

Look-Ahead ルーティングを用いたオンチップルータの動的パワーシャットダウン

松谷 宏紀[†] 鯉 淵 道 紘^{††}
王 代 涵[†] 天 野 英 晴[†]

オンチップルータはチップ内の通信インフラであるためスタンバイ時であっても完全に停止させることは難しく、省電力対策が進んでいるプロセッサコアと比べるとスタンバイ電力が大きい。そのため、Network-on-Chip (NoC) の動作に支障の無い範囲内でルータの各チャンネルの電源を走行時にシャットダウンする手法が NoC の省電力化のために有効であると考えられる。しかし、スリープ中の回路をウェイクアップさせるには一定の遅延がかかるため性能に影響が出るうえに、スリープ操作およびウェイクアップ操作のためにエネルギーが消費されるため、短期間のスリープは逆に消費電力の増加を招く。そこで、本論文では look-ahead ルーティングを応用し、あるルータのチャンネルが将来使われるかどうかの判断を 2 ホップ手前のルータで行うことで、ウェイクアップ遅延を隠蔽し、消費電力の増加を引き起こす短期間のスリープを削減する。実トラフィックを用いた評価によって、提案手法は 5 サイクル以下のウェイクアップ時間であれば隠蔽することができ、また、転送要求が無ければ常にスリープを試みるナイーブな方法と比べて実際にスリープできる期間を増やすことができた。

Look-Ahead Routing Based Dynamic Power Shutdown for On-Chip Routers

HIROKI MATSUTANI,[†] MICHIMIRO KOIBUCHI,^{††} DAIHAN WANG[†]
and HIDEHARU AMANO[†]

Since on-chip routers in Network-on-Chips (NoCs) play a key role for enabling on-chip communication between cores, they must be always preparing for packet injections even if cores are in standby mode, resulting in a larger standby power of routers compared with cores. The run-time power gating of individual channels in a router is one of attractive solutions to reduce the standby power of chip without affecting the on-chip communication. However, a state transition between sleep and active mode incurs performance penalty, and turning a power switch on or off dissipates overhead energy, which means a short-term sleep increases power consumption. In this paper, we propose a sleep control method based on the look-ahead routing that detects the arrival of packets two hops ahead, so as to hide the wake-up delay and reduce the short-term sleeps which increase power consumption. Simulation results using real application traces show that the proposed method conceals the wake-up delay of less than five cycles, and it also prolongs the sleep cycles compared with the original naive method.

1. はじめに

半導体技術の進歩によって単一チップ上にプロセッサやメモリ、I/O など複数の設計モジュールをタイル状に実装できるようになり、このようなタイル同士の結合に Network-on-Chip (NoC)¹⁾ が用いられるようになった。

タイルアーキテクチャの主要なアプリケーションは組込み機器であり、携帯機器や情報家電などへの応用が期待される。このような用途においては、バッテリー寿命を延ばすため、また、パッケージングや放熱にかかるコスト削減のため、ほぼすべての用途において消費電力の削減が求められている。

消費電力はスイッチング電力とリーク電力に大別され、スイッチング電力の削減にはクロックゲーティングやオバランドアイソレーション、リーク電力の削減にはパワーゲーティング等の技術が適用される。リーク電力に関しては、回路が動作していなくても電源が供給されている限り電力を消費し

てしまううえに、半導体技術の微細化にともない、全消費電力に占めるリーク電力の割合が近年著しく増加している。

このような状況を背景に、とりわけ、プロセッサコアにおけるリーク電力の削減が広く研究されており、文献 2) では携帯電話向けプロセッサのリーク電流を 11 μ A まで抑えることに成功している。このようにプロセッサコアの省電力対策が進むほど、オンチップルータのリーク電力が目立ってくる。したがって、コアのパワーゲーティングが行われるような環境ではルータのパワーゲーティングも同様に必須である。

ルータの省電力化の例として、文献 3) では Drowsy キャッシュを用いて入力バッファのリーク電力を削減している。文献 4) では、動的にリンクの電力をシャットダウンする際のルーティングアルゴリズムについて検討している。ただし、これらの研究では各チャンネルは数十から百数十フリットの比較的大きなバッファを持つことを想定している。一方、文献 5) や文献 6) で用いられるような小規模な wormhole ルータの場合、パワーゲーティングの際のウェイクアップ時間およびレジューム時間が通信性能に直に影響を及ぼす可能性がある。

そこで、本論文では、小規模な wormhole ルータを対象に、ウェイクアップ時間の性能への影響を隠蔽し、実際にスリー

[†] 慶應義塾大学大学院 理工学研究科
Graduate School of Science and Technology, Keio University

^{††} 国立情報学研究所
National Institute of Informatics

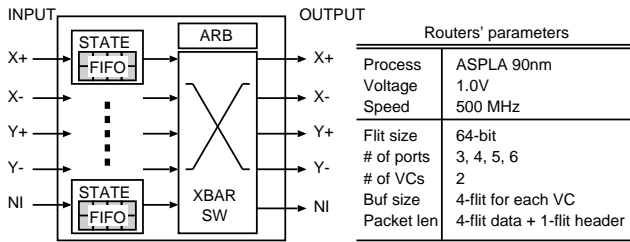


図1 ルータの概略図と実装パラメータ。

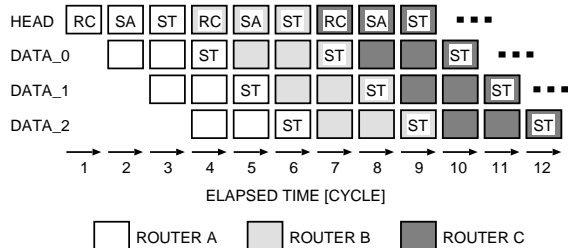


図2 ルータのパイプライン (ルータ A から C へのパケット転送)。

できる期間を増やすためのスリープ制御方法を提案する。本論文の構成は次のとおりである。まず、2章にてオンチップルータの消費電力を解析し、チャンネルの走行時パワーゲーティングが有効なことを示す。3章では走行時パワーゲーティングのための技術を紹介し、4章にて look-ahead ルーティングを応用したチャンネルの走行時パワーゲーティングを提案する。5章で評価を示し、6章で本論文をまとめる。

2. オンチップルータにおける消費電力の解析

本節では、文献5)と同様の方法を用いてオンチップルータの消費電力を解析する。

2.1 オンチップルータの構造

消費電力の解析のために、文献6)の wormhole ルータ回路をもとに典型的なオンチップルータ回路を 90nm スタンダードセルライブラリを用いて実装した。

図1に実装したルータ回路の概要を示す。このルータは入力バッファ、クロスバ、アービタ回路から構成されており、物理チャンネル当たり2本の仮想チャンネルを持つ。フリット幅は64-bitとし、仮想チャンネルごとに4-flit分の入力バッファを持つ。ルータ面積の約75%を入力バッファが占めている。

このルータは3段のパイプラインステージから構成され、入力されたヘッダフリットは1) routing computation (RC) ステージで出力ポートを計算、2) switch allocation (SA) ステージで出力ポートのアービトレーションを行い、3) switch traversal (ST) ステージにて隣接ルータへ転送される。

図2にルータのパイプライン構造を示す。この例では、ヘッダフリットと3個のデータフリットから成るパケットをルータAからルータCへ転送する場合を想定し、各ルータのパイプラインの中身をサイクルごとに表示している。

2.2 消費電力の解析結果

上記のルータの消費電力を求めるために、1) Design Compilerでルータ回路を合成、2) Astroで配置配線(CTSでバッファを挿入)、3) Verilog-XLで配置配線後シミュレーションを行い Switching Activity Interchange Format (SAIF) を生成し、4) Power CompilerでこのSAIFをもとに電力を解析した。なお、クロックゲーティングやオペランドアイソレーション等の省電力対策は十分に施してある。

手順3)では一定のトラフィック負荷を与え、500MHzで配

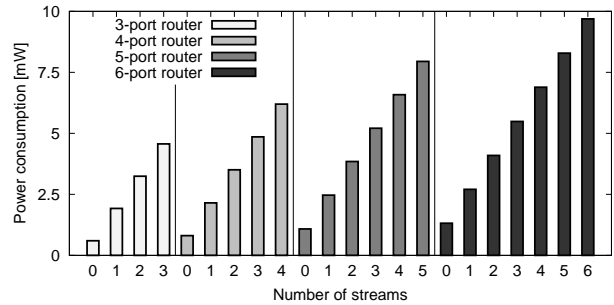


図3 負荷を一定としたときのルータの消費電力 (500MHzで動作)。

置配線後シミュレーションを行った。文献5)に倣い、リンクの最大帯域の30%の負荷になるように一定間隔で注入されるパケットの流れを stream と呼ぶ。ここでは、 n 個のチャンネルを持つルータに対し、0から n 本の stream を与えたときの消費電力を見積もった。ただし、 $3 \leq n \leq 6$ とする。

結果を図3に示す。当然、負荷が高くなるにしたがい消費電力が増えるが、スタンバイ時 (stream 数が0のとき)においても一定の電力が消費されている。表1に5-portルータを500MHzおよび200MHzで動作させたときのスタンバイ電力の内訳を示す。スタンバイ電力の63.2%~81.1%をリーク電力が占めていることが分かる。また、わずかに消費されている動的電力も、クロックゲーティング用に挿入されたラッチによるものであるため、動的電力のこれ以上の削減は難しい。

表1 5-portルータのスタンバイ電力 (500MHzと200MHz動作時)。

	500MHz	200MHz
Crossbar (dynamic)	33uW (3.1%)	13uW (1.6%)
Buffers (dynamic)	232uW (21.5%)	93uW (11.0%)
Controllers (dynamic)	133uW (12.3%)	53uW (6.3%)
Leakage (static)	683uW (63.2%)	683uW (81.1%)
Total	1,081uW	842uW

1章で述べたとおり、コアは使用されなければ完全に停止させることができるが、ウェイクアップに msec オーダの時間がかかる²⁾ため、通信インフラであるオンチップルータをこの方法で停止させることは難しい。省電力対策が進んでいるコアのスタンバイ電力²⁾と比べると、上記のルータのスタンバイ電力はかなり大きい。NoCの動作に支障の無い範囲内でルータのリーク電力を減らす必要がある。以上より、オンチップルータの各チャンネルの電源を走行時に動的にシャットダウンする手法がNoCの省電力化のために有効である。本章では、走行時のパワーゲーティング技術について調査する。

3. 走行時パワーゲーティング

パワーゲーティングは、スリープ対象の回路と Vdd ラインとの間、もしくは、GND ラインとの間にスリープトランジスタを挿入し、このスリープトランジスタを ON/OFF することで回路への電源を遮断/供給する。本章ではプロセッサを対象とした走行時パワーゲーティングの例を2つ紹介する。

3.1 MIPS R3000の細粒度動的スリープ制御

文献7)では芝浦工業大学で開発されたスリープトランジスタを用いて、MIPS R3000プロセッサの各種演算器を対象に走行時のパワーゲーティングを行った。このプロセッサは90nm CMOSプロセスを用いてレイアウトまで行われ、55%の消費電力削減を34%の面積オーバーヘッドで実現している。

3.2 Huのパワーゲーティングモデル

文献8)では、Huらはプロセッサの演算器単位のパワーゲーティング手法を示し、演算器単位の走行時パワーゲーティ

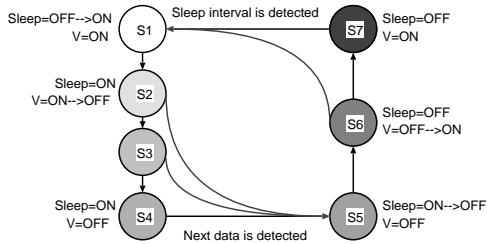


図 4 Hu のパワーゲーティングモデル (状態遷移図)。

ングをモデル化した。本節では、図 4 を用いて Hu らのパワーゲーティングモデルを説明する。

- S1: パワーゲーティング対象のユニットがしばらく使われないことを検出し, sleep 信号のアサートを開始. この操作に動的電力が消費される. sleep 信号のアサートが完了するまではユニットに電圧が供給され続ける.
- S2: sleep 信号のアサートが完了. この直後からユニットへの供給電圧が下がり始め, リーク電力も減り始める.
- S4: ユニットの電圧が OFF の値まで低下. スリープから覚めるまでこの状態が続く.
- S5: 対象ユニットが再び使われることを検出し, sleep 信号のネゲートを開始. この操作に動的電力が消費される.
- S6: sleep 信号のネゲートが完了. この直後からユニットの電圧が回復し始め, リークの削減効果も減り始める.
- S7: ユニットの電圧が ON の値まで回復し, リークの削減効果は消える.

スリープ中のユニットが再び使われることを検出してから動作可能になるまでに (S7 - S5) サイクルを要す. この時間を T_{wakeup} とする. T_{wakeup} が大きいとウェイクアップ時にストールが発生するため, T_{wakeup} は性能に影響を与える.

パワーゲーティングを実施するために S1 および S5 において動的電力が消費される. このオーバーヘッド電力とパワーゲーティングによって削減されるリーク電力が等しくなるタイミングを S3 とする. つまり, idle cycle が (S3 - S1) サイクルより長ければパワーゲーティングによって電力が削減されるが, 短ければ逆に消費電力が増加してしまう. この損益分岐点を $T_{breakeven} = (S3 - S1)$ とする. 文献 8) より, 最近のプロセスでは $T_{breakeven}$ は 10 サイクル前後となる.

動作中のユニットがしばらく使われないことを検出し, スリープするかどうかの判断を下すために S1 サイクルを要す. この時間を $T_{idledetect}$ とする. $T_{idledetect}$ が長いと判断に時間がかかり実際にスリープできる期間が減る.

4. Look-Ahead によるチャンネルシャットダウン

本章では T_{wakeup} による性能への影響を隠蔽し, 実際にスリープできる期間を増やすために, look-ahead ルーティングを応用したチャンネルのパワーゲーティング制御を提案する.

まず, look-ahead ルーティングを説明し, 次に提案手法を固定型および適応型ルーティングの場合に分けて説明する.

4.1 Look-Ahead ルーティング

パケットをルータ A からルータ C まで転送するとき, 通常のルータでは 図 2 に示すように i ホップ目の routing computation は i ホップ目のルータが行う.

一方, 1 ホップ先読みをする look-ahead ルーティングでは, i ホップ目の routing computation を $i - 1$ ホップ目のルータが行う. この様子を 図 5 に示す. 図中の NRC は next-hop routing computation の略である. この場合, i ホップ目の

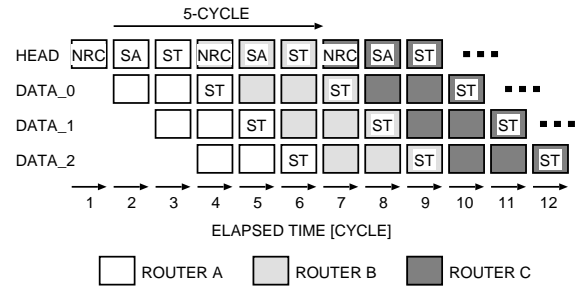


図 5 look-ahead ルータのパイプライン構造.

NRC ステージで計算される経路情報は, $i + 1$ ホップ目のルータで使われるため, i ホップ目の NRC ステージの結果が i ホップ目の SA や ST ステージに影響を与えることはない. このため, look-ahead ルーティングを用いれば, NRC ステージと他のステージをオーバーラップさせることができ, ルータのパイプライン段数を減らすことができる⁹⁾.

一方, 本論文ではこの look-ahead 機構をチャンネル単位のスリープおよびウェイクアップ判定に用いる.

4.2 固定型ルーティングの場合

ルーティングアルゴリズムは固定型と適応型に大別できる. 適応型ルーティングではパケットごとに動的に通信経路が選択されるが, 固定型ルーティングでは静的に経路が定まる. NoC では次元順ルーティングなどの固定型ルーティングが広く用いられているため, まず, 固定型ルーティングを想定して look-ahead によるチャンネルのスリープ制御を説明する. 適応型ルーティングの場合については 4.3 節にて述べる.

4.2.1 T_{wakeup} の隠蔽

固定型ルーティングの場合, 通信経路が動的に変化しないため look-ahead ルーティングを用いれば, $i - 1$ ホップ目の NRC ステージにおいて i ホップ目の出力ポートが判明する. これにより, 同時に $i + 1$ ホップ目のどの入力チャンネルが使われるかも判明する. つまり, 図 5 の例で, ルータ A の NRC ステージが完了した時点で, 2 ホップ先にあるルータ C の入力チャンネルをスリープまたはウェイクアップさせれば良いか判断できる. ルータ A の NRC ステージ完了後, パケットが実際にルータ C に到達するまで最低 5 サイクルの余裕があるため, $T_{wakeup} \leq 5$ であれば, パワーゲーティングによる通信遅延およびスループットへの影響を隠蔽できる.

4.2.2 busy 信号の生成

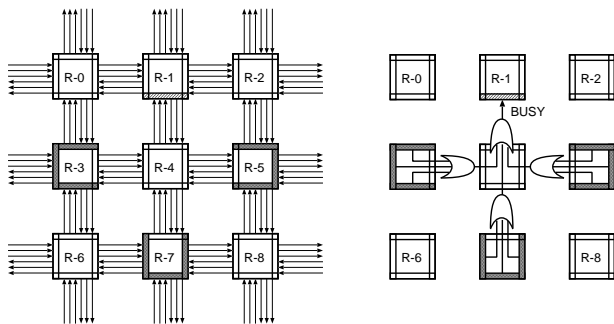
提案手法では, 各入力チャンネルに対し, 2 ホップ手前に位置するすべての入力チャンネルの情報を与える必要がある.

ルータ 1 の South チャンネルをスリープさせる場合の例を 図 6(a) に示す. 図中の R はルータを表し, 各ルータは 4 方向に入力チャンネル (North, East, South, West) を持つものとする. 提案手法において, ルータ 1 の South チャンネルをスリープ/ウェイクアップさせるには, ルータ 3, ルータ 5, ルータ 7 の入力チャンネルの情報が必要になる. 具体的には, 図 6(a) で着色されている入力チャンネルを監視する必要がある.

これらの入力チャンネルに, ルータ 1 の South チャンネル行きのパケットが含まれていれば, ルータ 1 の South チャンネルに対し busy 信号を与え, 5 サイクル後にパケットが到達することを通知する (図 6(b)).

4.2.3 スリープ/ウェイクアップ判定

すべての入力チャンネルは自分に与えられた busy 信号を監視し, スリープ中に busy 信号がアサートされればウェイクアップ



(a) busy 信号に関連する入力チャネル． (b) OR ゲートによる busy の生成．

図 6 ルータ 1 の South チャネルに対する busy 信号の生成．

ブシーケンスを開始する．一方、パケット転送直後に busy 信号がアサートされていないならば、しばらくパケットが来ないと判断しスリープを開始する．

4.3 適応型ルーティングの場合

適応型ルーティングの場合、ネットワークの混雑等に応じて通信経路が動的に変化する．具体的には、routing computation の際に入力ポートの候補が複数提示され、そのうち 1 つが output selection function (OSF) によって選択される．そのため、固定型ルーティングのときと同じ方法で 2 ホップ先の入力チャネルの状態を確実に判定することはできない．

この問題を解決するために、通信負荷に応じて OSF を次のように切替える．

- 低負荷時は、常に特定の出力ポートのみを選択する．
- 高負荷時は、ランダムに出力ポートを選択する．

ネットワーク負荷が小さいときは、ネットワークに注入されるトラフィック量によってスループットが頭打ちになるため、固定型ルーティングと適応型ルーティングの性能差は小さい．そこで、低負荷時は、常に特定の出力ポートのみを選択することで固定型ルーティング化し、look-ahead によるスリープ/ウェイクアップ判定が当たるようにする．

一方、ネットワーク負荷が大きいときは、ランダムに出力ポートを選択する OSF を用いて負荷分散を行い、スループットの向上を狙う．ただし、この場合、look-ahead によるスリープ判定が当たらなくなるため、走行時パワーゲーティングをあきらめる必要がある．このように、適応型ルーティングの場合、負荷が小さければスタンバイ電力を減らせるが、負荷が高いとスタンバイ電力を減らすことができない．

低負荷時 OSF と高負荷時 OSF の切替えは各チャネルが自律的に行う．具体的には各チャネルにおける転送待ちが n サイクルを越えたら負荷が高いと判断し、高負荷時 OSF に移行するものとする．ただし、最適な n はルータの構造、トポロジ、トラフィックパターンにより異なる．

OSF に切替えを各チャネルが自律的に行うため、低負荷と判断し look-ahead による走行時パワーゲーティングを試みるチャネルと、高負荷と判断しランダムな OSF を選択するチャネルが混在してしまい、look-ahead によるスリープ判定が外れる可能性がある．この場合、 $T_{wake up}$ を隠蔽できなくなり性能が低下してしまうが、通信を継続することはできる．

5. 評価

本章では次の 3 つのチャネルのスリープ制御 (look-ahead も含む) を比較する．

- ideal: パワーゲーティングのオーバーヘッドが存在しな

い理想的なモデル． $T_{wake up} = 0$, $T_{idle detect} = 0$ と等価．

- lookahead: 本論文で提案したモデル．ルータのパイプライン段数を 3 ステージとし、5 サイクル先まで先読みできるものとする．

- naive: いっさい先読みしないモデル．つまり 0 サイクル先まで先読みする lookahead ．

上記の 3 種類をハードウェア量、ウェイクアップ時間の性能への影響、スリープサイクル数について評価する．

5.1 ハードウェア量

ここでは、naive なチャネルのスリープ制御を look-ahead 化することで増えるハードウェア量を見積もる．

naive なチャネルのスリープ制御を look-ahead 化する際の主な変更点は次の 4 点である．1) RC ステージで次のホップの経路を計算する (つまり、既存の RC ステージを NRC ステージに置き換える) ．2) SA ステージでは 1 ホップ手前のルータの NRC ステージで計算された経路情報を用いる ．3) 各入力チャネルは 2 ホップ先のすべての入力チャネルに対し busy 信号を送信する ．4) 各入力チャネルは 2 ホップ手前のすべての入力チャネルからの busy 信号を受信する ．

1) と 2) については、RC ステージの経路計算関数を変更するだけでありハードウェア量の変化は小さい．図 6(b) の例では、変更点 3) として、各入力チャネルは 2 ホップ先の入力チャネル 9 個に対して busy 信号を送信する必要がある．また、4) として、各入力チャネルは 2 ホップ手前 9 個の入力チャネルからの busy 信号の論理和をとった結果を受信する．

以上の 1) ~ 4) の変更を施したルータを実装したところ、オリジナルのルータと比べて 1.08% 面積が増加した．面積の増分は 3) と 4) のための組合せロジックによるものが主である．

5.2 ウェイクアップ時間の性能への影響

$T_{wake up}$ は性能に影響を与える．ここでは、 $T_{wake up}$ を 0, 1, 2, 3 サイクルと変化させたときの性能を評価する．

評価方法

評価のためにフリットレベルシミュレータを用いた．ルータのスイッチング機構として、I/O バッファ、クロスバ、アービタを単純化したモデルを採用しており、2.1 節で示した wormhole ルータ回路と同じ挙動をするように設計した．ここでは 4-flit 分のデータに 1-flit 分のヘッダを付与したものを 1 パケットとし、ヘッダフリットが隣接ルータまたは計算コアに転送されるのに最低 3 サイクルかかるものとした．ネットワークポロジは 4×4 の 2 次元メッシュとし、ルーティングには次元順ルーティング⁹⁾を用いた．

シミュレーションに用いる通信パターンとして、ユニフォームトラフィックと NAS Parallel Benchmark (NPB) プログラムから得られた次の通信パターンを用いる：Block Tridiagonal solver (BT), Scalar Pentadiagonal solver (SP), Conjugate Gradient (CG), Multi-Grid solver (MG), large Integer Sort (IS)．プログラムのクラスは “W” とした．

なお、ここでは計算コア (NI) も 2 ホップ先の入力チャネルに対し busy 信号を送る機能を持つと仮定している．

評価結果

図 7 に naive 方式において $T_{wake up}$ を 0 ~ 3 に変化させたときのスループット (accepted traffic) とレイテンシのグラフを示す．スリープ中の入力チャネルにパケットが入力されると $T_{wake up}$ 分のストールが発生するので、 $T_{wake up}$ が増えるにしたがい性能が低下している．

ここで、ideal 方式の性能は $T_{wake up} = 0$ の場合に対応して

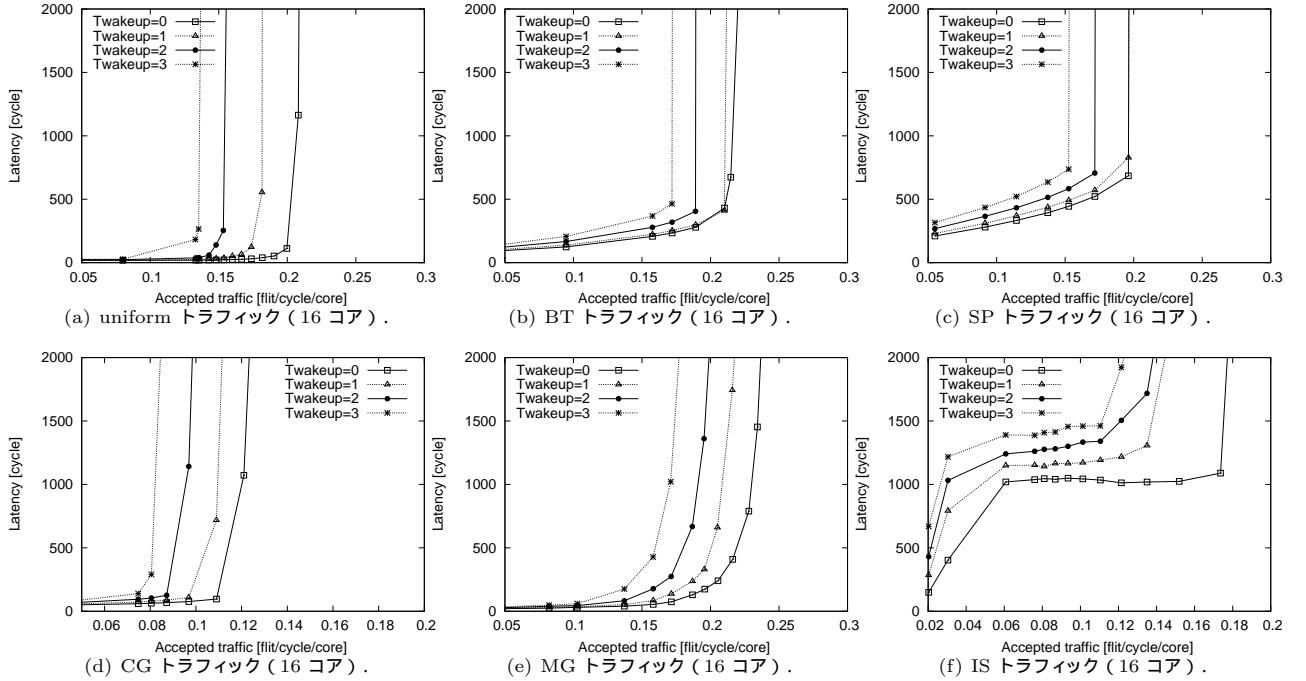


図 7 naive 方式のスループット ($T_{wakeup} = 0, 1, 2, 3$) .

いる．また，lookahead 方式では，4 章で示したとおり最低 5 サイクル前にパケットの到着を検出できるため， $T_{wakeup} \leq 5$ の場合であれば ideal と同じ性能が期待できる．

5.3 スリープ可能なサイクル数

ここでは ideal，lookahead，naive の各方式において，各チャネルが実際にスリープできたサイクル数を評価する．

評価方法

チャネルごとに，各トレースの全実行時間を次の 3 種類の状態に分類し，全実行時間に占めるこれらの状態の割合をシミュレーションによって求めた．

- N_{active} : チャネルが動作中のサイクル数．パワーゲーティングをしない状態と同量のリーク電力が消費される．
- N_{csc} : compensated sleep ($T_{breakeven}$ 以上のスリープ) のサイクル数． N_{csc} の長さに応じて消費電力が減る．
- N_{usc} : uncompensated sleep ($T_{breakeven}$ 未満のスリープ) のサイクル数． N_{usc} に応じて消費電力が増える．

シミュレーションでは， T_{wakeup} および $T_{idledetect}$ は 2 サイクルに固定とし， $T_{breakeven}$ は 10 サイクルの場合と 20 サイクルの場合に分けて評価した．

評価結果

図 8 に $T_{breakeven} = 10$ のときの結果を示す．傾向として，負荷 (accepted traffic) が高くなるにしたがい N_{csc} が減っている．とくに naive では T_{wakeup} によるストールが多発する分，パケットがバッファに滞在する期間が増え，チャネルをパワーゲーティングできる機会が減った．3 種類のスリープ制御方法を比べると，ideal，lookahead，naive の順に N_{csc} が多く，lookahead による効果が出ていることが分かる．

なお，最大スループットになっても， N_{csc} が 50% 以上あるトレースもある．これは，そもそも，リンクの利用率が 20% 程度でネットワークが飽和してしまうため，最大スループットになる負荷を与えても，パケットとパケットの到着間隔で十分スリープできるためである．

図 9 に $T_{breakeven} = 20$ の際の結果を示す． $T_{breakeven} =$

10 のときと比べ N_{csc} と N_{active} が若干減り N_{usc} が増えた．とくに uniform トラフィックにおける N_{usc} の増加が大きい．

6. まとめと今後の課題

オンチップルータのスタンバイ電力を減らすために，各チャネルの電源を走行時に動的にパワーゲーティングする．このとき， T_{wakeup} による性能への影響を隠蔽し，実際にスリープできる期間を増やすために，本論文では look-ahead ルーティングを応用したチャネルのスリープ制御を提案した．実トラフィックを用いた評価によって，提案手法は 5 サイクル以下の T_{wakeup} を隠蔽でき，先読みを行わないナイーブな方法と比べて実際にスリープできる期間を延ばすことができた．

今後は，走行時パワーゲーティングが可能な NoC を想定して，電力性能比の高いトポロジおよびルーティング方法について検討したい．例えば，仮想チャネル機構はスタンバイ電力の点で不利であったが，仮想チャネル単位でパワーゲーティングを施し，負荷に応じて仮想チャネルをアクティベートすれば，高いピーク性能を少ないスタンバイ電力で実現できる可能性がある．このような用途に適したルーティングとして，Descending Layers ルーティング¹⁰⁾ が考えられ，今後，これを NoC に適用して消費電力の評価を行う予定である．

謝辞

本研究は東京大学大規模集積システム設計教育研究センターを通じ，ローム (株)・凸版印刷 (株)・シノプシス株式会社・日本ケイデンス株式会社の協力で行われたものである．

本研究に関して貴重なご意見をいただきました芝浦工業大学 宇佐美教授に感謝いたします．

参考文献

- 1) Dally, W. J. and Towles, B.: Route Packets, Not Wires: On-Chip Interconnection Networks, *Proceedings of the Design Automation Conference*, pp. 684–689 (2001).
- 2) Ishikawa, M. and et. al.: A 4500 MIPS/W, 86 μ A Resume-Standby, 11 μ A Ultra-Standby Application Processor for

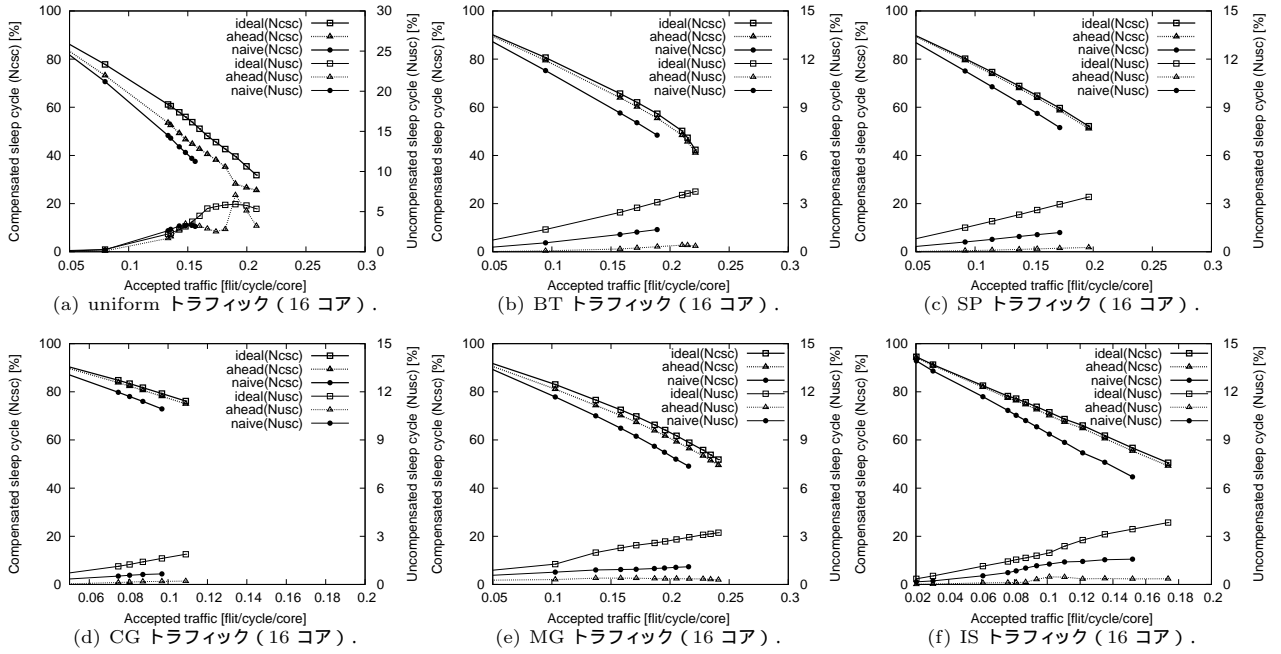


図 8 ideal, lookahead, naive における N_{csc} と N_{usc} の割合 ($T_{wakeup} = 2, T_{idledetect} = 2, T_{breakeven} = 10$) .

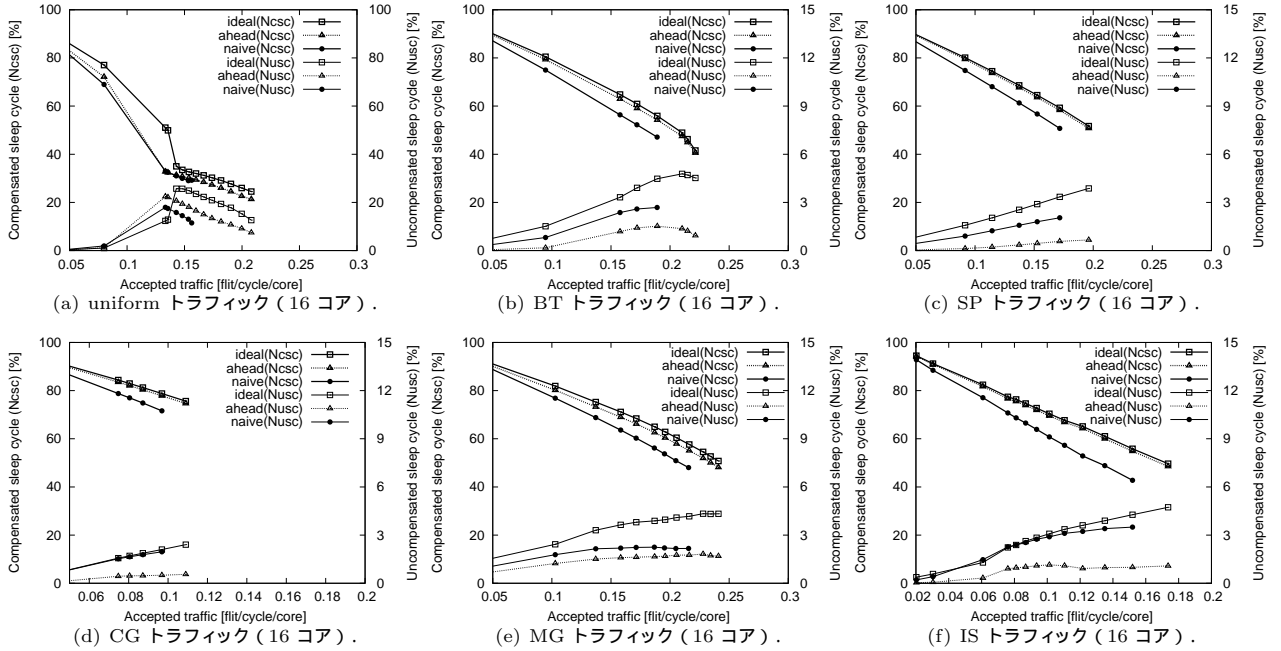


図 9 ideal, lookahead, naive における N_{csc} と N_{usc} の割合 ($T_{wakeup} = 2, T_{idledetect} = 2, T_{breakeven} = 20$) .

3G Cellular Phones, *IEICE Transactions on Electronics*, Vol. E88-C, No. 4, pp. 528–535 (2005).

3) Chen, X. and Peh, L.-S.: Leakage Power Modeling and Optimization in Interconnection Networks, *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 90–95 (2003).

4) Soteriou, V. and Peh, L.-S.: Design-Space Exploration of Power-Aware On/Off Interconnection Networks, *Proceedings of the International Conference on Computer Design*, pp. 510–517 (2004).

5) Banerjee, A., Mullins, R. and Moore, S.: A Power and Energy Exploration of Network-on-Chip Architectures, *Proceedings of the International Symposium on Networks-on-Chip*, pp. 163–172 (2007).

6) Matsutani, H. and et. al.: Performance, Cost, and Energy Evaluation of Fat H-Tree: A Cost-Efficient Tree-Based On-

Chip Network, *Proceedings of the International Parallel and Distributed Processing Symposium* (2007).

7) 関直臣ほか: MIPS R3000 における細粒度動的スリープ制御方式の提案, *情報処理学会研究報告 2007-ARC-173*, pp. 49–54 (2007).

8) Hu, Z. and et. al.: Microarchitectural Techniques for Power Gating of Execution Units, *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 32–37 (2004).

9) Dally, W. J. and Towles, B.: *Principles and Practices of Interconnection Networks*, Morgan Kaufmann (2004).

10) Koibuchi, M. and et. al.: Descending Layers Routing: A Deadlock-Free Deterministic Routing using Virtual Channels in System Area Networks with Irregular Topologies, *Proceedings of the International Conference on Parallel Processing*, pp. 527–536 (2003).