




8-1990

A High Performance Detector Electronics System for Positron Tomography

John Clifton Moyers, Jr.
University of Tennessee - Knoxville

Follow this and additional works at: https://trace.tennessee.edu/utk_gradthes

 Part of the [Electrical and Electronics Commons](#)

Recommended Citation

Moyers, Jr., John Clifton, "A High Performance Detector Electronics System for Positron Tomography. "
Master's Thesis, University of Tennessee, 1990.
https://trace.tennessee.edu/utk_gradthes/13

This Thesis is brought to you for free and open access by the Graduate School at TRACE: Tennessee Research and Creative Exchange. It has been accepted for inclusion in Masters Theses by an authorized administrator of TRACE: Tennessee Research and Creative Exchange. For more information, please contact trace@utk.edu.

To the Graduate Council:

I am submitting herewith a thesis written by John Clifton Moyers, Jr. entitled "A High Performance Detector Electronics System for Positron Tomography." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

Robert E. Bodenheimer, Major Professor

We have read this thesis and recommend its acceptance:

James M. Rochelle, Donald W. Bouldin

Accepted for the Council:


Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

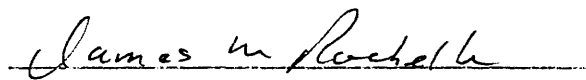
(Original signatures are on file with official student records.)

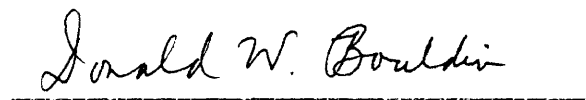
To the Graduate Council:

I am submitting herewith a thesis written by John Clifton Moyers, Jr. entitled "A High Performance Detector Electronics System for Positron Emission Tomography". I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree Master of Science, with a major in Electrical Engineering.



Robert E. Bodenheimer, Major Professor

We have read this thesis
and recommend its acceptance:





Accepted for the Council:


Vice Provost
and Dean of the Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at The University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of the source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in his absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature JCMorgan
Date AUGUST, 1990

A HIGH PERFORMANCE DETECTOR ELECTRONICS SYSTEM FOR POSITRON EMISSION TOMOGRAPHY

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

John Clifton Moyers, Jr.

May 1990

ACKNOWLEDGMENTS

The development of this new Detector Electronics System involved many participants. The one individual that conceived the idea of this system was Mike Casey. I am thankful for Mike's encouragement and forethought that went into the concepts of this new bucket. Having developed the previous generation bucket electronics, Mike was sensitive to the areas that needed special attention on this design. Through his guidance, many "learning curve" mistakes were avoided.

Many of the initial architecture and data flow ideas originated from Mike. The Pileup Rejector concept and theory originated from a paper that Mike authored while taking a class that was taught by Dr. T. V. Blalock. I am also thankful to Mike for encouraging the completion of this document and degree.

Other major contributions to this design include those by David Binkley and John Young. David designed the Analog Processor section with unequaled precision and thoroughness. John designed the firmware for the bucket. His highly motivated work ethic and tolerance of fuzzy (at times) hardware information made the development process a real pleasure.

A special thanks is also in order to Bill Jones for his persistence in stimulating the thoughts that led to the development of the Pseudo Activity generation circuitry. It has now become a frequently used tool for scanner verification and debugging.

Finally, I am particularly indebted to my wife, Dana, for her patience and attention during the writing of this thesis. Her technical proofreading skills have been an invaluable asset throughout the preparation of this document.

ABSTRACT

Positron Emission Tomography (PET) has been a very useful laboratory tool for the noninvasive study of dynamic physiological processes within the human body. Its clinical worth in cardiology, neurology, and oncology has been verified for well over a decade. Only with the most recent introduction of high performance analog and VLSI digital components yielding higher scanner resolutions at reduced costs, has clinical PET truly become a reality. The High Performance Detector Electronics System presented here represents indubitably the most advanced processing system available in the clinical PET market.

PREFACE

The continuing evolution of electronic component functionality and packaging density tends to make electronics-related products obsolete at a relatively quick pace. This accelerated obsolescence should not be viewed negatively, as each new generation that is produced by this process represents a refinement in functionality and performance. If an organization does not take advantage of the progress being made in the technology, then the competitiveness of their product(s) will begin to decline. This mechanism of change has driven the electronics industry since its inception, and is driving the PET industry today.

The ability to produce the detector electronics system described in this document was made possible through the introduction of new high density, high performance analog and digital electronic components. New electronic components that will appear within the next few months or years will tend to obsolete this design, forcing yet another iteration.

In an effort to represent and support the philosophy of this new design, the complete bucket electronics system will be represented in its entirety. Although the author did not directly design the Analog Processor or the firmware for this system, an in-depth discussion of both of these topics is presented herein for completeness.

TABLE OF CONTENTS

CHAPTER I

INTRODUCTION	i
Positron Emission Tomography (PET)	1
A Brief History of PET	2
A Modern PET Scanner	5
Scope of Thesis	5

CHAPTER II

THE EVOLUTION OF PET DETECTOR ELECTRONICS	9
The Need for the Multi-Element Detector	9
Operation of the Multi-Element Detector	10
An Early Multi-Element Detector Electronics System	15
Auto-Setup Multiplexed Electronics System	17

CHAPTER III

BUCKET ARCHITECTURE	19
Acquisition System Operation Summary	19
Bucket Data Flow and Block Diagrams	23
System Diagnostics	25
Firmware Requirements	29

CHAPTER IV

BUCKET ELECTRONIC COMPONENTS	35
Analog Processor	35
Analog Processor Concepts	36
Variable Gain Preamplifier/Baseline Restore Circuitry	36
Constant Fraction Discriminator	39
Axial/Transaxial/Energy Integrators	41
Ratiometric Converters	42
DC Control Circuitry	44
Position/Energy Processor Board	44
Detector Interface	46
Analog Processor Interface	48
Clock Generation Circuitry	48
Time Digitizer	49
Positioning/Energy Discrimination	54
Pileup Rejector	58
Time/Position Latches and Request Logic	61
Diagnostics/Histogram Mode Buffers	65
Event Sequencer	65
Run Mode	70
Position Histogram Mode	72
Energy Histogram Mode	73
Microcomputer Interface	74
Processor Interface	74
Pseudo Activity Generator	77

Bucket Controller Board	79
High Speed Data Path	79
Common Electronics Sequencer and Arbiter	83
Interrupt Throttling Circuitry	88
Time Correction Circuitry	88
Time and Position Capture Latches/High Speed Shift Registers	89
Event Counting Circuitry	90
Microcomputer	94
Microcomputer Core	97
Analog Input Channels	100
Microcomputer Peripherals	103
CHAPTER V	
CONCLUSIONS	105
Performance of the Bucket	105
Future Improvements to the Bucket	107
REFERENCES	108
APPENDICES	110
APPENDIX A	
SUMMARY OF BUCKET COMMANDS	111
APPENDIX B	
DESCRIPTION OF FUNCTIONALITY OF EXTENSIVE DIAGNOSTICS	
ROUTINES TO BE USED DURING BUCKET TESTING	116

APPENDIX C

PAL SOURCE LISTINGS OF BUCKET	143
VITA	178

LIST OF TABLES

Table 3.1. List of Position/Energy Processor Diagnostics	30
Table 3.2. List of Bucket Controller Diagnostics	31
Table 4.1. Position/Energy Processor Microcomputer Interface Memory Map.	76
Table 4.2. Bucket Controller Microcomputer Memory Map (Physical)	99

LIST OF FIGURES

Figure 1.1. A typical gamma camera detector arrangement.	3
Figure 1.2. A typical PET camera detector "ring" arrangement.	6
Figure 1.3. A modern PET scanner installation layout.	7
Figure 2.1. Multi-element detector block.	11
Figure 2.2. Ratioing electronics used for determination of X and Y position of an interaction.	11
Figure 2.3. One dimensional probability distribution of an 8-crystal row.	13
Figure 2.4. Probability distribution illustrating binning with infinitely thin bin borders.	14
Figure 2.5. Probability distribution illustrating binning with areas of rejection.. . . .	14
Figure 2.6. Diagram of an early position decoder electronics.	16
Figure 2.7. A modern 8 x 8 multi-element detector block.	18
Figure 3.1. Diagram of PET Scanner, highlighting system data flow during acquisition of a sinogram.	20
Figure 3.2. Bucket Controller time and position shift registers and associated data. .	21
Figure 3.3. Illustration of bucket data flow.	24
Figure 3.4. Block diagram of Analog Processor section.	26
Figure 3.5. Block diagram of Position/Energy Processor section.	27
Figure 3.6. Block diagram of Bucket Controller.	28
Figure 3.7. Topographic representation of a position histogram of an 8 x 8 block. . .	33
Figure 4.1. Schematic diagram of Variable Gain Preamplifier and Baseline Restoration circuitry.	37
Figure 4.2. Constant Fraction Discriminator schematic diagram.	40
Figure 4.3. Integrator and Ratiometric Converter circuitry.	43
Figure 4.4. DC Control circuitry.	45

Figure 4.5. Detector Interface schematic diagram.	47
Figure 4.6. Schematic diagram of Clock Generation Circuitry.	50
Figure 4.7. Timing diagram of signals generated by Clock Generation Circuitry. . . .	51
Figure 4.8. Schematic diagram of Time Digitizer.	52
Figure 4.9. Timing waveforms within Time Digitizer.	53
Figure 4.10. Schematic diagram of Position/Energy Processor section.	56
Figure 4.11. Schematic diagram of Time/Position section of Position/Energy Processor. .	57
Figure 4.12. Block diagram of the Pileup Rejector.	62
Figure 4.13. Event Sequencer schematic diagram.	66
Figure 4.14. Timing waveforms for Event Sequencer in Run Mode.	67
Figure 4.15. Timing waveforms for Event Sequencer in Position Histogram Mode. . .	68
Figure 4.16. Timing waveforms for Event Sequencer in Energy Histogram Mode. . .	69
Figure 4.17. Microcomputer Interface schematic.	75
Figure 4.18. Timing diagram of CTR_CLK Generator.	78
Figure 4.19. Schematic diagram of the four Position/Energy Processor sections. . . .	80
Figure 4.20. Top level schematic diagram of Bucket Controller.	81
Figure 4.21. Bucket Controller time/position data path electronics schematic.	82
Figure 4.22. Position Swap PAL functionality.	84
Figure 4.23. The Common Electronics Sequencer schematic.	85
Figure 4.24. Timing diagram of Common Electronics Sequencer.	86
Figure 4.25. Corrected singles counter block diagram.	92
Figure 4.26. Schematic diagram of event counting circuitry.	93
Figure 4.27. Schematic diagram of microcomputer electronics.	95
Figure 4.28. Schematic diagram of microcomputer electronics and peripherals. . . .	96
Figure 4.29. High voltage sampling electronics.	102

CHAPTER I

INTRODUCTION

Positron Emission Tomography (PET)

Positron Emission Tomography (PET) has gained significant popularity in nuclear medicine due to the ability to noninvasively study physiological processes within the human body [1]. By employing compounds such as ^{11}C labeled glucose, ^{18}F labeled glucose, ^{13}N labeled ammonia and ^{15}O labeled water, PET can be used to study such physiological phenomena as blood flow, tissue viability, and *in vivo* brain neuron activity, to name a few. These neutron deficient compounds interact with free electrons in the body area of interest resulting in the annihilation of the positron [2]. The resulting annihilation yields the emission of a pair of photons (gamma rays) approximately 180° apart. A compound having the desired physiological affect is administered to the patient and the radiation resulting from annihilation is detected by a PET camera. After acquiring these annihilation "event pairs" for a period of time, the isotope distribution in a cross section of the body can be reconstructed.

PET data acquisition occurs by detection of both photons emitted from the annihilation of the positron in a coincidence scheme. Due to the approximate 180° angle of departure from the annihilation site, the location of the two detectors registering the "event" lie on a line passing through the location of the annihilation. By histogramming these lines of response, a "sinogram" is produced that may be used by a process called backprojection to produce a two dimensional image of the activity concentration within the field of view (FOV).

In order to detect these lines of activity, a coincidence detection scheme is employed. A valid event line is registered if both photons are detected within a "coincidence window" of

time. Coincidence detection methods ensure that an event line is histogrammed only if both photons originate from the same positron annihilation. This coincidence measurement criteria places stringent requirements on the detector timing resolution.

A Brief History of PET

Diagnostic nuclear medicine began more than forty years ago with the use of radioiodine for the diagnosis and investigation of thyroid disease. By administration of this compound, and then monitoring of its concentration, the kinetics of thyroid iodine uptake can be derived.

Monitoring of two dimensional isotope concentration, such as radioiodine, was first performed by the scintillation counter proposed by Cassen [3]. The counter involved a photographic plate, moved rectilinearly, that represented count density by photographic intensity. Its limits were due to extremely long acquisition time.

The gamma camera (Figure 1.1), introduced by Anger in 1957 [4], became the first practical detection system in nuclear medicine. Photon detection is performed by a large sodium iodide crystal, with the scintillation location being determined by a large array of photomultiplier tubes placed along both dimensions of the crystal. A large lead collimator, placed in front of the crystal, serves to collimate the photons, thus allowing imaging of photon concentrations. Shortcomings of gamma cameras include the losses due to the large collimators. Sensitivities of gamma cameras are such that relatively large patient doses are required to overcome the collimator losses.

Radioiodine, for imaging of the thyroid, is a unique isotope in that the thyroid has an affinity for iodine and thus will readily uptake the substance. Availability of "radiolabeled" compounds that are used by other organs of the body is becoming increasingly desirable to the nuclear medicine community.

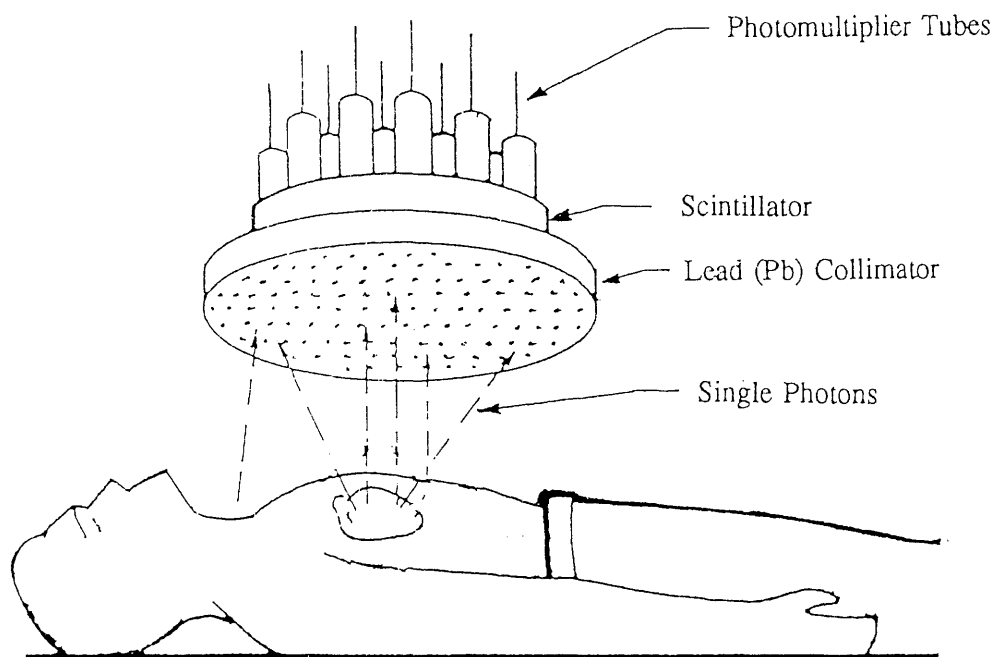


Figure 1.1. A typical gamma camera detector arrangement.

Researcher's interest in positron-emitting isotopes arose from the fact that three of the most basic elements in the human body (carbon, nitrogen, and oxygen) occur as positron emitters. These elements are involved in many of the physiological processes within the body. By replacing these non-radioactive elements with their chemically parallel positron-emitting form, the positron camera can be used to monitor, non-invasively, metabolic pathways without altering the physiology. These positron-emitting equivalents behave within the body in the same manner as their unlabelled counterparts.

Wrenn [5] was one of the first to investigate positron-emitting imaging agents in the early nineteen fifties. Brownell [6] performed some of the first work with positron annihilation coincidence detection in the early fifties. This early work, using a multi-detector system, brought "futuristic concepts into nuclear medicine" [7].

The first transverse section PET camera development is credited to Rankowitz [8], in 1962. Developed at Brookhaven National Laboratory, this scanner consisted of a single ring of 32 sodium iodide crystals. The design allowed each crystal to be in coincidence with a number of opposing crystals on the opposite side of the ring. Data was collected by a 2-dimensional pulse height analyzer, as computers were not available for use in reconstruction. Sensitivity gains due to the scanner's electronic collimation (and thus lack of lossy lead collimator) were lost by the poor stopping ability of the low density sodium iodide detectors. Another attempt at tomographic imaging from projection data was reported by Kuhl [9] in 1963.

Through the nineteen seventies, many single-ring PET camera designs were attempted. One of the first successful multi-slice (axial) PET cameras was developed in 1975 [10]. This scanner consisted of 48 detectors in six banks. An elaborate method of rotating the array of detectors involved stepping the array in 20 three-degree steps, thus filling in the sampling gaps between detectors. The detectors were made of a long cylinder of sodium iodide with one photomultiplier tube mounted at each end of the cylinder. These cylindrical detectors were

arranged with their length axis in parallel with the patient opening axis. Axial positioning was performed by the ratio of light seen by the two photomultiplier tubes. This ratio space was equally divided into four axial planes. By accepting adjacent ring coincidences, three crossplanes of data were also created, producing a total of seven planes of data.

A Modern PET Scanner

Today, PET scanners consist of detector rings (Figure 1.2) made up of two to 16 axial detector pairs that can, by using adjacent crossplanes, present up to 31 axial planes (slices) of the body at one time [11]. These detector arrangements are accompanied by acquisition systems that use large array processors to perform the complex reconstruction tasks. Multi-user, multi-tasking mini-computers are used to perform study archiving and displaying functions.

A modern scanner (Figure 1.3) consists of the acquisition rack and mini-computer located within an environmentally controlled room. This room has an electronics (halon) type fire protection system and is constructed to contain the noise of the equipment. The operator console is located in another room adjoining the room containing the gantry. A window to allow patient viewing separates these two rooms. The gantry is located in a room similar to a typical hospital examination room, with a large entrance door to allow easy passage of patient beds into the room. Another door separates the gantry room from the room containing the operator's console.

Scope of Thesis

The author began the development of the new bucket, with the intention of completing the entire analog, digital, and firmware design. As scheduling pressures mounted, it became apparant that more help was needed in order to meet timely deliveries. Under the project

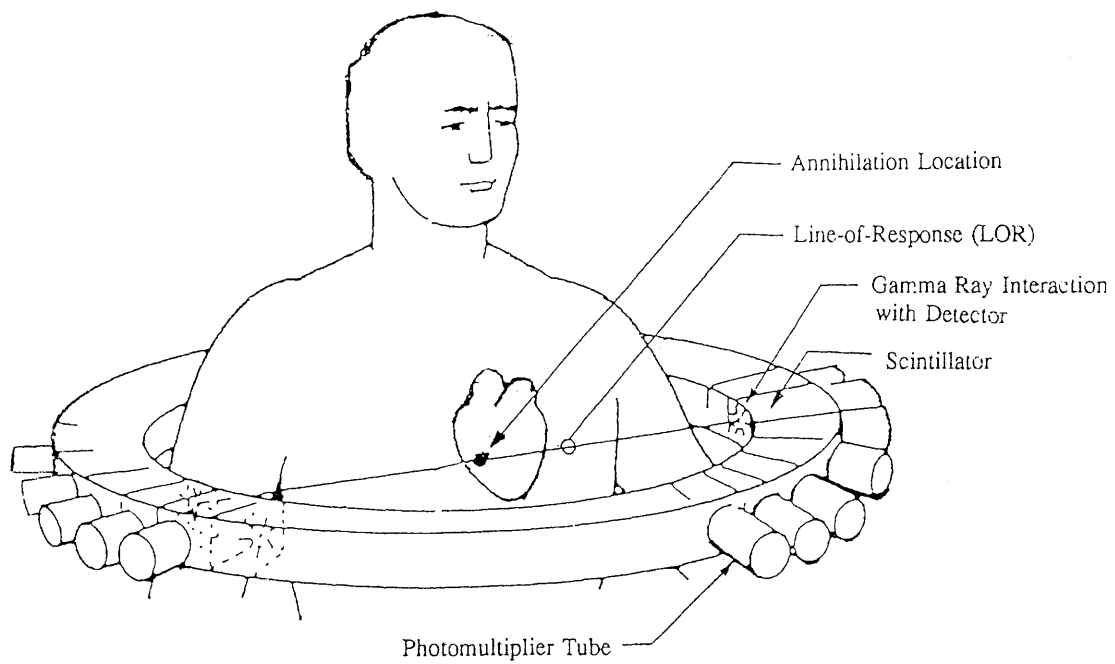


Figure 1.2. A typical PET camera detector "ring" arrangement.

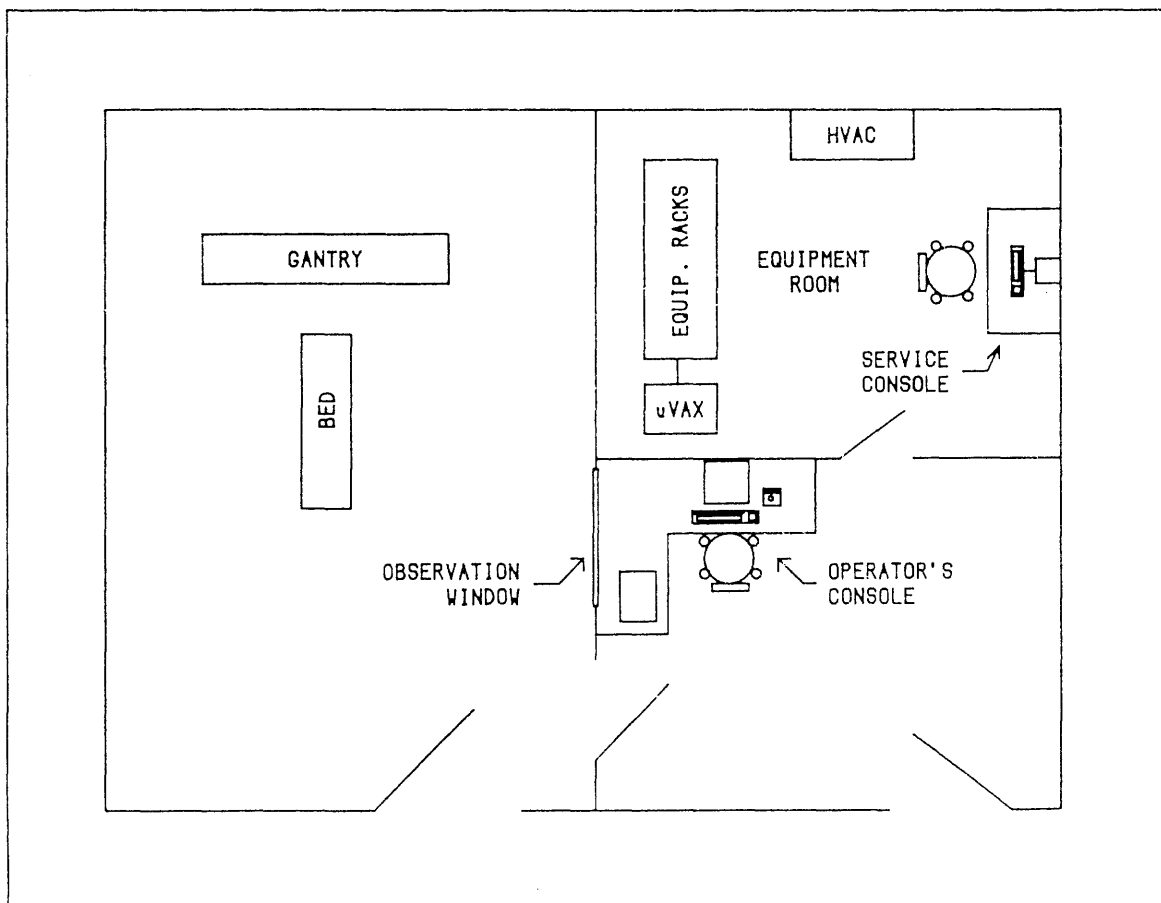


Figure 1.3. A modern PET scanner installation layout.

leadership of the author, two other development engineers were enlisted to develop the firmware and the analog processing electronics. The author completed the development of the digital processing and microcomputer circuitry of the bucket.

The purpose of this thesis is to present the philosophy and design considerations of a state-of-the-art detector electronics system for a modern PET system. Although the author participated in all facets of this design, the Analog Processor electronics and the firmware development were executed by other members of the development team. A thorough discussion of these components is included in this thesis for completeness.

Chapter II covers the evolution of PET electronics leading up to the need for the electronics system discussed within this thesis. This is followed by a chapter covering the architecture of the new electronics system including hardware and data flow diagrams. Chapter IV offers a thorough discussion of each of the elements of the bucket electronics. Finally, Chapter V summarizes the completed product and discusses the direction that future detector electronics systems may progress.

CHAPTER II

THE EVOLUTION OF PET DETECTOR ELECTRONICS

The desire for higher camera sensitivity and smaller image resolution has demanded detectors with both higher efficiencies and smaller crystal dimensions. Competitive market pressures have compelled reductions in PET scanner costs. These conflicting demands have resulted in the development of the multi-element detector "block" concept [12] and the electronics required to support it.

The Need for the Multi-Element Detector

A detector material's efficiency is determined by its ability to stop the high energy particles impinging upon it. Particles that do not get stopped by the detector scintillator pass through the detector undetected. Due to the high energy (511 keV) of the gamma rays resulting from the positron annihilation, a very dense scintillator material is required to stop these particles efficiently. Bismuth Germanate (BGO), with its high z number, has become the most widely used PET detector scintillator material [13].

Scanners produced in the early nineteen seventies were made up of rings of discrete scintillators mounted individually on photomultiplier tubes. The desire to reduce detector size, thus increasing the quantity of detectors, presented a major monetary problem due to the cost of the great numbers of photomultiplier tubes required. Another constraint arose from the fact that the physical size of an acceptable photomultiplier tube was larger than the desired crystal cross section. These limitations of single-photomultiplier-tube-single-scintillator-element detectors led to the concept of the modern multi-element detector block.

Operation of the Multi-Element Detector

The multi-element detector block (Figure 2.1) is made up of one large square scintillator crystal and four photomultiplier tubes mounted on the rear. The crystal is grooved to separate the actual discrete detectors, with the different groove depths producing a light guide effect within the crystal.

The processing electronics used to discriminate the discrete crystals within the block uses a ratioing scheme where the ratio of light seen by two adjacent photomultiplier tubes to the total light seen by all four tubes determines the appropriate detector row (or column) in which the particle impinged. By performing these ratio operations in both the X (transaxial) and Y (axial) directions (Figure 2.2) across the block, specific crystal element identification may be performed. The mathematical representation for the transaxial and axial position ratios are given by:

$$\text{TRANSAXIAL POSITION RATIO} = \frac{TUBE0 + TUBE2}{TUBE0 + TUBE1 + TUBE2 + TUBE3} \quad (1)$$

$$\text{AXIAL POSITION RATIO} = \frac{TUBE0 + TUBE1}{TUBE0 + TUBE1 + TUBE2 + TUBE3} \quad (2)$$

Due to the finite number of light photons produced by an interaction of a 511 keV gamma ray within a detector element, positioning accuracy is statistically governed. Assuming that N photons are produced in the interaction and on average N_{AC} photons are converted by tubes A and C, then N_{BD} photons are converted by tubes B and D. The probability of photons being detected in tubes A and C is given by p , while the probability of photons being detected in tubes B and D is given by q . The probability distribution of the number of photons converted by tubes A and C for an interaction along the X axis (transaxial) is approximated by the binomial

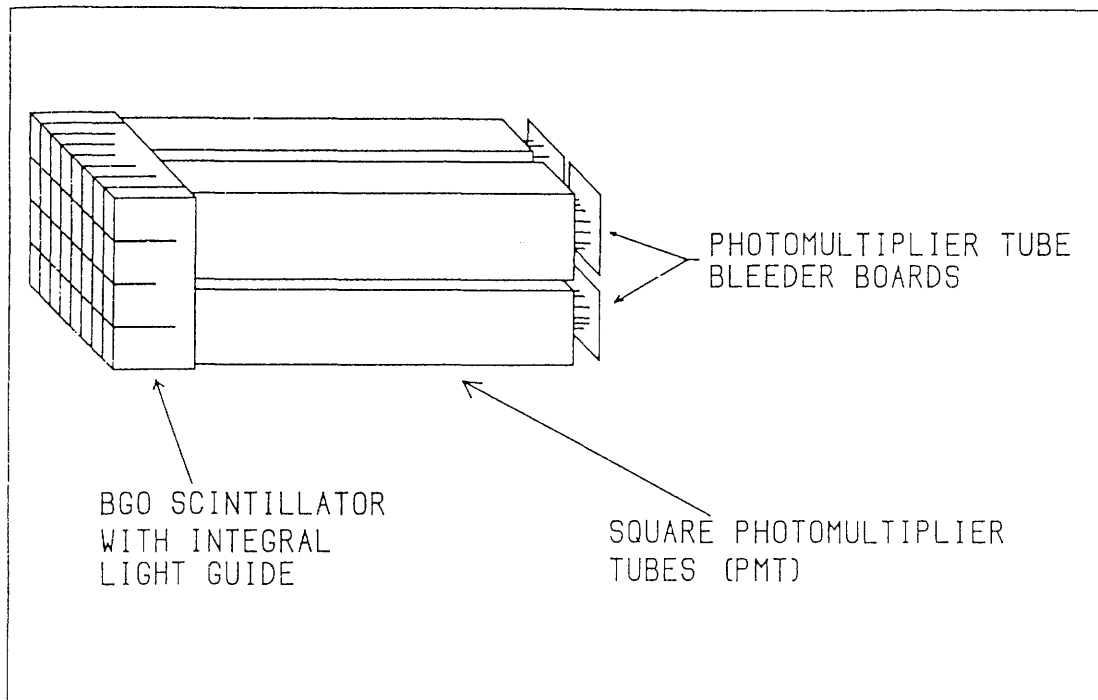


Figure 2.1. Multi-element detector block.

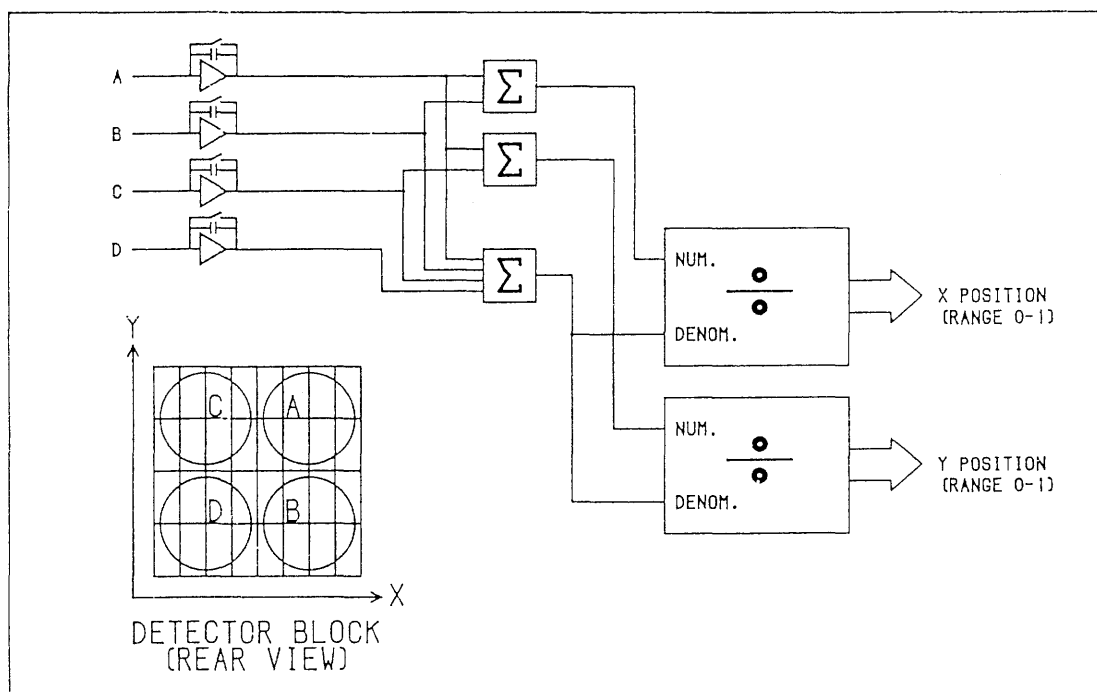


Figure 2.2. Ratioing electronics used for determination of X and Y position of an interaction.

distribution:

$$\Phi_{AB}(n) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{n-\mu}{\sigma}\right)^2} \quad (3)$$

where $\sigma = \sqrt{Npq}$ ($\sqrt{\text{Variance of Distribution}}$)

$$p = \frac{N_{AC}}{N}$$

$$q = \frac{N_{BD}}{N} = \frac{N - N_{AC}}{N} = 1 - p$$

$$\mu = Np = N_{AC}$$

The same distribution holds for positioning in the Y axis (axial). Figure 2.3 shows the probability distribution on the X axis with eight crystal rows. Each peak represents the probability distribution of locating the specific crystal row (or column) within the detector block. The horizontal axis represents the ratio value from the X (transaxial) axis equation, while the vertical axis represents the probability density. The area of overlap between two adjacent peaks represents areas where statistical uncertainty exists in correctly identifying the proper row.

In order to identify the specific crystal within which the interaction took place, discriminator values must be placed around each of the peaks, both in the X axis and the Y axis (Figures 2.4 and 2.5). A simplistic one-dimensional example of this is shown in Figure 2.4. Each discriminator represents a window, or bin, in which an event with a position ratio falling above the lower discriminator setting and below the upper discriminator setting is "binned" into that row. A method shown in Figure 2.4 represents the most efficient binning of events. All events are positioned into a bin even if the position ratio value falls into the region where overlap of

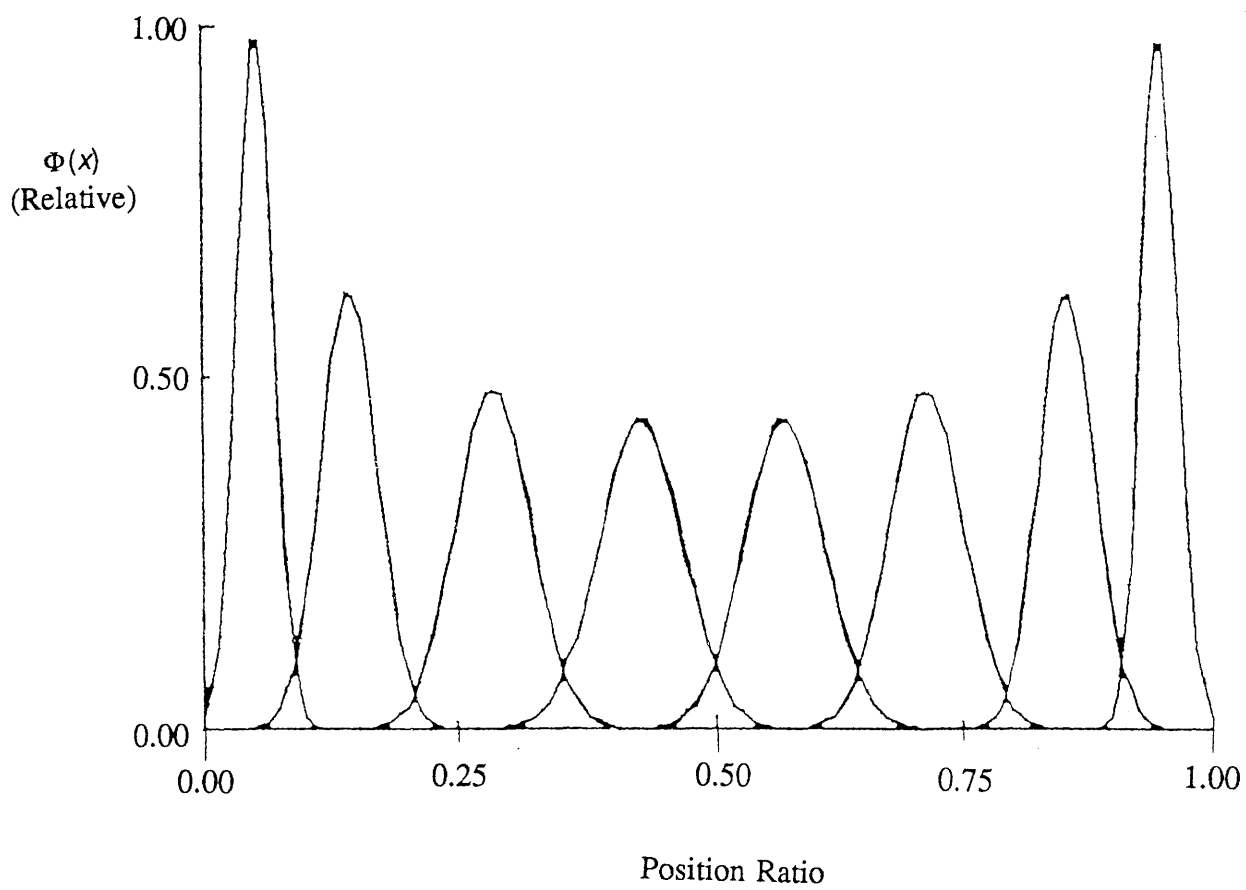


Figure 2.3. One dimensional probability distribution of an 8-crystal row.

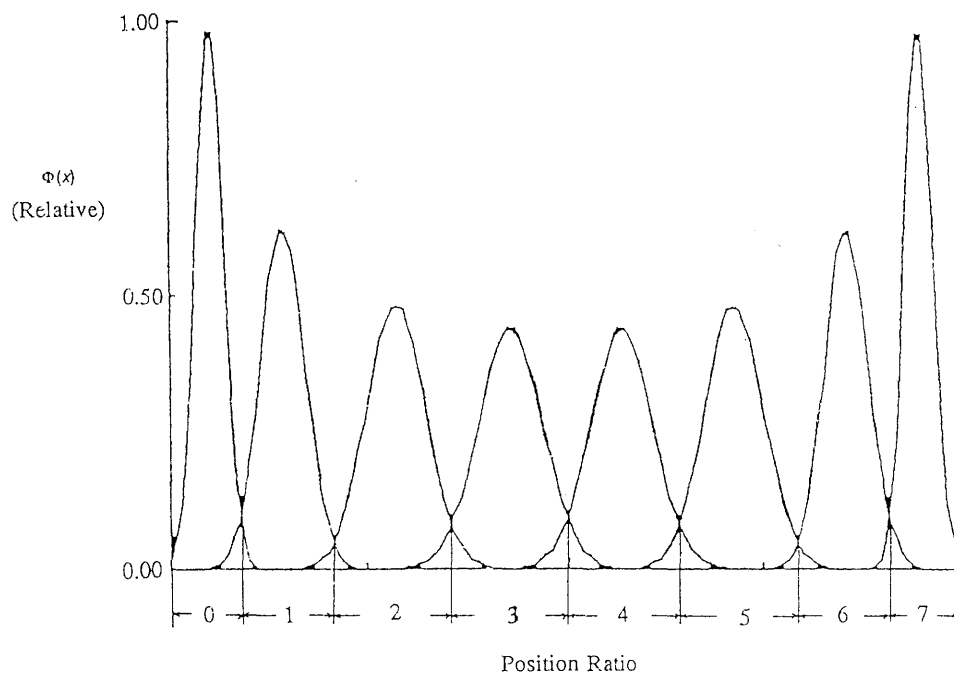


Figure 2.4. Probability distribution illustrating binning with infinitely thin bin borders.

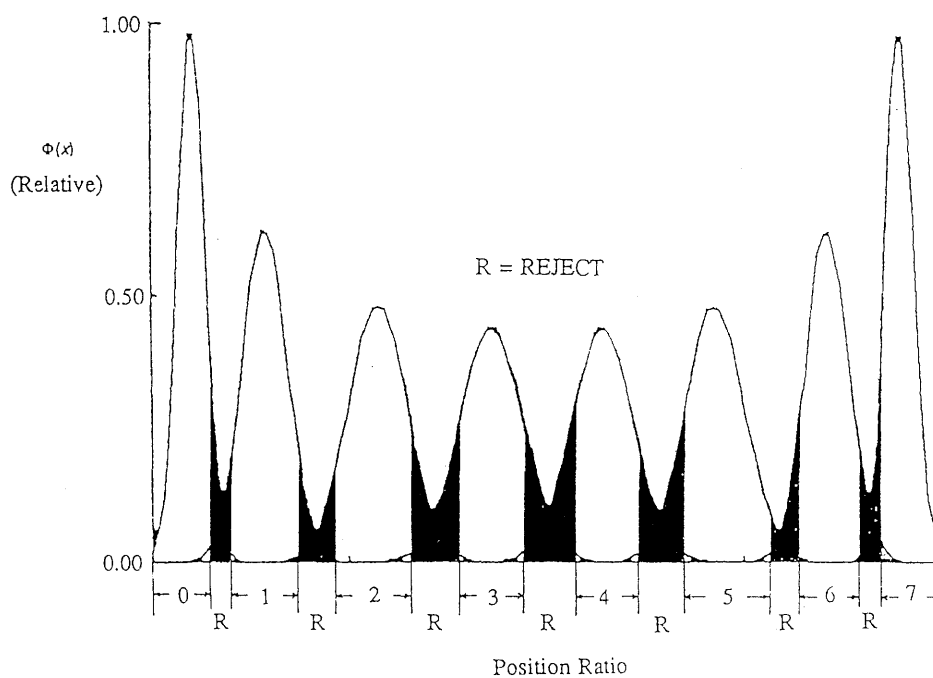


Figure 2.5. Probability distribution illustrating binning with areas of rejection..

two adjacent peaks exists. Another method of binning events (Figure 2.5) is to reject events that fall within the regions of overlap. In doing so, mispositioning error is reduced at the cost of the loss of counts. The ratio of areas under the overlapping regions and between the discriminators over the total area under the eight curves represents the losses due to this positioning scheme.

An Early Multi-Element Detector Electronics System

The first multi-element detector block represented a 32 element device arranged in an eight (transaxial) by four (axial) arrangement. This detector represented a four-fold increase in detector elements to photomultiplier tubes over the multi-plane detector described by Ter-Pogossian [11]. This detector represented a monumental achievement in detector-to-photomultiplier tube performance.

The first electronics system performed the positioning by fixed-position discriminator thresholds as shown in Figure 2.6. This discriminator system had fixed levels that separated adjacent peaks. Each threshold was set by the ratios of the resistors within the divider strings. No "rejection area" (Figure 2.5) was possible with this scheme as every location along the ratio axis was binned.

Performance of the first systems incorporating the multi-element block detectors was compromised by two issues: 1) the "bins" were fixed; and 2) the actual location of the position peaks relied significantly on processing parameters during the physical manufacturing of the block. The major effects of these variations in the processing parameters were the resulting changes in the spatial location of the peaks in the X and Y axes.

Elaborate schemes were devised to normalize the position spectra peaks in an effort to place these peaks within the fixed bins of the electronic system. Each of the four photomultiplier tubes in the block has a potentiometer in parallel with one of the bleeder-string resistors in order

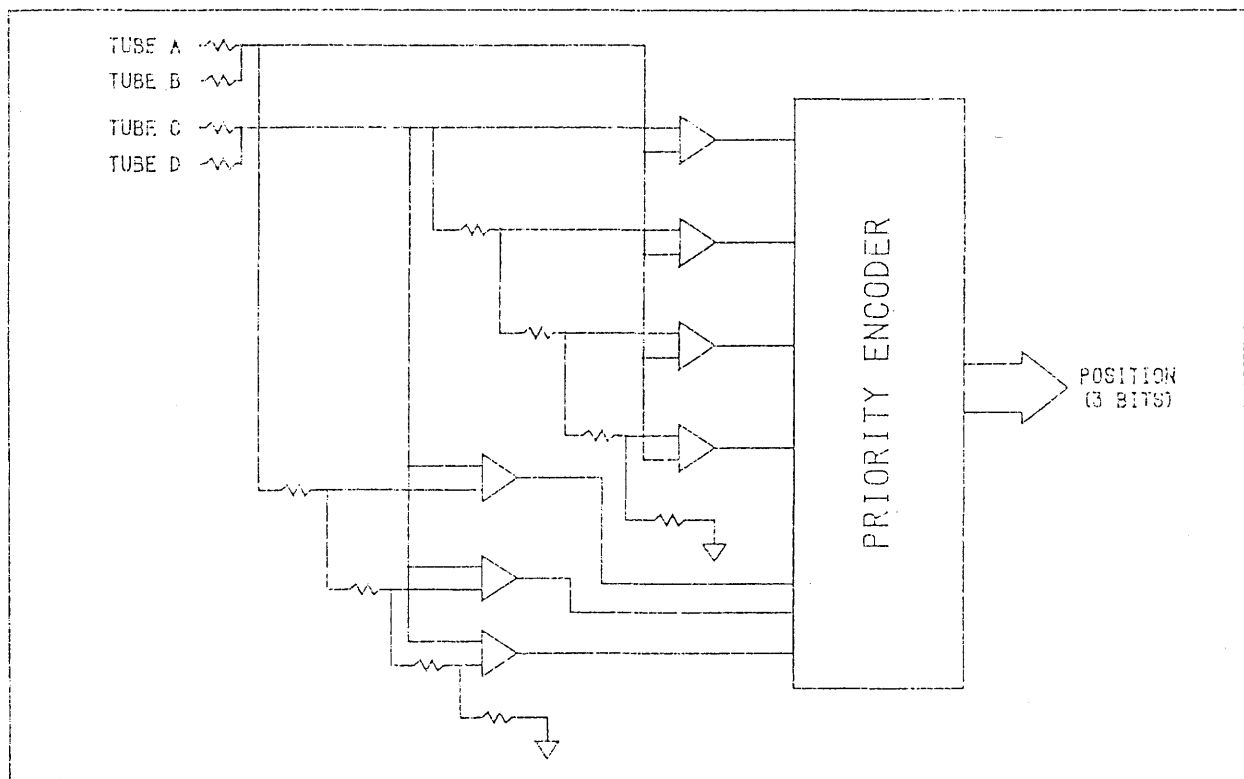


Figure 2.6. Diagram of an early position decoder electronics.

to allow adjustment of the tube's gain. By varying this resistance and thus varying the voltage between two adjacent dynodes, the gain of the tube can be set over an approximately five-to-one range. Adjustment of these four potentiometers on the block permits normalization of the block gain and optimization of the position spectra.

Although the gain potentiometers on each of the four photomultiplier tubes allows adjustment of the position spectra, they do not provide a means to overcome nonlinear position location of the position peaks that are due to processing irregularities. Variability of BGO crystal clarity, reflector reflection coefficient, and quality of photomultiplier tube-to-crystal bonding all affect spatial position linearity. With the first detector electronics system, mispositioning was the result of this nonlinear variability between detectors.

Auto-Setup Multiplexed Electronics System

As competitive market pressure mounted, the need for higher detector-to-photomultiplier tube ratios was required. In response to this, a 64-element detector block was proposed (Figure 2.7). This required equivalent axial positioning performance as in the transaxial positioning accuracy of the older 32 element detectors. The desire to produce this 64-element detector prompted the development of the new bucket electronics.

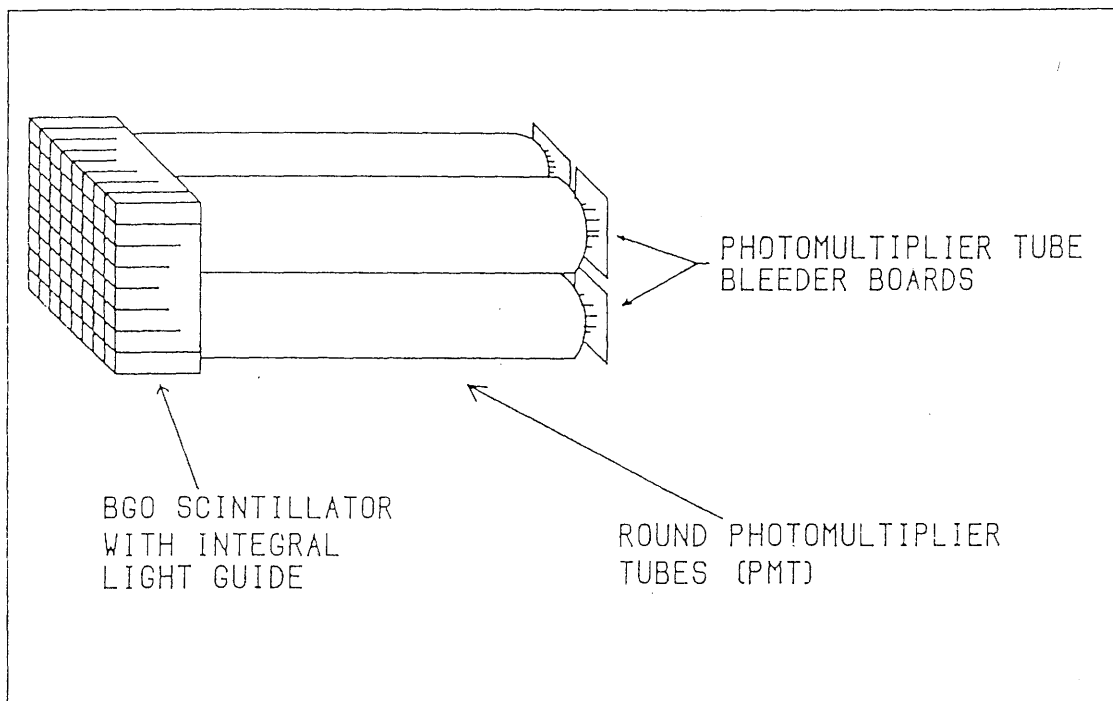


Figure 2.7. A modern 8 x 8 multi-element detector block.

CHAPTER III

BUCKET ARCHITECTURE

Acquisition System Operation Summary

The fundamental data block that contains the information required to reconstruct images in PET is called a Sinogram. This block of data contains the number of counts that each of the possible coincidence lines-of-response (LOR) have obtained during the acquisition. These lines-of-response represent a projection depicting a line integral of activity at a given angle and position. Each LOR has a unique address in sinogram memory. When a coincidence line (-of-response) is detected, the specific location representing that unique line is incremented by one. The number of counts that each LOR memory location contains represents the relative amount of activity (i.e. isotope) along that line. Image reconstruction takes this sinogram of one-dimensional projections and uses the number of counts associated with each LOR to reconstruct a two dimensional image of activity density.

A simplified diagram of the system data flow in construction of the sinogram is shown in Figure 3.1. The detector electronics system (bucket) is capable of supplying an event word to the coincidence processor every 256 ns. This 256 ns sample interval represents the fundamental data-packet period in which data moves throughout the system in a pipeline fashion.

One 16-bit word from the bucket contains the time when the gamma ray interacted with the detector and the crystal location of interaction. Six bits are used to represent the subdivision of the 256 ns sample interval into 64 equally spaced bins of 4 ns resolution. The position information in the bucket data word represents the axial (3 bits) and transaxial (5 bits) location of the interaction as shown in Figure 3.2. Two more bits within the bucket data word denote

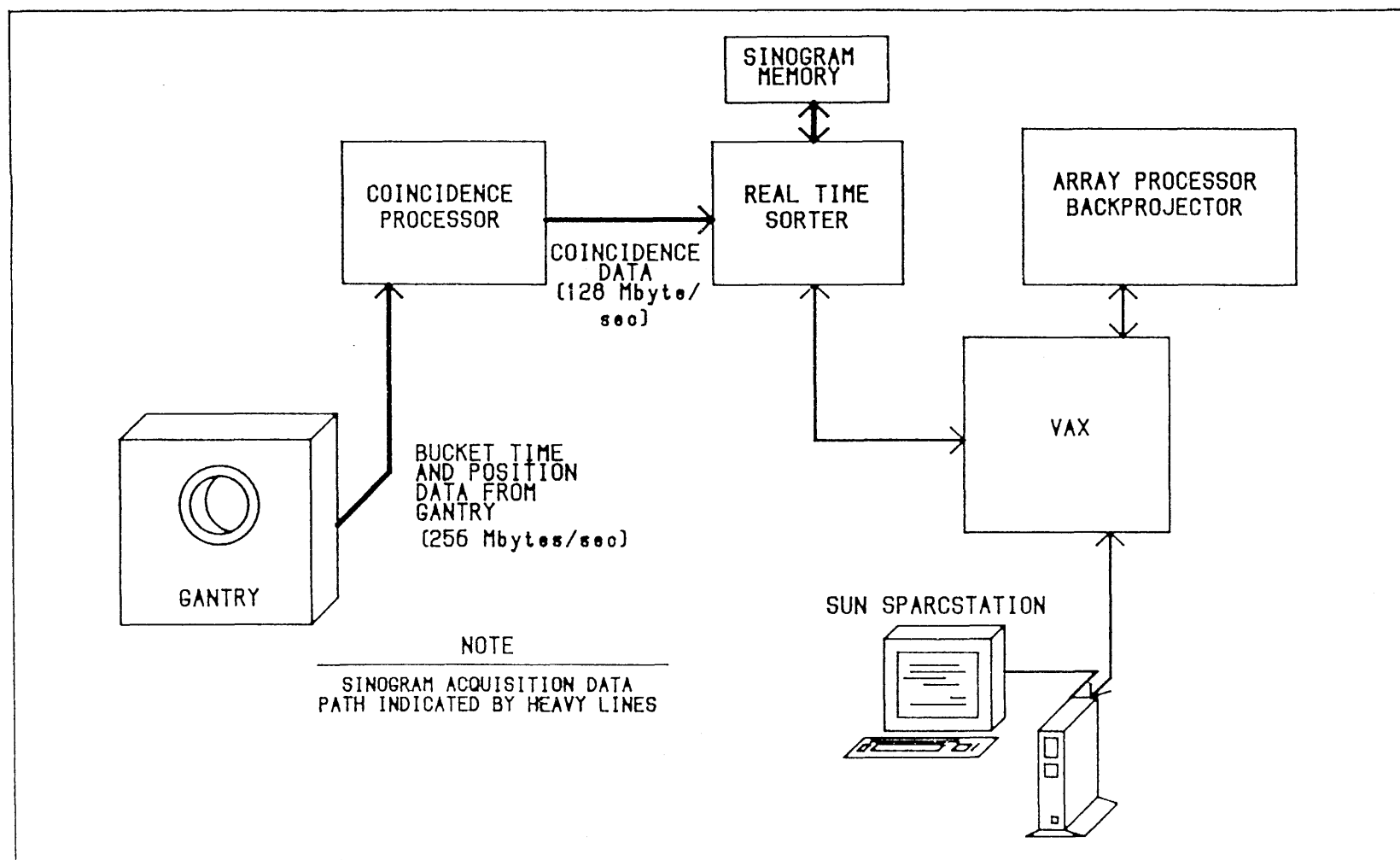


Figure 3.1. Diagram of PET Scanner, highlighting system data flow during acquisition of a sinogram.

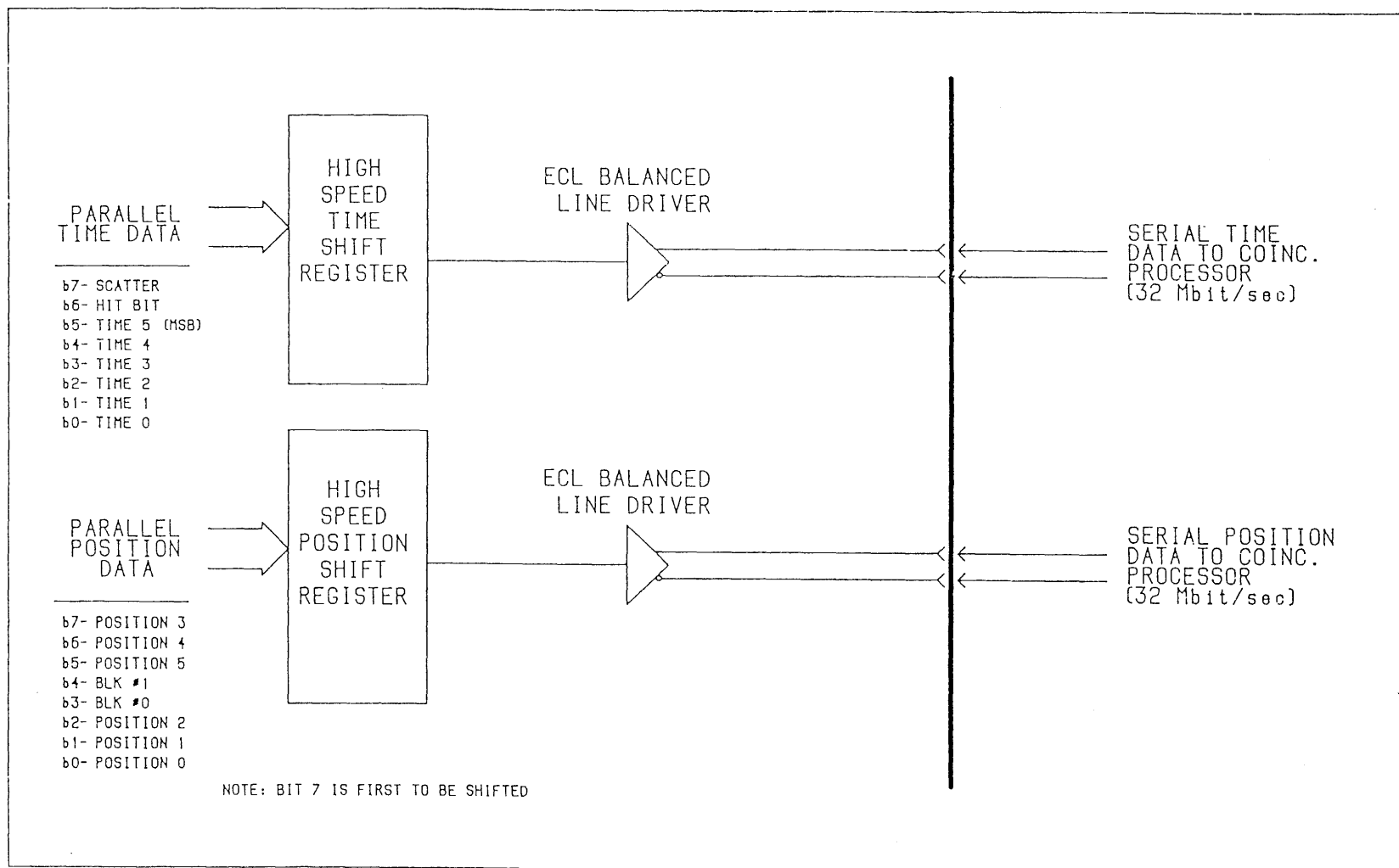


Figure 3.2. Bucket Controller time and position shift registers and associated data.

whether the bucket processed a valid event within the 256 ns interval (HIT bit) and whether the event that was processed fell within the scatter energy window (scatter bit).

In each 256 ns interval, the Coincidence Processor samples the HIT bit from all of the bucket data words and upon locating bucket data words with HIT bit set, examines the time and position information of each valid data word for time values that occurred at the same time (within the coincidence window described below) during the 256 ns interval. These coincidence comparisons take place only on detector pairs that can actually have a LOR through the field of view (FOV) of the gantry. Upon locating two bucket data words that appear to be in coincidence from their time bits, the Coincidence Processor passes this coincidence location (position and angle) to the Real Time Sorter (RTS) where this LOR is histogrammed. The Coincidence Processor immediately begins sampling the next 256 ns interval for another coincidence match. If more than one coincidence is located during a 256 ns interval the Coincidence Processor will randomly select one of these multiple events to pass to the RTS, discarding the other(s). Randomization of this selection process is required to prevent biasing of the accepted data during high rates where these multiple events occur frequently.

Due to the statistical nature of the detector system timing, the Coincidence Processor actually has a time "coincidence window" of acceptance. This coincidence window is an interval of several 4 ns bin widths in which two detectors are considered in coincidence. Timing resolution, the measure of timing accuracy with which a system can resolve the time at which an event occurs, of the detector/electronics system is critical in maintaining a narrow coincidence window. If the detector/electronics system does not have good timing resolution, a wide coincidence window is required to prevent significant loss in scanner sensitivity. If the window width is increased in order to recover this loss, the acceptance of accidental (randoms)

coincidences due to detection of simultaneous annihilation at two or more different sites will occur. The acceptance rate of these random events is given by:

$$S_{\text{ran}} = 2\tau S_1 S_2 \quad (4)$$

where τ = Coincidence Window(sec)

S_1, S_2 = Respective Detector Rates

From this equation, it is apparent that the randoms rate increases with the square of the true coincidence rate. Random coincidences that enter the sinogram as true events result in degradation of the image signal-to-noise (S/N) ratio.

Bucket Data Flow and Block Diagrams

Figure 3.3 shows the data flow through the bucket. When an event is detected by the Constant Fraction Discriminator (CFD) [14], after a well defined delay, the analog processor integrates the fast amplified PMT signals. After approximately 680 ns, the integrator output is summed and divided by the analog-to-digital converters (ADCs) to produce the desired position ratios. The digital representation of this information is then routed into the digital processing electronics where it is used in the determination of the crystal in which interaction took place. The digitized energy signal from the analog processor is used along with the determined position value to perform discrete crystal energy discrimination. This qualified position information is then moved to the output position shift register where it is serialized for transferal to the Coincidence Processor.

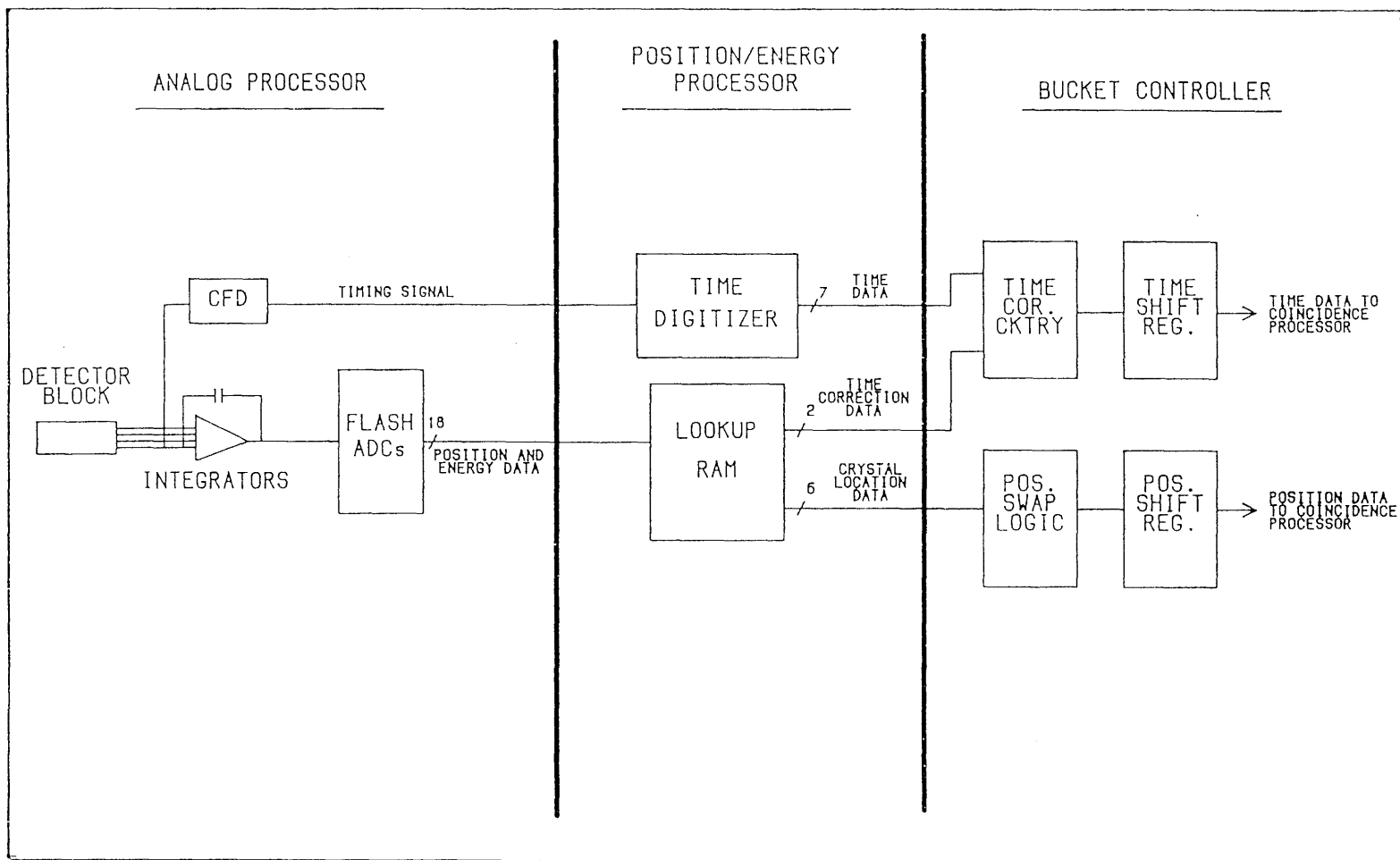


Figure 3.3. Illustration of bucket data flow.

When the Constant Fraction Discriminator observes an event, it produces a fast, very accurate timing pulse that is passed to the Time Digitizer. There the rising edge of this pulse is timed with relation to the end of the present 256 ns interval. This digital value is routed to the output time shift register where it is also serialized for transferal to the Coincidence Processor. These shift registers pass their respective eight bits of data in 256 ns (32 ns/bit).

Figure 3.4 shows the block diagram of one of the Analog Processor sections. Four of these sections (one section per block) make up the front end analog signal processing element of the bucket. Each Analog Processor section represents one printed circuit board.

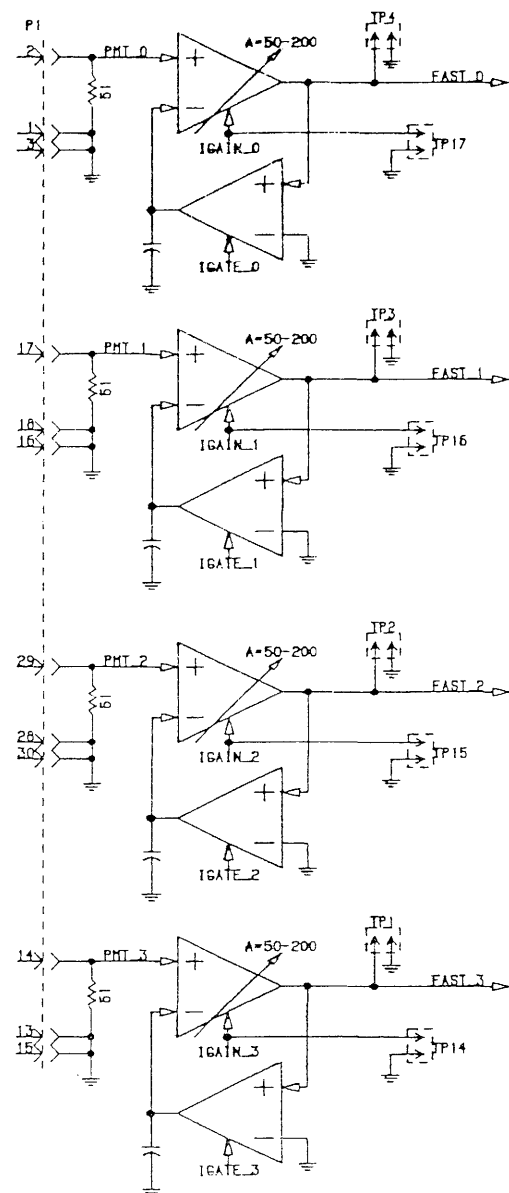
Figure 3.5 displays the block diagram of the Position/Energy processor of the bucket. As with the Analog Processor, there is one Position/Energy processor section for each block. All four Position/Energy processor sections are located on one printed circuit board.

Figure 3.6 displays the block diagram of the Bucket Controller. This segment contains the common circuitry that arbitrates information from the four Position/Energy sections, the time and position shift registers, and the microcomputer circuitry that administrates all of the functions of the bucket. Also included with this circuitry are all of the communications drivers and receivers and the system master clock recovery circuitry. This circuitry is located on one printed circuit board.

System Diagnostics

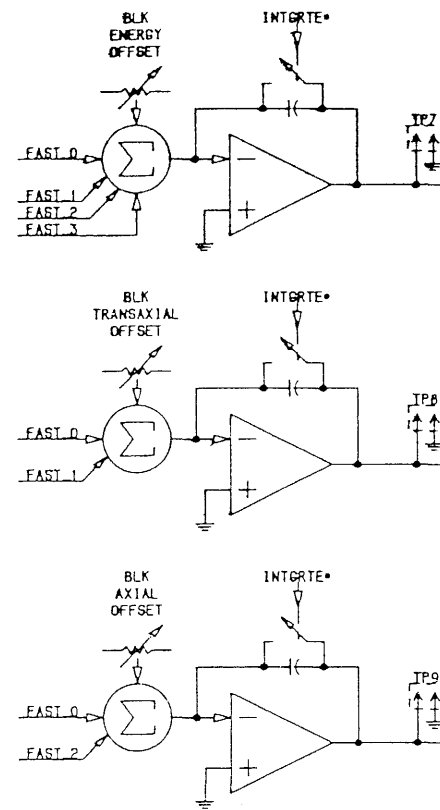
With the extreme complexity of the bucket design, having multiple data paths and busses, some means of diagnosing problems was required. Bucket circuitry volume constraints were the biggest limitation to the amount of hardware features for aiding self diagnosis. Analog Processor system diagnostics were not possible due to physical size limitations of the Analog Processor board. The omission of any ability to feed back DAC outputs through the circuitry in a manner

FRONT-END AMPLIFIERS WITH GATED BLRS

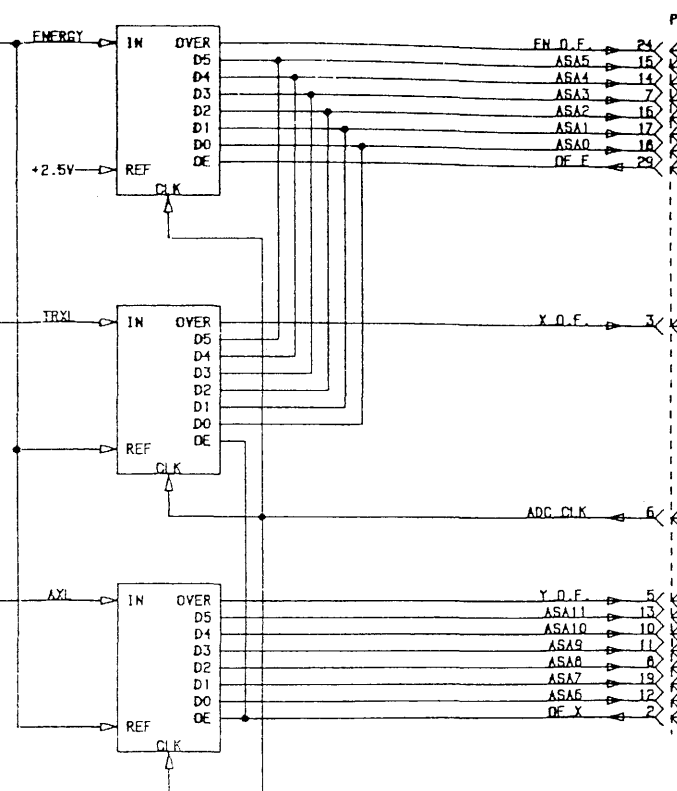


- NOTES: UNLESS OTHERWISE SPECIFIED
1. ALL RESISTOR VALUES ARE IN OHMS
2. ALL RESISTORS ARE 1/8W 5% C.F.
3. ALL CAPACITOR VALUES ARE IN MICROFARADS
4. THE "*" NOTATION INDICATES 2% TOLERANCE
5. THE "**" NOTATION INDICATES 1% TOLERANCE

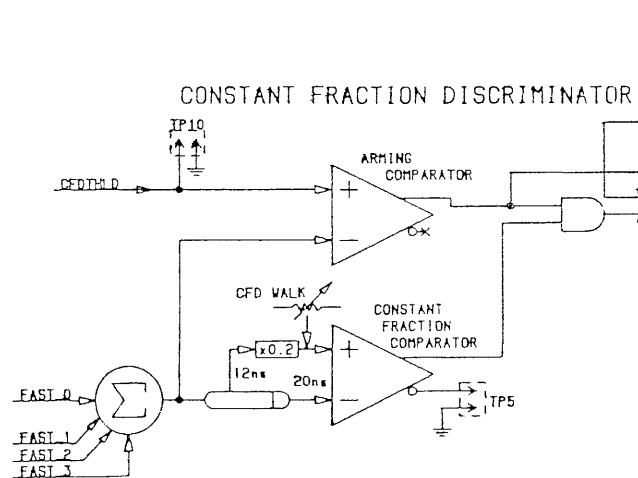
SUMMING INTEGRATORS



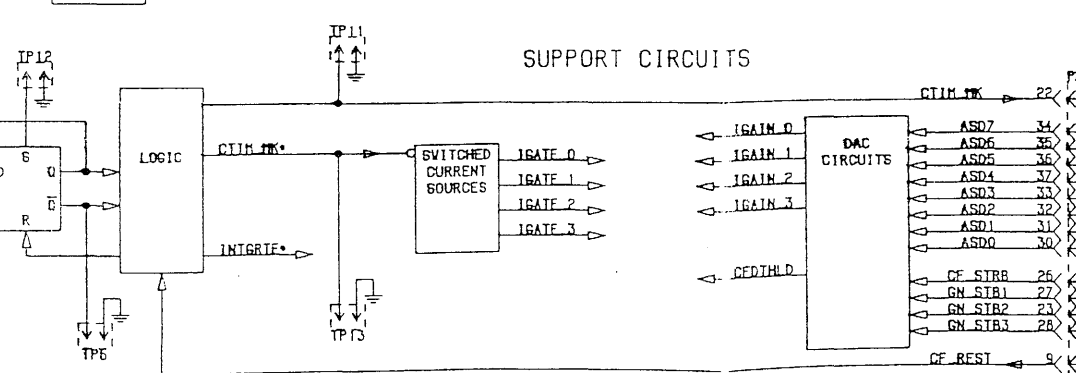
FLASH ADCS



CONSTANT FRACTION DISCRIMINATOR



SUPPORT CIRCUITS



951 BUCKET

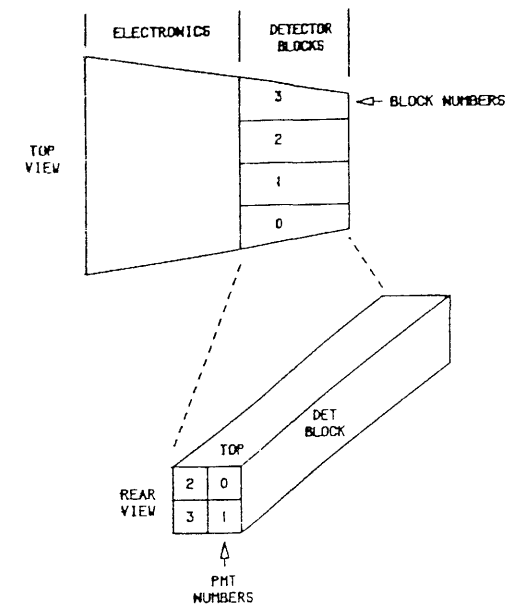


Figure 3.4. Block diagram of Analog Processor section.

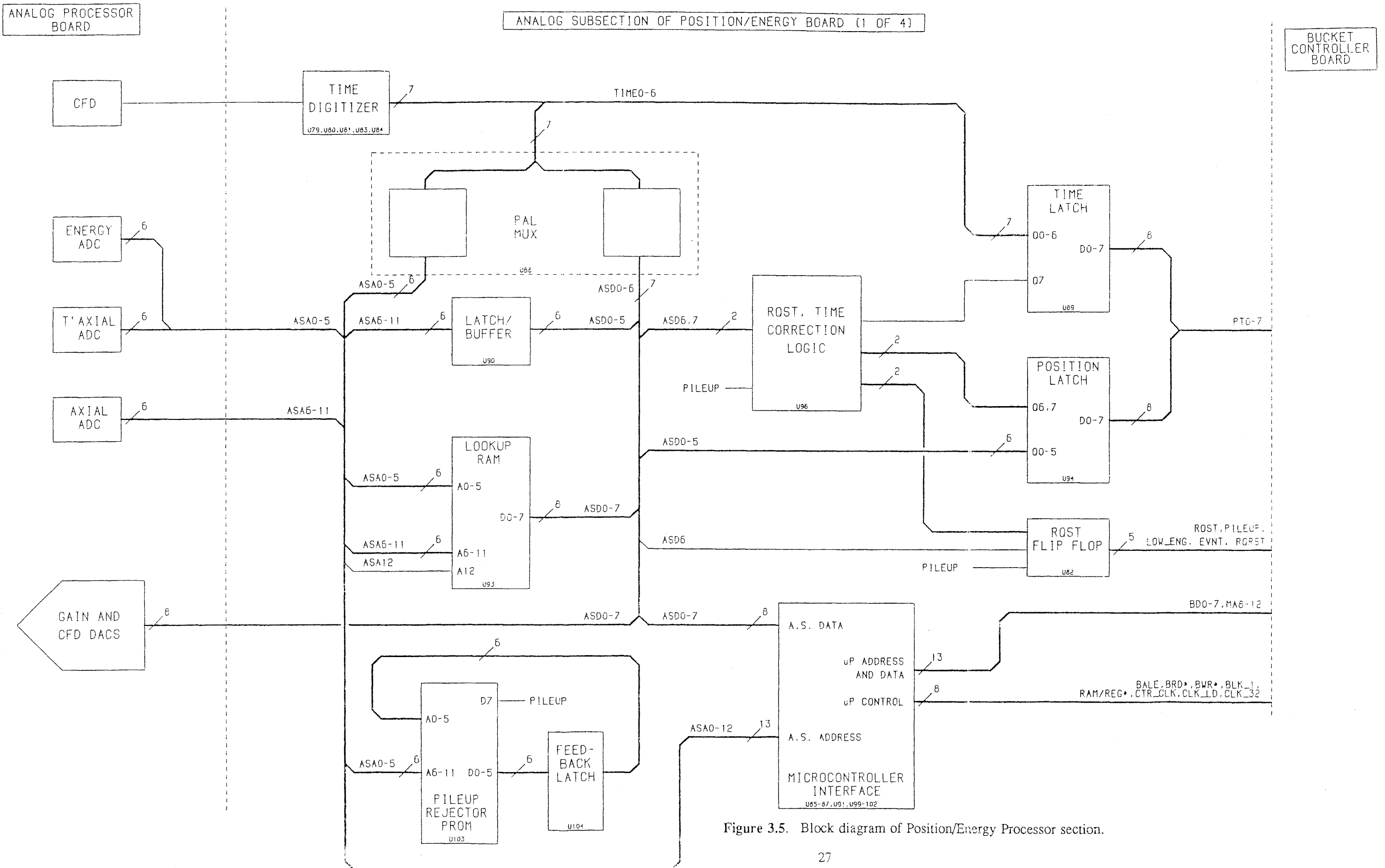


Figure 3.5. Block diagram of Position/Energy Processor section.

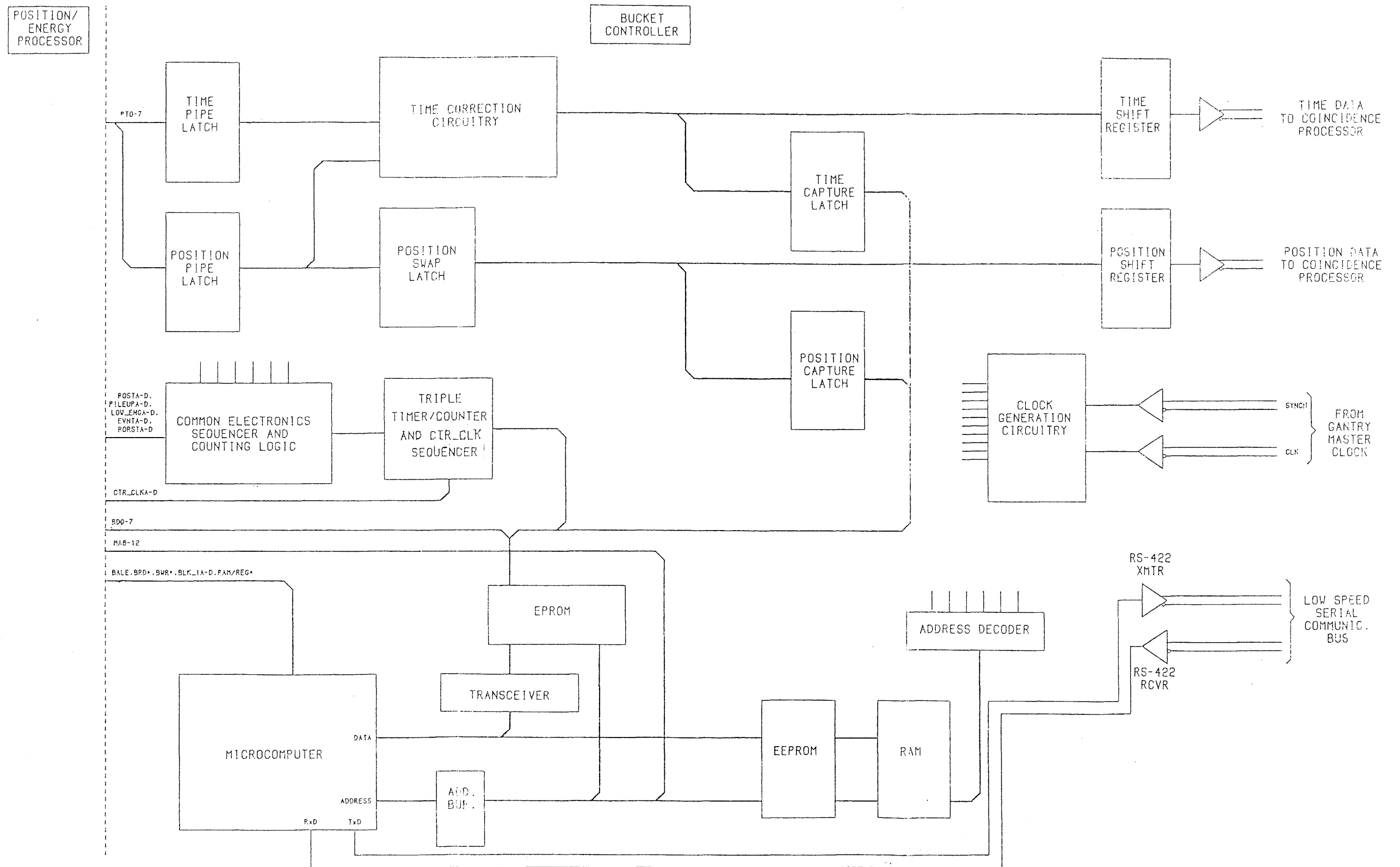


Figure 3.6. Block diagram of Bucket Controller.

in which it could be converted by the ADCs and read by the microcomputer has left a hole in an otherwise comprehensive bucket self-test. Future enhancements to the Analog Processor section which will increase the density of the electronics (i.e. analog ASICs to replace many of the discrete op amps) will allow this loop to be closed.

The Position/Energy processor architecture permitted testing of many of the discrete devices as well as many of the different data path busses within its segment. With each Position/Energy processor section having a direct connection to the microcomputer bus, extensive self testing of these sections is very methodical. The different diagnostics that can be performed on the Position/Energy processor sections and a brief description of their actions can be found in Table 3.1.

The Bucket Controller, containing the microcomputer and its associated peripherals, performs many diagnostics to check the performance of these different components. Diagnostics are included that check the integrity of the static RAM, EEPROM (code and data areas), EPROM, Timer/Counter, and other devices within this section. Table 3.2 lists the diagnostic commands that operate on the Bucket Controller. A complete list of bucket commands can be found in Appendix A.

Firmware Requirements

The hardware complexity of the bucket requires significant firmware to make all of the pieces "play". It must perform setup operations during characterization of the attached blocks. Communications through the low speed serial port must be performed by the firmware. The firmware also performs real-time monitoring of the count rates from the four attached blocks and performs execution of the diagnostic routines .

Table 3.1. List of Position/Energy Processor Diagnostics

Position/Energy Processor Diagnostics Commands		
X<n> - Diagnostics.		
Command	Command Title	Description
X21	Analog Subsection Sequencer Test.	Start sequencer in Pos. Hist. Mode and wait for an interrupt. Error if no interrupt.
X22	Test Analog Subsection Path.	Write data to time and position latches of selected Position/Energy Processor, writing a '1' to RQST bit. On interrupt, processor checks validity of data.
X23	Exercise Tube Gain DAC.	Ramps the selected tube gain-DAC until another <CR> is received.
X24	Exercise CFD DAC.	Ramps the CFD-DAC until another <CR> is received.
X27	Analog Subsection RAM Test.	Writes and reads different bit patterns to selected Position/Energy Processor section's lookup RAM.
X29	Report Analog Subsection RAM.	Reports status of last executed lookup RAM test.

Table 3.2. List of Bucket Controller Diagnostics

Bucket Controller Diagnostics Commands		
X<n> - Diagnostics.		
Command	Command Title	Command Description
X0	Report Bucket Status.	Reports failure of any Bucket Controller or Position/Energy Processor diagnostics.
X17	Report EPROM Checksum.	Executes checksum calculation on EPROM to determine EPROM data integrity.
X18	Report EEPROM Code CRC.	Executes a CRC (Cyclic Redundancy Check) calculation on the EEPROM code data.
X19	Report EEPROM Data Checksum.	Executes a checksum calculation on the EEPROM data segment.
X20	Time Correction Circuit Check.	Writes data to Position/Energy Processor section 0 time and position latch and verifies that Time Correction Circuitry is adding properly.
X25	Scratch Pad RAM Test.	Writes and reads different bit patterns to the scratch RAM.
X26	Histogram RAM Test.	Writes and reads different bit patterns to the histogram RAM.
X28	82C54 Timer/Counter Test.	Writes and reads from the 82C54 Timer/Counter registers.

The single greatest improvement offered by the new bucket is the ability to perform optimization of positioning to the attached detector blocks. This optimization occurs through the execution of a setup routine residing within the firmware. The setup function performed by the bucket firmware can be decomposed into two pieces. First, the firmware must normalize the four PMT signals (via adjustment of fast preamplifier gain DACs) so as to produce a uniform position spectrum across the block. Secondly, after tube gain normalization, the firmware must perform histogramming of first position and then energy so as to determine the location of the position peaks and energy peaks for each crystal.

Figure 3.7 represents, topographically, the position spectra of a typical block after normalization of the PMT gains. In order to normalize the PMT gains, the firmware first loads the lookup RAM to represent a four crystal detector, where each of these four "crystals" represents an area under each of the four PMTs. An iterative algorithm then repeatedly performs an normalization procedure on the four signals. This algorithm first performs an energy histogram for each of these four "crystals" and then, after locating the energy peak of each, adjusts that preamplifier's gain accordingly. The routine then reiterates the above procedure.

After performing the PMT gain normalization, the firmware next performs a position histogram where a three-dimensional histogram of X ADC-value versus Y ADC-value data is constructed. After histogramming for a period long enough to achieve data of low statistical noise, the firmware analyzes the data to determine the spatial locations of the detector peaks. Pattern recognition techniques are performed on the acquired histograms to aid in proper location of the position histogram peaks. Once these peaks are located, the lookup memory of each Position/Energy processor section is loaded to allow position identification.

After loading the lookup RAM with the position information, the firmware next performs energy histogramming, where the energy spectrum of each of the above identified crystals is recorded. The firmware locates the photopeak (representing 511 keV) and scales the high and

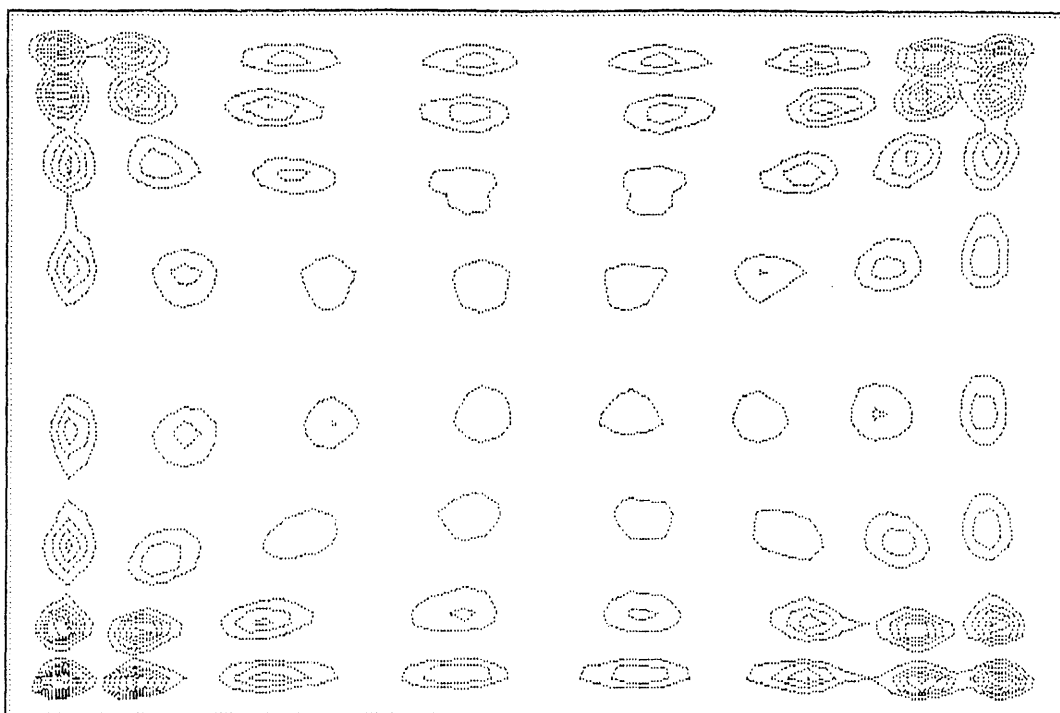


Figure 3.7. Topographic representation of a position histogram of an 8 x 8 block.

low energy discriminator values to this. This energy discriminator information is then loaded into the lookup ram, completing the setup procedure.

The firmware must be able to communicate through the low speed serial port to allow such functions as download of new firmware revisions, modification of acquisition parameters (i.e. changing energy discriminator levels), execution of self tests, and polling of the activity count rate of the four blocks attached to the bucket.

Another function of the firmware is to administrate execution of the different system diagnostics. Most of these diagnostics are performed by placing the specific hardware to be tested in its diagnostics mode and then writing a stimulus to the hardware and observing its response. Some of the diagnostics that are not to be run at power-up include comprehensive memory tests that can not only identify that there is a problem but reports also the stimulus and the response so that visual observation of the report will aid in locating (i.e. which data or address line, stuck

high, stuck low, etc.) the problem . Due to code space limitations, these diagnostics must be downloaded to the bucket for execution. Upon completion of testing, the standard bucket execution code is reloaded. This download process is performed through the low speed serial port. Appendix B [15] lists the different components that are tested by this exhaustive debugging routine. It also lists the tests that are performed on each component and what is reported.

CHAPTER IV

BUCKET ELECTRONIC COMPONENTS

The described bucket is the latest entry in a continuing evolution of PET high density detector electronics. It can support any detector arrangement up to an 8 x 8 block, thereby allowing up to 256 detector elements per bucket (four blocks per bucket).

Vast improvements have been made in the bucket to increase performance and reliability. Hardware improvements include improved CFD performance and threshold repeatability, adjustable fast-preamp gain, reduced integration time, energy discrimination on a per-crystal basis, and pileup rejection circuitry, to name a few.

Manufacturability and serviceability have been improved by placing all circuitry on 6 circuit boards. These boards consist of four Analog Processor boards (same board repeated four times, one for each block), one Position/Energy board, and one Bucket Controller board. The six boards are sandwiched together in a planar fashion with shrouded connectors used for interconnection.

In this chapter the different subassemblies that make up the bucket will be described. Reference to the block diagrams on pages 26, 27, and 28 may be useful.

Analog Processor

The bucket Analog Processor processes signals from the four Photomultiplier Tubes (PMTs) producing a 12-bit digital position value and a 6-bit digital energy value, all of which are used by the Position/Energy Processor Board. It also produces a timing signal for each event that is used by the Time Digitizer on the Position/Energy Processor Board.

Analog Processor Concepts

The philosophy for the design of the Analog Processor was to produce a reliable, repeatable analog front end that had a variable gain, baseline-restored, fast preamplifier followed by both a Constant Fraction Discriminator (CFD) and a set of Summer/Integrators. The output of the integrators produce the transaxial numerator ($TUBE0 + TUBE1$), axial numerator ($TUBE0 + TUBE2$), and energy signal (denominator) ($TUBE0 + TUBE1 + TUBE2 + TUBE3$) to be used in the ratio positioning scheme discussed previously.

Variable Gain Preamplifier/Baseline Restore Circuitry

An event's current pulse out of the photomultiplier tube is converted to a voltage signal by the 51 ohm resistor (i.e. R72, Figure 4.1). This value of resistance allows convenient termination of a standard shielded cable. The voltage developed across the termination resistance is then applied to the variable gain input preamplifier. A dual operational transconductance amplifier (OTA), U4, serves as both the variable gain element and the baseline restoration amplifier. A wideband current feedback operational amplifier, U6, is used for added gain and buffering after the OTA variable gain preamp.

The OTA gain block is a traditional (transistor) differential pair input stage which is followed by a pair of current mirrors connected to each collector of the differential pair. One collector current mirror is mirrored again off of the V- rail with this output tied to the current mirror output off of the other collector of the differential pair. Combining these two signals results in the current output of the amplifier which is then converted back to a voltage by the resistor R11. The voltage to current gain (I_{out}/V_{in}) of the OTA is set by the current through the

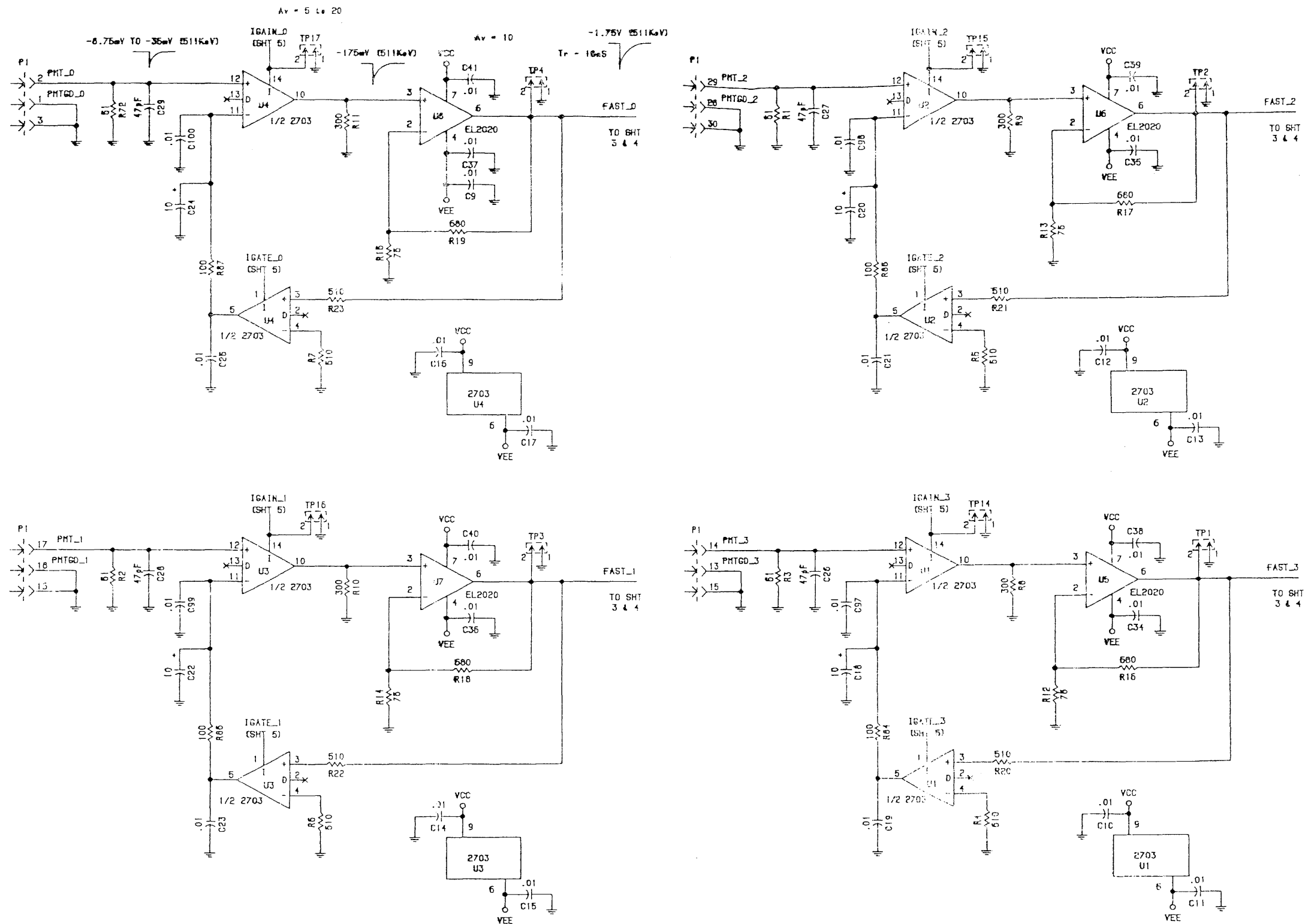


Figure 4.1. Schematic diagram of Variable Gain Preamplifier and Baseline Restoration circuitry.

front-end differential pair transistors, which is supplied externally to the amplifier via the IGAIN input.

One OTA does not provide the required gain or dynamic range to achieve the required output signal. The output impedance of this amplifier requires buffering before driving the summing stages. These limitations mandated the additional gain block U6. This amplifier provides the needed dynamic range and buffers the high output impedance (basically 300 ohms, R11) of the variable gain front-end OTA.

Due to operating the photomultiplier tubes with +1500 volts on their anodes, capacitive coupling from the anode to the preamplifier is required. This capacitive coupling creates a high pass filter that will have a DC charge buildup at high count rates. At high count rates, this charge buildup appears as a DC offset at the input to the preamplifier. This offset will be integrated by the gated integrator, adding a linear ramp (integration of the constant offset) to the integrated event signal. The detrimental effect of this is that it will cause movement of the photopeak over varying count rates, thus resulting in the need for dynamic offset compensation, i.e. baseline restoration.

The baseline restoration circuitry is comprised of a gated amplifier that samples the output during no-signal periods and feeds back a signal into the front-end OTA to zero any output offset. During the signal processing interval, this amplifier is gated off, and the fed back offset correction signal is held by capacitor C24. Capacitors C100 and C25 provide good high frequency attenuation which assures acceptable loop stability.

The gating of the Baseline Restore amps is performed by U24 and transistors Q11-14. When the CFD observes an event, the TIM_MRK signal goes active causing Q11-14 to "turn off" and no longer source current. This turns off the Baseline Restore amplifiers so that no correction takes place during signal processing.

Constant Fraction Discriminator

The Constant Fraction Discriminator (CFD) (Figure 4.2) produces a timing pulse that occurs exactly a predetermined delay-time after the occurrence of a detector pulse, independent of pulse height. The CFD consists of an arming comparator U12 (upper section of CFD schematic), a constant fraction (CF) timing comparator U12 (lower section of CFD schematic), and an arming/blocking logic circuit.

The arming comparator is basically a fast discriminator that monitors the amplified input signal from the preamps and produces an active output for signals of amplitude greater than the discriminator threshold (CFDTHLD). This signal "arms" the flip-flop U20, preparing it for a timing edge from the CF comparator.

Delay line DL1 along with resistors R38 and R26 produce the Delay input and Fraction input required to make the CFD zero-crossing decision. Inversion of the delayed signal is accomplished by feeding one signal (Fraction) into the "+" input while the other signal (Delay) is fed into the "-" input of the CF comparator. The zero crossing of this "summed" Fraction and inverted Delay signal is detected by the CF comparator, producing a precise clock edge to the flip-flop. This clock edge produces the TIM_MRK signal if the signal meets the level requirements of the arming comparator. Since the CF comparator is required to transition on zero-crossing signals, the comparator threshold is set precisely to zero by potentiometer R76.

Due to the CF comparator threshold being set to zero, the output of this comparator is seen to transition on system noise when no signal is present. This is why the arming comparator is a vital part of a CFD system. Only when a signal of valid amplitude is observed by the arming comparator (setting its output, and thus the D of the flip-flop, active) is the CF comparator transition allowed to propagate through to TIM_MRK. This delay in both of these signals allows the arming comparator enough time to make its threshold decision and meet the setup time for

D of the flip-flop before the clock edge generated by the CF comparator.

The logic consisting of U16 and transistor Q10 make up the reset circuitry of the CFD. When an event occurs in normal system operation the CFD produces a TIM_MRK signal. This signal starts a sequencer on the Position/Energy Processor that times the integration period and steps the data through the system. CF_RESET, from the Event Sequencer, goes active at the end of the integration period causing the CFD flip-flop to be reset via level-shifter Q10 and U16.

The CFD will operate without a sequencer driving CF_RESET due to the action of the monostable multivibrator consisting of R59, C81, and U16. When an event triggers the CFD, U16-3 (pin 3) falls at a time constant equal to R59 times C81. At the time when the level on U16-10 gets to a valid ECL-low this gate transitions, resetting U20. The delay created by this simple R-C circuit is designed to be longer than the sequencer start-to-reset interval so that this circuitry has no effect in normal system operation.

Axial/Transaxial/Energy Integrators

To achieve desirable statistics for accurate positioning and energy discrimination, the fast preamplifier outputs are summed together and then integrated over the period set by the sequencer on the Position/Energy Processor. There are three integrators that create the ENERGY, TRANSAXIAL, and AXIAL outputs used in the positioning and energy discrimination decisions.

The AXIAL signal is achieved by integrating the summed FAST_0 and FAST_2 signals. Likewise, the TRANSAXIAL signal is created by integrating the summed FAST_0 and FAST_1 signals. Finally, the ENERGY signal is created by integrating the FAST_0, FAST_1, FAST_2, FAST_3 signals.

Functionally, the three integrators have identical architecture. Summing of the fast signals takes place via a voltage input/current summing arrangement into the inverting(-) input of the

integrator op amp. The integration capacitor, attached between the output and the inverting(-) input of the op amp, is reset by the MOSFET U13a (pins 7,6, and 5). MOSFET U13b (pins 3,1, and 2), attached to the non-inverting(+) input of the op amp, is used to cancel the effect of the charge injection into the integration capacitor by MOSFET U13a.

Gating of the integrators occurs when the gate of the MOSFET across the integration capacitor is taken to a logic low value. This is performed by U24, which buffers the TIM_MRK* signal. When an event is observed by the CFD, TIM_MRK* goes low, driving U24-6 low, turning off the MOSFET and thus enabling the integrator.

Ratiometric Converters

The integrated outputs are fed to the ratiometric flash converters (Figure 4.3) where their analog values/ratios are converted to a 6-bit value. Six bits of data are produced by each ADC representing:

$$\text{ADC Value} = 63 * V_{in} / V_{ref} \quad (5)$$

Data conversion is a two step process for these flash converters. ADC_CLK is pulsed once to sample the analog ratio and to latch this sample into the latches attached to the internal comparators that make up the flash converter structure. ADC_CLK is then pulsed a second time to move the data from the comparator latches into the output latches. The converted ADC output data can only be observed after this second pulse of ADC_CLK.

Two ADCs, as described earlier, produce the AXIAL and TRANSAXIAL 6-bit values representing the one-dimensional linear position of the event. Another ADC produces the ENERGY value representative of the absolute energy of the event: the sum of energies out of all four PMTs.

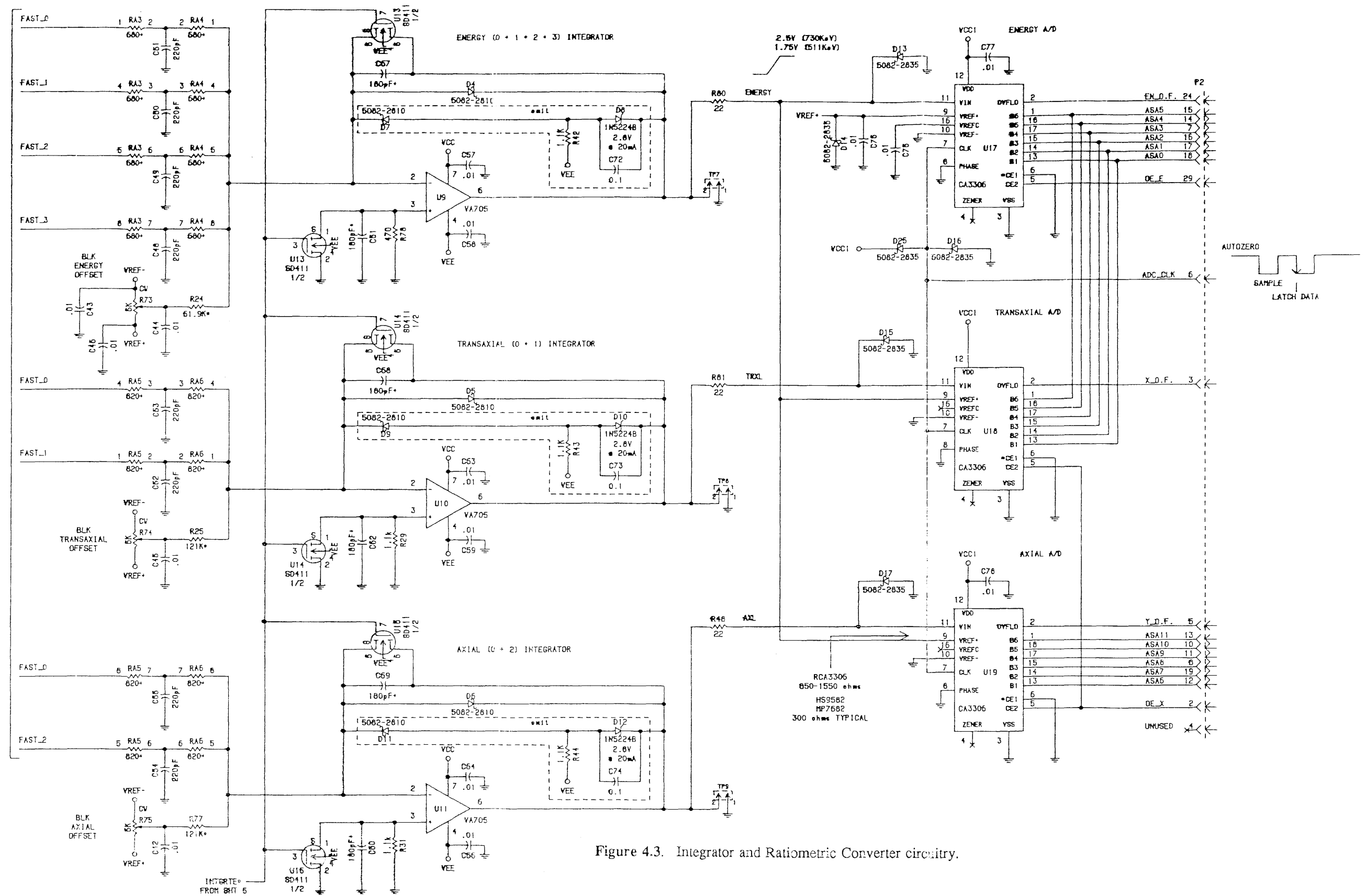


Figure 4.3. Integrator and Ratiometric Converter circuitry.

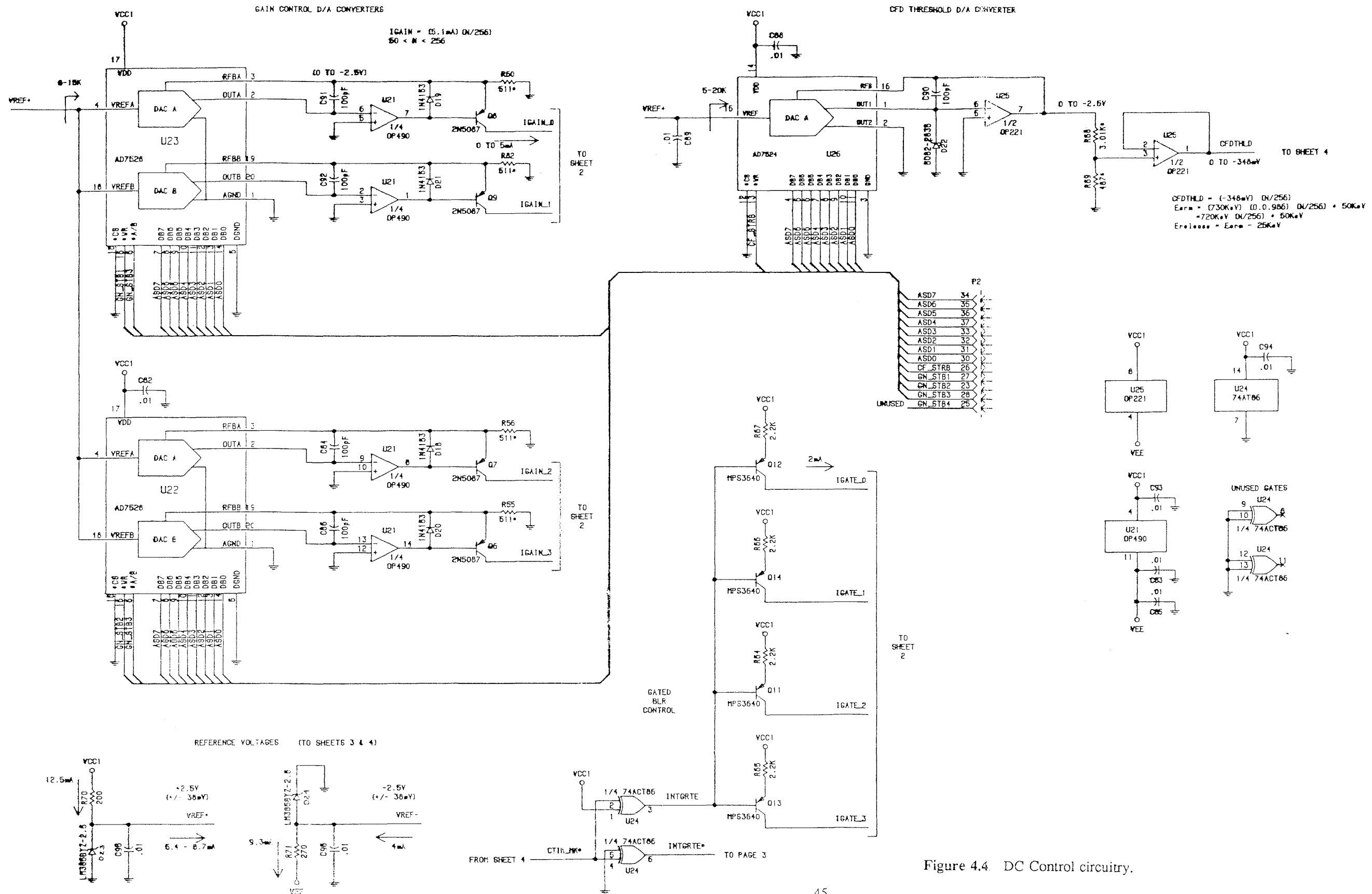
DC Control Circuitry

The CFD arming threshold and front-end preamplifier gain adjustment are under microcomputer control. Two dual DACs, U22 and U23 (Figure 4.4), along with the quad op amp U21 and transistors Q5, 7, 8, and 9 make up the four variable current sources used to set the preamp gains. DAC U26, and amplifier U25 make up the variable voltage source that is used as the arming threshold for the CFD.

Position/Energy Processor Board

The Position/Energy Processor board contains the circuitry to perform crystal position lookup and energy discrimination. This circuitry is repeated four times on the Position/Energy Processor board, one section for each block. Crystal position lookup and energy discrimination is performed from the data supplied from the ADCs on the Analog Processor circuitry.

By having independent circuitry for the four blocks, parallel processing of independent events in different blocks is possible. Independent events that occur in different blocks of the same bucket within different 256 ns intervals will be processed and transferred to the Coincidence Processor. As is discussed later, two events that occur in separate blocks during the same 256 ns interval will be arbitrated by the Bucket Controller circuitry with one of the two events being transferred while the other is lost. Only one of the four Position/Energy Processor sections is represented in this discussion.



Detector Interface

The four Analog Processors are located on separate printed circuit boards due to the two high density connectors required for interconnection to the associated Position/Energy Processor sections. Concern over dimensional tolerance buildup across the Position/Energy board during the manufacturing phase prompted isolation of each Analog Processor onto separate printed circuit boards.

The problem with separating the Analog Processor sections onto different boards is that of routing the high voltage (1500 Vdc) to the detector PMTs. Placing the connectors for the four PMTs on the associated Analog Processor board would have been highly desirable since lead length of the PMT output signal should be kept minimal. The problem with this method is that high voltage for the PMT bias string must be routed to the PMT through this connector. This means that high voltage must be connected to each Analog Processor either by wire or by connector from the Position/Energy Processor, neither of which is a very desirable option.

Because of the need for high voltage at each PMT connector, the PMT connectors were located on the Position/Energy Processor board (Figure 4.5). By mounting these connectors here, high voltage can be brought to this board at one point and then routed via printed circuit board copper to the many PMT connectors (16 connectors per bucket). High-voltage, high-frequency bypassing is performed where the high voltage is brought onto the Position/Energy Processor and at each PMT connector.

Much care was taken in routing the PMT signal from the PMT connector, up through the Analog Processor connector to the fast-preamp. PMT ground signal was routed through the connector also, eliminating connection of PMT ground to the noisy digital ground plane prevalent on the Position/Energy Processor board. A return resistor was included between this analog ground and the digital ground on the Position/Energy Board. By providing a return path for the

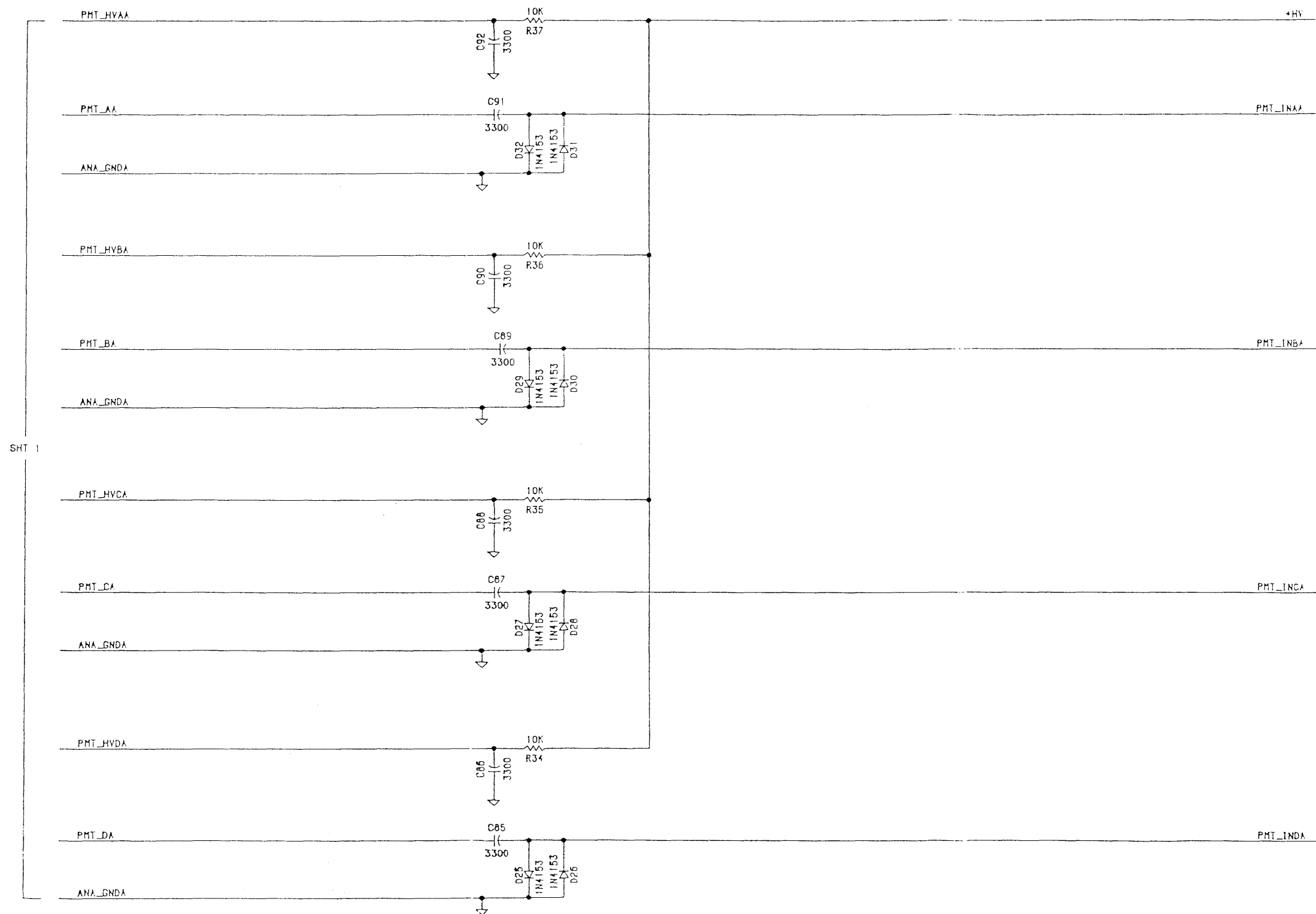


Figure 4.5. Detector Interface schematic diagram.

PMT bias string, this resistor prevents the analog ground from "floating up" to 1500 volts if the Analog Processor board is not present. The analog connection between the Position/Energy Processor and the Analog Processor was made through a shrouded 32-pin connector.

Analog Processor Interface

The Analog Processors not only require the analog interface to the PMT connectors of the Position/Energy Processor, but also require digital interface for both the ADCs and the DACs. Many control signals as well as Vcc and ground are also required by the Analog Processor. This interface is made through connector J24, a 40 pin shrouded connector located at the back of the Analog Processor.

Clock Generation Circuitry

The Clock Generation Circuitry creates the critical timing waveforms required within the bucket. Extreme care was taken in choice of circuit components as well as printed circuit board layout of many of the signals in this section. Microstrip techniques were used on a number of these signals in order to facilitate signal transmission with acceptable signal integrity. Series termination of these transmission lines was the method of choice, although parallel thevenin termination at the end of the CLK_16 line was required due to distributed loads throughout its length.

Two clock signals, TCLK and TSYNC, are brought on to the bucket via the gantry master clock. These signals arrive at the bucket via twisted-pair balanced transmission lines from the gantry master clock. This master clock has the circuitry to generate and distribute both signals through ECL balanced drivers. Special care in maintaining consistent component and circuit board

delays for all outputs from this clock board is applied. The cables for each bucket within the gantry are cut to exactly the same length so as to have the edges of these signals arrive at the buckets at precisely the same time. The ECL balanced signals from these clock cables is converted to TTL levels and then routed to the Clock Generation Circuitry. This circuit is shown in Figure 4.6.

Figure 4.7 shows the timing of the derived signals within the Clock Generation Circuitry. The CLK_EOC* signal is used by the Time Digitizer to determine the end of the 256 ns timing interval. CLK_256 follows the CLK_EOC skewed by 16 ns. It is also used in various circuitry to represent the end of the 256 ns interval. CLK_LD is used to synchronize the output time and position shift registers to the proper 256 ns interval. Finally, CLK_32 is used throughout the bucket for a general purpose, high speed, synchronous clock.

Time Digitizer

The Time Digitizer (Figure 4.8) produces a seven-bit number representing the time from the activation of the CFD to the end of the present 256 ns time interval. This seven-bit value subdivides the 256 ns timing interval into 2 ns bins. Later concatenation at the Bucket Controller produces a six-bit value of 4 ns resolution that is transmitted to the Coincidence Processor.

The Time Digitizer consists of two sections: a fine time interpolator and a coarse time counter. Since the finest resolution clock can only resolve 16 ns intervals, a delay line interpolator is used to subdivide this 16 ns period into 2 ns intervals. Figure 4.9 shows the timing waveforms present in the Time Digitizer circuitry. For a thorough understanding of the clock waveforms, refer to the section on the clock generation circuitry (page 48).

When an event is detected by the CFD, the TIM_MK signal rises. This rising edge propagates as a wave front through the delay line appearing at each delay tap 2 ns later than the

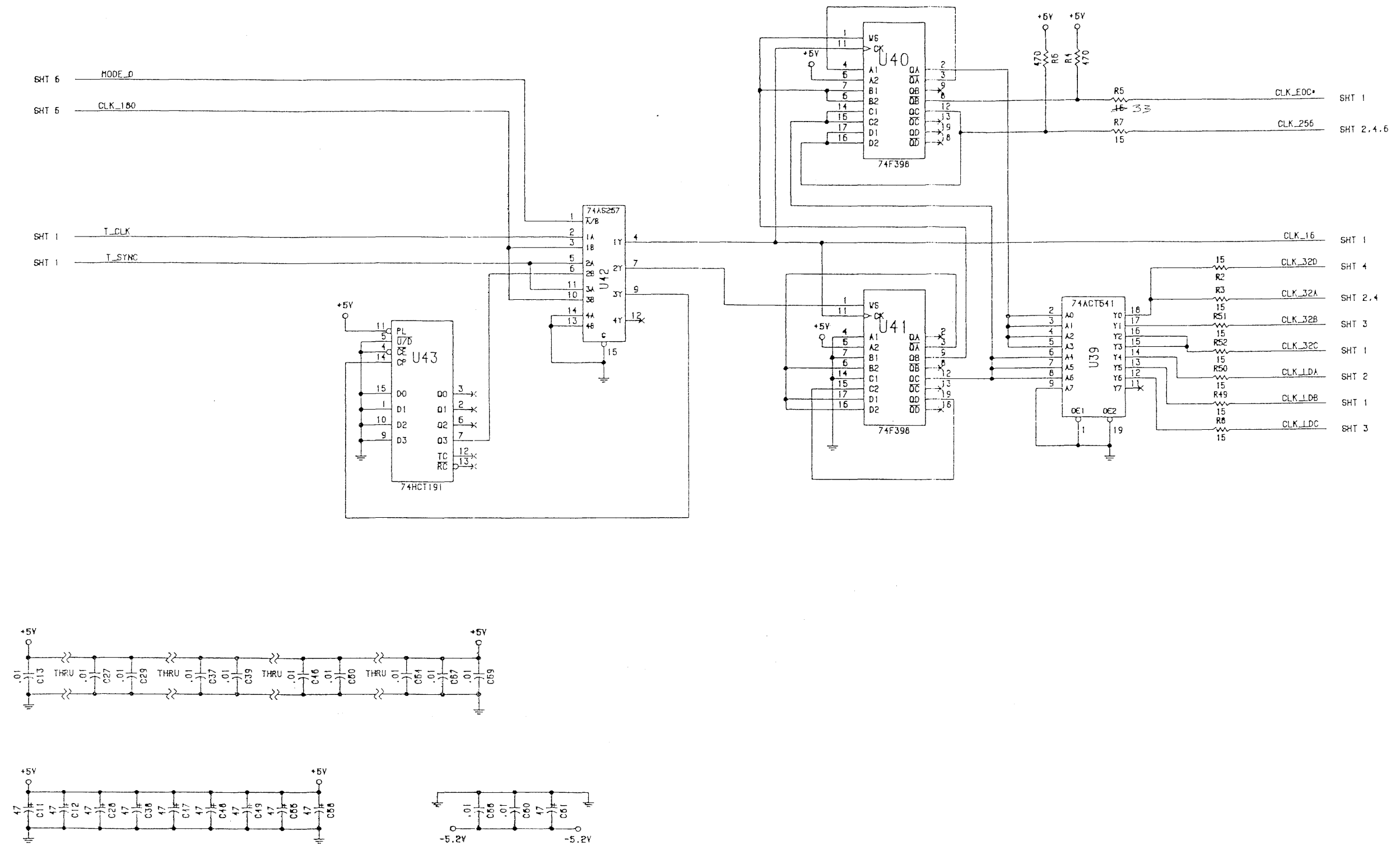


Figure 4.6. Schematic diagram of Clock Generation Circuitry.

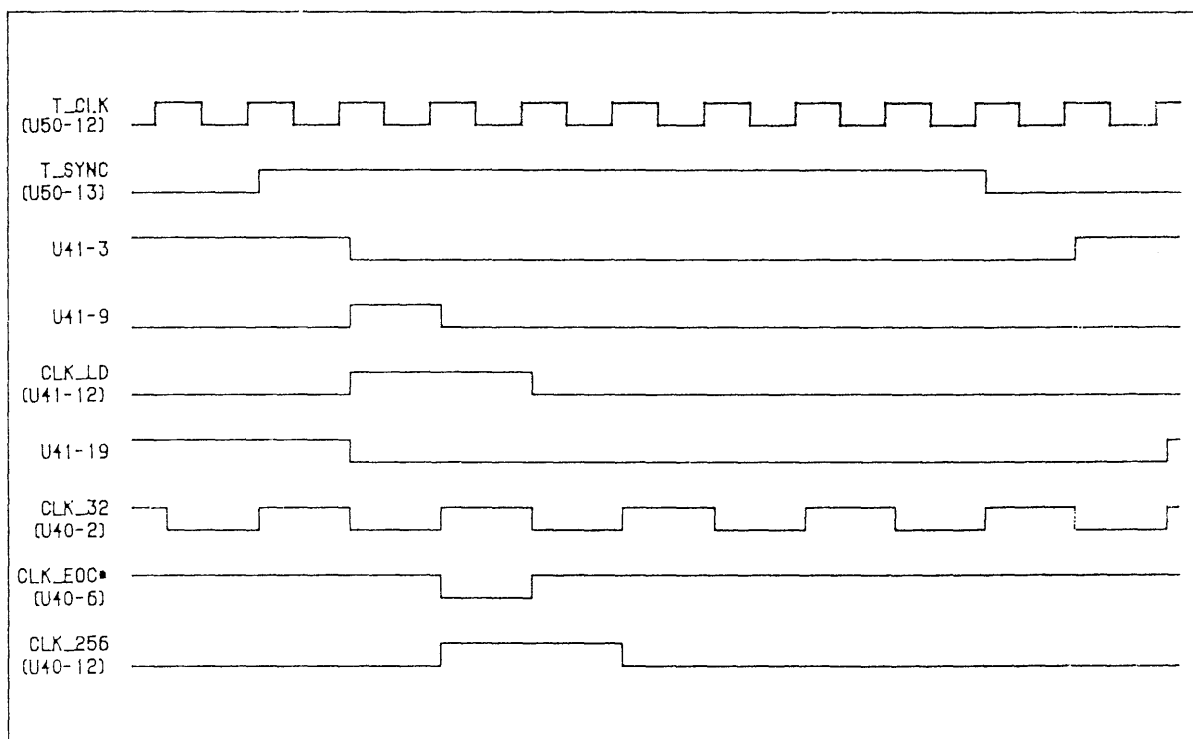


Figure 4.7. Timing diagram of signals generated by Clock Generation Circuitry.

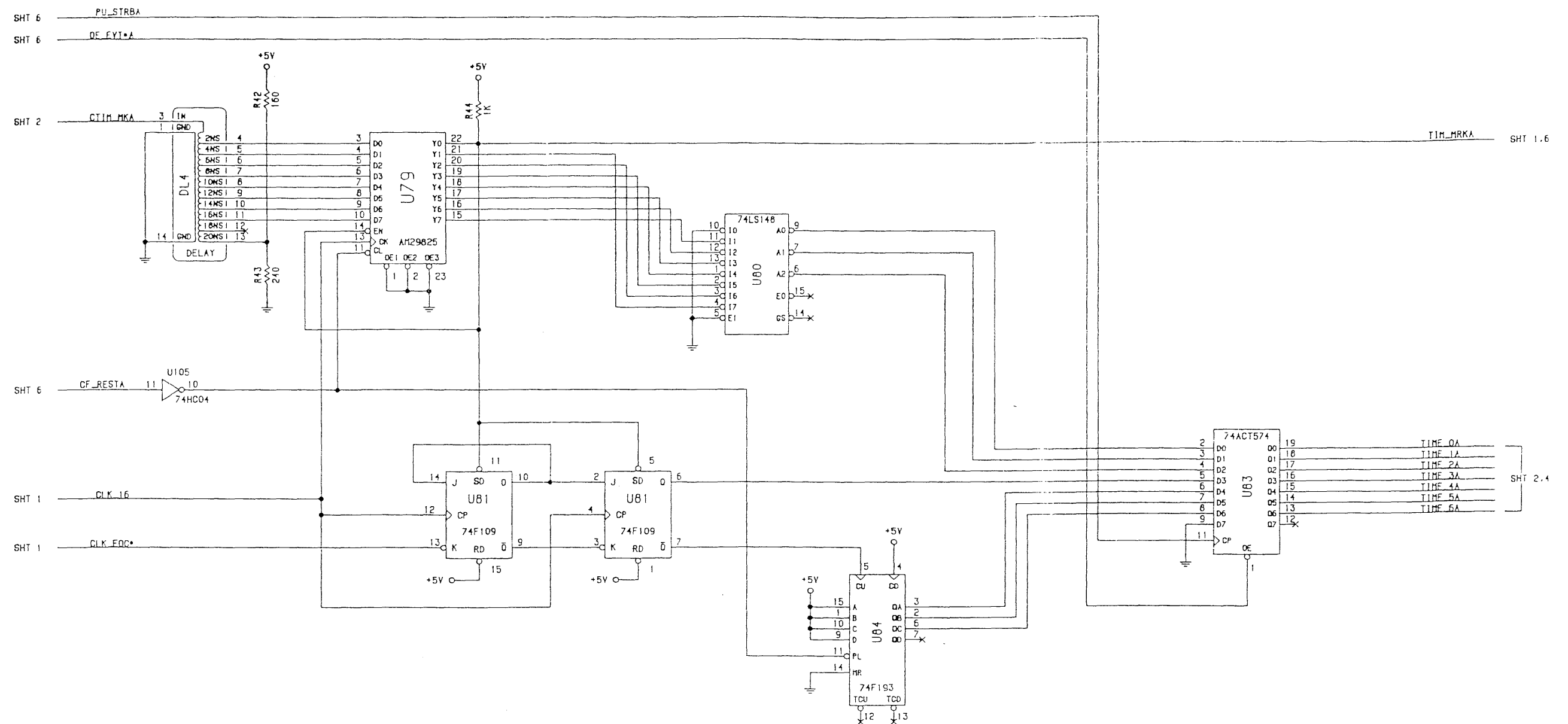


Figure 4.8. Schematic diagram of Time Digitizer.

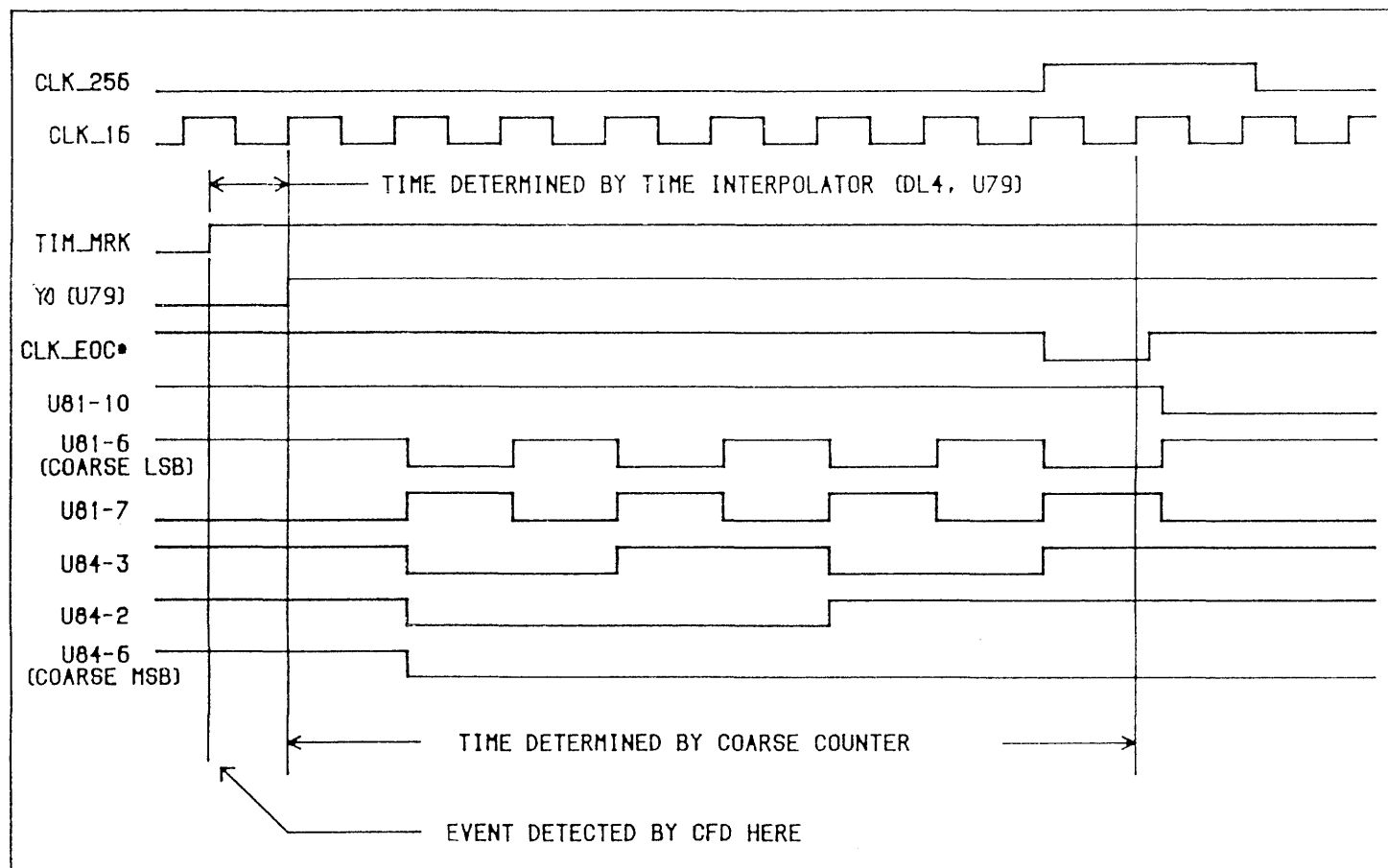


Figure 4.9. Timing waveforms within Time Digitizer.

previous. At the next rising edge of CLK_16, the status of each of these delay line taps is latched into U79. The output of the latch is directed to a priority encoder, whose output presents a three-bit binary value representing the location of the delay line wave front at the time that the CLK_16 signal rises. Flip flop U81 prevents retriggering of the latch until the present 256 ns interval is complete. To complete the subdivision of the 256 ns interval, a counter, U81 and U84, counts the number of CLK_16 intervals until the end of the period. This four-bit value is combined with the three bits from the priority encoder and latched into register U83 to produce the seven bits of Time Digitizer data.

Positioning/Energy Discrimination

Positioning and Energy discrimination is performed by a table lookup process where data from the ADCs on the Analog Processor is input to the address of the lookup memory. Via this process, a detector crystal number (one of 64 for an 8 by 8 detector block) is "looked up" from a previously loaded table. Energy discrimination is performed in a similar manner, where the crystal number (six bits) and the energy ADC value (six bits) are presented to another lookup table. This second lookup table references the energy histogram of the specified crystal (indexed by the crystal number) and outputs a bit indicating whether the energy value from the energy ADC is within the energy discriminator window.

When an event is processed by the Analog Processor and the information (X, Y, and Energy) has been converted by the ADCs, the Event Sequencer then proceeds to move this data through the lookup table. For economic reasons, one 8K x 8-bit static RAM is used for the lookup memory. Using only one device mandated a multiple pass scheme of data processing.

The first pass represents position lookup. X and Y data from the Analog Processor ADCs is presented to the lookup RAM as a 12-bit address (A0-11). The Event Sequencer drives A12

low at this time, pointing into the position lookup table of the RAM. The thirteen bits of address presented to the lookup RAM represent a unique two dimensional spatial location within the block that, by this lookup process, is binned to represent the crystal (one of 64) in which the event interacted.

The six-bit crystal value is output on data lines ASD0-5, where it is latched into the latch/buffer U90 for use during the second lookup pass. Generated during this lookup phase are two bits of data utilized for: (1) event rejection due to location of the event falling within an area of high positioning uncertainty, and (2) time correction of event occurrence time value. This time correction value allows, on a crystal by crystal basis, the ability to adjust the Time Digitizer value by 0 or ± 2 ns. This adjustment can be used to overcome nonuniformities in timing across the block (or bucket) caused by photomultiplier tube transit time differences, fast preamp timing skew, and clock skew on this board. These two bits are latched at the end of the first lookup pass by registers internal to U96, where they are routed to extra bits in the time and position latch for transmission to the Bucket Controller. The Bucket Controller contains the circuitry that performs the time correction addition/subtraction.

The second pass of the data processing involves routing the six-bit crystal (ASA6-11) value from U90, and the six-bit energy value (ASA0-5) from the Analog Processor onto the address lines of the lookup RAM. Event Sequencer output ASA12 goes high during this pass, pointing into the lookup RAM area for energy discrimination. This energy discrimination pass performs an energy histogram lookup for the specific crystal and then compares the energy ADC value to the values within the defined discriminator window. The comparison results are represented by data line ASD7. ASD7 is routed through U96 (Figures 4.10 and 4.11) to the RQST flip flop U82 where it is latched at the end of event processing by the rising edge of PE_STRB. PE_STRB is routed to the RQST flip flop through U96.

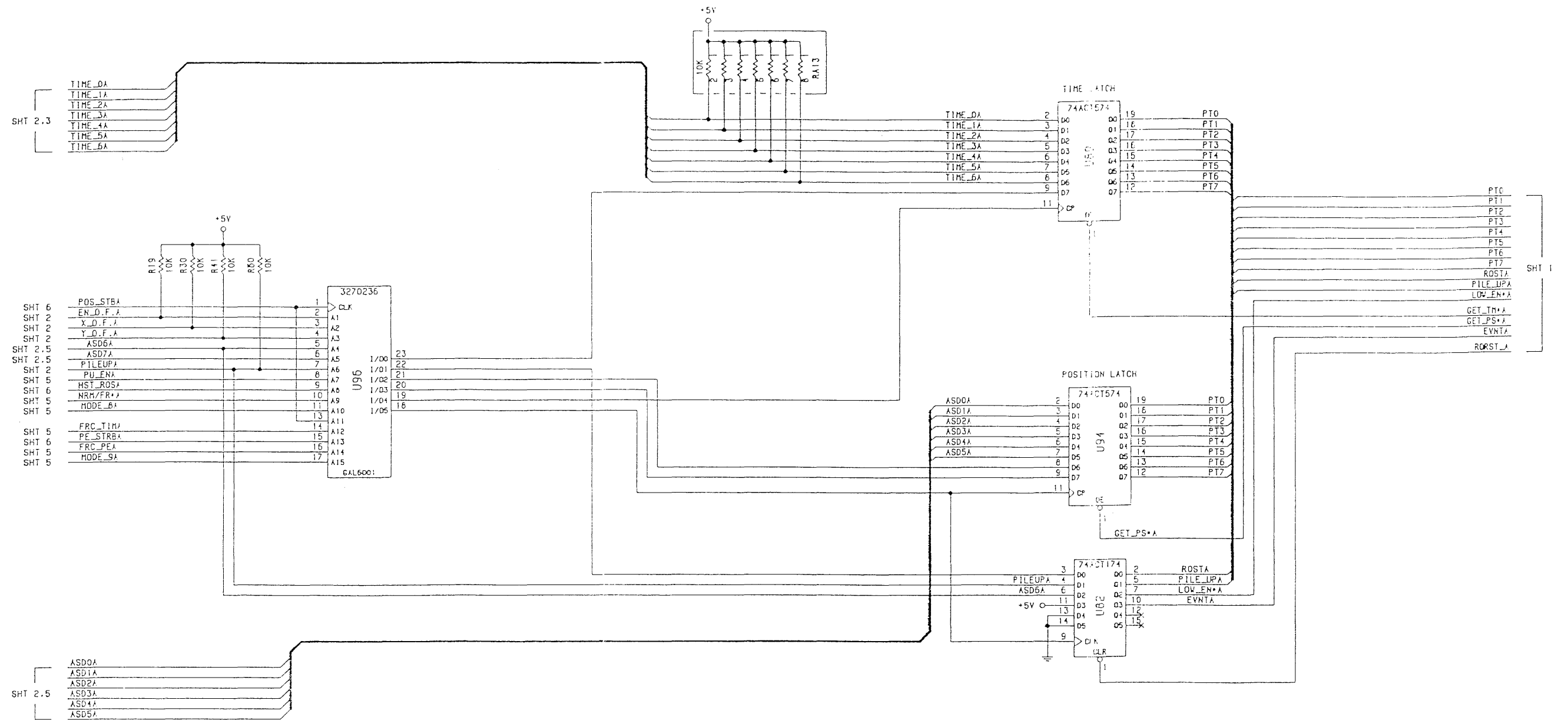


Figure 4.11. Schematic diagram of Time/Position section of Position/Energy Processor.

A second energy discriminator window (scatter window) is also available in the lookup scheme. This second discriminator's output is on ASD6 and gets routed through U96 to the position latch for transmission to the Bucket Controller. The Bucket Controller appends this bit to the six-bit time word to be sent to the Coincidence Processor. There, this bit is used to tag event data as to whether the event fell within the normal energy window or the scatter energy window. By histogramming these two types of events separately, a sinogram of scattered events as well as non-scatter events is produced. The scatter sinogram represents a two-dimensional scatter density which can then be used to correct the normal sinogram for tissue density variations, resulting in more accurate quantitation and increased image S/N ratio.

Pileup Rejector

One fundamental problem with the multi-element detector block philosophy arises from the fact that only one crystal out of the 64 can be identified during the electronics' processing period. This is not only a dead-time issue with the electronics processing, but is due to the relatively slow light decay constant of BGO (300 ns). If another particle interacts within the block during the time in which the light from the first interaction is being processed, mispositioning will occur. This phenomena is referred to as pileup, and results in blurring of image edges at high count rates. This blurring is the result of the position ratioing scheme placing the location of the two events on a line between where the two interactions took place. Thus mispositioning error can be no more than half of either dimension of the block.

A rudimentary method to perform pileup rejection is to place the high-level energy discriminator level near the upper edge of the 511 keV photopeak. In doing so, the majority of piled up events' summed energies fall above this level and are rejected. This phenomena is due to the fact that the event energy density function (histogram) is centered around the 511 keV

peak. Thus if two events occur during the processing period, summing produces a second smeared peak centered at twice this energy, or 1022 keV. Smearing is due to the random (time) occurrence of the second event. This smearing effect places more of the piled up events within any width (511 keV centered) of energy window than would be present if both events occurred at the same time and were random only in energy value.

The bucket Pileup Rejector [16] circuit is designed to identify, by waveshape observation, pileup events and block the transfer of their converted data to the Bucket Controller. Its operation falls into a class of circuits that are invariant of pulse intensity. This circuit operates on the principle that the shape of the output of a linear system is dependent only on input shape and not amplitude, similar to the Constant Fraction Discriminator philosophy of operation. For a system with a bipolar output, the zero-crossing point is dependent only on input waveshape, not amplitude. The design criteria is to locate the point at which this zero-crossing takes place.

The expected output of the Analog Processor integrator is given by:

$$\begin{aligned} E\{y(t)\} &= \int_0^t r_0 e^{(-\frac{a}{\tau})} da \\ &= -r_0 \tau (1 - e^{(-\frac{t}{\tau})}) \end{aligned} \quad (6)$$

where $r_0 \tau = \text{no. of photons converted by PMT}$

$\tau = 300 \text{ nS}$ (decay constant of BGO)

$t = \text{time of sample}$

$a = \text{attenuation constant}$

The expected value of the system output $z(t)$ is obtained by subtracting a delayed version of $E\{y(t)\}$ from an attenuated version of $E\{y(t)\}$ resulting in:

$$E\{z(t)\} = r_0 \tau \left[a(1 - e^{-\frac{t}{\tau}}) - [1 - e^{-\frac{(t-t_d)}{\tau}}] \right] \quad (7)$$

where t_d = time of first sample

t = time of second sample

With an attenuation constant, a , less than one, the response of $z(t)$ goes positive and then reverses and crosses zero, to settle at a final value of $-(1-a)$. The Pileup Rejector examines this response a short time after the predicted crossing and, if the result is negative, indicates that no pileup is present. If a second event occurs during processing of the first, a delayed version of it is added to the first, resulting in a shifting of the zero-crossing. Upon examination of the response, the Pileup Rejector observes a positive signal, and thus labels it a piled up event.

The solution for the attenuation constant is derived such that at the inspection time t , the expected value of $z(t)$ is two standard deviations below zero. This ensures that 98% of the single events are accepted. Selection of the attenuation constant, a , requires a solution for the variance of $z(t)$. This relation is represented by:

$$s^2(t) = r_0 \tau \left[(1-2a)(1 - e^{-\frac{(t-t_d)}{\tau}}) + a^2(1 - e^{-\frac{t}{\tau}}) \right] \quad (8)$$

Solving this equation such that the mean, $E\{z(t)\}$, equals two standard deviations, $2\sqrt{s^2(t)}$,

yields:

$$a^2(k_1^2 - 4k_1) - a(2k_1k_2 - 8k_2) + (k_2^2 - 4k_2) = 0 \quad (9)$$

with $k_1 = r_0\tau(1 - e^{-\frac{t}{\tau}})$

$$k_2 = r_0\tau(1 - e^{-\frac{-(t-t_d)}{\tau}})$$

Solving, by the quadratic equation, for a with $r_0\tau$ equal to 150 photons, t equal to 688 ns, t_d equal to 336 ns, and τ equal to 300 ns, results in an attenuation constant a of 0.6944.

The Pileup Rejector (Figure 4.12) is comprised of the Pileup Rejector PROM (U103) and a feedback register (U104). The first energy sample is processed by the prom and these results are latched by the feedback register. At the end of the integration cycle the second energy sample is passed to the prom. The second-sample data is the same data used by the lookup RAM to make the energy discrimination decision. This value in addition to the results of the first pass are used to make the zero-crossing sign decision.

Time/Position Latches and Request Logic

The time (U89) and position (U94) registers act as holding registers for event data that has been processed by the Position/Energy Processor. These two registers share a common output

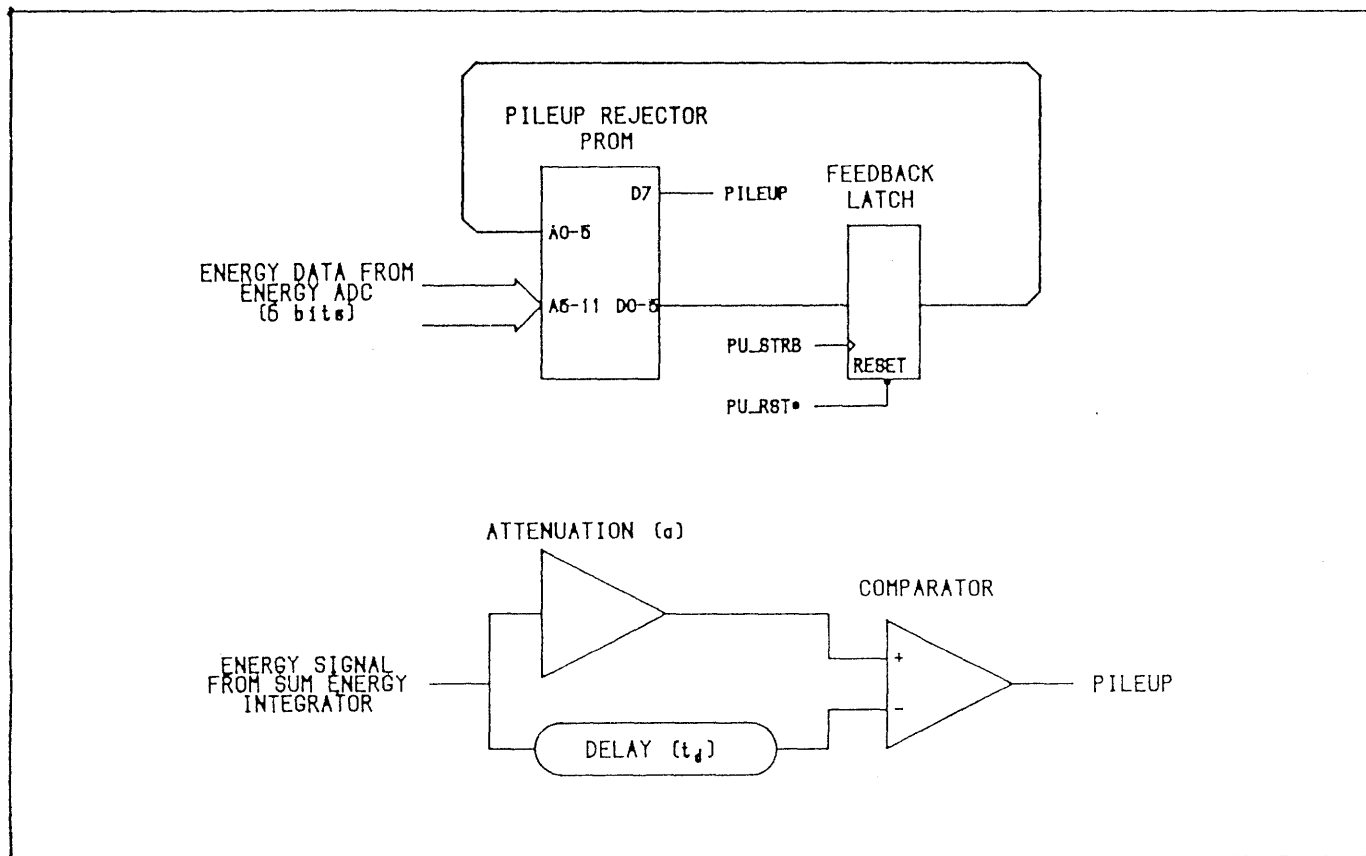


Figure 4.12. Block diagram of the Pileup Rejector.

bus by which data is moved to the time and position pipe registers on the Bucket Controller. When an event is processed by the Position/Energy Processor and is determined to be valid (within the energy discriminator window, not pileup, no ADC overflow), the RQST logic within U96 (pin 22) drives the D of the RQST flip flop high. This value is strobed into the RQST flip flop by the Event Sequencer, thus setting the Q of the RQST flip flop high. Through the mechanism of processing of the event (see Event Sequencer discussion, below), the time data, position data, time correction data, and scatter bit have been set up on the inputs to the time and position registers and get strobed into these devices on the same edge that strobes the RQST flip flop. After this data has been latched into the time and position registers, the Event Sequencer can return to the zero state, awaiting occurrence of the next event.

The RQST logic of U96 looks at many inputs in order to decide whether to summons processing by the Bucket Controller. In addition to looking at the energy discriminator bit (ASD7 of second lookup pass) and the pileup bit from the Pileup Rejector, it also looks at the two-bit time correction value for the one invalid time correction value (note that the Time Correction Circuitry only modifies time by 0 or ± 2 ns). This value is used to reject an event that is located in a low positioning-probability location in position space. The RQST logic also can be preempted by HST_RQST, thus allowing the sequencer to force the D of the RQST flip flop when no event is present. This is a requirement when the microcomputer starts the sequencer by toggling INIT_SEQ with the intention of propagating data out of the Position/Energy Processor. This function is useful for moving data out of the position and time latches during diagnostics.

Within 256 ns of the RQST flip flop being set, the Common Electronics Sequencer begins retrieving data from the time and position latches of the requesting Position/Energy Processor section. The time data is enabled onto the position/time bus PT0-7 and strobed into the time pipe register on the Bucket Controller. Then, the position latch is enabled onto the position/time bus and strobed into the position pipe register on the Bucket Controller. After having retrieved this

data, the Common Electronics Sequencer resets the RQST flip flop by driving RQRST low. A more thorough discussion of the Common Electronics Sequencer can be found in the Bucket Controller discussion (page 80).

The PAL (U96) performs many duties other than just the RQST logic during normal event processing. It is also responsible for transferring microcomputer data from ASD7 into D7 of the time latch and ASD6 and ASD7 into D6 and D7 of the position latch when executing the "data path check". This transfer takes place when the NRM/FR* line (U96-10) is low. When the microcomputer performs a write to the time latch, a decoded strobe appears as FRC_TIM (U96-14) causing the T_STRB (U96-19) to strobe data into the time latch. When the microcomputer performs a write to the position latch, a decoded strobe appears as FRC_PE (U96-16) causing the P_STRB (U96-18) to strobe data into the position latch.

The PAL (U96) routes data through itself differently depending on whether the Event Sequencer is in Run Mode, Position Histogram Mode, or Energy Histogram Mode. In Position Histogram Mode, the lookup RAM is loaded in such a way as to do complete block energy discrimination.

This rudimentary energy discrimination method provides significantly higher quality low-level energy discrimination than that offered by just the CFD threshold. Without this feature, every event that falls above the CFD threshold is histogrammed. Since the CFD threshold is a crude discriminator with a very poor threshold knee, it must be set to a low energy level in order to not reject desired events. The problem with only having this CFD discriminator for rejecting low energy events is that many low energy events will be histogrammed. Since these events present a signal of poor S/N ratio to the ratioing ADCs (X and Y), the position data has a high noise content, which significantly degrades the performance of the position-peak-location algorithm.

Diagnostics/Histogram Mode Buffers

PAL U88 is programmed to act as a tri-state multiplexer allowing transfer of either ASA0-5 or ASD0-6 onto the time bus (TIME0-6). During position histogramming, the X ADC data is routed through this device from ASA0-5 to TIME0-5, where it is loaded into the time register. Via this route, the microcomputer is able to actually acquire the X ADC data. In order for the microcomputer to directly write data to the time latch, U88 is used to direct the microcomputer data from ASD0-6 to TIME0-6, where this data is loaded into the time latch. The operation of writing microcomputer data to the time latch is used in various diagnostic routines.

The bidirectional buffer/register (U90) functions as the position feedback latch during normal Run Mode. In this mode, data flow is from ASD0-5 to ASA6-11. During position histogramming, data from the Y ADC is routed from ASA6-11, through this device, to ASD0-5, from which it is loaded into the position latch. In this mode, U90 acts as a buffer.

Event Sequencer

The Event Sequencer (Figure 4.13) represents the heart of the Position/Energy Processor section. This section of the Position/Energy Processor conducts the Analog Processor integration cycle, the staging of ADC conversion, and the movement of the data from the ADCs on the Analog Processor through the position lookup RAM and out to the time and position latches. It also conducts the routing of the data flow when the Position/Energy Processor is operating in one of the histogram modes. Figures 4.14, 4.15, and 4.16 show the timing waveforms created during execution by the Event Sequencer during Run Mode, Position Histogram Mode, and Energy Histogram Mode, respectively. These different sequencing modes are determined by inputs MODE_8 and MODE_9, which are derived from a latch addressed by the microcomputer.

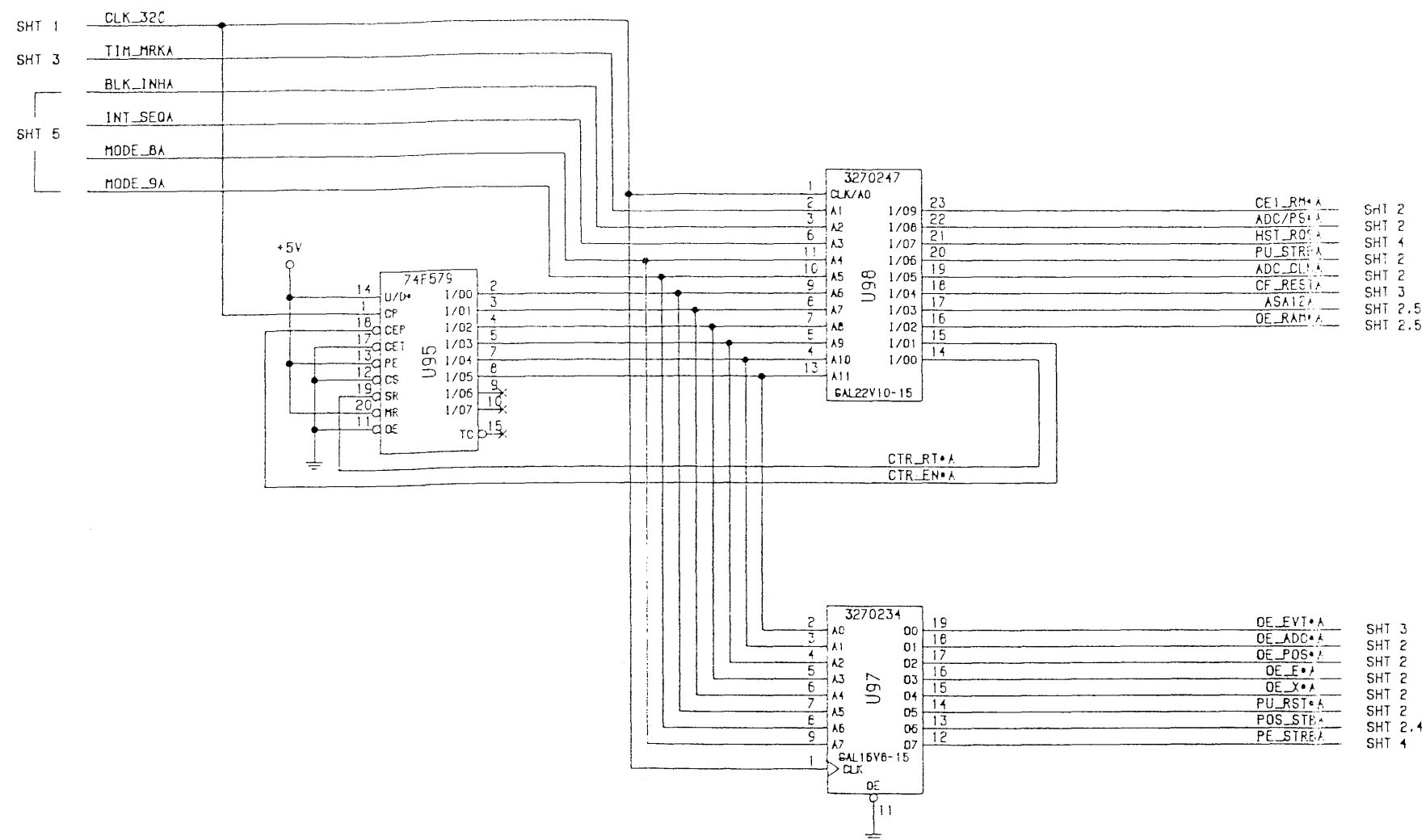


Figure 4.13. Event Sequencer schematic diagram.

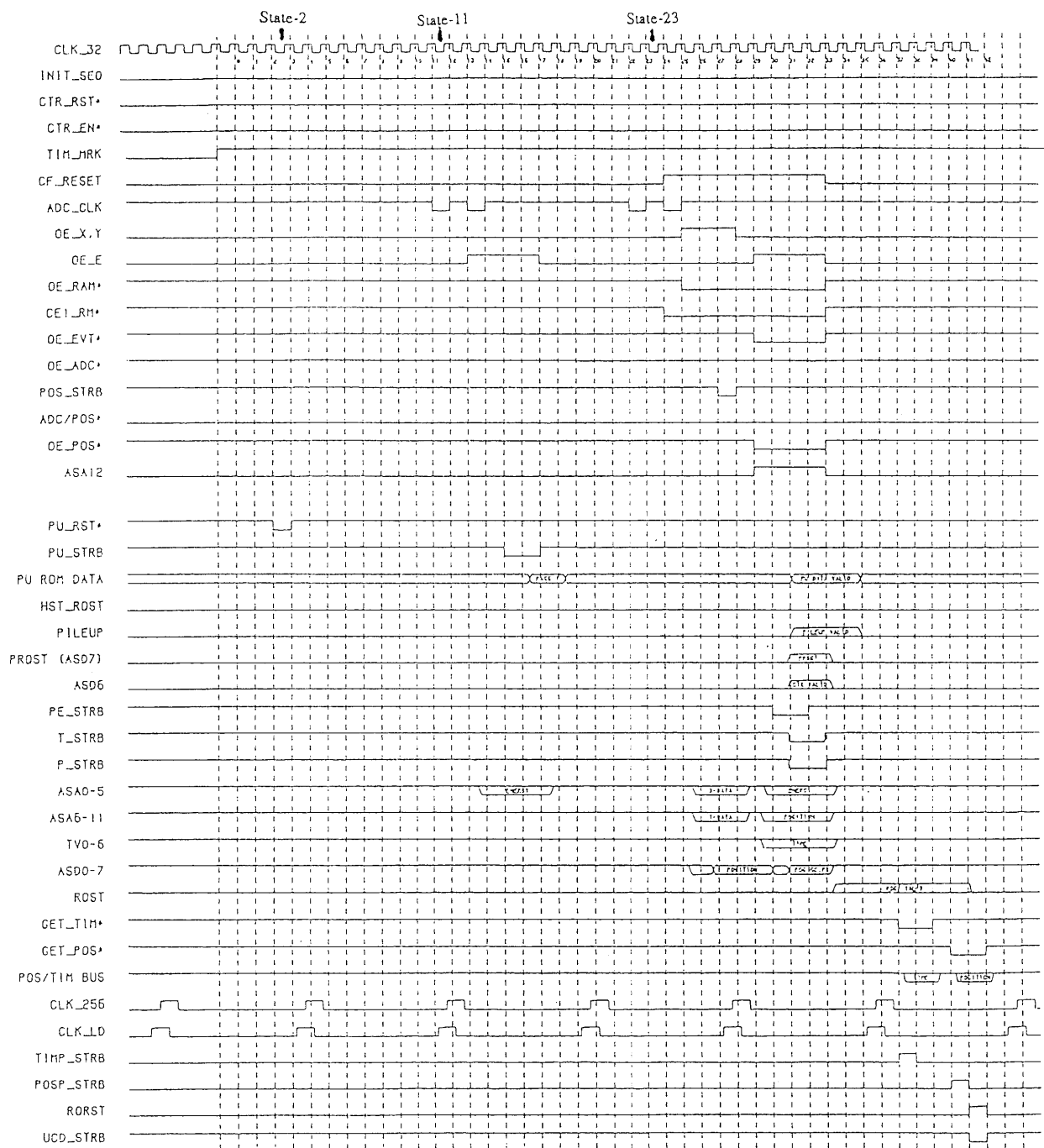


Figure 4.14. Timing waveforms for Event Sequencer in Run Mode.

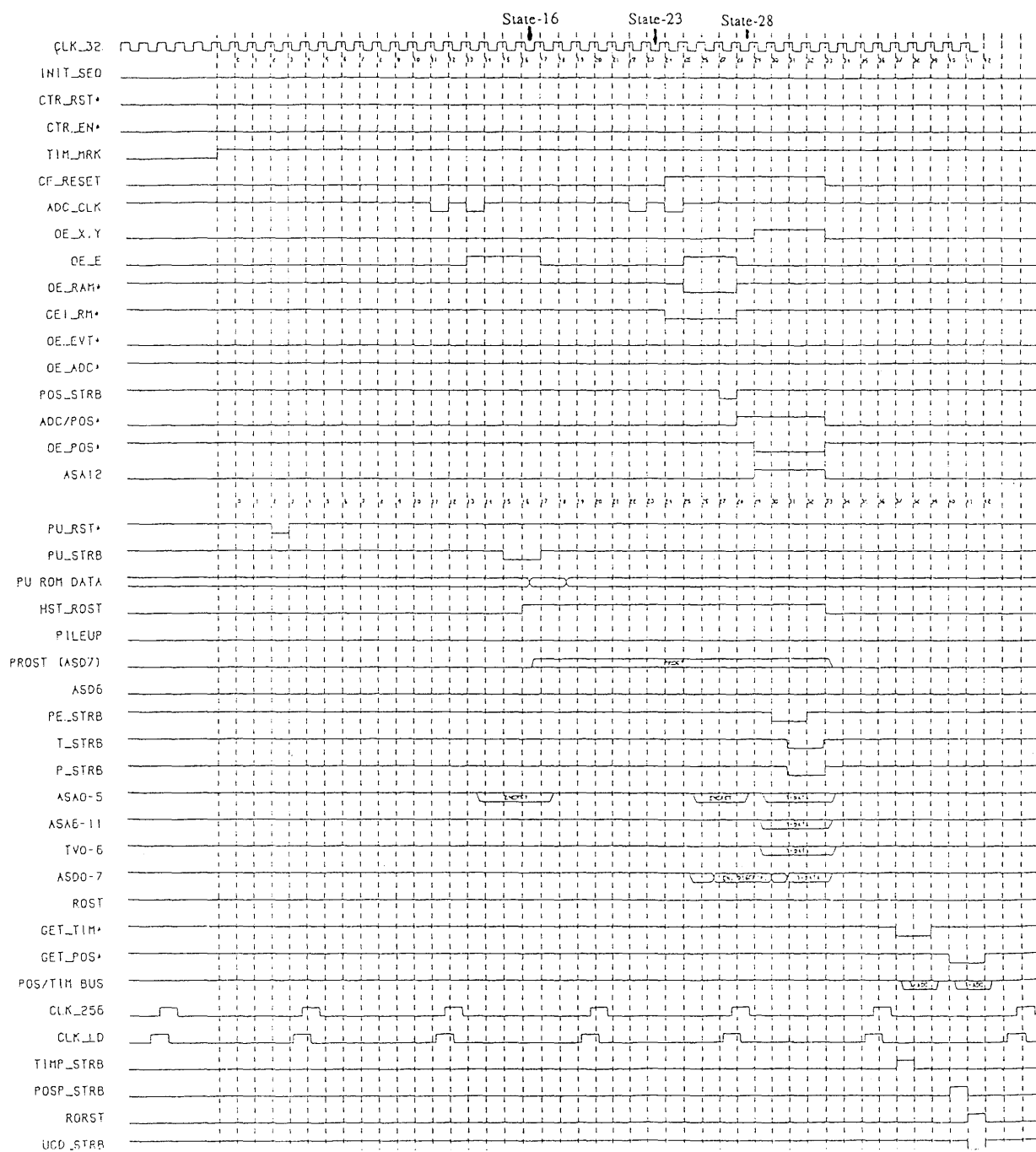


Figure 4.15. Timing waveforms for Event Sequencer in Position Histogram Mode.

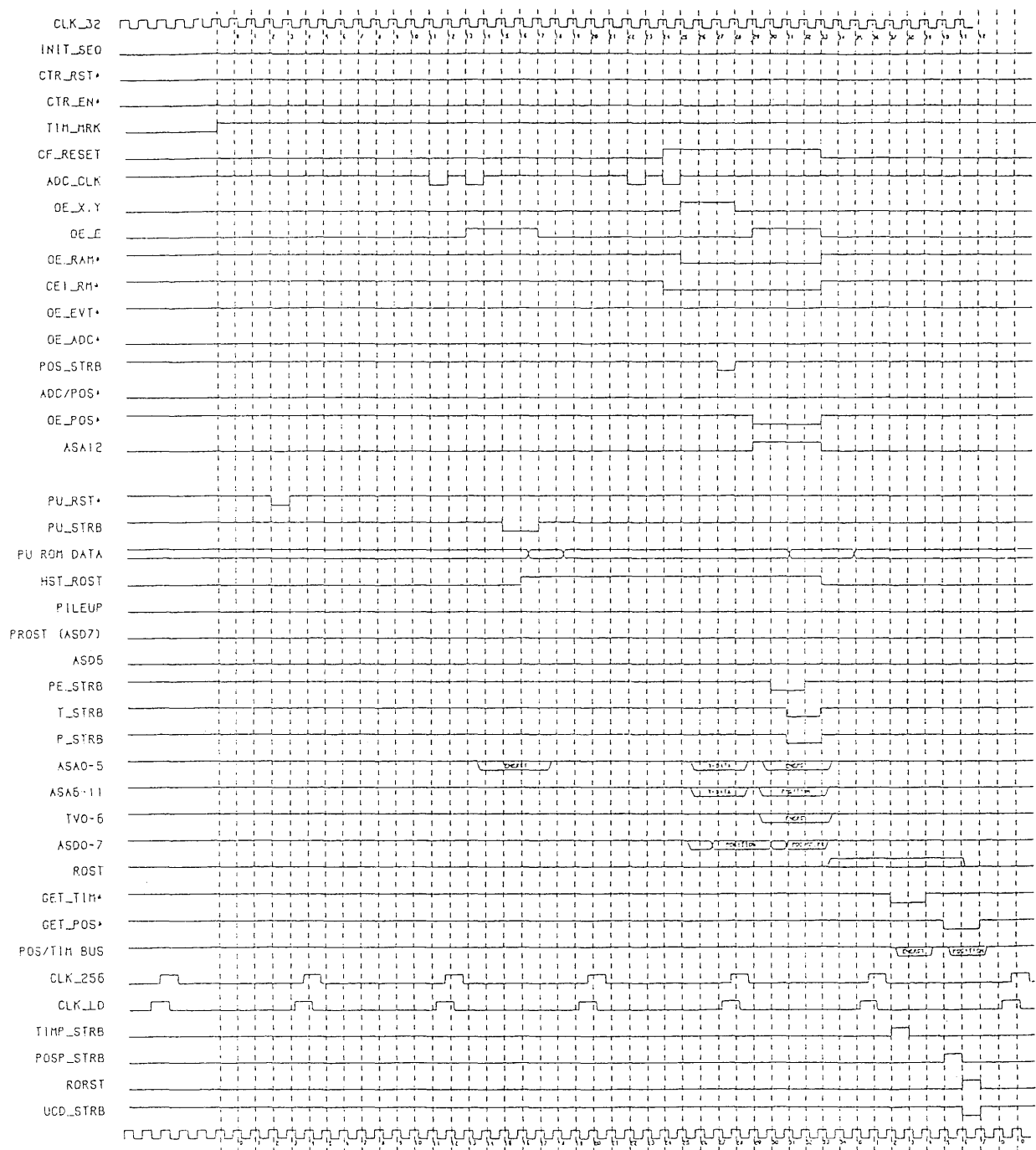


Figure 4.16. Timing waveforms for Event Sequencer in Energy Histogram Mode.

The Event Sequencer, comprised of PAL devices, allows easy modification of the data flow timing (i.e. modification of integration time) by the modification of the PAL code. The eight-bit counter chip produces binary values for each of the different sequencer states. Its modes of operation (start, stop, and hold) are produced by different states of CTR_RT* (reset) and CTR_EN* (enable).

There are two means by which the Event Sequencer begins its sequence: 1) the CFD detects an event to be processed, and 2) by microcomputer control via the INT_SEQ line. This second method is used mostly for diagnostic purposes.

Run Mode

Upon detection of an event, the CFD drives TIM_MRK active, causing the Event Sequencer to initiate processing of the event. The Event Sequencer resets the Pileup Rejector feedback register at state-two and then, at state-11, toggles ADC_CLK to perform the intermediate energy sample used for pileup detection. This data is placed on ASA0-5 by enabling the energy ADC bus driver (OE_E). It is then passed through the Pileup Rejector PROM whose output is latched into U104 on the rising edge of PU_STRB.

At the completion of the integration cycle (state-23), the ADCs are clocked (ADC_CLK) and the integrators are reset (CF_RESET). The lookup RAM chip-enable (CE_RAM*) and output-enable lines (OE_RAM*) are also activated. By not activating the RAM chip enable except during event processing or microcomputer loading/testing, considerable power savings occurs. X and Y ADC data are enabled on ASA0-11 by OE_X,Y. At this time, the Event Sequencer is driving ASA12 low. After the propagation delay of the lookup RAM and associated transceiver (U92), crystal position and time correction data appear on ASD0-7. The crystal number is strobed

into the feedback latch U90 while the time correction bits are strobed into the internal registers of PAL U96 by POS_STRB. This completes the first pass of position lookup.

Next, the Event Sequencer drives ASA12 high and enables the feedback latch containing the crystal position onto ASA6-11. The energy ADC is enabled onto ASA0-5 completing the setup of lookup address for the energy discrimination pass. Following the RAM and transceiver propagation delay, the energy discriminator bit and the scatter discriminator bit appear on ASD7 and ASD6, respectively. The Pileup Rejector uses this second energy sample to determine whether the wave shape is that of one event or many (pileup). The energy discriminator bit (ASD7) is routed to PAL U96 where it is used along with the Pileup Rejector signal (PILEUP), the Pileup Rejector enable bit (PU_EN), and the three ADC overflow bits (EN_O.F., X_O.F., and Y_O.F.) to determine whether the processed event is valid and should be passed along to the Bucket Controller. The output from this decision, U96-22, is transferred to the D input of the RQST flip flop where it is strobed in on the rising edge of P_STRB (U96-18). P_STRB follows PE_STRB in this mode.

At the end of the energy discrimination pass, crystal position appears on ASD0-5, as it did during the first position lookup pass. This data is strobed into the position latch U94 on the rising edge of P_STRB. Bit D6 of the position latch is loaded with ASD6 (scatter bit) via PAL U96, while bit D7 of the position latch is loaded with the least significant time correction bit that was stored in the internal register of U96 after the position lookup pass.

Time digitizer data, which is available no later than 256 ns after the detection of the event, is strobed into its holding register by the same rising edge of PU_STRB that latched the data from the Pileup Rejector PROM (after the intermediate energy sample). Choice of this signal for clocking time data into the Time Digitizer's latch was a matter of convenience. The only constraint on this choice was that the edge occur later than 256 ns after the start of event processing in order to ensure completion of the coarse timing in the Time Digitizer. At the same

time that the energy discrimination pass is being made through the lookup ram, the Time Digitizer holding register is enabled onto the time bus TIME0-6. This data is strobed into the time register U89 on the rising edge of T_STRB (U96-19). T_STRB follows PE_STRB in this mode. Bit D7 of the time register is loaded with the most significant time correction bit that was stored in the internal register of U96 after the position lookup pass.

Position Histogram Mode

The Event Sequencer operation during Position Histogram Mode is very similar to that of Run Mode. In this mode X and Y ADC data is transferred to the time and position latches where they will be conveyed to the Bucket Controller. Via latches on the Bucket Controller, the microcomputer can acquire and histogram X and Y ADC data.

There are two significant differences in Position Histogram Mode operation versus Run Mode operation. First, the raw X and Y data must be passed, unmodified, to the time and position latches where it can be accessed by the microcomputer. Second, the lookup RAM must be able to observe the energy ADC data so as to make an energy discrimination decision on each event.

The Event Sequencer is placed in Position Histogram Mode via the levels placed on the two inputs, MODE_8 and MODE_9. If an event is detected by the CFD while in this mode, the Event Sequencer will begin stepping through the states in a similar fashion to Run Mode. Many of the non-critical Event Sequencer output waveforms duplicate their Run Mode action in an effort to save product terms in the PALs. At the end of the integration period (state-23), the RAM chip-enable (CE_RAM*) and output-enable (OE_RAM*) are enabled in preparation for performing the energy discrimination. The energy ADC is enabled onto ASA0-5, the Event Sequencer drives ASA12 low, and the feedback latch, U90, ensures that ASA6-11 are driven to some arbitrary

value. The RAM is loaded with values in order to perform the discrimination function, which are repeated for all possible address ranges presented by ASA6-11. After the RAM propagation delay, ASD7 will represent the result of the energy discrimination operation. This value is latched into an internal register of the PAL U96 for later transfer to the position latch. The microcomputer looks at this bit in the position byte to decide whether the event passed the discrimination requirements and should be histogrammed.

At state-28 the RAM is disabled and then the X and Y ADCs are enabled (OE_X) onto ASA0-11. The feedback latch U90 is now reversed to act as a buffer driving the Y ADC data from ASA6-11 onto ASD0-5. PAL U88 has been enabled to direct X ADC data onto the time bus TIME0-5. HST_RQS, which went active at state-16, forces (through PAL U96) the D of the RQST flip flop active. Finally, PE_STRB strobes the data into the time latch, position latch, and RQST flip flop. PE_STRB gets transferred through PAL U96 to become T_STRB (U96-19) and P_STRB (U96-18). From here the data gets processed by the Bucket Controller as though it were truly time and position data. It is only through this Position Histogram Mode that the microcomputer can access the raw ADC data.

Energy Histogram Mode

Energy Histogram Mode is similar to Position Histogram Mode in that raw ADC data is passed directly to the microcomputer without being processed by the Position/Energy Processor. This mode is used to create the energy histograms for each of the crystals. Because of the requirement of energy histograms for each crystal, the Position/Energy Processor electronics must pass X and Y data through the loaded lookup RAM for crystal positioning while the raw energy ADC data is passed directly to the microcomputer via the time latch.

Execution of Energy Histogram Mode is almost identical to Run Mode with the exception that the energy discriminator pass results are ignored (HST_RQST from the Event Sequencer generates active D on RQST flip flop) and the energy ADC data on ASA0-5 is placed on the time bus TIME0-5 via PAL U88. The time digitizer holding register is not enabled onto the time bus in Energy Histogram Mode. Finally PE_STRB strobes the data into the time (containing energy data) latch and position (containing position data) latch.

Microcomputer Interface

The Microcomputer Interface (Figure 4.17) is the connection of the microcomputer to the Position/Energy Processor section. It provides the microcomputer access to the lookup RAM, the Analog Processor DACs, the mode latch (U100), and the time and position latches.

The Microcomputer Interface also provides a very useful scanner diagnostic tool, the generation of Pseudo Activity data. This feature allows each Position/Energy Processor section to produce pseudo "events", producing time and position data from data stored within the lookup RAM. This offers the ability to test the entire data channel from bucket to sinogram without activity being present. It also allows the bucket to produce extremely high apparent singles rates, a valuable tool in assessing Coincidence Processor and Real Time Sorter performance at extreme count rates.

Processor Interface

The fundamental function of the Microcomputer Interface is to map the components of the Position/Energy Processor section that must be accessed by the microcomputer into its memory map. Mapping of these devices is shown in Table 4.1. Each Position/Energy Processor section

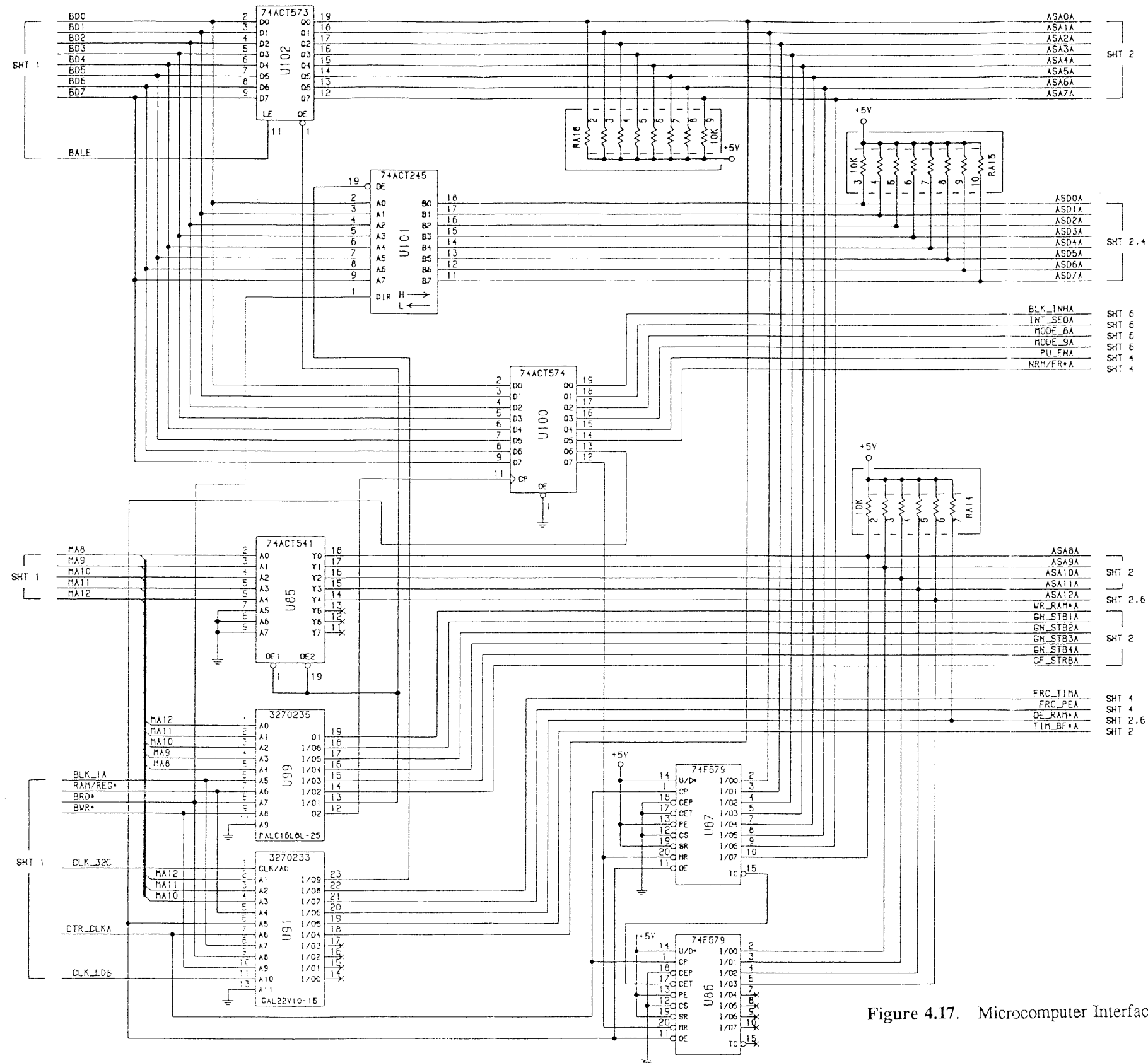


Figure 4.17. Microcomputer Interface schematic.

has its own Microcomputer Interface with the same logical addresses for the devices within the block. The Bucket Controller individually pages one of the four Position/Energy Processor sections into the eight-kilobyte physical block.

The buffered multiplexed data bus (BD0-7), the address lines MA8-12, the buffered read (BRD*) and write (BWR*) lines, and the buffered address latch signal (BALE) are routed to the Microcomputer Interface. The low-order address lines are stripped from the multiplexed data bus by latch U102. Transceiver U101 buffers the Position/Energy Processor section data bus (ASD0-7)

Table 4.1. Position/Energy Processor Microcomputer Interface Memory Map.

RAM/REG* (uC PORT P1.3) = 0	
Logical Address	Description
0100h	Gain DAC for PMT 0 (Write Only)
0200h	Gain DAC for PMT 1 (Write Only)
0300h	Gain DAC for PMT 2 (Write Only)
0400h	Gain DAC for PMT 3 (Write Only)
0500h	CFD Threshold DAC (Write Only)
1000h	Time Latch (Write Only)
1400h	Position Latch (Write Only)
1800h	ASMODE Latch (Write Only)
RAM/REG* (uC PORT P1.3) = 1	
Logical Address	Description
0000h - 1FFFh	Position/Energy Processor Lookup RAM

from the microcomputer data bus, while buffer U85 provides similar isolation of the higher address lines. Latch U100 is a control register used to provide such services as Event Sequencer mode control, inhibiting of a block, and enable/disable of the Pileup Rejector and Pseudo Activity circuitry. PAL U99 provides address decoding for the lookup RAM and the Analog Processor DAC registers, while PAL U91 provides additional address decoding as well as functioning as the sequencer in the Pseudo Activity function.

Pseudo Activity Generator

The Pseudo Activity Generator provides a means of allowing the bucket to produce "event" data without activity being present. Since this data is produced from data stored in the lookup RAM, data patterns can easily be modified. The rate at which this data is output from the bucket is adjustable over a very wide range, allowing high count rate simulation.

In order for each of the Position/Energy Processor sections not to interfere with its other three neighbors, a sequencer on the Bucket Controller produces four signals, CTR_CLKA, CTR_CLKB, CTR_CLKC, and CTR_CLKD as shown in Figure 4.18. The rate at which these signals are produced is governed by the divisor value loaded into the prescaler (counter 1 of triple timer/counter U55). This value can be any integer from one to 65,535, and represents the number of CLK_256 (256 ns) intervals that must transpire before the output of the prescaler transitions. Output CTR_CLKA is routed to PAL U91 and address counters U86 and U87. These two counters are enabled onto address lines ASA1-12 whenever Pseudo Activity mode is activated.

Each time CTR_CLKA goes active, the counters (cascaded to produce a 12-bit counter) increment the lookup RAM by two (ASA0 controlled by the sequencer within PAL U91). PAL U91, which functions as a sequencer when in Pseudo Activity mode, begins execution with ASA0 low. Data from the even address of the lookup RAM is driven onto ASD0-7, transferred through

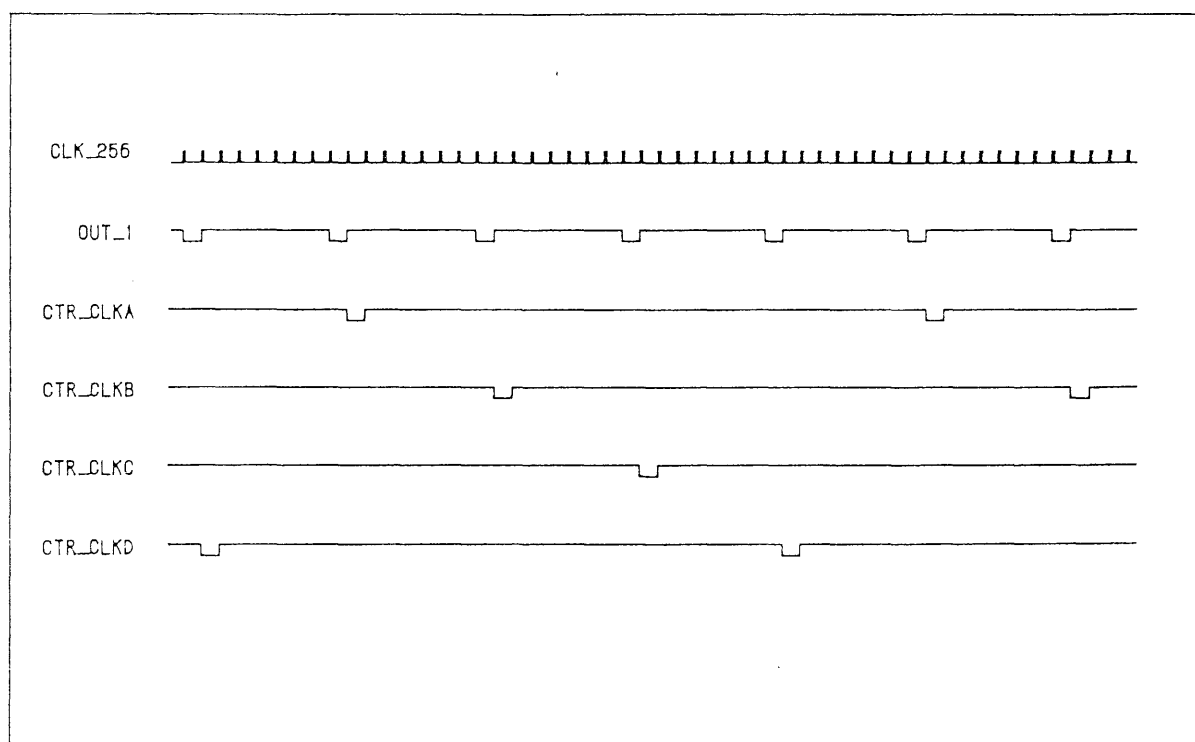


Figure 4.18. Timing diagram of CTR_CLK Generator.

PAL U88 to the time bus TIME0-6 from where it is strobed into the time latch by FRC_TIM. Line ASD7 is transferred to the time latch through PAL U96. The sequencer then drives ASA0 high and the lookup RAM data from the odd address is driven onto ASD0-7, where it is strobed into the position latch by FRC_PE. ASD6 and ASD7 are routed to the position latch through U96. The most significant bit of this data must be set high as it is also routed to the D of the RQST flip flop and must be set in order to get the attention of the Common Electronics sequencer. Upon completion of this sequence, the sequencer returns to the resting state, awaiting the next CTR_CLKA activation.

Bucket Controller Board

The Bucket Controller is the common interface and arbiter (Figures 4.19 and 4.20) for the event data from the four block processing channels of the Position/Energy Processor. Data from any of the four Position/Energy Processor sections is sequenced into the Bucket Controller, where it is transferred to the high speed time and position shift registers. The microcomputer that orchestrates the bucket functions is also located on the Bucket Controller board.

High Speed Data Path

Figure 4.21 shows the data path circuitry which processes the data from the active Position/Energy Processor section. Data from the Position/Energy Processor section is latched into the time and position pipe latches, U18 and U17, respectively.

The seven bits of time information are routed to the time correction adders, U31 and U29, where this value can be incremented or decremented by one or left unmodified. From here the

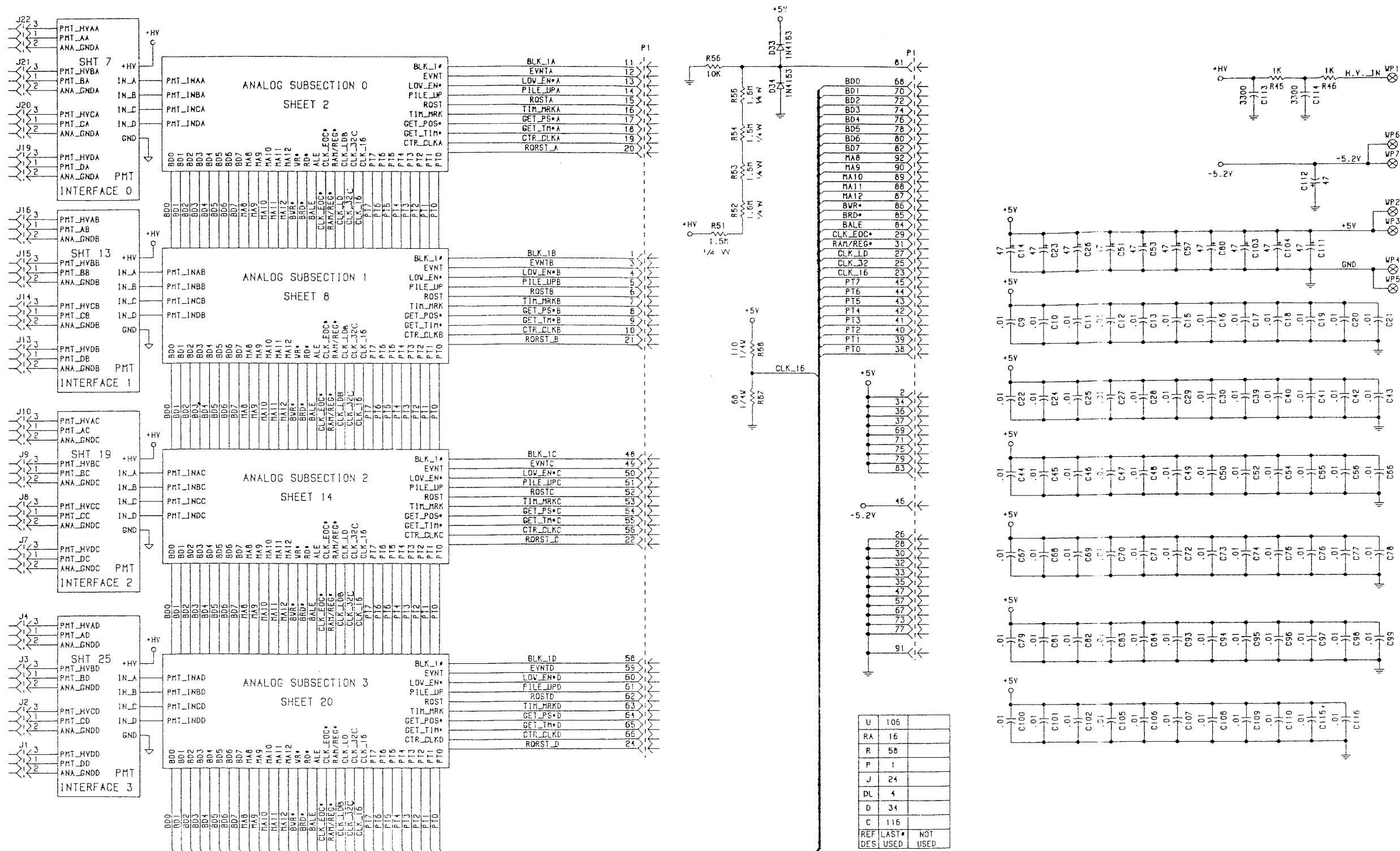
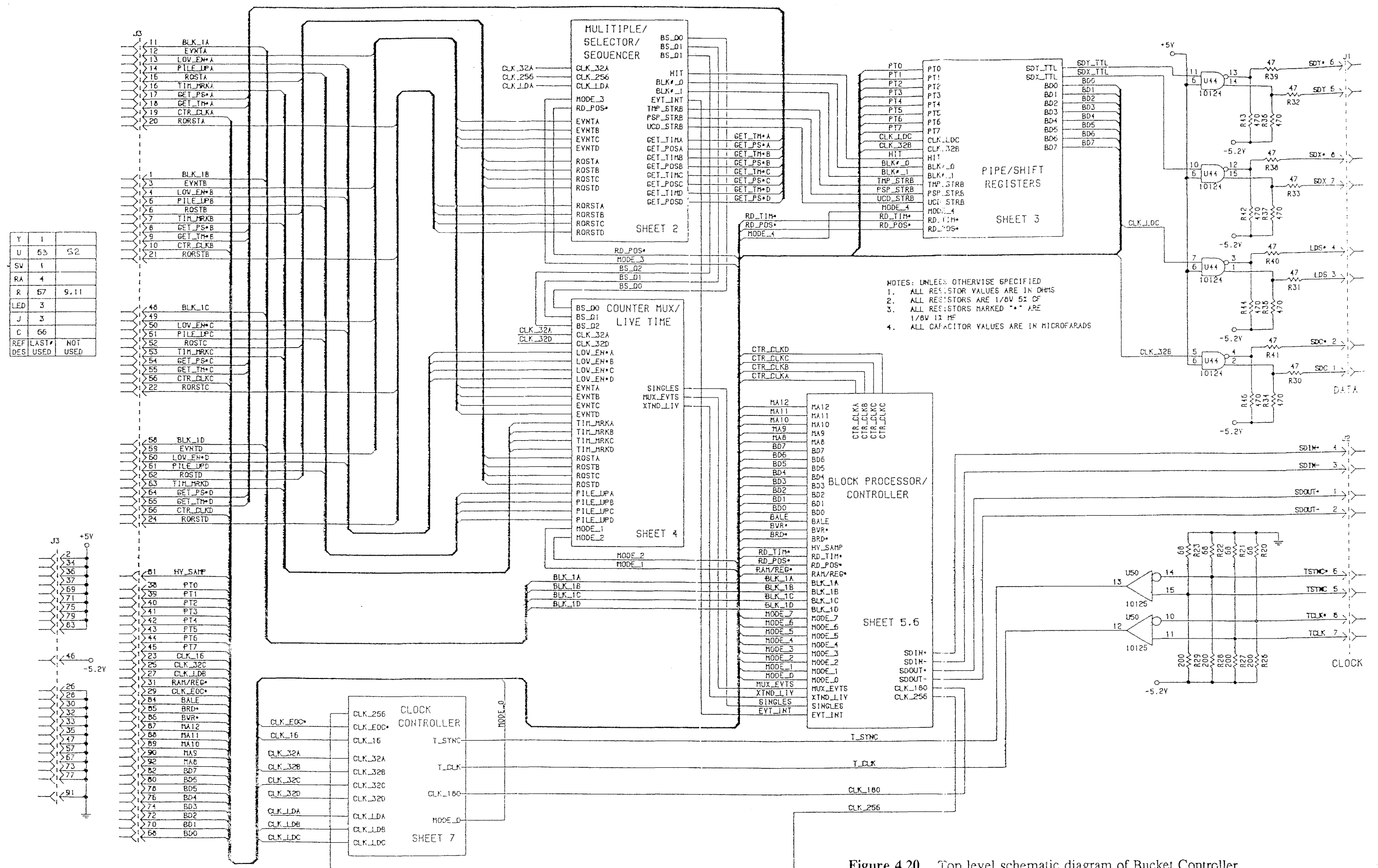


Figure 4.19. Schematic diagram of the four Position/Energy Processor sections.



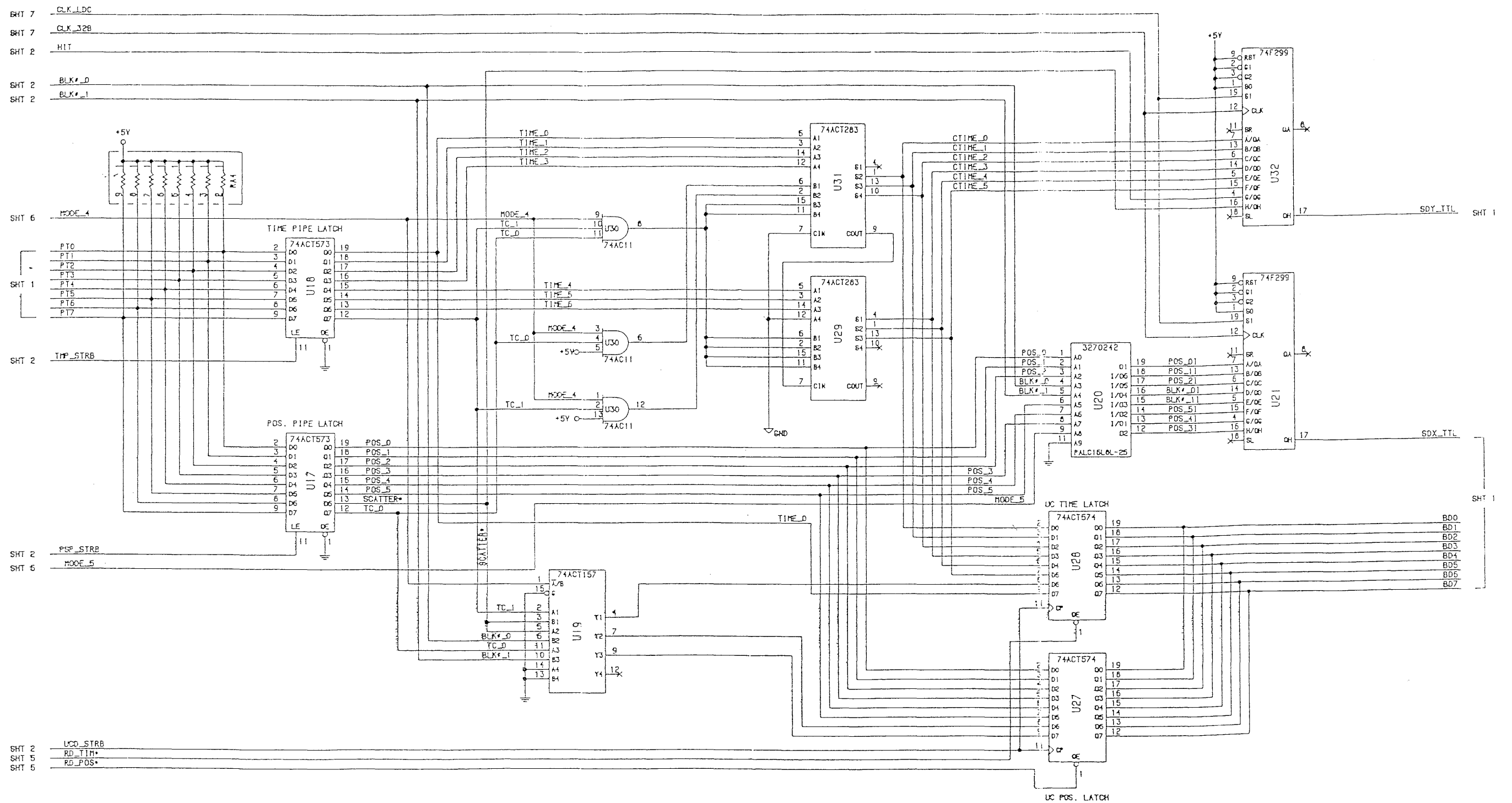


Figure 4.21. Bucket Controller time/position data path electronics schematic.

corrected time data proceeds to the time shift register for transmission to the Coincidence Processor.

The six bits of position information are transferred to PAL U20, where this data, along with two bits of block data (indicating which block of the bucket) from the arbiter, are either passed straight through or rotated end-to-end and then passed through to the position shift register. This byte rotation, controlled by the microcomputer, is required in order to properly orient the position bits for the Coincidence Processor depending on the orientation of the bucket (Ring 1 or Ring 2). From here the position data is transmitted to the Coincidence Processor. Figure 4.22 helps to clarify the position byte shift and the need for it.

Common Electronics Sequencer and Arbiter

The Common Electronics Sequencer (Figure 4.23) is responsible for staging the movement of data from the requesting Position/Energy Processor section, through the Bucket Controller data path, to the time and position shift registers. This section is also responsible for interrupting the microcomputer when new data is present in the time and position capture registers. This process is required when histogramming data.

PAL U10 is the nucleus of the Common Electronics Sequencer. It samples the four EVNT lines from the different Position/Energy Processor sections. The active EVNT flip flop becomes valid at the end of event processing by the specific Position/Energy Processor section, whether the event is valid (RQST would get set too) or not valid (failed energy discrimination, pileup, etc.). As shown in Figure 4.24, the Common Electronics Sequencer drives the GET_TIM (U10-18) active, placing the appropriate Position/Energy Processor section's time latch data onto the time/position bus PT0-7. This data is then strobed into the time pipe latch, U18, and the Common Electronics Sequencer then disables the Position/Energy Processor section's time latch

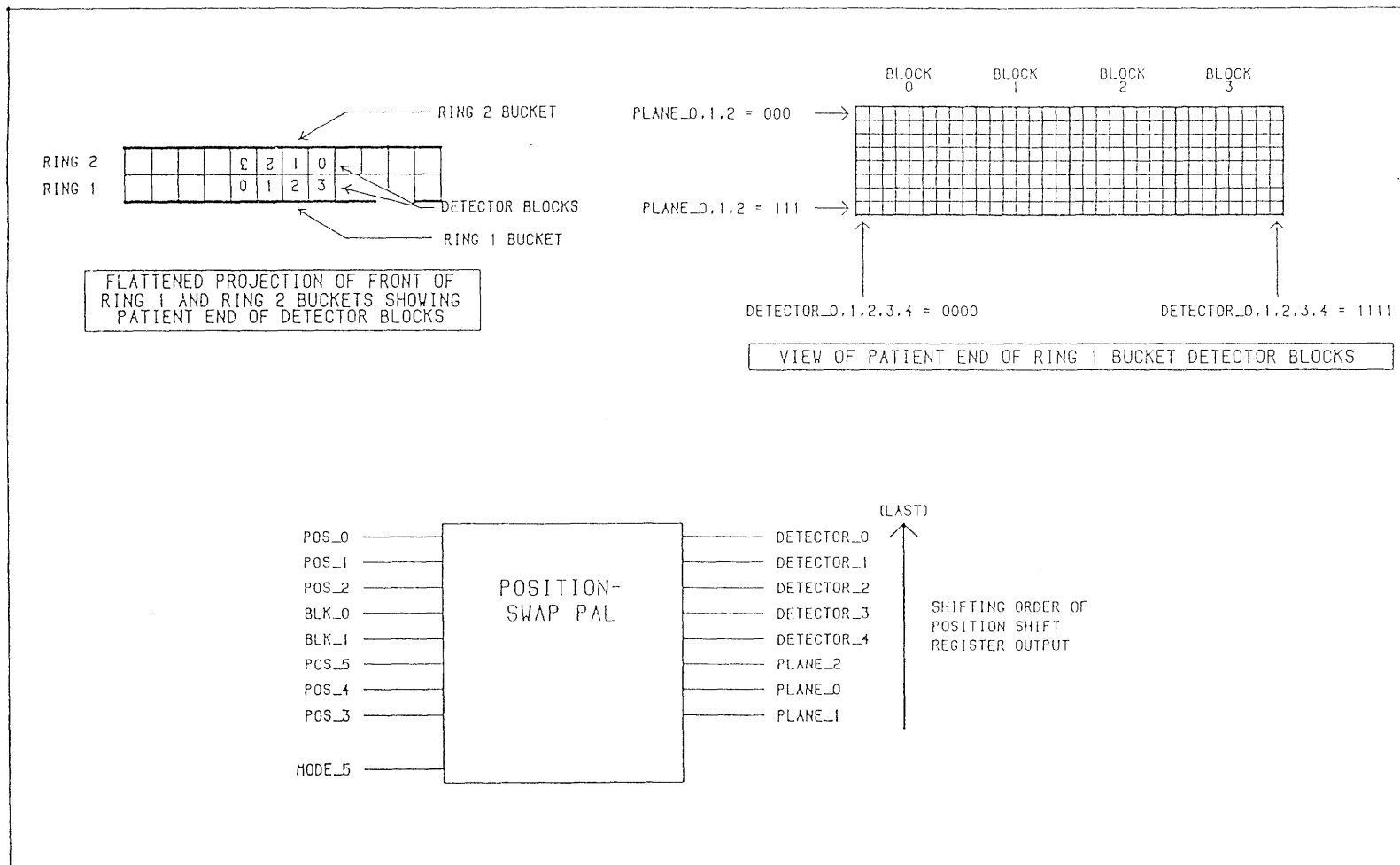


Figure 4.22. Position Swap PAL functionality.

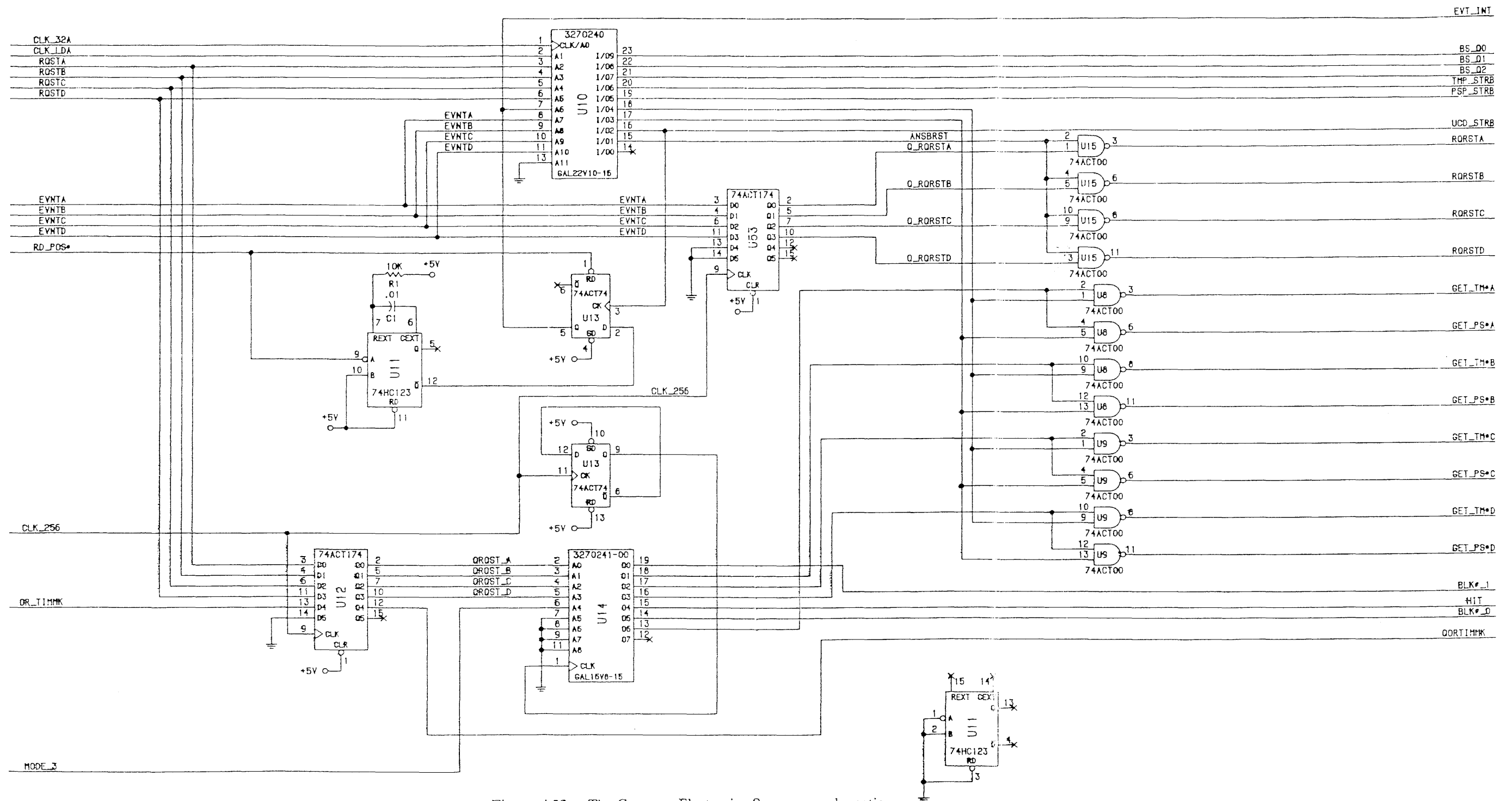


Figure 4.23. The Common Electronics Sequencer schematic.

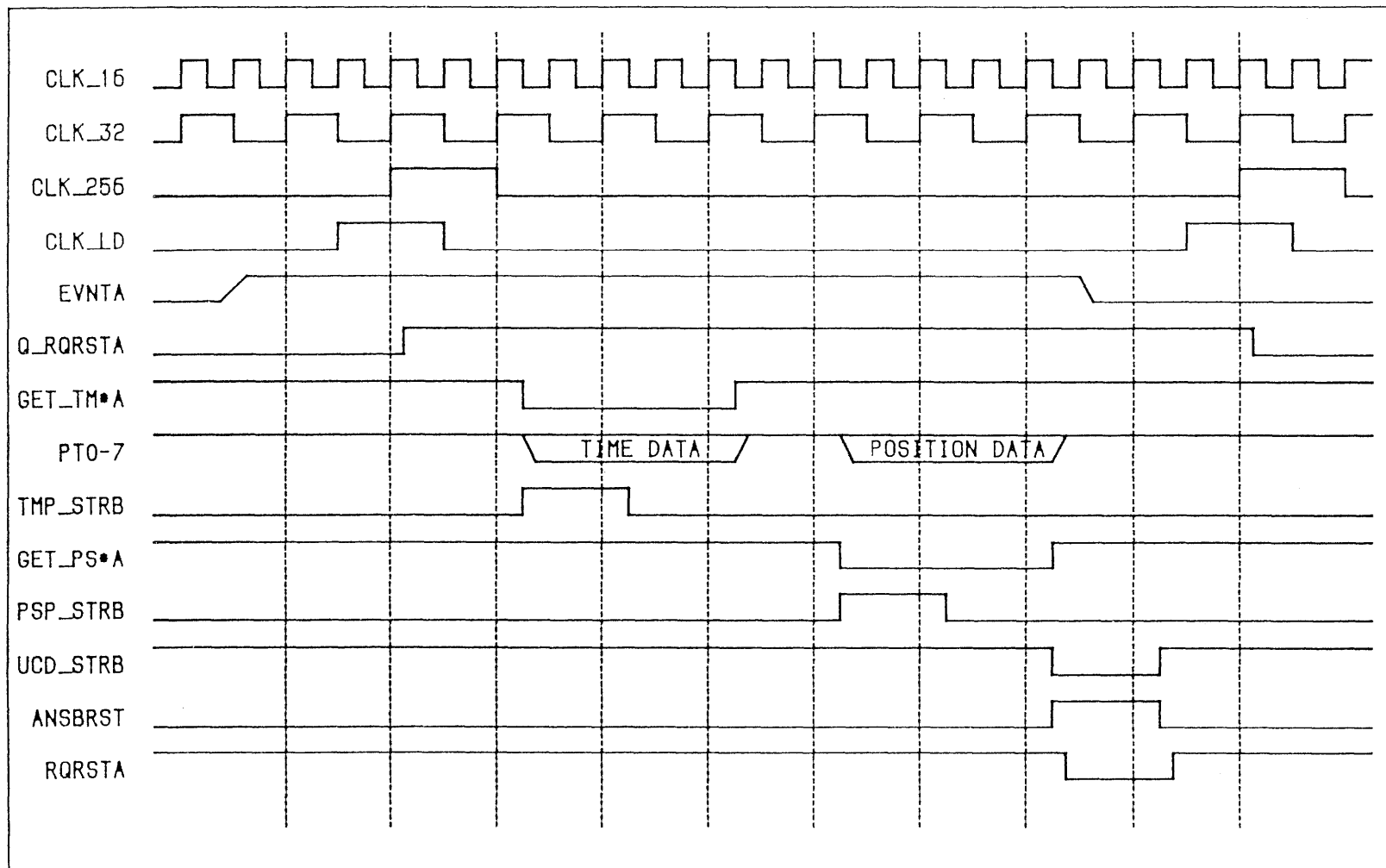


Figure 4.24. Timing diagram of Common Electronics Sequencer.

and enables its position latch. The Common Electronics Sequencer then latches this data into the position pipe latch, resets the appropriate Position/Energy Processor section's RQST flip flop, interrupts the microcomputer, and returns to its zero state, awaiting another event.

Synchronization of this processing to the 256 ns interval occurs by delaying the starting of the Common Electronics Sequencer until the CLK_32 transition when CLK_LD is active. This ensures that data will propagate through the Bucket Controller data path, arriving in time to be set up for the time and position shift registers. By this process, the asynchronous occurrence of an event gets synchronized to the scanner master clock.

PAL U14 performs the arbiting function of the Common Electronics Sequencer. This logic monitors the different Position/Energy Processor section's RQST lines to determine within which block the event occurred. Upon observing a Position/Energy Processor section RQST, this PAL drives the appropriate line (U14-16, 17, 18, or 19) active in order to select this section's time and position latch during execution by the Common Electronics Sequencer. This selection logic (U8 and U9) combines the active line from the arbiter with the GET_POS and GET_TIM lines from the sequencer to derive the output enable signals for the time and position latches.

The arbiter validates an event by setting the HIT bit (U14-15) high. This bit is embedded into the seventh bit of the time byte to be serialized by the time shift register. The Coincidence Processor samples this HIT bit to determine which buckets have valid data to be processed.

When observing requests from two different Position/Energy Processor sections during the same 256 ns interval, the arbiter passes data from one of the two, discarding the data from the other. The register U12 is required to sample the four Position/Energy Processor section's RQST lines only at the beginning of the Common Electronics Sequencer sequence, preventing errors due to the possibility of an asynchronous occurrence of another section's RQST during this sequence. The arbiter uses the state of flip flop U13b, toggled by CLK_256 so as not to transition during this process, to randomize the selection of which block to pass.

If three or four Position/Energy Processor sections present a RQST during the same 256 ns interval, the arbiter simply discards all events. Over the count rates present during normal operation, losses from this rejection are minimal, due to the extremely low probability of three or four blocks going active within the same 256 ns interval. This rejection is performed by leaving the HIT bit low during this interval.

Interrupt Throttling Circuitry

One-shot U11 and flip flop U13a form an interrupt throttling circuit to allow the microcomputer time to perform other activities when histogramming events at high count rates. Without this circuitry, at high count rates the microcomputer does not have time to complete and exit the interrupt routine that performs histogramming before another event occurs, queuing up another interrupt. The effect of this queuing of interrupts is that the microcomputer never exits the histogramming routine to execute any other code.

The microcomputer interrupt generated by the Common Electronics Sequencer is transferred from the sequencer PAL (U10) through flip flop U13a. This flip flop can only generate an interrupt (U13a-5 high) if one-shot U11 has timed out. This one-shot is triggered by the microcomputer performing a read of the position capture latch, during execution of the interrupt routine. Until the one-shot times out, events processed by the bucket are passed to the shift registers (U32 and U21) but do not generate an interrupt of the microcomputer.

Time Correction Circuitry

The Time Correction Circuitry takes the seven-bit time information acquired by the Position/Energy Processor section Time Digitizer and adds -1 (-2 ns), 0 (0 ns), or 1 (+2 ns) to

the value. This circuitry can be disabled by the microcomputer, causing the circuitry to add 0 to all events regardless of the time correction value supplied with the event data.

Upon acquiring an event's time and position data, the Common Electronics Sequencer strips the two time correction bits from the time and position bytes and moves these through the subtract/add logic (U30) to the adder (U31 and U29). The least significant bit of the seven-bit sum is concatenated and the resulting six-bit (four ns resolution) value is set up on the time shift register inputs.

Time and Position Capture Latches/High Speed Shift Registers

The final phase of processing that the bucket performs on the data is to convert the parallel time and position bytes into the serial string that is transmitted through the ECL line drivers (U44) to the Coincidence Processor. Figure 4.24, on page 85, represents the timing diagram of the data flow into these shift registers. Synchronization of the shifting process begins on the CLK_32 transition when CLK_LD is high. On this edge, the bit present on the H input (U32/21-16) appears, followed by the bit on the G input (U32/21-4) on the next rising edge of CLK_32, etc. Data present on the inputs to the shift registers is continuously shifted out, with true events always being validated by the presence of the HIT bit.

Data presented to the time and position shift registers is also strobed into the time and position capture registers by the Common Electronics Sequencer. Some rearrangement of the data is performed to allow acquisition of the time correction bits and the least significant bit of the Time Digitizer data. This information is available with the Time Correction Circuitry disabled via multiplexer U19. By transferring these bits of information to the time and position capture latch, all eight bits of both the time and position pipe latches are available to the microcomputer.

This allows very thorough testing coverage of both the Position/Energy Processor and the Bucket Controller data paths during execution of diagnostics.

Event Counting Circuitry

In order to have accurate quantitation in PET, deadtime of the processing electronics must be taken into account. The method used is to monitor, in real time, the uncorrected and corrected count rates at each of the buckets. Uncorrected count rate is the rate at which valid events are observed by the bucket processing electronics. Corrected count rate is the rate at which events are truly impinging on the bucket detectors, whether they get processed (electronics: live) or not (electronics: dead).

Counting uncorrected singles simply requires counting each time the HIT bit goes active. This function is implemented within U26, where the four RQST bits from the Position/Energy Processor sections are ORed together. The output of this logic is routed through U16 to the D of flip flop U7. Every processed event produces a transition of U7 which increments the 16-bit internal counter of the microcomputer U33.

Calculation of corrected singles is performed by placing U26 in a different "mode" where count rate calculation is performed using the relation that:

$$\lambda = \frac{1}{t_{event} - t_0} \quad (10)$$

where

λ = mean corrected countrate

t_{event} = time that electronics goes dead

t_0 = any arbitrary starting time

The above equation holds for processes that occur randomly in time, which is a valid assumption here.

The method of implementing this calculation requires counting the number of times the electronics goes live (n), with a binary counter. This counter is sampled at an interval proportional to the amount of time that the electronics is live, $n(t_{event}-t_0)$.

Figure 4.25 illustrates a block diagram of the circuitry to calculate corrected singles. The implementation of this circuit is shown in Figure 4.26. The gated modula-n divider measures actual time that the gate signal is active, thus accumulating each $(t_{event}-t_0)$ interval. This divider produces an output when the internal counter reaches a preset value, $n(t_{event}-t_0)$. The live counter counts the number of times that the electronics goes active (live). Using the output from the modula-n divider to sample the live counter, produces:

$$\text{Counter Capture Value} = \frac{n}{n(t_{event}-t_0)} \quad (11)$$

where $n = \text{no. of } (t_{event}-t_0) \text{ intervals} = \text{no. of live occurrences}$

Canceling n from top and bottom produces the desired corrected count rate value.

PAL U25 and the 16-bit counter within the microcomputer, perform the counting of live occurrences. PAL U38 and timer/counter U5 perform the modula-n divider function. The output from the modula-n divider is applied to the timer capture input to the microcomputer. Transitions on this line cause a snapshot of the 16-bit counter to be transferred to a holding register and an interrupt of the processor is produced.

PALs U26 and U38 also function as multiplexers, allowing the counting of other signals within the bucket. Depending on the value of MODE_1 and MODE_2, MUX_EVTS provides a pulse for every CFD event, Scatter event, or Pileup event. This output (MUX_EVTS) is routed

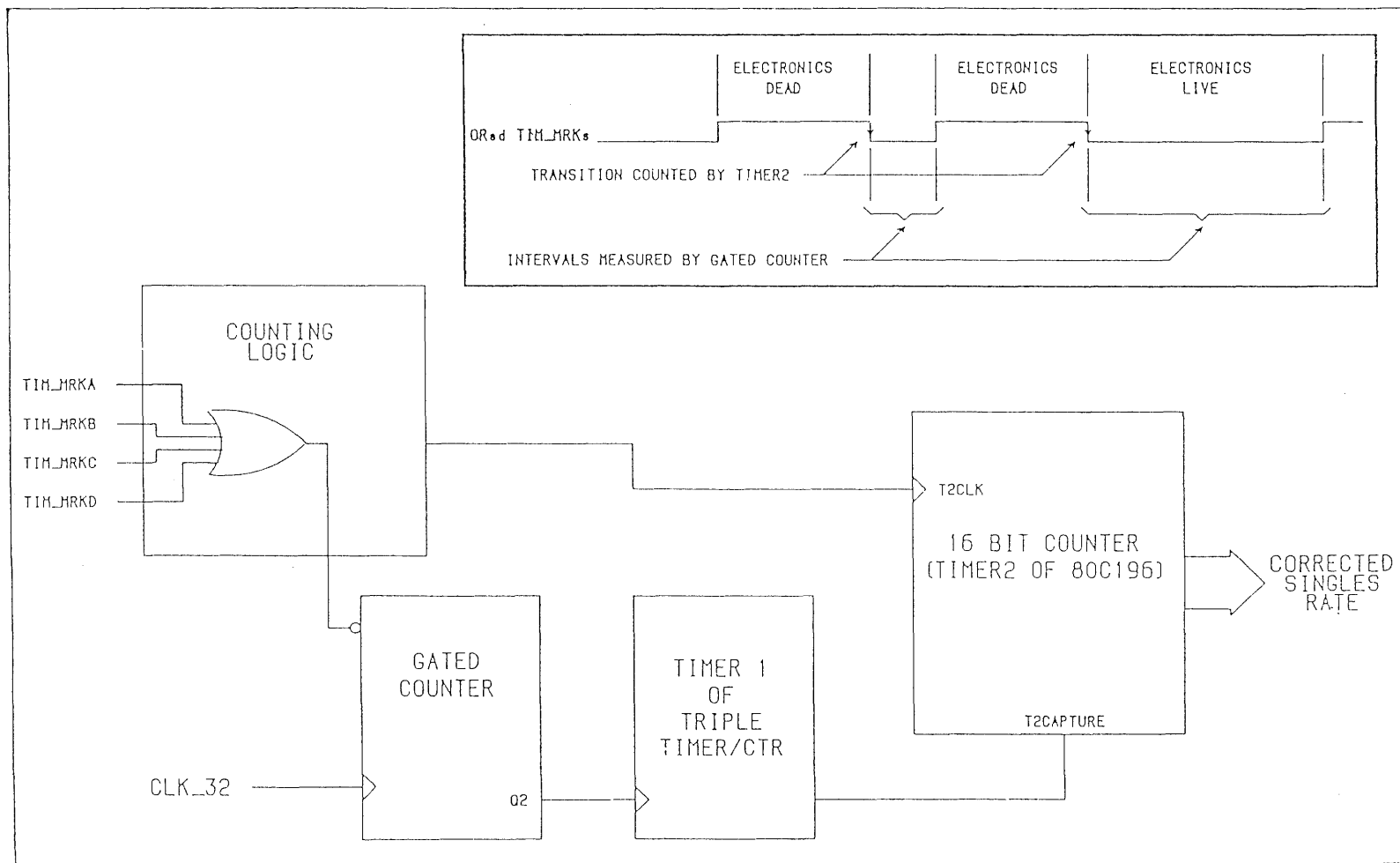


Figure 4.25. Corrected singles counter block diagram.

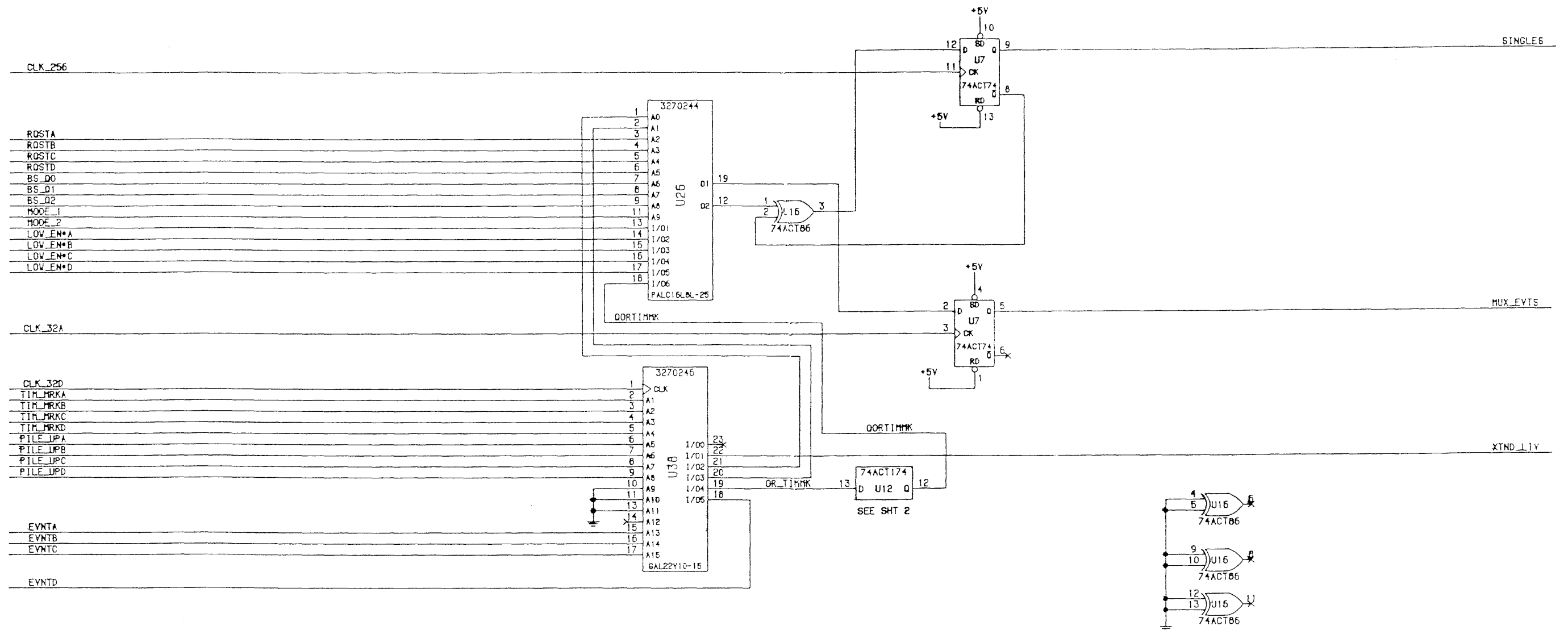


Figure 4.26. Schematic diagram of event counting circuitry.

to the third section of the triple timer/counter, where counting can be monitored by the microcomputer.

Microcomputer

The microcomputer orchestrates all activity on the bucket. It provides some level of administrative support to almost every bucket function performed. The microcomputer selected for this bucket needs to be very prompt in interrupt handling due to the variety of asynchronous events that must be processed. It needs to have an efficient instruction set that lends itself to high-level-language constructs. Due to FDA regulations, the programming language C was defined for use in this bucket design. Efficient math hardware (16 x 16 multiply) is also desirable due to the need to perform floating point operations in some routines of the code.

The Intel 80C196 was selected for its execution speed, instruction set versatility, and peripheral versatility. Another desirable attribute was the high-level-language support and low cost emulator, which fully supports high-level-language debugging.

The 80C196 is a CMOS superset of the NMOS 8096 processor line. It carries with it the same desirable register structure of the 8096. Execution time has been reduced from three oscillator periods per fundamental processor cycle to two, thus decreasing instruction processing time by 33%.

The Microcomputer circuitry is shown in Figures 4.27 and 4.28. It consists of the Microcomputer Core, the basic circuitry required for the Microcomputer to operate, Analog Input Channels, and the Microcomputer Peripherals.

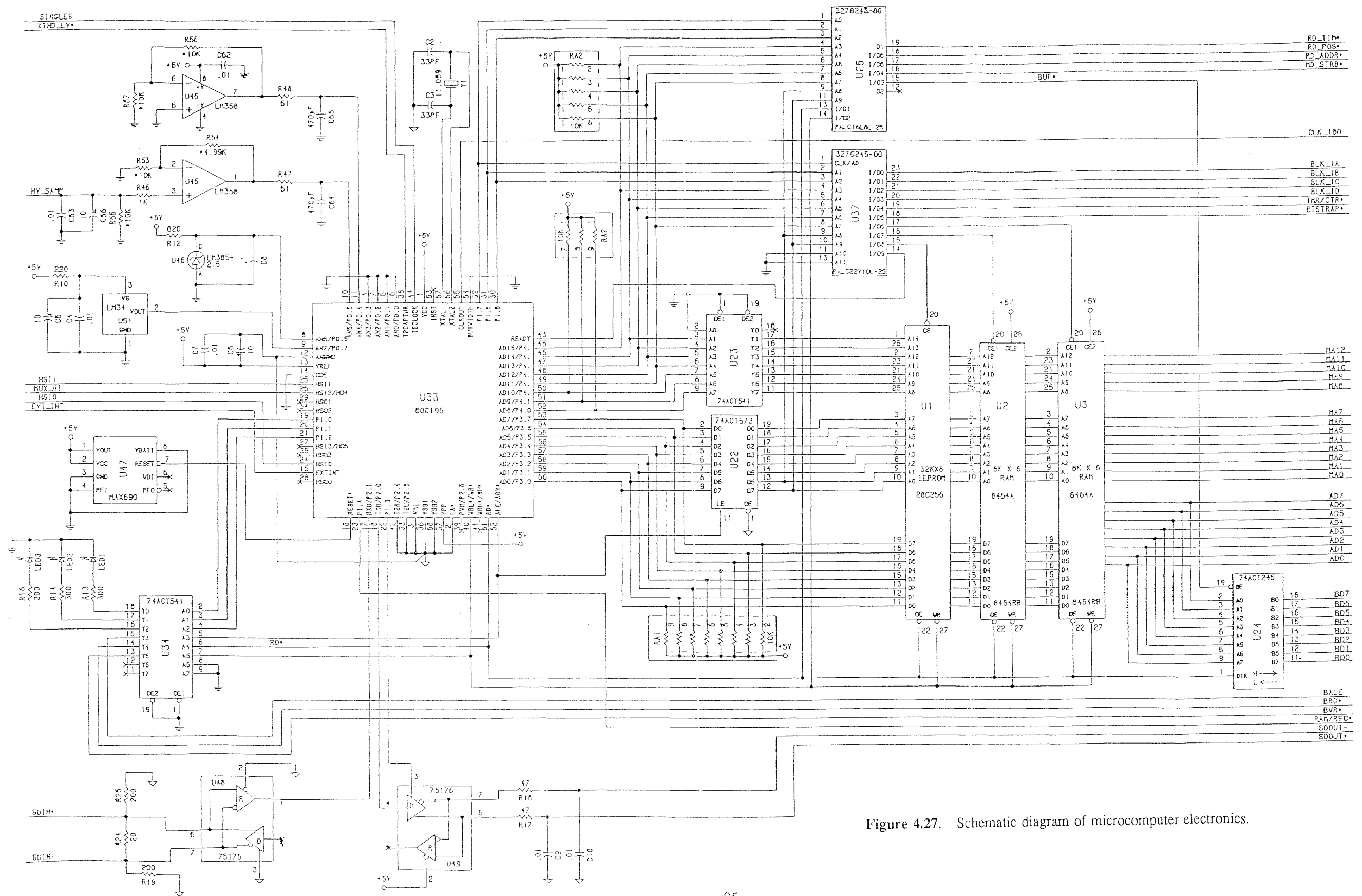


Figure 4.27. Schematic diagram of microcomputer electronics.

Microcomputer Core

The reset circuitry of the microcomputer consists of U47, which contains a precision voltage reference, two comparators, two monostable multivibrators, and a low-power battery switching apparatus. This device maintains its RESET* output (open collector) active until 50 mS after Vcc reaches 4.75 volts, and maintains RESET* high until Vcc drops below 4.65 volts, at which time RESET* goes active (low) again. By functioning in this manner, the microcomputer reset function is performed in an efficient and reliable manner.

The microcomputer clock, whose frequency is determined by crystal Y1, runs at 11.059 MHz to provide precise baud-rate selection through integer prescaling of this value. Maximum clock frequency of the selected microcomputer is 12 MHz.

Three LEDs are used to visually indicate bucket status. These devices, buffered by U34, are driven by port pins P1.0-2. In operation, these LEDs indicate when a serial character is received, when the bucket has been addressed, and the status of the power-up diagnostics.

Registers U22 and U23 serve to latch and buffer the microcomputer address lines. The 80C196 has a multiplexed data/address bus AD0-7, where address information is present for an interval during each processing cycle. By using ALE (U33-62), U22 latches this address information at the proper time. The upper address lines, AD8-15, are static through the entire bus cycle. Thus, U23 is required only to buffer this information. Transceiver U24 serves to buffer the multiplexed data/address bus before it is routed to many of the peripherals and to the Position/Energy Processor sections. Register U36 is an addressable latch used for mode selection such as enabling the Time Correction Circuitry.

RS-422 was chosen as the electrical interface for the low speed serial communications channel of the bucket. Its noise immunity and single-supply versatility are desirable attributes in a modern communications system. Due to standard NRZ serial data protocol with no clock

synchronization fields or SDLC/HDLC error checking, a half duplex transmit-pair/receive-pair scheme was chosen. The UART and baud rate generation circuitry are internal to the 80C196. Transceiver U49 is wired to function as the transmitter, while U48 is configured as the receiver. The transmitter is enabled only during transmission, via port pin P1.3. Cable termination is performed by resistors R17, R18, R19, R24, and R25.

The communications protocol between the Real Time Sorter and the buckets is one of a command/response approach. Under idle conditions, all bucket transmitters are disabled. The Real Time Sorter is the "master" in this scheme, always initiating a command. The first field to be transmitted within this protocol is the address. Each bucket has a unique address, set by DIP switch SW1, which is read on power-up by the microcomputer via buffer U35. Firmware within the monitor routine uses this address to identify commands intended for that specific bucket. The bucket always responds to a command with an acknowledgement, verifying that the command is being executed. This response can also contain data, as in an Upload Histogram command. Appendix A contains a list of bucket commands and a brief summary of their function.

Memory management is performed by PALs U25 and U37, as well as each Position/Energy Processor section's Microcomputer Interface. Table 4.2 (page 98) illustrates the mapping of the microcomputer memory space. Linear addressing of the different peripherals is used, with the four Position/Energy Processor sections being paged into a single eight-kilobyte area. Port 1 (P1.5-7) provides the extended "address" lines required for the paging operation. Port line P1.4 represents RAM/REG*, used to select either the registers or the lookup RAM within the Position/Energy Processor sections.

Table 4.2. Bucket Controller Microcomputer Memory Map (Physical)

Bucket Controller Memory Map

Physical Address	Description
0000h - 1FFFh	Internal microcomputer registers and external Scratch RAM.
2000h - 3FFFh	Bootstrap code space (EPROM).
4000h - 5FFFh	Histogram RAM.
6000h - 7FFFh	Position/Energy Processor section overlay (page) area, mode register, timer/counter, and time and position capture latches. (See description below)
8000h - EFFFh	Main code area (EEPROM).
F000h - FFFFh	Main data area (EEPROM).

Physical Mapping within Addresses 6000h - 7FFFh

uC PORT P1			Physical Address	Overlay Description
P1.7	P1.6	P1.5		
0	0	0	6000h - 7FFFh	Position/Energy Processor section 0.
0	0	1	6000h - 7FFFh	Position/Energy Processor section 1.
0	1	0	6000h - 7FFFh	Position/Energy Processor section 2.
0	1	1	6000h - 7FFFh	Position/Energy Processor section 3.
1	0	0	6000h	Time capture register.
1	0	0	6001h	Position capture register.
1	0	0	6002h	Address DIP switch.
1	0	0	6003h	Mode latch.
1	0	0	7000h - 7003h	Timer/Counter internal registers.

Analog Input Channels

One of the desires of this bucket development was to develop a more reliable bucket in order to produce a more reliable PET scanner. As well as offering many stability improvements through electronic refinements, the bucket is also designed with the ability to monitor board level temperature and high voltage level. Through polling by the Real Time Sorter, the buckets can be interrogated to determine if a problem has occurred that is causing the bucket temperature to rise or the high voltage to drop. Through this monitoring process, a controlled shutdown of the gantry can be effected before severe electronics or detector damage takes place.

The 80C196 internal peripherals include a ten-bit ADC with eight-input multiplexer included. Although not a very accurate ADC, its use for the above-mentioned environmental monitoring is appropriate.

The ADC, a successive approximation type converter, requires the use of V_{cc} for its reference. Noise filtering of this input is performed by capacitors C6 and C7. In order to reduce the uncertainty of the value of V_{cc} , a precision voltage reference, U46, is located on one of the ADC inputs. The firmware samples this input and uses its value to calculate the absolute value of the other inputs. Absolute value of the voltage reference is measured during the initial test

procedure and input into a location in the EEPROM. This allows a much less precise (absolute) reference to be used. The equation for input voltage of a sampled input is:

$$Input\ (volts) = \frac{X_{INPUT}}{X_{REF}} \times V_{REF} \quad (12)$$

where X_{INPUT} = ADC value of converted input

X_{REF} = ADC value of converted reference input

V_{ref} = value (volts) of reference (stored within EEPROM)

The bucket temperature is measured by the temperature sensor U51. This device has a voltage output that is related to device temperature by:

$$V_{out} = T\ (^{\circ}F) \cdot 10\ (\frac{mV}{^{\circ}F})$$

The converted temperature can be requested via the serial port and is reported in degrees fahrenheit.

The high voltage measurement (Figure 4.29) is made by dividing the 1500 volt signal by 750 via the divider string R51-56. Output from this string is clamped by diodes D33 and D34 to prevent a catastrophe in a situation such as the breakdown of the divider string. This divider string and clamp is located on the Position/Energy Processor board, with the low voltage sample being routed to the Bucket Controller. High frequency filtering is performed by the filter consisting of R55, R56, C63, and C66. This filtering serves to reduce noise pickup from the long circuit board trace. Op amp U45 amplifies the sample voltage boosting it by a factor of 1.5 and preventing the ADC from loading this signal. Resistor R47 isolates the output of the amplifier from the capacitance of C64, ensuring op amp stability. Along with capacitor C64, resistor R47

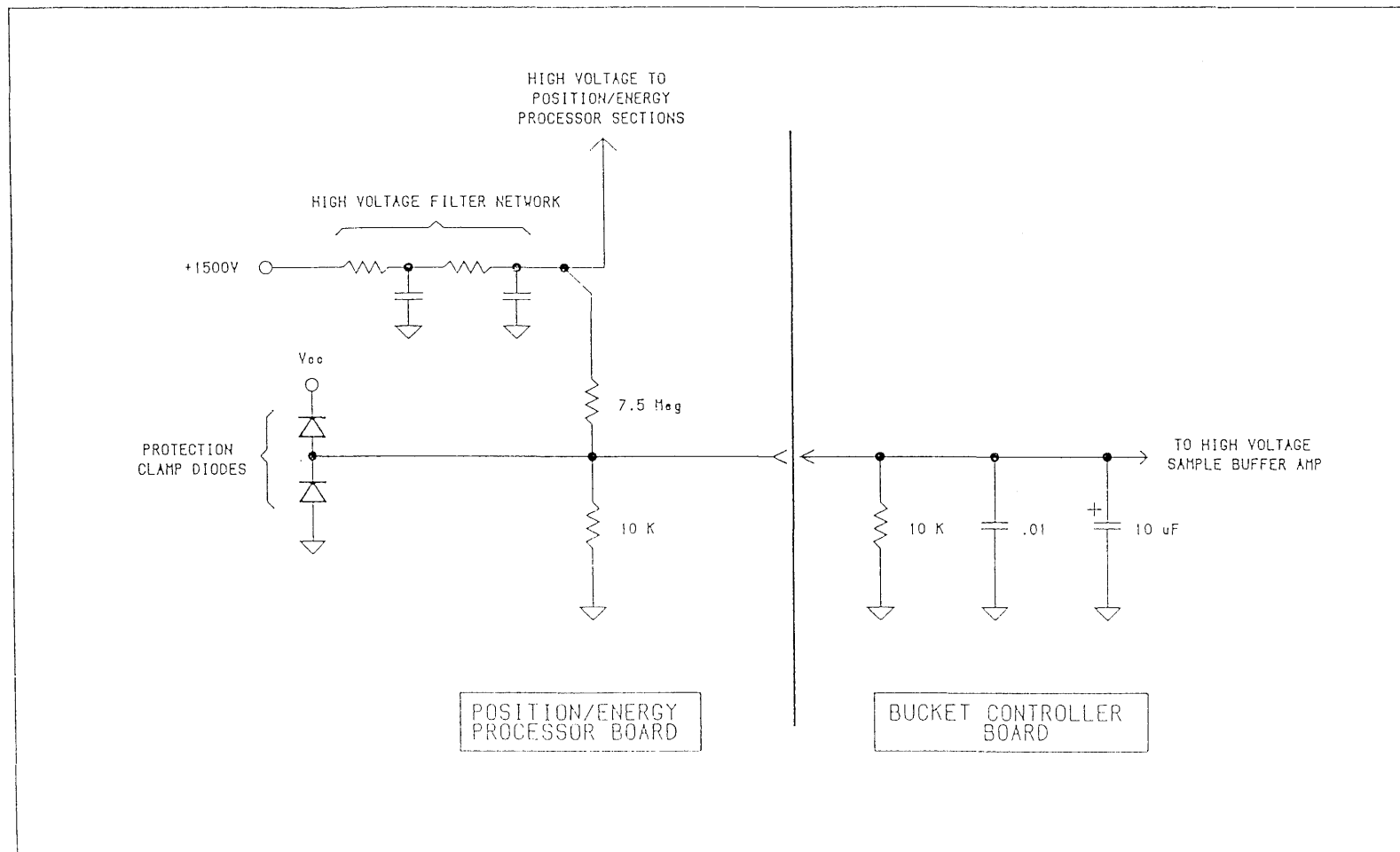


Figure 4.29. High voltage sampling electronics.

also creates a single pole low-pass filter, reducing noise into the ADC.

Microcomputer Peripherals

The microcomputer peripherals include the bootstrap EPROM (U4), EEPROM, histogram/scratch RAM, and the triple timer/counter. These devices reside at fixed physical addresses within the memory map (Table 4.2, page 98).

The EPROM contains the code that is executed upon power-up. This device is mapped in the area where physical constants, sacred to the operation of the microcomputer, are located. These locations include the reset vector and configuration register. The basics of the monitor, which arbitrates all serial communications command/response protocols, are located here. These basic parts perform power up diagnostics, including testing (CRC) of the data integrity within the EEPROM. Upon successful completion of the EEPROM check, code execution is transferred to the code located within the EEPROM. Also included within the EPROM is code to allow download of firmware updates to be loaded into the EEPROM. Locating the download code within this EPROM is mandatory in order to allow first time download of run code into a new EEPROM.

The EEPROM, U1, represents a considerable money savings during the introduction of an embedded product such as the bucket. Considering the complexity of the firmware required to perform the peak location during setup, firmware revisions were considered to be inevitable. The cost (time) required to remove the 32 buckets from a scanner, disassemble the bucket boards, and replace an EPROM in order to effect a firmware revision is major. The cost difference between using a 32k EPROM and a 32k EEPROM can easily be made up by providing this means to allow firmware revisions to be distributed by diskette or tape and downloaded to the

buckets from the host computer. The only stringent requirement is that the code located in the EPROM be right the first time!

Sixteen kilobytes of histogram/scratch RAM (U2 and U3) are located in the 80C196 memory map. The need to be able to efficiently perform position histograms required at least four kilobytes of contiguous ram. The scratch RAM is broken up (Table 4.2, page 98) by the location of internal registers located within the microcomputer hardware.

Triple timer/counter U5 performs the other counting and prescaling functions not performed within the microcomputer. These functions include modula-n division for the corrected singles counting, rate generation for the Position/Energy Processor section's Pseudo Activity Generator circuitry, and counting of events selected by the multiplexer U26/U38. Sequencing waveforms (Figure 4.18, page 76) for the four Position/Energy Processor sections are produced by PAL U6.

CHAPTER V

CONCLUSIONS

Performance of the Bucket

The performance of the new bucket has proven to be very satisfactory. Only through the development of this new bucket can the new eight-by-eight detector block be utilized. Improvements to image resolution, temperature stability, and serviceability are some of the secondary virtues of the new bucket. Many scanners have been assembled with the eight-by-eight detector blocks and these buckets have now been shipped to locations throughout the world.

Performance of the peak-finding routines within the firmware has proven to be excellent. A detector testing-unit has been devised that uses the same bucket hardware and slightly modified firmware to grade the performance of new detectors. By using this unit, a very predictable performance appraisal can be made of the detector before placing it within the gantry. This same testing unit is also used for guiding the Detector Research and Development group in their attempts to optimize the detector manufacturing process.

Little has been said in this thesis about the design of the firmware. This element of the development is considered to be the one highly confidential, and thus proprietary, component of the bucket. Since schematic diagrams of the buckets are distributed with each scanner that is shipped, little control over this information is possible. The machine-code within the EPROM and EEPROM, on the other hand, presents a much more difficult endeavor to reverse-engineer.

The intent of this new bucket development was to develop a reliable bucket that would support a variety of different detector block designs, up to the eight-by-eight configuration. This

objective has been satisfactorily met. Present detectors that have been integrated with the new bucket include the eight-by-eight and the six-by-eight design.

The image resolution of the new scanner shows, by visual observation of the reconstructed image, an obvious increase in transaxial resolution. This is evidenced by the ability to resolve the smallest elements within a Dorenzo phantom, a device with different diameter rods of activity immersed within a water scatter-media. Previously, the smallest elements (5.0 mm) of the Dorenzo phantom had been resolved only by enabling the wobble motion of the detectors. The ability to remove this wobble mechanism from a gantry results in a significant monetary savings.

Much attention to detail was applied to the design of the Analog Processor electronics. It is within this section that temperature dependencies rear their ugly heads, and every possible method to reduce them was applied. Since the bucket firmware can perform an auto-setup on the attached blocks, recalibration of the buckets to their blocks can be performed at the customer site.

This calibration procedure can be performed immediately following installation and at scheduled maintenance intervals by the customer staff. This allows any detector or electronics' drift to be overcome.

The decision to use the EEPROM for firmware storage has been correct. Throughout the development of the scanner, improvements to the firmware have been made. New improvements will continue to be made through the life of the scanner that mandate modification of the code in the buckets. By not having to disassemble the gantry each time a firmware modification is required, two positive effects have resulted: 1) improvements to the bucket firmware are encouraged, and 2) reliability of the scanner is not compromised by the wear of disassembly to perform such modifications.

Future Improvements to the Bucket

Competitive market pressures will continue to drive PET scanner developments toward increased resolution and increased axial FOV coverage. This pressure can only be answered by increasing the number of multi-element detector-block slices. In order to support this ever increasing number of detector elements, the bucket electronics density has to be increased. Realistically, this can be accomplished in one of two ways: 1) by using surface-mount devices, or 2) by moving more of the functionality to the silicon via VLSI and Analog ASIC technology. Either of these approaches support a doubling of electronics density in a next-generation bucket. In order to go beyond this limit (2 generations or more), it appears that the only solution is to go with the second approach.

Work has been in progress for approximately one year to develop certain high density devices that merge the functionality of sections that are now implemented by discrete devices. These high density analog and digital function blocks should support new scanner development for at least two more generations.

REFERENCES

REFERENCES

1. Phelps, M.E., Mazziotta, J.C., Huang, S.C.: *Positron Emission Tomography and Audiography*, Raven Press, 1986.
2. Evans, R.D.: *The Atomic Nucleus*, Kreiger, 1955.
3. Cassen, et al: Imaging of Two Dimensional Radioiodine Concentrations using a Scintillation Counting Scheme, *Nucleonics* 9:63-67, 1951.
4. Anger, H.O.: A New Instrument for Mapping Gamma-Ray Emitters, *Biology and Medicine Quarterly Report UCRL-3653*, 1957.
5. Wrenn, F.R., et al: The Use of Positron Emitting Radioisotopes for Localization of Brain Tumors, *Science* 113:525-527, 1951.
6. Brownell, G.L., Sweet, W.H.: Localization of Brain Tumors, *Nucleonics* 11:40-45, 1953.
7. Cho, Z.H., et al: Circular Ring Transverse Axial Positron Camera for 3-Dimensional Reconstruction of Radionuclides Distribution, *IEEE Trans. Nucl. Sci.* NS-23(1):613-622, 1976.
8. Rankowitz, et al: Positron Scanner for Locating Brain Tumors, *IRE International Convention Record* 9:49-56, 1962.
9. Kuhl, D. E., Edwards, R. Q.: Image Separation Radioisotope Scanning, *Radiology*, 80:653-661, 1963
10. Ter-Pogossian, M.M., et al: A Positron Emission Transaxial Tomograph for Nuclear Imaging (PETT), *Radiology* 114:89-98, 1975.
11. Hoffman, E.J., et al: A New PET System for High Resolution 3-Dimensional Brain Imaging, *J. Nucl. Med.* 28:758-759, 1987.
12. Casey, M.E. and Nutt, R.: A Multicrystal Two Dimensional BGO Detector System for Positron Emission Tomography, *IEEE Trans Nucl Sci* NS-33:570-574, 1986.
13. Cho, Z.H., Farakhi, M.R.: Bismuth Germanate as a Potential Scintillation Detector in Positron Cameras, *Jour Nucl Med* 18:840-44, 1977.
14. Gedcke, D.A., McDonald, W.J.: A Constant Fraction of Pulse Height Trigger for Optimum Time Resolution, *Nucl. Instr. and Methods* 55:337, 1967.
15. John Young, personal communications.
16. M. E. Casey, personal communication.

APPENDICES

APPENDIX A

SUMMARY OF BUCKET COMMANDS

Commands to the electronics are handled using the following format:

[<address>]<command>[<data>]<terminator>

where:

- <address> - [0 - 9] a decimal number with a range 0 to 255 that selects the board of interest. Address is optional, once a board is selected in babble mode and remains active until a different board is selected.
- <command> - [A - Z] a single alphabetic character.
- <data> - [0 - 9] a decimal number with a range 0 to 65564.
- <terminator> - a carriage return.

951 Bucket Commands:

- B<n> -
 - B0 respond with version number of bucket controller and turn on character echo (babble mode).
 - B1 respond version number of monitor.
 - B<n> respond version number of bucket controller.
 - Response: 'M <n>' - version <n> of monitor.
 - 'V <n>' - version <n> of block controller.
- C<n> -
 - C0 respond status of pseudo activity.
 - C1 abort psuedo activity.
 - C2 setup for psuedo activity with count values in analog subsection RAMs.
 - C3 setup for pseudo activity with count values for position and time values set to (55) hex.
 - C65530 run pseudo activity on block zero only.
 - C65531 run pseudo activity on block one only.
 - C65532 run pseudo activity on block two only.
 - C65533 run pseudo activity on block three only.
 - C<n> set pseudo activity counter to <n> (4 - 65529).
 - Response: 'N 0' - psuedo activity mode not active.
 - 'P 0' - psuedo activity mode running.

D<n> - Deposit <n> in open location (0 - 255) (see command "L").
Response: 'X 0' - attempt to deposit in EEPROM code.

E - Examine open location (see command "L").
Response: 'R <n>' - byte value pointed to by "L".

F<n> - F0 return the corrected and uncorrected singles average.
F1 stop singles calculations.
F2 start corrected and uncorrected singles calculations.
Response: 'S <c>,<u>,<d>' - corrected <c>, uncorrected <u> singles rates and diagnostics flag <d>.

G<n> - Set gain of selected tube in selected block to <n> (0 - 255) (see commands "S" and "P").

H<n> - H0 report status of histogram.
H1 select position and time histograms.
H2 select position profile histogram.
H3 select tube energy histogram.
H4 select crystal energy histogram.
H5 select time histogram.
H6 abort histogram.
H<n> start histogram for <n> seconds (7 - 65534).
Response: 'X <n>' - number of seconds remaining.

I<n> - I<n> inhibit block <n> (0 - 3) (also selects block).
I4 inhibit all blocks.

J<n> - J0 respond status of block setup.
J1 auto-setup of selected block (see commands "S" and "I").
J<n> setup selected block using <n> as histogram time (2 - 65534 seconds).
Response: 'S <n>' - stage of setup.

K<n> - K<n> select a crystal in the selected block (0 - 63) (see commands "S" and "I").
K64 select all crystals.

L<n> - Open address <n> for examine or deposit (0 - 65534).

M - Turn off character echo.

O<n> - O<n> set constant fraction discriminator of selected block to <n> (0 - 255).
O256 report the CFD DAC setting of selected block (see commands "S" and "I").
Response: 'C <n>' - CFD DAC setting.

P<n> - P<n> select tube <n> within selected block (0 - 3) (see commands "S" and "I").
P4 select all tubes of selected block.
P5 report tube gains of selected block.
Response: 'G <n0>, <n1>, <n2>, <n3>' - gain settings.

Q<n> - Q0 send next upload record.
Q<n> upload <n> bytes of data in hexadecimal format (0 - 65534).
Response: Data in Intel Intellec format.

R<n> - R0 set bucket to ring one.
R1 set bucket to ring two.
R2 report bucket ring setting.
R3 disable multiples.
R4 enable multiples.
R5 report multiples setting.
R6 disable energy pileup.
R7 enable energy pileup.
R8 report pileup setting.
R9 select local clock.
R10 select system clock.
Response: 'R <n>' - bucket ring mode (0 - ring 0; 1 - ring 1).
'M <n>' - bucket multiple mode (0 - disabled; 1 - enabled).
'P <n>' - bucket energy pileup mode (0 - pileup disabled; 1 - pileup enabled).

S<n> - S<n> enable block <n> (0 - 3) (also selects block).
S4 enable all blocks.

T<n> - Write timeshift for selected block and crystal (0, 1, 3 (-1)) (see commands "S", "I" and "K").

U<n> - U0 report setting of upper discriminator energy.
U<n> set upper level discriminator energy in KeV (0 - 65534).
Response: 'U <n>' - upper discriminator energy in KeV.

V<n> - V0 report setting of lower discriminator energy.
V<n> set lower level discriminator energy in KeV (0 - 65534).
Response: 'V <n>' - lower discriminator energy in KeV.

W<n> - W0 respond status of download.
W1 prepare to receive download data in hexadecimal format.
W2 abort download.
Response: 'C 0' - EEPROM code CRC in progress.
'D 0' - download in progress.
'F 0' - download error has occurred.
'X 0' - download failed.

X<n> -

Diagnostics.

X0 report bucket status.
X1 flash LED's until a carriage return is received.
X2 report reference voltage.
X3 report temperature.
X4 report high voltage.
X5 report spare ADC input.
X6 report position histogram.
X7 report scatter position histogram.
X8 report time histogram.
X9 report position profile histogram.
X10 report tube energy histogram.
X11 report crystal energy histogram.
X12 report total time histogram.
X13 report count of all events.
X14 report count of pileup events.
X15 report count of scatter events.
X16 report count of qualified events.
X17 report EPROM checksum.
X18 report EEPROM code CRC.
X19 report EEPROM data checksum.
X20 timer correction circuit check.
X21 analog subsection sequencer test.
X22 test analog subsection path.
X23 exercise tube gain DAC.
X24 exercise CDF DAC.
X25 scratch pad RAM test.
X26 histogram RAM test.
X27 analog subsection RAM test.
X28 82C54 timer/counter test.
X29 report analog subsection RAM.
X30 analog histogram test.

Response: 'N 0' - test passed.

'C 0' - EEPROM code CRC in progress.

'V <n>' - voltage in millivolts.

'F <n>' - temperature in degrees F.

'X 1' - EEPROM code CRC failed.

'S <n>' - status of electronics.

Bit

0 : 0x01 - EEPROM CRC.
1 : 0x02 - Processor reset detect.
2 : 0x04 - Voltage low.
3 : 0x08 - Temperature high.
4 : 0x10 - Scratch pad RAM.
5 : 0x20 - Histogram RAM.
6 : 0x40 - Processor decoder.
7 : 0x80 - EPROM checksum.

8 : 0x0100 - Analog subsection 0 RAM.
 9 : 0x0200 - " " 1 "
 10 : 0x0400 - " " 2 "
 11 : 0x0800 - " " 3 "
 12 : 0x1000 - Analog subsection 0 sequencer.
 13 : 0x2000 - " " 1 "
 14 : 0x4000 - " " 2 "
 15 : 0x8000 - " " 3 "

16 : 0x010000 - Analog subsection 0 data path.
 17 : 0x020000 - " " 1 " "
 18 : 0x040000 - " " 2 " "
 19 : 0x080000 - " " 3 " "
 20 : 0x100000 - EEPROM data checksum.
 21 : 0x200000 - Timer counter.
 22 : 0x400000 - Time correction.
 23 : 0x800000 - Analog subsection decoder.

Y<n> - Y0 report setting of low energy.
 Y1 disable low energy bit.
 Y<n> set scatter energy and enables scatter bit (2 - 65534).
 Response: "S <0>" - scatter energy setting in KeV.

Z<n> - Z0 execute system reset (RST).
 Z1 set all block parameters to default values.
 Z2 set crystal peak positions to ideal values.
 Z3 set crystal energies to ideal values.
 Z<n> execute code at address <n> (4 - 65534) (response determined by
 routine called).

Standard Responses:

N 0 - Command started or completed successfully.
 U 0 - Unimplemented command.
 E 0 - Invalid command input.
 X <n> - Error report.

APPENDIX B

DESCRIPTION OF FUNCTIONALITY OF EXTENSIVE DIAGNOSTICS ROUTINES TO BE USED DURING BUCKET TESTING

Bucket Diagnostics

EVAL2D.HEX Commands:

- | | | |
|-------|---|----------------------------------------------------------------------------------|
| B 0 | - | Turn on character echo and respond with controller version number. |
| B 1 | - | Respond with monitor version number. |
| B <n> | - | Respond with controller version number. |
| | | |
| C0 | - | Respond status of pseudo activity. |
| C1 | - | Abort pseudo activity. |
| C2 | - | Setup for pseudo activity with counting analog subsection RAM's. |
| C3 | - | Setup for pseudo activity with counting position data and time data set to 0x55. |
| C<n> | - | Set pseudo activity counter to <n> (4 - 65534) |

G <n>	-	Sets tube gain to <n>.
I <n>	-	Inhibits block <n>.
I 4 -		Inhibits all blocks.
O <n>	-	Sets CFD threshold to <n>.
O 256	-	Reports the CFD DAC setting.
P <n>	-	Selects tube <n> (0 <= n <= 3).
P 4 -		Selects all tubes.
P 5 -		Reports tube gains.
R 0	-	Select ring 0.
R 1	-	Select ring 1.
R 2	-	Report ring setting.
R 3	-	Disable multiples.
R 4	-	Enable multiples.
R 5	-	Report multiples setting.
R 6	-	Disable pileup.
R 7	-	Enable pileup.
R 8	-	Report pileup setting.

- R 9 - Select system clock source.
- R 10 - Select local clock source.

- S <n> - Selects block <n>.
- S 4 - Enables all blocks.

- X 0 - Report bucket status.
- X 1 - Flash LED's until a carriage return is received.
- X 2 - Report reference voltage.
- X 3 - Report temperature.
- X 4 - Report high voltage.
- X 5 - Report spare analog input.
- X 9 - Report histogram RAM.
- X 17 - Report EPROM checksum.
- X 18 - Report EEPROM code CRC test.
- X 19 - Report EEPROM data checksum.
- X 20 - Time correction circuit check.
- X 21 - Test analog subsection sequencer.
- X 22 - Test analog subsection data path.
- X 23 - Exercise tube gain DAC.
- X 24 - Exercise CFD DAC.
- X 25 - Scratch pad RAM test.
- X 26 - Histogram RAM test.
- X 27 - Analog subsection RAM test.
- X 28 - 82C54 Timer/counter test.

- X 29 - Report analog subsection RAM.
- X 31 - Run mode histogram test.
- X 32 - Position profile histogram test.
- X 33 - Energy histogram test.
- X 34 - Pseudo activity test.
- X 35 - Analog subsection decoder test.

Power-up Diagnostics (EVAL2.C)

The following tests are performed through the EVAL2.C (EVAL2D.HEX) firmware:

1. Determine the reference voltage.

Response: Power supply voltage (millivolts): x

2. Determine the temperature.

Response: Temperature in F: x

3. Determine the high voltage.

Response: High voltage: x

4. Scratch pad RAM test.

Pass: Scratch RAM test - Passed.

Fail: Scratch RAM test - Failed.

Address: aaaaaaaaaaaaaaaa
Byte written: bbbbbbbb
Byte read: bbbbbbbb

5. Histogram RAM read/write test.

Pass: Histogram RAM read/write test - Passed.

Fail: Histogram RAM read/write test - Failed.

Address: aaaaaaaaaaaaaaaa
Byte written: bbbbbbbb
Byte read: bbbbbbbb

Sliding bit RAM test - Passed.

or

Sliding bit RAM test - Failed.

Address: aaaaaaaaaaaaaaaa
Byte written: bbbbbbbb
Byte read: bbbbbbbb

6. Histogram RAM address test.

Pass: Histogram RAM address test - Passed.

Fail: Histogram RAM address test - Failed.

Address: aaaaaaaaaaaaaaaaaa

Byte written: bbbbbbbb

Byte read: bbbbbbbb

7. Verify EPROM checksum test.

Pass: EPROM checksum test - Passed.

Fail: EPROM checksum test - Failed.

Checksum: x

8. Verify EEPROM checksum test.

Pass: EEPROM checksum test - Passed.

Fail: EEPROM checksum test - Failed.

Checksum: x

9. Processor decoder test.

Pass: (No response).

Fail: Processor decoder test - Failed.

Byte written to scratch RAM: 10101010

Byte read: bbbbbbbb

or

Byte written to histogram RAM: 01010101

Byte read: bbbbbbbb

or

Byte in EPROM: 01011010

Byte read: bbbbbbbb

10. Timer counter test.

Pass: Timer counter test - Passed.

Fail: Timer counter test - Failed.

Timer/counter: x

Byte written: bbbbbbbb

Byte read: bbbbbbbb

11. Analog subsection RAM test.

Pass: Analog subsection n RAM read/write test - Passed.

Fail: Analog subsection n RAM read/write test - Failed.

Address: aaaaaaaaaaaaaaaa

Byte written: bbbbbbbb

Byte read: bbbbbbbb

Sliding bit RAM test - Passed.

or

Sliding bit RAM test - Failed.

Address: aaaaaaaaaaaaaaaa

Byte written: bbbbbbbb

Byte read: bbbbbbbb

12. Analog subsection RAM address test.

Pass: Analog subsection RAM address test - Passed.

Fail: Analog subsection RAM address test - Failed.

Address: aaaaaaaaaaaaaaaa

Byte written: bbbbbbbb

Byte read: bbbbbbbb

13. Analog subsection sequencer test.

Pass: Analog subsection n sequencer test - Passed.

Fail: Analog subsection n sequencer test - Failed.

14. Analog subsection data path test.

Pass: Analog subsection n data path test - Passed.

Fail: Analog subsection n data path test - Failed.

Byte written time latch: bbbbbbbb

Byte read: bbbbbbbb

Byte written position latch: bbbbbbbb

Byte read: bbbbbbbb

15. Position profile histogram mode test.

Pass: Analog subsection n position profile mode test - Passed.

Fail: Analog subsection n position profile mode test - Failed.

Byte written for X ADC value:	bbbbbbbbb
Byte read:	bbbbbbbbb
Byte written for Y ADC value:	bbbbbbbbb
Byte read:	bbbbbbbbb
Byte written for X overflow ADC:	bbbbbbbbb
Byte read:	bbbbbbbbb
Byte written for Y overflow ADC:	bbbbbbbbb
Byte read:	bbbbbbbbb
Energy discriminator RAM bit:	bbbbbbbbb
Byte read:	bbbbbbbbb

16. Energy histogram mode test.

Pass: Analog subsection n run mode test - Passed.

Fail: Analog subsection n run mode test - Failed.

Byte written for energy:	bbbbbbbb
Byte read:	bbbbbbbb
Byte written for position:	bbbbbbbb
Byte read:	bbbbbbbb
Byte written for X overflow ADC:	bbbbbbbb
Byte read:	bbbbbbbb
Byte written for Y overflow ADC:	bbbbbbbb
Byte read:	bbbbbbbb
Byte written for energy overflow ADC:	bbbbbbbb
Byte read:	bbbbbbbb

17. Run mode test.

Pass: Analog subsection n run mode test - Passed.

Fail: Analog subsection n run mode test - Failed.

No interrupt from run mode.

or

Interrupt occurred when should have been disabled by:

Time reject.

or Energy reject.

or ADC overflow reject.

or

Position written:	bbbbbbbbb
Byte read:	bbbbbbbbb
Scatter bit written:	bbbbbbbbb
Byte read:	bbbbbbbbb
Time correction bit 1:	bbbbbbbbb
Byte read:	bbbbbbbbb
Time correction bit 2:	bbbbbbbbb
Byte read:	bbbbbbbbb

or

Position written:	bbbbbbbbb
Byte read:	bbbbbbbbb
Scatter bit written:	bbbbbbbbb
Byte read:	bbbbbbbbb
Selected block:	bbbbbbbbb
Byte read:	bbbbbbbbb

18. Pseudo activity test.

Pass: Analog subsection n psuedo activity test - Passed.

Fail: Analog subsection n psuedo activity test - Failed.

No interrupt occured from psuedo activity.

or

Time count value:	bbbbbbbbb
Byte received:	bbbbbbbbb
Position count value:	bbbbbbbbb
Byte received:	bbbbbbbbb

19. Analog subsection decoder test.

Pass: Analog subsection decoder test - Passed.

Fail: Analog subsection decoder test - Failed.

Analog subsection written to: x

Byte written: bbbbbbbb

Byte read: bbbbbbbb

20. Time correction circuit test.

Pass: Time correction test - Passed.

Fail: Time correctino test - Failed.

Time byte written: bbbbbbbb

Properly corrected byte: bbbbbbbb

Corrected byte read: bbbbbbbb

Scratch Pad RAM Test (test_scratch_RAM).

The scratch pad RAM test routine performs a simple byte read/write for all values in the RAM.

Note: Since the scratch RAM hold the C variables the test does not alter the contents of the RAM.

Procedure:

1. Initialize the RAM pointer to the start of the scratch pad RAM.
2. Read byte from RAM and save it in a local register.
3. Write AA hex to RAM location.
4. Read RAM location and compare with AA hex.
5. If byte is not correct then return with error. Report the RAM address, byte written and byte read.
6. Restore the RAM byte and increment RAM pointer.
7. If completed all RAM locations then repeat step 2 through 6 writting 55 hex to RAM.

RAM read/write test (test_RAM_read_write).

The RAM read/write test routine performs a simple byte read/write for all values in the RAM.

Note: The routine is a destructive write to the RAM contents.

Procedure:

1. Initialize the RAM pointer to the start of the RAM.
2. Write AA hex and 55 hex to alternate RAM locations for the entire RAM.
3. Verify the contents of the RAM.
4. If a byte is not correct then return with error. Report the RAM address, byte written and byte read.
5. Repeat steps 2 through 4 writing 55 hex then AA hex to all RAM locations.

Sliding bit RAM test (test_RAM_sliding_bit).

The sliding bit RAM test routine slides a one through the RAM byte in which a RAM read/write test failed.

Procedure:

1. Set the RAM pointer to the RAM location.
2. Write a zero to the RAM location.
3. If the byte is not correct then return with error. Report the RAM address, byte written and byte read.
4. Shift a one to the next byte location.
5. Repeat steps 3 through 4 until all bits have been tested.
6. Repeat test sliding a zero through all bit locations.

RAM address test (test_RAM_address).

The RAM address test routine test the address bits by sliding a one and then a zero through the address bits and testing the bytes written to the RAM.

Procedure:

1. Zero all locations in RAM.
2. Write a one the first RAM location.
3. Shift a one in the RAM address and write the previous byte written to the RAM incremented by 11 hex for all RAM addressed locations.
4. If the bytes written to RAM are not correct then return with error and report the RAM address, byte written and byte read.
5. Repeat test by sliding a zero through the RAM address bits.

Processor decoder test.

The processor decoder test routine writes values to the scratch pad RAM and the histogram RAM to verify the processor decoder PAL.

Procedure:

1. The value A5 hex is written to the scratch pad RAM location 0FFD hex and 5A hex is written to histogram RAM location 4FFD.
2. The two bytes are then read and verified as well as the EPROM address 21FD hex.
3. If any of the bytes are not valid then an error is reported.

Timer counter test (test_timer_counter).

The counter test routine performs a read/write test of the counter/timer.

Procedure:

1. Different modes are written to each counter (mode 0 to counter 0, mode 1 to counter 1 and mode 2 to counter 2).
2. The counter modes are then read from the counter and verified.
3. If an error occurs then the counter, byte written and byte read are reported.

Analog subsection sequencer test (test_analog_sequencer).

The analog sequencer routine triggers the analog subsection sequencer and verifies an external interrupt occurs.

Procedure:

1. All block are inhibited.
2. The position capture latch is read to clear the interrupt line and the processor waits for the one-shot to time out.
3. The interrupt flag blk_flag_extint_test bit is cleared.
4. The analog subsection sequencer is triggered by writing EB hex then E9 hex.
5. The interrupt flag is then tested for an interrupt and the test result returned (Blocks are re-enabled and the analog subsection latch returned to run mode).

Analog subsection data path test (test_analog_subsection_path).

The analog subsection data path test routine performs several writes and reads through the analog subsection latches.

Procedure:

1. Set processor mode latch to system clock, count all events, disable multiples, disable time correction and ring 0 (20 hex).
2. Inhibit all blocks.
3. Set the analog subsection mode latch to inhibit block, position profile mode, pileup rejector disabled, permit position and time register writes and run mode.
4. Clear the position latch.
5. Write AA hex to the time latch and D5 hex to the position latch and verify values.
6. If an error then report the bytes written to the time and position latches and the bytes read.
7. Repeat steps 5 and 6 writing 55 hex to time latch and AA hex to position latch.
8. Return the mode latches to the run state and re-enable all blocks.

Analog subsection position profile histogram mode test (test_position_profile_histogram).

The analog subsection position profile histogram mode test routine test the sequencers that control the position profile histogram mode.

Procedure:

1. Inhibit all blocks.
2. Set processor mode latch to system clock, count all events, disable multiples, disable time correction and ring 0 (20 hex).
3. Set analog subsection mode to run mode, position histogram mode, pile-up rejector disabled (E9 hex).
4. Clear the entire analog subsection RAM.
5. Slide a one in each bit position of the X ADC DAC, Y ADC DAC, and the energy ADC DAC and verify the histogram results.
6. If an error occurs then report the X ADC byte written and read, the Y ADC byte written and read, X ADC overflow bit written and read, Y ADC overflow bit written and read, and the energy ADC overflow bit written and read.
7. Repeat steps 5 and 6 sliding a zero through all ADC DAC positions.
8. Now load the lower half of the analog subsection RAM with C0 hex to test the energy discriminator bit.
9. Set all ADC DAC values to zero and trigger the analog subsection RAM sequencer.
10. If an error is detected then report the bytes written and read.

Analog subsection energy histogram mode test (test_energy_histogram).

The analog subsection energy histogram mode test routine test the sequencers that control the energy histogram mode.

Procedure:

1. Inhibit all blocks.
2. Set processor mode latch to system clock, count all events, disable multiples, disable time correction and ring 0 (20 hex).
3. Set analog subsection mode to run mode, energy histogram mode, pile-up rejector disabled (ED hex).
4. Clear the entire analog subsection RAM.
5. Slide a one in each bit position of the X ADC DAC, Y ADC DAC, and the energy ADC DAC and verify the histogram results.
6. If an error occurs then report the energy byte written and read, the position byte written and read, X ADC overflow bit written and read, Y ADC overflow bit written and read, and the energy ADC overflow bit written and read.
7. Repeat steps 5 and 6 sliding a zero through all ADC DAC positions.

Run mode histogram test (test_run_mode_histogram).

Procedure:

1. Inhibit all blocks.
2. Set processor mode latch to system clock, count all events, disable multiples, disable time correction and ring 0 (20 hex).
3. Set analog subsection mode to run mode, pile-up rejector disabled (E1 hex).
4. Load the lower analog subsection RAM with count values (crystal values) zero to sixty-three and repeat.
5. Load the upper analog subsection RAM with the same values except append either C0 hex for crystal values less than 20 hex and 80 hex for crystal values greater than or equal to 20 hex.
6. Slide a one through all possible combinations of X ADC, Y ADC and energy ADC values.
7. Test the position, scatter bit and time correction bits (time correct bits should be zero) and report values if an error occurs.
8. Now write 80 hex to the first byte of the lower analog subsection RAM and set all ADC DAC values to zero to set the time correction bit 1.
9. Verify the time correction bit 1 is set.
10. Write C0 hex to the first byte of the lower analog subsection RAM to set both time correction bits.
11. Verify both time correction bits are set.
12. Write 40 hex to the first byte of the lower analog subsection RAM to disable the request bit through the time correction bits.

13. Verify that an interrupt does not occur.
14. Write zero to the first byte of the lower analog subsection RAM to allow request bits and write zero to the first byte of the upper analog subsection RAM for energy discrimination.
15. Verify that an interrupt does not occur.
16. Return the first byte of the upper analog subsection RAM to C0 hex and set each of the ADC overflow bits to ensure an interrupt does not occur.
17. Set the processor mode latch to enable time correction and set all the ADC values to zero.
18. Allow the analog subsection to trigger and verify the block bits.

Pseudo activity test (test_pseudo_activity).

Procedure:

1. Inhibit all blocks in run mode state.
2. Load the analog subsection RAM even bytes with count values 0 through 63 and the odd bytes with 0 through 63.
3. Set processor mode latch to system clock, count all events, disable multiples, disable time correction and ring 0.
4. Write 2000 to the rate generator counter.
5. Reset and enable the pseudo activity counter on the selected block.

6. Test the first 256 pseudo activity values to be output. If no interrupt is generated then report no interrupt else if an error is detected then report the position and time count values.
7. Turn off the pseudo activity generator and return.

Analog subsection decoder test (test_analog_subsection_decoder).

Procedure:

1. Inhibit all blocks.
2. Write the values zero through four to the first byte of the analog subsection RAM's for each analog subsection.
3. Verify the analog subsection RAM bytes.
4. If an error occurs then report the analog subsection number, the byte written and the byte read.

Time correction circuit test (test_time_correction).

Procedure:

1. Write 30 hex to the processor mode latch to enable time correction.
2. Inhibit all blocks and write C9 hex to the selected analog subsection mode latch to allow writes to the position and time latches.
3. Read the position capture latch to clear both the time and position capture latches.

4. Test the time correction addition ($127 + 1$) by writing 7F hex the time latch and 80 hex to the position latch of the analog subsection.
5. Read the time capture latch and verify result. If an error is detected then report the time and position latches and the time capture latch.
6. Test the time correction subtraction ($0 - 1$) by writing 80 hex the time latch and 80 hex to the position latch of the analog subsection.
7. Read the time capture latch and verify result. If an error is detected then report the time and position latches and the time capture latch.
8. Return the analog subsection to run mode and re-enable all blocks.

APPENDIX C

PAL SOURCE LISTINGS OF BUCKET

U98 - ANSBSQ81

```

PARTNO      3270247-00;
DATE        7/13/89;
REVISION    1;
DESIGNER    JCM;
COMPANY     CTI;
ASSEMBLY    ANALOG SUBSECTION SEQUENCER;
LOCATION     POS/ENERGY U18, U44, U72, U98;
DEVICE     P22V10 ;
  
```

\$INCLUDE PALASM.OPR

```

{ ----- }
{ FIRST OF TWO PALS USED FOR SEQUENCER IN ANALOG SUBSECTION }
{ }
{ SEE TIMING DIAGRAMS (PCAD PCCAPS FORMAT) FOR SPECIFIC DETAILS }
{ }
{ FILE NAMES FOR TIMING DIAGRAMS ARE AS FOLLOWS: }
{ }
{ 1. EHSTWAV1.SCH }
{ 2. PHSTWAV1.SCH }
{ 3. THSTWAV1.SCH }
{ 4. RUNWAV1 .SCH }
{ ----- }
  
```

```

{ ----- }
{ PIN DEFINITIONS }
{ ----- }
{ INPUTS : OUTPUTS }
{ ----- }
  
```

PIN	1	= CLK_32	;				
PIN	2	= TIM_MRK	;	PIN	23	= /CE1_RAM	;
PIN	3	= BLK_INHIB	;	PIN	22	= ADC_POS	;
PIN	4	= Q4	;	PIN	21	= HST_RQST	;
PIN	5	= Q3	;	PIN	20	= PU_STRB	;
PIN	6	= INIT_SEQ	;	PIN	19	= /ADC_CLK	;
PIN	7	= Q2	;	PIN	18	= CF_RESET	;
PIN	8	= Q1	;	PIN	17	= ASA12	;
PIN	9	= Q0	;	PIN	16	= /OE_RAM	;
PIN	10	= MODE_9	;	PIN	15	= /CTR_EN	;
PIN	11	= MODE_8	;	PIN	14	= /CTR_RST	;
PIN	13	= Q5	;				

```

{ ----- }
  
```

{----- DECLARATIONS AND INTERMEDIATE VARIABLES ----- }

FIELD CTRVAL = [Q5..Q0] ;{ COUNTER BITS FROM 74F579 }

RUN_MODE = /MODE_9 * /MODE_8 ;{ SEE RUNWAV .SCH }

TIM_HIST_MODE = /MODE_9 * MODE_8 ;{ SEE PHISTWAV.SCH }

POS_HIST_MODE = MODE_9 * /MODE_8 ;{ SEE THISTWAV.SCH }

ENG_HIST_MODE = MODE_9 * MODE_8 ;{ SEE EHISTWAV.SCH }

{ ----- }

{ ----- EQUATIONS ----- }

CE1_RAM.D = (RUN_MODE + TIM_HIST_MODE
+ ENG_HIST_MODE) * (CTRVAL:['D'23..'D'31])
+ POS_HIST_MODE * CTRVAL:['D'23..'D'26]
+ BLK_INHIB ;{REV3}

ADC_POS.D = POS_HIST_MODE * (CTRVAL:['D'27..'D'31]) ;

HST_RQST.D = (TIM_HIST_MODE + POS_HIST_MODE
+ ENG_HIST_MODE) * (CTRVAL:['D'15..'D'31]) ;

PU_STRB.D = /(CTRVAL:['D'14..'D'15]) ;

ADC_CLK.D = (CTRVAL:'D'10) + (CTRVAL:'D'12)
+(CTRVAL:'D'21) + (CTRVAL:'D'23) ;

CF_RESET.D = BLK_INHIB + (CTRVAL:['D'23..'D'31]) ;

ASA12.D = (CTRVAL:['D'28..'D'31]) ;

ASA12.OE = /BLK_INHIB;

OE_RAM.D = (RUN_MODE + TIM_HIST_MODE
+ENG_HIST_MODE) * (CTRVAL:['D'24..'D'31])
+ POS_HIST_MODE * CTRVAL:['D'24..'D'26];{REV3}


```

OE_RAM.OE      =    /BLK_INHIB ;

CTR_EN.D       =    (CTRVAL:['D'0..'D'29]) + /INIT_SEQ ; { THIS MAKES THE
                                COUNTER HANG IF
                                INIT_SEQ IS STILL
                                ACTIVE AFTER SEQ.}

CTR_RST.D      =    /((TIM_MRK + INIT_SEQ) * (CTRVAL:'D'0)
                                + (CTRVAL:['D'1..'D'31])) ; { COUNTER NOT RESET HERE
                                THIS, ALONG WITH CTR_EN
                                ALLOWS THE SEQUENCER TO
                                WAIT UNTIL INIT_SEQ GOES
                                INACTIVE TO RESET. }
                                {REV4}

[ADC_CLK,CE1_RAM,CF_RESET,ADC_POS,HST_RQST,OE_RAM,PU_STRB,ASA12,CTR_EN,
CTR_RST].sp     =    'b'0;

[ADC_CLK,CE1_RAM,CF_RESET,ADC_POS,HST_RQST,OE_RAM,PU_STRB,ASA12,CTR_EN,
CTR_RST].ar     =    'b'0;

```

U97 - ANSBSQ82

PARTNO 3270234-00;
 DATE 6/13/89;
 REVISION 1;
 DESIGNER JCM;
 COMPANY CTI;
 ASSEMBLY 0;
 LOCATION POS/ENERGY U19, U50, U71, U97;
 DEVICE G16V8 ;

\$INCLUDE PALASM.OPR

```
{ ----- }
{ SECOND OF TWO PALS USED FOR SEQUENCER IN ANALOG SUBSECTION }
{ }
{ SEE TIMING DIAGRAMS (PCAD PCCAPS FORMAT) FOR SPECIFIC DETAILS }
{ }
{ FILE NAMES FOR TIMING DIAGRAMS ARE AS FOLLOWS: }
{ }
{ 1. EHSTWAV1.SCH }
{ 2. PHSTWAV1.SCH }
{ 3. THSTWAV1.SCH }
{ 4. RUNWAV1 .SCH }
{ }
```

```
{ ----- }
{ PIN DEFINITIONS }
{ ----- }
{ INPUTS : OUTPUTS }
{ ----- }
```

PIN 1 = CLK_32 ;	
PIN 2 = Q5 ;	PIN 19 = /OE_EVT ;
PIN 3 = Q4 ;	PIN 18 = /OE_ADC ;
PIN 4 = Q3 ;	PIN 17 = /OE_POS ;
PIN 5 = Q2 ;	PIN 16 = OE_E ;
PIN 6 = Q1 ;	PIN 15 = OE_X ;
PIN 7 = Q0 ;	PIN 14 = /PU_RST ;
PIN 8 = MODE_9 ;	PIN 13 = /POS_STRB ;
PIN 9 = MODE_8 ;	PIN 12 = /PE_STRB ;

```
{ ----- }
```

{----- DECLARATIONS AND INTERMEDIATE VARIABLES ----- }

```
FIELD CTRVAL      = [Q5..Q0]                ;

RUN_MODE          = /MODE_9 * /MODE_8      ;{ SEE RUNWAV .SCH }
TIM_HIST_MODE     = /MODE_9 * MODE_8       ;{ SEE PHISTWAV.SCH }
POS_HIST_MODE     = MODE_9 * /MODE_8       ;{ SEE THISTWAV.SCH }
ENG_HIST_MODE     = MODE_9 * MODE_8        ;{ SEE EHISTWAV.SCH }
```

{ ----- }

{ ----- EQUATIONS ----- }

```
POS_STRB.D      =    CTRVAL:'D'26 ;

PE_STRB.D       =    CTRVAL:['D'29..'D'30] ;

OE_EVT.D        =    (RUN_MODE + TIM_HIST_MODE) * CTRVAL:['D'28..'D'31] ;

OE_ADC.D        =    (POS_HIST_MODE + ENG_HIST_MODE) ;

OE_POS.D        =    CTRVAL:['D'28..'D'31] ;

OE_E.D          =    ((RUN_MODE + TIM_HIST_MODE + ENG_HIST_MODE)
                     * (CTRVAL:['D'12..'D'15] + CTRVAL:['D'28..'D'31]))
                     + ( POS_HIST_MODE * ((CTRVAL:['D'12..'D'15] )
                     + CTRVAL:['D'24..'D'26] )); {REV3}

OE_X.D          =    ((RUN_MODE + ENG_HIST_MODE + TIM_HIST_MODE)
                     * (CTRVAL:['D'24..'D'26]))
                     +( POS_HIST_MODE * (CTRVAL:['D'28..'D'31])) ; {REV3}

PU_RST.D        =    CTRVAL:'D'1 ;
```

U88 - TVMUXV10

PARTNO 3270232-00;
 DATE 1/28/89;
 REVISION 1;
 DESIGNER JCM;
 COMPANY CTI;
 ASSEMBLY ANALOG SUBSECTION TIME BUS MULTIPLEXER ;
 LOCATION POS/ENERGY U8, U36, U62, U88;
 DEVICE P22V10 ;

\$INCLUDE PALASM.OPR

```
{ ----- }
{           TIME BUS MULTIPLEXER           }
{ ----- }
{ This pal is used stricly for bucket self-test and histogramming }
{ functions. It either places ASA0 - 5 or ASD0 - 6 onto the time }
{ bus (TIME0 - 6). The former allows placing either X-ADC or }
{ E-ADC data directly into Common Electronics data path (for }
{ histogramming), while the latter allows the processor (or }
{ UCIFACE pseudo activity sequencer) to force data into the }
{ time data path. All outputs are combinatorial. }
{ }
{ ----- }
```

```
{ ----- }
{           PIN DEFINITIONS           }
{ ----- }
{ INPUTS          :          OUTPUTS          }
{ ----- }
```

PIN	1	= ASA0	;	PIN	23	= TIME_0	;
PIN	2	= ASA1	;	PIN	22	= TIME_1	;
PIN	3	= ASA2	;	PIN	21	= TIME_2	;
PIN	4	= ASA3	;	PIN	20	= TIME_3	;
PIN	5	= ASA4	;	PIN	19	= TIME_4	;
PIN	6	= ASA5	;	PIN	18	= TIME_5	;
PIN	7	= ASD0	;	PIN	17	= TIME_6	;
PIN	8	= ASD1	;				
PIN	9	= ASD2	;				
PIN	10	= ASD3	;				
PIN	11	= ASD4	;				
PIN	13	= ASD5	;				
PIN	14	= ASD6	;				
PIN	15	= /OE_ADC	;				
PIN	16	= /TIM_BF	;				

```

{ ----- }
{ ----- EQUATIONS ----- }
{ ----- }

```

```

{ Each of the output lines gets the appropriate ASA line if OE_ADC }
{ is active, or the ASD line if TIM_BF is active }

```

```

TIME_0      =      OE_ADC * /TIM_BF * ASA0
               + OE_ADC * TIM_BF * ASD0      ;

```

```

TIME_1      =      OE_ADC * /TIM_BF * ASA1
               + OE_ADC * TIM_BF * ASD1      ;

```

```

TIME_2      =      OE_ADC * /TIM_BF * ASA2
               + OE_ADC * TIM_BF * ASD2      ;

```

```

TIME_3      =      OE_ADC * /TIM_BF * ASA3
               + OE_ADC * TIM_BF * ASD3      ;

```

```

TIME_4      =      OE_ADC * /TIM_BF * ASA4
               + OE_ADC * TIM_BF * ASD4      ;

```

```

TIME_5      =      OE_ADC * /TIM_BF * ASA5
               + OE_ADC * TIM_BF * ASD5      ;

```

```

TIME_6      =      OE_ADC * TIM_BF * ASD6      ;

```

```

{Turn on the output drivers for either OE_ADC or TIM_BF }

```

```

[TIME_0,TIME_1,TIME_2,TIME_3,TIME_4,TIME_5,TIME_6].OE = OE_ADC ;

```

U96 - TP_39

```

PARTNO      3270236-00;
DATE        2/6/89;
REVISION    6;
DESIGNER    JCM;
COMPANY     CTI;
ASSEMBLY    ANALOG SUBSECTION TIME-POSITION LOGIC;
LOCATION      POS/ENERGY U24, U49, U70, U96;
DEVICE      G6001 ;

```

```
$INCLUDE PALASM.OPS
```

```

{ ----- }
{           ANALOG SUBSECTION TIME-POSITION LOGIC PAL           }
{ ----- }
{   This pal performs a multitude of functions including latching }
{   the overflow signals from all three ADCs, multiplexing the    }
{   clock signals to the time and position latches, routing the   }
{   time correction bits to the proper latches, and creation of the }
{   RQST signal.                                                  }
{ ----- }

```

```

{ ----- }
{           PIN DEFINITIONS                                     }
{ ----- }
{   INPUTS           :           OUTPUTS                     }
{ ----- }
PIN    2    = EN_OF      ;    PIN    23   = TD7              ;
PIN    3    = X_OF      ;    PIN    22   = PRQST             ;
PIN    4    = Y_OF      ;    PIN    21   = PD6              ;
PIN    5    = ASD6      ;    PIN    20   = PD7              ;
PIN    6    = ASD7      ;    PIN    19   = T_STRB            ;
PIN    7    = PILEUP    ;    PIN    18   = PT_STRB           ;
PIN    8    = PU_EN     ;
PIN    9    = HST_RQST  ;
PIN    10   = NRM_FRC   ;
PIN    11   = MODE_8    ;

PIN    13   = POS_STRB  ;
PIN    14   = FRC_TIM   ;
PIN    15   = PE_STRB   ;
PIN    16   = FRC_PE    ;
PIN    17   = MODE_9    ;

{ ----- }

```

{----- DECLARATIONS AND INTERMEDIATE VARIABLES ----- }

```

RUN_MODE      =  /MODE_9 * /MODE_8  ;{ SEE RUNWAV .SCH }
TIM_HIST_MODE  =  /MODE_9 *  MODE_8  ;{ SEE PHSTWAV.SCH }
POS_HIST_MODE  =  MODE_9 * /MODE_8  ;{ SEE THSTWAV.SCH }
ENG_HIST_MODE  =  MODE_9 *  MODE_8  ;{ SEE EHSTWAV.SCH }
FORCE          =  /NRM_FRC           ;
SCATTER        =  /ASD6              ;{USED FOR PD6 }
PINNODE 25     =  LX_OF               ;{BURY REGISTER }
PINNODE 26     =  LY_OF               ;{BURY REGISTER }
PINNODE 27     =  LD6                 ;{BURY REGISTER }
PINNODE 28     =  LD7                 ;{BURY REGISTER }
VALLEY_REJECT  =  /LD7 * LD6          ;{VALLEY REJECT VALUE
                                         OF TIM.CORR.}

```

{ ----- }

{ ----- EQUATIONS ----- }

```

{Buried registers used as input registers for x-overflow, y-overflow,
{asd6, and asd7. These are buried because their output is not needed
{by the outside world, and we need their output pins. Notice how their
{outputs are used as inputs by tri-stating their outputs.

```

{ ----- }

```

LX_OF.D       =      X_OF           ;
LY_OF.D       =      Y_OF           ;
LD6.D         =      ASD6           ;
LD7.D         =      ASD7           ;

LX_OF.CE      =      'B'1           ;
LY_OF.CE      =      'B'1           ;
LD6.CE        =      'B'1           ;
LD7.CE        =      'B'1           ;

```

```

PRQST      =   ASD7 * FORCE
              + HST_RQST
              + ASD7 * /EN_OF * /LX_OF * /LY_OF * /(PILEUP * PU_EN)
              * /VALLEY_REJECT                      ;

TD7         =   /FORCE * RUN_MODE * LD6
              + /FORCE * TIM_HIST_MODE * LD7
              + /FORCE * POS_HIST_MODE * LD7
              + /FORCE * ENG_HIST_MODE * EN_OF
              + FORCE * ASD7                        ;

PD6         =   /FORCE * RUN_MODE      * SCATTER
              + /FORCE * TIM_HIST_MODE * LX_OF
              + /FORCE * POS_HIST_MODE * X_OF
              + /FORCE * ENG_HIST_MODE * LX_OF
              + FORCE * ASD6                        ;

PD7         =   /FORCE * RUN_MODE      * LD7
              + /FORCE * TIM_HIST_MODE * LY_OF
              + /FORCE * POS_HIST_MODE * Y_OF
              + /FORCE * ENG_HIST_MODE * LY_OF
              + FORCE * ASD7                        ;

T_STRB      =   NRM_FRC * PE_STRB
              + /NRM_FRC * FRC_TIM                      ;

PT_STRB     =   NRM_FRC * PE_STRB
              + /NRM_FRC * FRC_PE                      ;

[LX_OF,LY_OF,LD6,LD7].AR = 'B'0                      ;

```


U99 - UCI1

PARTNO 3270235-00;
 DATE 1/23/89;
 REVISION 2;
 DESIGNER JCM;
 COMPANY CTI;
 ASSEMBLY AN. SUB. MICROCONTROLLER INTERFACE DECODER;
 LOCATION POS/ENERGY U20, U45, U73, U99;
 DEVICE P16L8 ;

\$INCLUDE PALASM.OPR

```
{
{ -----
{          BLOCK CONTROLLER DECODER PAL
{ -----
{
{   One of two PALs that perform decoding for the 80c196 processor.
{   This PAL performs address decoding for the Time and Position
{   capture latches, the DIPSWITCH buffer, and the MODE latch.
{
{
{   Refer to drawing titled:
{
{               BLOCK CONTROLLER
{               BLOCK PROCESSOR/CONTROLLER 1
{
{ -----
{
{ REV 2 2-27-89 Remapped the Analog Subsections and Registers from
{           E000 to FFFF to 6000 to 7FFF.
{
{
{
{
{ -----
{          PIN DEFINITIONS
{ -----
{
{          INPUTS          :          OUTPUTS
{ -----
}
```

PIN	1	=	MA12;			
PIN	2	=	MA11;	PIN	19	= /WR_RAM ;
PIN	3	=	MA10;	PIN	18	= /GN_STB1 ;
PIN	4	=	MA9;	PIN	17	= /GN_STB2 ;
PIN	5	=	MA8;	PIN	16	= B_A ;
PIN	6	=	BLK_1_;			
PIN	7	=	RAM_REG;	PIN	14	= /CF_STRB ;
PIN	8	=	/BRD;	PIN	13	= /ADDR_BUF ;
PIN	9	=	/BWR;	PIN	12	= /MODE_STRB ;

{ ----- DECLARATIONS AND INTERMEDIATE VARIABLES ----- }

FIELD_ADDR = [MA12..MA8] ;
THIS_BLOCK = BLK_1_ ;

{ ----- }

{ ----- EQUATIONS ----- }

WR_RAM = THIS_BLOCK * RAM_REG * BWR ;

GN_STB1 = THIS_BLOCK * /RAM_REG * BWR * (ADDR:6100
+ ADDR:6200) ;

GN_STB2 = THIS_BLOCK * /RAM_REG * BWR
* (ADDR:6300 + ADDR:6400) ;

B_A = ADDR:6200 + ADDR:6400 ;

CF_STRB = THIS_BLOCK * /RAM_REG * BWR * ADDR:6500 ;

ADDR_BUF = THIS_BLOCK * RAM_REG ;

MODE_STRB = THIS_BLOCK * BWR * /RAM_REG * ADDR:7800 ;

U91 - UCI2

PARTNO	3270233-00;
DATE	1/26/89;
REVISION	4;
DESIGNER	JCM;
COMPANY	CTI;
ASSEMBLY	AN. SUB. MICROCONTROLLER INTERFACE DECODER/SEQUENCER ;
LOCATION	POS/ENERGY U13, U39, U65, U91;
DEVICE	P22V10 ;

SINCLUDE PALASM.OPR

ANALOG SUBSECTION uC INTERFACE PAL #2

Second of two pals used to perform the uC interface from the uC to the Analog Subsection. This pal enables the buffered data bus onto the Analog Subsection data bus for writing and reading RAM data and forcing time and position data. It also provides the strobes to force time and position data.

This pal also is used as a sequencer for the Pseudo Activity mode. In this mode it sequences addresses into the RAM address bus. These addresses sequentially select data stored in the RAM. This data is then forced into the time and position latches producing apparant data without activity. this mode would be used for testing the data collection and processing hardware (i.e. coincidence board, RTS,....).

```

{ ----- }
{           PIN DEFINITIONS           }
{ ----- }
{           INPUTS           :           OUTPUTS           }
{ ----- }
PIN      1      =      CLK_32      ;           PIN      23      =      /EN_DATA      ;
PIN      2      =      MA12      ;           PIN      22      =      /FRC_TIM      ;
PIN      3      =      MA11      ;           PIN      21      =      /FRC_PE      ;
PIN      4      =      MA10      ;           PIN      20      =      /OE_RAM      ;
PIN      5      =      RAM_REG      ;           PIN      19      =      /TIM_BUF      ;
PIN      6      =      RUN_CTR      ;           PIN      18      =      ASA0      ;
PIN      7      =      CTR_CLOCK      ;           PIN      17      =      Q2      ;
PIN      8      =      BLK_1_      ;           PIN      16      =      Q1      ;
PIN      9      =      /BRD      ;           PIN      15      =      Q0      ;
PIN     10      =      /BWR      ;           PIN      14      =      OERAM_OE      ;
PIN     11      =      CLK_LD      ;
{ ----- }

```

```

{----- DECLARATIONS AND INTERMEDIATE VARIABLES -----}

```

```

CTR_CLK      =      /CTR_CLOCK      ;
FIELD_ADDR   =      [MA12..MA10]      ;
FIELD_CTRVAL =      [Q2..Q0]      ;
E_R          =      CTR_CLK * CLK_LD * CTRVAL:0
              + CTRVAL:[1..7]      ;
THIS_BLOCK   =      BLK_1_      ;

```

```

{ ----- }

```

```

{ ----- EQUATIONS ----- }

```

{EN_DATA output enables the data transciever (U101). It goes active when forcing data to the time and position latches and the RAM. Notice that it is mapped up to 77FF. This is so that the processor does not drive the ASD_ bus when writing to the ASMODE latch. This could pose a contention problem because if writing a INIT_SEQ, the ASSEQ sequencer could drive the bus before this PAL has released it.....WHAM!!! }

```

EN_DATA      =      THIS_BLOCK * RUN_CTR
                  * (RAM_REG + (/RAM_REG * ADDR:[6000..77FF]))      ;

FRC_TIM.D    =      (RUN_CTR * THIS_BLOCK * /RAM_REG * BWR * ADDR:7000)
                  + (/RUN_CTR * (CTR_CLK * CLK_LD * CTRVAL:'d'0
                  + CTRVAL:['d'1..'d'2]))      ;

```

```

FRC_PE.D      =      (RUN_CTR * THIS_BLOCK * /RAM_REG * BWR * ADDR:7400)
                   + /RUN_CTR * CTRVAL:['d'1..'d'4]
                                                         ;

OE_RAM.D      =      THIS_BLOCK * RAM_REG * BRD
                   + /RUN_CTR
                   * ( CTR_CLK * CLK_LD * CTRVAL:'d'0
                   + CTRVAL:['d'1..'d'6] )
                                                         ;

OE_RAM.OE     =      OERAM_OE
                                                         ;

TIM_BUF.D     =      THIS_BLOCK * /RAM_REG * ADDR:7000
                   + /RUN_CTR
                                                         ;

ASA0.D        =      /RUN_CTR * CTRVAL:['d'3..'d'6]
                                                         ;

ASA0.OE       =      /RUN_CTR
                                                         ;

Q0.D          =      /Q0 * E_R
                                                         ;

Q1.D          =      (Q1 * /Q0 + /Q1 * Q0) * E_R
                                                         ;

Q2.D          =      (Q2 * /Q1 + Q2 * /Q0 + /Q2 * Q1 * Q0) * E_R
                                                         ;

{OERAM_OE had to be created because the .OE has only one product term.
We can live with the accumulated delay via this method }

OERAM_OE      =      RAM_REG * THIS_BLOCK + /RUN_CTR
                                                         ;

[EN_DATA,FRC_TIM,FRC_PE,OE_RAM,TIM_BUF,ASA0,Q2,Q1,Q0].sp = 'b'0
                                                         ;

[EN_DATA,FRC_TIM,FRC_PE,OE_RAM,TIM_BUF,ASA0,Q2,Q1,Q0].ar = 'b'0
                                                         ;

```

U10 - CE_SEQ1

PARTNO 3270240-00;
 DATE 2/14/89;
 REVISION 3;
 DESIGNER JCM;
 COMPANY CTI;
 ASSEMBLY COMMON ELECTRONICS SEQUENCER;
 LOCATION BKT CONTROLLER U10;
 DEVICE P22V10;

```
{
  -----
  {
    COMMON ELECTRONICS SEQUENCER
  }
  {
    -----
  }
  {
    This pal samples the EVNT lines from the four Analog
    and sequences the removal of the time and position data.
    It also resets the status flip flops on the appropriate
    analog subsection.
  }
  {
  }
  {
  }
  {
  }
  {
  }
  {
  }
  {
    -----
  }
  {
    PIN DEFINITIONS
  }
  {
    -----
  }
  {
    INPUTS      :      OUTPUTS
  }
  {
    -----
  }
```

PIN	1	= CLK_32	;	PIN	21	=	BS_Q0	;
PIN	2	= CLK_LD	;	PIN	22	=	BS_Q1	;
PIN	3	= RQSTA	;	PIN	23	=	BS_Q2	;
PIN	4	= RQSTB	;	PIN	20	=	TMP_STRB	;
PIN	5	= RQSTC	;	PIN	19	=	PSP_STRB	;
PIN	6	= RQSTD	;	PIN	18	=	GET_TIME	;
PIN	7	= HOLD_UCD	;	PIN	17	=	GET_POS	;
PIN	8	= EVNTA	;	PIN	16	=	!UCD_STRB	;
PIN	9	= EVNTB	;	PIN	15	=	ANSBRST	;
PIN	10	= EVNTC	;					
PIN	11	= EVNTD	;					

```

{ ----- }
{ ----- DECLARATIONS AND INTERMEDIATE VARIABLES ----- }
{ ----- }

```

```
FIELD_COUNT = [BS_Q2,BS_Q1,BS_Q0];
```

```

$define    S0      'b'000
$define    S1      'b'001
$define    S2      'b'010
$define    S3      'b'011
$define    S4      'b'100
$define    S5      'b'101
$define    S6      'b'110
$define    S7      'b'111

```

```
ITS_A_REQUEST    =    ( RQSTA # RQSTB # RQSTC # RQSTD )    ;
```

```

START_SEQ        =    CLK_LD & ( EVNTA # EVNTB # EVNTC # EVNTD
                        # RQSTA # RQSTB # RQSTC # RQSTD)    ;

```

```

{ ----- }
{ ----- EQUATIONS ----- }
{ ----- }

```

```

SEQUENCE    COUNT {
PRESENT      S0
              IF      START_SEQ      NEXT S1;
              IF      !START_SEQ      NEXT S0;
PRESENT      S1
              IF ITS_A_REQUEST      NEXT S2
                                OUT    TMP_STRB
                                OUT    GET_TIME;
              IF !(ITS_A_REQUEST)    NEXT S2,
PRESENT      S2
              IF ITS_A_REQUEST      NEXT S3
                                OUT    GET_TIME;
              IF !(ITS_A_REQUEST)    NEXT S3;
PRESENT      S3
                                NEXT S4;
PRESENT      S4
              IF ITS_A_REQUEST      NEXT S5
                                OUT    PSP_STRB
                                OUT    GET_POS;
              IF !(ITS_A_REQUEST)    NEXT S5;
PRESENT      S5
              IF (ITS_A_REQUEST & !HOLD_UCD) NEXT S6
                                OUT    UCD_STRB
                                OUT    GET_POS
                                OUT    ANSBRST;
              IF (ITS_A_REQUEST & HOLD_UCD) NEXT S6
                                OUT    GET_POS
                                OUT    ANSBRST;
              IF !(ITS_A_REQUEST)    NEXT S6
                                OUT    ANSBRST;
PRESENT      S6
                                NEXT S7;
PRESENT      S7
                                NEXT S0;
}

```

```

[BS_Q0, BS_Q1,BS_Q2,TMP_STRB,PSP_STRB,GET_TIME,GET_POS,
UCD_STRB,ANSBRST].ar = 'b'0;

```

```

[BS_Q0, BS_Q1,BS_Q2,TMP_STRB,PSP_STRB,GET_TIME,GET_POS,
UCD_STRB,ANSBRST].sp = 'b'0;

```


U14 - MLT_SEL

```

PARTNO      3270241-00;
DATE        2/14/89;
REVISION    1;
DESIGNER    JCM;
COMPANY     CTI;
ASSEMBLY    MULTIPLE SELECTOR SEQUENCER;
LOCATION     BKT CONTROLLER U14;
DEVICE      G16V8 ;
  
```

\$INCLUDE PALASM.OPR

```

{ ----- }
{           MULTIPLE SELECTOR SEQUENCER           }
{ ----- }
  
```

```

{
{   This pal selects the proper analog subsection time and position
{   latches when processing an event.  It also arbitrates between
{   two competing analog subsections when a double occurs.  This is
{   done by looking at state of a 512 ns clock when the multiple
{   occurs.  If the 512 ns clock is high, then select one block,
{   else select the other.  If more than two blocks are active in
{   the same time slot, the data is rejected from both.
{   This pal also determines which block the data is available
{   from, placing this data into the detector word.
{ ----- }
  
```

```

{ ----- }
{           PIN DEFINITIONS           }
{ ----- }
{   INPUTS      :      OUTPUTS      }
{ ----- }
  
```

PIN	1	= CLK_512	;				
PIN	2	= QRQSTA	;	PIN	13	=	EN_BLKA;
PIN	3	= QRQSTB	;	PIN	18	=	EN_BLKB;
PIN	4	= QRQSTC	;	PIN	17	=	EN_BLC;
PIN	5	= QRQSTD	;	PIN	16	=	EN_BLKD;
PIN	6	= MODE_3	;	PIN	15	=	HIT;
				PIN	14	=	BLK_0;
				PIN	19	=	BLK_1;

```

{ ----- }
{ ----- DECLARATIONS AND INTERMEDIATE VARIABLES ----- }
{ ----- }
ENABLE_MULT      =      MODE_3;
{ ----- }
{ ----- EQUATIONS ----- }
{ ----- }

```

```

EN_BLK_A      =      ( QRQSTA * /QRQSTB * /QRQSTC * /QRQSTD )
                    + ENABLE_MULT *
                    ( QRQSTA * QRQSTB * /QRQSTC * /QRQSTD * /CLK_512

                    + QRQSTA * /QRQSTB * QRQSTC * /QRQSTD * /CLK_512
                    + QRQSTA * /QRQSTB * /QRQSTC * QRQSTD * /CLK_512

```

```

EN_BLK_B      =      ( /QRQSTA * QRQSTB * /QRQSTC * /QRQSTD )
                    + ENABLE_MULT *
                    ( QRQSTA * QRQSTB * /QRQSTC * /QRQSTD * CLK_512

                    + /QRQSTA * QRQSTB * QRQSTC * /QRQSTD * /CLK_512
                    + /QRQSTA * QRQSTB * /QRQSTC * QRQSTD * /CLK_512

```

```

EN_BLK_C      =      ( /QRQSTA * /QRQSTB * QRQSTC * /QRQSTD )
                    + ENABLE_MULT *
                    ( QRQSTA * /QRQSTB * QRQSTC * /QRQSTD * CLK_512

                    + /QRQSTA * QRQSTB * QRQSTC * /QRQSTD * CLK_512
                    + /QRQSTA * /QRQSTB * QRQSTC * QRQSTD * /CLK_512

```

```

EN_BLK_D      =      ( /QRQSTA * /QRQSTB * /QRQSTC * QRQSTD )
                    + ENABLE_MULT *
                    ( QRQSTA * /QRQSTB * /QRQSTC * QRQSTD * CLK_512

                    + /QRQSTA * QRQSTB * /QRQSTC * QRQSTD * CLK_512
                    + /QRQSTA * /QRQSTB * QRQSTC * QRQSTD * CLK_512

```

```

HIT           =      EN_BLK_A + EN_BLK_B + EN_BLK_C + EN_BLK_D

```

```

BLK_0         =      EN_BLK_B + EN_BLK_D

```

```

BLK_1         =      EN_BLK_C + EN_BLK_D

```

U20 - SWAP_POS

```

PARTNO      3270242-00;
DATE        2/13/89;
REVISION    2;
DESIGNER    JCM;
COMPANY     CTI;
ASSEMBLY    POSITION SWAP PAL;
LOCATION      BKT CONTROLLER U20;
DEVICE      P16L8 ;
  
```

```
$INCLUDE PALASM.OPR
```

```

{ ----- }
{          POSITION SWAP PAL          }
{ ----- }
{   This pal takes care of rolling the position data, required   }
{   for the second ring of buckets on a gantry.  When mode5 is   }
{   high, the data is passed straight through the pal, while if }
{   mode5 is low, the position and plane bits are swapped end-to- }
{   end.                                                         }
{   }
{   }
{   }
{ ----- }
{          PIN DEFINITIONS          }
{ ----- }
{          INPUTS          :          OUTPUTS          }
{ ----- }

PIN    1    =    POS0      ;          PIN    19    =    POS0_      ;
PIN    2    =    POS1      ;          PIN    18    =    POS1_      ;
PIN    3    =    POS2      ;          PIN    17    =    POS2_      ;
PIN    4    =    BLK_0     ;          PIN    16    =    POS3_      ;
PIN    5    =    BLK_1     ;          PIN    15    =    POS4_      ;
PIN    6    =    POS5      ;          PIN    14    =    PLANE2_     ;
PIN    7    =    POS4      ;          PIN    13    =    PLANE0_     ;
PIN    8    =    POS3      ;          PIN    12    =    PLANE1_     ;
PIN    9    =    MODE5     ;
{ ----- }
  
```

{ ----- EQUATIONS ----- }

$$\text{POS0_} = \text{POS0} * \text{MODE5} \\ + \text{/POS0} * \text{/MODE5}$$

$$\text{POS1_} = \text{POS1} * \text{MODE5} \\ + \text{/POS1} * \text{/MODE5}$$

$$\text{POS2_} = \text{POS2} * \text{MODE5} \\ + \text{/POS2} * \text{/MODE5}$$

$$\text{POS3_} = \text{BLK_0} * \text{MODE5} \\ + \text{/BLK_0} * \text{/MODE5}$$

$$\text{POS4_} = \text{BLK_1} * \text{MODE5} \\ + \text{/BLK_1} * \text{/MODE5}$$

$$\text{PLANE2_} = \text{POS5} * \text{MODE5} \\ + \text{/POS5} * \text{/MODE5}$$

$$\text{PLANE0_} = \text{POS3} * \text{MODE5} \\ + \text{/POS3} * \text{/MODE5}$$

$$\text{PLANE1_} = \text{POS4} * \text{MODE5} \\ + \text{/POS4} * \text{/MODE5}$$

U26 - CTRMUX

PARTNO 3270244-00;
 DATE 2/13/89;
 REVISION 1;
 DESIGNER JCM;
 COMPANY CTI;
 ASSEMBLY COUNTER MUX/LIVE-TIME CORRECTION CIRCUITRY;
 LOCATION BKT CONTROLLER U26;
 DEVICE P16L8 ;

SINCLUDE PALASM.OPR

```
{ ----- }
{ COUNTER MULTIPLEXER }
{ ----- }
{ This pal works in concert with the pal (XTDLIV) to create }
{ countable pulses from the logic signals coming from the analog }
{ subsections. This pal also performs the combinatorial logic }
{ for the singles counter. Note that this logic along with the }
{ 74ACT86 and the 74ACT74 produce only a transition for each }
{ countable event. This is due to the Timer2 requirements of the }
{ 80C196. It counts edges (either rising or falling), not pulses. }
{ }
{ }
{ ----- }
{ PIN DEFINITIONS }
{ ----- }
{ INPUTS : OUTPUTS }
{ ----- }
```

```
PIN 1 = ALL_PU ;
PIN 2 = ALL_EVTS ;
PIN 3 = RQSTA ;
PIN 4 = RQSTB ;
PIN 5 = RQSTC ;
PIN 6 = RQSTD ;
PIN 7 = BSQ0 ;
PIN 8 = BSQ1 ;
PIN 9 = BSQ2 ;
PIN 11 = MODE1 ;
PIN 13 = MODE2 ;
PIN 14 = LOW_EN_A;
PIN 15 = LOW_EN_B ;
PIN 16 = LOW_EN_C ;
PIN 17 = LOW_EN_D ;
PIN 18 = SINGLES ;
PIN 19 = MUX_EVTS_D ;
PIN 12 = /VALID ;

{ ----- }
```

{----- DECLARATIONS AND INTERMEDIATE VARIABLES ----- }

MODE_0 = /MODE2 * /MODE1;
MODE_1 = /MODE2 * MODE1;
MODE_2 = MODE2 * /MODE1;
MODE_3 = MODE2 * MODE1;

LOW_EN = RQSTA * /LOW_EN_A
 + RQSTB * /LOW_EN_B
 + RQSTC * /LOW_EN_C
 + RQSTD * /LOW_EN_D

RQST_EVTS = RQSTA + RQSTB + RQSTC + RQSTD

{ ----- }
{ ----- EQUATIONS ----- }
{ ----- }

VALID = RQSTA * LOW_EN_A
 + RQSTB * LOW_EN_B
 + RQSTC * LOW_EN_C
 + RQSTD * LOW_EN_D

MUX_EVTS_D = MODE_0 * ALL_EVTS
 + MODE_2 * LOW_EN
 + MODE_1 * ALL_PU
 + MODE_3 * RQST_EVTS

U38 - XTDLIV

PARTNO 3270246-00;
 DATE 2/9/89;
 REVISION 2;
 DESIGNER JCM;
 COMPANY CTI;
 ASSEMBLY EXTENDED LIVE TIME TIME STRETCHER;
 LOCATION BKT CONTROLLER U38;
 DEVICE P22V10 ;

```

{ ----- }
{           EXTENDED LIVE TIME GATED COUNTER           }
{ ----- }
{
{   This pal performs both gated counter function for the extended
{   live time measurement as well as basic combinatorial logic on
{   the different inputs to produce "LOW_EN" and "ALL_EVT" to go
{   to the counter mux (U23).
{
{   The gated counter ORs the four TIM_MRK signals together and de-
{   gates the counter during their active period. The Q3 of the
{   counter is used to feed the Timer/Counter. Its low and high
{   time (minimum) will be 256 ns (8 X 32 ns).
{
{ ----- }
{
{ ----- }
{           PIN DEFINITIONS
{ ----- }
{           INPUTS           :           OUTPUTS
{ ----- }
PIN    1    =    CLK_32      ;
PIN    2    =    TIM_MRKA    ;    PIN    23    =    Q1          ;
PIN    3    =    TIM_MRKB    ;    PIN    22    =    Q3          ;
PIN    4    =    TIM_MRKC    ;    PIN    21    =    ALL_PU     ;
PIN    5    =    TIM_MRKD    ;    PIN    20    =    ALL_EVT    ;
PIN    6    =    PILE_UPA     ;    PIN    19    =    Q2          ;
PIN    7    =    PILE_UPB     ;    PIN    18    =    EVNTD     ;
PIN    8    =    PILE_UPC     ;    PIN    17    =    EVNTC     ;
PIN    9    =    PILE_UPD     ;    PIN    16    =    EVNTB     ;
                                   PIN    15    =    EVNTA     ;
                                   PIN    14    =    Q0          ;
  
```

```

{ ----- }
{ ----- DECLARATIONS AND INTERMEDIATE VARIABLES ----- }
{ ----- }

```

```

FIELD_COUNT      =      [Q3..Q0];
BLOCK_COUNT      =      TIM_MRKA # TIM_MRKB # TIM_MRKC # TIM_MRKD;

```

```

$DEFINE      S0      'B'0000
$DEFINE      S1      'B'0001
$DEFINE      S2      'B'0010
$DEFINE      S3      'B'0011
$DEFINE      S4      'B'0100
$DEFINE      S5      'B'0101
$DEFINE      S6      'B'0110
$DEFINE      S7      'B'0111
$DEFINE      S8      'B'1000
$DEFINE      S9      'B'1001
$DEFINE      S10     'B'1010
$DEFINE      S11     'B'1011
$DEFINE      S12     'B'1100
$DEFINE      S13     'B'1101
$DEFINE      S14     'B'1110
$DEFINE      S15     'B'1111

```

```

{ ----- }

```

```

{ ----- EQUATIONS ----- }

```

```

{ ***** GATED COUNTER ***** }

```

```

SEQUENCE COUNT {

```

```

PRESENT S0
    IF      !BLOCK_COUNT      NEXT S1;
    IF      BLOCK_COUNT      NEXT S0;

```

```

PRESENT S1
    IF      !BLOCK_COUNT      NEXT S2;
    IF      BLOCK_COUNT      NEXT S1;

```

```

PRESENT S2
    IF      !BLOCK_COUNT      NEXT S3;
    IF      BLOCK_COUNT      NEXT S2;

```

```

PRESENT S3
    IF      !BLOCK_COUNT      NEXT S4;
    IF      BLOCK_COUNT      NEXT S3;

```


PRESENT S4		
IF	!BLOCK_COUNT	NEXT S5;
IF	BLOCK_COUNT	NEXT S4;
PRESENT S5		
IF	!BLOCK_COUNT	NEXT S6;
IF	BLOCK_COUNT	NEXT S5;
PRESENT S6		
IF	!BLOCK_COUNT	NEXT S7;
IF	BLOCK_COUNT	NEXT S6;
PRESENT S7		
IF	!BLOCK_COUNT	NEXT S8;
IF	BLOCK_COUNT	NEXT S7;
PRESENT S8		
IF	!BLOCK_COUNT	NEXT S9;
IF	BLOCK_COUNT	NEXT S8;
PRESENT S9		
IF	!BLOCK_COUNT	NEXT S10;
IF	BLOCK_COUNT	NEXT S9;
PRESENT S10		
IF	!BLOCK_COUNT	NEXT S11;
IF	BLOCK_COUNT	NEXT S10;
PRESENT S11		
IF	!BLOCK_COUNT	NEXT S12;
IF	BLOCK_COUNT	NEXT S11;
PRESENT S12		
IF	!BLOCK_COUNT	NEXT S13;
IF	BLOCK_COUNT	NEXT S12;
PRESENT S13		
IF	!BLOCK_COUNT	NEXT S14;
IF	BLOCK_COUNT	NEXT S13;
PRESENT S14		
IF	!BLOCK_COUNT	NEXT S15;
IF	BLOCK_COUNT	NEXT S14;
PRESENT S15		
IF	!BLOCK_COUNT	NEXT S0 ;
IF	BLOCK_COUNT	NEXT S15;
}		

```

{ ***** ALL_PU and ALL_EVT ***** }

ALL_PU      =      PILE_UPA # PILE_UPB # PILE_UPC # PILE_UPD      ;
ALL_EVT      =      EVNTA # EVNTB # EVNTC # EVNTD      ;

[Q3,Q2,Q1,Q0,ALL_PU,ALL_EVT].sp = 'b'0      ;
[Q3,Q2,Q1,Q0,ALL_PU,ALL_EVT].ar = 'b'0      ;

```

U25 - BC1

PARTNO 3270243-00;
 DATE 1/20/89;
 REVISION 4;
 DESIGNER JCM;
 COMPANY CTI;
 ASSEMBLY BUCKET CONTROLLER;
 LOCATION BKT CONTROLLER U25;
 DEVICE P16L8 ;

\$INCLUDE PALASM.OPR

```
{
{-----}
{          BLOCK CONTROLLER DECODER PAL          }
{-----}
{ One of two PALs that perform decoding for the 80c196 processor. }
{ This PAL performs address decoding for the Time and Position }
{ capture latches, the DIPSWITCH buffer, and the MODE latch.   }
{
{ Refer to drawing titled:
{                               BLOCK CONTROLLER
{                               BLOCK PROCESSOR/CONTROLLER 1
{
{-----}
{          PIN DEFINITIONS          }
{-----}
{          INPUTS          :          OUTPUTS          }
{-----}
```

PIN	1	=	P1_7	;			
PIN	2	=	P1_6	;	PIN	19	= /RD_TIM ;
PIN	3	=	P1_5	;	PIN	18	= /RD_POS ;
PIN	4	=	A15	;	PIN	17	= /RD_ADDR ;
PIN	5	=	A14	;	PIN	16	= /MD_STRB ;
PIN	6	=	A13	;	PIN	15	= /BUF ;
PIN	7	=	A12	;			
PIN	8	=	A11	;			
PIN	9	=	A1	;			
PIN	11	=	A0	;			
PIN	13	=	/RD	;			
PIN	14	=	/WR	;			

```

{ ----- }
{-----  DECLARATIONS AND INTERMEDIATE VARIABLES  -----}
{ ----- }

```

```

FIELD_ADDR      =      [A15..A11,A1,A0];
ANSUB1           =      /P1_7 * /P1_6 * /P1_5 * ADDR:[6000..7FFF];
ANSUB2           =      /P1_7 * /P1_6 * P1_5 * ADDR:[6000..7FFF];
ANSUB3           =      /P1_7 * P1_6 * /P1_5 * ADDR:[6000..7FFF];
ANSUB4           =      /P1_7 * P1_6 * P1_5 * ADDR:[6000..7FFF];
REGISTERS        =      P1_7 * /P1_6 * /P1_5;
TIMER_CTR       =      REGISTERS * ADDR:[7000..7003];
BOOTSTRAP        =      ADDR:[2000..3FFF];
TIME_REGISTER    =      REGISTERS * ADDR:6000;
POS_REGISTER     =      REGISTERS * ADDR:6001;
ADDRESS_BUFFER   =      REGISTERS * ADDR:6002;
MODE_LATCH       =      REGISTERS * ADDR:6003;

```

```

{ ----- }
{-----  EQUATIONS  -----}

```

```

RD_TIM           =      RD * TIME_REGISTER

RD_POS           =      RD * POS_REGISTER

RD_ADDR          =      RD * ADDRESS_BUFFER

MD_STRB          =      WR * MODE_LATCH

BUF              =      TIMER_CTR
                      + BOOTSTRAP
                      + ANSUB1
                      + ANSUB2
                      + ANSUB3
                      + ANSUB4
                      + TIME_REGISTER
                      + POS_REGISTER

```

U37 - BC2

PARTNO 3270245-00;
 DATE 1/23/89;
 REVISION 2;
 DESIGNER JCM;
 COMPANY CTI;
 ASSEMBLY BUCKET CONTROLLER;
 LOCATION BKT CONTROLLER U37;
 DEVICE P22V10 ;

\$INCLUDE PALASM.OPR

```
{
  -----
  {
    BLOCK CONTROLLER DECODER PAL
  }
  {
    -----
  }
  {
    Second of two PALs that perform decoding for the 80c196 uC.
  }
  {
    This PAL performs address decoding for the four blocks,
  }
  {
    timer/counter, uC ram, eeprom, and bootstrap eeprom.
  }
  {
    -----
  }
  {
    Refer to drawing titled:
  }
  {
    BLOCK CONTROLLER
  }
  {
    BLOCK PROCESSOR/CONTROLLER 1
  }
  {
    -----
  }
}
```

```
{
  PIN DEFINITIONS
  {
    -----
  }
  {
    INPUTS          :          OUTPUTS
  }
  {
    -----
  }
}
```

PIN	1	=	P1_7	;			
PIN	2	=	P1_6	;	PIN	23	= BLK_1A ;
PIN	3	=	P1_5	;	PIN	22	= BLK_1B ;
PIN	4	=	A15	;	PIN	21	= BLK_1C ;
PIN	5	=	A14	;	PIN	20	= BLK_1D ;
PIN	6	=	A13	;	PIN	19	= /TMR_CTR ;
PIN	7	=	A12	;	PIN	18	= /BTSTRAP ;
PIN	8	=	A11	;	PIN	17	= /HST_RAM ;
PIN	9	=	A1	;	PIN	16	= /SCRTCH_RAM ;
PIN	10	=	A0	;	PIN	15	= /EEPROM ;
					PIN	14	= /READY ;

```

{ ----- }
{ ----- DECLARATIONS AND INTERMEDIATE VARIABLES ----- }
{ ----- }

```

```

FIELD_ADDR      = [A15..A11,A1,A0];
ANSUB1          = /P1_7 * /P1_6 * /P1_5 * ADDR:[6000..7FFF] ;
ANSUB2          = /P1_7 * /P1_6 * P1_5 * ADDR:[6000..7FFF] ;
ANSUB3          = /P1_7 * P1_6 * /P1_5 * ADDR:[6000..7FFF] ;
ANSUB4          = /P1_7 * P1_6 * P1_5 * ADDR:[6000..7FFF] ;
REGISTERS       = P1_7 * /P1_6 * /P1_5 * ADDR:[6000..7FFF] ;
TIMER_CTR       = REGISTERS * ADDR:[7000..7003] ;
BOOTSTRAP       = ADDR:[2000..3FFF] ;
HIST_RAM        = ADDR:[4000..5FFF] ;
SCRATCH_RAM     = ADDR:[0000..1FFF] ;
CODE_EEPROM     = ADDR:[8000..FFFF] ;

```

```

{ ----- }
{ ----- EQUATIONS ----- }
{ ----- }

```

```

BLK_1A          = ANSUB1 ;
BLK_1B          = ANSUB2 ;
BLK_1C          = ANSUB3 ;
BLK_1D          = ANSUB4 ;
TMR_CTR         = TIMER_CTR ;
BTSTRAP         = BOOTSTRAP ;
HST_RAM         = HIST_RAM ;
SCRTCH_RAM      = SCRATCH_RAM ;
EEPROM          = CODE_EEPROM ;
READY           = ANSUB1 {Only the rams are fast enough for }
                  + ANSUB2 { no wait states. }
                  + ANSUB3
                  + ANSUB4
                  + REGISTERS
                  + TIMER_CTR
                  + BOOTSTRAP
                  + CODE_EEPROM ;

```

U6 - CTRCLK

PARTNO 3270239-00;
 DATE 2/9/89;
 REVISION 1;
 DESIGNER JCM;
 COMPANY CTI;
 ASSEMBLY PSEUDO ACTIVITY COUNTER CLOCK GENERATOR;
 LOCATION BKT CONTROLLER U6;
 DEVICE P22V10 ;

```

{ ----- }
{ PSEUDO ACTIVITY COUNTER CLOCK GENERATOR }
{ ----- }
{ }
{ This pal divides pulses created by the Timer/Counter (8254) }
{ and generates discrete pulses for the four Analog Subsections. }
{ These pulses are used to cause the specific Analog Subsection }
{ to output Pseudo time and position data from its lookup RAM. }
{ The Pseudo Activity rate is set by the modulo-n divisor loaded }
{ into the Timer/Counter. Each pulse from Timer/Counter pulses }
{ the subsequent "CTR_CLK" line out of this pal. }
{ ----- }
{ }
{ CLK_256 _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ }
{ }
{ OUT1  _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ }
{ }
{ CTR_CKA _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ }
{ }
{ CTR_CKB _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ }
{ }
{ CTR_CKC _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ }
{ }
{ CTR_CKD _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ }
{ }
{ ----- }
{ PIN DEFINITIONS }
{ ----- }
{ INPUTS : OUTPUTS }
{ ----- }
PIN 1 = CLK_256 ; PIN 23 = !Q0 ;
PIN 2 = !OUT1 ; PIN 22 = !Q1 ;
PIN 21 = !Q2 ;
PIN 20 = !Q3 ;
PIN 19 = !Q4 ;
PIN 18 = !Q5 ;
  
```

```

{ ----- DECLARATIONS AND INTERMEDIATE VARIABLES ----- }

FIELD    state      =      [Q5,Q4,Q3,Q2,Q1,Q0]      ;

$define   S0        'b'000000
$define   S1        'b'000001
$define   S2        'b'010000
$define   S3        'b'010010
$define   S4        'b'110000
$define   S5        'b'110100
$define   S6        'b'100000
$define   S7        'b'101000

{ ----- EQUATIONS ----- }

sequence state {

present S0
            if      OUT1      next S1;
            if      !OUT1     next S0;

present S1
                        next S2;

present S2
            if      OUT1      next S3;
            if      !OUT1     next S2;

present S3
                        next S4;

present S4
            if      OUT1      next S5;
            if      !OUT1     next S4;

present S5
                        next S6;

present S6
            if      OUT1      next S7;
            if      !OUT1     next S6;

present S7
                        next S0;
}

[Q5,Q4,Q3,Q2,Q1,Q0].sp = 'b'0      ;

[Q5,Q4,Q3,Q2,Q1,Q0].ar = 'b'0      ;

```


VITA

John Clifton Moyers, Jr. (Clif) was born in Oak Ridge, Tennessee in April, 1960, and graduated from Oak Ridge High School in 1978. Having a keen interest in science, he entered the University of Tennessee College of Engineering, specializing in the Analog Option of the Electrical Engineering department curriculum. Upon completion of the coursework and seven Cooperative Education assignments with the Instruments and Controls division of Oak Ridge National Laboratory, he graduated with honors in 1983. While working full-time (40 hours per week) for Control Technology Inc. (CTI) and part-time (10 hours per week) for the Electrical Engineering department, Clif entered the Electrical Engineering graduate program in the Fall of 1983. The fluidity of his pursuit of this degree was interrupted after having satisfactorily completed 18 quarter-hours of coursework when his grades suffered due to the necessity of spending a great deal of time on a demanding project at CTI. In Fall, 1988, after receiving much encouragement from his new employer, Computer Technology and Imaging (CTI, also), Inc., Clif reentered the Graduate curriculum. In Spring, 1990, six and a half years after starting the work toward this degree, he finished his coursework and thesis. He is presently involved with development of the next generation Bucket electronics and scanner at CTI, Inc.