Adders for High-Performance Computing

HKUST 19/20 Fall Term (ELEC4410)

Prepared by: Yu, Kijun (20319196)

Abstract

This report is based on the research of Performance of various types of adders in computing systems, compared for addition of 8-bit, 32-bit, and 64-bit numbers. For fast, compact and power efficient adders, area of the silicon, propagation delay and power dissipations take major roles [1]. These criteria are need to be considered all together, calculated and compared together to decide which technology of provided adders have most efficient role in computing unit.

Key words: Carry-skip Adder, Carry-select Adder, Kogge-Stone Adder.

I. Introduction

Different types of adders have different characteristics both advantage and disadvantage while performing on any device as mentioned above and a single adder cannot have given characteristics all at once[2]. Therefore, it is required for the circuit designer to reduce the disadvantage to minimum and maximize its advantage over other types of adders.
The Adders which are compared in this project are: Carry-skip Adder, Carry-select Adder, and Kogge-stone Adder (all in 8bit, 32bit and 64bit units).
Types of adders are divided into three sections and each of adders will be compared in detail in section 5. Each adder is studied solely on section 2,3 and 4 in detail to figure out their characteristics and find out their outstanding aspects over other types of adders. Lastly the final result of selecting best adder will be discussed on section 6 as conclusion.

II. Carry-skip Adder

Static CMOS logic is popular and common, modern design. Each logic contains pull up and pull down networks and these network's functionality is controlled and decided by input signal [1]. Usually the pull up network is built by pmos device (p channel MOSFET). This makes sure that both networks are not 'ON' at same time and ensuring no static power consumption.

Figure 1: Design of 8-bit Carry-skip Adder

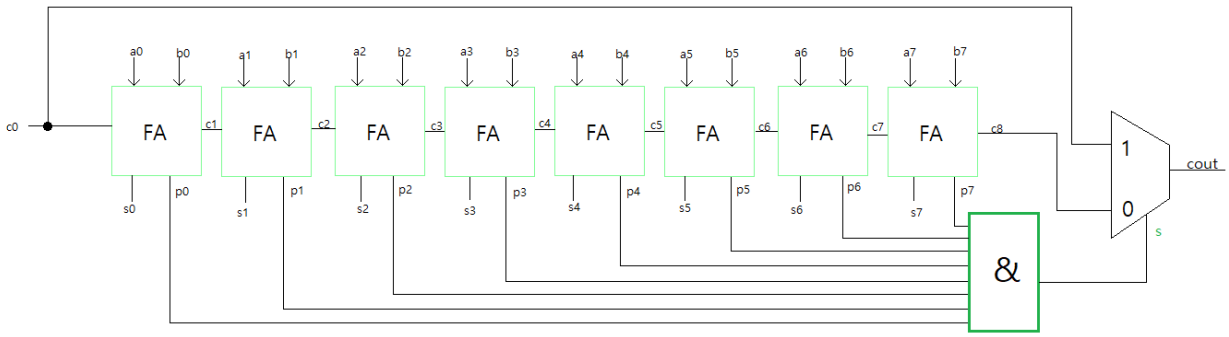
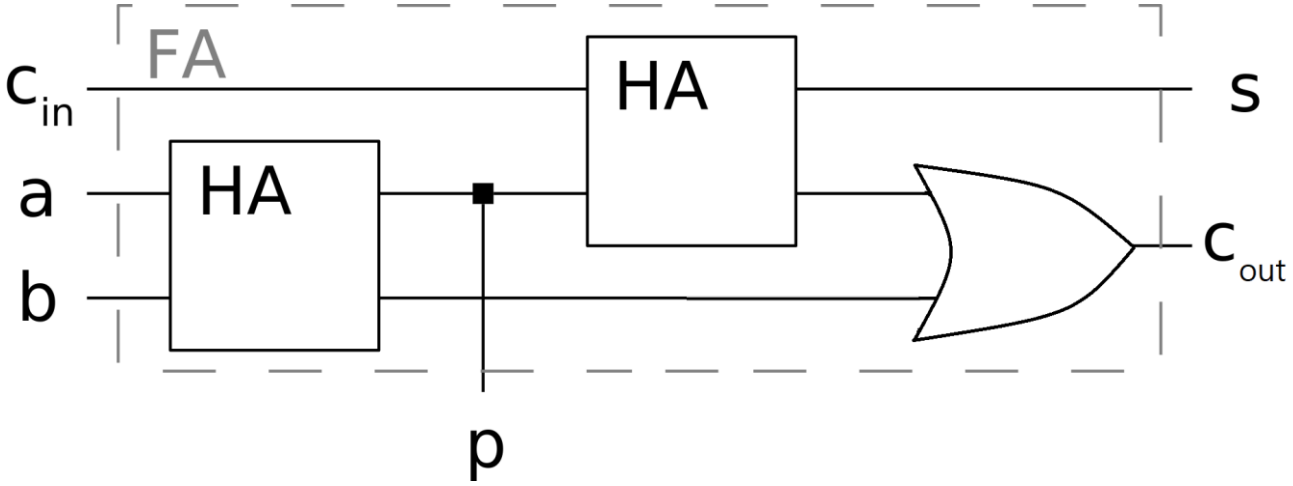


Fig 1 shows the architecture of 8-bit Carry-skip adder.

In carry skip adder :-

1. Given that both input for 1 bit is not equal we don't have to compute value for carry in that block (one Full adder). Since their sum is always 1. No matter what the carry in for that block is it can skip the block and propagated directly to the next bit block.
2. If both inputs are 1 the carry will be produced in which may be propagated up to the output of that group.
3. When both inputs are 0 carry cannot be generated. The carry-skip adder is designed to detect carry in to skip the blocks if it is needed to be skipped, it can be visualized from figure 1.

Figure 2: Design of Full Adder Block used in Carry-skip Adder



As it is visible in Figure 1, each of Full Adder blocks consist of extra outputs as p. this gives propagate signal as output to decide whether or not to skip the carry that has been input to the entire block. Figure 2 shows how this Full adder is modified to design each Full adder blocks in Figure 1.

The carry skip adders are made in blocked to produce various number of bit adders. For example, if we need to build 16-bit carry skip adder for better design that shown in Figure 1. We can group the bit groups into several blocks and save more time of computing all the bits one by one. Figure 3.1 and 3.2 shows how this division of various logic blocks can reduce the propagation time and each adder functioning simultaneously to reduce critical (total) time of circuit [6].

Figure 3.1: 4 of 4-bit blocks PG network

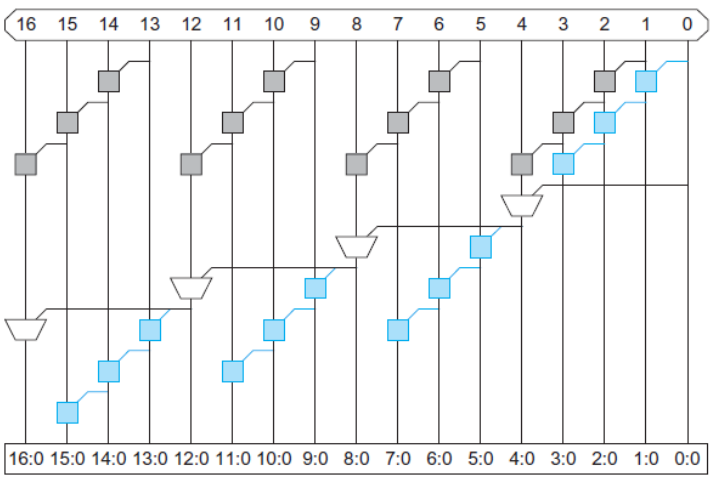
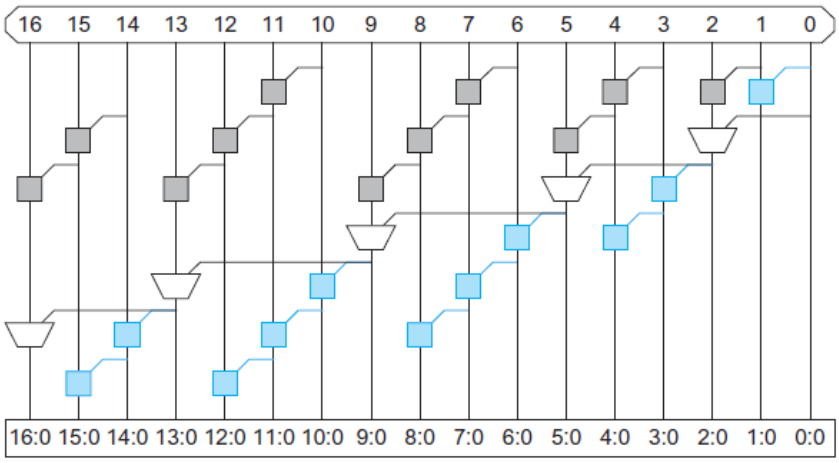


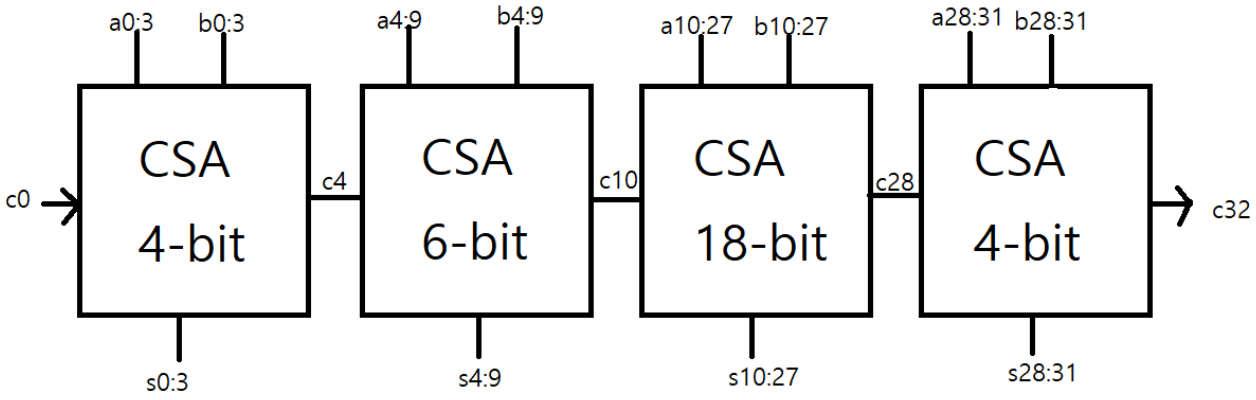
Figure 3.2: 2-3-4-4-3-bit blocks PG network.



A. 8bit adder

8-bit carry skip adder could be built by using similar logic. However, 8-bit carry skip adders is too small to group bits variously hence we could simply group 2 bits as one block. Totally 4 blocks are used to build the 8-bit carry skip adder [1] [7]. The delay time of 4*2bit carry skip adder was 16.9ns [8]. It is stated that Carry Skip Adder provides better results in generating carry output signal when compared with Ripple Carry Adder, whereas the area occupied is larger than that of Ripple Carry Adder.

Figure 4: 32-bit Carry-skip Adder in Various FA Block



B. 32bit adder

32-bit Carry skip adder we can use 4 blocks of 8-bit logic blocks where carry out of first logic block can be directly put to the skip logic block of second 8-bit block [3]. By doing so, 4 of 8-bit logic blocks will be simultaneously functioning and reduce time compared to 32-bit ripple carry adder [4]. Since the logic of carry skip adder is based on ripple carry adder and carry look-ahead adder. The carry computing unit significantly minimize the delay by skipping over the full adder groups.

Due to the skipping logic, the blocks are built in various size. This case, the performance can be improved. All carries propagated more quickly by varying the blocks. Accordingly the initial blocks of

the adder are made smaller so as to quickly detect carry generates that must be propagated the furthers, the middle blocks are made larger because they are not the problem case as the carry is skipped in middle blocks, and then the most significant blocks are again made smaller so that the late arriving carry inputs can be processed quickly [3].

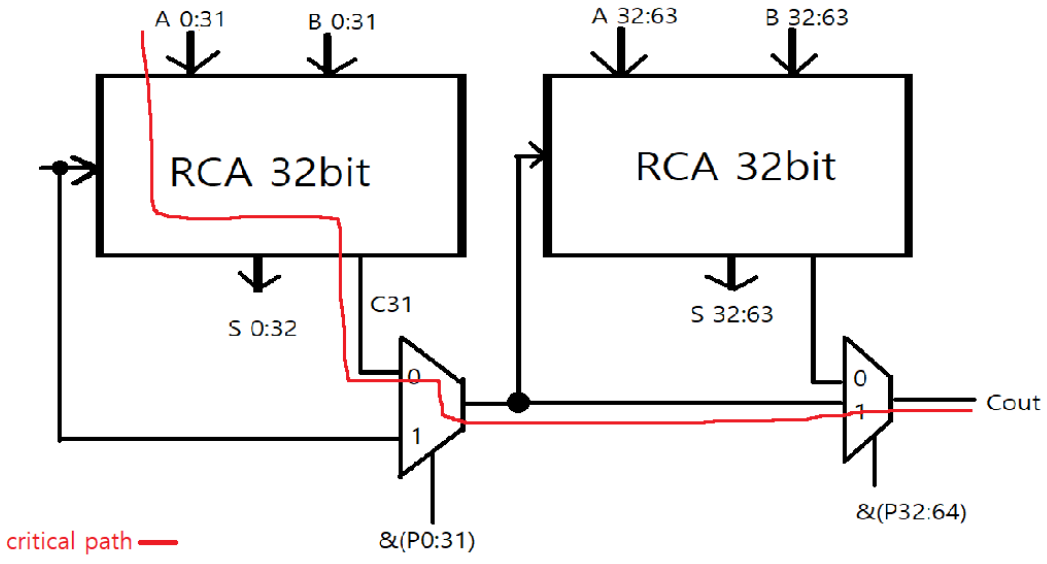
Figure 4 shows how the 32-bit adder can be split into several FA blocks as mentioned above. This design of adder was implemented by Chirca [5] and achieved high performance with lower power dissipation. The adder was divided into variable size blocks which could balance the inputs and the carry chain in whole circuit. This is 32-bit adder with delay of 7 logic levels divided into 4 blocks. It was mentioned that this circuit was built by AOI and OAI CMOS gates were used to reduce delay and power. It resulted in 4.15 ns of delay and 4.68 mw of power dissipation. Therefore, giving result of 19.4 of Power Delay Product.

Looking at the simulation result of Nagaraj. For 32-bit carry skip adder, the number of transistor was 2032 and power dissipation was 58.4nW the delay in this circuit came out to be 0.13765ns.

C. 64bit adder

64-bit adder's size would be much larger compared to that of 8-bit adder. Hence it would utilize more silicon area and the time taken will increase compared to smaller N-bit adders. Kumari has built and simulated several types of N-bit and 64-bit adders in which carry skip adder used simple logic of two of 32bit logic blocks grouped together to build it [9]. The critical path for the given adder would first pass through the 32-bit ripple carry adder where upper 32-bit would be functioning at same time. Hence. The 33rd carry bit would bypass upper 32-bits and directly input to the multiplexer on next logic block. Figure 5 shows how the circuit was implemented by Kumari [9].

Figure 5: Implementation and Critical path of 64-bit Carry Skip Adder

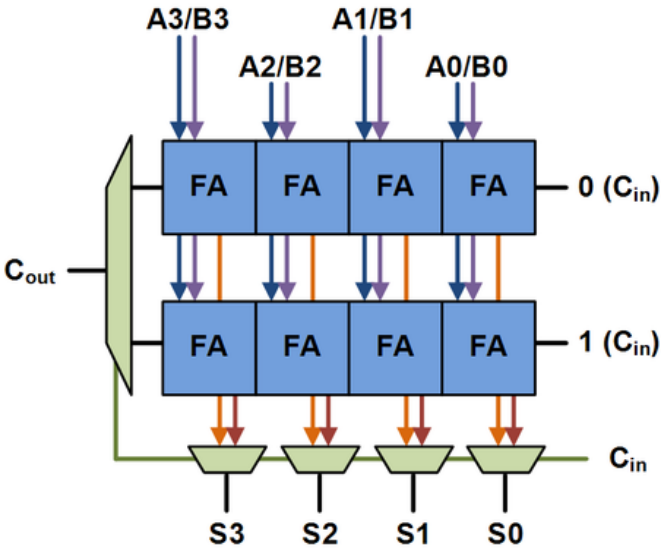


The result of this simulation came out to be 92.12ns of delay time. This result was gained by delay of first 32 full adders with delay of multiplexer. However, this delay time could be decreased with different building mechanism for the circuit. As it was mentioned above, with help of various adder blocks the total logic computing time would decrease. Such as 8 of 8-bit adders built together or even 16 of 4bit adders joined together. Regardless, we can see from the comparison of other Adders in larger number bit carry bypass adders do not perform very well to others [9]. Hence it would be best to utilize carry skip adders in lower number of bit adders to provide maximum optimized performance considering power, silicon area and delay time.

III. Carry-select Adder

The carry select adder contains two Ripple-carry adders and a multiplexer. It calculates addition of 2 n-bit groups twice by involving two adders, one with the assuming carry-in as zero and the other time as one. At next level with multiplexer which is given with correct carry-in known as selector the result sum bit signals and carry out bit signals will be selected. Basic building block of carry select adder is visualized in Fig 6.

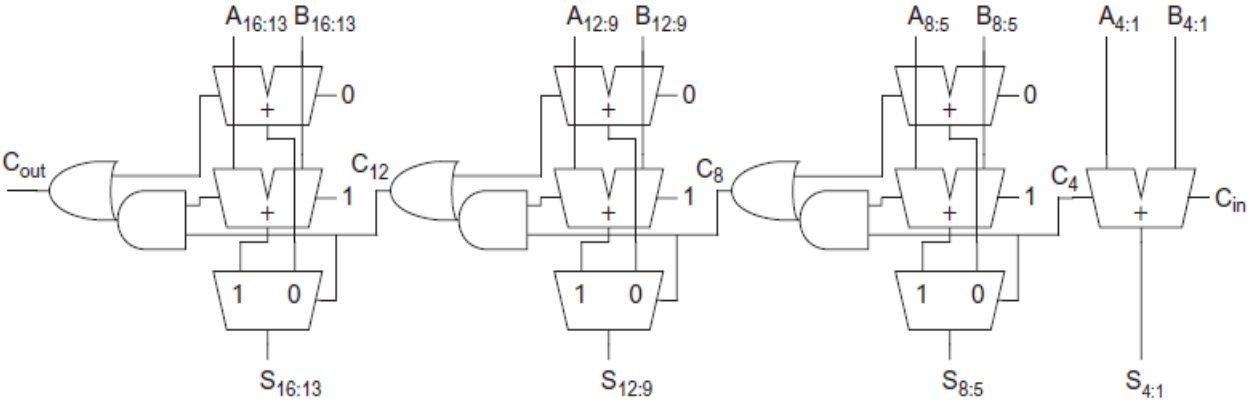
Figure 6: Basic building block of carry select adder



The block size of figure 6 is 4. 2 of 4-bit ripple carry adders are multiplexed together. The resulting carry and sum bits are selected by the carry-in. since both zero and one is input to simultaneous ripple carry adders the multiplexer is able to pick right number for the sum for each bit with help of correct carry-in.

carry select adder can be either have uniform block size or variable block size. Figure 7 shows three of uniform block size of 4bit put together with one 4 bit adder. Since the first bit of carry-in is provided always at the beginning of computation. It is not requied to put multiplexer to select sum and carry.

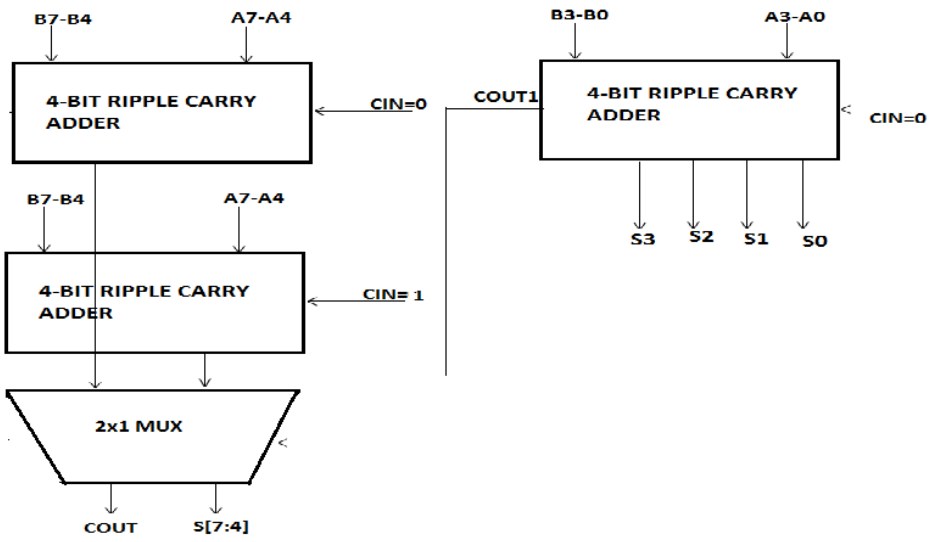
Figure 7: Carry select Adder.



The common structure of large carry select adder contains mainly 4 blocks. For any N bit carry select adder the first block is usually the N/2-bit adder, second and third block are also N/2-bit adders but these 2 adders provide with different carry-in computes simultaneously with 0 and 1. Lastly a multiplexer is put along with second and third block provided with correct carry out from the lower N/2-bit block. By doing so, multiplexer is able to choose the right upper N/2bit sum and carry out for last bit.

A. *8-bit adder*

Figure 8: 8-bit Carry Select Adder



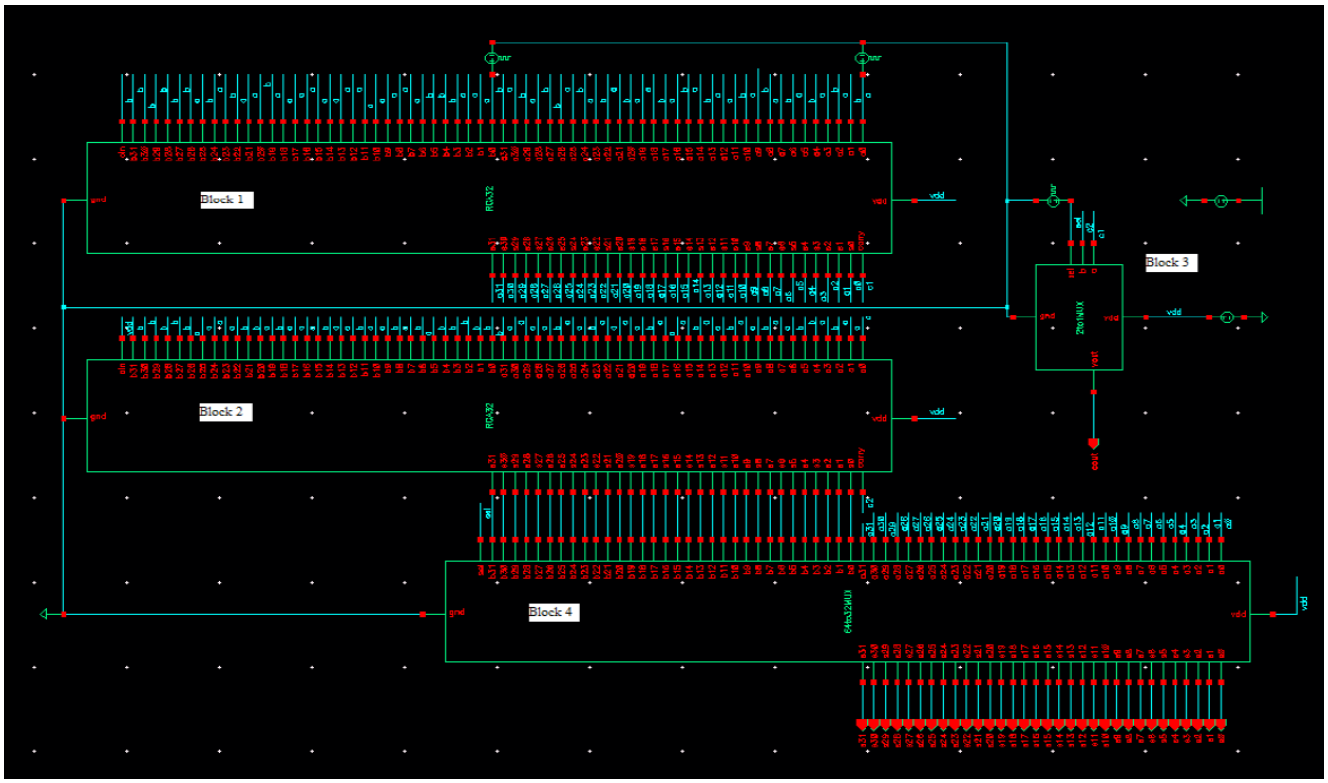
For 8-bit carry select adder, the circuit can be designed as mentioned above. (4 different blocks to build the whole circuit. Design of the usual 8-bit carry select adder is shown in figure 8.

Carry select adders use more silicon area always than carry skip adder as it is visualized. Carry skip adders use only N bit number of adders to compute the addition while for carry select. Since it needs to be selected between two same sums with different carry in it required more adders always. Hence resulting in more transistor being required and larger area. But the speed will be faster as logic level is less than that of carry skip.

B. *32-bit adder*

From the simulation of Nagaraj 32-bit carry select adder requires 3884 transistors and power dissipation was 115.9nW. The delay of given design was similar to that of carry skip adder which was 0.13603ns [7]. Speed, area and power of these adders are simulated using HSPICE for 180nm. Whereas the research by Rashmi used modified design of 32-bit carry select adder to reduce the number of transistors. The number of transistors are reduced to 998 and power consumption was 68.81mW with utilizing simulation from cadence virtuoso 180nm CMOS library [11]. Figure 9 shows the circuit design of 32-bit carry select adder built by Rashmi [11].

Figure 9: 32-bit Carry Select Adder Circuit Design



C. 64-bit adder

The Research and simulation from Palanirajan well show the efficiency of 64-bit carry select adders [10]. The modified design of carry select adder was built and it was able to achieve in reduction of area and reduce power dissipation with minimum penalty in delay time. For 64-bit Carry select adder delay has been increased by 3.185ns from 34.549ns resulted in 9% and power decreased by 0.72mW only (which was less than 0.2%) [10]. This exchange of characteristics of circuit would make this modified circuit to be less efficient.

However, looking at the area of modified and original version of circuits it was 404 areas in whole but it has been reduced to 295. This exchange sounds quite reasonable because the loss in delay was <20% yet, the gain of advantage in silicon area is >25%.

This exchange was achieved by building circuits with slight modification. Instead of using two ripple-carry adders simultaneously Palanirajan utilized Binary to excess one converter with ripple carry adder to reduce silicon area with penalty in delay time.

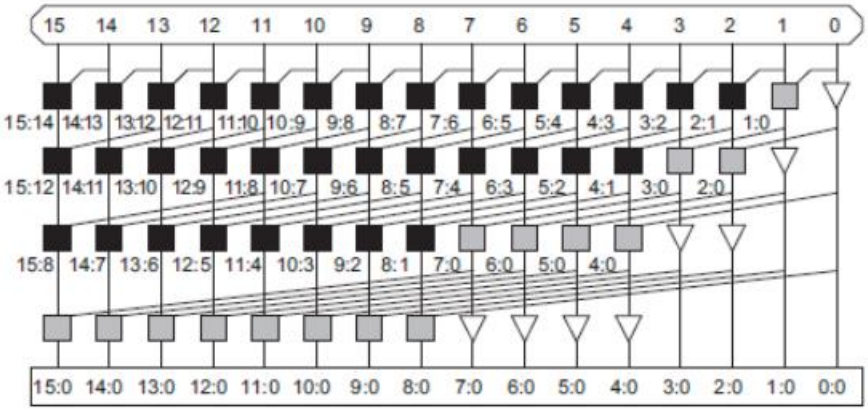
IV. Kogge-stone Adder

Kogge-stone adder design was quite different from above to adders. It falls under group of parallel prefix adder. The concept of this adder was developed by Peter M. Kogge and Harold S. Stone. Kogge-stone adder uses concept of generating and propagating carriers. It is usually very fast in performance and focused on design time. Throughput is optimized in Kogge-stone adder with minimum logic depth and restricted fan-out.

Various studies address that Kogge-stone adders are very fast in comparison with other adders. This means that their delay time is less compared to other adders, likely the adders mentioned above [2][8]. Parallel prefix adders often achieve faster speed, likely in O(log2n), compared to Ripple carry adder-based implementation, likely O(n), such as carry skip and carry select that was mentioned in previous

sections [13]. Parallel prefix adders, when designing and implementing the circuit it employs tree network to minimize latency of adding units. Nevertheless, there is trad- off for silicon area[8][13]. The circuits are built more complex for Kogge-stone adder to compromise with delay from carry being rippled. This ensures reduction in speed as the circuit do not have to wait for other lower bit block of adders to produce carry out, which is common in carry-skip and carry-select adders. The N-bit addition computing is done almost simultaneously in Kogge-stone adder with less logic level. Figure 10 displays the schematic of 16-bit Kogge-stone adder. It is observed that the logic level is much simpler compared to the schematics of ripple carry adder based adder circuits.

Figure 10: Kogge stone adder schematic (16-bit)



A. 8-bit Adder

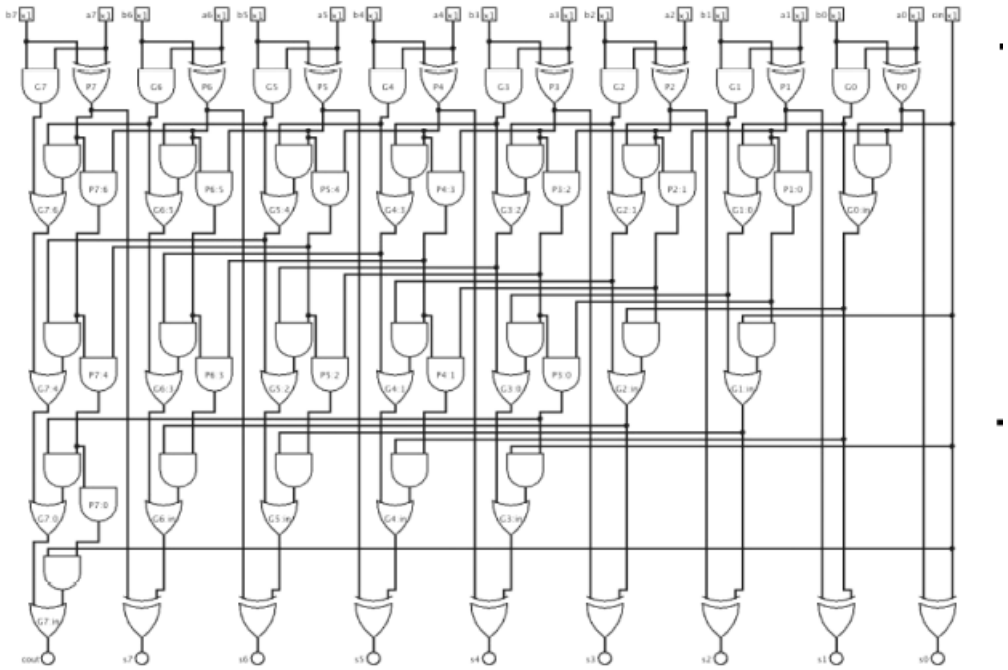
In various research for Kogge-stone adders, often in comparison with others, it is visible that Kogge stone adders are very fast computing adder. Study case done by Shinde implies that 8-bit Kogge stone adder achieved 15.8ns of delay but due to its complexity, the adder requires more slices of silicon resulting in extra area [8]. Another simulation performed by Shilpa shows that the 8-bit Kogge stone adder utilized 570 transistors with delay of 71 ns and 27.28mW of power dissipation. This study was performed and simulated with 180nm technology for CMOS using cadence [15]. Figure 10 is diagram of how the 8-bit Kogge stone adder is designed [8].

B. 32-bit Adder

Kogge-stone adders are used often in larger number of bit adders to reduce time delay to match with the clock cycle [13-15]. Since all the addition of each bits are done in parallel, the change in speed is not doubled in doubled number of bit adders. It gives fastest result in single blocks but power dissipation can be optimized with a greater number of blocks. Study done by Moudallal shows the relationship between compromised speed and power dissipation in Kogge-stone adder design models [14]. 90nm CMOS circuit design and HSPICE simulation was adopted for given adder.

The results showed that 2 cascaded blocks in ripple structure performed most well compared to one or 4 blocks. Power dissipation in this circuit was 19μW and time delay was 740ps. However, here was compromise in speed compared to 1 block adder where time delay was 469.4ps [14]. It is understood that the design can be adapted according to the need of computing unit either for speed or power. In addition, the area that the circuit needed reduced as number of blocks increased. This would be due to the smaller units of adders in parallel makes the design even more compact.

Figure 11: 8-bit Kogge stone adder design



C. 64-bit Adder

Its visible from Moudallal's study that larger Kogge stone adders may take more that double of silicon area compared to N/2-bit Kogge stone adder, and also power dissipation could increase in large margin. Nevertheless, the increase in delay for 2N-bit adders compared to N-bit adders, in Kogge-stone logic, do not increase by very big margin.

The 64-bit adder took 693.11ps of time delay to produce result, where the power dissipation came out to be 57.97µW. The simulation was again carried out with different setting of 2*32-bit Kogge stone adder with one ripple unit. As the result the delay was 1014ps and power dissipation was 51.33µW [14]. Same was carried out with 4*16-bit, but the power-delay product is less efficient compared to 2*32-bit adder. However, since it was similar to the single 64-bit Kogge stone adder, and consumed less area. It could be utilized accordingly.

V. Comparison of 3 Adder Architectures for 3 Different Operand Sizes

As it is discussed above, all three adders have advantages and disadvantages over others. The general ideas of each adders are considered for each cases and decision is made for various characteristic of the adders.

A. 8-bit adder

Study from Shinde shows that Kogge stone adder was ~5% faster than the carry skip and carry select adders but area utilization was doubled compared to other to adders [8]. Power dissipation was also not very different shown in research done by Harish [16]. Xilinx ISE 2014.7 tool was used to produce comparison of data, the Kogge-stone adder used 1.221µW and other two used 1.186µW which is around 2% of difference. Since for 8-bit Kogge-stone doesn't look very efficient, but still can be utilized to save even more time, carry select and carry skip need to be compared. for 8-bit adder carry skip adder required more space and the speed was also slow compared to that of carry select adder. Hence for 8-bit adder carry select adder would be most optimized.

B. 32-bit adder

Coming to the comparison of the 32-bit adder the research and simulation done by Gaur can be taken closer look to see the power dissipation difference [17]. The total power usage of carry skip, carry select and Kogge-stone adders are 0.125W, 0.124W, 0.125W respectively. It is observed that power consumption is not beneficial for any of these adders as one adder is not outstanding among all. Hence the area and time need to be considered to decided which adder is best. Study by Saini well explains about the area difference of carry select adder and Kogge-stone adder. The number of components is compared in the study and this could be utilized for general patter of N-bit adder area utilization. Area consumption is large for the carry select adder [18]. Even in the study done by Nagaraj shows that carry select used 3884 transistors in 32-bit adder and 2032 transistors for carry skip adder. However, the delay was <1% differed in these two adders. Therefore, it is understood that carry select adder is only well utilized for smaller number of bits [7]. For 32-bit adder carry-select adder is worst optimized and carry skip or Kogge-stone could be utilized under consideration of area and speed compromise. Kogge-stone adders are slightly faster and carry skip use less area respectively [2] [7-9] [16].

C. 64-bit adder

Similarly, in case for 64-bit adder comparison the carry select adder is crossed out as it was observed worse for larger N-bit adders. The delay comparison done by Kumari displays big difference in carry skip adder and Kogge-stone adder [9]. The delay outcome of the simulation was 92.12ns and 28.526ns for carry skip and Kogge stone respectively. This shows very large reduction of speed in case of larger N-bit adders, for Kogge-stone adder.

As for parallel prefix adder such as Kogge stone adder is much more useful in larger number of bit addition. Study done by Penchalaiah observed that for 64-bit or higher number of bit addition units, comparing carry skip adder and Kogge stone adder. It was observed that Kogge-stone adders utilized minimum energy consumption along with area compaction and higher speed. Hence it can be summarized that Kogge-stone adder is most optimized in 64-bit adder or even larger scaled adders, among all three adders compared in this research

VI. Conclusion

When designing and adopting adder for utilization in any required environment the engineer needs to consider the effectiveness of the pre-provided design and decide which one to be used in system.
For example, if small system is to be built the area need to be number one consideration, then power and speed respectively so often ripple carry adder based adders could be adopted. Suppose for 4-bit or 8-bit addition carry skip or carry select adders requires less space compared to parallel prefix adders as it is more complex in such cases. For larger scaled addition logic such as 32-bit or greater bits, the speed, area and power dissipation all considered together, Kogge stone adder showed best result in this research. Therefore, it is crucial for any engineers while choosing which is optimum adder for any system, the speed power and area need to be taken under consideration and cautiously approach to meet the requirements of system.

VII. References

[1] P. P. Patil and A. A. Hatkar, "Comparative analysis of 8 bit Carry Skip Adder using CMOS and PTL techniques with conventional MOSFET at 32 nanometer regime," *2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, Delhi, 2016, pp. 1-5.

[2] B. Koyada, N. Meghana, M. O. Jaleel and P. R. Jeripotula, "A comparative study on adders," 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, 2017, pp. 2226-2230.

[3] Y. S. Lin and D. Radhakrishnan, "Delay Efficient 32-bit Carry-Skip Adder," 2006 13th IEEE International Conference on Electronics, Circuits and Systems, Nice, 2006, pp. 506-509.

[4] S. Chang and C. Wey, "A Fast 64-bit hybrid adder design in 90nm CMOS process," 2012 IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS), Boise, ID, 2012, pp. 414-417.

[5] K. Chirca, M. Schulte, J. Glossner, H. Wang, S.Mamidi, P. Balzola, and S. Vassiliadis, "A Static low-power, high-performance 32-bit carry skip adder," *Proc. EUROMICRO Systems on Digital System Design (DSD'04)*, pp. 615-619, March 2004.

[6] N. H. E. Weste and D. M. Harris, *CMOS VLSI Design – A Circuits and Systems Perspective*, 4thEdition, Pearson, 2011.

[7] S. Nagaraj, G. M. S. Reddy and S. A. Mastani, "Analysis of different Adders using CMOS, CPL and DPL logic," 2017 14th IEEE India Council International Conference (INDICON), Roorkee, 2017, pp. 1-6.

[8] K. D. Shinde and S. Badiger, "Analysis and comparative study of 8-bit adder for embedded application," 2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kumaracoil, 2015, pp. 279-283.

[9] N Aruna Kumari, M Sai Srinivas, K Aravind "Implementation of 64-bit Arithmetic adders" International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-7, Issue-5S4, February 2019. pp. 437-444.

[10] Gowtham Palanirajan, "Design of carry select adder with area and power efficiency" 2013.

[11] Rashmi S.B and V. Oli, "32 bit power efficient carry select adder using 4T XNOR gate," 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), Bangalore, 2016, pp. 283-287.

[12] U. Penchalaiah and S. K. VG, "Design of High-Speed and Energy-Efficient Parallel Prefix Kogge Stone Adder," 2018 IEEE International Conference on System, Computation, Automation and Networking (ICSCA), Pondicherry, 2018, pp. 1-7.

[13] B. Tapasvi, K. B. Sinduri, B. G. S. S. B. Lakshmi and N. U. Kumar, "Implementation of 64-bit Kogge Stone Carry Select Adder with ZFC for efficient area," 2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), Coimbatore, 2015, pp. 1-6.

[14] Z. Moudallal, I. Issa, M. Mansour, A. Chehab and A. Kayssi, "A low-power methodology for configurable wide kogge-stone adders," 2011 International Conference on Energy Aware Computing, Istanbul, 2011, pp. 1-5.

[15] C. N. Shilpa, K. D. Shinde and H. V. Nithin, "Design, Implementation and Comparative Analysis of Kogge Stone Adder Using CMOS and GDI Design: A VLSI Based Approach," 2016 8th International Conference on Computational Intelligence and Communication Networks (CICN), Tehri, 2016, pp. 570-574.

[16] B. Harish, K. Sivani and M. S. S. Rukmini, "Design and Performance Comparison among Various types of Adder Topologies," 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2019, pp. 725-730.

[17] N. Gaur, D. Tyagi, Deepika and A. Mehra, "Performance comparison of adder architectures on 28nm FPGA," *2016 2nd International Conference on Advances in Computing, Communication, & Automation (ICACCA) (Fall)*, Bareilly, 2016, pp. 1-5.

[18] J. Saini, S. Agarwal and A. Kansal, "Performance, analysis and comparison of digital adders," 2015 International Conference on Advances in Computer Engineering and Applications, Ghaziabad, 2015, pp. 80-83.