CHAPTER I

INTRODUCTION

The advancement of VLSI technology that enhances a number of devices twice every year makes the manual design impossible and infeasible. The need to develop new techniques to automatically synthesize a VLSI circuit from a specification is a must and an essence in nowadays competition. Several levels of design and synthesis algorithms have been investigated and developed during the past 30 years [1]. The synthesis on the layout, transistor, and gate levels are well studied and developed. On the other hand, the high level circuit synthesis on the functional unit and on the memory levels are still in its infancy state in some aspects.

1.1. Problem Identification

High-level circuit synthesis is shortly called high level synthesis in this paper. Most of the high level synthesis studies focus on the problem of functional unit scheduling and assignment in order to minimize space and time of the final outcome [4,9]. The space is measured in terms of the number of registers, adders, multipliers, as well as multiplexors. The time is measured by the number of control steps. These studies do not consider the reliability of the circuit and the capability to recover itself from a transient fault [6] that is possibly caused by heat or electro-magnetic field.

The problem of recovery from the transient fault is not a new problem [5,6,12]. On a known architecture computer, the recovery is performed by inserting several

checkpoints and re-executing some instructions after the fault event in *rollback* fashion. Although this technique works well, it cannot be directly adapted to high level synthesis. In high level synthesis, the number of functional units and their assignment to the control steps are not known in advance until the synthesis is completed. Hence, it is impossible to apply the checkpoint insertion used on the known architecture to this high level synthesis with transient fault recovery. A new technique must be developed to cope with this situation.

The problem of high level synthesis with micro-rollback self-recovery was originally studied by Raghavendra and Lursinsap [2]. Then several approaches to this problem have been developed [3,10,11]. These approaches are based on graph models called data-flow and control-flow graphs with functional unit mobility cost function. The obtained solutions trend to be localized somewhere in solution space because they lack the capability to search the space. In addition, it is rather difficult to simultaneously consider several related designing factors controlling the outcome. Among these factors are the number of functional unit, the functional unit areas, chaining, the number of checkpoints, and the length of rollback interval.

Genetic algorithm has been successfully applied in various search and optimization problems, because it is able to search a complex space of problem solutions often involving a tradeoff between two apparently conflicting objectives. In other words, genetic algorithm exploits the best solutions currently available and robustly explores the design space. Therefore, we apply the technique of genetic algorithm to this problem.

1.2. Objectives

The main objectives of this study are the following:

1. To study the problem of high level synthesis with micro-rollback self-recovery.
2. To develop a genetic algorithm to obtain the solution of the problem constrained on the number of functional units, the number of registers, the number of checkpoints and the number of control steps.
3. To design an encoding scheme, crossover and mutation operators suitable for the problem.

1.3. Scope of Work

In this thesis, we have developed and implemented a genetic self-recovery micro-rollback synthesis algorithm and some benchmark problems are solved by using our approach. In solving the problem of self-recovery micro-rollback synthesis, we have considered the size and speed of ALUs, the number of registers and the number of checkpoints during the genetic algorithm optimization process. We have also compared the solutions of the benchmark problems generated by our approach with those by previous approach [14]. Formulation of the self-recovery micro-rollback synthesis in this thesis is defined as follows.

Firstly, we define a data-flow graph which has the constraints on the number of functional units with different sizes and delay times, the number of checkpoints, and the number of control steps. Then we find a new control-data-flow graph (CDFG) that fits the given number of functional units, checkpoints, and control steps. If the new graph cannot be found, we need to find the best graph that will not violate the given constraints. The constraints on multi-cycle functional units and chaining are also

considered. A functional unit can be implemented by a multi-functional ALU in order to save the area.

The thesis is organized into six chapters. Chapter 2 reviews the literatures. Chapter 3 discusses the problem of self-recovery micro-rollback synthesis and the concept of genetic algorithm. Chapter 4 provides details of the genetic algorithm used in our approach. Chapter 5 gives the experimental results and chapter 6 is the conclusion.