32-bit Kogge-Stone based Hybrid Adder Implemented using Standard Cells of Different Logic Families

Hema Singaravelan
Graduate Student
RV College of Engineering
Bangalore

Dr. Kiran V
Associate Professor
RV College of Engineering
Bangalore

Abstract: *Adders performs a critical role in all computational operations, thereby optimizing them with respect to design constraints for a system is essential. In this paper, standard cells of different logic families, namely- CMOS, Pseudo NMOS, and MGDI, are designed in Cadence Design Suite Virtuoso 6.1.7 in 180nm technology and characterized using Liberate 15.1.3. The standard cell libraries thus created are then applied to 32-bit KSA (Kogge-Stone Adder) and KSA based proposed hybrid adder that are implemented in Verilog, functionally verified on Xilinx Vivado 2020.2 and synthesized on Cadence Genus 15.22. Pseudo NMOS logic shows 14.03% area savings and MGDI offers 54.43% power saving based on area per cell over the traditional CMOS technology. It is also seen that the proposed adder offers a decrease in power and delay by 32.13% and 13.75% over KSA, respectively, in CMOS logic. Further discussions are made and suitable applications for all designs are also discussed.*

Keywords: **CMOS, Pseudo NMOS, MGDI, Kogge-Stone Adder, Cadence Design Suite, Virtuoso, Liberate 15.1.3, Genus 15.22, Xilinx Vivado 2020.2**

1. INTRODUCTION

Adders are the fundamental building blocks for all arithmetic and logical operations performed by the ALU (Arithmetic Logic Unit). The design of these modules will eventually determine the performance of the system. Thus, based on requirements, adders must be designed to function efficiently, either prioritizing area, speed, scalability or power, or optimizing all parameters. High-performance adders typically use a parallel prefix tree to compute the group generate and the group propagate signals that compute the intermediate and final carries along with the resultant sum bits. Therefore, among the several adder designs, the Parallel Prefix Adder (PPA) designs are the most preferred for their higher speed of operation. In the past few decades, several algorithms for addition were proposed aiming at enhancing the computational efficiency of PPAs.[6] The Kogge-Stone adder is widely used as the fastest adder since the carry generation is done in O ($\log_2 N$) time where n is the number of bits of each of the adder inputs. Kogge-Stone is thereby considered in this study to enhance the speed of carry propagation through the adder circuit while optimizing the area and power consumption overheads.[1]

In addition to the optimizations offered by the 32-bit KSA designed in this study, standard cells for Inverter, NAND and NOR gates were developed using traditional CMOS, Pseudo NMOS and MGDI techniques.[2] The need for this study is to perform a comprehensive analysis of the performance of the parallel prefix adders with respect to standard cells designed so that we may have data to comment on the performance and the associated application of the specific design. The following sections detail the parameters and design constraints used in the study along with brief explanations on the modules and devices used.

2. Adder Architectures

2.1. Kogge-Stone Adder

KSA performs the key role of fast addition operation and is often referred to as the prefix form of Carry Look ahead Adder (CLA). It entirely decreases the delay time in design to generate the carry signals [3] making it a popular choice in DSP (Digital Signal Processing) applications and Control System Industries for fast arithmetic and logic functions. The structure of a 32-bit KSA is detailed in Figure 1. The first two stages consist of the calculation of Propagate and Generate signals using the respective logic gates facilitating carry input. The third and fourth stage involve the generation of Carry Propagate and Carry Generate signals. The Final stage computes the sum bits based on the Carry Propagation and carry Generation values. [1]

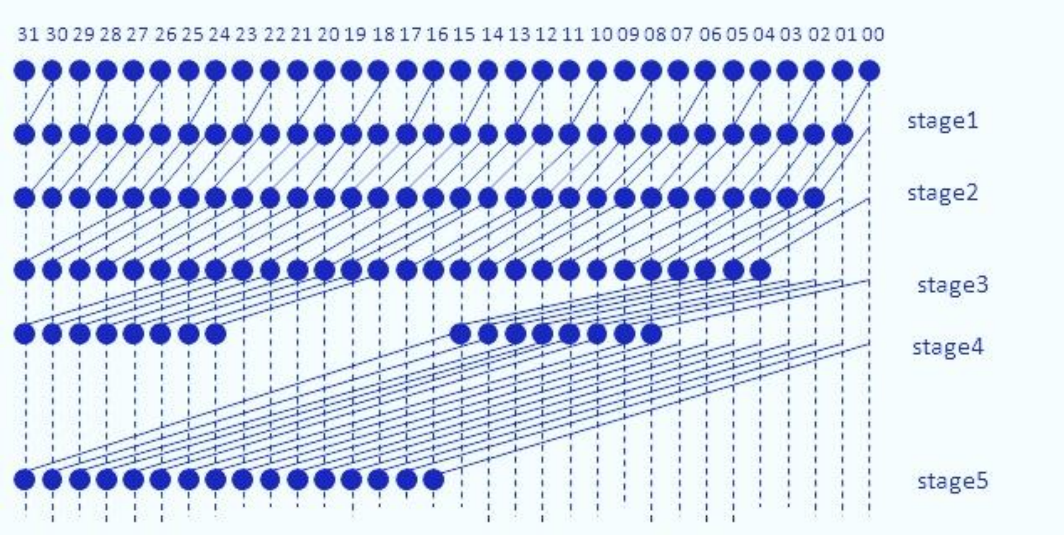


Figure 1. 32-bit Kogge-Stone Adder

There are three main computational steps involved in the design of a Kogge-Stone Adder, namely [1],

1. Pre-Computation

This stage involves the computation of the Generate and Propagate signals from the inputs which is then passed on to the next stage. The Group Generate signal (1) is computed as,

$$G[i] = A[i] \cdot B[i] \tag{1}$$

where the AND operation of each individual bits of inputs A and B are computed, with i= 0,1,2…,31, indicating the number of input bits and the associated generate signal. Similarly, the Propagate signal is computed (2) as,

$$P[i] = A[i] \oplus B[i] \tag{2}$$

Where the XOR operation of each individual bits of inputs A and B are computed, with i= 0,1,2…,31, indicating the number of input bits and the associated generate signal.

It can be observed that the Propagate and Generate signals are computed parallelly, thereby aiding in increasing the speed of computation, simultaneously decreasing the area of implementation as these signals are computed using basic logic gates in the pre-processing step.

2. Prefix Computation

This stage calculates the carry signal groups directly – reducing the carry propagation delay and reflecting the functions of CLA – using the Propagate and Generate signals obtained from the first stage. The Carry Propagation (3) and the Carry Generation (4) functions are as given below,

$$CP[i] = P[i] \cdot P[i+1] \tag{3}$$

$$CG[i] = (G[i] \cdot P[i+1]) + G[i+1] \qquad (4)$$

It must be noted that the Carry Generation signal utilizes more than two inputs thereby increasing the propagation delay, whereas the Carry Propagation utilizes only two inputs, comparatively reducing the delay.

3. Post-Computation

This final stage computes the sum bits of KSA through the XOR of the Carry Propagation signal and the previous carry signal. This is calculated as in the equation,

$$S[i+1] = P[i+1] \oplus C[i] \qquad (5)$$

2.2. Carry Select Adder

The carry select adder generally consists of two ripple carry adders each computing the sum of two N-bit numbers with carry input zero and carry input one, respectively. The output of this adder is then obtained by supplying the actual carry input a select line to a series of 2:1 multiplexers taking the two types of calculated sums and carry outs as inputs.

This structure is detailed in Figure 2.

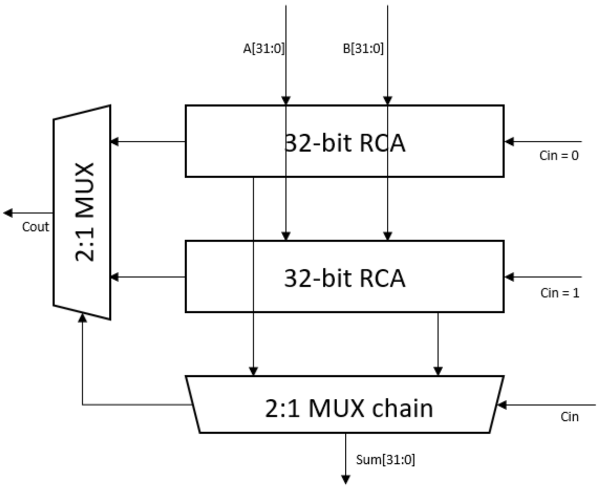


Figure 2. 32-Carry Select Adder

The advantage of this adder block is that the computation of the sum and carry out is performed before the carry in is supplied, thereby if this type of adder is inserted in the later stages of computation, i.e., computing MSBs (Most Significant Bit) of the inputs, this stage would only have to wait for the carry input from the LSBs (Least Significant Bit) of the inputs to produce the sum and carry out, since its computation occurs parallelly with the LSBs computation. This adder architecture offers a delay of O ($\sqrt{N}$) and it is derived from uniform sizing, where the ideal number of full-adder elements per block is equal to the square root of the number of bits being added, as that will yield an equal number of MUX delays.

2.3. Proposed Hybrid Adder

The proposed hybrid adder combines the above architectures of the adder blocks to produce a 16-bit KSA followed by a Carry Select Adder that utilizes 16-bit KSA instead of a 16-bit RCA (Ripple Carry Adder) to compute the sum and carry out bits. Figure 3 details the structure of the model and the result of the hybrid adder was studied in terms of area, power and delay.

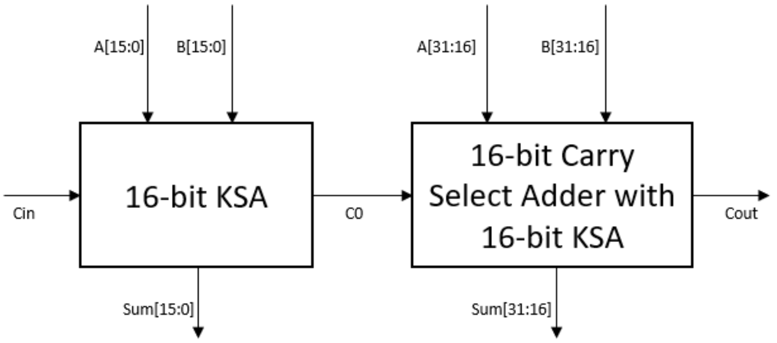


Figure 3. Proposed 32-bit Hybrid Adder

3. Logic Families

3.1. Complementary Metal Oxide Semiconductor

The CMOS (Complementary Metal Oxide Semiconductor) is a commonly used technique in the design of digital circuits. It consists of a PUN (Pull Up Network) and a PDN (Pull Down Network). The PUN consists of PMOS transistors connected to VDD source that will set the output to logic high when activated and the PDN consists of NMOS transistors connected to GND that will set the output to logic zero when activated. Both the PUN and PND networks cannot be simultaneously activated or deactivated, other than during logic switching when the VDD and GND are shorted and we have dynamic power dissipation.
The static power dissipation of CMOS circuits is extremely low in comparison to other logic family gates. The comprehensive general structure of CMOS circuits is given in Figure 4. This structure is usually implemented in the 2:1 PUN to PDN inverter width ratio.

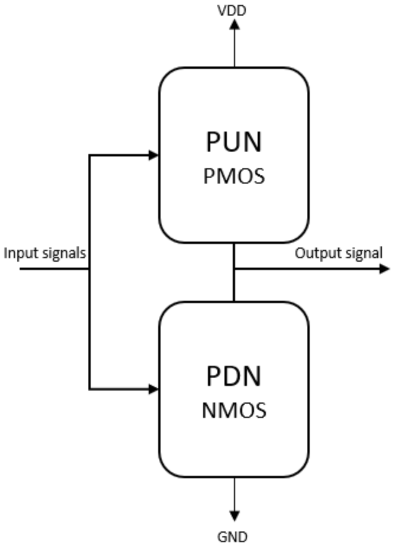


Figure 4. General structure of CMOS logic

The main advantage of this technique is the availability of full swing output and low static power dissipation. Although, it must be noted that the number of transistors is more for the design of each gate, therefore area and power consumption is increased.

3.2. Pseudo NMOS logic

In Pseudo NMOS logic, the PUN is replaced with a single PMOS transistor with its gate input permanently grounded and the PDN network implements the logic function using NMOS transistors. This is roughly equivalent to the use of a depletion load in NMOS technology and it thus termed as Pseudo NMOS logic. The PMOS replacing the PUN is constantly in the linear region, thus the resistance is low and the output is constantly maintained at logic high until the PDN network is activated and the output is pulled to logic level zero. The general structure of this logic is given as (Figure 5),

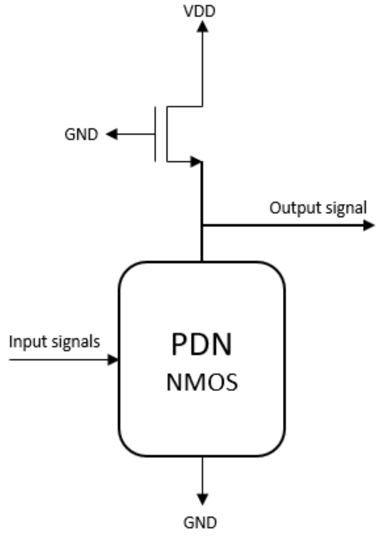


Figure 5. General structure of Pseudo NMOS logic

It must be noted that in order to drain the output capacitance before shorting the VDD to the GND, the transistors must be sized in such a way that the pull up PMOS and the PDN network must be sized in a way that the width of the two must be of the ratio 2:4. It must be noted that the Noise Margin Low is significantly higher than the CMOS logic gates. This technique offers full swing output with lower number of transistors. The main drawback is the static power dissipation at the PMOS transistor that cannot be avoided.

3.2. Modified Gate Diffusion Input Technique (MGDI)

This technique is a low power design which is the modification of the Gate Diffusion Input (GDI) logic. This method is the lowest design technique that is suitable for the design of fast, low power gates using a reduced number of transistors.[4] Figure 6. shows the structure of a basic MGDI cell. MGDI consists of three input terminals - G, (input of both PMOS and NMOS) P, (input to drain/source of PMOS) and N (input to drain /source of NMOS).

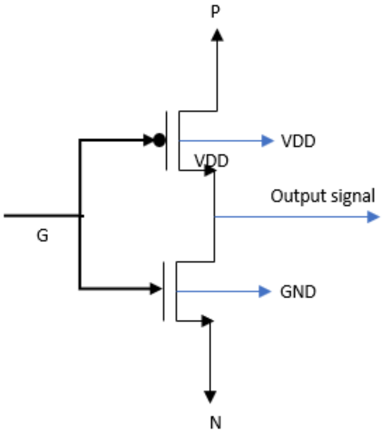


Figure 6. Basic MGDI cell

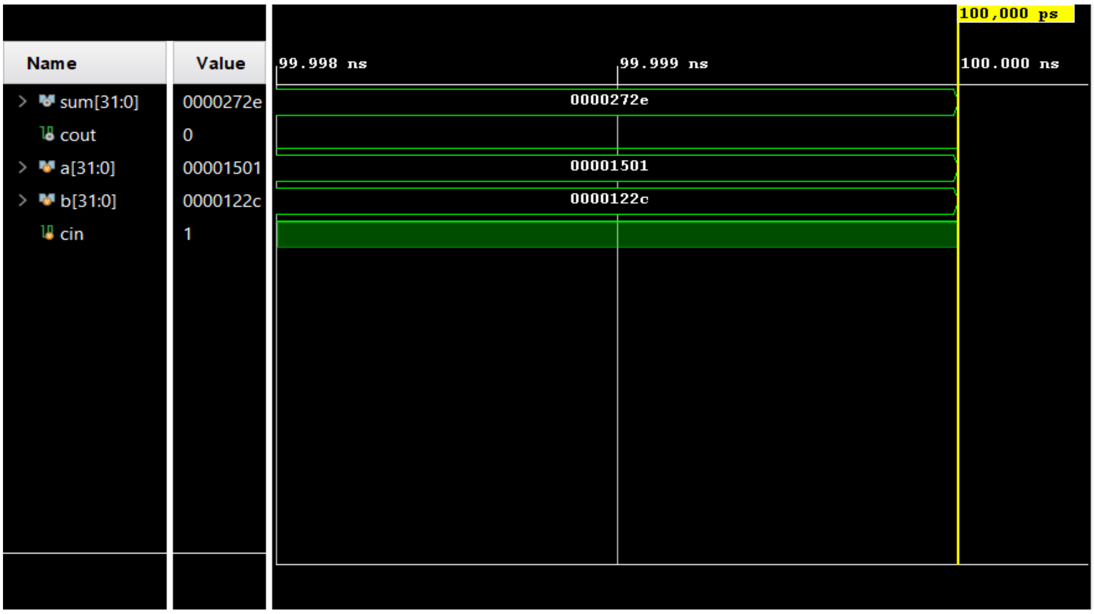
This technique reduced threshold and sub threshold leakage current as compared to CMOS technology. The fabrication of this technique is possible using the traditional pwell progression method. Thus, the substrate is permanently connected to GND and the nwell is connected to the supply and with technology scaling, the influence of source body voltage on transistor threshold voltage gets exceeding abridged. This technique is implemented with the same inverter ratio as CMOS technology, 2:1.

The layout of each of these logic gates were implemented using the multi-finger layout technique to reduce parasitics and delay.[5][7]

4. Simulation Results

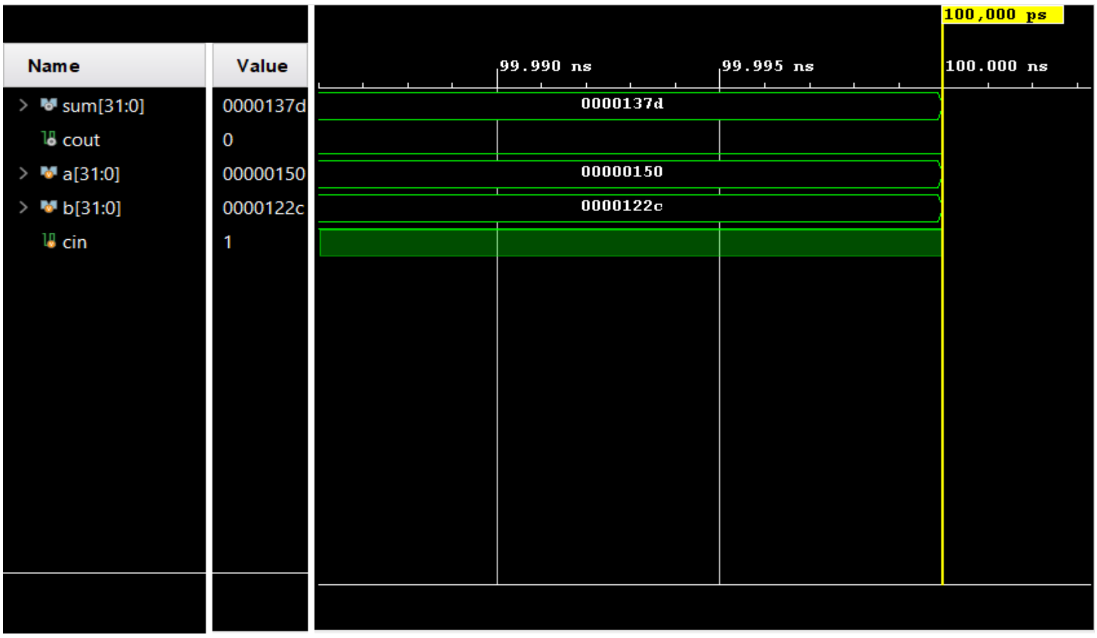
4.1. Design and verification of KSA and KSA based Hybrid Adder Verilog codes

The discussed architectures of KSA and the proposed hybrid adders were implemented using Verilog Hardware Description Language (HDL) and verified on the Xilinx Vivado 2020.2 platform. Figure 7 and Figure 8 represent the functional output of 32-bit KSA and the 32-bit Hybrid adder, respectively.



Name	Value
sum[31:0]	0000272e
cout	0
a[31:0]	00001501
b[31:0]	0000122c
cin	1

Figure 7. Testbench results of 32-bit Kogge-Stone Adder



Name	Value
sum[31:0]	0000137d
cout	0
a[31:0]	00000150
b[31:0]	0000122c
cin	1

Figure 8. Testbench results of 32-bit Hybrid Adder

4.2. Design and verification of Standard cell libraries and synthesis of the adders

The verified codes were then synthesized in Cadence Genus 15.22 using the standard cells characterized in Liberate 15.1.3 after their respective layouts were designed in Cadence Virtuoso 6.1.7. The cells were designed using GPDK 180nm technology library with the Operating Temperature 25^{0}C and Supply Voltage VDD 1.8V. The pitch size was determined as 0.6 µm and the cell height was set at 15µm. The clock period was set at 50ns for 32-bit KSA and the clock period was set at 80ns for 32-bit Hybrid adder. It must be noted that the edge triggered D-flip flops were the same for all three standard cells and were designed in traditional CMOS logic. All other cells were altered to the structures prescribed by the respective logic families.

The generic synthesis of the adder architectures resulted in the following schematics. Figure 9. represents 32-bit KSA and Figure 10. represents 32-bit proposed hybrid adder, respectively.

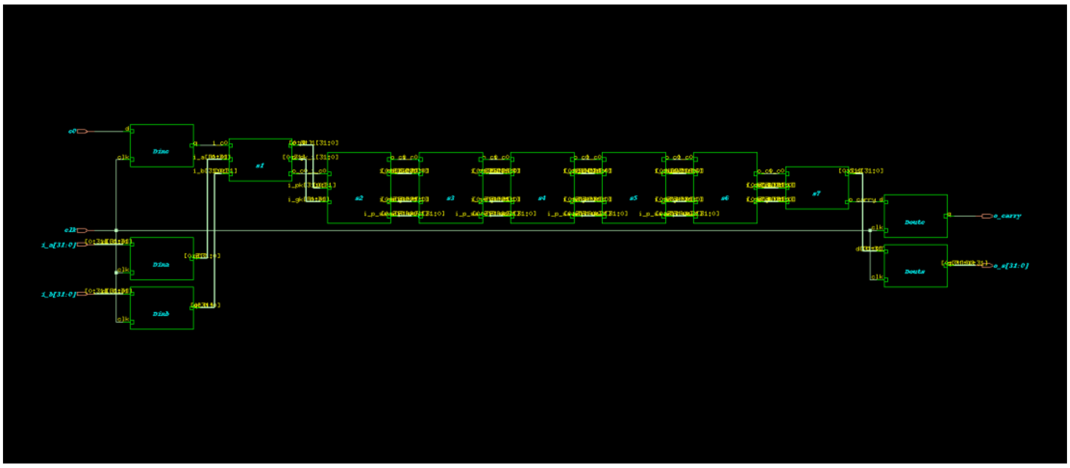


Figure 9. Schematic of 32-bit KSA

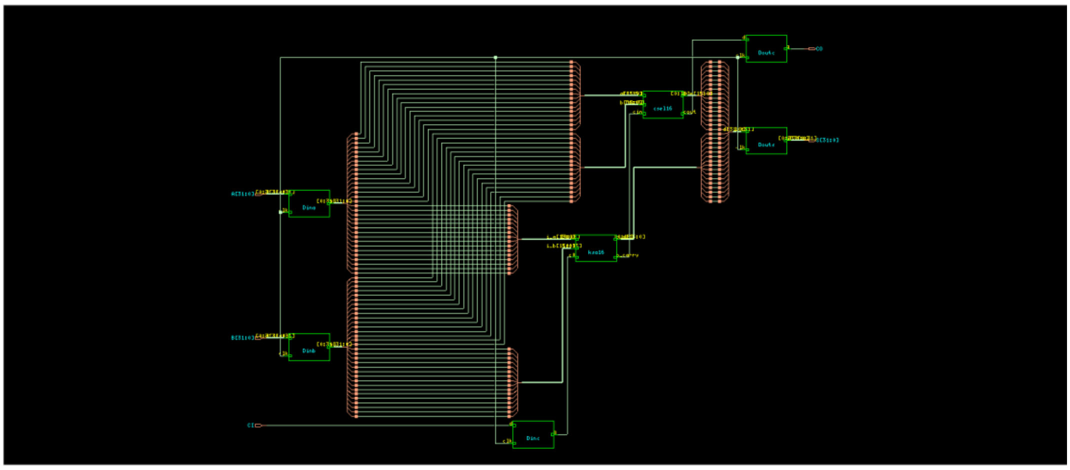


Figure 10. Schematic of 32-bit Proposed Hybrid Adder

4.3. Comparative analysis

The area of the synthesized outputs based on the logic families applied and the respective adder structure is as given in Table 1.

Table 1. Area of the adder architectures based on different standard cell libraries (μm^2)

Logic Families	32-bit KSA	32-bit Hybrid Adder
CMOS	70908.301 (590 cells)	77747.101 (721 cells)
Pseudo NMOS	63769.201 (625 cells)	70860.301 (760 cells)
MGDI	112164.302 (1469 cells)	111822.302 (1449 cells)

The power consumption of the synthesized outputs based on the logic families applied and the respective adder structure is as given in Table 2.

Table 2. Power consumption of the adder architectures based on different standard cell libraries (μW)

Logic Families	32-bit KSA	32-bit Hybrid Adder
CMOS	1355.051	919.659
Pseudo NMOS	283655.838	328260.390
MGDI	3813.373	2417.383

The delay of the synthesized outputs based on the logic families applied and the respective adder structure is as given in Table 3.

Table 3. Delay of the adder architectures based on different standard cell libraries (ps)

Logic Families	32-bit KSA	32-bit Hybrid Adder
CMOS	2895	2497
Pseudo NMOS	3059	2812
MGDI	22563	13289

4.3. Result discussion

The relative area per cell average of the CMOS logic is 114μm^2, Pseudo NMOS logic is 98 μm^2 and MGDI technique is 77.5 μm^2. It can be seen from the above analysis that the relative power consumption of 32-bit KSA adder and 32-bit Hybrid adder from CMOS logic has improved by 54.43% in MGDI with respect to area per cell. It can also be seen that the area is also significantly decreased by 14.03% and delay is the lowest for MGDI technique per cell. It is seen that the number of cells in the MGDI technique is significantly higher than the other logics. One of the reasons for this can be that in order to maintain a stable full swing output, additional gates were introduced in the circuit as buffers, thereby adding to the area, delay and power consumption. A possible solution to this can be implementing dedicated MGDI gates with different drive strengths, noise margins, and logic functions. This would eliminate the redundancy of implementing additional buffers in the circuit to maintain output swing.

In comparison with 32-bit KSA, the 32-bit hybrid adder has an increase in area by 10% in CMOS logic. Power and delay are decreased by 32.13% and 13.75%, respectively in CMOS logic. A similar pattern can be discerned in other logic standard library-based designs as well suggesting the efficiency of the proposed hybrid adder.

5. Conclusion

From the obtained results it can be concluded that the proposed hybrid adder, though increased in area by 10%, provides significant gains in the aspects of power consumption and delay of the design. It can also noted that while the Pseudo NMOS logic reduces the number of gates by a small amount the power loss incurred is a significant quantity, thereby suggesting MGDI as a better alternative. The MGDI technique offers reduced area per gate and power consumption per gate with the only drawback being that dedicated gates for different logic functions and other varieties must be implemented to avoid redundant implementation of gates in the design.

A future work can be suggested that the MGDI library can be rebuilt including several other layout designs for different basic gates and other operators so that the library can be optimized to use minimum number of gates per design. Another future work can be the investigation of the performance of the hybrid adder in larger modules and systems.

6. References

6.1. Journal Article

[1] Samraj Daphni, Kasinadar Sundari Vijula Grace, "*Design and Analysis of 32-bit Parallel Prefix Adders for Low Power VLSI Applications,*" Advances in Science, Technology and Engineering Systems Journal, 2019, doi:4. 10.25046/aj040213.

[2] 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), 2016, pp. 570-574, doi: 10.1109/CICN.2016.117.

[3] Megha Talsania and Eugene John, "*A Comparative Analysis of Parallel Prefix Adders*", Dept of Electrical and Computer Engineering, University of Texas at San Antonio Tx 78249.

[4] Sudeshna Sarkar, Monika Jain, Arpita Saha, Amitha Rathi, "*Gate Diffusion Input: A Technique for Fast Digital Circuits (Implementation on 180 nm Technology)*", IOSR Journals of VLSI and Signal Processing, 2014, Vol 4, Issue 2, Ver. IV.

[5] S.-L. Siu, W.-S. Tam, H. Wong, C.-W. Kok, K. Kakusima, H. Iwai, "*Influence of multi-finger layout on the subthreshold behaviour of nanometre MOS transistors*," Microelectronics Reliability, 2012, doi:52. 10.1016/j.microrel.2011.09.011.

[6] N. Poornima, V s kanchana Bhaaskaran, "Area Efficient Hybrid Parallel Prefix Adders," Procedia Materials Science, 2015, doi:10. 371-380. 10.1016/j.mspro.2015.06.069.

6.2. Book

[7] *Behzad Razavi, "Design of Analog CMOS Integrated Circuits", Tata McGraw-Hill Edition, **(2002)**.*