

An efficient approach to manage DMA descriptors and evaluate PCIe based DMA performance for ALICE Common Readout Unit(CRU)

Project leaders

Tapan Kumar Nayak
Tivadar Kiss

Senior Developers/Coordinators

Erno David
Filippo Costa
Jozsef Imrek
Amlan Chakrabarti

Developers

Jubin Mitra
Sanjoy Mukherjee*
Shuaib Khan
Tuan Mate Nguyen
Pascal Boeschoten
Rourab Paul

Agenda

- ❑ Overall view of CRU

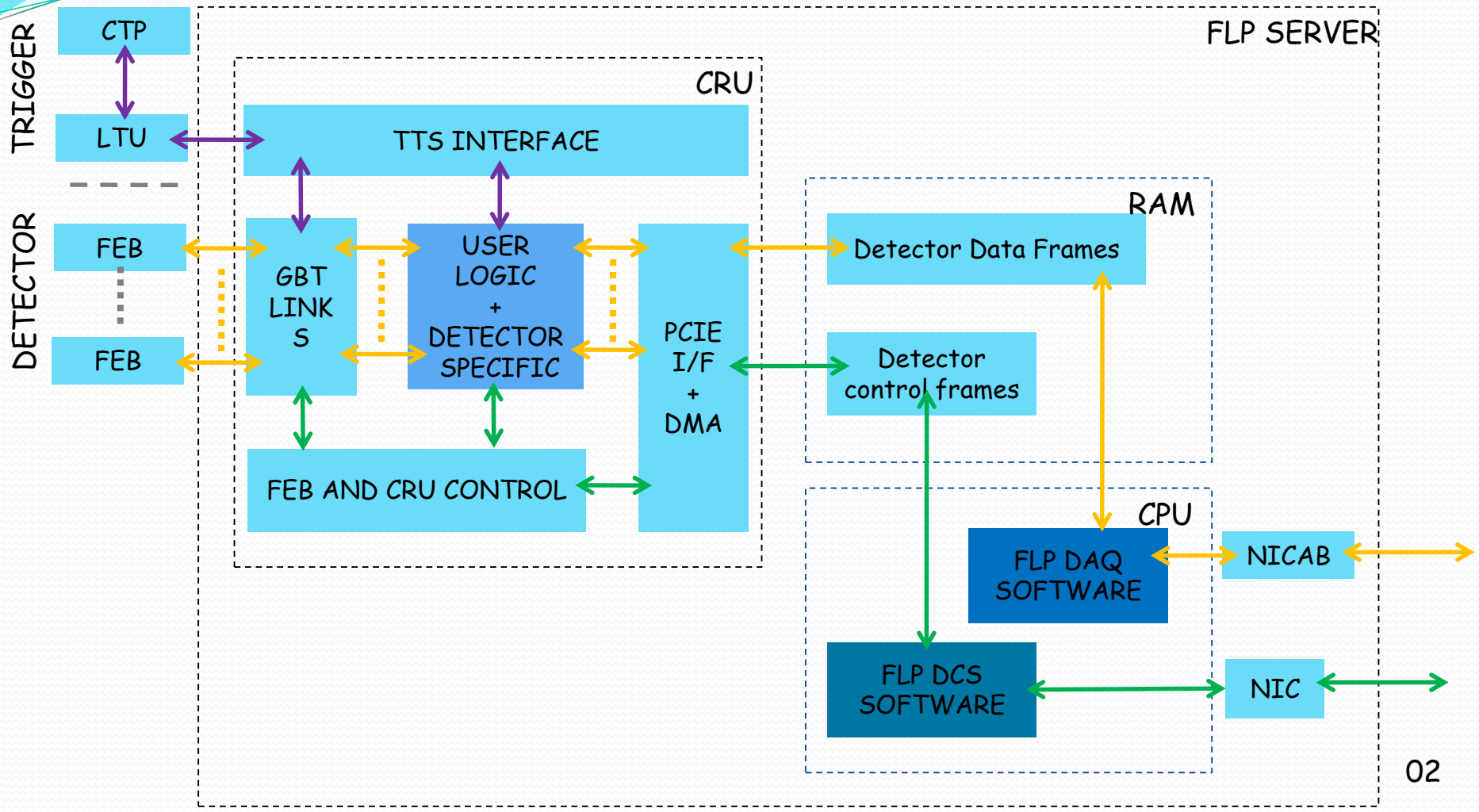
- ❑ PCIe-DMA
 - ❑ Basics
 - ❑ Architecture (FPGA based)

- ❑ DMA descriptor management

- ❑ User logic internal architecture(Block Diagram)

- ❑ Features of user logic

- ❑ Result



02

- **PCIe (Peripheral Component Interconnect) :**
 1. Protocol implemented in FPGA as per PCIe-SIG standard
 2. Can handle input BW (avg ~96 Gbps) as PCIe offers large BW (User BW ~114 Gbps in Gen3 x16 mode)
 3. Consume no FPGA resource as it is available as Hard IP from Altera.
- **DMA (Direct Memory Access) :**
 1. Protocol to push data through PCIe interface.
 2. DMA evaluation goal : Use as much possible the PCIe- BW.
 3. Mainly based on Look Up table approach known as descriptor table.

DMA PROFILE:

Throughput :

1. Old : ~51 Gbps

2. New :

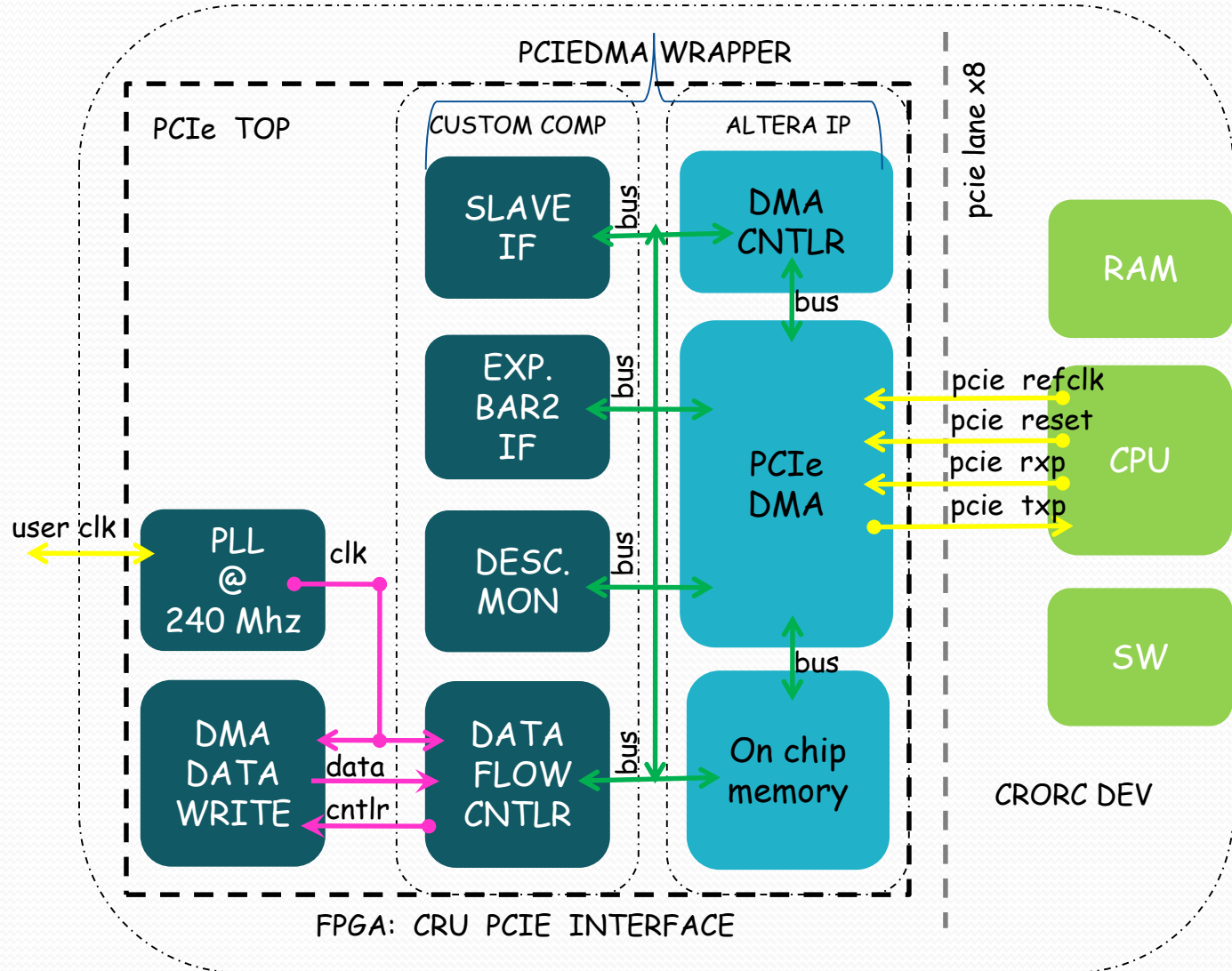
A. With clean FW:
~53 Gbps

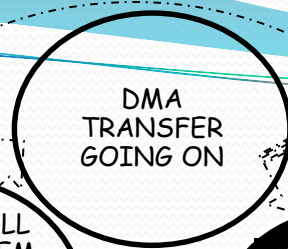
B. With clean FW
and SW:
~55Gbps

PCIe DMA idle time :

MAX : ~3 us

AVG : ~160 ns





SRC	SIZE	DST
SRC	8K	DST
SRC	8K	DST
SRC	8K	DST
SRC	8K	DST

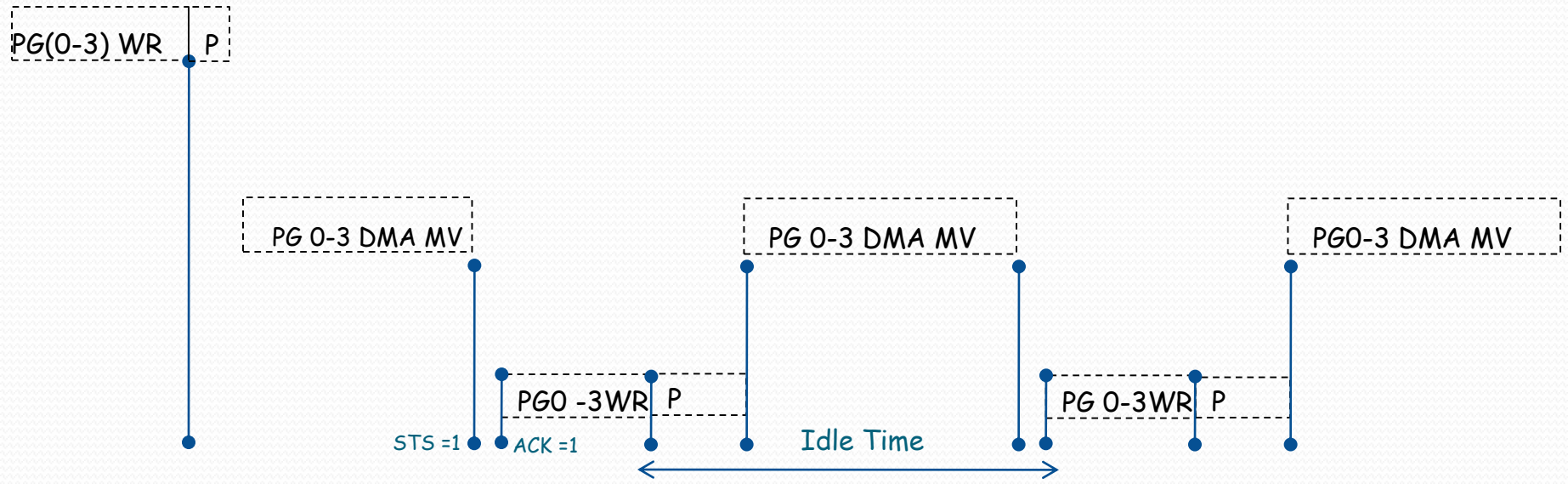
SRC	8K	DST
SRC	8K	DST
SRC	8K	DST
SRC	8K	DST

SRC	SIZE	DST
SRC	8K	DST
SRC	8K	DST
SRC	8K	DST
SRC	8K	DST

SRC	8K	DST
SRC	8K	DST
SRC	8K	DST
SRC	8K	DST

CARD FIFO

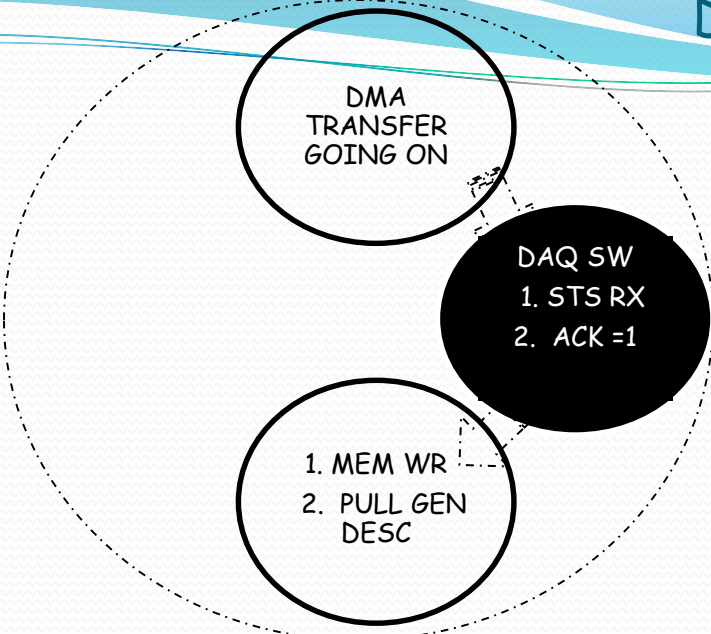
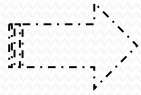
SW FIFO





PIPELINED DMA DESC. MANAGEMENT

FW WR FULL
ON CHIP MEM &
PULL FOUR
DESCRIPTORS



SRC	SIZE	DST
SRC	8K	DST
SRC	8K	DST
SRC	8K	DST
SRC	8K	DST

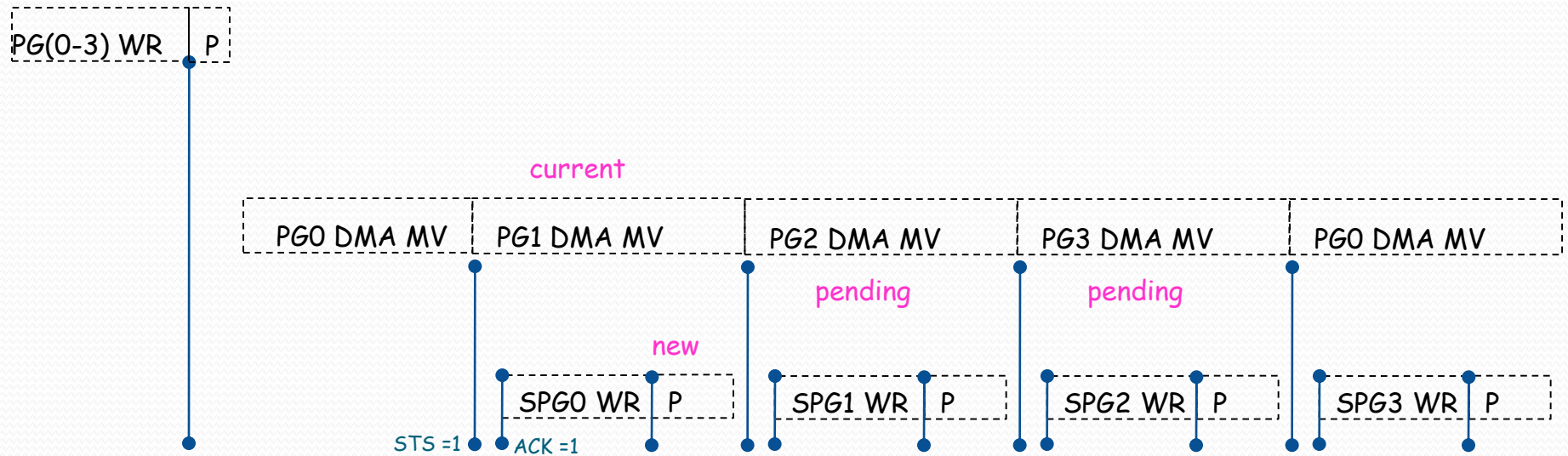
SRC	8K	DST
SRC	8K	DST
SRC	8K	DST
SRC	8K	DST

SRC	SIZE	DST
SRC	8K	DST
SRC	8K	DST
SRC	8K	DST
SRC	8K	DST

CARD FIFO

SRC	8K	DST
SRC	8K	DST
SRC	8K	DST
SRC	8K	DST

SW FIFO



	OLD DMA IF LOGIC	NEW DMA IF LOGIC
1.	No handshaking signal from FW side to SW side to trigger the DMA	Ready signal from FW side to SW side to fetch the data
2.	DMA reg. programming was done by SW	Later ready signal is used internally by FW for DMA programming
3.	Status memory has been used for ack. generation	Status memory has been used for ack. generation
4.	No monitoring signal for debugging	Added several monitoring processes to debug the current status of DMA transfer
5.	Good for DMA performance evaluation.	Good for DMA evaluation plus data consistency.



Run time :
5 Mins =>

```

00:05:00 243851392 n/a
Interrupted

Seconds      300.487
Pages       243856941
Bytes       1.9976e+12
GB          1997.68
GB/s        6.64813
Gb/s       53.185
GiB        1860.48
GiB/s      6.19155
Gibit/s    49.5324
Errors     0
idle_cnt lower 0xf4bd4b4a
idle_cnt upper 0x2

```

Run time :
30 Mins =>

```

Interrupted

Seconds      1801.18
Pages       1412352930
Bytes       1.157e+13
GB          11570
GB/s        6.42356
Gb/s       51.3885
GiB        10775.4
GiB/s      5.98241
Gibit/s    47.8593
Errors     0
idle_cnt lower 0xa972dd8a
idle_cnt upper 0x14

[root@crorcdev ~]#

```

Run time :
60 Mins =>

```

Interrupted

Seconds      3606.78
Pages       2961090180
Bytes       2.42575e+13
GB          24257.3
GB/s        6.72547
Gb/s       53.8037
GiB        22591.3
GiB/s      6.23358
Gibit/s    50.1086
Errors     0
idle_cnt lower 0x73a01348
idle_cnt upper 0x21

[root@crorcdev ~]#

```

Run time :
120 Mins =>

```

Interrupted

Seconds      7212.13
Pages       5921179944
Bytes       1.85065e+13
GB          48506.3
GB/s        6.72566
Gb/s       53.8053
GiB        45175
GiB/s      6.26376
Gibit/s    50.1101
Errors     0
idle_cnt lower 0xe141378b
idle_cnt upper 0x42

root@crorcdev ~]#

```

THANK YOU