FUZZIFIED SIMULATED EVOLUTION ALGORITHM FOR MULTI-OBJECTIVE OPTIMIZATION OF COMBINATIONAL LOGIC CIRCUITS

Sadiq M. Sait, Mostafa Abd-El-Barr, Uthman Al-Saiari and Bambang A. B. Sarif

Computer Engineering Department
KFUPM Box 673, Dhahran-31261, Saudi Arabia
{sadiq, mostafa, saiarios, sarif}@ccse.kfupm.edu.sa

ABSTRACT

In this paper, we employ fuzzified Simulated Evolution (SimE) Algorithm for combinational logic design. SimE algorithm consists of three steps: evaluation, selection and allocation. *Multilevel Logic Based Goodness* measure is designed to guide the selection and allocation operations of SimE. Area, power and delay are considered in the optimization of circuits. The performance of the proposed algorithm is evaluated using selected ISCAS'85 benchmark circuits. The results obtained are compared to those obtained using SIS.

1. INTRODUCTION

Design of logic circuits requires knowledge of a large collection of domain-specific rules. The process of implementing a logic circuit in hardware involves transforming the original specification into a form suitable for the target technology, optimizing the representation with respect to a number of user defined objectives and constraints (i.e., timing, fan-in/out, power, etc.), and finally carrying out technology mapping onto the target technology [1].

In designing a complex system, circuit designers usually have to tradeoff one design objective for another. For example, often a designer tries to find a possibly faster circuit compared to a previously designed one. However, the number of gates used and power dissipation are strongly related to delay. Thus, in seeking a faster circuit, the designer may end up having a complex system or a system that has higher power dissipation. Logic synthesis attempts to provide an answer to this problem. The purpose of a logic synthesis tool is to aid circuit designers reaching an optimal tradeoff. Several logic synthesis algorithms are found in the literature [2].

Evolutionary algorithms allow designers to define the search space of circuit design in a way that is natural to both the problem and the implementation. These algorithms have the tendency to search for a solution to the circuit design problem in a much larger, and often richer, design space beyond the realms of the conventional design tools. It is therefore possible to use evolutionary algorithms to obtain novel designs that are difficult to discover by conventional heuristics [3].

In this paper, fuzzified Simulated Evolution (SimE) algorithm for combinational logic design targeting area, power and delay optimization is proposed. The results from the fuzzified Simulated Evolution (SimE) algorithm using selected ISCAS'85 benchmark circuits are compared to the results obtained by a conventional logic design technique using SIS synthesis tool [4].

2. PROPOSED SIMULATED EVOLUTION

The SimE algorithm is a general search strategy for solving a variety of combinatorial optimization problems. It starts from an initial assignment, and then, following an evolution-based approach, it seeks to reach better assignments from one generation to the next. A ***goodness measure*** is used to guide the algorithm in its search [5].

Assume there are a set L of $|L|$ distinct locations and a set M of n elements of two inputs logic gates. SimE algorithm proceeds as follows. Initially, a population[1] is created at random. Therefore, $|L|$ locations (cells) will be filled randomly by different logic gates from M. The algorithm has one main loop consisting of three basic steps, *Evaluation*, *Selection*, and *Allocation*. The algorithm evaluates every location (cell) of the set L using the new ***Multilevel Logic Based Goodness Measure*** designed. Some cells will be selected for reallocation (mutation) by the *Selection* step according to their goodness values. Next, in *Allocation* step, selected cells will be assigned different gate types in order to improve their goodness. The three steps are executed in sequence until the population average *goodness* reaches a maximum value, or no noticeable improvement to the population *goodness* is observed after a number of iterations. Another possible stopping criterion could be to run the algorithm for a prefixed number of iterations. Figure 1 is an

[1] In SimE terminology, a population refers to a single solution. Individuals of the population are components of the solution; they are the movable elements.

ALGORITHM *Simulated_Evolution(E, ,L, Stopping-Criteria)*;
$INITIALIZATION$;
Repeat
    $EVALUATION$:
        ForEach $m \in M$ **Do** $g_m = \frac{O_m}{C_m}$ **EndForEach**;

    $SELECTION$:
        ForEach $m \in M$ **Do**
            If $Random \leq Min(1 - g_m + B; 1)$
                Then $P_s = P_s \cup \{m\}$;
            Else $P_r = P_r \cup \{m\}$;
            EndIf;
        EndForEach;
    Sort the elements of P_s;
    $ALLOCATION$:
        ForEach $m \in P_s$ **Do** $F_a(m)$ **EndForEach**;
Until *Stopping-criteria are met*;
Return ($BestSolution$);
End *Simulated_Evolution*.

Fig. 1. Simulated Evolution algorithm [5].

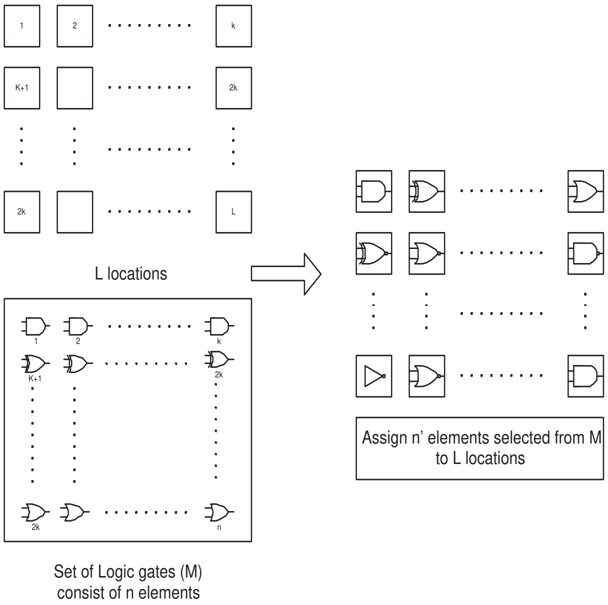

Fig. 2. Representation of logic design problem.

outline of the SimE algorithm and Figure 2 is a representation of the problem considered.

3. COST FUNCTION FORMULATION

The cost function or the fitness of a solution consists of two parts: functional fitness and objective fitness.

3.1. Functional Fitness

The functional cost measure is the correctness of the obtained logic circuit in matching the truth table of the required function. In this work, we propose functional cost measure called ***Multilevel Logic Based Goodness Measure*** and it is based on the multilevel logic synthesis. As an example, if we have a 4 inputs circuits, it is represented by a 4 $\times$ 4 matrix. Also, there are 16 outputs that should be generated correctly to get a functionally correct circuit. Initially, these patterns are distributed among the levels of the circuit evenly and progressively. Each column in the 4 $\times$ 4 matrix should cover 4 outputs more than it's previous column. If a logic gate located at the second level (second column of the matrix), it should cover 8 outputs. If the same logic gate is located at the second level and it covers less than 8 outputs, then it will be considered to have less goodness. Therefore, the ***goodness*** of a gate is affected by the number of outputs covered and the level where the gate is located. Figure 3 illustrates this goodness measure.

In general, for an n input circuit (2^n outputs), to have a goodness of 1 at a cell in level i, there should be $\lceil (2^n/n)i \rceil$ correct patterns produced at this cell. Thus, the *multilevel logic goodness measure* is formulated as follows:

$$g_i = \frac{\rho}{\lceil 2^n/n \rceil j}$$

where g_i is the *goodness* of cell i, j is the level number or column number, n is the number of inputs of the required circuit and ρ is the number of matching patterns at the output of cell i compared to the intended truth table.

3.2. Objective Fitness

The objective fitness (F_o) is a measure of the quality of solution in terms of optimization of area, delay and power consumption. It contains two aspects: constraints satisfaction and multi-objective optimization.

The cost for area of VLSI circuits is stated as follows.

$$Cost_{area} = \sum_{i \in G, i \neq WIRES} A(g_i) \qquad (1)$$

Where $A(g_i)$ is the area of a gate g_i.

The propagation delay of signals in a VLSI circuit consists of two elements, switching delay of gates and interconnect delay. If a path π consists of n gates $\{v_1, v_2, ..., v_n\}$, then, the delay T_π along π is expressed as

$$T_\pi = \sum_{i=1}^{n-1} (CD_i + ((LF_i + R_i) \times C_i)) \qquad (2)$$

Where CD_i is the switching delay of the cell driving gate v_i. LF_i is the load factor of the driving block, R_i is the

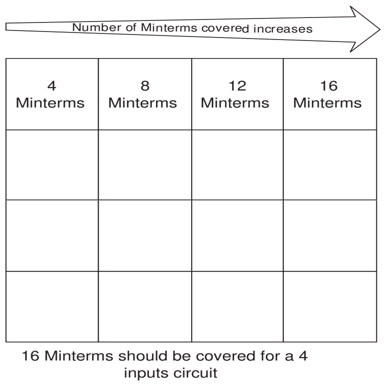

Fig. 3. Multilevel logic goodness assumption.

interconnect resistance of net v_i, and C_i is the load capacitance of cell i given by Equation 4. Since the value of R_i at logic synthesis level is not known, it can be neglected or estimated. The overall circuit delay is determined by the delay along the longest path (the most critical path)as given by Equation 3.

$$Cost_{delay} = Max(T_{\pi i}) \tag{3}$$

The total capacitance C_i of gate i consists of the interconnect capacitance at the output node of gate i and the sum of the capacitances of the input nodes of the gates driven by gate i.

$$C_i = C_i^r + \sum_{j \in M_i} C_j^g \tag{4}$$

Where C_j^g is the capacitance of the input node of a gate j driven by gate i and C_i^r represents the interconnect capacitance at the output node of cell i.

The total power consumption can be approximated by the following equation [6].

$$P_t \simeq \sum_{i \in M} \frac{1}{2} \cdot C_i \cdot V_{DD}^2 \cdot f \cdot S_i \cdot \beta \tag{5}$$

Where P_t is the total power consumption, V_{DD} is the supply voltage, S_i is the switching probability at the output node of cell i, i.e., the average number of transitions per clock cycle at the output of gate i, f is the clock frequency and β is a technology dependent constant.

The cost of the overall power consumption in VLSI circuits can then be estimated as follows.

$$Cost_{power} = \sum_{i \in M} S_i \cdot C_i \tag{6}$$

In this paper, fuzzy logic is used to represent the cost function for area, delay and power. In order to build the membership function, the lower bound and upper bound of the cost function must be determined. In order to guide the search intelligently, the maximum value must be carefully estimated. For this purpose, SIS tool [4] is used to estimate the minimum area and minimum delay of the target circuits.

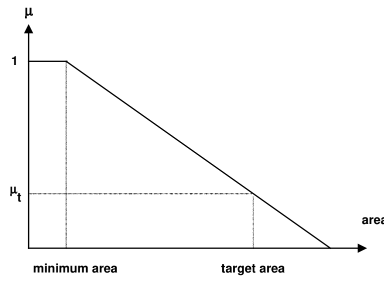

Fig. 4. Membership function for area as optimization objective

The estimated lower bound of maximum area (called $target_{area}$) is associated with a specific degree of membership called target membership (μ_{target}). The shape of the membership function is depicted in Figure 4.

The membership function for delay and power are built using similar rules. These three membership functions will be aggregated into one unit (the objective fitness) using ordered weighted average (OWA) operator [7].

4. EXPERIMENTS AND RESULTS

In this section, comparison of the proposed algorithm with an existing conventional technique is given. For this purpose, SIS is used. However, SIS does not consider capacitance load in its delay calculation and does not have power optimization. Therefore, the results obtained from SIS are in the form of *netlist* file. These netlist files will be used as input to the cost function calculation procedures of the proposed algorithm to determine the area, delay and power of the circuits. Both SIS and the proposed algorithm use the same gate library.

Area Optimization: The results from SIS are the area optimized circuits obtained by executing *rugged.script* mapped for area minimization.

Table 1 shows the results for delay optimization for both techniques. The table shows that the highest improvements are obtained in the cost of cm82a and mul3 circuits by 58.3% and 51.63 % respectively.

Delay Optimization: For delay optimization, the results from SIS are obtained by executing *delay.script* mapped for delay minimization. The test cases used are the same circuits used for area optimization.

Table 2 shows the results for delay optimization for both techniques. It can be seen that the results obtained from

Circuit	SIS			Proposed Algorithm			% Improvement		
	Area	Delay	Power	Area	Delay	Power	Area	Delay	Power
mul2	18225	6.587026	5.561531	12636	3.56	4.66	44.23	84.98	19.35
mul3	112752	43.385843	37.745321	74358	13.14	21.65	51.63	230.23	74.38
cm42a	40824	8.864164	13.648234	40824	8.86	13.64	0.00	0.00	0.00
cm82a	39609	19.539984	14.879328	25029	11.84	9.24	58.3	64.98	61.12
b1	10206	3.225844	3.994219	11215	2.91	2.78	-8.93	10.85	43.68
c17	9963	3.559452	3.64207	9963	3.55	3.64	0.00	0.00	0.00
con1	31590	8.637996	11.21212	30233	6.90	14.23	4.49	25.19	-21.21
majority	14823	6.276723	5.405396	13851	4.57	5.06	7.02	37.32	6.93

Table 1. Comparison with SIS in area optimization

Circuit	SIS			Proposed Algorithm			% Improvement		
	Area	Delay	Power	Area	Delay	Power	Area	Delay	Power
mul2	25272	4.331142	7.157387	12636	3.56	4.66	100	21.66	53.59
mul3	174231	31.663494	47.161334	74358	13.14	21.65	51.63	230.23	74.38
cm42a	43740	8.46137	12.233066	40824	8.86	13.64	0	0	0
cm82a	64638	19.011371	18.936375	28552	9.34	9.1	38.73	109.21	63.51
b1	10206	3.225844	3.994219	11215	2.91	2.78	-8.93	10.85	43.68
c17	9963	3.559452	3.64207	9963	3.55	3.64	0	0	0
con1	31590	8.637996	11.21212	30233	6.9	14.23	4.49	25.19	-21.21
majority	14823	6.276723	5.405396	13851	4.57	5.06	7.02	37.32	6.93

Table 2. Comparison with SIS in delay optimization

SimE algorithm for area and delay optimization are the same except cm82a improved in the delay with larger area. The table shows that the highest improvements are obtained at cm82a and mul3 circuits.

5. CONCLUSION

In this paper, the use of Simulated Evolution (SimE) algorithm in logic design is being proposed. A goodness measure to guide SimE algorithm through the search space of digital logic design is suggested. Comparison of the proposed approach with SIS is shown. The proposed approach has shown better results in most cases compared to SIS considering area optimization and delay optimization.

Acknowledgment: The authors would like to acknowledge the support received from King Fahd University of Petroleum & Minerals under project entitled "Iterative Heuristics for the Design of Combinational Logic Circuits".

6. REFERENCES

[1] J. F. Miller, D. Job, and Vassilev V. K., "Principles in the Evolutionary Design of Digital Circuits - Part I," *Journal of Genetic Programming and Evolvable Machines*, vol. 1, no. 1, pp. 8–35, 2000.

[2] R. Brayton, G. D. Hachtel, C. T. McMullen, and A.L. Sangiovanni-Vincentelli, *Logic Minimisation Algorithms for VLSI Synthesis*, Kluwer Academic Publisher, 1984.

[3] J. F. Miller and P. Thomson, "A Developmental Method for Growing Graphs and Circuits," *Fifth International Conference on Evolvable Systems: From Biology to Hardware*, vol. 2606, pp. 93–104, Mar 2003.

[4] E. M. Sentovic, K. J. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P. R. Stephan, R. K. Brayton, and A. L. Sangiovanni-Vincentelli, "SIS: A System for Sequential Circuit Synthesis," Technical Report UCB/ERL M92/41, University of California, Berkeley, May 1992.

[5] S. Sait and H. Youssef, *Iterative Computer Algorithms with Applications in Engineering: Solving Combinatorial Optimization Problems*, IEEE, 1999.

[6] Srinivas Devadas and Sharad Malik, "A Survey of Optimization Techniques Targeting Low Power VLSI Circuits," *32nd ACM/IEEE Design Automation Conference*, 1995.

[7] Ronald R. Yager, "On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decision Making," *IEEE Transaction on Systems, MAN, and Cybernetics*, vol. 18, no. 1, January 1988.