Military Institute of Science and Technology

Department of Computer Science and Engineering

Course Code: CSE-316

Course name: Digital System Design Sessional

CSE-20, Level-3, Term-II

Section: B

Group no: 3

Name of the project: Designing a 4-bit ALU

Group members:

Jamal Uddin Tanvin -202014016

MD Rifat Islam -202014034

Md Tausiful Haque -202014036

Nurshat Fateh Ali -202014040

Introduction:

An Arithmetic Logic Unit (ALU) is a combinational digital circuit that performs arithmetic, logical and bitwise operations on integer binary numbers. The inputs to an ALU are the data to be operated on, called operands, and a code indicating the operation to be performed; the ALU's output is the result of the performed operation. In many designs, the ALU also has status inputs or outputs, or both, which deliver information about a previous operation or the current operation, correspondingly, among the ALU and external status registers.

An ALU has a variety of input and output nets, which are the electrical conductors used to convey digital signals between the ALU and external circuitry.

A basic ALU has three parallel data buses containing of two input operands (A and B) and a result output (Y). Each data bus is a group of signals that delivers one binary integer number.

The opcode input is a parallel bus that conveys to the ALU an operation selection code, which is an enumerated value that identifies the desired arithmetic or logic operation to be performed by the ALU.

The status outputs of an ALU are various individual signals that convey supplemental information about the result of the current operation. General-purpose ALUs usually have status signals for instance:

- Carry-out: It conveys the carry resulting from an addition operation, the borrow resulting from a subtraction operation, or the overflow bit resulting from a binary shift operation.
- Zero: It indicates all bits of the result are logic zero.
- Sign: It indicates the result of an arithmetic operation is whether positive or negative.
- Overflow: It indicates the result of an arithmetic operation has exceeded the numeric range of result Y.

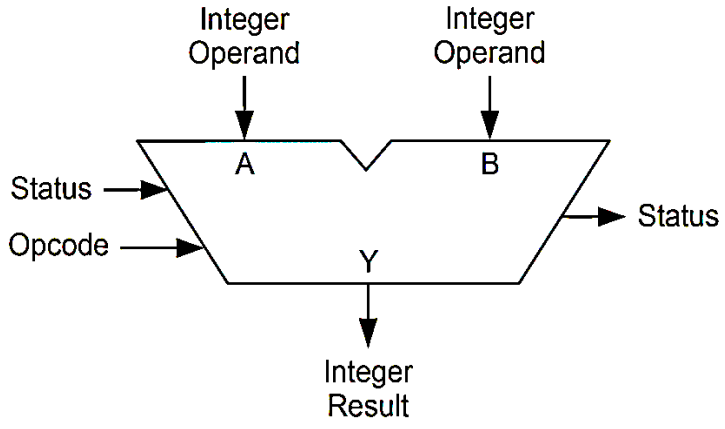


Figure: A symbolic representation of an ALU and its input-output signals along with status signals.

For our project we had designed a 4-bit Arithmetic Logic Unit (ALU), which generates the following operations and shows four status register (Sign flag, carry flag, Overflow flag, zero flag).

Operation	Function
F = B – A – 1	Subtract with borrow
F = B – A	Subtraction
F = B – 1	Decrement B
F = B	Transfer B
F = A XOR B	Exclusive-OR
F = B'	Complement B

Equipment:

1. Trainer board
2. IC 7408(AND gate)
3. IC 7432(OR gate)
4. IC 74LS25(NOR gate)
5. IC 7404 (NOT gate)
6. IC 7486(XOR gate)
7. IC 7483(4 Bit ADDER)
8. Logic probe
9. Wires

Design:

Truth table for the design is given bellow-

S_2	S_1	S_0	C_{in}	X	Y	F	Function
0	0	0	0	B	0	B	Transfer B
0	0	1	0	B	1	$B-1$	Decrement B
0	1	0	0	B	$\overline{A}$	$B-A-1$	Subtract with borrow
0	1	1	1	B	$\overline{A}$	$B-A$	Subtraction
1	0	0	0	$B \oplus A$	0	$A \oplus B$	Exclusive-OR
1	0	1	0	$\overline{B}$	0	$\overline{B}$	Complement B

Function for C_{in} , X, Y are given bellow-

$$C_{in} = S_1 S_0$$

$$X = \overline{S_2} B_i + S_2 \overline{S_0} (B_i \oplus A_i) + S_2 S_0 \overline{B_i}$$

$$Y = \overline{S_2}\,\overline{S_1}\,S_0 + S_1 \overline{A_i}$$

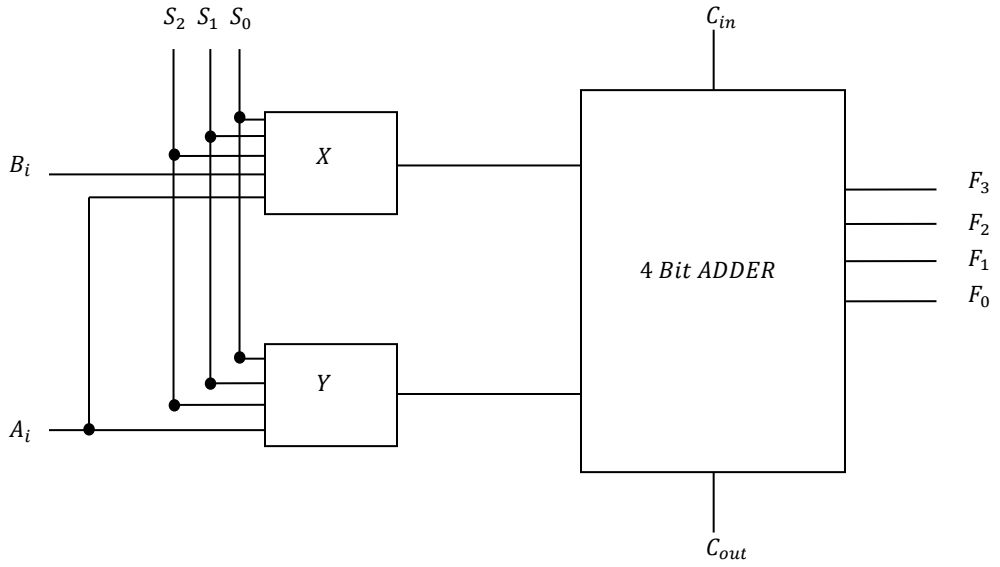
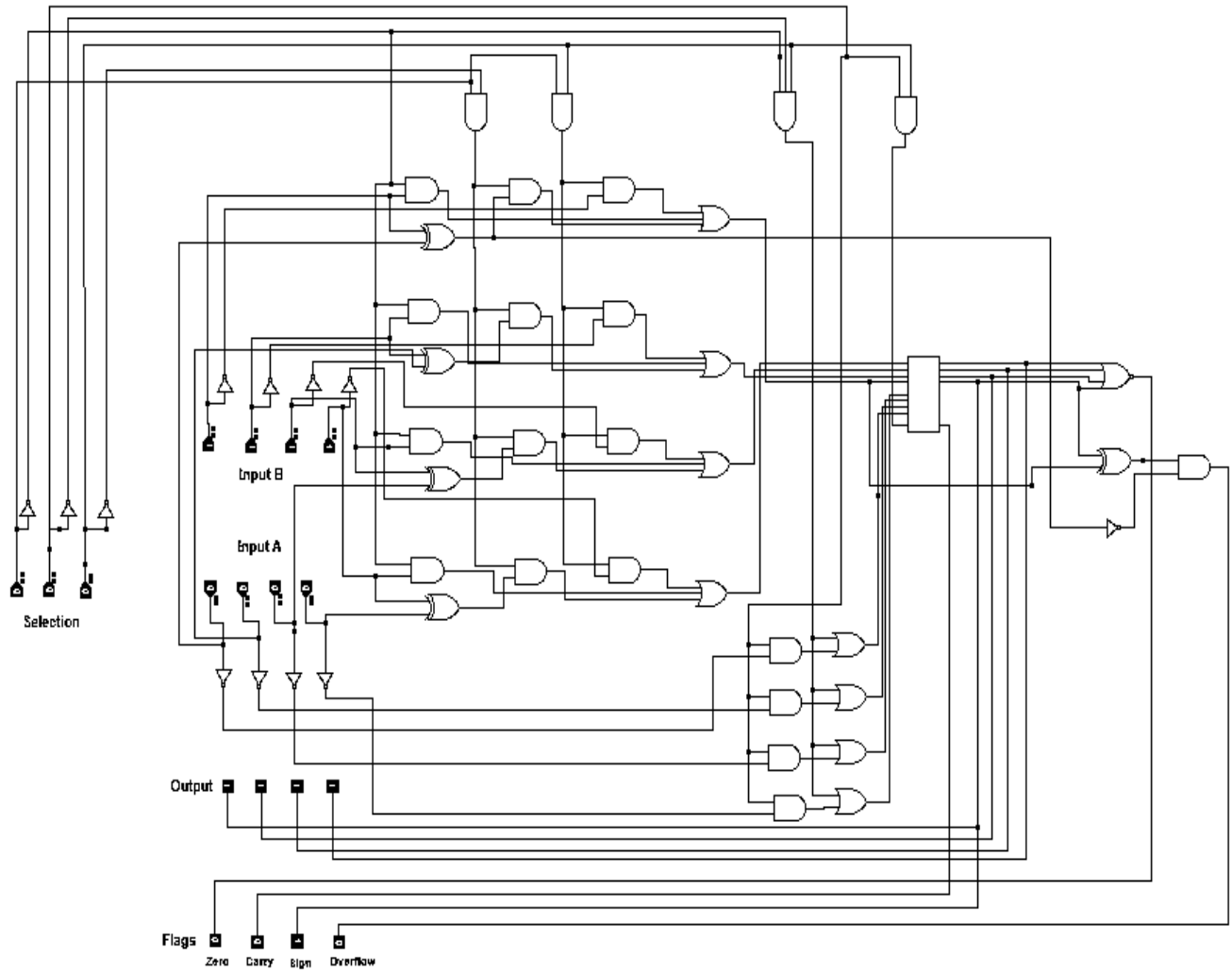


Figure: Block diagram of the designed ALU

The flag designs are given below-

- Sign flag = F_3
- Carry flag = C_{out}
- Zero flag = $\overline{F_3 + F_2 + F_1 + F_0}$
- Overflow flag = $(F_3 \oplus A_3)\,(A_3 \odot B_3)$

The proteus design of the ALU is given bellow



Number of gates and ICs with IC number:

IC type	IC model	Gates per IC	No of gate needed	No of IC needed
AND Gate (2 input)	7408	4	22	6
OR Gate (2 input)	7432	4	12	3
NOR Gate (4 input)	74LS25	2	1	1
NOT Gate	7404	6	12	2
XOR Gate (2 input)	7486	4	5	2
4 Bit ADDER	7483			1
Total			52	15

Limitations and Discussion:

- Limitations:
  1. Because of unavailability of 3 input OR gate we had to use only 2 input OR gates.
  2. As we designed our ALU with only one 4 Bit ADDER IC, we did not find the internal carry in pins of each adder in the 4 Bit ADDER IC.
  3. As we did not convey signal to the internal carry in pins of the adders in the IC, our combinational circuit X and Y was slightly larger than others.
  4. We had to make our Overflow flag slightly more complex because of the same reason of using only one 4 Bit ADDER IC.

- Discussion:
  1. The design was optimized as much as possible before implementing it in hardware.
  2. The design was checked in simulating software if it is working properly before implementation on hardware.
  3. It was checked that the required IC models were available and were working properly.
  4. Wires should be affix stiffly in the right place and it is recommended to use wires as less as possible as like the ICs.

Conclusion:

The given project which is a 4 Bit ALU gives us a perfect demonstration of the task of an ALU inside of a processor. Although our designed ALU can do some specific arithmetic and logical operation, the basic configuration is same as an ALU in of a microprocessor. We had the opportunity to use two 4 Bit ADDER ICs which could reduced more gates and make it much simpler. But we kept the current design and made it optimized as much we can and the design work perfectly without any flaw. There is still scope for improvements, but the project is a success.