FEC Firmware optimization for the VMM frontend

Yan Huang, CCNU Wuhan

CERN/RD51 and CERN/ESS team

Outline

- Motivation
- Default project
- My project
- Results
- Summary

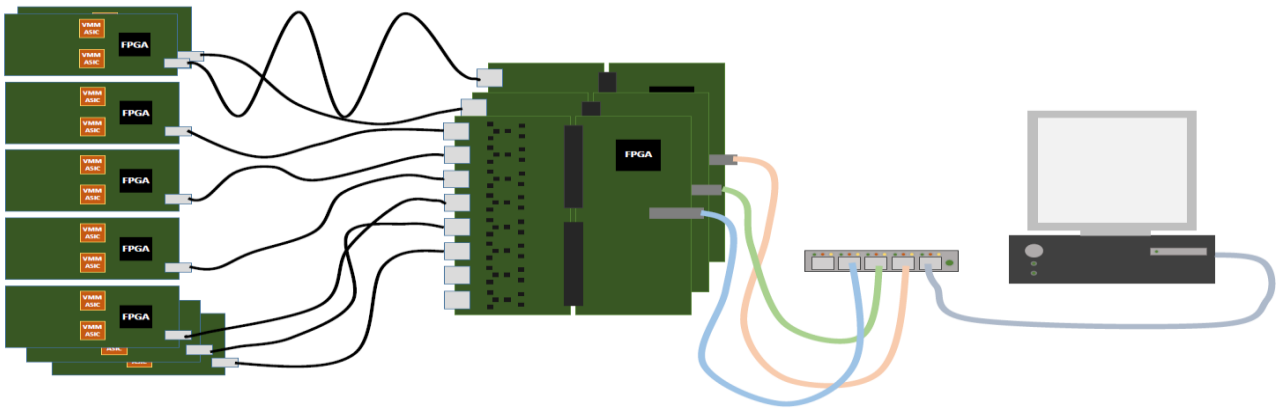
Motivation

- APV trigger rate: 1 KHz
- VMM : self triggered, can have 1000 times more data than APV !

FPGA in FEC card: **Virtex-6**

Firmware must be optimized first for current VMM hybrid:

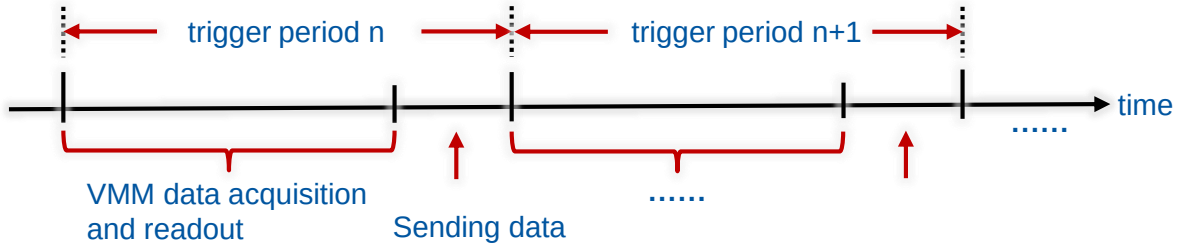
Continuous data flow from up to 16 VMM hybrids needs to be combined and controlled.

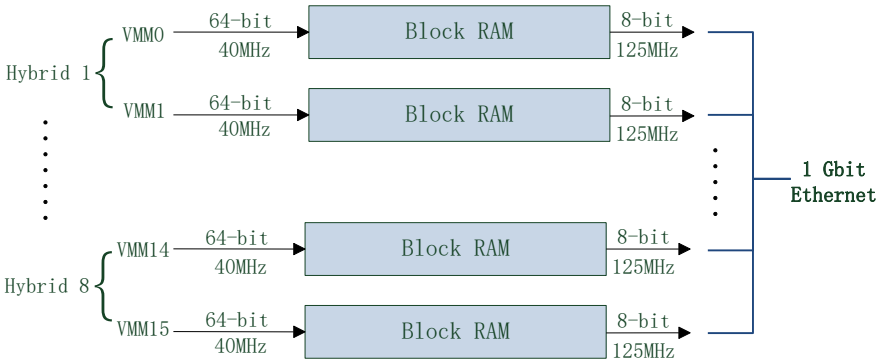


VMM Hybrid → HDMI cable → Adapter card + FEC → Ethernet → Switch →Ethernet → PC

Current SRS System

Default project





- 8 hybrids, 16 VMM ASICs in total.
- 1 hit : 38 bit of data → blown up to 64 bit to fit old APV data format.
- One VMM, one frame.

Default project problems

- Hits can be lost
- Limited number of hits
- Sequential operation induces dead time
- Ethernet bandwidth wasted by 0s and many small packets

Goals for my improved project

- No hit loss
- Throughput of datapath high enough for full VMM readout speed
- Parallel readout of data from VMM and sending though Ethernet
- Optimal usage of Ethernet bandwidth

➔ Steady flow of data at high speed

Unambiguous assignment of time needs to preserved.

Time information

- In data : 20 bit time information (bcid:12 bit + tdc:8 bit)

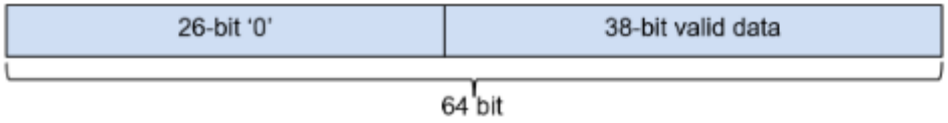
  → unique assignment within one trigger period.
- Default project : one trigger period = one Ethernet package with all hits + SRS timestamp + trigger period count.
- My project : continuous data flow

→ Timestamp needs to be in data

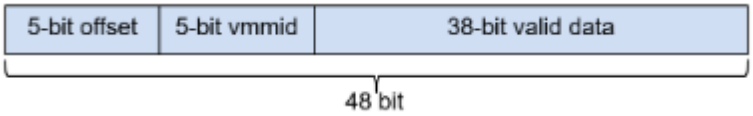
→ datamarkers + new data format

Data format comparison

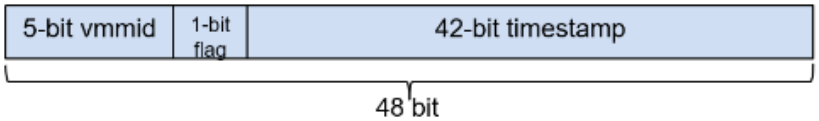
old data format :



new data format :



datamarker :



New data format : at least 38 bit + 5 bit to identify VMM in combined dataflow

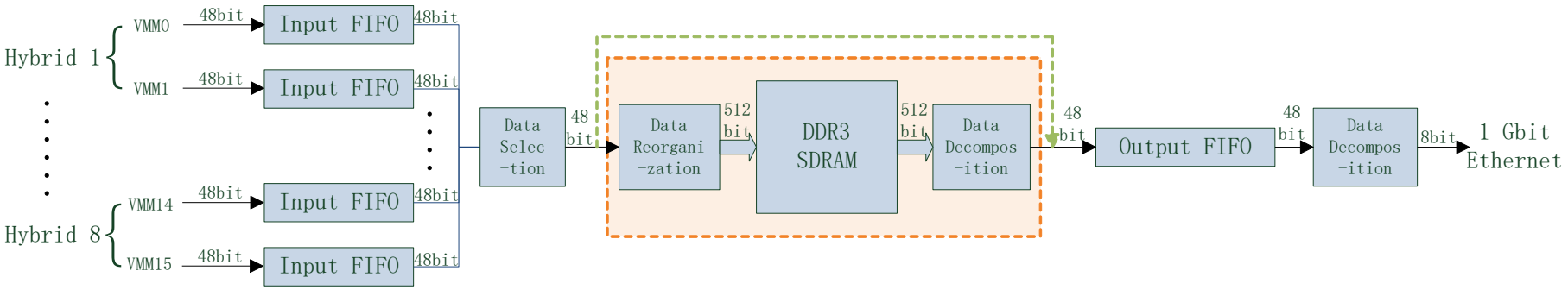
⇒ 5 bit left unused

⇒ use as an offset to the last timestamp in datamarker

⇒ datamarker only needed when offset overflows (reduce Ethernet payload)

- Every $2^5 = 32$ periods, there will be some new datamarkers.
- Flag : '0' → datamarker , '1' → hit (part of 38-bit valid data).
- More hits can be transmitted during the same time

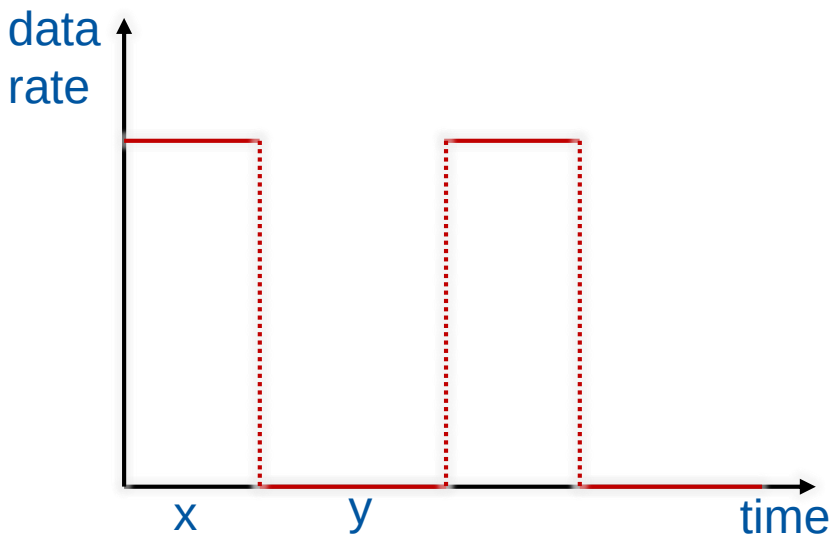
My project:



- Data selection with single hit token from VMM input FIFO
- DDR3 memory can be used or not.
  - 10 input data (10*48 bit) + 32bit ‘0’ = one ddr3 data (512bit)

  ⇒ 2GB DDR3 can store **335,544,320** hits.
- Send out jumbo frame when it is full.

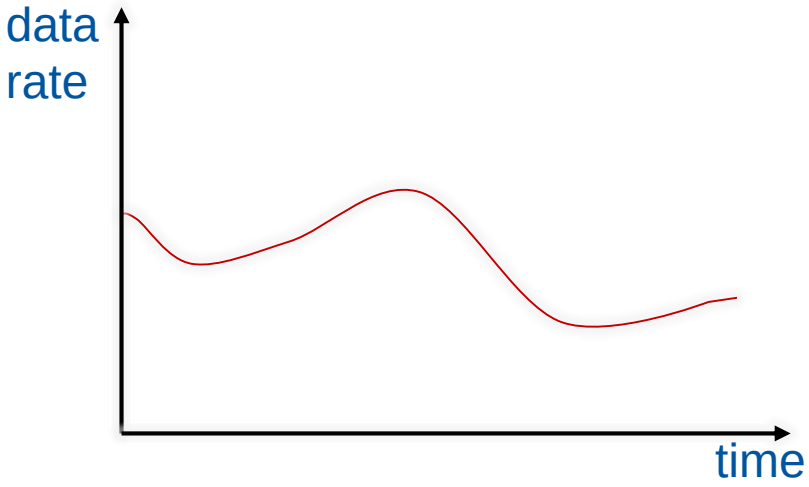
Example



In the simple case where data is generated by a pulsed machine during some time x, followed by a pause time y.

One can make the DDR3 buffer large enough such that all data generated during time x fits conveniently without overflow.
Then, during the pause time y, the buffer is emptied and written to the online system.

Example

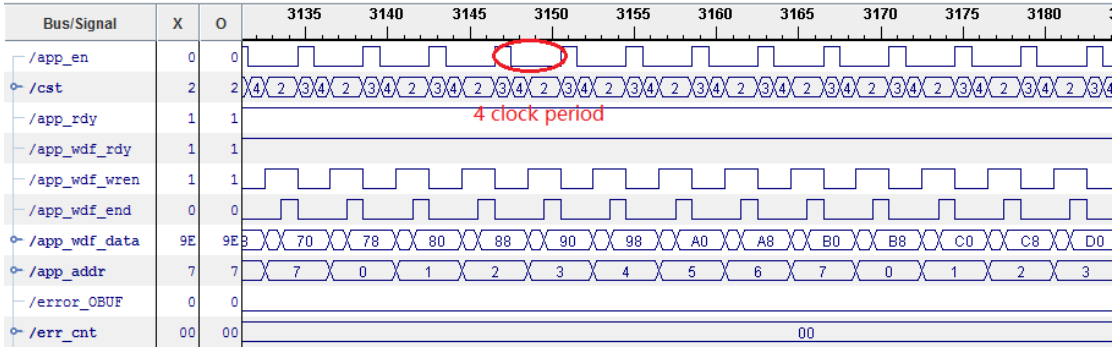


In the more difficult case, data is generated permanently at some fluctuating input rate there is no pause. The data must get read out simultaneously as it is written in.

As long as the average input rate *Rin* is only a little less than the average output rate *Rout*, a maximum readout performance can be achieved without data loss.

Results

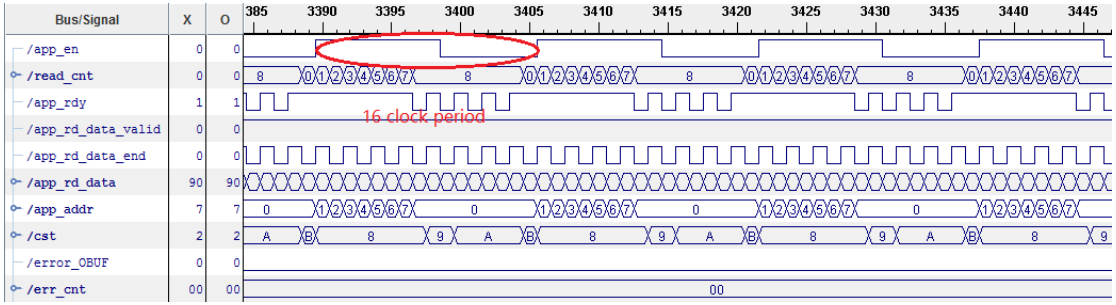
R/W bandwidth test of DDR3 memory



Write Timing Diagram of DDR3

writing one burst word (512 bit) takes 4 user clock periods (200MHz), one can thus get the write bandwidth: $512 \div 8 \div (4 \times 5) = 3.2$ Gbyte.

Reading eight burst words takes 16 user clock periods, so read bandwidth can be calculated as: $512 \times 8 \div 8 \div (16 \times 5) = 6.4$ GByte.



Read Timing Diagram of DDR3

Comparison data format

Test condition: 6 VMMs are selected, 4 of them are connected and have data

Previous:

8	0.000015036	10.0.0.2	10.0.0.3	SRSVMM2	166	VMM: 0, Hits: 14
9	0.000016237	10.0.0.2	10.0.0.3	SRSVMM2	78	VMM: 1, Hits: 3
10	0.000017439	10.0.0.2	10.0.0.3	SRSVMM2	60	VMM: 2, Hits: 0
11	0.000018663	10.0.0.2	10.0.0.3	SRSVMM2	60	VMM: 3, Hits: 0
12	0.000019878	10.0.0.2	10.0.0.3	SRSVMM2	150	VMM: 14, Hits: 12
13	0.000021076	10.0.0.2	10.0.0.3	SRSVMM2	150	VMM: 15, Hits: 12
14	0.000207162	10.0.0.2	10.0.0.3	SRSVMM2	60	End of Frame

Now:

7	0.019464445	10.0.0.2	10.0.0.3	SRSVMM…	8992	FEC: 0, Hits: 1489
8	0.019628312	10.0.0.2	10.0.0.3	SRSVMM…	60	End of Frame

Comparison data format

Previous:

```
▼ SRS Header
    Frame Counter: 11836579 (11836578)
    Data Id: VMM2 Data
    VMM ID: 1
    SRS Timestamp: 679673856 (679608320)
  ▼ Hit:   1, ch: 12, bcid:   46, tdc:   70, adc:  368, flag: 1, othr: 1
    ▶ Data1 00a7980e, (7019e500)
    ▶ Data2 cc000000, (00000033)
  ▶ Hit:   2, ch: 13, bcid:   46, tdc:   75, adc:  378, flag: 1, othr: 1
  ▶ Hit:   3, ch: 14, bcid:   46, tdc:   72, adc:  371, flag: 1, othr: 1
  ▶ Hit:   4, ch: 15, bcid:   46, tdc:   72, adc:  384, flag: 1, othr: 1
  ▶ Hit:   5, ch: 16, bcid:   46, tdc:   64, adc:  348, flag: 1, othr: 1
  ▶ Hit:   6, ch: 18, bcid:   46, tdc:   64, adc:  373, flag: 1, othr: 1
  ▶ Hit:   7, ch: 19, bcid:   46, tdc:   68, adc:  384, flag: 1, othr: 1
  ▶ Hit:   8, ch: 20, bcid:   47, tdc:  249, adc:  408, flag: 1, othr: 1
```

64 bit

Now:

48 bit

```
▼ SRS Header
    Frame Counter: 14 (13)
    Data Id: VMM3a Data
    FEC ID: 0
    UDP Timestamp: 11665408 (11075584)
    Offset overflow last frame: 49167
  ▼ Hit:   1, offset: 6, vmmID: 14, ch: 54, bcid:   46, tdc:   35, adc:  349, over thr: 1
    ▶ Data1 33aea9c0, (039575cc)
    ▶ Data2 dbc6, (63db0000)
  ▶ Hit:   2, offset: 6, vmmID: 15, ch: 55, bcid:   46, tdc:   43, adc:  318, over thr: 1
  ▶ Hit:   3, offset: 6, vmmID: 14, ch: 55, bcid:   46, tdc:   49, adc:  255, over thr: 1
  ▶ Hit:   4, offset: 6, vmmID: 15, ch: 56, bcid:   46, tdc:   16, adc:  336, over thr: 1
  ▶ Hit:   5, offset: 6, vmmID: 14, ch: 56, bcid:   46, tdc:   31, adc:  370, over thr: 1
  ▶ Hit:   6, offset: 6, vmmID: 15, ch: 57, bcid:   46, tdc:   24, adc:  339, over thr: 1
  ▶ Hit:   7, offset: 6, vmmID: 14, ch: 57, bcid:   46, tdc:   59, adc:  363, over thr: 1
  ▶ Hit:   8, offset: 6, vmmID: 15, ch: 58, bcid:   46, tdc:   15, adc:  352, over thr: 1
```

Comparison data format

```
▶ Hit: 731, offset: 31, vmmID:  0, ch: 20, bcid:   46, tdc:   46, adc:  385, over thr: 1
▶ Hit: 732, offset: 31, vmmID: 14, ch: 11, bcid:   46, tdc:   44, adc:  335, over thr: 1
▶ Hit: 733, offset: 31, vmmID:  0, ch: 21, bcid:   46, tdc:   47, adc:  400, over thr: 1
▶ Hit: 734, offset: 31, vmmID:  0, ch:  3, bcid:   46, tdc:   55, adc:  414, over thr: 1
▶ Marker:   1, SRS timestamp: 327680101, vmmid: 0
▶ Marker:   2, SRS timestamp: 327680101, vmmid: 1
▶ Marker:   3, SRS timestamp: 327680101, vmmid: 2
▶ Marker:   4, SRS timestamp: 327680101, vmmid: 3
▶ Marker:   5, SRS timestamp: 327680101, vmmid: 14
▶ Marker:   6, SRS timestamp: 327680101, vmmid: 15
▶ Hit: 735, offset: 0, vmmID: 15, ch:  0, bcid:   46, tdc:   24, adc:  316, over thr: 1
▶ Hit: 736, offset: 0, vmmID:  0, ch:  4, bcid:   46, tdc:    5, adc:  391, over thr: 1
▶ Hit: 737, offset: 0, vmmID:  1, ch:  1, bcid:   46, tdc:   17, adc:  405, over thr: 1
▶ Hit: 738, offset: 0, vmmID: 14, ch:  0, bcid:   46, tdc:   55, adc:  280, over thr: 1
```

datamarker

Outlook

- Current focus on continuous mode.
- Triggered mode to be implemented soon.

  → Acquire and store data until there is a trigger in, then read out data which are needed.
- More intense testing of firmware.
- Use new implementation at test beam.

Summary

FEC firmware is advancing very well

- A big DDR3 buffer is added to the project
- Data format has been changed into 48 bit
- Bandwidth of SRS is increased
- A new mode will be implemented

Thank you!