The Design and Implementation of a dedicated circuit to solve a simple photon transport problem by Monte Carlo Method

Tushar Kulkarni, Kedar Lahurikar

Department of Electrical Engineering, San Jose State university, San Jose, California 95192.

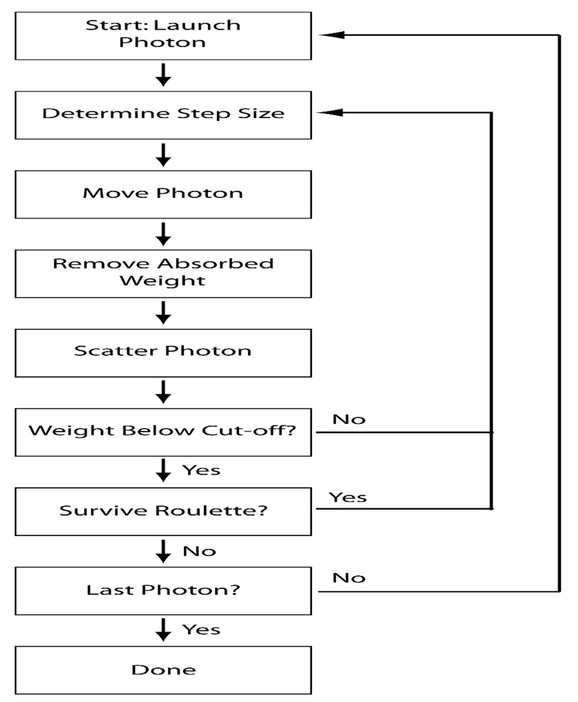
Introduction

Monte Carlo simulations for a photon transport problem involve the probabilistic calculation of location of a charged or an uncharged particle in some point of time for a given finite cross section area. For the highly complicated scientific calculations, at the moment, there are only two computation techniques that are used, standard personal computer based cluster computing and special purpose supercomputing. There have been many advances in the field of cluster computing which have made the Monte Carlo simulations possible at a reasonably faster speed. It has also become a standard solution for scientific calculations. However, cluster computing can also cannot match the speed of special purpose supercomputing systems for such kind of scientific calculations. Also, cluster computers are large enough and expensive. Monte Carlo simulations have been used in many real time applications such as photodynamic radiation therapy [6], monitoring of nuclear reactors to solve various physical problems. Monte Carlo simulations of photon propagation offer a very stable but rigorous approach for the photon transport problem in targeted cross sections for example, cancer causing tissues.

This project aimed at the Design and implementation of the Monte Carlo method and also verification and synthesis. The designs are implemented keeping in mind the future work to implement it on FPGA.

Design Flow

The figure below shows the flow chart of the Monte Carlo Photon transport problem, on which the design is based. The components of the design are listed below



- Random Number Generator
- Interpolation
- Step Size calculation
- Position Update
- Direction Cosines
- Weight Update
- Control Unit

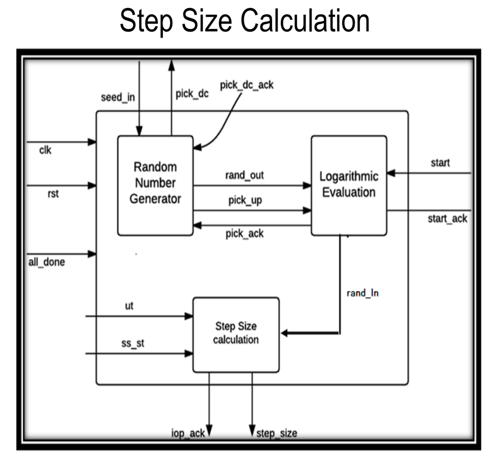
Implementation

The Random Number Generators are the cor of a Monte Carlo Simulation. The Algorithm used to implement is the Mersenne twister algorithm, Period - Mersenne prime value 2^{19937}-1

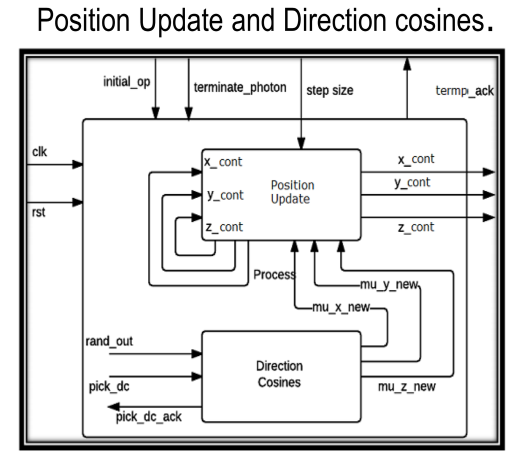
Interpolation data was obtained from the National Data Library. The design was implemented in the form of a lookup table, to speed up the process. Depending upon the energy, the cross section data is stored n to the on chip memory and then accessed from the memory when required.

The figures below show the complex calculation circuits that were implemented and verified. The designs were made synthesizable and designed to be ready to be downloaded on an FPGA. The future work includes implementation on FPGA and creating more real time Monte Carlo Problems.

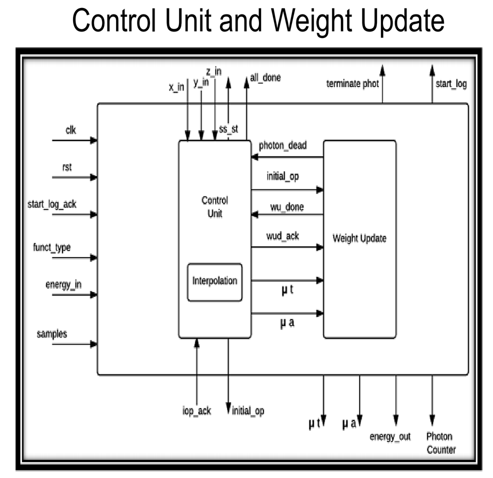
Step Size Calculation



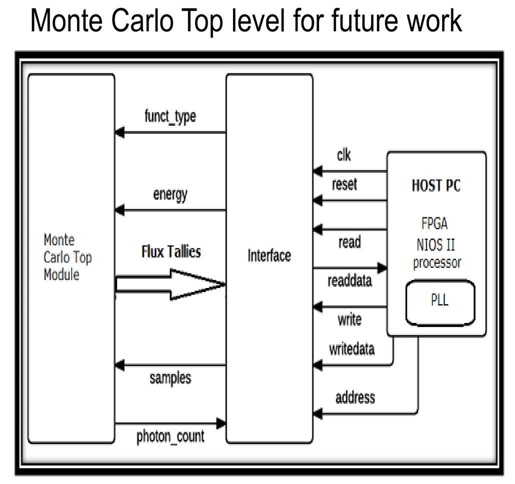
Position Update and Direction cosines.



Control Unit and Weight Update



Monte Carlo Top level for future work



- Step Size

When a photon beam is incident upon a medium, for this project which is Aluminum, it can scatter over any path. If a photon is considered, initially the position of the photon is origin. The step size of a photon is calculated as,

$$s = -\frac{\ln \xi}{\mu_t}$$

Where numerator is the natural log of a random number and denominator is the cross section data.

The position update and direction cosines block work in a cyclic fashion. Input of direction cosines comes from position update block and vice versa.

- Position update and Direction cosines

Calculates the current position of the photon. Use of Cartesian co-ordinates (x, y, z). Position is updated based on

$$x \leftarrow x + \mu_x s$$
$$y \leftarrow y + \mu_y s$$
$$z \leftarrow z + \mu_z s$$

Where, µx, µy, and µz are the direction cosines. And direction cosines are calculated based on

$$\mu'_x = \frac{\sin\theta(\mu_x\mu_z\cos\varphi - \mu_y\sin\varphi)}{\sqrt{1-\mu_z^2}} + \mu_x\cos\theta$$
$$\mu'_y = \frac{\sin\theta(\mu_y\mu_z\cos\varphi + \mu_x\sin\varphi)}{\sqrt{1-\mu_z^2}} + \mu_y\cos\theta$$
$$\mu'_z = -\sqrt{1-\mu_z^2}\sin\theta\cos\varphi + \mu_z\cos\theta$$

- μ'_x, μ'_y, and μ'_z are the updated direction cosines
- μ_x, μ_y and μ_z are the previous direction cosines
- θ is the photon scattering angle
- φ is the polar angle which varies as,

$$\varphi = 2\pi\xi$$

- Weight Update

The Controller module contains the weight update block, that updates the weight of a photon since it loses the weight at every interaction point. Hence it needs to be updated. And it is updated as,

$$\Delta W = \frac{\mu_a}{\mu_t} W \qquad W \leftarrow W - \Delta W$$

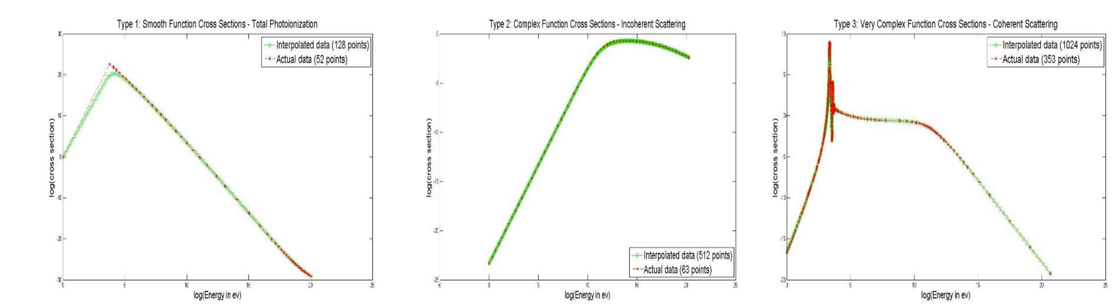
- Controller

The controller module acts as a CPU for the whole system. It sends out request response signals and keeps track of the data flow. This module can directly take the input from the software and accordingly carry out further operation

- Capacitance versus Voltage (C-V): A plot of the capacitance versus voltage will be used to extract the V_{FB}, V_T, t_{ox}, and Q_{ss} parameters.

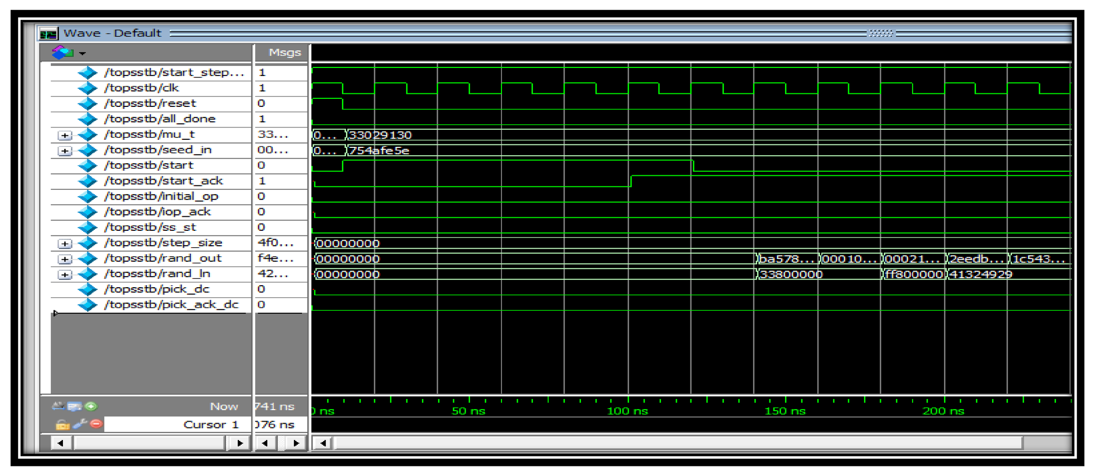
Results

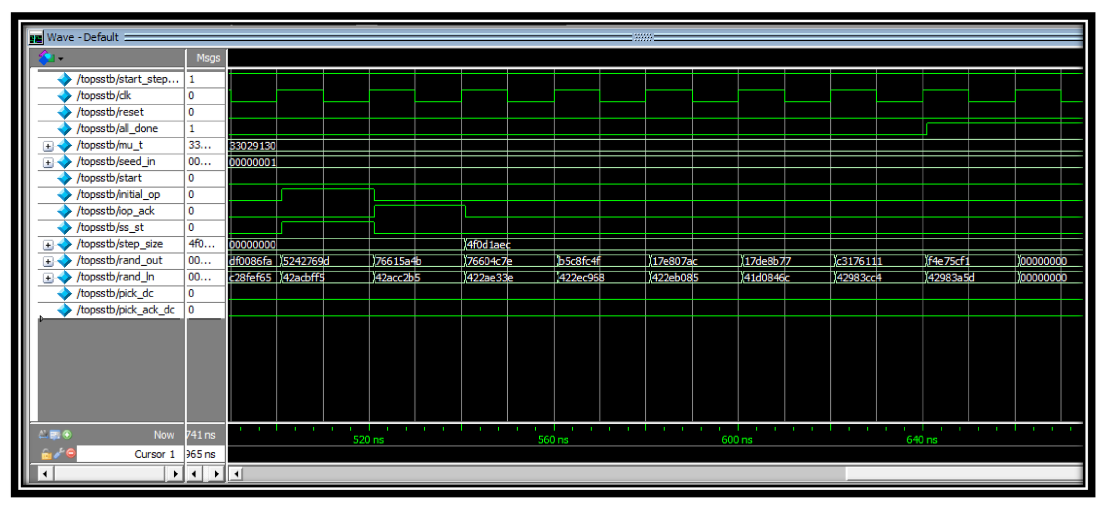
Interpolation data obtained from National Data Library was of three types as shown, Photoionization, Incoherent and Coherent scattering. The output graphs for the data is shown below.

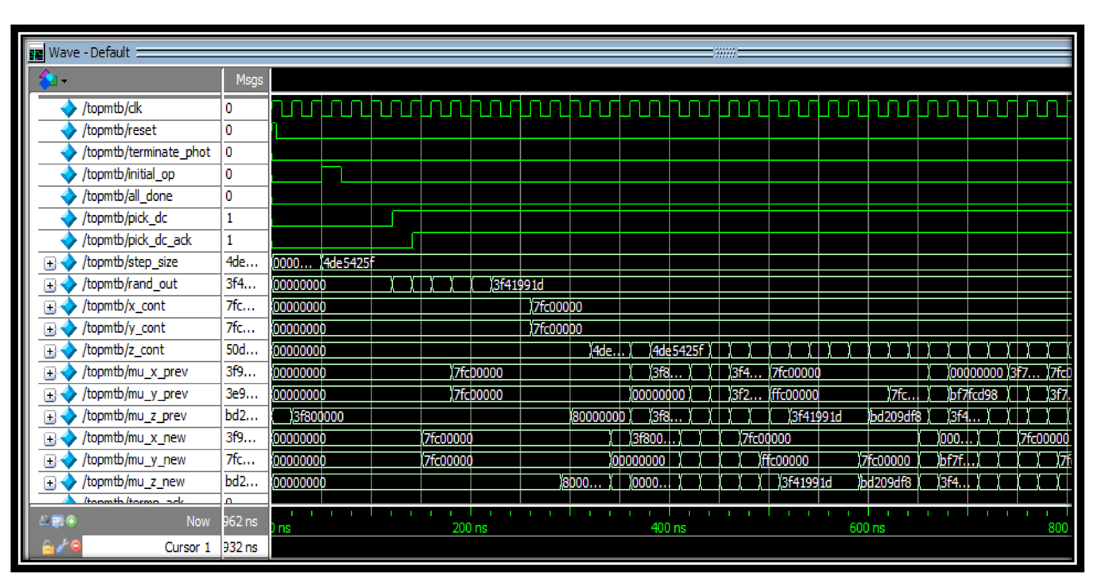


- Verification Results:

The design was successfully verified. The images below show the results of the RTL level verification of the different blocks in the design.

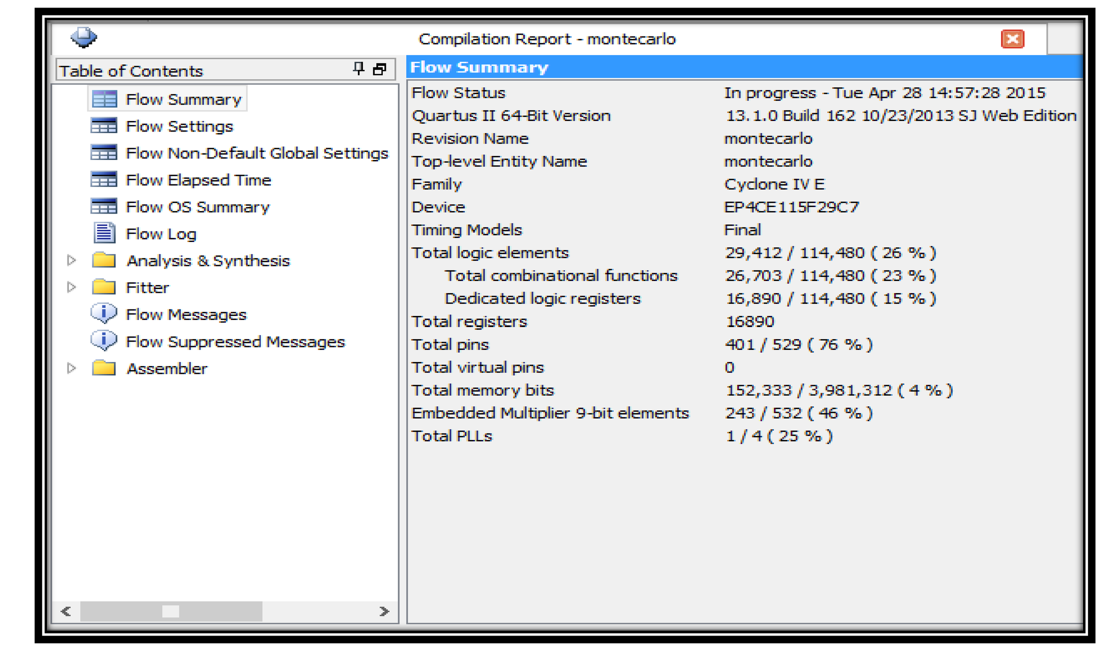






- Synthesis Results:

The designs implemented in the project were synthesizable. As the designs are targeted towards implementing on an FPGA, the tool used was Quartus II for Altera DE2 115 board. The report below shows he details about the logic elements that would be required when implemented on FPGA.



Summary

The Monte Carlo Simulation of Photon transport design was successfully implemented, synthesized and verified. The future work of this project involves downloading the design on an Altera De2-115 board, and more optimizations can be made based on real time applications. A Monte Carlo application can be developed that can be used as a hardware accelerator to speed up the operation. The speed that can be obtained from this will be more than any software only approach to implement the Monte Carlo simulation. The speed comparison can be effectively made between the hardware – software approach and Software only approach. Future work also includes creating Monte Carlo problems and data which can be similar to a real time Monte Carlo Problem.

Key References

[1] Personal Correspondence: Dr. Thuy Le.

[2] Alexander S. Pasciak and John R. Ford, "*A New High Speed Solution for the Evaluation of Monte Carlo Radiation Transport Computations*", IEEE TRANSACTIONS ON NUCLEAR SCIENCE, VOL. 53, NO. 2, APRIL 2006.

[3] Interpolation methods by Paul Bourke

[4] S. Palnitkar, Verilog HDL, 2nd ed. Englewood Cliffs, NJ: Prentice Hall, 2003

[5] M. Matsumoto and T. Nishimura, "Mersenne twister: A 623-dimensionally equidistributed uniform pseudorandom number generator," ACM Trans. Model. Comput. Simul., vol. 8, pp. 3– 30, 1998.

[6] Jason Luu, Keith Redmond1, William Chun Yip Lo2, Paul Chow1, Lothar Lilge2, Jonathan Rose, The Edward S. Rogers Sr. Department of Electrical and Computer Engineering, Department of Medical Biophysics, Ontario Cancer Institute
University of Toronto, Toronto, Ontario, Canada "FPGA-based Monte Carlo Computation of Light Absorption for Photodynamic Cancer Therapy"

[7] Quartus II Handbook Volume 1: Design and Synthesis

[8] Abhiram Prasad Bindigan Hariprasanna, Shreyas Nataraj Rajeswari, "The Study and Implementation of Expensive Operations of Monte-Carlo Photon Transport
Problem on FPGA", San Jose State University.

Acknowledgements

The authors wish to thank Dr. Thuy T Le for his support in helping them build a system involving very complex calculations. Dr. Thuy Le's invaluable guidance helped in studying the project requirements, and also understand the project thoroughly.