A2Q1_carry_look_ahead_adder

Description:

A **8 Bit Carry Look Ahead Adder** is an improvisation over the classic **8 Bit Ripple Carry Adder.**

In **Ripple Carry Adders**, the **two** bits that are to be added are present instantly for each block. However, each adder block has to wait for the carry to arrive from the previous block before generating the sum and carry. So, technically, no adder block can generate the sum and carry until the input carry from the previous block is known.
i.e. The $\mathbf{i^{th}}$ block needs to wait for the $\mathbf{(i-1)^{th}}$ block to produce its carry before further calculation. So, there is a considerable time delay which is known as **Carry Propagation Delay.**

A **Carry Look Ahead Adder** reduces this **Carry propagation delay** by bringing more complex hardware in use. In this adder, the original design of the ripple carry counter is transformed in a way, such that the carry logic over fixed groups of bits of the adder is reduced to two-level logic.. More details on the working are given in the 'working' section.

Working:

Consider two **8-bit** binary numbers $\mathbf{A_7A_6A_5A_4A_3A_2A_1A_0}$ and $\mathbf{B_7B_6B_5B_4B_3B_2B_1B_0}$ are to be added.

They are mathematically added as,

C8	C7	C6	C5	C4	C3	C2	C1	C0
	A7	A6	A5	A4	A3	A2	A1	A0
+	B7	B6	B5	B4	B3	B2	B1	B0
	S7	S6	S5	S4	S3	S2	S1	S0

From here,

C1 = C0(A0 + B0) + A0*B0

C2 = C1(A1 + B1) + A1*B1

C3 = C2(A2 + B2) + A2*B2

C4 = C3(A3 + B3) + A3*B3

.

.

.

C8 = C7(A7 + B7) + A7*B7

In more generalised manner,

- **Gi = AiBi,** where **G** is called carry generator
- **Pi = Ai + Bi,** where **P** is called carry propagator

Then from above we get,

C1 = C0P0 + G0 ——————(1)

C2 = C1P1 + G1 ——————(2)

C3 = C2P2 + G2 ——————(3)

C4 = C3P3 + G3 ——————(4)

.

.

.

C8 = C7P7 + G7 ——————(8)

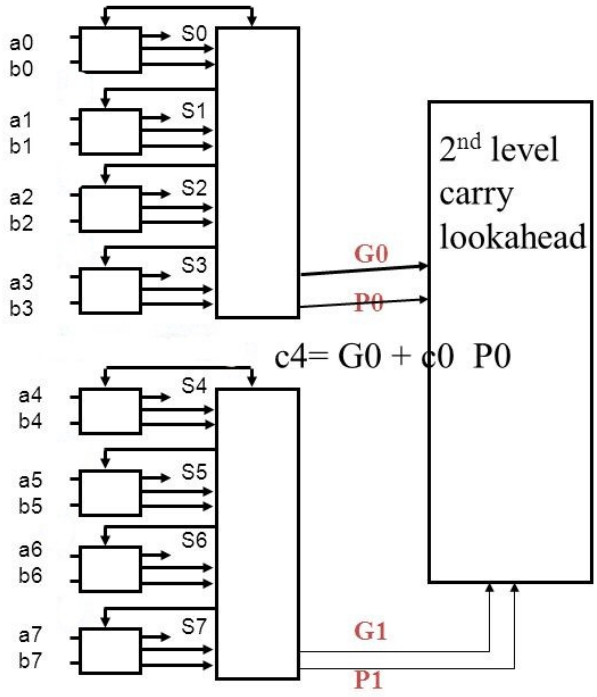
Finally,

- **C1 = C0P0 + G0**
- **C2 = C0P0P1 + G0P1 + G1**
- **C3 = C0P0P1P2 + G0P1P2 + G1P2 + G2**
- **C4 = C0P0P1P2P3 + G0P1P2P3 + G1P2P3 + G2P3 + G3**

....and so on.

❖ The **Carry Look Ahead Adder** Works in the same manner as described above

Circuit-Diagram:



Truth-Table:

A	B	C_i	C_{i+1}	Condition
0	0	0	0	No Carry Generated
0	0	1	0	
0	1	0	0	
0	1	1	1	No Carry Propagated
1	0	0	0	
1	0	1	1	
1	1	0	1	Carry Generated
1	1	1	1	