

# THE STAR CLUSTER-FINDER ASIC<sup>1</sup>

M. Botlo<sup>2,a</sup>, M. J. LeVine<sup>2</sup>, R.A. Scheetz<sup>2</sup>, M.W. Schulz<sup>2</sup>, P. Short<sup>3</sup>, J. Woods<sup>3</sup>, and D. Crosetto<sup>4</sup>

<sup>2</sup>Brookhaven National Laboratory, Upton, NY 11973

<sup>3</sup>InnovASIC, Inc., Albuquerque, NM 87107

<sup>4</sup>Bonner Nuclear Laboratory, Rice University, Houston, TX 77251

## Abstract

The STAR experiment reads out a TPC and an SVT (silicon vertex tracker), both of which require in-line pedestal subtraction, compression of ADC values from 10-bit to 8-bit, and location of time sequences representing responses to charged-particle tracks. The STAR cluster finder ASIC responds to all of these needs. Pedestal subtraction and compression are performed using lookup tables in attached RAM. We describe its design and implementation, as well as testing methodology and results of tests performed on foundry prototypes.

## I. INTRODUCTION

STAR is a large TPC-based experiment at RHIC, the relativistic heavy ion collider at Brookhaven National Laboratory. The largest detector in STAR is a TPC, with 140,000 analog channels (pads). Each TPC anode is sampled in 512 time bins, which are stored in analog form in a switched-capacitor array and subsequently digitized on the detector to a 10-bit precision. The digitized information is transported off the detector via 144 fiber optic links to the data acquisition area.

Even in the most complex collisions to be studied, the TPC time bin occupancy is about 10%. Suppression of pedestal-only time bins is required to reduce the event size. Additionally, the digitized data are to be used for level 3 trigger calculations. For these purposes, sequences of above-pedestal time bins need to be identified in a few msec.

The 10-bit range of the linear ADC's has been chosen for dynamic range; requirements for precision can be met with 8 bits. In order to economically store these data, a nonlinear compression is required to transform them into 8-bit quantities.

The STAR Cluster-finder ASIC was developed in response to these requirements for both the TPC and the SVT (see below). It is meant to perform the following:

- pedestal subtraction
- compression of 10-bit ADC values to 8-bits
- compilation of pointers to above-threshold time bin sequences for each pad

Table lookup was chosen for compression of 10-bit values for reasons of flexibility.

## II. FUNCTIONAL DESCRIPTION

The Cluster-finder ASIC has four ports (Fig. 1):

- an 8-bit microprocessor interface for control and programming
- a 10-bit input port for ADC data
- an 8-bit output port with elasticity buffer
- an 8-bit interface to SRAM

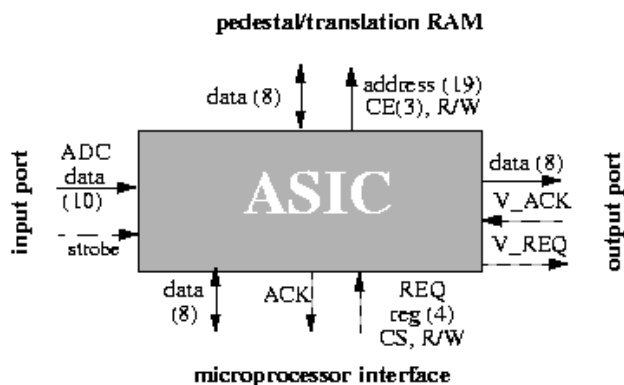


Figure 1: The STAR ASIC interfaces

The ASIC's role in processing a TPC event, to be described, is shown schematically in Fig. 2. Differences between the SVT and TPC are summarized later. The ASIC is responsible for 64 TPC pads. It accepts a new ADC value at its input port at intervals larger than 150 nsec. Each incoming ADC value results in an 8-bit pedestal being fetched from the attached SRAM, whose address is determined by the ADC sequence number. The result is unsigned; i.e., a negative result is replaced by zero. The pedestal-subtracted ADC value, together with the pad number, form an index into a lookup table, which is implemented in the same SRAM which holds the pedestal values. The inclusion of the pad number in the index allows the compression to also function as a pad-to-pad gain compensation; that is, there is a separate 1024-entry table for each pad.

For historical reasons, the ASIC was designed to accommodate 1024 time bins per pad, instead of the 512 time bins which now characterize the TPC.

The pedestal-subtracted ADC value is now replaced by the result of the table lookup described above. This 8-bit value is then passed to a 16-entry FIFO which forms an elasticity buffer at the output port. This elasticity is required since the data leaving the ASIC are destined for the sequential port of video RAM (VRAM), which can produce worst-case delays in our system of up to 150 nsec.

<sup>1</sup>This research was supported by the U.S. Department of Energy under Contract No. DE-AC02-76CH00016.

<sup>a</sup>Currently at Morgan, Stanley, Inc.

Simultaneously, the output of the lookup table is passed to a cluster processor, which determines whether the current ADC value is part of a valid time bin sequence. A valid time bin sequence is defined by four parameters, programmable via the microprocessor port:

- lower threshold
- higher threshold
- minimum number of consecutive time bins exceeding the lower threshold
- minimum number of (not necessarily consecutive) time bins exceeding the upper threshold

The cluster processor stores the first and last time bins for each sequence of each pad that meets the definition imposed by the four parameters. These *cluster pointers* are stored in on-chip RAM which is large enough to store pointers to 31 sequences for each of the 64 TPC pads. [The maximum TPC occupancy corresponds to  $\sim 3$  sequences per pad.] It is envisioned that the lower threshold will be set quite close to the noise [e.g., twice sigma], in order that the reported sequence will include adequate ‘tails’.

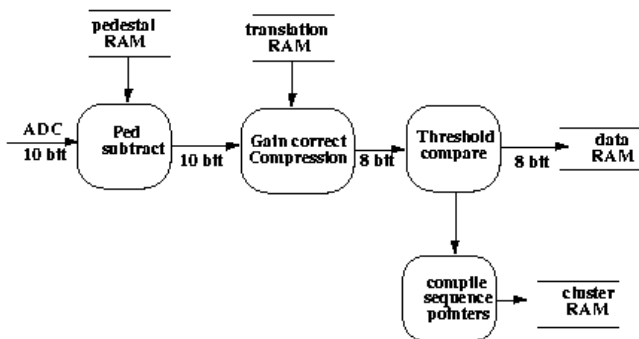


Figure 2: The ASIC processing model

Following the processing of the last ADC value, the ASIC asserts a signal `END_ACK`, intended for external logic managing the ASIC-to-VRAM transfer. This logic then asserts `CLUSTER_DUMP`, which results in the ASIC emptying its internal cluster pointer RAM via the exit port. Each cluster is represented by a pair of 10-bit values, which are broken into 8-bit pieces before passing through the exit port.

The ADC data arrive at the ASIC one time bin at a time; i.e., the data for a single time bin arrive in sequence, for [pad 0, ..., pad 63], followed by the data for time bin 1, etc. The sequences that are to be identified are sequences of consecutive time bins for each pad, separately. This complicates considerably the implementation of the cluster finder, since temporary storage has to be provided for, e.g., numbers of time bins exceeding the high threshold, or numbers of consecutive time bins exceeding the lower threshold, for each pad.

In order to avoid an (expensive) dual-port implementation of the attached SRAM (pedestal, translation memory), the ASIC provides read/write access to all memory locations via its microprocessor port. Three address registers which are used to provide an initial address and a data register implement sequential access to this SRAM. For the TPC application, a single 128 KByte SRAM is required; for the SVT, 512 KByte are needed.

The 150 nsec interval between arriving ADC data resulted in a choice of a 66 MHz clock frequency to drive the ASIC’s almost completely synchronous circuitry. This interval also determines the attached SRAM access time requirement, 30 nsec.

Flow control at the exit port is carried out via a pair of control signals, `V_REQ` and `V_ACK`. Two bits in a status register provide assurance that the event passed through the ASIC’s exit port without incident: one bit indicates whether there are orphan data in the FIFO at the end of the event (excess data, or too few exit handshakes), while a second indicates that there was a data overrun in the FIFO during the processing of the event.

An asynchronous clear allows the level 2 trigger (event abort) to clear the ASIC for a new event without intervention by the controlling CPU.

In addition to the TPC, STAR also includes a silicon drift detector (SVT) with 100,000 analog channels (anodes). The SVT data are stored in 128 time bins. The ASIC also has to serve as cluster finder for the SVT detector. It was decided, for economic reasons, to require that the ASIC serve 256 SVT anodes instead of the TPC’s 64 pads. In order to retain the cluster-pointer RAM size, a maximum of 8 sequences can be registered in SVT mode. This suits the SVT detector; its occupancy is expected to correspond to  $< 1$  sequence per anode. Some rearrangement of the ASIC’s internal address busses made it possible to utilize for the most part the same resources on the ASIC for either configuration. To accommodate the two modes of operation, a `MAX_TIME BIN` register allows the ASIC to be configured for operation with any number of time bins per pad, from 3 to 1023. To accommodate the two modes of operation, a `MAX_TIME BIN` register allows the ASIC to be configured for operation with any number of time bins per pad, from 3 to 1023.

The SVT application also requires that an externally supplied value be used for the starting time bin number used for indexing the pedestal array. A separate clock input (pedestal offset clock) serves to latch this value, which is entered through the ADC input port.

### III. TESTABILITY

The attached SRAM can be written and read via the microprocessor port. The ASIC provides a `TEST` register which can be written by the microprocessor, whose contents can be clocked into the processing stream when a bit is written to the `PULSE` register. These features allow the CPU to verify the correct operation of the ASIC subsystem, even in the absence of an external source of ADC data.

In addition, the on-chip cluster RAM is equipped with a built-in self test (BIST), allowing it to be tested at the foundry at a 1Mhz clock rate, sufficient to detect shorted or open transistors in every memory cell.

### IV. DESIGN

The development flow for the cluster-finder ASIC was divided into 4 phases: specification, capture, functional verification and synthesis.

## A. Specification

A baseline specification was established and approved before coding began. The specification provided the criteria from which verification code was written.

## B. Capture Phase

The digital capture phase of the cluster-finder development consisted of two primary tasks: partitioning and code development. The development approach utilized pseudo top-down VHDL-based methodology. The capture phase was performed on PCs, using the Model Technologies simulator to compile and simulate each partition. The specification provided the basis for the partitioning, coding, and most importantly, the verification of the design. The design was not behaviorally modeled. The intent was to go from specification to code which could be readily synthesized.

## C. Functional verification

The VHDL design had to be thoroughly verified to meet the functional requirements as outlined in the specification. In order to accomplish this, a VHDL test bench was written that used mnemonic codes to generate test vectors for the ASIC. Functional verification was performed using the Model Technologies simulator on a PC. For larger simulations, the Mentor Graphics workstation running QuickHDL at InnovASIC's facilities was used.

Additionally, An independent model/testbench written in "C" was used to generate ADC data and find clusters. The output from this simulation was used as input to the VHDL testbench and the results compared.

## D. Synthesis Phase

The synthesis phase converted the VHDL descriptions into gate level descriptions in the vendor's library. Synthesis was performed using the Mentor Autologic tool at InnovASIC's facility. The output of Autologic was a Vital Compliant VHDL netlist, which allowed the vendor to supply a standard delay format (SDF) file for back-annotated simulations. Gate-level verification was performed using the same set of tests generated in the functional verification phase of the program. No additional tests were developed for gate-level verification.

## V. VHDL MODEL DEVELOPMENT GUIDELINES

The complexity of the cluster-finder ASIC mandated synchronous design practices. Logic design of cluster-finder was synchronous, with positive edge-triggered storage elements operating from a primary input clock source. Lower frequency clock signals were not derived from the primary clock. Instead, the primary clock was qualified at lower frequency storage elements by an appropriate strobe which was the output of a synchronous divide-by-n circuit. Where a secondary clock source was absolutely necessary, then events which passed between two blocks of logic clocked by different sources were synchronized by a metastable-hardened synchronizer.

Asynchronous inputs to storage elements were used only for circuit initialization. In this case, all asynchronous inputs were derived from a buffer tree whose input was synchronized to the system clock. With this method, the point at which asynchronous resets arrive at clocked elements could be carefully placed within the clock cycle to avoid any proximity to a rising edge.

These are far-reaching design constraints which prohibited the following logic structures:

- Cross-coupled gates or other combinatorial feedback.
- Gated clocks.
- Cascaded combinatorial delay elements used to produce pulses.
- Derived flip-flop asynchronous set or clear signals.

In addition, level sensitive storage elements such as transparent latches were not used unless pre-defined interfaces at the ASIC primary I/O pins explicitly demand them. Primary input or output events which must occur on the edge of a specified external event represented an exception to these rules.

A large percentage of logic design failures for ASICs can be attributed to asynchronous logic. The performance of asynchronous logic is highly unpredictable over the operating parameters of temperature, voltage, and process variation. If a synchronous design operates correctly with best-case and worst-case timing delays, then it is guaranteed to operate correctly at every point on the delay curve between those extremes. An asynchronous design, however, may perform correctly over 95% of the interval between best and worst-case timing. The region in which it fails or demonstrates erratic behavior within that interval is not predictable. Maximum, minimum, and typical timing cases are not guaranteed to expose the problem. Commercial integrated circuits which utilize asynchronous or self-timed logic are carefully crafted at the physical layout level. ASICs, which are susceptible to the random interconnect lengths produced by autorouters, must not include these logic design techniques.

## VI. IMPLEMENTATION

The completed ASIC design was submitted to the chosen foundry, AMI, (American Microsystems, Inc.), where it was implemented as a gate array in 0.8 $\mu$  CMOS technology. The design requires approximately 57000 gates, including the 40 kbit of on-chip RAM. Following completion of final simulations, a prototype run of 50 ASICs was fabricated and delivered to Brookhaven for testing.

Based on detailed foundry estimates, the ASIC consumes 965 mW, of which 850 mW is attributable to the core, the remainder being dissipated in I/O.

## VII. ASIC TESTER

A VME tester card specific to the STAR ASIC was designed and constructed. It is intended to permit exhaustive testing of the ASIC as design clock and data rates, with test patterns supplied over VME. It provides a ZIF (zero insertion force) socket for the ASIC, as well as a generous allotment of

logic analyzer pod sites allowing observation of all external ASIC signals. It consists of four basic blocks.

### A. VME interface

The board is implemented as a VME D16 slave/interrupter, using PLX VME2000/3000 interface chips with most of the remaining logic in a Lattice ispLSI 2096-125 field-programmable CPLD.

### B. Input memory

The input memory is SRAM, large enough for a complete event (64 KB). It is accessible from VME for both read and write.

### C. Sequencer

The sequencer, started by an access to the board's control register, cycles through the entire event by incrementing the input memory address while clocking the ASIC's input data strobe at the design rate of 6.6 MHz.

### D. Output memory.

The output memory is identical to the input memory, except that its address is incremented by the V\_REQ/V\_ACK handshake with the ASIC's exit port.

## VIII. TEST PROCEDURES

The following *ansatz* was used to test the ASIC's functional behavior. The first battery of tests was intended to test the accessibility of the registers of the ASIC. This step included a test of the correctness of various registers after state changes triggered by signals like CLEAR, RESET and DAQ\_START. The number of possible combination of values that can be written into the registers is small enough to carry out a test that covers all possible combinations. The result of this test was that all functional registers worked as expected. The complete register test was carried out only on 2 sample ASICs, since the next steps in the test chain were expected to reveal any problems related to the registers.

The second step in testing used algorithmically constructed test data for the ADC input, the pedestal data and the translation tables. These patterns were processed by the ASIC using different sets of configurations of the ASIC. The resulting output data were then compared to the expected results (computed by a behavioral simulation of the ASIC). Due to the large combinatorial space it was not realistically possible to test all possible combinations. Instead the data were constructed with two different aims in mind:

- a) construct data similar to the expected data
- b) construct data that probed parts of the boundaries of the design.

A sequence of TPC-like pulses were generated to span different parts of the dynamic range of the ADCs. These patterns were then processed using different sets of pedestal and translation data (constant values, ramps,...). To check whether typical errors of ADCs (missing digitizations) are handled correctly a second sequence of TPC like pulses was

generated with some single channels set to 0. The effect of the pedestal offset used by the ASIC in the SVT mode was tested using different offset values.

Probing boundaries of the design was done by making guesses in which situations counters might overflow or certain conditions would not be handled correctly. Some examples might illustrate this approach: It was tested whether a hit was found correctly that included all time bins. To test the correct handling of the beginning and ending of sequences patterns have been constructed that start or end at the beginning or ending. The effect of critical numbers (powers of 2) has been checked by testing for hits which have a length of  $2^n$  and  $2^{n+1}$ , locating them at different positions in the time bin space. For the test of the SVT mode pedestal offsets of 0 and corresponding to the maximum number of time bins were used to test the boundaries. However, using too much knowledge of the internal working of a device during a test process can result in neglecting simple tests. To avoid this situation fairly general patterns (ramps, levels) have been produced and tested.

## IX. RESULTS

In general, the behavior of the ASIC in these tests was as expected, with two exceptions:

- The selection of the maximum length (16 time bins) for the sequence of consecutive samples above the lower threshold has the consequence that the ASIC fails to find any clusters. Since the average length of a TPC hit is about 7 it is simply possible to avoid the use of the value 16. A rerun of the simulation of the hardware design of the ASIC confirmed that this behavior is part of the design.
- The second unexpected behavior affects sequences which include the last time bin for a pad. For some specific patterns the ASIC returns a hit which is one time bin too long. This can in practice be neglected since no data corruption occurs and a single additional ADC value for a small subsample of all hits has no significant effect on the data volume or interpretation.

An additional small problem was encountered. For some patterns the very first data sample that was processed by the ASIC was corrupted. At this point it is not clear whether this problem has its source in the ASIC or in the tester board.

The above tests have been carried out a few thousand times using 2 ASICs. A reduced set of tests sets have been used to test 15 ASIC samples from the prototype series. No differences between the samples have been observed.

To exercise the ASIC at full speed with the maximal possible repetition rate the ASIC was tested using a static test pattern and performing only a coarse check of the output data. The ASIC was sent through the data processing and cluster dumping cycle for roughly 30 minutes. This time is sufficient to reach a stable temperature of the ASIC and test its reliability under conditions similar to the experiment. No problems were observed during this test.

The full test of an ASIC described here takes about 1.5 hour. The full test is therefore not practical for testing large quantities (>100) of chips, nor is it necessary. Most of the

testing described here was intended to expose design errors, not fabrication faults. Reduced versions of the test need between 30 sec and 15 minutes. Even these will be performed only on sample populations of the production chips.

## X. CONCLUSION

With the exception of two minor problems, the ASICs behave as expected. Production quantities of the ASIC are being fabricated.