Asian Journal of Engineering and Applied Technology (AJEAT)
Vol.3.No.1 2015 pp 13-19
available at: www.goniv.com
Paper Received : 08-04-2015
Paper Accepted: 20-04-2015
Paper Reviewed by: 1. R. Venkatakrishnan 2. R. Marimuthu
Editor : Prof. P.Muthukumar

A NOVEL IMPLEMENTATION OF DIGITAL CIRCUITS USING MVL FUNCTION

KAMAKIYA V., ANBARASU .R
Dept of VLSI Design
MAM College of Engineering
Trichy

ABSTRACT

The design of Multiple Valued Logic (MVL) digital circuits is performed by increasing the representation domain from the two level (N=2) switching algebra to N > 2 levels. Universal sets of MVL CMOS gates allow the synthesis and implementation of any MVL digital circuit. This paper proceed with: 1) the design and implementation of a universal set of IC gates, CMOS 0.25 um technology, that carry out extended AND operators: eANDl, eAND2, eAND3, Successor (SUC), and Maximum (MAX) operators to perform synthesis of any MVL digital circuits; and 2) the synthesis of an MVL multiplexer circuits based on the IC MVL gates. Implemented circuits demonstrate correct functionality of the implemented gates and feasibility of the MVL combinatorial circuit design.

Keywords: Multi valued logic.

I.INTRODUCTION

Synthesis of digital circuits is performed in the well-known two level logic *(N* = 2) switching algebra, *D* = {0, l},where *D* is the domain of the numerical representation *N*. By increasing the domain of the digital representation to *N* levels *D* = {0,1,2,3,..., *L* = *N* — 1} , it is possible to design MVL (Multiple Valued Logic) circuits. The current trend in Integrated Circuits (IC) is to embed multiple systems into a single IC, known as System On a Chip (SoCs) leading to, among other things, an increment in the quantity, the delay time, length, and complexity of the interconnections.

The multiple valued logic is a viable alternative to cope with the interconnections issues as they decrease the number of the interconnections *M* as the inverse of the Log_2M. This reduction in the area of the IC devoted to the interconnections has motivated many MVL proposals. In a universal set of gates suitable for CMOS implementation is presented. This paper addresses the first drawback of the MVL digital circuits synthesis as follows: 1) the design and implementation of a universal set of IC gates based on the CMOS 0.25 *um* technology, comprised of extended AND operators: eAND1, eAND2, eAND3, Successor (SUC), and Maximum (MAX) operators to allow synthesis of MVL digital circuits and the implementation of the circuits that are the basis to build the gates. These five CMOS gates are the universal set of the MVL algebra for *N* = 4 levels with domain (0,1,2,3); and 2) the design methodology will be applied to the synthesis of MVL multiplexer circuit to illustrate utilization of the proposed ICs. The timing results demonstrate correct functionality of MVL IC gates and feasibility of the MVL combinatorial circuit designs. The proposed gates allow designing any MVL digital circuit taking advantage of the knowledge coming from the binary circuits. This paper presents the implementation of a novel universal set of MVL gates, initially proposed in voltage mode utilizing CMOS technology. The main contributions of this paper are:

- Design of MVL IC gates and a set of experiments and results discussing the features of the implemented gates;
- Design and synthesis case studies, based on the proposed MVL algebra, on combinational (multiplexer) digital circuits.

The rest of this paper is organized as follows. Section II presents MVL principles. Section III describes the algebra supporting the MVL methodology. Section IV presents the methodology applied to combinatorial circuit synthesis. Section V addresses the implementation of the proposed IC MVL gates. Section VI presents the results of the implementations of the MVL digital circuits with the IC gates. Section VII discusses the methodology by pointing out its advantages and limitations. Finally, Section VIII summarizes the concluding remarks and future work.

II. MVL PRINCIPLES

The n-variables MVL function is a mapping f $:D^n$ —► *D*. The MVL function represents values in the domain *D* = {0,1,2,3,...,!/ = *N* — 1} depending on the assigned values to the n inputs variables. A unique representation in tabular form, named truth table, can be obtained. In order to synthesize any given MVL function, the set of operators of the MVL algebra must be functional completeness or weak functional completeness. A possible approach to define a universal set of operators (functional completeness) is to find a canonical form, analogously as: Sum of Products (SOP) or Products of Sums (POS) that is a unique representation of the function, as it was done for the binary algebra.

In this context, the proposed universal set of gates corresponds to the practical implemented IC gates of the universal set of operators of the MVL algebra. As each variable may be used an arbitrary number of times as a primary input and the canonical Sum of Extended Products(SOEP)form is defined, the five operators (eAND1, eAND2, eAND3, SUC, MAX), already mentioned in the introduction, define a universal set under the proposed algebra for quaternary logic.

Next step for the synthesis is related with the gates design: CMOS current mode and CMOS voltage mode. The voltage mode, here proposed, leads to the problem of how to define TV voltage levels to discriminate the logic reference levels to define the CMOS inverters threshold values. To cope with this issue under the chosen technology, some characteristics must be defined, as for example: frequency response, fan-in, fan-out, power consumption, symmetrical good noise margins that has an effect on the width/length relation between the PMOS and NMOS gates, and how to choose the reference voltage to discriminate the n logic levels for the MVL, among others.

First basic ideas on MVL come from ternary MVL proposed by Lukasiewicz in his logic. He claimed that the three-valued (ternary) logic is as consistent and free of contradictions as the two-valued logic. Three-valued logic is utilized to design ternary circuits with domain either with *D* = {0,1, 2} or balanced ternary with *D* = *{—1* , 0,1} . If balanced ternary logic (—1,0,1) is used, the same hardware may be used for addition and for subtraction. In a ternary Post algebra, the literal, the AND, and the OR form a complete set of operations, and also, the literal and the NAND operations form a complete set. In ternary for the proposed algebra the universal set is eAND1, eAND2, SUC, MAX. Quaternary circuits have the advantage that a four-valued signal can easily be transformed into a two-valued signal.

Then, to define an algebra, convenient to use and easy to learn, with a well-known methodology, feasible to implement from the algorithmic (minimization tools) and gates (IC CMOS hardware) point of views, a suitable criteria is to extend well known concepts of the binary

TABLE I

MAX(X,Y) OPERATOR (LEFT); SUC(X) OPERATOR (RIGHT)

X\Y	0	1	2	3
0	0	1	2	3
1	1	1	2	3
2	2	2	2	3
3	3	3	3	3

X	X^1	X^2	X^3
0	1	2	3
1	2	3	0
2	3	0	1
3	0	1	2

TABLE II

EXTENDED AND1 OPERATOR $X *^1 Y$ (LEFT); EXTENDED AND2 OPERATOR $X *^2 Y$ (RIGHT); EXTENDED AND3 OPERATOR $X *^3 Y$ (BELOW)

X \ Y	0	1	2	3
0	0	0	0	0
1	0	1	0	0
2	0	0	0	0
3	0	0	0	0

X \ Y	0	1	2	3
0	0	0	0	0
1	0	0	0	0
2	0	0	2	0
3	0	0	0	0

X \ Y	0	1	2	3
0	0	0	0	0
1	0	1	0	0
2	0	0	0	0
3	0	0	0	3

switching algebra. This is the approach adopted in this work which is detailed with examples for quaternary MVL in the next sections.

III. MVL ALGEBRA

The closed Multiple Valued Logic algebra is an ordered set with domain D = {0, 1, 2, 3, ..., $L = N - 1$} in which acting two binary operators (+, $*^l$) (Maximum and Extended AND, respectively) and one unary operator (Successor) as defined below, with the lower element 0 and the upper element $(L = N - 1)$.

Maximum operator of (X, Y) denoted by the symbol $X + Y$. By definition, If $X > Y$ then $X + Y$ = X; otherwise $X + Y = Y$. X, Y E D as shown in Table I in the quaternary logic $D = \{0, 1, 2, 3\}$. Maximum operator is introduced by Lukasiewicz.

Successor operator of (X) denoted by the symbol X^1.By definition, $X^1 = Y$, X, Y *element D*, where Y is the next element from the element X in the cyclic ordered set D. The Successor operator is presented by Post as operator cycle clockwise (right shifter). It is an operator of only one argument, as shown in Table I in the quaternary logic $D = \{0, 1, 2, 3\}$. For notation purposes, the Suc (X) operator is denoted by the symbol X^1 and Suc(Suc (X)) is denoted by X^2. Note that the Successor operator can also be calculated as$(a + n)$MOD N, V a, n G.D, where a represents the input value, n the number of times that the Successor operator is applied and MOD stands for the Modulo operator.

Extended AND operator of (X, Y) denoted by the symbol $X *^{Const} y$. By definition, If X = Y = *Const* then $X *^{Const} Y$ = Const;otherwise $X *^{Const} Y = 0$. *Const*, X, Y G D as shown in Table II in the quaternary logic $D = \{0,1,2,3\}$. One can observe that, based on these operators, for an N levels *(base = N)* MVL algebra, the number of universal set of operators is $N + 1$ given by the extended *AND:* eAND1, eAND2, eAND3,..., eAND $(N - 1)$,Successor(SUC),and Maximum (MAX) operators.

IV. SYNTHESIS OF MVL DIGITAL CIRCUITS

The methodology presented in and here replicated, briefly, is intended to follow the knowledge coming from the binary circuits synthesis and simplification. Details of binary synthesis can be found in. The methodology will be applied to the multiplexer synthesis, as follows.

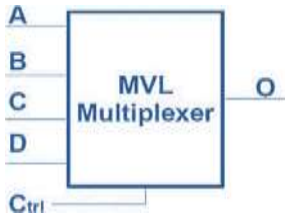

Fig: 1. Multiplexer Block diagram

A. MVL Quaternary Multiplexer Synthesis By designing an MVL multiplexer circuit adopting an N=4 MVL algebra D={0,1,2,3}, function.

The MVL multiplexer has five $(A, B, C, D, Ctrl)$ MVL inputs and one MVL output (O). *Ctrl* is the control input that controls the output. The multiplexer behavior is described as follows:

TABLE III

TRUTH THE QUATERNARY 4:1 MULTIPLEXER

Control input: ***Ctrl***	Output ***O***
0	***A***
1	***B***
2	***C***
3	***D***

- If Ctrl = 0 then O = A;
- If Ctrl = 1 then O = B;
- If Ctrl = 2 then O = C;
- If Ctrl = 3 then O = D;

V. IC MVL GATES IMPLEMENTATION

The main purpose of the IC implementation of this universal set is to prove concepts and feasibility for the discrete combinatorial circuits synthesis implementation utilizing the MVL proposed algebra. Therefore, main concerns are the functionality and the MVL circuit design process, so that, optimization of the IC is not the main objective. Then, a direct and straight forward implementation of these gates is proposed. Basically, based on the inputs, a comparison with different voltage

thresholds to identify the four quaternary levels is performed. The comparator output goes to a circuit that defines the logic to control switches that set the output to the four quaternary levels according to the gate behavior. This direct approach is followed by all the MVL implemented gates, as presented next.

The implementations of the eAND1, eAND2, eAND3, SUC, and MAX gates are performed in the 0.25 um CMOS technology, and divided into three circuit stages:
1) a discriminatory circuit for identifying the input into the four logical references (0, 1, 2, and 3) for the quaternary domain;
2) a binary logic circuit (domain 0, 1) for performing logical operations to control the switches and;
3) a set of switches for setting the output voltage. A voltage CMOS divider is designed to set the logic voltages in the output.

For notation purposes in the quaternary *(N* = 4) MVL gates implementation, the four logic levels are denoted in the algebra as 0, 1, 2, 3, and these logic levels are denoted as $0_q, l_q, 2_q, 3_q$ in the IC circuit implementation that are represented by voltage intervals, as shown in Fig. 5 and (4). For example, l_q is represented in the IC for a voltage between 0.7 V and 1.4 V. The 0.7V, 1.4V and2.2 Vthresholdsin the vertical axis corresponds with the *VTH* threshold in that illustrates the standard inverter logic gate behavior used as a comparator in which the discriminatory circuit is based on.

The circuits implementation are partially binary circuits levels 0_q and l_q, with voltage levels 0 V and 3.3 V, respectively. As shown in Fig. if the input voltage is less than *VTH* the output of the comparator is at 1& level, represented as 3.3 V, that is also the circuit polarization voltage *Vdd* = 3.3 V. Note that the voltage level of *Vdd* = 3.3 V represents 1_q and 3_q,. This strategy simplifies the gates implementations.

$$D_{IN}=f(V_{IN}) = \begin{cases} 0_q & \text{if } 0V < V_{IN} < 0.7\ V \\ 1_q & \text{if } 0.7\ V < V_{IN} < 1.4V \\ 2_q & \text{if } 1.4V < V_{IN} < 2.2\ V \\ 3_q & \text{if } 2.2\ V < V_{IN} < 3.3\ V \end{cases}$$

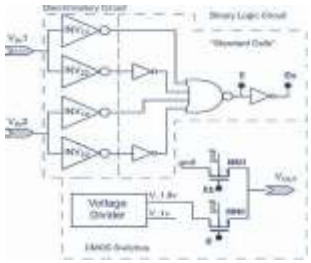

Fig. 3. Schematic of the

In this particular design and technology, 0_Q, l_q,2_q threshold voltage levels are evenly spaced and the 3_Q level has a range of 1.1 V (from 2.2 V to 3.3 V). To put a value greater than the threshold *VTH* = 2.2 V in order to evenly spaced all threshold voltages, the PMOS transistor width *(W)* must be changed, at least, from W_p = 10 *fimto* W_p = 60 jim in the CMOS inverter, according to the simulation results. So, higher threshold voltage requires larger PMOS transistor and unpractical design *(Wp* oc *VTH)* In the comparator, the transition voltages are determined by the size relation $(W/L)_P$ and $(W/L)_N$ of the NMOS and PMOS transistors with dimensions *W* and *L*.

Table summarizes the obtained dimensions *(W, L)* for the NMOS and PMOS transistors for each inverter logic gate; where the labels *INVQ-J,* JjVVi4,and *INV22* represent the inverter logic gates with thresholds 0.7 V, 1.4 V, and 2.2 V, respectively.
The output of !W07 is U if VJJV < *VTH* = 0.7 V and 0_b otherwise. The output of *INV14* is 1& if VJJV < VTH = 1-4

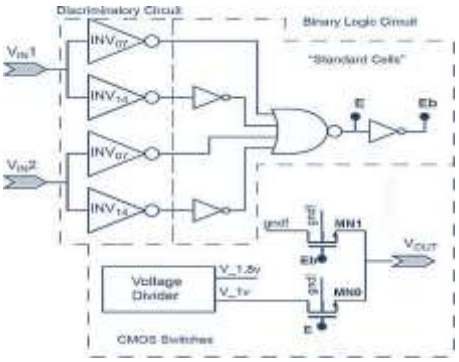

Fig.2. Schematic of the eAND1

V and 0& otherwise. The output of *INV22* is It if V/JV < *VTH* = 2.2 V and 0& otherwise.
Therefore, a unique quaternary digit *(DIM)* in the {0, l, 2g, 3g} domain is set by comparing the input voltage against the corresponding threshold in the (4), as follows:
DIM is 0g if the output voltage of the inverter *INV07* is 1_0 (3.3 V);
DIM is l_g if the output voltage of the inverter *INVQJ* is 0& (0 V) and the output voltage of the inverter *INV\4* is lj, (3.3 V);

DJN is 2g if the output voltage of the inverter *INV14* is 0_o and the output voltage of the inverter INV_{22} is 1_0;
DIM is 3_g if the output voltage of the inverter INV_{22} is 0&.

The output of the inverter logic gate is the input of a logic circuit that controls a set of switches, that enable or disable according to the gate logical operation. Each switch interconnects the gate output with the correspondent voltage reference quaternary digit, where their voltage values are: 0_q = *VRO* = 0 V, 1, = V_{Rl} = *1 V*, 2_q = *Via* = 1.8 V,and 3_ = V_M = 3.3 V, as shown in Fig. 5. These basic elements are the basis for all gates implementation. Specifically, the eAND1 gate, in Fig. 2, has two inputs Vj/vi and *V1M2* and one output *VQUT* taking two possible

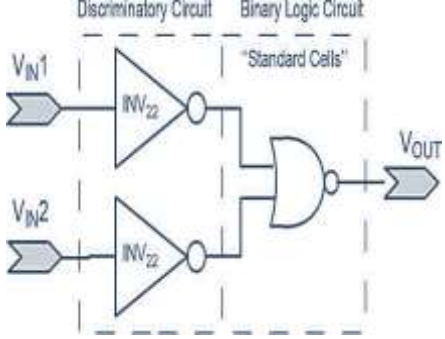

Fig. 4.Schematic of the eAND3

output values (0g and 1,). V_{QUT} = lg if VJ-JVI = lg and V_{IN2} = lg, and V_{OUT} = 0_G otherwise.
Then two sets of inverters with output $V_{OUTI_{INV07}}$ and $V_{OUTI_{INV11}}$ receiving both, the input 1; and $V_0UT2!_{nvo7}$ and $VouT2_{INV14}$, receiving both, the input 2; the circuit generates *E* and *Eb* signals controlling the two switches to set the output voltage in $V_{RO} = 0$ V (0,) or V_{RI} = 1 V (l_g). Signals E and *Eb* set, respectively^g and 0_g,where:

$E = NOT(V_{OUTI_{INVOT}}$ OR $NOT\{V_{OUTUNV_{14}})$ OR V_OUT2_{INV07} ORNOT($F_{or7T2j)VV14}$))
Eb = NOT(.B). NOT stands for the Complement binary operator and OR stands for the OR binary operator.

A Voltage Divider is designed for establishing the two reference voltages V_{RI} = 1 V and VR_ = 1-8 V (the ground and *Vdd* = 3.3 V sources are used for VRO and V_{R3}, respectively). This solution is enough in order to test the functionality of the IC gates.
Similar analysis is performed for the eAND2 gate implementation but changing the inverters to *INVu* and *INV22*, additionally, the output voltage is changed to Og(VRo = 0 V) and 2g

$\{V_{R2}$ = 1.8 V), as shown

in Fig. 3. The schematic of the eAND3 gate is shown in Fig. 4. The two inverters *(INVI22* and *INV222)* identify the quaternary digits 3g on its inputs and the binary NOR gate defines the logic operation and interconnects (internally) the gate output with the reference voltage VRO or VR_. This simple implementation is possible because the voltage values for 0_g and 3_g are equal to 0{, and 1_, respectively.
The schematic of the SUC gate is shown in Fig.5. The SUC gate has one input VJJV and one output V_{OUT} with four possible output values (0_g, l_g, 2g and 3_g). The full set of inverters (J./VV07, JiV"Vi4,and *INV22)* and four switches are needed to identify all quaternary digits on its input and to set the output voltage in
VR_O or VR_I or VR_2 or VR_3. The MAX gate has two inputs VJJVI, V_{INI} and one output V_{OUT} with four possible output values (0_q, l_q, 2_q,and 3_q).
For the implementation of the MAX operator,
Fig. 6, the comparison is performed in current mode, then a voltage to current converter to enable the operation in current mode to connect to both inputs, and their outputs go to a current comparator (outputs in binary voltage mode) to determine the input with the highest value. Finally, two switches connect the output with the highest input controlled by *E* and *Eb* binary outputs.

VI. Results of the MVL Digital Circuits With the Implemented Gates

To illustrate the obtained results two experiments have been performed:
• the characterization for each MVL gate by measuring functionality, response time, DC Fan-out, and maximum power consumption;

MVL Circuits Applications
In this section, two different digital circuits (a 4:1 multiplexer and an SR latch) have been designed using the gates described in Section V.
1) Implementation and Results of the 4:1 Multiplexer: To build the circuit with the available gates, the multiple input MAX gates are implemented by using multiple two input gates.
a) To verify the multiplexer output, two inputs with all possible MVL digital combinations are generated (00,01,02, . . . , 33).
b) The signal with the lower frequency is connected to the *Ctrl* input of the multiplexer.
c) The signal with the higher frequency is connected to the *A* input.
d) The correct result shows $(O = A^0)$ when *Ctrl* = 0, (O = A^1) when *Ctrl* = 1, $(O = A^2)$ when *Ctrl* = 2, and $(O = A^3)$ when *Ctrl* = 3.
The timing results in table 1.4.1 demonstrate the correct functionality of the 4:1 multiplexer, channel 1 shows *Ctrl*, channel 2shows *A* and channel 3 shows (O). The Figure shows: *O* = *A* when *Ctrl* = 0; *O* = *B* when *Ctrl* = 1; *O* = *C* when *Ctrl* = *2 ; 0* = *D* when *Ctrl* = 3, as expected.

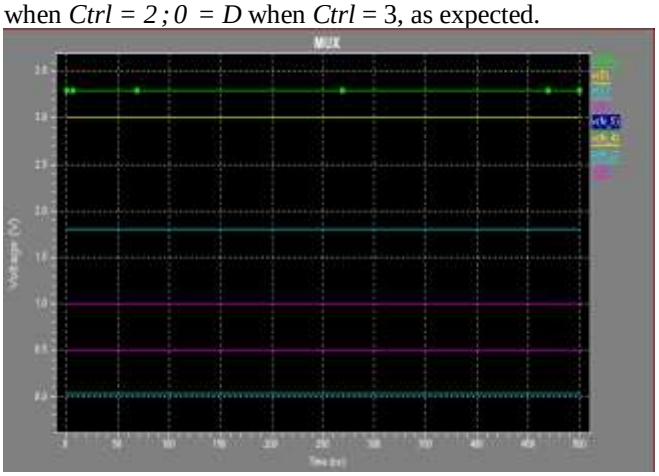

Fig. 7.Multiplexer Output Waveform.

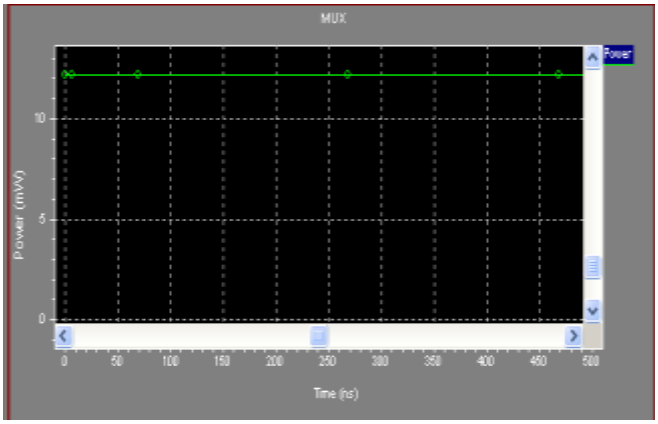

Fig. 8. Multiplexer power output.

VII. Results Discussion

The delay on the integrated circuit has been established according to the maximum output current and by the equivalent capacitance on the MVL gate output (output rails capacitance) PADs capacitance, protoboard rails capacitance, and further blocks capacitance. This behavior has been observed by the rise time on MVL gates output. Table XI shows the response time for each MVL gate and to enable a comparison between simulation and experiment.

The synthesis of any MVL circuit is feasible with the proposed gates: Successor, Maximum, and Extended ANDs. The synthesis methodology is quite akin to the binary methodology so that the process is straight for a binary designer. This is a good characteristic as it simplifies the MVL design learning curve. The electronic circuit for the implementation of the gates does not present any special challenge as those are well-known concepts to design electronic circuits. The main purpose ofthe gates is to

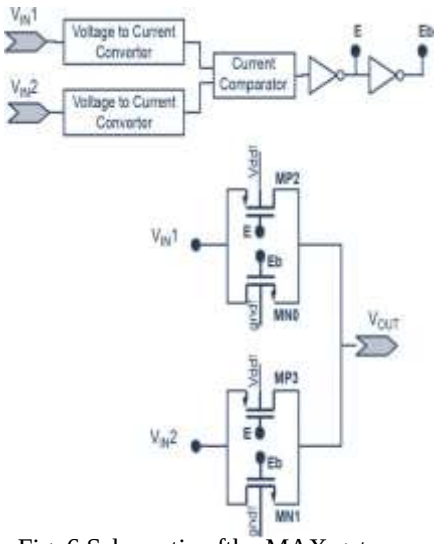

Fig. 6.Schematic ofthe MAX gate.

prove feasibility of the methodology from the algebra to the implementation of the MVL circuits. However, the gates are a subject for further improvement. The internal peaks presented in the gates, as shown in Fig. 24 in the transitions, eventually, can produce difficulties in the design of practical circuits, especially in memory systems. Additionally, for scaling up the number of gates, the MAX gate is not adequate because it connects the input directly to its output. Then, it inhibits scaling up the number of gates.

Since only 40 IC gates have been built, the case studies are very simple circuits that illustrate feasibility of MVL circuit design with the proposed algebra. For a high gate density and fast circuit, the reference voltages must be generated by external sources. Two fold issues must be addressed further, first, to improve IC gates electronic design to avoid internal peaks and to pay attention of frequency, delay, fan-in and fan-out characteristics, among others. And second, to develop minimization (MVL states and gates) software tools to permit practical circuits with thousands of MVL gates. As the MVL SR-latch definition is similar to the binary circuits, all others latches (Delay, Toggle, *J-K)* can be defined and implemented in a simple way following analogous definitions as in the binary design.

VIII. Concluding Remarks and Future Work

In this work an alternative algebra and methodology for the design of digital circuits based on an MVL algebra have been proposed. The operators of our proposed algebra comprise of a universal set of gates. The operators: extended *AND* (eAND1, eAND2, eAND3,

Successor (SUC), and Maximum (MAX) have been implemented to illustrate the design of any Multiple-Valued Logic (MV Logic) digital circuit. The design methodology has been illustrated for the MVL algebra for $N = 4$ levels with domain (0,1,2,3) for the synthesis of the MVL multiplexer and the latch memory circuits. The timing signals demonstrate correct functionality of MVL IC gates and feasibility of the MVL combinatorial (multiplexer) and memory circuit (latch) design. The proposed gates allow designing of any MV Logic digital circuit taking advantage of the knowledge coming from the binary circuits by extending it to the MVL digital circuit synthesis.

Further work will be performed in order to improve electronic gates characteristics such as frequency, delay, fan-in, and fan-out. Future work is also related to the design and implementation of minimization (MVL states and gates) software tools to permit practical circuits with thousands of MVL gates and new electronic gates with few transistors, lower power dissipation, and using less silicon area than the presented gates.

REFERENCES

[1] Shweta Hajare , P.K.Dakhole and Manisha Khorgade, "Design of Gates in Multiple Valued logic," in Proc.Int. Conf.on Recent Trends in Information,Telecommunication and Computing,ITC.

[2] Vasunara Patel.K.S, Gurumurthy.K.S "Arithmetic operation in Multi-Valued logic," International journal of VLSI design & Communication Systems, Vol.1, NO.1 ,March 2010.

[3] Vasunhara Patel.K.S, Gurumurthy K.S, Vinay Sheshadri, Sivaram "Static Random Access Memory Using Quaternary D Latch," International Journal of Engineering Science ant Technology,Vol.2(11),2010,6371-6379.

[4] C. Lazzari, P. Flores, and J. Monteiro, "Power and delay comparison of binary and quaternary arithmetic circuits," in Proc. 3rd Int. Conf. Signals, Circuits and Syst., 2009, pp. 1–6.

[5] M. E. R. Romero, E. M. Martins, and R. Santos, "Multiple valued logic algebra for the synthesis of digital circuits," in Proc. 39th IEEE Int. Symp. Multiple-Valued Logic, Los Alamitos,

CA, USA, 2009, pp. 262–267. [6] Dusanka Bundalo, Branimir Dordevic, and Zlatko Bundalo "Multiple-Valued Regenerative CMOS Logic Circuits With High-Impedance Output State,"International Journal of Electronics Engineering .vol.19.1,April 2006,39-46.

[7]Turgay Temel and Avni Morgul, "Implementation of Multiple –Valued logic Gates Using Full Current-Mode CMOS Circuits,"in International Journal of Analog Integrated Circuits and Signal Processing,39,191-204,2004.

[8] K.C.Smith, "The Prospects for Multivalued Logic: A Technology and Applications View,"in IEEE Transactions on Computers, Vol.c.30, No.9,1981.