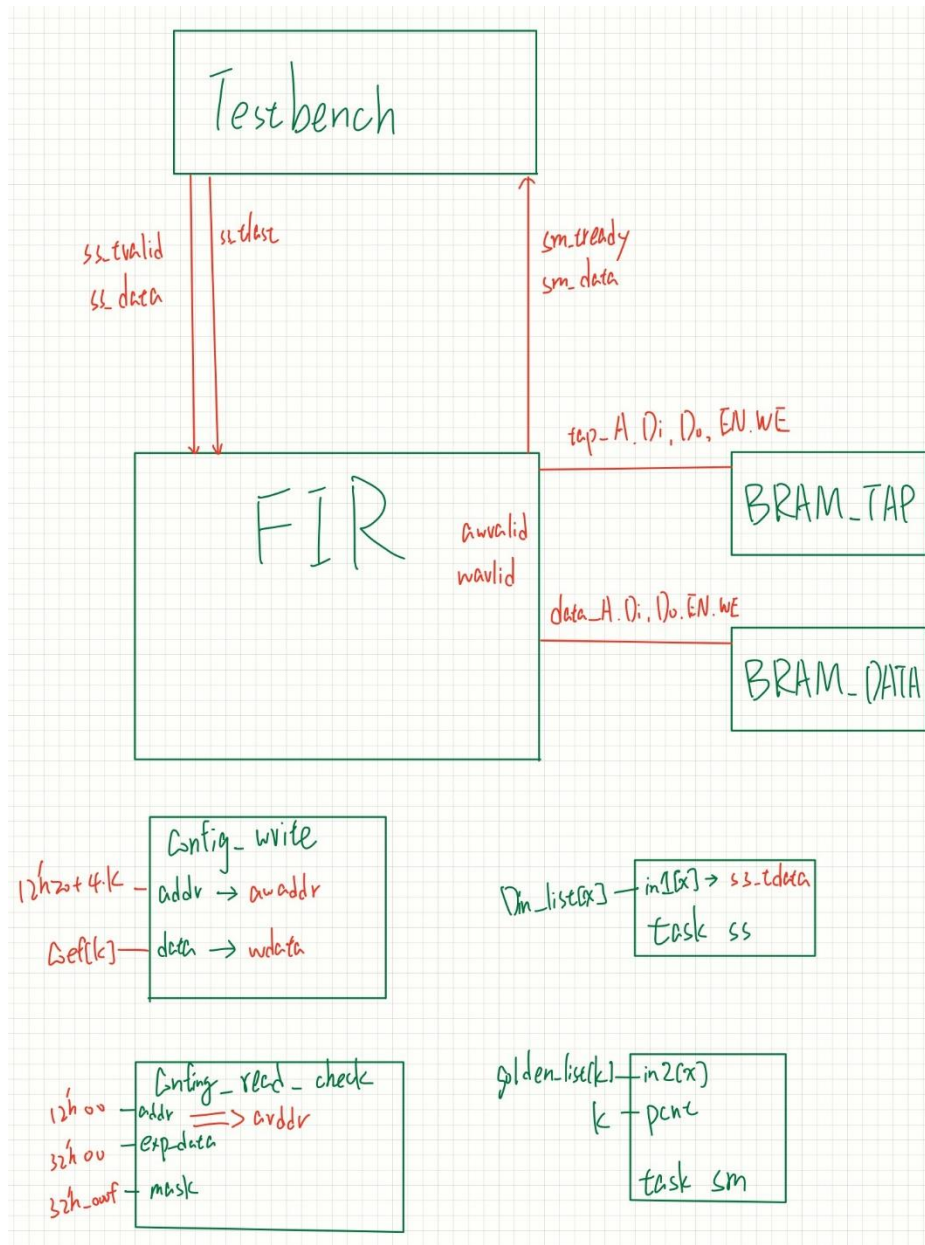


Lab3

電機碩二_111061629_洪啓恩

1. Block Diagram



2. Describe operation

FIR 是一種數字濾波器，它的運作方式是通過將一串數據（即輸入信號）與一組系數（也稱為脈衝響應）進行點積來生成輸出信號。在這個過程中，每一筆新的數據輸入都會與相應的系數相乘，然後將這些相乘的結果進行累加，直到所有的系數都被使用過，這時就會輸出該次計算的結果。

在硬件設計中，使用 BRAM (Block RAM) 來存儲這些數據和系數是一種常見的做法。然而，對於 FIR 濾波器，每次計算都需要將數據和系數進行移位操作，這樣才能確保每次計算都使用到正確的數據和系數。然而，在 BRAM 中進行這種移位操作相對比較複雜且耗時，因為讀取和寫入操作本身就需要較多的時間。

為了解決這個問題，我利用 pointer。通過使用 pointer，我們可以跳過將數據和系數進行實際移位的過程，而是直接通過改變指針的值來指向 BRAM 中的正確位置。這樣，我們就可以更高效地訪問需要的數據和系數，從而加快計算速度。

3. Resource usage

下圖可以看到 FF 和 LUT

```
28 1. Slice Logic
29 -----
30
31 +-----+-----+-----+-----+-----+-----+
32 | Site Type | Used | Fixed | Prohibited | Available | Util% |
33 +-----+-----+-----+-----+-----+-----+
34 | Slice LUTs* | 2181 | 0 | 0 | 53200 | 4.11 |
35 | LUT as Logic | 2181 | 0 | 0 | 53200 | 4.11 |
36 | LUT as Memory | 0 | 0 | 0 | 17400 | 0.00 |
37 | Slice Registers | 1096 | 0 | 0 | 106400 | 1.03 |
38 | Register as Flip Flop | 1093 | 0 | 0 | 106400 | 1.03 |
39 | Register as Latch | 3 | 0 | 0 | 106400 | <0.01 |
40 | F7 Muxes | 429 | 0 | 0 | 26600 | 1.61 |
41 | F8 Muxes | 204 | 0 | 0 | 13300 | 1.53 |
42 +-----+-----+-----+-----+-----+-----+
```

4. Timing Report

Slack 為 5.261 ns

Design Timing Summary			
Setup		Hold	Pulse Width
Worst Negative Slack (WNS): 5.261 ns		Worst Hold Slack (WHS): 0.143 ns	Worst Pulse Width Slack (WPWS): 4.500 ns
Total Negative Slack (TNS): 0.000 ns		Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0		Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 169		Total Number of Endpoints: 169	Total Number of Endpoints: 1124
All user specified timing constraints are met.			

Max delay path:

```

533 Max Delay Paths
534 -----
535 Slack (MET) : 5.261ns (required time - arrival time)
536 Source: genblk1.coef_lite_reg[93][3]/C
537 (rising edge-triggered cell FDCE clocked by axis_clk {rise@0.000ns fall@5.000ns period=10.000ns})
538 Destination: genblk1.rdata_r_reg[19]/D
539 (rising edge-triggered cell FDCE clocked by axis_clk {rise@0.000ns fall@5.000ns period=10.000ns})
540 Path Group: axis_clk
541 Path Type: Setup (Max at Slow Process Corner)
542 Requirement: 10.000ns (axis_clk rise@10.000ns - axis_clk rise@0.000ns)
543 Data Path Delay: 4.603ns (logic 1.145ns (24.875%) route 3.458ns (75.125%))
544 Logic Levels: 4 (LUT6=4)
545 Clock Path Skew: -0.145ns (DCD - SCD + CPR)
546 Destination Clock Delay (DCD): 2.128ns = ( 12.128 - 10.000 )
547 Source Clock Delay (SCD): 2.456ns
548 Clock Pessimism Removal (CPR): 0.184ns
549 Clock Uncertainty: 0.035ns ((TSJ^2 + TIJ^2)^1/2 + DJ) / 2 + PE
550 Total System Jitter (TSJ): 0.071ns
551 Total Input Jitter (TIJ): 0.000ns
552 Discrete Jitter (DJ): 0.000ns
553 Phase Error (PE): 0.000ns

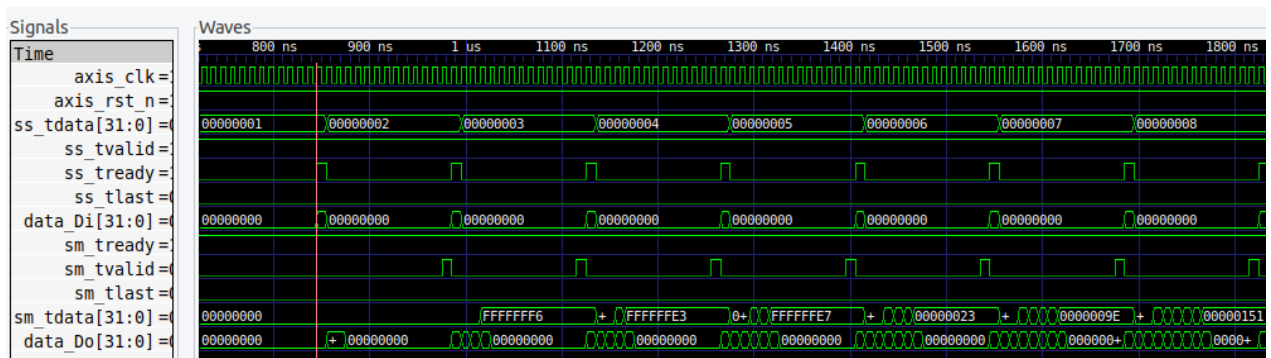
```

5. Simulation Waveform

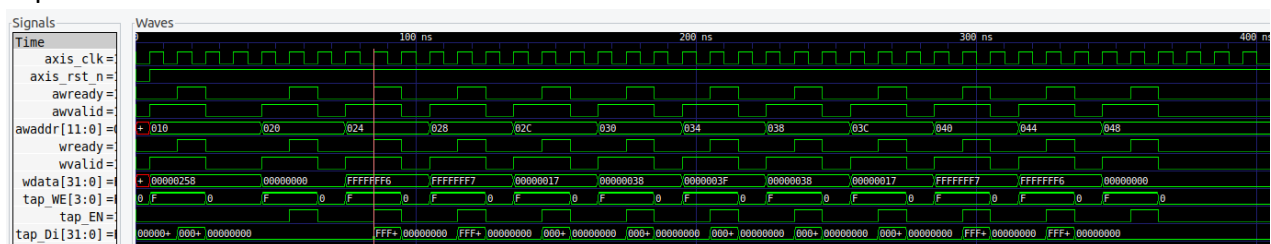
stream-in, and stream-out

Data BRAM 寫入操作中，輸入數據通過 AXI stream 進入，並直接寫入 Data BRAM 中。當 `ss_tready`, `ss_tvalid` 同時為 1，且 `data_we` 為 4'b1111，在下一個 clock cycle，數據將被寫入 BRAM。

完成 FIR 運算後，答案由 AXI stream 輸出，port 為 `sm_tdata`。當 `sm_data` 上出現此次運算的答案時，`sm_tvalid` 會被設置為 1。這時 `testbench` 會讀取並與 `golden data` 比對是否正確。



Tap BRAM write:



Tap BRAM read back:

