Development of high rate MWPC and data compression function with FADC (II)

Nguyen Minh Truong (Osaka U.),

Coworker: Masaharu Aoki(Osaka U.), Youichi Igarashi(KEK), Masatoshi Saito(KEK), Hiroaki Natori (KEK), Nguyen Duy Thong (Osaka U.)

Open-It, Nov. 20th, 2014, at J-PARC

Contents

- Motivation
- FADC readout board

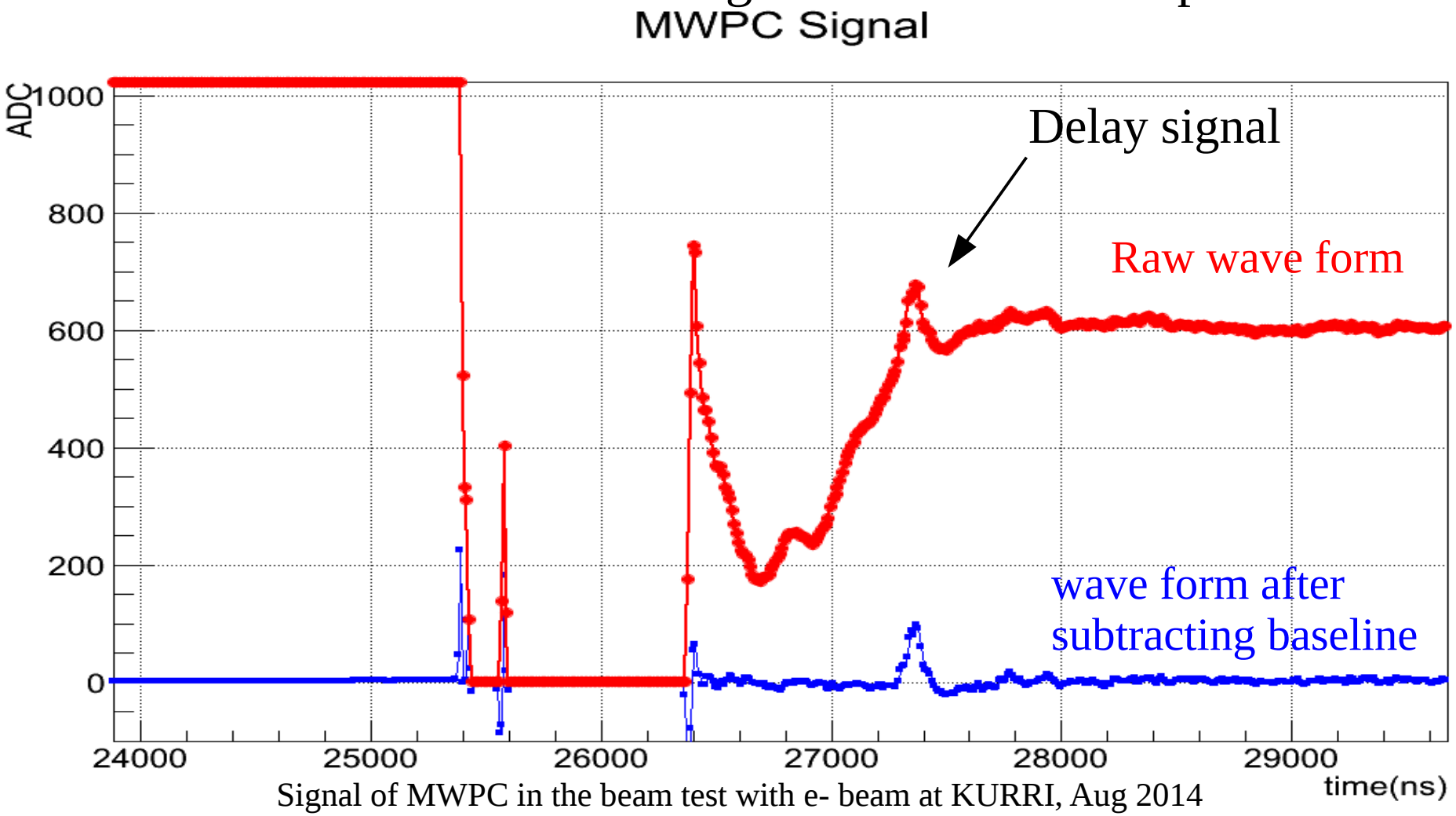
  + Original firmware design

  + New firmware design
- Test new firmware design

- Summary

Motivation

- MWPC will be used to take signal from DeeMe experiment



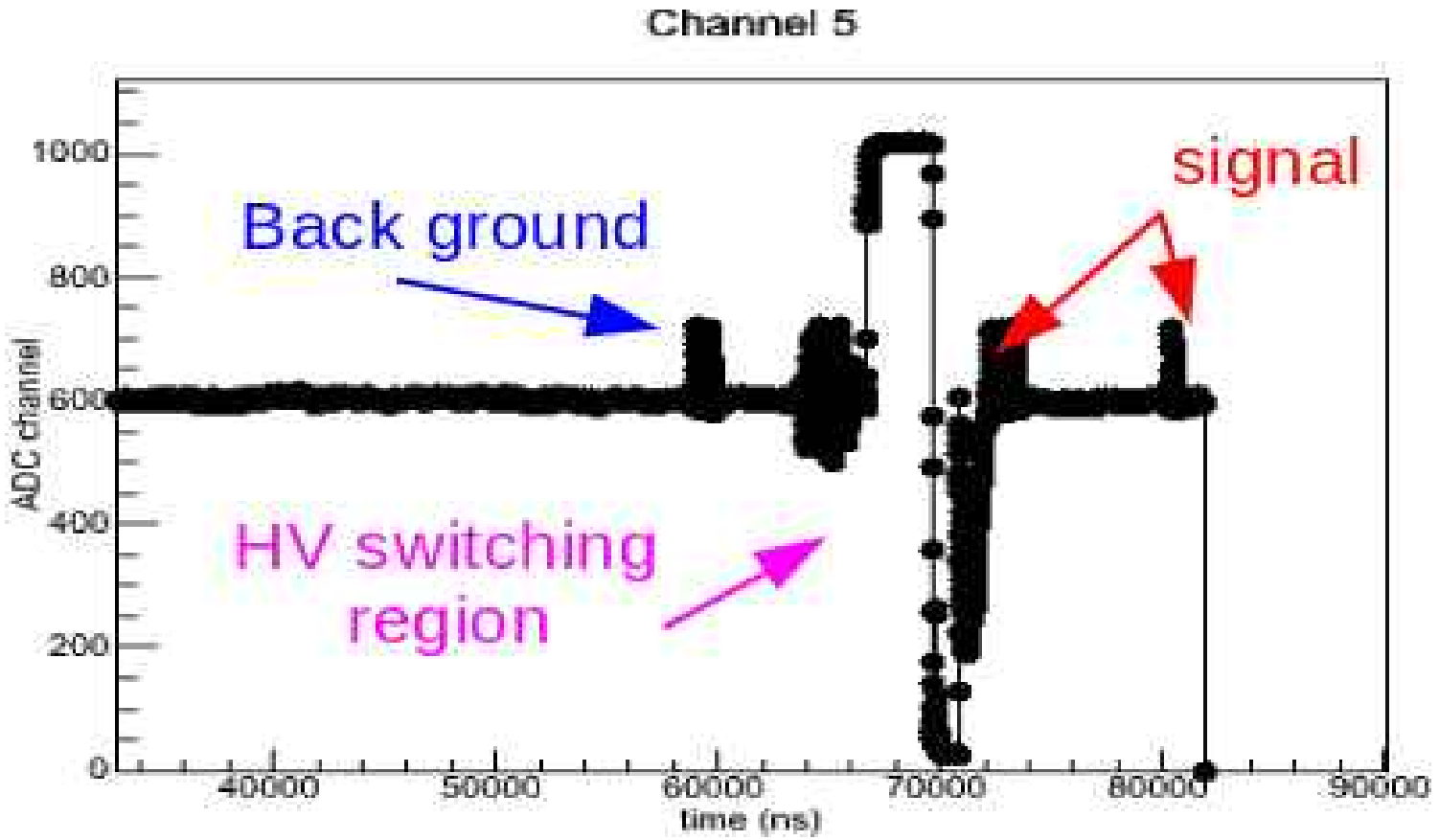
Signal of MWPC in the beam test with e- beam at KURRI, Aug 2014

We want to see delay signal but the base line is not flats

=> should use FADC board to readout signal

Motivation

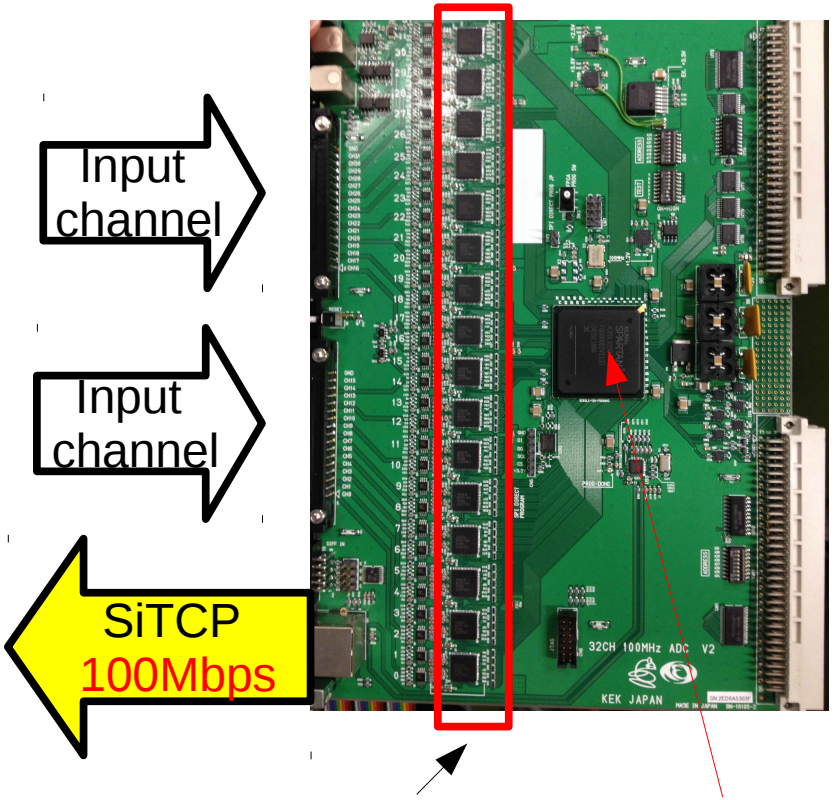
- In order to monitor beam off timing background, we want to read out data with time length as much as possible (~ 80 micro second)



Recode time length as long as possible

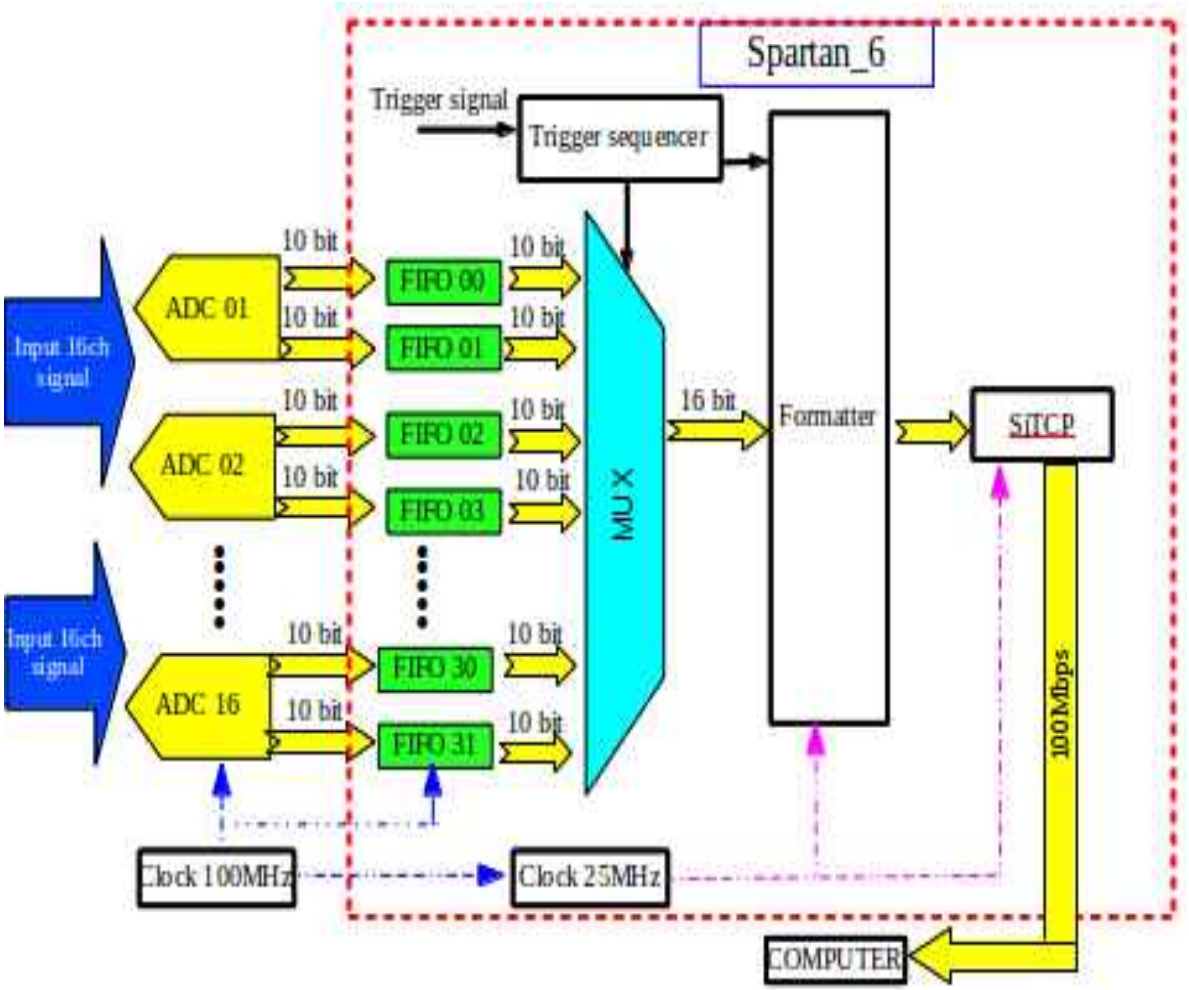
Original Readout FADC board

10-bits 100-MHz FADC system developed by IGARASHI Youichi for TREK experiment



16 FADC
(AD9216 100MHz)
2input and 2output/
1fadc

Spartan 6



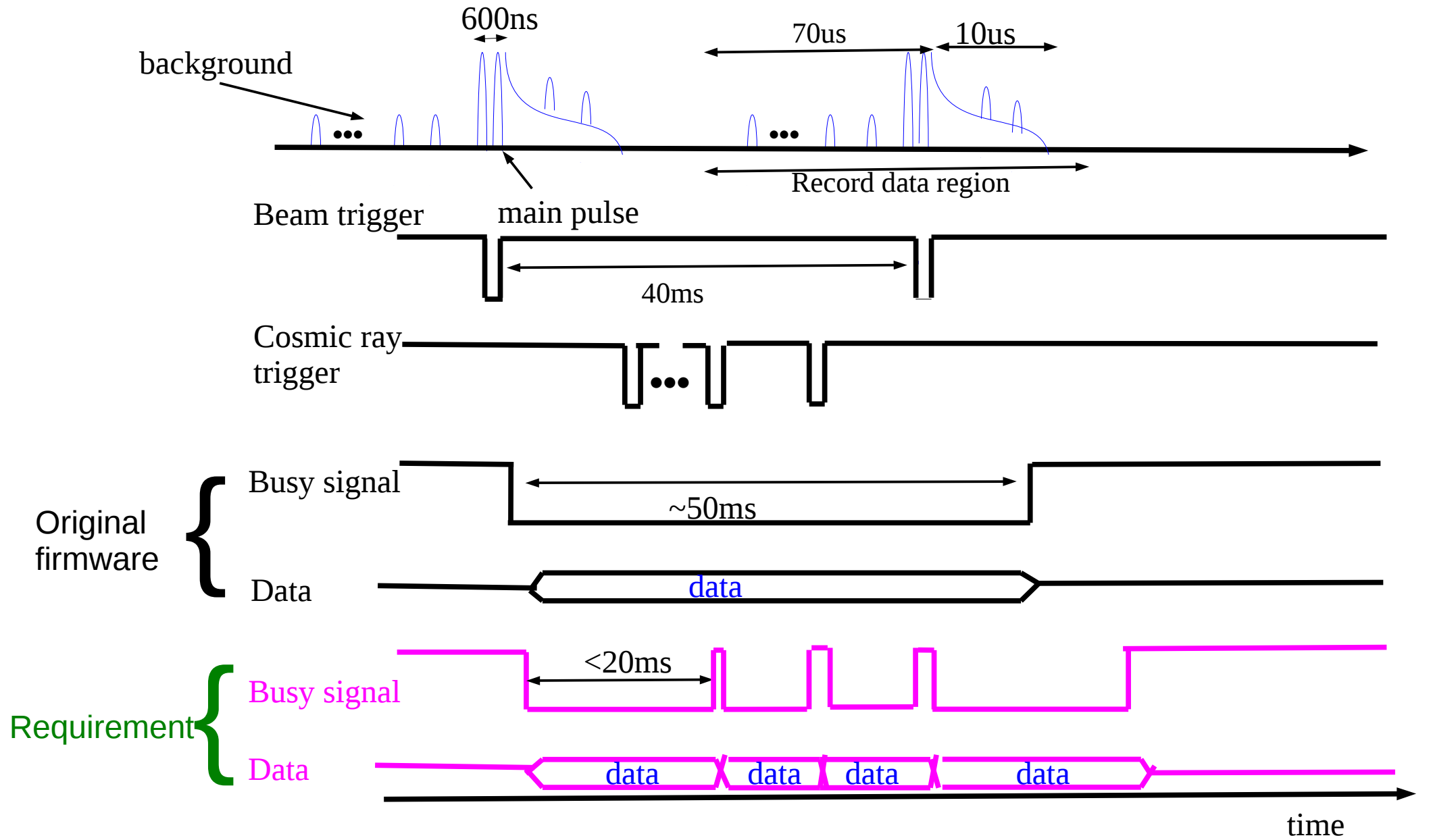
- Data speed transfer is 20 events/ s → dead time ~ 50ms

+ 32 channels/event

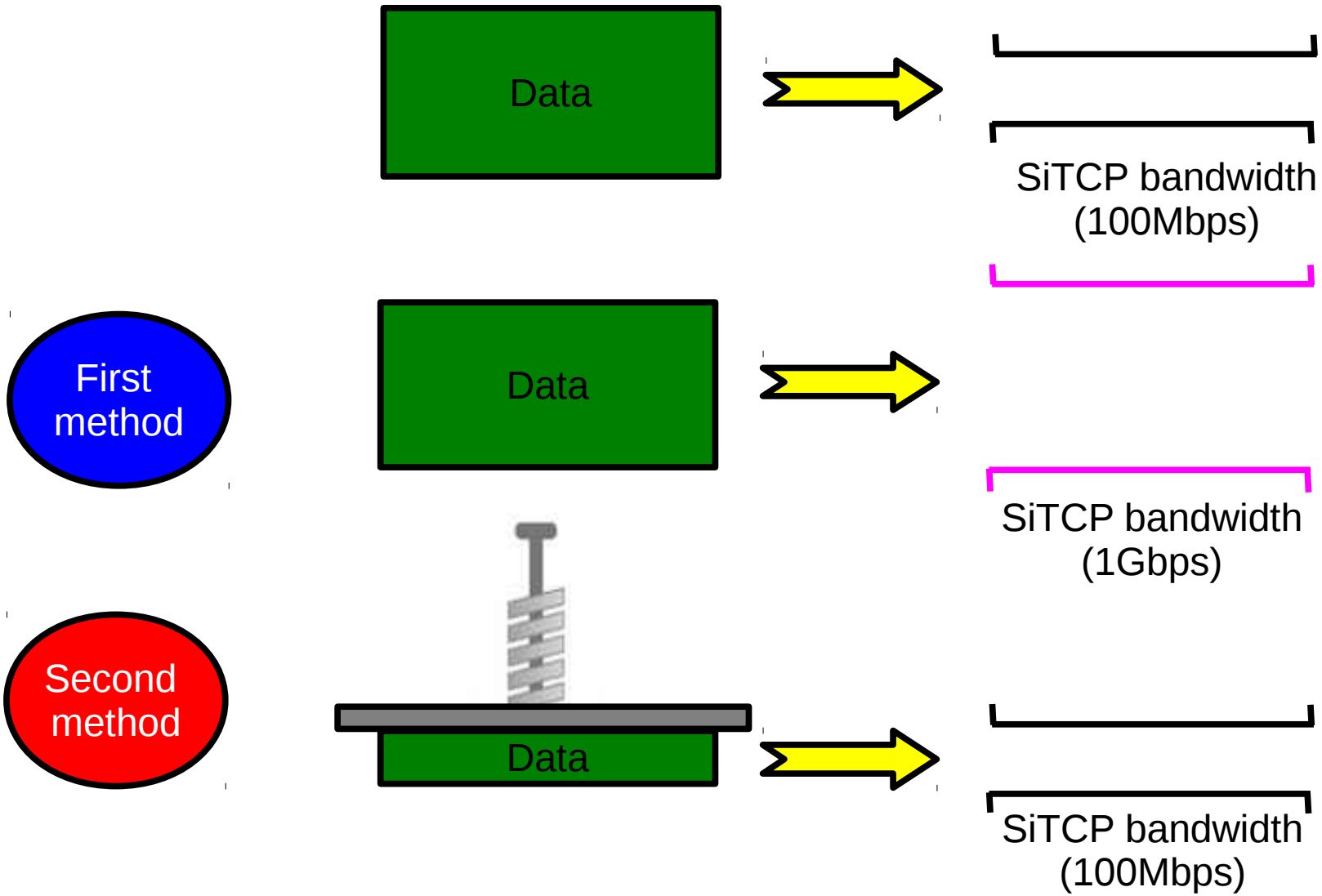
+ 8192 sample points /channel

Dead Time Of Original Firmware

(with 8192 sample point)



Original Readout FADC board



- To monitor cosmic rays background, the dead time of readout board should small than 20ms but original firmware have dead time ~50ms

$$\Rightarrow \text{compression ratio} = \frac{\text{Data before compress}}{\text{Data after compress}} > 2.5$$

New design for FADC board

- Fast data transfer → dead time < 20ms
- Moulder design

  +Extensibility → User can easy modify for their experiment

  +Easy for debug
- Multiple trigger:

  +External trigger

  +Self trigger
- Have slow control to control number of sample points, number of channels, threshold, …
- Time stamp and Event tag to synchronize multiple FADC readout board

Data format of FADC Readout board

Word	Event Format
0xFAFA	Begin of Event
0xF1F2	Byte oder, 0xF1F2 = Big Endian, 0xF2F1 = Little Endian
0xA1A1 or 0xB1B1	Trigger type: 0xA1A1 = External trigger 0xB1B1 = Self trigger
$10a^7b^7$	a^7 = Header Format Version Code, b^7 = Firmware Version Code
$10c^{14}$	c^{14} = Module ID, lower 14-bits of module's IP address
$10d^{14}$	$d^{14}e^{14}$ = Local Event Number,total 28bits
$10e^{14}$	
$10f^9g^5$	f^9 = reserved, g^9 = Event tag
$10h^{14}$	$h^{14}i^{14}j^{14}k^{14}$ = Local Time stamp, total 56 bits
$10i^{14}$	
$10j^{14}$	
$10k^{14}$	
	Channel data format
0xFBFB	End of Event Data

Word	Channel Format
0xFFFC	Start of Channel Data Block
0xFC01	Module ID
$10l^{14}$	$l^{14}m^{14}n^{14}$ = Bit-Mask of Active channels
$10m^{14}$	
$10n^{14}$	
$0xFFq^8$	Start of channel q^8 = channel number
	Compressor data format
0xFDFD	End of Channel
$0xFFq^8$	Start of channel q^8 = channel number
	Compressor data format
0xFDFD	End of Channel
●●●	●●●
0xFFFD	End of Channel Data Block

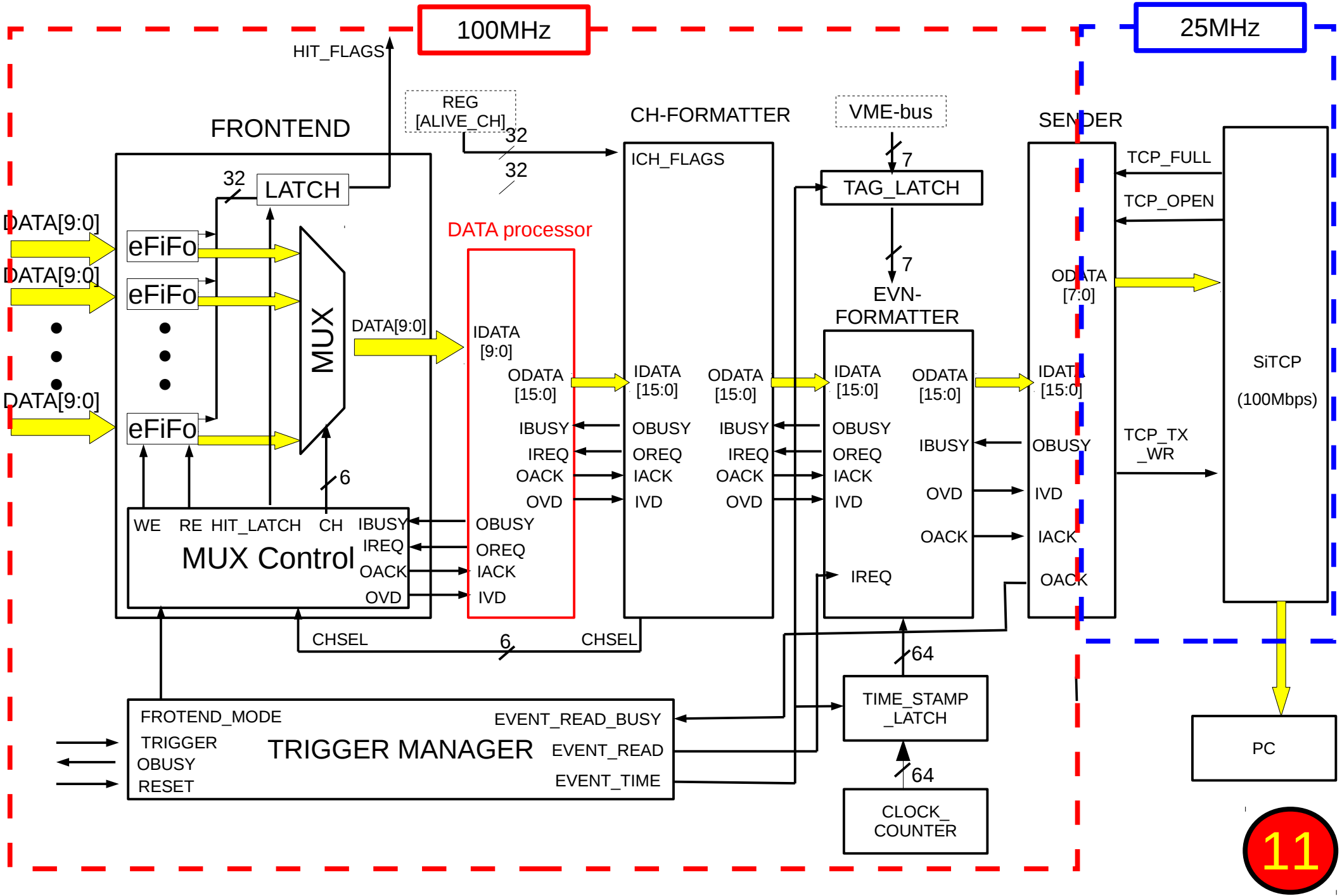
Comp. format

Comp. format

Data format of compressor

Begin of Compressor	**0xFEFE**									
Field size (raw data)	0	0	0	0						
Raw next	1									
Raw data	x	x	x	x	x	x	x	x	x	x
Raw data next	1									
Raw data	x	x	x	x	x	x	x	x	x	x
End raw data	0									
Field size(3-bits delta)	0	0	1	1						
3-bits delta	x	x	x							
3-bits delta	x	x	x							
●●●	●●●	●●●	●●●							
End of 3-bits delta	1	0	0							
Field size(n-bits delta)	n	n	n	n						
n-bits delta	x	x	x	x	●●●	x				
●●●	x	x	x	x	●●●	x				
End of n bits delta	1	0	0	0	●●●	0				
End of delta compressor stream	1	1	1	1						
End of Compressor	**0xFEFD**									

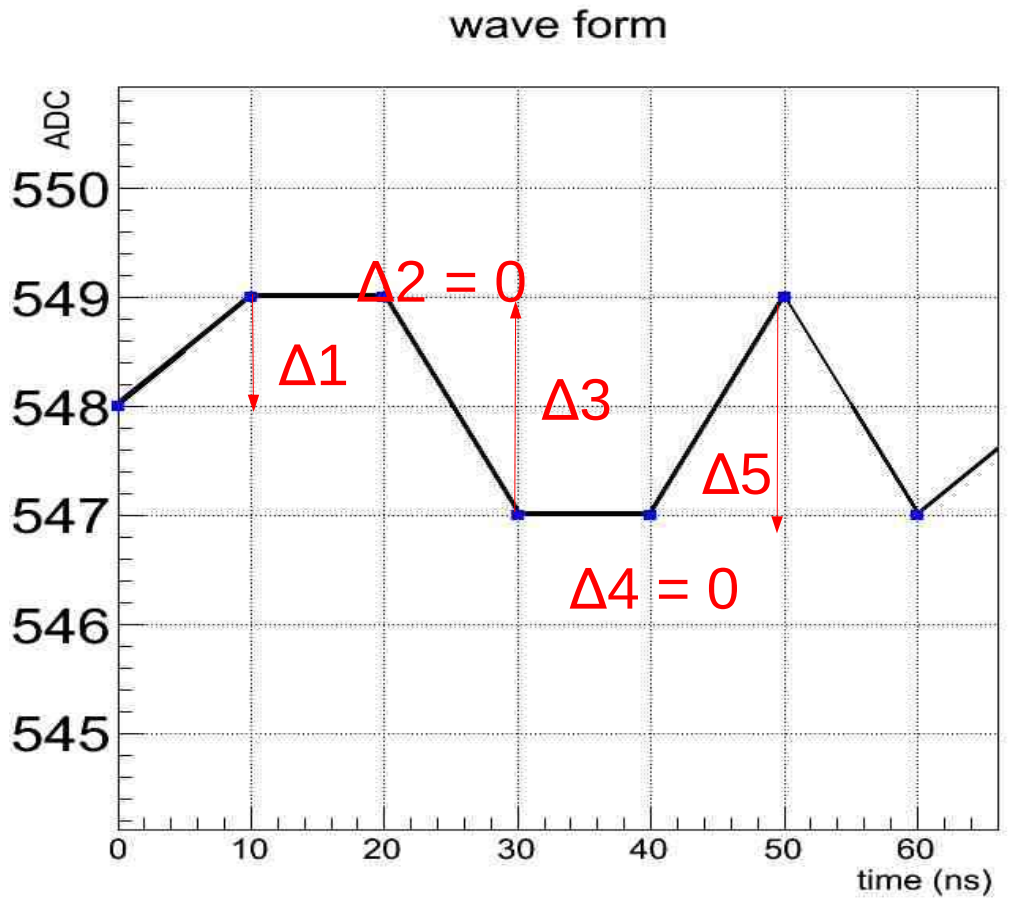
New design for FADC board



Delta compression algorithm

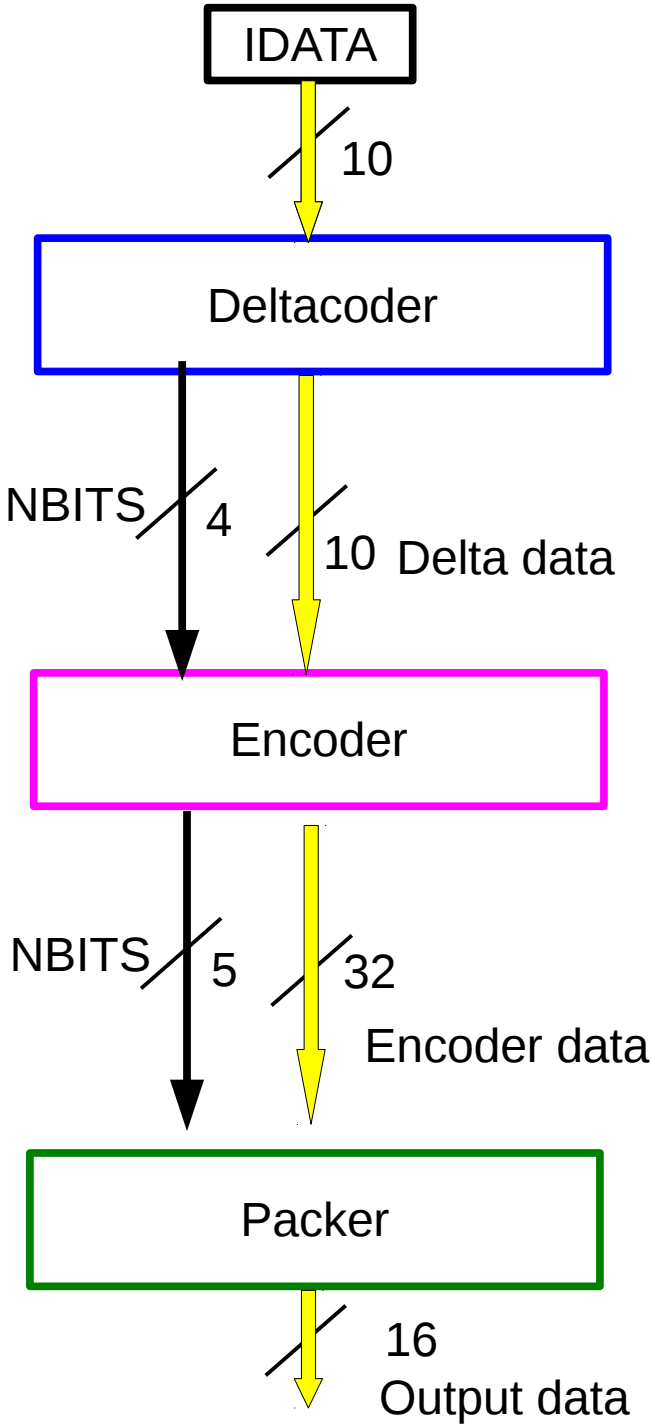
Delta $\Delta_{ADC} = ADC_{n+1} - ADC_n$

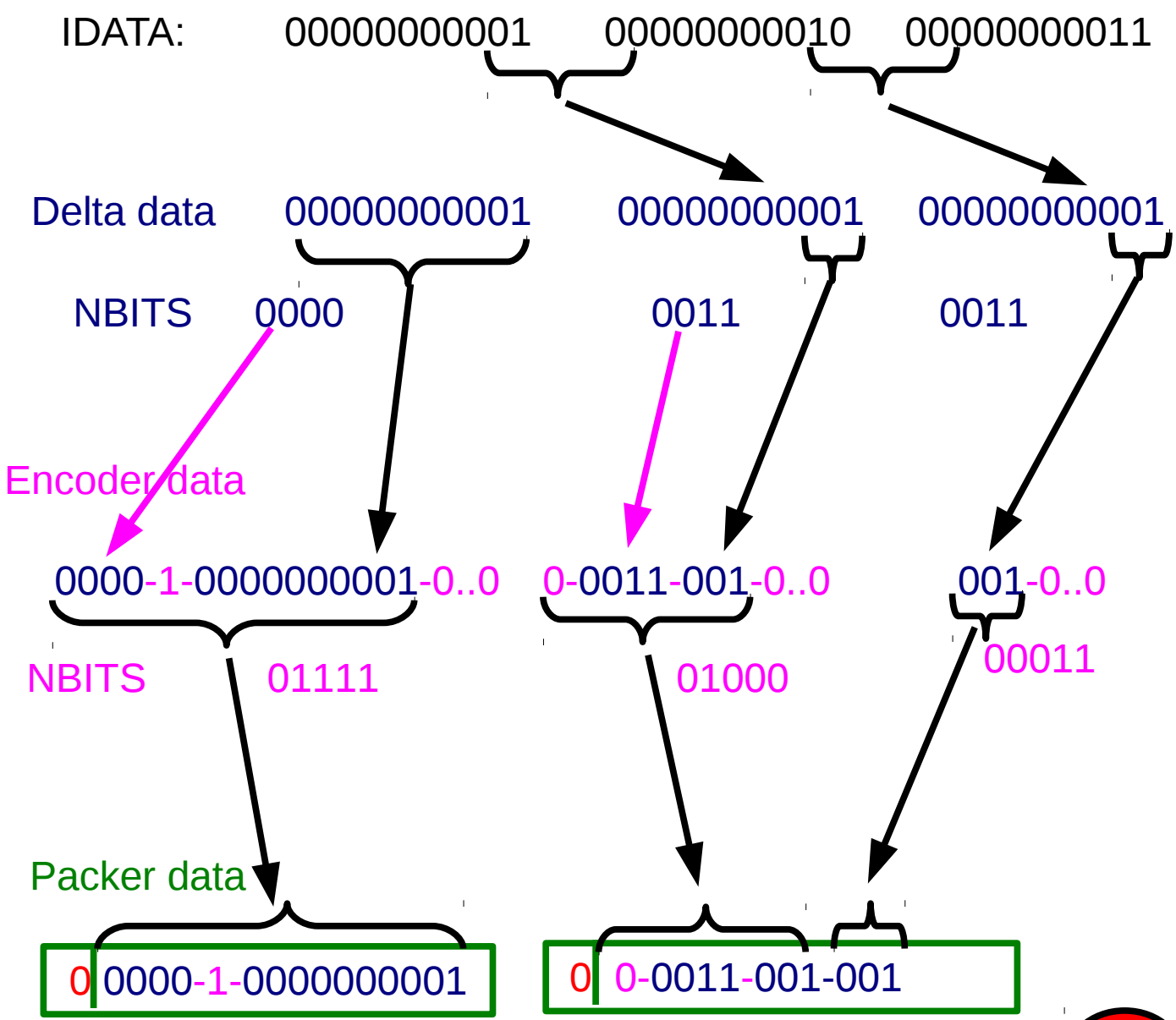
+Calculate delta
+Calculate delta average of some sample point
+Decide how many bits to use for delta code



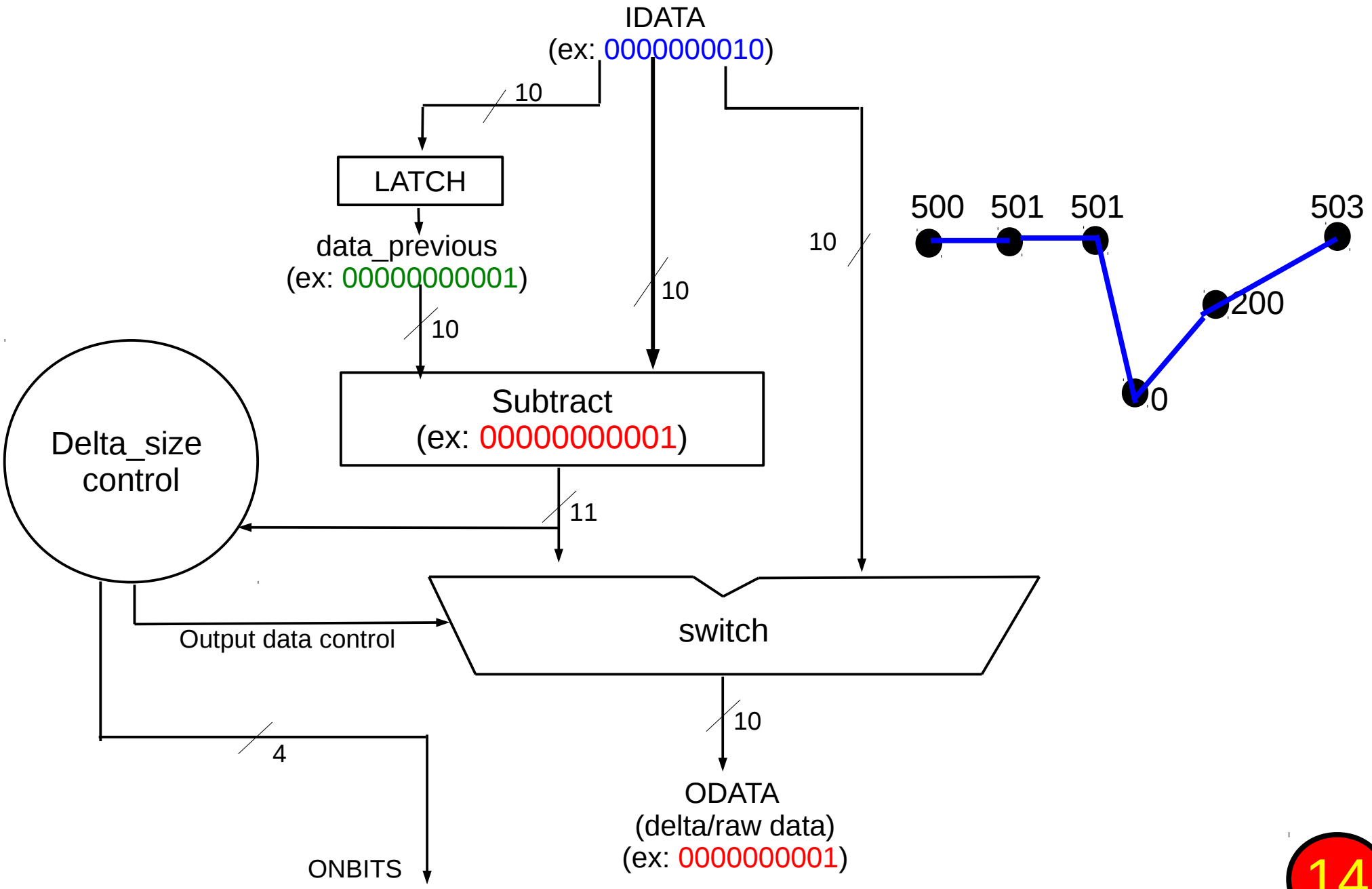
-original data:	548	549	549	547	547	549
-delta compression data:	548	549	549	-2	0	2

Compressor module

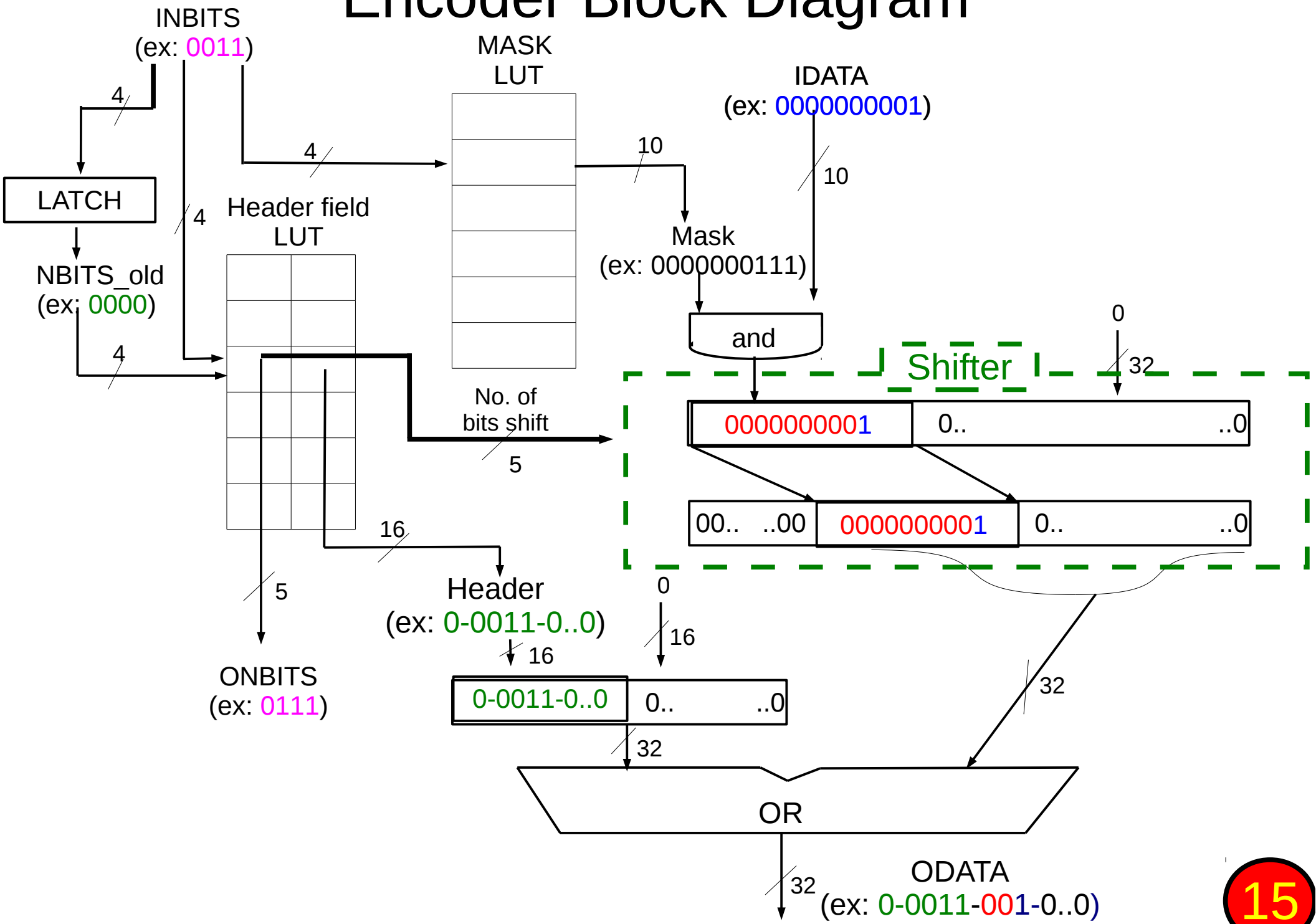




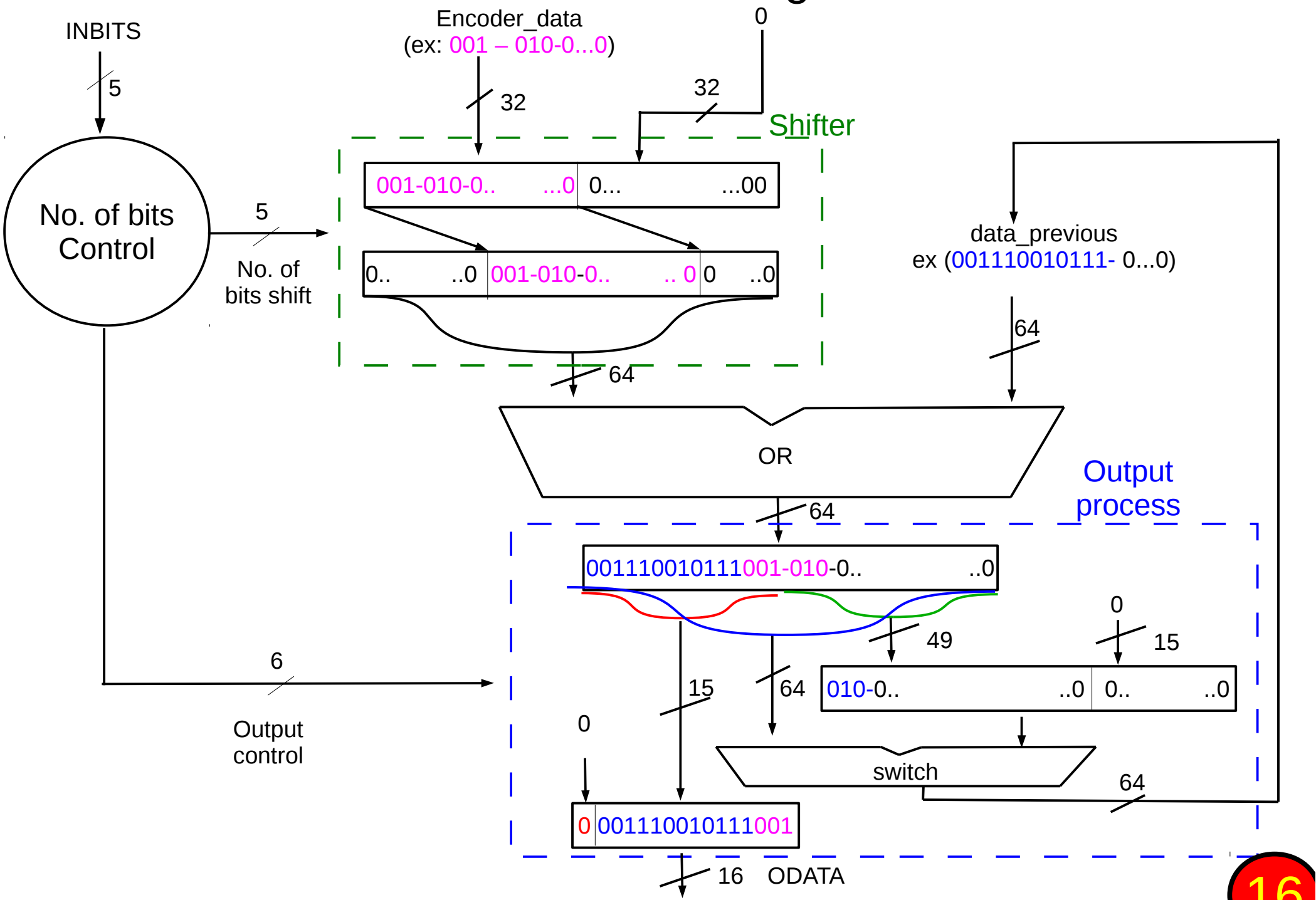
Deltacoder Block Diagram



Encoder Block Diagram



Packer Block Diagram

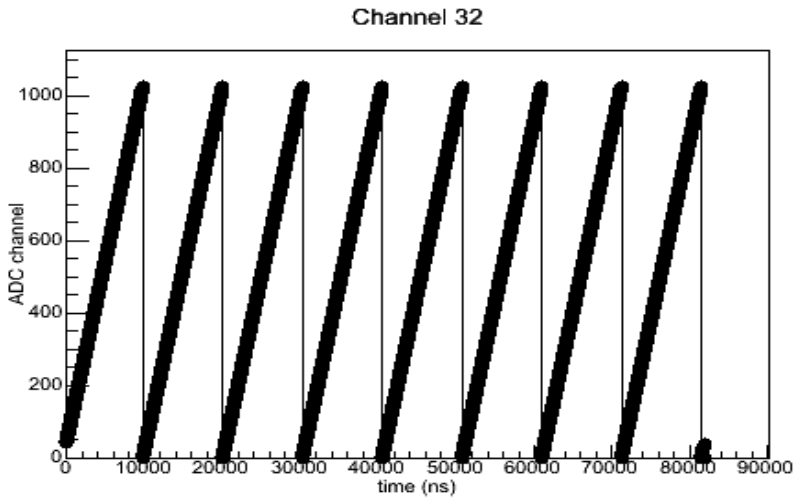


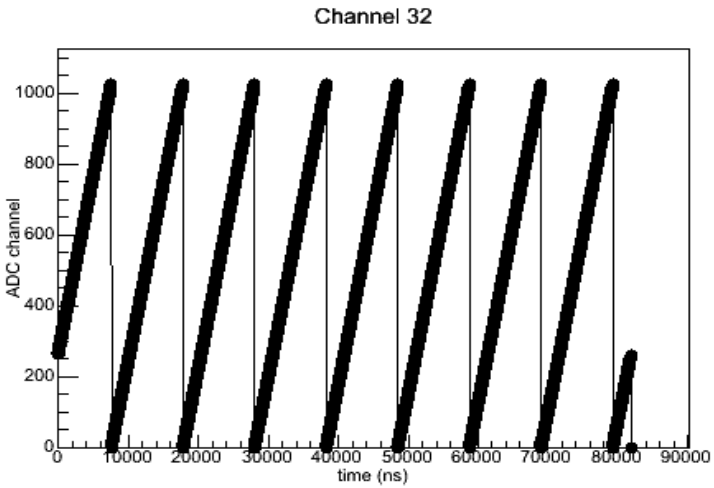
Test list of compression module

- Test with counter signal → no loss any sample points during data transfer
- Square signal and pulse signal → FADC readout board can read signal with different delta size
- Delta size change in fly → FADC readout board still work well with delta size changing and different noise level
- DeeMe estimate signal → how much we can compress

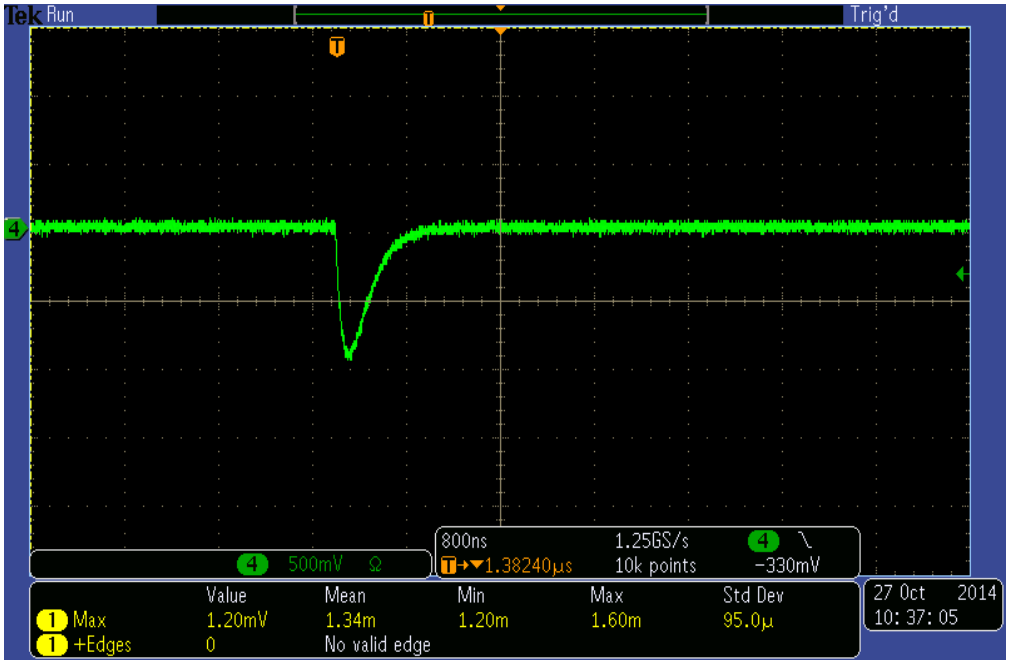
Test list of compression module

- Counter signal (signal will increase or reduce one by one at positive clock)

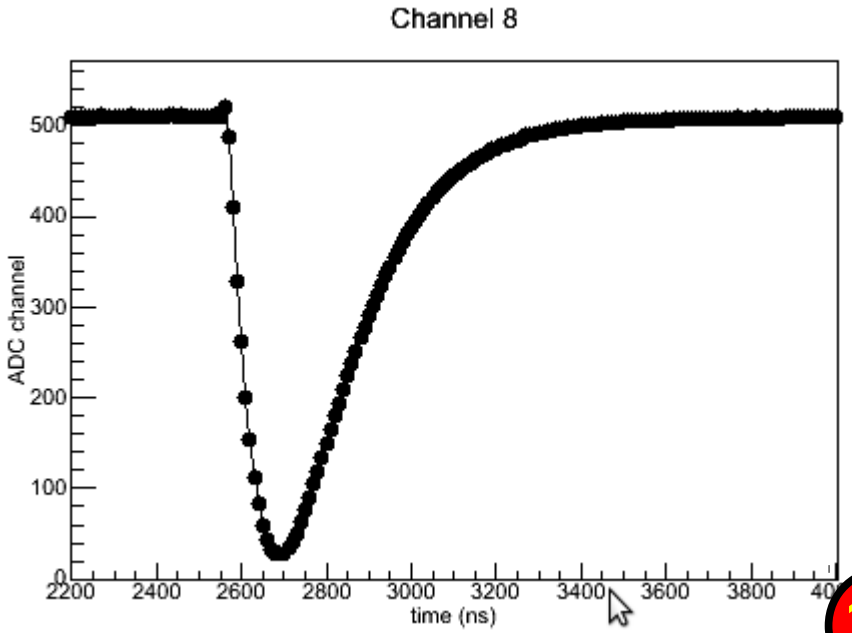




Input signal

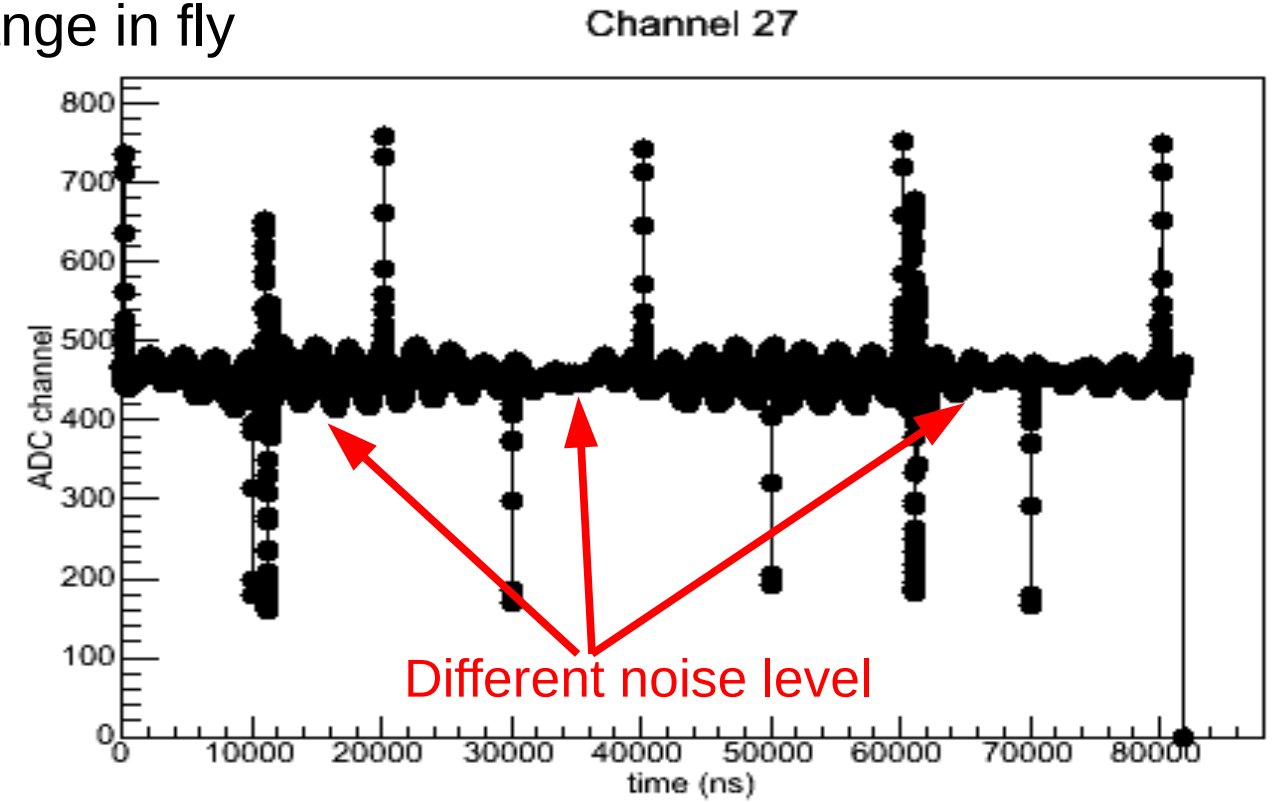


Output from readout board

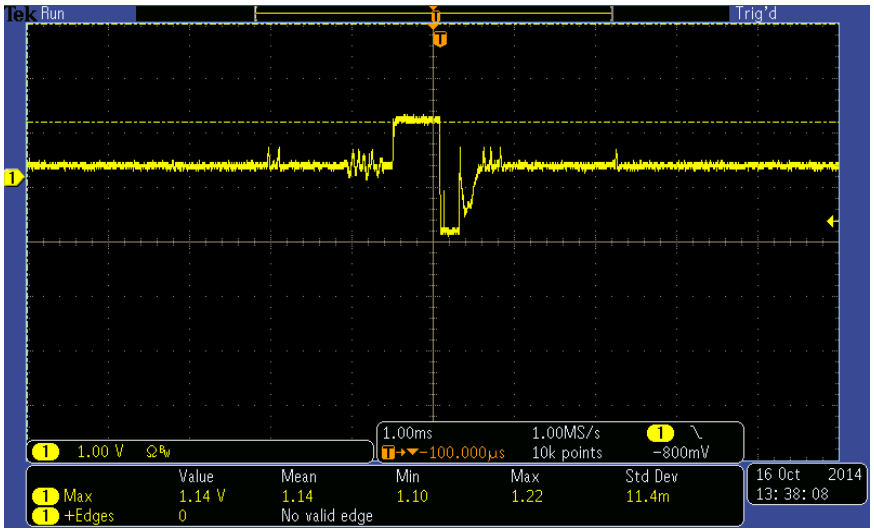


Test compressor module

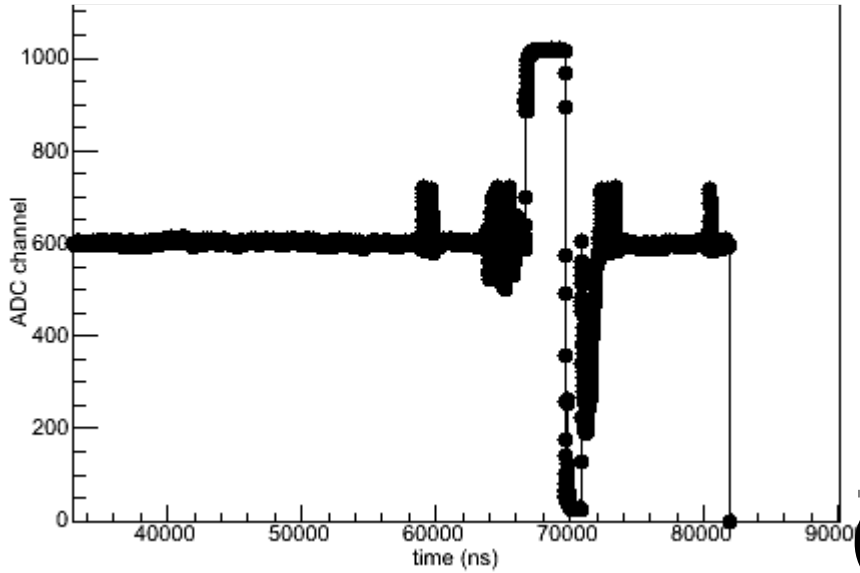
- Delta size change in fly



- DeeMe estimate signal



Compression ratio = 2.9



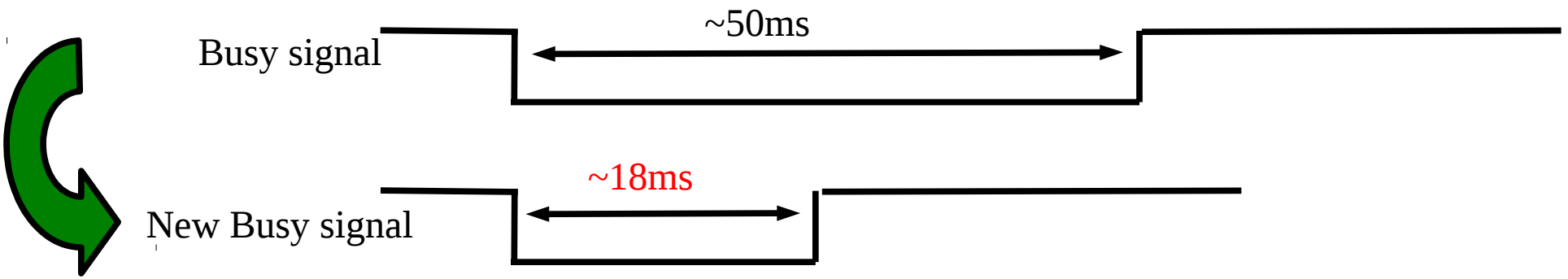
Summary

- We design new firmware for FADC readout board which satisfy for DeeMe project

- Apply delta compression algorithm to compress data

  +Achieved compression ration is 2.9

  + Busy signal with 8192 sample point and 32 channel ~ 18ms



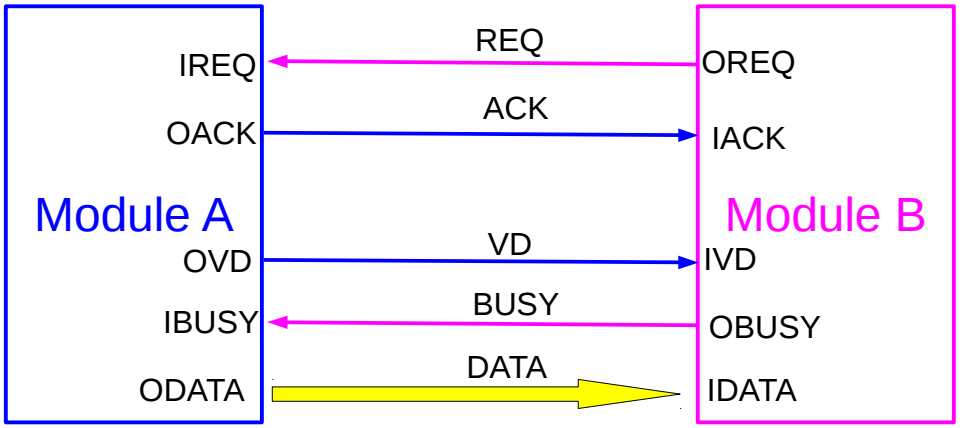
  +Test compressor module with some edge condition and get good result

Thanks for your attention

Backup slides

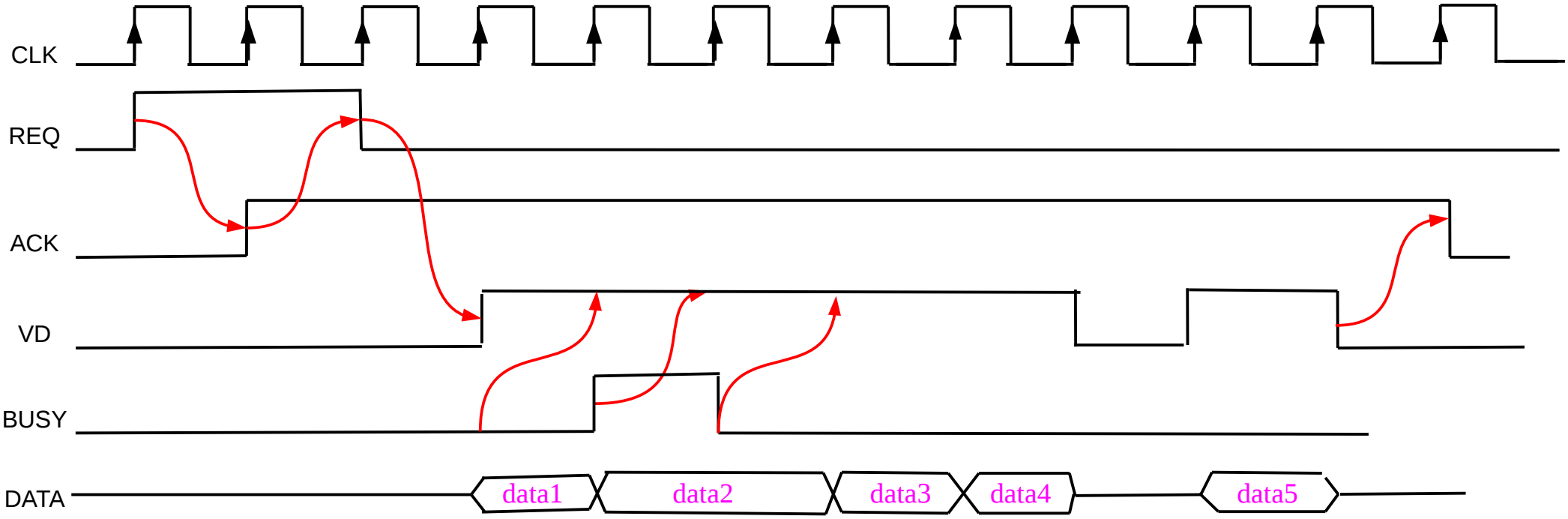
New design for FADC board

Handshake protocol between module A and module B

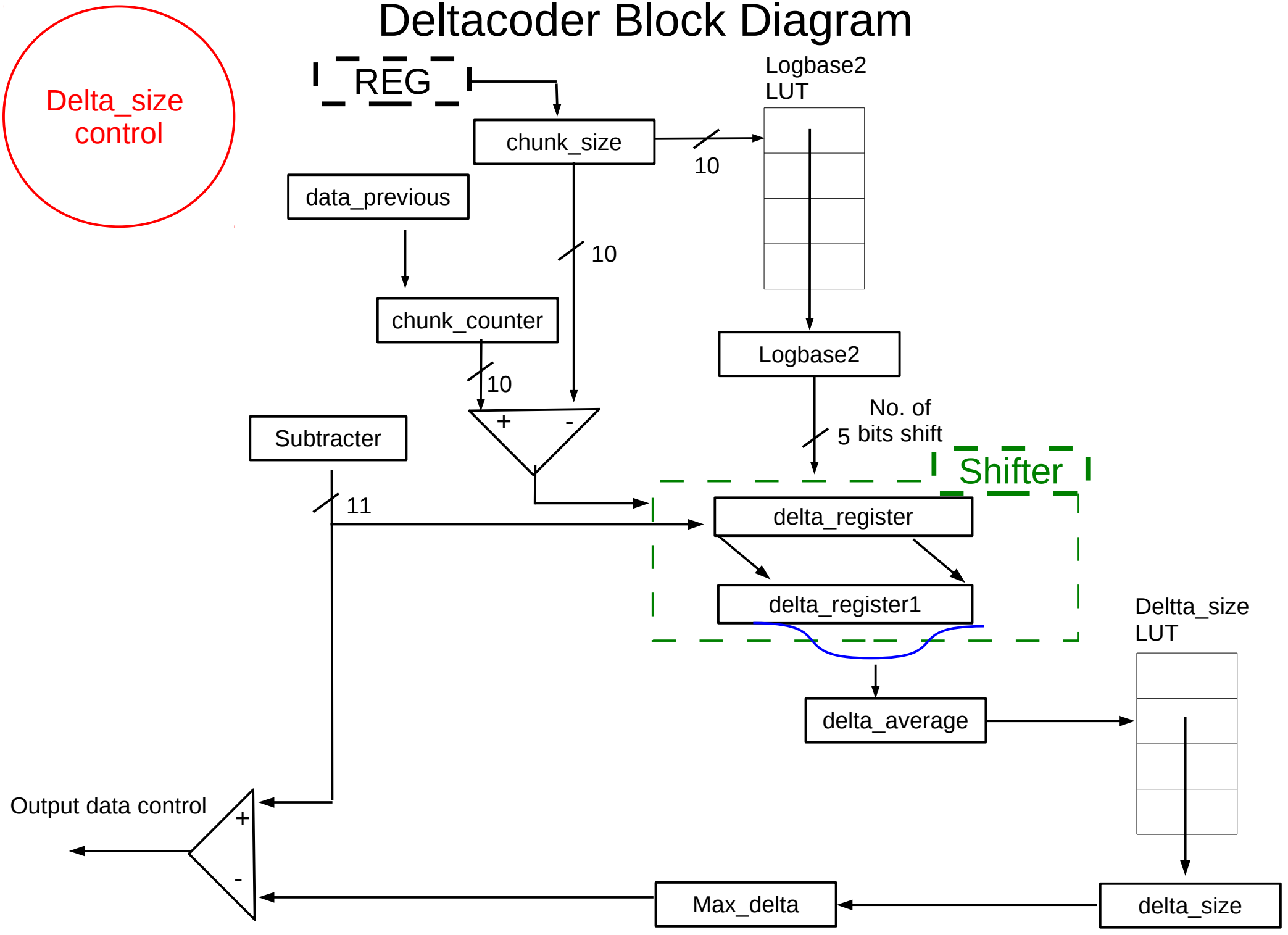


+User can modify their own data processor

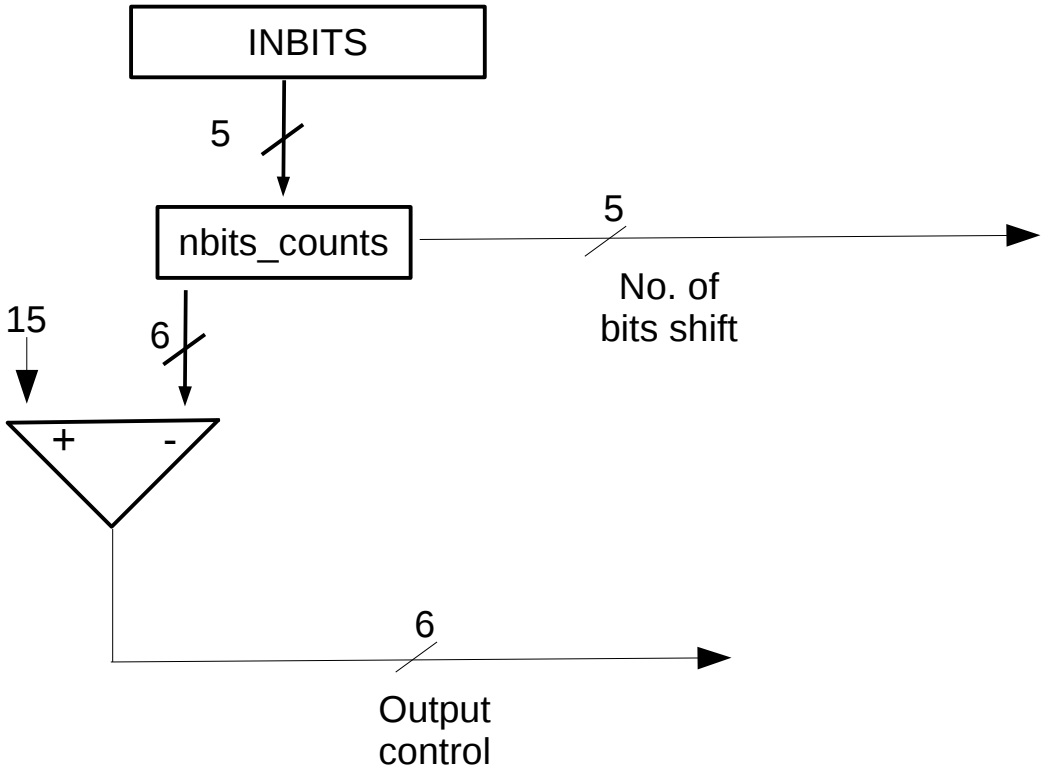
+Transfer data in one clock



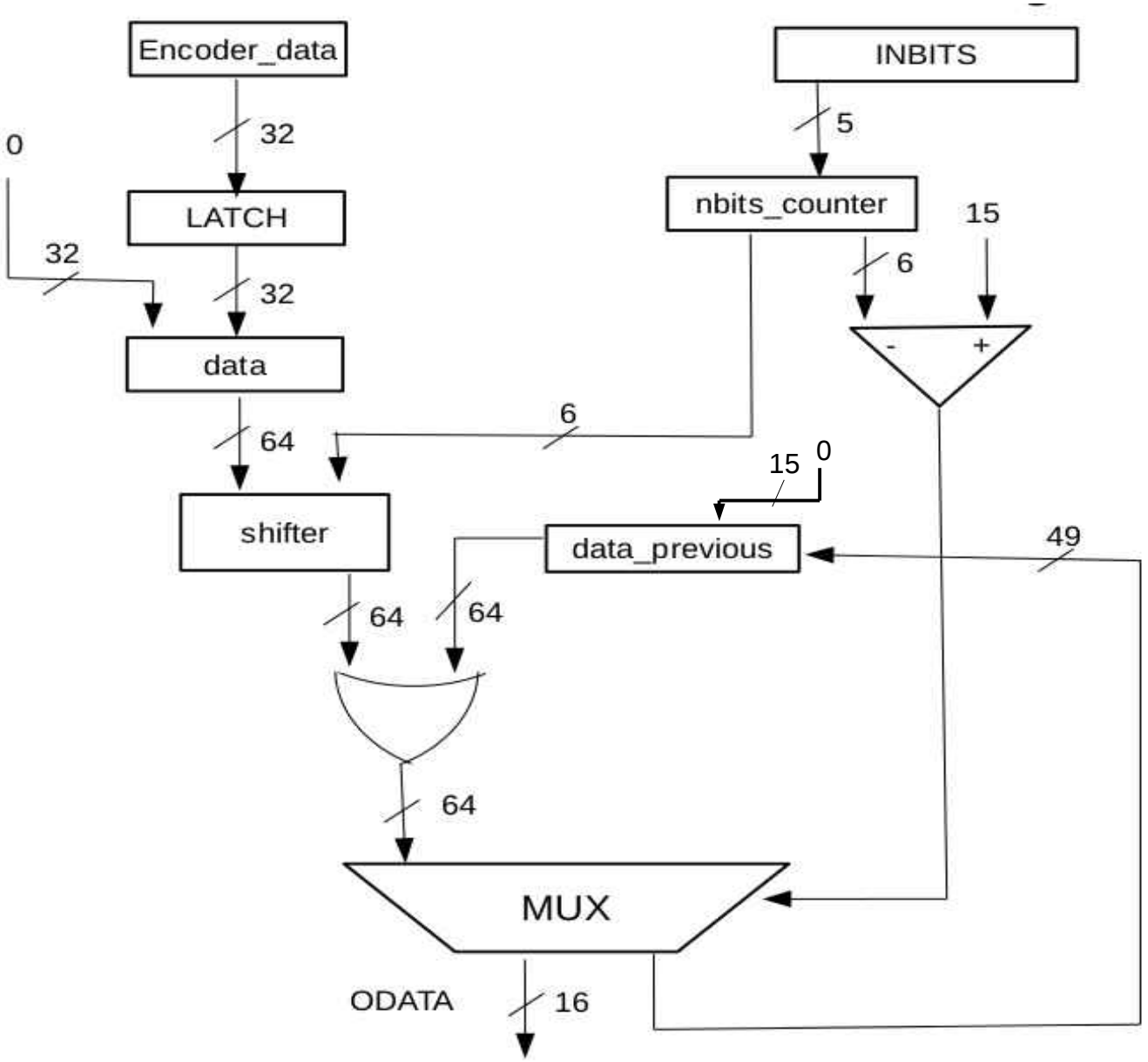
Deltacoder Block Diagram



Packer Block Diagram



Packer block diagram



Overload input signal