

Help

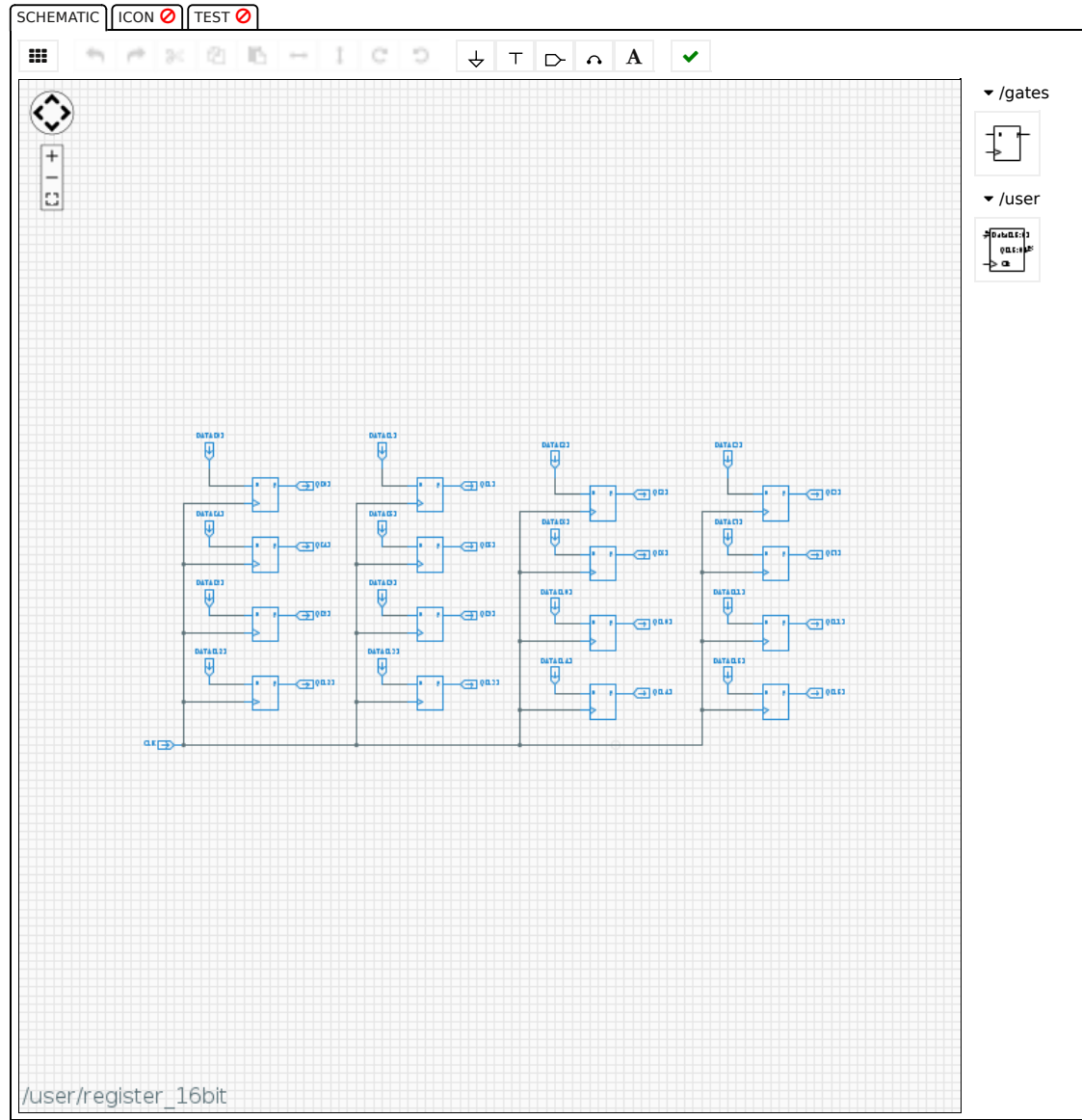
Design a 16-bit register using 16 individual flip-flops, all clocked by a common clock (CLK) and test it using the provided test file.

Remember not to drag the component itself (located in the /user/ section of the parts bin) onto the schematic, or you'll get a recursive error!

Test the 16-bit register with the included test vector and after passing the tests, save this as a library component.

16 BIT REGISTER (1/1 point)

Module: /user/register_16bit



Click component to select, click and drag on background for area select, shift-click and drag on background to pan

[jade 2.2.40 \(2015 © MIT EECS\)](#)

Check

Hide Answer

Help

Show Discussion

New Post

Design a loadable register for one bit.

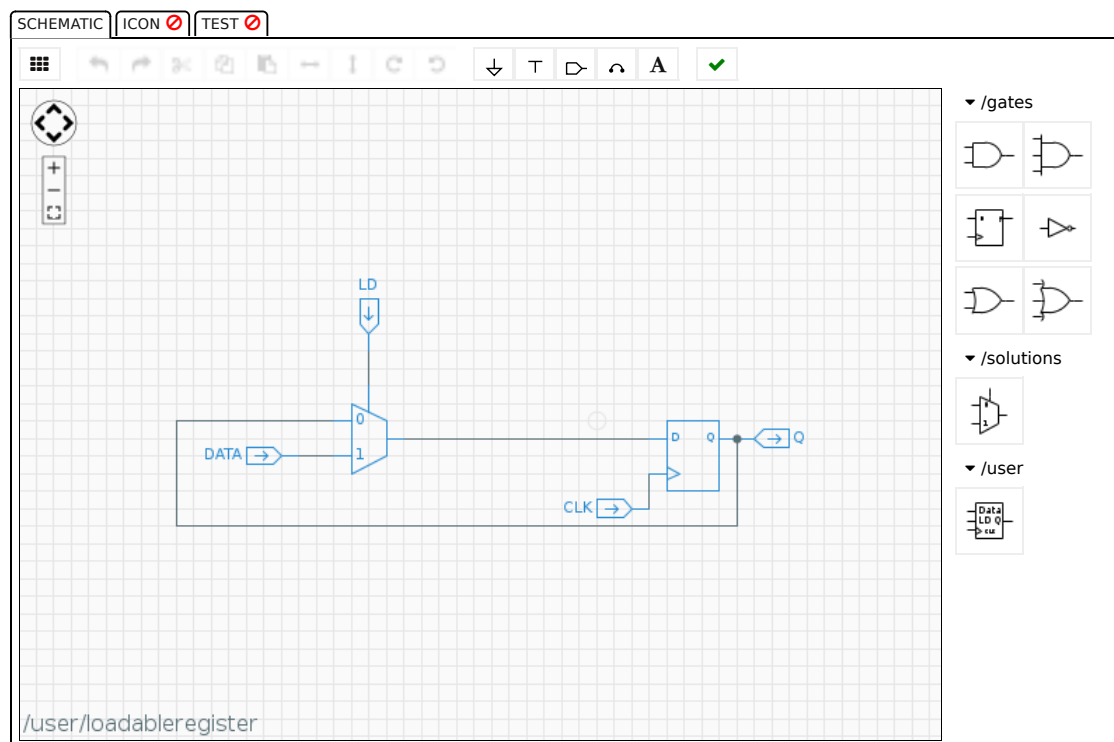
At each positive edge of the clock cycle, a loadable register loads either the input Data value when LD is 1, or it loads the previous Q value when LD is 0.

Remember not to drag the component itself (located in the /user/ section of the parts bin) onto the schematic, or you'll get a recursive error!

Test it and save it to your library.

LOADABLE REGISTER (D FLIP-FLOP WITH LOAD) (1/1 point)

Module: /user/loadableregister



Click component to select, click and drag on background for area select, shift-click and drag on background to pan

[Jade 2.2.40 \(2015 © MIT EECS\)](#)

Check

Hide Answer



Design a 16-bit loadable register using 16 loadable registers (D flip-flops with load capability). The LD input serves as the select input for all 16 muxes. When LD = 1, the 16-bit input is clocked into the flip-flops at the rising clock edge. When LD = 0, the current 16-bit output is retained.

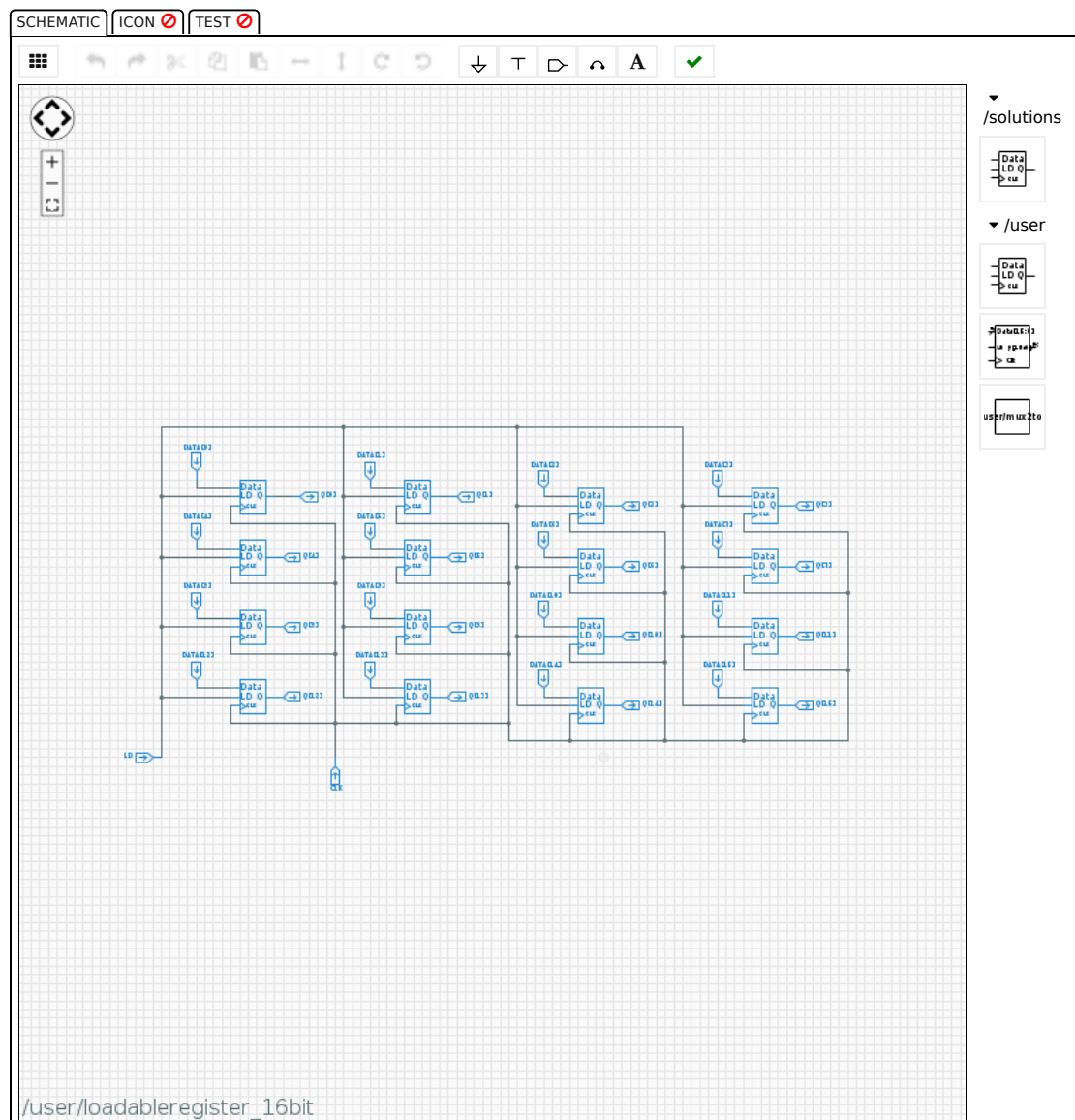
You will need to load "/user/loadableregister".

Remember not to drag this 16-bit loadable register component itself (located in the /user/ section of the parts bin) onto the schematic, or you'll get a recursive error!

Test your design using the provided test file and save this as a library component.

16 BIT LOADABLE REGISTER (1/1 point)

Module: /user/loadableregister_16bit   



Check

Hide Answer

Help

Show Discussion

New Post

This lab will familiarize you with the memory construct in Jade.

At the top of the schematic tab toolbar, there is a new icon, a text label "MEM". It is shown here.

MEMORY TEST LAB (1 point possible)



In this lab, we will instantiate one memory and load its contents so you will be familiar with the process.

Memory Parameters

A memory in Jade is a component that stores data. It has two parameters:

- Address width - This indicates the number of entries in the memory. A memory with an address width of 4 will have 2^4 entries (16).
- Data width - This indicates the size of each entry. Set to 32, this would mean every entry stores 32 bits.

You can set the parameters by double clicking on the memory itself, showing the Properties window. A memory set with the above parameters will indicate it on the schematic with the text 4x32 appearing.

Instantiating A Memory

To instantiate a memory, click and drag from the MEM icon location on the toolbar to the schematic. Line it up with the terminals in the schematic.

The memory has 4 inputs on the left side:

- A (Address)
- OE (Output Enable)
- WE (Write Enable)
- CLK.

And one input/output port on the right:

- D (Data).

When OE is set to 1, and Address is set, the data port D will output the data of the entry at the address given.

We can specify the contents by pasting data into this field. This is best done by copying and pasting data in from a text file. It needs to be formatted in a certain way.

- Every entry is a whitespace (space, tab, carriage return, newline) separated list of numbers.
- We must specify the contents starting from address 0. This means if we only want something at address 8, we would need to have 8 entries of 0s (or anything else) before it (note we start at address 0).
- We do not have to specify the entire contents. In our example, anything after address 8 could remain unwritten.
- Numbers can be decimal (1234), hex (i.e. 0xFFFF), or binary (0b10110).
- Comments are allowed. Starting a line with // will indicate a comment, as will anything between /* ... */. The characters + and _ are ignored and can be used to format entries for easier reading.

I often make my memory contents in a text file on my own computer, then paste it into the lab.

Assignment

For this lab, we would like to use a memory with an **8-bit address and 16-bit data**. The contents should be:

Address 1 | 1111 1111 1111 1111
Address 2 | 1111 1111 0000 0000
Address 8 | 0000 1111 0000 1111
Address 9 | 0000 0000 1111 1111
Address 10 | 1111 0000 1111 1111
Address 11 | 0011 0011 0011 1111



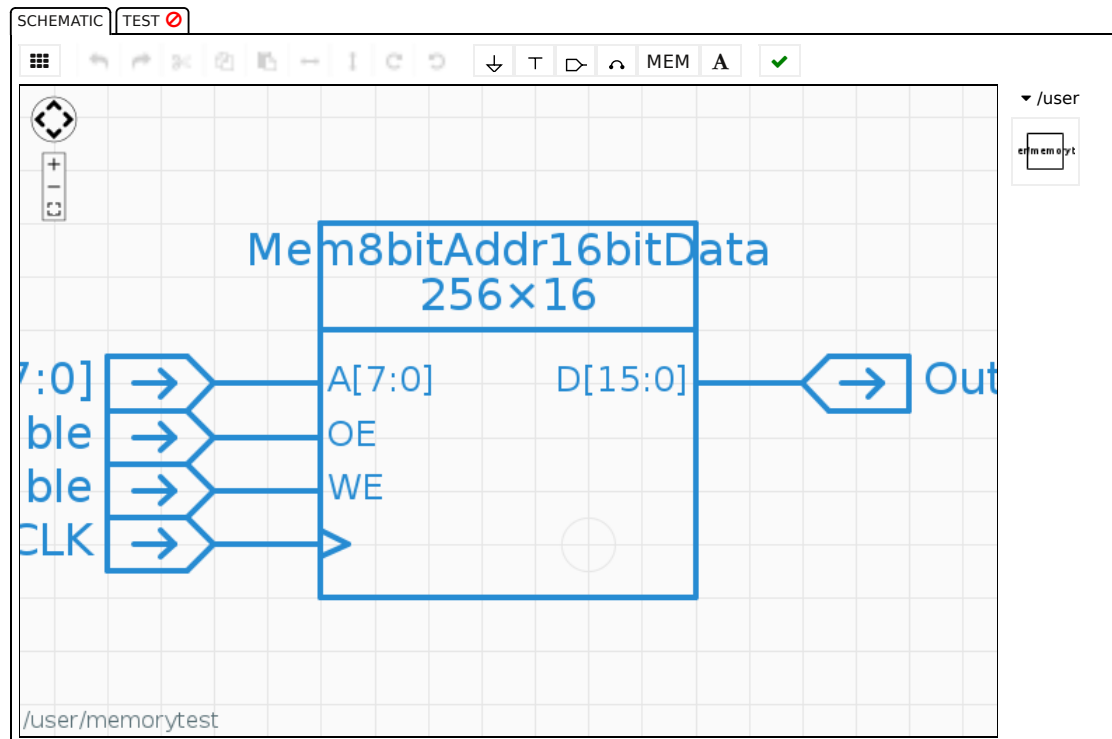

We have created a pre-formatted block of text specifying these contents below. Please copy and paste the "Memory Contents," load it into the memory, and test it. The test file will test each location for these contents.

We do NOT need to save this lab! It's just to get you familiarized.

Memory Contents:

```
/* address 0x00 */ 0x0000
/* address 0x01 */ 0xFFFF
/* address 0x02 */ 0xFF00
/* address 0x03 */ 0x0000
/* address 0x04 */ 0x0000
/* address 0x05 */ 0x0000
/* address 0x06 */ 0x0000
/* address 0x07 */ 0x0000
/* address 0x08 */ 0x0F0F
/* address 0x09 */ 0x00FF
/* address 0x0A */ 0xF0FF
/* address 0x0B */ 0x333F
```

MEMORY TEST LAB (1/1 point)

Module: /user/memorytest   

Copyright © MIT EECS 2011-2015

[Jade 2.2.40 \(2015 © MIT EECS\)](#)[Check](#)[Hide Answer](#)[Show Discussion](#)[New Post](#)

edX offers interactive online classes and MOOCs from the world's best universities. Online courses from MITx, HarvardX, BerkeleyX, UTx and many other universities. Topics include biology, business, chemistry, computer science, economics, finance, electronics, engineering, food and nutrition, history, humanities, law, literature, math, medicine, music, philosophy, physics, science, statistics and more. edX is a non-profit online initiative created by founding partners Harvard and MIT.

6 of 7

© 2015 edX Inc.

edX, Open edX, and the edX and Open edX logos are registered

04/16/2015 10:52 AM



Help

About edX

About

News

Contact

FAQ

edX Blog

Donate to edX

Jobs at edX

<https://courses.edx.org/courses/CornellX/ENGRI1210x/1...>


Follow Us


 Facebook


 Twitter


 LinkedIn

 Google+

 Tumblr

 Meetup

 Reddit

 Youtube