ASSIGNMENT 3 REPORT

Carry Look-ahead Adder (4 and 16 bit)

**Jatin Gupta (20CS10087)**

**Rushil Venkateswar (20CS30045)**

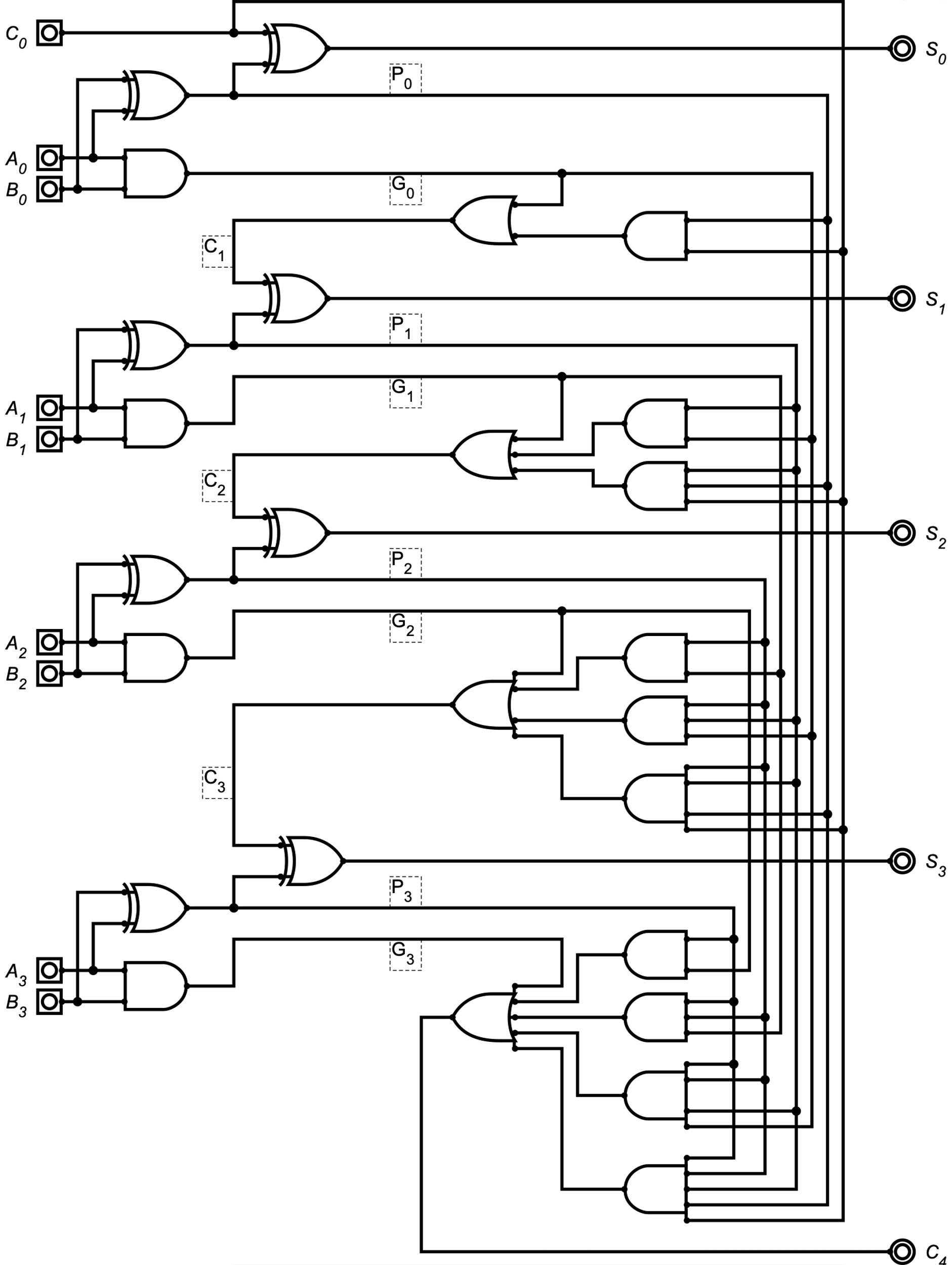
Group 69

Computer Organisation Laboratory

# CARRY LOOK-AHEAD ADDER (4-bit)

(Module File: half\_addr.v, Test: half\_addr\_TestBench.v)

Circuit Diagram:



A **carry look-ahead adder** improves speed by reducing the amount of time required to determine carry bits. It calculates one or more carry bits before the sum, which reduces the wait time to calculate the result of the larger-value bits of the adder (in **contrast to** the **ripple carry adder** which first calculates sum and carry and subsequent sum stages must wait until carry is calculated).

The logic is as follows:

Let then:

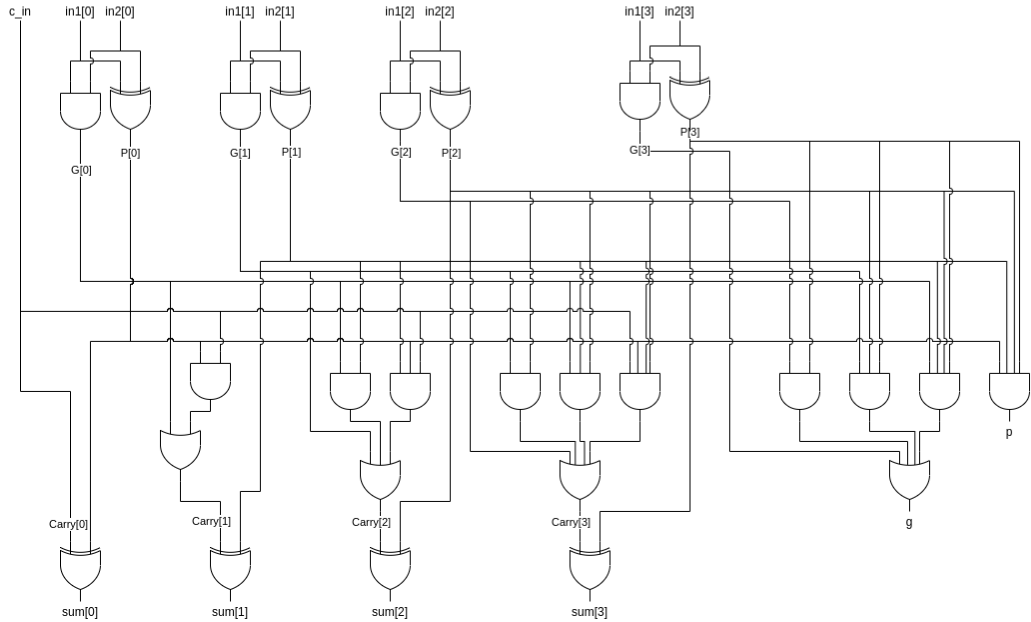
Recursively expanding, we get:

# 

# CARRY LOOK-AHEAD ADDER (4-bit, augmented)

(Module File: full\_addr.v, Test: full\_addr\_TestBench.v)

Circuit Diagram:



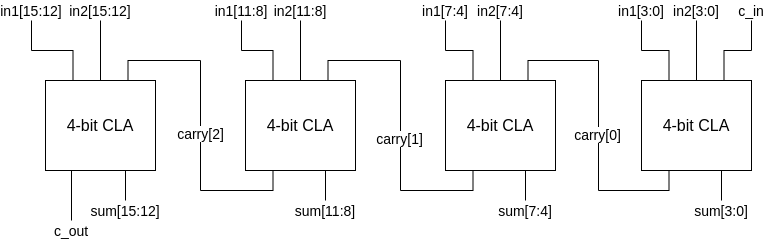
In this case instead of generating the carry out ,i.e., carry[4] we give the block propagate and generate as output which are then used by the carry lookahead unit. This leads to a modular design using which we can make 16, 32 and 64 bit adders by combining the block propagate and generate from the lower levels instead of rippling the carry out every time. The other logic remains the same as the normal 4-bit CLA.

The block propagate and generates are calculated as follows:

# CARRY LOOK-AHEAD ADDER (16-bit, cascaded)

(Module File: xyz.v, Test: abc.v)

This 16-bit adder is made by cascading four 4-bit CLAs by rippling the carry out from one block to another as shown in the figure.

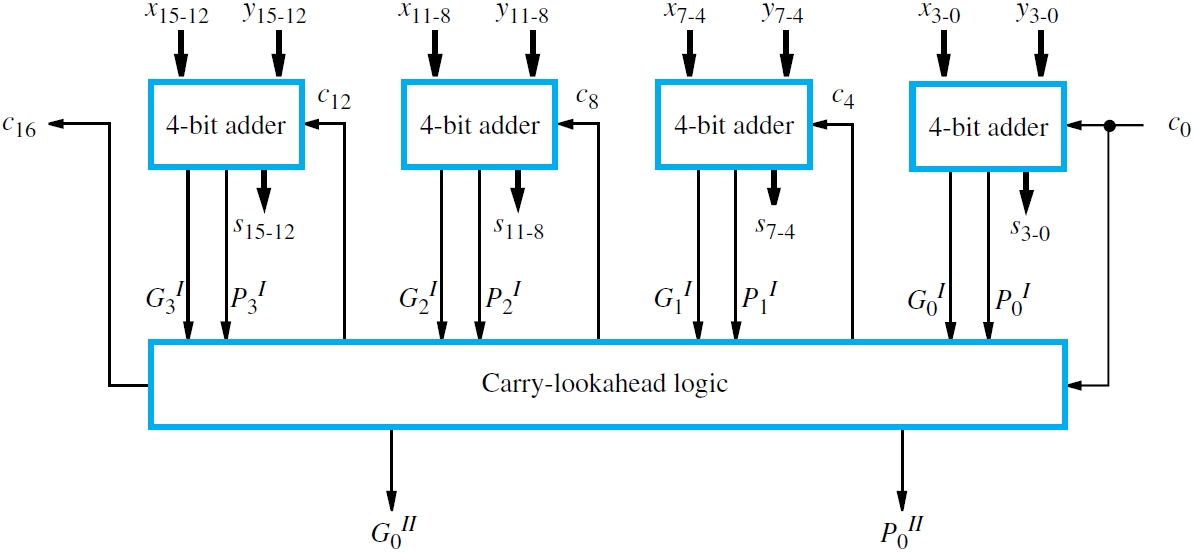


Here, are internal carries which are being propagated (or rippled) forward through the circuit.

# CARRY LOOK-AHEAD ADDER (16-bit, look-ahead)

(Module File: xyz.v, Test: abc.v)

Circuit Diagram:



Instead of rippling the carry bit, we take the carry-lookahead logic to a higher stage. The G and P bits returned by the 4-bit adder are labelled as and . Correspondingly, we extend the same logic which was used to form to the next stage to get and . This design reduces the delay in the circuit and allows us to use the 16-bit adder in the future for designing higher bit adders in a modular fashion.

The relevant equations are as follows:

Take to be then:

Recursively expanding, we get:

Final Propagate and Generate bits are calculated in just the same way as 4-bit CLA:

# SYNTHESIS SUMMARY & TIME DIFFERENCE

|  |  |  |  |
| --- | --- | --- | --- |
|  | 16-bit CLA Adder  (ripple carry unit) | 16-bit CLA Adder  (look-ahead carry unit) | Number of Slice LUTs |
| Delay (in ) |  |  |  |
| Levels of Logic |  |  |  |

CLA Adder with the additional look-ahead unit uses lesser logic levels, therefore propagating the sum and carry faster.

## Comparison with 4-bit RCA

|  |  |  |  |
| --- | --- | --- | --- |
|  | 4-bit RCA | 4-bit CLA | Number of Slice LUTs |
| Delay (in ) |  |  |  |
| Levels of Logic |  |  |  |

## Comparison with 16-bit RCA

|  |  |  |  |
| --- | --- | --- | --- |
|  | 16-bit RCA | 16-bit CLA | Number of Slice LUTs |
| Delay (in ) |  |  |  |
| Levels of Logic |  |  |  |

A CLA uses much lesser levels of logic as compared to an RCA which has a delay of for adding two -bit numbers. It can be observed that this additional delay contributes to a speed up of x