Queueing Simulation of Silicon Track Trigger Dead Time Using the Ptolemy Simulation Package DØ Note 3673

S.L. Linn
Florida State University
(August 16, 1999)

Abstract

A simple model of the Silicon Track Trigger (STT) was implemented in the Ptolemy simulation package. The dead time for different track fitting times was measured for one and six geometric sectors. The inclusion of all six sectors was found to increase the system dead time from 5% to 25% for a track fitting time of 80 μs. Spreading the track fitting over two modules per sector roughly doubles the system performance.

I. INTRODUCTION

In section II, the Ptolemy 1 simulation package (*http://ptolemy.eecs.berkeley.edu/*) is described. In section III, the baseline STT trigger(*Heuring DØ Note 3492*) is modeled, and in section IV the results of a simple example are described. Section V is a short term plan with the goal of using the simulation to guide finalzation of the STT design.

FIG. 1.

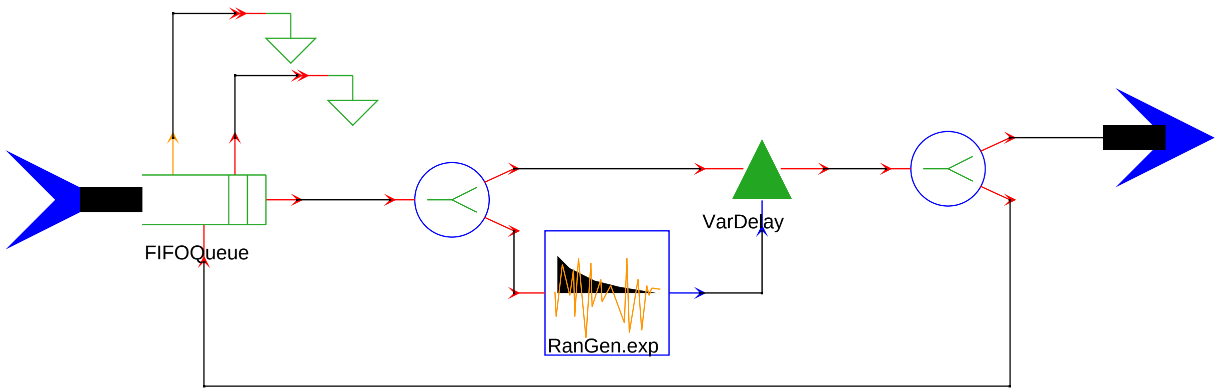

FIG. 2. Schematic of a queue and serve galaxy.

II. PTOLEMY

"The Ptolemy project studies modeling, simulation, and design of concurrent, real-time, embedded systems. The focus is on assembly of concurrent components. The key underlying principle in the project is the use of well-defined models of computation that govern the interaction between components. A major problem area being addressed is the use of heterogenous mixtures of models of computation. A software system called Ptolemy II is being constructed in Java. The work is conducted in the Department of Electrical Engineering and Computer Sciences of the University of California at Berkeley. The project is directed by Prof. Edward Lee. The project is named after Claudius Ptolemaeus, the second century Greek astronomer, mathematician, and geographer."

In other words, Ptolemy can simulate complex systems at different levels of detail and with different time scales. The so-called models of computation are called *domains* which are executed in closed systems called *universes* . Universes can communicate through well defined *worm holes*. For simple queueing simulations, we need not concern ourselves with this degree of generality, and limit ourselves to the *Discrete Event* (DE) domain. Hierarchy is implemented in a universe with galaxies which in turn contain *stars* (the basic elements of simulation).

A queueing simulation is created by construction of a universe containing galaxies and stars, which are chosen from a *palette* . Data flow between these objects is implemented by drawing lines between well defined input/output *ports*. This can be done through a GUI or by scripting. Ptolemy is written in C^{++} and uses dynamically loaded libraries. The source code is readily available, and user coded stars (functions) are simple to implement. One can choose from a variety of sources(signal types), sinks(histograms, plots, text), and operations (routers, queues, servers). In the DE domain, an event is defined by a starting time and a data value(event number). Delays incurred while waiting in queues or while being served are added to the starting time.

As a illustration, Fig.2 shows a schematic of a general queue and serve galaxy. The FIFO queue is of variable size. Events arrive from the left side and depart from the right when triggered by the bottom input. If an event arrives when the queue is full, it is routed to

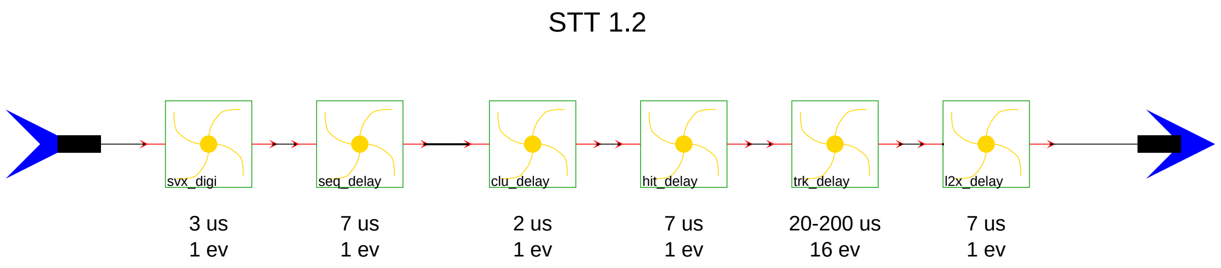

FIG. 3. Schematic of an STT sector galaxy.

Process	Service Time(μs)	Distribution	Buffer Size
L1 Accept	100	Poisson	-
SVX digitization	10(3)	fixed	1
Data to STT	7	Exponential	1
Cluster finding	2	Exponential	1
Data to track fitting	7	Exponential	1
Track fitting	20-200	Exponential	16
Data to Global L2	7	Exponential	1

TABLE I. Baseline STT Parameters

one of the top outputs, which in this case is terminated in a null sink(inverted triangle). The other top output carries the queue size and is terminated in this example. In order to implement a variable delay given by an exponential distribution, the output event is split and sent to trigger the random generator, and the other branch is delayed by that amount. In this example, the delayed event is split, with one branch goung to the output, while the other branch is sent to trigger release of the next event. The large arrows define the galaxy input and output points.

III. DØ STT MODEL

Figure 3 shows the sequence of queue and serve galaxies that comprise the STT processing chain for a single sector. The service time and queue depths for each element are summarized in Table I. The first element is the SVX digitization which is taken as a fixed delay of 10μs (A more recent estimate for digitization time is 3μs). The second galaxy represents the data transfer through the sequencer into STT processors. Next is the galaxy which performs the cluster finding which passes its data through a transfer galaxy to the track fitter and finally a transfer to the global L2.

Elements 2,3,4,6 all have buffers one event deep and exponential service times. The input to track fitting has a 16 event buffer. Each FIFO in the chain releases the next event after

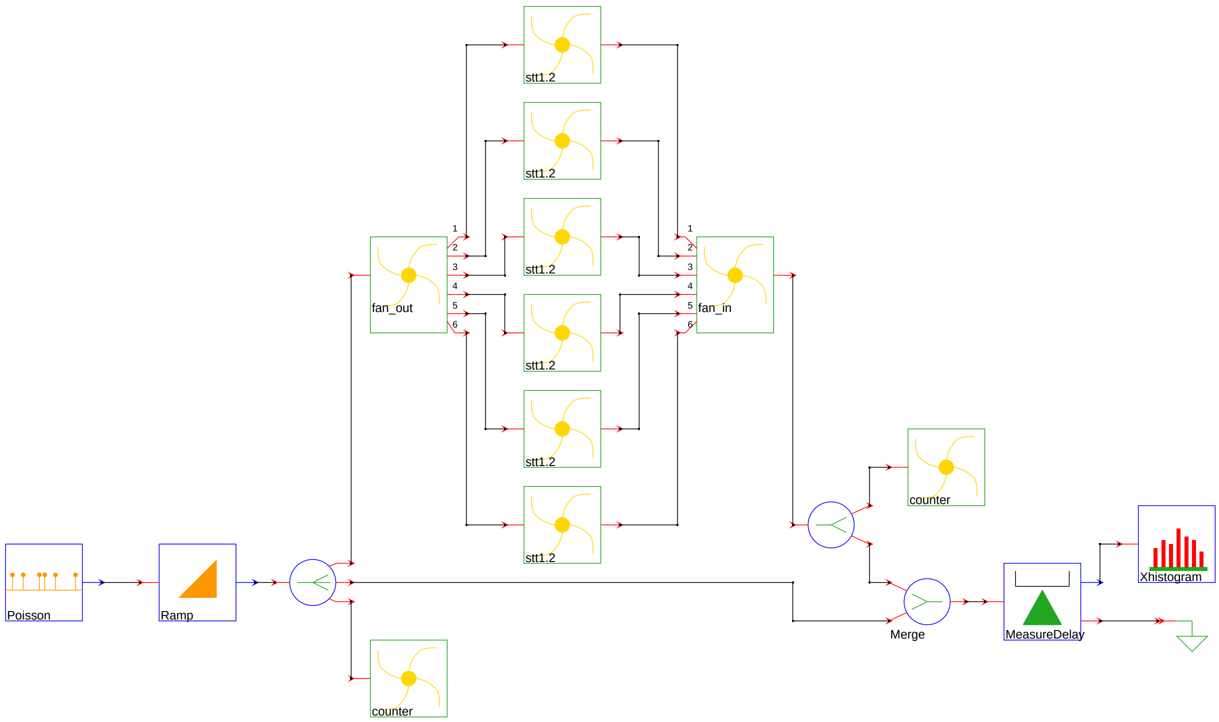

FIG. 4. Schematic of the STT universe.

servicing the previous event. In this model there is no global 'sync' and events that arrive at a processor with a full FIFO are lost.

Three models were tested. The first is a single sector for which time parameters apply, and the second is a simulation that has all six sectors running in parallel. At this time only a single exponential random distribution was available. As such, long tails characteristic of DØ events were inadequately modeled. Also, each galaxy representing the processing sequence used independently generated random numbers. It seems likely that processes in the chain are correlated. For example, large events will take longer to transfer and longer to process. In the future these shortcomings will be handled with custom coded galaxies. The last model assumed two fold parallelism in the track fitting modules, each with its own 16 event buffer.

IV. EFFECT OF TRACK FITTING

A schematic of the simulation universe is shown in Fig. 4 where in addition to STT elements are sinks for obtaining statistics and plotting distributions. The Level 1 accept rate is represented on the left as a Poisson generator source with an average of 100μs between

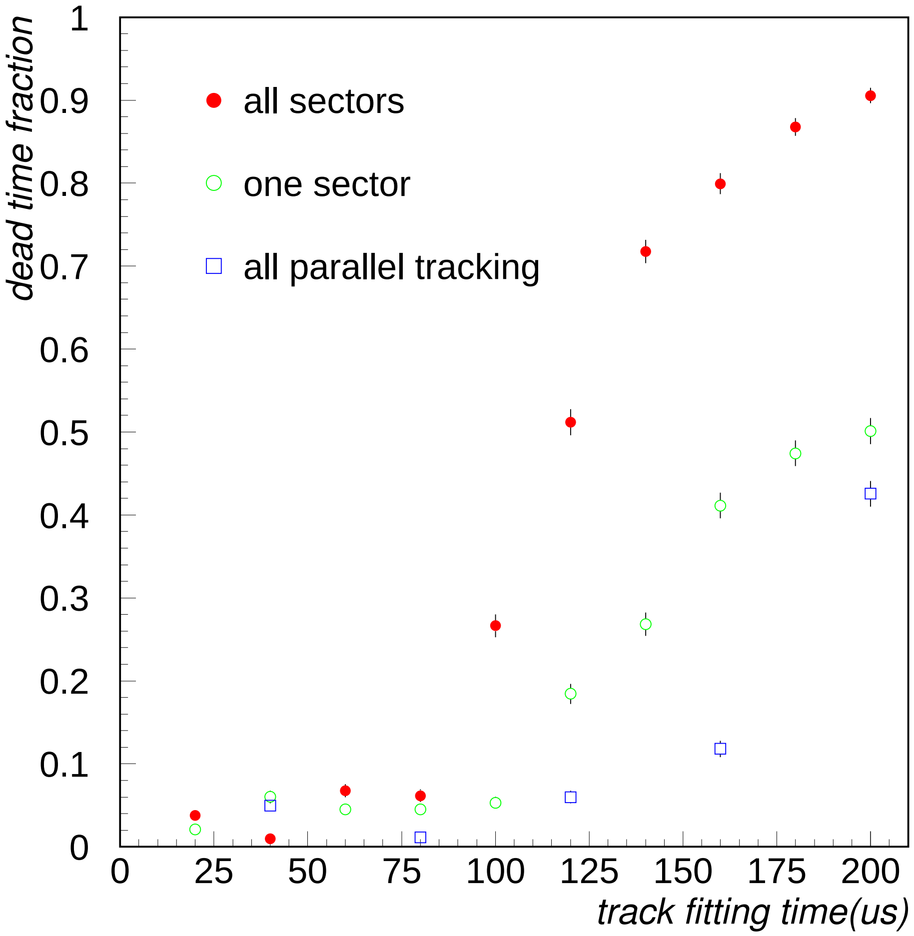

FIG. 5. Dead time vs. Track fitting time for one (open), six(solid) sectors, and six(square) sectors with parallel track fitting.

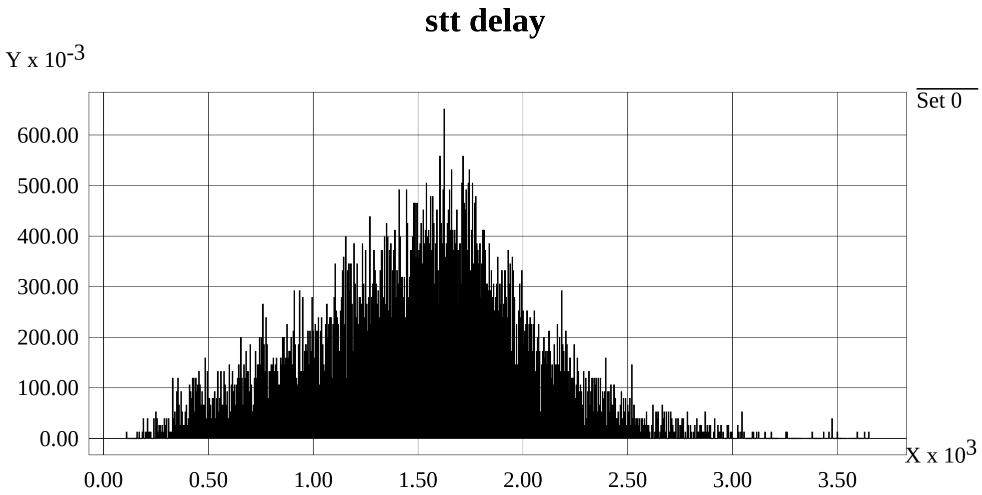

FIG. 6. Events vs. Delay Time through the system for a 100 μs track fitting time.

events. This is fed into a ramp generator which assigns a unique event number, which is used for later comparisons. For purposes of obtaining statistics, each event is sent to a counter and also routed around the STT with no delay. Delay times are measured relative to the un-delayed signal and histogramed. Live times are defined as the ratio of events going through the STT system delay, and not lost due to full buffers, to the total number of events simulated.

Simulations were done over a range of track fitting times of 20-200μs. The dead time results are shown in Fig.5 where both models have less than 5% dead time until a track fitting time less than 80μs is reached. The errors shown are binomial; however, at small service times the effect of buffers introduces additional correlations. Correlations between sectors are also not accounted for, since Monte Carlo events were not used to simulate sector occupancy. For longer fitting times the six sector model dead time rises steeply. The delay time distribution for a 100μs fitting time is shown in Fig.6, and it can be seen to depart significantly from an exponential distribution. The mean is slightly larger than the sum of all service times; however, long tails extend past twice the average time. Finally, a model with two track fitting modules per sector was simulated. In principle this should halve the *total* track fitting time. For comparison to the other models, twice the track fitting time is plotted in Fig. 5. As expected, this improves performance by yielding a 10% dead time for a track fitting time of 160μs.

V. CONCLUSIONS

The initial experience with Ptolemy can be summarized as follows:

- It can model the Level 2 DØ trigger at a greater level of detail than RESQ.
- The run-time environment is reliable and robust.
- It is well doccumented, with tutorials, and examples.
- Lack of 'support' is offset by a large user group.
- It has sufficient generality to implement more realistic queueing models than have thus far been realized.

To further design of the STT the following tasks should be completed.

- Implement hyperexponential random distribution
- Include other Level 2 branches
- Understand how correlations effect sequential service times
- Study more parallelism/buffering at the track fitting stage
- Implement realistic control structure