Simulation of the DØ Level-3 Trigger System D0Conf-97-4

Gordon Watts [a], Dave Cutts [a], Jan Hoftun [a], and Ray Zeller [b]

[a] *Physics Dept., Brown University, Providence, RI, 02912, USA*

[b] *ZRL, Bristol, RI, 02809, USA*

The upgrade to the DØ detector's data acquisition system and high level trigger has been simulated using the commercial programming language, MODSIM III. This paper contains a brief description of the simulated data acquisition system. The simulation program is described, including its object oriented design and other features which make it simple to simulate a range of configurations. Results are shown for various likely scenarios. The data acquisition system is highly scalable. The minimal version was shown to work well at a rate of 1300 Hz with 250 KByte events. We report on the status of the simulation project and the utility of these simulations.

Key words: DØ; trigger; farm; simulation; MODSIM; Windows NT

1 Introduction

This note describes work done to simulate the DØ Level-3 trigger/data acquisition system in MODSIM III, a discrete-event-simulation programming language. A general description of the DØ Level-3 upgrade can be found in another CHEP'97 paper[1], as well as the DØ upgrade home page[2].

A simulation cannot describe exactly a system of this complexity for a number of reasons: the system is not modeled down to the chip level, the configuration of the system will not be exactly the same as in the simulation, and a number of shortcuts are made to keep simulation processing times reasonable. Also, this simulation does not address failure modes, like corrupt or missing data, although it can expose bottle-necks related to design oversight. Simulating the failure modes is a future project.

There are two benefits to simulating the DØ Level-3 DAQ hardware. First, writing a simulation is instructive as to how the various pieces of Level-3 might fit together. Some untenable configurations were found during the design and writing of the simulation, and this information was fed back into the hardware design process.

The second benefit of a simulation comes, of course, with running it. It is simple to verify that a simulated system works at the designed rates. However, it is informative to push the system until there is significant deadtime to characterize the bottlenecks.

1.1 Level-3 in a Nutshell

Please refer to Reference [1] for a detailed description of the Level 3 upgrade, including figures. A very brief description follows. Each Front End Crate (FEC) generates one block of data per event. Each data block makes its way through the system independent of other data blocks. The data blocks from the same event recombine at their assigned Level-3 processor node. The system makes its routing decisions based on information contained in the data block's header, Level-2 trigger information, and pre-loaded routing tables. The Event Tag Generator (ETG) coordinates the event flow using this information by supervising the assignment of an event to a processing node.

The Level-3 system will be built in stages. That is, a smaller but fully functional system will be built first, and later its capabilities will be increased as more bandwidth is required. Each stage increases the available bandwidth by increasing the number of parallel data paths. The simulation allowed us to determine the capability of each stage, and was a strong influence in deciding which stage to use as the base system.

2 Details of the Simulation

This section describes the MODSIM programming language and then goes on to briefly describe the simulation program written to perform this study.

The MODSIM III programming language is based on the programming languages MODULA-2 (which, in turn, is based on PASCAL) and SIMULA, an object oriented simulation language. The language was first developed by the US Army to do battle simulations. The CACI Products company([3]) wrote MODSIM III, which runs under both NT and UNIX.

The MODSIM programming language is object oriented and discrete-event based. This means the most natural way to program the Level-3 system in MODSIM is as a collection of objects. In general, each object represents a physical part of the Level-3 system or a data packet that moves through the DØ Level-3 system: VBDs, collectors, tokens, data blocks, etc. An attempt is made to keep the code as close to the physical design of the system as possible.

We are not attempting to simulate everything in the hardware: that would require a true VHDL model and simulation. Details of the trade offs can be read in Reference [4]. Many of the approximations made are designed to improve the simulation's efficiency, and are likely to have little effect on the results. However, there are clearly some aspects of the model where better description will provide more accurate predictions. Because of the object oriented design of the simulation program, exploring these and other improvements will be relatively easy to do in the future.

The total size of each event is 250 Kbytes. The size of the data sent by each front end crate is determined by a gaussian distribution with a width 10% of the mean. There are 100 FECs, distributed evenly over the inputs to the DAQ system. There are 48 processors Level-3 multiprocessor farm nodes.

3 Simulation Results

The simulation results are expressed in terms of Level-2 trigger deadtime, not overall system deadtime (which would require simulating the full DØ trigger system to determine).

Figure 1a shows the Level-2 deadtime as a function of average Level-2 accept rate. The minimal configuration (stage A) does well up to 1300 Hz, stage B does well to 2000 Hz, stage C to about 2500 Hz, and stage D to almost 3500 Hz[1]. The first three stages have a bandwidth mismatch between the FEC output and the trigger-computer farm input. Only in stage D are the bandwidths matched.

The current DAQ system's trigger processors operate in serial mode. The processor sits idle while reading in a event, works on the event, discards it, and repeats. The upgraded processors, however, will operate in a parallel mode and the machines will be symmetric multiprocessing computers (4 CPUs). In this mode a new event is read in while the node is processing other events. In the parallel mode, 3 of the CPUs will work on event data while the fourth is left free for administrative tasks. For the simulation, CPU time to work on a single event is determined from an exponential distribution with no cut off. Figure 1b shows the Level-2 deadtime as a function of average CPU time for an event, for the first two stages. The simulations in Figure 1b were run at 1000 Hz Level-2 accept rate. Note that the first two stages are equivalent when the parallel model for the processors is used.

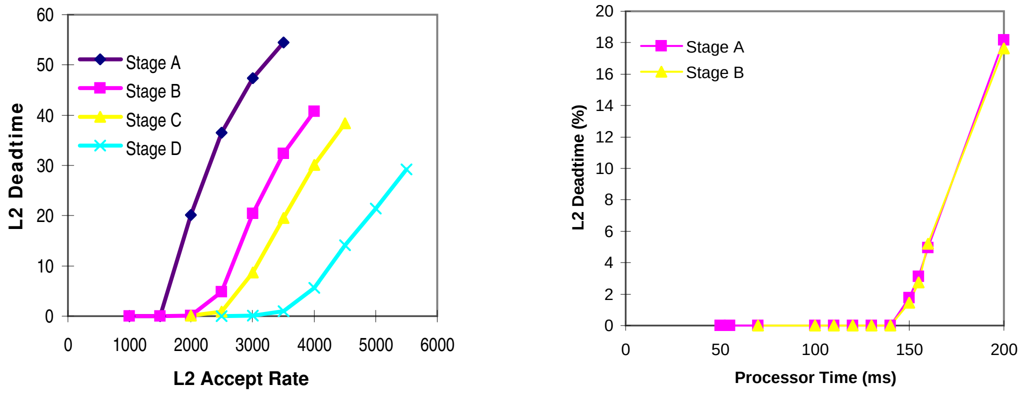

Fig. 1. Level-2 deadtime caused by Level-3 as a function of the Level-2 accept rate. Hardware configuration stage A (diamonds), stage B (squares), stage C (triangles), and stage D (x-es) are shown in a). The Level-2 deadtime as a function of time to process an event is shown in b).

Clearly, many more results can be derived from the simulation. See Reference [4] for more information.

4 Conclusions

We have developed a tool written in the MODSIM III programming language to simulate the Level-3 data acquisition and trigger system for the DØ upgrade. We stressed the simulation beyond the nominal operating points to discover which components in the extreme conditions limited the throughput. The results of the simulation were helpful in identifying bottlenecks in our hardware design.

We expect to improve this tool, as we better understand how and what to simulate, and as we further refine the design of Level-3.

This work is supported in part by U.S. Department of Energy Grant DE-FG02-91ER40688.

References

[1] Dave Cutts, *et. al.* CHEP'97 *The DØ Level-3 Trigger*

[2] `http://d0sgi0.fnal.gov/hardware/upgrade/upgrade.html`

[3] The CACI Products Company: `http://www.caciasl.com/`

[4] Cutts et. al DØNote 2973, *Simulation of the Level 3 DAQ and Trigger*