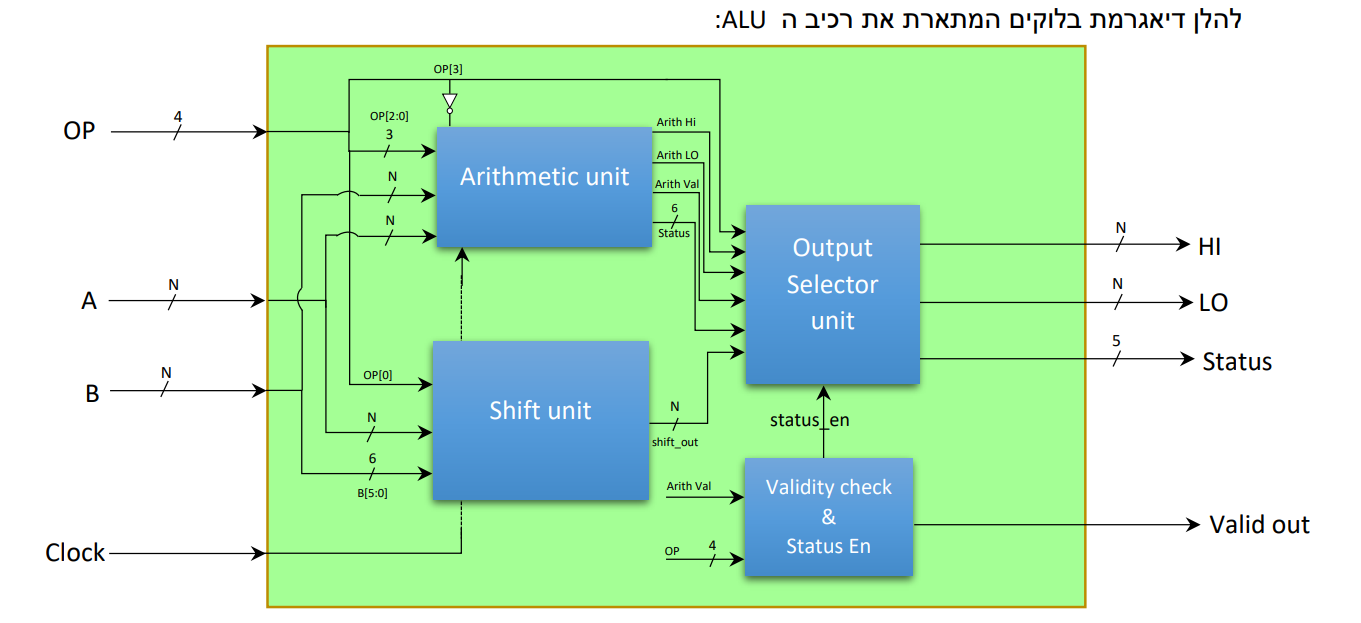
**1. Module description:**

* 1. **ALU top**:

The ALU top is composed of three main components:

* Arithmetic unit (Synchronous)
* Shift unit
* Output selector

In addition in the ALU top we compute the value of Valid out (described in table 2).



**Figure 1:**

**ALU top Schematic**

1.1.1 ALU Ports:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | In\Out | Size[bits] | Functionality |
| A | In | N | Operand |
| B | In | N | Operand |
| OP | In | 4 | OP code input (described in table 2) |
| clk | In | 1 | Clock in (for Arithmetic unit) |
| valid | Out | 1 | Valid = 0 ⟹ Outputs are Valid  Valid = 1 ⟹ Outputs are not Valid |
| HI | Out | N | Upper bits of result |
| LO | Out | N | Lower bits of result |
| status | Out | 6 | Status in case of sub operation.  Flag meanings:  Status[0] – A = B  Status[1] – A ≠ B  Status[2] – A ≥ B  Status[3] – A > B  Status[4] – A ≤ B  Status[5] – A < B |

**Table 1:**

**ALU Port description**

1.1.2 ALU Operations:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| OP code | Operation | HI | LO | Status | Valid out |
| 0001 | ADD | 0 |  | 0 | 1 |
| 0010 | SUB | 0 |  | Status\*\* | 1 |
| 0011 | MIN | 0 | Min{A,B} | 0 | 1 |
| 0110 | MAX | 0 | Max{A,B} | 0 | 1 |
| 0100 | MUL | (A×B)[2N-1:N] | (A×B)[N-1:0] | 0 | 1 |
| 0101 | MAC\* | (MAC(t-1)+ (A×B))  [2N-1:N] | (MAC(t-1)+ (A×B))  [N-1:0] | 0 | First clk - 0  Sec clk - 1 |
| 0111 | RST\* | 0 | 0 | 0 | First clk - 0  Sec clk - 1 |
| 1000 | SHL | 0 | A <<(B mod 64) | 0 | 1 |
| 1001 | SHR | 0 | A >>(B mod 64) | 0 | 1 |
| others | NOP | 0 | 0 | 0 | 0 |

\*the MAC unit is synchronous – for the DFFs, each MAC operation takes 2 clocks.

\*\*Status for SUB operations is calculated in Arithmetic unit and Enabled by output selection unit

**Table 2:**

**ALU OP description**

1.1.3 Proper use of the ALU

- Note that for starting to use the ALU, first OP **MUST** be RST. This is because the MAC register need to be reset to avoid unknown values.

- MAC operations take 2 clock cycles

1.2 **Shift unit**:

The Shift top is composed of 63 1-bit shift units. It performs Arithmetic shifts (Left or Right) According to the cnt input.

The i'th bit of cnt is used as Enable bit for 2i 1-bit shift units.

**SHIFT\_top**

A

cnt

Right\_Left

o

N

N

6

**Figure 2:**

**SHIFT top Schematic**

1.2.1 Shift top Ports

|  |  |  |  |
| --- | --- | --- | --- |
| Name | In\Out | Size[bits] | Functionality |
| A | In | N | Operand |
| cnt | In | 6 | Number of bits to shift A by. |
| Right\_Left | In | 1 | 1 – Shift Right  0 – Shift Left |
| o | Out | N | A << cnt |

**Table 3:**

**SHIFT top Ports**

1.2.2 **1-bit Shift** unit description:

**Nbit\_SHIFT**

A

En

Right\_Left

o

N

N

**Figure 3:**

**Nbit\_SHIFT unit Schematic**

1.2.2.1 **1-bit Shift** unit Ports:

**Table 4:**

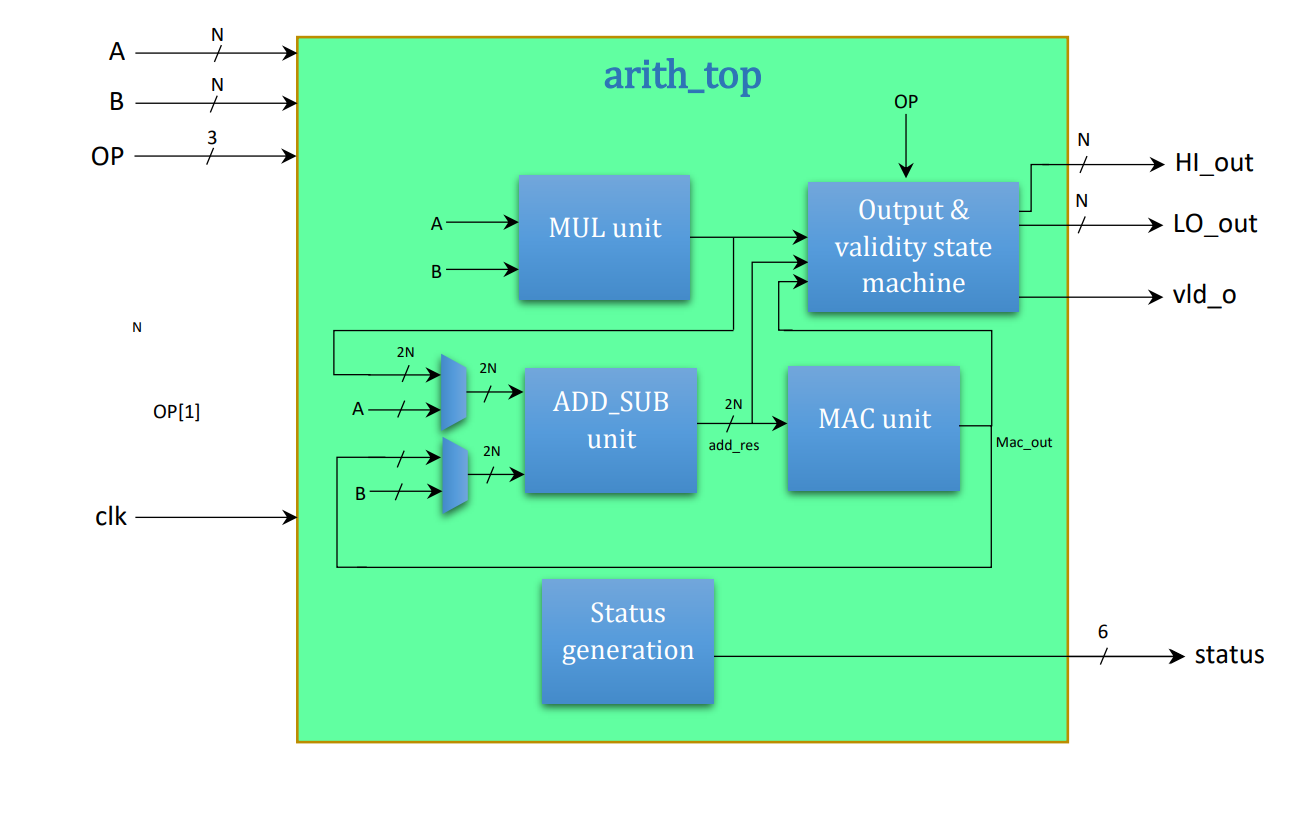
**Nbit\_SHIFT Ports**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | In\Out | Size[bits] | Functionality |
| A | In | N | Operand |
| En | In | 1 | 0 – don’t shift  1 - shift |
| Right\_Left | In | 1 | 1 – Shift Right  0 – Shift Left |
| o | Out | N | (A << cnt AND En) OR A |

1.3 **Arithmetic unit**:

The unit in charge of all the arithmetic operations

the Mac unit is synchronous, therefor the unit gets a clock as an input.



clk

1.3.1 Arithmetic top ports

|  |  |  |  |
| --- | --- | --- | --- |
| Name | In\Out | Size[bits] | Functionality |
| A  En | In | N | Operand |
| B | In | N | Operand |
| OP | In | 4 | OP code input (described in table 2) |
| clk | In | 1 | Clock in (for Arithmetic unit) |
| valid | Out | 1 | Valid = 0 ⟹ Outputs are Valid  Valid = 1 ⟹ Outputs are not Valid |
| HI | Out | N | Upper bits of result |
| LO | Out | N | Lower bits of result |
| status | Out | 6 | Status in case of sub operation.  Flag meanings:  Status[0] – A = B  Status[1] – A ≠ B  Status[2] – A ≥ B  Status[3] – A > B  Status[4] – A ≤ B  **Figure 4:**  **Arithmetic top Schematic description**  Status[5] – A < B |

1.3.1 Arithmetic top ports:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | In\Out | Size[bits] | Functionality |
| A | In | N | Operand |
| B | In | N | Operand |
| OP | In | 3 | OP code input (described in table 6) |
| clk | In | 1 | Clock in (for Arithmetic unit) |
| En | In | 1 | Enables output and intenal operations |
| HI | Out | N | Upper bits of result |
| LO | Out | N | Lower bits of result |
| vld\_o | Out | 1 | Valid = 0 ⟹ Outputs are Valid  Valid = 1 ⟹ Outputs are not Valid  In use in case of a mac operation |
| status | Out | 6 | Status Flags for SUB operations:  Status[0] – A = B  Status[1] – A ≠ B  Status[2] – A ≥ B  Status[3] – A > B  Status[4] – A ≤ B  Status[5] – A < B |

**Table 5:**

**Arithmetic top Ports**

1.3.2 Arithmetic unit operations

The OP code for the arithmetic unit is the same as for the ALU arithmetic Ops , Using the 3 Lower bits as OP code and the Inverted MSB of the ALU OP as En

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| OP code | Operation | HI | LO | Status | Valid out |
| 001 | ADD | 0 |  | Φ(Don't care) | 1 |
| 010 | SUB | 0 |  | Status | 1 |
| 011 | MIN | 0 | Min{A,B} | Φ | 1 |
| 110 | MAX | 0 | Max{A,B} | Φ | 1 |
| 100 | MUL | (A×B)[2N-1:N] | (A×B)[N-1:0] | Φ | 1 |
| 101 | MAC | (MAC(t-1)+ (A×B))  [2N-1:N] | (MAC(t-1)+ (A×B))  [N-1:0] | Φ | First clk - 0  Sec clk - 1 |
| 111 | RST | 0 | 0 | Φ | First clk - 0  Sec clk - 1 |
| others | NOP | 0 | 0 | Φ | 0 |

**Table 6:**

**Arithmetic top operations**

1.3.3 **Multiplier** unit

Performs multiplication between two signed numbers.

**Figure 5:**

**Multiplier Schematic description**

o

2N

a

b

N

N

**MUL\_top**

1.3.3.1 Multiplier Ports:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | In\Out | Size[bits] | Functionality |
| a | In | N | Operand |
| b | In | N | Operand |
| o | Out | 2N | Result – |

**Table 7:**

**Multiplier Ports**

1.3.4 **Add & Sub** unit

Responsible for the add and sub operations and the add in the MAC operation, MSB of output is used to determine output in MIN/MAX operations and to calculate the Status output in the arithmetic unit.

The unit is composed if 2N FA.

2N

a

**Nbit\_add\_sub**

**Figure 6:**

**Add\_Sub Schematic description**

2N

res

res[1]

res[0]

res[2N]

b[2N]⨁sub

b[0]⨁sub

a[0]

b[1]⨁sub

a[2N]

a[1]

sub

2N

FA1

111

FA2N

FA0

sub

b

1.3.4.1 Add & sub Ports

|  |  |  |  |
| --- | --- | --- | --- |
| Name | In\Out | Size[bits] | Functionality |
| a | In | 2N | Operand |
| b | In | 2N | Operand |
| sub | In | 1 | 0 – Add operation  1 – Sub operation |
| res | Out | 2N | Operation's Result |

**Table 8:**

**Multiplier Ports**

1.3.4.2 FA description

**Bit\_FA**

c\_out

s

a

b

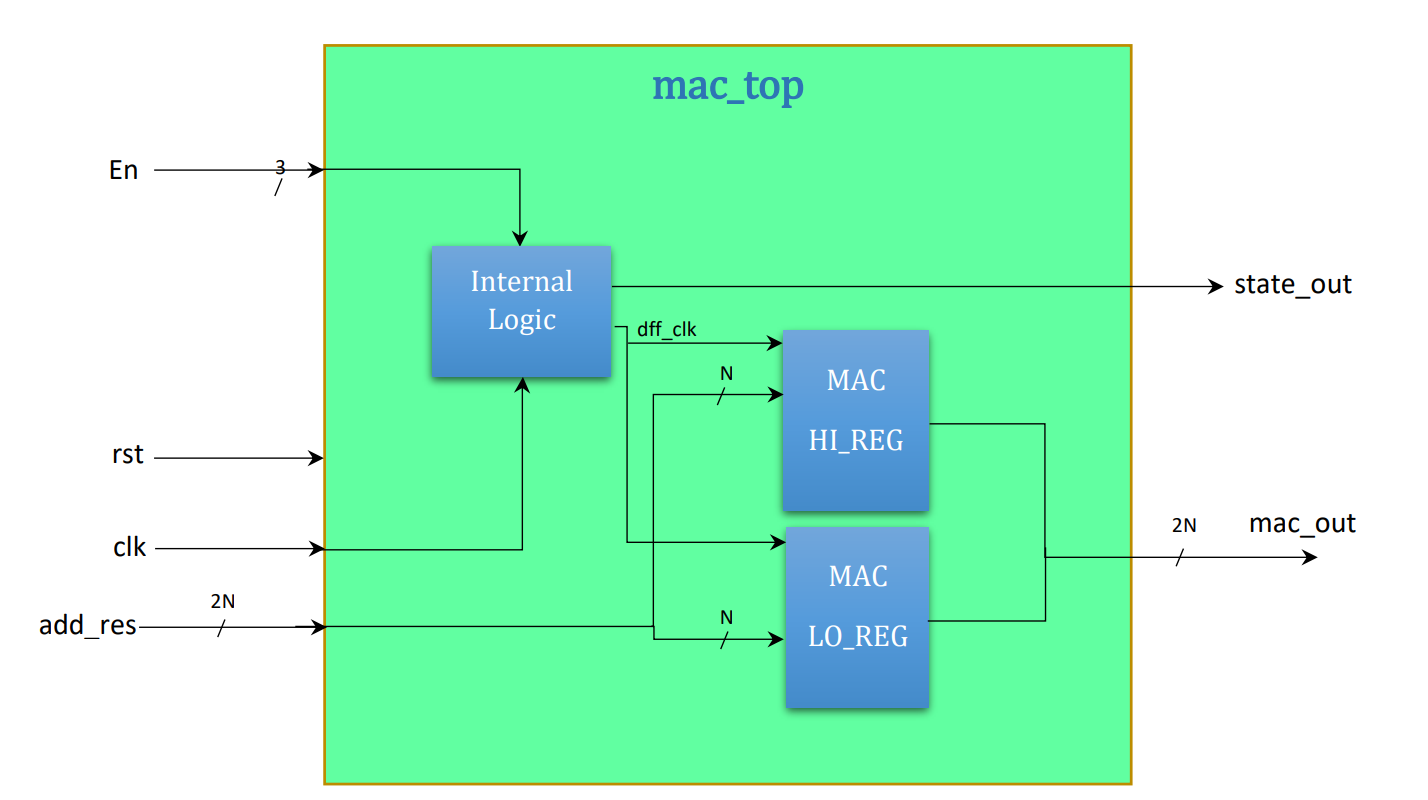
c\_in

**Figure 7:**

**FA Schematic description**

1.3.5 **MAC** unit

Multiply and Accumulate. Utilizes the Add\_Sub and Multiply components for MAC operations.



rst

rst

**Figure 8:**

**MAC Schematic description**

1.3.5.1 MAC Ports

|  |  |  |  |
| --- | --- | --- | --- |
| Name | In\Out | Size[bits] | Functionality |
| En | In | 1 | Enables mac operation |
| rst | In | 1 | Resets values in Registers |
| clk | In | 1 | Clock |
| add\_res | In | 2N | New value to save in MAC operation |
| state\_out | Out | 1 | 0 – output is not valid  1 – output is valid |
| o | Out | 2N | Current accumulated value  **Table 9:**  **MAC Ports** |

1.3.5.2 Ndff

A register made of N dffs with synchronous reset – used only in MAC top

**Ndff**

rst

d

clk

q

N

N

**Figure 9:**

**Ndff Schematic description**

1.4 **Output Select** Unit

Receives the output of the other top units and passes the right one to the ALU output

**select\_top**

arith\_HI

arith\_LO

shift\_in

status\_en

status\_in

OP

status

LO\_out

HI\_out

6

N

N

N

N

N

6

**Figure 10:**

**Select top Schematic description**

arith\_valid

valid

1.4.1 **Output Select** Ports

|  |  |  |  |
| --- | --- | --- | --- |
| Name | In\Out | Size[bits] | Functionality |
| arith\_HI | In | N | N Upper bits of arithmetic unit result |
| arith\_LO | In | N | N Lower bits of arithmetic unit result |
| shift\_in | In | N | Shift unit result |
| status\_en | In | 1 | Status enable |
| status\_in | In | 1 | Status in from arithmetic unit |
| OP | In | 4 | Uses to determine status and output |
| arith\_valid | In | 1 | Valid from arithmetic unit |
| valid | out | 1 | 0 – output is not valid  1 – output is valid |
| HI\_out | Out | N | N Upper bits of output |
| LO\_out | Out | N | N Lower bits of output |
| status | Out | 6 | Status out |

**Table 10:**

**Output Select**