British Journal of Computer, Networking and Information Technology

Vol.1, No.1, pp.1-18, 2018

www.abjournals.org

ALU DESIGN BY VHDL USING FPGA TECHNOLOGY AND MICRO LEARNING IN ENGINEERING EDUCATION

Ismail Said and M. S. Çavuş

[1]Material Science and Engineering, Institute of Sciences, University of Kastamonu
[2]Biomedical Engineering Department, Faculty of Engineering and Architecture, University of Kastamonu

ABSTRACT: *The aim of this study is to develop case-study called Allowing Complexity to Complex Project (ACCP) for micro- learning in order to achieve high performance in computer architecture education and to test the legitimacy of classical teaching methods. The Arithmetic/Logic Unit (ALU) design was used as an example of the ACCP which consists of many examples and models aimed at developing students' skills in using complex arithmetic logical shifting and rotation instructions. Moreover, in this study, a real-world project on Field Programmable Gate Arrays (FPGA) devices was also developed using micro learning (ML). To this end, various hardware and software programs designed for computer architecture education and training were combined with improved instructional and attractive examples.*

KEYWORDS: FPGA, Digital Design Electronic, Engineering Education, Micro Learning, VHDL, Arithmetic Logical Shifting and Rotating Units.

INTRODUCTION

The title of this paper is the design of the ALU by VHSIC Hardware Description Language (VHDL) for engineering education - using FPGA technology and the appropriate teaching method micro-learning. The ALU case study is designed by using FPGA with ML, that is, computer architecture education. The main objective of the study is to solve the problem which encountered the traditional education in satisfying the real-world educational needs and to keep pace with the developments (Kiray, Demir and Zhaparov, 2013). Accordingly, FPGA technology and micro-learning method should be used. There are increasing requirements needed to develop computer architecture teaching for more active learning and to satisfy the requirement of the real-world digital design by using FPGA technology and the ML method which supports the phases of the learning process (Soyler, Gunaratne and Akbas, 2017). As a result, we propose to present FPGA technology and ML method as a target of this study the implementation of (ALU) via VHDL (Very high-speed integrated circuit Hardware Description Language) design (Monmasson and Cirstea, 2007).

Many studies have been conducted emphasizing the importance for developing useful and attractive examples to teach digital electronics more efficiently and rapidly in order to improve the performance of student. Besides, FPGAs are widely utilized as educational materials by many electronic engineering and computer engineering departments at universities around the world. Its usage in the engineering field dates back to earlier years, but studies conducted in those years are very limited (Zhamanov and Zhamapor, 2013). Every day students are receiving vast amount of new materials; however, many of them do not really understand all this kind of information in a short time during the lecture (Job and

Ogalo, 2012). Student success and performance is dependent on acquiring and using relevant information content and appropriate communication tools to achieve task objectives through the new teaching method (Shir, Bidabadi, Isfahani, Rouhollahi, Khalili and Bidabadi, 2016). Therefore, the study is aiming at enhancing the simulator-based approach by integrating some hardware design and teaching method to motivate the computer architecture students to acquire a hands-on experience in hardware-software. Throughout this study, the intention is examining this new education method and FPGA technology to improve computer architecture education and making the learning process more effective, attractive, and instructive; in addition, overcoming the requirements of design complexity. In accordance, the study will provide an instructive and attractive projects and examples covering all the contents of computer architecture subject. The developed projects and examples combine the various hardware and software for effective and smart computer architecture education. The purpose of the project is developing case studies so-called "*Allowing Complexity to Complex Project*" for micro-learning to obtain high performance of computer architecture education.

ML is a new research area intended at exploring new ways of responding to the growing need for lifelong learning or learning on demand for members of our society such as workers. The education industry is regularly requiring to update the curriculum to cope with the changing demand of industry and business to meet the challenges in the internal and external environment of businesses (Tremblay, Lalancette, Roseveare, Dias and Amaral, 2013).

METHODOLOGY

In this research, a quantitative research methodology is used based on a questionnaire survey. The result of this study highlights the importance and need for micro-learning in the education sector. Accordingly, a comparison was made between micro-learning and traditional education. This study was carried out at Turgut Özal University and Kastamonu University. It was lasted for one year throughout a period of two semesters. The study has been applied to two groups where there are 46 students in one group and 43 students in the other.

At the beginning of the study, the problems that were facing students in traditional education were identified. A plan was then formulated to solve these problems using a new technique (FPGA) and teaching method (ML). This work has been done in two separate semesters. In the first semester, the rational teaching method was used without utilizing the FPGA+ML whereas the FPGA+ML approach was introduced in the second semester. For the purpose of assessing the applied approach and the teaching method, a questionnaire survey has been developed. The questionnaire comprises five factors; instructive, attractive, effective, proficiency, and allowing complexity. In addition, three homework's were distributed to the students and midterm and final exams were given to the students.

Eventually, the study was practically implemented as follows:

A complex project was designed and divided into useful and attractive examples. After that, the micro-learning method was applied to all the contents of the subject. These activities are summarized as follows:

- A project combining various hardware and software was developed.

- ALU was designed by VHDL.
- FPGA technology was applied.
- Finally, the education method, Micro-Learning was employed.

However, problems related to the traditional education methods can be grouped under five main topics:

- Slow comprehension
- Unable to design complex projects
- Long learning time
- The development of education cannot be accelerated.
- Use of breadboards

The organizational card of the proposed education method within the scope of this study is given in Figure 1.

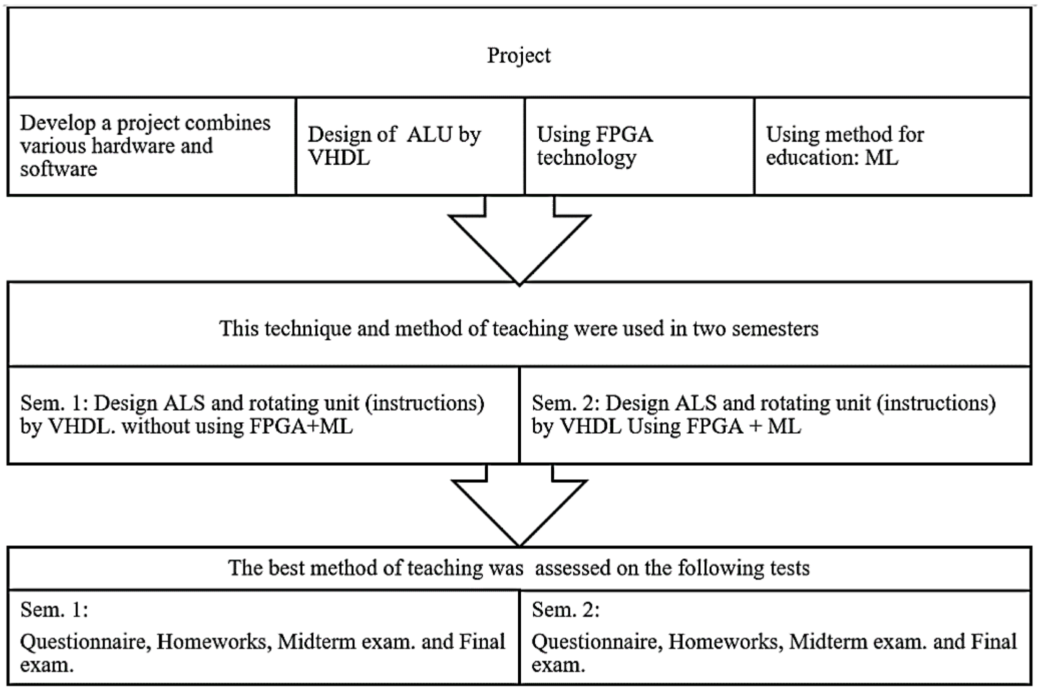

Project
Develop a project combines various hardware and software
Design of ALU by VHDL
Using FPGA technology
Using method for education: ML
This technique and method of teaching were used in two semesters
Sem. 1: Design ALS and rotating unit (instructions) by VHDL. without using FPGA+ML
Sem. 2: Design ALS and rotating unit (instructions) by VHDL Using FPGA + ML
The best method of teaching was assessed on the following tests
Sem. 1: Questionnaire, Homeworks, Midterm exam. and Final exam.
Sem. 2: Questionnaire, Homeworks, Midterm exam. and Final exam.

Figure 1. The Organizational Card of the Proposed Teaching Method

Micro Learning Method

In the scope of micro-learning, the students are learning how to deal with the sub-blocks components and how to create the modules for each part and making productive modules in the sense of education. The ML deals with relatively small learning units and short-term learning activities (Alqurashi, 2017). It is important to add large examples into the computer architecture course plans by splitting them into the small sub blocks (Kiray and Zhaparov, 2013). Each sub-block must address certain topic of a computer architecture course, and these blocks must be formed in terms of education.

To reach this goal, ML appears in a better position as an educational approach which makes it very significant, particularly, when dealing with the sub-block component in the class to take some questions in consideration and having to answer them (Baptiste and Solomon, 2005). For example, if we intend to teach the design for arithmetic logical shifting (ALS) and rotating instructions (RIs), then we should present a small introduction including adequate information to the students, as Figure 2.

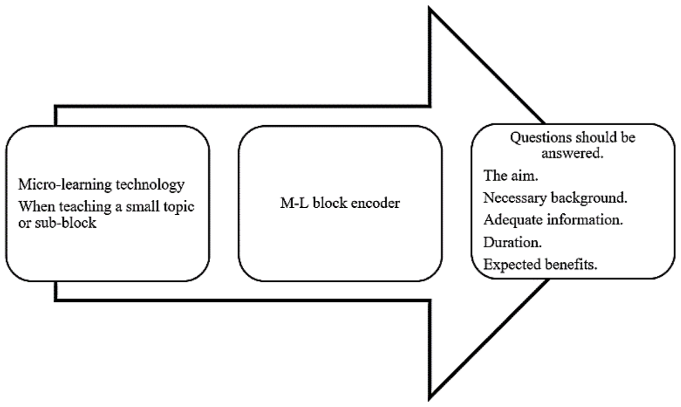

Micro-learning technology
When teaching a small topic or sub-block
M-L block encoder
Questions should be answered.
The aim.
Necessary background.
Adequate information.
Duration.
Expected benefits.

Figure 2. Micro-Learning Teaching Blocks

A new special designed example was prepared and established to teach VHDL. The example was associated with an imagined ALU in order to use all topics of the course. Consequently, the following questions should be taken into consideration:

- What is the aim of designing and implementing the ALS and Rotating Unit?
- What is the necessary background for the study of the ALS and Rotating unit?
- What sort of information that should be considered adequate, for example?
- What is the suitable method to design by VHDL?
- What are the expected questions from students?
- What is the time required for teaching this element?
- What are the things students should note?

The steps used to develop students understanding, three levels (V1-3) were identified in each lecture as given Table 1.

ISSN: 2689-5315
DOI: 10.52589/BJCNIT/m4UwNH4j

Table 1. Different Types of Levels

Code	Name of code	Type of code
V1	Version one	Homework I, as the title of the class.
V2	Version two	Homework II, between Version one and Version three, mid-project.
V3	Version three	Homework III, complex project.

According to the previous tests when using ML, access to the final design is easier than using PBL directly. Each title in each lecture was divided into three levels as easy to hard as shown in the Figure 3 below.

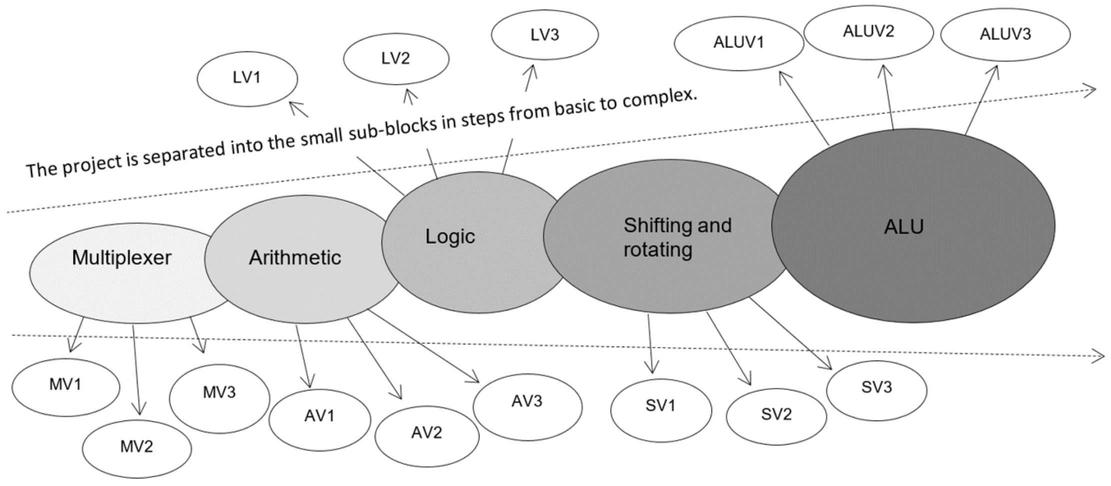


Figure 3. An Outlook of the Title Divided into Three Levels Using ML

The following table (Table 2) shows the meaning of each code.

Table 2. Codes Used in the Experiment

Code	Title Name
MV1	Multiplexer 4-1 by VHDL language.
MV2	Multiplexer 8-1 by VHDL language.
MV3	Multiplexer 16-1 by VHDL language.
AV1	4-Bit Design Arithmetic instructions by VHDL language.
AV2	8-Bit Design Arithmetic instructions by VHDL language.
AV3	16-Bit Design Arithmetic instructions by VHDL language. With a new operation design for example ADDC.
LV1	4-Bit Design logic instructions by VHDL language
LV2	8-Bit Design logic instructions by VHDL language
LV3	16-Bit Design logic instructions by VHDL language, With a new operation design for example NAND or NOR.
SV1	4-Bit Design shifting and RIs by VHDL language.

SV2	8-Bit Design shifting and RIs by VHDL language.
SV3	16-Bit Design shifting and RIs by VHDL language. With a new operation design for example SL by 2 bit.
ALUV1	Design 4-bit ALS and RIs by VHDL language.
ALUV2	Design 8-bit ALS and RIs by VHDL language.
ALUV3	Design 16-bit ALS and RIs by VHDL language. With a new operation design.

The design can start from easy to complex by using ML following the steps shown in the diagram. For example, from mux 4-1 to ALU 16-bit, as shown in Figure 4. You can follow any path in the design from start to end, as shown in Figure 4.

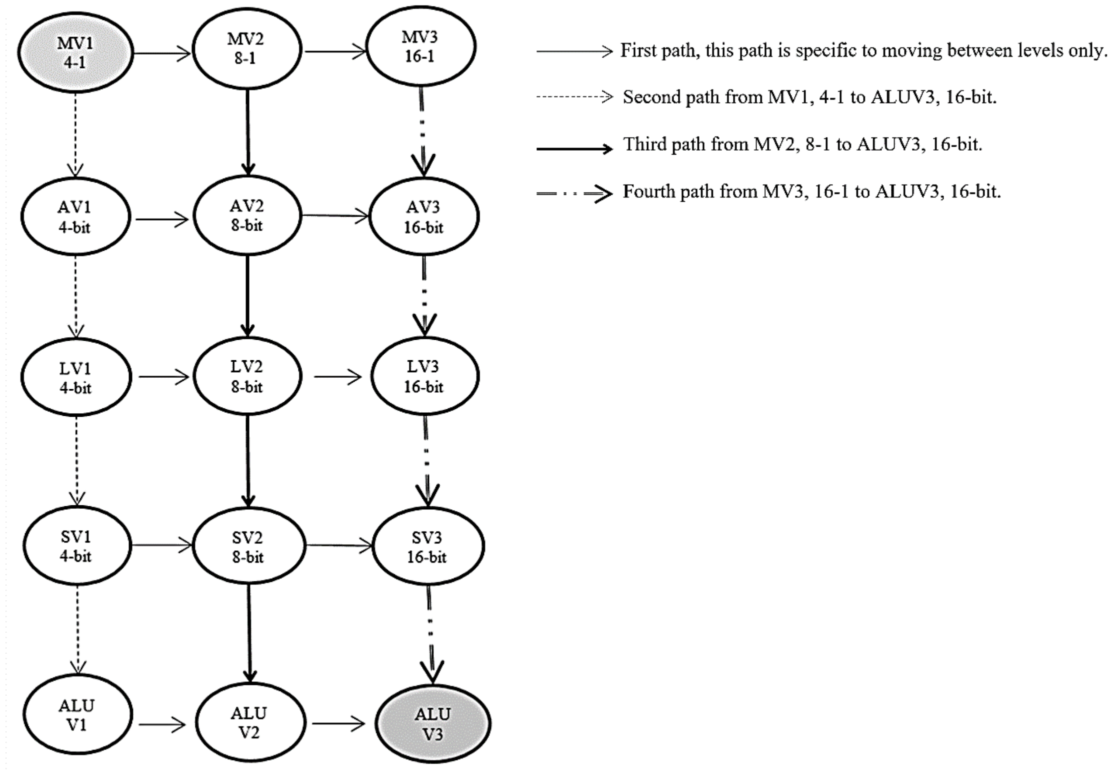

MV1 4-1
MV2 8-1
MV3 16-1
AV1 4-bit
AV2 8-bit
AV3 16-bit
LV1 4-bit
LV2 8-bit
LV3 16-bit
SV1 4-bit
SV2 8-bit
SV3 16-bit
ALU V1
ALU V2
ALU V3
First path, this path is specific to moving between levels only.
Second path from MV1, 4-1 to ALUV3, 16-bit.
Third path from MV2, 8-1 to ALUV3, 16-bit.
Fourth path from MV3, 16-1 to ALUV3, 16-bit.

Figure 4. Design Paths Using ML

Training Groups

- In this project, the methodology we used is a quantitative research (experimental approach). Moreover, in this project there are two groups as shown in Figure 5.
- The first group, which learns computer architecture course without applying FPGA technologies, and ML method) as shown in Figure 5-a. The second group, which applies FPGA technologies and ML method (studying attractive and instructive examples) as shown in Figure 5-b.

ISSN: 2689-5315
DOI: 10.52589/BJCNIT/m4UwNH4j

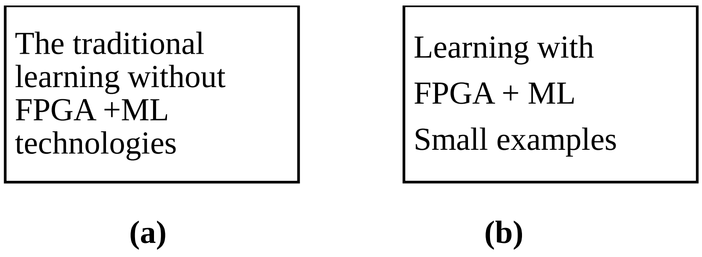

The traditional learning without FPGA +ML technologies
Learning with FPGA + ML Small examples
(a)
(b)

Figure 5. Groups of the Study

The project was divided into two semesters as shown in Figure 6.

Semester One: Teaching in this semester, design ALS and rotating unit by VHDL in the Computer Engineering Department by traditional learning without using FPGA+ ML technologies; in Spring Semester -2016. The number of students attending this class was 46 students, as illustrated in Figure 6.

Semester Two: Teaching in this semester, design ALS and rotating unit by VHDL use micro-learning method, in the Computer Engineering Department, using FPGA + ML technologies, in fall Semester -2016. The number of students attending this class was 43 students, as explained in Figure 6.

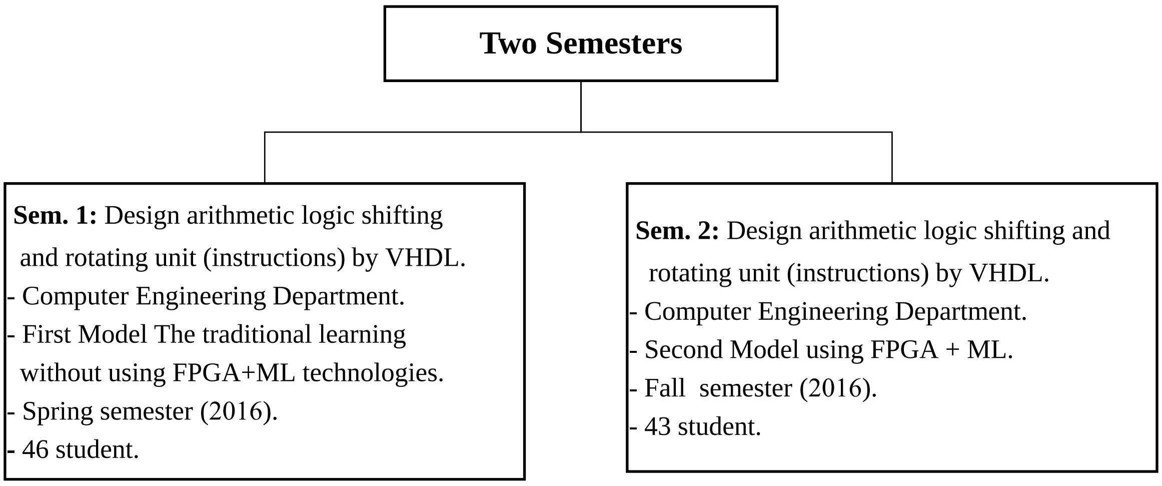

Two Semesters
Sem. 1: Design arithmetic logic shifting and rotating unit (instructions) by VHDL.
- Computer Engineering Department.
- First Model The traditional learning without using FPGA+ML technologies.
- Spring semester (2016).
- 46 student.
Sem. 2: Design arithmetic logic shifting and rotating unit (instructions) by VHDL.
- Computer Engineering Department.
- Second Model using FPGA + ML.
- Fall semester (2016).
- 43 student.

Figure 6. Contents of the two Semesters

The two methods were tested to as which were better in teaching, after the end of each semester, and that have to be decided by the use of questionnaire, homework, midterm exam, final exam and testing of the first and second groups.

It is important to add large examples into the computer architecture course plans by splitting them into small sub-blocks in a way that each sub-block must address certain topic of the computer architecture course. Furthermore, these blocks must be formed in term of

ISSN: 2689-5315
DOI: 10.52589/BJCNIT/m4UwNH4j

education, as shown in Figure 7, which illustrates how to access the final design stage using ML?

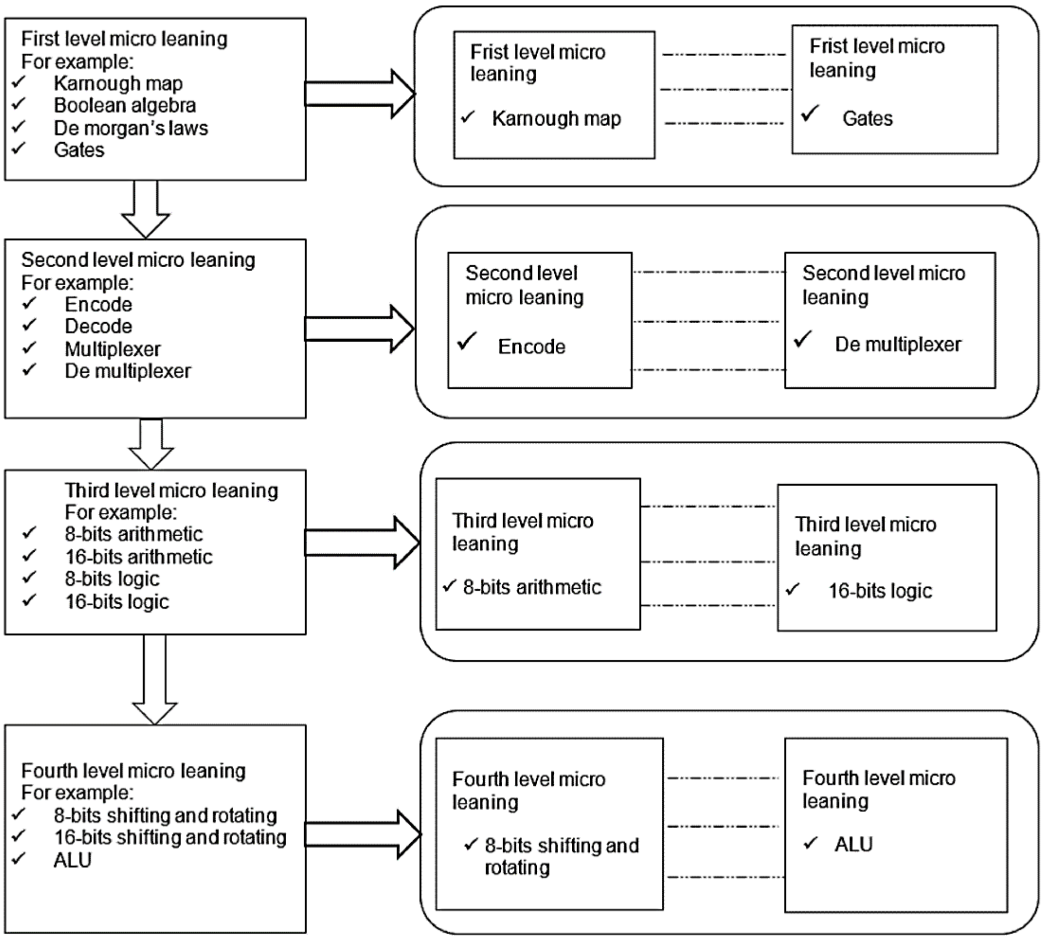

First level micro leaning
For example:
✓ Karnough map
✓ Boolean algebra
✓ De morgan's laws
✓ Gates
Frist level micro leaning
✓ Karnough map
Frist level micro leaning
✓ Gates
Second level micro leaning
For example:
✓ Encode
✓ Decode
✓ Multiplexer
✓ De multiplexer
Second level micro leaning
✓ Encode
Second level micro leaning
✓ De multiplexer
Third level micro leaning
For example:
✓ 8-bits arithmetic
✓ 16-bits arithmetic
✓ 8-bits logic
✓ 16-bits logic
Third level micro leaning
✓ 8-bits arithmetic
Third level micro leaning
✓ 16-bits logic
Fourth level micro leaning
For example:
✓ 8-bits shifting and rotating
✓ 16-bits shifting and rotating
✓ ALU
Fourth level micro leaning
✓ 8-bits shifting and rotating
Fourth level micro leaning
✓ ALU

Figure 7. The Final Design Phase Using ML

Each title of the curriculum has its own block with a specific code called ML block. The symbols of some syllabuses in courses is given in Table 3.

Table 3. Symbols of Some Syllabuses in Courses

The code	Example name	The code	Example name
T-ML-1	Number systems	T-ML-15	DE multiplexers
T-ML-2	Electronic signals and switches	P-ML-10	DE multiplexers
T-ML-3	Arithmetic operations and circuits	T-ML-16	Comparator
T-ML-4	Boolean algebra and reduction techniques	P-ML-11	Comparator
T-ML-5	Karnough map	T-ML-17	flip-flop
T-ML-6	Logic gates	P-ML-12	flip-flop
P-ML-1	Logic gates	T-ML-18	Registers

T-ML-7	shift Registers	P-ML-13	Registers
P-ML-2	shift Registers	T-ML-19	Counters
T-ML-8	Half adder	P-ML-14	Counters
P-ML-3	Half adder	T-ML-20	SM
T-ML-9	Full adder	P-ML-15	SM
P-ML-4	Full adder	T-ML-21	Arithmetic unit
T-ML-10	Half subtract	P-ML-16	Arithmetic unit
P-ML-5	Half subtract	T-ML-22	Logic unit
T-ML-11	Full subtract	P-ML-17	Logic unit
P-ML-6	Full subtract	T-ML-23	Shifting and rotating unit
T-ML-12	Encoder	P-ML-18	Shifting and rotating unit
P-ML-7	Encoder	T-ML-24	ALS and rotating unit
T-ML-13	Decoder	P-ML-19	ALS and rotating unit
P-ML-8	Decoder	T-ML-35	CPU
T-ML-14	Multiplexer	P-ML-30	CPU
P-ML-9	Multiplexer		

ALU Design by VHDL for Engineering Education and Simulation

We made this experiment according to the requirements of the courses. This experiment was divided into a number of parts according to each lecture. Each lecture was divided into three levels from simple to harder (V1, V2, V3). **V1**: Homework I, as the title of the class. **V2**: Homework II, between Version one and Version three, mid-project. **V3**: Homework III, complex project.

Each main title in the curriculum had its own table, showing the necessary things for the student, the name of these tables, the ML table. This table was given to the student before the beginning of the semester. Likewise, the first lecture on how to deal with the levels (V1, V2, and V3) was provided to the students.

Design arithmetic logical shifting and rotating ınstructions

ALS and rotating unit are given in Figure 8.

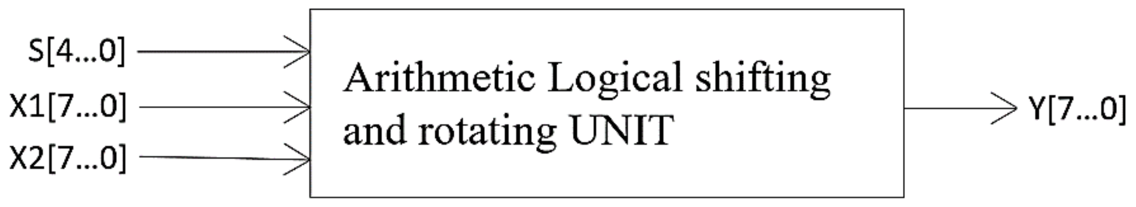


Figure 8. ALS and Rotating Unit

ML table, showing ALS and rotating unit.

Table 4. (T-ML-24) Micro Learning Table of ALS and Rotating Unit

Name	**The Code**	**Learning Duration**	**Needed Background**	**Content/ Curriculum**	**Place in course**	**Place of the MLB in Course Projects**	**Video Name/Time Use Animations**
ALS and Rotating Unit	T-ML-24	30 min	T1,T3,T4,T5,T6,T7,T8,T9, T10,T11,T14, P1,P2,P3,P4, P5,P6,T18,T21,T22,T23, P9,P13,P16,P17,P18	Theoretical & practical	Second week 1:30 to 2:00 pm	T35,P30	Arithmetic unit/2 min
Adequate Information	▪ All the arithmetic operation of microprocessor take place in the arithmetic logic unit (ALSRU). ▪ Using combination of gates and VHDL. ▪ The operation to be performed is specified by signals from the control unit. ▪ The date upon which operations are performed can come from memory or an external input. ▪ The data may be combined in some way with the contents of the accumulator and the results are typically placed in the accumulator, from there they may be transferred to memory or to an output unit.						
Questions	▪ What is the aim to design and implement the ALS and rotating unit? ▪ What is the necessary background to study the ALS and rotating unit? ▪ What is the suitable method to design by VHDL? ▪ How can you design 16-bits ALS and rotating unit? ▪ How can you design other part for example mixed part (zero & ones)?						

Name: Name of the experiment

The code : Each experience has a code unique and belongs to that particular one.

Learning duration: Each experiment has a specific time by the Version one.

Needed background: Before teaching any experience, you should understand some subjects because of time boundary.

Content/curriculum: Each experiment is taught both theoretically and practically, so that we can apply the ML method effortlessly.

Place in course: Each experiment has its time course; meaning in which time and week the experiment should be performed.

Place of the MLB in the final project: Each experience has a place in the complex project.

Video name/time Use animations: In each experiment, a two-minute video is used to help students understand how the complex project income experiment works.

Table for opcode ALS and RIs, as revealed in Table 5.

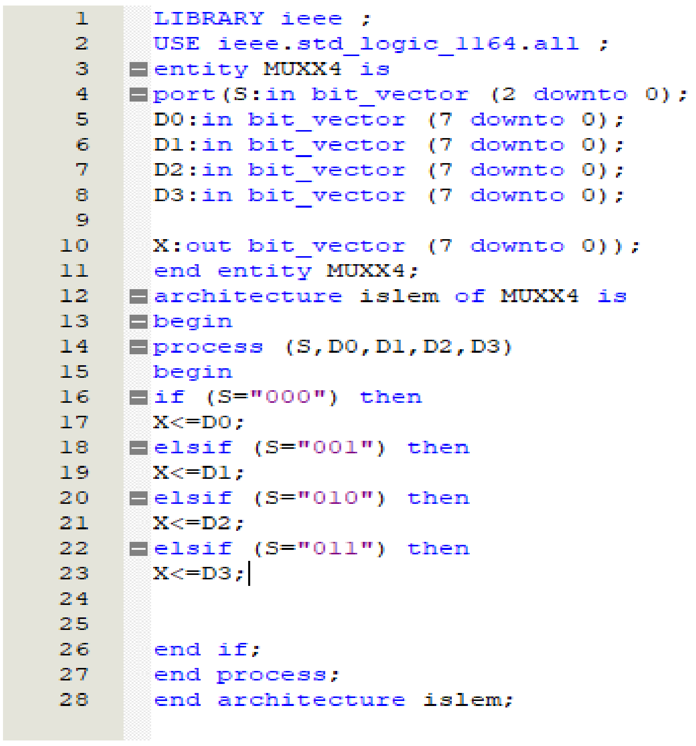
Table 5. Opcode ALS and RIs.

Instruction	S0	S1	S2	S3	S4	Type of instructions
ADD	0	0	0	0	0	Arithmetic instructions
SUB	1	0	0	0	0	
INC	0	1	0	0	0	
DEC	1	1	0	0	0	
AND	0	0	1	0	0	Logical instructions
OR	1	0	1	0	0	
XOR	0	1	1	0	0	
NOT	1	1	1	0	0	
SL	0	0	0	1	0	Shifting and RIs
SR	1	0	0	1	0	
RL	0	1	0	1	0	
RR	1	1	0	1	0	

- When $S2 = S3 = S4 = 0$ arithmetic unit is working, and when $S0 = S1 = 0$, **ADD** operation is working.
- When $S2 = S3 = S4 = 0$ arithmetic unit is working, and when $S0 = 1, S1 = 0$, **SUB** operation is working.
- When $S2 = 1, S3 = S4 = 0$ logical unit is working, and when $S0 = S1 = 0$, **AND** operation is working.
- When $S2 = 1, S3 = S4 = 0$ logical unit is working, and when $S0 = 1, S1 = 0$, **OR** operation is working.
- When $S2 = S4 = 0, S3 = 1$ shifting and rotating unit is working, and when $S0 = S1 = 0$, **SL** operation is working.
- When $S2 = S4 = 0, S3 = 1$ shifting and rotating unit is working, and when $S0 = 1, S1 = 0$, **SR** operation is working.

ISSN: 2689-5315
DOI: 10.52589/BJCNIT/m4UwNH4j

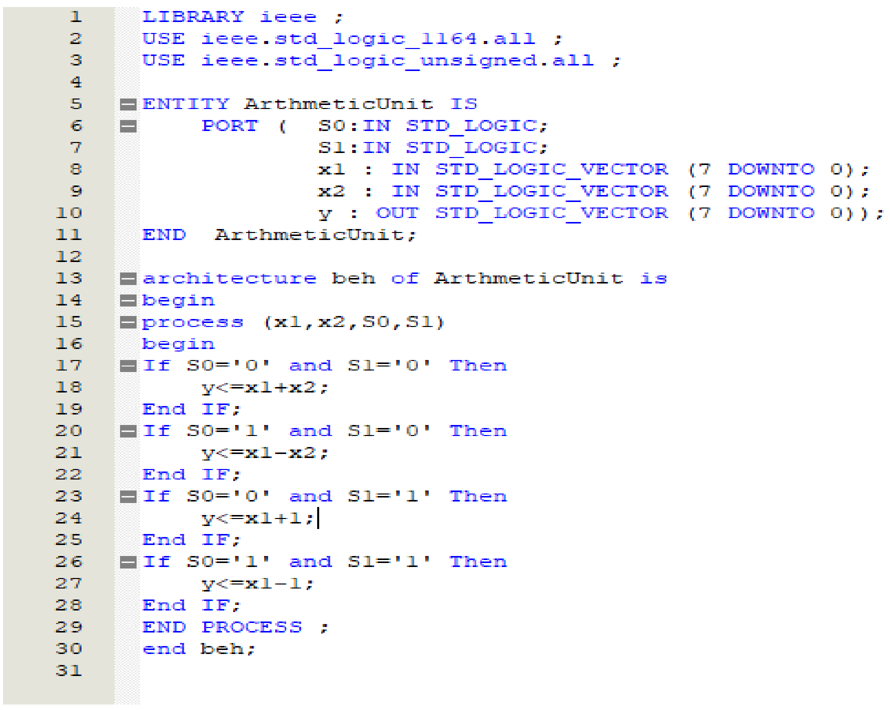
Design multiplexer 4_1 by VHDL is given in the Figure 9.



```vhdl
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;
entity MUXX4 is
port(S:in bit_vector (2 downto 0);
D0:in bit_vector (7 downto 0);
D1:in bit_vector (7 downto 0);
D2:in bit_vector (7 downto 0);
D3:in bit_vector (7 downto 0);

X:out bit_vector (7 downto 0));
end entity MUXX4;
architecture islem of MUXX4 is
begin
process (S,D0,D1,D2,D3)
begin
if (S="000") then
X<=D0;
elsif (S="001") then
X<=D1;
elsif (S="010") then
X<=D2;
elsif (S="011") then
X<=D3;


end if;
end process;
end architecture islem;
```

Figure 9. Multiplexer 4_1 by VHDL

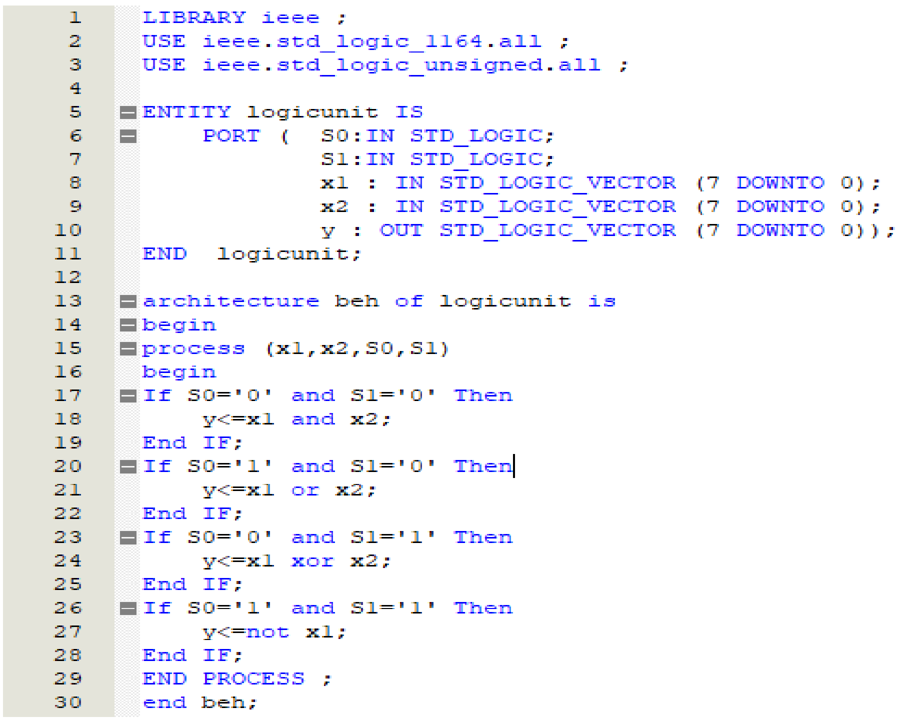
Design 8-bit Arithmetic instructions by VHDL is given in the Figure 10.



```vhdl
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;
USE ieee.std_logic_unsigned.all ;

ENTITY ArthmeticUnit IS
    PORT (  S0:IN STD_LOGIC;
            S1:IN STD_LOGIC;
            x1 : IN STD_LOGIC_VECTOR (7 DOWNTO 0);
            x2 : IN STD_LOGIC_VECTOR (7 DOWNTO 0);
            y : OUT STD_LOGIC_VECTOR (7 DOWNTO 0));
END  ArthmeticUnit;

architecture beh of ArthmeticUnit is
begin
process (x1,x2,S0,S1)
begin
If S0='0' and S1='0' Then
    y<=x1+x2;
End IF;
If S0='1' and S1='0' Then
    y<=x1-x2;
End IF;
If S0='0' and S1='1' Then
    y<=x1+1;
End IF;
If S0='1' and S1='1' Then
    y<=x1-1;
End IF;
END PROCESS ;
end beh;

```

Figure 10. Arithmetic instructions by VHDL

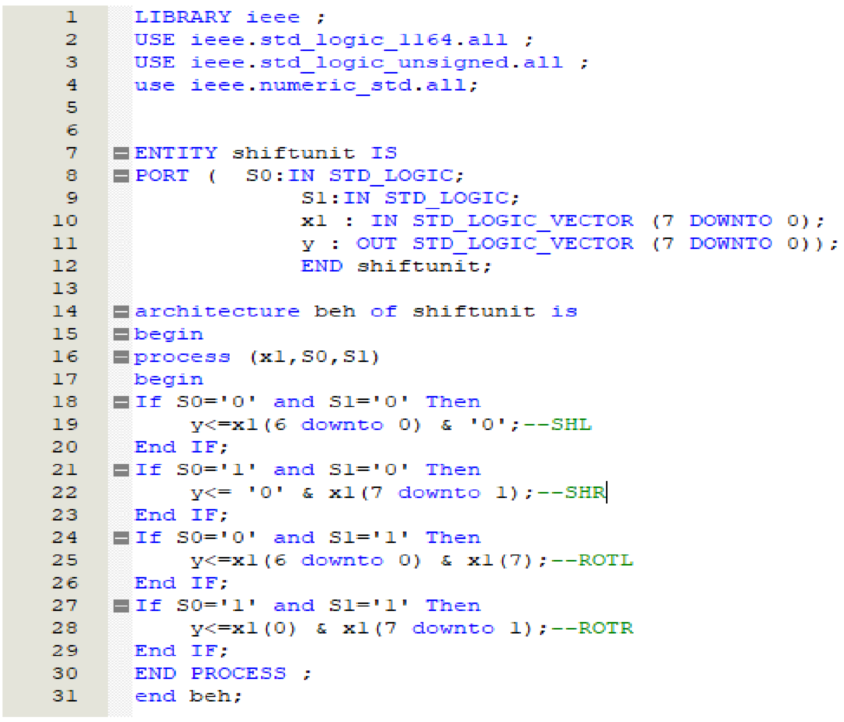
Design 8-bit Logical instructions by VHDL is given in the Figure 11.



```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;
USE ieee.std_logic_unsigned.all ;

ENTITY logicunit IS
    PORT (   S0:IN STD_LOGIC;
             S1:IN STD_LOGIC;
               x1  : IN STD_LOGIC_VECTOR (7 DOWNTO 0);
               x2  : IN STD_LOGIC_VECTOR (7 DOWNTO 0);
               y  : OUT STD_LOGIC_VECTOR (7 DOWNTO 0));
END  logicunit;

architecture beh of logicunit is
begin
process (x1,x2,S0,S1)
begin
If S0='0' and S1='0' Then
    y<=x1 and x2;
End IF;
If S0='1' and S1='0' Then
    y<=x1 or x2;
End IF;
If S0='0' and S1='1' Then
    y<=x1 xor x2;
End IF;
If S0='1' and S1='1' Then
    y<=not x1;
End IF;
END PROCESS ;
end beh;
```

Figure 11. Logical Instructions by VHDL

Design 8-bit Shifting and RIs by VHDL is given in the Figure 12.



```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;
USE ieee.std_logic_unsigned.all ;
use ieee.numeric_std.all;


ENTITY shiftunit IS
PORT (  S0:IN STD_LOGIC;
           S1:IN STD_LOGIC;
           x1 : IN STD_LOGIC_VECTOR (7 DOWNTO 0);
           y : OUT STD_LOGIC_VECTOR (7 DOWNTO 0));
           END shiftunit;

architecture beh of shiftunit is
begin
process (x1,S0,S1)
begin
If S0='0' and S1='0' Then
    y<=x1(6 downto 0) & '0';--SHL
End IF;
If S0='1' and S1='0' Then
    y<= '0' & x1(7 downto 1);--SHR
End IF;
If S0='0' and S1='1' Then
    y<=x1(6 downto 0) & x1(7);--ROTL
End IF;
If S0='1' and S1='1' Then
    y<=x1(0) & x1(7 downto 1);--ROTR
End IF;
END PROCESS ;
end beh;
```

Figure 12. Shifting and RIs by VHDL.

Design of an ALS and RIs is visualized in Figure 13.

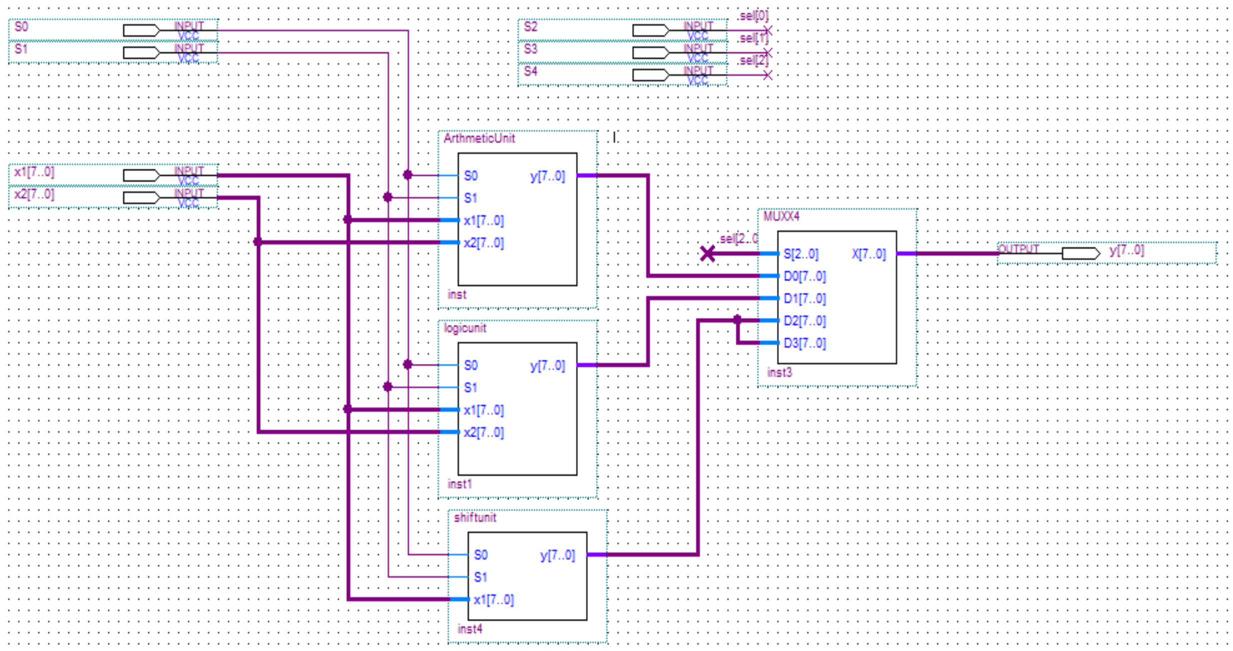


Figure 13. (P-ML-19) Design of an ALS and RIs

The simulation for ALS and RIs are given in Figure 14.

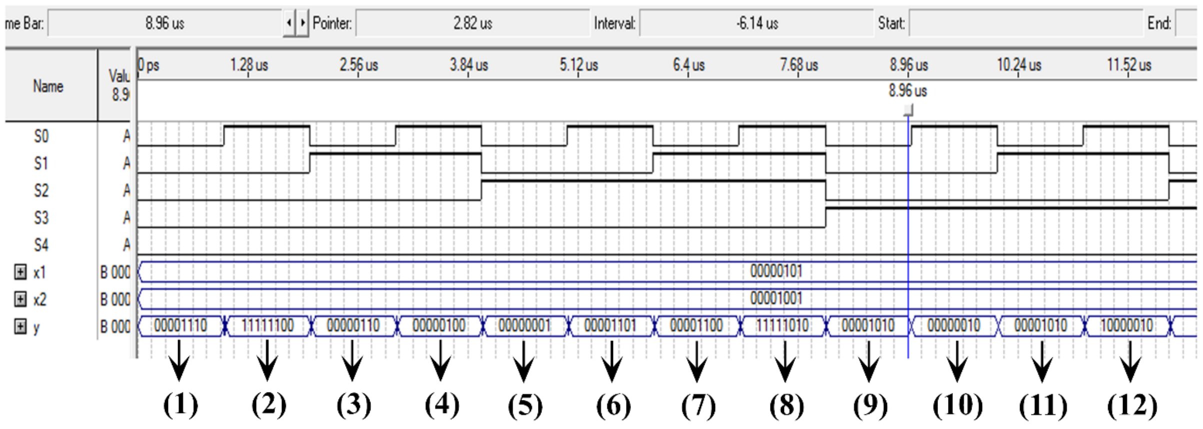


Figure 14. Simulation for ALS and RIs

Explanation of some tests are given as follow:

1- Number **(1)**, when $S0 = S1 = S2 = S3 = S4 = 0$, means that the operation is **ADD**, and when also there are in, $X1 = 00000101, X2 = 00001001$, then the result is $Y = 00001110$.

ISSN: 2689-5315
DOI: 10.52589/BJCNIT/m4UwNH4j

2- Number **(3)**, when $S1 = 1, S0 = S2 = S3 = S4 = 0$, means that the operation is **INC**, and when also there are in, $X1 = 00000101, X2 = 00001001$, then the result is $Y = 00000110$.

3- Number **(5)**, when $S2 = 1, S0 = S1 = S3 = S4 = 0$, means that the operation is **AND**, and when also there are in, $X1 = 00000101, X2 = 00001001$, then the result is $Y = 00000001$.

4- Number **(10)**, when $S0 = S3 = 1, S1 = S2 = S4 = 0$, means that the operation is **SR**, and when also there are in, $X1 = 00000101$, then the result is $Y = 00000010$.

5- Number **(12)**, when $S0 = S1 = S3 = 1, S2 = S4 = 0$, means that the operation is **RR**, and when also there are in, $X1 = 00000101$, then the result is $Y = 10000010$.

RESULTS AND DISCUSSION

The use of IBM SPSS Statistics 22 for calculating the Cronbach's alpha is the evidence of internal consistency reliability (Morgan, Leech, Gloeckner and Barrett, 2004). There is no standard cut-off point for the alpha coefficient, but the generally agreed upon lower limit for Cronbach's alpha is 0.7 (Nunnally and Bernstein, 1999). The tool quality achieved if Cronbach's alpha coefficient increased from 0.6 and above Researchers was advised to be 0.6 as the satisfactory cutoff for the Cronbach's alpha criterion (Evanschitzky, Iyer, Plassmann, Niessing and Meffert, 2006).

Testing our questionnaire for reliability is demonstrated a reliable internal consistency as shown in the table below.

Table 6. Illustration of the Results of Cronbach's alpha.

Reliability Statistics		
Cronbach's Alpha	Cronbach's Alpha based on standardized items	N of items
0.706	0.699	24

The results of the tests in this study were obtained from the questionnaire, homework's, midterm exam, and the final exam.

Based on tests and results of this study, the differences between traditional learning without using FPGA+ML and with using FPGA+ML is noticeably large. The main advantage of this test is that the two methods of teaching with FPGA + ML are evidently better from teaching without FPGA +ML.

Analyzing the data has clearly shown that the ML method has facilitated complexities in the projects. This has led to further interest and attention and thereby accelerating the learning process., as Figure 15.

ISSN: 2689-5315
DOI: 10.52589/BJCNIT/m4UwNH4j

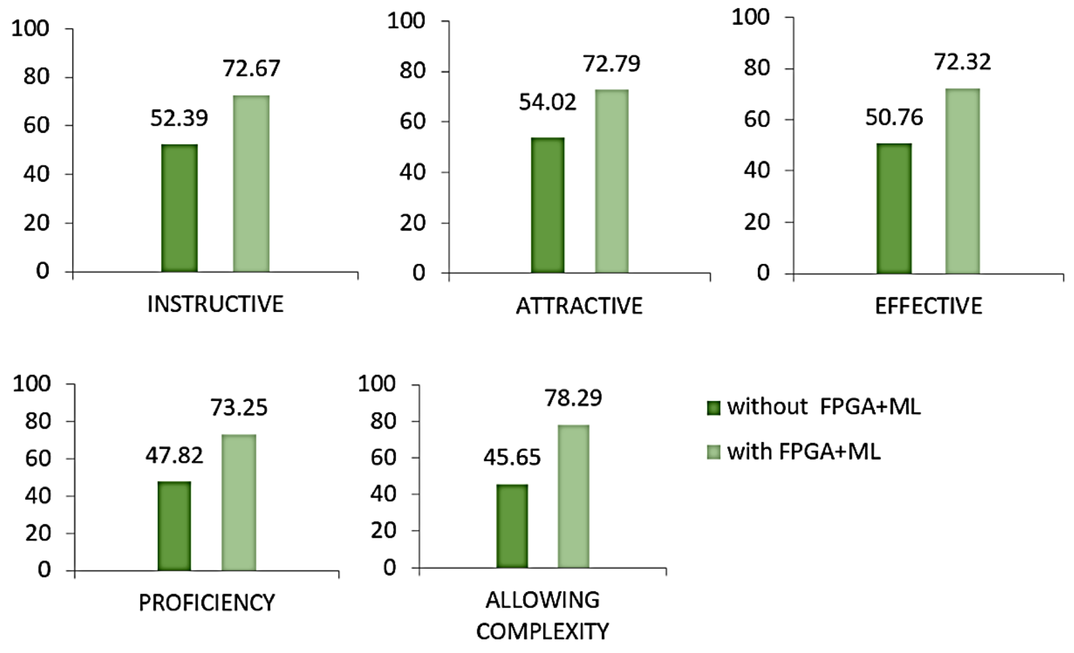


Figure 15. Statistical Data of for Factors.

In homework's, as the complexity of the project increased from Homework I to Homework III, simple to complex, it is advisable to use ML, as Figure 16.

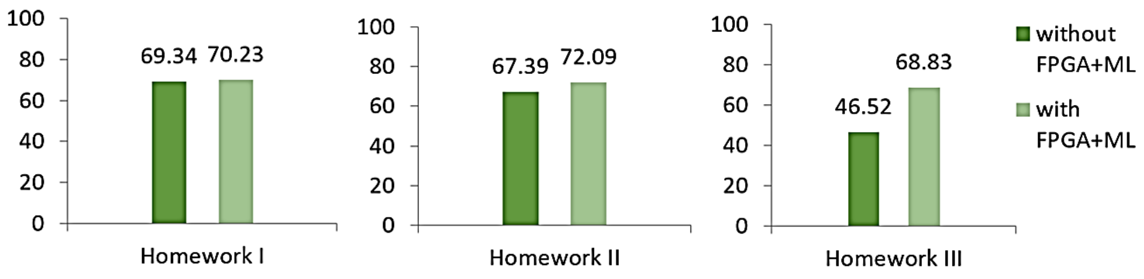


Figure 16. Statistical data of for homework's.

In the mid test and final exam, the ratio was not significant but it was the best when using ML methods and it was definitely higher than the traditional learning method as Figure 17.

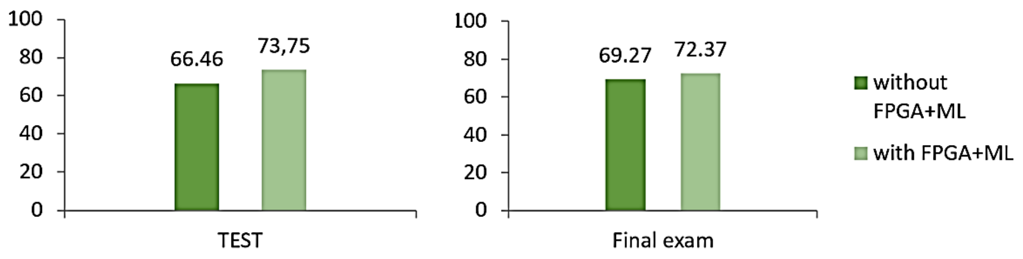


Figure 17. Statistical data of for mid exam and the final exam.

Differences between teaching methods by percentages as Table 7.

Table 7. Results of the Comparison of Teaching Methods.

Type of Test		**Traditional learning without FPGA+ML**	**with FPGA+ML**
Factors	Instructive	52.39	72.67
	Attractive	54.02	72.79
	Effective	50.76	72.32
	Proficiency	47.82	73.25
	Allowing complexity	45.65	78.29
Homework	Homework I	69.34	70.23
	Homework II	67.39	72.09
	Homework III	46.52	68.83
Test		66.46	73.75
Final exam		69.27	72.37

CONCLUSION

In this study, a simple and effective approach was presented to enable the students to properly utilize the computer architecture in the education area. The ALU design was employed for improving students' abilities to use ALS and RIs.

As a result of the study, utilizing the FPGA technology, ML, and associated examples and projects were demonstrated an effective, fast, and easier approach than the traditional education methods. In addition, the learning process had become increasingly effective, attractive, and instructive.

Moreover, the performance of the students with using FPGA technologies, micro-learning methods, is improved and the students' response was faster than the traditional learning methods. This result was observed through students' answers of the questionnaire particularly for the questions that were related the Allowing Complexity factor. Furthermore, learning with FPGA approach is important for both engineers and students. It develops their knowledge learning capabilities and enables them to use time efficiently.

To this end, employing the FPGA with ML facilitates the complex projects design in short time period. However, learning this technology without ML is perceived insufficient.

REFERENCES

Alqurashi, E. (2017), Microlearning: A pedagogical approach for technology integration. Turkish Online J. Educ. Technol.

Baptiste, S., & Solomon, P. (2005), Curriculum development and design, in Innovations in Rehabilitation Sciences Education. Preparing Leaders for the Future.

Evanschitzky, H., Iyer, G. R., Plassmann, H., Niessing, J., & Meffert, H. (2006), The relative strength of affective commitment in securing loyalty in service relationships. J. Bus. Res.

Job, M. A., & Ogalo, H. S. (2012), Micro learning as innovative process of knowledge strategy. Int. J. Sci. Technol. Res.

Kiray, V., & Zhaparov, M. K. (2013), Nauka elektroniki cyfrowej z wykorzystaniem układu FPGA - projektowanie układu zegarka i kalendarza. Prz. Elektrotechniczny.

Kiray, V., Demir, S., & Zhaparov, M. (2013). Improving Digital Electronics Education with FPGA technology, PBL and Micro Learning methods. in Proceedings of IEEE International Conference on Teaching, Assessment and Learning for Engineering, TALE 2013, 2013.

Monmasson, E., & Cirstea, M. N. (2007), {FPGA} {Design} {Methodology} for {Industrial} {Control} {Systems}—{A} {Review}. IEEE Trans. Ind. Electron.

Morgan, G., Leech, N., Gloeckner, G., & Barrett, K. (2004), IBM SPSS for introductory statistics. Use and interpretation.

Nunnally, J. C., & Bernstein, I. H. (1999), Psychometric theory (3rd Ed.), New York: McGraw-Hill. Journal of Psychoeducational Assessment.

Shir, N., Bidabadi, A., Isfahani, A. N., Rouhollahi, A., Khalili, R., & Bidabadi, N. S. (2016), hirani Bidabadi N et al. Effective teaching method in higher education Effective teaching methods in higher education requirements and barriers.

Soyler, E., Gunaratne, C., & Akbas, M. I. (2017), Advances in Applied Digital Human Modeling and Simulation. in Advances in Intelligent Systems and Computing.

Tremblay, K., Lalancette, D., Roseveare, D., Dias, D., & Amaral, A. (2013), Assessment of Higher Education learning outcomes (AHELO): Feasibility Study. Feasibility study Rep.

Zhamanov, A., & Zhamapor, M. (2013), Computer networks teaching by microlearning principles. in Journal of Physics: Conference Series.