

部分定数乗算を考慮した任意桁数乗算回路の自動生成

世古 忠 ・ 黒岩 潤平*

Automatic Generation of Any Bits Multipliers Considering Partial Constant Multiplication

Tadashi Seko and Jyunpei Kuroiwa*

This paper presents a new method to generate any bits multipliers automatically considering partial constant multiplication. When some constraint conditions such as the number of bits, power consumption, area and delay are given, we will generate the optimal multipliers automatically. The system will generate VHDL description for the optimal multipliers considering partial constant multiplications and then the circuits will be synthesized using CAD tools. The experimental evaluations for the generated multipliers using Synopsys's Design Vision provided by VDEC show that the proposed method generates the optimal multipliers.

1. はじめに

近年、携帯用機器の機能が高度化し、画像や音声などのデジタル信号処理の必要性が高まってきた。これらのデジタル信号処理では乗算回路が頻繁に用いられており、応用目的に対応した様々な入力桁数の乗算器が求められている。乗算のVLSI アルゴリズム [1, 2, 3] は数多く提案されており、代表的なものとしては Array 型乗算器や Wallace Tree 型乗算器がある。入力桁数と消費電力、面積、遅延といった制約条件によって、最適な乗算器は異なってくる。しかし、入力桁数が変わるたびに最適な乗算器を探し、乗算器を構成するのは、特に入力桁数が増えると大変な労力となる。

そこで本研究では任意の入力桁数に対して、最適な乗算器を自動的に作成する乗算回路の自動構成システムを作成することを目的とする。自動構成する乗算器は Array 型と Wallace Tree 型とする。自動構成システムそのものは C 言語で作成し、システムが生成する乗算回路はハードウェア記述言語の一種である VHDL [4, 5] で記述する。開発するシステムでは、入力として乗算回路の入力桁数と制約条件を与えることで、乗算回路の構成に必要な全ての VHDL ソースプログラムを生成する。

また、デジタル信号処理の乗算ではしばしば定数乗算が行われる。定数乗算を行う場合は部分積生成や部分積加算で無駄な部分が生じ、そこを削減することで低消費電力化を行うことができる。よって乗数の特定の桁を

定数化する部分定数乗算器の自動構成も行う。

乗算回路の評価は東大 VDEC 提供の Synopsys 社 Design Vision [6, 7] を用いる。自動構成システムで生成した乗算回路の消費電力、面積、遅延を Design Vision により事前に測定し、制約条件に対する最適な乗算回路を決定する。

2. 乗算アルゴリズム

2.1 Array 型乗算器

Array 型乗算器は比較的簡単なアルゴリズムであり、レイアウトが規則的である。図 1 に 4 ビット Array 型乗算器の部分積加算部の構成を示す。図 1 では、被乗数を $X = (x_3 \cdots x_0)$ 、乗数を $Y = (y_3 \cdots y_0)$ 、積を $F = (f_7 \cdots f_0)$ としている。図 1 に示すように、Array 型は全加算器 (FA) を用いて部分積 $x_i \cdot y_j$ ($i, j=0, \dots, 3$) の加算を行う。

2.2 Wallace Tree 型乗算器

Wallace Tree 型乗算器の部分積加算部を図 2 に示す。Wallace Tree 型では、図 2 のように桁上げ保存加算器 (CSA) を木状に構成する。CSA により部分積 $p_i = X \cdot y_i$ ($i = 0, 1, \dots, n-1$) を 3 つずつ加算し、最後に最終加算回路で加算する。縦方向に部分積を加算することで加算段数が減り、Array 型乗算器に比べて高速な乗算が可能となる。

* 現在、奈良先端科学技術大学院大学学生

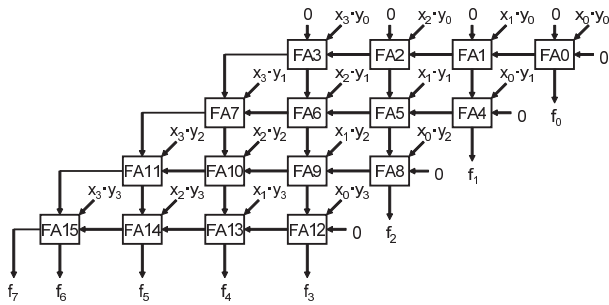


図1：Array 型乗算回路

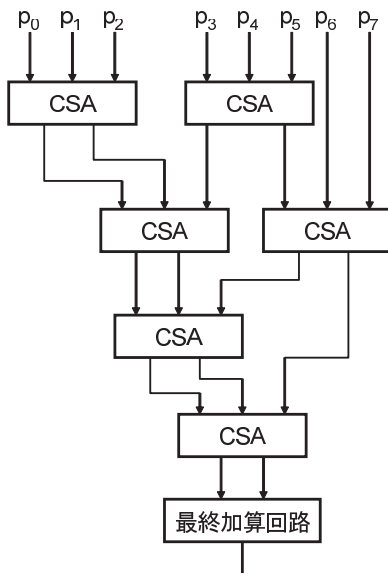


図2：Wallace Tree 型乗算回路

2.3 部分定数乗算器

デジタル信号処理では複数の定数と被乗数の乗算を行う場合がある。ここで定数の共通する桁のみを固定する部分定数化について考える。例えば、2つの定数 $Y_1 = (y_7 y_6 y_5 y_4 y_3 y_2 y_1 y_0) = 010111001$, $Y_2 = 10010101$ では、 y_0, y_4 が共通して1であり、 y_1, y_5 が共通して0である。今、被乗数 $X = (x_7 x_6 x_5 x_4 x_3 x_2 x_1)$ と定数 Y_1, Y_2 の乗算を行う場合、共通する1を固定したした場合の部分積の数は64個であるのに対して、共通する0を固定した場合は、48個となり大幅に部分積の数を削減することができる。このことにより、回路の一部を削減し、低消費電力化や高速化を行うことができる。本稿では、Array 型及び Wallace Tree 型の乗算回路に、乗数の一部を定数化する部分定数乗算回路を適用するシステムを開発する。

3. 自動構成アルゴリズム

3.1 システム概要

図3に開発する乗算回路の自動構成システムの概要を示す。図3のうち、システムの入力と出力について

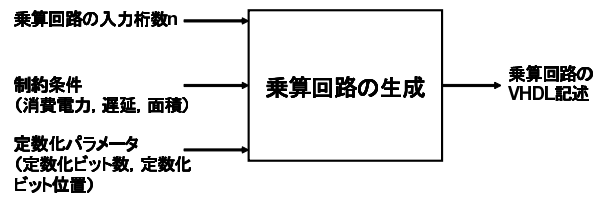


図3：システム概要図

詳細を以下に示す。

入力

- 乗算回路の入力桁数 n : 生成する乗算回路の入力桁数
- 制約条件 : 面積, 消費電力, 遅延のうち, 最も小さくしたい値を選択
- 定数化パラメータ : 乗数の n ビットのうち, 定数化するビットの数と, 定数化するビットの位置を指定

出力

- 乗算回路の VHDL 記述 : 制約条件を満たす乗算回路と, 必要な回路 (FA や RCA, CLA など) の VHDL 記述

制約条件

提案する自動構成システムの利点として、制約条件を満たす最適な乗算器を生成する点が挙げられる。制約条件を満たす最適な乗算器とは、面積, 消費電力, 遅延のうち, 指定した値が最も小さくなる乗算器である。まず、あらかじめ自動構成システムを用いて全乗算器のあらゆる入力桁数に対する消費電力, 面積, 遅延を測定する。それにより入力桁数と定数化パラメータに対して、消費電力, 面積, 遅延の値が最も小さくなる乗算器をそれぞれ調べ、制約条件を満たす乗算器を選択する。まず、自動構成部のみを作成し、各回路のパラメータを測定したのち、最適な乗算器を選択する部分を作成する。

3.2 自動生成手順

図4に、任意桁数乗算回路 (Wallace Tree 型乗算回路) の自動構成手順を示す。図4に示すように、本システムではまず入力桁数 (以下 n) と制約条件, 乗数の定数化に必要なパラメータを指定する。そのパラメータに基づき, 生成する乗算回路のアルゴリズムを決定する。その後, 各アルゴリズムに従って回路の自動生成を行い VHDL 記述を出力する。図4の場合, 各列に対して部分積の数の判定を行う。部分積の数が3個以上の場合, CSA を用いて部分積の加算を行い, 再び部分積の数の判定を行う。これをすべての列に対して適用して, 最後に最終加算回路で各列2個の部分積を加算することで, 任意桁数の乗算を行う。

定数乗算回路を生成する場合は、部分積の数が異なってくる。図4を定数乗算に対応させると、部分積の数を

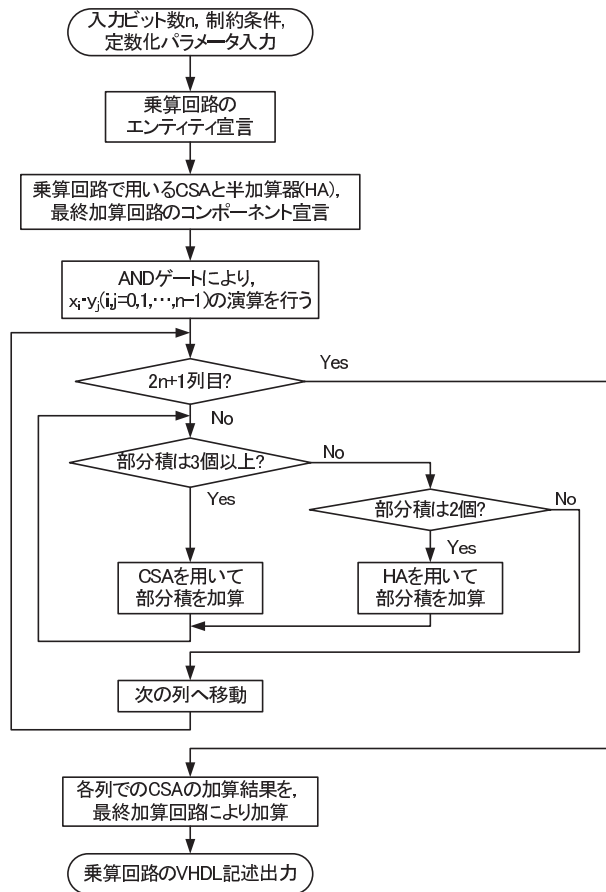


図4：任意桁数乗算回路の自動生成手順

判定する部分に変更が必要となる。定数化の桁に応じた判定を行うことにより、定数乗算回路の自動生成を実現することができる。

4. 実験結果

C言語を用いて任意桁数のArray型及びWallace Tree型乗算回路のVHDL記述を自動構成するシステムを開発した。さらに部分定数乗算回路の自動構成も行うようにした。本システムを用いて生成した乗算回路は、東大VDEC提供のSynopsys社のVHDLシミュレータVSSで論理検証を行った。その結果、任意の桁数に対して正しい乗算回路が生成されていることを確認した。またDesign Visionを用いて論理合成した。

表1に生成した64ビット×64ビット乗算回路の面積、消費電力および遅延を示す。表1で、Wallace-RCA

表1：生成乗算回路の例

乗算アルゴリズム	面積 (ゲート数)	消費電力 (mW)	遅延 (ns)
Array型	39524	62.9	424.2
Wallace-RCA	36662	56.9	189.4
Wallace-CLA	36688	57.1	178.5

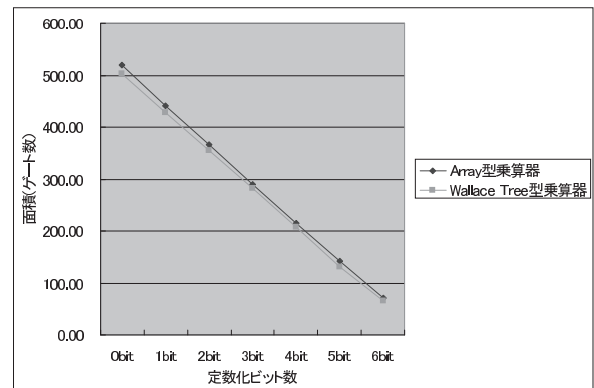


図5：定数化ビット数と面積

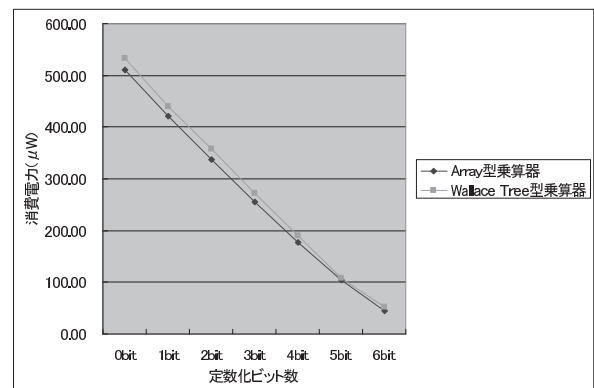


図6：定数化ビット数と消費電力

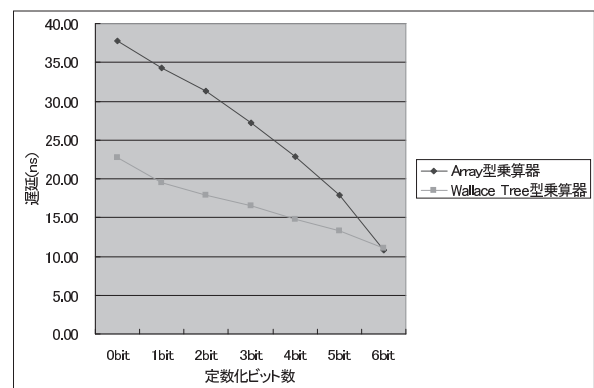


図7：定数化ビット数と遅延

は最終加算回路がRCA、Wallace-CLAはCLAである乗算回路を表す。表1より、面積と消費電力に関してはRCAを用いたWallace Tree型が、遅延に関してはCLAを用いたWallace Tree型が有効であった。

次に図5、図6、図7に8bits×8bits Array型及びWallace Tree型乗算回路の定数化ビット数に対する面積、消費電力、遅延の変化を示す。定数化ビット数0の場合は通常の乗算器を意味する。図より定数化ビット数が増加するとArray型及びWallace Tree型乗算回路ともに面積、消費電力、遅延時間が直線的に減少し、部分

定数化は回路の最適化に対して大きな効果があることが分った。また、図7より遅延時間はArray型乗算回路の方がWallace Tree型乗算回路に比べてより急速な効果が得られることが分った。

5. まとめ

本論文では任意桁数乗算回路の自動生成手法を提案し、C言語を用いて最適な乗算回路のVHDL記述を自動生成するシステムを開発した。また、開発したシステムでは部分定数乗算器の生成に対しても適用できるようにした。生成された回路を論理合成し、回路の面積、消費電力、遅延時間を評価した結果、制約条件を満足する最適な回路が合成されていることを確認した。今後の課題としては他の乗算アルゴリズム、例えばBooth型や基数が2以外の乗算アルゴリズムを自動構成システムに組み込むことなどが挙げられる。

参考文献

- [1] Israel Koren, *Computer Arithmetic Algorithms: Second edition*, A K Peters, Ltd., 2002.
- [2] 高木直史, “算術演算のVLSIアルゴリズム”, コロナ社, 2005.
- [3] THOMAS H. CORMEN, CHARLES E. LEISERSON, RONALD L. RIVEST 共著, 浅野哲夫, 岩野和生, 梅尾博司, 山下雅史, 和田幸一共訳, アルゴリズムイントロダクション第3巻, pp.28-43, 近代科学社, 1990.
- [4] K.C. Chang, *Digital Systems Design with VHDL and Synthesis: An Integrated Approach*, IEEE Computer Society Press, 1999.
- [5] 長谷川裕恭, VHDLによるハードウェア設計入門, CQ出版株式会社, 1995.
- [6] Synopsys, *Design Compiler Reference Manual: Constrains and Timings*, Version U-2003.03, March 2003.
- [7] Synopsys, *Design Compiler Reference Manual: Optimization and Timing Analysis*, Version U-2003.03, March 2003.