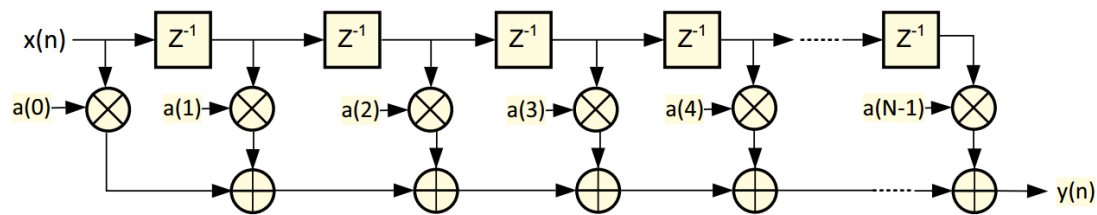
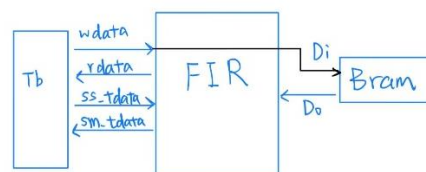


Block Diagram

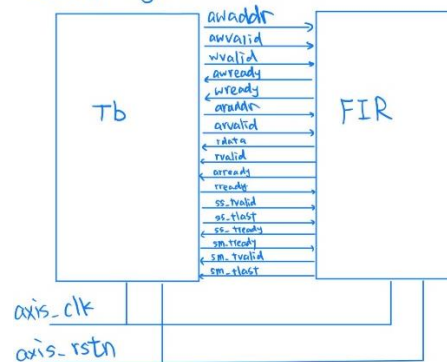


- Datapath – dataflow 、Control signals

Data Flow:



Control signal



- Describe operation

使用 AXI Lite 介面來接收 Tap 參數並將其寫入 BRAM。在地址方面需要注意，由於 Tap 參數位於 0x20-FF，因此將這些地址減去 0x20 後，即可正確映射到要寫入的 BRAM 地址。資料輸入理論上也是使用相同的形式，但資料輸入是透過 AXI Stream 進行傳輸。為了方便，我在 FIR 濾波器中使用了 shift register 來存儲 Tap 參數和資料輸入。當需要使用 RAM 中的值時，只需提供地址和啟用信號，在兩個時脈周期後，tap_Do 和 data_Do 就會輸出所需的值。在接收每個資料時，我會先暫停接收資料，經過 11 個時脈周期進行 $x * h$ 的運算來計算 sm_tdata，輸出 sm_tdata 後，再繼續接收下一個資料並進行一次計算，以此類推。

- How to receive data-in and tap parameters and place into SRAM

是用 axi-lite 來收取 tap parameters，同時將其寫至 bram，須注意在 address 的部分，因為 tap parameters 是 0x20-FF，所以我們將這些 address 減 0x20 後，即可正確對應到我們想填寫的 bram address。而 data-in 理論上也是用同樣的形式即可，但 data-in 是藉由 axi-stream 進行傳輸。

- How to access shiftram and tapRAM to do computation

為了方便，我在 fir 裡有 shiftregister 來存 tap_parameter 和 data-in，想要用 ram 裡面的值時，只須給 address 和 en，等兩個 clk 後，tap_Do、data_Do，即為所需。我在收到每一筆 data 時，會先暫停收 data，經過 11 個 clk 來做 $x \cdot h$ 的運算來算 sm_tdata，輸出 sm_tdata 後，進行收取下一筆 data，再算一次，以此類推。

- How ap_done is generated.

Ap_done 我是用 counter 去判斷，當我做 610 次的運算時，即會發出 ap_done 的訊號來說明 fir 已經完成運算，ap 是包含在 rdata 的後 3 個 bit。

FF, LUT, BRAM:

1. Slice Logic

| Site Type | Used | Fixed | Prohibited | Available | Util% |
|-----------------------|------|-------|------------|-----------|-------|
| Slice LUTs* | 353 | 0 | 0 | 53200 | 0.66 |
| LUT as Logic | 353 | 0 | 0 | 53200 | 0.66 |
| LUT as Memory | 0 | 0 | 0 | 17400 | 0.00 |
| Slice Registers | 765 | 0 | 0 | 106400 | 0.72 |
| Register as Flip Flop | 761 | 0 | 0 | 106400 | 0.72 |
| Register as Latch | 4 | 0 | 0 | 106400 | <0.01 |
| F7 Muxes | 0 | 0 | 0 | 26600 | 0.00 |
| F8 Muxes | 0 | 0 | 0 | 13300 | 0.00 |

2. Memory

| Site Type | Used | Fixed | Prohibited | Available | Util% |
|----------------|------|-------|------------|-----------|-------|
| Block RAM Tile | 0 | 0 | 0 | 140 | 0.00 |
| RAMB36/FIFO* | 0 | 0 | 0 | 140 | 0.00 |
| RAMB18 | 0 | 0 | 0 | 280 | 0.00 |

3. DSP

| Site Type | Used | Fixed | Prohibited | Available | Util% |
|--------------|------|-------|------------|-----------|-------|
| DSPs | 3 | 0 | 0 | 220 | 1.36 |
| DSP48E1 only | 3 | | | | |

Timing:

Clk cycle:12.9ns

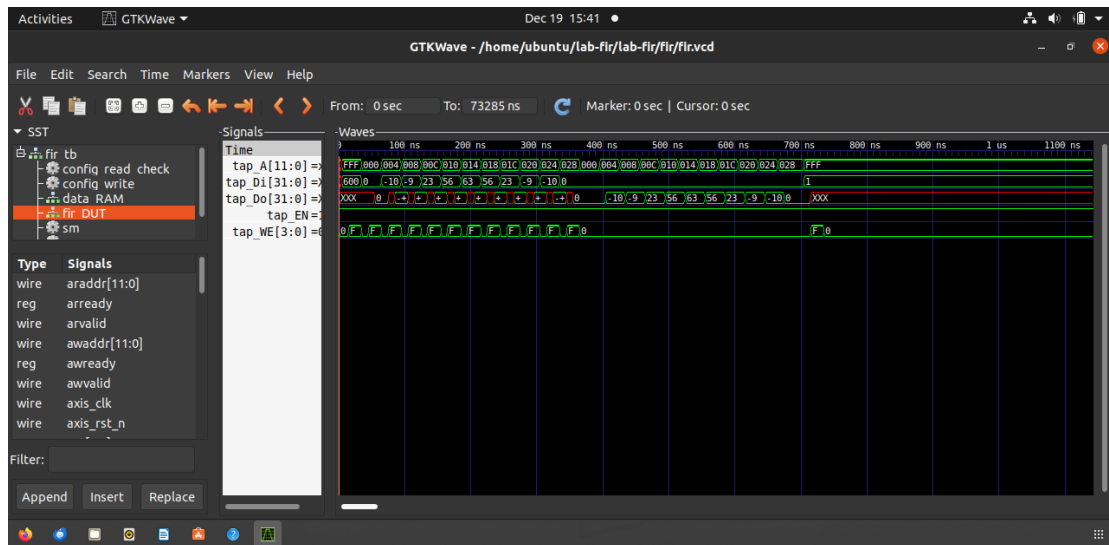
Critical path:

Max Delay Paths

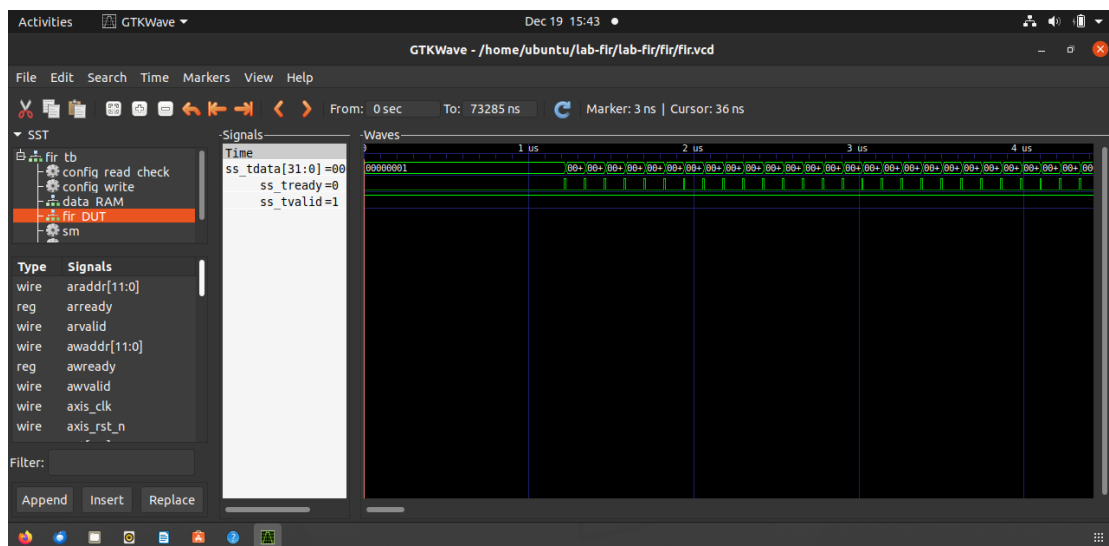
```
-----
Slack (MET) : 0.000ns (required time - arrival time)
Source:      h_reg[3][16]/C
              (rising edge-triggered cell FDCE clocked by axis_clk {rise@0.000ns fall@6.450ns period=12.899ns})
Destination: y_reg[29]/D
              (rising edge-triggered cell FDCE clocked by axis_clk {rise@0.000ns fall@6.450ns period=12.899ns})
Path Group:  axis_clk
Path Type:   Setup (Max at Slow Process Corner)
Requirement: 12.899ns (axis_clk rise@12.899ns - axis_clk rise@0.000ns)
Data Path Delay: 12.794ns (logic 8.624ns (67.404%) route 4.170ns (32.596%))
Logic Levels: 11 (CARRY4=5 DSP48E1=2 LUT2=2 LUT6=2)
Clock Path Skew: -0.145ns (DCD - SCD + CPR)
  Destination Clock Delay (DCD): 2.128ns = ( 15.027 - 12.899 )
  Source Clock Delay (SCD): 2.456ns
  Clock Pessimism Removal (CPR): 0.184ns
Clock Uncertainty: 0.035ns ((TSJ^2 + TIJ^2)^1/2 + DJ) / 2 + PE
  Total System Jitter (TSJ): 0.071ns
  Total Input Jitter (TIJ): 0.000ns
  Discrete Jitter (DJ): 0.000ns
  Phase Error (PE): 0.000ns
```

Simulation Waveform, show

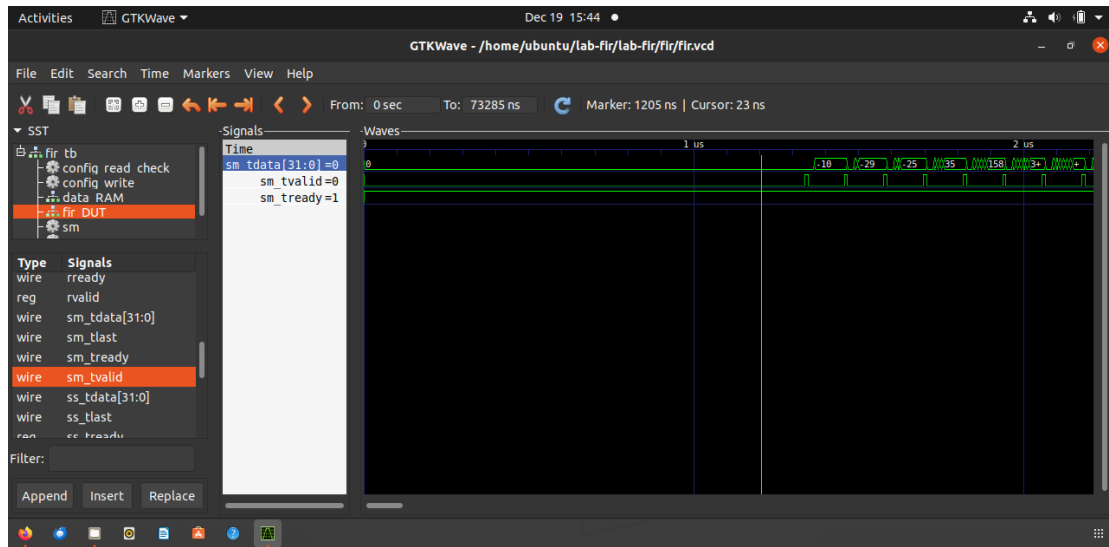
- Coefficient program, and read back



- Data-in stream-in



- Data-out stream-out



- RAM access control

