1. ACCUMULATOR
2. Parameters:

Length of data: DWIDTH = 16

1. Inputs/Outputs:

* *Clock:* is produced by a clock generator, oscillates between a high and a low state.
* *Reset:* clears pending erors or event and brings to normal condition (value = 0) by setting *reset* = 1
* *Ena:* allows accumulating by setting *ena* = 1
* *Clear:* return to value 0 by setting *clear* = 1
* *Din:* sequence input data for accumulating, with each data’s length is the parameter *DWIDTH*
* *Dout:* output value of accumulated process, DWIDTH bit length.

1. How it works:

Add a sequence of numbers that need to accumulate. The numberical value in the accumulator increases as each number is added.



1. ACCUMULATOR\_MEM
2. Parameters:

Length of data: DWIDTH = 16

Number of neurons: NO\_NEURONS = 2

1. Inputs/Outputs:

* *Clock:* is produced by a clock generator, oscillates between a high and a low state.
* *Reset:* clears pending erors or event and brings to normal condition (value = 0) by setting *reset* = 1
* *Next:* allows current value to be stored and move to next register for accumulating.
* *Clear:* return to value 0 by setting *clear* = 1
* *Din:* sequence input data for accumulating, with each data’s length is the parameter *DWIDTH*
* *Dout:* output value of accumulated process, DWIDTH bits length.

1. How it works

It writes the result of each calculation of accumulating to each register, that could be read right back again for use in the next operation.



1. ACTIVATION\*
2. Parameters:

Length of data: DWIDTH = 16

Length of address: AWIDTH = 8

Position of floating-point: SP = 6

1. Inputs/Outputs:

* *Clock:* is produced by a clock generator, oscillates between a high and a low state.
* *Ena:* Enable to read
* *Net\_value:* value of sigmoid function
* *Activation:* output value for sigmoid function

1. How it works

*AWIDTH* bits memory to store value of sigmoid calculation

|  |  |
| --- | --- |
| **Addr (*AWIDTH* bits)** | **Data (*DWIDTH* bits)** |
| 0­d 🡪 0b | 0.5d 🡪 00­\_0000.1000­\_00b |
| .  .  . | .  .  . |
| 7.93750d 🡪 0111.1111b | 0.9990d 🡪 00\_0000.0000\_00b |
| -8.0000­d 🡪 1000.0000b | 0.0000d 🡪 00\_0000.0000\_00b |
| .  .  . | .  .  . |
| -0.06250d 🡪1111.1111b | 0.4844d 🡪 00\_0000.0111\_1100\_00b |

net\_value

ena

Dout

*\*References:*

1. BIAS
2. Parameters:

Length of data: DWIDTH = 16

Initial bias value: INITIAL\_VALUE = 123

1. Inputs/Outputs:

* *Clock:* is produced by a clock generator, oscillates between a high and a low state.
* *Reset:* clears pending erors or event and brings to normal condition (bias value = INITIAL\_VALUE) by setting *reset* = 1
* *W\_ena:* Enable to write
* *Din:* input bias data, DWIDTH bits length.
* *Dout:* output value, DWIDTH bits length.

1. How it works

This module works as a register, storing bias value.



1. BUFFER\_MEM
2. Parameters:

Length of data: DWIDTH = 16

Number of elements in register: ELEMENTS = 256

1. Inputs/Outputs:

* *Clock:* is produced by a clock generator, oscillates between a high and a low state.
* *Reset:* clears pending erors or event and brings to normal condition (writing address = 0 and reading address = 0) by setting *reset* = 1
* *Ena\_in:* allows incoming data stream.
* *Ena\_out:* allows outgoing data stream.
* *Din:* input data stream for calculation, DWIDTH bits length
* *Dout:* output value of accumulated process, DWIDTH bits length.

1. How it works

This module stores the data being transferred to and from the immediate access storage.

A data item will be copied to the module ready for use at the next clock cycle, when it can be either used by the processor for reading or writing or stored in main memory after being written.



1. DELTA
2. Parameters:

Length of data: DWIDTH = 16

Position of floating-point: SP = 6

1. Inputs/Outputs:

* *Clock:* is produced by a clock generator, oscillates between a high and a low state.
* *Reset:* clears pending erors or event and brings to normal condition (register value = 0) by setting *reset* = 1
* *Load\_unit\_err:* enable to store *unit\_err* to register for next calculation
* *Act\_bus:* activation value, DWIDTH bits length
* *Unit\_err:* sequence unit error value, DWIDTH bits length.
* *Delta:* output value after calculating process.

1. How it works

This module provides value of the error at hidden units that would use for future calculation very slightly every time they are updated



1. DIFF\_ACT
2. Parameters:

Length of data: DWIDTH = 16

Position of floating-point: SP = 6

1. Inputs/Outputs:

* *Act:* Activation value
* *Diff\_act:* differential of activation function

1. How it works

Helping calculating the derivative of the sigmoid function is the sigmoid function times one minus itself.

With S(x) is the activation (sigmoid) function.



1. PISO\_REG
2. Parameters:

Length of data: DWIDTH = 16

Number of neurons: NO\_NEURONS = 2

1. Inputs/Outputs:

* *Clock:* is produced by a clock generator, oscillates between a high and a low state.
* *Reset:* clears pending erors or event and brings to normal condition (data in register = 0) by setting *reset* = 1
* *Ena\_in:* allows incoming data stream.
* *Shift:* parallel load if low, shift if high
* *Short\_switch:* saving 1 clock cycle by exporting output data in the 1st clock cycle
* *In\_bus:* parallel input data, DWIDTH bits length
* *Out\_bus:* serial output data, DWIDTH bits length

1. How it works

The parallel data is loaded into the register simultaneously and is shifted out of the register serially one bit at a time under clock control.

Save the 1st clock cycle to shift out the 1st data by setting *short\_switch* signal to high.



1. SUM\_OF\_PRODUCT
2. Parameters:

Length of data: DWIDTH = 16

Length of possible extended data: EXTEND = 4

Position of floating-point: SP = 6

1. Inputs/Outputs:

* *Clock:* is produced by a clock generator, oscillates between a high and a low state.
* *Reset:* clears pending erors or event and brings to normal condition (sum = 0) by setting *reset* signal to high.
* *Ena:* allows incoming data stream.
* *Pre\_clear:* allows calculating
* *Weight:* weight value, DWIDTH bit-length
* *Activation:* activation value, DWIDTH bit-length
* *Bias:* bias value, DWIDTH bit-length

1. How it works

Consider a neuron.

to check the Y value produced by a neuron and decide whether outside connections should consider this neuron as “fired” or not. Or rather let’s say — “activated” or not.



1. UNIT\_ERR\_B\_HIDDEN
2. Parameters:

Length of data: DWIDTH = 16

Number of output neurons: NO\_NEURONS\_OUT = 3

Position of floating-point: SP = 6

1. Inputs/Outputs:

* *Clock:* is produced by a clock generator, oscillates between a high and a low state.
* *Reset:* clears pending erors or event and brings to normal condition (value in the register = 0) by setting *reset* state to high.
* *Ena:* enable to calculate.
* *Act\_bus:* activation value, DWIDTH bit-length
* *Weight\_bus:* weight value, DWIDTH bit-length
* *Unit\_bus:* unit value, DWIDTH bit-length
* *Unit\_err:* unit error value after calculating.

1. How it works

This module calculates the value of unit error in the hidden layers, that equals to the sum of previous layers’ unit error multiply by the corresponding weight value.

*Diff\_act* is the module calculating the derivative of the sigmoid function that is the sigmoid function times one minus itself (see *7. Diff\_act*)



1. UNIT\_ERROR\_B\_OUT
2. Parameters:

Length of data: DWIDTH = 16

Position of floating-point: SP = 6

1. Inputs/Outputs:
   * *Clock:* is produced by a clock generator, oscillates between a high and a low state.
   * *Reset:* clears pending erors or event and brings to normal condition (value in the register = 0) by setting *reset* state to high.
   * *Ena:* enable to calculate.
   * *Act\_bus:* activation value, DWIDTH bit-length
   * *Sup\_bus:* supervisor value, DWIDTH bit-length
   * *Unit\_err:* output unit error value after calculating
2. How it works

This module calculates the value of unit error in output layer that equals to the difference between activation value and supervisor value, multiplied by the derivative of the sigmoid function.

*Diff\_act* is the module calculating the derivative of the sigmoid function that is the sigmoid function times one minus itself (see *7. Diff\_act*)



1. WEIGHT
2. Parameters:

Length of data: DWIDTH = 16

Length of address: AWIDTH = 8

1. Inputs/Outputs:
2. How it works



1. CONTROL ALGORITHM

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Phase 1 | | | | Phase 2 | | | | Phase 3 | | | | Phase 4 | | | | Phase 5 | | | |
|  | W(h) | |  | A(h) |  |  |  |  |  | ADB(o) |  |  |  |  |  |  | UW(o) | | |  |
|  |  |  |  |  | W(o) | | | A(o) |  |  |  |  |  |  |  |  | UW(h) | |  |  |
| Load | |  |  |  |  |  |  |  | E(o) | E(h) | | |  |  |  |  | U(b) |  |  |  |
|  |  |  |  | Load | | |  |  |  |  | ADW(h) | | | |  |  |  |  |  | Done |
|  |  |  |  |  |  |  |  |  |  | LEo |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  | ADW(o) | | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  | LE(h) | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  | ADB(h) | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

1 clock cycle