

Survey of VLSI Adders

Swathy.S¹, Vivin.S², Sofia Jenifer.S³, Sinduja.K³

¹UG Scholar, Dept. of Electronics and Communication Engineering, SNS College of Technology, Coimbatore-641035, Tamil Nadu, India

² UG Scholar, Dept. of Electronics and Communication Engineering, SNS College of Technology, Coimbatore-641035, Tamil Nadu, India

Abstract - Digital computation depends upon the efficiency of the rate of computations carried out. Addition is one of the most highly used computation process of any digital firm. These high speed addition computations are performed by highly efficient adders. On the most efficient and fastest adders developed in the parallel prefix adders. The popularity of the parallel prefix adders is that its ability to compute addition operation with a significantly high speed, reliability and efficiency, in the category of Very Large Scale Integration (VLSI). There are many types of parallel prefix adders that perform addition operation with various different logics and every type uses the prefix operation. The prefix operation is the logic of computing the output depending on the previous input. The different type of parallel prefix adders are namely, Brent Kung Adder, Kogge-Stone Adder, Ladner Fischner Adder, SQR CSLA, Han-Carlson Adder, CSA, Linear CSLA. The epitome of this paper is to make a survey about the most commonly used parallel prefix adders that are listed above, and compute their power, delay and efficiency.

Key Words: Kogge-Stone Adder, Ladner Fischner, Han-Carlson, CSA, SQR CSLA, Linear CSLA.

1.INTRODUCTION (Size 11 , cambria font)

The parallel prefix adders were designed to compute addition operation of any digital system that has very large scale integration capabilities. The VLSI chips heavily rely on the high speed efficient adders and almost every single VLSI chip has a series of parallel prefix adders in them to compute their arithmetic operations. The conventional adders can be used for the VLSI computations and these conventional adders will be sufficient, but as the width of them adder grows the delay of the path begins to overpower the adder, to overcome this delay domination the parallel prefix adders are used. The parallel prefix adders are the most efficient and fast adders till date. The most commonly used parallel prefix adders are

used in this survey to determine the efficiency, delay and power of each adder separately and then compare the results with one another.

2. BRENT KUNG ADDER

The Brent Kung adder is a familiar type of the parallel prefix adder. Practically, the Brent kung adder has low fan-out from each prefix which in turn does reduce the delay significantly but the adder has long circuit path, which is ideally not suitable for high speed arithmetic computation. The Brent Kung adder proposes an optimized high speed added that addresses the problems of gate count, delay and gate connection in a way to reduce the gate area noticeably, in spite of having long critical paths. The brent kung adder is considered as one of the better high speed adders for minimizing the wiring tracks, fan-out and gate counts used as a basis for many other networks.

In parallel prefix adders, binary addition is usually expressed in terms of generate signal, propagate signal, carry signal and the sum signal at each bit position ($1 \leq i \leq n$), all the above listed signals can be calculated by the equations^[1],

$$G_i = a_i + b_i \tag{1}$$

$$P_i = a_i + b_i \tag{2}$$

$$C_i = \begin{cases} g_i \\ g_i + p_i C_{i-1} \end{cases} \tag{3}$$

$$S_i = p_i + c_{i-1} \tag{4}$$

The Brent Kung adder computed the sum in three stages namely,

- Pre-Processing
- Prefix Carry Tree
- Post- Processing
-

a) Pre-Processing Stage

The n bit parallel prefix adder operation begins with the Pre-Processing stage for the generation of P_i and G_i [1][2].

b) Prefix Carry Tree Stage

The carry bits signal are obtained from the signal from the first stage that will proceed with the next stage. This stage contains three main complex logic categories namely; Black Cell, Grey Cell and the Buffer Cell. The black cell computes the $G_{i,j}$ and P_{j+1} , whereas the grey cell can only execute the $G_{i,j}$. The prefix carry tree stage is a part that differentiates or determines the adder that is used[1][2].

$$G_{i,j} = G_{i+1,k} + P_{i+1,k} \times G_{k-i,j} \quad (5)$$

$$P_{i,j} = P_{i+1,k} \times P_{k-i,j} \quad (6)$$

c) Post-Processing Stage

This stage is where the overall adder operation is completed, the carry signals of the second stage will pass through to the final stage (i.e.,) Post-Processing Stage. The final result of the entire adder operation is obtained from the equation (4).

2. KOGGE-STONE ADDER

The most popular carry look ahead adder amongst the high speed parallel prefix adders is the Kogge-Stone adder. It has the fastest adder design; the application of the fast computation is achieved at the cost of increased area.

The Kogge-stone adder has three segments in which the addition operation is performed, they are,

- Pre-Processing
- Carry look ahead network
- Post-Processing

a) Pre-Processing

The function of this stage is to generate the carry propagate signal P_i and the carry generate signal G_i from the input signal bits A and B . The carry propagate bit is obtained by performing the exclusive operation of the input bits A_i and B_i . The carry generate bit is obtained by performing the *and* operation of the input bits A_i and B_i . The above steps are illustrated as in the equations given below,

$$P_i = A_i \text{ xor } B_i \quad (7)$$

$$G_i = A_i \text{ and } B_i \quad (8)$$

b) Carry look ahead network

The function of this stage is to compute the carry propagate and generate signal bits of the next stage based on the inputs of the previous stage. The generate and propagate bits namely, $G_{i,j}$, $P_{i,j}$, are computed by this stage from the previous inputs (G_i , P_i) and (G_j , P_j). The bits $P_{i,j}$ and

$G_{i,j}$ are also calculate by this stage. The sequence o equation required to calculate the above mentioned bits are given below,

$$P_{i,j} = P_{i:k+1} \text{ and } P_{k,j} \quad (9)$$

$$G_{i,j} = G_{i:k+1} \text{ or } (P_{i:k+1} \text{ and } G_{k,j}) \quad (10)$$

c) Post-Processing

The sum and carry bits of the entire addition operation is performed at this stage and this is responsible for the generation of the sum S_i and Carry C_i bit of the addition operation, this can be achieved by computing the following equations,

$$S_i = P_i \text{ xor } C_{i-1} \quad (11)$$

$$C_i = G_{i,0} \text{ or } (C_{in} \text{ and } P_{i,0}) \quad (12)$$

This is the stage where the overall addition operation is full filled by the adder after the generation of various bits based on various logics of the previous stages[1][2].

3. LADNER FISCHNER ADDER

Ladner Fischner adder is a form of parallel prefix adder. It also falls under the category of carry look ahead adder that can be represented as a parallel prefix graph containing operator nodes. It is also the fastest adder with the focus on design time and the most common choice of high speed performance adder amongst the industries, as the time required to generate the carry signals is calculated using the expression ($\log n$). The better performance of the Ladner Fischner adder is because of its logic depth being minimum and bounded fan-out. The only downside to the Ladner Fischner adder is that it occupies a large silicon area.

The Ladner Fischner adders are more flexible and are used to speed up the binary additions and are obtained from Carry Look Ahead (CLA) structure. Tree structure form is used to increase the speed of arithmetic operation. The Ladner Fischner adder consists of black cells and grey cells. Each black cell consists of two *AND* gates and one *OR* gate. Multiplexer is a combinational circuit which consist of a single output obtained by multiplexing multiple inputs. Each grey cell consists of one *AND* gate only. The P_i denotes the propagate bit and it consists of only one *AND* gate. G_i denotes the generate bit and it consists of one *AND* gate and one *OR* gate. The computation equations for the propagate and the generate bits are given below,

$$P_i = B_i \text{ and } B_{i-1} \quad (13)$$

$$G_i = A_i \text{ or } (B_i \text{ and } A_{i-1}) \quad (14)$$

The generate bit can also be calculated by the equation,

$$G_{i-2} = A_{i-2} \text{ or } (B_{i-2} \text{ and } A_{i-1}) \quad (15)$$

The addition operation in the Ladner Fischner Adder is carried out in three stages namely,

- Pre-Processing Stage
- Carry Generation Stage
- Post-Processing stage

a) Pre-Processing Stage

Just like any other parallel prefix adder the first stage is to compute the carry propagate P_i and the carry generate G_i . The equations for the computations of the propagate and the generate signals are,

$$P_i = A_i \text{ xor } B_i \tag{16}$$

$$G_i = A_i \text{ and } B_i \tag{17}$$

b) Carry Generation Stage

As the name suggests, this stage is for the generation, of carry bits. The carry propagate and the carry generate are generated at each cell, but the final cell present in each bit is responsible for generating the carry. The last bit carry of each stage will help to produce the sum of the next bit simultaneously until the computation of the final bit of each stage. The carry generate and the carry propagate are computed from the equations (1) and (2).

c) Post- Processing Stage

The final stage of the Ladner Fischner adder is effect as the carry of the first bit of every cell is *xored* with the next bit if the propagates, and then the output is given as the final sum.

$$S_i = P_i \text{ and } C_{i-1} \tag{18}$$

4. SQUARE ROOT CARRY SELECT ADDER

The construction of the adder is by equalizing the delay through the dual carry chains and the multiplier block signal from the previous stage. The Square Root Carry Select Adder is also known as the Non-Linear Carry Select Adder. The SQRT CSLA uses Binary to Excess-1 Converter in the place of Ripple Carry Adder to achieve lower delay with a slightly increase in area.

In the case of a regular SRQT CLSA has 2 ripple carry adder with 2:1 multiplexer, it comparatively has a larger area due to the multiple adder parts which is considered as the main disadvantage of the SQRT CSLA. The adder is divided into 5 groups containing different bit sized carry adders, the carry out are calculated from the last stage by the multiplier [1][2][3].

5. HAN-CARLSON ADDER

Adders are more commonly used in DSP lattice to serve the purpose of arithmetic calculation with the maximum efficiency. The efficiency of an adder depends on the parameters like the reliability, processing speed, gain and delay performance of the adder and the chip size which is proportional to the functionality of the adder chip. The adders can be made more effective and efficient in the case of parallel prefix adders as the chip area size is minimized without the sacrifice of the functionality of the adder. The Han- Carlson adder is a combination of different stages if the Brent Kung Adder and the Kogge-Stone Adder.

The processing of the Han-Carlson adder is similar to that of the Brent Kung and Kogge-Stone adder containing the same number of stages and the process of the stages are similar too. The main difference is that the adder offers trade-offs between the number of stages of computation of the logic manipulation and number of logic gates. Brent Kung uses minimal number of computational nodes, which yield the maximum depth. Kogge-Stone achieves high speed in the computation process and low fan out with the downside of having complex circuitry which requires more number of wiring tracks.

The Han-Carlson adder uses comparatively fewer number of prefix operations by making changes to the number of computational stages of the adder unlike the Kogge-Stone and the Brent Kung adder, it also reduces the area of the adder chip which provides the flexibility of the adder with greater functionality and lesser chip size.

$$G = A \text{ and } B \tag{19}$$

$$P = A \text{ xor } B \tag{20}$$

The computation of the prefix operation is given by the expression,

$$(g_i, p_i) \circ (g_j, p_j) = (g_i + p_i \cdot g_j, p_i \cdot p_j) \tag{21}$$

5. CARRY LOOK AHEAD ADDER

The carry propagation time is the main constraint of the ripple carry adder family. Other arithmetic operations such as multiplication are performed on the basis of addition and subtraction operation. Thus by default the speed of the addition operation can be increased by increased the speed

of the adders, which can be achieved by reducing the carry propagation delay of the adders.

The carry look ahead adder employs the principle of carry look ahead which overcomes the issue of calculating the carry at the time of computation if each stage. The carry look ahead adder calculates the carry of each stage computation in advance based on the inputs of the corresponding stages. The propagate and the generate signals of each stage are calculated in advance and are compiled under two cases namely,

1. When both the A and B bits are 1
(or)
2. When one of the two bits is 1 and the carry signal of the previous stage is one.

The carry and propagate signals of the adder with the input signals A and B are given by,

$$P_i = A_i \oplus B_i \quad (22)$$

$$G_i = A_i B_i \quad (23)$$

The output of the adder sum and carry signals are given by,

$$S_i = P_i \oplus C_i \quad (24)$$

$$C_{i+1} = G_i + P_i C_i \quad (25)$$

The carry signals of various stages of the computation process of the adders are calculated from the expression,

$$C_1 = G_0 + P_0 C_0$$

$$C_2 = G_1 + P_1 C_1 = G_1 + P_1 (G_0 + P_0 C_0) \\ = G_1 + P_1 G_0 + P_1 P_0 C_0$$

$$C_3 = G_2 + P_2 C_2 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$$

$$C_4 = G_3 + P_3 C_3 \\ = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_0$$

6. CARRY SKIP ADDER

The main constraint of computing the arithmetic functions of the adders and their delay intervention during computation is rather high. To improve the delay intervention and the functionality of the adder chip carry skip adder principle can be used. The delay intervention of the adder chip can be improved depending upon the number of CSA used in coherence. The relation between the number of Carry

Skip Adders used to the delay interference of the adder system is inversely proportional. The propagate signal of the Carry Skip Adder can be expressed as

$$P_i = A_i \oplus B_i \quad (26)$$

The main advantage of the Carry Skip Adder is to reduce the latency. The number of input AND gates is equal to the width of the adder. This relation is not applicable if the bandwidth of the adder is large and if additional delays are present in the system.

The performance characteristics of the Carry Skip Adder can be explained as,

- It is chained.
- It reduces the critical path.
- No real speed benefit.

There are two types of Carry Skip Adder namely,

- Variable Carry Skip Adder
- Multi-level Carry Skip Adder

The structure of the carry skip adder consists of six full adders, C_{in} and C_{out} .

7. LINEAR CARRY SELECT ADDER

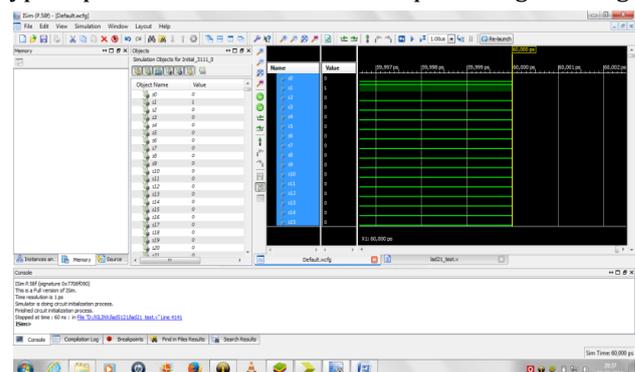
The main function of the Linear Carry Select Adder is that it computes $(n+1)$ at two bit of numbers and the depth of the Linear Carry Select Adder is $(0\sqrt{n})^{[3]}$.

The linear carry select adder implies a particular way to implement the adder function. There are two results that can be calculated from the correct sum C_s , which is taken and they are correct carry in, C_{in} and correct carry out C_{out} signals. So the delay of the block size is calculated as (\sqrt{n}) .

8. SAMPLE OUTPUT

8.1 CARRY SKIP ADDER

This device belongs to the family of Spartan 3. It is of a typical process with commercial temperature grade of fg900



package. The ambient temperature is 25 degrees in Celsius. The production is characterized as V1.2,06.25.09. It has fair thermal properties like effective JJA, maximum ambient and junction temperature as 13.1, 80.6, 29.4 respectively. The supply power in total is 0.338.

8.2 MODIFIED CSLA

This device belongs to the family of Spartan 3. gxc3s5000 part. Commercial grade and typical device of fg900 package and of -4 speed grade. The ambient temperature is 25.0 degrees in Celsius. The on-chip power for logic signals and IO's are 0 and the leakage is 0.338. It has fair thermal properties like effective JJA, maximum ambient and junction temperature as 13.1, 80.6, 29.4 respectively. The maximum combinational path delay is 87.728 ns.

8.3 MODIFIED SQRT CSLA

This device belongs to the family of Spartan 3. gxc3s5000 part. Commercial grade and typical device of fg900 package and of -4 speed grade. The ambient temperature is 25.0 degrees in Celsius. The on-chip power for logic signals and IO's are 0 and the leakage is 0.338. It has fair thermal properties like effective JJA, maximum ambient and junction temperature as 13.1, 80.6, 29.4 respectively. The maximum combinational path delay is 19.495 ns^{[3][4]}.

8.4 SQRT CSLA

This device belongs to the family of Spartan 3. gxc3s5000 part. Commercial grade and typical device of fg900 package and of -4 speed grade. The ambient temperature is 25.0 degrees in Celsius. The on-chip power for logic signals and IO's are 0 and the leakage is 0.338. It has fair thermal properties like effective JJA, maximum ambient and junction temperature as 13.1, 80.6, 29.4 respectively. The maximum combinational path delay is 13.771 ns^{[3][4]}.

3. CONCLUSIONS

The primary purpose of the paper is to make a survey of all the details and parameters of the Brent Kung Adder, Kogge-Stone Adder, Ladner Fischner, Han-Carlson, CSA, SQRTCSLA and Linear CSLA to identify the efficient adder by calculating the parameters like power gain and delay

REFERENCES

[1] Lecture 4 - Adders Computer Systems Laboratory- Stanford University, http://web.stanford.edu/class/archive/ee/ee371/ee371.1066/lectures/lect_04.2up.pdf

[2] A Comparative Analysis of Parallel Prefix Adders - Department of Electrical and Computer Engineering University of Texas at San Antonio San Antonio, TX 78249, <http://worldcomp-proceedings.com/proc/p2013/CDE4118.pdf>

[3] International Journal of Soft Computing and Engineering (IJSCE) - Low Power and Area-Efficient Carry Select Adder - <http://www.ijscce.org/attachments/File/v2i6/F1117112612.pdf>

[4] Design of Adders by J. A. Abraham Department of Electrical and Computer Engineering, The University of Texas at Austin, VLSI Design - <http://www.cerc.utexas.edu/~jaa/vlsi/lectures/8-1.pdf>