

A Design for the HFT Read Out

11/27/2005

The design is based on the daughter board designed by Fred Bieser and Robin Gareus and the synchronous cluster identifier and data compression scheme proposed by Leo Greiner. This is the daughter board currently used in the ladder prototype that has been under test. Before going into details here is a list of advantages in the design:

- The daughter board hardware will provide the full data rate and functionality required for a complete first generation 4 ms HFT solution.
- Much of the required VHDL firm ware is running, tested and understood
- The cluster identifier runs at the same speed as digitization providing immediate data compression. Only cluster center addresses are passed on for data storage.
- Data digitization and compression takes the same 4 ms for all events independent of data.
- The cluster identifier and data compression fits well in the FPGA environment. It requires few resources and can be implemented with simple straight forward VHDL coding.
- The design is triggered and fits the standard DAQ design.
- All the hits for an event are stored directly for that event. There are no complications with frame boundaries or hits for an event located in different frames. This is important because file handling software used in STAR data analysis does not have to be altered to accommodate the HFT.
- The latency is 4 ms, but the dead time is 1 ms matching the new TPC system.
- The HFT data size for an event is 90 kBytes, significantly less than the 2 Mbytes for TPC central collision event.

First a brief description of the system. This is followed with a diagram and a list walking through the elements of the design designated in the diagram. And, finally a section on a proposed data storage structure with expected data loads.

An HFT ladder has 10 MIMOSTAR MAPS chips with 640 by 640 pixels on each chip. Each chip is divided in half with two parallel analogue, differential current output buffers. The chips are continuously clocked at 50 MHz rastering repeatedly through all the pixels connecting them to the output buffer. Current to voltage buffers at the end of the ladder drive the analogue signals over twisted pairs (about a meter length) that are connected eventually to the daughter boards. The daughter boards contain an 8 channel ADC, two SRAM chips and an FPGA. The daughter boards will handle digitization, zero suppression and data compression. For each event they will generate a list of hits giving the address of the pixels at the cluster centers. They will also be able to operating in slow diagnostic mode providing amplitudes for all the pixels. They currently operate in the diagnostic mode, but this is not the subject of this discussion. The daughter board with

it's 8 ADC channels can handle 4 MIMOSTAR chips. Using 3 daughter boards per ladder leaves 4 channels unused.

The daughter boards function as follows. A common 50 MHz clock that drives the MIMOSTAR chip drives the ADCs at 50 MSPS digitizing one pixel after the other. Each ADC channel writes a 10 bit digital amplitude to a circular memory buffer repeatedly cycling through the 640 by 320 pixels on half of a MIMOSTAR chip. The ADCs have 12 b capability so we may use more than 10 b.

The baseline voltage of the pixels vary significantly from pixel to pixel, much more than the amplitude of the min-I signal so to extract a signal the pixel must be digitized twice, once before and once after the particle hits the pixel. The signal is obtained from the difference in the two values, Correlated Double Sampling (CDS). This is accomplished in this design by taking the difference between the old value saved on the circular buffer and the current ADC value. This is done before overwriting the circular buffer with the new ADC value. The CDS value or hit amplitude is then checked against a high and low threshold and the result is clocked into a shift register, again with the same common clock.

The shift register is long enough to contain 2 rows of the MAPS plus an additional 3 pixels into the third row. The purpose of the shift register is to provide a simple fast way to slide a 3 by 3 window over every pixel of the MAPS to test for clusters. Ports on the nine desired shift register cells are permanently connected to the cluster sensor (see diagram below). Demanding a high threshold in the center pixel plus at least one low threshold in at least one neighbor pixel should be sufficient to find the center reliably. This can be checked with existing data. If required a more sophisticated algorithm can be used. In addition to finding the center pixel the purpose is to be able to detect min-I hits with 98% efficiency while limiting false hits from noise to a few 10s of hits per cycle through the detector. Since the noise is essentially random, requiring two pixels above threshold instead of just one significantly reduces accidentals. This will also be a good filter against single hot pixels. However, if a hot pixel map is required it can be stored in another bit on the circular buffer and this can be interrogated during CDS.

Back to the cluster sensor function, if the cluster test is positive then the address of the pixel gets recorded into a FIFO, that is if the FIFO is enabled by the trigger. By timing from the trigger a fixed width acceptance window with a fixed delay is generated. The window width is just the time required to cycle through all the pixels once. This allows all the hits associated with the trigger event to be recorded into the FIFO. With this continuous raster scan view a time window provides the simple selection of event hits. There is no complication of event hits spanning separate frames.

Once all of the hits for the event are recorded into the FIFO then the FIFOs for all the channels are read, assembled, and cleared by the mother board and sent over the RORC/DLL to the STAR DAQ system. The event acceptance window for an event is the same for all channels and can be generated at a single location. As soon as the acceptance window is closed the FIFOs are ready for reading.

The system shown in the diagram will have a dead time of a little more than 4 ms, but this is reduced with a relatively simple addition. Instead of one FIFO per channel we will use 5. Each FIFO for the channel will have a separate triggered acceptance window generator. This approach will result in some data duplication, but it has the advantage that each event will be self contained, greatly simplifying file handling during data analysis. Data duplication is not an issue because the data volume is relatively small.

- A. MIMOSTAR MAPS, 640X640 pixels. Continuous parallel raster scan readout of the two halves at 50 MHz pixel rate.
- B. Two SRAM memory chips, 72 bits wide and 256 Kbits deep, rated at 150-333 MHz, operated as a circular buffer addressed by the pixel counter looping over $320 \times 640 = 204,800$ locations. Up to 18 bits available in the memory for each pixel, but only 10 are required for the ADC value plus a possible 11th bit used as a hot pixel marker. Two memory chips to handle 4 MIMOSTAR chips. VHDL code for memory control currently operating. GSI Technology
<http://gsitechnology.com/18MbBurst.htm>
- C. ADC chip with 8 parallel channels of 12 bit ADCs. Each ADC channel continuously digitizes the signals from one half of a MIMOSTAR chip at 50 MSPS. The output of each channel is serial LVDS. The ADCs are currently operating on the daughter board but not at the required data rate. The firmware used was adapted by Robin Gareus from code for another FPGA and is based on work appearing in: <http://direct.xilinx.com/bvdocs/appnotes/xapp774.pdf> Higher speed grades of the FPGA may provide speed sufficient for deserialization at our desired data rate. The ADC on the daughter board is an ADS5270 capable of 40 MSPS. The ADS5271 operates at 50 MSPS, see: Texas Instruments
<http://focus.ti.com/docs/prod/folders/print/ads5270.html>
- D. Correlated Double Sampling (CDS) is accomplished at full readout speed by reading the old pixel ADC value from the circular buffer and subtracting it from the current new ADC value. The new ADC value is written back to the circular buffer over writing the old value.
- E. A digital discriminator viewing the CDS value returns either 0, over low threshold or over high threshold and delivers the result to a shift register.
- F. A cluster detector following Leo Greiner's design uses shift registers to check for clusters centered on each pixel at the full chip readout speed. The signal to noise is supposed to be sufficient to identify clusters just by a single pixel over high threshold. So, a simple center pixel over high threshold with an adjacent vertical or horizontal neighbor over low threshold may provide more than enough sensitivity to cleanly identify cluster centers. The goal is to be 98% efficient for min-I with less than a few tens of false hits per half chip. All 9 pixels, the center and all its adjacent pixels are available for a more complicated sensor algorithm, but the proposed simple scheme may be enough. This also may be sufficient filtering to forgo using a hot pixel map. Note, however, the proposed daughter board architecture lends itself to using a hot pixel map. There is just the added complication of up loading the map to the circular buffer. In the figure the shift registers are shown as 3 separate items, but in practice they may well be implemented as a single register ported as required. The cluster sensor/ shift registers, digital discriminator (E), CDS (D), and ADC channel (C) are repeated 8 times. This should be well within the resources of the Xilinx Vertex-II FPGA, XC2V1000 current used on the daughter board, see:
http://www.xilinx.com/products/silicon_solutions/fpgas/virtex/virtex_ii_platform_fpgas/product_table.htm The selected FPGA has a total of 720 kb Block RAM and 160 kb distributed RAM. The shift registers require only 10 kb.

- G. When the cluster sensor detects a cluster it loads (if trigger enabled) the center pixel address (the pixel counter + a fixed offset) into a FIFO for export. This is the only data that gets exported. The FIFO shown is 18 bits wide as required to contain the pixel address for half the chip. Note the selected data loaded into the FIFO is the same as the address used to access the circular buffer. The expected number of hits on the inner ladders at 10^{27} luminosity is ~ 200 for half a chip so the 2k deep FIFO is generous. This FIFO is repeated 8 times for a total Block RAM requirement of 288 kb. This is well within the 720 kb available on the current daughter board. The system shown will operate triggered with a 4 ms dead time. As discussed a faster 1 ms dead time system will require 5 times as many FIFOs or 1440 kb which is over the XC2V1000 limit. A bigger FPGA, the XC2V3000, however would suffice or the FIFO depth could be reduced by half which is still 5 times the expected load.
- H. The trigger signal which arrives 1 micro second after the event collision of interest is delayed and stretched setting the recording window for storing cluster center addresses into the FIFO. The delay is set to 322 clock cycles less the 1 micro second trigger delay. This total delay is the time required for the first hit pixel of the event to ripple through the shift register to the center of the cluster sensor. The stretch time (acceptance window width) is 204,800 clock cycles, the time required for the last potentially struck pixel of the event to reach the center of the cluster sensor. These numbers can be changed to exclude pixels in the edge rows being counted as cluster centers. In this scheme all the clusters for the triggered collision are recorded. The dead time is roughly 4 ms the time required to read through all the pixels once. Some additional dead time may result in transferring the FIFO data down to the STAR DAQ system. Additional collisions occur during the 4 ms read time and these background clusters are included. A simple variation on this scheme can be used to reduce the dead time so that the HFT can be included with every TPC trigger. This is done by adding four more FIFOs and trigger stretchers for each ADC/cluster finder channel. When a FIFO is still busy accepting cluster addresses from one event and another trigger arrives it can be processed by the next available FIFO/trigger stretcher. The acceptance windows will partially overlap, but each event will carry all the clusters associated with the event, greatly simplifying STAR data analysis.

In the design outlined some event building and trigger handling gets done on the mother board. This differs from the current ladder prototype design where there are no FPGAs on the mother board. All data communication is handled by the daughter board FPGA. The advantage in this approach of using the daughter board FPGA for all the functions is the simplification of VHDL code development. Having all the code in one place for debugging and maintenance is certainly desirable. More thought will be given to following this example. Perhaps the trigger functions and event building can also be accomplished in the daughter board FPGA.

In any case additional work is required on for the data connection to the outside. Currently the connection is via Robin Gareus' pci connection protocol and his slow connection code scsn. We want to use instead the RORC/DLL connection that has

become the STAR standard. It is also hoped to have a USB connection to LabView providing more portable operation for testing and debugging.

Data rates, zero suppression, data structure

The data reduction achieved in the daughter board with this design is significant. For a half a chip, before CDS the data rate is 65 MB/s after CDS 50 MB/s after cluster identification the rate depends on hit density and luminosity. For chips at the inner radius and a luminosity $= 10^{27}$ the $\frac{1}{2}$ chip data rate is 0.5 MB/s for an event rate of 1 kHz. This is a reduction of over 100 for the inner, high exposure chips. The reduction is more for the outer layer.

The total data volume depends on the data structure used. A proposed data structure with example population is shown in Fig. 2. Half chip addresses and ladder addresses are distinguished from pixel addresses by adding constants to make them larger than the maximum pixel address of 204,799. If it is desirable to format the data into 3 bytes instead of 18 bits. Then the additional bits may be used to distinguish the 3 address types directly.

Data structure for HFT

18b wide

Event/Header	
ladder address 0	range 0-23
half chip address 0	range 0-19
pixel address 157,921	range 0-204,799
pixel address 159,203	
.	
.	
.	
.	
.	
.	
pixel address 142,888	
pixel address 148,321	
half chip address 1	
pixel address 155,423	
pixel address 155,231	
.	
.	
.	
.	
.	
ladder address 1	
half chip address 0	
.	
.	

Fig. 2

Some numbers:

Item	number
bits/address	18
inner ladders	6
outer ladders	18
half chips per ladder	20
ave hits/half chip, inner, $L = 10^{27}$	200
ave hits/half chip, outer, $L = 10^{27}$	40

Only the hit addresses generate significant data volume. The totals are:

Item	number
Event Size	90 kBytes
Data Rate at 1 KHz event rate	90 Mbytes/sec

The HFT event size is significantly smaller than the TPC which has an event size of 2 MBytes for central Au+Au (need to compare with the average event size)

The proposed data format can be compared with one that rather than saved hit addresses uses ordered bits to represent all of the pixels. Pixels centering a cluster get a 1 and all others are 0. The event storage size in this case is 12 MBytes, over 100 times larger than the proposed scheme.