Optimizing the Readout and DAQ Systems for the HyperCP Experiment

Christopher G. White, Timothy D. Jones, William Luebke, Yen-Chu Chen, Woon-Seng Choong, Daniel M. Kaplan, Ken Nelson, Howard A. Rubin, and Piotr Zyla

Abstract-- The 1999 run of Fermilab E871 (HyperCP) featured an upgraded data acquisition system which recorded 80 terabytes of data on 19,000 8mm tapes. Data was collected with an average trigger rate near 100 kHz. The readout electronics digitized data from 21,000 input channels with a nearly fixed deadtime of 1.7 micro-seconds. The typical livetime was 80%. The typical instantaneous rate of data generated by the HyperCP spectrometer was 36 MB/s. These data were written to tape at a sustained rate of 27 MB/s.

I. INTRODUCTION

The origin and nature of CP violation is a mystery with CP violation having been observed only in decays of neutral K mesons. Therefore, it is of interest to search as sensitively as possible for CP violation occurring outside the kaon sector [1]. The HyperCP experiment aims to achieve 1×10^{-4} sensitivity to possible hyperon CP violation [2]. This is two orders of magnitude beyond the most sensitive current limit [3]. A precision measurement at this level requires over one billion fully reconstructed cascade decays. Accounting for trigger efficiencies, this translates into more than 100 billion triggers digitized and recorded onto magnetic tape.

In order to achieve our goals, a high-rate, high-speed data acquisition system was built and operated by the HyperCP collaboration. The first run of HyperCP took place between March and September 1997. This run accumulated 38 terabytes of data on 9376 data tapes, corresponding to 0.5 billion reconstructed cascade decays. The data acquisition system for Run I is detailed elsewhere [4].

A second fixed target run was approved by Fermilab; however, the running conditions were changed from 19 seconds of spill every 61 seconds to 40 seconds of spill every 81 seconds. This change necessitated an upgrade of the data acquisition system. The nature and detail of these changes will be discussed here.

The HyperCP spectrometer is shown schematically in Figure 1. An intense unpolarized beam of charged hyperons was produced at 0 degrees by 800 GeV protons striking a metal target, with the secondaries momentum-selected by means of a curved magnetic channel set to 160 GeV/c with a 25% FWHM momentum bite. Following a 13m evacuated decay pipe, the hyperon decay products were detected in a high-rate magnetic spectrometer using multi-wire proportional chambers (MWPCs). To switch between hyperon and antihyperon running, the polarities of the hyperon channel and analyzing magnets were reversed, selecting either positive or negative secondary beam. To maintain equal rates in the detectors, different target lengths were used for the two modes. This approach also resulted in nearly identical trigger rates between positive and negative running.

A simple trigger was employed to minimize potential systematic effects that might mimic a CP asymmetry. Our primary physics trigger ("cascade trigger") required a coincidence of at least one particle on the right of the spectrometer with at least one on the left. This was done using the "pion" and "proton" hodoscopes, which were located sufficiently far downstream of the analyzing magnets that the proton and pions from the hyperon decay were all separated from the secondary beam. To suppress muon and low-energy backgrounds, the cascade trigger also required a minimum energy deposit in the hadron calorimeter. Nevertheless the trigger rate was dominated by background due to interactions of beam particles with the spectrometer material.

The muon detection system consisted of proportional-tube planes interspersed with iron absorber. Muon triggers were formed with signals from four planes of scintillation hodoscopes located between and behind the absorbers. These data provided sensitivity to rare decays with muons in the final state.

II. RUN I DAQ LIMITATIONS

The duty factor is defined as the ratio of the time with extracted beam on target divided by the total spill cycle time. During Run I, data was collected for 19 seconds out of every

Manuscript received November 5, 2000. This work was supported in part by the U.S. Department of Energy.

Christopher G. White, Daniel M. Kaplan, William Luebke, and Howard Rubin are with the Illinois Institute of Technology, Chicago, IL 60616 USA (telephone: 312-567-3734, e-mail: whitec @iit.edu).

Tim Jones, Woon-Seng Choong, and Piotr Zyla are with Lawrence Berkeley National Laboratory, Berkeley, CA 94720 USA (telephone: 510-486-5421).

Yen-Chu Chen is with Acedemia Sinica, Taipei, Taiwan (telephone: 630-840-3225).

Ken Nelson is with the University of Virginia, Charlottesville, VA 22901 USA (telephone: 804-924-3781)

61 seconds, resulting in a duty factor of 0.31. The instantaneous data-rate generated by the spectrometer was limited by the DAQ throughput (the rate at which data was written to tape) and equaled the throughput multiplied by the

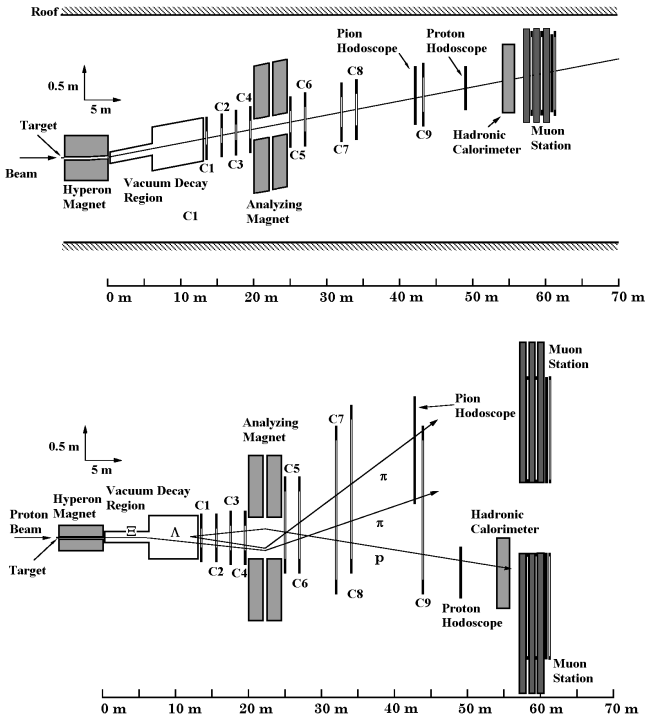

Fig. 1. The HyperCP spectrometer.

inverse of the duty factor. This was true provided that the excess data was safely stored in a buffer (called the event-buffer) and written to tape before the next period of extracted beam. The Run I data-to-tape rate was typically no more than 12.5 MB/s, limiting the instantaneous rate of data collection to 38 MB/s. Given these rates, the minimum required event-buffer size was 500 MB. The actual size of the event-buffer was 960 MB. For an average event size of 600 bytes, the maximum allowed inhibited trigger rate was 63 kHz. Sustained data-rates greater than 38 MB/s resulted in data loss due to the event-buffer filling to capacity.

Data loss occurred due to front-end deadtime. While digitizing spectrometer data, the front-end latch system inhibited future triggers, and was thus dead to new events. The Run I readout time (deadtime) was proportional to the number of hits recorded per event, which was proportional to the primary beam intensity and spill quality. The dead-time fraction equals the ratio of events lost due to deadtime to the total number of triggers generated within the spectrometer. The live-time fraction equals one minus the dead-time fraction. A typical Run I live-time fraction was 70%.

The primary beam intensity during Run I was typically set to maximize the number of events written to tape, corresponding to an instantaneous data rate near 38 MB/s. The DAQ wrote tape continuously and the event buffers were rarely empty. An increase in the number of events written to tape required either an increase in throughput or a decrease in the average event size. It should be noted that a decrease in the deadtime alone would not have significantly increased the number of events written to tape since the throughput bandwidth was already fully occupied. We ran during Run I with all available bandwidth and would have used more had it been available.

III. Upgrade Design Goals

The purpose of the HyperCP spectrometer and DAQ system was to digitize and record as many charged cascade and anti-cascade decays as possible. The number of produced hyperons increased with the primary beam intensity; however, the event size, trigger rate, and deadtime also increased. Furthermore, the yield of useful, reconstructable cascade decays per cascade trigger was found to drop with increasing primary beam intensity. In general, the optimum running conditions did not require running with the maximum allowed primary beam intensity; however, optimized running required an increase in the primary beam intensity over and above that used during Run I.

In addition, following Run I, the Beams Division at Fermilab announced that the duty factor for Run II would change from 19 seconds of spill per 61 seconds to 40 seconds of spill every 80 seconds. A duty factor of 0.5 implied a reduction in the maximum instantaneous data rate to no more than 25 MB/s, assuming no changes to the HyperCP spectrometer and DAQ system, thus pushing us further from optimum running. An upgrade of the data acquisition system was necessary to maximize the number of useful physics events written to tape during Run II. Therefore, our first goal was a doubling of the data-to-tape rate to accommodate an increase in the primary beam intensity. We also sought to decrease the deadtime since the number of events to tape increased with decreased deadtime (assuming sufficient headroom in the data-to-tape capability). Secondary goals included a decrease in the average event size and an increase in spill-buffer size.

The minimum throughput rate and event buffer size are shown in Table I as a function of the trigger rate and average event size. Notice that an event-buffer upgrade was necessary only for high trigger rates, and also notice that the minimum required throughput decreased with decreasing event size. All possible upgrade paths were pursued to obtain maximum headroom and flexibility during the run, and as a hedge against unforeseen problems.

TABLE I
PROJECTED MINIMUM THROUGHPUT AND EVENT BUFFER SIZE FOR THE GIVEN TRIGGER RATE AND EVENT SIZE ASSUMING A DUTY FACTOR OF 0.5. THE LISTED TRIGGER RATE REFERS TO ACCEPTED TRIGGERS. THE EVENT SIZE IS ESTIMATED FROM RUN I DATA EITHER WITH OR WITHOUT DATA COMPRESSION.

Trigger Rate	Event Size	Throughput	Event Buffer
50 kHz	550 bytes	14 MB/s	0.56 GB
75 kHz	600 bytes	23 MB/s	0.92 GB
100 kHz	650 bytes	33 MB/s	1.32 GB
120 kHz	700 bytes	42 MB/s	1.68 GB
50 kHz	400 bytes	10 MB/s	0.40 GB
75 kHz	430 bytes	17 MB/s	0.68 GB
100 kHz	460 bytes	23 MB/s	0.92 GB
120 kHz	490 bytes	29 MB/s	1.18 GB

IV. SPECTROMETER CHANGES

The spectrometer remained relatively unchanged between the 1997 and 1999 runs. Chamber C9 (a chamber originally intended to be used as a spare) was installed and partially instrumented for the 1999 run, resulting in 988 additional MWPC channels. The proton hodoscope was modified, and a new beam-hodoscope was added downstream of the muon station. In addition, a 32-bit time-stamp and a 16-bit flag module were added to the readout stream. In total, the 1999 run encoded data from 19,684 MWPC channels, 832 muon station channels, ten 14-bit ADCs (stored as 16-bit data words), and 288 bits of fixed-format data. The muon and MWPC hits were encoded as 16-bit data words.

Various minor improvements were made to the spectrometer, among which was the change to a fast gas mixture (CF4-isobutane) in all but one MWPC. In 1997 only the four upstream chambers used the fast gas. This enabled us to tighten the timing and reduce the number of out-of-time hits.

The most significant change to the spectrometer, however, was the movement of the target with respect to the collimator entrance. Detailed studies of the Run I data indicated that the target was not correctly positioned along the central orbit of the collimator. The target was moved both horizontally and vertically with respect to the collimator entrance. As such, the correlations between the primary beam intensity and the event size and trigger rates changed between Runs I and II.

V. FRONT-END UPGRADE

A. Re-organization and Additional Channels

Two distinct latch systems were employed during Run I. The FEM system [4] was used to latch and digitize the muon station data, while the Nevis system [5] was used for everything else. Data from these systems were sent along five parallel data paths linking the front-end with the event buffer (situated with the VME event-building and tape-writing system). The FEM and Nevis systems were incompatible resulting in unbalanced data paths. The collaboration decided to standardize the front-end latch system by expanding the Nevis system to include the muon station. Furthermore, as discussed above, an additional 988 MWPC channels needed to be read out. Consequently, four additional Nevis system latch crates were installed and instrumented for Run II. There was insufficient time and money to produce new crates such that old and less reliable units were used to instrument the additional channels. Finally, the entire Nevis system was re-organized to increase path balance thereby increasing overall readout efficiency. A detailed description of the Nevis system as employed during Run II can be found at [6].

B. Deadtime Reduction

The emphasis of the front-end hardware upgrade was to reduce deadtime and event size, and to re-package the raw data for more efficient event building. The chief method for deadtime reduction was the implementation of new encoders with internal buffering. The old encoders required one readout-clock cycle per latched hit. The buffered encoder read an entire latch card per clock cycle, storing the 32-bit hit pattern in 256-deep, 7ns CMOS FIFOs. Readout clock cycles were saved whenever a card had more than one hit. We estimated that the number of clock cycles required to read-out the latch system would be reduced on average by a factor of 2.7 (see Fig. 2). This reduction has been observed. During Run I, front-end buffering occurred after the encoding of latch hits. Furthermore, encoding and buffering had previously been done via two separate modules. The BE replaced these with a single unit, thereby simplifying the system.

The clock period was increased from 75 ns to 85 ns between Runs I and II. Newly manufactured crates ran reliably with a 50 ns clock period; however, the HyperCP latch system relied upon an assortment of crates manufactured at different times for various experiments. A clock period of 85 ns was chosen based on the reliable performance of the slowest crate.

Triggers were inhibited during readout by a "trigger-hold" signal. Due to variations in signal-cable lengths, detector positions within the spectrometer, and differing setup requirements (in particular, the conversion time required by the ADCs), the various readout crates functioned asynchronously (i.e. readout of data into the Buffered Encoders did not occur simultaneously throughout the system). During Run I, the trigger-hold signal was generated through a logic OR of the encoder Busy lines. Due to the asynchronous nature of the readout, a 300 ns delay was added to the trailing edge of the trigger-hold signal to prevent holes in trigger-hold. Furthermore, during Run I, data from different events mixed whenever an encoder missed or received an extra Start pulse (due to a glitch). The Trigger Release Module (TRM) streamlined the System-

Busy signal, reduced overhead associated with releasing the trigger inhibit, provided error trapping to prevent mixing of data between events, and increased diagnostic capabilities with regards to the latch readout. The TRM saved about 400 ns in deadtime per event and successfully prevented data mixing between unrelated events.

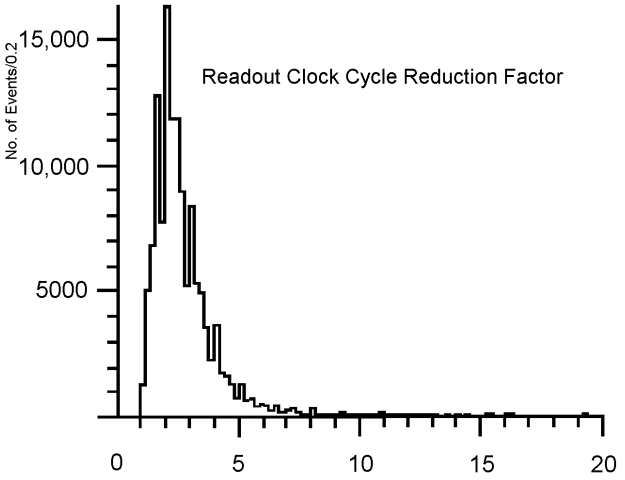

Fig. 2. This figure displays the event-by-event ratio of the number of latch hits within the readout crate with the most hits to the number of cards with hits in the crate with the most hit cards. This ratio is the factor by which the total number of readout clock cycles will be reduced due to implementation of the Buffered Encoder. The average for this distribution is 2.7. This plot was derived from Run I data.

The Buffered Encoder and Trigger Release Module met all expectations for reliability and performance. During the 1997 run, the average deadtime per event was more than 3.5 microseconds with significant variation from event to event. Despite the increased channel count and the increase in the readout clock period, the observed deadtime per event during the 1999 run was roughly constant at about 1.7 microseconds, resulting in a factor of 1.33 more events to tape than would otherwise have been possible (assuming no changes to the Run I front-end). The deadtime was roughly fixed since the total readout time was determined by the ADC readout for greater than 95% of events. The ADC readout never varied and equaled 1.7 microseconds. Further significant reduction in the observed deadtime required either a reduction in the clock period (a 50 ns clock period would have reduced the deadtime by 280 ns) or faster ADC chips.

C. Increasing Data Capacity

Encoding hits saved data space since the average hit density throughout the latching system was less than 1 hit per 100 channels. Each active (i.e. non-zero) bit was encoded into a 16-bit data word with 5 bits used for the channel number, 5 bits for the slot number, 4 bits for the crate number, and 2 bits unused (see figure 3). The Compressor Module (CM) was designed to reduce the event size by compressing consecutive data words with an arithmetic difference less than 192 into 8-bit offsets with respect to the previous data word or offset (offsets between 192 and 255 were dis-allowed in order to differentiate compressed data words from standard data words). There was no limit, other than crate boundaries, to the number of consecutive compressed hits. Compression reduced the encoded data size by 40%. The overall event compression was 30% including header words, ADC data, and other non-sparse data (which were not compressible). Compression was pursued in anticipation of higher trigger rates and as a hedge with regard to the final data-to-tape rate; however, for the trigger and data-to-tape rates observed during the run, compression of the raw data would not have appreciably increased the total number of events collected. Consequently, to avoid unnecessary complexity, the compression feature of the CM was not used during the bulk of HyperCP data collection. A version of this compression algorithm was implemented in the off-line analysis code repacking raw data for storage on data summary tapes.

A second feature of the compressor module was event aggregation. The compressor aggregated events into "super-events." The number of events per super-event was set to ten although a switch on the compressor permitted any number between 1 and 127. Event aggregation allowed for more efficient use of VME-back-plane bandwidth during event building (large data transfers are more efficient than small ones). The compressor boards were capable of operating in four modes: no compression and no aggregation, compression without aggregation, aggregation without compression, or compression with aggregation.

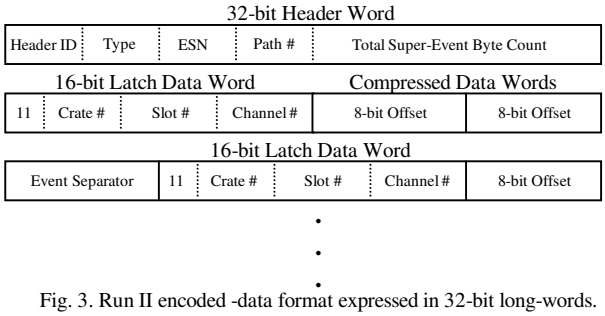

Fig. 3. Run II encoded -data format expressed in 32-bit long-words.

D. Data Format

As previously mentioned, data were split between five parallel paths, thus each event consisted of five separate event fragments. Aggregation was conducted path-by-path resulting in super-event fragments. The super-event fragment data format is shown in figure 3. Each fragment begins with a 32-bit header word identified as a header by the first 4 bits. The header contains a 4-bit flag indicating the data type (Run I versus Run II, compression or no compression, FEM or Nevis), the 4-bit Event Synchronization Number (ESN) for the fragment, the data

path, and a 16-bit byte count for the fragment. The first hit was always a standard 16-bit data word. Additional hits could be either 16-bit data words or 8-bit offsets with respect to the previous hit. Events were separated within the fragment using a unique 8-bit event separator (equal to zero). Compressed versus uncompressed data words are identified by their first two bits. If both bits are set the word is uncompressed, otherwise the word is an 8-bit offset. Legal offsets are thus any number from 1 to 191.

E. Event-Buffer Upgrade

In order to allow for a larger event-buffer, a set of 5 auxiliary VDAS Crates [7] were re-commissioned from other experiments and installed in the DAQ area. Each of the new crates was paired with one of the existing crates to produce an increase in memory capability. The VDAS I/O module in the existing crate required modification. A daughter board was produced that plugged into existing IC sockets. With the daughter board installed, the revised I/O board linked the two crates effectively into a single unit. The presence of data within the upstream crate above a minimum threshold triggered the downstream VDAS to siphon data. This allowed for a seamless upgrade. Additional memory modules were required to fill the new crates. Older VDAS memory modules were found and upgraded to 16 MB per board by replacing the old 256 KB simms with 1 MB simms. As with the compression feature discussed above, the additional memory was not required during Run II and thus the event-buffer upgrade was never fully implemented.

VI. VME Hardware Upgrade

Event building and tape writing were conducted within a VME based system [4,6] using MVME 167 processors. Data were read from the VDAS by the Event Buffer Interface (EBI) and then passed across the VME backplane to the processors. During Run I, assembled events were written to tape using Ciprico RF 3573 SCSI interface modules [8]. The VME processors and Cipricos communicated via the VME backplane; consequently, event building and tape writing competed for use of the backplane. Backplane competition was eliminated during Run II by writing tape directly from the MVME167 processors, thus freeing the VME backplane to be used exclusively for event building. Increased throughput was also achieved by upgrading the EBIs.

A. Fast EBI

The EBI was a custom VME design providing VME access to the VDAS data. Each VDAS output was bused to an EBI in each of the VME crates. The bus protocol was RS-485. Bus access was restricted to one EBI at a time through the use of an electronic token passed from one EBI to the next using front-panel lemo connectors. If in control of the bus and addressed by a CPU for a data transfer, the EBI sent a strobe pulse to the VDAS requesting data. Data on the bus were latched after a fixed delay time. The EBI converted the data to VME protocol and output the data along the VME backplane to be stored by the CPU within the event-building buffer. This cycle continued until the data request was satisfied. The DMA setup time was approximately 10 microseconds; consequently, small data transfers were inefficient. Event aggregation improved overall efficiency by increasing the effective size of the average event fragment (by a factor equal to the aggregation count). Aggregation also decreased the relative variation between super-event fragments, contributing to increased efficiency during multi-crate operations. The fast EBIs increased event-building speed by reducing the time between the VDAS strobe pulse and the latching of the input cable. The upgraded EBI (fast EBI) had a delay length of 192 ns, which was 120 ns less than the delay implemented during Run I. The distance between the VME and VDAS crates was reduced along with the cable length in order to accommodate this increase in speed.

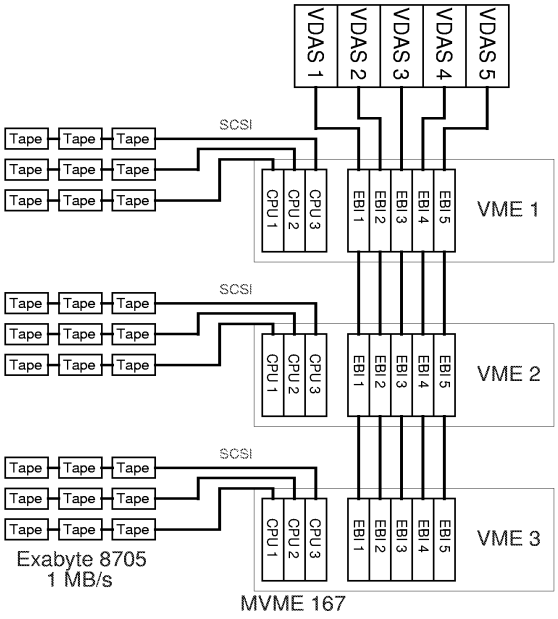

Fig. 4. Run II VME hardware configuration.

B. Direct Tape Writing

Direct SCSI tape writing from an MVME167 processor required the installation of MVME712 communication modules mounted along the VME backplane (one for each CPU). The SCSI port on the MVME712 was used to connect each CPU to three Exabyte 8705 tape drives.

The pre-existing Exabyte 8505 tape drives were obsolescent, worn, and too slow for our needs. We replaced them with new 8705 tape drives rated at 1 MB/s each. A bank of 27 drives was used during the 1999 run, for a

maximum tape rate of 27 MB/s. The Run II VME hardware configuration is shown in figure 4.

VII. VME Software upgrade

A. VxWorks

The proposed direct SCSI tape writing necessitated upgrading the MVME167 operating system to VxWorks version 5.3.1. Unfortunately, this version of VxWorks was not supported for SGI host computers; however, the HyperCP DAQ software was based on the Fermilab DART [9] standard, supported only for an SGI host. The VxWorks 5.3.1 kernel was built on a SUN workstation and copied to a directory on the SGI host computer. Fortunately, VxWorks version 5.3.1 was backward compatible such that the DART and HyperCP DAQ software did not require porting to the SUN.

B. Software Architecture

Direct comparisons between the Run I and Run II software are difficult since the event building, tape writing, and error handling routines were all completely re-written for Run II. The general architecture; however, remained the same. The Data Handler routines accessed the raw data through the EBI. Event fragments from each data path were assembled into complete events within a buffer and then written to tape. The overall process was managed by the DART Run Control (DRC) program running on the host SGI computer.

C. Event Building and Tape Writing

The event-building software structure was designed around modular VME crates, allowing the system to work with from one to five crates. Figure 5 displays the processes running on the VME CPUs. The Data Handler and Tape Spinner processes were run on each CPU, with one Tape Spinner process for each tape drive. The Data Handlers and Tape Spinners were loaded and started during the CPU boot sequence, along with various DART objects used to facilitate communication with the run control process running on the host SGI computer. The Listener processes, one per Data-Handler, were spawned during execution of the DRC Start command. The Listener processes managed event building within the crate and facilitated communication with the DRC program. Communication between the Listener and Data Handler was conducted through a direct socket connection, while the Listener communicated with DRC via a socket connection managed by DART processes running on CPU 1. The Data Handler and Tape Spinners communicated through global variables. The Sampler process provided packets of raw data to the DART Hoist program for use by online monitoring programs.

The Data Handler sent a strobe to the EBI causing the EBI to transfer data from the VDAS to the VME backplane. The header of the super-event fragment was read first. The fragment size and ESN number were extracted and a DMA transfer was initiated. The super-event fragment was written directly into one of three circular buffers. The process then repeated for the second, third, fourth, and fifth EBI in the crate. The assembled super-event was now complete with the data from all paths brought together. The ESNs from all path headers were compared to verify that the individual event fragments originated from a common event. Event building proceeded as long as all VDAS buffers were non-empty (i.e. at least 47 kilobytes of data were stored within each buffer).

Each circular buffer consisted of twenty-five 64-kilobyte blocks. The block size was stipulated by offline software, while the number of blocks per ring was set by the available memory on each CPU module. A block was tagged as full if, at the end of a complete super-event it contained more than 52 kilobytes. Should an unusually large super-event attempt to exceed the 64-kilobyte block size, the block was discarded, along with the current super-event. Events were never written across block boundaries. Blocks were filled sequentially; however, the Data-Handler would switch among circular buffers depending upon which buffer had the greatest number of free blocks.

Each circular buffer had an associated Tape Spinner process which ran in parallel with the Data Handler. Each Tape Spinner worked sequentially around its circular buffer, writing data to a single tape drive. If the current block was full, the Tape Spinner wrote that block to tape and marked the block as empty. It then proceeded to the next block. If the current block was not full, the Tape Writer waited indefinitely until the block was full. If a tape drive or Tape Spinner failed, the associated buffer simply filled, and was thereafter ignored.

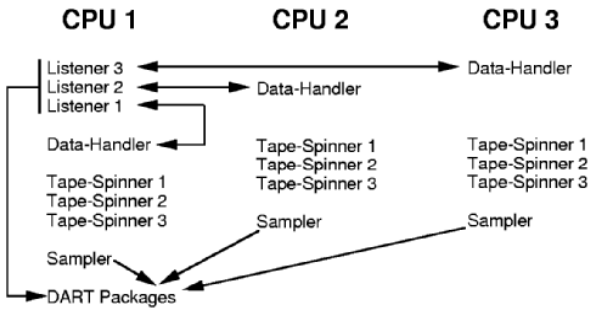

Fig. 5. Software architecture within a single VME crate.

Backplane access was restricted to a single process at a time. A Data Handler with available buffer space requested backplane access for event building via the Listener. If available, the Listener took the VxWorks semophore, which acted as the event-building key. The semophore was granted on a first-come, first served basis. The Listener and Data Handler sat idle until the key (semophore) was taken. After taking the key, the Listener notified the Data Handler which proceeded with event building. The key was released by the

Listener following a message from the Data Handler that all buffers were full.

Complex error correction code was deliberately excluded in order to maintain simplicity and speed. The Data Handlers checked only for ESN sequence errors and header-word format errors. The online monitoring programs were responsible for tagging all other forms of data corruption. The Data Handler notified the Listener concerning ESN or header-word errors. The Listener notified DRC, which then initiated an automatic cleanup procedure. After clearing all buffers and initializing the appropriate hardware, event processing resumed.

Tape Spinners maintained a counter of the number of bytes written to tape, setting a global variable should the counter exceed 4.3 gigabytes. During event processing, the Data Handler checked the value of this variable, reporting to the Listener when a tape was full. The Listener in turn reported to DRC, which initiated a DRC Stop command.

Much of the software was automated. The Tape Spinner and Sampler processes were restarted (if necessary) as a part of the DRC Init command. Most data errors trapped by the online monitoring programs were cleared by executing the DRC Pause command followed by the DRC Resume command. CAMAC function calls tied to DRC commands were used to hold triggers, reset critical hardware, and release triggers. Buffers were automatically cleared at every DRC Initialize or Resume. The software was designed to require a CPU reboot only if a Data Handler crashed, or if the DRC crashed.

VIII. Overall Upgrade Performance

The most probable rate of cascade triggers was 50 kHz during positive mode, and 60 kHz during negative mode. The most probable total trigger rate, including all calibration and secondary physics triggers, was 90 kHz during positive running and 95 kHz for negative running. The typical event size was slightly smaller than in 1997 (500 bytes versus 600 bytes) due to improved gate timing. The typical instantaneous data rate generated by the HyperCP spectrometer was 36 MB/s. The observed live-time distribution is shown in Figure 6. The long tail toward small livetimes was attributed to system errors requiring initialization of the electronics (e.g. ESN sequence error or hardware errors). The sustained data-to-tape rate was 27 MB/s and was limited only by the number of tape drives employed. Between August 1999 and January 2000, HyperCP wrote 71 tera-bytes of data to 18,838 8-mm tapes. We estimate that these data contain over 2 billion fully reconstructable cascade-hyperon decays, ten million omega-hyperon decays, and several hundred million charged K-meson to three charged π-meson decays. The limiting factor for these data was deadtime since additional data-to-tape headroom remained at the chosen operating conditions. Assuming no deadtime, the number of events written to tape would have increased by 25%. No further increase was possible as increased beam intensity would have reduced the yield of useful events.

IX. Acknowledgment

We thank the Fermilab Computer Division and Physics Division for their help and advice. We appreciate the support provided by the Beams Division operators. We also thank A. Baumbaugh, T. Carter, S. Hansen, C. James, K. Knickerbocker, J. Meadows, D. Northacker, and M. Vittone for their hard work and assistance. We also thank A. Chakravorty, C. Durendet, D. Rajaram, and S. White for their assistance. This work was supported in part by the U.S. Department of Energy and the National Science Council of the Republic of China.

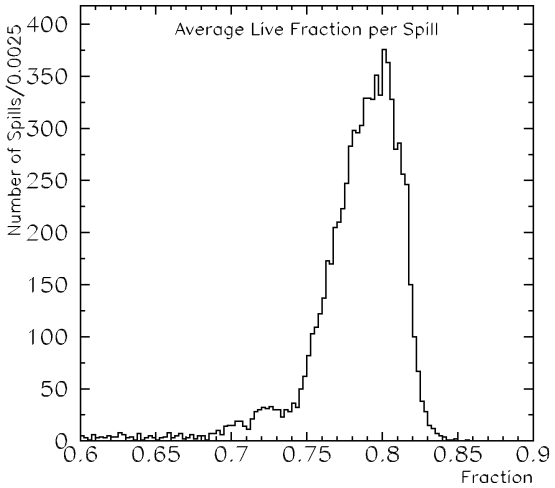

Fig. 6. The live fraction is shown for all Run II spills with an average raw trigger rate between 95 and 105 kHz.

X. References

[1] S. Pakvasa,"CP violation in hyperon decays," hep-ph/0002210.

[2] J. Antos *et al*., Fermilab Proposal 871 (revised), March 26, (1994) see also C. White *et al*., "Search for Direct CP Violation in Lambda and Xi Hyperon Decays", *Nucl. Phys. Proc. Suppl.* **71**:451-456,(1999).

[3] K. B. Luk *et al.*, submitted to *Phys. Rev. Lett.,* (2000), also P. D. Barnes *et al*., *Phys. Rev. C* **54**, 1877 (1996).

[4] Y. C. Chen *et al*., "High Speed DAQ System for the HyperCP Experiment", accepted for publication in *Nucl. Instrum. Meth*., (2000).

[5] Original design ca. 1970 by F. W. Sippach, Columbia University Nevis Laboratories, P.O. Box 137, Irvington, NY 10533, updated for fast readout by F.W. Sippach and H. Cunitz, Columbia University Nevis Laboratories, in 1981. The CR design was updated from ECL II to ECL 10,000 by J. A. Crittenden and D. M. Kaplan in 1984.

[6] C. G. White "Upgraded DAQ system for the HyperCP Experiment",to be submitted to *Nucl. Instrum. Meth*.,(2000).

[7] A. E. Baumbaugh *et al.*, *IEEE Trans. Nucl. Sci*. **NS-33**, 903 (1985); K. L. Knickerbocker *et al., IEEE Trans. Nucl. Sci*. **NS-34**, 245 (1986).

[8] Ciprico Inc., 1161 Homestead Ln., Chanhassen, MN 55317

[9] R. Pordes *et al*., "Fermilab's DART DA System," presented at Computing in High Energy Physics (CHEP 94), San Francisco, CA, 21-27 Apr. 1994, FERMILAB-CONF-94-103.