"Some Easily Testable and Diagnosable Network Realizations"

Masaru YAMAMOTO[+], Takayuki NAKAJIMA[++] and Yutaka ASHIDA[+++]

[+] *Department of Management Engineering*
[++] *I.B.M. Japan*
[+++] *Kobe Steel Co.*

(Received September 3, 1975)

Abstract—This paper considers some easily testable and diagnosable network realizations. First, we propose three types of network realizations by modifying Reddy's easily testable network. Second, we discuss each type of the network realization in detail and give the syntheses algorithms to obtain the optimum network realization for a given logical function. We also show that these network realizations are superior to other realizations in the fault detection and fault diagnoses. Last, we compare these network realizations with Reddy's easily testable network.

1. INTRODUCTION

In recent years, much efforts have gone into the developments of high reliability and the fault diagnoses techniques for the digital systems, in accordance with the achievement of the large scale on-line systems. Specially on the latter problem, as the systems get larger and more complicated by the influence of the Integrated Circuits, the usual diagnoses techniques[1)~3)] have become impractical. Therefore, in the previous paper[4)~6)] we had proposed some network realizations which are suitable for the fault diagnoses, as one solution. On the contrary, *S.M. Reddy*[7)] had investigated the desirable properties of easily testable networks and also proposed a realization for an arbitrary logical function. On the other hand, *Hayes*[8)] had indicated that *observability* and *controllability* of the networks are the two important properties for their testability and diagnosability. From these view points, in this paper we shall propose three types of *easily testable and diagnosable network realizations* (Ⅰ), (Ⅱ) *and* (Ⅲ).[9)~11)] As the same with (7), the collector parts of these network realizations are all constructed in the cascade of Exclusive OR gates. These realizations have many of desirable properties not only for the fault detection but also the fault location. In *section 2*, we shall give Reddy's network realization and some preparations. In *section 3, 4 and 5*, we shall present the easily testable and diagnosable network realizations (Ⅰ), (Ⅱ) *and* (Ⅲ) respectively and also give their properties, fault detecting tests and fault diagnoses procedures in detail. Last, in *section 6*, we shall compare these network realizations and give some conclusions.

2. *REDDY'S NETWORK REALIZATION AND PREPARATIONS*

Reddy[7)] has investigated the desirable properties of easily testable network and given an easily testable network realization of *Fig. 1* which is composed of

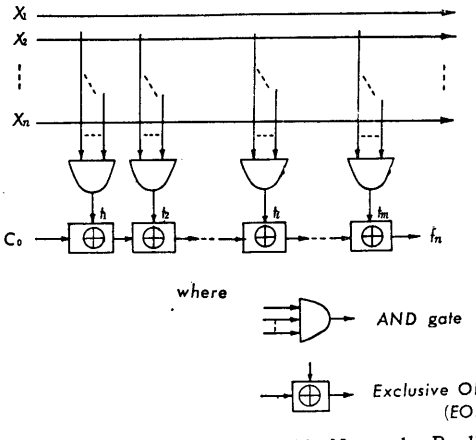

Fig. 1 Reddy's Easily Testable Network Realization

AND gates and *Exclusive OR* gates (from now on, represented as *EOR gates*). Then an n-variable logical function f_n can be expressed in a unique *Reed-Muller canonical form* [12] as given in (*1*);

$$f_n=c_o\oplus t_1\oplus t_2\oplus\cdots\oplus t_i\oplus\cdots\oplus t_m \quad (1)$$

where the symbol c_o is understood to be the binary constant *0* or *1*, and each term t_i is a product of some input variables without complements.

For example, a *5*-variable logical function f_5 can be expressed

$$f_5=1\oplus x_1x_3\oplus x_1x_2x_3\oplus x_1x_2x_4\oplus x_3x_4\oplus x_1x_3x_4\oplus x_2x_3\oplus x_2x_3x_4\oplus x_1x_2x_3x_4\oplus x_1x_5\oplus x_2x_5\oplus x_1x_2x_5\oplus x_3x_5\oplus x_3x_4x_5\oplus x_1x_3x_4x_5\oplus x_2x_3x_4x_5\oplus x_1x_2x_3x_4x_5 \quad (2)$$

In Reddy's network of *Fig. 1*, each AND gate can realize a term of (*1*) and the cascade of EOR gates is used to collect all terms by operation of *logical ring-sum*. Then the fault detection of Reddy's network was investigated in detail. If only permanent *stuck-at-0* (*s-a-0*) or *stuck-at-1* (*s-a-1*) faults occur in a single AND gate or a single EOR gate is faulty, the following results were derived on the fault detecting test sets for Reddy's networks; *1*) only $(n+4)$ *tests*, independent of the function being realized, are required if the primary inputs are fault-free. *2*) only $(n+4+2n_e)$ *tests* are required, if the primary inputs can be faulty, where n_e is the number of variables appearing in even number of product terms in the Reed-Muller canonical form of the function. It is noted that one of the major drawbacks of Reddy's network is the number of *logic levels, i.e.*, it may require a large number of logic levels in the collector of the network. Furthermore, Reddy's network is not suitable for the fault location. Therefore in the next sections, we propose three types of network realizations which are suitable not only for the fault detection but also for the fault location, by modifying Reddy's network.

Now we give some definitions and assumptions which are used in the following sections.

[*Definition 1*] A logical product of some literals is called as *a term*.

[*Definition 2*] t_i denotes a term containing only uncomplemented literals

[*Definition 3*] g_i denotes *a term* which may contain complemented and uncomplemented literals

[*Definition 4*] For a logical function f, $f(m_i)$ denotes the value of the function for a minterm m_i.

[*Definition 5*] The length $l(t_i)$ of a term t_i means the number of literale in it.

[*Definition 6*] For a logical function f_n, define the set of minterms m_i of $f_n(m_i)=1$ as $ON[f_n]$, *i.e.*, $ON[f_n]=\{m_i|f_n(m_i)=1,\ m_i$ *is a minterm.*$\}$.

[*Definition 7*] Let $|A|$ denote the number of elements in a set A.

[*Assumptions*] In this paper we set limitation to the fault as follows:

1) *range of fault*; stuck-at-0 and stuck-at-1 fault in AND gates and any faults in EOR gates.

2) *number of faults*; single fault, *i.e.*, there exists at most one fault in the network at a time.

3) *type of faults*; stationary fault, not intermittent fault.

3. EASILY DIAGNOSABLE NETWORK REALIZATION (*I*)

In this section, to improve further the controllability and diagnosability of Reddy's network, we insert some EOR gates and external inputs c^i_1 and c^i_2 (where $i=1, 2, \cdots, m$) as the control logic and inputs. As the result, *an easily diagnosable network realization* (*I*) can be obtained in *Fig. 2*. It is shown that each internal output function t'_i and h_i satisfy the following desirable properties for the controllability of *the network* (*I*).

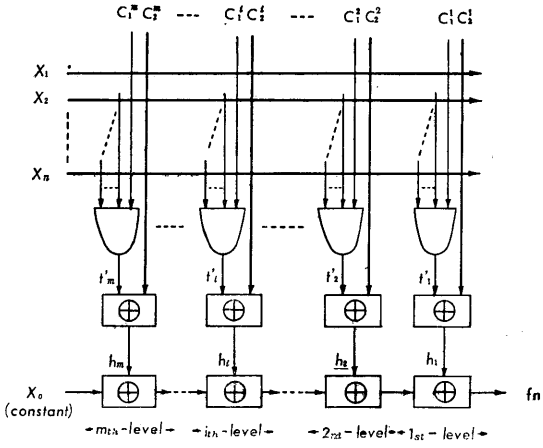

Fig. 2 Easily Diagnosable Network Realization (I)

Property 1; $t'_i=t_i\cdot c^i_1$, *for all* t_i (3)

Property 2; $h_i=t'_i\oplus c^i_2=(t_i\cdot c^i_1)\oplus c^i_2$, for all t_i (4)

Property 3; *Set* $c^i_1=1$ *and* $c^i_2=0$, *then* $h_i=t_i$ (5)

Property 4; *Set* $c^i_1=1$ *and* $c^i_2=1$, *then* $h^i=\bar{t}^i$ (6)

Property 5; *Set* $c^i_1=0$ *and* $c^i_2=0$, *then* $h_i=0$ (7)

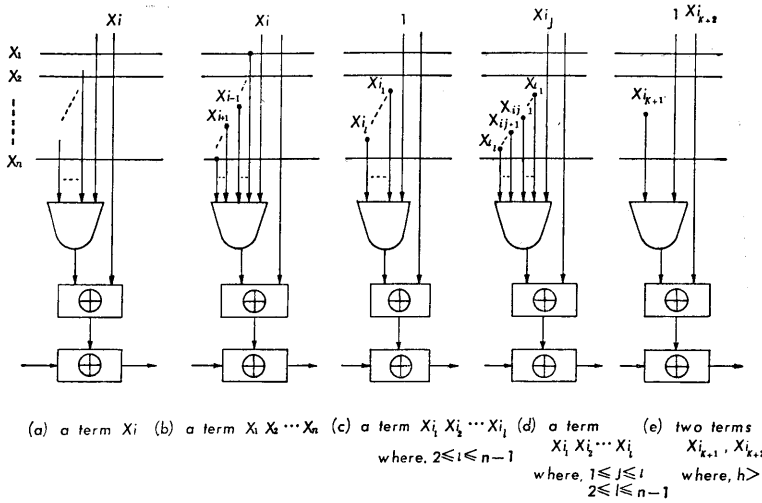

Fig. 3 Restrictions on Realizations of Each Term

Property 6; *Set* $c^i_1=0$ *and* $c^i_2=1$, *then* $h_i=1$ (8)

Obviously from (3), if we apply the logical constant *1* to all primary inputs $x_1, x_2, \cdots, x_n$, then each output function t'_i gets equal to the value of c^i_1. Hence, by applying the logical constant *0* or *1* properly to each control input c^i_1, we can set independently each t'_i to be an arbitrary constant *0* or *1*. As (6) holds for any t_i, h_i is said to be completely controllable by a set of lines $\{c^i_1, c^i_2\}$. [8] The properties 6 and 7 show that the output of each h_i can be set to an arbitrary constant, by applying logical constant to c^i_1 and c^i_2. Thus, it is clear that the *Network* (*I*) of *Fig. 2* is superior in the controllability to Reddy's network. As is clear from Property *3*, The network realization for a given logical function by *Fig. 2* is the quite same with [7]. However it is assumed that control inputs c^i_1 and c^i_2 can be used not only in the fault detection but also in realizing a given logical function. If an n-variable logical function f_n is given in a *Reed-Muller canonical form of* (*I*), then we realize the logical function by *the Network* (*I*) of *Fig. 2*, based on the following rules;

〈*Rule 1*〉 : *Construct such a term that is composed of a single variable, i.e., a term with length 1, in the form of Fig. 3*(*a*).

〈*Rule 3*〉 : *Construct a term with length n in the form of Fig. 3*(*b*).

〈*Rule 3*〉 : *Construct a term with length l* (*where* $2 \leqq l \leqq (n-1)$ *in the form of either Fig. 3*(*c*) *or Fig. 3*(*d*).

〈*Rule 4*〉 : *If, in a given logical function, the number h of terms with length 1 is greater than the number k of terms with length greater than or equal 2, then k terms of length 1 are constructed by Rule 1, and the remaining* $(h-k)$ *terms with length 1 are assumed to be realized in the form of Fig. 3*(*e*).

Next, the following theorem can be easily derived on the fault detecting tests set for the Network (I).

[*Theorem 1*] *Only* $(n+2)$ *tests* E_1, E_2 *and* T_i $(i=1, 2, , n)$ *shown in* (9) *are required for the fault detection of the Network* (*I*), *if the primary inputs are fault-free.*

$$
\begin{array}{lllllllll}
 & x_0; & x_1 & x_2 \cdots x_i \cdots x_n; & c^1_1 & c^2_1 \cdots c^m_1; & c^1_2 & c^2_2 \cdot c^j_2 \cdot c^m_2 & \\
E_1=[0; & 1 & 1\cdots1\cdots1; & 0 & 0\cdots0; & 0 & 0\cdots0\cdots0] & \\
E_2=[1; & 1 & 1\cdots1\cdots1; & 1 & 1\cdots1; & 1 & 1\cdots1\cdots1] & \\
T_1=[0; & 0 & 1\cdots1\cdots1; & 1 & 1\cdots1; & & &] \quad (9)\\
T_2=[1; & 1 & 0\ \ 1\cdots1; & 1 & 1\cdots1; & & &] \\
\vdots & & & & & & & \\
T_i=[-; & 1 & 1\cdots0\cdots1; & 1 & 1\cdots1; & \cdots e_{ij} \cdots & &] \\
\vdots & & & & & & & \\
T_n=[-; & 1 & 1\cdots1\cdots0; & 1 & 1\cdots1; & & &]
\end{array}
$$

where

$$
e_{ij}=\begin{cases} 0; & \text{if } t'_j=1, \text{ for the input pattern of } T_i \\ 1; & \text{if } t'_j=0, \qquad\qquad \text{〃} \end{cases} \quad (10)
$$

(The proof is omitted.)

Even if the primary inputs can be faulty, we can realize *the Network* (*I*) for which only $(n+2)$ *tests* of (9) or a few additional tests plus the $(n+2)$ *tests*, are required. The algorithm for realizing such a network were investigated in detail.[10] It is further shown that *the Network* (*I*) can be diagnosed easily and effectively. The fault locating procedures for *the Network* (*I*) by *the adaptive experiment* was studied

in [10]. However, an m-level *Network* (*I*) which realizes an n-variable logical function, needs $(n+2m+2)$ external terminals in all. On the contrary, Reddy's network needs only $(n+2)$ external terminals. In practice, the amount of terminals and delays of networks, are strictly limmited in the integrated circuits technology. Therefore, the network realization (I) will be unsuitable for the logical functions that possess the large value m in the Reed-Muller canonical form.

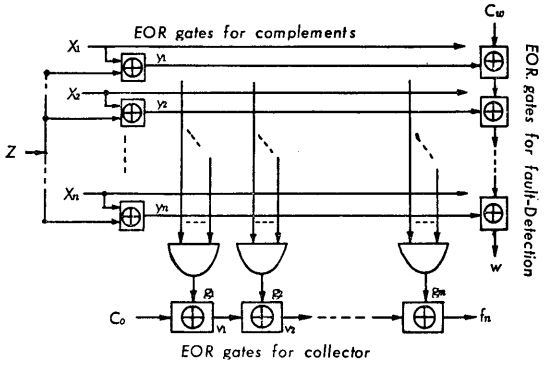

Fig. 4 Easily Testable Network Realization (Ⅱ); n is an odd number.

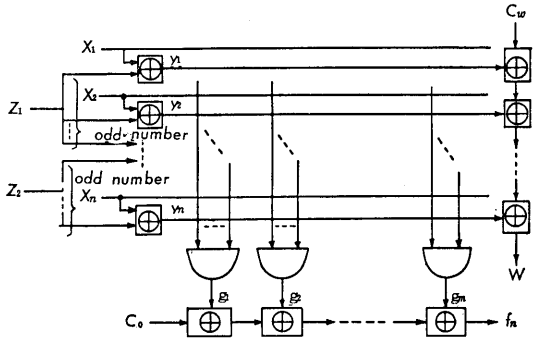

Fig. 5 Easily Testable Network Realization (Ⅱ); n is an even number.

4. *EASILY TESTABLE NETWORK REALIZATION*(Ⅱ)

In this section, modification of Reddy's easily testable network is given in *Fig. 4*, to reduce the number of logic levels in the collector of the network. In Reddy's network and *the Network* (Ⅰ), only uncomplemented input variables can be used. On the contrary, the modified network realization (called as *the Network* (Ⅱ)) allows each AND gate to be applied not only input variables but also their complements. We also annex an Exclusive OR cascade to the Network in order to detect all single faults at the primary input buses easily. Furthermore we will combine all input lines of the EOR gates for complements, to reduce the pin number of the network. Then the next two types of realizations are considered; *1*) the network realization of *Fig. 4*, if the number of input variables n is an odd number. *2*) the network realization of *Fig. 5*, if n is an even number. The requirement that two types of realizaitons are considered is due to the fact that EOR gates are *modulo 2 adder*, which imply that an even number of changes at the inputs to the EOR gates cancel out and do not sensitize the changes to the output w. In *the Network* (Ⅱ), let $z=1$ (or, $(z_1, z_2)=(1, 1)$), then we can obtain as $(y_1, y_2, \ y_n)=(\bar{x}_1, \bar{x}_2, .., \bar{x}_n)$. On the contrary, if we set as $z=0$ (or, $(z_1, z_2)=(0, 0)$), then we can obtain as $(y_1, y_2, .., y_n)=(x_1, x_2, .., x_n)$. Therefore both of input variables and their complements can be applied to AND gates of *the Network* (Ⅱ), by setting $z=1$ (or, $(z_1, z_2)=(1, 1)$). Accordingly we can reduce the logical levels of *the Network* (Ⅱ) in comparison with *Reddy's network* and *the Network* (Ⅰ). As obviously known, to obtain *the optimum Network*(Ⅱ) with the least number of logic levels, we must express an n-variable logical function f_n in the extended Reed-Muller form that has the minimum number of terms as

$$f_n=g_1\oplus g_2\oplus\cdots\oplus g_j\oplus\cdots\oplus g_m \qquad (11)$$

where each g_j is the binary constant or a term. However it is difficult and sometimes unpractical to obtain the optimum network realization.[14] Therefore we shall present the synthesis algorithm for obtaining *a near-optimum network realization.* Then the following theorem will be available.

[*Theorem 2*] *For a given n-variable logical function f_n, it can be expressed as* (11), *if and only if there exist an odd number of terms g_j which satisfy $g_j(m_i)=1$ for each minter m_i of $f_n(m_i)=1$, and that there exist an even number of terms g_j which satisfy $g_j(m_i)=1$ for each minter m_i of $f_n(m_i)=0$.*

(*The proof is omitted*)

By using the above *Theorem 2*, we have obtained the synthesis algorithm for a near-optimum network realization. By transforming (*10*), we can gain

$$f_n\oplus g_1\oplus g_2\oplus\cdots\oplus g_j\oplus\cdots\oplus g_m\equiv 0 \qquad (12)$$

where *0* denotes the function whose value is constantly *zero*.

Accordingly the problem of obtaining each

term of (*11*) is essentially to seek terms successively and operate ring-sum to the function untill the resulting function become *zero*. In the synthesis algorithm, we will choose each term g_j in accordance with the following criterion.

[*criterion*] *For a logic function f, we have priority to the term* g_j *whose* $|ON[f \oplus g_j]|$ *is the least number.*

the synthesis algorithm:

Step 1; *For a given n-variable logical function* f_n, *let* $f=f_n$.

Step 2; *Set* $i=1$ *as an initial condition.*

Step 3; *Compute the number of* $|ON[f]|$.

Step 4; *If* $|ON[f]|=0$, *then go to the step 9. Otherwise go to the step 5.*

Step 5; *Seek the integer number* m_i *such that* $2^{mi} \geq |ON[f]| > 2^{mi-1}$

Step 6; *Choose the term* g_i *of all terms whose length is equal or greater than* $(n-m_i)$, *in accordance with the criterion.*

Step 7; *Compute f as* $f=f \oplus g_i$.

Step 8; *Let* $i=i+1$ *and go to the step 3.*

Step 9; *Expand the logical function* f_n *as*

$$f_n=g_1 \oplus g_2 \oplus \cdots \oplus g_x \oplus \cdots \oplus g_{i-1}$$

where each term g_x *was given by the step 6.*

We have also programmed the algorithm and applied the 5-variable logical function of (*2*) in *section 2*. Then we can obtain the following result as

$$f_5=x_3x_5 \oplus x_1x_2x_4 \oplus x_1x_2x_3x_5 \oplus x_1x_2x_3x_4x_5 \quad (13)$$

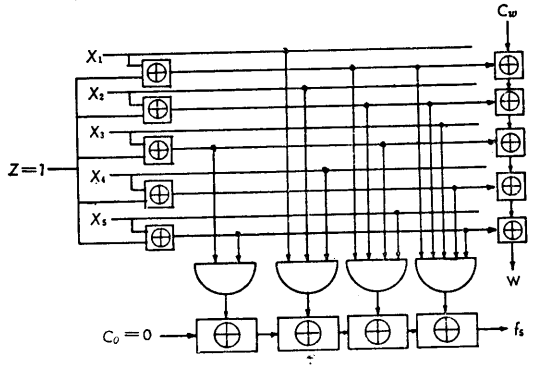

Fig. 6 The Network Realization f_5 of Eq. (2)

Note that we can realize the function by *the Network* (Ⅱ) of *Fig. 6* that needs only 4-logic levels. On the contrary, Reddy's network and *the Network* (Ⅰ) need *16*-logic levels.

Next, on the fault detecting tests for *the Network* (Ⅱ) of *Fig. 4* (or, *Fig. 5*), we can gain the following theorem.

[*Theorem 3*] *It is sufficient to apply at most* $(n+6)$ *test inputs of* (*14*), *to detect the single fault in the Network* (Ⅱ) *that realizes an n-variable logical function.*

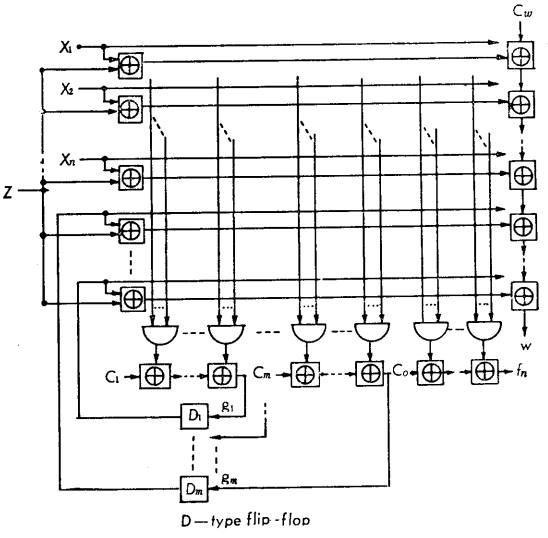

Fig. 7 Easily Testable Sequential Circuit Realization

$$\begin{array}{l} \qquad c_0;\ x_1\ x_2 \cdots x_i \cdots x_n;\ z;\ c_w \\ I_1=[0;\ 1\ \ 1\cdots 1\cdots 1;\ 0;\ 1] \\ I_2=[1;\ 1\ \ 1\cdots 1\cdots 1;\ 0;\ 0] \\ I_3=[0;\ 0\ \ 0\cdots 0\cdots 0;\ 0;\ 1] \\ I_4=[1;\ 0\ \ 0\cdots 0\cdots 0;\ 0;\ 0] \\ I_5=[-;\ 1\ \ 1\cdots 1\cdots 1;\ 1;\ -] \\ I_6=[-;\ 0\ \ 0\cdots 0\cdots 0;\ 1;\ -] \\ T_1=[-;0\ \ 1\cdots 1\cdots 1;\ 0;\ -] \\ T_2=[-;1\ \ 0\cdots 1\cdots 1;\ 0;\ -] \\ T_i=[-;1\ \ 1\cdots 0\cdots 1;\ 0;\ -] \\ T_n=[-;1\ \ 1\cdots 1\cdots 0;\ 0;\ -] \end{array} \quad (14)$$

where - can be taken any value.

(*The proof is given in* (*11*))

It is known that *the Network* (Ⅱ) can be extended to an easily testable sequential circuit of *Fig. 7*. The sequential circuit uses *D-type flip-flops* as memory elements and *the Network* (Ⅱ) as the combinational part. We have also discussed the fault detecting procedures of the sequential circuit in detail (*11*).

5. *EASILY DIAGNOSABLE UNIVERSAL NETWORK* (Ⅲ)

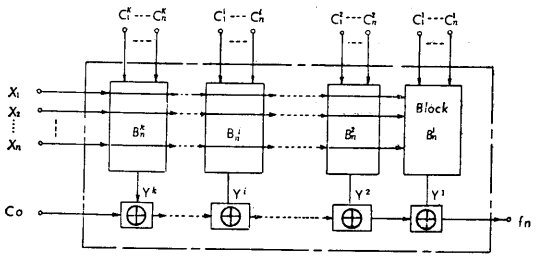

Fig. 8 Easily Diagnosable Network Realization (Ⅲ)

In this section, we propose an easily diagnosable network realization shown in *Fig. 8*, which hereafter we call as *the Network* (Ⅲ). It is constructed by the block part which is the iteration of the blocks, and the collector part which is a one-dimensional cascade of EOR gates. As shown in *Fig. 9*, each block B^i_n possesses n-primary control inputs terminals c^i_1, c^i_2, ···, c^i_n. Because of its geometrical regularity and its control terminals, *the Network* (Ⅲ) increases the controllability. In the case of allowing inputs to the control terminals c^i_j in the form of binary constant, uncomplemented input variables and also complemented input variables each block B^i_n can generate an arbitrary term as an output of y^i. Hence any logical function f_n of (*11*) can be realized by *the Network* (Ⅲ). Then we can obtain the following theorem about the fault detecting tests of *the Network* (Ⅲ).

[*Theorem 4*] *Independently of whether primary inputs could be faulty or not,* $(n+2)$ *tests are sufficient and enough to detect single faults of the Network* (Ⅲ) (*See* [*9*] *and* [*16*])

It is also shown that *the Network* (Ⅲ) is suitable for the fault location. The fault locating procedure is derived to be able to identify whether in the collector part or in which block there exist faults. Moreover a type of fault can be identified with some ambiguity.

Furthermore, by preparing some spare blocks and collectors to *the Network* (Ⅲ), it is possible to repair the faulty network systematically, after the faulty point and the type of fault have been found. It is also noted that *the Network* (Ⅲ) with more than 2^{n-1} blocks can organize an n-variable universal logical network.[17]

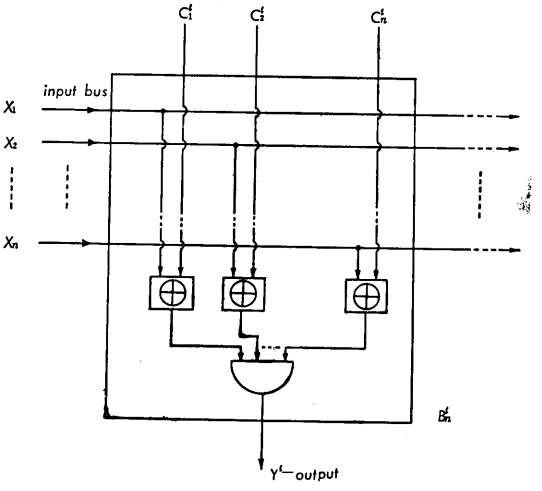

Fig. 9 Circuit Structure of Each Block B^i_n

Table. 1

	Reddy's network	Network (Ⅰ)	Network (Ⅱ)	Network (Ⅲ)
number of fault detecting tests	** 1) $n+4^*$	$n+2$	$n+6$	$n+2$
	*** 2) $n+4+2n_e$	$n+2+\alpha$ $(0\leqq\alpha\leq 2n_e)$	$n+6$	$n+2$
number of extemal input output terminals	$n+2$	**** $n+2m+2$	n : odd, $n+5$ ····· n : even $n+6$	$n+mn+2$
fault location (diagnosability)	*impossible*	*possible*	*impossible*	*possible*
logic levels	*large*	*large*	*small*	*small*
References	[7]	[10]	[11]	[9], [16]

* n denotes the number of the primary input variables of the network.
** In the case that the primary inputs are *fault-free.*
*** In the case that the primary inputs can be faulty.
**** m denotes the number of its logic levels.

6. CONCLUSIONS

In this paper we have presented three types of easily testable and diagnosable network realizations. Specially we have investigated the properties, the fault detecting tests and the synthesis algorithm of each network realization. Last we can obtain the *Table. 1* which summarizes the comparisons of these easily testable networks.

REFERENCES

1) *J.P. Roth, "Diagnosis of automata failures; A calculus and a method", I.B.M. J., pp. 278~291 (July 1966)*

2) *D.M. Armstrong, "On finding a nearly minimum set of fault detecting tests for combinational logic nets", IEEE Trans EC-15, pp. 66-73 (Feb. 1966)*

3) *W.H. Kautz, "Fault testing and diagnosis in combinational digital circuits", IEEE Trans., EC-17, pp.352-366 (April 1968)*

4) *M. Yamamoto, K. Kinoshita, "Modular structured circuits and their fault diagnosis", Trans. I.E.C.E. Japan, 54-C, 9, pp. 804-810 (Nov. 1971)*

5) ——, *"Synthesis of fault diagnosable combinational logic circuits," Trans. I.E.C.E. Japan, 55-D, 1, pp.31-38 (Janu. 1972)*

6) *M. Yamamoto, "Synthesis of fault diagnosable logical circuits," Trans. I.E.C.E. Japan, 55-D, 9, pp.617-625 (Nov. 1972)*

7) *S.M. Reddy, "Easily testable realizations for logic functions," IEEE Trans., C-21, 11, pp.1183-1188 (Nov. 1972)*

8) *J.P. Hayes, "On modifying logic networks to improve their diagnosability," IEEE Trans., C-23, 1, pp.56-62 (Janu. 1974)*

9) *M. Yamamoto, "Synthesis for easily diagnosable logic circuits," Trans. I.E.C.E. Japan, 57-D, 1, pp.8-14 (Janu. 1974)*

10) ——, *"Controllability of networks and its applications to fault diagnoses," Trans. I.E.C.E. Japan, 58-D, 8, pp.450-457 (Aug. 1975)*

11) ——, *"Modifying Reddy's easily testable networks and its application to sequential circuits", Trans. I.E.C.E. Japan, 59-D,3 (Mar.1976)*

12) *D.E. Muller, "Application of boolean algebra to switching design and to error detection," IRE Trans., EC-3, pp.6-12 (Sept. 1954)*

13) *H.Y. Chang, E. Manning and G. Metze, "Fault diagnosis of digital systems," John Wiley and Sons, Inc. (1970)*

14) *A. Mukhopadhyay, G. Schmitz, "Minimization of exclusive OR and logical equivalence switching circuits" IEEE Trans., C-19, 2, pp.132-140 (Feb. 1970)*

15) *K.K. Saluja, S.M. Reddy, "Easily testable two dimensional cellular logic arrays," IEEE Trans., C-23, 11, p. 1204 (Nov. 1974)*

16) *M. Yamamoto, T. Nakajima, "Optimal fault detecting tests set for easily diagnosable circuits," Trans. I.E.C.E. Japan, 57-D, 9, pp.552-553 (Sept. 1974)*

17) *S.S. Yau, C.K. Tang, "Universal logic modules and their applications," IEEE Trans., C-19, 2, pp. 114-149 (Feb. 1970)*