REPUBLIC OF TURKEY
ULUDAĞ UNIVERSITY
FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

PROCESSOR DESIGN

HÜSEYİN KAYA
031411503

ENGINEERING DESIGN I

BURSA 2019

REPUBLIC OF TURKEY
ULUDAĞ UNIVERSITY
FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

PROCESSOR DESIGN

HÜSEYİN KAYA
031411503

Project Advisor: Dr. GÖKHAN YENİKAYA

In this Engineering Design I study, which I have prepared in accordance with the thesis writing rules of the Faculty of Engineering of Uludağ University;

- That we obtained all the information and documents in this thesis within the framework of academic rules,
- That we presented all visual, auditory and written information and results in accordance with the rules of scientific ethics,
- That, where the works of others have been used, we cited the relevant works in accordance with scientific norms,
- That we listed all of the cited works as references,
- That we did not make any falsification in the data used,
- That we did not present any part of this thesis as another thesis study at our university or at any other university,

we hereby declare.

03.01.2019

Hüseyin Kaya

This Graduation Project study, prepared under my supervision, has been reviewed by me.

0 3.01.2019

Dr. Gökhan Yenikaya

SUMMARY

To design a 4-bit processor and, if feasible, to implement it. First of all, I started this project by designing the basic things, such as the ALU, because the ALU design is simple and the operations it can perform are limited — addition, subtraction, multiplication and other arithmetic operations — and as a result of this I obtained the necessary control signals; that was my purpose in starting from there.

And as a result, a simple processor architecture that runs at low frequencies but is functional has been put forward.

ABSTRACT

Design a 4 bit processor and do it if feasible. I started by designing the basic things in this project. Like the ALU, because the ALU design is simple and the operations I can do are limited addition, subtraction, multiplication and other arithmetic operations, and I have obtained the necessary control signals as a result of which I did not start.

And as a result, a simple but low-functioning processor architecture has been put forward.

LIST OF FIGURES

Page No

Figure 1.
Processors…… 9

Figure 2.
Semiconductors…… .11

Figure 3. P-N doping
process…… 12

Figure 4. N-type
semiconductor…… 13

Figure 5. P-type semiconductor……
14

Figure 6.
Diodes…… 15

Figure 7.
Transistors…… 16

Figure 8. Operation of transistors……
17

Figure 9. Integrated circuits…… 19

Figure 10. NAND
gate…… 20

Figure 11. AND
gate…… 20

Figure 12. NOR
gate…… 20

Figure 13. OR
gate…… 20

Figure 14. NOT
gate…… 20

Figure 15. XOR
gate…… 20

Figure 16. Logic gate truth
table…… 20

Figure 17. Delay in integrated circuits…… 21

Figure 18. Karnaugh map example…… 23

Figure 19. Half adder…… 24

Figure 20. Full adder…… 25

Figure 21. D flip-flop…… 26

Figure 22. 4-bit register design…… 26

Figure 23. Multisim ALU design…… 29

Figure 24. Multisim CU design…… 31

Figure 25. Multisim CU delay…… 32

Figure 26. Overview of the Quartus program…… 34

Figure 27. CPU Verilog…… 35

Figure 28. ALU Verilog…… 36

Figure 29. CPU RTL…… 37

Figure 30. ALU RTL…… 37

Figure 31. ALTERA SIM simulation…… 38

LIST OF TABLES

Page No

Table 1. Comparison of TTL and CMOS…… 19

TABLE OF CONTENTS

SUMMARY

ABSTRACT

LIST OF FIGURES

LIST OF TABLES

1.1 WHAT IS A PROCESSOR?.....9
1.2 HISTORY OF PROCESSORS.....10

2.1 SEMICONDUCTORS.....11
2.2 THE DOPING PROCESS.....12
2.3 N-TYPE SEMICONDUCTOR.....13
2.4 P-TYPE SEMICONDUCTOR.....14

3.1 DIODES.....15
3.2 STRUCTURE OF TRANSISTORS.....16
3.3 OPERATION OF TRANSISTORS.....17
3.4 DEFINITION AND CLASSIFICATION OF INTEGRATED CIRCUITS.....18
3.5 BY LEVEL OF INTEGRATION.....18
3.6 COMPARISON OF CMOS AND TTL ICs.....19
3.7 TTL INTEGRATED CIRCUITS.....19
3.8 CMOS INTEGRATED CIRCUITS.....19
3.9 LOGIC GATES.....20
3.10 DELAY AND LOGIC LEVELS IN ICs.....21
3.11 BOOLEAN ALGEBRA.....22
3.12 KARNAUGH MAP.....23
3.13 HALF ADDER (LOGIC CIRCUITS).....24
3.14 FULL ADDER (LOGIC CIRCUITS).....25
3.15 REGISTERS.....26

4.1 CPU HARDWARE.....27
4.2 ALU (ARITHMETIC LOGIC UNIT).....28
4.3 ALU DESIGN RESULTS.....29
4.4 CU (CONTROL UNIT).....30
4.5 CU DESIGN RESULTS.....31
4.6 CU TIMING ERROR RESULTS.....32
4.7 CREATING AN OPCODE LIBRARY.....33

5.1 OVERVIEW OF THE QUARTUS PROGRAM.....34
5.2 CPU VERILOG DESIGN.....35
5.3 ALU VERILOG DESIGN.....36
5.4 CPU RTL VIEWER.....37
5.5 ALTERA SIM SIMULATION RESULTS.....38

6. REFERENCES.....39

7. ACKNOWLEDGEMENTS.....40

8. CURRICULUM VITAE.....41

1.1 WHAT IS A PROCESSOR?

The processor is, in short, the brain of the computer. That is, just as the brain manages a human being and ensures that the organs communicate and work in an orderly manner, the processor manages the computer and ensures that its components work in an orderly manner. There are different names used for the processor, such as CPU, microprocessor and MİB (the Turkish acronym for central processing unit). In fact, they all mean the same thing. Some people, when doing research on search engines, type "what is a processor" into the search box. But in any case, search engines lead us to the right results on the subject.

The word CPU is formed by taking and combining the initials of the words Central Processing Unit. Central Processing Unit means "merkezi işlem birimi" in Turkish. As you can understand, the word MİB is likewise formed from the initials of these Turkish words.

Microprocessors consist of millions of transistors that work like on/off switches. Depending on how these switches are programmed, electrical signals flow through them. It is these signals that allow everything the computer does to be reduced to mathematical

operations. In addition, the processor uses the simplest counting system, the binary system — that is, the numbers 0 and 1.

Figure-1

1.2 HISTORY OF PROCESSORS

1971 – Intel 4004, the first single-chip CPU; clock speed 740 kHz, 4-bit data, instructions 8 bits long. Transistor count 2300, 16 pins, 4 KB of program memory, 640 bytes of addressable memory.

1972 – TMS 1000, the first microprocessor containing sufficient memory and enough space for a ROM program.

1972 – Intel 8008, clock speed 0.5 MHz–0.8 MHz, 8-bit bus width, 14-bit address bus, 16 KB of memory.

1974 – Intel 8080, clock speed 2 MHz, 8-bit bus width, 16-bit address bus, 64 KB of memory; the first processor used for home computers.

1975 – Motorola 6800.

1976 – Intel 8085, clock speed 3.5–6 MHz, 8-bit bus width, 16-bit address bus, 8-bit data bus, 64 KB of memory, 40 pins.

1978 – Intel 8086, clock speed 5–10 MHz, 16-bit bus width.

After the Intel 8086 came the Intel 8088, 80186, 80188, 80286, 80386 and 80486. Then, in 1993, Pentium processors began to be produced with the Pentium architecture. After the Pentium, Intel started producing Core and Celeron processors. Today, Intel has introduced its 2nd-generation processors featuring Sandy Bridge technology.

AMD, Intel's strongest rival, released its first processor, the AM386, in 1991. In fact, although it could have released this product earlier, its contract dispute with Intel delayed it; when AMD finally won the court case, this product also went on sale. The AM386 was followed by processors such as the AM486, AM5x86, K5, K6, K6-2, Athlon, Phenom, Turion and Opteron.

2.1 SEMICONDUCTORS

Atoms found in nature are divided into three groups — conductors, insulators and semiconductors — according to whether or not they conduct electricity. Iron, copper and gold can be given as examples of conductive materials; wood, air and pure water as examples of insulating materials. Two important elements that come to mind when speaking of semiconductor materials are silicon and germanium.

In terms of conductivity, they lie between conductors and insulators; in their normal state they are insulators. However, when they are exposed to heat, light or a magnetic effect, or when a voltage is applied, some of their valence electrons become free — that is, they gain conductive properties. This gain in conductivity is temporary; once the external effect is removed, the electrons return to their atoms. They are found in nature as simple elements and can also be obtained in the laboratory as compound elements. Semiconductors have a crystal structure; that is, their atoms are arranged in a specific order called a cubic lattice system. As mentioned above, such semiconductors can be made conductive to a certain degree by heat, light or applied voltage, and their conductivity can also be increased by adding certain special substances into them. Semiconductors whose conductivity is increased with dopants have a special place in electronics.

Figure-2

2.2 THE DOPING PROCESS

The conductivity of silicon and germanium can be increased in a controlled way. To increase the

conductivity in a controlled way, a dopant is added to the pure semiconductor material. This process

is called “doping”. Increasing the number of current carriers (electrons or holes) increases the

material’s conductivity, while decreasing it increases the material’s resistance. As a result of either

doping process, an N-type or a P-type material is formed.

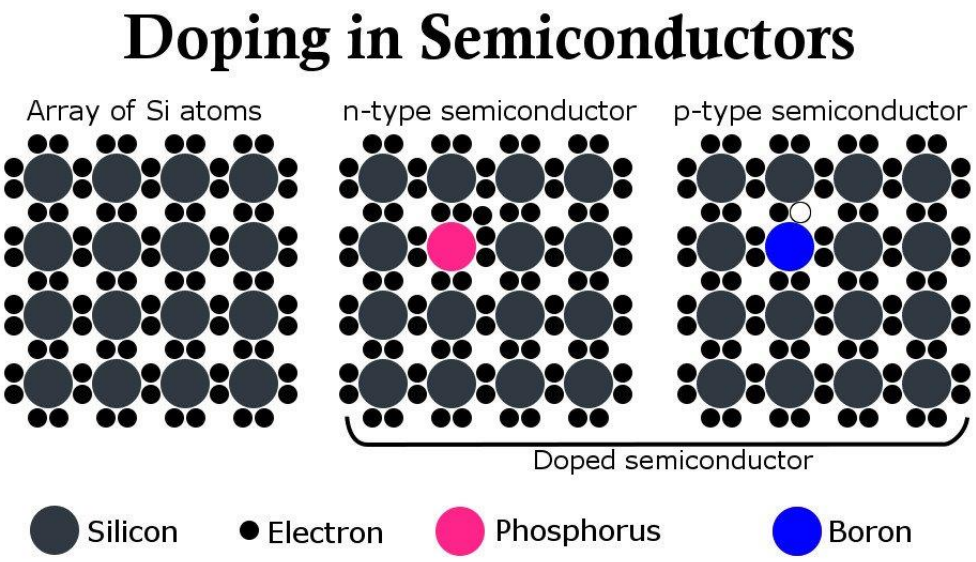
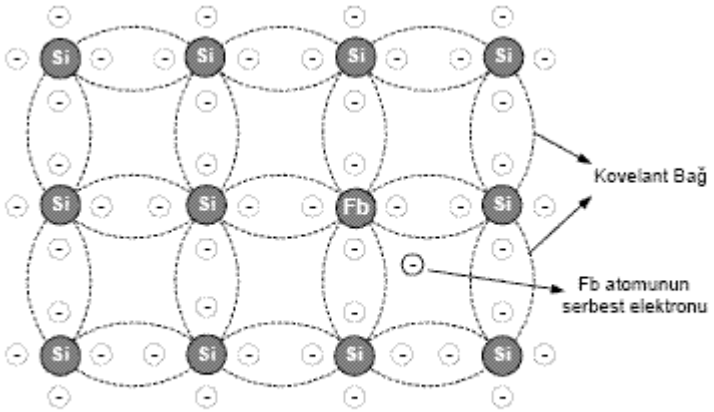


Figure-3

2.3 N-TYPE SEMICONDUCTOR

Increasing the holes in the conduction band of pure silicon is done by adding dopant atoms. These atoms are arsenic (As), phosphorus (P), bismuth (Bi) or antimony, which have 5 valence electrons. When phosphorus, which has 5 valence electrons, is added to silicon as a dopant in a certain proportion, the covalent bond it forms with the other silicon atoms is shown in Figure 1.4. Four valence electrons of the phosphorus atom form covalent bonds with 4 valence electrons of silicon. One valence electron of phosphorus is left over and breaks away. This free electron increases the conductivity, because it is not bound to any atom. Conductivity can be controlled by the number of electrons, and this is done through the number of atoms added to the silicon. This conduction electron created as a result of doping does not create a hole in the valence band.



Şekil-1.3 N tipi yarıiletken maddenin oluşturulması.

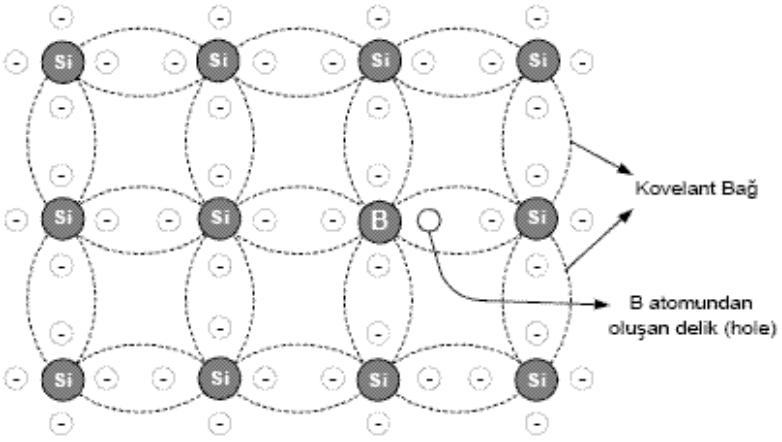
Figure-4

2.4 P-TYPE SEMICONDUCTOR

When atoms with 3 valence electrons (trivalent) are added into pure silicon in a certain proportion, a new crystal structure is formed. In this new crystal structure, the number of holes

is increased. As examples of atoms with 3 valence electrons we can give the elements aluminium (Al), boron (B) and gallium (Ga). For example, if boron is added into pure silicon in a certain proportion, the 3 valence electrons of the boron element form shared covalent bonds with 3 valence electrons of silicon. However, 1 valence electron of silicon cannot form a shared valence bond. In this case, a deficiency of 1 electron occurs. This is called a “hole”. The number of holes can be controlled by the amount of dopant added to the silicon. The new material obtained by this method is called P-type semiconductor material, because holes are positively charged. Therefore, in P-type material the majority current carriers are holes.

Electrons, on the other hand, are the minority current carriers in P-type material. A few free electrons have also formed in the P-type material. These came into being during the electron–hole pair generation caused by heat. These free electrons cannot be created during the doping of the silicon. Electrons are the minority current carriers in P-type material.



Şekil- 1.4 Silisyum kristaline 3 bağlı katkı atomu. Bohr katkı atomu merkezde gösterilmiştir.

Figure-5

3.1 DIODES

The semiconductor circuit element that emerges when a junction is formed between two materials, one N-type and one P-type, is the diode. This circuit element allows electric current to pass in one direction by providing a very low resistance, while blocking the passage of electric current in the other direction by showing a very high resistance. This characteristic structure has enabled its use in circuits where electric current is required to flow in only one direction.

In theory, an ideal diode can pass an infinite amount of current in one direction, while passing no current at all in the reverse direction. Moreover, if the smallest potential difference applied to an ideal diode is forward biased, the diode will immediately start conducting. On the other hand, the leakage current in the reverse direction is extremely small.

If a potential difference is applied across a semiconductor diode such that the P-type semiconductor material is the positive "+" terminal and the N-type semiconductor material is the negative "–" terminal, at a level exceeding the forward threshold voltage (0.6–0.7 V for silicon material, 0.2–0.3 V for germanium material), the diode freely starts to conduct current. In the opposite case — that is, if a voltage is applied in the reverse direction, with the negative terminal on the P-type material and the positive terminal on the N-type material (without exceeding the diode's reverse breakdown voltage) — no current flows through the diode. If the voltage applied to the diode exceeds this reverse breakdown voltage, we cause the diode to be damaged.

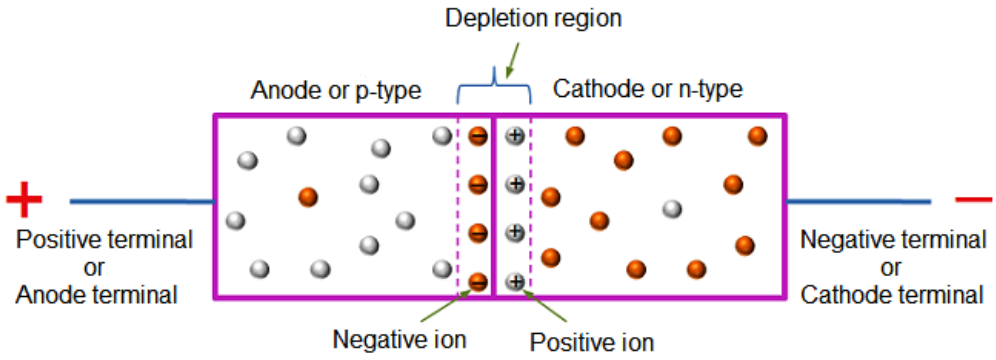


Figure-6

3.2 STRUCTURE OF TRANSISTORS

A transistor is an electronic circuit element used to control one current or voltage source with another voltage or current source. The transistor is a semiconductor electronic circuit element formed of PN diode junctions joined side by side, which provides current and voltage gain by amplifying the signal applied to its input, and which can be used as a switching element when required. The word transistor is derived from the combination of the words transfer and resistance. Thanks to the third terminal, created by adding a third layer (P-type or N-type) to the diode structure with a special technique, the external circuit currents can be controlled with this new element without changing the circuit load or the source voltage. The most widely used types are BJTs and FETs. BJTs operate with current, while FETs operate with the electric field created by voltage. Today, FETs are mostly used in integrated digital circuits. Transistors have three terminals: in a BJT transistor these are the Base, Emitter and Collector, while in FETs they are the Gate, Drain and Source.

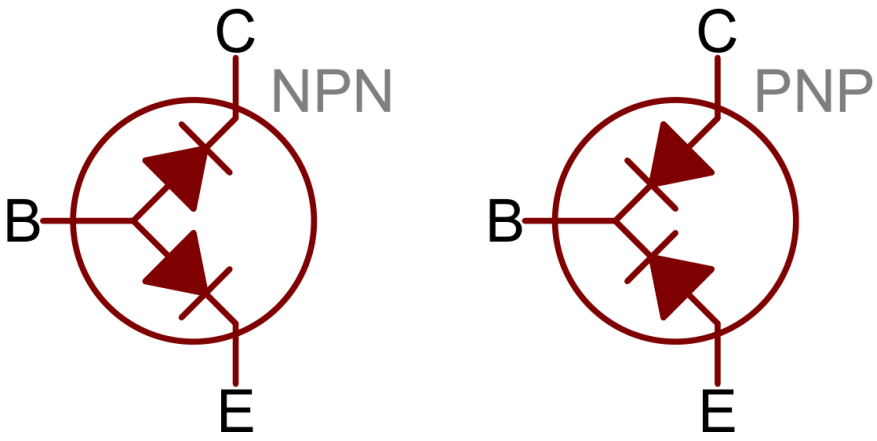


Figure-7

3.3 OPERATION OF TRANSISTORS

Whether it functions as a switch, an amplifier or an oscillator, every transistor operates based on a change in electrical resistance. When there is no Base current or Gate voltage, the resistance between the Collector and the Emitter (or between the Drain and the Source) is so high that almost no current can pass between these two terminals. But when a small current is applied to the Base terminal (or a small voltage to the Gate), there is a very large decrease in the resistance between the Collector and the Emitter (Drain and Source). Consequently, current can flow through it. In this way, a transistor can control a large current with the help of a small current or voltage.

When the transistor is used as a switch, applying a small current (a voltage in FETs) to its input terminal allows a strong electric current to complete its circuit. When used as an amplifier or an oscillator, it strengthens a weak signal. The weak signal is applied to the input (Base/Gate) in the form of a small electric current or voltage. This allows a large current to pass from the Collector to the Emitter (from the Source to the Drain). In this way, a strong signal is produced.

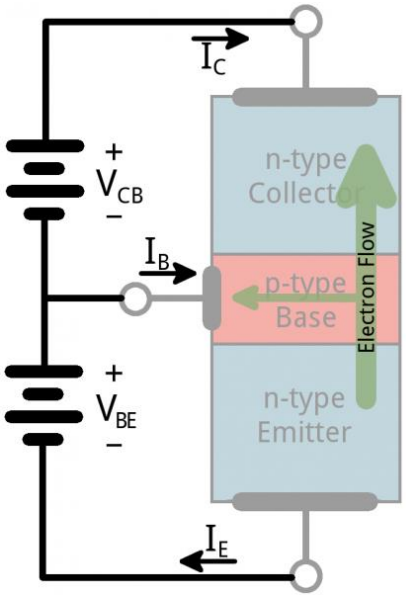


Figure-8

3.4 DEFINITION AND CLASSIFICATION OF INTEGRATED CIRCUITS

The structures known as CMOS and TTL belong to the circuit group called integrated circuits. For this reason, let me first explain the concept of an integrated circuit. In this way, we can make the definitions of CMOS and TTL easier and clearer to understand.

Integrated circuit: Circuits formed by bringing together a large number of resistors, diodes and transistors in order to perform a specific function are called integrated circuits. Integrated circuits can be classified by their level of integration and by the type of transistor used in their structure. The integrated-circuit family to be chosen in a logic application is determined according to the characteristics of the circuit. Today, except for very special circuits, integrated circuits of the TTL and CMOS families are generally used for circuit implementation.

3.5 By Level of Integration:

SSI (Small-Scale Integration): integrated circuits containing fewer than 12 transistors
Medium Scale Integration (MSI): integrated circuits containing between 12 and 99 transistors (e.g. flip-flops, counters)

Large Scale Integration (LSI): integrated circuits containing between 100 and 9,999 transistors (e.g. memory elements such as EPROM, ROM)

VLSI (Very Large Scale Integration): integrated circuits containing between 10,000 and 99,999 transistors (e.g. simple 8-bit microprocessors)

ULSI (Ultra Large Scale Integration): integrated circuits containing 100,000 or more transistors (e.g. advanced integrated circuits)

3.6 COMPARISON OF CMOS AND TTL INTEGRATED CIRCUITS

3.7 TTL (Transistor-Transistor Logic) Integrated Circuits:

Bipolar transistors are used in their structure. Their supply voltage is 5 V. Compared with CMOS integrated circuits, their power losses are very high.

Standard TTL (74XXX family): the oldest; slow, with very high power losses

Low Power TTL (74LXXX family): lower power losses

Schottky TTL (74SXXX family): fast, but with high power losses

Low Power Schottky TTL (74LSXXX family): fast and with low power losses

Advanced LS TTL (74ALSXXX): very good speed-to-power-loss ratio

FAST TTL (74FXXX): the best TTL IC in terms of speed and power losses

3.8 CMOS (Complementary Metal-Oxide Silicon) Integrated Circuits:

Their structure consists of FET-type transistors. Their supply voltage can be between 3 V and 15 V. They have much lower power losses than TTL.

40XX family: the oldest CMOS; slow, with very low power loss.

74HCTXXX family: TTL-compatible (5 V supply voltage); it combines the advantages of CMOS (very low power losses) with the advantages of TTL (very fast). The most popular IC.

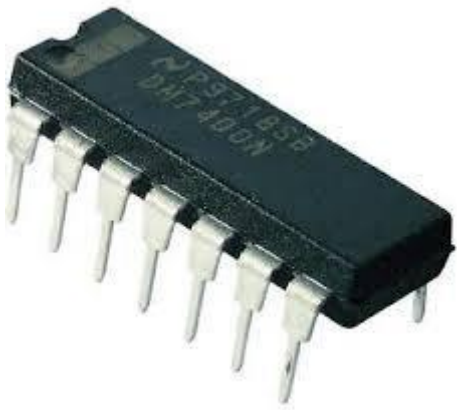


Figure-9

3.9 LOGIC GATES

Logic gates are electrical devices based on the principle of electrical switching. They are generally produced in integrated-circuit form, and basic electronic components such as transistors, diodes and resistors are used in their structure. Logic gates are built from three main logic elements: the AND, OR and NOT gates.

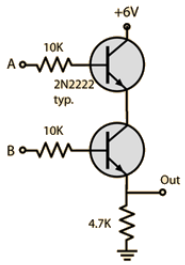


Figure-10

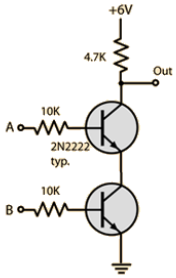


Figure-11

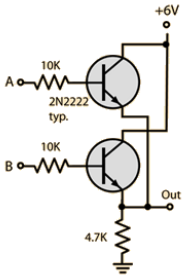


Figure-12

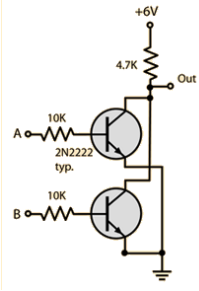


Figure-13

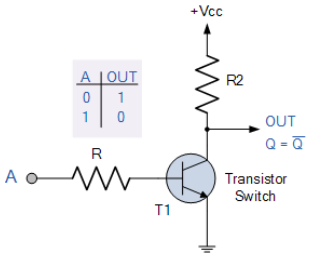


Figure-14

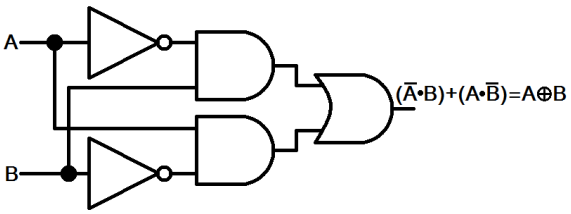


Figure-15

Logic Gates

Name	NOT	AND	NAND	OR	NOR	XOR	XNOR
Alg. Expr.	$\overline{A}$	AB	$\overline{AB}$	$A+B$	$\overline{A+B}$	$A \oplus B$	$\overline{A \oplus B}$
Symbol							
Truth Table	A X 0 1 1 0	B A X 0 0 0 0 1 0 1 0 0 1 1 1	B A X 0 0 1 0 1 1 1 0 1 1 1 0	B A X 0 0 0 0 1 1 1 0 1 1 1 1	B A X 0 0 1 0 1 0 1 0 0 1 1 0	B A X 0 0 0 0 1 1 1 0 1 1 1 0	B A X 0 0 1 0 1 0 1 0 0 1 1 1

Figure-16

3.10 DELAY AND LOGIC LEVELS IN INTEGRATED CIRCUITS

Transition time: when the logic level changes in an integrated circuit, the signal arriving at the output undergoes a certain delay; we call the time taken while rising from 0 to 1 the time Tr, and the time for the transition from 1 to 0 the time Tf.

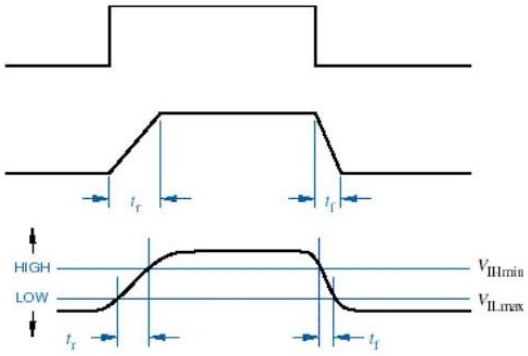


Figure-17

There are two states in logic gates, and these are the logic 0 and logic 1 values. The logic value 0 corresponds to values close to +0 volts, while the logic value 1 corresponds to +5 volts; the table below shows the voltage values of the logic levels.

Device Type	Logic 0	Logic 1
TTL	0 - 0.8v	2.0 - 5v (V_{CC})
CMOS	0 - 1.5v	3.0 - 18v (V_{DD})

Table-1

3.11 BOOLEAN ALGEBRA

Algebra, like other mathematical systems, is based on the principle of deduction. It can be defined by a set of elements, a set of operations and a certain number of unproven postulates. In 1854, George Boole set out to treat logic in a systematic way, and as a result he developed a system known today as Boolean algebra. In 1938, C. E. Shannon, by introducing a two-valued type of Boolean algebra called switching algebra, showed that the properties of bistable electrical switching circuits could be represented with this algebra. In this section, we will learn and apply the postulates formulated by Huntington in 1904 for the formal definition of Boolean algebra.

Boolean algebra is an algebraic structure defined on a set B with the binary operations + and *, provided that the Huntington postulates we will now list are satisfied.

Closure with respect to the + operation; the identity element with respect to + is defined as 0: x+0=x

Closure with respect to the * operation; the identity element with respect to * is defined as 1: x*1=x

Commutative property with respect to +: x+y=y+x

Commutative property with respect to *: x*y=y*x

* is distributive over +: x*(y+z)=(x*y)+(x*z)

+ is distributive over *: x+(y*z)=(x+y)*(x+z)

For every element x in B, the complement of x, denoted x', is also an element of B, so that (a) x+x'=1 and (b) x*x'=0.

There exist at least two elements x, y in B such that x is not equal to y.

3.12 KARNAUGH MAP

It is a map used to simplify the logical expressions given in Boolean algebra. Accordingly, it can also be used to reduce the number of elements in a logic circuit. For example, we can reduce the number of “and” and “or” gates used in a circuit with 3 inputs (3 different binary input values, each of which can be 0 or 1).

In this method, all the alternatives that the input values can take are shown on a table, and a 1 is written for those alternatives for which an output is desired. Then the written 1s are simplified by examining the adjacencies between them.

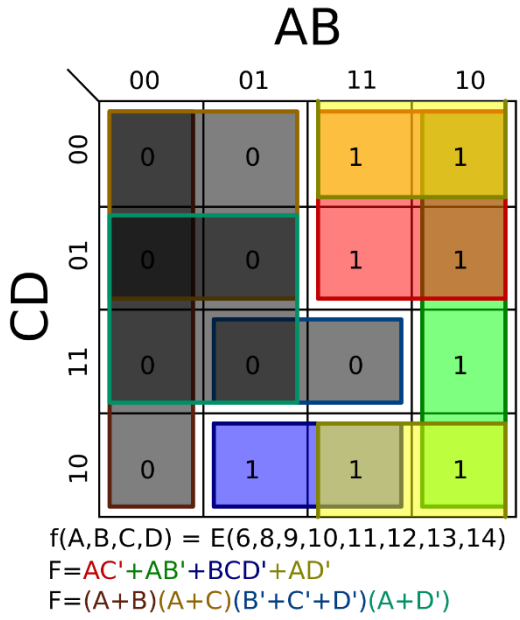


Figure-18

3.13 HALF ADDER (LOGIC CIRCUITS)

It is the circuit that adds two input values given in base two (binary). Accordingly, the following table is obtained for the inputs A and B. (The + sign in the table below should not be confused with the logical OR between propositions; the + sign denotes addition.)

A	B	E	T
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

In the table given above, the sum of the numbers A and B is shown in the E and T values. For example, for the values A=1 and B=1 the sum is 2 (1+1=2); the equivalent of this value in base two is 10. The reason there are 2 outputs is that the summed numbers cannot be expressed with a single bit. When designing the circuit that produces the sum values given above, the E and T bits must be considered separately. If Karnaugh maps are used in the design of the circuit performing this addition, the following maps can be drawn for the T and E bits:

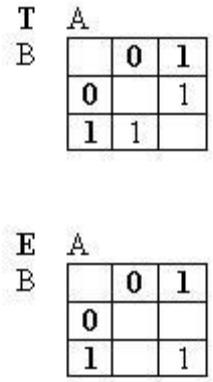


Figure-19

In the figure given above, the output values of the T and E bits are marked on the Karnaugh map. Since there are no adjacent 1 values, both bit values must be written, without simplification, as the following expressions:

T=AB'+A'B E=AB

3.14 FULL ADDER (LOGIC CIRCUITS)

It is the circuit that gives the sum of 3-bit input values (inputs in base two). Accordingly, the following table is obtained for the inputs A, B and C. (The + sign in the table below should not be confused with the logical OR between propositions; the + sign denotes addition.)

A	B	C	E	T
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

In the table given above, the sum of the numbers A, B and C is shown in the E and T values. For example, for the values A=1, B=1 and C=1 (the last row of the table) the sum is 3 (1+1+1=3); the equivalent of this value in base two is 11. The reason there are 2 outputs is that the summed numbers cannot be expressed with a single bit. When designing the circuit that produces the sum values given above, the E and T bits must be considered separately. If Karnaugh maps are used in the design of the circuit performing this addition, the following maps can be drawn for the T and E bits:

T

A \ BC	00	01	11	10
0		1		1
1	1		1	

E

A \ BC	00	01	11	10
0			1	
1		1	1	1

Figure-20

In the map above, the values of the two different output bits (T = sum and E = carry) that give the sum values for three inputs are shown. The adjacencies between these values are enclosed in red circles. When these islands made up of 1s are simplified:

T=A’B’C+A’BC’+ABC+AB’C’ E=AB+AC+BC

3.15 REGISTERS

A general sequential circuit: Registers — they are a good example for sequential circuit analysis and synthesis. They can also be used in the design of larger-scale sequential circuits.

While flip-flops hold a single bit, registers can hold larger amounts of data. Registers are at the heart of modern processor design. — There is more than one kind of register.

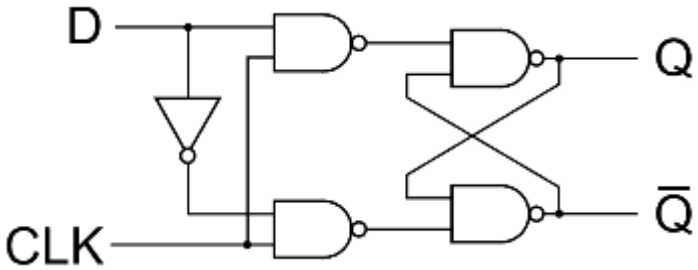


Figure-21

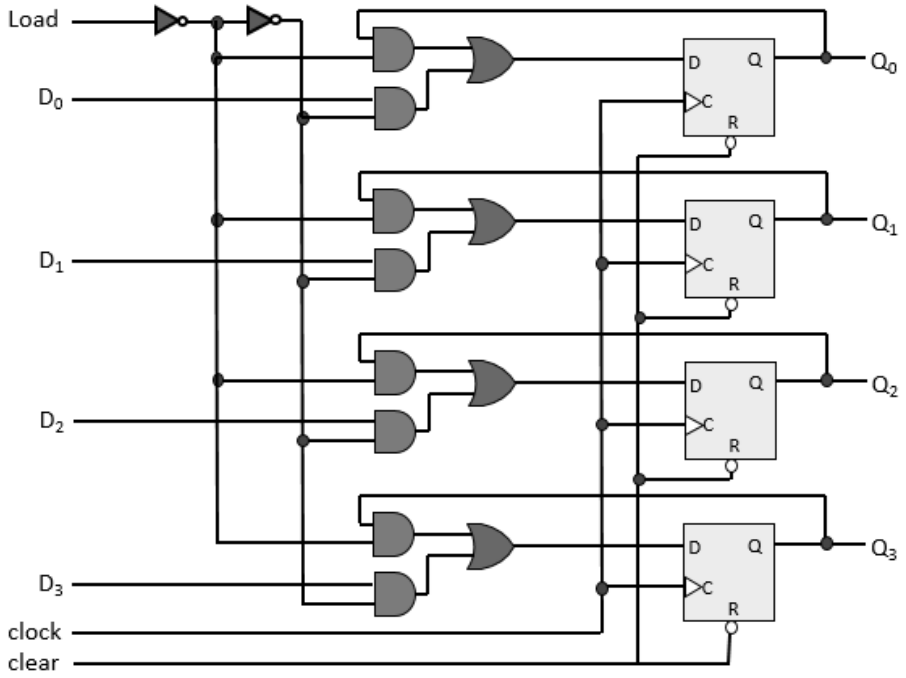


Figure-22

4.1 CPU HARDWARE

It is the section that carries out instruction execution. It is also known as the execution unit. This unit executes instructions; it is fed through paths called pipelines and, using integers, performs read, modify and instruction-execution operations. Inside the core there are the ALU, general-purpose registers, the Status Register (SR) and the Program Counter (PC).

ALU (Arithmetic Logic Unit): It is the section where the mathematical and logical operations to be performed by the processor are carried out. It constitutes the most important part of the processor. Advanced processors include an FPU (Floating Point Unit), which performs mathematical operations on the digits after the point. This unit is responsible for non-integer, floating-point calculations.

Control Unit: It ensures that the instructions sent to the processor are decoded (i.e. what the instruction means is identified) and executed. The control signals that make the units inside and outside the processor work in synchronization are generated by this unit. The Control Unit contains the Instruction Register (IR) and the Instruction Decoder (ID).

4.2 ALU (ARITHMETIC LOGIC UNIT)

The most important part of the microprocessor is the arithmetic and logic unit (ALU) (Figure 2.5). This unit performs addition, subtraction, comparison, shift and rotate operations on the registers. The result of the operation performed is stored in the registers. Sometimes it only affects the condition code register. As a result of an operation in the ALU, several of the flags in the condition code register may be affected, or none at all. For the programmer, the state of the flags affected as a result of the operation performed in the ALU is often more important. The ALUs inside advanced microprocessors are able to perform multiplication and division. The largest data the ALU can operate on is limited by the data size of the registers in the microprocessor. The ALU in a microprocessor with an 8-bit architecture operates on numbers of at most 8 bits.

The arithmetic operations performed in the ALU may vary according to the structure of the microprocessor. In a microprocessor with an 8-bit architecture, addition, subtraction, multiplication and division, as well as mathematical operations with decimal numbers, can be performed. Advanced processors also have a separate math processor for operations with large decimal numbers.

- Logical operations
- Logical multiplication: the AND operation
- Logical addition: the OR operation
- Exclusive OR: the XOR operation
- Negation: the NOT operation
- Operations such as comparison (=, =<, =>, <>, etc.) and shifting are performed in this unit.
- Shift right or shift left, and rotate operations
- Content increment or decrement operations

All these operations are carried out by a system made up of gates and flip-flops of various technological structures.

4.3 ALU DESIGN RESULTS

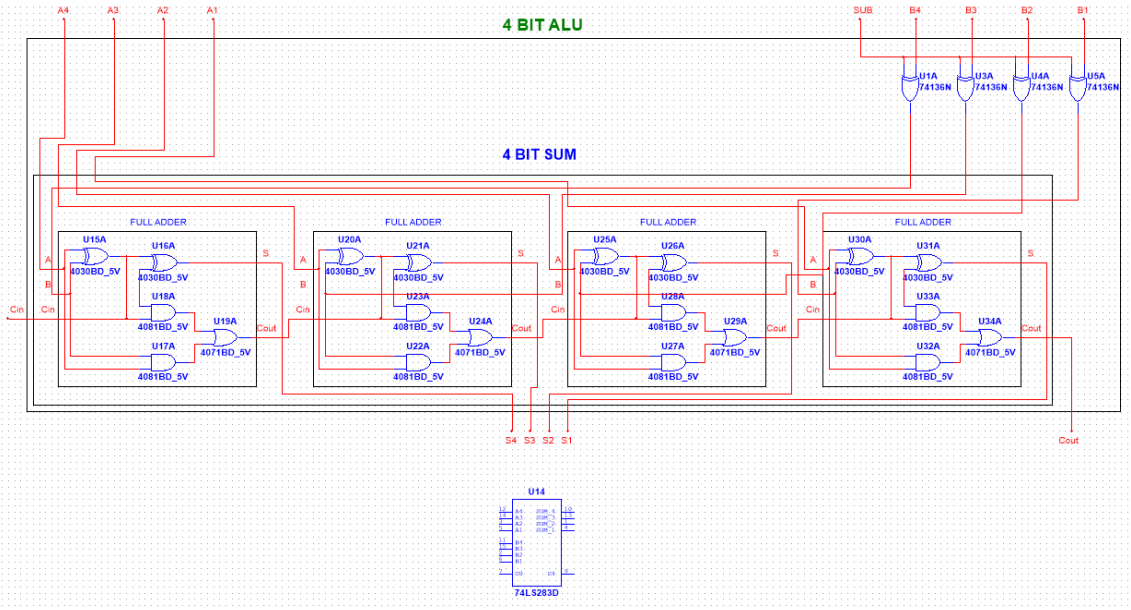


Figure-23

In the 4-bit ALU design seen above, the 4-bit data coming from the A and B terminals are added and sent to the S terminals. This ALU unit performs addition and subtraction, but if we wish, we can add other mathematical or logical operations.

If the SUB control signal is 0, it performs addition. We can see this operation below.

$$S0 = A0 + B0$$

$$S1 = A1 + B1$$

$$S2 = A2 + B2$$

$$S3 = A3 + B3$$

If A3 and B3 have the logic value 1, the Cout value gives an output of logic 1.

4.4 CU (CONTROL UNIT)

The lines used to carry the signals that coordinate the relationship between the units in a microprocessor system are called the control bus. Since the instruction set of each microprocessor and the signals used for specific purposes are different, each microprocessor may have a control bus containing a different number of lines. The signals on the control bus are used to carry out three different tasks:

Signal selection: signals that determine which signals will be used in the system and where those signals will be applied

Direction assignment: signals that determine in which direction the data in the system will travel (read or write)

Timing: signals that determine the order and timing of the operations to be performed

The number of lines in the control bus does not depend on the bit count of the microprocessor. Since the lines that make up the control bus spread through the microprocessor like a network, the term control lines can be used instead of the term control bus. The operation of the units in a microprocessor system is directed by the trigger signals transmitted over the control lines. For example, when data is to be read from a memory, the appropriate R/W signal required for the read operation must be applied together with the enable (CS – Chip Select) signal that will activate the relevant memory IC.

4.5 CU DESIGN RESULTS

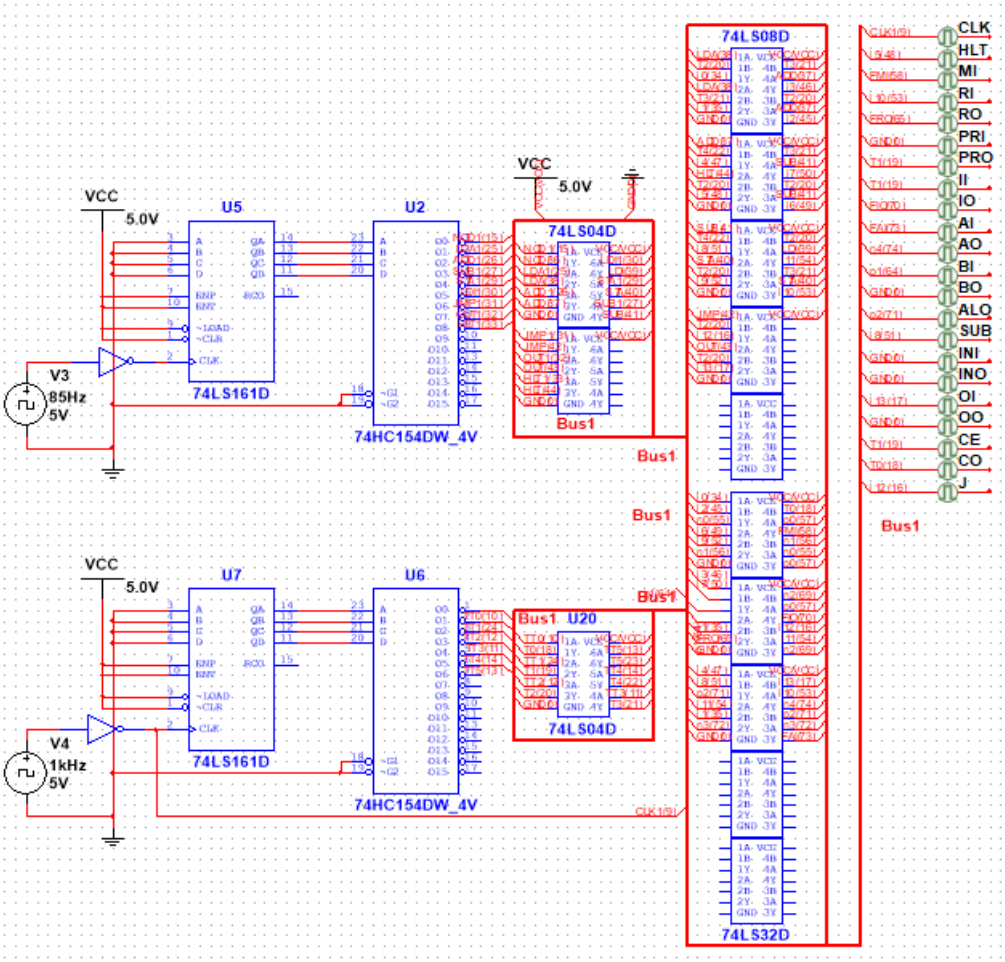


Figure-24

In this design of ours, about 9 OPCODEs have been written, and the circuit that will execute these codes has been designed. As a result of this design, at the output we see the control signals that we will use in our processor.

4.6 CU TIMING ERROR RESULTS

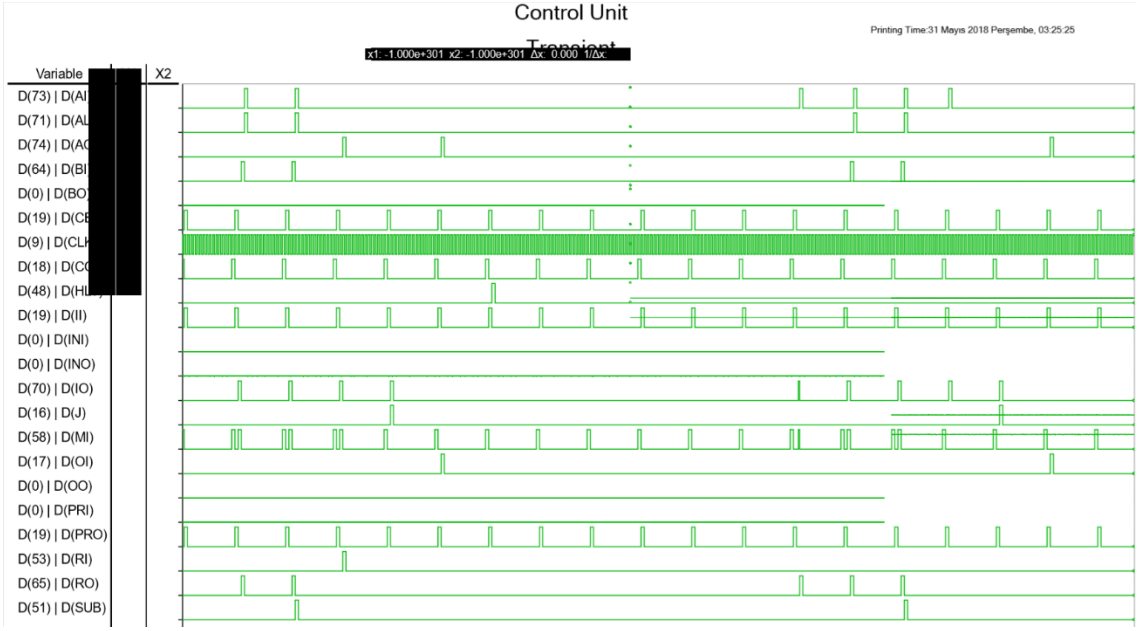


Figure-25

Here we see the signaling of the control signals at the output of the circuit we designed in the control unit; these signals may experience delays at the output, and these delays are related to the structure of the ICs.

And these delays must be equal; the necessary design will be made for this.

4.7 CREATING AN OPCODE LIBRARY

LDA (0000)

1. CO MI
2. PRO II
3. CE

FETCH

4. IO MI
5. RO AI

In this flow we see the execution of the LDA instruction; the first 3 steps are the fetch operation, while the remaining steps make up the LDA instruction.

We can write the other instructions in the same way.

CO: Puts the counter onto the Bus

MI: The value on the Bus enters the MAR

PRO: Puts the value in the Program onto the Bus

IO: Puts the value in the Instruction register onto the Bus

II: The value on the Bus goes into the Instruction register

CE: Increments the counter by one

RO: Puts the value in the RAM onto the Bus

AI: The value on the Bus goes into the A register

5.1 OVERVIEW OF THE QUARTUS PROGRAM

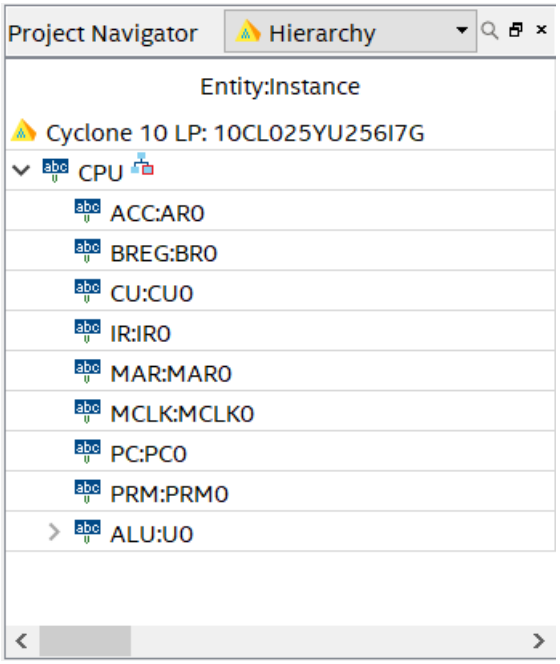


Figure-26

Intel Quartus Prime is programmable logic device design software produced by Intel; before Intel's acquisition of Altera, this tool was called Altera Quartus II.

Quartus Prime provides analysis and synthesis of HDL designs, enabling the developer to compile their designs, perform timing analysis, examine RTL diagrams, simulate a design's response to different stimuli, and configure the target device with the programmer. Quartus Prime includes an implementation of VHDL and Verilog for hardware description, visual editing of logic circuits, and vector waveform simulation.

5.2 CPU VERILOG DESIGN

Verilog is a hardware description language used to model electronic systems. Verilog (sometimes also called “Verilog HDL”) supports the design, verification and implementation of analog, digital and mixed-signal circuits at various levels. The designers of the Verilog language wanted it to have a syntax close to the C programming language, so that engineers familiar with that language could use it easily. The language is case-sensitive, and the keywords of its basic control flow, such as “if” and “while”, are similar to C. Verilog differs from C in a few fundamental ways: Verilog uses Begin/End instead of curly braces to define a block of code.

```
2  //                                                                          //
3  //                  +-----------------------------------+                   //
4  //                  |     Copyright 2018 HSYNKY :D      |                   //
5  //                  |Project: 4 Bit CPU Design (Inside)|                   //
6  //                  |Part: Inside Of CPU                |                   //
7  //                  |      Designer : Huseyin Kaya      |                   //
8  //                  +-----------------------------------+                   //
9  //                                                                          //
10 //////////////////////////////////////////////////////////////////////////////
11
12 module CPU (data);
13 //inout [3:0] databus;
14 //inout [3:0] adressbus,controlbus;
15 output [3:0] data;
16 wire clk,ce,co,mi,mo,ii,io,cout,carry,pcr,ai,bi,alue,pre,prw,hlt;
17 wire [3:0] databus,adressbus,a,b;
18 wire [2:0] alu;
19 wire [2:0] op;
20 reg r,hld;
21
22 assign data = databus;
23
24 initial begin
25 r=0; hld=0;
26 end
27
28 MAR MAR0 (clk,mi,r,databus,adressbus);
29 IR IR0 (clk,ii,io,r,databus,op,databus);
30 ACC AR0 (clk,ai,r,databus,a);
31 BREG BR0 (clk,bi,r,databus,b);
32 ALU U0 (a,b,databus,alu,cout,alue);
33 MCLK MCLK0 (clk,hlt || hld);
34 PRM PRM0 (databus,adressbus,pre,prw);
35 PC PC0 (databus,ce,co,pcr);
36 CU CU0 (clk,r,op,ce,co,mi,mo,ii,io,pcr,ai,bi,alue,pre,prw,hlt,alu);
37 //FLAG F0 (clk,cout,carry,cr);
38
39 endmodule
40
```

Figure-27

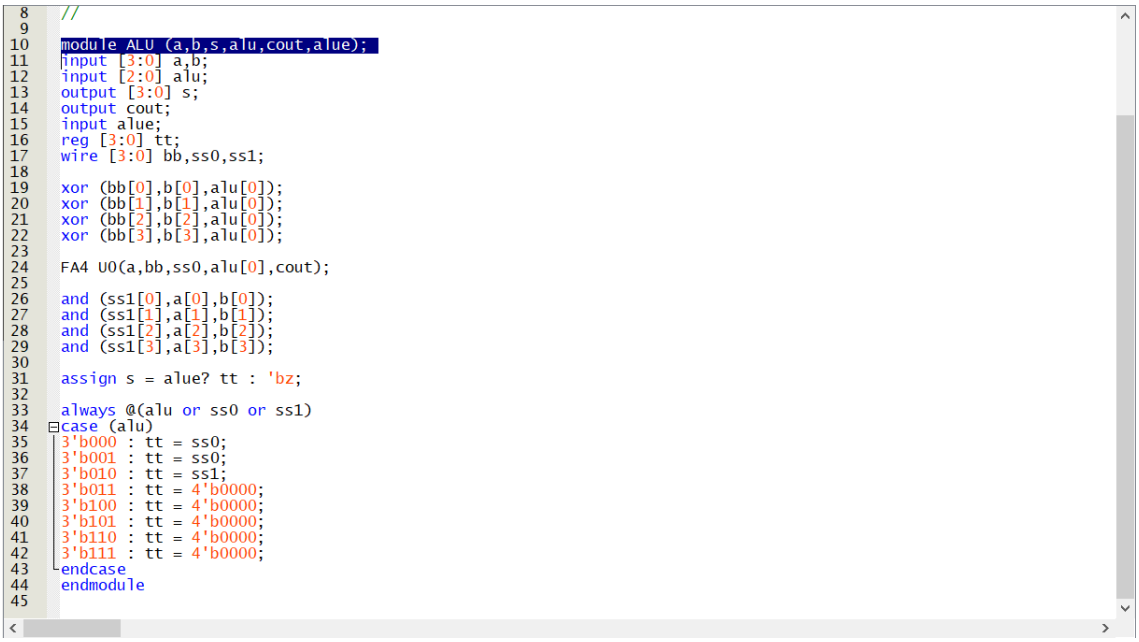
In this section we created the main CPU module together with its submodules; in the code used here, output means an output pin, wire an interconnection, and reg a register definition.

Assign – used to be able to join two interconnections.

Initial – lets us set initial values.

Endmodule – ends the module.

5.3 ALU VERILOG DESIGN



```
8   //
9
10  module ALU (a,b,s,alu,cout,alue);
11  input [3:0] a,b;
12  input [2:0] alu;
13  output [3:0] s;
14  output cout;
15  input alue;
16  reg [3:0] tt;
17  wire [3:0] bb,ss0,ss1;
18
19  xor (bb[0],b[0],alu[0]);
20  xor (bb[1],b[1],alu[0]);
21  xor (bb[2],b[2],alu[0]);
22  xor (bb[3],b[3],alu[0]);
23
24  FA4 U0(a,bb,ss0,alu[0],cout);
25
26  and (ss1[0],a[0],b[0]);
27  and (ss1[1],a[1],b[1]);
28  and (ss1[2],a[2],b[2]);
29  and (ss1[3],a[3],b[3]);
30
31  assign s = alue? tt : 'bz;
32
33  always @(alu or ss0 or ss1)
34  case (alu)
35  3'b000 : tt = ss0;
36  3'b001 : tt = ss0;
37  3'b010 : tt = ss1;
38  3'b011 : tt = 4'b0000;
39  3'b100 : tt = 4'b0000;
40  3'b101 : tt = 4'b0000;
41  3'b110 : tt = 4'b0000;
42  3'b111 : tt = 4'b0000;
43  endcase
44  endmodule
45
```

Figure-28

We see the design, in the Verilog language, of the ALU submodule from the CPU Verilog design we saw in the previous section.

In the ALU design, using full adders as in the FA4 module, we will select which operation the 4-bit full adder will perform via the alu code; as a result, we have the a and b inputs processed with the help of the ALU control signals and, through the alue input pin, send the result data to the s output.

And finally, if there is a cout carry bit, it is recorded into the flag registers.

5.4 CPU RTL VIEWER

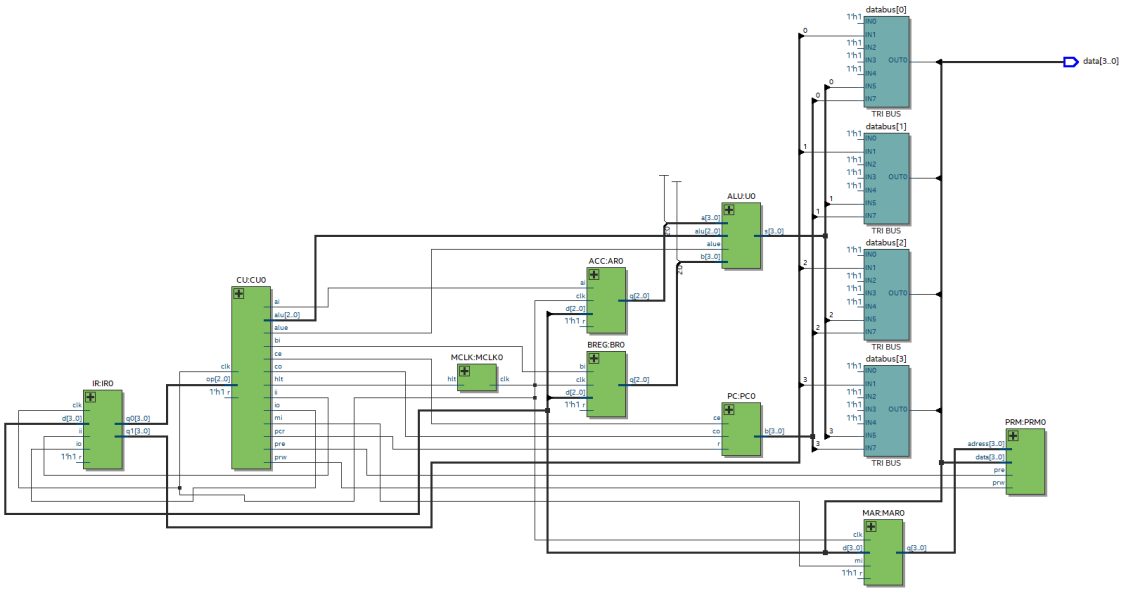


Figure-29

It helps us to see and analyze better, in real time and by visualizing them, the modules of the program code we wrote for the CPU in Verilog, and here we can see the data paths better.

And at the side we see the visualized interior of the ALU submodule.

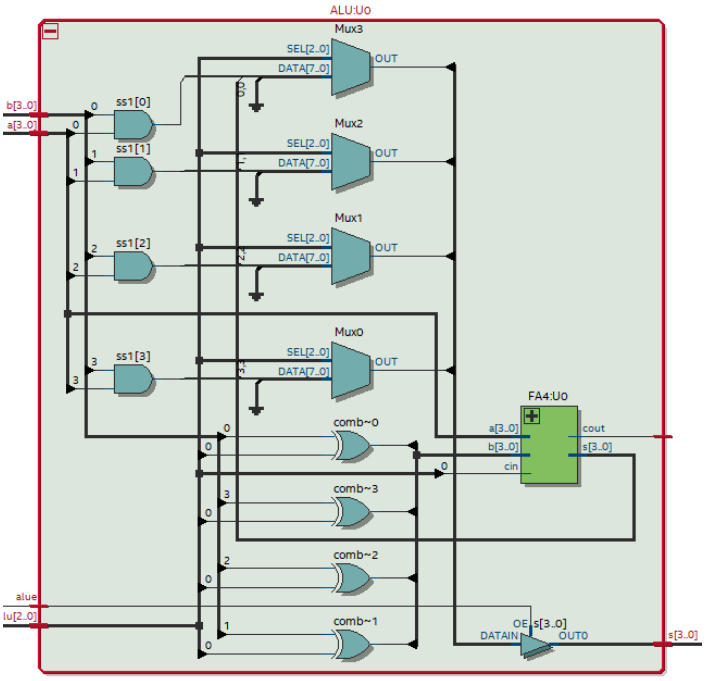


Figure-30

5.5 ALTERA SIM SIMULATION RESULTS

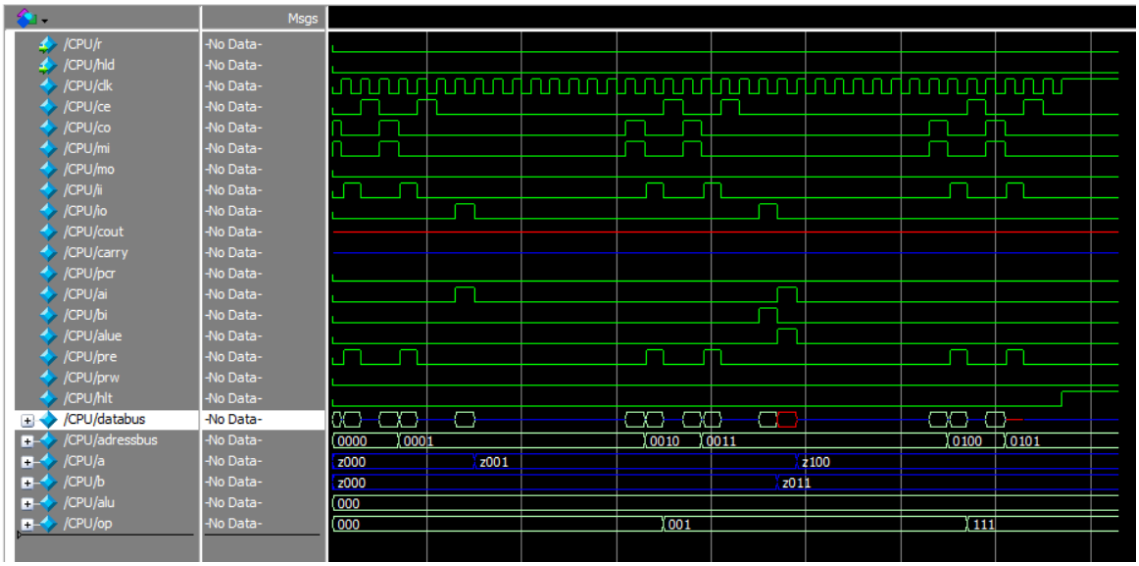


Figure-31

In this image, we simulate the module we created, and by evaluating the incoming results, changes and adjustments are made to the design if necessary.

For example, the red signals seen in the simulation mean an unknown value, which is not a situation we really want; for this, we must make changes to the design and design a more stably working system.

6. REFERENCES

http://diyot.net/transistor/

http://www.elektrikport.com/universite/transistorler-nasil-calisir-elektrikport-akademi/8946#ad-image-0

http://www.odevsel.com/egitim/2583/cmos-nedir-ttl-nedir-cmos-ile-ttl-arasindaki-farklar-nelerdir.html

https://www.muhendisbeyinler.net/lojik-kapi-devreleri/

Digital Computer Electronics (Albert Paul Malvino)

http://www.elektrikport.com/teknik-kutuphane/boole-cebri-sayisal-devreler/15399#ad-image-0

http://bilgisayarkavramlari.sadievrenseker.com/2007/12/07/karnaugh-haritasi-karnaugh-map/

http://bilgisayarkavramlari.sadievrenseker.com/2007/12/08/tam-toplayici-full-adder/

http://www.yildiz.edu.tr/~bataslar/Saklayici.pdf

http://www.selcuk.edu.tr/dosyalar/files/054003/%C4%B0%C5%9Flemciler%2

7. ACKNOWLEDGEMENTS

We offer our thanks and gratitude to our esteemed advisor Dr. Gökhan Yenikaya, who contributed to the preparation of this project, who did not withhold his help during the hardware and software stages, and who guided us during the construction stage.

8. CURRICULUM VITAE

HÜSEYİN KAYA

He was born in Manisa in 1996. He graduated from Turgutlu Anatolian High School in 2014. In 2014 he was admitted to the Department of Electrical and Electronics Engineering at Uludağ University. He is still continuing his education.