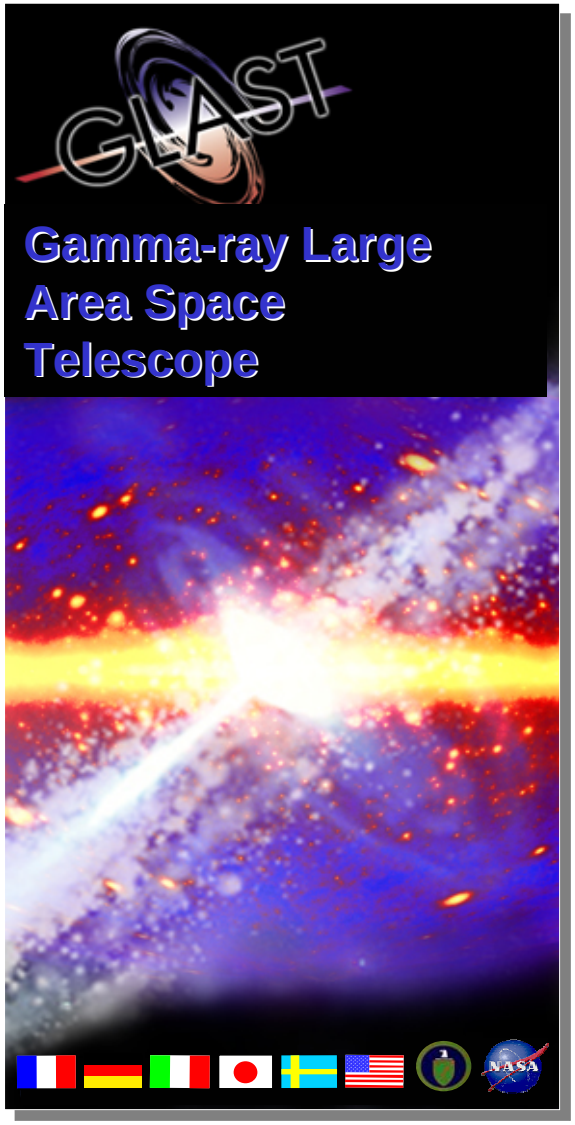

The Test Bed and The Front End Simulator (FES)

The
What,
Where,
Who, and
Status

J. Russell, A. Waite, O.Saxton, M.McDougald, E.Siskind, J.Wallace

Test Bed

- What is it?
  - Test bed is a full scale mock-up of the LAT
  - Includes almost everything but the sensors
    - TEMs
    - GASU
    - PDU
    - 5 PCI crates (although maybe only 3 will be used), complete with CPUs and SIBs
- Where is it?
  - Room B101, Central Lab Annex at SLAC

- What's it for?
  - Testing of electronics and FSW
- Who are the players?
  - O.Saxton - software
  - E. Siskind - hardware

Front End Simulator

- What does it do?
  - Simulates the input to the LAT at the cable level, inputs are:
    - TEM
      - TKR – 8 cables/tower
      - CAL – 4 cables/tower
    - ACD – 12 cables period
    - GASU – well, a whole lot
      - 16, one from each TEM (3 signals)
      - 12, consisting of 18 veto signals + 1 CNO each from the ACD
  - Without it the Test Bed is not worth much
- What does it consists of?
  - 10 COTs PCs running VxWorks
    - Each PC has
      - 2.4GHz P4
      - 120 Gbytes local disk
      - 512 Mbytes memory
    - 8 PCs handle 2 'TEMs' each => 1 TKR board + 1 CAL board
      - » 2 PCs handle ACD + control software => 4 ACD + 1 synch board(s)
  - Sized to handle 1 orbit's worth of data at nominal rate

FES Challenges

- Challenges
  - Build it quickly for a reasonable cost in $$'s and manpower
    - If it is late, it is useless
  - Bandwidth
    - Challenge simulating 1M parallel input channels
  - Reasonably high fidelity

- In response to this
  - Chose the cable level of simulation
    - Cabling is only in the 100s…
  - Leave front-end testing to others
  - Limited simulation of front-end register
    - Static properties of the registers are simulated
      - Reads/writes over the command fabric work
    - Dynamic properties are dicier.
      - Depends on how much CPU power is available to implement on the fly transformations
  - Limited testing of real-life effects like pile-up

FES, What's it good for?

- The poster-child application is
  - Loading GlastSim events onto the PC's disk
    - driving them through the FES and the LAT to EPUs
      - where the filter does it thing
  - IE, it is the acid performance of the complete LAT T&DF system including
    - The triggering system
    - The data flow (bandwidth)
    - Integrity of transfers
- Of course there are lots more mundane things like
    - Shipping test patterns
    - Measuring bandwidths at strategic points, etc

FES, How does it work?

- CPU transfers data to large memories (640Mbytes) on the electronics cards
  - Source is data stored locally on the disk
  - 3 pieces of information need to be formed
    - Detector data
      - In information content, what comes down the cable
    - Trigger data
      - To be shipped to the GASU
    - Transition vectors
      - Controls when the events are shipped
- On a given transition
  - data is held in the memory for a given time
  - If a trigger comes, the data is strobed down the cables,
    - Hey, just like the real thing, if the trigger is late, you miss the data…
- Typical event consists of two transitions
  - One containing the data
  - One returning the detector to the 'quiet' state.

FES, Status

- Current Status
  - 1 CPU now running
    - Can load test patterns into both the TKR and CAL
    - Currently the two boards are running unsynchronized
  - Transitions of up to 160KHz have been driven
  - ACD board design complete

- Next Steps
  - Multiple CPUs
  - Get ACD board running
  - Run the system as a whole
  - Build Test Pattern Library
    - Classic test patterns
    - GlastSim Events
  - Given its general purpose nature,
    - Use your imagination