

## Channel Processor in 2D Cluster Finding Algorithm for High Energy Physics Application

Rourab Paul<sup>1,\*</sup>, Amlan Chakrabarti<sup>2</sup>, Jubin Mitra<sup>3</sup>,  
Shuaib A. Khan<sup>4</sup>, Sanjoy Mukherjee<sup>5</sup>, and Tapan Nayak<sup>6</sup>

<sup>1,2</sup>University of Calcutta, Kolkata, INDIA

<sup>3,4,6</sup>Variable Energy Cyclotron Centre, HBNI, Kolkata, India and

<sup>5</sup>Centre for Astroparticle Physics and Space science, Bose Institute, Kolkata, India

### Introduction

In A Large Ion Collider Experiment (ALICE) at CERN 1 TB/s (approximately) [1] data comes from front end electronics. Previously, we had 1 GBT link operated with a cluster clock frequencies of 133 MHz and 320 MHz in Run 1 and Run 2 respectively. The cluster algorithm proposed in Run 1 and 2 could not work in Run 3 as the data speed increased almost 20 times. Older version cluster algorithm receives data sequentially as a stream. It has 2 main sub processes. 1.Channel Processor, 2.Merging process. The initial step of channel processor finds a peak  $Q_{max}$  and sums up pads (sensors) data from -2 time bin to +2 time bin in the time direction. The computed value stores in a register named cluster fragment data (*cfdo*). The merging process merges *cfdo* in pad direction.

The data streams in Run 2 comes sequentially, which processed by the channel processor and merging block in a sequential manner with very less resource over head. In Run 3 data comes paralelly, 1600 data from 1600 pads of a single time instant comes at each 200 ns interval (5 MHz) which is very challenging to process in the budgeted resource platform of Arria 10 FPGA hardware with 250 to 320 MHz cluster clock.

### 1. Channel Processor

In Run 3 the cluster block has 2 major components. **I.**Channel processor to calculate *cfdo* in time direction. **II.**Merging block to merge data in pad direction. Here we

are proposing a channel processor which contains 3 different components **1.** cp\_alu\_w5 **2.**buffer7x1600, **3.**cp\_alu\_controller.

#### A. nxcp\_alu\_w5

The features of *nxcp\_alu\_w5* are

1. We assume  $n$  number of cp\_alu\_w5 will process 1600 Pads ( $n < 1600$ ) in time division manner. At a time simultaneously  $n$  number of cp\_alu\_w5 can process  $n$  number pads. These  $n$  number of pads are named as 1 pad chunk which is processed by nxcp\_alu\_w5.
2. Each cp\_alu\_w5 needs few clocks (320 MHz) to process 1 pad data. These "few clocks" is named as one time instant. In 1st time instant  $n$  number of cp\_alu\_w5 named as nxcp\_alu\_w5 will process  $n$  number of pad data. In 2nd time instant the same nxcp\_alu\_w5 will process another  $n$  number of pad data. Similarly cp\_alu\_w5 will cover whole 1600 pads. Here number of time instants= $1600/n$ .
3. Inputs of nxcp\_alu\_w5 are Row, Pad,

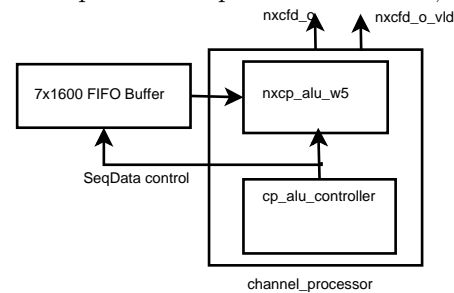


FIG. 1: Proposed Channel Processor

FLPad, Time, Charge and outputs are *cfdo*, *cfdo\_vld*. The function of cp\_alu\_w5 is to find  $Q_{max}$  and summing up pad parameters around  $Q_{max}$  from -2 time bin to +2 time bin. In table II it is shown that for each pad region 1 CRU will be dedicated which means 1 pad region data (maximum 1600 pads in pad

\*Electronic address: rourabpaul@gmail.com

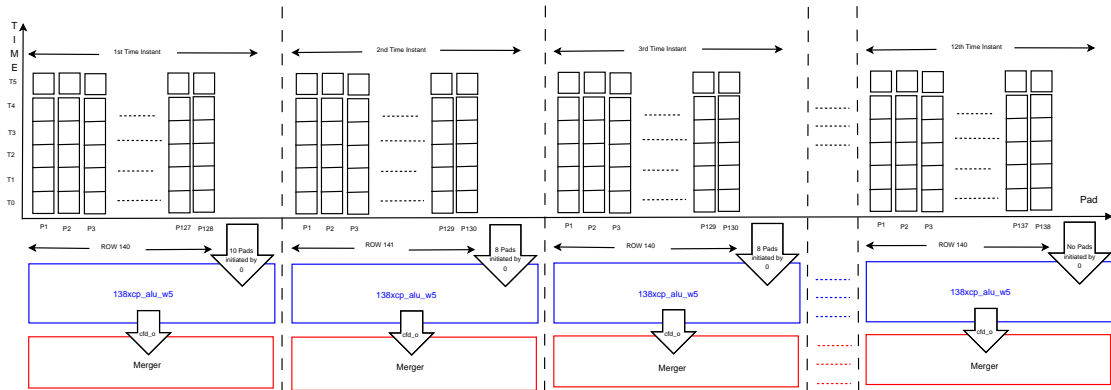


FIG. 2: Top level architecture of 2D Cluster Finding

region needs to be processed by 2D cluster algorithm. 1 pad chunk works on 1 row. The size of the pad chunk (n) is chosen according to the maximum number of pads in 1 row of 1 pad region. The details is shown in table II.

**B. Buffer7x1600**

This buffer consists write clock frequency @5 MHz (SAMPA Clock) and read clock frequency ≈@320 MHz (Cluster Clock), which is implemented in block RAM area of FPGA. Each data are 35 bits which has 4 parts charge(10 bits), time(10 bits), Pad (8 bits), row(6 bits) and a F/L Flag (1 bits). For OROC Pad Region-9 has maximum 1600 data. If we buffer 7 timebins in this block for cp\_alu\_w5, the total block RAM bits will be 35x1600x7=392000.

**C. cp\_alu\_controller**

cp\_alu\_controller generates 3 control signals for cp\_alu\_w5. SeqDataStart becomes high to indicate start of data stream. SeqDatavalid is to indicate valid data stream. As we have 7 data in time direction in a single frame, SeqDatavalid sustains high for 7 cluster clocks. SeqDataEnd flags the end of data stream. The resource usage for 300xcp\_alu\_w5 and along with 1600 pads are shown in table I.

**D. Implementation**

1. We consider 7xn frames to calculate cfd.o at a time instant which means in time direction we have 7 data and in pad direction we have n data.
2. We consider the available number of clocks

are  $\frac{320MHz}{5MHz} = 64$

3. Number of clocks needed to find peaks over 7 timbins are 9.

4.  $\frac{\text{number of rows per pad region}}{\text{number of rows process at one time instant}} \times 9 < 64.$

TABLE I: Resource of Channel Processor & 1xp\_alu\_w5

#	ALMs	Registers	DSPs	Block RAM
Channel Processor	344722	438041	1200	392000
1xp_alu_w5 With Gain	507	1510	7	
1xp_alu_w5 Without Gain	392	1180	4	

Lets us assume number of rows per pad region is  $R_p$  and number of row processes at one time instant is  $R_t$ . So  $\frac{R_p}{R_t} \times 9 < 64$ . Fig 2 shows the top level hardware block of 2D cluster finding algorithm

TABLE II: IROC

# of row /Pad Region	Row Range Pad Region	Pad Region	# Pads Range/Row	# Pads /Pad Region	# of rows process at a time	# of pads process at a time (n)
IROC						
17	0 to 16	0	66 to 76	1200	3	3x76=228
15	17 to 31	1	76 to 84	1200	3	3x84=252
16	32 to 47	2	86 to 94	1440	3	3x94=282
15	48 to 62	3	92 to 100	1440	3	3x100=300
OROC						
18	63 to 80	4	76 to 84	1440	3	3x84=252
16	81 to 96	5	86 to 94	1440	3	3x94=282
16	97 to 112	6	94 to 106	1600	3	3x106=318
14	113 to 126	7	110 to 118	1600	2	2x118=236
13	127 to 139	8	118 to 128	1600	2	2x139=278
12	140 to 151	9	128 to 138	1600	2	2x138=276

**Acknowledgments**

For the detailed information of the cluster algorithm we are thankful to TPC-CRU Group, CERN, Switzerland.

**References**

- [1] "Technical Design Report for the Upgrade of the ALICE Read-out & Trigger System" ALICE Collaboration.