

## 加算器の消費電力の形式による比較

水口 貴之<sup>†</sup> 味元 伸太郎<sup>†</sup> 橘 昌良<sup>‡</sup>

<sup>†</sup> <sup>‡</sup> 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の口 185

E-mail: <sup>†</sup> {095318n, 095319w}@gs.kochi-tech.ac.jp, <sup>‡</sup> tachibana.masayoshi@kochi-tech.ac.jp

あらまし 5月のVLSI設計技術研究会で「加算器の消費電力/面積/速度の形式による比較」を報告した。それに引き続き消費電力のシミュレーションにおいて、シミュレーションパターンを変化させた場合の挙動について検討を行ったので、これを報告する。本研究では、以下の2点について研究を行った。1点目はシミュレーションを行う回数を100回、1,000回、10,000回と変化させた場合に、消費電力に大きな変化が生じるか検討した。2点目は入力到達時間をずらして変化させた場合による消費電力の変化についての検討を行った。

キーワード 加算器, 消費電力, 比較, 回数, 時間

## Comparison of power consumption by form of adders

Takayuki MINAKUCHI<sup>†</sup> Shintaro MIMOTO<sup>†</sup> Masayoshi TACHIBANA<sup>‡</sup>

<sup>†</sup> <sup>‡</sup> Electronic and Photonic Systems Engineering Course, Kochi University of Technology

185 Miyanokuchi, Tosayamada-cho, Kochi 782-8502 Japan

**Abstract** We reported "Comparison of power consumption, area and speed by form of adders " at the VLSI design technical research meeting in May. Because we examined behavior of power consumption when we changed a simulation pattern in simulation of power consumption, we report this. In this research, we researched the following two points. The 1st points, when we changed the number of patterns simulation with 100 times, 1,000 times, 10,000 times, we examined whether a big change occurred in power consumption. The 2nd points, we examined a change of power consumption when we changed input arrival time each bit.

**Keyword** adder, power consumption, comparison, the number of times, time

### 1. はじめに

5月に行われたVLSI設計技術研究会において、加算器の消費電力/面積/速度についての研究を報告したが、シミュレーションを行う回数を変化させた場合に消費電力はどう変化するかという指摘を頂いた。

前回報告した研究では、100パターンの全ビット同時入力とし、その平均消費電力を求める結果となっていたが、パターンを増やした場合における平均消費電力がどのように変動するか、入力タイミングを変動させた場合に消費電力は変わってしまうのかという問題が残った。

本研究では、シミュレーションパターンを100パターン、1,000パターン、10,000パターンの3方式でシミュレーションを行い、変化があるかどうかを検証した。また、ビットごとに入力時間をずらす事で消費電力が変動するかどうかを検証した。これらのデータを元に、消費電力の大きな目安として利用できるデータを得ることが本研究の目的である。

### 2. 前回の研究結果

前回報告した研究結果を以下に示す。

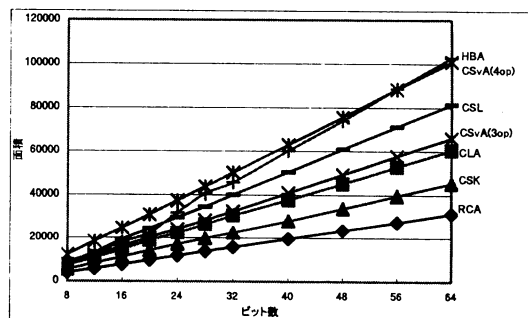


図 2.1 論理合成後の面積

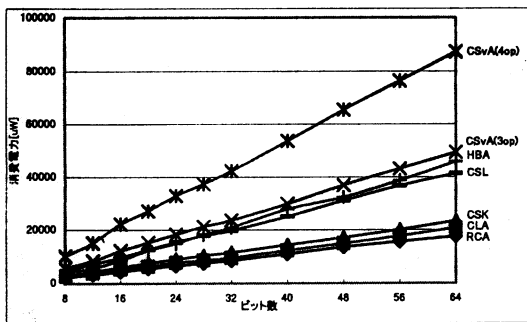


図 2.2 消費電力シミュレーション結果

回路面積では Carry look-ahead adder (以下 CLA) は Ripple carry adder (以下 RCA) の 2 倍近く、Carry skip adder (以下 CSK) は RCA と CLA の中間に位置していたが、消費電力の値は均衡する結果となった。特に、CSK と CLA では面積は CLA の方が大規模であるが、消費電力は CLA の方が少ないという結果が得られた。

Carry select adder (以下 CSL) と Hybrid adder (以下 HBA) は、面積においては 3 オペランド Carry save adder (以下 CSV(3op)) より大きくなっているが、消費電力においては双方とも CSV(3op) を下回っている。4 オペランド Carry save adder (以下 CSV(4op)) に至っては、面積は HBA とほぼ同等であるが、消費電力は他の加算器を引き離して高い値を示した。

### 3. 論理合成、シミュレーション手法

論理合成は前回同様 RCA, CLA, CSK, CSL, HBA の 5 種類の演算器について、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 を用いて行う。

ハードウェア記述言語 VHDL を用い、それぞれの加算形式について、8 ビットを最小、64 ビットを最大として設計する。設計した回路の論理合成には Synopsys 社提供の Design Compiler を用いる。ライブラリは、京都大学版の Rohm0.35um ライブラリを使用する。

論理合成の際には、トランジスタレベルでの最適化は行わず、ライブラリのパラメータをそのまま採用する。また、今回の研究でも回路の配線容量については考慮していない。

消費電力のシミュレーションには、同じく Synopsys 社提供の Nanosim を用いる。ライブラリは論理合成時と同様に、京都大学版 Rohm0.35um ライブラリを用いる。

入力パターンが 100 パターン、1,000 パターン、10,000 パターンである、それぞれのテストベンチを用意し、先ほど合成した演算器に、100[ns]毎にランダムな入力パターンを与え、その演算に消費する電力を得る。

入力到達時間をずらした場合の消費電力のシミュレーションには、最下位ビット (LSB) から順に 1[ns] 毎に入力を与えて、その演算に消費する電力を得る。

シミュレーションにおけるテストベンチは C 言語により自動生成させ、入力パターンは疑似乱数発生関数を用いて作成する。

#### 【入力パターンの乱数発生について】

今回シミュレーションを行う際、入力パターンの乱数発生について検討した。

乱数を発生させた際に 1,あるいは 0 の数のどちらかが極端に多い場合には、消費電力は変動してしまい、結果を出力するには不相当と考えた。最悪の条件を考えると、入力パターンに 0 あるいは 1 が極端に多くなると、低消費電力になったり高消費電力になったりといった、結果に偏りが生じてしまう。

そこで、1 と 0 の発生確率がほぼ同じになるように入力パターンの乱数を操作した。乱数の操作には以下の手順を用いた。

1. 乱数を 2 つ用意する
2. 2 つの乱数を乗算する

#### C 言語プログラム (抜粋)

```
Seed=rand(); // rand 関数で Seed に乱数を発生
Seed2=rand(); // 同じく Seed2 に乱数を発生
Seed = Seed * Seed2 * Seed * Seed2; //乱数操作
```

何故この操作で 1 と 0 がほぼ同じ発生率となるかという、以下の例を参考にして頂きたい。乱数 2 (乗数) に 0 があり、乱数 1 (被乗数) に 1 が多く発生した場合、その被乗数の 1 の数だけ乗数の 0 が結果に反映される。

例)	1 1 0	...乱数 1	例)	1 1 1 1 0	...乱数 1
	0 1 1	...乱数 2		1 0 1 1 1	...乱数 2
	1 1 0			1 1 1 1 0	
	1 1 0			1 1 1 1 0	
	1 0 0 1 0			1 1 1 1 0	
				1 1 1 1 0	
				1 0 1 0 1 1 0 0 1 0	

この操作により、ほぼ同じ回数 1 と 0 が発生するテストベンチ自動生成プログラムが完成し、偏りの無いシミュレーション結果を出力することに成功した。以下に実際に作成したテストベンチを記す。

Verilog テストベンチ (抜粋)

```
#100 input_a <= 8'b10010101;
#100 input_a <= 8'b10010011;
#100 input_a <= 8'b10110001;
#100 input_a <= 8'b11101101;
#100 input_a <= 8'b11011011;
#100 input_a <= 8'b10110110;
#100 input_a <= 8'b10010111;
#100 input_a <= 8'b11001110;
```

4. 結果

今回の研究では以下のようなデータを得る事が出来た。

4.1. パターンによる変化

今回の研究で得られたデータを掲載する。

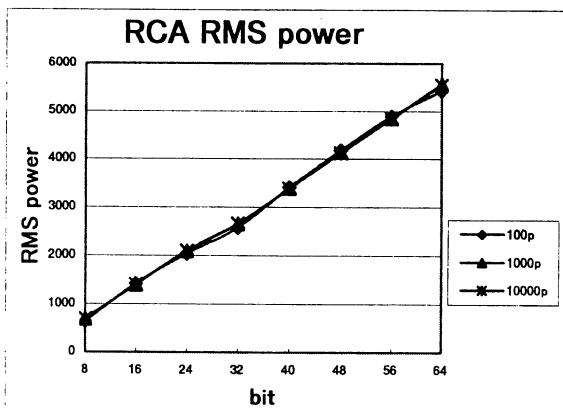


図 4.1.1 RCA の実効消費電力

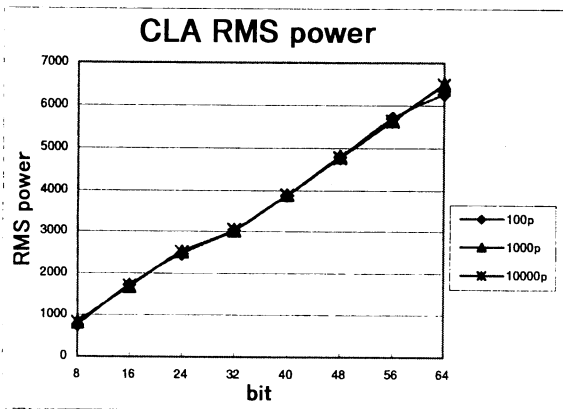


図 4.1.2 CLA の実効消費電力

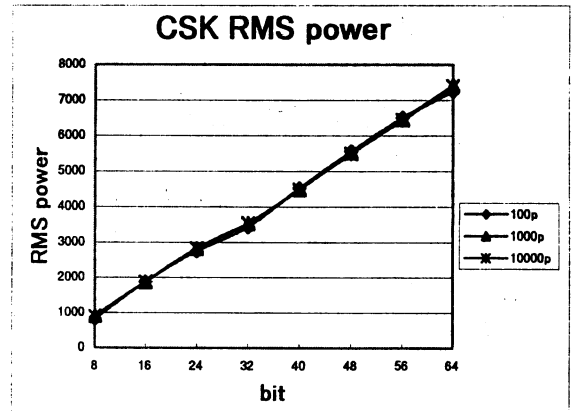


図 4.1.3 CSK の実効消費電力

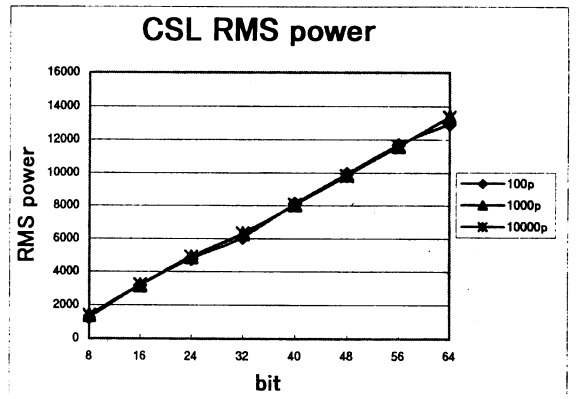


図 4.1.4 CSL の実効消費電力

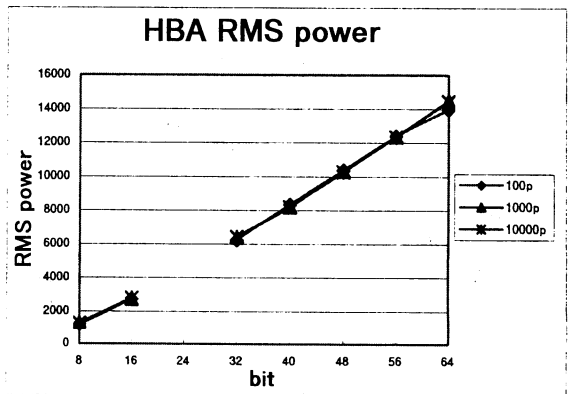


図 4.1.5 HBA の実効消費電力

以上の事から、パターン回数による変化はほとんど見られなかった。

## 4.2. 入力到達時間による変化

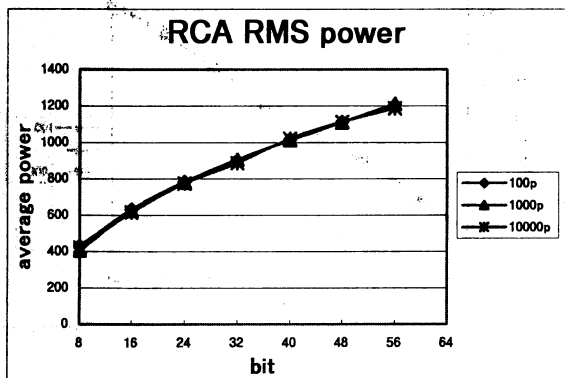


図 4.2.1 RCA の実効消費電力

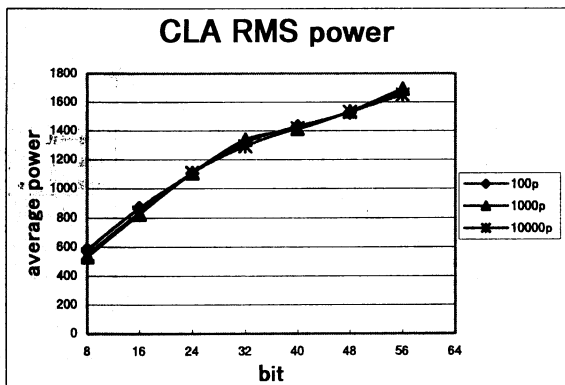


図 4.2.2 CLA の実効消費電力

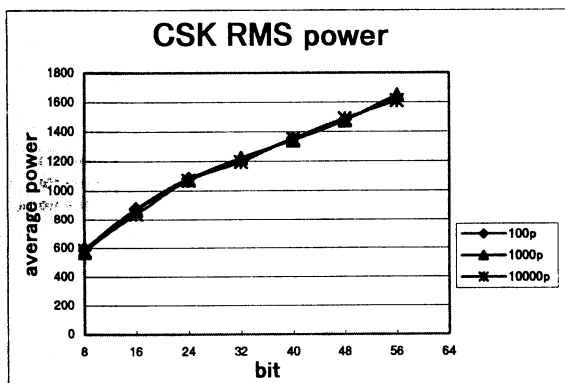


図 4.2.3 CSK の実効消費電力

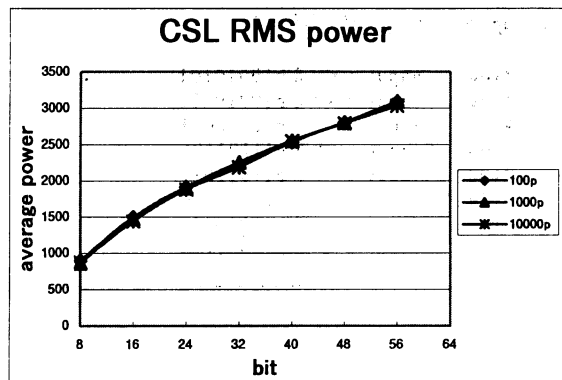


図 4.2.4 CSL の実効消費電力

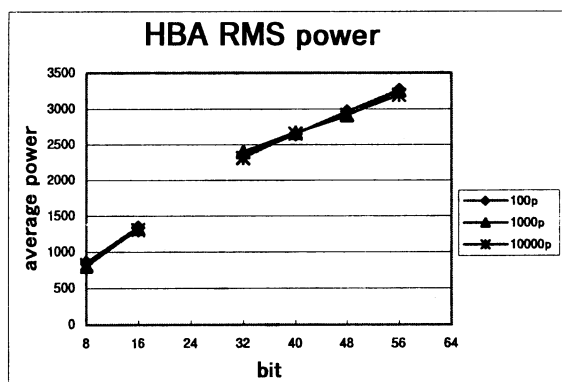


図 4.2.5 HBA の実効消費電力

入力到達時間を変化させた場合、消費電力は log に近い変化をしていくことが判明した。また、こちらもパターン回数を変化させてシミュレーションを行ってみたが、パターン回数による大きな変化はみられなかった。

## 5. 考察

シミュレーションの結果から、パターン数によって消費電力が大幅に変化することは無いので、100 パターンのシミュレーションでも十分な消費電力の値が得られる事が判明した。そこで、形式による消費電力をまとめて比較した結果を記す。

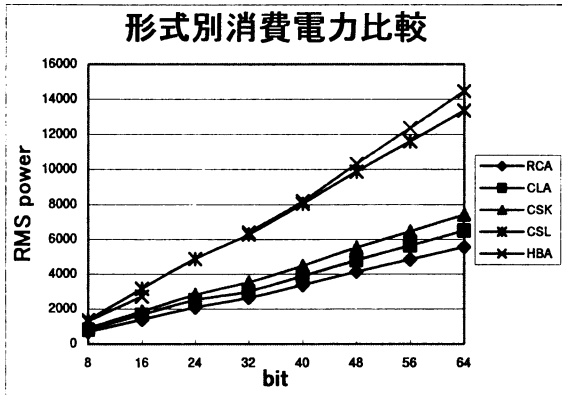


図 5.1 消費電力の形式による比較

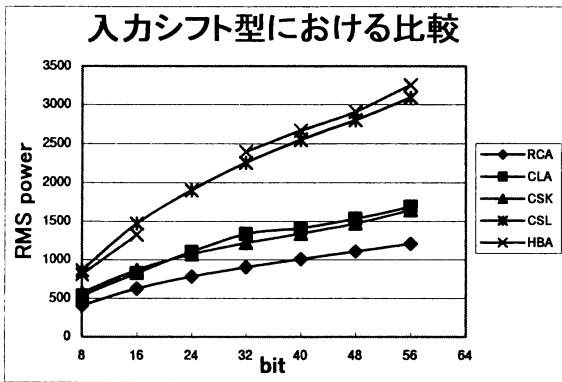


図 5.2 入力時間変化の場合の消費電力比較

また、入力が変化した場合とそうでない場合がどれほどの差があるのかという比較を次に示す。参考例として RCA に着目したものが次の図である。

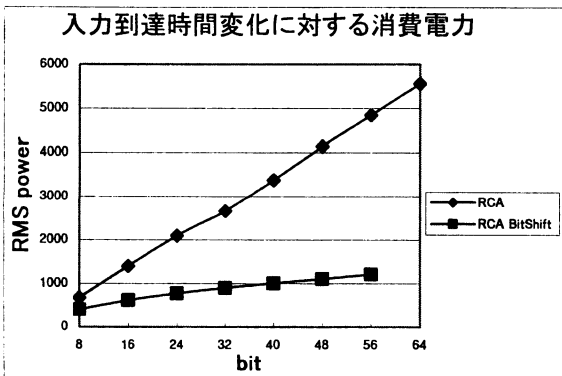


図 5.3 シフトをする場合としない場合の差

これによると、入力到達時間を変化させた場合における消費電力は、同時入力の場合と比べて低消費電力であり、ビット数が増えるほどその差は歴然である。

## 6. 今後の展望

今後もオペランドの到達時間を変化させた際の消費電力について検討を進めたいと考えている。

今回は LSB から順に到達するパターンを考察したが、他にも MSB から順番に到達するパターンや、LSB、MSB から到達し、中間位ビットへの到達が遅いパターンなどを検討する。

到達パターンごとに最適な加算器を検討し、乗算気の最終段加算回路の最適化について考慮する。乗算器の部分積生成回路である Wallace tree 回路は、中間位ビットの演算に最も時間を要するため、最終段の加算回路へのデータ到達時間は不揃いとなる。データの到達順を解析し、それに最適化した加算器の構成を検討していく。

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] 長谷川裕泰, "VHDL によるハードウェア設計入門," CQ 出版社, 東京, 1995.
- [2] Israel Koren, "Computer Arithmetic Algorithms Second Edition," A K Peters, Ltd, Canada, 2001.
- [3] Behrooz Parhami, "Computer Arithmetic ALGORITHMS AND HARDWARE DESIGNS," OXFORD UNIVERSITY PRESS, New York, 2000.
- [4] 味元伸太郎, 水口貴之, 橘昌良, "加算器の消費電力/面積/速度の形式による比較," 電子情報通信学会 VLD 研究会技術報告, VLD2005-2, PP13-17 (2005-05)