Efficient FPGA Implementation of Overclocking Friendly Online Arithmetic Operators

***Abstract*—The abstract goes here.**

I. Introduction

II. Background and Relevant Work

A. Online Arithmetic

III. Digit-parallel Online Adder on FPGAs

A. Related Works

There has been previous works focusing on efficient FPGA implementation of the digit-parallel online adder. Generally the existing approaches can be classified into three types:

1) efficient mapping of the digit-parallel online adder onto sophisticated FPGAs resources [xxx];
2) multiple operands addition by designing compressor trees based on bit counters [xxx];
3) modifying existing FPGA architecture for more efficient online addition [xxx] and for specific applications [xxx].

Type 2 and type 3 are beyond the scope of this paper, as we focus on the general purpose online adder that takes 2 inputs and generate 1 outputs, as shown in Figure xxx. Specifically in type 1, both works took advantage of the built-in carry resources in FPGAs. Conventionally the ASIC implementation of online adders is based on the 4:2 compressors, as shown within the gray background in Figure xxx. However, directly applying this approach in the FPGAs could be less efficient. This is because there is no carry propagation between the 2 full adders within a 4:2 compressor, and the net delay between them can be large. Instead, Kamp et al and Ortiz et al described very similar mapping techniques for online adders with 2 different data representations, respectively. The main idea is to map the logic block within the dotted circle in Fig. 1. In this case, the fast-carry logic in the FPGA can be employed, and the delay between the 2 FAs is largely reduced.

However, we notice the major limitations of both approaches that they only target on FPGAs with 4-input LUTs and 2 LUTs within a logic slice, such as the Xilinx Spartan series and the Altera Cyclone series. This is naturally reasonable because one LB can be mapped to a single slice. Nevertheless, for FPGAs with 6-input LUTs (6-LUT) and 4 LUTs in a slice, such as the Xilinx Virtex series and all Xilinx 7 series FPGAs, directly applying the approaches in [xxx] will result in either resource waste or logic fault. For instance, if 2 LBs are mapped to a slice with 4 LUTs as seen in Fig. 2s, the outputs of the second LB will be faulty because the its carry input cannot be explicitly initialized.

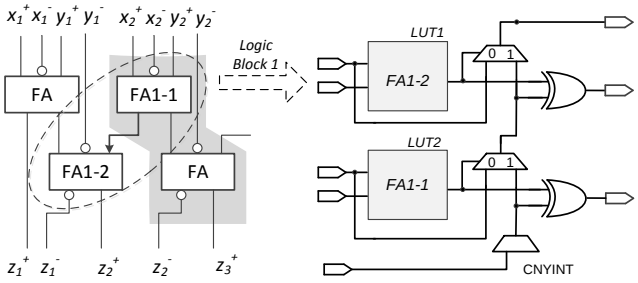

Fig. 1. Map the online adder onto Spartan FPGAs using the fast-carry resources. The grey background highlights the 4:2 compressor. Dotted circle indicates the logic block (LB) which can be mapped to the FPGA carry resources.

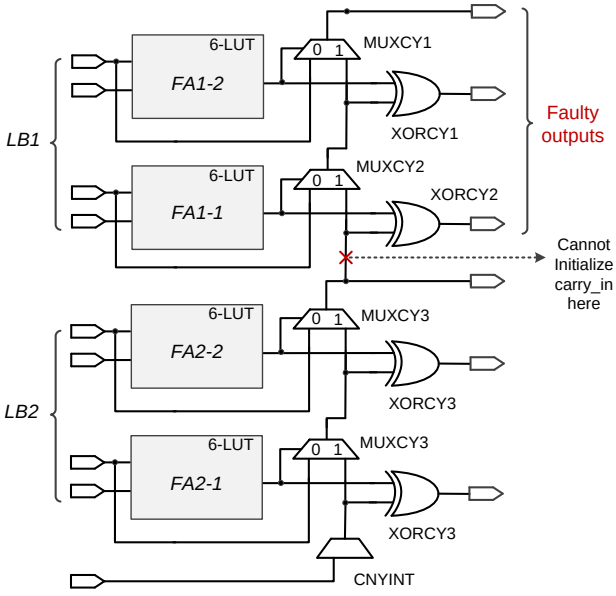

Fig. 2. Direct utilization of previous approaches on a Virtex-6 FPGA will results in faulty outputs.

B. Proposed Mapping Method

To tackle this problem, we first modify the structure of the online adder to enable an efficient FPGA mapping. The structure of a 3-digit online adder is given as an example in Fig. 3. In this equivalent structure, the first FA in each 4:2 compressor is split into 2 parts, which only generate carry and sum respectively. In this case they can be mapped individually on 2 LUTs. The second FA, which generates the outputs, is

unchanged and can be implemented using the fast-carry logic.

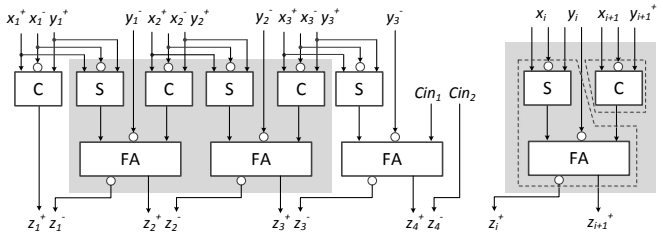

Fig. 3. Modified structure of online adder. Left: an example of 3-digit online adder. The shaded part refers to the 2 logic blocks (LBs) that can be mapped onto 1 slice. Right: one LB. the dotted box outlines the logic that can be mapped onto 1 LUT and the corresponding fast-carry logic.

The detailed slice mapping of the 2 LBs is shown in Fig. 4. The I/O signals are identical to the previous example in Fig. 3. It can be seen that a 6-LUT can be configured with two different output ports O6 and O5. For LB1, the carry input can be initialized by setting the O6 of LUT2 equal to 0 constantly. In this case, the output of MUXCY2 is O5 of LUT2, and the carry from LB2 will not affect the results of LB1. Using this mapping method the resources within 1 slice can be fully utilized, potentially leads to significant area reduction.

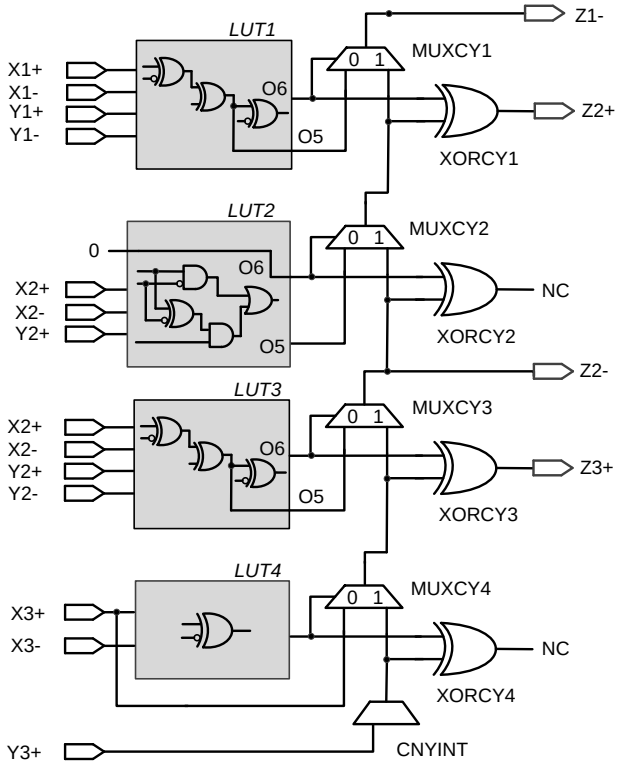

Fig. 4. Implementation of 2 logic blocks (LBs) in 1 FPGA slice which contains four 6-LUTs. NC stands for Not Care.

IV. Digit-parallel Online Multiplier on FPGAs

V. Results

VI. Conclusion

The conclusion goes here.

Acknowledgment