Switching Function using Edge-Valued Decision Diagram

Chun-Myoung Park, *Member, KIMICS*

Abstract— **This paper presents a method of constructing the switching function using edge-valued decision diagrams. The proposed method is as following. The edge-valued decision diagram is a new data structure type of decision diagram which is recently used in constructing the digital logic systems based on the graph theory. Next, we apply edge-valued decision diagram to function minimization of digital logic systems. The proposed method has the visible, schematic and regular properties.**

Index Terms— **Switching function, decision diagram, edge-valued decision diagram, function minimization etc.**

I. INTRODUCTION

IN recently, the method coming up important method that constructing the digital logic systems[1,2] based on graph theory[3-5]. S.B.Aker[6] defined concept firstly of decision diagram(DD) that is directed acyclic graph(DAG) type, also derived from binary decision diagram(BDD) above DD. R.E.Bryant[7] research firstly the Boolean function construction and function minimization using BDD that was proposed by S.B.Akers. Also, Yung-Te Lai etal[8] and S.B.K. Vrudhula etal[9] propose binary edge-valued decision diagram(EVBDD) that improved BDD's drawback. That has advantage higher state instead of Boolean function representation expression, also hierarchical verification. Therefore EVBDD has canonical representation characteristics. Also, D. M. Miller[10] propose firstly multiple-valued[11,12] decision diagram which extended the binary decision diagram. This paper propose algorithm of constructing the multiple-valued edge-valued decision diagram (MVEVDD) that extended EVBDD, also propose a method of constructing the digital logic systems after minimizing the p-valued n variable function. This paper's organization is as following. In section 2, we describe the switching function which is very important for all digital logic systems involve the embedded systems et al. In section 3 we discuss the decision diagram : properties of graph, function minimization using the decision diagram, edge-valued binary decision diagram, variable value extension of EVBDD, mathematical background, Reed-Muller extension. Also, in section 4, we discuss algorithm which is the generation MVEVDD form EVBDD and we extend to case of multiple variable. In section 5, we discuss the comparison result after compare proposed method with former methods, finally we review the characteristics for proposed method of switching function using edge-valued decision diagram.

Manuscript received March 23, 2011; revised June 3, 2011; accepted June 7, 2011.

Chun-Myoung Park is with the Department of Computer Engineering, Chungju National University, Chungju, 380-702, Chungbuk, Korea (Email: cmpark@cjnu.ac.kr)

II . SWITCHING FUNCTION

The model in which every decision is based on the comparison of two numbers within constant time is calledsimply a decision tree model. It was introduced toestablish computational complexity of sorting and searching. The simplest illustration of this lower bound technique is for the problem of finding the smallest number among n numbers using only comparisons. In this case the decision tree model is a binary tree. Algorithms for this searching problem may result in n different outcomes (since any of the n given numbers may turn out to be the smallest one). It is known that the depth of a binary tree with n leaves is at least $\log n$, which gives a lower bound of $\Omega(\log n)$ for the searching problem. However this lower bound is known to be slack, since the following simple argument shows that at least $n - 1$ comparisons are needed: Before the smallest number can be determined, every number except the smallest must "lose" (compare greater) in at least one comparison. Along the same lines the lower bound of $\Omega(n\log n)$ for sorting may be proved. In this case, the existence of numerous comparison-sorting algorithms having this time complexity, such as mergesort and quicksort, demonstrates that the bound is tight. In mathematics, a (finitary) Boolean function (or switching function) is a function of the form $f : B^k \rightarrow B$, where $B = \{0, 1\}$ is a Boolean domain and k is a non-negative integer called the arity of the function. In the case where $k = 0$, the "function" is essentially a constant element of B. Every k-ary Boolean formula can be expressed as a propositional formula in k variables $x_1, \ldots, x_k$, and two propositional formulas are logically equivalent if and only if they express the same Boolean function. There are 2^{2k} k-ary functions for every k. A Boolean function describes how to determine a Boolean value output based on some logical calculation from Boolean inputs. Such functions play a basic role in questions of complexity theory as well as the design of circuits and chips for digital computers. The properties of Boolean functions play a critical role in cryptography, particularly in the design of symmetric key algorithms (see substitution box). Boolean functions are often represented by sentences in propositional logic, and sometimes as multivariate polynomials over GF(2), but more efficient representations are binary decision diagrams (BDD), negation normal forms, and propositional directed acyclic graphs (PDAG). In computational complexity and communication complexity theories the decision tree model is the model of computation or communication in which an algorithm or communication process is considered to be basically a decision tree, i.e., a sequence of branching operations based on comparisons of some quantities, the comparisons being assigned the unit computational cost. The branching operations are called "tests" or "queries". In this setting the algorithm in question may be viewed as a computation of a Boolean function $f:\{0,1\}^n \rightarrow (0,1)$ where the input is a series of queries and the output is the final decision. Every query is dependent on previous queries, therefore it is described as a binary tree. Several variants of decision tree models may be considered, depending on the complexity of the operations allowed in the computation of a single comparison and the way of branching. Decision trees models are instrumental in establishing lower bounds for computational complexity for certain classes of computational problems and algorithms: the lower bound for worst-case computational complexity is proportional to the largest depth among the decision trees for all possible inputs for a given computational problem. The computation complexity of a problem or an algorithm expressed in terms of the decision tree model is called decision tree complexity or query complexity.

III. DECISION DIAGRAM

In this section, we discuss properties of graph in order to analysis and synthesis decision diagram, also discuss the concept function minimization using decision diagram.

3.1 Properties of Graph

The graph represented as following expression (1) in generally.

$$G(V, E) \tag{1}$$

Where, V is finite set of nonempty node, and E is set of two subset branch in node set.

Also, $|V|$ is order of graph and means number of node, $|E|$ is size of graph and means number of branch. In case of set for all node that placed adjacently node, we call neighbored N(V), also we call degree of node deg(V) in case of number of node that happened in node V. Specially, we call $deg_G(V)$ that happened in graph G. If node V placed in level k, we call child that placed in adjacently in level k+1, also we call parent that placed in adjacently in level k-1. You can see references[3-5] in case of any other properties besides above properties.

3.2 Decision Diagram

Decision diagram is directed acyclic graph, also use parent node and children node, and node means logic variable and branch means input variable. The node of end was called terminal node, and means logic constant. Also, the node of starting was called root node and the other nodes are called non-terminal node and mean variables. Each branch from each node have logic variable in relation to its node and branch. The node in relation to any logic variable be able to multiple presence, and branch from each node is equal to number of input variable. Decision diagram has tree property, node of 1st level must one, also sub-tree occurred as much as input variable from root node. Decision diagram was changed according to sequence of variable that mean node, also multiple decision diagram exit for one logic variable. The terminal node's value equal to function's value. We refer to references besides above properties of decision diagram.

3.3 Function Minimization using by Decision Diagram

The decision diagram is visible for representation of logic function, and function minimization or simplification is easy using decision diagram. The number of nodes over BDD is equal to gate when we design logic circuit, therefore reduction of node over decision diagram is function minimization. In generally, BDD that was constructed with n variables have 2^{n+1}-1 node and 2^n terminal nodes. Therefore, minimization processing and ordering variable are important factors.

3.4 Edge-Valued Binary Decision Diagram

In this section, we discuss properties of edge-valued binary decision diagram(EVBDD) and variable extension of EVBDD. EVBDD have advantage for handling integer operation and Boolean expression but BDD have handling only Boolean expression.

3.5 Variable Value Extension of EVBDD

EVBDD' node have each variable allocation, then edge was defined according to variable value. Variable have all $\{0,1\}^n$'s value, if variable have constant input instead of binary value, variable must represent binary vector.

3.6 Mathematical Background

In this section, we discuss important mathematical properties that used in opened this paper.

3.6.1 Literal

We define following definitions in order that we extend from binary over BDD to multiple-valued.

[Definition 1] Let X_i is multiple-valued variable which have any value among set Pi={0, 1, 2, …… , (P-1)}. For any subset $S_i \subseteq P_i$, X_i^{Si} is literal which represent the following expression (2).

$$X_i^{Si} = \begin{cases} 1 & \text{if } S_i \in X_i \\ 0 & \text{if } S_i \notin X_i \end{cases} \quad (2)$$

3.6.2 Reed-Muller Expansion

In case of general P-valued n variable extension, we use Reed-Muller expansion(RME) which extended Boolean arithmetic operation using by mod arithmetic operation. Also we derived general expression of multiple-valued edge-valued decision diagram from that. The Boolean arithmetic operation is represented by following expression (3), because that logic sum and logic product in Boolean arithmetic operation be able to replace with mod2 sum and mod2 product.

$$F(X_1, X_2, \ldots\ldots, X_{n-1}, X_n) = \sum_{i=0}^{K-1} C_i X_1^{ei,1} X_2^{ei,2}, \ldots\ldots, X_{n-1}^{ei,n-1} X_n^{ei,n} \quad (3)$$

where, $K=2^n$ and $\sum$ is mod2, $e_{i,j}$(i=0,1,……,2n-2 ; j=1,2,……,n) is 0 or 1, also C_i is constant.

On the other hand, C_i obtained using by following expression (4)

$$T_n = \begin{bmatrix} Tn-1 0 \\ Tn-1 Tn-1 \end{bmatrix} \quad (4)$$

Where, $T_1 = \begin{bmatrix} 10 \\ 11 \end{bmatrix}$

IV. MULTIPLE-VALUED EDGE-VALUED DECISION DIAGRAM

4.1 Definition of MVEVDD

[Definition 2] MVEVDD was constructed with pair <C, F>, where C is constant and F has a node following two case.

(1)one terminal node which was represented by **0**

(2)Non terminal node V was constructed P+2 tuple <variable(V), $child_1$, $child_{m,1}$(V), …… , $child_{m,P-2}$(V), $child_r$(V), value>, where variable(V) is multiple-valued variable.

[Definition 3] P-valued edge-valued decision diagram <C, F> represent arithmetic operation C+ Ψ, where Ψ is function which represent by F. Also, **0** means constant function, <X, l, (m,1), (m,2), …… , (m,P-2), r, v> is arithmetic operation function $X(l+v)+(X-1)_{m,1}+ \ldots\ldots + \{X-(P-2)\}_{m,P-2} + \{X-(P-1)\}$

For example, the following expression is P-valued 3 variable conversion following expression (5).

$$F(X_1, X_2, X_3)=\beta+X_1[\delta_1+X_2\{\zeta_1+(\zeta_2+\zeta_4)X_3\}]+\sum_{j=0}^{P-1}(X_2-j)\,\delta_3 X_3+\sum_{i=0}^{P-1}\sum_{j=0}^{P-1}(X_1-j)[X_2\{\delta_2+\zeta_2\}+(X_2-j)\,\delta_3 X_3] \quad (5)$$

4.2 Algorithm of generation MVEVDD from EVBDD

In this section, we discuss the algorithm of generation multiple-valued edge-valued decision diagram from edge-valued binary decision diagram.

[STEP 1] Insert middle edge for the purpose to make node which P tuple <X, l, m1, …… ,mp-2, r,v> which represent arithmetic operation of MVEVDD.

[STEP 2] After insert middle edge, convert arithmetic operation function at each node to $X(l+v)+(X-1)_{m,1}+ \ldots\ldots +[X-(P-2)]_{m,p-2}+[X-(P-1)]_r$.

[STEP3] Apply threshold detector to each node, the result was represented in following expression (6).

$$X = \begin{cases} 1 & \text{if } X=m \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

The expression (6) has logic 1 in case of input variable's value is P-valued, and has 0 in any other case.

The following figure 1 depicts the algorithm of generation MVEVDD from EVBDD.

4.3 MVEVDD Extension in Multiple Variable

The number of coefficient for each degree in n-variable function expansion is as following expression (7).

$${}_nC_0+{}_nC_1+ \ldots\ldots +{}_nC_{n-1}+{}_nC_n = \sum_{i=0}^{n} {}_nC_i = 2^n-1 \quad (7)$$

therefore, expression (3) can expressed in following expression (8).

$$F(X_1, X_2, \ldots\ldots, X_{n-1}, X_n) = C_0 + \sum_{i=0}^{K-1} C_i X^{ei,1} X^{ei,2}, \ldots\ldots, X^{ei,n-1} X^{ei,n} \quad (8)$$

The processing of edge-valued according variable X(i=1, 2, …… ,n) is as following expression (9).

$$F\,|_{Xi}(i=1, 2, \ldots\ldots, n) \quad (9)$$

First, we extended P-valued case.

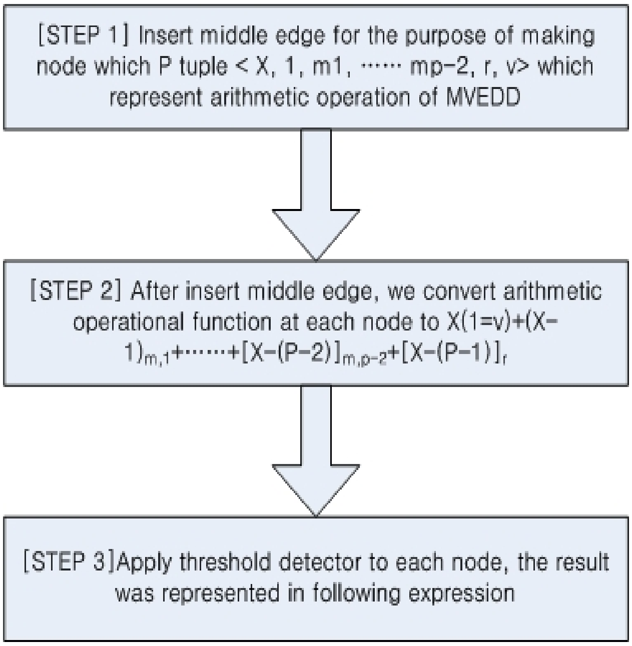

Fig. 1. Algorithm of generation MVEVDD from EVBDD.

$$F|_{Xi}= C_0+X_1(C_1+\sum_{i=2}^{K-1} C_i X_2^{ei,1} X_3^{ei,2}, \ldots\ldots, X_{n-1}^{ei,n-1} X_n^{ei,n}$$
$$+\sum_{j=2}^{K-1} C_i X_2^{ej,1} X_3^{ej,2}, \ldots\ldots, X_{n-1}^{ej,n-1} X_n^{ej,n})$$
$$+(1-X_1)(\sum_{j=2}^{K-1} C_i X_2^{ej,1} X_3^{ej,2}, \ldots\ldots, X_{n-1}^{ej,n-1} X_n^{ej,n}) \qquad (10)$$

$$F|_{Xi}= C_0+X_1(C_1+\sum_{i=2}^{K-1} C_i X_2^{ei,1} X_3^{ei,2}, \ldots\ldots, X_{n-1}^{ei,n-1} X_n^{ei,n}$$
$$+\sum_{j=2}^{K-1} C_j X_2^{ej,1} X_3^{ej,2}, \ldots\ldots, X_{n-1}^{ej,n-1} X_n^{ej,n}$$
$$+(X_1-1)(\sum_{j=2}^{K-1} C_j X_2^{ej,1} X_3^{ej,2}, \ldots\ldots, X_{n-1}^{ej,n-1} X_n^{ej,n})$$
$$+(X_1-2)(\sum_{j=2}^{K-1} C_j X_2^{ej,1} X_3^{ej,2}, \ldots\ldots, X_{n-1}^{ej,n-1} X_n^{ej,n})$$
:
:
$$+(X_1-(P-1))(\sum_{j=2}^{K-1} C_j X_2^{ej,1} X_3^{ej,2}, \ldots\ldots, X_{n-1}^{ej,n-1} X_n^{ej,n})$$
$$= C_0+X_1(C_1+\sum_{i=2}^{K-1} C_i X_2^{ei,1} X_3^{ei,2}, \ldots\ldots, X_{n-1}^{ei,n-1} X_n^{ei,n}$$
$$+\sum_{j=2}^{K-1} C_j X_2^{ej,1} X_3^{ej,2}, \ldots\ldots, X_{n-1}^{ej,n-1} X_n^{ej,n})$$
:
:
$$+\sum_{g=1}^{m-1} (X_1-g) (\sum_{j=2}^{K-1} C_i X_2^{ej,1} X_3^{ej,2}, \ldots\ldots, X_{n-1}^{ej,n-1} X_n^{ej,n}) \qquad (11)$$

where, $K=2^n$

Here, we put $F|_{X0}= F|_0=C_0$ for constant term, generalized n-variable expression is as following expression (12).

$$F(X_1, X_2, \ldots\ldots, X_{n-1}, X_n) =\sum_{i=0}^{m-1} F|_{X1} \qquad (12)$$

Here,

$$F|_{X0}= F|_0=C_0$$
$$F|_{X1}=\sum_{i=1}^{K-1} C_i X_1^{ei,1} X_2^{ei,2}, \ldots\ldots, X_{n-1}^{ei,n-1} X_n^{ei,n}$$
$$F|_{X2}=\sum_{i=2}^{K-1} C_i X_2^{ei,1} X_3^{ei,2}, \ldots\ldots, X_{n-1}^{ei,n-1} X_n^{ei,n}$$
:
:
$$F|_{Xn-1}=\sum_{i=n-1}^{K-1} C_i X_{n-1}^{ei,n-1} X_n^{ei,n}$$
$$F|_{Xn}=\sum_{i=n}^{K-1} C_i X_n^{ei,n}$$

Where, $K=2^n$

$$F|_{X1}=X_1(C_1+ F|_{X2}+F^*|_{X2}) + \sum_{g=1}^{P-1} (X_1-g) F^*|_{X2}$$
$$F|_{X2}=X_2(C_2+ F|_{X3}+F^*|_{X3}) + \sum_{g=1}^{P-1} (X_2-g) F^*|_{X3}$$
:
:
$$F|_{Xn-1}=X_{n-1}(C_{n-1}+ F|_{Xn}+F^*|_{Xn}) + \sum_{g=1}^{P-1} (X_{n-1}-g) F^*|_{Xn} \qquad (13)$$

Here, $F|_{X0}= F|_0=C_0$, and $F^*|_{Xi}$ is function which have non-edged branch.

Therefore, the generalized P-valued n-variable edged function can be represented as following expression (14).

$$F(X_1, X_2, \ldots\ldots, X_{n-1}, X_n) =$$
$$F|_{Xi}= X_i(C_i+ F|_{Xi+1}+F^*|_{Xi+1}) + \sum_{g=1}^{P-1} (X_{n-1}-g) F^*|_{Xi+1}$$

(i=1, 2, …… , n) (14)

The following figure2 depicted above MVEVDD.

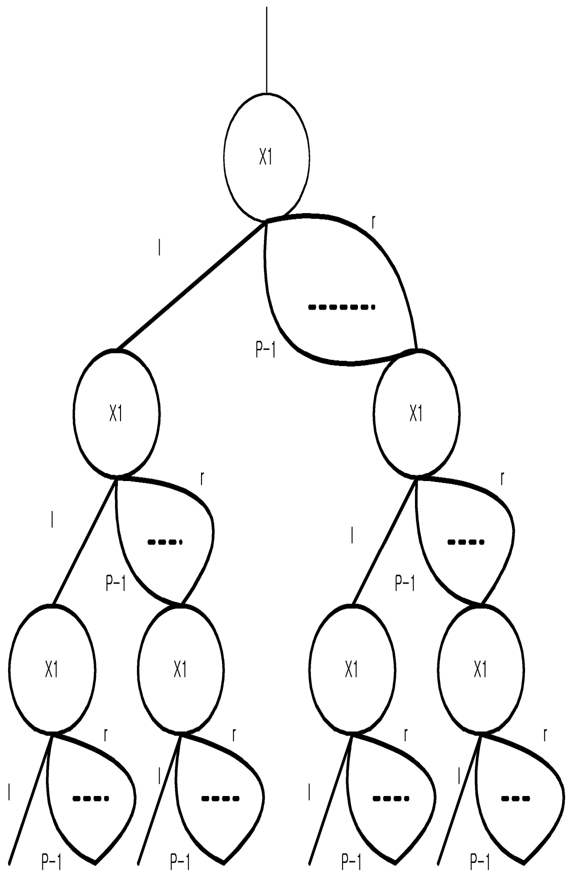

Fig. 2. General structure of P-valued n-variable MVEVDD.

V. COMPARISON AND DISCUSSION

In this section, we apply our proposed method to some example, we compare with result and investigate its result.

For example, we apply our algorithm for obtaining edge-valued decision diagram to Function $F(X_1, X_2, X3)=-2+5X_2+X_2X_3+3X_1X_2+4X_1X_2X_3-2X_1X_3+X_3$. Next, we obtain the transformation expression for EVBDD and EVTDD. First, we transfer function to EVBDD, it is as following expression (15).

$$F(X_1,X_2,X3)=-2+5X_2+X_2X_3+3X_1X_2+4X_1X_2X_3-2X_1X_3+X_3=-2+X_2[(5+X_1(3+X_3(4))+(1-X_1)(X_3(2)))+(1-X_2)[X_1(X_3(-1))+(1-X_1)(X_3(1))] \quad (15)$$

Next, we transfer function to EVTDD, it is as following expression (16).

$$F(X_1,X_2,X3)=-2+X_2[5+X_1(3+X_3(4))+(X_1-1)(X_3(2))+(X_1-2)(X_3(2))+(X_2-1)[X_1(X_3(-1))+(X_1-1)(X_3(1))+(X_1-1)(X_3(1))]+(X_2-2)[X_1(X_3(-1))+(X_1-1)(X_3(1))+(X_1-2)(X_3(1))] \quad (16)$$

TABLE 1.
THE COMPARISON TABLE FOR EACH DDS

	BDD	EVBDD	Constant input EVBDD	Constant input EVTDD
Number of Node	2^{n+1}-1	2^n	n•(2^n-1)	(n-1) • (2^n-1)

TABLE 2.
THE COMPARISON TABLE BETWEEN THIS PAPER METHOD AND PREVIOUS METHODS

	Edge-valued DD	Applied Function	Regularity & Schematics
Y.T.Lai et al	EVBDD	Boolean Function	▲
S.B.K. Vrudhula et al	EVBDD	Boolean Function	⦿
This paper	MVEVDD	P-Valued n-variable Function	●

Remarks 1 :
EVBDD : Edge-Valued Binary Decision Diagram
EVTDD : Edge-Valued Ternary Decision Diagram
MVEVDD : Multiple-Valued Edge-Valued Decision Diagram

Remarks 2 :
▲ : Available part
⦿ : Some good
● : Good

VI. CONCLUSION

This paper presents a method of constructing the switching function using edge-valued decision diagrams. The proposed method is as following. The edge-valued decision diagram is a new data structure type of decision diagram which is recently used in constructing the digital logic systems based on the graph theory. Next, we apply edge-valued decision diagram to function minimization of digital logic systems. The proposed method has the visible, schematic and regular properties. This paper present a method of constructing P-valued n-variable digital logic systems using multiple-valued edge-valued decision diagram which extended the concept of binary decision diagram and its new data structure edge-valued decision diagram. The proposed method is more regularity compare with former research. For the future, it require the more general type of MVEVDD, also we apply proposed method to any other digital logic design techniques.

REFERENCES

[1] D. Green, Modern Logic Design, Addison-Wesley Publishing Company, 2008.
[2] R. J. McEliece, Finite Fields for Computer Science and Engineers, Kluer Academic Publishers, 2007.
[3] R. Gould, Graph Theory, The Benjamin / Cummings Publishing Company, Inc. 2005.
[4] R. J. Wilson and J. J. Watkins, GRAPH an Introductory Approach, John Wiley & Sons, Inc. 2006.
[5] D. B. West, Introduction to Graph Theory, Prentice-Hall, 2003.
[6] S. B. Aker,"Binary Decision Diagrams," IEEE Trans. Comput., vol.C-27, no.6, pp.509-516, Jun. 1978.
[7] R. E. Bryant,"Grpah-Based Algorithms for Boolean Function manipulations," IEEE Trans. Comput., vol.C-35, no.8, pp.677-691, Aug. 1986.
[8] Yung-Te Lai and Sarma Satry,"Edge-Valued Binary Decision Diagrams for Multi-Level Hierarchical Verification" 29th ACM/IEEE Design Automation Conference, pp.608-613, 1992.
[9] Yung-Te Lai, M. Pedram and S. B. K. Vrudhula,"Formal Verification using Edge-Valued Binary Decision Diagrams," IEEE Trans. On Computers, vol.45, no.2, Feb. 1996.
[10] D. M. Miller,"Multiple-Valud Logic Design Tools," IEEE Proc. of ISMVL, Sacramento, California, pp.2-11, may. 1993
[11] Davis C. Rine, Computer Science and Multiple-Valued Logic Theory and Applications, North-Holland Publishing Company, 1977.
[12] G. Epstein, Multiple-Valued Logic : an Introduction, Institute of physics publishing , 1993.

Chun-Myoung Park received the B.S, M.S and Ph.D. degree from electronic engineering, Inha University, Incheon, Korea, in 1983, 1986 and 1994 respectively. He joined the faculty at Chungju National University at 1995, where he is currently a professor in the Dept. of computer engineering. From 2002 to 2003, he was a visiting scholar at UCI in USA. His research interests include the next generation digital logic systems & computer architecture, embedded computer systems, ubiquitous computing systems, e(U)-Learning , IT application etc.