Design and Embodiment of larger Quaternary Multiplexer and Demultiplexer

Khandakar Mohammad Ishtiak [1], Safayat-Al-Imam[2], Nahyan Al Mahmud[3]
Department of Electrical and Electronic Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
ishtiak.eee@aust.edu[1], safayat_imam.eee@aust.edu[2] , nahyan.eee@aust.edu[3]

***Abstract*— In this work, the implementation of the novel quaternary algebra over the larger quaternary multiplexer and demuliplexer are performed. Both arbitrary multiplexer and demultiplexer are designed using the larger quaternary encoders and decoders respectively. This design validates the quaternary algebra to perform in building larger arbitrary logic blocks using different logic blocks with same principle. The design complexity has been compared with truth table and mathematical formulation under quaternary algebra and found perfect agreement with it.**

Keywords- Quaternary multiplexer, quaternarydemuliplexer, building blocks

I. INTRODUCTION

Multi-valued logic (MVL) overcomes the limitation of the binary logic by sending information in one digits [1]-[3].With the progress in noble devices physics, the implementation of MVL works well in different in electrical and optical devices [4]-[6]. Quaternary logic, was developed to design all kinds of logic circuits with lesser number of digits to store information than its binary component. In earlier works[7],[8]encoders and decoders are two common logic circuits used in communication and computing applications, is presented to the development of these logic circuits. In this work, we develop different mathematical representation of quaternary multiplexer and demultiplexer of arbitrary size using the building blocks of encoders and decoders. We also present a cascade design of large quaternary decoders and encoders. Finally, with the help of mathematical equations and truth table, we validate our designs with novel quaternary algebra.

II. REVIEW OF RELATED WORKS

A set of operators and a set of values {0, 1, 2, 3} for any valid proposition can be imagined as 2-bit binary equivalents 00*(absolute low*), 01(*intermediate low*), 10(*intermediate high*), 11(*absolute high*) in quaternary algebra [7]. A single quaternary digit is called a qudit when it is expressed as a number. The list of operators, their symbols are given in Fig. 1 and Table I.

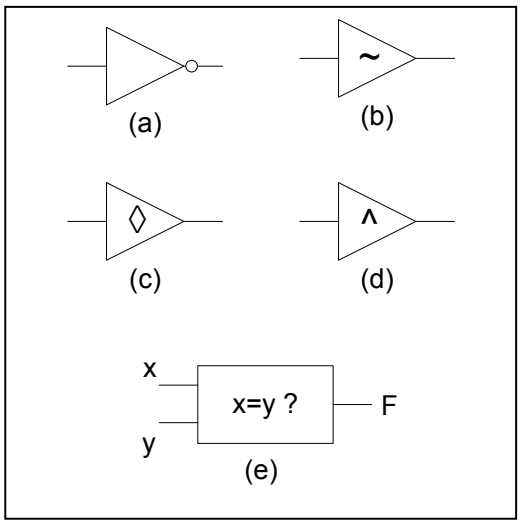

Figure 1. Circuit symbol of special quaternary operator (a) Basic inverter, (b) Bitswap inverter, (c) Outward inverter, (d) Inward inverter and (f) Equity Gate.

TABLE I. QUATERNARY SINGLE INPUT OPERATORS

A	*Basic NOT*	*Inward NOT*	*Outward NOT*	*Binary Bitswap*
0	3	2	3	0
1	2	2	3	2
2	1	1	0	1
3	0	1	0	3

Packed-binary representation of quaternary digits helps to explain the quaternary AND, OR, NOT, XOR and bitswap operator. The bitswap operator actually swaps the asymmetrical numbers living the symmetrical numbers same [8]. Hence the bitswap operator can be represented mathematically,

When a is symmetric then $\widetilde{a} = \overline{a}$

And when a is asymmetric then $\widetilde{a} = a$

In case of inward and outward inverters, the outputs may have only two states as shown below [7]

Condition	Inward inverter	Outward Inverter
for a<2;	$a' = \bar{a}.2$	$a' = \bar{a} + 3$
for a>1;	$a' = \bar{a} + 1$	$a' = \bar{a}.0$

In the case of equality operator,

$$E(A,B) = E(B,A) = A^B = B^A = \begin{Bmatrix} 0; A \neq B \\ 3; A = B \end{Bmatrix} \quad (1)$$

III. QUATERNARY DECODER

A quaternary decoder (fig 1a, Table I) for n-to- 4^n line, we have n number of input lines. In general case, for n-to-4^n decoder,

$$D_{(\sum_{j=0}^{j=n} 4^j * S_j^i)} = S_n^i \ldots\ldots S_1^i . S_0^i = \prod_{j=0}^{j=n} S_j^i ; \quad (2)$$

Here, i = 0, 1, 2, 3 and j = 0, 1, 2 …… n. and S_n^i is input line number and D is the output line number. For single input selection or 1-to-4 decoder,

$$D_{S^i} = S^i ;$$

where, i = 0, 1, 2, 3.

For 2 input selection or 2-to-16 decoder,

$$D_{(4*S_1^i + S_0^i)} = S_1^i . S_0^i ;$$

where,

i = 0, 1, 2, 3.

For this kind of structural construction, 1-to-4 decoders can be used as a building block [8]. We need 4^n AND gates for this.

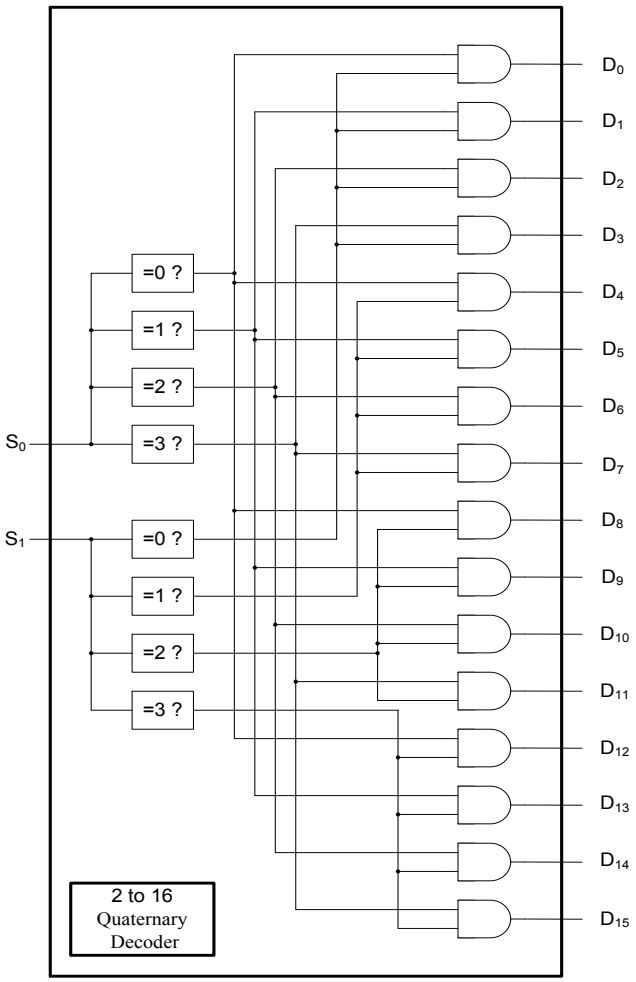

Figure 2. 2 to 16 Quaternary decoder.

TABLE II. TRUTH TABLE FOR 2 TO 16 QUATERNARY DECODER.

Selection		Decoder Output															
S1	*S0*	*D0*	*D1*	*D2*	*D3*	*D4*	*D5*	*D6*	*D7*	*D8*	*D9*	*D10*	*D11*	*D12*	*D13*	*D14*	*D15*
0	0	**3**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	**3**	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	2	0	0	**3**	0	0	0	0	0	0	0	0	0	0	0	0	0
0	3	0	0	0	**3**	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	**3**	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	**3**	0	0	0	0	0	0	0	0	0	0
1	2	0	0	0	0	0	0	**3**	0	0	0	0	0	0	0	0	0
1	3	0	0	0	0	0	0	0	**3**	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	**3**	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	**3**	0	0	0	0	0	0
2	2	0	0	0	0	0	0	0	0	0	0	**3**	0	0	0	0	0
2	3	0	0	0	0	0	0	0	0	0	0	0	**3**	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	**3**	0	0	0
3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	**3**	0	0
3	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**3**	0
3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**3**

IV. QUATERNARY ENCODER

For 4n input to n output position or 4n-to-n encoder,

$F_j=I_i . ((i/4^j) \bmod 4)$; where, $I_i = 3$, $j = 0, 1, 2, \ldots . \sqrt[4]{n+1}$ and i = 0, 1, 2, 3, ... n.

Here, (n+1) is a multiple of 4.

For four input to single output position or 4-to-1 encoder,

$F = i. I_i$; where, $I_i = 3$ and i = 0, 1, 2, 3.

For sixteen input to two output position or 16-to-2 encoder,

$F_j=I_i . (\left(\frac{i}{4^j}\right) \bmod 4)$; where, $I_i = 3$, j = 0 , 1. and i = 0, 1, 2, 3 . . . . 15.

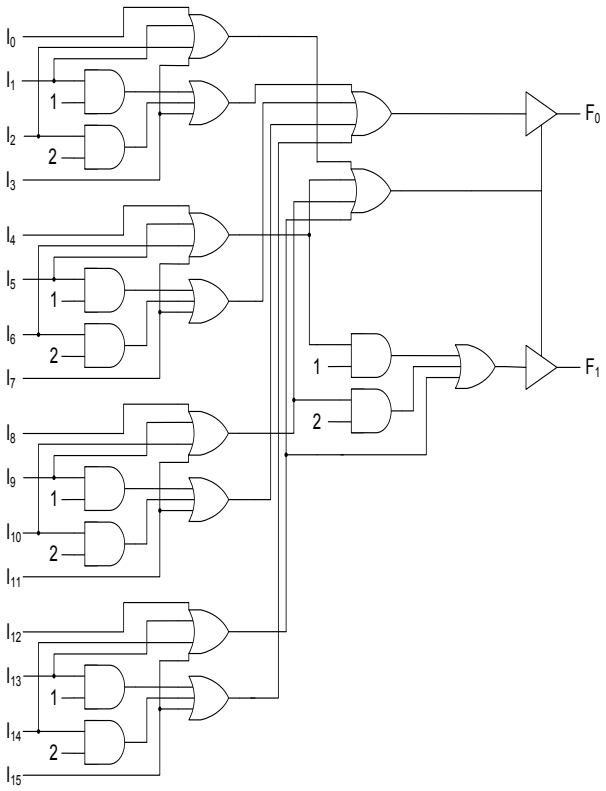

Figure 3. 16 to 2 Quaternary Encoder.

TABLE III. TRUTH TABLE FOR 16 TO 2 QUATERNARY ENCODER.

Encoder Input																Output	
I0	***I1***	***I2***	***I3***	***I4***	***I5***	***I6***	***I7***	***I8***	***I9***	***I10***	***I11***	***I12***	***I13***	***I14***	***I15***	***F1***	***F0***
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	**3**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	**3**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
0	0	0	**3**	0	0	0	0	0	0	0	0	0	0	0	0	0	3
0	0	0	0	**3**	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	**3**	0	0	0	0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	**3**	0	0	0	0	0	0	0	0	0	1	2
0	0	0	0	0	0	0	**3**	0	0	0	0	0	0	0	0	1	3
0	0	0	0	0	0	0	0	**3**	0	0	0	0	0	0	0	2	0
0	0	0	0	0	0	0	0	0	**3**	0	0	0	0	0	0	2	1
0	0	0	0	0	0	0	0	0	0	**3**	0	0	0	0	0	2	2
0	0	0	0	0	0	0	0	0	0	0	**3**	0	0	0	0	2	3
0	0	0	0	0	0	0	0	0	0	0	0	**3**	0	0	0	3	0
0	0	0	0	0	0	0	0	0	0	0	0	0	**3**	0	0	3	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	**3**	0	3	2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**3**	3	3

V. QUATERNARY MULTIPLEXER

A multiplexer can be made of decoders. For many data inputs all the outputs of the decoder is AND-ed. Hence, all the output data are OR-ed to get the output of the multiplexer. In general case. For 4n-to-1multiplexer,

$F= I_{(\Sigma_{j=0}^{j=n} 4^j * S_j^i)} . S_n^i \ldots\ldots S_1^i . S_0^i = I_{(\Sigma_{j=0}^{j=n} 4^j * S_j^i)} \prod_{j=0}^{j=n} S_j^i$;

where, i = 0, 1, 2, 3 and j = 0, 1, 2 …… n.

For single input selection or 4-to-1 multiplexer,

$F= I_{S^i} . S^i$; where, i = 0, 1, 2, 3.

For 2 input selection or 16-to-1 multiplexer,

$F= I_{(4*S_1^i+S_0^i)} . S_1^i . S_0^i$; where, i = 0, 1, 2, 3.

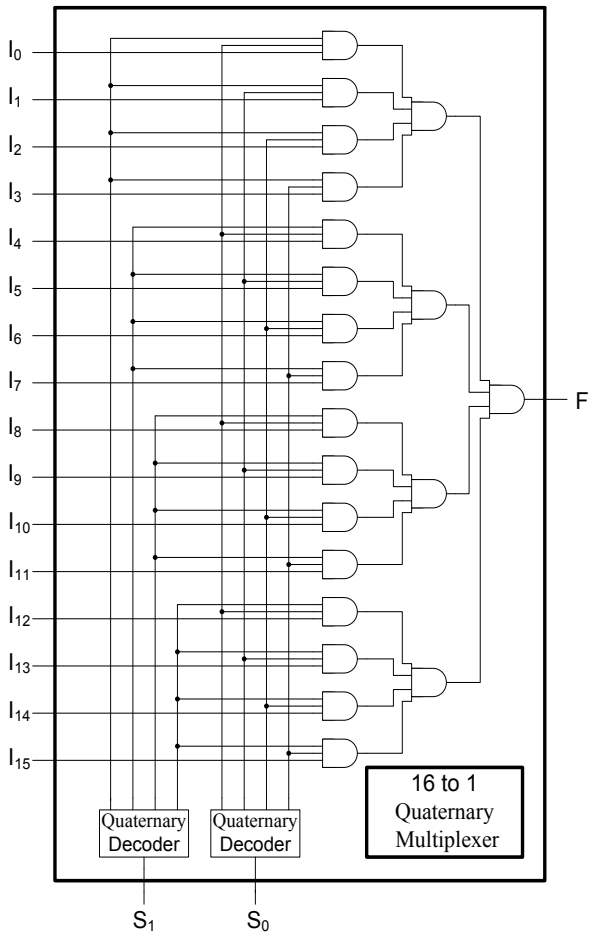

Figure 4. 16 to 1 Quarternary Multiplexer.

TABLE IV. TRUTH TABLE FOR 16 TO 1 QUARTERNARY MULTIPLEXER.

Selection		Output
S1	***S0***	***F***
0	0	I_0
0	1	I_1
0	2	I_2
0	3	I_3
1	0	I_4
1	1	I_5
1	2	I_6
1	3	I_7
2	0	I_8
2	1	I_9
2	2	I_{10}
2	3	I_{11}
3	0	I_{12}
3	1	I_{13}
3	2	I_{14}
3	3	I_{15}

VI. QUATERNARY DEMULTIPLEXER

Like decoder, Demultiplexer also works same way except outputs are AND-ed with data input D.

In general case. For 1-to-4n demultiplexer,

$D_{(\Sigma_{j=0}^{j=n} 4^j * S_j^i)} = D. S_n^i \ldots\ldots S_1^i . S_0^i = D \prod_{j=0}^{j=n} S_j^i$;

where, i = 0, 1, 2, 3 and j = 0, 1, 2 …… n.

For single input selection or 1-to-4 demultiplexer,

$D_{S^i} = . S^i$; where, i = 0, 1, 2, 3.

For 2 input selection or 1-to-16 demultiplexer,

$D_{(4*S_1^i+S_0^i)} = D. S_1^i S_0^i$; where, i = 0, 1, 2, 3.

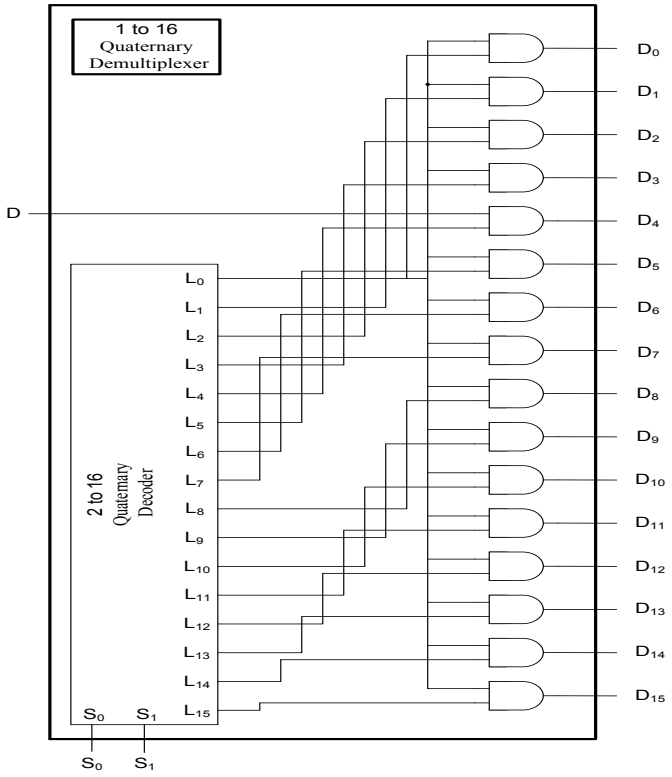

Figure 5. 16 to 1 Quarternary Multiplexer.

TABLE V. TRUTH TABLE FOR 16 TO 1 QUARTERNARYDEMULTIPLEXER.

Selection		Demultiplexer Output															
S1	***S0***	***D0***	***D1***	***D2***	***D3***	***D4***	***D5***	***D6***	***D7***	***D8***	***D9***	***D10***	***D11***	***D12***	***D13***	***D14***	***D15***
0	0	**D**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	**D**	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	2	0	0	**D**	0	0	0	0	0	0	0	0	0	0	0	0	0
0	3	0	0	0	**D**	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	**D**	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	**D**	0	0	0	0	0	0	0	0	0	0
1	2	0	0	0	0	0	0	**D**	0	0	0	0	0	0	0	0	0
1	3	0	0	0	0	0	0	0	**D**	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	**D**	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	**D**	0	0	0	0	0	0
2	2	0	0	0	0	0	0	0	0	0	0	**D**	0	0	0	0	0
2	3	0	0	0	0	0	0	0	0	0	0	0	**D**	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	**D**	0	0	0
3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	**D**	0	0
3	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**D**	0
3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**D**

VII. CONCLUSION:

In this work, we presented the overall design procedure for arbitrary larger quaternary multiplexer and demuliplexer along with encoders and decoders. Most importantly, the design follows an excellent agreement with the novel quaternary algebra. There have been different works on quaternary encoders and decoders but in this work the implementation of the encoders and decoders works well with larger larger quaternary multiplexer and demuliplexer. Hence the work shows the validation of the novel quaternary algebra in constructing building blocks like multiplexer and demuliplexer with lesser gate counts.

REFERENCES

[1] Hurst, S.L; "Multiple-Valued Logic; its Status and its Future",Computers, IEEE Transactions on, 33 (12), pp.1160-1179, Dec. 1984.

[2] Miller, D.M.; Thornton, M.A.; "Multiple Valued Logic: Concepts and Representations", Synthesis Lectures on Digital Circuits and Systems, Morgan & Claypool Publishers, 2007

[3] Smith, K.C.; "The Prospects for Multivalued Logic- A Technology and Applications View", Computers, IEEE Transactions on, 30, pp. 619-634, 1981

[4] Keshavarzian, A.P.; Navi, K.; "Optimum Quaternary Galois Field Circuit Design through Carbon Nano Tube Technology", Proceedings, International Conference on Advanced Computing and Communications (ADCOM 2007), pp. 214-219, Dec. 2007

[5] Chattopadhyay, T.; Taraphdar, C.; Roy, J.N.; "Quaternary Galois field adder based all-optical multivalued logic circuits", *Applied Optics*, **48** (22), pp. E35-E44, (2009)

[6] Jahangir, I.; Hasan, D.M.N.; Siddique, N.A.; Islam, S.; Hasan, M.M.; "Design of quaternary sequential circuits using a newly proposed quaternary algebra*", Proceedings, 12th International Conference on Computers and Information Technology (ICCIT 2009),* pp. 203, Dec. 2009

[7] Jahangir, I,Das A; and Hasan M;. "Formulation and Development of a Novel Quaternary Algebra.*" arXiv preprint arXiv:*1108.5497 (2011).

[8] Jahangir, I,Das A; and Hasan M; "Design of novel quaternary encoders and decoders." *Informatics, Electronics & Vision (ICIEV), 2012 International Conference on*. IEEE, 2012.