

Mentor Graphics Tutorial 3

Design Architect and Quicksim II

Created by: Dr. Khurram Waheed, San Diego State University
Last modified: Spring 2004

1 Introduction

The purpose of this tutorial is to get you familiar with Design Architect and Quicksim II in the Mentor Graphics package for the projects in CompE 475. We will be using the Smart Model Library to obtain frequently used components such as decoders and flip flops. In this tutorial, we will attempt to build an adder system with input/output port interface using 74HC574 Octal D-Flip flops, one 74HC541 Octal Buffer, two 74HC283 four-bit adders and the decoding logic using 74HC139 decoders.

2 To start Mentor Graphics

2.1 Running Mentor Graphics?

Good configuration management strategy is to always use a separate directory for each of your mentor graphics design/project. Set the *MGC_WD* variable to your appropriate work directory.

3 Design Architect

3.1 Opening a new sheet

To start Design Architect type **da** at the Unix command prompt.

NOTE: if you use the **&** behind the command you will not lose control of your command prompt and unix (xterm or terminal) window.

At the session palette on the right hand side of the screen, click on **Open Sheet**. Enter your path and filename for your mentor file. I would recommend creating a new sheet right in your MGC directory. To do this just type the name of your new sheet right after the given path name. For example, to create a new sheet named tutorial, in the **Component Name** field append to the path: **\$MGC_WD/tutorial3**

If open sheet fails, check and make sure that the path exists. Use **Navigator** to select an old schematic.

3.2 Obtaining library models

In this course, your assignments will require you to utilize actual models of IC components, i.e. the 74HC139 decoder. In Mentor Graphics, you can find the models by going to the **Libraries** option under the menu bar on top of the screen. Then click on **Logic Modeling Smart Library**. Depending on your component and the manufacturer, you need to choose the appropriate option in the palette. For example, if I am looking for a 32KByte ROM made by Texas Instrument, then I would choose **Memories->ROM->TI->27C256**.

4 Example

For this tutorial, we will build an interface and an adder subsystem using two 74HC139 decoders, two 74HC574 octal D-flip flops, one 74HC541 buffer, two 74HC283 four-bit adders and four input ports.

4.1 Getting ports

1. To get the ports
 - (a) Go to the **Schematic Palette**
 - (b) Click on **Library**
 - (c) Choose **gen lib**
 - (d) Find **portin**
2. Similar to the input port, the output port can be found by choosing **portout** under gen lib.

4.2 Getting 139, 541, 574 and 283 from the SmartModel Library

1. The 74HC139 can be found using the procedure described above about the Smart Model Library:

- (a) Under the SmartModel Library Palette, choose General Logic Model.
 - (b) Click on **General Purpose Logic** on the Palette.
 - (c) 74HC139 is a 2-4 decoder, therefore click on **Decoder**.
 - (d) We will use Harris's ICs in this example, select **Harris** as the manufacturer.
 - (e) You should be able to choose CD74HC139 at this point by selecting **139 Dual 2 to 4 Line Decoder/Demultiplexer** and then selecting **CD74HC139**.
2. The 74HC541, which is an octal buffer, can be found similar to the 139 decoder. See if you can find it using the SmartModel Library. The sequence of selection for the 74HC574 is as follows: **Smart Model Libraries->Gen Purpose Logic->Gates->Harris->541 Octal Buffer Line Driver, 3 State->CD74HC541**
 3. The 74HC574 is a octal D-flip flop. The sequence of selection for the 74HC574 is as follows: **Smart Model Libraries->Gen Purpose Logic->Flip Flop->Harris->574 Octal D-Type Flip Flop, 3 State->CD74HC574 or CD74HC374**
 4. The 74HC283 is a four-bit adder. The sequence of selection for the 74HC574 is as follows: **Smart Model Libraries->Gen Purpose Logic->Arithmetic->Harris->283 4-bit Full Adder, Fast Carry->CD74HC283**

4.3 Basic Layout

1. Now that you have obtained all the necessary components for this example. Put them in an orderly fashion. Clicking the right mouse button on the sheet lets you select, unselect, edit, move, delete or add wires between the components.
2. Next, name your ports such that your input ports are called address(1:0), RD, WR and data(7:0). address(n:0) means a “n+1”-bit wide bus. You can name the ports by placing your mouse over the text area next to the port, then press Shift-F7 to change the text.

4.4 Connection using wires and naming nets

1. The next step is make connections between the components using wires, or naming nets. For all your Mentor Graphics assignments I would recommend **NOT** using buses. Instead, use **Name Nets**. As you can see in the figure below all three different methods of connecting components are being used (actually drawing the complete wire, ripping a bus, and naming nets). **Remember do not use buses in your designs!**
2. To **Name Nets**, connect a short wire off the component pin (you will have to do this for each pin on each device). NOTE: components are blue, and wires are yellow. After you have drawn all your wires highlight (dashed white) all you wish to name, right click the mouse and go to **Name Nets**:
3. A text box will appear at the bottom of the screen. NOTE: if you highlighted more than one wire then a white cross will indicate which line your are currently naming. Type the line name in the **Property Value** box, leaving all other fields to defaults. Press ENTER. Move the mouse so you can see the linked Property Value to the wire. Locate on the design where you want the Property Value and press the left mouse button.
4. If you highlighted more than one wire to name, the computer will continue to prompt you for successive values. This is an easy way to name a lot of lines in a short period of time.
5. At this point, your design should look something similar to the schematic shown in figure 1 below.
6. MENTOR NAMING CONVENTION: you will notice in the figure below that there are 2 different naming conventions. For example, there is **address(0)**, and **RD2**. When you use a name with a () parentheses pair you are telling Mentor that this line is part of a bus of wires (notice the address(1:0) input port). When you do not use parentheses you are defining an independent line. If you name all your nets you do not need to use Input and Output ports (unless you want to generate a symbol).

4.5 Checking the Final Design

1. For checking the design, select **sheet** and **With Default** from the **Check** menu.
2. Make sure there are no error messages. Although, when you **Name Nets** warnings are generated by Mentor Graphics. Do not be concerned about this.
3. Save your sheet. (see the schematic on the next page)

4.6 Saving the Final Design

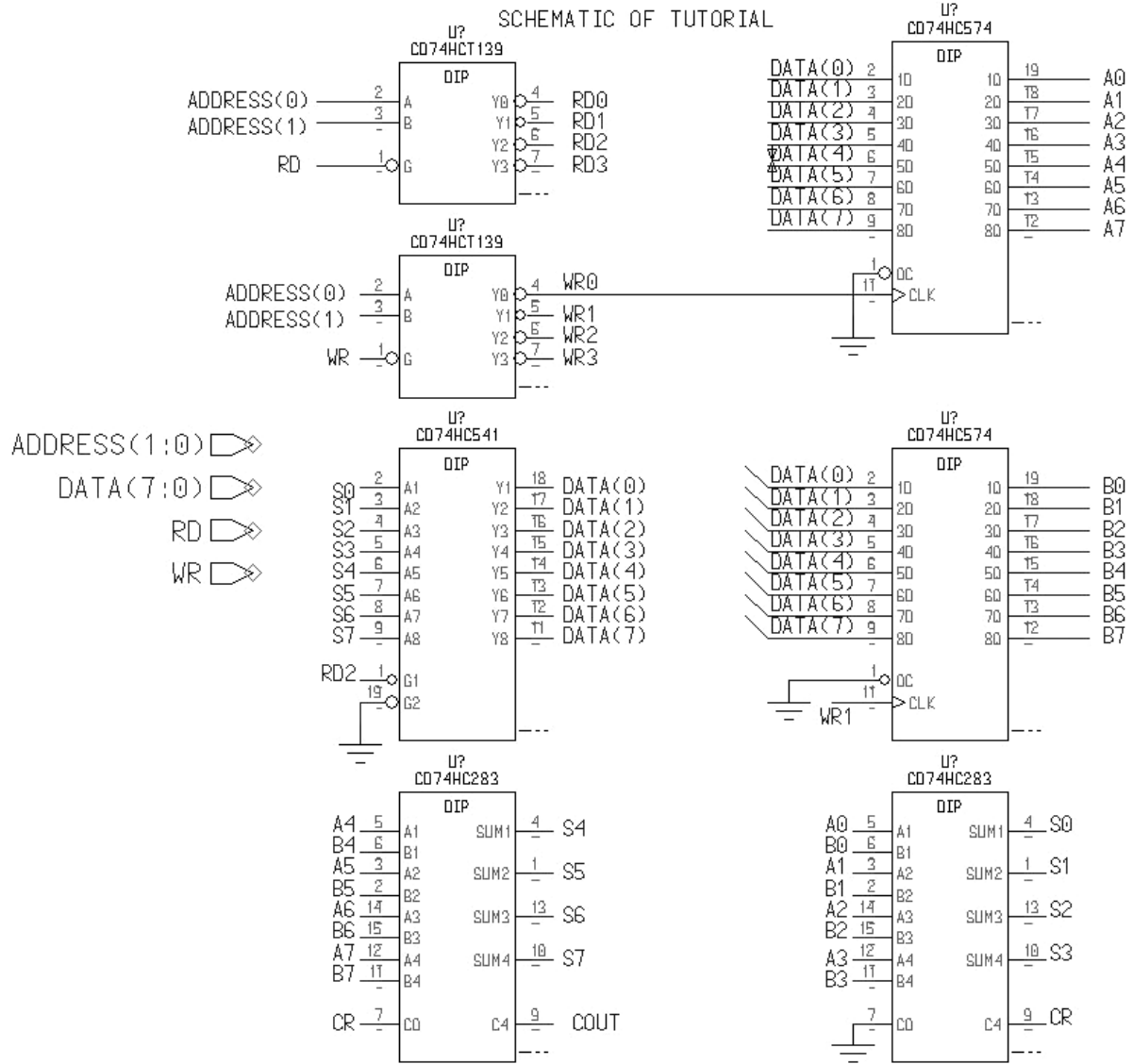
Save the design by selecting **Save** from the **File** menu.

5 QuickSim II

5.1 Starting Quicksim II

1. Make sure that you are in your MGC directory. At the Unix command prompt type:
quicksim <filename> If you did not use the correct sheet name, or there is a problem with your design, Quicksim will not open.

NOTE: if you use **&** at the end of the command you will not loose control of your command window.



** Notice that the top most 574 has a different technique for ripping the bus. This tutorial does not explain how to do this because we do not want to build your design by ripping buses. Instead you should “Name Nets” as is show with the other three 574s; refer to section 4.4 on how to properly Name Nets.*

5.2 Writing a Force File

Use any Unix editor to write the given example force file and save it in mgc directory. I would recommend saving your force files in the folder that Mentor Graphics generates for you when create a new sheet. Here is what your force file is going to look like when you are simulating your 188 design (see below for this tutorial example’s force file):

```

$$reset_state();
$$delete_forces(@all, void);

delete traces RESET CLK CLKOUT ALE WR RD UCS LCS S address data
add traces RESET CLK CLKOUT ALE WR RD UCS LCS S address data
    
```

```
force RESET 0 0 -fixed
force RESET 1 3000 -fixed

set clock period 100
force CLK 0 0 -repeat
force CLK 1 50 -repeat

run 200000
```

For the example in Figure 1, you could use the following force file:

```
// This is the force file for example design
// Specifying reset_state and delete_forces here means that you do not
// have to do these manually in quicksim.

$$reset_state();
$$delete_forces(@all, void);

// deleting traces and lists before adding them avoids having
// additional
// sets of redundant traces displayed. Otherwise, each time you use the
// force file you would get another set of the traces.
// When specifying the trace for a bus you can just use the name part.
// For example, data(7:0) is here just called data.

delete traces ADDRESS DATA RD WR
add traces ADDRESS DATA RD WR
delete lists ADDRESS DATA RD WR
add lists ADDRESS DATA RD WR

// the syntax for adding a force to a port is
// force portname value starttime <stoptime> <options>.
// For example the following force line sets the
// WR port to a logic 1 at time 0 and leaves it
// that way until it is changed by another force command.

force WR 1 0 -fixed ; // here "-fixed" means the strongest
force RD 1 0 -fixed ; // possible signal value.

// a comment can come after a semicolon

force ADDRESS 00 0 -fixed
force DATA 3C 0 -fixed
force WR 0 100 -fixed
force WR 1 200 -fixed

force ADDRESS 01 300 -fixed
force DATA A5 300 -fixed
force WR 0 400 -fixed
force WR 1 500 -fixed

force ADDRESS 02 600 -fixed
force DATA XXz 600
force RD 0 700 -fixed
force RD 1 800 -fixed
```

```
force ADDRESS 00 900 -fixed
force DATA AA 900 -fixed
force WR 0 1000 -fixed
force WR 1 1100 -fixed

force ADDRESS 01 1200 -fixed
force DATA 55 1200 -fixed
force WR 0 1300 -fixed
force WR 1 1400 -fixed

force ADDRESS 02 1500 -fixed
force DATA XXz 1500
force RD 0 1600 -fixed
force RD 1 1700 -fixed

// You can specify the run time right in the force file.

run 1800

// If you want to specify a periodic signal, you can do the following:
//   set clock period 100
//   force AB  0 0 -repeat
//   force AB  1 50 -repeat
// The above set of lines would create a clock signal on port AB with a
// period of 100 ns, and the signal is zero from 0 to 50 ns, and is one
// from 50
// to 100 ns.
//
```

5.3 Running the simulation – Adding Forces

1. Select **Add**, then **Force** and then **From File** from the pop up menu using the right most button of the mouse.
2. Navigate to your force file.
3. You can also add single forces from the pop up menu **Force** by selecting **Single Value** or multiple forces from **Multiple Values**.

5.4 Running the simulation – Step-by-step

- If you are adding forces manually, then select from the pop up menu **Run -> Simulation** and then choose **For Time, Until Time and Stop Time**. The “For Time” and “Until Time” are very powerful because you can run a simulation step-by-step to see exactly what is happening. In fact, you can run a force file for a certain amount of time (say 20000) then run “For Time” 1000 to have a total simulation time of 21000. The simulation will pick up right after 20000 if you use the “For Time” or “Until Time” feature. NOTE: I would stay away from the “Stop Time” feature for stability reasons, especially if you are initially starting with a force file.
- For both force file examples above, the run command has been included (found near the bottom).

6 Printing

- For any printing in Design Architect or Quicksim II, select **MGC**, then **Setup** and then **Printer** and select **lp** and press **Select Printer** and then **OK**. Now print from the menu.
- The setting or printer selection might change. So, please consult CAE if you have problems with printing.
- You can print to a file by changing your printer setup to **mgc_bw_ps_file**. The output will be saved in your locate **tmp** directory. To get to this directory simply type: `cd /tmp` To get back to your home directory type: `cd`

7 Appendix

7.1 Design Architect Strokes

You can use short-cut strokes while making your design.

- To obtain the short-cut strokes help screen hold the middle mouse button and draw a question mark on the schematic.
- Examine the strokes carefully, they will save you time in the future.

7.2 Naming bits on ripped lines

To name bits on ripped lines, do the following:

1. Click and hold on the right mouse button, you should see the option **Other Menus**, select that option, continue holding on the button.
2. Next click on **Property/Text Menu**.
3. Choose the option **Sequence Text**.
4. After you have chosen your desired options, click **OK**.
5. Next, point your mouse over the index of the bit line where you want the sequence to start, and click the left button.
6. Once the left button is clicked and the index has changed, MG automatically increments the index. Thus, point your mouse to the next desired bit and click the left button as desired.

7.3 Adding Comments

You can add comments by selecting **Text** from the pop up menu and then selecting **Add Comment**.

7.4 Generating a symbol

To create a symbol, do the following:

1. Go under **Miscellaneous** and click on **Generate Symbol**.
2. Look through your options, then click **OK**.
3. Check your sheet and save symbol.

(Acknowledgements: Dr. Michael G. Morrow, University of Wisconsin Madison)

PROJECT 3

Create schematics and properly simulate a digital circuit which does the following:

- Your inputs comprise of an 8 bit **AD** (Address Data) bus, and three control signals **ALE**, **DEN** and **DT/R** as discussed in Chapter 9 of text.
- Using the 74xx374 or 74xx574 Latches and 74xx245 buffers show the following events
 - Latching of an Address on the 8-bit **Addr** bus
 - Latching of a data byte on the 8-bit output data bus labeled **Out_data**
 - Input from 8-bit input data bus called **In-data**

(Note: xx in the device names accounts for the logic family and may be equal to LS, ALS or HC in the chosen devices)

Grading

This tutorial and the project comprise of 50 points, i.e.,

Tutorial	25 points
Exercise	25 points

Report Format

Submit the completed tutorial and the exercise in the following format

1. Title Page
2. Introduction
3. Tutorial
 - Design Description
 - Schematics
 - Force File
 - Simulation Results
 - Comments
4. Exercise
 - Design Description
 - Schematics
 - Force File
 - Simulation Results (with minimum delays & maximum delays)
 - Measured Latch-up and buffer-delay times (for above simulations)
 - Comments & Observations
5. Conclusions