Quaternary Addition Circuits Based on SUSLOC Voltage-Mode Cells and Modeling with SystemVerilog©

Satyendra R. Datla
Texas Instruments
Dallas, Texas
sdatla@ti.com

Mitchell A. Thornton
Southern Methodist University
Dallas, Texas
mitch@lyle.smu.edu

Luther Hendrix, Dave Henderson
Omnibase Logic
Austin, Texas
Luther.Hendrix@omnibaselogic.com
Dave.Henderson@omnibaselogic.com

Abstract

Multiple Valued Logic (MVL) has been gaining popularity and practical applications. In addition to the standard MVL benefits, quaternary logic offers the benefit of easy interfacing to binary logic due to the fact that the radix $4=2^2$ *allows for simple encoding/decoding circuits. Quaternary cells based on the Supplementary Symmetrical Logic Circuit Structure (SUSLOC)* [2] *are modeled and used for our adder circuit structures. Several different adder configurations are designed and modeled using the basic quaternary gates and are modeled with the SystemVerilog modeling language. Different adder configurations are compared for their size and estimated logic depth for area and performance estimation and compared with their binary counterparts.*

1. Introduction

Increased data density, reduced dynamic power dissipation, and increased computational ability are among some of the key benefits of Multiple Valued Logic (MVL) [4,5]. Several implementation methods have been proposed in the recent past to realize the MVL circuits. They can fundamentally be categorized as: current-mode, voltage-mode and mixed-mode circuits. Even though current-mode circuits have been popular and offer several benefits, the power consumption is high due to their inherent nature of constant current flow during the operation. Alternatively, voltage-mode circuits consume a large majority of power only during the logic level switching. Hence, voltage-mode circuits do offer lesser power consumption which has been the key benefit of traditional CMOS binary logic circuits from the perspective of dynamic switching activity.

Several approaches for quaternary circuit design have been proposed [14,15,16,17,18]. Recently, a self-sustaining and consistent circuit architecture called the Supplementary Symmetrical Logic Circuit (referred as SUSLOC) structure is proposed [2,9,13]. This proposed circuit architecture allows the use of currently available circuit elements to construct logic circuits based on any radix number system. Previous work has utilized SUSLOC technology for ternary systems [8,10,11].

Quaternary (radix-4-valued) is chosen as the base radix for the work reported here. Using a quaternary radix offers all the benefits of MVL such as reduced area due to signal routing reduction along with the important advantage of being able to easily interface with traditional binary logic circuits.

The quaternary full adder circuit is implemented using a subset of the basic SUSLOC gates as described in [2] but extended to base-4. 32-digit quaternary adders are constructed based on a radix-4 full adder circuit cell. Three different types of adders are implemented to compare the area and estimated performance comparison both among themselves and with 64-bit binary versions. These are the carry-ripple, carry-look ahead, and carry-select adders.

The SystemVerilog language offers efficient modeling capabilities to model and simulate large MVL circuits [3]. The quaternary adders described here are simulated by modeling them in SystemVerilog.

This paper is organized as follows. Section 2 provides details about the SUSLOC voltage-mode circuit technology along with basic transistor types and the basic quaternary gates using SUSLOC. Section 3 provides details of various quaternary adder circuits that are designed using the basic SUSLOC gates. Section 4 contains the details on SystemVerilog modeling of the gates and adder functions and Section 5 includes details on comparison results of various different adder architectures. Finally, Section 6 concludes the work and provides the details on future efforts.

2. The SUSLOC Voltage Mode Circuits

The Supplementary Symmetrical Logic Circuit structure (SUSLOC) is the primary motivation for the design of the logic circuits using a radix greater than 2. Voltage-mode circuits will be easily incorporated in modern circuitry; especially since the radix is a power of two. The SUSLOC structure allows the design of such circuits as voltage mode circuits. As elaborated further in [2], three requirements must be met by the circuit structure to design and fabricate MVL circuits:

1) there must be *r* different sources of power available, including ground, with each source of power representing one of *r* different logic levels

2) there must be one controllable path, or branch, from a source of power to an output terminal of the circuit per output logic level and

3) only one controllable path, or branch, conducts from a source of power to an output terminal per input logic level, contiguous group of input logic levels, or a unique combination of input logic levels.

The design rules and definitions of the SUSLOC structure must meet all the above three requirements.

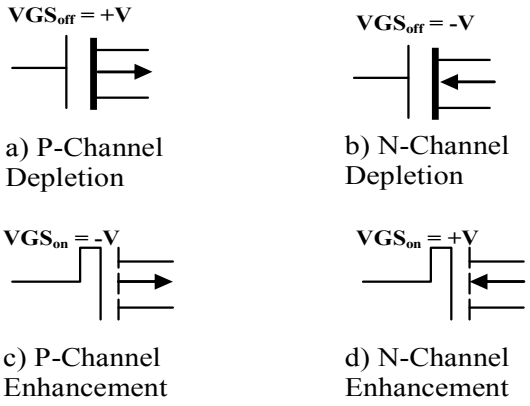

Figure 1. FETs used for SUSLOC structure

Due to their low cost and high reliability, Insulated Gate Field Effect Transistors, (IGFETs, FETs) were chosen in the ternary logic implementation described in [2].

2.1 Transistor types used for Quaternary logic implementation

Figure 1 depicts the four FETs that are used for the quaternary gate implementations used here. These four types of transistors with different threshold voltage selections can help in the aid of designing of circuits based on any radix [2].

2.2 Inverter, MIN, MAX Cells

There are several algebras proposed for expressing and manipulating MVL functions [1]. Functionally complete set of gates that are common to many of these algebras are:

Inversion : Single-place function
MIN and MAX : Two-place function

The actual underlying function requirement heavily depends on the logic of the function to be implemented and many circuits can be optimized by having rich set of single and two-placed functions.

IN	OUT
0	3
1	2
2	1
3	0

a) Inverter

		A			
		0	**1**	**2**	**3**
B	**0**	0	1	2	3
	1	1	1	2	3
	2	2	2	2	3
	3	3	3	3	3

b) MAX

		A			
		0	**1**	**2**	**3**
B	**0**	0	0	0	0
	1	0	1	1	1
	2	0	1	2	2
	3	0	1	2	3

c) MIN

Figure 2. Truth Tables of basic quaternary gates

The basic three functions that are chosen to demonstrate the quaternary implementation of the SUSLOC structure are defined as the Inverter, MAX and MIN. Figure 2 shows the truth tables of these quaternary functions. For the quaternary circuit implementation, the voltage step value chosen is 1.1V. The threshold voltage values for each transistor calculated according to the following formulae obtained from [2]:

$$P-Channel \ : V_{GS(TH)} = V_i - (V_o - (OP \times LSV))$$
$$N-Channel \ : V_{GS(TH)} = V_i - (V_o + (OP \times LSV))$$

Where V_i is input voltage level, V_o is output voltage level, *OP* is overlap percentage and *LSV* is logic step voltage. Table 1 lists the voltage sources used for various logic levels in the construction of the quaternary gates.

Table 1. Quaternary Voltage Sources

Logic level/Supply Voltage	Value (V)
1/V1	1.1
2/V2	2.2
3/V3	3.3

The implementation of the quaternary inverter with example threshold values is shown in the Figure 3.

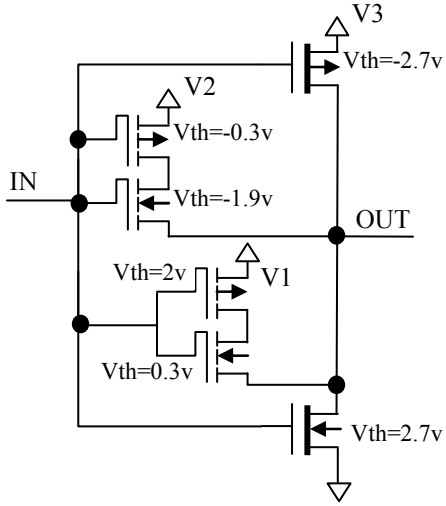

Figure 3. Quaternary Inverter Circuit

One important factor in designing these non-binary circuits is to carefully select the threshold voltage of the transistors used to have the necessary signal-to-noise margin that would guarantee a robust operation. SUSLOC allows a robust operation by forcing the "supplement" structure of *P*-type and *N*-type transistor pair for intermediate logic levels. This can be seen for the intermediate logic level driving logic of V1 and V2.

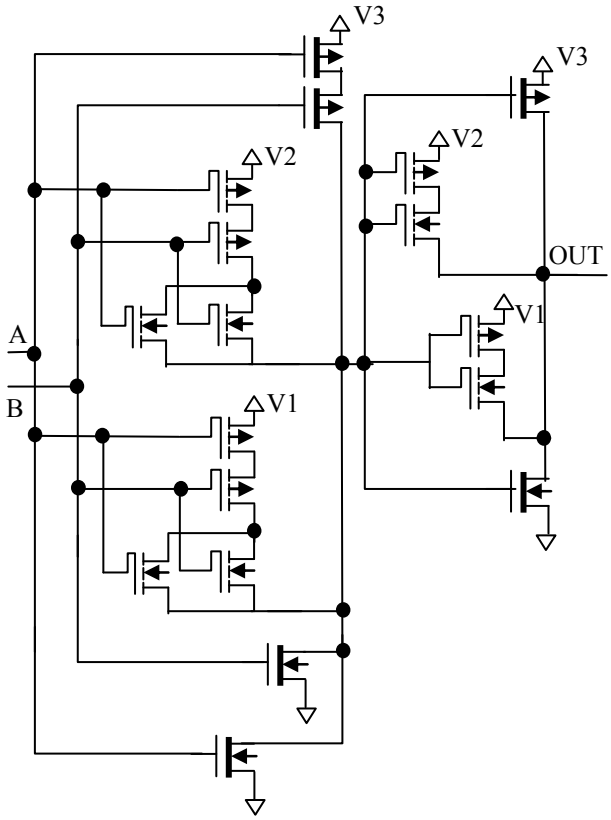

Figure 4. Quaternary MAX circuit

Figures 4 and 5 show the proposed implementation of quaternary MAX and MIN functions. MAX and MIN functions essentially correspond to traditional OR and AND functions respectively.

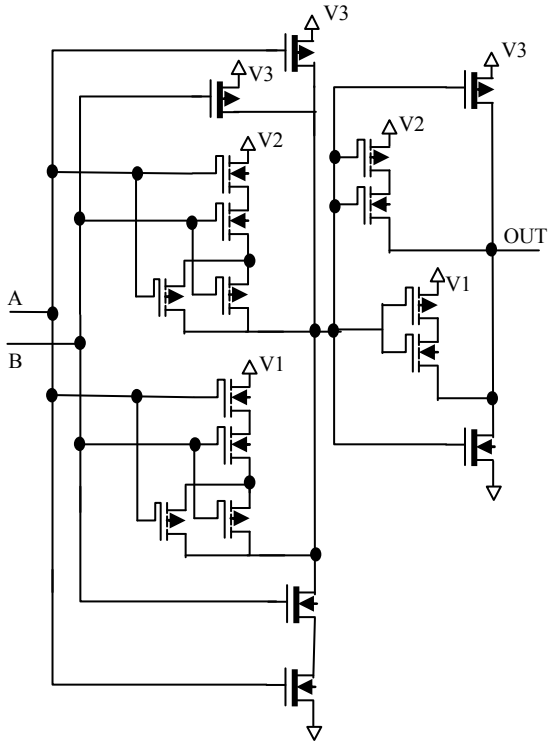

Figure 5. Quaternary MIN circuit

The two input MAX and MIN functions shown here can easily be extended to multiple inputs by adding additional paths from an additional input. This is similar to binary circuits.

3. Adder Circuit Architectures

Using the basic quaternary gates, different adder functions are designed.

3.1 Full Adder

The quaternary full adder is designed using the inverter, min, and max gates defined previously.

Table 2. Quaternary Full adder Truthtable

		Cin = 0		Cin = 1	
A	**B**	**Cout**	**Sout**	**Cout**	**Sout**
0	0	0	0	0	1
0	1	0	1	0	2
0	2	0	2	0	3
0	3	0	3	1	0
1	0	0	1	0	2
1	1	0	2	0	3
1	2	0	3	1	0
1	3	1	0	1	1
2	0	0	2	0	3
2	1	0	3	1	0
2	2	1	0	1	1
2	3	1	1	1	2
3	0	0	3	1	0
3	1	1	0	1	1
3	2	1	1	1	2
3	3	1	2	1	3

Table 2 contains the truth table of the quaternary full adder. Note that only carry-in digits {0,1} are possible for a two-operand adder, thus the case for carry-in={2,3} is not needed. Figure 6 below shows the equations for the Sum and Carry outputs of the quaternary full adder circuit according to the algebraic notation defined in [1].

$$
\begin{aligned}
SOUT \;=\; & 1 \bullet [A^{\{0\}} \bullet (B^{\{1\}}C^{\{0\}} + B^{\{0\}}C^{\{1\}}) + \\
& A^{\{1\}} \bullet (B^{\{0\}}C^{\{0\}} + B^{\{3\}}C^{\{1\}}) + \\
& A^{\{2\}} \bullet (B^{\{3\}}C^{\{0\}} + B^{\{2\}}C^{\{1\}}) + \\
& A^{\{3\}} \bullet (B^{\{2\}}C^{\{0\}} + B^{\{1\}}C^{\{1\}})] \\
& + 2 \bullet [A^{\{0\}} \bullet (B^{\{2\}}C^{\{0\}} + B^{\{1\}}C^{\{1\}}) + \\
& A^{\{1\}} \bullet (B^{\{1\}}C^{\{0\}} + B^{\{0\}}C^{\{1\}}) + \\
& A^{\{2\}} \bullet (B^{\{0\}}C^{\{0\}} + B^{\{3\}}C^{\{1\}}) + \\
& A^{\{3\}} \bullet (B^{\{3\}}C^{\{0\}} + B^{\{2\}}C^{\{1\}})] \\
& + 3 \bullet [A^{\{0\}} \bullet (B^{\{3\}}C^{\{0\}} + B^{\{2\}}C^{\{1\}}) + \\
& A^{\{1\}} \bullet (B^{\{2\}}C^{\{0\}} + B^{\{1\}}C^{\{1\}}) + \\
& A^{\{2\}} \bullet (B^{\{1\}}C^{\{0\}} + B^{\{0\}}C^{\{2\}}) + \\
& A^{\{3\}} \bullet (B^{\{0\}}C^{\{0\}} + B^{\{3\}}C^{\{1\}})]
\end{aligned}
$$

$$COUT = 1 \bullet [A^{\{0\}} B^{\{3\}} C^{\{1\}} +$$
$$A^{\{1\}} \bullet (B^{\{3\}} C^{\{0\}} + B^{\{2,3\}} C^{\{1\}}) +$$
$$A^{\{2\}} \bullet (B^{\{2,3\}} C^{\{0\}} + B^{\{1,2,3\}} C^{\{1\}}) +$$
$$A^{\{3\}} \bullet (B^{\{1,2,3\}} C^{\{0\}} + C^{\{1\}})]$$

Figure 6. Quaternary Full Adder equations

Different operators used in the equations of Figure 6 are defined in the below table 3. The truth tables of the basic quaternary functions used are shown before in Figure 2.

Table 3. Quaternary operator definitions

Operator	Definition
•	Quaternary MIN
+	Quaternary MAX
{ }	Values mentioned are logic states of the corresponding quaternary variable

Additional simplification in the resultant circuits would likely occur if additional single- and multiple-place function gates were formulated and used.

3.2 Ripple-Carry Adder

A ripple-carry adder is a chain of individual full adder circuits. A 16-digit Quaternary ripple carry adder is designed by chaining the 1-digit quaternary full adder cell. The 4-digit example is shown in Figure 7.

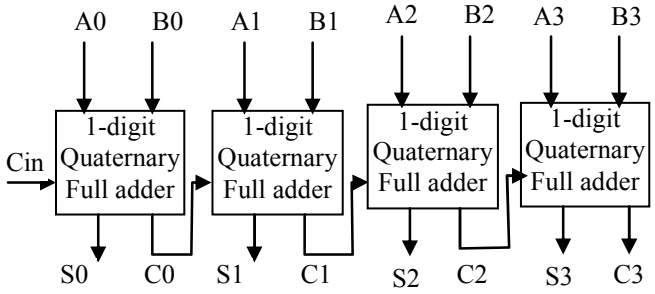

Figure 7: 4-digit Quaternary Ripple Carry Adder

The ripple-carry adder is simple and easy to design and is area efficient. The well known disadvantage with the ripple carry adder is the limited performance due to the need for carry propagation. This can be seen from the daisy chaining of the single digit carry-out values for each full adder in Figure 7. The most significant digit (MSd) result has dependency on the carry-out digit from the previous stage. Essentially, summation output for each digit is delayed until the carry out from previous digit is available.

3.3 Carry Look-Ahead (CLA) Adder

A carry look-ahead adder is a widely used alternative adder architecture that achieves higher performance by utilizing the so-called carry generate and propagate logic functions. The quaternary carry look-ahead adder is proposed in [6]. A 16-digit carry look-ahead adder is designed using the SUSLOC gates and the overall structure of the addition circuit is shown in Figure 8.

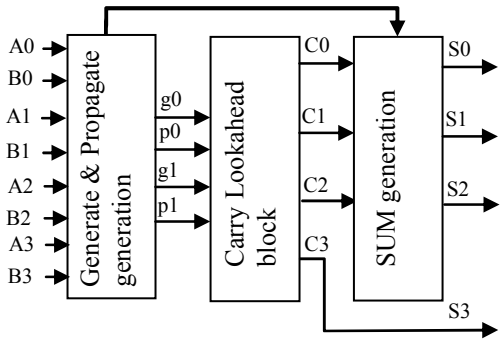

Figure 8. 4-digit Quaternary CLA Adder

The generate and propagate circuit blocks produce the required $g0$, $g1$, $p0$ and $p1$ signals. These functions for the quaternary case are shown in Figure 9. Note that a quaternary carry-out is either a 0 or a 1; for example, 3+3=12. The full adder was designed with this consideration which helps to conserve area. However, this also means that, unlike the binary case, the quaternary full adder cannot be used as a 3:2 compressor as commonly used in binary multiplier circuit partial product accumulation trees. The following equations contain the generate and propagate functions for the radix-4 full adder.

$$g0 = A^{\{0\}} + B^{\{0\}} + A^{\{1\}} B^{\{1,2\}} + A^{\{2\}} B^{\{1\}}$$
$$g1 = A^{\{1\}} B^{\{3\}} + A^{\{2\}} (B^{\{2\}} + B^{\{3\}}) + A^{\{3\}} B^{\{1\}}$$
$$p0 = g0$$
$$p1 = A^{\{0\}} B^{\{3\}} + A^{\{1\}} B^{\{2\}} + A^{\{2\}} B^{\{1\}} + A^{\{3\}} B^{\{0\}}$$

Figure 9. Equations for Generate and Propagate Signals of Radix-4 Addition Digits

The carry look-ahead adder elaborated above is expanded to 4-digits using the 1-digit quaternary adder and carry look-ahead logic. The increase in area is due to the carry generation and propagation and is compensated by the additional performance gain in CLA adders as is the usual case.

3.4 Carry Select Adder

The carry select adder is another well known adder circuit architecture that addresses the carry propagate delay challenge in a different way. The addition operation is duplicated with the assumption of both input carry being equal to 0 and equal to 1 in the case of binary addition. Due to arguments raised previously, this is also the case for radix-4 addition as well. The 4-digit example of the carry select adder is shown in Figure 10. As is seen in the example, the redundant adder would result in an area increase; however, the performance of the adder is increased because the result with all possible carry values are available and can be readily selected when the carry value is available reducing the propagation time. We note that the use of the radix-4 system does not affect the rules

for carry-select adders in addition to providing easy interfacing as compared to the binary case.

The area could be optimized further by replacing the full adders with equivalent half-adders because of the constant carry input.

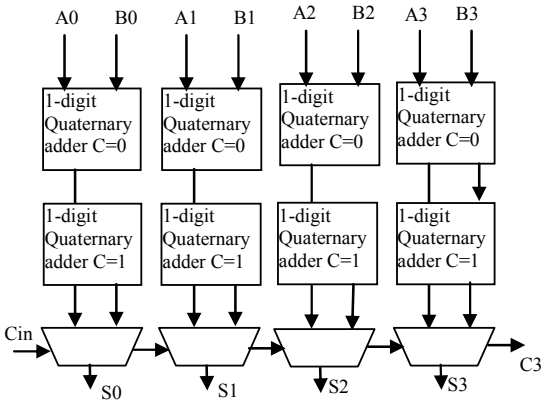

Figure 10. 4-digit Quaternary Carry Select Adder

4. SystemVerilog Modeling

Traditional modeling languages such as Verilog and VHDL offer very good modeling capabilities for binary circuits. On the other hand, MVL circuit implementations require extended data types and handling of values more than two. SystemVerilog offers a viable solution to the modeling of MVL logic with its rich set of extended features and data types. A similar approach was noted in [12] using SystemC. The basic quaternary gates inverter, MAX and MIN (referred as MAXQ and MINQ for quaternary logic) gates are modeled in SystemVerilog as shown in Figure 11.

```
function int INVQ(int in);
int out;
out = ((4-1)-in);
return out;
endfunction

function int MINQ(int a, int b);
int out;
out = ((a<b) ? a : b);
return out;
endfunction

function int MAXQ(int a, int b);
int out;
out = ((a>b) ? a : b);
return out;
endfunction
```

Figure 11. System Verilog models for quaternary gates

All the three different adder circuits mentioned in the previous section are modeled using the SystemVerilog language and verified in the Synopsys© VCS© environment. SystemVerilog was also very useful for development of the testbench modules used to validate the functionality of the addition circuits.

5. Circuit Modeling Results

Previously ternary logic circuits have been built, tested, and incorporated into a simple demo system using SUSLOC in [11]. The newly designed quaternary logic gates and the adder functions are both modeled and verified using the System Verilog language. A 32-digit version of the quaternary adder circuits is chosen so that comparisons to 64-bit circuits can be made. However, we note that the multitude of synthesis optimization techniques used for the binary cases had no equivalences in the radix-4 case and we are using a basic universal gate set with manual algebraic optimizations. . The comparison charts in Figures 12 and 13 summarize the observations of various adder circuits designed. Even though the binary circuits take advantage of the last 30 years of logic synthesis optimization results, the radix-4 circuits with virtually no optimizations exhibit superior properties.

In terms of area, we counted the number of transistors required for each of the quaternary and binary equivalent circuits. We realize that the transistors used in the quaternary cells will likely actually be larger than the binary FETs, however we did not include the reduction in conductor path routing since the quaternary versions of the circuits will require fewer data signals since two bits of information are carried per conductor. With the trends in modern device scaling and routing as of this writing, the area reduction due to halving the signal conductor traces is very significant but cannot be measured until a design is actually physically implemented on a die.

In terms of performance, we simply found the longest gate path from input to output. Again this is only an estimate of performance, but we currently do not have a quaternary standard cell library and hence could not analyze the cells for more accurate timing results.

As can be seen in the results shown in Figures 12 and 13, the area and estimated logic depth for 32-digit quaternary addition circuits are compared to their equivalent 64-bit binary counterparts for each configuration. The quaternary adder circuits do have larger transistor counts when compared to their binary counter parts in all the cases. As noted before, only transistor count is compared here and inclusion of wire area could bring quaternary circuits closer. The two additional power supplies required for implementing quaternary circuits does consume additional routing resources. But, their impact is estimated to be well within the 1-2% impact on routing congestion.

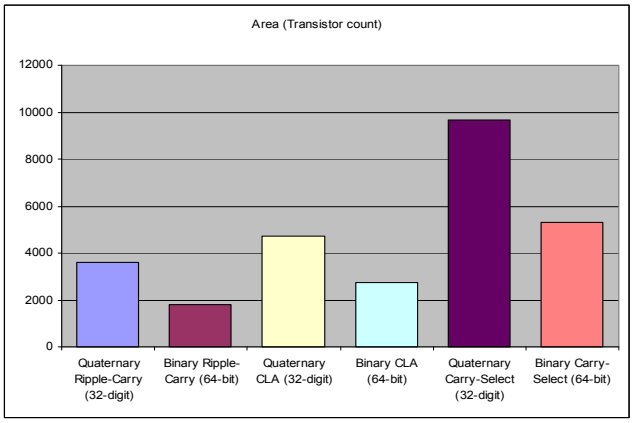

Figure 12. Area comparison

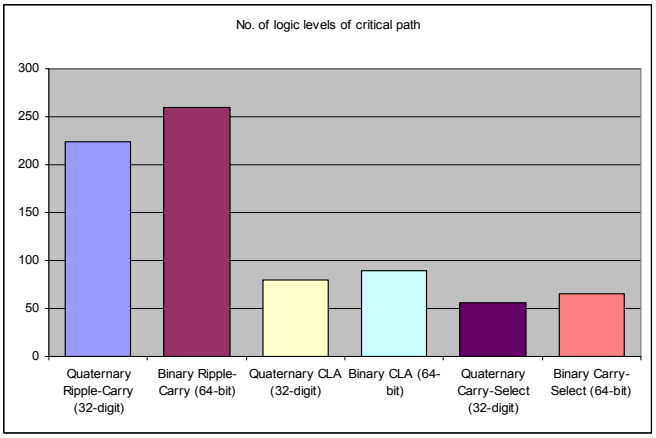

Figure 13. Logic depth comparison

On the other hand, in all cases, the quaternary adder circuits do have better timing performance than the binary equivalent circuits as measured in the depth (no. of transistors) of the critical path as shown in Figure 13.

6. Conclusions and Future Directions

This work illustrates how quaternary gates can be implemented using the SUSLOC circuit structure. These gates were then used to build three different addition circuit architectures: carry-ripple, carry-lookahead, and carry-select. These three quaternary addition circuits are then modeled using the SystemVerilog language. The three different adder architectures are designed and compared for area and estimated performance with their binary circuit counterparts.

In the future we plan to design the standard cell library of quaternary logic cells based on the SUSLOC structure. Also, we will develop and implement necessary synthesis algorithms to directly map the quaternary gates from the Register Transfer Level (RTL) coding. This will provide the platform to implement larger commonly used functions in quaternary logic. Designing a quaternary Arithmetic Logic Unit (ALU) based on SUSLOC structure to demonstrate these capabilities is the ultimate goal.

Acknowledgement

The team greatly acknowledges the use of tools donated by the Synopsys® Corporation to support this research at Southern Methodist University, especially the use of SystemVerilog under the VCS environment.

References

[1] D.M. Miller and M.A. Thornton, **Mutiple-Valued Logic: Concepts and Representations**. *Morgan & Claypool Publishers*, San Rafael, CA, ISBN 10-1598291904, 2008.

[2] U.S.Patent, *6,133,754,* Edgar Danny Olson, inventor, Multiple-Valued Logic Circuit Architecture; Supplementary Symmetrical Logic Circuit Structure (SUS-LOC).

[3] M. Amoui, D. Groβe, M.A. Thornton, and R. Drechsler, Evaluation of Toggle Coverage for MVL Circuits Specified in the System Verilog HDL, *Proc. of IEEE Int. Symp. On Multiple-Valued Logic*, 2007, Session 8B, Paper 2.

[4] S. Hurst, Multiple-valued logic –its status and its future. *IEEE trans. On Computers*. C-33(12), 1984, pp.1160-1179.

[5] M. Kameyama, Toward the Age of Beyond-Binary Electronics and Systems. *Proc. of IEEE Int. Symp. On Multiple-Valued Logic*, 1990, pp.162-166.

[6] I.M. Thoidis, D. Soudris, J.M. Fernandez, and A. Thanailakis, The circuit design of Multiple-Valued Logic Voltage-Mode Adders. *Proc. of IEEE International Symposium on Circuits and Systems*, 2001, pp.162-165.

[7] G.R. Cunha, B.H. Ivanov, and C. Luigi, A low power high performance CMOS voltage-mode quaternary full adder. *IFIP International conference on Very Large Scale Integration,* 2006, pp. 187-191.

[8] E.D. Olson, K.W. Current, Hardware Implementation of "Supplementary Symmetrical Logic Circuit Structure" Concepts**,** *Proc. of IEEE Int. Symp. On Multiple-Valued Logic*, 2000.

[9] E.D. Olson, "Supplementary Symmetrical Logic Circuit Structure", *International Symposium on Multiple Valued Logic*, pp. 42-47, 1999.

[10] M. Aline, T. Saidi, E. Kinvi-Boh, O. Sentieys, E.D. Olson, Design and Characterization of a Low Power Ternary DSP, *Proc. of International Signal Processing Conference*, 2003.

[11] E.D. Olson and K.W. Current, Demonstration of "Supplementary Symmetrical Logic Circuit Structure" Concepts Using a MOS Test Chip, *Journal of Multiple-Valued Logic*, vol. 7, 2001, pp. 1-23.

[12] D. Große, G. Fey, and R. Drechsler. Modeling Multi-Valued Circuits in SystemC, *International Symposium on Multiple Valued Logic*, pp. 281–286, 2003.

[13] E. Kinvi-Boh, M. Aline, O. Sentieys, and E.D. Olson, MVL Circuit Design and Characterization at the Transistor Level Using SUS-LOC, *International Symposium on Multiple Valued Logic*, 2003.

[14] F. Wakui and M. Tanaka, Comparison of Binary Full Adder and Quaternary Signed-Digit Full Adder using High-Speed ECL, *International Symposium on Multiple Valued Logic*, 1989, pp. 346-355.

[15] M.K. Habib and A.K. Cherri, Parallel Quaternary Signed-Digit Arithmetic Operations: Addition, Subtraction, Multiplication, and Division, *Optics and Laser Technology*, vol. 30, 1998, pp. 515-525.

[16] H. Shirahama and T. Hanyu, Design of High-Performance Quaternary Adders Based on Output-Generator Sharing, *International Symposium on Multiple Valued Logic,* 2008.

[17] R.G. Cunha, H. Boudinov, and L. Carro, A Novel Voltage-Mode CMOS Quaternary Logic Design, *IEEE Trans. On Electronic Devices*, vol. 53, no. 6, 2006, pp. 1480-1483.

[18] Y. Yasuda, Y. Tokuda, S. Zhaima, K. Pak, T. Nakamura, abd A. Yoshida, Realization of quaternary logic circuits by N-Channel MOS Devices, *IEEE Journal of Solid State Circuits*, vol. SC-21, no. 1, 1986, pp. 162-168.