TEST Engineering & Management

January-February 2020
ISSN: 0193-4120 Page No. 12545 - 12549

Implementation of VLSI Routing Technique forReal Time Circuits

Chiranjeevi G N, Asst prof, Dept. of ECE, PESIT Bangalore South Campus, Bengaluru, India
Rohit Hariharan, Dept. of ECE, PESIT Bangalore South Campus, Bengaluru, India
Jai Krishna Kumar,Dept. of ECE, PESIT Bangalore South Campus, Bengaluru, India
Veena Chandroji Rao, Dept. of ECE, PESIT Bangalore South Campus,Bengaluru, India

Article Info
Volume 82
Page Number: 12545 - 12549
Publication Issue:
January-February 2020

Article History
Article Received*: 18 May 2019*
Revised*: 14 July 2019*
Accepted*: 22 December 2019*
Publication*: 24 February 2020*

Abstract:
India being the largest democracy on Earth and the fastest growing economy is still in the mushrooming stage of Responsible investing. Past studies attribute this state to the lack of awareness on such investments among investors and to the uncertainty on Socially Responsible Investing (SRI) performance and the financial returns the investment could generate. As a first step towards clearing the apprehensions with SRI this study has been undertaken. The primary objective is to analyse the performance of Indian SRI stocks via the Sustainability indices of BSE. Past nine years (2010-2018) historical data of five indices from the Indian stock market is used. Three among them BSE 500, BSE SENSEX, NSE NIFTY are conventional indices and the other two BSE CARBONEX and BSE GREENEX are Sustainability indices. BSE 500 represents the market portfolio i.e., used as a benchmark for all other indices. While BSE Sensex and NSE Nifty represent the general stock portfolio, BSE Carbonex and BSE Greenex represent the Socially responsible investment portfolio. For the risk-free rate of return, yield of 364 days T-bills is used. The analysis has been carried out in three sections. First, the performance of SRI stocks against the General stocks with their average monthly return. Second, risk and return measures namely Sharpe, Treynor, Jensen's alpha and Sortino have been employed to study risk associated return of the SRI stocks portfolio. Third, the performance of the SRI stocks during the 2016 Demonetisation period has been studied to ascertain the stability of the investment during an economic crisis. The findings made from the study indicate that the Socially responsible stocks perform similar to the general stocks and do not display any significant variation from the performance of general stocks. The associated risk measured as systematic risk (β) in the study also appears consistent across indices. Even during the crisis period both SRI and conventional stocks behave in the same way.

Keywords: *BSE 500, SENSEX, CARBONEX, GREENEX.*

I. INTRODUCTION

This paper shows the implementation of Lee's Maze routing algorithms using BFS and DFS which is used to find the shortest possible path in lesser time as compared to Lee's Algorithm. It is a grid based routing algorithm.BFS is used to compute the shortest distance between the origin node and the target node in the forward wave propagation path. DFS is used to traverse all shortest paths during the backtracking phase from the target node to the origin node. This basic algorithm has improved both speed and memory requirements. [6]

A. *Wave Propagation Phase*

Lee's algorithm uses BFS method to implement the path tracing in the forward path. Initially we take any one point as the origin in the grid. This origin is marked as 0.The wave propagates in the forward direction towards the target. All the immediate adjacent blocks are marked as 1.Then every grid block adjacent to 1 are marked as 2.This process continues till we reach the target or any grid boundary or any obstacle is reached. We calculate the shortest path between the origin and the target node in this manner if it exists. [1]

January-February 2020
ISSN: 0193-4120 Page No. 12545 - 12549

B. Retrace Phase

The next phase of the Lee algorithm is retrace phase or backtracking phase. In this phase we use DFS algorithm. We start this phase from the target node and move to any adjacent grid block which has a label one less than that of the target node. If the path exists, then this label will be present as all the adjacent code to the target node has a label on less than the target label .Continue this process till we reach the origin node. We may have many shortest paths but we select the one which has the less number of bends. [1][4]

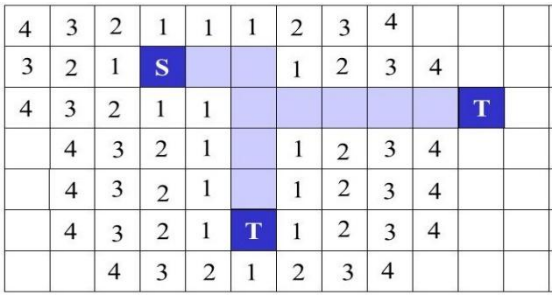

Fig. 1: Example for Lee's Algorithm

II. ALGORITHM

A. FOR WITHOUT OBSTACLE

1. Provide the grid dimension with the coordinates of D0 (m, n) and D1 (x, y) which will give us total grid size.
2. Origin and Target node Inputs: Instantiate origin O (p, q) and Target T (r, s) as coordinates and note that coordinates of origin and target should be less than D0 and D1.
3. Label origin index value O (p, q) as 'j = 0'.
4. Label adjoining cells in row as well in column with incremented value of last origin cell.
   i.e., O (p+1, q), O (p, q+1), O (p-1, q), O (p, q-1) are all taken as 'j+1'.
5. Repeat step 4 until Target is reached, i.e., O (p, q) = T (r, s).
6. After reaching the target start retracing towards the origin to find the shortest path using DFS method.
7. Total path length found is our required output and the traced path is shortest between origin O (p, q) and T (r, s).[2]

B. FOR WITH OBSTACLES

1. Provide the grid dimension with the coordinates of D0 (m, n) and D1 (x, y) which will give us total grid size.
2. Starting and Target node Inputs: Instantiate origin O (p, q) and Target T (r, s) as coordinates and note that coordinates of origin and target should be less than and D0 and D1.
3. Obstacle node Inputs: Provide Coordinates for Obstacle present in the grid, OBS1 (X1, Y1), OBS2 (X2, Y2) and So on.
4. Label origin index value O (p, q) as 'j = 0'.
5. Label adjoining cells in row as well in column with incremented value of last origin cell.
   i.e., O (p+1, q), O (p, q+1), O (p-1, q), O (p, q-1) are all taken as 'j+1'.
6. If obstacles coordinates is matched with incremented value skip that and continue.
7. Repeat step 4 and 6 until Target is reached, i.e., O (p, q) = T (r, s).
8. After reaching the target start retracing towards the origin to find the shortest path using DFS method.
9. Total path length found is our required output and that traced path is shortest between origin O (p, q) and T (r, s).[3]

III. RESULT ANALYSIS

Here we have taken different routing constraints in consideration such as size and type of blocks, shape of obstacle and the number of target and origin node. We have simulated the Verilog codes for the above mentioned constraints with two algorithms as with and without obstacles.

- The position of origin node is (p, q).
- The position of target node is (r, s).

A. Without Obstacle

1. One origin and one target

January-February 2020
ISSN: 0193-4120 Page No. 12545 - 12549

This is the first case study in which algorithm is simulated for the positions of origin O (p, q) =O (1, 1) and destination T (r, s) =T (8, 8) and finally the length is calculated as Length=14.

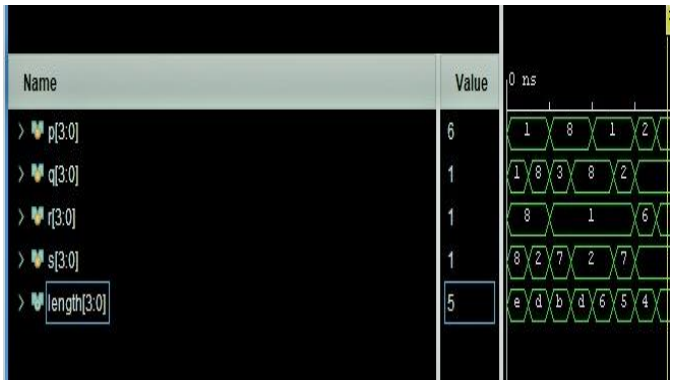

Fig. 2: Result for without obstacle one origin and one target.

From the Fig 2 simulation result is extracted using Xilinx Vivado simulator.

Inputs: Origin node: p = 1, q = 1
Target node: r = 8, s = 8
Output: Path length: Length = 14

2. One origin and many targets

In this case, we have considered one origin node and many target node (no. of target node=2). Here the algorithm first routes for the target which has the shortest length from the source and so on for multiple targets taking the other targets as the origin (If the length is less than the calculated length).

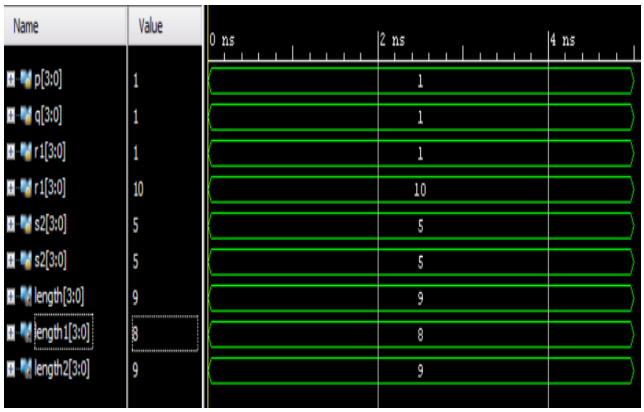

Fig. 3: Result for one origin and many targets.

From fig 3, simulation results are extracted for single origin and multiple target nodes.

Inputs: Origin node: p = 1, q = 1
Targets nodes: r1 = 1, s1 = 5, r2 = 10, s2 = 5
Outputs: Path length: length = 9, length1 = 8, length2 = 9

3. Many origins and Many Targets

In this case, we have considered many origin node (no. of origin node=2) and many target node (no. of target node=2).It also simulates in the similar way as that of one origin node and many target node. It finds the shortest path from any source to target or from any target to target considering any one node as the origin node.

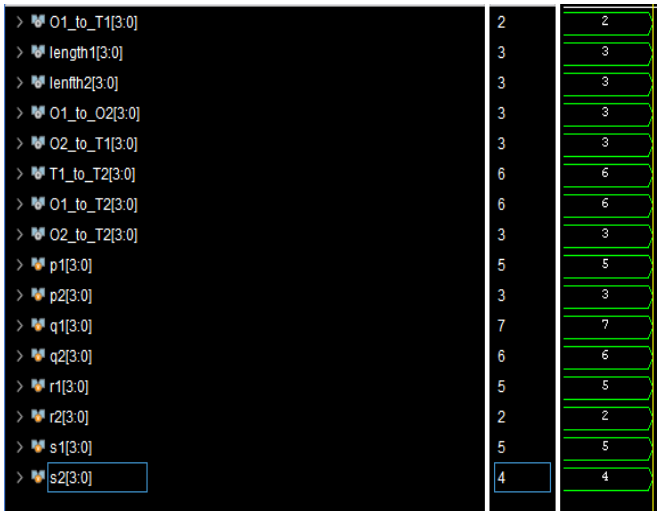

Fig. 4. Result for many origin and many targets.

From fig 4, simulation results is extracted for multiple origins and multiple targets

Inputs: Origin node: p1 = 5, q1 = 7, p2=3, q2=6
Targets nodes: r1 = 5, s1 = 5, r2 = 2, s2 = 4
Outputs: Path length: length1 = 3, length2 = 3, length2 = 9

B. With Obstacle

In this case, we consider the obstacles as the constraints. Some grid blocks are marked as the obstacles which are user defined. We then change the size and shape of the obstacle, which is divided into symmetrical and asymmetrical blocks.

January-February 2020
ISSN: 0193-4120 Page No. 12545 - 12549

- Symmetrical block: The obstacle is said to be symmetric if the obstacle has equal number of rows and columns.
- Asymmetrical block: If the obstacle has different number of rows and columns then the obstacle is said to be asymmetric. The diff asymmetric shapes are H, I and L.[2]

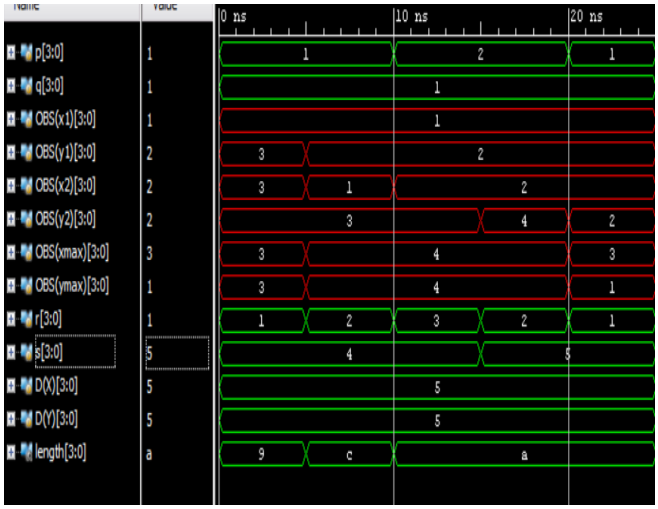

Fig. 5. Result for obstacle of symmetric type and asymmetric type like H, I, L Shapes.

TABLE I. EXAMPLE FOR SYMMETRIC AND ASYMMETRIC AS OBSTACLE
[7-8].

Column	Shapes	Origin	Target	Length
1.	Asymmetric	(1,1)	(1,4)	9
2.	"L"	(1,1)	(2,4)	C
3.	"I"	(2,1)	(3,4)	A
4.	"H"	(2,1)	(2,5)	A
5.	Symmetric	(1,1)	(1,5)	A

From the Fig 5, simulation results are extracted for symmetric obstacle in shape of square of four cells (m1, n1, m2, n2). The algorithm detects the obstacle and propagates towards the destination and gives the minimum path length.

Inputs: Origin node: p = 1, q = 1
Target node: r = 1, s = 4
Obstacle nodes: x1 = 1, y1 = 3, x2 = 3, y2 = 3, xmax = 3, ymax = 3
Output: Path length: length = 9

IV. CASE STUDY

This algorithm is going to work for real time circuits. As an example, switch level Inverter using PMOS & NMOS is constructed with PR Boundary of total size (,) using Cadence tool.

To establish a connection from Source of the PMOS to the Vdd (power line) with the following coordinates as (p, q) for Source and (r, s) for Vdd line.

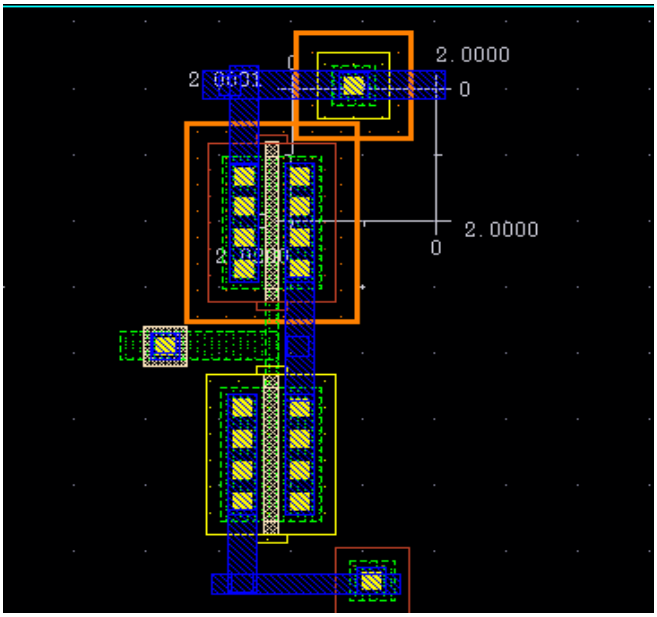

Fig. 6. Layout of Inverter (Real Time circuit)

From fig 6, the coordinates are extracted to find the shortest length. The algorithm takes these coordinates as the input and gives the shortest length as the output.

Inputs: Origin node: p =4, q = 6
Target node: r =4, s = 8
Output: Path length: length =2

V. SYNTHESIS RESULTS

Logic synthesis is the process of converting a RTL(Register Transfer Level) code to a technology specific gate netlist based on some inputs .This process is performed by a tool called synthesis tool which accepts the RTL code and some additional

January-February 2020
ISSN: 0193-4120 Page No. 12545 - 12549

inputs which are the target library and the constraints file and gives the gate level netlist as the output.
The RTL code is written in a Hardware Description Language (HDL) -Verilog. We have synthesized the algorithm to obtain different parameters such as area, power and temperature analysis for the total system designed. These results are then compared for different families of FPGA's such as Basys3 Board and Z-Board.

TABLE II. TABULATION OF AREA AND POWER FOR DIFFERENT FAMILIES

Family	**Constraints**	**Area (number of LUT's)**	**Power(W)**
Basys3 Board	One to one	64	2.071
	One to many	167	6.367
	Many to many	265	22.587
	With obstacles	103	3.750
Z-Board	One to one	64	2.102
	One to many	167	6.805
	Many to many	264	22.692
	With obstacles	146	4.463

VI. CONCLUSIONS

We have presented this paper on implementing Lee's routing algorithm using BFS and DFS for different routing constraints to achieve minimum area, minimum power and minimum length which is verified for different FPGA families.
The purpose of this algorithm is to find the shortest path for any real time circuit without actually calculating it.This algorithm will perform the required operations with the help of some inputs. This will lead to reduction in time and memory. This leads to an increase in the speed of approximately 50 times as compared to LEE's algorithm.

ACKNOWLEDGMENT

This research has been made possible due to the support and continuous guidance of many people. We want to express our heartfelt gratitude to our project guide Chiranjeevi GN, Assistant Prof. for his guidance, encouragement and co-operation throughout this research. We avail this opportunity to convey sincere thanks to all the individuals who have assisted us during the course of the research. We have tried our best to gather all the relevant details subjected to this paper.

REFERENCES

1. C. Y. Lee, "An algorithm for path connection and its application," IEEE Trans.on Electronic Computers, vol. EC-10, September 1961.
2. Vinay Reddy, Chiranjeevi GN, Vinoth, "Implementation of LEE'S Algorithm for different Routing constraints," RTEICT 2018.
3. Naveed A. Sherwani, "Algorithms for VLSI Physical Design Automation," 1999.
4. Xingu Deng, Yangguang Yao, Jia Chen, Yufeng Lin, "Combining Breadth-First with Depth-First Search Algorithms for VLSI Wire Routing," 2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE), 20-22 Aug 2010
5. P. Michael Preetam Raj, "Sliced BFS," International Refereed Journal of Engineering and Science (IRJES), Volume 5, Issue 2, February 2016
6. Baia Srinivasa Baia Srinivasa, Nallasamy Mani, "OPTIMISING ZIG-ZAGS IN MAZE ROUTING ALGORITHM," SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics, 14 Oct. 1998
7. F. Curatelli and P. Antognetti, "A reconfigurable wiring algorithm for three-layer maze routing," Integration, Volume, August 1989