Computer Organization and Architecture Laboratory

Assignment 3 (Verilog)

Group 63:
Pranav Nyati - 20CS30037
Shreyas Jena - 20CS30049

Part 1:
Design of Ripple Carry Adders for 8, 16, 32, and 64 bits:

(a) Half Adder:

- **Inputs: in1** and **in2 (both are 1 bit inputs)**
- **Outputs: sum** bit and **c_out** bit **(carry out)**
- **Truth Table:**

Inputs		**Outputs**	
in1	**in2**	**sum**	**c_out**
0	**0**	**0**	**0**
0	**1**	**1**	**0**
1	**0**	**1**	**0**
1	**1**	**0**	**1**

Sum = in1 XOR in2; c_out = in1 AND in2

- **Figure:**

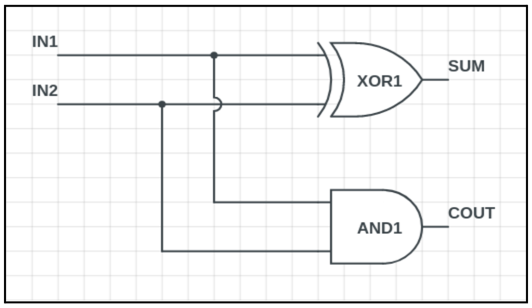

Fig 1. Logic Diagram of Half Adder

(b) Full Adder:

- **Inputs: in1, in2, c_in (all are 1 bit)**
- **Outputs: sum** bit and **c_out** bit
- **Truth Table:**

Inputs			**Outputs**	
in1	**in2**	**c_in**	**sum**	**c_out**
0	**0**	**0**	**0**	**0**
0	**0**	**1**	**1**	**0**
0	**1**	**0**	**1**	**0**
0	**1**	**1**	**0**	**1**
1	**0**	**0**	**1**	**0**
1	**0**	**1**	**0**	**1**
1	**1**	**0**	**0**	**1**
1	**1**	**1**	**1**	**1**

Sum = in1 XOR in2 XOR c_in;
c_out = (in1 AND in2) OR (in1 AND c_in) OR (in2 AND c_in)

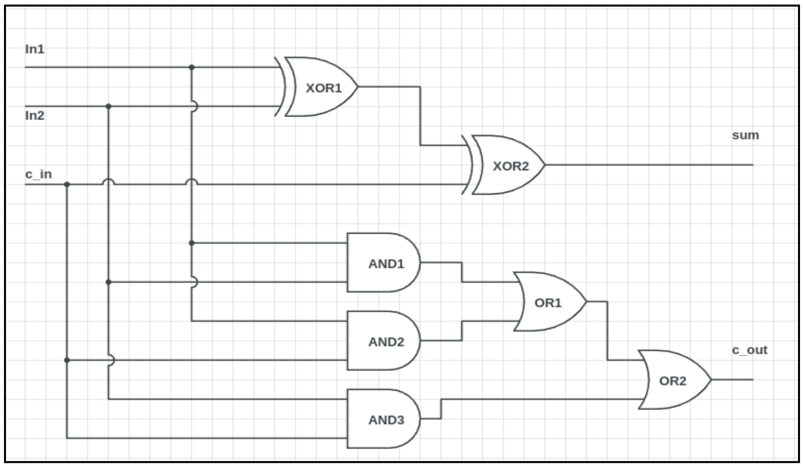

Fig 2. Logic Diagram of Full Adder

(c) Ripple Carry Adder (RCA):

We can create a 8-bit RCA by cascading 8 full adder circuits into a single circuit. Similarly, we can create a 16-bit RCA by cascading two 8-bit RCA modules, a 32-bit RCA by cascading two 16-bit RCA modules, and a 64-bit RCA by cascading two 32-bit RCA modules.

- **8-bit RCA:**

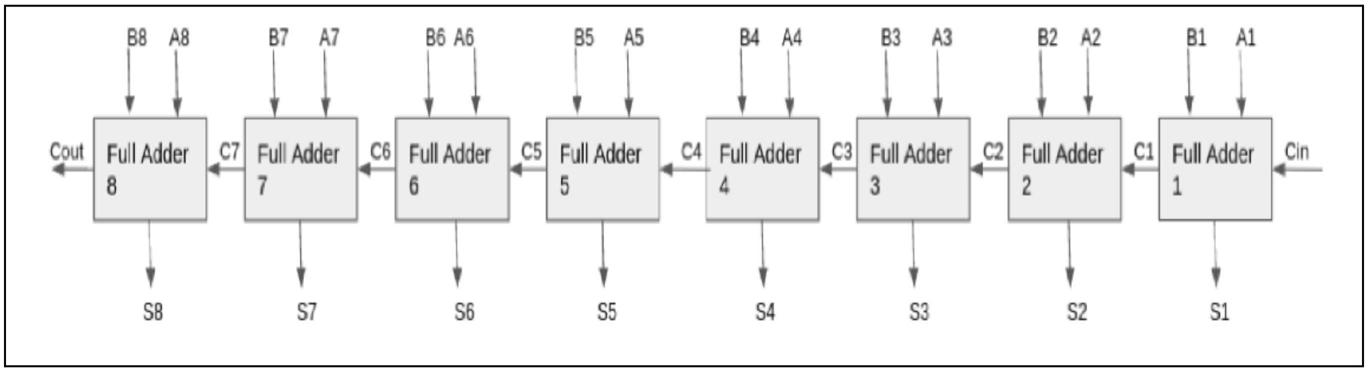

Fig 3. Logic Diagram of 8-bit RCA using 8 full adders in a cascading manner

- **16-bit RCA:**

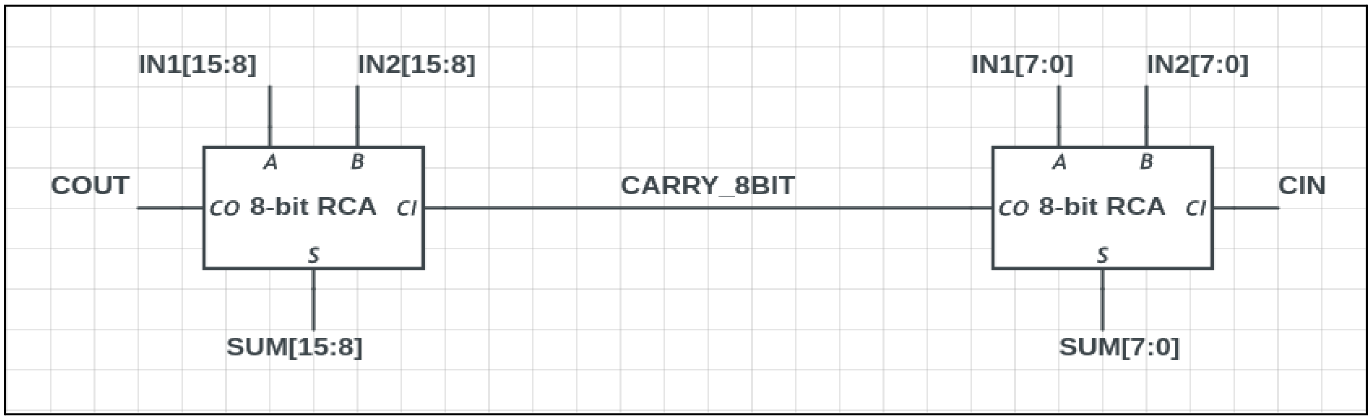

Fig 4. Logic Diagram of 16-bit RCA using two 8-bit RCA in succession

- **32-bit RCA:**

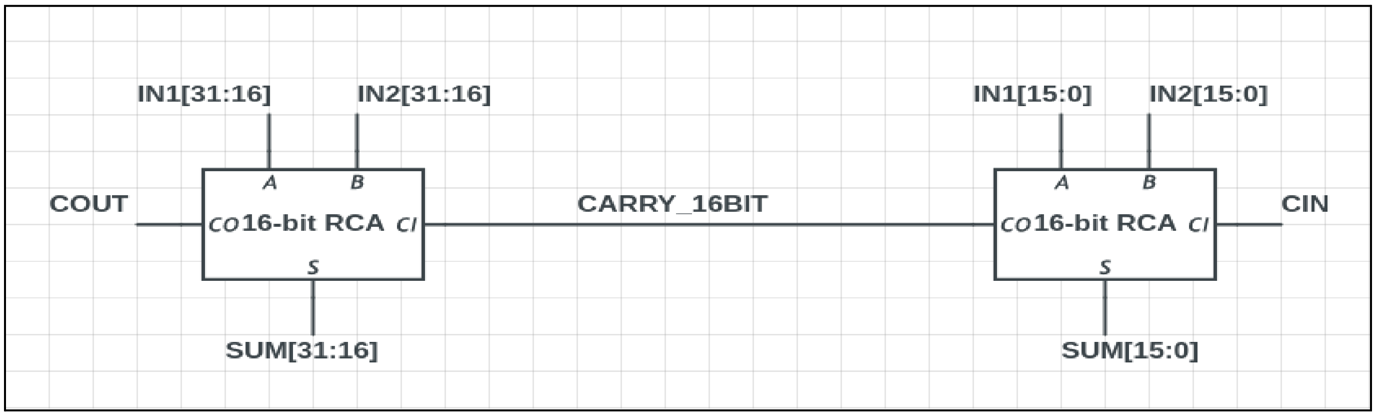

Fig 5. Logic Diagram of 32-bit RCA using two 16-bit RCA in succession

- **64-bit RCA:**

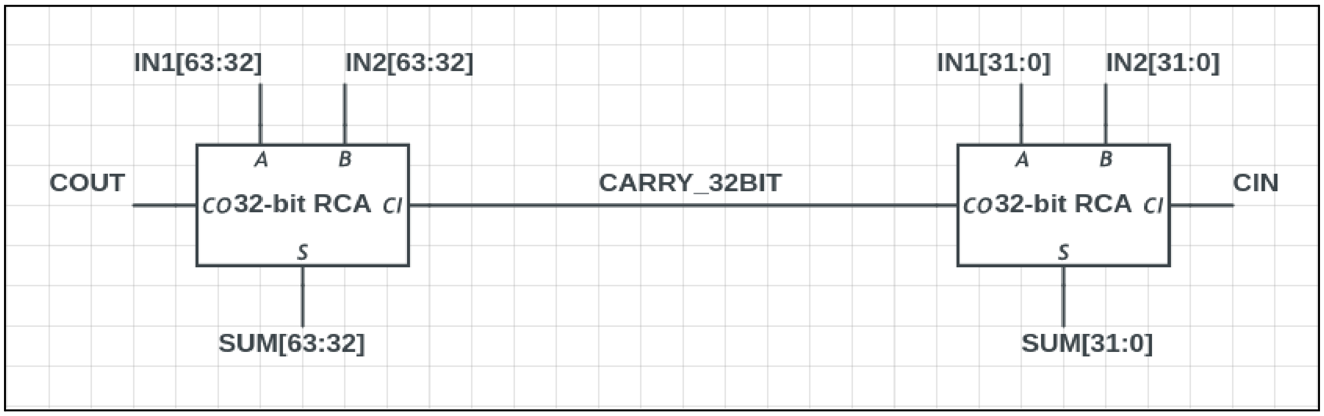

Fig 6. Logic Diagram of 64-bit RCA using two 32-bit RCA in succession

TIME DELAYS:

Observed longest timing delays:

- **8-bit RCA: 3.471ns** (0.497ns logic, 2.974ns route)
- **16-bit RCA: 6.167ns** (0.993ns logic, 5.174ns route)
- **32-bit RCA: 11.559ns** (1.985ns logic, 9.574ns route)
- **64-bit RCA**: **22.343ns** (3.969ns logic, 18.374ns route)

(d) How can you use the above circuit, to compute the difference between two n-bit numbers?

Soln:-

An n-bit RCA in a general case takes two inputs in1 and in2 with the carry-in bit = 0, and calculates the sum in1 + in2. In order to calculate the difference of two n-bit numbers in1 - in2 (let us assume that in1 > in2) , we can pass the first n-bit number as in1, the 2nd n-bit number as 1’s complement of in_2 (i.e, do bit wise complement of each bit in in2, and pass this 1’s compliment as in_2), and the initial carry_in bit as 1.

The result will be:

Res = (in1 + 1’s complement of in2 + 1)
= (in1 + (1’s complement of in2 + 1))
= (in1 + 2’s complement of in2)
= (in1 + (-in2))
= (in1 - in2)

Part 2 : Constructing 4-bit and 16-bit Carry Look-Ahead Adders

(a) Let X_i and Y_i (i = 0, 1, 2, 3) be the bit inputs to the ith full adder. This adder generates a carry-out bit C_{i+1}, and the corresponding generate and propagate signals are :

$G_i = X_i . Y_i$ (Carry generate signal)
$P_i = X_i \oplus Y_i$ (Carry propagate signal)

The Boolean equations describing the 4 carry-out bits C_1, C_2, C_3 and C_4 in terms of the generate and propagate signals G_0, G_1, G_2, G_3 and P_0, P_1, P_2, P_3 respectively are :

$C_1 = G_0 + P_0C_0$
$C_2 = G_1 + P_1C_1 = G_1 + P_1G_0 + P_1P_0C_0$
$C_3 = G_2 + P_2C_2 = G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_0$
$C_4 = G_3 + P_3C_3 = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 + P_3P_2P_1P_0C_0$

(where, c_0 is the initial carry-in bit for the 4-bit CLA)

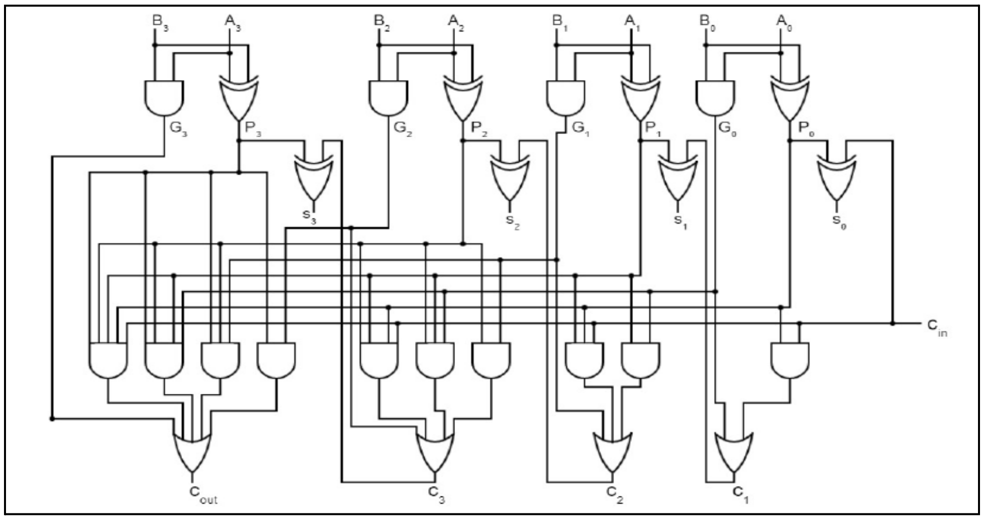

Fig 7 : Structure of a 4-bit CLA

(b)

Here, we test the execution speed of a 4-bit CLA and compare it with that of a 4-bit RCA.

Observed delay for a 4-bit CLA : **2.766ns** (0.373ns logic, 2.393ns route)

The observed delays for the 4-bit CLA can be attributed to the critical paths for the “carry-out” and “sum” operations.

Carry-out : The critical path for the carry-out bit is determined by tracing the route followed by all of the P_i and G_i bits to reach the four AND gates (located at the diagram's four far-left AND gates) and following the output of those gates to the OR gate.

Sum : The critical path for the sum is determined by how the P_i and G_i bits travel to the three AND gates (before the OR gate from which we receive C_3), how their results travel to the OR gate (from which we get C_3), and how C_3 travels to the subsequent OR gate (together with P_3).

For a 4-bit RCA, the following delay was observed :

Observed delay for a 4-bit RCA : **2.123ns** (0.249ns logic, 1.874ns route)

The observed delays for the 4-bit RCA can be attributed to the critical paths for the “carry-out” and “sum” operations.

Carry-out : The critical path for the carry-out bit is determined by tracing the steps taken to compute the carry-out bits, starting from the 1st full adder to the last full adder.

Sum : The critical path for the sum is determined by tracing the path taken by the carry bits to travel from the first full adder to the last full adder, followed by the sum computation in the last full adder.

(c) **Designing a 16-bit Carry Look-Ahead Adder :**

(i)

In the first part, we augment the created 4-bit CLA by adding lines for outputs P and G, the block propagate and generate signals respectively, calculated by :

$P = P_3P_2P_1P_0$ (block propagate signal)
$G = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0$ (block generate signal)

(ii)

In the second part, we implement a Look-Ahead Carry unit and integrate it with the 16-bit CLA as shown below :

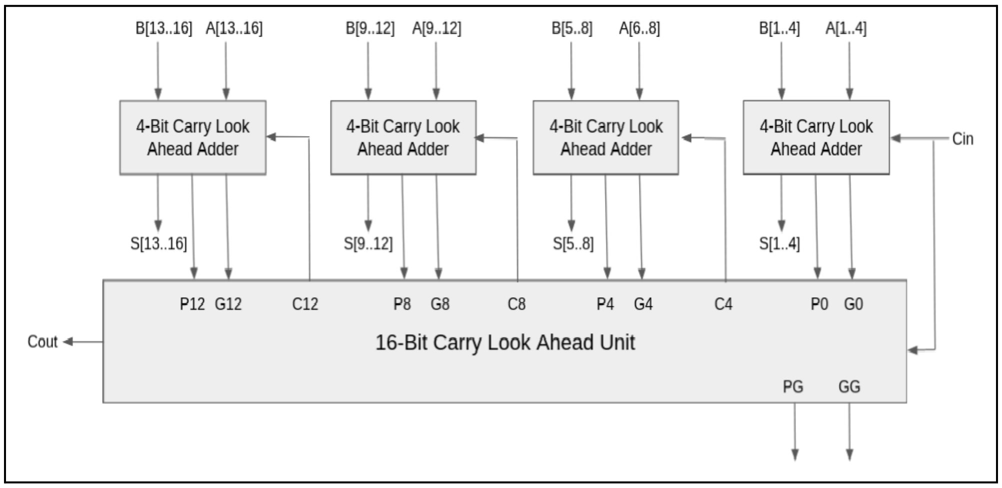
B[13..16] A[13..16]
B[9..12] A[9..12]
B[5..8] A[6..8]
B[1..4] A[1..4]
4-Bit Carry Look Ahead Adder
4-Bit Carry Look Ahead Adder
4-Bit Carry Look Ahead Adder
4-Bit Carry Look Ahead Adder
Cin
S[13..16]
S[9..12]
S[5..8]
S[1..4]
P12 G12 C12 P8 G8 C8 P4 G4 C4 P0 G0
Cout
16-Bit Carry Look Ahead Unit
PG GG

Fig 8 : Structure of a 16-bit CLA using Carry Look-Ahead Unit

(iii)

We need to compare the relative delays of a 16-bit CLA circuit using Look-Ahead Carry unit with the 16-bit CLA formed by rippling carry.

<u>For the 16-bit CLA with Carry Look-Ahead unit</u>,
<u>**Observed delay**</u> : **4.971ns** (0.621ns logic, 4.350ns route)
<u>**Best case Achievable Time**</u> : **4.436ns**

The delays observed for the 16-bit CLA with Look-Ahead Carry unit can be attributed to the critical paths for the "carry-out" and "sum" operations.

Carry-out : The 4-bit CLAs provide us with all the P_i and C_i bits at once. Next, C_{out} is calculated as $G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 + P_3P_2P_1P_0C_0$. This calculation uses the AND and OR gates in the look-ahead carry unit, contributing to the critical path.

Sum : Since the sum bits have already been calculated by the individual 4-bit CLAs itself, the critical path for the sum and carry-out bits is the same.

For the 16-bit CLA with rippling carry,
Observed delay : **8.094ns** (1.489ns logic, 6.605ns route)
Best case Achievable Time : **4.692ns**

The delays observed for the 16-bit CLA with rippling carry unit can be attributed to the critical paths for the "carry-out" and "sum" operations.

Carry-out : The intermediate carry bits C_4, C_8, and C12 need to be rippled through the individual 4-bit CLAs to produce the final carry-out bit, which determines the critical path in this case.
Sum : Since the sum is computed using the individual 4-bit CLAs itself, the critical path is the same as carry-out.

(iv)

After synthesizing the 16-bit CLA (using Look-Ahead Carry Unit) and 16-bit RCA, a comparative summary of the speed and LUT (Look-Up Table cost) for the two circuits is as follows :

Total Delay :
16-bit CLA : **4.971ns** (0.621ns logic, 4.350ns route)
16-bit RCA : **9.806ns** (1.985ns logic, 7.821ns route)

Slice LUTs used :
16-bit CLA : 48
16-bit RCA : 32

Tabular Summary of Time Delays of different RCA and CLA modules:

Module	**Total Delay (in ns)**	**Logic Delay (in ns)**	**Route Delay (in ns)**	**Number of Slice LUTs**
4-bit-CLA	2.766ns	0.373ns	2.393ns	7
4-bit-CLA-augmented	2.135ns	0.249ns	1.886ns	10
16-bit-CLA-ripple	8.094ns	1.489ns	6.605ns	28
16-bit-CLA	4.971ns	0.621ns	4.350ns	48
4-bit-RCA	3.194ns	0.497ns	2.697ns	8
8-bit-RCA	5.398ns	0.993ns	4.405ns	16
16-bit-RCA	9.806ns	1.985ns	7.821ns	32
32-bit-RCA	18.622ns	3.969ns	14.653ns	64
64-bit-RCA	36.254ns	7.937ns	28.317ns	128