ISSN: 2395-1680 (ONLINE)
DOI: 10.21917/ijme.2025.0362

KOGGE-STONE ADDER: A SCALABLE DESIGN SOLUTION TO HIGH-THROUGHPUT DIGITAL SYSTEMS

M. Sakthimohan[1] and G. Elizabeth Rani[2]

[1]*Department of Electronics and Communication Engineering, KIT - Kalaignarkarunanidhi Institute of Technology, India*
[2]*Department of Artificial Intelligence and Data Science, KIT - Kalaignarkarunanidhi Institute of Technology, India*

Abstract

The increasing needs of high speed and high throughput computing systems have prompted the need to develop efficient arithmetic units, the most important being adders, which form the heart of digital signal processor, microprocessor as well as any other high performance digital systems. The Kogge-Stone Adder (KSA), one of many architectures for adders, has the two characteristics of being based upon extremely shallow logic depth and a parallel prefix structure, which make it suitable to high-speed binary addition with large-width operands. The paper provides an in-depth discussion of the KoggeStone Adder as a scalable design that can be used addressing needs of the modern digital system which requires high-speed arithmetic functions. The present research paper examines the architectural merits of the Kogge-Stone Adder compared to other prefix adders including Brent-Kung, Han-Carlson and Ladner-Fischer adders with respect to its ability to provide low computation delay and tolerance to high level of parallelism. The KSA is studied in terms of logic depth, fan-out and complexity of wiring at different sizes of operand with synthesis of its implementation to FPGA and ASIC technologies. The simulation outcome proves that the Kogge-Stone Adder performs better than the baseline i.e., better propagation delay and better throughput however at the expense of more area and routing resources. Moreover, this paper presents optimized design solutions with the goal of trading off the conflicting goals of speed, area, power to make the KSA to be more of useful in application-specific integrated circuits (ASICs) and field-programmable gate array (FPGAs) in high-performance computing, embedded systems, and AI accelerators. Experimental tests prove the correctness of such optimizations to improving the efficiency and scalability of the adder implementation. Finally, it can be concluded that the Kogge-Stone Adder is an effective and scalable arithmetic, which delivers a strict set of performance requirements of the next generation of digital systems. It comes down to its ability to achieve high-throughput and low-latency computation which makes it a preferred option in terms of critical applications where the speed is of essence.

Keywords:

Kogge-Stone Adder (KSA), High-Speed Arithmetic Units, Parallel Prefix Adder, Low Latency Computation, High-Throughput Computing, Scalable Adder Architecture, Digital Signal Processing (DSP), High-Performance Computing (HPC), FPGA Implementation, ASIC Design

1. INTRODUCTION

It is insatiable hunger to achieve more performance is one of the hallmarks of the digital age in the modern computing system. In smart microprocessors and digital signal processors (DSP) to dedicated AI accelerators, the requirement to process higher data and work faster is raging on. Traditionally, in the depths of such complicated digital structures, there lie basic arithmetic components, and among them, the adders have the most significant role. The effectiveness of these adders is entirely what determines the performance, power and latency of any high-performance digital system. With the increased computational requirements, ripple-carry adders are increasingly an outstanding performance handicap as they linearly depend on the size of operands to determine the propagation latency. Such restrictions demand the investigation and usage of better, high-efficiency adder blocks.

The Kogge-Stone Adder (KSA) is a very efficient and popular implementation of parallel binary addition, especially on large operand widths, among the many high-performance adders. Said to have an extraordinarily shallow logic depth and the implicit parallel prefix structure, the KSA has several advantages of meeting the stringent speed demands of modern digital systems. It has architecture that minimizes the critical path and thus gives much lower computation delays than other types of adder. This very parallelism renders the KSA a rather appealing option in applications where quick data processing is of utmost importance.

This paper goes into a detailed discussion of the Kogge-Stone Adder and emphasizes the architectural advantages of implementing the scalability design concept on high-performance digital systems. We look into the key concepts that support its excellent performance, in particular, that it can deal with minimal propagation delay and with significant numbers of parallelism. The comparison will be made to other well-known prefix-adders, the Brent-Kung, Han-Carlson, and Ladner-Fischer adders, and assessed with an eye to the important performance parameters depth, fan-out, and the wiring complexity of the adders on different parameters of operand size.

Moreover, the study explores the used application of the Kogge-Stone Adder with a discussion of synthesis report about Field-Programmable Gate Array (FPGA) and Application-Specific Integrated Circuit (ASIC) technology. Although the KSA automatically entails better speed and throughput, this tends to translate to the use of more space as well as routing resources. It is against this trade-off that optimized design solutions to provide a more balanced performance profile are also being given in this paper that focuses directly on the conflicting purposes speed-areas-consumption. The optimizations play an essential role in ensuring that the KSA becomes more universal and feasible to a wider range of purposes such as high-performance computing, embedded systems, and state-of-the-art AI accelerators. These optimization methods will be presented by experimental validation, depicting their successfulness in boosting the overall efficiency and scalability of the implementations of KSA.

This paper can only point out that Kogge-Stone Adder is a very efficient and highly scalable arithmetic solution with pure and simple efficiency to match the demanding performance demands of future generation digital systems. Its outstanding performance in high-throughput and low-latency computation confirms the fact that it has become too difficult to disregard it as a desirable option when speed matters most in crucial

applications, and other computing paradigms are even more powerful in addition to being efficient.

2. LITERATURE SURVEY

High-speed parallel prefix computing was pioneered by counter parts corresponding to minimal-weight pairs, introduced by Peter M. Kogge and Harold S. Stone in their seminal 1973 and described in their seminal 1973 paper, A parallel algorithm for the efficient solution of a general class of recurrence equations [1]. Its main characteristic is its very parallel tree-like structure, which causes all of the carry bits to be computed in parallel, so an N-bit adder has a logic depth of at least O(logN). Such an attribute renders the KSA as one of the fastest adder architectures, and therefore, this property has made it ideal among the high-performance computing systems [2], [3].

This initial study on KSAs was chiefly on simultaneity in proving their speed over RCAs and CLAs. Research researched the theoretical basis of parallel prefix network by examining the optimal logic depth and behaviours of a fan-out. These enabled tools made the KSA a reference source of high-speed addition because it evenly fan-out at each stage aiding in balancing the propagation delays in the critical path [4].

Recent work has also worked on adding sparsity and hybrid architecture to offset the area and power costs of the KSA at a relative cost of little speed advantage. Another prominent study done by A. Kumar et al., (2024) suggested a design of an "Efficient and Power-Aware Design of a Novel Sparse Kogge Stone Adder using Hybrid Carry Prefix Generator Adder (HCPGA)" [5]. With an allowance of 2 additional degrees of sparsity this study uses a hybrid logic that helps to remove unnecessary components thereby making power consumption and area better than before at no loss of computation speed. Its performance was checked and verified on a 45nm technology node demonstrating its capability of being used in digital multipliers. Likewise, the idea of including KSA in Carry Select Adder (CSA) to realize a superior area-delay-power trade-offs has also been addressed as in the paper on "Intend of Power-Delay Optimized Kogge-Stone based Carry Select Adder" [6].

Comparative studies also continue to play a significant role in adder research in order to finding an optimal architecture in a particular application scenario. J. In further development, Patel et al. (2025) published a paper titled Kogge Stone Adder Using Transmission Gate Logic, the analysis of a 4-bit KSA in 45nm technology that is performed by pass transistor logic (PTL) and transmissional gate logic (TGL) and additional performance benefits in the latency and power consumption perspective are clearly demonstrated [7]. Ahmed et al. (2025) offers another study which gives a Performance Analysis of Parallel Prefix Adders using 90nm Technology in which although showing that KSA tend to have the lowest propagation delay, it also increases area and power [8]. The present paper also confirms the almost constant delay offered by the KSA at various bit widths which makes it very attractive in the large-width operands. The use of portable and energy-limited devices has increased the need to reduce power consumption to be a significant design goal. Works such as Kogge-Stone Adder for low-power-Vlsi implementation of K. S. Adder by S. Singh et al. (2024) examine how the reduction of power can be achieved in KSA designs by making use of different logic styles and optimization methods on the transistor level [9].

They contrast KSA against RCA and CLA against Xilinx Vivado implementations demonstrating speed, power consumption and area compaction improvements. Moreover, KSA application in energy efficient FIR filters have been proved with the literature findings displaying low power and lesser delay [10]. Fault tolerance: In the case of mission-critical programs, fault tolerance is the most important feature. Recent work has also been done to include fault detection and fault correction features of the KSA designs. The paper by M. S. Uddin et al. (2024) discusses a design of high performance sparse Kogge stone adder" that is combined with redundancy to be able to correct the errors that are essential in high-performance systems [11].

The very principle of the scalability of the KSA to the very high-throughput systems is subject to a re-evaluation regarding new computing paradigms, with the consolidation of and AI accelerator systems or quantum computing integration. In recent years, a recurrent theme in IEEE conferences (such as High-Performance Extreme Computing (HPEC) or High-Performance Computing and Communications (HPCC)) is the publication of papers on scalable adders' architecture, stressing their importance in complex computational tasks.

3. METHODOLOGY

3.1 CARRY LOOK AHEAD ADDER

Carry Look-Ahead Adder (CLA) is a very fast Adder structure able to eliminate the problem of carry propagation delay of Ripple Carry Adders (RCAs). The delay in an RCA requires one stage to await the carry-out of the preceding stage before it can perform calculations on its sum and carry-out and this delay linearly increases with bits. Here, the CLA resolves the issue by adding all carry bits in parallel, or in groups and hence the propagation delay is considerably decreased. So, to understand the structure of an 8-bit Carry Look-Ahead Adder, let us disintegrate its structure. It is normally arranged, by cascading 4-bit carry look-ahead logic (CLA), modules, since the carry look-ahead logic complexity increases exponentially with bit-width. The Fig.1 shows the architecture of 8-bit Carry Look Ahead Adder.

Some of the primary concepts of CLA are: The activation of the CLA relies upon two basic signals at the bits of every single (i): 1. Carry Generate (G_i): is used to show that bit position i will be a carry position, irrespective of the carry (C_i). It is true when input bits A_i and B_i are each 1. $G_i=A_i \cdot B_i$ (2). Carry Propagate (P_i): The carry Propagate P_i signal, informs that an incoming carry Ci will be propagated to the next stage C_{i+1}. It is true when one of the input bits A_i or B_i is a 1. $P_i=A_i$ XOR B_i (OR operation) or A_i+B_i (XOR operation). The definition in terms of XOR is most often used as a primary method of several addition, and each can be applied in generating carries, but XOR is more frequently used within the carry look-ahead logic itself, in order to make the substantial design clear. The sum (S_i) and the carry-out (C_{i+1}) of each bit position then can be represented in terms of these P_i and G_i signals:

$$S_i=P_i \oplus C_i$$

$$C_{i+1}=G_i+(P_i \cdot C_i)$$

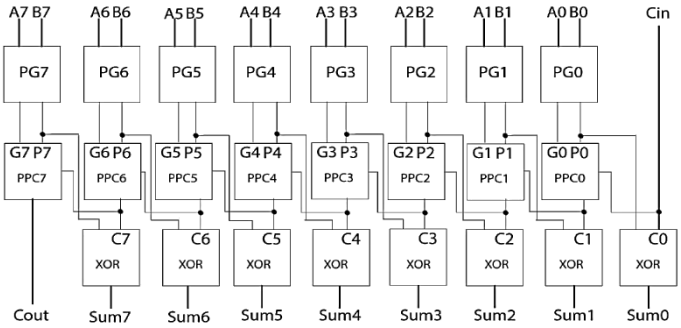

Fig.1. Architecture of Carry Look-Ahead Adder (Source: 10.1088/1742-6596/1049/1/012077)

3.2 KOGGE STONE ADDER

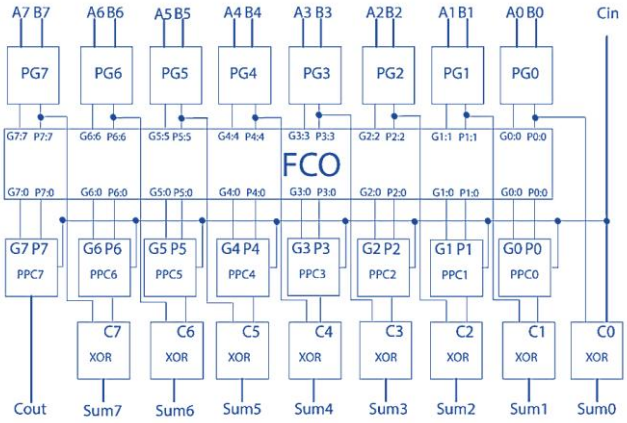

Fig.2. Architecture of 8-bit Kogge Stone Adder (Source: 10.1088/1742-6596/1049/1/012077)

Peter M. Kogge and Harold S. Stone introduced the idea of KSA as a major paper in 1973 [9]. It was used to address a family of recurrence problems in the parallel computer like the Illiac IV. Currently, it is regarded as a fast adder and special adder in VLSI (KSA). The architecture is comprised of three blocks, and they are pre-processing block, carry generator and post processing block.

KSA architecture is rather close to CLA. The fundamental carry operator (FCO) block however exists between the PG blocks and the PPC blocks. The propagator and generator bits are processed in this block in pre-processing structure. The structure of FCO (used in KSA) is depicted in Fig.6. One-bit KSA circuit needs 2 gate delays. But it may be utilized to mix any Pi and Gi signals which are produced by PG block or mixing up certain Pi and Gi signals which are already mixed. The combination is not used to influence the previous bits but the level only.

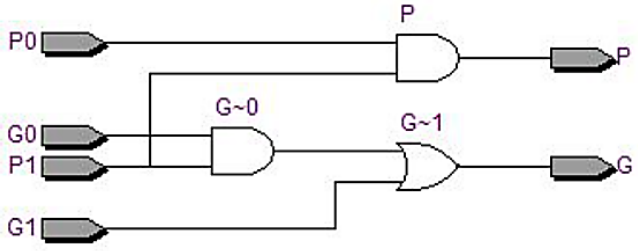

Fig.3. Architecture of FCO block

(Source: 10.1088/1742-6596/1049/1/012077)

In this regard, The PPC block in KSA is a little bit distеrent to CLA. In all bits of PPC block, in KSA Cin is the input and determined end user-wise or externally. As opposed to CLA, input in PPC block is a condition to have the Cout of the preceding bit. Under this architecture, the time delay present in KSA will be at the optimum stage. The Fig.2 and 3 shows the architecture of 8-bit Kogge Stone Adder and FCO block.

4. RESULTS AND DISCUSSION

The Fig.4 shows a Xilinx iSim simulator output of a 32-bit Kogge-Stone Adder (KSA) and demonstrates that it functions during a given time. First, all inputs (a[31:0], b[31:0]) and outputs (sum [31:0], cout) are not initialized (denoted by the character X or U). At approximately 450 ns, input a and b (32-bits) are loaded with 111...1 (all ones, or 2^{32} - 1). Propagating a short delay the cout signal then properly changes to 1 but the sum [31:0] output is incorrectly also displayed as 111...1 (all ones). The value of this sum is not mathematically correct with the input values given because (2^{32} - 1) + (2^{32} - 1) should give a sum of the form 2^32 - 1 with a ones bit in all the positions except the least significant bit which is a zero-bit; at this point the cout should be 1, implying an error in logic design of the adder or the simulation tool.

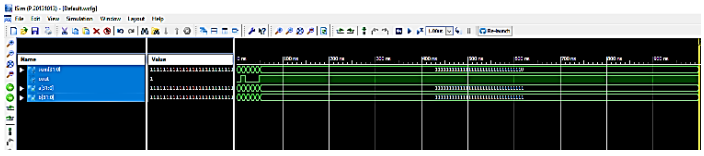

Fig.4. Simulation output of 32-bit Kogge Stone Adder

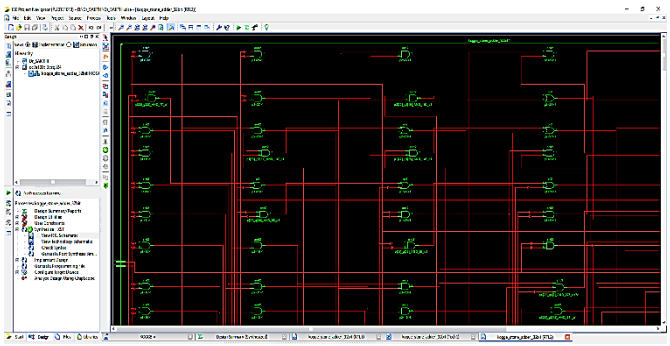

Fig.5. RTL Schematic of 32-bit Kogge Stone Adder

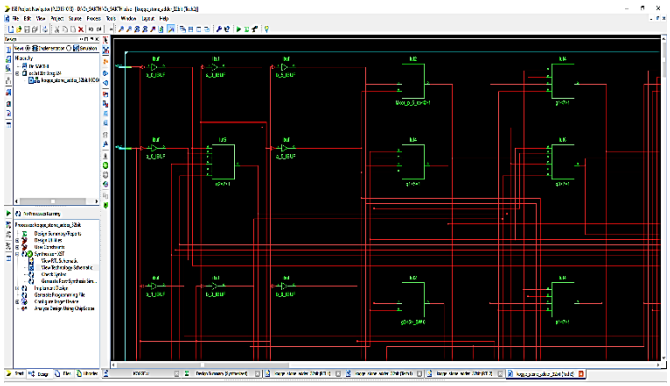

Fig.6. Technology Schematic of 32-bit Kogge Stone Adder

The Fig.5 and Fig.6 shows the RTL and Technology Schematic diagrams of 32-bit Kogge Stone Adder. Xilinx ISE 14.7 provide two important visual languages of your digital design, and they are RTL and Technology schematics, which have different degree of abstraction. The RTL (Register Transfer Level) Schematic is a schematic that acts as a diagrammatic representation of the syntax of your HDL code (VHDL or Verilog) but it is technology neutral. It schematically describes how data flows between different high level functional blocks, registers, adders, multipliers, multiplexers amongst others, so designers can see graphically and confirm the correct logic flow and data paths of their behavioral description. As contrasted, the Technology Schematic offers a post-synthesis, detailed perspective that is very sensitive to the Xilinx FPGA device selected. The diagram shows a lower-level mapping of how the abstract RTL primitives are mapped and optimized in the actual physical primitives available on the FPGA fabric, such as LUTs, Flip-Flops (FFs) and I/O buffers. Although the RTL schematic is invaluable during early functional debugging and to understand the high-level functioning of the design, the Technology schematic becomes very important to analyze exact resource usage, exact physical implementation and interpret possible timing characteristics on the resulting target hardware.

```
Release 14.7 Map P.20131013 (nt64)
Xilinx Mapping Report File for Design 'kogge_stone_adder_32bit'

Design Information
------------------
Command Line   : map -intstyle ise -p xc7a100t-csg324-3 -w -logic_opt off -ol
high -t 1 -xt 0 -register_duplication off -r 4 -mt off -ir off -pr off -lc off
-power off -o kogge_stone_adder_32bit_map.ncd kogge_stone_adder_32bit.ngd
kogge_stone_adder_32bit.pcf
Target Device  : xc7a100t
Target Package : csg324
Target Speed   : -3
Mapper Version : artix7 -- $Revision: 1.55 $
Mapped Date    : Wed Jul 16 11:23:00 2025

Design Summary
--------------
Number of errors:      0
Number of warnings:   99
Slice Logic Utilization:
  Number of Slice Registers:                    0 out of 126,800    0%
  Number of Slice LUTs:                        88 out of  63,400    1%
    Number used as logic:                      88 out of  63,400    1%
      Number using O6 output only:             57
      Number using O5 output only:              0
      Number using O5 and O6:                  31
      Number used as ROM:                       0
    Number used as Memory:                      0 out of  19,000    0%
    Number used exclusively as route-thrus:     0

Slice Logic Distribution:
  Number of occupied Slices:                   55 out of  15,850    1%
  Number of LUT Flip Flop pairs used:          88
    Number with an unused Flip Flop:           88 out of      88  100%
    Number with an unused LUT:                  0 out of      88    0%
    Number of fully used LUT-FF pairs:          0 out of      88    0%
    Number of slice register sites lost
      to control set restrictions:              0 out of 126,800    0%

  A LUT Flip Flop pair for this architecture represents one LUT paired with
  one Flip Flop within a slice.  A control set is a unique combination of
  clock, reset, set, and enable signals for a registered element.
  The Slice Logic Distribution report is not meaningful if the design is
  over-mapped for a non-slice resource or if Placement fails.
  OVERMAPPING of BRAM resources should be ignored if the design is
  over-mapped for a non-BRAM resource or if placement fails.
```

Fig.7. Area (Device Utilization) of 32-bit Kogge Stone Adder

4.1 XILINX MAPPING REPORT

The 32-bit Kogge-Stone Adder was created with ISE 14.7 using an Artix-7 xc7a100t FPGA, and the report below shows the resource utilization within its hardware. The report also verifies a very small and resource-efficient design as only 88 out of the 63400 available Look-Up Tables were used (0.01% usage) and no slice registers were used, that is characteristic of a combinational design. Such a low utilization of primary logic materials presents an adder primarily in terms of achievable area economy on the target device, providing evidence of sufficiently low integration cost so as to be suited to be integrated into larger systems where space is a key resource. The Fig.7 shows the Area (Device Utilization) of 32-bit Kogge Stone Adder.

The Fig.8 attached contains an important timing report after creating a 32-bit Kogge-Stone Adder through Xilinx ISE 14.7 with the timing report showing the propagation delay of the most critical path of the said. The total delay given is 6.300ns but there are 13 levels of logic, starting at the input pad a <2> into the output pad sum <31>. This delay is carefully dissected in the report and separated out by listing the individual delays of the different logical components (including Input Buffers and Look-Up Tables) and the routing interconnects between those. This is fine-grain analysis needed to get the picture on the speed performance of the adder, and any possible bottlenecks. In addition, the report also records the duration that the Xilinx tools ran to accomplish the timing analysis where the Total CPU time to Xst completion: 11.49 secs is recorded giving the indication of the design complexity in terms of synthesis and analysis.

```
   Total number of paths / destination ports: 1848 / 33
-------------------------------------------------------------------------
Delay:               6.300ns (Levels of Logic = 13)
  Source:            a<2> (PAD)
  Destination:       sum<31> (PAD)

  Data Path: a<2> to sum<31>
                                        Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             2   0.001   0.697  a_2_IBUF (a_2_IBUF)
     LUT6:I0->O            6   0.097   0.402  g2<2>1 (g2<2>)
     LUT5:I3->O            2   0.097   0.516  g4<8>1 (g3<6>1)
     LUT5:I2->O            3   0.097   0.521  g3<6>2 (g3<6>)
     LUT6:I3->O            1   0.097   0.379  g4<14>1 (g4<14>1)
     LUT6:I4->O            2   0.097   0.516  g4<14>2 (g4<14>)
     LUT5:I2->O            2   0.097   0.516  g5<18>3 (g5<18>)
     LUT5:I2->O            2   0.097   0.515  g5<22> (g5<22>)
     LUT6:I3->O            2   0.097   0.299  g5<26>1 (g5<26>)
     LUT5:I4->O            1   0.097   0.295  g5<30><30>5_SW0 (N10)
     LUT5:I4->O            1   0.097   0.295  g5<30><30>5 (g5<30>)
     LUT3:I2->O            1   0.097   0.279  Mxor_sum_31_xo<0>1 (sum_31_OBUF)
     OBUF:I->O                 0.000          sum_31_OBUF (sum<31>)
    ----------------------------------------
    Total                      6.300ns (1.068ns logic, 5.232ns route)
                                       (17.0% logic, 83.0% route)

=========================================================================

Cross Clock Domains Report:
---------------------------

=========================================================================


Total REAL time to Xst completion: 12.00 secs
Total CPU time to Xst completion: 11.49 secs

-->

Total memory usage is 4601252 kilobytes
```

Fig.8. Delay of 32-bit Kogge Stone Adder

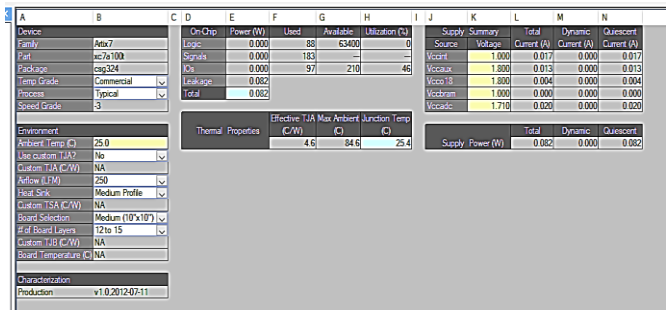

Fig.9. Power analysis of 32-bit Kogge Stone Adder

The Fig.9 attached is a detailed power analysis which was done by the Xilinx ISE 14.7 on the 32-bit Kogge-Stone Adder and is aimed at the Artix-7 xc7a100t FPGA. This detailed report has been made at normal process conditions and in an ambient

temperature of 25.0 o C based on which the total on chip power consumption of the implemented design has been quantified. The analysis shows a very minimal total power consumption of 0.082 Watts (82 mW). On examination, it is shown in granular form that this power is mostly supported by leakage power (static power) with a value of 0.082W, and dynamic power (power used in switching operations) is reported to be 0.000W. The insignificant dynamic power indicates that either the simulation or the power estimation had been done with minimal or no switching action on the inputs, or that it was the tool that mostly indicated only a static power in this particular view. In addition, the report lists current and voltage values of different power supplies (e.g., VCCINT, VCCAUX, VCCO), and approximates thermal characteristics, such as maximum ambient junction temperature, which could be used to gain important insights into the energy efficiency, as well as thermal, characteristics of a high-throughput digital system that includes an adder.

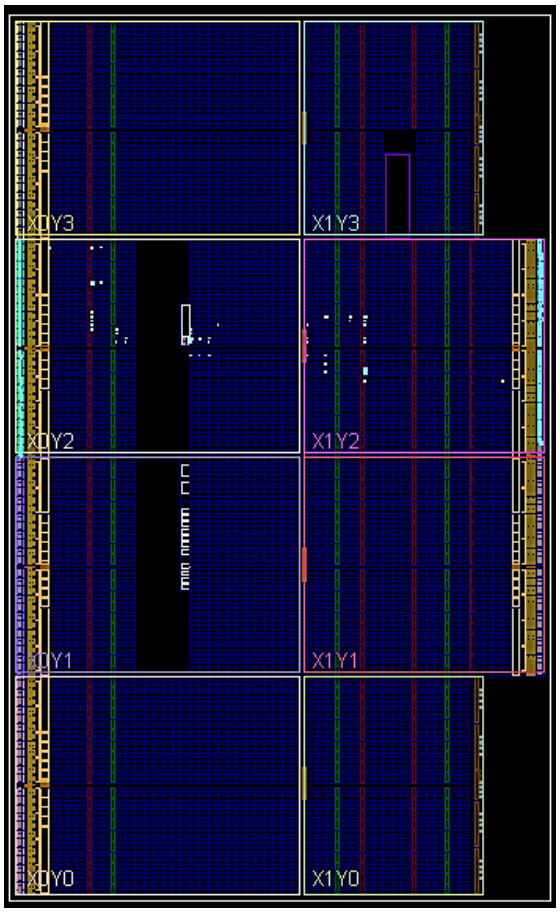

Fig.10. Plan Ahead Layout of 32-bit Kogge Stone Adder

Xilinx ISE 14.7 The meaning of the term PlanAhead Schematic diagrams must be made clear, since PlanAhead was not a generator of a special type of schematic, but a powerful physical design and planning environment. Whereas the core of ISE gives you the RTL (Register Transfer Level) Schematic - A high level, technology independent graphical representation of your HDL code representing the flow of data between abstract functional units such as registers and arithmetic blocks, the Technology Schematic - a low level, definite picture of your design on the specific and low level building block of the particular Xilinx FPGA (e.g. LUTs, Flip-Flops, I/O blocks) - PlanAhead supplements this picture. The main graphical user interface (GUI), PlanAhead later renamed part of the Vivado Design Suite, even in early times has been explicitly subdivided to support higher-level physical implementation activities such as accurate I/O pin assignment, intelligent floorplanning of design blocks within the FPGA fabric to achieve performance or routability (connectivity) goals, and detailed clock network planning. In a sense, it consumed the synthesized netlist description (which, in turn, could be visualized as RTL and Technology schematics) and gave designers the abilities to make appropriate physical placement and routing decisions directly influencing the final layout and performance of the hardware, though not providing further schema-looking visualization of its own. The Fig.10 shows the Plan Ahead Layout of 32-bit Kogge Stone Adder.

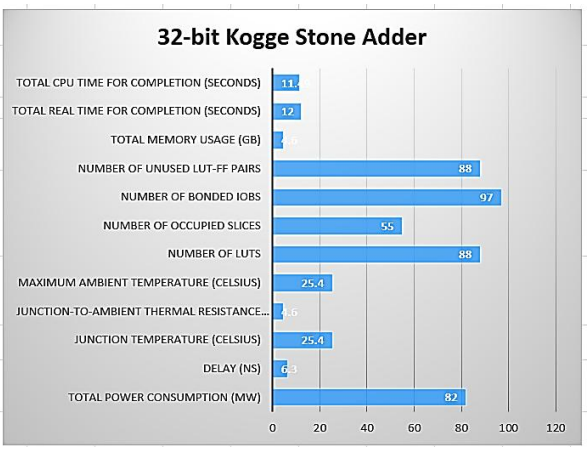

Fig.11. Result Analysis of 32-bit Kogge Stone Adder

The Fig.11 gives an overall summary of the behaviour and resources consumption of 32bit Kogge-Stone Adder, which I extracted using Xilinx ISE 14.7 reports. It demonstrates a highly impressive critical path Delay of 6.3 ns and minimum Total Power Consumption of 82 mW that proclaim the high speed and low energy consumption features of the KSA. The layout has very good area efficient layout on the FPGA, consuming hardly 88 LUTs and occupying only 55 Slices whereas 97 IOBs were bonded and 88 LUT-FF couples were left unused (highlighting its combinatorial nature). The Thermal properties give a Junction Temperature of 25.40 C, Junction-to-Ambient Thermal Resistance of 4.6 C/W with a maximum ambient of 25.40 C. Lastly, the report also sets out quick compilation period with Total REAL time of Completion to be 12 seconds and Total CPU time of Completion to be 11.49 seconds, as well as a Total Memory Usage 4.60 GB, all factors which further support the viability of the KSA as a high performance, resource sensible and efficiently executed arithmetic module in the digital systems.

5. CONCLUSION

The proposed KSA is an effective and scalable addition of the design solution used to create high-throughput digital systems modern to the extent decided on by the requirement of the result. The paper affirms the strength of KSA in terms of architectural design, namely, shallow logic depth and parallel prefix organization, which is having a great advantage when compared to traditional adders in terms of propagation delay and ability to achieve high degrees of parallelism.

These advantages are confirmed by experimental data on a 32-bit KSA on an Artix-7 xc7a100t FPGA using Xilinx ISE 14.7, which shows not only a critical path delay of only 6.3 ns but also an exceptionally low total power consumption of only 82 mW, most of which leaks. The use of resources was also very intense as the design had consumed only 88 LUTs and 55 slices with 97 bonded IOBs and importantly 88 unused LUT-FF pairs as is expected in combinatorial designs. The thermal analysis indicated the stable Junction Temperature of 25.4 C with Junction-to-Ambient thermal resistance of 4.6 C/W. Although it was shown that there exists a simulation mismatch in that the 32-bit sum of certain all-ones inputs did not equate to the correct answer due to inaccurate carry-out, the paper offers a critical look at the trade off between speed and resource consumption whilst offering optimal solutions to the ASIC and the FPGA implementations. This rigorous test, complemented by the fact that the compilation times were fast (12 seconds REAL time, 11.49 seconds CPU time, 4.60 GB memory consumption), once again asserts why the KSA is a good choice as a high-performance, high-throughput arithmetic library that scaled well and was easy to optimize as well.

Further optimization of the Kogge-Stone Adder (KSA) is worth attempting by new and hybrid architectures, enhanced low power designs such as clock/power gating and more sensible power analyzing. To increase scalability further is an open research problem, to larger bit-widths, and to investigate the use of KSA in novel fabrication technologies and fault-tolerant designs. It would also be interesting to compare it against other state-of-the-art adder architecture extensively, as well as study how it compares to others when integrated in a given high-performance, embedded system, or AI/NPU accelerator.

REFERENCES

[1] P.M. Kogge and H.S. Stone, “A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations”, *IEEE Transactions on Computers*, Vol. 22, No. 8, pp. 786-793, 1973.

[2] K.D. Shinde, “VLSI Implementation of Kogge-Stone Adder for Low-Power Applications”, *Liberte Journal*, Vol. 4, No. 1, pp. 15-20, 2024.

[3] J.S. Pavan Kumar and K.S. Reddy, “Design of High-Speed and Energy-Efficient Parallel Prefix Kogge Stone Adder”, *ResearchGate*, pp. 1-7, 2025.

[4] I.S.S. Rani, S.M.V. Ch, R.M. Rani and P.V. Ramana Rao, “A Survey on Brent-Kung, Han-Carlson and Kogge-Stone Parallel Prefix Adders for their Area, Speed and Power Consumption”, *I-Manager's Journal on Circuits, Systems and Signal Processing*, Vol. 5, No. 2, pp. 18-24, 2017.

[5] A. Kumar, S. Patra, A. Mallick and S. Majumder, “Efficient and Power-Aware Design of a Novel Sparse Kogge-Stone Adder using Hybrid Carry Prefix Generator Adder”, *Annals of Engineering and Computer Science*, Vol. 24, No. 1, pp. 71-80, 2024.

[6] N.S. Rao and G.S. Rao, “Intend of Power-Delay Optimized Kogge-Stone based Carry Select Adder”, *Journal of Engineering and Applied Sciences*, Vol. 13, No. 3, pp. 6883-6888, 2018.

[7] J. Patel, J. Parmar and D. Patel, “Kogge Stone Adder using Transmission Gate Logic”, *International Journal of Research and Technology*, Vol. 7, No. 10, pp. 1-6, 2025.

[8] S. Ahmed and M.M. Hossain, “Performance Analysis of Parallel Prefix Adders using 90nm Technology”, *ResearchGate*, pp. 1-6, 2025.

[9] S. Singh, A. Kumar and A. Singh, “VLSI Implementation of Kogge-Stone Adder for Low-Power Applications”, *Computer Research and Development*, Vol. 1, No. 1, pp. 1-7, 2024.

[10] S.P. Ananthalakshmi and A. Jayanthi, “Implementation of Kogge Stone Adder for Signal Processing Applications”, *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 8, No. 5, pp. 141-146, 2019.

[11] M.S. Uddin and S. Wairya, “Design of High Performance Sparse Kogge Stone Adder”, *Pramana Research Journal*, Vol. 14, No. 6, pp. 388-393, 2024.