Delay Insensitive 4-bit Carry lookahead Adder

Report Submitted in partial fullfilment of course
on VLSI Design Lab (EC364)

Under the guidance of
Dr.Ramesh Kini
Mr.Kiran Kumar Lad

Submitted by:
Santosh Prasad 11EC85
Shah Ankit Parag 11EC86
Shubham Yadav 11EC92

Department Of Electronics & Communication Engineering
National Institute Of Technology Karnataka, Surathkal
Srinivasnagar 575025 Karnataka India

May 2014

Contents

Introduction

Adder as we know is used widely in a computer as adding data is an important task in a processor.Arithmetic operations such as addition, subtraction, multiplication and division are widely used and play an important role in various digital systems such as digital signal processor (DSP) architecture, microprocessor and microcontroller and data process unit. The speed of execution is the most important factor in fast computing devices to meet our performance expectations. The simplest binary adder is ripple carry adder.It is easy to be understood and implemented.To reduce the delay caused by the eect of carry propagation through the ripple-carry adder, we can attempt to evaluate quickly for each stage whether the carry-in from the previous stage will have a value 0 or 1. If a correct evaluation can be made in a relatively short time, then the performance of the complete adder will be improved. Such concept is commonly called carry-lookahead. A more complex binary adder is carry lookahead adder (abbreviated as CLA) [3, 4]. It uses the same carry lookahead circuits to construct the higher-bit CLA recursively. It is widely used due to its superior performance over ripple carry adder. Traditional CLA is constructed by XOR, AND, and OR gates. The proposed circuit uses NAND gates to replace the AND and NOT gates in CLA, it can decrease the cost of CLA and increase the speed of CLA.
Carry Look Ahead (CLA) Addder (also known as Carry Look Ahead Generator) is one of the digital circuits used to implement addition of binary numbers. It is an improvement over 'Ripple carry adder' circuit. In Ripple Carry adders, carry propagation time is the major speed limiting factor as it works on the basic mechanism to generate carries as we generally do while adding two numbers using pen and paper. A ripple carry adder may be supposed to be built of a series of 1-bit adders (generally known as a full adder in digital electronics). Thus, the speed of ripple carry adder is a direct function of number of bits. On the other hand, Carry Look Ahead adder solves this problem by calculating carry signals in advance based upon input bits and does not wait for the input signal to propagate through different adder stages. Hence, it implements the adder with reduced delay at the cost of more area

MCLA vs SpCLA

The proposed **modified carry lookahead adder** (abbreviated as MCLA) is similar to CLA in basic construction. Hence, it also contains arithmetic adder circuit and carry lookahead circuit. The designed construction of carry lookahead circuit in MCLA is similar to CLA. With using the basic model of MCLA, it can use the carry lookahead circuit recursively to implement the higher-bit MCLA. In order to be analyzed in mathematics, a K -bit CLA model is defined in this paper. Where K is the number of bit consisted in each level of CLA and m is the level of carry lookahead circuit used in CLA.

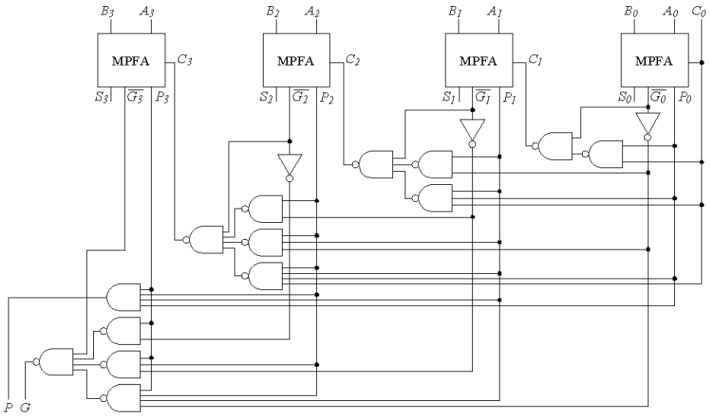

Figure 1: Block Diagram of 4-bit CLA

Let K=4 and m=1, the proposed 4-bit
simplified carry lookahead adder (abbreviated as SpCLA) is shown in Figure 1. It contains two parts. The one part is arithmetic adder circuit and the other one is carry lookahead circuit. In order to be used as the first level of arithmetic adder circuit in the proposed SpCLA, the part of this new full adder called metamorphosis of partial full adder (abbreviated as MPFA), as shown in Figure 2, is used in SpCLA. In carry lookahead circuit of 4-bit SpCLA, all of the components are implemented with NAND gates except for the outputs of P and G , which are implemented with AND gates. Since the output signal of MPFA is i i G implemented with NAND gates, it is faster than the G of PFA implemented with AND gate. Although the proposed SpCLA can be implemented via the same proposed carry lookahead circuit shown in Figures 1, it is not the simplest circuit. The method to simplify this circuit is using a NAND gate and a NOT gate to replace the AND gate of the output bit G in the 4m-1-bit (previous level) SpCLA circuit when m is greater than 1, and then cancel its NOT gate with the NOT gate in the 4m-bit (present level) SpCLA. The new circuit is named as MCLA. Let K= 4. For example, there are three NOT gates in the carry lookahead circuit of 16-bit SpCLA. If we move back the three NOT gates into 4- bit SpCLA and simplify it with another four NOT gates in the proposed carry lookahead circuit of 16-bit SpCLA, we can derive the simplest circuits of 4-bit and 16-bit MCLA, as shown in Figures 4 and 5, respectively.

Design of Standard Cells in Magic

In this project we have designed the standard cells in magic after verifying the functionality on ngspice

Inverter

In this case we have designed a normal inverter with basic CMOS Gates and it is a single line diffusion for inverter.

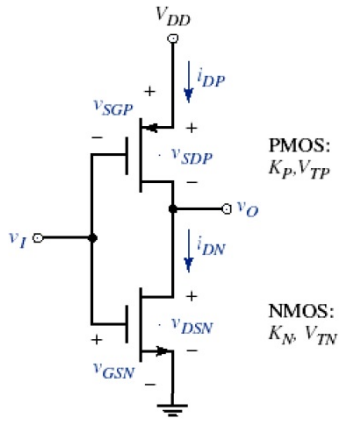

Figure 2: CMOS based Inverter

Area Specifications

Area = **12x80 unitsquare**

Thus the size was kept as 80 units since in the design the maximum length of the largest gate is XOR and based on that the sizing was decided for the standard cells

Delay Specifications

Rise Time Delay= 0.8 ns
Fall Time Delay= 0.68 ns

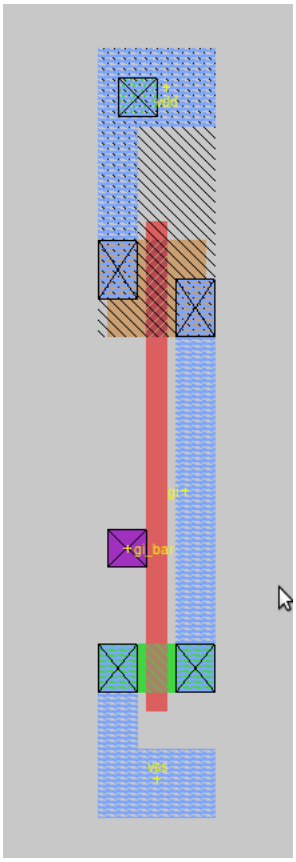

Figure 3: Layout on Magic for NOT/Inverter Gate

XOR Gate

A power efficient circuit topology is proposed to implement a low-voltage CMOS 2-input pass-transistor XOR gate. This design aims to minimize power dissipation and reduce transistor count while at the same time reducing the propagation delay. The XOR gate utilizes six transistors to achieve a compact circuit design and we have used this sort of configuration in order to get minimum gate delay. We have been able to get a Full Voltage Swing to Swing Output and a minimum gate delay of about **1.5 nanoseconds** during rise time and **0.84 nanoseconds** during fall time

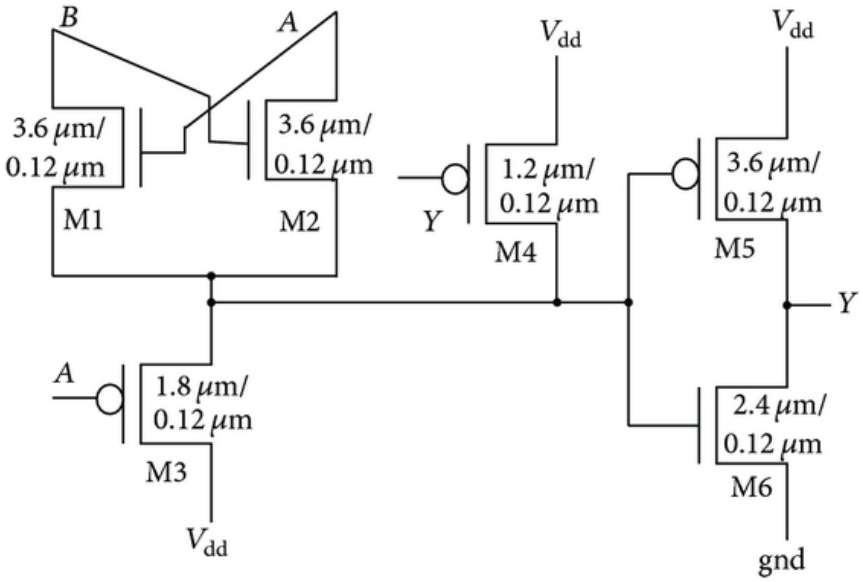

Figure 4: Circuit Diagram for XOR Gate

Thus we see that XOR gate gives a full swing to swing output and the following are the AREA and DELAY Specifications

Area Specifications

Area = **42x80 unitsquare**

Thus the size was kept as 80 units since in the design the maximum length of the largest gate is XOR and based on that the sizing was decided for the standard cells

Delay Specifications

Rise Time Delay= 1.5 ns
Fall Time Delay= 0.84 ns

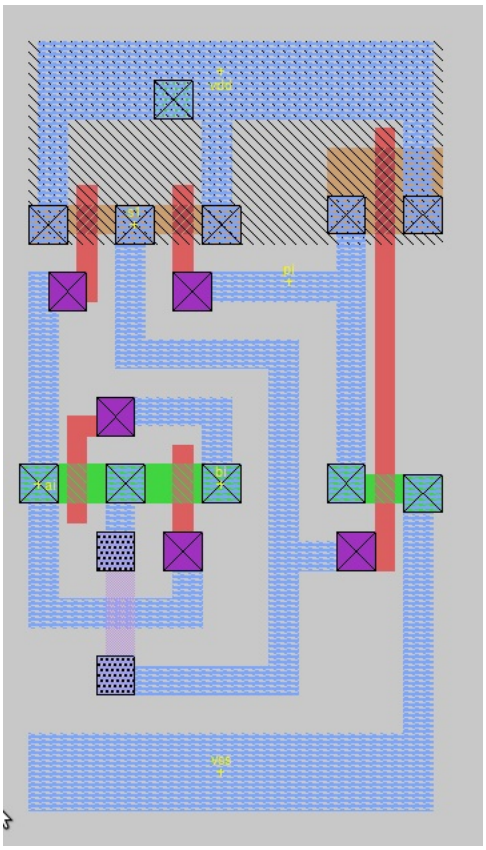

Figure 5: Layout on Magic for XOR Gate

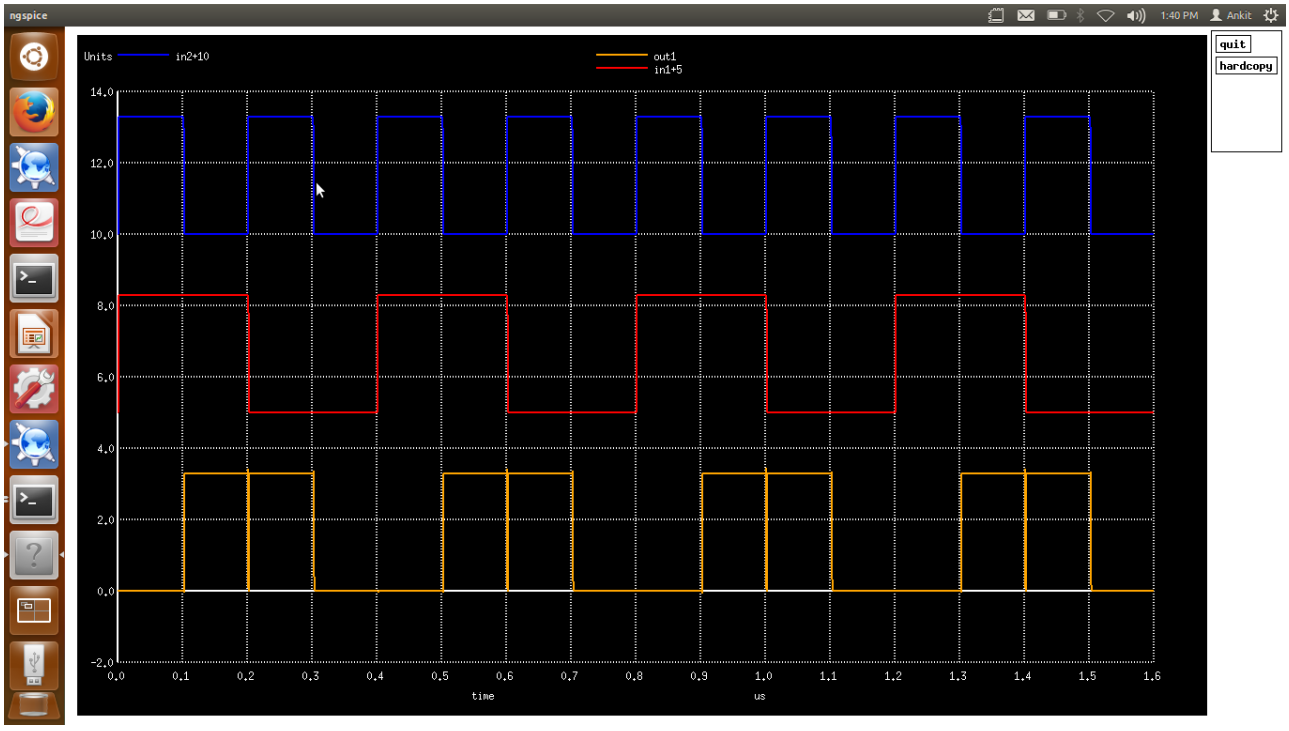

Figure 6: Simulation Results for XOR Gate

CMOS based 2 input NAND Gate

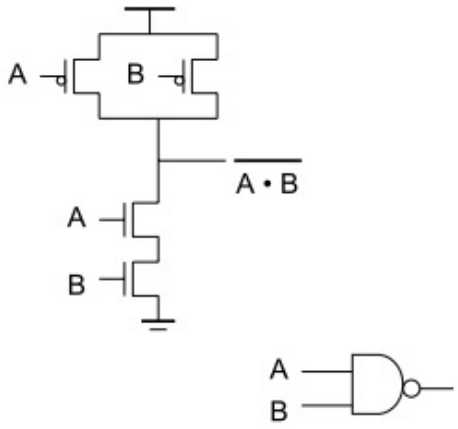

Figure 7: CMOS based 2 input NAND Gate design

Area Specifications

Area = **21x80 unitsquare**

Thus the size was kept as 80 units since in the design the maximum length of the largest gate is XOR and based on that the sizing was decided for the standard cells

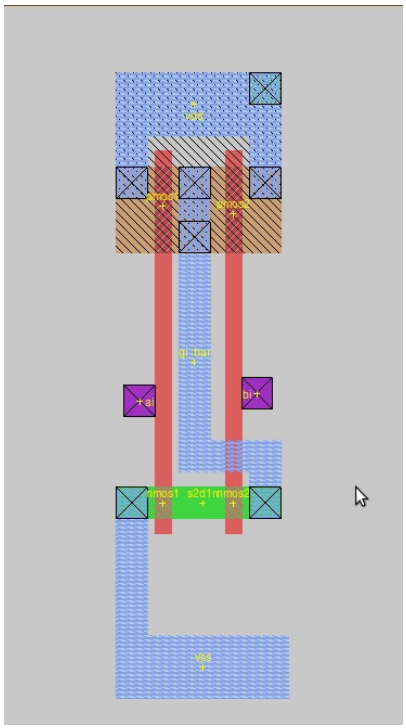

Figure 8: 2 Input Nand Layout

Delay Specifications

Rise Time Delay= 1.12 ns
Fall Time Delay= 1.2 ns

CMOS based 3 input NAND Gate

Area Specifications

Area = **31x80 unitsquare**

Thus the size was kept as 80 units since in the design the maximum length of the largest gate is XOR and based on that the sizing was decided for the standard cells

Delay Specifications

Rise Time Delay= 1.32 ns
Fall Time Delay= 1.5 ns

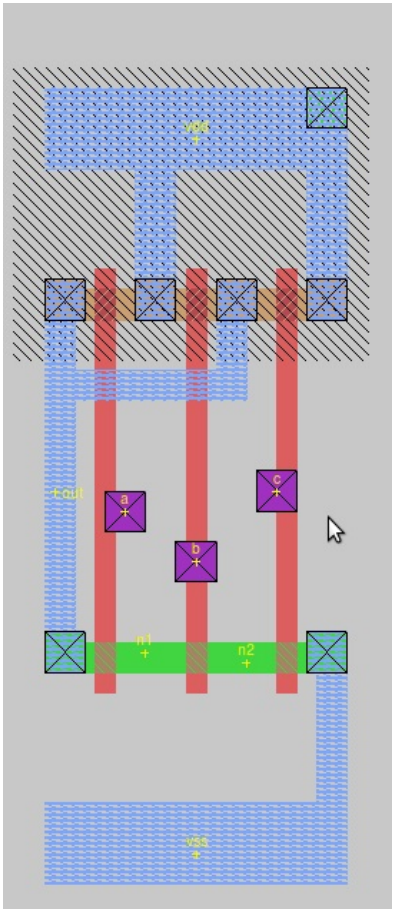

Figure 9: 3 input Nand Layout

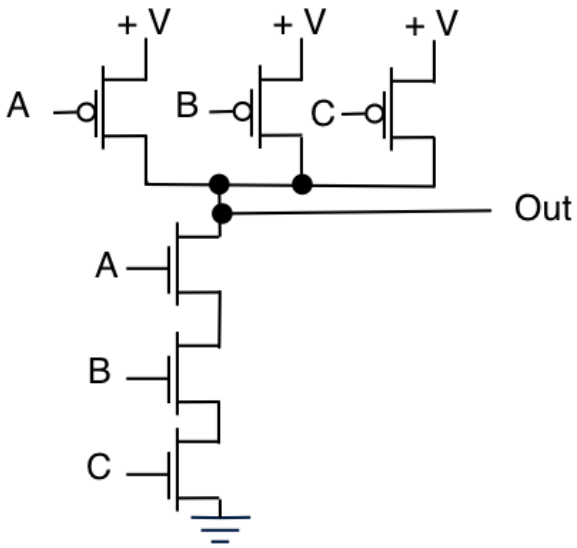

Figure 10: CMOS based 3 input Nand Gate

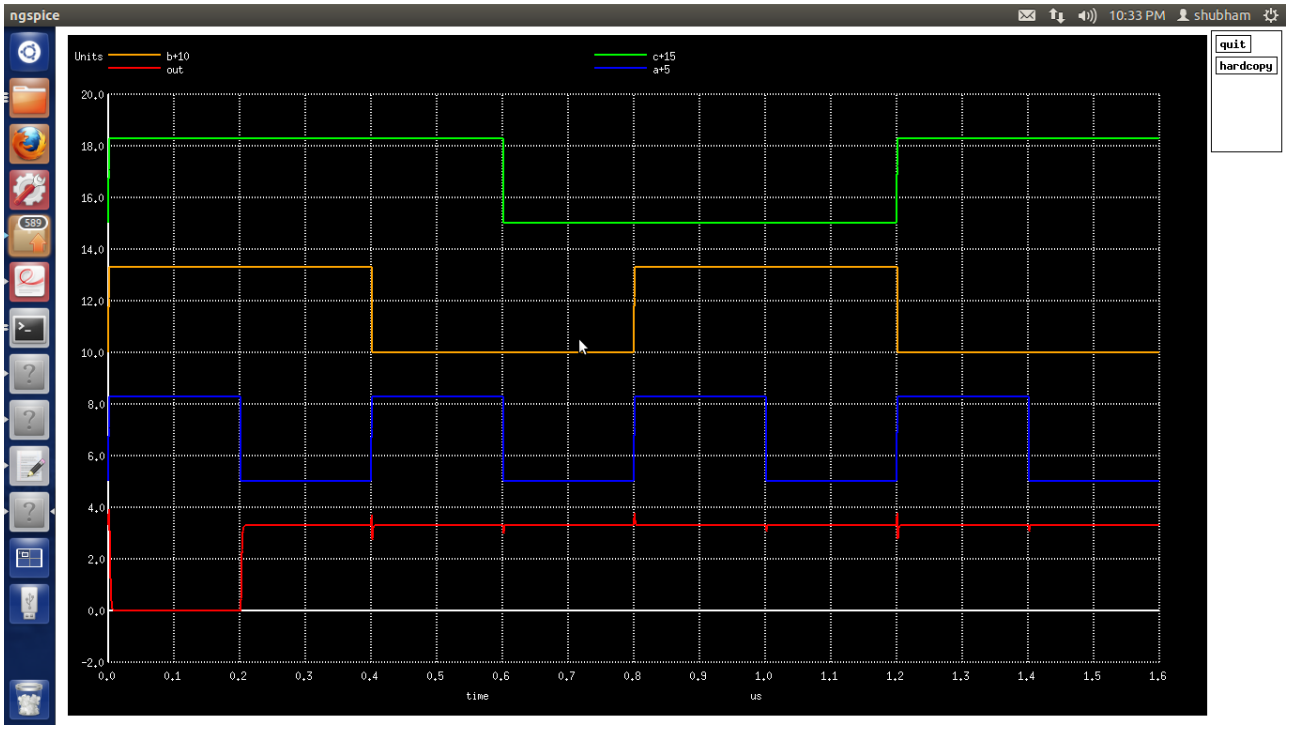

Figure 11: 3 input Nand Simulation Results

CMOS based 4 input NAND Gate

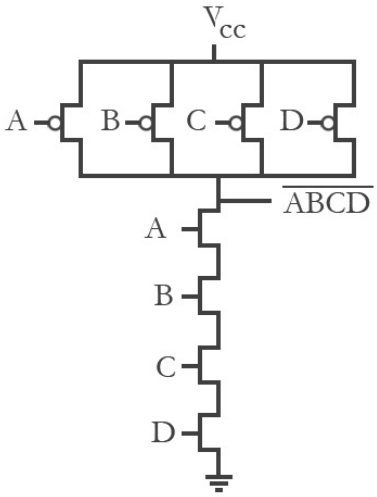

Figure 12: 4 input Nand layout

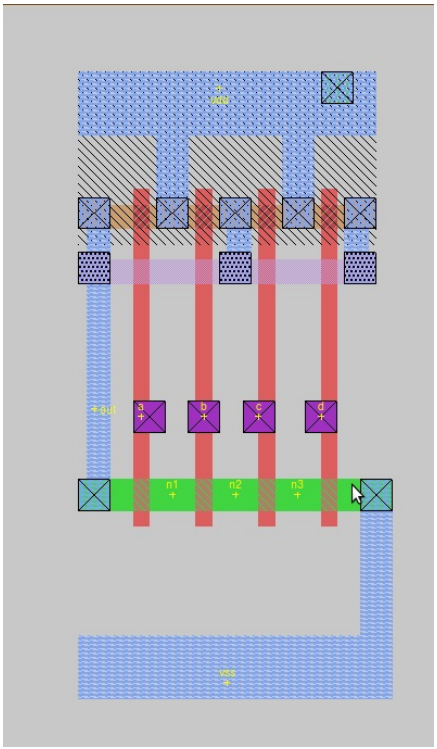

Figure 13: 4 input Nand layout

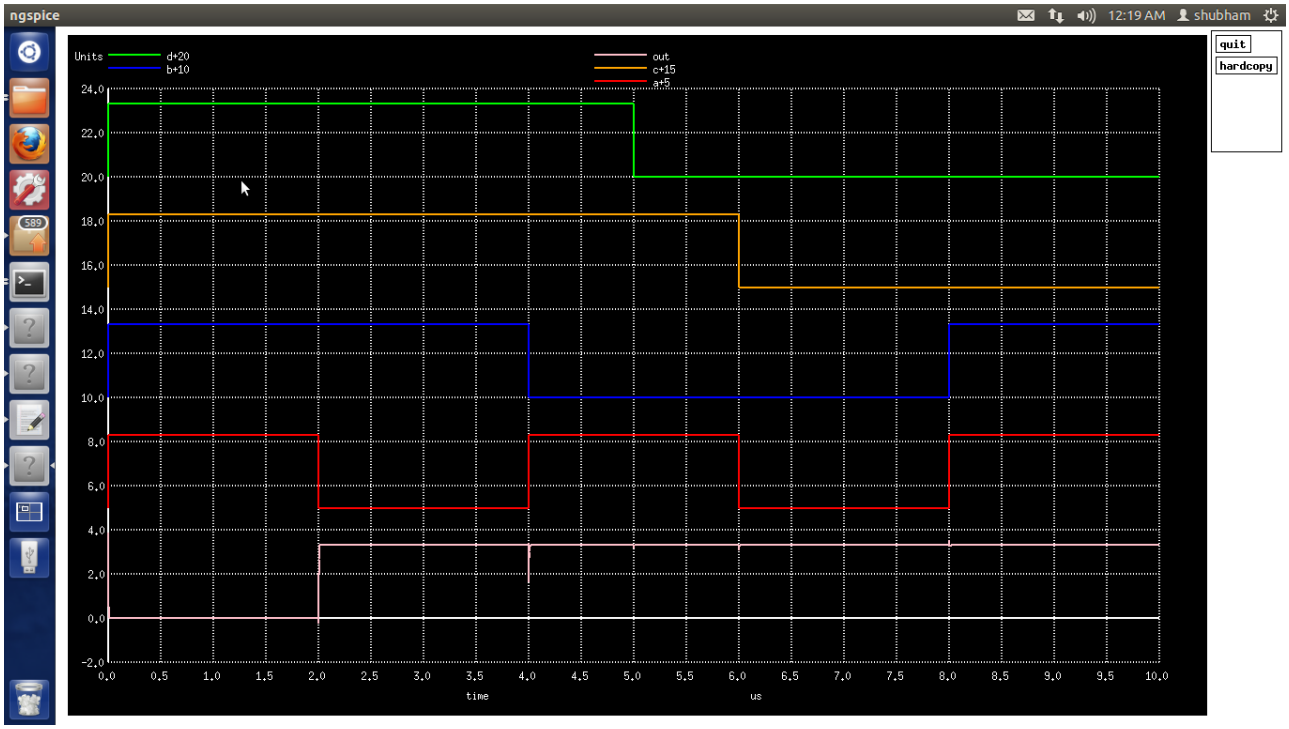

Figure 14: 4 input Nand Simulation Results

Thus we see that 4 input NAND gate gives a full swing to swing output and the following are the AREA and DELAY Specifications

Area Specifications

Area = **38x80 unitsquare**

Thus the size was kept as 80 units since in the design the maximum length of the largest gate is XOR and based on that the sizing was decided for the standard cells

Delay Specifications

Rise Time Delay= 1.3 ns
Fall Time Delay= 1.84 ns

Block Diagrams

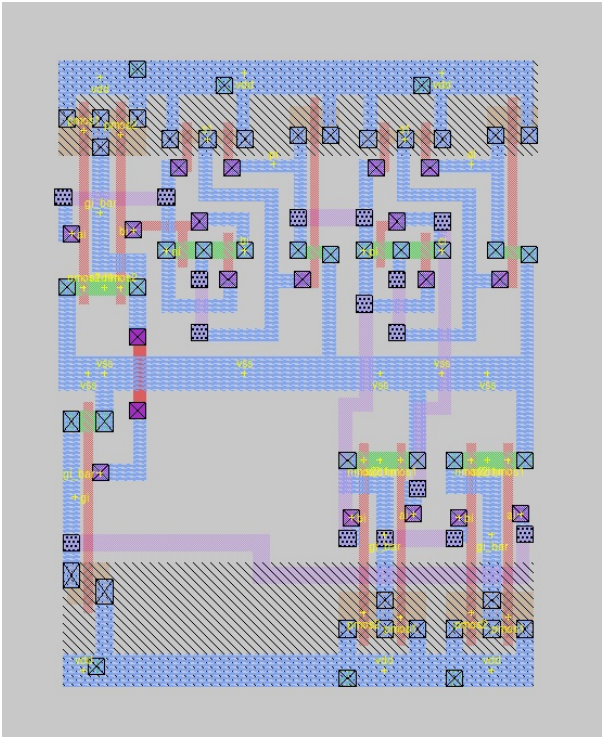

Figure 15: Stage 1

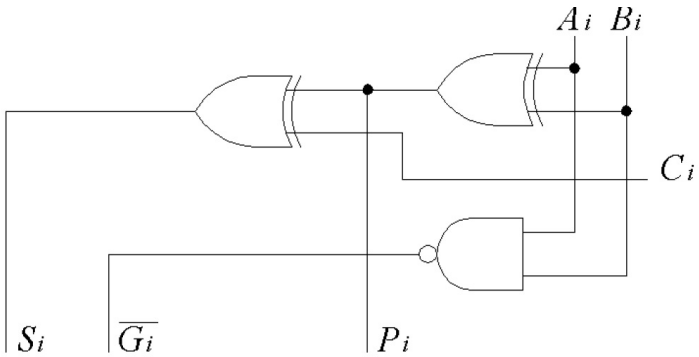

Figure 16: MPFA

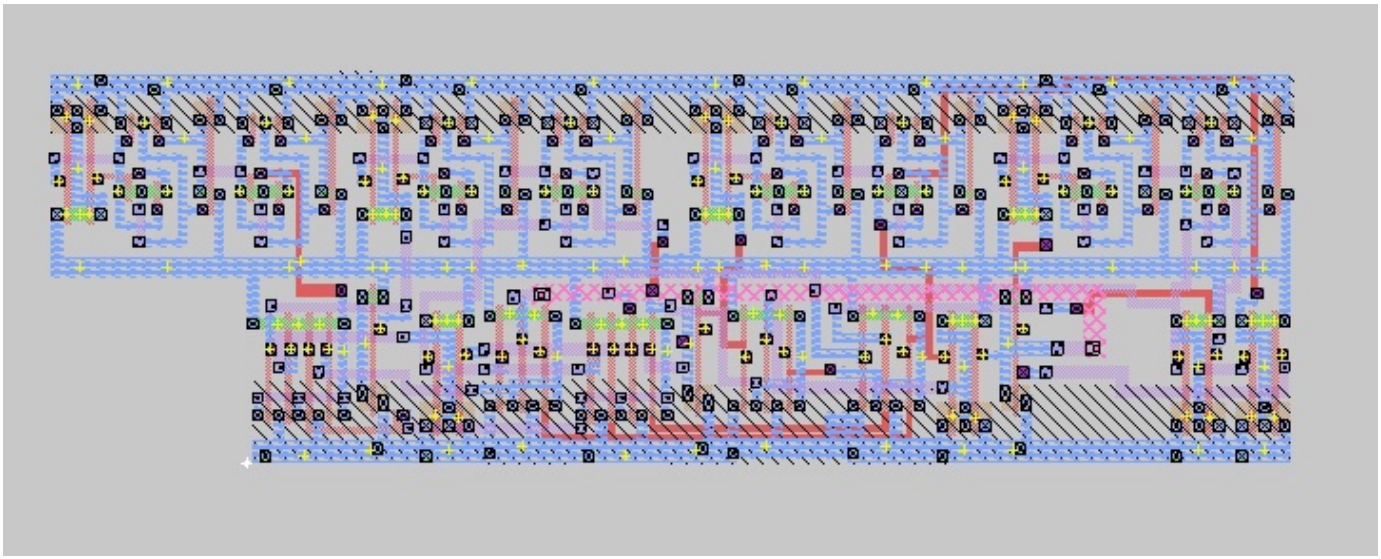

Figure 17: Layout For 4bit Carry Look Ahead Adder

AREA Specifications

AREA Specifications is **489x152 squareunits**

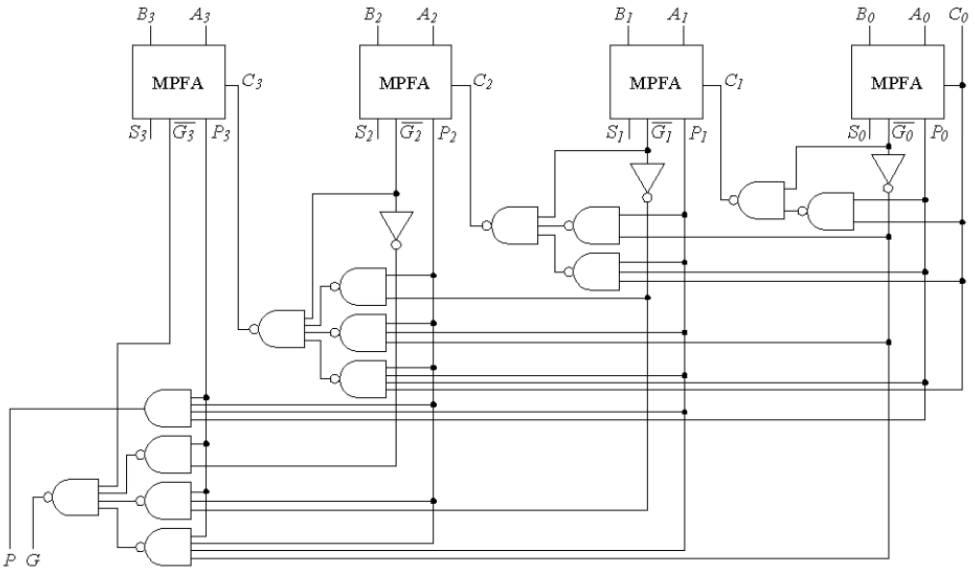

Figure 18: Block Diagram for 4 Bit MCLA

Simulations

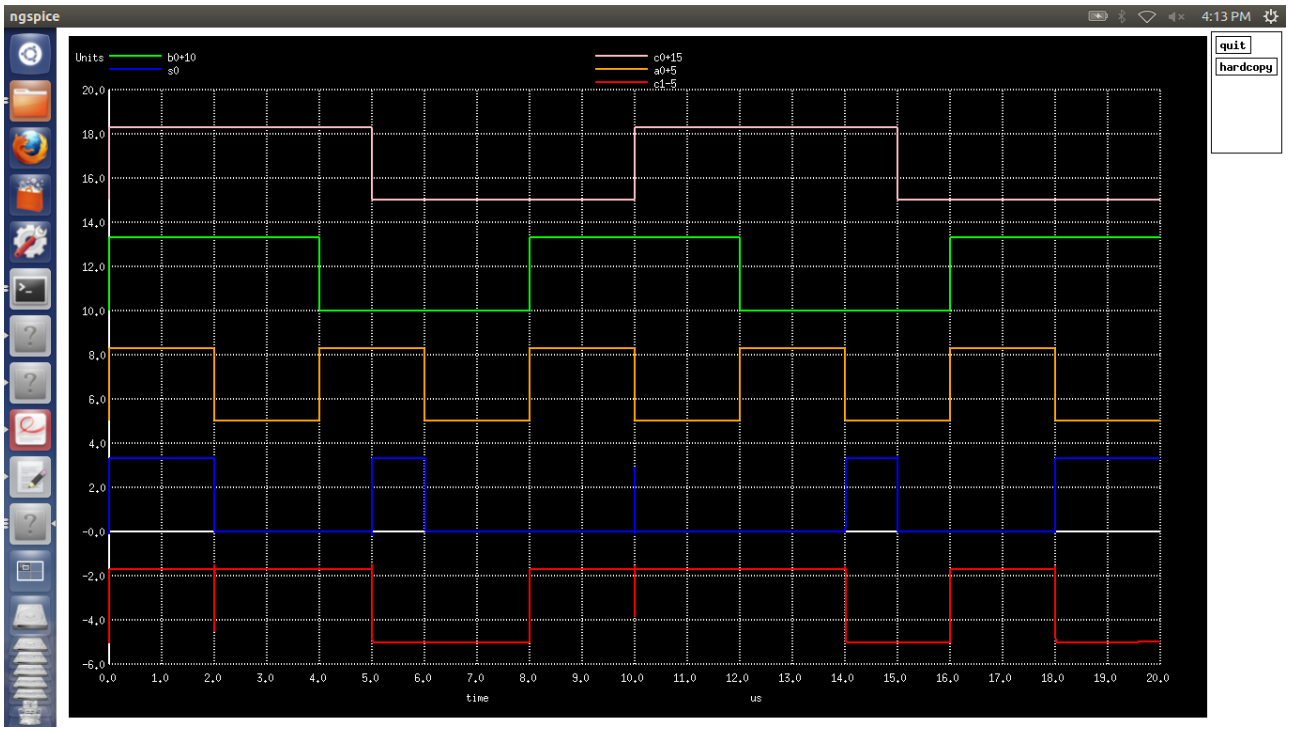

Figure 19: Bit 0 of 4 Bit Addition

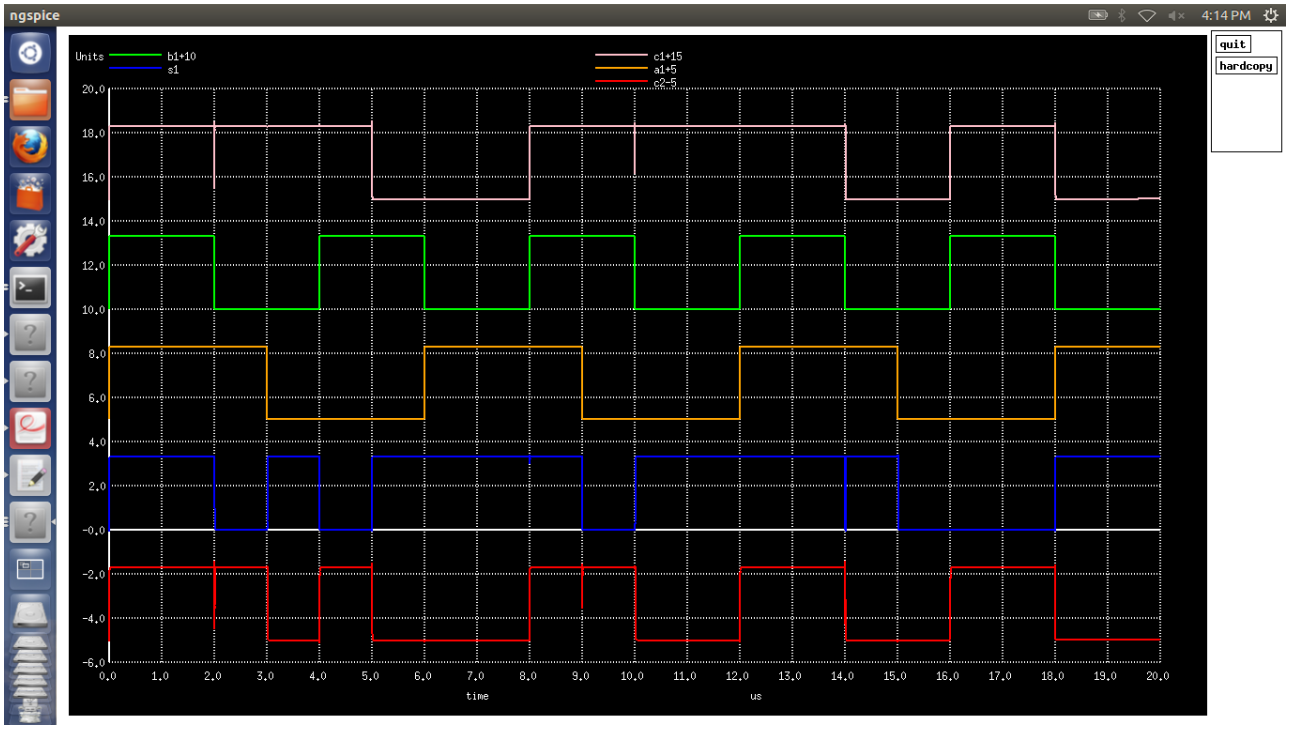

Figure 20: Bit 1 of 4 Bit Addition

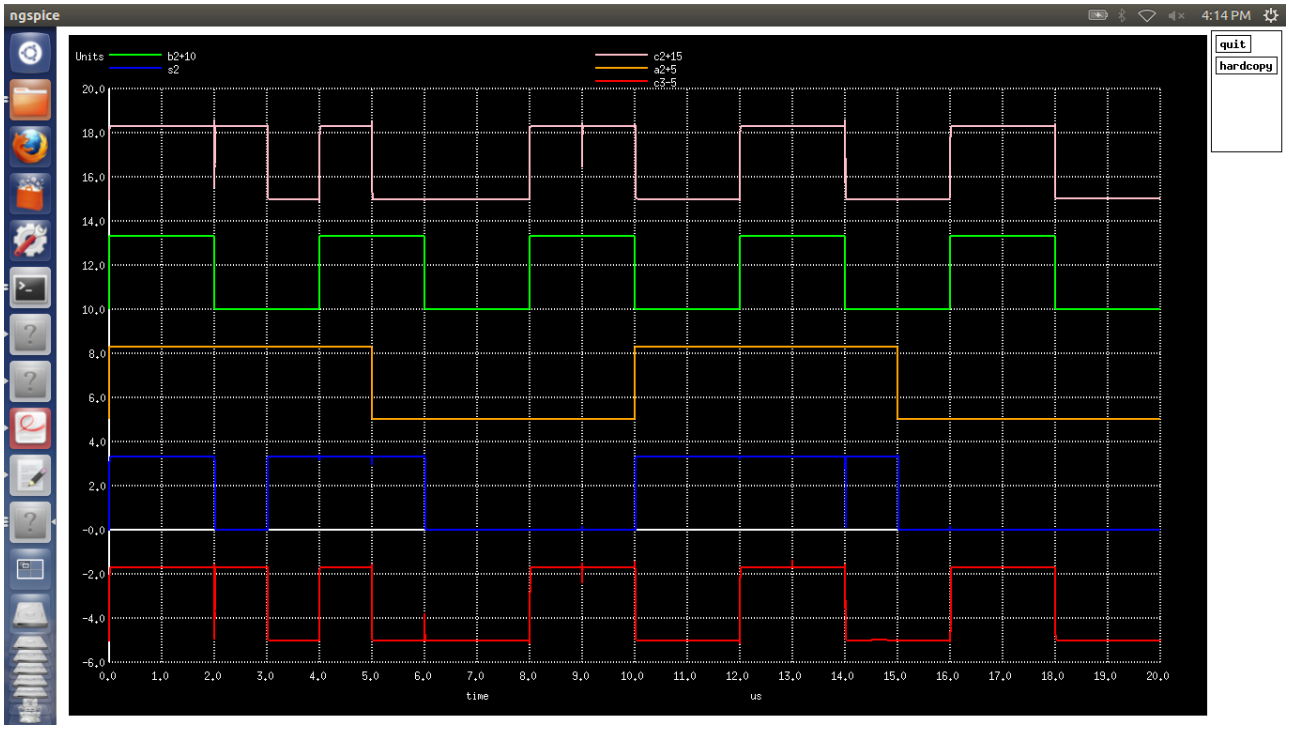

Figure 21: Bit 2 of 4 Bit Addition

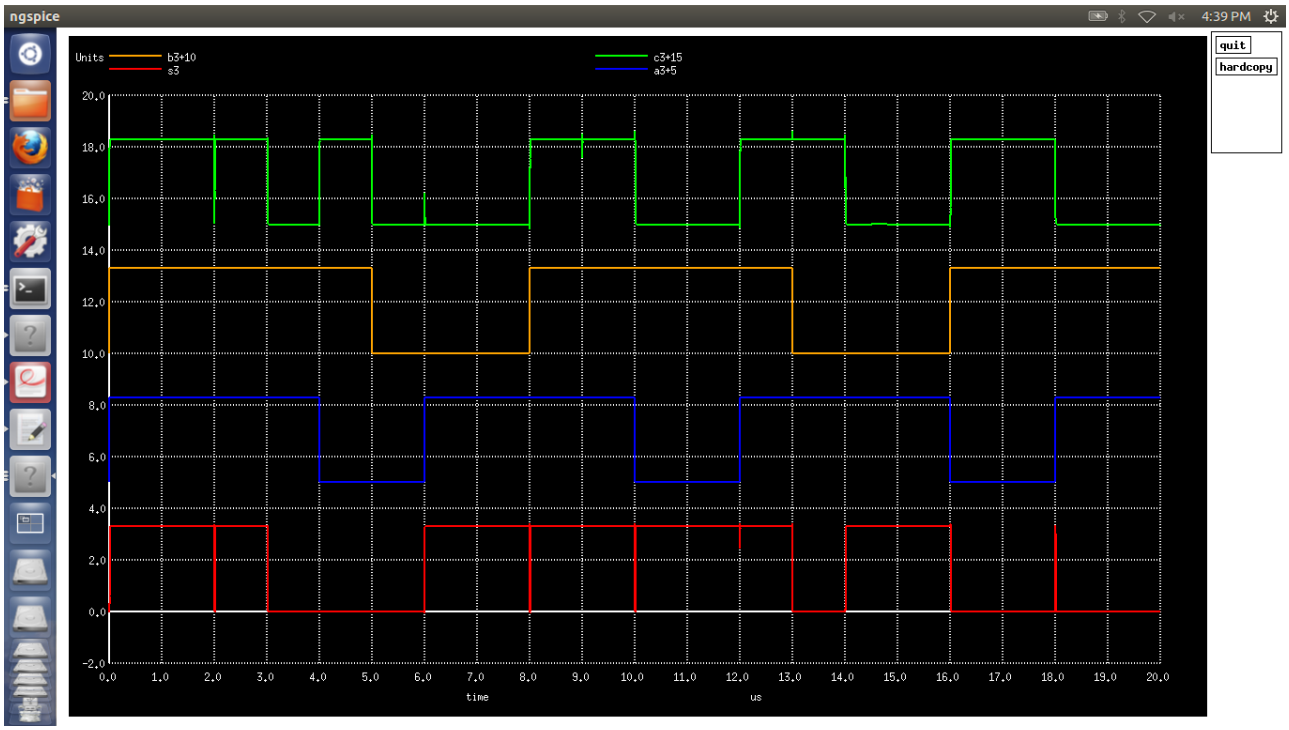

Figure 22: Bit 3 of 4 Bit Addition

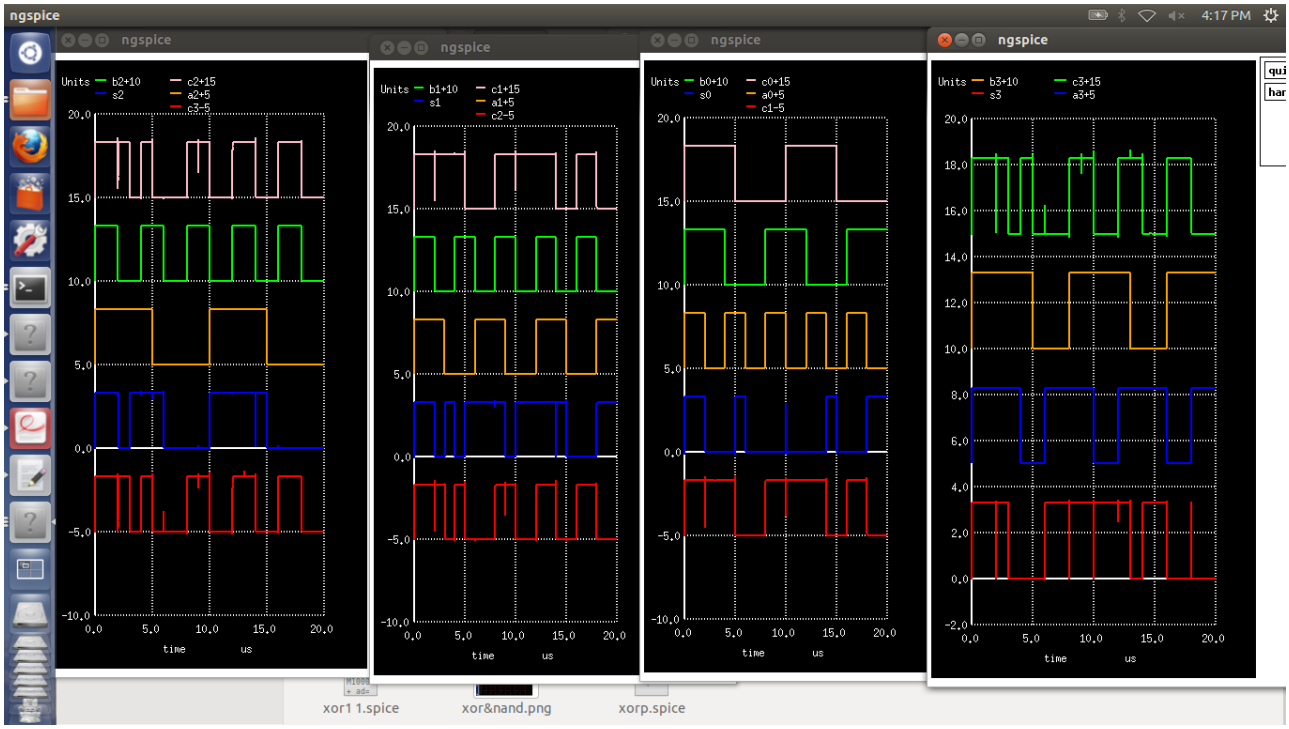

Figure 23: All Graphs shown together for 4 Bit Addition

The Magic circuit is efficient in terms of AREA and Gate Delay as mentioned above and hence can be used for fast and delay insensitive circuits
AREA Specifications of Final Design is **489x152 squareunits**
Delay Specification for 4 bit CLA is **Rise Time Delay 1.89ns and Fall Time Delay is 2.3ns**
Hence we see that from the simulations of the 4 bit CLA output can be easily verified.

Conclusion

The adder is the basic element in CPU. All of the adder-subtractor, and multiplier, etc. are constructed with adders. Therefore, to speed up the adder efficiently is very important to CPU or processor. In order to make it faster, a method to modify these circuits is also proposed. Since the circuit of CLA and MCLA are kinds of recursive circuits, this recursive circuit may be useful for speeding up the other digital logic circuits. For example, the carry lookahead adder-subtractor, can be reformed from MCLA or replaced the full adder in multiplier with MCLA, it will have an excellent improvement in efficiency and cost.
The Magic circuit is efficient in terms of AREA and Gate Delay as mentioned above and hence can be used for fast and delay insensitive circuits
AREA Specifications of Final Design is **489x152 squareunits**
Delay Specification for 4 bit CLA is **Rise Time Delay 1.89ns and Fall Time Delay is 2.3ns**

References

[1] The Fastest Carry Lookahead Adder by Yu-Ting Pai and Yu-Kumg Chen,Department of Electronic Engineering,Huafan University

[2] F. C. Cheng, S. H. Unger, M. Theobald, and W. C. Cho,Delay-Insensitive Carry-lookahead Adders,VLSI Design Proceedings, 1997, pp. 322-328.

[3] C. Nagendra, M.J.Irwin, and R.M.Owens, Area-time-power tradeoffs in parallel adders,IEEE Transactions on Circuits and Systems II,1996,vol.43,pp.689-702.