Quaternary Voltage-Mode Logic Cells and Fixed-Point Multiplication Circuits*

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

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

***Abstract*—Fixed-point multiplication architectures are designed and evaluated using a set of logic cells based on a radix-4, quaternary number system. The library of logic circuits is based on Field Effect Transistors (FETs) that have different voltage threshold levels. The resulting logic cell library is sufficient to implement all possible quaternary switching functions. The logic circuits operate in voltage mode where different ranges of voltages encode the logic levels. Voltage mode circuitry is used to minimize overall power dissipation characteristics. Analysis of the resulting multiplication circuits indicates that power dissipation characteristics are advantageous when compared to equivalent word-sized binary voltage mode configurations with no decrease in performance.**

Keywords-quaternary logic, arithmetic logic circuits

I. INTRODUCTION

Arithmetic circuits play a central role in both general-purpose and application specific computational circuits. Multiple-Valued Logic (MVL) circuits that use more than two discrete voltages to encode logic levels can be advantageous in that higher density per integrated circuit area can be achieved as compared to binary logic implementations. A quaternary (radix-4) logic system allows for the use of relatively simple encoding/decoding circuits to be employed for interfacing to binary logic since radix $4=2^2$.

Fixed-point multiplication circuits are fundamental building blocks for many computational algorithms ranging from simple arithmetic to graphic and signal processing applications. The increased data density due to the higher informational content per conductor is one of the main benefits of MVL circuit implementations [1-3]. Several transistor-based implementation technologies have been proposed to realize MVL circuits [4-18]. Conventional electronic MVL circuits can be categorized as current-mode, voltage-mode, or mixed-mode circuits. Current-mode circuits [4,19,20] have been popular and offer several benefits with respect to ease of implementation in arithmetic circuits. As an example, an addition operation can be implemented as a circuit node that relies on Kirchoff's additive current law. However, current mode power consumption can be high as compared to voltage mode circuits due to constant current flow during functional operation. Alternatively, voltage-mode circuits [1-3,21] consume dynamic power only during logic level switching. Static power consumption also occurs through any additional leakage currents that may be present, however, this power consumption mode is generally much less than that due to steady-state current flow in current mode implementations. Hence, voltage-mode circuits offer reduced power consumption from the perspective of dynamic switching activity. Lower power consumption is the key benefit of using traditional CMOS binary logic circuits for several technology generations. Because of the increased proliferation of portable battery-powered personal computational devices, reduced power dissipation is an important design constraint and motivates the investigation of the voltage-mode MVL multiplication circuits described here.

In order to design the multiplier architectures, a library of subcircuits is composed that is sufficient to realize any arbitrary combinational function. Additionally, an edge-triggered storage cell is included allowing for synchronous circuits to be realized including basic structures such as a quaternary register. The storage cells utilize a binary valued clock signal but can store any of the four different discrete voltage levels. Functional completeness is assured by showing that any possible 4-valued switching function may be decomposed into a set of logic operations that correspond to one of the library cells assuming that logic constants are available.

The paper is organized by first discussing relevant background information regarding the electronic characteristics of the voltage mode circuitry. Next, we demonstrate the algebraic functional completeness of a set of operators corresponding to the logic cell library. An overview of the logic cells is included with a discussion of the mode of operation of the cells. Finally, we present the architecture of two different quaternary multiplication circuits followed by simulation and analysis results that allow for performance evaluation. The paper concludes with a discussion of the results of the project.

II. BACKGROUND

A. SUSLOC Circuit Structure

Several approaches to MVL circuit design using traditional transistors in both current- and voltage-modes have been proposed in the past. Recently, an approach structurally similar to that of a static-CMOS binary circuit, called the Supplementary Symmetrical Logic Circuit structure (SUSLOC), was proposed and patented [22,23]. SUSLOC MVL circuits theoretically offer a stable circuit implementation structure for any integer radix, and also allow the use of readily available circuit elements to construct logic circuits. Previous work has utilized SUSLOC technology for ternary systems [23-27]. Insulated gate Field Effect Transistors (FET) are used as the basic building blocks for SUSLOC circuits. As described in [22], three requirements must be met by the circuit structure to design and fabricate quaternary MVL circuits using SUSLOC:

1) Three different voltage sources are required, with each voltage source representing one of three different logic levels and the ground plane represents the fourth level.
2) One controllable path, or branch, from a voltage source to an output terminal of the circuit per output logic level must be present.
3) Only one controllable conductor path, or branch, is allowed from a voltage source to an output terminal per input logic level, contiguous group of input logic levels, or unique combination of input logic levels.

Due to their low cost and high reliability, Insulated Gate Field Effect Transistors (FET) are used in the implementation of the ternary logic circuits described in [23-25]. For similar reasons, we also use the same types of transistors for the work described here. Figure 1 depicts the symbols used to represent the four FETs used for the quaternary gate implementations. It is necessary to fabricate each

of these four transistor types with different threshold voltages, V_T. Because of this requirement, the ability to vary the doping levels to obtain different V_T values is essential. In theory, these four types of transistors, with different threshold voltages, can be used in the design of SUSLOC circuits based on any radix [23]. From a practical point of view, as the underlying radix value increases, the number of different V_T must increase imposing pragmatic limitations on the size of the set of useful logic values.

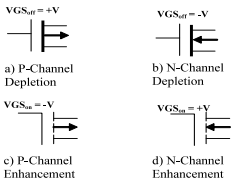

Figure 1. FET Symbols used for SUSLOC Circuits

B. Quaternary Logic System and Notation

The quaternary logic circuits used in this work are modeled by a four-valued switching algebra over variables that may take on the values {0, 1, 2, 3}. The decisive literal of a quaternary variable, x, is denoted by $x^{\{0\}}$, $x^{\{1\}}$, $x^{\{2\}}$, or $x^{\{3\}}$ [28] where $x^{\{a\}}=3$ when $x=a$, otherwise $x^{\{a\}}=0$.

There are many possible algebraic operators for this logic system; 256 one-place operators and in excess of four billion two-place operators. The small subset of operators used in this work corresponds to relatively easily implemented SUSLOC circuits. We use five different one-place operators; the quaternary inverter denoted as $\overline{x}$ where $\overline{x}=3-x$, and the four decisive literal functions, $x^{\{a\}}$, described previously. In terms of two-place operators, the *MIN* and *MAX* functions are utilized as defined in [13]. The *MIN* function is denoted as $x \cdot y$ where $x \cdot y=x$ if $x<y$, otherwise $x \cdot y=y$. When two literals or terms appear next to one another, the *MIN* operator, ·, is implied to be present. The *MAX* function is denoted by $x+y$ where $x+y=x$ if $x>y$ else $x+y=y$.

III. Functional Completeness of the Algebra

To show that the collection of operators over the quaternary-valued logic defined in Section II is sufficient to specify all possible switching functions, some definitions, lemmas, and a theorem are provided.

Definition 1: A specific *product selection term* is denoted by J_{Pi} where the integer index P_i denotes the particular product selection term for which J_{Pi} is non-zero. P_i is the numeric value (in the decimal system) given by the digit string $k_1k_2\ldots k_n$ where $k_i \in \{0, 1, 2, 3\}$ and is in the range $[0,4^n]$. □

As an example, consider the case for $n=4$, then $J_{35}=x_1^{\{0\}}x_2^{\{2\}}x_3^{\{0\}}x_4^{\{3\}}$ since $0\times4^3+2\times4^2+0\times4^1+3\times4^0=35$.

$$J_{35}=\begin{cases}3, & x_1=0, x_2=2, x_3=0, x_4=3\\ 0, & \text{otherwise}\end{cases} \quad (1)$$

Definition 2: A *minterm* of a quaternary switching function is formed by combining the P_i^{th} logic value in the truth table with the J_{Pi} product selection term using the *MIN* operation. □

Definition 3: A *sum-of-minterms* (SOM) form of a quaternary function is a form consisting of all minterms combined with a *MAX* operator. □

As an example, consider a quaternary switching function of n variables, f, where each function range value corresponding to the P_i^{th} product term in the truth table is denoted as f_{Pi}. The function f can be expressed in SOM form, f_{SOM}, as shown in Equation (2).

$$f_{SOM}=f_{P0}\bullet J_{P0}+f_{P1}\bullet J_{P1}+\cdots+f_{P4^n}\bullet J_{P4^n} \quad (2)$$

Lemma 1: The SOM form of a quaternary function is a canonical representation.

Proof: The truth table for a switching function consists of rows with all possible variable assignments and their corresponding function range values. For a completely specified quaternary switching function of n variables, the corresponding truth table representation contains 4^n rows. The truth table representation can be expressed as a column vector of function range values with each vector component in a well-defined order corresponding to the particular assignment of domain variables, the P_i^{th} term, denoted as **F**. Any permutation of the vector component values or any change in one or more of the vector component values yields a vector (or corresponding truth table) that represents a different switching function. Thus, every switching function has a unique vector representation. Because each product selection term is non-zero for one and only one product assignment, Equation (2) can be formed as the inner product of a row vector whose components are all unique selection product terms (denoted as $\mathbf{J_P}$) and **F** as $f_{\text{SOM}}=\mathbf{J_P}\cdot\mathbf{F}$. Since each **F** vector is unique for a given switching function and since Equation (2) can be formed using **F**, the proof is complete. □

Theorem 1: The collection of quaternary logic operators consisting of the two-place *MIN* and *MAX* functions, the four decisive literal functions $\{x^{\{0\}}, x^{\{1\}}, x^{\{2\}}, x^{\{3\}}\}$, and the constants {0, 1, 2, 3} form a functionally complete set over all possible quaternary-valued switching functions.

Proof: Consider an arbitrary quaternary switching function of n variables of the form $f:\{0, 1, 2, 3\}^n \rightarrow \{0, 1, 2, 3\}$. For a completely specified function, f, the range values associated with the P_i^{th} minterms are constants from the set {0,1,2,3} denoted by f_{Pi}. Clearly, Equation (2) is a general representation of any arbitrary quaternary switching function since the constants denoted by f_{Pi} can represent any assignment from {0, 1, 2, 3}. From Lemma 1, Equation (2) is also proven to be canonic. Therefore, any completely specified function could be represented in the form of Equation (2), which is based only on the quaternary *MIN*, *MAX*, decisive literal functions, and the set of constants {0, 1, 2, 3}. □

IV. Quaternary Logic Circuit Library

A collection of quaternary logic subcircuits is designed based on the SUSLOC voltage-mode technology. This library of cells includes five unary circuits, the *MIN* and *MAX* operations, a quaternary D-flip-flop, and a quaternary multiplexer. The five unary circuits correspond to the four decisive literal functions and the inverter as defined in the previous Section. The functionality of each of the library cells is verified through SPICE transistor level simulations. Custom 0.25 micron technology models from Omnibase Technologies are used for the SPICE simulations.

It is assumed that three different supply voltages are present, $V1$, $V2$, and $V3$ corresponding to constant logic values of {1, 2, 3}. The fourth logic value, {0}, is represented by the ground plane and denoted $V0$. The requirement for the distribution of multiple supply voltages is a tradeoff that must be considered when using voltage-mode MVL circuitry since the reduction in wiring congestion obtained through consolidation of multi-bit signals on a single conductor can be offset by the requirement for multiple supply voltage distribution networks.

A. Single-input Circuits

Only the decisive literal gates, along with *MIN* and *MAX* gates are required for the purposes of enabling the construction of any arbitrary combinational circuit; however, the inversion circuit is also very useful in the construction of switching functions and is sometimes

*Supported by US Office of Naval Research, project N000140910784.

more efficient than using the decisive literal functions. Figure 2 contains the schematic diagram of the inverter circuit.

During the operation of the inverter circuit, *V*0 at the *IN* input produces the required bias to turn the F5 transistor ON. The output pin *OUT* is then driven by supply *V*3, which forces *V*3 at the *OUT* pin. Voltage levels *V*1 and *V*2 trigger the required bias to turn ON the transistor pairs F3-F4 and F1-F2 respectively. The output pin *OUT* is driven by the *V*2 and *V*1 supplies to respectively force voltage levels *V*2 and *V*1 to the *OUT* pin. Finally, the voltage value *V*3 at the *IN* pin forces the required bias to transistor F0 to turn it ON. This condition forces the *OUT* pin to be shorted to ground (*V*0).

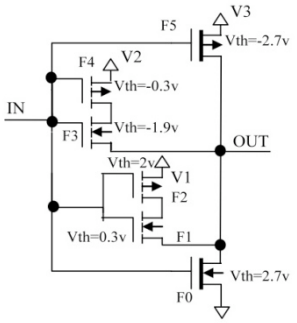

Figure 2. Quaternary Inverter Circuit

The decisive literal functions $x^{\{0\}}$ and $x^{\{3\}}$ are shown in Figure 3. The structure of $x^{\{0\}}$ is very similar to a standard binary CMOS inverter circuit. The threshold voltages of both the P-FET and N-FET are chosen to produce the quaternary $x^{\{0\}}$ operation. The $x^{\{3\}}$ circuit is constructed simply by connecting two $x^{\{0\}}$ circuits in tandem. The threshold voltages of the transistors in the $x^{\{3\}}$ circuit are selected to ensure that the circuit exhibits the required behavior.

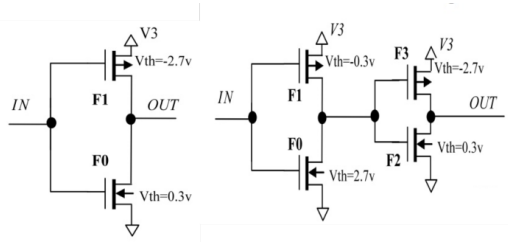

Figure 3. Decisive Literal Circuits $x^{\{0\}}$ and $x^{\{3\}}$

The circuit implementations for the $x^{\{1\}}$ and $x^{\{2\}}$ operations are structurally identical and shown in Figure 4. The difference lies in the selection of the transistor threshold voltages. In the case of the $x^{\{1\}}$ circuit, transistor F4 is turned ON only when the logic state at the *IN* input is between states *V*1 and *V*2. This forces supply *V*3 to drive the *OUT* pin to voltage level *V*3. Alternatively, in the case of the $x^{\{2\}}$ circuit, the logic state at the *IN* input has to be between states *V*2 and *V*3 to enforce a similar behavior.

B. MIN and MAX Circuits

The cell library contains two- and three-input *MIN* and *MAX* circuits. Additionally, *MIN*-inversion and *MAX*-inversion circuits are also included. Functionally, these circuits perform the quaternary inversion operation on the output of the *MIN* and *MAX* operations. The *MIN*- and *MAX*-inversion circuits actually require fewer transistors than the corresponding *MIN* and *MAX* circuits. The *MIN* and *MAX* circuits are implemented by connecting the inversion circuit to the output of the *MIN*- and *MAX*-inversion circuits. Figure 5 contains diagrams of the two-input *MIN*- and *MAX*-inversion circuits respectively.

In the operation of the *MAX*-inversion circuit, when either of the inputs is at voltage *V*1 and the other input is less than or equal to *V*1, power supply *V*2 drives the *OUT* pin. Similarly, when either of the inputs is at *V*2 and the other input is less than or equal to *V*2, the power supply *V*1 drives the *OUT* pin. The two-input *MIN*-inversion circuit in Figure 5 operates analogously to a binary-valued static CMOS two-input *NAND* gate circuit. The threshold voltages of the transistors are selected appropriately to derive the required voltage levels to the *OUT* pin.

C. Quaternary D-flip-flop

The implementation of synchronous sequential circuits such as counters and shift registers require memory elements. Flip-flops can be used as single-digit memory elements and a D-type flip-flop is included as part of the cell library. The synchronizing clock signal is binary valued with the rising edge used to write new values into the cell. The D-flip-flop is constructed through the use of *MIN*-inversion circuits and a diagram of the structure is shown in Figure 6.

The flip-flop circuit is capable of storing the four voltage values, *V*0, *V*1, *V*2, and *V*3. A rising edge (transition from *V*0 to *V*3) on the *CLK* input pin latches the voltage at the D input. The stored value is maintained at the output pin Q through the use of the internal feedback loop. The output signal $\overline{Q}$ is the *3-x* inverted value of the output Q.

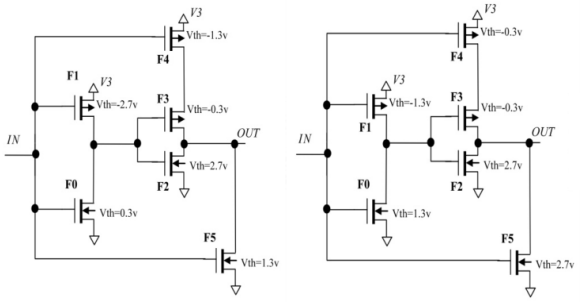

Figure 4. Decisive Literal Circuits $x^{\{1\}}$ and $x^{\{2\}}$

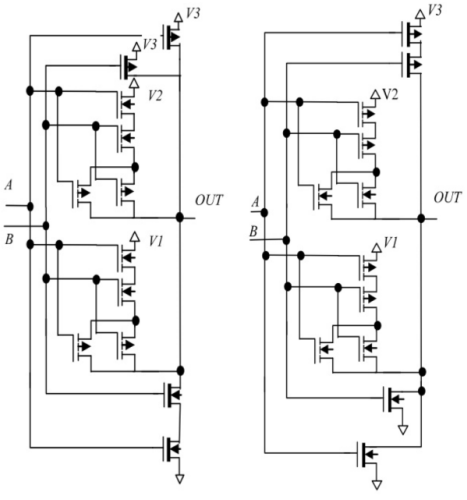

Figure 5. *MIN*-inversion and *MAX*-inversion Circuits

D. Quaternary Four-input Multiplexer

Another important function that is useful for implementing complex quaternary functions is the data selector or multiplexer. A multiplexer circuit selects one of many input signals and forwards the selected input to a single output line. In the case of quaternary logic, a multiplexer of 4^n inputs has n select digit inputs. The structure of

the multiplexer utilizes the *MIN*, *MAX*, and decisive literal circuits. Figure 7 contains a diagram of the multiplexer implementation.

The outputs of the decisive literal circuits are input to one input of four different two-input *MIN* gates with the other inputs serving as the data signal multiplexer inputs labeled as *A*, *B*, *C*, and *D*. All inputs of the decisive literal gates are connected to the multiplexer select input, *S*. The outputs of the four *MIN* gates are combined using a four-input *MAX* operation realized as a tree of two-input *MAX* gates. The output of the multiplexer, labeled as *OUT* in Figure 7, is produced by the tree of *MAX* gates.

V. Multiplication Circuit Architecture

The arithmetic multiplication operation can be viewed as consisting of two phases; generation of partial product terms, followed by the accumulation of the partial products to generate the product value. Two different multiplication circuits are designed with the chief difference being the manner in which the partial product values are accumulated. The first circuit is based on an iterative "shift and add" operation that trades performance for decreased area. The second circuit is based on a parallel type of operation intended to maximize performance. Because the parallel multiplier sums the partial products simultaneously, a multi-operand addition subcircuit is employed for partial product accumulation. The multi-operand adder requires significantly more circuitry than the two-operand adder used in the serial implementation, however, the resulting product output is produced in less time. Both multiplier circuits are fixed point and are implemented with varying operand wordsizes in order to observe scaling trends. To determine correct operation of the circuits, a validation methodology was employed as described in [29] where SystemVerilog models were created and simulated using the VCS tool.

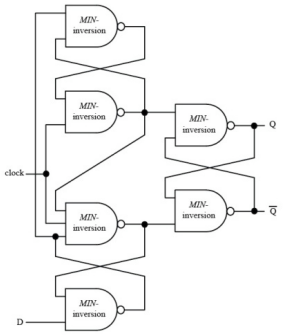

Figure 6. Diagram of D-flip-flop using *MIN*-inversion Gates

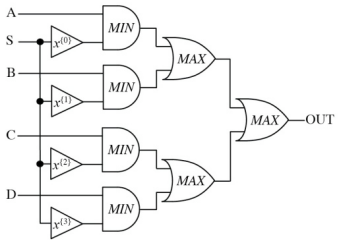

Figure 7. Diagram of 4:1 Mulitplexer

A. Serial Multiplication Circuit

Figure 8 contains a block diagram of the serial multiplication circuit configured for four-digit operands. Key components of the serial multiplier include quaternary registers (both parallel load and shift registers), single digit multiplication circuits, conditional increment circuits, and an array of HA cells configured in a carry-ripple organization. The multiplier is serial in that the partial product terms are accumulated one at a time with the shift register used to shift each intermediate accumulated value one place to the left. The intermediate shifts effectively multiply each intermediate accumulated value by the radix value of four.

Because the partial product accumulation operation is serial, the final product is available after *n* iterations occur where *n* is the maximum number of digits in each operand. The selective increment circuit is a modified version of the half-adder with the rightmost input operand serving as the value to be conditionally incremented and the leftmost operand restricted to logic values {0,1} where "1" indicates that the increment operation should be performed and "0" passes the rightmost operand value to the output.

Immediately preceding the product shift register, the six HA cells are configured in a two-operand carry-ripple adder organization. In general, any two-operand addition circuit could be used; however, in order to minimize required area, we chose to implement the carry-ripple configuration. Other types of adders would increase area while decreasing the overall delay of each iteration operation.

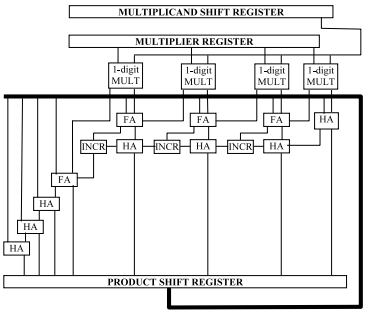

Figure 8. Diagram of 4×4 Serial Multiplier Circuit Architecture

The quaternary single-digit multiplier circuit has two single-digit quaternary inputs and produces the corresponding two-digit product output. The quaternary single-digit multiplier is implemented using the *MIN*, *MAX*, and quaternary inverter gates. The equations that describe the Most Significant and Least Significant digits (*MSd* and *LSd*) of the single-digit multiplication circuit are given in Equations (3) and (4) respectively where variables *A* and *B* represent the two input digit values.

$$MSd = 1 \cdot (A^{\{2\}} \cdot B^{\{2,3\}} + A^{\{2\}} \cdot B^{\{2\}}) + 2 \cdot A^{\{3\}} \cdot B^{\{3\}} \tag{3}$$

$$LSd = 1 \cdot (A^{\{1\}} \cdot B^{\{1\}} + A^{\{3\}} \cdot B^{\{3\}}) + 2 \cdot (A^{\{1\}} \cdot B^{\{2\}} + A^{\{2\}} \cdot B^{\{1,3\}} + A^{\{3\}} \cdot B^{\{2\}}) + 3 \cdot (A^{\{1\}} \cdot B^{\{3\}} + A^{\{3\}} \cdot B^{\{1\}}) \tag{4}$$

B. Parallel Multiplication Circuit

The entire multiplication operation is performed in a single clock cycle [31-34] in the parallel multiplication circuit. Single cycle operation is achieved since all partial products are generated simultaneously followed by summing them together with a multi-operand adder. Figure 9 contains details of the partial product generator where each partial product is generated in carry-save format consisting of a four-digit carry word and a separate four-digit sum word.

Figure 10 contains a diagram of the portion of the circuit used to combine the carry-save format of each partial product and to accumulate all partial products to form the final eight-digit product word. The form of the array of full-adders (FA) and half-adders (HA) in Figure 10 is described in [33] and the internal configuration of the FAs and HAs is described in [30]. Inputs to the addition array are denoted by $n[s,c]i$ where i represents the i^{th} digit of the n^{th} partial

product and [*s*,*c*] indicates whether the digit is from the carry or sum word of each partial product.

VI. Performance Analysis

The logic library gates and multiplier circuits are modeled and verified using the SystemVerilog language to validate proper functionality. Additionally, functionality of the library cells is verified at the transistor level using the HSPICE tool.

Characterization of the power dissipation, area, and performance of the multipliers is accomplished by benchmarking the circuits with their binary equivalents for operand wordsizes of 16, 32, and 64 quaternary digits. The corresponding binary circuits are synthesized using the Synopsys design compiler synthesis tool for 32, 64, and 128 bit operand word sizes. A 130 nm standard cell library from Texas Instruments is used for the binary circuits. It is noted that state-of-the-art synthesis optimization algorithms are used for the binary cases; however, no such synthesis tools are available for quaternary circuit synthesis hence the multiplier architectures are designed manually.

The benchmarking comparison results for the multiplier circuits are summarized in Table I.

A. Multiplier Circuit Area

Area is compared in the third and sixth rows of Table 2 and is measured in units of transistors. Area due to wiring interconnects is not included since the circuits were not placed and routed in physical design. The number of internal signal conductors is reduced by one-half in the quaternary circuits as compared to the binary case; however, there are also two additional supply voltage distribution networks required to supply the *V*1 and *V*2 rail voltages for the quaternary circuits. Based on this observation, it is anticipated that area due to interconnect networks will be approximately equivalent. In the case of the serial multiplication circuits, an average of 27% more transistors are required for the quaternary implementation. However, the parallel multiplier circuits resulted in a decrease of 14%, 24%, and 33% in transistors respectively for the increasing operand wordsizes. The reason that the area decreases as wordsize increases in the parallel case is due to the exponential growth in the number of transistors with respect to wordsize (in units of digits) and the corresponding wordsize is smaller for the quaternary circuits.

TABLE I. Benchmark Results of Mulitplier Circuits

		Quaternary			Binary		
	Wordsi	16	32	64	32	64	128
Serial	Area	1061	20731	40955	8238	16380	32654
	Power	90.2	176.2	348.15	154.3	306.8	611.77
	Levels	134	262	518	189	381	765
Parallel	Area	7505	28205	10559	8559	34974	14047
	Power	0.44	1.69	4.89	0.91	3.91	12.63
	Levels	240	442	806	332	638	1594

B. Multiplier Circuit Dynamic Power Dissipation

Rows four and seven of Table I compare the dynamic (switching) power dissipation among the binary and quaternary multipliers. For the binary circuits, dynamic power analysis was accomplished through the use of the Synopsys PrimeTime-PX tool with the assumption of a 50% switching activity of all internal signals.

The quaternary circuit dynamic power dissipation is estimated by using Equation (5) and techniques in [35, 36].

$$P_{\text{dyn}} = C_{\text{tot}} \times a_{sw} \times (V_{avg})^2 \times f \tag{5}$$

C_{tot} represents the total capacitance estimated from the total number of transistors in the circuit. Total capacitance of the binary circuits is estimated by multiplying the number of equivalent two-input *NAND* gates with their input capacitance as shown in Equation (6). Correspondingly, Equation (6) is also used for the quaternary circuits using two-input *MIN* gate equivalents.

$$C_{\text{tot}} = N_{gates} \times C_{gate} \tag{6}$$

V_{avg} is the average voltage swing occurring during a transition. The quaternary transistor models utilize a 3.3V supply with 1.1V swings between the voltage encoded logic levels of *V*0, *V*1, *V*2, and *V*3. Assuming all logic transitions are equally likely for consistency with the binary case, there are 12 cases when voltage swings occur ranging from swings of 3.3V to 1.1V. Therefore V_{avg}=1.83V.

a_{sw} represents the switching activity and is the probability that a non-zero voltage transition occurs within a given clock cycle. Each quaternary signal conveys the equivalent of two bits of information per conductor, thus the equivalent switching activity factor is 75%. The 75% switching factor arises from assuming the quaternary signal is equally likely to transition from its present logic level to any other within a clock cycle. Because there are a total of 12 non-zero transitions out of 16 possibilities, a_{sw} =75%.

The term *f* refers to the frequency of operation. The frequency used for the power calculations is derived using the Equation (7).

$$f = \frac{1}{2 \times T_{\max}} \tag{7}$$

The $T_{\max}$ term is obtained from the timing analysis results of the equivalent binary circuits. The factor two is used to accommodate enough margin for proper operation of both binary and quaternary circuits.

The serial multipliers indicated on average a 74% decrease in power dissipation for the quaternary case versus the binary circuits. In terms of the parallel multipliers, a very significant decrease in power dissipation is also observed averaging 132%.

C. Multiplier Circuit Performance

Rows five and eight of Table I estimate delay by measuring the maximum depth path in each circuit in terms of the number of transistors. In all cases, the quaternary multiplication circuits show reduced maximum pathlength when compared to the binary circuits. In terms of the serial multipliers, the quaternary maximum path length was, on average, reduced by 31%. In terms of the parallel case, similar path length decreases observed are 28%, 31%, and 50% respectively for the 16-, 32-, and 64-digit multipliers as compared to the equivalent binary circuits.

VII. Conclusions

The quaternary logic system is easily interfaced to binary circuitry through the use of voltage comparator circuits since the radix value of four is a power of two. A library of basic SUSLOC cells is described and is proven to form a functionally complete set of operators over the four-valued switching algebra. Implementations of serial and parallel fixed-point multiplcation circuits based on the cell library are presented and analyzed. The analysis results indicate that the multiplier circuits have decreased dynamic power dissipation characteristics and improved timing performance when compared to binary circuits with equivalent wordsizes.

Acknowledgments

The authors acknowledge and thank the Synopsys Corporation for the use of EDA tools donated to Southern Methodist University. In particular, the HSPICE, Design Compiler, PrimeTime, and SystemVerilog (under the VCS environment) tools were essential for this work. We also acknowledge and thank Omnibase Technologies

for supplying the 0.25 micron transistor models used for simulating the quaternary library cells.

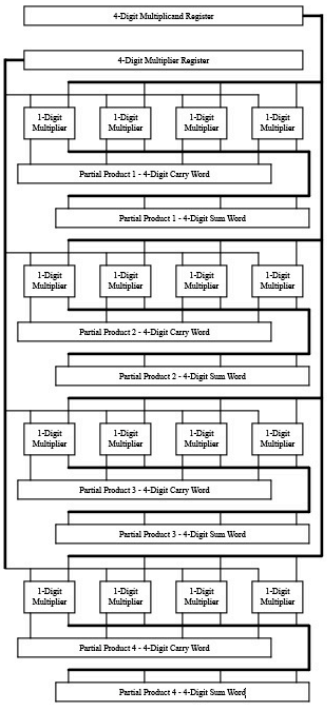

Figure 9. Diagram of 4×4 Partial Product Generator

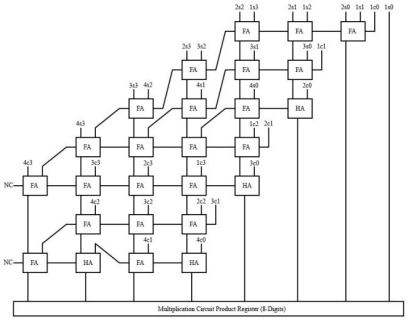

Figure 10. Diagram of 4×4 Partial Product Accumulation Array

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

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

[4] W.-S. Chu and W. Current, "Current-mode CMOS quaternary multiplier circuit," *Electronic Letters*, vol. 31, iss. 4, Feb. 16, 1995, pp. 267-268.

[5] J.T. Butler, (editor), **Multiple-Valued Logic in VLSI Design**, *IEEE Computer Society Press*, 1991.

[6] H.A. Castro, et al., "A 125 MHz burst mode 0.18µm 128 Mbit 2 bits per cell flash memory," *Proc. of VLSI Symp. Technology Circuits*, 2002.

[7] D. Entieble, and M. Israel, "Comparison of binary and multivalued ICs according to VLSI criteria," *IEEE Computer Magazine*, pp. 28-42, 1988.

[8] M. Kameyama, S. Kawahito, and T. Higuchi, "A multiplier chip with multiple-valued bidirectional current-mode logic circuits," *IEEE Computer Magazine*, pp. 43-56, 1988.

[9] K.C. Smith, "Multiple-valued logic: A tutorial and appreciation," *IEEE Computer Magazine*, pp. 17-27, 1988.

[10] The Intel Corporation, Intel® StrataFlash™ Memory Technology, Application Note AP-677, 1998.

[11] Y. Yasuda, Y. Tokuda, S. Zhaima, K. Pak, T. Nakamura, and 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.

[12] J. Lukasiewicz, *tr. on three valued logic*, Ruch Filozoficzny, vol. 5, pp. 169-171, 1920.

[13] E.L. Post, "Introduction to a general theory of elementary propositions," *American Journal of Mathematics*, June 1921.

[14] S.C. Kleene, "On a notation for ordinal numbers," *The Journal of Symbolic Logic*, vol. 3, pp. 150-155, 1938.

[15] M. Cohn, "Switching function canonical forms over integer fields," *Ph.D. Thesis*, Harvard University, 1960.

[16] D.K. Pradhan, "A multivalued switching algebra based on finite fields," *Proc. of IEEE Int. Symposium on Multiple valued Logic*, 1974, pp. 95-111.

[17] W. Quine, "A way to simplify truth functions," *Amer. Math. monthly*, vol. 62, pp. 627-631, 1955.

[18] J.C. Muzio and D.M. Miller, "On the minimization of many valued functions," *Proc. of IEEE Int. Symp. Multiple-Valued Logic*, 1979.

[19] W.-S. Chu and W. Current, **"**Quaternary multiplier circuit," *Proc. of IEEE Int. Symp. On Multiple-Valued Logic*, 1994, pp.15-18.

[20] O. Ishizuka, A. Ohta, K. Tannno, Z. Tang, and D. Handoko, **"**VLSI design of a quaternary multiplier with direct generation of partial products," *Proc. of IEEE Int. Symp. On Multiple-Valued Logic*, 1997, pp.169-174.

[21] 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.

[22] U.S. Patent, *6,133,754,* Edgar Danny Olson, inventor, Multiple-valued logic circuit architecture; supplementary symmetrical logic circuit structure (SUS-LOC).

[23] D. Olson and K.W. Current, "Hardware implementation of supplementary symmetrical logic circuit structure concepts," *Proc. of IEEE Int. Symp. On Multiple-Valued Logic*, pp. 371-376, 2000.

[24] M. Aline, T. Saïdi, E. Kinvi-Boh, O. Sentieys, and E.D. Olson, "Design and characterization of a low power ternary DSP," http://www.omnibaselogic.com/resources/TernaryDSP.pdf.

[25] D. Olson and K.W. Current, "Demonstration of supplementary symmetrical logic circuit structure concepts using a MOS test chip," *International Journal of Multiple-Valued Logic*, vol. 7, pp. 1-23, 2001.

[26] M. Aline, T. Saidi, E. Kinvi-Boh, O. Sentieys, and E.D. Olson, "Design and characterization of a low power ternary DSP," *Proc. of IEEE International Signal Processing Conference*, 2003.

[27] E. Kinvi-Boh, M. Aline, O. Sentieys, and E.D. Olson, "MVL circuit design and characterization at the transistor level using SUS-LOC," *Proc. of IEEE Int. Symp. on Multiple-Valued Logic*, 2003.

[28] G. Epstein, **Multiple-Valued Logic Design: An Introduction**, *IOP Publishing Ltd*, 1993.

[29] M. Amoui, D. Grosse, M.A. Thornton, and R. Drechsler, "Evaluation of toggle coverage for MVL circuits specified in the SystemVerilog HDL," *Proc. of IEEE Int. Symposium on Multiple valued Logic*, 2007, Session 8B, paper 2.

[30] S. Datla, M.A. Thornton, L. Hendrix, and D. Henderson, "Quaternary addition circuits based on SUSLOC voltage-mode cells and modeling with SystemVerilog," *Proc. of IEEE Int. Symposium on Multiple valued Logic*, 2009, pp. 256-261.

[31] B. Parhami, **Computer Arithmetic Algorithms and Hardware Designs**, *Oxford University Press*, 2000, pp. 143-145.

[32] M. Ercegovac and T. Lang, **Digital Arithmetic**, *Morgan Kaufmann Publishers*, 2004.

[33] S.D. Pezaris, "A 40-ns 17-bit array multiplier," *IEEE Trans. On Computers*, vol. C-20, pp. 442-447, 1971.

[34] 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.

[35] F. N. Najm, "A survey of power estimation techniques in VLSI circuits," *IEEE Transactions on VLSI Systems*, 2(4):446–455, 1994.

[36] T. Ishihara, H. Yasuura, "Experimental Analysis of Power Estimation Models of CMOS VLSI Circuits," *IEICE Trans. Fundamentals*, vol. E00-A, No. 6, June 1996.