
Commissioning L2 Plans: an example

James T. Linnemann
Michigan State University
Seattle Workshop
June 29, 1999

Commissioning L2 (an example)

- Crate Level: hardware checkout
- Crate Level: communication paths
 - monitoring data extraction and display
- fake data: algorithms
 - examine
 - verification against simulation
- Cosmic ray data
- roll-in; beam data

Strategy

- Start with L2 test stand (in FCH)
 - can serve as fake data source possible
color output cables differently
 - can receive real data for algorithm checkout
 - presently: no real SCL connection
need fanned-out, or fake L1SCL
need fake L2SCL generated locally
- Move to Moveable Counting House
 - preprocessor, then global crates
 - single crate, no decision, then full topology

L2 Hardware Checkout: Start with Alpha processor

- download Alpha via Ethernet
 - Alpha becomes VME crate master
- run test programs in Linux for other cards
 - configure/read-back card:
 - FIC, MBT, VBD, Alpha, SLIC (increasing complexity)
 - bit3 needed if card download from TCC/COOR
 - try 1-card self-test: (e.g. L2 cypress loop-back)
or other data source

Communication Paths

- Inputs:
 - data (CIC, G-link, L2 cypress formats)
 - Mbus (MBT broadcast; Alpha-Alpha, Alpha/MBT)
 - TCC/VME
 - SCL
 - individual signals: L1 busy, L2 accept ready, VBD done...
- Outputs:
 - L2 cypress outputs:
 - L1 SCL fanout (SFO)
 - Alpha/MBT Output to L2 Global
 - L2 global trigger mask
 - Alpha scaler gates to HWWF scalers
 - L3 Readout

Fake data sources

- test data locally injected (loopback tests possible)
- test data from fake source
- test data injected into real data source upstream
- calibration data from real data source
- Probably don't need all these, but likely do need one

A source for each link type?

- 20b G-link: many sources
 - download or pseudorandom
 - continuous or burst
 - BERT, test stand fic, test stand GTx, SVX tester?;
- L2 Cypress
 - download or BIST (pseudorandom self-test)
 - continuous or burst
 - can use SFO to fan out to multiple channels
 - G-link source + FIC
 - MBT
 - Alpha MBUS writes to MBT output, or by VME to output FIFO's
 - MBT test card (BIST or sine byte pattern)
- Muon cypress input tester? (required for CIC test)
 - SLIC: L2 cypress+special cabling OK until cabled to CIC
- SCL?? (needed for full MBT testing...)

Needs for 1-card tests:

- Fake data sources for each input type
 - how many channels to be useful?
 - Missing sources: special fpga programs?
Source control (cvs) for fpga programs
- local host to run test software
- lots of software: test programs
 - start from most basic operations
 - proceed to continues test
 - cvs...

Test program list...

- One for each card type at least
 - variant for each data source needed
 - which live in Alpha/Linux, vs live in TCC (NT)
 - to run test programs: is Linux always there?
 - Self-test suite for SLIC DSP's?
normally no user console????
- Multicard tests:
 - Test programs for worker/admin alphas?
Or just special version of administrator
 - test programs for SLIC DSP'S as well as alphas

Communication Path Checks

- BIST Installation checks simpler for L2 Cypress
 - can use source in same crate
 - put source, receiver in BIST mode
 - check receiver for BIST errors
- G-link, or realistic Cypress data harder:
 - sender often in another crate
 - use L2 test-stand sources if real source missing
 - TCC or higher level coordination to set up test
 - similar or worse for in-situ diagnostics (no rewiring!)

Ready to Run?

- Other inputs, outputs:
 - L1 SCL input
 - (L2Global: Answer to framework)
 - L2 SCL Accept/Reject; buffer management
 - L3 readout
 - Hook up Alpha monitoring scalers
framework monitoring software needed
- full download for parameters, programs...

Multiple crates strategy

- Preprocessor alone first
 - run parasite mode (no readout)
 - special framework setup to transport L1SCL info?
 - single alpha worker, then multiple workers
 - Run with readout to L3 (all L2 accepts for now...)
- Global alone
 - verify decision link to HWFW
 - verify L2 SCL accept/reject back
 - L2 rejects is new issue with L3 readout buffers
- Preprocessor + Global
 - all accepts, then accept/reject

Crate Checkout with Data fake or cosmic or real

- monitoring extraction and display
 - much more functioning software needed
- error messages
 - collection, distribution (with monitoring)
 - flow control
 - download limiting parameters
 - summaries
- L3 readout opens new possibilities:
 - stare at examine
 - fake data: should know outcome!
 - Fancier analysis offline
 - Verification of online result against simulation

Multiple Crate Hazards

- COOR:
 - multiple trigger scripts not independent
 - special issues in L2:
 - require SAME inputs EVERY event (MBT)
 - L2 preprocessor inputs tied to l3 readout of source crates
 - probably will force all readout lists to be same
- L2 crashes stop data taking
 - new algorithm checkout in test stand
 - MAYBE debugging also in test stand (data copies)
 - also hoping for event dump/playback

Verification: Online vs Simulation

- configure simulation to match online setup
 - database extraction of any script information needed
 - matching of releases of code, constants: version control
 - **Caution: separate jobs for L1, L2pre_i, L2G, L3**
your input from raw data not upstream simulation's output
- comparison reporting: a process running on simulation output
 - find matching objects (generic code from L2 I/O code generator?)
 - report and summarize mismatches
- eventually: automation
 - trigger script setup to generate monitor stream (haggle for bandwidth)
 - script run to verification automatically
 - planning for prompt analysis and reports (personnel)
notification and followup!
The normal output should have no discrepancies.....

More Tasks

- Monitoring data content design
 - see that things are healthy
 - localize faults when things die
- Monitoring software:
 - reading (L2 I/O code generator?)
 - display (decide behavior; use Taka's widgets)
- online examine
- simulation good enough for verification
- data vs simulation comparison program
- automation and follow-up planning

Algorithm Commissioning

- Algorithm Group Interaction
 - goes without saying? How about without doing?
 - code reading
 - Release control
 - regression testing in releases
- Calibration/Alignment
 - what precision needed to run?
 - What data needed to get it
 - how to extract the constants
- Algorithm certification
 - comparison with offline
 - MC (develop software, form group)
 - data (showtime)
 - turnon curves

Commissioning Workshop This Fall

- commissioner for each subsystem
 - come up with a plan
 - what you need to do
 - when you plan to do it
 - get what's needed to install and commission
 - installation technician time
 - special hardware
 - software
 - people
 - time (can you work in parallel with others?)
 - cosmic triggers what can you learn?
 - Beam triggers for checkout
 - offline analysis group(s)

Monitoring Data Collection

- TCC tries to collect fresh monitoring every 5 s
 - sends event through system with CollectStatus
 - reads hardware scalers in L1 framework
 - if L2 running, captures info and tags with event #
put in Bit3 MPM in each crate by Administrator
read out by TCC (after .5 s?) (no VME cycles in crate)
 - if L2 not running, TCC asks Administrators to collect monitoring data anyway?
Mark as stale unavailable blocks (crashed node?)

TCC & Monitoring Client

- TCC serves trigger monitoring information
 - handles scaler rollover
 - may be asked for integral or differential
- client handles display and reporting
 - reports, pacmen, flow displays, strip charts, histograms
 - decides what to do if not all monitoring fragments match event tags on blocks differ:
 - stale information in some blocks?
 - Show previous block (and warn “stale data”)
 - or make blank display?
 - Collected without events flowing, so not synchronized?
 - needs to know (fixed) format, trigger setup (both: database?)
 - presents subsets according to requested run (not all bits) and run summary?

Monitoring info database: many views needed

- Lots of info of various types
 - type of synchronization
 - time, event, rough interval
 - source
 - alpha, mbt, fic, slic, framework scaler
 - sources have different availability if things hang
 - interpretation
 - event (try, pass...) counts
 - error counters Integration time > 5 sec? By Client?
 - time in state
 - current state
 - buffer occupancy histogram
 - circular buffers of event characteristics (histogrammable)
 - error messages

Info Type vs Source

	FW Scaler	VME	Processor
Event Counts	x	x	x
State Occupancy	x	x	(x)
Current State		?	?
Buffer Occupancy	x	x	
Circular Buff/Histo			x

Sources of Monitoring Data

- **Scalers in L1 framework**
 - ECL gates, count beam crossings in given state
 - count of occurrences of a condition
 - Read by TCC directly--indep. of state of L2 crates
 - tells truth even if system hung
 - time-synchronous collection
- **Scalers on VME boards (MBT, FIC, SLIC)**
 - like L1 scalers but local, slower clocks: time in state
 - not collected at same time as L1 scalers
nor event-synchronized
 - collection works only if
Administrator alive (unless TCC knows crate contents!)
VME available (VME bus, VBD not hung)
smoothest if events actually flowing

Monitoring from Processors

Alphas, DSP's

- Can monitor `most anything we have time for
- best for event accounting, pass rates
 - if events flowing, collection is event-synchronous with L1 scalers
 - error condition when
 - any in/out counts mismatch
 - nonzero error counter
 - not time synchronized with L1 scalers
- if events not flowing, may be partial or clumsy
 - can Administrator read Workers, DSP's?
 - usually, they are notified to collect and move to a buffer

Whole Monitoring Catalog

- Now writing down all we intend to collect
- framework scaler count
- sizing of multiport memory of Bit3
- specifications of hardware cards
 - testing against debugging scenarios
- draft of monitoring collection class for Alphas

Monitoring Displays

- Propose drafts to Online group
 - someone working on this! GREAT!
- Performance monitoring displays
 - warn shifters when things go wrong
- Diagnosis displays
 - localize problems (scenarios)
 - hope to diagnose without sending commands...
 - couple to control system for reset????
- Expert displays
 - performance tuning
 - advanced diagnosis

Monitoring Reports

- Quantitative info shift monitoring
 - when visual doesn't work?
- Comparison with expected behavior
- Run summaries: capture run start values
 - TCC access by actual value, by time difference over n periods
 - who integrates scalers which roll over
 - VME scalers on boards: monitoring, not accounting
- Hope: Intermediate run summaries
 - pick **integral since run start**, or from “now”

Monitoring concerns

- Performance Monitoring and **Tuning**:
 - find bottlenecks, while data **flow** continues
 - where, why events backing? Need **averages, distributions**
 - average buffer occupancy
 - processing time
 - transfer time
 - deadtime= fraction of time buffers full
- Debugging of **hangups** (no data flow; **static** system)
 - eventually **FIX**, not diagnose!
 - time in state goes to 100% (anyone in a weird state?)
 - buffer occupancy stops changing (where the events? 16?)
 - identify card with problem
 - identify channel of card if isolated
 - **snapshots, error counts** may be enough--if they can be read!

Desirable compromise [like L3 Pacmen]

- Display *averages*
 - good for tuning/performance
 - *click to see details*
 - usually for experts
 - examples: buffer occupancy histograms; error counts
- Display becomes static on hangup
 - IF info can be read: so *prefer FW scalers*
 - but it's only available for Alphas
 - VME local scalers are nearly as accessible
 - for cards with buffers

Combine like cards (MBT's, SLIC's, FIC's)

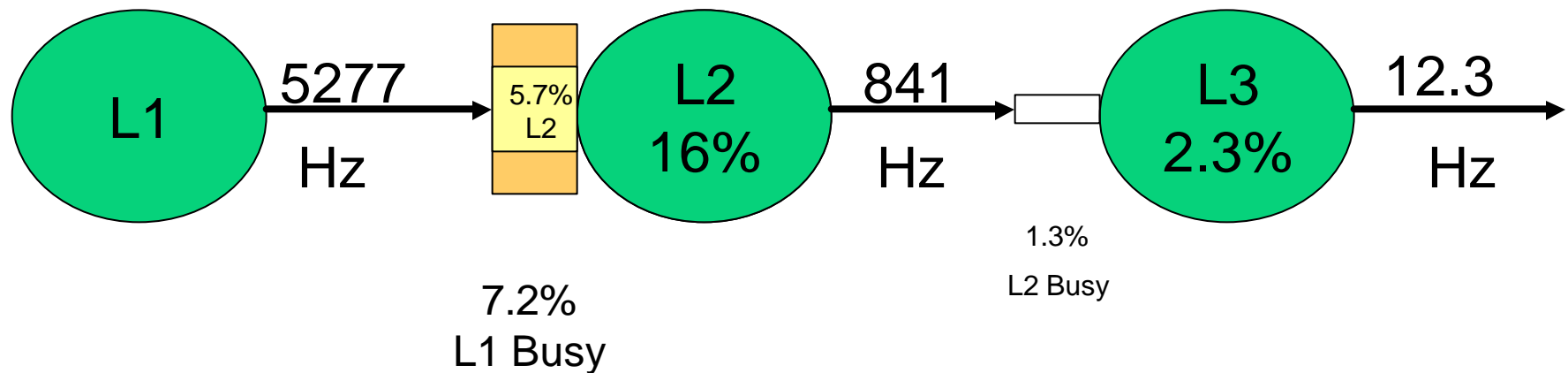
Performance Monitoring

- Baseline: “always there”
 - get to know normal behavior of system
 - Relatively fast response (few sec)
- Rates
- Rejection or pass fraction
 - on request, Rates, pass fraction for each bit
- Deadtime
- strip chart of some particular bit rates vs time

Trigger Framework Itself

- Beam crossings
 - = Time (allows conversion of counts to rates)
 - normalization if not all alternatives histo'd
- # L1 accepts
- # L2 accepts
- L1 Busy fraction (from SVX, L2) (“deadtime”)
- L2 Busy fraction (caused by L3)
- Histogram of # L2 decisions pending
 - fraction time with 0-16
(occupancy of FE “L1” buffers)

Highest Level Display



+ strip charts, of these + some L1, L2 bit rates

High Level Display: Notes

Data sources in L1, L2 are FW scalers; L3

L1 busy is total deadtime “L2” = % time 16 buffers

The busy boxes expand and contract logarithmically

For L2, L3 circles, % given = pass fraction

L2 Busy box also expands and contracts
fraction of time L2Busy raised

L1, L2, L3 turn different colors (or **BOLD borders**) if > threshold for:

errors recorded > threshold (what memory time scale...)

click to get error statistics, error messages, scl_init logs

rate, pass fraction out of tolerance (?)

very hard to set tolerances other than deadtime

L2 FW Scalers

- ~400 total, 32 bits each
- ~5 per Worker Alpha
 - time in states only, but most interesting states after all Administrator isn't doing the work
- ~32 per Administrator Alpha
 - time in states
 - time with given buffer occupancy in crate Alphas (Admin knows) where whole events likely collect partial events probably in MBT's; DMA into Alpha
- handy hardware window during commissioning!

Worker Alpha

FW Scalers: Time in State

- Things you want to time, or possible hangs
 - Wait/event
 - Processing
 - **Writing Output ?**
 - Interrupt
 - Waiting/Administrator Reply
 - Collecting Monitoring
- A concern: overhead to advertise states
 - ~.3 sec for PCI write

sometimes can pay overhead during wait states

Administrator Alpha

FW Scalers

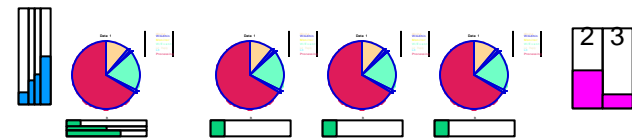
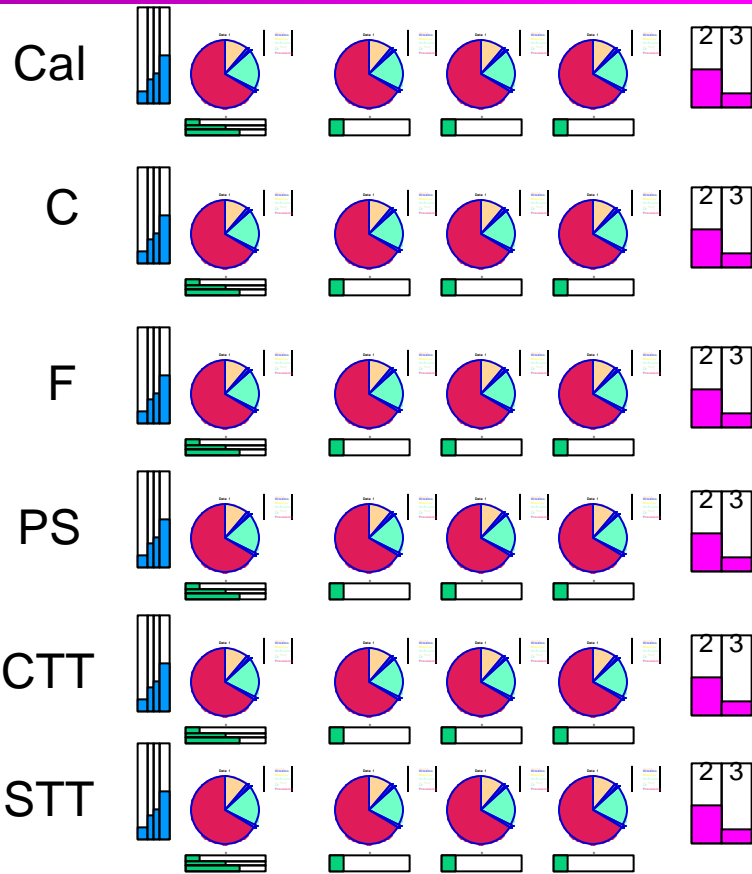
- Time in State:

- Wait/event
- Processing
- Interrupt
- Reply/worker
- Handle L2 Accept/Reject
- Managing L3 readout
- Collecting Monitoring
- VME Busy?
- Mbus Busy?

- Buffer Histograms (# bins)

- Allocated (4)
- Processing (8)
- Wait/L2 Acc/Rej (3)
- L3 Readout (3)
- Free (4)
- ToAllocate (1/Worker)

Main L2 Display



L2 Global



L2 Display Views?

- Select whether to show:
 - Workers, Admin/buffers, MBTs, SLICs FICs
- Workers: most basic info
 - others needed for debugging

Processor Displays

- Pacman: pie chart of time in state
 - idle time at 3 o'clock
- Bars underneath (for non-states)
 - Interrupt routine (if not subtracted from states)
 - VME Bus Busy; MBus Busy
- lump SLIC processors? (avg time in state?)
- Buffer before: <events> Processing Queue
- Buffers after: <events> Wait L2 answer, L3 R/O
- ERRORS: **BOLD** border if error counts > threshold
 - click for details?
 - Sum (or average) for longer than 5 sec!!

Buffer Displays

- Average Occupancy (scale 0-16)
 - based on histogram and boundary values
who calculates mean? Admin? TCC? Client?
- MBT split: min and max
- MBT Special: if $\langle \text{min} \rangle = 0$ and $\langle \text{max} \rangle = 16$
 - display number of channel(s) with 0 complete events: likely bad channel
- Lump like cards together (MBT's, FIC's)
 - max = max of max's, min = min of min's
- Useful options?
 - % time in highest occupancy state?

Diagnostic Displays

- Where to look next if things go wrong
 - can I get the info when things are messed up?
 - Hope from basic displays...
- Try some scenarios:
 - heavy SVX deadtime
 - slow preprocessor
 - crashed preprocessor
 - cypress input link down
 - G-link input down
 - L3 output hung
 - cypress input link slow

Scenarios: Deadtime diagnosis

- SVX deadtime: main L1/L2/L3 display
 - total deadtime \gg % L2 buffers full
- Slow L2 processor: main L2 display
 - small L2 idle fraction
 - details for processing time, distribution
- Full buffers mean processor is slow
- Empty buffers mean processor fast, OR source slow

Hangs: processors, L3

Crashed or hung L2 processor: 100% dead;
typically all L2 processors but one in IDLE state
clear unless happened in IDLE state

search processors: “data stale” in details

requires acquisition of details without data flow

L3 output hung: processors view same as L2 hang

need buffers view:

output buffers to L3 full

Hangs: Cypress link

All nodes idle, 100% dead

global MBT: max 16, min 0, empty link #

name if clicked (from database!)

if broken between preprocessor and global, that's all

if preprocessor with broken input:

ALSO its MBT shows this condition

Requires:

Admin extracts info WITHOUT event flow

MBT register: mask of 1 or more complete events

error counters vs channel MAY help

Hangs: G-link

- Looks like preprocessor cypress down
- but FIC looks empty
 - no full event, no data in output FIFO
- FIC in error state? display bad channel #
 - G-link RDY OFF during Reframe/L1 Busy?
 - Error counters nonzero?
 remember beyond 5 sec!
 - register for PARTIAL events? Or does it help?

Control and Recovery

- Can hangs be recognized in a single crate?
 - hung preprocessor link? (Global looks bad, too)
- SCL_INITIALIZE
 - should it reset links automatically?? Too slow?
- Need uniformity of responses to resets:
 - power, front panel
 - VME write to reset register
 - = power-up, or less violent ?
 - keep FPGA programming? Controls settings?
 - VME SYSRESET: VME bus release only
 - MBRESET: MBus release only

Expert Displays/Reports

- Detailed performance measurement
 - diagnosis of slow processor
 - where to expend optimization effort
 - trigger scripts, processor code...
 - comparison with queuing simulations
 - mean and rms processing times; distributions
 - per event only: do per-filter, per-tool level in simulator
 - # bits/script
 - # tools run/script
 - overlap between trigger bits (need special output?)
 - average latency
 - % capacity used
 - distribution of buffer occupancy

Issues: Alpha

- Need to measure overheads for
 - setting state for FW scalers
 - for entry in circular buffer
- Like to measure
 - interrupt handling time
 - not one of a set of exclusive states, unless stack...
 - VME and MBus busy
 - like hardware path to ECL outputs

Circular Buffers: Flexible Processor Output

- Scalers can give fractions
 - with time base, can give mean times
- Circular buffers of actual values: $(T_{\text{event}}, N_{\text{jet}}, \text{jet}, \dots)$
 - Software logic analyzer (Cutts): {state, transition time}
 - make a histogram to show a distribution (tails)
 - linear, log plot...
 - calculate rms values
 - only window to high pre-L2 statistics ($\sim N_c \times$ more than UBS)
 - trigger-independent “health” plots for preprocessors
 - L1 trigger masks (for trigger overlaps--pseudo data stream?)
 - show derived quantities
 - event latencies: $T(\text{decision}) - T(\text{event})$ [L2G Admin]
 - $\langle \text{Latency} \rangle \times \langle \text{Input Hz} \rangle$
 - = effective delay in events (buffers)

UBS vs Monitoring statistics

- Monitoring defined as ~ 1 per 5 sec
 - Circular buffer with N_c event entries gives $N_c/5$ Hz
- UBS events:
 - .5% of out bandwidth
another .5% L3 UBS
 - Say 20 Hz output
1/2 % is 1/10 sec UBS from L2
- Monitoring is $\sim 2N_c$ more--for **SELECTED** info
 - UBS events get WHOLE event--vastly more flexible
- need to be a monitoring client to see it
- must do own histo clearing at “run” start/end
- save to disk for offline analysis??

Histograms vs Circular Buffer

Histogram: concerns over software escalation

- another monitoring data type
 - more bins to define and match
 - less stable than buffer bins
- + scatterplot may be more natural?
- + Circular buffer gives less statistics? (size choice)
- ? Relative timing of two

attraction: Run I Trigger examine was weak statistics, strong dependence on event selection
less bias, but still L1 selected, from preprocessor

Issues: MBT

- Measuring min, max # events histogram
 - also specific channel histogram, but must be set...
- Display: “average” min, “average” max
- bad link gives min=0, max=16
 - enough to isolate if hung link on card
- need **register mask of FIFO states** (for active inputs)
 - 1 if any data? No. Useful only if hangs between evts
 - **1 if complete event?** Bad channel shows as 0.

Issues: FIC

- Not event-oriented, so should self-drain
- only purpose is to show if
 - getting behind (unlikely...)
 - **input lost** (unlikely--until it happens)
- Histos of # whole events (time fractions)
- could use a register like proposed for MBT
 - currently, have FIFO_FULL
 - Has **error counter**, **RDY**, per Glink
 - sufficient when no data flowing to identify channel?
 - Does **BUSY** because of reframe show up here?

Issues: SLIC

- No spec yet
- Event oriented: between MBT and Alpha
- error counters, event counters per input
 - also a **register** for fragments?
- Event occupancy **histogram**?
 - Of what buffer?
Input FIFO? Fast. Should self-drain--but event oriented?
FIFO in front of DSP?