

# EECS 470 Lab 6 Assignment

---

## Note:

- The lab should be completed individually.
- The lab must be submitted by 11:59PM on February 26th.

## 1 Introduction

In this lab you will be designing a simplified, generically-sized Content Addressable Memory (CAM). A CAM has the property that, when queried with a value, it will search through all of its contents in parallel, returning the index at which that data exists (if it does). Our simplified CAM has the following module header:

```
module CAM #(parameter SIZE=8) (
    input clock , reset ,
    input enable ,

    input COMMAND command,

    input    [31:0] data ,

    input    [$clog2(SIZE)-1:0] write_idx ,

    output   [$clog2(SIZE)-1:0] read_idx ,
    output   hit
);
```

CAM should have the following functionality:

- The design will maintain a set of "SIZE" (default of 8) 32-bit memory elements, which should only be updated on the positive edge of "clock".
- When "reset" is high on the rising clock edge, all entries in memory should be invalidated.
- If "enable" is high and "command" is set to "WRITE" (see sys\_defs.vh), CAM should store the value of "data" to memory address "write\_idx" and validate it. If the index is out of bounds (larger than "SIZE-1"), CAM should not modify its memory.
- If "enable" is high and "command" is "READ", CAM should write the lowest location of valid memory whose value equals "data" to "read\_idx". If the memory does not contain "data", "hit" should be set low. Otherwise, it should be high.

Before running 'make', set the "SIZE" by typing 'export CAM\_SIZE=[i]' in your terminal, replacing '[i]' with the appropriate integer. Your design should function correctly for any value of "SIZE". It is up to you how to implement this functionality. We recommend using a "for" loop inside of an "always" block. Remember that in a procedural block, all assignments happen simultaneously. If multiple assignments are made to the same variable, only the "later" one (the one listed furthest down in the block, or the furthest iteration of a "for" loop) takes effect.

Your design should pass the testbench after synthesis as well, however you needn't worry for this lab about clock period. Just make a design that works.

## 2 Check Off

A script has been provided (check.sh) that checks the correctness for a few different test SIZES. Once you are passing all those cases, submit the lab using the standard submission script (/afs/umich.edu/user/m/o/moorthya/Public/470labsubmit -l6 lab6):