

Comments on STT architecture proposal

J. Linnemann

9/22/99

Architecture

It might be good to talk with Rob Martin (UIC) and even perhaps or U of M collaborators (Myron Campbell, and Stephen Miller) who are building the Alpha board about their thoughts about how having 3 PCI buses. Can provide phone numbers.

Careful to preserve adequate board space for the TFC cards!

How much of John's direction/input buffering/coordinate normalizing function can be on the motherboard rather than the processing mezanines?

Just to be sure— do you want your readout by VBD in the crate, or to spend more real estate by having a completely standard VRB crate elsewhere do the buffering, with you firing off all the data you generate unconditionally on G-link? Might simplify the FRC and get you out of bus control, and windowing requirements on the dualport memories. If you stay with VBD, hope Hal can steal logic from the VRB controller. Tradeoff: must build a G-link driver (maybe useable elsewhere?)

I guess the SCL receiver slot is dedicated, as the inputs are different... seems a pity to waste real estate on other cards.

Inputs

VTM occupies whole space in back, so without redesign of an existing card, little or no opportunity for any more I/O on the back. Rest of L2 has the non G-link inputs on front, as well, for better or worse. Redoing the VTM engineering wasn't attractive to Saclay. Doing so would allow many fewer connections per input cardlet, and likely give up the PCI standard advantages.

Can you put all G-link infrastructure on the motherboard? Road data comes in 20 user-bit (24 transport-bit) format, with 16 bit data and 2 control bits (begin/end event: FIC input format). It contains a header/trailer structure including event number information. SMT data I understand less well, but I *believe* it comes in 16 user-bit format (20 transport-bit), has only in-band (special word) event end marker, no begin-event marker, and may come gray-coded, which implies further processing. Maybe different FPGA codes with (rather) different state machines do the trick?

If a VTM input is bad, need to kill the whole bad object (16 bits of SMT address/data, or the 32 bits of CFT road), looking at the various error signals the G-link produces. A of synch loss makes the FIC's declare end of event and send a L1Busy via the SCL. This attempts a re-synch rather than triggering a full SCL initialize.

Manuel's argument for LVDS input error *correction* seems very ambitious to me, as I think it means buffering a full event and analyzing it before passing it along to any circuitry.

Is there an event structure in the input FIFO's? I assume that event structure must be imposed/recognized once you're on the STC cards.

FRC: remember that SCL inputs (esp. from L2 accept/reject) can come at more or less any time, including overlapping with SMT data input. Puts some nontrivial buffering and processing requirements on FRC.

Outputs

My preferences (details below):

- 1) TFC outputs daisy-chained (LVDS?) so 1 Cypress output per crate.
- 2) ZVC also Cypress output (surely enough if just sending histograms to sum)

Less work to have the outputs for ZVC and TFC same; G-link output for both acceptable, if only one output each per crate. xxxxxnot really— where to put FIC's

If Z really needed G-link, that might push TFC to G-link. This would likely mean building some more FIC's to receive them. G-link out causes more space problems if not daisy-chained than Cypress does.

TFC outputs (and ZVC?): one output to L2CTT per crate, or one per card? Life is (mostly) easier in the receiving crate if one output per crate, but that means daisy chaining (maybe same mechanism as you already foresee for inputs?).

Why? There are only slots for 4 FICs in the L2CTT crate. FIC's can accept 4 inputs each, so run out of slots at 16 total inputs, perhaps including (at least temporarily) the present 4 CTT. So after 2 G-link outputs per crate, a problem arises. Ugly for diagnosis purposes to put FIC's in other crates, and all such crates are on a different MCH floor. Many of those slots are already set up for data copying to the test stand (see below).

Another issue is the ability to fan out signals for the test stand; obviously, more output signals means more fanout real estate (6 inputs per SFO slot) in the receiving crate. These presently compete for slots the FICs would use. Most (but not all) other crates are able to send copies of all their inputs to the test stand. A

smaller issue is receiving the eventual hot-link signals; another MBT is required for each 7 signals. These would need to be budgeted for. So, I'd have a prejudice towards trying to make both Z and TFC outputs cypress, possibly fanned out on the source cards. It would sure be nice if they TFC and Z cards each daisy-chained their outputs so there was only one output channel per crate. That raises other software and header issues: does each independently make a header/trailer around its results? How does the software like receiving multiple blocks on the same input? But I think these are resolvable.

Some of the same issues exist with a Z system. What's the destination? If your goal is Z for L3, it could be aimed at running only in case of L2 accept. If for L2, then the target could be Global if the number of inputs is not too high, and the amount of work there is small (say summing up histograms); otherwise a crate with MBT's/Alphas might be a candidate. I'm assuming that the hits themselves are sent by the STC's, not the Z's.

Monitoring and testing new downloads

Please carefully consider downloading and monitoring paths and especially their implications for software. Maximizing sharing of facilities with rest of L2, or at least with L1DFE, will save you a very considerable software effort. A Bit3 gives you access to TCC, while a CPU/1553 puts you on a different path. I don't know which is better, though I do know that the CPU/1553 means it will differ more from the rest of L2, making this part of the system harder for (say) a DAQ expert to understand. L2G and L2CTT will do little looking at distributions...

The basic monitoring we try to do on other parts of the system include buffer occupancy and processor state. We usually monitor buffer occupancy with asynchronous counters measuring the fraction of time N buffers are used; we try to think in terms of events, since that's where the hard boundary is, determined by the front ends (our buffers have enough bytes to hold largest events). Processor state we monitor by fraction of time processing, idle, waiting for some handshake, etc. We also try to monitor fraction of time VME is busy. Obviously, different FIFO's need different degrees of monitoring. If the input cardlets hold TFC's main buffers, they need to be capable of serious self-monitoring, as do the L3 output buffers.

Test stand: one thing to think about is whether you want some sort of facility like the L2 test stand where you get live copies of data, but can check out new downloads or debug while the main system is running. I think it's not quite as compelling if you don't have computers, which can crash, but there might be arguments. With your shorter range for LVDS, that might mean your test rig has to be in the MCH rather than FCH. Is there anything useful you could send at

Cypress speeds? Does the topology allow you to make 2 copies of something by enabling two targets, or configuring the small transmitter cards?

An operational question having nothing to do with electronics design (probably):
Where is the initial (x,y) of the vertex for this machine fill found? By the TFC?
By the L2CTT? By L3? By an offline process? How is it downloaded back to
whoever needs it (L2CTT? TFC?).