**Assignment 5: ALU**

Roshni Uppala

1011735230

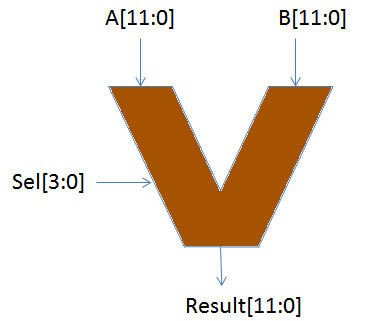
*Objective : To develop and demonstrate good design discipline and techniques in the design of combinational logic circuits using Verilog HDL . The ultimate goal in this report is to develop a Verilog HDL which supports 12-bit, two’s complement Arithmetic Logic Unit(ALU) and which would also satisfy certain functional behavior as stated further in this report.*

**Design Specifications :**

* Inputs : 12 bits A, 12 bits B and a ‘sel’ of 4 bits as selection operator
* Outputs : Result of the operation- ‘result’ ; AgtB , AltB , AeqB – 1 bit
* Functional behavior : According to the table 5.1
* Timing: Maximum propagation delay for any function shall be less than 25 ns.

**Design Structure :**

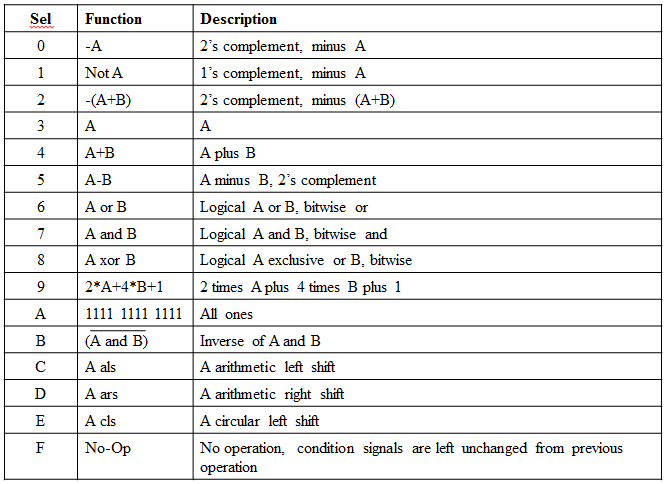
* Inputs :
* A[11:0] – 12 bit, as signed decimal value
* B[11:0] – 12 bit, as signed decimal value
* Sel[3:0] – 4 bit selection , specifies the output to perform , Given From 0-F.
* Outputs :
* Result[11:0]- 12 bit (hexadecimal )result according to the selection- sel . Table look up
* Agtb – 1 bit output if A value is greater than B, else output is ‘0’
* Altb – 1 bit output if A is less than B, else output is ‘0’
* Aeqb – 1bit output if A is equal to B , else output is ‘0’



*Fig 5.1 : ALU Block diagram*

* Functional behavior :

The functional behavior of the ALU is as per the truth table given below in table 5.1



*Table 5.1 : The functional behavior of the ALU .*

als : Arithmetic left shift : shift left and fill zeros from right.

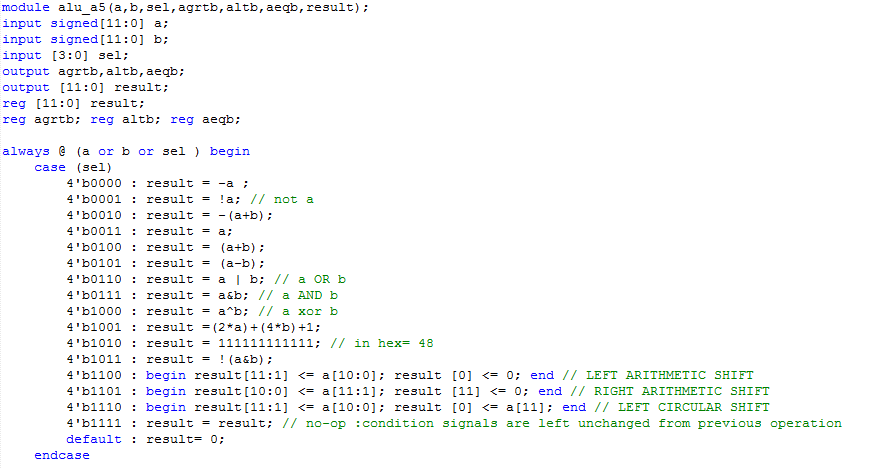
ars : Arithmetic right shift : shift right and fill zeros from left.

cls : circular shift left : All bits shits to the left by one place and the MSB moves to the LSB.

* Timing requirements: Maximum propagation delay for any function shall be less than 25 ns.

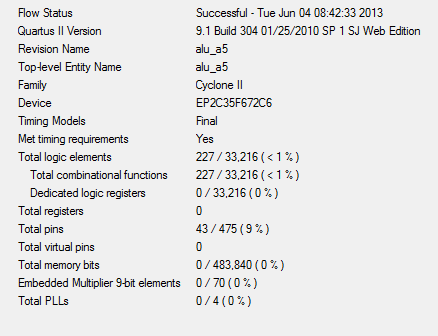
**Design Entry :**

* Verilog code:

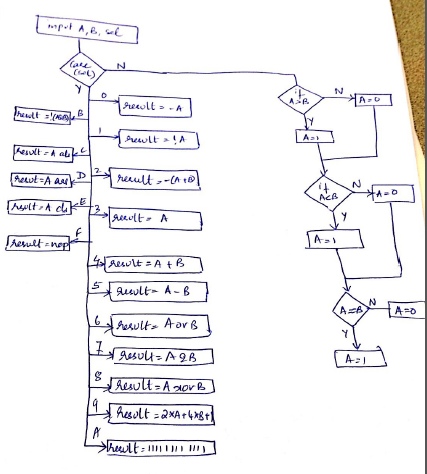




* Compilation report :



* Flowchart :



* Structure Write up :

The inputs are taken as per the tables below. The cases are selected as 0 to F to perform specific tasks as per the table in 5.1. The flow is given as in the flowchart. Respective to the “sel” , the operation is performed.

The a and b values are considered as signed decimals. They are also tested for greater, lesser and equal to.

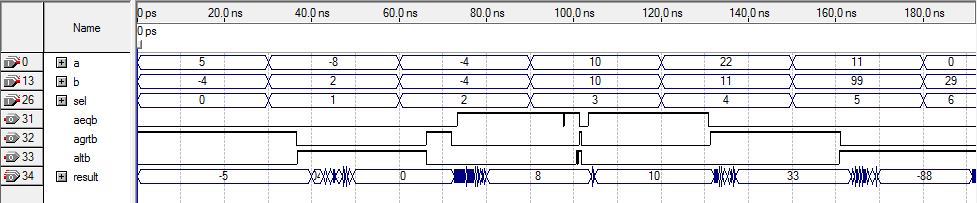
**Design verification :**

***Test Plan* :** The above code is tested for 4 tests . The case truth table is given below. The values of A and B are changed such that the desired output is obtained by specifying the desired action to be performed by specifying the ‘sel’ input value.

Test 1:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A(signed decimal)** | **B(signed decimal)** | **AGRTB(binary)** | **ALTB(binary)** | **AEQB(binary)** |
| 5 | **-4** | **1** | **0** | **0** |
| -8 | **2** | **0** | **1** | **0** |
| -4 | **-4** | **0** | **0** | **1** |
| 10 | **10** | **0** | **0** | **1** |
| 22 | **11** | **1** | **0** | **0** |
| 11 | **99** | **0** | **1** | **0** |

*Table 5.2 : truth table to check if A>B, A<B ,A==B*



glitch

A (=-4 )= B (= -4)

A (=5 )> B (= -4)

*Fig 5.2: Simulation report according to table 5.2*

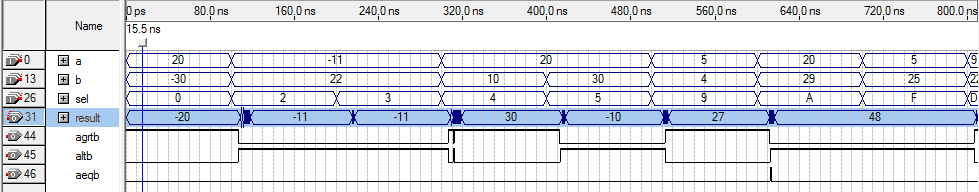
A (11 )< B (99)

Propagation delay : 8 ns

Test 2:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **A(Signed decimal)** | **B(signed decimal)** | **Result** |
| Sel=0 | 20 | -30 | -20 |
| Sel=2 | -11 | 22 | -11 |
| Sel=3 | -11 | 22 | -11 |
| Sel=4 | 20 | 10 | 30 |
| Sel=5 | 20 | 30 | -10 |
| Sel=9 | 5 | 4 | 27 |
| Sel=A | 20 | 29 | 48 |
| Sel=F | 5 | 25 | 48 |

*Table 5.3 : Truth table for the Sel bits – 0,2,3,4,5,9,A,F*



For sel= 2=( -(a+b) )= -11

*Fig 5.3 : Simulation report according to truth table 5.3*

For sel= F , value remain unchanged from prev selection.

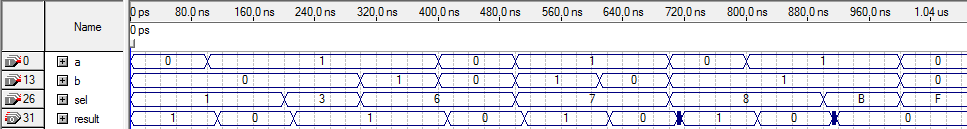
For sel= A = result = 11111111111= 48

Propagation delay : 10 ns

Test 3 :

|  |  |  |  |
| --- | --- | --- | --- |
|  | **A(Signed decimal)** | **B(signed decimal)** | **Result** |
| Sel =1 | 0 | 0 | 1 |
|  | 1 | 0 | 0 |
| Sel=3 | 1 | 0 | 1 |
| Sel-6 | 1 | 1 | 1 |
|  | 0 | 0 | 0 |
| Sel=7 | 1 | 1 | 1 |
|  | 1 | 0 | 0 |
| Sel=8 | 0 | 1 | 1 |
|  | 1 | 1 | 0 |
| Sel=B | 1 | 1 | 0 |

*Table 5.4: Truth table for the logical operations such as sel = 1,3,6,7,8,B*



A=1, b=1, A|B =1 , sel =6

A=1, b=1, !(a&b) =0 , sel =B

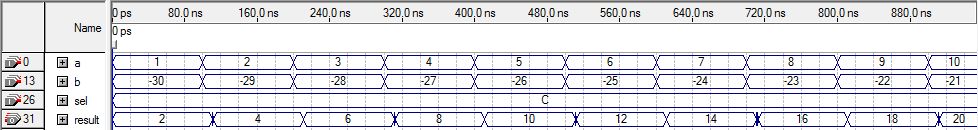
*Fig 5.4: Simulation report according to table 5.4*

Propagation delay < 25 ns

Test 4:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **A** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Sel- C** | **2** | **4** | **6** | **8** | **10** | **12** | **14** |
| **Sel-D** | **0** | **1** | **2** | **3** | **4** | **5** | **6** |

*Table 5.5: Truth table for the shift operations , sel= C*

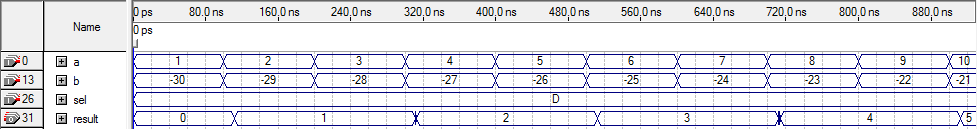


LEFT SHIFT, 0000 0000 0001 shifts to 0000 0000 0010

LEFT SHIFT, 0000 00000100shifts to 0000 0000 1000

*Fig 5.6: Left arithmetic shift using table 5.5*

Propagation delay < 25 ns



LEFT SHIFT, 0000 0000 1000 shifts to 0000 0000 0100

LEFT SHIFT, 0000 0000 0010 shifts to 0000 0000 0001

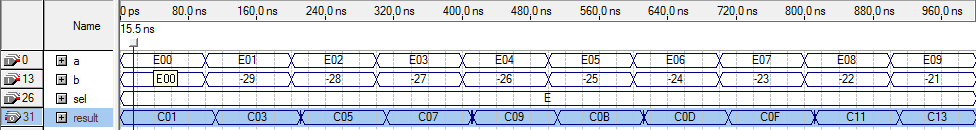
*Fig 5.7: Right arithmetic shift using table 5.5*

Propagation delay <25 ns

Test 5:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **A** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Sel- C** | **2** | **3** | **4** | **5** | **6** | **7** | **0** |

*Table 5.6 : Circular left shift truth table , sel=E*



LEFT SHIFT, 1110 0000 0000shifts to 1100 0000 0001

*Fig 5.8 : Circular left arithmetic shift using table 5.6*

Propagation delay <25 ns

**RESULT :**

The Verilog code has been designed to obtained the required operations for an ALU according to table 5.1. The simulation reports are attached as fig 5.2-8. All are maintained with a maximum propagation delay of less than 25 ns.