BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
CSE 306 (Computer Architecture Sessional)
July 2023 Term
All Lab Sections, November 25, 2023

---

1. Introduction

As part of this assignment, you will have to submit **both software simulation and hardware** implementation of a simplified Arithmetic Logic Unit (ALU). The functional design specification for each group of each section can be found in Appendix A. This is a group assignment, and all group members must participate equally. The software simulation submission deadline for all sections is the same, and it is **December 10, 2023 (Sunday) at 11:55 PM**. On the other hand, the hardware demonstration deadline for all sections is also the same, which is **December 15, 2023** (see the Deadline section for details). You will also have to write a group report and submit it during the evaluation day.

2. Specification for 4-bit ALU Simulation

I. The functional design specification for each group of each section can be found in Appendix A. First, read the specification of your group carefully. Then, go through the following requirements/instructions in this section.

II. Efficiently design **(with minimum possible ICs)** the ALU according to the specification.

III. In addition, you need to implement the following flags.
   A. Carry (C)
   B. Sign (S)
   C. Overflow (V)
   D. Zero (Z)

IV. Flags will be affected as per the rules of Assembly Language. The simplified rules can be found in the following links.
   A. https://www.geeksforgeeks.org/flag-register-8085-microprocessor/
   B. https://www.geeksforgeeks.org/flag-register-8086-microprocessor/

V. However, we have added some exceptions (only for this assignment) to incorporate flexibility to flag status bits after logical operations. Remember that this flexibility is to make your assignment easier, although it breaks some of the rules of the assembly programming language.

A. For NOT Operation:

1. After the NOT operation, which makes the result 0000, if Z becomes 0 from 1, it will not be accepted. However, if Z becomes 1 or if the Z flag remains unchanged, both will be accepted.
2. After the NOT operation, if S remains unchanged or it reflects the highest order bit of the result, both will be accepted. But if the S flag is changed and it is changed to a wrong value, it will not be accepted.
3. To make your life easier, we shall not check the C and V flags after NOT operation, i.e., you can consider these as Don't care.

B. For AND/OR/XOR Operation:

1. C and V should be cleared (0) after the operation.
2. S and Z should be changed according to the output.

VI. Any 2-input SSI (AND, OR, NOT, XOR, etc.) and MSI (MUX, Decoder, Adder, etc.) chip can be used.

VII. Emphasis should be given to the efficiency of design and minimization of ICs used.

VIII. For simulation, you can use any simulation software.

IX. Your software design **must be at IC level.**

X. **Software Simulation Submission:** A submission link will be opened on *Moodle* for submitting your ALU simulation. Make a folder containing all your simulation project files, zip it, and submit it following the naming format. The naming format should be your section name followed by your group id (e.g., B1_Group7). Please ensure a single submission from each group (only a single member of the group should submit).

XI. **Report Preparation Guideline:** You have to write a report containing the followings:

A. Introduction
B. Problem Specification with assigned instructions
C. Detailed design steps with k-maps (if applicable)
D. Truth Table
E. Block Diagram
F. Complete Circuit diagram
G. ICs used with count as a chart
H. The simulator used along with the version number
I. Discussions
J. Contribution of Each Member

The report can be handwritten/ typed.

XII. **Software Simulation, Hardware Implementation Evaluation, and Report Submission:** Both software simulation and hardware implementation will be evaluated in the regular laboratory for respective sections (See the Deadline section for the specific time and date). Each group will have to bring at least one laptop to show the software simulation for that group. You have to show the full working hardware implementation

during the sessional time. Also, you need to submit your report at the beginning of the sessional.

XIII. For hardware implementation, you can lend the required instruments from the laboratory from now on (on the condition of returning these immediately after the evaluations and without any alternation), or you can use your own instruments.

XIV. Any type of plagiarism will be punished.

XV. All the specified date and time is according to Bangladesh Standard Time. Late submission is not allowed.

3. Deadline

I. Software Simulation Submission Deadline:
For all sections: December 10, 2023 (Sunday) at 11:55 PM

II. Software Simulation and Hardware Evaluation Timeline:

- A1: December 15, 2023 (9:00 AM)
- A2: December 15, 2023 (10.30 AM)
- B1: December 15, 2023 (2:30 PM)
- B2: December 15, 2023 (4:00 PM)

4. For Clarification

For any query, you can ask your instructor during the theory or sessional class. You can also use the ***Moodle*** thread or email at "saem@teacher.cse.buet.ac.bd".

5. Version

This section contains the version of the assignment. It starts with Version 0. If we find some major problems in this assignment description file, then we shall change this pdf. If that case, we shall increase the version number and list the changes in this section. So, keep an eye on this version number of the pdf in ***Moodle*** to see whether the version has been changed or not. If it is changed, first read this section to see where the changes have been made and whether it is applicable to your group. On the other hand, if the changes are minor (for example, correcting the grammatical mistakes), then the version number will not be changed.

5.1 Version 0

This is the initial version of the problem description pdf.

Appendix A

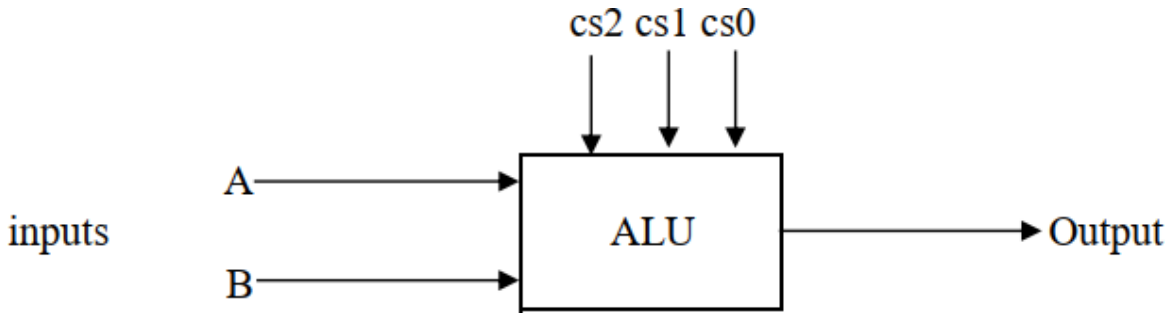
Functional Design Specifications for All Groups

List of Functions

S/L	Function Name	Description
1	Add	$A + B$
2	Add with carry	$A + B + 1$
3	Subtract	$A + B' + 1$ (i.e. $A - B$)
4	Subtract with borrow	$A + B'$ (i.e. $A - B - 1$)
5	Transfer A	Output is A
6	AND	$A \cap B$
7	OR	$A \cup B$
8	XOR	$A \oplus B$
9	Complement A	A'
10	Increment A	$A + 1$
11	Decrement A	$A - 1$
12	NEG A	$A' + 1$ (i.e. $-A$)

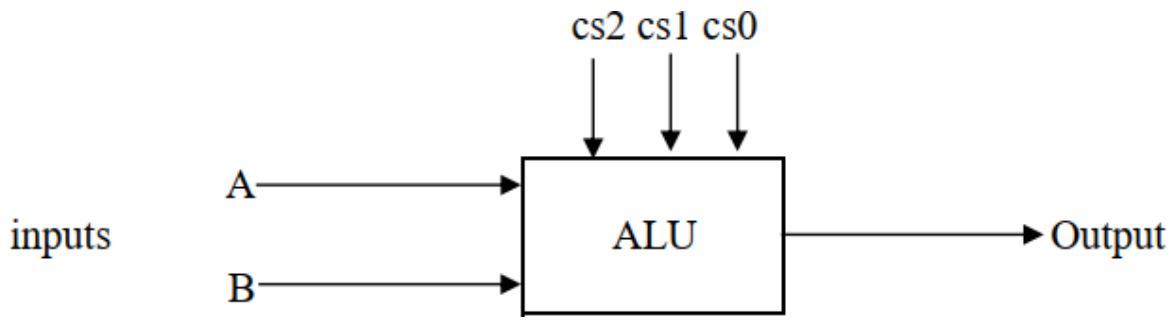
For Section A1

Control Signals			Functions For					
cs2	**cs1**	**cs0**	**Group 1**	**Group 2**	**Group 3**	**Group 4**	**Group 5**	**Group 6**
0	X	0	Add	Decrement A	Subtract with borrow	Decrement A	Add	Decrement A
0	0	1	NEG A	Subtract with borrow	Transfer A	Add	NEG A	Subtract with borrow
0	1	1	Add with carry	Transfer A	Subtract	Transfer A	Add with carry	NEG A
1	0	0	Increment A	Subtract	Increment A	Add with carry	Increment A	Subtract
1	0	1	AND	OR	AND	Complement A	OR	AND
1	1	X	XOR	Complement A	OR	XOR	Complement A	XOR



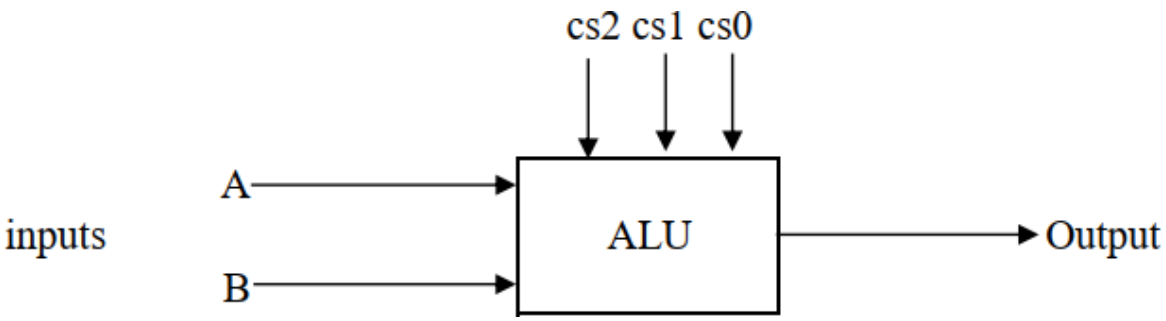
For Section A2

Control Signals			Functions For					
cs2	**cs1**	**cs0**	**Group 1**	**Group 2**	**Group 3**	**Group 4**	**Group 5**	**Group 6**
0	0	0	Add	Decrement A	Subtract with borrow	Decrement A	Decrement A	Sub
0	0	1	Transfer A	Subtract with borrow	NEG A	Add	Add	OR
0	1	X	Add with carry	Transfer A	Subtract	NEG A	XOR	NEG A
1	0	0	Increment A	Subtract	Increment A	Add with carry	AND	AND
1	0	1	AND	OR	AND	Complement A	Add with carry	Transfer A
1	1	X	XOR	Complement A	OR	XOR	Comple ment A	Add



For Section B1

Control Signals			Functions For					
cs2	**cs1**	**cs0**	**Group 1**	**Group 2**	**Group 3**	**Group 4**	**Group 5**	**Group 6**
0	0	0	Subtract with borrow	Decrement A	Decrement A	Subtract with borrow	Decrement A	Add
0	0	1	Subtract	Transfer A	NEG A	Subtract	Transfer A	Add with carry
X	1	0	OR	XOR	OR	AND	Complement A	AND
0	1	1	Transfer A	Add	Subtract with borrow	Transfer A	Add	NEG A
1	0	X	NEG A	Add with carry	Subtract	Increment A	Add with carry	Increment A
1	1	1	AND	Complement A	Complement A	OR	XOR	XOR



For Section B2

Control Signals			Functions For					
cs2	**cs1**	**cs0**	**Group 1**	**Group 2**	**Group 3**		**Group 5**	**Group 6**
X	0	0	Sub with borrow	Add	Decrement A		Decrement A	Subtract with borrow
0	0	1	NEG A	Add with carry	Transfer A		Transfer A	Subtract
X	1	0	XOR	XOR	OR		Subtract with borrow	NEG A
0	1	1	Increment A	NEG A	Subtract with borrow		Subtract	Add with carry
1	0	1	OR	Increment A	Subtract		AND	OR
1	1	1	Sub	AND	Complement A		Complement A	AND