International Journal of Advanced Trends in Engineering, Science and Technology ***(IJATEST-ISSN:2456-1126)*** ***Volume.6.Issue.3,May.2021***

DOI:10.22413/ijatest/2021/v6/i3/4

HIGH SPEED AND AREA-EFFECTIVE VLSI ARCHITECTURE OF THREE- OPERAND ADDER USING TANNER EDA TOOL

[1]S.SIVAKUMAR, [2]JUTURU LAKSHMI, [3]RAMANNAGARI KEERTHANA, [4]K.BHARATH
[1]Assistant Professor, Dept of ECE, AITS, Rajampet, AP, India.
[2,3,4]Student, Dept of ECE, AITS, Rajampet, AP, India.

Abstract: In different cryptography and pseudorandom bit generator (PRBG) methods, a three-operand binary adder is the basic functional unit for performing modular arithmetic. The carry-save adder is the most commonly used method for performing three- operand addition. The CS3A, on the other hand, has a long propagation delay due to the ripple-carry stage.Furthermore, a parallel prefix two- operand adder, such as the Han-Carlson (HCA), can be utilized for three-operand addition, reducing the criticalpath delay dramatically at the cost of additional hardware. As a result, a new high-speed and area-efficient adder architecture is proposed, which performs the three-operand binary addition using pre-compute bitwise addition followed by carry-prefix computation logic, consuming significantly less area, consuming low power, and significantly reducing the adder delay. The TANNER EDA tool is used to implement the proposed design. The HC3A adder features a smaller area, lower power dissipation, and a shorter delay. We strive for low ADP and PDP with the proposed adder.

Keywords: Three-operand adder, Carry save adder(CSA),Han-carlson adder(HCA), Modular arithmetic.

I. INTRODUCTION

In general, basic operations like addition, subtraction, and division may be performed using various types of binary adders in digitally based processors and control systems. The device's adder performance is solely used to measure a processor's or system's high speed and precision. Previously, 32-bit carry adders such as the Ripple Carry Adder (RCA), Carry Propagate Adder (CPA), and Carry Look ahead Adders (CLA) were employed in processors, each with its own addition time (delay), area, and power consumption. The latency of any binaryadder is estimated based on how fast the carry reaches each and every bit location. As a result, the carry chain, which creates the carry bit, is now the most difficult part of binary adder design. However, because each level of the adder must wait for the preceding carry result, the above current 32 bit basic carry adders have a significant delay value in higher order bits .Three Operand Adder[1] is a well-suited built adder for high speed addition processes with reduced delay in VLSI technology due to the aforesaid difficulty of 32 bit basic existing carry adders in today's world of technology. The Three Operand Adder[1] is also one of the most popular designs, offering reasonable compromise between area, speed, and power. Because the adder is the most widely utilized arithmetic block in the Micro Processor Unit (MPU) and Digital Signal Processor (DSP), it is critical to optimize its performance and power consumption. In today's technological environment, the Three Operand Adder is extremely useful. At the same time, due to the increased density of the chip, the power consumption per chip increases dramatically.

First stage has Bit Addition Logic,It takes three inputs a, b, c and generates two output signals sum and carry. Base Logic takes two inputs.

In second stage we have base logic , which takes two input signals sum and carry and computes the output signals Propagate(Pi) and Generate(Gi). Each cell in second stage takes output signal Sum (S) bit of present full adder and carry output of previous full adder and computes the propagate and generate signals. The proposed the method uses n+1 cells . In the suggested adder mechanism, the external carry-input signal (Cin) is also taken into account for three-operand addition. While computing the G0 (S0 Cin) in the first saltire-cell of the base logic, this additional carry-input signal (Cin) is used as an inputto the base logic.

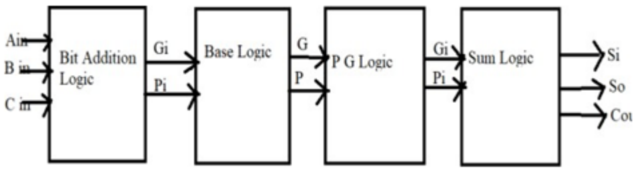

Fig.1.Block Diagram of Three OperandAdder

The addition of three operands is done infour steps, as seen in fig.

1.Bit Addition Logic
2.Base Logic
3.PG Logic
4.Sum Logic

Third step is the carry computation step, also known as "generate andpropagate logic" (PG), is a mixture of black and grey cell logics that pre- computes the carry bit. Figure shows a logical diagram of black and grey cells thatcomputes the carry create Gi: j and propagate Pi: j

Fourth step is the final stage in the proposed three operand adder. It takes carry generate and carry propagate signals from the previous stage and computes the Sum(Si).

Decimal numbers are simple to comprehend and use in arithmetic for humans. Binary numbers, on the other hand, are more practical for a given computation in digital systems such as a microprocessor, DSP (Digital Signal Processor), or ASIC (Application-Specific Integrated Circuit). This happens because binary values are the most economical way to express a large number of values. One of the most important logic units in a digital system is the binary adder. Binary adders are also useful in units other than ALUs, such as multipliers, divisions, and memory addressing.

As a result, binary addition is so important that any improvement in it can result in performance gain for computer system, and so help to enhance the whole system's performance. The carry chain is the most significant issue in binary addition. The length of the carry chain grows in proportion to the breadth of the input operand. The worst case scenario is when the carry takes the longest possible path from the least significant bit (LSB) to the most significant bit (MSB) (MSB). It is possible to accelerate, but not eliminate, the carry chain in order to increase the performance of carry-propagate adders. As a result, when it comes to optimising computer architecture, most digital designers turn to faster adders because they tend to determine the critical route for most computations.

In most digital circuit designs, such as digital signal processors (DSP) and microprocessor data route units, the binary adder is a key component. As a result, substantial research is still being done to improve the adder's power delay performance. Three operand adders are known to offer the best performance in VLSI implementations. Reconfigurable logic, such as Tanner EDA , has grown in popularity in recent years as a result of its superior performance in terms of speed and power over DSP-based and microprocessor-based solutions for a variety of practical designs involving mobile DSP and telecommunications applications. With the growing popularity of mobile and portable devices, which make full use of DSP functions, the power advantage is especially essential. Three operand adders, on the other hand, will operate differently than VLSI versions due to the topology of the programmable logic and routing resources on FPGAs. Most contemporary FPGAs, in particular, use a fast-carry chain that streamlines the carry path for the simple Ripple Carry Adder(RCA). The practical problems of designing and implementing tree-basedadders on Tanner EDA are explained in this study. Several tree-based adder structures are developed, described, and compared to the Carry Save Adder (RCA) and the Han Carlson Adder.

II. RESEARCH CONTRIBUTIONS

The implementations provided in this dissertation aid in the design of three operand adders and the computer systems that support them. This could have ramifications for a variety of application- specific and general-purpose computer architectures. As a result, this work has the potential to influence the design of many computing systems, as well as many fields of engineering and research. The practical problems of designing and implementing tree-based adders on Tanner EDA[5] are detailed in this study. Tanner EDA[5] is used to implement and characterise several tree-based adder structures, which are then compared to the Carry Save Adder (CSA) and the Han Carlson Adder (HCA).Finally, some recommendations and conclusions are made for upgrading Tanner EDA[5]designs to improve tree-based adder performance.

III. EXISTING METHODS

3.1. Carry Save Adder:

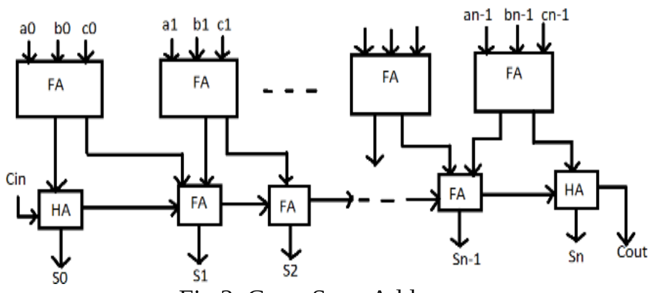

Fig.2. Carry Save Adder

Two two-operand adders or one three- operand adder are often used to perform the three-operand binary addition. The carry-save adder (CSA)[2] could be a space-efficient and mostly used adder. within the standard arithmetic utilized in cryptologic algorithms and PRBG approaches, a strategy for doing three- operand binary addition was adopted. The bigger carry propagation delay in CSA's ripple-carry stage, on the opposite hand, contains important impact on MDCLCG and alternative cryptography designs' performance on IoT-based hardware devices. The Carry Save Adder is used to add 3 bit or n-bit values. The CSA is same as the Full Adder. we will utilize CSA for the addition of the partial product terms of every cluster rather than the other adder. compared to CSA, alternative adders are slower. In many circumstances, we must add more than two integers together. The simplest approach to put m numbers together is to add the first two, where both numbers are n bits wide. Then the sum of the previous two is added to the third, and so on. For a total gate delay of O (i.e. m log n), building a tree of adders with just O (i.e. log m * log n) additions will be necessary. We can greatly lessen the delay with the use of carry saving addition.. we will greatly reduce the delay with the employment of carry saving addition.

Drawbacks :

1. Each stage of a carry-save addition mustbe completed.
2. We have instant access to the result ofthe addition.
3. Larger critical path delay.

3.2.Han Carlson Adder

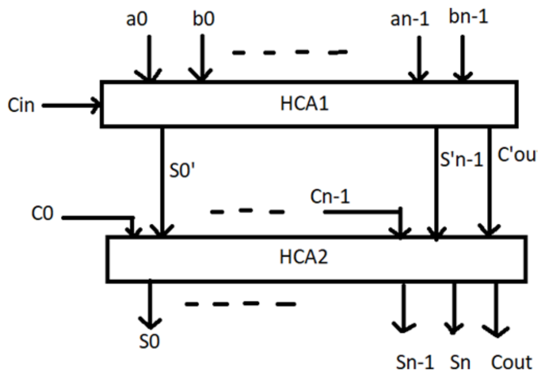

Fig.3. Han Carlson Adder

DOI:10.22413/ijatest/2021/v6/i3/4

For three-operand binary addition, a parallel prefixed two-operand adder such as Han-Carlson (HCA)[3] can be utilized to reduce the critical path delay. It decreases the critical path delay by O(log2 n), but increases the area by O(n log2 n). As a result, an efficient VLSI design must be developed in order to perform the fast three-operand binary addition with minimal hardware resources. Incomparison to the HCA-based three- operand adder, a new high-speed area-efficient adder technique is proposed in this paper that uses pre-compute bitwise addition followed by carry-prefix computation logic to perform the three- operand addition and consumes significantly less gate area while minimizing the propagation delay (HC3A).The proposed adder design is also built using Verilog HDL and then synthesized using a commercially available 32nm CMOS technology library. The new adder technique's area-delay and power-delay products are also tested and compared to the existing CS3A and HC3A three- operand adder approaches. Two stages of parallel prefix two-operand adder can also be employed to reduce the critical route delay. The quickest two operand adder approaches are parallel prefix and logarithmic prefix adders, according to the literature. There are six distinct topologies for these adder techniques: Brent-Kung, Sklansky, Knowles, Ladner-Fischer, Stone (KS), and Han-Carlson (HC). When the bit size grows (i.e. n > 16), Han-Carlson is the quickest of the three. Various parallel prefix two-operand adders, such as the Ling, Jackson-Talwar, ultra-fast adder, hybrid PP CSL, and hybrid Han-Carlson, have been studied in the Kogge-literature in recent years. The ultra-fast adder is said to be the fastest, and it is three gates quicker than the Han-Carlson. It does, however, utilize two times the gate area of the Han-Carlson adder. The hybrid Han- Carlson adder, on the other hand, has two Brent-Kung stages at the beginning and finish, as well as Kogge-Stone stages inthe centre. With a 10% to 18% reduction in gate complexity, this resulted in a somewhat higher latency (two gates delay) than the Han Carlson adder. In comparison to other existing two-operand adder approaches, the Han-Carlson adder gives a reasonable speed at a low gate complexity. The delay product (ADP) and power-delay product (PDP) are the lowest among all.

As shown in Fig. 2, the three-operand addition can be accomplished in two phases using the Han-Carlson adder (HCA).The architecture of the HCA-based three-operand adder (HC3A) is describedin depth. The propagation chain, i.e. the number of black-grey cell stages in the PG logic of the Han-Carlson adder.

Drawbacks:

- The area increases with the increase of bit length.

In the following part, a new high-speed, area-efficient three-operand adder technology and its efficient VLSI architecture are proposed to reduce this trade-off between area and latency.

IV.THREE OPERAND ADDER

The three-operand addition in modular arithmetic is performed using a new adder mechanism and its VLSI design, which is shown in this section. A parallel prefix adder is the adder mechanism which is proposed. The prefix adder, on the other hand, uses four-stage structures instead of three-stage structures to compute the addition of three binary input operands. They are:

1).Bit-addition logic 2).Base logic

3).PG (propagate and generate) logic, and4).Sum logic.

Stage 1: Bit-addition logic

In the first step (bit-addition logic), an array of full adders performs bitwise addition of three n-bit binary input operands, with each full adder computing "sum (S I and "carry (cyi)" signals as shown in Fig. Each F block in first stage takes three input bits a, b, c and performs bit addition logic and gives the two outputs sum(s) and carry(c). The logic expressionand the logic circuit used in the calculation of sum and logic is shown below

$$Si^{'} = ai \oplus bi \oplus ci$$

$$Cyi = ai.bi + bi.ci + ci.ai$$

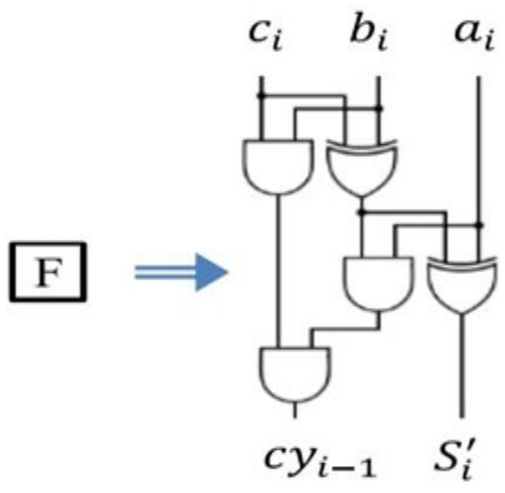

Fig.4. Bit Addition Logic

Stage 2: Base logicIn second stage we have base logic , whichtakes two input signals sum and carry and computes the output signals Propagate(Pi) and Generate(Gi).In second stage we have base logic , which takes two input signals sum and carry and computes the output signals Propagate(Pi) and Generate(Gi).

Each ⊠ cell in second stage takes output signal Sum (S) bit of present full adder and carry output of previous full adder and computes the propagate and generate signals. The proposed the method uses n+1 cells . In the suggested adder mechanism, the external carry-input signal (Cin) is also taken into account for three-operand addition. While computing the G0 (S0 Cin) in the first saltire-cell of the base logic, this additional carry-input signal (Cin) is used as an input to the base logic. Thelogic expression and logic circuit diagram used in the computation of P and G is shown below:

$$Gi:i = Gi = Si^{'}.Cyi - 1,$$

$$G0:0 = G0 = S0^{'}.Cin$$

$$Pi:i = Pi = Si^{'} \oplus Cyi1,$$

$$P0:0 = P0 = S0^{'} \oplus Cin$$

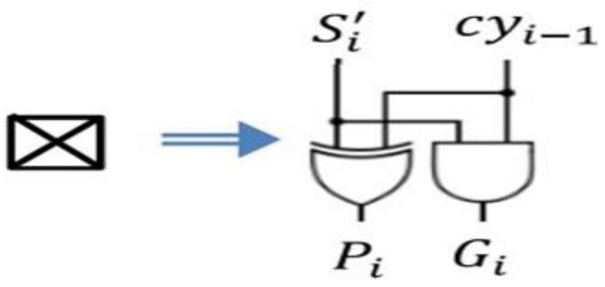

Fig.5. Base Logic

Stage 3:PG Logic

The carry computation step, also known as "generate and propagate logic" (PG), is a mixture of black and grey cell logics that pre-computes the carry bit. Figure 3(b) shows a logical diagram of black and grey cells that computes the carry create Gi: j and propagate Pi: j signals using the logical expression, and the logic circuit diagram is shown below ,

$$Gi{:}\,j = Gi{:}\,k + Pi{:}\,k.\,Gk - 1{:}\,j$$
$$Pi{:}\,j = Pi{:}\,k.\,Pk - 1{:}\,j$$

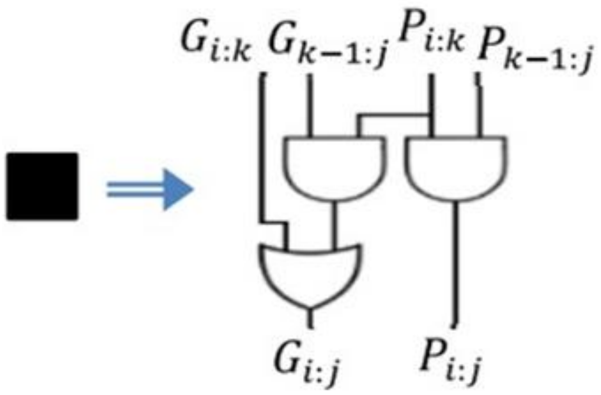

Fig.6. PG Logic

Stage 4: Sum Logic

Fourth stage is the final stage in the proposed three operand adder. It takes carry generate and carry propagate signals from the previous stage and computes the Sum(Si) using the logic expression and logic circuit shown below

$$Si = (Pi \oplus Gi - 1{:}\,0),$$
$$S0 = P0, Cout = Gn{:}\,0$$

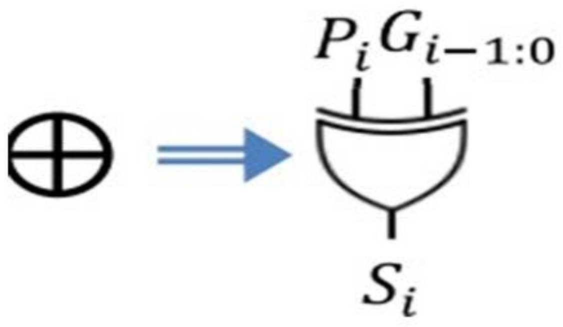

Fig.7.Sum Logic

Block diagram of the proposed 4 bit three operand binary adder is shown below

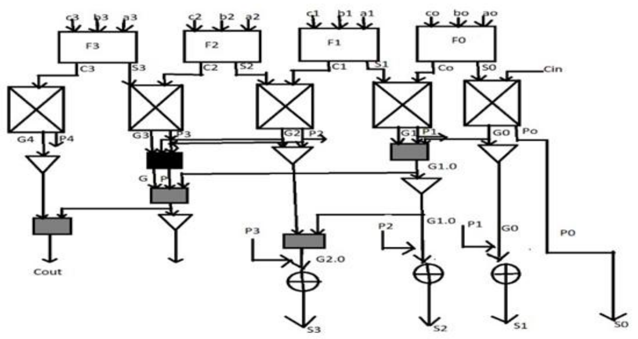

Fig.8.Block diagram of 4 bit three operandadder

Schematic of the 4 bit three operand adder in Tanner EDA tool is shown below figure.

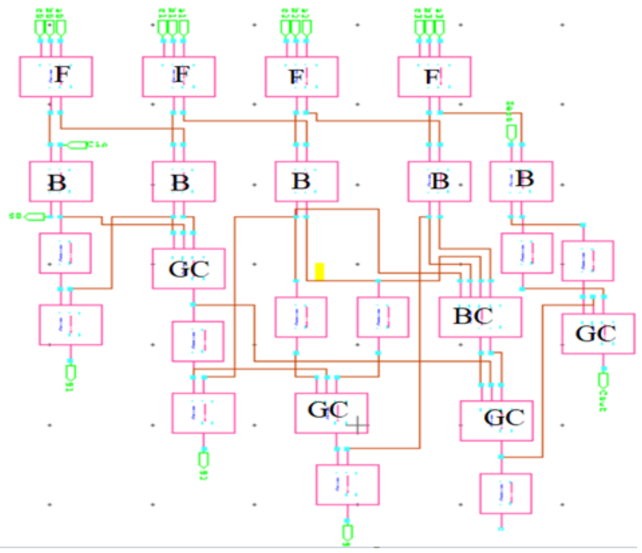

Fig.9. Schematic of 4 bit three operand adder in Tanner EDA tool

IV. RESULTS

Types of Three Operand Adder	Levels in Three Operand Adder tree	Total nodes used in Three Operand Adder tree	Total logic gates	Average power consumption (Micro Watts)
4-bit	4	2426	547	0.2445

Waveforms:INPUT

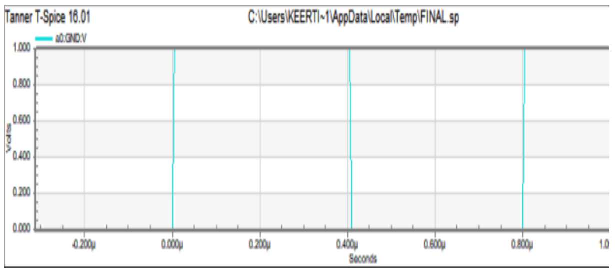

Fig.10a:1st Operand 4-bit input data a0,a1,a2,a3

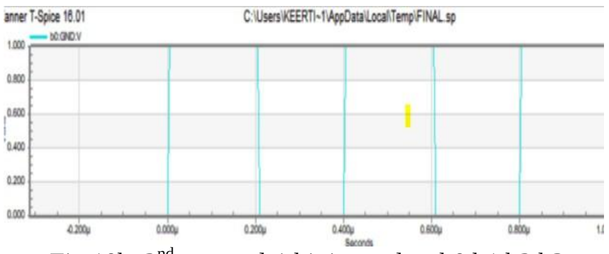
Fig.10b. 2nd operand 4-bit input data b0,b1,b2,b3

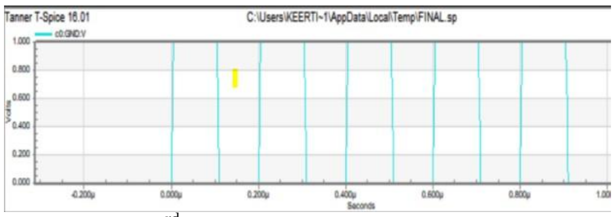
Fig.10c. 3rd Operand 4-bit input data c0,c1,c2,c3

OUTPUT

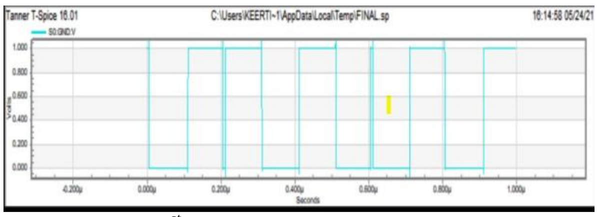
Fig.10d. 1st sum output S0

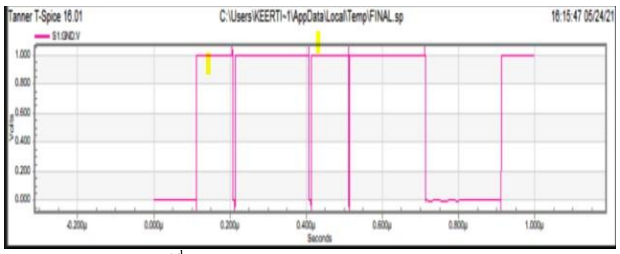
Fig.10e. 2nd sum output S1

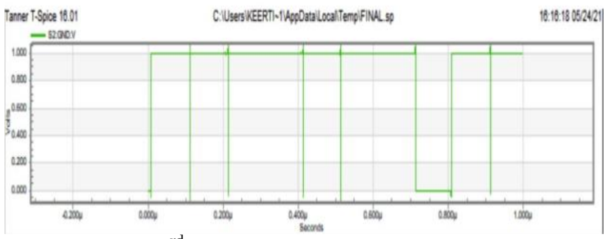
Fig.10f. 3rd sum output S2

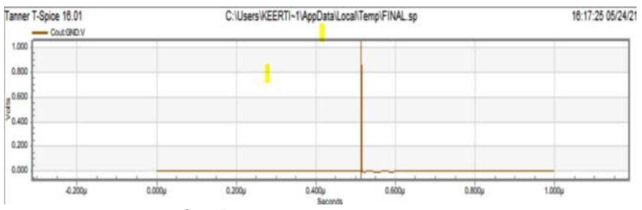
Fig.10g. final output Cout

Fig.10:Wave Forms

V. CONCLUSION

In this article, we have developed a High speed and Area effective Three Operand Adder circuit using tanner EDA tool to achieve lower power consumption and better speed performance. This 4-bit Three operand adder consumes 0.2445 micro watts power.

REFERENCES

[1] https://ieeexplore.ieee.org/document/9173549.
[2] https://www.researchgate.net/publication/301407573_Design_of_high_speed_carry_save_adder_using_carry_lookahead_adder
[3] https://www.researchgate.net/publication/344955631_Design_and_Analysis_of_Kogge-Stone_and_HanCarlson_Adders_in_130nm_CMOS_Technology
[4] https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/iet-cds.2019.0443
[5] https://www.academia.edu/Documents/in /Tanner