Application Steering in Heterogeneous Chip Multiprocessors, PACT, 2013.



③ Single ISA ヘテロジニアスCMP

● 構成

- 4種類の out-of-order コアをチップに搭載
 - ISAは共通
 - 各ユニットの資源量が異なる
Ex) 発行幅, 命令キュー,
レジスタ・ファイル, キャッシュ
- 最下位キャッシュを共有



● コアの切り替え方法

- 10K 命令ごとにコアの切り替えを判断
- 資源不足発生時に資源の豊富なコアへ切り替え

● 効果

- ホモジニアスCMPよりも33%高い電力効率

③ Single ISA ヘテロ CMP の問題点

1. コアは汎用のみ

- 汎用な分，電力削減量は小さい

← 未解決

2. コアの種類が 2 つと少ない

- 命令列によってはもっと省電力なコアがある

3. コア切り替えの時間粒度が粗い

- アプリ単位ではフェーズの変化に対応できない

4. コア切り替えのオーバーヘッドが大きい

- コアを切り替えた後はキャッシュ等が空の状態
- 頻繁に切り替えると性能の劣化が大きい

少し解決

④ Conservation Core[†]

● 狙い

- 特定の命令列の実行に特化したコアを用意
- 特定の命令列に対する電力効率の大幅な改善

[†] G. Venkatesh, et al., Conservation cores: reducing the energy of mature computations, ASPLOS, 2010.



④ Conservation Core[†]

● 構成

- 特定の関数/ループを実行するための専用コアを多数搭載
 - 汎用プログラム中の部分コードとほぼ1対1対応する専用HW
 - 普通の専用HWが実行の対象としない汎用プログラム内の不規則コードを実行

著作権上の問題
により削除

● コアの切り替え方法

- コンパイラが専用コアで実行可能な区間を検出
- 該当区間に切り替えのための専用命令を挿入
- 通常コアでの実行中に専用命令が出現したらレジスタ値等を専用コアに転送し、専用コアによる実行を開始
- 専用コアによる実行が終了したらレジスタ値を通常コアへと戻し、通常コアによる実行を再開

[GreenDroid processor
(IEEE Micro, 31 (2), 2011 より)]

[†] G. Venkatesh, et al., Conservation cores: reducing the energy of mature computations, ASPLOS, 2010.

④ Conservation Core の利点と欠点

😊 専用コアで実行できる命令列については大幅な省電力化を達成

- 関数単体では最大 94 % のエネルギー削減[†]
- アプリ全体では最大 52% のエネルギー削減[†]

😞 実行できる命令列が限られている

[†] G. Venkatesh, et al., Conservation cores: reducing the energy of mature computations, ASPLOS, 2010.

⑤ 回路資源の投入により電力効率を改善するプロセッサ⁺‡

● 方針

- ダーク・シリコン時代は「電力制約 ≧ 資源制約」
- 資源をもっと贅沢に使って電力を減らそう！

⁺ 三輪忍ほか, ダーク・シリコン時代のプロセッサ・アーキテクチャに関する初期検討, SWoPP新潟, 2014

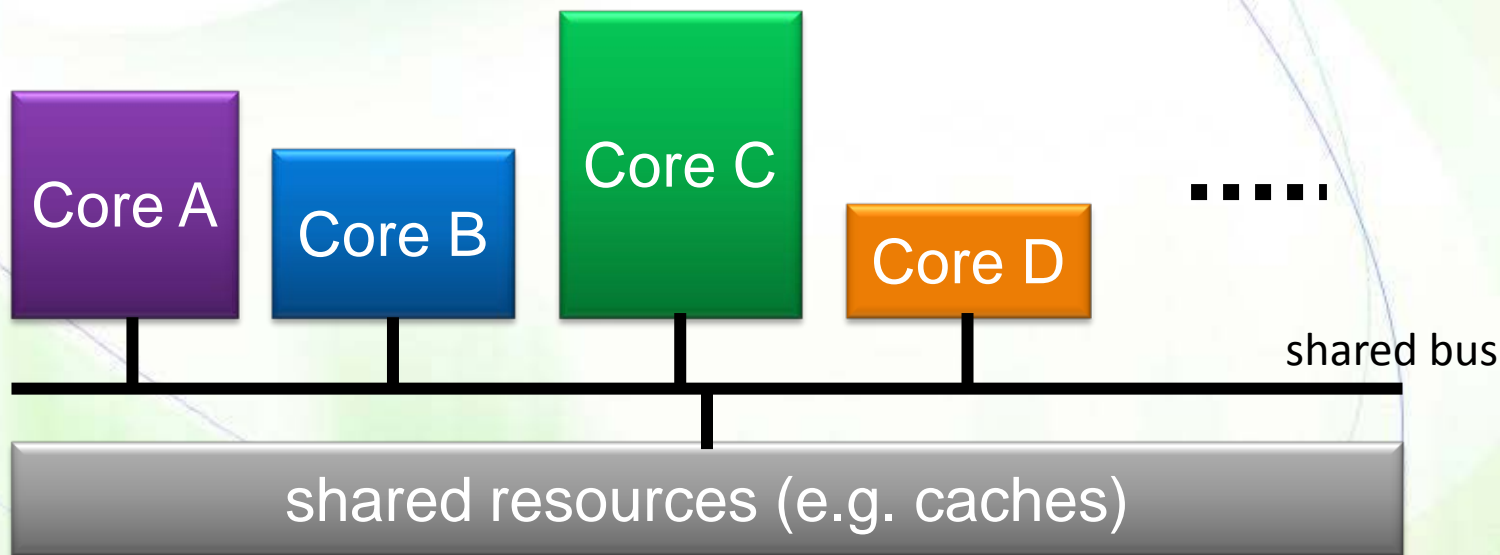
[‡] 三輪忍ほか, 回路資源の投入により電力効率を改善するプロセッサ・アーキテクチャ, ARC204, 2014



⑤ 回路資源の投入により電力効率を改善するプロセッサ[†]‡

● アーキテクチャの概要

- さまざまなタイプ（含：非汎用）のコアを多数搭載
- フェーズごとに最も電力効率の高いコアをハードウェアが自動的に選択
- 選択されたコアを用いてアプリを実行
 - 使用コアのみ電源オン，その他はオフ



[†] 三輪忍ほか，ダーク・シリコン時代のプロセッサ・アーキテクチャに関する初期検討，SWoPP新潟，2014

[‡] 三輪忍ほか，回路資源の投入により電力効率を改善するプロセッサ・アーキテクチャ，ARC204，2014

着眼点

- 汎用実行系では電力を浪費してしまう命令列

例) 456.hmmmer の1G + 17,000 命令目付近

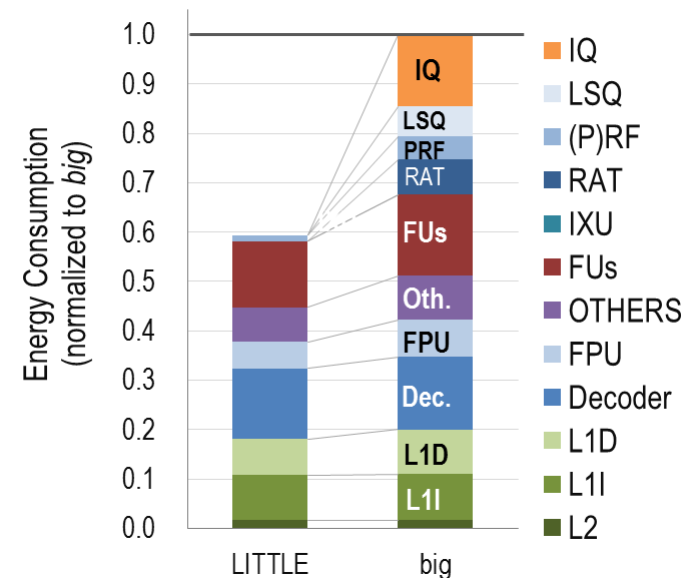
- 10 個の静的命令からなるループを 2,000 命令実行
- 命令の種類はたった 6 つ
- 使用データ・パス幅は最大 33 b

- 実行系の機能を制限すれば電力効率をもっと改善できるはず

- 10命令フェッチ/デコード後にフロントエンドを停止
- 6命令のみに対応したデコーダ
- データ・パスが 33 b 幅

```
12000e4b0: r24 <= ldl r6
12000e4b4: r7 <= ldl r2
12000e4b8: r5 <= addl r5
12000e4bc: r2 <= lda r2
12000e4c0: r8 <= cmple r5, r23
12000e4c4: r6 <= lda r6
12000e4c8: r7 <= addl r24, r7
12000e4cc: r24 <= cmple r4, r7
12000e4d0: r4 <= cmovne r24, r4, r7
12000e4d4: bne r8
```

[456.hmmmer の部分コード]



[big/LITTLE の消費エネルギーの内訳]

搭載予定のコアと切り替え粒度

- コアの種類
 - 汎用コア（任意の命令列の実行可）
 - OoO コア
 - inorder コアなど
 - 非汎用コア（特定の命令列のみ実行可）
 - ループ専用コア：ループ実行に特化したコア
 - スリムなコア：データ・パス幅が半分のコアなど
- コア切り替えの時間粒度
 - なるべく短く（目標：100～1,000 命令ごと）

各方式のまとめ

	big.LITTLE	Composite Core	Single-ISA Heterogeneous CMP	Conservation Core	回路資源投入方式
対象とする命令列	任意	任意	任意	特定	任意
Single ISA	Yes	Yes	Yes	No	Yes
非汎用コアを使用	No	No	No	Yes	Yes
コア数	2	2	4	≒10	>10
コア切り替えの時間粒度	粗	細	中	粗	細
切り替えオーバヘッド	大	小	中	小	小

予備実験

- 以下のアーキテクチャの消費エネルギーを評価
 - BASE
 - out-of-order コアのみ
 - BIG-LITTLE (← Composite Core を想定)
 - out-of-order コアと inorder コアの2つを切り替えながら実行する方式
 - 切り替えの判断は1Kコミット命令ごと
 - PROPOSAL-10K
 - 後述の17個のコアを切り替えながら実行する (提案) 方式
 - 切り替えの判断は10Kコミット命令ごと
 - PROPOSAL-1K
 - 1Kコミット命令毎に切り替えの判断を行う提案方式
 - PROPOSAL-100
 - 100コミット命令毎に切り替えの判断を行う提案方式

実験に使用したコアの種類

- 機能制限なし
 - **big** : Cortex-A57相当
 - **LITTLE** : Cortex-A53相当
 - **MLP** : L/S 周りを強化
- 機能制限あり
 - **LOOP** : ループ実行専用
 - **(INT/FP)-big** : INT/FP系パイプのみのbig
 - **(INT/FP)-LITTLE** : 同 LITTLE
 - **(INT/FP)-LOOP** : 同 LOOP
 - **SLIM-(big/LITTLE/LOOP)** : データ・パス幅が32bのbig/LITTLE/LOOP
 - **SLIM-(INT/FP)-(big/LITTLE/LOOP)** : データ・パス幅が32bで、かつ、INT/FP系パイプのみのbig/LITTLE/LOOP

コアのパラメータ一覧

● 32nm, 2GHz を仮定

	big	LITTLE	MLP	LOOP
Type	OoO	inorder	OoO	※パラメータは big と同じ ※IFU/RU は 所定の命令数をフェッチ/リネーム後に停止
Fetch width	2 inst.	2 inst.	1 inst.	
Issue width	3 inst.	2 inst.	2 inst.	
Issue queue	24 entries		128 entries	
FU (int, mem, fp)	2, 1, 1	2, 1, 1	2, 2, 1	
ROB	128 entries		256 entries	
int/fp PRF	160/160 entries		240/240 entries	
Ld/st queue	32/32 entries		64/64 entries	
Branch pred.	4KB g-share 512 entries BTB	←	←	
Br. miss penalty	12 cycles	8 cycles	12 cycles	
L1C (I)	48KB, 12 way, 64B/line, 2 cycles	←	←	
L1C (D)	32KB, 8 way, 64B/line, 2 cycles	←	←	
L2C	512KB, 8 way, 64B/line, 12 cycles	←	←	
Main mem	200 cycles	←	←	
Data path	64 bit	←	←	



その他のコアとコア選択の方法

- その他のコアについては以下の通り

	INT-*	FP-*	SLIM-*
Remarks	コア「*」から FPU, FPレジスタ, FP命令スケジューラ, FPマップ表を除いたもの	コア「*」から ALU, INTレジスタ, INT命令スケジューラ, INTマップ表を除いたもの	データ・パス幅が32b (キャッシュ・ライン・サイズも通常の半分)

- コア選択の方法

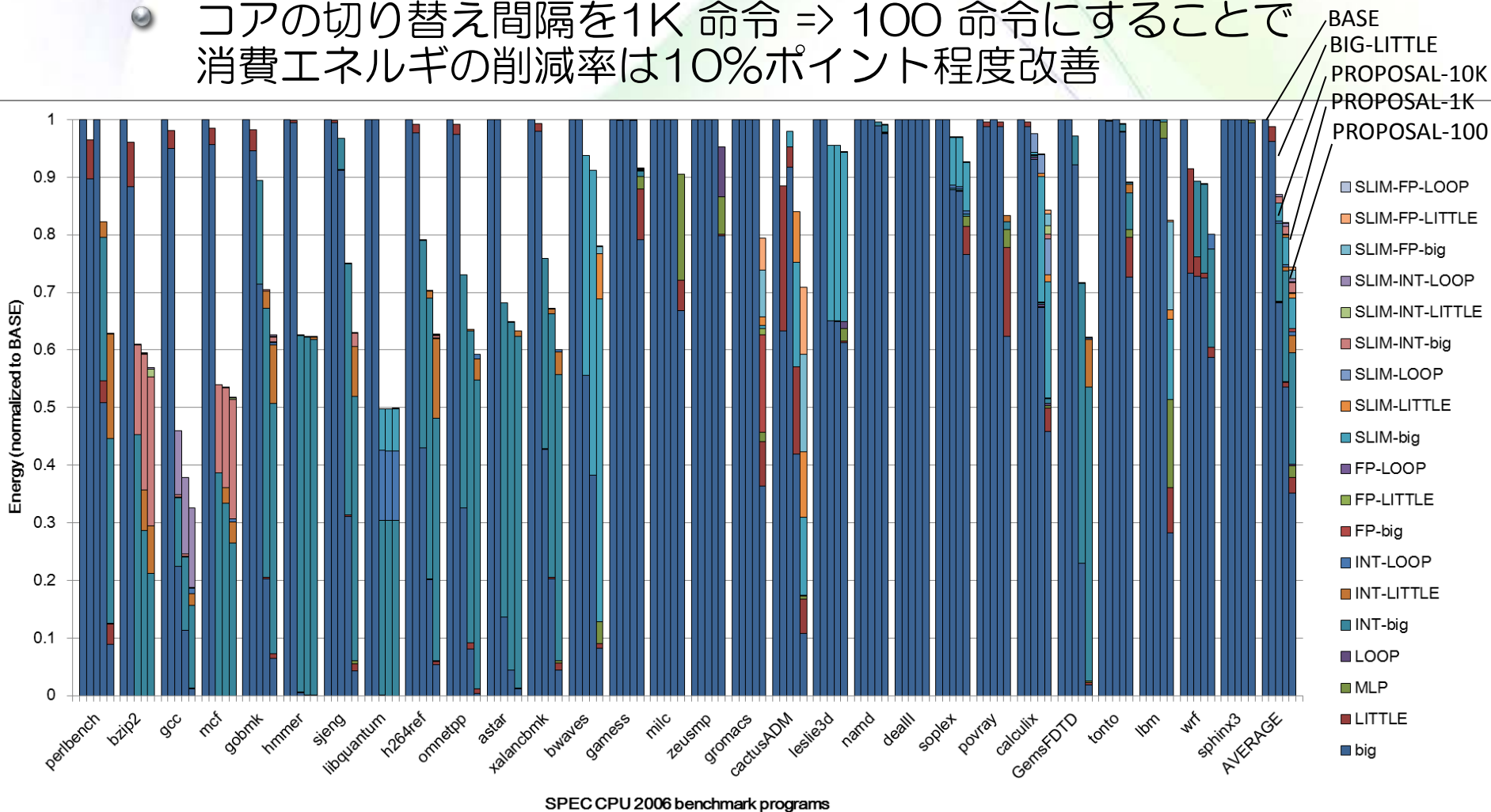
- 一定コミット命令数ごとに以下のアルゴリズムによりコアを1つ選択
 - 次に実行する命令列を実行可能なコア
 - bigコアに対する性能低下率が 5% 以内
 - 上記2つの条件を満たすコアの中で最も消費電力の少ないものを選択

実験方法

- シミュレータ
 - In-house シミュレータ (Onikiri2)
 - McPAT 1.0 (改造版)
- ベンチマーク・プログラム
 - SPEC CPU 2006
 - 1G 命令スキップ / 100M 命令実行
- 未使用コアに対しては理想的なパワー・ゲーティングが行われると仮定

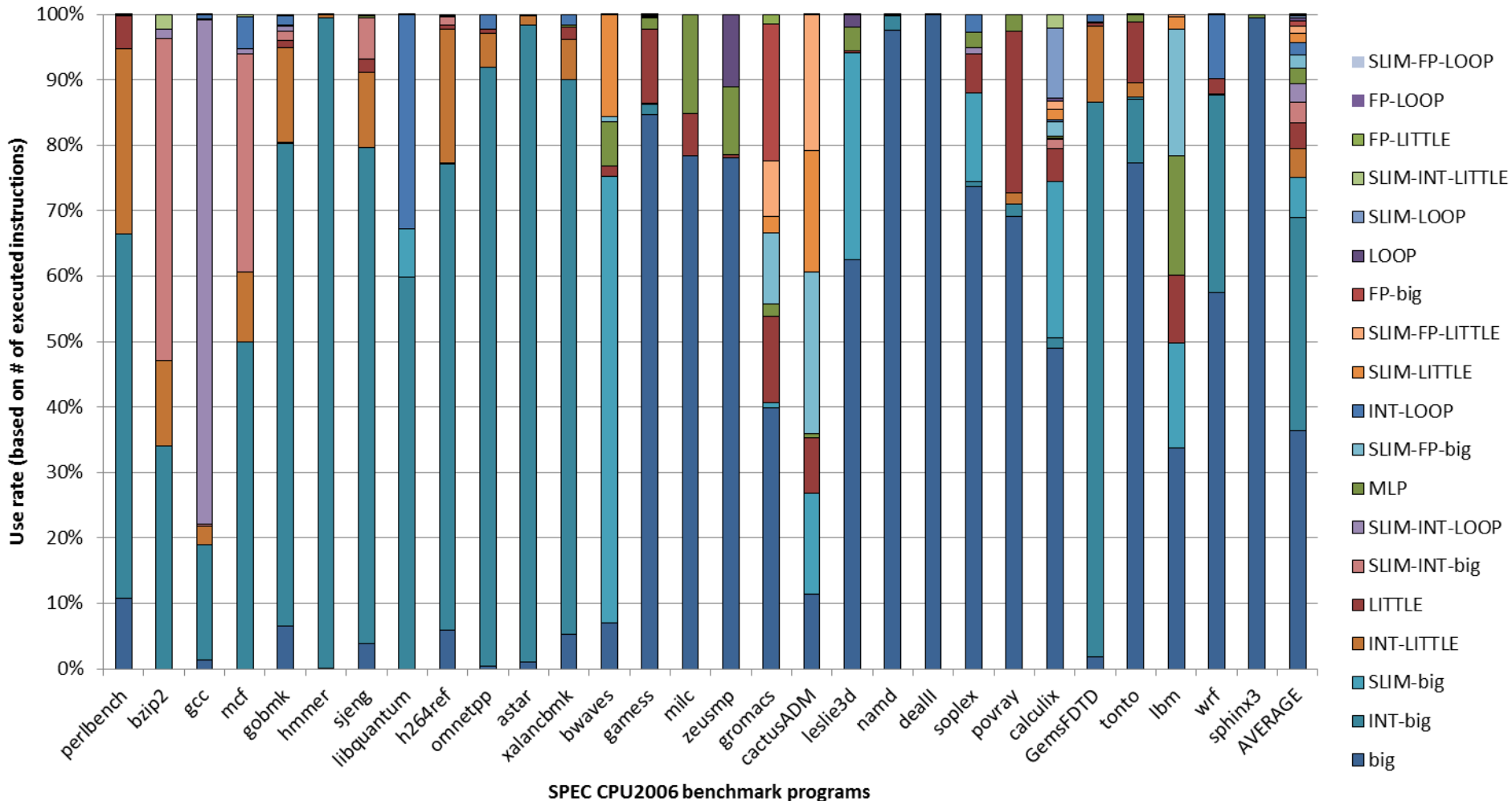
消費エネルギー

- 大幅なエネルギー削減ができる見込み
 - PROPOSAL-100 はBASEに対して最大67.5%， 平均25.5%
- 時間的に細粒度なコア切り替えは効果が高い
 - コアの切り替え間隔を1K 命令 ⇒ 100 命令にすることで消費エネルギーの削減率は10%ポイント程度改善



各コアの使用回数 (PROPOSAL-100)

- 1つのプログラムが使用するコアは高々6種類程度
- 全体で10種類 (big~INT-LOOP) 程度のコアがあれば、全コアを搭載した場合の95%をカバーできる



本日の内容

- プロセッサの省電力技術とダーク・シリコン
- ダーク・シリコン時代の省電力プロセッサ
- まとめと今後の展望

まとめ

- ダーク・シリコン時代の省電力プロセッサ
 - 電力を減らすために回路資源を贅沢に使う
- 省電力設計のパラダイム・シフト
 - 「1%の資源増で10%の電力減」はもう古い
 - 「N 倍の資源で 1/10 の電力」を目指す時代

今後の展望

- CMOS はそのうち限界が来る
 - 2025年頃？ (by ITRS)
 - ダーク・シリコンもその時に終わる
- ポスト CMOS[†] 向けのアーキテクチャをそろそろ本気で考え始める時期？
 - グラフェン
 - スピントロニクス
 - 分子エレクトロニクスなど

[†] James Meindl, Nanoelectronics in Retrospect, Prospect and Principle, ISSCC (plenary talk), 2010.