

Lab Report for Peak Detector

a lab assignment report authored by

Bowen Zhang, Weifeng Du,

Zixuan Luan, Yanyan Ge

supervised by

Dr. Shuangyi Yan,

Dr. Scott Tancock,

Dr. Tony Horseman

14 April, 2020

Contents

1	Group Composition and Task Division	2
1.1	Group Composition	2
1.2	Task Division	2
2	Project Plan	4
2.1	Plan for Team A	4
2.2	Plan for Team B	5
3	Data Processing Module by Team A	7
3.1	Part1: ANNN	7
3.2	Part2: P and L	8
4	Command Processing Module by Team B	11
5	Simulation Results	12
5.1	Simulation Results of Team A	12
5.2	Simulation Results of Team B	12
	References	12

Chapter 1

Group Composition and Task Division

In this chapter, we describe the group composition and task division among group members as well as the statement of each team member on their contribution to the project.

1.1 Group Composition

We are group 2 and our group members including Bowen Zhang, Weifeng Du, Zixuan Luan and Yanyan Ge. Bowen Zhang and Weifeng Du are in lab group A, while Zixuan Luan and Yanyan Ge are belong to lab group B.

We shall deal with the signed peak detector because our group number is even. We divided the whole group into two teams for two separate parts of the project. Team A including Bowen Zhang and Weifeng Du works on the design and implement of Command Processor. Team B including Zixuan Luan and Yanyan Ge works on the design and implement of Data Processor.

1.2 Task Division

Task division for Team A:

Task division for Team B:

Work together part:

Design the ASM chart before programming on Vivado. After finishing the separate parts, do the Stand-alone simulation for the Data Processor and debug.

Zixuan Luan:

Design the ANNN: Detect the request comes from Command Processor and fetch required number of data from Data Generator by reading ctrlIn and writing ctrlOut base on the two phase signaling protocol.

Yanyan Ge:

Desing the L and P part: Design the algorithm based on the observation on the simulation of the given synthesised version of signed Data Processor. Tackle the condition of comparing signed values and assign them to the the dataResults in proper sequence.

Chapter 2

Project Plan

In this chapter, we describe the overall project plan for Team 1 and Team 2 separately, including a diagrammatic work plan, Gantt chart, including the timelines, milestones as well as team or group meeting.

2.1 Plan for Team A

We started the project from 17th February 2020 and end the project on 17th April 2020.

On the first week (16th week), we took an overview on the given materials on the assignment, project function and the use of Vivado. Then we had a meeting on task division next Monday and came up with a roughly task division plan, then for the next few days of the 17th week, our team analysed the project and roughly worked out the ASM chart.

From 29th Feb. to 9th Mar, Zixuan Luan started to work on the communication between Data Generator and Data Processor as well as the ANNN part. From 3rd to 12th, Yanyan Ge started to work on designing the part of P and L. On 10th, our team had a meeting on reporting the ANNN task. From 11st to 18th, Zixuan started to optimize and debug the communication part. From 13rd to 21st, Yanyan Ge started to merge the ANNN part by Zixuan and the P and L part she designed before. From 22nd to 31st, the whole team worked together to do the final stand-alone simulation for the Data Processor, debugging and optimizing the whole work.

From 1st April to 4th April, the whole group worked on the final integration between

the Data processor and Command Processor. And on 5th April, we had a meeting on report writing. Finally, we started to work on writing report till 17th April.

Figure (2.1) describes the team plan in Gantt chart and the red lines in the chart show the milestones.

2.2 Plan for Team B

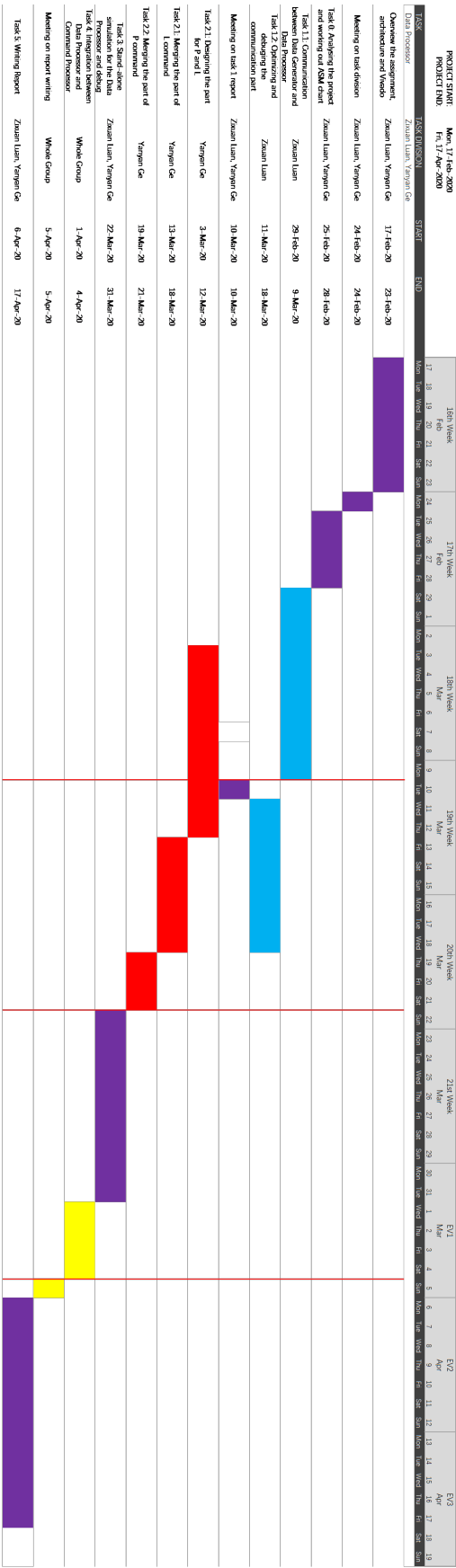


Figure 2.1: The whole project plan for Team A.

Chapter 3

Data Processing Module by Team

A

We designed and implemented the Data Processor in two parts. The first part is ANNN part which needs us to read the number comes from Command Processor and return the required number of words. The second part is PL and L part which requires us to return the largest signed data as well as the six data next to it and return the index of the largest number to the Command Processor.

3.1 Part1: ANNN

As for the first part, what we need to do first is to set up the communication between Data Generator and Data Processor using the two phase protocol. To realize the communication, we first determine “start = 1” and “curState = star_data_gen” to make sure the Data Processor fetches new data after the former data has been read and transmitted by Command Processor then we change the ctrlout signal and put the data returned from Data Generator on byte. From the simulation results, we can find that it takes only one clock cycle for Data Generator to return a new data while it takes more time for the Command Processor to finish the serial transmit.

We also set a counter to count the number of words that has been transmitted. We take the NNN of ANNN command as the final number and count up when the ctrlout signal changes and reset the counter as soon as the counter reach the final number. To achieve this

function, we use a state machine with 4 states. INIT is used as an idle state, means nothing will happen until a start signal is detected. The second state is start_data_gen, which is used to communicate and count and the next one is complete_data_gen, which writes the data fetched from Data Generator on byte. The last state is complete_all_data_gen, which is used to reset the counter and return “seqDone = 1” to Command Processor after the NNN words has been successfully fetched and transmitted.

The Block logic graph and the ASM chart of the above ANNN part is Figure (3.1)

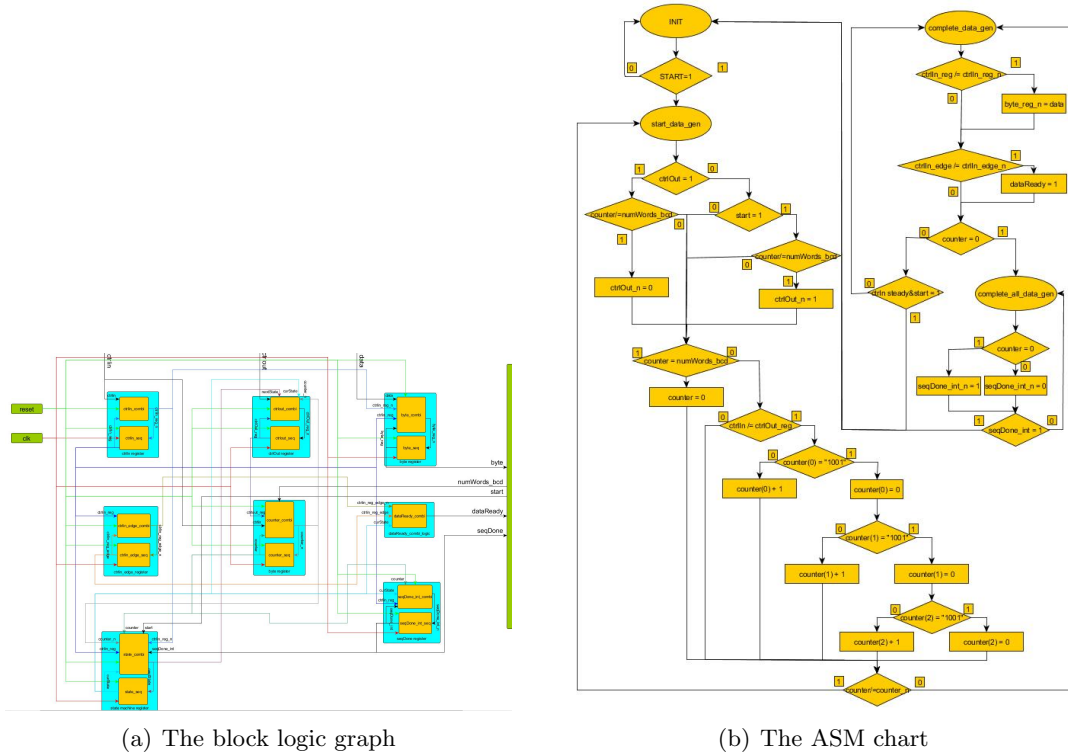


Figure 3.1: The block graph and ASM chart of ANNN part

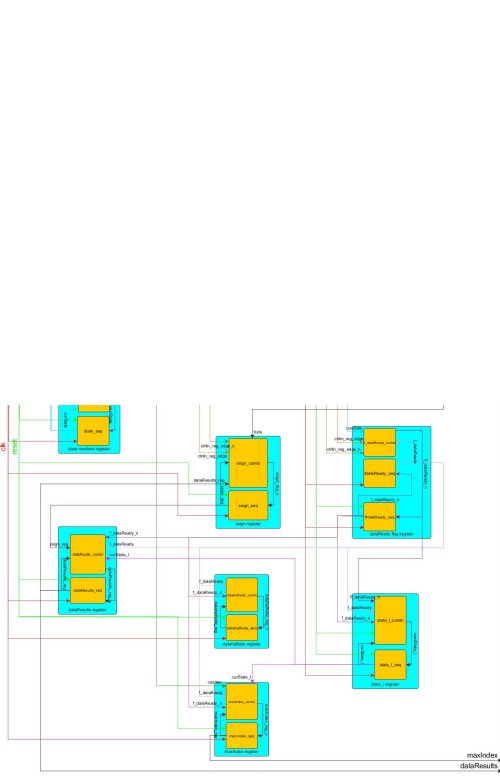
3.2 Part2: P and L

The second part is to determine the largest value and return the three numbers on both sides of it which is totally 7 bytes of data. What we do is to set a three bytes register (datahalfside register) to record the three data fetched before the current value, as soon as we determine the largest value we will substitute current middle data of dataResults array with the max data and the three bytes before which with the three bytes in datahalfside register and renew the last three bytes with three new bytes after the max data. And as

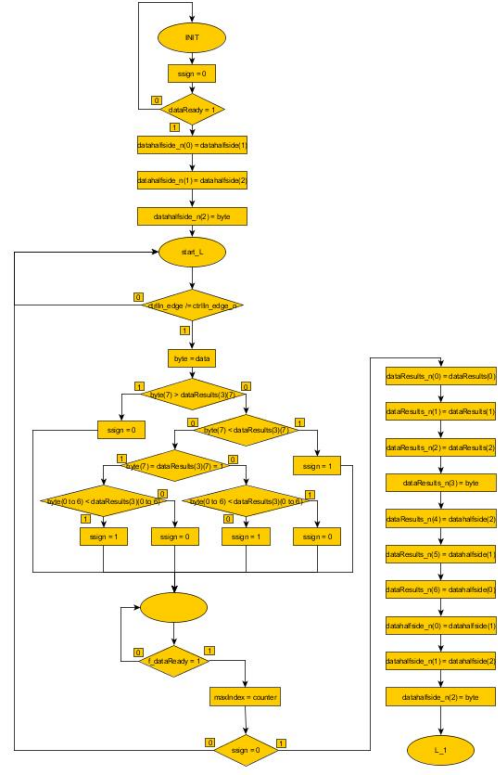
for the maxIndex, we will assign it with the value of counter as soon as we find the max data.

To realize this function, we use 6 states including INIT, start_l, L_1, L_2, L_3, L_4. Start_l is a state we jump to when we meet the max data and we do the substitution process. L_1 to L_3 is the state we use to fill the first, second and third bytes after the max data and L_4 is used to keep the dataResults don't change and record the data halvesides until the max data appears.

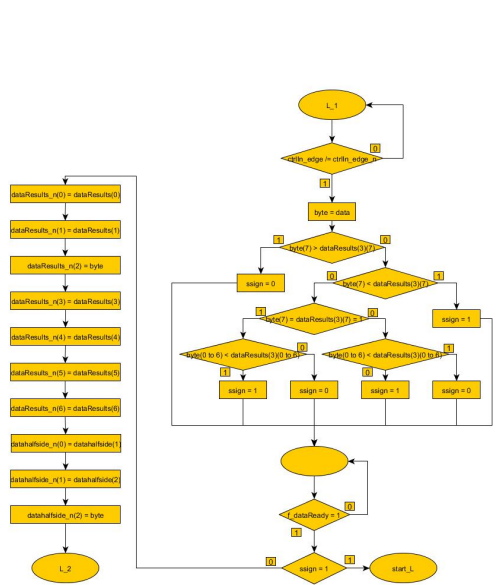
The Block logic graph and the ASM chart of the above P and L part is Figure (3.2) because the structure of state L1, L2, L3 are same, we only show the ASM chart of L1



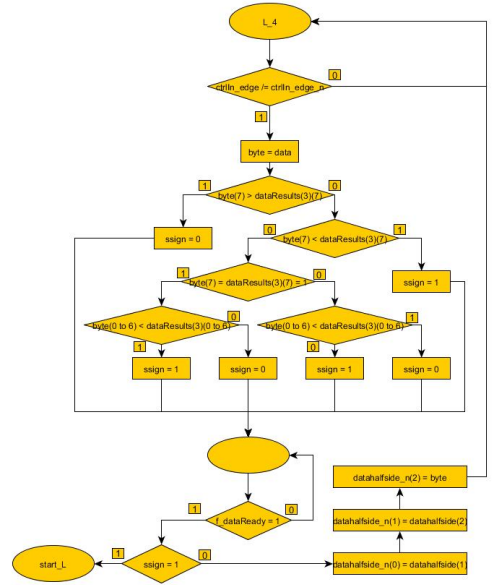
(a) The block logic graph



(b) The ASM chart (statel)



(c) The ASM chart (L1)



(d) The ASM chart (L4)

Figure 3.2: The block graph and ASM chart of P and L part

Chapter 4

Command Processing Module by Team B

Chapter 5

Simulation Results

5.1 Simulation Results of Team A

From the Figure (5.1)), we can find that the numWords comes from Command Processor is 12 and the Data Processor successfully output 12 bytes of data onto the byte signal.

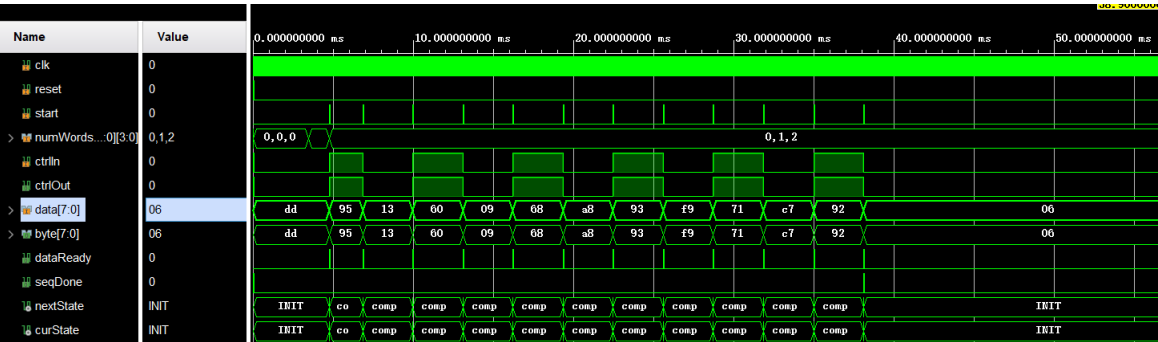


Figure 5.1: The simulation result of ANNN.

From the Figure (5.2), we can find that when a new data comes Data Processor will compare it with current max data and decide whether change the dataResults or not. Because we are group 2, among the first series of data, “71” is the largest one and the dataResults is ”06, 92, c7, 71, f9, a8” which is correct. What’s more, whenever a larger data is detected, the maxIndex will change to the index of the new data.

5.2 Simulation Results of Team B

