Implementation and study of quaternary multiplexer using universal set of gates

Govindam Sharma
Student(Mtech. VLSI)
Department of Electronics Engineering,
YMCA University, Faridabad, INDIA
ergovindamsharma@gmail.com

Nitin Sachdeva
Assistant Lecturer
Department of Electronics Engineering,
YMCA University, Faridabad, INDIA
n_julka@rediffmail.com

Shilpa Goyal
Student(Mtech. VLSI)
Department of Electronics Engineering,
YMCA University, Faridabad, INDIA
shilpagoyal30@gmail.com

Abstract: Multi-value logic is defined as a non-binary logic and involves the switching between more than two states. 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. More data can be transmitted by single wire having more than two levels. Multiple-valued logic (MVL) application in the design of digital devices opens large number of opportunities. It can reduce number of active elements and number of interconnection lines. In this paper we have studied universal set of quaternary gates which are further used for designing quaternary multiplexer. The circuit is designed in VHDL using ModelSim simulator. Structural modelling technique is used for designing the quaternary multiplexer.

Keywords — MVL (Multiple valued logic), eAND1, eAND2, eAND3, SUC, MAX, quaternary multiplexer etc

I. Introduction

Aristotle's logical calculus represents every proposition in two possible values (i.e., "true" and "false") whereas in multi valued logic two-valued logic may be extended to n-valued logic for n greater than 2. The subject of MVL is also known as multiple-valued, multi-valued or many-valued logic which is originated from the Lukasiewicz logic and Post algebra [1] [2].

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 G= [0, 1, 2] or balanced ternary with G = [-1, 0, 1]. 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. Ternary and quaternary circuits have been studied increasingly in recent years. Quaternary circuits have the practical 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 switching algebra.

The multiple valued logic is a viable alternative to cope with the interconnections M issues as they decrease the number of the interconnections as the inverse of the $\log_2 M$ [3]. This reduction in the area of the IC devoted to the interconnections has motivated many MVL contributions [4]–[8].

In this paper we have studied basic principles and algebra of MVL which is further used for designing of 4:1 quaternary Multiplexer. A 4:1 quaternary multiplexer is a device that selects one of four MVL input signals and forwards the selected input into a single line. Quaternary 4:1 MUX circuit is designed by the universal set of quaternary gates.

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 Section V presents the results of the implementations of the MVL digital circuits Finally, Section VI summarizes the concluding remarks.

II. MVL principles

The n-variables MVL function is a mapping f: $G^n \rightarrow G$. The MVL function represents values in the domain G= [0, 1, 2 … N-1] depending on the assigned values to the n inputs variables. 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), define a universal set under the proposed algebra for quaternary logic [9].

In the switching algebra, there are many methodological proposals for function minimization as: Karnaugh maps, Quine McCluskey etc. All of these can be extended to the MVL domain depending on the MVL algebra under consideration.

A\B	0	1	2	3
0	0	1	2	3
1	1	1	2	3
2	2	2	2	3
3	3	3	3	3

Table 1: MAX (A, B) operator

A	A^1	A^2	A^3
0	1	2	3
1	2	3	0
2	3	0	1
3	0	1	2

Table 2 : SUC (A) operator

A\B	0	1	2	3
0	0	0	0	0
1	0	1	0	0
2	0	0	0	0
3	0	0	0	0

*Table **3**: Extended and1 operator $A*^1B$*

A\B	0	1	2	3
0	0	0	0	0
1	0	0	2	0
2	0	0	0	0
3	0	0	0	0

*Table **4**: Extended AND2 operator $A*^2B$*

A\B	0	1	2	3
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	3

*Table **5**: Extended AND3 operator $A*^3B$*

III. MVL algebra

The closed Multiple Valued Logic algebra is an ordered set with domain G = [0, 1, 2, 3… N-1] in which acting two binary operators (Maximum and Extended AND, respectively) and Successor as defined below, with the lower element 0 and the upper element (N-1).
Maximum operator of denoted by the symbol A+B. By definition, If A ≥ B then A+B=A; otherwise A+B=B. A, BƐ G as shown in Table 1 in the quaternary logic G= [0, 1, 2, 3 … N - 1]. Max operator is introduced by Lukasiewicz [1].
Successor operator of denoted by the symbol A^1. By definition, A^1=B, A, B Ɛ G where B is the next element from the element A in the cyclic ordered set. It is an operator of only one argument, as shown in Table 2 in the quaternary logic G= [0, 1, 2, 3]. For notation purposes, the Suc (A) operator is denoted by the symbol A^1 and Suc (Suc (A)) is denoted by A^2. Note that the Successor operator can also be calculated as (a + n) MOD N, for a, n Ɛ G, 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 (A, B) denoted by the symbol $A *^{const} B$. By definition, If A=B=const then $A *^{const} B$ =const; otherwise $A *^{const} B$=0. Const, A, B Ɛ G as shown in table 3, table 4 and table 5 in the quaternary logic G = [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.
In the MVL algebra, for literals a_1, a_2, and a_3 Ɛ G and for constants i, p Ɛ G, the postulates in Table 4 hold, that are used for the multiplexer designs. There exist other properties that help the automatic minimization of the MVL digital circuits.

Identify for +: $a_1+0= a_1$; $0+ a_1= a_1$
Annihilator for +, $*^{i}$: $a_1+ (N-1)=(N-1)$; $a_{1*}{}^{i}0=0$;
Associativity of +, $*^{i}$: $a_1+ (a_2+ a_3)^{=} (a_1+ a_2)+ a_3$ $a_1{}^{*i} (a_2{}^{*i} a_3)^{=} (a_1{}^{*i} a_2)^{*i} a_3$
Commutativityof $+_{,}{}^{*i}$ $a_1+a_2=a_2+a_1$ $a_1*^{i} a_2=a_2*^{i}a_1$
Complement for $+_{,}{}^{*i}$ $a_1{}^{0}\vert a_1{}^{1}\vert a_1{}^{2}\vert\ldots.\ \vert a_1{}^{(N-1)}=(N-1)$; $a_1{}^{0*i}\ a_1{}^{1*i}\ a_1{}^{2*i}\ \ldots.\ {}^{*i}\ a_1{}^{(N-1)}=0$;
Reduction $(a_1{}^{p*i}a_2{}^{0}) + (a_1{}^{p*i} a_2{}^{1})+\ldots+a_1{}^{p*i}a_2{}^{(N-1)}=(a_1{}^{p*i} i)$;
Constant matching $i^{*i}i=i$;
Idempotency of + $a_1+a_1=a_1$
Distributivity of ${}^{*(N-1)}$ over +: $(a_1{}^{*(N-1)}a_2{}^{p})+(a_1{}^{*(N-1)}a_2{}^{q})=a_1{}^{*(N-1)}(a_2{}^{p}+a_2{}^{q})$.

Table 2 : MVL algebra postulates

IV. Synthesis of MVL circuits

The MVL circuit can be synthesised [12] as follows:

First, the MVL minterm is defined. For an MVL function of S literals, an Extended Product i minterm is comprised of S literals in an Extended Product operator, such that a literal appears once and only once in some form of the Successor operator. This MVL minterm is denoted as $m^{i}_{ka1,ka2,\ldots ka3}$, where m stands for the minterm, i identifies the Extended Product operator and the sub indexes K_{a1}, K_{a1} …. K_{as} are related to the Successor operator form for the literals $a_1,a_2,a_3\ldots a_s$ respectively. The sub indexes are computed as: k_{ai} = (i-succ) MOD $_N$, where suc is the number of times that the Successor operator is applied to the literal a_l Ɛ [a_1, a_2, a_3 … a_s], respectively, and N=4 (for MVL algebra of 4 levels). $a_1{}^{1}$ stands for the application of the Successor operator to variable a_1, then a_1=0, $a^{1}{}_{1}$=1, a2=1 for the $m^{1}{}_{01}=a_1{}^{1}*^{1}a_2$.
For two minterms $m^{i}_{ka1,ka2\ \ldots\ kas}$ and $m^{j}_{ka1,ka2\ \ldots\ kas}$, the $m^{i}_{ka1,ka2\ \ldots\ kas}$ is considered as the lower minterm and the upper minterm if and only if i < j . Otherwise, $m^{i}_{ka1,ka2\ \ldots\ kas}$ is considered as the upper and the minterm $m^{j}_{ka1,ka2\ \ldots\ kas}$ is considered as the lower minterm.

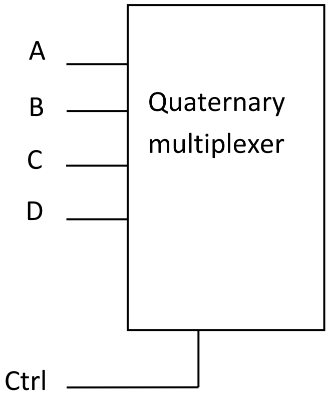

*Figure **1**: Block diagram of quaternary 4:1 multiplexer*

For minimization purposes lower minterms are considered as “don’t care” conditions. The proposed methodology for the synthesis of MVL circuits is based on functions $F_i(F_1,F_2,F_3 \ldots F_{(N-1)})$, represented in the canonical Sum of Extended Products (SOEP) form.

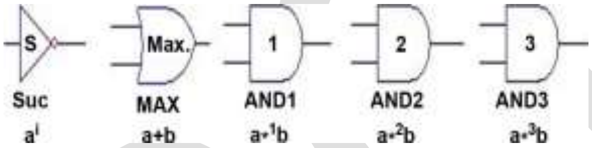

*Figure **2**: Gates representation.*

V. MVL Quaternary Multiplexer Synthesis

By designing an MVL multiplexer circuit adopting N=4 MVL algebra G=[0,1,2,3], function describes the output of the multiplexer in Fig. 1, as shown in Table 5. The MVL multiplexer has MVL inputs and one MVL output. Ctrl is the control input that controls the output. The

Multiplexer behaviour is described as follows:

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

Steps to synthesize an MVL function represented in the canonical Sum of Extended Products (SOEP):
STEP 1: extract each minterm of the function.
STEP 2: apply the Maximum operator to all min terms to form the SOEP representation for the functions
STEP 3: apply the Maximum operator to all functions

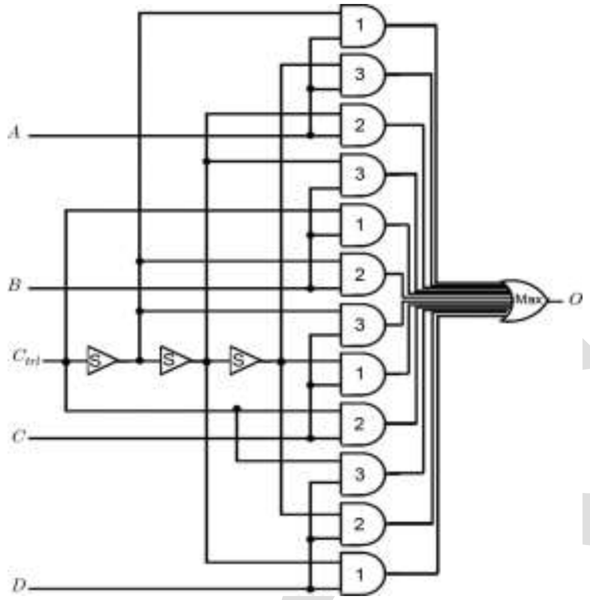

Figure 2 : 4:1 multiplexer circuit using Quaternary universal Gates

STEP 4: implement the MVL digital circuit

$$\begin{aligned} O = {} & Ctrl^{1}{}_{*}^{1} A + Ctrl^{2}{}_{*}^{2} A + Ctrl^{3}{}_{*}^{3} A + \\ & Ctrl_{*}^{1} B + Ctrl^{1}{}_{*}^{2} B + Ctrl^{2}{}_{*}^{3} B + \\ & Ctrl^{3}{}_{*}^{1} C + Ctrl_{*}^{2} C + Ctrl^{1}{}_{*}^{3} C + \\ & Ctrl^{2}{}_{*}^{1} D + Ctrl^{3}{}_{*}^{2} D + Ctrl_{*}^{3} D. \end{aligned}$$

VI. Results

Waveform of various quaternary MVL gates and multiplexer

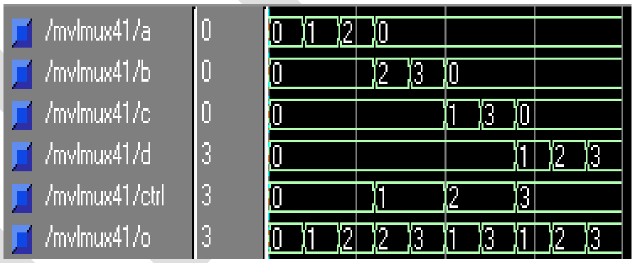

Figure 3: Multiplexer waveform

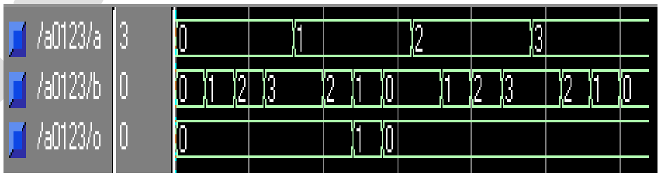

Figure 4: AND1 waveform

IJERGS

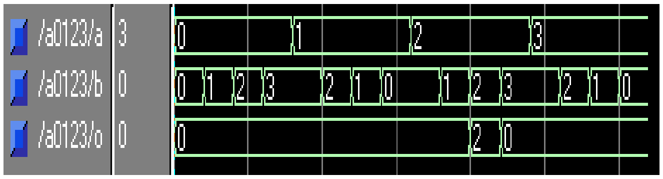

Figure 5 : AND2 waveform

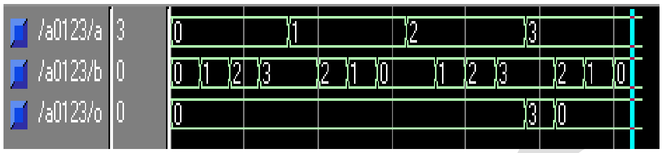

Figure 7: AND3 waveform

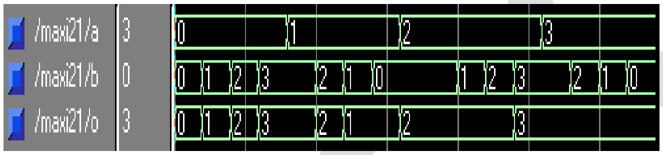

Figure 8: MAX waveform

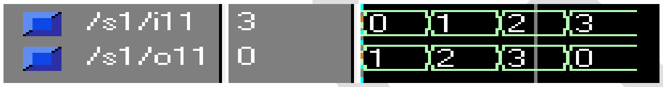

Figure 9: SUC waveform

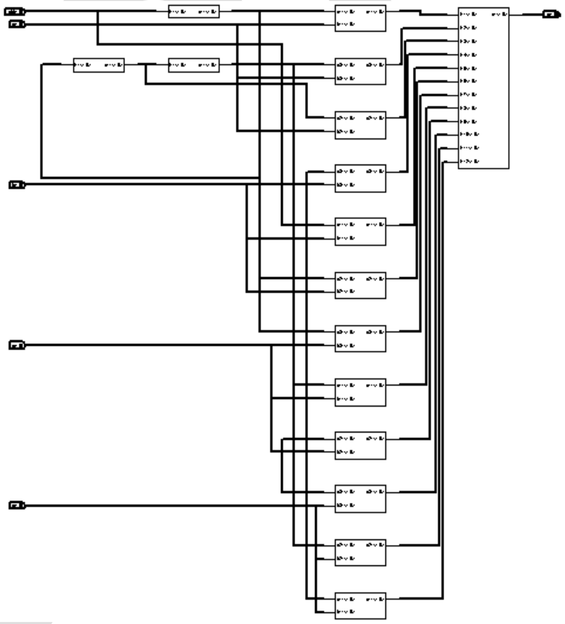

Figure 10: RTL schematic for multiplexer

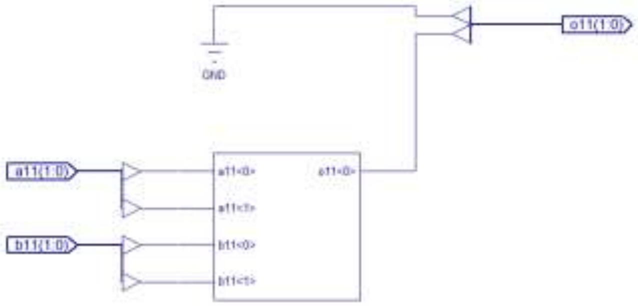

Figure 11: RTL schematic of EXTENDED AND(a11 and b11 as input and o11 as output)

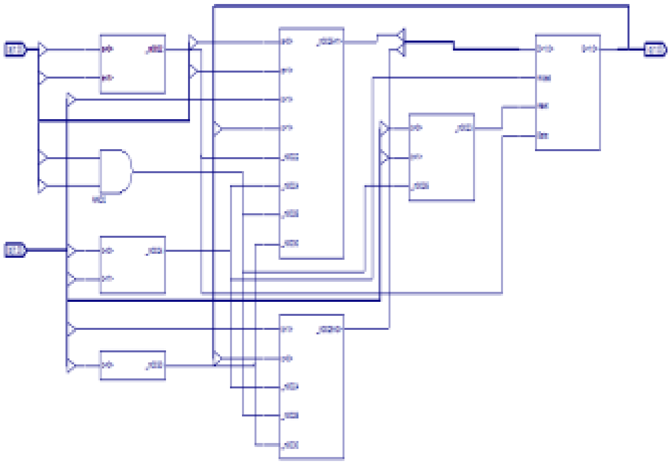

Figure 12: RTL schematic of MAX 2:1(a,b as input and o as output)

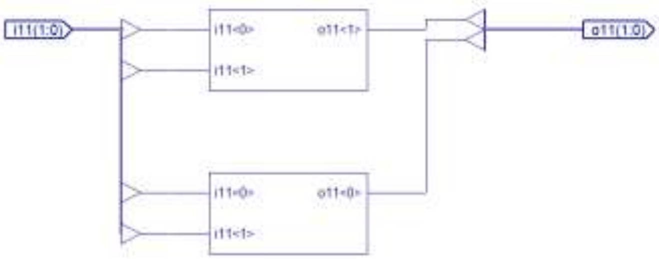

Figure 13: RTL schematic of SUC (i11 as input and o11 as output)

VII. Conclusion

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 levels with domain (0, 1, 2, 3) for the synthesis of the MVL multiplexer. VHDL Waveforms and RTL schematic for all MVL operators and quaternary 4:1 multiplexer are presented here.

REFERENCES:

[1] J. Lukasiewicz, “On three valued-logic,” in *L. Borkowski*, *SelectWorks*, Amsterdam, North-Holland, pp. 169–171, 1920.

[2] E. L. Post, “Introduction to a general theory of elementary propositions,” *Amer. J. Math.*, vol. 43, no. 3, Jul. 1921, pp. 163–185.

[3] P. Vasundara and K. Gurumurthy, “Quaternary CMOS combinational logic circuits,” in *Proc. Int. Conf. Inform. Multimedia Technol.* , pp. 538–542, 2009.

[4] C. R. D. A. Simovici, “Chain-based Ockham algebras,” in *Proc. Int. Symp. Multiple-Valued Logic*, vol. 19, pp. 456–462, 1989.

[5] Z. Dongmou, “Medium algebra MA and medium propositional calculus MP*,” in *Proc. Int. Symp. Multiple-Valued Logic*, vol. 19, pp. 289–294,1989.

[6] N. Takagi and K. Nakashima, “Discrete interval truth valued logic and its applications,” *IEEE Trans. Comput.*, vol. 49, no. 3, pp. 219–229,2000.

[7] A. K. Jain, R. J. Bolton, and M. H. Abd-El-Barr, “CMOS multiple valued logic design—Part I: function realization,” *IEEE Trans. Circuits Syst.*, vol. 40, no. 8, 1993.

[8] S. Kleene,May 2011 [Online]. Available: http://en.wikipedia.org/wiki/Kleene_algebra

[9] P. Kerntopf, M. A. Perkowski, and M. H. A. Khan, “On universality of general reversible multiple-valued logic gates,” in *Proc. 34th IEEE Int. Symp. Multiple-Valued Logic*, Toronto, ON, Canada, , pp. 262–267 May 2004.

[10] P. N. Victor, N. H. Troy, D. C. Bill, and J. I. David, *Digital Logic Circuit Analysis and Design*. Englewood Cliffs, NJ, USA: Prentice-Hall,2002.

[11] 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, pp. 262–267, 2009.

[12] 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, pp. 262–267, 2009

IJERGS