International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-3, January 2020

A Hierarchical Design of 128 Bit Carry Lookahead Adder in 65 nm CMOS Technology

Kishore Prabhala, Prabhandhakam Sangameswara Raju

Abstract*: As One Giga Hertz microprocessors power mobiles, laptops, tablets and personal computers in last few years, there is a massive need to reduce the number cycles to do addition which plays a significant role in Arithmetic Logic Unit (ALU) or Digital Signal Processing (DSP). The complexity of carry propagation is the critical variable once the designs requires addition over 32 bits. A hierarchical design has been developed to find Carry out at 16-bit stage from Propagate and Generate techniques from a 4-bit stage of Carry Lookahead Adder (CLA), so called Carry Lookahead Logic (CLL). Four blocks of CLL have been used to create another CLL block at a 16-bit level and similarly at 64 bit level and 128 bit level. A 65 nm CMOS technology library from Microwind used to simulate from logic to circuit level for the hierarchical design of 128 bit CLA and compared with 90 nm technology

***Keywords*: 65 nm CMOS, Carry Lookahead Adder, Hierarchical, Microwind, and VLSI.**

I. INTRODUCTION

A Nano second (ns) cycle time is common in many mobiles, laptops, tablets and personal computers which are sold close to three billion in last four years. This translates to a Giga Hertz (GHz) Microprocessor, but over 10 GHz microprocessors are powering higher end devices in last two year but addition which plays a key role in Arithmetic Logic Unit (ALU) or Digital Signal Processing (DSP). An adder has to be designed, verified, simulated at circuit level and layout at 65 nm to 22 nm for 128 bit data processing to meet the growing demand. Even technology moving towards 14 nm to 10 nm with 256 bit data processing is inevitable in next five years in all computation devices.

Addition has been done in two different ways. First bit addition or Least Significant Bit (LSB has only two bits for inputs A and B from 4 to 128 bit. This is done by "Half Adder" (HA), next bit addition will be done by three bits, two bits from inputs A and B while the third bit comes as carry in from the carry out at LSB as shown in table 1.

Revised Manuscript Received on January 30, 2020.
***** Correspondence Author
Kishore Prabhala*, Research Scholar, EEE PhD, Rayalaseema University, Kurnool, Senior Member IEEE, Principal, PLNM Degree College, Opposite Acharya Nagarjuna University Mens Hostel, Nagarjuna Nagar – 522 510, Guntur Dist., AP, India.
Prof. Prabhandhakam Sangameswara Raju, EEE, SVU Engineering College, Sri Venkateswara University, Tirupati – 517 502, AP, India.

This has been called "Full Adder" and truth table has been shown in table 2. Carry is generated when both bit are high in HA from truth table in table 1 or in FA at table 2. But in FA, Carry is generated for any combination of all three input A, B, Carry in (Cin) or A and B or A and Cin or B Cin being High or "1" as shown in table 2 (Mano, 2002). The logic diagram has been shown in figure with 2. Any design has to go through Kernaugh Map or K-Map, a standard process anyone can't skip, then get logical equations and finally define logic such CMOS - Complimentary Metal Oxide Silicon or ECL – Emitter Coupled Logic or TTL – Transistor Transistor Logic (Weste, 1993). The Logic process has been done by Register Transistor Logic (RTL) in Electronic Design Automation (EDA) Tools offered by Microwind or Mentor or Cadence or Synopsys (Uyemura, 2006).

Table 1: a) Truth table, b K-Map for Half-Adder

A	B	Sum	Co
0	0	0	0
0	0	1	0
0	1	1	0
0	1	0	1

K- Map For Sum

A \ B	0	1
0	0	1
1	0	0

For Carry Out

A \ B	0	1
0	0	0
1	0	1

The equation for Sum of Products for HA from K-Map

$$S = AB' + A'B \text{ or } A \oplus B \qquad \text{----- eq 1.1}$$

the symbol ⊕ stands for Exclusive OR (ExOR). A' is the complement of A. The equation for Carry Out of a HA from K-Map

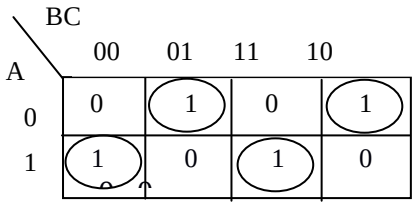
$$\text{Carry Out} = AB \qquad \text{----- eq 1.2}$$

Once the equations are obtained the logic logic design need to be done. Sum is generated by ExOR and Carry out generated by an AND.

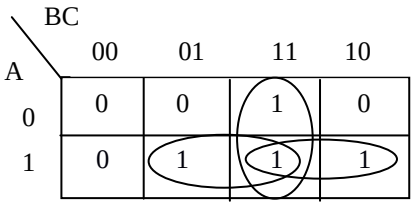
Table 2: a) Truth table & b) K-Map for Full Adder (FA)

A	B	Cin	Sum	Co
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

b) K- Map For Sum of Full Adder



K-Map For Carry Out of FA



The equation for Sum of Products from K-Map has been taken with individual terms for a value of "1"

S = AB'Cin' + A'B'Cin + A'BCin' + ABCin (take Cin' in 1st and 3rd term, 2nd and 4th terms) = (AB' + AB')Cin' + (A'B' + AB)Cin

= (A⊕B)Cin' + (A⊕B)Cin

= (A⊕B)Cin' + ((A⊕ B)')Cin

= A ⊕ B ⊕ Cin ----- eq 1.3

the symbol ⊕ stands for Exclusive OR, logically when inputs are not equal output will be ONE.

Equation for Carry out is

Co = AB + ACin + BCin or AB + (AB)Cin ----- eq 1.4

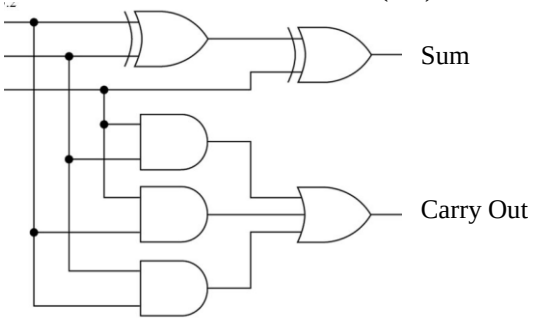


Figure 1: Logic Diagram for Sum and Carry out, Full Adder

A Very Large Scale Integrate (VLSI) design over thousands of Integrated Circuits in Metal Oxide Semiconductor (MOS) requires automation of logic with Electronic Design Automation (EDA) tools with Mentor or Cadence or Synopsys or Microwind. Generally, logic is defined as a behavioural model with respect to truth table as defined in table 2 for this case "Full Adder". Verilog is a open source logic accepted by any EDA tool vendor and after RTL, a logic diagram will be developed (Weste, 1993). A Verilog code for a Full adder is

```
module fulladder(a,b,c,sum,carry);
  input a,b,c;
  output sum,carry;
assign sum = a ^ b ^ c;
assign carry = (a & b) | (b & c) | (c & a);
endmodule
```

II. LOGIC DESIGN OF 4-BIT CLA

From 2nd bit onwards, there will be three bits needed to added and carry ripple through each adder stage. A propagation delay could have 2n gates if this design technique has to be used as Sum is generated from two "ExOR gates", shown in figure 1. Carry Lookahead Adder (CLA) has been the fastest to process addition and the design technique has been based on the principle of calculating "Carry Out" from four bit with Propagate or Generate terms (Mano, 2002). These techniques reduces the propagation delay of carry with more logic as defined by equations 2.2 to 2.7 but the area or number logic gates will be increased. "Generate" (G) will calculate Carry out if both A and B are "High" and "Propagate" (P) will calculate Carry at each stage. The generation values and propagation values are computed based on the carry out if A or B high but only if Carry in "High" as seen from equation 2.1.

Sum = A ⊕ B ⊕ Cin, Cout = AB + Cin(A+B) eq. 2.1

Generate, G = AB ----- eq. 2.2

Propagate, P = A + B ----- eq. 2.3

For a 4 bit CLA, the Generate and Propagate terms are $G_0 = A_0B_0$ and $P_0 = A_0+B_0$

$G_1 = A_1B_1$ and $P_1 = A_1+B_1$

$G_2 = A_2B_2$, and $P_2 = A_2+B_2$

$G_3 = A_3B_3$, and $P_3 = A_3+B_3$

Now the carry out at each stage of 4 bit CLA is

$C[1] = G_0 + P_0Cin$ ----- eq. 2.4

$C[2] = G_1 + P_1 G_0 + P_1 P_0 Cin$ ----- eq. 2.5

$C[3] = G_2 + P_2G_1 + P_2 P_1 G_0 + P_2P_1P_0Cin$ ----eq. 2.6

$Cout = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 + P_3P_2P_1P_0Cin$ ----- eq. 2.7

Carry generation from each stage has been calculated from the above equations and finally Carry out at the end of 4 bits will be developed as Carry Lookahead Logic (CLL) shown in the figure 2 from equation 2.7 (Mano, 2002)(Zhuang, 1992).

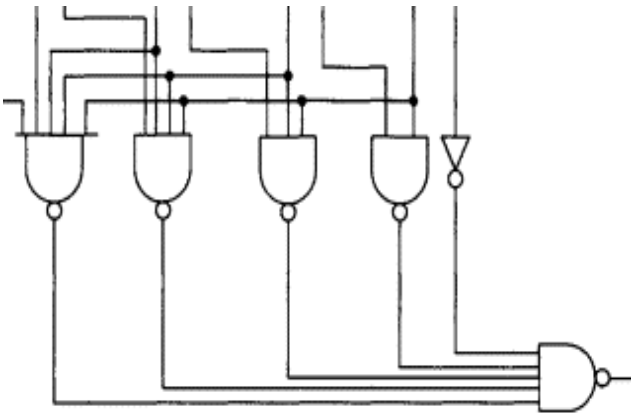


Figure 2: Carry Lookahead Logic (CLL) for Carry Out for a 4 bit adder for CLA

P_0, G_0 ry P_1, G_1 ead AP_2, G_2 CLA) P_3, G_3 o parts. One is addition and other is Carry Out logic. Full adder has been defined with four NAND gates as EXOR in two stages with Cin rate (G) coming from first NAND (it is G' or G Bar) and Propagate (P) coming from first EXOR output, A and B as inputs. This is shown in figure 3. The input carry-in will feed the second EXOR gate designed after four NANDs in the first stage shown in the bottom of the figure 3.

Cout

At bit 1, the carry out is generated from G0' from A0 and B0 stage (First NAND output), then Cin and P0 has taken from the fifth NAND and a NAND on top this two ExOrs have been shown with connection from G0' and (CinP0)'. The output will be fed as Cin at 1st level.

Co1 = ((G0')(Cin.P0)')' = ((G0)')' + ((Cin.P0)')' = G0 + Cin.P0

Similarly at bit 2 and bit 3 same design and drawing has been used to get carry generation. Right side of figure 3, CLA has been defined with equations for Generate and Propagate terms for any four bits.

Generate $G_{4bits} = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0$ ----- eq 2.9

Propagate $P_{4bits} = P_3P_2P_1P_0Cin$ ----- eq 3.0

With four types of NAND gates (2 input, 3 input, 4 input and 5 input) used in the first stage and all these four outputs from NANDs fed into 5-input NAND to generate final carry out. The CLL has been designed from figure 2 with equations 2.1 to 2.7 as Inverters, 2-input NAND, 3-input NAND, 4-input NAND for Generate term, a 5-input NAND for Propagate term and finally a 5-input NAND for carry out, feeds next stage (Weste, 1993) (Mano, 2002).

III. BEHAVIOUR MODEL AND LOGIC DESING FOR EXISTING 4 BIT CARRY LOOKAHEAD ADDER

A 4-bit CLA has been design through Verilog using two 4-bit input a and b and a sum output "s" with 4 outputs. At each adder stage a generate and propagate term have been defined from equations 2.3 to 2.7.

Verilog Code for a 4 bit Carry Lookahead Adder in Mentor is

```
module CLA (a,b,ci,co,s);
 input [3:0] a,b;
 output [4:0] s;
 input ci;
 output co;
 wire [3:0]G,P,C;
 assign G = a&b;
 assign P = a^b;
 assign co=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]&ci);
 assign C[3]=G[2]+(P[2]&G[1])+ (P[2]&P[1]&
G[0]) + (P[2]&P[1]&P[0]&ci);
 assign C[2]=G[1]+(P[1]&G[0]) + (P[1]&P[0]&ci);
 assign C[1]=G[0] + (P[0]&ci);
 assign C[0]=ci;
 assign s = {co,P^C};
endmodule
```

Four NAND gates using CMOS have been used for an ExOR, so there are eight NANDs have been used to calculate the output "Sum" as shown in figure. 3 while carry out has been designed with G term from 1st NAND and P with Cin from 5th NAND. So there are a total of 5 NAND gates used to propagate delay internal while each has only 2 inputs.

The stack of Four for input A0 to A3 and B0 to B3 with NANDs for ExOR for Sum and Carry out has been shown in figure 3 from bottom to top.

On the right side, there are three NAND gates for each of P and G term in overll Generate term and another NAND for Propagate term as part of Carry Lookahead Logic.

Three are a total of thirty two 2-input NANDs for Sum, three two 2-input NANDs for Carry Out, four Inverters, a 2-input NAND, 3-input NAND, two four Input NAND used for Carry Lookahead Logic.

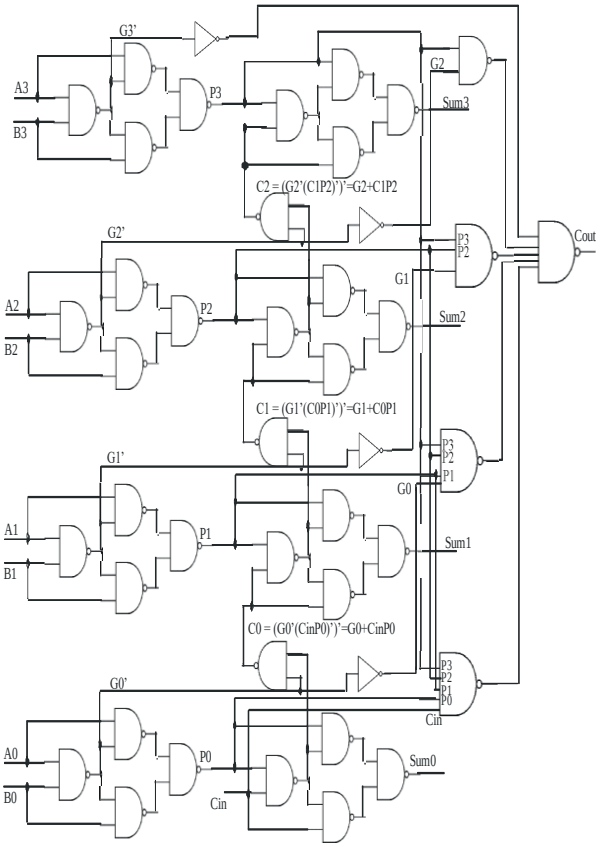


Figure 3: Logic Design of CLA with CMOS NAND gates, 4 bit adder

IV. PROPOSED HIERARCHICAL LOGIC DESIGN FOR 128 BIT CARRY LOOKAHEAD ADDER

If A 4-bit CLA is basic building block used in Hierarchical CLA for 32 or 64 or 128 bit as Abdulmajeed (2015) designed a 64 bits using VHDL and FPGA Virtex 7 family. None have explored 128 bit CLA hierarchically in Mentor or Microwind. rhrough the literature review gathered extensively.. There will more gates require to build Generate and Propagate carry at 16 bit stage then 64 bit stage, and finally at 128 bit stage.

A 4-bit CLA as shown in figure 3 will be split into basic 4-bit CLA and a 4-bit Carry Lookahead Logic (CLL) as shown in the right side of the figure 3. There are a total of five logic gates in CLL block, an inverter for G3, a 2-input NAND for P3G2, a 3-input NAND for P3P2G1, and a 4-input NAND for P3P2P1G0 for Generate term and then a 4-input NAND for P3P2P1P0 for a Propagate term, also shown in figure 2.

a) Design of 16-bit Hierarchical CLA

A 16-bit CLA has been designed from four 4-bit CLA with four Carry Lookahead Logic (CLL) blocks with four Generate and Propagate terms. For a 64 bit CLA, there are four 16 bit CLA blocks and two CLL blocks and for 128-bit CLA two 64 bit CLA and a CLL block have been used (Weste, 1993) (Mano, 2002) (Uyemura, 2006) (Abdulmajeed, 2015).

CLL block has four Generate terms and a Propagate term as inputs besides Carry-in. It has three outputs terms, Generate, Propagate, and Carry-Out.

A 16-bit CLA with CLL has been designed with four 4-bit CLA and along a CLL.

It has sixteen bits inputs from $A_{15\ to\ 0}$, $B_{15\ to\ 0}$ and Cin along with sixteen outputs for Sum, $\mathbf{Sum_{15\ to\ 0}}$.and a Generate term and a Propagate term as shown in figure 4.

In a Hierarchical CLA Desing, 'Propagate' term has been redefined from equation 2.9 as

$$P_{0\_3} = P_3P_2P_1P_0 \qquad \text{- eq 4.1}$$

The 'Generate term" from equation 3.0 has been redesigned as

$$G_{0\_3} = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 \qquad \text{- eq 4.2}$$

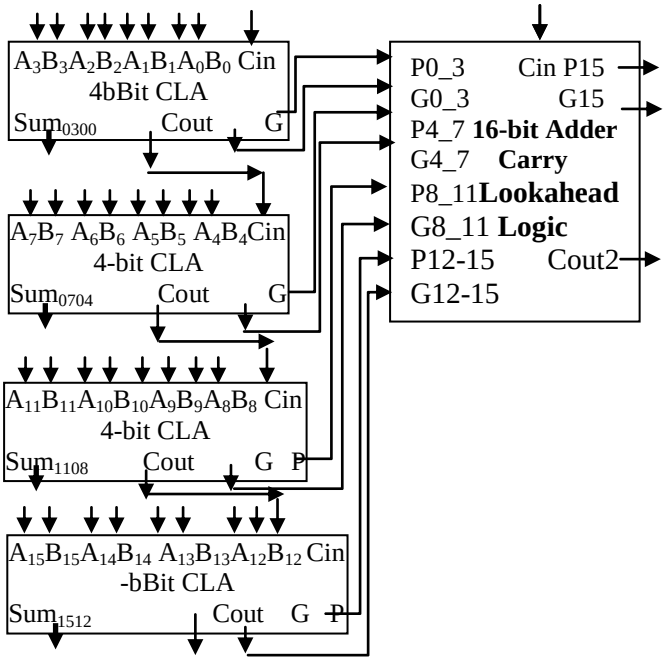


Figure 4: Hierarchical CLA for 16 bits from FOUR 4-bit CLA

A block diagram for a 16 bit CLA with CLL has been shown in figure 5 which will used to design for 64 bit CLA with CLL.

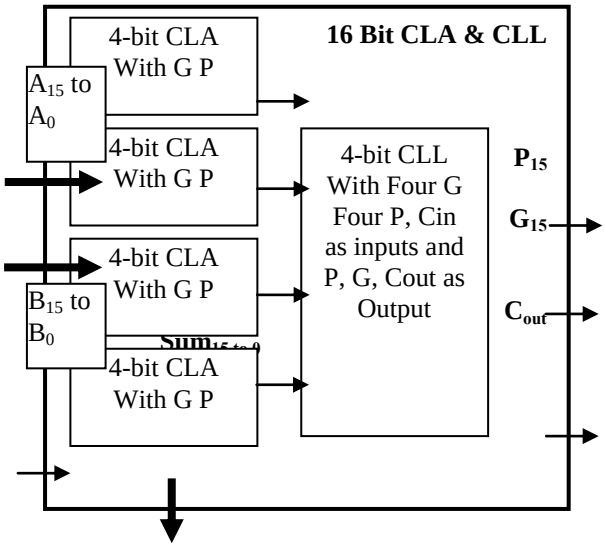


Figure 5: Block Diagram for Hierarchical 16-bit CLA

Generate Sum terms and then propagate with generate terms with four 4-bit CLA logic as shown in figure 5 from figure 4.

At first stage:

Propagate $P_{1stage} = P_3P_2P_1P_0$

Generate $G_{1stage} = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0$

At second stage:

Propagate $P_{2stage} = P_7P_6P_5P_4$

Generate $G_{2stage} = G_7 + P_7G_6 + P_7P_6G_5 + P_7P_6P_5G_4$

At third stage:

Propagate $P_{3stage} = P_{11}P_{10}P_9P_8$

GenerateG_{3stage} $=G_{11}+P_{11}G_{10}+P_{11}P_{10}G_9+ P_{11}P_{10}P_9G_8$

At fourth stage:

Propagate $P_{4stage} = P_{15}P_{14}P_{13}P_{12}$

Generate $G_{4stage} = G_{15}+P_{15}G_{14} +P_{15}P_{14}G_{13}+ P_{15}P_{14}P_{13}G_{12}$

Now the Generate as output term at 16-bit stage

Generate (1^{st} 16 bit CLL) $= G_{12\_15} + G_{8\_11}* P_{12\_15} + G_{4\_7}*P_{12\_15}*P_{8\_11} + G_{0\_3}*P_{12\_15}*P_{8\_11}*P_{4\_7}$

Propagate = $P_{12\_15}P_{8\_113stage}P_{4\_7}P_{0\_3}$

b) Design of 64-bit Hierarchical CLA

A Hierarchical design of 64-bit CLA takes "four 16-bit CLA with CLL" as shown in figure 5, designed from 4-bit CLAs with inputs Cin, A15 to A0 and B15 to B0 driving first CLA and A31 to A16 and B31 to B16 driving second CLA, with inputs A47 to A32 and B47 to B32 driving third CLA, and with inputs A63 to A48 and B63 to B48 driving the last and fourth CLA as shown in figure 6. Each 16-bit CLA has a CLL block creating three outputs, a Generate and a Propagate terms besides Carry out. A Carry Out and 64 outputs for $\mathbf{Sum_{63\ to\ 0}}$ have been created,.

In all eight terms namely P15, G15, P31, G31, P47, G47, P63, G63 have been created with equations 4.1 and 4.2.

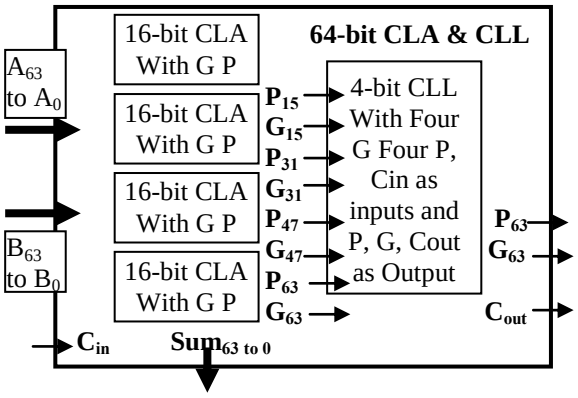


Figure 6: Block Diagram for Hierarchical 64-bit CLA

c) Design of 128-bit CLA with two 64-bit Hierarchical CLAs

Finally two 64-bit CLAs with CLL as shown in figure 6 have been used to create a hierarchical 128-bit CMOS CA. First block of 64-bit CLA/CLE have Propagate and Generate Terms as P15, G15, P31, G31, P47, G47, P63, G63 and second block 64-bit CLA/CLE have Propagate and Generate Terms as P79, G79, P95, G95. P111, G111, P127, G127.

Two sets of 64 outputs for Sum have been created, $\mathbf{Sum_{63\ to\ 0}}$ and $\mathbf{Sum_{127\ to\ 64}}$

Each 64-bit block generates a Propagate term four Propagate terms and a Generated Term from four Generate terms.

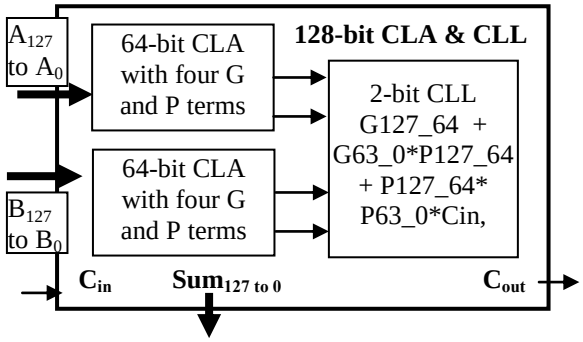


Figure 7: Block Diagram for Hierarchical 128-bit CLA

Finally there are two Propagate and Generate terms and would generate Carry out as

Carry Out = G127_64 + G63_0 * P127_64 + P127_64 * P63_0 * Cin, this is final Carry Out of 128 bit Hierarchical CLA designed in CMOS.

The final Carry out has been implemented with a 2 input NAND.

V. RESULTS AND DISCUSSION

The delay for Carry out can be in a logic form as One NAND (2 input) for P or G at 1-bit stage, two NANDs (4 input) at 4-bit stage. Sum is generated from six NAND gates (2 inputs) but carry in comes from previous stage using carry out equation G+PCin.

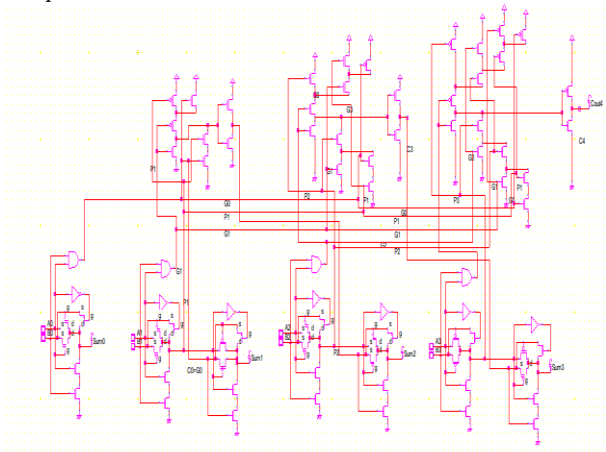


Figure 8: 4-bit CLA with NAND gates used in Hierarchical Design for 128 bits

At 4-bit stage the terms Propagate P = P3P2P1P0 or Generate G = G3 + P3G2 + P3P2G1 + P3P2P1G0 are designed with two NANDs (4 input) and similarly at 64-bit stage and two NANDs (3 input) at final stage with overall delay of 24 nMOS transistors, for a 2 input NAND has two nMOS transistors in series. A four bit CLA with NAND gates has been implemented in CMOS along with CLA logic as shown in figure 8. L and W of 90 nm and 65 nm have been used for nMOS transistors and for pMOS transistor width of 2.5 times used. Simulation has been done at room temperature, 26^{o}C.

A four input NAND has a delay of 1.0 ns at 90 nm and 0.7ns at 65nm technologies while 1.8 ns have taken to generate Sum at 90 nm and 1.5 ns at 65 nm technology as shown in figure 8.

Table 3: Analysis of Sum at 4 bit vs 4 input NAND (CMOS)

CMOS Technology	Sum Delay for 4 bit CLA	4 input NAND
90 nm	1.8 ns	1.0ns
65nm	1.5 ns	0.7ns

In this hierarchical design a 4 input NAND generates G and P terms at 4-bit for carry out, another at 16 bit, another 64 bit and only 2 input NAND at 128 bit as shown from figure 6. So top part addition from A127 to A63 dictated by the carry generation of three 4-input NANDs and a 2 input NAND as carrying drives the final summation stage.

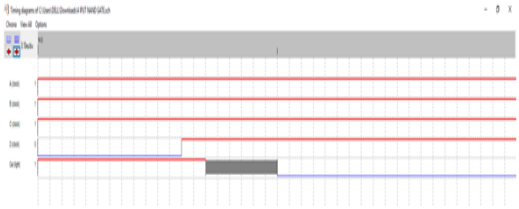


Figure 9: Propagation delay of simulation for 4-bit CLA with NAND gates used in Hierarchical CLA Design for 128 bits

A 128-bit Hierarchical CLA designed used to simulate with 90nm and 65 nm CMOS technology has been shown in figure 10 using Microwind EDA tools.

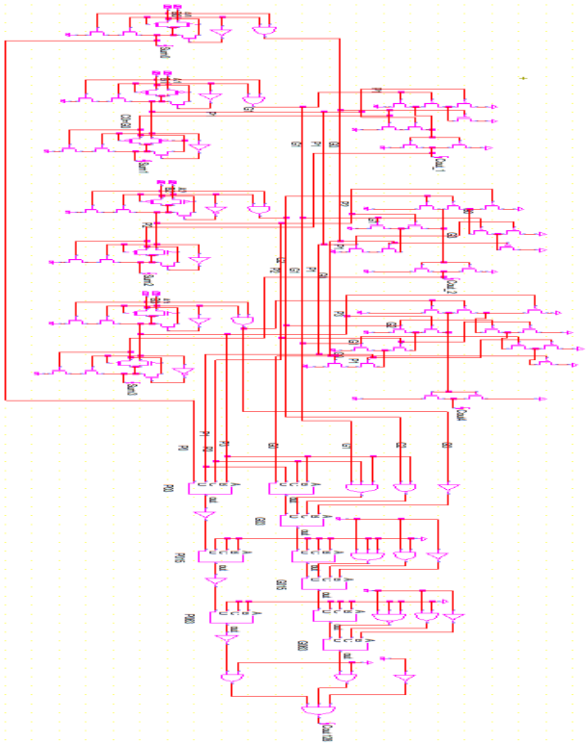


Figure 10: 128-bit Hierarchical CLA

Only inputs A3 to A0 and B3 to B0 have been changed in the simulation to find propagation delay because the "Carry out "calculation has to begin from the Least Significant Bits (LSB) for worst case delay. Figure 9 shows the beginning of x-axis scale of 3.4 ns with each sub-division with 0.1 ns for final carry called "Carry 128" for full cycle of inputs.

Inputs A0 to A127 have been kept at "High" and B0 has been toggled from "Low" to "High" while all other inputs of B have been kept at "Low" to have a carry propagated through all 128 bits. The final carry has a delay of 4.6 ns at 90 nm and 4.0 ns at 65nm technologies in Microwind simulation at using 90 nm and 65 nm libraries as shown in figure 11.

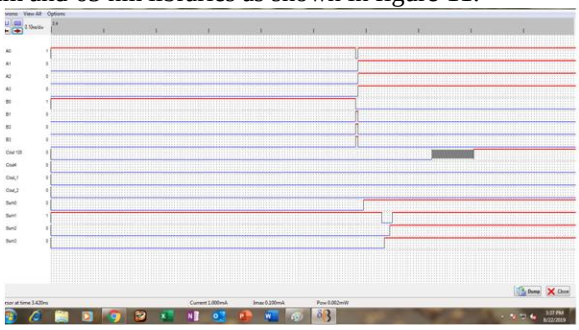


Figure 11: Propagation delay of simulation for Hierarchical CLA Design for 128 bits

Previous work of Abdulmajeed (2015) shows a delay 5.62 ns for a 64 bit hierarchical CLA with FPGA Virtex7 family. Even at 128 bits, this result is faster by 20 percent. Mangarao (2014) did a 128-bit Carry Select Adder with a delay of 36.52 ns at 90nm CMOS technology [10]. Balasubramanian (2016) did only 32 bit adder design with delay of 3.53 ns [2].

Bharathi A, et (2014) did a 128 bit CLA with a Domino Logic using HSPICE but through a graph could not read the propagation value for sum [3].

Table 4: Analysis of Propagation Delay of Sum at 128-bit Hierarchical CLA at 26^{0}C

CMOS Technology	Delay
90 nm	4.6 ns
65nm	4.0 ns
Previous work of Abdulmajeed (2015) at 130 nm	5.62 ns

VI. CONCLUSION

The analysis from the simulation of Microwind for a Hierarchical CLA design at 128 bits shows carry out of 4.6 ns at 90 nm and 4.0 ns at 65nm CMOS technology. The difference is 13 percent. Throughout my literature review there were only less than half dozen papers identified having research done on 128 bit addition with various architectures of adder. Only [1] Abdulmajeed (2015) did similar work with 64 bits having a delay of 5.62 ns. 4 ns is critical point of technology as microprocessors already have gone ahead with over 10 GHz which translates a cycle time of 100 ps. At this speed even a 128 bit adder will take 40 to 46 cycles to complete an addition. This need be improved as 50 to 100 GHz microprocessors have been experimented in the latest designs of smart phones as of 2019. This is need for the shrinking technology node of Length of a transistor to 45 nm and 22 nm.

REFERENCES

1. Abdulmajeed Alghamdi and Fayez Gebali (2015), "Performance Analysis of 64-bit Carry Lookahead Adders Using Conventional and Hierarchical Structure Styles", Department of Electrical and Computer Engineering, University of Victoria, Canada, https://www.academia.edu/14338863/ Performance_Analysis_of_64-bit_Carry_Lookahead_Adders_Using_ Conventional_and_ Hierarchical _Structure_Styles
2. Balasubramanian P and S. Yamashit, 2016, "Area/latency optimized early output asynchronous full adders and relative-timed ripple carry adders", SpringerPlus, 5:440, Pages 1-26.
3. Bharathi A, K. Manikandan, K. Rajasri, and Santhini, (2014), "High Speed Multioutput 128bit CarryLookahead Adders Using Domino Logic", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering Vol. 3 / 10
4. Bui, H.T., Y.Wang, and Y.Jiang (2002), "Design and Analysis of Low-Power 10-transistor Full Adders using Novel XOR-XNOR gates", IEEE Transactions on Circuits and Systems – II, Analaog and Digital Signal Processing, Vol. 49, No.1.
5. Kapil Mangla, and Shashank Saxena (2015), "Analysis of Different CMOS Full Adder Circuits Based on Various Parameters for Low Voltage VLSI Design", International Journal of Engineering and Technical Research (IJETR), Vol. 3 /
6. Kishore Prabhala and Prof. Prabhandhakam Sangameswara Raju (2019), "A Comparison of 128 bit Additon Using Ripple Carry Adder with Carry Lookahead Adder in Mentor EDA Tools", IJARTET - Volume 6 Issue 9, Sept. 2019. ISSN: 2394-3777, ONLINE: 2394-3785
7. Kishore Prabhala and Prof. Prabhandhakam Sangameswara Raju, "A CMOS Design of 64 bit ALU using Mentor Tools", IJSART - Volume 5 Issue 4 –April 2019, ISSN [ONLINE]: 2395-1052
8. Kishore Prabhala, Haritha Dasari, and Thrinadh Komatipalli (2018), "Performance Comparison of 64-Bit Adders", IJEDR, Volume 6, Issue 2, June 2018, ISSN: 2321 9939
9. Kishore Prabhala (2016), A Study of CMOS VLSI Technology Landscape Challenges towards 7 nm, International Conference in Electronic and Electrical Engineering, ICAECS 2016. 10th Dec., 2016, Vadlamudi, Guntur Dist., AP.
10. P. Manga Rao and K. Ashok Kumar (2014), "An optimized implementation of 128 bit Carry Select Adder using binary to excess-one converter for delay reduction and area efficiency", International Journal of Professional Engineering Studies, Vol. IV, Issue 4, December.
11. Morris Mano "Digital Design". Pearson Education Asia. 3rd Ed, 2002.
12. Sushma Kodagali, Sneha Solanki and Debajani Mahanta (2017), "Comparative Analysis of Various Fast Adder Circuits in Different Deep Sub-micron Technologies", International Journal of Electronics, Electrical and Computational System, IJEECS, Volume 6 / 8, ISSN 2348-117X.
13. Uyemura, John P, "Chip Design for Submicron VLSI: CMOS Layout and Simulation", Cengage Learning, 2006.
14. Weste, N. and K. Eshranghian, "Principles of CMOS VLSI Design: A System Perspective," Reading MA:Wesley, 1993.
15. Zhang, M.J. Gu, and C. H. Chang, (2003) "A novel hybrid pass logic with static CMOS output drive full-adder cell," in Proc. IEEE Int. Symp. Circuits Syst.,pp. 317–320.
16. Zhuang N. and H. Wu (1992), "A new design of the CMOS full adder," IEEE J. Solid State Circuits, vol. 27 / 5, pp. 840– 844.
17. Zimmermann R. and W. Fichtner (1997), "Low-Power Logic Styles: CMOS Vs Pass- Transistor Logic" IEEE Journal of Solid-State Circuits, vol. 32 / 7, pp. 1079–1090

AUTHORS PROFILE

Kishore Prabhala profile Kishore Prabhala is a research Scholar in EEE PhD, Rayalaseema University, Kurnool, AP and also Senior Member IEEE. He published six papers in CMOS VSLI design in India and guided over 8 projects for M.Tech., students in VLSI. He left USA in 1994 after working at Motorola, MMI and National Semiconductor from 1981. Currently, director of VLSI Design Centre, PSK Research Foundation, Opp ANU Mens Hostle, Nagarjuna Nagar – 522 510, and also Principal, PLNM Degree College, Opposite Acharya Nagarjuna University Mens Hostel, Nagarjuna Nagar – 522 510, Guntur Dist., AP, India. He received a MSEE from Georgia Institute of Technology, GA, USA in 1989 and BSEE from Purdue University, W.Lafatette, IN, USA in 1981, Cell: 9177408565

Prof. Prabhandhakam Sangameswara Raju is a Professor in Dept. of Electrical and Electronics Engineering, SVU Engineering College, Sri Venkateswara Univeristy, Tirupati – 517 502, Chitoor Dist., AP. He received M.Tech. and Ph.D. from SVU Engineering College. He has been teaching PG course for last 25 years and guided over 60 projects. He published over 80 papers. Currently there are 8 students pursuing Ph.D.