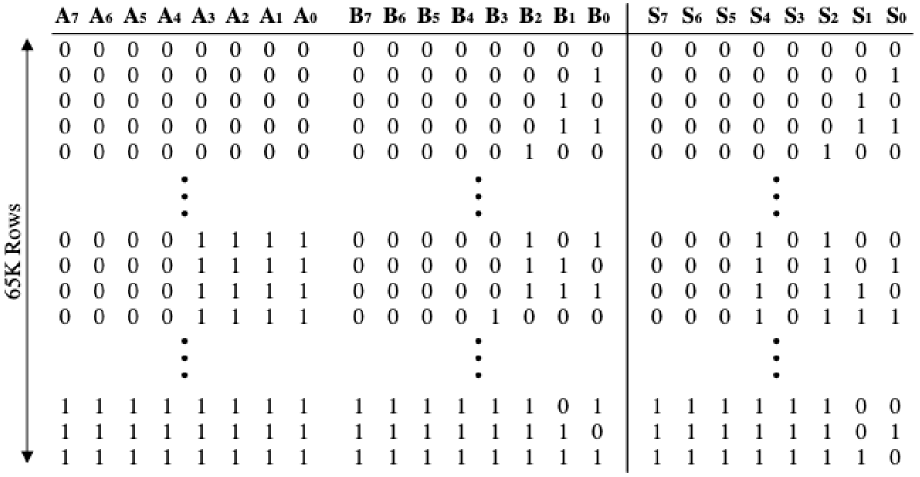
8-bit Full Adder Using NAND-Based Carry Look-Ahead Architecture

Project Task

- An 8-bit full adder
- Adds two 8-bit inputs plus a carry-in
- Outputs an 8-bit sum and a final carry-out
- Verified with LTSpice simulations and Microwind layout



65K Rows

A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0	S_7	S_6	S_5	S_4	S_3	S_2	S_1	S_0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
			⋮								⋮								⋮				
0	0	0	0	1	1	1	1	0	0	0	0	0	1	0	1	0	0	0	1	0	1	0	0
0	0	0	0	1	1	1	1	0	0	0	0	0	1	1	0	0	0	0	1	0	1	0	1
0	0	0	0	1	1	1	1	0	0	0	0	0	1	1	1	0	0	0	1	0	1	1	0
0	0	0	0	1	1	1	1	0	0	0	0	1	0	0	0	0	0	0	1	0	1	1	1
			⋮								⋮								⋮				
1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Image 1: 8-bit full adder truth table [1]

Team Members: John Gellerup, Brandon Markham, and Cassidy Miskovitz

TEXAS STATE
INGRAM SCHOOL OF ENGINEERING

Introduction

Goal: Design a high-speed digital adder that improves performance over traditional approaches.

Problem with RCA: Ripple Carry Adders suffer from sequential carry propagation, which slows down computation as bit-width increases

Our Solution: Implement a Carry Look-Ahead Adder (CLA) that calculates carries in parallel.

Key Constraint: Build the entire system using only NAND gates, demonstrating their universality and enabling a uniform, reusable layout structure

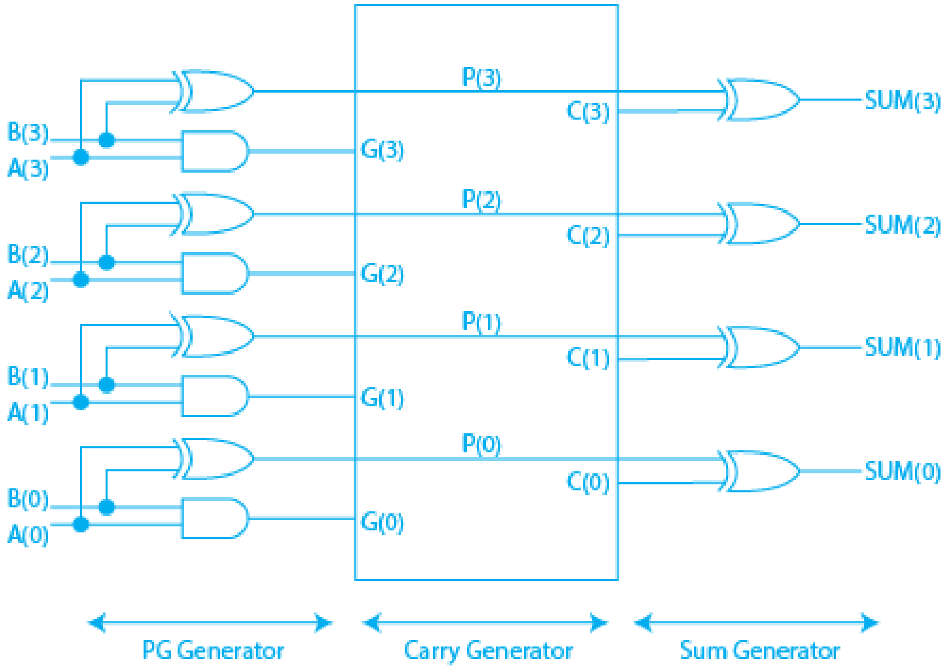


Image 2: Circuit layout for a carry lookahead adder. [2]

TEXAS STATE
INGRAM SCHOOL OF ENGINEERING

1-bit Full Adder Basics

Function of a full adder:

- **Inputs:** A, B, and Carry-In (Cin)
- **Outputs:** Sum and Carry-Out

Logic break down:

- **Sum Logic:** XOR of A, B, and Cin.
- **Carry Logic:** Combination of AND and OR gates.

- **Our Design:** Implemented entirely using **NAND gates**

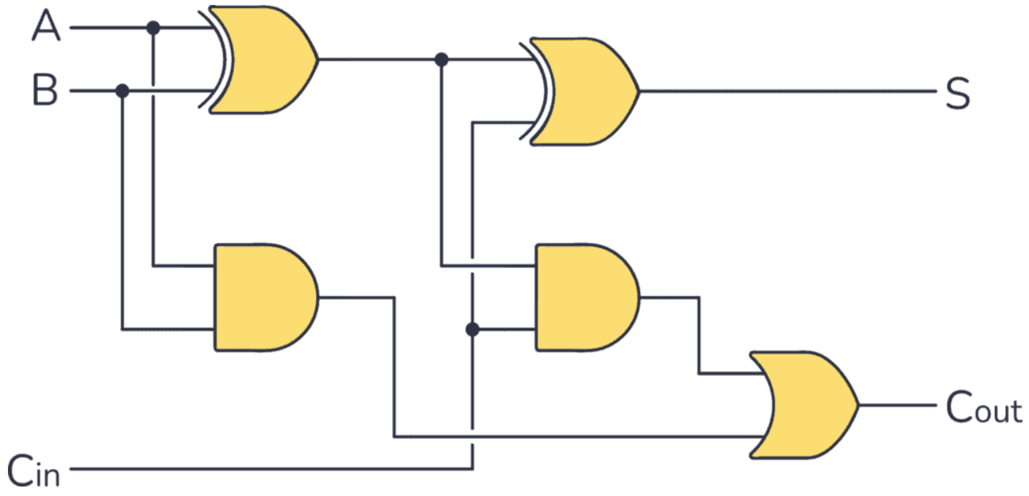


Image 3: Full Adder Circuit. [3]

TEXAS STATE
INGRAM SCHOOL OF ENGINEERING

Why NAND-Only Implementation?

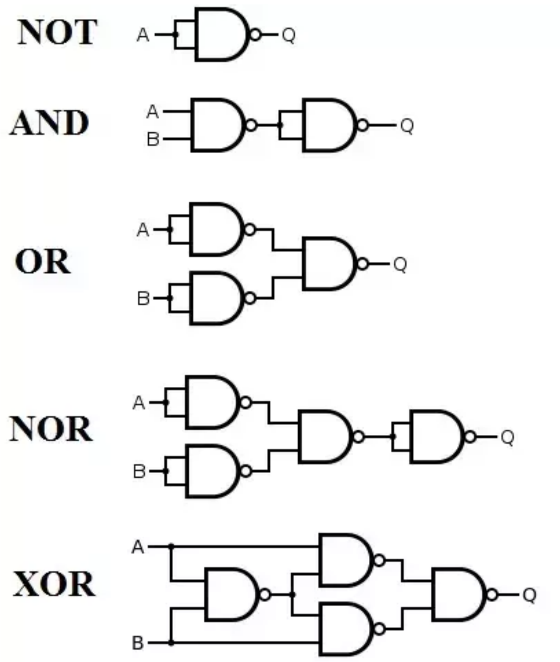


Image 6: Nand Gate Conversion [6]

Pros	Cons
Standardized layout: Same basic gate shape and size	**More gates** needed compared to using mixed gates
Simplifies fabrication: Easier for uniform VLSI cell libraries	**Increased power consumption** (due to more transistors switching)
Compact design: Easier to fit into tight layouts	**Could cause slightly slower individual gates** (more stages for XOR, etc.)

TEXAS STATE INGRAM SCHOOL OF ENGINEERING

Ripple Carry Adder (RCA)

Simple Structure: Chains 1-bit full adders sequentially. (bit-by-bit)

Each carry-out feeds into the next adder's carry-in

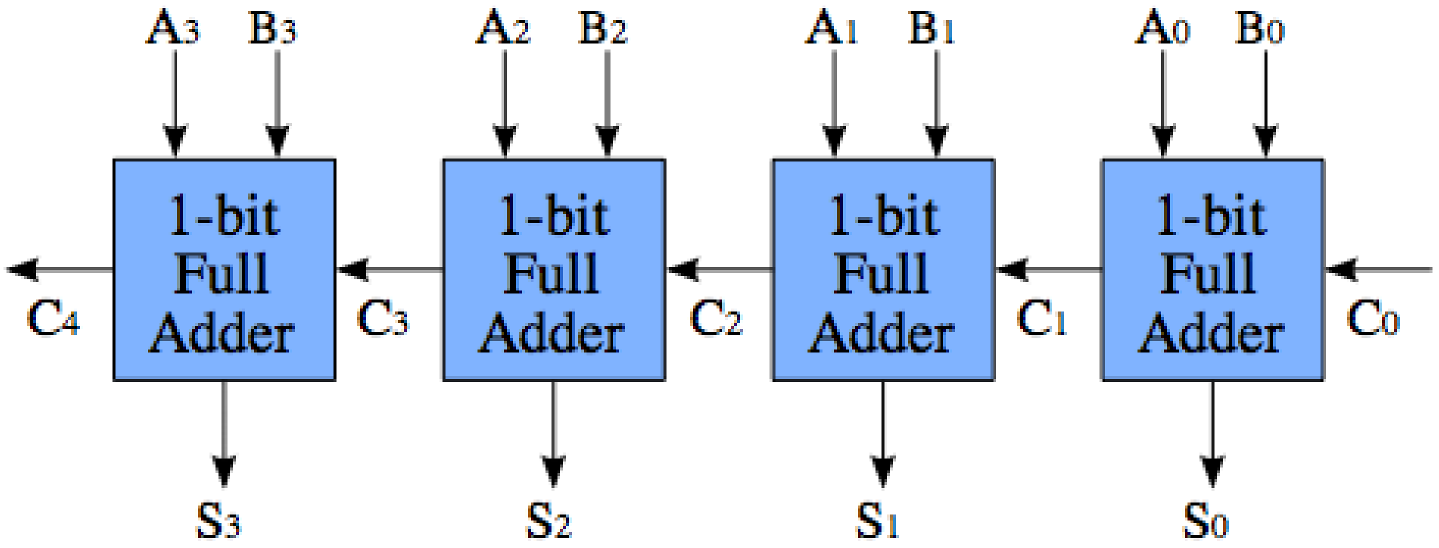


Image 4: Block diagram of 4-bit Ripple Carry Adder [4]

Delay increases **linearly** with the number of bits.

TEXAS STATE INGRAM SCHOOL OF ENGINEERING

Carry Skip Adder (CSA)

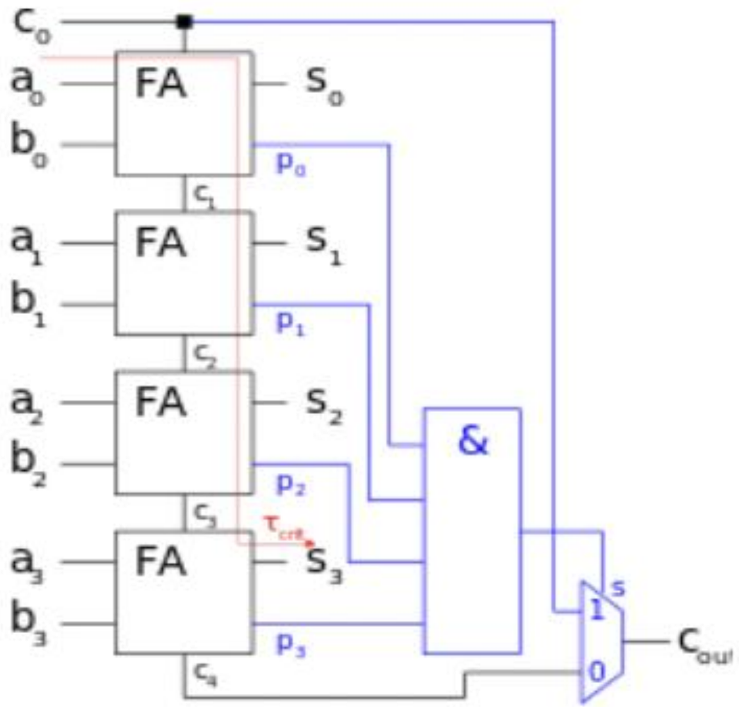
Each block has:

- **Full adders** for sum and carry
- **AND gates** to check propagate condition
- **OR gate** for skip logic
- **Skip Logic**: If all bits propagate, the carry skips the block.

CSA improves over Ripple Carry by skipping over blocks, but it still involves some serial computation.

Delay depends on:

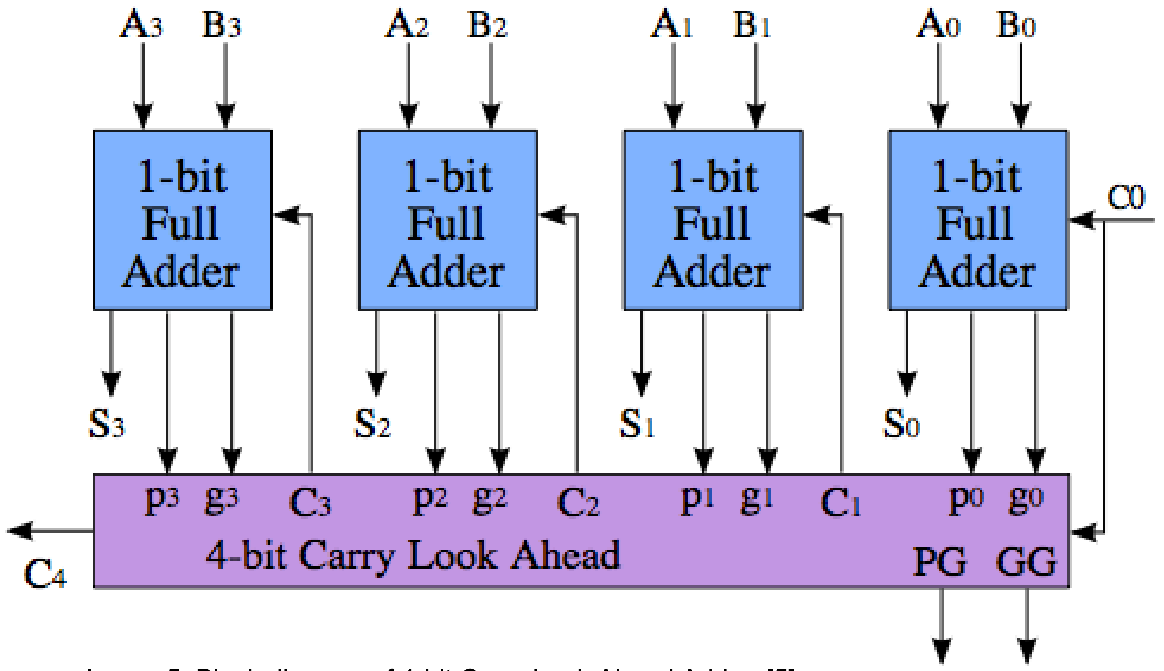
- Time to compute block propagate (AND gate chain)
- Time to skip vs. ripple through the block



TEXAS STATE
INGRAM SCHOOL OF ENGINEERING

Carry Look Ahead Adder (CLA)

Computes carries **in parallel** using **Generate (G)** and **Propagate (P)** signals



A_3 B_3 A_2 B_2 A_1 B_1 A_0 B_0
1-bit Full Adder
C0
S_3 S_2 S_1 S_0
p_3 g_3 C_3 p_2 g_2 C_2 p_1 g_1 C_1 p_0 g_0
4-bit Carry Look Ahead
PG GG
C_4

Image 5: Block diagram of 4-bit Carry Look Ahead Adder [5]

Carry values predicted early instead of waiting stage-by-stage

Removes cumulative delay seen in Ripple Carry Adders. As early as the mid 1800's, Charles Babbage realized propagation delay would be a problem in RCA circuits

First Modern CLA Patent Filed by IBM in 1957

TEXAS STATE INGRAM SCHOOL OF ENGINEERING

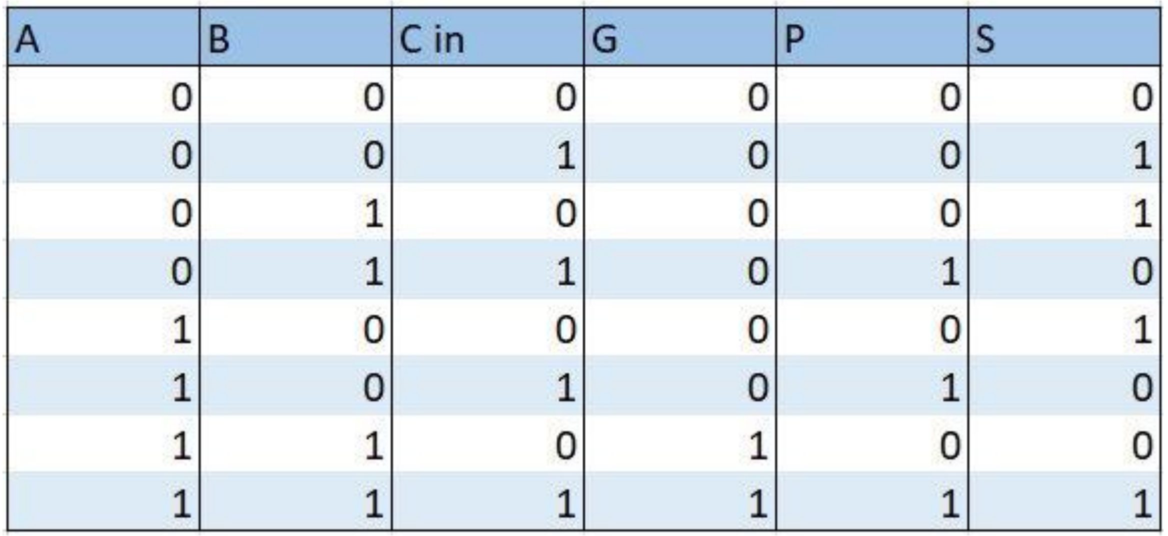
CLA Mathematics (Simple Form)

Generate signal (Gi)
Creates a carry bit
when... $Gi = A_i * B_i$

Propagate signal (Pi)
Passes along an incoming carry
when... $Pi = A_i \oplus B_i$

Carry-out equation
All carries computed in parallel
$Ci+1 = G_i + (P_i * C_i)$

CLA Truth Table



A	B	C in	G	P	S
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	0	1
0	1	1	0	1	0
1	0	0	0	0	1
1	0	1	0	1	0
1	1	0	1	0	0
1	1	1	1	1	1

Image 6: Truth Table for Carry Look Ahead Adder

Final Sum $S_i = Pi \oplus Ci = A_i \oplus B_i \oplus C_i$

TEXAS STATE® INGRAM SCHOOL OF ENGINEERING

CLA Block Structure

8-bit CLA divided into **two 4-bit CLA blocks....Why?**

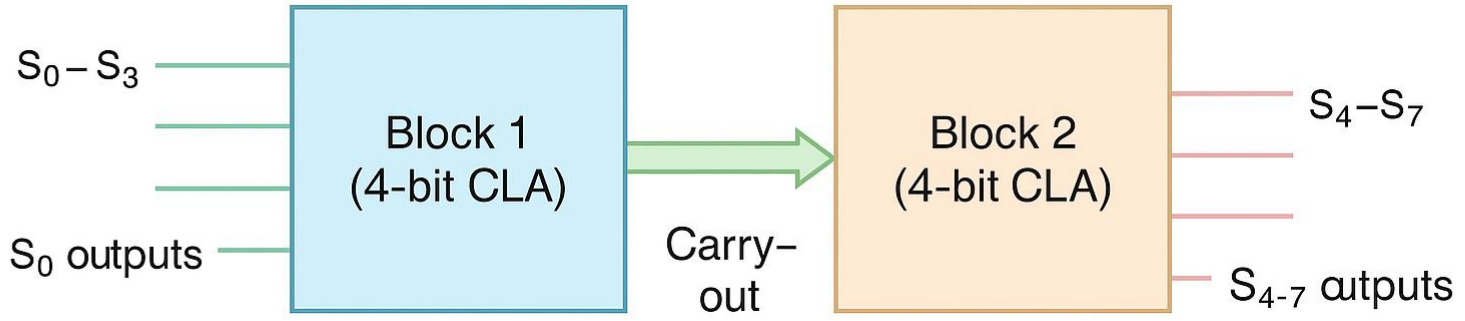


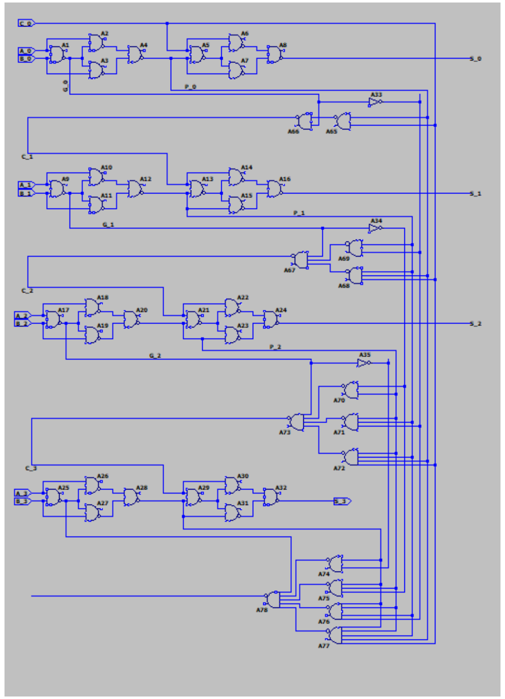
Image 7: Block diagram of two 4-bit Carry Look Ahead Adders making an 8-bit adder

- Simplifies design, simulation, and physical layout
- Enables reuse of logic

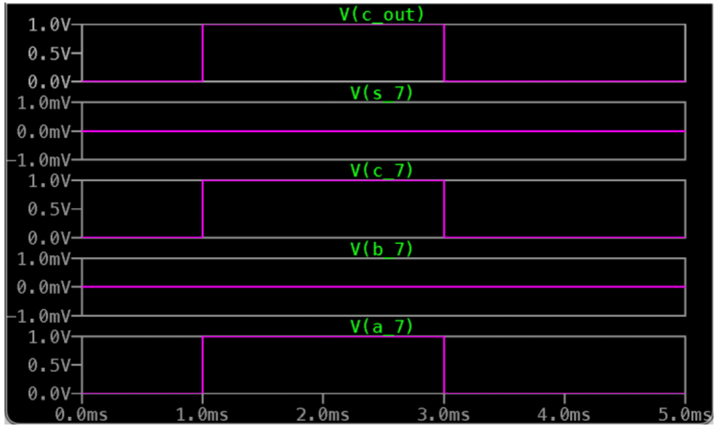
LTSpice Gate-Level Simulation

Bit	Delay (s)	Delay (ns)
S1	2.48E-09	2.48
S2	2.48E-09	2.48
S3	2.48E-09	2.48
S4	2.48E-09	2.48
S5	2.48E-09	2.48
S6	2.48E-09	2.48
S7	2.48E-09	2.48

- Consistent delays across all sum (S_0–S_7) and carry (C_1–C_7) outputs.
- Measured using .meas TRAN in LTspice simulations.
- Uniform delay confirms balanced logic.



Bit	Delay (s)	Delay (ns)
C1	2.48E-09	2.48
C2	2.48E-09	2.48
C3	2.48E-09	2.48
C4	2.48E-09	2.48
C5	2.48E-09	2.48
C6	2.48E-09	2.48
C7	2.48E-09	2.48

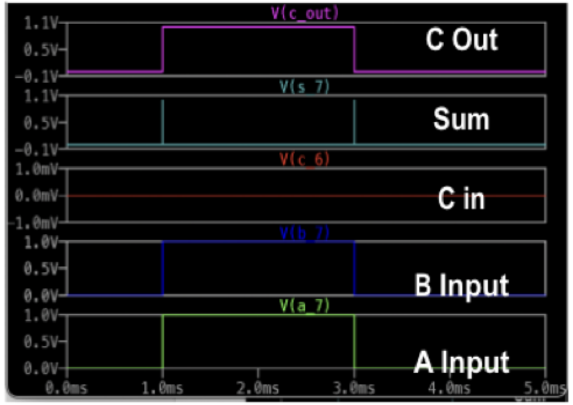


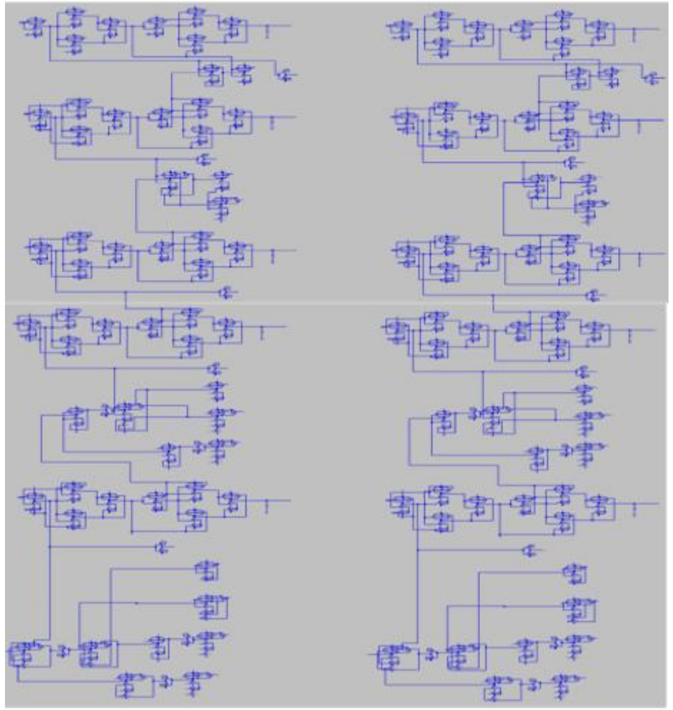
TEXAS STATE
INGRAM SCHOOL OF ENGINEERING

LTSpice Transistor-Level Simulation

Sum Bit	Measured Delay (ns)
S_0	1.74 ns
S_1	1.78 ns
S_2	1.76 ns
S_3	1.73 ns
S_4	1.73 ns
S_5	1.76 ns
S_6	1.73 ns
S_7	1.73 ns

Carry Bit	Delay (ns)
C_1	1.31 ns
C_2	1.94 ns
C_3	2.06 ns
C_4	1.96 ns
C_5	2.00 ns
C_6	2.00 ns
C_7	2.00 ns





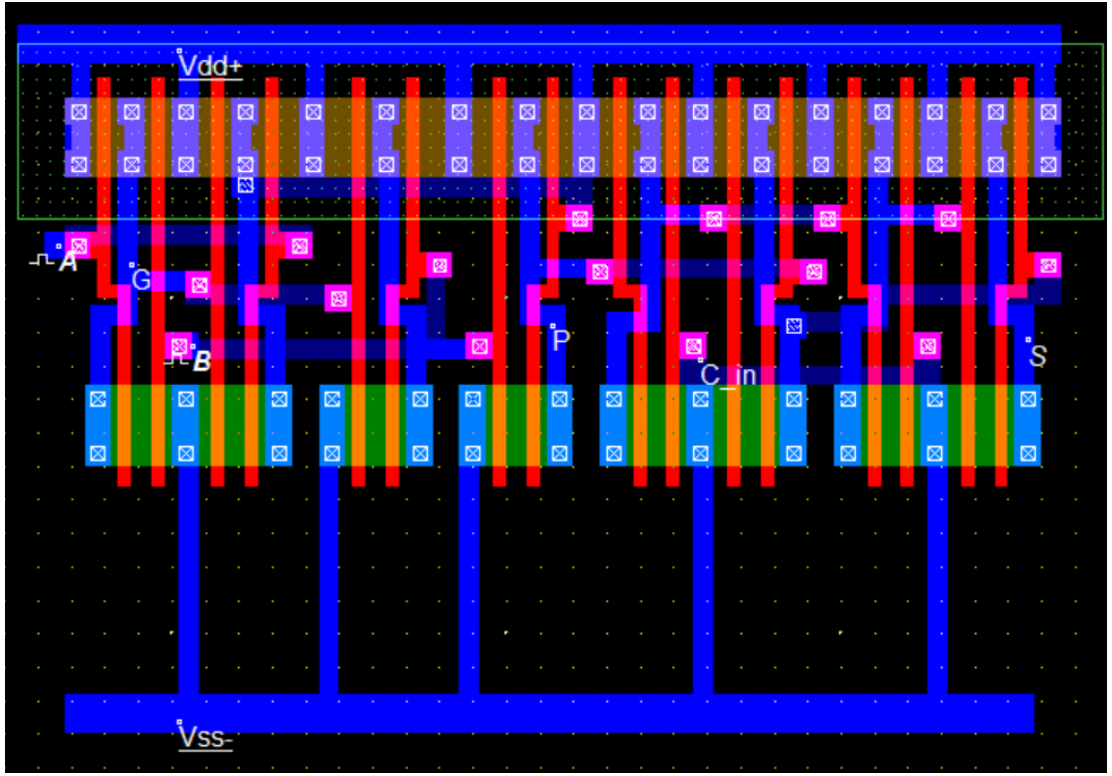
Delay variation across bits was observed:

- Carry outputs ranged from ~1.23 ns (C_1) to ~2.06 ns (C_6),
- Sum delays stayed between ~1.73–1.78 ns.
- Variation is due to gate loading and fanout differences,

TEXAS STATE INGRAM SCHOOL OF ENGINEERING

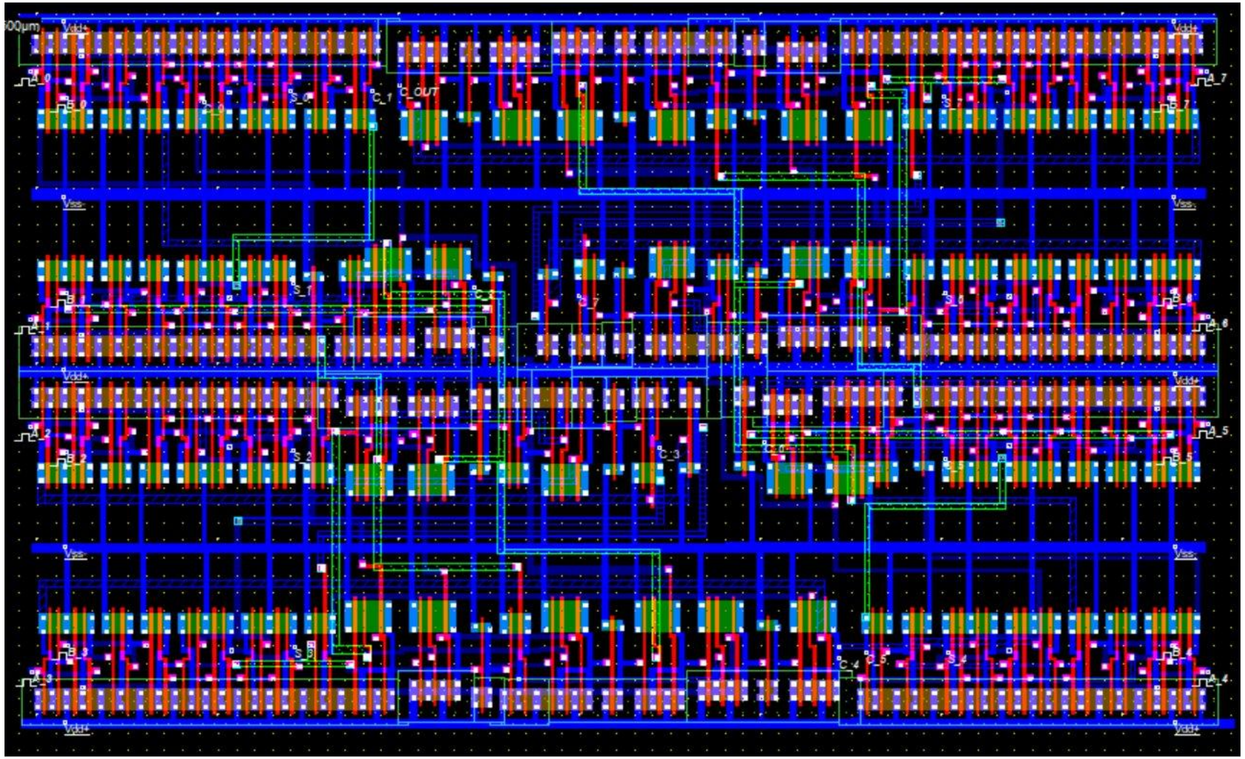
Microwind Layout – Full Adder

- Two 2 input XOR gates converted to NAND.
- Used in each stage
- Optimized for layout efficiency

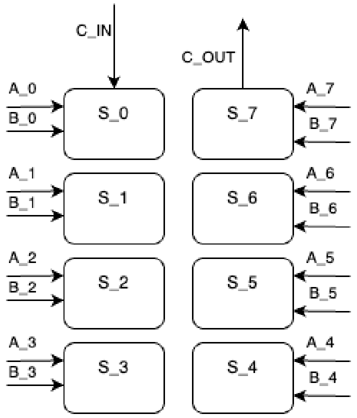


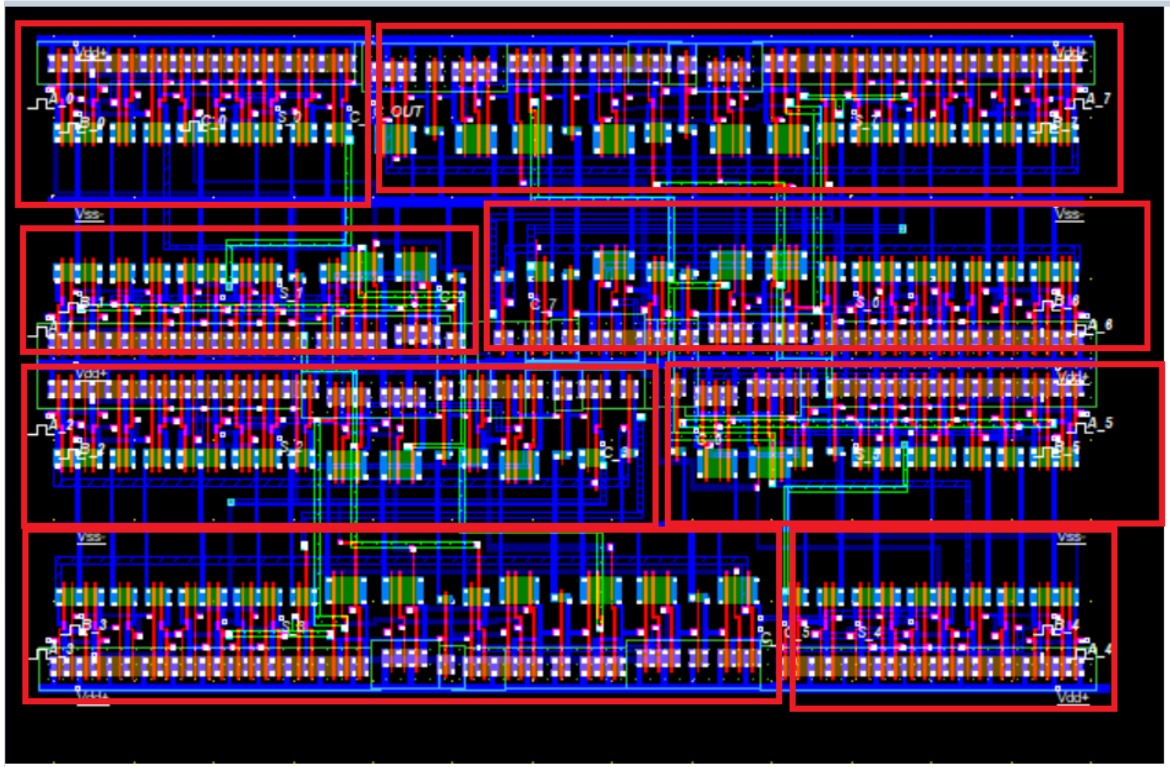
Microwind Layout

- Reused standard cell layouts from previous labs for modularity and consistency.
- Flipped Stages to reuse power rails; Mirrored and flipped the first block to create second block
- DRC clean!
- 6 layers of metal



Microwind Layout – Stages



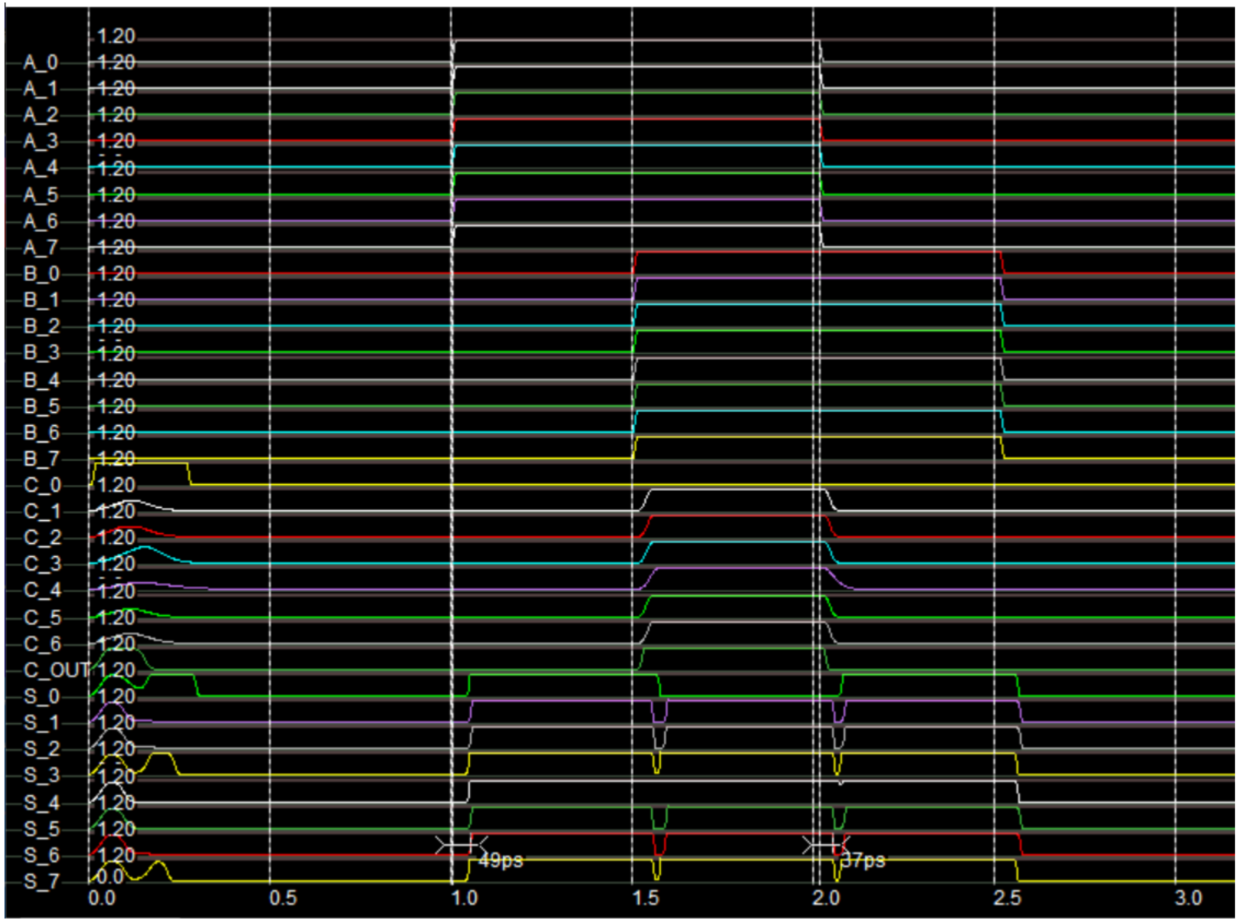


- First 4 bits make up first block of CLA, next 4 bits make up second block

- Starts at the top left and moves in a "U" pattern to end at top right

TEXAS STATE INGRAM SCHOOL OF ENGINEERING

Microwind Post-Layout Simulation



- Expected results

- Average delay from input to sum out of: 42ps

- Sum out are all available at the same time; demonstrates advantage of CLA

TEXAS STATE
INGRAM SCHOOL OF ENGINEERING

Analysis & Key Takeaways

Microwind CLA:

- lower delays reflect an efficient and compact **physical layout**

Gate-level CLA:

- Uniform delay due to symmetrical logic and balanced fanout.

Transistor-level CLA:

- Delay variations from internal parasitics

Metric	Gate-Level CLA (20 fF)	Transistor-Level CLA (20 fF)	Microwind Level CLA (15fF)
Sum Delay (avg)	2.48 ns	~1.75 ns	42ps
Carry Delay (avg)	2.48 ns	~1.92 ns	58ps

Conclusion & Future Work

Conclusion:

- Accomplished goals and met all requirements
- Successfully implemented 8 bit Carry Look Ahead Adder in an area-optimized layout

Main Challenges:

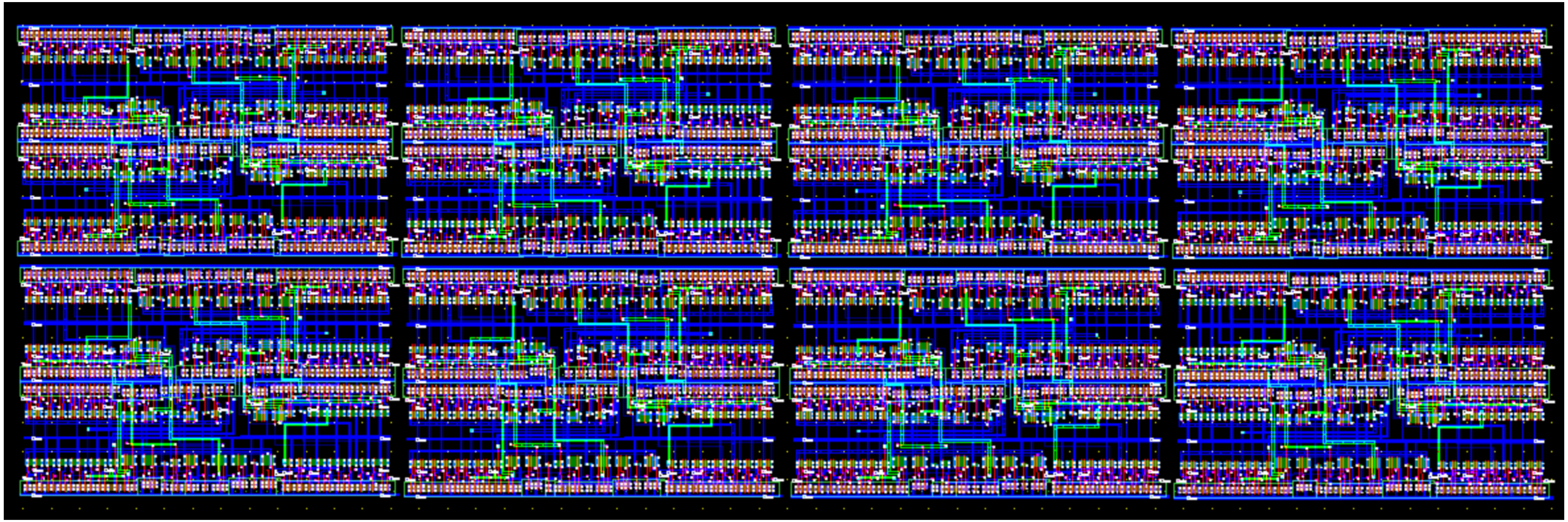
- Gate count: ~125gates; ~500 transistors
- Layout complexity
- Carry logic requires some long traces

Future:

- Reduce parasitic capacitance from long traces to improve performance
- Attempt improved CLA such as Kogge-Stone adder or Brent-Kung Adder - types of parallel-prefix CLA that allows less delay. [7]

TEXAS STATE®
INGRAM SCHOOL OF ENGINEERING

64 Bit CLA Adder



TEXAS STATE
INGRAM SCHOOL
OF ENGINEERING

Questions & Comments

TEXAS STATE®
INGRAM SCHOOL
OF ENGINEERING

References

[1] Real Digital, "Carry Lookahead Adders," RealDigital.org. [Online]. Available: https://www.realdigital.org/doc/9eca319eb242a07d31574667ef02360a. [Accessed: 26-Apr-2025].
[2] PrepBytes, "Carry Look Ahead Adder," PrepBytes.com, 2022. [Online]. Available: https://www.prepbytes.com/blog/computer-network/carry-look-ahead-adder/. [Accessed: 26-Apr-2025].
[3] Ø. Holmeide, "Full Adder," Build-Electronic-Circuits.com. [Online]. Available: https://www.build-electronic-circuits.com/full-adder/. [Accessed: 26-Apr-2025].
[4] Virtual Labs, IIT Kharagpur, "Implementation of Adder Circuits," vlabs.iitkgp.ernet.in. [Online]. Available: http://vlabs.iitkgp.ernet.in/coa/exp1/index.html. [Accessed: 26-Apr-2025].
[5] Virtual Labs, IIT Kharagpur, "Carry Lookahead Adder," vlabs.iitkgp.ac.in. [Online]. Available: http://vlabs.iitkgp.ac.in/coa/exp2/content.html. [Accessed: 26-Apr-2025].
[6] user207421, "I am trying to implement a function using only NAND gates," *Electronics Stack Exchange*, 2022. [Online]. Available: https://electronics.stackexchange.com/questions/588189/i-am-trying-to-implement-a-function-using-only-nand-gates. [Accessed: 26-Apr-2025].
[7]Kogge-stone adder," *Computer Science Wiki*. https://computersciencewiki.org/index.php/Kogge-stone_adder (accessed Apr. 30, 2025).

TEXAS STATE INGRAM SCHOOL OF ENGINEERING