

```
1 // Eugene Ngo
2 // 4/20/2023
3 // CSE 469
4 // Lab 2, Task 1 and 2
5
6 /* imem is the read only, 64 word x 32 bit per word instruction memory for our processor.
7 ** Its module is written in RTL, and it strongly resembles a ROM (read only memory) or LUT
8 ** (look up table). This memory has no clock, and cannot be written to, but rather it
9 ** asynchronously reads out the word stored in its memory as soon as an address is given.
10 ** The address and memory are byte aligned, meaning that the bottom two bits are discarded
11 ** when looking for the word. One important line to note is the
12 **     initial $readmemb("memfile.dat", memory);
13 ** which determines the contents of the memory when the system is initialized. You will
14 ** alter
15 ** this line to use programs given to you as a part of this lab.
16 */
17 // addr - 32 bit address to determine the instruction to return. Note not all 32 bits are
18 // used since this
19 //     memory only has 64 words
20 // instr - 32 bit instruction to be sent to the processor
21 module imem(
22     input logic [31:0] addr,
23     output logic [31:0] instr
24 );
25     logic [31:0] memory [63:0];
26
27     // modify the name and potentially directory prefix of the file within to load the
28     // correct program and preprocessing
29     initial $readmemb(
30         "C:\\Users\\egeen\\Desktop\\School\\EE 469\\Lab\\Lab 2\\memfile2.dat,"
31         memory);
32
33     assign instr = memory[addr[31:2]]; // word aligned, drops bottom 2 bits
34 endmodule
```