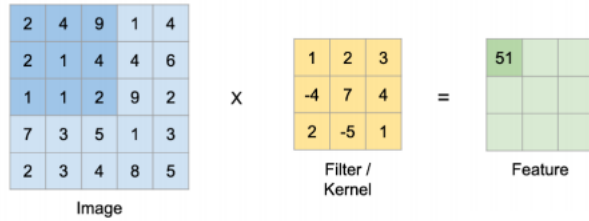


Homework#3

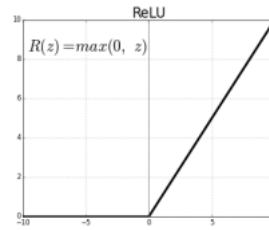
B083040045 李亦晴

>>design ideas



依照題目要求，掃圖並且和 Filter 數據相乘，形成特徵值
這些取出的特徵值有助於分析圖片

而依照題目要求，將小於 0 的值歸為 0，儲存起來



$$f(x) = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases}$$

Fig.2 ReLU function

>>code

>Conv.h

```
using namespace std;

SC_MODULE( Conv ) {
    sc_in < bool > rst;
    sc_in_clk clk;
    sc_out < bool > rom_rd;
    sc_out < sc_uint<7> > rom_addr;
    sc_in < float > data_in;
    sc_out < sc_int<19> > data_out;
    sc_out < bool > data_out_signal;
    float rom_m [111];
    sc_int<19> sum;
    int count,first;
    int idx;
    void run();
    // void rom();
    SC_CTOR( Conv )
    {
        count=0;
        first=0;
        SC_METHOD( run );
        sensitive << rst.pos() << clk.pos();
    }
};
```

當 rst 或 clk 正緣觸發時，觸發 run 函式

>Conv.cpp

```
void Conv::run() {
    // vvvvvv put your code here vvvvvv
    if(rst==1)
    {
        rom_rd.write(0);
        rom_addr.write(0);
        data_out.write(0);
        data_out_signal.write(0);
    }
    else if(count>111)
    {
        if(first<=78)
        {
            if(first%10!=0&&first%10!=9)
            {
                sum=0;
                for(int i=0;i<9;i++)
                {
                    idx=first+(i/3)*10+(i%3);
                    sum=sum+rom_m[idx]*rom_m[101+i];
                }
                sum+=rom_m[110];
                if(sum<0)
                {
                    data_out.write(0);
                    data_out_signal.write(1);
                }
                else
                {
                    data_out.write(sum);
                    data_out_signal.write(1);
                }
            }
            else
            {
                data_out_signal.write(0);
            }
            first++;
        }
        else
        {
            rom_rd.write(1);
            rom_addr.write(count);
            count++;
            if(count>111)
                rom_rd.write(0);
            if(count!=1)
            {
                if(data_in==0)
                    count-=2;
                else
                    rom_m[count-2]=data_in;
            }
        }
    }
}
```

當 reset 被觸發，將輸入及輸出信號都設為 0

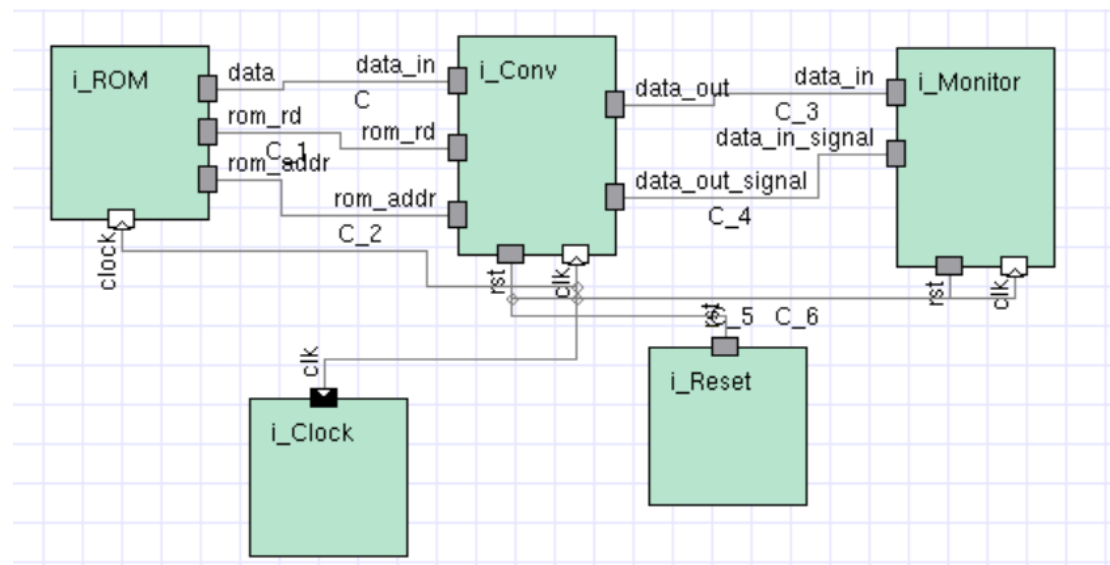
依照陣列以及應計算的結構，決定邊緣狀態。每一塊的計算，由最左上角的元素開始，故而左上角的邊緣選取最多到 78，而每一行的邊緣 (9、10) 也不能選取。

邊緣條件設定好後，需要注意的是暫存時，需要使用 `sc_int<19>` 作為暫存器，才能跟結果相符

最後的計算結果依照題意，小於 0 者歸於 0 輸出。

當計數次數尚未滿 111 (也就是資料尚未讀完) 時，持續將通往 rom 的輸入信號設為 1，並持續傳出 address 而傳回來的數值，前期不知為何會多傳幾個 0 回來 (可能是因為 rst 的原因?) 故在此多設定一個 if，過濾掉多餘的 0。並且將 count 退回去，以免跳掉應輸入的 address

>>block diagram,



>>simulation result of PA

```

Result:
Result:
0      174      0      103      0      92      22      102
22      0      0      61      0      22      0      0
11      106      36      0      71      13      66      55
0      23      0      0      18      0      66      0
44      8      93      94      10      76      0      0
0      14      94      11      0      0      0      197
53      171      0      0      78      52      80      0
108      0      93      0      141      0      67      0
SystemC: simulation stopped by user.
  
```