

## Duke ECE152 – Spring 2012 – Project Part 2: Memory

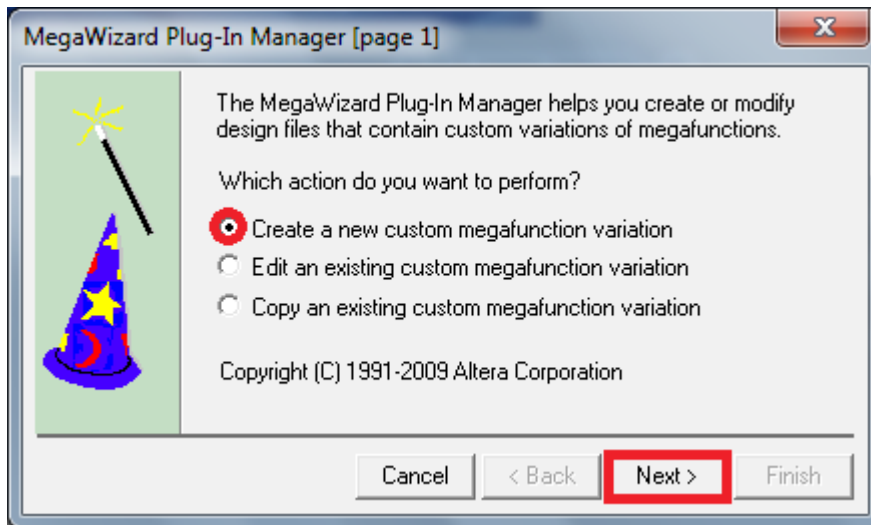
25 Points. Due electronically by 11:59pm on Friday, Jan 27.

In this part of the project, you will implement the instruction and data memories that you will use in your processor. This assignment is the *only* exception to the rule about not using Quartus megafunctions – memories must be implemented through megafunctions on the FPGA. You will use Quartus' Megafunction Wizard to create the two memories. Before proceeding, make sure that you are using Quartus II Web Edition 9.1 Service Pack 2, as the Megafunction Wizard can differ substantially from release to release.

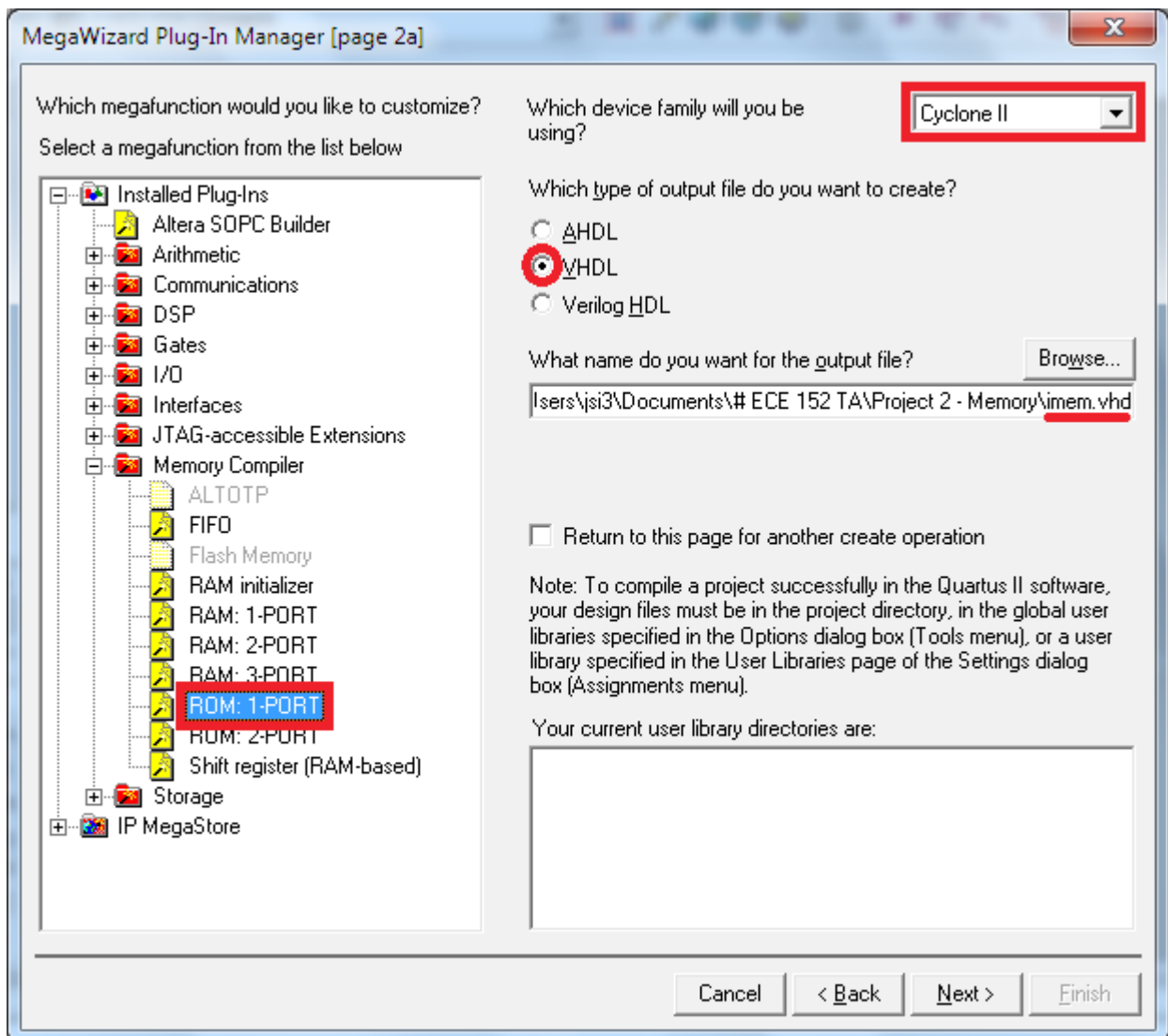
Since the architecture for your processor uses separate instruction and data memory spaces, your microarchitecture should have separate instruction and data memory blocks.

### **Project Part 2a: Instruction Memory**

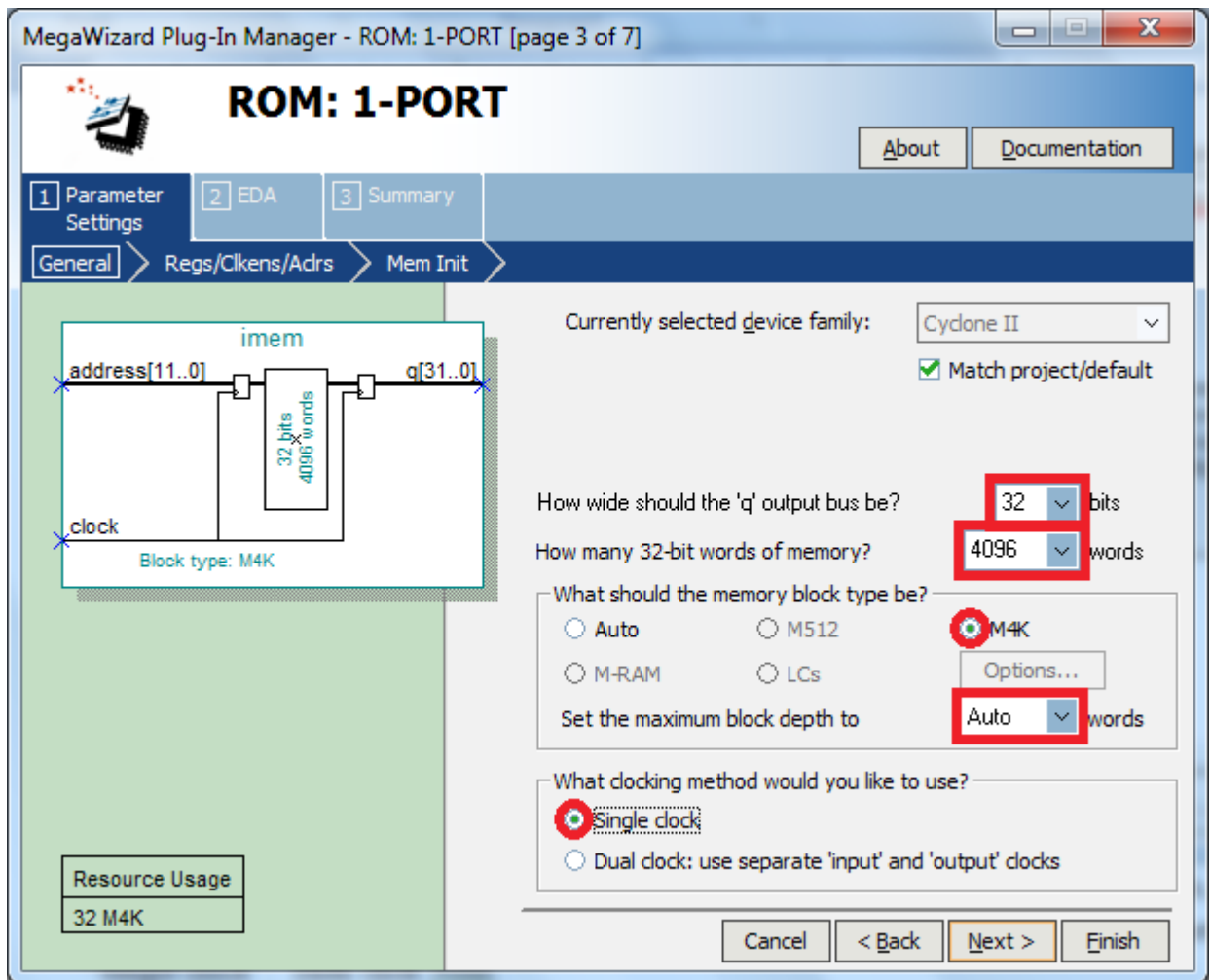
To make the instruction memory (imem.vhd), start by going to Tools □ MegaWizard Plug-In Manager. On Page 1, select “Create” and click Next.



On Page 2a, select “Cyclone II” as the device family, select “VHDL” as the output file type, and set the output file name to “imem.vhd” in your working directory. On the left side, select “Installed Plug-Ins” ▾ “Memory Compiler” ▾ “ROM: 1-PORT”. Then click Next.



On Page 3, set the output bus width to 32 bits, set the memory size to 4096 words, set the memory block type to “M4K”, set the block depth to “Auto”, and set the clocking method to “Single clock”. Then click Next.



On Page 4, uncheck the register on the output port and check the clock enable. Then click Next.

The screenshot shows the MegaWizard Plug-In Manager window for the ROM: 1-PORT component, page 4 of 7. The window title is "MegaWizard Plug-In Manager - ROM: 1-PORT [page 4 of 7]". The main title is "ROM: 1-PORT". There are buttons for "About" and "Documentation". The navigation tabs are "1 Parameter Settings", "2 EDA", and "3 Summary". The sub-tabs are "General", "Regs/Clkens/Adrs", and "Mem Init".

The block diagram on the left shows the ROM block with the following ports and signals:

- address[11..0]
- clock
- clken
- q[31..0]

The block is labeled "imem" and "32 bits 4096 words". The block type is "M4K".

The configuration options on the right are:

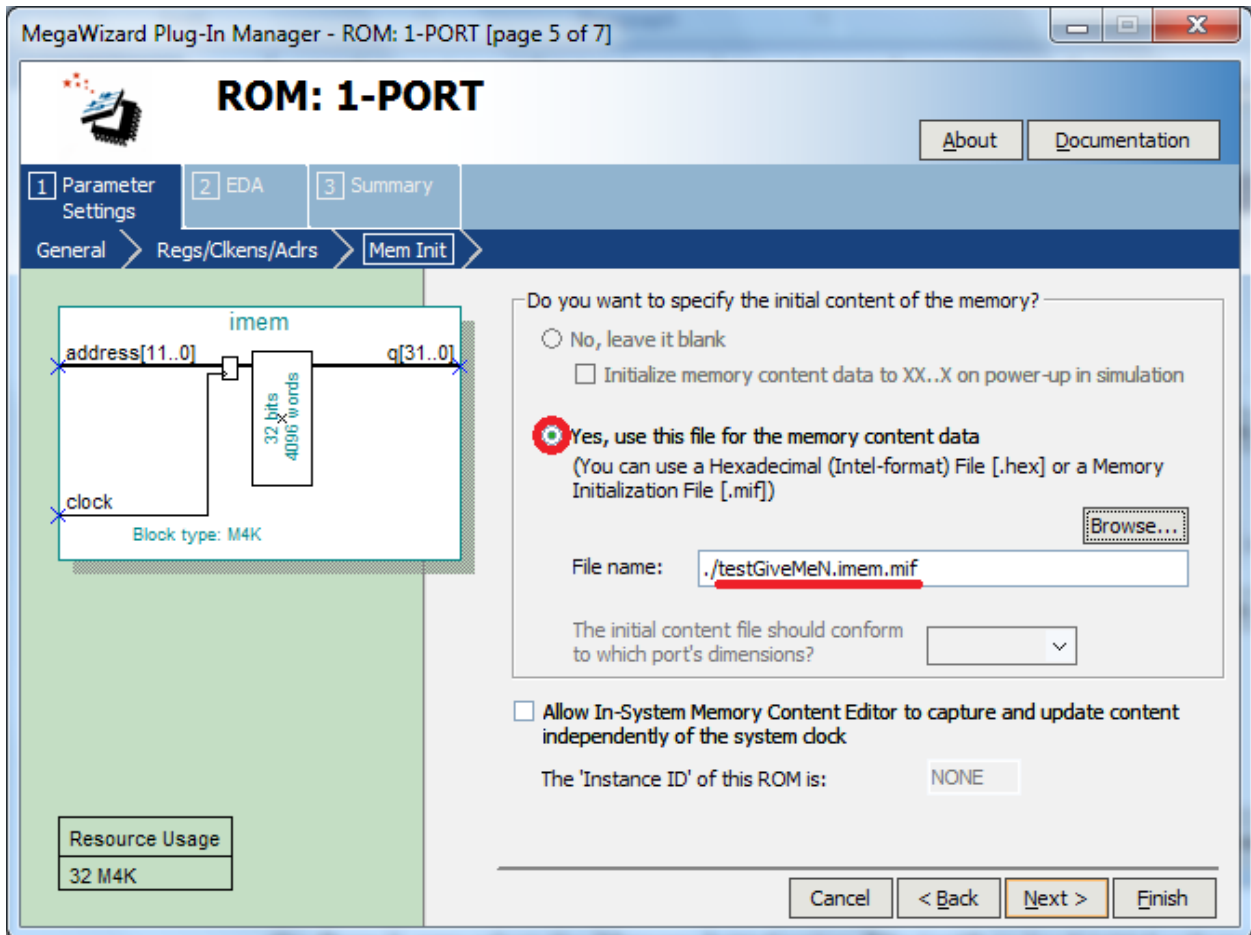
- Which ports should be registered?
  - 'data' input port
  - 'address' input port
  - 'q' output port
- Create one clock enable signal for each clock signal. All registered ports are controlled by the enable signal(s). [More Options ...](#)
- Create a byte enable port
- What is the width of a byte for byte enable?  bits
- Create an 'adr' asynchronous clear for the registered ports [More Options ...](#)

At the bottom left, there is a "Resource Usage" table:

Resource Usage
32 M4K

At the bottom right, there are buttons for "Cancel", "< Back", "Next >", and "Finish".

On Page 5, you select the Memory Initialization File (.mif) to load initial values into the memory with. A Memory Initialization File is available for you to download at <http://people.ee.duke.edu/~sorin/ece152/project/testGiveMeN.imem.mif>. Download, browse to, and select this file on your computer, then click Next.





On Page 7, uncheck everything but the “imem.vhd” file, then click Finish.

**MegaWizard Plug-In Manager - ROM: 1-PORT [page 7 of 7] -- Summary**

## ROM: 1-PORT

1 Parameter Settings | 2 EDA | 3 Summary

Turn on the files you wish to generate. A gray checkmark indicates a file that is automatically generated, and a red checkmark indicates an optional file. Click Finish to generate the selected files. The state of each checkbox is maintained in subsequent MegaWizard Plug-In Manager sessions.

The MegaWizard Plug-In Manager creates the selected files in the following directory:  
C:\Users\jsi3\Documents\# ECE 152 TA\Project 4 - Memory\

File	Description
<input checked="" type="checkbox"/> imem.vhd	Variation file
<input type="checkbox"/> imem.inc	AHDL Include file
<input type="checkbox"/> imem.cmp	VHDL component declaration file
<input type="checkbox"/> imem.bsf	Quartus II symbol file
<input type="checkbox"/> imem_inst.vhd	Instantiation template file
<input type="checkbox"/> imem_waveforms.html	Sample waveforms in summary
... imem_wave*.jpg	Sample waveform file(s)

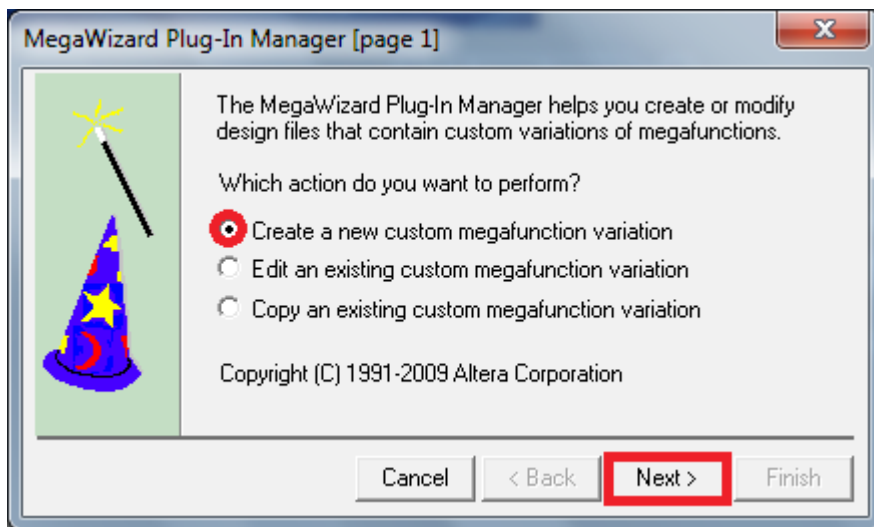
Resource Usage  
32 M4K

Cancel < Back Next > Finish

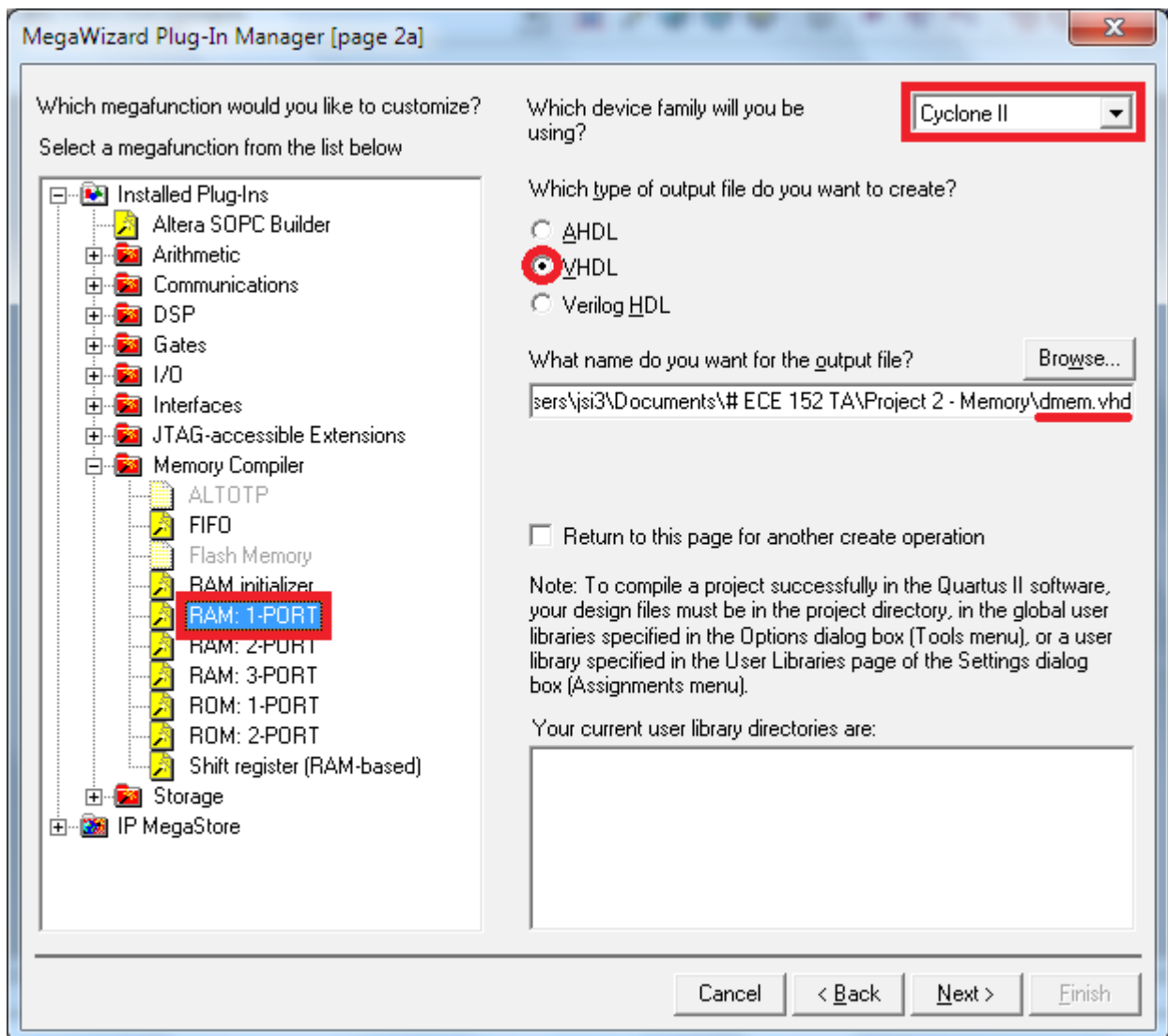
Your instruction memory is now created. Click Yes to automatically add the memory to your project, or click No to manually add just the .vhd file to your project later.

## **Project Part 2b: Data Memory**

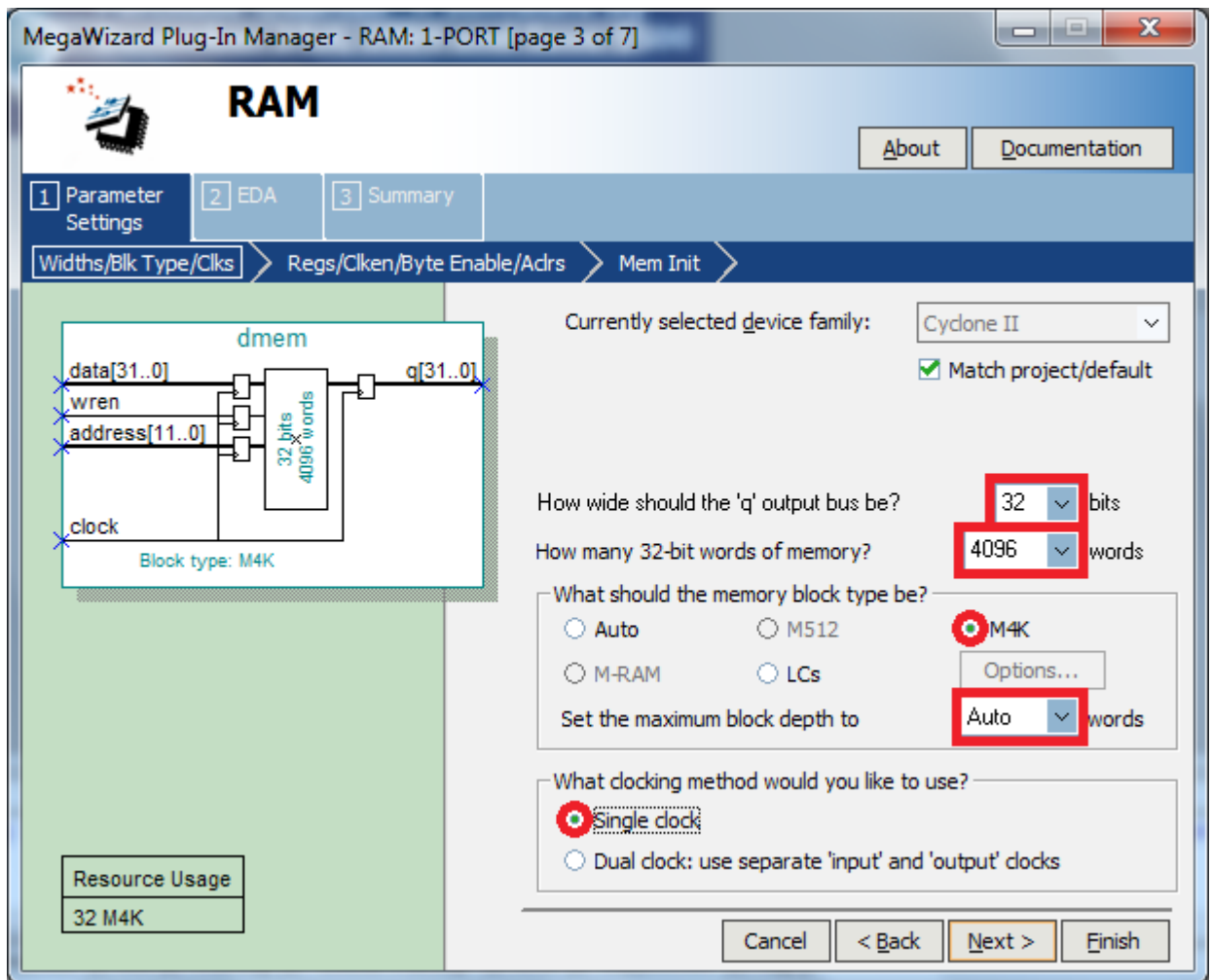
To make the data memory (dmem.vhd), start by going to Tools □ MegaWizard Plug-In Manager. On Page 1, select “Create” and click Next.



On Page 2a, select “Cyclone II” as the device family, select “VHDL” as the output file type, and set the output file name to “dmem.vhd” in your working directory. On the left side, select “Installed Plug-Ins” [ ] “Memory Compiler” [ ] “RAM: 1-PORT”. Then click Next.



On Page 3, set the output bus width to 32 bits, set the memory size to 4096 words, set the memory block type to “M4K”, set the block depth to “Auto”, and set the clocking method to “Single clock”. Then click Next.



On Page 4, uncheck the register on the output port. Then click Next.

MegaWizard Plug-In Manager - RAM: 1-PORT [page 4 of 7]

# RAM

About Documentation

1 Parameter Settings 2 EDA 3 Summary

Widths/Blk Type/Clocks > Regs/Clock/Byte Enable/Adrs > Mem Init >

dmem

data[31..0] wren address[11..0] clock q[31..0]

32 bits 4096 words

Block type: M4K

Resource Usage
32 M4K

Which ports should be registered?

- 'data' and 'wren' input
- 'address' input port
- 'q' output port

Create one clock enable signal for each clock signal. All registered ports are controlled by the enable signal(s). More Options ...

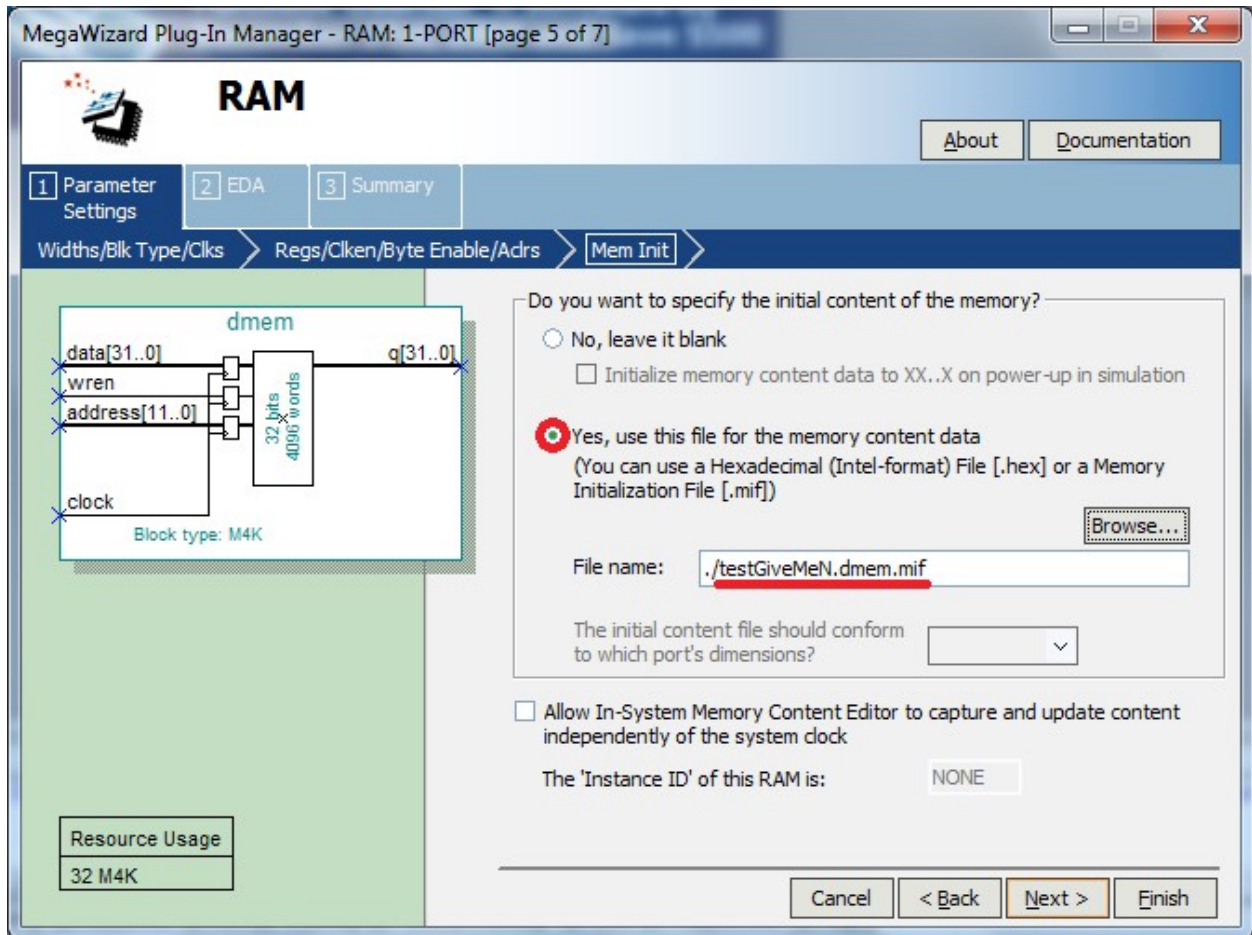
Create a byte enable port

What is the width of a byte for byte enable? 8 bits

Create an 'adr' asynchronous clear for the registered ports. More Options ...

Cancel < Back Next > Finish

On Page 5, you select the Memory Initialization File (.mif) to load initial values into the memory with. A Memory Initialization File is available for you to download at <http://people.ee.duke.edu/~sorin/ece152/project/testGiveMeN.dmem.mif>. Download, browse to, and select this file on your computer, then click Next.



On Page 6, click Next.

MegaWizard Plug-In Manager - RAM: 1-PORT [page 6 of 7] -- EDA

## RAM

About Documentation

1 Parameter Settings 2 EDA 3 Summary

Block type: M4K

File	Description
altera_mf	Altera megafunction simulation library

Timing and resource estimation

Generates a netlist for timing and resource estimation for this megafunction. If you are synthesizing your design with a third-party synthesis tool, using a timing and resource estimation netlist can allow for better design optimization.

Not all third-party synthesis tools support this feature - check with the tool vendor for complete support information.

Note: Netlist generation can be a time-intensive process. The size of the design and the speed of your system affect the time it takes for netlist generation to complete.

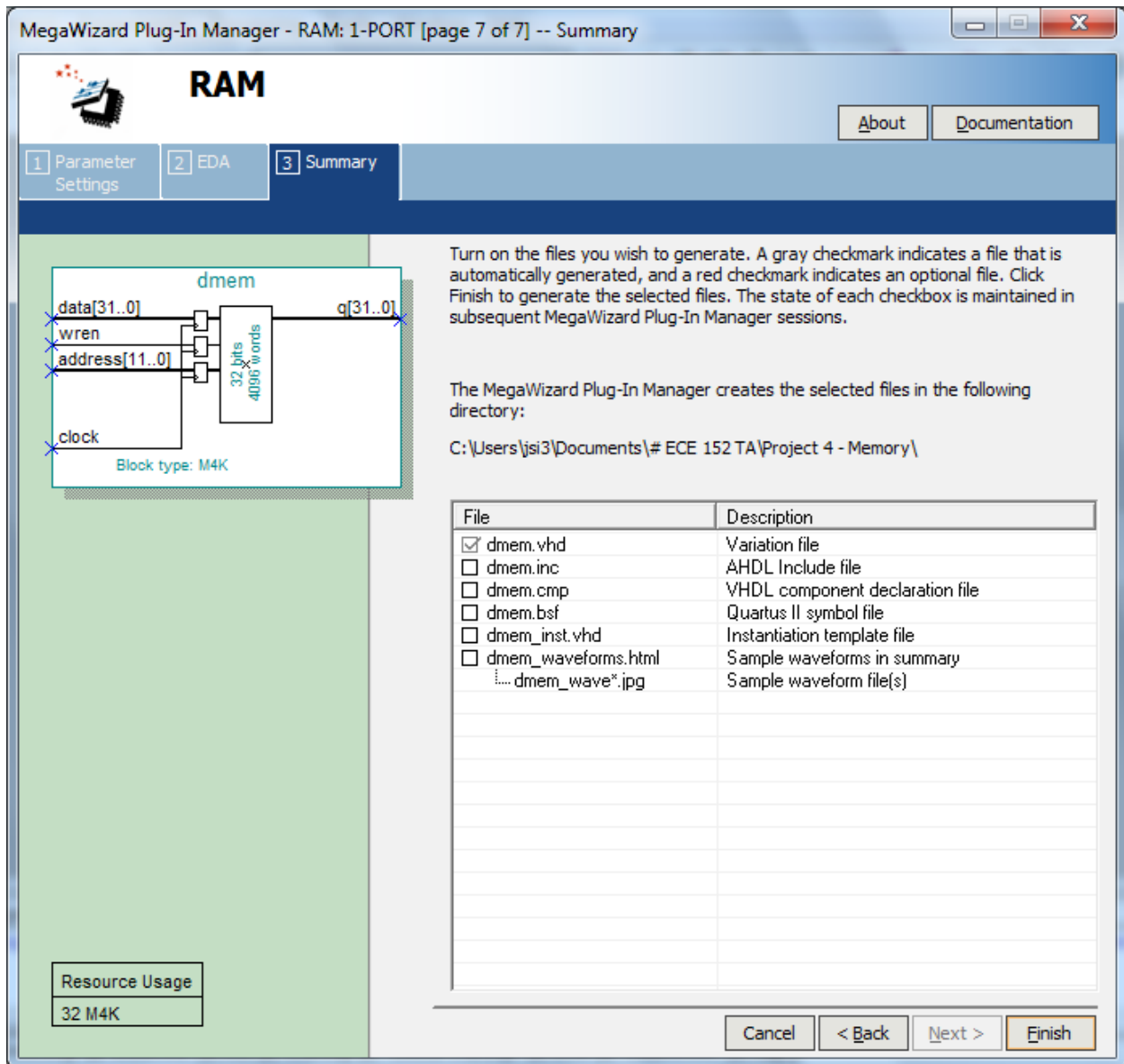
Generate netlist

Resource Usage

32 M4K
--------

Cancel < Back Next > Finish

On Page 7, uncheck everything but the “dmem.vhd” file, then click Finish.



Your data memory is now created. Click Yes to automatically add the memory to your project, or click No to manually add just the .vhd file to your project later.

## **Project Part 2c: Initializing the Memories**

To change the Memory Initialization File for your memories, open the imem.vhd or dmem.vhd file and go to line 89 or 94, respectively, and rename the “init\_file” path to the new file.

```
81
82 BEGIN
83     q    <= sub_wire0(31 DOWNT0 0);
84
85     altsyncram_component : altsyncram
86     ■    GENERIC MAP (
87         clock_enable_input_a => "BYPASS",
88         clock_enable_output_a => "BYPASS",
89         init_file => "testGiveMeN.imem.mif",
90         intended_device_family => "Cyclone II",
91         lpm_hint => "ENABLE_RUNTIME_MOD=NO",
92         lpm_type => "altsyncram",
93         numwords_a => 4096,
94         operation_mode => "ROM",
95         outdata_aclr_a => "NONE",
96         outdata_reg_a => "UNREGISTERED",
97         ram_block_type => "M4K",
98         widthad_a => 12,
99         width_a => 32,
100        width_byteena_a => 1
101    )
102    ■    PORT MAP (
103        clock0 => clock,
104        address_a => address,
105        q_a => sub_wire0
106    );
107
```

After implementing your memories, you should test them thoroughly to verify that they work correctly. One test waveform is provided for your instruction memory at [http://people.ee.duke.edu/~sorin/ece152/project/test\\_imem.vwf](http://people.ee.duke.edu/~sorin/ece152/project/test_imem.vwf) and for your data memory at [http://people.ee.duke.edu/~sorin/ece152/project/test\\_dmem.vwf](http://people.ee.duke.edu/~sorin/ece152/project/test_dmem.vwf). In addition, this assignment will be graded by running additional tests that are not provided, so do not assume that you can ignore bugs that do not manifest themselves on the one test that is provided.

## **Submitting This Assignment**

To submit this assignment, create a Quartus Archive (Project □ Archive Project) named project2.qar of all the files needed to implement your design. Make sure that your top-level files are named imem.vhd and dmem.vhd. Email your Quartus Archive file as an attachment along with all group members’ names and NetIDs to [duke.ece152.spring2012@gmail.com](mailto:duke.ece152.spring2012@gmail.com).