Optimization of Cell-based VLSI Circuit Design using a Genetic Algorithm: Design Approach

Manisha V. Wankhede, Amol Y. Deshmukh

Abstract---**A technique for structural system of VLSI circuit is proposed here. The technique uses a GA, which utilizes a library of devices for synthesis procedure. This will be proved successful in searching a highly complex space and structures by circuit criteria. In addition to this, it will satisfy the functional constraint for each of the output and also satisfy hardware specific criteria such as area and delay**.

Index Words- **Cell based VLSI circuit, Multi objective Genetic Algorithm.**

I. INTRODUCTION

Genetic Algorithms (GAs) have been applied to various aspects of the digital VLSI design area. Examples include cell placement, channel routing, test pattern generation, design for test and VLSI-based signal processing. However, the use of genetic algorithms for higher level *structural* VLSI design synthesis has been for severely restricted forms of structure. The need for this restriction arises mainly from the complexity of the solution space which grows exponentially with the number of parameters to be optimized. Such solution space will exemplify in the case of a GA which manipulates typical VLSI cell libraries which incorporate hundreds of cells with different attributes. The increase in the complexity of of structural design synthesis that carried out at a much higher level. The developed GA can synthesize an unrestricted range of circuits using a multiple-constrained fitness function which will consider aspects at different levels over the design hierarchy and can deal with multi-output circuits. current VLSI circuits has led to a subsequent increase in the design effort required by the VLSI system designer who has to consider aspects at various levels throughout the design hierarchy. For this reason, it is particularly desirable if attributes at different levels in the design hierarchy will be considered by the GA. Examples will be the considerations of attributes such as individual gate delays, their constituent transistor parameters, the length of the interconnections in the circuit and their fan-in/fan-out. This paper reports on a GA which overcomes the restrictions of the work carried out to date in the field

Manisha V. Wankhede is a Research scholar, GHRCE, Nagpur, India. email: manishakhorgade@hotmail.com

Amol Y. Deshmukh is an Assist. Prof. ECE Department, GHRCE, Nagpur. He is also IEEE member.
email: amolydeshmukh@yahoo.com

II. CELL LIBRARIES

The design libraries used include primitive cells such as AND, OR, NAND, INV, XOR, XNOR. A specific library may include some of these primitives and other more advanced ones. A library, which could be viewed as lookup-table includes information such as the following, about each cell.

A. *Code*: A number used in the cell-type parameter of a gene (see section III).
B. *Function*: Logical function of the cell.
C. *Delay*: This includes delay parameters such as typical and maximum input-output delay in nanoseconds.
D. *Area*: The physical area of the cell.

For the purpose of illustrating our technique, we will use cell libraries for this work containing two-input/one output primitives. However, the technique can easily be adapted to deal with cells with more i/o.

III. THE GENETIC ALGORITHM

Genetic algorithms (Holland, 1992) are computer based optimization methods that uses the Darwinian evolution of nature as a model. The solution base of the problem is encoded as individuals. Those are chromosomes consisting of several genes. In GAs the virtual individuals are tested against the problem represented as a fitness function. The better the fitness value, individual gets the better chance. It has to select to be a parent for new individuals. The worst individuals are removed from the population in order to make room for the new generation. Using crossover and mutation operations, GA creates new individuals.

A. *Biological Terminology*

1) *Gene*: Single Encoding part of the solution space i.e. either single bits or short blocks of adjacent bits that encode an element of the candidate solution.
2) *Chromosome:* A string of genes that represents the solution.
3) *Population:* Number of chromosomes that available to best.

B. *Features of Genetic Algorithm*

1) Not too fast but cover large search space, Capable of quickly finding promising regions of the search space but may take relative long time to reach the optimal solution. Solutions: Hybrid algorithms.
2) Good heuristics for combinatorial problems.
3) Usually emphasize combining information from good parents (Crossover).
4) Different GAs use different
   - Representation
   - Mutations
   - Crossovers
   - Selection Mechanism

C. *Benefits of Genetic Algorithm*

1) Concept is easy to understand.
2) *Modular*: Separate from application (representation); building block can be used in hybrid application.
3) Supports in "multi-objective optimization".
4) Good for 'Noisy' environment.
5) Can easily run in 'Parallel'.
6) The fitness function can be changed from iteration to iteration, which allows incorporating new data in the model if it becomes available.

The flow of GA will be as shown in Fig. 1.

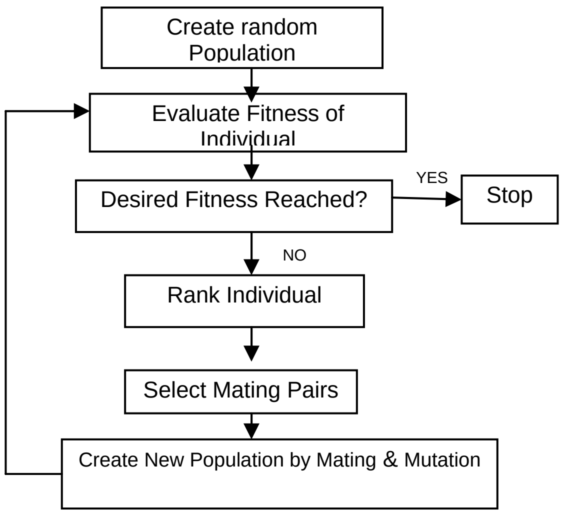

Figure 1: GA flow

The GA which we will be implemented is to suit the structural design synthesis problem. For example, the chromosome representation in Fig. 2 is proposed to design to incorporate individual library cells and their attributes (e.g., inputs and outputs) in a manner to incorporate circuits of any size. The representation using is compatible with the *extracted* circuit representation used by the majority of CAD tools (such as Cadence®, Mentor®, and ES2®) which provides the added flexibility of integration with such tools and hence being a valuable tool to the VLSI circuit designer who is facing increasingly challenging tasks such as designing for low-power, area, speed, etc.

In addition, the genetic operators will be designed to take the circuit aspects of the structural design into consideration. The GA will utilize a given design cell library in the structural design of a multi-input/multi-output logic function. This will be done by allowing unrestricted choice within libraries and using a multi-constrained fitness function which will allow the optimization of hardware aspects such as speed, area in addition to that of achieving the correct logic function(s). The GA will continuously reference the design cell library throughout its genetic evolution as the chromosomes will be processed for fitness calculation. The use of the design library within a multi-constrained fitness frame work is illustrated in Fig. 2 .

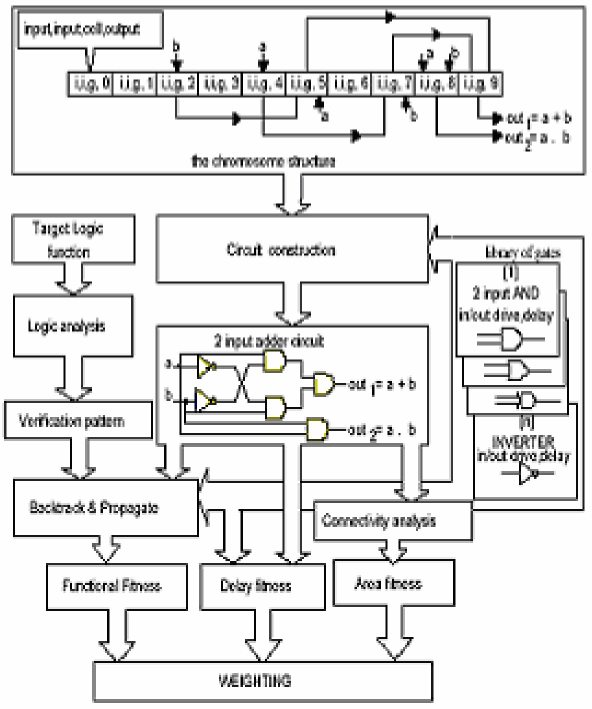

Figure 2: *Multi- Objective fitness evaluation*

The genetic procedure commences by the random initialization of each individual in a population of approximately fifty chromosomes. The roulette wheel method will be used in selecting pairs of individuals for crossover. This will result in two new child individuals contributing to the next population. Next, we will be employing two types of mutation in order to add diversity into the population. The first we will performing at the global chromosome level whereas the second is gene specific. Both crossover and mutation will be discussed in greater detail later in this section. Fitness will be evaluated; see later, for each chromosome in the new population. If any of the individuals in this population satisfies the specifications for logic function and hardware specific criteria then the genetic evolution terminates otherwise it will be continued.

D. Representation

Each chromosome represents a possible design for the target circuit. A chromosome is a sequence of genes, each of which holds information about an individual cell in the design. These are as follows:

1) *Input 1*: A number indicating the identity of a unique net (an internal connection or a primary one) which is input to the particular cell. This is the first input of the cell.
2) *Input 2*: The second input of cell.
3) *Cell-type*: Identifies the cell-type selected from the library.
4) *Output*: A number indicating the identification of a unique net which is output from the cell.

E. Crossover

The crossover function is defined to deal with the multi-output nature of the circuit's synthesis. This is illustrated in Fig. 3. The chromosome1 and chromosome2 will be the parents selected using the Roulette Wheel. The output fitnesses f1 and f2 for outputs *out1* and *out2* respectively in chromosome1 will be compared and the weakest will be selected for a *swap* operation. This operation will replace the set of cells and interconnections contributing to the weak output. The output corresponding to the smallest of f1 and f2, by the corresponding set in chromosome2. Following a crossover operation two repair /reorganization functions might be used. The first will ensure that the circuitries transferred to chromosome1 will be adequately accommodated. Although most of the transferred circuitry will occupy positions in chromosome1 which originally held by the circuitry contributing to the replace output, this function will be required because some extra positions may be necessary for cells which fan-out to more than one cell, contributing to more than one output in chromosome2. The crossover operation duplicates such cells in chromosome2 and then transfers them into chromosome1. The additional positions are usually found by searching the chromosome for redundant circuitry i.e. those not contributing to any output functionality.

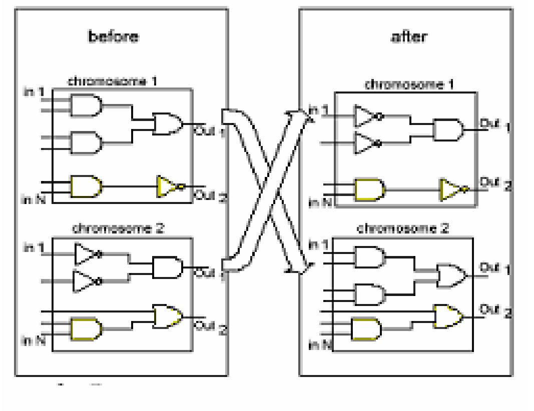

Figure 3: *Crossover operation*

The wiring information in a gene consists of numeric codes which could refer to different wires in chromosome1 and chromosome2. The second repair operation ensures that codes associated with wires to be placed in chromosome1 are not duplicated in the genes of this chromosome hence incorrectly altering the functionality of the circuit. If such duplication exists, then this repair operation associates new codes with the genes of the transferred circuitry. Crossover is performed at 50% rate.

F. Mutation

Two mutation operations will be employed. These are as follows:

Global Mutation: This will be performed at the global chromosome level. Where a gene, at random will be selected and an arbitrary number of its parameters will be selected. A single parameter or all parameters will be altered to a random value according to the position of each parameter in the gene. For example, if cell-type and *in1* will be selected then a random cell in the library will be chosen for cell-type and a random net will select for *in1*. This mutation will be performed at a rate of 1%.

Circuit Mutation: This is a more *circuit-specific* operation in which sections of the chromosome will be selected based on the fitnesses of each output in the circuit, see later. Circuit mutation will target cells and interconnections contributing to a weak output as mentioned earlier in this section, and tries to alter the cell-types and/or the interconnections in the aim of improving the output. Circuit mutation also monitors delay and area parameters in the circuit and will alter cells, based on their delay and area fitnesses by replacing them with other cells in the library. This mutation will be performed at a rate of 1-2%.

The fitness function: The multi-objective fitness function is a crucial part of the genetic system. The *target logic functions*, which are to be structurally synthesized will be processed by this part in order to obtain an optimum set of input/output patterns. This can test the functionality of each prospective circuit structure defined in a chromosome. A special circuit construction function will be used to encode the chromosome and put it in a form where it could be analysed for aspects such as connectivity and redundancy.

The patterns, above will then be used to evaluate the functionality of the constructed circuit through a sequence of operations which involve propagate logic values through the paths in the circuit and backtracking when necessary. This will especially true for the cases of multi-output circuits or where the circuit includes some unused redundancy.

Fitness will be calculated as the weighted sum of three individual sub-fitness for each of the functionality, area and delay parameters. The overall fitness is defined as follows:

Overall-fitness=α.fitness-output+β.fitness-area +λ.fitness-delay,

where *α, β, λ* are the respective weights.

The functionality fitness and output-fitness are the *normalized* fitness for all outputs in the circuit and will be obtained as follows: The fitness of each output will be calculated by the propagation of input vectors from the target function truth table throughout the circuit and evaluating the output. The number of correct matches with the truth table will be used as a measure of fitness for the particular output considered. This can be repeated for each output. Hence, *output fitness* for an *n* output circuit is calculated as follows:

$$\text{Overall-fitness} = \frac{\sum_{i=1}^{n} foi}{n}$$

where foi is the number of matches for the the output.

The delay fitness:

This is calculated as the sum of the delays in the critical path. The longest path in circuit of the circuit.

Area fitness:

This is calculated as the sum of the areas of the active cells in a chromosome. Active cells are those which contribute to the functionality of the circuit output since occasionally the GA produces some redundant circuitry.

$$\text{Area fitness} = \sum_{i=1}^{n} (cellarea)i$$

where n=number of active cells.

Both Delay and Area fitnesses will be calculated with continuous reference to the design library. The above fitness framework can accommodate additional sub-fitness functions for considering other aspects in a given circuit. The system flow which we are going to design is as shown below.

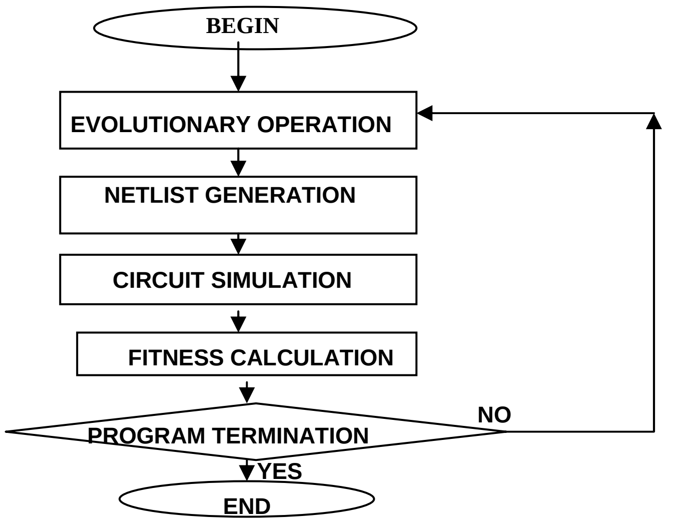

***Figure 4*: System Flow**

IV. REFERENCES

[1] T. Arslan, D. H. Horrocks and E. Ozdemir "Structural Cell-based VLSI Circuit Design using a Genetic Algorithm", Electronics Letters Volume 32, Issue 7, 28 Mar 1996 Page(s):651 – 652 Digital Object Identifier.

[2] Mazumder, P. and Rudnick E. (1999), Genetic Algorithm for VLSI Design Layout and Automation, Addisson-Wesley Longman Singapore Pte. Ltd, Singapore.

[3] H.A. Rahim, A.A. A. Rahman, R.B. Ahmad, W. N. F. W. Ariffin, M.I. Ahmad "The Performance Study of Two Genetic Algorithm Approaches for VLSI Macro-Cell Layout Area Optimization"Second Asia International Conference on Modeling & Simulation, 978-0-7695-3136-6/08 , 2008 IEEE

[4] Ning Xu; Feng Huang; Zhonghua Jiang, "Parallel Genetic Algorithm for VLSI Building Block Layout" International Conference on Wireless Communications, Networking and Mobile Computing, 2006.WiCOM-2006. Volume - 4, Issue, 22-24 Sept. 2006 Page(s):1 Digital Object Identifier 10.1109/WiCOM.2006.170

[5] Yiming Li Yen-Yu Cho " Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International" Publication Date: 25-29 April 2006,On page(s): 8 pp.-Current Version Published: 2006-06-26.

[6] Krishnaswamy, D.; Hsiao, M.S.; Saxena, V.; Rudnick, E.M.; Patel, J.H.; Banerjee, P."_Parallel genetic algorithms for simulation-based sequential circuittest generation", VLSI Design, 1997. Proceedings., Tenth International Conference on Volume , Issue , 4-7 Jan 1997 Page(s):475 - 481

[7] Sripramong, T.; Toumazou, C."_*The invention of CMOS amplifiers using genetic programming and current-flowanalysis*" Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on Volume 21, Issue 11, Nov 2002 Page(s):1237-1252Digital Object Identifier 10.1109/TCAD.2002.804109