SEQUENCE NTH TERM IDENTIFIER AND DECIMAL TO BINARY CONVERTER WITH MULTIPLEXER IN XILINX ISE

Alyanna Rozel Cruz[1], Alexis Malolos[1], Angelou Cheska Rellon[1]
Rionel Belen Caldo[2]
[1*]*Electronics Engineering Department*
[2]*Electronics and Computer Engineering Department*
Lyceum of the Philippines University – Laguna
Km. 54, National Highway, Makiling, Calamba City, Laguna
[1*]*alyannarozel.cruz@gmail.com*

ABSTRACT

Series or sequences are mathematical tools which are very useful in differential equations and analysis. This idea can be used to represent certain things like approximation of functions. The common types of sequence of numbers are Harmonic, Fibonacci, Arithmetic and Geometric sequence. The concept of this paper is to present an Arithmetic, Geometric, and Fibonacci sequence as a simple mathematical tool with the integration of a decimal to binary converter. Using Verilog code, the proponents develop a program that will identify the nth term of a certain sequence and outputs a binary value which was converted from a decimal value. The algorithm of the program includes a repeated division to obtain the process of conversion and repeated addition in getting the nth term of a certain sequence. Repeated subtraction is also implemented to perform the division operation. RTL designs and simulations of the project design are done in Xilinx ISE using the synthesis tool and ISIM simulation tool respectively.

Keywords: *Arithmetic sequence; Geometric Sequence; Fibonacci Sequence; Decimal; Binary; Verilog Code; Xilinx ISE*

INTRODUCTION

Background of the Study

Fibonacci sequence is a sequence of numbers in which the next term of the sequence is the sum of the two previous numbers. The starting terms of the sequence would either be 0 and 1 or 1 and 1[1].

$$a_n = a_{n-1} + a_{n-2}$$

(equation 1)

Arithmetic sequence is a sequence of numbers where in the terms in the sequence are added to the difference of the two previous terms of the sequence. This difference that is being added to the previous term is constant and is called common difference. The formula in finding the nth term of this sequence is [2]:

$$a_n = a_1 + (n-1)d \qquad \text{(equation 2)}$$

where: an = value of the nth term
a1 = value of the first term
n = number of term asked
d = common difference

Geometric sequence, just like the arithmetic sequence, is the sequence of the numbers where in the terms in the sequence are multiplied to the quotient of the two previous terms of the sequence. This quotient is constant and must be greater than zero and is called common ratio. The formula in finding the nth term of this sequence is[3]:

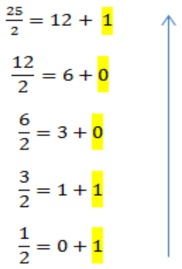
$$a_n = a_1 * r^{n-1} \qquad \text{(equation 3)}$$

where: an = value of the nth term
a1 = value of the first term
n = number of term asked
r = common ratio

Decimal number system is a system of number wherein the number is base 10. It consists of numbers from 0 to 9. Every decimal number has a decimal point (dot). Sometimes decimal point is not included when the numbers are whole numbers [4].

Binary number system is a number system that is base 2. Only 1 and 0 are the number used in binary system. It is used especially in digital systems [5].

In converting decimal number to binary number, the decimal number is to be divided continuously to two until the last quotient is 0. The last quotient is the most significant digit and the first quotient is the least significant digit. It is read from the first quotient to the last quotient [5]. For example, convert 25_{10} to binary.



$25_{10} = 10011_2$

Figure 1 Example of decimal to binary conversion

Problem Statement

There are lots of Verilog codes about Fibonacci sequence that are available online, yet, an integration of this sequence to the arithmetic and geometric approach is not usually presented. The proponents would like to introduce a system through Xilinx ISE Design Suite synthesizable of Verilog code that combines these sequences in addition of a decimal to binary converter. This will provide an output and is likely to be processed in a 4x1 multiplexer.

Objectives of the Study

The main objective of this study is to create and develop a program that will aid the students in solving the nth term of a sequence and converting decimal numbers to binary numbers through Verilog Codes. Specifically, this study aims to:

- create a Fibonacci sequence nth term identifier
- create an Arithmetic sequence nth term identifier
- create a Geometric sequence nth term
- convert decimal numbers to binary numbers
- multiplex the output of the 3 nth term identifier

METHODOLOGY

Xilinx ISE or Verilog Programming Language

Xilinx Inc. is the first semiconductor company to specialize fabless manufacturing and is known for creating the Field Programmable Gate Array (FPGA). This American company produces programmable logic devices and created the Xilinx ISE (Integrated Synthesis Environment) which is use to synthesis and analyze designs.

A powerful tool such as Xilinx ISE will allow the user to compile Verilog code design for the sequence identifier and the converter. When synthesized, a timing analysis will be performed to observe its responses to different conditions. The verification of the diagrams will also be produced for the system.

Proposed System

A synthesizable Verilog code is to be programmed specifically to output the user's desired term, given that the system is to perform operation of the Fibonacci, Arithmetic, Geometric sequence and conversion of decimal to binary. Inputs of the system would be limited depending on the selected sequence or operation. There will be up to 5-bit input for Fibonacci Sequence; 8- bit input for the Arithmetic Sequence; 4-bit input for the Geometric Sequence and up to 16- bit for the decimal to binary converter. All of these would output a value of up to 32- bit considering a selected function that will be assigned to a 2-bit value. This will be applied through a combinational logic circuit design of the multiplexer that switches several inputs to a single output line.

Block Diagram

The figure below shows the full block diagram of the proposed system.

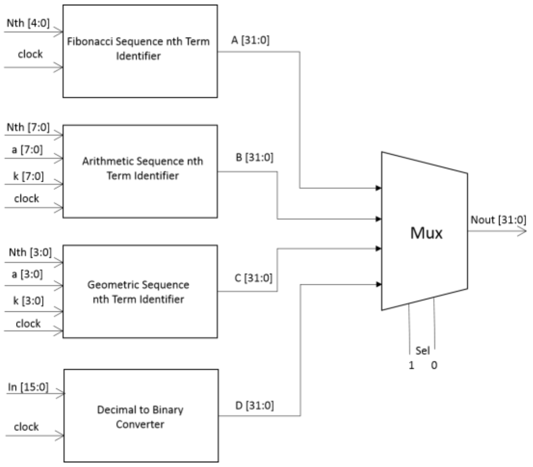

Nth [4:0]
clock
Fibonacci Sequence nth Term Identifier
A [31:0]
Nth [7:0]
a [7:0]
k [7:0]
clock
Arithmetic Sequence nth Term Identifier
B [31:0]
Nth [3:0]
a [3:0]
k [3:0]
clock
Geometric Sequence nth Term Identifier
C [31:0]
In [15:0]
clock
Decimal to Binary Converter
D [31:0]
Mux
Nout [31:0]
Sel
1
0

Figure 2 Block diagram of sequence nth term identifier and decimal to binary converter with multiplexer

IPO Chart

The figure below describes the process flow of the proposed system.

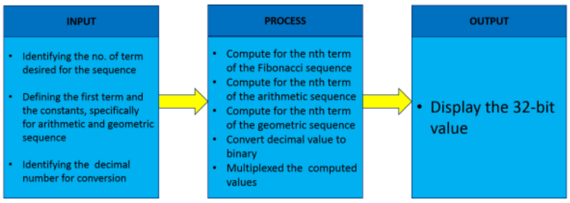

INPUT
Identifying the no. of term desired for the sequence
Defining the first term and the constants, specifically for arithmetic and geometric sequence
Identifying the decimal number for conversion
PROCESS
Compute for the nth term of the Fibonacci sequence
Compute for the nth term of the arithmetic sequence
Compute for the nth term of the geometric sequence
Convert decimal value to binary
Multiplexed the computed values
OUTPUT
Display the 32-bit value

Figure 3 IPO chart of sequence nth term identifier and decimal to binary converter with multiplexer

Input

Identifying the number of the term desired by the user applies to the three sequences namely; Fibonacci, arithmetic and geometric. In addition there is a need to define the value of the first term specifically for the arithmetic and geometric sequence. Also, the constants for the two sequences have to be defined, the difference and ratio for arithmetic and geometric sequence respectively. Identifying the decimal value desired for the conversion is required when the conversion tool is used.

Process

This is composed of the computations of the nth term for the selected sequence and the conversion of the decimal value as desired by the user. The processes are defined by the algorithms used by the proponents. The outputs for the operations are then multiplexed.

Output

The multiplexed output for the operations is displayed as selected by the user. The output is a 32-bit binary number.

Pseudocode

The system is composed of pseudocodes for each operation within. Aside from this, the systems own pseudocode is presented and they are as follows:

a. System Pseudocode

```
Start
Initialize sel and out
Input the value of the sel
If sel = 2’b00, then, identify Fibonacci term;
Else if sel = 2’b01, then identify Arithmetic term;
Else if sel = 2’b10, then identify Geometric term;
Else convert decimal number to binary number
End
```

b. Fibonacci Sequence Pseudocode

```
Start
Initialize N, n0 = 1, n1 = 1, temp = 2
Input value of N
If N = 1, then, print “n0”;
Else if N = 2, then, print “n1”;
Else if temp<n, let n0 = n1 and n1 = n, add n0 and n1 to get n and temp = temp +1
Print “n”;
Else print “n”
End
```

c. Arithmetic Sequence Pseudocode

```
Start
Initialize N, n, a, k, temp = 0
Input value of N, a and k
If temp  < N – 1, print “n”;
Else let a = n, add a and k to get n and temp = temp + 1
Print “n”
End
```

d. Geometric Sequence Pseudocode

```
Start
Initialize N, n, a, k, temp = 0
Input value of N, a and k
If temp  < N – 1, print “n”;
Else let a = n, multiply a and k to get n and temp[ = temp + 1;
Print “n”
End
```

e. Decimal to Binary Pseudoceode

```
Start
Initialize in, out, clk, temp0=0, temp1=0, temp2=0 and q=0
Input value og in
If posedge clock;
If temp0=0, let temp0=temp0+1 and temp1=in;
Else temp1=q;
If temp1=q, q=0;
Else temp1≥2;
If temp1≥2, let temp1=temp1-2 and q=q+1;
Else temp1=1 or temp1=0;
```

Flowchart

Below are the flowcharts of the system and the sequences used.

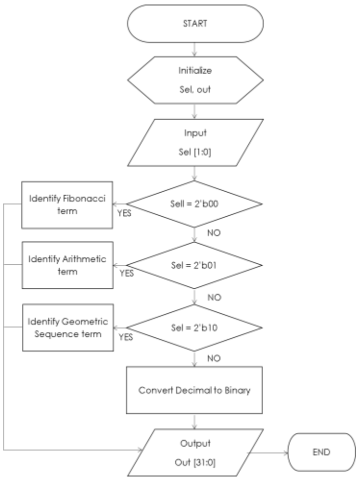


Figure 4 System Flowchart

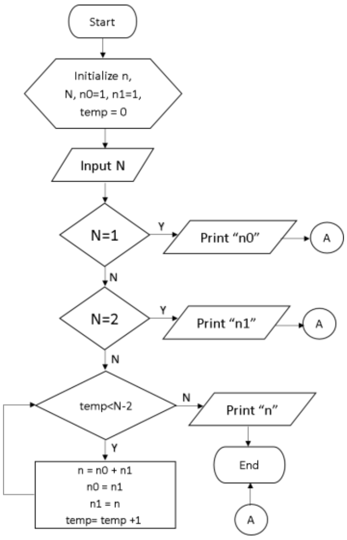


Figure 5 Fibonacci sequence nth term identifier flowchart

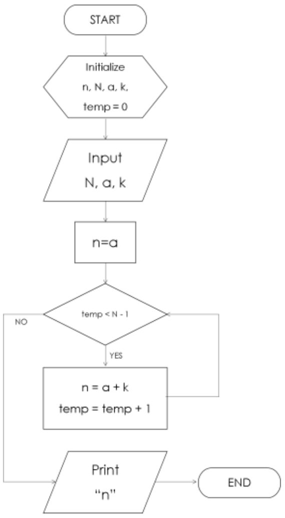


Figure 6 Arithmetic sequence nth term identifier flowchart

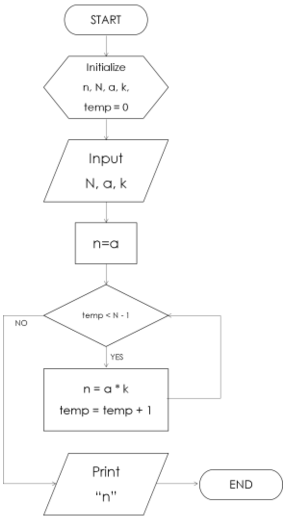


Figure 7 Geometric sequence nth term identifier flowchart

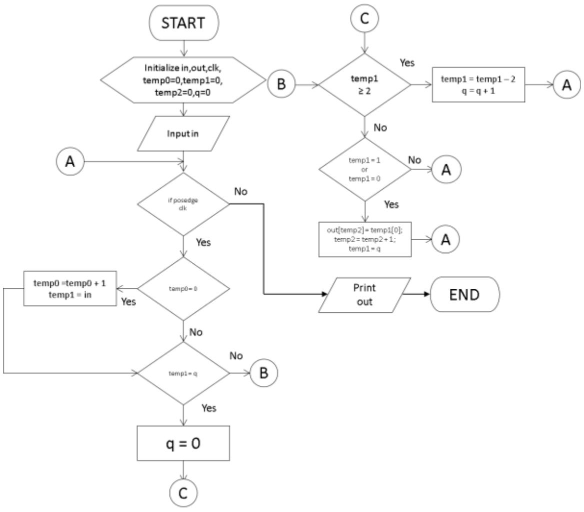


Figure 8 Decimal to binary converter flowchart

RESULTS AND DISCUSSION

Project Description

In this proposed system, the output will depend on what the user needs. The user will choose from the 2 – bit selector code of 00 to 11, and input the required values of the sequence or the decimal number to the test bench. Once simulated, the system will give the output the user needs.

Presented in figure 9 is the high-level RTL schematic diagram of the system. There are three 8-bit inputs labeled as a,k and N which corresponds to; the initial value given by the user, constant for the difference or ratio of the sequence and the nth term of the sequence as desired by the user respectively. A 15-bit input was also included for the decimal to binary converter represented on the RTL as the *in,* a 2-bit input for the selector and the clock. On the right part of the block is the output represented as Fout of 32 bits.

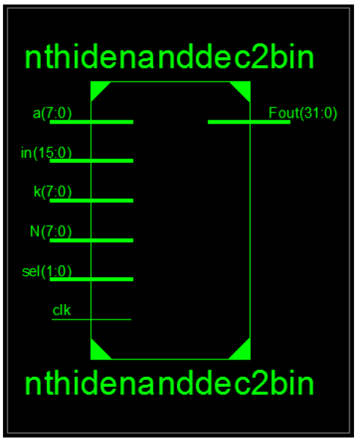


Figure 9 High-level RTL schematic diagram of the system

Breaking the high-level RTL schematic down, a more detailed representation will be seen. Four blocks will be the inputs of the 4x1 multiplexer which were named arith(metic),dec2bin(decimal-to-binary converter), fibo(nacci sequence) and geo(metric sequence). The multiplexer will then operates depending on the input of the user. Same values and parameters will be asked as to the high-level RTL schematic.

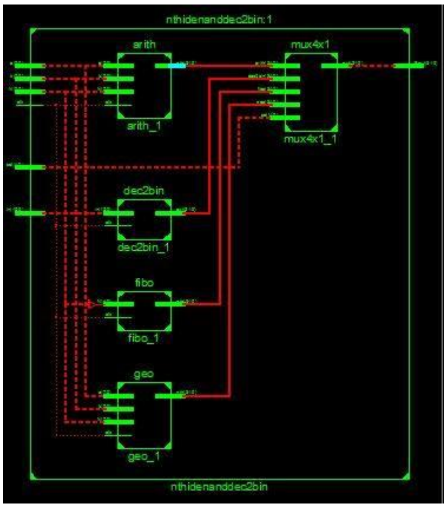


Figure 10 Low-level RTL schematic diagram of the system

Properties of the Project

A hierarchal design is constructed for the sequence nth term identifier and converter with multiplexer. The main fundamental block of the system calls it sub- module to properly execute the program. Sub-modules were mostly based on Verilog's reserved word *always* that, as the word itself implies, executes always and is associated with a sensitive list or a delay. They operate sequentially and would trigger once the clock reaches the positive edge. When a simulation starts, always statements will be queued for execution.

The reserved words *if else* also play an essential role in the execution of the program. This allows the user to set a condition once the other clauses have been falsified and would determine whether to perform a portion of the code or not. The code is executed if the condition is satisfied, else, the other condition would be applied.

The module *nthidenanddec2bin* performs the operation of the 4x1 multiplexer that gives a single output from the multiple inputs. The system's flexibility allows user to choose from four operations that the system could provide. Simulating the Verilog code through test benches verified each operation and the whole system itself.

Functions of the System

Execution of the system would allow a user to choose from four operations that it provides. Three of these will help the user determine the value of the nth term the user desired to identify through following procedures as programmed using the tool Xilinx ISE. The fourth one will convert a given decimal value to binary. In this paper, the proponents have used the Multiplexer's operating function to transform the input to the desired output system.

The integration of these sequences with a converter into a synthesizable Verilog code is a novelty that gives an application of mathematical equations to a system's function of transformation using programs like Xilinx ISE.

Tools and Methodologies of the System

The following are the simulation results of the Verilog code for each input sequence, varied using the selector. The variables used correspond to the following parameters. Fout= Output; sel= system's operation selector; N= nth term required; a=initial value set by the user; k= difference or ratio of the selected sequence and; in= decimal to binary converter output.

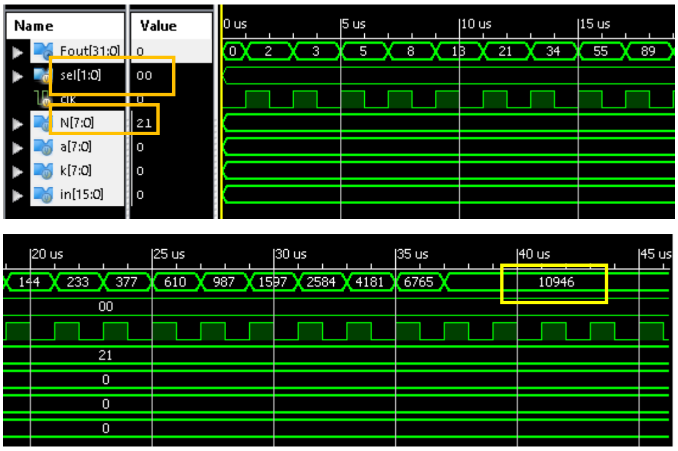
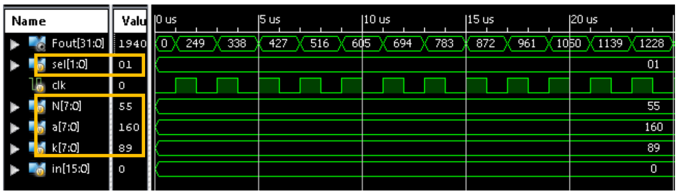


Figure 10 Simulation for Fibonacci sequence identifier



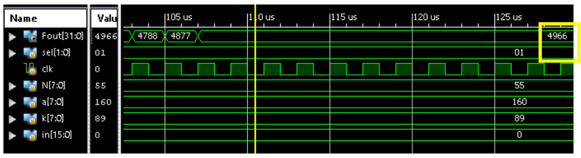


Figure 11 Simulation for arithmetic sequence identifier

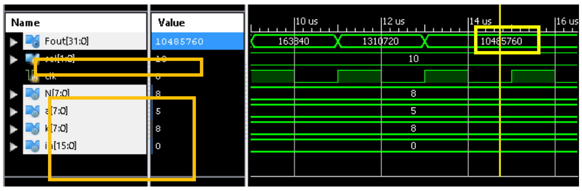


Figure 12 Simulation for geometric sequence identifier

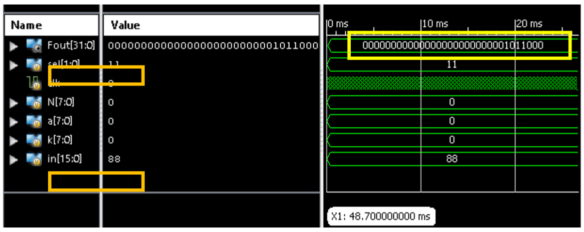


Figure 13 Simulation for decimal to binary converter

CONCLUSION

In this paper, the program – based system sequence nth term identifier and decimal to binary converter with multiplexer has been proposed using Xilinx ISE software. This study shows a simple, effective and reliable program – based system. This paper had introduced the software Xilinx ISE and its algorithm. Its concepts are used in sequence nth identifier and decimal to binary converter. The results the proponents get is credible and reliable. The sequence nth identifier and decimal to binary converter is reliable and credible since the proponents make use of the general equation each sequence has and the standard way of converting decimal to binary using repeated division.

RECOMMENDATIONS

By using a multiplexer synthesized through Verilog code, the system could still vary depending on the user's desire. The future researchers may modify or create a program with more inputs giving more sequences and mathematical operations or other number system converters.

ACKNOWLEDGEMENT

This success and the final outcome of this study have not been possible to come to the present shape without the help and support of the people around us.

With a deep sense of gratitude, we would also like to thank our professor Engr. Rionel Belen Caldo for his complete support and encouragement that he provided. Also, for giving us indispensable advice and information on our study.

We would like to acknowledge the institution, Lyceum of the Philippines - Laguna for helping us in developing our ideas and knowledge.

We also wish to express our sincere appreciation to our parents for the moral and financial support that we need. We thank them for their never ending motivation.

And most of all, we thank God for the wisdom and knowledge that He given us. We owe Him everything for without Him this study would not be possible.

REFERENCES

[1] E. J. Hom (2013) LiveScience homepage on Fibonacci sequence [Online]. Available: http://www.livescience.com/37470-fibonacci-sequence.html

[2] (2015) The MathWords webpage [Online]. Available: http://www.mathwords.com/a/arithmetic_sequence.htm

[3] S. Taylor (2013) AlgebraLab homepage on Geometric sequence Available: http://www.algebralab.org/lessons/lesson.aspx?file=algebra_geoseq.xml

[4] (2015) The Encyclopedia Britannica webpage [Online]. Available: http://www.britannica.com/topic/decimal-number-system

[5] (2015) The TI – Basic Developer webpage [Online]. Available: http://tibasicdev.wikidot.com/binandhex