

## 遅延変動を考慮したスタンダードセルライブラリの構築と評価

小暮千賀明 今井雅 近藤正章 中村宏 南谷崇†

† 東京大学 先端科学技術センター  
〒 153-8904 東京都目黒区駒場 4-6-1

E-mail: †{kogure,imai,kondo,nakamura,nanya}@hal.rcast.u-tokyo.ac.jp

あらまし 本稿では、Scalable Delay Insensitive (SDI) モデルに基づく非同期式 VLSI 設計において遅延変動に対する標準時のオーバーヘッドを小さくする設計手法を述べる。提案手法では、SDI モデルにおける信号遷移の順序関係を保証するために用いる相対遅延変動率の上限値  $K$  という値を  $K \approx 1$  とするセルライブラリをゲートサイジング等を利用した方法により構築することでオーバーヘッドを小さくする。そして、HSPICE・Design Analyzer を使用して構築したセルライブラリの評価を行なった結果を述べる。

キーワード 非同期式 VLSI・遅延変動・SDI モデル・セルライブラリ

## A Design Method for a Standard Cell Library Considering Delay Variation

C. KOGURE, M. IMAI, M. KONDO, H. NAKAMURA AND T. NANYA†

† The University of Tokyo, Research Center for Advanced Science and Technology  
The University of Tokyo, 4-6-1 Komaba, Meguro-ku, Tokyo 153-8904, Japan  
E-mail: †{kogure,imai,kondo,nakamura,nanya}@hal.rcast.u-tokyo.ac.jp

**Abstract** This paper describes a design method for asynchronous circuits that have small overhead in the standard condition for various environmental changes and process variations. The method is an SDI-model-based design methodology. In this model, the value of  $K$  represents the relative delay variation and an overhead of the circuit in the standard condition. We present how to design a cell library that is consisted of gates. The gates are adjusted to their  $K$  by using gate-sizing and so on. An overhead of the circuit made from the cell library is small. Some circuits based on the proposed method are simulated using the HSPICE and the Design Analyzer.

**Key words** Asynchronous VLSI, delay variation, SDI model, cell library.

### 1. はじめに

現在、VLSI システムでは、MOS の最小加工寸法が  $0.13\mu\text{m}$ 、 $0.09\mu\text{m}$  と微細化が進んでいる [9]。微細化によりゲートのスイッチング速度が向上し数 GHz のクロック周波数による演算が可能となっている。クロック周波数の高速化は消費電力を増加させ、発熱も増加させる。また、チップのサイズは縮小していないので相対的に配線遅延が増加している。高速化するほどに僅かなタイミングの誤差が誤動作を引き起こす。実際には、製造プロセスのばらつきや、温度・電圧などの環境変動に応じた遅延変動が起きている [1]。

本研究では、遅延変動に焦点を当てる。遅延変動が生ずる場合でも正しく動作し、かつ標準状態におけるオーバーヘッドの小さくなる設計手法について提案する。

VLSI システムには、クロックを使用する同期式 VLSI と使用しない非同期式 VLSI があり、従来、同期式が多用されている。同期式では、遅延変動に対応するためには、あらゆる変動状態において遅延が最大となる状態におけるクリティカルパスの遅延よりも大きなサイクルタイムを持つクロックを用いるので標準状態においてオーバーヘッドが大きくなる。もし、少しでもオーバーヘッドを小さくするために最悪遅延からのマージンを小さくした場合は遅延変動が生じたときに誤動作を起こし易い回路となる。それに対して非同期式 VLSI は遅延変動に対して耐性を持つ。なぜなら、非同期式 VLSI は要求応答信号に基づいて動作するので前の演算が終了しない限り次の演算は行なわれず、遅延が伸びた場合でもその遅延に合わせて回路が動作するからである。

非同期式 VLSI 設計では、素子や配線の遅延に関して設ける

仮定, すなわち遅延モデルが重要な役割を果たす。遅延変動に対して楽観的な遅延モデルに基づいて設計された回路は, 想定外の遅延変動が生じると誤動作を引き起こしやすく, 遅延変動に対して悲観的な遅延モデルに基づいて設計された回路は, 速度や消費電力の面でのオーバーヘッドが大きくなりやすい。近年では, これらのオーバーヘッドを低減するため, 遅延予測が可能な局所領域の設計では, 遅延情報を積極的に利用する設計方式 [2], [6], [8], あるいは素子遅延の全てを等しくする設計方式 [7] 等が注目されている。本研究では, 遅延モデルの一つである Scalable Delay Insensitive (SDI) モデルに基づいて設計を行なう [4]。SDI モデルは, 局所領域の変動状態を等しいと見なすことで, 制約条件の厳しい他の Delay Insensitive (DI) モデルや Quasi Delay Insensitive (QDI) モデルと比べて現実的でオーバーヘッドの小さい回路が設計ができる。以下では, SDI モデルに基づいた設計手法の一つを提案する。二章では, SDI モデルに関して述べる。三章では, SDI モデルに基づいた設計において標準状態におけるオーバーヘッドの小さくなるセルライブラリの設計手法を述べる。四章では, 三章で提案した手法の評価を述べる。五章でまとめる。

## 2. SDI モデル

### 2.1 SDI モデルの定義

SDI モデルは, チップ化された後の回路の環境変動に応じた遅延変動はチップ全体でほぼ一様に起きると予測されることに基づいた遅延モデルである [5], [8]。以下に SDI モデルの定義を述べる。

- SDI モデル [5], [8] : ある回路要素  $C$  に対して, 設計者が設計段階で予測した遅延を  $D_e$  とし, システムの生涯を通して起こりうる実際の遅延を  $D_a$  とする。このとき,  $R = D_a/D_e$  は回路要素  $C$  の時刻  $t$  における遅延変動率 (scaling ratio) を表す。任意の二つの回路要素  $C1$  と  $C2$  の時刻  $t$  における遅延変動率をそれぞれ  $R1, R2$  とすると,  $V = R2/R1$  は時刻  $t$  における相対遅延変動率 (scaling variation) を表す。このとき, 回路には定数  $K$  ( $K \geq 1$ ) (variation factor) が存在し, 任意の二つの回路要素の間の相対遅延変動率  $V$  に関して, システムの生涯を通じて  $1/K \leq V \leq K$  が常に成り立つ。

### 2.2 SDI モデルに基づく設計制約

このモデルに基づいた設計における制約に関して述べる。非同期式 VLSI 設計では, データバスにおけるデータ信号とその安定を示す完了信号の関係のように, 順序関係を保証しなければならない信号遷移が存在する。図 1 の様な経路を想定する。仕様として, 回路中の信号遷移  $t1$  が  $t2$  よりも早く生じるとする。 $t0$  から  $t1$  と  $t0$  から  $t2$  までのそれぞれの経路での遅延を  $D_{e1}, D_{e2}$  とする。変動が起きたときの遅延を  $D_{a1}, D_{a2}$  とすると, 遅延変動率は  $R1 = D_{a1}/D_{e1}, R2 = D_{a2}/D_{e2}$  となる。このとき, SDI モデルの定義より,

$$\frac{1}{K} < V = \frac{R2}{R1} = \frac{D_{a2} \cdot D_{e1}}{D_{e2} \cdot D_{a1}} < K \quad (1)$$

式変形をして,

$$\frac{D_{e2} \cdot D_{a1}}{D_{a2}} < K \cdot D_{e1} \quad (2)$$

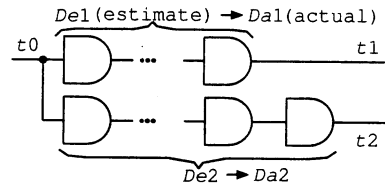


図 1 想定する経路の模式図

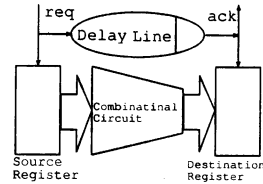


図 2 東データ転送方式に基づく回路の模式図

$$\frac{D_{a2} \cdot D_{e1}}{D_{a1}} < K \cdot D_{e2} \quad (3)$$

このとき, 二つの規則が成立する (証明は省く)。

規則 1 :  $K \cdot D_{e1} < D_{e2}$ , ならば,  $D_{a1} < D_{a2}$  ;

規則 2 :  $K \cdot D_{e2} < D_{e1}$ , ならば,  $D_{a2} < D_{a1}$

したがって, この場合は規則 1 を適用することで仕様を満たすことができる。規則 1 に基づいて設計を行なった場合, 仕様を満たした上でオーバーヘッドが少ないのは,  $K$  が 1 に近いときである (今後は  $V_{max} \approx 1$  と表現していく)。なぜなら, 標準状態の遅延は  $K \cdot D_e$  になるからである。 $K$  の値はプロセス変動や電圧・温度などの環境変動に応じて決定される。

### 2.3 SDI モデルに基づく東データ転送方式の設計

本研究ではレジスタ間のデータ転送方式として, 対象となる組み合わせ回路の安定を示す 1 ビットのタイミング信号を付加する東データ方式を用いる。図 2 の様な回路で示される。図 2 の上方の素子が遅延素子であり, 下方が組み合わせ回路である。その動作は, まず, 要求信号がソースレジスタへ入り組合せ回路におけるデータの入力をさせ, 同時にデスティネーションレジスタへレジスタ使用のための信号が入力される。データの保持完了と共に応答信号がソースレジスタより返される。従って, 2.2 節における  $t1$  が組み合わせ回路におけるクリティカルパスに相当し,  $t2$  が遅延素子に相当する。

### 2.4 セルライブラリと $V_{max}$ の関係

図 3 に, ある標準的なセルライブラリに対して遅延素子を使用した場合の  $V_{max}$  の分布図の例を示す。上図は遅延素子が AND 二段 (図 4 に示す) の場合で, 下図は OR 二段+INV の場合である (遅延素子の AND 等の段数は可変である)。 $V_{max}$  は様々な値をとる。各素子ごとに  $V_{max}$  の分布が異なり, 遅延素子の違う場合でも分布が異なる。また, 立上り (rise: ×) と立下り (fall: +) に関しても分布が異なる。

## 3. 提案する設計手法

2.2 節の規則 1 に基づいて設計を行なった場合, 仕様を満たした上で標準時においてオーバーヘッドが少ないのは,  $V_{max} \approx 1$

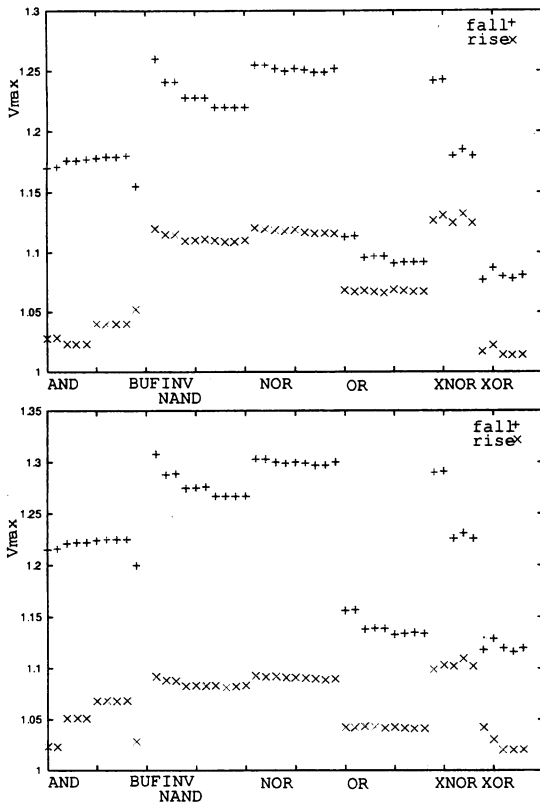


図3 セルライブラリの各素子と  $V_{max}$  の関係。上図は遅延素子が AND 二段で、下図は OR 二段+INV の場合である。



図4 使用した遅延素子 AND 二段

( $K \approx 1$ ) の時である。従って、 $V_{max}$  を1に近づけるのが重要である。これを実現する方法として、予め  $V_{max}$  が小さい素子を用意する方法と、組み合わせ回路を合成するとき、あるいは、合成後に遅延素子と組み合わせ回路の関係を調整して  $V_{max}$  を小さくする方法の二種が存在すると考えられる。

前者の方法は  $V_{max}$  が小さいセルライブラリを構築し、それを使用して組み合わせ回路を作成するという設計手法である。組み合わせ回路の  $V_{max}$  は、素子の  $V_{max}$  以下であるので、必ず、組み合わせ回路の  $V_{max}$  は小さくなる。配線についてはゲート出力に配線容量を付加した。配線容量に関しては、予め大きな負荷容量を素子に課してセルライブラリを構築すればよい。なぜなら、実際の負荷容量が小さかった場合は、 $V_{max}$  はこれより小さくなるからである。そして、予めセルライブラリを調整した場合、その後の処理ではどんな組み合わせ回路を合成する場合でも手間は変わらない。

後者の方法では、合成すべき回路の構成、規模は様々に変化する。さらに、異なる状態における遅延変動率  $R$  は素子毎に

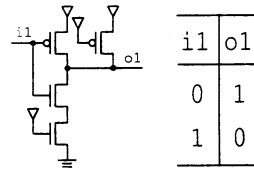


図5 NAND 型の INV

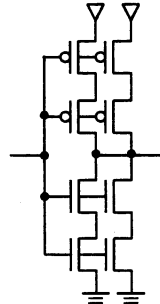


図6 INV の段数を変化させた図でこの場合は  $2 \times 2$  である。

大きく異なるため状態によってクリティカルパスが変動しうる。従って、組み合わせ回路ごとに遅延素子に対応させるのは困難である。

そこで、前者による設計手法を今回新たに提案する。このとき、セルライブラリの構築方法が問題となる。セルライブラリは  $V_{max} \approx 1$  となる素子の集合によって構築される必要がある。そのため、素子の構成などを変化させて遅延を調整し  $V_{max} \approx 1$  となる素子を設計する。以下では、 $V_{max} \approx 1$  となる素子を設計する方法について述べる。三種類を挙げる。

(1) 回路構成を等しくする方法

素子の構成が等しいセルライブラリを構築するために NAND のみで様々な素子を設計する。これは、例えば、NAND 型の INV を図5の様に設計するということである。回路構成は NAND であるが、一方のゲート入力を「1」に固定することで INV を実現する。他の素子 AND や OR などはこの INV や NAND を組み合わせる設計できる。同様に NOR のみでも様々な素子を設計する。

(2) 構成を変更する方法

図3での  $V_{max}$  立上りと立下りの分布差を考えると、素子の  $V_{max}$  を小さくするには、立上り遅延と立ち下がり遅延の関係を調整することにより  $V_{max} \approx 1$  となる素子を設計できると思われる。ここでは、CMOS の構成を変えることで遅延を変化させる。ここでの構成とは INV を基に説明すると、図6の様に縦横に CMOS の段数に変化を加えることである。ただし、縦の段数を増やした場合は遅延に、さらに面積的に問題のある方法である。また、通常の設計における制約として縦は4段までがせいぜいである。

(3) ゲートサイジングによる方法

ゲートサイジングによって  $V_{max}$  を変化させる方法を述べる。ここで述べるゲートサイジングとはチャネル長やチャネル幅を

変化させることである。PMOS と NMOS のゲートサイズを適切に設計することで  $V_{max} \approx 1$  を実現できると思われる。一般にチャンネル長を伸ばした場合は遅延が増加し、チャンネル幅を広げた場合は遅延は減少する。ただし、いずれを大きくした場合でも当然面積は増加する。

#### 4. 提案手法の評価

##### 4.1 シミュレーション環境と評価手順

0.13 $\mu\text{m}$  プロセス技術における HSPICE シミュレーションにより遅延を求めた。セルライブラリ ( $t1$  を引き起こす素子) と遅延素子 ( $t2$  を引き起こす素子) について遅延を立上りと立下りでシミュレーションした。セルライブラリの各素子として INV・NAND・NOR・BUF・AND・OR・XOR・XNOR について HSPICE により評価した。また、遅延素子は、図 4 の AND 二段、もしくは、OR 二段+ INV を使用した。

このとき、

- 標準状態
  - 25 $^{\circ}\text{C}$ , 1.2V, CMOS の速度普通
- 変動範囲 ( $2^4 = 16$  通りについて測定)
  - 温度: 25 $^{\circ}\text{C}$ , 100 $^{\circ}\text{C}$
  - 電圧: 1.1V, 1.3V
- 加工: PMOS と NMOS の (速い・遅い) の組み合わせ

において、出力結果の遅延情報を基に遅延変動率  $R$  を算出した。タイミング信号は立上りで意味があるとしたので、遅延素子の立上りについての  $R2$  と各素子の  $R1$  とで、相対遅延変動率  $V$  を算出した。算出された  $V$  の中で最高値のものを各素子における  $V_{max}$  とした。

##### 4.2 セルライブラリの構築方法の評価

###### (1) 回路構成を等しくする方法

セルライブラリの構成を NAND によって作成した。遅延素子もセルライブラリ内にある AND を使用して AND 二段のものを作成した。同様にして NOR 型のセルライブラリも作成した。ただし、相性のために NOR 型のセルライブラリでは遅延素子を OR 二段+ INV とした。これらのセルライブラリの各素子の  $V_{max}$  を HSPICE シミュレーションによって求めた。

結果は図 7、図 8 となった。図の様に、 $V_{max}$  の値は十分に小さくならない。NAND で設計した INV と NOR で設計した INV の振る舞いは殆ど標準の INV と変わらないので、大きく  $V_{max}$  が変化しなかったと思われる。従って、この方法だけでは求める  $V_{max} \approx 1$  にはならない。

###### (2) 構成を変更する方法

構成を変化させる方法について INV を使って評価した。PMOS・NMOS 共に縦横 4 段づつ変化させた。構成として計  $4^4$  の場合を評価した。結果は図 9 のようになる。上方に分布しているのは立下り (+) に関してが多く、下方に分布しているのは立上り (x) に関してが多くなった。この場合では、PMOS の個数の増加よりも、NMOS の個数の増加によって  $V_{max}$  が小さくなった。図 9 より、重要なこととして、INV の様な入力のトランジスタゲートから出力のドレインまでが、一段である様な構成について  $V_{max}$  を小さくすることは困難であることが言

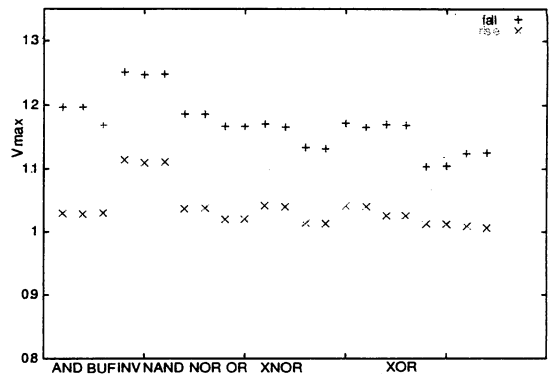


図 7 NAND を利用して構築したセルライブラリの  $V_{max}$  の分布

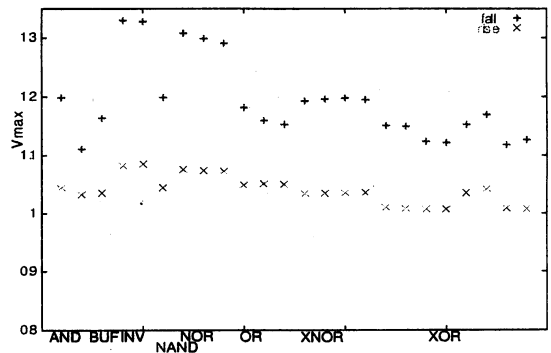


図 8 NOR を利用して構築したセルライブラリの  $V_{max}$  の分布

える。

###### (3) ゲートサイジングによる方法

今回は PMOS・NMOS のチャンネル幅を変化させた。チャンネル長については常に一定とした。チャンネル幅の変更は CMOS 毎に行なった。まず、2 入力 AND についてチャンネル幅を変化させた。AND は NAND と INV を合わせて作成される。ここでは、NAND の部分のチャンネル幅を NMOS を 11 通り、PMOS を 9 通り、INV の部分のチャンネル幅を NMOS を 11 通り、PMOS を 8 通りにそれぞれ変化させた。従って組み合わせは  $11 \times 9 \times 11 \times 8$  通りとなる。これらに対して 4.1 で解説した処理を行なった。

結果は、図 10 のようになる。ゲートサイジングにより様々な  $V_{max}$  を取るということが分かる。また、NAND や INV という部分ごとの CMOS のチャンネル幅に対しておおよそ線形となる。幅が広いほどに  $V_{max}$  は小さくなる。ゲートのサイズ次第では  $V_{max} \approx 1$  を実現した。

三種類の方法を述べた。回路構成が等しいだけの素子では  $V_{max}$  を小さくできず、INV の様な単純な構成の素子では  $V_{max}$  を小さくできない。AND の様な NAND+INV の構成を持つ素子ではゲートサイジングという方法で  $V_{max} \approx 1$  を実現できた。構成を等しくすることでは  $V_{max}$  を小さくすることはできず、立上り遅延と立下り遅延の関係をただ変えるだけでは  $V_{max}$  を小さくすることはできない。

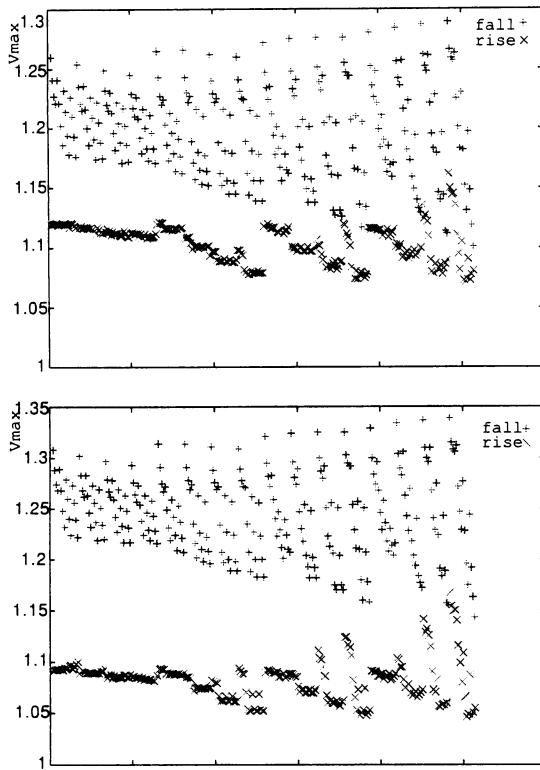


図9 構成を変化させた INV の  $V_{max}$  の分布図。上図は遅延素子が AND 二段で、下図は OR 二段+INV の場合である。

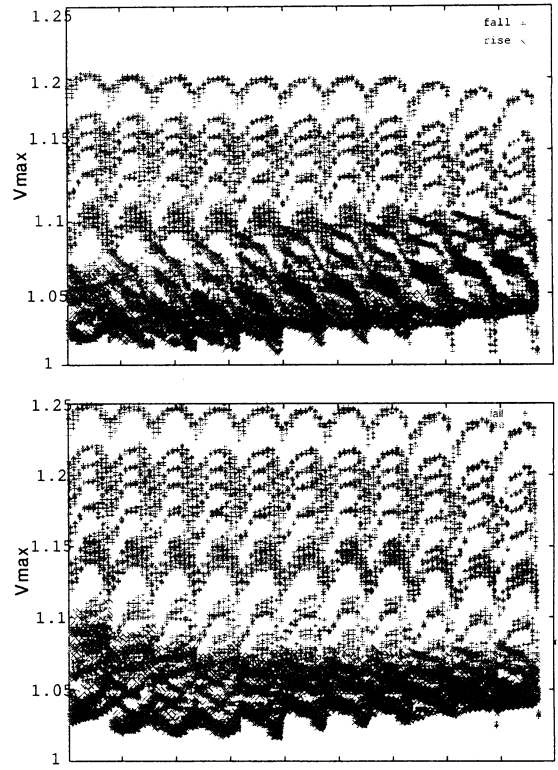


図10 ゲートサイジングをした2入力 AND の  $V_{max}$  の分布である。上図は遅延素子が AND 二段で、下図は OR 二段+INV の場合である。

入力信号が入力側のトランジスタゲートから出力側のドレインまで一段で出力される INV・NAND・NOR などの回路ではゲートサイジングのみでは  $V_{max} \approx 1$  を実現できない。そこで、INV では INV を三個直列につなげ、NAND 等では AND+INV の様につなげた後にゲートサイジングを行なう。AND 等の他の素子では入力信号は入力側のトランジスタゲートからドレインへさらにもう一段以上続いてトランジスタゲートからドレインまで行って出力される構成になっているので、ゲートサイジングによって  $V_{max} \approx 1$  を実現できる。結論として、入力側のトランジスタゲートに入力信号が入ってドレインより信号が出力され、さらに、もう一段トランジスタゲートに入力が入ってドレインより信号が出力される構成とした後で、ゲートサイジングによって立上りと立下りの遅延を調整することで  $V_{max} \approx 1$  を実現できる。

### 4.3 8ビット加算器による提案手法の評価

#### 4.3.1 ゲートサイジングによるセルライブラリの構築

ゲートサイジングを利用して  $V_{max} \approx 1$  となるような INV・NAND・NOR・BUF・AND・OR・XOR・XNOR で構成されるセルライブラリを構築した。PMOS・NMOS のチャンネル幅を変化させた。チャンネル長については常に一定とした。チャンネル幅の変更は CMOS 毎に行なった。そして、チャンネル幅は小さめと大きいめの2通りで変化させた。従って、CMOS  $n$  個から構

成される素子では  $2^n$  通りで変化させた。その中で、 $V_{max} \approx 1$  であるものを選択した。これを各素子ごとに行なった。ただし、4.2節のから、INV に対して  $V_{max} \approx 1$  は INV を直列に3個つなげる構成をとった。NAND と NOR も同様に、それぞれ AND と INV をつなげたもの、OR と INV をつなげたものを使用した。作成したセルライブラリの遅延・面積は表1となる。遅延については各素子ごとに適用前の立上り遅延に関して正規化を行なった。選択の基準として今回は遅延が同じくらいであるもので、 $V_{max} \approx 1$  であるものを選んだ。結果としてゲートのサイズが大きいものを使用しているので面積の大きさが増加する。また、INV、NAND、NOR は上記の理由により面積が増大する。

#### 4.3.2 手順

4.3.1で選択したセルライブラリと適用前サイズのセルライブラリについて、各素子に対して HSPICE を用いて遅延情報を求め、その情報を用いて Design Analyzer 用のデータベースを作成し8ビット加算器を合成して評価した。

#### 4.3.3 結果

まず、合成をかけたときの Design Analyzer による解析結果を表2に示す。無制約で合成した場合とタイミング制約をかけた場合の面積・遅延を示す。面積は適用前より明らかに大きくなった。遅延はそれほど変化はない。

次に、合成結果の VerilogHDL 記述を HSPICE 用記述に変換

表 1 セルライブラリの比較表

素子名	標準遅延 (各素子ごとに正規化)		面積 (μgrid)	
	適用前 (rise, fall)	提案手法 (rise, fall)	適用前	提案手法
INV	1.00, 0.85	1.21, 1.42	2	6
NAND2	1.00, 2.60	2.16, 2.59	3	7
NOR2	1.00, 0.23	0.74, 0.69	3	7
AND2	1.00, 0.65	0.86, 0.49	4	5
AND3	1.00, 0.49	0.89, 0.33	5	6
OR2	1.00, 1.40	0.72, 1.20	4	5
OR3	1.00, 1.95	0.76, 1.88	5	6
BUF	1.00, 0.91	0.80, 0.87	3	4
XNOR2	1.00, 0.64	0.90, 0.70	10	11
XOR2	1.00, 1.14	0.86, 0.85	10	11

表 2 8ビット加算器の合成結果

制約 [ns]	適用前		提案手法	
	面積 (μgrid)	遅延 [ns]	面積 (μgrid)	遅延 [ns]
無制約	244	2.10	312	1.58
0.5	480	0.73	570	0.57
1.0	320	0.93	508	0.80
1.5	316	1.22	326	1.21

表 3 8ビット加算器のシミュレーション結果

ケース 1	面積 (μgrid)	遅延 [ns]	$V_{max}$	加算器の遅延 [ns]
適用前	311	0.686	1.127	0.773
提案手法	414	0.718	1.057	0.758
ケース 2	面積 (μgrid)	遅延 [ns]	$V_{max}$	加算器の遅延 [ns]
適用前	264	0.972	1.133	1.101
提案手法	312	1.007	1.076	1.064

シミュレーションを行なった。表 3 に面積、標準遅延、 $V_{max}$ 、遅延  $\times V_{max}$  (加算器の標準時の遅延) を示す。表 3 より、提案手法の設計法では回路の  $V_{max}$  が小さくなる。遅延は適用前の方が小さいにもかかわらず、加算器の遅延は提案手法の方が小さくなる。回路構成が同じ場合はセルライブラリの  $V_{max} \approx 1$  とすることで、回路も  $V_{max} \approx 1$  となり標準時の遅延のオーバーヘッドを減少できる。しかし、セルライブラリの各素子の面積が大きいため、同じ構成の加算器を作った場合の面積は大きくなる。提案手法では、面積に制約をかけた場合は優位性がない。これは、ゲートサイジングの最適性の甘さにあると思われる。面積・遅延を損なわない様にして  $V_{max} \approx 1$  を達成する必要がある。

## 5. まとめ

遅延変動に対してロバストかつオーバーヘッドの小さい回路の設計法について提案した。SDI モデルに基づいて設計する場合  $v_{max} \approx 1$  となるセルライブラリを予め構築し、そのセルライブラリを用いて回路を合成することで遅延変動のある場合でもオーバーヘッドの小さくなる回路が設計されることを示した。同時に、セルライブラリの構築方法についても幾つか述べた。トランジスタの入力から出力までの段数を複数段にした後にゲートサイジングをすることで  $V_{max} \approx 1$  は実現された。

提案手法は、配線遅延変動を考慮した場合・局所領域での環境変動を考慮した場合については取り上げていない。今後、トランジスタの微細化と共に影響が現れるであろうこれらの遅延変動を含めて考える必要がある。

謝辞 本研究の遂行にあたり、文部科学省科学研究費補助金若手研究 16700051 のご支援を頂いた。また、本研究の一部は(株)半導体理工学研究センターとの共同研究によるものである。

## 文 献

- [1] A. Chandrakasan et al. : *DESIGN OF HIGH-PERFORMANCE MICROPROCESSOR CIRCUITS*, IEEE Press, chapter 6, pp. 98-115, 2000.
- [2] J. Cortadella et al. : *Lazy transition systems and asynchronous circuit synthesis with relative timing assumptions*, IEEE Trans. on CAD of IC and Systems, Vol. 21, No. 2, Feb. , 2002.
- [3] S. Hauck : *Asynchronous Design Methodologies: An Overview*, Proceedings of the IEEE. Vol. 83, No. 1, Jan. , 1995.
- [4] 今井雅 et al. : SDI モデルに基づく局所同期型非同期式 VLSI 設計方式, 情報処理学会, 情報処理学会論文誌, Vol. 44, No. 5, May, 2003.
- [5] T. Nanya, et al. : *Scalable-Delay-Insensitive Design : A high-performance approach to dependable asynchronous systems*, Proc. International Symp. on Future of Intellectual Integrated Electronics, pp. 531-540, Mar. , 1999.
- [6] K. Stevens et al. : *Relative timing*, Proc. International Symposium on Advanced Research in Asynchronous Circuit and Systems, pp. 208-218, April, 1999.
- [7] I. Sutherland et al. : *GasP : A minimal FIFO control*, Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems, pp. 46-53, IEEE Computer Society Press, March, 2001.
- [8] A. Takamura et al. : *TITAC-2 : An asynchronous 32-bit microprocessor based on scalable-delay-insensitive model*, Proc. ICCD, pp. 288-294, Oct. , 1997.
- [9] ITRS : <http://public.itrs.net/>