

リンク長及びレイテンシ制約下での ネットワークオンチップのトポロジ自動生成

谷田 英生[†] 吉田 浩章^{††,†††} 松本 剛史^{††} 藤田 昌宏^{††,†††}

[†] 東京大学大学院工学系研究科電気系工学専攻 〒113-8656 東京都文京区本郷 7-3-1

^{††} 東京大学大規模集積システム設計教育研究センター 〒113-0032 東京都文京区弥生 2-11-16

^{†††} 科学技術振興機構 戦略的創造研究推進事業 CREST

E-mail: {tanida,hiroaki,matsumoto}@cad.t.u-tokyo.ac.jp,fujita@ee.t.u-tokyo.ac.jp

あらまし 半導体プロセスの微細化に伴い、SoCの性能はよりインターコネクットの性能に影響される部分が大きくなっている。多くのNoC（ネットワークオンチップ）のアーキテクチャが提案されているが、自動的にアプリケーションに応じた最適なNoCを決定しその上に機能ブロックを配置することは、実現されていない。本稿は、アプリケーションの通信要求に応じて、自動的に各機能ブロックを配置し、NoCトポロジを生成する手法を提案する。問題を整数線形計画法により定式化した従来手法[5]は、消費電力・面積の最適化を実現したが、動作周波数に影響を与える一ホップあたりのリンク長、および、レイテンシの保証を実現していなかった。提案手法は、各機能ブロックを配置する際に低レイテンシで通信する機能ブロック間の距離に、NoCトポロジの生成の際に接続する機能ブロック・スイッチ間距離に制約を加え、それらの性能が保証された設計の生成を実現する。

キーワード ネットワークオンチップ、整数線形計画法、性能保証、フロアプランニング

Automatic generation of Network-on-Chip topology under link length and latency constraint

Hideo TANIDA[†], Hiroaki YOSHIDA^{††,†††}, Takeshi MATSUMOTO^{††}, and Masahiro FUJITA^{††,†††}

[†] Dept. of Electrical Engineering and Information Systems, The University of Tokyo

7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656 Japan

^{††} VLSI Design and Education Center, The University of Tokyo

2-11-16 Yayoi, Bunkyo-ku, Tokyo, 113-0032 Japan

^{†††} CREST, Japan Science and Technology Agency

E-mail: {tanida,hiroaki,matsumoto}@cad.t.u-tokyo.ac.jp,fujita@ee.t.u-tokyo.ac.jp

Abstract With wire delay becoming dominant compared to transistor delay in deep-submicron era, the performance of SoC is more affected by interconnect. Although many NoC (Network-on-Chip) architectures which improve interconnect performance are proposed, automatically finding the most efficient one for a given application and mapping the function blocks onto it, is still an open issue. This paper proposes a method for generating a custom NoC which meets communication link-length and latency requirements. Additional constraint for floor-planning and interconnect architecture generation, to existing integer-linear-programming-based approach [5], enables link-length and latency requirement to be met in the generated NoC architecture.

Key words Network-on-Chip, integer linear programming, guaranteed performance, floor planning

1. はじめに

半導体製造プロセスの微細化により、SoC (System-on-a-Chip) と呼ばれる大規模なシステムを単一のチップ上に実装し

た半導体製品の製造が可能になった。一方、プロセス微細化に伴い、トランジスタに由来する遅延が減少する一方、システム全体を接続するグローバル配線の遅延は相対的に増大する傾向にあり、チップ内の各機能ブロックを効率的に接続する技術が

必要とされている。チップ内の各機能ブロックを効率的に接続するために、NoC (Network-on-Chip/オンチップネットワーク) と呼ばれる考え方が提唱され [2], 特に多数の機能ブロックを含む SoC の設計において、注目を集めている。

本稿では、第 2 節において NoC 技術の概要について述べ、NoC 技術を使用した SoC を設計する際の問題点について記す。そのような問題点をふまえた上で、アプリケーションに応じた NoC の設計支援を行う既存手法を第 3 節において紹介し、既存手法の問題点を挙げる。その後、第 4 節において動作周波数、レイテンシ性能を保証するかたちでの NoC トポロジ設計支援手法を提案、第 5 節において提案手法の評価を行う。最後に、第 6 節においてまとめを行う。

2. Network-on-Chip

NoC は図 1 に示すように、チップ内の機能ブロックにネットワークインターフェースを取り付け、スイッチ同士およびスイッチ・インターフェース間を接続することにより、チップ内に構成されるネットワークである。以下に本稿で使用する NoC を構成する要素の名称の定義を記す。

スイッチ リンク同士を接続して、トラフィックを交換するノード

リンク トラフィックを転送することが可能な配線。本稿では、全二重に通信が可能であるとする。つまり、同一リンク上で方向の違う通信が同時に可能であるとする

ネットワークインターフェース CPU や DSP, Memory などの各機能ブロックをリンクに接続するためのインターフェース

NoC は以下のような点において、従来、チップ内の機能ブロック間の接続に用いられてきた共有バスにおける問題を解決している。

- スwitch同士およびスswitch・インターフェース間の接続は基本的に 1 対 1 となっており、デバイスの増加による入力容量の増加などの影響が生じない
- 同一のリンクを使用しない複数の通信は、同時に行うことが可能である
- 長いリンクをスswitchによって分割することにより、分割後の各リンクに違うデータを保持させることが可能となり、パイプラインが可能となる

NoC のトポロジ (リンク・ネットワークインターフェース・スswitchの接続関係) には、大きな自由度が存在し、いくつかの代表的なトポロジが提案されているが、それらのトポロジには実装に必要な面積コストや消費電力と、帯域やレイテンシなどの性能にトレードオフがあり、アプリケーションの要求に応じた選択を行う必要がある。また、オフチップネットワークに比較して、NoC のトポロジは FPGA などの特殊なアーキテクチャの上に実装しない限り、チップ製造時に決定しておく必要があり製造後の変更を行うことは困難である。そのため、NoC はオフチップネットワークに比較して実装前の性能評価と設計支援がさらに重要であると考えられる。

2.1 NoC のトポロジ, 各種性能のトレードオフ

NoC のトポロジはさまざまなものが提唱されているが、その

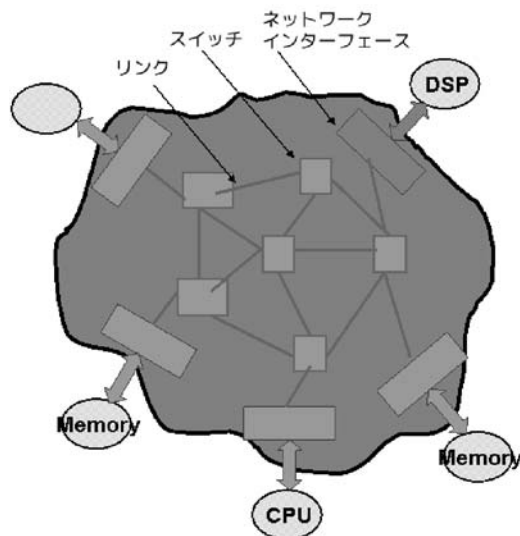


図 1 NoC の構成要素 (スイッチ, リンク, ネットワークインターフェース) ([1] より改変)

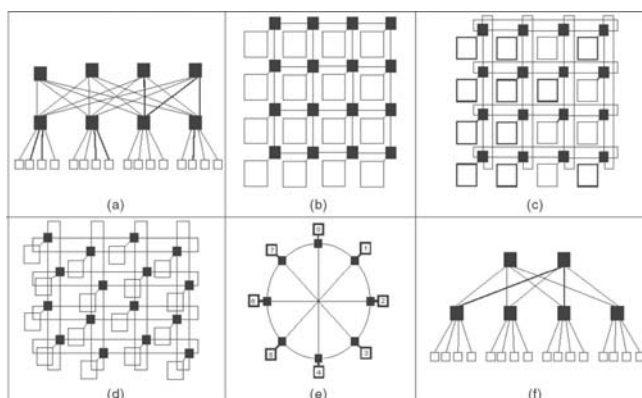


図 2 各種 NoC アーキテクチャ (a) SPIN, (b) CLICHÉ (2-D mesh), (c) Torus, (d) Folded torus, (e) Octagon, (f) BFT [4]

中で代表的なものとして図 2 に示すようなものが挙げられる。

これらのトポロジは [4] において、転送可能なトラフィック量や、レイテンシ、消費電力、実装面積が比較評価されているが、一般に広帯域・低レイテンシなトポロジ (例:SPIN, Octagon) ほど消費電力が大きく、実装面積が増加する傾向となっている。

このような傾向を考慮すると、アプリケーションの要求に応じたトポロジや各パラメータの決定が重要になると考えられる。また、場合によっては、チップ全体のネットワークに均質なネットワークを採用するのではなく、周辺の機能ブロックのトラフィック要求に応じたトポロジを柔軟に作成する必要が生じる場合も考えられる。しかし、これらの決定を設計者自身が行うことは、NoC 内に存在する機能ブロックの数が増え調整可能なパラメータの数が増えるにつれ、困難になると考えられる。本稿では、アプリケーションの要求性能に応じて消費電力・実装面積を最適化された形で機能ブロックを配置し、機能ブロック・スswitchの接続トポロジを自動生成する手法を提案する。

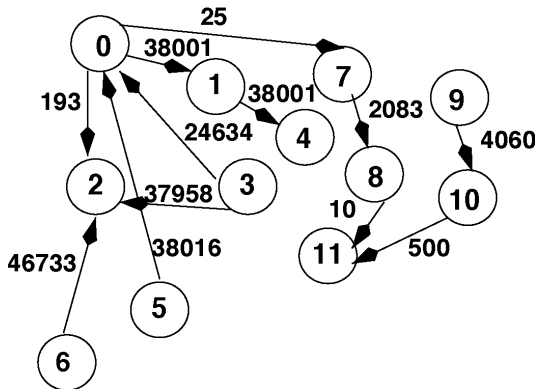


図3 Communication trace graph [5]

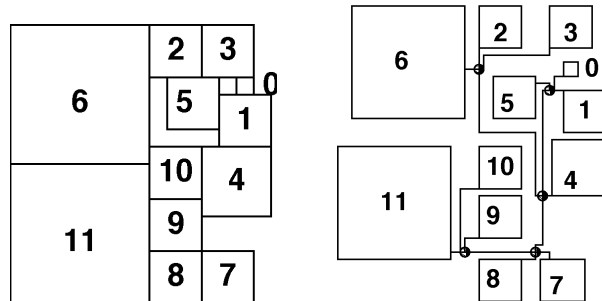


図4 通信要求に応じて生成されたフロアプラン [5]

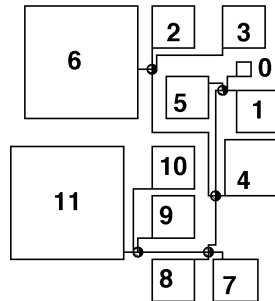


図5 図4のフロアプラン上に生成されたNoCトポロジ [5]

3. Network-on-Chip の設計支援に関する既存研究

[5]においては、各機能ブロック間の通信要求が与えられた際に、通信路の帯域・レイテンシに関する制約を満たした上で、消費電力・実装面積の線形結合に対応するコスト関数を最小化するフロアプラン・トポロジを生成する手法が提案されている。

3.1 問題への入力

問題への入力としては、図3に示すような各機能ブロック間の通信関係を表すグラフ（以下、CTG: communication trace graph とする）が与えられる。グラフのノードが各機能ブロックに対応し、グラフのエッジが機能ブロック間の通信が生じることに対応する。各エッジは、通信帯域、レイテンシ制約を表す二つの重みを持っている。また、各機能ブロックのサイズ (W_i, H_i) も与えられる。

3.2 フロアプランの生成

図3に示した、CTGから図4に示すような、フロアプランを生成する。この問題は、以下に示すような評価関数を最小化する、混合整数線形計画問題として定式化されている。

$$\alpha \left[\sum_{\forall(u,v) \in E} dist(u,v) \cdot \Psi_l \cdot \frac{\omega(e)}{\sigma^2(e)} \right] + \beta [X_{max} + Y_{max}] \quad (1)$$

上式で、第一項は消費電力に対応する項^(注1)であり、第二項

(注1): 但し、要求レイテンシの二乗を分母として持つ。これは、既存研究 [5] では制約条件として与えていない、レイテンシに対する要求を考慮することを目的としたものと考えられる

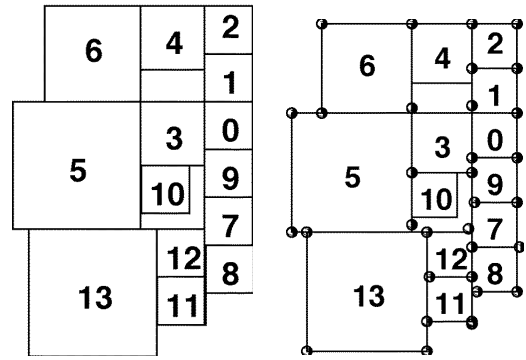


図6 フロアプラン上にスイッチが配置される可能性のある箇所 (○) [5]

は実装面積に対応する項である。 $dist(u, v)$ は、機能ブロック u, v 右上端同士のマンハッタン距離を与え、 Ψ_l は単位帯域・長さあたりの消費電力を表す。 $\omega(e), \sigma(e)$ はそれぞれ、要求帯域・レイテンシを与える。 X_{max}, Y_{max} はレイアウト全体のサイズを表す。 α, β は、各項の重み付けを与える定数である。ここでは、問題を簡単にするために、 $X_{max} \times Y_{max}$ に対応する実装面積を最小化問題を、 $X_{max} + Y_{max}$ を最小化する問題としている。

独立変数としては、各機能ブロック v_i の左下端の座標に対応する、 $(X_{i,min}, Y_{i,min})$ が存在する。また、その従属変数として、機能ブロックの寸法 (W_i, H_i) を加えた、 $(X_{i,max}, Y_{i,max})$ が与えられる。また、評価関数に含まれる $dist(u, v)$ などの変数も複数の機能ブロック i に対応する $X_{i,min}, Y_{i,min}$ の従属変数として与えられる。

制約条件としては、以下のような条件が存在する。二つの機能ブロック $(v_i, v_j \in V)$ が重なり合って配置されない条件は、次式のように与えられる。

$$\begin{aligned} X_{i,min} \geq X_{j,max}, X_{j,min} \geq X_{i,max}, \\ Y_{i,min} \geq Y_{j,max}, Y_{j,min} \geq Y_{i,max} \end{aligned} \quad (2)$$

各機能ブロックが、全体のレイアウトの寸法内部に配置されるという条件は、次式のように与えられる。

$$X_{i,max} \leq X_{max}, Y_{i,max} \leq Y_{max} \quad (3)$$

以上のように定式化された混合整数線形計画問題を解くことによって、フロアプランの出力が得られている^(注2)。

3.3 トポロジの生成

図3に示したCTGを元に、生成されたフロアプラン上に図5に示すようなスイッチ・リンク・各機能ブロックの接続関係を定義するネットワークトポロジを生成する。

この際に、スイッチが置かれる可能性のある場所を、図6に示すようにフロアプラン上で各機能ブロックを囲む bounding box の頂点に限定している。また、スイッチ・リンクの幅、サイズは各機能ブロックに対して十分に小さいものとしてモデル化を行っている。

(注2): 詳細な定式化は紙面の都合上、省略したが文献 [5] を参考にされたい

トポロジに対応する独立変数は、以下のようなものが与えられる。但し、 r_i を i 番目のスイッチ、 $p_{i,j}$ を r_i の j 番目のポートとする ($0 \leq j < \nu$, ν はスイッチの持つポート数を表す定数)。 $NR_{k,i,j}$ 各スイッチと機能ブロックの接続関係を表す。機能ブロック v_k が、スイッチのポート $p_{i,j}$ に接続された場合のみ 1 となり、その他の場合には 0 となる。

$RR_{i,j,k,l}$ スwitchのポート同士の接続関係を表す。 $p_{i,j}, p_{k,l}$ 間が接続された場合にのみ 1 となり、その他の場合には 0 となる。

$O_{i,j,k,l}$ 機能ブロック v_i から v_j へのトラフィックがスイッチのポート $p_{k,l}$ から出力される場合のみ 1 となり、その他の場合には 0 となる。

$I_{i,j,k,l}$ 機能ブロック v_i から v_j へのトラフィックがスイッチのポート $p_{k,l}$ に入力される場合のみ 1 となり、その他の場合には 0 となる。

上記の独立変数によって以下のような従属変数が与えられる。

$BO_{k,l}$ ポートから出力される帯域の総和に対応する。

$$BO_{k,l} = \sum_{\forall e_m = v_i, v_j \in E} \omega(e_m) * O_{i,j,k,l} \quad (4)$$

となる。

$BI_{k,l}$ ポートに入力される帯域の総和に対応する。

$$BI_{k,l} = \sum_{\forall e_m = v_i, v_j \in E} \omega(e_m) * I_{i,j,k,l} \quad (5)$$

となる。

$Z_{i,j,k,l,m,n}$ トラフィックの接続関係を表す。 v_i, v_j 間のトラフィックが、 $p_{k,l}$ を通り、 $p_{k,l}, p_{m,n}$ が接続されている場合のみ 1 となり、その他の場合は 0 となる。

$$Z_{i,j,k,l,m,n} = O_{i,j,k,l} \times RR_{k,l,m,n} \quad (6)$$

で与えられ、非線形であるが、以下のような条件を与えることにより線形化することができる。

$$O_{i,j,k,l} + RR_{k,l,m,n} \geq 2 \times Z_{i,j,k,l,m,n} \quad (7)$$

$$O_{i,j,k,l} + RR_{k,l,m,n} \leq Z_{i,j,k,l,m,n} + 1 \quad (8)$$

トポロジ生成は、以上の変数を用いて与えられる、動的消費電力に対応する以下の評価関数の最小化を目的とする整数線形計画問題となっている。

$$(P_R + P_L) \quad (9)$$

P_R, P_L はそれぞれ、スイッチ・配線における消費電力に対応する。

スイッチにおける、消費電力 P_R は以下のようにモデル化されている。

$$P_R = \Psi_i \cdot \sum_{\forall r_i \in R} \sum_{\forall p_{i,j}} BI_{i,j} + \Psi_o \cdot \sum_{\forall r_i \in R} \sum_{\forall p_{i,j}} BO_{i,j} \quad (10)$$

但し、 Ψ_i, Ψ_o はそれぞれ、各入力ポート・出力ポートにおける、単位トラフィックあたりの消費電力である。

また、リンクにおける消費電力 P_L は以下のようにモデル化されている。

$$P_L = \Psi_L (\sum_{i,j,k,l,m} \omega(i,j) \cdot RD_{k,m} \cdot Z_{i,j,k,l,m,n} + \sum_{i,j,k,l} ND_{i,k} \cdot \omega(i,j) \cdot NR_{i,k,l} + \sum_{i,j,k,l} ND_{j,k} \cdot \omega(i,j) \cdot NR_{j,k,l}) \quad (11)$$

但し、 Ψ_L は、単位リンク長・単位帯域あたりのダイナミック消費電力に対応する。 $RD_{k,m}$ はスイッチ k, m 間のマンハッタン距離を表し、 $ND_{i,k}$ は、機能ブロック・スイッチ間のマンハッタン距離を表す。

また、与えられる制約条件は以下のようなものとなる^(注3)。

- スwitchの各ポートにおける帯域が、最大値を越えない
- スwitchの各ポートは、他のスswitchのポート一つもしくは機能ブロックに接続される
- 各機能ブロックは一つのスswitchの一つポートにのみ接続される
- 要求された二つの機能ブロック v_s, v_d 間の通信エッジ全てについて、スswitchを経由した経路

$$p = \{(v_s, r_a), (r_a, r_b), \dots, (r_z, v_d)\} \quad (12)$$

が存在する。

- 経路のホップ数が機能ブロック間の通信のレイテンシ制約以下に合致する

以上のような定式化を行った、混合整数線形計画問題を解くことによって、図5に示したようなフロアプランに応じて消費電力を最適化したスswitch・機能ブロック間のリンクによる接続関係(トポロジ)が得られている。また、消費電力やスswitchの数においてアプリケーションに特化したトポロジが規則的なメッシュトポロジに対して優位であることが示されている。

4. ホップ長を制限した、帯域・レイテンシ制約を満たす NoC トポロジの生成

4.1 既存手法の問題点

第3節で紹介した手法においては、NoCに接続される各機能ブロック間の通信要求(帯域、クロックサイクル数でのレイテンシ)を充足する形での、消費電力・実装面積を最適化したフロアプランおよびトポロジの生成を実現している。しかし、既存手法においては、以下のような問題点が存在する。

既存手法においては、どのような長さの配線に対しても、一クロックでのデータ転送を前提としており、各ホップの配線長に制約が設けられていない。このため、長いリンクが存在した場合には、システムの動作周波数が低下して、性能が低下してしまう可能性があると考えられる。通常、一クロックでのデータ転送が可能な配線長は、ネットワークの動作周波数が上昇するにつれ短くなる傾向があり、各ホップの配線長に制約を与えることにより、システムの動作周波数を保証する必要があると考えられる。実際に、Intel TeraFLOPS Processor [6] におい

(注3) : ここでは、紙面の都合上、定式化された制約の記述は省く。詳細は [5] を参照されたい

ては、各ホップの配線長に制約を設けることにより 4GHz 以上の動作周波数を実現している。

4.2 提案手法

提案手法は、従来手法に対してフロアプラン及びトポロジ生成のそれぞれの工程に関して、長いリンクの生成を制限する制約を追加することにより、サイクル数によるレイテンシと動作周波数に影響を与える、一ホップあたりの配線長の最悪値が保証された NoC の生成を実現する。

4.2.1 フロアプラン生成に対する追加制約

各ホップの配線長を制約した上で、クロック単位のレイテンシ制約（ホップ数）を満たす条件として、通信を行う二つのノードの距離が各ホップの配線長とホップ数の積より短いことが要求される。よって、フロアプランの工程に対して以下のような式で表される制約を追加する。

$$\forall (u, v) \in E \quad dist(u, v) \leq D_{max} \times \sigma(e_{u,v}) \quad (13)$$

上式の中で、 D_{max} は一ホップあたりのリンク長の最大値である。また、 $\sigma(e_{u,v})$ は機能ブロック間の通信において許される最大レイテンシである。ただし、通信にレイテンシ制約がないノードの組 u, v については、 $\sigma(e_{u,v})$ は十分に大きな値とする。

4.2.2 トポロジ生成に対する追加制約

トポロジ生成の段階においては、第 3.3 節で与えられた、スイッチとスイッチ、機能ブロックとスイッチの接続に対応する 1,0 の独立変数 $RR_{i,j,k,l}, NR_{i,j,k}$ に対し、以下のような制約を追加する。

互いのマンハッタン距離が D_{max} 以内に存在しないスイッチ r_p のポート $p_{i,j}$ 、機能ブロックの v_k の全ての組み合わせについて、

$$NR_{k,i,j} = 0 \quad (14)$$

とする。

互いのマンハッタン距離が D_{max} 以内に存在しないスイッチ r_n, r_n のポート $p_{m,i}, p_{n,j}$ 全ての組み合わせについて、

$$RR_{m,i,n,j} = 0 \quad (15)$$

とする。

以上の追加制約によって、互いの距離が D_{max} 以下でない、スイッチのポート同士、及び、機能ブロックとスイッチのポートの接続が禁止される。

5. 提案手法の実装および評価結果

本稿で提案した、リンク長・レイテンシに制約を加えた NoC 生成手法を、実装し評価を行った。ILP ソルバとしては、lp_solve 5.5.0.14 [3] を用い、CPU として Intel Xeon X5470 3.33GHz を搭載するワークステーションの上で評価を行った。

評価実験での、NoC 内の機能ブロックを表 1 内に、機能ブロック間の通信関係を表 2 に示す。機能ブロック b を起点とし、a を終点とする、使用帯域は少ないものの、最大レイテンシが 1 という非常に小さい通信が存在する。

表 1 評価実験での NoC 内の機能ブロック

機能ブロック	幅	高さ
a	1	0.5
b	2	1
d	4	1
e	2	1
f	4	1

表 2 評価実験での NoC 内の機能ブロック間の通信関係

起点	終点	帯域	最大ホップ数
a	e	3000	6
d	e	3000	6
d	f	3000	6
b	d	3000	6
e	f	3000	6
b	a	10	1
b	e	3000	6
f	e	3000	6

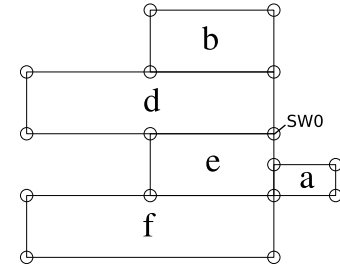


図 7 従来手法によって生成されたフロアプラン (図中の ○ は配置されたスイッチを表す)

5.1 フロアプランの生成

従来手法である、一ホップあたりの長さに制約を加えない場合と、提案手法である、一ホップあたりの長さに制約を加える場合について、それぞれフロアプランの生成を行った。一ホップあたりの長さに制約を加える際には、図内の機能ブロック b,d,e,f の高さに対応する 1 以下を指定した。フロアプランの生成の際の ILP ソルバの実行には、従来手法の場合に 4.75 秒、提案手法の場合に 4.68 秒の時間を要した。

従来手法および提案手法によって生成されたフロアプランを図 7、図 8 に示す。

通信レイテンシの要求が 1 と厳しい、機能ブロック b,a 間の距離について従来手法と提案手法を比較する。従来手法によって生成されたフロアプラン上では 1.5 となっているが、提案手法ではレイテンシ要求を反映して、0 となっていることが確認できる。今回の実験条件では、一ホップあたりの最大長に 1 を指定されており、従来手法で生成されたフロアプラン上では機能ブロック b,a 間の接続には二つのスイッチを介さずを得ず、レイテンシが制約に違反してしまうが、提案手法では一つのスイッチを介するのみで機能ブロック間を接続できる距離となっており、提案手法の有効性が示されている。

5.2 トポロジの生成

従来手法、提案手法で生成されたフロアプラン上に、それぞれ従来手法、提案手法でネットワークトポロジの生成を行った。各スイッチには 7 個のポートがあるものとした。また、実験の際には、10 式での入力ポート、出力ポートでの単位トラフィックあたりの消費電力である Ψ_i, Ψ_o として、それぞれ、328.66 を、11 式でのリンク上での単位トラフィック単位長さあたりの消費電力である Ψ_L として、80 を指定した。これらの値は既存手法の評価 [5] において用いられたものと同じものである。

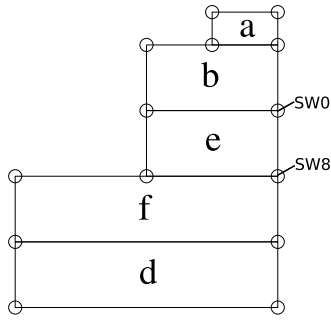


図8 提案手法によって生成されたフロアプラン
(図中の○は配置されたスイッチを表す)

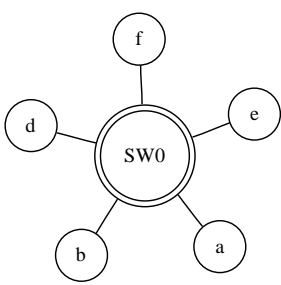


図9 従来手法によって生成されたトポロジ

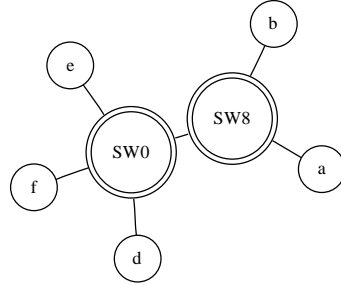


図10 提案手法によって生成されたトポロジ

トポロジ生成の際の ILP ソルバの実行を行った結果、従来手法の場合に 824.21 秒で最適解が得られたが、提案手法の場合には設定しておいた 1800 秒の最大実行時間を過ぎても最適解を得ることはできず、準最適解に基づくトポロジ生成を行った。

従来手法では全ての機能ブロックが一つのスイッチに接続されたトポロジが生成されたが(図9)、提案手法では機能ブロック a,b がスイッチ 0 に、機能ブロック d,e,f がスイッチ 8 に接続されたトポロジが生成された(図10)。また、従来手法、提案手法によってトポロジ上での通信経路はそれぞれ表3, 表4のように決定された。

一ホップのリンク長に制約を与えない既存手法では、全ての機能ブロックが同一スイッチに接続されるトポロジが生成された(図9)。これは、機能ブロック間の通信路上のホップ数が増加すると入力ポート、出力ポートでの消費電力が増加するためであると考えられる。一方、提案手法では、レイテンシ要求の厳しい通信を相互に行っている機能ブロック a,b がスイッチ 0 に、その他の機能ブロック d,e,f がスイッチ 8 に接続され、スイッチ 0-8 間が接続されたトポロジが生成された。提案手法では、通信路一ホップあたりの長さが与えられた制約を満たしつつ、機能ブロック間が与えられたレイテンシ要求以内で接続されることが確認された。

6. まとめと今後の課題

本稿では、混合整数線形計画問題によって、NoC 設計工程のうち、

- 各機能ブロックの配置を行うフロアプラン生成
- 各機能ブロック間の接続関係を定めるトポロジ生成

表3 従来手法によって生成されたトポロジ上での通信経路

起点	終点	経路
a	e	a→SW0→e
d	e	d→SW0→e
d	f	d→SW0→f
b	d	b→SW0→d
e	f	e→SW0→f
b	a	b→SW0→a
b	e	b→SW0→e
f	e	f→SW0→e

表4 提案手法によって生成されたトポロジ上での通信経路

起点	終点	経路
a	e	a→SW0→e
d	e	d→SW8→e
d	f	d→SW8→f
b	d	b→SW0→SW8→d
e	f	e→SW8→f
b	a	b→SW0→a
b	e	b→SW0→SW8→e
f	e	f→SW8→e

を行う既存手法を紹介した。その上で、既存手法において問題と考えられる各ノード間の距離の取扱いについて、

- フロアプラン生成に対する通信機能ブロック間距離の最大長に対する制約の追加

- トポロジ生成に対する機能ブロック・スイッチ及びスイッチ同士の距離に対する制約

の追加を提案・実装し、生成される NoC の各ノードの動作周波数に影響を与える通信路一ホップの長さの最大値が保証され、通信を行う機能ブロック同士がレイテンシ要求以下のホップ数で接続されることを確認した。

今後の課題としては、大規模な問題への対応が挙げられる。一ホップあたりの長さに制約を加えてトポロジ生成を行った場合、今回の実験を行った時間内には既存手法のみ、トポロジ生成を完了することができた。既存手法においても、ノード数が 10 程度に増大した際に実行時間が長くなり、NoC 生成を完了できないことが知られており、大規模な問題に対応するために設計を分割してフロアプラン・トポロジの生成を行うなどの手法を実装・評価する必要があると考えられる。

文 献

- [1] M. Dall'Oso, G. Biccari, L. Giovannini, D. Bertozzi, and L. Benini. xpipes: a Latency Insensitive Parameterized Network-on-chip Architecture For Multi-Processor SoCs. *Proceedings of the 21st International Conference on Computer Design*, pp. 536–539, 2003.
- [2] W.J. Dally and B. Towles. Route packets, not wires: on-chip interconnection networks. *Proceedings of the 38th Design Automation Conference (DAC)*, pp. 684–689, 2001.
- [3] lp.solve. <http://lpsolve.sourceforge.net/5.5/>.
- [4] P.P. Pande, C. Grecu, M. Jones, A. Ivanov, and R. Saleh. Performance Evaluation and Design Trade-Offs for Network-on-Chip Interconnect Architectures. *IEEE Transactions on Computers*, pp. 1025–1040, 2005.
- [5] Krishnan Srinivasan, Karam S. Chatha, and Goran Konjevod. Linear-programming-based techniques for synthesis of network-on-chip architectures. *IEEE Trans. Very Large Scale Integr. Syst.*, Vol. 14, No. 4, pp. 407–420, 2006.
- [6] S.R. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, A. Singh, T. Jacob, S. Jain, V. Erraguntla, C. Roberts, Y. Hoskote, N. Borkar, and S. Borkar. An 80-tile sub-100-w teraflops processor in 65-nm cmos. *IEEE Journal of Solid-State Circuits*, Vol. 43, No. 1, pp. 29–41, Jan. 2008.