



ZIBELINE INTERNATIONAL

ISSN: 2590-4043 (online)

CODEN : AEMCDV

DOI : <https://doi.org/10.26480/aem.02.2019.31.36>

REVIEW ARTICLE

FPGA SYNTHESIS AND VALIDATION OF PARALLEL PREFIX ADDERS

Qasem Abu Al-Haija^{1,2}, Mohamad Musab Asad³, Ibrahim Marouf¹, Ahmad Bakhuraibah¹, Hesham Enshasy¹¹Department of Electrical Engineering, King Faisal University, Al-Ahsa, 31982, P.O. Box 380, Saudi Arabia²Department of Computer & Information Systems Engineering, Tennessee State University, Nashville, TN, USA³University of Bristol, Department of Electrical & Electronic Engineering, Bristol, BS8 1QU, United Kingdom.*Corresponding Author Email: eng_qasem1982@yahoo.com, qabualha@my.tnstate.edu

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.

ARTICLE DETAILS

ABSTRACT

Article History:

Received 10 June 2019

Accepted 15 July 2019

Available online 1 August 2019

The main objective of this paper is to attain the best achievable time delay reduction with better performance (i.e. frequency) running on FPGA platforms and prove their applicability in high performance reconfigurable computing in addition to evaluate the FPGA design area and thermal power dissipation. The paper presents description on the implementation of five fast radix-2 parallel prefix adders, namely: Ladner-Fischer Adder (LFA), Brent-Kung Adder (BKA), Kogge-Stone Adder (KSA), Hans-Carlson Adder (HCA), and Sklansky Adder (SkA), with variable data path sizes ranging from 8 bits to 64 bits. The PPA topologies were implemented using VHDL description language and synthesized using Altera Cyclone IV E (EP4CE115 F29C7) FPGA chip device. Intensive tests and verifications were conducted and analyzed to validate and evaluate the design cost factors: total path delay time, maximum frequency, design area and the total FPGA thermal power dissipations of the FPGA design as well as the hardware utilization. The results on the code synthesizing demonstrated that the proposed FPGA implementation of KSA has recorded the best values of critical path delay with 4.504 ns for 64 bits while BKA recorded the least design area results with 223 logic elements for the same bit length. In terms of power dissipation, KSA and SkA adders have recorded the best outcomes since they consume the minimum total thermal power dissipation among all other PPAs and for all bit lengths. Thus, the performance of the proposed PPA adders was benchmarked against other state-of-the-art designs which results reflected its superiority in terms of throughput of two or more multiple times as compared to others.

KEYWORDS

FPGA Design, Parallel Prefix Adders (PPAs), Kogge-Stone Adder (KSA), Brent-Kung Adder (BKA), Han-Carlson Adder (HCA), Sklansky Adder (SkA), Ladner-Fischer Adder (LFA), Critical Path Delay, FPGA Synthesize Area, FPGA thermal power dissipation.

1. INTRODUCTION

Recently, the tremendous emergence and growth in computing technologies and electronics set forth intrinsic infrastructure to develop today's embedded systems for different applications ranges from low-power, low-cost design to high performance computing coprocessor design. This growth increasingly contributed in many major areas such as the environmental monitoring, automation, transportation and logistics, healthcare, smart power grids, security and surveillance. An embedded system is one type of several computer systems that are fundamentally designed for specific functions such as storing, processing, and controlling the data in different electronics-based systems. Embedded systems can be found in a variety of common electronic devices such as consumer electronics like cell phones, pagers, digital cameras, portable video games, calculators,etc.

Embedded systems can be considered as computer hardware systems having software embedded in it. They can be either independent systems or parts of other large systems performing particular tasks [1]. Embedded systems are controlled by single or multiple processing cores. One of the most important processing cores is the field-programmable gate arrays (FPGA) [2], which are semiconductors devices that consist of configurable logic blocks (CLBs, also called logic elements LEs) that can be configured and reconfigured after manufacturing [3]. An FPGA logic block can be as simple as a transistor or as complex as a microprocessor, and it is typically capable of implementing many different combinational and sequential logic functions [3]. FPGAs, however, compose a significant revolution in

the field of digital hardware design with more than millions of logic gates and flip-flops embedded in one digital hardware design to enable users to configure and implement several functions.

The most interesting feature of FPGA devices is the fixable configurability since it can be programmed in the field to accomplish specific tasks by using HDLs (Hardware description languages) such as VHDL [4] along with other corresponding CAD packages such as ModelSim simulator, ISE Synthesizer and Quartus II Synthesizer. Because of this flexibility, researchers have adopted FPGAs in distinctive design environment to analyze and verify their systems. Computer arithmetic designers have significantly composed a substantial part of this adoption as can be clearly recognized by the state-of-art designs. These designs are heavily depending on FPGA techniques and device in building efficient embedded and digital signal processors such as cryptographic processors [5].

Computer or digital arithmetic concerns mainly with number systems and their arithmetic operations such as addition, subtraction, multiplication, division, square roots, and others. Almost all arithmetic operations are constructed mainly from addition operation. It contributes in almost every processors task to do complex computations such as the random number generators [6] and computation of greatest common divisor [7]. Also, in building cryptographic coprocessor as in elliptic curve cryptographic (ECC) design [8], low power applications [9] as in the tiny RSA Cryptoprocessor design [10], communication systems as in the MIMO System design in [11] and many other applications. Due to this significant role of adders, researchers of arithmetic computing and processing

proposed extensive alternatives. Striving to replace ordinary slow ripple adders with more efficient fast adders such as carry lookahead adder (CLA) which was the first fast adder to manipulate the carry propagate toward achieving faster performance [12]. Thereafter, more precise adder topologies were differentiated from CLA carry generation stage were proposed to finally obtain the Parallel Prefix Adders (PPAs) which offer distinctive design alternatives to attain area, power, and delay efficiencies and they also provide optimization of the trade-offs [13].

PPAs are implemented in Very Large-Scale Integration (VLSI) chips which rely heavily on fast and reliable arithmetic computation, therefore, PPAs are very useful in today's world of technology [14]. Practically, five PPAs were proposed independently by different researchers based on the distribution of carry propagate and generate signals, they are: Ladner-Fischer Adder (LFA), Brent-Kung Adder (BKA), Kogge-Stone Adder (KSA), Hans-Carlson Adder (HCA), and Sklansky Adder (SkA). Each adder of these is superior in one certain aspect such as area, speed, power, and fan in/out. In this paper, we propose efficient FPGA implementations for the five prefix adders' blocks using Altera Cyclone IV FPGA kit with variable bit-length (8- to- 64 bits). Larger adders can be implemented by cascading several adder blocks in scalable effective connection. Also, we compare between the adders in terms of area (the number of logic elements) and the delay (critical path delay), in addition to the comparison of many states of the art designs. The contributions of this paper can be summarized as follows:

- A state-of-the-art review of various design techniques of parallel prefix adder variations (hardware, software or hybrid).
- Details on the hardware implementation with variable datapath PPA coprocessors using efficient modules, including schematic diagram and cost analysis.
- Comparative performance evaluation and discussion of the FPGA implementation and synthesize results related to area of the design, total delay of the design, minimum delay, maximum frequency and total FPGA thermal power dissipation complexities for various PPAs.
- Comparison of the proposed PPA implementations with many state-of-the-art published works.

The rest of this paper is organized as follows. Section 2 discusses the related works on PPA designs and implementations. Section 3 provides a brief background of PPA arithmetic along with illustration diagrams for hardware specifications. Section 4 contains experimental results with their associated discussions, including performance measures, hardware utilization of the proposed implementation, FPGA total thermal power dissipation and a state-of-the-art benchmarking study. Finally, Section 5 concludes the paper findings.

2. LITERATURE REVIEW

In the last decade, considerable number of hardware solutions have been proposed to address the cost-effective design and performance analysis of various Parallel Prefix Adders topologies. The most commonly used solutions include FPGA implementations and high-level synthesis. Recently, I. Marouf, et.al [15] presented efficient FPGA implementations of five parallel prefix adders namely: Kogge-Stone Adder (KSA), Brent-Kung Adder (BKA), Han-Carlson Adder (HCA), Sklansky Adder (SkA), and Ladner-Fischer Adder (LFA) using Altera Cyclone IV as a target FPGA device. Their experimental results showed that KSA adder recorded the best values of path delay with 4.504 ns for 64 bits while BKA adder recorded the least design area results with 223 logic elements for the same bit length. Also concluded that their proposed adders' implementations have enhanced the performance over many state-of-the-art designs with even more than 200%. Alternatively, Avinash et.al [16] investigated four types of PPA's (Kogge Stone Adder (KSA), Spanning Tree Adder (STA), Brent Kung Adder (BKA) and Sparse Kogge Stone Adder (SKA)) in addition to Ripple Carry Adder (RCA), Carry Look ahead Adder (CLA) and Carry Skip Adder (CSA). They implemented their adders in Verilog Hardware Description Language (HDL) using Xilinx Integrated Software Environment (ISE) 13.2 Design Suite with target chip device Xilinx Spartan 6 Field Programmable Gate Arrays (FPGA). As a result, they concluded that the Kogge Stone Adder has the least delay figures among all other adders

with almost 11.9 ns for 16-bit KSA over the target FPGA Kit.

Also, contributors in [17] proposed a VHDL implementation for eight adders: CLA, KSA, BKA, HCA, LFA, SkA, Sparse Kogge Stone Adder (SKSA), and Carry Save Adder (CSA) using four-bit lengths 4, 8, 16 and 32 bits. They have been categorized and ranked as per delay, device utilization and cell usage. Their simulation results showed that no single type of architecture is the best for all bit values, rather offer enough insights of which type of adders is the best for a specified bit value. They recommended that it would be worthwhile for future FPGA designs to include an optimized carry path, so that tree-based adder designs can be optimized for place and routing. Similarly, authors of [18] investigated the FPGA delay performance of implementing three types of Carry Tree Adders (KSA, SKSA and spanning tree adder -STA) in comparison with Ripple Carry Adder (RCA) and Carry Skip Adder (CSA) targeting the Xilinx Spartan 3E FPGA device with varied bit-widths. They concluded that due to the presence of a fast carry-chain, RCA designs exhibit better delay performance up to 128 bits while carry-tree adders are expected to have a speed advantage over the RCA as bit widths approach 256.

Moreover, Manjunatha and Poornima [19] represented a comparison between the different prefix trees and the implementation of radix4, 32-bit parallel prefix adder with sparseness of 4. The work involves the implementation of radix-2 and radix-4 32-bit parallel prefix adder structures by comparing power, delay, power delay product and number of computational nodes. Their comparison results revealed that radix-4, 32-bit parallel prefix adder consumes minimum power and offer less delay. Cadence Sim Vision tool used for Verilog implementation and Cadence virtuoso tool used for schematic implementation and the Encounter tool is used for the physical design of both adders. Kiran and Srikanth [20], proposed 128-bit Kogge-Stone, Ladner-Fischer, spanning tree parallel prefix adders and compared them with Ripple carry adder. Their simulation results proved that parallel prefix adders are faster and area efficient and they can be used as an efficient technique for increasing the speed in DSP processor while performing addition. Also, they showed that among the three implemented prefix adders, STA has recorded the best performance in terms of area and delay.

Furthermore, Padmajarani and Muralidhar [21] investigated the performance of different 16-bit parallel prefix adders implemented on Xilinx Spartan 3E FPGA in terms of two parameters: speed and area. Their study considered speed performance comparison as the path delay in Nano seconds while they considered area comparison as the number of Look up tables (LUTs), slices and overall gate count. As a result, they found that BKA proves to be a better choice in terms of area or cost while in terms of computational delay or time propagation delay (tpd), KSA recorded the best results among the three PPAs. Indeed, the use of fast two operand adders to calculate the addition operation in a conventional representation system is required as a complementary part of many high radix systems such as the coprocessors developed in [22-24].

In this paper, we are reviewing and evaluating the FPGA designs for five PPAs units (KSA, BKA, LFA, SkA, HCN) in terms of three major design factors: the area of the design (hardware cost in LEs), the time propagation delay (in Nano seconds) and the thermal FPGA power dissipation (in mW). To have better benchmarking results, we have adopted variable precision sizes for the adder blocks (i.e. 8-, 16-, 32-, 64-bits) with target FPGA kit selected to be Altera Cyclone IV FPGA Device: EP4ACE115 F29C7. The higher precisions of the adder unit (i.e. 128-bits, 256-bits...etc.) are still valid and can be implemented by cascading several adder blocks in scalable effective connection. Finally, we have compared our proposed implementations with some previous designs from the literature to show the benefits and enhancement in our FPGA implementations.

3. TAXONOMY OF PARALLEL PREFIX ADDERS

Parallel prefix adders (PPAs) [25] are fast two operands adders that execute addition on parallel manner. PPAs are just like CLA but with an enhancement in the carry propagation stage (called the middle stage). There are five different vibration of PPAs namely: Ladner-Fischer Adder (LFA), Brent-Kung Adder (BKA), Kogge-Stone Adder (KSA), Hans-Carlson Adder (HCA), and Sklansky Adder (SkA). These adders differ by the tree

structure design to optimize certain aspects such as, performance, power, area size, and fan in/out. For instance, KSA utilizes large area size to achieve higher performance comparing to the others, where LFA suffers from large fan out. PPA compute addition in three stages (illustrated in Fig.1.a) as follows:

Pre-processing stage: The computation of generate and propagate of each bit from A and B are done in this step. These signals are given by the logic equations:

$$P_i = A_i \text{ xor } B_i \text{ and } G_i = A_i \text{ and } B_i \quad \dots (1)$$

Carry generation network: PPA differentiates from each other by the connections of the network. It computes carries of each bit by using generate and propagate signals from previous block. In this block two blocks are defined group generation and propagation in addition to group generation only as shown in Fig.1. b. Logic blocks used for the calculation of generate and propagate bits can be describe by the following logic equations:

$$P_{out} = P_{in1} \text{ and } P_{in2} \text{ and } G_{out} = G_{in1} \text{ or } (P_{in1} \text{ and } G_{in2}) \dots (2)$$

Where the generation group have only logic equation for carry generation:

$$G_{out} = G_{in2} \text{ or } (P_{in2} \text{ and } G_{in1}) \dots (3)$$

Post processing (Calculating the Sum): This is the last step and is common to all adders of this family (carry look ahead). It involves computation of sum bits. Sum bits are computed by the logic given below:

$$S_i = P_i \text{ xor } G_{i-1} \dots (4)$$

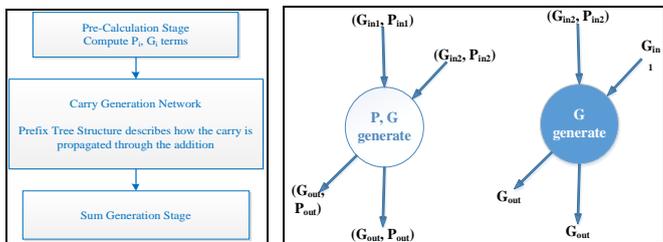


Figure 1: Parallel prefix adder stages. (b) Group generation and propagation.

Parallel prefix structures may be classified with three-dimensional taxonomy (l, f, t) corresponding to the number of logic levels, fan-out, and wiring tracks. For an N bit parallel prefix structure with $L = \log_2(N)$, l, f , and t are integers in the range $[0, L-1]$ indicating [26]: Logic Levels $(L + l)$, Fan-out $(2^f + 1)$, and Wiring Tracks (2^t) . The classic prefix networks include Sklansky, Kogge-Stone, and Brent-Kung, adders. They achieve three extreme cases: minimal logic levels and wire tracks, minimal fan-out and logic levels, and minimal wire tracks and fan-out, respectively. Therefore, they are lying on the vertex of taxonomy triangle as shown in Fig.2. In addition, Ladner-Fischer implemented the tradeoff between each pair of the extreme cases [27].

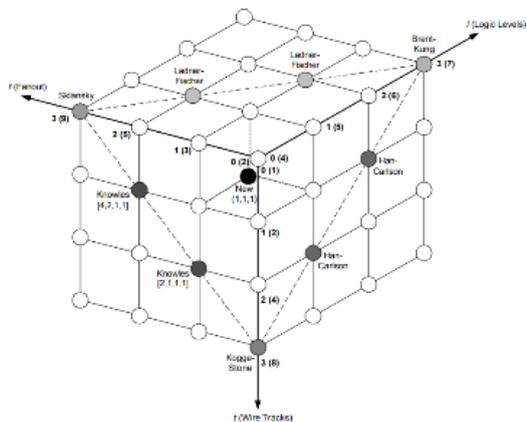


Figure 2: Taxonomy of prefix graphs [26]

The connection of five types of PPA is shown below where the green squares represent group generate and propagate as in equations set (2) and the blue squares represent group generate only as in equation (3). Thus, the most common prefix adders can be summarized as follows:

Brent-Kung Adder (BKA): BKA is a very popular and widely used adder. It avoids explosion of wires for simpler build and less area. In addition, it has a minimum number of fan-out where it is limited to two only. However, it has the highest logic level as shown in Fig.3 for 16-bit BKA. The critical path delay also shown in the figure where it takes 8 levels.

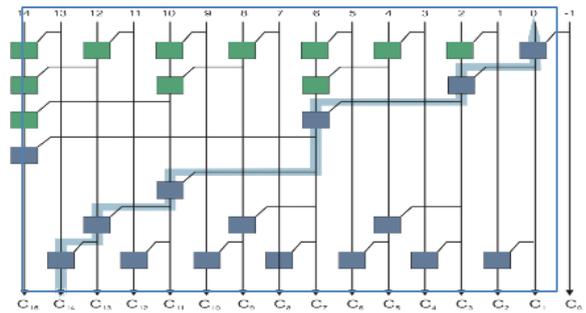


Figure 3: 16-Bit Brent-Kung Adder (BKA)

Sklansky Adder (SkA): The structure of SkA can be viewed as a compacted version of BKA, where logic levels are reduced, and fan-out increased as shown in Fig 4. In this adder, binary tree of propagate and generate cells will first simultaneously generate all the carries, C_{in} . It builds recursively 2-bit adders then 4-bit adders, 8-bit adders, 16-bit adder and so on by abutting each time two smaller adders. The architecture is simple and regular, but it suffers from fan-out problems [28] where it doubles at each level. On the other hand, it has low logic levels and simple wire track connection, but because of its high fan-out it produces large delay at end.

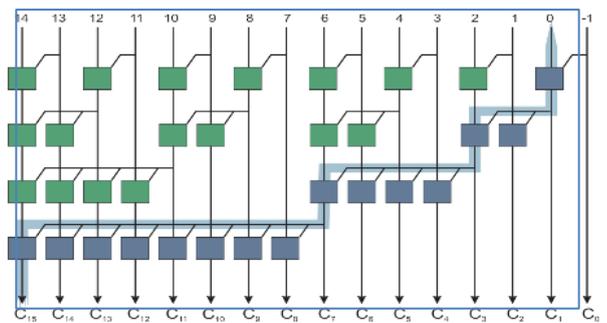


Figure 4: 16-Bit Sklansky Adder (SkA)

Kogge-Stone Adder (KSA): Kogge-Stone Adder is widely used for designing high performance adders. It is a parallel form of carry look ahead adder [29]. It has the minimal logic levels and fan-out which gives faster performance. The structure of KSA is shown in Fig.5 with its critical path delay in blue arrow. However, high wire track results in large area where it uses the largest area among other adders. It is generally considered as the fastest adder [30].

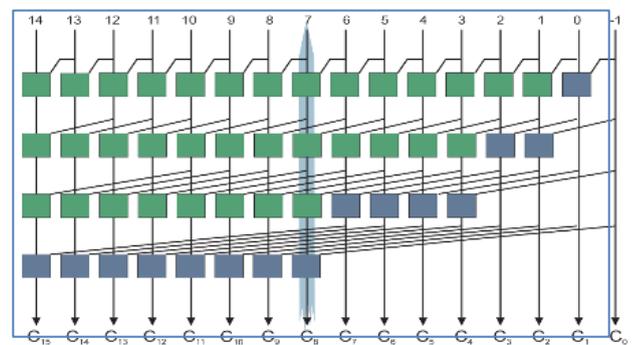


Figure 5: 16-Bit Kogge-Stone Adder (KSA)

Han-Carlson Adder (HCA): It is a mix of Kogge-Stone and Brent-Kung. It has less number of cells and wire tracks as compared to Kogge Stone at the cost of one extra logic level. It trades logical level for wire length [20] as shown in Fig.6. This scheme performs carry-merge operations on even bits only. Generate and propagate signals of odd bits are transmitted down the prefix tree. They recombine with even bits carry signals at the end to produce the true carry bits [31]. Thus, the reduced complexity is at the cost of adding an additional stage to its carry-merge path.

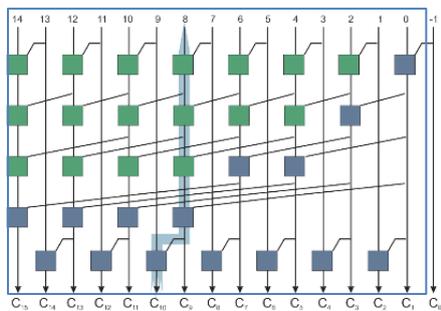


Figure 6: 16-Bit Han-Carlson Adder (HCA)

Ladner-Fischer Adder (LFA): It is hybrid from Brent-Kung and Sklansky, so it compromises logic depth and fan-out. This adder structure has minimum logic depth but has large fan-out requirement up to $n/2$ [32]. The network is shown in Fig.7.

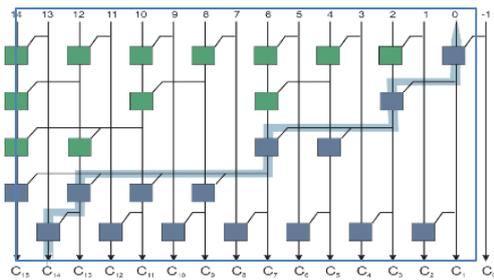


Figure 7: 16-Bit Ladner-Fischer Adder (LFA)

To sum up, all prefix adders share the same procedural stages with major difference in the interconnection networks between group generation and propagation. Table 1 provides a structural comparison between all n -bit parallel prefix adders in terms of three significant design factors: the number of logic levels (i.e. the critical path delay), the area of the design (i.e. the average number of logic gates), the fan-out at for each computation level. The comparison clearly figures out the differences between the PPAs. For instance, the same path delay has been registered for KSA, HCA, and SkA since they have similar number of logic levels, however, SkA adder lowest area requirements for the adder design. The other two adders (i.e. BKA and LFA) are still competitive adders as even though they have the longest path delays, nevertheless, they listed minimum design area with almost $O(n)$ which is much lower than that of KSA and HCA which listed an average design area of $O(n \log n)$. Such factors will be verified in the experimental results section.

Table 1: Comparison between different PPA adders in terms of structural design

Types	Logic Level	Area	Fan-out
Kogge-Stone Adder	$\log_2 n$	$n \log_2 n - n + 1$	2
Brent-Kung Adder	$2 \log_2 n - 1$	$2n - \log_2 n - 2$	2
Ladner-Fischer Adder	$\log_2 n + 1$	$(n/4) \log_2 n + 3n/4 - 1$	$n/4 + 1$
Han-Carlson Adder	$\log_2 n$	$(n/2) \log_2 n$	2
Sklansky Adder	$\log_2 n$	$(n/2) + \log_2 n$	$n/2 + 1$

4. COST FACTORS ANALYSIS AND SYNTHESIZE RESULTS

As alluded before, we have targeted Altera Cyclone IV (EP4CGX-22CF19C7) FPGA device to implement the afore mentioned PPAs using structural VHDL coding as hardware description language along with Altera Quartus II and Modelism-Altera 10.1d for simulation and synthesizing purposes [33]. Each PPA design included four blocks: partial full adder block, a group generation block, a group propagation block, and a sum block to get the last answer. Also, we have used the Altera Quartus II software to accomplish the work in this paper. Furthermore, a high-performance multiprocessor platform has been used in the phases of coding, simulation, and verification as well as synthesizing and tests. The simulation results are given in tables (2-4) along with figures (8-9). The path delay results were generated using TimeQuest timing analyzer tool of Quartus II package with Fast 1200mV OC Model and area estimation results were generated using Analysis and Synthesize tool after Post-Fitting Mapping and port assignments.

Table 2: Comparison between adders in terms of path delay (ns) & design area

Precision	Cost factors	KSA	LFA	BKA	HCA	SkA
8	Delay (ns)	3.187	3.337	3.337	3.200	3.048
	F. Max (MHz)	313.8	299.7	299.7	312.5	328.1
	Area (LEs)	29	22	19	28	24
16	Delay (ns)	3.572	3.680	4.367	4.144	3.463
	F. Max (MHz)	280	271.7	229	241.3	288.8
	Area (LEs)	85	56	50	75	55
32	Delay (ns)	4.280	4.600	4.045	4.322	4.380
	F. Max (MHz)	233.6	217.4	247.2	231.4	228.3
	Area (LEs)	230	125	105	145	130
64	Delay (ns)	4.504	4.855	5.317	4.509	5.259
	F. Max (MHz)	222	206	188.1	221.8	190.2
	Area (LEs)	569	292	223	336	282

Table 2 compares both the path delay (in Nano Second ns- ignoring the interconnection delay), the maximum operational frequency (F_{Max} in MHz) and the total design area (in Number of Logic Elements LEs) between the five parallel prefix adders with variable design length sizes (8, 16-, 32-, and 64-bits). However, the path delay and maximum frequency can both used to express the computational delay/speed for the adder units, therefore, they can be used interchangeably. Also, the relationships between bit length and both cost factors (i.e. delay and area) have been drawn out in Fig.8 (a-b). Fig. 2 obviously shows that BKA recorded the longest propagation delay with 5.317 ns for 64-bit adder followed by LFA for almost all bit sizes where the shortest path delay (i.e. faster adder unit) was related to KSA which listed a delay of 4.504 ns for 64-bit and for SkA with small adder sizes (8- and 16- bits). As for Han Carlson adder which is a hybrid adder between KSA and BKA, the critical path delay becomes closer to KSA delay values as the number of bit increases. On other hand, it's clearly seen that BKA recorded the best area results as it occupies the least number of logic element (LE). BKA required only 223 LEs to implement 64-bit adder unit whereas KSA, HCA, LFA, and SkA are implemented using 569, 336, 292, and 282 LEs, respectively, for the same bit length. Each logic element is consisted of four Look-Up-Tables (LUTs). Thus, KSA implementation costs 2276 LUTs, HCA costs 1344 LUTs and (LFA and SkA) costs about 1148 LUTs. On the other hand, BKA compensates for long path delay with only 892 LUTs and considered as the most costless.

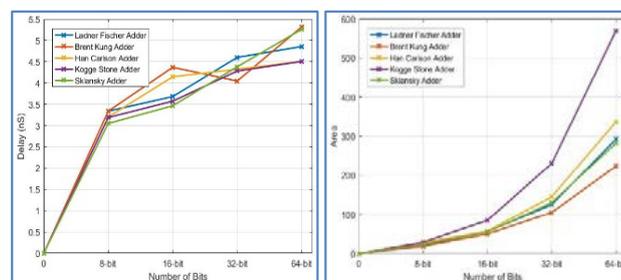


Figure 8: (a) Plot of the path delay in the five adders. (b) Plot of the design area in the five adders

In regard of the comparison with other state-of-art implementations, it is valid, and it shows that our proposed implementation is competitive with many dedicated solutions. For example, the FPGA implementation of KSA in [34] has reported an adder delay of 11.260 ns and 12.840 ns at operand sizes 8-bit and 16-bit, respectively (generated by Xilinx 14.1 Software tool) which are approximately 3.5 times slower than our 8- and 16-bits KSA adders respectively. In related work, Adusumilli and Kumar [35] synthesized their KSA design using Verilog HDL for Xilinx ARTIX-7 and the design tools reported a maximum delay of 8.959 and 10.30 ns of operand size 32-bit and 64-bit, respectively. Instead, our design (using Altera cyclone IV E instead of Xilinx Artix-7) computes the addition in 4.3 ns and 4.5 ns for similar operand sizes which almost double the speed of execution.

Also, Kumari and Nagendra in [36] implemented three different 32-bit PPAs where they got path delay 20.2 ns, 23.2 ns and 22.9 ns for KSA, LFA, and BKA respectively. These designs are slower than their counterparts in our proposed adders by at least a factor of 4. Moreover, authors of [37] simulated their 64-bit KSA using VIRTEX-4 (XC4VFX140 device, 11FF1517

package, -11 speed degree) whereas our chip technology offers c7 speed degree which result in more than 50% of optimization in terms of performance. Furthermore, N. D. Gundi [38] designed a 32-bit BKA using Complementary Pass Transistor Logic (CPL) and CMOS technology and he listed a total delay of 21.427 and 10.650 ns for CPL and CMOS technology which is considered much slower than our 32-bit BKA (four and two times faster respectively). Another comparison can be made with the 64-bit BKA implementation in [39] which recorded a path delay of 13.275 ns which considered 2.5 times slower than counterpart BKA.

In addition, Rani and Kumar [40] implemented five different 64-bit PPAs and they got maximum delays of 17, 18.1, 14.9, 15.1, and 38.2 ns for KSA, BKA, LDA, HCA, and SkA implementations respectively. Their synthesized delay outcomes are much slower than mine (shown in table 2) as our 64-bit SkA adder optimized the delay 700% compared to its peer in [40]. While, Fariddin and Vijay implementations of 8- and 16-bit LFA illustrates an improvement of our LFA implementations by a factor of 3.5 for similar operand size. Finally, Authors in [41] and [42] implemented several types of PPAs with datapath size of 64- and 16-bit where they best delay results are slower than our designs by even more than 400%.

Table 3: Hardware Utilization for 64-bits of different PPAs using Altera Cyclone IV E (EP4CE115 F29C7)

Precision	KSA	BKA	LFA	HCA	SkA
Registers	129 (<1%)	129 (<1%)	129 (<1%)	129 (<1%)	129 (<1%)
LUTs	2276 (1%)	1168 (1%)	892 (1%)	1344 (1%)	1128 (1%)

Table 3 shows the hardware utilization results for PPA coprocessors when implemented using Altera Cyclone IV E (EP4CE115 F29C7) represented by the number of utilized registers (the total number of registers in the target device is 460000) and the number of utilized Lookup Tables - LUTs (the total number of LUTs in the target device is 460000). It is clear that the implementation with different precisions utilizes fewer resources of the target device. The largest implementation length (i.e., 64-bit) utilizes a maximum of almost 1% device registers and LUTs, respectively. This indicates that the implementation area is scalable and can be easily increased or embedded with many other design applications.

Table 4: Total FPGA Thermal Power Dissipation (mW) using Altera Cyclone IV E (EP4CE115 F29C7)

Adders Precision	Core Static Thermal Power Dissipation	I/O Thermal Power Dissipation	Total Thermal Power Dissipation	
Brent-Kung Adder	8	98.47	33.81	132.28
	16	98.48	36.82	135.3
	32	98.54	56.15	154.69
	64	98.63	79.98	178.61
Han Carlson Adder	8	98.48	37.56	136.04
	16	98.5	44.08	142.58
	32	98.55	55.54	154.09
	64	98.57	56.57	155.14
Kogge Stone Adder	8	98.46	30.79	129.25
	16	98.5	44.08	142.58
	32	98.54	51.86	150.4
	64	98.61	61.14	159.75
Sklansky Adder	8	98.46	31.75	130.21
	16	98.49	39.97	138.46
	32	98.55	56.63	155.18
	64	98.58	63.81	162.39
Lidner Fischer Adder	8	98.47	33.81	132.28
	16	98.49	39.81	138.3
	32	98.53	49.55	148.08
	64	98.56	52.92	151.48

Table 4 shows the total FPGA thermal power dissipation (mW) values consumed from applying various PPA schemes with different Datapath lengths (8-bit to 64-bits) to Altera Cyclone IVE (EP4CE115 F29C7) FPGA kit, where:

$$Power_{Total\ FPGA} = Power_{I/O} + Power_{Static}$$

The estimated power results for the design with different precisions from 8-bit to 64-bits were generated using the powerplay early power estimator tool in the Quartus II CAD simulation pack. The power values for the larger designs (i.e., 256, 512 and 1024 bits) can be extrapolated from the general tendency of power figures. The 164-bit precision design can be the largest Datapath that allowed for the power estimation tool due the number of I/O pins provided by the target FPGA kit. The 512 pins cover, two 164-bit inputs, one 164 bits output result, and other pins, are for control signals such as clock, enable, acknowledge and reset. The total FPGA design power is mostly affected by the I/O power, while the term of power is constant (i.e., static power), as articulated in Figure 9.

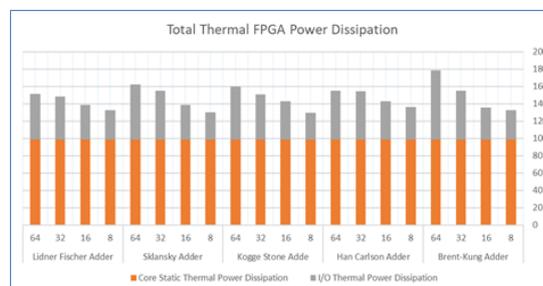


Figure 9: Total FPGA thermal power dissipation (mW)

In short, the synthesise results showed that KSA outperforms the other adders since it has the smallest time delay for all bit lengths. This result is very useful and confirms the theatrical modeling of KSA which has the least number of logic levels. Then, HCA comes second with very close delay values to KSA especially for larger bit lengths. LFA stands in the middle, where BKA and SkA are relatively slow with a long path delay for larger bit lengths. In terms of area size, the equality reverse in which BKA is highly optimized and KSA is not. In terms of power dissipation, KSA adder recorded the best outcomes since it consumes the minimum total thermal power dissipation among all other PPAs and for all bit lengths. SkA is another competitive alternative for KSA in terms of power consumption as it recorded very close figures to that for KSA. To sum up this paper generates attractive synthesized results that can be used to implement a PPA modules using parallel arithmetic units which are useful for many embedded system designs such as those that are commonly used in cryptographic systems over a known finite field. It was found that choosing the best chip technology would increase the throughput of the arithmetic operations [43]. Indeed, according to this results that shows the superiority of KSA adder, we have employed it to compute the final conventional results in every stage of our most recent coprocessor design for the lightweight SSC system developed in [44].

5. CONCLUSION

A comparable study of FPGA implementations for five parallel prefix adders (Kogge Stone, Han Carlson, Brent Kung, Ladner Fisher, and Sklansky Adders) using Altera FPGA devices technology along with four different datapath sizes (8-64bit) to improve the computation process has been investigated thoroughly in this paper. The performance evaluation for these PPAs were studied in terms of time propagation delay (in ns), the maximum execution frequency (in MHz), the area size (in LEs) and the FPGA thermal power dissipation (in mw) in order to compare between the various PPA adders and with the existing implementations/simulations. It was found that KSA was found to be the fastest among the others with minimum amount of total FPGA power dissipation making use of large numbers of LUTs, where BKA is featuring optimization in area size. However, HCA, which is a hybrid adder that constitutes of a good trade-off between speeds (KSA) and low Area (BKA), is proved to be the most adequate adder for achieving high speed closer to KSA with lower area. Finally, the proposed PPA implementations can be synthesized for many other FPGA kits such as Vertex and Spartan kits for the purpose of benchmarking with other kit families.

REFERENCES

- [1] Kamal, R. 2007. Embedded systems: architecture, programming and design. McGraw-Hill Education.
- [2] Maxfield, C. 2004. The Design Warrior's Guide to FPGAs: devices, tools and flows. Mentor Graphics Corporation and Xilinx, Elsevier.

- [3] Rose, J., El Gamal, E., Sangiovanni-Vincentelli, A. 1993. Architecture of field-programmable gate arrays. *Proceedings of the IEEE*, 81 (7), pp. 1013-1029.
- [4] LaMeres, B.J. 2017. *Introduction to Logic Circuits & Logic Design with VHDL*. Electronics & Electrical Engineering, Springer.
- [5] Paar, C., Pelzl, J. 2010. *Understanding Cryptography*. Springer-Verlag Berlin Heidelberg Publisher. <https://doi.org/10.1007/978-3-642-04101-3>.
- [6] Al-Haija, Q.A., Jebri, N.A., AlShua'ibi, A. 2015. Implementing variable length Pseudo Random Number Generator (PRNG) with fixed high frequency (1.44 GHz) via Vertix-7 FPGA family. *Network Security and Communication Engineering*, CRC press, Pp. 105 -108.
- [7] Al-Haija, Q.A., Al-Jafari, M., Smadi, M. 2016. A comparative study up to 1024-bit Euclid's GCD algorithm FPGA implementation & synthesizing. 2016 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA), Ras Al Khaimah, United Arab Emirates, pp. 1-4. <https://doi.org/10.1109/ICEDSA.2016.7818535>.
- [8] Abu Al-Haija, Q., Alkhatib, M., Jaafar, A.B. 2011. Choices on Designing Gf (P) Elliptic Curve Coprocessor Benefiting From Mapping Homogeneous Curves In Parallel Multiplications. *International Journal on Computer Science and Engineering (IJCSE)*, ISSN : 0975-3397 3 (2).
- [9] Sabaghi, M., Marjani, S., Majdabadi, A. 2016. The Design of Ultra-Low Power Adder Cell in 90 and 180 nm CMOS Technology. *Circuits and Systems*, 07,58-67. doi: 10.4236/cs.2016.72007
- [10] Al-Haija, Q.A., Tarayrah, M.A., Al-Qadeeb, H., Al-Lwaimi, A. 2014. A tiny RSA cryptosystem based on Arduino microcontroller useful for small scale networks. *Procedia Comput. Sci.*, 34, 639-646.
- [11] Smadi, M.A., Al-Haija, Q.A. 2014. Precise error rate analysis of MIMO system with interference and imperfect channel state information. *IEEE 27th Canadian Conf. on Electrical and Computer Engineering (CCECE)*, Toronto, ON, pp. 1-5.
- [12] Rosenberger, G.B. 1960. Simultaneous Carry Adder, U.S. Patent 2,966,305, Dec. 27.
- [13] Poornimaa, M., Bhaaskaran, V.S.K. 2015. Area Efficient Hybrid Parallel Prefix Adders, 2nd International Conference on Nanomaterials and Technologies (CNT 2014). *Procedia Materials Science*, Elsevier, 10, Pp. 371-380.
- [14] Raj, D. 2015. Design and Implementation of different types of efficient parallel prefix adders. *National Conference on Advanced Innovation in Engineering and Technology*, 3 (1).
- [15] Marouf, I., Asad, M.M., Bakhuraibah, A., Al-Haija, Q.A. 2017. Cost analysis study of variable parallel prefix adders using altera cyclone IV FPGA kit. 2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA), Ras Al Khaimah, pp. 1-4.
- [16] Shrivastava, A., Churhe, S., Bhagat, H., Wamankar, R. 2017. Design and Estimation of delay, power and area for Parallel prefix adders. 7 (4). (April - 2017), *International Journal of Engineering Research and Applications (IJERA)*, ISSN: 2248-9622, www.ijera.com
- [17] Raju, A., Patnaik, R., Babu, R.K., Mahato, P. 2016. Parallel prefix adders — A comparative study for fastest response. 2016 International Conference on Communication and Electronics Systems (ICCES), Coimbatore, Pp. 1-6.
- [18] Reddy, A.M., Vijayalakshmi, G.V. 2015. Speed Comparison Parallel Prefix Adders with RCA and CSA in FPGA. *International Journal of Innovative Research in Electronics and Communications (IJIREC)* 2 (5), Pp. 8-16. ISSN 2349-4042 (Print) & ISSN 2349-4050 (Online)
- [19] Naik, V.M., Poornima, N. 2015. Performance Analysis of Parallel Prefix Adder. *International Journal of Electrical, Electronics and Data Communication*, 3 (7).
- [20] Kumar, T.K., Srikanth, P. 2014. Design of High Speed 128 bit Parallel Prefix Adders. *Journal of Engineering Research and Applications (IJERA)*, 4 (11).
- [21] Padmajarani, S.V., Muralidhar, M. 2012. Comparison of Parallel Prefix Adders Performance in an FPGA. *International Journal of Engineering Research and Development*, 3 (6), Pp. 62-67.
- [22] Asad, M.M., Marouf, I., Al-Haija, Q.A. 2019. Radix-8 Design Alternatives of Fast Two Operands Interleaved Multiplication with Enhanced Architecture. *International Journal of Advanced Network, Monitoring and Controls*, Exeley publication, 4 (2).
- [23] Al-Haija, Q.A., Marouf, I., Asad, M.M., Nasr, K.A. 2019. Cost-Effective FPGA Implementation Of Parallel Schmidt-Samoa Cryptosystem (SSC). *Journal Of Theoretical and Applied Information Technology*, Little Lion Scientific, 97 (14).
- [24] Al-Haija, Q.A., Al-Zahouri, Y., Ahmed, M. 2010. Efficient New Design and Verification of Sign-Digit-Adder for Two Symmetric Redundant Radix-4 Numbers. *International Journal of Computer Science and Engineering (IJCSE)*, Engg Journal Publications, Singapore, 2 (4).
- [25] Ercegovic, M.D., Lang, T. 2004. *Digital Arithmetic*, Morgan Kaufmann Publishers, Elsevier, 1, Pp. 51-136.
- [26] Harris, D. 2003. A Taxonomy of Parallel Prefix Networks. *Proc. Asilomar Conference of Signals, Systems, and Computers*, Pp. 2213C2217.
- [27] Liu, J., Zhu, Y., Zhu, H., Cheng, C.K., Lillis, J. 2007. Optimum Prefix Adders in a Comprehensive Area, Timing and Power Design Space. 2007 Asia and South Pacific Design Automation Conference, Yokohama, Pp. 609-615.
- [28] Raju, A., Patnaik, R., Babu, R.K., Mahato, P. 2016. Parallel prefix adders — A comparative study for fastest response. 2016 International Conference on Communication and Electronics Systems (ICCES), Coimbatore, pp. 1-6.
- [29] Adusumilli, R., Kumar, K.K. 2015. Design and Implementation of a High Speed 64 Bit Kogge-Stone Adder Using Verilog Hdl. *International Journal of Electrical and Electronic Engineering and Telecommunication (IJEETC)*, 4 (1).
- [30] Chakali, P.K., Patnala, M.K. 2014. Design of High Speed Kogge Stone Based Carry Select Adder. *International Journal of Emerging Science and Engineering (IJESE)*, 1 (4).
- [31] Sudhakar, S.M., Chidambaram, K.P., Swartzlander, E.E. 2012. Hybrid Han-Carlson adder. 2012 IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS), Boise, ID, pp. 818-821.
- [32] Kumari, P.C., Nagendra, R. 2013. Design of 32 bit Parallel Prefix Adders. *IOSR journal of Electronics and communication Engineering (IOSR-JECE)*, 6 (1).
- [33] Altera Corporation. 2017. *Simulation Quick-Start for ModelSim - Intel FPGA Edition*. Intel Quartus Prime Standard Edition UG-01102.
- [34] Kaur, J., Kumar, P. 2014. Analysis of 16 & 32 Bit Kogge Stone Adder Using Xilinx Tool. *JECET*, 3 (3), 1639-1644.
- [35] Adusumilli, R., Kumar, V. 2015. Design and Implementation of a High Speed 64 bit Kogge-Stone Adder using Verilog HDL. *International Journal of Electrical and Electronic Engineering & Telecommunications*, 4 (1).
- [36] Kumari, P., Nagendra, R. 2013. Design of 32 bit Parallel Prefix Adders, *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)*, PP. 01-06.
- [37] Kaul, A., Kumar, A. 2016. Simulation of 64-bit MAC Unit using Kogge Stone Adder and Ancient Indian Mathematics. *Journal of Engineering Research and Application*, 6 (6), pp. 01-05.
- [38] Gundi, N.D. 2008. Implementation of 32 Bit Brent Kung Adder Using Complementary Pass Transistor Logic, Visvesvaraya Technological University.
- [39] Jeevan, V. 2017. Design and Implementation Of 64-Bit Parallel Prefix Brent Kung Adder, *International Journal of Electrical and Electronic Engineering*, 9 (1).
- [40] Rani, G., Kumar, S. 2012. Delay Analysis of Parallel-Prefix Adders. *International Journal of Science and Research (IJSR)*, 6 (6).
- [41] Butchibabu, S., Bab, S.K. 2014. Design and Implementation of Efficient Parallel Prefix Adders on FPGA. *International Journal of Engineering Research & Technology*, 3 (7).
- [42] Raj, D. 2015. Design and Implementation of different types of efficient parallel prefix adders. *National Conference on Advanced Innovation in Engineering and Technology*, 3 (1).
- [43] Al-Haija, Q.A., Ahmad, S.M.S. 2018. Fast Radix-2 Sequential Multiplier Using Kintex-7 FPGA Chip Family. *The Open Cybernetics & Systemics Journal*, 12, Pp. 30-41.
- Al-Haija, Q.A., Marouf, I., Asad, M.M., Nasr, K.A. 2019. Implementing a lightweight Schmidt-Samoa Cryptosystem for Sensory Communications. *International Journal on Smart Sensing and Intelligent Systems*, Exeley Inc. (New York), 12(1).