EE2003 : Computer Organisation

**Tutorial 1**

EE15B025 | Ganga Meghanath

horizontal line

# 

# ripple_carry_adder.jpg

# 

# Question

Show that the logic expression cₙ⊕cₙ₋₁ is a correct indicator of overflow in the addition of 2’s complement integers, by using an appropriate truth table

## Answer

The (n-1)th bit is the ‘sign’ bit. It denotes the sign of the numbers to be added. If the (n-1)th bit is 0, then the number is positive and if it’s 1, then the number is negative. Overflow during addition happens while adding either two positive numbers or while adding two negative numbers. We can figure out whether there is an overflow by checking if the (n-1)th bit of the resultant sum, ie, sₙ₋₁ is of the opposite sign to that of xₙ₋₁ and yₙ₋₁. The truth table for the same has been depicted below.

## Truth Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Cases | yₙ₋₁ | xₙ₋₁ | cₙ₋₁ | cₙ | sₙ₋₁ | **cₙ⊕cₙ₋₁** |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 | 1 | 1 |
| 3 | 0 | 1 | 0 | 0 | 1 | 0 |
| 4 | 0 | 1 | 1 | 1 | 0 | 0 |
| 5 | 1 | 0 | 0 | 0 | 1 | 0 |
| 6 | 1 | 0 | 1 | 1 | 0 | 0 |
| 7 | 1 | 1 | 0 | 1 | 0 | 1 |
| 8 | 1 | 1 | 1 | 1 | 1 | 0 |

As we can see from the truth table, when the output sign bit is different from that of the inputs, cₙ⊕cₙ₋₁ becomes 1. Hence, we can check for overflow using cₙ⊕cₙ₋₁.