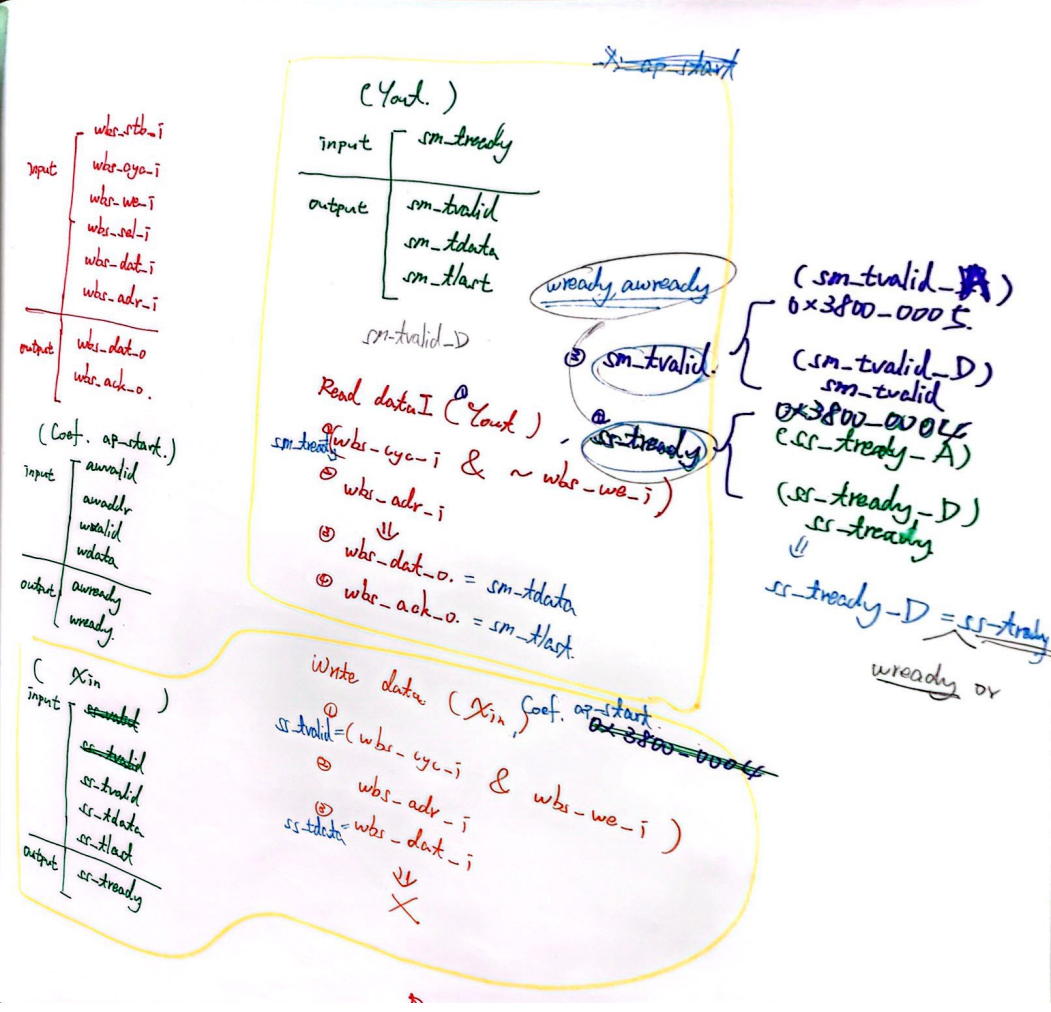


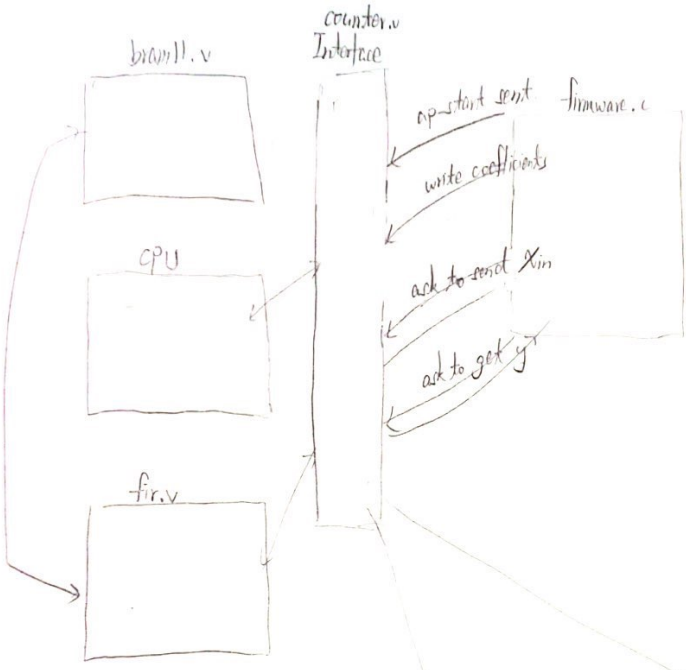
Scanned with CamScanner

Write data II (Coef, ap_start)

wvalid = awvalid ① = (wbs_cyc_i & wbs_we_i)

awaddr ② = wbs_adr_i

wdata ③ = wbs_dat



MMIO

ap_start	0x3000_0000
fir_coef_reg	0x3800_0040
fir_x_reg	0x3800_0080
fir_y_reg	0x3800_0084
fir_begin_send_x	0x3000_0004
fir_able_receive_y	0x3000_0008

Coef
aw_xx

Xin
ss_xx

Yout
sm_xx

wbs_xx

(1) (wbs_cyc_i & wbs_we_i)
↓
Write data from FW → HW.

(2) (wbs_cyc_i & ~wbs_we_i)
↓
Read data from HW to FW.

↔ Need { fir_begin_send_x, fir_able_receive_y } to judge whether the data/accelerator has been prepare to send/get the corresponding data

Report : SoC lab4-2

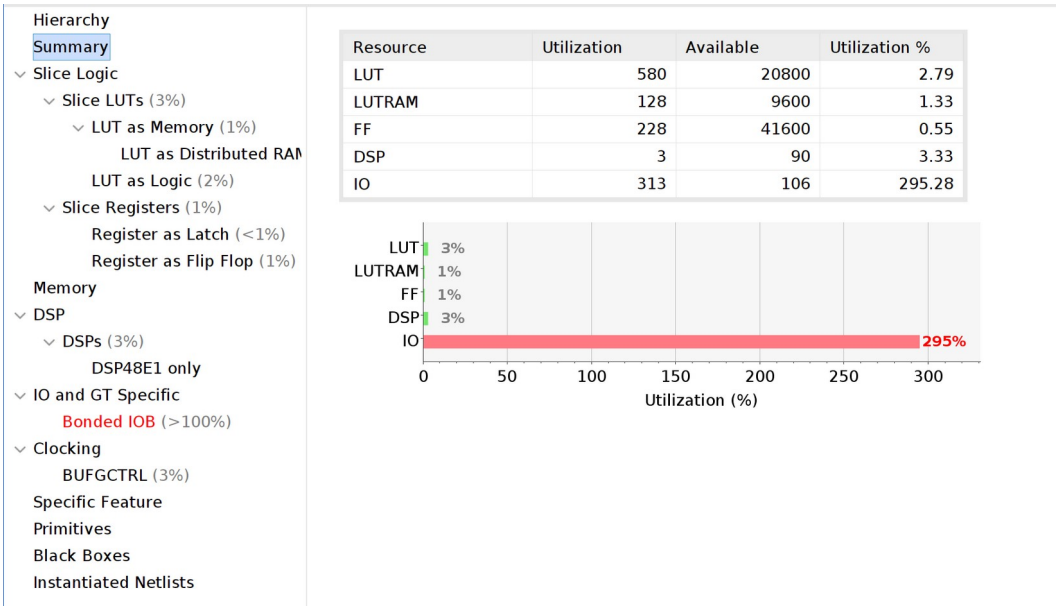
1. Introduction:

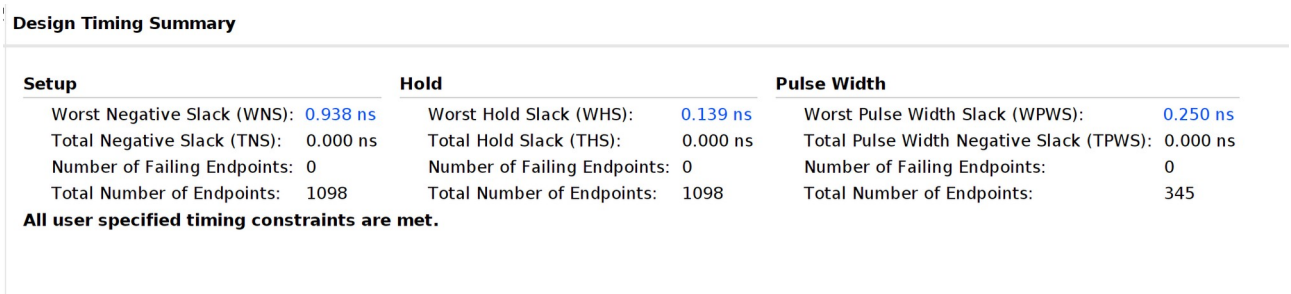
* There are two brams in the current design. One is for coefficients and the other is for the X value and the ap control values.
* The briefing data flow is as follows:

After reading one X, the Y could start be generated, and the X would be paused to generate till the Y has be computed.

Now let's turn to the firmware and accelerator interface. The data X and coefficients have been sent from firmware to hardware. Before sending the X value, firmware has to make sure the accelerator is able to get the data, and so does the Y output. Here, I create 2 registers to check the availability, fir_begin_send_x and fir_able_receive_y.

2. Area and timing report:

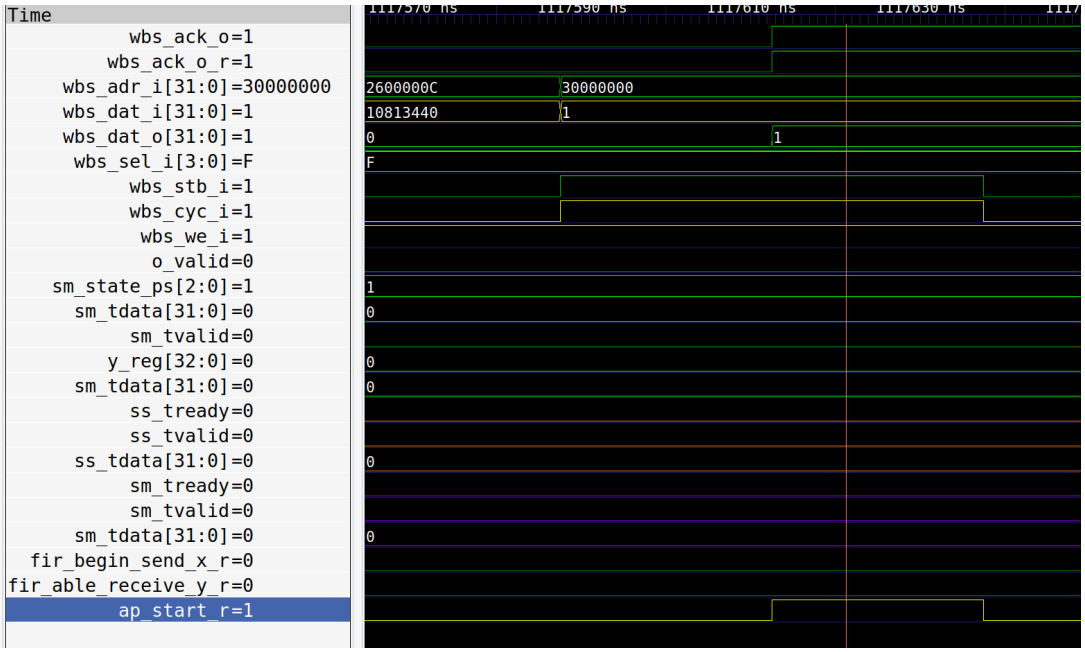




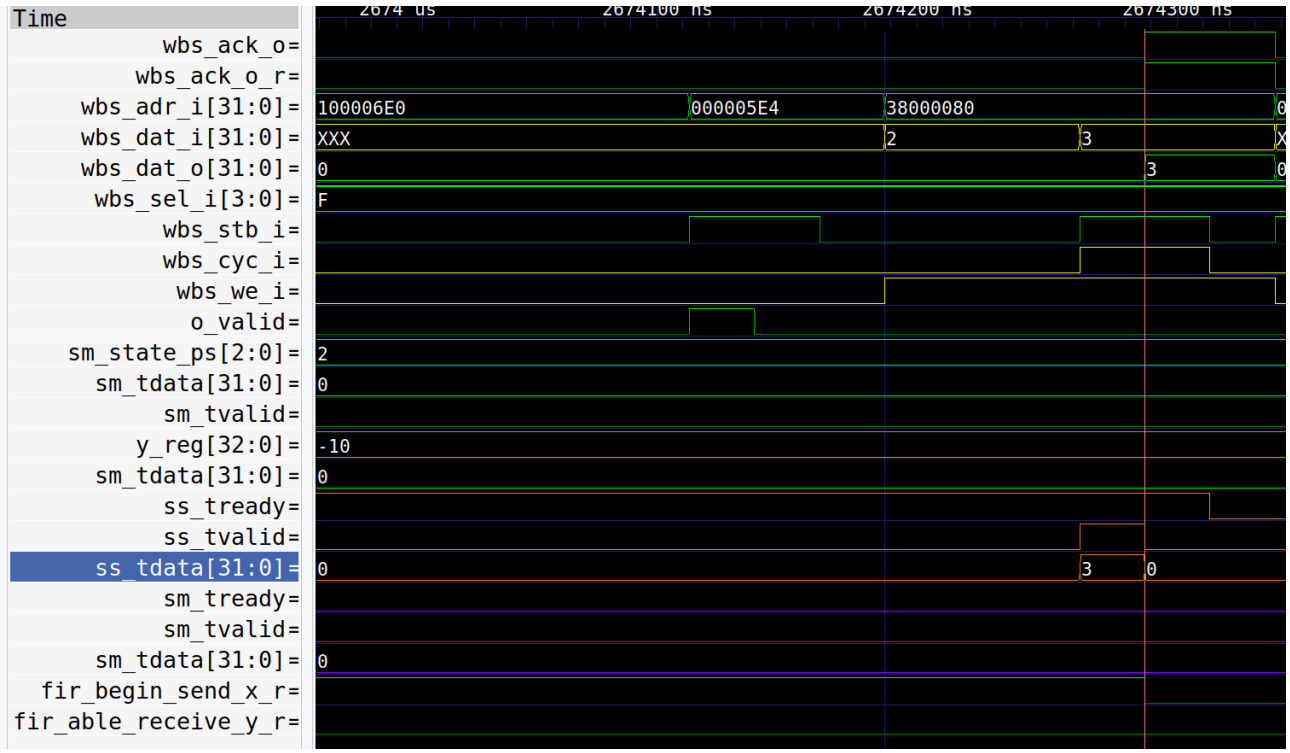
Slack is 0.938 ns under the cycle time = 15

3. Waveform and analysis:

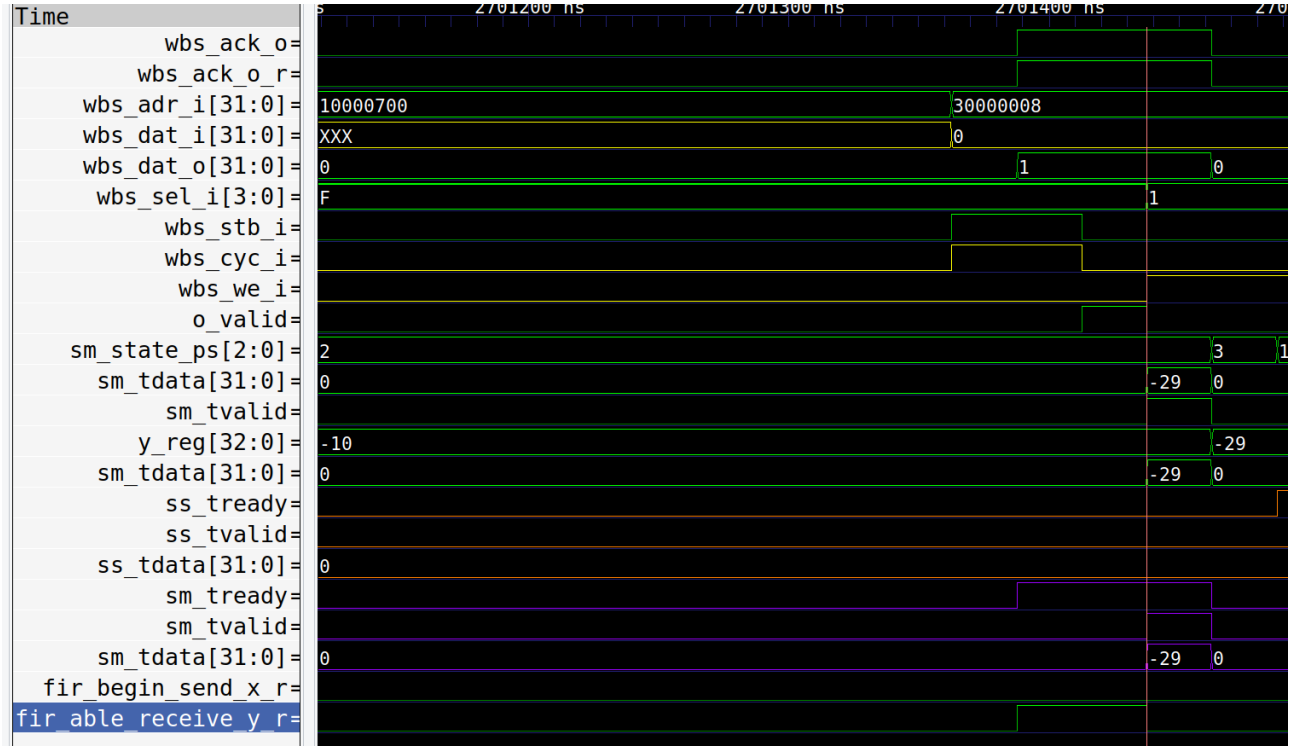
ap_start:



Xin:



Yout:



4. Let’s say the number of Yout we want to produce is N. The total Xin is approximate to 11N. The design trick used in this design is that the bram we use to store the 11 necessary X for computing a Y is 10(one of the bram block we use to store the ap signals), therefore, we have to stall there while the current Y has not been produced. The latency would be a bit longer compared to the one used the bram12. The throughput is approximately 1 OP per cycle.
The another way to improve the performance is to add some buffers to prefetch the X so that we could decrease the stall time.
One thing want to mentioned is that the design of firmware sometimes would also important for performance while integrated to the hardware accelerator. The handshake mechanism could also be improved in order to enhance the performance.