

University of Saskatchewan
Subatomic Physics Internal Report

SPIR 143-1

Tagger Simulation Software Version 1.1

Rob Pywell*

October 9, 2009

Abstract

This document contains a brief description of the simulation software used to describe the TDC spectra from the Saskatchewan Accelerator Laboratory Photon Tagging Facility.

Contents

Contents	1
1 Introduction	2
2 Simulation Code	4
3 Installing the Simulation	6
4 Running the Simulation	7
5 Description of the Input Data	8
5.1 Simulation Description File	8
5.2 Beam Intensity Profile File	10
6 Description of the Output	11
7 Improvements to the Code	12
A Input File	13
B Output Summary and Statistics File	16

*rob.pywell@usask.ca

1 Introduction

The focal plane of the Saskatchewan Accelerator Laboratory (SAL) photon tagging spectrometer contains a set of overlapping scintillators for detecting the electrons. There are two sets of these electron counters: a front set, closer to the spectrometer and a back set. In an effort to reduce background a coincidence between a front counter and a back counter is required for a hit on the focal plane to be recorded. The front and back counters are normally offset so that the energy width of channel, defined by the coincidence requirement, is approximately the same for each channel. The principle is illustrated in figure 1. Additional details of the photon tagging spectrometer can be found in reference [1].

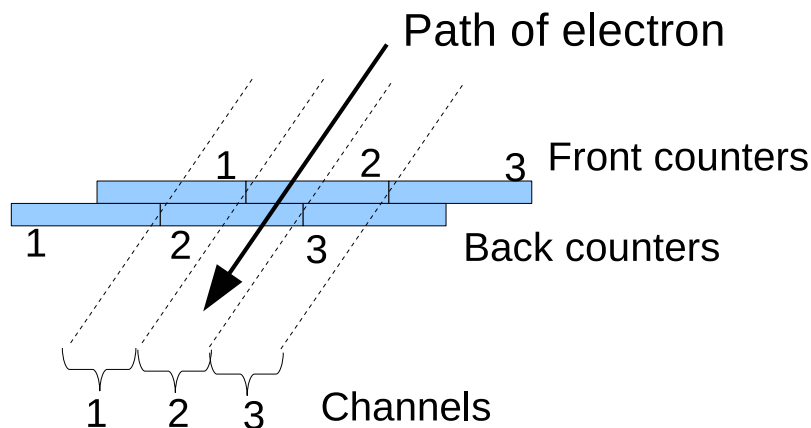


Figure 1: An illustration of the use of overlapping electron counters to define tagger channels. For example, when a single electron passes through the focal plane, a coincidence between front counter 1 and back counter 2 ensures that the electron passed through the region defining tagger channel 1. A “ghost” hit can occur at high count rates. If two electrons pass through the focal plane, one through channel 1 and another through channel 3, there will be hits in front counter 2 and back counter 2. If these two hits occur within the coincidence resolving time, a hit in channel 2 will be recorded when no electron actually passes through that channel.

Several issues associated with using the photon tagging technique have been discussed by other authors e.g.. Owens [2]. However the method of employing overlapping counters introduces additional effects that need to be taken into account when calculating the cross section for a reaction measured using tagged photons. One such effect has come to be called the “ghost” effect. This can occur when there are high count rates across the tagger focal plane. At high count rate random coincidences can occur. Referring to figure 1 If two electrons pass through the focal plane, one through channel 1 and another through channel 3, there will be hits in front counter 2 and back counter 2. If these two hits occur within the coincidence resolving time, a hit in channel 2 will be recorded when no electron actually passes through that channel. Such a non-existent hit is called a ghost. This ghost hit will be recorded in the scaler for that channel which is supposed to record the total number of

tagged photon in the energy range defined by that channel. As well, the ghost could form a coincidence with an event in the “X-arm” of the experiment and therefore end up in TDC for that channel.

A detailed calculation of the effects caused by these ghost events is difficult. Although the rate of ghosts can be estimated, the time distribution of where these ghosts appear in a TDC spectrum is difficult to quantify. Even without the ghost effect, the distribution of events in a TDC spectrum can be non-obvious. For example, because of the use of a simple overlap coincidence between the front and back counter, the time of an event in a channel can be shifted by a random coincidence with an adjacent channel. Therefore this simulation code was written in order to help quantify the contribution of these effects to a cross section calculation.

2 Simulation Code

The simulation code was written in the early 1990's using standard FORTRAN 77. If one was to write this code today it would be very much easier to write it using an object oriented language such as C++. Such a code would be shorter, more transparent to the user, easier to debug with far fewer repeated sections of code, and would probably run faster.

The simulation code uses a Poisson statistical model to determine the times when electrons arrived at the focal plane given an average rate in each tagger channel. These electrons generate pulses from the discriminators attached to each counter. These pulses are then tracked nanosecond by nanosecond through a simulation of the tagger electronics.

The “electronics” as simulated is illustrated in figure 2. The simulation track information for 5 tagger channels labeled 0–5. Ultimately however, only the 3 middle channels (1–3) are tracked completely through the system as complete coincidence information is not available for the outer 2 channels. As well, if one wishes to examine the effects of certain software cuts, such as ignoring events where there is a triple coincidence between a channel and its neighbors, only the center channel can be used.

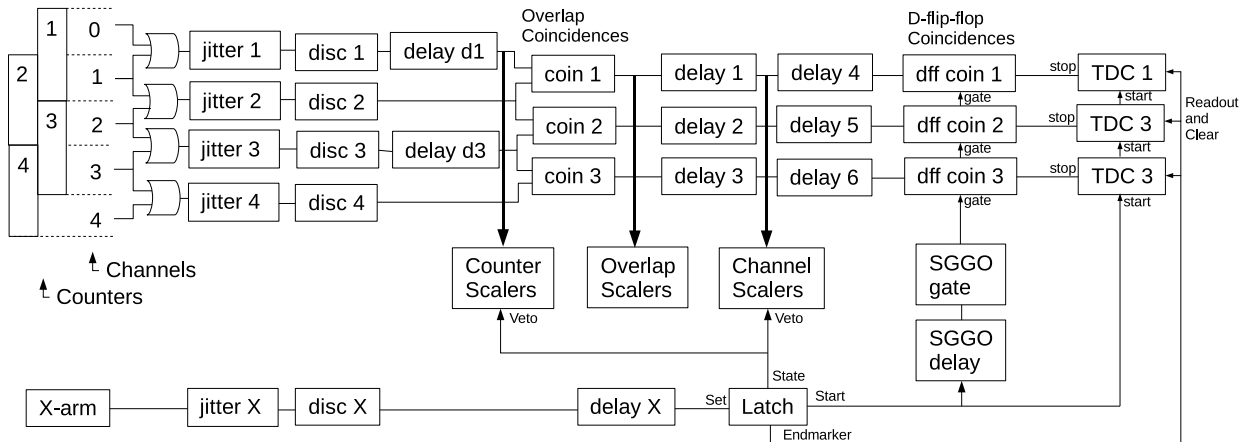


Figure 2: A block diagram of the tagger electronics as it is represented by the simulation.

Digital information is tracked through the system in 1 ns steps. A non-zero number represents the high state of a voltage pulse and a zero represents a low voltage state. (While the time steps are in principle selectable, it is my recollection that there were subsequent modifications to the code that essentially hard wired the time step to be 1 ns.) At each step, each module in the system examines the state of its input and determines the state of its output from the input state and the state of its internal memory.

At the beginning of each time-step loop Poisson statistics is used to determine if there is an electron in each tagger channel 0–5. This is calculated from the average electron rate on each channel along with information on the relative beam intensity as a function of time. This information is supplied by a file which gives the relative intensity as a function of time. The ‘or’ of channels 0 and 1 is used to determine if there is a signal in counter 1. Similarly the state of the other counters is determined. It is also possible at this stage to insert hits to

tagger channels from “radiator out” electron. That is, electrons not correlated with photons that could produce an X-arm trigger. (Since radiator out rate are usually very low this feature has not often been used.) Poisson statistics are also used to determine if there is a signal from the X-arm.

To simulate the jitter in the arrival time of real pulses a time shift is introduced based on a Normal distribution with a given standard deviation.

When a high signal reaches a discriminator module an output is generated with a given time width. The discriminators act as updating discriminators so that if the output is still high when an input is received the output width is extended.

The delays ‘d1’ and ‘d2’ are so that the front counters will normally determine the timing of the output of the simple overlap coincidences that follow.

The delays 1–3 simulate the long delays in the real tagger. In a real delay the rise and fall time of a digital signal can degrade. A very short pulse can degrade sufficiently that it will not be registered by a following module. Therefore in the simulation there is the option of specifying a minimum width signal that will reach the end of these delays.

Meanwhile the X-arm determines the presence or not of an SGGO gate that will allow signals to pass on to the stop inputs of the TDCs. The coincidence between the SGGO gate and the channel signals is determined by the D-flip-flop coincidence modules which are non-updating. The SGGO gate is generated by a Latch which is set to stay high for 200 ns. The output of this is used as the veto for the channel scalers simulating the latch in the SAL Tagger Module. The 200 ns also represents the maximum run time for the TDCs. At the end of this 200 ns a readout and clear of the TDCs is triggered. This simulates, as well the computer dead-time associated with the read-out of the modules by the data acquisition system. This is of course very much smaller than in a real experiment but this does not matter as we will only record and compare information gathered in the live time.

After the TDC readout histograms are incremented. The advantage of the simulation is that it is possible to label the pulses being tracked through the system. The original channel signals are labeled with its channel. Then when signals reach the overlap coincidence it is possible to determine if the coincidence is a ghost or not. From that point on the signal carries a ghost or not label. Therefore it is possible to build a histogram of just the ghost signals.

A histogram is also built for channel 2 for events where there is also no TDC stop in channel 1 and no TDC stop in channel 3. Rejecting these triple coincidences certainly rejects ghosts but it also rejects many real events.

The simulation also tries to simulate the operation of a duty factor meter. Here the rate of coincidences between two uncorrelated channels is used to estimate the duty factor as described in [1].

3 Installing the Simulation

The simulation code may be obtained from

```
http://nucleus.usask.ca/ftp/pub/rob/
```

and the file name is

```
tag_sim-1.1.tgz
```

It may be unpacked using

```
tar zxvf tag_sim-1.1.tgz
```

This will create the directory named `tag_sim-1.1`. Change to this directory and type `make`. The output should be something similar to the following:

```
rob@koala:~/tagger/tag_sim-1.1> make
make tag_sim
make[1]: Entering directory '/home/rob/tagger/tag_sim-1.1'
f77 -O -Wuninitialized -fno-automatic -c -o tag_sim.o tag_sim.f
f77 -O -Wuninitialized -fno-automatic -c -o tag_sim_sub.o tag_sim_sub.f
f77 -O -Wuninitialized -fno-automatic -c -o ran1.o ran1.f
f77 -O -Wuninitialized -fno-automatic -c -o ranorm.o ranorm.f
f77 -o tag_sim tag_sim.o tag_sim_sub.o ran1.o ranorm.o
make[1]: Leaving directory '/home/rob/tagger/tag_sim-1.1'
make caldf
make[1]: Entering directory '/home/rob/tagger/tag_sim-1.1'
f77 -O -Wuninitialized -fno-automatic -c -o caldf.o caldf.f
f77 -o caldf caldf.o
make[1]: Leaving directory '/home/rob/tagger/tag_sim-1.1'
rob@koala:~/tagger/tag_sim-1.1>
```

The code is now ready for running.

Contained in the directory `tag_sim-1.1` are several useful files.

1. `README` gives basic information about the code and where to find documentation (this document!).
2. `LICENSE` gives information about the copyright under the GNU public license.
3. `simulate.data` is an example input file for the simulation.
4. The files `profile.xxx` are example beam intensity profile files.
5. `tag_sim` is the simulation code.
6. `caldf` is a code for estimating the duty factor from a beam intensity profile file.
7. The directory `latex` contains all files used to create this document.
8. The file `sim_runnumber` keeps track of the simulations so that each run creates a unique set of output files.

4 Running the Simulation

Two files are needed to run the simulation: a “Simulation Description File” and a “Beam Intensity Profile” file. Complete descriptions of the contents of these files are given in section 5.

A directory must be created where the output files will be placed. The name of this directory is specified in the simulation description file.

If the simulation description file is named `simulate.data`, then the code would normally be started, by typing

```
tag_sim < simulate.data &
```

This would start the code running in the background.

At start-up a run number, which is stored in the file `sim_runnumber`, is incremented and all files created by the run will have names with the new runnumber appended to them. It is therefore possible to have more than one simulation running at the same time on the same computer.

While the simulation is running all the output files are written every 1,000,000 loops of the code. Therefore, if the simulation is terminated prematurely, those output files will be properly synchronized and may be analyzed correctly. The exception is the event-by-event data file if the user has requested that to be written. See section 6 below for more information about the output files and their contents.

5 Description of the Input Data

5.1 Simulation Description File

The simulation is controlled by an input file which lists all the parameters specific to a particular simulation. The file is read by the code from the standard input. Only every second line of the file is scanned for input parameters. The other lines are comments that are used to describe what parameters appear in the following line. This makes it much easier to edit the input parameter file. An example input file is listed in appendix A.

The following lists the input file line number followed by a description of the input parameters on that line. Multiple numbers on a line are white space delimited.

1. **One String:** A general comment describing this particular simulation run. (The string may contain spaces.)
3. **Two Integers:** The first integer describes the run mode. If ‘zero’ the simulation is for a normal tagger channel requiring a coincidence between the front and back counters. If ‘one’ the simulation ignores the coincidence requirement. This is known as “counter” mode. (This mode has not been significantly tested.)
The second integer determines if event-by-event data is written to file. If ‘one’ event-by-event data is written, if ‘zero’ only the histograms are written. It has been our experience that the event-by-event data is not normally necessary except for debugging or if you wish to try specific event-by-event software cuts.
5. **One Float:** The total rate of electrons that produce bremsstrahlung, whether the electrons reach the focal plane or not.
7. **Five Floats:** The fraction of the electrons that produce bremsstrahlung that reach each tagger channel (0–4).
9. **Five Floats:** The probability that a photon, associated with a recoil electron that reaches each tagger channel (0–4), actually produces an X-trigger, if that photon actually reaches the X target. This is essentially the reaction probability that we are trying to measure. It is related to the cross section and the target density and thickness.
11. **One Float:** The fraction of radiator out hits in the tagger channels. Radiator out hits are not associated with post bremsstrahlung electrons and so will have no correlation with X-arm triggers. However the rate of these events on a tagger channel is assumed to be proportional to the rate of post bremsstrahlung recoil electrons on that channel. (Usually this number is very small so it is often set to zero.)
13. **One Float:** The probability that a photon, associated with a recoil electron that *does not* reach any of the tagger channels (0–4). This corresponds to recoil electrons that hit other parts of the tagger focal plane, or miss the focal plane altogether, but nevertheless are associated with photons that *could* cause an X-trigger.

15. **One Float:** The probability that a photon, associated with a recoil electron that *does not* reach any of the tagger channels (0–4), actually does cause an X-trigger. This would normally be close to the probabilities given on line 9.
17. **One Float:** The absolute tagging efficiency. This is the fraction of photon, associated with recoil electrons hitting the tagger channels, that actually reach the X target. Since this just appears as an overall scale factor in the total number of events recorded in the simulation, this is usually just set to one.
19. **One Integer:** The total number of X-triggers to be recorded before the simulation stops.
21. **Five Floats:** The standard deviation (in nanoseconds) of a normal distribution of time jitter that is added to the signals to each counter (the first four numbers) and the X-arm signal (the fifth number).
23. **One Float:** The rate of a pulser signal that can be added to the X-arm. (To be honest I cannot remember why this feature was added. I think it was used for debugging.)
25. **Five Floats:** The width of the discriminator output pulses in nanoseconds. The first four numbers are the widths for the counter discriminators and the last number is for the X-arm discriminator.
27. **Three Floats:** The delay time in nanoseconds for the delays 1, 2 and 3.
29. **Three Floats:** The delay time in nanoseconds for the delays 4, 5 and 6.
31. **Three Floats:** The minimum width, in nanoseconds, that a pulse needs to have for it to make it through a delay. The first number is a common minimum with for delays 1 and 4. The second number is a common minimum with for delays 2 and 5. The third number is a common minimum with for delays 3 and 6.
33. **Two Floats:** The delays, in nanoseconds, for the front counters relative to the back counters so that, usually, the front counters determine the timing for the channels. The first number is for delay d1. The first number is for delay d3.
35. **One Float:** The X-trigger delay in nanoseconds.
37. **Two Floats:** The SGGO gate generator delay (first number) and width (second number), both in nanoseconds.
39. **Two Integers:** This describes the repeat time of the beam intensity profile. The first integer is the repeat time in nanoseconds. The second integer is granularity of the intensity profile used internally by the code. This integer is the width (in ns) of the bins. It must be an integer divisor of the repeat time.
41. **One String:** The name of a file containing a description of the beam intensity profile as a function of time.

43. **One String:** The name of a directory where the output files are to be written. It may be a full path name or relative to the directory from where the simulation was started.

5.2 Beam Intensity Profile File

This file describes the beam intensity as a function of time. It is assumed in the following example that the beam intensity profile repeats with a 360 ns period, the pulse stretcher ring period at SAL. However this period can be changed by changing line 39 of the simulation description file.

The format of the file is a white space delimited pair of numbers on each line. The first number is a time (in ns) and the second number is an intensity between 0 and 1. A maximum of 100 lines may be included. The first line must be for time 0 ns and the last line must be for time 360 ns (or as defined in the simulation description file). The intensities at time 0 and time 360 ns must be equal. The following is an example file which describes a flat top triangular intensity profile.

```
rob@koala:~/tagger/tag_sim-1.1> cat profile.80p
0. 0.0
155 1.0
180 1.
205 1.0
360 0.0
rob@koala:~/tagger/tag_sim-1.1>
```

The program will interpolate the points read from the file to generate an intensity profile with a granularity of 10 ns. This granularity can be changed by modifying line 39 of the simulation description file. That profile will be used to determine the number of photons arriving at a tagger channel at each time using Poisson statistics. The simulation will calculate the approximate duty factor represented by the profile. However it is often useful to know this in advance of running the simulation. For this purpose a utility program called “caldf” is provided. An example of running it is:

```
rob@koala:~/tagger/tag_sim-1.1> caldf < profile.80p
Calculated duty factor = 0.761
rob@koala:~/tagger/tag_sim-1.1>
```

Figure 3 illustrates several profiles, all of which have duty factors near 80%.

6 Description of the Output

The simulation produces several files of output. The files are written to the directory specified in the simulation description file (see section 5.1). The names of the files written are:

```
sim_stats.nnn
sim_tdc1.nnn
sim_tdc1_ghost.nnn
sim_tdc2.nnn
sim_tdc2_ghost.nnn
sim_tdc2_no3.nnn
sim_tdc3.nnn
sim_tdc3_ghost.nnn
sim_tdc_event.nnn
```

Where ‘*nnn*’ is replaced by the run number for the simulation. All of the files, except `sim_tdc_event.nnn`, are written at the end of the simulation. As well the files are written after every 1,000,000 loops of the simulation.

The file `sim_stats.nnn` contains summary information and statistics for the run. An example of such a file is given in appendix B.

The files `sim_tdc1.nnn`, `sim_tdc2.nnn`, and `sim_tdc3.nnn` contain histograms of the three TDC spectra.

The files `sim_tdc1_ghost.nnn`, `sim_tdc2_ghost.nnn`, and `sim_tdc3_ghost.nnn` contain histograms of the three TDC spectra for only those events which are labeled as ghosts.

The file `sim_tdc2_no3.nnn` contains a histogram for TDC 2 for the situation when there is *not* something recorded in all 3 TDCs in the same TDC readout. Clearly the resolving time for this triple coincidence is the SGGO width, so this cut will remove *many* more events than just the ghosts.

Examples of the histograms in the above mentioned files are shown in figure 4.

The file `sim_tdc_event.nnn` is only written when the “write event data” run mode is selected. In that case the file is appended to at every TDC readout for which there is a hit in at least one of the TDCs. The format of the file is one line per TDC readout. The first three numbers are the channel numbers of the hits in the three TDCs. the fourth number is a bit pattern which indicates whether the TDC hits are labeled as ghosts or not. Bit 0 is set if a hit in TDC 1 is labeled as a ghost, bit 1 is set if a hit in TDC 2 is labeled as a ghost, and bit 2 is set if a hit in TDC 3 is labeled as a ghost.

7 Improvements to the Code

A number of improvements to the tagger simulation code come to mind. These are listed below in no particular order. The author would be grateful to receive additional suggestions. Send comments and suggestions to `rob.pywell@usask.ca`.

- Port to C++.
- Modify to allow different repeat times for the beam intensity profile. At present only 360 ns is allowed. This has been done in Version 1.1.
- ...

A Input File

The following is an example input file that describes the simulation to be performed.

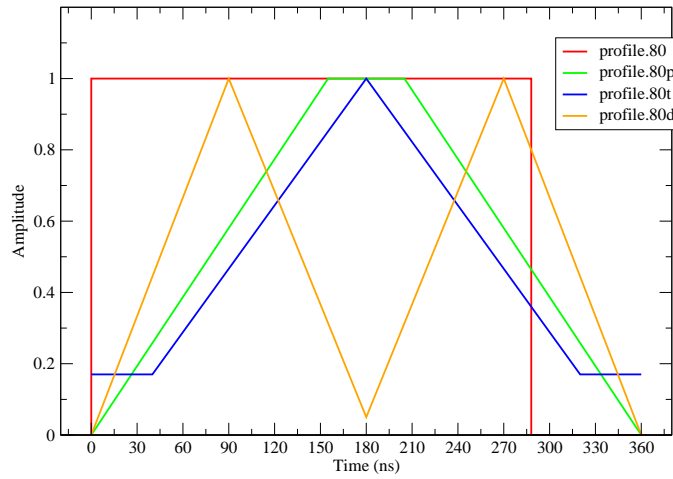


Figure 3: Examples of beam profile descriptions.

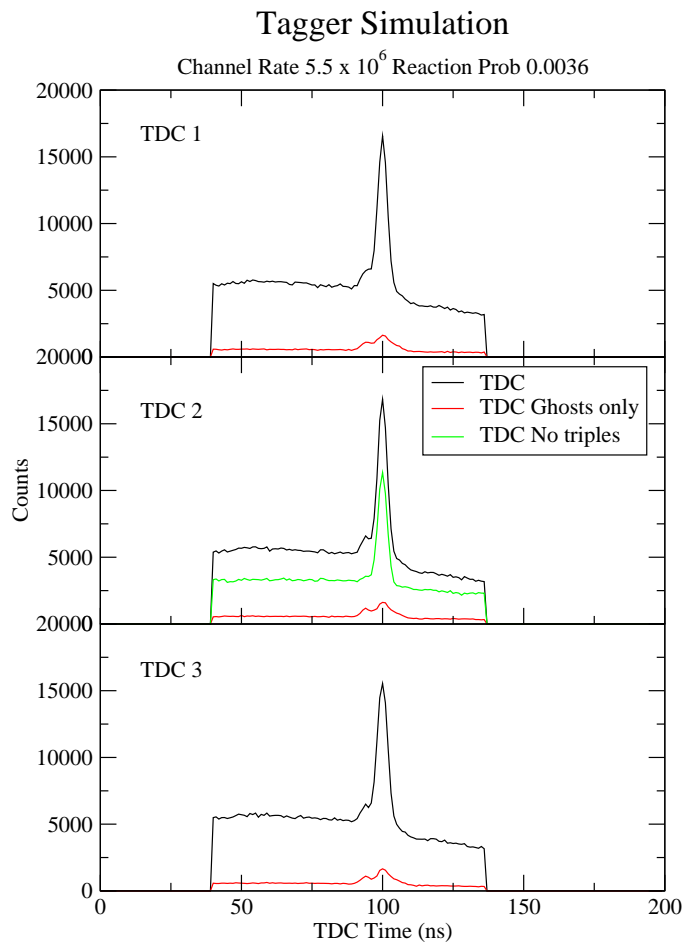


Figure 4: Examples of the histograms produced by the simulation code.

```

Run description comment
# Run Mode: 0=channels,1=counters, Event data: 0=don't write,1=write
0 0
# Rate of electrons that produce bremsstrahlung
5.5e8
# Prob. of recoil electron in tagger channel i
0.01 0.01 0.01 0.01 0.01
# Prob. that a photon, in tagger channel i, produces X-trigger
0.0036 0.0036 0.0036 0.0036 0.0036
# Fraction of radiator out events
0.00
# Prob. electron outside range 0-4 and photon can produce X-trigger
0.100
# Prob. that such a photon does produce an X-trigger
0.0036
# Absolute tagging efficiency
1.
# Number of X triggers to simulate
1000000
# Discriminator jitter sigmas (1, 2, 3, 4, x)
0.45 0.45 0.45 0.45 1.5
# x trigger pulser rate
0.
# Discriminator output widths
15. 20. 15. 20. 1.
# Delays 1, 2, 3
20. 20. 20.
# Delays 4, 5, 6
113. 113. 113.
# Minimum width in delays 1&4, 2&5, 3&6
8. 8. 8.
# Delays for front counters relative to back (d1, d3)
6. 6.
# Delay for x trigger
40.
# SGG0 delay and width
39. 97.
# beam intensity profile file name
profile.80
# directory for output files
sim_save

```

B Output Summary and Statistics File

The following is an example of the output file `sim_stats.nnn`.

Tagger simulation running on machine koala.usask.ca, run number 1

Comment:

Run description comment

----- Channel Mode -----

Brems electron rate = 5.500E+08 Hz, Time Interval (delta_t) = 1.00 ns

Fraction of radiator out electrons on channel i = 0.00E+00

Absolute tagging efficiency = 100.0% Calculated Duty Factor = 80.6%

Beam Profile file name: profile.80

Chan: 0 Chan: 1 Chan: 2 Chan: 3 Chan: 4 Off Chan.

Prob. of post-brems electron on channel i:

0.0100 0.0100 0.0100 0.0100 0.0100 0.1000

Prob. of x-trigger from post-brems electron on channel i:

0.0036 0.0036 0.0036 0.0036 0.0036 0.0036

Brems electron rate on each channel i:

5.50E+06 5.50E+06 5.50E+06 5.50E+06 5.50E+06 5.50E+07

X-trigger pulser rate = 0.

Chan: 1 Chan: 2 Chan: 3 Chan: 4 X

Discriminator jitter sigma (ns)

0.4500 0.4500 0.4500 0.4500 1.5000

Discriminator output widths (ns)

15.0000 20.0000 15.0000 20.0000 1.0000

Delay Lengths (ns) 1: 20.0 2: 20.0 3: 20.0

Delay Lengths (ns) 4: 113.0 5: 113.0 6: 113.0

Delay Lengths (ns) x: 40.0 d1: 6.0 d3: 6.0

Minimum width in delays (ns) 1&4: 8.0 2&5: 8.0 3&6: 8.0

SGGO delay = 39. ns, width = 97. ns

Number of loops = 3.581E+09 Run time = 8.289E+03 s Rate = 4.32E+05 /sec

Inputs counters:

Brems electrons: 19696838 19694332 19696576 19725254 19704415 out: 196826455

X-triggers: 71042 71220 71056 70380 70274 out: 705143

Rad out electrons: 0 0 0 0 0

Disc. inputs: 38984620 38980024 39012057 39020788

Overlap outputs: 21303871 21311559 21330801

Real time = 3.580716 s Live time = 3.381716 s (94.4425%)

X-triggers = 1000000 Cntr X: 1000000 (ungated: 1058898) SGG0: 1000000

Interrupts = 858812 (hits - tdc1: 525258 tdc2: 526815 tdc3: 525804)

Multiplicities - 0: 141188 1: 341288 2: 315983 3: 201541

Counters: Counter 1 = 30412211
Counter 2 = 28550783
Counter 3 = 30437482
Counter 4 = 28571142
Counter sum = 117971618

Scalers: Scaler 1 = 18562220 Ghosts = 1914202
Scaler 2 = 18564304 Ghosts = 1914702
Scaler 3 = 18583122 Ghosts = 1915202
Scaler sum = 55709646

Duty Factor: Counter A = 18562220
Counter B = 18583122
Counter AB = 1220465
Duty factor = 75.2 %

References

- [1] J. M. Vogt, R. E. Pywell, D. M. Skopik, E. L. Hallin, J. C. Bergstrom, H. S. Caplan, K. I. Blomqvist, W. Del Bianco, J. W. Jury, The photon tagging facility at the Saskatchewan Accelerator Laboratory, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 324, Issues 1-2, 1 January 1993, Pages 198-208, ISSN 0168-9002, DOI: 10.1016/0168-9002(93)90977-P.
- [2] R. O. Owens, Statistical treatment of tagged photon experiments, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 288, Issues 2-3, 15 March 1990, Pages 574-584, ISSN 0168-9002, DOI: 10.1016/0168-9002(90)90154-X.