MSc in Computer Engineering

Electronics and Communication Systems

MAGNITUDE OF A COMPLEX NUMBER

Brief Introduction

Design and implementation of a circuit which approximate the magnitude of a complex number.

Saverio Mosti
s.mosti6@studenti.unipi.it

Summary

Introduction

When we are dealing with complex numbers, we may need to compute the magnitude of a complex number $C = P + JQ$.

$$Magnitude = \sqrt{P^2 + Q^2}$$

Where P is the real part and Q is the imaginary part of the number. But compute the square root can be quite a challenge in Hardware because require a pretty complex numeric algorithm which may need several iterations. Moreover, before the square root we need to compute the power of two of two number of N bits, so we may require two multiplicator at N bit and then a ripple carry adder at 2N bit.
So, the aim of this project is to design a circuit which implement an approximation of the magnitude.

$$Magnitude \cong \max\{|P|, |Q|\} + \frac{1}{2}\min\{|P|, |Q|\} - \frac{1}{16}[\max\{|P|, |Q|\} + \min\{|P|, |Q|\}]$$

Requirements

The logical view of the circuit is shown in the image below:

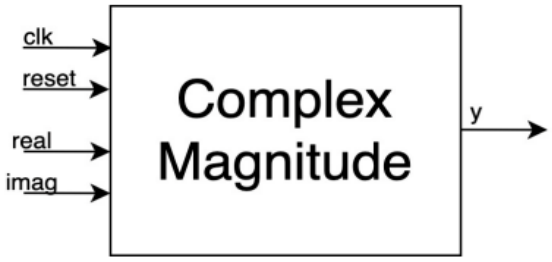

Figure 1: The logical view of the circuit.

The clock and the reset are composed by 1 bit each. Real and Img are two signed integer number represented with 10 bits each. A constraint given by the requirements is that their absolute value is inside the range:

$$|Real|, |Img| \in [0, 2^{N-1} - 1]$$

The secondary aim of this project is to compute the quality of the approximation through the computation of the Mean Square Error (MSE).

Possible Solutions

The formula can be implemented (*and it has been*) with a logical network without any iterative operations because it only need to perform simple arithmetic operations.

Theoretical Model

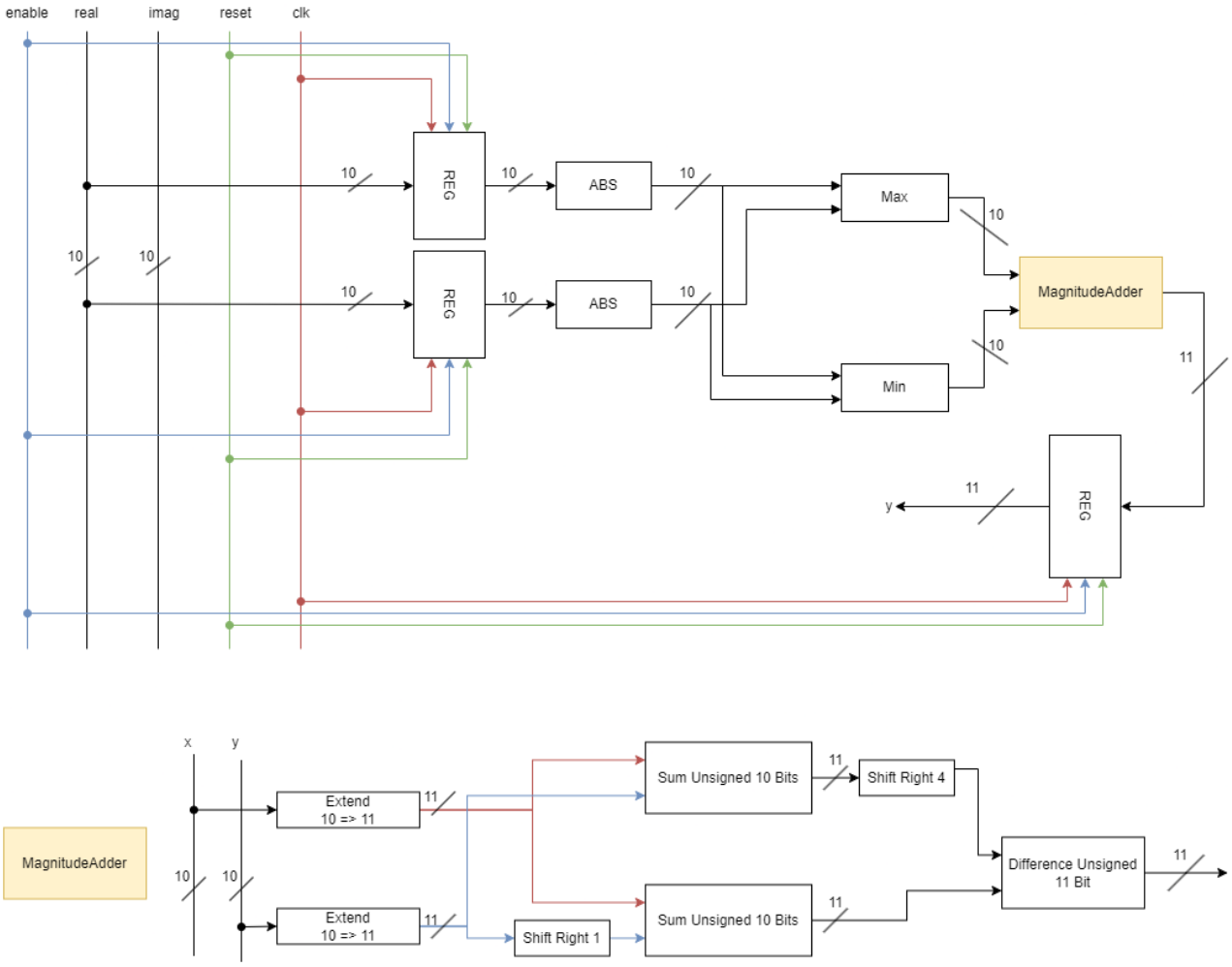

Figure 2- Theoretical design of the circuit.

Before we said that the circuit has been implemented by using only logical components. But to realize a statistic reports with Vivado (*mainly for the critical path*), some register has been added: one for each main input and one for the output.

MagnitudeAdder

The image above shows how the main logic is inside the component MagnitudeAdder; the latter takes as input $\max\{|P|,|Q|\}$, $\min\{|P|,|Q|\}$ and return the result of the approximated formula.
In the project, this component's inputs are both absolute values, so they are unsigned.
At the beginning the two inputs are extended from 10 bits to 11 bits (*the same number of bits of the output*), in this way the two adders will always have the carry out at 0.
In the formula we can see how in two occasion a value is divided for a power of two, instead of use a DIV we simply shift to right. And finally, we got a Difference block, which always return a number in the same number of bits of the inputs, so the result is still on N+1 bits.

VHDL Code

Directory

The structure of the VHDL source code is the following:

- `absolute_value.vhd`
  - Component which computes the absolute value of a signed number.
- `d_flip_flop.vhd`
  - Component which implements a D Flip Flop.
- `max.vhd`
  - Component which takes two unsigned as input and returns the greater one.
- `min.vhd`
  - Component which takes two unsigned as input and returns the littler one.
- `MagnitudeAdder.vhd`
  - Component which implements the MagnitudeAdder net.
- `Magnitude.vhd`
  - The top-level component.

Overall Schematic

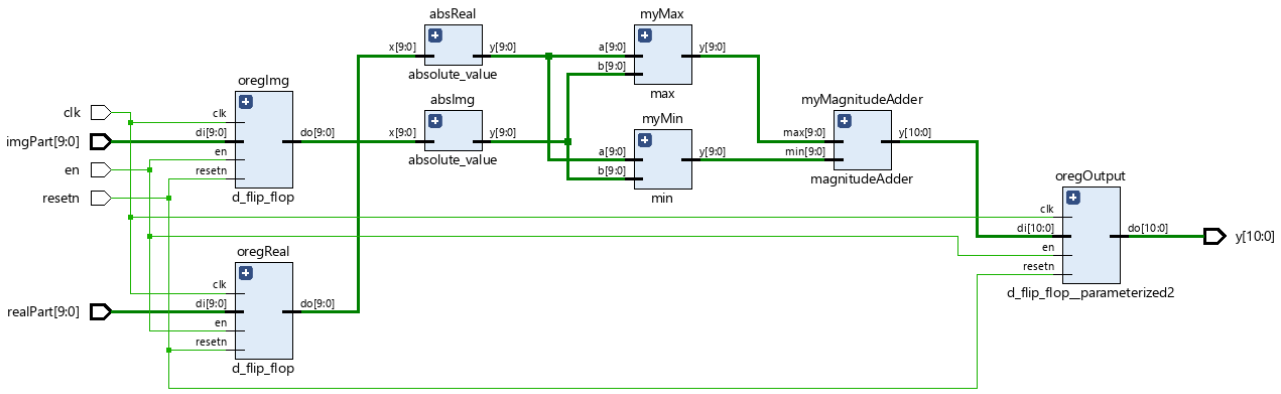

Figure 3- The overall schematic.

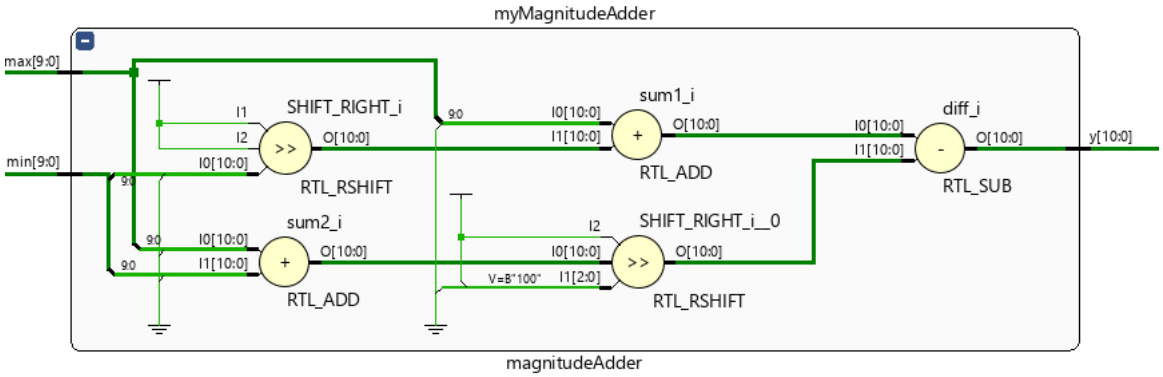

Figure 4 - Schematic of MagnitudeAdder.

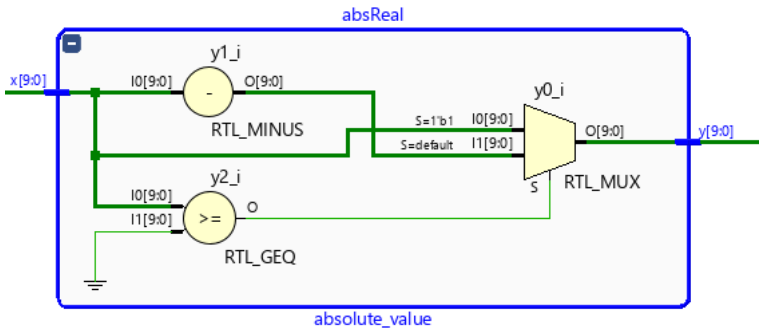

Figure 5 - Schematic of absolute value.

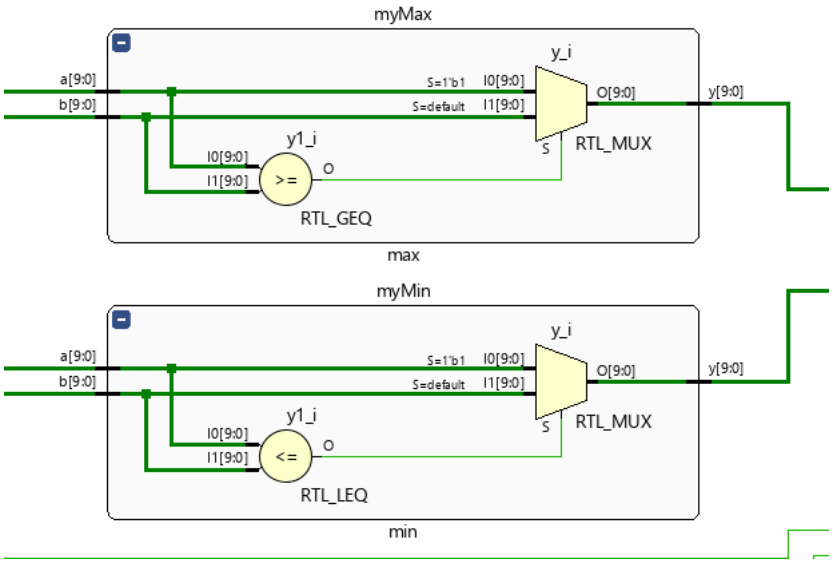

Figure 6- Schematic of min and max.

Testbench and Model Sim

The testbench made to test this implementation is composed in 3 parts:

Limit case test

In this section we test the limit cases (*the bounds of the representation range).*

Normal cases test

In this section we test many cases (*mainly for compute the MSE*). The cases in this section are all the combination with the values inside the range $[-2^6+1,\ 2^6-1]$.

Reset test

In this section we test the reset. So, we put reset to 0 and then to 1 again (*as we can see in the image below, the reset goes down for two clock cycles*).

Waveform

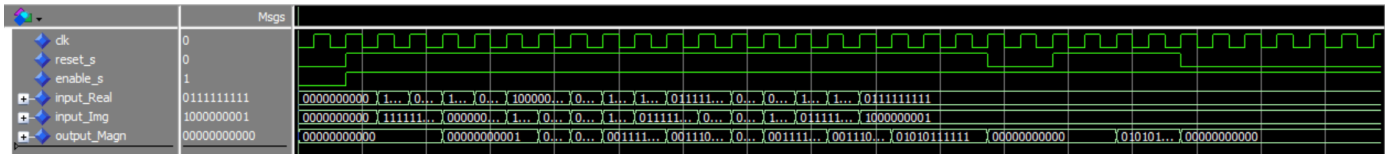

Figure 7- Example of the waveform of the limit case tests (simulation with registers).

Statistics and MSE

The testbench prints the result on the file `Result.csv`, then a MATLAB script reads the file and compute the statistics. Moreover, the MATLAB script verifies that the circuit produces the right results according to the wanted formula.

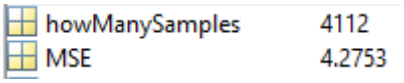

Figure 8- MSE computed by MATLAB.

Must be notified that for make simpler the MATLAB script, the latter has been executed on the results produced by the circuit without the registers (*so the output was update in the same clock instant instead of two clocks forward*). The input numbers (*used for computing the MSE*) are in the order of 2^5 so the quality approximation is not excellent. Maybe the same approximation used with real numbers (*with numbers after the comma*) may produce better results.

Vivado

To use Vivado I created a new project called Magnitude in the Vivado folder.

Open elaborated phase

I choose the same board used during lectures (`xc7z010clg400-1`) and i initially choose a clock period of 10 nanoseconds.

Synthesis phase

The synthesis phase was completed without any error or warning. The result of the timing report was that the timing constraints were largely met so I proceed at the implementation phase.

Implementation phase

The implementation phase was completed without any error or warning. The number of I/O pins of the circuit is (10 + 10 + 11 + 3) = 34 so the project does not need to do a wrapper.

Timing report

Setup

Worst Negative Slack (WNS):	1,023 ns
Total Negative Slack (TNS):	0,000 ns
Number of Failing Endpoints:	0
Total Number of Endpoints:	11

Hold

Worst Hold Slack (WHS):	0,466 ns
Total Hold Slack (THS):	0,000 ns
Number of Failing Endpoints:	0
Total Number of Endpoints:	11

Pulse Width

Worst Pulse Width Slack (WPWS):	4,500 ns
Total Pulse Width Negative Slack (TPWS):	0,000 ns
Number of Failing Endpoints:	0
Total Number of Endpoints:	32

All user specified timing constraints are met.

Figure 9- The timing report.

Critical Paths

Name	Slack ^1	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement
Path 1	1.023	13	12	32	oregImg/d...reg[2]/C	oregOutpu..._reg[9]/D	8.940	4.086	4.854	10.0
Path 2	1.118	13	12	32	oregImg/d...reg[2]/C	oregOutpu...eg[10]/D	8.845	3.991	4.854	10.0
Path 3	1.134	13	12	32	oregImg/d...reg[2]/C	oregOutpu..._reg[8]/D	8.829	3.975	4.854	10.0
Path 4	1.138	12	11	32	oregImg/d...reg[2]/C	oregOutpu..._reg[5]/D	8.826	3.972	4.854	10.0
Path 5	1.159	12	11	32	oregImg/d...reg[2]/C	oregOutpu..._reg[7]/D	8.805	3.951	4.854	10.0
Path 6	1.233	12	11	32	oregImg/d...reg[2]/C	oregOutpu..._reg[6]/D	8.731	3.877	4.854	10.0
Path 7	1.249	12	11	32	oregImg/d...reg[2]/C	oregOutpu..._reg[4]/D	8.715	3.861	4.854	10.0
Path 8	1.404	11	10	32	oregImg/d...reg[2]/C	oregOutpu..._reg[3]/D	8.560	3.528	5.032	10.0
Path 9	1.464	11	10	32	oregImg/d...reg[2]/C	oregOutpu..._reg[2]/D	8.500	3.468	5.032	10.0
Path 10	1.817	11	10	32	oregImg/d...reg[2]/C	oregOutpu..._reg[1]/D	8.147	3.115	5.032	10.0

Figure 10- The worst the paths of the circuit

So, from the latter we obtain the maximum clock frequency:

$$f_{max} = \frac{1}{T_{clk} - WNS} = 111{,}396\ MHz$$

Power Consumption

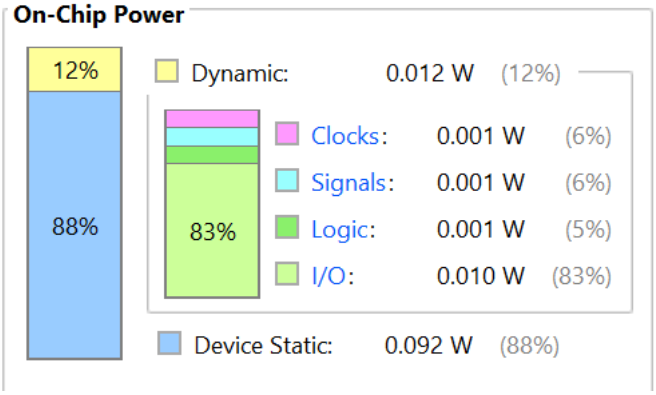

Figure 11- Power consumption report.

Conclusions

In conclusion, the journey of designing and implementing this circuit has been both fruitful and enlightening. Beginning with the conceptualization phase, where the theoretical circuit model was drawn, provided a solid foundation for understanding the intricate requirements and functionalities of the circuit. This initial step not only guided the selection of appropriate components but also laid the groundwork for the subsequent stages of the design process.

Moving forward, the transition to writing VHDL code was a pivotal moment in bringing the theoretical model to life. Through meticulous coding and attention to detail, the VHDL implementation encapsulated the essence of the circuit's functionality, paving the way for rigorous testing and verification.

The utilization of ModelSim for verification purposes proved to be invaluable, as it enabled thorough examination of the VHDL code's correctness through waveform analysis and MSE computation. This step was essential in identifying and rectifying any potential issues or discrepancies, ensuring the integrity and reliability of the final design.

Subsequently, the integration of Vivado into the design pipeline facilitated the synthesis and implementation of the circuit, culminating in the generation of comprehensive reports encompassing timing and power consumption analyses. These reports provided invaluable insights into the performance characteristics of the circuit, allowing for optimization and refinement where necessary.