Increasing Fault Coverage in Benchmark Circuit using Design for Testability and Test Pattern Generation using 6NCA

Shashank Srivastava[1]

[1]MTech& Department of computer science and Engineering&IIT(ISM) Dhanbad Jharkhand,India

Abstract
There is high overhead in testing of full scan chain based sequential circuit. Partial scan chain can be used in testing because it involves less overhead. In this paper we propose clustering method to find center of gravity for undetected faults ,this is achieved by placing test point in fan in cone region where we have large number of undetected faults clustered in fan in cone region.we have used 6 Neighborhood cellular automata for generating patterns.we used Hope Tool for circuit Testing.

Keywords —. *Cellular Automata(CA),Test Pattern Generation(TPG),Design For Testability(DFT),Fault Coverage(FC)*

I. INTRODUCTION

Scan chain design is a method for converting sequential circuit to equivalent combinational circuit. Full scan chain involves high overhead in the circuit. Partial scan chain is an alternative to full scan chain .It includes subset of flip flop in the chain. Method used for generating partial scan chain is discussed below.

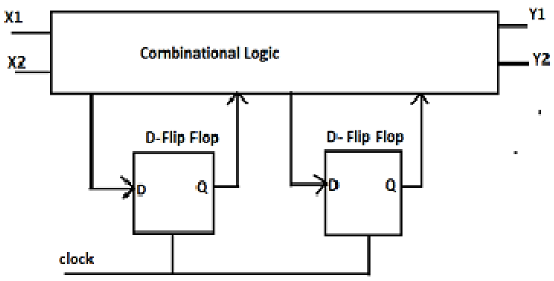


Fig 1: sequential circuit

x1,x2 are inputs and y1,y2 are output.To convert the sequential circuit to equivalent combinational circuit we use scan register in place of D-Flip Flop which can be achieved easily by adding 2x1 Multiplexer before D-Flip Flop. when the select line to 2x1 Multiplexer is at 0 then scan register is in data mode i.e value at data input line is given to D-Flip Flop. when the select line to 2x1 Multiplexer is at 1 then scan register is in Test mode i.e value at scan input line is given to D-Flip Flop.This increases thecontrollability in the circuit.

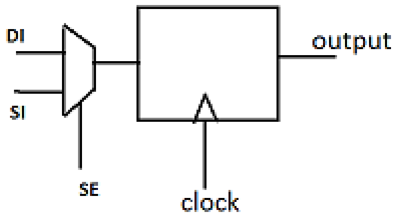


Fig 2: scan register

DI is data input,SI is scan input,SE is select line.when we connect all scan register in a chain then we obtain multiplexer based scan chain sequential circuit.In other words we can say that we have converted the sequential circuit into equivalent combinational circuit.

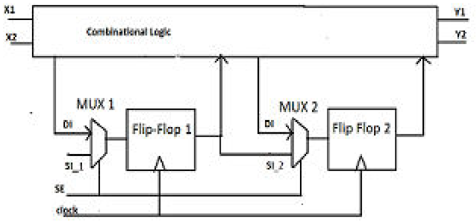


Fig 3: full scan chain based sequential circuit

SI1,SI2 are select line. To decrease the overhead in the Full scan based sequential circuit while maintaining the good fault coverage we implement Partial scan chain based sequential circuit.

II. PARTIAL SCAN CHAIN CIRCUIT GENERATION METHOD

Below figure shows an overflow of generating partial scan chain.In this paper we consider single stuck at fault stuck at 0 ,stuck at 1.Fault coverage (FC) is defined as FC = (no of detected faults) / (no of possible faults) .Fault simulation has been performed on the circuit using patterns generated through 3NCA.

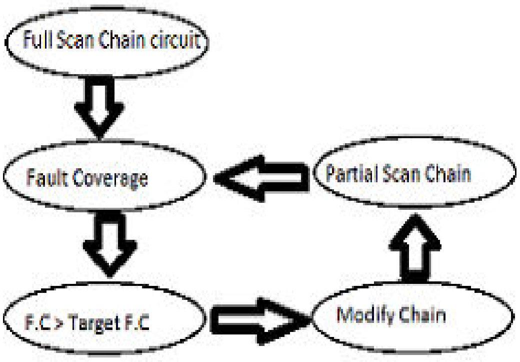


Fig 4: Flow chart for generating partial scan chain

Steps for generating partial scan chain

```
Input: Full scan chain circuit
N=number of flip flops
Output: partial scan chain circuit
i=1
while(i<=n)
{
Remove the  i^th flip flop from the scan chain
and     calculate the fault coverage.
If  Fault_coverage> Target_Fault_coverage
Remove the scan chain
Else
Do not remove the scan chain
i++;
}
```

The above algorithms make trade-offs in the percentage of flip-flops to be removed from the scan chain and the fault coverage achieved with the approach.

In our approach, the Target fault coverage can be selected prior to solving the partial scan problem. For the given Test Set. If the Fault coverage is greater than target Fault coverage then scan chain is modified by removing a flip-flop from the chain. The above process is repeated till we get the optimized partial scan chain circuit.

III.CLUSTERING OF UNDETECTED FAULTS IN FAN-IN CONE REGION

For a given line as output to a gate ,we can find fan in cone region in which we have all those lines which are in-degree/input to this gate.If we recursively follow the above step i.e taking in-degree lines to gate as output line and finding in-degree lines to this gate. In this way we are back-tracing to get bigger fan in cone region for a given line.

To achieve this we apply set of patterns to the circuit.we get fault coverage for the circuit and list of remaining undetected faults present in the circuit.So our objective is to find fan in cone region for the circuit which will contains large number remaining undetected faults present in thecircuit.we find out the location and apply testpoint insertion in that region. This observability point detect large number of faults present in the circuit which helps us to get high fault coverage.

We try to find out different Fan-In cone Region for the remaining undetected faults in the circuit.we add observability point in different fan in cone region to get high fault coverage.we may add more than one observability point in same fan in cone region. This increases our probability of getting high Fault coverage.

Steps to add Observability point

Step1 We try to find Fan-In cone region for the selected linename by backtracing every non primary input line to the gate.Backtracing for the selected line is done for 5 times to get large fan-in cone region for the selected Fan-In cone region.

Step2 Repeat step1 for every linename in the circuit.

Step3 Count the number of faults present in every fan-in cone region.Arrange them in decreasing order of their count and show them on print screen.

Step4 Add observabilty point to fan-in cone region according to their decreasing order of their count.Adding observability point can be done either by placing a single observability point at the selected line for which fan-in cone region is calculated or by placing more than one observability point in Fan-in cone region for the selected line this can be done by adding observability point then calculate fault coverage if the fault coverage is high we repeat step 4 for different fan in cone else we discard the current observability point,webacktrace and add different observability point in same Fan in cone region.

Our objective is to find fan in cone region which can cluster large number of undetected faults into region ,in other words we can say that large number of remaining undetected faults are present in this region.Lines on which these faults are present we say those lines as undetected lines.Adding observability point in this region will fulfill our objective because this observability point has capability to detect large number of remaining undetected faults.

This can be done by finding set of fan-in cone region in the circuit and mapping with the faults list to get the count for faults which lies on undetected lines of fan-in cone region,which means that these faults can be detected if observability point is placed in fan-in cone region which has highest count for faults.

we take a Test Set T.Fault coverage obtained is 93.051 % and number of undetected fault is 41.our objective is to place observability point efficiently such that large number of undetected fault can get detected.

Table 1,Table 2,Table 3 represent the fault coverage obtained by adding observability point at different location based on decreasing order of values presentat different level.we add only one observability point in the circuit.

List of remaining undetected faults present in the circuit.

I216 /1,SS0 /0,T14 /0, I5 /1,I298 /1,I827 /1,I531->I827 /0,I1062_1 /0,I598 /1,I889 /1, st_4->I1092_1 /0,I478->I1092_1 /0,I1116_2 /0,I230 /1,cnt284->I958_2 /0,I642->I958_2 /0,I57->I958_1 /0,I59->I958_1 /0,I1116_1 /0,I282 /1,I483->I978_1 /0,I638->I747 /0,I1044_1 /0,I70 /1,I539->I298 /1,I574->I298 /1,I535->I1062_1 /1,cnt13->I598 /1,I56->I598 /1,I870 /0,I566->I870 /1,I1092_1 /1,I61->I1116_2 /1,I958_2 /1,I958_1 /1,I95->I1116_1 /1,I603->I1116_1 /1,st_1->I282 /1,I978_1 /1,I747 /0,cnt284->I1044_1 /1

List of remaining undetected faults present in the circuit after adding observability point at I5 based on Table 1.

I216 /1,SS0 /0,T14 /0,I5->T14 /1

List of detected faults

I5 /1,I298 /1,I827 /1,I531->I827 /0,I1062_1 /0,I598 /1,I889 /1,st_4->I1092_1 /0,I478->I1092_1 /0,I1116_2 /0,I230 /1,cnt284->I958_2 /0,I642->I958_2 /0,I57->I958_1 /0,I59->I958_1 /0,I1116_1 /0,I282 /1,I483->I978_1 /0,I638->I747 /0,I1044_1 /0,I70 /1,I539->I298 /1,I574->I298 /1,I535->I1062_1 /1,cnt13->I598 /1,I56->I598 /1,I870 /0,I566->I870 /1,I1092_1 /1,I61->I1116_2 /1,I958_2 /1,I958_1 /1,I95->I1116_1 /1,I603->I1116_1 /1,st_1->I282 /1,I978_1 /1,I747 /0,cnt284->I1044_1 /1.

Based on table 1 we found 4 fan in cone .similarly we can find many fan in cone.Adding observability point in those fan in cone which has high count at level 3 will results in overal high fault coverage.

TABLE I shows Adding observability point based on decreasing order of values at level 3.

TABLE II shows Fault coverage by adding observability point based on decreasing order of values at level 4.

TABLE III shows Fault coverage by adding observability point based on decreasing order of values at level 5.

when we compare first starting rows of Table 1,Table 2,Table 3.High Fault Coverage is obtained in Table 1.Therefore in our further experiment we will add observability point in fan in cone region of linename based on their values obtained in level 3.

Based on Table 1
I5is lying in Fanin cone of T14.
T14 is lying in fan in cone of T16.
I889 is lying in fan in cone of I5.
I827 is lying in fan in cone of T16.
I1116_2 is lying in fan in cone of T16.
I4 is another fan in cone region.
I230 is lying in T14.
I270 is lying in fan in cone of I4.
I232 is another fan in cone region.
I595 is another fan in cone region.

When observability point are placed in all these fan in cone region then we get high fault coverage for the circuit.since we found out the region where hard faults can be detected now the modified circuit after adding different observability point will always give high fault coverage for any test set .

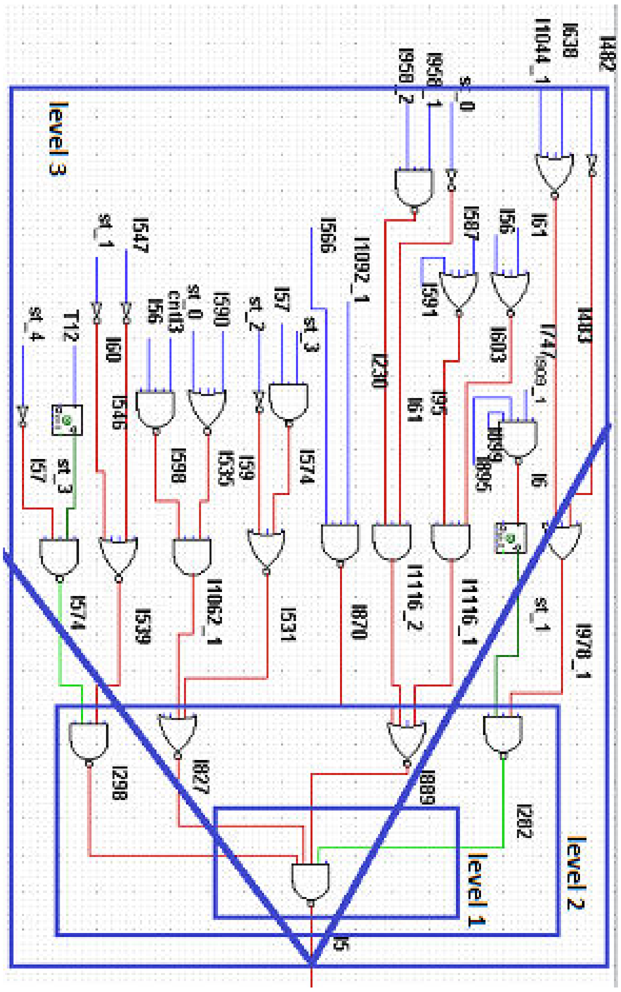


Fig 5: fan in cone region for I5 in s510

TABLE I

Line name	Fault Coverage	No of undetected faults	No of faults present on undetected lines of level 1	No of faults present on undetected lines of level 2	No of faults present on undetected lines of level 3	No of faults present on undetected lines of level 4	No of faults present on undetected lines of level 5	Fault coverage	No of undetected faults
I5	93.051	41	9	34	88	98	99	99.327	4
T14	93.051	41	2	12	69	98	99	99.327	4
I889	93.051	41	8	25	51	86	99	95.960	24
T16	93.051	41	1	4	37	98	99	93.266	40
I827	93.051	41	8	21	31	63	99	94.108	35
I1116_2	93.051	41	5	11	23	51	99	94.414	33
I4	93.051	41	0	4	22	59	99	93.098	41
I230	93.051	41	7	17	21	64	99	94.108	35
I270	93.051	41	0	0	9	73	99	93.264	40
I232	93.051	41	0	2	6	33	99	93.266	40
I595	93.051	41	0	0	4	92	99	93.243	40

TABLE II

Line name	Fault Coverage	No of undetected faults	No of faults present on undetected lines of level 1	No of faults present on undetected lines of level 2	No of faults present on undetected lines of level 3	No of faults present on undetected lines of level 4	No of faults present on undetected lines of level 5	Fault coverage	No of undetected faults
T14	93.051	41	2	12	69	98	99	99.495	3
st_2	93.051	41	0	1	12	98	99	93.243	40
I642	93.051	41	4	6	10	95	99	93.074	41
I566	93.051	41	4	6	10	91	99	93.074	41
I1089_1	93.051	41	2	4	5	80	99	93.266	40
st_4	93.051	41	2	2	4	21	99	93.074	41
I475	93.051	41	0	4	6	12	92	93.074	41
I598	93.051	41	7	7	7	9	99	93.434	39
T7	93.051	41	0	2	2	4	99	93.266	40

TABLE III

Line name	Fault Coverage	No of undetected faults	No of faults present on undetected lines of level 1	No of faults present on undetected lines of level 2	No of faults present on undetected lines of level 3	No of faults present on undetected lines of level 4	No of faults present on undetected lines of level 5	Fault coverage	No of undetected faults
T8	93.051	41	0	2	6	22	99	93.098	41
T11	93.051	41	4	4	4	21	99	93.266	40
I1113_1	93.051	41	2	2	8	43	99	93.098	41
I810	93.051	41	2	2	4	56	99	93.098	41
I483	93.051	41	2	2	8	21	94	93.074	41
cnt13	93.051	41	2	2	2	2	2	93.074	41
cnt28	93.051	41	4	2	2	2	2	93.074	41

TABLE IV

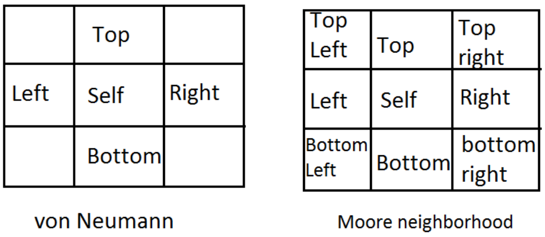
Line name	Fault Coverage	No of undetected faults	No of faults present on undetected lines of level 1	No of faults present on undetected lines of level 2	No of faults present on undetected lines of level 3	No of faults present on undetected lines of level 4	No of faults present on undetected lines of level 5	Fault coverage	No of undetected faults
I359	80.198	80	9	15	51	93	144	87.432	69
I378.4	80.198	80	6	14	45	85	144	87.432	69
I384.1	80.198	80	10	19	41	98	148	87.432	69
I378.2	80.198	80	4	10	40	70	102	82.396	97
I1.4	80.198	80	9	22	39	91	101	72.414	152
I355	80.198	80	7	16	39	101	158	72.414	152
I306	80.198	80	8	19	35	94	158	72.051	154
T18	80.198	80	0	8	8	16	27	72.777	150
T23	80.198	80	2	2	4	19	27	72.777	150

Similary for s420.1
TABLE IV shows the fault coverage by adding observability point based on decreasing order of values at level 3.

Adding observability point in fan in cone of I359 gives high fault coverage.

IV.Generating patterns using 6NCA

The neighborhood of a cell is the adjacent cells. The two most common types of neighborhoods are the von Neumann neighborhood and the Moore neighborhood.



	Top	
Left	Self	Right
	Bottom	

von Neumann

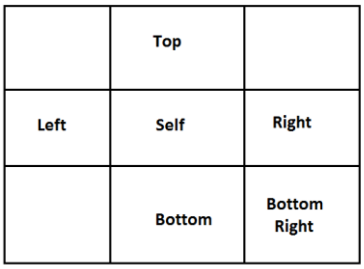
Top Left	Top	Top right
Left	Self	Right
Bottom Left	Bottom	bottom right

Moore neighborhood

Fig 6: different neighborhood model

A 2-D CA is a generalization of 1-D CA ,where the cells are arranged in 2-dimensional grid with Connection among the neighboring cells .

consider the 3x3 array with 3 rows and 3 columns comprising 9 cells.Theneighborhood function specifying the next state of a cell is assumed to depend on left neighbor ,itself,rightneighbor,topneighbour,bottomneighbor,bottomright neighbour.

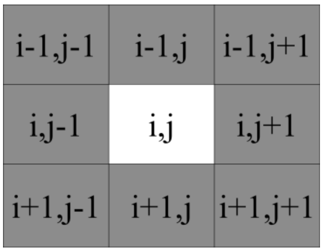


	Top	
Left	Self	Right
	Bottom	Bottom Right

Fig 7: 6 Neighborhood model

Thus the next state m of the (i,j)th cell of 2-D CA is given by

$$m_{i,j}\ (t+1)=f[m_{i,j}(t),\ m_{i-1,j}\ (t),\ m_{i,j-1}\ (t),m_{i+1,j}(t)\ ,\ m_{i,j+1}\ (t)\ ,\ m_{i+1\ ,j+1}\ (t)\ ]$$

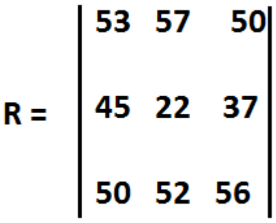


i-1,j-1	i-1,j	i-1,j+1
i,j-1	i,j	i,j+1
i+1,j-1	i+1,j	i+1,j+1

Fig 8: Adjacent cells

the dependency or rule can be expressed as 6-bit number.[self,top,left,bottom,right,bottom-right] , where each bit signifies the presence of the corresponding dependency.

Consider a 3x3 2-D CA dependency. Below is an example of hybrid CA where not all the rules of the cells aree identical .This hybrid dependency can be expressed by a rule matrix R , whose (i,j)th element gives the rule of the (i,j)th cell of the CA.



$$R = \begin{vmatrix} 53 & 57 & 50 \\ 45 & 22 & 37 \\ 50 & 52 & 56 \end{vmatrix}$$

Fig 9: Rule matrix

For 2DCA 6NCA

2^6 = 64 possible "neighborhoods"

2^{64} possible rules

For 2D CA:

2^9 = 512 possible "neighborhoods"

2^{512} possible rules

TABLE V 6NCA rules

Rule	Self	Top	Left	Bottom	Right	Botto m-Right
53	1	1	0	1	0	1
57	1	1	1	0	0	1
50	1	1	0	0	1	0
45	1	0	1	1	0	1
22	0	1	0	1	1	0
37	1	0	0	1	0	1
50	1	1	0	0	1	0
52	1	1	0	1	0	0
56	1	1	1	0	0	0

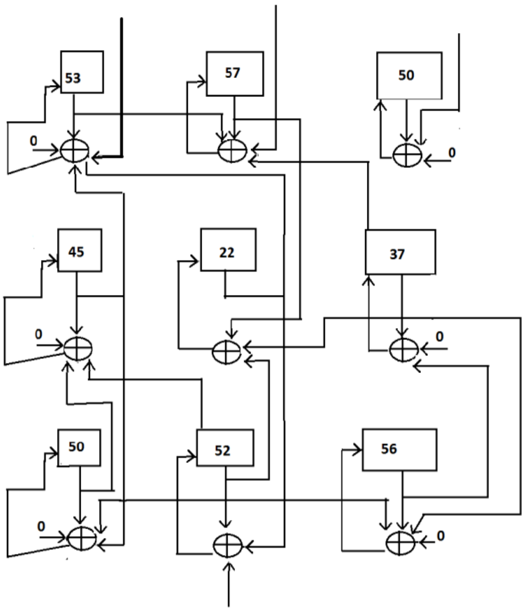


Fig 10: 3x3 6NCA Rule

A simple way of representing 2-D CA is to consider it as an extended neighbourhood 1-D CA,define its characteristics matrix and use the technique already exists for 1-D CA.however in 2-D CA we have to deal with n^2 x n^2 T matrix.The T matrix for the above Rule matrix 3x3 2-D CA is as below

For example

$$T = \begin{vmatrix} 1\ 0\ 0 & 1\ 0\ 0 & 1\ 0\ 0 \\ 1\ 1\ 0 & 0\ 0\ 0 & 0\ 1\ 0 \\ 0\ 0\ 1 & 0\ 0\ 0 & 0\ 0\ 0 \\ 0\ 0\ 0 & 1\ 0\ 0 & 1\ 0\ 0 \\ 0\ 1\ 0 & 0\ 0\ 1 & 0\ 1\ 0 \\ 0\ 0\ 0 & 0\ 0\ 1 & 0\ 0\ 1 \\ 1\ 0\ 0 & 1\ 0\ 0 & 1\ 1\ 0 \\ 0\ 0\ 0 & 0\ 1\ 0 & 0\ 1\ 0 \\ 0\ 0\ 1 & 0\ 0\ 1 & 0\ 1\ 1 \end{vmatrix} = \begin{vmatrix} A & B & C \\ D & E & F \\ G & H & I \end{vmatrix}$$

Fig 11: characteristic T matrix

The sub-matrix A,E,I contains complete information regarding the horizontal dependency of the CA Cells.

$$E = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{vmatrix}$$

Fig 12: horizontal dependency matrix

The Sub-matrix B,F contains information regarding bottom dependencies.

$$B = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix}$$

Fig 13: bottom dependency matrix

The sub-marix D,H contains information regarding the top dependencies.

$$D = \begin{vmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{vmatrix}$$

Fig 14: top dependency matrix

The sub-matrix C,G contains information regarding bottom-Right dependencies.

$$C = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{vmatrix}$$

Fig 15: bottom right dependency matrix

This can be drawn from table 6 that good fault coverage can be achieved using partial scan chain with clustering which results in less/comparable overhead in the circuit as compared to overhead due to Full scan chain.

TABLE VI

Circuit name	No. of primary input	No. of output	Fault coverage [%]	Total no of faults	Fault coverage using paper related to CA
c17	5	2	100.000	22	NA
c432	36	7	99.237	524	97.09
c499	41	32	98.945	758	98.69
c880	60	26	100.000	942	96.39
c1355	41	32	99.492	1574	98.79
c1908	33	25	99.521	1879	97.33
c6288	32	32	99.561	7744	99.37
c2670	233	140	49.399	2747	88.24
c3540	50	22	96.004	3428	96.06
c5315	178	123	53.215	5350	97.79
c7552	207	108	64.874	7550	93.24

TABLE VI shows Experimental Result of Fault Coverage Table for 6NCA in combinational circuit without DFT

TABLE VII shows Experimental Result of Fault Coverage Table for 6NCA in sequential circuit with DFT

cuit name	Number of lines (input + output)	no of D-Flip Flops	No of faults	Applying Full scan chain			Applying Partial Scan chain				Hybrid (Partial + Observability			Fault coverage of previous paper
				Number of extra inserted lines (scanin+ no 2X1 MUX + scan out)	No of faults	Fault coverage using 6NCA	Number of extra inserted line(Select line and scan in	No of faults	no of D-Flip Flop in scan chain	Faultcoverage usi6NCA [%]	Extra added observability point	No of faults	Fault coverage using 6NCA [%]	
S386	7+7	6	384	1+6+1	434	96.313	2	404	3	80.198	7	432	91.435	95.8
s510	19+7	6	564	1+6+1	614	99.674	2	590	4	92.881	1	594	99.158	99.29
S820	18+19	5	850	1+5+1	892	97.534	2	864	2	94.907	4	878	98.178	93.29
S838	35+2	32	857	1+32+1	1115	53.901	2	991	22	48.940	13	1025	61.951	57.14
S838.1	34+1	32	931	1+32+1	1127	76.664	2	1113	30	49.416	7	1139	76.910	NA

TABLE VII

IV CONCLUSION

This paper presents a better way to generate Test pattern using Cellular Automata.Using DFT Techniques we converted sequential circuits into equivalent combinational circuit.From experiment we can conclude the Test patterns gives good fault coverage.

V ACKNOWLEDGEMENT

I am thankful to my guide,parent who motivated me to do well.i am thankful to IIT(ISM) Dhanbad for providing good research facility.

I am also thankful to my juniors sashikantsharma and vikasojha.

REFERENCES

[1] Hierarchical cellular automata as an on-chip test pattern generator . B.K. Sikdar ; P. Majumder ; M. Mukherjee ; N. Ganguly ; D.K. Das ; P.P. Chaudhuri. Date of Conference: 7-7 Jan. 2001.

[2] Nonlinear CA based Design of Test Set Generator Targeting Pseudo Random Pattern Resistant Faults. SukantaAnirbanKunduBiplab K Sikdar. Date of Conference: 7-7 Jan. 2001

[3] A Fault Coverage-Driven Partial Scan Chain Selection Technique. Clay Gloster, Jr, Siva Subramanian. Department of Electrical and Computer Engineering North Carolina State University P.O. Box 7911, Raleigh, N.C. 27695-7911. Date of Conference: 19-23 Sept. 1994

[4] Additive Cellular Automata: Theory and Applications, Volume 1 . By Parimal Pal Chaudhuri, Dipanwita Roy Chowdhury, Sukumar Nandi, SantanuChattopadhyay July 1997, Wiley-IEEE Computer Society Press.

[5] Digital Systems Testing and Testable Design - MironAbramovici, Melvin a. Breuer, D Arthur Friedman, 1990, . Computer Science Press.

[6] On Clustering of Undetectable Single Stuck-At Faults and Test Quality in Full-Scan Circuits. IrithPomeranz ; Sudhakar M. Reddy. Date of Publication: 17 June 2010

[7] VLSI Test Principles and Architectures Design for Testability Book by Cheng Wen Wu, LaungTerng Wang, and Xiaoqing Wen. Elsevier Aug 2006 Technology and Engineering.

[8] Theory and application of GF(2/sup p/) cellular automata as on-chip test pattern generator. B.K. Sikdar ; K. Paul ; G.P. Biswas ; V. Boppana ; C. Yang ; S. Mukherjee ; P.P. Chaudhuri. Date of Conference: 3-7 Jan. 2000