中國科學院高能物理研究所
Institute of High Energy Physics Chinese Academy of Sciences

Development of the ABCStar front-end chip for the ATLAS Silicon Strip Upgrade

Weiguo Lu

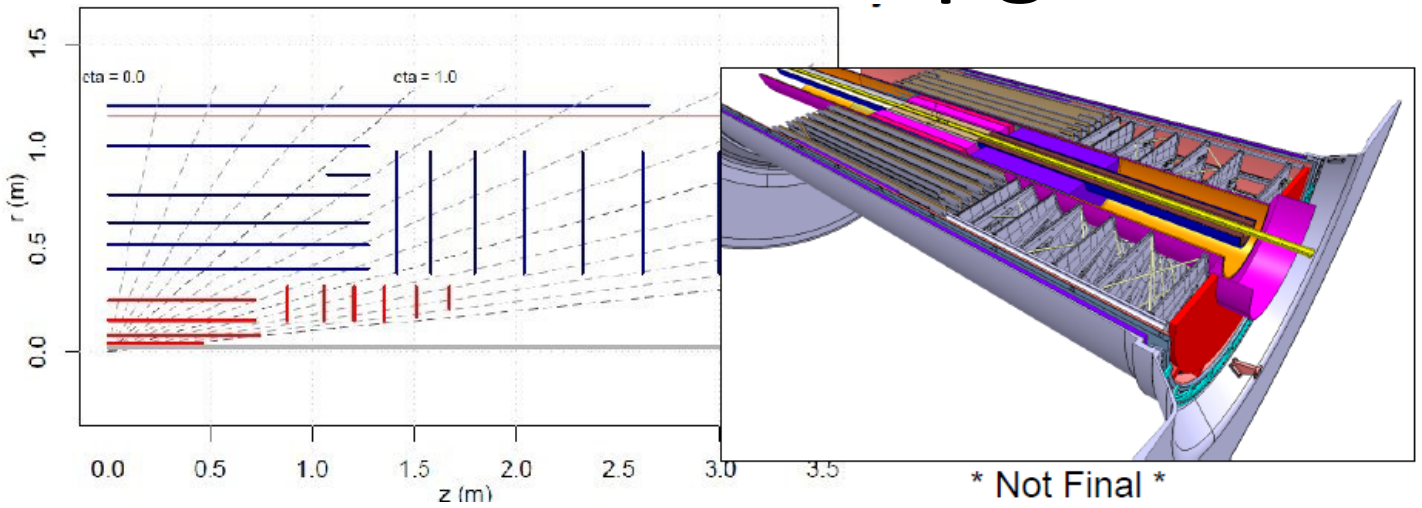
IHEP-CAS

On behalf of the ITk Strip collaboration

preface

- Specifications of ABCStar not fully defined
- Several options under discussion
- Focus on the new design features and current status

ITk Phase II upgrade

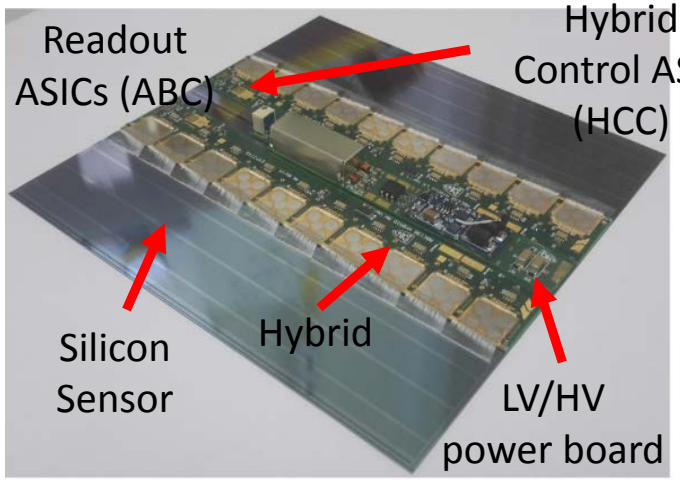


- The Phase II upgrade of the ATLAS Inner Detector is now named the ITk(Inner Tracker)
- It will be an all silicon detector made up of pixel detector and strip detector
- A very large increase in size and complexity for strip
  - A factor of 4 more modules, a factor of 10 more channels

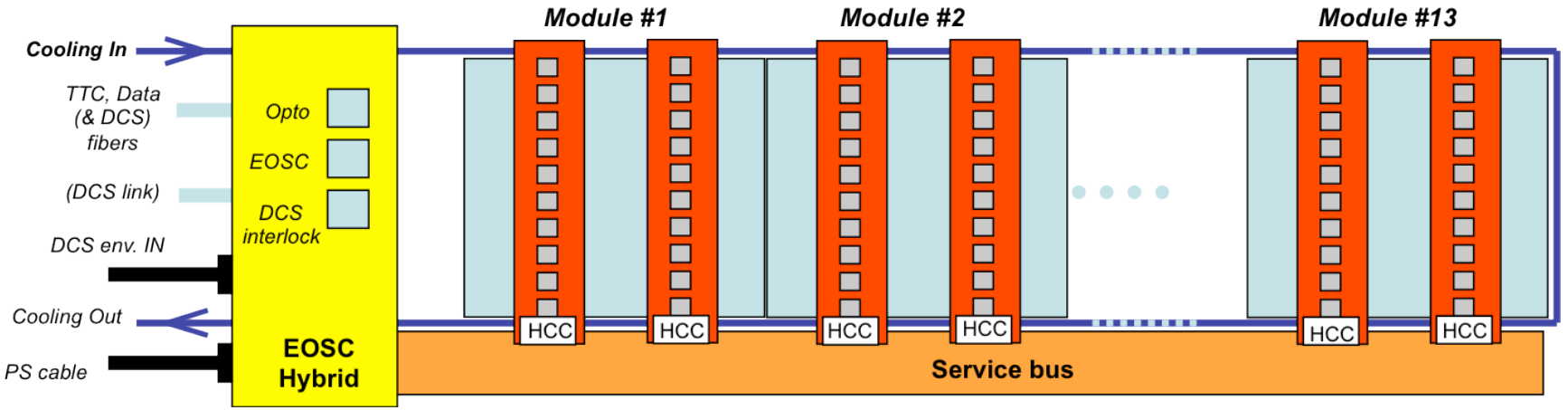
The New ATLAS ITk Strip Tracker						The Existing ATLAS SCT Tracker		
Number of Full Length Double-Sided Staves	Number of Barrel Modules	Number of Double-Sided Petals	Number of End-Cap Modules	Number of Readout Channels		Number of SCT Barrel Modules	Number of SCT End-Cap Modules	Number of Readout Channels
196	10,976	384	6,912	60M		2,112	1,976	6.3M

Stave, Petal, Module concepts

- Barrels are segmented into staves
- End-cap disks are segmented into petals
- Module is the basic building block for stave and petal
  - each of the two readout hybrids contains 10ABCs and one HCC module controller ASIC
  - Sitting between the two hybrids is a power board



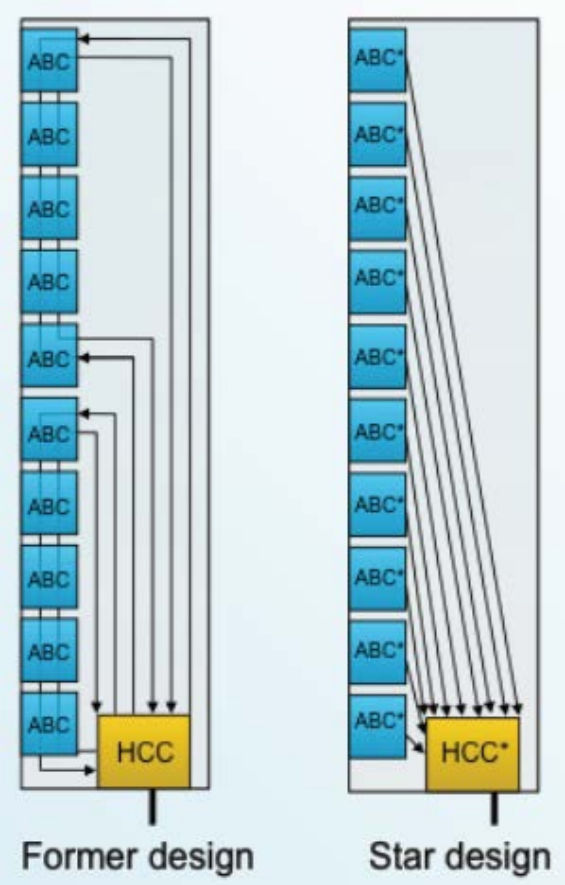
Barrel Module with Short Strips



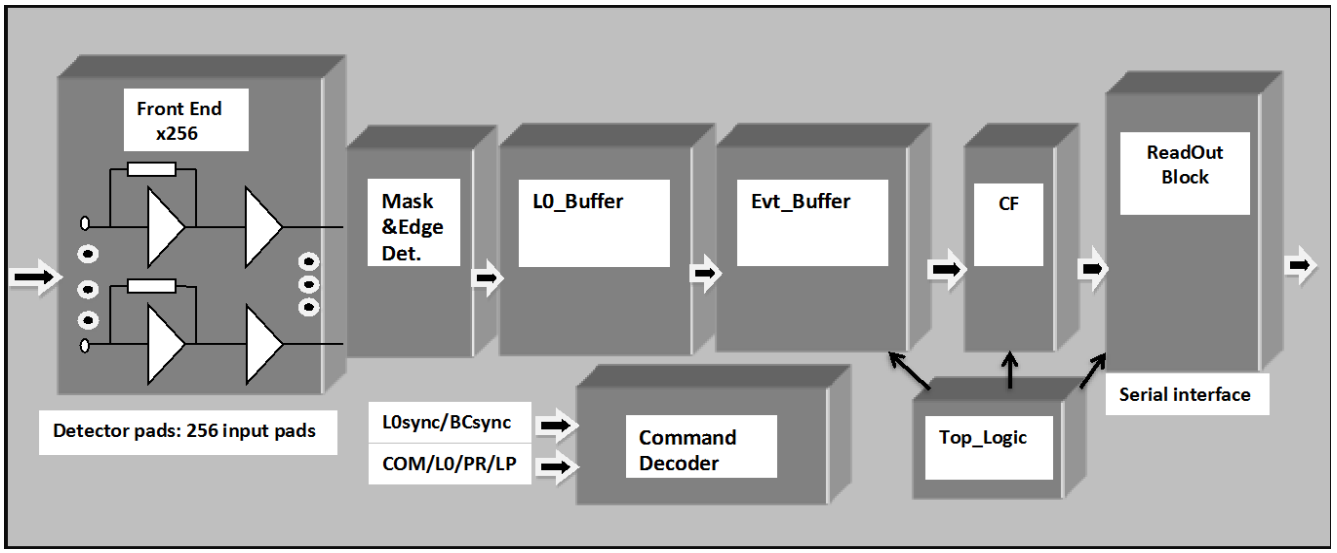
Barrel stave with 13 modules

ABC130 to ABCStar

- From ABC130 to ABCStar
  - Change chip design to meet the requirement of increased trigger rate
  - Interface from ABCs to the HCC: Serial transfer of data to direct communication, to remove the bottleneck of bandwidth
- The ABCStar chip will be the front-end ASIC for the readout of the ITk Silicon Strip detector of Phase II upgrade
  - ABC--ATLAS Binary Chip
  - Star--Star readout with point to point connection

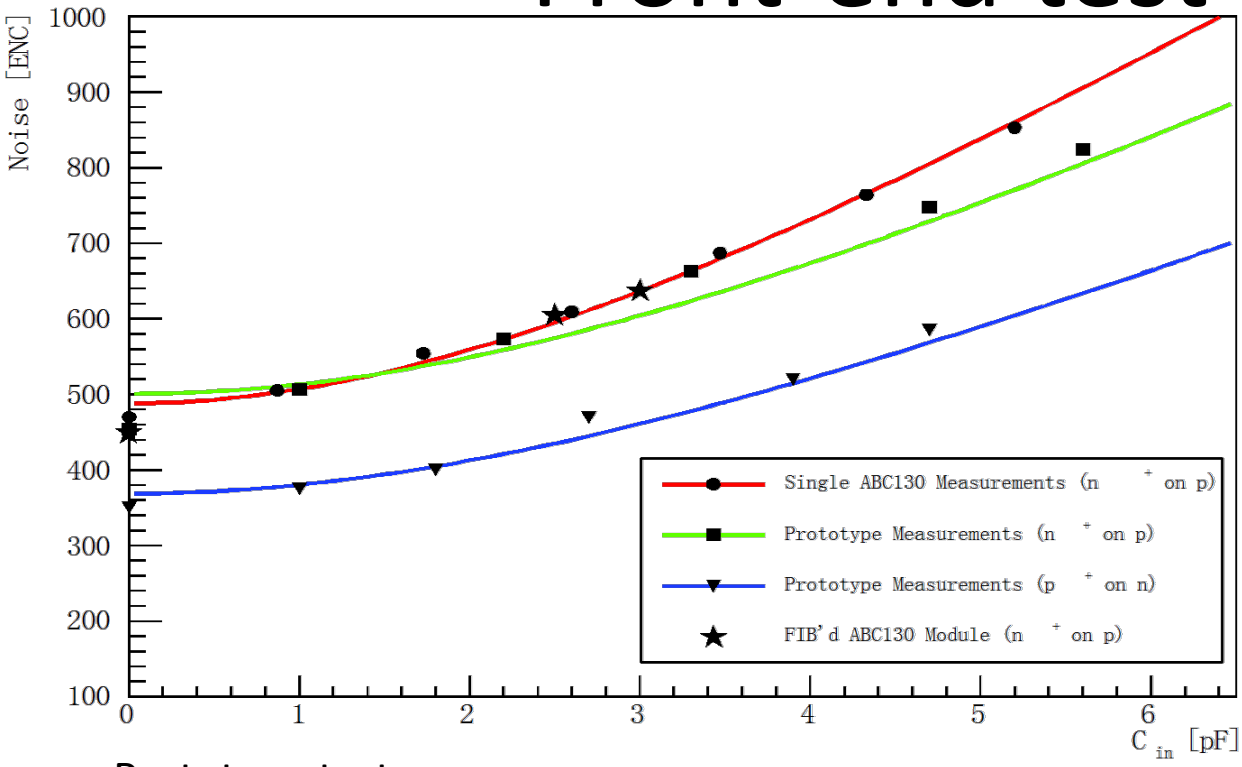


ABCStar ASIC



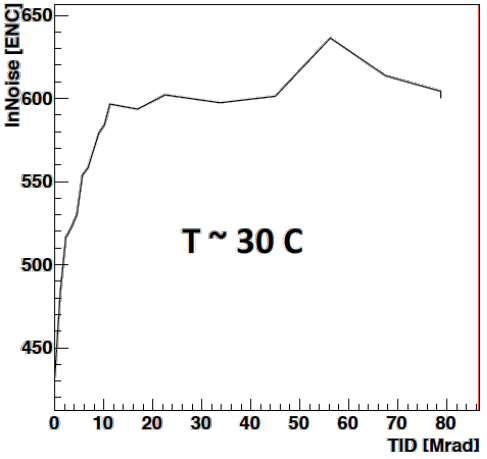
- It uses the standard binary readout employed by the present ATLAS SCT tracker readout
- It Includes amplifier, discriminator, input register block, pipeline, event buffer and a cluster algorithm to compress data for output
- It is being designed to support more than one trigger modes
- It will be built on GF130nm technology

Front-end test



Proto ENC for negative signal

Proto ENC for positive signal

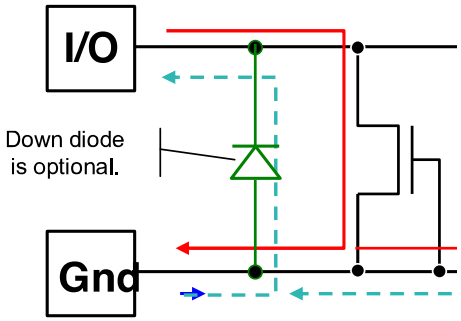


- Prototype test
  - The ABCStar is still in design. The ABC130 is being used for testing
  - ENC for the module(FIB repaired)agrees well with the prototype front-end chip and the single ABC130 chip
  - But the ENC is higher than simulation, and we also observed the gain discrepancy between estimates and measurements
  - Excess noise after irradiation was also observed.

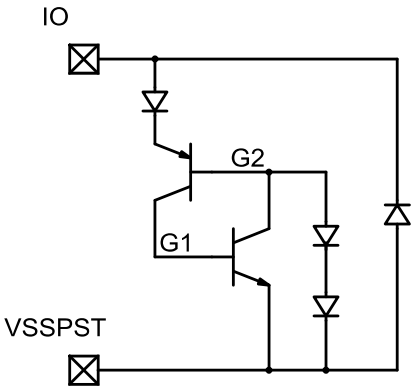
Front-end redesign

- The original thought is to reuse the front-end for ABCStar
- New specs of sensor
  - Lower signal after irradiation
  - Signal polarity change
- Improvements
  - Preamp Change to resistive feedback
  - Critical NMOS devices in enclosed geometry
  - Optimization for the measured sensor parameters after irradiation
  - New biasing scheme for preamp and shaper

ESD protection



Protection used :
Non silicide NMOS
Cpar ~1pF
HBM 1.5KV
Vthreshold/Hold ~4V
Ron 5 ohms

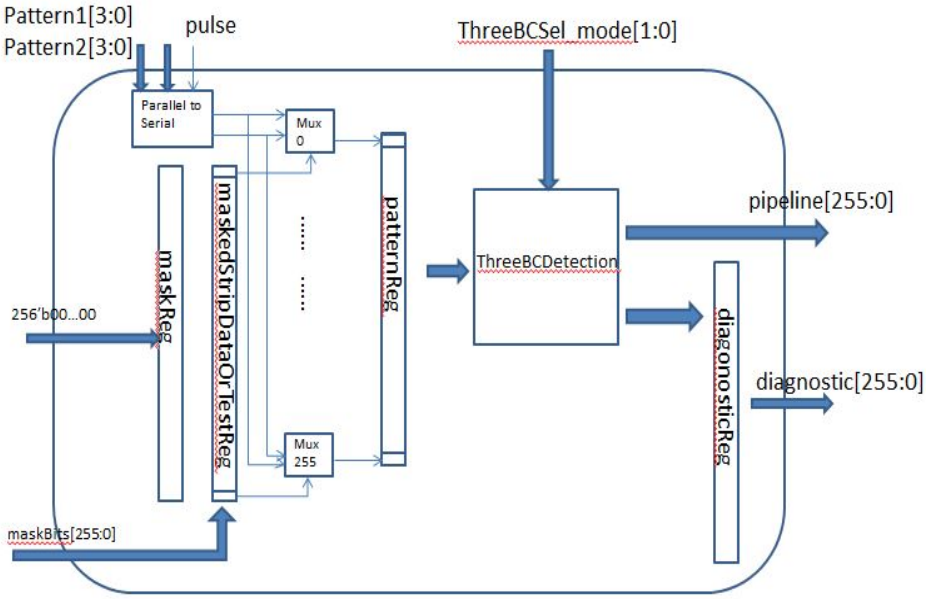


Protection proposed :
SCR (thyristor) device
Cpar ~0.1pF
HBM 1KV
Vthreshold ~3.15V,
Hold voltage 2.4V
Ron < 1 ohm

- Observed multi-channels ESD failure in case of detector breakdown
- A company has demonstrated good ESD structures and model capability, that add low capacitance and better switching characteristics
- For FrontEnd and multi-drop I/Os

Input Registers block

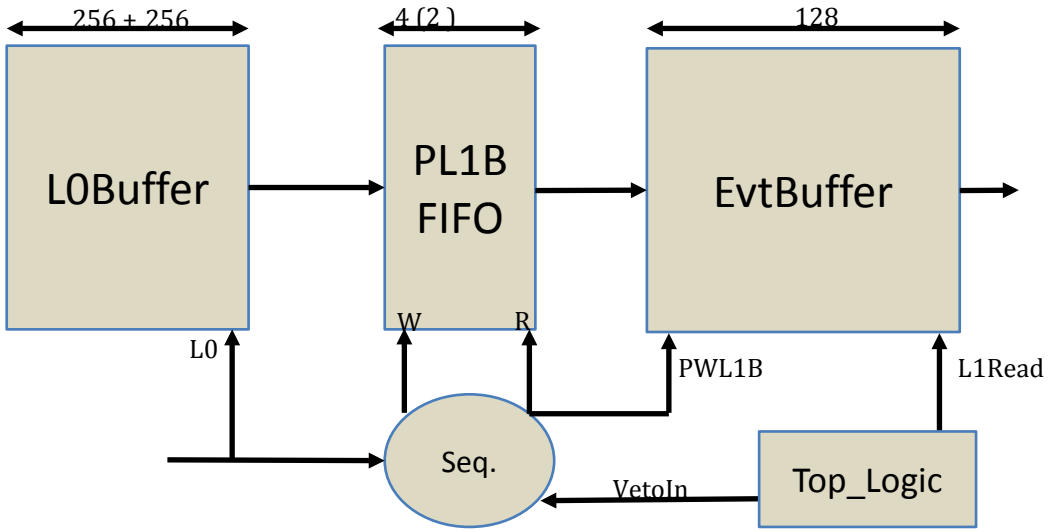
- Input register latches data with BC clock
- Mask/test registers for dual purpose
- Static and pulse test modes
- Edge detection circuit with different selection criteria
  - detect the input signal looking on 3 consecutive BC clock periods
  - only a single '1' is written into the pipeline



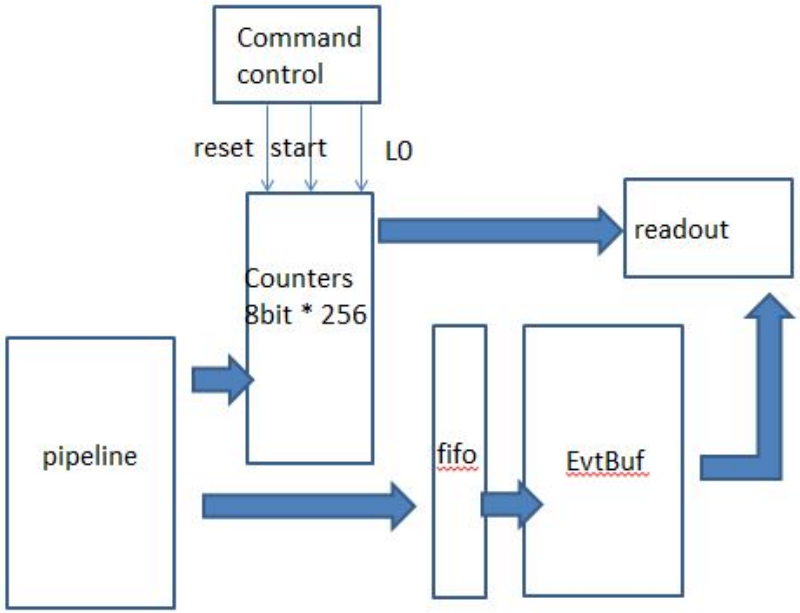
Det_mode (1:0)	Name of Selection Criteria	Hit Pattern (Oldest data bit 1st on the left)	Usage
00	Hit	1XX or X1X or XX1	Detector alignment
01	Level	X1X	Normal Data Taking
10	Edge	01X	Normal Data Taking
11	Clear	None	Special Mode

Two stage buffers

- The two stage buffers inherit the structure of ABC130
- The pipeline and EvtBuffer are all realized with single port RAM block
- Pipeline(L0Buffer)extended to 512bit length
- EvtBuffer reduced to 128bit length(128 events)
- Transfer 1 event per L0 from Pipeline to EvtBuffer(instead of 3)
- Intermediate fifo to give the priority to EvtBuffer read operation, in case of consecutive L0s

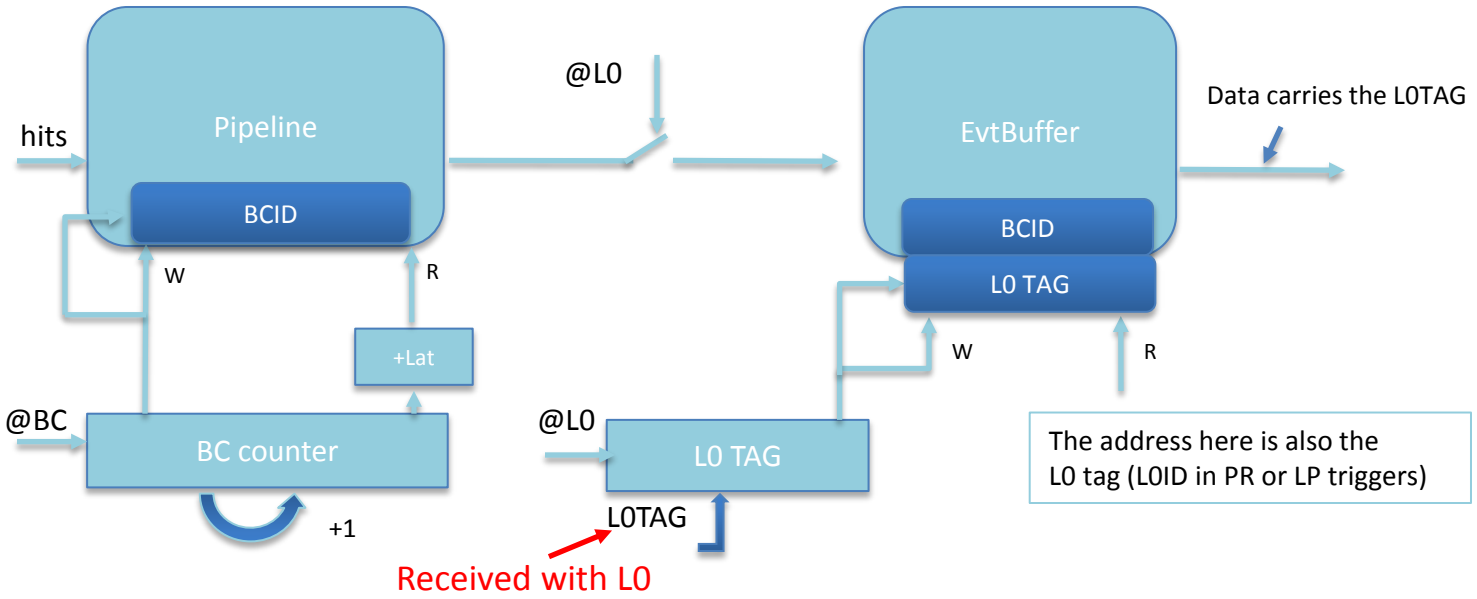


Hits Accumulators



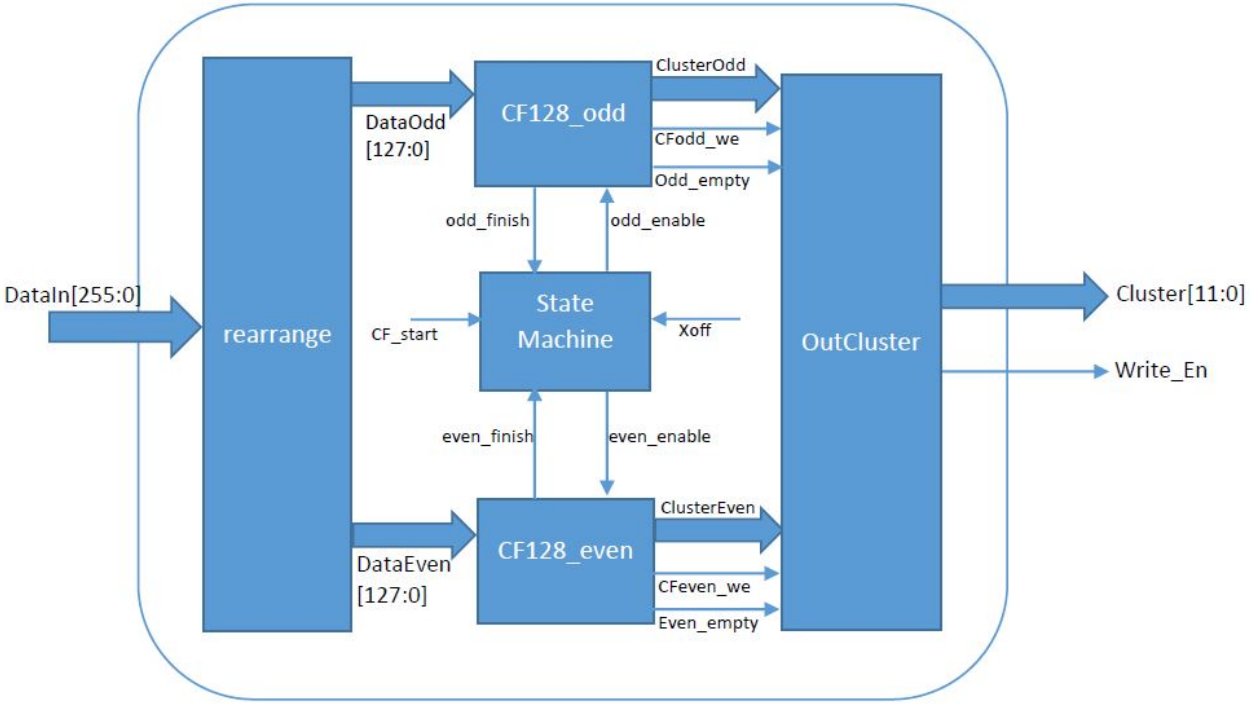
- A feature that can be useful in some calibration procedures
- With an 8 bit counter attached to each channel at the output of the pipeline
- It accumulates hits counts moving from the pipeline to the EvtBuffer at L0
- All the counters can be disabled and reset by the command control
- Counter values are readout through a regular read command

L0 tag insertion



- The internal counter for event identification is sensitive, once in error, all following events appear as out of order until the counter is reset.
- We will employ new scheme--sending a L0tag with L0
- At present, L0tag is a 7bits number provided for each L0, that is used internally as the "physical" address of the event in the EventBuffer
- To correctly address the event to readout, the L0ID value in PR and LP triggers has to be equal to the L0_tag value that has been written at L0 time
- This helps to improve the reliability

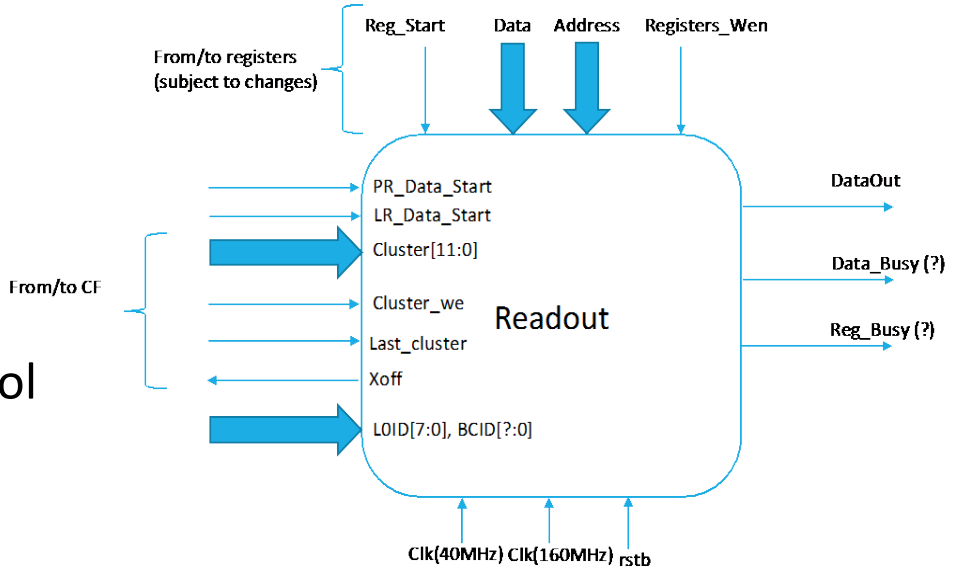
Cluster finder

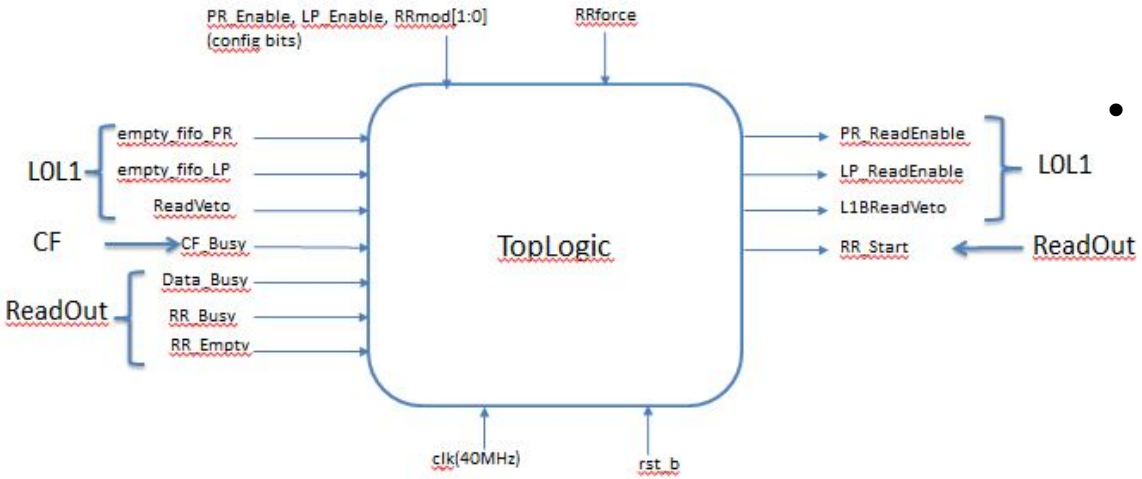


- The cluster finder takes in 256 bits of strip data and reports out 12 bit clusters at 40MHz
- The rearrange block separates the input data into 2 banks of 128bit called odd and even banks, to group together bits of adjacent channels in a strip row of the detector
- Each cluster finder sub-block treats 128 strips of the same row
- They are enabled in turn by a state machine. The 2 banks report out clusters one bank at a time in an alternate fashion to avoid bias to any 1 bank

Readout and TopLogic

- TopLogic
  - Sequencer for the control of EvtBuffer, ClusterFinder and ReadOut
  - Register readback control
  - PR/LP readout priority control





- Readout
  - Responsible for building data packets with the hit patterns coming from the Cluster Finder blocks or from the registers to be read
  - A controller defines the order in which packets are formatted
  - Each packet is transmitted to the fast serializer

Data formats and rates

- In ABCStar the output format is a variable length readout packet with maximum length of 68 bits
- The estimated average event size per chip is 2 to 3 clusters (therefore one 56 bits packet)
- The average data rate at the output of one ABCStar chip is 56 Mb/s
- 160Mb/s readout rate was rather chosen to reduce the transmission latency for L1-track

- Packet framing

Start Bits	Header	Payload	Trailer
3	16	48	1

- Physics data header and payload

TYP	L0ID	BCID
4	8	4

Last Cluster Flag	Cluster Address	Next Strips Pattern
1	8	3

X 4 (max)

- Register readback packet header and payload

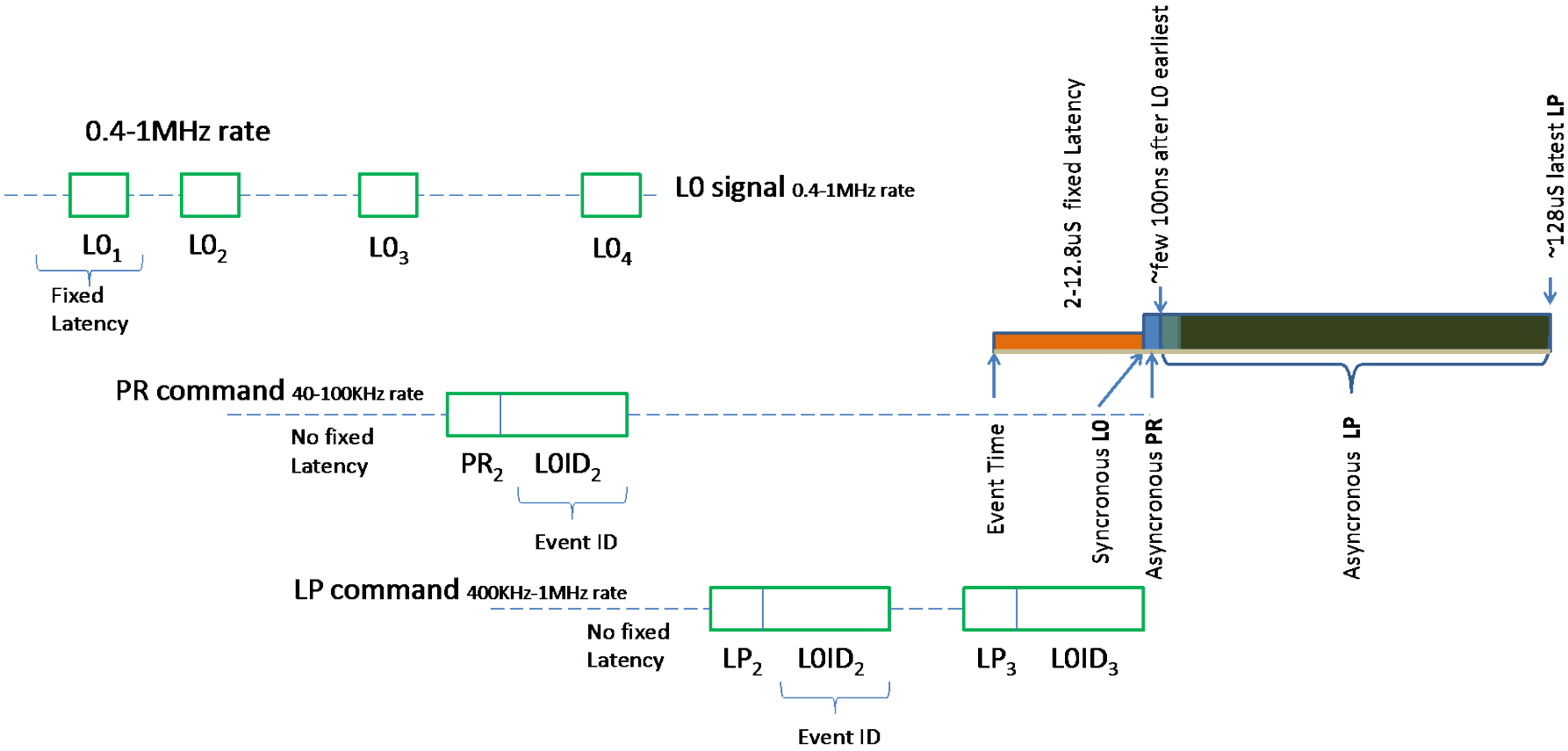
TYP	Register Address	TBD
4	8	4

Register Contents	Monitor Data
32	16

Clock and control signals

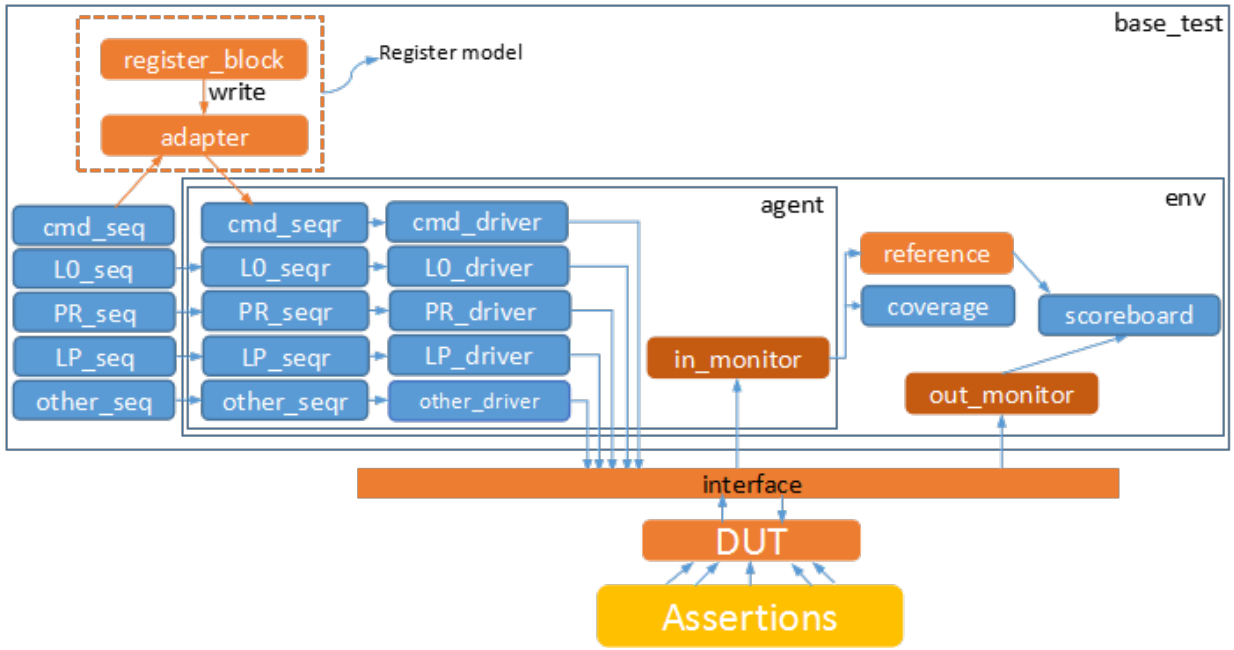
Name	Type	Description
RCLK	SLVS	160MHz Clock input primarily intended for Data Readout
BC	SLVS	Beam Crossing Clock at 40MHz
L0_CMD/LCB 1)	SLVS	80Mb/s, L0 Synchronous Trigger, CMD bits (legacy mode of ABC130)
L0_CMD/LCB 2)	SLVS	160Mb/s, L0 Synchronous Trigger, L0_tag, CMD and BCR bits (6b/8b encoded frame, extending over 4BC)
LP_PR	SLVS	Physical line that receives the LP and PR triggers muxed at 80Mb/s

Trigger rates and latencies



- The sequence of the internal L0, PR, LP signals used to trigger the physics data readout, and the latencies.
- The original trigger scheme is 1MHz L0, 100KHz of PR, and 400KHz LP.
- At present, different trigger models are still under discussion.

UVM setup



- A top verification setup based on (UVM)Universal Verification Methodology was built for the current design of ABCStar.
- Its capabilities
  - Functional coverage with customized random stimulus
  - Result comparison with reference model through scoreboard
  - SystemVerilog assertions for validating key design features

Different trigger modes

- The UVM setup is used to verify the current design under several possible trigger conditions
- With the different rate, latency and distribution model of triggers, exponential for L0 and poisson for LP and PR, the current design all works well
- This may be useful for the discussion of various readout scenarios

L0	Lat.	LP	Lat.	PR	Lat.	Status
1MHz Exponential	6us	400KHz Poisson	12us	100KHz Poisson	12us	OK
1MHz Exponential	6us	1MHz Poisson	12us	No PR		OK
4MHz Exponential	6us	1MHz Poisson	12us	No PR		OK
4MHz Exponential	6us	1MHz Poisson	12us	100KHz Poisson	1.2us	OK
4MHz Exponential	6us	600KHz Poisson	12us	400KHz Poisson	100ns	OK

SEU mitigation

- Triplicated configurations registers with bit flip auto correction
- Triplicated FIFO controls and some critical logics
- Protected state machines by hamming code distance
- The L0_Tag schema should provide a protection by preventing the propagation of SEU errors to multiple events

Power options

- A large current increase in the digital part after irradiation was observed during the test of ABC130
- In order to reduce the increased power consumption due to this TID current bump
- We are considering to switch off the power of half of the pipeline in case of low L0 latency
  - A switch on the regular regulated power
  - A Voltage regulator dedicated to the half Pipeline
- Another method is to extend the digital voltage regulator regulation range from [1.15 – 1.25] volts to [1.00 – 1.25] volts
  - using a lower voltage down to 1 volt for the digital part to reduce the power

Options under consideration

- eFuse for chip identification
  - Adding an individual chip identifier programmed with eFuses (eFuses have to be burnt at the wafer test step) ?
- Analogue monitor of voltage and temperature
  - Adding an analogue monitor circuit like in the HCC to measure regulated VDDs, and temperature ?
- Shunt devices removal
  - Removing the shunt device ?

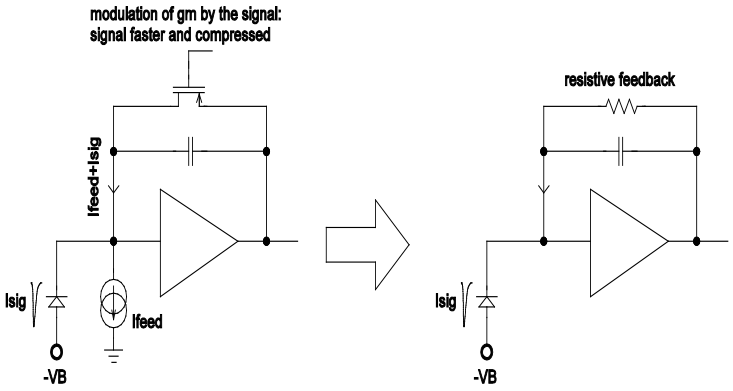
Summary

- Until now, many new features are adopted for ABCStar design
- And verification setup based on UVM has validated the function of present design
- However, some decisions need to be made soon, several issues to be solved
- At present, design and verification are still ongoing, many efforts needed

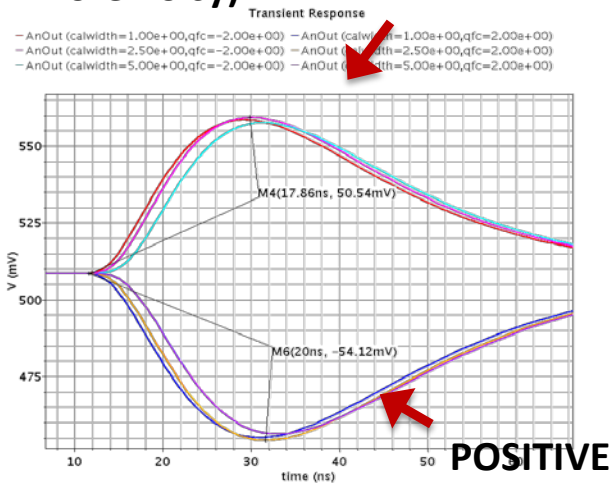
Backup Slides

Performance with different signal polarity

- Noise performance is worse after sensor polarity swap: effect of signal compression.



NEGATIVE (faster, lower in amplitude, more noisy)



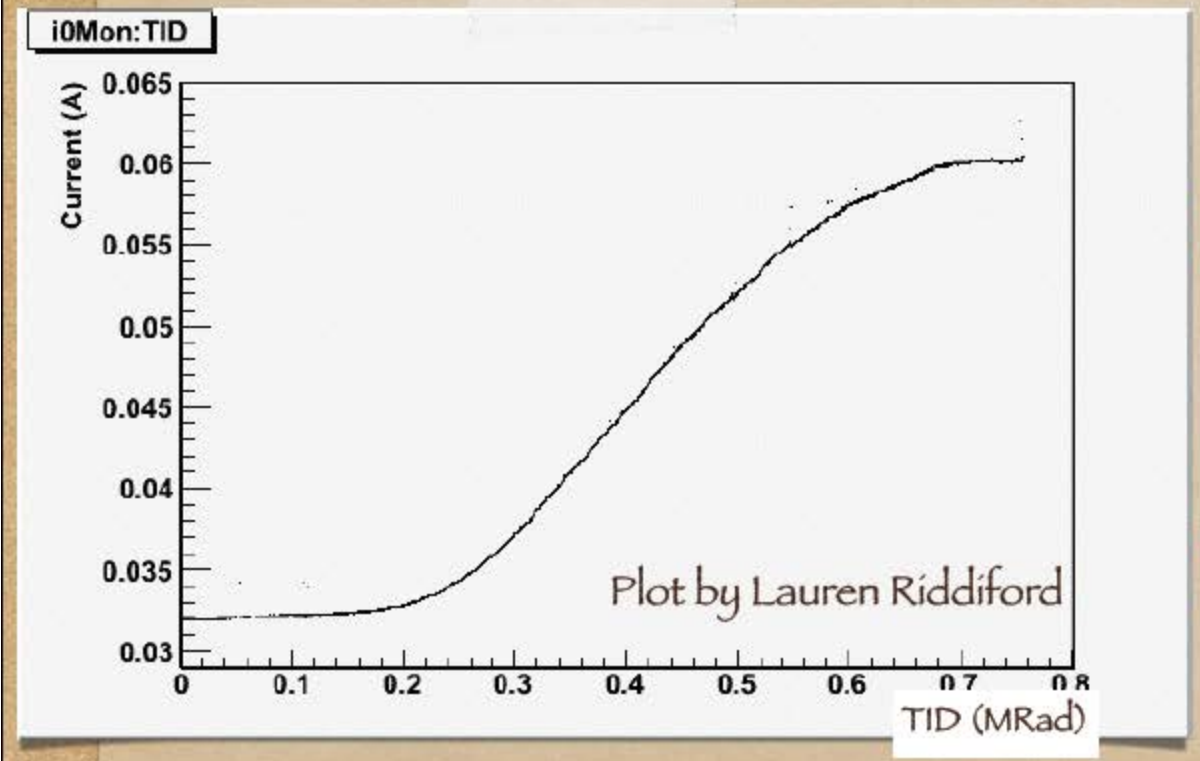
POSITIVE

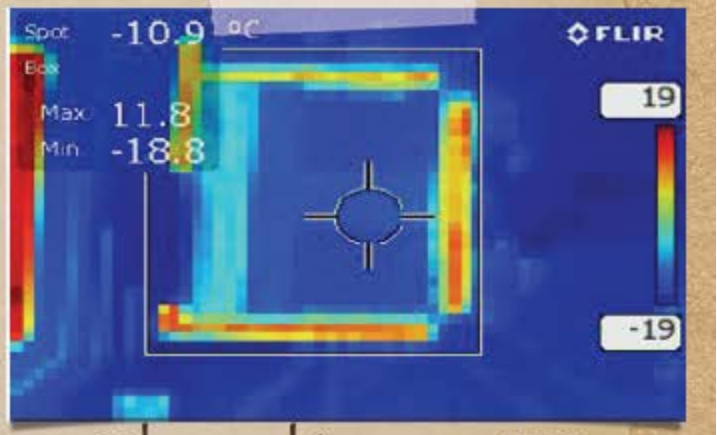
- Effect of compression for negative signals (modulation of feedback transistor gm) simulated at the 5-8% level, in reality (prototype measurements) as high as 20%.
- This can be resolved by changing to a resistive feedback.

ABC 130 TID Update

Status on June 27, 2016

- Running @ Co-60 source 10 TBq at CERN, T≃ -10 C, 2.3 kRad/h
- Factor **1.9** (predicted ~2) increase in current from 32 mA to ~ 60.5 mA





Thermal image: ~10C

LCB Protocol

- LCB contains L0A, Command and BCR information
- LCB is 6b8b encoded and sent at 160 Mbps
  - Six bits of payload are encoded into eight bits of output
  - Four control codes: K0, K1, K2, K3 (8-bit patterns that have no 6 bit decodings)
  - DC balanced, detect single bit errors, longest run of identical bits = 6
- Frames are pairs of 8b symbols
  - 100 ns transmission time per frame
  - 12-bit payload per frame
- L0/BCR Frame
  - 4 bits of L0 status for 4 BC
  - 7 bits of tag of the first L0A – subsequent L0A have incrementing tags
  - 1 bit of BCR – Frames are aligned with LHC orbit so BCR is always in first BC

LCB (2)

- Command Frame Sequence
  - Sent when no L0 or BCR in frame
  - First frame is K2 + (ABC/HCC, HCC id)
  - Subsequent frames contain command bit stream (9 frames)
  - Command frames are not guaranteed to be consecutive but will be in order
  - Last frame is a K3 + 6-bit action
- Fast Reset Frame
  - K3 + 6-bit action
- IDLE Frame
  - If no L0, BCR, Command or Fast Reset for frame: send IDLE
  - K0 + K1
  - Enables Sync at Startup and Resync

@4 Mhz L0, Max 8 L0 in 80 BC:
1 Command / 2 µs

Action	Meaning
0	NOP
1	Execute CMD
2	SYS Reset
3	Soft Reset
4	SEU Reg Reset

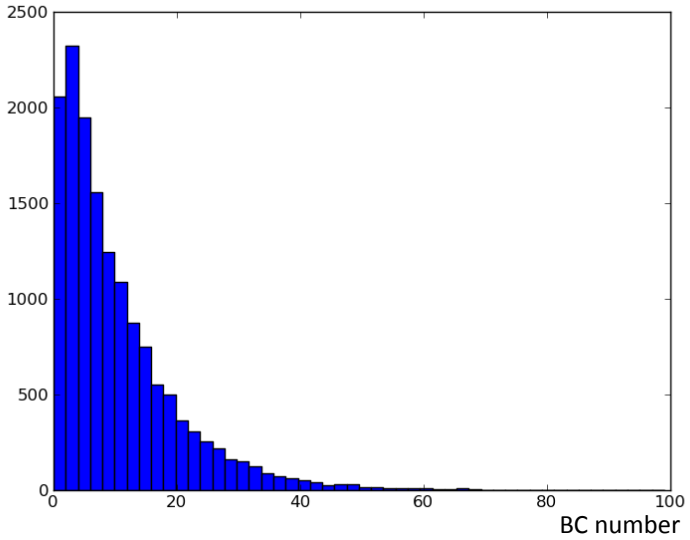
	Payload		Sent on line	
What	**6b - Binary**	**6b - Hex**	**8b - Binary**	**8b - Hex**
Idle: No L0, No BCR: K0+K1			0100 0111 0101 0101	47 55
L0: 0001, Tag: 0x55 (1010101)	0000 1101 0101	03 15	1100 0011 1001 0101	C3 95
No L0, BCR	1000 0000 0000	20 00	0110 0011 0101 1001	63 59
L0: 1000, BCR, Tag: 0x07(0000111)	1100 0000 0111	30 07	0111 0100 1000 0111	74 87
Begin CMD, ABC Cmd, HCC 5: K2+5	00 0101	05	0111 1000 1100 0101	78 C5
CMD Frame 0	0000 0101 1111	01 1F	0111 0001 0101 1100	71 5C
CMD Frame 1	0000 0000 1010	00 0A	0101 1001 1100 1010	59 CA
CMD Frame 2	0000 0100 1000	01 08	0111 0001 0110 1001	71 69
CMD Frame 3	0000 0000 0011	00 03	0101 1001 1100 0011	59 C3
CMD Frame 4	0000 0010 1010	00 2A	0101 1001 1010 1010	59 AA
CMD Frame 5	0000 0101 0101	01 15	0111 0001 1001 0101	71 95
CMD Frame 6	0000 0001 0101	00 15	0101 1001 1001 0101	59 95
CMD Frame 7	0000 0010 1010	00 2A	0101 1001 1010 1010	59 AA
CMD Frame 8	0000 0101 0000	01 10	0111 0001 0101 0011	71 53
End CMD, Load: K3+1	00 0001	01	0110 1010 0111 0001	6A 71

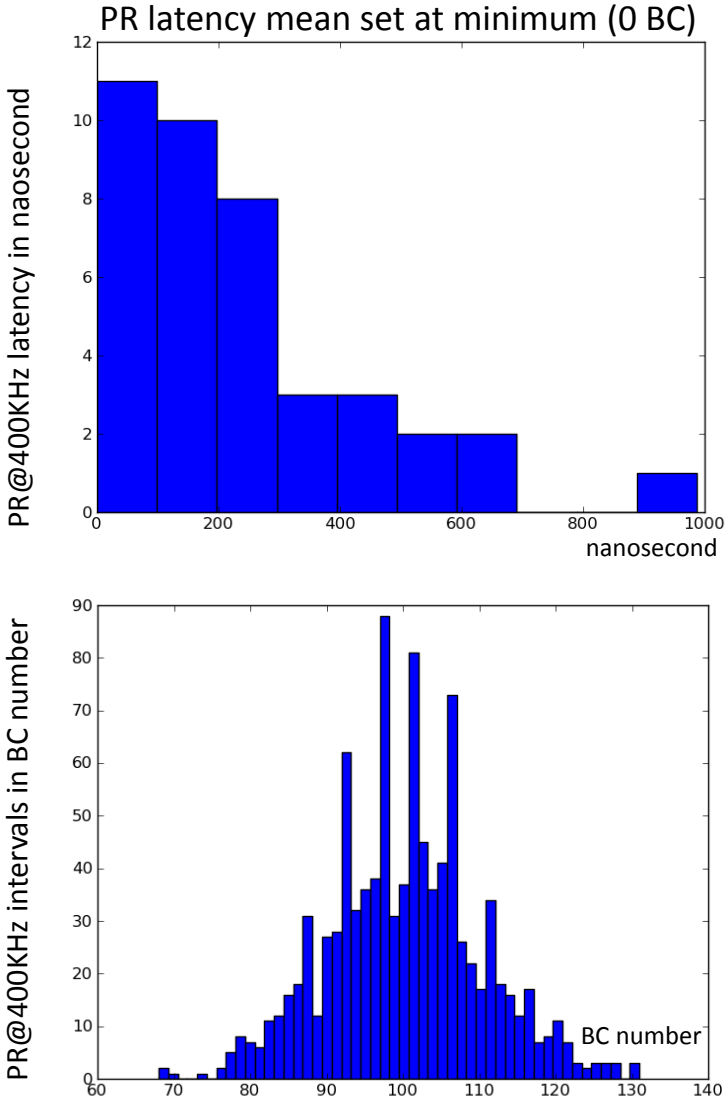
HCC/ABC addr 5/9, Reg0, Value 0xAA5555

ABCStar Design

- UVM verification setup

L0@4MHz intervals in BC number

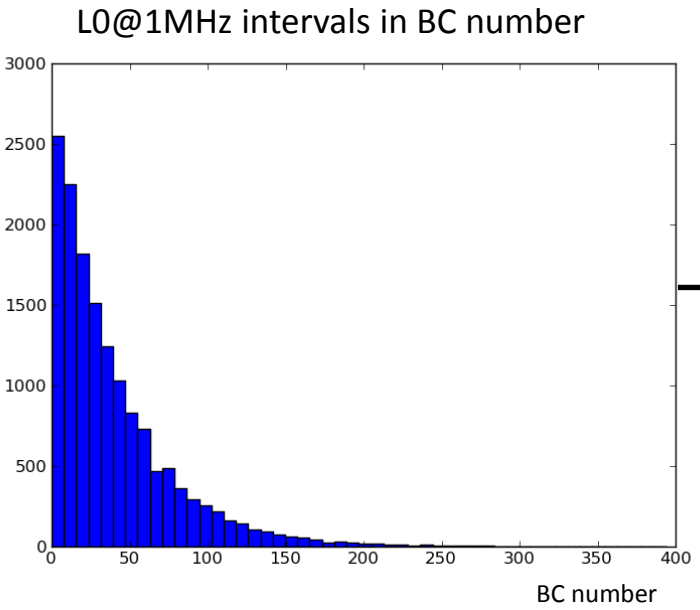


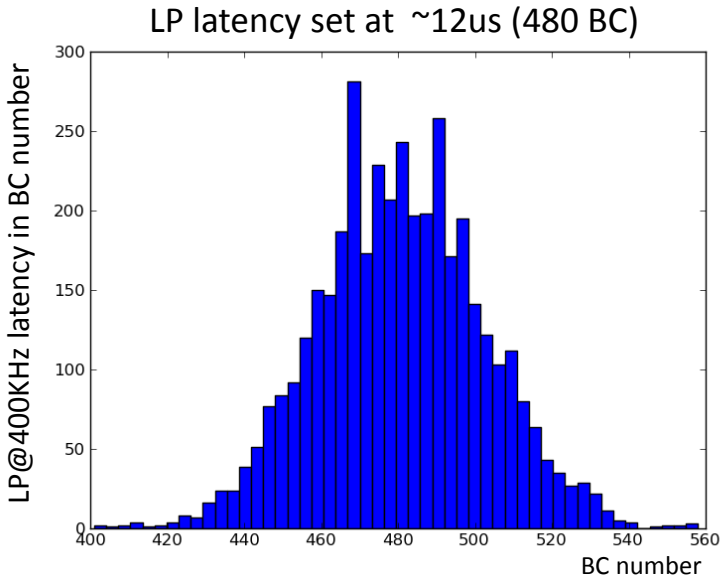


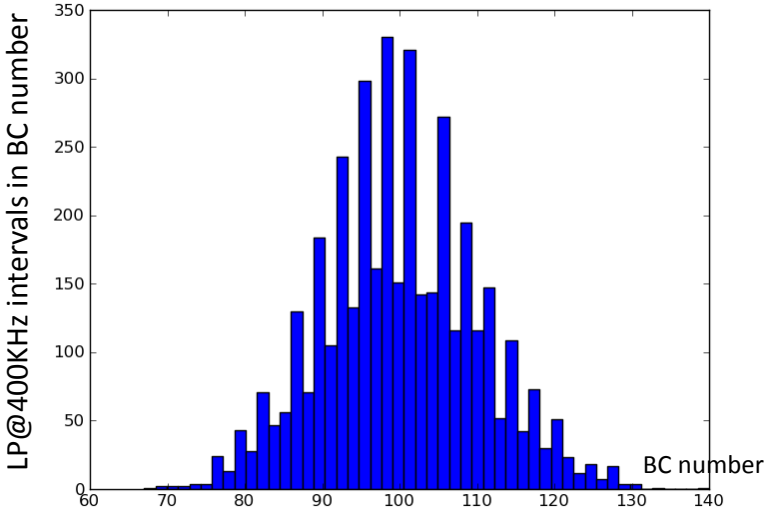
Low Latency L0 Scheme: 4 MHz L0s, 400KHz ROIs (PR), 600KHz L1s (LP)

ABCStar Design

- UVM verification setup



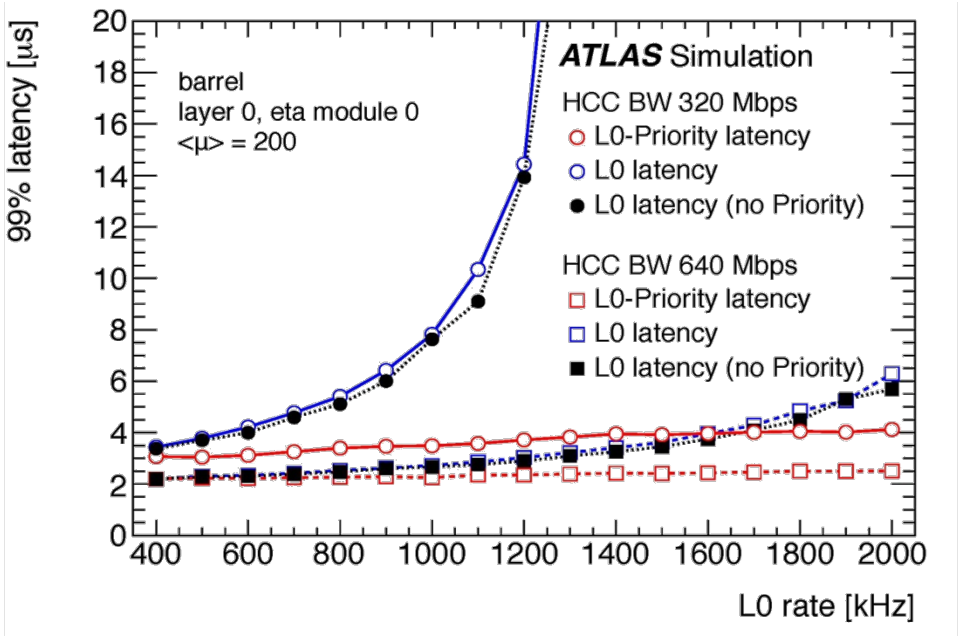


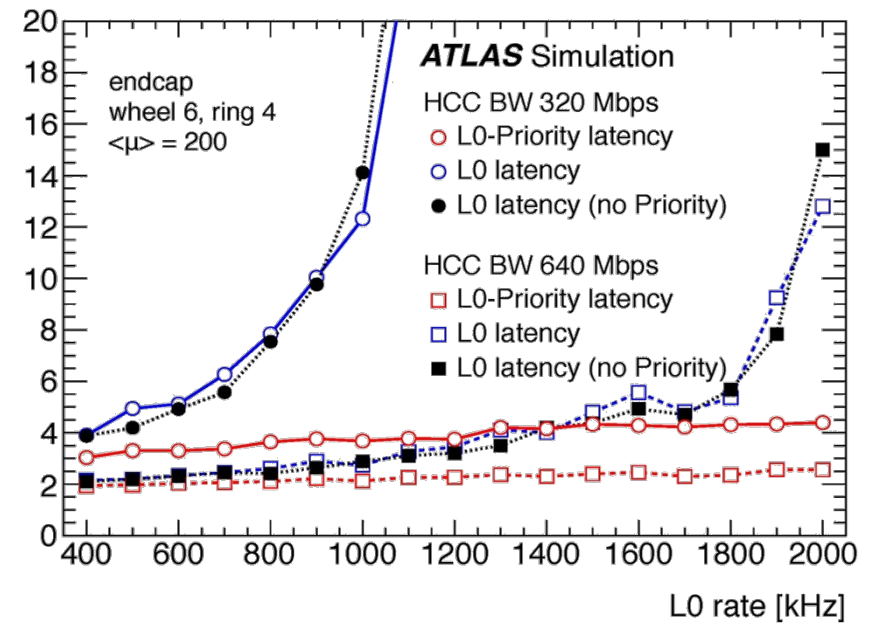


Original Trigger Scheme: 1 MHz L0s, 10% ROIs (PR), 400 kHz L1s (LP)

Managing the 1 MHz L0 Rate

- The change to the "star" hybrid architecture was not the only concern with the increase to 1 MHz event readout rate:
- The latency requirement for L0-P trigger is below 5us, to feed data to L1-Track
- Here are simulations of the readout time at two different output bandwidths.





The simulated latency for all data from 99% of all requests to arrive at the end of stave/petal for the highest occupancy Barrel and End-Cap layer module of the ITK as a function of the L0 rate for the scenario where all L0 events are read out from the detector. Detector occupancies commensurate with a mean occupancy of 200 separate pileup interactions per bunch crossing have been used.

ABCStar Design

SEU mitigation : protected state machines by Hamming code distance

1 bit flip sensitive

```
//state machine state table
parameter       STATE_IDLE = 0,
                STATE_START_BITS = 1,
                STATE_TYPE_CODE = 2,
                STATE_BCID = 3,
                STATE_LOID = 4,
                STATE_CLUSTER_VALUE = 5,
                STATE_REG_VALUE = 6,
                STATE_TERMINATOR = 7;
//state machine sequencer
  always @ ( posedge clk )
    if ( rstb == 'b0 ) begin
       State <= STATE_IDLE;
       popCVfifo <= 'b0;
       popRegfifo <= 'b0;
       cvc <= 'b0;
       last <= 'b0;
       data_busy <= 'b0;
       reg_busy <= 'b0;
    end
    else
    case ( State ) //synopsys parallel_case full_case
        STATE_IDLE:  begin
                bc <= 'h2;
                cvc <= 'b0;
                last <= 'b0;
                if ( data_send || reg_send ) begin
                    sr[63:0] <= {3'b111, 61'b0};
                    State <= STATE_START_BITS;
                    if ( data_send )
                        data_busy <= 'b1;
                    if ( reg_send )
                        reg_busy <= 'b1;
                end
```

1 bit flip Hamming code distance

```
//state machine state table
parameter       STATE_IDLE = 0,
                STATE_START_BITS = 11,
                STATE_TYPE_CODE = 21,
                STATE_BCID = 30,
                STATE_LOID = 38,
                STATE_CLUSTER_VALUE = 45,
                STATE_REG_VALUE = 51,
                STATE_TERMINATOR = 56;
//state machine sequencer
  always @ ( posedge clk )
    if ( rstb == 'b0 ) begin
       State <= STATE_IDLE;
       popCVfifo <= 'b0;
       popRegfifo <= 'b0;
       cvc <= 'b0;
       last <= 'b0;
       data_busy <= 'b0;
       reg_busy <= 'b0;
    end
    else
    if ( State == STATE_IDLE | State == 1 | State == 2 | State == 4 | State == 8 | State == 16 | State == 32) // Hamming code tolerance
                begin
                bc <= 'h2;
                cvc <= 'b0;
                last <= 'b0;
                if ( data_send || reg_send ) begin
                    sr[63:0] <= {3'b111, 61'b0};
                    State <= STATE_START_BITS;
                    if ( data_send )
                        data_busy <= 'b1;
                    if ( reg_send )
                        reg_busy <= 'b1;
                end
                else begin
                    State <= STATE_IDLE;
                    sr <= 'b0;
                    data_busy <= 'b0;
                    reg_busy <= 'b0;
                end
```

1 bit flip invariant state