

SPFDを用いた新しい置換法に基づくLUT型FPGAの最適化

世古 忠

Optimization of LUT-typed FPGA Using SPFD Based on New Replacement Method

Tadashi SEKO

This paper presents a new optimization method to reduce the number of LUTs and wires using SPFD based on new replacing method. The new replacing method selects the LUTs and wires to be replaced according to the priority. In order to remove much LUTs and wires the proposed method introduces the idea of the distance between the output function of LUT and the wire's function. The experimental evaluations using MCNC benchmark circuits show that the proposed method attains much reduction ratio of the number of LUTs compared with the previous method.

1. はじめに

半導体技術の発展に伴い、集積回路の構成法も進化し、プログラム可能なハードウェアが実現されるようになった。FPGA[1,2](Field Programmable Gate Array)は、書き換え可能な論理素子であり、内部に3～6入力の任意の論理関数を実現できる多数のLUT(Look-Up Table)およびそれらの間の接続を任意に行なうことができるスイッチボックスを備え、所望の論理回路をプログラマブルに実現できる。近年その性能が著しく向上し、信号処理や画像処理など様々な応用面で注目されている。このようなFPGAで論理回路を実現するためには、ゲートレベルの論理回路を部分回路に分割し、LUTのネットワークから成るFPGA回路に最適マッピングを行なう必要がある。このようなLUT型FPGAへのマッピング手法として、従来から多段論理回路の最適化手法として用いられてきた許容関数CSPF[3,4](Compatible Sets of Permissible Functions)や拡張許容関数SSPFを利用した手法[5]が提案されてきた。

最近、山下等によりSPFD[6](Sets of Pairs of Functions to be Distinguished)という論理関数の自由度を表す新しい概念が提案され、それを利用したFPGA回路の最適化手法が注目されている。この手法では回路変形において結線とLUTの置換を行なう際に、回路内のそれらの選択順序に対して特に考慮されてこなかったが、結線と

LUTの選択のパターンによっては最終的な結線数やLUT数が変わる可能性がある。本研究では、SPFDを用いた回路変形を行なう際に、新しく優先順序を付けたLUTと結線の選択法を用いて、さらにLUT数と結線数の削減を行なう手法を提案する。

2. 準備

2.1 記号の定義

本稿では図1に示すようなLUTと結線からなるLUT型FPGAを扱う。ここで用いる記号と文字を次のように定義する。

L_i : FPGA回路のあるLUT L_i を表す。

c_j : FPGA回路内のある結線 c_j を表す。

$f(L_i)$: 外部入力に対するLUT L_i の出力論理関数を表す。

$f(c_j)$: 外部入力に対する結線 c_j の論理関数を表す。

2.2 SPFDの定義

SPFDは論理関数の自由度の新しい表現法である。以下で用語の定義を行う。

[定義1] 2つの論理関数 f と g において、 $g(\mathbf{x})=1$ を満たすすべての入力 \mathbf{x} に対して $f(\mathbf{x})=1$ が成立するとき、 f は g を包含するという。

[定義2] 関数 f が以下の条件のいずれかを満たすとき、 f は2つの関数 g_1 と g_2 を区別しているという。

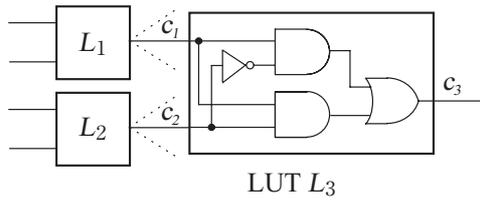


図1：FPGA回路例

(条件1) f が g_1 を包含し, \bar{f} が g_2 を包含する.

(条件2) f が g_2 を包含し, \bar{f} が g_1 を包含する.

[定義3] 区別すべき関数対からなる集合をSPFDと呼ぶ. SPFDは区別すべき関数対を用いて, $\{(g_{1a}, g_{1b}), (g_{2a}, g_{2b}), \dots, (g_{na}, g_{nb})\}$ のように表現される.

[定義4] あるSPFD $=\{(g_{1a}, g_{1b}), (g_{2a}, g_{2b}), \dots, (g_{na}, g_{nb})\}$ に関して, 関数 f がすべての i に対して, g_{ia} と g_{ib} を区別するとき, f はそのSPFDの条件を満たしている, という.

[定義5] LUTの出力関数に対するSPFDと結線が実現する関数に対するSPFDを以下のように表す.

SPFD(L_i): LUT L_i の出力関数に対するSPFD

SPFD(c_i): 結線 c_i 上の論理関数に対するSPFD

回路内のあるLUT L_i の出力 $f(L_i)$ が, 回路内のある結線 c_j のSPFDの条件を満たすとき, その c_j は $f(L_i)$ で置換できる. 本論文ではこのことを利用してLUT数と結線数を削減する.

3. SPFDの計算法

SPFDの計算手順を以下に示す. SPFDは回路の外部出力側のノードから外部入力側のノードへ伝播され計算される.

【SPFDの計算手順】

Step1: 回路内のすべてのLUT L_i に対して, $f(L_i)$ を計算する.

Step2: 回路内のすべてのLUT L_i に対して, SPFD(L_i)= $null$ として初期化する.

Step3: 回路内のすべての結線 c_i に対して, SPFD(c_i)= $null$ として初期化する.

Step4: 出力関数が外部出力 f であるLUTに対してSPFDを $\{f, \bar{f}\}$ とする.

Step5: Step4, Step7で計算したSPFDをさらに細かい区別すべき関数対に分割する.

Step6: Step5で分割した区別すべき関数対をLUTの入力結線に割り当て, それぞれをその結線に対するSPFDとする.

Step7: SPFDを計算した結線を出力結線として持つLUTにSPFDを伝播する.

Step8: すべてのLUTに対してSPFDを計算したのなら終了. そうでなければStep5へ行く.

4. 優先順位付き置換法を用いたFPGA回路の最適化法の提案

4.1 前処理

結線とLUTの番号付けの手順を以下に示す.

【結線とLUTの番号付けの手順】

Step1: 回路中のLUTの属性を $null$ として初期化する.

Step2: 回路中のLUTで, 外部入力のみを入力として持つLUTの属性を s とする.

Step3: 回路中のLUTで, 出力を外部出力とするLUTの属性を e とする.

Step4: 回路中の属性NULLのLUTと属性 s のLUTの間のLUT段数を計算し, その数字をそのLUTの属性とする.

Step5: 属性 s のLUT i 個に対して1から i の番号を付ける. それらのLUTの出力結線 j 本に対して1から j の番号を付ける.

Step6: 属性が数字であるLUT m 個に対して, 属性の昇順に $i+1$ から $i+m$ の番号を付ける. それらのLUTの出力結線 n 本に対して, $j+1$ から $j+n$ の番号を付ける.

Step7: 属性が e のLUT x 個に対して, $i+m+1$ から $i+m+x$ の番号を付ける.

この手順において, 属性が同じになるLUTが複数出てくる可能性があるが今回の実験ではその順番に関しては特に考慮しない.

4.2 FPGA回路最適化手法の提案

【基本アイデア】

基本アイデアを図2を用いて説明する.

図2(a)の例において, c_j を $f(L_i)$ で置換すると削除できるLUTは L_j のみであり図2(b)となる. しかし, c_k を $f(L_i)$ で置換すると L_j と L_k が削除でき, 図2(c)となる. このように置換処理を工夫することによってLUT数と結線数をより多く削減することができる. ただし, $f(L_i)$ はSPFD(c_j)及びSPFD(c_k)の条件を満たすと仮定する. 提案法のアルゴリズムを以下に示す.

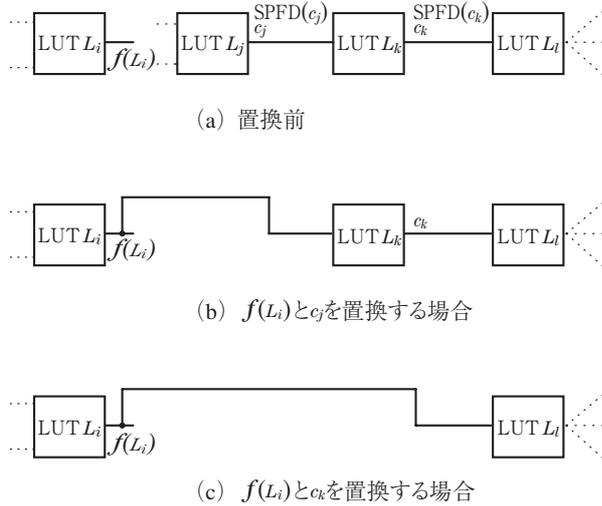


図2：提案法の基本アイデア

[FPGA回路の最適化アルゴリズム]

- Step1**：回路内のすべてのLUTの出力及び結線の論理関数のSPFDを計算する。
- Step2**：回路内から結線を結線番号の降順に1本ずつ選択し、 c_i とする。
- Step3**：すべての結線に対して処理済みなら終了。未処理ならStep4へ行く。
- Step4**： c_i のSPFDが空集合なら結線 c_i を削除してStep7へ行く。空集合でないならStep5へ行く。
- Step5**：回路内からLUTをLUT番号の昇順に1つずつ選択し、 L_j とする。

- Step6**： L_j に対して、もし $f(L_j)$ がSPFD(c_i)の条件を満たしていれば、 c_i を L_j の出力で置き換えてStep7へ行く。そのような L_j が存在しなければStep2へ行く。
- Step7**：回路の変形によって出力関数とそのSPFDの条件を満たさなくなったノードに対して内部論理を変更してSPFDの条件を満たすようにする。そして、Step2へ行く。

5. 実験結果

提案法のプログラムを、Dell OPTIPLEX GX260 (CPU: Pentium4 2.4GHz, Memory: 512MB) 上のVisual C++ 6.0を用いて作成した。実験にはMCNC多段ベンチマーク回路をSIS[7]により、最大5入力のLUTにマッピングしたものを使用した。また、SPFDの論理関数の表現には当研究室で開発したBDDパッケージを用いた。

表1に20個のベンチマーク回路に対する実験結果を示す。表1においてLUTの削減率は初期回路に対する削減率を表す。表1より提案法は、20回路中10回路において文献[6]より削減率が向上し、平均としては6.1%向上した。また、ベンチマーク回路cm151aは28.5%、muxは18.2%向上した。20回路中残りの10回路においては削減率は同じであった。以上のことから、提案法により、従来法よりLUT数が少ない回路を構成できることが分かり、LUT数の削減において有効であることが分った。

表1：実験結果

回路名	初期回路			文献[6]				提案法の結果			
	LUT数	結線数	段数	LUT数	結線数	段数	削減率 [%]	LUT数	結線数	段数	削減率 [%]
term1	69	313	7	47	207	4	3.9	35	149	4	49.3
sct	21	92	3	20	72	3	4.8	20	71	3	4.8
mux	11	48	5	5	22	2	54.5	3	12	2	72.7
lal	36	161	4	31	119	3	13.9	31	119	3	13.9
i3	38	170	7	10	32	2	73.7	10	32	2	73.7
i4	50	240	6	18	80	2	64.0	14	60	2	72.0
frgl	33	151	9	12	55	2	63.6	10	46	2	69.7
example2	105	515	5	100	426	4	4.8	100	424	4	4.8
cu	19	85	4	18	71	3	5.3	17	66	2	10.5
cordic	17	78	8	8	36	2	52.9	6	27	2	64.7
cmb	12	54	5	10	40	3	16.7	10	40	3	16.7
cm162a	12	55	5	10	43	3	16.7	10	43	3	16.7
cm152a	5	22	3	4	17	3	20.0	4	17	3	20.0
cm151a	7	28	4	4	12	2	42.9	2	6	2	71.4
cm150a	10	43	4	6	24	3	40.0	4	18	3	60.0
c8	30	141	3	28	114	2	6.7	28	114	2	6.7
apex7	73	328	6	72	287	6	1.4	70	279	6	4.1
x1	111	454	6	74	289	3	33.3	69	268	3	37.8
x2	13	63	3	12	51	2	7.7	12	51	2	7.7
tft2	53	258	4	48	214	4	9.4	48	214	3	9.4
平均	36.25	164.95	5.05	26.85	110.55	2.90	28.21	25.15	102.80	2.80	34.3

6. まとめ

本論文ではSPFDを用いたLUT型FPGAの最適化の手法を新しく提案し、ベンチマーク回路に対して適用した。その結果、提案法により従来より多くの回路でLUT数、結線数が削減され、本手法の有効性を確認した。今回提案した手法のLUTと結線の選択法は、外部入力側と外部出力側からそれぞれ走査を開始する方法であった。しかし、LUTや結線の選択パターンはこれ以外にも数多く存在し、さらに別の選択法による方法を試みる必要がある。また、今後、大規模なベンチマーク回路に適用する必要がある。

謝辞

実験に用いたプログラムの作成に御協力頂いた平成15年度卒研究生の酒田理人君（現在、東京大学大学院在学中）、平成16年度卒研究生の古川修平君（現在、豊橋技術科学大学大学院在学中）に深謝致します。

参考文献

- [1]身次 茂, “FPGAの現状と将来”, 情報処理, Vol. 35, No. 6, pp.505-510, 1994.
- [2]末吉 敏則, 天野英晴 編著, リコンフィギャラブルシステムFP, オーム社, 2005.
- [3]S. Muroga, Y. Kambayashi, H. C. Lai and N. Culliney, “The transduction method design of logic networks based on permissible functions”, IEEE Trans. Computers, vol. 38, pp.1404-1424, 1989.
- [4]古賀 一博, 澤田 直, 安浦 寛人, “トランスダクション法を利用した組み合わせ回路の消費電力設計について”, 情報処理学会DAシンポジウム論文集, pp.105-110, 1997.
- [5]K. Seko, H. Tanaka and T. Kikuno, “Reducing power dissipation of CMOS circuits by improved transduction method”, Proc. ITC-CSCC '01, pp.720-723.
- [6]山下 茂, 澤田 宏, 名古屋 彰, “SPFD: 論理関数の自由度の新しい表現方法”, 電子情報通信学会論文誌 (A), Vol. J82-A, No. 7, pp.1047-1056, 1999.
- [7]E.M.Sentovich, et al. “SIS: A System for Sequential Circuit Synthesis”, Memorandum No. UCB/ERL M92/41, University of California, 1992.