ISTANBUL TECHNICAL UNIVERSITY

COMPUTER ENGINEERING DEPARTMENT

BLG 242E

DIGITAL CIRCUITS LABORATORY EXPERIMENT REPORT

EXPERIMENT NO	:	9
EXPERIMENT DATE	:	19.04.2019
LAB SESSION	:	FRIDAY - 14.00
GROUP NO	:	G13

GROUP MEMBERS:

150180704	:	CİHAT AKKİRAZ
150180707	:	FATİH ALTINPINAR
150180734	:	SİNAN ŞAR

SPRING 2019

ETHIC FORM
for
BLG242E Logic Circuits Laboratory

As a student of
Istanbul Technical University Faculty of Computer and Informatics Engineering;

1. I will not attempt to cheat in quizes and final exam,
2. I will not use disallowed sources or tools (mobile phone, calculator etc.) during the exam,
3. I will not write any information (formula, text, figure etc.) on the table, sheets or books that are allowed to be used during the exam,
4. I will give reference when using printed or online published sources,
5. I will not use the results in a source as they are, or by changing a part of them without giving a reference,
6. I will not show unused sources as used,
7. I will not present someone elses idea as my own idea,
8. I will not make someone do my homework, project or thesis for money or anything else,
9. I will not take an exam or enter a lecture on behalf of others,
10. I will not make excuses for not attending in exams or lessons by taking reports from someone I know (medical doctor parents or relatives),
11. I will refrain from deliberately harming the public materials at our university,
12. I will comply with the safety rules in laboratory work,
13. I will behave in accordance with the rules of respect for the lecturers and teaching assistants

signed by

150180704	:	CİHAT AKKİRAZ
150180707	:	FATİH ALTINPINAR
150180734	:	SİNAN ŞAR

Date: / / **SIGNED**

Contents

1 INTRODUCTION

In this experiment, 1-bit ALU is designed and implemented, then capability of this 1-bit ALU is extended to 4-bit ALU using serial approach.

2 REQUIREMENTS

Tools Used[1]

- C.A.D.E.T
- 74000 series ICs
  - 74xx08 - Quadruple 2-input Positive AND Gates
  - 74xx32 - Quadruple 2-input Positive OR Gates
  - 74xx83 - 4-bit Binary Full Adder
  - 74xx86 - Quadruple 2-input Positive Exclusive Or (XOR) Gates
  - 74xx151 - 8:1 Multiplexer
  - 74xx174 - Hex D-Type Flip-Flops

2.1 PART 1

In the first part of the experiment, 1-bit ALU is designed and implemented using necessary logic gates as given in the figure below. Logic switches are used as input(A,B,C_{in}) and outputs are observed via LEDs.

This 1-bit ALU is performing 7 different operations below.

Operations	RTL
Addition with Carry	$C_{out}O \leftarrow$ A + B + C_{in}
Subtraction	$C_{out}O \leftarrow$ A + B' + C_{in}
XOR	$O \leftarrow$ A $\oplus$ B
Clear	$O \leftarrow$ 0
Complement	$O \leftarrow$ A'
Two Complement	$O \leftarrow$ A' + C_{in}
Transfer B	$O \leftarrow$ B

Table 1: 1-bit ALU Operations

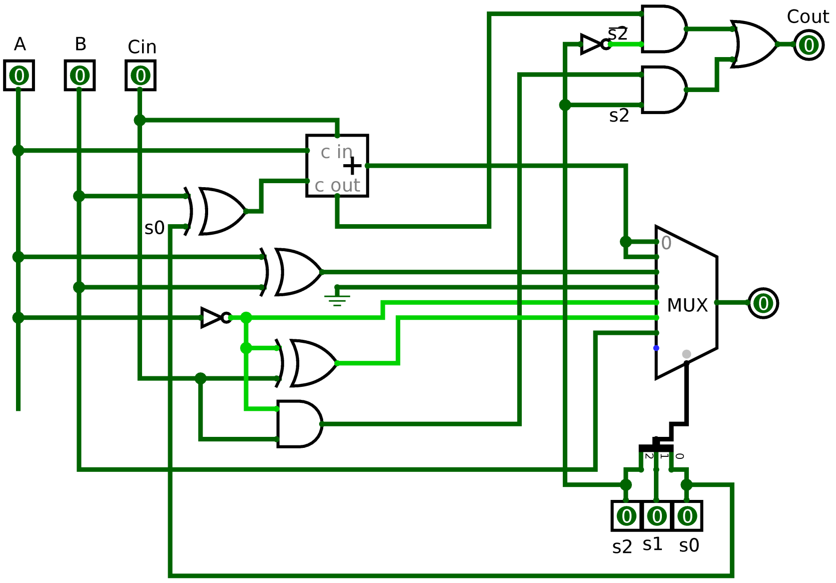

Figure 1: 1-bit ALU Circuit

Selector	A	B	C_{in}	C_{out}	Output
000	1	1	1	1	0
001	1	1	1	1	0
010	1	0	-	-	1
011	0	1	-	-	0
100	0	-	-	-	1
101	1	-	1	-	0
110	-	1	-	-	1

Table 2: Test Cases for 1-bit ALU

After this implementation, tests are performed for given cases above. Observed outputs are validated with calculations before the experiment.

2.2 PART 2

In the final part of the experiment, capabilities of 1-bit ALU are extended to 4-bit variables by using memory units. 4-bit ALU is implemented by using 1-bit ALU which is implemented in the previous part of the experiment. Using additional ALUs connected in parallel dramatically increases the complexity of the circuit. In this part, serial approach is used to overcome this challenge. 4-bit ALU is created via using a 1-bit ALU and memory units as shown given figure below.

To do 4-bit operations on a 1-bit ALU, 4 clock cycles are required. This is achieved by applying least significant bit of the operand to B and applying least significant memory

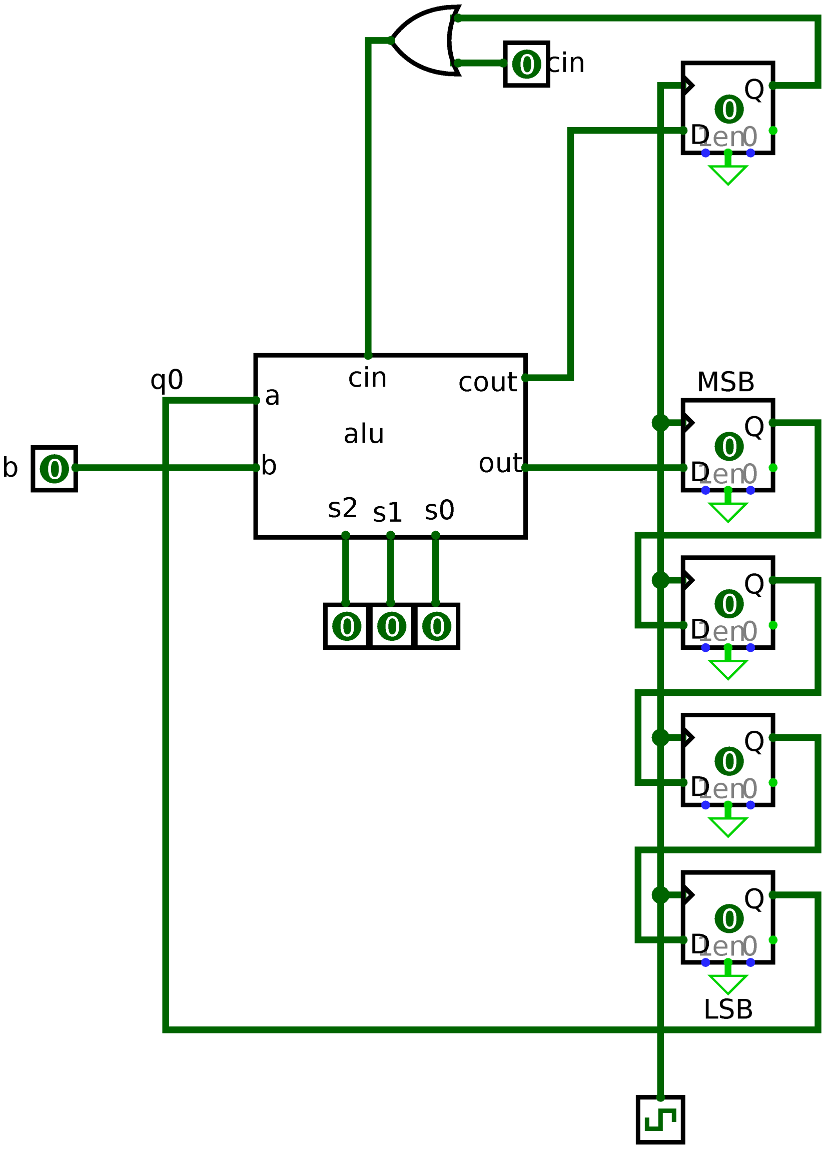

Figure 2: 4 Bit ALU using 1-bit ALUs

unit to A then the result is written to most significant memory unit every clock cycle. At the same time memory units are shifted from most significant towards least significant. Thus after the 4^{th} cycle, result of the operation that is applied on least significant bit is stored in least significant memory unit. C_{in} is connected to C_{out} which allows carry transferring between bits when required. User can also write C_{in} and it should be only applied while doing an operation over least significant bit.

Selector	A	B	C_{in}	C_{out}	Output
000	0010	1001	1	0	1100
001	0101	0111	1	0	1110
010	1010	0010	-	-	1000
011	1110	0101	-	-	0000
100	0111	-	-	-	1000
101	1100	-	1	-	0100
110	-	1011	-	-	1011

Table 3: Test Cases for circuit

3 INTERPRETATION OF THE RESULTS

We were able to get consistent results throughout the both parts of the experiment (Details of both designs are given in the sections above.). The 1-bit ALU we designed worked without any issues. (However, display units on the CADET were causing some trouble while we were observing the results. After trying a few different LED's, we found some that worked without any issues.) Our extended 4-bit design also functioned flawlessly and was consistent with its implementation in Logisim.

4 CONCLUSION

The experiment taught us a different implementation technique to build an ALU with multiple bits. It decreases the complexity of the circuit but requires more clock cycles than a parallel approach. We also organised our actions in this experiment better than any previous experiments. Which led to faster and more reliable progress.

REFERENCES

[1] Istanbul Technical University Department of Computer Engineering. Blg 242e logic circuits laboratory experiments booklet version 1.9, Spring 2019.