Neural Network and HW design

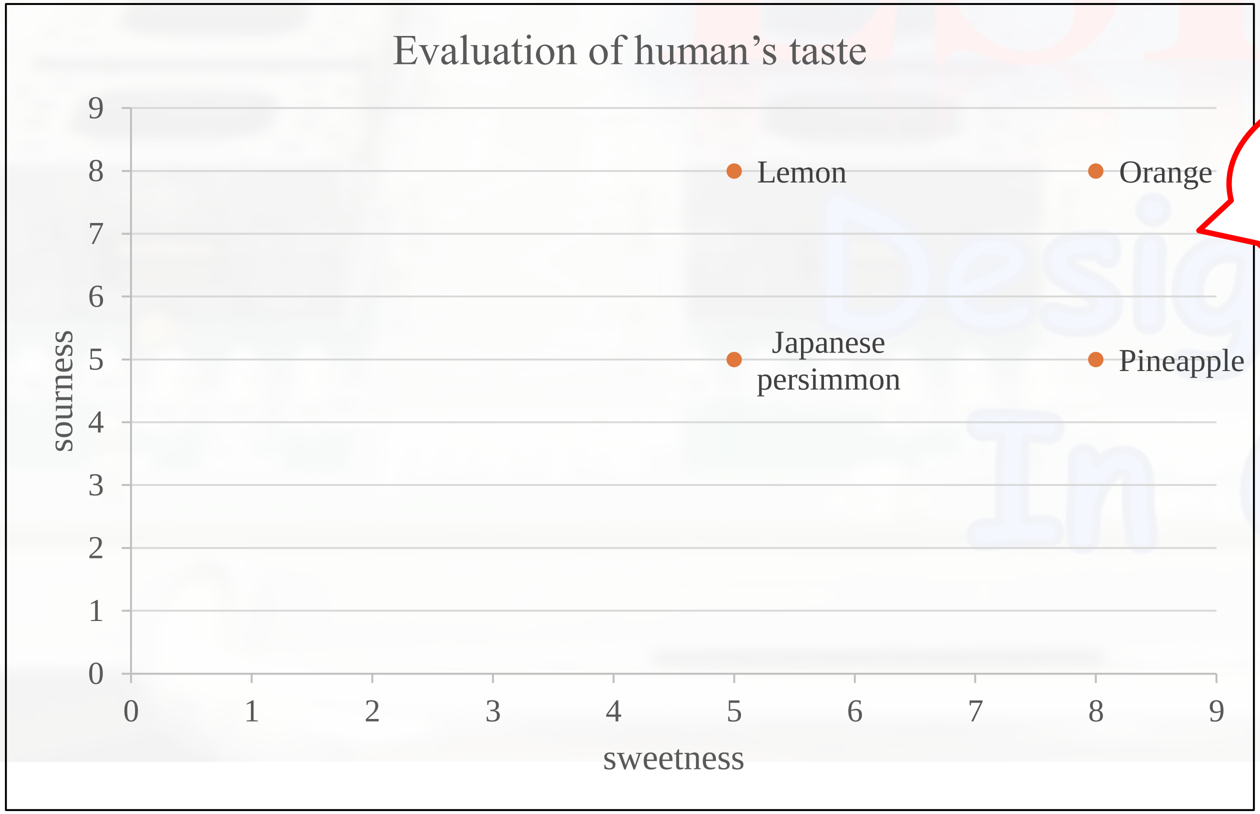
LSI Design Contest

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 is delicious and which fruits does not delicious
- So let’s consider delicious fruits as [1,0] and not delicious fruits as [0,1]



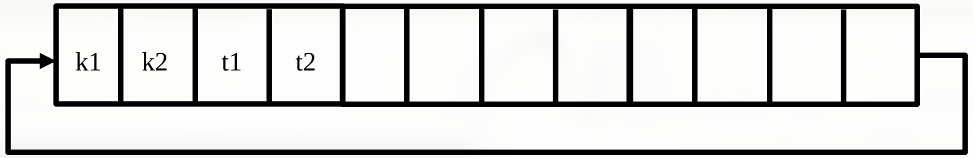
How to measure the man’s taste using Neural Network?

HW specification

forward process

- $z_i^2 = w_{1\,i}^2 k_1 + w_{2\,i}^2 k_2 + b_i^2$ (z2)
- $a_i^2 = a(z_i^2) = \frac{1}{1+e^{-z_i^2}}$ $(i=1,2,3)$ (a2)
- $z_j^3 = w_{1\,j}^3 a_j^2 + w_{2\,j}^2 a_j^2 + w_{3\,j}^2 a_j^2 + b_j^3$ $(z3)$
- $a_j^3 = a(z_j^3) = \frac{1}{1+e^{-z_j^3}}$ $(j = 1,2)$ (a3)

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



backward process

- $a'(z_i^n) = \frac{e^{-z_i^n}}{\left(e^{-z_i^n}+1\right)^2} = ((1 - a_i^n)a_i^n)$ $(n=2,3)$ (dadz)
- $\delta_1^3 = (a_1^3 - t_1)a_1^{3\prime}(z_1^3)$ (delta3)
- $\delta_1^2 = (\delta_1^3 w_{1\,1}^3 + \delta_2^3 w_{2\,1}^3 + \cdots)a_1^{2\prime}\,(z_1^2)$ (delta2)
- $\frac{\partial C}{\partial w_{i\,j}^n}[\mathrm{m}] = \delta_j^n a_i^{n-1}[m]$, however, $a_i^1 = K_i$ (dw2 , dw3)
- $\frac{\partial C}{\partial b_j^n}[m] = \delta_j^n[m]$ (db2 , db3)
- $\Delta \mathrm{w}_{\mathrm{ij}}^{\mathrm{n}} = -\eta \frac{\partial C}{\partial w_{i\,j}^n} = -\eta(\frac{\partial C}{\partial w_{i\,j}^n}[1] + \ldots + \frac{\partial C}{\partial w_{i\,j}^n}[\mathrm{m}])$

(dw_adder_w2,dw_adder_w3)

- $\frac{\partial C}{\partial b_j^n} = \frac{\partial C}{\partial b_j^n}[1] + \cdots + \frac{\partial C}{\partial b_j^n}[m]$ (n=2,3 m=1,2,3,4)

(db_adder_b2,db_adder_b3)

NN_CORE

Module Name: NN_CORE

Description:

Calculation of capital_delta_w(and b) at 10000times and renewal of w and b at once

Input:

clk: 1 bit : clock signal

res: 1 bit : reset signal

din : 1 bit : read enable signal (when din =1, can read the data from memory, when din=0, cannot read the data from memory)

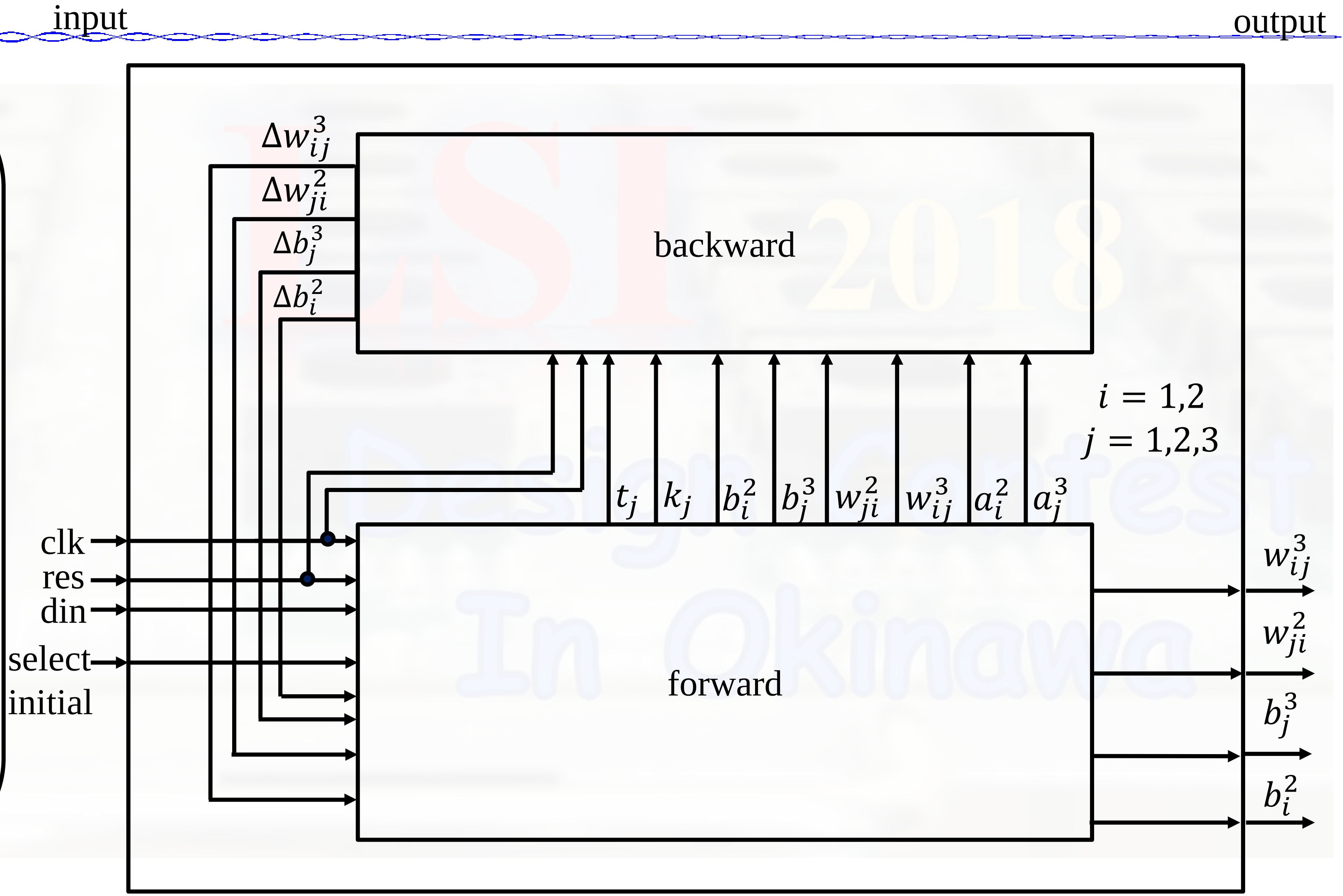
select_initial : 1bit : use the initial value of bias and weight. select_initial=1 only at the beginning of simulation.

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

w_{ij}^3 : 32bits,signed : renewal w3_ij weight

w_{ji}^2: 32bits,signed : renewal w2_ji weight

(32bits,signed has 00000000 . 000000000000000000000000)



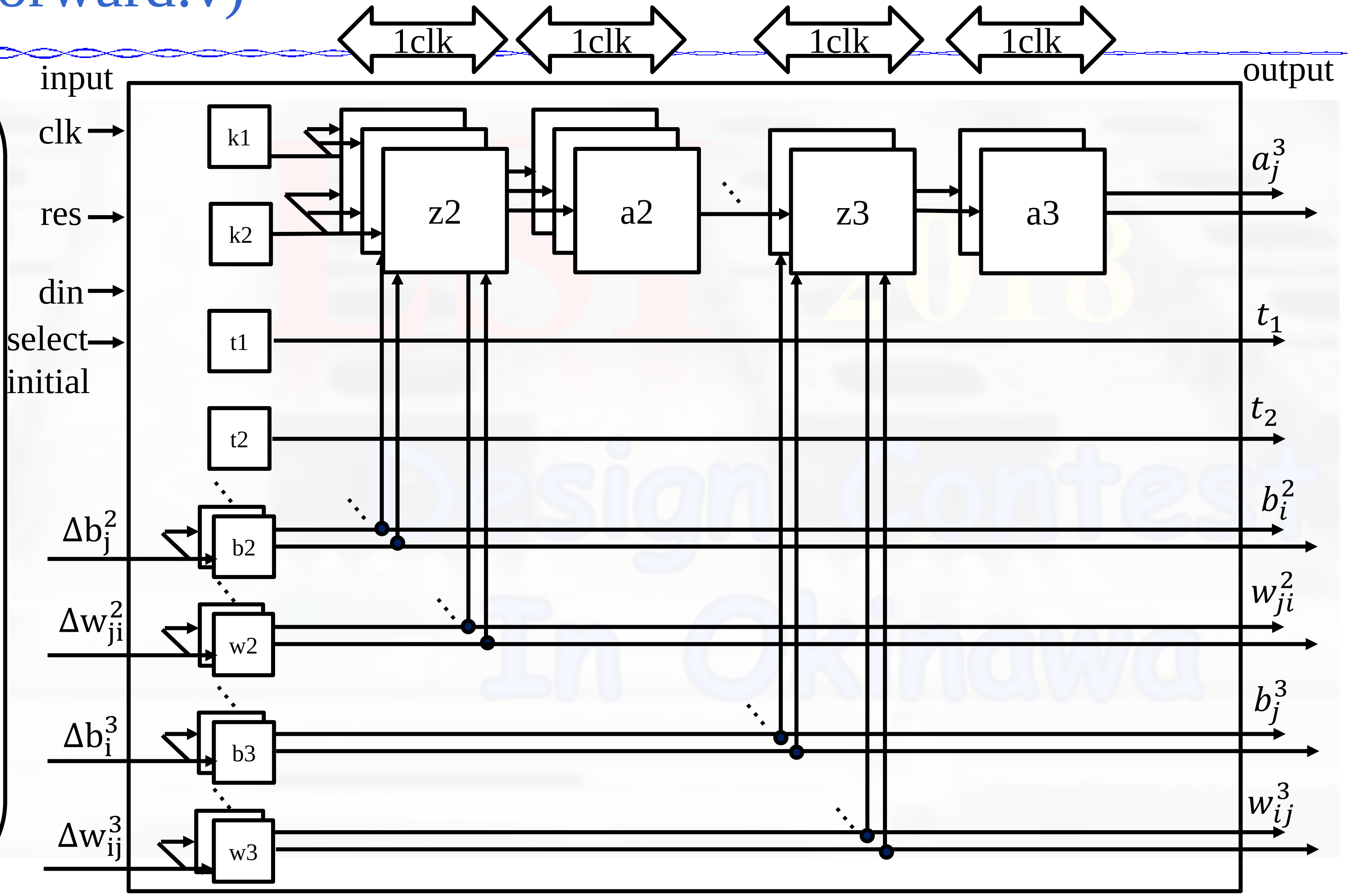
forward (forward.v)

Module Name: forward
Description: Calculation of a2 and a3

Input:(i=1,2,3 , j=1,2)
clk: 1 bit : clock signal
res: 1 bit : reset signal
din : 1 bit : read enable signal
(when din =1, can read the data from mem.)
select_initial : 1bit :
(when din =1, use the initial value of bias and weight.)
Δw_{ij}^{3}: 32bits : the amount of w_{ij}^{3} change
Δw_{ji}^{2}: 32bits : the amount of w_{ji}^{2} change
Δb_{i}^{3} : 32bits : the amount of b_{i}^{3} change
Δb_{j}^{2} : 32bits : the amount of b_{j}^{2} change

Output:(i=1,2,3 , j=1,2)
a_j^3 : 32bits : output_layer_output
k_i : 32bits : supervisor data
t_i : 32bits : supervisor value
w_{ij}^3 : 32bits : output_layer weight
w_{ji}^2 : 32bits : hidden_layer weight
b_j^3: 32bits : output_layer bias
b_i^2 : 32bits : hidden_layer bias
(32bits,signed has
00000000 . 000000000000000000000000)

※配線の一部省略



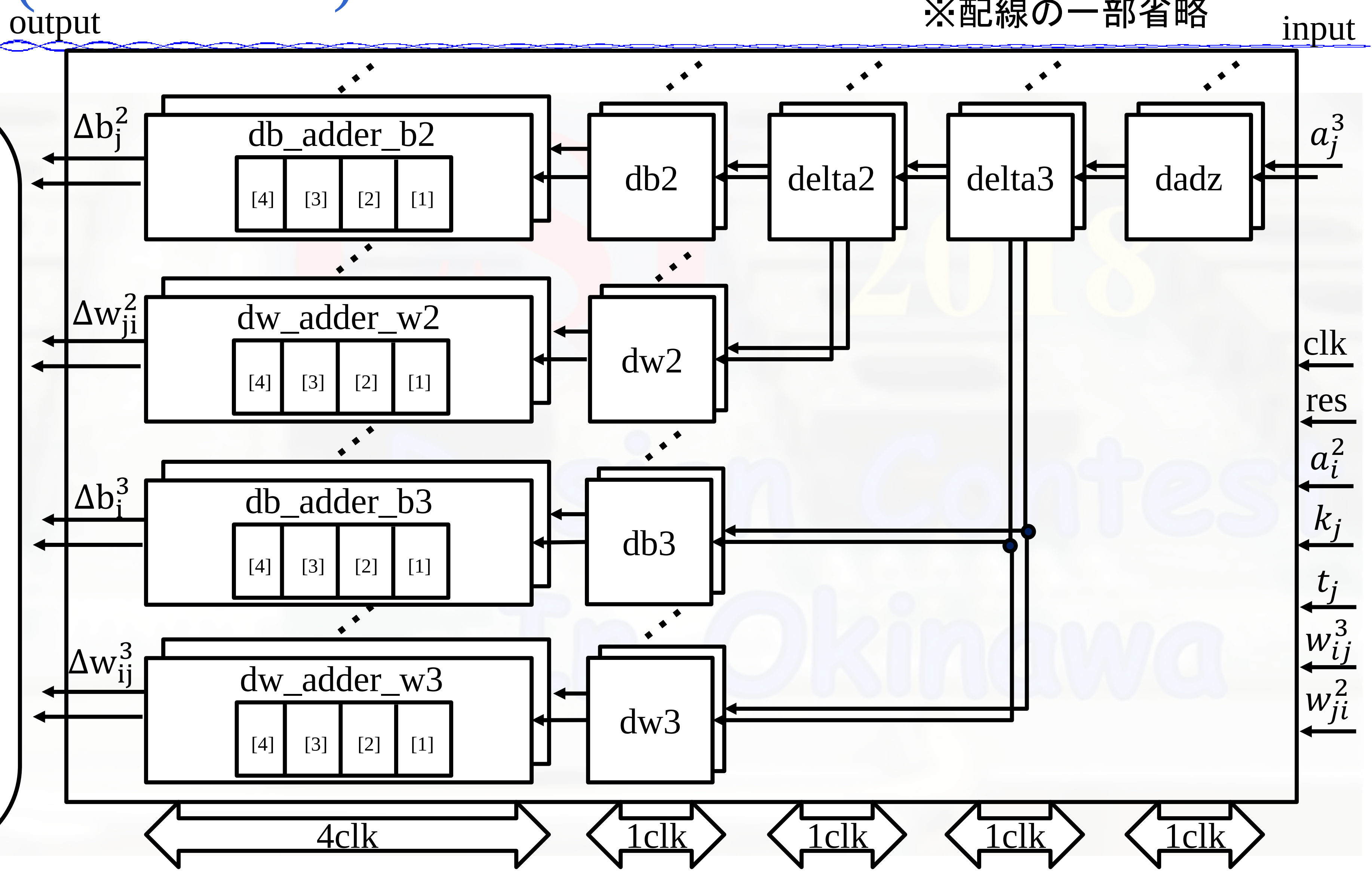
backward (backward.v)

Module Name: backward
Description:
Calculation of capital_delta_w(and b)

Input:(i=1,2,3 , j=1,2)
clk: 1 bit : clock signal
res: 1 bit : reset signal
a_j^3 : 32bits : output_layer output
a_i^2 : 32bits : hidden_layer output
k_j : 32bits : supervisor data
t_j : 32bits : supervisor value
w_{ij}^3 : 32bits : output_layer weight
w_{ji}^2 : 32bits : hidden_layer weight

Output:(i=1,2,3 , j=1,2)
$\Delta \mathrm{w}_{\mathrm{ij}}^3$: 32bits : the amount of w_{ij}^3 change
$\Delta \mathrm{w}_{\mathrm{ji}}^2$: 32bits : the amount of w_{ji}^2 change
$\Delta \mathrm{b}_{\mathrm{i}}^3$: 32bits : the amount of b_i^3 change
$\Delta \mathrm{b}_{\mathrm{j}}^2$: 32bits : the amount of b_j^2 change

※配線の一部省略


output
input
$\Delta \mathrm{b}_{\mathrm{j}}^2$
db_adder_b2
[4] [3] [2] [1]
db2
delta2
delta3
dadz
a_j^3
$\Delta \mathrm{w}_{\mathrm{ji}}^2$
dw_adder_w2
[4] [3] [2] [1]
dw2
clk
res
a_i^2
$\Delta \mathrm{b}_{\mathrm{i}}^3$
db_adder_b3
[4] [3] [2] [1]
db3
k_j
t_j
$\Delta \mathrm{w}_{\mathrm{ij}}^3$
dw_adder_w3
[4] [3] [2] [1]
dw3
w_{ij}^3
w_{ji}^2
4clk
1clk
1clk
1clk
1clk