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

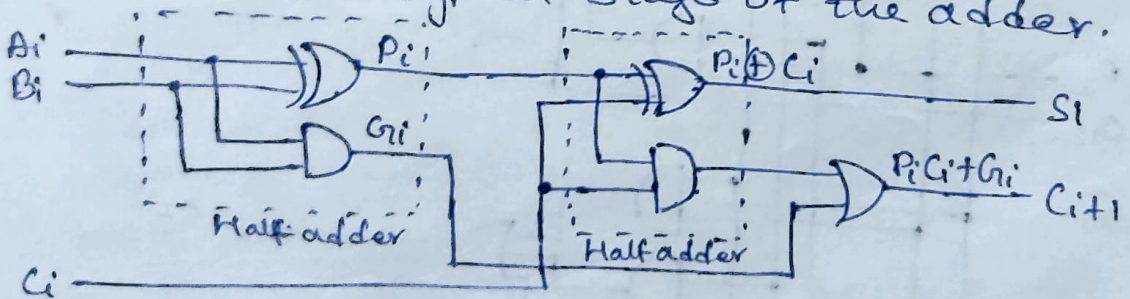
→ In any combinational circuit, the signal must propagate through the gates before the correct output is available in the output terminals.

→ The total propagation time is equal to the propagation delay of a typical gate, times the number of gate levels in the circuit.

→ The longest propagation delay time in adder is the time it takes the carry to propagate through the full adders.

→ Since each bit of the sum output depends on the value of the input carry, the value of S_i at any given stage in the adder will be in it's steady-state final value only after the input carry to that stage has been propagated.

→ The input and output variables use the subscript i to denote a typical stage of the adder.



→ The signals at P_i and G_i settle to their steady-state values after they propagate through their respective gates.

→ These two signals are common to all half adders and depend on only the input augend and addend bits.

→ 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 constitute two gate levels.

→ For an n-bit adder, there are 2n gate levels for the carry to propogate from input to output.

→ The most widely used technique for reducing the carry propagation time in a parallel adder employs the principle of carry lookahead logic.

$$P_i = A_i \oplus B_i$$

$$G_i = A_i B_i$$

$$S_i = P_i \oplus C_i$$

$$C_{i+1} = G_i + P_i C_i$$

→ G_i is called a carry generate, and it produces a carry of 1 when both A_i and B_i are 1, regardless of the input carry C_i.

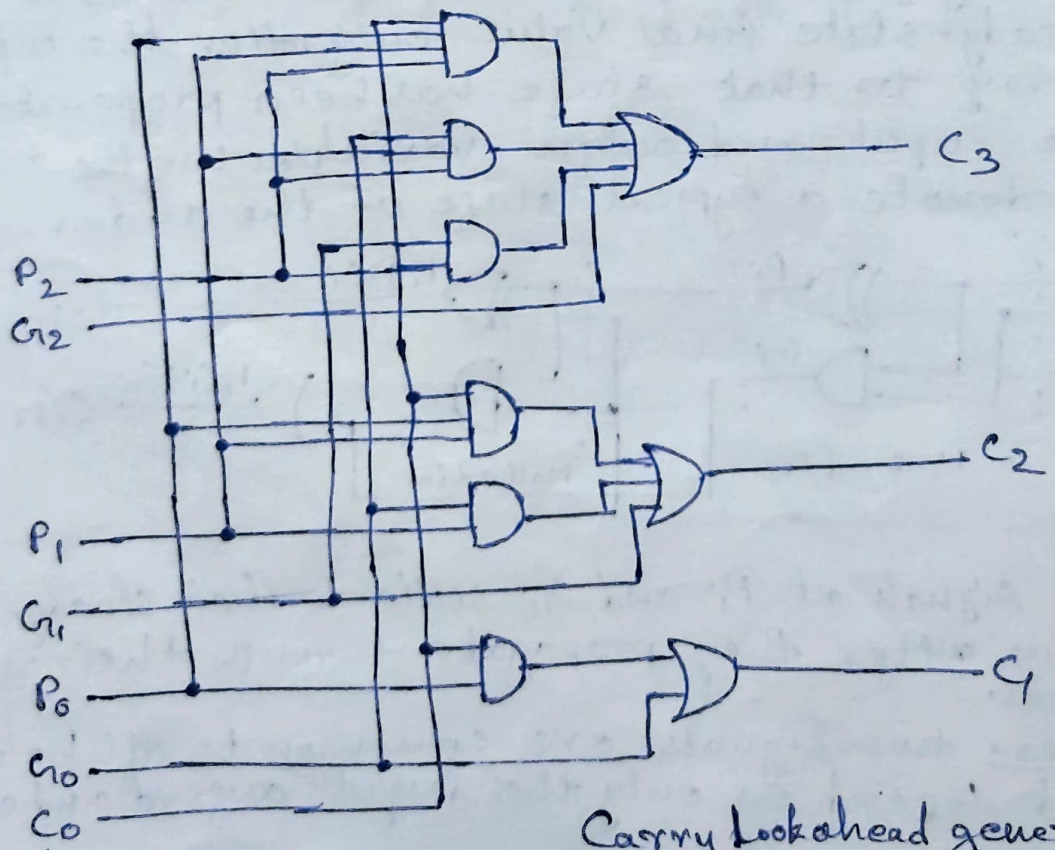
→ P_i is called a carry propagate, because it determines whether a carry into stage i will propogate into stage i+1.

$$C_0 = \text{input carry}$$

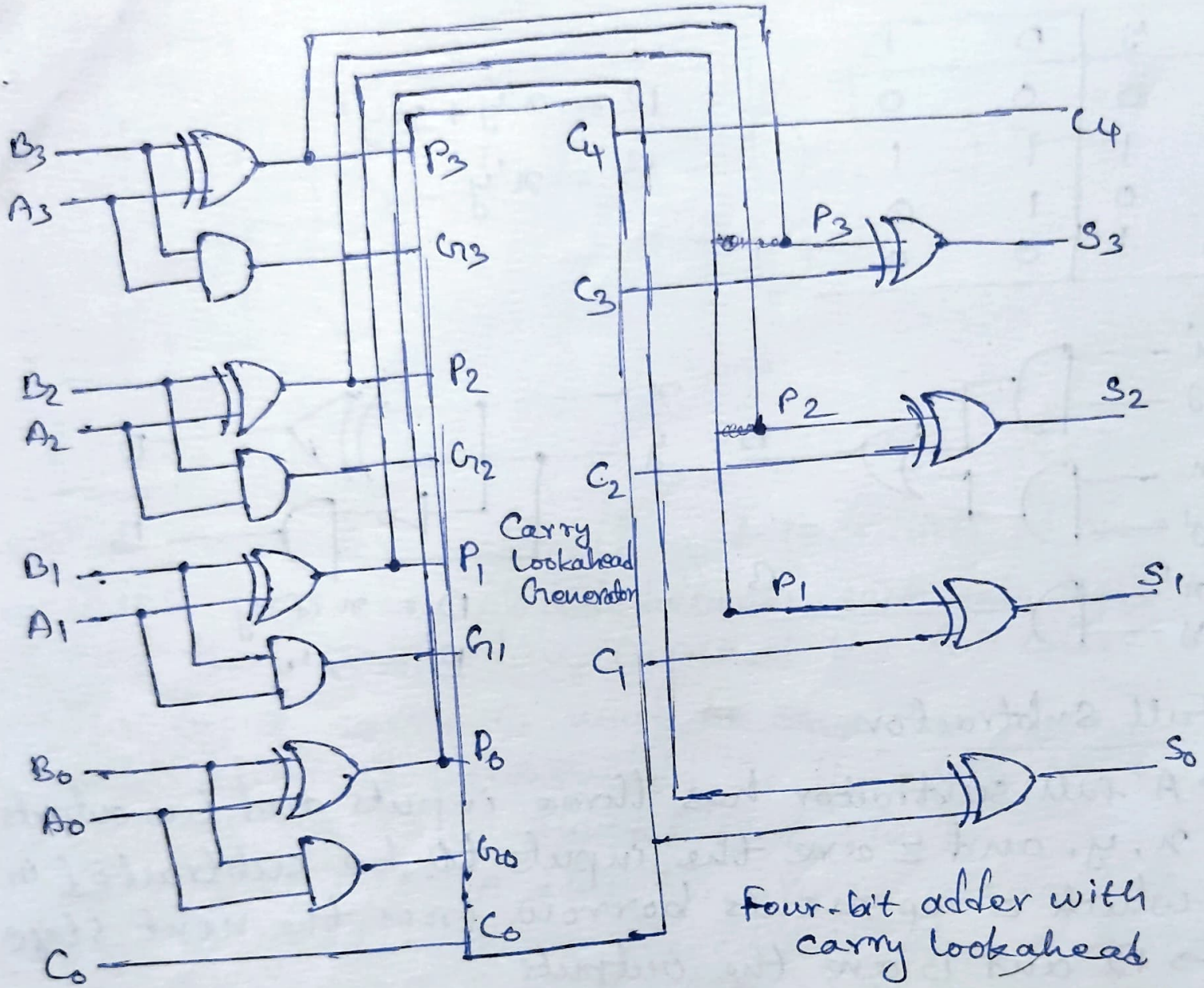
$$C_1 = G_0 + P_0 C_0$$

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

$$C_3 = G_2 + P_2 C_2 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$$



Carry Lookahead generator



Four-bit adder with carry lookaheads

→ Each sum output requires two exclusive-OR gates.

→ The output of the first exclusive-OR gate generates the P_i variable, and the AND gate generates the G_i variable.

→ The carries are propagated through the carry lookahead generator and applied as inputs to the second exclusive-OR gate.

→ All output carries are generated after a delay through two levels of gates. Thus, outputs have equal propagation delay times.