2/2022

King Abdulaziz University
Faculty of Engineering
Department of Electrical and Computer Engineering

*EE 460: Digital Design II*
*Fibonacci Number Generator*
*Section: CA*

| Name | ID |
|---|---|
| Asaad Waleed Dadoush | 1766585 |

**Instructor**

Dr. Mohammad Awedh

# Contents

# List of Figures

## Introduction:

This project is about designing a Fibonacci number generator that takes an input of n and outputs the Fibonacci number sum like (n-1) and (n-2) Fibonacci numbers, so the first two numbers in the sequence are 0 and 1, in the top-level module there are a control unit and a datapath, the control unit controls the data flow in the datapath, whenever the input is entered that can have a maximum value 24, the go signal will initial, then the control unit will take the nth number and start to send signals to the datapath to start to execute the operations, meanwhile, the counter will start counting until it reaches the specified value, the top-level design includes two outputs dataOut which is the results of the operation, and done signal, the done signal for receiving the results, this cycle can be repeated by including a reset signal.

## Top-level block diagram:

This is the first step for the design starting with identifying all the required singles (inputs, outputs) then making sure all these signals that the design required.

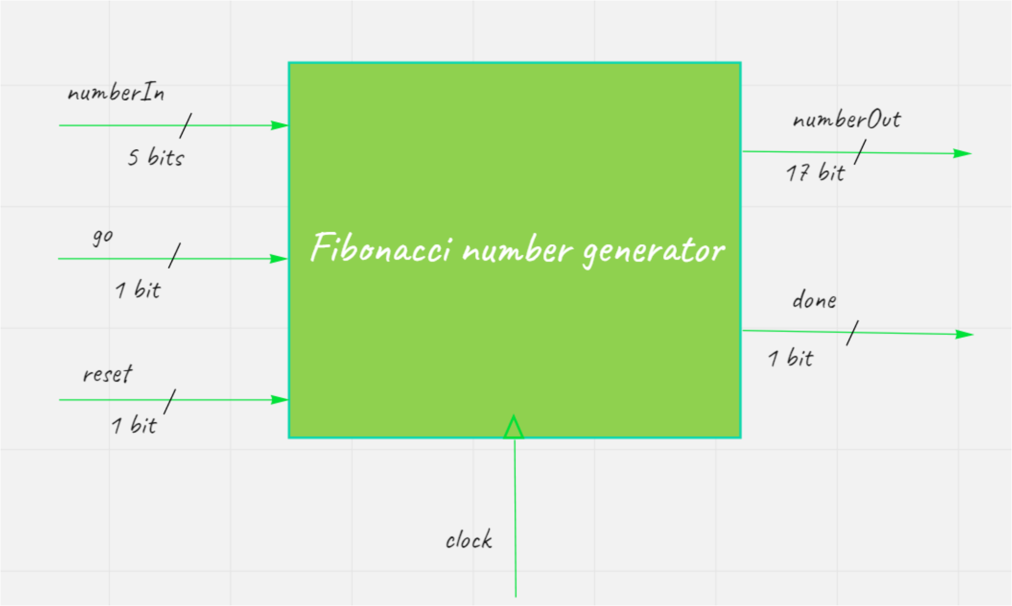


*Figure 1: Top-level block diagram*

## Fibonacci Number Generator:

After making the top-level block now it is time to go deeper and identify the datapath and the control unit and all the signals and connections required for the design.

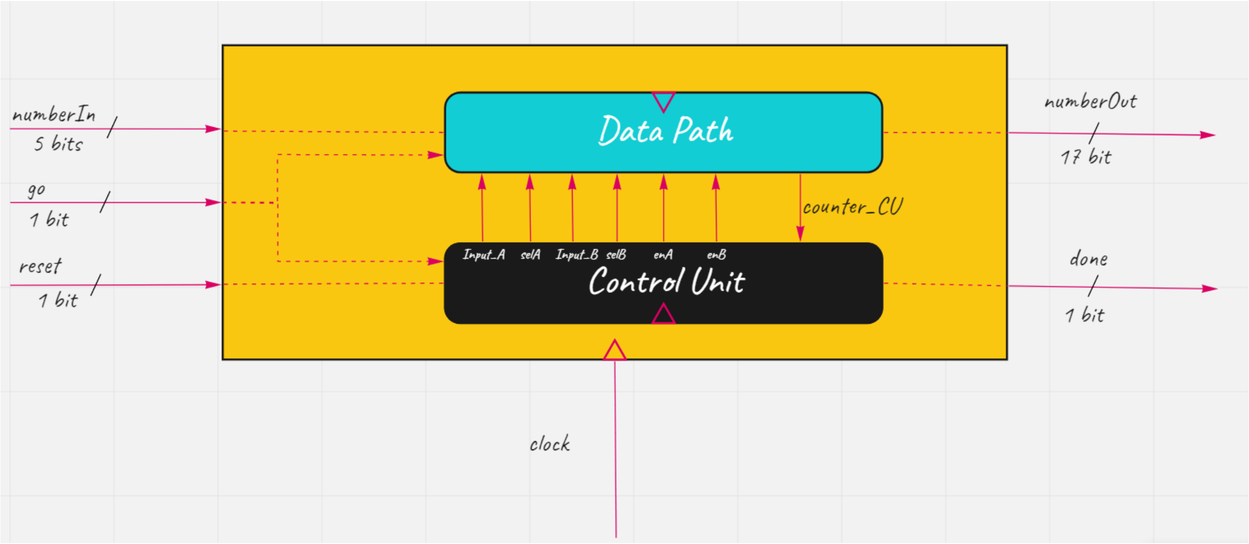


*Figure 2: Fibonacci Number Generator block*

## Inside the block of Fibonacci Number Generator:

In this stage of design, it is time to start defining all the components that are needed for the design, so starting with defining inside each of the datapath and control units blocks as shown below.

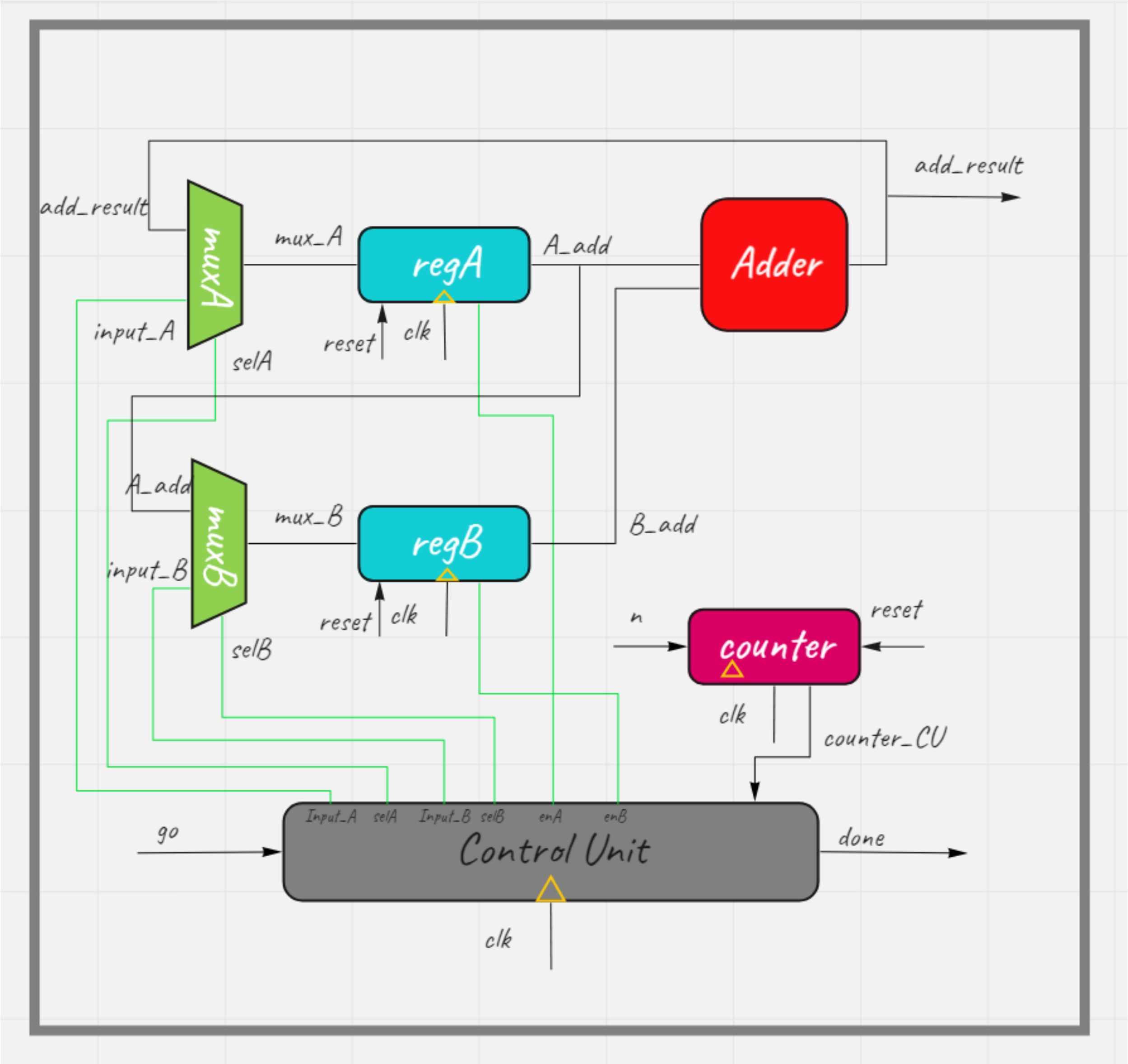


*Figure 3: inside the block of Fibonacci Number Generator*

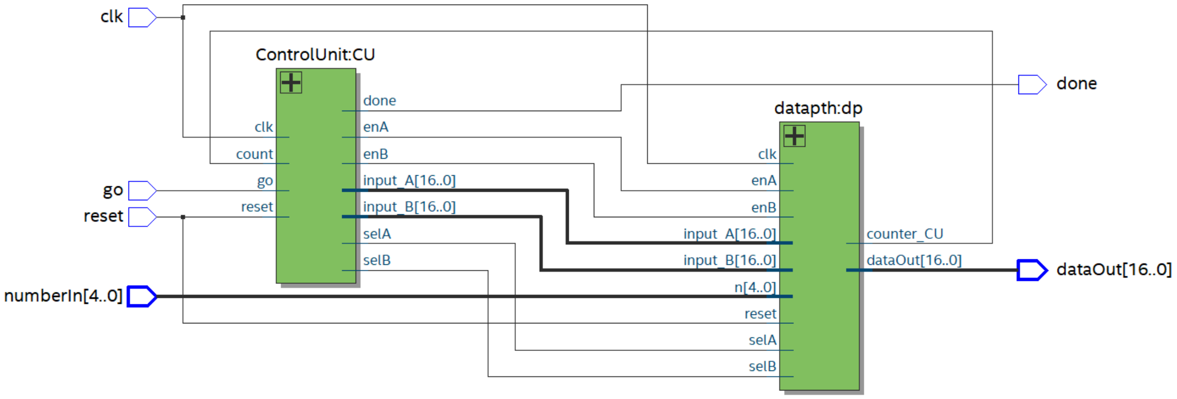


*Figure 4: Fibonacci Number Generator synthesis*

## Datapath:

Let’s have a deeper look inside the datapath and see what are the main computers inside it and how it works, how they get together to build the datapath.

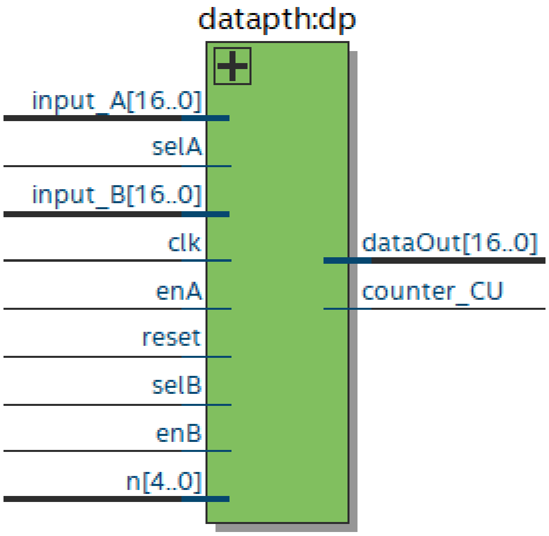


*Figure 5: Datapath diagram*

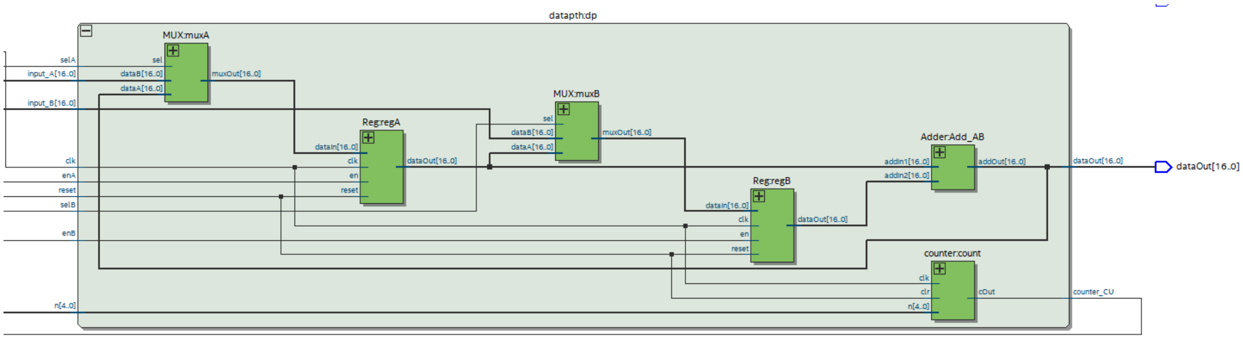


*Figure 6: Datapath synthesis*

As shown above in figure 5 the datapath block diagram and figure 6 inside the block diagram as shown inside the datapath the main components and the signals that come from the control unit to control the data flow in this datapath then the output which is the data out so for the main components there are 2 multiplexers for control the input for the two registers (regA, regB)  then there is an adder for performing the addition operations, at last, the counter for the count the number of addition operations then send a signal to the control unit.

## Control Unit:

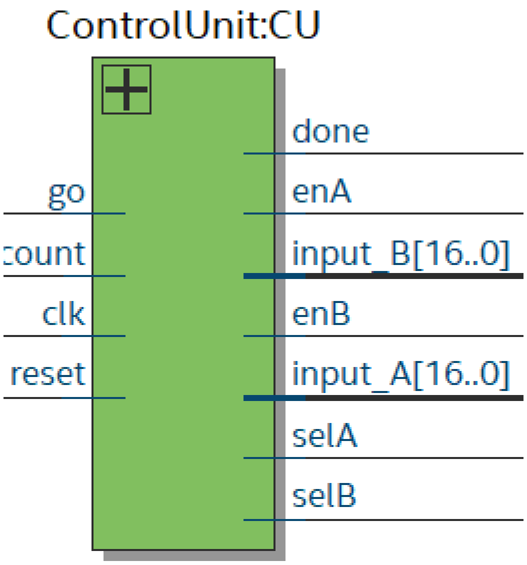


*Figure 7: Control Unit block*

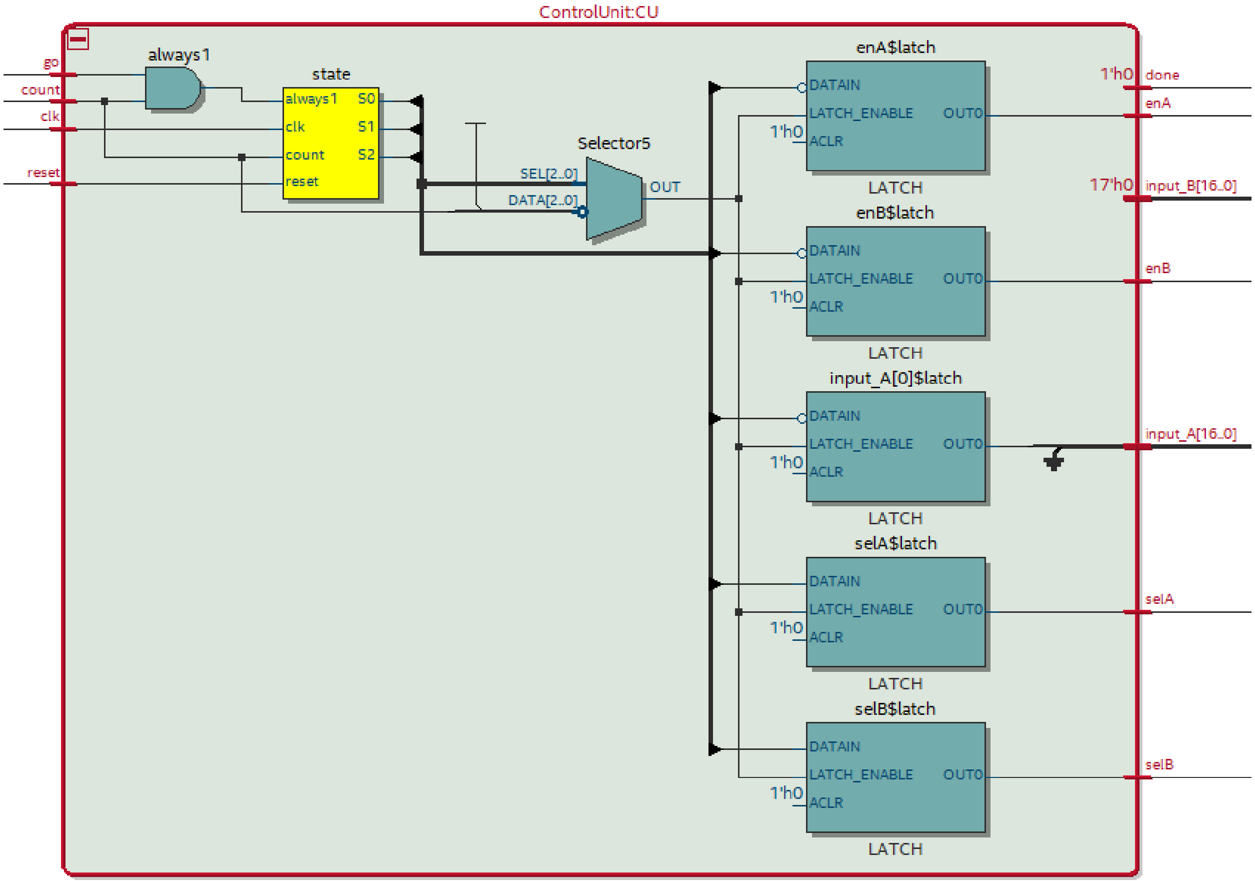


*Figure 8: Control Unit synthesis*

As shown in figure 8 the control unit has 4 inputs (go, count, CLK, reset) and 7 outputs so the inputs signals coming from the user go signal and form the datapath the count signal for the counter, in the control unit the output signals 5 of them to control the datapath and 2 of them for the first two inputs 1 and 0 (Input_A, Input_B).

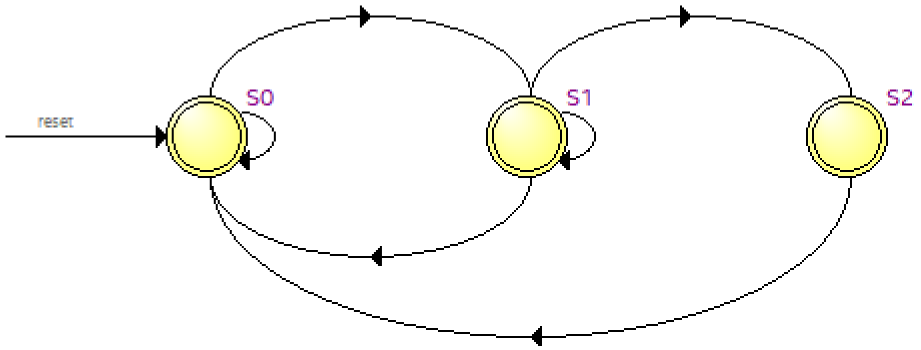
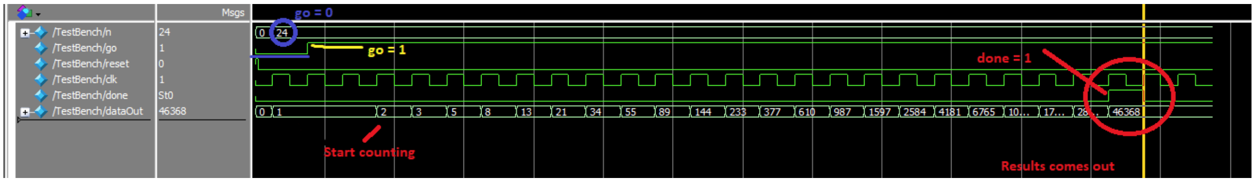


*Figure 9: State diagram*

## Simulations and Testbench:



| | Msgs |
|---|---|
| /TestBench/n | 24 |
| /TestBench/go | 1 |
| /TestBench/reset | 0 |
| /TestBench/clk | 1 |
| /TestBench/done | St0 |
| /TestBench/dataOut | 46368 |

*Figure 10: Testbench Results*

```
always
   #5 clk = ~clk;

initial begin
   // inital the values
   n = 0;
   go = 0;
   reset = 1; // the reset signal to remove the redline in the first number of the outputs in the wave
   clk = 0;
   #1         // just give a small pulse for the reset signal
   reset = 0;


   @(posedge clk);
   n = 24;   // max 24
   go = 0;


   @(posedge clk);
   n = 24;
   go = 1;      // start counting
   #400;
   $stop;
```

*Figure 11: Testbench code*