BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
CSE 210 (Computer Architecture Sessional)
January 2024 Term
All Lab Sections, September 6, 2024

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 **September 27, 2024 (Friday) at 11:55 PM**. On the other hand, the hardware demonstration deadline is the next sessional day and is different for each section (see the Deadline section for details). You will also have to write a group report and submit its hardcopy on 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)

4টা task

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, Z flag is 1 if the answer is 0000 and the Z flag is 0 otherwise; ie. The Z flag functions as it normally would.
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.

logical operation
No carry/
overflow

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 (encouraged).

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 hardcopy at the beginning of the sessional evaluation.

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 (refer to *Section 5* for an approximate count).

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: **September 27, 2024 (Friday)** at **11:55 PM**

II. Software Simulation and Hardware Evaluation Timeline (Bring Report Hardcopy):

Week 5 (week of **28th September):**

- **A1:** September 28, 2024 (Saturday) at 11:00 AM
- **B1:** September 30, 2024 (Monday) at 2:30 PM
- **C1**: September 28, 2024 (Saturday) at 2:30 PM

Week 6 (week of **5th October):**

- **A2:** October 5, 2024 (Saturday) at 11:00 AM
- **B2:** October 7, 2024 (Monday) at 2:30 PM
- **C2**: October 5, 2024 (Saturday) at 2:30 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 *zaman.tanjeemazwad@gmail.com*

5. IC counts (approximate)

This contains an approximate count of each IC required.
(***You must minimize the total number of IC's used***)

	IC	Quantity
4x1 Mux ←	74153	2-3
Quad 2input Mux ←	74157	2-3
AND ←	7408	2-3
Quad 2input OR Gate ←	7432	2-3
NOT ←	7404	2-3
Quad 2input XOR Gate ←	7486	2-3
4bit full adder ←	7483	1-2

7402 → NOR

6. 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.

6.1 Version 0

This is the initial version of the problem description pdf.

6.2 Version 1

A Clarification in Section 2.V.A (regarding the Z flag after NOT operation)

Appendix A

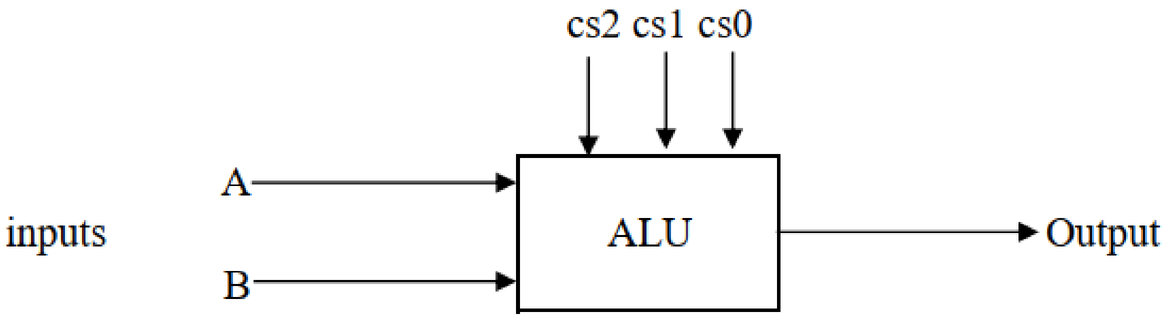
Functional Design Specifications for All Groups

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. $- A$)

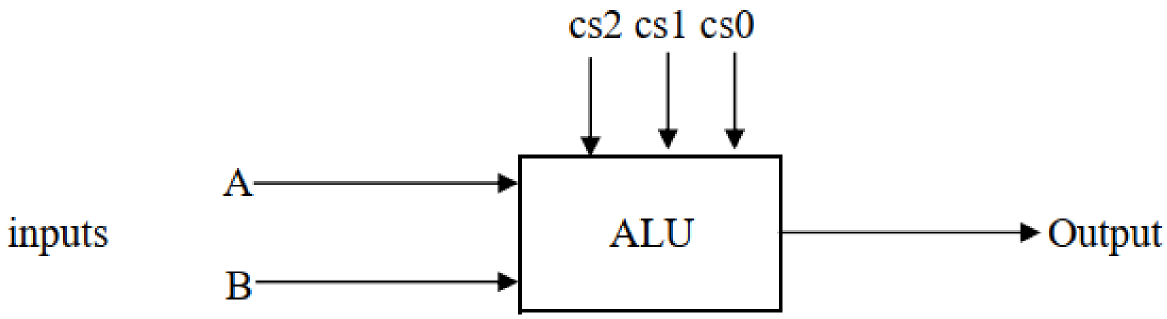
For Section A1

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



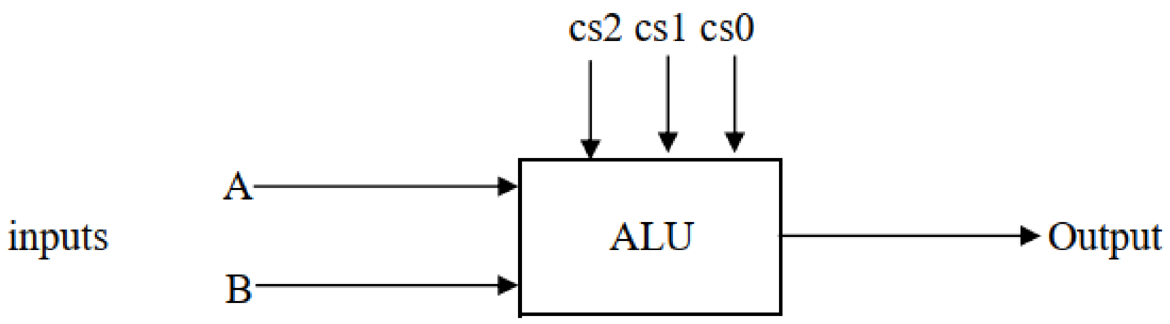
For Section A2

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



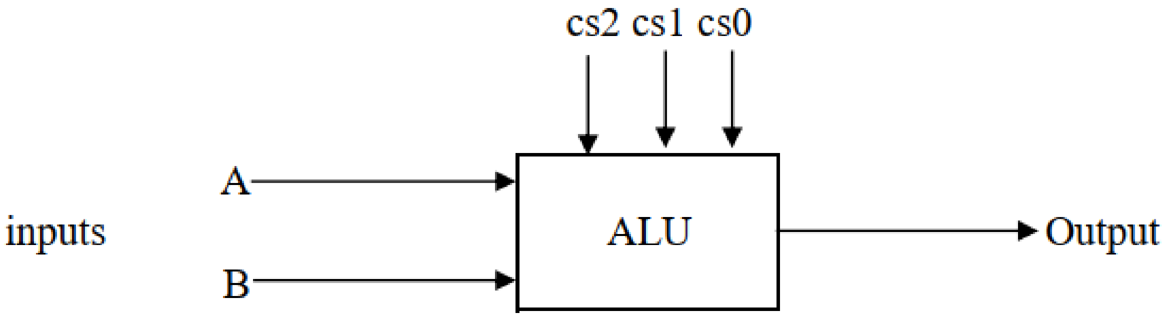
For Section B1

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



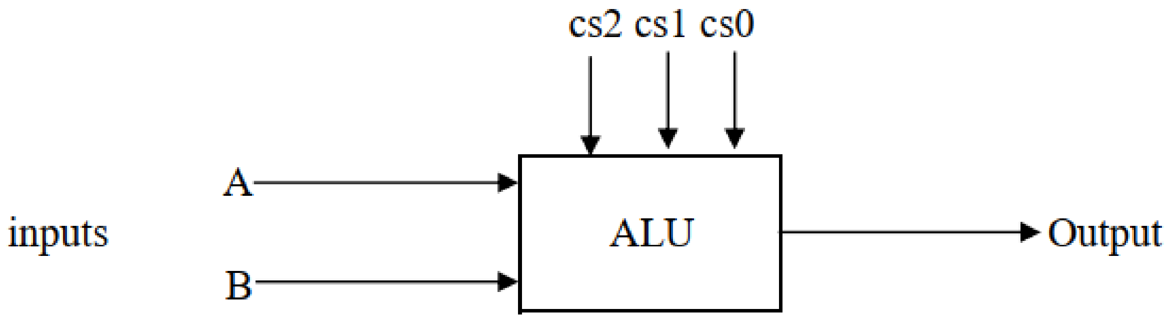
For Section B2

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



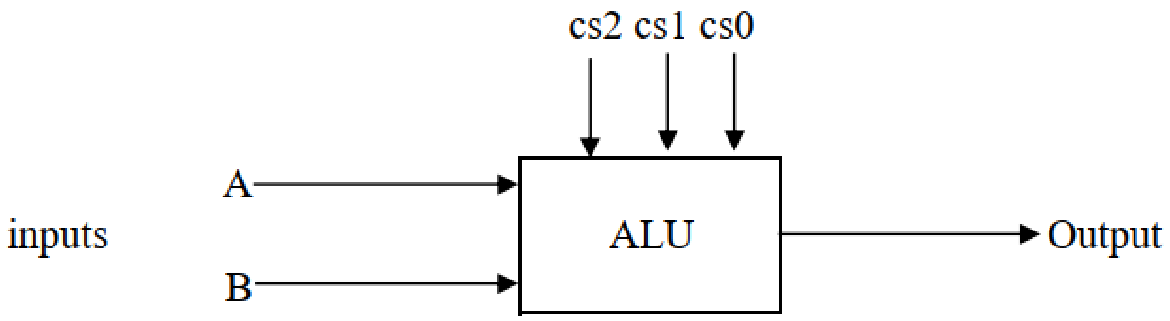
For Section C1

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



For Section C2

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


cs2 cs1 cs0
A
inputs
B
ALU
Output