UNIVERSITY OF TEHRAN

Electrical and Computer Engineering Department

Core-Based Embedded System Design
ECE 160 - Spring 1403-1404
Computer Assignment 1 – Part A: A Tiny System
Due Date: Esfand 24

Description:

In this assignment, you will simulate a tiny embedded system at two different abstraction levels using VHDL and SystemC. The system is based on the AFTAB processor core, which is a RISC-V processor implementing the RV32IM ISA. AFTAB is an in-order, single-issue core with sequential stages, supporting the base integer instruction set (RV32I) and the multiplication extension (RV32M). The VHDL code for AFTAB is available on GitHub: https://github.com/RHESGroup/aftab. You can also find the VHDL codes in the "AFTAB_Core/Hardware/src" folder.

Part A (Due Date: Esfand 24):

1. **Simulate the AFTAB Core in ModelSim (RTL with VHDL):**
    a. **Write a C++ Program:** Write a C++ program to calculate the factorial of n = 10. You may use the provided C++ code in the "AFTAB_Core/Software/src" folder as a reference.
    b. **Generate Assembly Code:** Use a RISC-V compiler to convert your C++ code into RISC-V assembly code. You can use an online compiler like Godbolt: https://godbolt.org/. Paste your C++ code into the editor, select a RISC-V target, and generate the assembly code. *(Note: Ensure your VPN is active if required.)*
    c. **Convert Assembly to Machine Code:** Use an online assembler like Venus: https://venus.kvakil.me/ to convert the assembly code into machine code. Paste the assembly code into the editor, dump the machine code, and save it in a new text file. *(Note: Ensure your VPN is active if required.)*
    d. **Load Machine Code into HDL Memory:** Copy the machine code and paste it into the memory file of the HDL memory model. Follow the template provided in the "dram_dump.txt" file located in the "AFTAB_Core/Software/sim/" folder.

e. **Simulate in ModelSim:** Create a new project in ModelSim and add all the HDL codes. Simulate the system to calculate the factorial of n = 10. *(Note: The memory model and testbench are located in the "AFTAB_Core/Software/sim/" folder.)*
f. **Verify the Result:** Verify that the result is correctly written to the expected memory location. You can also check the processor's register file to confirm the result. *(Note: Refer to the memory model HDL file to identify the memory-mapped addresses.)*
g. **Report Execution Time:** Measure and report the execution time as a processor performance metric. *(Note: The code runs in a loop, so calculate the time for only one iteration.)*

2. **Change the Application to Matrix Multiplication:**
   h. **Write a C++ Program for Matrix Multiplication:** Write a C++ program to perform 2×2 matrix multiplication. Repeat steps a. to g. to simulate the system for this new application and report the execution time.
   i. **Rewrite the C++ Code without Multiplication:** Assume the processor does not support multiplication. Rewrite your C++ code by replacing the multiplication operation with a function call that calculates the product using repetitive addition in a loop. Add this function to your C++ code. Repeat steps a. to g. to simulate the system for this new application and report the execution time.
   j. **Compare Execution Times:** Compare the execution times of the two matrix multiplication implementations (with and without hardware multiplication support). Analyze the performance difference and explain the results.

Deliverables for Part A - Due Date: Esfand 24

- C++ Code:
  - Factorial calculation.
  - Matrix multiplication (original and modified for no multiplication support).
- Assembly and Machine Code:
  - Generated assembly and machine codes for all programs.
- Simulation Results:
  - Screenshots or logs of the simulation results, including memory or register values.
- Execution Time Report:
  - A table comparing the execution times for all implementations.
- Put all codes (C++, Assembly, Machine, and HDL) and the report file (including only screenshots and the execution time report) in a folder and zip it. Name your zipped file using the following format:

'YourName_YourFamily_Last5digitOfYourStudentID_CA#1-A'

Part B (Due Date: Farvardin 15):

…