CSE471 A96 Snyder

# Assignment 3

**Goal:** Design a verilog behavioral description of a TLB for a pipelined microprocessor and demonstrate that it works.

The TLB will have 32 64-bit entries and translate virtual to physical addresses for 4K pages. Thus, the high-order 20 bits of a 32-bit virtual address are presented to the TLB together with a 20-bit User-Id, and if the translation is found in the TLB and valid, the high-order 20 bits of the 32-bit physical address are returned (hereafter called the paddress). The entry in the TLB will have the following form:

| V | U | User-ID | Vaddress | Paddress |
|---|---|---|---|---|

where V is the valid bit, U is unused, User-Id, Vaddress and Paddress are all 20-bit fields. The outputs of the TLB are:

| | |
|---|---|
| Hit/No Hit | -- The tag did/did not match anything. |
| Valid/Not Valid Bit | -- The entry is/is not valid |
| Physical Address | -- The 20-bit Paddress Field |

The TLB will implement the following functionality, where the virtual address and User-Id bits form the tag:

- Lookup -- The tag is "associatively looked-up". If a match is found, the Hit is set to true, the V-bit and the Paddress are returned; otherwise the Hit bit is set to 0.
- Invalidate -- The tag is "associatively looked-up". If a match is found, its V bit is set to 0, and the Hit bit is set to true; otherwise the Hit bit is set to 0.
- Flush -- All entries with the given User-ID are set to invalid, i.e. their V bit is set to 0. The Hit bit is set if any are found, otherwise it is not.
- Set -- An invalid entry is located, its User-ID bits are filled, and its Vaddress bits are filled; the entry remains invalid. The Hit bit is set to 1 unless no invalid entry was found. If on the next cycle the Set operation is also requested, then bits of the virtual address lines are taken to be the physical address bits of the entry set on the previous cycle, and are assigned to the Paddress field; the entry is set valid. The Hit bit is the same as on the previous cycle. Otherwise the specified operation (Lookup, Invalidate, Flush) is performed.

An associative search is implemented in the behavioral specification by simply looping through the entries looking for a match. An array of memory bits (local to the TLB module) is declared as

```
reg   [63:0] memory [0:32];
```

The ith 64-bit quantity is referenced `memory[i]`. For convenience, work with an array that is only 4 entries long, and define it to 32 when the design is complete. Notice that the clocking has to be considered when constructing this TLB in order to implement the two-cycle set operation.

**Tasks:** Decide what inputs must be presented to the TLB, and their encoding. Define the TLB module to implement the behavior described above. Implement a tester module to exercise the TLB module, illustrating all of the operations and the significant cases of their operation.

**Turn In:** Submit the two modules and the output of a simulation proving that the TLB has been fully tested.

**Due:** Both parts are due by 5:00pm on 4 November 1996 by automatic submission.