

グローバルクロック同期型シストリックアレイ並列モデルに対する モデル検査器を用いた多様な振る舞い解析

Various Behavioral Analyses Using A Model Checker for Global Clock-Synchronized Systolic Array Parallel Models

千葉 悠矢 † 和崎 克己 ††
Yuya Chiba Katsumi Wasaki

1 はじめに

ハードウェア設計においてハードウェア回路検証は実際のハードウェア製造のコストが大きいため、開発コスト削減の面で重要な要素である。ハードウェア回路をモデル化し、モデル検査を行う形式手法も動作検証の1つとして用いられる。LOTOS 言語は主に2つのパートから構成される。1つは R.Milner による Calculus of Communicating Systems(CCS) と Hoare による Communicating Sequential Processes(CSP) に基づくプロセス代数であり、もうひとつは抽象データ型言語 (ACT ONE) に基づくデータ代数である。前者によりモデルシステムの状態遷移を表現し、後者によりプロセスがやりとりするデータ構造や値式を扱う [1]。本研究では LOTOS 言語で記述した1次元 W1 型シストリックアレイモデルを CADP toolbox[2] を用いて、振る舞いを検証する。シストリックアレイモデルでは、グローバルクロックによる同期型により構成する。同期型構成によりシストリックアレイの抽象度を下げ、それぞれの素子の振る舞いを観測可能とすることで、局所的なメッセージ非同期性とグローバル同期性の両方を検査する。

2 シストリックアレイにおける並列計算モデル

シストリックアレイは大量の単純な処理を行う Process Element(PE) を1次元または2次元上配置し、1度の入力により、生成される計算途中の一時データをメモリに書き込むことなく、演算を行うよう構成されたアーキテクチャである。加算や乗算などの単純な処理を行う PE を複数個用意し、PE の入力と出力を PE 同士で接続し、規則的に配置することにより、システム全体として行列積計算などの演算が可能である。それぞれの PE の入出力は全体で同期しながら処理を行う必要がある。1次元シストリックアレイは加算器、乗算器及びレジスタから構成された PE(セル) を1次元上に配置し、システム全体としてコンボリューション積を得るアーキテクチャである。コンボリューション演算は次のように定義される。重みの数列 w_1, w_2, \dots, w_k と入力列 x_1, x_2, \dots, x_n に対し $y_i = \sum_{j=1}^k w_j x_{i+j-1}$ となる出力列 $y_1, y_2, \dots, y_{n+1-k}$ を得る演算。システム全体としてコンボリューション積を得る1次元シストリックアレイが提案されている [3]。

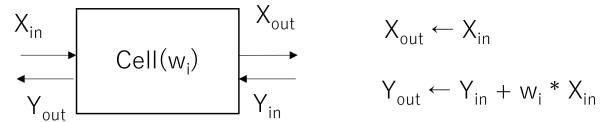


図1 W1型におけるセル単体の仕様

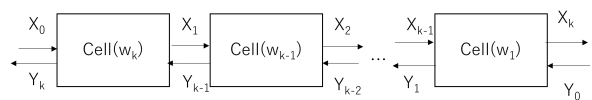


図2 W1シストリックアレイの構成図

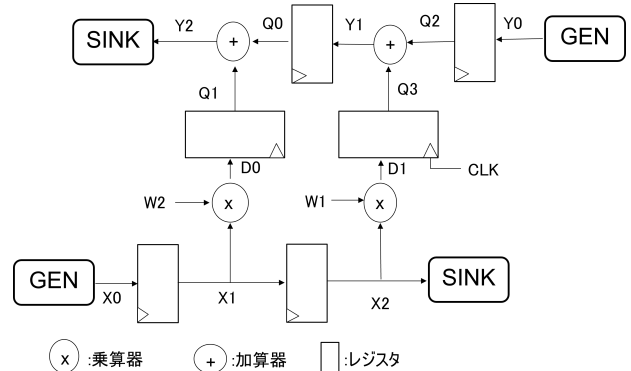


図3 W1型シストリックアレイの同期型構成図

本研究ではコンボリューション演算を行うシストリックアレイのアーキテクチャの1つである W1 型シストリックアレイを対象とする (図1,2)。グローバルクロック同期型1次元 W1 型シストリックアレイの構成要素として加算器、乗算器、レジスタがある [3]。これらをプロセスとして LOTOS 上で定義し、それぞれのプロセスの組み合わせによってセル単体を記述する。図3に2つのセルでのシストリックアレイの構成を示す。加算器1つ、乗算器1つ、レジスタ3つで1つのセルを構成し、複数のセルを用意して互いに接続することでシストリックアレイモデルを作成する (図4)。またグローバルクロックと入力を制御するプロセス GENERATOR(GEN) を定義し、適切なタイミングで入力とクロック制御を行う。

3 状態遷移系の生成

LOTOS 言語によって記述された W1 型シストリックアレイに対して LOTOS モデル検査器である CADP toolbox[2] によりモデルの振る舞いを表すラベル付き

† 信州大学大学院総合理工学研究科, Graduate School of Science and Technology, Shinshu University

†† 信州大学工学部電子情報システム工学科, Department of Electrical and Computer Engineering, Faculty of Engineering, Shinshu University

```

....
process Combine[X2,X1,X0,Y2,Y1,Y0,
                Q3,Q2,Q1,Q0,D1,D0,CLK] : noexit :=
  Gen[X0,Y0,CLK]
  |[X0,Y0,CLK]|
  (
    Sink[Y2]
    |[Y2]|
    Cell[X0,X1,Y1,Y2,CLK,Q0,Q1,D0](W2)
    |[CLK,X1,Y1]|
    Cell[X1,X2,Y0,Y1,CLK,Q2,Q3,D1](W1)
    |[X2]|
    Sink[X2]
  )
endproc(*Combine*)
....

```

図4 W1型のLOTOS記述(一部抜粋)

表1 LTSの状態数及び状態遷移数

種類	状態数	状態遷移数
モデルA(2cell)	32,279	94,326
モデルB(3cell)	15,035,761	60,412,237

状態遷移系(LTS)を生成する。対象モデルとして、2つのセル(PE)を接続したモデルAと、3つのセルを接続したモデルBを構築した。モデルAでは入力列 $\langle x_1, x_2, x_3 \rangle$, 重み列 $\langle w_1, w_2 \rangle$ を与え、コンボリユーション演算結果の列 $\langle y_1, y_2 \rangle = \langle w_1 * x_1 + w_2 * x_2, w_1 * x_2 + w_2 * x_3 \rangle$ を得る。プロセスGENERATORに入力列を記述し、2つのセルに重み w_1, w_2 を記述して割り当てる。モデルBでは入力列 $\langle x_1, x_2, x_3, x_4, x_5 \rangle$, 重み列 $\langle w_1, w_2, w_3 \rangle$ を与え、同様にコンボリユーション演算結果の列 $\langle y_1, y_2, y_3 \rangle$ を得る。プロセスGENERATORに入力列を記述し、3つのセルに重み w_1, w_2, w_3 を記述して割り当てる。表1にそれぞれのモデルから生成したLTSの状態数及び状態遷移数を示す。表2に各グローバルクロックCLKの立ち上がり回数別のLTSの状態数および状態遷移数を示す。

4 振る舞い解析

4.1 優先度を用いた検証

前節で説明した2つのモデルは表1に示すように、シストリックアレイモデルが示す振る舞いはセル(PE)の段数により大幅に増える。多様な振る舞いの要因の1つとして、加算器や乗算器などの遅延の存在がある。各素子から出力される信号に対し、優先度を付加することにより内部信号の到着順序の非決定性を抑える。各素子における入出力がモデルの状態遷移のラベルとなるため、各素子の入出力すべてに対して優先度の順序を定める。加算器及び乗算器の出力タイミング優先度の設定として、グローバルクロックの信号が、そのほかの信号と比較して最も優先度が低くなるように割り当てる。CADP toolbox[2]を用いて優先度を適用したモデルにおける、演算結果の出力Y2の振る舞いは仕様に求められる望ましい出力順序と一致する。

表2 モデルBにおける状態数及びトランジション数

CLK	状態数	トランジション数
1	3,211	11,489
2	39,850	149,064
3	186,697	726,901
4	626,475	2,437,588
5	1,736,045	6,844,717
6	3,444,537	13,701,601
7	5,907,817	23,592,417
8	8,630,453	34,614,403
9	11,634,885	44,621,019
10	15,035,761	60,412,237

```

<until> "X0 !'X_i'"
<until> "CLK !1"
<until> "CLK !0"
<until> "CLK !1"
<until> "X0 !'X_{i+1}'"
<until> "CLK !1"
<until> "CLK !0"
<until> "CLK !1"
<until> "X0 !'X_{i+2}'"
<until> "CLK !1"
<until> "CLK !0"
<until> "CLK !1"
<until> "Y3 !'X_i*W1+X_{i+1}*W2+X_{i+2}*W3'"

```

図5 シーケンス記述

4.2 パターン検査

入力に対して出力の値が入力された値を用いた結果となるかを検証する。3つのセルのシストリックアレイモデル(モデルB)において入力列 x_i, x_{i+1}, x_{i+2} を入力した後に最初の入力 x_i から6回のグローバルクロックの立ち上がりで計算結果である $x_i * w_1 + x_{i+1} * w_2 + x_{i+2} * w_3$ が出力される振る舞いがモデル上に存在するかを検証する($i = 1, 2, 3$)。検証では望ましい振る舞いのシーケンスを記述(図5)することにより、CADP toolboxを用いて探索[2]し、記述した状態遷移順を満たす振る舞いの1つを探索し、出力する。検証の結果それぞれの検証事項に対して記述した振る舞いを満たすモデル上のシーケンスは存在した。

5 まとめと今後の課題

本研究では1次元W1型シストリックアレイにおける振る舞い解析を行った。今後の課題としてシストリックアレイの繰り返し構造に着目した検証手法の確立がある。1次元シストリックアレイはセルを1次元上に並べた構造を持つため、セルに対する段階的な検証が必要である。

参考文献

- [1] Tommaso Bolognesi, Ed Brinksma "Introduction to the ISO Specification Language LOTOS" Computer Networks and ISDN Systems, vol. 14(1), pp. 25-59 January 1987, 66 pages
- [2] CADP(Construction and Analysis of Distributed Processes) INRIA/CONVECS, France, <http://cadp.inria.fr/>
- [3] Hsiang-Tsung Kung "Why systolic architectures?" Computer, vol.25, pp.37-46, January, 1982.