RADIATION DETECTION USING INTEGRATED CIRCUITS

A Dissertation

by

MANIT DEVENDRABHAI SHAH

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Craig M. Marianno
Committee Members,	Sunil P. Khatri
	David R. Boyle
	John W. Poston, Sr.
Head of Department,	Yassin A. Hassan

August 2016

Major Subject: Nuclear Engineering

ABSTRACT

The research objective of this dissertation is to evaluate the capability of 'radiation' integrated circuits (RICs) to serve as a new type of radiation detection medium. Designed at Texas A&M University, the RIC contains both radiation-sensitive areas (RSAs) and radiation-hardened areas (RHAs). RSAs are designed so that their electrical properties change when exposed to charged particles. RHAs monitor such changes in RSAs to detect the presence of radiation. Novel detector designs utilizing the RICs were assessed and optimized, using both analytical and simulation methods, to register the major types of radiation: alpha particles, beta particles, gamma rays, and neutrons. The detector system materials and components were varied to characterize different configurations and recommend optimized RIC detector designs.

The proposed revolutionary RIC-alpha/beta probe design has two regions: one to detect alpha particles and another to detect betas along with their $E_{\beta max}$ (maximum beta-particle energy). In order to perform the $E_{\beta max}$ discrimination, the maximum penetration depth property of betas in the attenuator was utilized. In MCNPX, plate glass, Pyrex® glass, Lucite®, and natural rubber were studied as attenuator materials. For the proof of concept, attenuator materials in the shape of a wedge were analyzed. The natural rubber in the form of a wedged attenuator was observed to show superior $E_{\beta max}$ discrimination compared to other defined attenuators. The $E_{\beta max}$ resolution capability of 50-keV is possible using natural rubber attenuator.

The proposed RIC-neutron detector design uses enriched boron (96% ^{10}B) as a neutron-reactive coating to generate secondary charged particles. In MCNPX, other

neutron-reactive materials (natural boron, B_4C, and LiF) were also studied. Analysis showed that compared to other charged particles, alpha-particles were the major contributor to the signal generation in RSAs designed to detect neutrons. With the optimal thickness of 3-μm enriched boron, the signal to noise ratio for thermal neutron detection was increased by approximately three orders of magnitude.

The proposed RIC-gamma ray detection system uses a sodium iodide crystal, photocathodes, and the RIC. Unlike the traditional design, the proposed approach places photocathodes on all crystal surfaces to collect optical photons and generate photoelectrons. These photoelectrons interact to generate an electrical signal in the RSAs, and thereby the RIC detects gamma rays. The collection ratio (i.e., the number of collected optical photons under the proposed design compared to the number using a traditional design) was found to be a function of the crystal size, gamma-ray energy, and the source position. The concentration of photoelectrons as a function of the RSA radius size was also assessed to optimize the RSA size.

DEDICATION

To,

Dr. Paul Nelson, Jr., Dr. Craig Marianno & my family.

ACKNOWLEDGEMENTS

I would like to thank my committee chair, Dr. Marianno, and my committee members, Dr. Khatri, Dr. Boyle, and Dr. Poston, Sr. for their constant guidance and support throughout the course of this research. I would also like to acknowledge Peter Gumplinger associated with the TRIUMF as a scientific simulation scientist for the detector development division, and Jonathan Madsen, a Ph. D. student at Texas A&M University in the nuclear engineering department for their help with GEANT4. Also, I thank Dr. Sunil Chirayath, a research assistant professor at Texas A&M University in the nuclear engineering department for his valuable guidance on MCNP.

Last but not least, my professors in the past, Aggieland, my fellow Aggies especially James, Jose, Rob, Cricket-mates, and College/hostel-buddies who have one way or another, encouraged me.

NOMENCLATURE

APD	Avalanche Photodiodes
^{10}B	Boron-10
β	Starting beta-particle, source term
B_4C	Boron Carbide
c	Speed of light
CP	Charged-particle
dE	Energy Loss
dx	Thickness
E	Energy
$E_{\beta max}$	Maximum beta-particle energy
FL	Fractional Loss
GAPD	Geiger-mode Avalanche Photodiodes
GEANT	Geometry and Tracking
Ge(Li)	Lithium-Drifted Germanium Detector
HCP	Heavy charged-particle
^{3}He	Helium-3
IC	Integrated Circuit
LiF	Lithium Fluoride
LRAD	Long-Range Alpha Detection
MCNPX	Monte Carlo N-Particle eXtended
MFP	Mean Free Path

MPD	Maximum Penetration Depth
m_e	Mass of the electron
n	Starting neutron, source term
N	Number of Histories
NaI	Sodium Iodide
PA	Proposed Approach
PE	Photoelectric Event
PIN	P-I-N Diode
PIPS	Passivated Implanted Planar Silicon
PMT	Photo-Multiplier Tube
Q	End-point energy in MeV
RE	Relative Error
RIC	Radiation Integrated Circuit
RHA	Radiation-Hardened Area
RSA	Radiation-Sensitive Area
S	Linear Stopping Power
SiC	Silicon Carbide
Si(Li)	Lithium-Drifted Silicon Detector
SiPM	Silicon Photo-Multipliers
TA	Traditional Approach
T_e	Beta-particle energy
TRIUMF	TRI-University Meson Facility

VisEd	Visual Editor
Z	Atomic Number
ZnS	Zinc Sulfide

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

CHAPTER I

INTRODUCTION

Motivation

Researchers at Texas A&M University have designed an integrated circuit (IC) for deployment as a new radiation detection medium [1]. Most ICs are susceptible to radiation-induced failures, and decades of research have gone into solving this problem. Research at Texas A&M University has led to a new type of IC that contains both radiation-sensitive areas (RSAs) and radiation-hardened areas (RHAs) [2], [3]. These specially designed ICs can be used to detect radiation, and thus, they are referred to as radiation integrated circuits (RICs). The research undertaken for this Ph.D. dissertation uses computer simulations to explore the potential uses and advantages of using this IC as a detector. Additionally, simulations were used to optimize the design of the IC. The proposed IC can find an application in a variety of detection regimes.

What is unique about these RICs is how they can perform radiation detection. RICs contain RSAs that are sensitive to charged particles in addition to RHAs that are hardened against charged particles and have a high tolerance against all types of radiation [1]-[3]. The main function of the circuitry inside the RHA is to detect the presence of radiation by monitoring the electrical properties of the co-located RSAs [1]. The primary types of radiation that are emitted by radioactive materials are alpha particles, beta particles, gamma rays, and neutrons [4]-[6]. Of these, alphas and betas are charged particles while gamma rays and neutrons are neutral. RICs are sensitive only to charged particles. For the

detection of gamma rays and neutrons, RICs must utilize a complementary medium to generate detectable secondary charged particles. Part of the research presented here explores different system designs that incorporate the RIC to detect the above types of radiation.

Background

Figure 1.1 provides a simple understanding of the RIC design. The RSAs contain electrical circuits that are sensitive to charged particles, and they are arranged in an array.

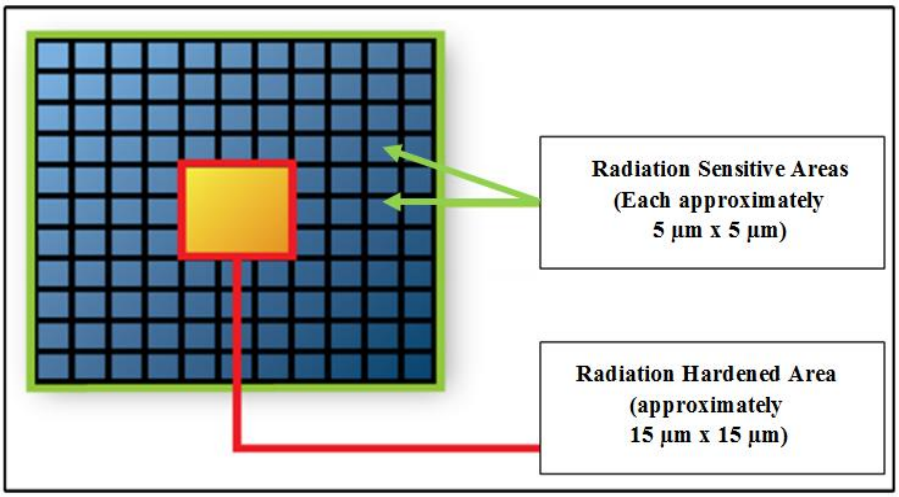

Figure 1.1. An overview of the RIC.

An RHA, designed to be highly resistant to radiation effects, is used to monitor RSAs for changes in their electrical properties (Figure 1.1). Sampling circuits in each RHA are responsible for observing several RSAs, and these sampling units in the RHA carries out this function in a round-robin manner. If an RHA registers “no change” in a

particular RSA, that RSA is skipped, and the RHA observes the next RSA. The RHA detects the presence of radiation by sensing a "change" in the electrical behavior of an RSA.

The electrical properties of the RSA change based on the radiation which has interacted in the RSA. For the gamma-ray detection system, the recovery time for the RSA will be about 230-ns, which is the decay time constant in the sodium iodide crystal [1]. On the other hand, for the neutron detector, the recovery time will be only about 40-ns. This time represents the relaxation time for alpha-particles. In general, depending on whether the RIC is being used to detect gamma-rays or neutrons the recovery time for the RSAs will be between 40-ns and 230-ns. Any RHA on the RIC can sample about 40 RSAs in a manner that is electrically feasible, because this requires a worst-case internal clock rate of 1-GHz [1].

For a 1-cm x 1-cm RIC, nearly a million ($\sim 10^6$) RSAs can be fabricated composing about 75% of the chip area. A part of the remaining area would be comprised of tens of thousands ($\sim 10^4$) of RHAs, equaling about 5% of the chip area [1], [2]. The remainder of the RIC is used to implement signal processing and clocking circuits. The RIC design offers a large detection area which improves geometric efficiency when compared to other semiconductor-based detectors. The RIC also simplifies the electronics, requiring no additional peripheral devices. These design features result in a relatively inexpensive, small, and low-power sensor when produced in bulk [1].

Similar to conventional detectors (such as sodium iodide and ionization chambers), RIC detectors can find applications in a broad range of disciplines. These can include

securing nuclear facilities by deterring theft of radioactive material and ensuring the safety of personnel by measuring dose [7]. The detectors can play an important role in national security by detecting the movement of illicit radioactive material across borders. The potential uses of these detectors include reactor monitoring and safeguards to identify undeclared activities. Also, these detectors can be an asset for consequence management to measure radioactivity following the accidental release of material [8].

Research Objective

The objective of this research is to evaluate the RIC as a new type of radiation detection medium. The novel detector designs using RICs are assessed and optimized using both theoretical calculations and numerical simulations. The simulated RIC detectors in this effort are designed to register the major types of radiation: alpha particles (α), beta particles (β), gamma rays (γ), and neutrons (n). The detector system materials and components were varied to characterize different configurations, and to recommend final detector designs that use the same RIC. Furthermore, an analysis was carried out on the RSA size to optimize the RIC. This research will act as a guide to fabricate RICs for their deployment as a new type of radiation detection medium.

Literature Review

This dissertation describes a significant innovation in radiation detector design approaches for alphas, betas, gammas, and neutrons through the use of the RIC. A literature review was carried out to describe the past and current approaches to detect these radiations based on semiconductor technology.

Various techniques have been implemented to detect alpha particles. Starting in 1908, when Hans Geiger along with Ernest Rutherford developed the first experimental method for the alpha particle detection [9]-[13]. In the early 1960s, a reverse biased p-n diode doped with an impurity (known as silicon surface barrier devices) was used for alpha-particle measurement [14]-[20]. There exists various other alpha-particle detection approaches like the utilization of a fluorescent material along with a photomultiplier tube (PMT), long-range alpha detection (LRAD), and the use of a compound semiconductor detector such as silicon carbide (SiC) [21]-[26]. In a fluorescent material like a Zinc Sulfide (ZnS) crystal, alpha particles produce scintillation for collection at the PMT. With the appropriate energy discrimination setting, a ZnS crystal is insensitive to beta particles and low energy gamma rays [21]. This detector has poor energy resolution, and the detection efficiency varies based on the alpha-particle energies [27].

A very standard approach for alpha-particle measurement has been the unique practice of ion-implantation combined with special passivation to realize the ion-implanted silicon detector [20]. As the implant dose and energy of ion-implantation are specified directly in the equipment settings, this practice is unique. These types of detectors are commercially referred to as passivated implanted planar silicon (PIPS) detectors, 'ULTRA' or 'IPC/IPA'. These detectors have found application for heavy charged-particle measurements, as they use reverse biased p-n junctions to collect charged carriers. The depletion width increases with increasing bias voltage, allowing measurements for heavy charged particles [4]. The PIPS detector has several advantages such as better energy resolution for alpha-particle spectroscopy, low electronic noise, and

higher geometric efficiency [20], [28], [29]. This type of detector suffers from relatively high cost when the sensitive areas grow above 20-cm^2. The PIPS detector has a maximum geometric efficiency of 45% [28], [30], [31].

Conversely, when using RICs, radiation detection is carried out in RSAs, which contain PN junctions. Alpha-particles interacting in the RSAs change the electrical properties of RSAs, which is registered by RHAs. With RSAs occupying ~75% of the total chip area, a higher geometric efficiency is expected when compared to traditional methods [1], [28]. Additionally, detectors utilizing RICs will be comparatively inexpensive when produced in bulk, since they use an unmodified IC fabrication process [1].

Semiconductor-based, lithium-drifted silicon Si(Li), and germanium Ge(Li) detectors have been developed for measurements of beta particle [16], [20], [28], [32]. These detectors are operated in laboratories to perform beta measurements. Since beta-radiation has more penetrating power than alpha particles, about 5-mm thick silicon PIN diodes are traditionally employed. For beta-particle detection, the Si(Li) detectors are favored over Ge(Li) detectors, as they have lower gamma-ray sensitivity and backscattering [28]. Based on the beta-radiation energy, the detection efficiency varies. To achieve high energy resolution, liquid-nitrogen cooling is used to reduce the thermal noise [32].

The design of a beta-particle survey probe that uses RICs is another element of this research. As RSAs are sensitive to charged particles, our research utilized simulated RICs to detect beta radiation directly. In any given material, the range (g-cm^{-2}) of a beta-particle of known energy can be obtained [5]. This range property can be exploited to determine beta-particle energy when it is detected at a known penetration depth. To measure the

maximum energy of a beta, a known attenuator material in a suitable configuration is required. Through this research, a beta survey probe for use in the field can be specially designed not only to detect beta particles but also to determine the maximum beta-particle energy. In this research, the analysis was carried out to assess the attenuation of betas in the attenuator material, which helps to estimate its maximum energy. When fabricated, this will be the only survey probe in the world that can perform in-field maximum beta-energy discrimination.

Our analysis also aims to evaluate the applicability of RICs to detect neutrons. Various research groups, such as McGregor, [33] have been working on the semiconductor-based neutron detector [33]-[37]. This group has studied neutron detection and dosimetry for over a decade. The basic semiconductor elements are from group IV of the Periodic Table - e.g. Si and Ge, and compound semiconductors are derived from groups III and V - e.g. GaAs and InP, and groups II and VI - e.g. CdTe. Montag synthesized and characterized LiZnP and LiZnAs semiconductor materials from the group III-V to highlight critical factors in the development of neutron detector-grade materials [34]. Though this type of detector is yet to be fabricated, research suggests that greater intrinsic efficiency compared to ^{3}He and $^{10}BF_3$ is possible [34]. Due to the shortage of ^{3}He, research efforts in the development of alternative neutron detectors have been promoted [38].

McGregor presents an analysis carried out on thin-film-coated detectors and solid-form or bulk detectors while employing boron carbide as the material to generate neutron interactions [33]. A similar analysis, using lithium fluoride as an external layer of neutron-reactive material on a silicon detector, was carried out by Barbagallo [35]. The neutron

interaction occurs in a sensitive film adjacent to a detector for thin-film-coated devices or inside the detector for solid-form detectors - theoretically, an intrinsic efficiency of 100% can be realized. McGregor used a Si diode for detection, and results show higher total counts for a solid-form device compared to a thin-film-coated device, since reaction products were captured inside the detector with a relatively small loss [33].

McGregor also presents a distinctive approach to micro-structured patterns etched deeply into the substrate, followed by backfilling of the neutron-reactive materials [39]. This technique was initially suggested by Muminov and later by Schelten, yet neither group worked to perfect it. Following McGregor's first demonstration of a micro-structured detector, this concept gained interest around the world. Here, semiconductor columns along with neutron-reactive material allow improved retention of reaction products compared to a coated detector and a larger reaction area. Lithium fluoride was found to be the most desirable neutron-reactive material for high counts, better noise rejection capability, and lower susceptibility to process damage [39]. This approach has shown neutron detection efficiency of about 40% which is significantly higher compared to traditional techniques [39].

Barbagallo also carried out a study examining a thermal neutron detector that employed lithium on scintillating fibers coupled to a photo-detector [40]. As the alpha and triton reaction products are emitted at 180º from each other, in this configuration a geometrical efficiency of close to 100% was possible. Barbagallo's approach struggles with variable energy losses in the fiber for different impact angles, resulting in a complex

design. For the proof of principle, this technique was implemented to highlight the importance of further developments [40].

Celik and Unlu conducted research on a neutron intercepting silicon chip to analyze soft error on memory devices to detect thermal neutrons [41]. The soft errors are the transient errors caused due to excess charge carriers. In the analysis, boron-enriched borophosphosilicate glass was introduced as an insulation layer in the semiconductor memories to increase neutron reaction rates. This increase corresponds to a soft error, which helps to determine the presence of neutrons. The neutron analysis was performed using GEANT4 and needs further study to evaluate its feasibility [41].

The neutron detection approach taken in this dissertation is similar to other methods. In our simulations, neutron-reactive materials were used to generate charged-particles to interact in the RSAs of the RIC to produce an electrical signal [33]-[44]. However, there are differences in how the RIC accomplishes neutron detection. For instance, in McGregor's effort, a coated p-n junction Si diode was employed as opposed to a PN junction used in the RSA. McGregor's coated diodes were fabricated using a technique similar to the process of fabricating thin-film-coated diodes, but the RIC uses a shielding metal layer on the RHAs [33], [1]. For radiation detection, PIN diodes outperform p-n junction diodes as they offer better bandwidth and dynamic range and better signal to noise ratio for the intense radiation environment [14], [20]. Also, preliminary RIC simulations have shown thermal neutron detection efficiency of about 24%, which is six times higher than the traditional efficiency [1]. This detector efficiency can be further improved by stacking the RICs.

A part of our research addressed the replacement of traditional PMTs, for which considerable efforts already have been made [45]-[57]. In the 1960s, the first semiconductor-based photodiodes were developed and soon offered an alternative to traditional PMTs [45]. A scintillator crystal and PMT can play a vital role in gamma-ray detection. A traditional PMT has a thin photocathode followed by dynodes that amplify the electrical signal initiated by the scintillation [4]. PMTs are vacuum tubes and have several drawbacks such as their limited ruggedness, sensitivity to magnetic fields, and their requirement of a continuous high-voltage power supply [45], [46]. A significant amount of research has been done to develop solid-state detectors that can replace PMTs, as they are small, rugged, provide a linear response, and have low voltage requirements [34], [45]-[56]. A key advantage of solid-state detectors is their high quantum efficiency (up to 90%) [46].

The main types of semiconductor-based gamma-ray detectors that have been developed or are under investigation are photodiodes, photoconductors, avalanche photodiodes (APDs), Geiger-mode APDs (GAPDs), and silicon photomultipliers (SiPM) [35], [57]. The reviewed semiconductor-based detectors include both photo-detectors and semiconductors in conjunction with a scintillator. The photodiode has a p-n junction structure or PIN structure, without any internal gain to amplify the generated signal [35]. The photoconductor is a highly reverse biased p-n junction that can amplify the signal [13], [15]. Photodiodes are faster than photoconductors but have no internal gain [15], [35]. APDs are specialized PIN diodes operated under high, reverse-voltage (photoconductive mode) to generate additional charge carriers following the primary charge carriers [35]. This process of amplifying charge carriers is called impact ionization. Donati has worked on

APD characterization, to compare them against PMTs [45], and Niclass has shown the application of APD in performing photon measurements [49]. The use of an impact ionization process allows APDs to achieve an internal amplification of about 20 to 200 times the primary signal carriers, but it needs a high operating voltage, suffers from high noise levels, and the output is non-linear [14], [35], [55].

A GAPD is essentially a p-n junction operated at a voltage above the breakdown level, which means a signal charge carrier can trigger a self-sustaining avalanche to provide a significant gain (~10^6) [48]. Thus, the response of a single GAPD is independent of incident light intensity, but when an array of GAPDs is used, the number of triggered GAPDs represents the light intensity. The array of GAPDs is termed as solid-state photomultipliers or SiPMs. Christian has worked on solid-state photomultipliers to examine their capability to replace PMTs [48], [51]-[53]. The drawbacks associated with this SiPM method are lattice defects, impurities in the material, and the dark current, which cannot be eliminated due to its thermal origin. The dark current can be mitigated by cooling the device, but reducing the detector temperature has the counter-effect of slowing down the detector response [55]. In an ample light application such as nuclear medical imaging, SiPMs have found application, since the dark current has negligible effects [47], [56]. Some of the recent SiPM-based products are Kromek's SIGMA (gamma-ray detector) and TN15 (thermal neutron detector), and Hamamatsu's MPPC (precision measurement) [58]-[60].

In this research, an analysis was also carried out to evaluate the capability of photocathode(s) and RICs to replace the traditional PMT [1], [61]. A RIC-based PMT will

have the advantage of higher quantum efficiency, low power requirement, simplified electronic peripherals, and reduced detector footprint. Similar to other semiconductor-based detectors, in the RIC-gamma ray detector, the dark current can exist, but its impact will be so small that it would not be detected by the RHA. Moreover, calibration can further address this noise. Compared to the traditional approach where a PMT is positioned on only one surface of a six-sided scintillation crystal, the proposed approach positions RICs along with photocathodes on all six crystal surfaces to perform gamma detection. The conventional approach of PMT placement on only one crystal surface is performed as it is an effective approach, considering detector operation and cost. The scintillator crystal generates optical photons which are converted to photoelectrons at the photocathode. These photoelectrons interact in RSAs of the RIC (which are heavily reverse biased PIN diodes). With photocathodes on all crystal surfaces, the collection efficiency for the proposed approach can be improved compared to the traditional approach [4], [1], [61]. The concentration of the electrons produced by the photocathode could stress the ability of the RHA to detect a gamma ray strike. As such, this is not a key focus of this dissertation.

Tools: MCNPX and GEANT4

The RIC has been designed to detect four types of radiation: alphas, betas, gamma rays, and neutrons. To analyze the particle interactions and propose the final detector designs, two widely known radiation transport codes, MCNPX (Monte Carlo N-Particle eXtended) and GEANT4 were employed. To determine how effectively the RIC technology can be used in a new type of alpha/beta survey probe and neutron detector,

radiation transport simulations in MCNPX were carried out. For the PMT replacement analysis (gamma rays), GEANT4, also, a particle transport tool was used.

MCNPX, a general-purpose radiation transport code, was used to model the interactions of beta particles and neutrons in relation to the RIC detector. MCNPX is popular as it has extended capabilities to simulate nearly all particles, using a broad range of energies for several applications [62]. The code can model time-dependent, three-dimensional problems. In MCNPX, the radiation source geometry can be varied, and various functions/tallies exist to calculate flux, current, and energy deposition [63]. The code uses a point-wise radiation interaction cross section for materials. Overall, MCNPX is a multipurpose tool, and it was employed here for the alpha, beta, and neutron transport calculations.

GEANT4, an object-oriented computer code implemented in the C++ programming language, was used in this work to analyze gamma-ray interactions and the subsequent production of optical photons through scintillation. Additionally, GEANT4 was useful in examining the transport of optical photons in the scintillation crystal. GEANT4 can simulate radiation transport for nearly all particles over a wide energy range and includes a complete range of functionality to handle complex geometries, composite materials, various physics processes, tracking, and hits [64]. A worldwide collaboration of physicists and software engineers continue to improve and expand the use of GEANT4. Since its release in 1998, the code has undergone continuous development and refinement to meet new standards and optimize its performance [64]. Overall, GEANT4 is a

comprehensive multipurpose toolkit that can efficiently simulate radiation transport problems such as our gamma-ray detection system.

CHAPTER II

ALPHA/BETA SURVEY PROBE

Overview

With the advancements and miniaturization in semiconductor technology, circuits on the chip have become increasingly sensitive to radiation damage and, therefore, fail to function properly in radiation fields. Researchers at Texas A&M University have been able to convert this disadvantage into an advantage by designing the radiation integrated circuit (RIC). The RIC has two regions, radiation-sensitive areas (RSAs) and radiation-hardened areas (RHAs) [61]. The RSAs are sensitive to charged-particle interaction, and the RHAs are hardened, with a high resistance to radiation damage [1]-[3]. When charged particles such as betas deposit their energy in the RIC, the electrical properties of RSAs change in a way which is detected and registered by the RHAs. The affected RSAs will recover to their normal functioning status within several nanoseconds (approximately 230-ns for gamma-ray detection and 40-ns for neutron detection, as mentioned earlier). With RSAs sensitive to charged-particles, RICs can be employed to detect alpha and beta particles. However, beta-particles have a higher penetrating power compared to alphas, which constrains the RIC to a lower detection probability for beta particles. Utilizing an attenuator material for the beta detection not only increases the detection probability, but it also provides the capability to perform maximum beta-particle energy discrimination in the field. This discrimination functionality, when realized, would be the first of its kind,

and in order to realize this potential, an analysis is required which was carried out using the Monte Carlo N-Particle eXtended (MCNPX) software.

Probe Design

The proposed alpha/beta probe consists of two regions, one to detect both alpha and beta particles, and the other to detect only betas (Figure 2.1). In the first probe region (the Alpha + Beta probe), no attenuator material is present between the source and the probe face. In the second region, an attenuator is present to enhance the detection efficiency of betas and perform the maximum beta-particle energy ($E_{\beta max}$) measurement.

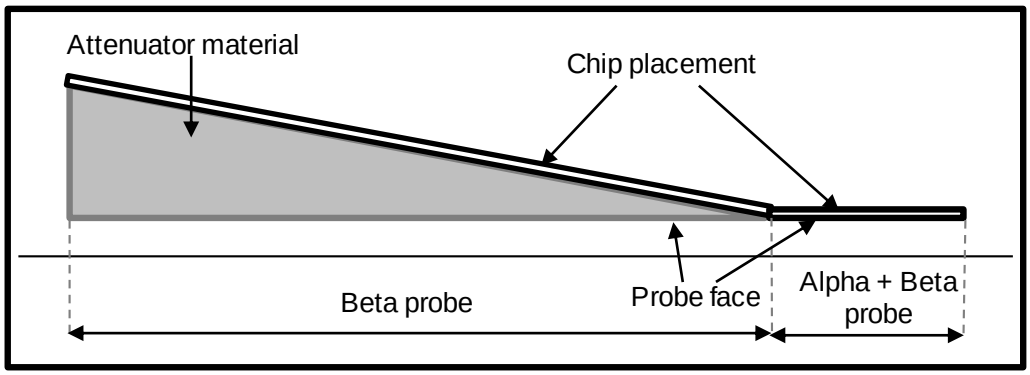

Figure 2.1. Schematic of the alpha/beta probe.

When the beta particles pass through the attenuator, they lose energy and thus, their probability to interact in RSAs increases. Alphas are heavy charged-particles and have lower penetrating power compared to betas. Hence, in the first region, RICs have higher detection efficiency for alphas relative to betas. Conversely because there is an

attenuator over the second region, alphas will be shielded by the attenuator, and only betas will be registered in the RIC region. Thus, in Figure 2.1, the first region is referred to as the alpha probe and the second region as the beta probe.

The range of betas in the matter is a function of their energy and the material on which they are impinging. In a particular material, the higher the beta-particle energy, the deeper the beta can penetrate [5]. To take advantage of this fact for the proof of concept, the beta probe is simulated with a wedged-shaped attenuator. When the probe is fabricated, RICs will be placed on the inclined surface of the wedge. In this arrangement, the attenuator thickness that the beta has to traverse is known over each RSA. The beta-particle emitted from the source, placed below the attenuator, can only interact in an RSA if it loses nearly all of its energy for interactions in the attenuator material. After traversing the maximum penetration depth (MPD), the beta's residual kinetic energy is low enough that it is likely to interact within an RSA rather than to escape the RSA without causing a measurable signal. The energy of the betas after traversing attenuation thickness slightly lower than the MPD ranges from 0 to the starting energy of the beta (i.e., the rare case of the beta experiencing zero interactions as it traverses the attenuator). However, the majority of attenuated betas have energy lower than a few hundred keV. The relationship between the energy of the betas and the signal strength (cross section) in the RIC RSA must be determined in future experiments. The signal generated in the RSA allows the RIC to estimate $E_{\beta max}$. When realized, this ability to perform $E_{\beta max}$ discrimination in-field would dramatically improve the capabilities of mobile hand-held survey equipment.

Figure 2.2 displays the working principle that was used to determine $E_{\beta max}$. In the figure, a beta source is depicted below the probe base, shown along with the wedge-shaped attenuator. A beta-particle with a known energy has a known MPD in a particular material. In the figure, an arc is shown that represents the MPD in the material. In Figure 2.2, the arrowed lines indicate the tracks of five identical beta particles that strike the beta probe at different angles. For beta-particles 1 and 5, the attenuator thickness is greater than the MPD (assuming linear penetration, which is a rough approximation used in working problems [5]). These particles will not reach the chip. For beta-particle 3, the attenuator thickness is less than the MPD. This beta, which will penetrate the wedge, can either pass through the chip without affecting an RSA or interact in the chip.

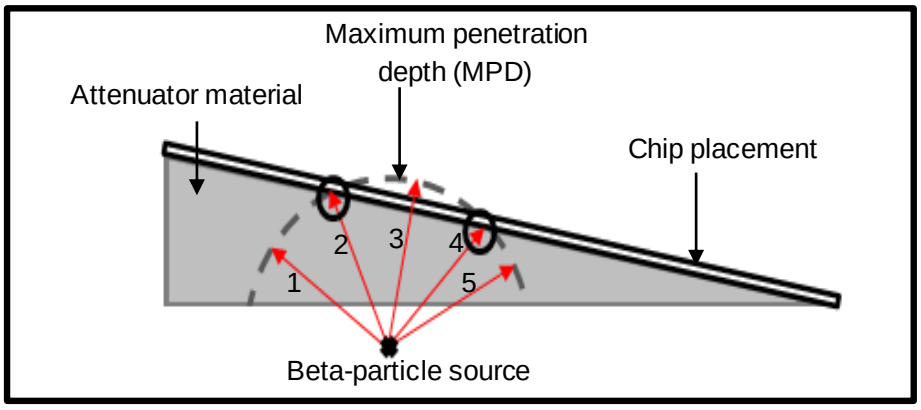

Figure 2.2. Illustrating maximum penetration distance for a beta particle.

For beta-particles 2 and 4 in Figure 2.2, the attenuator thickness is slightly less than the MPD in the material. These betas will be at the end of their track and mostly

slowed-down. As a result, they can interact in the RIC with the highest probability. The mostly slowed-down energy of betas, the desired energy range, will vary depending on the starting energy of the beta and attenuator material, but ranges from 0 to few hundred keV. The resulting interaction in the chip can generate a signal to detect the beta-particles [5]. The interaction probability in the RSA will be a function of the beta energy and can be determined experimentally in the future. When signals generated in the RICs in this fashion are evaluated, the $E_{\beta max}$ can also be estimated. Thus, regions where the attenuator thickness is slightly less than the MPD, are referred to as *the region of interest*. This region of interest is a factor of the beta source position, $E_{\beta max}$, and the attenuator material. For the attenuator, simulation analyses were focused on the region of interest as explained later in this chapter.

One of the objectives of the research is to propose the design of the alpha/beta survey probe. In this chapter, the analysis carried out on the revolutionary beta probe design is presented. In the analysis, a few attenuator materials were investigated. The low-Z materials considered for this research included plate glass, Pyrex® glass, Lucite®, and natural rubber. For the attenuator, low-Z materials were assessed so that the designed survey probe is light-weight and easy to carry. Several simulations were executed to assess the energy discrimination capability by varying the source position and $E_{\beta max}$. The alpha-particle simulations and analysis are a part of the neutron detector analysis, which relies on the generation of secondary alpha particles. Hence, in this chapter, an optimal design for the beta probe is proposed that offers a superior energy discrimination capability.

Material and Methods

The beta survey probe in the form of a wedge was modeled in MCNPX. The low-Z attenuators used in these simulations were plate glass, Pyrex® glass, Lucite®, and natural rubber (Appendix A). A light-weight beta probe is important to the user because as a hand-held field deployable system it can be easily carried. Table 2.1 lists the dimensions for each attenuator and their respective densities.

Table 2.1. Dimension and density details for each attenuator material along with the corresponding beta source position.

Attenuator material	**Wedge dimension [cm]**			**Density [g-cm^{-3}]**	**Source position (X,Y,Z) [cm]**
	Length (X-axis)	Height (Y-axis)	Width (Z-axis)		
Plate glass	5	0.5	5	2.4	(3.95, -1, 2.5)
Pyrex® glass	5	0.5	5	2.23	(4.26, -1, 2.5)
Lucite®	7	1	5	1.19	(5.58, -1, 2.5)
Natural rubber	7	1.1	5	0.92	(6.56, -1, 2.5)

The X, Y, and Z values in Table 2.1 represent the attenuator's length, maximum thickness, and width (Figure 2.3). For example, the plate glass attenuator was simulated 5-cm long by 5-cm wide, with the thickness of wedge linearly increasing from 0 to 0.5-cm. These dimensions were selected to enable the $E_{\beta max}$ discrimination analysis for 2-MeV

beta particles. For each attenuator material, the linear stopping power for betas varies, and to accomplish this discrimination analysis, the dimensions were fine-tuned accordingly.

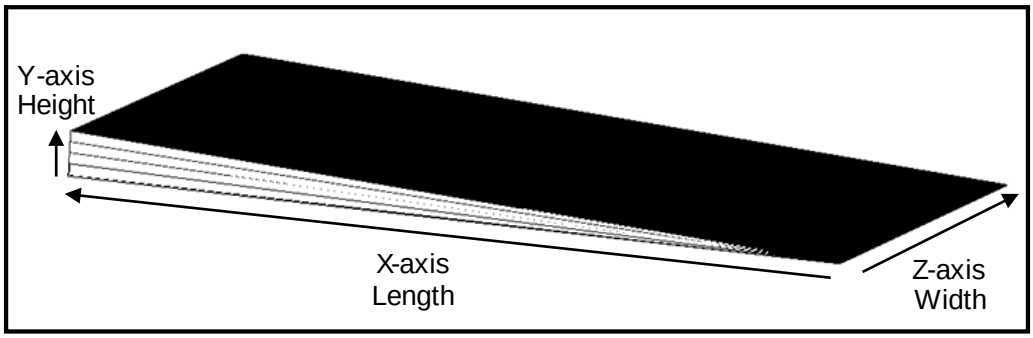

Figure 2.3. A reference coordinates axis for the beta probe.

The simulations were carried out to evaluate the maximum beta-particle energy discrimination capability of each attenuator material. For this assessment, the source position was fixed, and the starting energy of betas was varied while analyzing attenuator effectiveness. For each material, MCNPX simulations used an isotropic source that emitted monoenergetic betas. Simulations were run that varied beta energy by 0.01-MeV from 1.95 to 2.1-MeV. In the analysis, $E_{\beta max}$ discrimination capability was assessed for the 2-MeV beta. The source position was set at a point where the attenuator thickness equaled the MPD for the 2-MeV beta in the material under consideration (Table 2.1). Furthermore, the source was placed 1-cm below the attenuator. The beta energy in the above simulations represent the maximum energy of the beta emitted [5]. However, the beta-decay is a three-body problem, and thus, a beta has a continuous energy spectrum.

This spectrum can be predicted by the Fermi theory [66]. The detail of simulations employing the continuous energy beta source is provided later in this chapter.

A method was developed to tally attenuated betas on the inclined surface of the wedge (XZ plane in Figure 2.3). Based on the source location, placed under the beta probe containing the attenuator material in the wedge shape, and the maximum beta-energy it emits, the beta attenuation will vary. This variation in attenuation is representative of the source and attenuator material and its dimensions. For each simulation in MCNPX, the number of attenuated betas were tallied in twenty-five 1-mm^2 areas (on the inclined surface of the wedge) which were randomly picked across the probe length. The tally areas are selected to tally the count of attenuated betas crossing a 1-mm^2 area per starting beta-particle (β) [# β^{-1} mm^{-2}].

Each scoring/tally area is a square and contains 40,000 RSAs. Each 1-mm^2 scoring area was bounded by two lines in both the X and Z-planes (Figure 2.3). Two lines along the Z-plane were conceptualized at Z=2.45 and 2.55-cm, a separation of 1-mm, while two lines along the X-plane were theorized at various positions such that there were twenty-five 1-mm intervals. For each attenuator material, these intervals were selected to assess $E_{\beta max}$ discrimination capability with reference to the 2-MeV beta; as listed in Table 2.2. After preliminary runs in MCNPX, for each attenuator material, the twenty-five 1-mm intervals along the X-axis were selected based on the region of interest.

Table 2.2. Twenty-five 1-mm^2 intervals along the X-axis.

Intervals along the X-axis (units in cm)							
Plate glass		Pyrex® glass		Lucite®		Natural rubber	
0.0-0.1	3.3-3.4	0.0-0.1	3.4-3.5	0.0-0.1	4.6-4.7	0.0-0.1	5.3-5.4
0.3-0.4	3.4-3.5	0.3-0.4	3.5-3.6	0.5-0.6	4.7-4.8	0.5-0.6	5.4-5.5
0.7-0.8	3.5-3.6	0.7-0.8	3.6-3.7	1.1-1.2	4.8-4.9	1.1-1.2	5.5-5.6
1.1-1.2	3.6-3.7	1.1-1.2	3.7-3.8	1.5-1.6	4.9-5	1.5-1.6	5.6-5.7
1.5-1.6	3.7-3.8	1.5-1.6	3.8-3.9	2.1-2.2	5-5.1	2.1-2.2	5.7-5.8
2.3-2.4	3.8-3.9	1.9-2.0	3.9-4.0	2.5-2.6	5.1-5.2	2.5-2.6	5.8-5.9
2.6-2.7	3.9-4.0	2.3-2.4	4.0-4.1	3.1-3.2	5.2-5.3	3.1-3.2	5.9-6
2.7-2.8	4.0-4.1	2.7-2.8	4.1-4.2	4-4.1	5.3-5.4	4.1-4.2	6-6.1
2.8-2.9	4.1-4.2	2.9-3	4.2-4.3	4.1-4.2	5.4-5.5	4.5-4.6	6.1-6.2
2.9-3	4.2-4.3	3-3.1	4.3-4.4	4.2-4.3	5.9-6.0	4.9-5	6.2-6.3
3-3.1	4.3-4.4	3.1-3.2	4.7-4.8	4.3-4.4	6.5-6.6	5-5.1	6.4-6.5
3.1-3.2	4.7-4.8	3.2-3.3	4.9-5.0	4.4-4.5	6.9-7.0	5.1-5.2	6.9-7.0
3.2-3.3		3.3-3.4		4.5-4.6		5.2-5.3	

The linear stopping power for betas in each attenuator material varies, and as a consequence, the region of interest varies as well. In this region, along the X-axis, the attenuation thickness that a beta passes through is slightly less than the MPD. This difference can be from a fraction of millimeter to a few millimeter, based on the tally scoring region and attenuator material.

For each attenuator material, the $E_{\beta max}$ discrimination capability is with reference to the 2-MeV beta. Thus, results obtained from the set of simulations in which the beta source energy was varied from 1.95 to 2.1-MeV were normalized by a reference result, selected from the 2-MeV beta simulation. This normalization allowed characterizing an attenuator for the energy discrimination capability. For the reference value, the highest count of attenuated betas scored in a tally area from the 2-MeV beta simulation was selected.

All of the measurements obtained from the 1.95-MeV to 2.1-MeV beta simulations were normalized to the reference value. The quotients were ratios of counts, representing the difference in counts with respect to the highest count from the 2-MeV simulation. The attenuated beta counts directly reflect the strength of the signal generated in the respective RSA. The greater the difference in counts, the better the prospect of performing $E_{\beta max}$ discrimination. The factors (quotients) calculated for each attenuator material were compared to determine the optimal attenuator material that possesses a superior energy discrimination capability.

MCNPX simulation results were compared against analytical results, obtained using Equation 2.1. This equation requires a beta emission rate, the angle subtended by the 1-mm^2 area to an isotropic point source, and the linear attenuation coefficient for beta particles. This coefficient was calculated using Thummel's empirical formula [65] (Equation 2.2).

$$\#\ Betas\ Reaching\ Chip\ (x_{att}) = \frac{1}{4\pi(x_{air}+x_{att})^2} * A * e^{-\mu_{air}x_{air}} * e^{-\mu_{att}x_{att}} \quad \text{(Eq. 2.1)}$$

$$\mu = \rho * \ 15.2(Z)^{4/3} \frac{1}{AE_m^{1.485}} \quad \text{(Eq. 2.2)}$$

In Equation 2.1, x_{air} is the attenuation thickness in the air; x_{att} is the thickness in the attenuator material; μ is the linear attenuation coefficient for a material; *A* is the 1-mm^2 area on the inclined surface of the wedge. In Equation 2.2, E_m signifies the maximum beta energy in MeV of a beta; *Z* is the atomic number of the target material; *A* is the atomic weight of the target material, and ρ is the density of the material. Simulations were run for each attenuator and results were compared against the analytical results obtained using Equation 2.1. This comparison was carried out to examine how closely the results match. The analytical calculation uses the empirical formula, as a first principle model is not available. Although the numerical simulations have their inherent limitation based on the physics parameters available to perform analysis, it is still a good approximation to the working problem. These two analyses were used to assess the beta attenuation and thereby characterize the attenuator materials. This portion of the research focused on comprehending the attenuation of betas through an attenuator material to investigate the capability of the RIC to discriminate $E_{\beta max}$ and subsequently, determine the optimal attenuator material.

Once the most efficient attenuator was determined based on its energy discrimination capability and light-weight, two sets of simulations were carried out. In the first set of simulations, the end-point energy of the beta was varied. The continuous energy beta particle source was modeled using the Fermi theory [66] (Equation 2.3). This analysis was carried out to examine the ratio of counts when a continuous energy beta source was utilized instead of a monoenergetic beta source.

$$N(T_e) = \frac{C}{c^5}(T_e^2 + 2T_e m_e c^2)^{0.5}(Q - T_e)^2(T_e + m_e c^2) \qquad \text{(Eq. 2.3.)}$$

In Equation 2.3, $N(T_e)$ represents the number of beta particles in the energy range of T_e and $T_e + dT_e$ per decay, C is the overall normalization constant, c is the speed of light, Q is the end-point energy, and m_e is the mass of the electron. The end-point energy (Q) was considered as 1.95, 2, and 2.1-MeV to compare obtained results with the monoenergetic beta source simulation results. For each Q value, the spectrum was projected over twenty uniform energy intervals. For the T_e value, an average energy of the interval was used. For example, with the Q value as 2-MeV, and interval width as 0.1-MeV, the first T_e value is 0.05-MeV. Figure 2.4 displays the continuous energy spectrum for the end-point energy of 2-MeV. The Y-axis value indicates the probability of beta-particle decay in an energy range. The probability of the source to decay by emitting the 2-MeV is greater than zero (on the order of 0.001), but to the naked eye, it seems zero. For a 1-mCi isotropic source, placed 1-cm away, the number of betas with the 2-MeV end-point energy crossing a 1-cm^2 surface area per second will be about 10^3, ignoring the attenuation in the material. When considering the material attenuation and counting for an interval of about 60-sec, the attenuated beta counts can be enough to generate a suitable signal in the RIC RSA. Analysis of the first set of simulations helped to understand the beta attenuation when using a continuous energy beta source.

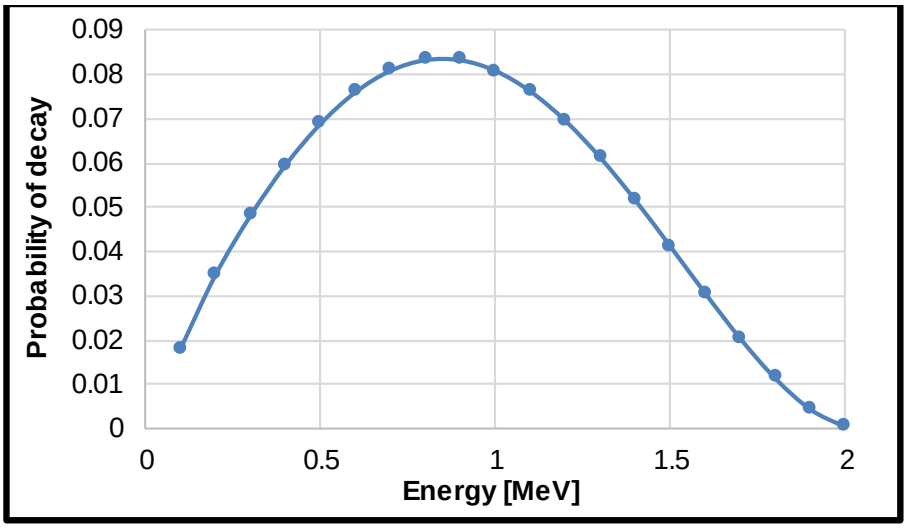

Figure 2.4. The continuous energy spectrum for the beta particle having an end-point energy of 2-MeV obtained using the Fermi theory (Equation 2.3).

The second set of simulations focused on investigating the beta attenuation by varying the source position. The source position was changed along the length of the probe from 0-cm to its end over an interval of 1-cm while centered and 1-cm away from the probe face. For each source position, an isotropic monoenergetic beta source of 2-MeV was assumed. A monoenergetic source was used to speed up the computation. The source movement analysis was carried out to understand the probe performance when used in a sweeping motion. These are rough simulations of a probe being used in a contamination survey for health physics purposes.

Results and Discussion

In MCNPX, plate glass, Pyrex® glass, Lucite®, and natural rubber were simulated as attenuator material in a wedge shape to characterize the performance of a RIC-based

survey instrument. In each simulation, the number of beta particles crossing the attenuator and eventually entering sensitive regions of the RIC was tallied. The energy of these attenuated betas varies from the 0 to end-point energy depending on the location of the tally area across the probe length. However, as described earlier, for each attenuator material in the region of interest, the betas tallied will be within the desired energy range being at the end of their track. For these betas, the probability of interaction in the RSAs will be highest, resulting in the signal generation in the RIC to detect their presence.

Plate Glass

For the plate glass, the source position was (3.95, -1, 2.5) in units of cm. The numbers of beta particles tallied across the twenty-five 1-mm^2 areas for the starting energy of betas as 1.95, 2, and 2.1-MeV, respectively, are presented in Figure 2.5.

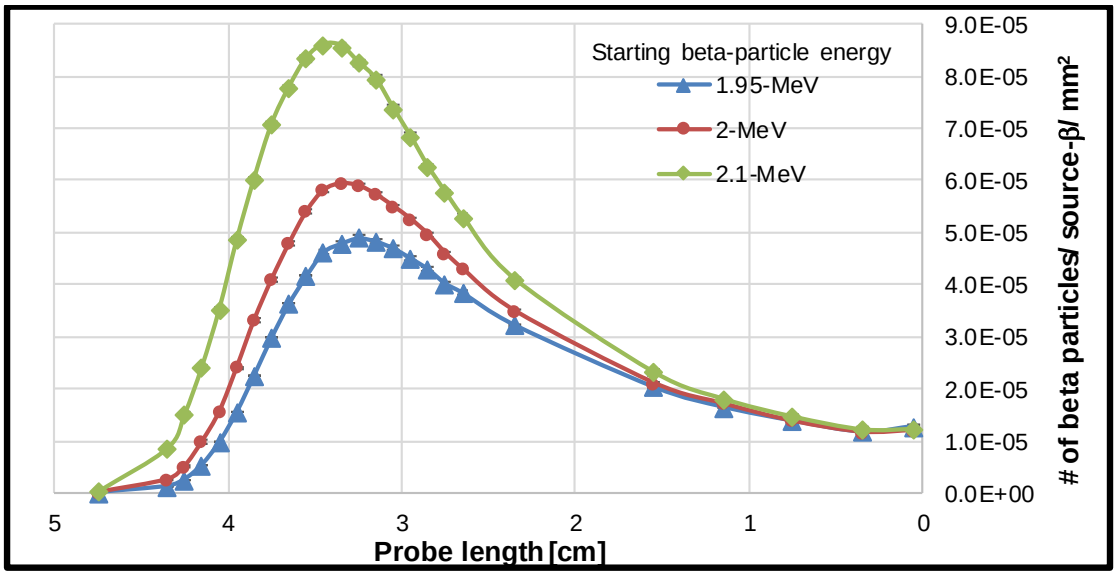

Figure 2.5. The number of beta particles tallied across all 1-mm^2 tally areas along the length of the probe with a plate glass attenuator.

The count across each 1-mm^2 area is per starting beta particle [# β^{-1} mm^{-2}] (Figure 2.5). The corresponding relative error for each data point was under 7%. The result for a tally region was plotted at the respective center; as presented in Figure 2.5. A smooth curve fit has been shown to highlight the trend in results. For other starting energies of betas, counts were tallied across all tally areas, and their trends were similar to the presented data. The trend in simulation results (Figure 2.5) depends on the distance a beta has to traverse through the plate glass attenuator. Figure 2.6 illustrates calculated distances a 2-MeV beta has to travel through the attenuator. For these calculations, the source was placed at (3.95, -1, 2.5) units in cm, which is the same location as used in simulations.

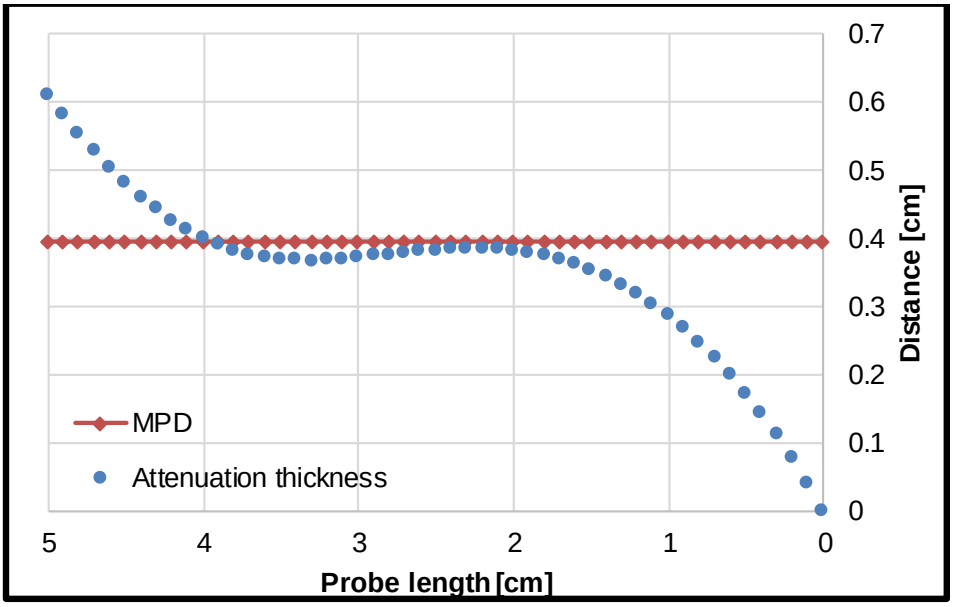

Figure 2.6. The attenuation thickness a beta traverses from the source position 3.95, -1, 2.5-cm to reach the inclined plate glass wedged attenuator surface varies along the probe length.

The attenuation thickness is the distance a beta has to traverse from its source position to reach the inclined wedge surface adjacent to the RIC (Figure 2.6). This thickness is a property of the probe length and source position, fixed in this case. For example, a beta traversing normal to the attenuator base has to traverse a 0.395-cm of plate glass to reach the RIC at the far end (inclined surface of the attenuator).

The MPD for a 2-MeV beta in plate glass is 0.395-cm (Figure 2.6). The attenuation thickness for a beta from the source at (3.95, -1, 2.5) in units of cm was less than the MPD between 0 and 3.95-cm of the probe length. However, between 2.65-cm and 3.95-cm of the probe length the attenuation thickness for a beta is only slightly less than the MPD. The betas after traversing this distance will be at the end of their track and being within the desired low energy range can readily interact in the RIC RSA. The difference between distance traversed and MPD ranged in the fraction of a millimeter based on the tally scoring region (and attenuator material). Similar trends were observed for other beta starting energies.

For the plate glass attenuator, the region between 2.65-cm and 3.95-cm of the probe length is the region of interest for the $E_{\beta max}$ discrimination analysis with reference to the 2-MeV beta. The defined probe region encompasses maxima for all starting energies of betas (from 1.95-MeV to 2.1-MeV). Figure 2.7 is a zoom-in view of Figure 2.5, presenting results between 2.65-cm and 3.95-cm of the probe length. The trend for the attenuation thickness (Figure 2.6) in the region of interest was convex upward. In this region, there is a local minimum which corresponds to the maximum number of attenuated

counts. For the 2-MeV beta, the peak occurs at 3.35-cm into the probe length with the least attenuation thickness to traverse.

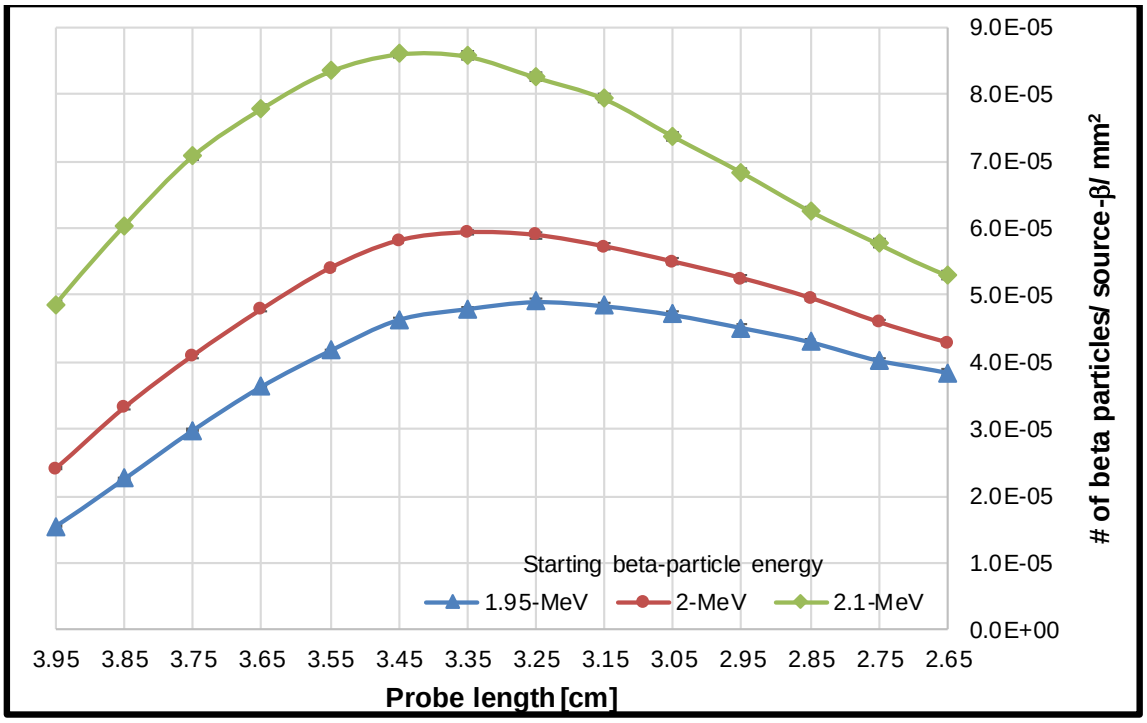

Figure 2.7. An enlarged view of Figure 2.5 for the region of interest 2.65-cm to 3.95-cm.

The analytic Equations 2.1 and 2.2 were also used to calculate the number of beta particles crossing the attenuator material, i.e., unattenuated betas with their original energy. The resulting trend was similar to Figure 2.7, but with a difference in count magnitude. This difference is because simulations count both unattenuated and the slowed down or scattered beta particles, but the empirical formula of Thummel (Equation 2.1) calculates only the unattenuated particles. Moreover, as the attenuation thickness that

betas traverse increases to more than one mean free path (inverse of the linear attenuation coefficient, Equation 2.2) the beta scattering/slowing down becomes substantial [5]. Besides, betas due to their small mass and charge are easily scattered/slowed down [67]. This scattering was understood by scoring the beta counts for a specified energy bin containing the starting energy of the betas, i.e., 1.9 to 2.1-MeV.

In the energy bin analysis, the results from the simulations and the analytical calculation for the unattenuated beta counts differed less than by a factor of four. This difference is acceptable and exists as the numerical simulations counted betas between the 1.9 and 2.1-MeV rather than only the 2-MeV betas. The energy bin was considered to tally betas as the probability of tallying a beta having a fixed energy (in this case, the 2-MeV) is zero. In addition, the beta traverses attenuation thickness greater than a mean free path which impacts the calculations carried out using the empirical formula. The trend noticed in results was similar to Figure 2.7 with the concave shape while having the counts peak at 3.45-cm into the probe length.

Figure 2.8 shows ratios of counts obtained for tally areas between the 2.65 and 3.95-cm probe length as a function of starting energy of the beta. The highest count from the 2-MeV beta simulation used as a reference value was at 3.35-cm into the probe length. Results obtained from the 1.95 to 2.1-MeV beta simulations were normalized to this reference value to examine factors, representing the difference in counts. These factors are the function of the attenuator material, beta energy, and attenuation thickness, which is a function of the probe length. Subsequently, the measurement in a tally region depends on the attenuation thickness a beta has to traverse. For the plate glass attenuator, a minimum

factor of 0.26X and a maximum factor of 1.45X were observed for the 1.95-MeV and 2.1-MeV beta simulation, respectively.

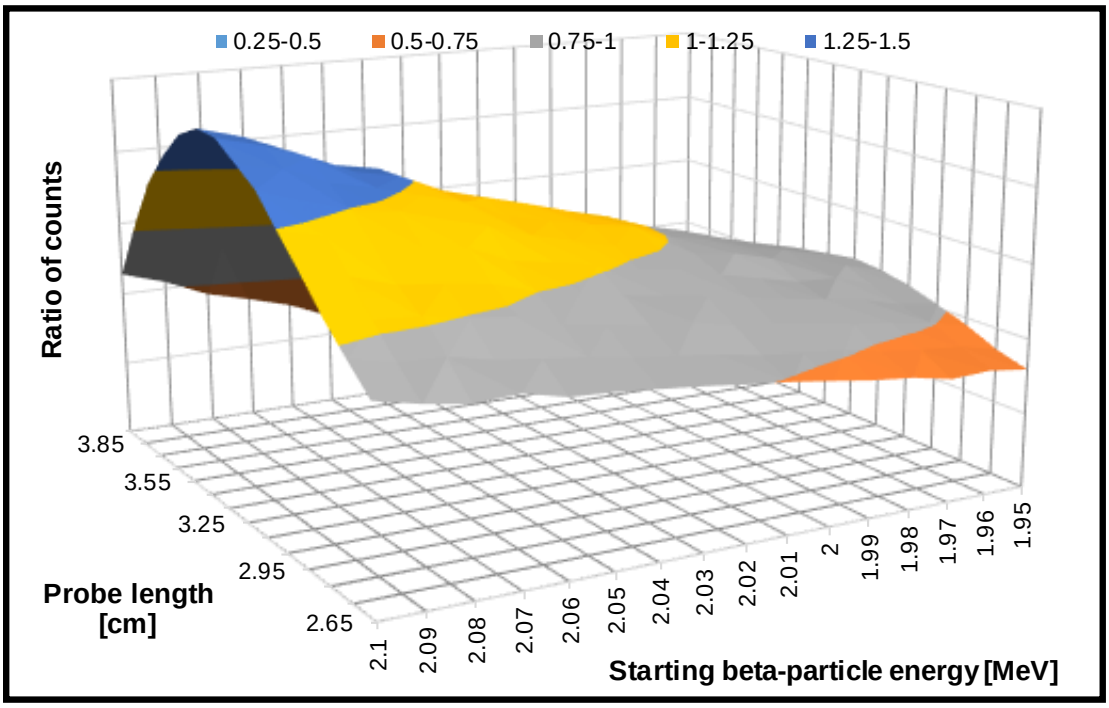

Figure 2.8. Factors representing differences in counts with respect to the reference value obtained from the 2-MeV beta simulation when using the plate glass attenuator. These factors are presented as a function of the starting energy of beta and probe length.

The larger the difference in counts in adjacent tally regions, the better the potential to perform $E_{\beta max}$ discrimination. The minimum factor for the 1.95-MeV simulation was observed at 3.95-cm into the probe length as betas had to traverse the largest attenuation thickness to reach this tally. Conversely, for the 2.1-MeV simulation, betas had to traverse

the least attenuation thickness to reach the tally region at 3.45-cm into the probe length, resulting in the maximum factor. The ratio of counts represents the difference in beta counts interacting in the respective RSA with respect to the tally RSA that measures the highest count from the 2-MeV simulation. The beta count drives the signal generated in the RSA. The larger the difference in counts the easier the signal processing and thus, the better the energy resolution capability for an attenuator.

Figures 2.9, 2.10, and 2.11 display an enlarged view of results for Pyrex® glass, Lucite®, and natural rubber. The simulation results are the beta count tallied across 1-mm^2 areas per starting beta [# β^{-1} mm^{-2}] and are presented as a function of the beta energy and probe length.

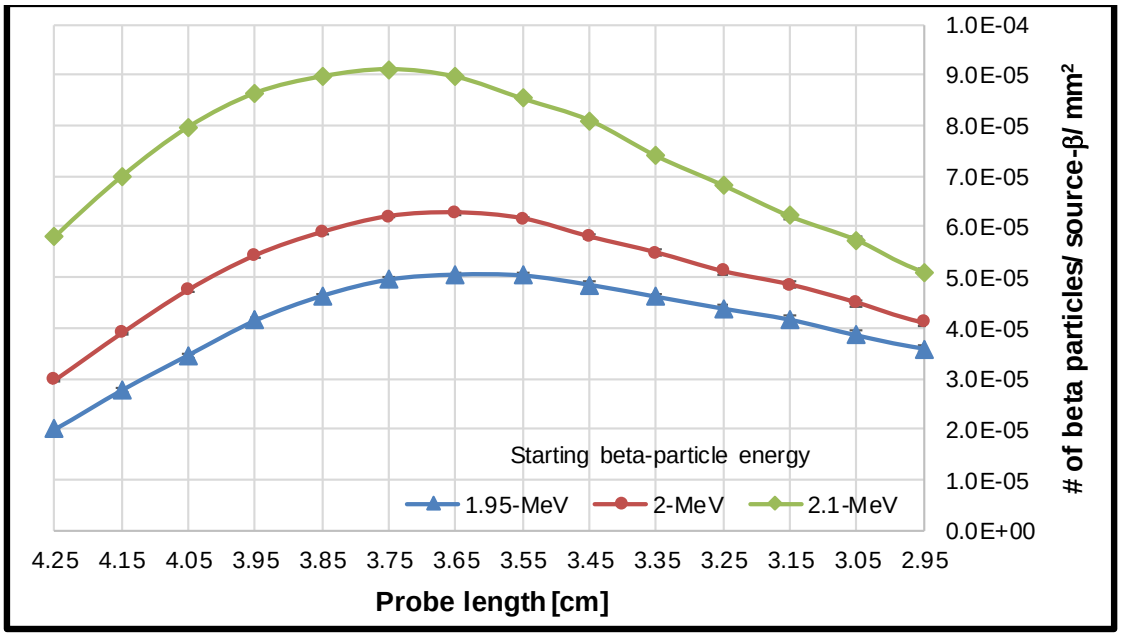

Figure 2.9. The number of beta particles tallied in the region of interest along the probe length when using the Pyrex® glass attenuator.

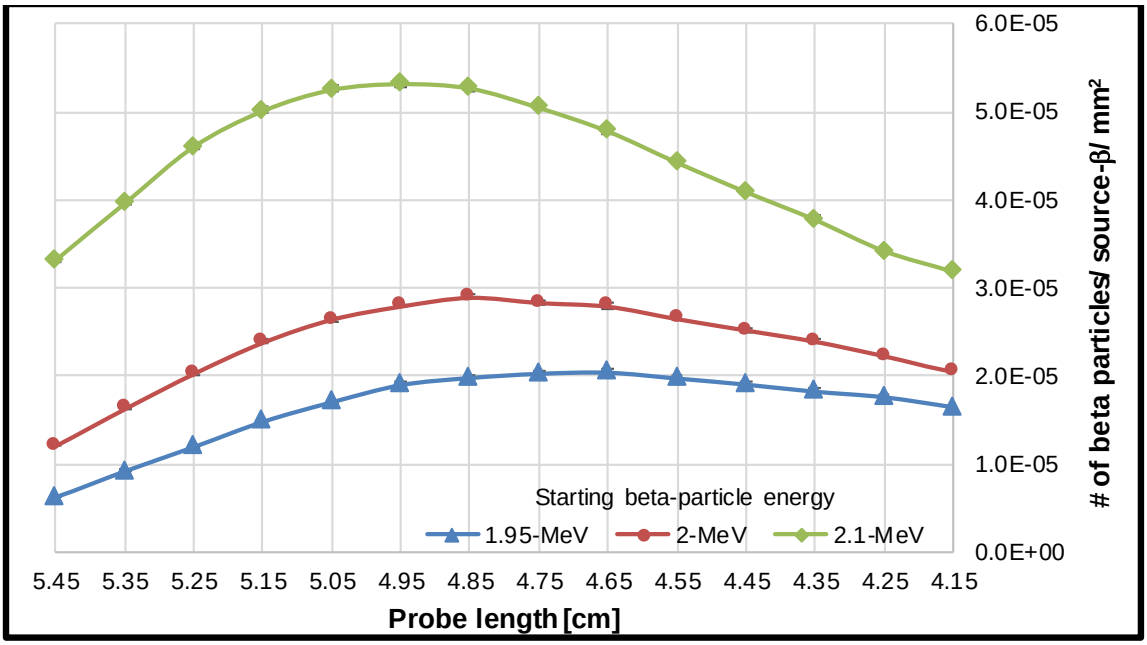

Figure 2.10. The number of beta particles tallied in the region of interest along the probe length when using the Lucite® attenuator.

The results in Figures 2.9, 2.10, and 2.11 are presented for a beta starting energy of 1.95, 2, and 2.1-MeV, respectively. The source position for each attenuator material is listed in Table 2.1. The trends in these figures were similar to Figure 2.7. The key difference was the difference in the MPD for the beta based on the attenuator material and its starting energy. The MPD for the 2-MeV beta is 0.426-cm, 0.798-cm, and 1.032-cm for Pyrex® glass, Lucite®, and natural rubber, respectively. Thus, the region of interest for energy discrimination analysis will be different in these attenuator materials. For the $E_{\beta max}$ discrimination analysis with respect to the 2-MeV beta, the region of interest were 2.95-cm to 4.25-cm for Pyrex® glass, 4.15-cm to 5.45-cm for Lucite®, and 4.95-cm to 6.25-cm for natural rubber.

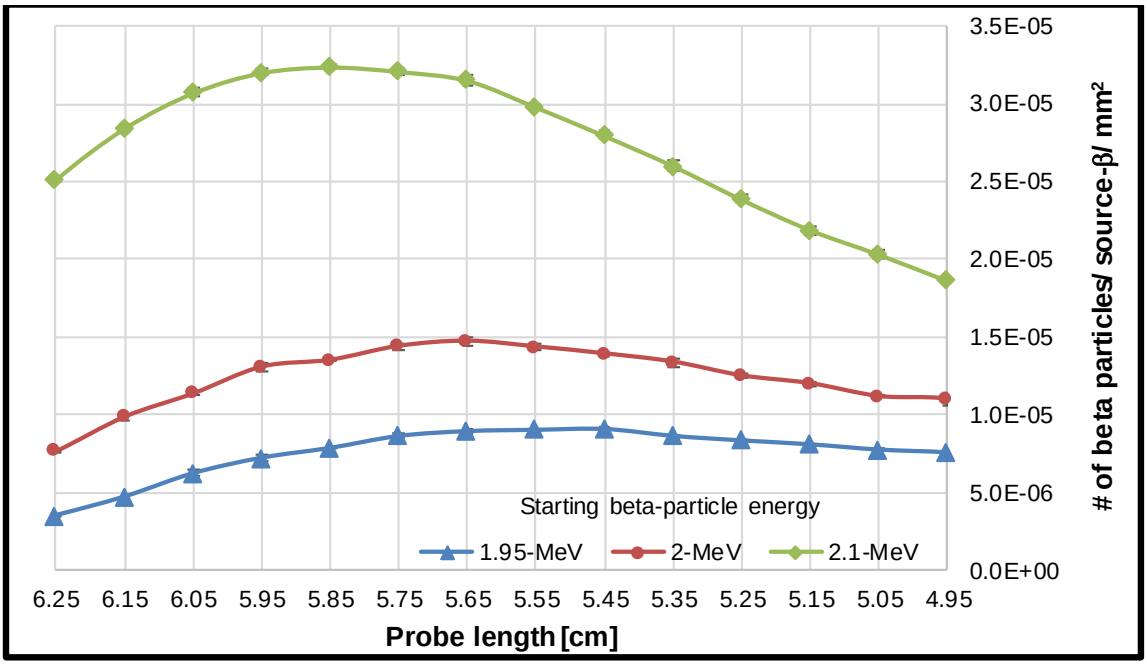

Figure 2.11. The number of beta particles tallied in the region of interest along the probe length when using the natural rubber attenuator.

Similar to the ratio of counts analysis for plate glass attenuator, analyses were carried out on the Pyrex® glass, Lucite®, and natural rubber. The trend of ratios as a function of starting energy of beta and probe length for these attenuators was similar to Figure 2.8. Table 2.3 lists the ratio of counts extremes from the region of interest for the Pyrex® glass, Lucite®, and natural rubber attenuator material. Natural rubber was observed to show the highest difference in the ratio of counts with a minimum factor of 0.24X and a maximum factor of 2.2X. Lucite® as an attenuator material had the smallest

minimum factor of 0.22X. However, the maximum factor for Lucite® in relation to natural rubber was lower by a factor of 0.83X.

Table 2.3. The ratio extremes for the Pyrex® glass, Lucite®, and natural rubber as an attenuator.

Ratio extremes	**Pyrex® glass**	**Lucite®**	**Natural rubber**
Minimum	0.32	0.22	0.24
Maximum	1.45	1.84	2.2

The highest difference in the ratio of counts property of natural rubber can be understood as it had the lowest effective atomic-number (5.41) and mass-number (10.71) with the highest weight fraction of hydrogen among other attenuators. Hydrogen presence enhanced the attenuation of betas which thereby offered a high dependence on its energy [5], [6]. With the beta count driving the signal generated in the RSA, the larger the difference in counts, the easier the signal processing. These collective properties contribute to achieve a superior energy resolution when using the natural rubber attenuator in the beta probe. For the difference in $E_{\beta max}$ of 50-keV, roughly a factor of two difference in counts was observed. This difference in counts can be exploited using the RSAs, which are a few μm in width. Thereby, there exists a potential to perform $E_{\beta max}$ discrimination within 50-keV using natural rubber wedged attenuator.

A natural rubber attenuator wedge was an optimal design with respect to a superior $E_{\beta max}$ discrimination capability and weight. A light-weight beta probe is important to the user because as a hand-held field deployable system it offers the ability to be easily carried. The wedge design was 7-cm in length, 5-cm in width while the height linearly increased from 0 to 1.1-cm. This design can measure $E_{\beta max}$ up to 2.1-MeV. For covering a broader $E_{\beta max}$ range, the length and height need to be extended. An option to realize a compact design while using the natural rubber attenuator is at the cost of $E_{\beta max}$ range. For the $E_{\beta max}$ range of 0.5 to 2.1-MeV, the height (thickness) would linearly increase from 0.18 to 1.1-cm while the length would reduce to 5.87-cm. Alternatively, by reducing the $E_{\beta max}$ discrimination capability, a compact design using plate glass attenuator can be realized, as it is only 5-cm in length and width while the height linearly increases from 0 to 0.5-cm. For this compact design, the weight of plate glass attenuator was lower by a factor of 0.84X the natural rubber attenuator. However, the $E_{\beta max}$ resolution capability for plate glass attenuator was lower by a factor of 0.92X natural rubber.

Using natural rubber wedged attenuator as the optimal beta probe design, two sets of simulations were carried out. The first set of simulations used a continuous energy beta source (placed at (6.56, -1, 2.5) with units in cm) with end-point values of 1.95, 2, and 2.1 MeV. Figure 2.12 presents these results along the probe length between 4.95 and 6.25-cm, the region of interest for natural rubber attenuator. The relative errors (1- σ) are included and are less than 9%.

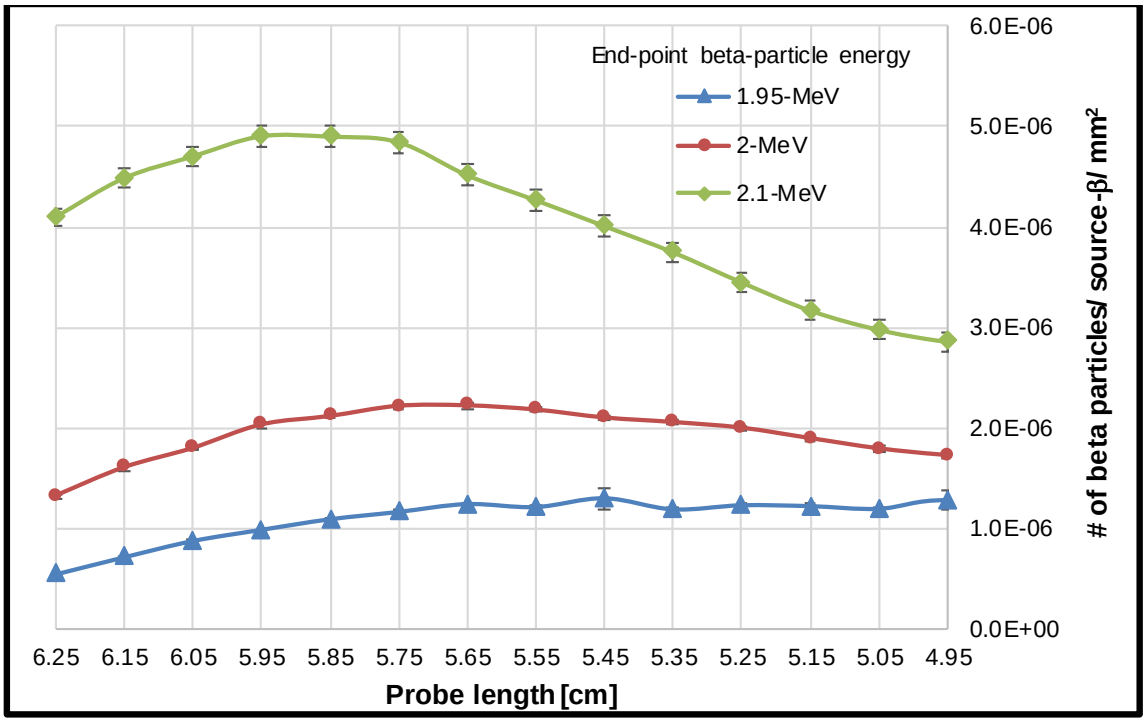

Figure 2.12. The number of beta particles tallied in the region of interest along the probe length when using natural rubber attenuator.

The trends in Figure 2.12 were similar to the trends observed in Figure 2.11. The beta counts were normalized to the highest count from the simulation that used the beta of 2-MeV as the end-point energy. The ratio of counts had a minimum factor of 0.24X and a maximum factor of 2.2X. These factors (ratios of counts) had the same magnitude as factors obtained using the monoenergetic beta-particle simulations (Table 2.3). However, counts were lower when using a continuous energy beta source compared to the monoenergetic beta source, as visible in Figures 2.11 and 2.12. This observation was as expected since the continuous energy beta source had a lower probability of starting the beta-particle with the end-point energy. As the ratio of counts is primarily necessary to

differentiate the end-point value of beta-particle ($E_{\beta max}$ discrimination), the selection of the type of source was immaterial. The ratio of counts represents the difference in the signal strength generated in the corresponding RSA with respect to the reference RSA. These differences allow the RIC to effectively perform the $E_{\beta max}$ discrimination.

The second set of simulations was carried out for the natural rubber by varying the source position to assess the RIC probe's $E_{\beta max}$ discrimination capability during a survey. The source centered on the probe width was moved along the probe length from 0-cm to 7-cm over an interval of 1-cm while being 1-cm below the probe face (Figure 2.13).

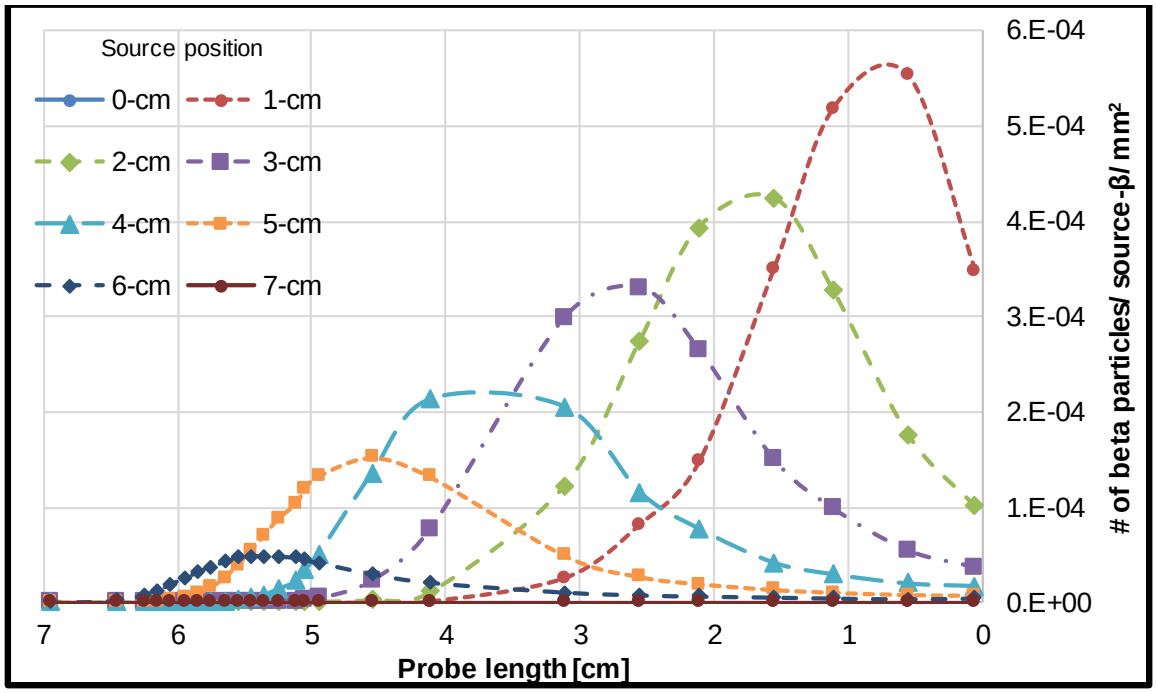

Figure 2.13. The number of beta particles tallied across the pre-defined tally areas along the length of the probe with the parameter of beta source position when using the natural rubber attenuator.

Figure 2.13 presents the results of the simulations as a function of the source position along the probe length. The results are the number of beta-particles tallied across the 1-mm^2 area per starting beta-particle of 2-MeV. The tally area that measures the highest beta count in a simulation was observed to depend on the source position along the probe length. The count of betas for source positions 0-cm, the beginning of the wedged probe with the least attenuation thickness and 7-cm, the end of the probe with the maximum attenuation thickness are not visible. These counts are not visible as the peak count from source positions 1-cm to 5-cm were larger as seen in Figure 2.13. This difference occurs because the attenuator thickness a beta has to traverse to reach the inclined wedge surface varies as a function of the source position. The thickness a beta traverses to reach the inclined surface as a function of the probe length and source position is shown in Figure 2.14.

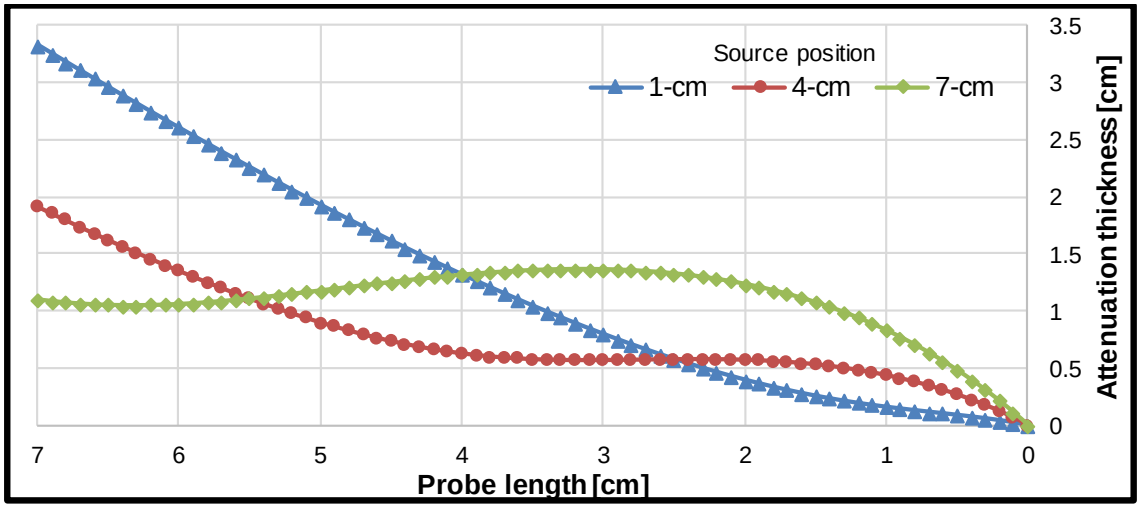

Figure 2.14. Variation in the attenuation thickness (natural rubber) a beta traverses to reach the inclined surface as a function of the probe length; parameters show source position.

The attenuation thickness across the wedged probe is presented in Figure 2.14 for the source positions of 1-cm, 4-cm, and 7-cm into the probe length. This variation in thickness is similar to Figure 2.6 for plate glass attenuator. With the change in the source position, the attenuation thickness for the beta (and so, the region of interest) varies resulting in the movement of the peak across the probe as expected.

Summary

The RIC RSAs are sensitive to charged-particles, and thus, can detect alpha and beta-particles. However, while alphas can be easily shielded, betas can penetrate many materials. The maximum depth they can penetrate is a function of their initial energy and the material through which it is traversing. RICs can detect betas, and this chapter investigated the use of wedge-shaped, low-Z attenuator materials to enhance their detection capabilities. The linear slope of the wedge offers the ability to exploit the MPD and determine $E_{\beta max}$. This ability would dramatically improve capabilities of the deployable hand-held survey equipment. Analysis of plate glass, Pyrex® glass, Lucite®, and natural rubber as the attenuator material was carried out using MCNPX simulations. Also, analytic calculations were used to verify simulation results.

For each attenuator material, a wedge-shaped probe was investigated to characterize their design, and thereby evaluate their energy resolution capability. The energy resolution capability was assessed using ratios of counts. Counts for an attenuator obtained from simulations using beta source energies ranging from 1.95 to 2.1-MeV were normalized to a reference value from the 2-MeV beta simulation. The ratio of counts directly reflects the strength of the signal generated in the respective RSA. The larger the

ratio, the better the $E_{\beta max}$ discrimination capability. The natural rubber wedged attenuator had the highest differences in counts (of ratio extremes) with the minimum factor of 0.24X and a maximum factor of 2.2X. The minimum factor and maximum factor were the difference in counts between the end-point energy of 1.95-MeV and 2.1-MeV source betas, respectively with regard to the 2-MeV source beta. The next highest difference in counts was for Lucite®, followed by plate glass and Pyrex® glass. For the difference of 50-keV in the end-point energy of betas ($E_{\beta max}$), when using the natural rubber attenuator, the counts in the region of interest differ roughly by a factor of two. This difference in counts can be exploited using the RSA, which is a few μm in width. Thereby, there exists a potential to perform $E_{\beta max}$ discrimination within 50-keV using natural rubber wedged attenuator.

In order to assess the performance of the natural rubber-RIC probe when the source position varies, simulations were carried out in MCNPX. When the source position was varied along the probe length, results showed an expected trend. With the source moved along the length, the attenuator thickness a beta traverses to reach the inclined surface varies. Subsequently, the region of interest corresponding to a source position moves along the probe length. The tally area in the region of interest that measures the peak count of betas also moves into the probe. This behavior in the count peak with the source movement was attributed to the property of the attenuation thickness.

Natural rubber was observed to be the optimal attenuator material for the beta probe with respect to the $E_{\beta max}$ discrimination capability and weight. The wedge-shaped natural rubber was 7-cm in length, 5-cm in width, and height linearly ranged from 0 to

1.1-cm. A compact probe can be realized using the plate glass attenuator by reducing the $E_{\beta max}$ discrimination capability by a factor of 0.92X. This probe design was 5-cm in length and width while the height linearly increased from 0 to 0.5-cm. The weight of plate glass attenuator was lower by a factor of 0.84X natural rubber as their density differed by a factor of 2.6X.

CHAPTER III

THE NEUTRON DETECTOR

Overview

Recently various efforts have been made to develop semiconductor-based neutron detectors that are similar to the RICs. Primary factors for this development are device footprint and power requirement. Moreover, the shortage of Helium-3 (^{3}He) for neutron detectors has encouraged the development of semiconductor-based detectors. Detectors of this type can be compact and less power-intensive than gaseous and scintillator-based systems. The semiconductor neutron detectors have shown a better sensitivity (intrinsic efficiency) to neutrons than the widely deployed ^{3}He detectors [33]-[38]. RICs possess the same advantages. In order to employ RICs to detect neutrons, a neutron-reactive material is required to generate charged particles which can interact with the RSAs. There exist various methodologies to employ a neutron-reactive material on the semiconductor, but one approach is to coat them with a material having a high capture cross-section for thermal neutrons. To assess the performance of neutron-reactive coatings, simulations were performed using MCNPX code. The analysis focused on determining the optimal yield of charged-particles at the RIC-coating interface.

Material and Methods

MCNPX code was used to evaluate the capability of the proposed RICs to detect neutrons. Widely employed neutron-reactive materials, namely boron, boron carbide (B_4C), and lithium fluoride (LiF) were studied using this code (Appendix B). Such

materials, following the absorption of a neutron, produce charged-particles that will interact with RSAs. Hence, the RICs can be used to detect neutrons. For example, natural boron is comprised of two isotopes, ^{10}B (Boron-10) and ^{11}B (Boron-11). Of these, the ^{10}B isotope makes up 19.9% of natural boron and has a high absorption cross-section for thermal neutrons (Figure 3.1). Following the neutron absorption, heavy charged-particles (HCPs), namely lithium (^{7}Li) and an alpha particle (α), are emitted (Equation 3.1) instantaneously (with a time scale of the order of 10^{-18}-sec). The RIC can detect these HCPs. For the neutron-reactive material analysis of boron, simulations were performed using both natural boron and enriched boron (96% ^{10}B).

$$^{10}_{5}B + ^{1}_{0}n \rightarrow ^{7}_{3}Li^{*} + ^{4}_{2}\alpha + 2.31\ MeV$$

$$^{7}_{3}Li^{*} \rightarrow ^{7}_{3}Li + 0.48\ MeV$$

$$^{10}_{5}B + ^{1}_{0}n \rightarrow ^{7}_{3}Li + ^{4}_{2}\alpha + 2.792\ MeV \qquad \text{(Eq. 3.1)}$$

In B_4C, boron content by weight is 78%. In order to increase the probability of HCP generation, the boron was enriched to 96% ^{10}B. Thus, in the simulations utilizing B_4C, the ^{10}B content was ~75%.

The ^{6}Li isotope in LiF absorbs a thermal neutron to produce two charged-particles (CPs) instantaneously, i.e., a tritium (^{3}H) and an alpha particle (Equation 3.2). To enhance CP generation, ^{6}Li content was enriched to 89% (by weight). In our simulations employing a LiF coating, the neutron-reactive ^{6}Li isotope was ~24%.

$$^{6}_{3}Li + ^{1}_{0}n \rightarrow ^{3}_{1}H + ^{4}_{2}\alpha + 4.78\ MeV \qquad \text{(Eq. 3.2)}$$

Figure 3.1 provides neutron capture cross sections using the ENDF/B-VII.0 library as a function of neutron energy (E) for ^{10}B and ^{6}Li isotopes [68]. It is discernible from this

figure that the cross sections for these two isotopes are large for thermal neutrons (0.025-eV). The cross-section varies as $E^{-0.5}$ from thermal energy up to a few hundred keV for LiF and up to a few MeV for natural boron, enriched boron, and B_4C. The cross section for ^{6}Li rises and falls rapidly above the 25-keV neutrons.

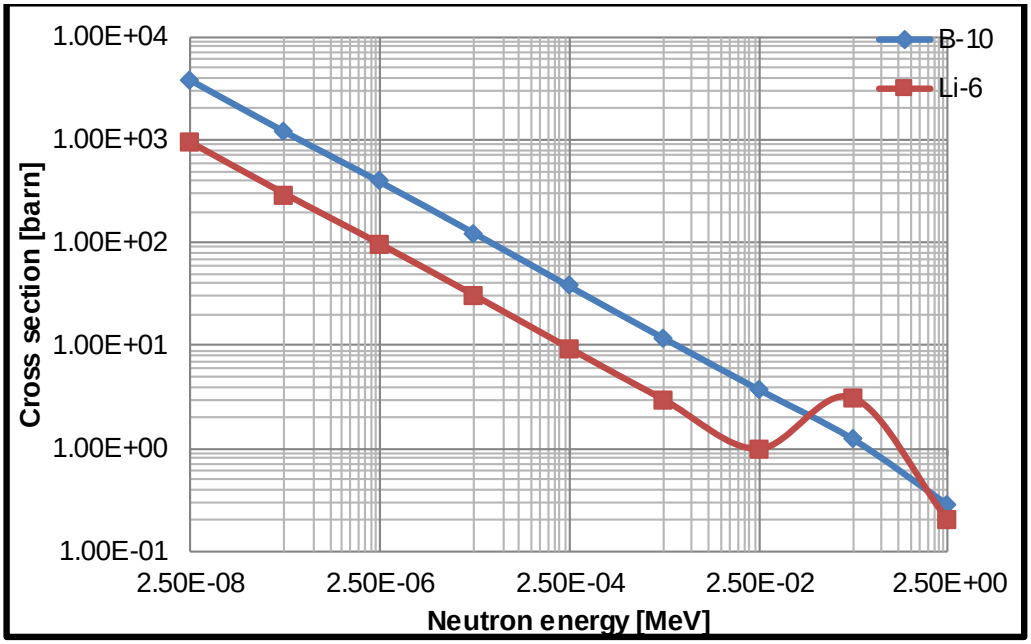

Figure 3.1. Neutrons capture cross sections for ^{10}B and ^{6}Li isotopes.

Table 3.1 lists the maximum penetration depth (MPD) of reaction products in boron, B_4C, LiF, and silicon. The MPD is the ratio of range ($g\text{-}cm^{-2}$) and density ($g\text{-}cm^{-3}$) of the material. The range is defined as the distance a particle travels before coming to rest. This range is expressed in units of $g\text{-}cm^{-2}$ so that it is independent of the density of material – approximately same for materials of similar composition [5]. For each material,

it was observed that the MPD of products was a function of the linear stopping power and the energy shared between them. The linear stopping power (S) of charged-particles in a material is the loss of energy (dE) per unit thickness (dx) (Equation 3.3). This energy loss can be represented by Bethe's classical function [5].

$$S = -\frac{dE}{dx} \qquad \text{(Eq. 3.3.)}$$

Table 3.1. The MPD of reaction products based on their energy in enriched boron, B_4C, LiF, and silicon.

Material	**Reaction product**	**MPD [µm]**			
		Enriched boron	**B_4C**	**LiF**	**Silicon**
B/B_4C	^{7}Li	1.9/1.69	1.8/1.6	-	3/2.7
	α	4.03/3.25	3.91/3.16	-	5.4/5.2
LiF	^{3}H	-	-	34.97	44.1
	α	-	-	6.35	8.6

Boron, following the absorption of a neutron, decays in two ways, but with the same heavy reaction products. Accordingly, these products share the energy, and each has a set of two MPD values in a material as listed in Table 3.1. For example, the ^{7}Li reaction product has either a 1.01 or 0.84-MeV following the neutron absorption in boron. For a 1.01-MeV ^{7}Li-particle, the MPD is 1.9-µm, 1.8-µm, and 3-µm in enriched boron, B_4C, and silicon material, respectively. However, for a 0.84-MeV ^{7}Li-particle, the

corresponding MPD values are 1.69-µm, 1.6-µm, and 2.7-µm. Similarly, the alpha-particle has a set of two MPD values for enriched boron, B_4C, and silicon material. The reaction products for LiF following the neutron absorption always share the energy in the same manner and thus, a single value of the MPD is listed.

As ^{7}Li is the heavier element, it has a lower penetrating power and MPD compared to an alpha particle. Most of the ^{7}Li-particles would be absorbed inside the neutron-reactive coating layer (~2-µm) ensuring a substantial reliance on alpha-particles to generate an electric signal. Conversely, ^{3}H is a lighter element in relation to the alpha particle and thus, has a higher penetrating power and MPD. With a large MPD in silicon, the ^{3}H-particles are less likely to interact in the RSA. The simulations in MCNPX helped to comprehend the MPD property.

MCNPX simulations focused on measuring the number of charged-particles that reach the RIC after crossing the neutron-reactive coating. For each simulation, the coating layer had a surface area of 30-mm x 50-mm and was placed on top of a 0.2-µm thick silicon layer that represented the RIC surface. All of this was enclosed in a 1-mm thick stainless steel casing to model the detector casing. The number of charged-particles crossing through the neutron-reactive layer and entering the silicon was tallied across 1-mm^2 areas (represents the collective 40,000 RSAs). Each 1-mm^2 scoring area was bounded by two lines in both the X and Y-planes. Two lines along the Y-plane were conceptualized at various positions such that there were five uniformly separated 1-mm intervals; as listed in Table 3.2. Two lines along the X-plane were theorized at various positions such that there were seven 1-mm intervals; as seen in Table 3.2. These intervals were separated

uniformly except the first one (as the probe length was 30-mm). The intervals give a total of thirty-five 1-mm^2 tally surfaces across which the yields of charged-particles were tallied. The tally regions distributed across the detector were used to assess the trend of the charged particle yield. The yield or number of charged-particles obtained from MCNPX simulations represents the count of particles entering the silicon material per incident neutron (n) [# n^{-1} mm^{-2}].

Table 3.2. The intervals over X and Y-axes.

1-mm Intervals				
X-axis [mm]			**Y-axis [mm]**	
0-1	14-15	29-30	4.5-5.5	34.5-35.5
4-5	19-20		14.5-15.5	44.5-45.5
9-10	24-25		24.5-25.5	

The numbers of charged-particles were also tallied at the farther silicon surface in 1-mm^2 tally surfaces to calculate the fractional loss (FL). The FL was defined as the ratio of the difference in count of HCPs entering and exiting the silicon material to the count of HCPs entering the silicon material (Equation 3.4).

$$FL = \frac{\#\ HCPs\ entering\ silicon\ layer - \#\ HCPs\ exiting\ silicon\ layer}{\#\ HCPs\ entering\ silicon\ layer} \quad \text{(Eq. 3.4.)}$$

Three sets of simulations were carried out for each coating. In the first set of simulations, the coating thickness was varied while using a monoenergetic neutron point source of 0.025-eV. This source was centered and placed 1-cm away from the detector face. These simulations were employed to determine the optimal thickness at which the highest number of heavy ions enter the RICs.

In the coating material, the HCPs generate electrons through ionization that interact with the RSAs to generate an electrical signal. When the boron absorbs a neutron, a gamma-ray is emitted 94% of the time. This gamma-ray may also generate electrons through ionization in the coating material. These electrons can be a source of noise in RIC neutron detection since they increase the signal generated by HCPs. The analysis was carried out to assess the signal to noise ratio while employing an optimal thickness in the detector. Measurements were made for HCPs and electrons across the center 1-mm^2 tally area. This was employed to gauge their individual contribution.

For the signal to noise analyses, two terms were defined; Count signal to noise (S/N) (Equation 3.5), and Energy S/N (Equation 3.6). The Count S/N is the ratio of the number of HCPs entering the silicon to the number of electrons entering the silicon. The Energy S/N is the amount of energy deposited within the silicon by HCPs to the energy deposited by electrons. In addition, the amount of energy deposited in the silicon by each HCP was tallied. Both terms were estimated for the center 1-mm^2 area (directly in front of the source) to compare the performance of each neutron-reactive material.

$$Count\ S/N = \frac{\#\ HCPs\ entering\ the\ silicon}{\#\ electrons\ entering\ the\ silicon} \qquad \text{(Eq. 3.5.)}$$

$$S/N\ Energy = \frac{amount\ of\ energy\ deposited\ by\ HCPs\ in\ the\ silicon}{amount\ of\ energy\ deposited\ by\ electrons\ in\ the\ silicon}$$ (Eq. 3.6.)

The second set of simulations was performed using an optimal thickness of neutron-reactive material and a monoenergetic neutron source of 0.025-eV while varying distance to the source. Following is the list of distances to the source (from detector face) that were simulated: 0.5-cm, 1-cm, 1.5-cm, 2-cm, 2.5-cm, and 3-cm. These source-to-detector simulations helped to analyze the yield of charged-particles as a function of distance to the source. The third set of simulations was performed in which the neutron energy was varied by a factor of 10 from 0.025-eV to 2.5-MeV. In these simulations, the coating layer was the optimal thickness that was determined earlier with the source placed 1-cm away from the center of the detector face. This analysis provided an understanding of the yield of charged-particles as a function of neutron source energy. These three sets of simulations were performed on each neutron-reactive material to analyze them comprehensively and conclude the optimal detector design.

Results and Discussion

Boron

Using MCNPX, an analysis was performed on both natural boron and enriched boron to compute the number of HCPs crossing the coating layer to interact in the RSAs. Table 3.3 highlights the simulation results for both natural boron and enriched boron. From each simulation (for a thickness layer), only the highest number of HCPs tallied across the thirty-five 1-mm^2 areas is presented. The values listed are counts across 1-mm^2 area per incident neutron. Relative errors associated with these results are less than 10%.

In addition, the contribution from alpha-particles and lithium ions are listed for each result in Table 3.3. The highest count for all these simulations was tallied across the same 1-mm^2 area that was directly in front of the source (X-axis interval [mm]: 14-15; Y-axis interval [mm]: 24.5-25.5). Other 1-mm^2 areas had lower yield, and the magnitude successively decreased for areas that were further away from the source. These results show the highest yield of HCPs is for the 4-μm thick natural boron and 3-μm thick enriched boron. The difference in the optimal thickness was primarily due to the difference in neutron-reactive content as natural boron had 19.9% ^{10}B and enriched boron had 96% ^{10}B. The yield was larger for enriched boron by at least a factor of 3.6X.

Table 3.3. The number of HCPs reaching the RIC as a function of coating thickness for the natural and enriched boron detectors.

Thickness [μm]	**Natural boron yield (19.9% ^{10}B)**			**Enriched boron yield (96% ^{10}B)**		
	Alpha-particle	Li-particle	Total	Alpha-particle	Li-particle	Total
1	3.64×10^{-6}	2.91×10^{-6}	6.55×10^{-6}	1.68×10^{-5}	1.31×10^{-5}	2.98×10^{-5}
2	6.25×10^{-6}	3.48×10^{-6}	9.73×10^{-6}	2.75×10^{-5}	1.36×10^{-5}	4.11×10^{-5}
2.5	7.04×10^{-6}	3.31×10^{-6}	1.03×10^{-5}	3.02×10^{-5}	1.32×10^{-5}	4.33×10^{-5}
3	7.69×10^{-6}	3.31×10^{-6}	1.1×10^{-5}	3.28×10^{-5}	1.32×10^{-5}	4.6×10^{-5}
4	8.35×10^{-6}	3.26×10^{-6}	1.16×10^{-5}	3.19×10^{-5}	1.22×10^{-5}	4.4×10^{-5}
5	8.13×10^{-6}	3.28×10^{-6}	1.14×10^{-5}	3.01×10^{-5}	1.15×10^{-5}	4.15×10^{-5}

For both natural boron and enriched boron detectors, it was observed that the contribution of alpha-particles was greater than lithium-particles by at least a factor of 1.25X. This factor was 2.57X for the case of 4-µm thick natural boron and 2.47X for the case of 3-µm thick enriched boron. The alpha-particles have a larger MPD than lithium ions in boron (Table 3.1 for the MPD). As a result, the number of alpha-particles entering the silicon is large compared to lithium ions. Subsequently, the amount of energy deposited in the silicon was greater for alpha-particles by a factor of 2.09X and 2.10X for the 4-µm thick natural boron and 3-µm thick enriched boron, respectively. The thermal neutron sensitivity of the RIC was estimated to be 17-cps-nv^{-1} (counted pulses per neutron flux density), which is larger by a factor of 2.23X than the sensitivity of the ^{3}He tube of 2" diameter and 1" length (gas fill pressure of 2 atm) [69]. The sensitivity is the number of charge carriers generated in the sensitive region of the detector to the neutron flux density observed in the neutron-reactive material. The Count S/N (Equation 3.5) and Energy S/N (Equation 3.6) were estimated as 21.9X and 3633X, respectively. This Energy S/N implies HCPs, namely alpha and ^{7}Li-particles, deposit energy up to three orders of magnitude times higher than electrons in silicon. Overall, the major contributor to the electric signal generation in the RICs is alpha-particle.

The individual contribution analysis among HCPs also enabled assessment of the alpha probe. In the alpha/beta survey probe, alphas are detected directly. In this chapter, analysis of the generated alphas, including their interaction in the silicon (RIC), facilitated investigation of their detection. Alpha-particles were observed to deposit about 33% of their energy in silicon. These interactions enable alpha detection using the RIC.

Based on the yield of charged particles, the next two sets of simulations were performed with enriched boron as the coating material. The dimensions of this coating were 30-mm x 50-mm x 3-µm. In addition, 0.2-µm thick silicon was placed in contact with it. All of this was enclosed in 1-mm thick stainless steel to represent the detector configuration. In Figure 3.2, the highest count of HCPs is displayed as a function of source-to-detector distance. The results are per incident neutron and have relative errors less than 3.4%. The 1-σ error bars are not visible on the plot because data points are obscuring them. The results show a decrease in the yield of HCPs with an increase in the distance between the detector face and the source. The reduction in the yield was observed to follow an inverse-square law as expected. The best-fit trend-line has an exponent value of -1.795 and an R^2 value of 0.9992.

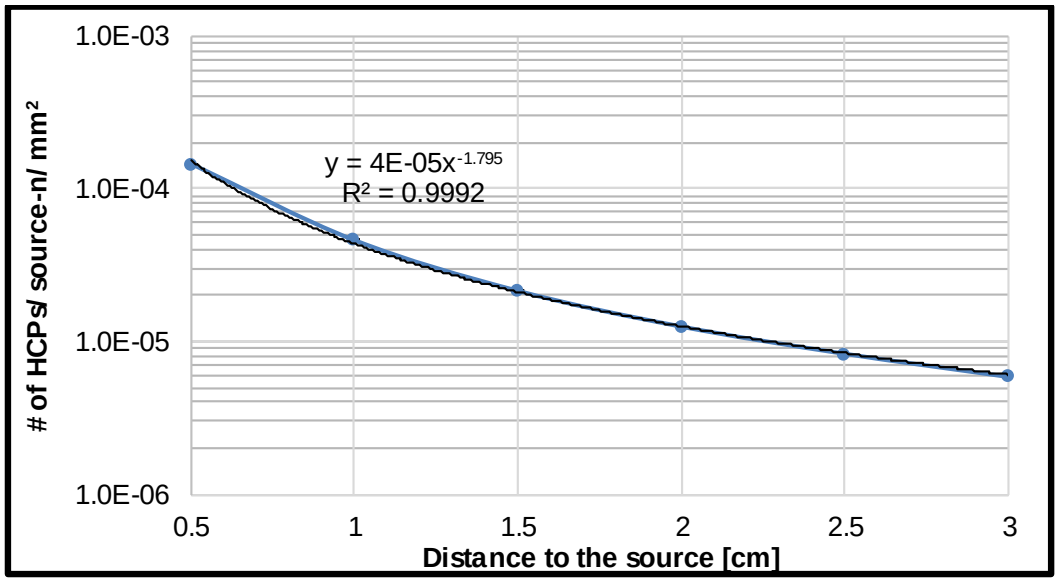

Figure 3.2. The number of HCPs tallied across the center tally as a function of distance to the source for the enriched boron detector.

A decrease in the yield (across 1-mm^2 area) was observed for tally areas, which were further away from the source. Figures 3.3, 3.4, 3.5, and 3.6 are heat maps comprised of thirty-five 1-mm^2 areas for the source-to-detector distance as 0.5, 1, 2, and 3-cm, respectively. When the source was 0.5-cm away from the center of the detector face, the tallies further away from the source registered fewer neutron interactions. This reduction in interactions can be observed by the difference in yields between the center tally and the rest of the thirty-four tallies (Figure 3.3). The difference in yields is attributed to the property called a solid angle that is a two-dimensional angle subtended by a cone in three-dimensional space. The solid angle presented by a tally surface decreases as a function of the distance to the source nadir, resulting in lower yields. Conversely, when the source was placed further away, the solid angle presented by a tally surface to the source increases. As a result, tallies had relatively minor differences in the count, which can be observed in Figures 3.4, 3.5, and 3.6.

1.74E-09	2.80E-11	1.36E-13	6.24E-09	1.57E-09	2.95E-09	8.21E-14
5.67E-09	2.08E-08	6.51E-08	1.19E-07	4.73E-08	7.90E-09	2.22E-08
1.53E-08	8.17E-08	8.05E-07	1.45E-04	1.06E-06	1.26E-07	1.37E-08
1.87E-13	2.50E-08	6.44E-08	7.34E-08	3.36E-08	2.40E-09	5.29E-09
1.36E-13	3.20E-11	5.11E-09	6.28E-09	1.78E-13	1.63E-09	2.88E-09

Figure 3.3. The number of HCPs tallied across thirty-five 1-mm^2 areas when the source was 0.5-cm away from the enriched boron detector face.

4.94E-11	5.12E-14	2.82E-09	1.52E-09	4.25E-09	2.65E-09	3.73E-16
6.89E-10	7.32E-09	4.34E-08	9.14E-08	1.3E-07	2.03E-08	8.45E-09
2.08E-08	1.28E-07	2.9E-05	4.6E-05	2.87E-05	1.09E-07	1.22E-08
9.78E-09	3.32E-08	6.55E-08	1.22E-07	6.24E-08	2.59E-08	1.49E-08
6.78E-11	7.24E-14	1.12E-08	4.18E-11	6.26E-09	9.58E-10	2.19E-12

Figure 3.4. The number of HCPs tallied across thirty-five 1-mm^2 areas when the source was 1-cm away from the enriched boron detector face.

1.89E-09	1.19E-09	1.05E-08	2.14E-13	2.25E-08	7.63E-09	5.21E-09
6.94E-09	2.71E-08	2.68E-07	6.56E-06	3.74E-07	1.01E-07	5.49E-09
4.43E-08	6.25E-06	1.15E-05	1.26E-05	1.18E-05	6.73E-06	3.96E-08
1.87E-08	7.14E-08	2.43E-07	6.56E-06	3.70E-07	6.76E-08	2.09E-08
1.57E-08	6.13E-09	2.18E-09	1.18E-08	7.46E-10	2.61E-13	1.50E-09

Figure 3.5. The number of HCPs tallied across thirty-five 1-mm^2 areas when the source was 2-cm away from the enriched boron detector face.

2.01E-10	2.02E-09	1.66E-08	1.65E-08	2.05E-08	1.89E-08	6.07E-09
7.40E-08	4.45E-06	4.88E-06	5.15E-06	5.05E-06	4.47E-06	7.95E-08
2.24E-06	5.10E-06	5.63E-06	5.90E-06	5.46E-06	5.07E-06	4.42E-06
7.20E-08	4.42E-06	4.83E-06	5.18E-06	4.78E-06	4.36E-06	5.12E-08
6.24E-09	6.88E-10	7.22E-09	3.11E-09	4.33E-08	7.15E-09	8.31E-12

Figure 3.6. The number of HCPs tallied across thirty-five 1-mm^2 areas when the source was 3-cm away from the enriched boron detector face.

The third set of simulations was performed with the monoenergetic neutron source placed 1-cm away while varying its energy. Figure 3.7 illustrates the highest yield of HCPs obtained across the center tally area as a function of the neutron energy. In addition, the corresponding uncertainty (1-σ) for each result is shown in the plot. The yield across each 1-mm^2 tally is per incident neutron. All of these results have relative errors less than 3%. It was observed the best-fit trend-line was following a power function with an exponent value of -0.457 and had an R^2 value of 0.9899. The power function is a distinctive feature which can be expected as the ^{10}B neutron capture cross section that is proportional to the neutron energy (E) as $E^{-0.5}$.

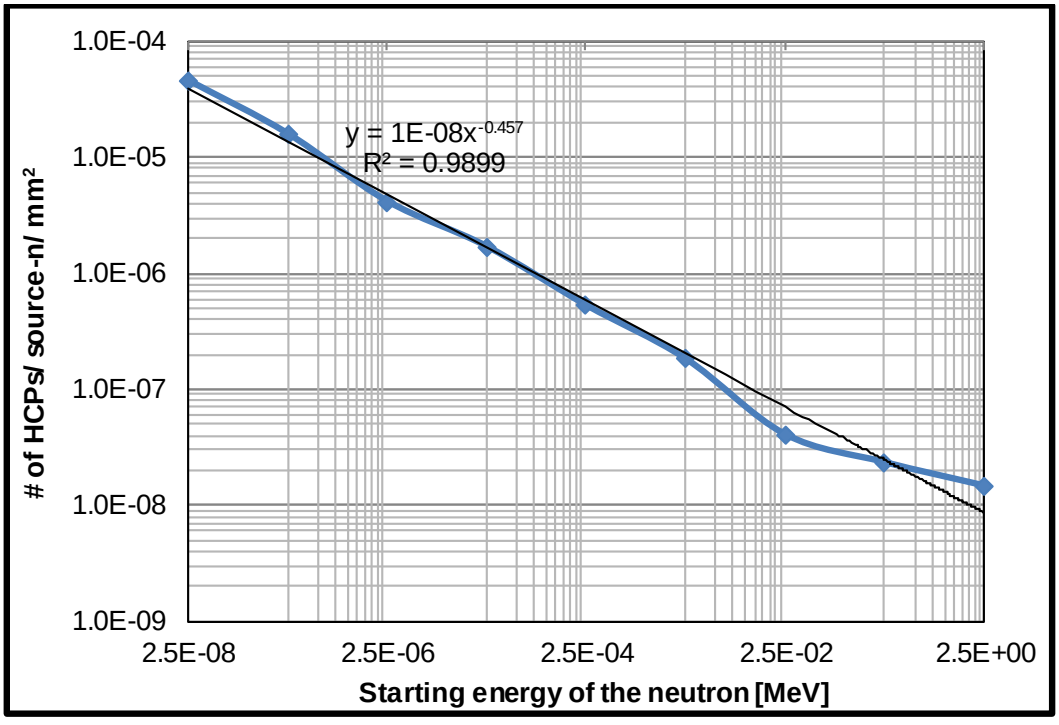

Figure 3.7. The number of HCPs tallied across at the center tally as a function of starting energy of the neutron for the enriched boron detector.

It can be observed that the yield for the 25-keV and 2.5-MeV neutron does not fit on the trend line and thus, the difference in the expected and observed trend (Figure 3.7). The difference for the 25-keV case is explained by the presence of iron, a primary constituent of the stainless steel casing, which has an affinity for 25-keV neutrons. Subsequently, this cross section property of the iron lowers the yield of HCPs for the 25-keV neutrons. The neutron capture cross section for ^{10}B has a resonance for the 2.5-MeV neutron. This resonance has caused the slight increase in the yield for the 2.5-MeV neutron; as seen in Figure 3.7.

Boron Carbide

MCNPX simulations were performed to determine the optimal thickness for boron carbide that yields the maximum number of HCPs for interaction in the RIC. Table 3.4 provides results for the highest measurement in a set of thirty-five 1-mm^2 areas as a function of thickness. The reported results are per incident neutron and have relative errors less than 5%. Similar to enriched boron, boron carbide has an optimal thickness at 3-μm. However, the yield for boron carbide was lower by 0.8X compared to enriched boron due to the lower number of ^{10}B atoms in B_4C.

The 3-μm thick B_4C coating shows a significant contribution from alpha particles, which was larger than lithium ions by 2.51X. It was also observed that the energy deposited in the silicon was 2.09X larger for alpha particles. The Count S/N (Equation 3.5) and Energy S/N (Equation 3.6) were estimated as 18.8X and 3372X, respectively. These indicate the number of HCPs (alpha and ^7Li-particles) entering the silicon is 18.8X the number of electrons, and thereby, the amount of energy deposited by alpha and ^7Li-

particles was three order of magnitude larger than electrons. Overall, in the boron carbide, the primary dependence is on alpha-particles to generate an electrical signal in the RICs.

Table 3.4. The number of HCPs reaching the RIC as a function of coating thickness for the B_4C detector.

Thickness [μm]	**Boron carbide yield (~75% ^{10}B)**		
	Alpha-particle	Li-particle	Total
1	1.38×10^{-5}	1.05×10^{-5}	2.43×10^{-5}
2	2.24×10^{-5}	1.09×10^{-5}	3.33×10^{-5}
2.5	2.47×10^{-5}	1.08×10^{-5}	3.55×10^{-5}
3	2.64×10^{-5}	1.05×10^{-5}	3.69×10^{-5}
4	2.61×10^{-5}	1.03×10^{-5}	3.63×10^{-5}
5	2.44×10^{-5}	9.21×10^{-6}	3.36×10^{-5}

Based on the yield of HCPs, the next two sets of simulations were carried out with the B_4C coating dimension as 30-mm x 50-mm x 3-μm. In the second set of simulations, the position of the monoenergetic 0.025-eV neutron source was varied. Figure 3.8 demonstrates the highest yield of HCPs, tallied across the center tally region, is a function of distance to the source. This feature closely follows an inverse-square law as the best-fit trend-line has an exponent value of -1.797 and an R^2 value of 0.9994. These results across

1-mm^2 areas are per incident neutron and have a corresponding relative error less than 1.5%.

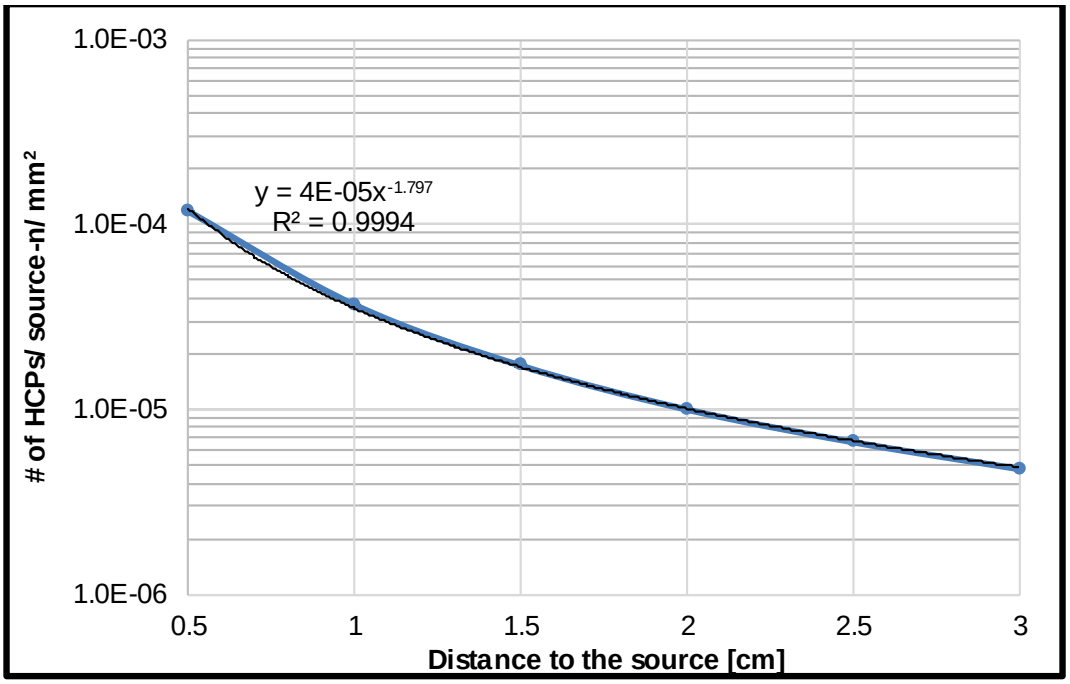

Figure 3.8. The number of HCPs tallied across the center tally as a function of distance to the source for the B_4C detector.

The heat map presented in Figure 3.9 displays the difference in yields across all thirty-five tally areas when the source was 1-cm away from the center of the B_4C detector face. The presented results are per incident neutron and function of the solid angle. It was observed the differences in counts across all tallies decreased when the source was moved away from the detector face. The decrease in difference occurs because tally areas that are further away from the center present larger solid angles to the source when the source,

centered on the detector face, was placed further away. This analysis was similar to the enriched boron analysis of distance to the source (Figure 3.4).

1.27E-09	6.29E-14	8.28E-09	8.07E-09	1.31E-10	6.81E-14	4.66E-14
4.75E-09	2.52E-08	6.74E-08	8.83E-08	9.44E-08	5.70E-08	1.85E-08
2.00E-08	1.10E-07	2.34E-05	3.69E-05	2.37E-05	9.13E-08	1.75E-08
3.87E-09	1.77E-08	5.70E-08	9.04E-08	4.76E-08	1.66E-08	7.53E-09
5.70E-10	1.68E-09	2.14E-09	4.80E-09	7.80E-09	3.86E-09	1.04E-14

Figure 3.9. The number of HCPs tallied across thirty-five 1-mm^2 areas when the source was 1-cm away from the B_4C detector face.

However, boron carbide had lower yields than enriched boron as the number of ^{10}B atoms was lower. The rest of the heat maps for the boron carbide detector were similar to Figures 3.3, 3.5, and 3.6, and thus, are not presented. However, Figure 3.9 offer an understanding of the trend in results. In the heat maps, the yields of HCPs for boron carbide were lower compared to the case of enriched boron as boron carbide had a lower concentration of neutron-reactive element.

The third set of simulations presents the highest yield of HCPs tallied across the center tally per starting energy of the neutron for nine energies (Figure 3.10). The relative errors associated with these results are less than 3.8%. The 1-σ error bars are not visible on the plot because the data point is obscuring them. In these simulations, the source was

placed 1-cm away from the detector face, which encloses a 3-µm thick B_4C coating on top of a 0.2-µm thick piece of silicon.

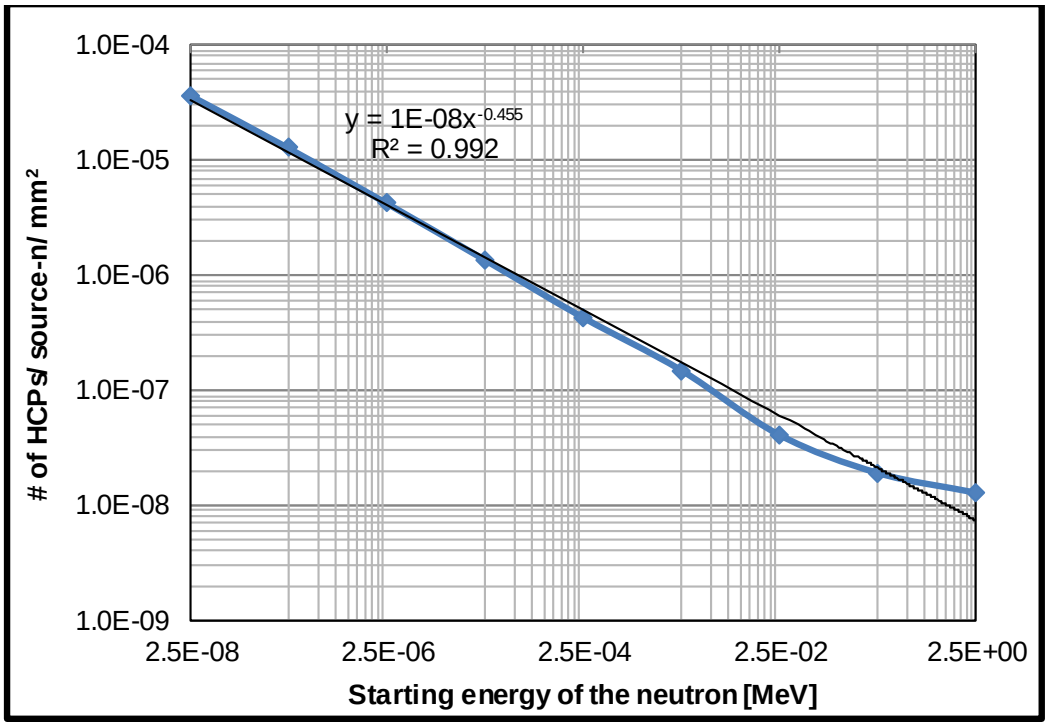

Figure 3.10. The number of HCPs tallied across the center tally as a function of starting energy of the neutron for the B_4C detector.

A power function was found to be the best-fit having an exponent value of -0.455 and R^2 value as 0.992 (Figure 3.10). This function was expected because ^{10}B's neutron capture cross section is proportional to neutron energy as $E^{-0.5}$ over a wide energy range (Figure 3.1). Again, similar to the boron analysis, it can be observed that the yield for the 25-keV and 2.5-MeV incident neutron does not fit on the trend line. For the 25-keV

neutrons, iron, a primary constituent of the stainless-steel casing has a higher cross section than the boron. The neutron capture cross section for ^{10}B has a resonance for the 2.5-MeV neutron resulting in a slightly higher yield. Overall, differences in cross section have resulted in the distinction between the expected and observed trend (Figure 3.10).

Lithium Fluoride

The first set of simulations with a lithium fluoride coating shows a lower yield of CPs compared to both the enriched boron and boron carbide analysis (Table 3.5). This observation is attributed to the concentration of the neutron-reactive element which is lower in lithium fluoride. The neutron-reactive content by weight was 96% ^{10}B in enriched boron, ~75% ^{10}B in boron carbide, and ~24% ^{6}Li in lithium fluoride.

Table 3.5. The number of CPs tallied across the center tally as a function of coating thickness for the LiF detector.

Thickness [μm]	**Lithium fluoride yield (~24% ^{6}Li)**		
	Tritium-particle	Alpha-particle	Total
10	1.71×10^{-5}	4.62×10^{-7}	1.76×10^{-5}
20	2.56×10^{-5}	4.17×10^{-7}	2.6×10^{-5}
25	2.70×10^{-5}	4.08×10^{-7}	2.74×10^{-5}
30	2.66×10^{-5}	3.75×10^{-7}	2.7×10^{-5}
40	2.49×10^{-5}	3.48×10^{-7}	2.53×10^{-5}
50	2.31×10^{-5}	3.08×10^{-7}	2.34×10^{-5}

Each result in Table 3.5 is the yield of CPs measured across 1-mm^2 area per incident neutron and have a relative error less than 7%. The highest yield of CPs was realized for 25-µm thick lithium fluoride. It was noticed that ^{3}H-particles contribute largely to the tally by a minimum factor of 37X alpha particles. Also, the amount of energy deposited by ^{3}H-particles was greater by a factor of 17X for the 25-µm thick LiF. This difference in contribution is explained by the MPD of alpha-particles, which was lower than ^{3}H-particles in the LiF (Table 3.1).

The fractional losses (Equation 3.4) were estimated for both ^{3}H and alpha-particles. Since the MPD of ^{3}H-particles is large compared to alpha-particles in silicon (Table 3.1), there is a lower probability of interaction within 0.2-µm thick silicon. The lower probability means the number of charged-particles crossing through the silicon without depositing their energy will be larger for ^{3}H particles. It was observed that ^{3}H and alpha-particles had FL of 3.5% and 21.6%, respectively, for the 25-µm thick LiF. As the number of ^{3}H-particles entering silicon is relatively higher (Table 3.5), the energy deposited by ^{3}H-particles was higher compared to the energy deposited by alpha particles. However, complete energy deposition within the 0.2-µm thick silicon occurs more for alpha-particles by a factor of 6.17X than ^{3}H particles. These events of complete energy deposition within the silicon, representing the RIC sensitive region, mainly serve to detect the presence of neutrons. The Count S/N was estimated using Equation 3.5 as 18.3X while the Energy S/N was estimated using Equation 3.6 as 1354X. While this ratio indicates higher energy deposition occurs by CPs in LiF simulations, this S/N ratio was still lower than the enriched boron case (by 0.37X) and the B_4C case (by 0.40X). However, these

results imply that for all three coating materials alpha-particles are the dominant source of electric signal generation in the RICs.

For the second set of lithium simulations, where the distance to the source was varied over an interval of 0.5-cm from 0.5 to 3-cm, the CP yield across the center 1-mm^2 area was observed per incident neutron. In these simulations, a 25-μm thick LiF coating layer was employed. The results are for 0.025-eV incident neutrons with relative errors less than 6%. The best-fit trend-line had an exponent value of -1.795 and an R^2 value of 0.9993 indicating the yield approximately follows an inverse-square law. The highest yield of CPs in this analysis is lower than the highest yield of HCPs in enriched boron (by 1.65X) and B_4C (by 1.34X) (Figure 3.11). The heat maps were similar in nature with significant contributions from ^3H-particles but had lower yields of CPs compared to HCPs in both the enriched boron and B_4C systems. An understanding of the trend in results can be made from Figure 3.11 and thus, heat maps for LiF are not presented. The lower yields were due to the lower thermal neutron capture cross section for ^6Li and minor content of the neutron-reactive element in the coating layer, which was mere ~24% ^6Li.

In the third set of LiF simulations, the 25-μm thick LiF was modeled to assess the yield as a function of incident neutron energy. Figure 3.12 presents the highest yield across thirty-five tally areas relative to the monoenergetic neutron source. The plot is on a log-log scale to show the trend in yield. The results have relative errors less than 6%. It is evident the highest yield was at 0.025-eV but was lower than the highest yield for enriched boron at 0.025-eV by a factor 0.6X. This difference is a property of the cross-section and neutron-reactive content.

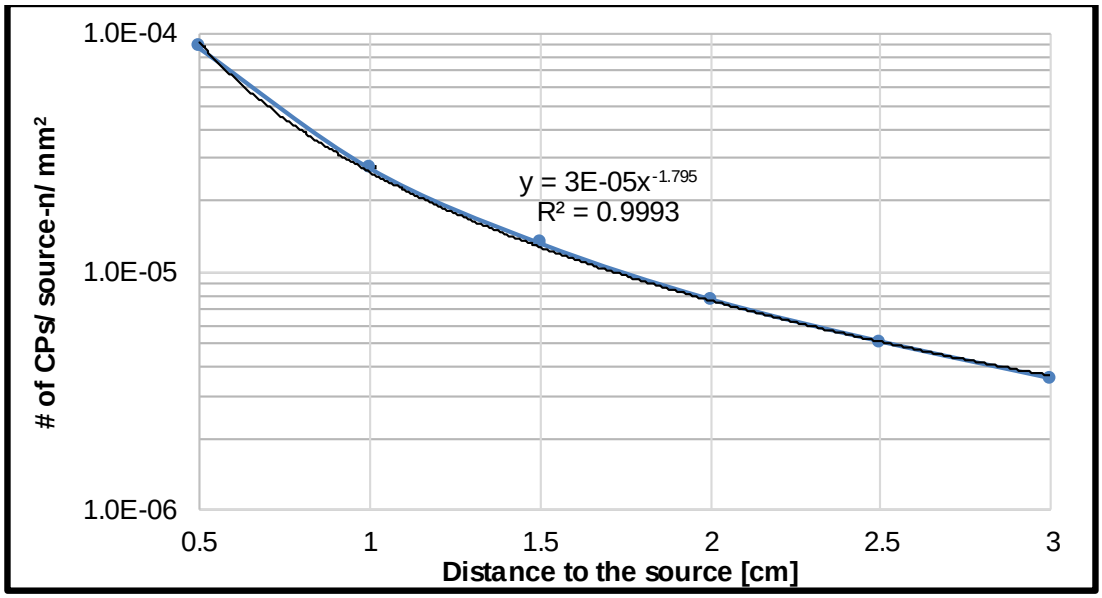

Figure 3.11. The number of CPs tallied across the center tally as a function of distance to the source for the LiF detector.

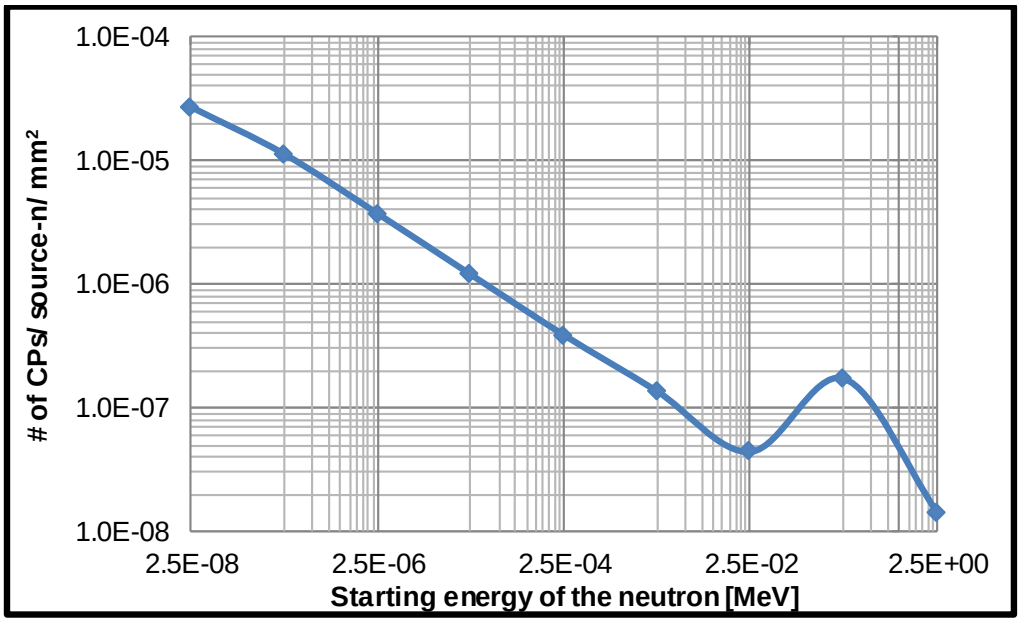

Figure 3.12. The number of CPs tallied across the center tally as a function of starting energy of the neutron for the LiF detector.

The trend in yield as a function of incident neutron energy was as expected (Figure 3.1). From thermal energy to a few hundred eV, the yield was proportional to neutron energy as $E^{-0.5}$. Above this energy, the ^{6}Li cross section first rises and then falls. This phenomenon widely known as resonance was also observed in the yield of CPs. For the 250-keV neutron, the yield of CPs was larger than the 25-keV neutron and 2.5-MeV neutron. In other words, for the 250-keV neutron, a resonance was observed in the yield of charged particles, which is similar to the resonance in cross section of ^{6}Li (Figures 3.1 and 3.12).

Summary

The natural and enriched boron, boron carbide, and lithium fluoride are widely popular neutron-reactive material for the semiconductor-based neutron detector. These materials were investigated using MCNPX code to gain design information for a RIC-neutron detector. The neutron-reactive content in these material was 19.9% ^{10}B, 96% ^{10}B, ~75% ^{10}B, and ~24% ^{6}Li, respectively. It was observed that the highest yield of charged-particles was tallied for the enriched boron, followed by boron carbide, lithium fluoride, and natural boron. In other words, the content of the neutron-reactive material regulates the yield of charged particles.

The enriched boron and boron carbide have an optimal thickness of 3-μm at which these materials yield the highest number of charged particles. The lithium fluoride and natural boron have an optimal thickness of 25-μm and 4-μm, respectively. Each of these coatings had a surface area of 30-mm x 50-mm. For each coating material, the yield of charged-particles was tallied highest across the same center tally area that was directly in

front of the source. Other 1-mm^2 areas had lower yields, and the yield magnitudes decreased for areas that were further away from the source, which is centered on the detect face.

Consistently, a decrease in the yield of alpha-particles was observed across any of the thirty-five tally areas when the distance to the source was increased. The yield of charged-particles was observed to follow an inverse-square law as a function of distance to the source. In addition, a decrease in the yield of charged-particles was observed when the starting energy of the neutron was increased. The yield was observed to be proportional to neutron energy as $E^{-0.5}$ from thermal energy up to a few hundred keV for LiF and up to a few MeV for enriched boron and B_4C. The trends in the yield of charged-particles are similar to the trends of their corresponding cross sections (a function of the neutron energy).

Another important observation was the individual contribution made by the different charged-particles to the total yield. The ^{7}Li-particles being heavy were not able to contribute much to the yield as compared to alpha-particles in enriched boron and boron carbide systems. Although ^{3}H-particles contributed significantly to the yield in LiF simulations, the numbers of events where they completely deposited their energy were lower by a factor of 0.16X than the alpha-particles. The event of complete energy deposition within the silicon (RIC sensitive region) facilitates neutron detection. With respect to this aspect, in all coating materials, as alpha-particles contribute most significantly, they dominate signal generation in the RICs.

The Energy S/N for the thermal neutron source was 3633X for the enriched boron system. This ratio for enriched boron was higher than B_4C and LiF by 1.08X and 2.7X, respectively. This S/N analysis helped to evaluate the RICs capability to detect neutrons and propose the neutron detector design. The RIC-neutron detector will use enriched boron (96% ^{10}B) as the neutron-reactive material having a coating thickness of 3-μm. The thermal neutron sensitivity of the RIC was estimated to be 17-cps-nv^{-1} (counted pulses per neutron flux density), which can be increased by stacking the RICs. This sensitivity is comparable to the sensitivity of the ^3He tube, and so the RIC neutron detector, when fabricated, can be an alternative to the ^3He detectors.

CHAPTER IV

THE PMT REPLACEMENT

Overview

Radiation detectors are primarily used to detect four types of radiation: alpha particles, beta particles, gamma rays, and neutrons. There is significant emphasis on detecting gamma rays, which are emitted by the majority of nuclear materials attractive to those with malicious intent. For the RICs to detect gamma rays, a medium is required to generate secondary charged-particles that can interact with the sensitive regions of the chip. The scintillation-based detector, in which a scintillator crystal and a photocathode generate secondary charged particles, is currently used to detect gamma rays. When a gamma-ray interacts in the scintillation crystal, optical photons are created and are either absorbed in the crystal or reach the photocathode. Photons reaching the photocathode produce photoelectrons that are accelerated and multiplied in the photomultiplier tube (PMT). Generally, the PMT and photocathode are placed only on one surface of the scintillation crystal to collect optical photons [4]. The remaining crystal surfaces have a reflective coating in the hope that scintillation photons striking these areas will be reflected and eventually reach the PMT. In this traditional approach, only a fraction of the total optical photons are converted to photoelectrons, resulting in a loss of optical photons (signal).

A conventional gamma-ray detection system will use a sodium iodide (NaI) crystal, doped with thallium as a scintillator to detect, identify, and monitor radiation

levels. NaI crystals can be manufactured in large sizes to attain higher detection efficiency. Additionally, these detectors are relatively inexpensive, which makes them widely popular. Such a detection system is commonly deployed for nuclear security and radiological emergency response. The main disadvantages associated with NaI-detectors are their relatively low-energy resolution capability and high-power requirement. The poor energy resolution is typical as the dynodes in a PMT amplify not only the primary photoelectrons but also the noise [4]. In this dissertation, the RIC-gamma ray detection system employs a NaI crystal for scintillation.

Using the GEANT4 code, analyses were carried out on two NaI detector configurations to observe the change in light (optical photon) collection efficiency when photocathodes are used with the RICs to collect optical photons. This software package has an advantage over other Monte Carlo transport codes as it offers the ability to not only model but also visualize scintillation light transport. In the proposed approach, the photocathode and RICs were placed on all crystal surfaces to collect optical photons, as opposed to the PMT approach in which a photocathode is placed on a single surface of the crystal [61]. The code was used to model two NaI crystals and observe gamma-ray interactions in the NaI crystal. Results were used to analyze optical photon generation and their collection across each crystal surface.

In the proposed approach, where the light collection efficiency can be increased, the amplifying dynodes may not be needed. In other words, the bulky and power-intensive PMT can be replaced with a unique system of photocathode(s) and RICs. In order to evaluate this capability of RICs, a part of our research tallied the amount of scintillation

light produced and the intensity with which it interacts with different parts of the photocathode. At the photocathode, based on the quantum efficiency, some optical photons get converted to photoelectrons. For higher energies, the concentration of photoelectrons may stress the ability of the RHA to perform radiation detection. The electronics of the RHA are, as such, not the topic of this dissertation.

Material and Methods

Simulations of gamma-ray interactions inside a NaI crystal were used to generate and track optical photons traveling through the crystal to the photocathodes to produce photoelectrons. The RICs placed adjacent to the photocathodes collect the photoelectrons and generate a detection signal. In these simulations, photocathodes and RICs were placed on each crystal surface to collect optical photons. In the traditional approach, only one crystal surface uses a photocathode to collect optical photons while other crystal surfaces act as reflectors to direct the optical photons to the crystal face where the PMT is attached. These two approaches were analyzed and compared to observe the amount of scintillation light produced and to determine the number of optical photons exiting the crystal, i.e., the number of optical photons that can potentially interact with the photocathode(s) [61].

GEANT4 was used to simulate the NaI crystal (Figure 4.1) to observe the generation of scintillation light following the interaction of gamma rays (Appendix C). In each simulation, a 0.1016-cm thick stainless-steel casing surrounded the crystal, and a reflection coefficient (for optical photons) of 0.95 was used. Additionally, the crystal was placed in the center of a room filled with air. Two sets of simulations were performed. The first employed a 5.08-cm x 10.16-cm x 40.64-cm (2” x 4” x 16”) NaI crystal,

henceforth called the "large crystal". The second used a 5.08-cm x 5.08-cm x 5.08-cm (2" x 2" x 2") NaI, termed as the "small crystal".

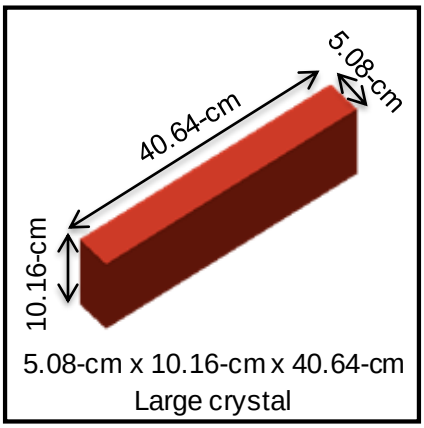

Figure 4.1. Visualization of the GEANT4 simulated NaI.

For the large crystal, an initial analysis was performed with a photocathode located only on one of the 5.08-cm x 10.16-cm surfaces (traditional approach (TA)). In the second set of simulations, all crystal surfaces had photocathodes (proposed approach (PA)). Similar RIC/PMT placement simulations were conducted for the small crystal. For each scenario, 25 sets of simulations were performed and the sample mean, representing an average result along with the standard error of the mean, was determined. The relative error (RE) associated with the result reduces when the number of particle histories (N) increases as $RE \propto \sqrt[-1]{N}$. Using this relation, the number of gamma-rays required for each set of simulations was determined to maintain the relative standard error below 10%.

For simulations involving the large crystal, the gamma-ray source was considered to be located at two separate positions. Position I and Position II were 10-cm from the end of the crystal along the center-line of the 10.16-cm x 40.64-cm and 5.08-cm x 40.64-cm faces, respectively (Figure 4.2) while 10-cm away from the face of the crystal surface. For simulations with the small crystal, the gamma-ray source was placed 10-cm away from the scintillator while centered on a crystal face. In each set of simulations, different monoenergetic gamma-ray sources were used with energies of, 60-keV (^{241}Am), 186-keV (^{235}U), 414-keV (^{239}Pu), 662-keV (^{137}Cs), 1000-keV (^{234m}Pa), and 1332-keV (^{60}Co).

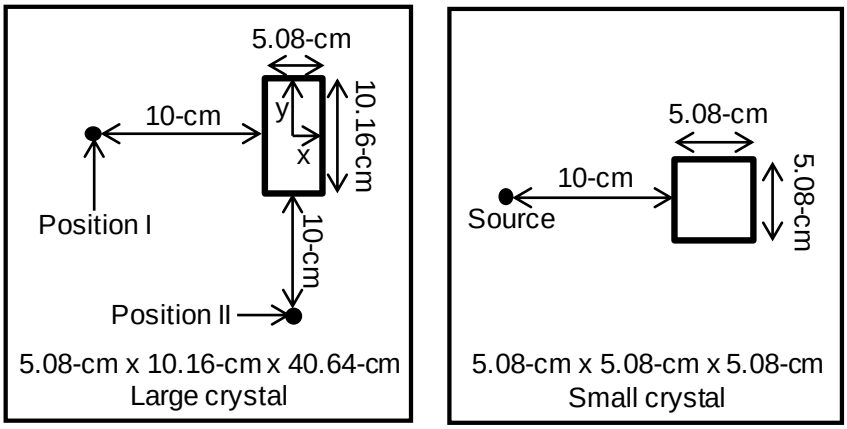

Figure 4.2. Visualization of source positions.

In simulations, the generation of optical photons was considered only after the gamma ray was completely absorbed in the crystal, widely known as the photoelectric (PE) event. In each simulation, the number of PE events, the number of generated optical photons, and the number of collected optical photons were tallied. The NaI crystal doped

with thallium intrinsically generates 40,000 optical photons per MeV deposited. The analysis was carried out to observe the linear relationship between the gamma-ray energy deposited and the number of optical photons generated.

A term called the *light collection efficiency* was defined for the collection analysis (Equation 4.1). This term is the ratio of the total number of optical photons collected to the total number of optical photons generated. Another term called *normalized PE* was defined as the ratio of photoelectric events for a particular energy to the total number of photoelectric events over all six-energies (Equation 4.2). These fractions were determined for each case where the gamma-ray energy was varied.

$$Light\ collection\ efficiency = \frac{\#\ Optical\ photons\ collected}{\#\ Optical\ photons\ generated} \qquad \text{(Eq. 4.1)}$$

$$Normalized\ PE = \frac{\#\ PE\ for\ a\ gamma{-}ray\ energy}{\#\ Total\ PE} \qquad \text{(Eq. 4.2)}$$

A term called the *collection ratio* was defined as the ratio of the total number of optical photons collected through the proposed approach (PA) compared to the traditional approach (TA) (Equation 4.3). The number of photoelectrons generated from the number of optical photons absorbed at the photocathode is given by a term called the quantum efficiency. In general, this value is 25-30% [4]. The collection ratio represents the ratio of photoelectrons generated through the PA compared to the TA. Under these simulation conditions, the light collection efficiency, normalized PE, and collection ratio were observed. The observations allowed a comparison between the performances of the two different sized crystals.

$$Collection\ ratio = \frac{\#\ Optical\ photons\ collected\ via\ PA}{\#\ Optical\ photons\ colleced\ via\ TA} \qquad \text{(Eq. 4.3)}$$

To assess the signal generation capability of photoelectrons, the analysis was carried out on the distribution of optical photons. This concentration analysis was necessary as it helps to determine the optimal size of the RSA in which a suitable signal can be generated to detect gamma rays. Following each PE event, even if initially Compton-scattered, the concentration of collected optical photons on the internal surface facing the source (gamma-ray incident surface) was noted. The concentration analysis on the incident surface was performed for each collected optical photon. Each optical photon was assumed to be at the center of a circle while the rest of optical photons collected on that surface would be in its surrounding. The circle was conceptualized to measure the concentration of optical photons within the circular area representing the RSA. This concentration is proportional to the signal generated in the RSA. To determine the optimal size of the RSA, the radius of the circle was varied. The number of optical photons concentrated on an RSA was grouped as a function of radius (of the circle) representing variation in the RSA size. The radius size ranged from 1-μm to 10-μm, with an increment of 1-μm, to determine the optimal RIC RSA size.

The maximum concentration of optical photons for each radius in a PE event was utilized for the further study. From each set of simulations (a total of 25 sets), the maximum concentration for each radius size was used to calculate the sample mean and the sample standard deviation. The collection of dispersed optical photons on a surface can be difficult to detect. Accordingly, the maxima of optical photons for radii sizes were observed which can generate a suitable signal in the RIC. The sample mean thus represents the average of maximum concentrations of optical photons on an RSA, obtained from each

set of simulations. The averaged maximum concentrations were evaluated for both crystal sizes while using the 60-keV, 662-keV, and 1332-keV gamma rays. The number of optical photons concentrated within a 5-μm X 5-μm RSA would have to exceed 1000 to generate a suitable signal in an RSA. This threshold was determined using a set of electrical simulations. The distribution analysis of optical photons on the incident surface for both crystal sizes was carried out to optimize the RIC RSA size.

Results and Discussion

Large Crystal

In the simulation, the depositions were scored when a gamma interaction concluded with a PE event. For instance, if a photon Compton-scattered out of a crystal, it was tallied. However, if the photon scattered and then was absorbed in a PE, then the entire energy deposition and scintillation photon production were recorded. For each simulation, the number of photoelectric (PE) events, the total number of optical photons generated, and the total number of optical photons exiting the crystal boundary (i.e., interacting with the photocathodes) were scored. The number of PE events multiplied by the gamma-ray energy for each scored event equals the total energy deposited. To verify GEANT4 generated the correct number of optical photons the estimated number of produced photons was plotted against the scored number of generated scintillation photons (Figure 4.3). A linear relation was noted with an R^2 of 0.99999 when the numbers of optical photons generated versus the number expected were plotted for source Position II under the traditional approach; as shown in Figure 4.3. The estimate was calculated by

knowing that approximately 40,000 photons are produced per MeV deposited in the NaI crystal. Similarly, for other cases including the small crystal, this relationship was observed to be linear. These results validate that GEANT4 simulations produce an expected number of optical photons proportional to the gamma-ray energy deposited within the crystal.

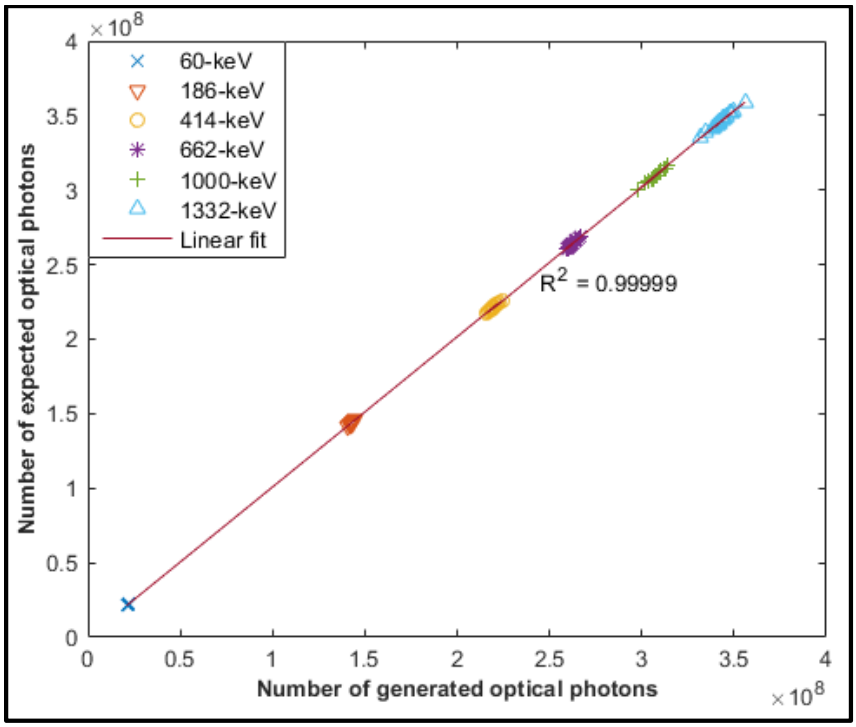

Figure 4.3. A plot displaying the number of optical photons generated to the corresponding number of optical photons expected for source Position II under the TA.

Figure 4.4 illustrates the normalized PE fraction as a function of gamma-ray energy when the source was placed at Position I under the TA. The number of PE events

for a particular energy was normalized to the total number of PE events over all six gamma-ray energies. These results had a relative standard error within 1%.

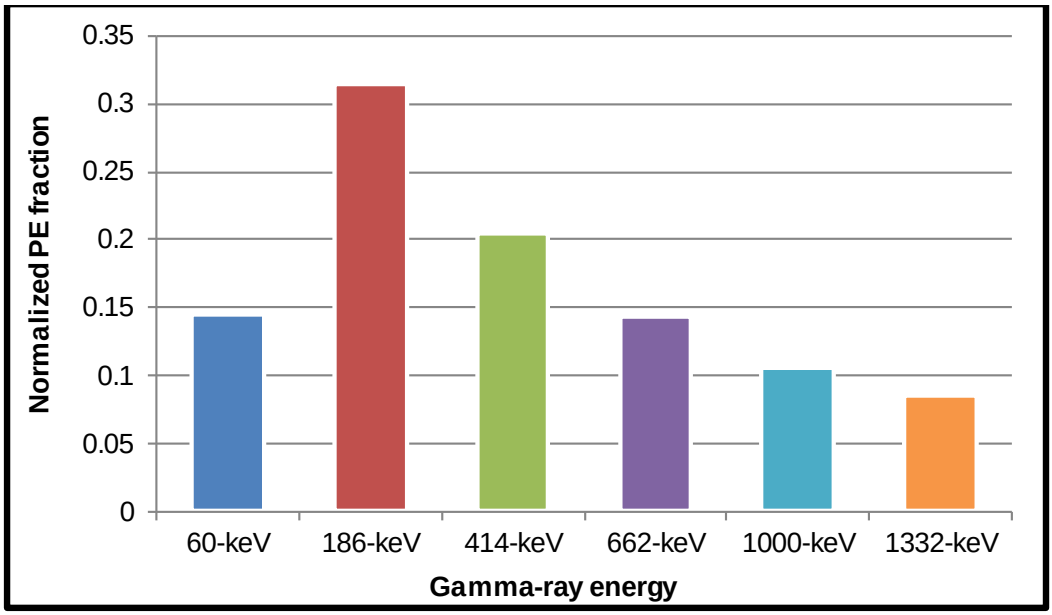

Figure 4.4. Normalized PE fractions as a function of gamma-ray energy for the source at Position I.

For the 186-keV gamma rays, the number of PE events was highest (Figure 4.4). With an increase in the gamma-ray energy, the number of PE events decreased. The only exception was the 60-keV gamma ray. Many of these gamma rays were absorbed in the crystal casing, which was stainless-steel. This absorption resulted in a lower number of PE events for the 60-keV gamma ray and subsequently, a lower normalized PE fraction. Similarly, the normalized PE fractions for the source at Position II under the TA are presented in Figure 4.5.

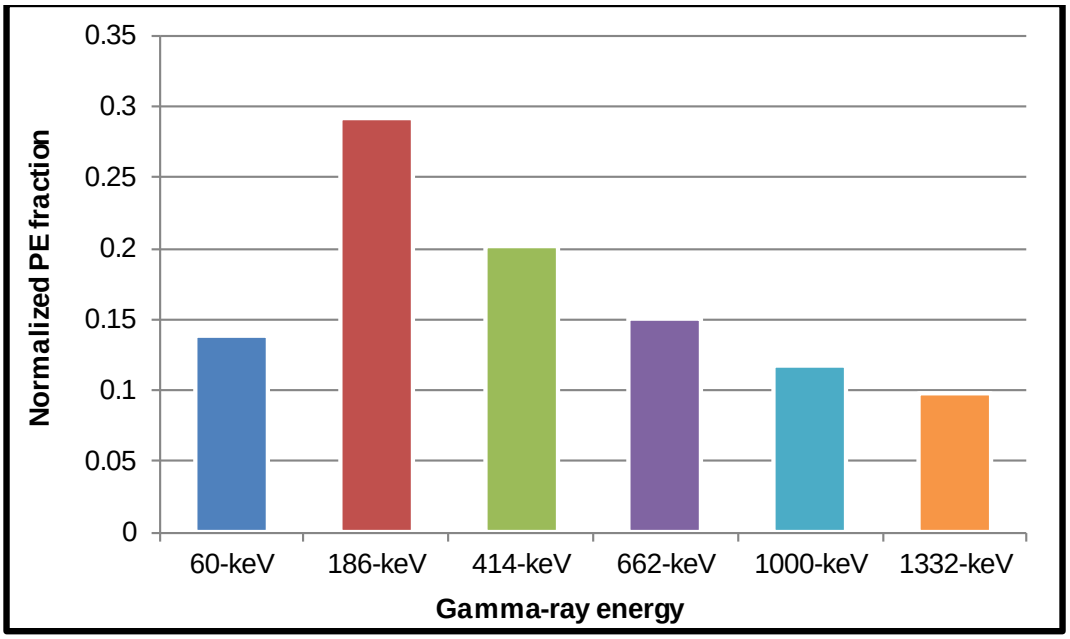

Figure 4.5. Normalized PE fractions as a function of gamma-ray energy for the source at Position II.

It was observed that the normalized PE fractions for the source at Position II followed a trend similar to the source at Position I (Figures 4.4 and 4.5). This finding was anticipated as the probability of PE event decreases with an increase in gamma-ray energy. For both source positions, normalized PE fractions under the PA were the same as the fractions under the TA. The normalized PE fractions when the source was placed at Position I and Position II were approximately same; as shown in Figures 4.4 and 4.5. However, the number of PE events for a specific energy differed for both source locations. Figure 4.6 displays the ratio of a number of PE events for the source at Position I to the source at Position II as a function of gamma-ray energy.

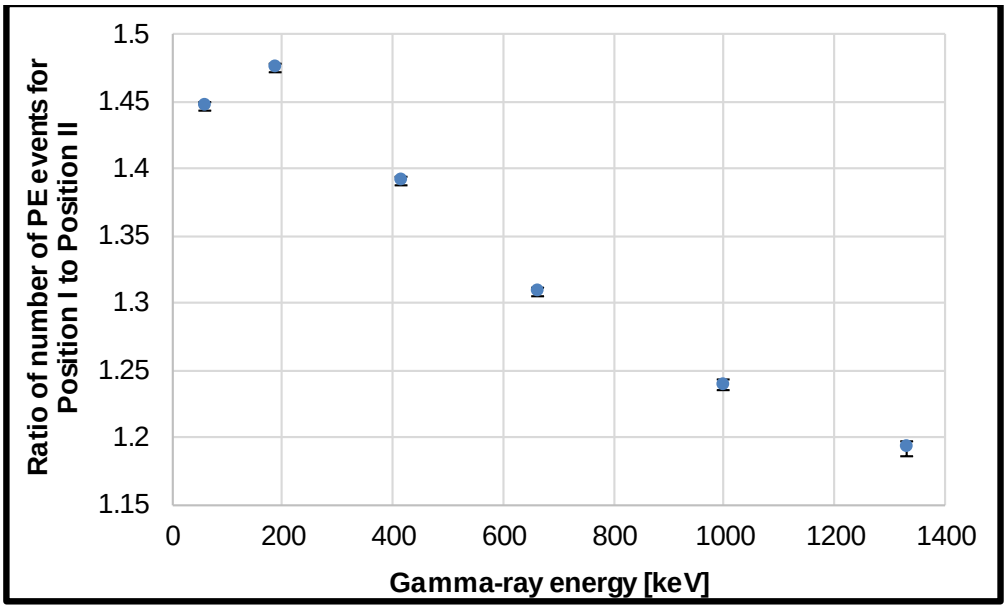

Figure 4.6. The ratio of a number of PE events for the source at Position I to the source at Position II as a function of gamma-ray energy.

The trend in Figure 4.6 holds true for either defined approach, TA or PA, as the number of PE events for a specific energy were the same for both approaches. The 1-σ error bars are included but are masked by data points with the relative standard errors less than 1%. Source Position I faces one of the 10.16-cm x 40.64-cm surfaces, and Position II faces one of the 5.08-cm x 40.64-cm surfaces. Thus, the large NaI crystal presents a larger solid angle to the source at Position I compared to Position II. The difference in the solid angle resulted in a greater number of PE events for source Position I.

The light collection efficiency is the ratio of the total number of optical photons collected to the number generated. Figures 4.7 and 4.8 display the light collection efficiency as a function of gamma-ray energy for the source at Position I and Position II,

respectively, for the large NaI crystal. Also, the plots include their corresponding standard error of the mean (1-σ), but the relative standard errors were lower than 1%, and thus are not visible in plots.

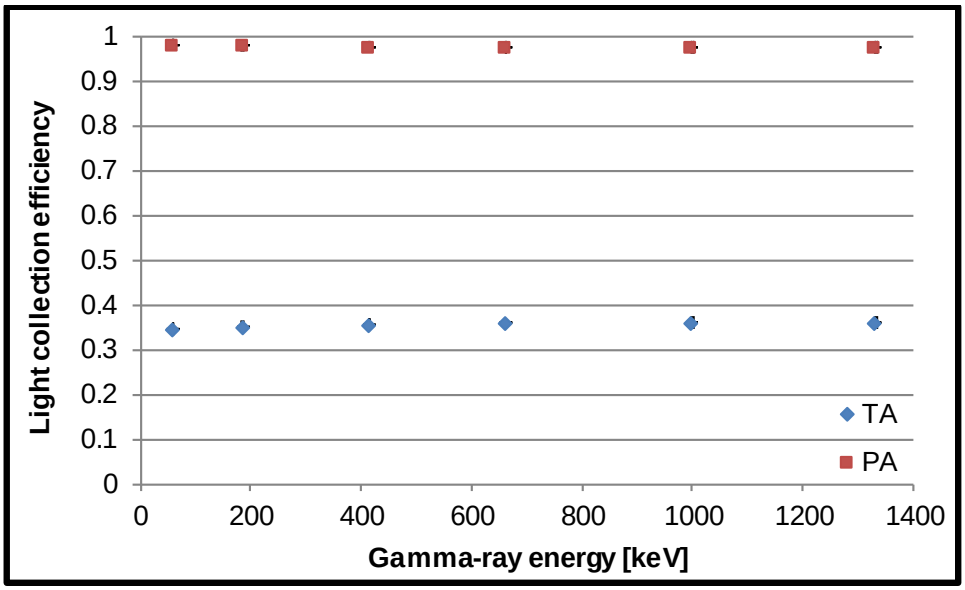

Figure 4.7. Light collection efficiency as a function of gamma-ray energy for the large crystal when the source at Position I.

The prime observation was an increase in the light collection efficiency when utilizing the PA (Figures 4.7 and 4.8). In relation to the TA, the efficiency under the PA was large by a factor of 2.7X, regardless of the source position and gamma-ray energy. The number of total PE events for both the PA and the TA was the same, which indicates the larger light collection efficiency for the PA resulted from an increase in the total number of optical photons collected. Thus, when using a single photocathode, a fraction

of optical photons was lost internally via bulk absorption [70], [71]. The internal loss of optical photons is a function of the crystal size and the collection area. For the large crystal, the optical photon collection area is only ~4% of the total surface area. This collection area property has resulted in a lower collection efficiency under the TA compared to the PA.

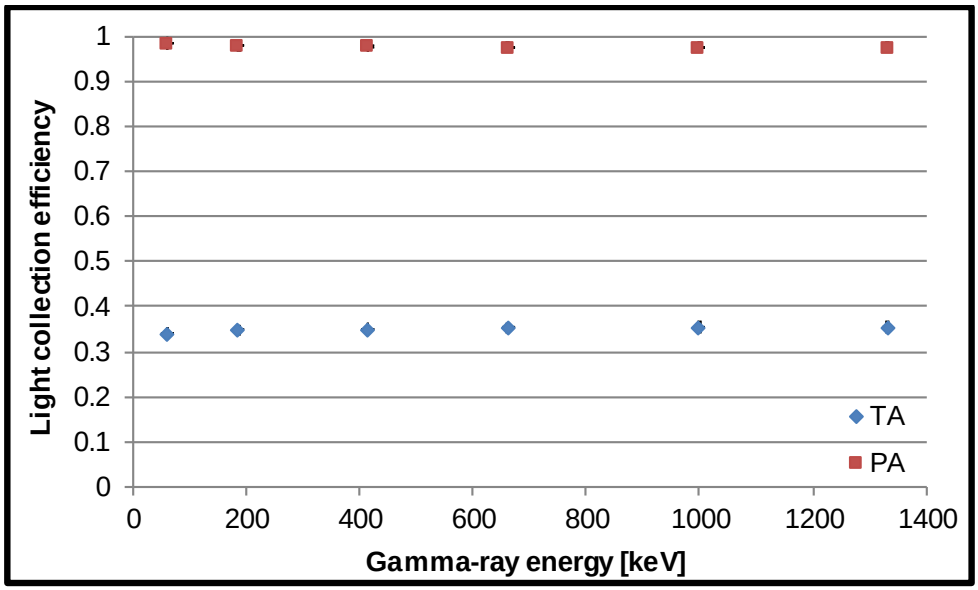

Figure 4.8. Light collection efficiency as a function of gamma-ray energy for the large crystal when the source at Position II.

Also, for the TA, as gamma-ray energy increased, the light collection efficiency was observed to remain constant (Figures 4.7 and 4.8). This constant efficiency was due to the probability of absorption being constant. As gamma-ray energy increases, even though more optical photons are being produced, their probability of reaching the

photocathode without the internal loss remains the same. This property results in the constant light collection efficiency over a broad range of gamma-ray energies. Similarly, using the PA, a constant light collection efficiency over the defined gamma-ray energies was observed. The physics is the same here as in the TA; as gamma-ray energy increases, more optical photons are generated, but their probability of absorption remains the same. This constant absorption probability has resulted in a steady light collection efficiency.

Figures 4.7 and 4.8 also illustrate the influence that source position has on the number of optical photons generated and collected. Source Position I faced the 10.16-cm x 40.64-cm surface, and Position II faced the 5.08-cm x 40.64-cm surface. When a gamma-ray interacts in the crystal, the scintillation photons travel isotropically from their point of generation. The optical photons are collected on a single crystal surface using the TA and on all crystal surfaces using the PA. Under each approach, since the probability of an optical photon being absorbed in the crystal is constant, the light collection efficiency was statistically the same for both source positions. However, the crystal presents a larger solid angle for Position I in relation to Position II and thus, more PE events occur when the source is at Position I. Eventually, more optical photons will be generated and collected for the source at Position I, but the light collection efficiency remains constant. As a result, data points for Position I are "masked" under data points for Position II, as seen in Figures 4.7 and 4.8.

To further illustrate the advantage of the PA over the TA, a collection ratio was calculated. This ratio is the number of optical photons collected in the PA compared to the TA. These results along with their corresponding standard error of the mean (1-σ) are

plotted in Figure 4.9. The relative standard errors for these results were less than 1%. When using the PA, bulk absorption was decreased because optical photons were collected on all crystal surfaces. The collection ratio was observed to be roughly constant over all gamma energies (Figure 4.9). The ratio ranged from 2.74 to 2.84 and 2.75 to 2.89 for the source at Position I and Position II, respectively, with a corresponding average of 2.76 and 2.8. The constant collection ratio was as expected with the constant light collection efficiency under both approaches over all energies and increased efficiency under the PA. The light collection efficiency under the PA was larger than the TA by a factor of at least 2.7X. Overall, these properties have resulted in the increase in collection ratio under the PA.

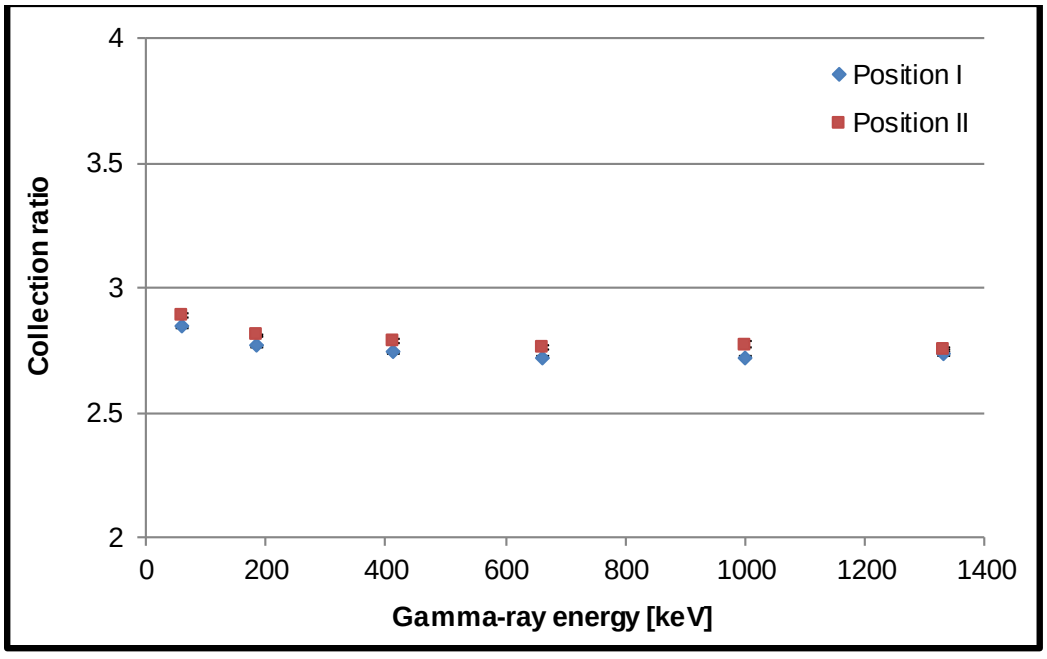

Figure 4.9. Collection ratio as a function of energy for the large crystal.

The analysis of the signal generation capability of photoelectrons in the RICs was carried out by investigating the distribution of optical photons on the incident surface. The concentration of collected optical photons was assessed by assuming each optical photon is at the center of the circle (representing RSA). The circle was conceptualized to perform concentration analysis on optical photons collected on the incident surface. The number of optical photons concentrated within the circle was grouped as a function of radius from 1-μm to 10-μm, with an increment of 1-μm. From each PE event, the maximum concentrations of optical photons corresponding to the radius (of the circle) were noted. For a set of simulations, the maximum concentrations for each radius were found.

Table 4.1 lists the averaged maxima of optical photons as a function of radius for the 60-keV, 662-keV, and 1332-keV gamma rays. The results presented were observed on the incident surface (Position I). The sample standard deviations associated with results are itemized and had an average relative standard error less than 10%. For the 60-keV gamma ray, the concentration of optical photons observed across all radii sizes was largest followed by the 662 and 1332-keV gammas. As gamma-ray energy increases, the MFP of the gamma ray also increases. This increase means, on an average, the first gamma-ray interaction occurs much deeper in the crystal. With optical photons leaving their generation site isotropically, the solid angle presented by the collecting surface area to this point of optical photons generation also increases. Thus, for the 60-keV gamma ray, optical photons were localized within a few μm, while for the 662 and 1332-keV gamma rays optical photons were more dispersed (Table 4.1). This dispersion can be expected as

the mean free path in a NaI crystal for the total interaction of 60, 662, and 1332-keV gamma-ray is approximately 0.04, 3.31, and 5.28-cm, respectively [72].

Table 4.1. The mean maximum concentration of optical photons as a function of radius for the large crystal.

Radius	**Number of optical photons**		
[μm]	60-keV	662-keV	1332-keV
1	652 ± 119	95 ± 75	85 ± 52
2	837 ± 101	204 ± 137	169 ± 90
3	937 ± 88	296 ± 184	258 ± 121
4	996 ± 72	378 ± 216	344 ± 148
5	1033 ± 62	456 ± 246	435 ± 169
6	1061 ± 53	526 ± 273	511 ± 188
7	1083 ± 47	596 ± 302	584 ± 201
8	1099 ± 44	662 ± 329	656 ± 202
9	1112 ± 42	726 ± 356	728 ± 201
10	1122 ± 41	790 ± 379	798 ± 198

The above analysis provided an estimate of the mean maximum concentration of optical photons over a broad range of gamma-ray energies. The number of optical photons is proportional to the number of photoelectrons that interact in the RIC RSAs to generate

a signal. The number of optical photons concentrated on a 5-µm x 5-µm RSA needs to surpass 1000 for the RSA to generate an appropriate signal. The concentration of optical photons for the 60-keV gamma-ray was suitable enough to generate a signal. However, for the higher energy gamma-rays, the number of optical photons concentrated on the RSA did not meet the threshold.

Small Crystal

The normalized PE fractions as a function of gamma-ray energy for the small crystal are presented in Figure 4.10. The results had a relative standard error within 1%. The trend of normalized PE fractions is similar to the large crystal normalized PE fractions for the source at Position I (Figure 4.4) and Position II (Figure 4.5).

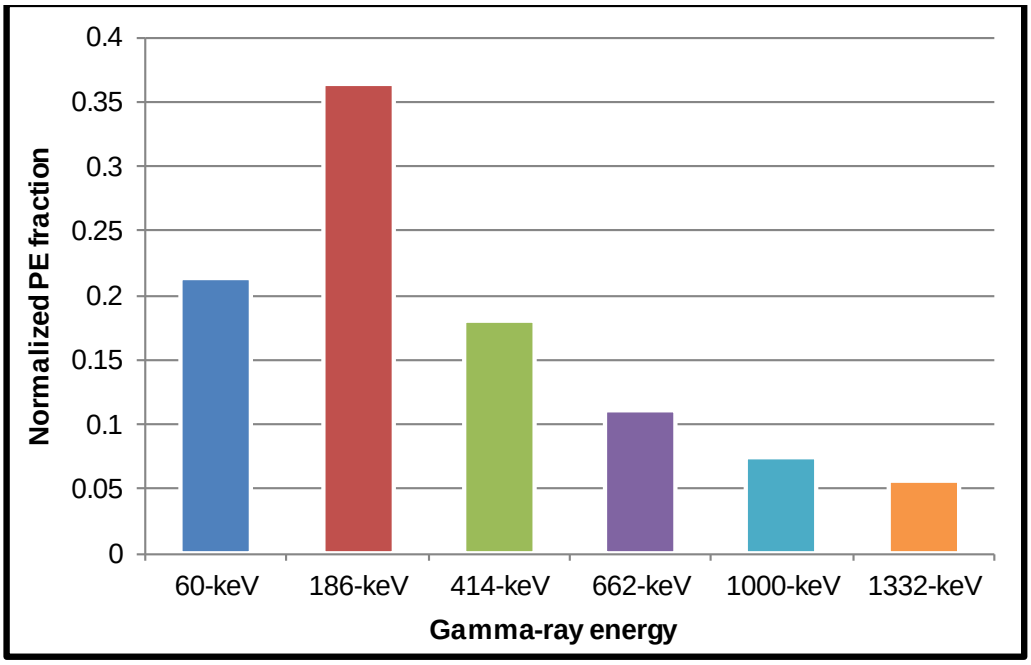

Figure 4.10. Normalized PE fractions as a function of gamma-ray energy for the small crystal.

For the 186-keV gamma rays, the number of PE events was highest and then, with an increase in energy the number of PE events decreased (Figure 4.10). The only exception was the 60-keV gamma rays as they were absorbed in the crystal casing. This absorption resulted in the lower number of PE events for the 60-keV gamma rays and subsequently, a lower normalized PE fraction.

Figure 4.11 shows the light collection efficiency for the small NaI crystal as a function of gamma-ray energy. The corresponding standard errors of the mean (1-σ) are included, but as the relative standard errors were less than 1%, they are obscured by data points.

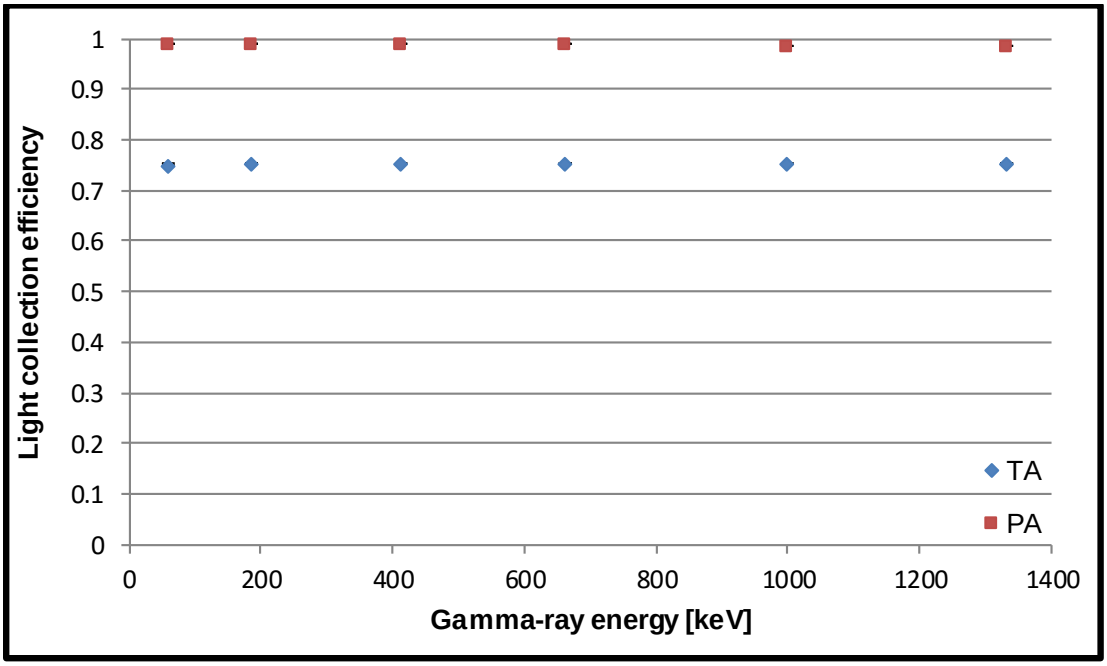

Figure 4.11. Light collection efficiency as a function of gamma-ray energy for the small crystal under both approaches.

The light collection efficiency for the small crystal under both defined approaches is included in Figure 4.11. Under the TA, the efficiency over all gamma-ray energies was constant. With an increase in gamma-ray energy, the increase in the number of optical photons generated was annulled by a decrease in the number of PE events due to the crystal size. Bulk absorption played less an effect here as the crystal volume is reduced, which meant a smaller travel distance to the single photocathode. This resulted in a higher efficiency compared to the large crystal, by a factor of 2X.

Under the PA, the light collection efficiency trend was similar to the traditional approach which is a constant efficiency as gamma-ray energy increased. As gamma-ray energy increases, even though more optical photons are being produced, their probability of reaching the photocathode without the internal loss remains the same. It was still nearly a constant and greater than the TA by a factor of 1.3X. Overall, under the TA, efficiencies were greater in the small crystal compared to the large crystal because the optical photons traveled shorter distances and thus exhibited lower internal losses via bulk absorption. With the PA, the collection efficiencies for the small and large crystal were the same.

The collection ratio as a function of gamma-ray energy is presented in Figure 4.12. The trend is similar to the large crystal analysis of collection ratio – constant ratios across all energies (Figure 4.9). However, the collection ratio magnitude was lower because fewer optical photons were lost via bulk absorption compared to the large crystal. The reduction in losses was due to the reduced volume of the NaI crystal, which provides a shorter distance for the scintillation photons to reach the single photocathode. For the gamma-ray energies employed in this research, the collection using the PA was still a

minimum of 1.3X higher than the TA due to the shorter distance to reach photocathodes across all surfaces.

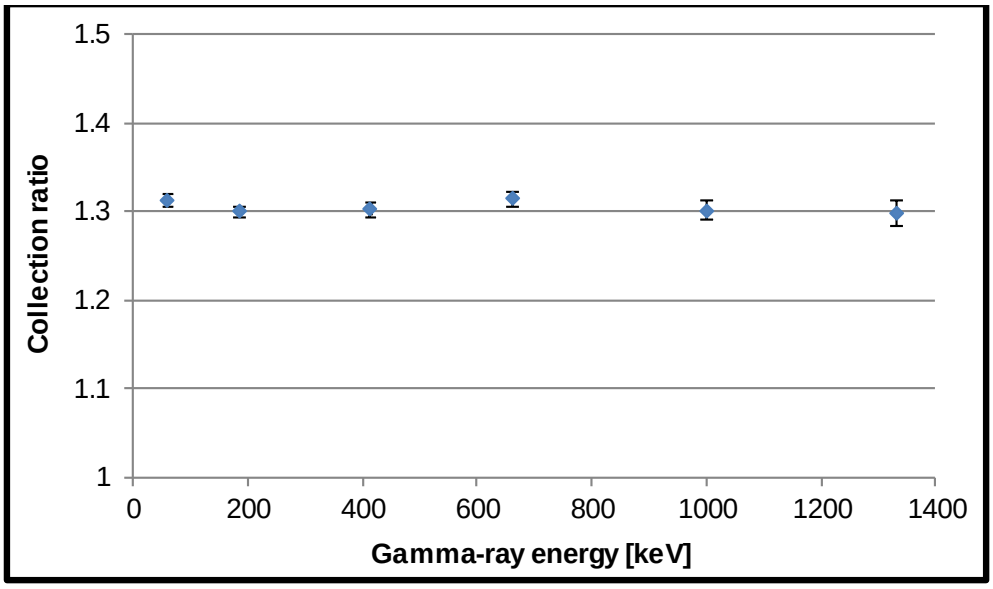

Figure 4.12. Collection ratio as a function of gamma-ray energy for the small crystal.

Analysis of the concentration of optical photons for the small crystal was carried out using the 60-keV, 662-keV, and 1332-keV gamma rays. Table 4.2 presents the averaged maximum concentration of optical photons as a function of radius, which represents the size of the RSA. The results presented along with sample standard deviations were analyzed for the incident surface facing the gamma-ray source. The average relative standard error was below 7%.

Table 4.2. The mean maximum concentration of optical photons as a function of radius for the small crystal.

Radius	**Number of optical photons**		
[μm]	60-keV	662-keV	1332-keV
1	605 ± 160	191 ± 54	82 ± 106
2	789 ± 122	358 ± 105	158 ± 126
3	888 ± 101	458 ± 134	236 ± 156
4	946 ± 91	534 ± 155	308 ± 185
5	987 ± 80	653 ± 174	379 ± 210
6	1018 ± 72	751 ± 192	441 ± 230
7	1042 ± 65	820 ± 206	499 ± 244
8	1061 ± 62	876 ± 216	558 ± 259
9	1076 ± 57	947 ± 226	618 ± 274
10	1090 ± 53	1064 ± 237	677 ± 290

The 60-keV gamma ray shows the maximum concentration of optical photons, followed by the 662-keV and 1332-keV gamma rays (Table 4.2). Similar to the large crystal analysis, the concentration of optical photons on the incident surface of the small crystal depended on the gamma-ray energy (MFP property). In addition, the magnitude of the concentrations of optical photons was comparable for both crystals. This similarity in the average maxima of optical photons was as expected. Following a PE event, the optical photons are isotropically emitted and then collected on the crystal surface. The crystal size has a bearing on the probability of a PE event and its location for a particular gamma ray.

However, on average, when the concentration of optical photons is analyzed in such a small area of μm^2, the crystal size plays a minimal role.

The number of optical photons interacting in a 5-µm X 5-µm RSA is not sufficient enough to realize a RIC-based PMT replacement. The number of optical photons collected over the size of the RSA is proportional to the number of photoelectrons that interact in the RIC RSAs to generate a signal. The number of optical photons concentrated on a 5-µm X 5-µm RSA would have to exceed 1000 to generate a suitable signal in the RSA. As with the large crystal analysis, the concentration of optical photons for the 60-keV gamma-ray generated just enough photons to produce a signal. However, for the higher energy gamma-rays, the number of optical photons concentrated within the RSA were below the workable limit.

RSA Limitation

Based on the above results, the RIC cannot be utilized as a PMT replacement for gamma ray detection using a NaI scintillator for higher gamma energies. It can be used for lower energies. However, the size of RIC RSA determines its capacitance, and the voltage produced under a strike. The RSA can be made larger, but this results in increased capacitance and lower voltage. With an increased capacitance the developed voltage decreases, which in turn means that a higher number of photoelectrons are needed to generate a detectable signal. Even for a large RSA the optical photons were not sufficiently concentrated enough to generate a signal. With the low number of optical photons interacting in the RSA, the signal strength to measure the interacting gamma-ray energy

is poor. The PA can be used to realize a gamma counter, but to perform energy measurement, further study is required.

Summary

The RICs are specially designed integrated circuits which contain radiation hardened and radiation sensitive areas that can be used to perform radiation detection. A new gamma-ray detection system is proposed that will use a NaI crystal, photocathodes, and RICs. Simulations were accomplished with the GEANT4 code to investigate changes in the light collection efficiency when the photocathode and RICs were placed on all crystal surfaces to collect optical photons, as opposed to the PMT approach in which a photocathode is placed on a single crystal surface. Results obtained under both approaches were analyzed and compared. The effect of source position and crystal size was also investigated.

Differences between the traditional and proposed approaches were mainly influenced by the absorption of light in the crystal. In the PA, the light was collected closer to the points of gamma-ray interactions while under the TA the light traveled further and had a greater probability of being absorbed in the crystal before reaching the photocathode. The increase in collection efficiency seen with the PA design corresponds to the reduction in internal losses compared to the TA. This effect was most pronounced in the large NaI crystal where over 2.7X more optical photons were collected. As gamma-ray energy increased, the collection ratio was observed to remain constant. This observation resulted because the probability of each optical photon being absorbed is constant. Similar trends

were noted in the small crystal analysis of the collection ratio but were only 1.3X larger because of the reduced optical photon absorption.

For the large crystal, analyses of the light collection efficiency and collection ratio were carried out by changing the source position. The collection efficiency and collection ratio were observed to be independent of the source position. However, the PE event was observed to depend on the source position. When the source was at Position I, the number of PE events was larger by a factor of 1.17X than when the source was at Position II. This increase in the number of PE events for the source at Position I was due to the difference in the solid angle, i.e., the large NaI crystal presents a larger solid angle to the source at Position I compared to Position II.

Analysis of the concentration of optical photons on an RSA was carried out for both crystal sizes. The number of optical photons collected over the size of the RSA is proportional to a number of photoelectrons that interacts in RSAs of the RIC to generate a signal. For the 60-keV gamma ray, the concentration of optical photons observed across defined radii sizes was largest followed by the 662 and 1332-keV gammas. As gamma-ray energy increases, the MFP of the gamma ray also increases. Consequently, the first gamma-ray interaction occurs much deeper in the crystal. With optical photons being emitted isotropically from their point of generation, the solid angle presented by the collecting surface area to the point of optical photon generation also increases. Thus, for the 60-keV gamma ray, optical photons were localized within a few μm while for the 662 and 1332-keV gamma rays the optical photons were more dispersed.

The number of optical photons concentrated within a 5-µm x 5-µm RSA needs to be on the order of 1,000 for the RSA to generate a suitable signal. However, for higher energy gamma-rays interacting in the crystal, the number of optical photons concentrated within the RSA were below this workable limit. The PA can be used to realize a gamma counter, but to perform energy measurement, further study is required.

CHAPTER V

CONCLUSION

The RIC design was proposed by researchers at Texas A&M University to develop new types of radiation detectors to detect alpha particles, beta particles, gamma rays, and neutrons. This dissertation evaluated the scope of RICs for three aspects, namely a) the alpha/beta probe design, b) the neutron detector design, and c) the PMT replacement. The unique and cost-efficient aspect about these three detectors is that they all use the same RIC.

The proposed RIC-alpha/beta probe design has two regions: one to detect alpha particles and another to detect betas along with their $E_{\beta max}$. In order to perform the $E_{\beta max}$ discrimination, the MPD property of betas in the attenuators was utilized. In MCNPX, plate glass, Pyrex® glass, Lucite®, and natural rubber as attenuators were studied. For the proof of concept, materials in the wedged form were analyzed. The natural rubber as wedged attenuator was observed to show a superior $E_{\beta max}$ discrimination capability compared to other defined attenuators. The $E_{\beta max}$ resolution capability of 50-keV is possible using natural rubber attenuator.

The proposed RIC-neutron detector design uses enriched boron (96% ^{10}B) as the neutron-reactive coating to generate secondary charged particles. As RSAs are sensitive to charged particles, to detect neutrons using the RIC such coating is required. In MCNPX, other neutron-reactive materials (natural boron, B_4C, and LiF) were also studied. Alpha interactions were observed to facilitate predominantly the signal generation in RSAs to detect neutrons. With the optimal thickness of 3-μm enriched boron, the signal to noise

ratio of thermal neutrons was increased by approximately three orders of magnitude. In the present era, where neutron detection suffers either from a high cost or low sensitivity, the use of these RICs to detect thermal neutrons will be valuable. The thermal neutron sensitivity of the RIC was estimated to be 17-cps-nv^{-1} (counted pulses per neutron flux density), which can be increased by stacking the RICs. This sensitivity is comparable to the sensitivity of the ^{3}He tube, and so the RIC neutron detector when perfected can be an alternative to the ^{3}He detectors, which are increasingly becoming expensive with the shortage of ^{3}He.

The proposed RIC-gamma ray detection system used a sodium iodide crystal, photocathodes, and RICs. Photocathodes under the proposed approach (PA) were placed on all crystal surfaces to collect optical photons and generate photoelectrons. This "all surfaces" approach is feasible using RICs because they are relatively cheap, physically small, and less-power intensive as compared to the photomultiplier tube (PMT). The photoelectrons interact to generate an electrical signal in RSAs and thereby, the RIC detects gamma rays. The collection ratio, defined as the ratio of the total number of optical photons collected through the PA compared to the traditional approach (TA), was found to be a function of the crystal size, gamma-ray energy, and the source position. The collection of scintillation photons was higher using the PA (proposed approach) by a factor of 2.7X for the large crystal (regardless of position and energy) and 1.3X for the small crystal. The concentration of photoelectrons as a function of the RSA size was assessed to optimize the RIC RSA size. The number of optical photons concentrated within the RSA for the higher energy gamma-rays was below the feasible limit to generate a suitable

signal. With the low number of optical photons interacting in the RSA, the signal strength to measure the interacting gamma-ray energy is poor. The PA can be used to realize a gamma counter, but to perform energy measurement, further study is required.

The use of the RICs simplifies power requirements and reduces external electronic peripherals (since signal processing units are integrated on the RIC), thereby offering a great advantage. In addition, radiation detectors developed using the RICs will also reduce the detector footprint. Existing detectors do not provide these benefits of the RIC. The chip design will be modified while carrying out the analysis to maximize their detection capabilities.

REFERENCES

1. C. Marianno and S. Khatri; *ARI-MA: A Low-Cost Integrated Circuit Based Detector for Gamma and Neutron Detection*. Proposal.
2. R. Garg, N. Jayakumar and S. Khatri; *A Design Approach for Radiation-hard Digital Electronics*. ACM/IEEE Design Automation Conference (DAC), July 24-28 2006, pp. 773-778.
3. R. Garg; *Analysis and Design of Resilient VLSI Circuits*. Dissertation, Texas A&M University, 2009.
4. G. Knoll; *Radiation Detection and Measurement*. John Wiley & Sons, New York Fourth Edition.
5. J. Turner; *Atoms, Radiation, and Radiation Protection*. Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, Third Edition, 2007.
6. K. Shultis and R. Faw; *Radiation Shielding*. American Nuclear Society, Inc., 2000.
7. *Compendium of Neutron Spectra and Detector Responses for Radiation Protection Purposes*. Technical Reports Series No 403, The International Atomic Energy Agency, Vienna, 2001.
8. *Detection of Radioactive Materials at Borders*. Jointly sponsored by IAEA, WCO, EOROPOL and INTERPOL, IAEA-TECDOC-1312, The International Atomic Energy Agency, Vienna, 2002.
9. E. Rutherford and H. Geiger; *An Electrical Method of Counting the Number of Alpha Particles From Radioactive Substances*. Harrison & Sons, Proceedings of the Royal Society (London), Series A, Vol. 81, No. 546, pp. 141–161, 1908.

10. J. Townsend; *The Conductivity Produced in Gases by the Motion of Negatively Charged Ions*. Nature, Philosophical Magazine, Series 6, 1 (2), pp. 198-227, 1901.
11. S. Bhade, P. Reddy, A. Narayan, K. Narayan, D. Babu and D. Sharma; *Standardization of Calibration Procedures for Quantification of Gross Alpha and Gross Beta Activities using Liquid Scintillation Counter*. Springer, Journal of Radioanalytical and Nuclear Chemistry, Vol. 284, Issue 2, pp. 267-375, 2010.
12. S. Liebson; *The Discharge Mechanism of Self-quenching Geiger-Mueller Counters*. APS Physics, Physical Review, Vol. 72, No. 7, pp. 602–608, 1947.
13. *History of Portable Radiation Detection Instrumentation from the period 1920–60*, February, 2014. http://www.national-radiation-instrument-catalog.com/
14. G. Bertolini and A. Coche; *Semiconductor Detectors*. Elsevier Publishing Co., Inc., 1968.
15. G. Dearnaley; *Nuclear Radiation by Solid State Devices*. IOP Science, Journal of Scientific Instruments, 43, 869, 1966.
16. J. Duggan, W. Adams, R. Scroggs, and L. Anthony; *Charged-Particle Detector Experiments for the Modern Physics Laboratory*. AAPT, American Journal of Physics, 35(7), 631, 1967.
17. S. Gotoh and Z. Takagi; *Silicon Surface Barrier Detector*. AESJ, Journal of Nuclear Science and Technology 1, No. 8, pp 311-315, 1964.
18. H. Kim, S. Park, J. Ha, S. Cho and Y. Kim; *Characteristics of Silicon Surface Barrier Radiation Detectors for Alpha Particle Detection*. Springer, Journal of the Korean Physical Society, Vol. 52, No. 6, pp. 1754-1758, 2008.

19. W. Leo; *Techniques for Nuclear and Particle Physics Experiments: A How-to Approach*. Springer Science & Business Media, Dec 6, 2012.
20. F. Goulding and R. Pehl; *Semiconductor Detectors*. Section IIIA, Nuclear Spectroscopy and Reactions, J. Cerny, Ed. Academic Press, 1974.
21. D. Bouldin; *The Measurement of Alpha Particle Emissions from Semiconductor Memory Materials*. Springer, Journal of Electronics Materials, Vol. 10, No. 4, 1981.
22. J. Coltman; *The Scintillation Counter*. IEEE, Proceedings of the I.R.E. – Waves and Electrons Section, pp. 671-682, 1949.
23. J. Koster, J. Johnson, D. MacArthur and S. Walters; *Alpha Detection as a Probe for Counter Proliferation*. IEEE Security Technology Proceedings, pp. 6-19, 1994.
24. X. Tuo, Z. Li, K. Mu and X. Li; *Alpha Contamination Monitoring Instrument Based on LRAD Technique for Complex Surface Monitoring*. AESJ, Journal of Nuclear Science and Technology, Supplement 5, pp. 282-285, 2008.
25. J. Ha and H. Kim; *Alpha-Ray Spectrometry at High Temperature by using a Compound Semiconductor Detector*. Elsevier, Applied Radiation and Isotopes 81, pp. 165-168, 2013.
26. J. Yao, J. Brenizer, R. Hui and S. Yin; *Standoff Alpha Radiation Detection via Excited State Absorption of Air*. American Institute of Physics, Applied Physics Letters 101, 254101, 2013.
27. J. Foster; *A Comparison of the ZnS (Ag) Scintillation Detector to the Silicon Semiconductor Detector for Quantification of Alpha Radioactivity in Aqueous Solutions*. Clemson University, All Theses, Paper 10, 2006.

28. Ortec; *Introduction to Charged-Particle Detectors*, August 2015. http://www.ortec-online.com/download/introduction-charged-particle-detectors.pdf
29. Ortec; *Passivated Implanted Planar Silicon (PIPS) Detectors*, August 2015. http://www.canberra.com/products/detectors/pips-detectors.asp
30. *Nuclear Spectroscopy at Room Temperature*. Technical Note, Enertec/Schlumberger, 1983.
31. D. Bouldin; *The Measurement of Alpha Particle Emissions from Semiconductor Memory Materials*. IEEE Journal of Electronics Materials, Vol. 10, No. 4, 1981.
32. D. Williams and B. Carter; *Transmission Electron Microscopy: A Textbook for Materials Science*. Springer, Second Edition, 2006.
33. B. Montag, M. Reichenberger, K. Arpin, M. Sunder, K. Nelson, P. Ugorowski and D. McGregor; *Synthesis and Characterization of LiZnP and LiZnAs Semiconductor Material*. Elsevier, Journal of Crystal Growth 412, pp. 103-108, 2015.
34. D. McGregor and K. Shultis; *Spectral Identification of Thin-Film-Coated and Solid-Form Semiconductor Neutron Detectors*. Elsevier, Nuclear Instruments and Methods in Physics Research A 517, pp. 180-188, 2004.
35. B. Saleh and M. Teich; *Fundamentals of Photonics*. John Wiley & Sons, Inc., 1991.
36. M. Barbagallo, L. Cosentino, V. Forcina, C. Marchetta, A. Pappalardo, P. Peerani, C. Scire, S. Scire, M. Schillaci, S. Vaccaro, G. Vecchio and P. Finocchiaro; *Thermal Neutron Detection Using a Silicon Pad Detector and 6LiF Removable Converters*. American Institute of Physics, Review of Scientific Instruments 84, 033503, 2013.

37. D. McGregor and K. Shultis; *Reporting Detection Efficiency for Semiconductor Neutron Detectors: A Need for Standard*. Elsevier, Nuclear Instruments and Methods in Physics Research A 632, pp. 167-174, 2011.

38. R. Kouzes, J. Ely, L. Erikson, W. Kernan, A. Lintereur and others; *Neutron Detection Alternatives to ^{3}He for National Security Applications*. Elsevier, Nuclear Instruments and Methods in Physics Research Section A 623, pp. 1035-1045, 2010.

39. D. McGregor, S. Bellinger and K. Shultis; *Present Status of Microstructured Semiconductor Neutron Detectors*. Elsevier, Journal of Crystal Growth 379, pp. 99-110, 2013.

40. M. Barbagallo, L. Cosentino, G. Greco, R. Montereali, A. Pappalardo, C. Scire, S. Scire, M. Vincenti and P. Finocchiaro; *Development of a Thermal Neutron Detector based on Scintillating Fibers and Silicon Photomultipliers*. American Institute of Physics, Review of Scientific Instruments 81, 093503, 2010.

41. C. Celik, K. Unlu, V. Narayanan and M. Irwin; *Soft Error Modeling and Analysis of the Neutron Intercepting Silicon Chip (NISC)*. Elsevier, Nuclear Instruments and Methods in Physics Research A 652, pp. 370-373, 2011.

42. C. Marianno and S. Khatri; *Unique Radiation Detection Method Using Si based Integrated Circuits*. Annual Meeting of the Health Physics Society, Salt Lake City, UT. Jun 27 - July 1, 2010.

43. C. Marianno and S. Khatri; *Unique Radiation Detection Method Using Si based Integrated Circuits*. Annual Meeting of the Health Physics Society, 2010.

44. D. Peterson and B. Rose; *Low Cost, Low Power, High Sensitivity, Real Time Neutron Detection Microsystem*. American Institute of Physics Conference Proceedings 632, 3, 2002.
45. S. Donati; *Single-Photon Detectors: From Traditional PMT to Solid-State SPAD-Based Technology*. IEEE Journal of Selected Topics in Quantum Electronics, Vol. 20, No. 6, 2014.
46. C. Zavaleta; *Photodiodes as Substitutes for Photomultiplier Tubes*, August, 2015. http://ric.uthscsa.edu/personalpages/lancaster/DI2_Projects_2003/Photodiode_vs_PMT.pdf
47. R. Agishev, A. Comeron, J. Bach, A. Rodriguez, M. Sicard, J. Riu and S. Royo; *Lidar with SiPM: Some Capabilities and Limitations in Real Environment*. Elsevier, Optics & Laser Technology 49, pp. 86-90, 2013.
48. E. Johnson, C. Stapels, X. Chen, F. Augustine and J. Christian; *Large-area CMOS Solid-State Photomultipliers and Recent Developments*. Elsevier, Nuclear Instruments and Methods in Physics Research A 652, pp. 494-499, 2011.
49. C. Niclass, M. Sergio and E. Charbon; *A CMOS 64 x 48 Single Photon Avalanche Diode Array with Event-Driven Readout*. ESSCIR, 2006.
50. S. Sordo, Leonardo Abbene, E. Caroli, A. Mancini, A. Zappettini and P. Ubertini; *Progress in the Development of CdTe and CdZnTe Semiconductor Radiation Detectors for Astrophysical and Medical Applications*. Sensors 9, pp. 3491-3526, 2009.
51. C. Stapels, W. Lawrence and J. Christian; *CMOS Solid-State Photomultiplier for Detecting Scintillation Light in Harsh Environment*. SNIC Symposium, 0218, 2006.

52. C. Stapels, W. Lawrence, J. Christian, M. Squillante, G. Entine, F. Augustine, P. Dokhale and M. McClish; *Solid-State Photomultiplier in CMOS Technology for Gamma-Ray Detection and Imaging Applications*. Radiation Monitoring Devices Inc., IEEE Conference, Vol. 5, 2005.

53. J. Schmall, J. Du, M. Judenhofer, P. Dokhale, J. Christian, M. McClish, K. Shah and S. Cherry; *A Study of Position-Sensitive Solid-State Photomultiplier Signal Properties.* Manuscript, IEEE Trans Nuclear Science, pp. 1074-1083, 2014.

54. Hamamatsu; *Photomultiplier Tubes and Assemblies: For Scintillation Counting & High Energy Physics*, August 2015. http://neutron.physics.ucsb.edu/docs/High_energy_PMT_TPMO0007E01.pdf

55. G. Bonanno, M. Belluso, S. Billotta, P. Finocchiaro and A. Pappalardo; *Geiger Avalanche Photodiodes (G-APDs) and Their Characterization*, August 2015. http://cdn.intechopen.com/pdfs-wm/17227.pdf

56. Sense Light; *An Introduction to the Silicon Photomultiplier*, August 2015. http://www.sensl.com/downloads/ds/TN%20-%20Intro%20to%20SPM%20Tech.pdf

57. E. Charbon; *Advanced Optoelectronic Circuits: Detectors and Image Sensors*. Circuits and Systems, Encyclopedia of Life Support Systems, August 2015. http://www.eolss.net/sample-chapters/c05/e6-195-08.pdf

58. Kromek; *Introducing Kromek's New Generation of Scintillator Products*, August 2015. http://www.kromek.com/index.php/component/dtracker/?filetitle=brochure&filename=SIGMA.pdf

59. Kromek; *TN15 High Sensitivity Ultra-Compact Thermal Neutron Scintillation Detector*, August 2015. http://www.kromek.com/index.php/component/dtracker/?filetitle=brochure&filename=TN15.pdf

60. Hamamatsu; *MPPC and MPPC Module for Precision Measurement*, August 2015. https://www.hamamatsu.com/resources/pdf/ssd/mppc_kapd0004e.pdf

61. M. Shah, C. Marianno, S. Khatri and D. Boyle; *Simulation Analysis of Scintillation in a Sodium Iodide Detector*. Institute of Nuclear Material Management 55th Annual Meeting, 2014.

62. D. Pelowitz, J. Durkee, J. Elson, M. Fensin and others; *MCNPX 2.7.E Extensions*. Los Alamos National Laboratory report LA-UR-11-01502, March 2011.

63. D. Pelowitz; *MCNPX User's Manual, Version 2.7.0*. Los Alamos National Laboratory report LA-CP-11-00438, April 2011.

64. S. Agostinelli, J. Allison, K. Amako and others; *Geant4 – A Simulation Toolkit*. Elsevier, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Vol. 506, Issue 3, pp. 250-303, July 2003.

65. C. Mahajan; *Mass Attenuation Coefficients of Beta Particles in Elements*. JSRR, Science Research Reporter 2(2), pp. 135-141, April 2012.

66. University of California, Berkeley, August 2015; http://physics111.lib.berkeley.edu/Physics111/Reprints/BRA/03-Beta_Decay.pdf

67. K. Holbert; *Charged Particle Ionization and Range*, Arizona State University, August 2015. http://holbert.faculty.asu.edu/eee460/IonizationRange.pdf

68. M. Chadwick and others; *ENDF/B-VII.0: Next Generation Evaluated Nuclear Data Library for Nuclear Science and Technology*. Elsevier, Nuclear Data Sheets 107, 2931, 2006.

69. Massachusetts Institute of Technology, August 2015; http://web.mit.edu/8.13/www/JLExperiments/38/tgm-neutron-detectors.pdf

70. G. Hull, S. Du, T. Niedermayr, S. Payne, N. Cherepy, A. Drobshoff and L. Fabris; *Light Collection Optimization in Scintillator-Based Gamma-Ray Spectrometers.* Elsevier, Nuclear Instruments and Methods in Physics Research Section A, Vol. 588, pp. 384-388, 2008.

71. F. Vittori, F. Notaristefani, T. Malatesta, D. Puertolas; *A Study on Light Collection of Small Scintillating Crystals. Elsevier,* Nuclear Instruments and Methods in Physics Research Section A, Vol. 452, pp. 245-251, 2000.

72. M. Berger, J. Hubbell, S. Seltzer, J. Chang, J. Coursey, R. Sukumar, D. Zucker, and K. Olsen; *XCOM: Photon Cross Sections Database.* NIST Standard Reference Database 8 (XGAM), PML, Radiation Physics Division. http://physics.nist.gov/PhysRefData/Xcom/html/xcom1.html

APPENDIX A

```
------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------
                                        PLATE GLASS
------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------
c
c      MCNP Code for Plate Glass Attenuator
c
c     Cell Cards
c
1       1  -2.4  -1 1029  4 -5 2 -3  imp:e=1  imp:p=1
2       1  -2.4  -1 1029  4 -5 3 -13 imp:e=2  imp:p=1
3       1  -2.4  -1 1029  4 -5 13 -14 imp:e=4  imp:p=1
4       1  -2.4  -1 1029  4 -5 14 -15 imp:e=8  imp:p=1
5       1  -2.4  -1 1029  4 -5 15 -16 imp:e=16  imp:p=1
6       1  -2.4  -1 1029  4 -5 16 -17  imp:e=32  imp:p=1
7       1  -2.4  -1 1029  4 -5 17 -18 imp:e=64  imp:p=1
8       1  -2.4  -1 1029  4 -5 18 -19 imp:e=128  imp:p=1
9       1  -2.4  -1 1029  4 -5 19 -20 imp:e=256  imp:p=1
10      1  -2.4  -1 1029  4 -5 20 -21 imp:e=512  imp:p=1
20      2  -0.001205 (1:-1029:-4:5:-2:21)  -12 imp:e=1  imp:p=1
30      0  12 imp:e=0  imp:p=0
c
c


c
c    Surface Cards
c
1      px       0
1029       px  -5
2      py       3
3  1    py      3
13 2    py      3
14 3    py      3
15 4    py      3
16 5    py      3
17 6    py      3
18 7    py      3
19 8    py      3
20 9    py      3
21 10   py      3
```

```
4     pz      0
5     pz      5
12    rpp -100 100 -100 100 -100 100
141   pz   2.45
142   pz   2.55
30    px   -0.1
31    px   -0.2
32    px   -0.3
33    px   -0.4
34    px   -0.5
35    px   -0.6
36    px   -0.7
37    px   -0.8
38    px   -0.9
39    px   -1
40    px   -1.1
41    px   -1.2
42    px   -1.3
43    px   -1.4
44    px   -1.5
45    px   -1.6
46    px   -1.7
47    px   -1.8
48    px   -1.9
49    px   -2
50    px   -2.1
51    px   -2.2
52    px   -2.3
53    px   -2.4
54    px   -2.5
55    px   -2.6
56    px   -2.7
57    px   -2.8
58    px   -2.9
59    px   -3
60    px   -3.1
61    px   -3.2
62    px   -3.3
63    px   -3.4
64    px   -3.5
65    px   -3.6
66    px   -3.7
67    px   -3.8
68    px   -3.9
```

```
69    px    -4
70    px    -4.1
71    px    -4.2
72    px    -4.3
73    px    -4.4
74    px    -4.5
75    px    -4.6
76    px    -4.7
77    px    -4.8
78    px    -4.9
79    px    -5
80    px    -5.1
81    px    -5.2
82    px    -5.3
83    px    -5.4
84    px    -5.5
85    px    -5.6
86    px    -5.7
87    px    -5.8
88    px    -5.9
89    px    -6
90    px    -6.1
91    px    -6.2
92    px    -6.3
93    px    -6.4
94    px    -6.5
95    px    -6.6
96    px    -6.7
97    px    -6.8
98    px    -6.9
99    px    -7
c
c


c
c
c    Material Cards
c
c
mode  e p
TR1 0  0 0 0.99995  -0.01000  0.00000  0.01000  0.99995  0 0 0 1
TR2 0  0 0 0.99980  -0.02000  0.00000  0.02000  0.99980  0 0 0 1
TR3 0  0 0 0.99955  -0.02999  0.00000  0.02999  0.99955  0 0 0 1
TR4 0  0 0 0.99920  -0.03997  0.00000  0.03997  0.99920  0 0 0 1
```

```
TR5 0 0 0 0.99875  -0.04994 0.00000  0.04994 0.99875  0 0 0 1
TR6 0 0 0 0.99820  -0.05989 0.00000  0.05989 0.99820  0 0 0 1
TR7 0 0 0 0.99756  -0.06983 0.00000  0.06983 0.99756  0 0 0 1
TR8 0 0 0 0.99682  -0.07975 0.00000  0.07975 0.99682  0 0 0 1
TR9 0 0 0 0.99597  -0.08964 0.00000  0.08964 0.99597  0 0 0 1
TR10 0 0 0 0.99504 -0.09950  0.00000 0.09950  0.99504 0 0 0 1
nps 10000000
sdef  x=-3.95 y=2 z=2.5 erg=1.95 par=3
cut:e j 1e-3
F1:e 21
F21:e  21
F31:e  21
F41:e  21
F51:e  21
F61:e  21
F71:e  21
F81:e  21
F91:e  21
F101:e 21
F111:e 21
F121:e 21
F131:e 21
F141:e 21
F151:e 21
F161:e 21
F171:e 21
F181:e 21
F191:e 21
F201:e 21
E0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 1.2 1.4 1.6 1.8 2 2.01
FS1     55     -141    -56    142
FS21    56     -141    -57    142
FS31    57     -141    -58    142
FS41    58     -141    -59    142
FS51    59     -141    -60    142
FS61    60     -141    -61    142
FS71    61     -141    -62    142
FS81    62     -141    -63    142
FS91    63     -141    -64    142
FS101   64     -141    -65    142
m1     8000.03e  -0.4598
      11000.03e  -0.096441
      14000.03e  -0.336553
      20000.03e  -0.107205
```

```
m2     6000.03e  -0.000124
       7014.03e  -0.752290
       7015.03e  -0.002977
       8016.03e  -0.231153
       8017.03e  -0.000094
       8018.03e  -0.000535
      18000.03e  -0.012827
```

END

PYREX® GLASS

```
c
c      MCNP Code for Pyrex® Glass Attenuator
c
c     Cell Cards
c
1       1  -2.23  -1 1029 4 -5 2 -3  imp:e=1  imp:p=1
2       1  -2.23  -1 1029 4 -5 3 -13 imp:e=2  imp:p=1
3       1  -2.23  -1 1029 4 -5 13 -14 imp:e=4  imp:p=1
4       1  -2.23  -1 1029 4 -5 14 -15 imp:e=8  imp:p=1
5       1  -2.23  -1 1029 4 -5 15 -16 imp:e=16  imp:p=1
6       1  -2.23  -1 1029 4 -5 16 -17  imp:e=32  imp:p=1
7       1  -2.23  -1 1029 4 -5 17 -18 imp:e=64  imp:p=1
8       1  -2.23  -1 1029 4 -5 18 -19 imp:e=128  imp:p=1
9       1  -2.23  -1 1029 4 -5 19 -20 imp:e=256  imp:p=1
10      1  -2.23  -1 1029 4 -5 20 -21 imp:e=512  imp:p=1
20      2  -0.001205 (1:-1029:-4:5:-2:21)  -12 imp:e=1  imp:p=1
30      0  12 imp:e=0  imp:p=0
c
c

c
c    Surface Cards
c
1      px       0
1029       px  -5
2      py       3
3  1    py      3
13 2    py      3
```

```
14 3    py      3
15 4    py      3
16 5    py      3
17 6    py      3
18 7    py      3
19 8    py      3
20 9    py      3
21 10   py      3
4       pz      0
5       pz      5
12      rpp -100  100 -100  100 -100  100
141     pz   2.45
142     pz   2.55
30      px    -0.1
31      px    -0.2
32      px    -0.3
33      px    -0.4
34      px    -0.5
35      px    -0.6
36      px    -0.7
37      px    -0.8
38      px    -0.9
39      px    -1
40      px    -1.1
41      px    -1.2
42      px    -1.3
43      px    -1.4
44      px    -1.5
45      px    -1.6
46      px    -1.7
47      px    -1.8
48      px    -1.9
49      px    -2
50      px    -2.1
51      px    -2.2
52      px    -2.3
53      px    -2.4
54      px    -2.5
55      px    -2.6
56      px    -2.7
57      px    -2.8
58      px    -2.9
59      px    -3
60      px    -3.1
```

```
61    px   -3.2
62    px   -3.3
63    px   -3.4
64    px   -3.5
65    px   -3.6
66    px   -3.7
67    px   -3.8
68    px   -3.9
69    px   -4
70    px   -4.1
71    px   -4.2
72    px   -4.3
73    px   -4.4
74    px   -4.5
75    px   -4.6
76    px   -4.7
77    px   -4.8
78    px   -4.9
79    px   -5
80    px   -5.1
81    px   -5.2
82    px   -5.3
83    px   -5.4
84    px   -5.5
85    px   -5.6
86    px   -5.7
87    px   -5.8
88    px   -5.9
89    px   -6
90    px   -6.1
91    px   -6.2
92    px   -6.3
93    px   -6.4
94    px   -6.5
95    px   -6.6
96    px   -6.7
97    px   -6.8
98    px   -6.9
99    px   -7
c
c


c
c    Material Cards
```

```
c
mode e p
TR1 0 0 0 0.99995  -0.01000  0.00000  0.01000  0.99995  0 0 0 1
TR2 0 0 0 0.99980  -0.02000  0.00000  0.02000  0.99980  0 0 0 1
TR3 0 0 0 0.99955  -0.02999  0.00000  0.02999  0.99955  0 0 0 1
TR4 0 0 0 0.99920  -0.03997  0.00000  0.03997  0.99920  0 0 0 1
TR5 0 0 0 0.99875  -0.04994  0.00000  0.04994  0.99875  0 0 0 1
TR6 0 0 0 0.99820  -0.05989  0.00000  0.05989  0.99820  0 0 0 1
TR7 0 0 0 0.99756  -0.06983  0.00000  0.06983  0.99756  0 0 0 1
TR8 0 0 0 0.99682  -0.07975  0.00000  0.07975  0.99682  0 0 0 1
TR9 0 0 0 0.99597  -0.08964  0.00000  0.08964  0.99597  0 0 0 1
TR10 0 0 0 0.99504  -0.09950  0.00000  0.09950  0.99504  0 0 0 1
nps 6000000
sdef  x=-4.26 y=2 z=2.5 erg=1.95 par=3
cut:e j 1e-3
F1:e 21
F21:e 21
F31:e 21
F41:e 21
F51:e 21
F61:e 21
E0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 1.2 1.4 1.6 1.8 2 2.01
FS1     58     -141     -59     142
FS21     59     -141     -60     142
FS31     60     -141     -61     142
FS41     61     -141     -62     142
FS51     62     -141     -63     142
FS61     63     -141     -64    142
m1     5000.03e  -0.040064
       8000.03e  -0.539562
       11000.03e  -0.028191
       13000.03e  -0.011644
       14000.03e  -0.37722
       19000.03e  -0.003321
m2     6000.03e  -0.000124
       7014.03e  -0.752290
       7015.03e  -0.002977
       8016.03e  -0.231153
       8017.03e  -0.000094
       8018.03e  -0.000535
      18000.03e  -0.012827
```

END

LUCITE®


```
c
c      MCNP Code for Lucite® Attenuator
c
c     Cell Cards
c
1       1  -1.19  -1  1029 4 -5 2 -3  imp:e=1  imp:p=1
2       1  -1.19  -1  1029 4 -5 3 -13 imp:e=2  imp:p=1
3       1  -1.19  -1  1029 4 -5 13 -14 imp:e=4  imp:p=1
4       1  -1.19  -1  1029 4 -5 14 -15 imp:e=8  imp:p=1
5       1  -1.19  -1  1029 4 -5 15 -16 imp:e=16  imp:p=1
6       1  -1.19  -1  1029 4 -5 16 -17  imp:e=32  imp:p=1
7       1  -1.19  -1  1029 4 -5 17 -18 imp:e=64  imp:p=1
8       1  -1.19  -1  1029 4 -5 18 -19 imp:e=128  imp:p=1
9       1  -1.19  -1  1029 4 -5 19 -20 imp:e=256  imp:p=1
10       1  -1.19  -1  1029  4 -5 20 -21 imp:e=512  imp:p=1
20      2  -0.001205 (1:-1029:-4:5:-2:21)  -12 imp:e=1  imp:p=1
30      0  12 imp:e=0  imp:p=0
c
c

c
c    Surface Cards
c
1      px       0
1029       px  -7
2      py       3
3  1    py      3
13 2    py      3
14 3    py      3
15 4    py      3
16 5    py      3
17 6    py      3
18 7    py      3
19 8    py      3
20 9    py      3
21 10   py      3
4      pz       0
5      pz       5
12     rpp -100  100 -100  100 -100  100
```

141 pz 2.45
142 pz 2.55
30 px -0.1
31 px -0.2
32 px -0.3
33 px -0.4
34 px -0.5
35 px -0.6
36 px -0.7
37 px -0.8
38 px -0.9
39 px -1
40 px -1.1
41 px -1.2
42 px -1.3
43 px -1.4
44 px -1.5
45 px -1.6
46 px -1.7
47 px -1.8
48 px -1.9
49 px -2
50 px -2.1
51 px -2.2
52 px -2.3
53 px -2.4
54 px -2.5
55 px -2.6
56 px -2.7
57 px -2.8
58 px -2.9
59 px -3
60 px -3.1
61 px -3.2
62 px -3.3
63 px -3.4
64 px -3.5
65 px -3.6
66 px -3.7
67 px -3.8
68 px -3.9
69 px -4
70 px -4.1
71 px -4.2

```
72    px    -4.3
73    px    -4.4
74    px    -4.5
75    px    -4.6
76    px    -4.7
77    px    -4.8
78    px    -4.9
79    px    -5
80    px    -5.1
81    px    -5.2
82    px    -5.3
83    px    -5.4
84    px    -5.5
85    px    -5.6
86    px    -5.7
87    px    -5.8
88    px    -5.9
89    px    -6
90    px    -6.1
91    px    -6.2
92    px    -6.3
93    px    -6.4
94    px    -6.5
95    px    -6.6
96    px    -6.7
97    px    -6.8
98    px    -6.9
99    px    -7
c
c


c
c    Material Cards
c
mode  e p
TR1 0  0 0 0 0.99990  -0.01428  0.00000  0.01428  0.99990  0 0 0 1
TR2 0  0 0 0 0.99959  -0.02856  0.00000  0.02856  0.99959  0 0 0 1
TR3 0  0 0 0 0.99908  -0.04282  0.00000  0.04282  0.99908  0 0 0 1
TR4 0  0 0 0 0.99837  -0.05705  0.00000  0.05705  0.99837  0 0 0 1
TR5 0  0 0 0 0.99746  -0.07125  0.00000  0.07125  0.99746  0 0 0 1
TR6 0  0 0 0 0.99635  -0.08540  0.00000  0.08540  0.99635  0 0 0 1
TR7 0  0 0 0 0.99504  -0.09950  0.00000  0.09950  0.99504  0 0 0 1
TR8 0  0 0 0 0.99353  -0.11355  0.00000  0.11355  0.99353  0 0 0 1
TR9 0  0 0 0 0.99184  -0.12752  0.00000  0.12752  0.99184  0 0 0 1
```

```
TR10 0 0 0 0.98995 -0.14142 0.00000 0.14142 0.98995 0 0 0 1
nps 10000000
sdef x=-5.58 y=2 z=2.5 erg=1.95 par=3
cut:e j 1e-3
F1:e 21
F21:e 21
F31:e 21
F41:e 21
F51:e 21
F61:e 21
F71:e 21
F81:e 21
F91:e 21
F101:e 21
F111:e 21
F121:e 21
F131:e 21
F141:e 21
F151:e 21
F161:e 21
F171:e 21
F181:e 21
F191:e 21
F201:e 21
E0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 1.2 1.4 1.6 1.8 2 2.01
FS1    66    -141   -67   142
FS21   67    -141   -68   142
FS31   68    -141   -69   142
FS41   69    -141   -70   142
FS51   70    -141   -71   142
FS61   71    -141   -72   142
FS71   72    -141   -73   142
FS81   73    -141   -74   142
FS91   74    -141   -75   142
FS101  75    -141   -76   142
FS111  76    -141   -77   142
FS121  77    -141   -78   142
FS131  78    -141   -79   142
FS141  79    -141   -80   142
FS151  80    -141   -81   142
FS161  81    -141   -82   142
m1     01000.03e  -0.080538
       06000.03e  -0.599848
       08000.03e  -0.319614
```

```
m2     6000.03e  -0.000124
       7014.03e  -0.752290
       7015.03e  -0.002977
       8016.03e  -0.231153
       8017.03e  -0.000094
       8018.03e  -0.000535
      18000.03e  -0.012827
```

END

NATURAL RUBBER

```
c
c      MCNP Code for Natural Rubber Attenuator
c
c     Cell Cards
c
1       1  -.92  -1 1029  4 -5 2 -3  imp:e=1  imp:p=1
2       1  -.92  -1 1029  4 -5 3 -13 imp:e=2  imp:p=1
3       1  -.92  -1 1029  4 -5 13 -14 imp:e=4  imp:p=1
4       1  -.92  -1 1029  4 -5 14 -15 imp:e=8  imp:p=1
5       1  -.92  -1 1029  4 -5 15 -16 imp:e=16  imp:p=1
6       1  -.92  -1 1029  4 -5 16 -17  imp:e=32  imp:p=1
7       1  -.92  -1 1029  4 -5 17 -18 imp:e=64  imp:p=1
8       1  -.92  -1 1029  4 -5 18 -19 imp:e=128  imp:p=1
9       1  -.92  -1 1029  4 -5 19 -20 imp:e=256  imp:p=1
10       1  -.92  -1 1029 4 -5 20 -21 imp:e=512  imp:p=1
11       1  -.92  -1 1029 4 -5 21 -22 imp:e=1024  imp:p=1
20       2  -0.001205 (1:-1029:-4:5:-2:22)  -12 imp:e=1  imp:p=1
30       0  12 imp:e=0  imp:p=0
c
c


c
c    Surface Cards
c
1      px       0
1029       px  -7
2      py       3
3  1    py      3
```

```
13 2    py      3
14 3    py      3
15 4    py      3
16 5    py      3
17 6    py      3
18 7    py      3
19 8    py      3
20 9    py      3
21 10   py      3
22 11   py      3
4      pz      0
5      pz      5
12     rpp -100 100 -100 100 -100 100
141    pz   2.45
142    pz   2.55
30     px   -0.1
31     px   -0.2
32     px   -0.3
33     px   -0.4
34     px   -0.5
35     px   -0.6
36     px   -0.7
37     px   -0.8
38     px   -0.9
39     px   -1
40     px   -1.1
41     px   -1.2
42     px   -1.3
43     px   -1.4
44     px   -1.5
45     px   -1.6
46     px   -1.7
47     px   -1.8
48     px   -1.9
49     px   -2
50     px   -2.1
51     px   -2.2
52     px   -2.3
53     px   -2.4
54     px   -2.5
55     px   -2.6
56     px   -2.7
57     px   -2.8
58     px   -2.9
```

59 px -3
60 px -3.1
61 px -3.2
62 px -3.3
63 px -3.4
64 px -3.5
65 px -3.6
66 px -3.7
67 px -3.8
68 px -3.9
69 px -4
70 px -4.1
71 px -4.2
72 px -4.3
73 px -4.4
74 px -4.5
75 px -4.6
76 px -4.7
77 px -4.8
78 px -4.9
79 px -5
80 px -5.1
81 px -5.2
82 px -5.3
83 px -5.4
84 px -5.5
85 px -5.6
86 px -5.7
87 px -5.8
88 px -5.9
89 px -6
90 px -6.1
91 px -6.2
92 px -6.3
93 px -6.4
94 px -6.5
95 px -6.6
96 px -6.7
97 px -6.8
98 px -6.9
99 px -7
c
c

```
c
c    Material Cards
c
mode e p
TR1 0 0 0 0.99990  -0.01428 0.00000  0.01428 0.99990  0 0 0 1
TR2 0 0 0 0.99959  -0.02856 0.00000  0.02856 0.99959  0 0 0 1
TR3 0 0 0 0.99908  -0.04282 0.00000  0.04282 0.99908  0 0 0 1
TR4 0 0 0 0.99837  -0.05705 0.00000  0.05705 0.99837  0 0 0 1
TR5 0 0 0 0.99746  -0.07125 0.00000  0.07125 0.99746  0 0 0 1
TR6 0 0 0 0.99635  -0.08540 0.00000  0.08540 0.99635  0 0 0 1
TR7 0 0 0 0.99504  -0.09950 0.00000  0.09950 0.99504  0 0 0 1
TR8 0 0 0 0.99353  -0.11355 0.00000  0.11355 0.99353  0 0 0 1
TR9 0 0 0 0.99184  -0.12752 0.00000  0.12752 0.99184  0 0 0 1
TR10 0 0 0 0.98995 -0.14142  0.00000 0.14142  0.98995 0 0 0 1
TR11 0 0 0 0.98788 -0.15524  0.00000 0.15524  0.98788 0 0 0 1
nps 10000000
sdef  x=-6.56 y=2 z=2.5 erg=1.95 par=3
cut:e j 1e-3
F1:e 22
F21:e 22
F31:e 22
F41:e 22
F51:e 22
F61:e 22
F71:e 22
F81:e 22
F91:e 22
F101:e 22
F111:e 22
F121:e 22
F131:e 22
F141:e 22
F151:e 22
F161:e 22
F171:e 22
F181:e 22
F191:e 22
F201:e 22
E0 0.001  0.002 0.003  0.004 0.005  0.006 0.007  0.008 0.009  0.01 &
  0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1 0.2 0.3 0.4 0.5 &
  0.6 0.7 0.8 0.9 1 1.2 1.4 1.6 1.8 2 2.01
FS1     75     -141    -76    142
FS21    76     -141    -77    142
FS31    77     -141    -78    142
```

```
FS41    78    -141    -79    142
FS51    79    -141    -80    142
FS61    80    -141    -81    142
FS71    81    -141    -82    142
FS81    82    -141    -83    142
FS91    83    -141    -84    142
FS101   84    -141    -85    142
FS111   85    -141    -86    142
FS121   86    -141    -87    142
FS131   87    -141    -88    142
FS141   88    -141    -89    142
FS151   89    -141    -90    142
FS161   90    -141    -91    142
m1     01000.03e  -0.118371
       06000.03e  -0.881629
m2      6000.03e  -0.000124
       7014.03e  -0.752290
       7015.03e  -0.002977
       8016.03e  -0.231153
       8017.03e  -0.000094
       8018.03e  -0.000535
      18000.03e  -0.012827
```

END

APPENDIX B

ENRICHED BORON


```
c
c      MCNP Code for Enriched Boron Neutron Detector
c
c     Cell Cards
c
mphys  on
c
c
1      1  -2.37    1 -2 3 -4 5 -6
2      2  -2.32    2 -27 3 -4 5 -6
3      3  -8.03    #1 #2 21 -28 23 -24 25 -26
4      4 -0.001293  #1 #2 #3 -7
99     0           7

c
c    Surface Cards
c
1        pz  0.0
2        pz  0.0003
3        px  0
4        px  3
5        py  0
6        py  5
7        so  10
21       pz  -0.10
23       px  -0.10
24       px   3.10
25       py  -0.10
26       py   5.1
27       pz   0.00032
28       pz   0.10032
10       py   2.5000
11       py   2.5005
100      py   1.45
110      py   1.55
200      py   3.45
```

```
210     py   3.55
300     py   0.45
310     py   0.55
400     py   4.45
410     py   4.55
30      px      0.1
31      px      0.2
32      px      0.3
33      px      0.4
34      px      0.5
35      px      0.6
36      px      0.7
37      px      0.8
38      px      0.9
39      px      1
40      px      1.1
41      px      1.2
42      px      1.3
43      px      1.5000
44      px      1.5005
45      px      1.6
46      px      1.7
47      px      1.8
48      px      1.9
49      px      2
50      px      2.1
51      px      2.2
52      px      2.3
53      px      2.4
54      px      2.5
55      px      2.6
56      px      2.7
57      px      2.8
58      px      2.9

c
c    Material Cards
c
mode  n a e # p
c SDEF  SUR=7 NRM=-1 ERG=0.025E-6  dir=d1
sdef pos=1.45  2.5  -1.1 vec= 0 0  1 dir=d1  erg=2.5e-8
si1  -1 0.9 1
sp1   0 0.95 0.05
sb1   0 0 1
```

```
c sb1 -21 2
imp:n  1 1 1 1 0
imp:a  1 1 1 1 0
imp:p  1 1 1 1 0
imp:e  1 1 1 1 0
imp:#  1 1 1 1 0
fcl:n  1 1 0 0 0
c f361:#  2
c fs361 44 -10 -43 11
c e361    0.001 0.02 0.1 0.2 0.3 0.5 1 1.1 1.2 1.3 1.4 1.44 1.46
c         1.48 1.5 1.6 1.7 1.75 1.77 1.79
c         1.9 2 3 4 5 6
f371:#  27
fs371 44 -10 -43 11
e371    0.001  0.02 0.1 0.2 0.3 0.5 1 1.1 1.2 1.3 1.4 1.44 1.46
       1.48 1.5 1.6 1.7 1.75 1.77 1.79
       1.9 2 3 4 5 6
c f391:a  2
c fs391 44 -10 -43 11
c e391    0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
c         1.1 1.2 1.3 1.4 1.44 1.46 1.48 1.5 1.6 1.7 1.75 1.77 1.79
c         1.9 2 3 4 5 6
f401:a  27
fs401 44 -10 -43 11
e401    0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
       1.1 1.2 1.3 1.4 1.44 1.46 1.48 1.5 1.6 1.7 1.75 1.77 1.79
       1.9 2 3 4 5 6
phys:n  6j 2
phys:p  5j
c phys:a  100  3j 1 j j 3j 0.9
cut:n  2j 0 0
cut:p  2j 0 0
cut:a  j 0
cut:#  j 0
F26:a  2
e26 0 .05 .1 .15 .189 .193 .25 .3 .35 .4 .45 .5 .55 .57 0.574  0.6 1 2 3 4 5 6
F16:#  2
e16 0 .05 .1 .15 .189 .193 .25 .3 .35 .4 .45 .5 .55 .57 0.574  0.6 1 2 3 4 5 6
f41:a  2
fs41 44 -10 -43 11
e41    0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
       1.1 1.2 1.3 1.4 1.44 1.46 1.48 1.5 1.6 1.7 1.75 1.77 1.79
       1.9 2 3 4 5 6
f451:#  2
```

```
fs451 44 -10 -43 11
e451   0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
       1.1 1.2 1.3 1.4 1.44 1.46 1.48 1.5 1.6 1.7 1.75 1.77 1.79
       1.9 2 3 4 5 6
m1     5010. -0.96
       5011. -0.04
m2     14028. -1
m3     26000 -0.66598      6000 -0.00052      24000 -0.13800
       28000 -0.15200      42000 -0.01460     14000 -0.00920
       25055 -0.01740      22000 -0.00230     $ SS
m4     7014. -0.79
       8016. -0.21
c mx1:n model
c mx1:n model
c mx2:n model
nps 800000
```

END

BORON CARBIDE

```
c
c      MCNP Code for Boron Carbide Neutron Detector
c
c     Cell Cards
c
mphys on
c
c
1      1  -2.52   1 -2 3 -4 5 -6
2      2  -2.32   2 -27 3 -4 5 -6
3      3  -8.03   #1 #2 21 -28 23 -24 25 -26
4      4 -0.001293  #1 #2 #3 -7
99     0          7

c
c    Surface Cards
c
1        pz  0.0
2        pz  0.0003
```

```
3       px  0
4       px  3
5       py  0
6       py  5
7       so  10
21      pz  -0.10
23      px  -0.10
24      px   3.10
25      py  -0.10
26      py   5.1
27      pz   0.00032
28      pz   0.10032
10      py   2.45
11      py   2.55
100     py   1.45
110     py   1.55
200     py   3.45
210     py   3.55
300     py   0.45
310     py   0.55
400     py   4.45
410     py   4.55
30      px      0.1
31      px      0.2
32      px      0.3
33      px      0.4
34      px      0.5
35      px      0.6
36      px      0.7
37      px      0.8
38      px      0.9
39      px      1
40      px      1.1
41      px      1.2
42      px      1.3
43      px      1.4
44      px      1.5
45      px      1.6
46      px      1.7
47      px      1.8
48      px      1.9
49      px      2
50      px      2.1
51      px      2.2
```

```
52        px        2.3
53        px        2.4
54        px        2.5
55        px        2.6
56        px        2.7
57        px        2.8
58        px        2.9

c
c    Material Cards
c
mode n a e # p
c SDEF SUR=7 NRM=-1 ERG=0.025E-6 dir=d1
sdef pos=1.45 2.5 -1.1 vec= 0 0 1 dir=d1 erg=2.5e-8
si1 -1 0.9 1
sp1  0 0.95 0.05
sb1  0 0 1
c sb1 -21 2
imp:n  1 1 1 1 0
imp:a  1 1 1 1 0
imp:p  1 1 1 1 0
imp:e  1 1 1 1 0
imp:#  1 1 1 1 0
fcl:n  1 1 0 0 0
c f361:t  2
c fs361 44 -10 -43 11
c e361    0.001 0.02 0.1 0.2 0.3 0.5 1 1.1 1.2 1.3 1.4 1.44 1.46
c         1.48 1.5 1.6 1.7 1.75 1.77 1.79
c         1.9 2 3 4 5 6
f371:#  27
fs371 44 -10 -43 11
e371    0.001 0.02 0.1 0.2 0.3 0.5 1 1.1 1.2 1.3 1.4 1.44 1.46
       1.48 1.5 1.6 1.7 1.75 1.77 1.79
       1.9 2 3 4 5 6
c f391:a  2
c fs391 44 -10 -43 11
c e391    0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
c         1.1 1.2 1.3 1.4 1.44 1.46 1.48 1.5 1.6 1.7 1.75 1.77 1.79
c         1.9 2 3 4 5 6
f401:a  27
fs401 44 -10 -43 11
e401    0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
       1.1 1.2 1.3 1.4 1.44 1.46 1.48 1.5 1.6 1.7 1.75 1.77 1.79
       1.9 2 3 4 5 6
```

```
phys:n  6j 2
phys:p  5j
c phys:a  100  3j 1 j j 3j 0.9
cut:n  2j 0 0
cut:p  2j 0 0
cut:a  j 0
cut:#  j 0
F26:a  2
e26 0  .05 .1 .15 .189 .193 .25 .3 .35 .4 .45 .5 .55 .57 0.574  0.6 1 2 3 4 5 6
F16:#  2
e16 0  .05 .1 .15 .189 .193 .25 .3 .35 .4 .45 .5 .55 .57 0.574  0.6 1 2 3 4 5 6
f41:a  2
fs41 44 -10 -43 11
e41   0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
      1.1 1.2 1.3 1.4 1.44 1.46 1.48 1.5 1.6 1.7 1.75 1.77 1.79
      1.9 2 3 4 5 6
f451:#  2
fs451 44 -10 -43 11
e451   0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
      1.1 1.2 1.3 1.4 1.44 1.46 1.48 1.5 1.6 1.7 1.75 1.77 1.79
      1.9 2 3 4 5 6
m1        5010.  -0.7488
          5011.  -0.0312
          6012.  -0.21758
          6013.  -0.00242
m2        14028.  -1
m3     26000  -0.66598        6000  -0.00052        24000  -0.13800
          28000  -0.15200        42000  -0.01460        14000  -0.00920
          25055  -0.01740        22000  -0.00230     $ SS
m4        7014.  -0.79
          8016.  -0.21
c mx1:n  model
c mx1:n  model
c mx2:n  model
nps 800000
```

END

LITHIUM FLUORIDE

```
c
c      MCNP Code for Lithium  Fluoride  Neutron Detector
c
c      Cell Cards
c
c
c
1      1  -2.64    1 -2 3 -4 5 -6
2      2  -2.32    2 -27 3  -4 5 -6
3      3  -8.03    #1 #2 21  -28 23  -24 25  -26
4      4 -0.001293  #1 #2 #3 -7
99     0           7

c
c    Surface Cards
c
1        pz  0.0
2        pz  0.003
3        px  0
4        px  3
5        py  0
6        py  5
7        so  10
21       pz  -0.10
23       px  -0.10
24       px   3.10
25       py  -0.10
26       py   5.1
27       pz   0.00302
28       pz   0.10302
10       py   2.45
11       py   2.55
100      py   1.45
110      py   1.55
200      py   3.45
210      py   3.55
300      py   0.45
310      py   0.55
400      py   4.45
410      py   4.55
30        px         0.1
31        px         0.2
32        px         0.3
33        px         0.4
```

```
34        px        0.5
35        px        0.6
36        px        0.7
37        px        0.8
38        px        0.9
39        px        1
40        px        1.1
41        px        1.2
42        px        1.3
43        px        1.4
44        px        1.5
45        px        1.6
46        px        1.7
47        px        1.8
48        px        1.9
49        px        2
50        px        2.1
51        px        2.2
52        px        2.3
53        px        2.4
54        px        2.5
55        px        2.6
56        px        2.7
57        px        2.8
58        px        2.9

c
c    Material Cards
c
mode n a e t p
c SDEF SUR=7 NRM=-1 ERG=0.025E-6  dir=d1
sdef pos=1.45  2.5 -1.1 vec= 0 0 1 dir=d1  erg=2.5e-8
si1  -1 0.9 1
sp1   0 0.95 0.05
sb1   0 0 1
c sb1 -21 2
imp:n  1 1 1 1 0
imp:a  1 1 1 1 0
imp:p  1 1 1 1 0
imp:e  1 1 1 1 0
imp:t  1 1 1 1 0
fcl:n  1 1 1 0 0
f361:t  2
fs361 44 -10 -43 11
```

```
e361   0.001  0.02 0.1 0.2  0.3 0.5  1 1.1  1.2 1.3  1.4 1.44  1.46
       1.48 1.5  1.6 1.7  1.75 1.77  1.79
       1.9 2 3 4 5 6
f371:t  27
fs371 44 -10 -43 11
e371   0.001  0.02 0.1 0.2  0.3 0.5  1 1.1  1.2 1.3  1.4 1.44  1.46
       1.48 1.5  1.6 1.7  1.75 1.77  1.79
       1.9 2 3 4 5 6
f391:a  2
fs391 44 -10 -43 11
e391   0.1 0.2 0.3  0.4 0.5  0.6 0.7  0.8 0.9  1
       1.1 1.2 1.3  1.4 1.44  1.46 1.48  1.5 1.6 1.7  1.75 1.77  1.79
       1.9 2 3 4 5 6
f401:a  27
fs401 44 -10 -43 11
e401   0.1 0.2 0.3  0.4 0.5  0.6 0.7  0.8 0.9  1
       1.1 1.2 1.3  1.4 1.44  1.46 1.48  1.5 1.6 1.7  1.75 1.77  1.79
       1.9 2 3 4 5 6
phys:n  6j 1
phys:p  5j
c phys:a  100  3j 1 j j 3j 0.9
cut:a j 0
cut:t j 0
F26:a  2
e26 0 .05 .1 .15 .189 .193 .25 .3 .35 .4 .45 .5 .55 .57 0.574  0.6 1 2 3 4 5 6
F16:t  2
e16 0 .05 .1 .15 .189 .193 .25 .3 .35 .4 .45 .5 .55 .57 0.574  0.6 1 2 3 4 5 6
m1        3006.  -0.23763
          3007.  -0.02937
          9019.  -0.733
m2        14028.  -1
m3     26000  -0.66598        6000  -0.00052        24000  -0.13800
          28000  -0.15200        42000  -0.01460        14000  -0.00920
          25055  -0.01740        22000  -0.00230     $ SS
m4        7014.  -0.79
          8016.  -0.21
c mx1:n  model
c mx1:n  model
c mx2:n  model
nps  3e6
```

END

APPENDIX C

---

exampleMS.cc


```
//
// Using the ExampleN06 as the reference, NaI Crystal has been modeled.
// For more information, study the basics of optical photon simulation in ExampleN06.
//
#include "G4RunManager.hh"
#include "G4UImanager.hh"
#include "ExMSPhysicsList.hh"
#include "ExMSPrimaryGeneratorAction.hh"
#include "ExMSDetectorConstruction.hh"
#include "ExMSRunAction.hh"
#include "ExMSStackingAction.hh"
#include "ExMSEventAction.hh"
#include "ExMSSteppingAction.hh"
#include "ExMSSteppingVerbose.hh"
#ifdef G4VIS_USE
#include "G4VisExecutive.hh"
#endif
#ifdef G4UI_USE
#include "G4UIExecutive.hh"
#endif
int main(int argc,char** argv)
{

  CLHEP::HepRandom::setTheEngine(new CLHEP::RanecuEngine);
  G4VSteppingVerbose* verbosity = new ExMSSteppingVerbose;
  G4VSteppingVerbose::SetInstance(verbosity);
  G4RunManager* runManager = new G4RunManager;
  G4VUserPhysicsList* physics = new ExMSPhysicsList;
  runManager-> SetUserInitialization(physics);
  G4VUserPrimaryGeneratorAction* gen_action = new ExMSPrimaryGeneratorAction;
  runManager->SetUserAction(gen_action);
  G4VUserDetectorConstruction* detector = new ExMSDetectorConstruction;
  runManager-> SetUserInitialization(detector);

#ifdef G4VIS_USE
  // visualization manager
```

```
  //
  G4VisManager* visManager = new G4VisExecutive;
  visManager->Initialize();
#endif
    ExMSEventAction*  event_action = new ExMSEventAction();
    runManager->SetUserAction(event_action);
   G4UserRunAction*  run_action = new ExMSRunAction(event_action);
   runManager->SetUserAction(new  ExMSSteppingAction(event_action));
   G4UserStackingAction*  stacking_action  = new ExMSStackingAction;
   runManager->SetUserAction(stacking_action);
   runManager->Initialize();
   G4UImanager*  UImanager = G4UImanager::GetUIpointer();

  if (argc==1)   // Define UI session for interactive mode
    {
#ifdef G4UI_USE
      G4UIExecutive  * ui = new G4UIExecutive(argc,argv);
#ifdef G4VIS_USE
        UImanager->ApplyCommand("/vis/sceneHandler/create");
        UImanager->ApplyCommand("/control/execute  vis.mac");
#endif
      ui->SessionStart();
      delete ui;
#endif
    }
  else         // Batch mode
    {
      G4String  command = "/control/execute  ";
      G4String  fileName  = argv[1];
      UImanager->ApplyCommand(command+fileName);
    }
#ifdef G4VIS_USE
  delete visManager;
#endif
  delete runManager;
  delete verbosity;
  return 0;
}
```

END

```
#ifndef exMSdetectorconstruction_hh
#define exMSdetectorconstruction_hh  1
#include "G4Material.hh"
#include "globals.hh"
#include "G4VUserDetectorConstruction.hh"
#include "G4ThreeVector.hh"
class G4Box;
class G4Tubs;
class G4Sphere;
class G4LogicalVolume;
class G4VPhysicalVolume;
class G4Material;
#include "G4ios.hh"
class ExMSDetectorConstruction  : public  G4VUserDetectorConstruction
{
public:
   ExMSDetectorConstruction();
   ~ExMSDetectorConstruction();
   static ExMSDetectorConstruction*  Instance();
   inline  G4ThreeVector GetWorldDimensions()  { return volume;  }
   inline  G4double  GetWorldSize()  {return fWorldSize;}
public:
   G4VPhysicalVolume*  Construct();
   const G4VPhysicalVolume*  GetNaI() {return detector_phys;}
   G4ThreeVector GetDimensionsNaI()  { return cathode;}
   const G4VPhysicalVolume*  GetGreaseC() {return op_grease_phy;}
   const G4VPhysicalVolume*  GetWorld()  {return volume_phys;}
protected:
   void sendetector();
private:
   static ExMSDetectorConstruction*  fgInstance;
   void ColorVolume();
   G4LogicalVolume*  volume_log;
   G4LogicalVolume*  detector_log;
   G4LogicalVolume*  ss_casing_log;
   G4LogicalVolume*  uranium_log;
   G4LogicalVolume*  op_quartzc_log;
   G4LogicalVolume*  op_quartz_log;
   G4LogicalVolume*  op_greasec_log;
   G4LogicalVolume*  op_grease_log;
   G4VPhysicalVolume*  volume_phys;
```

```
    G4VPhysicalVolume*  detector_phys;
    G4VPhysicalVolume*  ss_casing_phys;
    G4VPhysicalVolume*  uranium_phys;
    G4ThreeVector volume, detector, cathode,  op_windowz,  op_windowxy,
    op_grease, op_wincover,  op_greasecover;
    G4double  fWorldSize;
    G4VPhysicalVolume*  op_quartzc_phy;
    G4VPhysicalVolume*  op_quartz_phy;
    G4VPhysicalVolume*  op_greasec_phy;
    G4VPhysicalVolume*  op_grease_phy;
};
#endif
```

END

ExMSEventAction.hh

```
#ifndef ExMSEventAction_h
#define ExMSEventAction_h  1
#include "G4UserEventAction.hh"
#include "globals.hh"
#include "G4ThreeVector.hh"
#include <list>
#include <map>
#include <vector>
#include <queue>
#include <iostream>
#include <fstream>
#include <istream>
typedef std::vector<G4ThreeVector>  G4PVPosVector;
typedef std::vector<G4String>  G4PVString;
typedef std::map<G4int, std::pair<std::pair<G4ThreeVector, G4int>, G4int> >
G4PVPosEvent;
typedef std::queue<std::pair<G4ThreeVector,  G4double> > G4PVQue;
typedef std::queue<G4double  > G4PVQue1;
typedef std::vector<G4int>  G4PVCount;
typedef std::vector<G4int>  PhotoE;
class ExMSSteppingAction;
class ExMSEventMessenger;
```

```
class ExMSEventAction : public G4UserEventAction
{
 public:
   ExMSEventAction();
   virtual ~ExMSEventAction();
   static ExMSEventAction* Instance();
   virtual void BeginOfEventAction(const G4Event* event);
   virtual void EndOfEventAction(const G4Event* event);
   void Reset();
   void PrintNumbers();
   void RepeatCheck();
   G4double GetEnergySum() const { return fEnergySum; }
   G4double GetEnergy2Sum() const { return fEnergy2Sum; }
   void SetPrintModulo(G4int val)  {fPrintModulo = val;};
   void AddDecayChain(G4String val) {fDecayChain += val;};
   void SortingforPosX();
   void SortingforNegX();
   void SortingforPosY();
   void SortingforNegY();
   void SortingforPosZ();
   void SortingforNegZ();
   void InitializeVariable();
   void InitializePrinting();
   void InitializeComments();
   void printingoutput();
   void PhotoEff(G4bool PhotC, G4int Event)
   {
      photoeff.push_back(PhotC);
      eventn1.push_back(Event);
      logPE=PhotC;
      c2PE=Event;
   };
   void AddStep2(G4ThreeVector PhotPos, G4int check, G4int eventno)
   {
      totalpos = PhotPos;
      if (check == -2) {
         comptpos.push_back(PhotPos);
         comptevent.push_back(eventno);
      }
      else
      {
       photopos.push_back(PhotPos);
       photoevent.push_back(eventno);
      }  }
```

```
  void AddStep3(G4ThreeVector Data_Phot)
  {
     ThirdDisEvent.push_back(Data_Phot);
  }
  void AddStep(G4ThreeVector Pos, G4int count)
  {
     G4double p1 = (round(Pos.x()*1000));
     G4double q1 = (round(Pos.y()*1000));
     G4double r1 = (round(Pos.z()*1000));
     posVectorN.push_back(G4ThreeVector(p1,q1,r1));
     G4double X = (round(Pos.x()*1000))/10000;
     G4double Y = (round(Pos.y()*1000))/10000;
     G4double Z = (round(Pos.z()*1000))/10000;
     post_tracked.push_back(G4ThreeVector(X,Y,Z));
     preTracked.push_back(Pos);
     Dphotoncount.push_back(count);
  };
private:
  static ExMSEventAction* fgInstance;
   G4int evexno;
   G4bool logPE;
   G4double counteve;
  G4ThreeVector CurrentVector;
  G4double iter;
  G4double ixn, ix, iyn, iy, izn, iz;
  G4double OnPosX[41][11];
  G4double OnNegX[41][11];
  G4double OnPosY[41][6];
  G4double OnNegY[41][6];
  G4double OnPosZ[11][6];
  G4double OnNegZ[11][6];
  G4double TOnPosX[42][21];
  G4double TOnNegX[42][21];
  G4double TOnPosY[42][11];
  G4double TOnNegY[42][11];
  G4double TOnPosZ[21][11];
  G4double TOnNegZ[21][11];
  G4int noj, nok;
  G4int c1PE, c2PE;
  G4double strx , strxn, stry, stryn, strz, strzn, total_Accum;
  G4int evtNb;
  PhotoE photoevent, comptevent;
  G4PVCount Dphotoncount;
```

```
   G4PVPosEvent posevent;
   G4PVPosVector pre_tracked, posttracked, preTracked, post_tracked, posvector, prevector, left_outs, posVectorN;
   G4PVPosVector vectorXn, vectorYn, vectorZn, vectorX, vectorY, vectorZ, photopos, comptpos;
   G4ThreeVector totalpos;
   G4PVQue1 repQute;
   G4PVQue  repQuT;
   G4PVPosVector FirDisEvent, SecDisEvent, ThirdDisEvent, FourDisEvent;
   G4PVString PreN, PostN;
   G4int totalCounted, totaloptibeingdone;
   G4int   totatsame, counterforposition;
   G4int    fPrintModulo;
   G4double  fEnergySum, total_Accounted, alongsurfaces;
   G4double  fEnergy2Sum;
   G4double alongsurfacesX, alongsurfacesXn, alongsurfacesY, alongsurfacesYn, alongsurfacesZ, alongsurfacesZn;
   G4double total_AccountedX, total_AccountedXn, total_AccountedY, total_AccountedYn, total_AccountedZ;
   G4double  avgaccX, avgaccXn, avgaccY, avgaccYn, avgaccZ, avgaccZn;
   G4double  tavgX, tavgXn, tavgY, tavgYn, tavgZ, tavgZn;
   G4double  bavgX, bavgXn, bavgY, bavgYn, bavgZ, bavgZn;
   G4double  sc1, sc2, sc3, sc4, sc5, sc6, sc7, sc8, sc9, sc10, sc11, sc12, sc13, sc14, sc15, sc16, sc17, sc18, sc19;
   G4int  ph1, ph2;
   G4double  ph3, ph4;
   G4double  p;
   G4double  q;
   G4int  t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11,t12;
   G4double  total1a, total2a, total3a, total4a, total5a, total6a;
   G4double  total1b, total2b, total3b, total4b, total5b, total6b;
   G4double  total1, total2, total3, total4, total5, total6;
   G4double  avgX, avgXn, avgY, avgYn, avgZ, avgZn;
   G4double  total_AccountedZn;
   G4String        fDecayChain;
   PhotoE photoeff, eventn1;
   std::ofstream  myfile;
   ExMSEventMessenger*  fEventMessenger;
};
#endif
```

END

ExMSPhysicsList.hh


```
#ifndef ExMSPhysicsList_h
#define ExMSPhysicsList_h 1
#include "globals.hh"
#include "G4VUserPhysicsList.hh"
class G4Cerenkov;
class G4Scintillation;
class G4OpAbsorption;
class G4OpRayleigh;
class G4OpMieHG;
class G4OpBoundaryProcess;
class G4PhotoElectricEffect;
class G4ComptonScattering;
class G4GammaConversion;
class G4RayleighScattering;
class G4GammaConversionToMuons;
class G4eIonisation;
class G4eBremssrathlung;
class G4eMultipleScattering;
class G4eplusAnnihilation;
class G4AnnihiToMuPair;
class G4eeToHadrons;
class G4MuIonisation;
class G4MuBremsstrahlung;
class G4MuPairProduction;
class G4MuMultipleScattering;
class G4hIonisation;
class G4ionIonisation;
class G4hhIonisation;
class G4mplIonisation;
class G4hMultipleScattering;
class G4hBremsstrahlung;
class G4hPairProduction;
class G4CoulombScattering;
class G4ScreenedNuclearRecoil;
class G4SynchrotronRadiation;
class G4TransitionRadiation;
class G4PAIModel;
class G4PAIPhotModel;
class G4BraggIonGasModel;
```

```
class G4BetheBlochIonGasModel;
class G4WentzelVIModel;
class G4UrbanMscModel;
class G4LivermorePhotoElectricModel;
class G4LivermorePolarizedPhotoElectricModel;
class G4LivermoreComptonModel;
class G4LowEPComptonModel;
class G4LivermorePolarizedComptonModel;
class G4LivermoreRayleighModel;
class G4LivermorePolarizedRayleighModel;
class G4LivermoreGammaConversionModel;
class G4LivermoreNuclearGammaConversionModel;
class G4LivermoreGammaConversionModelRC;
class G4LivermorePolarizedGammaConversionModel;
class G4LivermoreBremsstrahlungModel;
class G4LivermoreIonisationModel;
class ExMSPhysicsList : public G4VUserPhysicsList
{
 public:
   ExMSPhysicsList();
  ~ExMSPhysicsList();
 public:
   void ConstructParticle();
   void ConstructProcess();
   void SetCuts();
   void ConstructBosons();
   void ConstructLeptons();
   void ConstructMesons();
   void ConstructBaryons();
   void ConstructGeneral();
   void ConstructEM();
   void ConstructOp();
   void SetVerbose(G4int);
   void SetNbOfPhotonsCerenkov(G4int);
 private:
   G4Cerenkov*          theCerenkovProcess;
   G4Scintillation*     theScintillationProcess;
   G4OpAbsorption*      theAbsorptionProcess;
   G4OpRayleigh*        theRayleighScatteringProcess;
   G4OpMieHG*           theMieHGScatteringProcess;
   G4OpBoundaryProcess* theBoundaryProcess;
   G4PhotoElectricEffect* thePhotoelectriceffect;
   G4ComptonScattering*  theComptonscattering;
   G4GammaConversion*  theGammaconversion;
```

```
  G4RayleighScattering*  theRayleighscattering;
  G4GammaConversionToMuons*  theGammaConversiontomuons;
  G4eIonisation*  theeIonisation;
  G4eBremssrathlung*  theeBremsslung;
  G4eMultipleScattering*  theeMultipleScatt;
  G4eplusAnnihilation*  theeplusAnni;
  G4AnnihiToMuPair*  theAnnihiToMuPair;
  G4eeToHadrons* theetohadrons;
  G4MuIonisation*  theMuIonisation;
  G4MuBremsstrahlung*  theMuBremss;
  G4MuPairProduction*  themupairprod;
  G4MuMultipleScattering*  themultiplescatt;
  G4hIonisation*  thehIonisation;
  G4ionIonisation*  theionIonisation;
  G4hhIonisation*  thehhIonisation;
  G4mplIonisation*  themplIonisation;
  G4hMultipleScattering*  thehMultipleScattering;
  G4hBremsstrahlung*  thehBremsstrahlung;
  G4hPairProduction*  thehpairproduction;
  G4CoulombScattering*  theCoulombic;
  G4ScreenedNuclearRecoil*  theScrrened;
  G4SynchrotronRadiation*  theSynchroton;
  G4TransitionRadiation*  theTransition;
  G4PAIModel*  thePAIMOdel;
  G4PAIPhotModel*  thePAIPhotmodel;
  G4BraggIonGasModel*  theBraggIonGasModel;
  G4BetheBlochIonGasModel*  theBethe;
  G4WentzelVIModel*  theWentzelMo;
  G4UrbanMscModel*  theUrbanMo;
  G4LivermorePhotoElectricModel*  thLiverPhoto;
  G4LivermorePolarizedPhotoElectricModel*  theLiPPhoto;
  G4LivermoreComptonModel*  theLiCompton;
  G4LowEPComptonModel*  theLiLoCompton;
  G4LivermorePolarizedComptonModel*  theLivPCompton;
  G4LivermoreRayleighModel*  theLivRModel;
  G4LivermorePolarizedRayleighModel*  theLivGammaComversion;
  G4LivermoreGammaConversionModel*  theLivGammConvModel;
  G4LivermoreNuclearGammaConversionModel*  theLivNucle;
  G4LivermoreGammaConversionModelRC*  theLivGammaConv;
  G4LivermorePolarizedGammaConversionModel*  theLivPGammaCon;
  G4LivermoreBremsstrahlungModel*  theLivBrem;
  G4LivermoreIonisationModel*  theLivIoniModel;
};
```

```
#endif /* ExMSPhysicsList_h  */
```

END

ExMSPrimaryGeneratorAction.hh

```
#ifndef ExMSPrimaryGeneratorAction_h
#define ExMSPrimaryGeneratorAction_h  1
#include "G4VUserPrimaryGeneratorAction.hh"
#include "globals.hh"
class G4ParticleGun;
class G4Event;
class ExMSPrimaryGeneratorAction  : public  G4VUserPrimaryGeneratorAction
{
 public:
   ExMSPrimaryGeneratorAction();
  ~ExMSPrimaryGeneratorAction();
 public:
   void GeneratePrimaries(G4Event*);
   void SetOptPhotonPolar();
   void SetOptPhotonPolar(G4double);
   G4double  xt, yt, zt, testenergy;
   G4double  Energy;
 private:
   G4ParticleGun*  particleGun;
};
#endif /*ExMSPrimaryGeneratorAction_h*/
```

END

ExMSRunAction.hh

```
#ifndef ExMSRunAction_h
#define ExMSRunAction_h  1
#include "ExMSEventAction.hh"
#include "globals.hh"
```

```
#include "G4UserRunAction.hh"
class G4Timer;
class G4Run;
class ExMSEventAction;
class ExMSDetectorConstruction;
class ExMSRunAction : public G4UserRunAction
{
 public:
   ExMSRunAction(ExMSEventAction* );
  ~ExMSRunAction();
 public:
   void BeginOfRunAction(const G4Run* aRun);
   void EndOfRunAction(const G4Run* aRun);
 private:
   G4Timer* timer;
   ExMSEventAction* runactionp;
};
#endif /*ExMSRunAction_h*/
```

END

ExMSStackingAction.hh

```
#ifndef ExMSStackingAction_H
#define ExMSStackingAction_H 1
#include "globals.hh"
#include "G4UserStackingAction.hh"
#include "ExMSDetectorConstruction.hh"
#include "ExMSSteppingAction.hh"
class ExMSStackingAction : public G4UserStackingAction
{
public:
   ExMSStackingAction();
   ~ExMSStackingAction();
public:
   G4ClassificationOfNewTrack ClassifyNewTrack(const G4Track* aTrack);
   void NewStage();
   void PrepareNewEvent();
private:
   ExMSSteppingAction* stepping;
```

```
   ExMSDetectorConstruction*  detectorstack;
   G4int  gammaCounter, cerenkovcounter;
   G4int  gammaconv, photoele, comptons, totalelectrons, totalpart2;
};
#endif
```

END

ExMSSteppingAction.hh

```
#ifndef ExMSSteppingAction_h
#define ExMSSteppingAction_h  1
#include "ExMSDetectorConstruction.hh"
#include "ExMSEventAction.hh"
#include "ExMSPrimaryGeneratorAction.hh"
class ExMSDetectorConstruction;
class ExMSEventAction;
#include "G4Timer.hh"
#include "G4UserSteppingAction.hh"
#include "globals.hh"
#include <set>
#include <ar.h>
#include <map>
#include <vector>
#include <deque>
#include "G4ThreeVector.hh"
#include "G4Material.hh"
#include "G4Track.hh"
#include "G4UserTrackingAction.hh"
#include "G4TrackingManager.hh"
#include "G4Trajectory.hh"
class G4VSteppingVerbose;
#include "G4SmoothTrajectory.hh"
#include "G4RichTrajectory.hh"
#include "G4ios.hh"
using namespace std;
#include <fstream>
#include <istream>
#include "G4OpBoundaryProcess.hh"
class G4VPhysicalVolume;
```

```
typedef
std::map<G4VPhysicalVolume*,std::pair<std::pair<G4double,G4double>,unsigned>    >
G4PVmap;
typedef std::map<G4VPhysicalVolume*,  unsigned>  G4PVflux;
typedef std::set<G4VPhysicalVolume*>  G4PVset;
typedef std::set<G4VPhysicalVolume*>  G4PVBVolume;
typedef std::vector<const  G4Track*> G4PVTrack;
typedef std::set<const  G4Track*> G4PSTrack;
typedef  std::map<G4int,  std::pair<std::pair<G4ThreeVector,  G4int>,  G4int  >  >
G4PVPosTrack;
typedef std::map<G4VPhysicalVolume*,  G4int>  G4PVTrackID;
typedef std::vector<G4double  >  G4EnergyVec;
class ExMSSteppingAction  : public  G4UserSteppingAction
{
 public:
   ExMSSteppingAction(ExMSEventAction*  );
   virtual ~ExMSSteppingAction();
   static ExMSSteppingAction*  Instance();
   virtual  void UserSteppingAction(const  G4Step*);
   void Reset();
   void countprint();
   void resetvalues();
   void AddVolume(G4VPhysicalVolume*  volume1)  { fVolumes.insert(volume1);  }
   inline  const G4int&  counter() const {return fscintillationcounter;}
   inline  const G4PVset& GetVolumes() const { return fVolumes; }
   inline  const G4PVmap& GetEnergyMap() const { return fEnergies; }
   inline  const G4PVflux& GetFluxes()  const { return fFlux; }
   inline  const G4PVBVolume&  GetBVolumes() const { return fBVolume;}
   inline  const G4PVTrackID& GetTrackID() const { return fTrackID;}
   inline  const G4PVPosTrack& GetPosTrack() const { return fPosTrack; }
   G4int  LastTrackID, CheckID, Exitque,  LastTrackIDx, CheckIDx, Exitquex;
 private:
   static ExMSSteppingAction*  fgInstance;
   ExMSDetectorConstruction*  detector;
   ExMSEventAction*  eventaction;
   G4ThreeVector position;
   G4PVBVolume  fBVolume;
   G4PVset fVolumes;
   G4PVmap fEnergies;
   G4PVflux fFlux;
   G4PVTrackID fTrackID;
   G4PVPosTrack fPosTrack;
   G4PSTrack secondaries_to_track;
```

```
   G4int fscintillationcounter, fcerenkovcounter, feventnumber, total_optical_photons,
detector_volume_optical_photon;
   G4int fscintillationcounterx, fcerenkovcounterx, absrpcount, photondetecount,
backreflcount;
   std::vector<G4int > trackIDno;
   std::vector<G4int > parentIDno;
   G4int EnergyTtoal, DeporTotal, ExtraEnergy;
   G4int PhotCount;
   G4int DPhotCount;
   G4int OpAbsCount;
   G4int ExitScinCount, backcount;
   G4int electroncounts, withincounts, onestepcounts, countingtotal;
   G4int trakID, prantId, dtrackId, dprantD;
   G4int totalop;
   G4int ExtraP;
   G4int ElectronCount;
   G4int Extracheck, Extracheck2;
   G4int CountingPhotons, CountingDetectedPhotons;
   G4int PhotoEffCounter, PEC1, PEC2, PEC3;
   G4long coun1, coun2, coun3, coun4, coun5, coun6, coun7, coun8;
   G4double minEn, maxEn;
   G4EnergyVec EnergyVec;
   G4int eTotalInternalReflection;
   G4int eLambertianReflection;
   G4int eLobeReflection;
   G4int eSpikeReflection;
   G4int eTER;
   G4Timer* timerboss;
   G4int temp1, temp2, temp3;
   G4int Counting;
   G4double avgEnergy;
   G4int OpPhotonCount;
   G4bool setconv;
    G4double stter, photoset;
   G4int counterforcuttingcompton;
   G4int eventNumber;
   G4int comptonscacount, photocounter, comptset;
   G4int countotalphoton;
   G4int temp4, temp5;
   G4bool fOneStepPrimaries;
   G4double fEnergy;
   G4long temp6, temp7, temp8, temp9, temp11, temp12, temp80;
   G4int temp20, temp22;
   G4double temp10;
```

```
   G4int temp40, temp41;
   G4ThreeVector photoposition;
   G4int temp21;
   ofstream myfile;
   G4long temp81;
   G4double energyto;
   G4int run_j1, run_j2, run_j3, run_j4, run_j5, run_j6, run_j7, run_j8, run_j9, run_j10,
run_j11, run_j12;
   G4int run_j13, run_j14, run_j15, run_j16, run_j17, run_j18, run_j19, run_j20, run_j21,
run_j22;
   G4OpBoundaryProcessStatus fExpectedNextStatus;
};
#endif
```

END

ExMSSteppingMessenger.hh

```
#ifndef ExMSSteppingMessenger_h
#define ExMSSteppingMessenger_h 1
#include "G4UImessenger.hh"
#include "globals.hh"
class ExMSSteppingAction;
class G4UIcmdWithABool;
class ExMSSteppingMessenger: public G4UImessenger
{
 public:
   ExMSSteppingMessenger(ExMSSteppingAction*);
   virtual ~ExMSSteppingMessenger();
   virtual void SetNewValue(G4UIcommand*, G4String);
private:
   ExMSSteppingAction*        fStepping;
   G4UIcmdWithABool*   fOneStepPrimariesCmd;
};
#endif
```

END

ExMSSteppingVerbose.hh

```
class ExMSSteppingVerbose;
#include <set>
#include <map>
#include <vector>
#include <deque>
#include "G4ThreeVector.hh"
#ifndef ExMSSteppingVerbose_h
#define ExMSSteppingVerbose_h 1
#include "G4SteppingVerbose.hh"
struct TrackingHistory
{
   TrackingHistory(){}
   TrackingHistory( G4double step_, G4ThreeVector pos_, G4String namevol_, G4String
processn_, G4double energyt_, G4double energyd_)
     : stepno(step_), position(pos_), volumename(namevol_), processna_(processn_),
       energytotal(energyt_), energydep(energyd_)
  { }
  G4double stepno;
  G4ThreeVector position;
  G4String volumename;
  G4String processna_;
  G4double energytotal;
  G4double energydep;
};
class ExMSSteppingVerbose : public G4SteppingVerbose
{
 public:
   ExMSSteppingVerbose();
  ~ExMSSteppingVerbose();
   void StepInfo();
   void TrackingStarted();
private:
   G4long countsno;
   std::map<G4double, TrackingHistory > tracking_history;
};
#endif
```

END

---
exampleMS.in

/run/beamOn 10000000

END

CMakeLists.txt


```
# Setup the project
cmake_minimum_required(VERSION 2.6 FATAL_ERROR)
project(Fianl_Edits)
option(WITH_GEANT4_UIVIS "Build example with Geant4 UI and Vis drivers" ON)
if(WITH_GEANT4_UIVIS)
  find_package(Geant4 REQUIRED ui_all vis_all)
else()
  find_package(Geant4 REQUIRED)
endif()
include(${Geant4_USE_FILE})
include_directories(${PROJECT_SOURCE_DIR}/include
                    ${Geant4_INCLUDE_DIR})
file(GLOB sources ${PROJECT_SOURCE_DIR}/src/*.cc)
file(GLOB headers ${PROJECT_SOURCE_DIR}/include/*.hh)
add_executable(exampleMS exampleMS.cc ${sources} ${headers})
target_link_libraries(exampleMS ${Geant4_LIBRARIES} )
set(MS_SCRIPTS
    exampleMS.in exampleMS.out vis.mac
  )
foreach(_script ${MS_SCRIPTS})
  configure_file(
    ${PROJECT_SOURCE_DIR}/${_script}
    ${PROJECT_BINARY_DIR}/${_script}
    COPYONLY
    )
endforeach()
add_custom_target(MS DEPENDS exampleMS)
install(TARGETS exampleMS DESTINATION bin)
```

END

---

vis.mac

```
# Use this open statement to create an OpenGL view:
/vis/open OGL 600x600-0+0
/vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
/vis/viewer/zoom 20
/vis/ogl/set/displayListLimit 10000000
/vis/set/textColour green
/vis/set/textLayout right
/vis/scene/add/text2D 0.5 -.7 24 ! ! NaI_Scintillation
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
/vis/scene/endOfEventAction accumulate -1
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
```

END

ExMSDetectorConstruction.cc

```
#include "ExMSDetectorConstruction.hh"
#include "ExMSSteppingAction.hh"
#include "G4SDManager.hh"
#include "G4RunManager.hh"
#include "G4Box.hh"
#include "G4Tubs.hh"
#include "G4Orb.hh"
#include "G4Sphere.hh"
#include "G4LogicalVolume.hh"
```

```
#include "G4VPhysicalVolume.hh"
#include "G4Material.hh"
#include "G4NistManager.hh"
#include "G4PVPlacement.hh"
#include "G4PVReplica.hh"
#include "G4VisAttributes.hh"
#include "G4Colour.hh"
#include "G4UnitsTable.hh"
#include "G4SystemOfUnits.hh"
#include "G4Material.hh"
#include "G4UserLimits.hh"
#include "G4Element.hh"
#include "G4LogicalBorderSurface.hh"
#include "G4LogicalSkinSurface.hh"
#include "G4OpticalSurface.hh"
ExMSDetectorConstruction*  ExMSDetectorConstruction::fgInstance  = 0;
ExMSDetectorConstruction*  ExMSDetectorConstruction::Instance()
{
   G4cout<<"Here  I am in Detector"<<G4endl;
   return fgInstance;
}
ExMSDetectorConstruction::ExMSDetectorConstruction()
   :  volume_log(0),  detector_log(0), ss_casing_log(0), uranium_log(0),
      volume_phys(0),  detector_phys(0), ss_casing_phys(0), uranium_phys(0)
{
   G4cout<<"Here  I am in Detector1"<<G4endl;
   fgInstance = this;
   G4double  expHall_x = 2.0*m;
   G4double  expHall_y = 2.0*m;
   G4double  expHall_z = 2.0*m;
   volume  = G4ThreeVector(expHall_x,  expHall_y,  expHall_z);
   G4double  detector_x = 5.08*cm;    // 2 inch -- 5.08 cm .18cm ss
   G4double  detector_y = 10.16*cm;     // 4 inch  -- 10.16 cm .18cm ss
   G4double  detector_z = 40.64*cm;   // 16 inch  -- 40.64  cm .09cm ss
   detector = G4ThreeVector(detector_x, detector_y, detector_z);
   G4double  pmt_cathode_x = 5.2832*cm;
   G4double  pmt_cathode_y = 10.3632*cm;
   G4double   pmt_cathode_z  =  40.8432*cm;  //  1.7958*cm;       // 0.04  inch  aluminium
housing  one side
   cathode = G4ThreeVector(pmt_cathode_x,pmt_cathode_y,pmt_cathode_z);
   fWorldSize   = 2*m;
}
ExMSDetectorConstruction::~ExMSDetectorConstruction()
{
```

```
   G4cout<<"Here  I am in DetectorEnd"<<G4endl;
   fgInstance = 0;
}
#include "G4Scintillation.hh"
#include "G4MaterialPropertiesTable.hh"
G4VPhysicalVolume*  ExMSDetectorConstruction::Construct()
{
   G4cout<<"Here  I am in Detector2"<<G4endl;
   G4NistManager*  nist = G4NistManager::Instance();
   G4Material*  Air = nist->FindOrBuildMaterial("G4_AIR");
   G4Element*  Chromium  = nist->FindOrBuildElement(24);
   G4Element*  Carbon = nist->FindOrBuildElement(6);
   G4Element*  Iron = nist->FindOrBuildElement(26);
   G4Material*  SSMaterial  = new G4Material("SS",8*g/cm3,3);
   SSMaterial->AddElement(Chromium,0.1);
   SSMaterial->AddElement(Carbon,0.01);
   SSMaterial->AddElement(Iron,0.89);
   G4Element*  elNa = nist->FindOrBuildElement(11);
   G4Element*  elI = nist->FindOrBuildElement(53);
   G4Material*  NaI =  new G4Material("NaI",  3.67*g/cm3,  2);
   NaI->AddElement(elNa,  0.5);
   NaI->AddElement(elI,  0.5);
   const G4int  NUMENTRIES = 46;
   G4double   rind[NUMENTRIES]={1.85,   1.85,1.85,   1.85,1.85,   1.85,1.85,   1.85,1.85,
1.85,
      1.85,  1.85,1.85,  1.85,1.85,  1.85,1.85,  1.85,1.85,  1.85,
      1.85,  1.85,1.85,  1.85,1.85,  1.85,1.85,  1.85,1.85,  1.85,
      1.85,  1.85,1.85,  1.85,1.85,  1.85,1.85,  1.85,1.85,  1.85,
      1.85,  1.85,1.85,  1.85,1.85,  1.85};

   G4double  abs[NUMENTRIES]={
      2*m,  2*m,  2*m,  2*m,  2*m,  2*m,  2*m,  2*m,  2*m,  2*m,  2*m,  2*m,  2*m,  2*m,  2*m,
2*m,  2*m,
      2*m,  2*m,  2*m,  2*m,  2*m,  2*m,  2*m,  2*m,  2*m,  2*m,  2*m,  2*m,  2*m,  2*m,  2*m,
2*m,  2*m,
      2*m,  2*m,  2*m,  2*m,  2*m,  2*m,  2*m,  2*m,  2*m,  2*m,  2*m};
   G4double  Scnt_PP[NUMENTRIES] = {
      2.25454545454545*eV,
      2.27522935779816*eV,
      2.2962962962963*eV,
      2.31775700934579*eV,
      2.33962264150943*eV,
      2.36190476190476*eV,
      2.38461538461538*eV,
```

```
    2.40776699029126*eV,
    2.43137254901961*eV,
    2.45544554455446*eV,
    2.48*eV,
    2.50505050505051*eV,
    2.53061224489796*eV,
    2.55670103092784*eV,
    2.58333333333333*eV,
    2.61052631578947*eV,
    2.63829787234043*eV,
    2.66666666666667*eV,
    2.69565217391304*eV,
    2.72527472527473*eV,
    2.75555555555556*eV,
    2.78651685393258*eV,
    2.81818181818182*eV,
    2.85057471264368*eV,
    2.88372093023256*eV,
    2.91764705882353*eV,
    2.95238095238095*eV,
    2.98795180722892*eV,
    3.02439024390244*eV,
    3.06172839506173*eV,
    3.1*eV,
    3.13924050632911*eV,
    3.17948717948718*eV,
    3.22077922077922*eV,
    3.26315789473684*eV,
    3.30666666666667*eV,
    3.35135135135135*eV,
    3.3972602739726*eV,
    3.44444444444444*eV,
    3.49295774647887*eV,
    3.54285714285714*eV,
    3.59420289855073*eV,
    3.64705882352941*eV,
    3.70149253731343*eV,
    3.75757575757576*eV,
    3.81538461538462*eV,
};
G4double Scnt_FAST[NUMENTRIES] =    {
    0,
    0,
    0.001,
```

```
    0.005,
    0.01,
    0.02,
    0.033,
    0.05,
    0.07,
    0.08,
    0.1,
    0.133,
    0.2,
    0.26,
    0.3,
    0.333,
    0.4,
    0.5,
    0.55,
    0.6,
    0.65,
    0.74,
    0.89,
    0.9,
    0.96,
    0.98,
    0.99,
    1,
    0.99,
    0.98,
    0.97,
    0.95,
    0.89,
    0.8,
    0.74,
    0.7,
    0.65,
    0.55,
    0.49,
    0.45,
    0.39,
    0.32,
    0.2667,
    0.2,
    0.07,
    0
};
```

```
   G4MaterialPropertiesTable*  Scnt_MPT = new G4MaterialPropertiesTable();
   Scnt_MPT->AddProperty("RINDEX",  Scnt_PP, rind, NUMENTRIES);
   Scnt_MPT->AddProperty("ABSLENGTH",  Scnt_PP, abs, NUMENTRIES);
   Scnt_MPT->AddProperty("FASTCOMPONENT",  Scnt_PP,  Scnt_FAST,
NUMENTRIES);
   Scnt_MPT->AddConstProperty("SCINTILLATIONYIELD",  40000./MeV);
   Scnt_MPT->AddConstProperty("RESOLUTIONSCALE",  2.0);
   Scnt_MPT->AddConstProperty("FASTTIMECONSTANT",   230.*ns);
   Scnt_MPT->AddConstProperty("YIELDRATIO",  1);
   NaI->SetMaterialPropertiesTable(Scnt_MPT);
   NaI->GetIonisation()->SetBirksConstant(0.0034*mm/MeV);
   G4double  airRefr[NUMENTRIES] = {2.57,  2.57,  2.57,  2.57,  2.57,  2.57,  2.57,  2.57,
2.57,  2.57,
      2.57,  2.57,  2.57,  2.57,  2.57,  2.57,  2.57,  2.57,  2.57,  2.57,
      2.57,  2.57,  2.57,  2.57,  2.57,  2.57,  2.57,  2.57,  2.57,  2.57,
      2.57,  2.57,  2.57,  2.57,  2.57,  2.57,  2.57,  2.57,  2.57,  2.57,
      2.57,  2.57,  2.57,  2.57,  2.57,  2.57};
   G4MaterialPropertiesTable*  AirTable  = new G4MaterialPropertiesTable();
   AirTable->AddProperty("RINDEX",  Scnt_PP, airRefr, NUMENTRIES);
   SSMaterial->SetMaterialPropertiesTable(AirTable);
   G4UserLimits*  limiter  = new G4UserLimits(1.*mm);
   G4Box*  volume_box
     = new G4Box("volume_box",volume.x(),volume.y(),volume.z());
   volume_log  = new G4LogicalVolume(volume_box,
                                     Air,"volume_log",  0,0,0);
   volume_phys  = new G4PVPlacement(0,G4ThreeVector(),
                              volume_log,"volume_phys",0,false,0,true);
   G4Box*  ss_casing = new
G4Box("SS_Casing",cathode.x()/2,cathode.y()/2,cathode.z()/2);
   ss_casing_log = new G4LogicalVolume(ss_casing,SSMaterial,  "ss-casing",0,0,0);
   G4Box*                    nai_box                    =                    new
G4Box("NaI_Box",detector.x()/2,detector.y()/2,detector.z()/2);
   detector_log = new G4LogicalVolume(nai_box,NaI,"NaI_Box",0,0,0);
   G4RotationMatrix  prot= G4RotationMatrix();
   prot.rotateZ(0*deg);
   prot.rotateX(0*deg);  //45
   prot.rotateY(0*deg);//-135
   G4ThreeVector pos =  G4ThreeVector(0*cm,0*cm,cathode.z()/2);
   G4Transform3D  transform = G4Transform3D(prot,  pos );
   ss_casing_log->SetUserLimits(limiter);
   detector_log->SetUserLimits(limiter);
   ss_casing_phys  =  new  G4PVPlacement(transform,  ss_casing_log,
"Detector_Casing",volume_log,  false, 1, true);
   pos = G4ThreeVector(0*cm,0*cm,((cathode.z()/2)-(detector.z()/2)));
```

```
   detector_phys = new G4PVPlacement(0,G4ThreeVector(0*cm,0*cm,
0.1016*cm),detector_log,"Detector_NaI",ss_casing_log,false,1, true);
   G4OpticalSurface* opnaiSurface = new G4OpticalSurface("NaISurface");
   opnaiSurface->SetType(dielectric_metal);
   opnaiSurface->SetFinish(polished);
   opnaiSurface->SetModel(glisur);
   new G4LogicalBorderSurface("NaISurface", detector_phys, ss_casing_phys,
opnaiSurface);
  G4double efficiency[NUMENTRIES] =
{0.0, 0.0, 0, 0, 0,0.0, 0.0, 0, 0, 0,
      0.0, 0.0, 0, 0, 0,0.0, 0.0, 0, 0, 0,
      0.0, 0.0, 0, 0, 0,0.0, 0.0, 0, 0, 0,
      0.0, 0.0, 0, 0, 0,0.0, 0.0, 0, 0, 0,
      0.0, 0.0, 0, 0, 0,0.};
   G4double reflectivity[NUMENTRIES] =
                {0.95, 0.95, 0.95, 0.95, 0.95,0.95, 0.95, 0.95, 0.95, 0.95,
                           0.95, 0.95, 0.95, 0.95, 0.95,0.95, 0.95, 0.95, 0.95, 0.95,
                           0.95, 0.95, 0.95, 0.95, 0.95,0.95, 0.95, 0.95, 0.95, 0.95,
                            0.95, 0.95, 0.95, 0.95, 0.95,0.95, 0.95, 0.95, 0.95, 0.95,
                            0.95, 0.95, 0.95, 0.95, 0.95,0.95};
   G4MaterialPropertiesTable* scintWrapProperty
   = new G4MaterialPropertiesTable();
   scintWrapProperty-
>AddProperty("REFLECTIVITY",Scnt_PP,reflectivity,NUMENTRIES);
   scintWrapProperty-
>AddProperty("EFFICIENCY",Scnt_PP,efficiency,NUMENTRIES);
   opnaiSurface->SetMaterialPropertiesTable(scintWrapProperty);
   G4OpticalSurface* PhotoSurface = new G4OpticalSurface("PhotoSurface");
   PhotoSurface->SetType(dielectric_metal);
   PhotoSurface->SetFinish(polished);
   PhotoSurface->SetModel(glisur);
   new G4LogicalBorderSurface("PhotoSurface", detector_phys, volume_phys,
PhotoSurface);
   G4double efficiency_P[NUMENTRIES] =
   {1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
      1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
      1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
      1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
      1, 1, 1, 1, 1, 1};
   G4double reflectivity_P[NUMENTRIES] = {0.0, 0.0, 0, 0, 0,0.0, 0.0, 0, 0, 0,
      0.0, 0.0, 0, 0, 0,0.0, 0.0, 0, 0, 0,
      0.0, 0.0, 0, 0, 0,0.0, 0.0, 0, 0, 0,
      0.0, 0.0, 0, 0, 0,0.0, 0.0, 0, 0, 0,
      0.0, 0.0, 0, 0, 0,0.   };
```

```
   G4MaterialPropertiesTable*  PhotoProperty
   = new G4MaterialPropertiesTable();
   PhotoProperty-
>AddProperty("REFLECTIVITY",Scnt_PP,reflectivity_P,NUMENTRIES);
   PhotoProperty-
>AddProperty("EFFICIENCY",Scnt_PP,efficiency_P,NUMENTRIES);
   PhotoSurface->SetMaterialPropertiesTable(PhotoProperty);
ColorVolume();
   ExMSSteppingAction*  steppingAction  = ExMSSteppingAction::Instance();
   steppingAction->AddVolume(detector_phys);
   return volume_phys;
}
void ExMSDetectorConstruction::ColorVolume()
{
   G4cout<<"Here  I am in Detector3"<<G4endl;
   G4VisAttributes*  det_vis = new G4VisAttributes(G4Color(0.,0.,0.,1.0));    /// Black
   det_vis->SetForceSolid(true);
   det_vis->SetVisibility(true);
   detector_log->SetVisAttributes(det_vis);
   G4VisAttributes*  pu_orb_vis  = new G4VisAttributes(G4Color(1.,0.,0.,1.0));
   pu_orb_vis->SetForceSolid(true);
   pu_orb_vis->SetVisibility(true);
   ss_casing_log->SetVisAttributes(pu_orb_vis);
}
```

END

ExMSEventAction.cc

```
#include "ExMSEventAction.hh"
#include "ExMSRunAction.hh"
#include "ExMSSteppingAction.hh"
#include "G4RunManager.hh"
#include "G4Event.hh"
#include "G4UnitsTable.hh"
#include "G4TrajectoryContainer.hh"
typedef std::map<G4ThreeVector,  G4int > G4PVCounter;
ExMSEventAction*  ExMSEventAction::fgInstance  = 0;
ExMSEventAction*  ExMSEventAction::Instance()
{
```

```
   G4cout<<"Here  I am in Event"<<G4endl;
   return fgInstance;
}
ExMSEventAction::ExMSEventAction()
: G4UserEventAction(),
fPrintModulo(1000),
fEnergySum(0.),
fEnergy2Sum(0.),
fEventMessenger(0)
{
   fgInstance = this;
   strx= strxn = 0;
   avgaccX= avgaccXn= avgaccY= avgaccYn = avgaccZ = avgaccZn =0;
   avgX = avgX=  avgY= avgYn= avgZ = avgZn = 0;
   posevent.clear();
   posvector.clear();
   prevector.clear();
   posttracked.clear();
   post_tracked.clear();
   preTracked.clear();
   pre_tracked.clear();
   totalCounted  = 0;
   totaloptibeingdone  = 0;
   vectorY.clear();
   vectorYn.clear();
   vectorX.clear();
   vectorXn.clear();
   Dphotoncount.clear();
   posVectorN.clear();
   vectorZ.clear();
   vectorZn.clear();
   left_outs.clear();
   eventn1.clear();
   photoeff.clear();
   ph1=ph2=ph3=ph4=0;
   sc1= sc2= sc3= sc4= sc5= sc6= sc7= sc8= sc9= sc10=0;
   sc11= sc12= sc13= sc14= sc15= sc16= 0;
   alongsurfacesX = alongsurfacesXn = alongsurfacesY = alongsurfacesYn =
alongsurfacesZ = alongsurfacesZn = 0;
   total1a=total2a=total3a=total4a=total5a=total6a=0;
   total1b=total2b=total3b=total4b=total5b=total6b=0;
   total1=total2=total3=total4=total5=total6=0;
   FirDisEvent.clear();
   SecDisEvent.clear();
```

```
    ThirdDisEvent.clear();
    FourDisEvent.clear();
    t1=t2=t3=t4=t5=t6=t7=t8=t9=t10=t11=t12=0;
    comptpos.clear();
    comptevent.clear();
    photopos.clear();
    photoevent.clear();
    for (G4int  i=0; i < 100; ++i) {
          for (G4int  j=0; j < 100; ++j) {
             OnPosX[i][j]=0;
             OnNegX[i][j]=0;
          }
       }
    for (G4int  i=0; i < 100; ++i) {
          for (G4int  j=0; j < 100; ++j) {
             OnPosY[i][j]=0;
             OnNegY[i][j]=0;
          }
       }
     for (G4int  i=0; i < 100; ++i) {
          for (G4int  j=0; j < 100; ++j) {
             OnPosZ[i][j]=0;
             OnNegZ[i][j]=0;
          }
       }
    tavgX = tavgXn= tavgY= tavgYn= tavgZ= tavgZn=0;
    bavgX = bavgXn= bavgY= bavgYn= bavgZ= bavgZn=0;
    counteve=0;
    p=1;
    q=0.5;
}
ExMSEventAction::~ExMSEventAction()
{
    G4cout << " \n\nData---\n\n";
    for (G4int  i=0; i < ThirdDisEvent.size()  ; ++i)
    {
       G4cout << FirDisEvent.at(i).x()  <<"\t" <<
       FirDisEvent.at(i).y()   <<"\t" <<
       FirDisEvent.at(i).z()   <<"\t" <<
       SecDisEvent.at(i).x()  <<"\t" <<
       SecDisEvent.at(i).y()  <<"\t" <<
       SecDisEvent.at(i).z() << "\t" <<
       ThirdDisEvent.at(i).x()  <<"\t" <<
       ThirdDisEvent.at(i).y()  <<"\t" <<
```

```
      ThirdDisEvent.at(i).z()  << G4endl;
   }
   G4cout << " \n\n---\n\n";
   G4cout << "Exit...";
   fgInstance = 0;
}
void ExMSEventAction::BeginOfEventAction(const  G4Event* event)
{
   fDecayChain = " ";
   c1PE=c2PE=0;
   logPE=false;
   G4int  eventNb = event->GetEventID();
   evexno = eventNb;
}
void ExMSEventAction::EndOfEventAction(const  G4Event* evt/*event*/)
{
   evtNb = evt->GetEventID();
   PrintNumbers();
   if (post_tracked.size() > 0)
   {
      ++sc7;
      InitializeComments();
   }
   else
   { ++counteve; //G4cout << "No Photon Detected ... Number of such events: "<<
counteve << G4endl;
   }
   strx= strxn = 0;
   avgaccX= avgaccXn= avgaccY= avgaccYn = avgaccZ = avgaccZn =0;
   avgXn = avgX=  avgY= avgYn= avgZ = avgZn = 0;
   posevent.clear();
   posvector.clear();
   prevector.clear();
   posttracked.clear();
   post_tracked.clear();
   preTracked.clear();
   pre_tracked.clear();
   totalCounted = 0;
   totaloptibeingdone  = 0;
   vectorY.clear();
   vectorYn.clear();
   vectorX.clear();
   vectorXn.clear();
   Dphotoncount.clear();
```

```
    posVectorN.clear();
    vectorZ.clear();
    vectorZn.clear();
    left_outs.clear();
    alongsurfacesX = alongsurfacesXn = alongsurfacesY = alongsurfacesYn =
alongsurfacesZ = alongsurfacesZn = 0;
    if ( remainder(evexno+1, 400000) ==0)
    {
       printingoutput();
       eventn1.clear();
       photoeff.clear();
       ph1=ph2=ph3=ph4=0;
       sc1= sc2= sc3= sc4= sc5= sc6= sc7= sc8= sc9= sc10=0;
       sc11= sc12= sc13= sc14= sc15= sc16= 0;
       comptpos.clear();
       comptevent.clear();
       photopos.clear();
       photoevent.clear();
       total1a=total2a=total3a=total4a=total5a=total6a=0;
       total1b=total2b=total3b=total4b=total5b=total6b=0;
       total1=total2=total3=total4=total5=total6=0;
       tavgX = tavgXn= tavgY= tavgYn= tavgZ= tavgZn=0;
       bavgX = bavgXn= bavgY= bavgYn= bavgZ= bavgZn=0;
       counteve=0;
       for (G4int i=0; i < 100; ++i) {
          for (G4int j=0; j < 100; ++j) {
             OnPosX[i][j]=0;
             OnNegX[i][j]=0;
          }
       }
       for (G4int i=0; i < 100; ++i) {
          for (G4int j=0; j < 100; ++j) {
             OnPosY[i][j]=0;
             OnNegY[i][j]=0;
          }
       }
       for (G4int i=0; i < 100; ++i) {
          for (G4int j=0; j < 100; ++j) {
             OnPosZ[i][j]=0;
             OnNegZ[i][j]=0;
          }
       }
    }
}
```

```
#include <math.h>
void ExMSEventAction::printingoutput()
{
   if (sc7>1) {
      FirDisEvent.push_back(G4ThreeVector(sc1,sc2,sc3));
      SecDisEvent.push_back(G4ThreeVector(sc4,sc5,sc6));
      FourDisEvent.push_back(G4ThreeVector(sc7,0,0));
      G4cout << sc1 << "\t" <<
      sc2 << "\t" <<
      sc3 << "\t" <<
      sc4 << "\t" <<
      sc5 << "\t" <<
      sc6 << "\t" << G4endl;
   }
}
void ExMSEventAction::PrintNumbers()
{
   totatsame=0;
   iter=0;
   ixn = ix = iyn = iy = izn = iz = 0;
   if (post_tracked.size()>0)
   {
      do {
         if ( post_tracked.at(iter).z() == 40.8432 ) {
            //  G4cout << "Got  the Z " << G4endl;
            vectorZ.push_back (post_tracked.at(iter)); ++alongsurfacesZ; ++alongsurfaces;
++avgZ; ++bavgZ;}
         else if ( post_tracked.at(iter).x() == -2.54 ) {
            vectorXn.push_back(post_tracked.at(iter));                    ++alongsurfacesXn;
++alongsurfaces; ++avgXn;++bavgXn;}
         else if ( post_tracked.at(iter).z() == 0.2032 ) {
            vectorZn.push_back(post_tracked.at(iter));                    ++alongsurfacesZn;
++alongsurfaces; ++avgZn; ++bavgZn;}
         else if ( post_tracked.at(iter).y() == 5.08 ) {
            vectorY.push_back(post_tracked.at(iter)); ++alongsurfacesY; ++alongsurfaces;
++avgY; ++bavgY;}
         else if ( post_tracked.at(iter).y() == -5.08 ) {
            vectorYn.push_back(post_tracked.at(iter));                    ++alongsurfacesYn;
++alongsurfaces; ++avgYn; ++bavgYn;}
         else if ( post_tracked.at(iter).x() == 2.54 ) {
            vectorX.push_back(post_tracked.at(iter)); ++alongsurfacesX; ++alongsurfaces;
++avgX; ++bavgX;}
         else
         {
```

```
            left_outs.push_back(post_tracked.at(iter));}
          ++iter;
      } while (post_tracked.size()>iter);
   }
      SortingforPosX();
      SortingforNegX();
      SortingforPosY();
      SortingforNegY();
      SortingforPosZ();
      SortingforNegZ();
}
void ExMSEventAction::InitializeComments()
{
      ++ph1;
      if (avgXn > avgX)
      {  ++ph2;
         ph3 = ph3 + avgXn/(avgXn+avgX+avgYn+avgY+avgZn+avgZ);
      }
      else
         ph4 = ph4 + avgX/(avgXn+avgX+avgYn+avgY+avgZn+avgZ);
   if (sc1 == 1 ) {
      sc8=avgXn;
      sc9=avgX;
      sc10=avgYn;
      sc11=avgY;
      sc12=avgZn;
      sc13=avgZ;
      sc14=avgXn;
      sc15=avgX;
      sc16=avgYn;
      sc17=avgY;
      sc18=avgZn;
      sc19=avgZ;
   }
   if (sc7>1) {
      if (avgXn > sc8) {
         sc8=avgXn;
      }
      if (avgXn < sc14) {
         sc14=avgXn;
      }
      if (avgX > sc9) {
         sc9=avgX;
      }
```

```
        if (avgX < sc15) {
           sc15= avgX;
        }
        if (avgYn > sc10) {
           sc10=avgYn;
        }
        if (avgYn < sc16) {
           sc16=avgYn;
        }
        if (avgY > sc11) {
           sc11=avgY;
        }
        if (avgY < sc17) {
           sc17= avgY;
        }
        if (avgZn > sc12) {
           sc12=avgZn;
        }
        if (avgZn < sc18) {
           sc18 = avgZn;
        }
        if (avgZ > sc13) {
           sc13=avgZ;
        }
        if (avgZ < sc19) {
           sc19 = avgZ;
        }
  }
    sc1 = sc1+avgXn;
    sc2 = sc2+avgX;
    sc3 = sc3+avgYn;
    sc4 = sc4+avgY;
    sc5 = sc5+avgZn;
    sc6 = sc6+avgZ;
}
void ExMSEventAction::SortingforPosX()
{
    for (G4double  i=0; i < vectorX.size(); ++i) {
       total1a  = total1a + 1;
       noj = 0;
       for (G4double  k=0.2032; k < 40.8432; k=k+p)
       {
          nok = 0;
               for (G4double  j=-5.08; j < 5.08; j=j+q){
```

```
            if (vectorX.at(i).y()>= j && vectorX.at(i).y()< j+q && vectorX.at(i).z()>=k  &&
               vectorX.at(i).z()<  k+p ) {
               total1  = total1  + 1;
               OnPosX[noj][nok]  = OnPosX[noj][nok]  + 1;
            }
               ++nok;
         }
      ++noj;
      }
   }
   t1=noj;
   t2=nok;
}
void ExMSEventAction::SortingforNegX()
{
   for (G4double  i=0;  i < vectorXn.size();  ++i)  {
      total2a  = total2a  + 1;
      noj = 0;
      for (G4double  k=0.2032;  k < 40.8432;  k=k+p)
      {
         nok = 0;
      for (G4double  j=-5.08;  j < 5.08;  j=j+q){
            if (vectorXn.at(i).y()>= j && vectorXn.at(i).y()< j+q && vectorXn.at(i).z()>=k
&&
               vectorXn.at(i).z()<  k+p )
            {
               OnNegX[noj][nok]  = OnNegX[noj][nok]  + 1;
               total2  = total2  + 1;
            }
            ++nok;
         }
         ++noj;
      }
   }
   t3=noj;
   t4=nok;
}
void ExMSEventAction::SortingforPosY()
{
   for (G4double  i=0;  i < vectorY.size();  ++i)  {
      noj = 0;
      total3a  = total3a  + 1;
      for (G4double  k=0.2032;  k < 40.8432;  k=k+p)
      {
```

```
        nok = 0;
           for (G4double  j=-2.54;  j < 2.54; j=j+q){
           if (vectorY.at(i).x()>= j && vectorY.at(i).x()< j+q && vectorY.at(i).z()>=k &&
              vectorY.at(i).z()< k+p )
           {
                              OnPosY[noj][nok]  = OnPosY[noj][nok]  + 1;
              total3  = total3  + 1;

           }
           ++nok;
        }
        ++noj;
     }
  }
  t5=noj;
  t6=nok;
}
void ExMSEventAction::SortingforNegY()
{
  for (G4double  i=0;  i < vectorYn.size();  ++i)  {
     noj = 0;
           total4a  = total4a  + 1;
        for (G4double  k=0.2032;  k < 40.8432;  k=k+p)
     {
        nok = 0;
            for (G4double  j=-2.54;  j < 2.54; j=j+q){
           if (vectorYn.at(i).x()>= j && vectorYn.at(i).x()< j+q && vectorYn.at(i).z()>=k
&&
              vectorYn.at(i).z()< k+p )
           {
              OnNegY[noj][nok]  = OnNegY[noj][nok]  + 1;
              total4  = total4  + 1;
           }
           ++nok;
        }
        ++noj;
     }
  }
  t7=noj;
  t8=nok;
}
void ExMSEventAction::SortingforPosZ()
{
  for (G4double  i=0;  i < vectorZ.size();  ++i)  {
```

```
      noj = 0;
            total5a  = total5a + 1;
       for (G4double  k=-5.08; k < 5.08; k=k+q)
       {
          nok = 0;
          for (G4double  j=-2.54; j < 2.54; j=j+q){
             if (vectorZ.at(i).x()>= j && vectorZ.at(i).x()< j+q  && vectorZ.at(i).y()>=k  &&
                vectorZ.at(i).y()< k+q)
             {
                OnPosZ[noj][nok]  = OnPosZ[noj][nok]  + 1;
                total5  = total5  + 1;
          }
             ++nok;
          }
          ++noj;
       }
   }
   t9=noj;
   t10=nok;
}
void ExMSEventAction::SortingforNegZ()
{
   for (G4double  i=0; i < vectorZn.size(); ++i)  {
      noj = 0;
            total6a  = total6a + 1;
      for (G4double  k=-5.08; k < 5.08; k=k+q)  {
          nok = 0;
          for (G4double  j=-2.54; j < 2.54; j=j+q){
             if (vectorZn.at(i).x()>= j && vectorZn.at(i).x()< j+q  && vectorZn.at(i).y()>=k
&&
                vectorZn.at(i).y()< k+q  )
             {
               OnNegZ[noj][nok]  = OnNegZ[noj][nok]  + 1;
                total6  = total6  + 1;
            }
              ++nok;
          }
          ++noj;
      }
   }
   t11=noj;
   t12=nok;}
void ExMSEventAction::InitializeVariable()
{
```

```
   G4cout << "\n\n Initialization of storing variable done \n\n" << G4endl;
}
void ExMSEventAction::InitializePrinting()
{
   std::ostringstream str1;
   G4String stringval="Points";
   str1 << ph1;
   std::string str2= str1.str();
   stringval.append(str2);
   myfile << "\n\n Data \n\n" << G4endl;
   myfile << "\t" << totalpos.x() << "\t" << totalpos.y() << "\t" << totalpos.z() << G4endl;
   if (vectorXn.size()>0) {
   myfile << "\n\nOn -Ve X at -2.54 Cm \n\n" << G4endl;
   for (G4double i=0; i< vectorXn.size(); ++i) {
      myfile << i << "\t" << vectorXn.at(i).x() << "\t" << vectorXn.at(i).y() << "\t" <<
vectorXn.at(i).z() << G4endl;
   }
   }
   if (vectorX.size()>0) {
   myfile << "\n\nOn +Ve X at 2.54 Cm \n\n" << G4endl;
   for (G4double i=0; i< vectorX.size(); ++i) {
            myfile << i << "\t" << vectorX.at(i).x() << "\t" << vectorX.at(i).y() << "\t" <<
vectorX.at(i).z() << G4endl;
         }
   }
   if (vectorYn.size()>0) {
   myfile << "\n\nOn -Ve Y at -5.08 Cm \n\n" << G4endl;
   for (G4double i=0; i< vectorYn.size(); ++i) {
      myfile << i << "\t" << vectorYn.at(i).x() << "\t" << vectorYn.at(i).y() << "\t" <<
vectorYn.at(i).z() << G4endl;
   }
   }
   if (vectorY.size()>0) {
   myfile << "\n\nOn +Ve Y at 5.08 Cm \n\n" << G4endl;
   for (G4double i=0; i< vectorY.size(); ++i) {
      myfile << i << "\t" << vectorY.at(i).x() << "\t" << vectorY.at(i).y() << "\t" <<
vectorY.at(i).z() << G4endl;
   }}
   if (vectorZn.size()>0) {
   myfile << "\n\nOn -Ve Z at 0.2032 Cm \n\n" << G4endl;
   for (G4double i=0; i< vectorZn.size(); ++i) {
      myfile << i << "\t" << vectorZn.at(i).x() << "\t" << vectorZn.at(i).y() << "\t" <<
vectorZn.at(i).z() << G4endl;
```

```
   }}
   if (vectorZ.size()>0) {
   myfile << "\n\nOn +Ve Z at 40.8432 Cm \n\n" << G4endl;
   for (G4double i=0; i< vectorZ.size(); ++i) {
      myfile << i << "\t" << vectorZ.at(i).x() << "\t" << vectorZ.at(i).y() << "\t" <<
vectorZ.at(i).z() << G4endl;
   }
   }
   myfile.close();
}

void ExMSEventAction::Reset()
{
   G4cout<<"Here I am in EventReset"<<G4endl;
}
```

END

ExMSPhysicsList.cc

```
#include "globals.hh"
#include "ExMSPhysicsList.hh"
#include "G4ParticleDefinition.hh"
#include "G4ParticleTypes.hh"
#include "G4ParticleTable.hh"
#include "G4ProcessManager.hh"
#include "G4Cerenkov.hh"
#include "G4Scintillation.hh"
#include "G4OpAbsorption.hh"
#include "G4OpRayleigh.hh"
#include "G4OpMieHG.hh"
#include "G4OpBoundaryProcess.hh"
#include "G4OpticalPhoton.hh"
#include "G4LossTableManager.hh"
#include "G4EmSaturation.hh"
#include "G4ChargedGeantino.hh"
#include "G4Geantino.hh"
#include "G4Gamma.hh"
#include "G4OpticalPhoton.hh"
#include "G4MuonPlus.hh"
```

```
#include "G4MuonMinus.hh"
#include "G4NeutrinoMu.hh"
#include "G4AntiNeutrinoMu.hh"
#include "G4Electron.hh"
#include "G4Positron.hh"
#include "G4NeutrinoE.hh"
#include "G4AntiNeutrinoE.hh"
#include "G4PionPlus.hh"
#include "G4PionMinus.hh"
#include "G4PionZero.hh"
#include "G4Eta.hh"
#include "G4EtaPrime.hh"
#include "G4KaonPlus.hh"
#include "G4KaonMinus.hh"
#include "G4KaonZero.hh"
#include "G4AntiKaonZero.hh"
#include "G4KaonZeroLong.hh"
#include "G4KaonZeroShort.hh"
#include "G4Proton.hh"
#include "G4AntiProton.hh"
#include "G4Neutron.hh"
#include "G4AntiNeutron.hh"
#include "G4Deuteron.hh"
#include "G4Triton.hh"
#include "G4Alpha.hh"
#include "G4GenericIon.hh"
ExMSPhysicsList::ExMSPhysicsList() : G4VUserPhysicsList()
{
   G4cout<<"Here I am in PhysicsList"<<G4endl;
  theCerenkovProcess           = NULL;
  theScintillationProcess      = NULL;
  theAbsorptionProcess         = NULL;
  theRayleighScatteringProcess = NULL;
  theMieHGScatteringProcess    = NULL;
  theBoundaryProcess           = NULL;
  SetVerboseLevel(0);
}
ExMSPhysicsList::~ExMSPhysicsList()    {                G4cout<<"Here    I    am    in
PhysicsListEnd"<<G4endl;
}
void ExMSPhysicsList::ConstructParticle()
{   G4cout<<"Here I am in PhysicsListConstr1"<<G4endl;
  ConstructBosons();
  ConstructLeptons();
```

```
  ConstructMesons();
  ConstructBaryons();
    G4Eta::EtaDefinition();
    G4EtaPrime::EtaPrimeDefinition();
    G4KaonPlus::KaonPlusDefinition();
    G4KaonMinus::KaonMinusDefinition();
    G4KaonZero::KaonZeroDefinition();
    G4AntiKaonZero::AntiKaonZeroDefinition();
    G4KaonZeroLong::KaonZeroLongDefinition();
    G4KaonZeroShort::KaonZeroShortDefinition();
    G4Proton::ProtonDefinition();
    G4AntiProton::AntiProtonDefinition();
    G4Neutron::NeutronDefinition();
    G4AntiNeutron::AntiNeutronDefinition();
    G4Deuteron::DeuteronDefinition();
    G4Triton::TritonDefinition();
    G4Alpha::AlphaDefinition();
    G4GenericIon::GenericIonDefinition();
}
void ExMSPhysicsList::ConstructBosons()
{    G4cout<<"Here I am in PhysicsListCBosons2"<<G4endl;
  G4Geantino::GeantinoDefinition();
  G4ChargedGeantino::ChargedGeantinoDefinition();
  G4Gamma::GammaDefinition();
  G4OpticalPhoton::OpticalPhotonDefinition();
}
void ExMSPhysicsList::ConstructLeptons()
{
       G4cout<<"Here I am in PhysicsListLeptons3"<<G4endl;
  G4Electron::ElectronDefinition();
  G4Positron::PositronDefinition();
  G4MuonPlus::MuonPlusDefinition();
  G4MuonMinus::MuonMinusDefinition();
  G4NeutrinoE::NeutrinoEDefinition();
  G4AntiNeutrinoE::AntiNeutrinoEDefinition();
  G4NeutrinoMu::NeutrinoMuDefinition();
  G4AntiNeutrinoMu::AntiNeutrinoMuDefinition();
}
void ExMSPhysicsList::ConstructMesons()
{    G4cout<<"Here I am in PhysicsListMesons4"<<G4endl;
  G4PionPlus::PionPlusDefinition();
  G4PionMinus::PionMinusDefinition();
  G4PionZero::PionZeroDefinition();
}
```

```
void ExMSPhysicsList::ConstructBaryons()
{   G4cout<<"Here I am in PhysicsListBaryons5"<<G4endl;
  G4Proton::ProtonDefinition();
  G4AntiProton::AntiProtonDefinition();
  G4Neutron::NeutronDefinition();
  G4AntiNeutron::AntiNeutronDefinition();
}
void ExMSPhysicsList::ConstructProcess()
{   G4cout<<"Here I am in PhysicsListConProcess5"<<G4endl;
  AddTransportation();
  ConstructGeneral();
  ConstructEM();
  ConstructOp();
}
#include "G4Decay.hh"
void ExMSPhysicsList::ConstructGeneral()
{
  G4cout<<"Here I am in PhysicsListGeneral6"<<G4endl;
  G4Decay* theDecayProcess = new G4Decay();
  theParticleIterator->reset();
  while( (*theParticleIterator)() ){
    G4ParticleDefinition*  particle = theParticleIterator->value();
    G4ProcessManager* pmanager = particle->GetProcessManager();
    if (theDecayProcess->IsApplicable(*particle))  {
      pmanager ->AddProcess(theDecayProcess);
      pmanager ->SetProcessOrdering(theDecayProcess, idxPostStep);
      pmanager ->SetProcessOrdering(theDecayProcess, idxAtRest);
    }
  }
}
#include "G4ComptonScattering.hh"
#include "G4GammaConversion.hh"
#include "G4PhotoElectricEffect.hh"
#include "G4eMultipleScattering.hh"
#include "G4MuMultipleScattering.hh"
#include "G4hMultipleScattering.hh"
#include "G4eIonisation.hh"
#include "G4eBremsstrahlung.hh"
#include "G4eplusAnnihilation.hh"
#include "G4MuIonisation.hh"
#include "G4MuBremsstrahlung.hh"
#include "G4MuPairProduction.hh"
#include "G4hIonisation.hh"
#include "G4OpticalPhysics.hh"
```

```
void ExMSPhysicsList::ConstructEM()
{
      G4cout<<"Here  I am in PhysicsListConEM7"<<G4endl;
 theParticleIterator->reset();
 while( (*theParticleIterator)() ){
  G4ParticleDefinition*  particle = theParticleIterator->value();
  G4ProcessManager* pmanager = particle->GetProcessManager();
  G4String  particleName = particle->GetParticleName();
  if (particleName == "gamma")  {
    pmanager->AddDiscreteProcess(new G4GammaConversion());
    pmanager->AddDiscreteProcess(new G4ComptonScattering());
    pmanager->AddDiscreteProcess(new G4PhotoElectricEffect());
  } else if (particleName == "e-")  {
    pmanager->AddProcess(new G4eMultipleScattering(),-1,  1, 1);
    pmanager->AddProcess(new G4eIonisation(),        -1, 2, 2);
    pmanager->AddProcess(new G4eBremsstrahlung(),    -1, 3, 3);
  } else if (particleName == "e+")  {
    pmanager->AddProcess(new G4eMultipleScattering(),-1,  1, 1);
    pmanager->AddProcess(new G4eIonisation(),        -1, 2, 2);
    pmanager->AddProcess(new G4eBremsstrahlung(),    -1, 3, 3);
    pmanager->AddProcess(new G4eplusAnnihilation(),   0,-1, 4);
  } else if( particleName == "mu+"  ||
             particleName == "mu-"     ) {
   pmanager->AddProcess(new G4MuMultipleScattering(),-1,  1, 1);
   pmanager->AddProcess(new G4MuIonisation(),       -1, 2, 2);
   pmanager->AddProcess(new G4MuBremsstrahlung(),   -1, 3, 3);
   pmanager->AddProcess(new G4MuPairProduction(),   -1, 4, 4);
  } else {
    if ((particle->GetPDGCharge()  != 0.0) &&
        (particle->GetParticleName()  != "chargedgeantino"))  {
     pmanager->AddProcess(new G4hMultipleScattering(),-1,1,1);
     pmanager->AddProcess(new G4hIonisation(),        -1,2,2);
    }
   }
 }
}
void ExMSPhysicsList::ConstructOp()
{   G4cout<<"Here  I am in  PhysicsListContOp7"<<G4endl;
 theCerenkovProcess           = new G4Cerenkov("Cerenkov");
 theScintillationProcess      = new G4Scintillation("Scintillation");
 theAbsorptionProcess         = new G4OpAbsorption();
 theRayleighScatteringProcess = new G4OpRayleigh();
 theMieHGScatteringProcess    = new G4OpMieHG();
 theBoundaryProcess           = new G4OpBoundaryProcess();
```

```
    G4OpticalPhysics*  opticalphysics  = new G4OpticalPhysics();
    opticalphysics->SetWLSTimeProfile("delta");
    opticalphysics->SetScintillationByParticleType(false);
    opticalphysics->SetScintillationExcitationRatio(0.0);
    opticalphysics->SetScintillationYieldFactor(1.0);
  theCerenkovProcess->SetMaxNumPhotonsPerStep(20);
  theCerenkovProcess->SetMaxBetaChangePerStep(10.0);
  theCerenkovProcess->SetTrackSecondariesFirst(true);
  theScintillationProcess->SetScintillationYieldFactor(1.);
  theScintillationProcess->SetTrackSecondariesFirst(true);
  G4EmSaturation*  emSaturation = G4LossTableManager::Instance()->EmSaturation();
  theScintillationProcess->AddSaturation(emSaturation);
  theParticleIterator->reset();
  while( (*theParticleIterator)() ){
    G4ParticleDefinition*  particle = theParticleIterator->value();
    G4ProcessManager* pmanager = particle->GetProcessManager();
    G4String  particleName = particle->GetParticleName();
    if (theCerenkovProcess->IsApplicable(*particle))  {
      pmanager->AddProcess(theCerenkovProcess);
      pmanager->SetProcessOrdering(theCerenkovProcess,idxPostStep);
    }
    if (theScintillationProcess->IsApplicable(*particle))  {
      pmanager->AddProcess(theScintillationProcess);
      pmanager->SetProcessOrderingToLast(theScintillationProcess,  idxAtRest);
      pmanager->SetProcessOrderingToLast(theScintillationProcess,  idxPostStep);
    }
    if (particleName == "opticalphoton")  {
      G4cout << " AddDiscreteProcess to OpticalPhoton  " << G4endl;
      pmanager->AddDiscreteProcess(theAbsorptionProcess);
      pmanager->AddDiscreteProcess(theRayleighScatteringProcess);
      pmanager->AddDiscreteProcess(theMieHGScatteringProcess);
      pmanager->AddDiscreteProcess(theBoundaryProcess);
    }
  }
}
#include  "G4Gamma.hh"
#include  "G4Electron.hh"
#include  "G4Positron.hh"
void  ExMSPhysicsList::SetVerbose(G4int  verbose)
{   G4cout<<"Here  I am in  PhysicsListVerb8"<<G4endl;
  theCerenkovProcess->SetVerboseLevel(verbose);
  theScintillationProcess->SetVerboseLevel(verbose);
  theAbsorptionProcess->SetVerboseLevel(verbose);
  theRayleighScatteringProcess->SetVerboseLevel(verbose);
```

```
  theMieHGScatteringProcess->SetVerboseLevel(verbose);
  theBoundaryProcess->SetVerboseLevel(0);
}
void ExMSPhysicsList::SetNbOfPhotonsCerenkov(G4int  MaxNumber)
{      G4cout<<"Here  I am in PhysicsListCeren9"<<G4endl;
  theCerenkovProcess->SetMaxNumPhotonsPerStep(MaxNumber);
}
void ExMSPhysicsList::SetCuts()
{    G4cout<<"Here  I am in  PhysicsListCutset10"<<G4endl;
  SetCutsWithDefault();
    G4double  cut = 0.1*CLHEP::mm;
   SetCutValue(cut,  "gamma");
   SetCutValue(cut,  "e-");
   SetCutValue(cut,  "e+");
   SetCutValue(cut,  "opticalphoton");
      DumpCutValuesTable();
}
```

END

ExMSPrimaryGeneratorAction.cc

```
#include  "ExMSPrimaryGeneratorAction.hh"
#include  "Randomize.hh"
#include  "G4Event.hh"
#include  "G4ParticleGun.hh"
#include  "G4ParticleTable.hh"
#include  "G4ParticleDefinition.hh"
#include  "G4SystemOfUnits.hh"
#include  "G4UnitsTable.hh"
ExMSPrimaryGeneratorAction::ExMSPrimaryGeneratorAction()
{    G4cout<<"Here  I am in  PrimaryGen"<<G4endl;
  G4int n_particle = 1;
  particleGun  = new G4ParticleGun(n_particle);
    G4ThreeVector position=  G4ThreeVector (-15*cm,0*cm,10*cm);
    particleGun->SetParticlePosition(position);
  G4ParticleTable*  particleTable  = G4ParticleTable::GetParticleTable();
  G4ParticleDefinition*   particle = particleTable->FindParticle("gamma");
    Energy = 1332;
    G4double  rand = G4UniformRand();
```

```
    G4double  z = 2*G4UniformRand()  - 1;
    G4double  x = (sqrt(1.-(z*z)))*cos(CLHEP::twopi*rand);
    G4double  y = (sqrt(1.-(z*z)))*sin(CLHEP::twopi*rand);
    G4ThreeVector v(x,y,z);
    particleGun->SetParticleMomentumDirection(v);
    particleGun->SetParticleDefinition(particle);
    particleGun->SetParticleEnergy(Energy*keV);
}
ExMSPrimaryGeneratorAction::~ExMSPrimaryGeneratorAction()
{
  delete particleGun;
}
void ExMSPrimaryGeneratorAction::GeneratePrimaries(G4Event*  anEvent)
{
  particleGun->GeneratePrimaryVertex(anEvent);
    G4double  rand = G4UniformRand();
    G4double  z = 2*G4UniformRand()  - 1;
    G4double  x = (sqrt(1.-(z*z)))*cos(CLHEP::twopi*rand);
    G4double  y = (sqrt(1.-(z*z)))*sin(CLHEP::twopi*rand);
    G4ThreeVector v(x,y,z);
    particleGun->SetParticleMomentumDirection(v);
}
void ExMSPrimaryGeneratorAction::SetOptPhotonPolar()
{    G4cout<<"Here  I am in  PrimaryGenOptPhoton2"<<G4endl;
 G4double  angle = G4UniformRand()  * 360.0*deg;
 SetOptPhotonPolar(angle);
}
void ExMSPrimaryGeneratorAction::SetOptPhotonPolar(G4double  angle)
{
       G4cout<<"Here  I am in  PrimaryGenOptPhot3"<<G4endl;
 if (particleGun->GetParticleDefinition()->GetParticleName()  != "opticalphoton")
   {
    G4cout << "--> warning  from PrimaryGeneratorAction::SetOptPhotonPolar()  :"
           "the particleGun  is not an opticalphoton"  << G4endl;
    return;
   }
G4ThreeVector normal (1., 0., 0.);
 G4ThreeVector kphoton  = particleGun->GetParticleMomentumDirection();
 G4ThreeVector product = normal.cross(kphoton);
 G4double  modul2       = product*product;
G4ThreeVector e_perpend (0., 0., 1.);
 if (modul2  > 0.) e_perpend = (1./std::sqrt(modul2))*product;
 G4ThreeVector e_paralle    = e_perpend.cross(kphoton);
```

```
 G4ThreeVector polar = std::cos(angle)*e_paralle  + std::sin(angle)*e_perpend;
 particleGun->SetParticlePolarization(polar);
}
```

END

ExMSRunAction.cc

```
#include "G4Timer.hh"
#include "ExMSRunAction.hh"
#include "G4Run.hh"
#include "G4RunManager.hh"
ExMSRunAction::ExMSRunAction(ExMSEventAction*  eveAct)
: runactionp(eveAct)
{
      G4cout<<"Here  I am in RunAction"<<G4endl;
  timer = new G4Timer;

   eveAct          =          (ExMSEventAction*)G4RunManager::GetRunManager()-
>GetUserEventAction();
//              eventaction    =(ExMSEventAction*)G4RunManager::GetRunManager()-
>GetUserEventAction();
}
ExMSRunAction::~ExMSRunAction()
{
        G4cout<<"Here  I am in RunActionEnd"<<G4endl;
  delete timer;
}
void ExMSRunAction::BeginOfRunAction(const  G4Run* aRun)
{
   timer->Start();
   runactionp->InitializeVariable();
}
void ExMSRunAction::EndOfRunAction(const  G4Run*  aRun)
{
        G4cout<<"Here  I am in RunActionEndAction2"<<G4endl;
  timer->Stop();
  G4cout << "number  of event = " << aRun->GetNumberOfEvent()
       << " " << *timer << G4endl;
}
```

END

ExMSStackingAction.cc

```
#include "ExMSStackingAction.hh"
#include "G4VProcess.hh"
#include "G4ParticleDefinition.hh"
#include "G4ParticleTypes.hh"
#include "G4Track.hh"
#include "G4ios.hh"
#include "G4Scintillation.hh"
#include "ExMSSteppingAction.hh"
#include "G4RunManager.hh"
ExMSStackingAction::ExMSStackingAction()
   :  gammaCounter(0),  cerenkovcounter(0),  totalelectrons(0),  totalpart2(0),
gammaconv(0), comptons(0), photoele(0)
{//G4cout<<"Here I am in Stacking"<<G4endl;
detectorstack  =  (ExMSDetectorConstruction*)G4RunManager::GetRunManager()-
>GetUserDetectorConstruction();
}
ExMSStackingAction::~ExMSStackingAction()
{//G4cout<<"Here I am in StackingEnd"<<G4endl;
}
G4ClassificationOfNewTrack
ExMSStackingAction::ClassifyNewTrack(const G4Track * aTrack)
{
   if(aTrack->GetDefinition()==G4OpticalPhoton::OpticalPhotonDefinition()){
      if(aTrack->GetParentID()>0){
         if(aTrack->GetCreatorProcess()->GetProcessName()=="Scintillation")
            gammaconv++;
         else if(aTrack->GetCreatorProcess()->GetProcessName()=="Cerenkov")
            cerenkovcounter++;
      }
   }
   else{
   }
   return fUrgent;
}
```

```
void ExMSStackingAction::PrepareNewEvent()
{
  gammaCounter = 0;  cerenkovcounter = 0;
    photoele=0;  comptons=0;  gammaconv=0;
    totalpart2 = totalelectrons=  0;
}
```

END

ExMSSteppingAction.cc

```
#include "ExMSSteppingAction.hh"
#include "G4VTouchable.hh"
#include "G4OpBoundaryProcess.hh"
#include "ExMSDetectorConstruction.hh"
#include "G4OpticalPhoton.hh"
#include "G4Track.hh"
#include "G4Step.hh"
#include "G4RunManager.hh"
#include "G4UnitsTable.hh"
#include "G4Event.hh"
#include "G4Gamma.hh"
#include "G4VProcess.hh"
#include "G4UserTrackingAction.hh"
#include "G4TrackingManager.hh"
#include "G4Trajectory.hh"
#include "G4VTrajectory.hh"
#include "G4TrackVector.hh"
class G4VSteppingVerbose;
#include "G4SmoothTrajectory.hh"
#include "G4RichTrajectory.hh"
#include "G4ios.hh"
#include "G4Scintillation.hh"
ExMSSteppingAction*  ExMSSteppingAction::fgInstance  = 0;
ExMSSteppingAction*  ExMSSteppingAction::Instance()
{
   return fgInstance;
}
#include "G4Timer.hh"
ExMSSteppingAction::ExMSSteppingAction(ExMSEventAction*  evtact)
```

```
   : G4UserSteppingAction(), eventaction(evtact), fOneStepPrimaries(false)
{  G4cout<<"Here I am in Stepping1"<<G4endl;
   fgInstance = this;
   timerboss = new G4Timer;
   detector = (ExMSDetectorConstruction*)G4RunManager::GetRunManager()-
>GetUserDetectorConstruction();
   eventaction =(ExMSEventAction*)G4RunManager::GetRunManager()-
>GetUserEventAction();
   resetvalues();
   energyto = 1332;
   fEnergy = 1.33; // 0.058, 0.18, .41, .66, 0.998, 1.33,
   timerboss->Start();
}
ExMSSteppingAction::~ExMSSteppingAction()
{
   G4cout << " Termination Stepping Action " << G4endl;
   countprint();
   timerboss->Stop();
   G4cout << " Time: " << *timerboss << G4endl;
   G4cout << " RealTime: " << timerboss->GetClockTime() << G4endl;
   fgInstance = 0;
}
void ExMSSteppingAction::resetvalues()
{
   fscintillationcounter = 0;
   fcerenkovcounter = 0;
   detector_volume_optical_photon = 0;
   total_optical_photons = 0;
   Exitque = 1;
   countotalphoton=0;
   DPhotCount = 0;
   electroncounts = 0;
   withincounts = 0;
   onestepcounts = 0;
   PhotCount =0;
   OpPhotonCount=0;
   feventnumber = -1;
   LastTrackID = -1;
   temp1 = temp2= temp3=0;
   minEn=0;
   maxEn=0;
   avgEnergy=0;
   Extracheck=0;
   Extracheck2=0;
```

```
totalop=0;
fscintillationcounterx=0;
fcerenkovcounterx =0;
absrpcount = 0;
photondetecount  = 0;
backreflcount = 0;
ExitScinCount  = 0;
backcount = 0;
EnergyTtoal=0;
DeporTotal=0;
ExtraP = 0;
ExtraEnergy=0;
dtrackId = -1;
dprantD = -1;
EnergyVec.clear();
CountingPhotons=0;
CountingDetectedPhotons=0;
PhotoEffCounter=  PEC1= PEC2= PEC3=0;
eTotalInternalReflection=0;
eLambertianReflection=0;
eLobeReflection=0;
eSpikeReflection=0;
eTER=0;
fExpectedNextStatus = Undefined;
temp4=temp5=0;
temp6=-1;
temp21=-2;
coun1 = coun2 = coun3=  coun4  = coun5 = coun6=coun7  = coun8 = 0;
counterforcuttingcompton=0;
comptonscacount=  photocounter=  0;
comptset=0;
stter=0;
photoset=0;
temp10=0;
temp20=0;
temp22=0;
temp40=temp41=0;
temp7=temp8=temp9=temp80=temp81=0;
temp11=  temp12  = 0;
run_j1=  run_j2=  run_j3=  run_j4=  run_j5=  run_j6=  run_j7=  run_j8=  run_j9=  run_j10=
run_j11=  run_j12=0;
run_j13  =  run_j14=  run_j15=  run_j16=  run_j17=  run_j18=  run_j19=  run_j20=
run_j21=  run_j22=0;
```

```
    fscintillationcounter  = 0;
    fcerenkovcounter = 0;
    total_optical_photons  = 0;
    ElectronCount  = 0;
    Counting=0;
    electroncounts = 0;
    withincounts  = 0;
    onestepcounts = 0;
}
void ExMSSteppingAction::countprint()
{
    G4double  check = 0;
    check=absrpcount + photondetecount+  eTotalInternalReflection+
    eTER + backreflcount+OpAbsCount;
    if (comptset==-2)  {
       ++comptonscacount;
    }
    if (coun1 != -3) {
       if (eventNumber > 0) {
          ++coun7;
       }
    }
    if (temp2>0 && temp5>temp2  && temp6 == 0) {
       temp5=temp2;
    }
    if (temp2>0 && temp4<temp2  && temp6 == 0) {
       temp4=temp2;
    }
    if (coun1 == -3 && coun3 == -2) {
       if (temp10 > fEnergy) {
          ++temp7;
          if (temp8 <= 2.3E8)
             temp8 = temp8 + temp2;
          else
             temp80 = temp80 + temp2;
             temp81 = temp81 + temp2;
             temp9= temp9+ temp11;
       }
    }
    if (coun1 == -3 && coun3 != -2) {
       if (temp10 > fEnergy) {
          ++temp7;
          if (temp8 <= 2.3E8)
             temp8 = temp8 + temp2;
```

```
        else
           temp80 = temp80 + temp2;
        temp81 = temp81 + temp2;
        temp9= temp9+ temp11;
      }
   }
   G4double first, second, third, fourth, fifth, sixth, seventh, eigth, nine, tenth, e1, e2, e3, e4, e5, e6, e7;
   G4double efirst, esecond, ethird, efourth, efifth, esixth, eseventh, eeigth, enine, etenth, ee1, ee2, ee3, ee4, ee5, ee6, ee7;
   first= second= third= fourth= fifth= sixth= seventh= eigth= nine= tenth= e1= e2= e3= e4= e5= e6= e7= 0 ;
   efirst= esecond= ethird= efourth= efifth= esixth= eseventh= eeigth= enine= etenth= ee1=ee2=ee3=ee4=ee5=ee6=ee7= 0;
   G4long total_pe;
   total_pe = coun5+coun6;
   G4long xg=temp81;
   G4long yg=temp9;
   G4long zg=total_pe;
   G4ThreeVector tpe_thee;
   tpe_thee=G4ThreeVector(xg,yg,zg);
   eventaction->AddStep3(tpe_thee);
   G4double totalCouE;
   if (EnergyVec.size()>0) {
      for (G4int i=0; i< EnergyVec.size(); ++i) {
         if (EnergyVec.at(i) < 2.3) {
            ++first;
            efirst = efirst + EnergyVec.at(i);
         }
         if (EnergyVec.at(i) >= 2.3 && EnergyVec.at(i) < 2.4) {
            ++second;
            esecond = esecond + EnergyVec.at(i);
         }
         if (EnergyVec.at(i) >= 2.4 && EnergyVec.at(i) < 2.5) {
            ++third;
            ethird = ethird + EnergyVec.at(i);
         }
         if (EnergyVec.at(i) >= 2.5 && EnergyVec.at(i) < 2.6) {
            ++fourth;
            efourth = efourth + EnergyVec.at(i);
         }
         if (EnergyVec.at(i) >= 2.6 && EnergyVec.at(i) < 2.7) {
            ++fifth;
            efifth = efifth + EnergyVec.at(i);
```

```
}
if (EnergyVec.at(i) >= 2.7 && EnergyVec.at(i) < 2.8) {
   ++sixth;
   esixth = esixth + EnergyVec.at(i);
}
if (EnergyVec.at(i) >= 2.8 && EnergyVec.at(i) < 2.9) {
   ++seventh;
   eseventh = eseventh + EnergyVec.at(i);
}
if (EnergyVec.at(i) >= 2.9 && EnergyVec.at(i) < 3) {
   ++eigth;
   eeigth = eeigth + EnergyVec.at(i);
}
if (EnergyVec.at(i) >= 3 && EnergyVec.at(i) < 3.1) {
   ++nine;
   enine = enine + EnergyVec.at(i);
}
if (EnergyVec.at(i) >= 3.1 && EnergyVec.at(i) < 3.2) {
   ++tenth;
   etenth = etenth + EnergyVec.at(i);
}
if (EnergyVec.at(i) >= 3.2 && EnergyVec.at(i) < 3.3) {
   ++e1;
   ee1 = ee1 + EnergyVec.at(i);
}
if (EnergyVec.at(i) >= 3.3 && EnergyVec.at(i) < 3.4) {
   ++e2;
   ee2 = ee2 + EnergyVec.at(i);
}
if (EnergyVec.at(i) >= 3.4 && EnergyVec.at(i) < 3.5) {
   ++e3;
   ee3 = ee3 + EnergyVec.at(i);
}
if (EnergyVec.at(i) >= 3.5 && EnergyVec.at(i) < 3.6) {
   ++e4;
   ee4 = ee4 + EnergyVec.at(i);
}
if (EnergyVec.at(i) >= 3.6 && EnergyVec.at(i) < 3.7) {
   ++e5;
   ee5 = ee5 + EnergyVec.at(i);
}
if (EnergyVec.at(i) >= 3.7 && EnergyVec.at(i) < 3.8) {
   ++e6;
   ee6 = ee6 + EnergyVec.at(i);
```

```
        }
        if (EnergyVec.at(i) >= 3.8 ) {
          ++e7;
          ee7 = ee7 + EnergyVec.at(i);
        }
      }
      totalCouE=
first+second+third+fourth+fifth+sixth+seventh+eigth+nine+tenth+e1+e2+e3+e4+e5+e6
+e7;
   }
   else{
      G4cout << "\n\n From Detector NaI, " <<std::setw(14)<< CountingPhotons <<
G4endl;
      // G4cout << "TotalAccountedFfromCaseFunction: " << std::setw(14) << check <<
G4endl;
   }
  resetvalues();
}
#include "G4UserTrackingAction.hh"
#include "G4ParticleDefinition.hh"
#include "G4ParticleTypes.hh"
#include "G4VProcess.hh"
void ExMSSteppingAction::UserSteppingAction(const G4Step* step)
{
   eventNumber = G4RunManager::GetRunManager()->GetCurrentEvent()-
>GetEventID();
   if (eventNumber != feventnumber && eventNumber>0)
   {
      if (remainder (eventNumber, 400000) == 0) {
         countprint();
      }
   }
   static G4VPhysicalVolume* volume;
   G4Track* theTrack = step->GetTrack();
   G4StepPoint* thePrePoint = step->GetPreStepPoint();
   G4VPhysicalVolume* thePrePV = thePrePoint->GetPhysicalVolume();
   G4StepPoint* thePostPoint = step->GetPostStepPoint();
   G4VPhysicalVolume* thePostPV = thePostPoint->GetPhysicalVolume();
   G4int trackno = step->GetTrack()->GetTrackID();
   G4int parentno = step->GetTrack()->GetParentID();
   if(!thePostPV || !thePrePV){//out of world
      return;
   }
```

```
if (eventNumber != feventnumber)
{
   if (coun1 != -3) {
      if (eventNumber > 0) {
         ++coun7;
      }
   }
   temp21=-2;
   if (temp2>0 && temp5>temp2  && temp6 == 0) {
      temp5=temp2;
   }
   if (temp2>0 && temp4<temp2  && temp6 == 0) {
      temp4=temp2;
   }
   if (temp2>0 && temp6 == -1) {
      temp6=0;
      temp4=temp2;
      temp5=temp2;
   }
   if (coun1 == -3 && coun3 == -2) {
      if (temp10 > fEnergy) {
         ++temp7;
         if (temp8 <= 2.3E8)
            temp8 = temp8 + temp2;
         else
            temp80 = temp80 + temp2;

         temp81 = temp81 + temp2;
         temp9= temp9+ temp11;
      }
   }
   if (coun1 == -3 && coun3 != -2) {
      if (temp10 > fEnergy) {
         ++temp7;
         if (temp8 <= 2.3E8)
            temp8 = temp8 + temp2;
         else
            temp80 = temp80 + temp2;
         temp81 = temp81 + temp2;
         temp9= temp9+ temp11;
      }
   }
   feventnumber = eventNumber;
   fscintillationcounter  = 0;
```

```
        fcerenkovcounter = 0;
        total_optical_photons  = 0;
        setconv = false;
        electroncounts = 0;
        withincounts  = 0;
        onestepcounts = 0;
        Counting=0;
        temp1 = temp2= 0; //temp3=0;
        temp11=0;
OpAbsCount=absrpcount=photondetecount=CountingDetectedPhotons=eTotalInternalR
eflection=eLambertianReflection=eLobeReflection=eSpikeReflection=eTER=backreflco
unt=0;
        temp10=0;
        comptset=-1;
        stter=-2;
        photoset=-1;
        coun1 = coun2 = coun3= coun4 = 0;
    }
    G4String particlename = step->GetTrack()->GetDynamicParticle()-
>GetParticleDefinition()->GetParticleName();
    DeporTotal = DeporTotal + thePrePoint->GetKineticEnergy() - thePostPoint-
>GetKineticEnergy();
    G4OpBoundaryProcessStatus boundaryStatus = Undefined;
    static G4OpBoundaryProcess* boundary = NULL;
    if (!boundary) {
       G4ProcessManager* pm = step->GetTrack()->GetDefinition()-
>GetProcessManager();
        G4int nprocesses = pm->GetProcessListLength();
        G4ProcessVector* pv = pm->GetProcessList();
        G4int i;
        for (i=0; i< nprocesses; i++){
          if ((*pv)[i]->GetProcessName() =="OpBoundary")  {
             boundary = (G4OpBoundaryProcess*)(*pv)[i];
             break;
          }
        }
    }
    if (thePostPoint->GetProcessDefinedStep()->GetProcessName() == "phot")
    {
       ++DPhotCount;
    }
    if (thePostPoint->GetProcessDefinedStep()->GetProcessName() == "OpAbsorption")
    {
       ++OpAbsCount;
```

```
   }
   if(theTrack->GetParentID()==0){
    if      (      (step->GetPostStepPoint()->GetProcessDefinedStep()->GetProcessName()
=="compt"  ||
           step->GetPostStepPoint()->GetProcessDefinedStep()->GetProcessName()      ==
"phot"  ) && thePostPV->GetName()  != "Detector_NaI" && temp21  ==  -2)
       {
          ++temp40;
          temp21=-4;
    }
    G4TrackVector* fSecondary=fpSteppingManager->GetfSecondary();
      G4int  tN2ndariesTot = fpSteppingManager->GetfN2ndariesAtRestDoIt()
      + fpSteppingManager->GetfN2ndariesAlongStepDoIt()
      + fpSteppingManager->GetfN2ndariesPostStepDoIt();
      if( !setconv && tN2ndariesTot>0  ){
         G4double  intg = 0;
         if (thePostPV->GetName() == "Detector_NaI")
            temp10=   temp10+   (step->GetPreStepPoint()->GetKineticEnergy()   -   step-
>GetPostStepPoint()->GetKineticEnergy());
         if      ((*fSecondary)[(*fSecondary).size()-tN2ndariesTot]->GetCreatorProcess()-
>GetProcessName() == "phot"  &&
            (*fSecondary)[(*fSecondary).size()-tN2ndariesTot]->GetTouchableHandle()-
>GetVolume()->GetName()  == "Detector_NaI"
            && stter==-2) {
            ++photocounter;
         }
         stter=0;
       for(size_t lp1=(*fSecondary).size()-tN2ndariesTot;
            lp1<(*fSecondary).size();  lp1++){
            const G4VProcess* creator=(*fSecondary)[lp1]->GetCreatorProcess();
          if  (((*fSecondary)[lp1]->GetCreatorProcess()->GetProcessName())   ==   "phot"
&& thePostPV->GetName() == "Detector_NaI" && temp21  != -4) {
               coun1 = -1;
               coun2 = step->GetTrack()->GetCurrentStepNumber();
               G4double  x= step->GetPostStepPoint()->GetPosition().x()/10;
               G4double  y= step->GetPostStepPoint()->GetPosition().y()/10;
               G4double  z= step->GetPostStepPoint()->GetPosition().z()/10;
               photoposition  = G4ThreeVector(x,y,z);
               eventaction->AddStep2(photoposition,  coun3, eventNumber);
            }
            if       (((*fSecondary)[lp1]->GetCreatorProcess()->GetProcessName())       ==
"compt"  && thePostPV->GetName() == "Detector_NaI"  && temp21  != -4) {
               coun3 = -2;
               coun4 = step->GetTrack()->GetCurrentStepNumber();
```

```
            }
            if(creator){
               ++intg;
               G4String creatorName=creator->GetProcessName();
               if (creatorName=="phot" && thePostPoint->GetPhysicalVolume()-
>GetName() == "Detector_NaI") {
                  ++PhotCount;
                  //step->GetTrack()->SetTrackID(-10);
                  photoset=-2;
                  EnergyTtoal = EnergyTtoal + thePrePoint->GetKineticEnergy() -
thePostPoint->GetKineticEnergy();
               }
               if(creatorName=="phot"||creatorName=="compt"||creatorName=="conv"){
                  position = (*fSecondary)[lp1]->GetPosition();
            }
             }
            if                                    (((*fSecondary)[lp1]->GetParticleDefinition()-
>GetParticleName())=="opticalphoton")
             { ++OpPhotonCount;
               ++temp1;
            }
          }
         }
       if((thePostPoint->GetProcessDefinedStep()->GetProcessName() == "compt" ||
          thePostPoint->GetProcessDefinedStep()->GetProcessName() == "conv"))
       {
                     comptset=-2;
       }
       if (thePostPoint->GetProcessDefinedStep()->GetProcessName() == "phot") {
          ++ExtraP;
          ExtraEnergy = ExtraEnergy + thePrePoint->GetKineticEnergy() - thePostPoint-
>GetKineticEnergy();
       }
    }
    if (theTrack->GetParentID() > 0 && temp10 > fEnergy && coun1 == -1) {
       ++temp20;
    }
     if (theTrack->GetParentID() > 0 && coun1 == -1) {
        coun1=-3;
        ++temp22;
    if (coun3 != -2) {
       ++coun5;              // Just a photo electric effect
```

```
    }
    if (coun3 == -2 && coun2 > coun4) {
      ++coun6;              // Compton then Photo
    }}
    G4ParticleDefinition*  particleType  = theTrack->GetDefinition();
    if(particleType==G4OpticalPhoton::OpticalPhotonDefinition()  && coun1==  -3){
      if (step->GetTrack()->GetCurrentStepNumber()==1)  {
        ++CountingPhotons;
        ++temp2;
      }
      if          (thePrePV->GetName()=="Detector_NaI"          &&          (//(thePostPV-
>GetName()=="Detector_Casing")  ||
        thePostPV->GetName()=="volume_phys"  ))
      {
        ++temp11;
        ++CountingDetectedPhotons;
        ++countotalphoton;
        eventaction->AddStep(thePostPoint->GetPosition(),  eventNumber);
        avgEnergy = avgEnergy + thePrePoint->GetTotalEnergy();
        G4double  tempEn = thePrePoint->GetTotalEnergy();
        EnergyVec.push_back(tempEn*1e6);
        if (photondetecount==1)  {
          minEn=thePrePoint->GetTotalEnergy();
          maxEn= tempEn;
        }
         if (photondetecount>1)  {
          if (minEn > tempEn) {
            minEn  = tempEn;
          }
          if (maxEn < tempEn)  {
            maxEn = tempEn;
          }
        }
      }
      if(thePrePV->GetName()=="Detector_NaI")
      {  fscintillationcounterx++;
        if          (thePostPV->GetName()=="volume_phys"          ||          thePostPV-
>GetName()=="Detector_Casing")  {
          ExitScinCount++;
        }
      }
      if(thePostPoint->GetProcessDefinedStep()->GetProcessName()
        =="OpAbsorption"  && thePrePV->GetName()=="Detector_NaI" && thePostPV-
>GetName()=="Detector_NaI"){
```

```
        fcerenkovcounterx++;
      }
}
}
void ExMSSteppingAction::Reset()
{  G4cout<<"Here  I am in SteppingReset3"<<G4endl;
   for(G4PVmap::iterator  ite = fEnergies.begin();  ite != fEnergies.end(); ++ite)  {
      ite->second = std::make_pair(std::make_pair(0.*CLHEP::eV,  0.*CLHEP::eV), 0);
   }
}
```

END

ExMSSteppingMessenger.cc

```
#include "ExMSSteppingMessenger.hh"
#include "ExMSSteppingAction.hh"
#include "G4UIdirectory.hh"
#include "G4UIcmdWithABool.hh"
ExMSSteppingMessenger::ExMSSteppingMessenger(ExMSSteppingAction*  step)
 : fStepping(step)
{
  fOneStepPrimariesCmd = new G4UIcmdWithABool("/LXe/oneStepPrimaries",this);
  fOneStepPrimariesCmd->SetGuidance("Only  allows  primaries  to  go  one  step  in  the
scintillator  volume  before being killed.");
}
ExMSSteppingMessenger::~ExMSSteppingMessenger(){
  delete fOneStepPrimariesCmd;
}
void
ExMSSteppingMessenger::SetNewValue(G4UIcommand*                command,G4String
newValue){
  if( command == fOneStepPrimariesCmd ){
    //    fStepping->SetOneStepPrimaries(fOneStepPrimariesCmd
    //                        ->GetNewBoolValue(newValue));
  }
}
```

END

ExMSSteppingVerbose.cc


```
#include "ExMSSteppingVerbose.hh"
#include "G4SteppingManager.hh"
#include "G4UnitsTable.hh"
#include "G4VProcess.hh"
#include "G4ParticleDefinition.hh"
#include "G4ParticleTypes.hh"
#include "G4Track.hh"
#include "G4ios.hh"
#include "G4Scintillation.hh"
ExMSSteppingVerbose::ExMSSteppingVerbose()
{}
ExMSSteppingVerbose::~ExMSSteppingVerbose()
{}
void ExMSSteppingVerbose::StepInfo()
{
   CopyState();
   G4int  prec = G4cout.precision(3);
   if( verboseLevel >= 1 ){
      if( verboseLevel >= 4 ) VerboseTrack();
      if( verboseLevel >= 3 )
      {
         if (fTrack->GetDefinition()  == G4OpticalPhoton::OpticalPhotonDefinition()
               || fTrack->GetDefinition()  == G4MuonPlus::MuonPlusDefinition())
         {
            {
               G4cout << G4endl;
               G4cout << std::setw(  5) << "#Step#"     << " "
                      << std::setw(  6) << "X"          << "    "
                      << std::setw(  6) << "Y"          << "    "
                      << std::setw(  6) << "Z"          << "    "
                      << std::setw(  9) << "KineE"      << " "
                      << std::setw(  9) << "dEStep"     << " "
                      << std::setw(10)  << "StepLeng"
                      << std::setw(10)  << "TrakLeng"
                      << std::setw(10)  << "Volume"     << "  "
                      << std::setw(10)  << "Process"   << G4endl;
            }
            {
```

```
            G4cout << std::setw(5) << fTrack->GetCurrentStepNumber() << " "
                << std::setw(6) << G4BestUnit(fTrack->GetPosition().x(),"Length")
                << std::setw(6) << G4BestUnit(fTrack->GetPosition().y(),"Length")
                << std::setw(6) << G4BestUnit(fTrack->GetPosition().z(),"Length")
                << std::setw(6) << G4BestUnit(fTrack->GetKineticEnergy(),"Energy")
                << std::setw(6) << G4BestUnit(fStep->GetTotalEnergyDeposit(),"Energy")
                << std::setw(6) << G4BestUnit(fStep->GetStepLength(),"Length")
                << std::setw(6) << G4BestUnit(fTrack->GetTrackLength(),"Length")
                << "  ";
            if( fTrack->GetNextVolume() != 0 ) {
               G4cout << std::setw(10) << fTrack->GetVolume()->GetName();
            } else {
               G4cout << std::setw(10) << "OutOfWorld";
            }
            if(fStep->GetPostStepPoint()->GetProcessDefinedStep() != 0){
               G4cout << "  "
                   << std::setw(10)
                   << fStep->GetPostStepPoint()->GetProcessDefinedStep()
                     ->GetProcessName();
            } else {
               G4cout << "   UserLimit";
            }
            G4cout << G4endl;
         }
      }
   }
   if( verboseLevel ==2){
      G4int tN2ndariesTot = fN2ndariesAtRestDoIt +
           fN2ndariesAlongStepDoIt +
           fN2ndariesPostStepDoIt;
      if(tN2ndariesTot>0){
         for(size_t lp1=(*fSecondary).size()-tN2ndariesTot;
           lp1<(*fSecondary).size(); lp1++){
           if ((*fSecondary)[lp1]->GetDefinition()->GetParticleName() !=
"opticalphoton") return;
         }
    }
   }
  }
  G4cout.precision(prec);
}
void ExMSSteppingVerbose::TrackingStarted()
{
```

```
    CopyState();
    G4int  prec = G4cout.precision(3);
    if( verboseLevel > 0 ){
       if (fTrack->GetDefinition()  == G4OpticalPhoton::OpticalPhotonDefinition()
            || fTrack->GetDefinition()  == G4MuonPlus::MuonPlusDefinition())
       {
          if(fTrack->GetNextVolume()){
             G4cout << std::setw(10)  << fTrack->GetVolume()->GetName();
          } else {
             G4cout << "OutOfWorld";
          }
          G4cout << "    initStep"  << G4endl;
       }
       G4cout.precision(prec);
    }
}
```

END
