

SystemVerilog Assertion for Microarchitecture Education considering Situated Nature of Learning : A Senior Project

Ryuichi TAKAHASHI

Faculty of Information Sciences
Hiroshima City University
Hiroshima, 731-3194, Japan
ryuichi@ieee.org

Yoshiyasu TAKEFUJI

Faculty of Environment and Information Studies
Keio University
Kanagawa, 252-8520, Japan
takefuji@sfc.keio.ac.jp

Abstract—SystemVerilog assertion (SVA) is a way to express properties that are expected to be true in a design described in Verilog HDL IEEE1364 standard. We have already reported that legitimate peripheral participation (LPP) works very well for the fine grain microprocessor design education on FPGA where the heart of the system is chosen as the way-in which is the first step for the observation in LPP. We have demonstrated its effectiveness on superscalar design education, while the prior pipeline design education failed. The failure was caused by the top down design methodology guided in the education for the pipelining which appeared to be too difficult. Appropriate scheme to observe the heart of the pipelining is needed. We have found that SVA plays a key role where two senior students succeeded to design pipelined RISC having 3 stages and pipelined CISC having 4 stages in 2 months. White box test by using SVA enables the two senior students to observe the heart of the pipelining very effectively.

Keywords-SystemVerilog; Assertion; Legitimate peripheral participation ; Pipelining; Microarchitecture education;

I. INTRODUCTION

An effective verification methodology has become critically important as the scale of the system has become very large. SystemVerilog is a standard, where the SVA [1] is used to describe the properties of the system under verification expressed in Verilog HDL.

Many strategies have been proposed for the usage of SVA. For example, SVA should be used to check corner cases which are difficult to verify on prior simulations without SVA. Besides the effectiveness for the industries, we found that the description of the heart of the system in SVA is very effective also for the educational point of view. Two senior students succeeded to design pipelined RISC and CISC [2] only in 2 months which was very difficult in the prior education.

In the next section, we describe the LPP which appeared to be very effective in superscalar design education [4]. Then we describe a successful senior project for the pipelining which resulted in a failure in prior education without SVA.

II. SITUATED NATURE OF LEARNING

The legitimate peripheral participation (LPP) is a model introduced by Lave and Wenger [3] by investigating apprenticeships. They named the first period of observation “way-in” to construct a first approximation of the products.

We chose the instruction issue logic for the superscalar processor as the way-in for the LPP in an educational environment for junior students to learn the mechanism of the computer, and succeeded [4], since it is the logic to be tuned at the final phase of the superscalar design and is the heart of the system. The idea to use the reversed production steps is by the LPP. We extended the idea to the choice of the way-in which should be the central part of the design.

III. A SENIOR PROJECT

Since the cause of the failure of the prior pipelining education seemed to be the lack of the opportunity to observe the heart of the system without having the responsibility for the completion of the design, we decided to try the very same subject with SVA which was expected to be a mean to observe the central part of the system. The subject was to modify an example description of a pipelined CISC having 3 stages named CISC-3 into faster RISC having the same 3 stages or more sophisticated CISC having 4 stages.

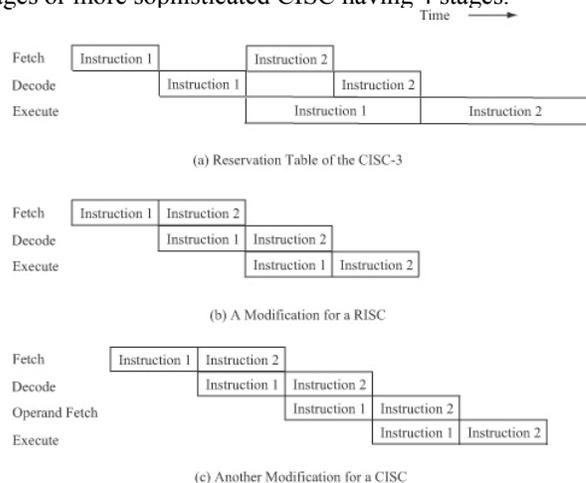


Figure 1. The reservation tables for the design

Figure 1 (a) illustrates the reservation table for the CISC-3. The reservation table for the new RISC is illustrated in figure 1 (b). The new sophisticated CISC must have a special stage for the operand fetch as illustrated in the figure 1 (c).

In the prior pipelining education, even a RISC with three-address instructions was difficult to design for the junior students. Few of the students succeeded to complete the CISC having 4 stages. In our new trial, the students were with the SVA.

A. RISC with three address instructions

Figure 2 illustrates the block diagram drawn by a senior student for the RISC with three-address instructions. The key mechanism is the pipeline interlock to detect the hazard and resolve it.

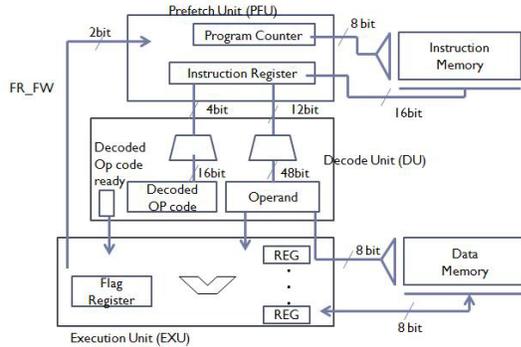


Figure 2. A New RISC block diagram

The RISC was written in 784 lines for 9 modules in Verilog HDL to be implemented by 5,270 gates on FPGA.

B. CISC having 4 stages

Figure 3 illustrates the block diagram drawn by another senior student for the CISC having 4 stages.

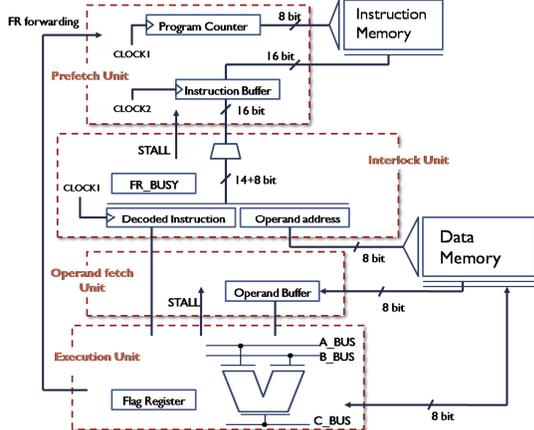


Figure 3. A New CISC block diagram

The CISC was written in 727 lines for 10 modules in Verilog HDL to be implemented by 1,898 gates on FPGA.

IV. SYSTEMVERILOG ASSERTIONS

At first, two students continued to design their machines with ordinary simulators. When they found some difficulties, they started to use an SVA system.

A. PIPELINE INTERLOCK

Figure 4 illustrates an assertion example written by the senior student who completed the RISC. The letters FR stand for Flag Register. FR_FW is for forwarding the contents of FR. “STATE==2” means that the prefetch unit is in the WAIT state waiting for the new value of the FR.

“STATE==1” means that the unit is in the FETCH state to update the contents of the program counter.

```

module pfu_prop(CLOCK1, RESET, STATE, FR_FW, FR);
input CLOCK1,RESET;
input [1:0] STATE,FR_FW,FR;
core_fetch_fr_forward: assert property
  (@(posedge CLOCK1)(STATE == 2) ##1 (STATE == 1)
   |-> $past(FR_FW) == FR);
endmodule

```

Figure 4. An assertion for the Flag Register

B. OPERAND FETCH

Figure 5 illustrates an assertion example written by the senior student who completed the CISC. He decided to forward the data to be stored to the operand buffer in the case when the new instruction uses the data on the same address.

In most cases, the assertion which resembles original Verilog HDL description is useless since it is likely to be satisfied. The student wrote the assertion to check the stimulus to the forwarding of the data on the same address.

```

module cpu_prop(CLOCK1, STORE_DATA, OPERAND_BUFFER,
  INSTRUCTION_BUFFER, STORE_ADDRESS);
input CLOCK1;
input [7:0] STORE_DATA,OPERAND_BUFFER,STORE_ADDRESS;
input [15:0] INSTRUCTION_BUFFER;
core_store_data: assert property
  (@(posedge CLOCK1)
   $past(STORE_DATA) == OPERAND_BUFFER
   |-> $past(INSTRUCTION_BUFFER[7:0], 2)
   == $past(STORE_ADDRESS));
endmodule

```

Figure 5. An assertion for the Operand buffer

V. COCLUSION

A successful senior project to complete the pipelining with SVA has been demonstrated. The past difficult problem for undergraduate students was resolved by observing the heart of the system through the assertion based verification environment as the white box test. We believe that the SVA works well in microarchitecture design education by showing the heart of the system appropriately.

ACKNOWLEDGMENT

The authors would like to thank senior students Ryota Tomioka and Yasuo Matsuoka for their highly motivated participation in the project. We also would like to thank Cadence Design Systems, Japan for their SVA system.

REFERENCES

- [1] Harry D. Foster, Adam C. Krolnik and David J. Lacey, Assertion-Based Design 2nd ed, Kluwer Academic Publishers, 2004
- [2] Peter M. Kogge, The Architecture of Pipelined Computers, Hemisphere Publishing Corporation, 1981
- [3] Jean Lave and Etienne Wenger, Situated learning, Legitimate peripheral participation, Cambridge university press, 1991
- [4] Ryuichi Takahashi and Hajime Ohiwa, “Legitimate Peripheral Participation on FPGA for Fine-Grain Microprocessor Design Education,” Proc. MSE’2005, pp.99-100, 2005