Delay Efficient Retiming of Fixed Point Circuits using Brent–Kung adder

Kshipra Tiwari[1], Prof. Sharad Mohan Shrivastava [2]

[1]MTech. Scholar, [2]Research Guide

Department of Electronics and Communication Engineering, SISTEC, Bhopal

Abstract- The expanding complexity of VLSI frameworks and shrinking time to requirements of market require good optimization tools equipped of handling extensive circuits. Retiming is an intense transformation that keeps functionality, and can be utilized to reduce complexity of sequential circuits for an extensive variety of target works by sensibly relocating the memory elements. In a digital circuit retiming is classically based on the propagation delay estimation across critical path in the DFG (data flow graph) circuit. The discrete component timing model fundamentally results higher estimation of the propagation delays. The computation of DFG nodes refers to implementation of fixed-point arithmetic operations such as multiplications and additions. Then again, it is imperative to deal always with the DFGs of such higher granularity at the architecture-level reflection of digital system design for mapping an algorithm to the desired design architecture, where the overestimation of propagation delay leads to undesirable pipelining and bothersome increase in pipeline overheads. This examination proposed a transformation scheme of circuit such a retiming in which registers are added and removed at some point in a circuit in such a manner that functional behavior of proposed circuit remains as it is. To achieve proposed delay efficient architecture of proposed fixed point retiming circuit the Brent–Kung adder architecture is used.

Keywords- Brent–Kung adder, Retiming, Fixed Point Circuits, Delay Efficient Circuit, Data Flow Graph, MCP(Minimum Clock Period).

I. INTRODUCTION

The rapid growth of semiconductor technology has fuelled large scale innovation in the electronic product design space. The electronic gadgets available in the market today are not only useful in their own right but there is intense competition between various players for making them efficient, multi-functional and less expensive. In order to keep up with the increasing competition, the product design houses find themselves in a perpetual cycle of design and product delivery. With every iteration of the cycle, the designers need to work with increased technological complexity while they aim to deliver more value for every unit cost they charge from the consumer.

The advances in the telecommunication industry has been able to draw benefits from the growing semiconductor technology. Smart-phones of the modern day is a very good example for a complex piece of telecommunication, signal processing and semiconductor engineering. The modern day telecommunication standards such as the fourth generation (4G) communication system technology, wireless communication protocols such as the 802.11x are good examples that demand very high computational power. In all these technologies, the implementations are expected to comply with the performance demand of both voice and data traffic. In case of the smart-phone, many such algorithms are implemented on a single device. Moreover, they are also loaded with various multimedia features for recording and playing back music and video in real time. The fact that all this would happen under strict energy, form-factor and time constraints makes it a remarkable piece of engineering.

it may be generalized that the design of any modern day electronic gadget has to be such that the system cost which is usually measured in terms of silicon area, power profile and the execution time is kept to a minimum while not compromising on the system performance expressed in terms of various metrics such as computational accuracy and the response time.

Floating-point and fixed-point operators are two popular choices available for the implementation of all arithmetic operations. Among them, the fixed-point arithmetic operators are known to take significantly lesser area, shorter latency and are known to consume lesser power. Implementation of telecommunication algorithms usually have rigorous performance parameters to be achieved and demand high computational power. In such cases, the use of fixed-point operations is a popular and accepted choice.

Retiming is a powerful transformation that has great potential for sequential circuit optimization. It is the concept of moving storage devices across computation nodes to improve performance without changing the input-output behavior, and can operate at gate level netlists or higher abstrictions (e.g. data flow graphs, communication graphs, processor schedules).

At the circuit level these storage devices are called registers which can be either edgetriggered flip-flops (or FF's) or level sensitive latches (or latches), and the computation nodes are combinational gates. Retiming moves registers across gates without changing the number

of registers in any cycle or on any path from the primary inputs to the primary outputs.

This preserves the input-output latency of the circuit. Since retiming does not directly ciflFect the combinational part of the circuit the circuit behavior remains unchanged. However since retiming can change the boundaries of combinational logic, it has the potential to affect the results of combinational synthesis as well.

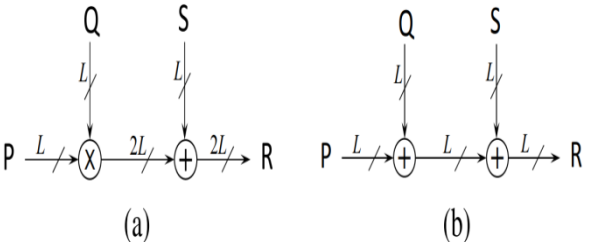

Fig. 1.1 (a) Multiply-add circuit, (b) 3- operand addition circuit.

The DFG in Fig.1(a) represents the computation of $R = P \times Q + S$ and the DFG in Fig.1.1(b) represents the calculation of $R = P \times Q + S$. The word length of each P,Q and S , the propagation delay of the circuit in Fig. 1.1 (a) and Fig. 1.1(b) considered respectively.

$$T_{MA} = T_{MULT} + T_{ADD} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (1.1)$$

$$T_{AA} = 2T_{ADD} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (1.2)$$

Where T_{MULT} *and* T_{ADD} respectively, are the times required for multiplication and addition.

II. BRENT-KUNG ADDER

This is the adder from group “b” or carry look ahead. The main idea of carry look ahead (CLA) is an attempt to generate all incoming carries in parallel and avoid waiting until the correct carry propagates from the first stage. A new Boolean operator which is called “Dot operator or (.)” is introduced as,

$$(G, P).(G', P') = (G + PG', PP') \dots\dots\dots\dots (2.1)$$

The Brent Kung adder computes the prefixes for 2 bit groups. These prefixes are used to find the prefixes for the 4 bit groups, which in turn are used to compute the prefixes for 8 bit groups and so on. These prefixes are then used to compute the carry out of the particular bit stage. These carries will be used along with the Group Propagate of the next stage to compute the Sum bit of that stage. Brent Kung Tree will be using 2log2N - 1 stages. Since we are designing a adder the number of stages will be n. The fanout for each bit stage is limited to 2. The diagram below shows the fanout being minimized and the loading on the further stages being reduced. But while actually implemented the buffers are generally omitted

Brent-kung tree - prefix algorithm

A similar structure with quite different characteristics was presented by Brent and Kung. There, the parallel distribution of intermediate signals from the Sklansky algorithm is replaced by a tree-like and partially serial signal propagation. This almost doubles the number of node delays but reduces the number of black nodes to O(n) and limits the fan-out to $log\, n$ or even to 3, if the maximum fan-out on single rows is regarded (makes sense if white nodes are allowed to contain buffers). Therefore, this prefix structure is regarded.

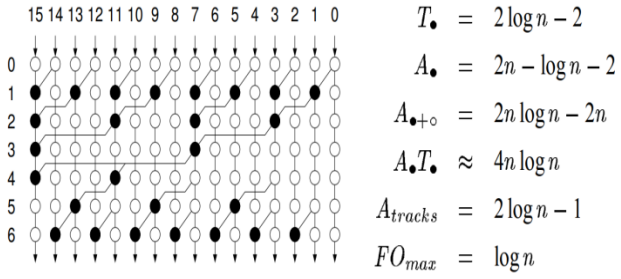

Fig. 2.1 Tree- Prefix algorithm: Bret-Kung.

III. PROPOSED METHODOLOGY

The goal of proposed methodology is to speed up the design of a proposed system without sacrificing the quality of implementation. A common means of achieving this goal is through the use of optimization tools that improve the quality of a quickly designed circuit. This work has shown the optimize clocked circuits by relocating registers so as to reduce combinational rippling. In this examination a delay efficient retiming of fixed point circuits using Brent–Kung adder is reported. The implementation and synthesis of proposed work has completed in Xilinx ISE design suite. Fig. 3.1 Shows the top module of proposed design.

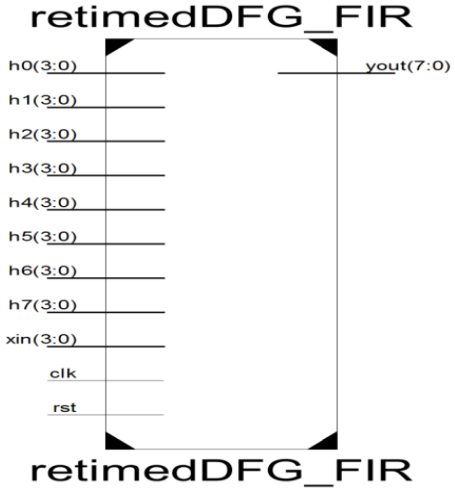

Fig.3.1 RTL Schematic of Top Module.

FPGA's Field Programmable Gate Arrays (FPGA's) present some different requirements. In LUT-based FPGA's the amount of logic is dependent on the number of inputs and not on the complexity of the logic. Further since

FPGA's have limited resources with memory elements at fixed locations extra constraints are placed on the movement of memory elements during retiming. The proposed system is a fixed point retiming circuit which does not change the latency of the system it reduces the critical path of the system using node retiming.

In this work a Brent-Kung adder algorithm is used to make system parallel. The retiming of circuit dose not change the number of delays in each cycle. The retiming in this work is used to minimize clock period and number of registers which results optimization of delay and area.

The retiming algorithm for clock period minimization can be represented as.

$$\omega'(U,V) \geq 0$$

$$\omega'(U,V) \geq r(U) - r(V)$$

Critical Path Constraints

$r(U) - r(V) \leq W(U,V) - 1$ for all vertices U and V in the graph such that $D(U,V) > c$ where c = target clock period. The two quantities $W(U,V)$ and $D(U,V)$ are given as:

$$W(U,V) = min\{w(p): U \longrightarrow V\}$$

$$D(U,V) = max\{t(p) : U \longrightarrow V \; and \; w(p) = W(U,V)$$

Fig. 3.2 shows the RTL schematic of sub module of proposed work where there are 11 sub modules shown in Fig. 3.2 is a internal architecture of top module 3.1. There are 7 input vectors are there in proposed system as shows in Fig. 3.1. h0,h1,h2,h3 and x are input vector with the data width of 4 bit. one clock and one reset pin and an output vector y with 8 bit width.

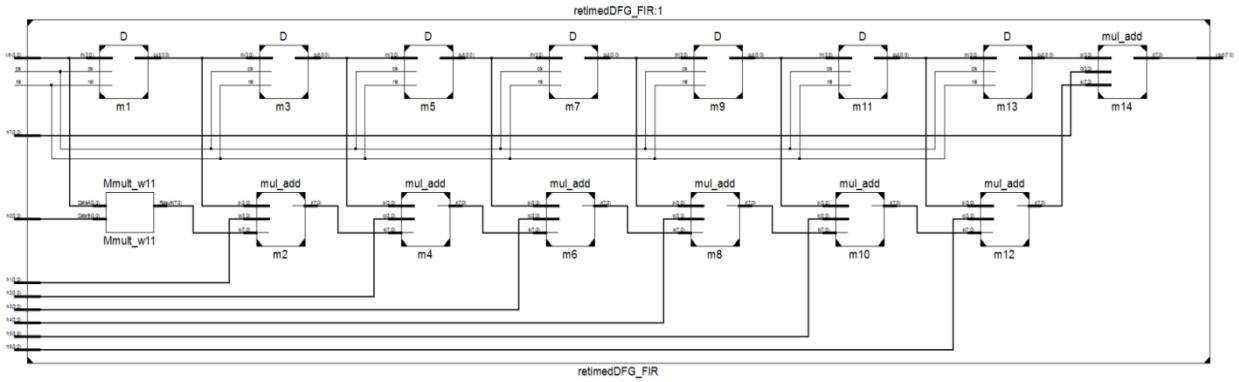

Fig.3.2 RTL Schematic of Sub- Modules.

Steps of Designing Proposed Architecture

Before start designing of any VLSI HDL Architecture for a circuit, one need to create a project in XILINX software and define device family on which synthesis will be performed. After creating a project there is a need to choose language on among Verilog/VHDL.

1. Define top module and its input and output ports with the size in bits

2. Compute the relationship between input and output using assignment statements, registers, internal connecting wires and sub modules.

3. Define sub-modules in Top Module to complete the output requirements and their input and output ports with size in bits.

4. Defining of sub-modules inside may be repetitive, but should have unique identity

5. Define sub-modules and internal functionalities of the Sub-modules

6. Repeat step 5 for all the sub-modules.

7. Check all the syntaxes before synthesis

8. Synthesis top module

9. Analyze the results from synthesis report and power analyzer.

IV. SIMULATION RESULTS

Synthesis of proposed work has completed in Xilinx synthesis technology XST and simulation has performed Xilinx ISIM HDL simulator. Fig. 4.1 shows the Xilinx synthesis screen of proposed work. From the synthesis screen device utilization summary of proposed deign has summarized as number of bounded IOBs are used is 30 out of 400 shows the 7% utilization of device.

Number of BUFG used in the proposed design is 1 out of 32 shows the 3% utilization of device where as number of DSP 48E is 5 out of 1540 shows the 0% utilization of device resources in Fig.4.1 Xilinx synthesis screen.

Timing summary of implementation proposed system has shown in Fig.4.2. In Fig. 4.2 timing analysis of proposed design has tested on speed grade 2, has achieved minimum period of 0.628ns at maximum frequency 1591.72MHZ. Minimum input arrival time before clock: 0.758ns. Maximum output required time after clock: 2.716ns. Maximum combinational path delay: 2.574ns. Table 1 has given results comparison of proposed work with existing

work. Graphical representation of comparison table of proposed work with existing work is given in Fig.4.3.

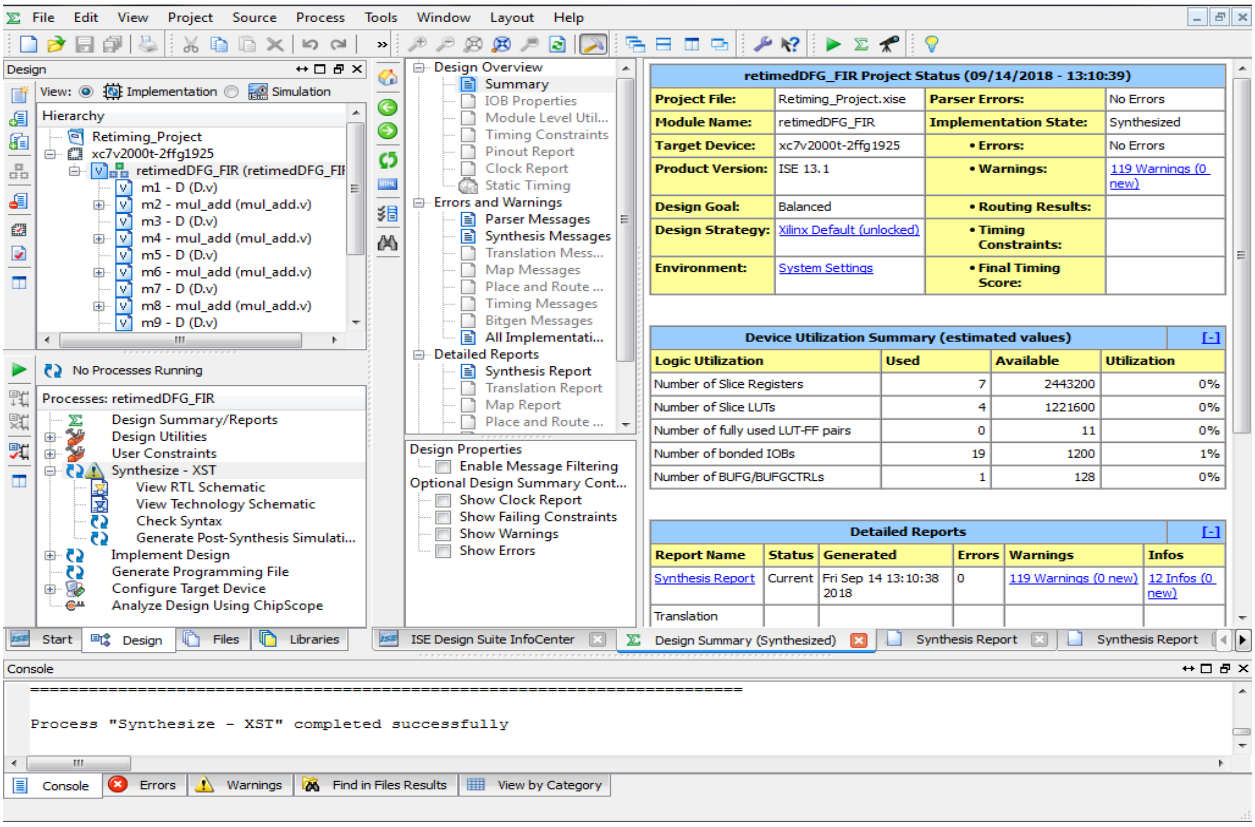

Fig. 4.1 Screen Shots of Summary of Proposed Architecture.

```
Timing Summary:
---------------
Speed Grade: -2
   Minimum period: 0.628ns (Maximum Frequency: 1591.723MHz)
   Minimum input arrival time before clock: 0.758ns
   Maximum output required time after clock: 2.716ns
   Maximum combinational path delay: 2.574ns
Timing Details:
---------------
All values displayed in nanoseconds (ns)
=========================================================================
Timing constraint: Default period analysis for Clock 'clk'
  Clock period: 0.628ns (frequency: 1591.723MHz)
  Total number of paths / destination ports: 6 / 6
-------------------------------------------------------------------------
Delay:               0.628ns (Levels of Logic = 0)
  Source:            m11/out_0 (FF)
  Destination:       m13/out_0 (FF)
  Source Clock:      clk rising
  Destination Clock: clk rising
  Data Path: m11/out_0 to m13/out_0
                              Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FDR:C->Q              2   0.277   0.344  m11/out_0 (m11/out_0)
     FDR:D                     0.007          m13/out_0
    ----------------------------------------
    Total                      0.628ns (0.284ns logic, 0.344ns route)
                                       (45.2% logic, 54.8% route)
```

Fig. 4.2 Timing Summary of Proposed Methodology

Table: 1 Performance Comparison.

Parameter	*Previous [1]*	*Proposed (our)*
Filter Length	8	8
Minimum Clock Period	2.3ns	**0.628ns**

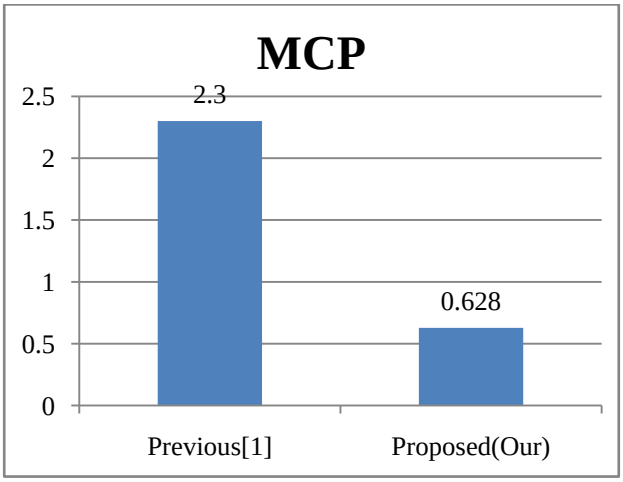

Fig.4.3 Performance Comparison Chart.

Table: 2 Performance on Different FPGA Family.

Parameter	*Virtex 7*	*Spartan 6*	*Kintex 7*
Power	79 mW	14 mW	79 mW
Delay	6.676 ns	19.392 ns	7.886 ns
Min. Clock Period	0.628 ns	1.165 ns	0.740 ns
Frequency	1591.723 MHz	858.185 MHz	1351.35 MHz
Slice Registers	7	7	7
Slice LUTs	4	4	4

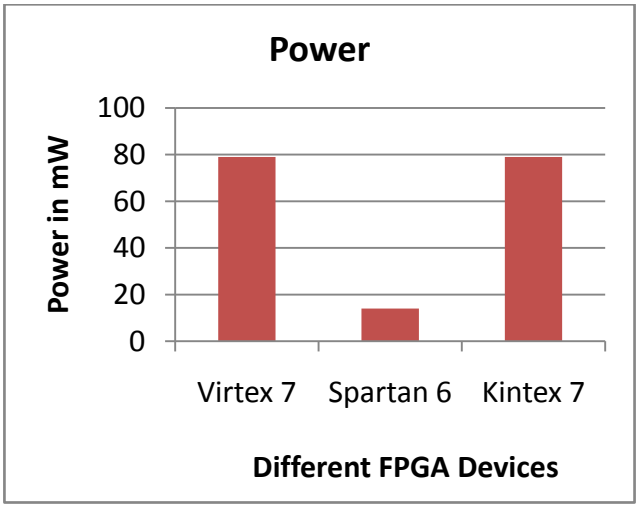

Fig. 4.4 Power Analysis on Different FPGA Devices

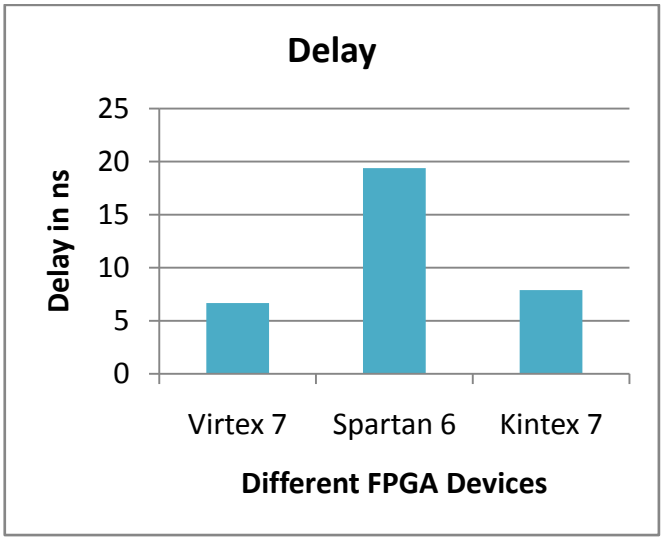

Fig. 4.5 Delay Analysis on Different FPGA Devices

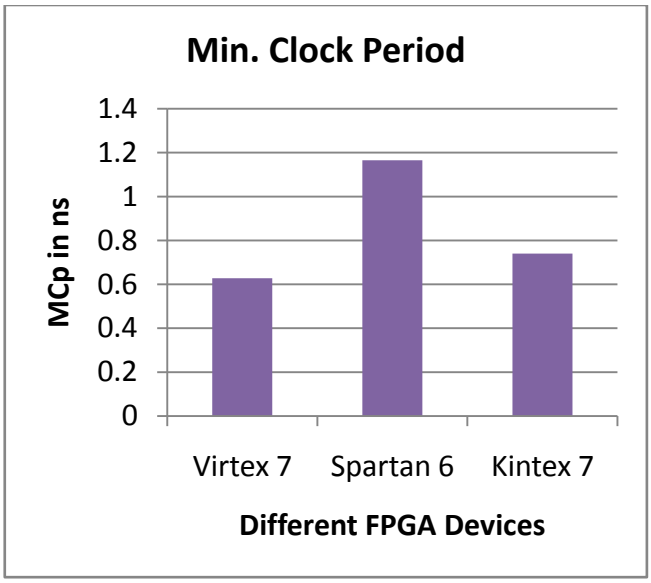

Fig. 4.6 MCP Analysis on Different FPGA Devices

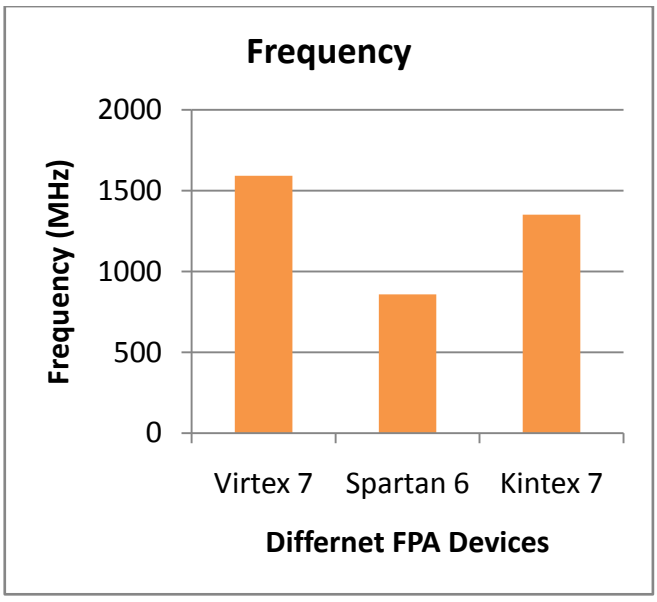

Fig. 4.7 Frequency Analysis on Different FPGA Devices

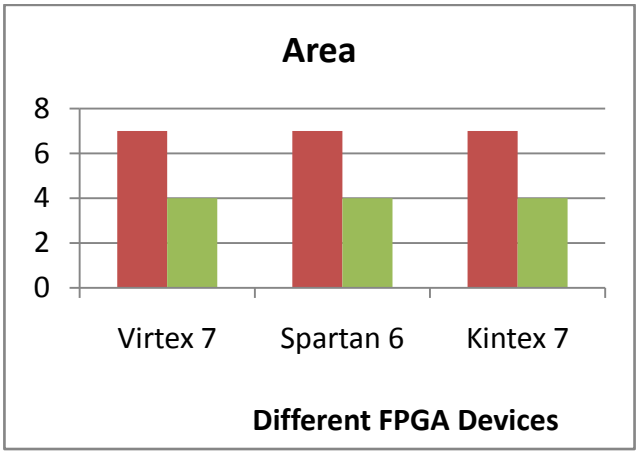

Fig. 4.8 Area Analysis on Different FPGA Devices

V. CONCLUSION

The scalability issues with respect to the fixed-point performance evaluation and delay optimization have been the main drivers. The contributions made in this examination have been able to diligently address them and propose workable solutions to address these issues. In this examination work implementation, synthesis and simulation of delay efficient retiming of fixed point circuits using Brent–Kung adder has completed. The results are obtained by synthesis and timing analysis of proposed retiming of fixed point circuit used Brent–Kung adder shows satisfactory outcome in terms of optimization of circuit delay and complexity of previous methodology. A comparative analysis of proposed work vs previous work has shown with tabular form and graphical form as well. Proposed work shows minimum clock period 0.628 ns which is much less as compared to previous work which was 2.3ns. The retiming is a very straight forward and power full transform widely used for any sequential circuit instead of any other logic optimization scheme. The nature of retiming is sequential which makes it possible to improve results quality of subsequent logic optimization.

The work centrally focuses on the problem of scalability by addressing system-level issues. A lot of work remains to be done to find a practical solution to the retiming verification problem. In this implementation, limitations of the proposed solutions are explored and some of the key assumptions are questioned to define possible scope for future work.

REFERENCES

[1]. P. K. Meher, "On Efficient Retiming of Fixed-Point Circuits," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 24, no. 4, pp. 1257-1265, April 2016.

[2]. H. Kim, Y. Lee and J. H. Kim, "Low-complexity CRC-aided early stopping unit for parallel turbo decoder," in Electronics Letters, vol. 51, no. 21, pp. 1660-1662, 10 8 2015.

[3]. X. Lou, Y. J. Yu and P. K. Meher, "Fine-Grained Critical Path Analysis and Optimization for Area-Time Efficient Realization of Multiple Constant Multiplications," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 62, no. 3, pp. 863-872, March 2015.

[4]. P. K. Meher and S. Y. Park, "Critical-path analysis and low-complexity implementation of the LMS adaptive algorithm," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 61, no. 3, pp. 778-788, Mar. 2014.

[5]. X. Y. Zhu, T. Basten, M. Geilen and S. Stuijk, "Efficient Retiming of Multirate DSP Algorithms," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 31, no. 6, pp. 831-844, June 2012.

[6]. X. Y. Zhu, M. Geilen, T. Basten and S. Stuijk, "Static Rate-Optimal Scheduling of Multirate DSP Algorithms via Retiming and Unfolding," 2012 IEEE 18th Real Time and Embedded Technology and Applications Symposium, Beijing, 2012, pp. 109-118.

[7]. Y. Diao and Y. Wu, "A Fast Retiming Algorithm Integrated with Rewiring for Flip-Flop Reductions," 2011 12th International Conference on Computer-Aided Design and Computer Graphics, Jinan, 2011, pp. 471-477.

[8]. M.R.C.M. Berkelaar. LP.SOLVE User's Manual. Eindhoven University of Technology, Eindhoven, The Netherlands, 1992.

[9]. F. Brglez, D.Bryan, and K. Kozminski. Combinational profiles of sequential benchmark circuits. In Proceedings of the IEEE International Symposium on Circuits and Systems, pages 1929-1934, 1989.

[10]. T. M. Burks and K. A. Sakallah. Optimization of critical paths in circuits with level-sensitive latches. In Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, pages 468-473, 1994.

[11]. T. M. Burks, K. A. Sakallah, and T. N. Mudge. Critical paths in circuits with level- sensitive latches. IEEE Transactions on VLSI Systems, 3(2):273—291, June 1995.