Fermi National Accelerator Laboratory

FERMILAB-Conf-93/309

The Simulation of a Data Acquisition System for a Proposed High Resolution PET Scanner

C. Rotolo, M. Larwill and S. Chappa

Fermi National Accelerator Laboratory
P.O. Box 500, Batavia, Illinois 60510

C. Ordonez

The University of Chicago
Chicago, Illinois

October 1993

To be published in the Conference Record of the *IEEE Medical Imaging Conference,*
San Francisco, California, November 4-6, 1993 and the Transactions on Medical Imaging

Operated by Universities Research Association Inc. under Contract No. DE-AC02-76CHO3000 with the United States Department of Energy

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

The Simulation of a Data Acquisition System for a Proposed High Resolution PET Scanner

C. Rotolo*, C. Ordonez†, M. Larwill*, S. Chappa*

*Fermi National Accelerator Laboratory, †The University of Chicago

Abstract

The simulation of a specific data acquisition (DAQ) system architecture for a proposed high resolution Positron Emission Tomography (PET) scanner is discussed. Stochastic processes are used extensively to model PET scanner signal timing and probable DAQ circuit limitations. Certain architectural parameters, along with stochastic parameters, are varied to quantitatively study the resulting output under various conditions. The inclusion of the DAQ in the model represents a novel method of more complete simulations of tomograph designs, and could prove to be of pivotal importance in the optimization of such designs.

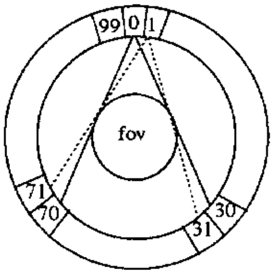
I. INTRODUCTION

The DAQ system under study was designed for possible use with a proposed PET scanner for the University of Chicago. Design of the scanner and DAQ is presently on hold due to funding uncertainties and the desire to explore recent technological developments. This paper will focus only on the simulation of the primary DAQ system architecture. Although several other aspects of the DAQ system including control, diagnostics, attenuation scan filtering, and accidental rate measurements were briefly considered, none of them will be addressed in this paper.

Soon after the DAQ architecture was proposed, it became apparent that due to the inherent stochastic processes, a simulation was needed to study its behavior. The simulation is written using the high level Verilog hardware descriptive language using mostly behavioral models rather than structural constructs. This enables easy modifications to DAQ characteristics within the context of probable hardware limitations without the restraint of dealing with a specific hardware configuration. However, the ultimate goal of the design process is to replace these behavioral models with models of "real" parts in order to migrate toward pc board design and simulation in a Cadence CAE environment.

II. SCANNER CHARACTERISTICS

Figure (1) shows the geometry of the proposed scanner which has 2400 detectors that are arranged in 4 rings of 600 each. The detectors are divided into 100 banks around the scanner each containing 24 detectors. The scanner's field of view (fov) is limited, as shown in Figure (1), by searching for time coincidence of events from bank_n with events only from banks n+30 through n+70. Of the resulting 4100 possible coincident pairs, 2050 pairs are unique and are represented by the shaded area marked "A" in Figure (2).



2400 detectors

100 banks
24 detectors/bank

4 rings
600 detectors/ring

Fig. 1 - PET Scanner Bank Structure and Field of View

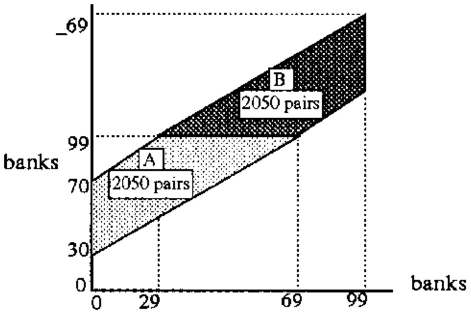


Fig. 2 - Bank Pair Combinations

III. THE DAQ ARCHITECTURE

The DAQ system consists of 100 front-end modules (fe_0 to fe_99) that correspond to the 100 banks. Each front end module outputs a strobe pulse, ds[n], which is carefully timed to the event, and a 5 bit detector address, dda_n, that identifies which of the 24 detectors in the bank detected the event. Figure (3) is a block diagram of the DAQ simulation structure but to a certain extent, it parallels the DAQ system architecture. The front_end in Figure (3) is shown as a single module driving 100 instances of a module called sbuf. The sbuf module is a part of the simulation only for the purpose of introducing timing skews and jitter to s[n] and da_n. Its outputs, ds[n] and dda_n, simulate the actual outputs that would be expected from a real front end module. Each of the 100 bank circuits, identified as bank_n, receives ds[n] and dda_n from its corresponding front end module plus 41 other strobe pulses, ds[n+30:n+70], from the other front end modules. This requires each strobe pulse to be fanned out to 42 bank circuits. Accurate timing distribution of the strobe pulses is by no means trivial and is a problem addressed in the simulation.

* Work supported by the United States Department of Energy under contract No. DE-AC02-76CHO3000.

Each bank circuit detects coincidence between ds[n] and any other strobe pulse in the range of ds[n+30] to ds[n+70]. As a result, 4100 bank pair combinations are compared throughout the DAQ and are represented by the areas "A" plus "B" in Figure (2) where every pair in "A" a has complementary pair in "B". Upon detecting a coincidence in bank_n, a busy_n signal is produced but it is unknown as to which of the 41 other possible banks that bank_n is coincident with. That information is determined when another bank circuit, having bank_n in its field of view, detects a coincidence and produces a second busy signal. Further processing continues when two appropriate busy signals are generated near simultaneously (t_b≈3 ns).

The 100 bank circuits are split into four groups (a-d) in which the bank circuit outputs have a common 12 bit data bus and a common busy line. Although the number of groups has a practical limit, the only functional limitation is that a group does not contain any banks that are within each other's field of view. When a bank circuit detects a coincidence, it produces a fixed-width (tbusy≈20 ns) busy pulse. This pulse is output to the group's busy line and is promptly received by all other bank circuits in the group within a maximum time called t_a (≈2.5 ns). If the group's busy line is not being asserted by another bank at the time a coincidence is detected, the bank circuit outputs a 12 bit data word. This word consists of a 7 bit bank address that corresponds to the bank circuit identifier (0-99) and the 5 bit detector address received from the front end. When a second coincidence occurs within tbusy and in any bank circuit of the same group, the data from the second coincidence is not output and thus rejected. However, the group busy line is then extended for an additional tbusy time. If the coincidence occurred within the time t_a before the second bank received the busy signal, data from the second bank would also be output to the group's data bus thus corrupting it by wire-ORing each bit. This would bias the output to higher numbered banks and higher detector addresses.

The data bus and busy line from each group are input to the computer interface (comp_int) module which processes the bank circuit data and forms a 24 bit word that is output to a histogramming computer. The busy signals from each group are OR-ed together to produce a signal called busy_sum. After a delay (t_b≈3 ns) from the leading edge of busy_sum, the status of all four busy signals are latched and become the input to a lookup table. Outputs of the lookup determine which two of the four group data buses form the output word. If other than two busy signals are latched, no data will be passed and the event will be rejected. If t_b is either too small or too large, the amount of data rejected will be more than necessary and the output rate will be reduced.

The width of the busy pulse is set to be longer than the minimum processing time of the input data acceptance logic in the computer interface module. As long as any of the group busy signals are asserted, busy_sum remains asserted which prevents further data transfers. Hence, the width of the busy pulse, tbusy, plus the decision time, t_b, determines the minimum time between successive output events and therefore the maximum throughput rate.

IV. SIMULATION OF DAQ ARCHITECTURE

A. General Description

The simulation is composed of seven Verilog modules and is represented graphically in Figure (3). Previously simulated PET data files are input to the front_end module which determines timing as to the interval between events and outputs the strobe pulse (s[n]) and the detector address data (da_n) for all banks. One-hundred instances of the module sbuf, corresponding to the 100 banks, introduce timing jitter and skew to each s[n] pulse and da_n data passing through it. The one-hundred instances of the module, called bank_ckt, select the appropriate set of strobe pulses and detector address data. Another two modules, called coinc_oran and monitor, are located within bank_ckt. Upon detecting a suitable coincidence, the bank_ckt modules output group busy and data information to a module called comp_int. The comp_int module determines and produces the output data from this information. Although data for the analysis of the architecture, which is discussed later, is obtained from within the simulation, the comp_int module also generates a data_xy output file for additional analysis outside of the simulation.

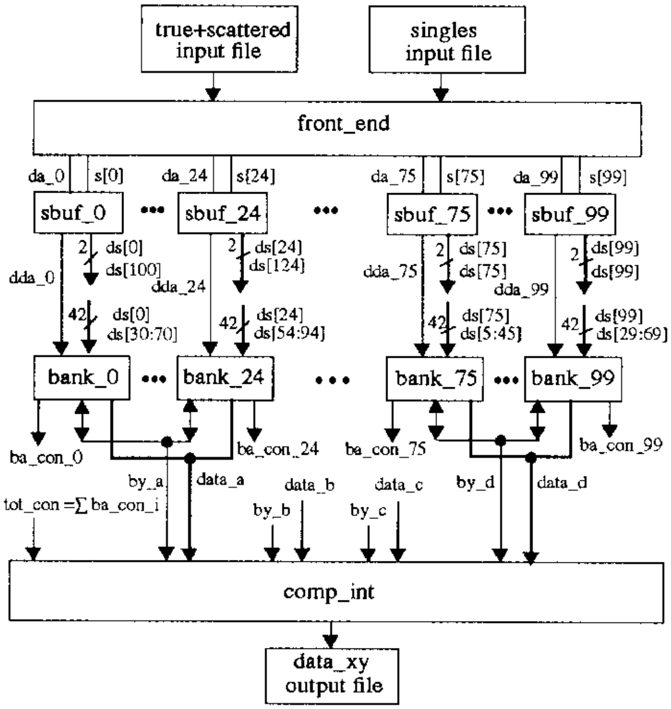


Fig. 3 - Block Diagram of Simulation Modules

B. Input Data Generation

True+scatter coincidences and singles event data were generated by Monte Carlo simulation of the given PET geometry for an activity of 1.5 mCi in a cylindrical phantom (20 cm in diameter and 13.2 cm long) and a threshold energy of 350 keV. The simulation has no time information. The expected average rates for true, scattered, and singles events were calculated analytically [1] [2] resulting in:

$$R_{true+scat} = R_t = 25.38 \text{ kHz} \tag{1}$$

$$R_{singles} = R_s = 656.79 \text{ kHz} \tag{2}$$

The resulting data from the simulation is placed in two files. One file contains singles event bank and detector addresses, and the other file contains pairs of bank and detector addresses for true+scattered events.

C. Pet_daq

Pet_daq is the top level module and provides the mechanism to interconnect the front_end module, 100 instances of sbuf, 100 instances of bank_ckt, and the comp_int module. It also provides the mechanism to implement the number of groups. Although the number of groups could be varied, the study under consideration is limited to four groups only. All parameters and random variable seeds are established in pet_daq and used as global variables in respective modules. Table (1) contains a listing of these parameters along with a brief description. The significance of these parameters will be discussed later.

Determinate	Values (ns)			
Parameters	Typ.	Ideal	Description	Module
ttru_sys	39,401	39,401	average tru+scat interval	front_end
tsing_sys	1,563	1,563	average singles interval	front_end
s_jitter_sd	2.5	0.0	strobe pulse, rms s[n] jitter	sbuf
fe_skew	1.0	0.0	front end strobe pulse skew	sbuf
coinc_skew	1.3	0.0	coincidence logic skew	coinc_oran
coinc_min_width	2.0	0.1	min. coincident overlap	coinc_oran
bank_ckt_skew	1.0	0.0	bank circuit logic skew	bank_ckt
t_a	2.5	1.0	group busy delay	bank_ckt
Indeterminate				
Parameters				
res_time	14.0	12.0	resolving time	front_end
t_b	3.0	3.0	coincident acceptance time	comp_int
tbusy	20.0	20.0	busy pulse width	bank_ckt

Table (1) - Simulation Parameters

D. Front_end

The front_end module reads the singles data file and the true+scatter data file to generate event input data for the simulation. Two statistically independent timing generators, called tru_gen and sing_gen, are used to generate timing intervals between successive true+scattered and single events, respectively. Assuming that event timing is a Poisson process, the first-order interarrival time, t_1, is an exponential random variable with the probability density function [3] [4] [5]

$$f_{t1}(t) = r \exp(-rt) \tag{3}$$

where r = the average event rate in events/sec.
t_1 = a continuous random variable defined as the interval of time between successive events.

To find the probability that the interval is $t > t_1$, we integrate Eq (3) from t_1 to infinity and obtain

$$P(t > t_1) = \exp(-rt_1) \tag{4}$$

Solving for t_1, we obtain

$$t_1 = -(1/r) \ln(P) \qquad \text{where } 0 \leq P \leq 1 \tag{5}$$

Successive timing intervals, t_1, in the range of 0 to infinity are obtained by using Eq (5) after generating a uniform random number in the range of 0 to 1 for P. Equations (1) and (2) are used to obtain r for tru_gen and sing_gen, respectively. As expected, the average interval obtained from this process was

$$t_{1avg} = 1/r \tag{6}$$

Each event is generated in such a manner that the interval between successive events of the same type can occur at intervals as small as 1 ns. However, the smallest interval between events involving the same bank is determined by the width of the strobe pulse which is set by the resolving time parameter called res_time. Whereas the ideal resolving time for the scanner is estimated at 12 ns, the typical value used in the simulation is 14 ns to compensate for the minimum detectable strobe pulse overlap, coinc_min_width. As each true+scattered event pair is generated, it is checked to determine if the banks involved are within each others field of view. If so, the event data is placed into a queue called tru_fov_que. The contents of this queue are checked to determine performance characteristics of the DAQ each time the comp_int module outputs data.

E. Sbuf

Each of the 100 instances of the sbuf module selects the corresponding strobe pulse and detector address data as inputs. A fixed delay for each instance, having a range determined by a uniform random value between 0 and fe_skew, is applied to both the strobe pulse and da data. The parameter fe_skew is meant to represent the timing differences among banks in the generation and distribution of the critically timed strobe pulses and is estimated to have a typical value of 1.0 ns. In addition, every strobe pulse and da data passing through the module is randomly jittered on a pulse by pulse basis using a normal distribution that has a standard deviation called s_jitter_sd. The purpose of this parameter is to characterize the bank timing differences between the two true+scattered event components. With the jitter of each component, s_jitter_sd = 2.5 ns, the resulting rms jitter between the two components is 3.5 ns.

Since some bank circuits have a field of view that includes the boundary between 0 and 99, duplicate outputs of each strobe pulse are produced in order for Verilog to handle the boundary. For n<39, the two identical versions of the delayed strobe pulse, produced by sbuf, are identified as ds[n] and ds[n+100]. For n≥39, the two outputs are simply identified as ds[n]. The delayed da data is identified as dda_n.

F. Bank_ckt

Each of the 100 instances of the bank_ckt module selects its own strobe pulse and detector address data, ds[n] and dda_n, respectively. It also selects the range of 41 other strobe pulses, namely ds[n+30:n+70], which correspond to the columns of Figure (2). Two modules, called coinc_oran and monitor, are contained inside the bank_ckt module. Coinc_oran represents the coincidence logic and produces a variable width pulse, called cout, that is equal to the overlap of ds[n] and any of the strobe pulses in the range ds[n+30:n+70]. The monitor is not part of the data acquisition path but is rather used to facilitate computing DAQ output statistics.

Upon sensing the leading edge of cout, a busy pulse is generated after a fixed delay for each bank_ckt instance. This fixed delay has a uniform random value between 0 and bnk_ckt_skew. The parameter bnk_ckt_skew represents the maximum difference among banks in processing time and is estimated to have a typical value of 1.0 ns. The width of the busy pulse generated is dependent upon the required minimum processing time of each event. In the simulation, this width is controlled by a parameter called tbusy and has a typical value of 20 ns. The busy pulse is output to the busy line of the group in which the bank_ckt is associated. Coincidences that occur in the same group and are detected while the group's busy line is being asserted, will extend the group's busy signal to the comp_int module.

Each bank_ckt also inputs the group busy line, and if it is not being asserted at the time a given bank_ckt detects a coincidence, the bank_ckt outputs 12 bits of coincidence data onto the group's data bus. The output data is composed of the bank address, ba_n, and the detector address dda_n. Ba_n is passed to the bank_ckt instance as a Verilog "parameter" and the dda_n is input to the bank_ckt from the corresponding sbuf_n instance. There is a delay (t_a) from the time a bank circuit asserts busy to the time that all other bank circuits in the group receive it. Therefore, data from more than one bank circuit can be placed on the bus causing erroneous data to be input to the comp_int module. Because of the bit wise wire-OR nature of the bus, the erroneous data will not be random but will rather be biased to higher numbers. This busy delay is characterized by a parameter, called t_a, and is assumed to have a typical value of 2.5 ns. In real situations, this delay would be dependent on the physical proximity of various banks in a group. In the simulation, however, t_a is constant for all bank_ckts in a group.

G. Coinc_oran

The coinc_oran module is located within each bank_ckt instance and has the same strobe pulse inputs as the bank_ckt instance. It is composed of Verilog primitive gates which perform the function

$$\text{cout} = \text{ds}[n] \bullet (\text{ds}[n+30] + \text{ds}[n+31] + \ldots + \text{ds}[n+70]) \quad (7)$$

Because the width of cout can be very narrow, a parameter called coinc_min_width is used to establish the minimum pulse width for cout. If cout's pulse width is less than coinc_min_width, then no output will be produced.

There is a fixed delay associated with each strobe pulse input to coinc_oran and has its range determined for each input by a uniform random value between 0 and the parameter called coinc_skew. This parameter is meant to represent propagation delay differences from various inputs to cout plus strobe pulse distribution differences. Coinc_skew is estimated to have a typical value of 1.3 ns which is based on its intended implementation in ECL programmable logic devices.

H. Monitor

The monitor module is located within each bank_ckt instance and its purpose is to determine all true+scattered and accidental coincidences within the DAQ's field of view. It also contains logic similar to the coinc_oran module but has no internal skewing. A variable, called ba_con (bank coincidences), is incremented for every coincidence detected and is passed up to pet_daq. It is summed for all banks in pet_daq and becomes a variable called tot_con (total coincidences). Besides counting each coincidence, monitor places the ba and da data for each coincidence in a queue, called tot_con_que, for later analysis in the comp_int module.

I. Comp_int

The comp_int module receives the busy line and data bus from each group. The four busy lines are OR-ed together within comp_int to produce the busy_sum signal. After a delay, which is started at the leading edge of busy_sum and is determined by a parameter called t_b, the busy lines are latched and input to a lookup table. This table is programmed to reject all events in which the number of busy signals latched is not equal to two. When an event is accepted, the lookup table outputs determine which two of the group data buses will form the data_xy output. If t_b is either too small or too large, the amount of data rejected will be more than necessary.

Several characteristics of the architecture, such as the number of busy signals and how they are detected, are recorded within comp_int for the purposes of trying to understand the effects of changes in the system parameters. To measure DAQ performance, the data_xy output word is compared to the contents of both the tru_fov_que and tot_con_que. Performance is then determined by categorizing the events as shown in Figure (4). The entire box represents the total number of coincident events (tot_con) in the DAQ's field of view that are input to all the bank circuits. These include all true+scattered events and all accidental events in the field of view that are formed by two trues, two singles, or a true plus a single. The top three sections of the box represent all the events that are output and the bottom section represents those events that are rejected. The unbiased true+scattered events, represented by the first output section, are the only desirable PET events. The second output section represents the unbiased accidental events which are inherent to PET data. Finally, the third output section represents those biased true+scatted and accidental events that are altered by the DAQ. These events are biased such that the bank addresses and detector address are skewed toward larger numbers. This last category is the most troublesome to PET data analysis.

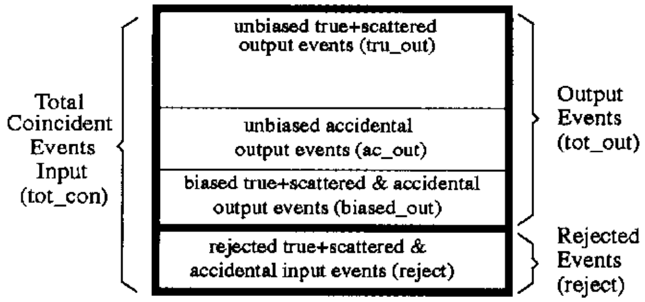

unbiased true+scattered output events (tru_out)
unbiased accidental output events (ac_out)
biased true+scattered & accidental output events (biased_out)
rejected true+scattered & accidental input events (reject)
Total Coincident Events Input (tot_con)
Output Events (tot_out)
Rejected Events (reject)

Fig. 4 - Categorization of Simulation Events

V. SIMULATION RESULTS

The simulation parameters described in Table (1) and throughout this paper are divided into determinate and indeterminate groups. Values selected for the determinate group are based on an estimate of probable circuit properties and remain fixed for the simulation condition indicated. The indeterminate parameters are considered control parameters and are varied one at a time. Typical values for these parameters were selected as a best guess. Simulation times for all runs were fixed at 1.0 sec.

A. Ideal vs Typical

Values indicated in Table (1) for determinate parameters in the ideal case were selected to reveal the best possible results that could be obtained for this architecture. Results obtained for the ideal and typical cases are indicated in Table (2).

	Ideal		Typical	
total output events	27,264	98.5%	27,292	96.9%
rejected events	419	1.51%	859	3.05%
total coincident events	27,683	100%	28,151	100%
unbiased true+scattered output events	25,214	92.5%	25,152	92.2%
unbiased accidental output events	1,971	7.23%	2,058	7.54%
biased output events	79	0.29%	82	0.30%
total output events	27,264	100%	27292	100%
unbiased true+scattered output events	25,214	99.3%	25,152	99.0%
true +scattered events missing in output	184	0.72%	246	0.97%
total true+scattered events input	25,398	100%	25,398	100%

Table (2) - Typical vs Ideal

The resolving time (τ) in the typical case is 14 ns as compared to 12 ns in the ideal case to compensate for the difference in the coinc_min_width parameter. As a result, 468 (28,151-27,683) additional accidental coincidences are generated in the typical case. The pure (single-single) accidental rate is calculated [6] from

$$R_{ac} = 2\tau R_s^2 \qquad (8)$$

where R_s = singles rate (656.79 kHz)
τ = resolving time

and predicts an additional 336 events. The other 132 accidental events are due to singles being coincident with true events and other accidental events.

Most of the additional accidental events generated in the typical case are rejected (859-419=440). This is evidenced by the fact that the number of biased output events only increase by 3 (82-79), and the number of unbiased accidentals output events increase by 87 (2,058-1,971). The number of unbiased true+scattered output events, in the typical case, only decreases by 62 events (25,214-25,152) or 0.3%. In summary, it appears that for this architecture, efforts to minimize skews and the like are only marginally worthwhile.

B. Optimization of Resolving Time

Figure (5) is a plot of the generation of various types of events as a function of resolving time. The number of unbiased accidental output events (ac_out) increases linearly with a slope of 168 events/ns as predicted by Eq (8) and intersects the res_time axis at 2 ns due the value of coinc_min_width. The number of unbiased true+scattered output events (tru_out) increases sharply at low res_time, peaks at 15 ns, and then slowly decreases (-9.5 events/ns). Biased output events (biased_out) increase slightly with res_time at a slope of 4 events/ns. Unbiased true+scattered output events, which represent the real PET data, increase as a percentage of total output at lower resolving times. Hence, the optimum resolving time is the lowest possible resolving time for which rejections are at an acceptable level. Rejected events reach a minimum (859 events, 3.05%) at a resolving time of 14 ns.

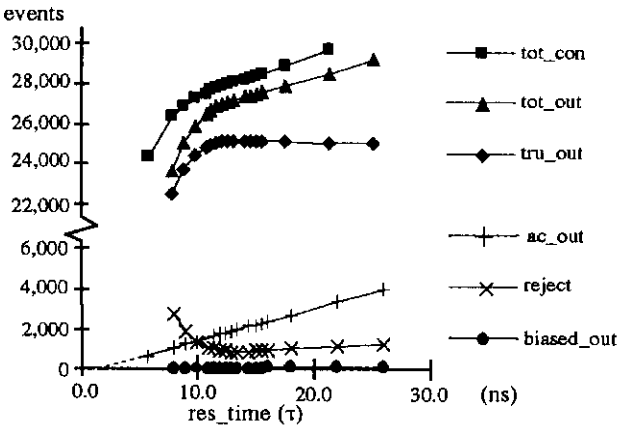


Fig. 5 - Coincident Event Generation vs Resolving Time

C. Optimization of t_b

Figure (6) is a plot of event generation as a function of t_b. Both unbiased true+scattered (tru_out) and accidental (ac_out) events increase sharply with t_b at values of t_b < 2 ns, peak at 2 ns, then slowly decrease at nearly the same rates of -4.6 events/ns and -3.2 events/ns, respectively. Since the number of biased output events (biased_out) remains constant at 82 (0.30%) for t_b > 2 ns, the ratio of tru_out/(ac_out + biased_out), which represents the signal/noise ratio, increases slightly from 11.67 at 2 ns to 11.83 at 8 ns. Hence, t_b should be > 2 ns but small enough to maintain adequate throughput rate.

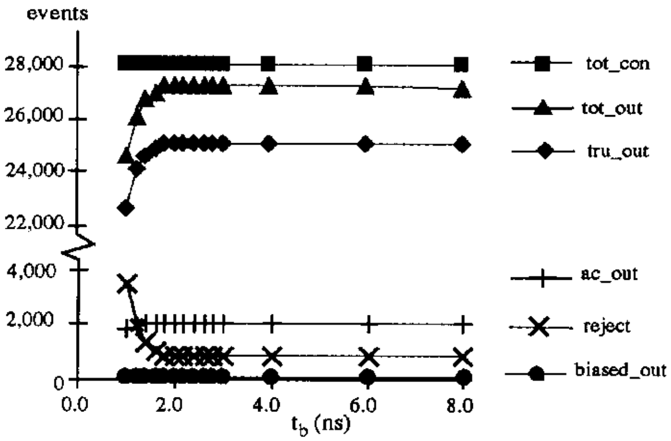


Fig. 6 - Coincident Event Generation vs t_b

D. Optimization of tbusy

As with t_b, the variation of tbusy does not effect total coincident events generated (tot_con). Event output varies only slightly as a function of tbusy from the typical values in Table (2). Rejected events increase at a rate of 0.7 events/ns and are distributed proportionally over true+scattered, accidentals, and biased output events. The percentage of total coincident events rejected is 3.04% at tbusy = 10 ns, but increases linearly to only 5.10% at the extreme value of tbusy = 800 ns. Since decreasing tbusy does not improve performance significantly, efforts to increase computer interface processing speeds would not significantly improve DAQ performance.

E. Rate Effects

Table (3) shows the effect of changing the singles input rate (R_s) and the true+scattered input rate (R_t) by ± 50% of the typical rates indicated in Eq (1) and Eq (2) while all other parameters retain their typical values. Corresponding output characteristics of the typical case are found in Table (2). Whereas the ± 50% rate variation was chosen to show near typical rate effects, the actual PET rates are expected to be as high as five times the typical rate in some cases.

From Eq (8), the generation of the unbiased accidental events increases as R_s^2, whereas the. unbiased true+scattered events increase linearly with R_t. As expected, the percentage of output events that are unbiased true+scattered events decreases from 94.6% for the low rate (-50%) to 85.7% for the high rate (+50%), thus decreasing the signal/noise ratio. This difference of 8.9% is due mostly to the increase of the unbiased accidental outputs. Changing the input rates has little effect on the percentage of the biased outputs since it only increases from 0.18% for the low rate to 0.59% for the high rate. The percentage of rejected events increases from 1.96% for the low rate to 6.02% for the high rate.

	rates @-50%		rates @+50%	
true+scattered rate (R_t)	16,920 Hz		50,759 Hz	
singles rate (R_s)	426,439 Hz		1,278,772 Hz	
total output events	17,864	98.0%	58,176	94.0%
rejected events	357	1.96%	3,727	6.02%
total coincident events	18,221	100%	61,903	100%
unbiased true+scattered output events	16,897	94.6%	49,873	85.7%
unbiased accidental output events	934	5.23%	7,960	13.68%
biased output events	33	0.18%	343	0.59%
total output events	17,864	100%	58,176	100%
unbiased true+scattered output events	16,897	99.3%	49,873	98.1%
true +scattered events missing in output	121	0.71%	984	1.93%
total true+scattered events input	17,018	100%	50,857	100%

Table (3) - Variation Due to Changes in Input Rates

VI. CONCLUSION

The underlying theme obtained from the results of the simulation indicate that the effect of changing various parameters on the performance of the DAQ are not critical. Additionally, the comparison of the performance obtained under ideal conditions vs typical conditions reveals the architecture's fundamental performance limitations. The simulation allows the performance of the architecture to be characterized quantitatively in terms of the percentage of unwanted events in the output and the percentage of rejected events under varying conditions. However, it is unclear how these percentages affects ultimate PET analysis. Further studies would require analysis of the data_xy output file produced by the simulation.

VII. ACKNOWLEDGMENTS

The authors gratefully acknowledge Dennis Black of Fermilab for helping structure the simulation and debug the Verilog code. The authors are also grateful to David Khazins of Rockefeller University and Hector Gonzales of Fermilab for their help in understanding and implementing the interval distribution used for event timing. Lastly, the entire project would not have been possible without the guidance of Nick Yasillo of The University of Chicago in establishing the system goals and requirements.

VIII. REFERENCES

[1] S. E. Derenzo, et. al., "Analytical Study of a High-Resolution Positron Ring Detector System for Transaxial Reconstruction Tomography", Journal of Nuclear Medicine, vol. 16, pp. 1166-1173, 1975.

[2] E. Tanaka, et. al., "Analytical Study of the Performance of a Multilayer Positron Computed Tomography Scanner, "Journal of Computer Assisted Tomography, vol. 6, No. 2, pp. 350-364, 1982.

[3] R. D. Evans, *The Atomic Nucleous, New York: McGraw Hill, 1955, pp. 753-754.*

[4] A. W. Drake, *Fundamentals of Applied Probability Theory*, New York: McGraw-Hill, 1967, pp. 137-142.

[5] I. A. Shreider, *Method of Statistical Testing; Monte Carlo Method*, New York: Elsevier Publishing Company 1964, pp.252-265.

[6] J. A. Sorenson, M. E. Phelps, *Physics in Nuclear Medicine*, Orlando: Grune & Stratton, 1987, pp 438-442.