

# 一般化並列カウンタ追加による コンプレッサツリーの効率的FPGA実装

Efficient FPGA Implementation of Compressor Tree by Adding Generalized Parallel Counters

野田 麦<sup>1</sup>  
Mugi Noda

石浦 菜岐佐<sup>2</sup>  
Nagisa Ishiura

関西学院大学 理工学部<sup>1</sup>  
School of Science and Technology, Kwansai Gakuin Univ.

関西学院大学 工学部<sup>2</sup>  
School of Engineering, Kwansai Gakuin Univ.

## 1 はじめに

コンプレッサ (compressor) は複数の2進整数を加算する回路であり、乗算等の様々な算術演算の回路構成に用いられる。コンプレッサの効率的なFPGA実装法として、全加算器を拡張した一般化並列カウンタ (Generalized Parallel Counter; GPC) により樹状の回路を構成する方法 [1] が提案されているが、この方法では GPC に使用されない入出力が発生することがある。本稿では、そのような入出力を削除した GPC を追加することにより、回路規模とクリティカルパス遅延の削減を図る方法を提案する。さらに、ソルバーを効率的に実行することにより求解の高速化を図る。

## 2 GPC を用いたコンプレッサツリー構成問題

GPC  $(p_{m-1}, p_{m-2}, \dots, p_0; q)$  は、 $2^i$  の重みを持つ  $p_i$  個のビット ( $0 \leq i < m$ ) の重み和を  $q$  ビットで出力する回路である。例えば、全加算器は GPC (3;2) である。GPC は図 1 のように1つのビットをドットで表した図で表現される。GPC はFPGAのLUTやキャリーチェーンを使って効率的に実装することができる。

コンプレッサツリー構成問題は、2進数の個数と各2進数のビット数、GPCの集合を入力として、ステージ数 (GPCの段数) が最小で、回路コスト (LUT数やスライス数等) が最小の回路構成を求める問題である。文献 [1] ではこの問題を整数線形計画問題 (以下 ILP) に定式化して解いている。

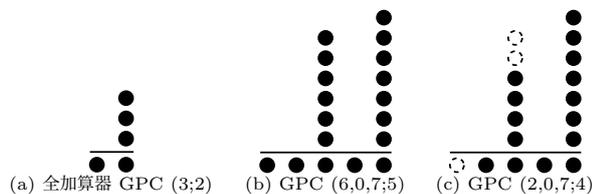


図 1: GPC (一般化並列カウンタ) のドット表記

## 3 提案手法

### 3.1 GPC の追加

文献 [1] の手法では、GPC の全ての入力が使用されるとは限らない。このため、例えば図 1(c) の破線で示したドットにあたる入力が使われなかった場合には、出力1ビットが無駄に使われることになる。破線で示したビットを削除した GPC を予め GPC 集合に追加しておけば、回路中のビット数が減少するため、回路規模やステージ数の削減が期待できる。

### 3.2 ソルバーの効率的実行

最小ステージ数未満での実行不能判定や、最小ステージ数での実行可能解の発見は、ILP ソルバーによっては目的関数を設定せずに制約充足可能性問題として解いた方が速いことがある。そこで本稿では、まず制約充足可能性問題を解き、実行可能解が見つければそれを初期解として最小化問題を解くことにより、実行の高速化を図る。

## 4 実験

提案手法に基づいて、 $N$  桁の2進数  $N$  個を入力とするコンプレッサツリー構成問題を解いた。Xilinx Artix-7 をターゲットとし、表 1 に示す GPC 集合を用いた。下3つが本手法に基づき追加したものである。ILP ソルバーには IBM ILOG CPLEX を使い、Ryzen 7 3700X の24スレッドを用いて、7200秒を制限時間とした。得られた回路のステージ数とスライス数を表 2 に示す。提案手法によりいくつかの回路でステージ数やスライス数を削減できた。

表 1: GPC 集合

GPC	slices
(1;1)	0
(3;2)	1
(7;3)	1
(1,5;3)	1
(2,3;3)	1
(6,2,3;5)	1
(6,0,6;5)	1
(6,0,7;5)	2
(6,1,5;5)	1
(1,4,1,5;5)	1
(1,4,0,6;5)	1
(1,4,0,7;5)	2
(1,3,2,5;5)	1
(1,3,4,3;5)	1
(2,1,3,5;5)	1
(1,3,5;5)	1
(2,2,3;4)	1
(2,0,7;4)	1
(2,1,5;4)	1

表 2: 結果

N	提案手法		文献 [1] の手法	
	stages	slices	stages	slices
15	3	29	3	29
16	3	33	3	33
17	3	38	3	38
18	3	43	3	43
19	3	48	3	48
20	3	55	3	55
21	3	60	3	60
22	3	<b>67</b>	3	70
23	3	<b>77</b>	3	79
24	3	<b>83</b>	3	84
25	<b>3</b>	107	4	85
26	4	<b>92</b>	4	93
27	4	101	4	101
28	4	108	4	108
29	4	116	4	<b>115</b>
30	4	125	4	125
31	4	134	4	<b>133</b>
32	4	142	4	142

## 5 むすび

本稿では、使用されない入出力を削除した GPC の追加、および ILP ソルバーの効率的な実行による、コンプレッサツリーの効率的 FPGA 実装方法を提案した。本手法はニューラルネットワークの FPGA 実装にも有用であると考えられる [2]。

## 参考文献

- [1] M. Kumm and J. Kappauf: "Advanced Compressor Tree Synthesis for FPGAs," *IEEE Trans. Computers*, vol. 67, no. 8, pp. 1078–1091 (Aug. 2018).
- [2] 谷川, 野田, 石浦: "一般化並列カウンタ木に基づく2値化ニューラルネットワークの効率的FPGA実装," 信学技報 (VLD) (Jan. 2023).