# [ארכיטקטורת יחידת עיבוד מרכזית](http://hl2.bgu.ac.il/PortalCategories.asp)

## מגישים: אייר גלעד 300309937

## דודו אבו 038109484

# תוכן עניינים

[1. מטרת המעבדה 2](#_Toc245477475)

[2.רכיב ה ALU 2](#_Toc245477476)

[3. Arithmetic Unit 4](#_Toc245477477)

[4. Decoder Unit 5](#_Toc245477475)

[5. Add Sub Unit 7](#_Toc245477476)

[6. N bit adder 8](#_Toc245477477)

[7.1 bit adder 9](#_Toc245477475)

[8.Mul Unit 01](#_Toc245477476)

[9. Abs Unit 1](#_Toc245477477)1

[10. Mac Unit](#_Toc245477475) 12

[11. Mac clock Unit](#_Toc245477476) 13

[12. Max Min Unit](#_Toc245477477) 14

[13. Shift Unit](#_Toc245477477) 15

[14. Output selector Unit](#_Toc245477475) 16

[15.Testing of the ALU](#_Toc245477476) 17

[16. Conclusions 2](#_Toc245477477)2

מטרת המעבדה

* רכישת ידע בסיסי בשפת VHDL ו – ModelSim .
* צבירת ידע בסיסי במערכות דיגיטליות .
* רכישת ידע והבנה בסיסית לניתוח תכנון ארכיטקטורה.

1. **ALU**

במעבדה זו נדרשנו לתכנן רכיב ALU- Arithmetic Logic Unitהמבצעת פעולות מסוימות (ראה טבלה 1.2) על הקלט אותה היא מקבלת, ובהתאם לקוד פעולה אותה היא מקבלת גם כן. בסיום ביצוע הפעולה היא מוציא את תוצאת הפעולה כפלט, וסטטוס במקרה מסוים(יפורט בהמשך).

הערה:

לאורך כל המסמך נסמן ב- N את אורכו של באס הכניסה (אורכם של וקטורי הקלט הלוגיים) ליחידת ה- ALU , N הנו generic ומשמש כמשתנה עבור שאר היחידות בתכנון. גודלו של N נע בין 8-32.

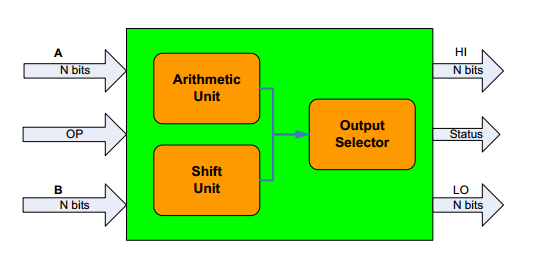
 1.1 ALU schematic design

Figure 1. Alu schematic design

* 1. ALU port table

|  |  |  |  |
| --- | --- | --- | --- |
| Port description | Port length | Port direction  - in/out | Port name |
| Input logical vector | N | in | A |
| Input logical vector | N | in | B |
| Code for operation to be executed(see table 1) | 4 | in | Op Code |
| Present the N msb bits of the result of the operation executed on A and B. | N | out | Hi |
| Present the N Lsb bits of the result of the operation executed on A and B.  \* Some operation use **only** LO for output in this case HI is irrelevant | N | out | LI |
| register presents the status for the sub operation(#1) only, (see table 4.1)  \*on other operation status output is irrelevant | 6 | out | status |

**Table 1.1: ALU port map**

1.3 ALU available operations and codes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Op code | operation | RTN | Description |
| 0 | 0000 | ADD | LO = A + B | Arithmetic Add |
| 1 | 0001 | SUB | LO = A - B | Arithmetic Sub |
| 2 | 0010 | MIN | LO=Min(A,B) | Return minimum between A and B |
| 3 | 0011 | MAX | LO=Max(A,B) | Return maximum between A and B |
| 4 | 0100 | ABS | LO=Abs(B) | Return absolute value of B |
| 5 | 0101 | RST | MAC=0 | Reset MAC |
| 6 | 0110 | MAC | MAC= MAC+A\*B  HI,LO)=MAC) | Multiply Accumulate (MAC is internal N\*2 bits  register) signed numbers |
| 7 | 0111 | MUL | (HI,LO) = (A\*B) | Multiply two signed numbers (Result is N\*2 bits register) |
| 8 | 1000 | SHL | LO = A << B | Shift left register A, B times |
| 9 | 1001 | SHR | LO = A>>B | Shift right register A, B times |

**Table 1.2: ALU Op Codes and operation**

Note: Other op codes available for input (4 bits) will consider as nop- no operation.

The ALU unit contains 3 sub units- Arithmetic unit (1), shift unit (8), output selector (9) as detailed below

**2. Arithmetic Unit**

The arithmetic unit performs all the arithmetic operation available in the ALU, as see in table (1.2) the operation codes are numbers # (0-7).

2.1 arithmetic unit schematic design

(N..1)

Output

(2N..1)

(N..1)

Input 1

Arithmetic Unit

Input 2

Status

(6..1)

(3..1)

OP code

Figure 2. Arithmetic unit

2.2 Arithmetic unit port table

|  |  |  |  |
| --- | --- | --- | --- |
| Port description | Port length | Port direction  - in/out | Port name |
| Input logical vector | N | in | Input1 |
| Input logical vector | N | in | Input2 |
| Code for operation to be executed | 3 | in | Op Code |
| Present the result of the operation executed on A and B. | N2 | out | output |
| register presents the status for the sub operation(#1) only, (see table 4.1) \*on other operation status is irrelevant | 6 | out | status |

**Table 2.1: Arithmetic unit port map**

The arithmetic unit contains 5 sub-units: decoder (3) add/sub (4), mul (5), abs (6), mac (7) max/min (8) as detailed below.

***3. Decoder unit***

The decoder selects in respecting to 3 digits (a b c), actually part of original op code to the ALU, which unit inside the arithmetic unit is relevant to requested operation and gives it, the enable signal.

By this logic:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a | b | c | D0 | D1 | D2 | D3 | D4 | Selected unit |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | Add/sub unit |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | Add/sub unit |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | Max/Min unit |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | Max/Min unit |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | Absolute unit |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | Mac unit |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | Mac unit |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | Mul unit |

**Table 3.1: decoder truth table**

3.1 decoder schematic design

(1) 

(1) 

D0

(3..1)

(1) 

D1

Dec

Decoder

(1) 

D2

(1) 

D4

D3

Figure 3. Decoder

3.2 decoder port table

|  |  |  |  |
| --- | --- | --- | --- |
| Port description | Port length | Port direction  - in/out | Port name |
| Input for selecting relevant line (d0, d1, d2, d3) to put logical “1” init. see table(4.1) | 3 | in | Dec |
| Output signal takes “1” when selected “0” otherwise | 1 | out | D0 |
| Output signal takes “1” when selected “0” otherwise | 1 | out | D1 |
| Output signal takes “1” when selected “0” otherwise | 1 | out | D2 |
| Output signal takes “1” when selected “0” otherwise | 1 | out | D3 |
| Output signal takes “1” when selected “0” otherwise | 1 | out | D4 |

**Table 3.2 decoder port map**

***4. add/sub*** ***unit***

The add/sub makes adding( a+b ) or subtraction(a-b) between to input logical vector its gets (a, b). The selecting between the operations needed is by input signal called sel.

The function of adding and subbing was implemented both behaviorally and structurally, we chose the implementation for the unit to be connected to complete design will be the behaviorally one.

The structured implementation of the adding/subbing function is by connecting N, one bit adders to get an N bit full adder. In section called structural implementation we will describe both the one bit adder, and the N bit adder

This unit also responsible for the status output for the all system which is relevant only on subtraction operation(#1).

The status output contains 6 bits(0-5), when ‘0’ is the lsb, and ‘5’ is the msb. Each bits has it significance, when a specific bit has value of”1” that means the condition fits to this bit is true. The conditions and the output bits relation showed is table 4.1

|  |  |  |
| --- | --- | --- |
| Bit number | Condition name | description |
| 0 | Eq | A=B |
| 1 | Ne | A!=B |
| 2 | Ge | A>=B |
| 3 | Gt | A>B |
| 4 | Le | A<=B |
| 5 | Lt | A<B |

**Table 4.1 status output bits relative to conditions**

A. behavioral implementation

4.1 add/sub schematic design

(N..1)

(N..1)

A

Output

(N..1)

B

Add Sub Unit

(6..1)

(1)

En

Status

(1)

Sel

Figure 4. Add Sub unit

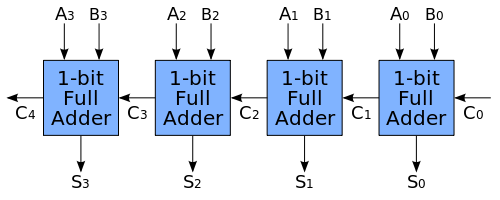
4.2 add/sub port table.

|  |  |  |  |
| --- | --- | --- | --- |
| Port description | Port length | Port direction  - in/out | Port name |
| Input logical vector | N | in | A |
| Input logical vector | N | in | B |
| Enable signal for the unit | 1 | in | En |
| Selects between add or sub operation | 1 | in | Sel |
| The result of the executed operation | N | out | Output |
| register presents the status for the sub operation(#1) only, (see table 4.1) \*on other operation status is irrelevant | 6 | out | Status |

**Table 4.2 add/sub unit port map**

B. structural implementation

4.4 add/sub schematic design-N bit adder



Sum3

Sum2

Sum1

Sum0

Figure 5. N bit Adder unit ( in this example N=4)

4.5 N bit adder port table.

|  |  |  |  |
| --- | --- | --- | --- |
| Port description | Port length | Port direction  - in/out | Port name |
| Input logical vector | N | in | A |
| Input logical vector | N | in | B |
| The result of the executed operation | N | out | sum |
| Output carry | 1 | out | cout |

**Table 4.3 N bits adder port map**

4.6 add/sub schematic design-One bit adder

(N..1)

(1)

A

Sum

(N..1)

B

1 bit Adder

(1)

Cin

(1)

Cout

Figure 6. 1 bit Adder unit

4.7 one bit adder port map

|  |  |  |  |
| --- | --- | --- | --- |
| Port description | Port length | Port direction  - in/out | Port name |
| Input bit | 1 | In | A |
| Input bit | 1 | In | B |
| Input carry | 1 | In | cin |
| The result | 1 | Out | sum |
| output carry | 1 | Out | cout |

**Table 4.4 one bits adder port map**

***5. Mul unit***

The mul makes multiply between to input logical vector its gets (a, b). The multiply operation is signed.

Overflow would be ignored in signed operation.

5.1 mul schematic design

(N..1)

(2N..1)

A

Output

(N..1)

B

Mul Unit

(1)

En

Figure 7. Mul/Umul unit

5.2 mul port table

|  |  |  |  |
| --- | --- | --- | --- |
| Port description | Port length | Port direction  - in/out | Port name |
| Input logical vector | N | in | A |
| Input logical vector | N | in | B |
| Enable signal for the unit | 1 | in | en |
| The result of the executed operation | 2N | out | output |

**Table 5.1: mul/umul port map**

***6.*** ***Absolute*** ***unit***

Gives the Absolute value of the signed input vector(N) . The output is 1 output vector.

6.1 Absolute schematic design

(N..1)

B

Absolute Unit

D\_Lo

(N..1)

(1)

En

Figure 8. Absolute Unit

6.2 Absolute port table

|  |  |  |  |
| --- | --- | --- | --- |
| Port description | Port length | Port direction  - in/out | Port name |
| Input logical vector | N | in | B |
| Enable signal for unit | 1 | in | en |
| Register represents the reminder of the division Integer part of the division | N | out | D lo |

**Table 6.1: Absolute** **port map**

***7. mac unit***

This unit has its internal register called mac(length N), doing the operation between the inputs (a,b) which is mac+a\*b ( signed multiplication). Mac is initialized as logical vector equal to “0”.

This unit is synchronic by 1/10 ns clock(rising edge). Allowing multiply and accumulation for same input. As different from other opcodes the mac/umac opcode needs the possibility to sum the same inputs with mac register,(loops for example), and that’s why a clock is needed for this unit design.

This unit contains 3 other sub-units (add/sub, mul which detailed above),and **mac clock unit** especially for this unit only(8). The unit also knows to reset the mac to “0” no matter which value in contains.

7.1 mac /umac schematic design

Clk

(N..1)

(2N..1)

A

Output

(N..1)

Mac Unit

B

(1)

En

Res

Figure 9. Mac/Umac Unit

7.2 mac port table

|  |  |  |  |
| --- | --- | --- | --- |
| Port description | Port length | Port direction  - in/out | Port name |
| Input logical vector | N | in | A |
| Input logical vector | N | in | B |
| Enable signal for unit | 1 | in | En |
| Reset signal for mac register | 1 | in | Res |
| Clk enters for mac/umac clock unit, the output value is according to current input upon clock rising . | 1 | in | Clk |
| The result of the executed operation | 2N | out | Output |

**Table 7.1: mac/umac port map**

**8.mac clock**

This unit is a 1/10ns clock, especially for the mac unit. This unit has no input, only output of switching state from "1" to "0" every 5ns. This unit enters as clock signal for the mac/umac unit.

8.1 mac/umac clock schematic design

(1)

Clk

Clock Unit

Figure 10. Clock unit

8.2 mac clock port table

|  |  |  |  |
| --- | --- | --- | --- |
| Port description | Port length | Port direction  - in/out | Port name |
| Switching output form "1" to "0" every 5ns | 1 | out | Clk |

**Table 8.1: mac/umac clock port map**

**9.max/min unit**

Similar to the Add/Sub unit the Max/Min unit takes two signed input vectors and gives the minimum or maximum of them according to the Sel input**.**

4.1 max/min schematic design

(N..1)

(N..1)

A

Output

(N..1)

B

Max Min Unit

(1)

En

(1)

Sel

Figure 11. Max Min unit

4.2 add/sub port table.

|  |  |  |  |
| --- | --- | --- | --- |
| Port description | Port length | Port direction  - in/out | Port name |
| Input logical vector | N | in | A |
| Input logical vector | N | in | B |
| Enable signal for the unit | 1 | in | En |
| Selects between add or sub operation | 1 | in | Sel |
| The result of the executed operation | N | out | Output |

**Table 9.1 add/sub unit port map**

**10.shift unit**

This unit makes shifting arithmetic to input A number of times as in input b(the significance of be is an integer with unsigned binary presentation), the shifting can be done on right or left direction according to code signal. This unit has a relevant output only for shift op code on if it gets code numbers #(10-15) it does none.

10.1 shift unit schematic design

(2N..1)

(N..1)

A

(N..1)

Shift Unit

Output

B

Code

(3..1)

Figure 12. Shift Unit

10.2 shift unit port table

|  |  |  |  |
| --- | --- | --- | --- |
| Port description | Port length | Port direction  - in/out | Port name |
| Input logical vector | N | in | A |
| Input logical vector | N | in | B |
| Selects between operation(ssl/ssr/nop) | 3 | in | Code |
| The result of the executed operation | N | out | Output |

**Table 10.1: shift unit port map**

**11. Output selector unit**

This unit selects with unit in the ALU has a relevant output for the entire system. If it’s the arithmetic unit, it takes is status output also, if it’s the shit unit the status output will be “0”.if selection signal is “0” the output will be from the arithmetic unit otherwise it will be from the arithmetic unit. This output will go directly to the ALU output ports. This unit is work similar to multiplexer.

11.1 output selector schematic design

Out\_Hi

(N..1)

(2N..1)

(N..1)

A

(N..1)

Output Selector Unit

Out\_Hi

B

Status Out

(6..1)

Status in

(6..1)

(1)

Sel

Figure 13. Output Selector Unit

11.2 output selector port table

|  |  |  |  |
| --- | --- | --- | --- |
| Port description | Port length | Port direction  - in/out | Port name |
| Input logical vector from arithmetic unit | N2 | In | A |
| Input logical vector | N | In | B |
| Status input dorm arithmetic unit | 6 | In | Status in |
| Selecting between units | 1 | In | Sel |
| Relevant status to be out of entire system | 6 | out | Status out |
| Output for HI register from the entire system | N | out | Out hi |
| Output for LO register from the entire system | N | out | Out lo |

**Table 11.1: output selector port map**

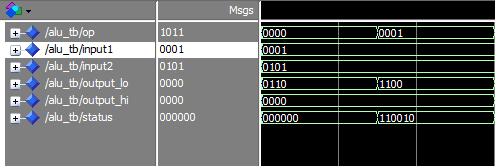
***12. Testing the ALU***

In this section we will presents test for every operation of the unit and check their correctness in respect of the input signals. We will use the ModelSim simulation and show the signals of the input and output of the system.

\* Add and sub unit - testing (through the ALU unit):

Op=”0001” sub : 1-5

Op=”0000” add : 1+5



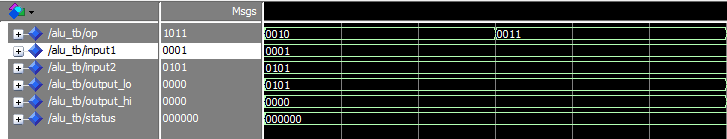
The Status is changed – A<B as expected

The result is -4 in the Low bits as expected

The result is 6 in the Low bits as expected

\* Mul unit - testing (through the ALU unit):

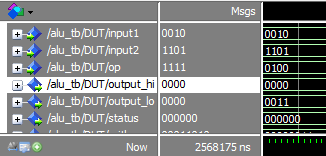
Op=”0010” mul : 1\*5



The result is 5 in the Low bits as expected

\* Abs unit - testing (through the ALU unit):

-3 in 2's complement representation



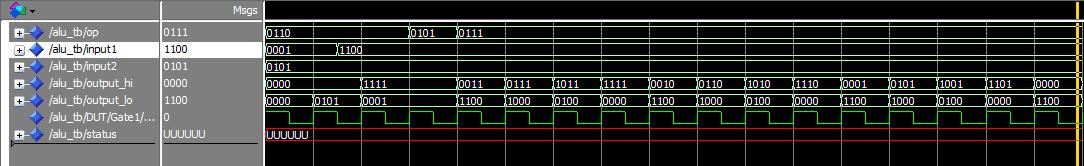
Op=”0100” abs: 1/5

`

The result is 3 in the Low bits and 0 in the Hi as expected

\* Reset Mac \ Mac - testing (through the ALU unit):

Op=”0110” mac



Reset for the mac register, mac=0

The result of mac, 5+(-4)\*5= -15 as excepted

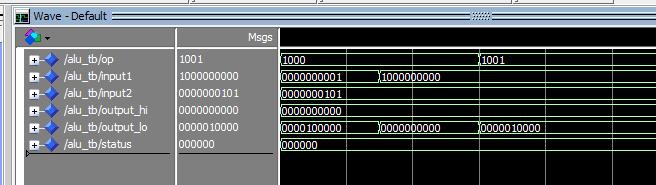
Output of mac Operation when clock rise Result is 5

Clock signal, output in respect to rising edge of clock

\*Shift Left \ Right - testing (through the ALU unit):

Op=”1001” Shift right

Op=”1000” Shift Left



Shifting left 5 times(0101) input1

Shifting right 5 times(0101) input 1

**\*Note:** all other Op codes are ignored (by the Output selector) and instead we get 0`s on all the output lines.

***13. Conclusions***

1. we have learned that when you build a complex component it is better to do it in a “structural” way – which simplifies its complexity.

2. design before writing code saves a lot of time

3. it is smarter to work top-down when working on a large project as this.

4. we used the “auto-Generate” function of the ModelSim for creating our “Test Bench” for each component/Entity – which made the testing process a lot more efficient and easy.

5.using clock for unit that their output depends on changeable inner state is necessary.

6.hardware is different than software, input on a bus for a unit has immediately output, a signal is equivalent to a real wire.

7.meannigfull name of signals and ports make the code more easy to understand after a while

8.you should be very accurate when writing a sensitive list for a process, and know exactly which change is relevant for to process to re execute.