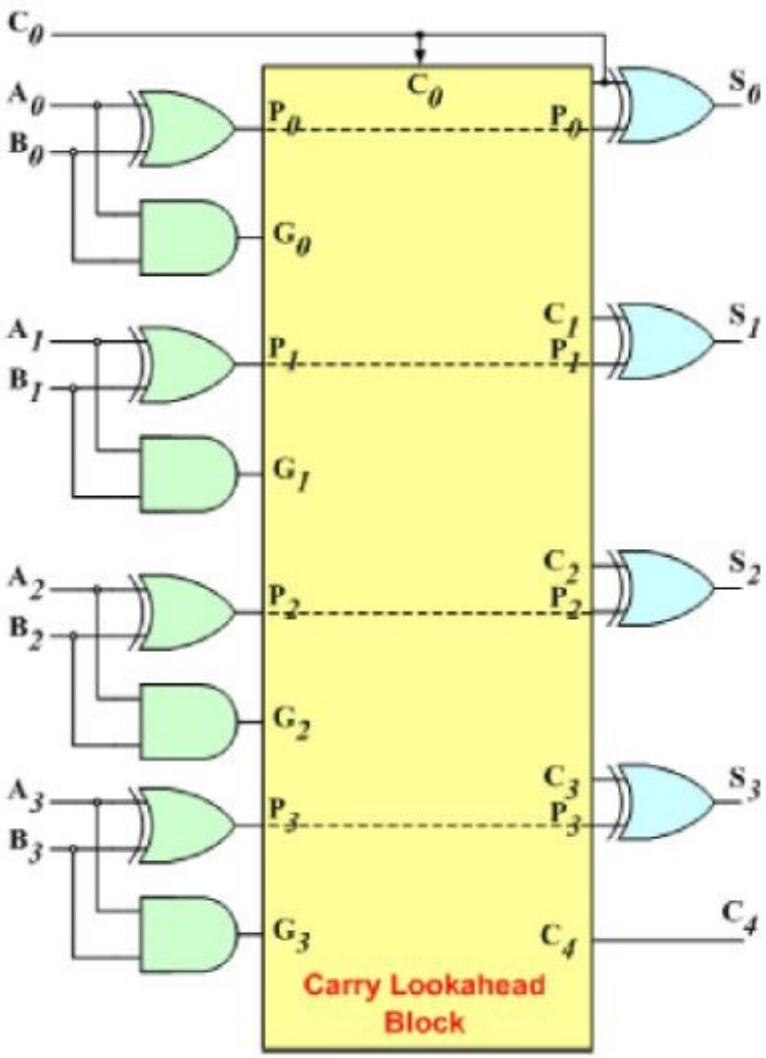


Demystifying Speed: Exploring the 4-Bit Carry Lookahead Adder

Introduction to advanced adder architectures for faster arithmetic operations

EEM410 - IC DESIGN

SAMET BAYAT - 22293730
RAMAZAN ALPER - 22098516

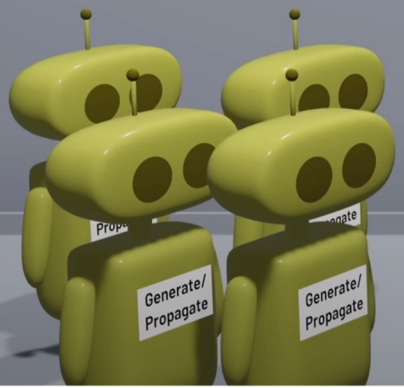
Nazım Görkem ALPYILDIZ (21793608)

$$C_1 = G_0$$

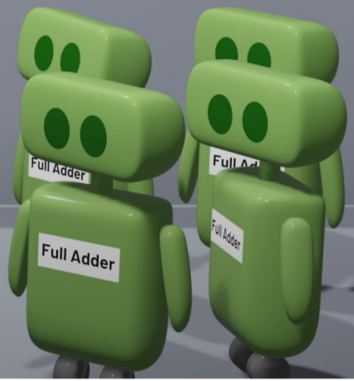
$$C_2 = G_1 \text{ or } G_0 P_1$$

$$C_3 = G_2 \text{ or } G_1 P_2 \text{ or } G_0 P_1 P_2$$

$$C_4 = G_3 \text{ or } G_2 P_3 \text{ or } G_1 P_2 P_3 \text{ or } G_0 P_1 P_2 P_3$$



$$C_{n+1} = G_n \text{ or } C_n P_n$$



$$P_i = A_i \oplus B_i$$

$$G_i = A_i \cdot B_i$$

$$C_{i+1} = P_i \cdot C_i + G_i \qquad i = 1,2,3,4 \quad Sum_i = P_i \oplus C_i$$

$$C_1 = P_0 \cdot C_0 + G_0 = P_0 \cdot C_{in} + G_0$$

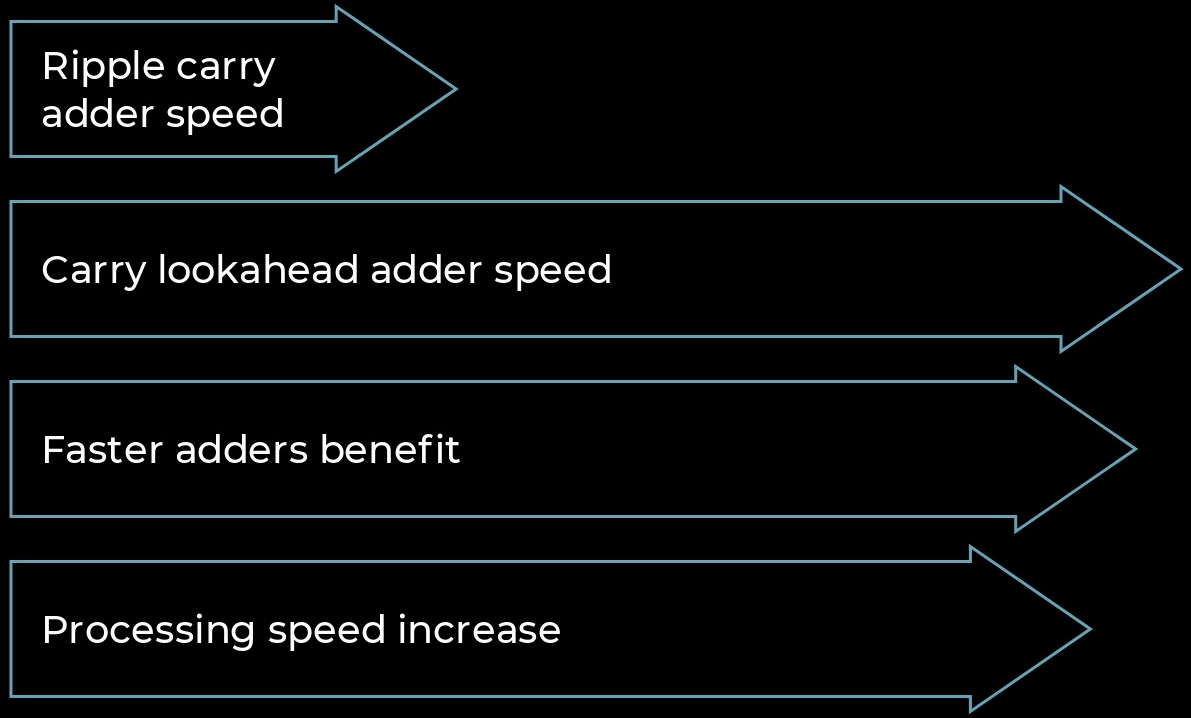
$$C_2 = P_1 \cdot C_1 + G_1 = P_1 \cdot (P_0 \cdot C_0 + G_0) + G_1$$
$$= P_1 \cdot P0 \cdot C_0 + P_1 \cdot G_0 + G_1$$

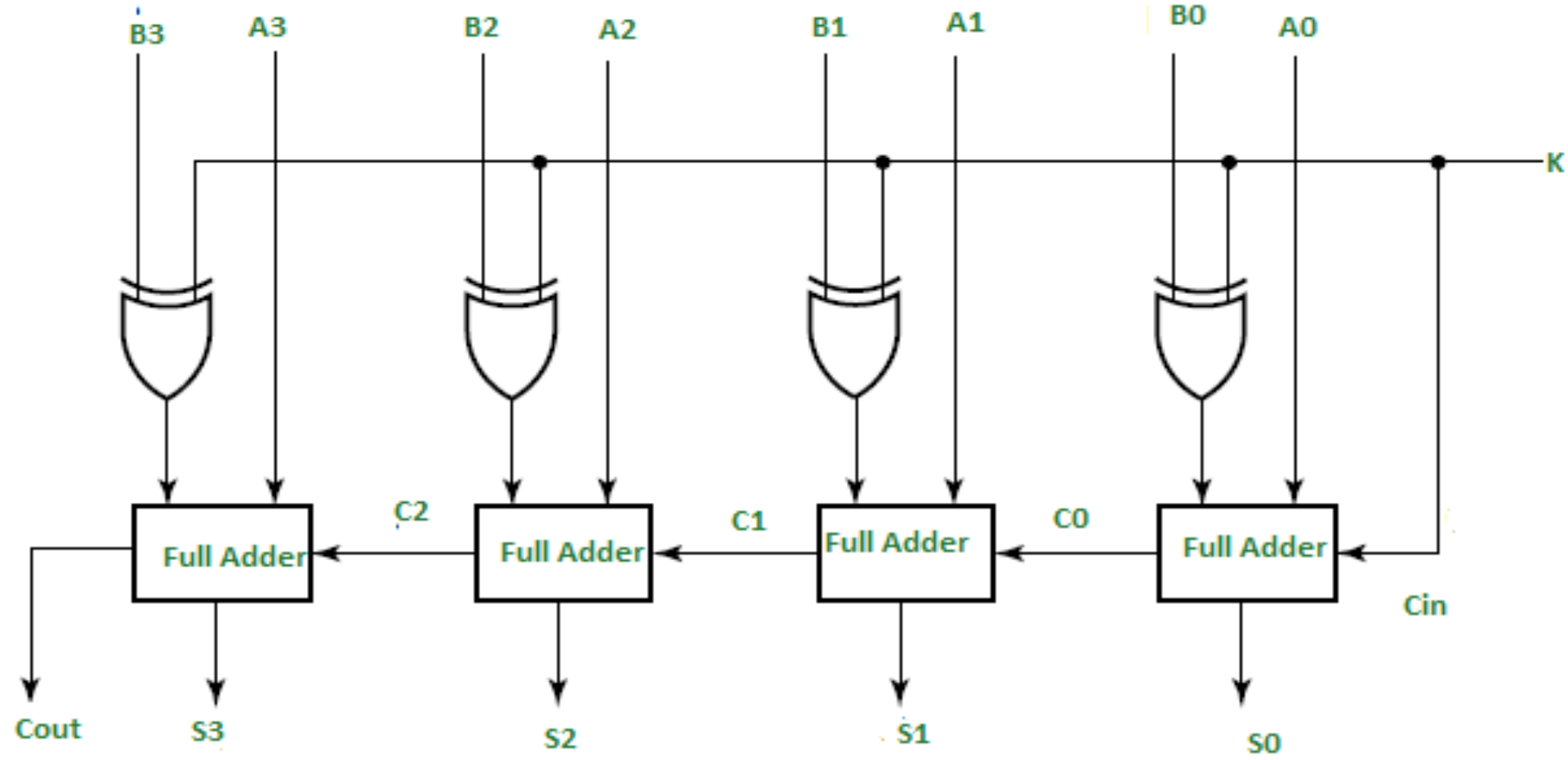
$$C_3 = P_2 \cdot C_2 + G_2 = P_2 \cdot (P_1 \cdot C_1 + G_1) + G_2$$
$$= P_2 \cdot P1 \cdot C_1 + P_2 \cdot G_1 + G_2$$
$$= P_2 \cdot P1 \cdot (P_0 \cdot C_0 + G_0) + P_2 \cdot G_1 + G_2$$
$$= P_2 \cdot P1 \cdot P_0 \cdot C_0 + P_2 \cdot P1 \cdot G_0 + P_2 \cdot G_1 + G_2$$

$$C_{out} = P_3 \cdot C_3 + G_3 = P_3 \cdot (P_2 \cdot C_2 + G_2) + G_3$$
$$= P_3 \cdot P2 \cdot C_2 + P_3 \cdot G_2 + G_3$$
$$= P_3 \cdot P2 \cdot (P_1 \cdot C_1 + G_1) + P_3 \cdot G_2 + G_3$$
$$= P_3 \cdot P2 \cdot P_1 \cdot C_1 + P_3 \cdot P2 \cdot G_1 + P_3 \cdot G_2 + G_3$$
$$= P_3 \cdot P2 \cdot P_1 \cdot (P_0 \cdot C_0 + G_0) + P_3 \cdot P2 \cdot G_1 + P_3 \cdot G_2 + G_3$$
$$= P_3 \cdot P2 \cdot P_1 \cdot P_0 \cdot C_0 + P_3 \cdot P2 \cdot P_1 \cdot G_0 + P_3 \cdot P2 \cdot G_1 + P_3 \cdot G_2 + G_3$$

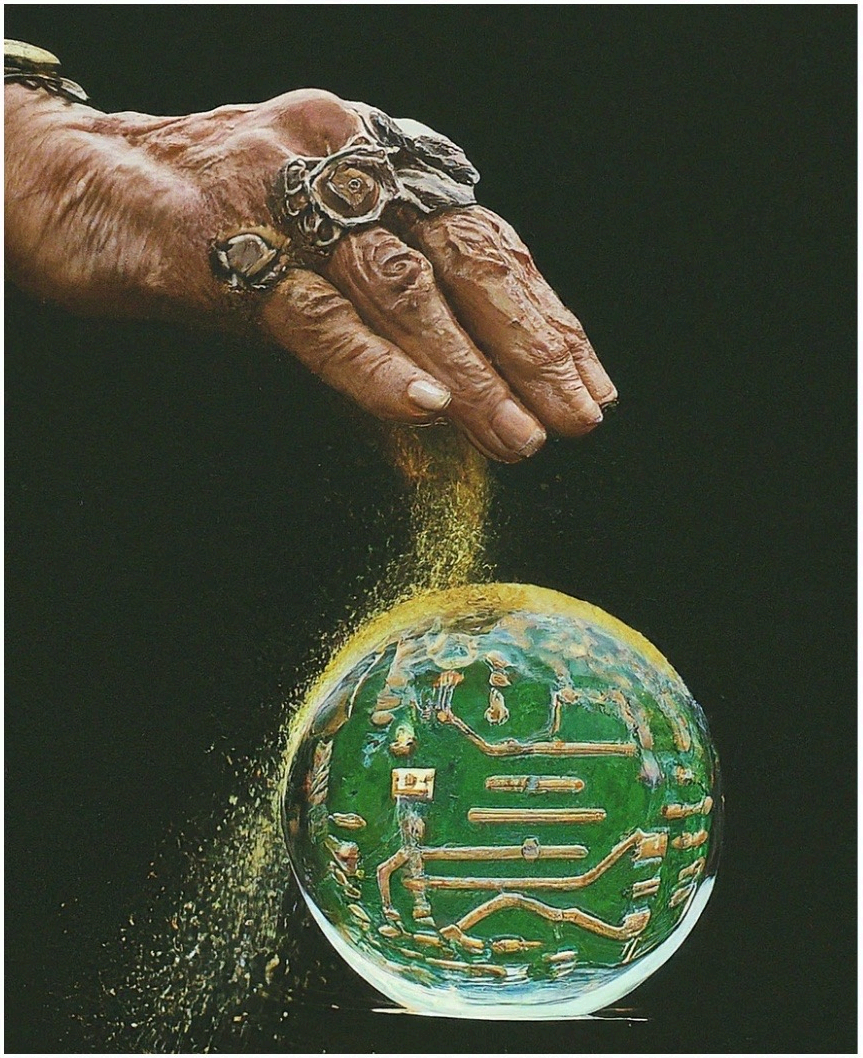
$$C_{i+1} = P_i \cdot C_i + G_i = (A_i \oplus B_i) \cdot C_i + A_i \cdot B_i$$

The Need for Speed





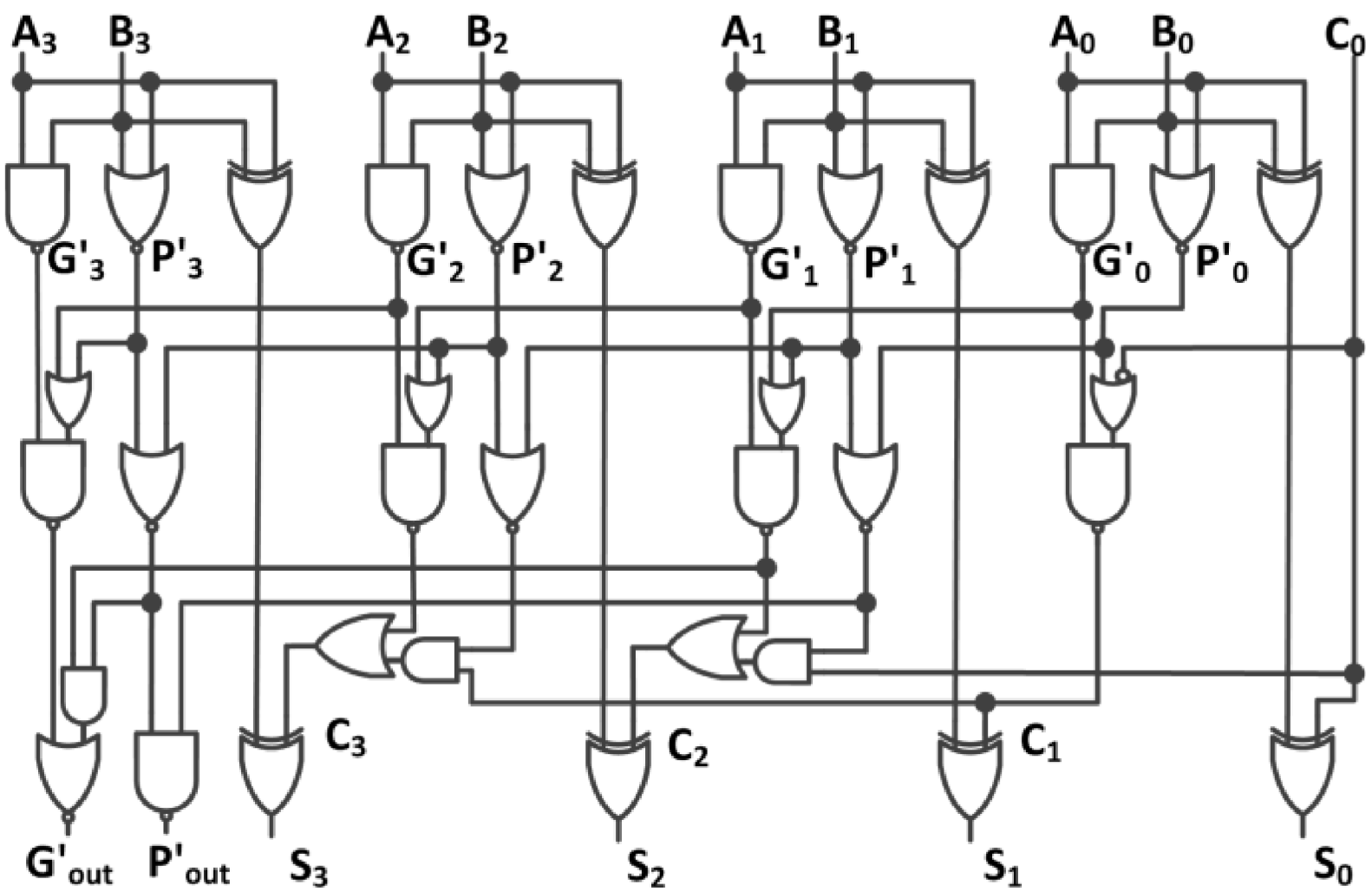
geeksforgeeks.com



CLA

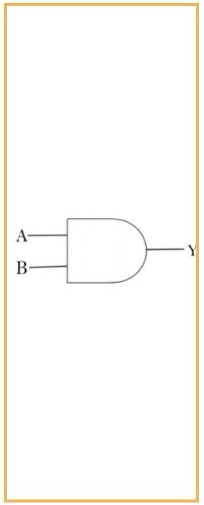
carry lookahead adder

image generated by Gemini.



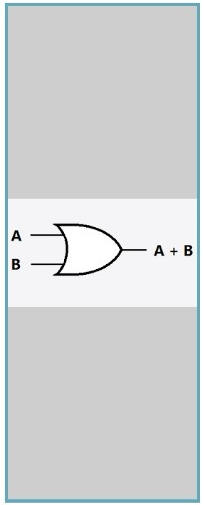
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8126457

Building Blocks: Logic Gates



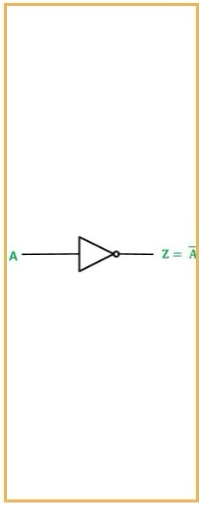
AND gate symbol

The AND gate outputs 1 only if both inputs are 1, otherwise outputs 0.



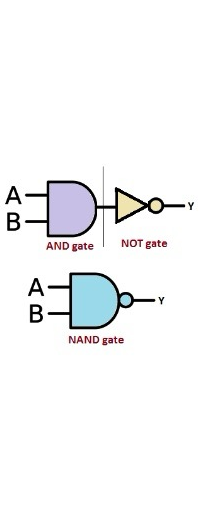
OR gate symbol

The OR gate outputs 1 if one or both inputs are 1, otherwise outputs 0.



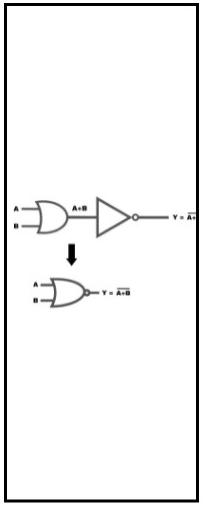
NOT gate symbol

The NOT gate outputs the opposite of its single input.



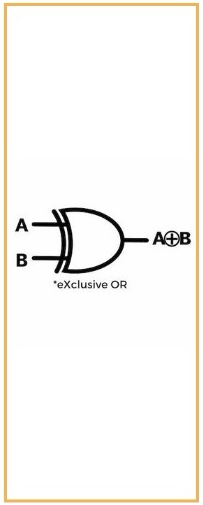
NAND gate symbol

The NAND gate outputs the opposite of the AND gate.



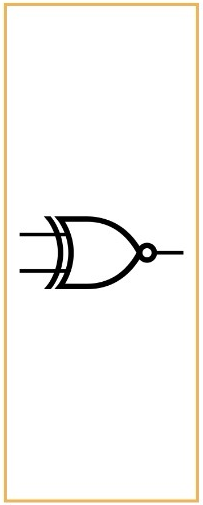
NOR gate symbol

The NOR gate outputs the opposite of the OR gate.



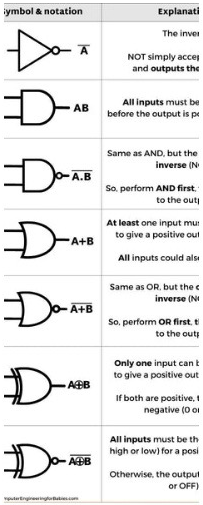
XOR gate symbol

The XOR gate outputs 1 if inputs differ, otherwise outputs 0.



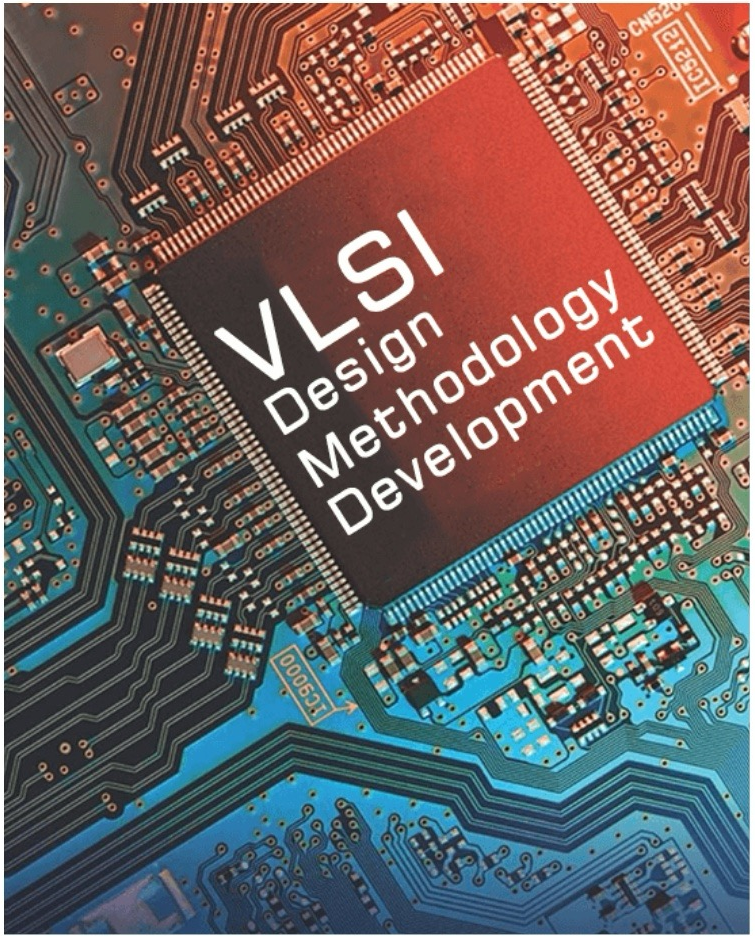
XNOR gate symbol

The XNOR gate outputs the opposite of the XOR gate.



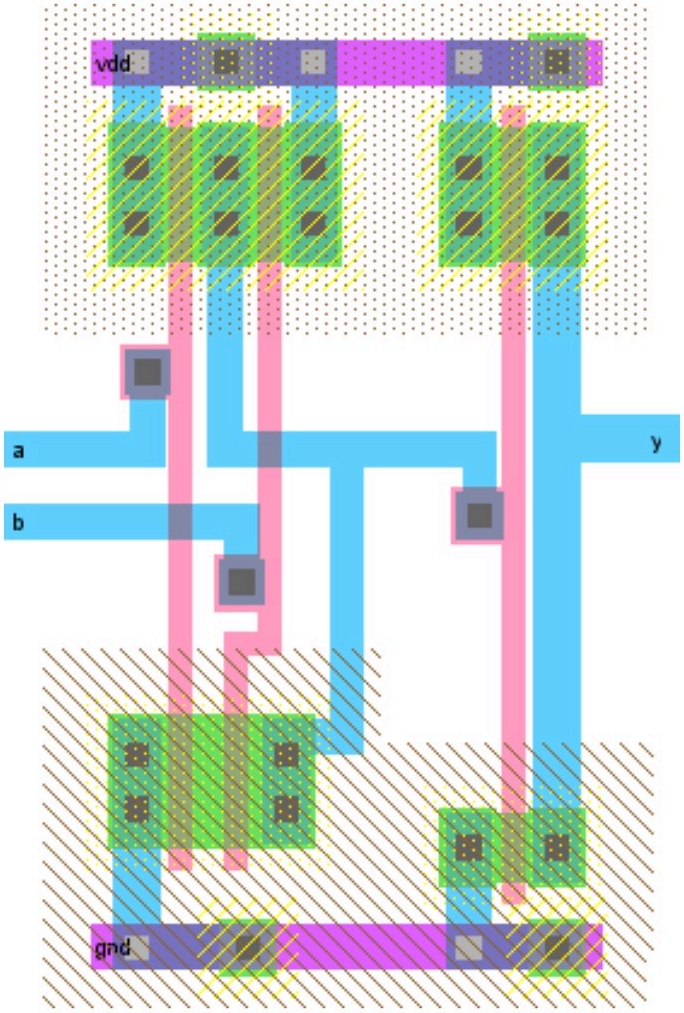
Logic gate circuit

A simple logic gate circuit combining AND, OR, and NOT gates.



A Look Inside: Electric VLSI

VLSI (Very Large Scale Integration) enables complex digital circuits by packing millions of transistors onto a single integrated circuit chip. This high level of integration allows for greater functionality and performance. The carry lookahead adder explored in this presentation takes advantage of VLSI to quickly perform 4-bit addition in a single clock cycle.


vdd
a
b
y
gnd

Total Chip Area

A = 4,040,400 * (.180 µm)^2

A = 130,908.96 sq. µm

Total Transistor Count

Gates

$T_inv=2$

$T_nand=4$

$T_xor=12$

LCU Modules

$T_C1=2(4)+2=10$

$T_C2=4(4)+2(2)=20$

$T_C3=6(4)+3(2)=30$

$T_PG4=8+2=10$

4-bit CLA

$T_(CLA-4)=4(30)+10+20+2(30)+10$

$T_(CLA-4)=220$

Introducing the Carry Lookahead Adder

What is a Carry Lookahead Adder?

A CLA calculates carries ahead of time to overcome limitations of ripple carry adders.

How Does it Work?

A CLA uses propagate and generate signals to determine carries before the sum.

Advantages Over Ripple Carry

Faster computation by reducing wait time for carries to ripple through each full adder.

Implementation

CLAs can be implemented with basic logic gates like AND, OR, and Inverters.

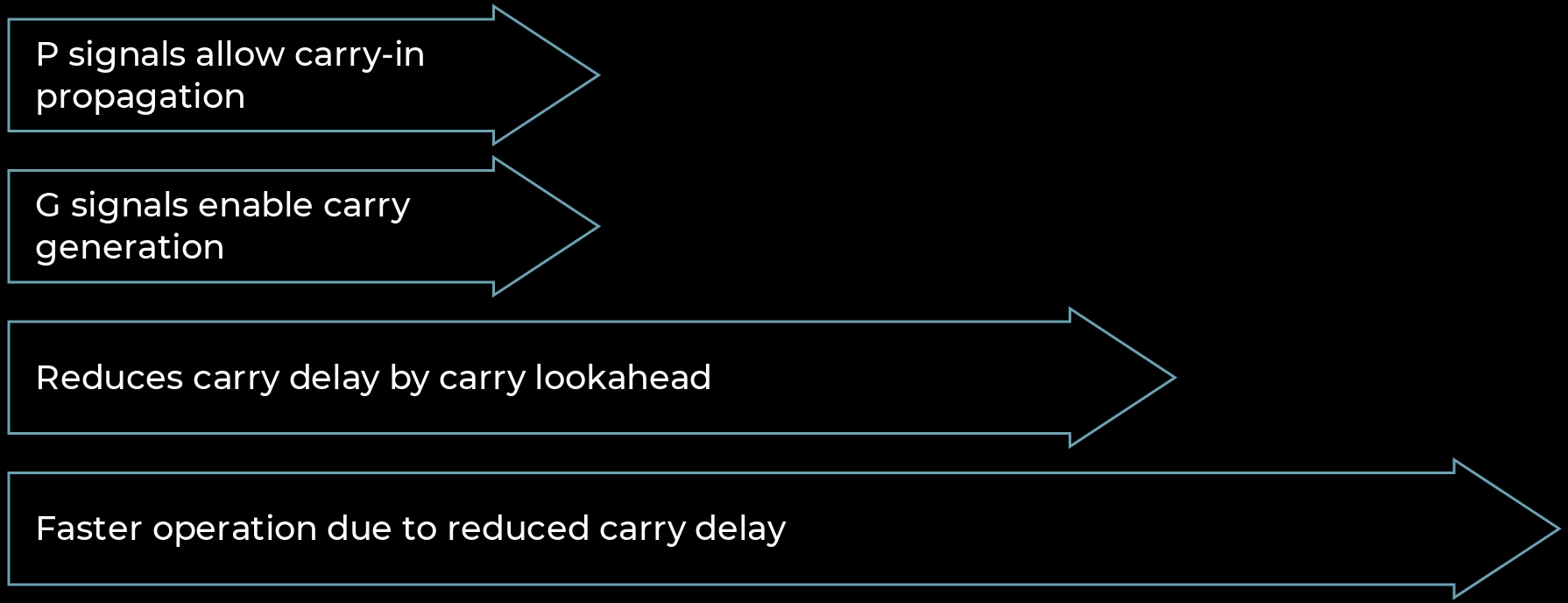
Applications

Used in CPUs and other digital circuits needing fast addition.

Challenges

More complex logic and circuitry than ripple carry adders.

Demystifying P and G Signals



Benefits and Applications

Faster clock speeds for CPUs

Since the CLA performs carry propagation faster, it enables CPUs to operate at higher clock frequencies.

Lower latency for operations

The CLA reduces the latency for arithmetic logic unit (ALU) operations involving carries.

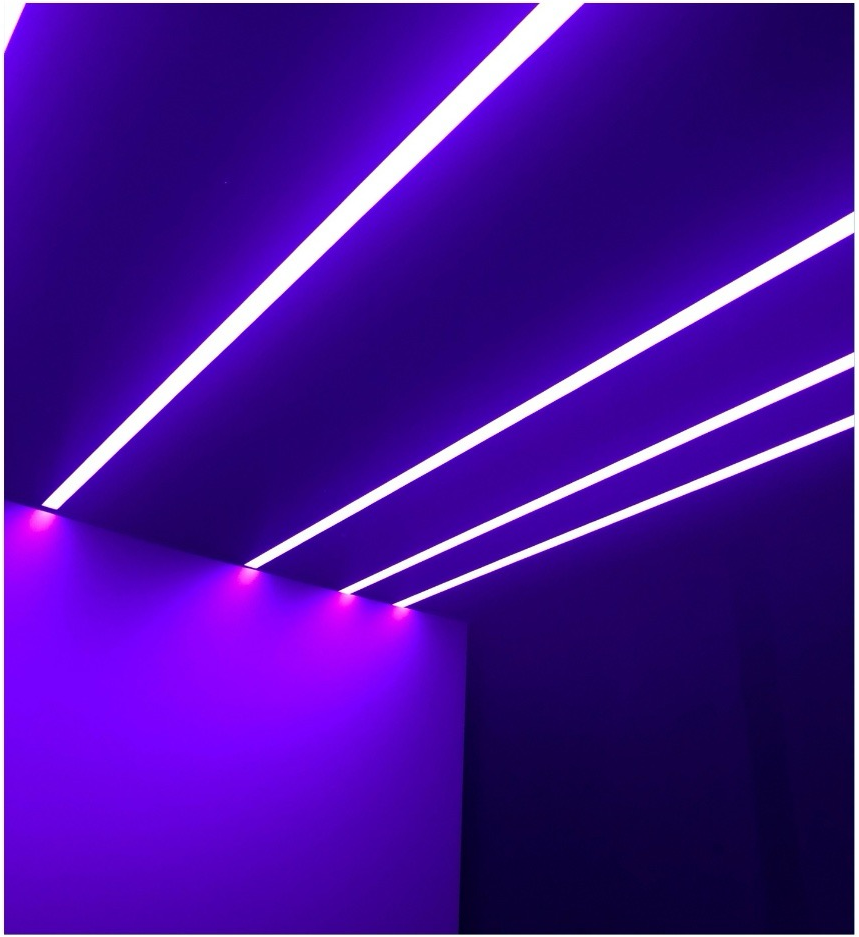
Faster processing

By reducing operation latency, the CLA boosts overall processing speed for applications like video encoding, 3D rendering, scientific computing, etc.

The 4-bit carry lookahead adder enables higher CPU clock speeds and faster processing for compute-intensive applications by accelerating carry propagation.

Where..?

- **CPUs (Central Processing Unit)**
  The ALU (arithmetic logic unit)in CPUs uses CLAs to speed up addition, subtraction, etc.
- **DSPs (Digital Signal Processors)**
  CLAs improve speed and efficiency of addition/multiplication in DSPs
- **Cryptography**
- **HPCs (High-performance computing)**
  Supercomputers use CLAs for fast scientific/engineering computations
- **CLAs assist in error correction**
  for fast encoding/decoding



Conclusion

The 4-bit carry lookahead adder allows for faster addition by pre-computing the carry bits. This enables all the sum bits to be calculated in parallel, significantly speeding up the addition operation. The key takeaways are that carry lookahead reduces delay by avoiding rippling carries, and parallelizes the addition process by calculating multiple sums simultaneously.

References

1. J. Miao and S. Li, "A novel implementation of 4-bit carry look-ahead adder," *2017 International Conference on Electron Devices and Solid-State Circuits (EDSSC)*, Hsinchu, Taiwan, 2017, pp. 1-2, doi: 10.1109/EDSSC.2017.8126457.
2. https://github.com/jitendrasb24/Carry_Lookahead_Adder (Access Date: Apr 15, 24.)

Q
A