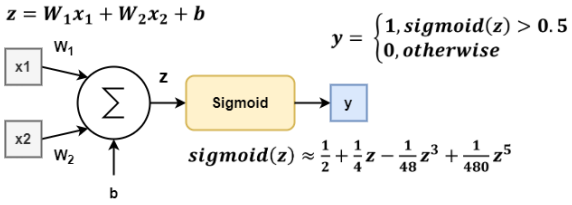
2023 Spring NYCU-EE Machine Learning Intelligent Chip Design – Homework2

IEE 陳冠瑋 310510221 / May. 4, 2023

1. Design idea

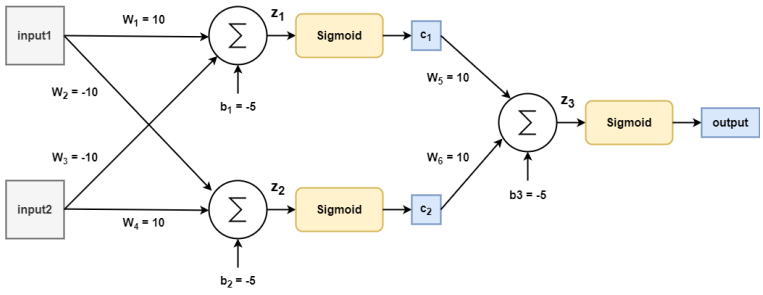
(1) 首先，整個 neuron network 是由 3 顆 neuron 所構成，所以先完成單一顆 neuron 的功能，後面可以直接呼叫這個 object 來使用。



Neuron

```
#include "Neuron.h"

void Neuron::neuron() {

    // vvvvv put your code here vvvvv
    output_temp = input1.read()*w1 + input2.read()*w2 + b;
    y = (1.0/2) + (1.0/4)*output_temp - (1.0/48)*output_temp*output_temp*output_temp + (1.0/480)*output_temp*output_temp*output_temp*output_temp*output_temp;
    y = (y > 0.5) ? 1 : 0;
    output.write(y);
    // ^^^^^ put your code here ^^^^^
}
```

(2) 接著將三顆 neuron 實例化，拉線來完成 spec 所要求的 neural network。



Neural Network

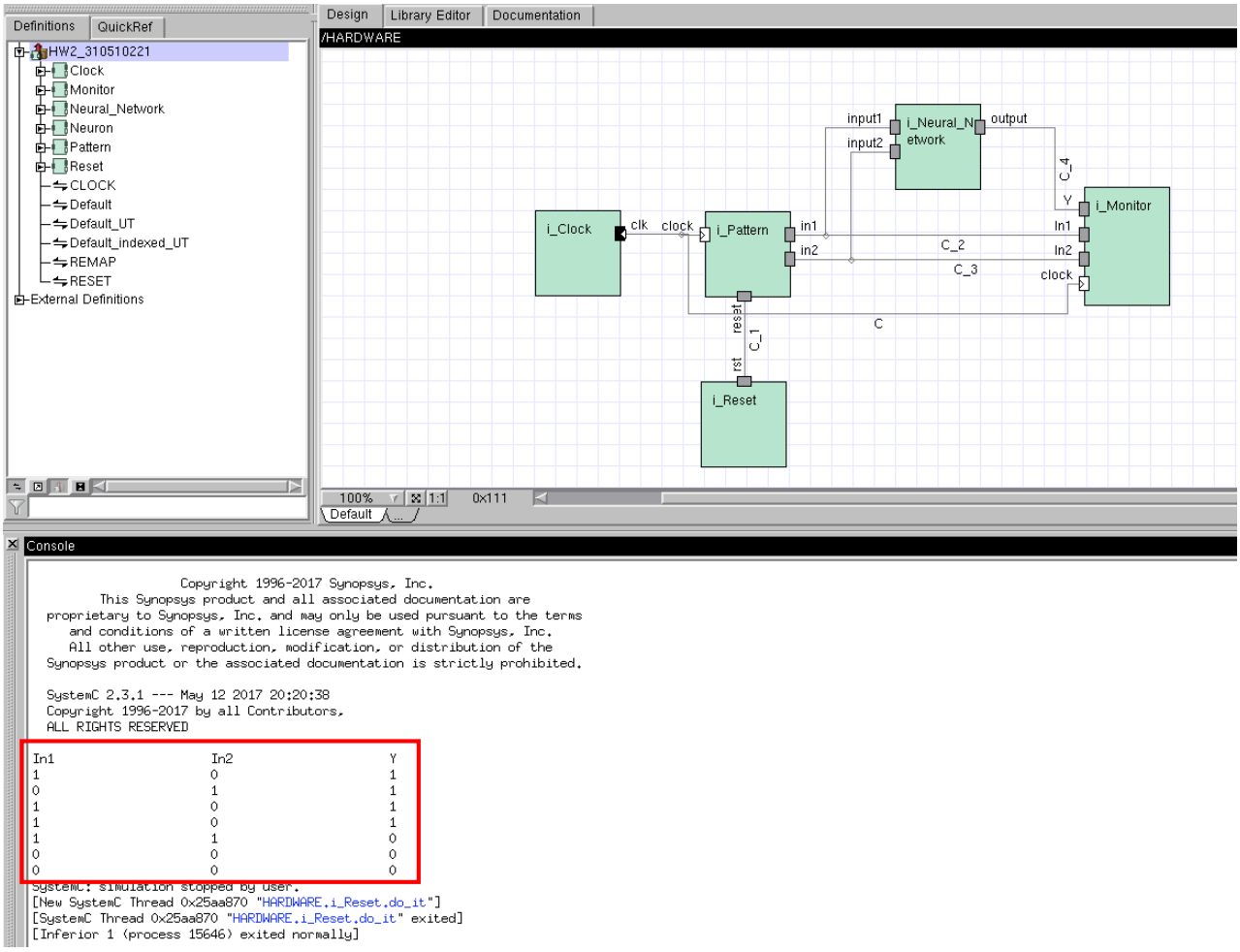
```
#include "systemc.h"
#include "Neuron.h"

SC_MODULE( Neural_Network ) {

    sc_in < float > input1, input2;
    sc_out < float > output;

    sc_signal < float > c1, c2;

    Neuron *N1;
    Neuron *N2;
    Neuron *N3;

    SC_CTOR( Neural_Network ) {

        // vvvvv put your code here vvvvv
        N1 = new Neuron("N1");
        N2 = new Neuron("N2");
        N3 = new Neuron("N3");

        N1->input1(input1);
        N1->input2(input2);
        N1->output(c1);
```

```
        N1->input1(input1);
        N1->input2(input2);
        N1->output(c1);

        N2->input1(input1);
        N2->input2(input2);
        N2->output(c2);

        N3->input1(c1);
        N3->input2(c2);
        N3->output(output);

        sensitive << input1 << input2;
        // ^^^^^ put your code here ^^^^^

        N1->w1 = 10;
        N1->w2 = -10;
        N1->b = -5;
        N2->w1 = -10;
        N2->w2 = 10;
        N2->b = -5;
        N3->w1 = 10;
        N3->w2 = 10;
        N3->b = -5;
    }
};
```

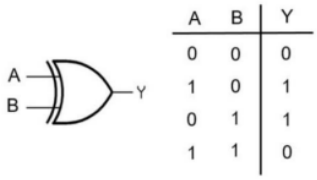
input1	input2	z_1	z_2	c_1	c_2	z_3	output
0	0	-5	-5	0	0	-5	0
0	1	-15	5	0	1	5	1
1	0	5	-15	1	0	5	1
1	1	-5	-5	0	0	-5	0

(3) 透過上表可以發現，只有當 input1、input2 相異的時候，訊號才會通過第一層 neuron，否則會因為與 weight 和 bias 計算出來的結果為負，被 threshold 擋下來變成輸出值為 0。另外，此時把 activation function 拿掉，直接判斷值是否有超過 threshold 也不會影響結果，或是可以選擇用更簡單的 ReLU 作為 activation function。最後，觀察 input 與 output 的結果可以知道這個 neural network 的行為就如同 XOR gate，input 為奇數個 1 的時候 output 才為 1，否則為 0。

2. Block diagram



Run simulation 後觀察 Console 可以確認使用 Platform Architect IP 拉線後的結果。由此可知，此 neural network 的行為配合這組 weight 與 bias 是符合 XOR 的行為。



A	B	Y
0	0	0
1	0	1
0	1	1
1	1	0

3. Simulation result

```
ML310510221@ZEUS /home/ML310510221/HW#2_code% make
g++ -I . -I  /usr/systemc/include -L . -L /usr/systemc/lib-linux64 -o LAB *.cpp -lsystemc -lm -DSC_INCLUDE_FX
ML310510221@ZEUS /home/ML310510221/HW#2_code% ./LAB

        SystemC 2.3.3-Accellera --- Nov 16 2021 18:55:23
        Copyright (c) 1996-2018 by all Contributors,
        ALL RIGHTS RESERVED
In1                     In2                     Y
1                       1                       0
0                       0                       0
1                       1                       0
0                       1                       1
0                       1                       1
1                       0                       1
0                       0                       0

Info: /OSCI/SystemC: Simulation stopped by user.
```

透過 Make 編譯後，執行./LAB 模擬的結果也符合 XOR 的行為。