 GLAST LAT TECHNICAL DOCUMENT	Document # <b>tbd</b>	Date Effective <b>September 11, 2001</b>
	Prepared by(s) <b>Curt Brune</b>	Supersedes <b>None</b>
	Subsystem/Office Electronics Subsystem	
Document Title <b>LAT Event Simulation</b>		

**Gamma-ray Large Area Space Telescope (GLAST)**  
**Large Area Telescope (LAT)**  
**LAT Event Simulation**

**Created: July 24, 2001**

**Revised: September 11, 2001**

**CHANGE HISTORY LOG**

<b>Revision</b>	<b>Effective Date</b>	<b>Description of Changes</b>
1	8/16/01	Added simulation parameter table

<b>1</b>	<b>Overview.....</b>	<b>5</b>
1.1	Goals .....	5
1.2	Tools.....	5
1.2.1	DAQEngine.....	5
1.2.2	OMNeT++ .....	5
1.2.3	ROOT.....	5
1.3	General Assumptions.....	5
<b>2</b>	<b>Base Architecture.....</b>	<b>7</b>
2.1	Overview.....	7
2.2	Components.....	7
2.2.1	Generator.....	7
2.2.2	Anti-coincidence Detector – ACD .....	8
2.2.3	Global Trigger – GLT.....	8
2.2.4	Tower Module.....	8
2.2.4.1	CAL/TRG ROE .....	9
2.2.4.2	CAL/TRG FIFO.....	10
2.2.4.3	Tracker ROE.....	10
2.2.4.4	Tracker FIFO .....	14
2.2.4.5	TEM Event Builder – TEM EB.....	14
<b>3</b>	<b>Simulation Environment.....</b>	<b>15</b>
3.1.1	Input Event Data.....	15
3.1.2	Simulation Model.....	17
3.1.3	Simulation Parameters.....	17
<b>4</b>	<b>Results.....</b>	<b>19</b>
4.1.1	Dead Time .....	19
4.1.1.1	Absolute Dead Time .....	19
4.1.1.2	Relative Dead Time .....	19
4.1.2	FIFO Depths.....	20
4.1.2.1	Tracker ROE FIFO .....	20
4.1.2.2	Calorimeter FIFO.....	21
4.1.2.3	Tracker FIFO .....	22

## Figures

Figure 1.	Base Architecture .....	7
Figure 2.	Tower Component Detail .....	9
Figure 3.	Calorimeter Latency.....	10
Figure 4.	Tracker layer group .....	11
Figure 5.	GTRC Data Structure .....	12
Figure 6.	Layer Group and GTBRC connections .....	13
Figure 7.	Tracker Event Size.....	15

Figure 8. Calorimeter Event Size .....	16
Figure 9. Total Event Size .....	16
Figure 10. Relative Dead Time.....	20
Figure 11. Tracker ROE FIFO Polulation.....	21
Figure 12. CAL/TRG FIFO Population.....	22
Figure 13. Tracker FIFO Population.....	23

# 1 Overview

LATsim simulates the flow of event data through the LAT data acquisition electronics.

## 1.1 Goals

LATsim measures event processing throughput, dead times and live times for several proposed DAQ hardware architectures while varying the input event rate. In addition LATsim also investigates the ranges of several hardware parameters including:

- FIFO depths for the calorimeter and tracker data paths.
- Bandwidths between interconnection hardware modules.

The key aspects of this model are the latencies of the electronic components and the size of the event data.

## 1.2 Tools

### 1.2.1 DAQEngine

DAQEngine simulates the physical interaction of cosmic rays and gamma rays with the massive material of the LAT. DAQEngine provides LATsim with the digital values of the detector read out electronic lines following an “event”. DAQEngine is thoroughly described elsewhere.

### 1.2.2 OMNeT++

OMNet++ is a software framework for simulating “discrete packet events”, written by Andras Varga<sup>1</sup>. Discrete event simulators are used widely in the telecommunications and networking industries to model computer networks, multi-processors and other distributed systems. LATsim uses OMNeT++ to model the flow of event data through the LAT electronic network.

### 1.2.3 ROOT

ROOT<sup>2</sup> is a high energy physics analysis tool used for graphing simulation result histograms.

## 1.3 General Assumptions

All simulations attempt to model an physical phenomenon through the use of simplifying assumptions – LATsim is no different. Knowing what to model and what to simplify is difficult to know a priori . The best strategy is careful documentation of the model’s

---

<sup>1</sup> OMNeT++ home page: <http://www.hit.bme.hu/phd/vargaa/omnetpp.htm>

<sup>2</sup> ROOT home page: <http://root.cern.ch/>

assumptions and simplifications before starting to trust the simulation results.

The following assumptions apply to all hardware architectures studied.

- DAQEngine provides a reasonable and accurate model of the physical interactions.
- The input event rate is a *Poisson process*, i.e. the time between successive events has an exponential distribution.
- Pseudo-random number generators and random seeds are “good enough”

## 2 Base Architecture

### 2.1 Overview

The goal of the simulation is to analyze the proposed hardware architectures for the DAQ system. All the architectures share a common design for the electronics within a tower model – the differences lie in how event data from the TEMs rendezvous and propagate to the event processors.

Figure 1 uses color to depict the system components common to all the architectures. The box labeled *Event Builder Architecture Study* contains the rendezvous and propagation logic unique to each architecture, i.e. the white box is different for each hardware design studied.

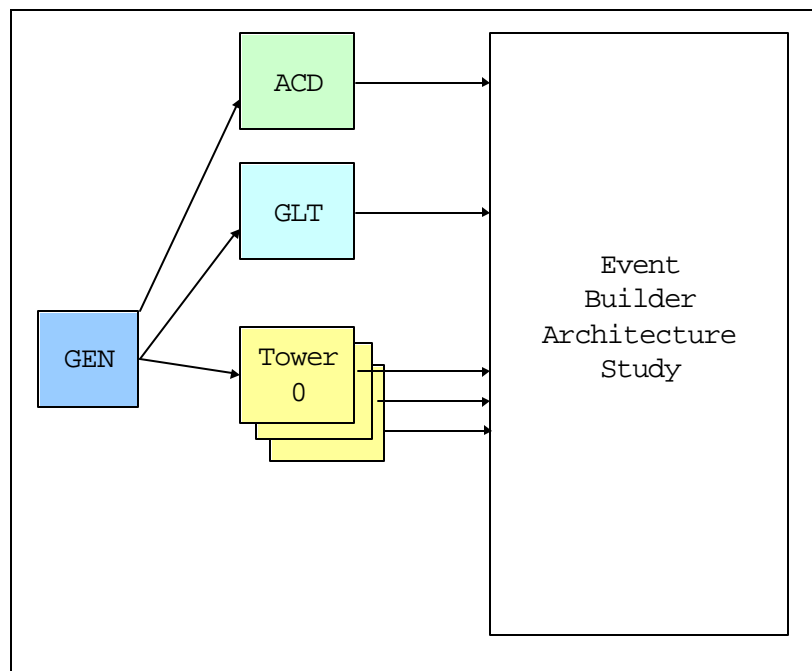


Figure 1. Base Architecture

### 2.2 Components

#### 2.2.1 Generator

Referring to Figure 1, the generator component is a logical ( not a physical ) component of the simulation model that sends event data to the other components of the model. It's primary duties are:

- Calculate the time between events, modeled as a *Poisson process*.
- Record dead times if any receiving component is unable to handle a new event.
- Read DAQEngine events from a file and prepare them for injection into the LAT

simulation network. For each event it creates an event packet for:

- The ACD component – See below
- The GLT component – See below
- For each tower component ( 16 towers ) the generator creates a calorimeter/trigger packet and a tracker packet – See below.

## 2.2.2 Anti-coincidence Detector – ACD

The ACD component simply models latency as a constant, i.e. for every ACD event packet the ACD component puts the entire system into a “busy” state which contributes to the overall dead time. This latent time is small compared to other latencies in the system.

Latency Source	Latency (mseconds)
ACD	10

## 2.2.3 Global Trigger – GLT

The GLT component is similar to the ACD in that it only models latency and contributes to the overall system dead time.

Latency Source	Latency (mseconds)
GLT	10

## 2.2.4 Tower Module

Referring to Figure 2 below, the 16 tower modules each receive two event packets from the generator: a calorimeter/trigger (CAL/TRG) packet and a tracker packet. Each data path begins with a **R**ead **O**ut **E**lectronics (ROE) sub-component, followed by a FIFO sub-component and terminates in the TEM Event Builder.

The ROE sub-component simply models latency, similar to the GLT and ACD components previously described. After delaying the data packet for a finite amount of time the ROE sub-component transmits the packet to the FIFO sub-component. While the ROE is in its latent period the entire system experiences dead time.



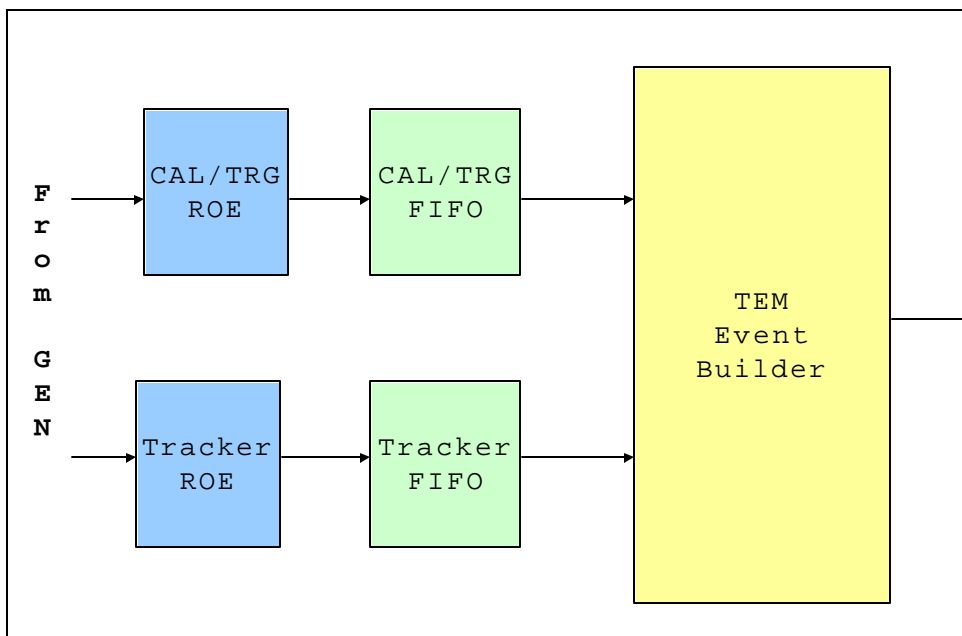


Figure 2. Tower Component Detail

The FIFO sub-components store and forward event packets as the TEM Event Builder (TEM EB) requests data.

### 2.2.4.1 CAL/TRG ROE

The incoming CAL/TRG event packet contains data for both the calorimeter and for the tower trigger. The trigger event size is modeled as a constant of 6 32-bit words.

Table 1 shows the calorimeter data structure as a list of 32-bit words.

Header 1
Header 2
Hit 1
...
Hit n

Table 1. Calorimeter Data Structure

The size is given in bits by the following formula:

$$\text{CAL Event Size} = 32 \cdot (2 + \text{totalHits})$$

Eqn 1

The maximum size of a CAL/TRG packet is then known since the maximum number of hits is 96 ( 8 layers X 12 logs ). Adding the fixed size of the trigger data the maximum

size is 416 bytes.

The latency of the CAL/TRG ROE is modeled as a constant 22  $\mu\text{sec}$  with contributions from<sup>3</sup>

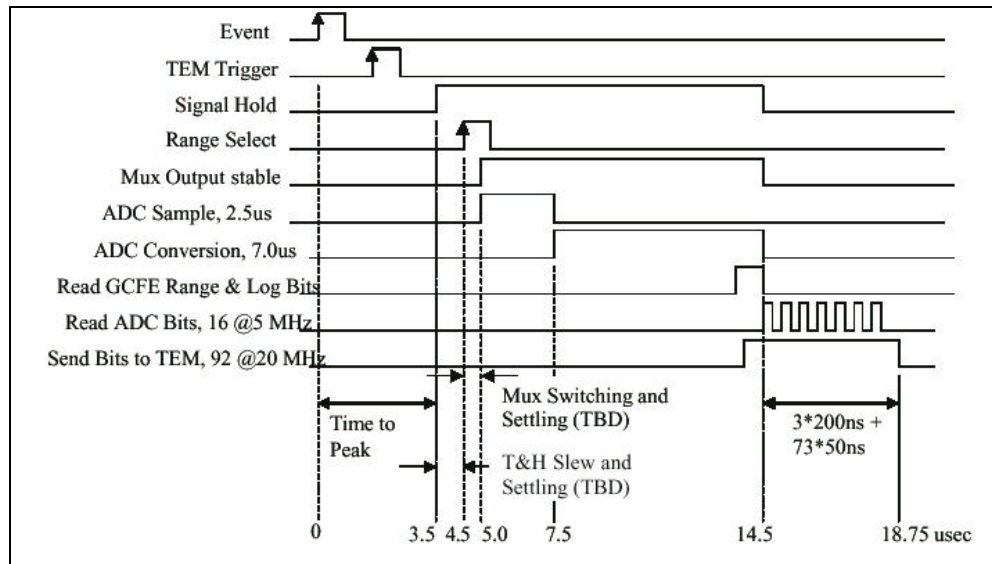


Figure 3. Calorimeter Latency

#### 2.2.4.2 CAL/TRG FIFO

The CAL/TRG FIFO sub-component stores and forwards event packets coming from the CAL/TRG ROE sub-component to the TEM EB. The CAL/TRG FIFO sends its event data to the TEM EB when the TEM EB is ready and when the complimentary event data is available in the Tracker FIFO.

One of the main purposes of this study is to size the FIFO depths so that dead time is minimized while conserving FIFO memory. By running the simulation at different input event rates and FIFO depths we can analyze the impact on total system dead time.

#### 2.2.4.3 Tracker ROE

The tracker ROE is the most complicated of all the tower systems. A complete description of the tracker is beyond the scope of this document<sup>4</sup>, but a description of the simulation model is appropriate.

##### 2.2.4.3.1 Tracker model

The tracker consists of 36 layers ( 18 for X and 18 for Y ) read out by 8 cables. The layers are further sub-divided into groups of 9 layers with each group serviced by 2 cables (one cable on each end of the layer group) – this gives a total of 8 cables servicing 4 layer groups. Figure 4 is a diagram of a layer group (9 layers) with two read out cables.

<sup>3</sup> CAL Dead Diagram courtesy [Calorimeter Peer Design Review Presentation, NRL](#)

<sup>4</sup> See document TBD for complete tracker details.

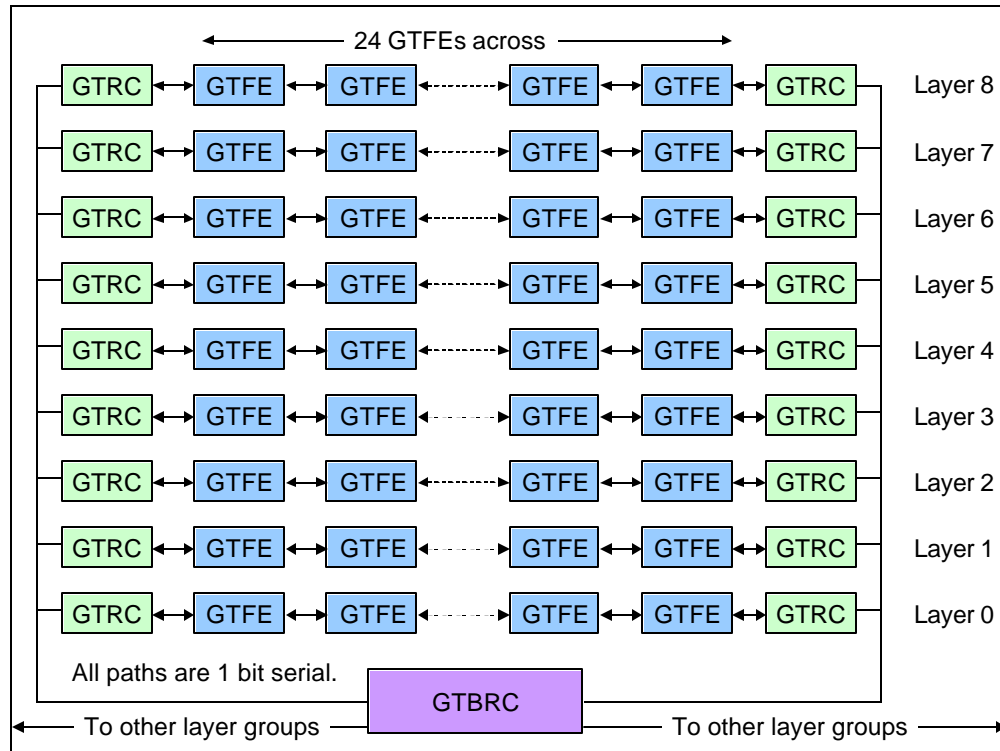


Figure 4. Tracker layer group

### 2.2.4.3.2 GTFE

The GTFEs are the front-end silicon detector components, 24 per layer. The GTFE record the address of the silicon strips hit for each event, i.e. a single GTFE can record multiple hits.

The GTFEs have the capacity to buffer up to four complete events. This buffering is crucial to the success of the instrument and is modeled in LATsim.

When an event trigger occurs an active GTFE latches its data for further processing. Every active GTFE records at least 3 bits of information per trigger, while GTFEs with hits record an additional 64 bits of information. Therefore the amount of data in a layer is given by:

$$\text{Layer Data (bits)} = 3 \cdot GTFE_{active} + 64 \cdot GTFE_{hit}$$

Eqn 2

In addition to the hit information all the GTFEs are wire-OR-ed together to record the time over threshold (TOT) value.

### 2.2.4.3.3 GTRC

The GTRCs are the layer read out controllers, 2 per layer. The GTRCs read and package the data for a layer from the GTFE components. The GTRC records the addresses of the layer hits and the TOT for the layer. On an event trigger every layer GTRC creates a structure of 12-bit words as follows:

Layer #
# of hits
Hit 1
Hit 2
.....
Hit 3
TOT

Figure 5. GTRC Data Structure

From this it follows that the size of the GTRC data in bits is:

$$\text{GTRC size} = 12 \cdot (3 + \text{totalHits})$$

Eqn 3

A portion of the total tracker latency comes from clocking the data over from the GTFEs to the GTRCs – see below.

**ASSUMPTION:** The GTRCs have the capacity to buffer two complete events for a layer. The GTBRC selects between these buffers when draining the GTRCs. This buffering and selecting is NOT currently modeled in LATsim – see below.

#### 2.2.4.3.4 Event Read Out

Referring to Figure 4 each layer of the tracker consists of 24 GTFEs with 1 GTRC on each end. Half of the GTFEs (12) read out to one GTRC and half read out to the other GTRC (1 bit serial data path). The number 12 may change as GTFEs die or connections between GTFEs break, e.g. in the event of a broken connection 1 GTRC could service 18 GTFEs while the other one only services 6.

The data cable connects the 9 layer GTRCs to the GTBRC. The GTBRC is connected to all 8 data cables and round robin selects between them as shown in Figure 6. Within a Layer Group the GTBRC requests event data from each layer GTRC in order ( GTRC0, GTRC1, GTRC2, etc..). After assembling the layer event data the GTBRC stores the data in the Tracker FIFO (review Figure 1 and Figure 2).

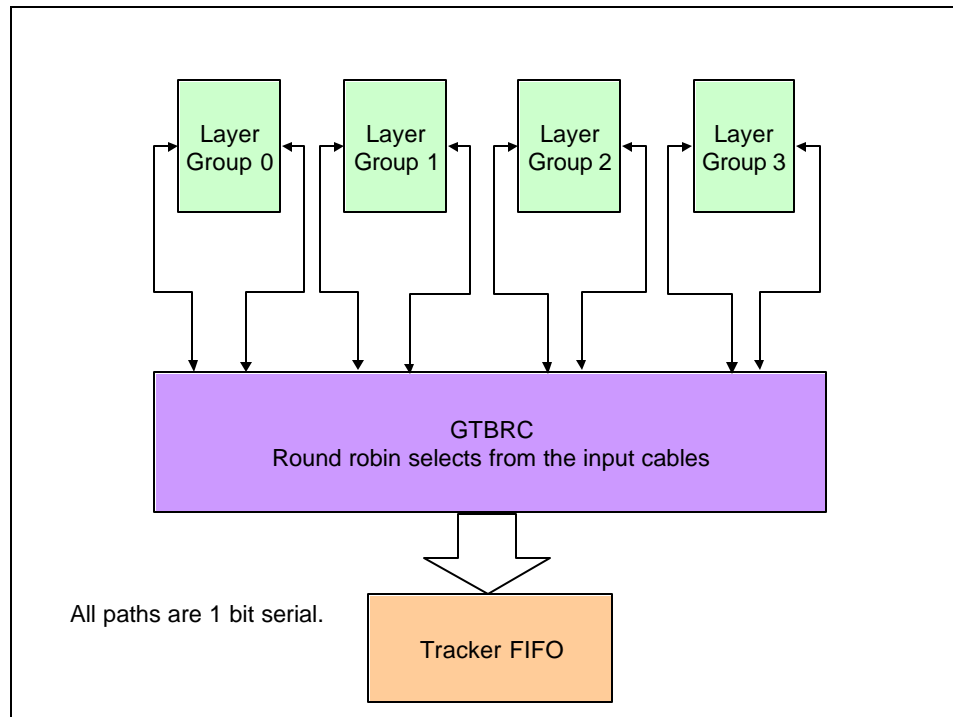


Figure 6. Layer Group and GTBRC connections

### 2.2.4.3.5 Latency Modeling

To avoid having an overly complex simulation the model must be simplified. The simplifications are chosen so that the model represents the worst-case scenario, i.e. under normal operations the instrument will perform better than LATsim.

The major components of latency in the Tracker ROE are:

1. The time to move the layer hit data from the GTFEs to the GTRC.

**Note:** This step happens in parallel for all layers of all layer groups.

2. The time to digitize the layer TOT value. This occurs in parallel with step 1.
3. The time to move the GTRC data to the GTBRC. As previously noted this is somewhat complex. First, the GTBRC is required to read the data from a cable in layer order. Second, the GTRCs are double buffered meaning the GTBRC needs to select which buffer to read from. Third, in some circumstances the data move operations are overlapped with the operations from step 1, reducing the latency.

LATsim models the Tracker ROE latency as the sum of the larger of step 1 and step 2, plus the maximum of step 3. In other words the TKR latency is the sum of step 3 plus whichever is larger, step 1 or step 2. Considering the maximums of these latency

sources frees LATsim from modeling the overlapping data move operations of the GTFE, GTRC and GTBRC. This is reasonable for steps 1 and 2 since the GTFEs transmit their data in parallel with the TOT digitization. For step 3 the resulting maximum latency is higher than expected for the actual instrument, but it places an upper bound on the latency and is simple to calculate and understand.

Referring to Figure 4, the maximum of step 1 is straightforward to calculate from the GTFEs that have hits. The maximum layer data size is calculated using Eqn 2 for each of the 9 layers of the group. The transfer time is the time required to clock the maximum data size from the GTFEs at the global clock frequency of 20 MHz. Since the data path is 1 bit wide, this gives a bandwidth of 20Mbit/s.

For a given layer group the maximum latency from step 1 and step 2 is the larger of the transfer time and the TOT value.

Finding the maximum of step 3 reduces to finding the cable that has the maximum data size since the bandwidth is fixed at 20 Mbits/sec. From Eqn 3 the data size on any cable is given by:

$$\text{Cable data size} = \sum_{\text{layer } 0}^{\text{layer } 8} 12 \cdot (3 + \text{totalHits}_{\text{layer}})$$

Eqn 4

Using a bandwidth of 20Mbit/s and Eqn 1 to find the maximum cable data size, one computes the maximum time for the GTRC to GTBRC transfer.

#### Simplifications:

1. As noted above the GTFEs can clock over while the GTRCs are clocking down. LATsim does not model this overlapped data move operation.
2. The GTRCs are double buffered – the GTBRC knows how to select between the buffers. LATsim does not model this.
3. The GTBRC round robin selects between the 8 cables, sucking 12 bits from each cable with data before going to the next -- at this step the GTBRC attaches 3 bits of data (the cable ID) to form 15-bit words. LATsim rounds this up to 16-bit words.

#### 2.2.4.4 Tracker FIFO

The Tracker FIFO sub-component behaves similarly to the CAL/TRG FIFO previously described.

#### 2.2.4.5 TEM Event Builder – TEM EB

The TEM EB assembles the data from the CAL/TRG FIFO and the Tracker FIFO into a single “event” packet for that tower. Next the TEM EB forwards the packet onto the next stage of the event pipeline, which depends on the individual hardware architecture designs. Recall Figure 1 and Figure 2.

## 3 Simulation Environment

### 3.1.1 Input Event Data

The input event data consists of 1000 chime events generated by DAQEngine. Below are histograms characterizing the total event data size subdivided into *Tracker Event Size*, *Calorimeter Event Size* and *Total Event Size*. The *Total Event Size* is the sum of the TKR size, the CAL size, the GLT size and the ACD size. These sizes represent the total contribution from all 16 towers of the LAT.

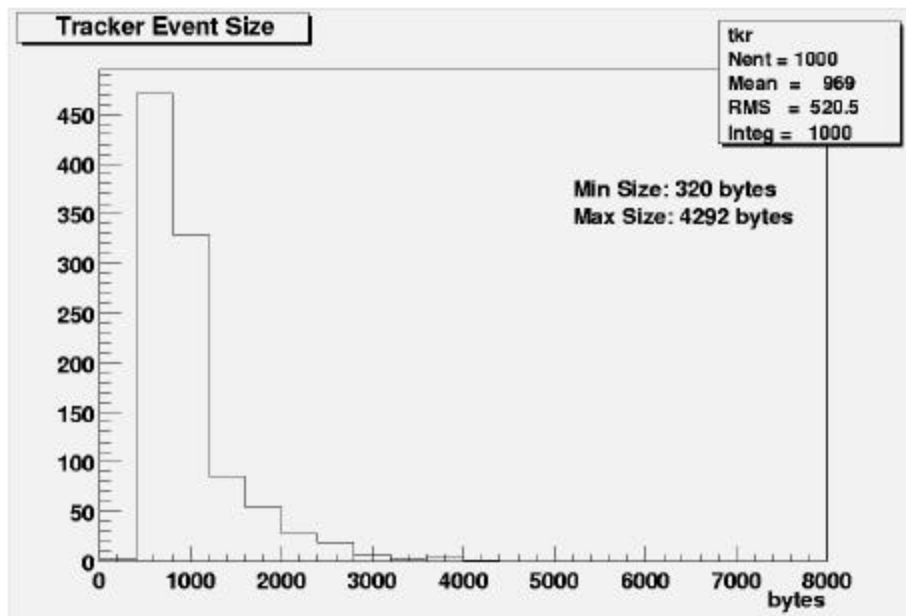


Figure 7. Tracker Event Size

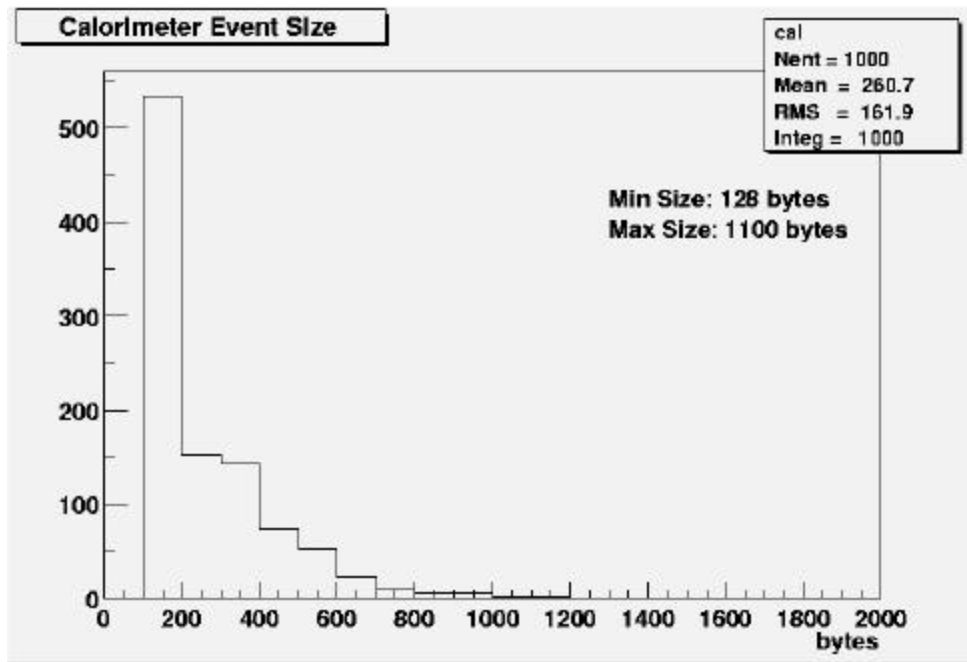


Figure 8. Calorimeter Event Size

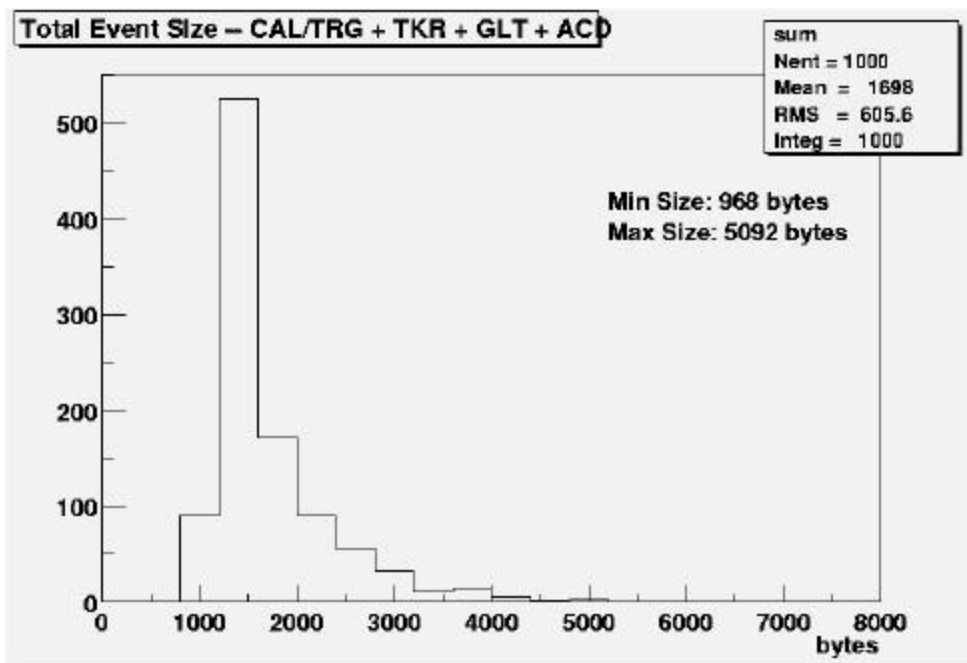


Figure 9. Total Event Size



Note the total number of events is 1000. The range of the *Total Event Size* is about 1KB to 5KB with a mean of 1.6KB.

Modeling the event generation as a Poisson process leads to an exponential distribution of the *inter-arrival* time with a mean of *m*. The inter-arrival time distribution is given by,

$$\frac{e^{-\frac{x}{m}}}{m}$$

Eqn 5

This study simulated average inter-arrival times of 10 *µsec*, 50 *µsec* and 100 *µsec*, corresponding to event rates of 100kHz, 20kHz and 10kHz respectively.

### 3.1.2 Simulation Model

The base architecture simulation model is shown in Figure 1, where the *Event Builder* box is simply an infinite event sink, providing no “back pressure” to the front-end electronics. This provides a baseline for the simulation study.

In addition every FIFO has infinite depth in order to measure the maximum FIFO depth required for each run.

### 3.1.3 Simulation Parameters

The following tables summarize the parameters used in the base simulation.

Global Parameters	
System clock	20 MHz

Fixed Message Sizes	
GLT per tower message size	16 32-byte words
ACD per tower message size	5 32-byte words

Bit Widths	
TKR ROE	8
CAL ROE	0 <sup>5</sup> See footnote

---

<sup>5</sup> A bit width of 0 implies “infinite” bandwidth in the simulation. Infinite bandwidth is used for the CAL ROE because the latency due to the bandwidth is already included in the total fixed latency for the CAL ROE.

CAL/TRG FIFO	16
TKR FIFO	16
TEM	16
ACD ROE	1
GLT ROE	1

The bit width is how many 1-bit lines leave a component. The bandwidth leaving the component is found by multiplying the bit width times the system clock. E.g. the bandwidth of the TKR FIFO is  $16 * 20\text{MHz} = 320\text{Mbit/sec}$ .

Fixed Latencies	
GLT	10 <i>msec</i>
ACD	10 <i>msec</i>
Calorimeter	22 <i>msec</i>

## 4 Results

### 4.1.1 Dead Time

Dead time is measured in two ways – *absolute* dead time and *relative* dead time. These are defined as follows:

Absolute dead time	The elapsed time the instrument remains in a “busy” state, unable to service an event. E.g. the calorimeter incurs an <i>absolute dead time</i> of 22 <i>msec</i> on every event trigger.
Relative dead time	The percentage of the total simulation run time the instrument is in a “busy” state, unable to service events. E.g. the <i>relative dead time</i> is approximately 30% at an input event rate of 20kHz.

Table 2. Dead time definitions

#### 4.1.1.1 Absolute Dead Time

For all input event rates (ranging from 1Hz to 100kHz) the absolute dead time is 22 *msec*. This is a direct consequence of the CAL latency model, which is modeled as a constant of 22 *msec*. The dead times of the other systems are hidden by the large latency of the CAL.

Interestingly the input event rate has no effect on the absolute dead time – after an event trigger the system is dead for 22 *msec*, regardless if more events are pounding on the instrument during that time.

#### 4.1.1.2 Relative Dead Time

Figure 10 is a graph of the relative dead time versus input event rate. As shown the relative dead time increases as the input event rate increases. This behavior is expected – at higher event rates more and more of the “inter-arrival” time exponential distribution lies below the fixed latency of the CAL, thus increasing the relative time.

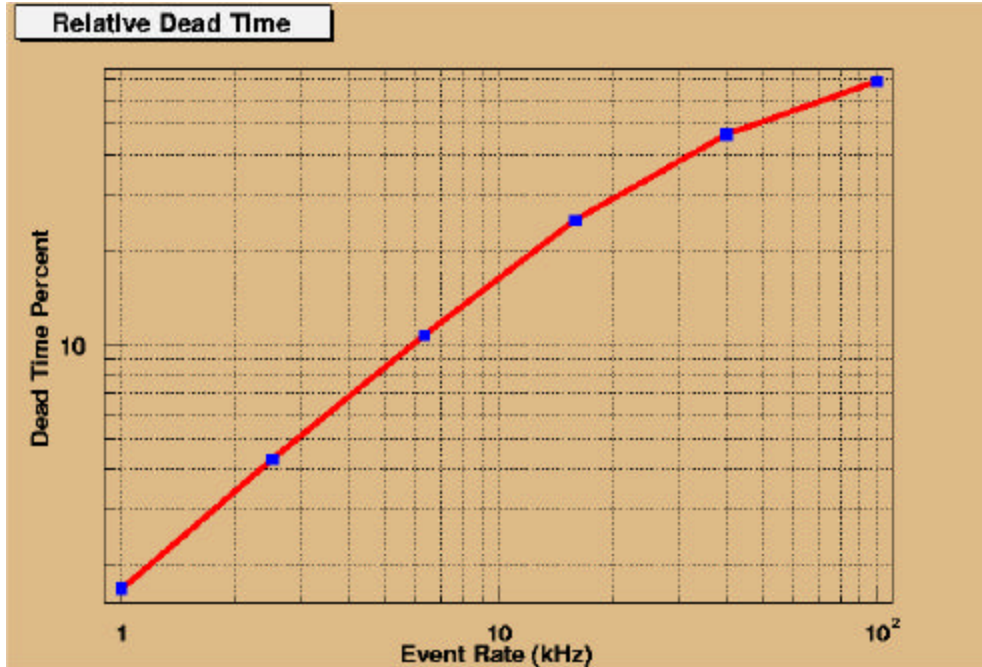


Figure 10. Relative Dead Time

## 4.1.2 FIFO Depths

Referring to Figure 2 the base architecture consists of two FIFOs – one for the CAL and one the TKR. In addition to these two FIFOs the tracker ROE module contains a *FIFO-like* object in its GTFE components with the ability to buffer 4 events. These three FIFOs are discussed next.

### 4.1.2.1 Tracker ROE FIFO

As noted in section 2.2.4.3.2 on the GTFE, the front end read out electronics has the ability to buffer up to 4 complete events. In many regards the GTFE buffer functions as a FIFO and is modeled as such. Figure 11 below is a histogram of the population of this “FIFO-like” object. The entries in the histogram correspond to the maximum GTFE buffer depth of all 16 tower for each event.

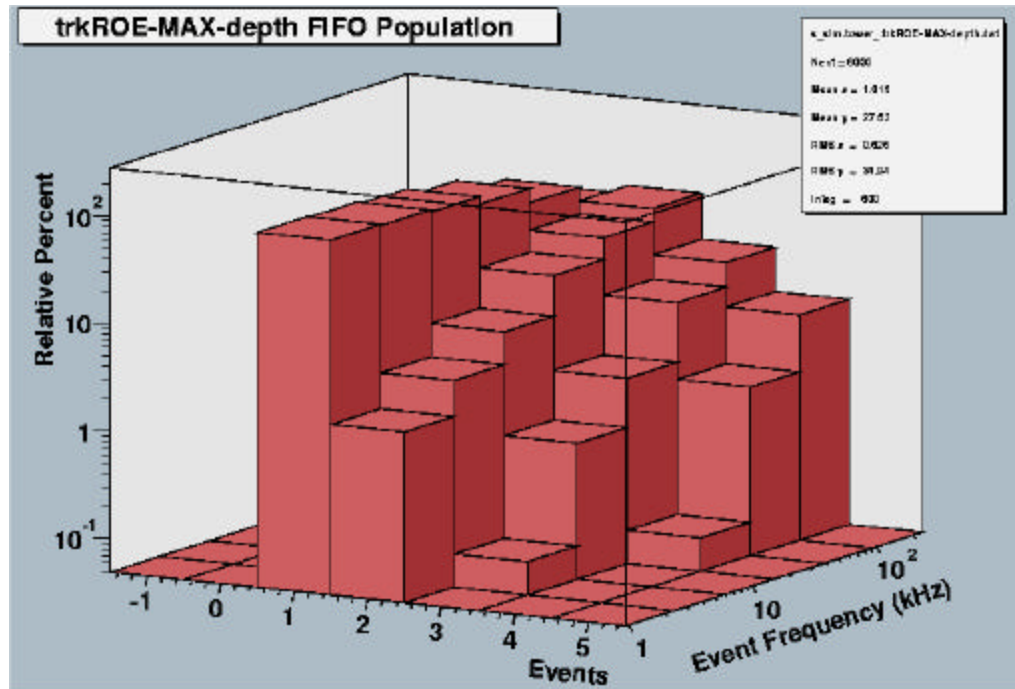


Figure 11. Tracker ROE FIFO Population

Note: The X-axis is the GTFE buffer depth in *events* as the GTFE stores **complete** events. The Y-axis is the input event frequency on a log scale, with 6 entries covering the range 1kHz to 100kHz. The Z-axis is the number of events, normalized to a percentage on a log scale.

For the most part the tracker ROE FIFO holds a single event, rarely using the extra buffer capacity. At a high event rate of 50kHz the 4<sup>th</sup> GTFE buffer is populated at approximately the 5% level.

#### 4.1.2.2 Calorimeter FIFO

As noted in section 2.2.4.2 on the CAL/TRG FIFO is a simple “store and forward” object. Figure 12 below is a histogram of the population of this FIFO object. The X-axis is the FIFO memory used in bytes. The Y-axis is the input event frequency on a log scale, with 6 entries covering the range 1kHz to 100kHz. The Z-axis is the number of events, normalized to a percentage on a log scale.

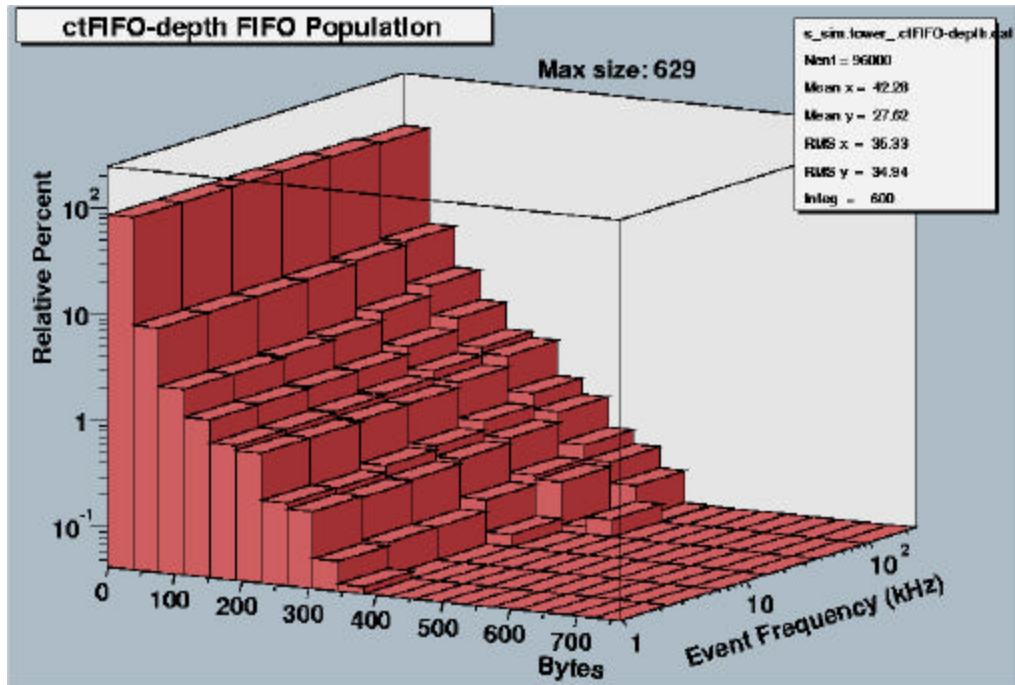


Figure 12. CAL/TRG FIFO Population

Note the units of the histogram are in bytes. The maximum CAL/TRG event stored in any single tower is 629 bytes, while the average size is 42 bytes. A 4KB FIFO would be able to store 6 maximum events or almost 100 average sized events.

#### 4.1.2.3 Tracker FIFO

As noted in section 2.2.4.4 on the Tracker FIFO is a simple “store and forward” object. Figure 13 below is a histogram of the population of this FIFO object. . The X-axis is the FIFO memory used in bytes. The Y-axis is the input event frequency on a log scale, with 6 entries covering the range 1kHz to 100kHz. The Z-axis is the number of events, normalized to a percentage on a log scale.

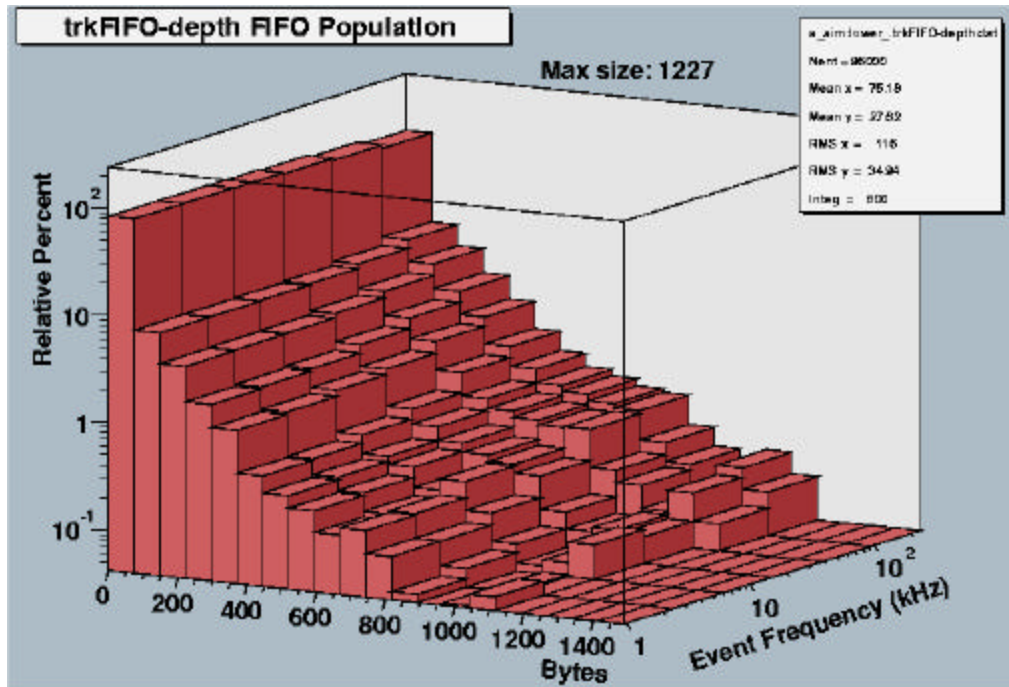


Figure 13. Tracker FIFO Population

Note the units of the histogram are in bytes. The maximum TKR event stored in any single tower is 1227 bytes, while the average size is 75 bytes. An 8KB FIFO would be able to store 6 maximum events or over 100 average sized events.