WIB Planning

Major Near-term Tasks:

- Distribution and Development of firmware for ICEBERG
- Discussion and Fleshing out of DUNE WIB Requirements

WIB Requirements

- Presentation by Marco (7/22) on DUNE-specific issues:
  - Cleanliness underground (for entire WIEC)
  - Temperature monitoring
  - WIB and CISC interface
  - Data monitoring (?)
  - Data processing (?) [Move DAQ TP generation to WIB…?]
  - Interaction with Run Control---does data format need to be adjusted?
    - Particularly an issue for scalability
    - And even more so for strict “downtime” requirement
  - Flexibility in firmware/software design to anticipate evolution
  - Long-term reliability and failure rates

Let’s focus today’s discussion on the DUNE WIB (not necessarily what is being built for ICEBERG)

WIB Requirements

- Presentation by JRK/Van Berg (7/29) on High-Level Formal Requirements
  - Baseline Functionality
    - Receive data from CE
    - Transmit data to DAQ
    - Receive clock, timing, and calibration commands
    - Distribute timing and synchronization to CE
  - [Logical] Interfaces
    - Configuration by Run Control/CISC
    - Dynamic reconfigurability during run, including re-synchronization
    - Availability of state to DAQ and CISC via register reads and status bits
    - Interlocks for safety of personnel and equipment
    - Monitoring information to DAQ CCM
  - "Good design" requirements
    - Maintainability for at least 6000 APA-years (about x1000 more than ProtoDUNE)
    - MTBF > 20 year lifetime
    - As simple and inexpensive as possible, commensurate with other requirements
    - As flexible as possible, given other requirements

Missing?: Deadtime, uptime, latency, throughput, cold starts,testing, commissioning…

WIB Requirements

Need to flesh these out further…And perhaps break into logical pieces:

DAQ Side:

- Physical and logical interfaces defined in Interface Doc *EDMS id 2088713*
- Not defined is explicit interaction with Run Control or CISC
- Or resynchronization of WIBs/FEMBs

  "The CE and DAQ consortia will work together to define how the WIBs and RunControl interact when including/excluding individual WIBs (or FEMBs) into/from an ongoing run and how the WIBs will be synchronized. "
- What errors and how they are handled
- Commissioning and testing

CE Side:

- Physical interfaces defined (10 twinax, 2 twisted pairs)
- Logical interface depends on COLDATA/CRYO down-select?
  - For COLDATA, detailed interface document already written (I2C, Fast Commands)
- Deadtime == 0
- Uptime: top-level is >98% full across modules; 90% for one module

Internal:

- Error generation and handling
- Channel re-mapping

WIB Requirements

"Use case": Resynchronization

- Data is 100% synchronous---how do we know timestamps are reliable?
- (8-bit counter rolls over every 128 μs << 1 drift time(!?))
- Time stamps can be missing or just wrong
  - What is requirement on how long we can allow the former to go on?
  - Is the latter a requirement for the WIB and CE, or DAQ, or calibrations…?
- Does resynchronization get initiated by DAQ, in response to WIB error?
  - ProtoDUNE Timing System checks every 16-bit rollover
- How is synchronization checked across WIBs, APAs, modules?
  - Is it a CE requirement for inter-WIB synchronization (requirements doc not clear: "10 ns across channels")

Path Forward

- Need to flesh out additional requirements (see Volodya's talk)
- May want to break up discussion by category (DAQ interface, Internal, CE…)
- Re-start of conversations with DAQ consortium (probably next Tuesday)
- Do these need to be formalized in requirements spreadsheet/doc?

Timeline: Full list by end of calendar year

Backups---slides from 7/29 discussion of requirements

ProtoDUNE WIB→ DUNE WIB

Already heard about this from Marco last week, but just a summary:

- Performed very well on ProtoDUNE
- Critical functionality demonstrated
- Integrated exposure is about 6 APA-years
- Two SP modules will integrate 300 WIB-years/year
- New front-end will need new design elements and/or new WIB
- Added capabilities (e.g. monitoring) worth some thought
- Should consider also opportunities (e.g, more channels/WIB)

Requirements Definition

From DocDB #7626-v5

Requirement: A statement of what the experiment/detector/system/subsystem/component needs to do to enable DUNE to meet its science objectives. Anything tagged as a requirement for a given subsystem or component is to be fulfilled by that same subsystem or component. (E.g., in the Subsystem A requirement set, each requirement begins with: "Subsystem A shall …") In general, requirements will not contain numerical values; they prescribe or constrain specifications and design choices from which design parameters are chosen.

- In practice, for the kinds of things we do, the line between "requirements" and "specifications" can be murky
- Neither DUNE nor various consortia have been particularly consistent

Requirements

“Baseline Functionality” (i.e., why we have a WIB)

1. Accept data from the cold electronics in the logical format and the physical interface defined by the cold electronics.

2. Transmit data from the WIEC to the DAQ system in the (negotiated) logical format and over the (negotiated) physical interface – this includes all required re-ordering of data and provision of makers, error flags, status flags, timing information and any other ancillary information required in the interface document.

3. Accept timing, synchronization and calibration information from the Timing System over the (negotiated) physical interface in the (negotiated) format.

4. Provide timing and synchronization signals to the cold electronics in the logical format and physical interface defined by the cold electronics.

Requirements

Necessary Interactions

5. Accept configuration information for the WIB and/or the cold electronics from the DAQ / Slow Controls system(s) in the (negotiated) logical format and over the (negotiated) physical interface.

6. Allow dynamic reconfigurability by RC during a “run”, including resynchronization

7. Allow Slow Controls and/or DAQ to interrogate any or all registers or status bits as needed over the agreed upon interface channel.

8. Provide hardware interlock(s) for any potential dangers to equipment or personnel.

9. Provide monitoring information to Slow Controls (DAQ?) to enable alarms for any potential dangers to equipment or personnel.

Requirements

Good Design Requirements

10. Ensure maintainability for anticipated exposure of 6000 APA-years.
11. Ensure a MTBF that is greater than the expected 20-year+ lifetime of the experiment.
12. Be as simple and inexpensive as possible while satisfying all of the above.
13. Allow for whatever level of possible additional operational flexibility that will still satisfy all of the above

Comments

- Have not said anything about channels/WIB
- Have not included (yet) power requirements
- Have not included (yet) online monitoring (beyond CISC)
- Not included “debugging/diagnostic” path
  - Very useful, particularly in beginning
  - Would be nicer if this was integrated into primary firmware
  - So perhaps make parametrically configurable