FVTX Readout Electronics Design Details

The design of the readout electronics for the FVTX detector is based on three major constraints, imposed by the detector:

- Large instantaneous bandwidth 2(out lines)*200(MHz)*8448(chips) = 3.38 Tb/s
- Radiation hardness of readout components near the interaction point
- Large amount of I/O lines 2.5*8448(chips) ≈ 21,000 LVDS pairs

As a result, the readout electronics are logically divided into two independent blocks (see Figure 1):

- ***Read Out Card (ROC)*** – module which is located close to the detector and:
  - Receives LVDS data over a kapton cable
  - Combines and synchronizes data stream from several FPHX chips
  - Sends the data to the counting house (CH) over optical fiber
- ***Front End Module (FEM)*** – module, which is located in the CH and used to:
  - Receive data from ROC over a fiber link
  - Sort the incoming data according to the Beam Clock Counter(BCO)
  - Buffer the data from the last 64 Beam Clocks
  - Upon LVL1 trigger decision, ship the data from the Beam Clock of interest to the output buffer which ships data to the PHENIX Data Collection Modules

The output of the FEM connects to the standard PHENIX DAQ board – Data Collection Module (DCM) and from this point on the data stream becomes a part of the standard PHENIX DAQ.

ROC Design Implementation

The ROC board design utilizes radiation hard FLASH-based ACTEL FPGAs in order to be not susceptible to single event upsets. The proposed ROC board diagram is shown in Figure 2. The board contains 4 large scale ACTEL A3PE3000-FG896 FPGAs, 17 16-bit Serializer/Deserializer chips (TLK2711) and two 12 optical fiber transmitters (HFBR-772BEZ) to send the data to 2 FEM boards. Each FPGA holds two completely independent “ROC channels” which send out 32 bit data at the Serial Clock frequency (135 MHz). The outgoing data is logically split into two hi/lo 16-bit data portions which are each sent to separate Ser/Des chips and a single fiber channel. The fact that Station 1 has fewer chips to read out per wedge enables us to use the FPGA fabric from one of the ROC channels to distribute Slow Control signals to appropriate chips and wedges. Slow Control data are being sent by the FEM to the ROC over a single fiber interface.

The block diagram of a single “ROC channel” is shown in Figure 3. The channel combines the data from 52 FPHX chips (2 large wedges). Taking into account that the data word consists of 20-bits, serialized over 2 output lines, it takes 10 Serial Clocks to deserialize a single data word. This means that we can combine exactly 10 chips without incurring additional delay into the data flow.

The most optimal division of data flow from 52 chips into groups of 10 is 9+9+9+9+8+8 chips, leaving us with 6 almost equal groups of 9(8) chips. Each of the groups feeds an individual deserializer/combiner block and corresponding FIFO. Data from the 6 FIFOs is sampled by Round-Robin arbiter to the output buffer.

Figure 4 shows the diagram of a single "Deserializer and Combiner block" from Figure 3. The LVDS data from a single chip are latched to the Serial Clock with the Phase Follower block and are thus synchronized, and deserialized. On the output, we have 20 bit data words, lasting for exactly 10 clock cycles. A multiplexer sample the incoming data stream once every 10 clocks, adds the chip ID and ships the data to the output FIFO.

More information on the design of the "phase follower" and "data synchronization block" can be found in Figures 5,6. The standard methodology from Xilinx technical notes (with little modification) was used for the "phase follower" blocks. As a result, the data nicely latch to the edge of the clock opposite to the one, where the signal transition is found. With this design the ROC board becomes immune to problems that might typically arise from possible variations of input data cable length or propagation delays. The deserializer and Synchronization block searches for a Sync Words – data stream "fillers" that are sent by the FPHX chip whenever it has no data to send. Sync Words have a unique set of bits set, which can not be found in valid data words. Data are latched by the Sync Word flag and at the same time the counter, synchronized with the Sync Word flag, starts counting 10 clock intervals. The condition when the counter reaches the threshold and there is no sync word flag is treated as decoding of the Data Word. Sync Word propagation stops at this point as they are not needed after this stage.

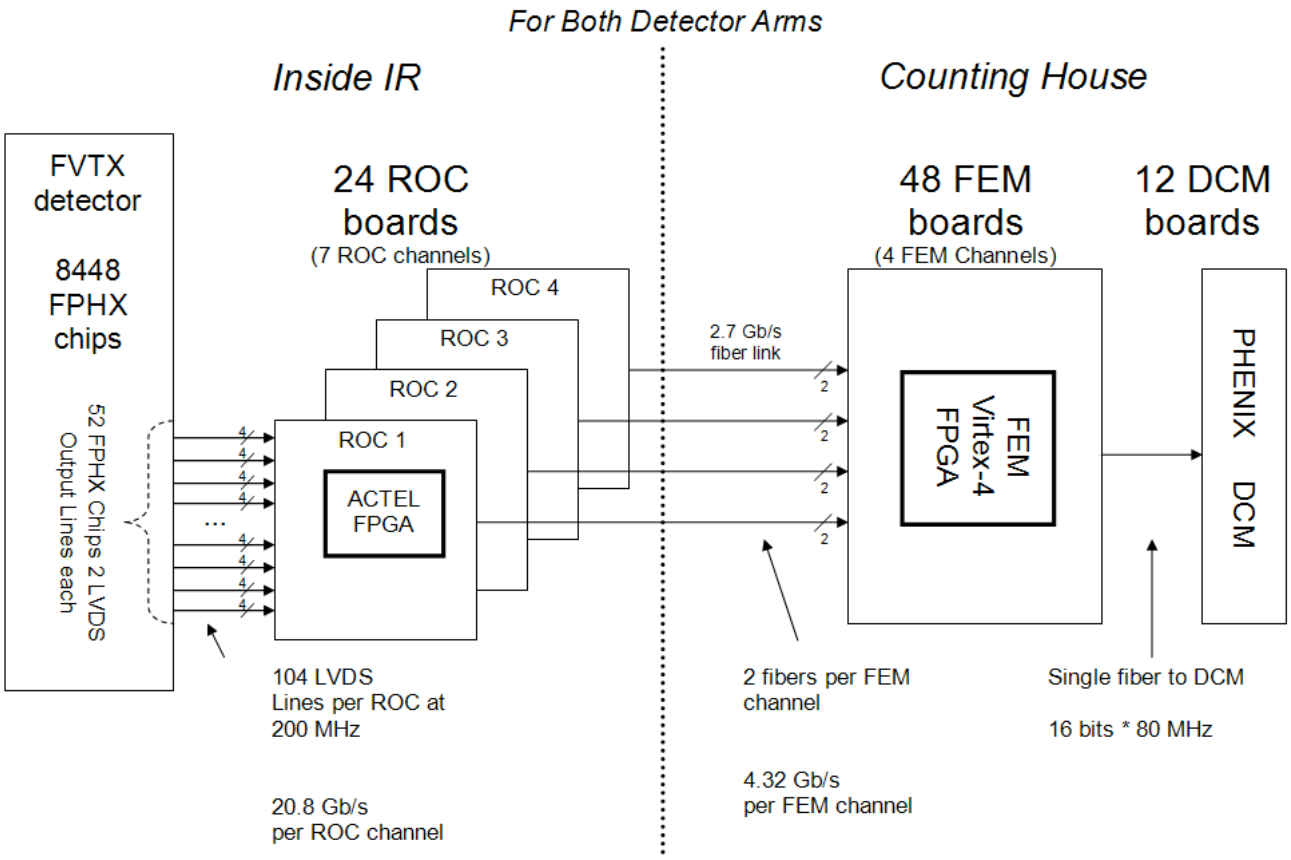


Figure 1. FVTX Detector Readout Diagram

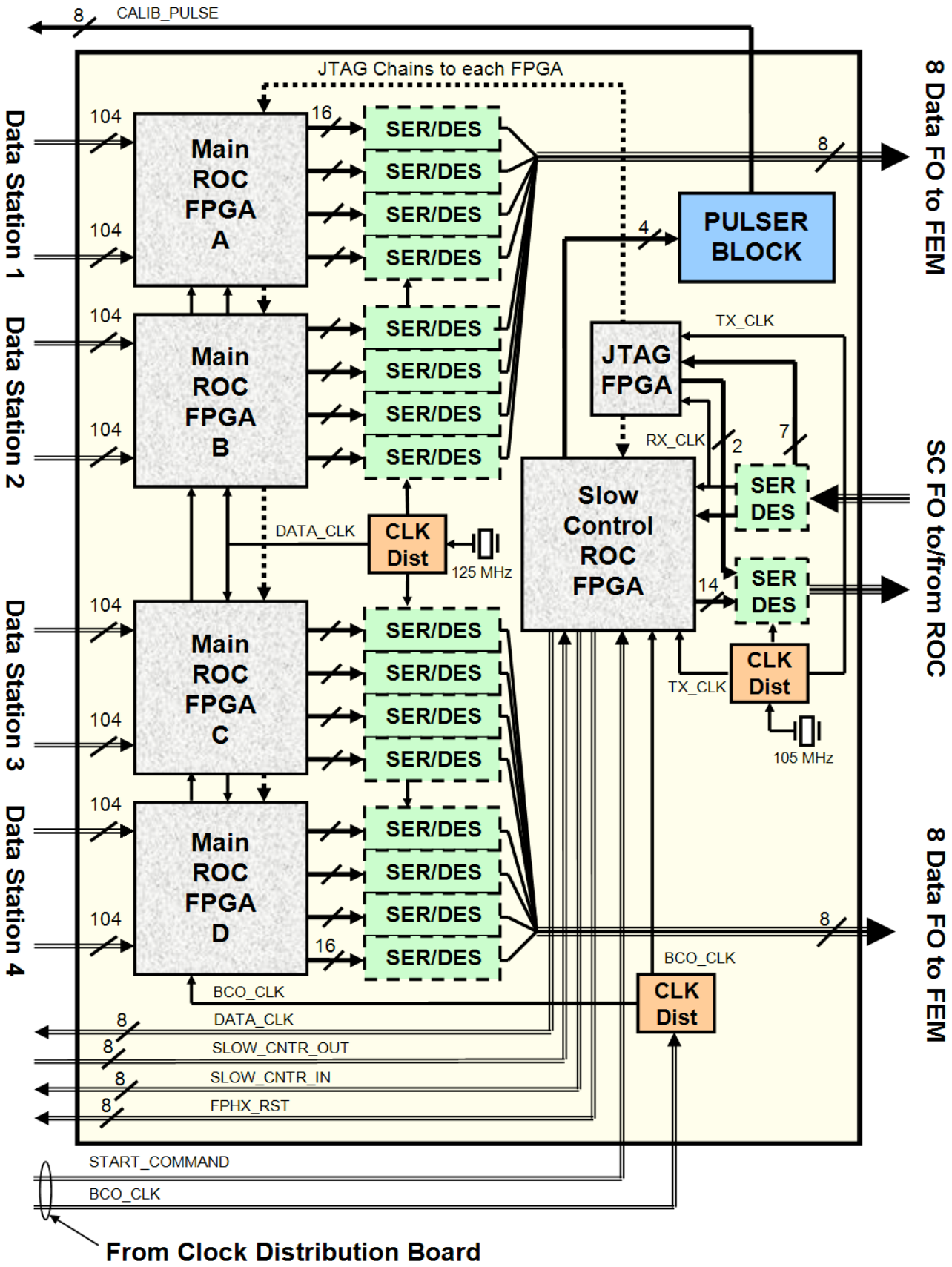


Figure 2. ROC Board Block Diagram

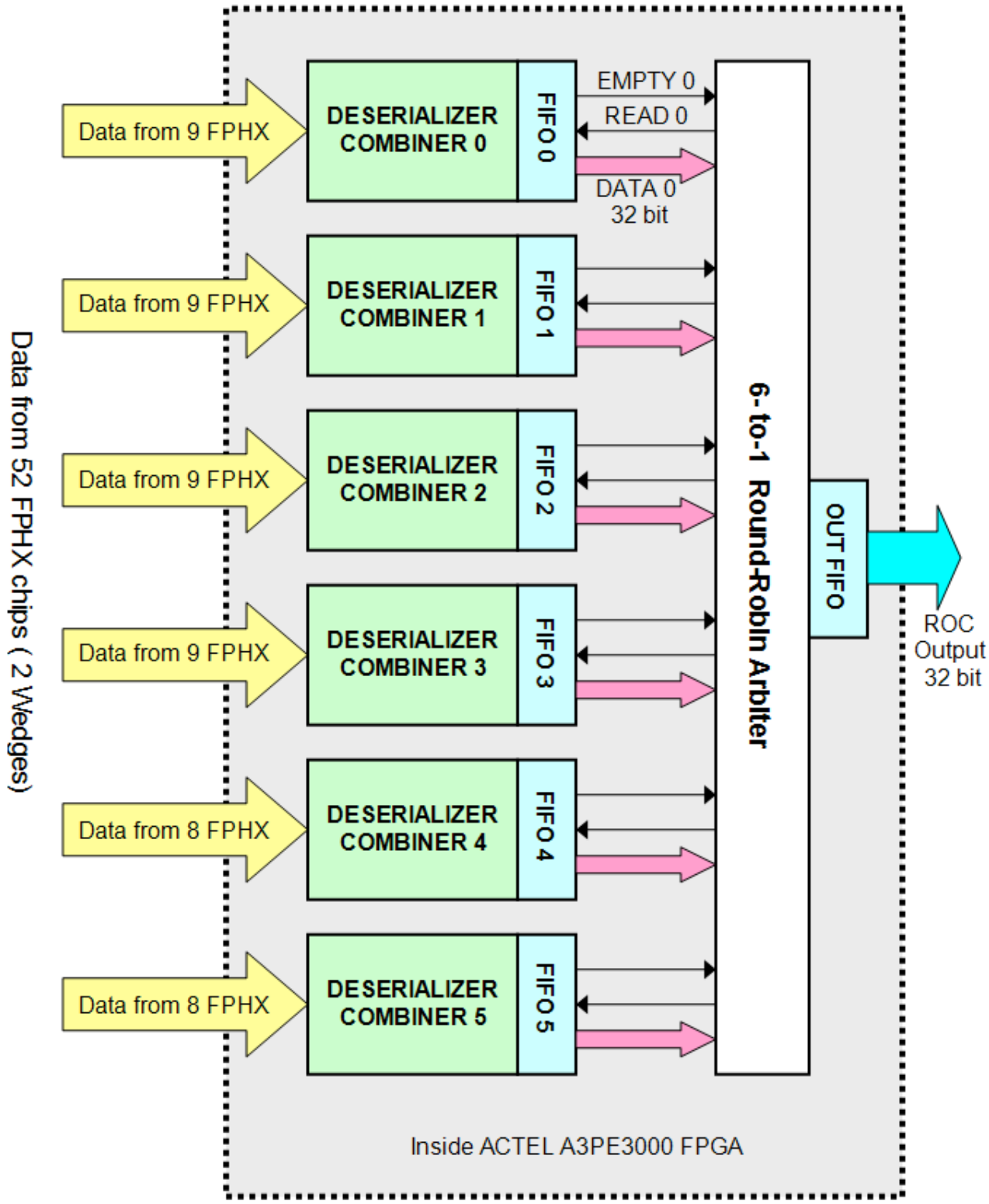


Figure 3. Single ROC Channel Block Diagram

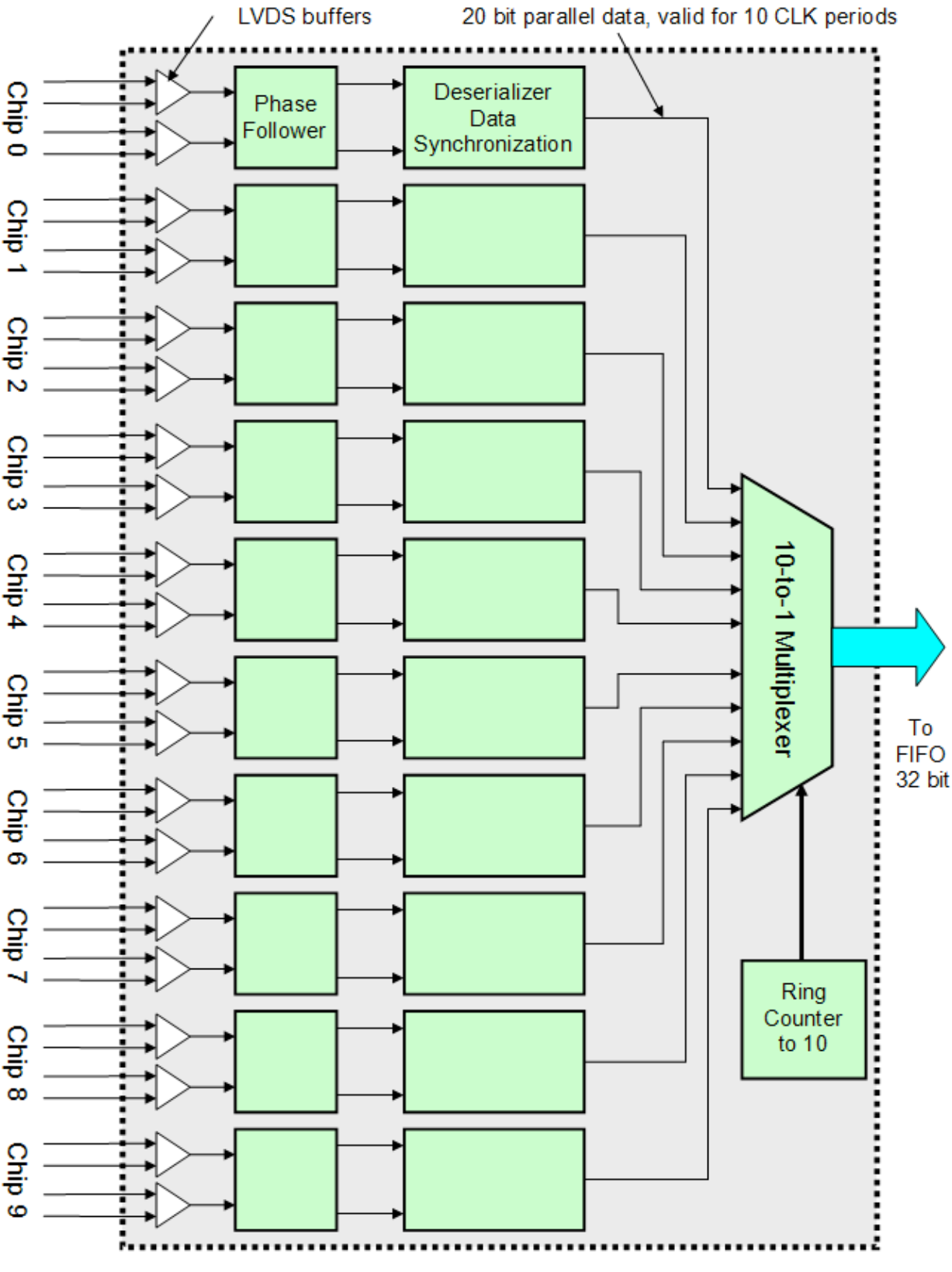


Figure 4. Single 10 Chip Combiner Block Diagram

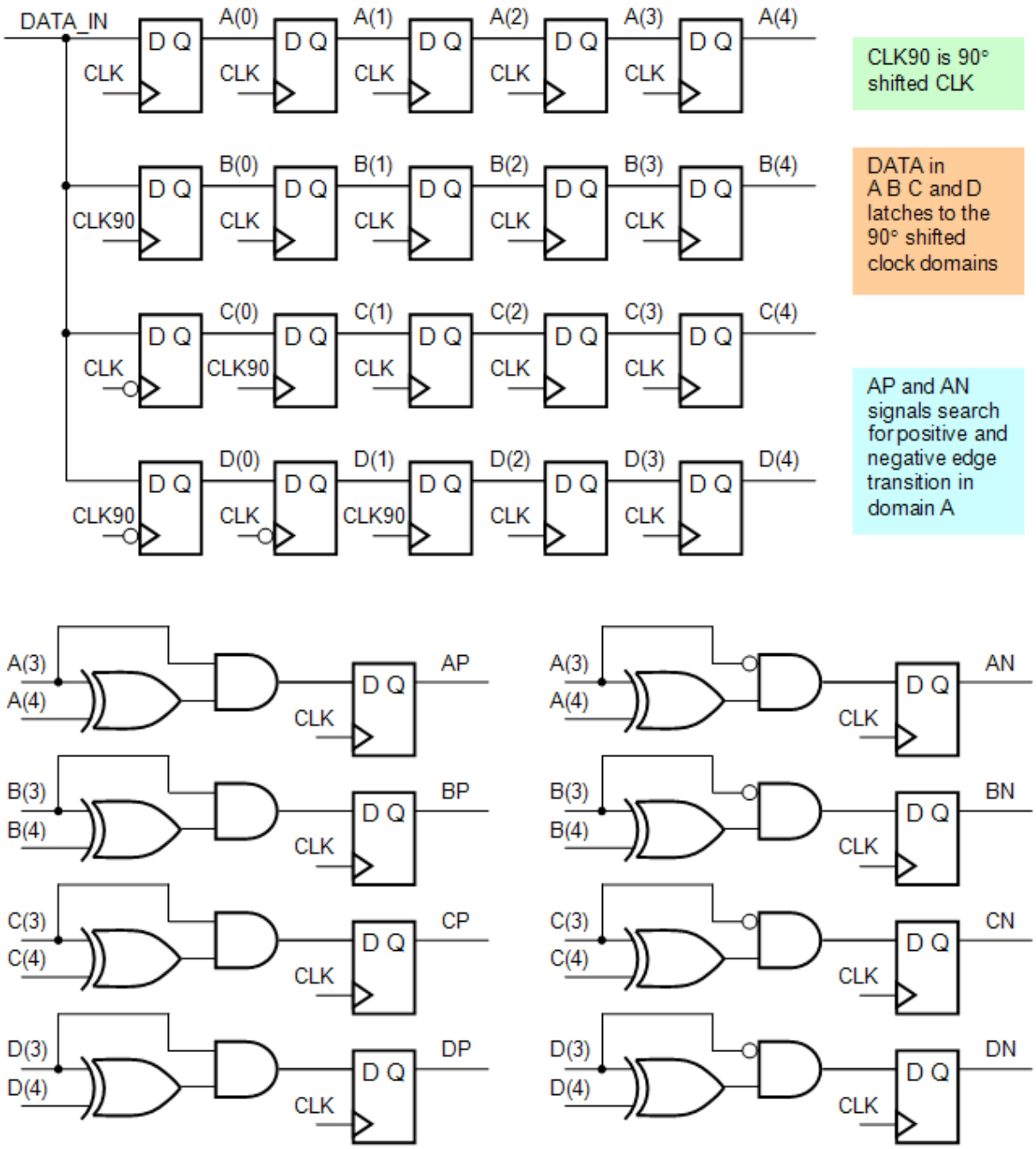
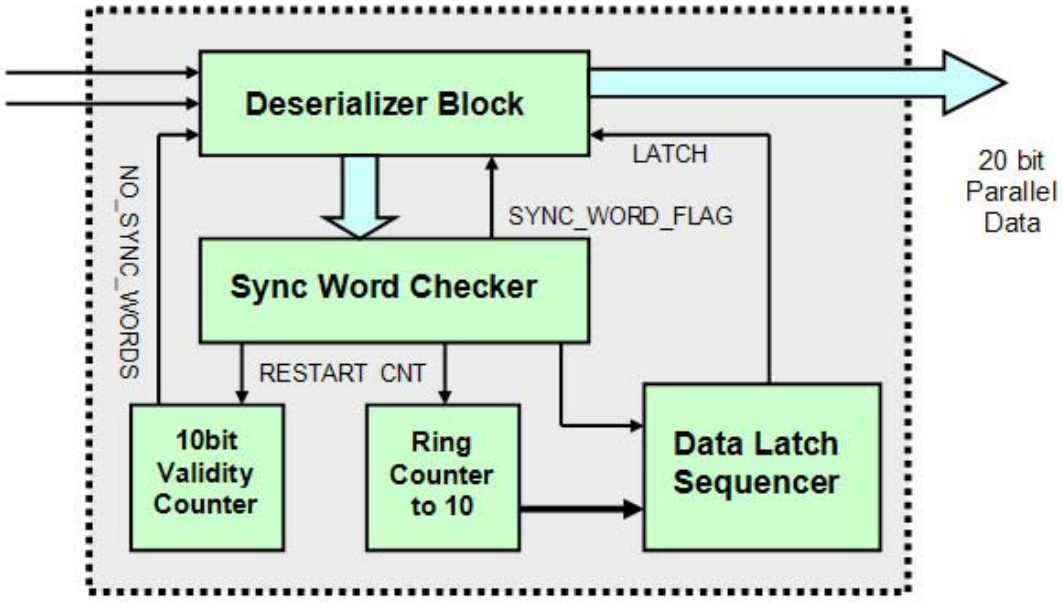


Figure 5. Single 10-Chip Combiner Block Diagram



Deserialized 20 bit Data Word is sampled on every clock for a Sync Word pattern

If Sync Word is found, **ring counter to 10** is restarted

If there were no Sync Words detected in 1024 consecutive clocks, RST is issued to deserializer

Data Latch Sequencer sends **Latch** request whenever

- Sync Word is detected
- Counter reached threshold value (assume Data Word)

If **Latch** is sent for the Sync Word → **DATA_OUT = 0 (remove Sync Words)**
If **Latch** is sent for the Data Word → **DATA_OUT = DESERIALIZED_DATA**

Figure 6. Data Synchronization Block

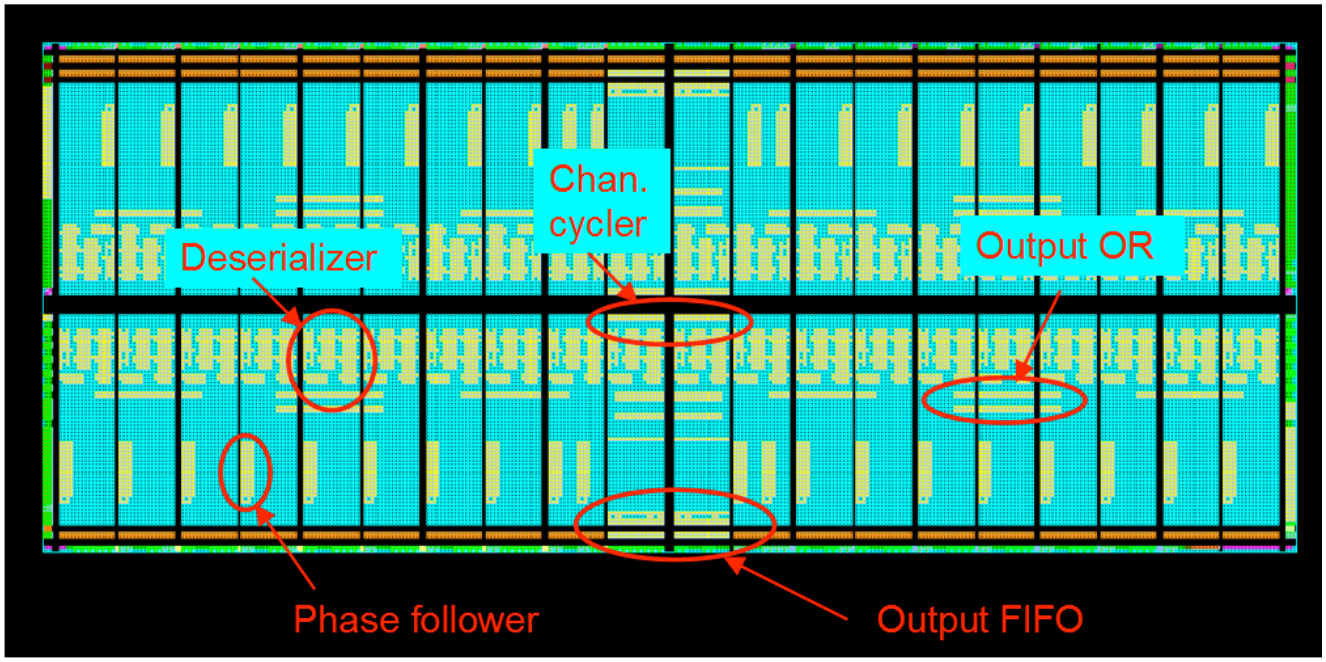


Figure 7. iFVTX ROC design placement on A3P1500 fabric.

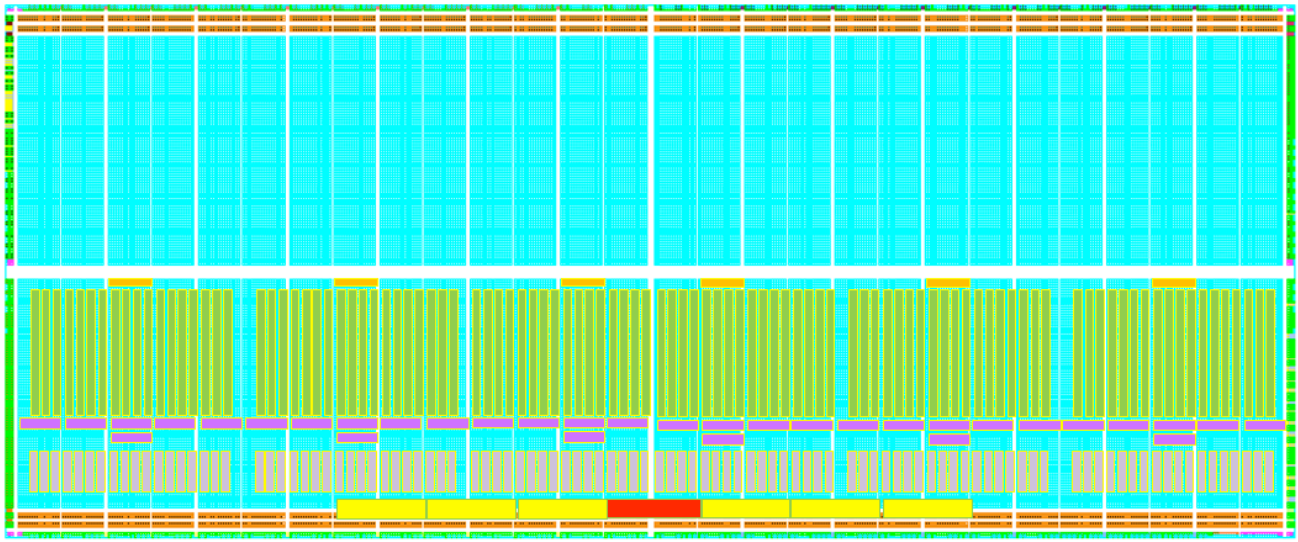


Figure 8. Proposed placement of the design blocks on A3P3000 fabric
for a single ROC channel of FVTX ROC. No real placement yet.
This is a test whether we have enough space on the chip.

FEM Design Implementation

The FEM board is going to be located in the Counting House and thus has no radiation tolerance requirements. As the design of the FEM requires a lot of memory management, we propose to use the VSX series Xilinx Virtex-4 FPGAs for the main FEM logic. Optimization of cost/FPGA suggested us to use one of the largest possible FPGA devises (XC4VSX55) and combine 4 FEM channels onto a single FPGA. An alternative solution of using 4 smaller FPGA devices (XC4VSX35) on a board increases the overall cost of a FEM board by 50%.

The FEM board block diagram is shown in Figure 9. One FEM board receives 8 optical fibers (½ of a ROC board). The incoming 16-bit data nibbles from the ROC are aligned and combined into 32-bit data words ("comma" bits are planned to be used in transmission). Data are buffered for 64 Beam Clocks in one of 4 FEM channels.

A separate small-scale FPGA (Spartan3 XC3S200) processes incoming timing signals and slow control commands. Timing in PHENIX is distributed through a specially designed Granular Timing Module (GTM). The GTM signals include a 9.4 MHz Beam Clock, a Level-1 Trigger and Control Mode Bits. These signals are copied to the main FEM FPGA and also sent to the ROC board.

The Slow Control FPGA also controls download of initial register values into each FPHX chip that a given FEM controls. Initial values for registers are stored in a 1MB EEPROM on the FEM board. The content of the EPROM is transferred to the FPHX chips, as is, upon request and individual values are modified via the slow control interface.

Upon receiving of a Level-1 trigger, the FEM channels start sending the data from a particular "clock bucket" to the output sub-event buffer. This process should be as fast as possible so the data from beam crossing X+64 does not arrive until the bucket for clock X has been emptied. The current speed for the read is set to 300 MHz. The resulting sub-events are combined and packetized in the "Channel Combiner" into 16-bit wide packets (see Figure 11). Packets carry fixed data (Event Counter, BCO Counter and Longitudinal Parity) in header and trailer words.

The structural diagram of a single FEM channel is shown in Figure 10. The main idea of the block is to sort the data, according to the Beam Crossing Counter, which is embedded into the data stream. The lowest 6 bits of the BCO Counter are used to address the data into one of 64 FIFOs (we use Xilinx 36bit x 512 built-in FIFO blocks, rated up to 320 MHz for writing and 450 MHz for reading). Each FIFO in the array has control logic around it, which is used to reset the particular FIFO in case of clock counter "wrap around" or prevents the data from reading if they were in the FIFO for more then 64 Beam Clocks (expired). The read Sequencer block selects a FIFO to read upon Level-1 trigger and quickly moves the data out to a sub-event buffer. The address to read has a constant adjustable delay with respect to the current Beam Crossing. Synchronization of Beam Clock Counters on all FPHX chips and the FEM is guarantied by a start sequence we use which synchronizes all the clock counters.

An additional Trigger FIFO (8 words deep) is used to store late Trigger Requests. This is needed if, during a Level-1 Request, data are still being transferred from a previous request. The read address of this trigger will be stored in the Trigger FIFO and the new "read" will be executed once the previous "read" is completed.

A single FEM channel design has been successfully implemented on an ML402 VSX35 evaluation board and has been proven to work at our specified input and output data rates. The design of a full 4 channel FEM module is coded and ready to be implemented (see Figure 12).

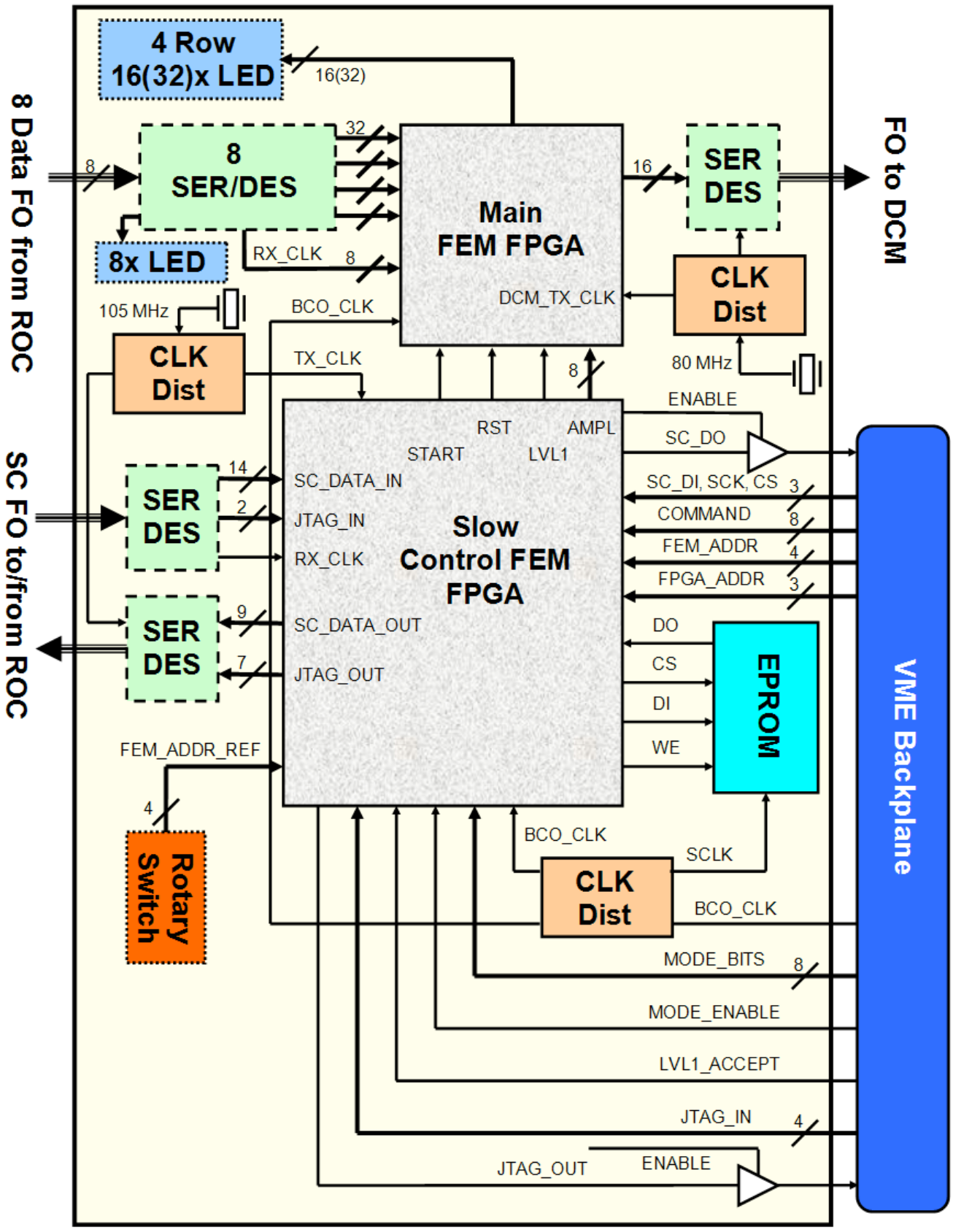


Figure 9. FEM Board Block Diagram

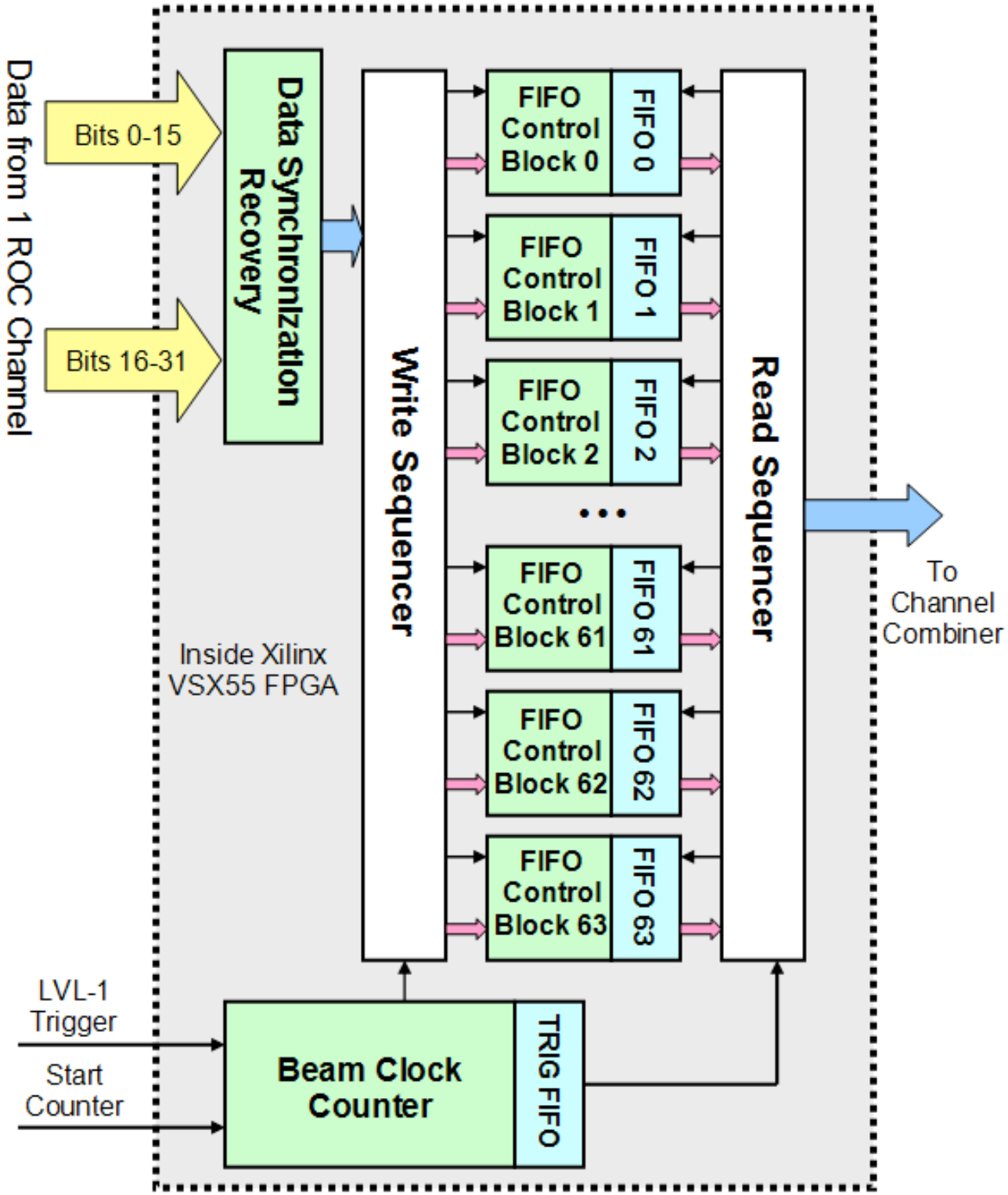


Figure 10. Single FEM Channel Block Diagram

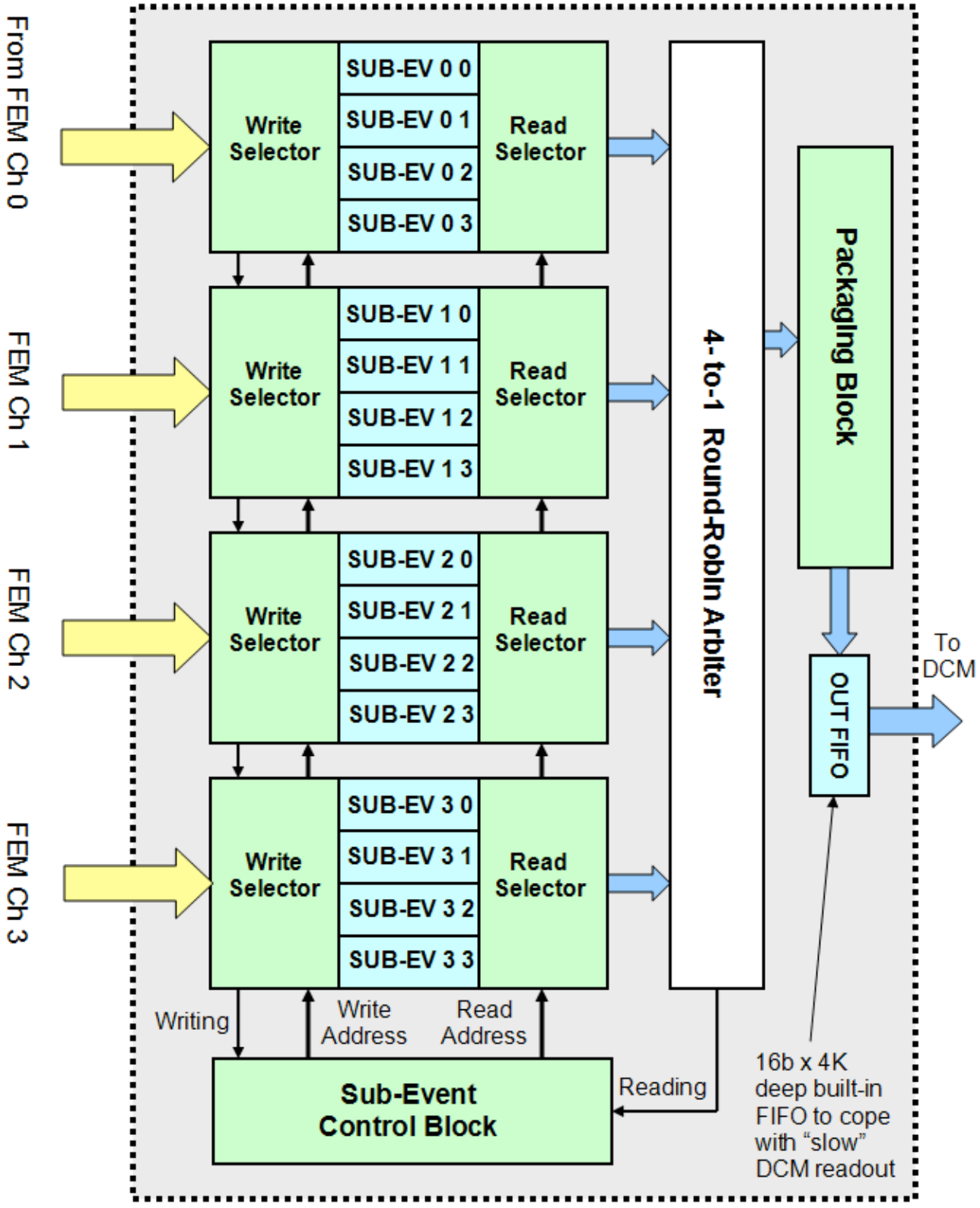


Figure 11. FEM Channel Combiner Block Diagram

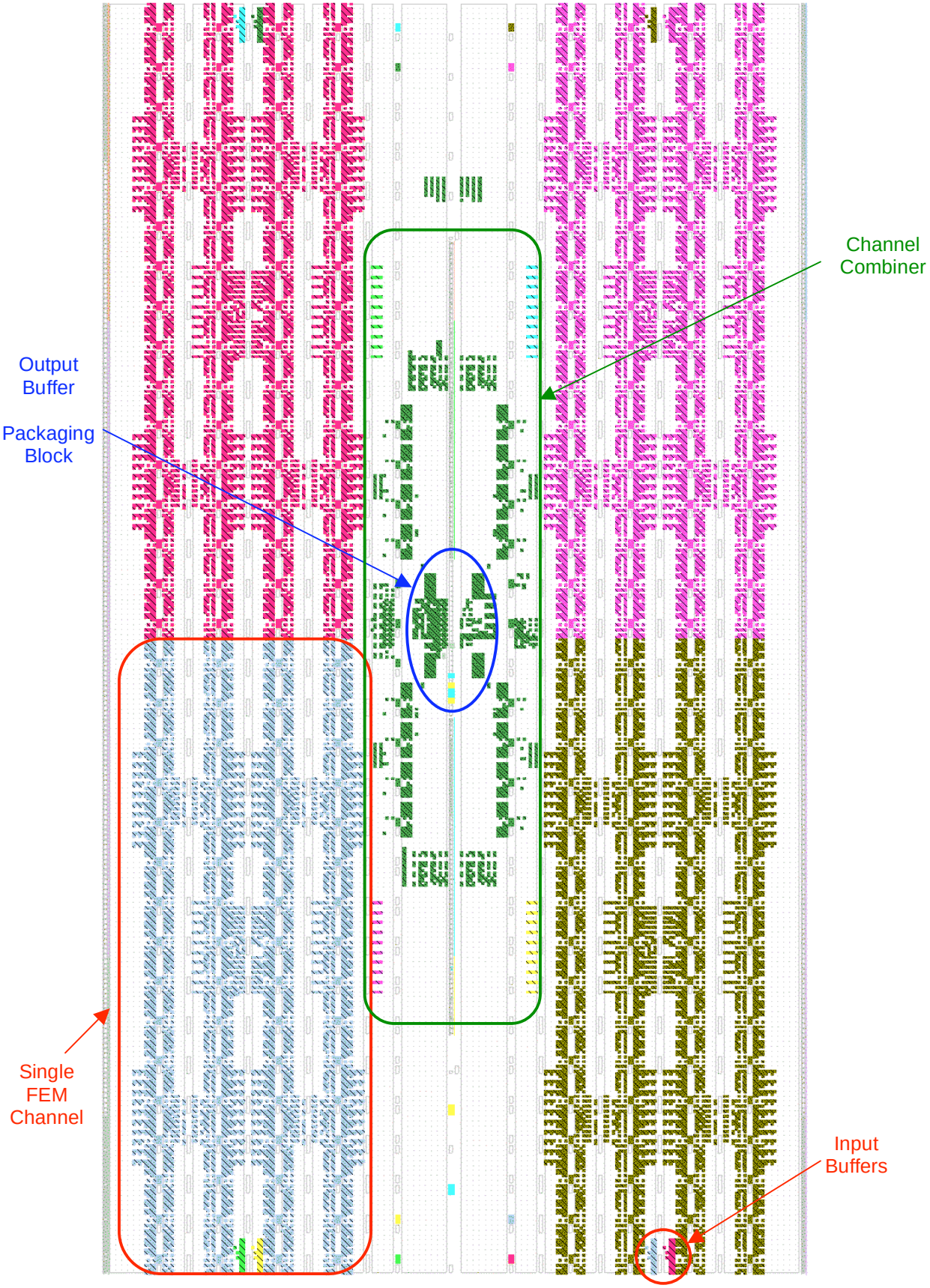


Figure 12. FEM design on Vertex-4 VSX55 fabric (with Channel Combiner)

FEM and ROC Prototype Performance

The design of the FEM and ROC channels was tested using a single FPIX2.1 and an 8-chip FPIX2.1 hybrid module. The FPHX chip is designed by the same FNAL group, which is designing the FPHX chip. The FPHX chip will have a very similar digital interface as the FPIX 2.1.

Tests of the proposed designs included testing signal propagation throughout the design using “fake” data and real data generated through injection of the test pulse into a particular FPIX pixel, cosmic data. All the components have been optimized and tested to perform at proposed design speed levels. The ultimate test for the ROC and FEM design was a full calibration scan of a particular detector at full speed, which was performed successfully.

The ROC channel data combiner block was tested by implementing a 24-chip combiner (8 input channels were connected) on an ACTEL ProASIC3 Starter Kit board (with A3PE600 FPGA). The serial data word from an FPIX chip consists of 24 bits and we used single line readout for testing. Figure 13 shows the serial output (yellow) from the one of 8 chips, Sync Word detection flag (blue) and “ring counter to 24” threshold output (green). One can clearly see the Data Word in the data stream and it is identified with 100% efficiency.

Figure 14 shows the data combining from different chips. One can clearly see a packet of 7 hits is being written into an output FIFO at full speed (green). Each chip is sampled only once per 24 clocks and we combine the data correctly (one chip out of 8 had a broken LVDS line and could not be read-out). The yellow line shows those data being read out from the Output FIFO by the NI DAQ card at 20 MHz to the PC memory. The read and write clock are running at 135 MHz, which is close to the limit for FPIX chip. We need to wait for the FPHX chip to test the readout chip operation and data writing at 200 MHz, but for reading, we’ve reached the design goal.

The FEM channel was prototyped on an ML402 Veirtex-4 evaluation board and only a single FPIX chip was used for testing. The prototype included imbedded deserializer and slow control parts, which will not be present in the final FEM core. In order to run the design, a “fake” trigger signal was created from the “fast OR” of all the pixels analog signals (so called GOT_HIT signal). The GOT_HIT was delayed by a specified number of clocks to simulate PHENIX Level-1 decision latency. The data was requested for read from a calculated Beam Clock bucket. One can see on Figure 15 that we read the data out of the array to the output buffer only if the readout delay is correctly adjusted. Whenever, we are off, the data does not show on the output. One can also see that data does not stay in the bucket forever, but has a 64 Beam Clock expiration time (otherwise we would observe random hits, written way back in time, while reading from an incorrect bucket). The FEM design functioned reliably at up to 150 MHz for writing and 300 MHz for reading.

Figure 16 shows the results of full calibration scan of 8 FPIX chip module performed with the ROC prototype. During the scan, fully controlled from slow control FPGA, one pixel is being enabled on the chip and then we send 100 pulses of fixed amplitude, gradually increasing the amplitude in 64 steps. Once the pixel is tested, the next one turns on, etc. As a result, we count the number of recorded hits, with the given amplitude. One can clearly see the threshold behavior of hit detection probability. The results clearly indicate that 100% of the signal propagates through the design and the signal is being transmitted without errors.

The full scale design of the FEM FPGA code has been tested using Xilinx XC4VSX55 evaluation board. The inputs were connected to a fake event generator block which generated predefined sequence of input hits and LVL1_ACCEPT at a predefined beam crossing. The output of a single packet, generated on the readout for iFVTX detector is shown in Figure 17. **There is no suppression mechanism for empty packets in the design as we do not know the size of the packet a-priory**. And finally, the simulation and read-back of several events, triggered in consecutive beam crossings is shown in Figure 18,19. Up to **4 events** can be consecutively requested for reading guarantied by construction. **5 event** buffer implementation is possible (one more sub-buffer added in each channel), but **very undesirable** due to complications in routing and overall design complication.

Finally the proposed input/output and internal data structure is shown in Figure 20,21.

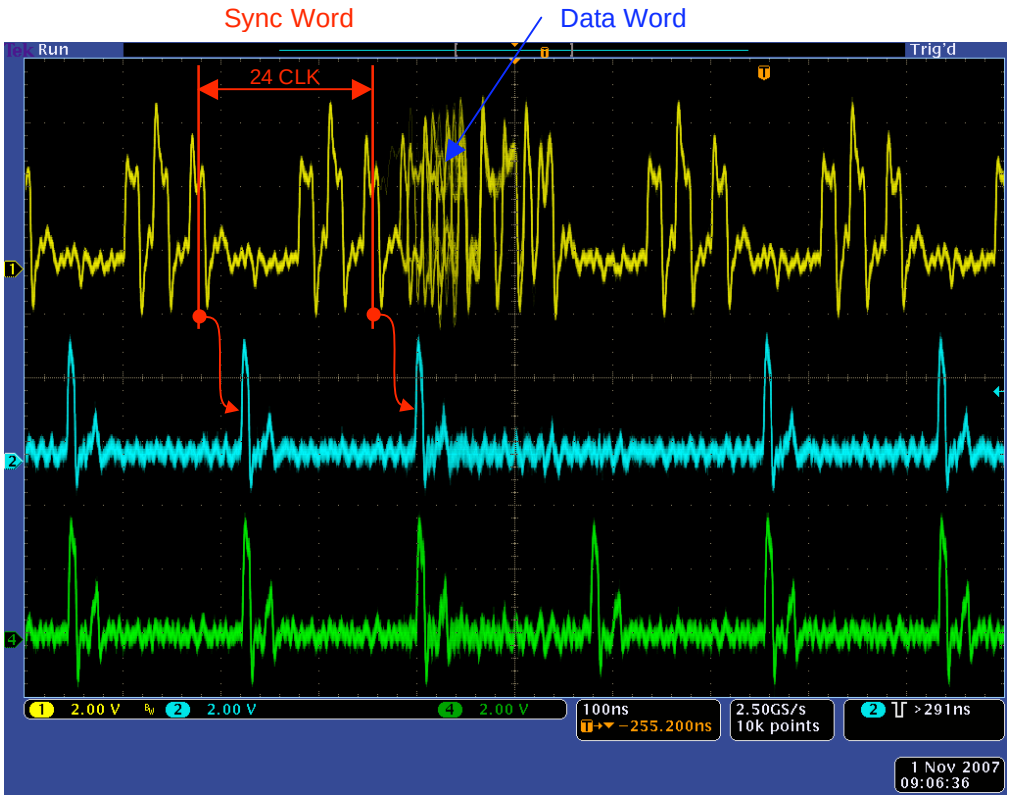


Figure 13 Scope capture of data synchronization in ROC prototype

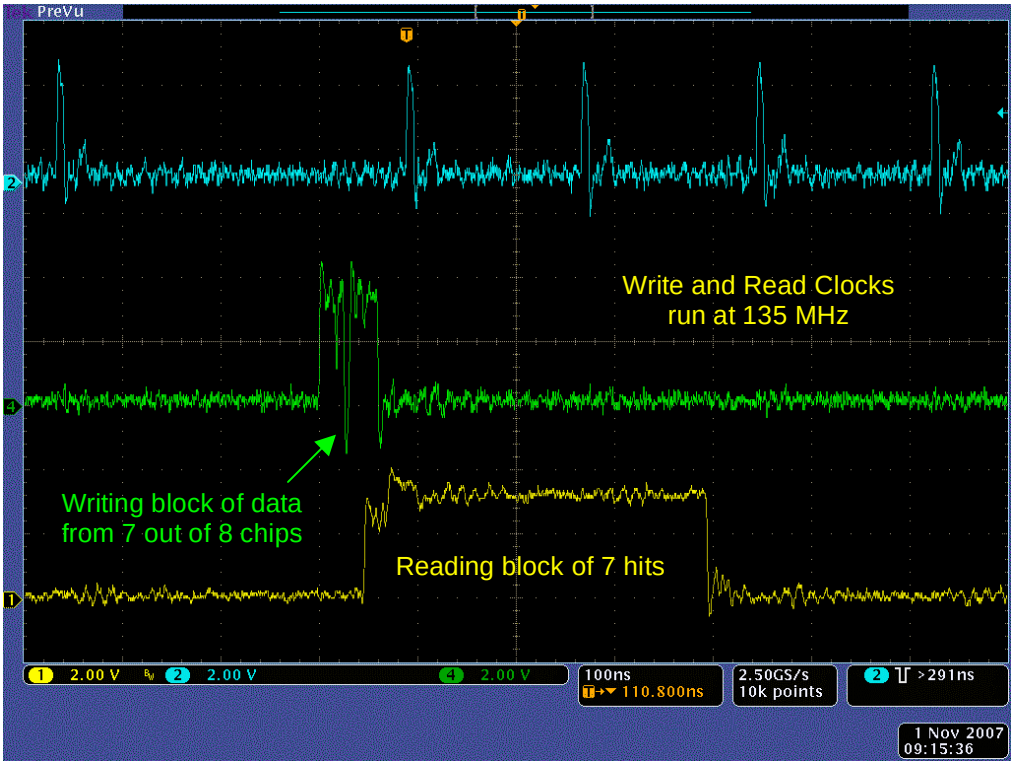


Figure 14. Combining data from 8 chips in ROC prototype

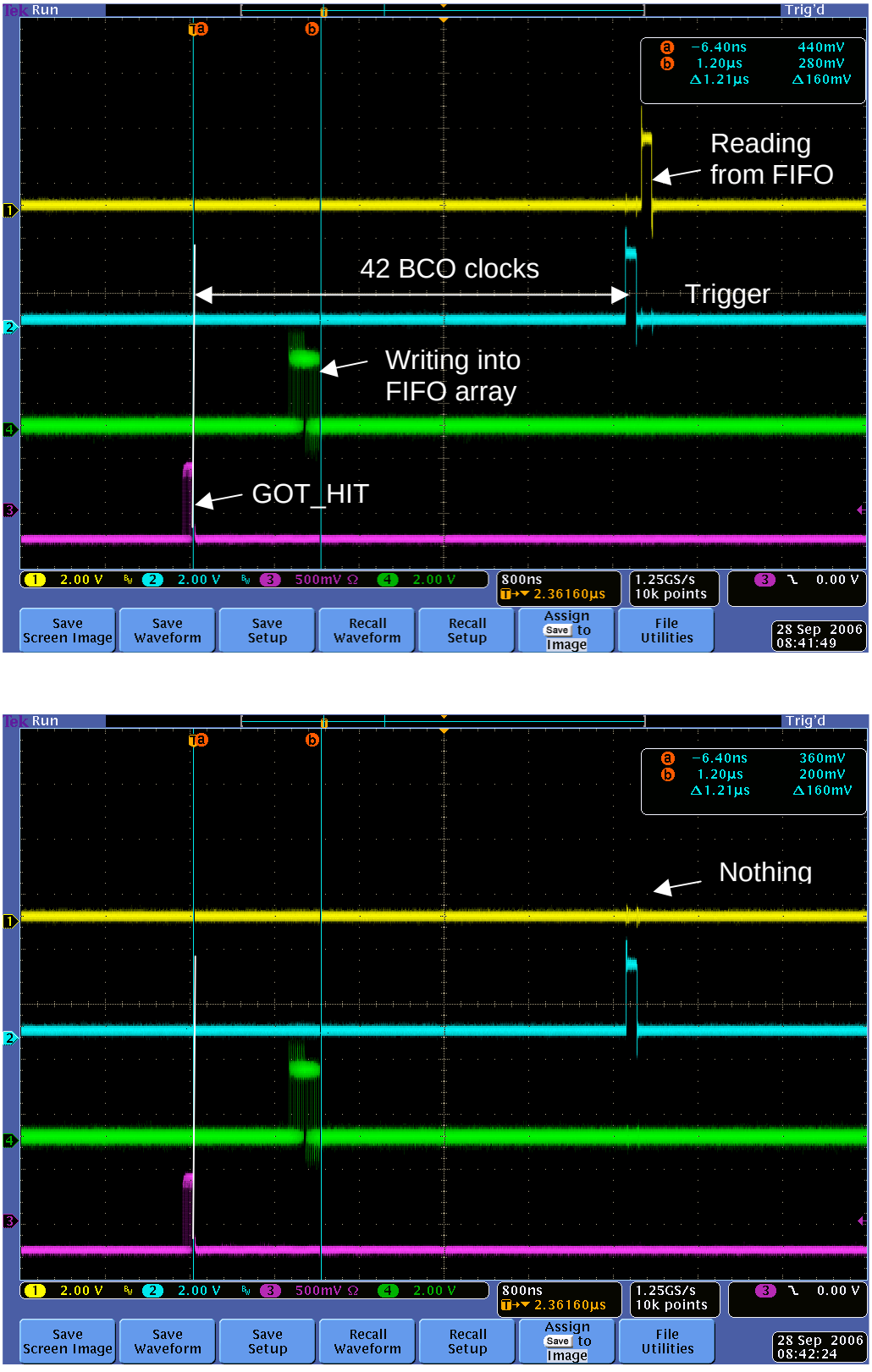


Figure 15. FEM prototype testing oscilloscope pictures.
Reading from the correct Beam Clock bucket (top)
and from an incorrect one (bottom)

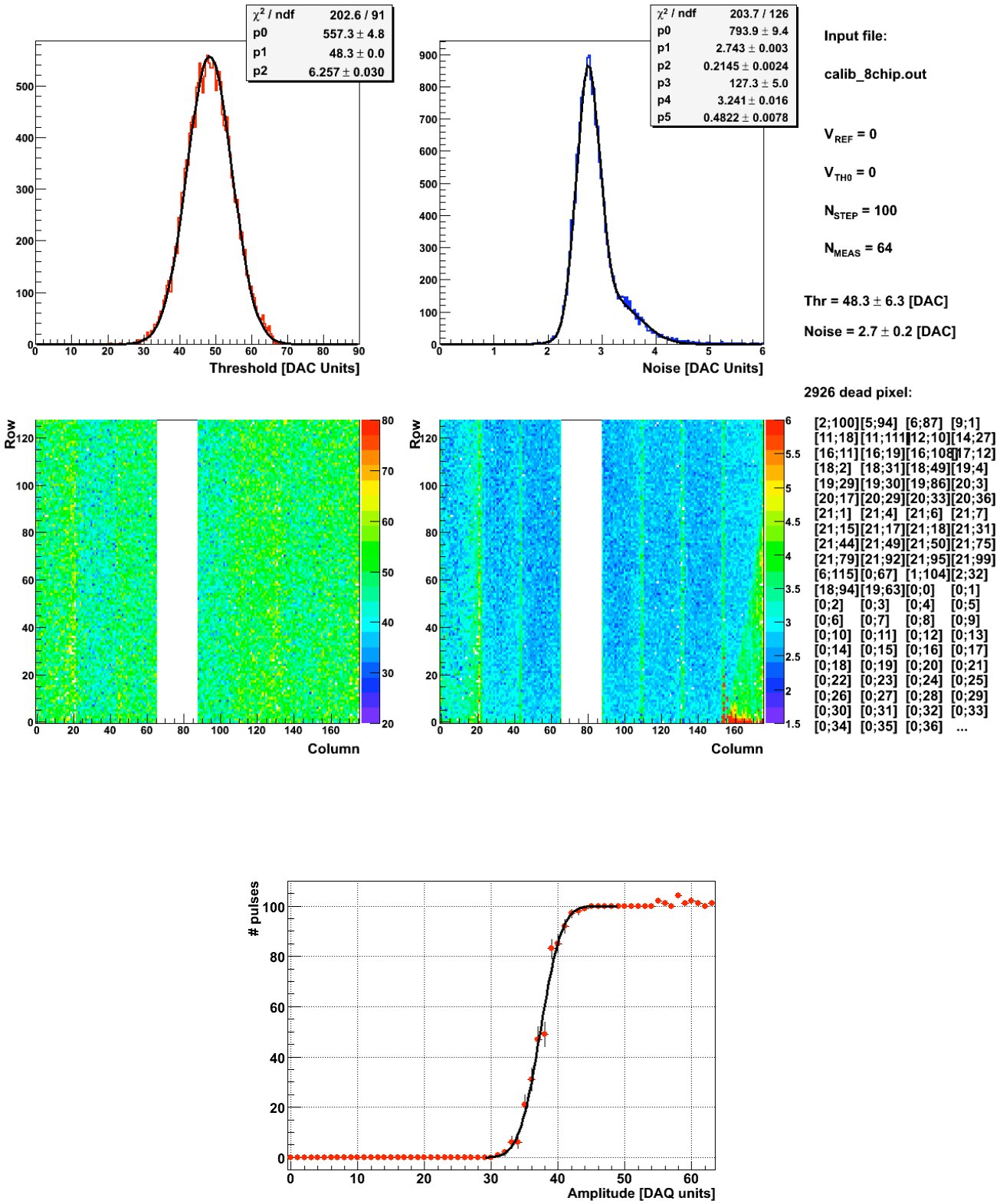
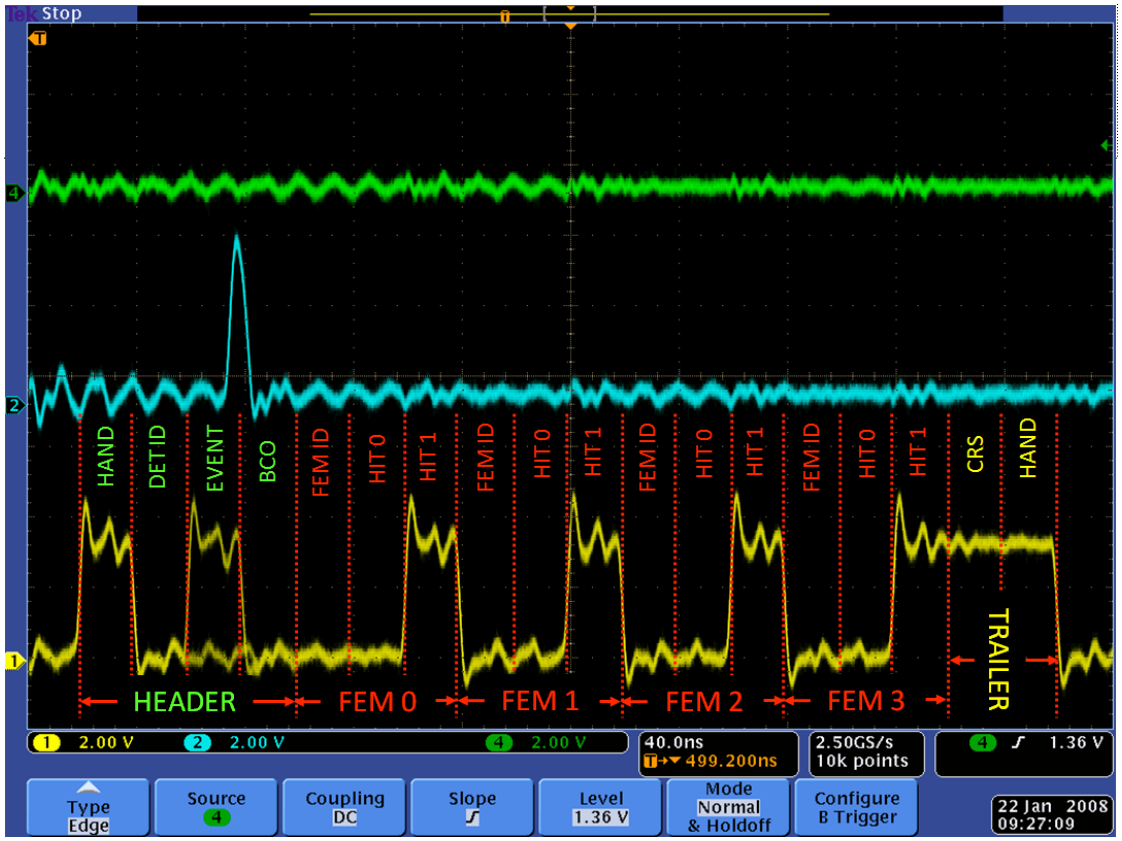


Figure 16. Threshold scan of 8-chip FPIX module using ROC channel prototype.
Threshold and noise distribution of each pixel (top)
and fit to the turn-on curve for one of the pixels (bottom)



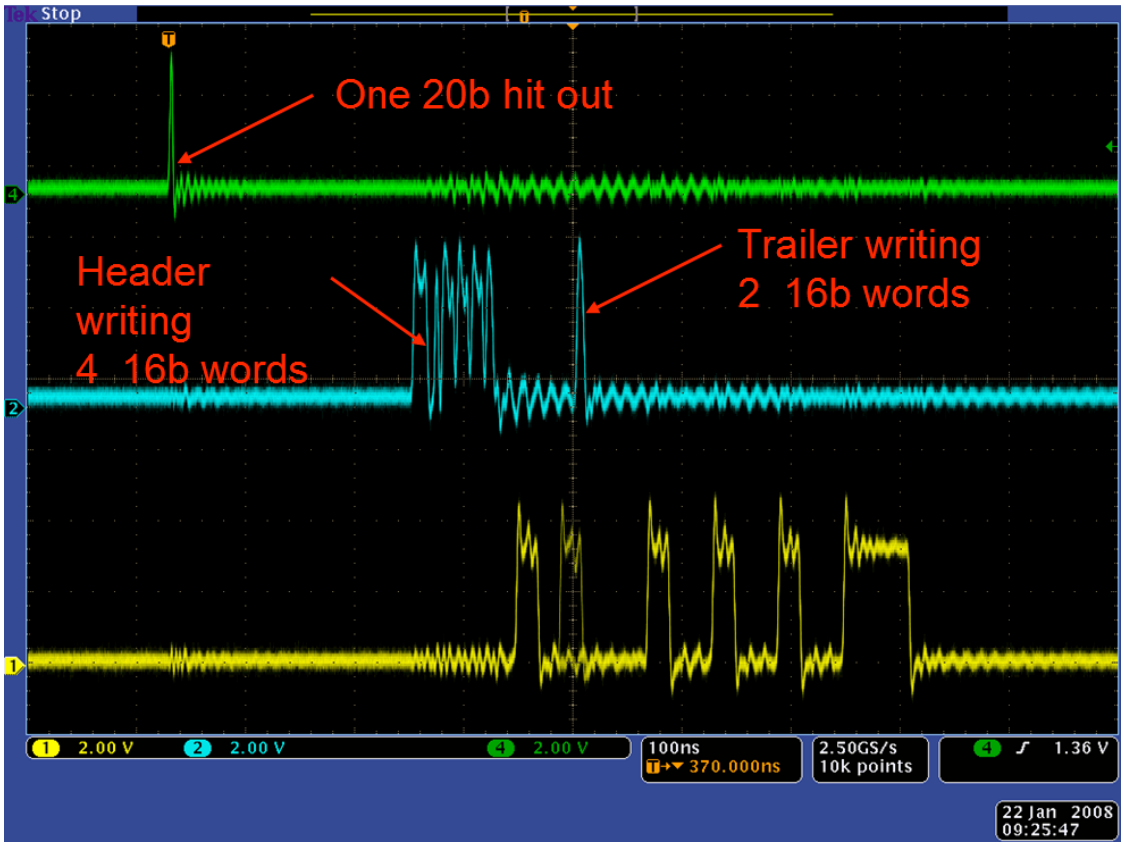


Figure 17 Scope shots of a single packet readout with 1 hit in each of 4 FEM channels

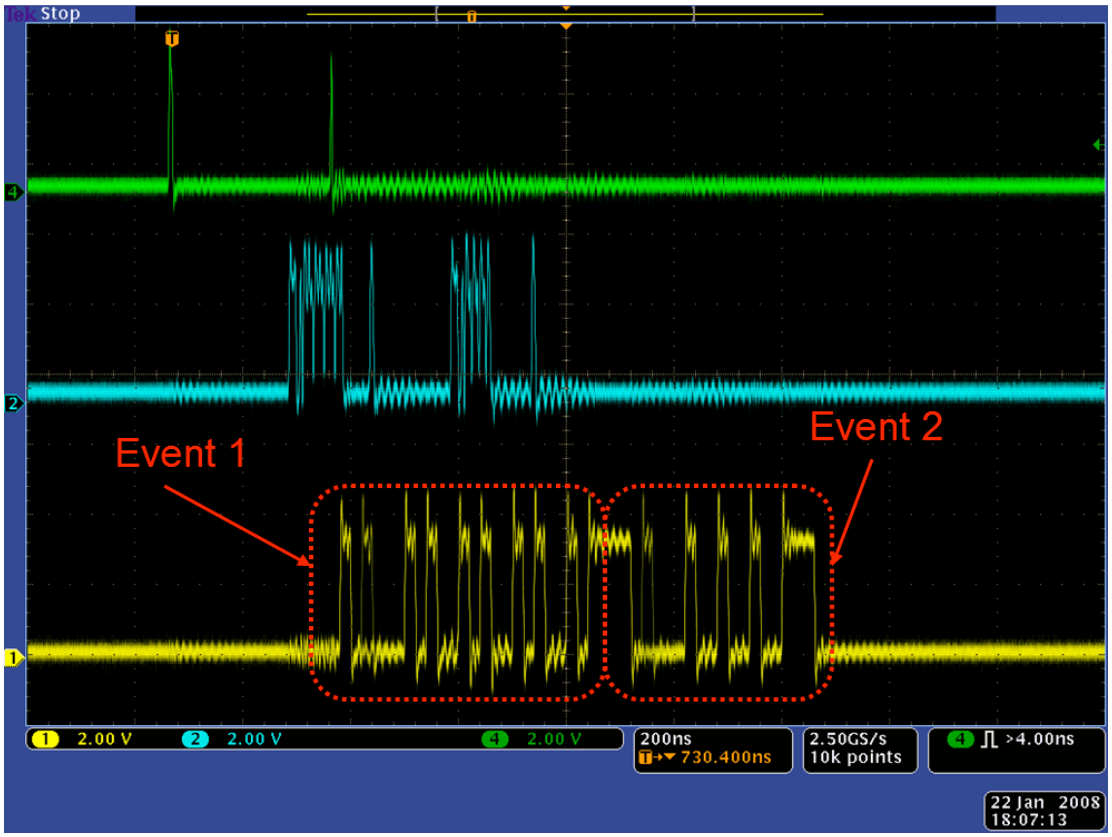


Figure 18 Scope screenshot of a two events being outputted from two consecutive clocks. Green trace show transfer of the data to sub-buffers, blue – write enable for output buffer.

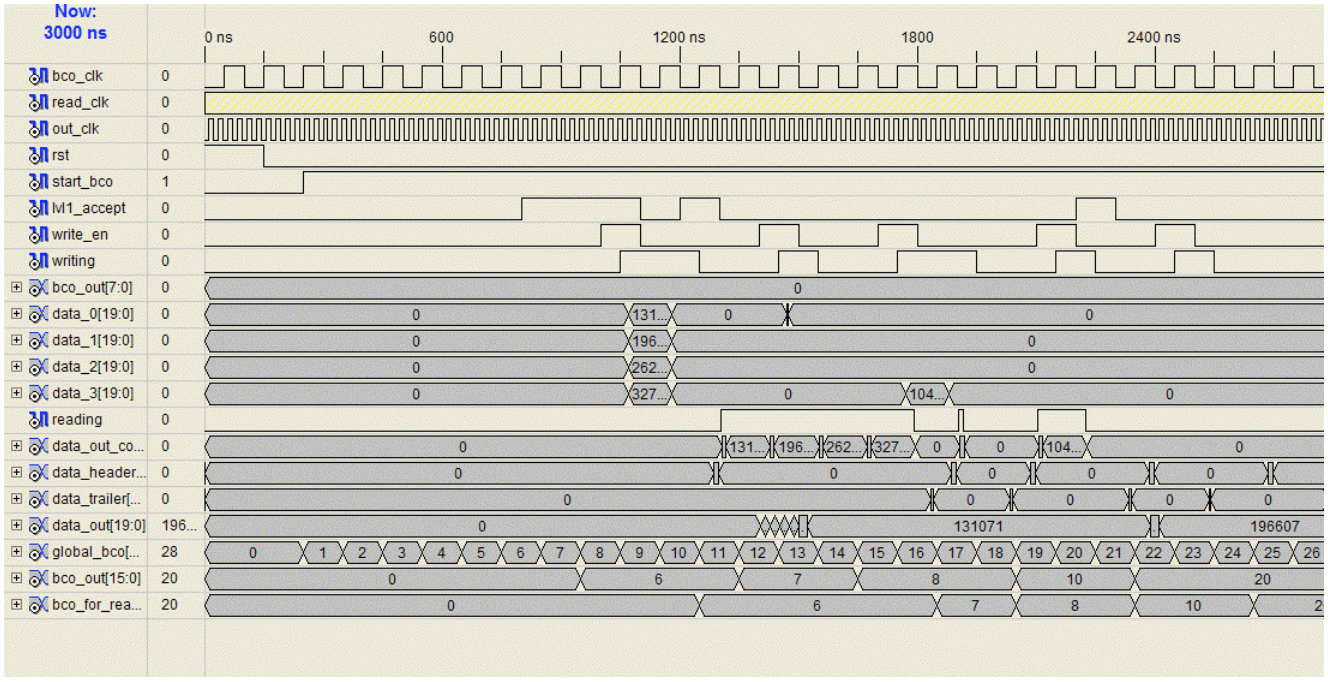


Figure 19 Behavioral simulation of multi event read sequence.

Bit	Fiber data format	Internal hit format		Channel marker format	Output hit format		
0	COL 0	ADC 0		FEM 0	0	1 *	This bit define which
1	COL 1	ADC 1		FEM 1	0	ADC 0	part of the data word is sent
2	COL 2	ADC 2		0	0	ADC 1	
3	COL 3	COL 0	*	0	0	ADC 2	
4	COL 4	COL 1	*	0	1	COL 0 *	Those bits are not = 0
5	BCO 0	COL 2		0	1	COL 1 *	for data
6	BCO 1	COL 3		0	0	COL 2	
7	BCO 2	COL 4		0	0	COL 3	
8	BCO 3	ROW 0		0	0	COL 4	
9	BCO 4	ROW 1		0	0	ROW 0	
10	BCO 5	ROW 2		0	0	ROW 1	
11	BCO 6	ROW 3		0	CHAN 0	ROW 2	
12	BCO 7	ROW 4		0	CHAN 1	ROW 3	
13	ADC 0	ROW 5		0	CHAN 2	ROW 4	
14	Alignment Mark 0	ROW 6		0	CHAN 3	ROW 5	
15	Alignment Mark 1	CHAN 0		0	CHAN 4	ROW 6	
16	ADC 1	CHAN 1					
17	ADC 2	CHAN 2					
18	ROW 0	CHAN 3					
19	ROW 1	CHAN 4					
20	ROW 2	BCO 6	Ocasionally				
21	ROW 3	BCO 7	Ocasionally				
22	ROW 4						
23	ROW 5						
24	ROW 6						
25	CHAN 0						
26	CHAN 1						
27	CHAN 2						
28	CHAN 3						
29	CHAN 4						
30	Allignment Mark 0						
31	Allignment Mark 1						

Figure 20 Data Format for iFVTX detector

Bit	FPHX data format	Fiber data format	Channel marker format	Output hit format		
0	SYNC	SYNC	FEM 0	ADC 0		
1	BCO 0	ROW 0	FEM 1	ADC 1		
2	BCO 1	ROW 1	0	ADC 2		
3	BCO 2	ROW 2	0	CHIP 0	*	CHIP should not be = 0
4	BCO 3	ROW 3	0	CHIP 1	*	to differentiate from Channel Marker
5	BCO 4	ROW 4	0	CHIP 2	*	1 to 52 may be fine
6	BCO 5	ROW 5	0	CHIP 3	*	
7	ROW 0	ROW 6	0	CHIP 4	*	
8	ROW 1	BCO 0	0	CHIP 5	*	
9	ROW 2	BCO 1	0	ROW 0		
10	ROW 3	BCO 2	0	ROW 1		
11	ROW 4	BCO 3	0	ROW 2		
12	ROW 5	BCO 4	0	ROW 3		
13	ROW 6	BCO 5	0	ROW 4		
14	WORD_MARK	Alignment Mark 0	0	ROW 5		
15	ADC 0	Alignment Mark 1	0	ROW 6		
16	ADC 1	SYNC				
17	ADC 2	ADC 0				
18	LAST_WORD	ADC 1				
19	FIFO_FULL	ADC 2				
20		LINE 0				
21		CHIP 0				
22		CHIP 1				
23		CHIP 2				
24		CHIP 3				
25		CHIP 4				
26		CHIP 5				
27		n/a				
28		n/a				
29		n/a				
30		Allignment Mark 0				
31		Allignment Mark 1				

Figure 21 Data Format for FVTX detector