NE-3102: Electronics-II Laboratory

Roll _____ Date _____ Experiment No. _____

Name of the experiment

Implementation of a 4-bit parallel adder using 7483 IC.

Contents

1 Objective

1. To implement a 4-bit parallel adder using 7483 IC.

2 Theory

A parallel adder can be constructed using any number of full adders (FA) by connecting them parallelly. An FA circuit adds two inputs and a carry input and generates one sum and a carry output. One FA's carry output is carry input to the next higher order FA. Thus, all the bits of both augend and addend are tied into the circuit parallelly and addition in each adder always happens simultaneously. For a 4-bit parallel adder, 4 FAs are required. In Figure 2, the variables A_0, A_3, A_2, A_1, represent the bits of the augend which is stored in the accumulator register, and the variables B_3, B_2, B_1, B_0 represents the bits of the addend which is stored in the B register. variables C_3, C_2, C_1, C_0 represents carry bits into the corresponding FAs. Sum appears at S_3, S_2, S_1, S_0 outputs.

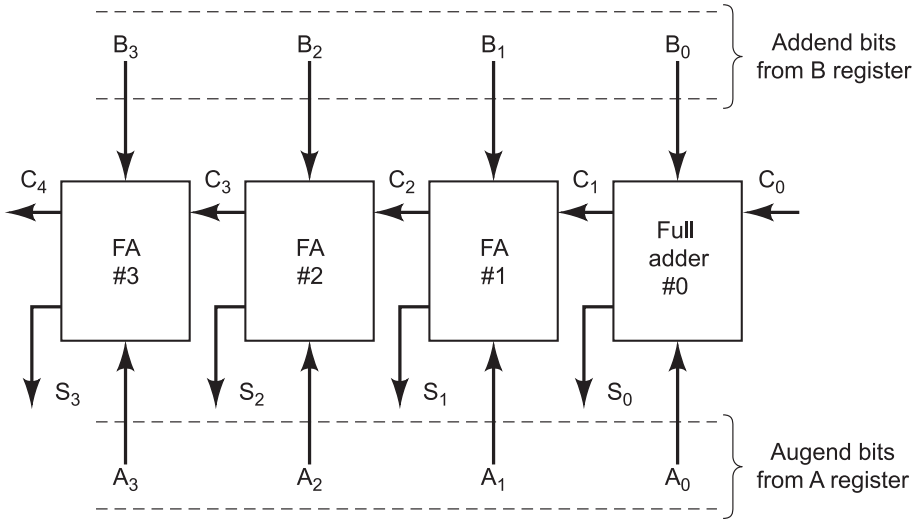

Figure 1: Block diagram of a four-bit parallel adder circuit using full adders.

A				B				Sum				Carry
A_3	A_2	A_1	A_0	B_3	B_2	B_1	B_0	S_3	S_2	S_1	S_0	C_4
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	0	0	1	0	0
0	0	1	0	0	0	1	0	0	1	0	0	0
0	0	1	1	0	0	1	1	0	1	1	0	0
0	1	0	0	0	1	0	0	1	0	0	0	0
0	1	0	1	0	1	0	1	1	0	1	0	0
0	1	1	0	0	1	1	0	1	1	0	0	0
0	1	1	1	0	1	1	1	1	1	1	0	0
1	0	0	0	1	0	0	0	0	0	0	0	1
1	0	0	1	1	0	0	1	0	0	1	0	1
1	0	1	0	1	0	1	0	0	1	0	0	1
1	0	1	1	1	0	1	1	0	1	1	0	1
1	1	0	0	1	1	0	0	1	0	0	0	1
1	1	0	1	1	1	0	1	1	0	1	0	1
1	1	1	0	1	1	1	0	1	1	0	0	1
1	1	1	1	1	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	1	1	1

Table 1: Truth table tor 4-bit parallel adder.

3 Components and apparatus

1. 7483 4-Bit binary full adder with internal carry lookahead IC
2. 7474 dual D flip-flop IC
3. 7486 quad Ex-OR gate IC
4. Passive components
5. Breadboard and connecting wires
6. Bench power supply

4 Circuit diagram/setup

The following is the implementation of a four-bit parallel adder circuit:[1]

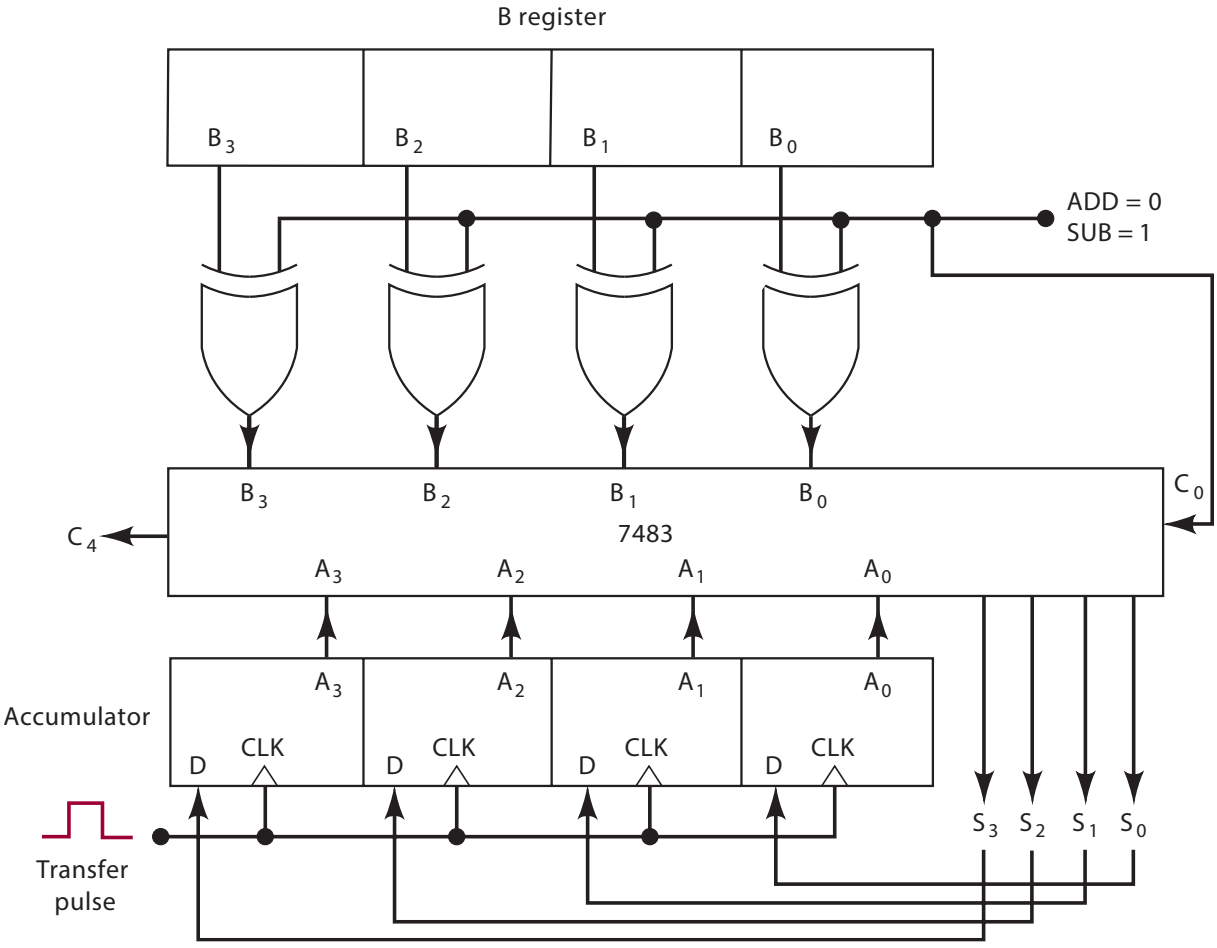

Figure 2: Implementation of a four-bit parallel adder circuit using a 7483 IC.

[1]Perform the arithmetic operation $6+1-4$ (or any other) in the 2's-complement system using the adder. The operation should be done in 2 steps: first, perform $6+1$ and then perform $7-4$. Values in the intermediate and final steps should be stored/accumulated in the Accumulator or the A register. Use two 7474 D flip-flop ICs for constructing the Accumulator register and asynchronously load the initial augend/minuend: in this case, the value is 6 in 4-bit binary, including the sign bit (2's-complements). Manually simulate the B register by applying HIGH/LOW on the bits. Note that an array of Ex-OR gates (plus a simulated carry) performs the plain binary to 2's-complement conversion between the B register and the points where the values are actually input to the 7483 IC, meaning that you do not need to perform the 2's-complement conversion; the circuit will do that for you. Do not put a timing generator/clock; manually synchronize the transfer between the adder output and the Accumulator instead. The implementation is essentially a 4-bit arithmetic logic unit (ALU).

5 Data collection and analysis

6 Result

7 Discussion

(The implementation may consist of 3 functional modules, (1) the 4-Bit binary full adder with internal carry lookahead IC that does exactly what it's supposed to, add A_n and B_n along with C_0 and generate S_n and C_4; (2) An array of Ex-OR gate that performs 1's-complement based on the logic level of the $\overline{\text{ADD}}$/SUB line; (3) a 4-bit Accumulator or A register made up of 4 D flip-flops w/ both asynchronous overrides and synchronous (parallel) inputs. Implement each of the modules independently and verify that they are functioning as expected. In the verification of the respective modules,

(1) See that you can add two random 4-bit binary numbers, e.g., $0001b + 0001b$, and get the expected results. Beware that some of the 7483 ICs available in the lab are actually 74283 ICs! 74283 is also a 4-bit parallel full adder IC but with a different pin layout. This is clearly a manufacturing defect. Use the IC tester to be sure.

(2) Take a random binary number, e.g., $0001b$, and put it as input to the Ex-OR array. The binary number should pass through to the output unchanged when $\overline{\text{ADD}}/\text{SUB} = 0$, and see that you the 1's-complement on the output when $\overline{\text{ADD}}/\text{SUB} = 1$.

(3) You should be able to asynchronously load a 4-bit random binary number, e.g., $0101b$, in the Accumulator using the asynchronous pins $\overline{\text{PR}}$ or $\overline{\text{CLR}}$ of each D flip-flop. Also, synchronously and parallelly load another number by making it available in the D inputs of the registar and pulsing the clock pin (you have to connect all the clock pins of all the flip-flops to a momentary switch).

Finally, integrate the modules to achieve the intended over-all functionality.

Asynchronously load the very first A_n value in the Accumulator using the asynchronous pins $\overline{\text{PR}}$ or $\overline{\text{CLR}}$.

If the very first number that you want to load in the Accumulator is negative, you have to do the 2's-complement yourself and asynchronously load the corresponding bit; the Accumulator does not have the fancy complement circuitry that the downstream of the B register has.

Hard-wire the B_n value of the B registrar in the inputs of the Ex-OR array. Set the correct logic level to the $\overline{\text{ADD}}$/SUB line.

If you want to add a number B_n to the number A_n stored in the Accumulator, hard-wire the number in the sign-magnitude form in the B register and unset $\overline{\text{ADD}}/\text{SUB} = 0$. The number will be passed to the adder as-is through the Ex-OR gates with $C_0 = 0$.

If you want to subtract a number B_n from the number A_n stored in the Accumulator, hard-wire the number in the sign-magnitude form in the B register and set $\overline{\text{ADD}}/\text{SUB} = 1$. Setting the wire will do two things; it will (1) take the 1's-complement through the Ex-OR gates and (2) set $C_0 = 1$; hence 2's-complement.

The output of the adder S_n will immediately be available to the D inputs of the Accumulator and ready to be loaded inside the flip-flops. Load the added output S_n by pulsing the clock pin (you have to connect all the clock pins of all the flip-flops to a momentary switch). The Accumulator will now have the new A_n value. Hard-wire the second B_n value of the B registrar in the inputs of the Ex-OR array. Set the correct logic level to the $\overline{\text{ADD}}$/SUB line. The output of the adder S_n will immediately be available to the D inputs of the Accumulator and ready to be loaded inside the

flip-flops. Load the added output S_n by pulsing the clock pin. The operation will go on like this.

Note that the range of valid magnitudes is $000b - 111b$ because a sign bit is to be pre-pended to the three magnitude bits, making it a four-bit number (the word size of the system) in the 2's-complement form.)

8 References

1. Tocci Ronald J, Neal W, Greg M. Digital Systems Principles and Applications.

Appendix