The Construction of Universal Mulitple Processing Unit based on De Bruijn Graph

Chun-Myoung Park[1] and Hong-Bok Song[2]
[1] Division of Electrical Electronic & Information Engineering, Computer Engineering Major,
Chungju National University, Korea.
123 Iryumyun Gumdanri, Chungju City, 380-702, Chungbuk, Korea
Tel. +82-43-841-5346, Fax.: +82-43-841-5340
[2] Division of Electrical Electronic Information Communication,
Dongeui University, Korea.
San24 Gayadong Busanjingu, Busan City, 614-714, Korea
e-mail : cmpark@gukwon.chungju.ac.kr, hbsong@hyomin.dongeui.ac.kr

Abstract: This paper presents a method of constructing the universal multiple processing element unit(UMPEU) based on De Bruijn Graph. The proposed method is as following. Firstly we propose transformation operators in order to construct the De Bruijn graph using properties of graph. Secondly we contruct the transformation table of De Bruijn graph using above transformation operators. Finally we construct the De Bruijn graph using transformation table. The proposed UMPEU is capable of constructing the De Bruijn geraph for any prime number and integer value of finite fields. Also the UMPEU is applied to fault-tolerant computing system, pipeline class, parallel processing network, switching function and its circuits.

1. Introduction

Recently the research for efficiently suitable multiple processing structure of VLSI structure is studied.
Specially parallel processing based on De Bruijn graph[1-2] is studied. These are linear array, Ring, CBT(Complete Binary Tree), TM(Tree Machines), SE(Shuffl-Exchage), CCC(Cube Connected Cycles). Also it is introduced to apply fault-tolerant computing system.
B.Arazi[3] propose the generator of De Bruijn Sequence using 2n linear equation increasing root of polynomial. A.Lempel[4] research the homomorphism of De Bruijn graph using binary logic feedback register.
D.Goldfeld and T.Etzion[5] discussed properties of length n mono path and length 2n mono path and applied to internal connect of network.
M.A.Sridhar and C.S.Raghavendra[6] proposed the method of constructing the fault tolerant network based on De Bruijn graph.
These research is based on binary De Bruijn graph(BDBG) and BDBG is used to pipeline class, multiplex class, ascend and descend class. These types are classified to SIPO(Serial Input parallel Output), PISO(Parallel Input Serial Output), PIPO(Parallel Input and Parallel Output) and HIHO(Hybrid Input and Hybrid Output).
This paper's constructing is as following. chapter 2 describe the properties of graph theory and chapter 3 discuss the each transformation operator. Chapter 4 constructed the De Bruijn graph transformation table and chapter 5 constructed the De Bruijn graph over Finite fields. Finally chapter 6 describe the characteristics of the proposed UMPEU 4 and application fields, future research fields and prospectiveness.

2. Properties of Graph

This chapter described the properties of graph theory that we expanded this paper.

In generally, the graph is expressed as following.

$$G(V, E) \qquad (2\text{-}1)$$

Where, V is the set of node for nonempty set and E is a set of 2 subset in node set.

We called the neigboared of V in case of all adjacent node for node V, denoted in N(V), and degrr of V in case of the number of edge that linked the node V, denoted deg(V). Specially, we denoted $deg_G(V)$ in case of graph G.
On the other hand, the isomorphism means the graph that have the same adjacent and non-adjacent. That is, in the two graph G_1 and G_2, if one to one function and $F:V(G_1)\Rightarrow V(G_2)$ exist, we called isomorphism, it ce mappimg. Also, in case of isomorphism from G_2 to G_1, it is determied by inverse of F, F^{-1}. The arc is defined by the ordered pair in stead of set of edge between of nodes, we call ed directed graph or digraph. The number of arc that outcoming for node of digraph is denoteed outdegree of V, od(V), and incoming case, denoted indegree of V, id(v). And for degree is denoted by deg=id(V)+od(V).
Also, if node V is in level k, we called the child of V that have neighboar of V over (k+1) level and called the parent that have neighboar of V over (k-1) level.
We cited the refernce for the other useful properties.[7-8]

3. Transformation Operator

Iin this section, we discuss the transformation operator in oder to constructing the De Bruijn graph.

3.1 General transformation operator

[GOP#1] LR[digit stream] : one digit left rotate

$LR[D_{m-1}D_{m-2} \ldots\ldots D_1D_0] = [D_{m-2}D_{m-3} \ldots\ldots D_1D_0D_{m-1}]$
For example, each digit number over $GF(3^2)$ are D_1 and D_0, $D_i \in \{0,1,2\}(i=0,1)$ and $P=3$. If $D_1D_0=02$ then $LR[D_1D_0=02]=[D_0D_1=20]$.

[GOP#2] RR[digit stream] : one digit right roatate

$RR[D_{m-1}D_{m-2} \ldots\ldots D_1D_0] = [D_0D_{m-1}D_{m-2} \ldots\ldots D_2D_1]$

For example, each digit number over $GF(3^2)$ are D_1 and D_0, $D_i \in \{0,1,2\}(i=0,1)$ and $P=3$. If $D_1D_0=21$then $RR[D_1D_0=21]=[D_0D_1=12]$.

[GOP#3] CLR[digit stream] : left rotate of modP complement for MSD

$CLR[D_{m-1}D_{m-2} \ldots\ldots D_1D_0] = [D_{m-2}D_{m-3} \ldots\ldots D_1D_0D_{m-1}']$

For example, each digit number over $GF(3^2)$ are D_1 and D_0, $D_i \in \{0,1,2\}(i=0,1)$ and $P=3$. If $D_1D_0=20$ then $CLR[D_1D_0=20]=[D_0D_1'=02']=[00]$.

[GOP#4] CRR[digit stream] : right rotate of modP complement for LSD

$CRR[D_{m-1}D_{m-2} \ldots\ldots D_1D_0] = [D_0'D_{m-1}D_{m-2} \ldots\ldots D_2D_1]$

For example, each digit number over $GF(3^2)$ are D_1 and D_0, $D_i \in \{0,1,2\}(i=0,1)$ and $P=3$. If $D_1D_0=20$ then $CLR[D_1D_0=20]=[D_0'D_1=0'2]=[22]$.

For above general transformation operator, $D_i \in \{0,1,2, \ldots\ldots, P-1\}$ and D_i' is the complement of D_i.

3.2 modP transformation operator

[MOP#1] LR(K)modP[digit stream] : one digit left rotate of modP(MSD+K)

$LR(K)[D_{m-1}D_{m-2} \ldots\ldots D_1D_0]$
$= [D_{m-2}D_{m-3} \ldots\ldots D_1D_0 \, modP(D_{m-1}+K)]$

For example, each digit number over $GF(3^2)$ are D_1 and D_0, $D_i \in \{0,1,2\}(i=0,1)$ and $P=3$. If $D_1D_0=02$ and $K=2$ then $LR(K)[D_1D_0=02]=LR(2)[D_0D_1=20]=LR[(2)[2 \bmod 3(0+2)]=[22]$.

[MOP#2] RR(K)modP[digit stream] : one digit right rotate of modP(MSD+K)

$RR(K)[D_{m-1}D_{m-2} \ldots\ldots D_1D_0]$
$= [modP(D_0+K) \, D_{m-1}D_{m-2} \ldots\ldots D_2D_1]$

For example, each digit number over $GF(3^2)$ are D_1 and D_0, $D_i \in \{0,1,2\}(i=0,1)$ and $P=3$. If $D_1D_0=10$ and $K=2$ then $RR(K)[D_1D_0=10]=RR(2)[D_0D_1=10]=RR(2)[\bmod 3(0+2)]=[21]$

[MOP#3] CLR(K)modP[digit stream] : left rotate of modP complement for (MSD+K)
$CLR(K)modP[D_{m-1}D_{m-2} \ldots\ldots D_1D_0]$
$= [D_{m-2}D_{m-3} \ldots\ldots D_1D_0 (D_{m-1}+K)']$

For example, each digit number over $GF(3^2)$ are D_1 and D_0, $D_i \in \{0,1,2\}(i=0,1)$ and $P=3$. If $D_1D_0=10$ and $K=2$ then $CLR(K)modP[D_1D_0=10]=CLR(2)mod3[D_1D_0=10]=CLR(2)mod3[0\ (1+2)']=[02]$

[MOP#4] CRR(K)modP[digit stream] : right rotate of modP complement for (LSD+K)

$CRR(K)modP[D_{m-1}D_{m-2} \ldots\ldots D_1D_0]$
$= [(D_0+K)' \, D_{m-1}D_{m-2} \ldots\ldots D_2D_1]$

For example, each digit number over $GF(3^2)$ are D_1 and D_0, $D_i \in \{0,1,2\}(i=0,1)$ and $P=3$. If $D_1D_0=10$ and $K=2$ then $CRR(K)modP[D_1D_0=10]=CRR(2)mod3[D_1D_0=10]=CRR(2)mod3[(0+2)'\ 1]=[01]$

For above modP transformation operator, $D_i \in \{0,1,2, \ldots\ldots, P-1\}$ and D_i' is the complement of D_i, also $0 \leq K \leq P-1$(K=integer)

3.3 Reverse operator

In section 3-1, the general transformation operator GOP#1 LR[digit stream] and GOP#2 RR[digit stream] is reverse relationship, also GOP#3 CLR[digit stream]and GOP# 4 CRR[digit stream] is reverse relationship.
In section 3-2, MOP#1 LR(K1)modP[digit stream] and MOP#2 RR(K2)modP[digit stream] is reverse relationship, also MOP#3 CLR(K1)modP[digit stream] and MOP#4 CRR(K2)modP[digit stream] is reverse relationship. Where the condition is K1+K2=P.

The following is summarized to the reverse relationship for each transformation operators.

$LR[\text{digit stream}]^{-1} = RR[\text{digit stream}]$
$RR[\text{digit stream}]^{-1} = LR[\text{digit stream}]$
$CLR[\text{digit stream}]^{-1} = CRR[\text{digit stream}]$
$CRR[\text{digit stream}]^{-1} = CLR[\text{digit stream}]$
$LR(K1)modP[\text{digit stream}]^{-1}$
$= RR(K2)modP[\text{digit stream}]$
$RR(K2)modP[\text{digit stream}]^{-1}$
$= LR(K1)modP[\text{digit stream}]$
$CLR(K1)modP[\text{digit stream}]^{-1}$
$= CRR(K2)modP[\text{digit stream}]$
$CRR(K2)modP[\text{digit stream}]^{-1}$
$= CLR(K1)modP[\text{digit stream}]$

4. Adjacent Matrix and Transformation Table of De Bruijn Graph

In this Section, we obtain adjacent matrix and construct tranformation table of De Bruijn graph.

4.1 Adjacent matrix Construction

The loop in the adjacent matrix means self element, 1 means the connection of edge for its row and column, 0 means non-connection.

For the first example, we obtain the adjacent matrix over GF(3^2) in case of P=3 and m=2, it is as following.

	(00)	(01)	(02)	(10)	(11)	(12)	(20)	(21)	(22)
(00)	loop	1	1	1	0	0	1	0	0
(01)	1	0	0	1	1	1	1	0	0
(02)	1	0	0	1	0	0	1	1	1
(10)	1	1	1	0	1	0	0	1	0
(11)	0	1	0	1	loop	1	0	1	0
(12)	0	1	0	0	1	0	1	1	1
(20)	1	1	1	0	0	1	0	0	1
(21)	0	0	1	1	1	1	0	0	1
(22)	0	0	1	0	0	1	1	1	loop

For the 2nd example, we obtain the adjacent matrix over GF(5) in case of P=5 and m=1, it is as following.

	(0)	(1)	(2)	(3)	(4)
(0)	loop	1	1	1	
(1)	1	loop	1	1	1
(2)	1	1	loop	1	1
(3)	1	1	1	loop	1
(4)	1	1	1	1	loop

,

4.2 Transformation Table of De Bruijn Graph

For the first example, we obtain the transformation table of De Bruijn graph over GF(3^2) in case of P=3 and m=2, it is as following.

Table 4-1. Tranformation table of De Bruijn graph over GF(3^2)

	(00)	(01)	(02)	(10)	(11)	(12)	(20)	(21)	(22)
LR(0) mod3	(00)	(10)	(20)	(01)	(11)	(21)	(02)	(12)	(22)
RR(0) nod3	(00)	(10)	(20)	(01)	(11)	(21)	(02)	(12)	(22_
LR(1) mod3	(01)	(11)	(21)	(02)	(12)	(22)	(00)	(10)	(20)
RR(1) mod3	(10)	(20)	(00)	(11)	(21)	(01)	(12)	(22)	(02)
LR(2) mod3	(02)	(12)	(22)	(00)	(10)	(20)	(01)	(11)	(21)
RR(2) mod3	(20)	00)	(10)	(21)	(01)	(11)	(22)	(02)	(12)

In the above example, LR(0)mod3 tranformation operator equal to RR(0)mod3 tranformation operator.

For the 2nd example, we obtain thetransformation table of De Bruijn graph over GF(5) in case of P=5 and m=1, it is as following.

Table 4-2. Tranformation table of De Bruijn graph over GF(5)

	(0)	(1)	(2)	(3)	(4)
LR(0)mod5	(0)	(1)	(2)	(3)	(4)
RR(0)mod5	(0)	(1)	(2)	(3)	(4)
LR(1)mod5	(1)	(2)	(3)	(4)	(0)
RR(1)mod5	(1)	(2)	(3)	(4)	(0)
LR(2)mod5	(2)	(3)	(4)	(0)	(2)
RR(2)mod5	(2)	(3)	(4)	(0)	(2)
LR(3)mod5	(3)	(4)	(0)	(1)	(3)
RR(3)mod5	(3)	(4)	(0)	(1)	(3)
LR(4)mod5	(4)	(0)	(1)	(2)	(4)
RR(4)mod5	(4)	(0)	(1)	(2)	(4)

,

5. The Construction of De Bruijn Graph

In this section, we construting the De Bruijn graph absed on Contents of section 3 and section 4.

For the 1st example, we construct the De Bruijn graph over GF(2^4), it is as following Fig. 5-1.

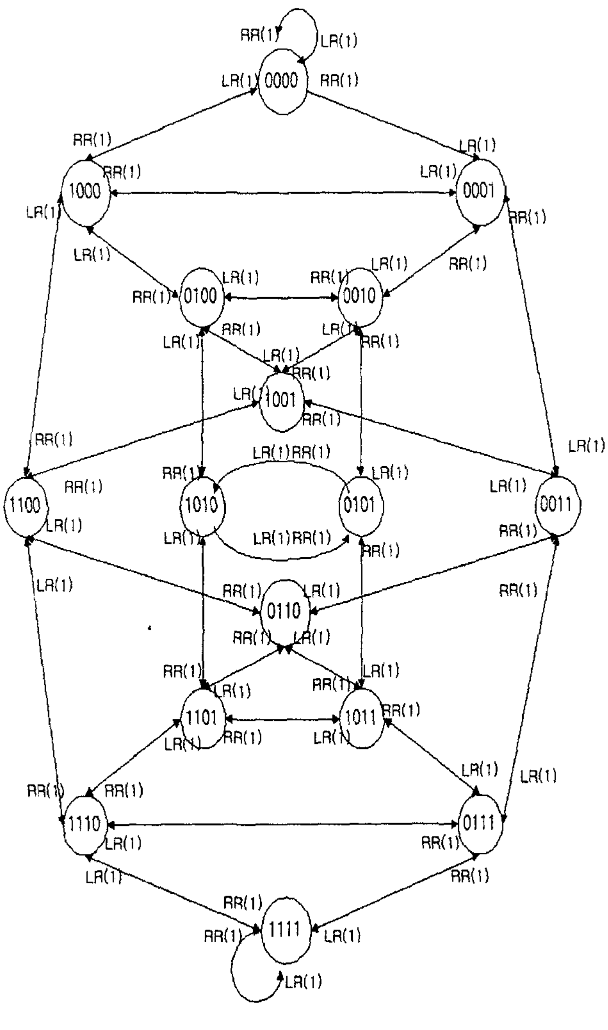

Fig. 5-1. The De Bruijn grpah construction over GF(2^4).

For the 2nd example, we construct the De Bruijn graph over $GF(3^2)$ and $GF(3^3)$, it is as following Fig. 5-2 and Fig. 5-3.

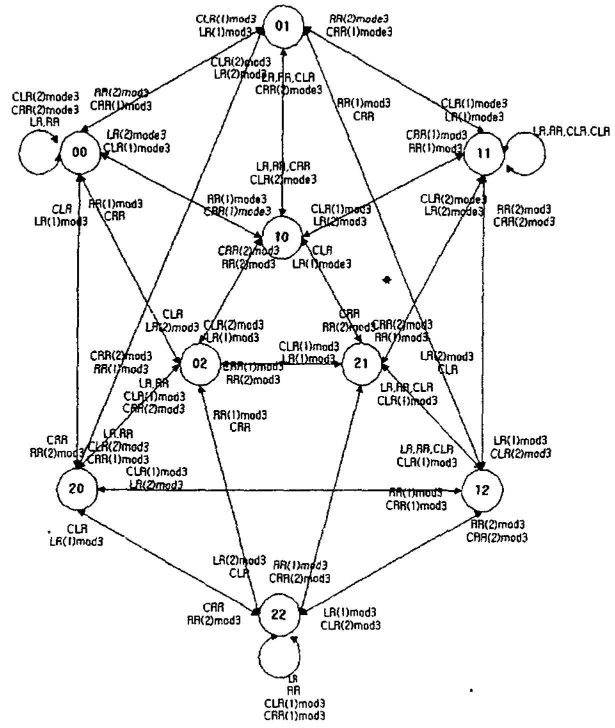

Fig. 5-2. The De Bruijn grpah construction over $GF(3^2)$.

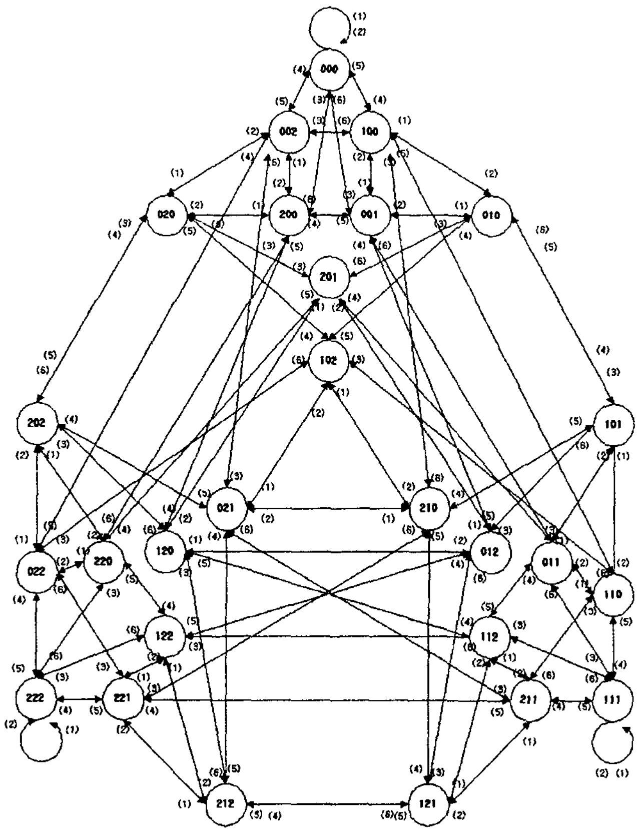

Fig. 5-3. The De Bruijn grpah construction over $GF(3^3)$.

For the 2rd example, we construct the De Bruijn graph over GF(5), it is as following Fig. 5-4.

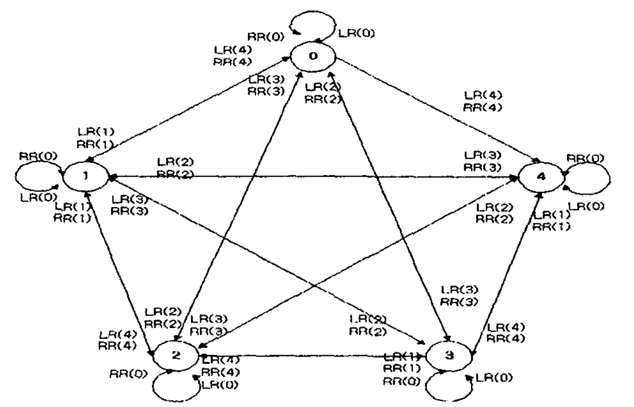

Fig. 5-4. The De Bruijn grpah construction over GF(5)

5. Conclusion

In this paper, we proposed the method of constructing the universal multiple processing element unit(UMPEU) based on De Bruijn graph.
The proposed method is applied to free transformation of digital logic systems and switching circuit.
Also, the proposed method can be expanded for expasion of P and applied to fault-tolerant computing system.
Future research is demanded to combinational digital logic systema and sequential digital logic systems.

References

[1] R.A.Games,"A Generalized Recursive Construction for De Bruijn Sequencees", IEEE Trans. INFORMATION and THEORY,Vol.IT-29, pp.843-850, Nov. 1993.

[2] M.R.Samathan and D.K.Pradhan,"The De Bruijn Mutiprocessor Network: A Versatile Parallel Processing and Dorting Net-works for VLSI", IEEE Tran. Comut. Vol.38,NO.4, pp.567-581, Apr. 1971.

[3] B.Arazi,"On the Sythesis of De-Bruijn Sequences", INFORMATION and CONTROL 49., pp.81-90, 1981.

[4] A.Lempel,"On a Homomorphism of the DE- Bruijn Graph and its Applications to the Design of Feedback Shift Registers", IEEE Trans. Comut. Vol.C-19, NO12, pp.1204-1209, Dec. 1970.

[5] D.Goldfeld and T.Etziopn,"UPP Graphs and UMPA Networks-Architecture for Parallel Systems", IEEE Trans. Comput. Vol.41, NO.11, pp.1479-1483, Nov. 1992.

[6] M.A.Sridhar and C.S.Reghavendra,"Fault-Tolerant Networks based on the De Bruijn Graph", IEEE Trans. Comut. Vol.41,NO.11, pp.1167-1174, Oct. 1991.

[7] R.Gould, Graph Theory, The Bejamin/Cummings Publishing Company, Inc., 1988.

[8] M.Gondran and M.Minoux, Graph and Algorithms, A Wiley-interscience publication, Toronto, 1984.