EECE 525 Project 1

HDL Implementation of Dynamic Branch Predictors

Total Points: 100

This is a group project. Only teams of two are allowed to work on this project. Each team should submit a project report by 11:59 PM on March 1st ,2024.

Objective

This project is intended to help you better understand the architecture and performance of modern branch predictors. In this project, you will implement several dynamic branch predictors in System Verilog and compare their performance using simulations. In modern processors, branch predictors are used to overcome control hazard limitations. Dynamic branch predictors use information from the behavior of the program at run time to predict the outcome of a branch.

Project Overview

Your task in this project is to develop a hardware implementation of several dynamic branch predictors in System Verilog and evaluate and compare their performance using an MIPS testing program. The following branchpredictors will be implemented and tested:

1. 1-bit Branch Predictor:
   The 1-bit predictor uses a 1-bit branch history table, indexed by the lower bits of PC, to predict the behavior of the branches (Taken or Not Taken).
2. 2-bit Branch Predictor (2-bit counters):
   The 2-bit predictor uses a 2-bit branch history table, indexed by the lower bits of the PC, to predictthe behavior of the branches. The 2-bit predictor is a 2-bit counter that counts upward in Taken branches and saturates at state three and counts downward in Not Taken branches and saturates atstate 0. The states are usually referred to by (Strong Not Taken 00, Weak Not Taken 01, Weak Taken 10, and Strong Taken 11)
3. 2-bit global history predictor with 2^2 2-bit counters:
   The history predictors keep track of the behavior of previous branches and use it to predict the behavior of the current branch. For example, the 2-bit history predictor uses the history of the last two branches to choose between 4 different predictors, each with a 2-bit saturation counter.

Implementation

You will be responsible for the branch predictor implementation only. You don't need to implement the CPU Datapath as part of this project.

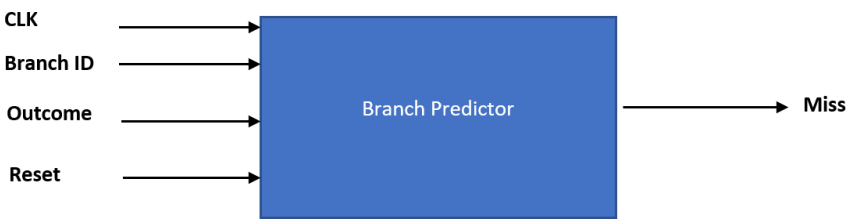

Figure 1: Branch Predictor Top Module

As shown in Figure 1, inputs and outputs to the predictor will include:

- The branch ID: A 32-bit number representing the branch address.
- The correct branch outcome: A one-bit value, 0 indicating Not Taken branch or 1 Indicating Takenbranch.
- Clk: the clock signal input
- Reset: an input to initialize the branch history table to 0.
- Miss: A 1-bit signal output is used to indicate whether the branch prediction was correct or not.

Part 1: One-Bit Predictor:

1. Develop the 1-bit branch predictor in Verilog using behavior or structural description.
2. Use the provided testbench on Canvas to test your program and collect the misprediction rate.
3. Take a snapshot of the simulation waveform and the test results.