

# 一般化並列カウンタ (6,0,7;5) による 多入力加算器の効率的FPGA実装

Efficient FPGA Implementation of Multiple-Input Adders Using Generalized Parallel Counter (6,0,7;5)

野田 麦<sup>1</sup>      叶 亮<sup>2</sup>      石浦 菜岐佐<sup>2</sup>  
Mugi Noda      Ryo Kanai      Nagisa Ishiura

関西学院大学 大学院理工学研究科<sup>1</sup>  
Grad. School of Sci. and Tech, 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

多入力の加算回路は、乗算や積和演算等の様々な算術演算回路の構成に用いられる。多入力加算器の効率的な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) の木を構成する方法が提案されている [1]。しかし、この方法は最適な回路構成を求めるために多大な計算時間を要する。本稿では、GPC(6,0,7;5) を利用した 6 入力 2 出力の加算器を樹状に接続する構成により、多入力加算器の効率的なFPGA実装を短時間で求める手法を提案する。

## 2 多入力加算器の GPC による構成

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。GPC は FPGA の LUT とキャリーロジックを用いて効率よく実装できる。

GPC を用いて  $m$  個の  $n$  ビット 2 進数の和を求める多入力加算器の構成 [1] を図 1(a) に示す。Wallace 木 [2] が全加算器の木を構成するのに対し、GPC の木により  $m$  個の数を 2 個にまで加算し、最後に 2 入力 1 出力の加算器でその和を求める。文献 [1] では、与えられた  $m$  と  $n$ 、および使用可能な GPC の集合に対して、GPC の段数が最小でかつ回路コスト (LUT 数やスライス数) が最小となる回路構成を線形計画問題により求めているが、 $m$  と  $n$  が大きくなると求解に膨大な時間を要する。

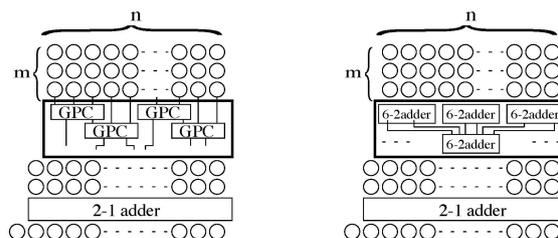
## 3 提案手法

本稿では、図 1(b) に示すように、図 1(a) における GPC 木の代わりに、GPC を用いた 6 入力 2 出力加算器の木により多入力加算器を構成する手法を提案する。6 入力 2 出力加算器は、図 1(c) に示すように、GPC(6,0,7;5) の縦続接続を 2 つ用いることにより構成できる。

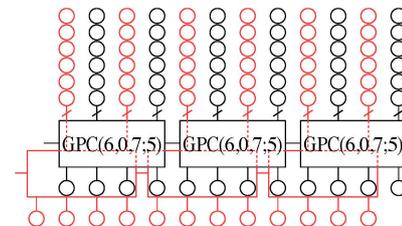
本手法の回路構造は単純であり、従来手法 [1] のように回路構成を決定する最適化問題を解く必要がない。また、GPC(6,0,7;5) は既知の GPC の中でビットの削減効率が最も高いため、回路規模の点でも有効と考えられる。

## 4 実験結果

提案手法に基づいて  $m$  個の  $n$  ビット 2 進数を加算する回路を実装した。Python で生成した Verilog HDL 記述を Xilinx Vivado 2023.2 で Artix-7 をターゲットに論理合成した。得られた回路のスライス数、遅延時間、及び回路構造の決定に要した計算時間を表 1 に示す。従来手法 [1] の回路構造は、図 2 に示す GPC 集合を用いて ILP ソルバー IBM ILOG CPLEX で求めた。求解には AMD Ryzen 9 3900X を用い、3600 秒を制限時間とした。提案手法により回路構造の決定に要する計算時間を大幅に削



(a) GPC 木を用いた多入力加算 [1]      (b) 提案手法



(c) GPC(6,0,7;5) を用いた 6 入力 2 出力加算器

図 1: 多入力加算回路の構成

表 1:  $n$  ビット 2 進数  $m$  個を加算する回路の合成結果

$n$	$m$	従来手法 [1]			提案手法		
		slice	delay (ns)	計算時間 (s)	slice	delay (ns)	計算時間 (s)
12	32	57	11.3	992	58	11.3	0.02
	54	95	12.9	129	91	12.9	0.02
	64	118	12.8	255	113	13.9	0.02
	128	223	16.6	3633	220	16.6	0.03
	162	282	17.6	3623	273	16.7	0.03
	189	328	16.7	3630	323	16.9	0.03
16	32	76	12.9	3626	75	12.4	0.02
	54	130	13.1	3629	118	12.6	0.02
	64	150	15.1	3621	146	14.0	0.02
	128	297	17.1	3623	285	16.0	0.03
	162	377	17.2	3621	354	17.2	0.03
	185	432	18.2	3618	408	17.4	0.03

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

図 2: 従来手法 [1] の求解に用いた GPC 集合

減できた。また、表 1 の平均で、回路規模を 3.5%、遅延時間を 1.8%削減できた。

## 5 むすび

本稿では、GPC(6,0,7;5) による多入力加算器の効率的なFPGA実装法を提案した。今後の課題としては、従来手法 [1] との組み合わせにより、さらに効率的な多入力加算回路を構成することが挙げられる。

## 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] S. C. Wallace: "A suggestion for a fast multiplier," *IEEE Trans. Electronic Computer*, vol. EC-13, issue 1 (Feb. 1964).