Quantum Generators: A Formulation of Computational Models of Multiplication

Poondru Prithvinath Reddy

November 25, 2020

Quantum Generators: A Formulation of Computational Models of Multiplication

Poondru Prithvinath Reddy

ABSTRACT

Quantum Generators is a means of achieving mass food production with short production cycles, and when and where required by means of machines rather than land based farming which has serious limitations. The process for agricultural practices for plant growth in different stages is simulated in a machine with a capacity to produce multiple seeds from one seed input using computational models of multiplication (generating multiple copies of kernel in repetition). In this paper, we provide a formal definition of Quantum Generators and then present concepts related to computational models of multiplication by three different stages i.e. Seed Generation Model, Multiple copies of kernel in repetition and Linking tissues to the kernel. We use simulation to show that we achieve runtime generation with respect to the size of the input space and realize good accuracy to a model. The results suggest that it is possible to achieve an efficient quantum generation.

INTRODUCTION

A **Quantum** (plural quanta) is the minimum amount of any physical entity (physical property) involved in an interaction. On the other hand, **Generators** don't actually create anything instead, they generate quantity prescribed by physical property through multiplication to produce high quality products on a mass scale. The aim of Quantum Generators is to produce multiple seeds from one seed at high seed rate to produce a particular class of food grains from specific class of **seed** on mass scale by means of machine rather than land farming.

The process for agricultural practices include preparation of soil, seed sowing, watering, adding manure and fertilizers, irrigation and harvesting. However, if we create same conditions as soil germination, special watering, fertilizers addition and plant growth in different stages in a machine with a capacity to produce multiple seeds from one seed input using computational models of multiplication(generating multiple copies of kernel in repetition) then we will be closure to achieving mass food production by means of quantum generators(machine generated) rather than traditional land based farming which has very serious limitations such as large space requirements, uncontrolled contaminants, etc. The development of Quantum Generators requires specialized knowledge in many fields including Cell Biology, Nanotechnology, 3D Cellprinting, Computing, Soil germination and initially they may be big occupying significantly large space and subsequently small enough to be placed on roof-tops.

The Quantum Generators help world meet the food needs of a growing population while simultaneously providing opportunities and revenue streams for farmers. This is crucial in order to grow enough food for growing populations without needing to expand farmland into wetlands, forests, or other important natural ecosystems. The Quantum Generators use significantly less space compared to farmland and also results in increased yield per square foot with short production cycles, reduced cost of cultivation besides easing storage and transportation requirements.

In addition, Quantum Generators Could Eliminate Agricultural Losses arising out of Cyclones, Floods, Insects, Pests, Droughts, Poor Harvest, Soil Contamination, Land Degradation, Wild Animals, Hailstorms, etc.

Quantum generators could be used to produce most important *food* crop *like* rice, wheat and maize on a mass scale and on-demand when and where required.

Computers and Smartphones have become part of our lives and Quantum Generators could also become very much part of our routine due to its potential benefits in enhancing food production and generating food on-demand wherever required by bringing critical advanced technologies into the farmland practices.

3D Bioprinting

3D Bioprinting is a form of additive manufacturing that uses cells and other biocompatible materials known as bioinks, to print living structures layer-by-layer which mimic the behavior of natural living systems. Three dimensional bioprinting is the utilization of 3D printing–like techniques to combine cells, growth factors, and biomaterials to fabricate biomedical parts that maximally imitate natural tissue characteristics.

Bioprinting (also known as **3D bioprinting**) is combination of **3D printing** with biomaterials to replicate parts that imitate natural tissues, bones, and blood vessels in the body. It is mainly used in connection with drug research and most recently as cell scaffolds to help repair damaged ligaments and joints. In this paper, we are looking at natural tissues related to food crops like rice, wheat or maize.

The Formulation of Computational Models of Multiplication

In this section, we develop a mathematical formulation for Computational Models of Multiplication. To this end, we first examine the necessary elements of a Quantum

Generating (QG) system. A QG system iteratively improves its current program on the ability to generate “good” future seeds. There are four crucial concepts that should be considered. First, a QG system can be viewed as a number n is a number x such that multiplication of x with its digits is equal to n. The task is to find all seeds of a given number n. Second, is about creating multiple copies of kernel in repetition and the methods for these programs to reap. Third, the QG system requires the hardware (kernels) side of the system and for creating multiple instances, we need to instantiate multiple instances of the kernel (called compute units). Finally, we define a QG system with a set of programs and procedures that can generate programs and an order of seed programs’ ability to improve future programs which are needed for the quantum generation. The following is the sequence of steps that generate programs in repetition and a total order over it.

- Seeds of a Number
- Generating multiple copies of a kernel

    --- Software Emulation

    ---- Hardware Emulation

- System

Seeds (Or Seed Roots) of a Number

A random **seed** (or **seed** state, or just **seed**) is a **number** (or vector) used to initialize a pseudorandom **number** generator. For a **seed** to be used in a pseudorandom **number** generator, it does not need to be random.

A Seed of a number n is a number x such that multiplication of x with its digits is equal to n. The task is to find all seeds of a given number n until no seed exists.

A number X is said to be ‘seed’ of number Y if multiplying X by its digit equates to Y.

For example, 123 is a seed of 738 because 123*1*2*3 is equal to 738.

In this paper, we use seed of number to determine the number of input seeds required for quantum generation.

Generating Multiple Copies of Kernel in Repetition

We should be able to create multiple copies in repetition for the program to reap the benefits of quantum generation.

First we should be able to manage to create one. Repeat this step before exiting and we'll get two. Do that in a loop and you'll get as many as you want. For a given seed of a number, we should repeat this as many times.

Building the QG System requires building both the hardware (kernels) and the software side of the system. In the compute units, we need to create multiple instances of the kernel by instantiating multiple instances of the kernel (called compute units). In order to build target system, we need to link the process for both these software and hardware elements of the system. During the linking stage, there is need to specify the number of instances of kernel referred to as a compute unit.

The build target defines the nature of crop tissue generated by the build process. There are two emulation targets (software and hardware emulation) used for debug and validation purpose.

Software Emulation – The main goal of software emulation is to ensure functional correctness and to partition the application into kernels.

Hardware Emulation – While the software emulation flow is a good measure of functional correctness, it does not guarantee correctness on the execution target. The hardware emulation flow enables the programmer to check the correctness of the logic generated for the custom compute units before deployment where a compute unit is an instantiation of a kernel.

The System

Our objective is to build a target system, we need to generate the cell for the device by running synthesis and implementation on the design. The cell includes custom logic for every Compute unit in the cell container. The generation of custom compute units uses the High-Level synthesis tool, which is the computer unit generator in the application compilation flow. Therefore, it is normal for this step to run for longer period of time than the other steps in the system build flow.

After all compute units have been generated, these units are connected to the infrastructure elements provided by the target device in the solution. The infrastructure elements in a device are all of the memory, control and output data planes which the device is formulated to support an application. The environment combines the custom compute units and the base device infrastructure to generate a cell binary which is used to program the QG device during application execution.

The processing flow of application execution is given as below:-

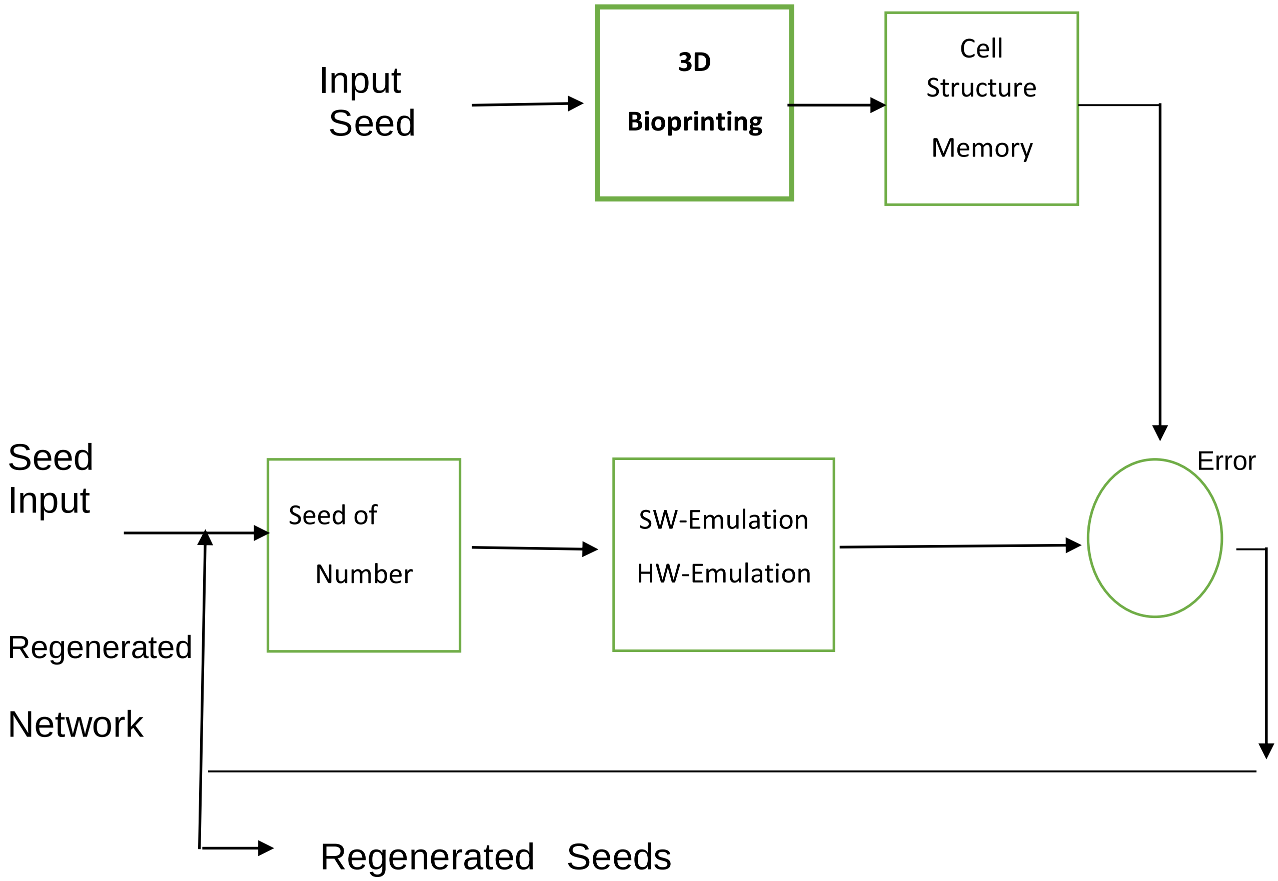

Fig. 1 Process Flow in a Quantum Generator.

The different steps in application are as below:-

1. 3D print a seed and copy its cell structure to memory.
2. Input seed with a seed of a number required.
3. Generate a seed kernel once.
4. Compare the kernel with 3d printed cell
5. If error in seed structure, generate the kernel again.
6. Repeat many times till the seed number is met.

SIMULATION RESULTS

In this paper, we have not implemented 3D bioprinting of crop tissues but taken the results of bioprinted cell structures as representative basis for testing the architecture of Quantum Generating system.

We test the performance of the proposed Computational Models of Multiplication in simulation with randomly generated seed of numbers. For each of the experiments, a fixed number of seed of numbers is chosen. The first seed program is designed to generate crop kernels which is different from other seed programs in loop.

We have tested the synthesis of compute units or creation of multiple kernels over a different subset of seed of numbers and found that as the seed of numbers increases, the emulation of logic in creating multiple instances of compute units is also increasing but not linearly.

Therefore, the simulation results suggest that we achieve good accuracy with respect to the size of the input space and also a possibility of efficient quantum generation if simulated over large seed data and infrastructure.

CONCLUSION AND FUTURE WORKS

Quantum Generators (QG) create new seeds iteratively using the single input seed and the process leads to a phenomenon of generating multiple copies of kernel in repetition. We presented Quantum Generators model by three different stages of growth to achieve multiplication. We used simulation to achieve runtime generation with respect to the input seed space and realize good accuracy. The results suggest a possibility of more comprehensible Quantum Generation by building complexity by combining the results of similar pattern encountered previously.

For future works one may expand the model by different optimized way of creating compute units when generating a new kernel program. Another possible extension is to model the system taking a program as an argument and return a suggested improvement of the given program in repetition. Also we may try to model by adopting convolution technique as in the mathematics of many fields including Artificial Intelligence.

REFERENCE

1. Poondru Prithvinath Reddy : "**The Future of Food Production Is Quantum Generators, Not Land Farming"**

   URL. https://medium.com/