Survey on FPGA Routing Techniques

Ms. Nagalakshmi Venugopal
Department of Computer Science and Engineering,
Research Scholar, Anna University of Technology,
Coimbatore, India.
laxmivenugopal@rediffmail.com

R. Manimegalai
Department of Computer Science and Engineering,
Research Supervisor, Anna University of Technology,
Coimbatore, India.

Abstract - Field Programmable Gate Array (FPGA), a programmable integrated circuit, has gained great popularity in the circuit design. Routing is an important part of FPGA design step which determines the routing in horizontal and vertical channels of FPGA. In this paper, a number of routing techniques are reviewed. FPGA routing can be achieved in various approaches like based on geometric routing, and based on Boolean Satisfiability. The algorithms based on geometric routing are proved to be efficient algorithms for FPGA routing. The advantage of Boolean based algorithm is that it finds routing possibility and simultaneously route all nets.

Key words - Boolean Satisfiability, FPGA, Hybrid routing algorithm, Geometric routing algorithm.

I. INTRODUCTION

The Field Programmable Gate Array (FPGA) is configured to design and implement various combinational and sequential digital circuits. It involves with many design flow steps between design and implementation. Routing is the one of the important area to be focused more to reduce the interconnection delay, minimum usage of wire length and run time.

Most of the algorithms discussed in this survey have used symmetric island style FPGA model which is shown in Fig 1. This FPGA model consists of three kinds of elements: Configurable Logic Blocks (CLB here referred as L), which is used to configure the logic of the various digital circuits, Input/Output Blocks (IOB), which connects the CLB with external devices and the Interconnection Resources (IR) such as Connection Box (C) and Programmable Switches (S) which are used to connects the CLBs between them and with external device through IOB.

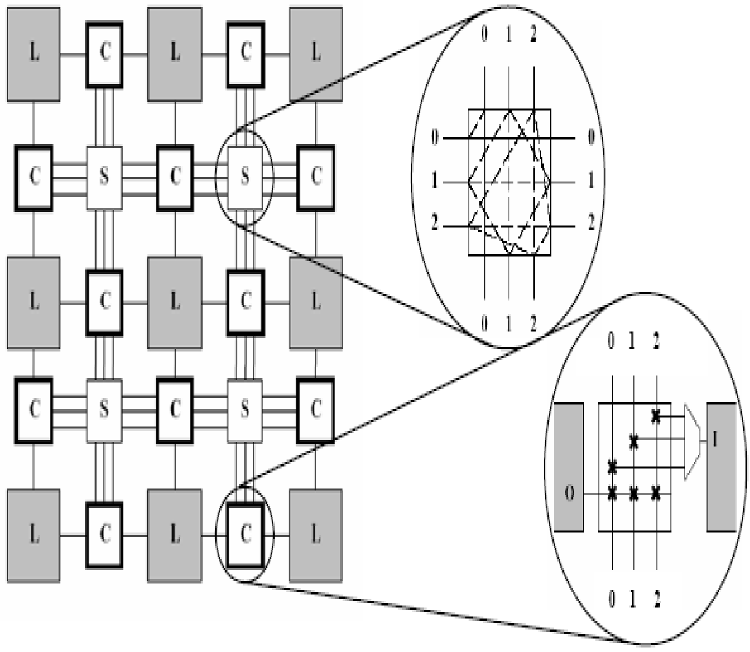

Figure 1. Structure of FPGA

A. FPGA Design Flow

With the evolution of FPGA architecture and increasing complexity of circuit design, CAD (Computer Aided Design) software's have become more matured now when compared to earlier days. Today, most FPGA vendors provide a fairly complete set of design tools that allow automatic synthesis and compilation from design specification in Hardware Description Languages, such as Verilog or VHDL, all the way down to a bit-stream to program FPGA chips. A typical FPGA design flow is shown in Fig 2.

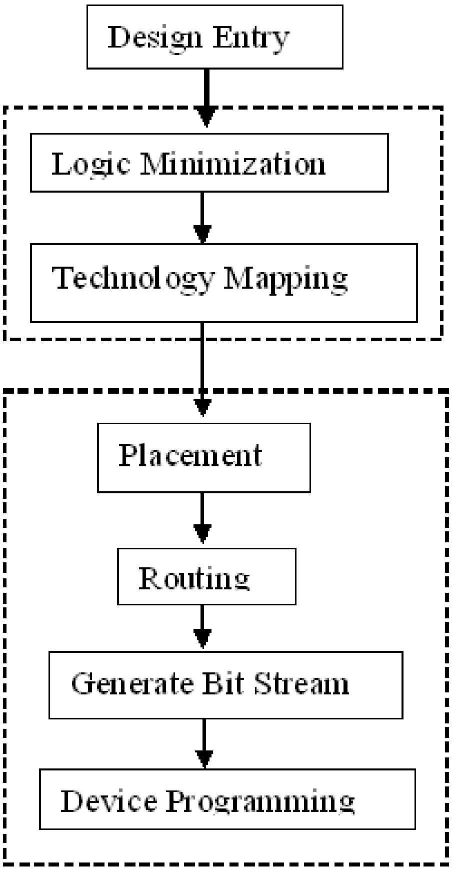

Figure 2. FPGA Design Flow

B. Design Entry

The design entry can be done using: a) HDLs such as verilog and VHDL, b) schematic diagrams, and, c) combination of both a) and b). The schematic input provides more visibility of hardware to designers than HDL input.

C. Synthesis

During synthesis, the circuit which is represented using verilog or VHDL code is translated into netlist. The design hierarchy is analyzed and the netlist is generated for each design element in the circuit. Some amount of design optimization is also done during synthesis.

D. Logic Minimization

Logic minimization is a process by which the redundant logic circuits are identified and removed. Employing the logic minimization there are two approaches namely, Boolean and Netlist approaches. The Boolean based approach analyzes the logic representation used in the circuit and optimizes it. The Boolean methods are very effective in logic minimization and less efficient in execution time. The Netlist method depends on the identification of faults for describing the redundant elements. The existence of redundant circuit elements is equivalent to undetectable faults, means the presence and absence of that does not change the functionality of the circuit.

- Technology Mapping

FPGA technology mapping transforms a network of technology-independent logic gates into one comprised of logic cells in the target FPGA architectures. It divides the whole circuit with logical elements into sub blocks that can be fitted into the FPGA logic blocks. As a result, technology mapping has a significant impact on the quality (area, performance, power, etc.) of the final implementation.

E. Place and Route

The place and route is the next step in the FPGA design flow after technology mapping. Placement algorithm places different partitioned blocks of the net-list on the logic resources of the FPGA. After the blocks are placed in the FPGA, routing algorithms establish connections among those blocks.

F. Device Programming

The device programming is final step of the CAD flow diagram. After the completion of placement and routing, a bit-stream file is generated this is used to configure the FPGA. The device can be programmed through the dedicated pins of configuration ports. The ports can be either parallel or USB port. After the implementation the final verification takes place on FPGA based on various parameters. Completion of successful verification the FPGA design can be migrated as ASIC.

II. ROUTING METHODOLOGIES

There are various routing approaches available for FPGA routing. Some algorithms works with rip-up and re-route, based on shortest path, based on negotiation and based on Boolean Satisfiability.

A. FPGA Routing - based on Geometric Routing

The symmetrical FPGA has restricted resource for routing the CLBs. Routing is a NP hard problem which is separated in two phases, a global routing and a detailed routing. The geometric routers like CGE [1], SEGA [2], PathFinder [3], VPR [6] uses the global and detailed routing. These algorithms uses the ripped up and re-route and negotiation approach to route the given layout. The geometric routing algorithms are efficient even though it has several short comings. The basic routing algorithms has the limitations like dependence on net ordering, conventional one-net-at-a time routing is unable to predict the routability if a problem has tight routing constraints.

- Coarse Graph Expansion (CGE)

In earlier days FPGA routing is done by employing two steps namely, global routing and detailed routing. In general, detailed routing is employed after global routing is done. A routing algorithm called CGE (Coarse Graph Expansion) for symmetrical FPGAs is proposed by Stephen Brown et al [1]. In CGE, the global router divides multiple nets into two point connections. The detailed routing is done in two phases. In the first phase, it enumerates a number of alternative solutions for the detailed route of each coarse graph. It expands each coarse graph and records the subset of possible ways to find suitable path for the connection. For each alternative solution a cost function is calculated and recorded. In the second phase, viewing all the alternative solutions, it [1] makes specific choices for each connection. The decisions made in second phase are driven by the cost function that is computed in first phase.

This algorithm selects the path based on cost if the path corresponds to a time critical connection. The CGE expands the coarse graph to enumerate all paths between the logic blocks which increase the complexity. The complexity of the expanded graph is controlled by pruning algorithm. The maximized alternate paths are eliminated by pruning algorithm which are abandoned permanently, if any routing conflicts occurs, CGE revokes the expansion process to obtain the new set of paths if exists.

- SEGment Allocator (SEGA)

Guy G. Lemieux and Stephen D. Brown has proposed [2] SEGA (SEGment Allocator) routing algorithm for symmetrical FPGAs with segmented routing. The SEGA routing algorithm addresses three issues namely: a) achieving the complete routing of the required connections, b) preserving long wire segments for long connection, and c) avoiding the usage of long wire segments in short connection.

The detailed routing is done in SEGA as two phases. The detailed router observes the wire segments and routing switches available in FPGA and then enumerates all the alternatives for each course graph in phase 1. The decision is taken by considering all the alternatives in phase 2. The cost function is evaluated for the selected connection in phase 2.

The experiments are performed based on the channel density parameter. The MCNC benchmark circuits are used for the experiment. The results are compared with CGE and Maze routers. The SEGA router uses less number of tracks than CGE and Maze routers.

- PathFinder

The complete routing of all signals is difficult in FPGA because of lack of routing resources. It can be achieved by constructing minimum routing trees for each signal and there by using minimum routing resources. This reduces the demand for routing resources of FPGA and some signals will compete for the same resource. Larry McMurchie and Carl Ebling have proposed [3] a negotiation approach, PathFinder, which allocates resources to all signals. PathFinder uses an iterative approach to route all the signals. Routability is achieved by forcing signals to negotiate for a resource and there by determine which signal needs the resource most. It minimizes the delay by allowing the critical signals in this negotiation.

PathFinder consists of two routers a signal router and global router. A signal router routes a signal at a time using shortest path algorithm. Global router achieves the complete routing by calling signal router for all routes considering the resource cost.

PathFinder performs experiments on two different architectures, Triptych and Xilinx. The PREP bench mark circuits are used in Triptych architectures, which results small degradation of critical path delay for a given placement. This algorithm minimizes the congestion however meeting delay constraints. The circuits are selected from ISCAS93 benchmark for Xilinx architecture. The proposed algorithm is compared with Xilinx tools. This algorithm produces a higher degree of routability as well as 11% shorter critical path than commercial tools.

- Versatile Place and Route (VPR)

A FPGA CAD tool for placement and routing, VPR (Versatile Place and Route), is proposed by Vaughn Betz and Jonathan Rose [6]. The inputs to VPR consist of a technology mapped netlist and a text file describing the FPGA architecture. VPR can place the circuit, or a pre-existing placement can be read in. VPR can then perform either a global route or a combined global/detailed route of the placement. For Placement, VPR uses simulated annealing to provide better placement. VPR's router is based on the Pathfinder negotiated congestion algorithm [3]. Pathfinder [3] algorithm initially routes each net by the shortest path it can find, regardless of any overuse of wiring segments or logic block pins that may result.

Experiments are conducted with various input parameters using with MCNC benchmark circuits. The result of VPR is compared with SPLACE/ SROUTE, SEGA, and TRACER. The outcome of this comparison is that the number of tracks required to place and route the complete circuits are lesser than the other existing tools.

B. FPGA Routing – based on Boolean Satisfiability

Another popular routing approach for FPGA is Boolean Satisfiability. It transforms the complex interactions of the routing constraints as atomic Boolean function. It has two main functions, all paths for all nets are considered simultaneously as routing constraints and to be satisfied simultaneously and the Boolean function is satisfiable if and only if the given design is routable. The Boolean function does not produce satisfying element then it is proved that there is no routing solution exist for the design of the given placement. The detailed routing based on Boolean Satisfiability identifies the routability of the given layout by satisfying the Boolean function.

The Boolean based router produces two types of constraints, Connectivity Constraint and Exclusivity constraint. The connectivity constraint ensures that for each two pin connection should have connectivity path through the sequence of C and S blocks specified by the global router. Each two pin connection should have one connectivity constraint function. The exclusivity constraint guarantees that the distinct nets are assigned with different tracks of routing channel. The routing channel forms one exclusivity constraint function per horizontal/vertical channel.

- FPGA Detailed Routing Approach via Search – Based Boolean Satisfiability

A new routing approach for FPGA detailed routing is proposed as search-based Boolean Satisfiability (SDR) [4]. Island-style FPGA architecture is used for modeling the routing. The assumption is made that every wire is fully segmented, which is each wire segment extent one block only. A signal track is formed by connecting a set of wire segments on the same row/column in vertical/horizontal channels. A net is formed by connecting electrically a set of CLB pins. Then the net is separated into set of two-pin connections which characterize source and sink pairs. Further a two pin connection is divided into a set of horizontal and vertical wire segments. The global router provides a confined routing area for a detailed net which consists of wire segments and routing switches.

The FPGA detailed routing problem is converted to net-to-track (two-pin connection) assignment problem. The index of the horizontal/vertical track in which the one part of net is routed can be indicated by track variable of each net segment. These track variables are used to form a routing constraint function.

The routing constraints are represented in either conjunctive normal form (CNF) or integer linear programming (ILP). In CNF the routing constraints are expressed in Boolean function. These Boolean function models the routability of all nets in the conjunction of the entire connectivity constraint and exclusivity constraint requirement. In ILP the routing constraints are expressed in integer domain. It does not need any encoding schemes for track variables. It requires large number of binary variable which increases the size of ILP problem.

The experimental setup is done on two different scenarios. The first one is to determine whether the circuit is routable or not when a placed and globally routed circuit with target FPGA architecture is given. Another is to find the smallest FPGA architecture when only a placed and globally routed circuit is given. The placement and global routings are generated by VPR. The routability functions are evaluated by the GRASP SAT for Boolean SAT and CPLEX version 6.5.2 for ILP solver. The results of Boolean SAT is compared with ILP routing, the Boolean SAT produces less search CPU time and less number of tracks. The result of Boolean SAT is compared with conventional routers like SEGA, VPR, CGE and etc. SDR is producing better result by achieving routability with fewer tracks than SEGA.

- Hybrid Algorithm for FPGA Routing (F-R-SDR)

A hybrid algorithm means which combines the conventional router Frontier and the Boolean Satisfiability [7]. This approach provides the benefits of Frontier and Boolean Satisfiability. In SAT approach the complex routing constraints are represented as a Boolean function. These Boolean functions are expressed as conjunctive normal form clauses. A set of Boolean equation represented for all paths for all nets are considered and satisfied simultaneously. At the every iteration the Frontier cannot route all the nets then it run the SAT based routing and transforming routing constraints into Boolean function. If the Boolean function is satisfiable the algorithm returns the solution.

The Frontier [5] router for FPGA is based on negotiated A* router. The Frontier cares the original routing of VPR [6] uses breadth-first routing and also supports the depth-first and domain negotiation. So the depth-first based router significantly reduces the space for searching which leads to reducing the run time.

The conventional routers like Frontier, PathFinder mainly depends on net ordering by which all nets are routed. This may leads to collide or block the previously routed nets. The solution for this problem is to combine the conventional router with SAT. At the beginning the net input is given to Frontier algorithm. Then it attempts to route the entire net. If the routing is not successful then it sends to the SAT solver. If the SAT solver produces the assignment solution then the routing is a successful otherwise increase the cost then attempting the routing of the net.

The Hybrid algorithm (F-R-SDR) is compared with Frontier, PathFinder with SDR (Satisfiability Detailed Router – (P-R-SDR)). The VPR 399 is used to generate all the placements for all 20 circuits. These three algorithms are applied to all the 20 circuits. From the result the F-R-SDR save 25% CPU time compared with Frontier and 14% CPU time with P-R-SDR.

- Optimization of FPGA Routing Algorithm (V-sub-PBS)

The technique used in this paper is to overcome the drawback of conventional VPR routing and sub-SAT method (V-sub-PBS) [8]. It employs the PBS (Pseudo Boolean Satisfiability) to overcome the drawback of sub-SAT, which is represented as sub- PBS. The routing constraints are expressed as Boolean formula. A Boolean formula consists of a conjunction of clauses. Each clause represents as a disjunction of literals. The Boolean formula is satisfied if all the clauses are satisfied otherwise unsatisfied if any one clause is not satisfied. Deploying this algorithm reduces the CPU running time.

The VPR can support global or combined global-detailed routing for island FPGA architecture. The advantage of combined router has the ability to change the global routing configuration of a net if the net do not route the detailed routing. Even though having many advantages, VPR has some limitations such as cannot produce convergence solution for nets which have restricted routing constraints.

The detailed routing based on Boolean Satisfiability has the advantage to overcome the limitations of geometric routers. It decides the routability of the given layout by generating routing constraints. The routing constraints are represented by Boolean function as conjunctive normal form clauses. If the Boolean function produces the satisfying assignment then the given layout is routable otherwise not. The Boolean Satisfiability can formulate the satisfiability for all nets of all clauses. It fails to produce satisfiability in only one clause leads to unsatisfiability solution for the Boolean function. So the useful satisfiability solution cannot be used to find the routability. This disadvantage can be overcome by setting the threshold to obtain the routing solution.

The sub-PBS express the routing constraints in PB encoding which is more efficient the representation of CNF because it uses less number of literals. The integration of sub-SAT with PBS provides two types of constraints, connectivity and capacity constraints. The connectivity constraints are encoded to only one variable per track per two pin connection. The number of clauses relate to variables are required to express capacity constraints.

The VPR router can produce the convergence solution after the number of iterations, if the design is routable otherwise it simply repeat iterations without producing convergence solution. By combining VPR and sub-PBS (V-sub-PBS) can provide convergence routing solution. The sub-PBS algorithm is not necessary to run every iteration of VPR algorithm. After generating the sufficient number of detailed routes by VPR algorithm then running the sub-PBS algorithm is more effective.

This hybrid algorithm combines VPR with sub- PBs routing which is name as V-sub-PBs. The result is compare with sub- SAT and VPR. In some circuits the VPR producing better result but if the circuit is unroutable then it runs with iterations. The combine algorithm reduces the number of variable up to 50%, number of clauses to an average 83% and save 56% of CPU running time.

- Optimization of FPGA Routing Algorithm (P-PBS)

Another hybrid algorithm is to combine the PathFinder with pseudo-Boolean Satisfiability. Effective routing algorithm for FPGA is significant to reduce running time, reducing the routing area, reducing the total wire length and improving the performance of the circuit. All the parameters are not achieved simply by the basic routing algorithm. By combining PathFinder with PBS (P-PBS) [9] algorithm, runs both global and detailed routing simultaneously which eliminates the short coming of basic routing algorithm.

The PathFinder derives complete routing with the help number of routing iterations. It route the nets in predefined order using ripped up and re-routed. The cost function is used for each routing node to find the congestion in the routing area; the uncongested area can be used for other net routing. The PathFinder consists of two parts, global router and signal router. The global router achieves complete routing by calling the signal router to route all signals based on resource cost. The signal router uses the shortest path algorithm to route one signal at a time.

The pseudo-Boolean constraint is expressed in inequality form which comprises of set of Boolean literals. These literals can be represented either the variable as true or its complement. The PB formula holds true (1) if satisfied. There are two types of PB solvers. The first type converts the PB constraints into a SAT problem and provides the solution. The second type uses the recent developments in SAT solver which use the PB constraint directly.

The objective of combined algorithm, at each iteration of the PathFinder algorithm, we can enumerate all routing paths explored for each connection until current iteration and examine them all simultaneously via pseudo-Boolean SAT technique. The execution of PBS – based flow is not necessary to run for every iteration of PathFinder algorithm because of the overhead involved in converting the routing problem in to pseudo-Boolean SAT instance. The running of PBS-based routing flow is most effective after a sufficient number of detailed routing are accumulated for each signal.

To show the result, a set of benchmark circuits of standard FPGA are taken. The proposed algorithm (P-PBS) is compared with PBS and PathFinder. The P-PBS has the advantage of geometric and symmetry-breaking technology, which reduces the search path and speeds up solving progress. The hybrid algorithm (P-PBS) predicts the routability of a given circuit soon, which is the effective complement short come of PathFinder in this area.

The Boolean search based detailed router [4] reduces the number of tracks compared with SEGA, and achieves routability of the circuit. The remaining three [7, 8, 9] SAT based routing algorithms reduces the computing time. The newly evaluated computing time is compared with Sub- SAT, PBS and PathFinder.

III. CONCLUSION

A comprehensive survey of FPGA routing techniques that are available in the literature is presented in this paper. Routing for FPGA is a very challenging problem because of the limited routing resource. The geometric routing algorithms are very much efficient in routing one-net at a time. This work is motivated how the Boolean Satisfiability is used to combine with geometric routing algorithm to produce better routing solutions for FPGA. By using Boolean Satisfiability we can able to solve much larger SAT instances to formulate the entire routing problem, entrenching all nets simultaneously, as a single SAT instance.

By considering the results of papers reviewed above Boolean Satisfiability based produces better determination of routability of the circuit, which is the limitation of conventional routers. The hybrid algorithm [7] achieves the reduction in computing time and as well as overcome the limitation of conventional routers. One of such hybrid algorithm [8] is V-sub-PBS, combines with VPR and produces the better results by reducing the number of variables, clauses, and PB constraints.

The combined algorithm is producing the efficient routing solution for FPGA. On the continuing improvements of PBS engines will produce better results for SAT solvers, could give optimal solution for FPGA routing.

REFERENCES

[1] S. Brown, J. Rose, and Z. G. Vranesic, “A detailed Router for Field Programmable Gate Arrays”, IEEE Transactions on CAD, pp 620-628, vol. 11, no. 5, May 1992.

[2] G. Lemieux and S. Brown, “A detailed Router for Allocating wire segments in FPGAs”, ACM/SIGDA Physical Design Workshop, pp 215-226, 1993.

[3] L. McMurchie and C. Ebeling, “PathFinder: A Negotiation-Based Performance-Driven Router for FPGAs”, ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, ACM Press, New York, pp 111-117, 1995.

[4] Gi-Joon Nam, Karem A. Sakallah, Rob A. Rutenbar, “A New FPGA Detailed Routing Approach Via Search – Based Boolean Satisfiability”, IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, vol 21, No. 6, pp 674-684, June 2002.

[5] R.Tessier, “ Negotiated A* Routing for FPGAs”, 5th Canadian Workshop on Field Programmable Devices, Montreal, Quebec, Canada, pp 14-19, June 1998.

[6] Vaughn Betz and Jonathan Rose “VPR: A New Packing, Placement and Routing Tool for FPGA Research”, International Workshop on Field Programmable Logic and Applications, pp 213-222, 1997.

[7] Zhan Liu, Zongguang Yu, Xiaofeng Gu, “A New Efficient Algorithm for FPGA Routing” Second IEEE Conference on Industrial Electronics and Applications, pp 1431-1436, 2007.

[8] Yulan Tang , Hao Gao, Zongguang Yu, “Solution and Optimization of FPGA Routing Algorithm”, International Conference on Future Networks, pp 79-82, 2009.

[9] Yulan Tang, Zhan Liu, Jian-hui Chen, “ Optimization of Pseudo-Boolean Satisfiability Algorithm for FPGA Routing”, International Conference on Information Science and Technology, IEEE, pp 45-49 March 26-28, 2011.