Multiobjective VLSI Cell Placement

Using Distributed Simulated Evolution Algorithm

Sadiq M. Sait, Mustafa I. Ali, Ali Zaidi

What this is Paper is About

- Parallelization of an Evolutionary Heuristic for wire length, power and delay optimized VLSI cell placement is presented
- An improved Parallel SimE Algorithm for Cell Placement is proposed and results are compared with a previous approach

Need for Parallelism

Å For large test cases and multiobjective optimization, SimE has large runtime requirements

Å SimE, like other stochastic heuristics, is blind and has to be told when to stop

Å Can consume hours of CPU time depending upon problem size, complexity and stopping criteria

Cost Functions

Å Objectives

- ï Reducing overall wire length
- ï Optimizing power consumption
- ï Improving timing performance (delay)

Å Contraint

- ï Layout width should be within set limit

Wire Length Estimation

- Wire length for each net is estimated using an approximate Steiner Tree Algorithm
- Total wire length of whole placement is computed by adding individual wire length estimates of each net

$$Cost_{wire} = \sum_{i \in M} l_i$$

  - where l_i is the wire length estimation for net and M denotes total number of nets in circuit.

Power Estimation

- Power consumption p_i of a net i in a circuit can be given as

$$p_i \simeq \frac{1}{2} \cdot C_i \cdot V_{DD}^2 \cdot f \cdot S_i \cdot \alpha$$

  - C_i where is total capacitance of net , V_{DD} is the supply voltage, f is the clock frequency,
  - S_i is the switching probability of net, and α is a technology dependent constant

Power Estimation (cont.)

Å Assuming a fix supply voltage and clock frequency, we have:

$$p_i \simeq C_i \cdot S_i$$

Å The capacitance C_i of cell i is given as:

$$C_i = C_i^r + \sum_{j \in M_i} C_j^g$$

Å Moreover, C_i^r depends on wire length l_i of net i, so above equation can be written as:

$$p_i \simeq l_i \cdot S_i$$

Power Estimation (cont.)

- The cost function for estimate of total power consumption in the circuit can be given as:

$$Cost_{power} = \sum_{i \in M} p_i = \sum_{i \in M} (l_i \cdot S_i)$$

Delay Estimation

Å Delay along the longest path in the circuit

Å Delay T_π of a path π consisting of nets $\{v_1, v_2,$é $,v_k\}$, is expressed as:

$$T_\pi = \sum_{i=1}^{k-1}(CD_i + ID_i)$$

ï Where CD_i is switching delay of cells driving net v_i, ID_i is interconnect delay of net v_i

Å Since CD_i is placement independent, delay cost is given by:

$$Cost_{delay} = max\{T_\pi\}$$

Width Cost

- Given by the maximum of all the row widths in the layout

- w_{avg} is minimum possible layout width
  - obtained by dividing the total width of all the cells in the layout by the number of rows in the layout

Width Constraint

- Layout width should not exceed a certain positive ratio α to the average row width w_{avg}

$$Width - w_{avg} \leq \alpha \times w_{avg}$$

  - where Width is the width cost computed

Fuzzy Multi-Objective Function

Å A cost function that represents the effect of all three objectives in form of a single quantity

Å Use of fuzzy logic to integrate multiple, possibly conflicting objectives into a scalar cost function

Fuzzy Logic Rule

Å Fuzzy logic allows us to describe the objectives in terms of linguistic variables

Å Fuzzy rules are used to find the overall cost of a placement solution

Å Following rule is used

“**IF** a solution has SMALL wirelength **AND** LOW power consumption **AND** SHORT delay **THEN** it is an GOOD solutionò

Fuzzy Membership Function

- Fuzzy rule is translated to and-like OWA fuzzy operator
- Membership $\mu(x)$ of a solution x in fuzzy set GOOD solution is given as:

$$\mu(x) = \begin{cases} \beta \cdot \min_{j=p,d,l} \{\mu_j(x)\} + (1-\beta) \cdot \frac{1}{3} \sum_{j=p,d,l} \mu_j(x); & \\ \quad \text{if } Width - w_{avg} \leq \alpha \cdot w_{avg}, & \\ 0; \quad \text{otherwise.} & \end{cases}$$

  - where $\mu_j(x)$ for j= p, d, l, width are membership values in fuzzy sets for power, delay and wire length,
  - β is a constant in the range [0,1]

Evolution (SimE) Algorithm

- A general search strategy
- Operates on a single solution termed as population
- Has a main loop consisting of 3 main steps
  - Evaluation
  - Selection
  - Allocation

Three Operators in SimE Algorithm

Å **Evaluation** ï calculation of goodness of each element of population

Å **Selection** ï process of selecting elements to be reassigned locations in the current solution

Å **Allocation** ï Mutate the population by altering locations of selected cells

SimE Algorithmic Description

Algorithm Slave_Process($CurS$, Φ_s)
Notation
(* B is the bias value. *)
(* $CurS$ is the current solution. *)
(* Φ_s are the rows assigned to slave s. *)
(* m_i is module i in Φ_s. *)
(* g_i is the goodness of m_i. *)
Begin
Receive Placement_ And_ Indices
Evaluation:
ForEach $m_i \in \Phi_s$ evaluate g_i;
Selection:
ForEach $m_i \in \Phi_s$ **DO**
Begin
If $Random > Min(g_i + B, 1)$
Then
Begin
$S = S \cup m_i$; Remove m_i from Φ_s
End
End
Sort the elements of S
Allocation:
ForEach $m_i \in S$ **Do**
Begin
Allocate(m_i, Φ_s)
(* Allocate m_i in local partial solution rows Φ_s. *)
End
Send_Partial_Placement_Rows
End. (*Slave_Process*)

Distributed SimE Algorithm

Å Workload partitioning by dividing rows in a placement (population)

Å Each PE computes the 3 SimE operators on assigned rows (a sub-population)

Å Individual Sub-populations are merged after each iteration and new sub-populations created and distributed among PEs

Distributed SimE Algorithm

Master PE does:

Å Receive partial placement from all PEs, combine them and evaluate fitness,

Å Re-partition to obtain new allocations,

Å Distribute new partial placements among PEs

```
 Algorithm Parallel_Simulated_Evolution
   Read_User_Input_Parameters
   Read_Input_Files
Begin
   Construct_ Initial_ Placement
   Repeat
   Generate Random_ Row-Indices
       ParFor
         Slave_ Process(CurS, Φs)
   (* Broadcast Cur Placement And Row-Indices. *)
       EndParFor
       ParFor
         Receive_Partial_Placement_Rows
       EndParFor
     Construct_Complete_Solution
     Calculate_Cost
   Until (Stopping Criteria is Satisfied)
 Return Best_Solution.
End. (*Parallel_Simulated_Evolution*)
```

Proposed Improvement

Å Originally proposed row distribution comprises alternating block and row assignments

Å Solution qualities inferior due to

1. Lack of a global placement view to all PEs
2. Restrictive cell movement due to a fixed allocation pattern

Å Our solution addresses the second problem

Randomized Rows Assignment

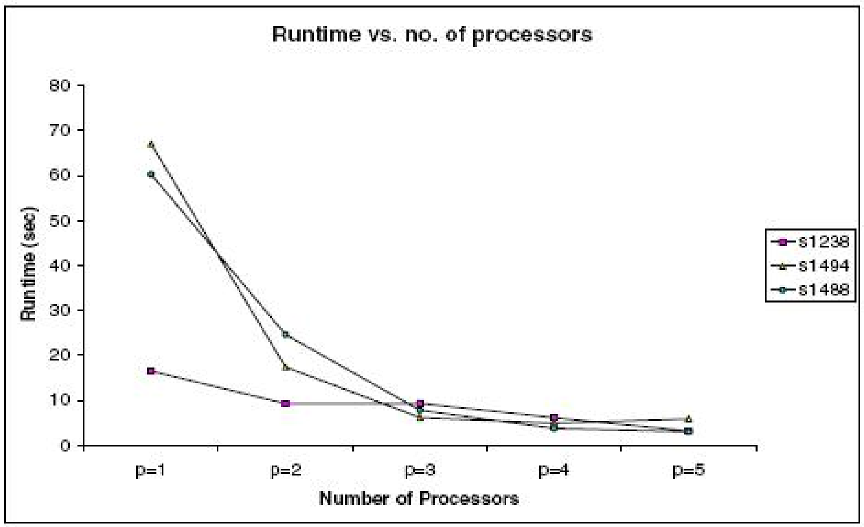
Å Restrictive cell movement can be alleviated using better row assignment

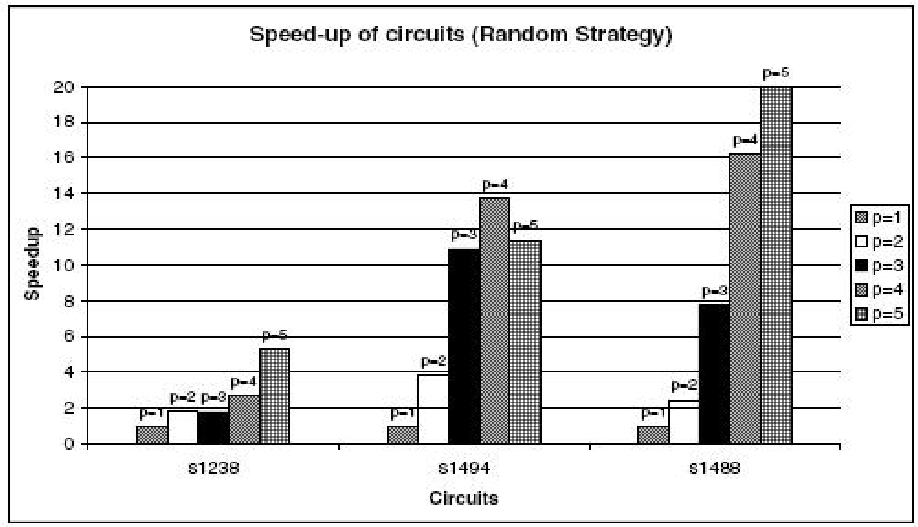
Å An assignment that facilitates better inter-mixing among partitions would be intuitively better

Å Our experimentation with a randomized row assignment gave better results

Experimental Results

Circuit Name	# of Cells	Random Row Distribution					Qual Fixed	Fixed Row Distribution			
		N_p=1	N_p=2	N_p=3	N_p=4	N_p=5		N_p=2	N_p=3	N_p=4	N_p=5
s641	433	UH	4.99	4.97	3.99	3.87	79.7%	9.14	1.08	0.76	0.55
s1238	540	16.5	9.24	9.29	6.12	3.14	95.8%	17.83	8.47	11.30	5.71
s1494	661	67	17.4	6.15	4.88	5.89	82.3%	2.77	1.85	1.76	4.34
s1488	667	60.23	24.6	7.78	3.72	3.02	96.6%	22.0	4.89	5.1	16
s3330	1961	UH	678.02	115	108.5	49.14	33.8%	316	215	4.6	3.4
s5378	2993	UH	1620	338.2	286.6	178.6	46.8%	UH	UH	124.3	95.0





Future Work

Å Evaluation of SimE algorithm parameters for further improvement in parallel version

Å Use of processor relieve strategy as quality stagnates to enable final solution qualities equivalent to serial version but with improved runtimes

References

[1] Sadiq M. Sait and Habib Youssef. *Iterative Computer Algorithms with Applications in Engineering: Solving Combinatorial Optimization Problems*. IEEE Computer Society Press, California, December 1999.

[2] Ralph M. Kling and Prithviraj Banerjee. ESP: A new standard cell placement package using simulated evolution. *Proceedings of 24th Design Automation Conference*, pages 60–66, 1987.

[3] Sadiq M. Sait and Habib Youssef. VLSI Physical Design Automation: Theory and Practice. *World Scientific Pubishers*, 2001.

[4] John A. Chandy, Sungho Kim, Balkrishna Ramkumar, Steven Parkes, and Prithviraj Banerjee. An evaluation of parallel simulated annealing strategies with application to standard cell placement. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 16:398–410, April 1997.

[5] Robert Azencott, editor. *Simulated Annealing Parallelization Techniques*. John Wiley & Sons, 1992.

[6] Erick Cant-Paz. Designing efficient master-slave parallel genetic algorithms. *Genetic Programming*, 1998.

[7] E. Taillard. Robust tabu search for the quadratic assignment problem. *Parallel Computing*, 17:443–455, 1991.

[8] Prithviraj Banerjee. *Parallel Algorithms for VLSI Computer-Aided Design*. Prentice Hall International, 1994.

[9] Sadiq M. Sait, Mahmood R. Minhas, and Junaid Asim Khan. Performance and low-power driven VLSI standard cell placement using tabu search. *Proceedings of the 2002 Congress on Evolutionary Computation*, 1:372–377, May 2002.

[10] Ronald R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Transaction on Systems, MAN, and Cybernetics*, 18(1), January 1988.