

# VLSIの微小遅延検出のための組み込み遅延測定方式の研究

代表研究者 加藤 健太郎 鶴岡工業高等専門学校 准教授

## 1 はじめに

VLSIの製造工程において、その出荷前に行う製造テストは、その高信頼化において非常に重要な位置付けを有する。インテル、AMD製のパソコン向けCPUが顕著な例であるが、今日のVLSIの動作速度は飛躍的に高速化しており、逆に動作電圧は低消費電力のニーズに伴い低下している。このため、VLSIの故障において信号伝搬のタイミングの遅れに起因する故障が大きな比重を占めてきており、遅延故障テストが大きな課題となってきた。特に近年の高スペックな電化製品、PCなどに用いられるVLSIでは、Go-Nogoテストでは検出できないビアの高抵抗化や配線間のブリッジなどによる故障、すなわち微小遅延故障が大きな問題となっている。微小遅延故障は経年変化によりやがて動作不良をもたらすことが多く、このようなチップは出荷前の製造テストにおいて除かれる必要がある。微小な異常遅延の検出、診断には、遅延測定によるアプローチが有効である。そこで、本研究でもこの遅延時間の測定を目指している。しかし、チップの動作速度の高速化に伴い、外部試験装置を用いた遅延測定はコスト的に大きなボトルネックとなってきた。またVLSIの内部パスの遅延測定は、外部試験装置が高性能であったとしても困難である。故に遅延測定は、一部若しくはすべての測定回路を内部に組み込むオンチップ方式が不可欠であり、外部試験装置を要したとしても安価で低速なもので済ませる方式、若しくは外部試験装置を全く要さない方式が望ましい。本研究では、低コストかつ低消費電力なオンチップの効率的遅延測定方式の考案、検討を研究課題とする。

オンチップの遅延測定法にはさまざまな手法があるが[1], [2], 組み込み遅延測定回路を用いた手法は最も著名なものの一つである[3], [4]。組み込み遅延測定回路を用いた手法は、一般に測定分解能が高いため、その微小遅延テストのテスト品質が高い。また測定の際通常のテストと異なりテスト応答のキャプチャ動作が必要ないためIRドロップによる測定精度の低下が少ないという長所を有する。

しかしながらこれを用いた測定系は1回の測定で実装されている組み込み遅延測定回路と同数のパスしか測定できないという制約がある。このため通常のスキャンテストと同様のテスト手順では測定時間及び測定に要するデータ量が膨大となるという問題点があった。

本報告では、組み込み遅延測定回路を用いたオンチップ遅延測定の測定コスト削減のための2つの手法を提案する。1つは測定時間削減のための手法である時分割遅延測定法である。2つ目は、1つ目の時分割遅延測定適用時の測定データ量の削減法である。本報告の後の構成を述べる。まず2においてオンチップ遅延測定による統計的微小遅延検出法の概要及び組み込み遅延測定回路を用いたオンチップ遅延測定系について述べる。3において1つ目の提案である時分割遅延測定法について述べる。続いて4において2つ目の提案である時分割遅延測定適用時の測定データ量の削減法を説明する。5においてこれら2つの手法の評価実験及び実験結果に基づく考察について述べる。最後に研究のまとめを述べる。

## 2 準備

### 2.1. オンチップ遅延測定による統計的微小遅延故障検出法

本手法では、文献[1]で提案されているパス遅延測定に基づく統計的微小遅延テスト判定を用いる。あるテスト対象パス $p$ の遅延分布が正規分布 $N(\mu_p, \sigma_p)$ に従うものとする(図1.(a))。パス $p$ を、統計的手法を用いて遅延故障テストする場合、許容遅延時間 $T_p$ は $p + m\sigma_p$  ( $m > 0$ )となる。この時 $p$ をパス遅延の実測により求める必要があるが直接測定方式の組み込み遅延測定回路では測定値には冗長配線の遅延が含まれるため、測定精度が悪化する。一方、図1.(b)の $p$ を部分パスとして含む経路 $p_m$ の微小遅延検出を考える。 $p$ の入出力にそれぞれ正規分布 $N(\mu_{ck}, \sigma_{ck})$ ,  $N(\mu_{sg}, \sigma_{sg})$ に従う冗長配線 $p_{ck}$ ,  $p_{sg}$ が接続されているものとする(図1.(b))。その許容遅延時間 $T_{pm}$ は、 $(\mu_{ck} + \mu_p + \mu_{sg}) + m\sqrt{\sigma_{ck}^2 + \sigma_p^2 + \sigma_{sg}^2}$ となる。 $p$ の微小遅延検出は、 $p_m$ の遅延量を $T_{pm}$ と比較する事によっても行う事ができる。

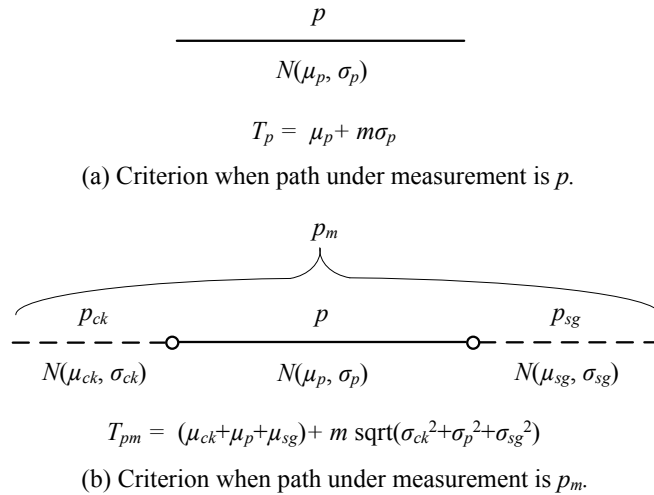


図1 冗長配線を含んだ経路の測定による統計的微小遅延検出。

## 2.2. 組み込み遅延測定回路による遅延時間測定系

図2に想定する組み込み遅延測定回路を用いた遅延測定系を示す。図中のDVMC (Delay Value Measurement Circuit) は直接測定方式の組み込み遅延測定回路であり、CUTは測定パスを含む組み合わせ回路である。 $clk$ はクロック線である。 $clk$ はDVMC、CUT双方に接続されている。 $FF_0 \cdots FF_{N-1}$ は、回路中のフリップフロップを示す。各フリップフロップは、スタンダードスキャン設計がなされており、テストモードでは、 $sci$ をスキャン入力とする被線で示されるスキャンパスが構成される。DVMCには2本の入力線  $start$ 、 $stop$  が存在する。この回路は  $start$  に遷移が入力されてから  $stop$  に遷移が入力されるまでの遅延測定ができる。CUTには、 $p_0 \cdots p_{n-1}$ の  $n$ 本の内部パスが存在する。内部パスの入出力は、フリップフロップに接続されている。各内部パスの出力端は、冗長配線を介して信号遷移をDVMCに伝えるためのストップシグナルジェネレータ(SSG ( $N$ -to-1 マルチプレクサ))の入力に接続されており、マルチプレクサの出力はTRブロックを介してDVMCの  $stop$  線に接続されている。TRブロックは入力された遷移を立ち上がり遷移に変換する回路である。測定時、測定パスが活性化されるように各フリップフロップに適切な値が設定される。クロック線の立ち上がり遷移の到達に同期してパスの入力端に接続されているフリップフロップから遷移が与えられ、その遷移がパスの出力端に伝搬する。出力端に伝搬した遷移は、SSG、TRブロックを経由して  $stop$  に到達する。

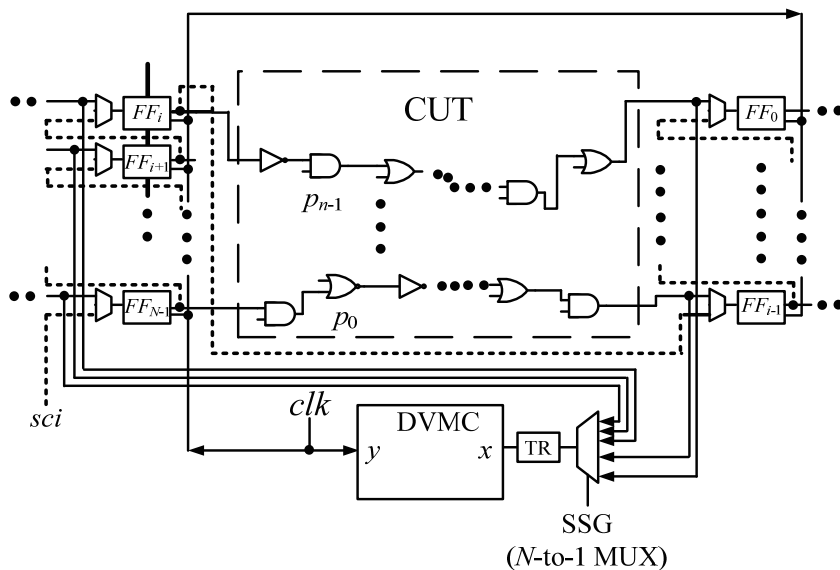


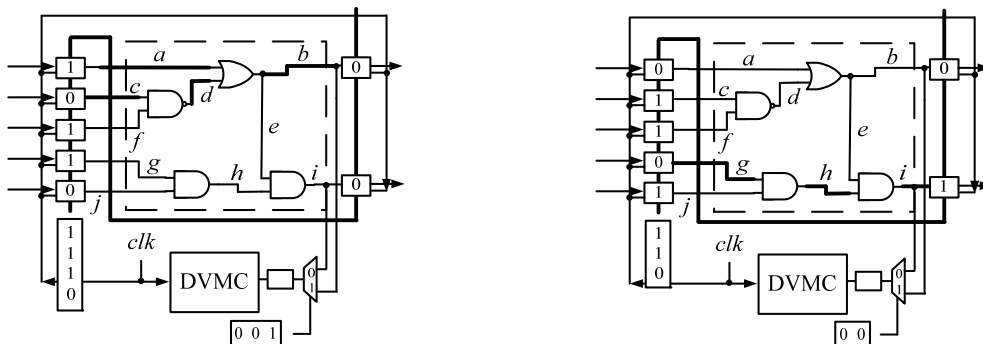
図2. 組み込み遅延測定回路を用いた遅延時間測定系。

### 3 時分割遅延時間測定

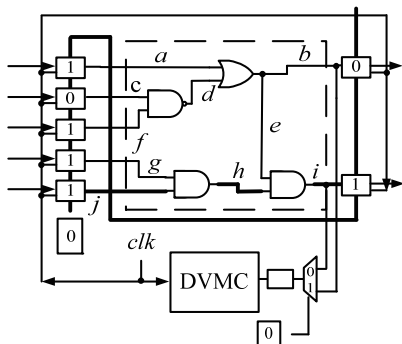
ここでは、1 つ目の提案である組み込み遅延測定回路を用いた時分割遅延時間測定について述べる。3.1. において時分割遅延時間測定手順を示す。3.2. において提案時分割遅延時間測定のための入力系列の構成法を示す。

#### 3.1 時分割遅延時間測定手順

図2に示される測定系はスキャンベースであるため、ある測定とその次の測定の間にはスキャンシフト操作を要する。時分割遅延時間測定では、その次の測定までのスキャンシフト操作に必要なスキャンシフトクロック数を極力減らす事により測定時間を短縮する。図3に時分割遅延測定の例を示す。時分割遅延測定には、スキャン入力系列  $sd$  以外に測定パスを選択するための測定パス選択入力系列  $s_{ep}$ 、及び測定間のシフトクロック数を制御するためのシフト量制御入力系列  $c_s$  が必要となる。この例では、 $sd = 00101101110$ 、 $c_s = (111) (000) (001)$ 、 $s_{ep} = 100$  である。測定対象パスは、 $\{cdb, ghi, ab, jhi\}$  である。まず7スキャンクロック動作させた後、SSGがエンドポイント  $FF_0$  を選択するようにする(a)。この時  $c_{s0} = 111$ 、 $s_{ep0} = 1$  となる。この後、LOS操作により  $ab$ 、 $cdb$  を活性化して測定を行う。次にエンドポイント  $FF_1$  を選択する (b)。この時  $c_{s1} = 000$ 、 $s_{ep1} = 0$  となる。この後 LOS操作により  $ghi$  を活性化して測定を行う。最後にさらに1スキャンクロック動作させた後、SSGがエンドポイント  $FF_1$  を選択するようにする(c)。この時  $c_{s2} = 001$ 、 $s_{ep2} = 0$  となる。この後 LOS操作により  $jhi$  を活性化して測定を行う。



(a) LOS after 7 clock shift-in operation and selection of line  $b$ . (b) LOS after 0 clock shift-in operation and selection of line  $i$ .



(c) LOS after 1 clock shift-in operation and selection of line  $i$ .

図3. 時分割遅延時間測定.

$n_{sd}$  をスキャン長とすると、時分割測定手順は以下のようなになる。

- Step 1  $i$  と  $j$  を共に 0 に初期化.
- Step 2  $c_{si}$  をセット.  $sd$  を  $c_{si}$  クロックだけシフトさせる.  $j \leftarrow j + c_{si}$  に更新.
- Step 3  $s_{epi}$  を SSG にセット.
- Step 4 LOS テスト操作を行い測定開始.  $j \leftarrow j + 1$  に更新.
- Step 5 測定終了後測定結果を DVMC から取り出す.  $i \leftarrow i + 1$  に更新.
- Step 6  $j$  が  $n_{sd}$  と同じなら終了. そうでなければ Step 2 へ.

### 3.2 入力系列構成法

提案する時分割測定法は、スキャン入力系列  $sd$ 、測定パス選択入力系列  $s_{ep}$ 、シフト量制御入力系列  $c_s$  が必要となる。図4に入力系列構成法を示す。まず初めに測定対象パスのそれぞれを決定的に活性化するテストベクトルを用意する。この例の場合、 $tv_0$ 、 $tv_1$ 、 $tv_2$ の3つのベクトルを用意する。 $tv_0$ は、 $ab$ 、 $cdb$ を、 $tv_1$ は、 $ghi$ を、 $tv_2$ は  $jhi$ をそれぞれ活性化する。テストベクトル長は8ビットである。まず  $tv_0$ を8bitの仮想的バッファ  $qv = q_0q_1q_2q_3q_4q_5q_6q_7$ にコピーする。 $q_0$ - $q_6$ はそれぞれ  $FF_0$ - $FF_6$ に対応する (a)。ここでバッファ  $qv$ の値を添え字の低い方に1ビットシフトする。空いた  $q_7$ には  $X$ を設定する。その後  $qv$ と  $tv_1$ 、 $tv_2$ がマージ可能かどうかチェックを行う (b)。マージ可能なものを一つ選びマージを行う。この場合  $tv_1$ のみマージ可能であるので  $qv$ と  $tv_1$ のマージを行い、 $qv$ の値を更新する。更新された  $qv$ は、さらに1ビットシフトを行い、 $q_7$ に  $X$ を設定する。そして残った  $tv_2$ と  $qv$ がマージ可能かどうかチェックを行う (c)。この場合  $tv_2$ は  $qv$ とマージを行う事ができない。マージができない場合さらに1ビットシフト及び  $q_7$ に  $X$ を設定する操作を行う。そして再度  $tv_2$ と  $qv$ がマージ可能かどうかチェックを行う。この場合は、 $qv$ と  $tv_2$ のマージが可能であるため、2つのベクトルのマージを行い、 $qv$ を更新する。すべてのベクトルのマージが完了したら処理を終了する (d)。 $qv$ の値の履歴が  $sd$ に相当する。構成した  $sd$ のすべての  $X$ ビットを0に設定することにより図3のスキャン入力系列となる。(e)に示すように  $tv_0$ 、 $tv_1$ 、 $tv_2$ はこの順で  $sd$ に組み込まれている。まず  $tv_0$ の最初の7ビットを各フリップフロップに設定するために  $sd$ に対し7ビットだけシフト入力動作を行う。故に  $c_{s0}=7$ となる。続いて  $tv_1$ の最初の7ビットを設定するためには、1ビットさらにシフト入力動作を行う必要があるが測定時のLOS操作を考慮するとシフト入力動作は必要ない。故に  $c_{s1}=0$ となる。続いて  $tv_2$ の最初の7ビットを設定するためには、2ビットだけさらにシフト入力動作を行う必要があるが測定時のLOS操作を考慮するとシフト入力動作は1ビットとなる。故に  $c_{s2}=1$ となる。故に  $c_s=(111)(000)(001)$ となる。 $tv_0$ で測定される  $ab$ 、 $cdb$ のエンドポイントは  $FF_0$ である。また  $tv_1$ 、 $tv_2$ で測定されるパス  $ghi$ 、 $jhi$ のエンドポイントは共に  $FF_1$ であるので、 $s_{ep}=100$ となる。回路中の微小遅延故障検出のために、すべての活性化可能故障を抽出する。そしてそれぞれの故障に対して決定論的ATPGをかけて、テストベクトルを生成する。注意したいのは、通常テスト生成と異なり各1個の故障に対して1つのテストベクトルを生成する点である。入力系列構成の手順を記述するために、以下の2つの関数を定義する。

$push(a, b)$ : 配列  $a$ に後ろより要素  $b$ を追加する。

$SSG(tv)$ : テストベクトル  $tv$ を用いた測定の  $s_{ep}$ を返す。

$merge(a, b)$ : ベクトル  $a$ とベクトル  $b$ のマージにより生成されるベクトルを返す。

この時入力系列生成スキームは以下のようになる。

- Step 1 活性化可能な遷移故障  $TF = \{tf_0, tf_1, \dots, tf_{n-1}\}$ を得る。
- Step 2  $tf_i$ を決定的に活性化して故障数と同数のテストセット  $TV = \{tv_0, tv_1, \dots, tv_{n-1}\}$ を得る。
- Step 3  $i \leftarrow 0$ ,  $c_s \leftarrow \emptyset$ ,  $s_{ep} \leftarrow \emptyset$ ,  $qv \leftarrow tv_0$ ,  $n_{sft} \leftarrow N$ 。
- Step 4  $push(c_s, n_{sft})$ .  $push(s_{ep}, SSG(tv_i))$ .  
 $TV \leftarrow TV - \{tv_i\}$ .  
 $n_{sft} \leftarrow n_{sft} - 1$ .
- Step 5  $TV = \emptyset$  なら終了。  $TV \neq \emptyset$  なら Step 6 へ。
- Step 6  $push(sd, qv_0)$  .1ビット  $qv$ をシフト。  $qv_N \leftarrow X$ .  $n_{sft}$ をインクリメント。
- Step 7  $tv_j$ と  $qv$ がマージ可能である  $j$ が存在するならば、Step 8 へ。 そうでなければ Step 6 へ。
- Step 8  $qv \leftarrow merge(qv, tv_j)$ ,  $i \leftarrow j$ . Step 4 へ。

### 4 時分割遅延時間測定適用時の測定データ量削減法

ここでは、2つ目の提案である時分割遅延時間測定適用時の測定データ量削減法について説明する。この方法では、測定パス選択入力系列  $s_{ep}$ 、シフト量制御入力系列  $c_s$  双方のデータ量を削減することにより全体のデータ量の削減を行う。4.1においてシフト量制御入力系列のデータ量削減法について述べる。4.2において測定パス選択入力系列のデータ量削減法について述べる。4.3においてデータ量削減を考慮した入力系列構成法を示す。

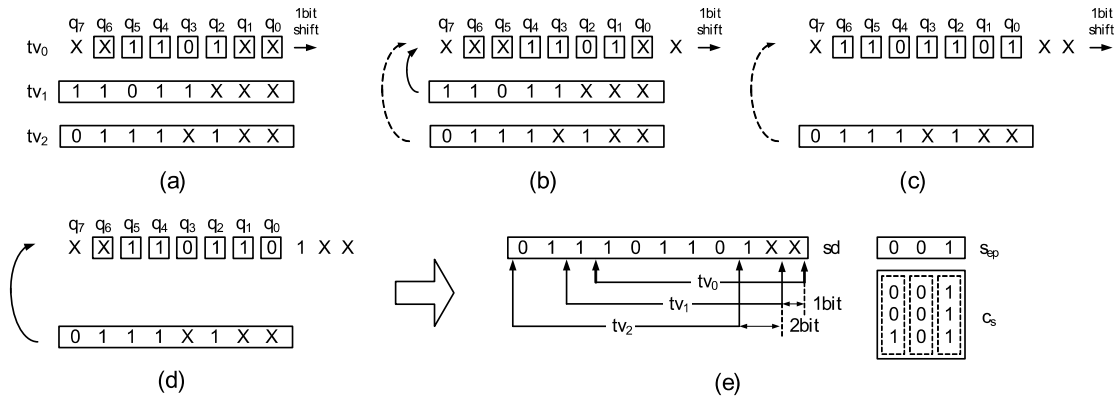


図4. 入力系列構成手順.

#### 4.1 シフト量制御入力系列データ量削減法

回路中のフリップフロップ数が  $N$  の時、LOS 操作を用いて測定パスを活性化することを考慮すると測定間のシフトクロック数の最大値は  $N-1$  となる。故に  $0-N-1$  すべてのシフト量を表現するためにはビット幅  $\log_2 N$  (ビット) のシフト量制御データを付加する必要がある。(図の例ではフリップフロップ数7であるのでビット幅は3ビットである。) しかしながら、 $0-7$  のシフトクロック数のうち実際に用いるシフト数は、7クロック、0クロック、1クロックと高々3通りである。故にデータをテスター上で  $(00) \rightarrow 7$  クロック、 $(01) \rightarrow 0$  クロック、 $(10) \rightarrow 1$  クロックと符号化すればデータ量を3ビットから2ビットへ削減することができる(図5)。故に全体で9bitから6bitへ削減することができる。このようにシフト量制御入力系列の削減が可能となる。

#### 4.2 測定パス選択入力系列データ量削減法

また従来法では、測定パス毎に幅  $\log_2 N$  (ビット) の測定パス選択データを付加していた。しかしながら測定パス選択データの種別は高々測定パスの終端の個数と同じである。そこでできるだけ同じエンドポイントを有するパスを連続して測定するようにすることにより測定パス選択データのビット幅とデータ数を削減する。図6を用いて制御入力系列の削減を説明する。この場合  $s_{ep0}=001$ ,  $s_{ep1}=000$ ,  $s_{ep2}=000$  となる。この時測定するパスのエンドポイントが  $FF_0$ ,  $FF_1$  の2通りのみなので、0, 1と符号化する事によりデータのビット幅を3ビットから1ビットとなる。この時切り替えのタイミングの同期を取るための同期ビット  $snc$  を付加する。同期ビットは、測定パス選択データの切り替えのタイミングの同期をとるためのビットである。 $i=0$  若しくは  $s_{epi} \neq s_{epi-1}$  の時  $snc_i = 1$  とする。また  $s_{epi} = s_{epi-1}$  の時  $snc_i = 0$  とし各  $s_{epi}$  に1ビットのデータを付加していく。 $snc_i = 0$  の時、パス選択データは1つ前のものと同じであるので不要となる。このように制御データ量の削減を行う。この例の場合従来法のパス選択入力系列が9ビットであったものが提案法適用により5ビットまで削減される。測定パス数を  $N_p$ 、スキャン入力系列のデータ量を  $N_{sd}$ 、出現するシフト量制御データが  $N_{cs}$  通り、測定パスのエンドポイントの総数が  $N_{ep}$  の時、入力系列の総データ量は、 $N_{sd} + N_p \log_2 N_{cs} + N_{ep} \log_2 N_{ep} + N_p$  となる。 $i$  番目のシフト量制御データを  $c_{si}$ 、シフト量制御データの復号データ(実際のシフト量)を  $dec(c_{si})$  とする。パス選択データの同期ビット列を  $snc$  とし  $i$  番目の同期ビット列を  $snc_i$  とする時、時分割測定手順は以下のようになる。

- Step 1  $i \leftarrow 0, j \leftarrow 0$ .
- Step 2  $c_{si}$  をスキャンクロックコントローラにセットする.  $j \leftarrow j + dec(c_{si})$ .
- Step 3  $snc_i$  が1の時  $s_{epi}$  をSSGに設定.
- Step 4 LOSテスト操作により測定パスを活性化して測定を実行.
- Step 5 DVMCから測定結果を取り出す.  $i \leftarrow i + 1$ .
- Step 6  $j$  が  $n_{sd}$  と等しい場合終了. そうでない場合 Step 2へ.

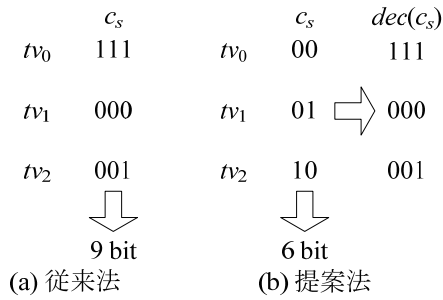


図 5. シフト量制御入力系列データ量削減.

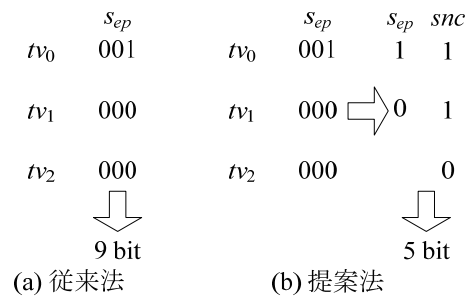


図 6. 測定パス選択入力系列データ量削減.

### 4.3 時測定パス選択入力系列データ量削減法

時分割測定のためのスキャン入力系列を構成する時、3 で示した手法では候補となるすべてのテストベクトルのうち最もシフト操作が短いものを選択してそれに基づいて入力系列を生成した。提案法では、シフト系列があるスレッシュホールドより小さくかつ同一のエンドポイントとなるパスを活性化するベクトルの中で最もシフト操作の少ないものを選択してそれに基づいて入力系列を選択する。これにより、同一のエンドポイントを有するパスを、シフト量が与えられたスレッシュホールドの制限下で、連続して測定することが可能となる。スレッシュホールドの値を大きくすることにより連続して測定する同一エンドポイントのパス数は増加する可能性が高くなるが、その結果逆に生成されるスキャン入力系列の長さが増加する可能性が高くなる。4.1, 4.2 でそれぞれ示したシフト量制御入力系列データ量削減、測定パス選択入力系列データ量削減を考慮に入れたテスト入力系列構成スキームを以下に示す。

- Step 1 活性化可能な遷移故障  $TF = \{tf_0, tf_1, \dots, tf_{n-1}\}$  を得る。
- Step 2  $tf_i$  を決定的に活性化して故障数と同数のテストセット  $TV = \{tv_0, tv_1, \dots, tv_{n-1}\}$  を得る。
- Step 3  $i \leftarrow 0, c_s \leftarrow \emptyset, s_{ep} \leftarrow \emptyset, qv \leftarrow tv_0, n_{sft} \leftarrow N$ .
- Step 4  $push(c_s, n_{sft}), push(s_{ep}, SSG(tv_i))$ .  
 $TV \leftarrow TV - \{tv_i\}$ .  
 $n_{sft} \leftarrow -1$ .  
 $mode \leftarrow 0$ .
- Step 5  $TV = \emptyset$  なら終了.  $TV \neq \emptyset$  なら Step 6 へ.
- Step 6  $n_{sft} = th$  かつ  $mode = 0$  なら  $mode \leftarrow 1, n_{sft} \leftarrow -1$ .  
 $push(sd, qv_0)$ . 1 ビット  $qv$  をシフト.  $qv_N \leftarrow X$ .  $n_{sft}$  をインクリメント.
- Step 7  $mode = 0$  の場合,  $tv_i$  の測定パス  $p_i$  と  $tv_j$  の測定パス  $p_j$  のエンドポイントが同じ, かつ  $tv_j$  と  $qv$  が  $n_{sft} < th$  でマージ可能である  $j$  が存在するならば, Step 8 へ. そうでなければ Step 6 へ.  
 $mode = 1$  の場合,  $tv_j$  と  $qv$  がマージ可能である  $j$  が存在するならば, Step 8 へ. そうでなければ Step 6 へ.
- Step 8  $qv \leftarrow merge(qv, tv_j), i \leftarrow j$ . Step 4 へ.
- Step 9 提案データ量削減法を  $c_s, s_{ep}$  へ適用して終了.

## 5 評価

ここでは提案する 2 つの手法の評価を行う。5.1 において時分割遅延測定法の評価を行う。続いて 5.2 において時分割遅延測定適用時の測定データ量の削減法の評価を行う。

### 5.1 時分割遅延時間測定法

ここでは時分割測定法の測定時間、データ量、面積について先行研究と比較評価を行う。評価回路は IWS2005 ベンチマーク回路を用いる。提案法は、測定パスの入力にはスキャンフリップフロップが挿入されている必要がある。そこで本評価では評価回路の外部入力にもスキャンフリップフロップを挿入して、各外部入力値を設定できるものとする。DVMC は文献[5] のものを用いる。この DVMC はリングオシレータベース

の測定回路であり、測定データは14ビットである。このため測定結果の取り出しに14クロックを要する。テスト生成には、シノプシス社のTetra Maxを用いる。検出対象故障は遷移故障である。故障リスト中のすべてのLOS活性化可能故障をLOS活性化規範に基づき逐次活性化を行い、テストベクトル集合を生成する。求めたテストベクトル集合に対して提案時分割遅延時間測定法の入力系列生成スキームを用いて入力系列を構成する。提案法は、文献[5]の手法と比較を行う。文献[5]の手法は提案法と同様のテストベクトル集合を用いる。また参考データとして通常のスタンダードスキャンテストとも比較を行う。用いるテストベクトル集合は、LOC活性化規範に基づく最小テストベクトル集合とする。評価に用いたテストセットの特性を表1に示す。PI, PO, FF,  $N_g$  は、それぞれ各評価回路の外部入力線数, 外部出力線数, フリップフロップ数, ゲート数を示す。 $N_{trn}$  は遷移遅延故障数を示す。LOCはLOCテストセットの特性, LOSはLOSテストセットの特性を示す。 $N_{Dtrn}$  は検出された故障数を示す。 $N_v$  は、テストベクトル数を示す。LOSのテストベクトル数は、 $N_{Dtrn}$ の故障数と同数となる。 $Cov.$  は故障検出率を示す。まずテスト実行時間の評価を行う。表2にテスト実行時間を示す。2列目のデータは、スタンダードスキャンを用いたテストに要する時間 $T_{STD}$ を示す。3, 4, 5列目は、それぞれ $T_{STD}$ に対する文献[5]の手法のテスト実行時間の割合,  $T_{STD}$ に対する提案法のテスト実行時間の割合, 文献[5]の手法のテスト実行時間に対する提案法のテスト実行時間の割合を示す。フリップフロップ数と同数のスキャンシフト操作で1本のパスを測定する文献[5]の手法は、回路規模が大きくなるにつれて、測定パス数が多くなり測定の効率が悪くなる。このため $T_{STD}$ に対する測定時間の割合が大きくなり、平均値は4,488.8(%)となる。一方提案法は、テスト実行時間が回路規模への依存は少なく、平均値は、90.4(%)となる。平均で提案法のテスト実行時間は、文献[5]のテスト実行時間の3.0(%)となる。表3にテスト実行時間におけるデータシフトに要する時間( $T_{SFT}$ )と測定に要する時間( $T_{MEAS}$ )の内訳を示す。 $T_{PROP}$ は、それらの和である。通常のテストではスキャンシフトに要する時間がテスト時間において多く占めるが、提案測定法では、データシフトに要する時間は少ない。表4にシフトビット数の分布を示す。1番最後の列が平均シフト数で、それ以外の列は各シフト量のシフト数制御データの個数をシフト数制御データの総数で正規化したものである。どの回路においても、ほとんどのスキャンシフト数制御データのシフト量は0であることがわかる。平均値にて約75%のパスの測定に要するシフト量は0となっている。この結果この平均値は2.7となる。データ量の評価結果を表5に示す。2列目のデータは、スタンダードスキャンを用いたテストに要するデータ量 $V_{STD}$ を示す。3, 4, 5列目は、それぞれ $V_{STD}$ に対する文献[5]の手法のデータ量の割合,  $V_{STD}$ に対する提案法のデータ量の割合, 文献[5]の手法のデータ量に対する提案法のデータ量の割合を示す。1回のスキャンインで1本のパスを測定する文献[5]の手法は、回路規模が大きくなるにつれて、 $V_{STD}$ に対する割合が大きくなり、平均値は1,921.9(%)となる。一方提案法は、データ量も回路規模に依存せず、平均値は、47.8(%)となる。平均で提案法の測定実行時間は、文献[5]の測定実行時間の3.6(%)となる。最後に面積オーバーヘッドの評価を行う。提案測定系を各評価回路に対してRohm0.18 $\mu$ mスタンダードセルで実装を行う。論理合成は、シノプシス社のDesign Compilerを用いる。面積オーバーヘッド $O$ を、 $O = (S/S_0 - 1) \times 100.0(\%)$ と定義する( $S$ は評価対象回路の面積,  $S_0$ をスタンダードスキャン実装時の面積)。表6に面積の評価結果を示す。 $S_{STD}$  (mm<sup>2</sup>),  $O$  [5] (%),  $O_{ENH}$  (%),  $O_{PROP}$  (%) は、それぞれスタンダードスキャンを実装した時の面積, 文献[5]の手法の面積オーバーヘッド, エンハンスドスキャンの面積オーバーヘッド, 提案法の面積オーバーヘッドを示す。提案法は、DVMCを要し、かつ外部入力にスキャンフリップフロップを挿入して、外部出力から冗長配線を出してSSGへ伝搬するようにしている。このため回路規模が小さい場合、若しくはFF数と比較して相対的に外部入出力線数が多い時は面積オーバーヘッドが大きくなる。s5378, s9234, wb\_dma, aes\_core, systemcaes, dma では面積オーバーヘッドがエンハンスドスキャンより大きくなる。平均値では、ほぼエンハンスドスキャンと同じ面積オーバーヘッドとなる。

表1. 評価に用いたテストセット.

circ.	PI	PO	FF	$N_g$	$N_{trn}$	LOC			LOS		
						$N_{Dtrn}$	$N_v$	$Cov.$	$N_{Dtrn}$	$N_v$	$Cov.$
s5378	35	49	163	2,779	5,638	5,085	379	90.35	5,499	5,499	97.71
s9324	36	39	145	5,597	4,456	4,270	370	95.96	4,361	4,361	98.00
s13207	31	121	330	7,951	6,420	5,802	335	92.26	6,025	6,025	95.80
s35932	35	320	1,728	16,065	35,750	35,169	145	98.38	32,637	32,637	91.30
s38417	28	106	1,564	22,179	39,472	38,331	1,663	97.11	39,105	39,105	99.08
s38584	12	278	1,172	19,253	39,782	35,832	1,291	90.65	38,619	38,619	97.71
tv80	12	32	359	2,458	23,558	17,797	1,306	75.60	22,633	22,633	96.14
wb_dma	215	215	528	1,046	16,574	16,306	535	98.48	16,434	16,434	99.25
aes_core	257	129	530	10,333	87,978	87,670	1,244	99.65	85,082	85,082	96.71

systemcaes	258	129	670	2,329	29,554	28,172	728	95.39	25,369	25,369	85.90
mem_ctrl	112	152	1,071	2,972	34,192	32,555	2,060	95.37	33,675	33,675	98.65
usb_funct	125	121	1,738	4,429	53,372	52,557	3,162	98.62	52,787	52,787	99.06
dma	682	262	1,803	4,741	64,866	63,055	5,468	97.80	62,891	62,891	97.54
pci_bridge32	158	207	3,314	5,473	76,590	74,761	3,473	97.83	75,909	75,909	99.33

表2. テスト実行時間.

circ.	$T_{STD}(LOC)$ ( $10^6$ clocks)	$T[5]/$ $T_{STD}$ (%)	$T_{PROP}/$ $T_{STD}$ (%)	$T_{PROP}/$ $T[5]$ (%)
s5378	0.06	1868.1	164.4	8.8
s9324	0.05	1567.4	154.8	9.9
s13207	0.11	2030.8	87.9	4.3
s35932	0.25	22974.5	213.0	0.9
s38417	2.61	2411.6	29.8	1.2
s38584	1.52	3052.7	41.5	1.2
tv80	0.47	1851.6	100.3	5.4
wb_dma	0.28	4385.1	99.3	2.3
aes_core	0.66	10,302.3	215.0	2.1
systemcaes	0.49	4883.4	91.6	1.9
mem_ctrl	2.21	1824.3	30.0	1.6
usb_funct	5.50	1801.3	16.0	0.9
dma	9.87	1592.7	10.9	0.7
pci_bridge32	11.52	2297.7	11.4	0.5
Ave.	-	4,488.8	90.4	3.0

表3. テスト実行時間の内訳 ( $10^3$ clocks).

circ.	$T_{SFT}$	$T_{MEAS}$	$T_{PROP}$
s5378	20.6	82.5	103.0
s9324	19.0	65.4	84.4
s13207	7.7	90.4	98.0
s35932	48.5	489.6	538.1
s38417	191.2	586.6	777.8
s38584	49.8	579.3	629.1
tv80	133.6	339.5	473.1
wb_dma	35.5	246.5	282.0
aes_core	147.8	1276.2	1424.1
systemcaes	68.4	380.5	448.9
mem_ctrl	157.7	505.1	662.8
usb_funct	90.8	791.8	882.6
Dma	129.2	943.4	1,072.6
pci_bridge32	175.6	1,138.6	1,314.3

表4. シフトビット数の分布.

circ.	Shift-bit num.												
	0	1	2	3	4-5	6-9	10-19	20-49	50-99	100-199	200-399	400-799	Ave
s5378	0.62	0.06	0.04	0.04	0.07	0.07	0.06	0.03	$3.0e^{-3}$	$1.9e^{-3}$	-	-	3.7
s9324	0.64	0.04	0.03	0.03	0.04	0.05	0.08	0.07	0.01	-	-	-	4.7
s13207	0.92	0.02	0.01	0.01	0.01	0.01	0.01	0.01	$0.2e^{-3}$	-	-	-	0.6
s35932	0.61	0.19	0.09	0.05	0.04	0.01	$0.2e^{-3}$	$0.1e^{-3}$	-	-	-	-	0.8
s38417	0.72	0.02	0.02	0.02	0.03	0.04	0.07	0.07	0.02	$2.9e^{-3}$	-	-	6.1
s38584	0.80	0.10	0.05	0.02	0.02	0.01	$4.8e^{-3}$	$3.7e^{-3}$	$0.5e^{-3}$	-	-	-	0.6
tv80	0.45	0.06	0.05	0.03	0.10	0.14	0.12	0.06	0.01	$0.5e^{-3}$	-	-	6.2
wb_dma	0.86	0.02	0.01	0.01	0.02	0.02	0.03	0.02	$3.9e^{-3}$	$0.6e^{-3}$	$0.1e^{-3}$	-	1.7
aes_core	0.95	0.01	$4.4e^{-3}$	$3.0e^{-3}$	$4.6e^{-3}$	0.01	0.01	0.01	$3.4e^{-3}$	-	-	-	0.9
systemcaes	0.64	0.05	0.04	0.04	0.07	0.08	0.06	0.02	$2.0e^{-3}$	$2.0e^{-3}$	-	-	2.9
mem_ctrl	0.75	0.02	0.02	0.01	0.02	0.04	0.06	0.07	0.01	$1.3e^{-3}$	$0.3e^{-3}$	-	5
usb_funct	0.91	0.01	0.01	0.01	0.01	0.02	0.02	0.01	$0.9e^{-3}$	$0.3e^{-3}$	$0.2e^{-4}$	$0.2e^{-4}$	1
dma	0.78	0.05	0.03	0.03	0.04	0.04	0.03	0.01	$0.9e^{-3}$	$0.3e^{-3}$	$0.1e^{-3}$	-	1.5



pci_bridge32	0.85	0.02	0.02	0.01	0.02	0.03	0.03	0.02	0.01	$0.9e^{-3}$	-	-	1.9
Ave.	0.75	0.05	0.03	$2.2e^{-3}$	0.04	0.04	0.04	0.03	0.01	$0.9e^{-3}$	$0.1e^{-3}$	$0.2e^{-4}$	2.7

表 5. データ量.

circ.	$V_{STD}$ ( $10^3$ bits)	$V[5]/$ $V_{STD}$ (%)	$V_{PROP}/$ $V_{STD}$ (%)	$V_{PROP}/$ $V[16]$ (%)
s5378	155.4	759.1	71.5	9.4
s9324	135.1	638.0	67.5	10.6
s13207	272.0	841.3	44.4	5.3
s35932	552.6	10,546.9	142.9	1.4
s38417	5,424.7	1,163.9	19.8	1.7
s38584	3,400.5	1,370.1	26.9	2.0
tv80	995.2	885.2	54.8	6.2
wb_dma	795.0	1,578.5	47.0	3.0
aes_core	1,798.8	3,817.6	103.4	2.7
systemcaes	1,257.3	1,913.9	46.8	2.4
mem_ctrl	4,956.4	819.0	16.4	2.3
usb_funct	11,769.0	845.7	10.8	1.3
dma	24,879.4	634.4	6.7	1.1
pci_bridge32	24,286.7	1,092.9	8.4	0.8
Ave	-	1921.9	47.8	3.6

表 6. 面積評価結果.

circ.	PI	PO	FF	$S_{STD}$ ( $mm^2$ )	$O[5]$ (%)	$O_{ENH}$ (%)	$O_{PROP}$ (%)
s5378	35	49	163	0.04	17.0	18.7	29.7
s9234	36	39	145	0.03	7.3	24.0	34.6
s13207	31	121	330	0.06	7.4	25.5	22.1
s35932	35	320	1,728	0.36	6.7	23.0	8.8
s38417	28	106	1,564	0.33	4.3	22.7	10.2
s38584	12	278	1,172	0.27	4.0	20.7	9.0
tv80	12	32	359	0.11	-	15.5	9.2
wb_dma	215	215	528	0.10	-	24.7	42.3
aes_core	257	129	530	0.30	-	8.6	16.6
systemcaes	257	129	670	0.17	-	18.7	30.0
mem_ctrl	112	152	1,071	0.22	-	23.2	16.5
usb_funct	125	121	1,738	0.36	-	23.5	14.4
dma	682	262	1,803	0.41	-	21.3	30.6
pci_bridge32	158	207	3,314	0.63	-	25.5	13.0
Ave	-	-	-	-	-	21.1	20.5

## 5.2 時分割遅延時間測定法適用時の測定データ量の削減法

入力系列のデータ量及び測定時間について、提案時分割測定法と従来時分割測定法の比較評価を行う。評価は、ISCAS89 ベンチマーク回路を用いて行った。提案法は、測定パスの入力にはスキャンフリップフロップが挿入されている必要がある。そこで本評価では評価回路の外部入力にもスキャンフリップフロップを挿入して、各外部入力値を設定できるものとする。DVMC は文献[5] のものを用いる。この DVMC はリングオシレータベースの測定回路であり、測定データは 14 ビットである。このため測定結果の取り出しに 14 クロックを要する。各評価回路に対して C++ で実装した ATPG を用いて遷移故障検出テストパターン生成を行った。遷移故障検出パターンは、単一経路活性化パターンのもとのロバスト活性化パターンのもとの 2 通りのものを生成した。生成したそれぞれのテストパターンに対してデータ削減を考慮した時分割遅延測定法とデータ削減を考慮しない時分割遅延測定法の入力系列生成スキームをそれぞれ適用した。提案法の連続測定のスレッシュホールド  $th$  は 100 とした。表 7 に入力系列のデータ量の評価結果を示す。circ. は評価回路を示す。m の

$p$ はデータ削減を考慮したもの、 $c$ はデータ削減を考慮していないものの結果を示す。 $sns$ は用いたテストベクトルのパスの活性化規範を示し、 $robust$ はロバストパス活性化、 $single$ は単一経路活性化を表す。 $L_{sft}$ は測定中用いるシフト量の総数を示す。 $V_{sd}$ はスキャン入力系列のデータ量を示す。 $V_{cs}$ は、シフト量制御入力系列のデータ量を示す。 $R_{cs}$ は、シフト量制御入力系列のデータ量削減率を示す。これは圧縮を考慮した手法のシフト量制御入力系列データ量の圧縮を考慮しない手法のシフト量制御入力系列データ量に対する割合(%)で定義される。 $V_{sep}$ は、測定パス選択入力系列のデータ量である。 $R_{sep}$ は、測定パス選択入力系列のデータ削減率を示す。これは圧縮を考慮した手法の測定パス選択入力系列データ量の圧縮を考慮しない手法の測定パス選択系列データ量に対する割合(%)で定義される。 $V$ は入力系列の総データ量を示し、 $R_V$ は提案法の総入力系列のデータ量削減率を示す。これは圧縮を考慮しない手法の測定パス選択入力系列データ量に対する割合(%)で定義される。 $Ave$ は平均値を示し、 $robust$ はロバスト活性化パターンの結果の平均値、 $single$ は単一経路活性化パターンの結果の平均値を示す。 $all$ はすべての結果の平均値を示す。ロバストパス活性化パターンと単一経路活性化パターンの結果を比べると、データ量削減率 $R_V$ は、ロバストパス活性化パターンが85.4%であるのに対し、単一経路活性化パターンが79.0%である。またすべての評価回路においてロバストパス活性化パターンの $R_V$ が単一経路活性化パターンのもより大きい傾向がある。入力系列の総データ量は、スキャン入力系列と制御入力系列のデータ量の総和であるが、制御入力系列のデータ量の削減処理は、スキャン入力系列のデータ量に大きく影響する。このためデータ削減率 $R_V$ は、回路毎に傾向が異なり、s35932の単一経路活性化パターンのように59.7%まで削減されるものもあれば逆にs9234のロバスト活性化パターンのように103.1%と逆にデータ量が従来法より増加してしまうものもある。シフト量制御入力系列のデータ削減率の平均値が72.6%なのに対してパス選択系列のデータ削減率は32.2%となる。これは全体のデータ削減においてパス選択系列のデータ削減の方が貢献していることを意味している。表8は測定時間の評価結果である。 $T$ は測定時間の評価結果を示し、 $R_T$ は(提案法測定時間/従来法測定時間-1)×100.0で計算される。評価結果より、増加率は、ロバストテストパターンの場合28.0%、単一経路活性化パターンの場合24.8%、全体で26.4%となる。単一経路活性化パターンのデータ削減率はロバスト活性化パターンのもより小さい。また単一経路活性化パターンの測定時間の増加率はロバスト活性化パターンのもより小さい。よって本評価結果より提案手法は単一経路活性化パターンへの適用により効果的であるといえる。

## 6 まとめ

本研究では、VLSIの微小遅延検出テストのための組み込み遅延測定回路を用いたオンチップ遅延測定法に関する研究を行った。そのために2つの取り組みを行った。まず初めに、その測定時間の削減の取り組みを行った。提案する時分割遅延時間測定により、平均で測定時間を従来法の3.0%まで削減できることを示した。次に、測定に要する入力系列のデータ量の削減の取り組みを行った。提案した入力系列データ量削減法の適用により平均で17.8%削減されることを確認した。時分割遅延時間測定は回路内の経路の網羅的タイミング解析を短時間で行う事ができ、微小遅延故障検出だけでなく、タイミング不良の高信頼化において有効な技術となる可能性がある。しかしながらこの手法の実装には、追加面積が必要となり、追加面積の通常動作への影響の低減の検討がまだ十分になされていない。これは今後の課題として早急に検討を行う。

表7. データ量.

circuit	m.	Sns	$N_p$	$L_{sft}$	$V_{sd}$	$V_{cs}$	$R_{cs}$	$V_{sep}$	$R_{sep}$	$V$	$R_V$
s5378	$p$	Robust	1,378	83	18,776	9,646	87.5	3,074	27.9	31,496	94.3
	$c$	Robust		-	11,346	11,024		11,024		33,394	
	$p$	Single	847	59	6,647	5,082	75.0	2,279	33.6	14,008	81.4
	$c$	Single		-	3,654	6,776		6,776		17,206	
s9234	$p$	Robust	1,739	95	27,276	12,173	87.5	4,123	29.6	43,572	103.1
	$c$	Robust		-	14,420	13,912		13,912		42,244	
	$p$	Single	1,280	82	13,773	10,240	100.0	3,272	32.0	27,285	99.6
	$c$	Single		-	6,907	10,240		10,240		27,387	
s13207	$p$	Robust	3,206	87	19,483	22,442	70.0	11,236	35.0	53,161	74.4
	$c$	Robust		-	7,311	32,060		32,060		71,431	
	$p$	Single	2,370	64	11,212	14,220	60.0	9,960	42.0	35,392	68.1
	$c$	Single		-	4,597	23,700		23,700		51,997	

s35932	<i>p</i>	<i>Robust</i>	12,005	98	69,640	84,035	63.6	39,395	29.8	193,070	66.7
	<i>c</i>	<i>Robust</i>		-	25,428	132,055		132,055		289,538	
	<i>p</i>	<i>Single</i>	8,816	86	30,906	61,712	63.6	31,905	32.9	124,523	59.7
	<i>c</i>	<i>Single</i>		-	14,749	96,976		96,976		208,701	
s38417	<i>p</i>	<i>Robust</i>	14,751	147	252,105	118,008	72.7	61,710	38.0	431,823	97.6
	<i>c</i>	<i>Robust</i>		-	118,087	162,261		162,261		442,609	
	<i>p</i>	<i>Single</i>	12,339	112	174,308	86,373	63.6	45,405	33.5	306,086	91.8
	<i>c</i>	<i>Single</i>		-	62,033	135,729		135,729		333,491	
s38584	<i>p</i>	<i>Robust</i>	16,090	112	159,466	112,630	63.6	50,047	28.3	322,143	76.5
	<i>c</i>	<i>Robust</i>		-	67,140	176,990		176,990		421,120	
	<i>p</i>	<i>Single</i>	14,044	111	123,870	98,308	63.6	36,099	23.4	258,277	73.2
	<i>c</i>	<i>Single</i>		-	43,905	154,484		154,484		352,873	
Ave.	-	<i>Robust</i>	-	-	-	-	74.2	-	31.5	-	85.4
	-	<i>Single</i>	-	-	-	-	71.0	-	32.9	-	79.0
	-	<i>All</i>	-	-	-	-	72.6	-	32.2	-	82.2

表 8. 測定時間.

<i>Circuit</i>	<i>m.</i>	<i>sns</i>	<i>T</i>	<i>R<sub>T</sub></i>
s5378	<i>p</i>	<i>robust</i>	39,446	23.2
	<i>c</i>	<i>single</i>	32,016	
	<i>p</i>	<i>robust</i>	19,352	18.3
	<i>c</i>	<i>single</i>	16,359	
s9234	<i>p</i>	<i>robust</i>	53,361	31.7
	<i>c</i>	<i>single</i>	40,505	
	<i>p</i>	<i>robust</i>	32,973	26.3
	<i>c</i>	<i>single</i>	26,107	
s13207	<i>p</i>	<i>robust</i>	67,573	22.0
	<i>c</i>	<i>single</i>	55,401	
	<i>p</i>	<i>robust</i>	46,762	16.5
	<i>c</i>	<i>single</i>	40,147	
s35932	<i>p</i>	<i>robust</i>	249,715	21.5
	<i>c</i>	<i>single</i>	205,503	
	<i>p</i>	<i>robust</i>	163,146	11.0
	<i>c</i>	<i>single</i>	146,989	
s38417	<i>p</i>	<i>robust</i>	473,370	39.5
	<i>c</i>	<i>single</i>	339,352	
	<i>p</i>	<i>robust</i>	359,393	45.4
	<i>c</i>	<i>single</i>	247,118	
s38584	<i>p</i>	<i>robust</i>	400,816	29.9
	<i>c</i>	<i>single</i>	308,490	
	<i>p</i>	<i>robust</i>	334,530	31.4
	<i>c</i>	<i>single</i>	254,565	
Ave.	-	<i>robust</i>	-	28.0
	-	<i>single</i>	-	24.8
	-	<i>all</i>	-	26.4

【参考文献】

- [1] K. Noguchi, K. Nose, T. Ono, and M. Mizuno, "A small-delay defect detection technique for dependable LSIs," Proc. IEEE Symposium on VLSI Circuits, pp. 64–65, Jun. 2008.
- [2] K. Katoh, K. Namba, and H. Ito, "An On-Chip Delay Measurement Technique Using Signature Registers for Small-Delay Defect Detection," IEEE Trans. on VLSI Systems, vol.20, no.5, pp.804–817, 2012.
- [3] R. Datta, A. Sebastine, A. Raghunathan, and J. A. Abraham, "On-chip delay measurement for silicon debug," Proc. Great Lakes Symposium on VLSI (GLSVLSI'04), pp. 145–148, Apr. 2004.

- [4] S. Pei, H. Li, and X. Li, "A low overhead on-chip path delay measurement circuit," Proc. IEEE Asian Test Symposium (ATS'09), pp. 145–150, Nov. 2009.
- [5] K. Katoh, K. Namba, and H. Ito, "A low-area on-chip delay measurement system using embedded delay measurement circuit," Proc. IEEE Asian Test Symposium (ATS'10), pp. 343–348, Dec. 2010.

〈 発 表 資 料 〉

題 名	掲載誌・学会名等	発表年月
クロック信号利用のパス遅延測定による微小遅延テスト	電子情報通信学会技術研究報告、FIIS-11-304、2011	2011. 8
組み込み遅延測定回路を用いた時分割オンチップパス遅延測定のための入力系列データ量削減の1手法	電子情報通信学会技術研究報告、DC2012-10、vol. 112, no. 102、2012	2012. 6