Suresh Palaka, Dr. K. Srinivasa Rao / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 4, July-August 2012, pp.1237-1242

# Fault Tolerant Logic Design For Multiple Faults In Combinational Circuits

**SURESH PALAKA [1], Dr. K. SRINIVASA RAO [2]**

[1] Research Scholar, JNTUH, Hyderabad , Andhra Pradesh
[2] Principal & Professor, Department of ECE, TRRCE, Hyderabad

**Abstract:-**
**A fault detection and analysis of stuck-at faults in multiple fault situation is developed in a combinational circuit with basic Boolean logic gates. Especially Boolean logic gate is proposed for the computation of multiple faults in parallel bus lines. During programming FPGA devices there is a probability of lines getting fused to either stuck at zero or stuck at one faults. Where conventional diagnosing tools analyze stuck zero and stuck one faults only multiple fault cases are not considered. Using eminent logic fault tolerant design problem is solved out in this paper.**

*Keywords:* **Multiple Fault, FPGA, Boolean logic, stuck-at fault**

## I. INTRODUCTION

Fault analysis is currently one of the principal research areas of digital systems theory. One of the most important problems in this area is the generation of fault-detection test sets for combinational logic circuits. Much effort on this problem has resulted in some significant results pertaining to problems characterized by the assumption that only single-fault situations can occur in the circuits under consideration. Recently, however, there has been increasing emphasis given to multiple fault situations. Arguments for the importance of this work have been made quite well elsewhere and will not be repeated here except to say that it is evident that there are many situations in which the possibility of more than one fault existing in the system cannot be ignored.

Fault detection in digital circuits has emerged as an important principal research area in fault-tolerant computing. The increasing complexity and popularity of today's LSI realization of digital circuits have rendered the problem of fault detection, fault analysis, and test generation extremely difficult. Significant results have been achieved in generating tests for combinational logic circuits under the assumption that only the occurrence of single faults of a stuck-at nature is more probable. However, there are cases where multiple fault situations must be considered. Several authors [l], [2] have extended the Boolean difference technique to the multiple fault case in combinational circuits, and [3] in synchronous sequential circuits. In this paper, the analysis discussed in [3] is reviewed and slightly modified to present a new method for deriving the shortest test sequence. Treating the present and next state of the circuit as pseudo-input and pseudo-output vector respectively, a vector Boolean difference technique is utilized to determine the set of input/state pair that will produce a difference in output between fault-free and faulty circuits. Assuming that fault-free and faulty circuits start in the same initial state, they must be driven by applying a sequence of length one input vectors to a state in which a difference in output is evidenced. If a difference in output cannot be achieved immediately based on the difference in next state, the shortest sequence, of length > 1, of input vectors is determined, which when applied will propagate the fault to produce an output difference.

## II. DEFINITIONS AND NOTATION

Consider a Boolean function $f(u_1,\ldots,u_n)$ of the Boolean variables $u_1,\ldots u_n$. Given a combinational logic circuit realizing this Boolean function, the lines of the circuit is labeled using the terminal numbering convention [3] where each line of the circuit is labeled with an integer such that the output of any gate is labeled with an integer greater than that used to label the input terminals of the gate. In addition, the primary inputs will be denoted as $u_1,\ldots,u_n$. The logical value of line j, j = 1,…,m, of the circuit clearly depends on the values of the primary inputs $u_1,\ldots,u_n$, and is been denoted by this relation for all lines other than the primary output line as $X_j(u_1,\ldots,u_n)$. The logical value of the primary output line of the circuit can, of course, be expressed entirely in terms of the primary inputs, but it can also be expressed in terms of the primary inputs and some specified subset of lines of the circuit, say lines $i_1$, $i_2$,.., and $i_p$. This is denoted as $F(u_1,\ldots,u_n, X_{i1},\ldots,X_{ip},)$. For example, consider the combinational logic circuit of Fig. 1. The primary output can be expressed in terms of the primary inputs as

$$F(u_1,u_2,u_3,u_4) = \overline{u_1}\,u_2 + u_2\overline{u_3}u_4 + u_1\overline{u_2}u_3 + u_1\overline{u_4} \quad .$$

However, the primary output based on 11 and 12 can be expressed as

$$F(u_1,u_2,u_3,u_4,X_{11},X_{12}) = \overline{u_1}u_2 + u_2\overline{u_3}u_4 + X_{11}X_{12}$$

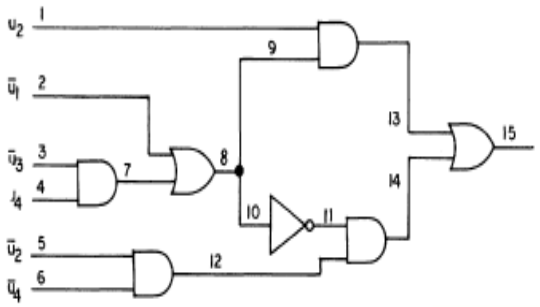

Figure. 1:

$$F(u_1,u_2,u_3,u_4) = \overline{u_1}\,u_2 + u_2\overline{u_3}u_4 + u_1\overline{u_2}u_3 + u_1\overline{u_4}$$

In the following, to simplify the formulas which will be developed, we will abbreviate this expression for the logical value of the primary output line as $F(X_{i2},...X_{ip})$. It will be understood that primary input variables may also be present in the expression. Referring to the circuit of Figure 1,

$$F(X_{11},X_{12}) = \overline{u_1}\,u_2 + u_2\overline{u_3}u_4 + X_{11}X_{12}, \text{ and } F(X_{13},X_{14}) = x_{13} + x_{14}$$

When it is clear that we are concerned with some specific subset of the lines, say $i_1,\ldots,i_p$, and have expressed the primary output accordingly as $F(X_{i2},...,X_{ip},)$, the expression resulting from setting $Xi, = a_{i2},\ldots,X_{ip} = a_{ip}$, where $a_i$ E {0,1}, j = 1,…,p, is denoted as $F(X_i, = a_{il}, X_i, = a_i,) = F(a_{i1},\ldots,a_i,)$. Referring again to the circuit of Figure1,

$$F(X_{11}=1, X_{12}=0) = \overline{u_1}u_2 + u_2\overline{u_3}u_4$$

or if it were clear that we were considering $F(X_{11},X_{12})$, it can be written as

$$F(1,0) = \overline{u_1}u_2 + u_2\overline{u_3}u_4$$

## III CIRCUIT AND FAULT DESCRIPTIONS

Consider a synchronous sequential circuit with *m* inputs, *n* outputs, and b-bits of memory as shown in Figure 2. In vector notation, the circuit can be described as:

Input Vector = $\overline{X}(k) = [x_i(k),\ldots, x_m(k)]$

Output Vector = $\overline{Y}(k) = [Y_1(k),\ldots,Y_n(k)]$

State Vector = $\overline{S}(k) = [S_i(k), \ldots, S_n(k)]$

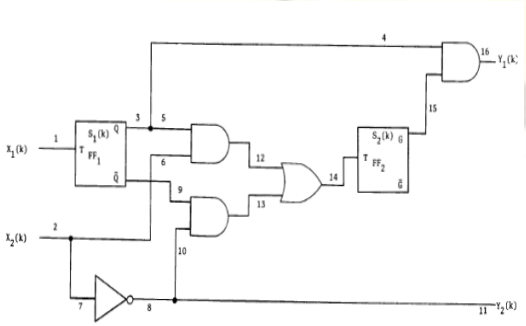

Figure 2: Synchronous Sequential Circuit

where k is the time parameter, and $X_i;(k)$, $1 \le i \le m$, $Y_i(k)$, $1 \le I \le n$ , and $S_i(k)$, $1 \le i \le b$ are Boolean variables.

Each line of the circuit is labeled with a unique integer. A line numbered *j* has a logical value $I_2[\overline{X}(k),\overline{S}(k)]$ which is dependent on the current values of the input and state vectors. Assuming that the initial state of the circuit is known as $\overline{S}(0)$, the behavior of the circuit can be described by vector-valued Boolean output and next state functions as follows:

$$\overline{Y}(k) = G[\overline{X}(k),\overline{S}(k),\overline{I}(k)], k \ge 0 \quad (1)$$

$$\overline{S}(k+1) = F[\overline{X}(k),\overline{S}(k),\overline{I}(k)], k \ge 0, \quad (2)$$

where $\overline{I}(k)$ has the form

$$\overline{I}(k) = [I_{j1}(k), I_{j2}(k), \ldots\ldots\ldots\ldots\ldots I_{jp}(k)] \quad (3)$$

$I_{ji}$,i = 1,2, . . . , *p* , is either primary input variable from the set $X_l,X_2,\ldots,X_m$, or one of the specified set of internal variables of the circuit.

In considering multiple faults, we must determine the dominating faults and discuss about equivalent faults. Two faults α, and β in the circuit starting in an initial state S'(0) are said to be strongly equivalent if the output and next-state function of the circuit with fault α, are identical to those of the circuit with fault β. Strong equivalence of faults in a synchronous sequential circuit reset-able to an initial state S'(0) is identical to the concept of equivalence of faults in the combinational circuit derived from the synchronous sequential circuit by breaking all feedback loops, and considering the current or present state vector as a pseudo-input and the next-state vector as a pseudo-output.

## IV BOOLEAN DIFFERENCE FORMULAS FOR MULTIPLE FAULT ANALYSIS

The Boolean difference [1], [2] of a Boolean function $F(u_{l,\ldots}u,X_k) = F(X_k)$ with respect to the variable $X_k$ is defined as:

$$\frac{dF(X_k)}{dXk} = F(X_k) \oplus F(\overline{X_k}) = F(1) \oplus F(0) \quad (4)$$

where 0 denotes the EXCLUSIVE-OR operator. Similarly, the double Boolean difference of $F(u_{l,}u_{2,\ldots}U_mX_i,X_i) = F(X_i,X_j)$ with respect to $X_i$ and $X_j$ is defined as:

$$\frac{d^2F(X_i,X_j)}{dX_i\,dX_j} = \frac{d}{dX_i}\left(\frac{dF(X_i,X_j)}{dX_j}\right) = \frac{d}{dX_j}\left(\frac{dF(X_i,X_j)}{dX_i}\right) \quad (5)$$

Or

$$\frac{d^2F(X_i,X_j)}{dX_i\,dX_j} = F(1,1) \oplus F(1,0) \oplus F(0,1) \oplus F(0,0) \quad (6)$$

And still more generally,

$$\frac{d^2F}{dX_{i1}\ldots.dX_i.} = \frac{d}{dX_i}\left(\frac{d^{p-1}F}{dX_{i1}\ldots.dX_{i-1}}\right) \quad (7)$$

where F-F($X_{i1,\ldots,}X_i$) = F($u_l$,…,$u_n$,$X_{i1,\ldots,}X_{ip}$).
Some other interesting properties of the Boolean difference and the EXCLUSIVE-OR operation which will be useful in the following are:

$$\frac{d}{dx}\left(F(X_i)G(X_i)\right) = G(X_i).\frac{dF(X_i)}{dX_i} \oplus F(X_i).\frac{dG(X_i)}{dX_i} \oplus \frac{dG(X_i)}{dX_i}.\frac{dF(X_i)}{dX_i} \quad (8)$$

and

$$X_I + X_J = X_I \oplus X_J \oplus X_I X_J \quad (9)$$

## V. BOOLEAN DIFFERENCE EXPRESSION FOR MULTIPLE FAULTS IN SEQUENTIAL CIRCUITS

Consider a fault involving p internal variables I'= [$I_{j1}$, $I_{j2,\ldots,}I_{jp}$] simultaneously. The internal inputs can be primary inputs, primary outputs, and memory lines. The vector Boolean difference of the output Y' with respect to I' is defined as [3].

$$\Omega_{j1j2\ldots jp}\,\overline{Y} = \sum_{i=1}^{n} {}^1G\left[\overline{X},\ \overline{S(0)},\overline{i}\right] \oplus {}^1G\left[\overline{X},\ \overline{S(0)},\overline{i}\right], \overline{I} = [\overline{I_{j1}},\overline{I_{j2}},\ldots\ldots,\overline{I_{jp}}] \quad (10)$$

One of the modifications to Goldstein's work [3] should be pointed out here. As mentioned earlier, since Figure. 3 is just a conventional combinational circuit, Eqn. (10) should be modified with a sense of sequential circuit, i.e., the state vectors cannot be treated as primary input as in a "pure" combinational circuit.

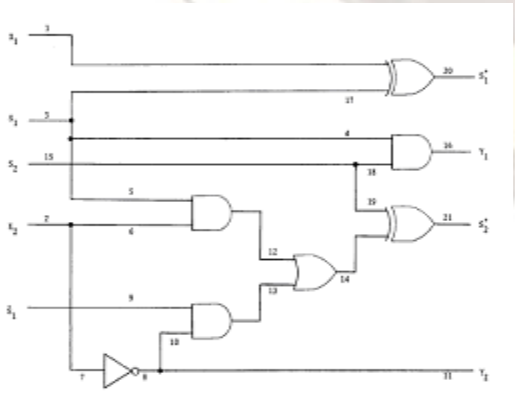

Figure 3. Combinational Circuit obtained from Synchronous Sequential Circuit of Figure 2.

The state vector S' in Eqn. (10) must be replaced by the initial state vector S'(0) as follows:

$$\Omega_{j1j2\ldots jp}\,\overline{Y} = \sum_{i=1}^{n} {}^1G\left[\overline{X},\ \overline{S(0)},\overline{i}\right] \oplus {}^1G\left[\overline{X},\ \overline{S(0)},\overline{i}\right], \overline{I} = [\overline{I_{j1}},\overline{I_{j2}},\ldots\ldots,\overline{I_{jp}}] \quad (11)$$

However, if the output functions do not depend on the selected internal variables with the initial state setting, i.e., Eqn. (11) = 0, the vector Boolean difference of the next state $\overline{S^+}$ with respect to $\overline{I}$ should be considered as in the next equation:

$$\Omega_{j1j2\ldots jp}\,\overline{S^+} = \sum_{i=1}^{b} {}^1F\left[\overline{X},\ \overline{S},\overline{i}\right] \oplus {}^1F\left[\overline{X},\ \overline{S},\overline{i}\right] \quad (12)$$

The vector Boolean differences in Eq.(11) and (12) equal to one if any term in the sum equals to one, and equal to zero only if every term in the sum equals to zero. Our next objective is to solve Eq. (11) for the input/state pairs that will sensitize the output to the differences in the specified internal variables. However, if the output functions do not depend on the selected internal variables, then Eq. (12) must be solved for the input/state pairs that sensitize the next state to the differences in the internal variables under consideration. The simplified forms for Eq.(11) and (12) are [2]:

$$\Omega_{j1j2\ldots jp}\,\overline{Y} = \sum_{i=1}^{2^{p-1}-1} (m_i + m_{2^p-1-i}) \sum_{i=1}^{n} \left({}^1G_i \oplus {}^1G_{2^p-1-i}\right) \quad (13)$$

$$\Omega_{j1j2\ldots jp}\,\overline{S^+} = \sum_{i=1}^{b} {}^1F[\overline{X},\ \overline{S},\ \overline{I}] \oplus {}^1F[\overline{X},\ \overline{S},\ \overline{I}] \quad (14)$$

Where, ${}^1G\alpha$ = ${}^1G$[X',S'(0), p-tuple of decimal equivalent of α], ${}^1F\alpha$ = ${}^1F$[X',S',p-tuple of decimal equivalent of α], and $m_r$ = p-tuple of decimal equivalent of r.

## VI DETECTING STAGES

The circuit is in a detecting state for a fault a, if an applied input can produce a difference in output between fault-free and fault-n circuits. Suppose that at time *t* the fault-free and fault-a circuits are in state $S_{ff}$ *( t )* and $S_{\pi}$*(t)*, respectively. Let $S_{ff}(t) = S_x$ and $S_{\pi}$ *(t)* = $S_y$ where $S_x$ may be the same as $S_y$. The pair *($S_x$,$S_y$)* is called a *detecting* pair *of states*, or shortly *detecting* pair, if there exists at least one input vector which when applied to both circuits will

produce a difference in output between them at time $(t + 1)$ which means that the fault π is detected.

Goldstein [3] has used a successor tree for keeping track of the states reached and the outputs generated by the fault-free and faulty circuits. We use a different technique called detecting tree in which the transition tables of both circuits are used to find the set of detecting pairs $\Sigma^*_\pi$ for fault π, and to form the detecting tree searching for the shortest input sequence which will lead the fault-π circuit from the initial state to a detecting state.

Since faulty lines may be memory or state lines, and since we are looking for test sequences of length greater than one, we may drive some of the faulty state lines to logical values that are complements of their faulty values in the process of driving the fault-a circuit through some different states before reaching the first detecting state. So for simplicity, we look for both type I and type I1 test sets for fault π.

## VII DETECTING TREE

If the test sequence of length one does not exist, or the output difference between the two circuits cannot be obtained immediately, we can try to find a test sequence of length greater than one, if one exists, from the detecting tree. The procedure for construction of the detecting tree is as follows:

*Step* 1: Form the output and next-state equations and solve Eqns.(9) and (10). Case I (Eqn. (9) = 0, Eqn. (10) # 0): This indicates that the output functions do not depend on the specified multiple fault and the selected internal variables at all, but the next state functions do.

Case I1 (Eqn. (9) # 0): This is the case in which the output functions depend on the specified multiple fault, but not on the selected internal variables. Note that there is no need to check if Eqn.(lO) # 0 in this case, Case I11 (Eqns.(S) and (10) = 0): This indicates that the fault is undetectable.

$$\alpha\left[I_{j_1}(s-\alpha-\alpha_{j_1}), I_{j_2}(s-\alpha-\alpha_{j_2}), \ldots\ldots\ldots, I_{j_p}(s-\alpha-\alpha_{j_p})\right] = \{I_{j_1}(\overline{\alpha_{j_1}})[\Omega_{j_1}\overline{Y}] + I_{j_2}(\overline{\alpha_{j_2}})[\Omega_{j_2}\overline{Y}] + \cdots\ldots + I_{j_p}(\overline{\alpha_{j_p}})[\Omega_{j_p}\overline{Y}] + I_{j_1}(\overline{\alpha_{j_1}})I_{j_2}(\overline{\alpha_{j_2}})[\Omega_{j_1 j_2}\overline{Y}] + \cdots + I_{j_{p-1}}(\overline{\alpha_{j_{p-1}}})I_{j_p}(\overline{\alpha_{j_p}})[\Omega_{j_{p-1}j_p}\overline{Y}] \cdots + I_{j_1}(\overline{\alpha_{j_1}})\cdots I_{j_p}(\overline{\alpha_{j_p}})\Omega_{j_1 j_2 \ldots j_p}\overline{Y}] = 1\} \quad (15)$$

$$\gamma[I_{j_1}(s-\alpha-\alpha_{j_1}), I_{j_2}(s-\alpha-\alpha_{j_2}), \ldots, I_{j_p}(s-\alpha-\alpha_{j_p})] = \{I_{j_1}(\overline{\alpha_{j_1}})[\Omega_{j_1}S^+] + I_{j_2}(\overline{\alpha_{j_2}})[\Omega_{j_2}S^+] + \cdots + I_{j_p}(\overline{\alpha_{j_p}})[\Omega_{j_p}S^+] + I_{j_1}(\overline{\alpha_{j_1}})I_{j_2}(\overline{\alpha_{j_2}})[\Omega_{j_1 j_2}S^+] + \ldots + I_{j_{p-1}}(\overline{\alpha_{j_{p-1}}})I_{j_p}(\overline{\alpha_{j_p}})[\Omega_{j_{p-1}j_p}\overline{S^+}] + \ldots + I_{j_1}(\overline{\alpha_{j_1}})\ldots\ldots I_{j_p}(\overline{\alpha_{j_p}})[\Omega_{j_1 j_2 \ldots j_p}\overline{S^+}] \neq 0 \quad (16)$$

## VIII APPLICATIONS AND DISCUSSION

In the previous section we have developed Boolean difference formulas for various types of multiple fault situations. These results are not only interesting in the sense that they give a physical meaning to expressions which have heretofore had no such interpretation associated with them, but are also clearly useful in the fault analysis of combinational circuits. The attractive completeness of the Boolean difference concept has been extended to the multiple fault case. For example, suppose we are given some multiple fault. To use the developed results, we would first determine the corresponding reduced multiple fault, and then express the Boolean function realized by the circuit in terms of the primary inputs and the lines on which the reduced multiple fault existed. If two or more components of this reduced multiple fault were on paths of the circuit which emanated from the same fan-out point, or if one of the components was at a fan-out point and one or more of the remaining components were on paths emanating from this fan-out point, then with the resulting Boolean expression, we would be effectively considering an equivalent circuit where such a point of fan-out has been removed. This is a significant observation since it indicates that the results of the previous section involve an algebraic form of the conversion of a circuit to an equivalent fan-out-free circuit used by Schertz and Metze [4] for the purposes of making combinational circuits readily diagnosable for multiple faults.

## IX RESULTS AND OBSERVATIONS

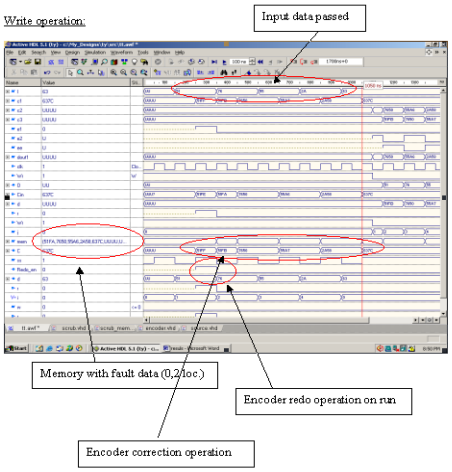


Figure 4. Write operation

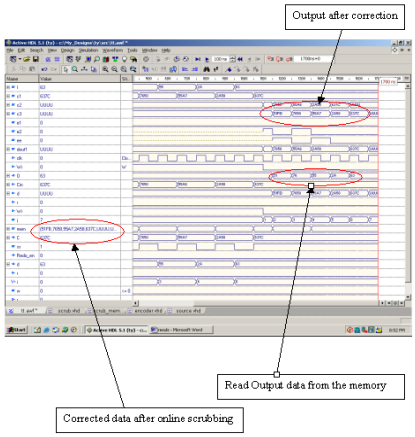


Figure 5. Read Operation

**TRB Partition Summary**

No partition information was found.

**Device Utilization Summary**

| Logic Utilization | Used | Available | Utilization | Note(s) |
|---|---|---|---|---|
| Number of Slice Flip Flops | 113 | 88,192 | 1% | |
| Number of 4 input LUTs | 123 | 88,192 | 1% | |
| **Logic Distribution** | | | | |
| Number of occupied Slices | 134 | 44,096 | 1% | |
| Number of Slices containing only related logic | 134 | 134 | 100% | |
| Number of Slices containing unrelated logic | 0 | 134 | 0% | |
| **Total Number of 4 input LUTs** | 261 | 88,192 | 1% | |
| Number used as logic | 123 | | | |
| Number used as a route-thru | 138 | | | |
| Number of bonded IOBs | 34 | 1,164 | 2% | |
| IOB Flip Flops | 14 | | | |
| Number of PPC405s | 0 | 2 | 0% | |
| Number of GCLKs | 1 | 16 | 6% | |
| Number of GTs | 0 | 20 | 0% | |
| Number of GT10s | 0 | 0 | 0% | |
| **Total equivalent gate count for design** | 2,633 | | | |
| Additional JTAG gate count for IOBs | 1,632 | | | |

Synthesis Report

Figure 6. Summarized synthesis report for the developed estimation system.

The obtained power analysis report is obtained as,

```
--------------------------------------------------------------------------------
Release 9.1i - XPower SoftwareVersion:J.30
Copyright (c) 1995-2007 Xilinx, Inc.  All rights reserved.
Design:       C:\Xilinx91i\BMA\video_codec.ncd
Preferences:  video_codec.pcf
Part:         2vpx70ff1704-6
Data version: PREVIEW,v1.0,05-28-03

Power summary:                          I(mA)     P(mW)
--------------------------------------------------------------------------------
Total estimated power consumption:                  181
                                          ---
                        Vccint 1.50V:       85       128
                        Vccaux 2.50V:       20        50
                        Vcco25 2.50V:        2         4
                                          ---
                             Clocks:         0         0
                             Inputs:         0         0
                              Logic:         0         0
                            Outputs:
                               Vcco25        0         0
                            Signals:         0         0
                                          ---
                Quiescent Vccint  1.50V:       85       128
                Quiescent Vccaux  2.50V:       20        50
                Quiescent Vcco25  2.50V:        2         4

Thermal summary:
--------------------------------------------------------------------------------
   Estimated junction temperature:                   25C
                        Ambient temp:  25C
                           Case temp:  25C
                           Theta J-A:   0C/W
--------------------------------------------------------------------------------
```

Figure 7. Power analysis

Timing report for the implementation is observed as,

Minimum period: 7.693ns (Maximum Frequency: 129.984MHz)
Minimum input arrival time before clock: 2.950ns
Maximum output required time after clock: 3.615ns

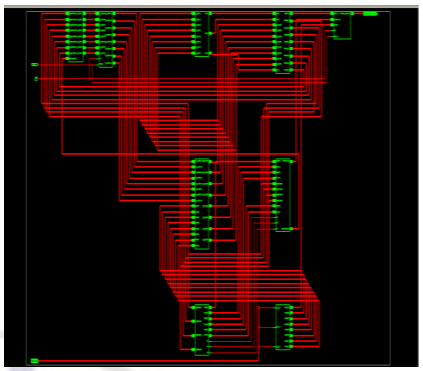

Figure 8: RTL view of the implemented system using Xilinx synthesizer.

## X CONCLUSION

This paper contributes in developing simplified Boolean logic architecture for the detection and analysis of multiple faults in programmable devices. A diagnosing approach for multiple faults at one time in various lines or locations is been suggested the simulation observations were carried out for the suggested approach defined using VHDL definition. The fault detecting system is analyzed for both sequential and combinational circuits with stuck-at one and stuck-at zero fault simultaneously.

## XI REFERENCES

[1] C. T. Ku and G. M Mason, "The Boolean Differ ence and Multiple Fault Analysis", ZEEE Trans. on Computers, Vol. C-21, No.1 Jan 1975, pp.62-71.

[2] S. R. Das, P. K. Srimani, and C R. Dutta, "On Multiple Fault Analysis in Combinational circuits by Means of Boolean Difference", P.m of the ZEEE Vol 64, No.9, Sept.1976, pp.1447-1449

[3] L.H Goldstien "Analysis of Multiple Faults in Synchronous Sequential Circuits by Boolean Difference Techniques", Private Communication

[4] D R Schertz and G Metze "On the design of multiple fault diagnosable network", IEEE Trans. Comput., 1361-1364, Nov 1971. vol.C-20,pp

[5] A Dehon and M. J Wilson, "Nanowire based Sub lithographic programmable logic arrays", in Proc. Int. Symp, Field program Gate Arrays, Feb 2004, pp 123-132

[6] H. Naeimi and A DeHon, "Fault secure encoder and decoder for memory applications", in Proc. IEEE Int Symp Defect Fault Tolerance VLSI Syst., Sep 2007, pp.409-417

[7] S.J. Piestrak, A.Dandache and F Monterio, "Design ing fault secure parallel encoders for systematic linear error correcting codes", IEEE Trans. reliability , vol.52 no.4, pp.492-500, jul 2003.

[8] Naeimi H and DeHon A, "Fault-tolerant nano-memory with fault secure encoder and decoder," presented at the Int. Conf. Nano-Netw., Catania, Sicily, Italy, Sep. 2007.

**About the authors:**

Mr. Suresh Palaka is working as an Assistant Professor in the Dept of ECE, HITS, Hyderabad, AP, India He is currently pursuing Ph. D. from JNT University, Hyderabad, India. His research interests include Fault tolerance, Testability, VLSI and Embedded Systems.

Dr. K. Srinivasa Rao received Ph.. D. degree from Andhra University, India. He is currently working as Professor of ECE and Principal, TRRCE, Hyderabad, AP. He has 27 years of experience in the field of education. He is an active member of IEEE, ISTE, SEMCEI, BMESI. He is a Fellow of IE(l), and IETE. He has several publications in various journal and conferences of National and International repute. He is a resource person for SONET, Visiting professor for ISTE . He organized many technical events. His research interests include Communications, VLSI and Embedded Systems. He is guiding 09 scholars affiliated to JNTUH & on these submitted to JNTUH. Awaiting for vivo.