International Journal of Advance Research in Science and Engineering
Vol. No.5, Issue No. 03, March 2016
www.ijarse.com
IJARSE
ISSN 2319 - 8354

ROUTING IN REPEATED AND SYMMETRIC FLOORPLANS IN VLSI LAYOUTS USING QUANTUM ALGORITHMS

Prakash Sahni[1], Shiroman Prakash[2], Huzur Saran[3]

[1]Research Scholar, Faculty of Engineering, Dayalbagh Educational Institute, Dayalbagh, Agra

[2]Assistant Professor, Faculty of Science, Dayalbagh Educational Institute, Dayalbagh, Agra

[3]Professor and Head, Deptt of Computer Science, IIT Delhi

ABSTRACT

In this paper we define the problem of routing in a repeated and symmetric floorplan in a VLSI layout. We use quantum algorithms employing Grover's search algorithm on an entangled database to show that the searching for all solutions which would take runtime $O(2^n)$ on a classical computer takes O(sqrt (n/k)) on a quantum computer in the best case, where n are the number of possible edges, k the number of possible solutions.

Keywords: Quantum Algorithms, Entanglement, Grovers' algorithm, VLSI, Routing

I. INTRODUCTION

The class of computational problems solving which requires more than polynomial time is of special interest for theoretical and practical reasons. Quantum computing which exploits quantum physical effects to perform computation has been shown to provide polynomial time solutions with a bounded error probability to problems for which only exponential or super-polynomial solutions are known using classical computation.

This class of problems is generalized in group theory as the Hidden Sub-group Problem. The techniques involve applying Fast Fourier Transform methods to finite groups. Shor's algorithm for factoring and discrete logarithms, period finding and order finding are all instances of the hidden subgroup problem for finite abelian groups. Abelian groups are isomorphic to the additive groups over the integers in modular arithmetic.

Grover's quantum search algorithm reduces the time of the search in space of size N with M solutions to O (sqrt (N/M)), given an oracle G in an unstructured database.

We combine ideas from both the above techniques and apply to a real world problem.

We specify the problem of determining a path in a VLSI layout in which there a large number of repeated components. In the floorplanning stage, there are additional constraints which apply to the layout due to design considerations and we work on only a specific type of netlist and layouts which are highly repeated and symmetric. See figures 1, 2, 3.

International Journal of Advance Research in Science and Engineering
Vol. No.5, Issue No. 03, March 2016
www.ijarse.com
IJARSE
ISSN 2319 - 8354

Netlist 1

```
module modA(in)
input in;
end module

module modB(in)
input in;
end module

module modC(in)
input in;
end module

module TOP(in)
input in;
modA A(.in(in));
modB B1(.in(in));
modB B2(.in(in));
modC C1(.in(in));
modC C2(.in(in));
modC C3(.in(in));
modC C4(.in(in));
end module
```

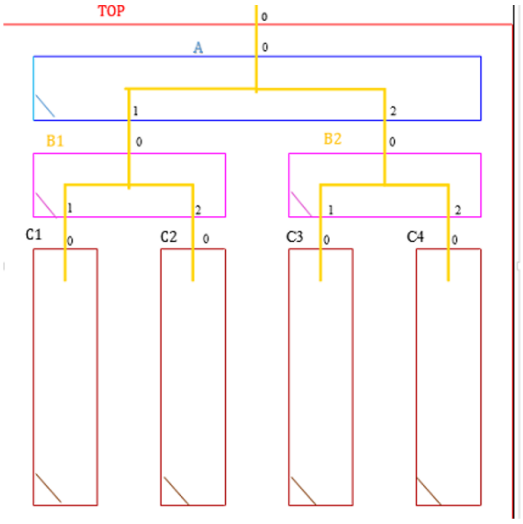

Figure 1 Sample netlist and floorplan

Netlist 2

```
module modX(in)
input in;
end module

module modY(in)
input in;
end module

module modZ(in)
input in;
end module

module W(in)
input in;
modX X(.in(in));
modY Y1(.in(in));
modY Y1(.in(in));
modZ Z1(.in(in));
modZ Z2(.in(in));
modZ Z3(.in(in));
modZ Z4(.in(in));
end module
```

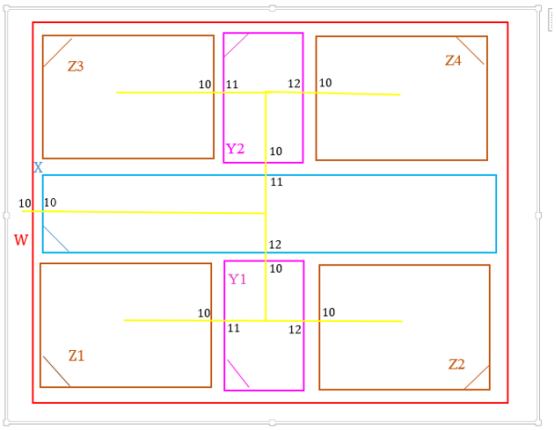

Figure 2. Sample netlist and floorplan

Tree for floorplan 1

Top-0 to A-0
A-0 to A-1
A-0 to A-2
A-1 to B1-0
A-2 to B2-0
B1-0 to B1-1
B1-0 to B1-2
B2-0 to B2-1
B2-0 to B2-2
B1-1 to C1-0
B1-2 to C2-0
B2-1 to C3-0
B2-2 to C4-0

Tree for floorplan 2

W-10 to X-10
X-10 to X-11
X-10 to X-12
X-11 to Y1-10
X-12 to Y2-10
Y1-10 to Y1-11
Y1-10 to Y1-12
Y2-10 to Y2-11
Y2-10 to Y2-12
Y1-11 to Z1-10
Y1-12 to Z2-10
Y2-11 to Z3-10
Y2-12 to Z4-0

Figure 3. Routing tree

II. METHOD

In general the floorplan can be divided into a uniform grid. The points adjacent to each other can be connected with a wire. In general if the grid is of size N*N, there can be 4^N edges over which a wire can be laid. Let this number be M. Then there are 2^M edges, which is equal to the number of combinations over which wires can be laid. Of course the valid configurations are the ones in which all the components of all nets are connected.

If these M states are populated in a Grover search database, then it reduce the search time from M to sqrt (M/K), where K is the number of possible configurations. The Grover oracle needs to decide whether the nodes which constitute the ports belonging to a net in the Verilog netlist are connected in the graph which can be implemented using a Breadth First Search in polynomial time on a classical computer.

However, only some of these configurations are valid because they violate the design rules.

To filter these bad configurations from the database and to further reduce the size of the search space we entangle the edges which cannot exist independently. Each bit represents an edge.

See figure 4:

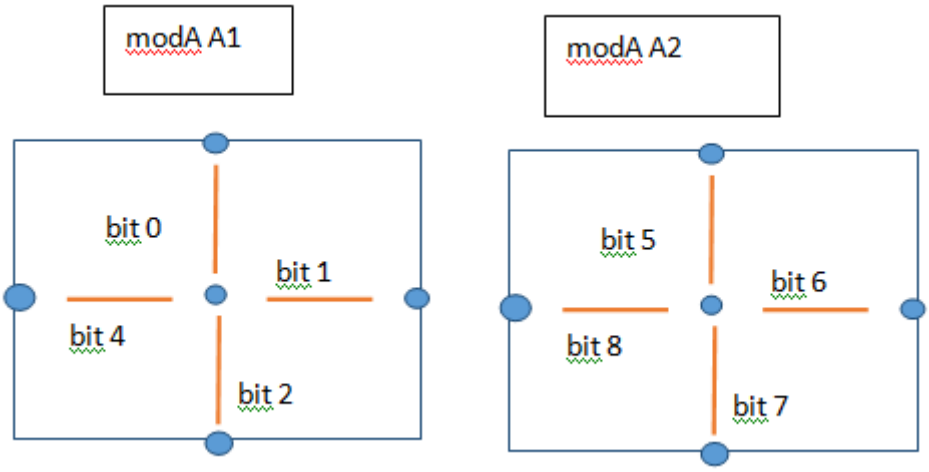

Figure 4. Edges or wires whose bits need to be entangled

There are 8 bits, which means 256 configurations are possible, but due to design configuration only 4 configurations are possible as possible.

|00010001> + |00100010> + |01000100> + |10001000> with amplitude normalized.

This will reduce the size of the database by a factor of the number of repeated components in the floorplan. For the type of floorplans shown in the figures which are common in multi-core cpus (figure 2) and gate arrays (figure 1), there is a high amount of repeatedness which results in an exponential speedup.

III. CONCLUSION

We have shown how the known algorithms in Quantum Computing can be applied to real world problem in VLSI design. For future work, we propose to incorporate the heuristics employed in the real world technique to further speed up the search in the current theoretical framework.

REFERENCES

[1]. Nielsen, M.A., and Chuang I.L., 2000, Quantum Computation and Quantum Information, Cambridge University Press

[2]. Sahni, Vishal, 2007, Quantum Computing, Tata McGraw Hill

[3]. Deo, Narsingh, 1974, Graph Theory with Applications to Computer Science and Engineering, Prentice Hall of India

[4]. McCoy Neal, and Berger, Thomas, 1977, Algebra: Groups, Rings and other Topics, Allyn and Bacon

[5]. Oppenheim, Alan and Schafer, Ronald, 1989, Discrete-Time Signal Processing, Prentice Hall

[6]. Hopcroft, J.E., and Ullman, J.D., 1979, Introduction to Automata Theory, Languages, and Computation, Narosa

[7]. Jozsa, Richard, 2000,Quantum factoring, discrete logarithms and the hidden subgroup problem , arXiv:quant-ph/0012084v1

[8]. Lomont, Chris, 2004, The Hidden Subgroup Problem - Review and Open Problems, arXiv:quant-ph/0411037

[9]. László Babai, 2015, Graph Isomorphism in Quasipolynomial Time, arXiv:1512.03547v2