

# Power and Performance Evaluation of Globally Asynchronous Locally Synchronous Processors

著者： Anoop Iyer, Diana Marculescu

出典： *Proceedings of the 29th annual international symposium on Computer architecture, pp.158-168, 2002*

発表者： 所属:本多・近藤研究室 学籍番号:0953020 氏名:山藤 友紀

## 1 はじめに

近年、半導体の微細化や高クロック化による LSI の性能向上が顕著である。しかし、その反面データの書き込みタイミングを制御しているクロック信号がチップ全体に同時に伝搬されなくなるクロックスキューの発生や、消費電力の増大といった問題も深刻化している。上記問題点の解決策の一つとして、クロックを用いない非同期式设计手法がある。非同期式设计はクロックを用いないため、クロックスキューは存在しない。また、消費電力も抑えられる。しかし、同期式设计に比べ非同期式设计は設計資産がなく、設計が複雑である。

そこで、本論文はクロックスキューと消費電力の増大を抑制するために、局所的なブロックは同期式回路で構成し、同期ブロック間のデータ通信は非同期で制御を行う *Globally Asynchronous Locally Synchronous* (以下 *GALS*) プロセッサを提案する。また、同期プロセッサと比較した場合の性能と消費電力を評価する。

## 2 GALS プロセッサ

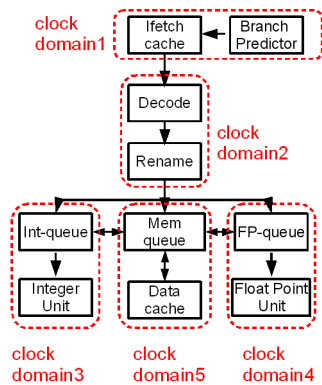


図 1: GALS プロセッサ

本論文で対象とするプロセッサは、アウトオブオーダのスーパーカラプロセッサである。図 1 はスーパーカラプロセッサを GALS に対応させた構成図である。図 1 において、点線で囲まれている部分はクロックドメインと呼び、それぞれ個別にクロックを持っている。また、各クロックドメインは動作周波数を個別に変更することが可能である。クロックドメイン間は、要求応答方式でデータ

のやり取りを行うためクロックは用いない。

### 2.1 提案するクロックドメイン

アウトオブオーダ実行のスーパーカラプロセッサにはスループットに差が生じる箇所が存在する。そこで、プロセッサを GALS 化したとき、同程度のスループットをもつ箇所をクロックドメインとしてまとめれば性能低下を防げる。

以下にスーパーカラプロセッサにおけるスループットに差が生じる箇所とその要因を列挙する。

1. *instruction flow* : 命令キャッシュミス & 分岐予測ミス
2. *register dataflow* : 演算資源の枯渇 & データ依存
3. *memory dataflow* : データキャッシュミス

パイプラインステージにおいて、*instruction flow* はフロントエンド部、*register dataflow* は整数と浮動小数点それぞれの演算ユニット、*memory dataflow* はロード/ストアユニットに関係している。上記 4 つを別のクロックドメインとすることで、各クロックドメインの動作周波数を最適化し性能低下を抑制できる。

クロックドメインの範囲を決定するとき、範囲が大きい程クロックスキューの抑制は困難に、範囲が小さい程クロックスキューの抑制が容易となる。

図 1 における点線で囲われている箇所が本論文で提案するクロックドメインの構成である。

### 2.2 非同期インターフェース

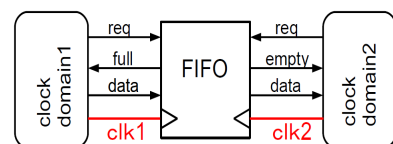


図 2: 非同期インターフェース FIFO

異なるクロックドメイン間のデータ生成と消費のスピード差を吸収するバッファとして、非同期インターフェース FIFO[1] を用いる。図 2 に FIFO の構成を示す。

図 2 において、clock domain1 がデータ送信準備を完了した場合、FIFO のデータ容量に空きがあるかないかの情

報を保持する full 信号によって空きがあると確認できたら req 信号とデータを FIFO に送信する。もし、空きがない場合は clock domain1 のクロックを止めデータ生成作業を止める。同様に、clock domain2 がデータ受信準備を完了した場合、FIFO のデータ容量が空か空でないかの情報を保持する empty 信号によって空でないことが確認できたら req 信号を送信しデータを受信する。もし、空であった場合は clock domain2 のクロックを止める。これにより、異なる動作周波数で動作するクロックドメイン間のデータ通信を行うことが可能である。

しかし、同期式プロセッサは 1 サイクルでデータのやり取りを行えるのに対して、GALS プロセッサにおける異なるクロックドメイン間でデータのやり取りを行う場合、FIFO を通過するため余分にサイクルがかかってしまうという問題もある。

### 2.3 供給電圧と動作周波数の関係

GALS プロセッサは使用率の低いクロックドメインの供給電圧を下げ動作周波数を低くすることで消費電力の抑制を行う。

また、本論文で採用する動作周波数の変更方法の例として、整数演算命令実行時、浮動小数点を行う FloatingPoint ユニットの使用頻度は低いと仮定して以下の様に行う。

- FloatingPoint クロックドメイン:  
動作周波数を 50% 落とす
- Fetch&Memory クロックドメイン:  
動作周波数を 10% 落とす

## 3 実験

本論文ではアウトオブオーダのスーパースカラプロセッサを対象とし、GALS プロセッサのシミュレータを作成して評価を行う。使用するベンチマークは SPEC 95 と Mediabench ベンチマークである。

### 3.1 GALS プロセッサの性能低下要因

各クロックドメインを同一の動作周波数で動作させた GALS プロセッサと同期式プロセッサのシミュレーションを行ったところ、GALS プロセッサは 5-15% の性能低下が確認できた。

そこで GALS プロセッサの性能低下要因を調べるため、GALS プロセッサがプログラムにおける 1 命令の実行完了サイクルに対してパイプラインのステージに費やすサイクルと FIFO に費やすサイクルの相対値を調査した。図 3 に結果を示す。

縦軸は、命令の平均実行サイクルに対して、FIFO とマークされているのが命令実行中に FIFO に費やすサイクル、Pipeline とマークされているのが命令実行中に演算ユニットなどパイプライン中のステージに費やすサイクルの相対値を示している。図 3 を見ると FIFO に費やすサイクルは 10-30% もあることが分かる。

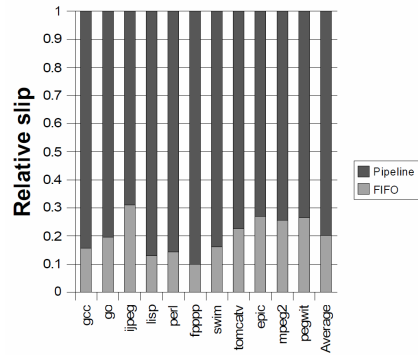


図 3: GALS のパイプラインステージと FIFO に費やしたサイクルの相対値

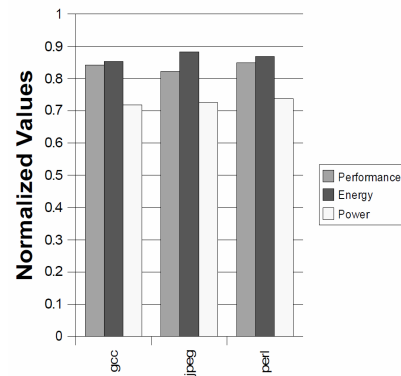


図 4: 動作周波数を個別に変更させた GALS プロセッサの評価

### 3.2 GALS プロセッサの評価

前節では、GALS プロセッサの全クロックドメインを同一の動作周波数で動作させていた。本節では、同期式プロセッサに対して各クロックドメインの動作周波数を変更する GALS プロセッサの性能と消費電力をシミュレーションにより調査する。図 4 に結果を示す。図 4 より、平均約 18% の性能低下で消費電力を約 28% 削減、そしてエネルギーは約 13% 削減可能であることが分かる。

## 4 まとめ

本論文ではクロックスキューを厳密に測定することが困難であるため調査は断念した。しかし、技術の進歩によって将来的に半導体の微細化や高クロック化はより進むと考えられるためクロックスキューは無視できない問題である。よって、クロックスキューや消費電力の抑制の観点から今後も GALS プロセッサの研究を進めることが重要である。

## 参考文献

- [1] T. Chelca and S. M. Nowick, "A Low-Latency FIFO for Mixed-Clock Systems," in Proc. IEEE Computer Society Workshop on VLSI, 2000.