实验报告

目录

实验环境

硬件环境

CPU：Intel 酷睿 i7 8700
内存：64GB
硬盘：1TB SSD

软件环境

操作系统：Windows 10 1903 专业版
IDE：ISE 14.7
编程语言：Verilog

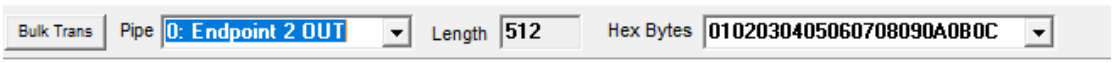
实验结果

实现了一个写死的卷积操作。buff 数组的每一个元素是 16 位，即 2 个字节。

定义矩阵为 2 行 3 列，卷积核是 2x2 的$\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix}$，即每次卷积操作将 4 个元素加起来。

每次读数据的过程中有噪音，buff[0]不是自己 write 的内容，还没来得及查。

输入数据如下



```
01C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Bulk IN Transfer
Bulk IN success.
Buffer Contents
0000  01 02 01 02 03 04 05 06 07 08 09 0A 0B 0C 00 00
0010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0020  0E 12 14 18 00 00 00 00 00 00 00 00 00 00 00 00
0030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

如图所示，第一次卷积为$0102 + 0102 + 0506 + 0708 = 0E12$，是数组的$buff[0] + buff[1] + buff[3] + buff[4]$，第 2 次卷积同第一次，是$buff[1] + buff[2] + buff[4] + buff[5]$

代码修改如下：

增加了一个状态 JUANJI

```
    READ_FROM_USB: begin
       if((counter == `MAXPKG - 1) || (i_usb_flaga == 1'b0))begin
           state_nxt = JUANJI;
       end
       else begin
           state_nxt = READ_FROM_USB;
       end
    end
    SELECT_WRITE_FIFO: begin
        if(i_usb_flagd == 1'b1) begin
            state_nxt = WRITE_TO_USB;
        end
        else begin
            state_nxt = IDLE;
        end
    end
    WRITE_TO_USB: begin
       if(counter == number) begin
           state_nxt = IDLE;
       end
       else begin
           state_nxt = WRITE_TO_USB;
       end
    end
    JUANJI: begin
        state_nxt = SELECT_WRITE_FIFO; // 这里可能有问题，是不是应该计算完成再转成IDLE状态，要考虑一下
    end
    default: begin
        state_nxt = IDLE;
    end
endcase
d
```

当 FPGA 读入数据后，不马上进入 IDLE，而是跳去进行卷积操作，当卷积操作进行完之后，转而向 buff 中写，方便 PC 端进行读操作。

卷积操作如下：

```
        usb_data <= buff[counter];
    end
    JUANJI: begin
    // 写死，一定是2x3矩阵，然后卷积核是2x2的，全是1
    buff[8'd16] <= buff[4'd0] + buff[4'd1] + buff[4'd3] + buff[4'd4];
    buff[8'd17] <= buff[4'd1] + buff[4'd2] + buff[4'd4] + buff[4'd5];
    end
    default: begin
        usb_slrd <= 1'b1;
        usb_slwr <= 1'b1;
```

有点 low，都是写死的= =

小结

遇到的问题

1. 最开始板子可以 write，但是无法 read，刷了固件也不行
2. 然后跟一个兄台讨论了一下，又刷了一次固件，突然可以 wirte，也可以 read 了
3. 然后又试了一下，发现有时候可以 read，有时候不能= =随机性 read 成功，感觉十分尴尬
4. 感觉指导资料比较少= =有些盲人摸象的感觉。。。