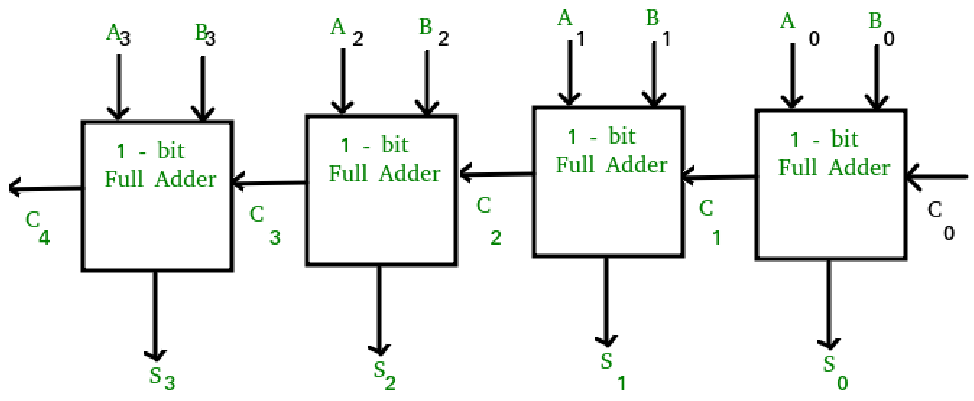
Digital logic | Carry Look-Ahead Adder

Motivation behind Carry Look-Ahead Adder :

In ripple carry adders, for each adder block, the two bits that are to be added are available instantly. However, each adder block waits for the carry to arrive from its previous block. So, it is not possible to generate the sum and carry of any block until the input carry is known. The i^{th} block waits for the $i-1^{th}$ block to produce its carry. So there will be a considerable time delay which is carry propagation delay.

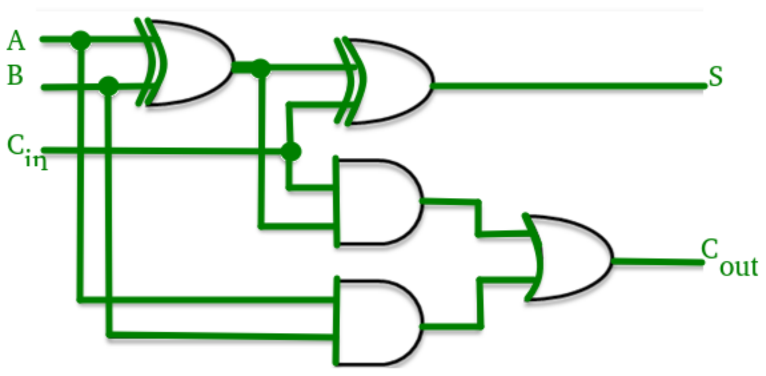


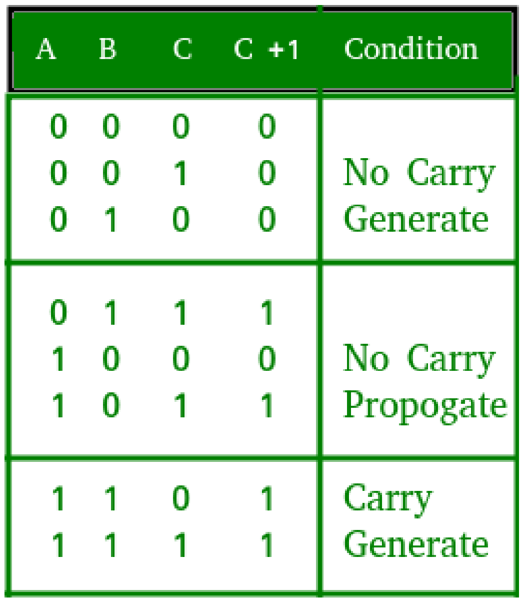
Consider the above 4-bit ripple carry adder. The sum S_4 is produced by the corresponding full adder as soon as the input signals are applied to it. But the carry input C_4 is not available on its final steady state value until carry C_3 is available at its steady state value. Similarly C_3 depends on C_2 and C_2 on C_1. Therefore, though the carry must propagate to all the stages in order that output S_3 and carry C_4 settle their final steady-state value.

The propagation time is equal to the propagation delay of each adder block, multiplied by the number of adder blocks in the circuit. For example, if each full adder stage has a propagation delay of 20 nanoseconds, then S_3 will reach its final correct value after 60 (20 × 3) nanoseconds. The situation gets worse, if we extend the number of stages for adding more number of bits.

Carry Look-ahead Adder :

A carry look-ahead adder reduces the propagation delay by introducing more complex hardware. In this design, the ripple carry design is suitably transformed such that the carry logic over fixed groups of bits of the adder is reduced to two-level logic. Let us discuss the design in detail.





A	B	C	C +1	Condition
0	0	0	0	No Carry Generate
0	0	1	0	
0	1	0	0	
0	1	1	1	No Carry Propogate
1	0	0	0	
1	0	1	1	
1	1	0	1	Carry Generate
1	1	1	1	

Consider the full adder circuit shown above with corresponding truth table. We define two variables as **'carry generate'** G_i and **'carry propagate'** P_i then,

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

The sum output and carry output can be expressed in terms of carry generate G_i and carry propagate P_i as

$$S_i = P_i \oplus C_i$$
$$C_i + 1 = G_i + P_i C_i$$

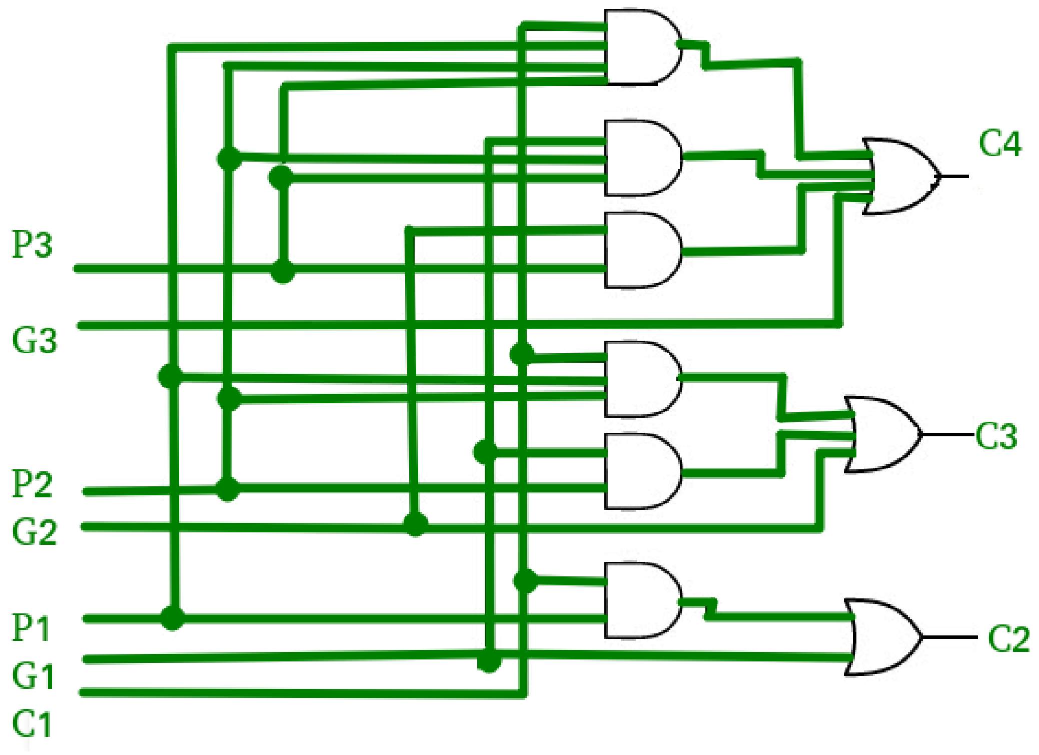
where G_i produces the carry when both A_i, B_i are 1 regardless of the input carry. P_i is associated with the propagation of carry from C_i to $C_i + 1$.

The carry output Boolean function of each stage in a 4 stage carry look-ahead adder can be expressed as

$$C_1 = G_0 + P_0 C_{in}$$
$$C_2 = G_1 + P_1 C_1 = G_1 + P_1 G_0 + P_1 P_0 C_{in}$$
$$C_3 = G_2 + P_2 C_2 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_{in}$$
$$C_4 = G_3 + P_3 C_3 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_{in}$$

From the above Boolean equations we can observe that C_4 does not have to wait for C_3 and C_2 to propagate but actually C_4 is propagated at the same time as C_3 and C_2. Since the Boolean expression for each carry output is the sum of products so these can be implemented with one level of AND gates followed by an OR gate.

The implementation of three Boolean functions for each carry output (C_2, C_3 and C_4) for a carry look-ahead carry generator shown in below figure.



Time Complexity Analysis :

We could think of a carry look-ahead adder as made up of two “parts”

1. The part that computes the carry for each bit.
2. The part that adds the input bits and the carry for each bit position.

The $log(n)$ complexity arises from the part that generates the carry, not the circuit that adds the bits.

Now, for the generation of the n^{th} carry bit, we need to perform a AND between (n+1) inputs. The complexity of the adder comes down to how we perform this AND operation. If we have AND gates, each with a fan-in (number of inputs accepted) of k, then we can find the AND of all the bits in $log_k(n+1)$ time. This is represented in asymptotic notation as $\Theta(logn)$.

Advantages and Disadvantages of Carry Look-Ahead Adder :

Advantages –

- The propagation delay is reduced.
- It provides the fastest addition logic.

Disadvantages –

- The Carry Look-ahead adder circuit gets complicated as the number of variables increase.
- The circuit is costlier as it involves more number of hardware.

GATE CS Corner Questions

Practicing the following questions will help you test your knowledge. All questions have been asked in GATE in previous years or in GATE Mock Tests. It is highly recommended that you practice them.

1. GATE CS 2016 (Set-1), Question 43
2. GATE CS 2004, Question 90
3. GATE CS 2007, Question 85
4. GATE CS 2006, Question 85
5. GATE CS 1997, Question 15

References –

iitkgp.virtual-labs

Carry-lookahead adder – Wikipedia

Samujjal Das

Check out this Author's contributed articles.

Digital Electronics & Logic Design GATE CS Login to Improve this Article

Recommended Posts:

Mathematics | Graph theory practice questions
Operating System | Process Table and Process Control Block (PCB)
Data Communication | Transmission Impairment
Digital logic | Master Slave JK Flip Flop
Digital Logic | Parallel Adder & Parallel Subtractor
Digital Logic | Implicants in K-Map
Non-Restoring Division For Unsigned Integer
Restoring Division Algorithm For Unsigned Integer
Amortized analysis for increment in counter
Digital Electronics | Operational Amplifier (op-amp)

(Login to Rate)

0 Average Difficulty : **0/5.0**
No votes yet.

Basic Easy Medium Hard Expert

Add to TODO List
Mark as DONE

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments Share this post!

▲