

The 21st

LSI 2018

Design Contest
In Okinawa

Neural Network and HW
Design

LSI Design Contest
2017/11/10

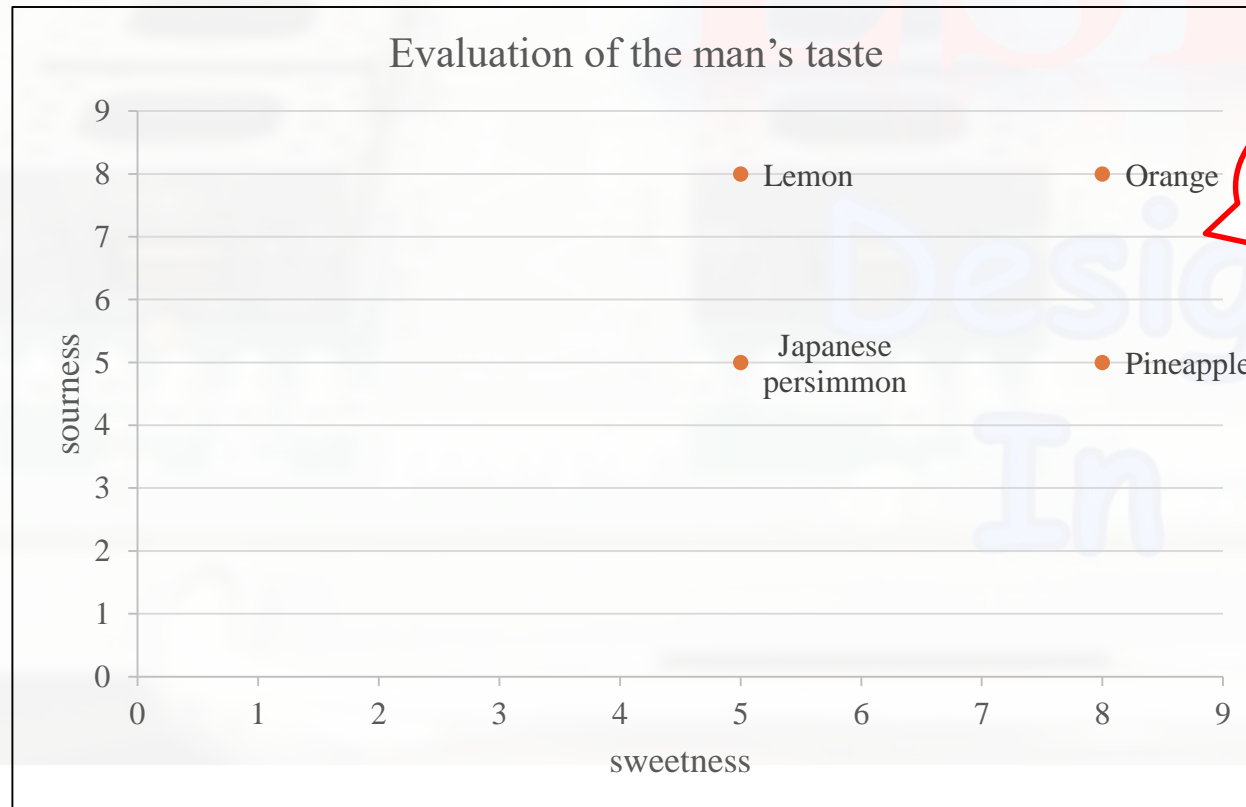


Introduction

- This is an example of Neural network HW design.
- The neural network structure used here is 3-layer structure.
- It consist of 2 input units, 3 hidden units and 2 output units.

State condition

- 4 types of fruits : Orange, lemon, pineapple and Japanese persimmon
- A man eat these 4 types of fruits and decide the level of sweetness and sourness of the fruits from the range of 0 to 10
- After deciding the level of sweetness and sourness, he then decide which fruits he likes and not which fruit he dislikes
- So let's consider the fruits he likes as $[1,0]$ and the fruits he dislike as $[0,1]$



How to measure the man's taste using Neural Network?

Hardware specification

$i = 1,2,3$
 $j = 1,2$
 $n = 2,3$
 $m = 1,2,3,4$

forward process

$$\bullet \quad z_i^2 = w_{1i}^2 k_1 + w_{2i}^2 k_2 + b_i^2 \quad (\text{z2})$$

$$\bullet \quad a_i^2 = a(z_i^2) = \frac{1}{1+e^{-z_i^2}} \quad (\text{a2})$$

$$\bullet \quad z_j^3 = w_{1j}^3 a_1^2 + w_{2j}^3 a_2^2 + w_{3j}^3 a_3^2 + b_j^3 \quad (\text{z3})$$

$$\bullet \quad a_j^3 = a(z_j^3) = \frac{1}{1+e^{-z_j^3}} \quad (\text{a3})$$

k or t is read from memory recursively 11 clk after.



backward process

$$\bullet \quad a'(z_i^n) = \frac{e^{-z_i^n}}{(e^{-z_i^n} + 1)^2} = ((1 - a_i^n) a_i^n) \quad (\text{dadz})$$

$$\bullet \quad \delta_j^3 = (a_j^3 - t_j) a_j^{3'}(z_j^3) \quad (\text{delta3})$$

$$\bullet \quad \delta_i^2 = (\delta_1^3 w_{1i}^3 + \delta_2^3 w_{2i}^3 + \dots) a_i^{2'}(z_i^2) \quad (\text{delta2})$$

$$\bullet \quad \frac{\partial C}{\partial w_{ij}^n} [m] = \delta_j^n a_i^{n-1} [m], \quad \text{however, } a_i^1 = K_i \quad (\text{dw2, dw3})$$

$$\bullet \quad \frac{\partial C}{\partial b_j^n} [m] = \delta_j^n [m] \quad (\text{db2, db3})$$

$$\bullet \quad \Delta w_{ij}^n = -\eta \frac{\partial C}{\partial w_{ij}^n} = -\eta \left(\frac{\partial C}{\partial w_{ij}^n} [1] + \dots + \frac{\partial C}{\partial w_{ij}^n} [m] \right) \quad (\text{dw_adder_w2, dw_adder_w3})$$

$$\bullet \quad \frac{\partial C}{\partial b_j^n} = \frac{\partial C}{\partial b_j^n} [1] + \dots + \frac{\partial C}{\partial b_j^n} [m] \quad (\text{db_adder_b2, db_adder_b3})$$

Forward (forward.v)

Module Name: forward

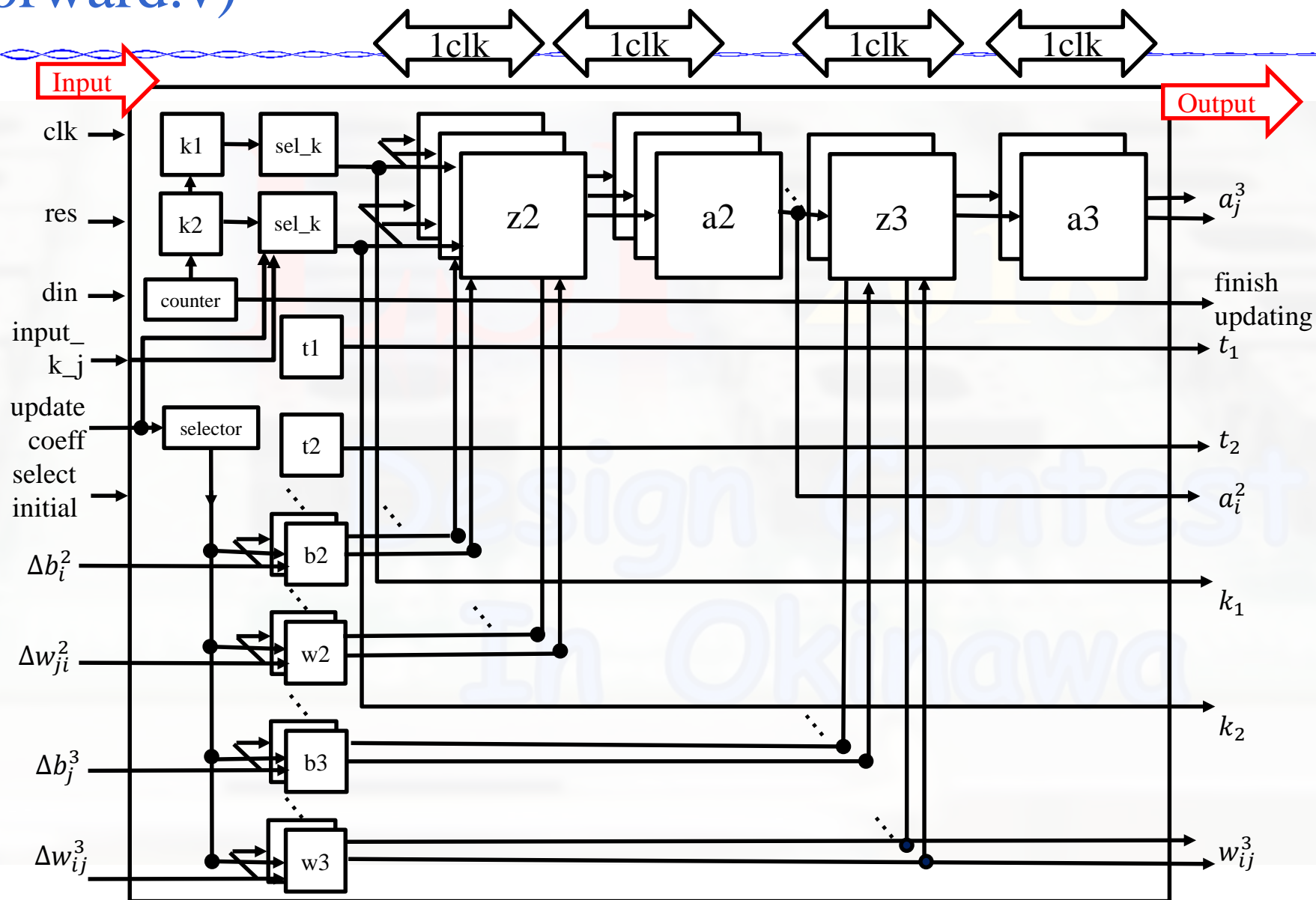
Description: Calculation of a2 and a3
($i=1,2,3, j=1,2$)

Input:

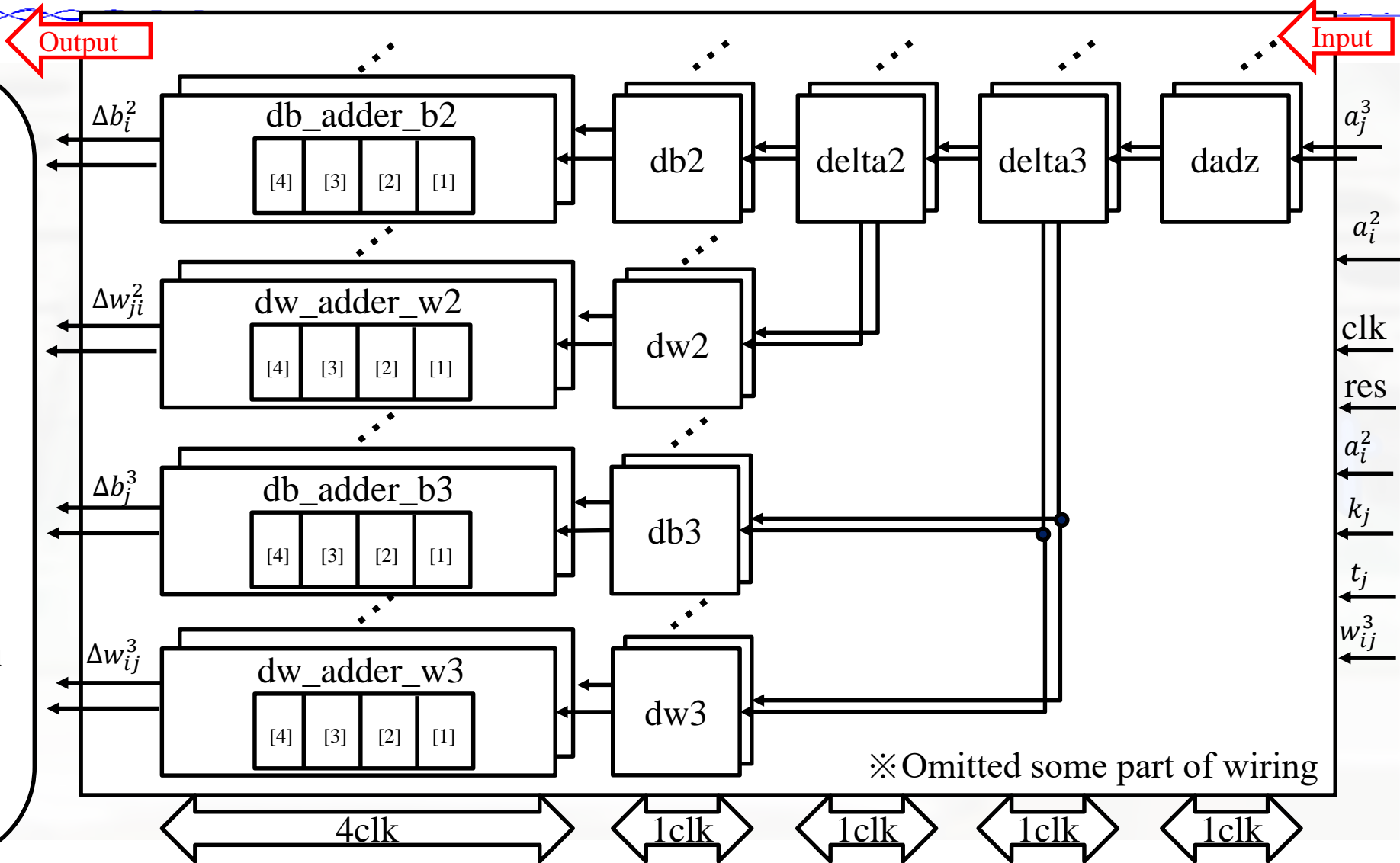
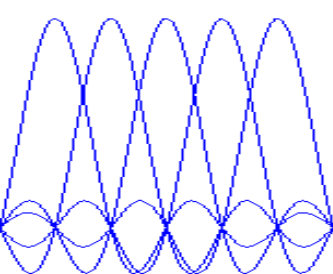
- clk : 1 bit : clock signal
- res : 1 bit : reset signal (high active)
- din : 1 bit : read enable signal
(when active, can read the data from memory)
- select_initial : 1bit : (when active, use the initial value of bias and weight.)
- update_coeff : 1 bit : coefficient to update parameter
- input_k_j : 16 bits : insert value of k
- Δw_{ij}^3 : 16bits : the amount of w_{ij}^3 changed
- Δw_{ji}^2 : 16bits : the amount of w_{ji}^2 changed
- Δb_j^3 : 16bits : the amount of b_j^3 changed
- Δb_i^2 : 16bits : the amount of b_i^2 changed

Output:

- a_j^3 : 16bits : output_layer_output
- a_i^2 : 16bits : output_layer_output
- k_j : 16bits : supervisor data
- t_j : 16bits : supervisor value
- w_{ij}^3 : 16bits : output_layer weight



Backward (backward.v)



Module Name: backward

Description:
Calculation of change in parameter (weight and bias)
($i=1,2,3, j=1,2$)

Input:
 clk : 1 bit : clock signal
 res : 1 bit : reset signal (high active)
 a_j^3 : 16 bits : output_layer output
 a_i^2 : 16 bits : hidden_layer output
 k_j : 16 bits : supervisor data
 t_j : 16 bits : supervisor value
 w_{ij}^3 : 16 bits : output_layer weight
 w_{ji}^2 : 16 bits : hidden_layer weight

Output:
 Δw_{ij}^3 : 16 bits : the amount of w_{ij}^3 changed
 Δw_{ji}^2 : 16 bits : the amount of w_{ji}^2 changed
 Δb_i^3 : 16 bits : the amount of b_i^3 changed
 Δb_j^2 : 16 bits : the amount of b_j^2 changed



List of block used in forward propagation

- Bias
- Weight for hidden layer
- Weight for output layer
- Z2
- Z3
- Memory for activation function
- Counter
- Selector
- Supervisor data
- Supervisor value
- Input selector

Bias2 block (b2_i.v (i=1,2,3))

※ Same case for bias3 block (i=1, 2)

Module Name: b2_i

Description:

Calculation of b2_i, when the select initial signal is active, the output will be the initial value of b2_i, and when the select update signal is active, the output will be the new value of b2_i

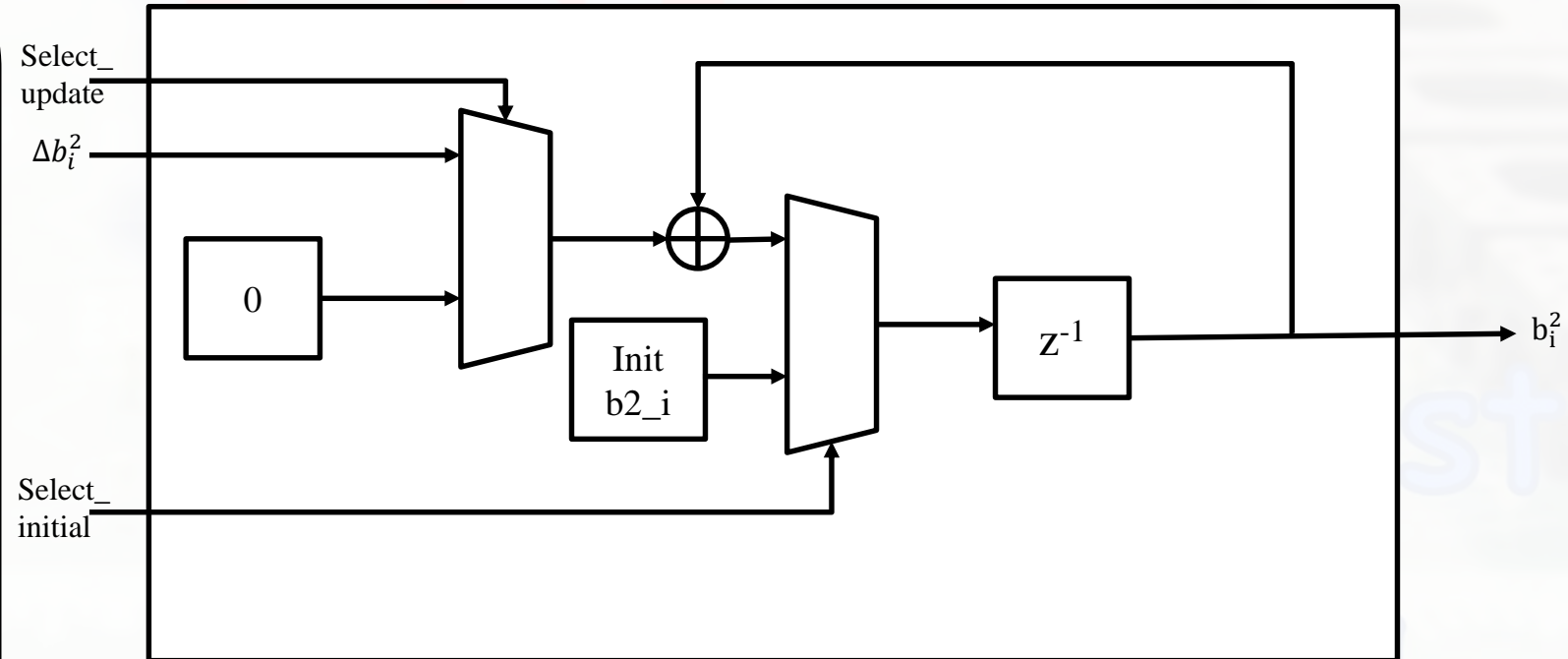
(i=1,2,3)

Input:

clk : 1 bit : clock signal
res : 1 bit : reset signal (high active)
select_update : 1 bit : signal to choose for update value
select_initial : 1 bit : signal to choose for initial value
 Δb_i^2 : 16bits : the amount of b_i^2 change

Output:

b_i^2 : 16bits : the amount of b_i^2 change



Weight for hidden layer block ($w_{2_ij.v}$ ($i=1,2, j=1,2,3$))

Module Name: w_{2_ij}

Description:

Calculation of w_{2_ij} , when the select initial signal is active, the output will be the initial value of w_{2_ij} , and when the select update signal is active, the output will be the new value of w_{2_ij}

($i=1,2,3, j=1,2$)

Input:

clk : 1 bit : clock signal

res : 1 bit : reset signal (high active)

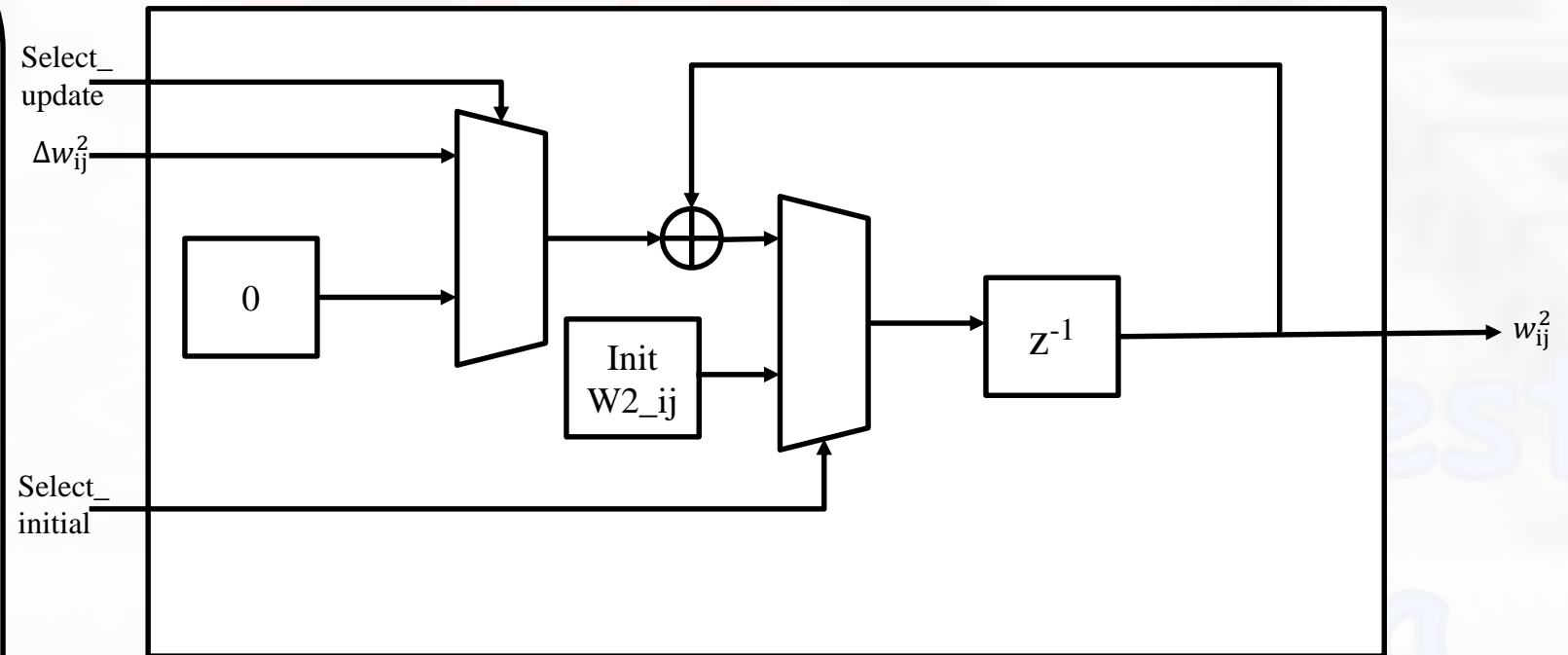
select_update : 1 bit : signal to choose for update value

select_initial : 1 bit : signal to choose for initial value

Δw_{ij}^2 : 16bits : the amount of w_{ij}^2 change

Output:($i=1,2,3, j=1,2$)

w_{ij}^2 : 16bits : the amount of w_{ij}^2 change



Weight for output layer block (w_{ij}^3 ($i=1,2,3$, $j=1,2$))

Module Name: w_{ij}^3

Description:

Calculation of w_{ij}^3 , when the select initial signal is active, the output will be the initial value of w_{ij}^3 , and when the select update signal is active, the output will be the new value of w_{ij}^3

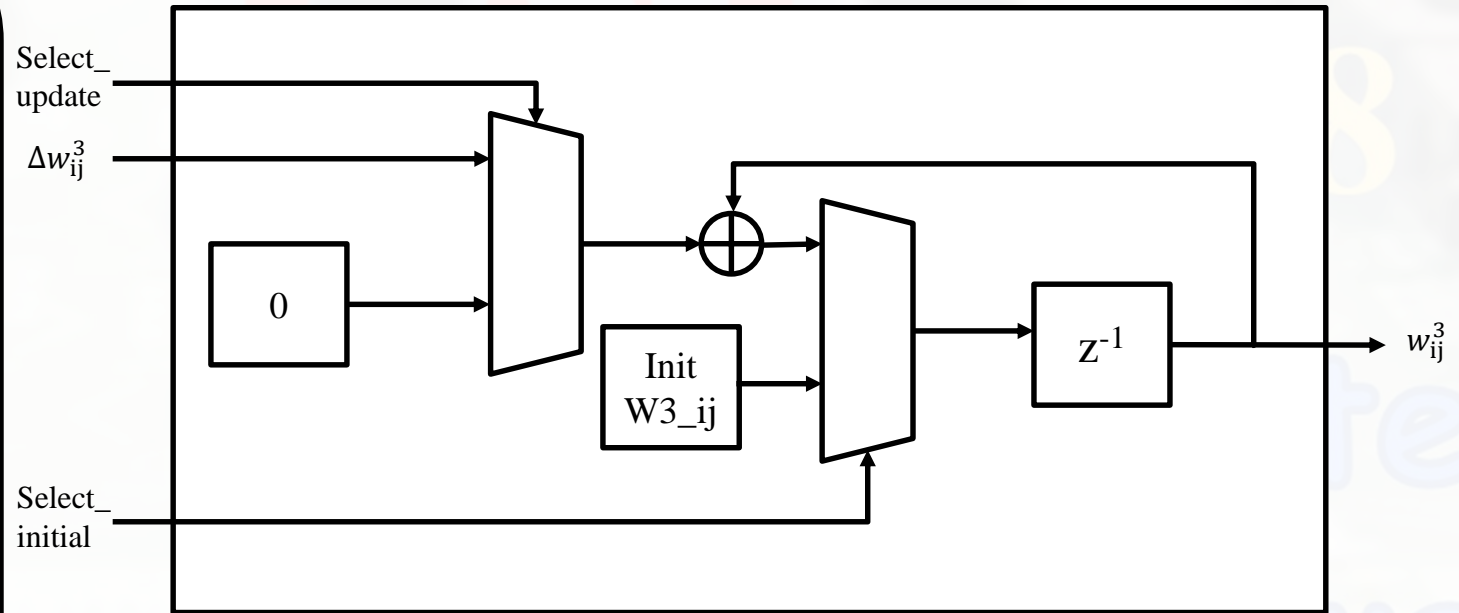
($i=1,2,3$, $j=1,2$)

Input:

clk : 1 bit : clock signal
res : 1 bit : reset signal (high active)
select_update : 1 bit : signal to choose for update value
select_initial : 1 bit : signal to choose for initial value
 Δw_{ij}^3 : 16bits : the amount of w_{ij}^3 change

Output:

w_{ij}^3 : 16bits : the amount of w_{ij}^3 change



Z2 (z2.v)

Module Name: z2

Description:

Calculation of 2 input (k_1, k_2) and 1 output (z^2)

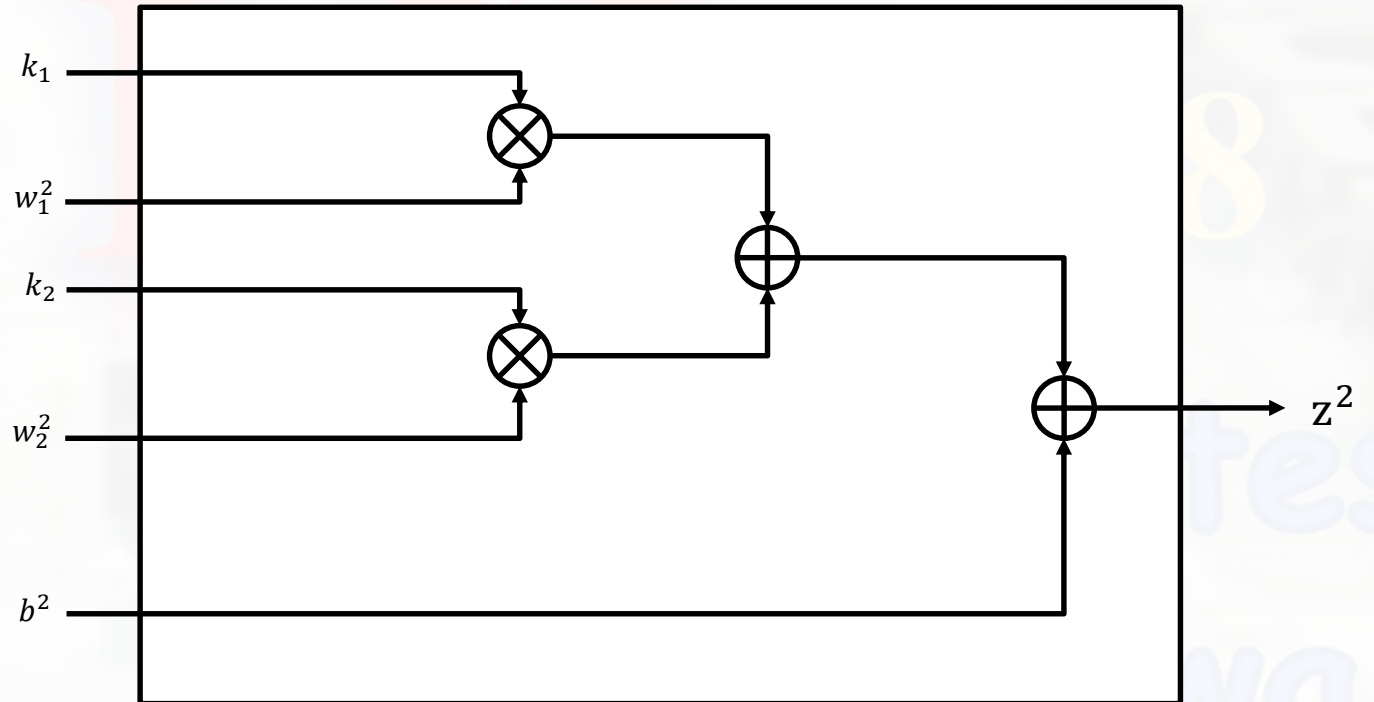
($i=1,2$)

Input:

clk : 1 bit : clock signal
res : 1 bit : reset signal (high active)
 k_i : 16 bits : supervisor data
 w_i^2 : 16 bits : hidden_layer weight
 b^2 : 16 bits : hidden_layer bias

Output:

z^2 : 16 bits : total product of calculation between input(k), weight and bias



Z3 (z3.v)

Module Name: z3

Description:

Calculation of 3 input (a2) and 1 output (z3)

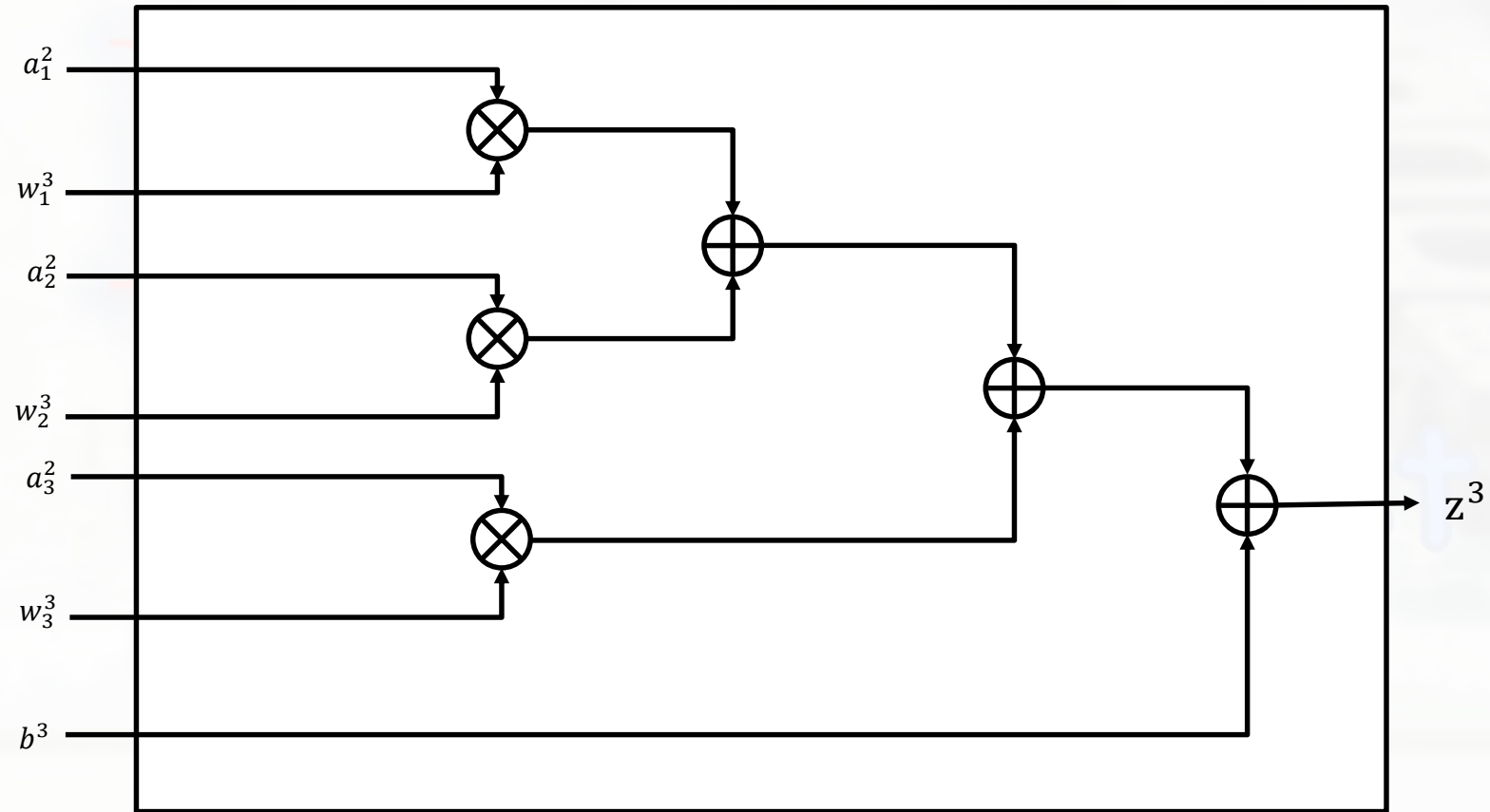
(i=1,2,3)

Input:

clk : 1 bit : clock signal
res : 1 bit : reset signal (high active)
 a_i^2 : 16 bits : output value from hidden layer
 w_i^3 : 16 bits : output layer weight
 b^3 : 16 bits : output layer bias

Output:

z^3 : 16 bits : total product of calculation between input(a2), weight and bias



Memory for activation func. (mem.v)

Module Name: mem

Description:

8 bits memory to store value of sigmoid calculation

Input:

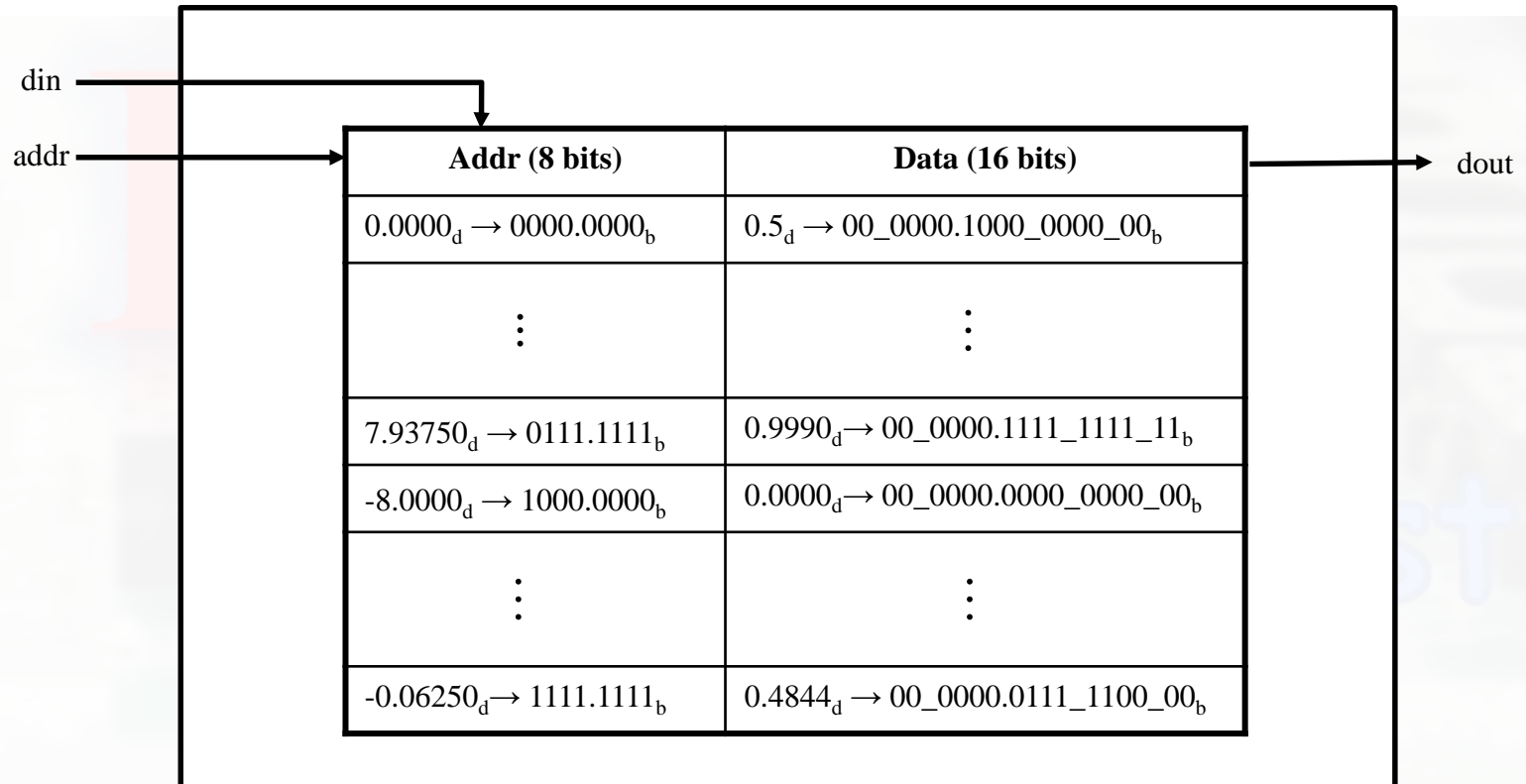
clk : 1 bit : clock signal

din : 1 bit : read enable

addr : 8 bits : value of sigmoid func. with input from -8.0 to 7.93750

Output:

dout : 16bits : output value for sigmoid func.



Counter (counter.v)

Module Name: counter

Description:

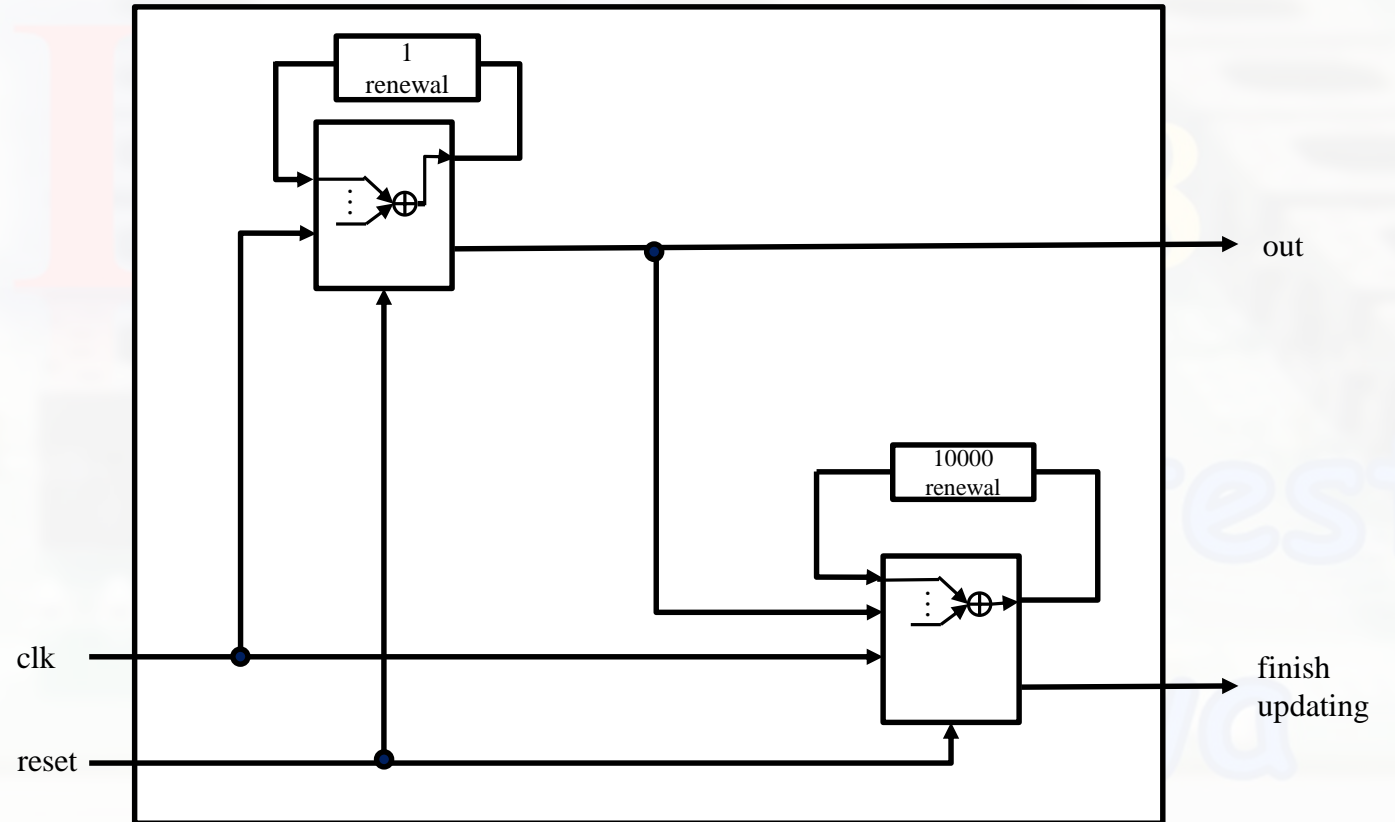
counter for signal to insert in k1, k2, t1 and t2 every 13 clock and signal for finished updating the parameter (weight and bias)

Input:

clk : 1 bit : clock signal
res : 1 bit : reset signal (high active)

Output:

out : 4 bits : output signal active after 13 clocks (1 renewal)
finish updating : 1 bit : finish updated signal after 10000 renewal



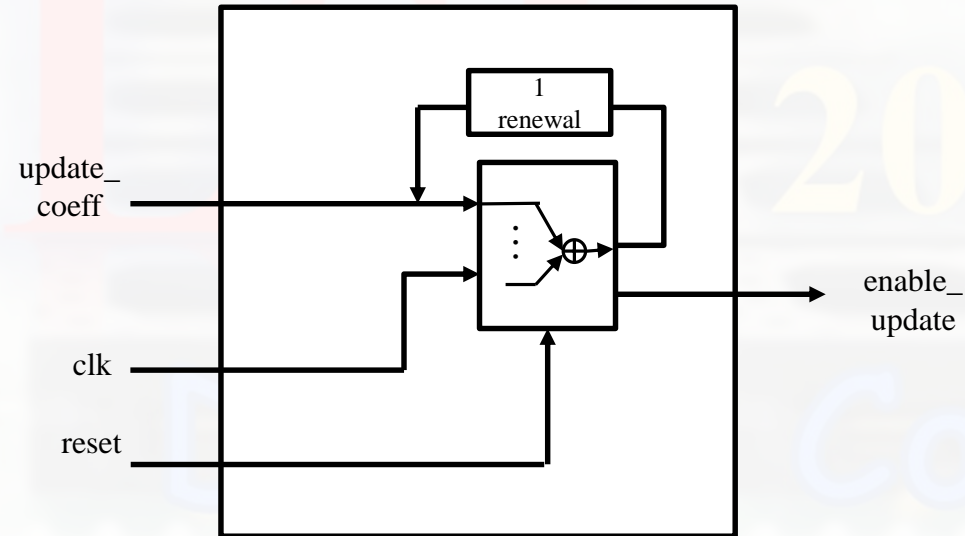
Selector (selector.v)

Module Name: selector

Description:
selector for enable update for every renewal

Input:
clk : 1 bit : clock signal
res : 1 bit : reset signal (high active)
update_coeff : 1 bit : coefficient to update

Output:
enable update : 1 bits : output signal active after 13
clocks (1 renewal)



Supervisor data (k1.v)

※ Same case for k2 = 8, 5, 8, 5, 0...

Module Name: k1

Description:

8 bits memory to store value of k1

Input:

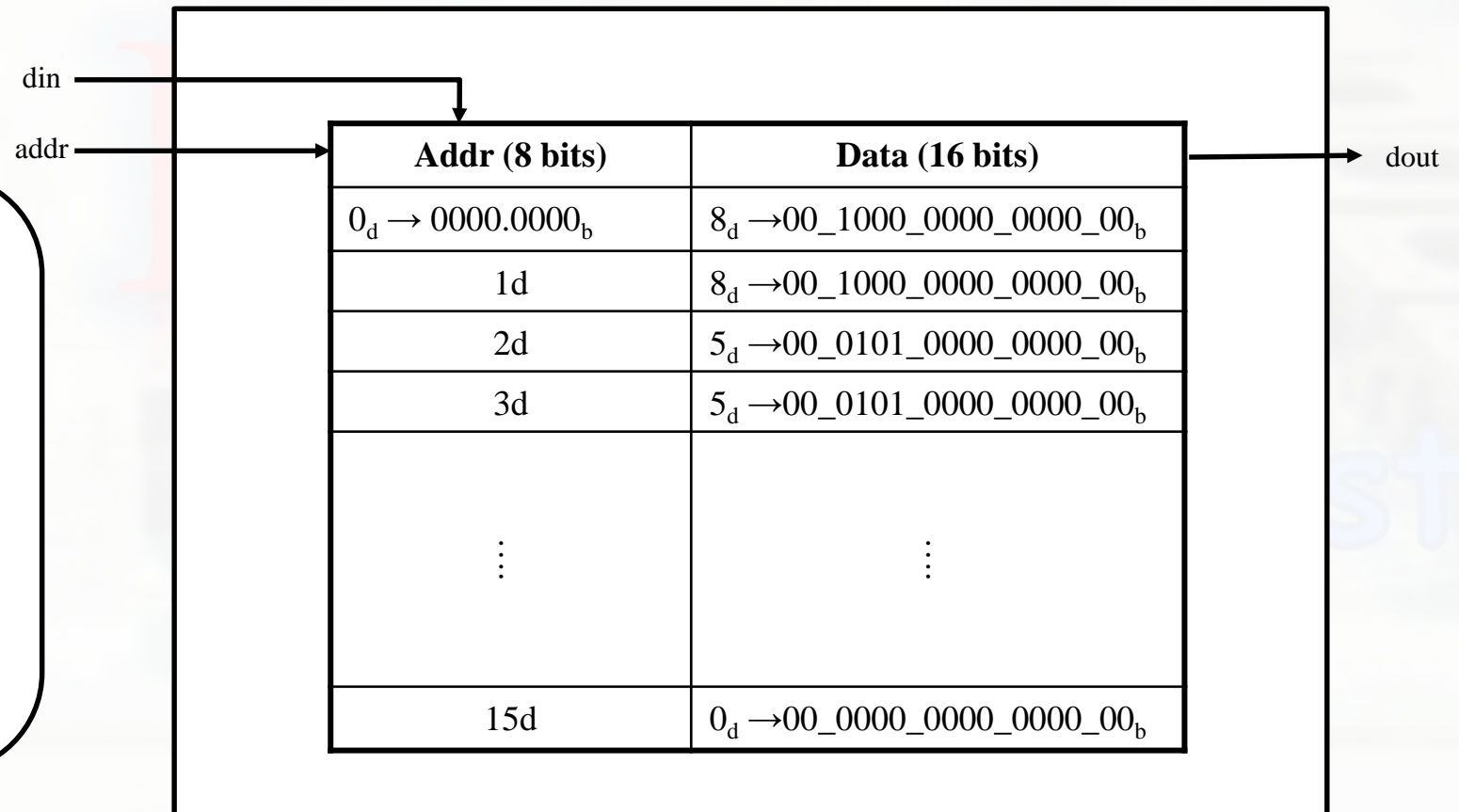
clk : 1 bit : clock signal

din : 1 bit : read enable

addr : 8 bits count number of clock with input from 0 to 15

Output:

dout : 16 bits : output value for k1



Supervisor value (t1.v)

※ Same case for t2 = 0, 1, 1, 1, 0...

Module Name: t1

Description:

8 bits memory to store value of t1

Input:

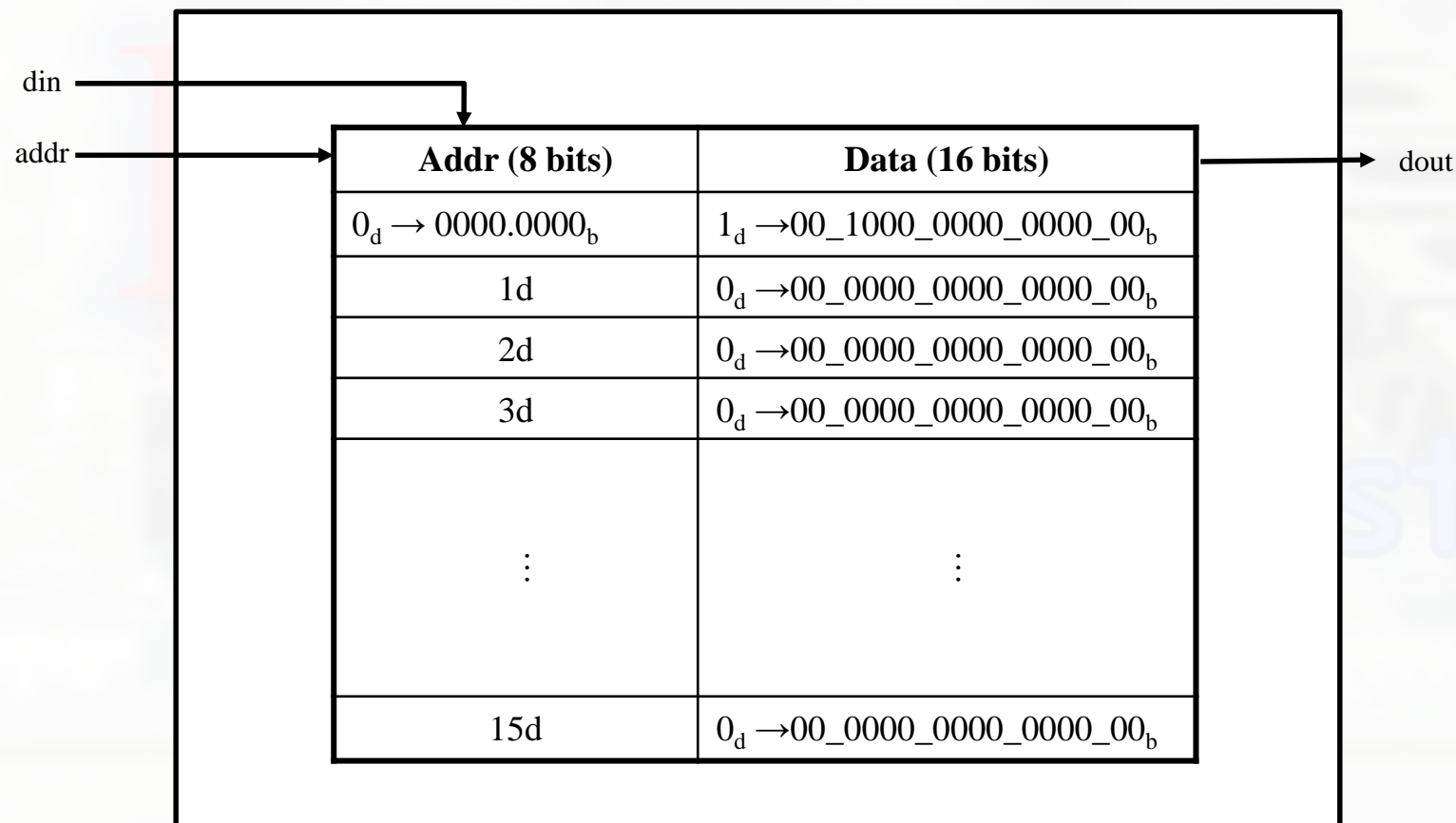
clk : 1 bit : clock signal

din : 1 bit : read enable

addr : 8 bits count number of clock with input from 0 to 15

Output:

dout : 16 bits : output value for t1



Input selector (sel_k.v)

Module Name: sel_k

Description:

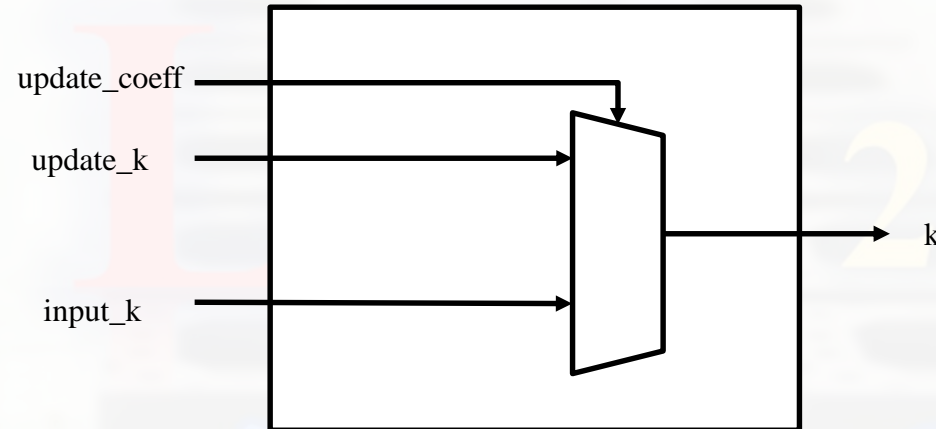
Select k from input data k or stored data k for updating coefficients.

Input:

clk : 1 bit : clock signal
res : 1 bit : reset signal (high active)
input_k : 16 bits : insert input value(k)
update_k : 16 bits : update the input value (k)
update_coeff : 1 bit : coefficient to update

Output:

k : 16 bits : when update_coeff = 1, k will be the update k, else will output the input_k





List of block used in backward propagation

- Differential of activation func.
- Delta3
- Delta2
- Delta weight
- Delta bias adder
- Delta weight adder

LSI 2018
Design Contest
In Okinawa

Differential of activation func. (dadz.v)

Module Name: dadz

Description:

Calculate differential of a

Input:

clk : 1 bit : clock signal

res : 1 bit : reset signal (high active)

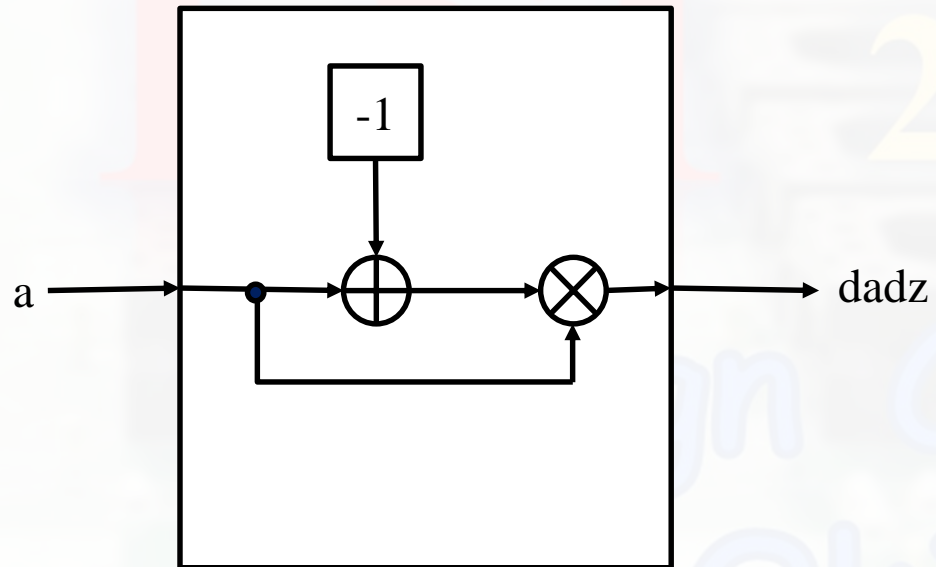
a : 16 bits : activation func. value from forward block

Output:

dadz : 16 bits : differential of a

(a : sigmoid function)

$$\frac{da}{dz} = \left(\frac{1}{1+e^{-z}}\right)' = (a - 1)a'$$



Delta3 (delta3.v)

Module Name: delta3

Description:

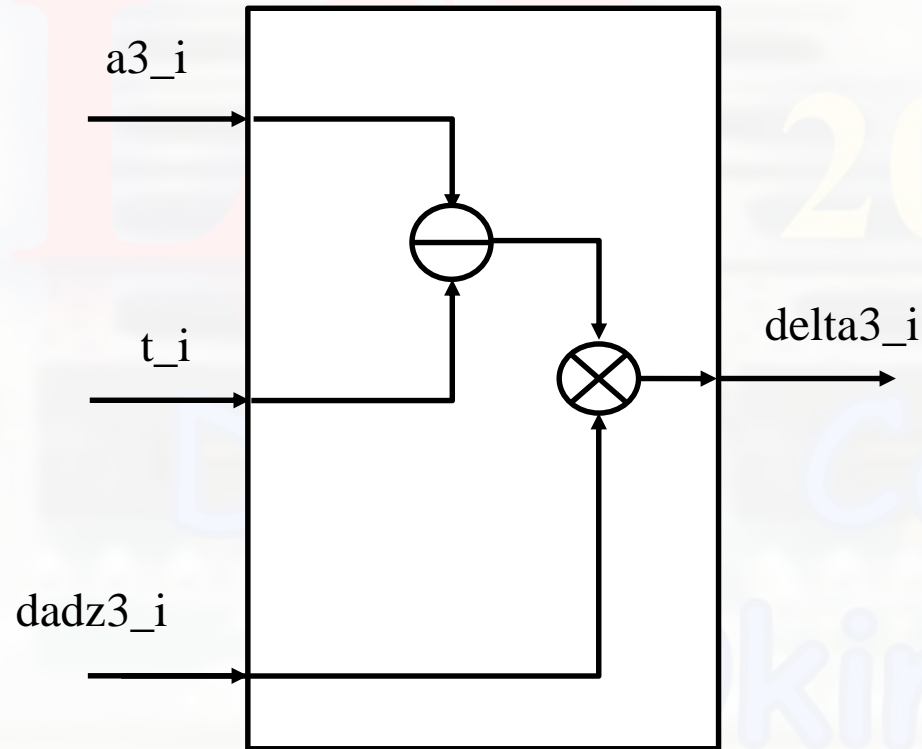
Calculation of output layer's unit error
($i=1,2$)

Input:

clk : 1 bit : clock signal
res : 1 bit : reset signal (high active)
a3_i : 16 bits : output layer's output
t_i : 16 bits : supervisor value
dadz3_i : 16 bits : differential of a3_i

Output:

delta3_i : 16 bits : output layer's error



Delta2 (delta2.v)

Module Name: delta2

Description:

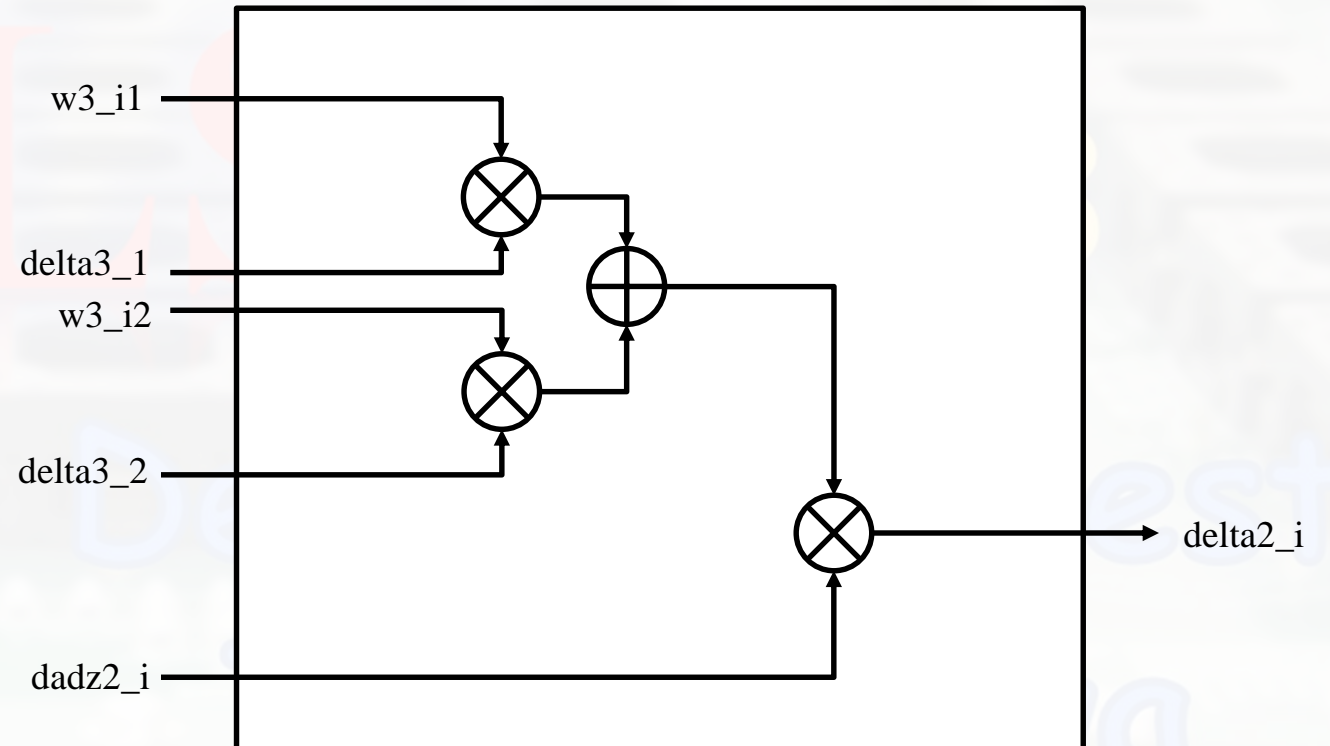
Calculation of hidden layer's unit error
($i=1,2,3$)

Input:

clk : 1 bit : clock signal
res : 1 bit : reset signal (high active)
dadz2_i : 16 bits : differential of a2
w3_i1 : 16 bits : weight of output layer from first unit
w3_i2 : 16 bits : weight of output layer from second unit
delta3_1 : 16 bits : output layer's first unit error
delta3_2 : 16 bits : output layer's second unit error

Output:

delta2_i : 16 bits : output layer's error



Delta weight (dw.v)

Module Name: dw

Description:

Calculation of hidden layer's unit error

Input:

clk : 1 bit : clock signal

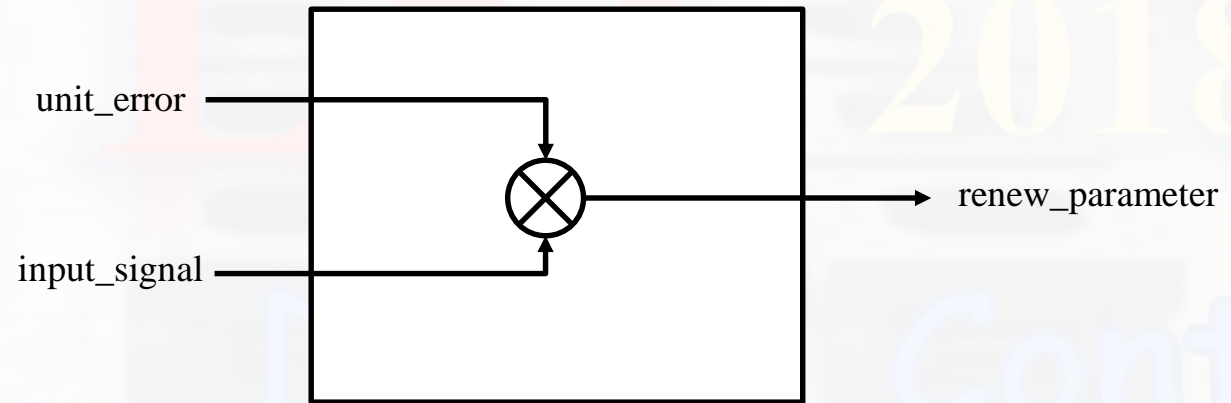
res : 1 bit : reset signal (high active)

unit_error : 16 bits : hidden/output layer's unit error

input_signal : 16 bits : input signal for those units

Output:

renew_parameter : 16 bits : new weight value



Delta bias adder (db_adder_b.v)

Module Name: db_adder

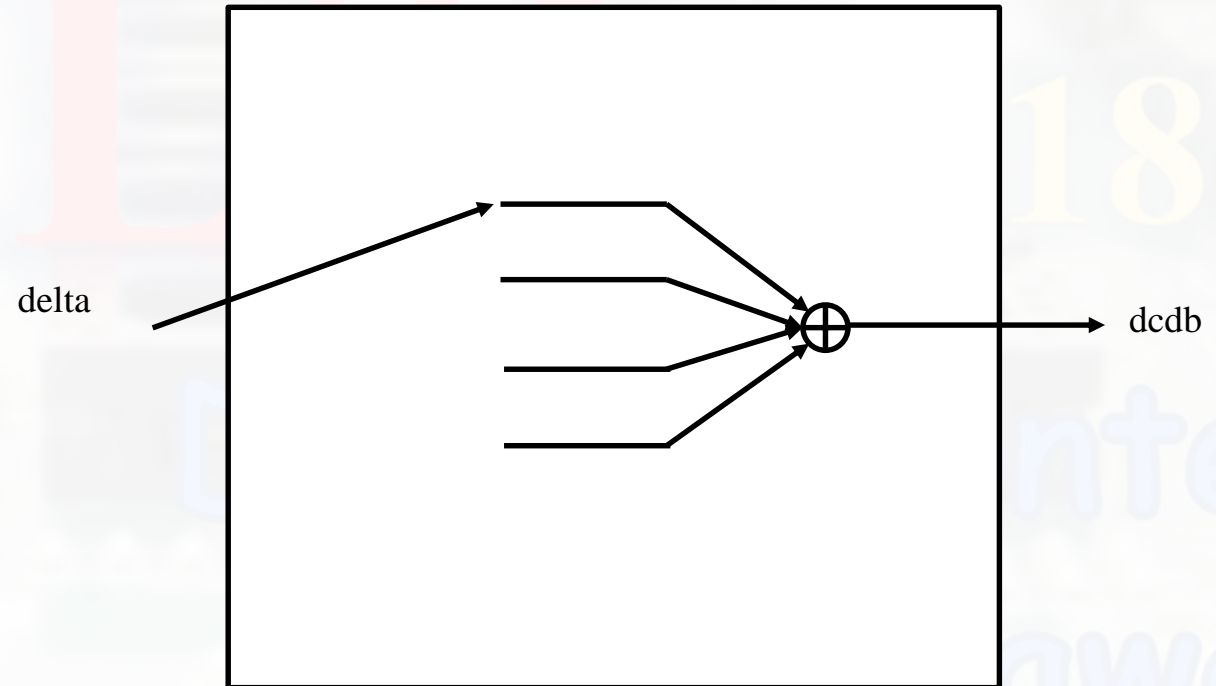
Description:
calculation of capital delta bias

Input:

clk : 1 bit : clock signal
res : 1 bit : reset signal (high active)
delta : 16 bits : delta value of bias

Output:

dcdb : 16 bits : capital delta bias



Delta weight adder (db_adder.v)

Module Name: dw_adder

Description:
calculation of capital delta weight

Input:

clk : 1 bit : clock signal
res : 1 bit : reset signal (high active)
dw : 16 bits : delta value of weight

Output:

dcdw3 : 16 bits : capital delta weight

