Letter to the Editor

Comment on “High Efficiency Generalized Parallel Counters for Look-Up Table Based FPGAs”

Martin Kumm and Peter Zipf

Digital Technology Group, University of Kassel, 34121 Kassel, Germany

Correspondence should be addressed to Martin Kumm; kumm@uni-kassel.de

Received 15 March 2016; Accepted 14 July 2016

Academic Editor: Martin Margala

Copyright © 2016 M. Kumm and P. Zipf. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This brief points out some problems when mapping the optimized GPCs using the heuristic of the paper above. A thorough analysis revealed that a significant number of additional LUTs are required to route the signals when mapping the optimized designs on current FPGAs. Taking these resources into account, the optimized GPCs require at least the same resources as previous state of the art.

1. Introduction

In a recent paper from Khurshid and Mir [1], a heuristic is presented to optimize the mapping of generalized parallel counters (GPCs), as used in compressor trees, to the look-up tables (LUTs) of field programmable gate arrays (FPGAs). The authors claim in their results that the optimized GPCs provide a significant reduction in LUTs compared to previously proposed GPC mappings of our group [2, 3] and other groups [4, 5]. However, as pointed out in this brief, their optimized designs require a significant number of additional LUTs to route the signals to the used resources.

2. Problems When Mapping to Xilinx FPGAs

To illustrate the mapping problems, a (1,4,1,5;5) GPC is used, which was also used in [1] as a detailed example. Their optimization result for a Xilinx Virtex 5 FPGA is shown in Figure 1(a). The FPGA mapping of our GPC [2] is shown in Figure 1(b). In Figure 1(b), a simplified Slice with all relevant routing multiplexers is used. The authors in [1] claim to reduce the LUT resources from four LUTs to two LUTs. However, they did not consider the fact that the LUTs and the fast carry chain cannot be connected arbitrarily. They are organized in a Slice which provides a limited set of multiplexers for internal signal routing [6]. Hence, the only way to route all four outputs of the two LUTs in

Figure 1(a) to the carry chain is by using additional LUTs. These are listed as “route-thru LUTs” in the Xilinx reports and cannot be ignored when comparing complexity as they cannot be used for any other logic. Figure 1(c) shows the best mapping to a Xilinx Virtex 5 Slice we found for the GPC of Figure 1(a). It can be seen that the leftmost and the rightmost LUTs are required to route the results of the two LUTs in the middle to the corresponding inputs of the carry chain. Even if only a fraction of these LUTs is used, they cannot implement any additional logic as either both LUT outputs or the corresponding Slice outputs are occupied, leading to an overall LUT cost of four, the same as previously reported [2].

Looking at the delay, the GPC delay can be broken down into the delays of LUTs, carry chain sections, and local routing, which are denoted as T_L , T_{CC} , and T_R , respectively [1, 2]. The critical path of the GPC in Figure 1(c) starts at the second right LUT and runs along a local routing through the rightmost LUT and three sections of the carry chain. This leads to a delay of $2T_L + T_R + 4T_{CC}$. Assuming $T_L \approx T_R \ll T_{CC}$ [2], this is about three times slower than the delay of the GPC in [2] which is $T_L + 4T_{CC}$. Table 1 lists the results of [1], the corrected results when considering the routing restrictions as well as the best design from the literature. The results of GPCs (3;2), (6;3), and (5,3;4) are not listed as they were correct in [1] but did not show any improvement compared to previous designs. For GPGs (7;3), (5,0,6;5), and (2,0,4,5;5), no mapping for Xilinx FPGAs was provided in [1], so they

TABLE 1: Corrected mapping results of GPCs in [1] when using Xilinx Virtex 5 FPGAs (corrected values are marked in bold).

GPC	Results claimed in [1]			Corrected results			Ref.	Best design from the literature		
	LUTs	Delay	Efficiency	LUTs	Delay	Efficiency		LUTs	Delay	Efficiency
(1,5;3)	1	$T_L + 2T_{CC}$	3	2	$T_L + 2T_{CC}$	1.5	[2]	2	$T_L + 2T_{CC}$	1.5
(2,3;3)	1	$T_L + 2T_{CC}$	2	2	$T_L + 2T_{CC}$	1	[2]	2	$T_L + 2T_{CC}$	1
(1,6;4)	3	$2T_L + T_R + 3T_{CC}$	1	4	$3T_L + 2T_R + 3T_{CC}$	0.75	[4]	4	$2T_L + T_R + 4T_{CC}$	0.75
(3,5;4)	2	$2T_L + T_R + 3T_{CC}$	1	4	$3T_L + 2T_R + 3T_{CC}$	1	[4]	4	$2T_L + T_R + 4T_{CC}$	1
(4,4;4)	3	$2T_L + T_R + 3T_{CC}$	1.33	4	$2T_L + T_R + 3T_{CC}$	1	[4]	4	$2T_L + T_R + 4T_{CC}$	1
(6,2;4)	2	$2T_L + T_R + 3T_{CC}$	2	4	$3T_L + 2T_R + 3T_{CC}$	1	[2]	3	$2T_L + T_R + 3T_{CC}$	1.33
(1,4,1,5;5)	2	$T_L + 4T_{CC}$	4	4	$2T_L + T_R + 4T_{CC}$	1.5	[2]	4	$T_L + 4T_{CC}$	1.5
(1,4,0,6;5)	4	$T_L + 4T_{CC}$	1.5	5	$2T_L + T_R + 2T_{CC}$	1.2	[2]	4	$T_L + 4T_{CC}$	1.5

FIGURE 1: Mapping of (1,4,1,5;5) GPC to Xilinx Virtex 5.

could not be reproduced. It can be concluded for the Xilinx designs that none of the proposed GPCs could be improved in terms of resources but most of them require a larger delay than previous GPCs.

3. Problems When Mapping to Altera FPGAs

Similar issues occur for the proposed Altera mappings. First, unlike Xilinx Virtex 5 FPGAs, the carry inputs of Altera's

adaptive logic module (ALM) used in Stratix IV cannot be fed from the global routing. The carry chain can only start from the first or fifth ALM of a logic array block (LAB) [7]. Figure 2 shows the solution of the (1,4,1,5;5) GPC as provided in [1]. It is claimed that two LUTs are sufficient to realize the GPC. Note that a LUT in [1] refers to a two-output function or half of an ALM. Hence, to route inputs a_0 and a_1 to the inputs of the first full adder (FA), two additional LUTs are required. Next, the signals b_0 and c_0 are directly connected with an FA.

FIGURE 2: Mapping of (1,4,1,5;5) GPC to Altera Stratix IV from [1], claimed to use two LUTs.

FIGURE 3: Correct mapping of a (1,4,1,5;5) GPC to Altera Stratix IV requiring six LUTs.

Again, this can only be realized by routing through a LUT or by bypassing the LUT which makes the output of the LUT inaccessible. Finally, the carry output from an FA also cannot be routed to the output (unlike Xilinx Virtex 5 FPGAs). Thus, an addition with zero is necessary to access the carry output. Figure 3 shows the best mapping to Stratix IV Altera ALMs we found. It can be seen that three ALMs are required to implement the GPC which corresponds to six LUTs. The same problems occur for the other GPC mappings provided in [1]. As there were no previous results reported for Altera FPGAs, a detailed evaluation of Altera GPCs is omitted.

Competing Interests

The authors declare that they have no competing interests.

References

- [1] B. Khurshid and R. N. Mir, "High efficiency generalized parallel counters for look-up table based FPGAs," *International Journal of Reconfigurable Computing*, vol. 2015, Article ID 518272, 16 pages, 2015.
- [2] M. Kumm and P. Zipf, "Efficient high speed compression trees on xilinx FPGAs," in *Methoden und Beschreibungssprachen zur Modellierung und Verifikation von Schaltungen und Systemen (MBMV)*, pp. 171–182, 2014.
- [3] M. Kumm and P. Zipf, "Pipelined compressor tree optimization using integer linear programming," in *Proceedings of the 24th IEEE International Conference on Field Programmable Logic and Applications (FPL '14)*, pp. 1–8, IEEE, September 2014.
- [4] H. Parandeh-Afshar, A. Neogy, P. Brisk, and P. Ienne, "Compressor tree synthesis on commercial high-performance FPGAs," *ACM Transactions on Reconfigurable Technology and Systems*, vol. 4, no. 4, article 39, 2011.
- [5] H. Parandeh-Afshar, P. Brisk, and P. Ienne, "Efficient synthesis of compressor trees on FPGAs," in *Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC '08)*, pp. 138–143, IEEE, Seoul, South Korea, March 2008.
- [6] *Virtex-5 FPGA User Guide (UG190)*, 2007.
- [7] Altera Corporation, *Stratix IV Device Handbook*, vol. 1, 2012.