CHAPTER II

LITERATURE REVIEW

2.1. Review of Literatures Related to High Level Synthesis with Micro-Rollback Capability

The problem of self-recovery micro-rollback is first introduced and discussed by ***Raghavendra and Lursinsap*** [2]. According to them, an efficient design of micro-rollback and checkpoint insertion can be achieved by considering them during the scheduling and allocation steps. The rollback and recovery scheme is designed to satisfy the constraints on the number of available registers and the maximum allowable recovery time, which assumes that the maximum number of available registers is given, and attempts to minimize the number of recovery points inserted. Moreover, the authors attempted to examine practical ways of arriving at suitable values of the number of registers, the recovery time and the number of retries, in spite of the conflicting requirements.

Orailoglu and Karri have described the problem of recovery point insertion within both minimizing the number of recovery points and minimizing the number of hardware overhead for synthesizing reliable microarchitecture [3,10]. Their proposed synthesis system comprises a heuristic and an optimal subsystem. The heuristic synthesis subsystem has two components: the checkpoint insertion algorithm and the edge-based scheduler.

The checkpoint insertion algorithm investigates appropriate checkpoints by eliminating clock cycle boundaries that either have a high checkpoint overhead or violate the retry period constraint. On the other hand, the edge-based scheduler assigns edges to the clock cycles boundaries, in addition to scheduling nodes to clock cycles.

Blough, Kurdahi and Ohm have continuously considered the problem of automatic insertion of recovery points in recoverable microachitectures [11]. The authors attempted to identify polynomial-time algorithms that provide probably optimal solutions to minimize computation time with a bounded hardware overhead and to minimize hardware overhead with a bounded computation time by transforming the recovery insertion point into a graph problem. Their algorithms take as their input a scheduled control-data flow graph describing the behavior of the system, and they output either a minimum time or minimum cost set of recovery point locations.

2.2. Review of Literatures Related to Genetic Algorithm

Ahmad, Dhodhi and Chen present the use of problem-space genetic algorithm (PSGA) to solve the problem of high-level synthesis consisting of scheduling, allocation and binding [14]. The problem is to map the operations to time steps while maintaining the dependencies, assigning hardware resources (functional units) to each operation, and allocating register to variables. This means a given cost function based on the area of the hardware resources and on the total time of execution is minimized.

Representation of chromosome consists of two parts: priorities of nodes, and the type of functional units and the number of functional units of each version. The selection method is implemented using a biased roulette wheel. The crossover operator is a simple one-point crossover operator in the classical GA. Likewise, the mutation operator is implemented by selecting a gene at random with mutation rate and replacing its value.

Their approach combines using the concept of heuristic problem. For instance, a scheduling and allocation is applied to generate a schedule from a given chromosome by decoding these chromosomes.