

木構造部分積加算回路をもつ乗算器の高速化に関する研究

水口 貴之[†] 味元 伸太郎[†] 橘 昌良[†]

[†] 高知工科大学大学院工学研究科電子・光システム工学コース
〒782-8502 高知県香美市土佐山田町宮の口 185

E-mail: [†]{095318n,095319w}@gs.kochi-tech.ac.jp, [†]tachibana.masayoshi@kochi-tech.ac.jp

あらまし 乗算器においては、部分積加算回路と最終段加算回路の性能が乗算器全体の性能を大きく左右する。本研究では、高速部分積加算回路として代表的な木構造部分積加算回路である Wallace tree 型部分積加算回路と Dadda tree 型部分積加算回路の性能をシミュレーションにより比較・検討する。そして、シミュレーション結果に最適な最終段加算回路の構成について検討する。

キーワード 部分積, Wallace tree, Dadda tree, 最終段加算回路

Design of High Speed Multiplier with Tree-structured partial product adders

Takayuki MINAKUCHI[†] Shintaro MIMOTO[†] Masayoshi TACHIBANA[†]

[†] Electronic and Photonic Systems Engineering Course, Kochi University of Technology
185 Miyanakuchi, Tosayamada-cho, Kochi 782-8502 Japan

Abstract In multipliers, performance of a partial product addition circuit and a last step addition circuit controls performance of the whole multiplier. In this research, we compare and examine performance of a Wallace tree partial product addition circuit and the Dadda tree partial product additional circuit. And we examine construction of the last stage additional circuit which is suitable for a simulation result.

Keyword Partial product, Wallace tree, Dadda tree, Last stage additional circuit

1. はじめに

乗算器の性能を決定する重要な要素として、部分積加算回路と最終段加算回路が挙げられる。演算時間を短縮する部分積加算回路として代表的なものに、Wallace tree 型部分積加算回路と、Dadda tree 型部分積加算回路がある。この2つの部分積加算回路を設計、論理合成し、シミュレーションを行うことで、LSB から MSB までの演算到達時間を測定する。

Wallace 型、Dadda 型共に Tree 構造となっているため、演算到達時間は bit 毎に異なる。中間位ビットは演算到達時間が遅くなり、LSB・MSB に近くなるに従い早くなる傾向にある。シミュレーション結果を踏まえ、Wallace 型、Dadda 型両形式の特性を確認する。その結果を踏まえ、最終段加算回路の構成を検討する。

2. 部分積加算回路

図 1.1 は、8 ビット乗算器における部分積である。部分積を表す○の中の数字は、通過したアダーのステージ数である。

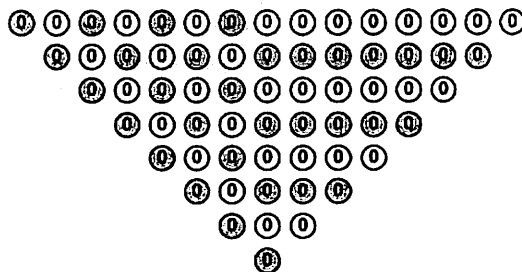


図 1.1 8 ビット乗算器における部分積

Wallace tree 型と Dadda tree 型の 2 形式のアルゴリズムを適用した部分積を、図 1.2、図 1.3 に示す。



図 1.2 Wallace Tree 型部分積加算アルゴリズムを用いた 8 ビット乗算回路の部分積



図 1.3 Dadda Tree 型部分積加算アルゴリズムを用いた 8 ビット乗算回路の部分積

以上のように、Wallace Tree においてはアダーの最多ステージ数が 5 段に、Dadda Tree においては 4 段に短縮される。Wallace 型においても Dadda 型においても、10 ビット目前後の演算到達時間が最長となると予想される。

部分積加算回路において演算時間が最長となるパスは、乗算器全体の演算速度に直結してくる。最終段加算回路において、最長パスより上位のビットは、最長パスの値が決定しないと演算が無駄になるためである。最長パスよりも上位のビットは、最長パスの値の決定を待たなければならない。最長パスが MSB 側にあるほど、上位ビットは高速に確定する。また、下位ビットの演算が無駄にならず、消費電力は低減される。

3. シミュレーション

シミュレーションは、VDEC(大規模集積回路システム設計教育研究センター)より提供されている CAD ツールを用いて行う。シミュレーションの手順は以下のようなになる。

1. ハードウェア記述言語 VHDL を用い、Wallace Tree 型乗算器、Dadda Tree 型乗算器について、最終段加算回路を除いた状態で作成する。
2. Synopsys 社提供の Design Compiler を用いて論理合成を行う。その後、最終段加算回路への出力信号の Timing 値を得る。
3. Synopsys 社提供の Nanosim を用いて、消費電力シミュレーションを行う。

3.1. シミュレーションの条件

論理合成に用いるライブラリは京都大学版 Rohm0.35um ライブラリである。

Nanosim での消費電力シミュレーションに用いるテストベンチの入力パターンは、C 言語により生成した乱数を用い、ランダムに生成している。100 パターンの入力値を作成し、100ns 毎に 1 パターンを入力する。全体のシミュレーション時間は 10us となる。

Timing 値、消費電力値ともに配線容量は考慮されていない値である。

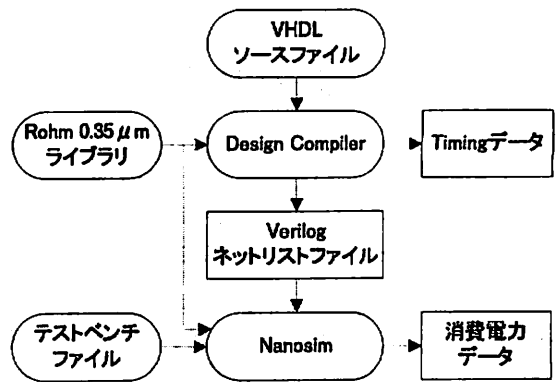


図 2 シミュレーションのフロー図

3.2. シミュレーション結果

Design Compiler を用いて算出した部分積加算回路の Timing 値をグラフで表す。Wallace 型の結果を図 3.1 に、Dadda 型の結果を図 3.2 に示す。ビット数は 8 から 24 ビットまでとしている。なお、グラフ中の a、b は、部分積加算回路から出力される信号を意味し、LSB から順番に値を割り振っている。

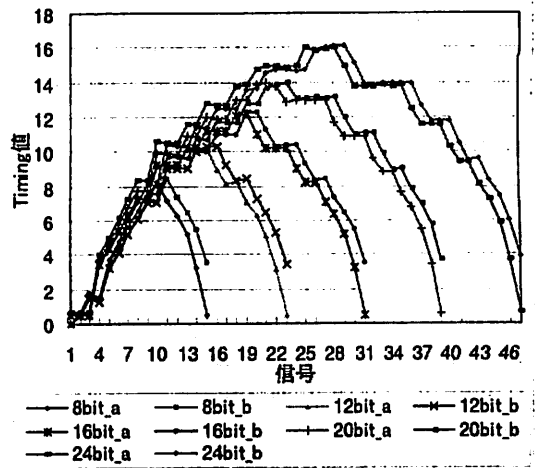


図 3.1 Wallace Tree 型部分積加算回路の Timing 値

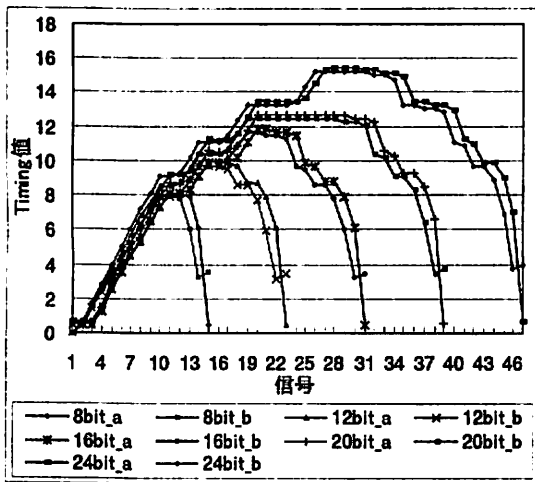


図 3.2 Dadda Tree 型部分積加算回路の Timing 値

ここで、Dadda 型と Wallace 型のシミュレーション結果について、24 ビット部分積加算回路のデータを用いて比較したものをグラフ 3.3 に示す。

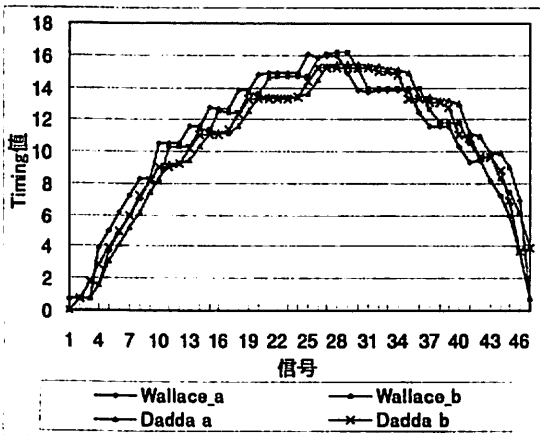


図 3.3 24bit 部分積加算回路における Wallace Tree 型と Dadda Tree 型の Timing 値比較

Dadda Tree 型は Wallace Tree 型と比較して、演算時間のピークとなる最長パスが短く、MSB 側に近くなることが分かる。また、演算時間のピークより下位の信号到達時間も全体的に短くなっている。

最長パスより下位のビットに用いる最終段加算回路は、最長パスの決定を待つために低速で低消費電力の加算器を用いることが出来る。このため、最長パスが MSB に近い位置にあるほど、電力特性に優れた加算器で構成する割合を増やすことが出来る。

今回の測定では、Dadda 型が Wallace 型よりも理想的な特性を持つという結果になった。

最長パスより上位のビットは予め部分積の値が確定するため、最終段加算回路に桁上げ先見方式を用いて高速化を図る等の方法が考えられる。桁上げ先見方式は順次桁上げ方式と比較して電力消費が大きくなる傾向があるが、最長パスが MSB に近いほど桁上げ先見加算器で構成する割合は少なくなる。結果、消費電力特性に優れた構成となる。

4. 最終段加算回路

最終段加算回路は、各ビットの演算到達時間を考慮する必要がある。仮に部分積加算回路の 10 ビット目がクリティカルパスとなる場合は、それより下位のビットは低消費電力かつ低速な加算器を用い、10 ビット目より上位のビットは桁上げ先見などで高速化を図る等の手法が考えられる。

以前の報告会において、加算器の消費電力・面積・速度についての研究を報告した。報告した加算器の形式は、

- Ripple carry adder (以下 RCA)
- Carry look-ahead adder (以下 CLA)
- Carry select adder (以下 CSLA)
- Carry skip adder (以下 CSA)
- Hybrid adder (以下 HBA)

の 5 形式である。

加算器の性能を、RCA の値を 1 として正規化したデータを表 1.1 から 1.3 に示す。

bit	RCA	CLA	CSLA	CSKA	HBA
8	1	0.94	0.65	1.30	0.64
12	1	0.71	0.53	1.27	0.66
16	1	0.61	0.41	1.25	0.57
20	1	0.55	0.39	1.24	0.45
24	1	0.44	0.37	1.24	0.38

表 1.1 正規化した加算器の Timing 値

bit	RCA	CLA	CSLA	CSKA	HBA
8	1	1.20	2.13	1.29	1.90
12	1	1.15	2.18	1.32	1.70
16	1	1.21	2.18	1.35	1.97
20	1	1.15	2.27	1.36	2.21
24	1	1.19	2.33	1.37	2.27

表 1.2 正規化した加算器の消費電力値

bit	RCA	CLA	CSLA	CSKA	HBA
8	1	1.87	2.05	1.39	2.26
12	1	1.82	2.25	1.41	2.19
16	1	1.85	2.45	1.43	2.29
20	1	1.90	2.49	1.44	2.27
24	1	1.90	2.51	1.45	2.71

表 1.3 正規化した加算器の面積値

消費電力のデータは、加算器にオペランドを全ビット一斉に入力した場合の値である。

乗算器における最終段加算回路は、入力を受け付けるタイミングがビット毎に異なる。オペランド一斉入力時の値を1として、LSBから順番に入力を受け付けた場合の消費電力の正規化データを、表 1.4 に示す。

入力パターンはC言語の乱数発生関数を用いて100パターン作成する。1nsに1ビットLSBから入力し、100ns毎に次のパターンを入力する。

全体のシミュレーション時間は10nsである。その他の条件は、以前報告した加算器のシミュレーションと同一である。

bit	RCA	CLA	CSLA	CSKA	HBA
8	0.65	0.73	0.66	0.69	0.70
12	0.56	0.60	0.55	0.54	0.55
16	0.45	0.48	0.47	0.46	0.48
20	0.42	0.46	0.44	0.42	
24	0.38	0.41	0.40	0.39	

表 1.4 正規化した加算器の消費電力(入力 LSB→MSB)

オペランドをLSBから順次入力した場合、一斉入力の場合と比べて消費電力は低減するという結果になった。ビット数が増加するにつれて、その傾向は強くなる。特に逐次桁上げ加算器において、その傾向が強くなっている。

5. 考察

最終段加算回路を、最長パスの位置に着目して決定する。図 3.3 において、最長パスの位置は14ビット目、27番目の信号となる。最終段加算回路は、27番目の信号の位置までは演算の速度を問われないため、最も低速であるが消費電力特性に優れるRCAにより、低速かつ低消費電力に計算することが可能であると考えられる。

28番目以上の信号は既に値が決定しているため、高速に演算を収束させるために桁上げ先見を用いる。28

番目から最上位の46番目の信号は、ビット数では10ビットに相当する。表 1.1、表 1.2 より、CLAを用いることで、消費電力を大きく増やすことなく高速化を図ることができる。

CLAはビット数が多くなるほどに高速化に優れた性能を発揮するため、大規模な乗算器になるほど、その性能は顕著に現れると考えられる。

6. 今後の展望

RCAを最長パスの信号まで用いることで、低消費電力化が狙えることは述べたが、そのためには最長パスの発生する位置を設計以前に知らなければならない。ツリー構造はビット数によって構成が変化するため、事前にパスの長さや位置を知る手法を考案する必要がある。

また、最終段加算回路への入力タイミングによっては、部分積加算回路の演算が最終段加算回路に追いつかず余計な遅延が発生することも考えられる。最長パスまでRCA構成することが一概に正しいとは言えない。

図 4.1 のように、RCAを最終段加算回路として最長パスまで構成した場合、最長パス以下の信号の到達時間が最終段加算回路に対して遅れる可能性がある。これは回路の再計算を促し、演算の収束を遅らせる。演算速度、消費電力特性共に悪影響を受ける可能性があり、好ましくない。

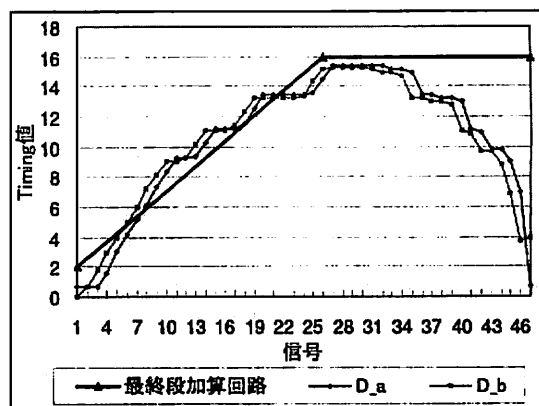


図 4.1 最終段加算回路と部分積加算回路のタイミング

これを改善する方法として、部分積加算回路を3段階に分割して考える。図 4.2 に例を示す。

LSBから任意段までをRCAで構成し、それ以降の上位ビットについては桁上げ先見加算器を二段階に分

割して用いる。分割構成とすることで、より部分積加算回路の性能に適合する最終段加算回路を用いることができ、最適化が容易になる。

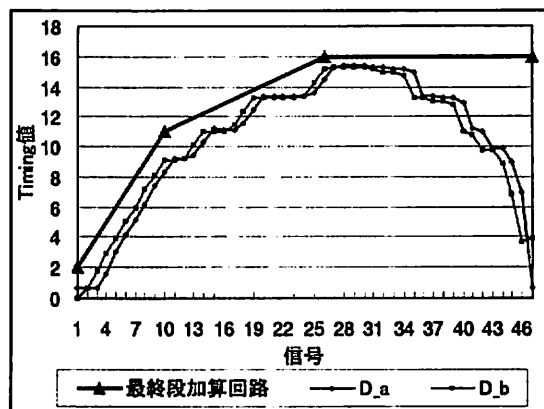


図 4.2 3 段階分割の最終段加算回路例

最終段加算回路の上位ビットでは、MSB から順番に信号が到達する。加算器は MSB の値を受け取っても、下位ビットの値が確定しない限り遷移が無駄になる。逐次桁上げ方式は遷移がダイレクトに上位ビットに反映するため、演算速度と共に考慮しても桁上げ先見形式が適すると考えられる。

MSB 順次到達の加算器の構成については、今後検討する。

本研究は、東京大学大規模集積システム設計教育研究センターを通し、シノプシス株式会社の協力で行われたものである。

文部科学省学術研究高度化推進プロジェクト
「柔軟でセキュアなフォトニックユビキタスネットワークの実現」の一部として行われたものである。

文 献

- [1] Israel Koren, "Computer Arithmetic Algorithms Second Edition", A K Peters, Ltd, Canada, 2001.
- [2] Behrooz Parhami, "Computer Arithmetic ALGORITHMS AND HARDWARE DESIGNS", OXFORD UNIVERSITY PRESS, New York, 2000.
- [3] K.C.Chang, "Digital Systems Design with VHDL and Synthesis", IEEE Computer Society Press, 1999.
- [4] Giovanni De Micheli, "Synthesis and Optimization of Digital circuits", Mcgraw-Hill College, Inc, U.S.A., 1994.