SUMIT KUMAR YADAV
18CS30042

AVIJIT MANDAL
18CS30010

EXPERIMENT-4
(ADDER DESIGN)

Aim:

1. You are required to build a 4-bit ripple carry adder using IC chips.
2. A ripple carry adder is known to be slow; think of a way to measure its speed and experimentally determine how your scheme works.

Apparatus Required:

1. Connecting Wires
2. TTL IC 74LS86-2 input (quad)XOR gate *2
3. TTL IC 74LS32-2 input (quad)OR gate *1
4. TTL IC 74LS08-2 input (quad)AND gate *3
5. Breadboard

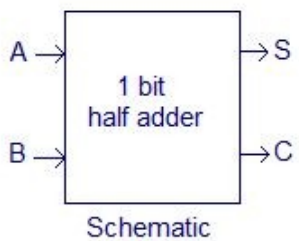
Theory:

Half Adder:

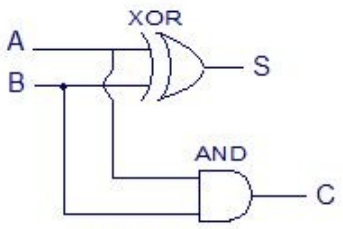
Half adder is a combinational arithmetic circuit that adds two numbers and produces a sum bit (S) and carry bit (C) as the output. If A and B are the input bits, then sum bit (S) is the X-OR of A and B and the carry bit (C) will be the AND of A and B. From this, it is clear that a half adder circuit can be easily constructed using one X-OR gate and one AND gate. Half adder is the simplest of all adder circuit, but it has a major disadvantage. The half adder can add only two input bits (A and B) and has nothing to do with the carry if there is any in the input. So if the input to a half adder have a carry, then it will be neglected it and adds only the A and B bits. That means the binary addition process is not complete and that's why it is called a half adder. The truth table, schematic representation and XOR//AND realization of a half adder are shown in the figure below.

Inputs		Outputs	
A	B	S	C
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

Truth table



Schematic



Realization

A half adder is able to add two bits (say) A and B to product the sum S=A ⊕ B and the carry
C=A · B.

Full Adder:

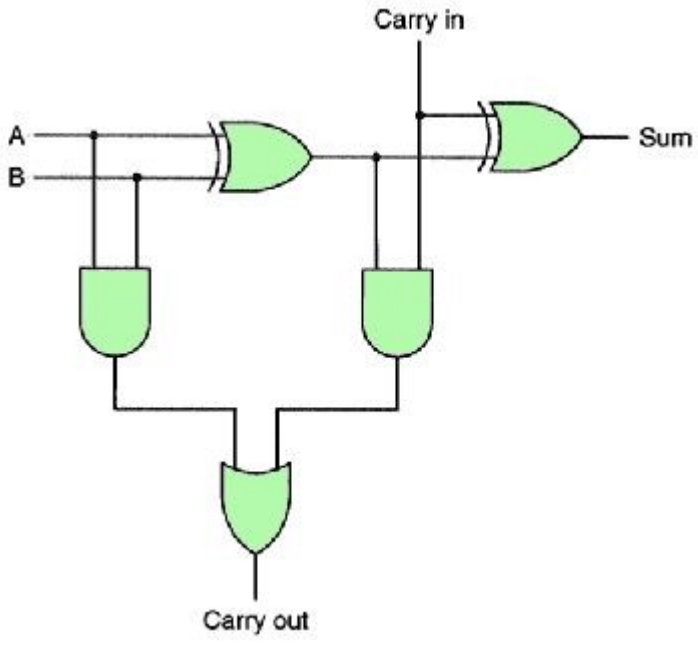
Full Adder is the adder which adds three inputs and produces two outputs. The first two inputs are A and B and the third input is an input carry as C-IN. The output carry is designated as C-OUT and the normal output is designated as S which is SUM.

A full adder adds binary numbers and accounts for values carried in as well as out. A one-bit full-adder adds three one-bit numbers, often written as A, B, and Cin; A and B are the operands, and Cin is a bit carried in from the previous less-significant stage. The full adder is usually a component in a cascade of adders, which add 8, 16, 32, etc. bit binary numbers. The circuit produces a two-bit output. Output carry and sum typically represented by the signals Cout and S, where the sum equals 2Cout + S.

A full adder logic is designed in such a manner that can take eight inputs together to create a byte-wide adder and cascade the carry bit from one adder to the another.

Addition of numbers also requires carries to be handled. Two half adders can be used to handle the two bits A and B along with the carry Ci, as indicated in the circuit, above. Now, sum S=A ⊕ B ⊕ Ci and the carry Co = (A · B) + Ci · (A ⊕ B) = (A · B) + Ci · (A + B)

A	B	Carry In	Sum	Carry 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



RCA (Ripple Carry Adder):

When multi-bit words are to be added, the corresponding bits may be added pairwise and the carrier can be rippled through to the next stage. With this scheme, the time taken by the carry to ripple through n bits is proportional to n. While the circuit is simple, its delay is significant, making this schene unsuitable for adding a number with several bits.

Multiple full adder circuits can be cascaded in parallel to add an N-bit number. For an N- bit parallel adder, there must be N number of full adder circuits. A ripple carries adder is a logic circuit in which the carry-out of each full adder is the carry-in of the succeeding next most significant full adder. It is called a ripple carry adder because each carry bit gets rippled into the next stage. In a ripple carry adder the sum and carry out bits of any half adder stage is not valid until the carry-in of that stage occurs. Propagation delays inside the logic circuitry are the reason behind this. Propagation delay is the time elapsed between the application of input and the occurrence of the corresponding output. Consider a NOT gate, When the input is “0” the output will be “1” and vice versa. The time taken for the NOT gate’s output to become “0” after the application of logic “1” to the NOT gate’s input is the propagation delay here. Similarly, the carry propagation delay is the time elapsed between the application of the carry-in signal and the occurrence of the carry out (Cout) signal. Circuit diagram of a 4-bit ripple carries adder is shown below.

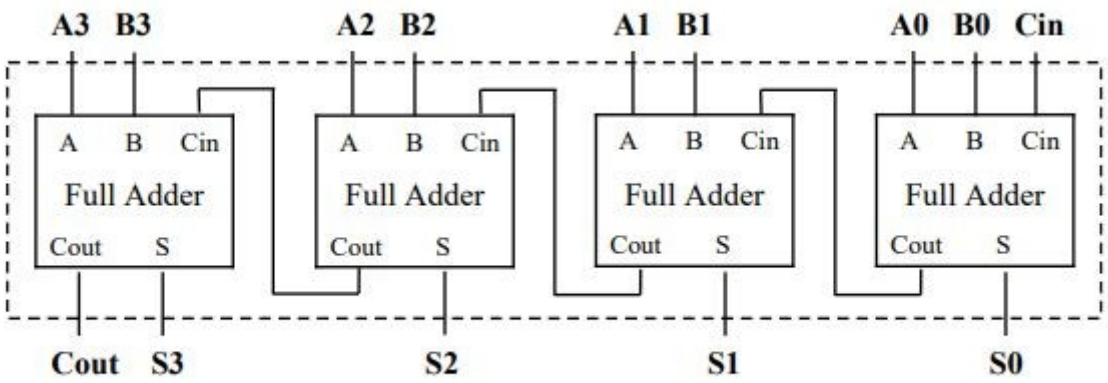


Figure 1: 4-bit Ripple-Carry Adder

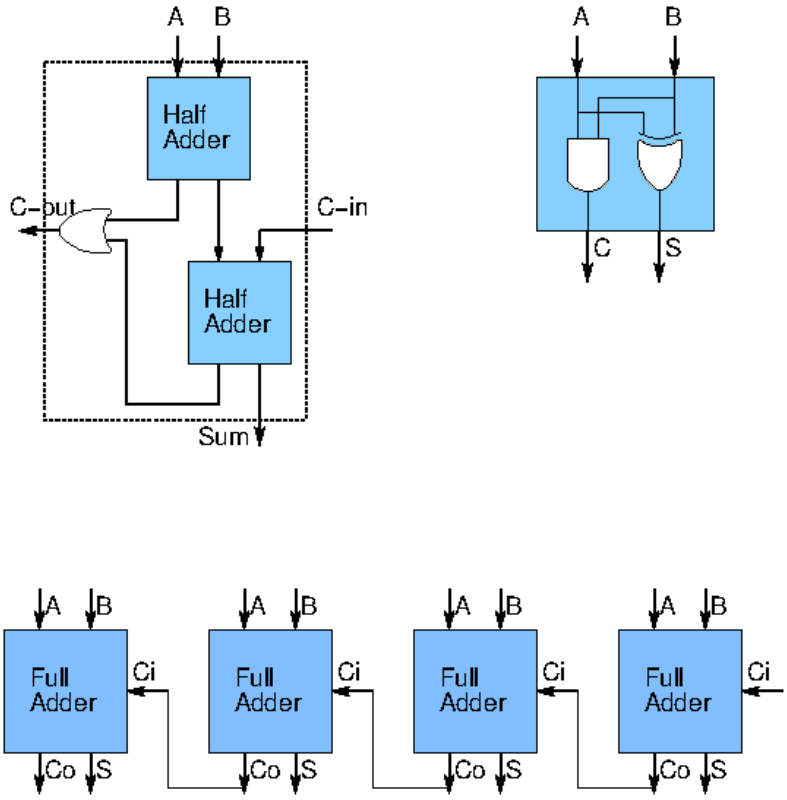
Setup for the Observations:

1. In this experiment, we used the 8 inputs available on the microelectronic circuit board to simulate all the combinations of the 4-bit inputs A (a0a1a2a3a4) and B (b0b2b2b3).
2. The connections were made using connecting wires and , 2 input(quad) AND(74LS08), 2 input(quad) XOR(74LS86) and 2 input(quad)

OR(74LS32) gates.
3. The entire circuit was broken down into subparts and the circuit was built in the following order:

a. Half Adder (Made from AND and XOR gates)
b. Full Adder (Made from Half Adder and OR gates)
c. Ripple Carry Adder (Made from full adders)

Circuit Diagram:



Calculations and Results:

To calculate the afore-mentioned delay, we set the C0 (input carry) to clock signal and give inputs in such a way that change in C0 induces a change in Cout.
In our case, we set A = 1000 and B = 0111 (both in binary), so that when C0 = 0, Cout = 0 and when C0 = 1, Cout = 1.
Now we set the clock frequency in kHz range (in fact, any frequency would do, as long as the oscilloscope can produce stable output with it). C0 and Cout are then connected to the oscilloscope channels.
The time interval between which C0 and Cout are unequal.

We observe that the ripple carry adder is a slower adder circuit and hence there is a delay in the output as compared to the input (as we can observe that from the output of the oscilloscope).

Discussion:

1. The ripple adder is a recursive module that can be implemented using the full adders which in turn is implemented using half adders.
2. There is a delay in the output as there is a waiting time for the carry input, which can be reduced using the pre-calculation (Carry Look Ahead).