Abstract

Validation is a major task in the processor design cycle. Simulation based validation is among the most widely accepted methodologies adopted for verification of large scale industrial designs. Modern simulation based validation attempts to incorporate as much of formal and semi-formal techniques in its ambit. In this thesis, we examine several new approaches to simulation based validation for pipelined processors. We address the problem of pipeline validation from the view point of Instruction Set Architecture (ISA) equivalence, Assertion Based Verification and Test Generation, respectively. We address three problems in this thesis.

Firstly, we address the problem of validating and debugging a pipeline simulator from the specific perspective of instruction scheduling. We propose a general framework that applies to most standard pipeline models. The framework does not need any formal specification of the pipeline logic and hence can be readily integrated into a simulation and iteration-based architectural design space exploration process. We propose a concept of semantic equivalence between two simulations called D* equivalence which effectively captures the dataflow between instructions through registers. We then proceed to propose an algorithm which decides this equivalence in time polynomial in the number of instructions executed and the number of registers. We implement the algorithm and demonstrate how the framework facilitates fault location and debugging.

Secondly, we propose a specification logic called Temporally Attributed Boolean (TAB) Logic for Assertion Based Verification which allows us to: (i) represent assertions succinctly, (ii) incorporate data-orientation and (iii) associate timing to design intentions. TAB Logic allows us to write specifications functionally linking system variables from different temporal contexts. We present examples to show the motivation for this logic especially in the context of high level modeling of complex real time systems. We formally define TAB Logic, formulate the problem of verification on a simulation trace and present efficient algorithms to check TAB assertions, both offline and online. We present results of application of TAB Logic for Instruction Semantics, Bus Transaction and Interrupt mode behavior Verification of a bus integrated pipelined processor core implementation.

Finally, we present a directed test generation framework to generate test cases for microprocessor pipelines targeted towards user defined scenarios. The framework has three important components: a language for description of the user specific scenario, pipeline specification and test generation algorithms. We firstly introduce SDL (Scenario Description Language) which is defined using predicates and is used for specifying the desired user scenarios. We present two approaches for generating test cases for pipelined processors which include generation of hazard free and hazard based test cases. We present test generation results for three different configurations of DLX pipeline.

This thesis advances the field of pipeline verification by allowing the user to validate pipelines in the presence as well as absence of detailed pipeline specification, at various refinement stages in architecture design. The approach based on ISA equivalence works on the availability of register accesses and does not

need a detailed pipelined specification, which is often not readily available. Writing assertions on the other hand needs a detailed pipeline specification as they specify timing and data-oriented behavior of the instructions, interrupts and bus transactions. Generating the test cases for complex user defined scenarios also require the user to be aware of the pipeline architecture. We believe that the proposed methods cater to specific verification needs in the design flow of pipelined processors.

Keywords: Verification, Pipeline, Equivalence, Simulation, Temporal Logic, Test Generation