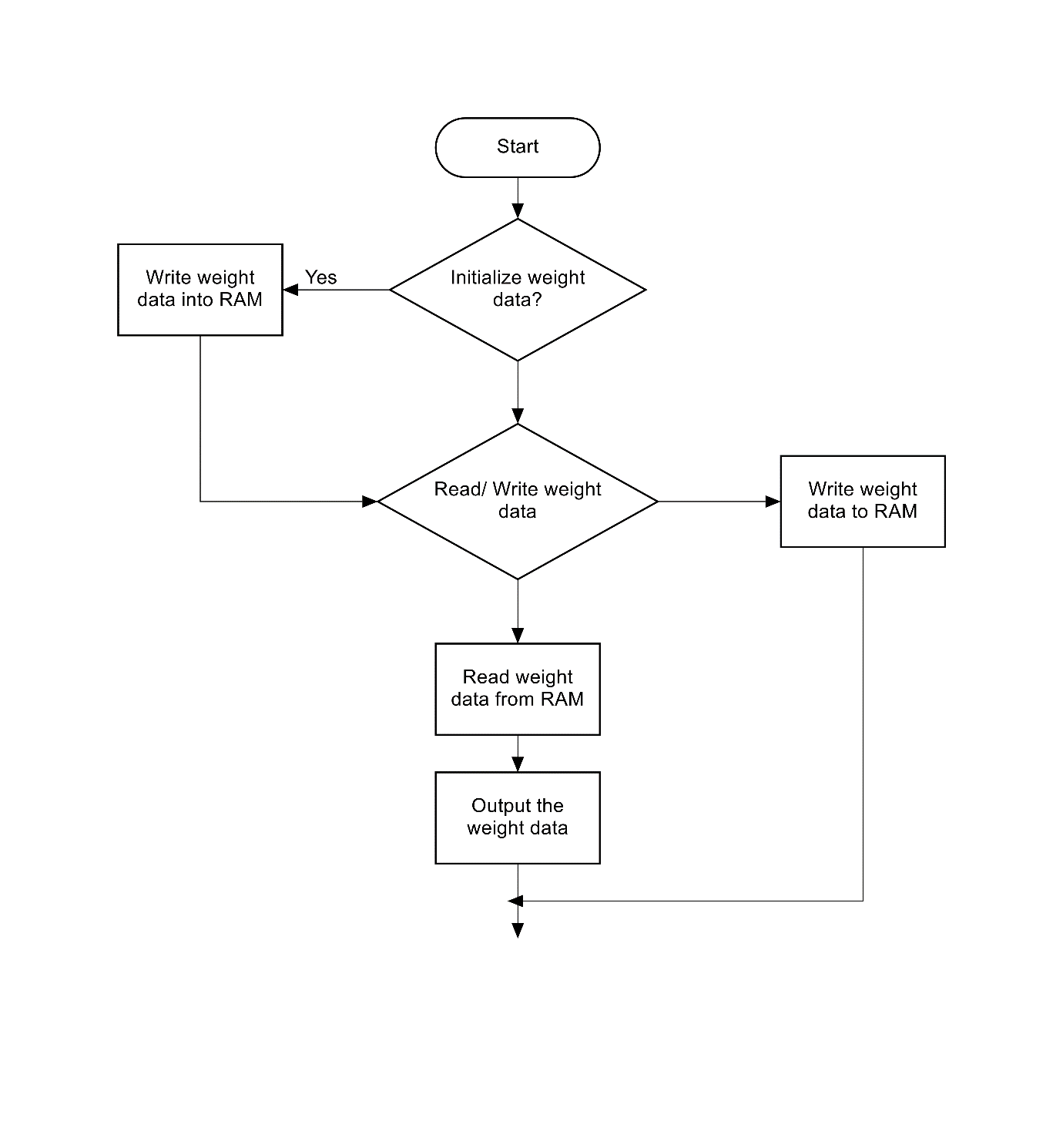
Weight Initialization Block (Block 1) is a module used to initialize the value of weight that will be used by the Hidden Layer Block. For Artificial Neural Network, every inputs must have their own weight respect to every neuron in the hidden layer. In this system, one hidden layer with five neurons is used, therefore 50 weights are needed. Besides that, there will have three neurons in output layer, thus 15 more weights are needed. In total, there are 65 weights for the whole system.

There are two modules in this block, the first module is LFSR. This module is a linear-feedback shift register that generate a 10 bits random number by combining the exclusive-OR configuration to form a feedback mechanism. The second module is WeightInitialize which initialize all 65 weights that needed with random number that generated by LFSR. Besides that, this module also act as RAM to store the weights’ value so that other modules can get the weights’ value in the future.



module WeightRAM(

input Clock,In,

input [9:0] D[0:9],

input [4:0] Address,

input WE,

output reg [9:0] Q[0:9]);

reg [9:0] REGISTER [29:0];

always @(posedge Clock)begin

if(In) begin

REGISTER[0] <= 10'b1010101010;

REGISTER[1] <= 10'b0101010101;

REGISTER[2] <= 10'b1010101010;

REGISTER[3] <= 10'b0101010101;

REGISTER[4] <= 10'b1010101010;

REGISTER[5] <= 10'b0101010101;

REGISTER[6] <= 10'b1010101010;

REGISTER[7] <= 10'b0101010101;

REGISTER[8] <= 10'b1010101010;

REGISTER[9] <= 10'b0101010101;

REGISTER[10] <= 10'b0101010101;

REGISTER[11] <= 10'b0101010101;

REGISTER[12] <= 10'b0101010101;

REGISTER[13] <= 10'b0101010101;

REGISTER[14] <= 10'b0101010101;

REGISTER[15] <= 10'b0101010101;

REGISTER[16] <= 10'b0101010101;

REGISTER[17] <= 10'b0101010101;

REGISTER[18] <= 10'b0101010101;

REGISTER[19] <= 10'b0101010101;

REGISTER[20] <= 10'b1111100000;

REGISTER[21] <= 10'b1111100000;

REGISTER[22] <= 10'b1111100000;

REGISTER[23] <= 10'b1111100000;

REGISTER[24] <= 10'b1111100000;

REGISTER[25] <= 10'b1111100000;

REGISTER[26] <= 10'b1111100000;

REGISTER[27] <= 10'b1111100000;

REGISTER[28] <= 10'b1111100000;

REGISTER[29] <= 10'b1111100000;

end

else begin

if(!WE) begin

Q[0] <= REGISTER[Address];

Q[1] <= REGISTER[Address+1];

Q[2] <= REGISTER[Address+2];

Q[3] <= REGISTER[Address+3];

Q[4] <= REGISTER[Address+4];

Q[5] <= REGISTER[Address+5];

Q[6] <= REGISTER[Address+6];

Q[7] <= REGISTER[Address+7];

Q[8] <= REGISTER[Address+8];

Q[9] <= REGISTER[Address+9];

end

else begin

REGISTER[Address] <= D[0];

REGISTER[Address+1] <= D[1];

REGISTER[Address+2] <= D[2];

REGISTER[Address+3] <= D[3];

REGISTER[Address+4] <= D[4];

REGISTER[Address+5] <= D[5];

REGISTER[Address+6] <= D[6];

REGISTER[Address+7] <= D[7];

REGISTER[Address+8] <= D[8];

REGISTER[Address+9] <= D[9];

end

end

end

endmodule

module WeighInitiallize(

input Clock, WE,In,

input [4:0] address,

output reg [9:0]weight [0:9]

);

reg [9:0] X[0:9];

wire [9:0] Y[0:9];

integer i;

always@(posedge Clock)begin

weight = Y;

end

WeightRAM part2(Clock,In,X,address,WE,Y);

endmodule

module WeightInitiallize\_tb;

//input

reg Clock, WE, In;

reg [4:0] address;

//output

wire [0:9]weight[0:9];

initial begin

Clock = 0;

forever #50 Clock = ~Clock;

end

initial begin

WE = 0;

In = 1;

address = 5'b00000;

#500 In = 0;

#500 address = 5'b01010;

#500 address = 5'b10100;

end

WeighInitiallize test(Clock,WE,In,address,weight);

initial begin

$display("weight");

$monitor(" %d\n %b \n %b \n %b \n %b \n %b \n %b \n %b \n %b \n %b \n %b ",

address,weight[0],weight[1],weight[2],weight[3],weight[4],weight[5],weight[6],weight[7],weight[8],weight[9]);

end

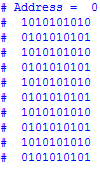
endmodule

**Console result**

In = 1, initialize the data into RAM, thus the weight value is unknown.



In = 0, load the first to tenth data from RAM into the weight.



The value of address changed but the data will be loaded when the clock is posedge, therefore the address changed but the weight data didn’t changed.

