Fault Detection in Combinational Circuits Using Boolean Satisfiability

Ljubomir Cvetković
Teacher Training College
22000 Sremska Mitrovica, Serbia

Darko Dražić
Teacher Training College
22000 Sremska Mitrovica, Serbia

Abstract— **In this paper we propose a procedure for detection of single and multiple stuck-at faults in combinational circuits. Faults in a combinational circuit are described by the stuck-at fault model. Test signal propagation in a combinational circuit is described by the use of the logical operation $\wedge$. The problem of automatic test generation is transformed into the SAT-problem. The part of the combinational circuit used for test signal propagation is described by a conjunctive normal form with additional conditions defined by some previously introduced logical relations. On the basis of the proposed procedure all test vectors are obtained. The procedure is more efficient than the well-known Boolean difference method, since, when activating a path, only the part of combinational circuit to which the path belongs is taken into consideration.**

Keywords: Combinational Circuits, Boolean Satisfiability, Conjunctive Normal Form, Fault Detection, Stuck-at faults, Test Vector.

I. INTRODUCTION

A combinational circuit can be represented by several types of Boolean functions or formulas. The combinational circuit is frequently described by a Boolean function given in a Conjunctive Normal Form (CNF) because this form is suitable for calculations. CNF formulas are easy for manipulation and can be easily obtained from combinational circuits [1]. The size of CNF grows linearly with circuit size.

The Boolean Satisfiability Problem (SAT) is reduced to checking whether there exist variables in the formula, represented by CNF, for which the formula is true. It is proved that SAT is NP-complete [2]. A class of solvers based on Davis-Logemann-Loveland algorithms (DPLL) is well-known [3]. This algorithm is a branching search algorithm with backtracking and forms the framework for the development of new SAT solvers, for example see [4], [5]. It finds solution if and only if the CNF formula is satisfiable. For Automatic Test Pattern Generation (ATPG) in combinational circuits with stuck-at faults algorithms D algorithm [6], PODEM [7] and FAN [8] are traditionally used. These algorithms are based on the search for a test pattern which ensures the detection of the desired fault. The problem of ATPG is transformed into SAT problem [9]. ATPG using Boolean satisfiability was proposed in 1990s, with using Boolean formula expressed in CNF, without the analysis of circuit structure [10], [11]. For some combinational circuits structural algorithms do not give satisfiable solutions, especially concerning the time needed for finding solutions. In that way SAT-based ATPG algorithms are complements to structural testing methods [12]. In [13] tool PASSAT is described. Its purpose is the detection of stuck-at faults in combinational circuits. There is a special class of algorithms where the same set of SAT instances is repeatedly processed with different constraints [14]. A flaw of these SAT-based algorithms is that they are rather time-consuming. The automatic test pattern generator TIGUAN (Thread-parallel Integrated test pattern Generator Utilizing satisfiability) is based on a thread-parallel SAT solver [15]. TIGUAN can detect stuck-at faults in large circuits which contain up to several million gates.

By fault detection procedure we describe only a part of combinational circuit by CNF. At first we write logical relations for test signal propagation, and they uniquely define additional constraints for solving CNF. More details about logical relations which describe test signal propagation can be seen in [16] and [17].

II. PRELIMINARIES

A conjunctive normal form of a Boolean function is a conjunction of clauses, i.e. elementary disjunctions. Boolean variables appear in clauses without or with negations. Clauses may have different numbers of variables. For example, the function $f = (x_1 + x_3 + x_4) \cdot (\overline{x_1} + x_2 + x_3) \cdot (x_1 + x_3 + \overline{x_4})$ is represented as a CNF.

Basic logic elements can be represented by CNF. Table I gives conjunctive normal forms for basic logical elements.

A combinational circuit can be represented by a unique conjunctive normal form. One should label uniquely each line in the combinational circuit. For each element g in combinational circuit C we find a conjunctive normal form according to Table I. CNF for the entire combinational circuit is represented as a conjunction of conjunctive normal forms of elements $g_1, g_2, ..., g_n \in C$:

$$K = \prod_{i=1}^{n} Kg_i \tag{1}$$

TABLE I. CNF FORMULAS FOR SIMPLE GATES.

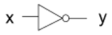
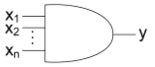
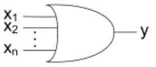
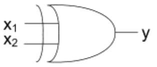
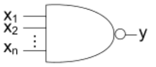
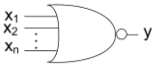
Gate	Function	CNF
	$y=\overline{x}$	$(x+y)\cdot(\overline{x}+\overline{y})$
	$y=x_1\cdot x_2\cdot\ldots\cdot x_n$	$(x_1+\overline{y})\cdot(x_2+\overline{y})\cdot\ldots\cdot(x_n+\overline{y})\cdot$ $(\overline{x}_1+\ \overline{x}_2+\ldots+\overline{x}_n+y)$
	$y=x_1+x_2+\ldots+x_n$	$(\overline{x}_1+y)(\overline{x}_2+y)\cdot\ldots\cdot(\overline{x}_n+y)\cdot$ $(x_1\ +x_2+\ldots+x_n+\overline{y})$
	$y=x_1\ \oplus\ x_2$	$(\overline{x}_1+x_2+y)\cdot(x_1+\overline{x}_2+y)\cdot$ $(x_1+x_2+\overline{y})\cdot(\overline{x}_1+\overline{x}_2+\overline{y})$
	$y=\overline{x_1\cdot x_2\cdot\ \ldots\cdot x_n}$	$(x_1+y)\cdot(x_2+y)\cdot\ldots\cdot(x_n+y)\cdot$ $(\overline{x}_1\ +\overline{x}_2+\ldots+\overline{x}_n+\overline{y})$
	$y=\ \overline{x_1+x_2+\ldots+x_n}$	$(\overline{x}_1+\overline{y})\cdot(\overline{x}_2+\overline{y})\cdot\ldots\cdot(\overline{x}_n+\overline{y})\cdot$ $(x_1+x_2+\ldots+x_n+y)$

Fig. 1 shows a combinational circuit by which test vectors for a single fault are determined.

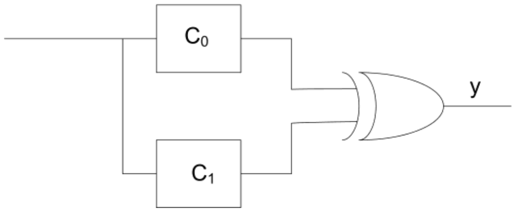


Figure 1. Boolean difference.

Block C_0 represents a fault-free combinational circuit, while the circuit with a fault is represented by block C_1. A fault free combinational circuit has an arbitrary number of input lines and only one output line. On output line y test vectors generate the value 1 in the case when the fault F_1 appears in the circuit C_1 For combinational circuit on Fig. 1 we find CNF and in this way the problem of automatic generation of tests ATPG is transformed into problem SAT. We apply a SAT-solver to solve CNF. Thus, test vectors are obtained by determining conditions under which CNF has value 1. A solution exists if each clause is satisfied, i.e. if for each clause at least one variable has value 1. A formula is contradictory if there do not exist variable values which satisfy the formula and in this case test vectors do not exist.

III. ACTIVE PATHS METHOD

The procedure for the single stuck-at fault detection in a combinational circuit is based on the following two conditions. First, one should generate the faulty line a test signal whose value is opposite to the fault signal. Secondly, one should enable the test signal propagation to the output line of the combinational circuit, which means that the path from the faulty line to the output line must be active. Such an approach to fault detection is known as the path sensitization method.

Consider any line h from which a unique path leads to the output of the combinational circuit. Let h/0 and h/1 be signals of the fault 0 and 1 respectively. We have already mentioned that a test vector detects a single fault if the output signals of the fault free and faulty circuit are complementary. Obviously, the test vector, which detects the fault h/0, generates at line h a signal of value 1 in a fault free circuit, and vice versa. In addition, at each line of a sensitive path in a fault free circuit the test vector generates a signal complementary to the fault signal. A signal which at a line has value 0 in a fault free circuit and value 1 in an faulty circuit, and vice versa, is called a test signal. This signal is transferred from the faulty line to the output of the combinational circuit along a path. This path is called an active or a sensitive path. In order to detect a single fault one should find one or several input vectors of the combinational circuit in such a way that the path between the fault line and the output line of the circuit is active.

The values of test signals are denoted by D and C. Test signal D has value 1 in a fault-free combinational circuit, and it has value 0 in a faulty combinational circuit. Test signal C has value 0 in a fault-free combinational circuit, and it has value 1 in a faulty combinational circuit. Test signals have values from the set {0,1,C,D} where C,D∈{0,1}. The set {0,1,C,D} and operations +, • and ‾ represent a Boolean algebra if the operations are defined by the following tables:

TABLE II. OPERATION "+"

+	0	1	C	D
0	0	1	C	D
1	1	1	1	1
C	C	1	C	1
D	D	1	1	D

TABLE III. OPERATION "•"

•	0	1	C	D
0	0	0	0	0
1	0	1	C	D
C	0	C	C	0
D	0	D	0	D

TABLE IV. OPERATION "‾"

–	0	1	C	D
	1	0	D	C

Consider now logic gate G with n input lines and output line v. Gate G can be OR, NOR, AND or NAND. Suppose that test signals of value D or C appear on input lines $u_1,u_2,\ldots,u_p$ of gate G, while propagation signal values 0 or 1 appear on the remaining lines $z_1,z_2,\ldots,z_q$ (p+q=n).

The test signal propagation to the output line v can occur in the case when test signals of the same value (D or C) appear on input lines $u_1,u_2,\ldots,u_p,$ while signal values on the remaining lines $z_1,z_2,\ldots,z_q$ must have 0 for gates OR and NOR, and value 1 for gates AND and NAND.

Consider the situation presented in Fig. 2. Test signal value D appears on r input lines of the gate G, while value C appears on s input lines. Lines $z_1,z_2,\ldots,z_q$ have propagation signal values for gate G.

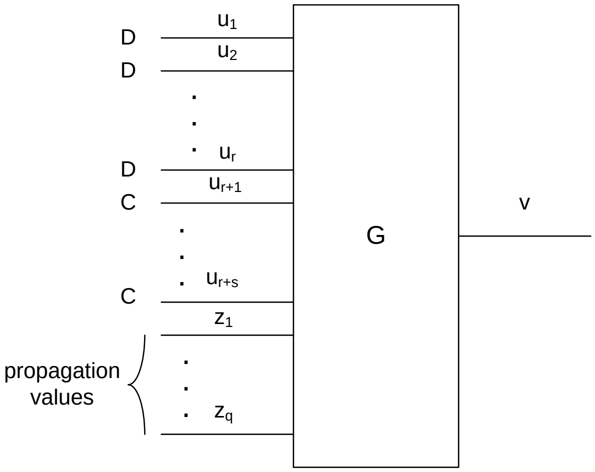


Figure 2. Test signals with distinct values.

Suppose that gate G on Fig.2 is an OR gate.

Test signal of value D is propagated to the output line v according to the relation

$$(u_1=D)\wedge\ldots\wedge(u_r=D)\wedge(u_{r+1}=0)\wedge\ldots(u_{r+s}=0)\wedge (z_1=0)\wedge\ldots\wedge(z_q=0)\rightarrow(v=D) \tag{2}$$

Test signal value C appearing s times in relation (2) is replaced by 0. Hence, test signal values D appear r times in relation (2), while the remaining s + q (r + s + q = n) input lines of the gate OR must have values of propagation signals. For the propagation of the test signal of value D to the output line one can write 2^r-2 relations. Test signal of value C is propagated to the output line v according to the Relation

Test signal value C is propgated to the output line v according to the relation

$$(u_1=0)\wedge\ldots\wedge(u_r=0)\wedge(u_{r+1}=C)\wedge\ldots(u_{r+s}=C)\wedge (z_1=0)\wedge\ldots\wedge(z_q=0)\rightarrow(v=C) \tag{3}$$

Analogously to the above analysis, for the propagation of the test signal of value C to the output line v can write 2^s - 2 relations.

The following logical relations hold:

for NOR gate

$$(u_1=D)\wedge\ldots\wedge(u_r=D)\wedge(u_{r+1}=0)\wedge\ldots(u_{r+s}=0)\wedge (z_1=0)\wedge\ldots\wedge(z_q=0)\rightarrow(v=C), \tag{4}$$

$$(u_1=0)\wedge\ldots\wedge(u_r=0)\wedge(u_{r+1}=C)\wedge\ldots(u_{r+s}=C)\wedge (z_1=0)\wedge\ldots\wedge(z_q=0)\rightarrow(v=D). \tag{5}$$

for AND gate

$$(u_1=D)\wedge\ldots\wedge(u_r=D)\wedge(u_{r+1}=1)\wedge\ldots(u_{r+s}=1)\wedge (z_1=1)\wedge\ldots\wedge(z_q=1)\rightarrow(v=D), \tag{6}$$

$$(u_1=1)\wedge\ldots\wedge(u_r=1)\wedge(u_{r+1}=C)\wedge\ldots(u_{r+s}=C)\wedge (z_1=1)\wedge\ldots\wedge(z_q=1)\rightarrow(v=C). \tag{7}$$

for NAND gate

$$(u_1=D)\wedge\ldots\wedge(u_r=D)\wedge(u_{r+1}=1)\wedge\ldots(u_{r+s}=1)\wedge (z_1=1)\wedge\ldots\wedge(z_q=1)\rightarrow(v=C), \tag{8}$$

$$(u_1=1)\wedge\ldots\wedge(u_r=1)\wedge(u_{r+1}=C)\wedge\ldots(u_{r+s}=C)\wedge (z_1=1)\wedge\ldots\wedge(z_q=1)\rightarrow(v=D). \tag{9}$$

IV. FAULT DETECTION

In this paper we propose a procedure for the detection of single stuck-at faults in combinational circuits, in which active path method is used. The problem of ATPG is transformed into SAT problem. The path for test signal propagation is described by CNF, with taking into consideration additional constraints with the aim of path activation. The propagation of test signals in combinational circuit is described by logical relations, whereby logical operation $\wedge$ is used. On the basis of logical relations we determine additional constraints which are needed for solving CNF. Logical relations are written for a part of combinational circuit, which is described by CNF. It represents an advantage of the proposed procedure.

Example 1. Determine test vectors detecting a single fault h/1 in the combinational circuit in Fig. 3.

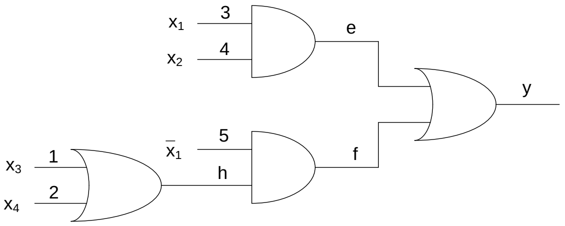


Figure 3. Combinational circuit

We describe a combinational circuit by CNF. On the basis of Table I we have:

$$(\bar{x}_3+h)(\bar{x}_4+h)(x_3+x_4+\bar{h})$$
$$\cdot(x_1+\bar{e})(x_2+\bar{e})(\bar{x}_1+\bar{x}_2+e)$$
$$\cdot(\bar{x}_1+\bar{f})(h+\bar{f})(x_1+\bar{h}+f)$$
$$\cdot(\bar{e}+y)(\bar{f}+g)(e+f+\bar{y}) \quad . \tag{10}$$

The signal having the value C should be generated on the faulty line h. The only way from the faulty line to output line is (h, f, g). Logical relations defining conditions for test signal propagation to the output line i of the combinational circuit are:

$$(h=C)\wedge(5=1)\rightarrow(f=C) \tag{11}$$

$$(f=C)\wedge(e=0)\rightarrow(y=C) \tag{12}$$

On the basis of relations (11) and (12) we determine additional constraints: h=C, 5=1, f=C, e=0 and y=C. We put these constraints into CNF (10):

$$\begin{aligned}&(\bar{x}_3 + C)(\bar{x}_4 + C)(x_3 + x_4 + \bar{C})\\ &\cdot(x_1 + 1)(x_2 + 1)(\bar{x}_1 + \bar{x}_2 + 0)\\ &\cdot(\bar{x}_1 + \bar{C})(C + \bar{C})(x_1 + \bar{C} + C)\\ &\cdot(1 + C)(\bar{C} + C)(0 + C + \bar{C})\,. \qquad (13)\end{aligned}$$

In this way we get:

$$(\bar{x}_1 + \bar{x}_2)(\bar{x}_3)(\bar{x}_4)\,.$$

The above CNF is satisfied by the following values for x1, x_2, x_3 and x_4: 0000, 0100 and 1000. Combination 1000 is rejected, because x1 cannot have value. In the relation (11) there is the following condition: (5=1). Accordingly, the set of test vectors which detect fault h/1 is {0000, 0100}.

This procedure enables the determination of all test vectors for a single stuck-at fault. Comparing with the method of Boolean difference, the proposed procedure is less complex because the number of clauses in CNF is significantly reduced.

In the next example a double fault is located on two paths in the combinational circuit, the two paths having some lines in common.

Example 2. Let us detect the double fault {1/0, 3/0} in combinational circuit on Fig. 4.

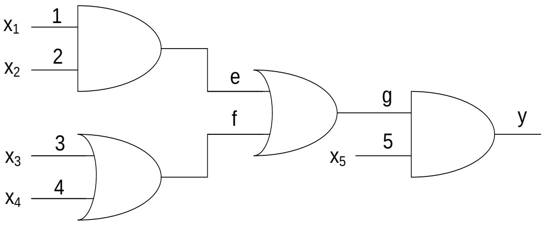


Figure 4. A double fault in a combinational circuit.

We describe a combinational circuit by CNF. On the basis of Table I we have:

$$\begin{aligned}&(x_1 + \bar{e})(x_2 + \bar{e})(\bar{x}_1 + \bar{x}_2 + e)\\ &\cdot(\bar{x}_3 + f)(\bar{x}_4 + f)(x_3 + x_4 + \bar{f})\\ &\cdot(\bar{e} + g)(\bar{f} + g)(e + f + \bar{g})\\ &\cdot(g + \bar{y})(x_5 + \bar{y})(\bar{g} + \bar{x}_5 + y) \quad . \qquad (14)\end{aligned}$$

To lines 1 and 3 we associate the test signals, having values opposite to the fault signals. We have relations:

$$(1 = D) \wedge (2 = 1) \rightarrow (e = D) \qquad (15)$$

$$(3 = D) \wedge (4 = 0) \rightarrow (f = D) \qquad (16)$$

$$(e = D) \wedge (f = D) \rightarrow (g = D) \qquad (17)$$

$$(g = D) \wedge (5 = 1) \rightarrow (y = D) \qquad (18)$$

Relations (15) – (18) define additional constraints, which enable solving CNF.

$$\begin{aligned}&(x_1 + \bar{D})(x_2 + \bar{D})(\bar{x}_1 + \bar{x}_2 + D)\\ &\cdot(\bar{x}_3 + D)(\bar{x}_4 + D)(x_3 + x_4 + \bar{D})\\ &\cdot(\bar{D} + D)(\bar{D} + D)(D + D + \bar{D})\\ &\cdot(D + \bar{D})(x_5 + \bar{D})(\bar{D} + \bar{x}_5 + D) \quad . \qquad (19)\end{aligned}$$

In this way we get x_1=1, x_2=1, x_3=1, x_4=0 and x_5=1. Accordingly, the set of test vectors {11101} detect double fault {1/0, 3/0}.

For determining the rest of test vectors we use the following logical relations:

$$(e = D) \wedge (f = 0) \rightarrow (g = D) \qquad (20)$$

$$(e = 0) \wedge (f = D) \rightarrow (g = D) \qquad (21)$$

On the basis of the above relations we get constraint values which are included in (19).

On the basis of logical relations we determine additional constraints which are needed for solving CNF. Logical relations are written only for a part of a combinational circuit, which is described by CNF, and it represents an advantage of the proposed procedure. Nowadays efficient SAT solvers are on disposal for big CNFs. SAT solvers are constantly improved, so that nowadays they are able to solve formulas with many thousands of variables and millions of clauses. With the aim of efficient solving CNF formula a greater number of heuristic methods is also proposed.

V. CONCLUSION

In this paper we propose a procedure for the single stuck-at fault detection in combinational circuits. With the aim of fault detection a path of test signal propagation is determined, from the faulty line to the output line of the combinational circuit. In applying the proposed procedure we go only forward, thus avoiding the problem of unsuccessful steps, which appear in structural testing methods. Writing logical relations for the test signal propagation is easier, since one uses only operation "∧". For the choosen path the Boolean expression in Conjunctive Normal Form is written. On the basis of logical relations we get additional constraints, which are needed for solving CNF. In that way test determination is transformed into the determination of the variable values in the Boolean formula, and only of those which satisfy the formula. Comparing with the method of Boolean difference, the proposed procedure is less complex because the number of clauses in CNF is reduced.

REFERENCES

[1] T. Larrabee, "Test Pattern Generation Using Boolean Satisfiability", IEEE Trans. on Computer-Aided Design, vol. 11, no. 1, pp. 4-15, January 1992.

[2] S. A. Cook, "The complexity of theorem proving procedures," in 3. ACM Symposium on Theory of Computing, 1971, pp. 151–158.

[3] M. Davis, G. Logemann, and D. Loveland, "A machine program for theoremproving", Communications of the ACM, 5(7):394–397, July 1962.

[4] J. Marques-Silva and K. Sakallah, "GRASP: A search algorithm for propositional satisfiability", IEEE Transactions on Computers, 48(5):506–521, 1999.

[5] M. Moskewicz, C. Madigan, Y. Zhao, L. Zhang, and S. Malik, "Chaff: Engineering an efficient SAT solver", In Proceedings of the Design Automation Conference, pages 530–535, 2001.

[6] J. P. Roth, "Diagnosis of automata failures: a calculus and a method", IBM J. Res. Dev., 10(4):278–291, 1966.

[7] P. Goel, "An implicit enumeration algorithm to generate tests for combinational logic circuits", IEEE Trans. On Computers, 1981, pp. 221-222.

[8] H. Fujiwara and T. Shimono, "On the acceleration of test generation algorithms," IEEE Trans. Comput., C-32(12) , 1983, pp. 1137-1144.

[9] A. Biere, "PicoSAT essentials, "Jour. of Satisfiability, Boolean Modeling and Computation", vol. 4, no. 2–4, pp. 75–97, 2008.

[10] P. Stephan, R. K. Brayton, and A. L. Sangiovanni-Vincentelli, "Combinational test generation using satisfiability", IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol. 15, no. 9, pp. 1167–1176, 1996.

[11] J. P. Marques-Silva and K. A. Sakallah, "Robust search algorithms for test pattern generation", in Int'l Symp. on Fault-Tolerant Computing, 1997, pp. 152–157.

[12] R. Drechsler, S. Eggersglüß, G. Fey, A. Glowatz, F. Hapke, J. Schloeffel, and D. Tille, "On acceleration of SAT-based ATPG for industrial designs", IEEE Trans. on Computer-Aided Design for Circuits and Systems, vol. 27, no. 7, pp. 1329–1333, 2008.

[13] J. Shi, G. Fey, R. Drechsler, A. Glowatz, F. Hapke, and J. Schloeffel, "PASSAT: Efficient SAT-based test pattern generation", in IEEE Annual Symp. on VLSI, 2005, pp. 212–217.

[14] Balcárek, J., Fišer, P., Schmidt, J. "Test Patterns Compression Technique Based on a Dedicated SAT-based ATPG", Proc. of 13th Euromicro Conference on Digital Systems Design (DSD'10), Lille (France), 1.-3.9.2010, pp. 805-808.

[15] A. Czutro, I. Polian, M. Lewis, P. Engelke, S. M. Reddy, and B. Becker, "TIGUAN: Thread-parallel Integrated test pattern Generator Utilizing satisfiability Analysis", In Int'l Conf. on VLSI Design, pages 227–232, 2009.

[16] Lj.Cvetković, "An addition to the methods of test determination for fault detection in combinational circuits", Acta Cybernetica, Vol 16, No 4, pp. 545-566, Department of Informatics, University of Szeged, 2004.

[17] Lj. Cvetković, D. Dražić, "Determining Covers in Combinational Circuits", International Journal of Computer Science Issues, Vol.8, Issue 3, pp. 59-63, 2011.

AUTHORS PROFILE

Ljubomir Cvetkovic: received the PhD degree from the University of Belgrade (Faculty of Electrical Engineering) in 2005. He is currently a professor in Teacher Training College in Sremska Mitrovica in Serbia. His major research interests include Digital VLSI architecture, Fault-tolerance and Fault detection. He has published about 20 publications in journals and international conferences and has written 3 books.

Darko Drazic: received the BSc degree from the University of Belgrade (Faculty of Organizational Sciences) in 2005. He is currently a PhD student on software engineering at Faculty of Organizational Sciences. His research interests include Computer architecture, Information system, Audio and video processing.