Digital Design

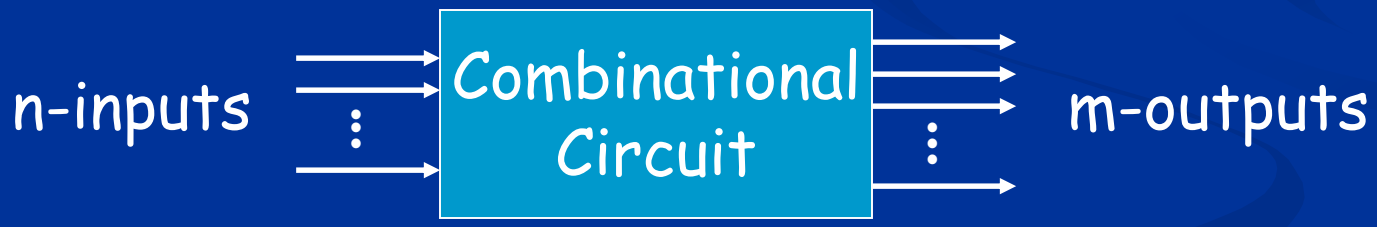
Lecture 11: Combinational Logic and Arithmetic Circuits

Combinational Circuits

- A combinational circuit consists of logic gates whose outputs, at any time, are determined by combining the values of the inputs.
- For n input variables, there are 2^n possible binary input combinations.
- For each binary combination of the input variables, there is one possible output.

Combinational Circuits (cont.)

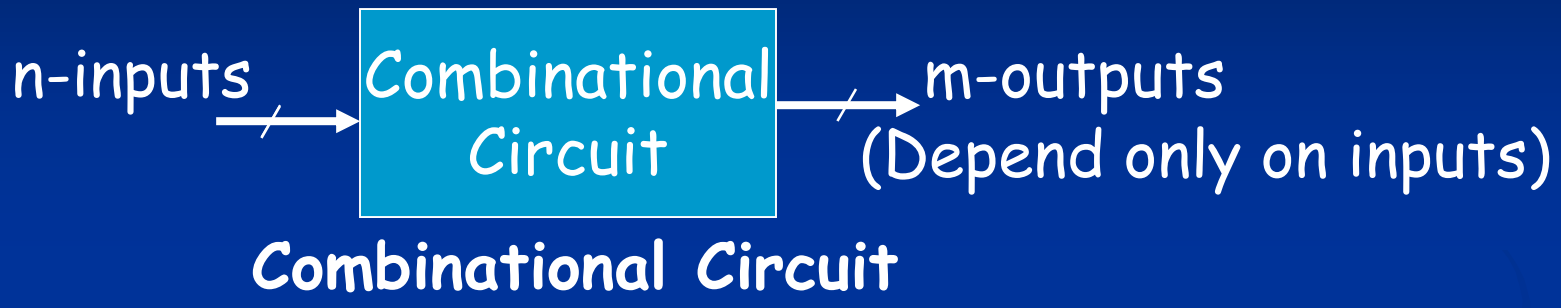
- Hence, a combinational circuit can be described by:
  1. A truth table that lists the output values for each combination of the input variables, or
  2. *m* Boolean functions, one for each output variable.



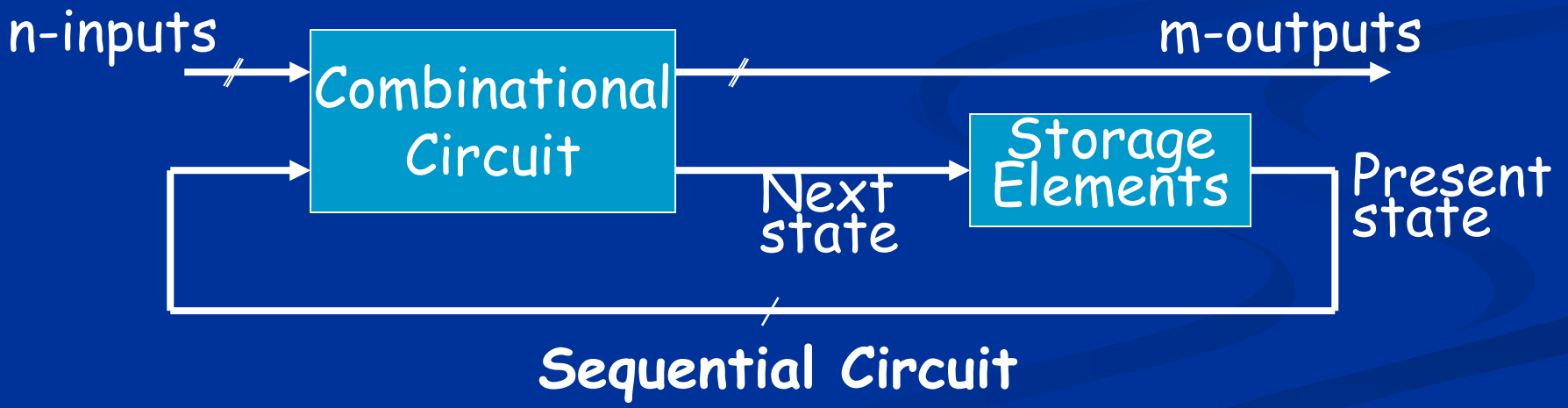
Combinational vs. Sequential Circuits

- Combinational circuits are memory-less. Thus, the output value depends ONLY on the current input values.
- Sequential circuits consist of combinational logic as well as memory elements (used to store certain circuit states). Outputs depend on BOTH current input values and previous input values (kept in the storage elements).

Combinational vs. Sequential Circuits



Combinational Circuit



Sequential Circuit

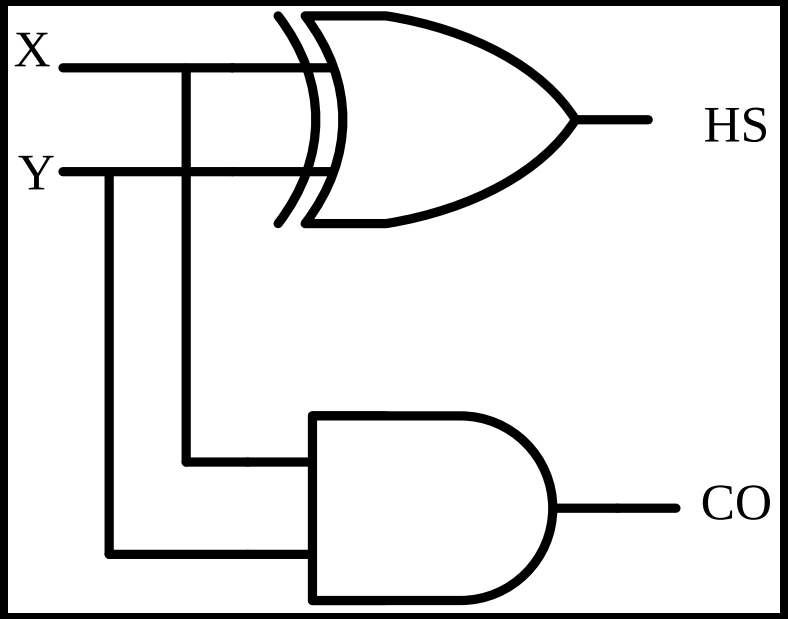
Half Adder

Input is limited to 2 bits

HS = X xor Y

= X•Y' + X'•Y

CO = X•Y



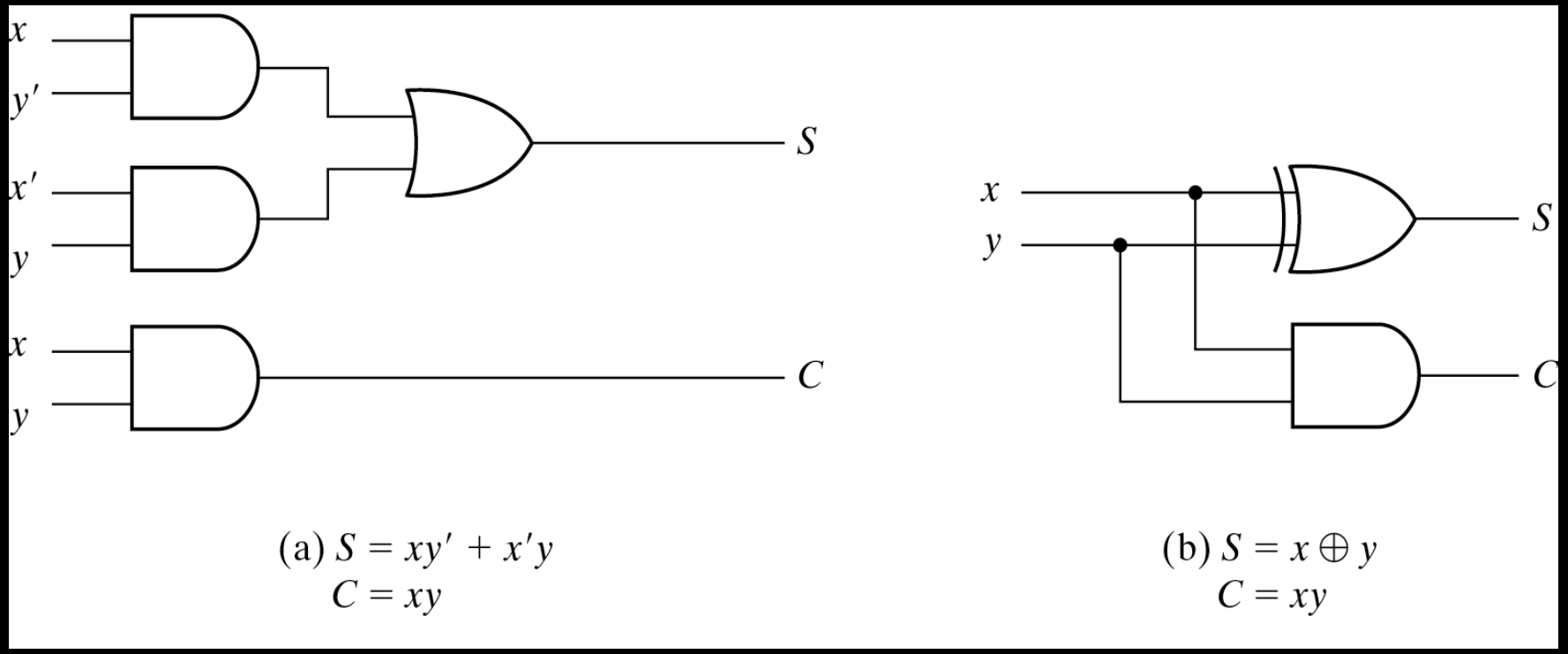
Time for the Half Adder?

1 gate delay

A gate delay of an xor for the half sum

A gate delay of an AND gate for the carry out

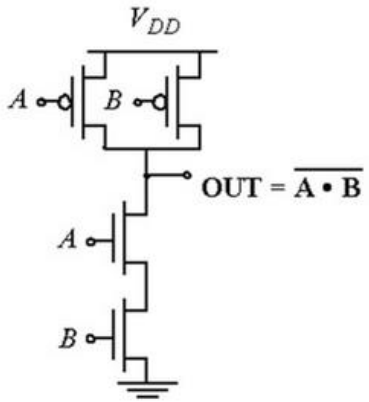
Half Adder



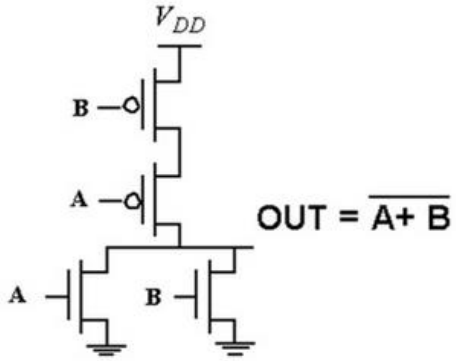
(a) $S = xy' + x'y$
$C = xy$

(b) $S = x \oplus y$
$C = xy$

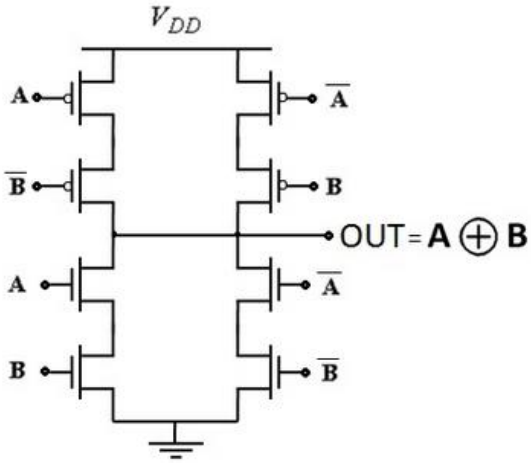
NAND vs. XOR



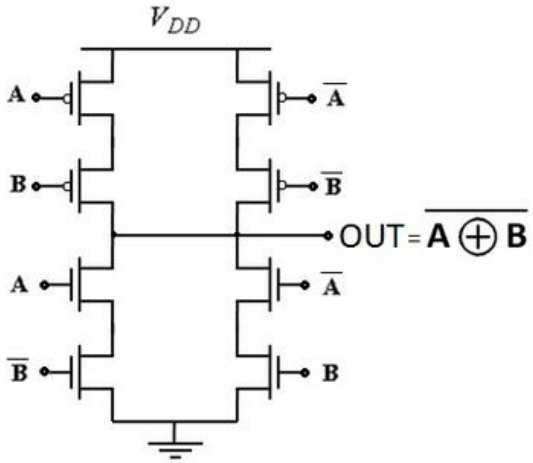
2 input NAND gate (Static Logic)



2 input NOR gate (Static Logic)



2 input XOR gate (Static logic)



2 input XNOR gate (Static logic)

Full Adder

Input is now X Y and the Carry in.

So have

$S = X \bullet Y' \bullet Cin' + X' \bullet Y \bullet Cin' + X' \bullet Y' \bullet Cin + X \bullet Y \bullet Cin$

$= X \text{ xor } Y \text{ xor } Cin$

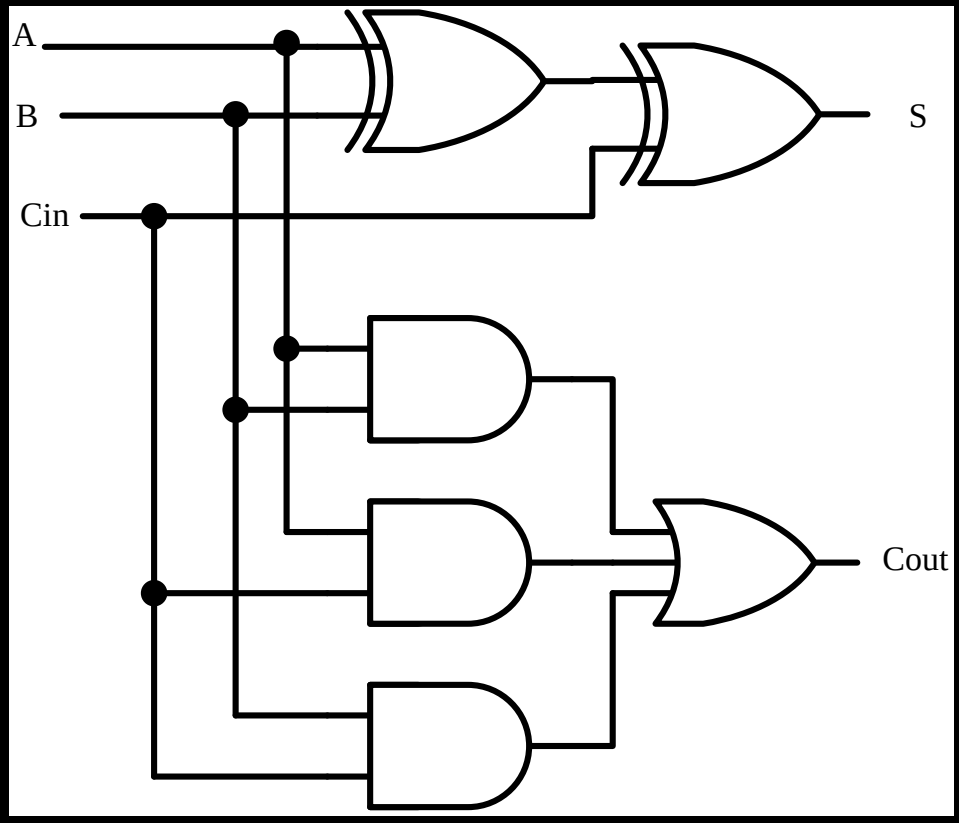
$Cout = X \bullet Y + X \bullet Cin + Y \bullet Cin$

Full Adder

Two level logic

Output delay

- A, B Cin → Sum
- Delay through 2 xor gates
- $t_s = 2\ t_{xor}$
- A, B Cin → Cout
- Delay through an AND gate and an OR gate
- $t_{Cout} = t_{and} + t_{or}$

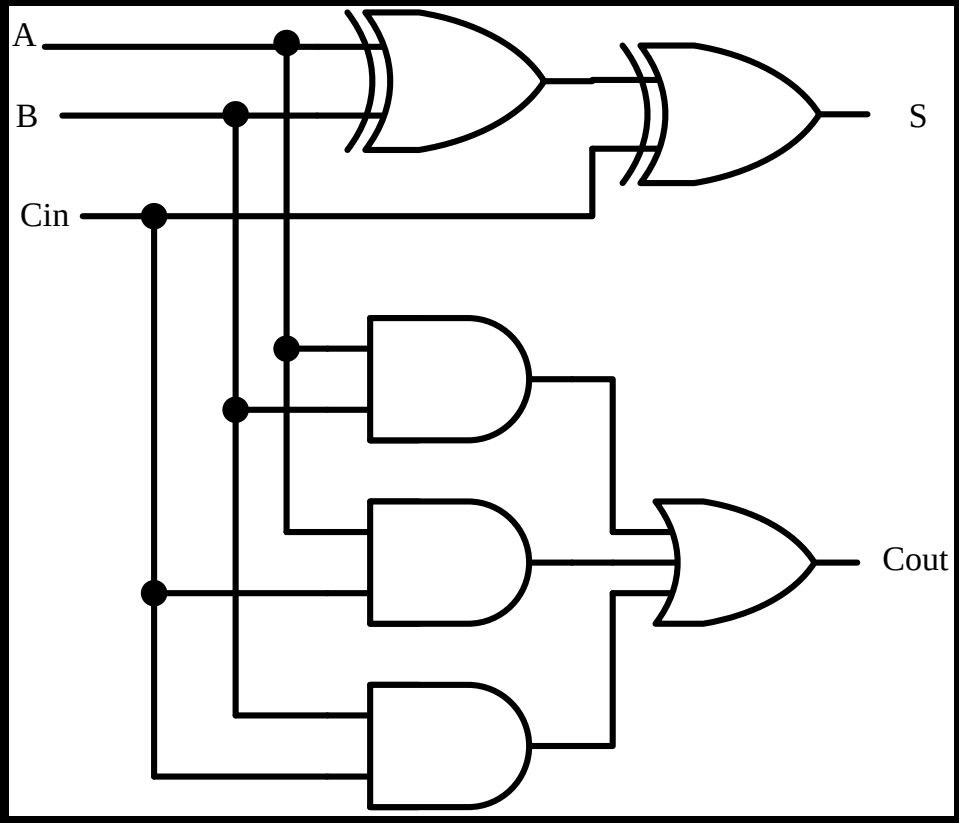


Full Adder

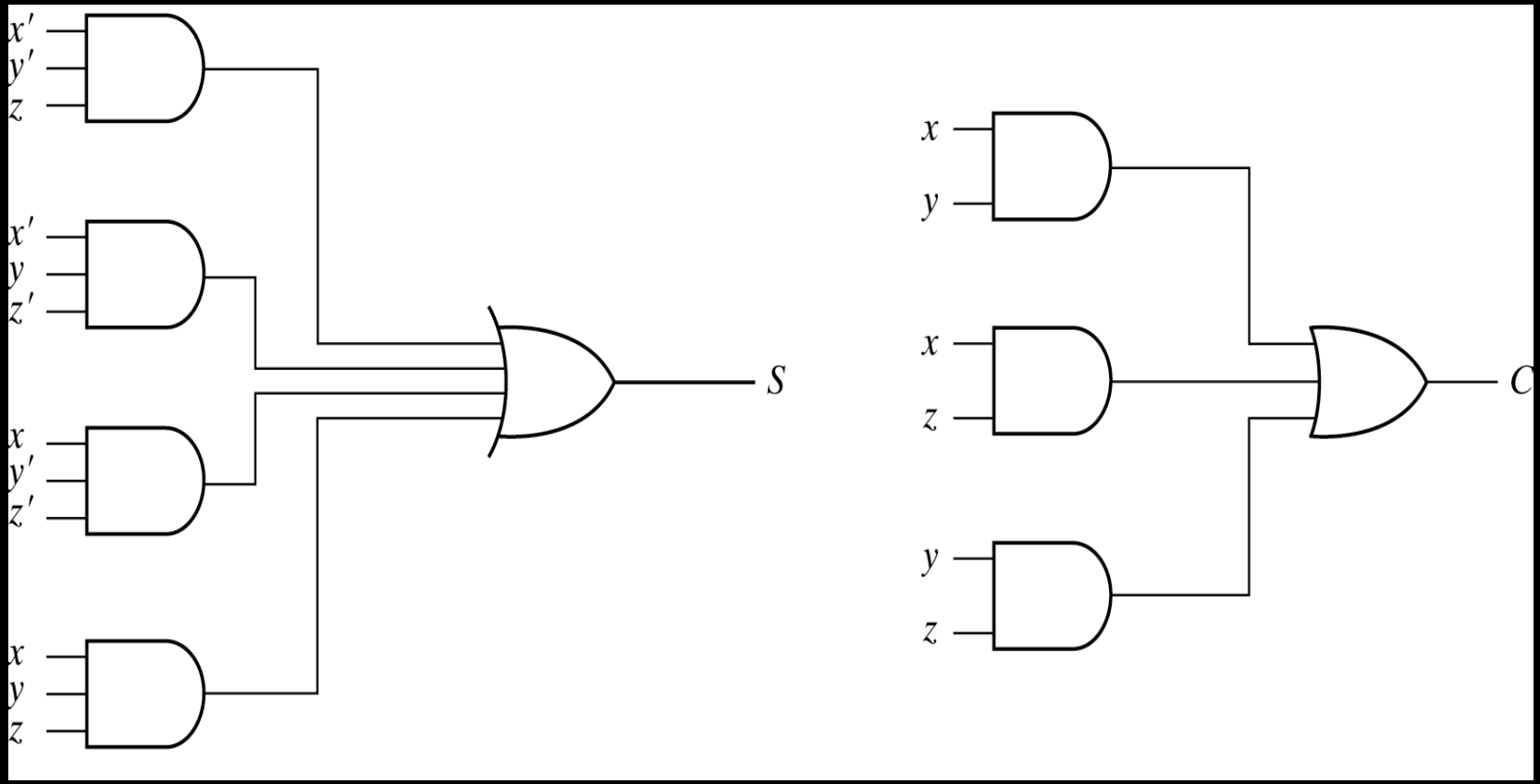
Two level logic

Output delay

- A, B Cin → Sum
- Delay through 2 xor gates
- $t_s = 2\ t_{xor}$
- A, B Cin → Cout
- Delay through an AND gate and an OR gate
- $t_{Cout} = t_{and} + t_{or}$



Full Adder

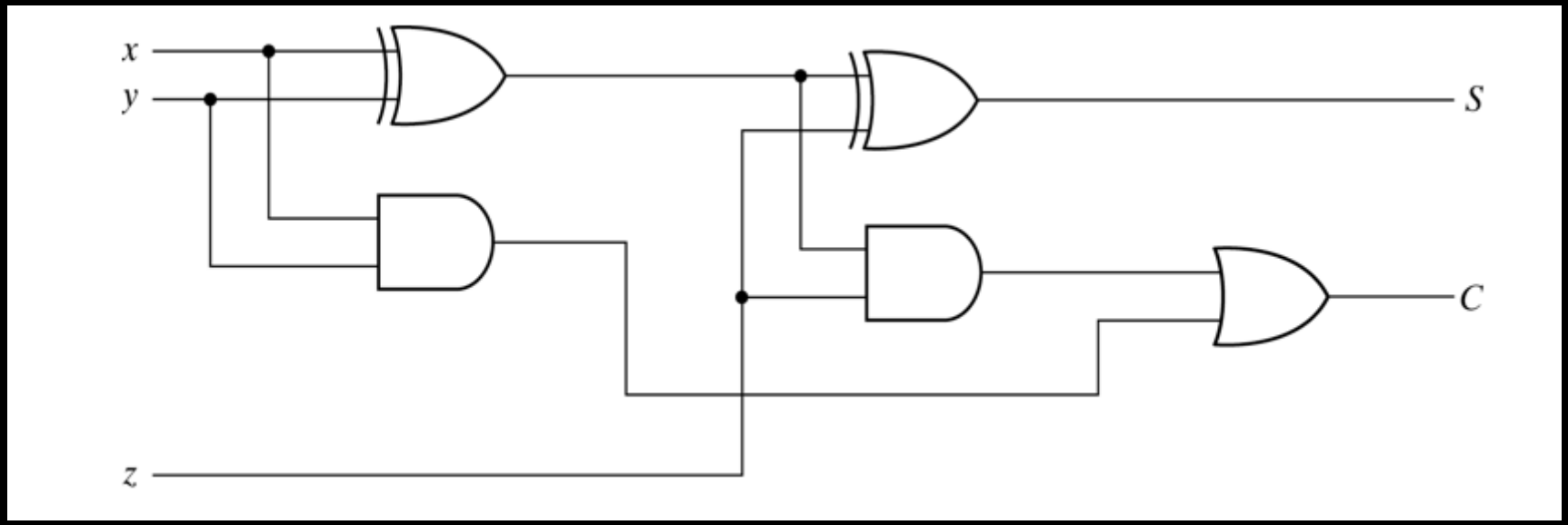


Delay= 2 Gate Delay ?

Full Adder

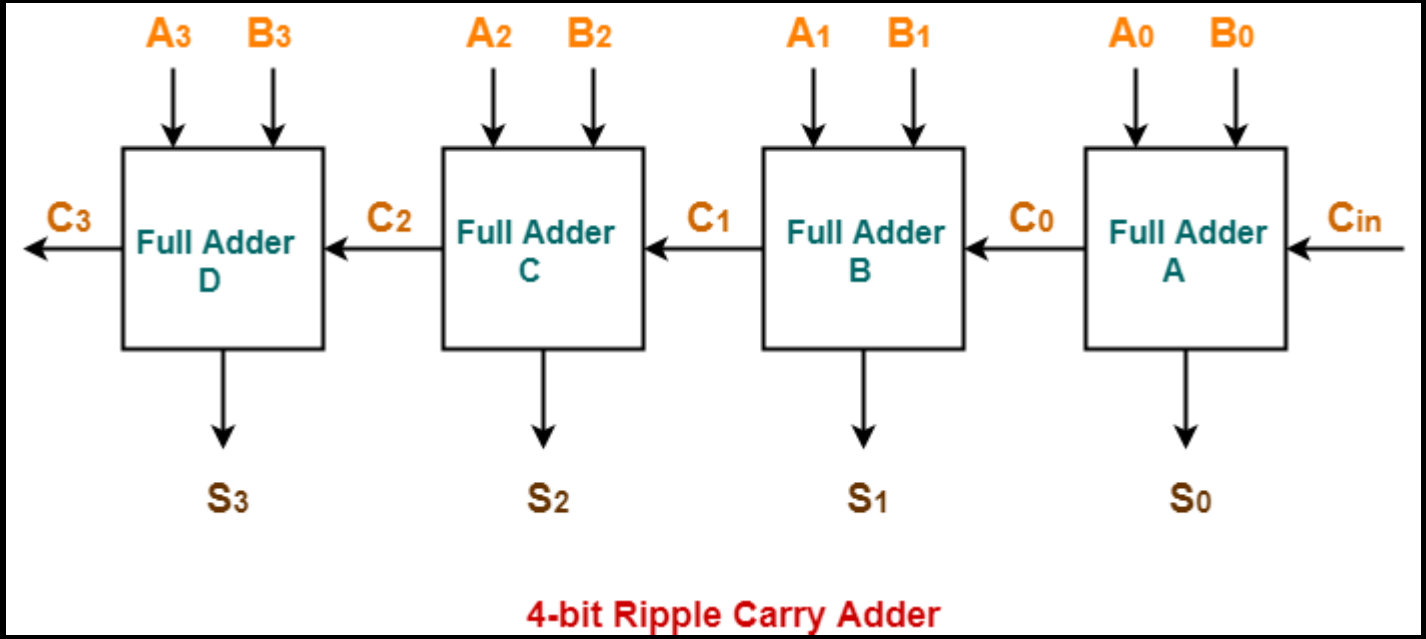
Full-adder can also implemented with two half adders and one OR gate.

$$S = z \oplus (x \oplus y)$$
$$= z'(xy' + x'y) + z(xy' + x'y)'$$
$$= xy'z' + x'yz' + xyz + x'y'z$$
$$C = z(xy' + x'y) + xy = xy'z + x'yz + xy$$



Ripple Carry Adder

This is called Ripple Carry Adder ,because of the construction with full adders are connected in cascade.

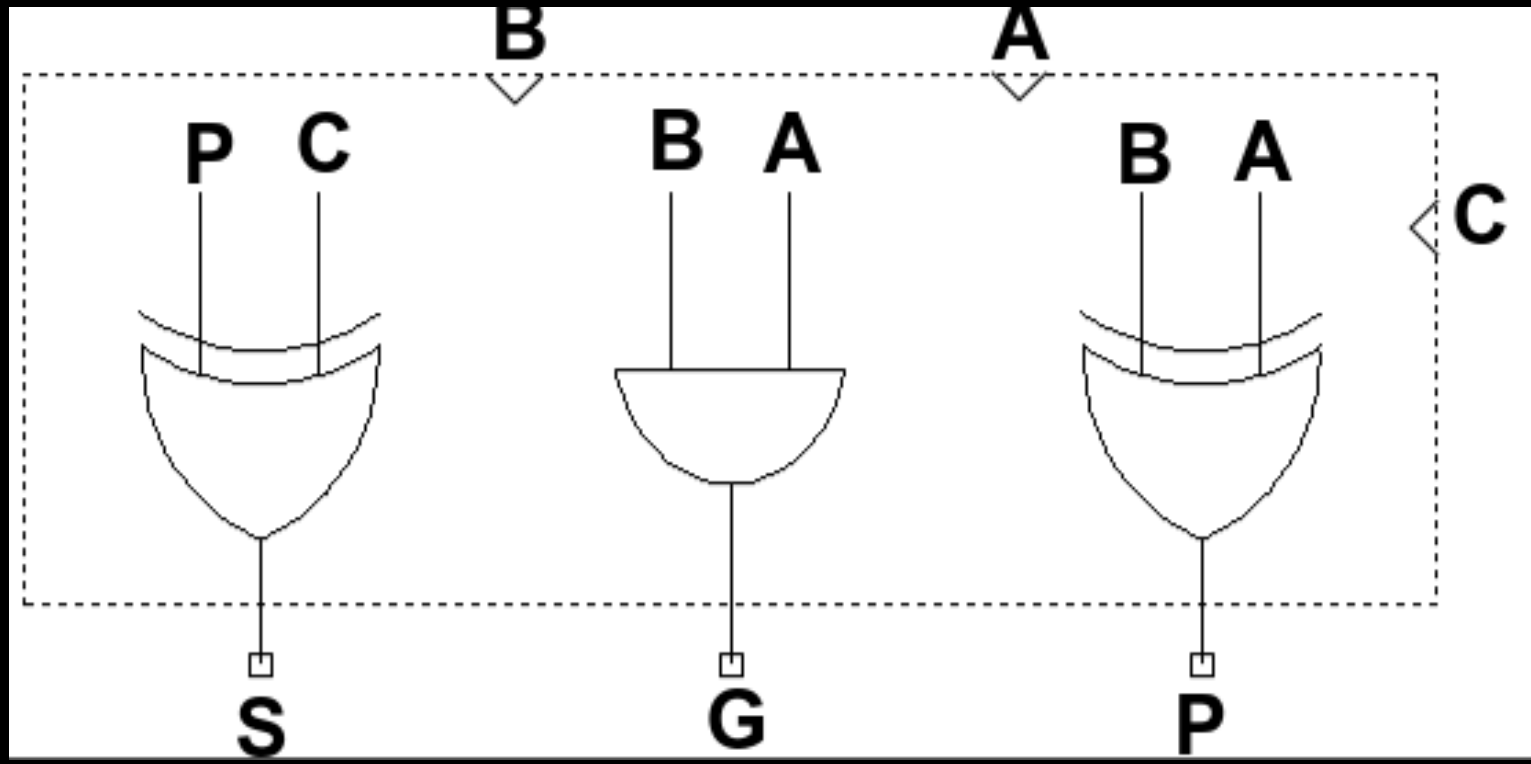


4-bit Ripple Carry Adder

Delay= 4 X Full Adder Delay

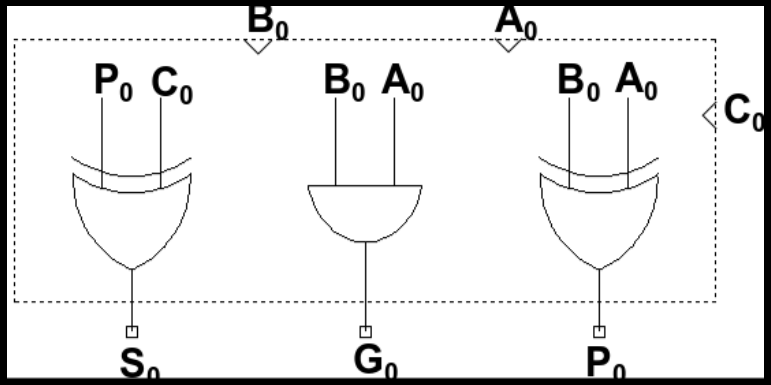
Carry Look-Ahead Adder

1-bit CLA

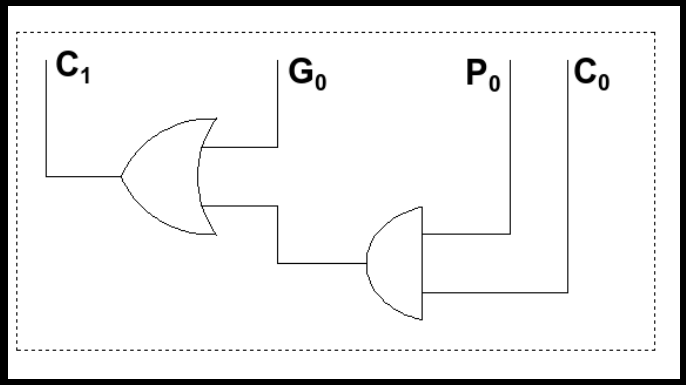


Carry Look-Ahead Adder

CLA

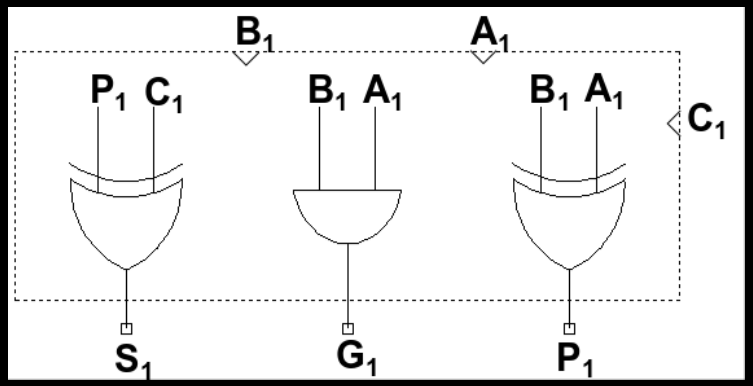


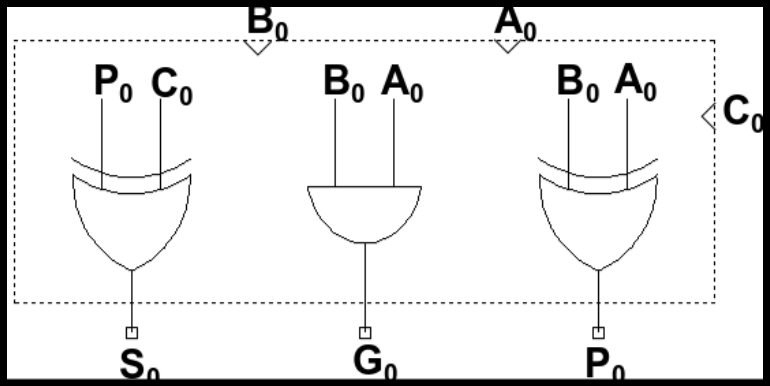
CLLB



$C_1=G_0+P_0C_0$

Carry Look-Ahead Adder

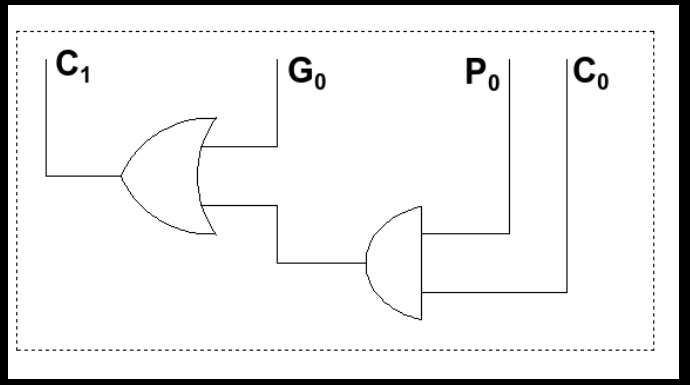




$$C_2=G_1+P_1C_1$$

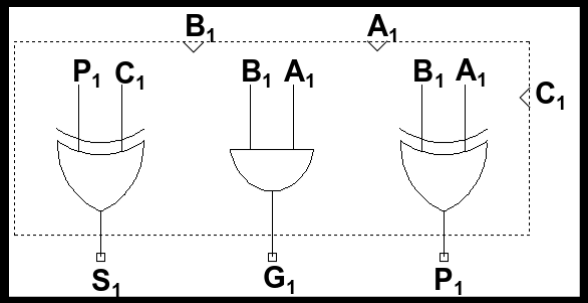
$$C_2=G_1+P_1.(G_0+P_0C_0)$$

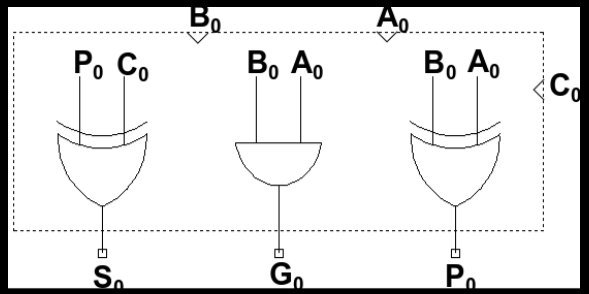
$$C_2=G_1+P_1G_0+P_1P_0C_0$$

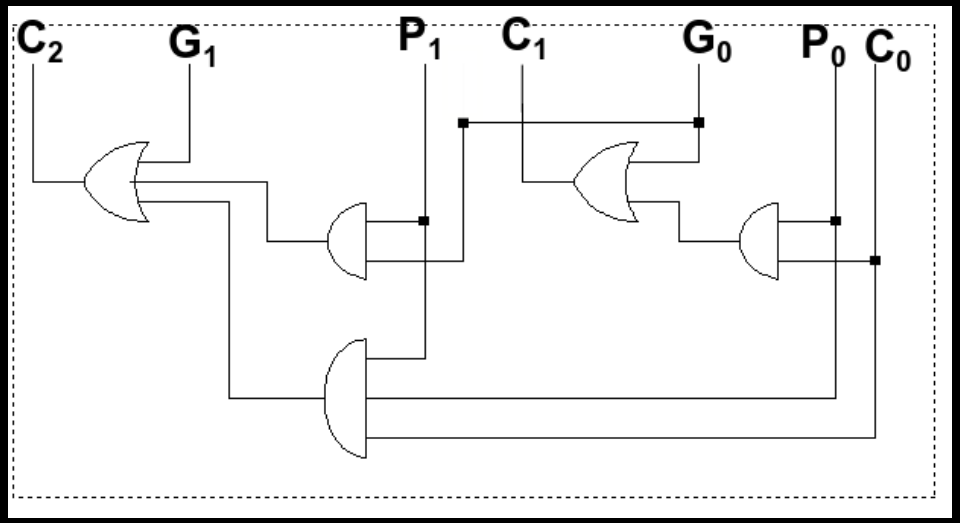


$$C_1=G_0+P_0C_0$$

Carry Look-Ahead Adder







$C_2=G_1+P_1G_0+P_1P_0C_0$

$C_1=G_0+P_0C_0$

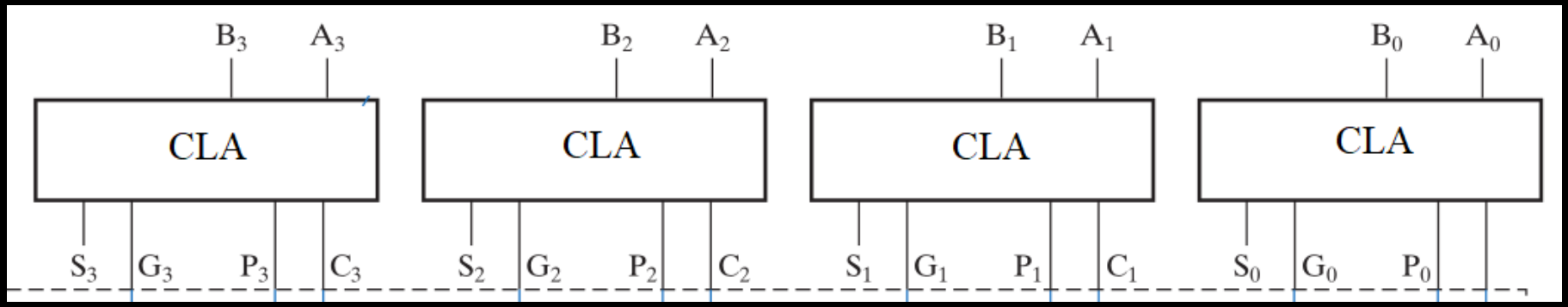
Carry Look-Ahead Adder

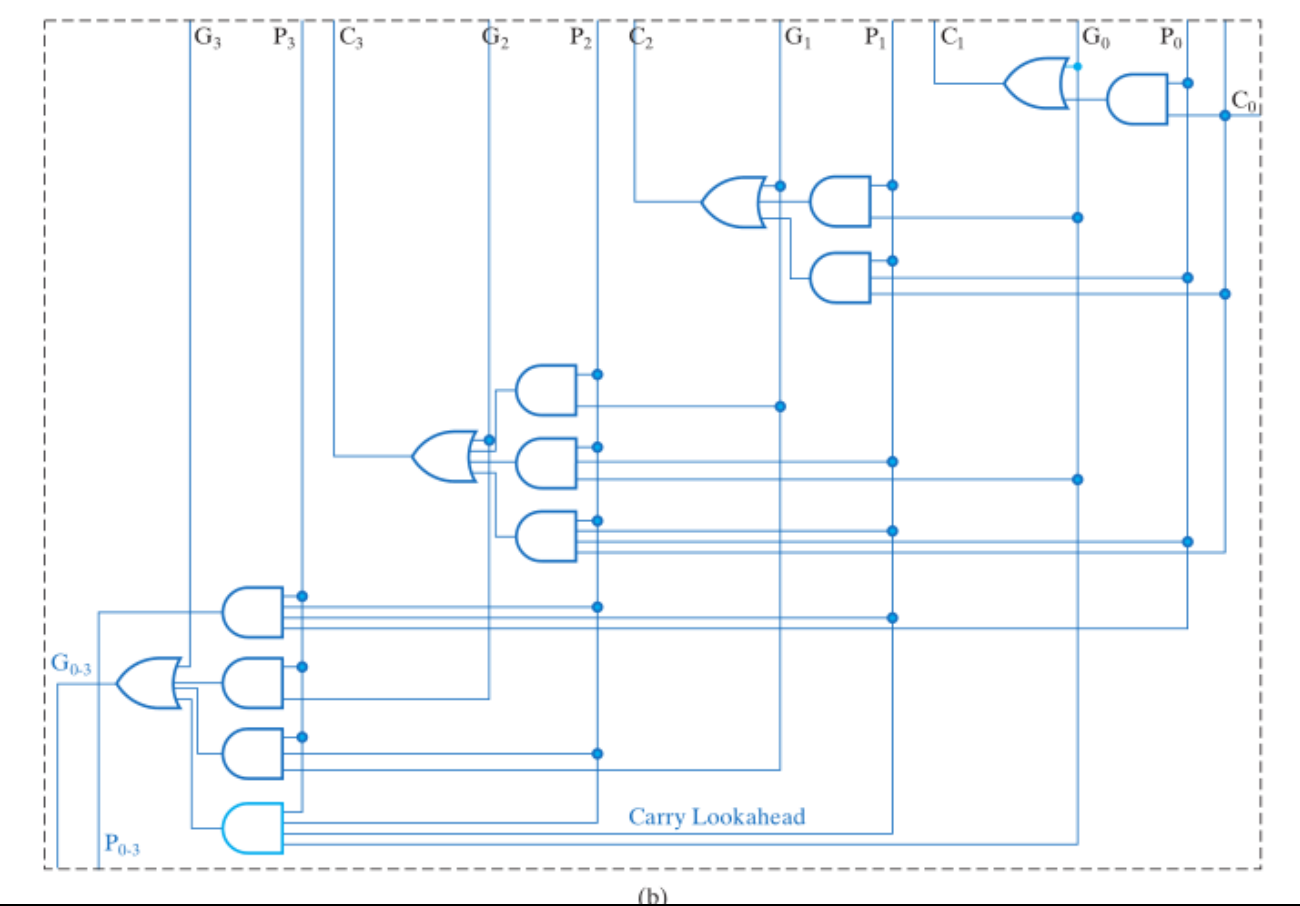
$$C_1=G_0+P_0C_0$$

$$C_2=G_1+P_1G_0+P_1P_0C_0$$

$$C_3=G_2+P_2G_1+ P_2P_1G_0 +P_2P_1P_0C_0$$

$$C_4=G_3+P_3G_2+ P_3P_2G_1 +P_3P_2P_1G_0 +P_3P_2P_1P_0C_0$$


B3
A3
B2
A2
B1
A1
B0
A0
CLA
CLA
CLA
CLA
S3
G3
P3
C3
S2
G2
P2
C2
S1
G1
P1
C1
S0
G0
P0


G3
P3
C3
G2
P2
C2
G1
P1
C1
G0
P0
C0
G0-3
P0-3
Carry Lookahead

(b)