Efficient Optimization Technique for FPGA Using PSO

Shobana V
PG Scholar
Dr. N.G.P. Institute of Technology
Coimbatore, India
shobanasudha99@gmail.com

Nagalakshmi Venugopal
Associate Professor
Dr. N.G.P. Institute of Technology
Coimbatore, India
nagalakshmi.ngp@gmail.com

Abstract— The Field Programmable Gate Array (FPGA) is popular medium to develop digital circuits. In FPGA design flow placement plays major role to optimize the wirelength and channel width. Here to minimize the interconnection length between blocks uses Particle Swarm Optimization algorithm. Using this optimization procedure the wirelength and channel width of Microelectronics Center of North Carolina (MCNC) circuits are reduced. And their results are compared with existing techniques.

Keywords-Placement, Particle Swarm Optimization

I. INTRODUCTION

The Field Programmable Gate Arrays (FPGAs) represent one of today's most popular digital logic implementation options. Those are flexible and reusable circuits that can be easily reconfigured by the designer and re-programmability [7]. An important component of the FPGA-based design is the computer-aided design (CAD) software necessary to efficiently use the circuits. A typical FPGA design process consist synthesis, mapping, placement, and routing. After the synthesis phase, the logic specification is divided in the technology mapping phase into logic functions that are directly implemented in the FPGA circuit. In the placement phase, these logic functions are assigned to specific cells of the circuit. The objective of placement is to minimize the total wire length, is NP-complete problem [1]. Finally, in the routing phase, the logic signals are connected by programmable switches.

The typical FPGA architecture is shown in figure 1. The FPGA architecture consist Configuration Logic Block (CLB), Input/Output Block (IOB), Switch Block and Interconnection wires.

Based on the programming technology, FPGAs are mainly classified into three types, namely, Static Random Access Memory (SRAM) based, Fuse based and Flash based. SRAM based FPGAs are commonly used. The CAD flow of FPGA includes various steps such as design entry, mapping, placement, routing and bit stream generation. The responsibility of place-and route tool is to produce a physical implementation of a netlist on the FPGA's hardware.

II. THE PLACEMENT PROBLEM

The process of placement consists of finding suitable physical locations for each cell on the entire layout. The locations are suitable if they minimize given objective functions, performed by designer. The cells may be logical blocks and input/output blocks, etc.

More formally, the placement problem can be defined as follows. Given a set of m modules, $M = \{M_1, M_2, \ldots, M_m\}$, a set of n nets $N = \{N_1, N_2, \ldots, N_n\}$, and a set of p primary input pins and primary output pins $R = \{R_1, R_2, \ldots, R_p\}$, we associate with each module $Mi \in M$ a set of nets , where $i \subseteq N$. Similarly, we associate with each net $Ni \in N$ a set of modules i, where $= \{Mj \mid Ni \in \}$. We are also given a set of locations $L = \{LNMiNMMNMNiNMj_1, L_2, \ldots, L_k\}$, where $k \geq n$. The placement problem is to assign each $M_i \in M$ to a unique location L_j such that some objective function is optimized. Normally each module is considered to be a particle, and if M_i is assigned to location L_j then its position is defined by the coordinate values (x_j, y_j). Sometimes a subset of the modules in M are fixed, i.e., pre-assigned to locations, and only the remaining modules can be assigned to the remaining unassigned locations.

Many heuristic algorithms are used for the placement. A placement is acceptable if 100% routing can be achieved within a given area.

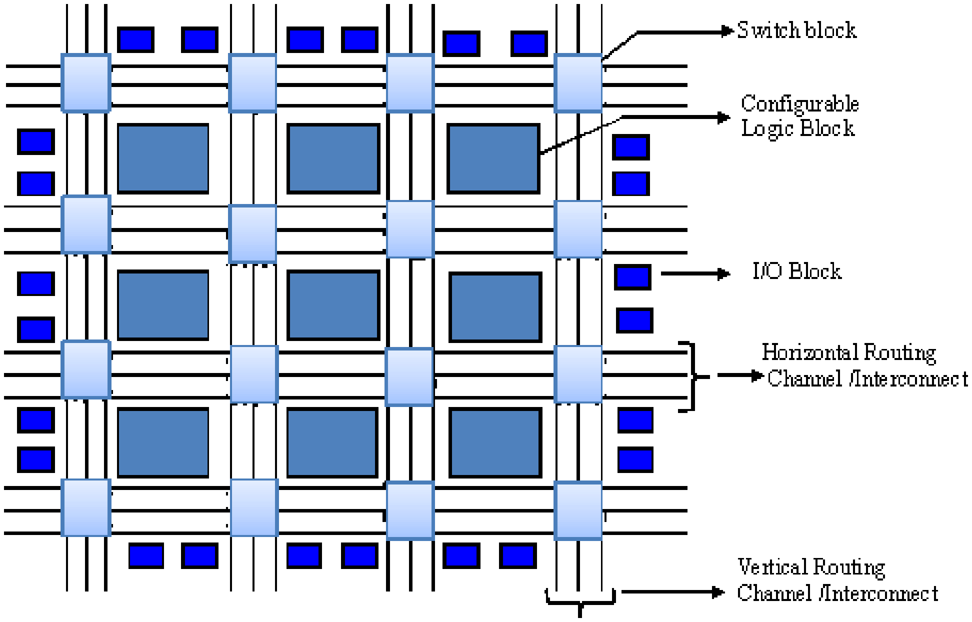
Switch block
Configurable Logic Block
I/O Block
Horizontal Routing Channel /Interconnect
Vertical Routing Channel /Interconnect

Figure 1. FPGA Architecture

III. RELATIVE WORKS

Ant Colony Optimization [2]

In [2] every ant in the colony assigns block *i* to physical location *j*, consequently all the blocks in the circuit netlist will be mapped to the specified physical FPGA array. The optimization goals are to minimize the required wiring, balance the wiring density and maximize circuit speed. To avoid that different blocks would be assigned to the same location, we must construct a data structure, called a tabu list, whose length is the same as the number of blocks, for each ant. After all the ants have assigned logic blocks implementing the circuit to physical locations in FPGA, an effective evaluation is a must for estimating the quality of the current placement solution.

Half-Perimeter Wire Length [3]

In [3] the Minimization of Half-Perimeter Wire length (HPWL) is a commonly used method for circuit placement. [3] Proposes a new mathematical model to approximate the HPWL cost function. Here the error bounds of the new cost function and show several desirable properties of the new approximation model. They use the global and detailed placements produced by the NTU placer on ISPD 2004 benchmark suite to compare the smoothed approximation to two other approximation schemes namely the LogSumExp and CHKS based approximations. The experiments validate theoretical results and their scheme has an average of 5% error in the total wire length.

Ultra Low Temperature Simulated Annealing [4]

In [4] propose a placement method for island style FPGAs, based on fast yet very good initial placement followed by refinement using ultra-low temperature Simulated Annealing [4]. The initial placement is the keystone of the method and the steps to obtain it are [4]: top down coarse partitioning, allocation of partitions to regions on FPGA array [5], placement of logic blocks within each region and finally the IOs [4]. The solutions thus obtained require 66% fewer moves i.e. about 3x speed-ups during final iterative refinement by simulated annealing, whereas the quality of solution is on the average within 2% of optimal. The critical path length obtained after routing does not degrade for the set of 9 benchmark circuits [4].

IV. PROPOSED WORK

Here the Particle Swarm Optimization (PSO) algorithm is proposed for the FPGA placement.

Particle Swarm Optimization

The Particle Swarm Optimization (PSO) algorithm is a population based optimization algorithm. Each particle guides itself to the best possible solution in its neighboring space, also known as personal best (pbest). The pbest can be related to the particle’s cognition of its own history in finding different results when it moves through the space.

Also before taking the next move each particle consider the location of the particle which achieved the best fitness, that's known as global best (gbest). Both the above two factors and the present velocity of the particle affects the velocity in the next iteration. The velocity is added to the present location of the particle to get the next location which will help it move towards the best location (gbest), achieved by the swarm, while still looking for an even better location(improving pbest).

The velocity and positions are calculated by the following equation

$$V_{i+1} = V_i + C_1 * rand_1 ( ) * (pbest_i - X_i) + C_2 * rand_2 ( ) * (gbest - X_i) \quad (1)$$

$$X_{i+1} = X_i + V_i \quad (2)$$

Where rand ( ) = any random in a range of (0,1) generated each time when the function is evaluated. And C_1 and C_2 are two constants known as acceleration constants.

V. EXPERIMENTAL SETUP

This work is implemented by the MATLAB. Matlab is tool which implements the works by matrix. Here the input is Microelectronics Center of North Carolina (MCNC) Benchmark Circuits. This benchmark consist many circuits. Those MCNC benchmark Net files are given as a input particle. Those results are compared in table 1.

It places the particles in random positions. That is known as initial population. Then the particle positions are updated by the velocity of the particles and distance between the particles. The initial population is shown in figure 2 and optimized population is shown in figure 3.

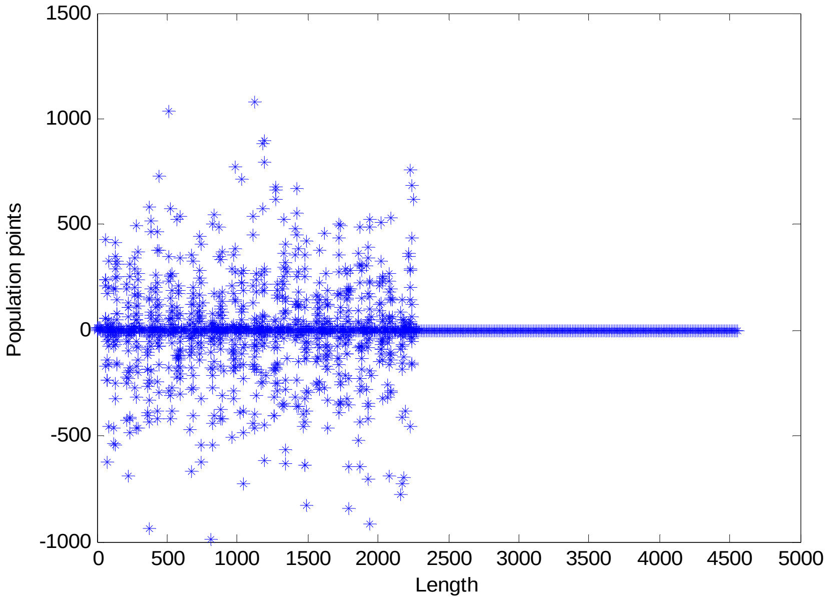

Figure 2. Initial Population

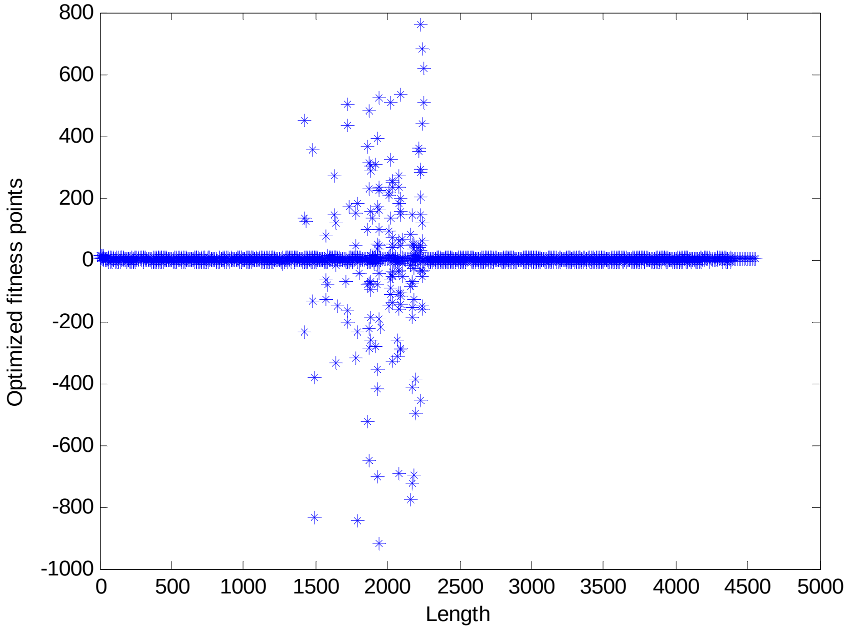

Figure 3. Optimized Population

VI. CONCLUSION

The FPGA placement has been optimized by using the PSO algorithm in order to minimize the wirelength. Those results are compared with the previous works such as Ant Colony Optimization (ACO), VPR, and Simulated Annealing etc.

Circuit	PSO	LTSA [5]	ACO [2]	VPR [6]
Ex5p	3	14	14	14
Apex4	6	13	15	13
Alu4	7	11	10	10
seq	8	12	12	12
Apex2	3	12	11	13
pdc	6	17	18	17
Ex1010	6	13	10	11
spla	7	15	17	15

REFERENCES

[1] Sait, S. M., and Youssef, H., “VLSI Physical Design Automation”, McGraw-Hill Book Company, 1995

[2] Kai Wang, Ning Xu, “Ant Colony Optimization for Symmetrical FPGA Placement”, IEEE, pp 561-563, 2009

[3] BNB Ray, Shankar Balachandran, “ A New Wirelength Model for Analytical Placement’, IEEE, pp 90-95, 2011

[4] M. Xu, G. Grewal, S. Areibi, “StarPlace: A new analytic method for FPGA placement”, Elsevier, Vol. No.44, pp 192-204, 2011

[5] P. Banwrjiee, S. Surkolay, “Faster Placer For Island Style FPGAs”, Computing: Theory and Application, International Conference on pp. 117-121, 2007

[6] V. Betz, J.Rose, “VPR: A New Packing, Placement and Routing Tool for FPGA Research”, in proceedings of the 7th International Conference on Field Programmable logic and Application, pp 213-222, 1997

[7] Sang Joon Lee , Dr.Kaamran Raahemifar, “FPGA Placement Optimization Methodology Survey”, Canadian Conference on Electrical and Computer Engineering, pp 1981-1986, 2008