

Muon Endcap Trigger Chambers:

Read Out Driver

Lorne Levinson

Weizmann Institute of Science

TGC Off-Detector Preliminary Design Review

CERN, 16 February 2001

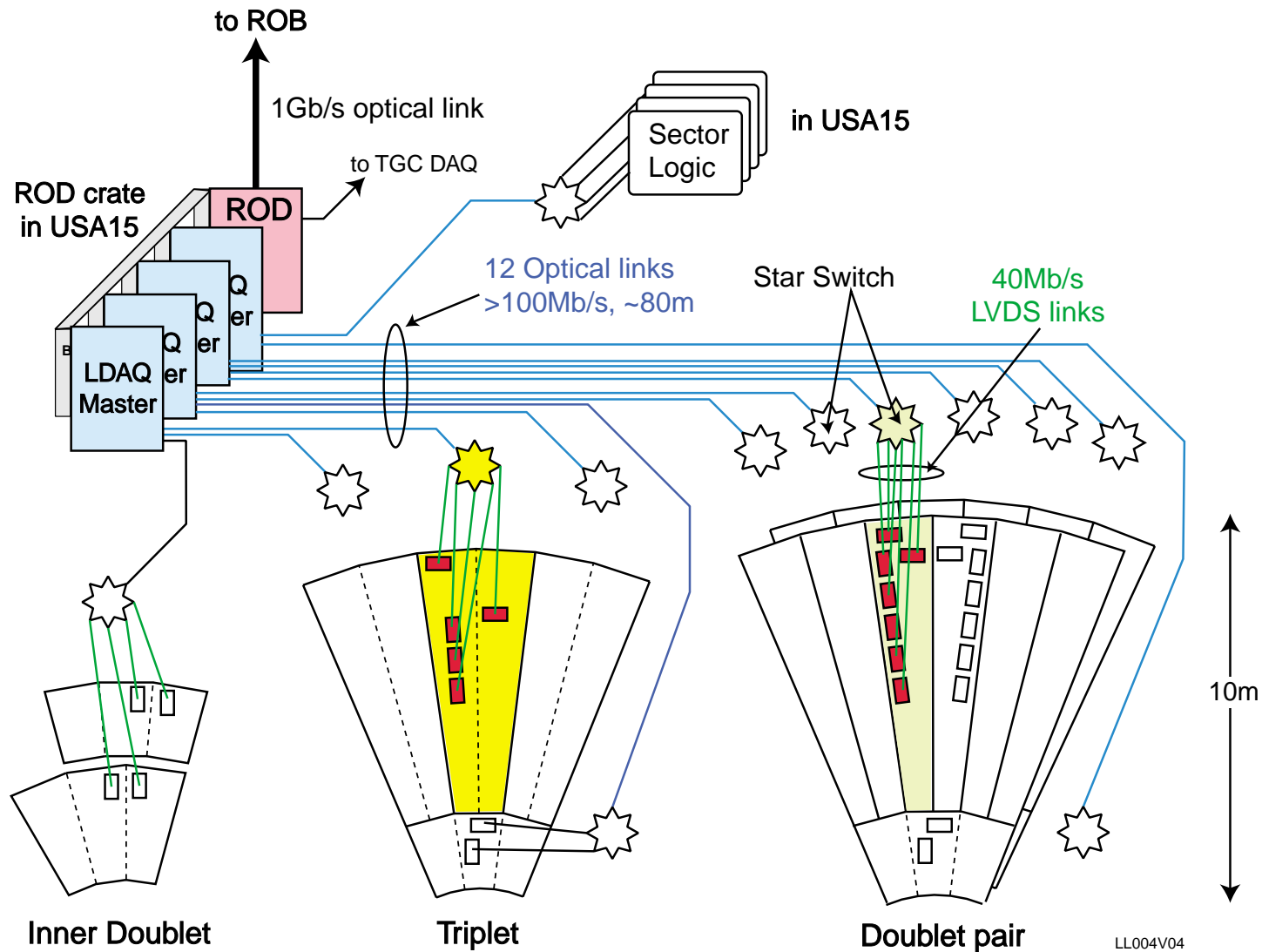
Outline

- Front End Protocol
- Front End Data rates
- Front End link implementation: G-link daughter boards
- ROD
- ROD input buffer simulation
- Status and Schedule

ROD crate

- **ROD:**
reads **N** Front End links, writes to **one** ROB via a Read out Link
 - can be implemented as a set of boards or one board
- in the past we referred to Local DAQ Masters as the front end to the ROD, now just have ROD boards
- a ROD is coupled to a conventional processor, not for the main data flow, but for monitoring, control, configuration, display and serious error handling

Muon Endcap trigger chambers (TGC) readout for 1 of 16 octants



FE link protocol and data format

- one **and only one** ‘event record’ for each Level-1 Accept
 - no other flow control (ROD can assert busy to CTP if buffers fill)
- ATLAS standard link (S-link standard) is planned (~1Gbit/sec)
- data is partially zero-suppressed
 - i.e. only the non-zero 8-bit slices of the 160-bit SB data are sent
 - non-zero suppressed data also can be sent
- option to pass $\pm 1BC$ data
- Slave Board ID is transparent to Star Switch
 - this allows arbitrary wiring from Slave Board to Star Switch to ROD
- simplex (radiation tolerant transmitter only)
- ROD in USA15
- 208 optical links

Data rates

- Data rates from Front end link to ROD crate

Source type	# per octant	# of slave boards	# of chan	Total hits per event	correlated hits per event	Bytes per event	data rate (MB/s)
Doublet F	1	15	1902	0.60	0.12	20.9	2.1
E	6	15	1854	0.78	0.16	21.4	2.1
Triplet F	1	15	1200	0.46	0.10	20.5	2.1
E	3	18	1724	0.69	0.14	21.1	2.1
Inner F&E	1	6	716	0.92	0.13	21.9	2.2
Octant total	12	180	7396	8.73	1.73		25.5

Note multiplicity of each link type

20%

- Hits are uncorellated background from Fluka “AV5” (Ian Dawson Jun-00) plus electronics noise
 - ◆ ×3 since “TP43”
- 6 bytes per hit
- 14 bytes header plus status per event
- 100kHz Level-1 rate

- Data rates (for hits) within the ROD crate

$$\begin{aligned}
 & (8.7 \text{ hits/event } 4\text{bytes/hit} + 32 \text{ header bytes/event}) \times 100\text{Kevents/sec} \\
 & = 3.5\text{MB/sec} + 3.2\text{MB/sec headers} \\
 & = 6.7\text{MB/sec}
 \end{aligned}$$

★ need a safety factor of at least 5 or even 10 for **hits** → 21 to 38MB/sec

Table 2 Format of the event record. It may be preferable to put the missing Slave Board map at the end of the event record. The number of bits in each sub-field is shown in parenthesis.

item		comments	
S-link control word for framing: x'B0F0rrrr' (rrrr is error field, set on xmit to 0). See [ref.4], Sect 5.11			
record type (3)	version (5)	Type: 0: not allowed 1: 0,±1BC data 2: 0 BC data only 3: diag/snap data? 4: unique ID? 6: see extended type in version field 7: not allowed	
b'0000'	LDB ID (4)	LDB ID (=SSW ID) in the octant: 0..13	
map of Slave Boards not responding		24 bit map: 0 means error: Slave Board did not provide data	
map of Slave Boards not responding			
map of Slave Boards not responding			
Data for each Slave Board	b'000'	Slave Board ID (5)	0..17 (numbered only within an LDB)
	BCID hi 8 bits (8)		Bunch ID is 12 bits Level-1 ID is 4 bits
	BCID lo 4 bits (4)	Level-1ID (4)	
	reserved (3)	cell address (5)	0..20
	cell bitmap for central BC (8)		for record type = 1 or 2
	cell bitmap for previous BC (8)		for record type = 1
	cell bitmap for following BC (8)		for record type = 1
	... for n non-zero slices		
	reserved (3)	cell address (5)	0..20
	cell bitmap for central BC (8)		
End-of-Slave-Board marker (8) x'DF'			
another Slave Board	b'000'	Slave Board ID in LDB	
	... rest of data for this Slave Board		
	End-of-Slave-Board marker (8)		
... for m Slave Boards			
another Slave Board	b'000'	Slave Board ID in LDB	
	... rest of data for this Slave Board		
	End-of-Slave-Board marker (8)		
0 to 3 bytes padding, value x'B3', so that the end-of-event marker is aligned on a 4 byte boundary			
end-of- event marker (32) x'FCFCA55A'			
S-link control word for framing: x'E0F0rrrr' (rrrr is error field, set on xmit to 0). See [ref.4], Sect 5.11			

Front end link with G-link

- G-link is gigabit link – overkill
- same technology as trigger links and other links in ATLAS
 - no real saving in choosing something else, opto-transceiver is main cost
- Use daughter board for prototyping
 - reuse if proto-board changes
 - could try other link technologies
- Using new low power CMOS G-link HDMP-1034/1032 (not older 1022/1024)

G-link based daughter boards

- G-link deserializer with opto-transceiver and oscillator
- S-link VHDL 'core' in main FPGA rather than on daughter board
 - can use with only simple VHDL instead of full S-link core
- design copied from one by E. van der Bij with different connector
- standard S-link board, connector and power/ground pins def'n
 - can use instead of S-link board if VHDL core moved to main FPGA
- boards can be cut to smaller size to allow 4 per 6U format
 - boards also used by Mannheim Atlantis Level-2 trigger proto
- same card simplex dst, simplex src, duplex src, duplex dst

ROD – required functions

- Receive TTC information and queue expected event IDs
- Collect event fragments from several Front End links corresponding to each of the Event IDs
- Detect and recover from input link errors and data errors
- Assert RODBUSY to the ATLAS CTP module when necessary, but as infrequently as possible
- Provide hits and tracklets to the ROD Crate Processor for monitoring the data
- Format events into ATLAS standard ROB format
- Send the data to the ROB and/or Rod Crate Processor
- Respond to flow control signals from the ROB
- Requirements for calibration and monitoring as detailed in Use Cases

Error handling

- List of error conditions in document
- Recovery important to reduce data losses due to throwing away events during resynchronization
- First implementation will be less sophisticated in recovery

TGC ROD data flow

- FPGA processor good at handling many small data fragments
 - can parallelize and pipeline
- Pipelined processor, but
 - data dependent latencies for different branches
- prepares monitor data for histogramming on conventional processor
 - only sampling is needed
- flow control from ROB

TGC ROD data flow

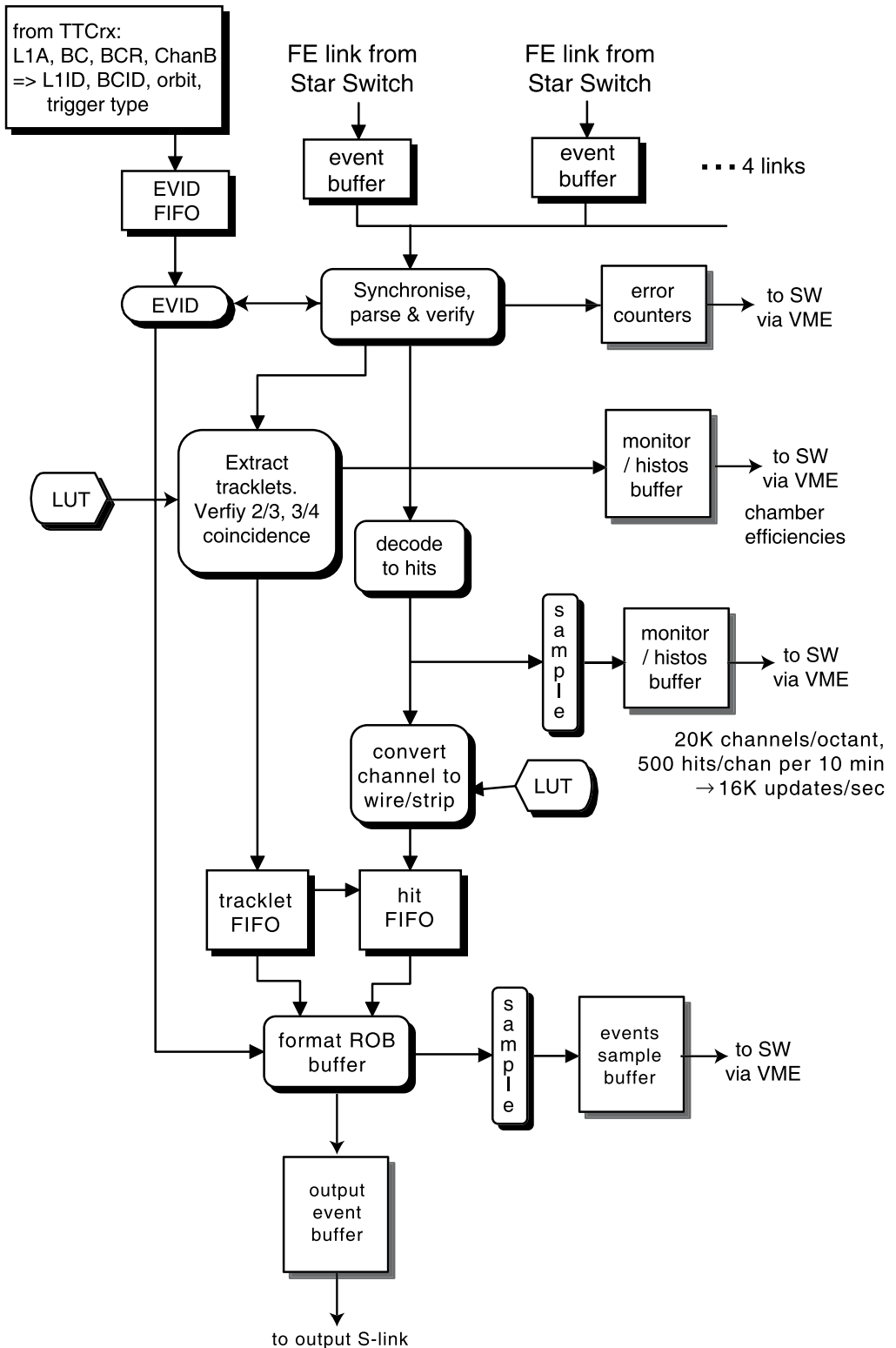
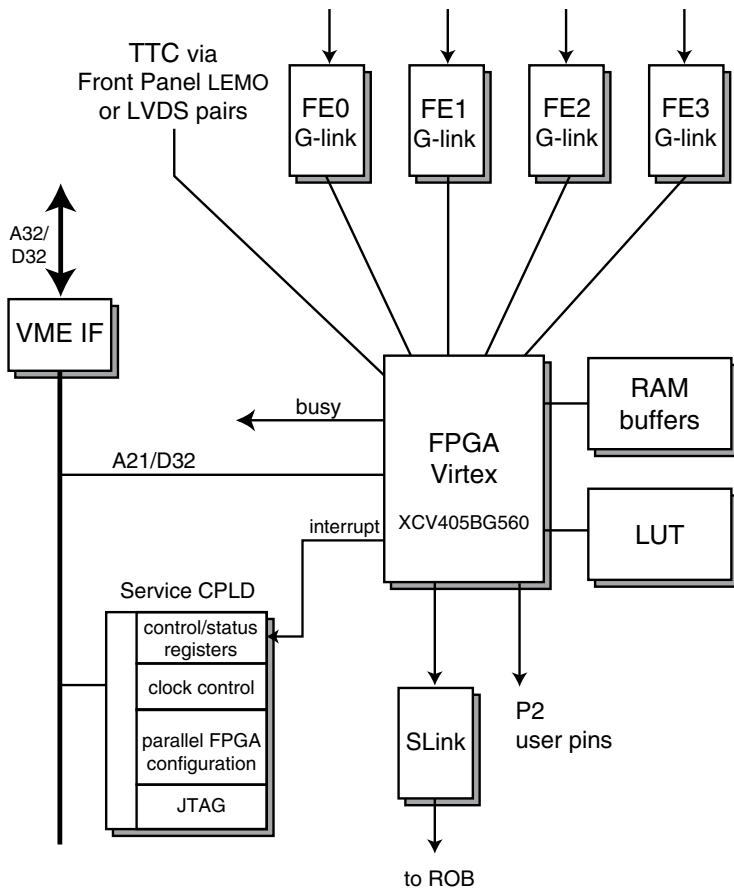


Figure 1 Procedural block diagram of the main data flow. Each rounded rectangle is a process. The processes are pipelined. Each 'cornered' rectangle is a memory element. Not shown, but in addition, there is a path to include the raw data in the output buffer.

Xilinx FPGA – XCV405E

- Large: 10,800 logic cells (1 flip-flop each)
- Memory intensive: 70KB dual ported RAM in 140 configurable (width) blocks distributed over the chip
- Fast: “130 MHz internal performance”
 - “toggle frequency”: 357 to 416MHz
 - 0.18 micron, 6 layer copper interconnect
- Double the memory and double the logic cells available in the same package and pin out
- Memory enables elimination of separate off-chip FIFOs and flexible allocation of memory to different buffers
- 8 DLLs for clock de-skew and management

TGC ROD prototype block diagram



TTC signals

- One TTC partition per Endcap = 8 RODs
- ROD requires L1A, ECR, BC, BCR, ChanB for Trigger Type
 - would also like orbit count reset so we can count orbits from Run start
- ROD does not distribute TTC to Front End
- ROD does not care when TTC signals arrive, as long as there is no skew between them:
 - I.e. as long as BCR, ECR, L1A are associated to the correct BC
- offset of ROD's L1A from BCR is different from Front End's
- for the prototype: TTC signals supplied via (jumper selected)
 - front panel LEMOs: NIM signals
 - BusLVDS flat twisted pairs: signals from a TTC VME module

FPGA internal design

- 5 clock domains connected by FIFOs
- 3 internal busses
- pipeline: variable length data => data dependent latency
 - => FIFOs between stages
 - FIFOs smooth the spikes and resynchronize between different paths
 - should FIFOs fill, the writer's execution is blocked... back up to Front End buffers
- Use FIFO pairs:
 - Data FIFO: contains a variable number of items to be processed followed by an end mark
 - Control Word FIFO: contains fixed length data items whose scope covers a list of items of variable length in the data FIFO

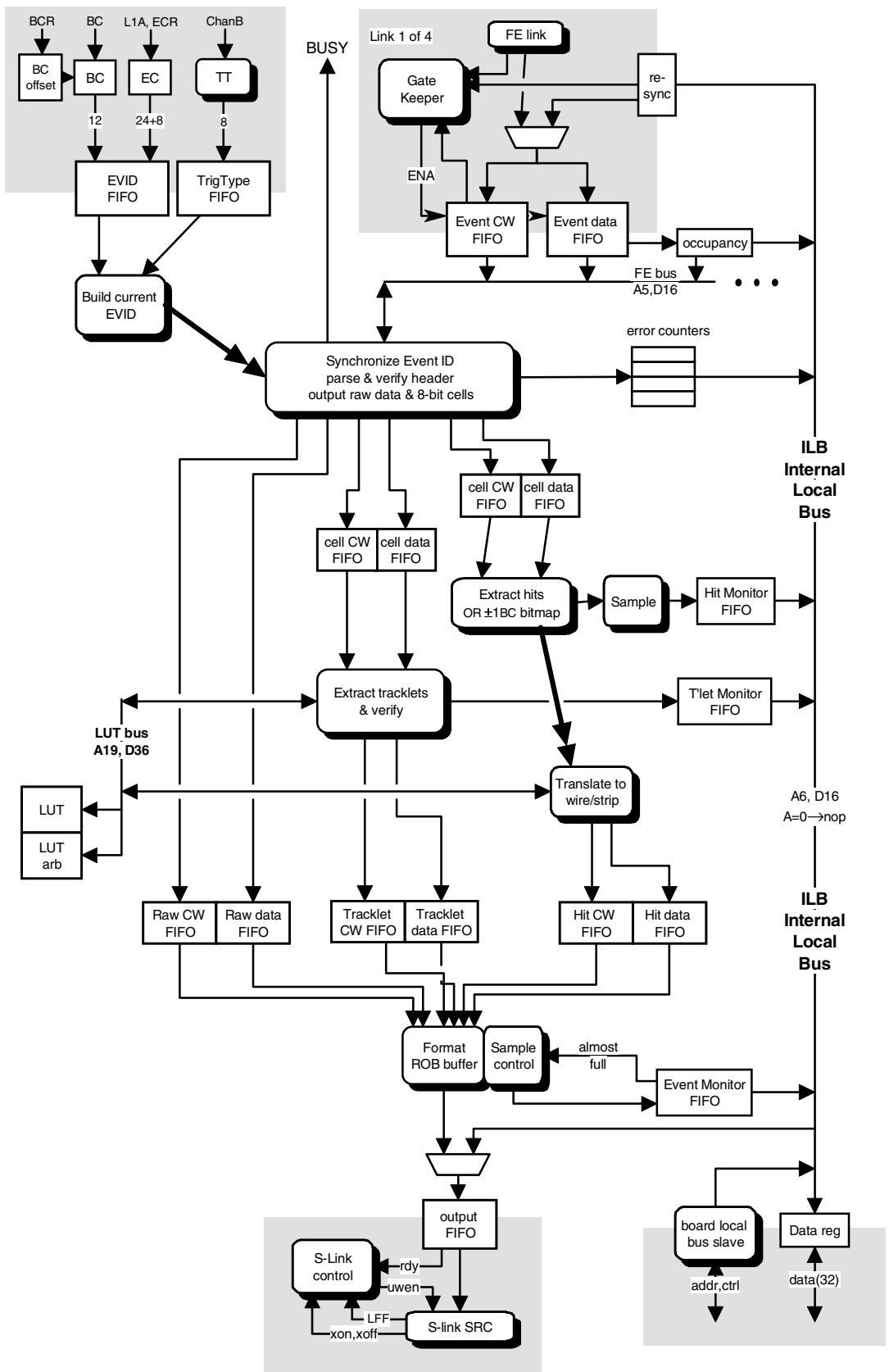


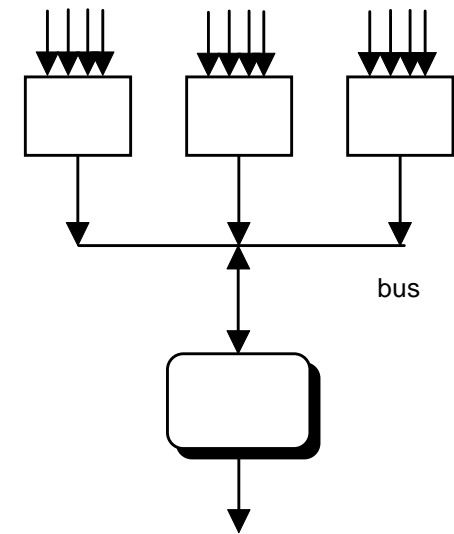
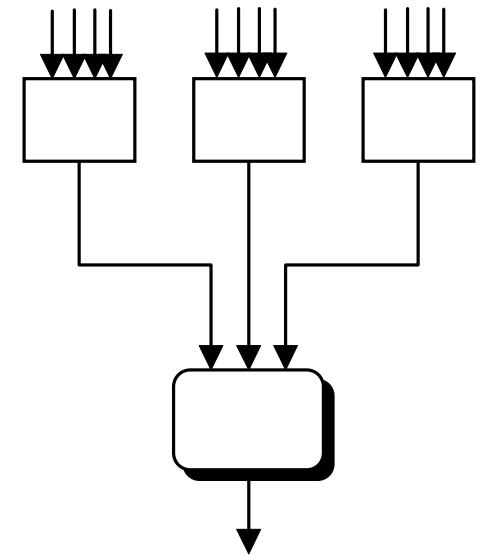
Figure 4 Structural block diagram of the FPGA design. The shaded areas are separate clock domains. Processes (rounded boxes) are connected by FIFOs since they have data dependent latencies.

Debugging, testing, instrumenting facilities

- JTAG boundary scan for PCB test: add connector-to-connector jumpers for generating signals one connector and checking on another
- Output pins for logic analyzer, lots more if output or not all input links are used
- can load test data into EVID, input and output link FIFOs via VME
- TTC signals can be simulated by the FPGA, or sent via front panel NIM signals
- Xilinx readback of all flip-flops and memories via configuration port
- all buffer occupancies can be read directly at any time via VME
- FPGA can be configured for special event sampling based on BC ID, Trigger Types, etc.

Transition to final ROD

- Go from merging 4 to 13 Front End Links
 - bottleneck is probably the merge, but merging must be faster than processing: data copy
- Estimate merge BW at:
 - 7MB hit rate (no safety)
 - plus 25MB raw data rate
- 9U board
 - either option OK
- Multiple 6U boards
 - point-to-point serial link via P2 or front panel
 - P2 bus probably slower than point-to-points
- Tree merging can be adapted to adding more processors with fewer inputs to adapt to higher than expected data rates.

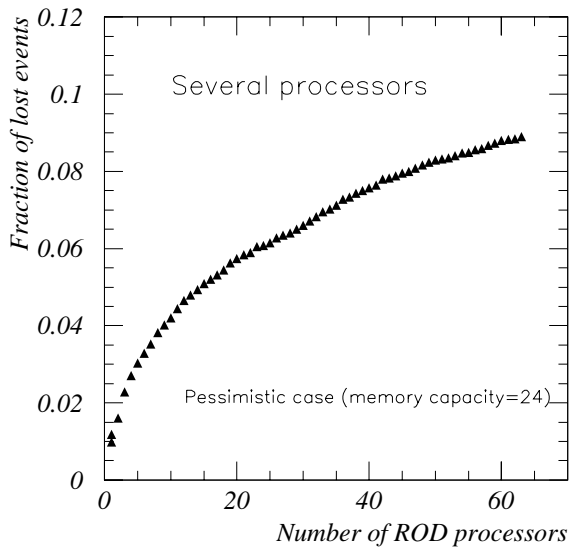
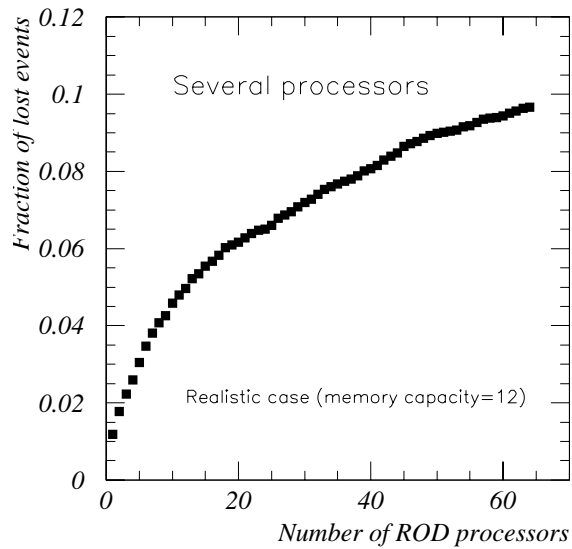
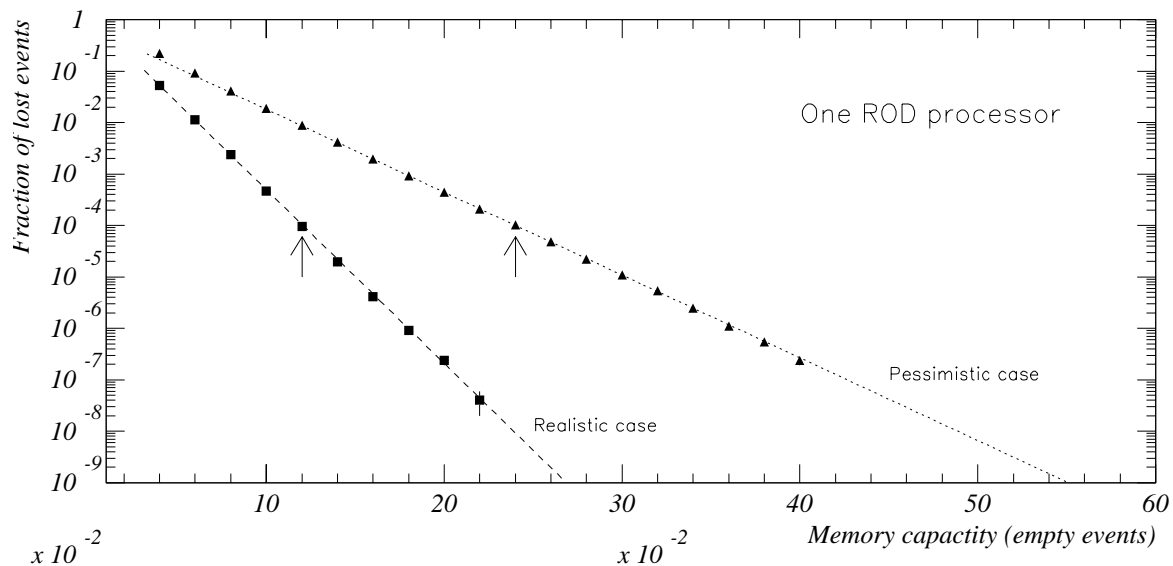


Transition to final ROD – 2

- Planned changes
 - on-board front end links
 - move to VME64 and implement VME protocol in CPLD instead of commercial parts
- Open issues
 - S-link daughter board or S-link core: ATLAS decision
 - on-board TTCrx and its configuration or distribute TTC signals from TTC master board in the crate
 - depends on 9U vs multiple 6U decision

Simulation of ROD input buffering

- Not yet fully realistic but shows that memory required to meet RODBUSY spec is reasonable
- we will add details from measurements with the prototype
 - modeling of the sub-processes and pipelining
 - processing rates for hits and tracklets
 - processing latency
- no ATLAS model yet of what ROB XON, XOFF rate and duration might be
- already gave feedback:
 - to modify the front end data format if no hits for a link:
store BCID and L1ID in header or trailer
 - losses do not scale with number of processors, they are correlated by L1A



Status of prototype

- ROD and G-link daughter board schemas complete
- some trial placements done
- G-link layout begun, ROD layout begins next week
- G-link mechanical details designed, to be confirmed by Erik van der Bij
- all parts in-hand for 3 RODs and 8 G-links
- send board for production ~7 March, back end of March
- HDL underway
 - simple FE link input control, FE FIFOs, occupancy and FE bus done
 - place holders for edge structures (memory address regs, S-link out, Local bus IF) used to check pin outs
 - processing core will start on my return
- Linux driver will start in next 10 days – using Windriver

Schedule

- May 2001: Verifying the connection and data format from the Star Switch, Includes:
 - reading data from one Star Switch, parsing the format and transferring the data to the RCP for display
 - simple VHDL control of G-link, simplex S-link later
 - checking link and transmission errors
- September 2001: Support for the slice test, Adds:
 - reading event fragments from more than one Star Switch
 - building the fragments into an event, providing data for monitoring to the RCP
 - integration with the TTCrx signals
 - simple re-synchronization logic on BCID or L1ID error
 - handling the Sector Logic read-out data
 - proof of principle for tracklet logic

Schedule – 2

- October 2001--Summer 2002: Design and construction of prototype-2
- Spring/Summer 2002: H8 Beam testing, Adds:
 - formatting the output in standard ROB format
 - sending the formatted events via the output S-link
 - full tracklet checking logic
 - extended re-synchronization and error recovery
- Fall 2002 to winter 2003: Final ROD design