International Journal of Electrical Electronics & Computer Science Engineering
Special Issue - NCSCT-2018 | E-ISSN : 2348-2273 | P-ISSN : 2454-1222
March, 2018 | Available Online at www.ijeecse.com

# FPGA Placement and Routing

Kartikeya Tiwari[1], Abhishek Dadhich[2]
[1]UG Scholar, [2]Associate Professor, Department of CSE, Poornima Institute of Engineering & Technology, Jaipur, India
[1]2015pietcskartikeya@poornima.org, [2]abhishek.dadhich@poornima.org

***Abstract:*** ***Field Programmable Gate Arrays (FPGAs) are semiconductor devices that are based around a matrix of configurable logic blocks (CLBs) connected via programmable interconnects. FPGAs can be reprogrammed to desired application or functionality requirements after manufacturing. This feature distinguishes FPGAs from Application Specific Integrated Circuits (ASICs), which are custom manufactured for specific design tasks. With growing complexity, diverse design objectives, high heterogeneity, and evolving technologies, further, modern FPGA placement and routing bring up many emerging research opportunities. In this paper, we introduce basic architectures of FPGAs, describe the placement and routing problems for FPGAs, and improve the placement and routing problem in FPGA architecture (including three major placement paradigms: partitioning, simulated annealing, and analytical placement).***

***Keywords: FPGA, Heterogeneity, Partitioning, Simulated Annealing, Analytical Placement.***

## I. INTRODUCTION

Field Programmable Gate Arrays famously known as FPGAs is an option for usage of computerized rationale in frameworks. They are pre-assembled silicon chips that can be modified electrically to actualize any computerized outline. FPGAs have turned out to be a standout amongst the most p-opular styles for circuit plans and the quickest developing fragments in semiconductor industry since its first introduction duction in mid-1980s.

*A. FPGA Architecture:* Typically FPGAs include:

- Programmable logic blocks which implement logic functions.
- Programmable routing that connects these logic functions.
- I/O blocks that are connected to logic blocks through routing interconnect and that make off-chip connections.

An established FPGA, as outlined in Figure 1, It contains ten thousand to in excess of a million rationale doors with programmable interconnection. Programmable interconnections are accessible for clients or creators to perform given capacities effortlessly.

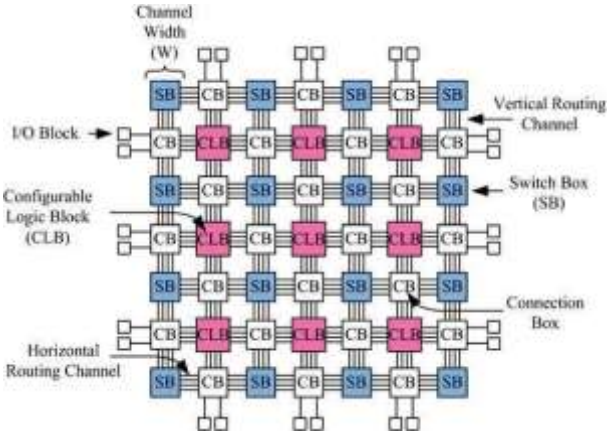


Fig. (a)

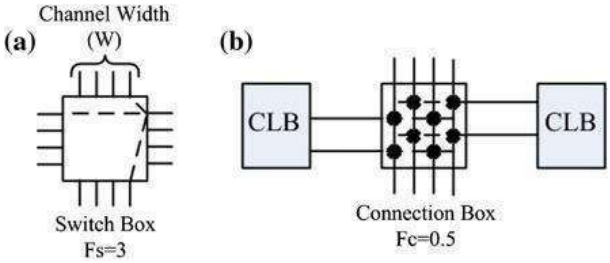


Fig. (b)

Fig. 1. An established FPGA engineering. (a) An island-style FPGA design with three noteworthy parts: rationale modules (CLBs), steering assets with switch boxes (SB), and I/O cells. (b) A case CLB structure. (c) Case of switch and association box.

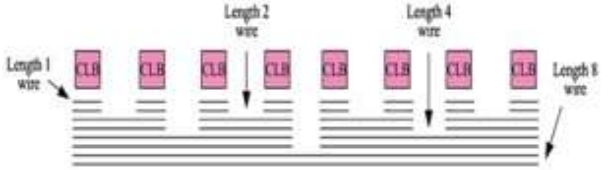


Fig. 2. Channel Segment Distribution

A normal model FPGA chip is appeared in the given figure. There are I/O squares, which are planned and numbered by work. For every module of rationale level arrangement, there are CLB's (Configurable Rationale Pieces). CLB plays out the rationale task given to the module. The interconnection amongst CLB and I/O pieces are made with the assistance of level steering channels, vertical directing channels and PSM (Programmable Multiplexers). The quantity of CLB it contains just chooses the intricacy of FPGA. The usefulness of CLB's and PSM are outlined by VHDL or some other equipment distinct dialect. In the wake of programming, CLB and PSM are set on the chip and associated with each other with directing channels. As

International Journal of Electrical Electronics & Computer Science Engineering
Special Issue - NCSCT-2018 | E-ISSN : 2348-2273 | P-ISSN : 2454-1222
March, 2018 | Available Online at www.ijeecse.com

appeared in Figure 1(b), a CLB essentially comprises of flip-flops (FFs), query tables (LUTs), and multiplexers (MUXs). The FF is utilized to store the state data of a circuit. A LUT is generally a section of static arbitrary access memory (SRAM).

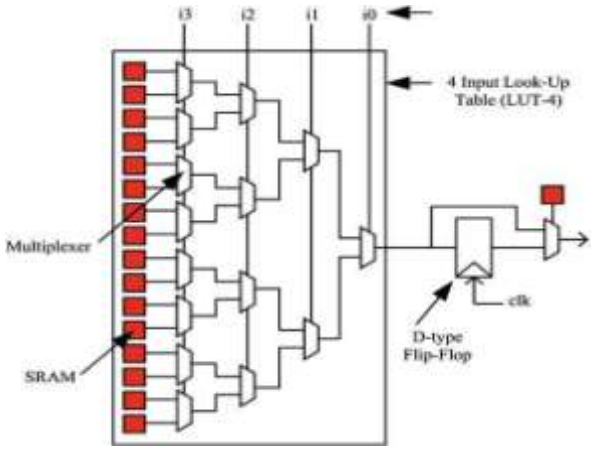


Fig.3. (a). LUT Architecture

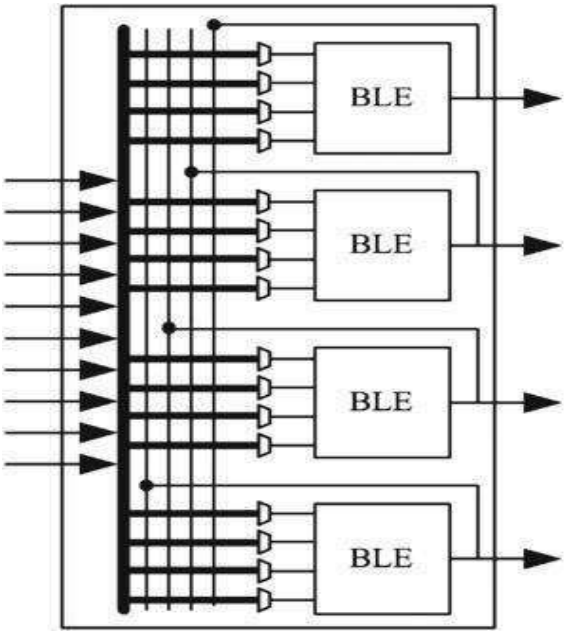


Fig. 3. (b). A Configurable Logic Block (CLB) Having Four BLEs

A CLB can contain a solitary basic logic element (BLE), or a group of privately interconnected BLEs ( as appeared in

Fig. 3 (b). A straightforward BLE comprises of a LUT, and Flip-Tumble. A LUT with k inputs (LUT-k) contains 2k arrangement bits and it can actualize any k-input boolean capacity. Figure 2(a) shows a straightforward BLE involving a 4 input (LUT-4) and a D-type Flip-Tumble. A CLB can contain a bunch of BLEs associated through a neighborhood steering network. Figure 2(b) demonstrates a group of 4 BLEs; each BLE contains a LUT-4 and a Flip-Flop. The BLE yield is available to different BLEs of a similar group through a nearby directing system. The quantity of yield pins of a group are equivalent to the aggregate number of BLEs in a bunch (with each BLE having a solitary yield). Be that as it may, the quantity of info pins of a bunch can be not exactly or equivalent to the entirety of information pins required by all the BLEs in the group.

*B. FPGA Design Process:*

The FPGA design process consists of three major stages (see Figure 3):

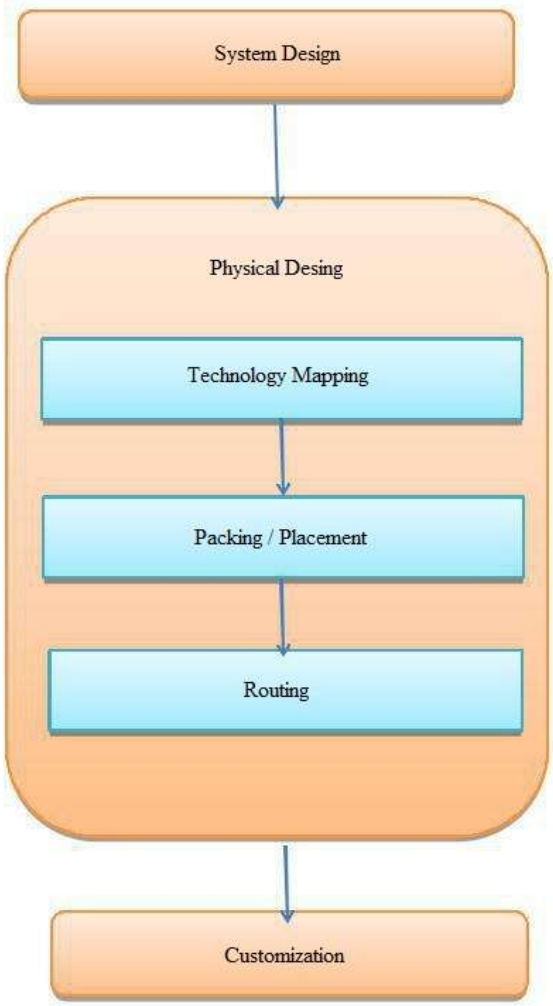


Fig. FPGA Design Process

*1) System Design:* A FPGA-based outline starts with framework configuration, including characterizing formal determinations of a framework and planning framework models and rationale capacities to understand the framework. In this stage, a planned circuit is enhanced by a rationale amalgamation instrument and spoke to by Boolean articulations.

*2) Physical Design:* Physical plan changes over an improved circuit into a geometric portrayal called a format. This procedure is made out of innovation mapping, situation, and directing. An innovation mapper maps a streamlined circuit into an arrangement

of LUTs, FFs, and rationale doors. At that point, LUTs, FFs, and rationale entryways are stuffed into rationale parts, including CLBs, DSPs, and additionally Slams (likewise I/Os). These segments are then set at wanted destinations in a given FPGA by a placer. At long last, a switch doles out wire portions and chooses programmable changes to build the required associations among the rational parts.

*3) Customization:* Upon fruitful directing finish, the yield design is encouraged into a programming unit to modify the FPGA chip.

An advanced FPGA has more limited rationale and steering assets, different structures, stringent outline limitations, and so on. For instance, the construction of directing assets and programmable switches causes serious asset rivalries and in this way forces more limitations on physical plan, particularly the innovation subordinate advances, position (counting pressing) and steering. Further, the assorted variety of FPGA models keeps any single calculation from being connected to a wide range of structures. Subsequently, the format outline (particularly, arrangement and directing) of a FPGA turns out to be considerably more difficult.

The arrangement and directing are characterized as takes after:

*1) Placement:* It is a fundamental advance in electronic plan automation -the segment of the physical outline stream that allocates correct areas for different circuit segments inside the chip's central zone. A substandard situation task won't just influence the chip's execution, however, may likewise make it non-manufacturable by creating unnecessary wirelength, which is past available routing resources.

*2) Routing:* In electronic design, wire directing, usually called simply routing, is a stage in the outline of printed circuit boards (PCBs) and integrated circuits (ICs). It expands on a former advance, called placement, which decides the area of every dynamic component of an IC or segment on a PCB. After position, the directing advance adds wires expected to appropriately interface the put segments while obeying all design rules for the IC.

## II. PLACEMENT

A great variety of algorithms were proposed to solve the FPGA placement problem. These algorithms can be classified into three major categories: partitioning-based method, simulated annealing, and analytical placement.

### *A. Partitioning-Based Placement:*

Throughout the years parceling based arrangement has seen numerous corrections and upgrades, yet the basic system stays much the same.It can be acknowledged as recursively calling the apportioning procedure by picking a locale containing some circuit modules, isolating the district into an arrangement of subregions, and appointing every module to one of the subregions to streamline some predefined metric (e.g., wirelength and cut size). The dividing process is recursively called until the point when the quantity of modules in every area is littler than an edge with the end goal that their position can be fathomed by count or other methods. Management of whitespace (otherwise called free space) is a key issue in physical plan as it profoundly affects the nature of a situation. The measure of whitespace in a plan is the distinction between the aggregate placeable zone in an outline and the aggregate mobile cell zone in the plan.

### *B. Simulated Annealing:*

Simulated annealing is a technique for finding a decent (not really impeccable) answer for an improvement issue. In case you're in a circumstance where you need to boost or limit something, your concern can likely be handled with recreated tempering.

The traveling salesman problem is a decent illustration: the sales representative is hoping to visit an arrangement of urban areas in the request that limits the aggregate number of miles he ventures. As the quantity of urban communities gets expansive, it turns out to be too computationally serious to check each conceivable schedule. By then, you require an algorithm. There are numerous enhancement calculations, including slope climbing, hereditary calculations, inclination plunge, and that's only the tip of the iceberg. Reproduced toughening's quality is that it abstains from getting captured at neighborhood maxima - arrangements that are superior to any others close-by, however, aren't the very best. You can envision this by envisioning a 2D chart like the one underneath. Every x-facilitate speaks to a specific arrangement (e.g., a specific schedule for the businessperson). Every y-facilitate speaks to how great that arrangement is (e.g., the backward of that schedule's mileage).

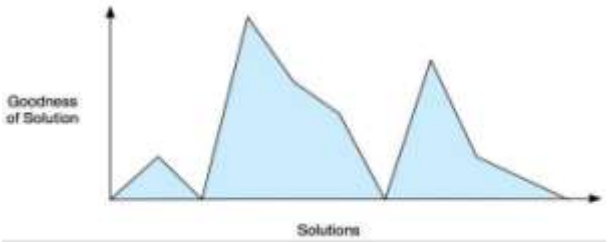


Fig. 4. (a)

Comprehensively, an improvement calculation scans for the best arrangement by creating an arbitrary starting arrangement and "investigating" the territory close-by. On the off chance that a neighboring arrangement is superior to anything the present one, at that point, it moves to it. If not, at that point the calculation stays put.This is superbly sensible, however, it can prompt circumstances where you're stuck at a problematic place. In the chart underneath, the best arrangement is at

the yellow star on the left. Be that as it may, if a straightforward calculation discovers its way to the green star on the right, it won't move far from it: the greater part of the neighboring arrangements are more terrible. The green star is a nearby greatest.

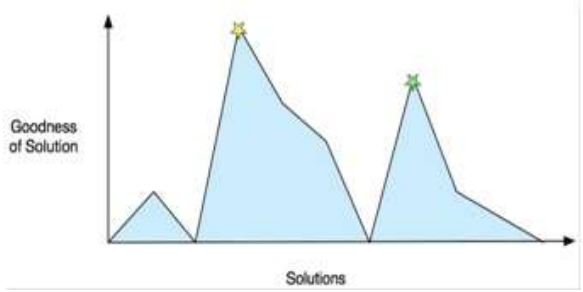


Fig. 4. (b)

Simulated annealing infuses only the perfect measure of haphazardness into things to escape neighborhood maxima right on time in the process without getting off kilter late in the diversion when an answer is close-by. This makes it really great at finding a tolerable answer, regardless of its beginning stage.

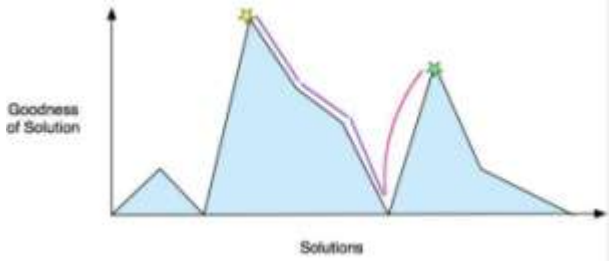


Fig. 4. (c)

*C. Analytical Placement:*

Current FPGA situation, for the most part, comprises of three noteworthy stages: global placement, legalization, and detailed placement. Global placement processes the best position for every module to limit some cost metric (e.g., wirelength), overlooking the module non-covering requirement. Sanctioning at that point evacuates module covers and places the modules into their comparing locales to limit the dislodging from a worldwide situation. Point by point situation additionally refines the arrangement quality. The global position assumes the essential part in deciding the quality and effectiveness of the arrangement process. The global situation is regularly displayed as a wirelength advancement issue, with cover diminishment requirements. In this way, the key issues of worldwide situation lie in the logical models of wirelength, the methods of cover expulsion, and the incorporation and streamlining of target capacities.

 *Wirelength Model:* Half-perimeter wirelength (HPWL) of a net is the most prevalent wirelength demonstrate for a situation. Be that as it may, a numerical definition requires its target capacity and limitations to be differentiable, yet HPWL isn't differentiable all around.

 *Overlap Reduction:* To spread the concentrated overlap modules (due to wirelength minimization), various overlap reduction techniques were proposed. For FPGA placement, density guidance and rough legalization are two common techniques.

 *Integration:* Amid global placement, we regularly need to deal with the synchronous enhancement for numerous targets. Subsequently, it is attractive for an investigative detailing to incorporate these destinations for successful co-improvement.

To accomplish better placement results, a pressing procedure to pack rationale components into CLBs is frequently consolidated into the arrangement as a preprocessor. The general pressing calculation just thinks about the fondness of a rationale module (i.e., the quantity of regular contributions to a rationale module). Be that as it may, overlooking physical data may bring about poor wirelength. The perfect position of a module with high proclivity might be far away. As appeared in Figure 8, LUTs and FFs are pressed into a CLB thinking about their physical areas, not just their liking. By regarding worldwide situation comes about, physical-mindful pressing can lessen the aggregate relocation adequately.

Table 1: Comparisons of the Three Placement Methods

| | Partitioning | Simulated Annealing | Analytical |
|---|---|---|---|
| Runtime | Fast | Slow | Fast |
| Quality | Worst | Good | Best |
| FPGA Architecture | Basic Architecture with CLBs and | Basic, Fault Tolerance, Library Based | Basic, Segmented Wire, |
| Awareness in Existing Works | Uniform Wire Segments | One | Heterogeneity, Physical-Aware Packing, Clocking |

*D. Comparisons:*

Table I summarizes the key strengths and weaknesses of the three types of placement algorithms. The partitioning-based placement is robust and has good scalability, but its quality is typically limited. The SA-based placement often can achieve good quality and apply to various metrics and constraints, but its scalability is often poor. The analytical placement is the most effective and also general for handling various objectives and constraints.

International Journal of Electrical Electronics & Computer Science Engineering
Special Issue - NCSCT-2018 | E-ISSN : 2348-2273 | P-ISSN : 2454-1222
March, 2018 | Available Online at www.ijeecse.com

## III. ROUTING

Routing is a complex combinational improvement issue, particularly for current FPGAs with constrained, discrete directing structures. To deal with the FPGA steering issues, scientists regularly demonstrate a FPGA as a weighted diagram to such an extent that the chart topology can speak to the engineering of the FPGA.The FPGA directing issue comprises in doling out nets to directing assets to such an

extent that no steering asset is shared by in excess of one net. Way discoverer is the present, cutting edge FPGA directing calculation. Way f inder works on a coordinated diagram reflection G(V, E) of the steering assets in a FPGA. The arrangement of vertices V in the diagram speaks to the IO terminals of rationale pieces and the directing wires in the interconnect structure. An edge between two vertices speaks to a potential association between them. Figure 2.24 presents a piece of a steering diagram in a Work based interconnect. Given this chart deliberation, the steering issue for a given net is to locate a coordinated tree inserted in G that associates the source terminal of the net to every one of its sink terminals. Since the quantity of steering assets in a FPGA is restricted, the objective of discovering special, non-meeting trees for every one of the nets in a net rundown is a troublesome issue. Way discoverer utilizes an iterative, transaction based way to deal with effectively course every one of the nets in a net rundown. Amid the principal steering cycle, nets are openly directed without focusing on asset sharing. Singular nets are steered utilizing Di jkstra's most limited way calculation [1.1]. Toward the finish of the primary cycle, assets might be congested in light of the fact that various nets have utilized them. Amid ensuing emphasess, the cost of utilizing an asset is expanded, in view of the quantity of nets that offer the asset, and the historical backdrop of clog on that asset. Therefore, nets are made to consult for directing assets. On the off chance that an asset is profoundly congested, nets which can utilize bring down clog options are compelled to do so.On the other hand, if the options are more congested than the asset, at that point a net may even now utilize that asset. The cost of utilizing a directing asset n amid a steering cycle is given by Eq. 1.1.

$$cn = (bn + hn) \times pn \quad (1.1)$$

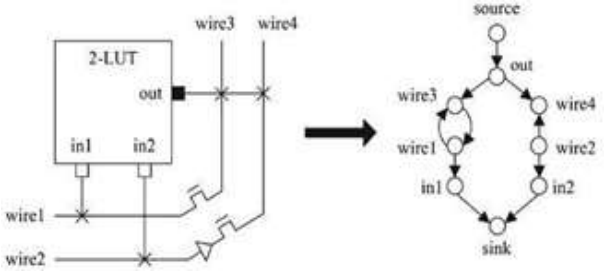


Fig. 4. Modeling FPGA Architecture As A Directed Graph

bn is the base cost of using the resource n, hn is related to the history of congestion during previous iterations, and pn is proportional to the number of nets sharing the resource in the current iteration. The pn term represents the cost of using a share resource n, and the hn term represents the cost of using a resource that has been shared during earlier routing iterations. The latter term is based on the intuition that a historically congested node should appear expensive, even if it is slightly shared currently. The Pseudo-code of the Path finder routing algorithm is presented in algorithm 1.1.

```
Let: RTi be the set of nodes in the current routing of
net i while shared resources exist do

/*Illegal routing*/
foreach net, i do
rip-up routing tree RTi ;
RT(i ) = si ;
foreach sink ti j do
Initialize priority queue PQ to RTi at cost 0; while
sink ti j not found do

Remove lowest cost node m from PQ; foreach
fanout node n of node m do

Add n to PQ at PathCost (n) = cn + PathCost (m);
endfch
endw
foreach node n in path ti j to si do
/*backtrace*/
Update cn;
Add n to RTi ;
endfch
endfch
endfch
update hn for all n;
endw
```

Algorithm 1.1: Pseudo-Code Of The Path Finder Routing Algorithm

An imperative measure of directing quality created by an FPGA steering calculation is the basic way delay. The basic way deferral of a directed netlist is the most extreme postponement of any combinational way in the netlist. The most extreme recurrence at which a netlist can be timed has a converse association with basic way delay. Consequently, bigger basic way defers back off the activity of netlist. Defer data is consolidated into Path finder by rethinking the cost of utilizing an asset n (Eq. 1.2).

$$cn = Ai\,j \times dn + (1 - Ai\,j) \times (bn + hn) \times pn \quad (1.2)$$

The *cn* term is from Eq. 1.1, *dn* is the delay incurred in using the resource, and *Ai j* is the criticality given by Eq. 1.3

$$Ai\,j = Di\,j \;/\; Dmax \quad (1.3)$$

Di j is the greatest postponement of any combinational way experiencing the source and sink terminals of the net being directed, and Dmax is the basic way deferral of the netlist.

Condition 1.2 is planned as an aggregate of two cost terms. The primary term in the condition speaks to the defer cost of utilizing asset n, while the second term represents the clog cost. At the point when a net is steered, the estimation of Ai j decides if the postponement or the blockage cost of an asset rules. On the off chance that a net is a close basic (i.e. its Ai j is near 1), at that point clog is to a great extent disregarded and the cost ofusing an asset is essentially dictated by the postpone term. In the event that the criticality of a net is low, the clog term in Eq. 1.2 rules, and the course found for the net maintains a strategic distance from blockage while conceivably bringing about postponement. Pathfinder has turned out to be a standout amongst the most intense FPGA steering calculations to date. The arrangement based system that exchanges off deferral for clog is an amazingly successful strategy for steering signals on FPGAs.More essentially, Pathfinder is a genuinely design versatile directing calculation. The calculation works on a coordinated diagram deliberation of a FPGA's steering structure, and would thus be able to be utilized to course netlists on any FPGA that can be spoken to as a coordinated directing chart.

## IV. FUTURE RESEARCH DIRECTIONS

Albeit late works have essentially progressed FPGA arrangement and steering, numerous developing difficulties emerge from developing intricacy, assorted destinations, high heterogeneity, and advancing advancements. In this area, we display some potential research bearings for current FPGA situation and directing.

### *A. Mixed-Size Placement:*

IP reuse and library-based executions with presynthesized modules wind up well known in current FPGA plans, which can essentially abbreviate the outline cycle. They both prompt rationale modules like the macros in traditional ASIC outlines. With more confined rationale and steering assets, further, the arrangements of large scale like modules to their comparing locales make this FPGA position issue significantly more troublesome than that for ASICs.

### *B. Simultaneous Analytical Placement and Routing:*

Because of the tight common reliance of arrangement and directing with the uncommon current FPGA structures, it ought to guarantee to investigate such reliance for concurrent situation and steering, particularly under an expository system like the work in light of the fact that such a system has been appeared to be the best for present-day FPGA plans.

### *C. Machine Learning for FPGA Placement and Routing:*

Machine learning (ML) is a rising bland calculation that can give a coveted forecast and answer for some perplexing outline issues. Amid situation, it is regularly hard to foresee the resulting steering practices (e.g., directed wirelength, clog, timing). ML could give some smart expectation and assessment to encourage the FPGA arrangement and directing procedures.

## V. CONCLUSIONS

We have presented the fundamental structures of FPGAs, depicted well known arrangement and routing algorithms for FPGAs, and gave key future research headings to FPGA position and routing.With the expanding plan intricacy, differing outline necessities, high heterogeneity, and new advances, we expect all the more rising difficulties for FPGA situation and directing are yet to come sooner rather than later.

## VI. REFERENCES

[1] Y. Xue, C. Yang, and J. Hu, “Age-aware logic and memory co-placement for RRAMFPGAs,” Proc. DAC, June 2017.

[2] H. Fraisse, A. Joshi, D. Gaitonde, and A. Kaviani, “Boolean satisfiability-based routing and its application to Xilinx ultraScale clock network,” Proc. FPGA, pp. 74–79, Feb. 2016.

[3] W. Li, S. Dhar, and D. Z. Pan, “UTPlaceF: a routability-driven FPGA placer with physical and congestion aware packing,” Proc. ICCAD, Nov. 2016.

[4] Y. Xue, P. Cronin, C. Yang, and J. Hu, “Fine-tuning CLB placement to speed up reconfigurations in NVM-based FPGAs,” Proc. FPL, Sept. 2015.

[5] E. Wegley and Q. Zhang, “Application of specific delay window routing for timing optimization in FPGA designs,” Proc. FPGA, pp. 37–45, Feb. 2015.

[6] S.-Y. Chen and Y.-W. Chang, “Routing-architecture-aware analytical placement for heterogeneous FPGAs,” Proc. DAC, June 2015.

[7] W. Li, S. Dhar, and D. Z. Pan, “UTPlaceF: a routability-driven FPGA placer with physical and congestion aware packing,” Proc. ICCAD, Nov. 2016.

[8] Wikipedia,“What-are-the-simulated-annealing-algorithms-used-for-placement-in-FPGAs-”

[9] Katrina Ellison Geltman, “http://katrinaeg.com/simulated-annealing.html”