

## リアルタイムシステム向けオンチップネットワークスイッチの設計と実装

向後 卓磨<sup>†</sup> 水頭 一壽<sup>††</sup> 山崎 信行<sup>††</sup>

<sup>†</sup> 慶應義塾大学理工学部情報工学科

<sup>††</sup> 慶應義塾大学大学院理工学研究科開放環境科学専攻

〒 223-8522 横浜市港北区日吉 3-14-1

E-mail: †{kogo,suito,yamasaki}@ny.ics.keio.ac.jp

あらまし 近年では、組み込みリアルタイムシステムにおいても System-on-Chip (SoC) のように 1 チップに様々なモジュールが集積されたプロセッサが要求されている。1 チップに集積されるモジュール数が増加すれば、従来のチップ内通信網のバスでは高負荷時に多大な通信遅延が発生する。リアルタイムシステムでは、多大な遅延によって時間制約を保証できなくなるため、接続するモジュール数が増加しても通信遅延が増加しない Network-on-Chip (NoC) をチップ内通信網として採用することは良い方法と考えられる。しかし、NoC は従来のチップ内通信網のバスと比較してバッファや配線数が多いためチップ面積が大きくなる問題がある。本論文では、リアルタイム機能を持ち、面積を削減できるネットワークスイッチを提案する。提案手法が有効であることを確認するための評価を行った。

キーワード リアルタイム, 優先度, NoC

## Design and implementation of on-chip-network switch for real-time system

Takuma KOGO<sup>†</sup>, Kazutoshi SUITO<sup>††</sup>, and Nobuyuki YAMASAKI<sup>††</sup>

<sup>†</sup> Department of Information and Computer Science, Faculty of Science and Technology, Keio University

<sup>††</sup> Department of Computer Science, Graduate School of Science and Technology, Keio University

3-14-1 Hiyoshi, Kouhoku-ku, Yokohama, Kanagawa 223-8522 Japan

E-mail: †{kogo,suito,yamasaki}@ny.ics.keio.ac.jp

**Abstract** In recent years, processors like System-on-Chip (SoC) which various modules were integrated on have been demanded even in embedded real-time systems. If number of modules on chip increase, the bus which is classic on-chip interconnects causes significant latency when traffic becomes heavy. It is good approach to adopt Network-on-chip (NoC) which does not cause significant latency when number of connected modules increase, because it is problem for real-time systems not to be able to guarantee deadline due to significant latency. However, NoC has the problem that the area of the chip grows because the buffer and the amount of wire are larger than the bus which is common interconnects. In this paper, we propose the network-switch which has a real-time function and be able to reduce the area. we evaluated to confirm the proposal technique was effective.

**Key words** Real-time, Priority, NoC

### 1. はじめに

近年、チップ内通信の手段として Network-on-Chip (NoC) が注目されている。NoC は従来の通信網であるバス機構と比較して、接続モジュール数が増加しても性能が低下しない性質を持つため拡張性が高い。

リアルタイム処理を行うチップは多くのモジュールを搭載するため、拡張性の高い NoC との相性が良い。リアルタイムシステム向け NoC では、時間制約を保証するのに締切時間の近いパケットのトランザクションを早期に完了させるようにする。

これを実現するために、リアルタイムスケジューリングを通信に対して行う。スケジューラは締切時間の近いパケットに対して高い優先度を与える。またネットワークスイッチは高優先度パケットを受信するとトランザクションをプリエンブションする。これによって、締切時間の近いパケットに対して優先的にトランザクションを割り当てることができ、時間制約が保証される。通信のプリエンブションを実現するために、ネットワークスイッチ内では優先度に基づいたアービトレーションを行う。優先度付きアービトレーションでは物理リンク数だけ優先度を比較する必要がある。優先度の比較の計算量は  $O(n^2)$  なので、

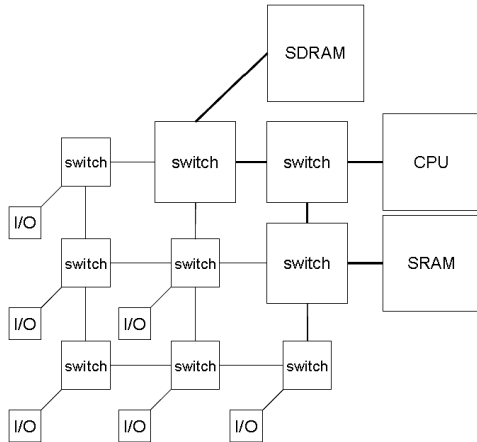


図1 異なるリンクバンド幅のスイッチの接続

物理リンク数の2乗のオーダーでアービトレーション論理が大きくなる。このことからリアルタイム性を持つネットワークスイッチは面積が大きくなる事がわかる。またNoCはバッファを多く必要とする上、配線量が多くなるので面積が大きくなり、チップ内の物理的配置が問題となる。NoCの面積において支配的なのはバッファと配線である。これらを必要最小限の量に抑えることでNoCの面積を最適化できる。

一般的にネットワークバンド幅を大きくするとネットワークの面積が大きくなるトレードオフの関係がある。System-on-Chip (SoC) では高いバンド幅を要求するSDRAMなどだけでなく低いバンド幅を要求するUARTなどのI/Oも多数存在する。高通信速度のモジュールの性能を十分に引き出すために高いネットワークバンド幅を確保することになるが、多くの低通信速度のI/Oにとって過剰であるため、バンド幅を全て使い切らないことがしばしば起こる。そのため、必要以上にネットワークの面積が大きくなってしまふ。そこで、図1に示すように低通信速度モジュールと高通信速度モジュールとで分離し、低通信速度のモジュールが接続されるスイッチのバンド幅を削減することでネットワークリンクのバンド幅を最適化して、面積を削減する方法を提案する。

## 2. 関連研究

リアルタイム機能を持つNoCの研究について述べる。

### 2.1 RMT Processor

分散リアルタイム処理用システムLSIであるResponsive Multithreaded Processor (RMT Processor) [1] は以下に示すようなリアルタイム処理機能、リアルタイム通信機能、コンピュータ用周辺機能、各種周辺制御機能、これらの機能全てを1チップに集積 (System on Chip) している。

- リアルタイム処理機能

Responsive Multithreaded Processing Unit (RMT PU) [2]~[4] を搭載している。

- リアルタイム通信機能 (Responsive Link II [5])

リアルタイム通信規格 Responsive Link [6] に準拠したI/Oを搭載している。

- コンピュータ用周辺機能

PCI-X I/F, IEEE1394 I/F, Ethernet I/F, UART I/F, 外部バスI/F, DDR SDRAM I/F, DMA Controller等を搭載している。

- 各種周辺制御機能

PWM Generator I/F, Pulse Counter I/F, SPI I/F, Parallel I/O I/F等を搭載している。

### 2.2 リアルタイム NoC

RMT Processorでは、チップ内通信網として分割バスを採用している。しかしバスでは高負荷時に大きな通信遅延が発生し、トランザクションの実行時間予測性が低下する。分割バスをリアルタイム機能を持つNoCで置き換えたことによって、高優先度トランザクションの遅延を削減し、トランザクションの実行時間予測性を高めることができた。また、既存の分割バスが抱えていたバス間のトランザクションが頻繁に発生するような状況下において極端な性能の低下が発生することが防ぐことができた。一方で、チップの面積は分割バス時と比較して13.8倍になり、ネットワークを構成するスイッチとNetwork Interfaceによって大幅に面積が増加した。

## 3. リアルタイムスイッチ

オンチップネットワークスイッチのリアルタイム機能の設計および実装を行った。

### 3.1 優先度制御

チップ内でリアルタイム通信を実現するために、スケジューラは通信のリアルタイムスケジューリングを行う。スケジューリングを行うためには通信においてプリエンプション機構が必要である。このプリエンプションは締切時間の近いパケットに優先的にトランザクションを割り当てる目的で行われる。スケジューラは締切時間や周期をもとにパケットに優先度情報や制御情報を付加する。通常、高い優先度は早い締切時間または短い周期を表す。付加された情報をもとにネットワークスイッチではパケットのプリエンプションが行われる。このパケットのプリエンプションは、スイッチに後から到着した高優先度パケットが先に到着した低優先度パケットよりも先にパケットの転送が行われることから、パケット追い越し [6] と呼ぶ。

パケット追い越しは、スイッチ内の図2に示す優先度付きアービタにおいて優先度付きアービトレーションを行うことで、実現される。

優先度付きアービトレーションは以下のアルゴリズムで行う。

- パケットの優先度比較を行う。基準とする側の優先度が比較対象の優先度以上であれば比較結果はHighとする。
- 優先度の比較結果と比較対象の優先度が有効であるかを表すsw\_req\_信号の論理和をそれぞれ求める。
- 2.で求めた結果の全てに対する論理積を求めると、要求を出しているパケットのうち最も優先度が高ければ結果はHighになる。
- 同一優先度のパケットが存在すると結果がHighにな

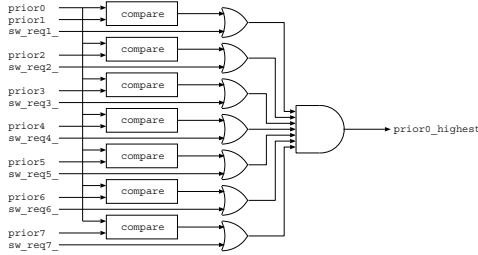


図2 優先度付きアービタ

るパケットが複数存在することになるが、この場合ラウンドロビンでトランザクション権を順に回す。

また、低優先度のパケットは高優先度のパケットにトランザクションをブロッキングされ続け、スタベーションを起こす問題がある。これを解決するためにスイッチ内でバッファリングされたパケットはブロッキングされたサイクル数に応じて、動的に優先度を付け替える。ブロッキング時間が長いと優先度が上がるため、スタベーションを起こさない。これを優先度のエイジングと呼ぶ。

本論文では、パケット追い越しと優先度のエイジングをネットワークスイッチに実装した。

### 3.2 アーキテクチャ

設計および実装したスイッチの特徴は以下の通りである。

- 優先度付きアービトレーション
- 物理リンク数 4
- 各物理リンクに対してバッファ4個
- Virtual Cut Through スイッチング
- Stop&Go フロー制御

パケット追い越しを実装するために、優先度付きアービトレーションが必要であることは前で述べた。

パケット追い越し機能を持つスイッチにおいて、物理リンク数を  $n$  としたとき、パケット追い越し機能のための優先度の比較の計算量は  $O(n^2)$  となり、物理リンク数が多いときはこれがクリティカルパスとなる。後述する提案手法においてネットワークスイッチは高い動作周波数を要求するため、物理リンク数を 4 とした。

バッファの数が不十分だと、通信遅延が大きくなるだけでなく、パケット追い越しが十分に機能しなくなるためパケット到着時間を保証できなくなってしまふ。バッファ数が 4 を超えると、バッファ数に対するバンド幅利用率の向上率が低くなる。そのため、各物理リンクに対するバッファ数を 4 とした。

スイッチはバッファを十分持つため、スイッチングに Virtual Cut Through を採用する。

また、動作周波数を高くすることを考慮して、フロー制御には単純な Stop&Go を採用した。

スイッチの構成を表すブロック図を図 3 に示す。スイッチは 4 段パイプラインとなっている。

図 3 の各ブロックの機能は以下の通りである。

**dec** デコードユニット. 受信したデータ信号をデコード (増

幅) する。

**buf** バッファ. デコードユニットから転送されたパケットを格納する。

**ctrl** バッファコントロールユニット. バッファの空き状況进行管理し、受信パケットを空いているバッファに割り当てる。トランザクション権を得たら最高優先度のパケットを転送する。また優先度のエイジングを行う。

**parb** 優先度付きアービタ. 優先度付きアービトレーションを行う。各入力物理リンクにあるアービタは 4 個のバッファそれぞれが格納するパケットの優先度を比較する。各出力物理リンクにあるアービタは出力リンク割当権を決定する。

**route** ルーティングユニット. 各バッファに格納されているパケットのルーティングを行い、その結果を保持する。

**flow** フロー制御ユニット. 下位スイッチから空きバッファ数の通知を受ける。送信先のバッファに空きがない場合はアービトレーションの結果を変更する。

**enc** エンコードユニット. 送信するデータ信号をエンコード (増幅) する。

### 3.3 パケットフォーマット

以下の構成の 64Byte 固定長パケットを使用する。パケット全体から見てペイロードのサイズが比較的小さいパケットを使用するのは、ブロッキングによるパケットの通信遅延を小さくするためである。このような設計はスループットが低くなる問題がある。しかし、ペイロードのサイズを大きくすると、低優先度パケットによって高優先度パケットの転送が長時間妨げられる問題がある。このような状況は実行時間予測性の低下につながるため、リアルタイムシステムには小さい固定長パケットが向いているといえる。

- ヘッダ (4byte)
  - ルーティングアドレス (8bit)
  - 優先度 (8bit)
  - 制御情報 (16bit)
- ペイロード (60byte)

優先度を 8bit としたのは、Rate Monotonic スケジューリングは 256 レベル以上の優先度があればスケジューリング可能 [7] ためである。

## 4. 提案手法

本節では、バンド幅を最適化することでチップ内のネットワークの面積を削減するためのネットワークスイッチの設計および実装を行った。

### 4.1 物理リンクの構成

図 1 のようにスイッチは異なるリンクバンド幅のスイッチ間の接続を可能とするために、物理リンクを図 4 のようにシリアル伝送路であるレーンを多重化することで構成した。このリンク内のレーンの多重度を上げることで、容易にバンド幅を大きくできる。また、この方法はレーンの切り取りを行うことで、異なるバンド幅のスイッチ間の接続ができるので、モジュールを追加する必要がない。さらにシリアル化によって配線量が大幅に削減できる。

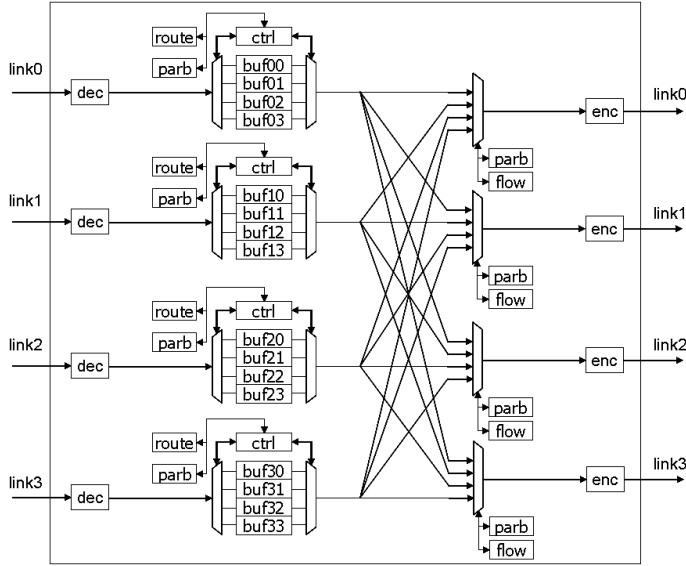


図3 スイッチ内部のブロック図

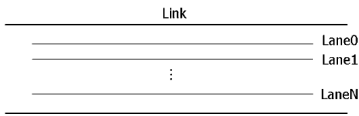


図4 リンクの構成

リンクをシリアル化することで、スループットが低下するため、配線長の小さいNoCにおいてシリアルリンクは不利である。しかし、レーンの多重度を上げることによってスループットが大きくなるため、この問題は解決できる。これはパラレルリンクの場合のビット幅を増やす方法と似ているが、ビット幅を増やすことでデータ線の同期をとるために動作周波数が低下する問題がある。シリアルリンクの場合、リンクの内の各レーンは互いに独立しているため同期を取る必要がない。そのため動作周波数を落とさず、レーン多重化によってバンド幅を向上させることができる。つまりスループットを維持するためにレーンの多重化を行い動作周波数を高くする。

#### 4.2 パケット多重化

リンク内のレーンを多重化することでリンクバンド幅を向上できる。ここでは、多重化したレーンにパケットを割り当てる方法を説明する。

スイッチ内では全物理リンクに対してアービトレーションが行われる。トランザクション権を獲得した入力物理リンクからパケットの転送が開始する。ここで、その入力物理リンクのバッファ内で、最大優先度のパケットと出力物理リンクが同じものがあれば空いているレーンを用いることで同時に転送する。フロー制御としてStop&Goを採用しているため、同時に転送して良いパケット数は、スイッチのリンク内レーン数と転送先

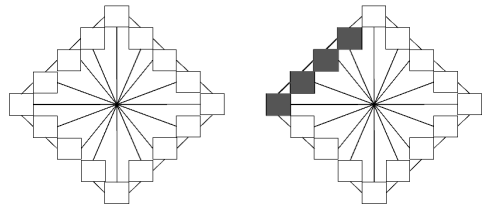


図5 比較する2種類のNoCの構成

スイッチの空きバッファ数の最小値となる。このパケット多重化はバッファコントロールユニットが行う。

## 5. 評価

設計および実装したネットワークスイッチがリアルタイム性を持ち、またネットワークの面積削減するのに有効であること示すために評価を行った。

### 5.1 評価環境

Cadence社のNC-Verilogを用いてRTLシミュレーションで評価を行った。

図5に示すレーン多重度1のスイッチ16個によって構成されるNoC Aと16個スイッチの内12個がレーン多重度1、4個がレーン多重度2によって構成されるNoC Bの2種類のネットワークを用いて評価を行った。

各ネットワークが共通するパラメータを表1に示す。

トポロジ	Spidergon
ルーティングアルゴリズム	最短経路ルーティング
ノード数	16

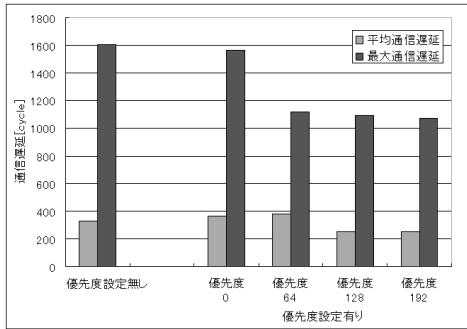


図6 優先度制御による通信遅延への影響

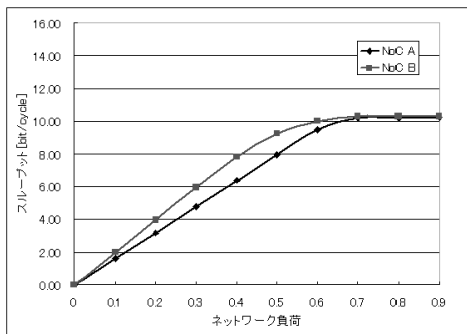


図7 ネットワークバンド幅とスループット

パケットを送受信するノードは一定周期でパケットを送信し、送信の際にルーティングアドレスをランダムに決定する。また、無限にバッファを持つと仮定し、いつでもパケットを受信可能である。

### 5.2 優先度制御

NoC A を用いて通信遅延の評価を行った。図6はネットワーク負荷30%時のパケットに優先度設定していないときの通信遅延とパケットに優先度を設定したときの通信遅延を表している。

優先度の設定を行わない場合は、ネットワークの各スイッチでラウンドロビンでアービトレーションが行われているため、長時間ブロッキングされるパケットが存在する。そのため、優先度設定がない場合の最大通信遅延は大きい。一方、優先度を4段階で設定した場合は、優先度が大きくなるにつれて通信遅延が小さくなっていることがわかる。ネットワークの負荷がさらに大きくなると、ブロッキングされるパケットが増加し、優先度の設定の影響より大きくなると考えられる。

### 5.3 バンド幅とスループットの関係

NoC A と NoC B を用いてスループットの評価を行った。図7はネットワーク負荷毎のスループットを表している。

NoC B はネットワークスイッチのうち25%がリンクバンド幅が他のスイッチの2倍であるため、ネットワークバンド幅はNoC A より25%大きい。ネットワーク負荷が10%~40%においてNoC B のスループットはNoC A より25%程度大きい。こ

れはネットワークのある部分をバンド幅の大きいスイッチで置き換えてネットワークバンド幅を増加させることで、増やしたバンド幅の分だけスループットが向上するということである。リンクバンド幅がネットワーク全体で不均一になる場合、ネットワークポロジやトラフィックによって、スループットに影響が出る。しかし、パケット多重化を阻害しやすいポロジかつトラフィックで十分にスループットを向上させていることから、提案手法が有効であることがわかる。

### 5.4 論理合成

本節では実装したスイッチの評価を行う。TSMC社の130nmプロセスのライブラリを使用し、Synopsys社のDesign Compilerにより合成を行った。レーン多重度1のスイッチAとレーン多重度2のスイッチBはともに搭載するバッファの容量が同じである。

表2 スwitchの面積と動作周波数

	レーン多重度	面積 [ $\mu\text{m}^2$ ]	動作周波数 [MHz]
スイッチ A	1	63,217	935
スイッチ B	2	65,463	900

バッファ数を変えずにレーン多重度を1から2にした場合、パケット多重化の追加論理の分として2.2%面積が増加した。同じ理由で動作周波数も低下した。スイッチの各物理リンクのバッファの数はレーン多重度の2倍以上必要となる。スイッチの面積のうちバッファが90%を占めることを考慮すると、レーン多重度を2倍にすることでスイッチの面積はおおよそ倍程度になるといえる。つまり、NoC B のようにバンド幅を25%増加させるために面積は25%程度の増加に留められるということがいえる。

## 6. 結 論

本論文では、パケット追い越しと優先度エイジング機能を持つネットワークスイッチの設計と実装を行った。また異なるバンド幅のスイッチ間の接続を容易とする設計と実装を行った。異なるバンド幅のスイッチによってNoCを構成することでバンド幅を最適にし、ネットワークの面積を削減できることを確認した。

今後の課題としては、シリアル化によって増大した通信遅延を削減することが考えられる。これはパケット長を小さくなどの方法で解決できる。また、多重するパケットの優先度を考慮していないため、優先度逆転現象が起こる問題がある。これは優先度付きパケット多重化の論理を追加することで解決できる。またフロー制御にStop&Goを用いたが、優先度を考慮したフロー制御を行うことで、高優先度パケットの通信遅延が削減できる。

謝辞 本研究の一部は科学技術振興機構CRESTの支援によるものであることを記し、謝意を表す。また、本研究の一部は文部科学省グローバルCOEプログラム「環境共生・安全システムデザイン」の先導拠点によるものであることを記し、謝意を表す。

## 文 献

- [1] Y. Nobuyuki: "Design concept of responsive multithread processor for distributed real-time control", *Journal of Robotics and Mechatronics*, **16**, 2, pp. 194–199 (2004).
- [2] 薄井, 内山, 伊藤, 山崎: "Responsive multithreaded processor における実時間処理用命令供給機構", *電子情報通信学会技術研究報告実時間処理に関するワークショップ*, **2004**, 33, pp. 15–20 (2004).
- [3] 薄井, 内山, 伊藤, 山崎: "Responsive multithreaded processor の命令供給機構", *情報処理学会論文誌コンピューティングシステム*, **45**, 11, pp. 105–118 (2004).
- [4] 伊藤, 山崎: "Responsive multithreaded processor の命令実行機構", *情報処理学会論文誌コンピューティングシステム*, **44**, 11, pp. 226–235 (2003).
- [5] 山崎: "分散制御用リアルタイム通信 responsive link の設計および実装", *情報処理学会論文誌: コンピューティングシステム*, **45**, pp. 50–63 (2004).
- [6] [http://www.itsecj.ipsj.or.jp/ipsj-ts/02\\_06/toc.html](http://www.itsecj.ipsj.or.jp/ipsj-ts/02_06/toc.html).
- [7] J. W. Liu: "REAL-TIME SYSTEMS", Prentice Hall (2000).