社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2007－ＳＬＤＭ－129／2007－ＥＭＢ－4 (20)
2007／3／16

回路変更を用いたプロトタイプ設計検証における高速化

井上 敬太† 邢 唯頡† 木村 晋二†

†早稲田大学院 情報生産システム研究科 〒808-0135 北九州市若松区ひびきの2-7
E-mail: †inoino5656@fuji.waseda.jp, shinji_kimura@waseda.jp

あらまし 電子機器の普及に伴い，高機能化が進むSoC（System on Chip）開発において，設計期間の60%以上を占める検証期間の短縮が求められている．検証では，シミュレーションが主に用いられているので，シミュレーションの高速化が重要である．一般的には，FPGAなどのハードウェアを用いたエミュレーションによってシミュレーションの高速化を図るが，エミュレーションによる高速化だけでは十分ではない．そこで同期式マイクロパイプライン方式を提案し，高速なクロックで機能検証を行う手法を示す．同期式パイプライン法は，一次元的に処理が行われる場合には有効に適用できる．また，パイプライン各段の組合せ回路の高速化として，組合せ回路の最長経路のfalse path化による手法，とくに0信号伝播と1信号伝播を分けて伝播する手法を示す．

キーワード FPGA，false path，同期式マイクロパイプライン方式，プロトタイピング，エミュレーション

Acceleration of Prototyping Design Verification Using Circuit Modification

Keita INOUE†, Xing WEIJIE† and Shinji KIMURA†

† Graduate School of Information, Production, and Systems, Waseda University
2-7 Hibikino, Wakamatsu-ku, Kitakyusyu-shi, 808-0135 Japan
E-mail: †inoino5656@fuji.waseda.jp, shinji_kimura@waseda.jp

Abstract In recent SoC (System on Chip) design, more then 60% of design period has been spent by the verification, so we need efficient verification method to reduce the verification time. In the verification, functional simulation is mainly applied, and the acceleration of the simulation by using hardware emulation with FPGA is considered effective. The emulation for large circuits, however, is rather slow, and the speed-up is expected for the reduction of the verification time. In this report, we show an accelerator method based on synchronous pipelining and false-path based combinational circuit delay reduction method. The synchronous pipelining is effective to one-dimensional processing circuits. In the false path-based methods, we focus on the 0&1 skip method where we propagate 0-signal and 1-signal separately.

Keyword FPGA, false path, Synchronous micro pipeline, Prototyping, Emulation

1. はじめに

電子機器の普及に伴い，高機能化が進むSoC (System on Chip) の開発においては，設計期間の短縮が重要な問題となっている．統計的なデータによれば，設計期間の60%以上が機能検証に費やされていると言われ，その短縮は非常に重要である．機能検証では，形式的な手法の適用も始まりつつあるが，形式的な手法は適用回路規模が小さいという問題を持つため，機能や論理のシミュレーションが多用される．よって，シミュレーションの高速化が重要な問題となっている．

シミュレーションには，ソフトウェアで行われるものと，ハードウェアで行われるものの2種類がある．ソフトウェアを用いたシミュレーションは，観測点の設定などで自由度は高いが，低速である．一方ハードウェアを用いたエミュレーションは，観測点の設定などの自由度は低いが高速にできるというメリットがある．例えば，実デバイスで100 MHzで動作するチップのソフトウェアシミュレーションの速度は30 Hz程度，エミュレーションの速度は10 MHz程度である．シミュレーションの高速化という観点からは，エミュレーションは有効ではあるが，回路規模の増大とともに，エミュレーション速度が不足するという問題も生じている．そこで本報告では，エミュレーションの高速化について述べる．

エミュレーションは通常回路の書換えが自由にできるFPGA (Field Programmable Gate Array) を用いて

行われるので，FPGA の特性を用いた高速化が有益である．エミュレーション対象の回路の高速化については，順序回路を仮定すると，クロック周波数を増加させることが必要である．このためには，組合せ回路部分の遅延を減らすことが必要であるが，そのためには二つの手法が考えられる．一つは，組合せ回路部を二つ以上の直列なブロックに分割して，それらの間にレジスタを入れることでクロック周波数を上げるものである．ここではそれを同期式マイクロパイプライン方式と呼ぶ．これは非同期で用いられているマイクロパイプライン方式[5],[6] を同期回路に適用したものである．FPGA では，基本素子内に論理素子と同時にレジスタが含まれるので，組合せ部を分けてレジスタを挿入する時に，余分の資源を必要とせず，効果的に適用できる可能性がある．

また，組合せ部分の遅延最適化を行い，クロック周波数を向上させる手法も有効である．これまでにもいくつかの研究が行われているが，ここでは，最長経路を false path 化する手法に着目し[1]，[2]，[3]，最長経路の信号を 0 と 1 の信号伝播に分けてスキップする 0&1 スキップ法を新たに提案する．この手法は，最長経路の信号伝播がなくなるように付加回路を追加するもので，生成された回路の面積はもとの回路に比べて大きくなる．方式の評価の上では，面積増加がどの程度となっているかも重要である．最長経路を false path 化する手法としては，3 重化法[1] やマルチプレクサ法[7]が知られているが， 0&1 スキップ法の方が付加回路を小さくできる可能性がある．実際にいくつかの回路に適用して，高速化手法の評価を行った．

2. 同期式マイクロパイプライン方式

ここでは，FPGAエミュレーションにおける高クロック化を実現するために，組合せ回路を分割し，高速なクロックでパイプライン処理を行う同期式マイクロパイプラインについて述べる．

同期式マイクロパイプラインは，通常の順序回路として設計された回路を自動的にパイプラインにするものである．同期式マイクロパイプラインの概念を図で説明する．まず，フリップフロップ間に遅延時間をDとする組合せ回路(図 1)があったとする．この回路のクロック周期は D である．そこで処理を高速化するために，クロック周期を D よりも短くすることを考える．このためには，組合せ回路の遅延を D よりも小さくしないといけない．

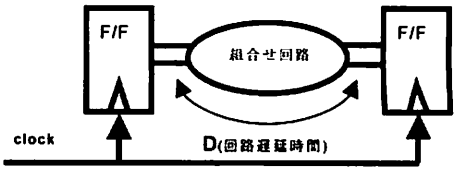

図 1　同期式回路

そこで，この組合せ回路部を二つの直列なブロックに分割して，分割した回路の遅延時間が半分になるようにフリップフロップを挿入する．分割した回路をそれぞれC1，C2とする（図 2）．このとき，C1，C2の回路の遅延時間はそれぞれD/2になるので，クロック周期をD/2にできる．これで処理のスループットは2倍になった．ただし，データの投入もD/2 で行う必要がある．

画像処理における画素データの RGB を YcrCb に変換するような，一方向の直列的な処理では，このようなパイプライン化を自動で行うことが可能である．組合せ回路部はどこで分割しても良く，データの再利用がない限りはデータフォワーディングの必要もない．なお，クロック周波数の最大化と同時に挿入するレジスタをなるべく少なくする最適化を考える必要がある．

さらにC1，C2の遅延をD/2より小さくすることができれば，C1とC2をあわせた遅延を分割前の回路の遅延時間Dよりも小さくでき，同じ回路部分が繰り返し用いられる場合は，削減の効果が繰り返し回数に比例して表れる．組合せ回路部の遅延削減については，次節で述べる．

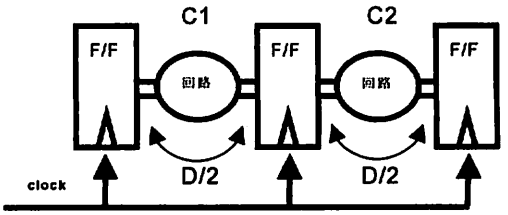

図 2　分割後の回路

同期式マイクロパイプライン方式の効果を調べるため，64bitの加算器を例題として評価を行った．分割前および分割後の回路を設計して，Altera 社のQuartus II Version6.0でマッピングを行い，デバイスはstratix I (EP1S25F1020C7:25,660LE(Logic Elements))を使用した．64ビット加算での評価は通常の加算が242Hz(4.116ns)，192LE，２分割後が，308MHz(3.247ns)，290LEという結果が得られ，速度は22%の改善が見られた．よって，回路の分割によって高速化の効果があると言える．なお，この実験ではレジスタ挿入により，LEの数が増加

したが，FPGAでは，基本素子のLE内に論理素子と同時にレジスタが含まれるので，組合せ部がある程度複雑であれば，レジスタを挿入する時に，余分の資源を必要としない可能性が高い．

3. 最長経路の false path 化による高速化

組合せ回路部を分割し，高速なクロックでパイプライン処理を行う方式を示した．分割後の組合せ回路部のさらなる高速化を実現するために，最長経路の false path 化による高速化について述べる．これまで最長経路の false path 化による高速化には，MUX 法[7]や 3 重化法が知られているが，ここでは，0 の信号伝播と 1 の信号伝播に分けてバイパスする 0&1 スキップを提案する．

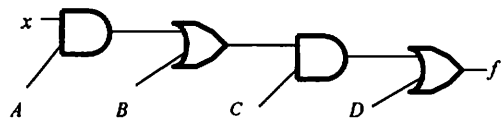

図 3　サンプル回路

高速化手法の説明において図 3の回路を例として示す．図 3の回路の論理関数は以下の式で表せる．

$$f = (x \cdot A + B) \cdot C + D = x \cdot A \cdot C + B \cdot C + D$$

3.1 3 重化法

3 重化法[1](Triple Modular Redundancy, TMR)は，組合せ回路を 3 つコピーして，(1)もとの回路，(2)最長経路の AND を除いた回路，(3)最長経路の OR を除いた回路とし，3 つの回路の出力の voting により出力を得る手法である．最長経路に対し，OR ゲートを削除した回路と AND ゲートを削除した回路を付加する．削除することにより従来の回路が最長経路でなくなり，最長経路が活性化されている場合にはそれぞれ削除した回路の経路を通って先に信号が到達する．

例として図 3の回路に 3 重化法を適用すると図 4 のようになる．入力は X，コピーされる回路の出力を Y，最長経路に対して AND ゲートを削除した回路 A1 の出力を W，最長経路に対して OR を削除した回路 A2 の出力を Z とする．3 重化構造を構成する時には，クリティカルパスを構成している入力およびゲートを除かないように構成しなといけない．つまり，回路の機能に影響しないようにワイヤやゲートを取り外さないといけない．クリティカルパスでない部分は共有可能で，共有することにより全体の面積を削減できる．最長経路以外の論理については，OR 側は AND ゲートを削除した回路 A1 と従来の回路，AND 側は従来の回路と OR ゲートを削除した回路 A2 で Voting を保証する．

Voting による出力 F，回路による出力 Y，回路 A1 による出力 W，回路 A2 による出力 Z，は以下の式によって求められる．

$$F = Y \cdot W + Y \cdot Z + W \cdot Z$$

$$Y = f(x, A, B, C, D) = (x \cdot A + B) \cdot C + D$$

$$W = x + B + D$$

$$Z = x \cdot A \cdot C$$

3 重化法の論理等価性を示すと次のようになる．

$$W \cdot Y = x \cdot A \cdot C + x \cdot A \cdot B \cdot C + x \cdot A \cdot C \cdot D = x \cdot A \cdot C = W$$

$$W \cdot Z = x \cdot A \cdot C + x \cdot A \cdot B \cdot C + x \cdot A \cdot C \cdot D = x \cdot A \cdot C = W$$

$$Y \cdot Z = x \cdot A \cdot C + x \cdot B \cdot C + x \cdot D + x \cdot A \cdot B \cdot C + B \cdot C + B \cdot D + x \cdot A \cdot C \cdot D + B \cdot C \cdot D + D = x \cdot A \cdot C + B \cdot C + D = Y$$

$$F = f$$

ここで興味深いのは，W と Y や Z との AND の結果が W になることで，これが後で述べる 0&1 スキップ法と関連している．

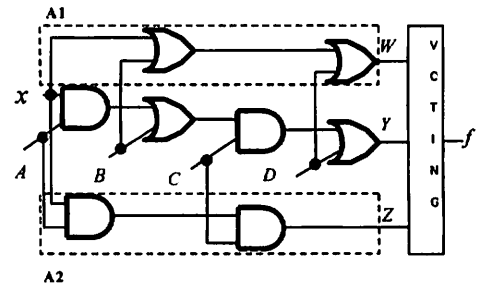

図 4　TMR(3 重化法)

3.2 セレクタ法

セレクタ法は，最長経路を活性化する条件で最長経路の始まりの信号をマルチプレクサでバイパスし，false path 化する手法である．図 3の回路にセレクタ法を適用した例を図 5に示す．このとき，AND をとるゲートを高速伝播回路として付加し，マルチプレクサの制御入力とする．最長経路上の OR ゲートは，0 のとき活性化するため，最長経路のサイドインプットを否定して，AND をとっている．高速伝播の条件は，$A \& \overline{B} \& C \& \overline{D} = 1$である．マルチプレクサによって，高速伝播回路の出力(sel)が 1 なら f の出力は x，0 なら出力を y とする．マルチプレクサの出力 f の式は以下の通りである．

$$f = x \cdot (sel) + y \cdot (\overline{sel})$$
$$sel = A \cdot \overline{B} \cdot C \cdot \overline{D}$$
$$y = x \cdot A \cdot C + B \cdot C + D$$

セレクタ法でも以下に示すように論理の等価性を示すことができる．

$$f = x \cdot (A \cdot \overline{B} \cdot C \cdot \overline{D}) + (x \cdot A \cdot C + B \cdot C + D) \cdot \overline{(A \cdot \overline{B} \cdot C \cdot \overline{D})}$$
$$= x \cdot A \cdot \overline{B} \cdot C \cdot \overline{D} + \overline{A} \cdot B \cdot C + \overline{A} \cdot D + \overline{C} \cdot D +$$
$$x \cdot A \cdot C + B \cdot C + B \cdot D + x \cdot A \cdot C \cdot D + B \cdot C \cdot D + D$$
$$= x \cdot A \cdot C + B \cdot C + D$$

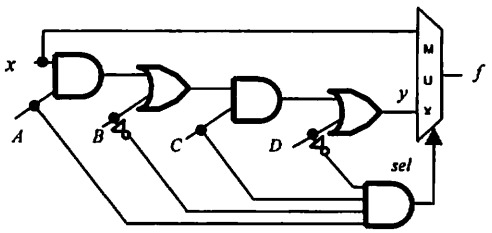

図 5 セレクタ法

3.3 0&1 スキップ法

0&1 スキップ法は，最長経路の入力が 0 として出力へ伝わる場合と，1 として出力へ伝わる場合を分けて高速伝播する手法である．これは，桁上げスキップ加算で用いられているアイデアを一般化したものである．

一般に桁上げ飛び越し加算と呼ばれている回路は，文献[8]の実験結果でも示されているように，最大遅延が順次桁上げ加算に比べて改善されない．これは，桁上げ飛び越しの回路が 1 の伝播のみを考え，最長経路を false path 化したことになっていないためである．最長経路は 1 が伝わる場合だけでなく，0 が伝わる場合もある．

そこで，1 伝播のスキップだけではなく，0 伝播のスキップも考慮した false path 化手法を考案した．0 伝播，1 伝播の条件を一般化して考えた結果，3 重化法と非常に近い条件となることがわかった．すなわち，最長経路の部分の AND を除いた回路が 0 伝播回路となり，OR を除いた回路が 1 伝播回路となることがわかった．なお，これは内部に否定ゲート意がない場合である．0 伝播回路は AND でもとの回路出力と接続し，1 伝播回路は OR でもとの回路出力と接続する．

内部に否定ゲートがある場合は，ドモルガンの法則で否定を入力側か出力側に移動して，内部の AND, OR 回路で信号伝播を考える．

例として，図 3の回路図をクリティカルパス上の回路として考えると，0&1 スキップ法を適用した場合は，図 6 のようになる．x が入力で，出力を f とし，各々のサイドインプットを A, B, C, D とする．1 の伝播スキップ回路は，AND ゲートのサイドインプット(A, C)の AND をとり，OR ゲートを付加する．0 の伝播スキップ回路は，OR ゲートのサイドインプット(B,D)の OR をとり，AND ゲートを付加する．

$0skip = x + B + D$

$1skip = x \cdot A \cdot C$

$x+B+D$ が 1 でないと 0 が確定し，$x \cdot A \cdot C$ が 1 だと 1 が確定する．

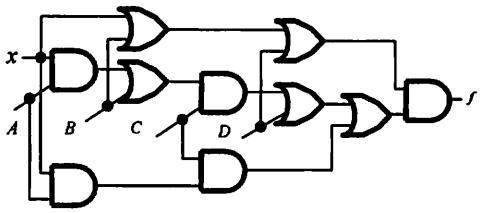

図 6 0&1 スキップ法

0&1 スキップ法の論理等価性を示すと，以下のようになる．

$$(f(x,A,B,C,D) + x \cdot A \cdot C)(x + B + D)$$
$$= (x \cdot A \cdot C + B \cdot C + D + x \cdot A \cdot C)(x + B + D)$$
$$= (x \cdot A \cdot C + B \cdot C + D)(x + B + D)$$
$$= x \cdot A \cdot C + x \cdot B \cdot C + x \cdot D + x \cdot A \cdot B \cdot C + B \cdot C +$$
$$B \cdot D + x \cdot A \cdot C \cdot D + B \cdot C \cdot D + D$$
$$= x \cdot A \cdot C + B \cdot C + D$$

図 6 では，出力に近い側に 0 スキップ用の AND を 1 段，入力側に 1 スキップ用の OR を 1 段置いたが，論理関数の性質から，これらの順序を入れかえても論理的には等価であることがわかった．1-skip, 0-skip の順の回路を図 7 に，0-skip, 1-skip の順の回路を図 8 に示す．

図 7の回路の論理関数は，先に示したように，$(f(x,A,B,C,D)+x \cdot A \cdot C)(x+B+D)$ と表せる．1-skip である $x \cdot A \cdot C$ はもとの関数に含まれており，0-skip である $(x+B+D)$ はもとの関数に含まれる積項を再構築していることが分かる．一方図 8 の回路を関数で表すと $f(x,A,B,C,D)(x+B+D)+x \cdot A \cdot C$ となる．この関数を展開しても，もとの関数と同じ $x \cdot A \cdot C + B \cdot C + D$ となる．これらの二つの回路の時間的な性質については，より詳しい解析が必要であるが，順序を入れ替えても良いという性質を使うことで，全体の遅延時間の最適化を行うことができる．すなわち，どちらの順序がより小さな遅延となるかに応じて順序を決定すればよい．

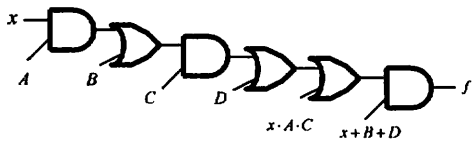

図 7　1-skip，0-skip の順

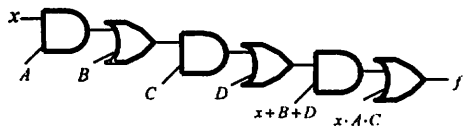

図 8　0-skip，1-skip の順

4. 評価方法

前節では，false path化による高速化手法について述べた．提案した手法の有効性を検討するために，加算器を例題にして，評価を行った．評価方法としては，与えられた回路に対し，それを高速化した回路を作成し，それらの論理合成，レイアウトを行う．レイアウトにより，レイアウト後の回路データを用いてタイミング解析を行う．8bit，16bitの順次桁上げ加算器，桁上げ飛び越し加算器，３重化法，セレクタ法，0&1スキップ法を適用した回路を設計し，論理合成，レイアウト，ポストレイアウトシミュレーションについて，評価した．評価方法としては，ripple(順次桁上げ加算器)回路を基準として遅延時間がどの程度削減されたかを計測した．

図9, 10, 11 に評価で用いた回路構造を示す．これらは，順次桁上げ加算器の合成結果の回路をもとに，手で設計を行った．図9の3重化法の中央部の回路が順桁上げ加算の回路図である．3重化法では，回路中のAND ゲートおよびORゲートのみに着目して回路変更を行い，NOR ゲートやインバータなどはそのままとした．図11の0&1スキップ回路の構成では，もとの回路の出力にインバータがあることを考慮し，0-スキップ回路と1-スキップ回路とを入れ替えている．通常では，AND を除いた回路が 0-伝播となるはずであるが，出力のインバータの影響で，この回路では1-伝播となっている．

これらの回路の合成およびレイアウトにより，配線の詳細な遅延情報が得られるので，レイアウト後の回路データを用いてタイミング解析を行った．今回の解析では，静的なタイミング解析ではなく，Synopsys VCSでSDFファイルを用いたタイミングシミュレーションを行った．SDFファイルを用いたタイミングのアノテーションにより，配置配線も考慮した遅延が計算される．タイミングシミュレーションでは，最長経路と思われるパスを活性化する入力パターンを与えて，出力の遅延の評価を行った．静的な手法では，最長経路が真に false path になっているかどうかがわからないため，タイミングシミュレーションを用いた．

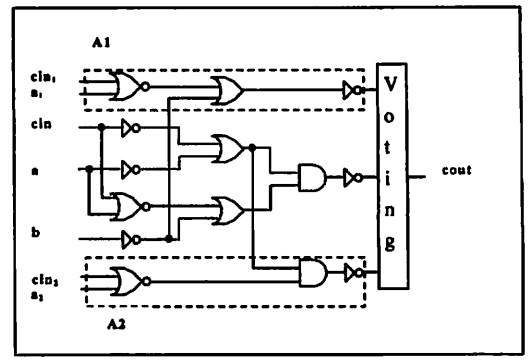

図 9　３重化法　1bit 評価回路

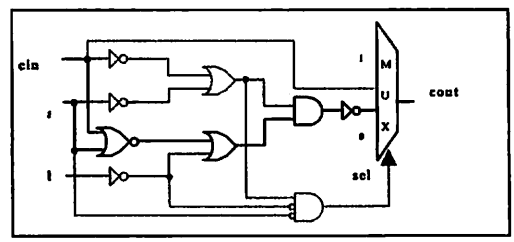

図 10　セレクタ法　1bit 評価回路

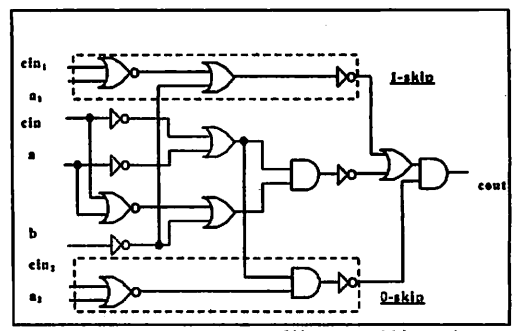

図 11　0&1 スキップ法　1bit 評価回路

5. 評価結果

表 1　8bit タイミングシミュレーション結果

bit 数	8bit					
	面積 (n㎡)	面積増加率 (%)	0->1		1->0	
			遅延 (ps)	削減率 (%)	遅延 (ps)	削減率 (%)
ripple	17799	-	477	-	324	-
carry skip	19086	7.23	221	53.67	362	-11.73
TMR	19722	10.80	206	56.81	218	32.72
select	21644	21.61	191	59.96	194	42.59

0&1-skip	19633	10.31	196	58.91	210	35 19

表 2　16bit タイミングシミュレーション結果

bit 数	16bit					
	面積 (n㎡)	面積増加率 (%)	0->1		1->0	
			遅延 (ps)	削減率 (%)	遅延 (ps)	削減率 (%)
ripple	35015	-	671	-	506	-
carry skip	37679	7 61	313	53.35	731	-44.47
TMR	38490	9.92	313	53.35	348	31 23
select	42336	20.91	309	60.66	329	34 98
0&1-skip	38402	9.67	305	54 55	327	35 38

タイミングシミュレーションの結果を表 1 に表す．面積の増加率は平均で 12%ほどだった．最も面積が増加したのは，セレクタ法を適用した回路で 21%増加した．0→1 の変化の遅延の削減率は，全手法が 16bit,8bit ともに 50%以上削減できた．1→0 の変化も含め，全体的に最も遅延が削減できたのがセレクタ法であった．提案手法である 0&1 スキップの結果は，面積増加率は約 10%，0→1 の変化に対する遅延の削減率は約 57%，1→0 の変化に対する遅延の削減率は約 35%という結果になった．

それぞれの手法を適用した回路を Quartus II Version6.0 でマッピングを行い，デバイスは stratix I (EP1S25F1020C7:25,660LE(Logic Elements))を使用した．結果を表 3 に示す．0&1 スキップでは，TMR に比べて 16 ビットで LE 数の削減が見られた．詳しい解析は今後の課題である．

表 3　8bit，16bit FPGA マッピング結果

8bit	LE
Ripple	33
Carryskip	38
TMR	41
Select	36
0&1-skip	41

16bit	LE
Ripple	65
Carryskip	76
TMR	82
Select	72
0&1-skip	81

6. おわりに

ハードウェアの設計検証期間の高速化を目的とし，エミュレーション回路の高速化手法についての研究を行った．手法としては，同期式マイクロパイプラインの提案と同時に，各段の組合せ回路部の高速化に着目し，最長経路を false path 化して高速化するという手法を提案した．まず同期式マイクロパイプラインについては，FPGA を用いて速度の高速化がどの程度はかれるかをチェックした．また，最長経路を false path 化する時には，0 の伝播と 1 の伝播を分けて考え，0&1 スキップ法を提案した．まず提案手法により生成される回路が，もとの回路と論理的に等価な回路であることを示し，いくつかの回路に対して評価を行った．この過程で，0 と 1 のスキップ回路の順序を入れ替えても論理的には等価であることが分かった．加算器を例題として提案した手法を評価した結果，面積は平均で 15%前後増加し，遅延は平均で 50%ほど削減できた．

今回提案した 0&1 スキップ法により，面積は約 12%増加したが遅延は 50%前後削減されたことから，エミュレーションの高速化として有効であると考えられる．ただし，今回の実験は非常に限定された回路であるので，一般的な回路への適用は今後の課題である．

今後は，(1) 回路の高速化部分の検討，高速化回路の構造の検討，(2) 自動的な回路の付加手法の考察と評価とベンチマーク回路を用いた評価，(3) FPGA でのエミュレーションに向けた最適化手法の検討を行う予定である．

謝辞

日頃から御討論いただく，本学吉村猛教授および渡邊孝博教授はじめ木村研究室の皆様に感謝します．研究にあたり，補助をいただく松下電器産業株式会社の九郎丸 俊一様，木村　智生様に感謝します．本研究は一部，松下電器産業株式会社委託研究費，文部科学省北九州知的クラスタプロジェクト，日本学術振興会科学研究費補助金による．

参考文献

[1] Shih-Chieh Chang, Cheng-Tao Hsieh, Kai-Chiang wu, "Re-synthesis for Delay Variation Tolerance", Proc.DAC 2004, pp.814-819, June 2004.

[2] Jacques Benkoski, Erik Vanden Meersch, L.J.M. Claesen, Hugo De Man, "Timing Verification Using Statically Sensitizable Paths", IEEE Transactions on Computer-Aided Design. Vol.9, No.10, pp.0723-10784, Oct 1990.

[3] Karem A. Sakallah , "False Path Elimination in Timing Verification", EECS Department The University of Michigan, Ann ArbOR, July 10, 2000.

[4] Petlin,O.A., Furber,S.B., "Built-In-Self-Testing of Micropipelines", Advanced Research in Asynchronous Circuits and Systems, IEEE, pp. 22 -29, 1997

[5] Sutherland, I. E., "Micropipelines", ommunications of the ACM,Vol. 32, no. 6, pp. 720-738, , June 1989.

[6] Y.Zafar, M.M.Ahmed, "A Novel FPGA Compliant Micropipeline", IEEE Transactions on Circuits and Systems □:Express Briefs, Vol.52, No.9, pp.611-615, Sept 2005.

[7] 樋口 博之，松永 祐介，" 順序回路のタイミング例外パス検出のための実用的方法 " , 2004-SLDM-117, pp.203-208, Dec 2004.

[8] 味元伸太郎，橘　昌良，水口　貴之，"加算器の消費電力/面積/速度の形式による比較", VLD, pp.13-17, May 2005.