EM1 EGSE Configuration

- Subsystem (or simulator)
- AEM/TEM
- DC Power
- Back door debug RS-232
- LAT-COMM
- VME/cPCI Chassis
- CPU
- Online Workstation
- Ethernet Router and Firewall
- Main Dataflow Ethernet
- Ethernet
- Back door debug RS-232
- Online Workstation
- Central Database

Subsystem/simulator
Electronics/Online

WAN
Hardware hierarchy

LAT

TEM

CC

RC

AEM

RC

GASU

GEM

GLT

FE
Software hierarchy

- GTEM
  - GCCC
    - registers
    - GCRC
      - registers
      - GCFE
        - registers
  - GTIC
  - GTCC
    - GTRC
      - GTFE

Tower Electronics Module

- GAEM
  - GARC
    - registers
    - GAF
    - GGEM
      - GGLT
        - registers

AntiCoincidence Detector Electronics Module

Global Trigger Electronics Module
## Quantities (1)

<table>
<thead>
<tr>
<th>TEM Functional Block</th>
<th>Total Number per TEM</th>
<th>Number of Registers per Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>GTEM  Tower Electronics Module</td>
<td>1</td>
<td>6 x 32-bits</td>
</tr>
<tr>
<td>GCCC  Calorimeter Cable Controller</td>
<td>4</td>
<td>7 x 32-bits</td>
</tr>
<tr>
<td>GCRD  Calorimeter Readout Controller</td>
<td>16 = 4 x 4</td>
<td>8 x 16-bits</td>
</tr>
<tr>
<td>GCFE  Calorimeter Front-End ASIC</td>
<td>192 = 16 x 12</td>
<td>3 x 16-bits</td>
</tr>
<tr>
<td>GTCC  Tracker Cable Controller</td>
<td>8</td>
<td>6 x 32-bits</td>
</tr>
<tr>
<td>GTRC  Tracker Readout Controller</td>
<td>72 = 8 x 9</td>
<td>2 x 64-bits</td>
</tr>
<tr>
<td>GTFE  Tracker Front-End ASIC</td>
<td>1728 = 72 x 24</td>
<td>5 x 64-bits</td>
</tr>
<tr>
<td>GTIC  Trigger Interface Controller</td>
<td>1</td>
<td>18 x 32-bits</td>
</tr>
</tbody>
</table>

**Totals:** ~2000 ~10000
# Quantities (2)

<table>
<thead>
<tr>
<th>AEM Functional Block</th>
<th>Total Number per AEM</th>
<th>Number of Registers per Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAEM ACD Electronics Module</td>
<td>1</td>
<td>5 x 32-bits</td>
</tr>
<tr>
<td>GARC Calorimeter Readout Controller</td>
<td>12</td>
<td>43 x 16-bits</td>
</tr>
<tr>
<td>GAFE Calorimeter Front-End ASIC</td>
<td>216 = 18 x 12</td>
<td>11 x 16-bits</td>
</tr>
</tbody>
</table>

Totals: ~200 ~3000

1 LAT = 16 TEMs + 1 AEM + 1 GEM

= ~160000 + ~3000 + few registers

= ~165000 registers
Nodes and attributes

- **Gnode**: Base class describing a major block in the LAT hierarchy
  - Subclassed to create GLAT, GTEM, GCCC, GCFE, etc. nodes
- **Gattr**: Base class describing attributes of a Gnode
  - Examples are registers, dataless commands, etc.
  - Some Gattrs can have an associated constraint, rule and or raw/engineering unit conversion class
Constraints, Rules and EGU

- **Gconstraint**: Base class for describing a constraint
  - Constraints are evaluated when attempting to *write* a mnemonic
  - Two types of constraint subclasses are currently supplied
    - **GconstraintLimitRE**: RE => Raises Exception when attempt is made to violate limits.
    - **GconstraintLimit**: Pegs value to limits when attempt is made to violate limits. No indication given to caller that violation attempt was made.

- **Grule**: Base class for describing a rule
  - Rules are evaluated when a mnemonic is *read*
  - One example rule subclass is currently supplied
    - **GRuleLimit**: When value is outside limits, a message is printed

- **GEGU**: Base class for converting between raw and engineering units
  - One example subclass is currently supplied
    - **GEGU_linear**: Provides linear conversions
  - Could be used to convert between values and state names
Nodes and Attributes

- **Set()** evaluates constraint in *engineering* units
- **Set()** then converts to raw units before loading the value on the hardware
- **Get()** reads the hardware and converts the raw value to engineering units
- **Get()** then evaluates rule in *engineering* units
- If no EGU is defined, raw and engineering units are the same
Cautions

- Beware of naming inconsistencies
  - GlimitConstraint should be GconstraintLimit, etc.
  - We’ll try to get these fixed as we go along
- Beware of constraints and rules
  - These are evaluated directly in mnemonic accessors
  - Don’t provide one if it is not needed
  - Ensure that evaluation is success oriented and fast
    - Violations should not happen frequently and are thus not as CPU time critical
    - Minimize memory usage
    - Attempt to be generic so it can be reused with multiple attribute instances
- Beware of rounding issues with GEGUs
  - Constraints and rules are evaluated using engineering units
Software Architecture

>>> test()

DB server

Observer

App

DB repeater

DB server

...
Migration

• EM-1: No FSW beyond hardware drivers
  – Will be used for the EM test (cosmics, Van de Graaff photons)

• EM-2: Development platform for multi-tower support
  – Embedded systems run FSW code
  – Commanding will be done through a more realistic dictionary
  – Event format changes from TEM/AEM output style
  – Test bench scripts will still be able to be run

• CU: Four tower system used for the Beam Test at SLAC
  – No ACD contribution
  – Evolution of EM-2
  – Will need to handle external sources of data
  – 1553, SSR and SIS communications not required to satisfy test
  – Test bench scripts will still be able to be run

• FU/LAT: The complete system
  – Communications only through SIS, 1553 and SSR
  – Will need IOC/MOC-like interface
  – Test bench scripts will not be able to be run
Schedule

- EM test
  - March/April 2003
- CDR
  - April 2003
- SIS (Spacecraft Interface Simulator)
  - April 2003 (preliminary version January ’03?)
- CU beam test
  - May/June 2004
- FU/LAT integration
  - October 2004
- Airplane end-to-end test
  - February 2005