Placement in FPGA Using Hybrid PSO-SA Technique

Tarun Bali
Department of ECE,
NIT Jalandhar, India

Dr. Mamta Khosla
Department of ECE,
NIT Jalandhar, India

Naveed Anjum
Department of ECE,
NIT Jalandhar, India

Abstract

Field-Programmable Gate Arrays (FPGA) provide a means for fast prototyping and a cost-effective chip design. Their reconfigurable nature, low cost of implementation, and shorter time needed to realize a given design has made them a viable implementation alternative for larger and complex designs. Placement is a critical step in FPGA design because the quality of placement determines overall performance of the logic implemented in the FPGA. In this work, a new hybrid PSO-SA algorithm is proposed for solving the placement problem in FPGA. The performance of Synchronous FIFO circuit is studied by using Simulated Annealing (SA), Particle Swarm Optimization (PSO) and the proposed PSO-SA technique. It is found that the proposed hybrid technique gives better placement solutions when compared with those obtained from SA and PSO. This work considers the wire length driven placement objective. The Xilinx CAD tool is used for generating the netlist. The information obtained from the netlist is given to placement algorithms for obtaining optimized placement with minimum value of wirelength cost function.

Keywords—*FPGA, Placement, PSO, SA, Hybrid PSO-SA*

1. Introduction

FPGA are pre-fabricated silicon devices that can be electrically programmed in the field to realize almost any kind of digital circuit or system [1]. A FPGA comprises of an array of programmable logic blocks that are connected to each other through programmable interconnect network. FPGA are often used to prototype Application Specific Integrated Circuits (ASIC) designs or to provide a hardware platform on which to verify the physical implementation of new algorithms [2]. One of the greatest factor that determines the performance of circuit in case of FPGA is placement. It is an important stage in the FPGA design flow, because it affects routability, performance, heat distribution and power consumption of a design [3]. So optimization of placement is a challenge in FPGA based design.

The FPGA placement optimization has been carried out by several methods like min-cut [4], SA, Genetic algorithm (GA) [5] and PSO [6] [7]. The min-cut technique employs the fundamental divide and conquers method. It uses recursive partitioning to divide a net-list of circuits into increasingly smaller sub-circuits. But it does not reach global solution. Also, the solution may vary on how the partition is performed. SA technique is motivated by the physical annealing process and use an analogous set of cooling operations for placement optimization problem [8]. It provides a global optimal solution to the problem but it is slow. GA is a search technique based on the based on Charles Darwin's theory of natural selection and natural genetics [5]. It uses genetic operators such as crossover and mutation for finding a good placement solution. But this optimization technique is complex. PSO is a biologically inspired computational search and optimization method that uses based the movement and intelligence of swarms for finding good solution. But PSO has a problem of premature convergence and it easily gets trapped into local minima [9].

In this work, a hybrid PSO-SA technique for FPGA placement is presented. The hybrid algorithm combines the high speed of PSO with the powerful ability to avoid being trapped in local minimum of SA for obtaining optimized placement. The hybrid algorithm involves an initial placement with PSO and refining of result with SA.

2. Placement Problem

FPGA placement usually begins with a net-list of logic blocks, which includes CLBs and I/O pads and their interconnections. The result of placement is the physical assignment of all blocks on the target FPGA, [10] that minimises one or more objective cost functions. Broadly there are three placement objectives i.e. wire length-driven placement, routability-driven placement and timing-driven placement. Wire length-driven placement tries to place logic blocks close together to minimize the required wiring [7]. The routing-driven placement balances the wiring density across the FPGA where as the timing-driven placement to minimise the length of a critical path to meet timing constraints.

This work considers the wire length driven placement objective. The total interconnection length of all the CLBs used in FPGA can be calculated as given in equation(1)[7]

$$wirelength\ cost\ function = \sum_{i,j}\left(\left|x_i - x_j\right| + \left|y_i - y_j\right|\right)$$

(1)

where $\left(x_i, y_i\right)$ and $\left(x_j, y_j\right)$ represents the position of of i_{th} and j_{th} CLB location

But placement is a computationally difficult and much more complicated problem [3]. The general placement problem is Non deterministic Polynomial time complete (NP-complete). NP-complete basically mean that there is no efficient way to compute an exact solution [8]. Only approximate solutions can be found using heuristic methods. In this work we have used three heuristic techniques for solving the placement problem; SA, PSO and proposed hybrid PSO-SA.

3. SA Based Placement

The technique was introduced in 1983 by Kirkpatrick, Gellet and Vecchi. It is motivated by the physical annealing process in which material is heated and slowly cooled into a uniform structure. Simulated annealing techniques use an analogous set of controlled cooling operations for nonphysical optimization problems, transforming a poor unordered solution into a highly optimized desirable solution [8]. Thus, simulated annealing offers an appealing physical analogy for the solution of optimization problems. The simulated annealing technique has been successfully used in many phases of VLSI physical design, e.g., circuit partitioning, floorplanning etc. In this work we have used simulated annealing for Placement problem. The algorithm starts with an initial placement which is created by randomly placing the logic blocks [1]. The cost function is used to evaluate the quality of placement of logic blocks. Then a new neighboring placement solution is created incrementally from the current solution. The change in cost function (ΔE) that results from moving the selected logic block to the proposed new location is computed. If the new cost, derived from a specific objective function, is reduced the new solution is accepted [12]. However, for an inferior cost, the new solution may still be accepted with a probability of exp (- ΔE/T), where ΔE is the change in cost and T is analogous to temperature in the metal crystallization process. Parameter T is used to control the acceptance probability of cost-increasing bad solutions. The rate of change of T is referred to as annealing schedule which has a great influence on the quality of the final solution. Initially, [11] T is set to a high value such that most inferior solutions can be accepted. As the process goes on, T is gradually decreased (simulating cooling), reducing the probability of accepting poor solutions. In the final stages, T is only a small fraction of its original value and only improving solutions are accepted most of the time. The cooling schedule for T is given by equation (2) [9]

$$T_{new} = \alpha T_{old} \qquad (2)$$

where α is the cooling rate and its value lies between 0< α <1

4. PSO Based Placement

Particle swarm optimization is a heuristic global optimization method put forward originally by Kennedy and Eberhart in 1995 [13]. It is developed from swarm intelligence and is based on the research of bird and fish flock movement behaviour [14]. In PSO, a swarm of particles moves through D dimensional search space. The particles in the search process are the potential solutions, which move around the defined search space with some velocity until the error is minimized or the solution is reached, as decided by the fitness function [15]. Fitness function is the measure of particles fitness which is the deviation of the particle from the required solution. The particles reach to the desired solution by updating their position and velocity according to the PSO equations. In PSO model, each individual is treated as a volume-less particle in the D-dimensional search space with initial random velocity. Each particle has memory which keeps track of following information [6]:

(i) Its current position

(ii) Its current velocity

(iii) The best position (*pbest*), the one associated with the best fitness value the particle has achieved so far.

(iv) The global best position (*gbest.*), the one associated with the best fitness value found among all of the particles In every iteration, each particle adjusts its own trajectory in the space in order to move towards both its best position and the global best according to the following mathematical equations (3) and (4) [7]

$$v_i(t+1) = wv_i(t) + c_1 rand_1()(pbest_i - x_i) + c_2 rand_2()(gbest_i - x_i)$$

(3)

$$x_i(t+1) = x_i(t) + v_i(t+1) \qquad (4)$$

where x_i and v_i are current position and velocity of particle i at any time step t. $c_1$and c_2 are two acceleration coefficients.$rand_1()$ and $rand_2()$ are random functions uniformly distributed in the range of [0,1]. $pbest_i$ is the personal best that stores the best position of the particle has ever visited and $gbest_i$ is the global best that stores the best position value among the whole particles. w is the inertia weight introduced to accelerate the convergence speed of PSO.

Each PSO particle represents the position vector of CLBs used in the design. The fitness or cost function *of* the particles is the total interconnection length of all the CLBs [16] used in FPGA which is given by the equation (1). The *pbest* of each particle stores the position vector of all the CLBs where the fitness function is the lowest. The *gbest* stores the position vector all the CLBs with the lowest fitness function of the particle in the whole swarm. The *pbest* and the *gbest* are continuously updated whenever a position vector with a lower fitness is found for each particle and the swarm respectively.

5. Proposed Hybrid PSO-SA Algorithm

The basic PSO algorithm gets easily trapped into local minimum and may lead to the premature convergence. In the worst case, when the best solution found by the group and the particles are all located at the same local minimum, it is almost impossible, due to the velocity update equation, for particles to fly out and do farther searching. The premature convergence of PSO could be effectively avoided if the mechanism of SA is incorporated into PSO. SA makes the search jump out of local optima due to its strong local-search ability. This hybrid approach makes full use of the exploration capability of both PSO and SA and offsets the weaknesses of each [17].

The basic idea of the hybrid algorithm presented here have two major operations first running PSO algorithm and obtaining a global best solution and then improving the result with SA to get the global optimal solution [18]. The Figure 1 explains the operation of the algorithm

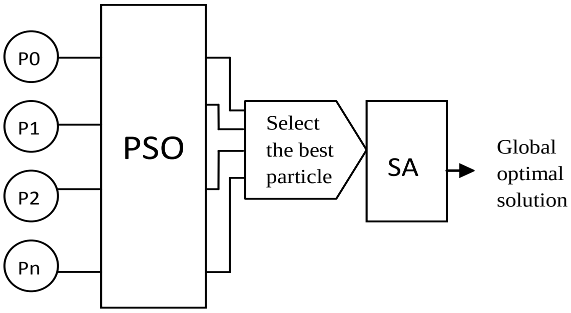

Figure 1: The operation schema of Hybrid PSO-SA [18] algorithm

Steps in the algorithm:

```
Begin
STEP 1: Initialization

Initialize swarm population, each particle's position and velocity;
Evaluate each particle's fitness;
Initialize gbest position with the lowest fitness particle in swarm;
Initialize pbest position with a copy of particle itself;    Initialize w,c1, c2, maximum iterations and iteration=0;

STEP 2: Operation

For PSO
Do {
While (iteration<maximum iteration)
Generate new position and velocity of swarm as per PSO mathematical equations;
Find new gbest and pbest;
Update gbest of swarm and pbest of the particle;
Iteration++;
}

For SA

Do {
$T_i$= set initial temperature;
 $T_f$= set final temperature;
 Initial solution= gbest;
While (current temperature T >$T_f$)
Generate a neighbor solution $S_0$ from S;
Calculate fitness of $S_0$;
Evaluate ΔE= f ($S_0$) –f(S);
If ΔE<0 then
Accept$S_0$;
S=$S_0$;
Else if exp(-ΔE/T)>random(0,1)
Accept $S_0$ ;
S=$S_0$;
 Update current temperature $T_i = \alpha T_i$ ;
    }

STEP 3: Output optimized results.
End
```

6. Results

In this work three heuristic techniques are used for solving the placement problem i.e. SA, PSO and proposed hybrid PSO-SA technique .We have applied these placement algorithms on *Synchronous FIFO* circuit. The design entries of these are done in Verilog HDL using Xilinx ISE Simulator and the netlist is generated by targeting a Xilinx FPGA device. The device used is XC3S500E in which the CLBs are arranged in a two dimensional array with 46 rows and 34 columns The design counts 10 CLBs in the device. The information obtained from the netlist is given to Placement algorithms for obtaining optimized placement. All the three algorithms i.e. SA, PSO and hybrid PSO-SA are implemented in Matlab

It is observed that with SA based placement we have attain a value of 321 for wirelength cost function and with PSO, a minimum value of 192 is obtained. It has been found that with the proposed hybrid PSO-SA, a minimum value of 168 is obtained which is better than both SA and PSO. The Figures 2, 4, 6 shows the convergence of cost function using SA, PSO, Hybrid PSO-SA respectively and the Figures 3, 5, 7 shows the CLBs locations with SA , PSO and Hybrid PSO-SA respectively. The table 1 shows the comparison of results.

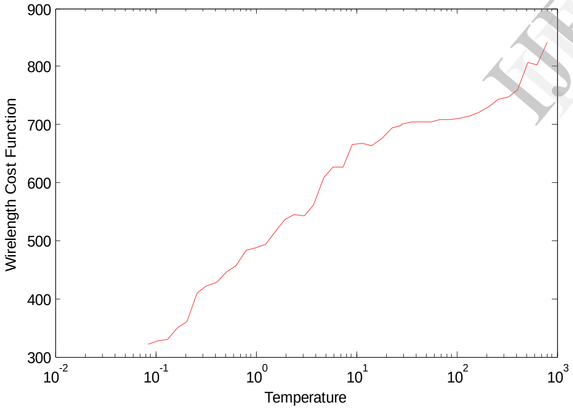

Figure 2: Convergence of wirelength cost function for to a minimum value by SA Algorithm

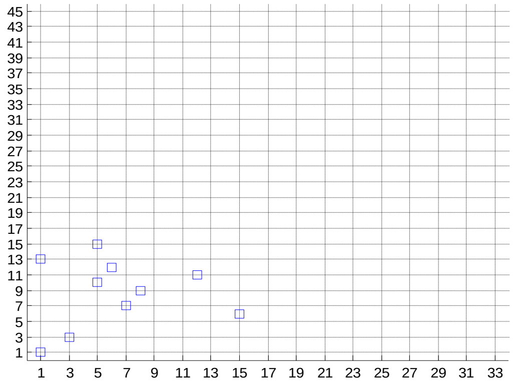

Figure 3: CLB locations of SA based placement

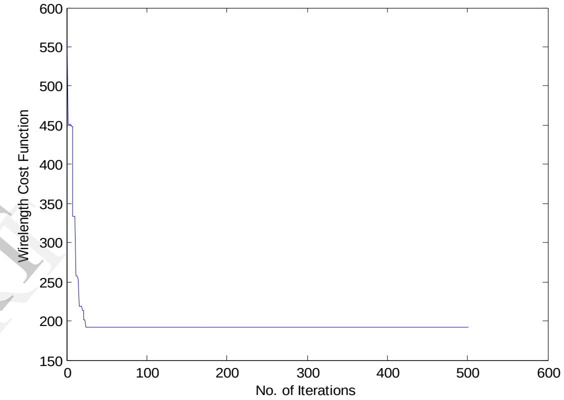

Figure 4: Convergence of wirelength cost function to a minimum value by PSO

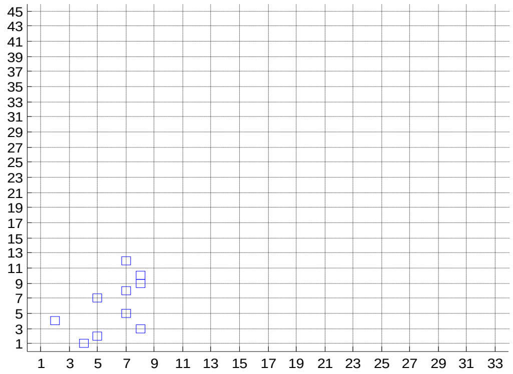

Figure 5 : CLB locations of PSO based placement

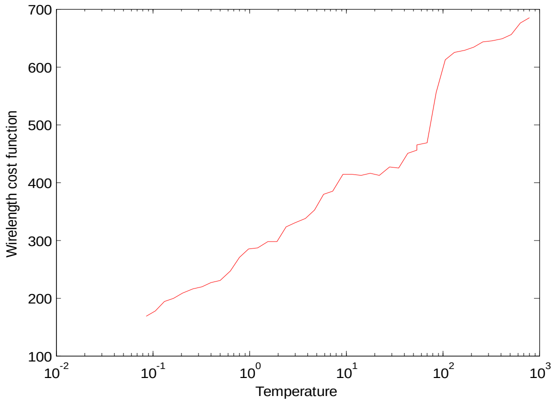

Figure 6: Convergence of wirelength cost function for to a minimum value by hybrid PSO-SA Algorithm

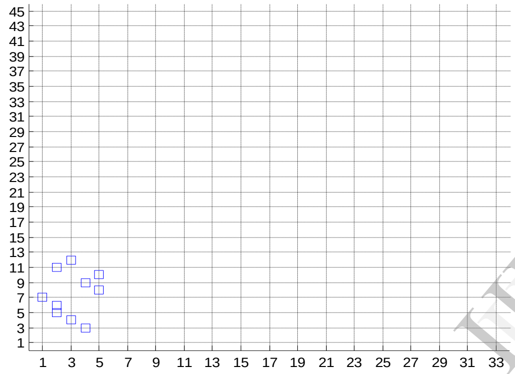

Figure 7: CLB locations of hybrid PSO-SA basedplacement

Table 1 : Comparison of different placement techniques

Placement Algorithm	Value of the cost function
SA	321
PSO	192
Hybrid PSO-SA	168

7. Conclusion

In this paper a new hybrid PSO-SA algorithm for solving the placement problem in FPGA is proposed. This work considers the wirelength driven placement objective. The result is implemented on Synchronous FIFO. The hybrid algorithm combines the high speed of PSO with the powerful ability to avoid being trapped in local minimum of SA. The hybrid algorithm involves an initial placement with PSO and refining of result with simulated annealing. The results obtained from the hybrid PSO-SA are compared with the results obtained from SA and PSO applied individually. It is found that by integrating the SA and PSO algorithm we have obtained better results than applying them individually.

8. References

[1] Farooq, Umer, Zied Marrakchi, and Habib Mehrez. "FPGA Architectures: An Overview." *Tree-based Heterogeneous FPGA Architectures*. Springer New York, pp 7-48 2012..

[2] Maxfield, Clive. *FPGAs: Instant Access: Instant Access*. Newnes, 2011.

[3] Chu, Chris. "Placement." *International Symposium on Physical Design* 2007.

[4] P. Maidee, C. Ababei, and K. Bazargan, “Time-driven partitioning based placement for island style FPGA, “ *IEEE Transaction on Computer Aided Design of Integrated circuits and Systems*, Volume 24, Issue , pp 395-406, March 2005

[5] Yang, M., A. E. A. Almaini, and L. Wang. "FPGA placement by using genetic algorithm." *EngineerIT, June* 2007.

[6] El-Abd, Mohammed, et al. "Discrete cooperative particle swarm optimization for FPGA placement." *Applied Soft Computing* pp284-295 2010.

[7] Rout, P. K., D. P. Acharya, and G. Panda. "Digital circuit placement in FPGA based on efficient particle swarm optimization techniques." IEEE, *International Conference on. Industrial and Information Systems (ICIIS), 2010*.

[8] Rutenbar, Rob A. "Simulated annealing algorithms: An overview." *Circuits and Devices Magazine, IEEE* 5.1 pp 19-26, 1989.

[9] Ercan, M. Fikret, and Xiang Li. "Particle Swarm Optimization and Its Hybrids 2013"

[10] Areibi, S., et al. "A Comparison of Heuristics for FPGA Placement." *ACTA International Journal of Computers and Applications* 16.1 pp13-33, 2006 .

[11] University of Guelph. Dept. of Computing and Information Science, and Peng Du. *Fast heuristic techniques for FPGA placement based on multilevel clustering*. University of Guelph, 2003

[12] Sherwani, Naveed A. *Algorithms for VLSI physical design automation*. Kluwer academic publishers, 1995.

[13] Kennedy, James, and Russell Eberhart. "Particle swarm optimization." *IEEE International Conference on Neural Networks*, Vol. 4 , *1995.*

[14] Bai, Qinghai. "Analysis of particle swarm optimization algorithm." *Computer and information science* 3.1 2010

[15] Luitel, Bipul. *Applications of Swarm, Evolutionary and Quantum Algorithms in System Identification and Digital Filter*

Design. Diss. Missouri University of Science and Technology, 2009.

[16] Gudise, Venu G., and Ganesh K. Venayagamoorthy. "FPGA placement and routing using particle swarm optimization." *IEEE Computer society Annual Symposium on VLSI, 2004. Proceedings.*

[17] Idoumghar, Lhassane, et al. "Hybrid PSO-SA type algorithms for multimodal function optimization and reducing energy consumption in embedded systems."*Applied Computational Intelligence and Soft Computing* 2011.

[18] Saffarzadeh, Vahid Mohammadi, Pourya Jafarzadeh, and Masoud Mazloom. "A Hybrid Approach Using Particle Swarm Optimization and Simulated Annealing For N-queen Problem." *Journal of World Academy of Science, Engineering and Technology* pp 974-978 , 2010 .