Dead Time Free STAR Trigger Architecture

V. Lindenstruth, H. Crawford, L. Greiner, E. Judd, G. Visser
UC Berkeley Space Sciences Laboratory
K. Marks
Lawrence Berkeley National Laboratory
J. Whitfield
Carnegy Mellon University

This paper describes the hard and software architecture of the STAR Trigger system. STAR is a large scale physics detector with an aggregate bandwidth of more than 50 GB/sec requiring a very large degree of selectivity in order to maximize the information content in the recorded data on tape given the taping speed of less than 20MB/sec. This data compression and selection task is done by the STAR trigger system. Two low Trigger layers are a twin multi processor farm that performs an event rate reduction of two orders of magnitude while accepting trigger rates of 10^5/sec. It is implemented as shared memory based write-only data driven push architecture completely avoiding any resource locks. Modsim performance simulations are shown, prooving that there is no dead time imposed by the trigger system.

Introduction

The **S**olenoidal **T**racker **A**t **R**HIC (STAR) is a large scale high energy particle detector currently under construction at the **R**elativistic **H**eavy **I**on **C**ollider (RHIC). The main physics interest is to observe the phase transition of nuclear matter to a *quark-gluon-plasma*. Such a plasma is expected to have existed during the first few microseconds after the big bang. The signatures of the quark-gluon-plasma are unknown. Therefore the STAR detector system was designed to measure the more than 2000 expected decay particles of a central Au+Au reaction nearly hermetically. Figure 1 shows a sketch of the STAR detector system. The corresponding event sizes and data rates are summarized in table 1.

The raw data rates stated here exceed taping capabilities by many orders of magnitude. Current tape technology allows one zero suppressed TPC event per second be written to tape, which is sufficient if these events are physically interesting. Therefore the performance of the STAR trigger system is essential for the performance of the whole detector.

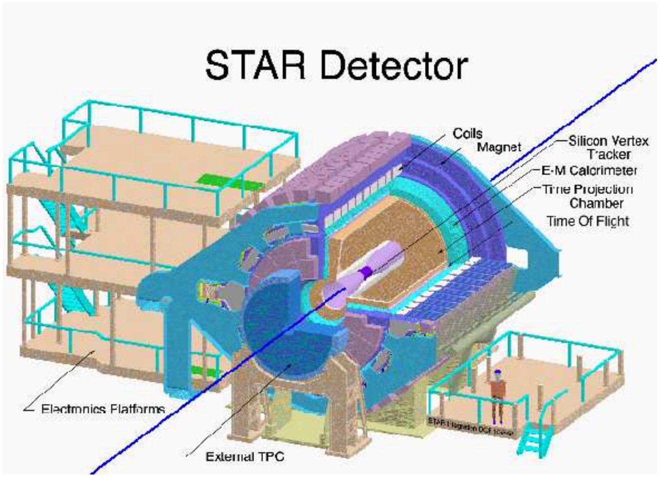

Figure 1: The STAR detector

Table 1: *The STAR system parameters*

detector	*channels*	*raw evt*	*Trg. rate*	*data rate*
TPC	140000	140MB	100/sec	14GB/sec
SVT	72000	18MB	100/sec	1.8GB/sec
TOF	8000	32kB	10^5/sec	3.2GB/sec
EMC	1200	2.4kB	10^5/sec	2.4GB/sec
Trigger	2000	2kB	10^7/sec	20GB/sec
Tape	1			<20MB/sec

STAR Trigger Architecture

There are two groups of subdetectors in STAR with different readout times and trigger rate capabilities. They can operate in coincidence but also be used stand alone. The trigger system therefore has to be capable of tracking individual dead times and to fire coincidence triggers based on the availability of any given subsystem at any given time. The L0 trigger system performs this task producing up to 10^5 triggers per second. All higher trigger layers perform either a data selection or compression task. The main difference between these layers is the complexity of the input data set, the input/output event rate, and the available time budget for the event analysis.

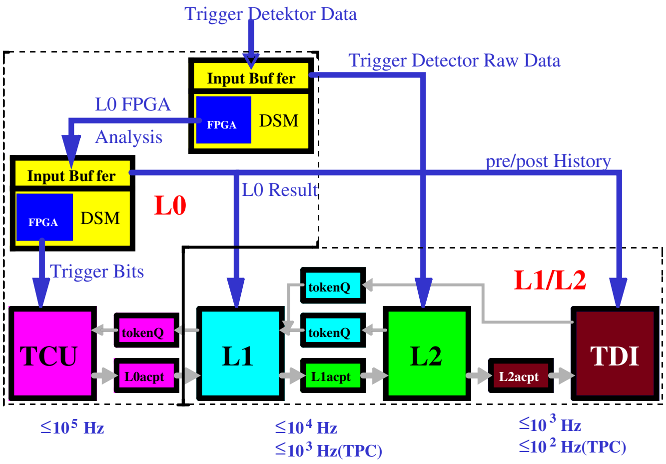

Figure 2: The first layers of the Trigger system

Figure 2 sketches the functional blocks of the high-speed Trigger layers. The L0 trigger decision is based on a hierarchy of *data storage and manipulation boards* (DSM), which use an FPGA each, which can be reconfigured on a run-by-run basis, to perform the required logic steps. The central device of the L0 trigger is the *trigger control unit* (TCU) which ultimately broadcasts L0 triggers based on the DSM result. After a L0 trigger each event is analyzed with increasingly complex analysis and increasingly tight cuts. Essentially each of the next two Trigger layers reduces the event rate on average by an order of magnitude. The required event input rates are stated in figure 2. The available time of each of the L1/L2 analyses is expected to be of the order of 100µs/1ms. The Trigger architecture is scalable to accommodate longer analysis times with more processors.

Trigger Data Flow

Figure 3 sketches the Trigger data flow. The L1 analysis is based on a summary subset of the available trigger data, which amounts to about 100 bytes. Upon a L1 accept one L2 CPU receives the full trigger detector data including the L1 data set (about 1kB). Which subset is being sent to L1 can be configured very elegantly because all DSM boards are memory mapped into the L1/L2 address space. The high L1 trigger rate leaves only 10µs for the complete data transfer of one event. Therefore all DSM boards containing L1 relevant data are required to be in one crate. This requirement allows implementing the VMEP chain block transfer [2], a multi module DMA burst transaction.

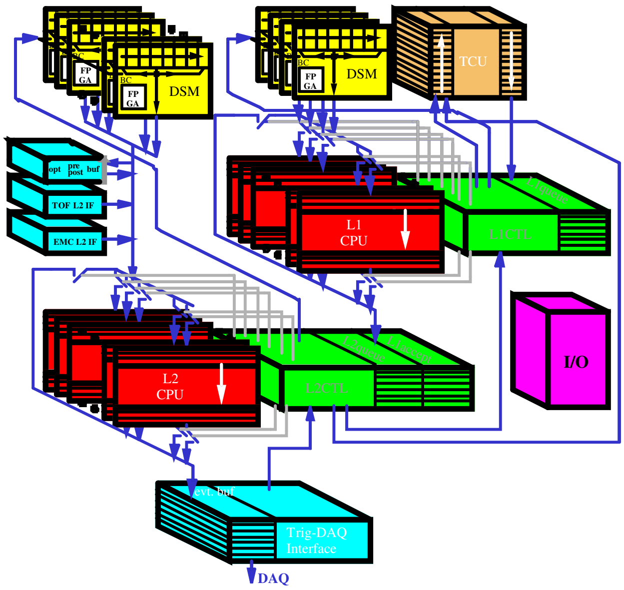

Figure 3: Trigger data flow

A framework was developed that allows adding or upgrading of L1/L2 processors according to later upcoming processing requirements. There is no need for these CPUs to be any particular kind. The Trigger architecture fully supports a heterogenous CPU environment. It is conceivable that different kinds of analysis are analyzing the same event simultaneously on different CPU types. The L1 trigger rate, however does not allow time to make multiple copies of one event within the budget of 10µs. Therefore the L1 and L2 processors have to be implemented as multi processor modules if heterogenous trigger processing on an event by event basis is required. One possible implementation uses VME PMC carrier cards with different PMC (signal) processors. In that scenario the input data will be stored in a local common shared memory, which is accessed through PCI.

Trigger Data Flow Control
Token Principle

The STAR detector setup implements several hundred readout modules, which produce upon receiving a trigger independent data sets that have to be merged into one event. It is essential to have a mechanism that allows to guarantee that an event is compiled of the correct subevents. This is accomplished by distributing a unique 12 bit identifier together with the trigger command, which stays attached to the given event until it is either aborted or completely built and queued for taping. The number of such tokens required is determined by the maximum trigger rates and the maximum latencies allowed within Trigger and the data acquisition system (DAQ). For STAR these are less than 4k. Because these identifiers are meaningless numbers they were called *trigger tokens*.

For every trigger one idle trigger token is used. Trigger tokens are returned to the trigger system upon aborts or after the event was completely built within DAQ. In order to guarantee the uniqueness of the token the STAR trigger system will stop issuing triggers if it runs out of tokens. This scenario resembles global STAR system dead time and can happen only if there are excessively long trigger bursts. Those trigger bursts are statistically possible but very unlikely so that their contribution to deadtime is negligible.

This unique token allows a great simplification of the whole detector architecture including the Trigger system, because the number of outstanding events is limited. Due to the fairly small maximum number of outstanding events (4k) it is possible to design a subdetector system to supply enough buffer space to store 4k outstanding events. Because a trigger token will be returned by DAQ only after the event was completely built any latencies in any subdetector system will cause a given token to be alive for a longer period of time but will not cause any buffer allocation problems or data overrun hazards.

Trigger uses the token concept for the L1/L2 data flow management. All resources within the Trigger system are managed by virtue of the trigger tokens. Internal state tables allow the identification of the state of any event based on its token.

In order to understand the average behavior of that system and to get an idea of the over all system dead time extensive Modsim simulations have been performed [3] based on the trigger token flow. The parameters used in the following are $t_{L0readout}=10\mu s$, $t_{L1readout}=100\mu s$, $t_{L2readout}=100\mu s$, $t_{L1max}=100\mu s$, $t_{L2max}=1 ms$. There are 10 CPUs in each layer. The analysis time was assumed to be a constant. The L0 trigger rate was assumed to be a Poisson distribution with an average of 10^5 events/sec. In order to save simulation time only a small number of tokens was chosen.

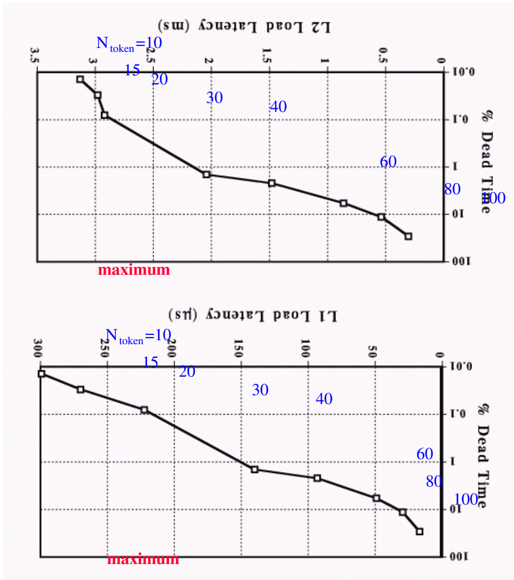

Figure 5: Dead time as function of load latency

Figure 5 shows the dependence of the system dead time as a function of the L1/L2 load latency, the time between the trigger and the start of the analysis. The data points are for different numbers of tokens in each the *Trigger Token Queue* and the *L2 Token Queue* (figure 4). Obviously with an increasing number of tokens increasingly long trigger bursts can be absorbed reducing the dead time. Due to the limited processing capability this is accomplished at the cost of extra event processing latency. Using only 100 out of the maximum 4095 tokens in each trigger layer the dead time imposed by the unavailability of trigger tokens drops to about 10^{-4}.

References

[1] http://www.rhic.bnl.gov/star/starlib/doc/www/star.html

[2] http://www.cern.ch/ECP-ESS/VSC/VSC_Welcome.html

[3] http://www.rhic.bnl.gov/star/starlib/doc/www/html/trg_l/soft_l/data_flow/token.html

Acknowledgments

This work was supported by the US DOE under grant DE-FG03-90ER-40571. V. Lindenstruth was partly supported by the German Humboldt Foundation under the Feodor Lynen program.