EE1212-

Electronic System I

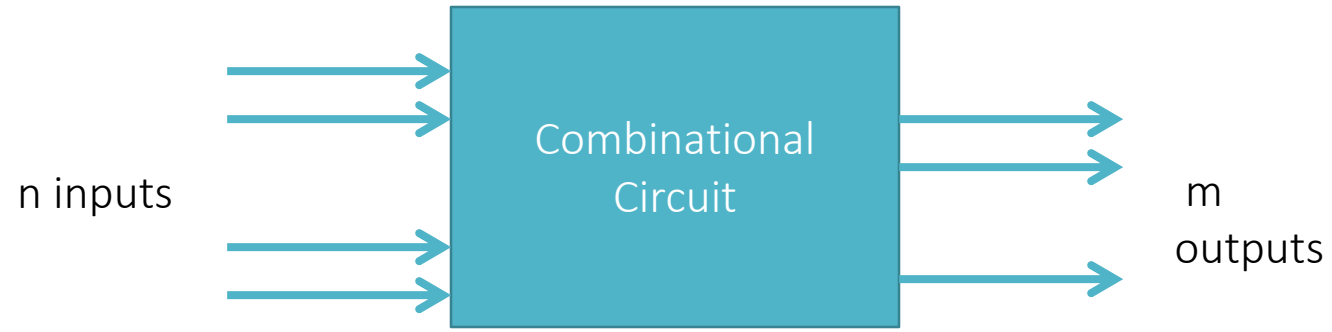
Lecture 1 - Combinational Logic

Capt Geeth Karunarathne

Combinational Circuits

- Outputs at any time are determined from present combination of inputs.
- Performs an operation that can be specified logically by a set of Boolean functions.
- Consists of input variables, logic gates and output variables.
- Transforms binary information from the given input data to a required output data.

Combinational Circuits



- For n input variables 2^n possible input combinations.
- For each possible input combination there is one possible output combination.
- Combinational circuit can be described by truth table or m number of Boolean functions.

Combinational Circuits

Combinational circuits such as Adders, Subtractors, Comparators, Decoders, Encoders and Multiplexers are available in integrated circuits such as MSI (Medium Scale Integration) and VLSI.

Half Adder

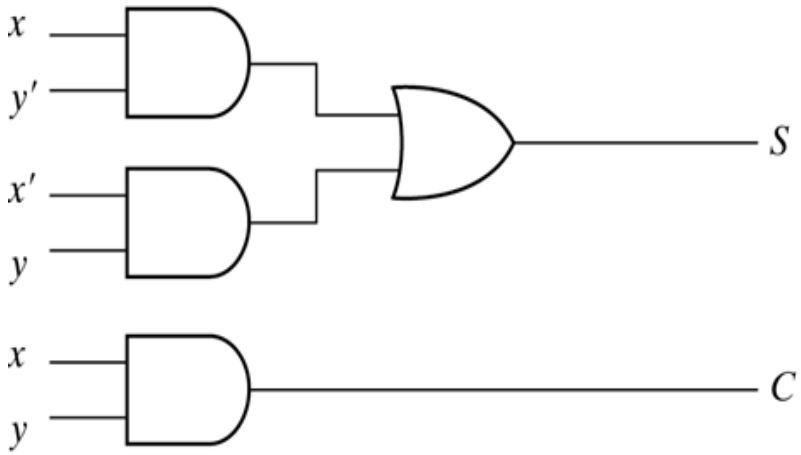
- A combinational circuit that performs the addition of two bits is called a half adder.
  - Adds only single bit nos
  - Does not take carry from previous sum
- The truth table for the half adder is listed below:

x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

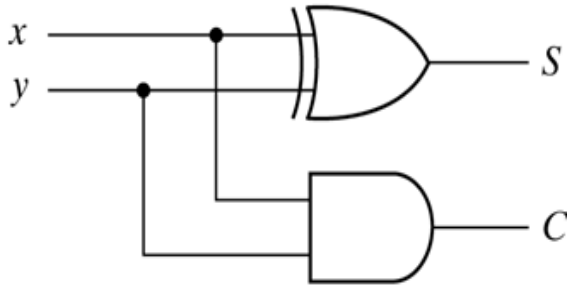
S: Sum
C: Carry

$S = x'y + xy'$
$C = xy$

Implementation of Half-Adder



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



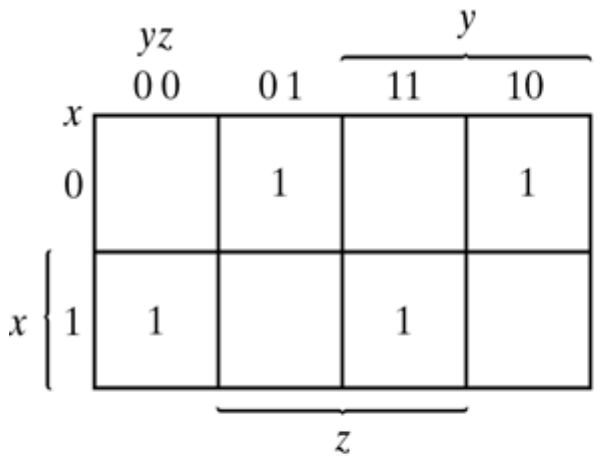
(b) $S = x \oplus y$
$C = xy$

Full-Adder

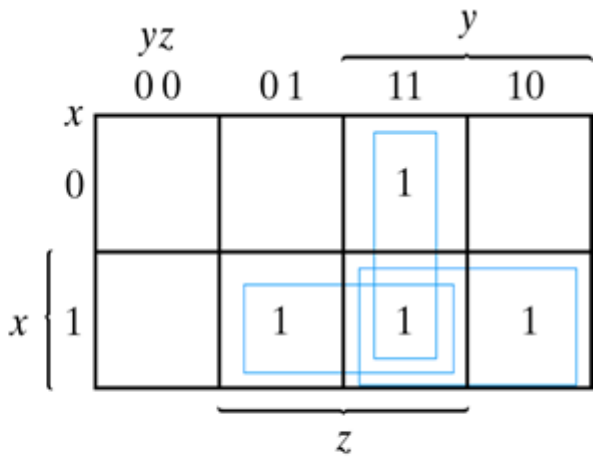
- One that performs the addition of three bits(two significant bits and a previous carry) is a full adder.

x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Simplified Expressions



$S = x'y'z + x'yz' + xy'z' + xyz$

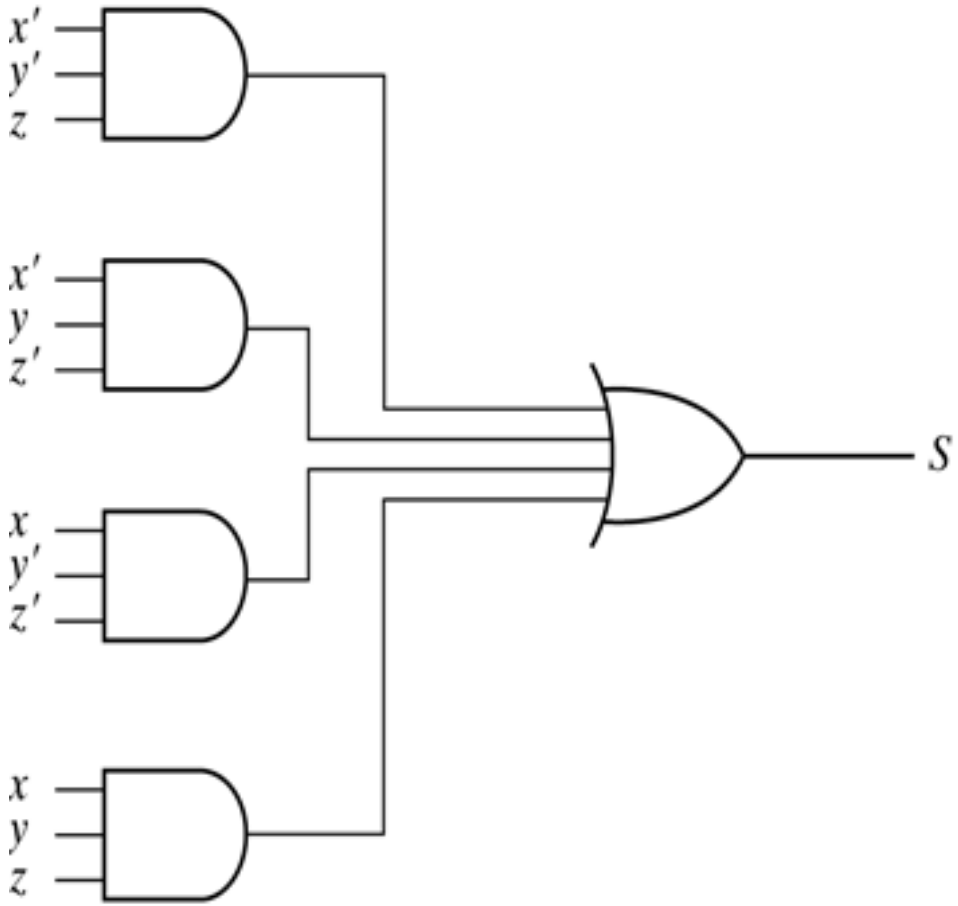


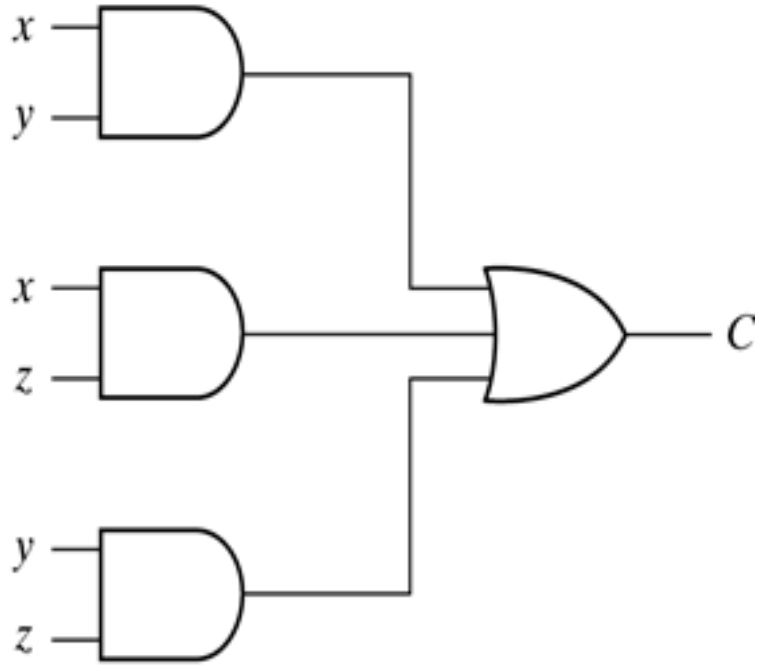
$S = xy + xz + yz$
$= xy + xy'z + x'yz$

S = x'y'z + x'yz' + xy'z' + xyz

C = = xy + xz + yz

Full Adder implementation

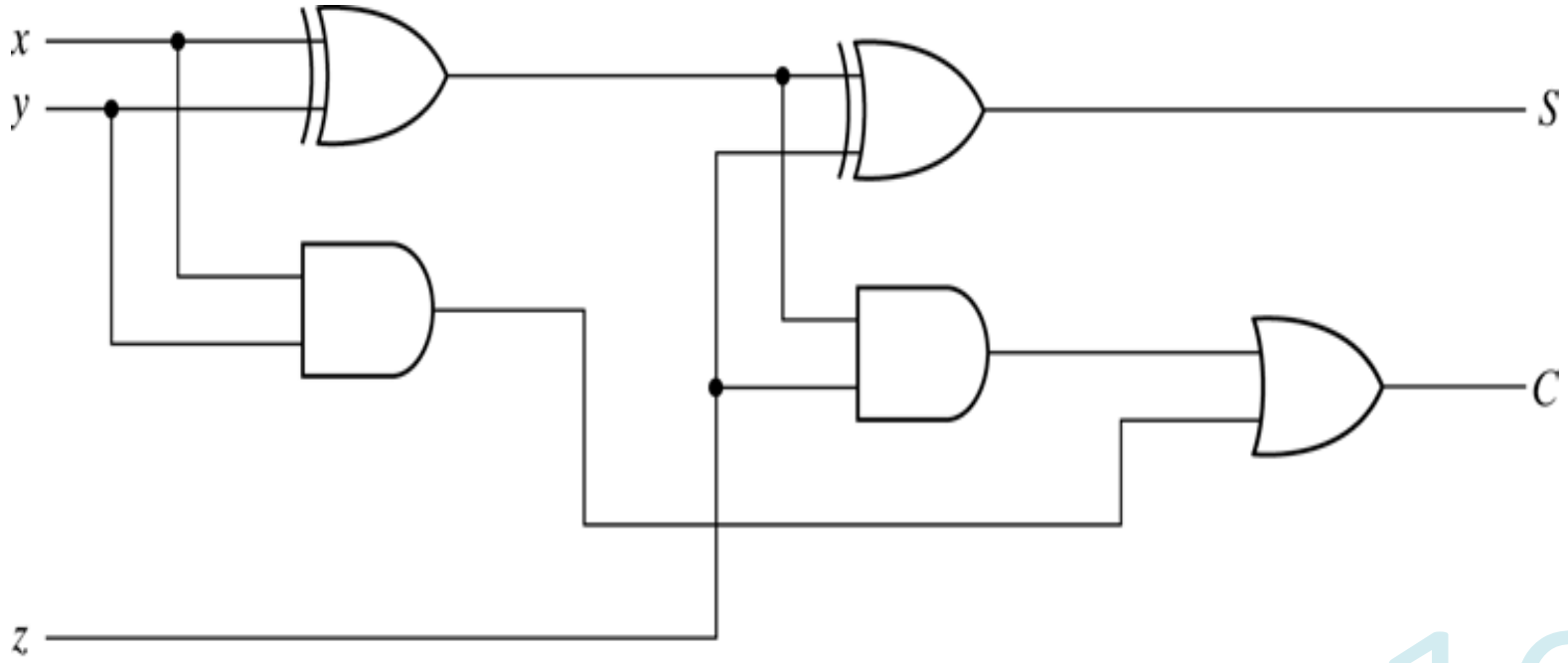




Full-adder can also implement with two half adders and one OR gate (Carry Look-Ahead adder).

$S = x \oplus y \oplus z$

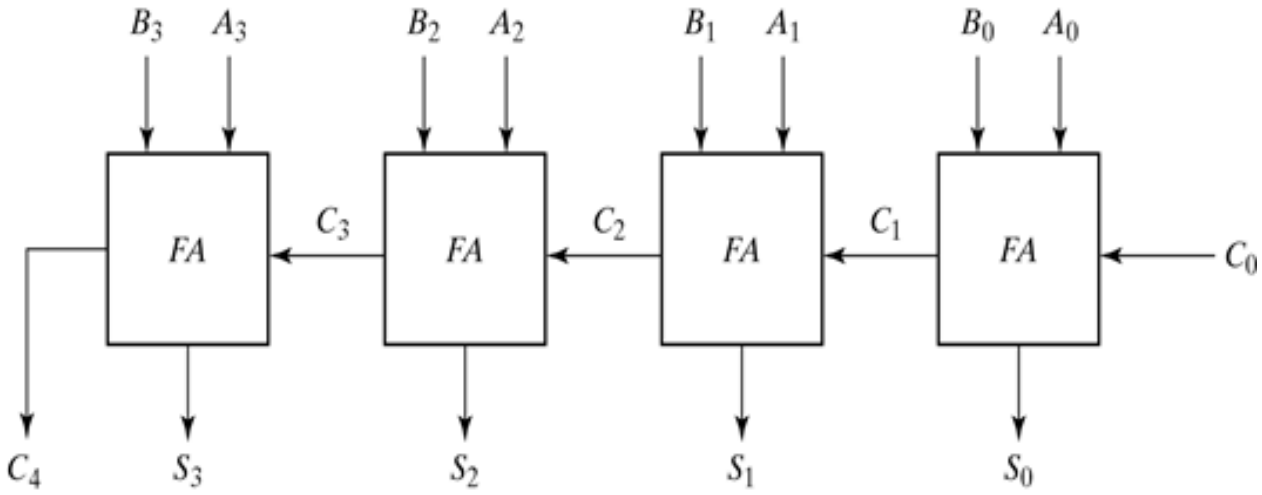
$C = (x \cdot y) + (z \cdot (x \oplus y))$



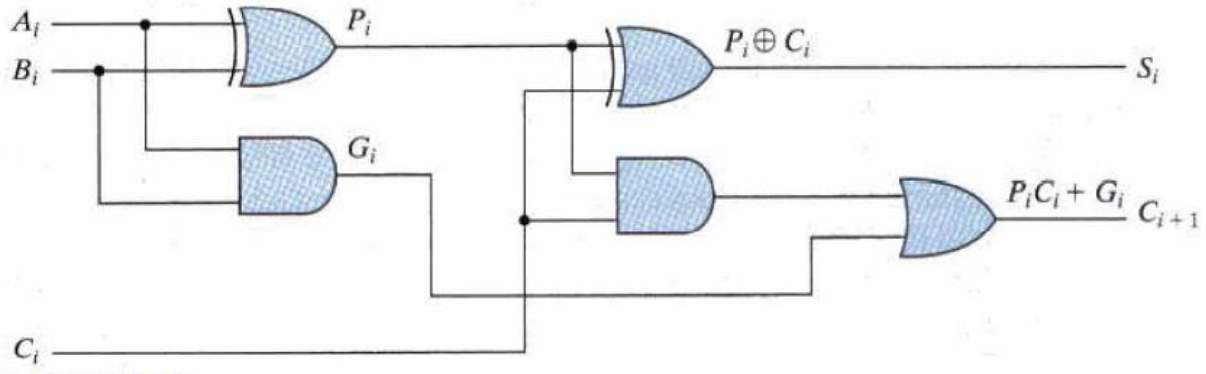
Binary adder

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

Subscript i:	3	2	1	0	
Input carry	0	1	1	0	C_i
Augend	1	0	1	1	A_i
Addend	0	0	1	1	B_i
Sum	1	1	1	0	S_i
Output carry	0	0	1	1	C_{i+1}



Carry Propagation



- The signals must propagate through the gates before the correct output sum is available in the output terminals.
- Total propagation time is equals to:
  - *Propagation delay of a gate x number of gate levels*
- Carry (C_i) propagation requires the highest amount of time in an full adder.
- S_i depends on C_i
- The signal from the input carry C_i to the output carry C_{i+1}, propagates through an AND gate and an OR gate. Which consists of two level.

Carry Propagation

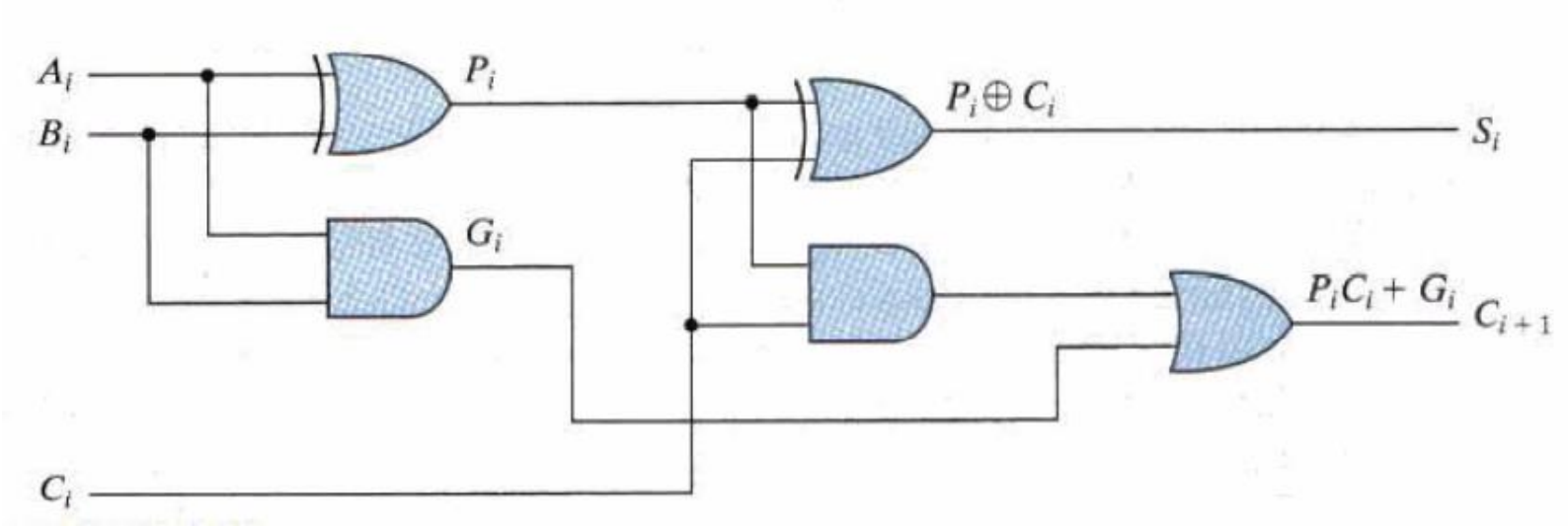
- Assume an adder consists of 4 Full Adders.
- The output carry C_4 would have

  2x4 = 8 gate levels from C_0 to C_4.

- For n-bit adder, there are 2n gate levels for the carry to propagate from input to output.
- Outputs will not be correct unless the signals are given enough time to propagate through the gates connected from inputs to outputs.
- Solutions for reducing carry propagation delay
  a. Faster gates with reduced delay
  b. Increase the circuit complexity – *"Carry Lookahead"*

Carry Lookahead

- Increase the equipment complexity in such way the carry delay time is reduced.
- Example : Carry Lookahead



i – stage in the adder

Carry Lookahead

$$P_i = A_i \oplus B_i$$
$$G_i = A_i B_i$$

Then the output sum and carry can be expressed as follows:

$$S_i = P_i \oplus C_i$$
$$C_{i+1} = G_i + P_i C_i$$

- G_i is called a **carry generate** and it produces a carry of 1 when both A_i & B_i are 1, regardless of the input carry C_i
- P_i is called **carry propagate** because it is the term associated with the propagation of the carry from C_i to C_{i+1}

Carry Lookahead

- Boolean functions for the carry outputs of each stage.

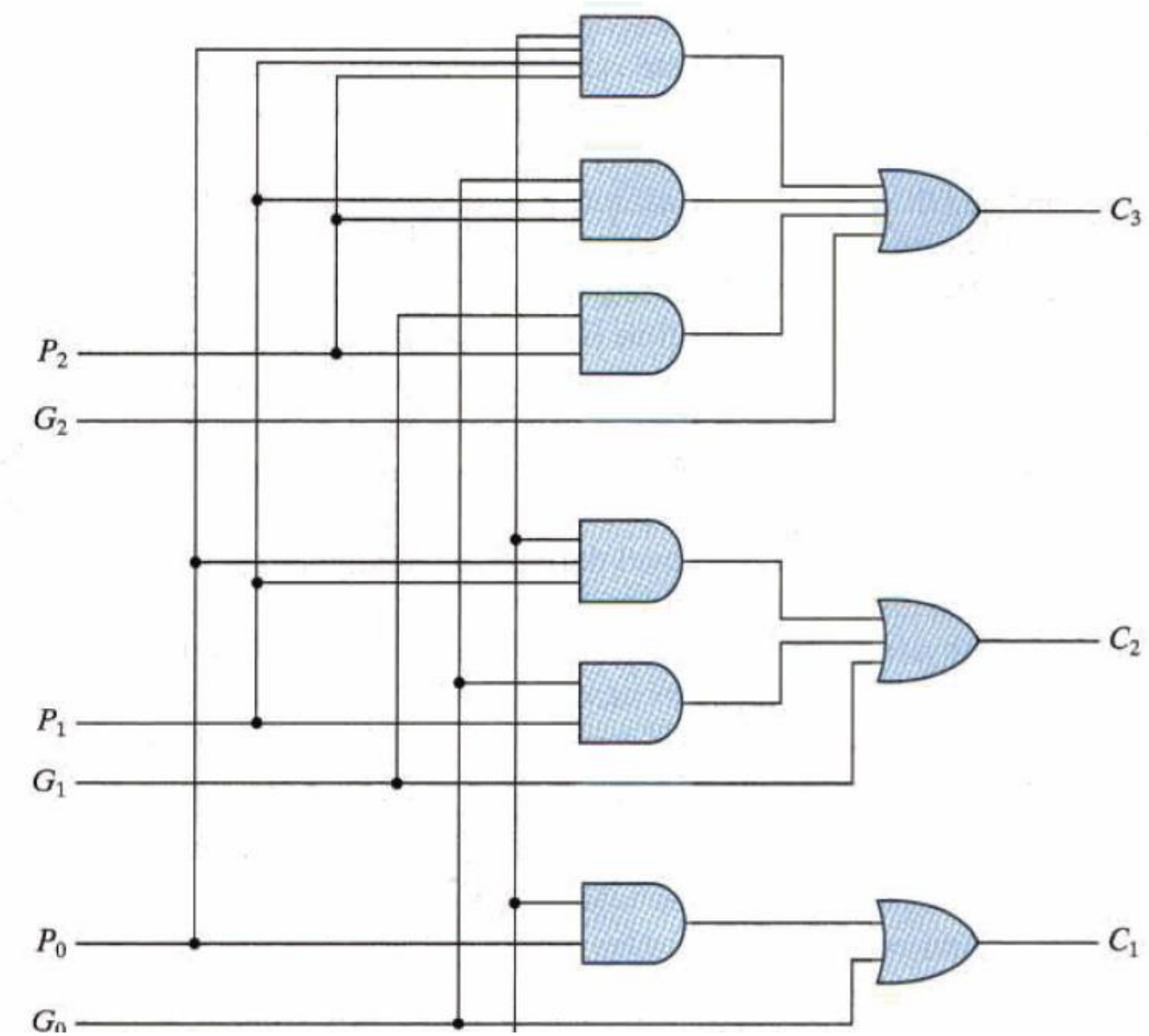
$$C_0 = input\ carry$$

$$C_1 = G_0 + P_0C_0$$

$$C_2 = G_1 + P_1C_1 = G_1 + P_1(G_0 + P_0C_0) = \boldsymbol{G_1 + P_1G_0 + P_1P_0C_0}$$

$$C_3 = G_2 + P_2C_2 = \boldsymbol{G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_0}$$

- Each output carry is expressed in sum of products.
- Each function can be implemented with one level of AND gates followed by an OR gate.



Carry Lookahead Generator

4-bit Adder with Carry Lookahead