FRL design issues

- FRL and crates
  - How many FRLs per crate
  - Address space issues
  - Where to put the DAQ PC
- sTTS interface
  - Operation with or without sTTS
- FRL in debug (stand-alone) mode
- Functionalities of the FRL
- Some technical issues
  - FPGA configuration

FRL and crates: Presentation of Dominique

- Problem: address space shortage in compact-PCI-crate
  - NIC needs 64MB (Lanai9) ==> FRL will require 128 MB (27 bits)

    HOW MUCH WILL NEED THE LAN_ai10 ?

  - standard PC has 4 GB address space
  - **? IF ?** 2 GB are available for PCI ==> 16 FRL are addressable with one PC
- Possible Solutions

  1) Accept to have 512 / 16 = 32 DAQ PCs
     - Advantage:Standard solution, straight forward addressing, no “bricolage” necessary, One PC needs to control less FRLs ==> faster control loops (FPGA config!)
     - Disadvantage: More PCs needed, Does not work if LAN_ai10 needs more address space

  2) Variable Address Window
     PCI bridge makes only one part at a time of total Address space visible. Which window is visible is programmed by host in a register of bridge.
     - Advantage: Smaller Addresspace ==> less PCs needed

- Disdadvantage: needs “page switching” to program : slower, logically more complicated, the internal PCI bus is not directly visible (e.g. NIC does not exist in PC as PCI card connected to system: Can this cause problems?)

3) Activate only one FRL at a time
needs to make use of the user defined lines in compact PCI.

- Advantage: Address space of ONE FRL at a time is taken, only small overhead in software, all can be master at any time, trivial to change to option 4) or 1)
- Disadvantage: needs jumper to work in normal compact PCI environment

4) Buy a PC which is capable of 64 bit addressing (alpha?...)

- Advantage: No address space problem, no software change
- Disadvantage: Not yet standard (but: today 1GB of memory is “normal” ==> current “standard” address space of a PC starts to become small, in some years 64 bit addressing is very probable to be standard)

- Problem: Where to put the PCs
  - Can we take “standard PCs” ?
    We can reuse PCs we are currently accumulating
  - IF PCs must go into rack: Is this ok with the cooling? (Airflow up - down is blocked)

FRL with or without sTTS interface (discussion with Per)

- With sTTS: Who could use this?
  - FRL itself
  - RUI could send via Myrinet a message ===> triggers FBI to write into a FRL register ===> triggers sTTS message

Useful to send "Out of synch" (and may be "fatal error")
Who detects Out of synch ?

- FED: If it has data sources connected to different TTCrx chips. Could do internal cross checks?
- FRL: No merger: Can check for EvtNumber(n+1) = (EvtNumber(n) || 0)
  Merger: Can check EvtNumber(chan1) = EvtNumber(chan2)
  BCNumber(chan1) = BCNumber(chan 2)
- RUI: Can check EvtNumber and BCNumber of incoming fragments
- BU: Can check consistency for the entire event

In addition

- FRLcould check the SLINK CRC and issue fatal error in case of permanent (repetitive) failure

- Without sTTS
  - FRL can change status bits in data stream => might need an extra word
  - DAQ PC could detect status bits in FRL registers and issue ?aTTS? messages (long latency)
  - RUI / RU can issue aTTS messages

Decision on implementing or NOT implementing depends on how ofter errors are expected which are not already detected in FED:

seldom => aTTS is sufficient

often => sTTS is necessary

FRL in Debug (stand-alone) mode (discussion with Frans)

- Asynchronous Mode
  - event generator in FRL size options: random size, fixed size, lookup table
    data options: random, pattern, from internal memory
  - event rate could be programmable
  - FRLs would run NON synchronously
- Synchronous mode
  - use user lines on backplane for: Trigger, backpressure
  - 1 slot taken by a GIII which control these lines
  - could even implement a TTCrx receiver (Claude's card) to achieve synchronization over manu cartes

  Is this useful ?

FRL functionalities

- Status in FRL crate
  - user defined backplane could be used to make a wired OR of status / error bits (see Dominique's presentation last week)
  - Possibility to make a "LAM" (look at me) line
  - a "Master-FRL" (same card, different firmware) could be receiver of these lines and main Interface to DAQ PC:
    - DAQ PC asks Master-FRL for crate status (avoids polling all FRLs)
    - Only Master-FRL contains sTTS interface, use user backplane to "OR" the sTTS lines
- Data consistency checks
  - only point to check SLINK CRC in hardware:
    On failure: status bit
    On repetitive failure (algorithm in FPGA) :
    issue sTTS message if present
    set internal status bit
    sends error status on user-defined lines
  - Synchronization:
    On failure: status bit and sTTS if available
    one input mode: check event number

merger mode : check BC and event number

- event length
  could be checked here since SLINK Control word flags are available
  if the Filter the first who uses this event length
  on failure: status bit

- Where to put status information in the data stream
  - Trailer of SLINK: This means that they must not be used by FED
  - Other possibility: add a trailer word in FRL

Spy Buffer

- Algorithms to put event in spy-buffer:
  - Downscaled events

  Algorithms which require the whole event to have passed
  - If specific data consistency check failed
  - If specific event fragment status bits are set
  - If the Monitoring flag is set (does it still exist?)

  Implementation:
  - write event always into memory using a write pointer;
  - by default event n+1 overwrites event n (reset write pointer);
  - if spying condition is met do not overwrite (= do not reset write pointer);

Table synchronization (Frans)

- Save mechanism: (not relying on any latencies)
  - two tables can be loaded, one is active
  - during operation: load inactive table
  - stop triggers
  - retrieve last triggered event number
  - check in each FRL if last event number has passed, if so change tables
  - reactivate trigger (may be reset event number before)

Further functionalities

- SLINK test
  - Is it possible to start a test in the sender card from the FRL (without intervention of some FED software and assuming that the FED is powered) ?
- Masking
  - Send empty events in case of certain error conditions?
  - Cut to maximal event fragment length (adjusting the trailer)

Technical issues

- FPGA configuration
  - Configuring a whole crate takes a long time (approx. 60s x N(FPGA)) with simple JTAG interface a la GIII)
  - Do we need to be able to put more than one firmware into some non volatile memory on board to switch between Debug - Data taking mode?