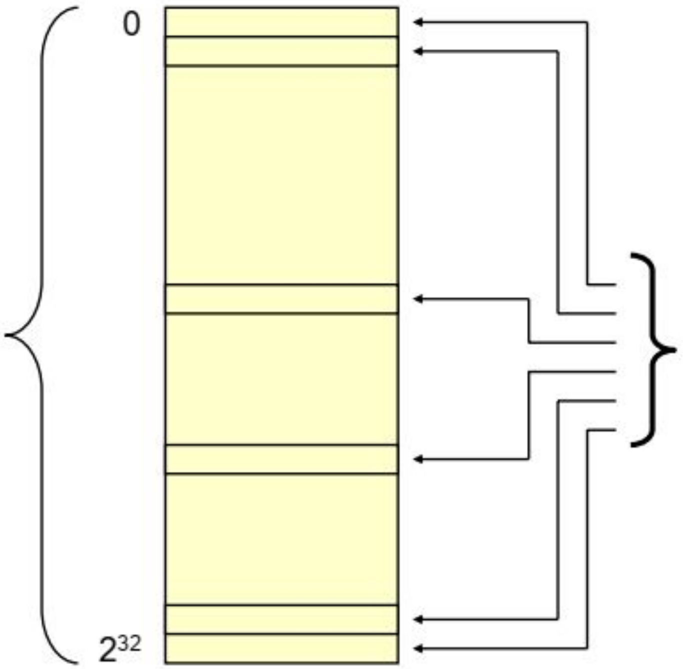
LAB 1: Sparse Memory

- How to model a memory with a large address space?
  - For example a 32 bit address
  - Impractical to allocate the entire memory
- One technique: describe a sparse memory model
  - Associative arrays are a way to implement a sparse memory model

LAB 1: Sparse Memory



Huge memory space Inefficient to model statically!

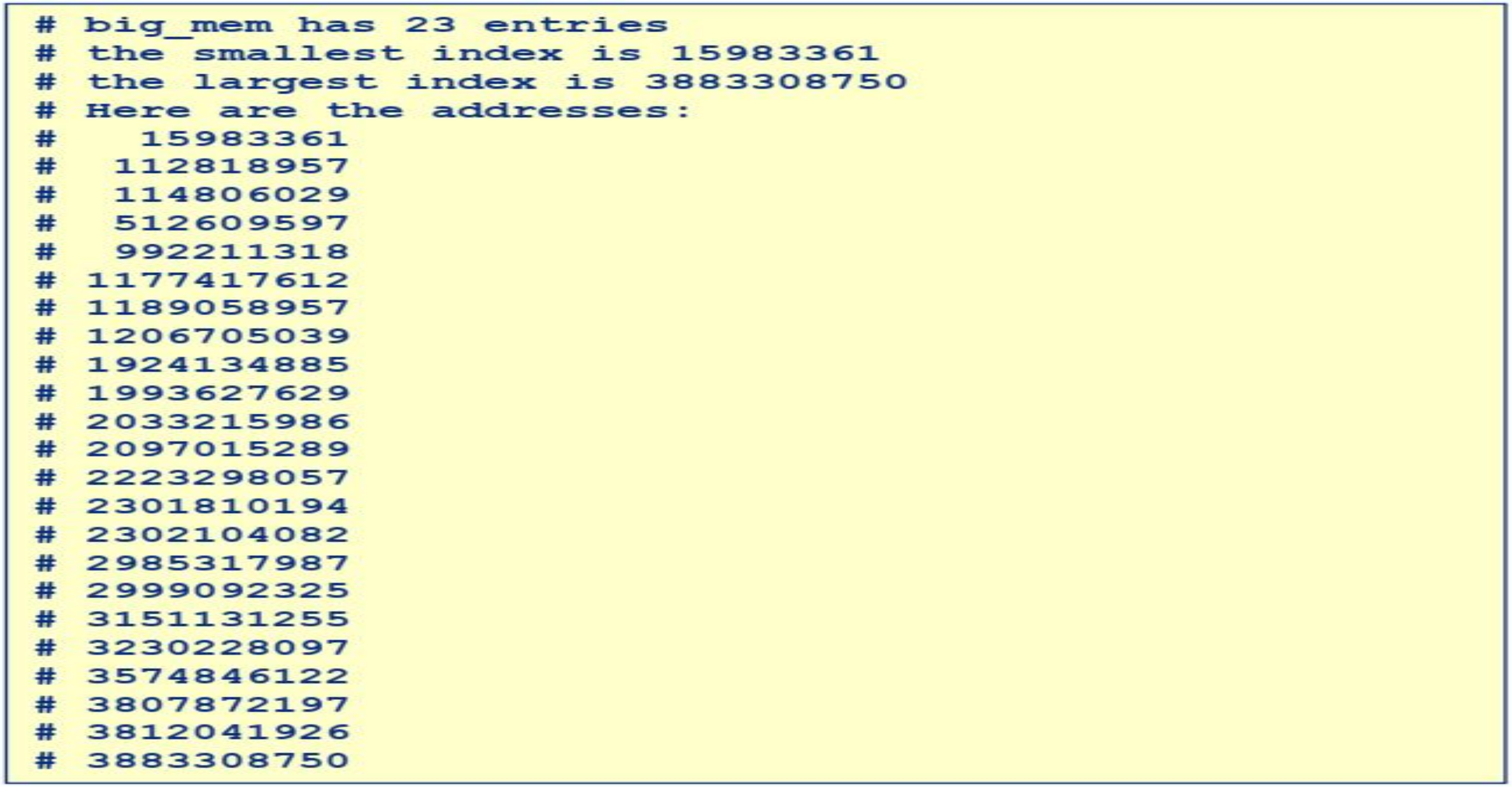
Sparse model allocates memory only for addresses actually written / read!

LAB 1: Sparse Memory

- Instructions
  - In file sparse_mem.sv edit the module sparse_mem
    - Create an enumerated type called "boolean" ("FALSE" and "TRUE“)
    - Create an associative array called "big_mem"
      - – data type is boolean
      - – index type is an unsigned 32 bit value
      - – Default value of each entry should be false
  - Write to big_mem
    - Write a random number of entries (max 25), each at a random location
    - Use $random() to generate random numbers
    - Write the value TRUE at these random locations
  - Display the following information about big_mem:
    - How many entries it has with the value TRUE
    - What is the smallest index with the value TRUE
    - What is the largest index with the value TRUE
    - The index value of all entries with the value TRUE

LAB 1: Sparse Memory

Sample output



```
# big_mem has 23 entries
# the smallest index is 15983361
# the largest index is 3883308750
# Here are the addresses:
#   15983361
#  112818957
#  114806029
#  512609597
#  992211318
# 1177417612
# 1189058957
# 1206705039
# 1924134885
# 1993627629
# 2033215986
# 2097015289
# 2223298057
# 2301810194
# 2302104082
# 2985317987
# 2999092325
# 3151131255
# 3230228097
# 3574846122
# 3807872197
# 3812041926
# 3883308750
```