Journal of Engineering and Development, Vol. 16, No.1, March 2012 ISSN 1813- 7822

A New Theory for Multiple Valued Logic Using Convert-Coded-Collect (CCCi)Space

Assist. Prof. Dr. Dhafer R. Zaghar
Department of Computer & Software Engineering
College of Engineering/University of Al-Mustansiriyah /Iraq,
Email:drz_raw@yahoo.com

Abstract

The Multiple Valued Logic (MVL) is one of the keys to building processors in the futurebecause the use of the MVL in control and uP will reduce the number of instruction that necessary to solve problems and it increases the parallelism. The MVL will increase the speed of the systems and reduce the required memory size and reduce the connections.
This paper proposed a new theory to extend the binary logic as operations in new space called Convert-Coded-Collect space (CCCi). The CCCi space is a closed space has i integer values, it used to convert the input to the output in three phases called convert phase, coded phase and collect phase respectively.
The CCCispace carries out with any integer number of MVLs that depend on the value of i. This paper will discuss two cases of the CCCi space, first two values (i=2) that called CCC2; it will prove the CCC2 is more efficient from than the Boolean algebra. The other case for this space is CCC4 that has 4 values MVL. This theory is a useful MVL so it has simple functions witha package of advantages.
This paper will discuss an example to design a logic multiplier under Boolean logic, under CCC2 space and under CCC4space to show the advantages of the new theory.

الخلاصة

المنطق المتعددة المستويات (MVL) هو احد المفاتيح الاساسية لبناء المعالجات في المستقبل لان استخدام المنطق المتعددة المستويات في السيطرة والمعالج الدقيق (uP) سوف يقلل من عدد التعليمات اللازمة لحل المشاكل وانها سوف تزيد من عملية التوازي. المنطق المتعددة المستويات سوف يزيد سرعة النظم وتقليل حجم الذاكرة المطلوبة وتقليل الاتصالات.
اقترحت هذه الورقة نظرية جديدة لتوسيع المنطق الثنائي كعمليات في فضاء جديد يسمىفضاء تحويل -ترميز -تجميع (Convert-Coded-Collect space (CCCi)). الفضاء هو فضاء مغلق يملك i من القيم الصحيحة تستخدم لتحويل المدخلات الى المخرجات في ثلاث مراحل تدعى مرحلة تحويل و مرحلة ترميز و مرحلة تجميع على التوالي.
فضاء تحويل -ترميز -تجميع يعمل مع أي عدد صحيح من MVLs/عتمادا على قيمة i. هذا البحث سيناقش حالتين من حالات فضاء تحويل - ترميز - تجميع الاول للقيمه اثنين (i=2) والذي سيدعىCCC2 وهي سوف تثبت انه أكثر كفاءة

من الجبر البوليني (Boolean algebra). حالة أخرى لهذا الفضاء هوCCC4 والذي يملك ٤ قيم للمنطق المتعددة المستويات (4 MVL). هذه النظرية للمنطق المتعددة المستويات هي مفيدة بحيث تعطي منطق متعددة المستوياتيملك دوال بسيطة معحزمة من الميزات. ان هذا البحث سوف يستعرض مثال للتصميم لدائرة ضرب منطقي بواسطة المنطق البلوني و فضاء CCC2 و فضاء CCC4 لبيان ميزات هذه النظرية.

Keywords: Multiple valued logic, multiple valued logic, convert-code-collect space, Boolean algebra, PLA, logic gates, multiplier.

1- Introduction

Multiple Valued Logic(MVL)or Many-Valued Logic (in some references)is where the number of discrete logic levels is not confined to two. This is unlike binary, which only has two levels, logic level 0 and logic level 1, ie {0, 1}. Ternary logic has three logic levels {0, 1, 2} while quaternary has four logic levels {0, 1, 2, 3}.The number of logic levels is equal to the radix of the number system employed [1].

Even before the era of semiconductor technology, switching relays were binary in nature. They were energized, logic 1, or de-energized, logic 0. In the electronics and computing industry the status of two-valued, or binary, logic has reached its present level of complexity, sophistication, and application largely because of the continuous development of microelectronics and their ability to provide efficient two state devices and circuits. Binary logic has been very successful down through the years. There are mathematical tools available which have helped in its development and these were originally developed by the mathematician George Boole who generated Boolean algebra which aided in the success of binary logic [2].

The main reason for research beyond the present day binary logic systems is interconnection problems, both on-chip and between chips. With increase in capability per chip, the difficulties of placement and routing, on chip, of the digital logic elements are escalating. It is often the case that the silicon area used for interconnections may be greater than that used for the active logic elements. Also the difficulties of bringing an increasing number of connections off-chip is promoting a new consideration of packaging concepts in an attempt to overcome problems which are becoming mechanically, thermally, and electrically extreme. All these factors point to the use of a logic system higher than that of binary circuits so that the information content per interconnection and per line can be increased [2].

Within the electronics industry, binary logic is used in almost all applications but multiple valued logic has the potential for enriching the current two valued reality. Because of the necessary co-existence with binary logic, radix conversion is a topic of immediate concern. But multiple-valued to binary and binary to multiple-valued converters would just be another type of converter as used in analog-to-digital and digital-to-analog converters [3].

In binary systems the number and naming of one-variable functions isn't as complex as it is in multiple valued logic systems. An r-valued system has ***r*** possible outputs for r possible input values, and accordingly r^r outputs of a single r-valued variable [3].
So if we calculated each of the possible output functions; With a radix of r =2 for binary, ie 2 possible input states 1 & 0, the number of functions is $2^2 = 4$.But with a radix of r = 4 for quaternary, ie 4 possible input states 0, 1, 2 & 3, the number of functions is $4^4 = 256$ [3].This in effect proves that in any numerical system, the smaller the radix the larger the number of digits necessary to express a given quantity [2].
Nevertheless, multiple valued circuits have many advantages for digital filtering circuits. One such advantage is that the digital filtering application is a prime example of the use of logic continuously at a fixed speed. This, along with the reduction in gate count brought about by switching from binary to MVL, combines to make it possible that a high gate count, binary-CMOS implementation operating continuously at high speed, can consume a lot more power than the multiple valued equivalent given the devices [4].
One obvious feature that multiple valued data representation has is its potential for reducing the number of lines required for the parallel transmission of large amounts of data. The intense need for compaction in memory arrays has led to several commercial memory developments using multi-valued data coding by companies such as Motorola and Intel, who have used four-valued read only memories (ROMs) in some of their commercial products [2].
Within the communications sector, specialists in long distance transmission have recognized the potential for information compaction and bandwidth reduction using multiple valued coding, a form of MVL. In communications however, problems have arisen in the transmission of signals and when the signals are received they are often very difficult to detect. But the potential is still there for development of multiple valued techniques in the areas of pre and post processing of communications signals [2].
This paper extend the binary logic to MVL, it proposed a new theory that will demonstratea new definition for the MVL and its functions; it has seven sections that will discuss the theory and its results as follows:section 1 is an introduction to MVL and itsapplications; section 2 is a briefedhistory of MVL. Section 3 clear some previous works for more reading.
The second part will discuss the new theory in MVL, in section 4 we give a hint to the theory by discussion the PLA design approach (NOT-AND-OR). Section 5 is the main section in this paper that states the new theory from the Convert-Coded-Collect (CCCi) space with some proofs for this theoryby the discussions of the CCC2 and the CCC4 spaces. An example (logical multiplier circuit) will clear the advantages for this theory is presented in section 6. The properties of the CCCi MVL discuss in details in section 7.

2- History of MVL[5]

Many-valued logic as a separate subject was created by the Polish logician and philosopher Łukasiewicz (1920), and developed first in Poland. His first intention was to use a third, additional truth value for "possible", and to model in this way the modalities "it is necessary that" and "it is possible that". This intended application to modal logic did not materialize. The outcome of these investigations is, however, the Łukasiewicz systems, and a series of theoretical results concerning these systems. Essentially parallel to the Łukasiewicz approach, the American mathematician Post (1921) introduced the basic idea of additional truth degrees, and applied it to problems of the representability of functions. Later on, Gödel (1932) tried to understand intuitionistic logic in terms of many truth degrees. The outcome was the family of Gödel systems, and a result, namely, that intuitionistic logic does not have a characteristic logical matrix with only finitely many truth degrees. A few years later, Jaskowski (1936) constructed an infinite valued characteristic matrix for intuitionistic logic. It seems, however, that the truth degrees of this matrix do not have a nice and simple intuitive interpretation.

A philosophical application of 3-valued logic to the discussion of paradoxes was proposed by the Russian logician Bochvar (1938), and a mathematical one to partial function and relations by the American logician Kleene (1938). Much later Kleene's connectives also became philosophically interesting as a technical tool to determine fixed points in the revision theory of truth initiated by Kripke (1975). The 1950s saw (i) an analytical characterization of the class of truth degree functions definable in the infinite valued propositional Łukasiewicz system by McNaughton (1951), (ii) a completeness proof for the same system by Chang (1958, 1959) introducing the notion of MV-algebra and a more traditional one by Rose/Rosser (1958), as well as (iii) a completeness proof for the infinite valued propositional Gödel system by Dummett (1959). The 1950s also saw an approach of Skolem (1957) toward proving the consistency of set theory in the realm of infinite valued logic. In the 1960s, Scarpellini (1962) made clear that the first-order infinite valued Łukasiewicz system is not (recursively) axiomatizable. Hay (1963) as well as Belluce/Chang (1963) proved that the addition of one infinitary inference rule leads to an axiomatization of L_∞. And Horn (1969) presented a completeness proof for first-order infinite valued Gödel logic. Besides these developments inside pure many-valued logic, Zadeh (1965) started an (application oriented) approach toward the formalization of vague notions by generalized set theoretic means, which soon was related by Goguen (1968/69) to philosophical applications, and which later on inspired also a lot of theoretical considerations inside MVL. The 1970s mark a period of restricted activity in pure many-valued logic. There was, however, a lot of work in the closely related area of (computer science) applications of vague notions formalized as fuzzy sets, initiated e.g. by Zadeh (1975, 1979). And there was an important extension of MVL by a graded notion of inference and entailment in Pavelka (1979). In the 1980s, fuzzy sets and their applications remained a hot topic that called for theoretical foundations by methods of many-valued logic. In addition, there were the first complexity results e.g. concerning the set of logically valid formulas in first-order infinite valued Łukasiewicz logic, by Ragaz (1983). Mundici (1986) started a deeper study of MV-algebras. These trends have continued since the 1980s. Research has included applications of MVL to fuzzy set theory and their applications, detailed investigations of algebraic structures related to systems of MVL, the study of graded notions of entailment, and investigations into complexity issues for different problems in systems of MVL. This research was complemented by interesting work on proof theory, on automated theorem proving, by different applications in artificial intelligence matters, and by a detailed study of infinite valued systems based on t-norms.

3- Some Previous Works

This section is a short review for some examples of the modern works in the different MVL fields that listed in the previous section. Most works in MVL fill in the fields of the “fuzzy logic” such as Ref. [6] and “Łukasiewicz logic and its applications” such as Ref. [7 and 8],Colin Keng-Yan Tan in Ref. [9] discusses N-valued Lukasiewiczlogic, fuzzy logic and belief augmented frames MVL.

The n-value MVL is the field of this paper, in this field we have a different and distinct ideas, such exampleRef.[10] focuses on the fixed polarity Reed- Muller (FPRM) expression of MVL symmetric functions, Ref.[11] discuss the Multiple-valued decision diagrams (MDDs) that give a way of approaching problems by using symbolic variables which are often more naturally associated with the problem statement than the variables obtained by a binary encoding.Robert K Brayton and Sunil P Khatri in Ref.[12] try to expand Multiple-valued decision diagrams using mod-p decision diagrams, it is a good work but the final results are not suitable to implement free MVL systems.Christian Lang and Bernd Steinbachin Ref. [13] proposed a nice work that presents algorithms that allow the realization of multi-valued functions as a multi-level network consisting of min- and max-gates. Hence Ref.[14] and Ref. [15] are propose a set of reversible logic functions, the results proposed a large number of gates, so all gate required to LUT for function definition.Ref.[16] proposesspecial mathematics logic to implement ternary and penta logic registers in the MVL, that is a complex functions with very limited flexibility.Galois Field logic discuss inRef.[17] use GF(4) for 4 values MVL and Ref.[18] for ternary MVL that are results complex functions with limited flexibility.

Some methods and problems in the implementation of MVL discuss in Ref.[19] this dissertation summarizes the study conducted to research the MVL circuits and feasibility of design and validation of quaternary arithmetic circuits using SUSLOC technology and also quaternary dual recoding squaring circuits using CMOS gates. The research indicates that the quaternary circuits do offer the benefit of lower power consumption compared to the traditional two-valued (binary) logic circuits. While Ref.[20] discuss the development of INGAAS-based multiple-junction surface tunnel transistors for multiple-valued logic circuits. Ref.[21] discusses the implementation of multi-valued logic gates using full current-mode CMOS circuits. Mojtaba Jamalizadehet al in Ref. [22] use the minimum, maximum, complement, truncated difference and some other limited operations to implement the MVL the final results is lowflexibility functions with limited applications. Fergal Tuffyin Ref.[1] show a good report discusses the based upon the research and implementation of multiple valued memory circuits using resonant tunnelling devices (RTDS) and MVL. Ref.[23] solves a special problem in high speed multiple valued logic full adder using carbon NANO tube field effect transistor. Finally Ref.[24] andRef. [25] have good examples in the modern applications of MVL.

4- PLA Design Approach(NOT-AND-OR)

The first PLDs were programmable logic arrays (PLAs). A PLA is a combinational, three-levelNOT-AND-OR device that can be programmed to realize any sum-of-products logic expression, subject to the size limitations of the device.
An *n x m* PLA with *p* product terms can contains *p*(2n-input) AND gates and *m*(p-input) OR gates. Fig (1) below shows a small PLA with four inputs, six AND gates and three OR gates and outputs. Each input is connected to a buffer that produces both a true and complemented version of the signal for use within the array. Potential connections within the array are indicated by X's; the device is programmed by establishing only the connections that are actually needed. Thus, each AND gate can be connected to any subset of the primary input signals and their complements. Similarly each OR gate can be connected to any subset of the AND-gate outputs [26].

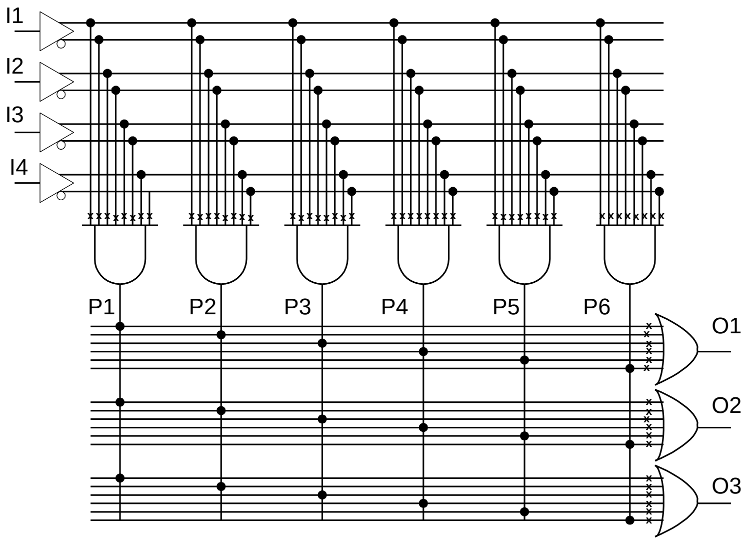


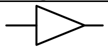
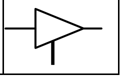
Fig (1): A 4 x 3 PLA with six product terms [26].

The NOT-AND-OR approach in PLA is able to implement any logical function. This approach is the staring point to expand binary logic to general MVL. The NOT-AND-OR approach is a special case the general case proposed a new approach has three phases each phase is expanded to the NOT-AND-OR approach. The first phase is *Convert* phase that has two jobs first is convert any input value to other possible values (NOT gate that convert 0-to-1 and 1-to-0), while the second job is the rejection any input (tri-state buffer as hidden in Fig (1)). The second phase is *Coded* phase that able to coded any two inputs data to give a single output this phase use AND (we can use NAND, NOR, XOR, XNOR and any other types of Boolean gates) to detect any input case (1 and 1 as example) as value 1 and other input cases to 0s as in equation (1).

$$F_{AND}(A,B)=\begin{cases}1 & A=1\ and\ B=1\\ 0 & ow\end{cases} \qquad (1)$$

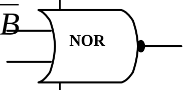
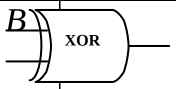
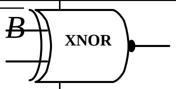
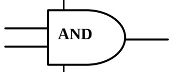
The third phase is the *Collect* phase that use to collect each outputs of the second phase, this phase use generally the maximum relation (OR gate in Boolean). The three phases have different relations in closed set of values (0, 1). The combination for the three phases will called *Convert-Coded-Collect* (CCC). Fig (2) shows the functions and symbols of the Boolean gates for (a) Convert phase, (b) Coded phase and (c) Collect phase.

Input	**NOT**
A	$\overline{A}$
0	1
1	0

Symbol	**function**	**equation**	**figure**
NOT	inverter	$\overline{A}$	
REG	Tri-state	Z	

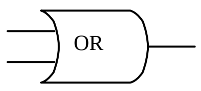
(a)

Inputs		**NOR**	**XOR**	**XNOR**	**AND**
A	B	$\overline{A+B}$	$A\oplus B$	$\overline{A\oplus B}$	AB
0	0	1	0	1	0
0	1	0	1	0	0
1	0	0	1	0	0
1	1	0	0	1	1

Symbol	**function**	**equation**	**figure**
NOR	NOT-OR	$\overline{A+B}$	
XOR	Exclusive -OR	$A\oplus B$	
XNOR	Exclusive -NOR	$\overline{A\oplus B}$	
AND	And	AB	

(b)

Inputs		**OR**
A	B	$A+B$
0	0	0
0	1	1
1	0	1
1	1	1

Symbol	**function**	**equation**	**Figure**
OR	Max (A, B)	$A+B$	

(c)

Fig(2): The functions and symbols of the Boolean gates for (a) Convert phase, (b) Coded phase and (c) Collect phase.

5- The Convert-Coded-Collect (CCCi)Space

This section try to clear the mean of the CCCi space and its operations, it has three sub-sections first give the theory and a general definition for the CCCi space while the other two sub-sections are a proofs for this theory using two examples for CCCi space. The first example is CCC2 that is equivalent to Boolean algebra to clear some ambiguous in the first sub-section, while the second example is proof how the CCC4 able to implement a MVL with 4 levels.

5.1- Theory and Definition

Theory: The *Convert-Coded-Collect*(CCCi)space is any closed space has i integer values setSi ={0,…, i-1}. This space has three phases Convert, Coded and Collect; itis able to containmentlogical functionsthat use to convertany input in the closed set Si to any output in the same set Si underthe*logical* conditions.The Convert, Coded and Collectphases can be definedas follows:

1- Convert phase

This phase has one input and one output (1-to-1), this phase able to convert any input $A \in Si$under the *logical* conditionsto any other values in Si and convert any maximum valueto the minimum valuein Si. This phase required at least to $2\log_2^i$ functions.

2- Coded phase

This phase has two inputs and multi-outputs (2-to-n), in this phase *each* acceptable combination for the two inputs $A, B \in Si$must be coded under the *logical* conditionsto a *unique* value in Si.Hence, an acceptable set to satisfy the conditions for this phase is Ei and Fi gates that define as in equations (2) and (3), this phase required at least to $2i$ functions.

$$Ei(A,B) = \begin{cases} \max & A = B = i \\ \min & ow \end{cases} \qquad (2)$$

$$Fi(A,B) = \begin{cases} B & A = i \\ \min & ow \end{cases} \qquad (3)$$

3- Collect phase

This phase has multiple inputs and one output (n-to-1), in this phase then inputs $A, B, C, D \in Si$will generate under the *logical* conditionsa single value in Si.Hence, an acceptable function to satisfy the conditions for this phase is MAX (maximum) that define as in equation (4) also we can use MIN (minimum) or DIF (maximum -minimum). Another important gate supplemented with this phase is X gate that is an exchange switch and it is defined as in equation (5).This phase required at least to 2 functions, the CCCi space has at least to $2(i+\log_2^i+1)$ functions.

$$MAX(A,B) = \text{maximum } (A,B) \quad (4)$$
$$XFi(A,B) = Fi(B,A) \quad (5)$$

5.2- The CCC2 Space

The definition of CCCi space in the previous section has some ambiguous, the application of CCCi with two values S2 = {0, 1}(binary logic) will called**CCC2.**The comparison of the CCC2 with the classical logic will clear the ambiguous. The three phases of CCC2 discuss as follows:

1- Convert phase

This phase has one input and one output (1-to-1), and it can be defend simply as shown in Fig (3a) that has same result of Fig (2a).

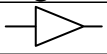
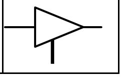
2- Coded phase

This phase has two inputs and multi-outputs (2-to-n), and it can be defend simply as shown in Fig (3b), hence compare with the result of Fig (2b).

3- Collect phase

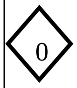
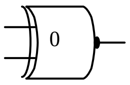
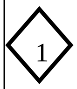
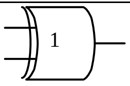
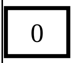
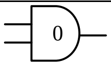
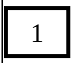
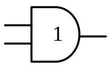
This phase has multi-inputs and one output (n-to-1), and it can be defend simply as shown in Fig (3c). This phase is same inFig (2c) with additional supplemented gate (X gate).

Input	**NOT**
A	$\overline{A}$
0	1
1	0

Symbol	**function**	**equation**	**figure**
NOT	inverter	$\overline{A}$	
REG	Tri-state	Z	

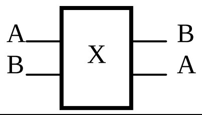
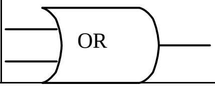
(a)

Inputs		**E0**	**E1**	**F0**	**XF0**	**F1**
A	B	NOR	AND	XOR		AND
0	0	1	0	0	0	0
0	1	0	0	1	0	0
1	0	0	0	0	1	0
1	1	0	1	0	0	1

Sym-bol	**Fun-ction**	**Equiv-alent**	**Equat-ion**	**figure**
E0		NOR	Equ (2)	
E1		AND	Equ (2)	
F0		XOR	Equ (3)	
F1		AND	Equ (3)	

(b)

Inputs		**OR**
A	B	$A+B$
0	0	0
0	1	1
1	0	1
1	1	1

Symbol	**function**	**equation**	**Figure**
X	Equ 3 Exchange	$\underline{X}$	
OR	Equ 4 maximum	$A+B$	

(c)

Fig(3): The functions and symbols of the CCC2 space for (a) Convert phase, (b) Coded phase and (c) Collect phase.

The simple comparison between Fig (3) and Fig (2) shows the CCC2 able to generate all the relations of classical logic (NOT, AND, OR, XOR, …). The convert and collect phases have same functions of the classical logic, the coded phase has distinct functions in the definitions but they have same outputs except XOR. The CCC2 functions are not commutative but they more flexibility from the Boolean functions. The functions E1 and F1 have same truth table as special case therefore F1 will be neglected. Some case in the coded phase of CCCi hasn't output code but these cases solved with the XFi gate combination as in case of F0 and XF0 in Fig (3b).

5.3- The CCC4 Space

The definition of CCCi space in the previous section requires to enhancement.The application of CCCi with the values S4 = {0, 1, 2, 3} (quaternary logic) will called**CCC4**it will enhancementthesolutions and the applications of the CCCi space. The three phases of CCC4 discuss as follows:

1- Convert phase

This phase has one input and one output (1-to-1), so it can be defend simply as shown in Fig (4a&4b). This phase has 4 main gates LN, UN, LR and UR these gates able to presentall the requirements of the convert phase in CCC4, the additional gate AN is auxiliary function because it replace by LN and UN serially, it (AN) will be neglected to reduce the number of gates.

2- Coded phase

This phase has two inputs and multi-outputs (2-to-n), and it can be defend simply as shown in Fig (4c&4d).The solutions of the equations (2) and (3) in CCC4 (max=3 and min=0) under S4 set gives 8 gates. The design in the next section will clear the meaning and using of these gates.

3- Collect phase

This phase has multi-inputs and one output (n-to-1), and it can be defending simply as shown in Fig (4e&4f). The MAX and X gatesare the same in equations (4) and (5) with max=3.

Input	**LN**	**UN**	**AN**	**LR**	**UR**
0	1	2	3	0	0
1	0	3	2	0	1
2	3	0	1	2	0
3	2	1	0	2	1

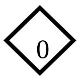
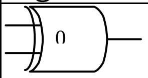
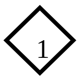
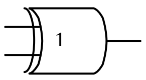
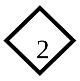
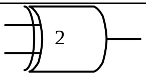
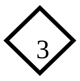
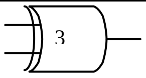
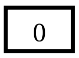
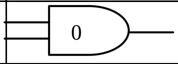
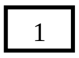
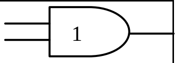
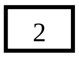
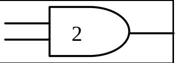
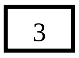
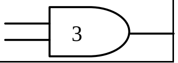
(a)

Symbol	**function**	**equation**	**figure**
LN	**Low inverter**	$\overset{\lrcorner}{A}$	▽
UN	**Up inverter**	$\overset{\llcorner}{A}$	△
LR	**Low reject**	$\overset{\urcorner}{A}$	▼
UR	**Up reject**	$\overset{\ulcorner}{A}$	▲

(b)

Inputs		**Outputs**							
A	**B**	**E0**	**E1**	**E2**	**E3**	**F0**	**F1**	**F2**	**F3**
0	**0**	3	0	0	0	0	0	0	0
0	**1**	0	0	0	0	1	0	0	0
0	**2**	0	0	0	0	2	0	0	0
0	**3**	0	0	0	0	3	0	0	0
1	**0**	0	0	0	0	0	0	0	0
1	**1**	0	3	0	0	0	1	0	0
1	**2**	0	0	0	0	0	2	0	0
1	**3**	0	0	0	0	0	3	0	0
2	**0**	0	0	0	0	0	0	0	0
2	**1**	0	0	0	0	0	0	1	0
2	**2**	0	0	3	0	0	0	2	0
2	**3**	0	0	0	0	0	0	3	0
3	**0**	0	0	0	0	0	0	0	0
3	**1**	0	0	0	0	0	0	0	1
3	**2**	0	0	0	0	0	0	0	2
3	**3**	0	0	0	3	0	0	0	3

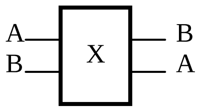
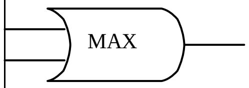
(c)

Symbol	**function**	**equation**	**figure**
E0	Equ 1		
E1	Equ 1		
E2	Equ 1		
E3	Equ 1		
F0	Equ 2		
F1	Equ 2		
F2	Equ 2		
F3	Equ 2		

(d)

Inputs		**Output**
A	B	$A+B$
0	0	0
0	1	1
0	2	2
0	3	3
1	0	1
1	1	1
1	2	2
1	3	3
2	0	2
2	1	2
2	2	2
2	3	3
3	0	3
3	1	3
3	2	3
3	3	3

(e)

Symbol	**function**	**equation**	**Figure**
X	Equ 3 Exchange	X	
MAX	Equ 4 maximum	$A+B$	

(f)

Fig(4): The functions and symbols of the CCC4 space for (a) truth table for convert phase, (b) symbols table for convert phase,(c) truth table for coded phase, (d) symbols table for coded phase, (e) truth table for collect phase, (f) symbols table for collect phase.

6- Design Example: Multiplier in the CCCi Space

This section is an enhancement to the proof of the theory in the pervious section. This section selects a logic circuit (multiplier) as an important circuit in the real word. It has two subsections first shows the implementation of 2*2bits multiplier in Boolean and in CCC2 space. These 2*2 bits in CCC2 equivalent to 1*1 bit in CCC4 this multiplier will design in the second example. Hence, we can use this approach to design themultiplier for any value of i in CCCi space.

6.1- 2x2 binary multiplications in CCC2

The truth table of 2*2 bits in CCC2 space that equivalent to the truth table in Boolean algebra is shown in Fig (5). The results of the simplification of the truth table are shown in equation (6) for the classic Boolean and in equation (7) for the CCC2 space. The methods design that convert the truth table to logical equations are not important also they have a large details fill out of the range of this paper.

Inputs		**Outputs**			
A1A0	B1B0	O0	O1	O2	O3
00	00	0	0	0	0
00	01	0	0	0	0
00	10	0	0	0	0
00	11	0	0	0	0
01	00	0	0	0	0
01	01	1	0	0	0
01	10	0	1	0	0
01	11	1	1	0	0
10	00	0	0	0	0
10	01	0	1	0	0
10	10	0	0	1	0
10	11	0	1	1	0
11	00	0	0	0	0
11	01	1	1	0	0
11	10	0	1	1	0
11	11	1	0	0	1

Fig(5): The truth table of 2*2 bits multiplier in CCC2 space (Boolean algebra).

$$
\begin{aligned}
O0 &= A0B0 \\
O1 &= A1\overline{B1}B0 + A0B1\overline{B0} + A1\overline{A0}B0 + \overline{A1}A0B1 \\
O2 &= A1\overline{A0}B1 + A1B1\overline{B0} \\
O3 &= A1A0B1B0
\end{aligned}
\qquad (6)
$$

$$\begin{array}{llll} \Diamond & & & \\ \Diamond & \Diamond & \Diamond & \\ \Box & \Diamond & & \Diamond \\ \Box & \Diamond & & \end{array} \qquad (7)$$

The equations (6) & (7) shows that the Boolean design required to 26 gates while the CCC2 required to 9 gates.

6.2- 1x1 quartered multiplication in CCC4

The truth table of 1*1 bits multiplier under CCC4 space is shown in Fig (6). The results of the simplification of the truth table are shown in equation (8) for the CCC4 space. This design required to 22 gates in CCC4.

$$O0 = (A\boxed{1}\mathrm{B}) + (\underline{A\boxed{1}\mathrm{B}}) + \overline{(A\boxed{3}\,\mathrm{B})} + \overline{(A\Diamond 3\,\mathrm{B}) + (A\Diamond 3\,\overline{\mathrm{B}})} + \overline{(A\Diamond 3\,\mathrm{B})} \qquad (8)$$

$$\mathrm{O1} = \overline{(A\Diamond 3\,\overline{\mathrm{B}})} + \overline{(\overline{A}\Diamond 3\,\mathrm{B})} + \overline{(A\Diamond 3\,\mathrm{B})}$$

Inputs		**Outputs**	
A	B	O0	O1
0	0	0	0
0	1	0	0
0	2	0	0
0	3	0	0
1	0	0	0
1	1	1	0
1	2	2	0
1	3	3	0
2	0	0	0
2	1	2	0
2	2	0	1
2	3	2	1
3	0	0	0
3	1	3	0
3	2	2	1
3	3	1	2

Fig(6): The truth table of 1*1 bits multiplier in CCC4 space.

7- The Advantages of the CCCi Space MVL

This section will clear the advantagesof the CCCi space, it has two sub-sections, first is the general CCCi MVL advantageswhile the second shows the advantagesand some important relations for CCC2 and CCC4.

7.1- General CCCi MVL Advantages

The new theory has given a new form of the MVL and this form has made it a package of advantagesdescribed as successful or useful. Someadvantagesare fundamental conditions for successful MVL. The most important general advantages for the CCCi space are:

1- Comprehensivenessall cases:

Any case of inputs in MVL hasn’tcodein the output will require to special solution or circuit (interfacing and additional cost) or the designer must reject this case from the design. This condition will enforce as condition for success.Hence, in section 5.1 the truth table in Fig (3b) the case of (01) hasn’t output (has 0 output for all gates) but it solved with additional X gate as the case of XF0 in Fig (3b).This condition is not realized in most previous works in MVL, therefore they not completely useful.

***2- Unique solution for each case*:**

The output of the functions in CCCi (as in Figures (3 and 4)) is unique value. This condition gives a more flexibility to the system designer. The XOR gate in Fig (2b) has some ambiguous in its results because the output represent an overlap between 01 and 10 cases while CCC2 solved this case using two combinations F0 and XF0.

3- Logical meaning:

This advantage will simplify the design algorithms and simplify the implementation of the functions. Some previous works such asRef. [11], Ref. [12] and Ref. [25] required to truth tables to define the relations between inputs and outputs while CCCi give a simple logical equations to define the relations between inputs and outputs as in equations (2, 3, 4 and 5).

4- Minimum number of functions (gates):

Generally the decreasing of the functions number will simplify the design algorithms. Hence the CCCispace has very low number of functions it require to $2(i+\log_2^i+1)$ functions,or in details the CCC2 has 7 functions and the CCC4 has 14 functions.As example the Ref. [25] and Ref. [22] shows the number of functions is $O(i^i)$it mean 16 gates(functions) for the binary and 256 gates for 4 MVL.

5- High flexibility:

The high flexibility functions will reduce the number of gates that required implementing any circuit or system. The functions ofthe CCCi space are flexible as example the design example in section (6.1) shows that the design using Boolean algebra required to 26 gates while the same circuit design using CCC2 required to 9 gates. Generally, this advantage has tradeoff with the advantage 4. The proposed relations must be balance between advantages 4 and 5.

6- Simple design:

This advantage is a result to the advantages 4 and 5 so it also depend on the good balance between the advantages 4 and 5.The different practical design examples (as D-flip-flop, adder and MUX) shows the CCCi space give a simple design, so its approaches are suitable for the object oriented systems.

7- Easy to implement in semiconductors media:

This advantage is satisfiedin the functions of the CCCi because all the functions (the functions of convert phase and the equations (2, 3, 4 and 5)) are implemented in the real word as electronic circuits.This property gives the CCCi system facilityto successful as a commercial systems.

8- Compatible with binary logic:

All the CCCi spacefunctions can be implemented using the classical logic so they can be use inside any binary system with a simple interface as hybrid systems.

7.2- Special CCCi MVL Advantages

This sub-section will discuss some additional special advantages for the CCCi application in the MVL field. It focuses on the CCC2 and CCC4 spaces as practical cases to use in the future to substitutes the Boolean logic.

1- Special Advantages of the CCC2Space

The nearest form to the CCC2 space MVL is the Boolean algebra but the practical applications show that it is more efficient from then the Boolean algebra. As example the 2*2 bits multiplier required in Boolean algebra to 26 gates as in equation (6) while the CCC2 space requires to 9 gates as in equation (7) ie.less than 35% from the cost of Boolean algebra. The results of some practical applications such as D-flip-flop, adder, MUX, and other standardcircuits that used in the ALU unit show the ratio of the gates that used in CCC2 with respect to the Boolean algebra fill in the range 20%-to-120% with average less than 65%.
Also the CCC2 is more simplicity from the Boolean algebra because it required to define 6 gates (NOT, E0, E1, F0, X and MAX) while Boolean algebra required to define 7 gates

(NOT, AND, NAND, XOR, XNOR, OR and NOR) with special function to solve the overlap in XOR and XNOR gates.

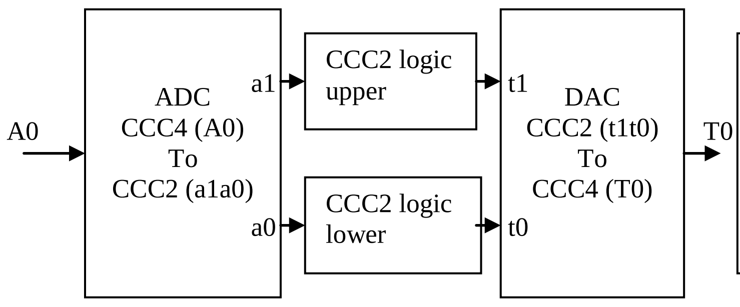
2- Special Advantages of the CCC4Space

The design example of the 1*1 bits multiplier that required to 22 gates the equivalent 2*2 bits multiplier in CCC2 required to 9 gates ie. about 250%. This ratio reversed in the complex design as example the 2*2 bits in CCC4 in compare with the 4*4 bits in CCC2 required to about 90%.

3- The CCC2 and CCC4 Relation

This point refers to the advantage 8 (compatible with binary logic)but this relation is more simplicity.Any circuit implement using CCC2 can be satisfy using CCC4 if all the values except the 0s in the CCC4 space are convert to 1s.

While any function implement using CCC4 can be implement as two bits CCC2. As example,the implement of UN function in CCC4 in terms of the CCC2 required to NOT gate for the upper bit and direct pass for lower bit that output of the CCC4 to CCC2 converter as in Fig (7).



(a)

A0 CCC4	a1a0 CCC2	T0 UN	t1t0 $\overline{a1}a0$
0	00	2	10
1	01	3	11
2	10	0	00
3	11	1	01

(b)

Fig(7): The implementation of CCC4 in circuit of CCC2 (a) general diagram, (b) converter function with example of the results of UN gate.

4- CCCi for Powers of Two

The relations between CCC2 and CCC4 can be expand to the relation betweenthe CCC2 and the CCC8 so they relations found with all the CCCi if *i* is a power of 2 (2, 4, 8, 16,…..). Same case repeated with CCC4 and CCC16 and so on.

The relations between CCC2 and the general powers of two CCCi are repeated with other values of i but these cases (not powers of 2) required to complex interfaces to convert the values from CCCi space to CCC2. As example the relation between CCC2 and CCC3 has same properties of the CCC2 and the CCC4 but with complex interface and lower efficiency.

References

1. Fergal Tuffy, "**Multiple Valued Logic Circuits using RTDs**", BEng (Hons) Electronics and Computing, Department of Informatics, in the Magee Campus, 2007, www.infm.ulst.ac.uk/~projects/ireport.PDF.
2. Stanley L. Hurst, "**Multiple -Valued Logic – Its Status and Its Future**", IEEE Transactions on Computers - TC , vol. 33, no. 12, pp. 1160-1179, 1984 DOI: 10.1109/TC.1984.1676392.
3. K.C. Smith, "**Multiple -Valued Logic: A Tutorial and Appreciation**", Computer archive Volume 21, Issue 4 (April 1988), portal.acm.org/citation.cfm?id=44844.
4. K.C. Smith and P. Glenn Gulak, "**Prospects for Multiple -Valued Integrated Circuits**", IEICE Trans. Electron.E76-C(3), pp. 372- 382, Mar. 1993, portal.acm.org/citation.cfm?id=290367.
5. This is a file in the archives of the Stanford Encyclopedia of Philosophy,Copyright © 2000Siegfried Gottwaldgottwald@rz.uni-leipzig.de.
6. fuzzy landmark-based navigation", Journal of Multiple-Valued Logic and Soft Computing, Vol. 9, pp. 195-220, Old City Publishing, 2003, www.cs.cmu.edu/~didac/ct/publications_ct.htm.
7. Josep Maria Font, "**An Abstract Algebraic Logic view of some multiple-valued logics**", University of Barcelona,Physica-Verlag GmbHHeidelberg, Germany, Germany©2003, ISBN:3-7908-1541-1.
8. Hiroki Nakahara, Tsutomu Sasao, and Munehiro Matsuura, "**Comparison of Heterogeneous Multi-valued Decision Diagram Machines for Multiple-output Logic Functions**", 41st IEEE International Symposium on Multiple-Valued Logic, 2011.
9. Colin Keng-Yan Tan, "**Belief Augmented Frames**", a thesis submitted for the degree of doctor of philosophy department of computer science school of computing national university of Singapore 2003.
10. S. N. Yanushkevich, J. T. Butler, G. W. Dueckz and V. P. Shmerko, "**Experiments on FPRM Expressions for Partially Symmetric Logic Functions**", Proceedings of the 30th IEEE International Symposium on Multiple Valued Logic, Portland, Oregon, USA, pp.141-146, 2000.
11. Harald Sack, Elena DubrovaandChristophMeinel, „Mod-p Decision Diagrams:A Data Structure for Multiple-Valued Functions",Proceedings of Int. Symposium on Multiple Valued Logic, Portland, USA, pp. 233-238, 2000.
12. Robert K Brayton and Sunil P Khatri, " **Multi-valued Logic Synthesis**", IEEE/ACM 11th International Workshop on Logic & Synthesis Chateau Sonesta Hotel, New Orleans, Louisiana June 4-7, 2002.

13. C. Lang and B. Steinbach, “**Decomposition of Multi-Valued Functions into Min- and Max-Gates**“,Proc. of the 31st IEEE International Symposium on Multiple-Valued Logic, May 22-24, 2001, Warsaw, pp 233 – 267, web.cecs.pdx.edu/~alanmi/research/bidec/bidec.htm.

14. D. Michael Miller Gerhard W. Dueck and Dmitri Maslov, “**A Synthesis Method for MVL Reversible Logic**“,Multiple-Valued Logic, IEEE International Symposium on, pp. 74-80, 34th International Symposium on Multiple-Valued Logic (ISMVL'04), 2004, http://doi.ieeecomputersociety.org/10.1109/ISMVL.2004.1319923.

15. De Vos, A., B. Raa and L. Storme, “**Generating the Group of Reversible Logic Gates**,” J. of Physics A: Mathematical and General, 35, pp. 7063-7078, 2002.

16. D. Venkat Reddy, Ch. D. V. ParadesiRao, E. G. Rajan, “**Sequential Circuits in the Framework Of (2n+1)-ary Discrete Logic**”, IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.7, July 2008.

17. Anas Al-Rabadi, and MarekPerkowski, “**Multiple-Valued Galois Logic S/D Trees, S/D Canonical Forms, and their Complexity**”, Dept. of Electrical and Computer Engineering ,www.ee.pdx.edu/~mperkows/=PUBLICATIONS/PDF2003/WARAQAH.doc.

18. Mozammel H. A. Khan and Marek A. Perkowski, “**Evolutionary Algorithm Based Synthesis of Multi-Output Ternary Functions Using Quantum Cascade of Generalized Ternary Gates**”, Preliminary version of this work will be presented in the Congress of Evolutionary Computing 2004 (CEC 2004), Portland, Oregon, USA, 19-23 June 2004.

19. Satyendra Ravi Prasad RajuDatla, “**Design and Validation of Quaternary Arithmetic Circuits**”, A Dissertation Presented to the Graduate Faculty of School of Engineering Southern MethodistUniversity, December 19th 2009.

20. T. Baba and T.Uemura, “**Development of InGaAs-Based Multiple-Junction Surface Tunnel Transistors for Multiple-Valued Logic Circuits**”, IEEE International Symposium on Multiple-Valued Logic,pp7-12, http://csdl.computer.org/dl/proceedings/ismvl/1998/8371/00/83710007.pdf.

21. Turgay TEMEL and Avni MORGUL,”**Implementation of Multi-Valued Logic Gates Using Full Current-Mode CMOS Circuits**”, Analog Integrated Circuits and Signal Processing, Volume 39 Issue 2, May 2004 Kluwer Academic PublishersHingham, MA, USA, doi>10.1023/B:ALOG.0000024066.66847.89.

22. MojtabaJamalizadeh, FazelSharifi, Mohammad HosseinMoaiyeri, KeivanNavi and OmidHashemipour, “**Five new MVL current mode differential absolute value circuits based on carbon nano-tube field effect transistors (CNTFETs)**”, Nano-Micro Lett. 2,227-234 (2010).doi:10.5101/nml.v2i4.p227-234.

23. Ashkan Khatir1 , ShaghayeghAbdolahzadegan and ImanMahmoudi, “**High Speed Multiple Valued Logic Full Adder Using Carbon Nano Tube Field Effect Transistor**”,

International Journal of VLSI design & Communication Systems (VLSICS) Vol.2, No.1, March 2011.

24. MarekPerkowski, “**New Awards for Research in Multiple-Valued Logic and Soft-Computing and a student competition**”, Journal on Multiple-Valued Logic and Soft Computing, 2005.
25. www.ece.pdx.edu/Faculty/Perkowski.php.
26. MarekPerkowski, David Foote, Qihong Chen, Anas Al-Rabadi, LechJozwiak, "**Learning Hardware Using Multiple-Valued Logic, Part 1: Introduction and Approach**," IEEE Micro, vol. 22, no. 3, pp. 41-51, May/June, 2002.
27. Lala, P. K., “**Combinational Logic Design, in Principles of Modern Digital Design**”, John Wiley & Sons, Inc., Hoboken, NJ, USA. doi: 10.1002/9780470125212.ch3, 2006.

Acknowledgment

I try to solve this problem since 15 years and I have more than 5 serious attempts withbeautiful ideas I have not posted them because they not fulfill the terms of the complete success of this research, but I think this work is worththe publishing and development.