計算機アーキテクチャ　103－10
設　計　自　動　化　69－10
(1993. 12. 16)

FPGA 上の同期回路に対する高速化を目的としたラッチ挿入法

宮崎敏明　中田広　筒井章博
山田一久　太田直久

NTT 伝送システム研究所
〒238-03 神奈川県横須賀市武 1-2356, Y-807C
e-mail: miyazaki@ntttsd.ntt.jp

あらまし　本稿では、FPGA に配置／配線された回路情報を保存したまま、回路内へのラッチ挿入操作によって、動作速度を改善する手法を提案する。本手法では、問題を整数線形計画法によって定式化し、ラッチ挿入位置をも求めている。配置／配線を変更しないため、本手法を適用した場合、原回路の動作速度は保証され、問題の解が存在すれば、多くの場合、動作速度の改善できる。問題の定式化を示した後、実験により本手法の有効性を示す。

和文キーワード　リタイミング、整数線形計画法、伝送

Latch Insertion Method for Synchronous Circuits Realized as LUT-Based FPGAs that Improves Performance

Toshiaki Miyazaki,　Hiroshi Nakada,　Akihiro Tsutsui
Kazuhisa Yamada,　Naohisa Ohta

NTT Transmission Systems Laboratories

Y-807C, 1-2356 Take, Yokosuka-shi, Kanagawa, 238-03 JAPAN

Abstract　This paper presents a new method for improving the performance of synchronous circuits configured in look-up table based FPGAs; the initial circuit configuration is unchanged except for the addition of some latches. We formulate the problem in Integer Liner Programming (ILP) and find the latch insertion points. Because our method does not change the initial configuration, the circuit performance must be improved if the method can be applied to the circuit. After formulating the problem, the effectiveness of the method is shown by some experimental results.

英文 key words　Retiming, ILP, Digital signal transport

1 まえがき

近年、その利便性と扱える規模や性能が向上したこともあり、Field Programmable Gate Array (FPGA) が盛んに使用されるようになってきた。通常、FPGA に対するプログラミングは、CAD ツールを用い自動的に行なわれることが多く、最終的な動作速度は、それらツールの性能に左右されると言っても過言ではない [1]。例えば、Lookup Table(LUT) 方式の FPGA を考えた場合、FPGA のプログラミングとは、LUT への論理のマッピングおよびその配置配線問題を解くことであり、よく知られているようにそれら問題は、NP 困難または NP 完全な問題である。それゆえ、従来の CAD ツールでは、ヒューリスティクなアルゴリズムが使用されてきた [2][3][4][5][6]。そのため、最終的な FPGA のプログラミング結果は、動作速度の面から最適である保証はなく、改善する余地が残されている。

過去、動作速度改善手法として、論理を変更せずに回路内のラッチを移動することによって、ラッチ間にはさまれた組合せ回路内の遅延を小さくし、供給クロック周波数を上げようという「リタイミング」という手法が検討されてきた [7][8][9][10]。

また、伝送処理分野では、ラッチを積極的に回路に挿入して、クロック周波数を高くすることが一般的手法として広く用いられている [11]。この場合、一般的リタイミングとは異なり、原回路に対して入力信号のクロック応答遅延を生じるが動作速度を重視する上記分野等では、速度改善の確実な手法として定着している。同様な手法は CPU などにおけるパイプライン設計手法の中に見い出すことができる [12][13]。

上記のいずれの手法も、残念なことに、同期回路を PCB または ASIC 等で実現することを暗に仮定しており、ラッチを挿入／移動する際に、それが物理的に可能かどうかは、深く考慮されていない。しかし、FPGA に回路を実現することを想定すると、FPGA 内で使用可能なラッチ数は有限であり、しかも、その位置が固定的であることを考慮せねばならず、単純に上記手法を回路高速化手法として適用することはできない。

本稿では、FPGA を伝送処理に適用する場合を想定し、FPGA にプログラミングされた回路に対してラッチを挿入／移動することのみで動作速度を改善する手法を提案する。本手法が仮定する FPGA の構成は、各 LUT の出力にラッチが挿入できる機構を持ち、しかもそれらの操作が、初期のプログラミング結果、すなわち配置／配線結果を変更せずにラッチ挿入／非挿入が制御できるものとする。

過去、遅延情報を保存するために配置／配線結果を変更せずに LUT 内の論理を変更することによって小規模な論理変更に対応することを狙った研究が報告されている [14]。しかし、LUT 内の論理変更によって所望の論理変更が達成できることが少なく、結局、再配置／配線を行なわなければならないことが多いことを考えるとその手法は、実用的なものとは言い難い。しかも、その手法は、動作速度改善を狙ったものではないため、論理変更後の回路の動作速度は、元の回路の速度と同じである。

本稿の構成は、以下の通りである。2 章で、我々が取り扱う問題を定義したのち、3 章で整数線形計画法 (ILP) を用いた問題の定式化および解法を示す。4 章では、幾つかのベンチマークデータに対して提案手法を適用した結果を示す。5 章では、まとめを行なうとともに今後の課題についても触れる。

2 問題の定義

FPGA 内部の初期配置／配線結果を変更せずにラッチ挿入／移動のみを行なうことは、配線遅延を、そのまま保存することを意味し、クリティカル・パス上にラッチが新たに挿入できれば、必ず動作速度の改善が可能である。はじめに、本稿で仮定する FPGA の基本セル (BC) の構造を図 1に示す。それぞれの BC は、LUT とラッチを一つずつ持つ。k 入力の LUT は、2^{2^k} 通りの異なる論理関数を実現することができる [1]。また、LUT の出力をラッチするかしないかを、他の配置／配線に影響を与えることなく任意に制御できるものとする。さらに、FPGA 内に実現された回路は、その経路に一つもラッチを含まないループ、すなわち、非同期のフィードバック・ループを持たないものとする。

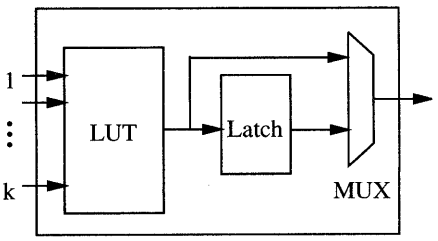

図 1: 基本セルの構造

上記の仮定のもとで、本問題を以下の様に定義する。

問題

各LUTの出力に位置するラッチを使用するかしないかを他の配置／配線に影響を与えることなく独立に制御できる機構を持ったFPGAを仮定する。このとき、各LUT出力のラッチ挿入／非挿入を制御することのみで、原回路に対する「qクロック応答遅延回路」が構成できるか？ □

ここで、「qクロック応答遅延回路」とは、文献[11]と同様に、以下のように定義する。ある回路Aにおいて、すべての入力端子におけるある入力系列Xに対する時刻t_nの出力ベクトルを$Y_A(X,t_n)$とする。ここで、回路Aと入出力端子が一対一対応にある回路Bを考える。いま回路の初期時刻をt_0とし、すべての$n \geq 0$, および、任意の入力系列Xについて、0以上の整数qが存在し

$$Y_B(X,t_{n+q}) = Y_A(X,t_n) \qquad (1)$$

が成り立つとき、「回路Bは回路Aのqクロック応答遅延回路である」と定義する。

3 整数線形計画法による問題の定式化

前述の問題を解くために整数線形計画法(ILP)に基づく定式化を行なう。

3.1 準備

配置配線後のネットリストから、図2のような回路グラフG=(V,E)を作成する。ここで、Vは、BCの集合である。ただし、始点{v1, v2}および終点{v15, v16, v17}ノードは、それぞれ入力端子および出力端子を示し、それらにはラッチが存在するものとする。Eは、各CB間を結ぶ配線の集合であり、それぞれ遅延を重みとして持つ。ここで、重みは、一般に負でない実数とする。図上、v8, v12, v14は、原回路において、対応するBCのラッチが使用されていることを示している。また、v3, v5, v6, v8, v9, v14は、提案手法によってラッチが配置されたノードを示す。

つぎに、作成した回路グラフの入出力ノード間のすべてのパスおよびグラフ内のループを列挙する。図2に示したグラフの例では、表1に示したように、パスは14個、ループは1個存在する。

表1: 図2のグラフに存在する入出力端子間のすべてのパスおよび内在するループ

	パス							
1	v1	v3	v7	v11	v14	v15		
2	v1	v3	v8	v11	v14	v15		
3	v1	v3	v8	v13	v16			
4	v2	v4	v3	v7	v11	v14	v15	
5	v2	v4	v3	v8	v11	v14	v15	
6	v2	v4	v3	v8	v13	v16		
7	v2	v4	v5	v9	v12	v13	v16	
8	v2	v4	v5	v9	v12	v17		
9	v2	v4	v6	v8	v11	v14	v15	
10	v2	v4	v6	v8	v13	v16		
11	v2	v4	v6	v10	v9	v12	v13	v16
12	v2	v4	v6	v10	v9	v12	v17	
13	v2	v5	v9	v12	v13	v16		
14	v2	v5	v9	v12	v17			
	ループ							
1	v9	v10	v12					

3.2 記法

以後に示す定式化において使用する記号を以下のように定める。

1. P: 回路グラフGの始点から端点までのパスの個数.
2. N_i: パスi上のノード数. ただし、始点／終点ノードは含まない.
3. L_i: パスi上のラッチが存在するノードの数.$L_i \leq N_i$.
4. n: 各パス上のラッチの数.
5. $v_{i,j}$: あるノードに対応した0-1変数. パスi上のノードjにラッチが存在すれば1、その他は0. $1 \leq i \leq P$. $1 \leq j \leq N_i$. $v_{i,j} \in V$.
6. L: ループを形成するパスの総数.
7. V_i: ループi内に存在するノード数. $V_i > 0$.
8. M_i: ループi内に存在するラッチを含むノード数. $M_i > 0$. $1 \leq i \leq L$.
9. v_j^i: あるノードに対応した0-1変数. ループi内に存在するノードjにラッチが存在すれば1、その他は0. $1 \leq i \leq V$. $1 \leq j \leq V_i$. $v_j^i \in V$.

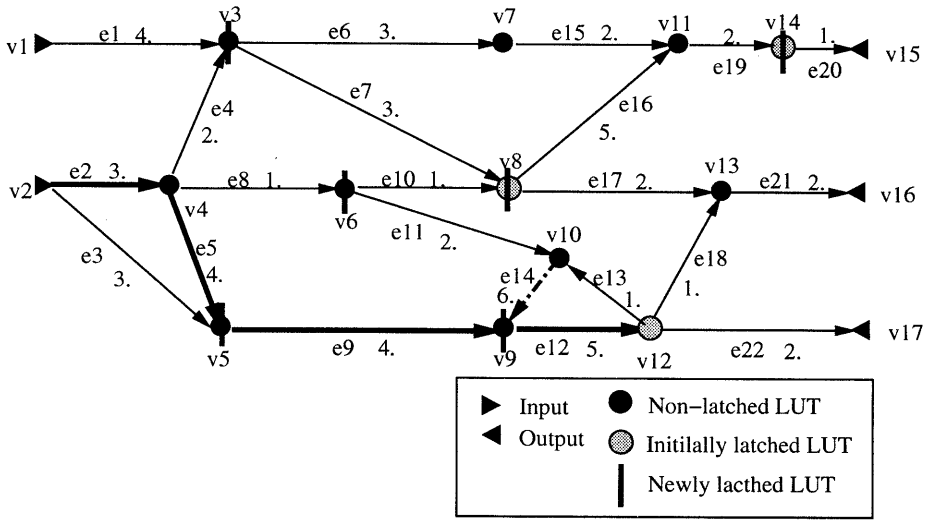

図 2: 回路グラフ。各ノードは一つの基本セルを表し、エッジは、基本セル間の接続を表す。エッジは、遅延を重みとして持つ。

3.3 定式化

ここでは、目的関数 F を式 (2) とし、以下に示す 4 つの制約条件を満たすように、F を最大にするときの、全てのノード変数 $v_{i,j}$, v_j^i の値を求める問題とする。

$$F = n \quad \rightarrow \quad max. \tag{2}$$

制約条件 1

各パス上に新たに挿入されるラッチの数は等しく、その数は n である。(n クロック応答遅延の保証)

$$\sum_{j=1}^{N_i} v_{i,j} - n = L_i, \qquad 1 \leq i \leq P. \tag{3}$$

制約条件 2

各パスに挿入可能なラッチ数の上限。

$$n \leq \min_{1 \leq i \leq P} (N_i - L_i) \tag{4}$$

ここで、$n = 0$ とすると、本手法は、いわゆるリタイミング問題そのものとなる。(**0** クロック応答遅延回路の保証)

制約条件 3

ループ内のラッチ数は、原回路のまま変化しない。

$$\sum_{j=1}^{V_i} v_j^i = M_i, \qquad 1 \leq i \leq L. \tag{5}$$

制約条件 4

最大遅延を持つ配線によって接続された 2 つの BC の内、最低 1 つにはラッチを挿入する。(ヒューリスティックな条件)

$$v_i + v_j \geq 1 \tag{6}$$

ただし、回路グラフ G 内で、ノード v_i と v_j を結ぶエッジは、最大遅延値を持つ。

3.4 例

図 2の回路グラフに対する定式化の例を示す。
目的関数

$$F = n \rightarrow max. \tag{7}$$

制約条件 1 ：入出力間のパスに対する条件

$$v3 + v7 + v11 + v14 - n = 1$$
$$v3 + v8 + v11 + v14 - n = 2$$
$$v3 + v8 + v13 - n = 1$$

$$
\begin{aligned}
v4+v3+v7+v11+v14-n &= 1\\
v4+v3+v8+v11+v14-n &= 2\\
v4+v3+v8+v13-n &= 1\\
v4+v5+v9+v12+v13-n &= 1\\
v4+v5+v9+v12-n &= 1\\
v4+v6+v8+v11+v14-n &= 2\\
v4+v6+v8+v13-n &= 1\\
v4+v6+v10+v9+v12+v13-n &= 1\\
v4+v6+v10+v9+v12-n &= 1\\
v5+v9+v12+v13-n &= 1\\
v5+v9+v12-n &= 1 \qquad (8)
\end{aligned}
$$

制約条件2：一パスに対するラッチ挿入数の上限

$$n \leq 2 \qquad (9)$$

制約条件3：ループに対する条件

$$v10+v12+v9 = 1 \qquad (10)$$

制約条件4：最大遅延経路を分断する条件

$$v10+v9 \geq 1 \qquad (11)$$

上記の例を解いた結果を、図2に合わせて示す。ノードv3, v5, v6, v8, v9, v14にラッチが設けられ原回路に対して「1クロック応答遅延回路」が実現されている。

4 実験結果

前述した問題を解くプログラムを、C言語および市販線形計画解法パッケージLINDOを用いて、UNIXマシン(SUN SparcStation 10/41, メモリ:32MB)上に実現した。また、FPGAとしてXilinx社製LCA 3042PG132-125を用い、配置／配線は、Xilinx社のCADツールを用いて行なった[15]。ここで用いたFPGAは、Configurable Logic Block (CLB)という基本ブロックを単位として構成されており、それぞれのCLBは、5つの入力端子とラッチ可能な2つの出力端子を持っている。また、"F", "FG", "FGM"という3つの異なるコンフィギュレーション・モードがあり、一つのCLBを異なる論理ブロックとして使用できるようになっている。そのため、回路グラフを作成する際、個々のCLBのモードを考慮した。例えば、"FG"モードでコンフィギュレーションされたCLBは、図1で示した基本セルの2つ分に相当し、そのCLBからは2つのノードを生成している。実験で使用したLCA 3042PG132-125は144個のCLBを持っている。

Logic Synthesis Workshop 1991のベンチマーク・データに対して本手法を適用した結果を表2に示す。表は、左の欄から、データ名、該当データが使用しているCLBの個数、原回路の最大遅延、本手法適用後の最大遅延、改善比、本手法の実行時間をそれぞれ表している。図中、NGとあるのは、本手法により、ラッチ挿入ができなかったことを示す。本手法が、適用できなかった回路は、入出力が直結している信号線を持っているか、ある入出力パス上に新たにラッチが挿入できるノードが存在しないものであった。それぞれの回路は、組合せ回路であり、改善が行なえたものは、すべて「1クロック応答遅延回路」となった。組合せ回路の場合、ループが存在しないことから、クリティカル・パスは、必ず入出力端子間に存在し、ラッチを挿入できればそのクリティカル・パスを分断するため、動作速度は向上できる。式(6)に示したヒューリィスティックな制約条件は必ずしも必要としないため、ここでは、式(6)の制約条件を与えずに解を求めた。この方が、解が存在する可能性が多くなる。

また、表中最終行のデータ"bool"は、我々が用意した順序回路であり、入出力間の各パスにラッチが1つ存在する回路である。本手法を適用した結果「0クロック応答遅延回路」が構成できた。動作速度が向上しているのは、ラッチの位置が移動されたことによるものであり、一般的なリタイミングが行なわれたこと示している。

ここで使用したFPGAは、ラッチの挿入に際して、クロック信号をクロック端子から配線しなければならないため、ラッチ挿入後、実際は、再配線を行なっている。このとき、クロックは、クロック専用線を用いて配線し、その他の配置／配線結果は原回路のものをできるだけ変更しないようにCADツールに指示して行なった。そのため、実際には、クロック配線だけがラッチの挿入に対して追加された形となり、また、その追加されたクロック用配線がクリティカル・パスになることもなかった。これより、「ラッチ挿入に際して他の配置配線情報を変更しない」と仮定は実質的に満たされている。

5 まとめ

FPGA上に実現される伝送処理回路を想定したラッチ挿入による動作速度改善手法を提案した。本手法は、整数線形計画法を用いて問題をモデル化し、FPGA上に配置／配線された回路構造を変更することなく、各LUT出力のラッチの有無のみを制御することによって速度改善を行なう。配置／配線後の最終結果を何ら変更することなく動作速度改善が図れる本手法は、実用的に有効な手法であると考える。

表 2: 実験結果 (Xilinx 3042PG132-125:144CLB)

data	CLB	max. delay		ratio	CPU (s)
		A: init. (ns)	B: applied (ns)	(B/A)	
9symml	47	90.1	70.1	0.78	0.5
cm138a	5	42.4	25.8	0.61	0.0
cm150a	12	60.6	40.6	0.67	0.0
cm151a	6	50.5	29.2	0.58	0.0
cm152a	5	47.0	25.8	0.55	0.0
cm162a	10	55.5	33.6	0.61	0.1
cm163a	9	53.3	31.6	0.59	0.1
cm42a	7	47.4	26.2	0.55	0.0
cm82a	2	38.9	25.8	0.66	0.0
cm85a	9	59.9	38.6	0.64	0.1
cmb	11	70.4	50.4	0.72	0.1
mux	20	85.2	63.3	0.74	0.3
majority	1	30.2	24.5	0.81	0.0
c17	2	44.5	25.8	0.58	0.0
c499	66	117.3	97.3	0.83	52.8
c1355	66	120.0	97.5	0.81	55.1
b1	1	33.6	NG	1.00	0.0
b9	31	65.4	NG	1.00	0.3
x1	96	85.3	NG	1.00	1.1
x2	16	64.9	NG	1.00	0.1
c8	33	73.4	NG	1.00	0.2
bool	7	48.2	26.4	0.55	0.1

現在のモデル化では、入力端子から出力端子に至る回路内の全経路を列挙する必要があるが、ループの存在も許すため一般には多項式時間で解けない。ループの存在やその位置を事前に知ることは、実際の伝送処理回路等では比較的容易であることから、それら情報を用いて計算量を削減する手法やモデルの簡単化を行なって行く予定である。

参考文献

[1] Brown S.D., Francis R.J., Rose J. and Vranesic Z.G. : "Field-Programmable Gate Arrays," Readings, Kluwer Academic Publishers, 1992.

[2] Francis R.J., Rose J. and Chung K.: "Chortle: a Technology Mapping Program for Lookup Table-Based Field-Programmable Gate Arrays," Proc. 27th DAC, pp.613-619, June 1990.

[3] Francis R.J., Rose J. and Vranesic Z.: "Chortle-crf: Fast Technology Mapping for Lookup Table-Based FPGAs," Proc. 28th DAC, pp.227-233, June 1991.

[4] Karplus K.: "Xmap: a Technology Mapper for Table-lookup Field-Programmable Gate Arrays," Proc. 28th DAC, pp.240-243, June 1991.

[5] Rose J.: "Parallel Global Routing for Standard Cells," IEEE Trans. on CAD, Vol.9, No.10, pp.1085-1095, October 1990.

[6] Rose J. and Brown S.: "The Effect of Switch Box Flexibility on Routability of Field Programmable Gate Arrays," Proc. 1990 CICC, pp.27.5.1-27.5.4, May 1990.

[7] Leiserson C.E., Rose F.M. and Saxe J.B.: "Retiming Synchronous Circuitry," Algorithmica, Vol.6, pp. 5-35, 1991.

[8] Dey S., Potkonjak M. and Rothweiler S.G: "Performance Optimization of Sequential Circuits by Eliminating Retiming Bottlenecks," Proc. ICCAD-92, pp. 504-509, November 1992.

[9] Iqbal Z., Potkonjak M., Dey S. and Parker A.: "Critical Path Minimization Using Retiming and Algebraic Speed-Up," Proc. 30th DAC, pp.573-577, June 1993.

[10] Malik S., Singh K.J. and Brayton R.K.: "Performance Optimization of Pipelined Logic Circuits Using Peripheral Retiming and Resynthesis," IEEE Trans. CAD, Vol. 12, No. 5, pp. 568-578, May 1993.

[11] 中田, 山田, 筒井, 太田: "同期回路における論理を保存したラッチの挿入手法," 信学会論文誌 A, J75-A, No.12 pp. 1849-1858, December 1992.

[12] Cloutier R.J. and Thomas D.E.: "Synthesis pf Pipeline Instruction Set Processors," Proc. 30th DAC, pp.583-588, June 1993.

[13] Park N. and Parker A.C.: "Sehwa: A Software Package for Synthesis of Pipelines from Behavioral Specifications," IEEE Trans. on CAD, Vol.7, No.3, pp.356-370, March 1988.

[14] Kukimoto Y. and Fujita M.: "Rectification Method for Lookup-Table Type FPGA's," Proc. ICCAD-92, pp. 54-60, November 1992.

[15] —: "The Field Programmable Gate Array Data Book," Xilinx Inc., 1992.