Assignment 3
Branch Target Buffer
Due Wednesday, November 14, before class

The goal of this assignment: To design a branch target buffer with 2-bit predictive capability in Verilog and demonstrate that it works. To appreciate the complications of hardware support for least-recently-used replacement policies.

The BTB will have 64 entries, each with the following structure:

Branch Instr Addr	Branch Pred Stats	BranchTarget	LRU Data

where
- Branch Instruction Address is a 32-bit address (the tag),
- Branch Target Address is a 32-bit address,
- Branch Prediction Statistics is a 2-bit saturating counter that counts up when the target is selected, and counts down when the fall-through code is selected, and
- LRU Data is a 10-bit field used to replace entries in the BTB.

There will be a global counter called LRU Time.

Inputs to the BTB are as follows:
- Operation -- defined below.
- Instruction PC (32 bits).
- Branch Target Address (32 bits)

Outputs of the BTB are as follows:
- BTB Hit/Miss {0 = miss, 1 = hit}
- BTB Prediction {0 = fall-through, 1 = target}
- Branch Target Address (32 bits)

Operation: In selecting an encoding of these values, give some thought to efficiently presenting the information to the BTB to make its circuit design simpler. Note that entries that predict fall-through code are stored in the BTB.

Clear -- sets all bits in the Branch Instruction Address field and the LRU Data field to 0. Initializes LRU Time to 1.

Lookup -- using the Instruction PC as input, performs an associative search of the BTB on the Branch Instruction Address tag. If no match is found, return BTB Hit = 0. Otherwise return BTB Hit = 1, return the most significant bit of the Branch Prediction Statistics field as the BTB Prediction, return the contents of the Branch Target Address field, assign LRU Time to LRU Data and increment LRU Time.

Verify Fall-Through -- Locate the branch in the table using the Instruction PC. If it is found, decrement the Branch Prediction Statistics field and set BTB Hit.

Verify Target -- Locate the branch in the table using the Instruction PC. If it is found, increment the Branch Prediction Statistics field, store the Branch Target Address input bits into the Branch Target Address field and set BTB Hit.

Insert Fall-Through -- Find the BTB entry with the smallest LRU Data field, replace its Branch Instruction Address field with the Instruction PC, set its LRU Data field to LRU Time and increment LRU Time, and set its Branch Prediction Statistics field to 1.

Insert Target -- Find the entry with the smallest LRU Data field, replace its Branch Instruction Address field with the Instruction PC, replace the Branch Target Address field with the Branch Target Address bits, set its LRU Data field to LRU Time and increment LRU time, and set its Branch Prediction Statistics field to 2.

The task: Create a behavioral Verilog design of the BTB and demonstrate its correct operation.

For your design, answer the following questions:

1) What is the rationale behind the encoding you chose for the operation field?
2) As specified, the LRU field simply counts up continuously. This strategy has some potential difficulties, such as the fact that it could eventually overflow. This will cause a branch to be entered into the table at a low LRU Data value, which will make it a victim on the next entry. How might this problem be fixed? Justify that your solution is efficient.

What should be turned in:

1. A commented BTB design.
2. Evidence of its correct operation, i.e., a BTB tester and output indicating that your have tested the BTB thoroughly.
3. Answers to the questions above.