

部分定数化による乗算器の低消費電力化について

杉本 秀一[†] 遠藤 彬浩[†] 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 高知県香美郡土佐山田町宮ノ口
185

E-mail: † {085306f,085304t}@gs.kochi-tech.ac.jp, ‡ tachibana.masayoshi@kochi-tech.ac.jp

あらまし 本稿では、乗算器の低消費電力化の方法として乗数の一部分を固定する手法を検討する。本手法は、いくつかの乗数のみの乗算を行う場合に、常に0で遷移しない桁を固定することにより冗長である部分を削除し、低消費電力化を行っている。今回は部分定数乗算器のシミュレーションを行い、シミュレーション結果より部分定数化による乗算器の低消費電力化の有効性を確認した。

キーワード 部分定数化, 乗算器, 低消費電力化

Design and Simulation of low power consumption of multiplier by using the partial constant

Syuuichi Sugimoto[†] Akihiro Endou[†] and Masayoshi Tachibana[‡]

[†] ‡ Electronic and Photonic Systems Engineering Course, Kochi University of Technology,

185 Miyanakuchi, Tosayamada-chou, Kochi, 782-8502 Japan

Abstract In this paper, we consider a method of designing low power consumption multiplier by using partial constant. The main idea of the method is to remove redundant part of multiplier by fixing the digit that does not change with always 0. Here, we simulate to the partial constant multiplier. Simulation results show that the method can do low power consumption.

Keyword Partial constant, Multiplier, Low power consumption

1. はじめに

今日、集積回路はさまざまな製品、分野で使用され最近では、携帯電話やモバイル端末を利用して屋外においてもネットワークを通じて様々なデータ送受信し、処理することが可能となった。このようなユビキタスコンピューティングは更なる発展が期待できる分野であるが、端末の小型化とともに駆動可能時間（バッテリーライフ）が重要なファクターとなっており、駆動可能時間を延ばすためにはバッテリーの改善の他、端末の中核となって処理を行う集積回路の低消費電力化が重要となってくる。

一般的に、消費電力を低減する方法として

- (1)冗長である回路を停止、あるいは除去する
 - (2)パフォーマンス制約が許す限り低速化、低電圧化する
 - (3)面積と電力のトレードオフを考える
 - (4)入力信号を最適化し、内部の信号遷移確率を低下させる。
 - (5)演算処理などを最適化する
- などの手法があるが、(1)は回路の規模にかかわらず回路を動かさない、もしくは冗長な部分を除去するの

で十分に低消費電力化が期待できる。(2)と(3)は面積制約、リソース制約が存在しない、もしくは十分に余裕があるという事が前提条件となり、あらかじめマッピングできる演算器のライブラリが必須となる。(4)、(5)も、回路が小規模であった場合はスケジューリング余地が少なくなるので、ある程度回路規模が大きくない限り最適化を施しても十分な低消費電力化は望めない。

本稿では、乗算器の低消費電力化として乗数の一部分のみを固定する部分定数化の手法を検討する。また、検討した手法の有効性を評価するためにVHDLで乗算器を記述し、論理合成を行い消費電力をシミュレーションした。以下、2節では本稿で取り扱う部分定数化した乗算器について説明する。3節にてシミュレーション手順を示し、4節にてシミュレーション結果を示す。5節では、まとめと今後の課題について述べる。

2. 部分定数化

今回使用する乗算器はすべてWallaceTree型で構成されている。Array型で構成した方が固定位置が変わるたびに再構成する必要はないが、Array型の面積は $O(n^2)$ であるのに対しWallaceTree型は $O(n \log n)$ である

ため、演算器としてみた場合には WallaceTree 型の方が優位だからである。

2.1 部分定数化

信号処理などで使用する場合、乗数はいくつかの定数に限られる場合が多いが、完全定数化した乗算器ではその定数でしか演算できないため乗数が複数ある場合は完全定数化はできない。そこで、完全定数化するのではなく、0 が共通する桁を固定する部分定数化を考える。

乗数 1 11100101

乗数 2 11000111

上記のような 8 ビットの 2 つの乗数があった場合、4 ビット目と 5 ビット目がともに 0 であるために固定することができ、これにより 4 ビット目、5 ビット目の部分積生成回路、部分積加算回路を除去することができる。通常の乗算器の 4 ビット目、5 ビット目を 0 で固定することで信号遷移を減らすことで低消費電力化も望めるが、演算数は変わらないため固定した乗算器を形成した方が低消費電力化することができる。

2.2 部分定数化した乗算器

4 ビット目と 5 ビット目を固定することにより、部分積は 2 列分削減することができる(図 1(a), (b))。通常の乗算器と部分定数化した乗算器を構成する Full Adder(FA)と Half Adder(HA)の数を表 1 に示す。なお、最終加算を行う加算器の部分は含めていない。

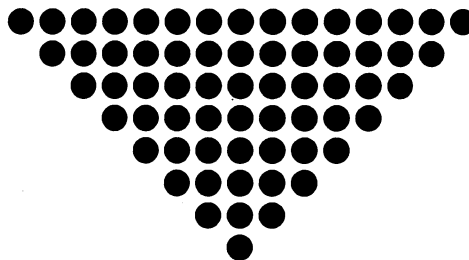
	HA	FA
通常	7	36
部分定数化	7	20

表 1 通常乗算器と部分定数化乗算器の HA,FA 数

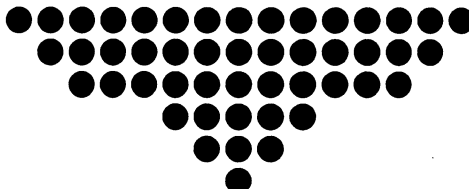
通常乗算器に比べて部分定数化した方が HA の数は変化はないが、FA は 16 個削減できている。Area と Timing の表を表 2 に示す。このとき、配線は考慮せず、Timing は論理合成時のクリティカルパスの遅延時間である。表 1、2 から、部分定数化した乗算器を構成した方が、Area、Timing とともに改善されており、削減できた FA16 個分の消費電力の削減が見込まれる。

	Area (μm^2)	Timing (ns)
通常	24748.919922	9.28
部分定数化	15611.399414	5.89

表 2 通常乗算器と部分定数化乗算器の Area, Timing



(a)



(b)

図 1 8 ビット乗算器における通常の乗算器の部分積(a)と、4,5 ビット目固定の乗算器の部分積(b)

3. シミュレーション

3.1 シミュレーション手順

消費電力のシミュレーション手順を以下に、フローを図 1 に示す。なお、使用したツールはすべて VDEC から提供される Synopsys 社のツールである。

- 1) 乗算器を VHDL で記述し、論理シミュレーションする。
- 2) Design Compiler で京大版 Rohm0.35 μm ライブラリを使用し、論理合成を行う。
- 3) VCS AD および、Nanosim を使用し、消費電力をシミュレーションする。

3.2 シミュレーションにおける条件

- テストベンチファイル

論理シミュレーション(Scirocco)に使用するテストベンチファイルは全パターンを入力を与え、消費電力のシミュレーション(Nanosim)の入力パターンは C 言語の擬似乱数を用いて乗数、非乗数ともに 100 パターンとしている。また、シミュレーション時間は全て $5\mu\text{s}$ とした。

- Area、Timing

Area、Timing とともに論理合成時におけるセル面積のみで配線は考慮していない。Area は論理合成を行ったときに算出されるセル面積の合計、Timing は論理合成時のクリティカルパスの遅延時間である。

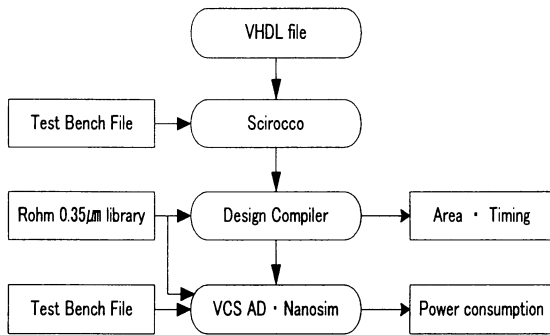


図 2 シミュレーションフロー

● 消費電力

消費電力のシミュレーションは Nanosim で行い、平均電力と RMS 電力をシミュレーションする。Area、Timing と同じく配線は考慮していない。

3.3 シミュレーション範囲

消費電力のシミュレーション範囲は、部分積生成回路、部分積加算回路である。一般的に Wallace Tree 型の乗算器を構成した場合、乗算器のビット数に合わせて最終加算器を形成した方が速度的にいい結果が得られるため、最終加算器はシミュレーション範囲に含めなかった。

4. シミュレーション結果

8 ビット、12 ビット、16 ビットの乗算器を用意し、それぞれ固定なし、1 ビット固定、2 ビット連続、非連続固定、3 ビット連続、非連続固定、4 ビット連続、非連続固定をシミュレーションした。ただし、8 ビット乗算器においては、4 ビット非連続固定はビット数の関係上行っていない。

4.1 8 ビット乗算器

8 ビット乗算器のシミュレーション結果をそれぞれ図 3.1、3.2、3.3 に示す。固定するビット数と削減できる量はほとんど比例関係であり、連続、非連続であってもほぼ同じ結果となっている。表 3 に連続固定時の固定ビット数と消費電力を示す。RMS Power では、1 ビット固定で約 15%、2 ビット固定で約 35%、3 ビット固定で約 50%、4 ビット固定で約 70% 低消費電力化できている。

4.2 12 ビット乗算器

12 ビット乗算器のシミュレーション結果をそれぞれ図 4.1、4.2、4.3 に示す。8 ビット乗算器の時と同じく連続、非連続固定であっても差はなく、ビット数と

固定ビット数	AVE Power (μW)	AVE 割合	RMS Power (μW)	RMS 割合
0	4363.04	100%	11349.47	100%
1	3547.57	81.31%	9481.89	83.54%
2	2599.31	59.58%	7492.46	66.02%
3	1806.63	41.41%	5644.79	49.72%
4	1065.11	24.41%	3611.34	31.82%

表 3 8 ビット乗算器の固定ビット数と消費電力

削減できる量は比例関係である。表 4 に連続固定時の固定ビット数と消費電力を示す。RMS Power では、1 ビット固定で約 10%、2 ビット固定で 20%、3 ビット固定で 30%、4 ビット固定で 40% 低消費電力化できている。

固定ビット数	AVE Power (μW)	AVE 割合	RMS Power (μW)	RMS 割合
0	6513.82	100%	19892.91	100%
1	5653.70	86.8%	17931.27	90.14%
2	4830.40	74.16%	15937.14	80.11%
3	4019.40	61.71%	13798.13	69.36%
4	3268.49	50.18%	11820.41	59.42%

表 4 12 ビット乗算器の固定ビット数と消費電力

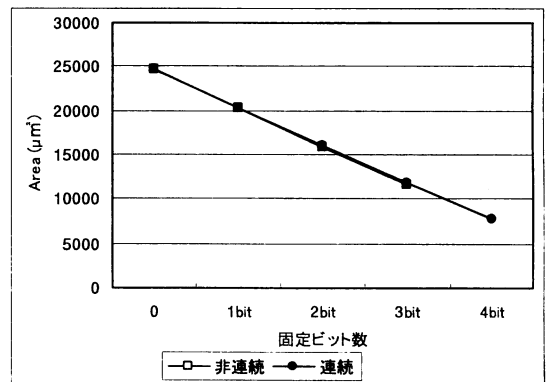


図 3.1 8 ビット乗算器の固定ビット数と Area

4.3 16 ビット乗算器

16 ビット乗算器のシミュレーション結果をそれぞれ図 5.1、5.2、5.3 に示す。8 ビット、12 ビットの時と同じく連続、非連続とほとんど差はない。表 5 に固定ビット数と消費電力を示す。RMS Power では、1 ビット固定で約 10%、2 ビット固定で約 15%、3 ビット固定で約 25%、4 ビット固定で約 30% 省電力化できている。

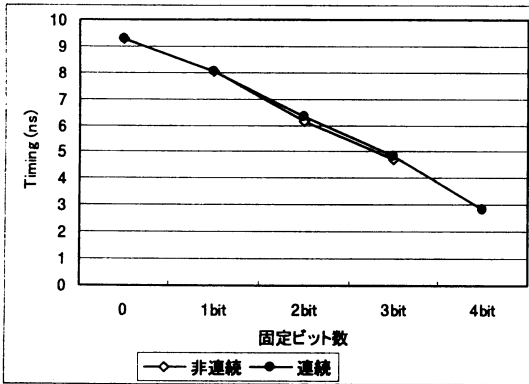


図 3.2 8 ビット乗算器の固定ビット数と Timing

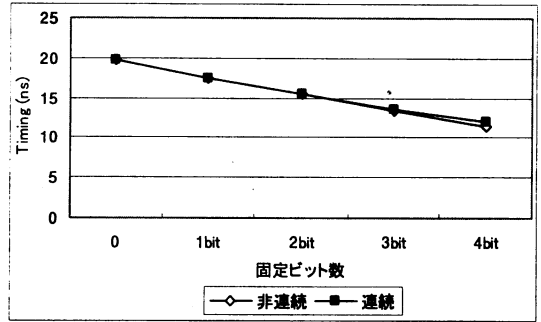


図 4.2 12 ビット乗算器の固定ビット数と Timing

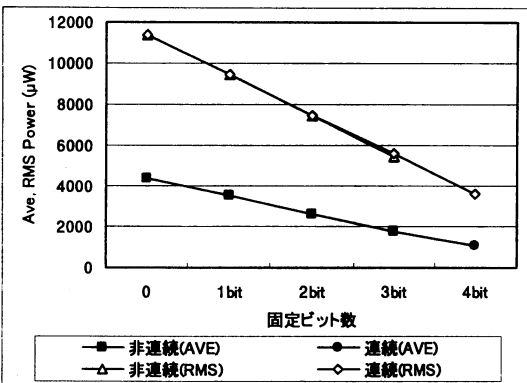


図 3.3 8 ビット乗算器の固定ビット数と Ave、RMS Power

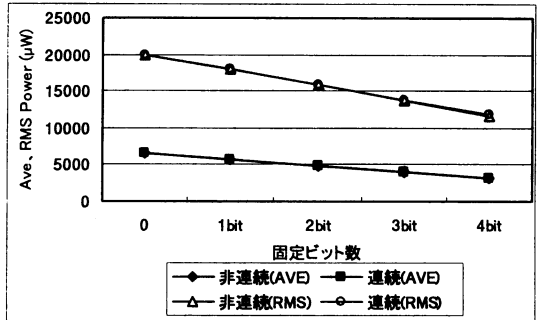


図 4.3 12 ビット乗算器の固定ビット数と Ave、RMS Power

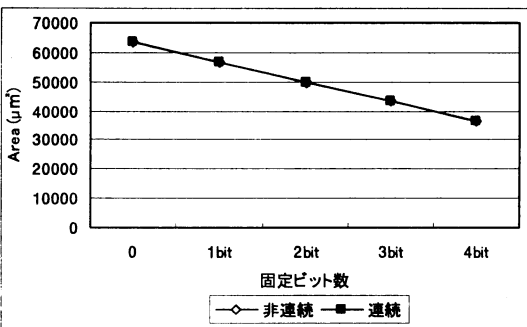


図 4.1 12 ビット乗算器の固定ビット数と Area

固定ビット数	AVE Power (μW)	AVE 割合	RMS Power (μW)	RMS 割合
0	120569.39	100%	38288.53	100%
1	111537.72	88.87	35185.61	91.9
2	102466.34	77.45	32199.74	84.1
3	93384.40	69.21	29349.76	76.65
4	84287.58	60.29	26540.81	69.32

表 5 16 ビット乗算器の固定ビット数と消費電力

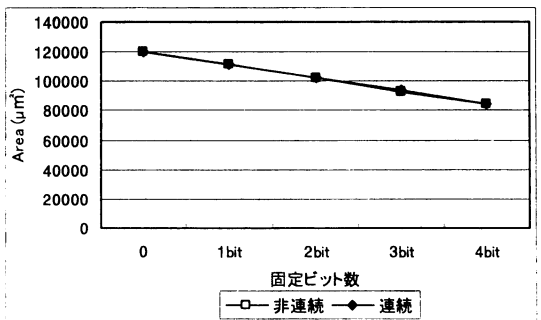


図 5.1 16 ビット乗算器の固定ビット数と Area

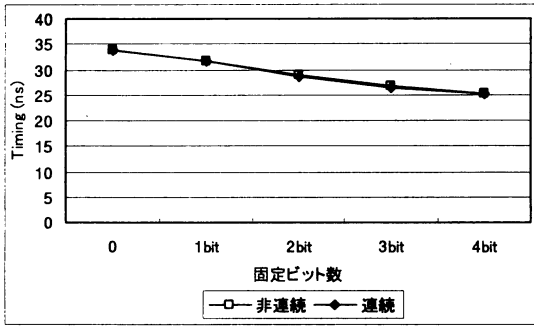


図 5.2 16 ビット乗算器の固定ビット数と Timing

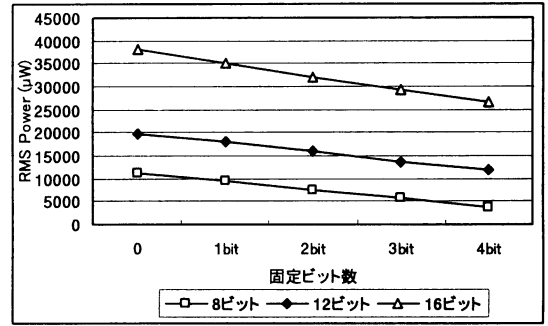


図 6.2 各乗算器の固定ビット数における RMS Power

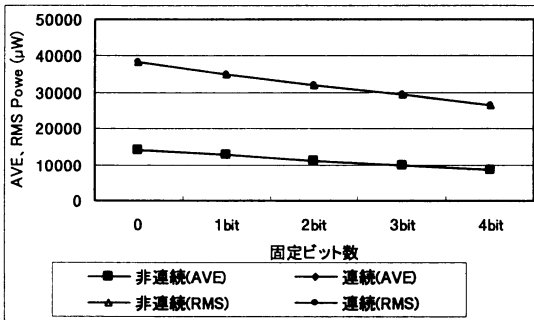


図 5.3 16 ビット乗算器の固定ビット数と AVE, RMS Power

4.4 乗算器のビット数と固定ビット数

乗算器のビット数と固定ビット数のグラフを図 6.1、図 6.2 にそれぞれ示す。8 ビット、12 ビット、16 ビット乗算器もほとんど傾きは変わらない。これらのシミュレーション結果から、8 ビット乗算器で 1 ビット固定数ごとに約 800 μ W、12 ビット乗算器では約 800 μ W、16 ビット乗算器では約 1500 μ W 削減できる。

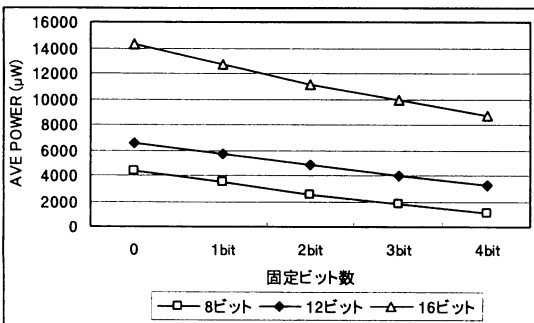


図 6 各乗算器の固定ビット数における AVE. Power

5. まとめ

本稿では、乗算器の低消費電力化の手法の一つとして乗数が 0 で共通する部分を固定する部分定数化を検討した。

8 ビット、12 ビット、16 ビット乗算器において消費電力をシミュレーションし、8 ビット、12 ビット乗算器では 1 ビット固定するごとに約 800 μ W、16 ビット乗算器では約 1500 μ W 消費電力を削減できることがわかった。

今後、最終加算器を含んだ乗算器における低消費電力化を検討していき、通常乗算器と部分定数化した乗算器を LSI に実装し消費電力を実測していく予定である。

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。

文献

- [1] K.C. Chang, "Digital Systems Design with VHDL and Synthesis", IEEE Computer Society Press, 1999.
- [2] Isarael Koren, "Computer arithmetic algorithms -2nd ed". A K Peters, Ltd, Canada, 2002.
- [3] Giovanni De Micheli, "Synthesis and Optimization of Digital circuits", Mcgraw-Hill College, Inc, U.S.A., 1994.