On Implementation and Usage of Muller C-element in FPGA-based Dependable Systems

Petr Pfeifer and H.T.Vierhaus
Brandenburg University of Technology Cottbus-Senftenberg, Faculty of Mathematics, Computer Science, Physics, Electrical Engineering and Information Technology (MINT), Cottbus, Germany
email: {petr.pfeifer | heinrich.vierhaus}@b-tu.de

***Abstract* – Muller C elements are key digital blocks especially used in asynchronous circuits and digital systems and correction of transient (glitch) errors. This paper presents an overview of implementation of Muller C elements in FPGAs. A new scheme is proposed in this paper and the impact of LUT input signal selection is discussed. Finally, unique measurements and experiments performed in Xilinx FPGAs and improved parameters are presented as well.**

Keywords- Muller C-element, FPGA, Xilinx, Artix

I. Introduction and Motivation

The Muller C-element, also called C-gate or coincident flip-flop, is a digital circuit or logic block widely used especially in asynchronous systems. It has been specified formally in 1955 by American mathematician and computer scientist David E. Muller [1]. Implementation of this element in ASIC (Application Specific Integrated Circuits) is widely discussed in many publications, for example [2,3], and many high-performance and reliable circuits and designs exist. Even when containing or implementing multidomain technologies, most of today's FPGA (Field Programmable Gate Arrays) architectures are not designed and ready for implementation of asynchronous systems. This paper deals especially with the implementation of Muller C-element in Field Programmable Gate Arrays (FPGA). We are especially focused to SRAM-based types and solutions using look-up table (LUT) [4].

This paper is organized as follows: Motivation, details about Xilinx FPGA circuits related to our solution, our new proposed solution, performed experiments, measurements and results.

We have presented a small part of our novel architecture for dependable systems already in [5]. One of the partial solution also gains from asynchronous systems and solutions based on GALS (Globally Asynchronous Locally Synchronous) approach. For this our dependable system, a general element with optional features and a better testability was required as well. It is obvious that the LUTs used in FPGAs do not include Muller C-element keeper functions, nor any similar equivalent circuit. There is some similar circuit available at device IOs, but not internally. If a design requires the Muller C-element capability, then one cannot (or should not) use FPGAs. However, all the designed systems must be tested and new architectures validated somehow. Hence, we are back to FPGA devices, since only FPGAs are still the best and cheapest solutions available for prototyping and first development, before all the final design is moved to the final ASIC solutions, utilizing dedicated technologies and especially designed circuits in large libraries. Doing our best, we did not find any paper dealing in detail with the issue of the best selection and circuits of LUT-based Muller C-elements in modern FPGAs.

II. Muller C-elements in General

A great overview over many issues in design and application of Muller C-elements can be found in [6]. Table 1, based on [7], shows the state truth table for a generalized Muller C element:

Table 1: Muller C-element state table

Inputs A, B, C, … N	Output Q
0 0 0 0 0 … 0	0
1 1 1 1 1 … 1	1
Other combinations	previous Q

The following figure 1 shows a general implementation of a Muller C-element using a 2-input multiplexer and std. IEC 60617-12 for gate symbols. The upper AND-gate has a logic one at its output only and only when all inputs i_1 to i_n are set to one. In this case, the following multiplexer selects its input 1, which is routed from the OR gate, and hence having one at its output until at least one of all inputs i_1 to i_n is set to one. In other words, OUTPUT tends toward the value of all the inputs whenever they agree.

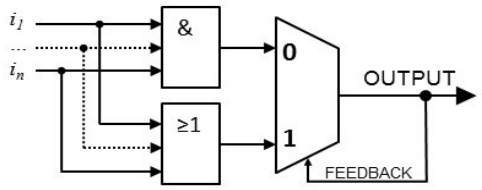

Figure 1. An implementation of Muller C-element using a MUX.

Obviously, all the three parts can also be implemented into one single component in an FPGA. This component can be a single LUT (look-up table). However, FPGAs are not best suited for implementation of asynchronous systems. Hence also the hazard behaviour of the elements in FPGAs should be discussed. Regarding this and other issues discussed further, [8] has focused especially on implementations using FPGAs and issues linked to multiplexers and signal paths. Since also metastability is a critical part in all asynchronous or GALS systems and those issues must be addressed and solved by following many complicated and key design rules, work presented in [9] can give a significant help. Other work [10] and especially [11] shows a hazard–free implementation of a Muller C-Element. Now we will focus ourselves to Xilinx FPGAs.

ISBN 978-80-261-0642-5, © University of West Bohemia, 2017

Figure 2 shows a SLICE with purely logic LUTs, called SLICEL. The Xilinx FPGA architecture contains also SLICEM structures with RLUTs, used for implementation of distributed memories. Typically, two SLICEL or one SLICEL and one SLICEM form one CLB (Configuration logic block). One can find more information in [12]. Differences between SLICEM and SLICEL are not important for this case, hence we can stay with this simpler SLICEL structure.

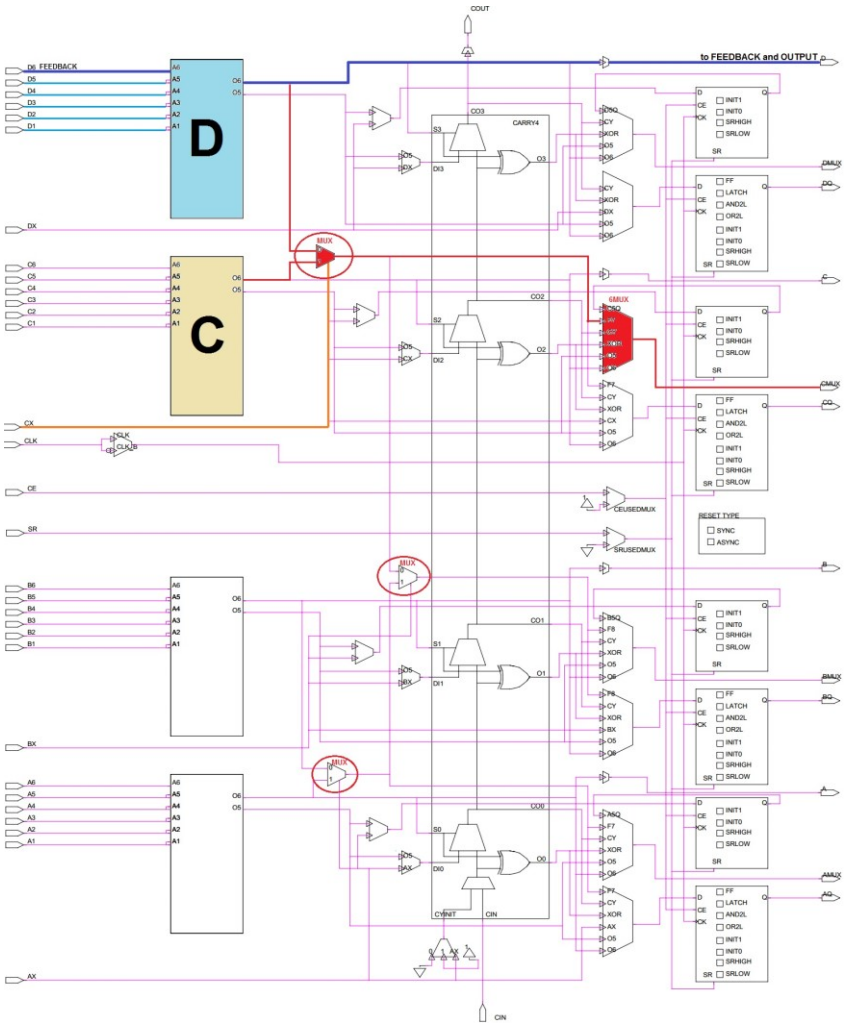

Figure 2. A SLICEL in a Xilinx 28 nm Artix®-7 series FPGA with examples of Muller C-element implementation and highlighted data paths – Modified from ISE14.7 FPGA Editor.

III. Implementation in FPGA using LUTs and one MUX

There are more ways of implementation of Muller C-elements in FPGAs. Some special structures can be found in several special IO blocks in selected FPGA families. Even though new structures were added to the Ultrascale architecture, the very basic LUT style still remains in all FPGAs.

The first implementation is the case, when two LUTs are used for implementation of both the logic gates AND (in for example LUT D) and OR only (for example in LUT C), followed by one of multiplexers labelled MUX. In the FPGA SLICE architecture, there are three controllable multiplexers in total. Other multiplexer can only be controlled by the configuration stream, but not by one of the MUX control auxiliary signals labelled CX, AX or BX. Using multiplexers requires much more LUTs and interconnects resources, because the signals need to be routed twice. There will also be significant timing penalties, since not only one LUT, but also an additional MUX is present in the signal path. This solution has a clear timing disadvantage, but can be used for intentionally slower circuits or for multi-input solutions, where more than five inputs are required.

IV. Implementation in FPGA using a single LUT only

The second solution and implementation of a Muller C-element using one single LUT has clear advantages in saving on the FPGA resources and configuration elements. Figure 3 shows the details of LUTs in Xilinx and the inner real circuit implementation [13]. Each 6-input LUT internally consists of two 5-input patented Xilinx LUTs and one multiplexer. Output O5 is routed to the output of the first LUT. Output O6 is internally multiplexed between LUTA and LUTB, using the last i5 input. The i5 input also corresponds to A6, B6, C6 or D6 SLICE input signal. Now much more similarities to the figure 1 can be seen. We will analyse the impact to the timing parameters of the Muller C-element.

Our new proposed implementation of the advanced Mueller C-element has the following inputs:

- Inputs **A, B, C** - where the output is 0, when all three inputs are set to logic. zero, or the output is 1, when all three inputs are set to logic one; the element keeps its previous output state otherwise.
- Input **ENV** - logic 1 at this input enables the inputs A, B, C.
- Input **INV** - logic 1 at this input inverts the input values.

The last LUT input i5 is routed as a feedback to the LUT output. We have used input A6/i5, because this internal MUX input is a bit faster than other inputs, since routed to the internal dual 5-input LUTs, see [13] in detail.

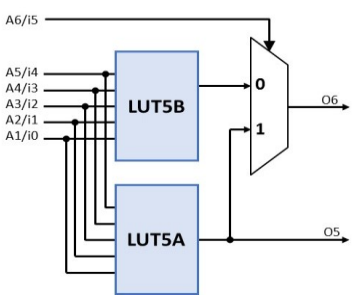

Figure 3. The inner circuits and implement. of 6-input LUTs in Xilinx FPGAs.

V. Implementation

We have implemented the proposed solution in Xilinx 28 nm SoCs and 20 nm FPGAs. We have considered the following Xilinx Artix FPGA and SoC Zynq [14] XC7Z020-1-CLG484 for our experiments, manufactured using TSMC's 28 nm high performance low power (28HPL) process, combining FPGA and an ARM hardware processor into an SoC technology solution, and as used also on the Zedboard development kit [15]. It has a core power supply of 1.0 V, 85K logic cells, 53200 LUTs, 106400 DFFs and 560 KB in Block RAM. Vivado 2016.4 (Build 1733598) with optimization technique set to speed and performance was used. Source codes were generated in VHDL.

Figure 4 shows the schematic of the implemented Mueller C-element in Zynq FPGA. The **LUT content is 64'h7FFFEFF01008000**. Figure 5 shows the exact implementation in the Artix FPGA and Zynq SoC with the feedback route using local interconnects and routing junction box.

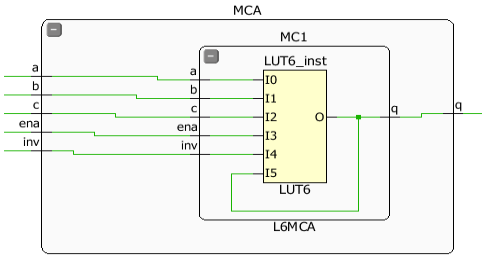

Figure 4. Schematic of the final implemented MC-element in LUT.

VI. Selection of LUT inputs and the Experiment

Up to now there has been no any publication and detailed analysis dealing with suitable selection of LUT inputs for optimal design of the Muller C element. As mentioned above, the feedback is routed to input A6/i5. In order to evaluate the circuit performance, we have created ring oscillators using the proposed elements. There were 29 ring oscillators with 32 chained elements implemented in the FPGA, each having only one (the very first) stage inverted and followed by 31 non-inverted elements, placed one after each other from the left down corner of the die, from LUT A up. There is a 4 SLICE intentional inter-ring space between each two consecutive rings with no overlapped parts. This design ensures maximal circuit isolation and mutual independence of all the test circuits. The rings and circuits were created and all the files generated by A_mBRAM^s tool version 2.1, see [16] for more information.

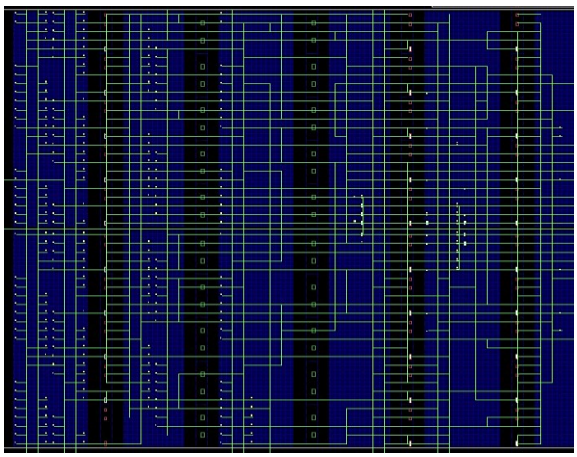

Figure 6. Implementation of test rings in Xilinx Artix and Zynq 7Z020 FPGA and SoC

During the experiment, we ran many ring oscillators on the FPGA. Routing two of them to the inputs A or B and sampling them serves to analyse a complete set of the Muller C-elements using BRAM blocks and A_mBRAMs tools only. Naturally, the output and feedback signal of the Muller C-element is sampled as well. Thanks to this solution and multiple copies with selection of inputs without a need of re-routing, the behaviour of this implementation of Muller C-element can statistically be analysed. The inputs of LUT were changed, BRAM samples were processed, and the impact of each given configuration of input signals was calculated and analysed. All the LUTs under test were fixed in FPGA in their locations by constraints. In order to perform statistical analysis on higher number of circuits, this FPGA is populated by the same circuit in many copies. We have acquired 131072 samples in total. It means that an acceptable error of 0.00000763 (approx. 10^{-5}) corresponds to a one single sample. The following figure 6 shows the placement of all test rings with chained elements in the FPGA. The orange dots represent all the utilized SLICEs and LUTs, green lines represent all the used interconnect resources. Red-blue objects in the dark vertical area are BRAM blocks used by the A_mBRAM^s tool. There is a small A_mBRAM^s engine placed on the chip as well.

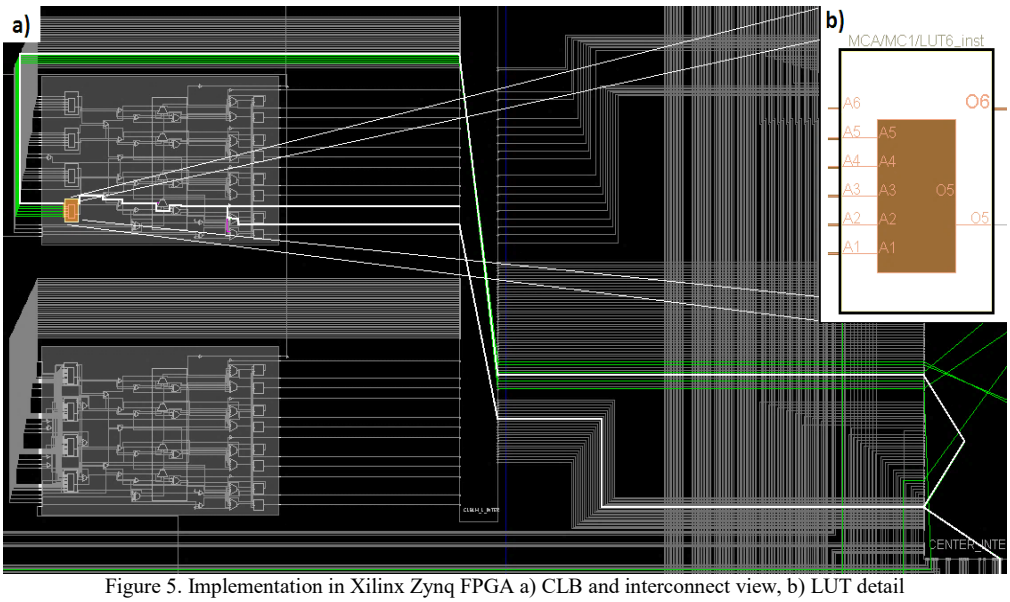

Figure 5. Implementation in Xilinx Zynq FPGA a) CLB and interconnect view, b) LUT detail

VII. Results and Discussion

We have performed sets of 100 measurements during our experiment. The die temperature was 44.1 ± 0.2 °C, core voltage 0.994 V – 0.997 V (0.995 V is the mean value). The minimal calculated delay per one single element (including LUT and interconnects) was 491 ps, the maximal value was 713 ps. The average value was 568 ps per stage or C element, the highest number of values is at 519 ps, median is 574 ps, mode 498 ps. Reported duty factors show that a logic zero at the ring outputs is present with a probability of 0.2 % (in average) higher than logic one. The duty factors were from 47 to 53 % pointing to small variances in internal gate thresholds. Figure 7 shows the distribution of the measured average delays. It is not completely in the Gaussian way, at least two subgroups are clearly visible, obviously caused by different types of interconnect paths utilized. A slightly moved shape is obviously caused by self-heating and die temperature change during the experiment.

Table 2: Muller C-element and LUT input selection

Scheme/LUT input	Probability of signal equality
i0	0.5254 (+0.4%)
i1	0.5234 (0.0%, reference input)
i2	0.5230 (-0.076%)
Sampled inputs	**0.5987** (0.0%, calculated refer. output)
Feedback = LUT i5	**0.5986** (-0.006%, 8 samples difference)
Feedback = LUT i4	0.5974 (-0.21%, 172 samples difference)

Table 2 shows one clear fact that routing the feedback signal to *A6/i5* input is very close to the probability calculated on raw data. It means that the difference is 6.37e-5 (0.0064%), while this one corresponds to max. 8 different samples per all the sampled set of data. This is sufficiently low and significant number to say that Muller C element implementation in LUT with feedback routed to the input *i5* results in a significantly better solution than using the other one for the feedback signal. Using other input may result in more delays of logic 0.

Figure 7 shows that the measurements performed during the experiments confirm the data in datasheets. One has to be sure also about the input signal conditions. Obviously, very short signal peaks below approximately 490 ps may be filtered internally in FPGAs and may not be observed out of the generating circuits. It is due to internal capacitances especially on the connections and distributions path, but also due to signal conditioning circuits implemented in the SLICEM or CLB units. Signals with returning transitions longer than approx. 720 ps will properly be latched by this circuit. This value also limits the overall Muller C-element circuit performance and impacts the performance of all the entire asynchronous system created on base of this implementation and C-element unit.

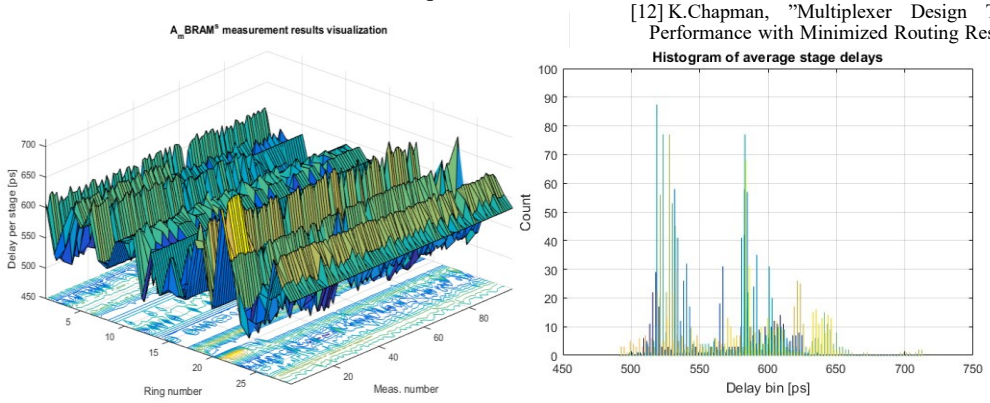

Figure 7. Measurement results using MATLAB toolbox **a)** 3D visualization, **b)** histogram of measured values

VIII. Conclusion

This paper is focused on the area of Muller C-element in asynchronous digital circuits and dependable systems using FPGAs. A new LUT-based solution was presented, being suitable also for many areas of industry applications. This solution was analysed in detail and with respect to the internal real FPGA circuits. Our proposed solution with improved parameters was successfully implemented. All the measurements performed during the experiments and the results clarify the impact of the real FPGA internal circuit structures to the final performance of Muller C-elements. This contribution shows that even the very basic circuits and implementations of the element in FPGAs can easily be improved just by adding the right set of constraints, which can also be generated automatically during the development.

References

[1] D. E. Muller, "Theory of asynchronous circuits. Report no. 66," Digital Computer Laboratory, Univ. Illinois Urbana-Champ., 1955.

[2] Jackson AH, Tyrell AM, "Asynchronous embryonic. ICES 2001," Japan,2001. pp. 88–97.

[3] Nachiketa Das, Pranab Roy, Hafizur Rahaman, "Bridging fault detection in cluster based FPGA by using Muller C element," Computers & Electrical Engineering, Vol. 39, Issue 8, Nov. 2013.

[4] Ian Kuon, Russell Tessier, and Jonathan Rose, "FPGA Architecture: Survey and Challenges", Foundations and Trends in Electronic Design Automation, Volume 2, Issue 2, Feb. 2008.

[5] Petr Pfeifer, H.T. Vierhaus: " A new area-efficient reconfigurable encoder architecture for flexible error detection and correction in dependable communication systems", IEEE BEC2016, Estonia, October 2016, DOI: 10.1109/BEC.2016.7743735

[6] Oleg Petlin, Steve Furber, Designing C-elements for Testability, Technical Report UMCS-95-10-2, Computer Science, University of Manchester, UK, 1995.

[7] Yoshiya Komatsu, Masanori Hariyama, and Michitaka Kameyama , "Area-Efficient Design of Asynchronous Circuits Based on Balsa Framework for Synchronous FPGAs," Int'l Conf. Reconfigurable Systems and Algorithms ERSA'12 , pp 113-118.

[8] L. Fesquet, B. Folco, M. Steiner and M. Renaudin, "State-holding in Look-Up Tables: application to asynchronous logic," 2006 IFIP Internat. Conference on Very Large Scale Integration, Nice, 2006.

[9] Thomas Polzer, Andreas Steininger, „Metastability Characterization for Muller C-Elements," 2013 23rd International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS), Karlsruhe, Sept. 2013, pp.164-171.

[10] Maheswaran, "Implementing Self-Timed Circuits in Field Programmable Gate Arrays," MS Thesis, Univ. Calif. Davis, 1995.

[11] Kapilan Maheswaran, V. Akella, "Hazard-Free Implementation of the Self-Timed Cell Set in a Xilinx FPGA", doi:10.1.1.39.4276

[12] K.Chapman, "Multiplexer Design Techniques for Datapath Performance with Minimized Routing Resources", Xilinx XAPP522, v1.2, Oct. 31, 2014

[13] Xilinx, "7 Series FPGAs Configurable Logic Block", UG474, v1.8, Sept. 2016

[14] Xilinx, DS190 - Zynq-7000 All Programmable SoC Overview, v1.11, 2017

[15] Digilent, "ZedBoard development kit", http://www. zedboard.org/, Product/Zedboard

[16] Petr Pfeifer, "AmBRAMs tools", FPL2015, doi:10.1109/FPL.2015.7293963 and http://www.ambrams.org