

画像処理用 NoC におけるルーティングアルゴリズムの形式的検証 Formal verification of packet routing algorithm in NoC-based image processing circuit

伊藤 駿介[†] 近藤 真史[†] 横川 智教[‡]
Shunsuke ITO Masafumi KONDO Tomoyuki YOKOGAWA

1 まえがき

近年、演算コアの増加に伴って、コア間のデータ転送をパケット通信で行う Network on Chip (NoC) が注目されている [1]. この応用として、格子状に配置された演算コア群に画素値を対応付けた上で、近傍画素に対して螺旋状のパケットルーティングを展開することにより電氣的に画像処理を実現する手法を提案してきた [2]. しかし、回路シミュレーションを通じてその実現可能性は示されているものの、ルーティングにおけるパケットの競合やデッドロックの可能性等については検討されていなかった. そこで本研究では、形式的検証技術の一つであるモデル検査を用いて、パケットの競合が生じないことを示す. 具体的には、パケット伝搬の振る舞いをモデル化し、モデル検査ツール NuSMV により検証を行う.

2 NoC に基づいた汎用画像処理

NoC はインターネットの概念を集積回路内の演算コア間の通信に応用したものであり、一般的には格子状に配置した一対のルータとコアで構成され、コア間のデータ転送は経路上のルータを介したパケット通信により行われる. 一方、汎用画像処理は、注目画素周辺の近傍画素とフィルタカーネルとの積和演算によって実現される. 例えば、ノイズ除去等に用いられる平滑化処理をカーネルサイズ 3 (3×3 画素) の範囲に適用する場合、周囲 8 近傍画素を含む 9 画素に対し、1/9 をカーネル係数とした積和演算結果を逐次的に求めればよい.

文献 [2] では、NoC 上のコアに各画素を割当て、螺旋状のパケットルーティング過程で積和演算結果を求める手法を提案している (図 1). 具体的には、カーネル係数と累積画素値を内包したパケットをルータが螺旋状に伝搬し、ルータに内蔵された積和演算器により累積画素値を更新することで汎用画像処理を実現する. また、効率的な螺旋状ルーティングを実現するため、パケットの伝搬方向が右-下、左-上方向を単位として移動量が 1 ずつ増える点に着目し、現在の伝搬方向に対する規定移動量 r と現在の移動量 m をパケットに内包している. 各ルータは伝搬時に m をインクリメントし、これが r に達した場合は 0 で初期化するとともに伝搬方向を切り替える. さらに、右-下、左-上方向へ伝搬方向が切り替わる際には r をインクリメントして伝搬範囲を拡張する. これにより単純な比較とインクリメント回路のみでルーティングを実現できる. また、パケットは 1 ビットの返送フラグを内包しており、全ての近傍画素に対する積和演算が完了するとこれを設定し、以後は積和演算を伴うことなく返送経路を経て注目画素にパケットが返送される.

[†] 岡山理科大学 工学部 情報工学科

Department of Information and Computer Engineering,
Okayama University of Science

[‡] 岡山県立大学 情報工学科 情報システム工学科

Department of Systems Engineering,
Okayama Prefectural University

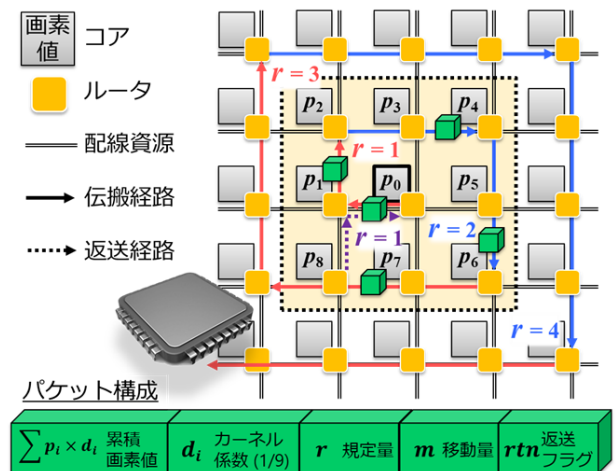


図 1 螺旋状ルーティングによる汎用画像処理

3 螺旋状ルーティングのモデル化

文献 [2] では上述の NoC を設計し、回路シミュレーションを通じて平滑化が行われていることを確認している. 上述の回路の設計にあたっては、各ルータへのパケットの入出力が一つに限定されることを前提としてルータ回路の簡略化・省面積化を図っている.

しかし、送出されたパケットがルーティング過程において競合する可能性については検討されておらず、内包される累積画素値の妥当性も保証されていない.

そこで本研究では、全ての状態の網羅的探索を行うモデル検査により、パケットの競合可能性について検証を行う. モデル化にあたっては、全てのパケットが規則的に伝搬する点に着目し、パケット伝搬の振る舞いをモジュールとして定義した上でそれらを格子状に配置し、NoC 全体のモデルを構築する方針を採る. パケット伝搬をモデル化したモジュール routing を図 2 に示す. モジュール内で定義された変数の役割と取り得る値は以下の通りである.

k_size : カーネルサイズ
mov : 移動量 m ($0 \sim k_size$)
req : 要求量 r ($0 \sim k_size$)
(x_init, y_init) : パケットの初期座標
(x, y) : パケットの現在座標
 $(x_init, y_init \pm (k_size-1)/2)$
dir : 伝搬方向 (上:U, 右:R, 下:D, 左:L)
rtn : 返送フラグ (TRUE, FALSE)

各パケットの座標を変数 x および y で管理し、ルータの動作に基づいてこれらの値を更新することでルーティングの振る舞いをモデル化する. 以下、各変数の遷移について述べる. なお、 i, j は NoC における行、列のインデックスを表し、実際のモデルではモジュールごとに定数として与えられる. また、紙面の都合上、各変数が定義域に収まるための遷移条件についての記述は省略している点に注意する.

```

1 MODULE routing{i*k_size+j}(x_init, x_init)
2 VAR
3   mov:0..{k_size};
4   req:0..{k_size}; dir:{U,R,D,L};
5   x  :{j-(k_size-1)/2}..{j+(k_size-1)/2};
6   y  :{-i-(k_size-1)/2}..{-i+(k_size-1)/2};
7   rtn:boolean;
8 ASSIGN
9   init(mov):=0; init(req):=1; init(dir):=L;
10  init(x):=x_init; init(y):=x_init;
11  init(rtn):=FALSE;
12  next(mov):=case
13    !next(rtnflag):case
14      mov=req:1; req>0:mov+1;
15    next(rtn)=TRUE:case
16      mov=req:1;
17      next(rtn)=TRUE&rtn=FALSE:1;
18      next(x)=x_init&next(y)=x_init:0;
19      req>0: mov+1;
20  next(req):=case
21    !next(rtnflag):case
22      dir=U&mov=req:req+1; dir=D&mov=req:req+1;
23    next(rtn)=TRUE:case
24      x=x_init&y=x_init:0;
25      TRUE:{k_size/2};
26  next(x):=case
27    next(req)>0:case
28      next(dir)=R:x+1; next(dir)=L:x-1;
29  next(y):=case
30    next(req)>0:case
31      next(dir)=U:y+1; next(dir)=D:y-1;
32  next(dir):=case
33    next(rtn)=TRUE&rtn=FALSE|mov=req:case
34      dir=U:R; dir=R:D; dir=D:L; dir=L:U;
35  next(rtn):=case
36    req={k_size}&mov={k_size-1}:TRUE;
37    rtn=TRUE:TRUE;
38    TRUE:FALSE;

```

図2 パケット伝搬をモデル化したモジュール

初期化:8-10行目 移動量 $mov:0$, 要求量 $req:1$, 伝搬方向 $dir:L$, 座標 x, y を初期座標 x_init, y に設定する。

mov:11-18行目 現在の方向に対して規定量を移動した場合, すなわち $mov=req$ の場合は次の伝搬に備えて $mov:1$, それ以外の場合は $mov+1$ でそれぞれ更新する。

req:19-24行目 上または下方向への伝搬途上において mov と同様に規定量を移動した場合, 螺旋状に伝搬するため次の伝搬方向への要求量を $req+1$ として更新する。

x:25-28行目 $req>1$ かつ伝搬方向が右の場合は $x+1$, 左の場合は $x-1$ でそれぞれ更新する。

y:29-32行目 $req>1$ かつ伝搬方向が上の場合は $y+1$, 下の場合は $y-1$ でそれぞれ更新する。

dir:33-35行目 $mov=req$ の場合, もしくは返送フラグ rtn が有効になるタイミングで, 現在の方向に対して時計回りの方向(上 → 右 → 下 → 左 → ...)に更新する。

rtn:33-35行目 $req=k_size$ かつ, $mov=k_size-1$ の場合, 返送フラグ rtn を有効にし, 以降これを維持する。
 rtn 有効後に各変数の振る舞いは変化し, req を $k_size/2$ で更新することにより初期座標への返送が行われ, 初期座標への到達を以って状態遷移を終了する。

4 検証結果

本研究では, 上記に基づくモデルを生成するツールを開発し, 得られたモデルを用いて検証を行った。ここで, 対象の NoC 規模が大きくなると状態数が指数関数的に増大し, 検証が困難となる。そこで, 注目画素とその近傍画素のルーティングに影響するモジュールのみからなる一辺が $(2*k_size-1)$ の NoC を対象に検証した。

```

1 LTLSPEC F (
2   (r4.x = r0.x & r4.y = r0.y) |
3   (r4.x = r1.x & r4.y = r1.y) |
4   ...
5   (r4.x = r8.x & r4.y = r8.y) )

```

```

-- specification F ((((((r4.loc_x = r0.loc_x & r4.loc_y = r0.loc_y) |
(r4.loc_x = r1.loc_x & r4.loc_y = r1.loc_y) | (r4.loc_x = r2.loc_x & r4.
loc_y = r2.loc_y) | (r4.loc_x = r3.loc_x & r4.loc_y = r3.loc_y) | (r4.l
oc_x = r5.loc_x & r4.loc_y = r5.loc_y) | (r4.loc_x = r6.loc_x & r4.loc_y
= r6.loc_y) | (r4.loc_x = r7.loc_x & r4.loc_y = r7.loc_y) | (r4.loc_x
= r8.loc_x & r4.loc_y = r8.loc_y)) is false

```

図3 パケットの競合に関する検査 ($k_size:3$)

```

1 LTLSPEC F (
2   (rD1.x = r0.x & rD1.y = r0.y) |
3   (rD1.x = r1.x & rD1.y = r1.y) |
4   ...
5   (rD1.x = r8.x & rD1.y = r8.y) )

```

```

-- specification F ((((((rD1.loc_x = r0.loc_x & rD1.loc_y = r0.loc_y) | (rD1.loc_x
= r1.loc_x & rD1.loc_y = r1.loc_y) | (rD1.loc_x = r2.loc_x & rD1.loc_y = r2.loc_y)
| (rD1.loc_x = r3.loc_x & rD1.loc_y = r3.loc_y) | (rD1.loc_x = r5.loc_x & rD1.loc_y
= r5.loc_y) | (rD1.loc_x = r6.loc_x & rD1.loc_y = r6.loc_y) | (rD1.loc_x = r7.loc
x & rD1.loc_y = r7.loc_y) | (rD1.loc_x = r8.loc_x & rD1.loc_y = r8.loc_y)) is true

```

図4 パケットの遅延に関する検査 ($k_size:3, i:1$)

まず, 図1のNoCにおけるパケット競合の可能性を検証する。パケット競合を表す検査式とNuSMVによる検証結果を図3に示す。この検査式は「将来的に注目画素($r4$)のパケットが近傍画素($r0\sim r3, r5\sim r8$)のパケットと同一座標上に存在する」ことを表す。検証結果はFALSEとなり, 注目画素のパケットが近傍画素のパケットと同一座標上に決して存在しないことが示された。

次に, パケット遅延によるパケット競合の可能性を検証するための検査式と検証結果を図4に示す。伝搬の状態遷移が i ステップ遅れたパケットの振る舞いを $rD\{i\}$ として定義し「将来的にこのパケットと近傍画素のパケットが同一座標上に存在する」ことを検査式として表す。検証結果はTRUEとなり, 送りが遅れる全ての場合について, 複数のパケットが同時に同一座標上に存在し, パケット競合が発生することが示された。

以上より, 全てのパケットが遅延なく螺旋状ルーティングを展開する限りにおいて, パケット競合は発生せず, 適切に画像処理を実現できることが示された。したがって, NoCの設計においてはパケット伝搬に係るタイミング制約の十分な考慮が重要となる。

5 あとがき

本研究では, 螺旋状ルーティングにおけるパケット競合の発生可能性についてモデル検査を用いて検証した。検証の結果, 全てのパケットが遅延なく同一動作を行った場合にのみ競合が発生しないことが示された。今後の課題としては, 複数のパケットが同一ルータから伝搬される風車状ルーティング [3] 等への応用が挙げられる。

謝辞 本研究の一部は, JSPS 科研費 25K15048 の支援を受けて実施されたものである。

参考文献

- [1] I.J. Barge, et al., “H.264 video decoder implemented on FPGAs using 3×3 and 2×2 networks-on-chip,” Proc. ReConfigurable Computing and FPGAs, pp.1–6 (Dec. 2017)
- [2] 三好 隼人, 他, “Network on Chip に基づく画像処理アクセラレータの基礎検討-リングバス型ルータ回路の設計-,” 2024年電子情報通信学会総合大会, ISS-SP-006 (Mar. 2024)
- [3] 吉原 蓮人, 他, “汎用画像処理用 NoC を対象とした風車状ルーティング回路の設計,” 情報処理学会第 87 回全国大会, 1J-05 (Mar. 2025)