Fastest Carry Look-Ahead Adder Design for Enhanced Computational Efficiency

Vedant Pahariya
ECD IIIT-H
2023112012
vedant.pahariya@research.iiit.ac.in

***Abstract*—This paper proposes the design for the fastest CLA. It mentions about all the blocks and topology used in the design. It also mentions about the simulation results and the comparison with existing designs. The design is implemented in 180nm technology. The design is compared with the existing designs and it is found that the proposed design is the fastest among all.**

***Index Terms*—Carry Look-Ahead Adder, CLA, VLSI, Computational Efficiency, 180nm Technology, Digital Design, High-Speed Arithmetic**

I. Introduction

The Carry Look-Ahead Adder (CLA) is a digital circuit that is used to add two binary numbers. It is a basic unit component for all arithmetic processes which goes in processor. Making it faster can enhance the computational efficiency of whole system. The CLA is a fast adder because it can generate the carry signals for all the bits in parallel. The CLA is faster than the Ripple Carry Adder (RCA) because the RCA generates the carry signals sequentially, faster than the Carry Select Adder (CSA) because the CSA generates the carry signals for a group of bits in parallel, faster than the Carry Skip Adder (CSKA) because the CSKA generates the carry signals for a group of bits in parallel, faster than the Carry Increment Adder (CIA) because the CIA generates the carry signals for a group of bits in parallel.

II. Carry Look-Ahead Adder Design

CLA design consists of three blocks: the Propagate & Generate block, Carry Look Ahead (CLA) block and Sum block. The Propagate & Generate block gives carry generate (G_i) and propagate generate (P_i) signal for every i^{th} bit of inputs. The output of this block is used in CLA block to generate Carry C_i which is then utilised in the Sum block to get the Sum output. The basic CLA design is based on the following equations:

$$G_i = A_i \cdot B_i \tag{1}$$

$$P_i = A_i \oplus B_i \tag{2}$$

$$C_i = G_i + P_i \cdot C_{i-1} \tag{3}$$

where G_i is the Generate signal, P_i is the Propagate signal, C_i is the Carry signal, A_i is the i^{th} bit of the first number, B_i is the i^{th} bit of the second number, and C_{i-1} is the $(i-1)^{th}$ Carry signal.

Here, we are focusing on the 4-bit CLA design. Therefore, using recursive approach for carry in above equations, we can write the equations for the 4-bit CLA design as follows:

$$C_0 = G_0 + P_0 \cdot C_{in} \tag{4}$$

$$\begin{aligned} C_1 &= G_1 + P_1 \cdot C_0 \\ &= G_1 + P_1 \cdot (G_0 + P_0 \cdot C_{in}) \\ &= G_1 + P_1 \cdot G_0 + P_1 \cdot P_0 \cdot C_{in} \end{aligned} \tag{5}$$

$$\begin{aligned} C_2 &= G_2 + P_2 \cdot C_1 \\ &= G_2 + P_2 \cdot (G_1 + P_1 \cdot (G_0 + P_0 \cdot C_{in})) \\ &= G_2 + P_2 \cdot G_1 + P_2 \cdot P_1 \cdot G_0 + P_2 \cdot P_1 \cdot P_0 \cdot C_{in} \end{aligned} \tag{6}$$

$$\begin{aligned} C_3 &= G_3 + P_3 \cdot C_2 \\ &= G_3 + P_3 \cdot (G_2 + P_2 \cdot (G_1 + P_1 \cdot (G_0 + P_0 \cdot 0))) \\ &= G_3 + P_3 \cdot G_2 + P_3 \cdot P_2 \cdot G_1 + P_3 \cdot P_2 \cdot P_1 \cdot G_0 \end{aligned} \tag{7}$$

where C_i, G_i, and P_i are the Carry, Generate, and Propagate signals for the i^{th} bit, respectively. C_{in} is the input Carry signal.

A. Improving the Basic CLA Design

The above basic CLA design can be improved extensively by making the following changes:

1) Use of OR gate instead of XOR gate for Propagate block: XOR gate can be replaced by OR gate which can be further reduced to the NOR logic for the Propagate block to reduce the number of gates in the design. The Propagate block can be designed as follows:

$$P_i = A_i + B_i \tag{8}$$

To demonstrate that both XOR and OR gates give the same results for the Propagate block, we can see in the following truth table that results are same for both XOR and OR gates except when both inputs are 1.

TABLE I
Truth Table for XOR and OR Gates

A_i	B_i	$A_i \oplus B_i$ (XOR)	$A_i + B_i$ (OR)
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	1

In case of carry generation, when both inputs are High (1), that is both $A_i = 1$ and $B_i = 1$:

$$\begin{aligned} G_i &= A_i \cdot B_i = 1 \cdot 1 = 1 \\ C_i &= G_i + P_i \cdot C_{i-1} \\ &= 1 + P_i \cdot C_{i-1} \\ &= 1 \text{ (regardless of } P_i \text{ and } C_{i-1}) \end{aligned}$$

Therefore, when both inputs are 1, the carry signal is always generated ($C_i = 1$) regardless of the value of the propagate signal (P_i) or the previous carry (C_{i-1}).
As shown above, for the purpose of carry propagation in the CLA design, the OR gate can be used in place of XOR to reduce the number of gates.

2) Modification of the equations for the Carry signals to reduce the number of gates: The equations for the Carry signals for basic CLA use AND and OR gates which inherently includes two extra transistors for each gate becuase of the presence of the inverter. The number of gates can be reduced by modifying the logic such that it uses NAND and NOR gates in place of AND and OR wherever possible. Starting with simplification of C_1, we know that:

$$C_1 = G_0 + P_0 \cdot C_0$$

Here, we can convert AND gate between P_0, C_0 to NOR gate as shown below:

$$C_1 = G_0 + \overline{\left(P_0' + \overline{C_0}\right)}$$

Further, we can convert OR gate to NAND gate as shown below:

$$C_1 = \overline{G_0' \left(P_0' + \overline{C_0}\right)}$$

Similarly, for the other carry signals, the modified equations can be written as follows:

$$C_1 = \overline{G_0' \left(P_0' + \overline{C_0}\right)} \tag{9}$$

$$C_2 = \overline{G_1' \left(P_1' + G_0'\right)} + \left(\overline{P_1' + P_0'}\right) C_0 \tag{10}$$

$$C_3 = \overline{G_2' \left(P_2' + G_1'\right)} + \left(\overline{P_2' + P_1'}\right) \overline{G_0' \left(P_0' + \overline{C_0}\right)} \tag{11}$$

$$G'_{\text{out}} = \overline{G_3' \left(P_3' + G_2'\right)} + \left(\overline{P_3' + P_2'}\right) \overline{G_1' \left(P_1' + G_0'\right)} \tag{12}$$

$$P'_{\text{out}} = \overline{\left(P_3' + P_2'\right)\left(P_1' + P_0'\right)} \tag{13}$$

$$C_4 = \overline{G'_{\text{out}} \left(P'_{\text{out}} + \overline{C_0}\right)} \tag{14}$$

where G_i' and P_i' are the modified Generate and Propagate signals for the i^{th} bit, respectively.

B. Topology and Sizing of Each Block

Based on the equations and logic functions derived above, the topology of the 4-bit CLA design can be implemented as shown in the figure below. The design is implemented in 180nm technology. The sizing of the transistors in each block is done considering the speed and power consumption of the design. I have followed the CMOS logic for all the logic gates except XOR. For XOR, I used Pass Transistor Logic because it uses less no. of transistors. Here, I implemented the method of logical effort for sizing each block as following:

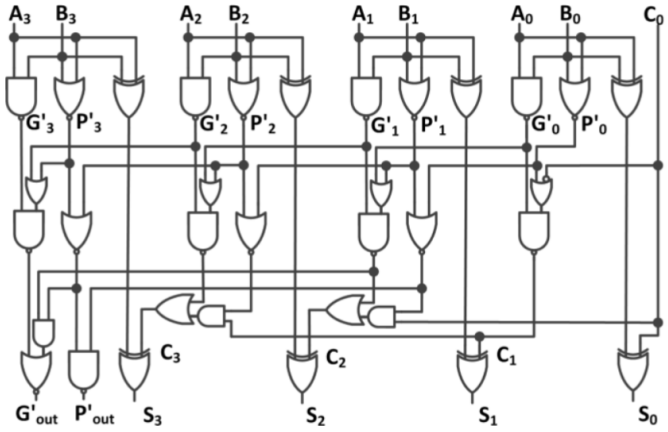

Fig. 1. 4-bit Carry Look-Ahead Adder Design

1) Propagate & Generate Block: This block includes three gates for each carry bit: One NOR gate for the Propagate signal, one NAND gates for the Generate signal and one XOR gate for the Sum signal.

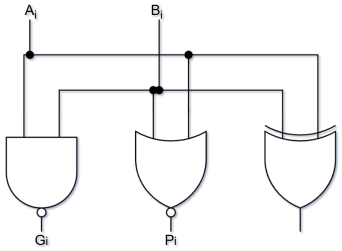

Fig. 2. Propagate & Generate Block

Accoroding to the method of logical effort, the size of these gates is larger i.e. just double the proceedings gates of the intermediate block of CLA. So, the table below shows the sizing of gates in this block:

TABLE II
SIZING OF GATES IN PROPAGATE & GENERATE BLOCK

Gate	Width (W_N/W_P) (μm)	Length (μm)	Number of Gates
NOR	3.6 / 7.2	0.18	4
NAND	3.6 / 3.6	0.18	4
XOR	0.36 / 0.72	0.18	4

2) Carry Look Ahead Block: This block again further divided into two more subblocks namely intermediate block and AndOr block. The intermediate and the AndOr block includes three gates and two gates respectively for each carry bit. The intermediate block includes a NOR gate, NAND and OR gate and the AndOr block includes one AND gate and one OR gate.

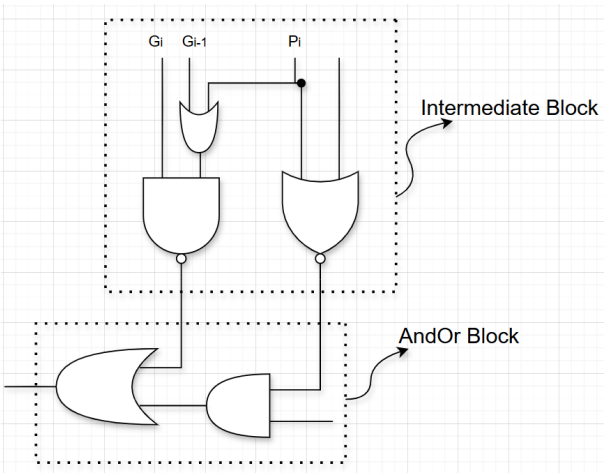

Fig. 3. Carry Look Ahead Block for Single Carry Bit

The sizing of gates in the intermediate and AndOr block is done as per the method of logical effort. As a result, also mentioned in the previous block that this block sizing is half of the propagate & generate block sizing. The OR gate is implemented using the NOR gate with an inverter at the output. Similarly for the AND gate, the NAND gate is used with an inverter at the output. The table below shows the sizing of gates in these blocks:

TABLE III
SIZING OF GATES IN INTERMEDIATE AND ANDOR BLOCKS

Block	Gate	Width (W_N/W_P) (μm)	Length (μm)
Intermediate	NOR	0.9 / 1.8	0.18
	NAND	1.8 / 1.8	0.18
	OR	0.9 / 1.8	0.18
AndOr	AND	1.8 / 1.8	0.18
	OR	0.9 / 1.8	0.18
Inverter	-	0.9 / 1.8	0.18

3) Sum Block: This block includes single XOR gate for the Sum signal which is implemented using Pass Transistor Logic. The sizing of this is set to be least possible just enough to drive the W-sized inverter and satisfy the DRC constraints because it is the last block in the design and don't have to drive any other block.

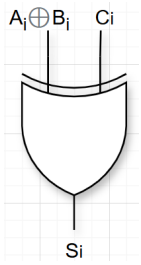

Fig. 4. Sum Block

TABLE IV
SIZING OF GATES IN SUM BLOCK

Gate	Width (W_N/W_P) (μm)	Length (μm)	Number of Gates
XOR	0.36 / 0.72	0.18	6

4) D-Flip Flop: It is implemented using the True Single Phase Clocked (TSPC) technology. It includes total no. of 12 transistors in CMOS logic style. The sizing of this is done by equating the resistances with the minimum W-sized inverter. There are two parts in the postive edge trigerred TSPC D-Flip Flop: Low-Level Latch and High-Level Latch.

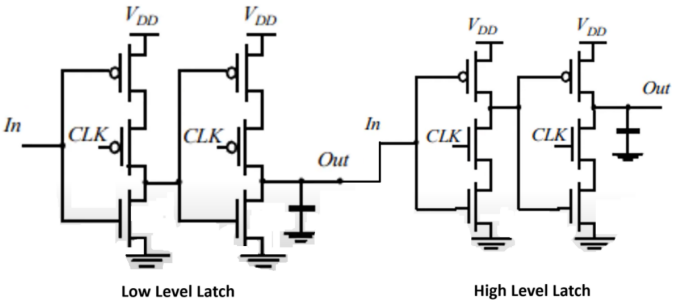

Fig. 5. TSPC D-Flip Flop

TABLE V
SIZING OF TRANSISTORS IN TSPC D-FLIP FLOP

Block	Transistor	Width (μm)	Length (μm)
Low-Level Latch	PMOS	3.6	0.18
	NMOS	0.9	0.18
High-Level Latch	PMOS	1.8	0.18
	NMOS	1.8	0.18

III. DELAY CHARACTERISTICS OF D-FLIP FLOP

The TSPC D Flip-Flop was simulated using NGSPICE with a clock period of 2ns and input period of 4ns. Both clock and input signals were given sharp rise/fall times of 50ps to analyze the flip-flop's delay characteristics.

A. Clock to Q Delay

Upon simulating, we observe the following waveforms and terminal outputs:

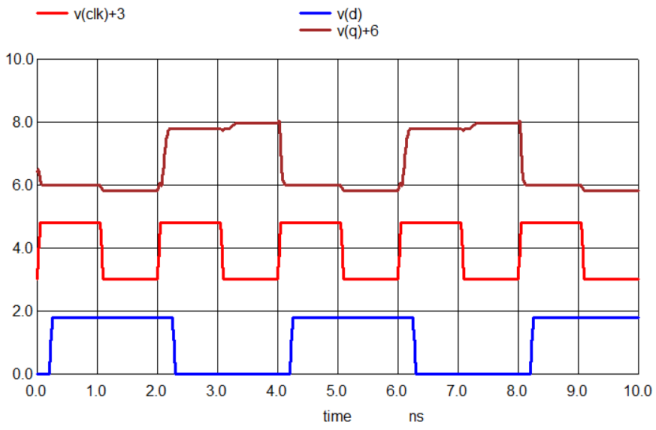

Fig. 6. Waveforms Output of D-Flip Flop

```
  Measurements for Transient Analysis

rise_delay          =  9.157922e-11 targ=  2.116579e-09 trig=  2.025000e-09
fall_delay          =  3.076888e-11 targ=  4.055769e-09 trig=  4.025000e-09
propagation_delay   =  6.11740e-11
```

Fig. 7. Terminal Output for Delay

The simulation gives the following timing parameters:

- Rise Delay: 91.58ps - Time taken for the output to rise after the triggering clock edge
- Fall Delay: 30.77ps - Time taken for the output to fall after the triggering clock edge
- Average Propagation Delay: 61.17ps - Mean of rise and fall delays

The asymmetry between rise and fall delays (ratio $\approx$ 3:1) can be attributed to:

- Different sizing of PMOS/NMOS transistors in the low-level and high-level latches
- Inherent mobility difference between PMOS and NMOS devices
- Cascaded structure of the latches affecting signal propagation paths

B. Setup Time

Setup time (t_{setup}) is the minimum time before the active clock edge during which the data input (D) must remain stable. For TSPC D flip-flop, setup time is equal to the propagation delay of the Low-Level Latch because when clock is low, the Low-Level Latch is active and as it rises to high, the correct stable value of D must be present at the output of the Low-Level Latch for ensuring correct operation. This stable value of D appears at the output of the Low-Level Latch after the propagation delay of the Low-Level Latch.

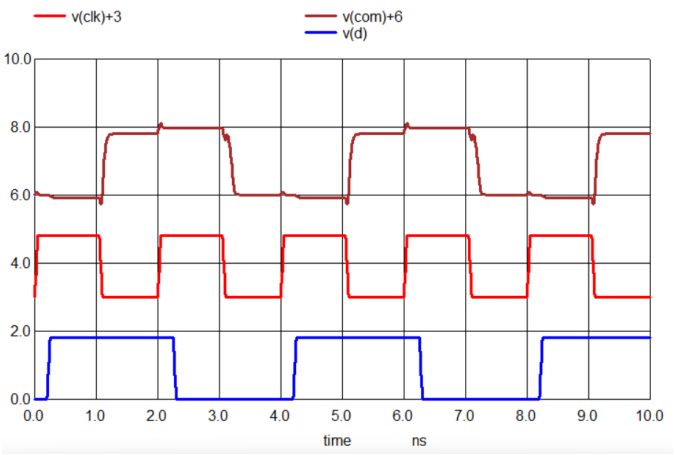

Fig. 8. Waveforms Output of Low-Level Latch

```
  Measurements for Transient Analysis

rise_delay          =  4.883258e-11 targ=  1.123833e-09 trig=  1.075000e-09
fall_delay          =  1.264689e-10 targ=  3.201469e-09 trig=  3.075000e-09
propagation_delay   =  8.76507e-11
```

Fig. 9. Terminal Output for Setup Time

C. Hold Time

Hold time (t_{hold}) is the minimum time after the active clock edge during which the data input must remain stable. For TSPC D flip-flop, hold time is zero because the High-Level Latch is active when the clock is high and the output of the High-Level Latch is not dependent on the input D when the clock is high. Therefore, the input D can change immediately after the active clock edge without affecting the output Q.

TABLE VI
TIMING CHARACTERISTICS OF TSPC D-FLIP FLOP

Parameter	Value (ps)
Rise Delay	91.58
Fall Delay	30.77
Propagation Delay	61.17
Setup Time	87.65
Hold Time	0

IV. STICK DIAGRAM OF UNIQUE GATES

A. Inverter

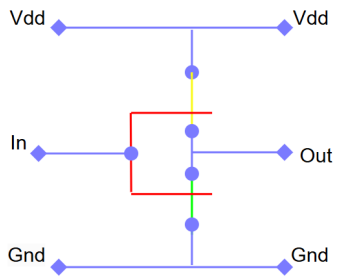

Fig. 10. Inverter Stick Diagram

B. NOR Gate

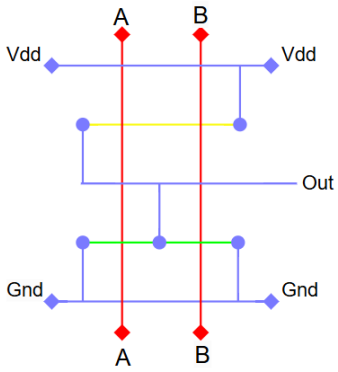

Fig. 11. NOR Gate Stick Diagram

C. NAND Gate

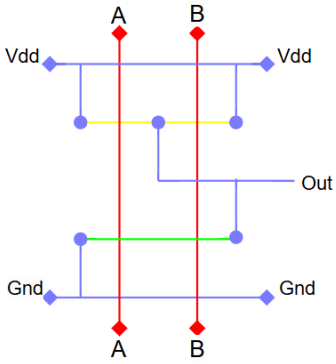

Fig. 12. NAND Gate Stick Diagram

V. Prelayout Simulation Results & Analysis of CLA

The 4-bit Carry Look-ahead Adder was simulated using NGSPICE with the following test conditions:

A. Verification of functionality

1) Input Specifications:

- Supply Voltage (VDD): 1.8V
- Input A (4-bit):
  - A0: Pulse starting at 10ns, Period = 40ns
  - A1: Pulse starting at 15ns, Period = 50ns
  - A2: Pulse starting at 20ns, Period = 60ns
  - A3: Pulse starting at 25ns, Period = 70ns
  - Rise/Fall times: 50ps
- Input B (4-bit):
  - B0: Pulse starting at 12ns, Period = 44ns
  - B1: Pulse starting at 18ns, Period = 56ns
  - B2: Pulse starting at 24ns, Period = 68ns
  - B3: Pulse starting at 30ns, Period = 80ns
  - Rise/Fall times: 50ps
- Carry Input (Cin): Pulse starting at 36ns, Period = 88ns
- Clock (CLK): Period = 8ns

2) Output Analysis: The simulation produces the following outputs:

- Sum outputs (S0-S3): Generated through XOR combinations of inputs and internal carries
- Final Carry (C4): Represents the overflow from the 4-bit addition

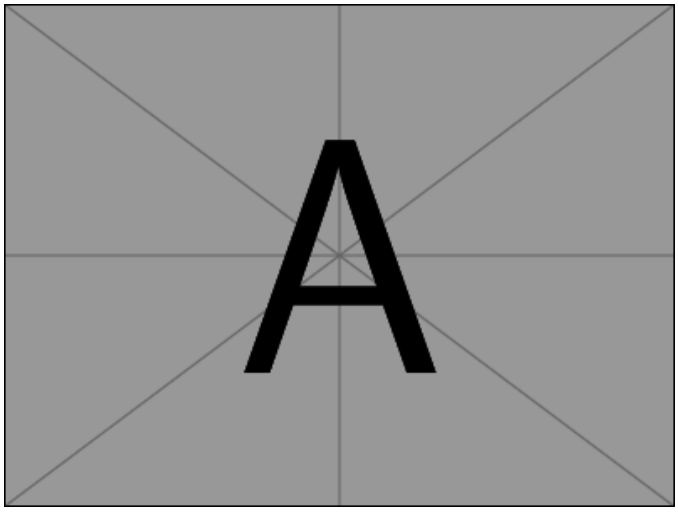

Fig. 13. Simulation Waveforms of 4-bit CLA

The waveforms demonstrate correct operation of the CLA with proper carry propagation and sum generation across all bit positions. The progressive delays in input signals help verify the adder's functionality under varying timing conditions.

B. Worst Case Delay of CLA Adder Block

Worst case delay is the maximum time taken for the output to stabilize after the input changes. This happens in the path which has highest resistance or very large number of gates. In the 4-bit CLA design proposed above, the worst case delay is observed in the carry propagation path from B_0 through C3 to S3 because it is the longest path with total 9 gates in midway. For considering the worst case delay, the S3 bit must change when B0 bit changes. So, here we take the input signals as follows:

- $A_0 = A_1 = A_2 = A_3 = 0, B_1 = B_2 = 1, B_3 = 0$
- B_0: Pulse starting at 25ns, Period = 70ns
- C_{in} = 1
- Clock (CLK): Pulse starting at 0ns, Period = 8ns

With the above inputs, we get the simulation results for the worst case delay as follows:

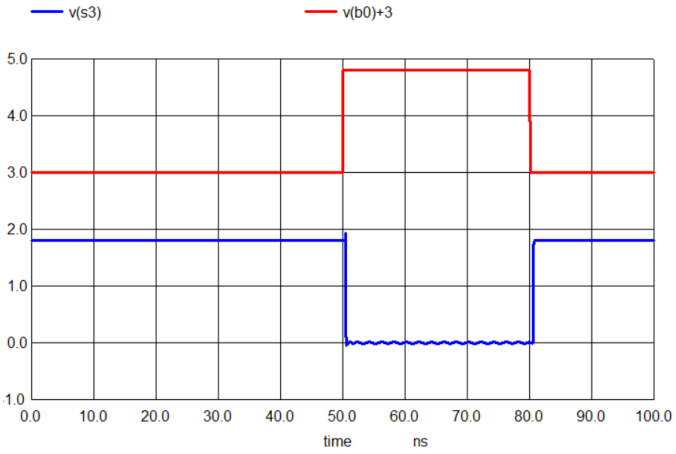

Fig. 14. WaveForms for the Worst Case Delay

```
  Measurements for Transient Analysis

rise_delay          =  4.621690e-10 targ=  5.048717e-08 trig=  5.002500e-08
fall_delay          =  4.961677e-10 targ=  8.057117e-08 trig=  8.007500e-08
tpd3                =  4.79168e-10
```

Fig. 15. Worst Case Delay Terminal Output

The simulation results show that the worst case delay for the 4-bit CLA design is 479.16ps. This delay is observed in the carry propagation path from B_0 through C3 to S3.

C. Maximum Clock Speed

1) Theoritical Calculation: The maximum clock speed of the 4-bit CLA design can be calculated using the following inequality:

$$t_{CQ_1} + t_{pd} + t_{su} \leq T_{clk} \tag{15}$$

$$f_{max} = \frac{1}{t_{CQ_1} + t_{pd} + t_{su}} \tag{16}$$

where t_{CQ_1} is the clock-to-Q delay of the first flip-flop, t_{pd} is the worst-case propagation delay of the CLA adder block, and t_{su} is the setup time of the flip-flop.

Given the values we calculated earlier:

- $t_{CQ_1} = 61.17$ ps
- $t_{pd} = 479.16$ ps
- $t_{su} = 87.65$ ps

The maximum clock frequency is:

$$f_{max} = \frac{1}{61.17 \text{ ps} + 479.16 \text{ ps} + 87.65 \text{ ps}} \approx 1.59 \text{ GHz} \tag{17}$$

2) Found using Simulation:

D. Maximum Power Consumption of CLA Adder Block

VI. Post-Layout Simulation Results & Analysis of CLA

A. Verification of functionality

B. Worst Case Delay of CLA Adder Block

C. Maximum Clock Speed

D. Maximum Power Consumption of CLA Adder Block

VII. Floor Plan for Complete Circuit Layout

VIII. Verilog HDL Simulation Results

A. Abbreviations and Acronyms

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, ac, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

B. Units

- Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary units (in parentheses). An exception would be the use of English units as identifiers in trade, such as “3.5-inch disk drive”.
- Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.
- Do not mix complete spellings and abbreviations of units: “Wb/m^2” or “webers per square meter”, not “webers/m^2”. Spell out units when they appear in text: “. . . a few henries”, not “. . . a few H”.
- Use a zero before decimal points: “0.25”, not “.25”. Use “cm^3”, not “cc”.

C. Equations

Number equations consecutively. To make your equations more compact, you may use the solidus (/), the exp function, or appropriate exponents. Italicize Roman symbols for quantities and variables, but not Greek symbols. Use a long dash rather than a hyphen for a minus sign. Punctuate equations with commas or periods when they are part of a sentence, as in:

$$a + b = \gamma \tag{18}$$

Be sure that the symbols in your equation have been defined before or immediately following the equation. Use “(18)”, not “Eq. (18)” or “equation (18)”, except at the beginning of a sentence: “Equation (18) is . . .”

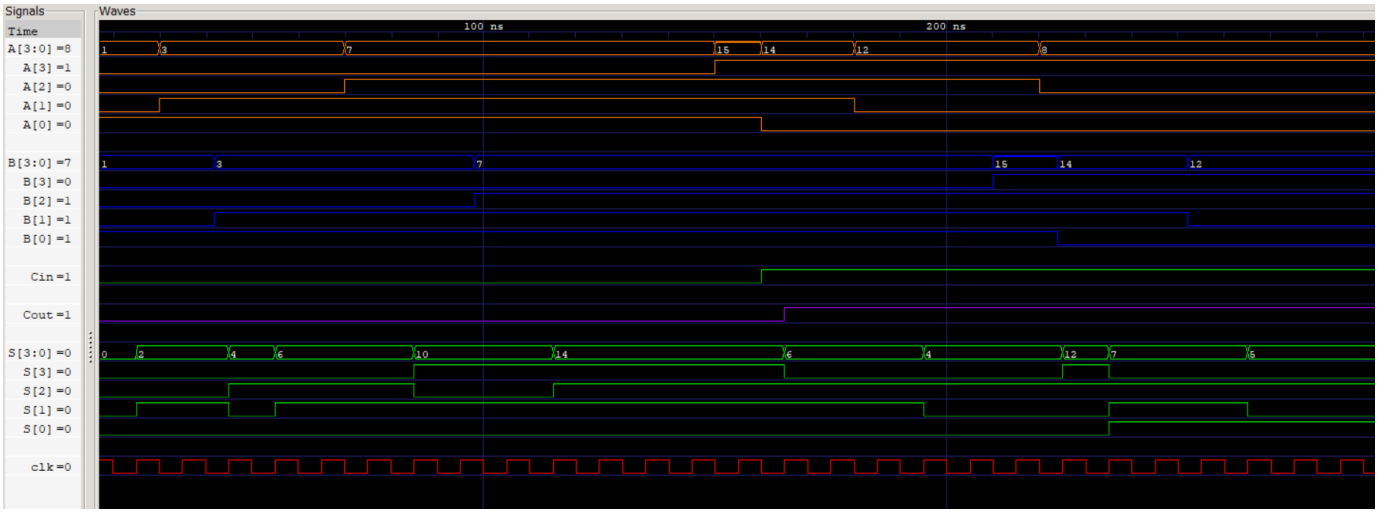

Fig. 16. Verilog HDL Simulation Waveforms

D. LaTeX-Specific Advice

Please use "soft" (e.g., `\eqref{Eq}`) cross references instead of "hard" references (e.g., `(1)`). That will make it possible to combine sections, add equations, or change the order of figures or citations without having to go through the file line by line.

Please don't use the `{eqnarray}` equation environment. Use `{align}` or `{IEEEeqnarray}` instead. The `{eqnarray}` environment leaves unsightly spaces around relation symbols.

Please note that the `{subequations}` environment in LaTeX will increment the main equation counter even when there are no equation numbers displayed. If you forget that, you might write an article in which the equation numbers skip from (17) to (20), causing the copy editors to wonder if you've discovered a new method of counting.

BibTeX does not work by magic. It doesn't get the bibliographic data from thin air but from .bib files. If you use BibTeX to produce a bibliography you must send the .bib files.

LaTeX can't read your mind. If you assign the same label to a subsubsection and a table, you might find that Table I has been cross referenced as Table IV-B3.

LaTeX does not have precognitive abilities. If you put a `\label` command before the command that updates the counter it's supposed to be using, the label will pick up the last counter to be cross referenced instead. In particular, a `\label` command should not go before the caption of a figure or a table.

Do not use `\nonumber` inside the `{array}` environment. It will not stop equation numbers inside `{array}` (there won't be any anyway) and it might stop a wanted equation number in the surrounding equation.

E. Some Common Mistakes

- The word "data" is plural, not singular.
- The subscript for the permeability of vacuum μ_0, and other common scientific constants, is zero with subscript formatting, not a lowercase letter "o".
- In American English, commas, semicolons, periods, question and exclamation marks are located within quotation marks only when a complete thought or name is cited, such as a title or full quotation. When quotation marks are used, instead of a bold or italic typeface, to highlight a word or phrase, punctuation should appear outside of the quotation marks. A parenthetical phrase or statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.)
- A graph within a graph is an "inset", not an "insert". The word alternatively is preferred to the word "alternately" (unless you really mean something that alternates).
- Do not use the word "essentially" to mean "approximately" or "effectively".
- In your paper title, if the words "that uses" can accurately replace the word "using", capitalize the "u"; if not, keep using lower-cased.
- Be aware of the different meanings of the homophones "affect" and "effect", "complement" and "compliment", "discreet" and "discrete", "principal" and "principle".
- Do not confuse "imply" and "infer".
- The prefix "non" is not a word; it should be joined to the word it modifies, usually without a hyphen.
- There is no period after the "et" in the Latin abbreviation "et al.".
- The abbreviation "i.e." means "that is", and the abbreviation "e.g." means "for example".

An excellent style manual for science writers is [7].

F. Authors and Affiliations

The class file is designed for, but not limited to, six authors. A minimum of one author is required for all conference articles. Author names should be listed starting from left

to right and then moving down to the next line. This
author sequence that will be used in future citations a
indexing services. Names should not be listed in colum
group by affiliation. Please keep your affiliations as succi
possible (for example, do not differentiate among depart
of the same organization).

G. Identify the Headings

Headings, or heads, are organizational devices that guid
reader through your paper. There are two types: comp
heads and text heads.

Component heads identify the different componen
your paper and are not topically subordinate to each
Examples include Acknowledgments and References an
these, the correct style to use is "Heading 5". Use "
caption" for your Figure captions, and "table head" for
table title. Run-in heads, such as "Abstract", will requir
to apply a style (in this case, italic) in addition to the
provided by the drop down menu to differentiate the head
the text.

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced.

H. Figures and Tables

a) Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation "Fig. 17", even at the beginning of a sentence.

TABLE VII
TABLE TYPE STYLES

Table Head	Table Column Head		
	Table column subhead	*Subhead*	*Subhead*
copy	More table copy[a]		

[a]Sample of a Table footnote.

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity "Magnetization", or "Magnetization, M", not just "M". If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write "Magnetization (A/m)" or "Magnetization {A[m(1)]}", not just "A/m". Do not label axes with a ratio of quantities and units. For example, write "Temperature (K)", not "Temperature/K".

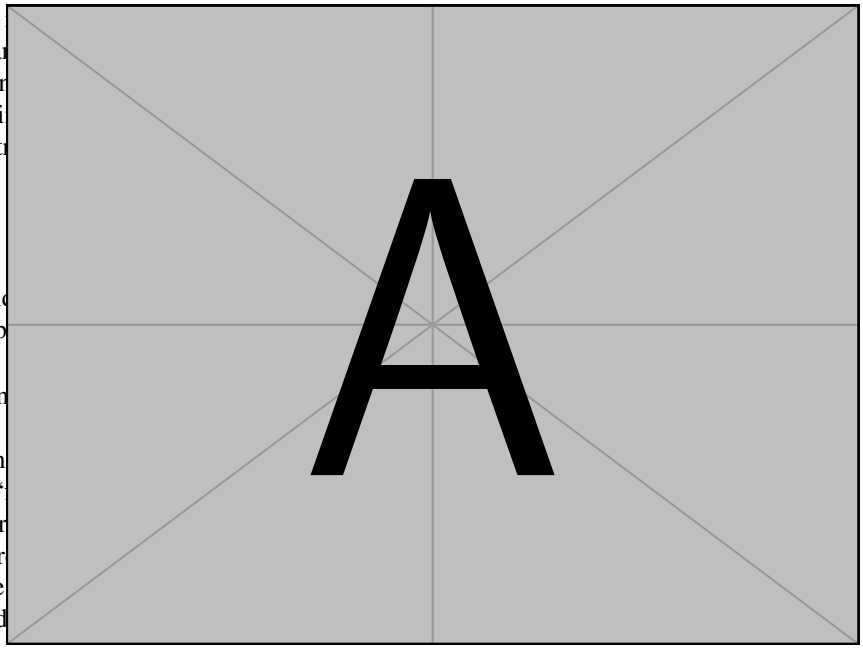

Fig. 17. Example of a figure caption.

ACKNOWLEDGMENT

The preferred spelling of the word "acknowledgment" in America is without an "e" after the "g". Avoid the stilted expression "one of us (R. B. G.) thanks ...". Instead, try "R. B. G. thanks...". Put sponsor acknowledgments in the unnumbered footnote on the first page.

REFERENCES

Please number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use "Ref. [3]" or "reference [3]" except at the beginning of a sentence: "Reference [3] was the first ..."

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors' names; do not use "et al.". Papers that have not been published, even if they have been submitted for publication, should be cited as "unpublished" [4]. Papers that have been accepted for publication should be cited as "in press" [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
[4] K. Elissa, “Title of paper if known,” unpublished.
[5] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
[7] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.