Optimization of Critical Path Tracing Through Output Based Clustering Technique

P.Premalatha[1], S.Saravanan[2], R.VijaySai[2]
[1]M.Tech-VLSI Design, SASTRA University
[2]Assistant Professor, SASTRA University
Thanjavur, India
[1]p.premalatha90@gmail.com
[2]saran@core.sastra.edu

***Abstract* - Designing of modern digital circuits require high performance with reduced cost and minimal time to market. In order to achieve greater performance, timing analysis is done to meet all the timing constraints. It also leads to increase the complexity of emerging Very Large Scale Integration (VLSI) design. Timing analysis eliminates the occurrence of non-functional path. In this work, path tracing and clustering algorithms are proposed to optimize the critical path. The path elimination technique based on clustering is tested on some combinational benchmark circuits.**

Keyword- Critical path, Path Tracing, Clustering, Timing analysis

I. INTRODUCTION

The development cycles of integrated circuits include design, optimization and testing. In all the above phases, timing analysis is basic and very much essential step. Because, timing analysis gives the rate (speed) at which the design operates. The longest propagation delay of the circuit should be less than the overall system clock cycle time. All the available timing analysis tools take the longest structural path (critical path) of the circuit as the maximum delay. Sometimes, the above calculated delay maybe wrong. So the timing analysis is failed.

The fundamental need of the timing analysis process is the accurate prediction about the circuit delay. Most of the high performance circuits have nearly equal length paths. The exact delays of these equal paths are based on their operational environment and process parameters. It also increases the test suits. Hence the critical path must be optimized. It is more cost effective and reduces the time to market constraints.

II.RELATED WORK

Numerous research works are developed to achieve the path elimination and critical path reduction. The simultaneous switching of circuit paths can be eliminated through slack value based clustering. The slack value is based on the delay value which can be calculated through the application of the two algorithms namely As Soon As Possible (ASAP) and As Late as Possible (ALAP) [1]. Critical path tracing is implemented through stem analysis. Graph based approach is used here. Dynamic data structure is used during the backward pass for achieving path sensitization and self masking. This approach eliminates the forward propagation of individual stem faults [2].The extension of critical path tracing is done for the sequential circuits. Finally it produces optimistic results for the sequential circuits based on the pessimistic approach to combinational circuits. This approach is well suited for the synchronous sequential circuits [3]. The transition activity reduction is proposed to achieve low power requirements and the power issues also focused here. The resource binding and scheduling steps are operates on data path through high level synthesis [4].

In paper [5], fault simulation is achieved through a series of tests. By using backtracing algorithm, critical path tracing is achieved. But the major drawback of this approach is that it only gives the approximated results. The critical path (which decides the overall process run time) is calculated in a highly scalable way. The real world parallel applications and the comparison between the traditional profiles are discussed here [6]. An efficient critical path tracing algorithm is proposed for both stuck at and transition faults. This approach is linear in terms of the number of gates being traced. This is the fast and efficient approach [7].

III.PROPOSED WORK

In this paper, the observation on a set of paths for an ISCAS85 benchmark circuit [8] is described. An approach to determine a set of longest paths is developed. A clustering scheme for the elimination of critical path is implemented and some combinational benchmark circuits are used to demonstrate the performance.

Fig.1 shows the process flow for the proposed work. The initial step is the creation of a structural netlist. The netlist can be generated through the design compiler tool by taking the behavioral description of a

benchmark circuit as input. Then the bench file is modified so that the each gate in a file should have minimum of two inputs. The next step is the application of the path tracing algorithm. The algorithm is effective in tracing the number of possible paths from primary input to primary output. Hence, it results with the '**n**' number of paths through the network.

This work primarily focuses the reduction of non-functional paths through the network. To achieve this, clustering algorithm is implemented. Here clustering (grouping) is based on the primary output. And, the algorithm is mainly concentrates on the critical paths that is, the longest path through the network. It is cleared that the number of clusters are equal to the number of primary outputs. Clustering algorithm takes '**n**' number of possible paths as inputs.

The output will be the number of clusters with reduced set of possible paths. The algorithmic steps and the description of the two algorithms are discussed below. Finally the reduced set of critical path is obtained with respect to the primary outputs.

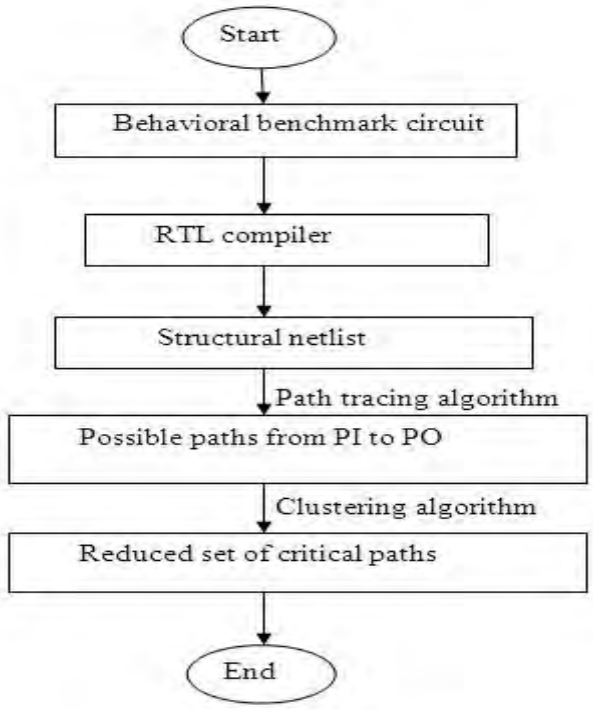

Fig.1.Process flow

A. Path tracing algorithm

The input for the path tracing algorithm is the structural netlist. Structural netlist can be generated with the help of RTL compiler. The operational steps for this algorithm are given. It monitors the number of primary inputs and primary outputs. The entire functional gates should have only two inputs. The output for this algorithm is the number of possible paths from PI to PO.

Here, for demonstration C17-ISCAS85 benchmark circuit is considered. It gives all the possible paths through the network. A path with the longest duration is known as the critical path. Critical path impacts mainly on the delay, speed and timing constraints. Hence, we have to optimize the critical path.

Pseudo code for Path Tracking algorithm:

1. Create a parser function.
2. Check the variables from input file.
3. Copy root id→o/p.
4. Copy op1→operand 1.
5. Copy op2→operand 2.
6. Create a binary tree with root id and child as op1 and op2.initiaize cf=0.
7. If Op1→already root,
8. Then Create a binary tree with root id and take op1 as left child.op1.cf=1.
9. If op2→already root,

Then create a binary tree with root id and take op2 as right child.op2.cf=1.

10. If op1 and op2 are not root,
    Then use the initial condition.
11. Increment the rules count by 1.
12. While rules count>1, then do traversal.

B. Clustering algorithm

A cluster is a collection of similar objects. And therefore objects of one cluster dissimilar with another cluster. Simply, cluster refers to grouping. Hence, clustering algorithm is used for grouping. The input for this algorithm is the number of possible paths from the path tracing algorithm. Here, clustering is based on the primary outputs.

Pseudo code for Clustering algorithm:

1. Open the input file.
2. Find the path length.
3. Set a loop for 0 to n and check the path length.
4. Take cluster number as 1 and do it for 0 to i (i<n).
5. Check cluster elements and take the next cluster element for i+1 to n.

IV.EXPERIMENTAL RESULTS

Some combinational circuits and ISCAS85 benchmark circuit [8] are targeted by these proposed two algorithms. The results are obtained. Initially, RTL compiler is used for the extraction of bench file from the behavioral description.

The relationship between the number of clusters and number of primary outputs for some combinational benchmark circuits are shown in Table 1. Clustering is done for the paths with maximum delay (critical path) based on the primary outputs.

TABLE 1. Comparison results for clustering

Circuit	#PI	#PO	#Clusters
HS(Half subtractor)	2	2	1
FA(Full adder)	3	2	1
MUX	6	1	1
DEMUX	3	4	1
C17	5	2	2
C432	36	7	3
C499	41	32	18

TABLE 2. Comparison of path reduction

Circuit	#Paths	#Necessary paths	%Path reduction (formula 1)
HS(Half subtractor)	5	1	80
FA(Full adder)	9	4	55
MUX	12	2	83
DEMUX	12	2	83
C17	11	6	45
C432	288	37	87
C499	1376	218	84

Path reduction (%) is calculated by,

((#Paths - #Necessary paths)/ (#Paths))×100 ---- (formula 1)

For discussion, take the C17-ISCAS85 benchmark circuit. The netlist is formed from the behavioral level description. Now the netlist file is ready. This circuit has five numbers of primary inputs and two numbers of primary outputs. The bench file is modified in order to have the gates with two numbers of inputs.

Then, our first algorithm is applied. It gives the possible paths from primary input to primary output. It gives 11(say n) numbers of possible paths. These paths are given as the input and given to our second proposed algorithm. Only the critical paths are taken and then they are clustered together according to their primary outputs. Finally, the reduced set of paths is obtained. That is, two numbers (#PO=2) of clusters with the total of 6 paths. The comparison chart for path reduction for the reference circuits are given in Fig.2.

Fig.2.Comparison chart for path reduction

V.CONCLUSION

The clustering based critical path reduction is achieved in this paper. The proposed two algorithms are discussed and implemented for the set of combinational circuits. The results are also compared with ISCAS85 benchmark circuits. The maximum reduction for the critical path is about 84% and it is achieved for the C499 benchmark circuit. Future work is related to the critical path reduction in sequential circuits and the speed characteristics of the same.

REFERENCES

[1] Ranford Hyman, Nagarajan Ranganathan, Thomas Bingel and Deanne Tran Vo, A clock control strategy for peak power and RMS current reduction using path clustering, IEEE transaction on Very Large Scale Integration systems, 2011.

[2] Wuudiann ke, Shared Seth and Bhargah B.Bhattacharya, A fast fault simulation algorithm for combinational circuits. 20th IEEE international symposium on defect and fault tolerance in VLSI systems, 2005.

[3] Prema chandran menon, Ytzhak Levendel and Miron Abramovici, A critical path tracing algorithm for synchronous sequential circuits, IEEE transactions on CAD, 1991.

[4] E.Musoll and J.Cortadella, Scheduling and resource building for low power, 1995.

[5] M.Abramovici, P.R.Menon and D.T.Miller, Critical path tracing-an alternative to fault simulation, IEEE 20th design automation conference, 1983.

[6] David Bohme, Felix Wolf, Bronis R.de supinski, Martin Schulz and Markus geimer, Scalable critical path based performance analysis, IEEE 26th international parallel and distributed processing symposium, 2012.

[7] A.Bosio, P.Girard, S.Pravossoudovitch, P.Bernardi and M.Sonza Reorda, An exact and efficient critical path tracing algorithm, Fifth IEEE international symposium on electronic design test applications, 2010.

[8] L.Hamzaoglu and J.H.Patel, Test set compaction algorithms for combinational circuits, Proc of CAD, 1998.