5 Microprocessor Design

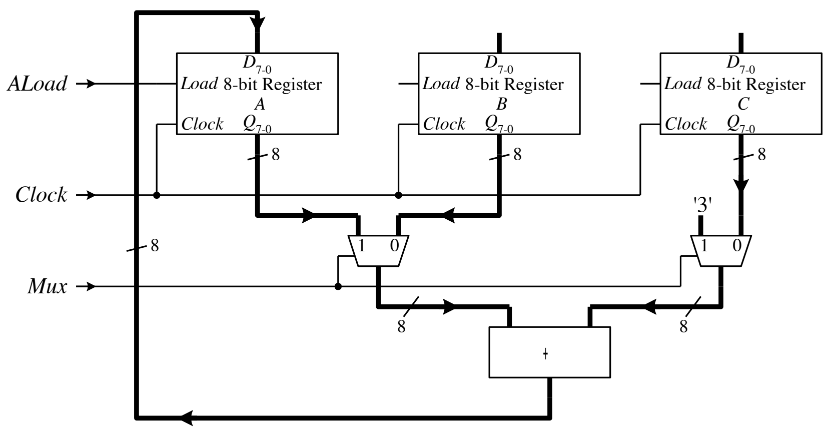
In building the final microprocessor, we simply have to combine the control unit together with the datapath. This involves the connection of all of the control signals and the status signals together between the two units. All of the clock signals from both the datapath and the control unit are connected together to the master clock signal, and all of the reset signals are connected together to a master reset switch.

5.1 Examples of Microprocessor Design

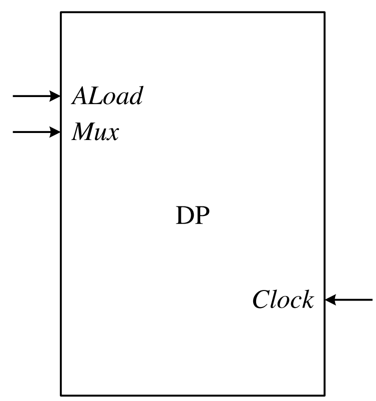
We will now illustrate the design of microprocessors with two examples.

5.1.1 Example 5: Microprocessor for the two-statement problem

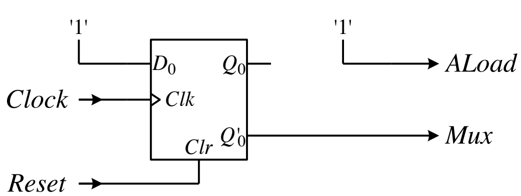
For the two-statement problem from Section 3.2, we have derived the datapath shown in Figure 16 and repeated here in Figure 35(a) for convenience. To simplify the microprocessor schematic drawing, we will represent the datapath circuit with the logic symbol shown in Figure 35(b). The logic symbol simply consists of all of the input and output signals for this circuit.



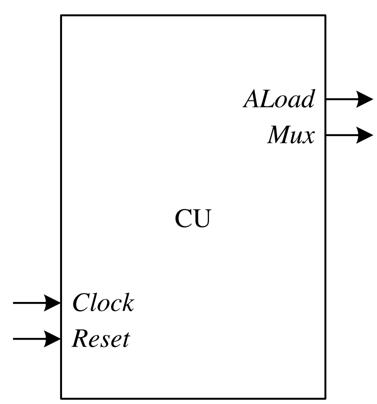
(a)



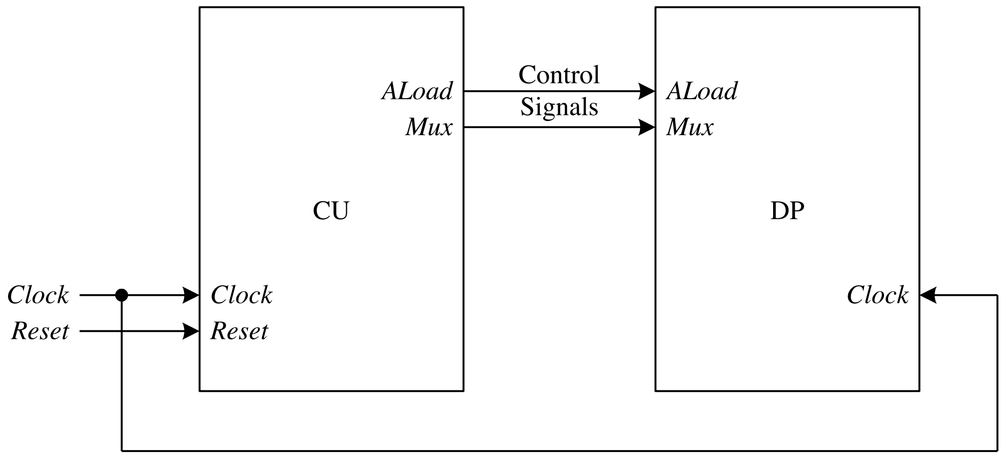
(b)



(c)



(d)



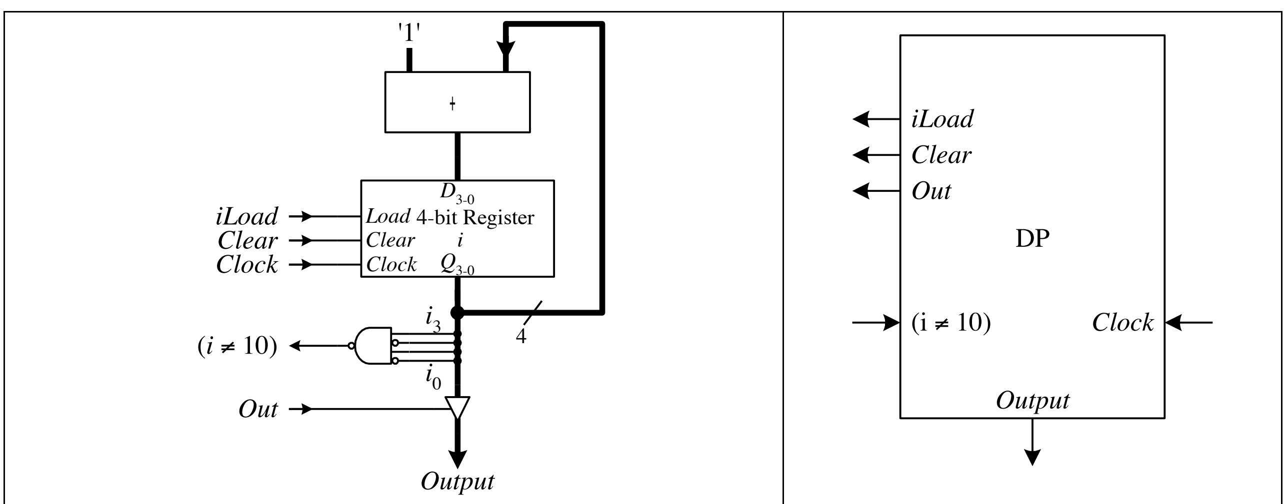
(e)

Figure 35: Construction of the microprocessor for the two-statement problem: (a) datapath; (b) symbol for the datapath; (c) control unit; (d) symbol for the control unit; (e) microprocessor circuit.

Similarly, the control unit for this datapath derived in Section 4 and repeated here in Figure 35(c) is represented by the logic symbol shown in Figure 35(d). The final microprocessor circuit is made by connecting these two components together using their corresponding signals as shown in Figure 35(e). In this example, there are no external data signals.

5.1.2 Example 6: Microprocessor for the counting from 1 to 10 problem

For the counting from 1 to 10 problem, we have derived the datapath shown in Figure 28(a) and the control unit shown in Figure 34(f). These two components are repeated here in Figure 36 (a) and (c) for convenience, and their respective logic symbol are shown in Figure 36 (b) and (d). The final microprocessor circuit is made by connecting these two components together using their corresponding signals as shown in Figure 36(e).



(a) (b)

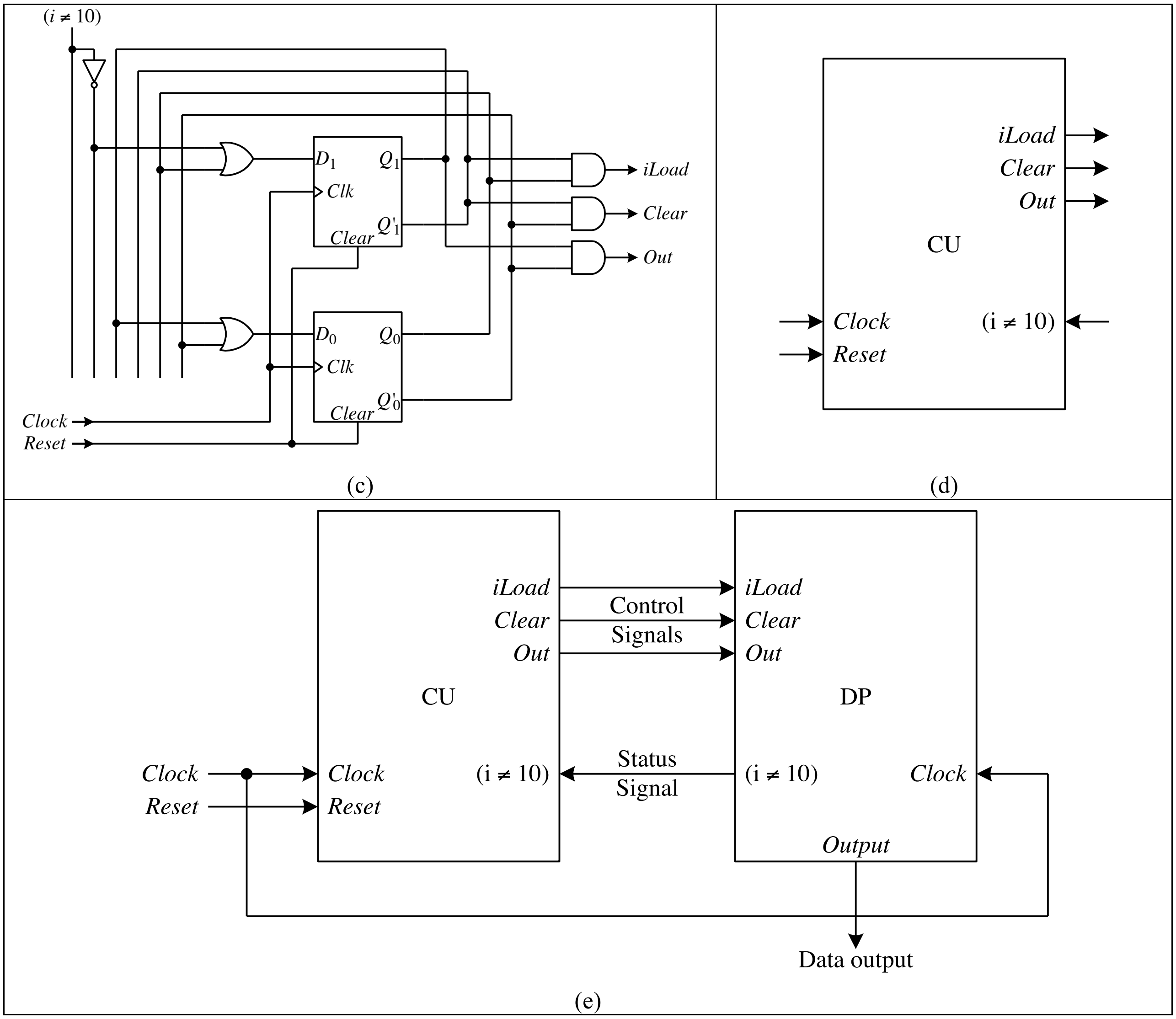


Figure 36: Construction of the microprocessor for the counting from 1 to 10 problem: (a) datapath; (b) symbol for the datapath; (c) control unit; (d) symbol for the control unit; (e) complete microprocessor circuit.

6 Labs

The following labs will teach you how to design and implement microprocessor circuits. Each lab will first show how the datapath is designed, followed by the control unit, and finally the complete microprocessor. You will then implement the microprocessor using the Quartus development software and then load it onto the Microprocessor Design Trainer to test and verify its operation.

The complete microprocessor project for all of the labs are available on the DVD under the folder `\Circuits`. You can open up these projects and directly load it onto the trainer board to see its operation without having to create them yourself. However, it is highly recommended that you do create them yourself, and only refer to these complete projects if you have difficulties.

6.1 Lab 1: Quartus Development Software

Purpose

In this lab you will learn how to use the Altera Quartus II development software to design a simple combinational circuit.

Introduction

The Altera Quartus II development software and the Microprocessor Design Trainer development board provide all of the necessary tools for implementing and trying out all of the circuits, including building the final general-purpose microprocessor, discussed in this courseware. The Quartus II software offers a completely integrated development tool and easy-to-use graphical-user interface for the design, and synthesis of digital logic circuits. Together with the Microprocessor Design Trainer development board, these circuits can actually be implemented in hardware. The main component on the Microprocessor Design Trainer development board is a field programmable gate array (FPGA) chip which is capable of implementing very complex digital logic circuits. After synthesizing a circuit and downloading it onto the FPGA, you can see the operation of the circuit in hardware.

The Web Edition version of the Quartus II software is included on the accompanying DVD. This lab assumes that you are familiar with the Windows environment, and that the Quartus II software has already been installed on your computer. If you have not done so, go back and follow the instructions in Sections 1.3 and 1.4. The rest of this lab will provide a step-by-step instruction for the schematic entry of an 8-bit 2-to-1 multiplexer circuit.

6.1.1 Starting Quartus II

After the successful installation of the Quartus II software, there should be a link to the program under the Windows' ***Start*** button named ***Quartus II 10.0 Web Edition***. Click on this link to start the program. You should see the main Quartus II window similar to Figure 37.

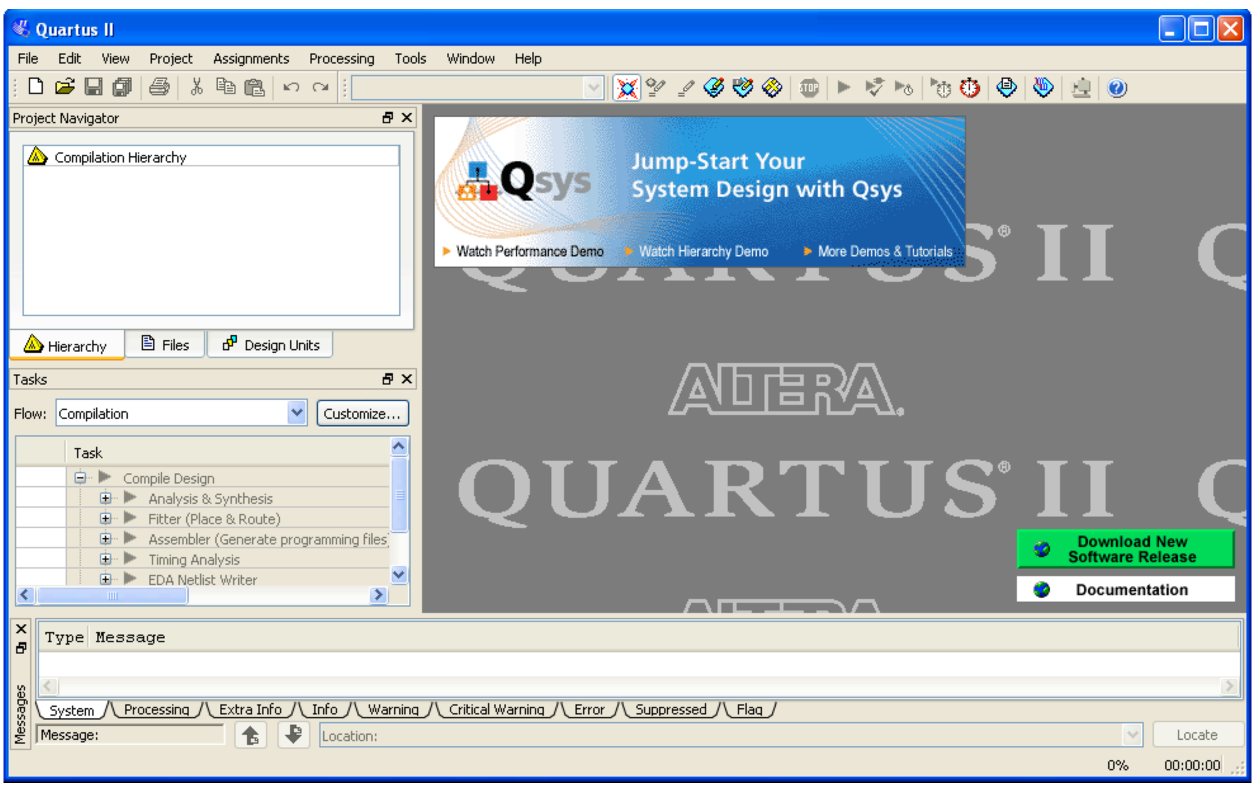


Figure 37: The Quartus II main window.

6.1.2 Creating a New Project

Each circuit design in Quartus II is called a project. Each project should be placed in its own folder, since the program creates many associated working files for a project. Perform the following steps to create a new project and a new folder for storing the project files.

From the Quartus II menu, select ***File > New Project Wizard***. If the ***New Project Wizard Introduction*** screen appears and you don't want to see it again the next time you start the new project wizard, you can select the check box that says ***Don't show me this introduction again***, and then click ***Next*** to go to the next screen. You should see the ***New Project Wizard: Directory, Name, Top-Level Entity [page 1 of 5]*** window as shown in Figure 38.

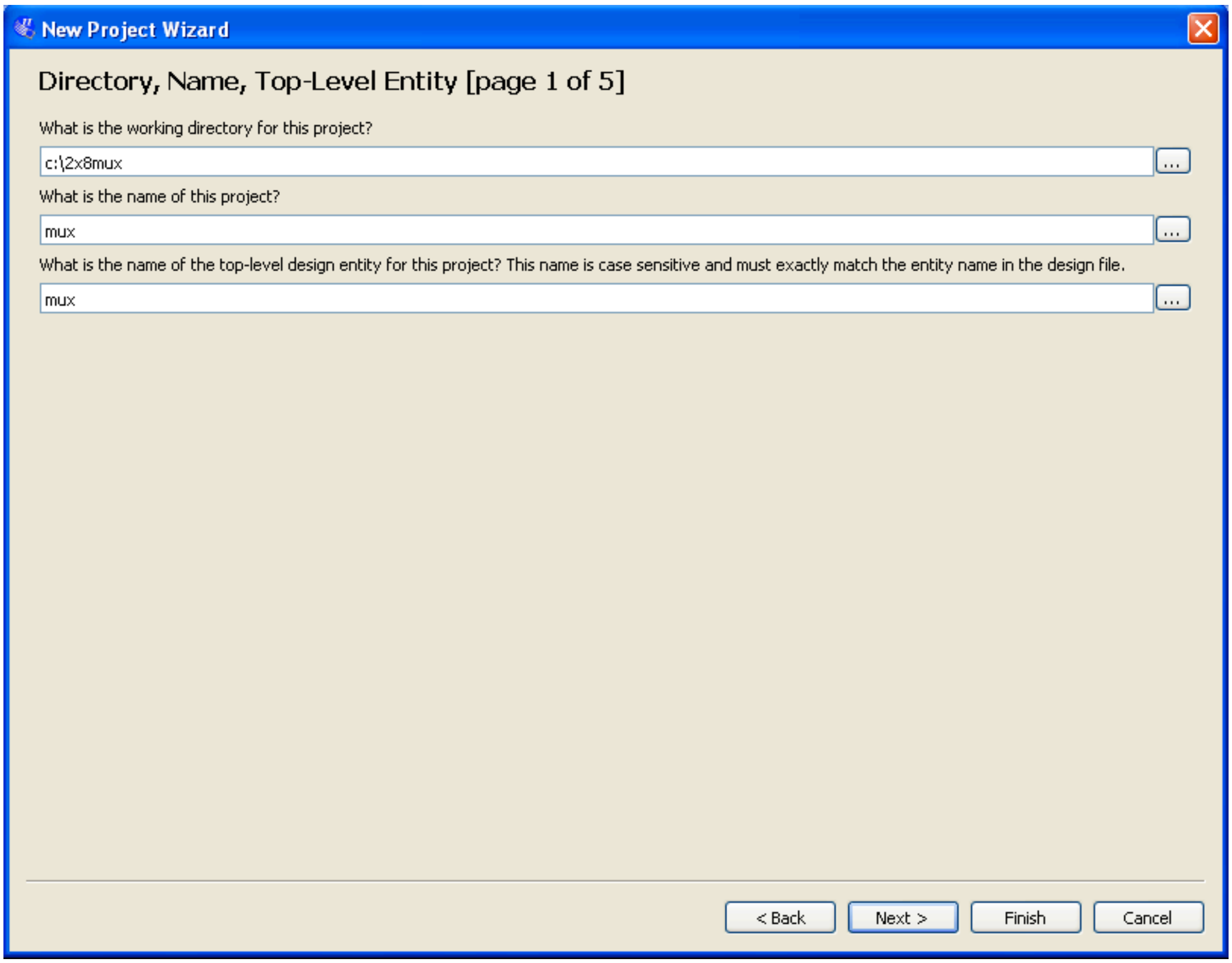


Figure 38: The New Project Wizard: Directory, Name, Top-Level Entity window with the working directory, the project name, and the top-level entity name filled in.

Type in the directory for storing your project. You can also click on the  icon next to it to browse to the directory.

- For this lab, type in `c:\2x8mux` to create a folder named `2x8mux` in the root directory of the C drive.

You also need to give the project a name.

- For this lab, type in the project name `mux`.

A project may have more than one design file. Whether your project has one or more files, you need to specify which design file is the top-level design entity. The default name given is the same as the project name. However, you can use a different name.

- For this lab, leave the top-level file name as `mux`, and click ***Next*** to continue to the next window.

Since the directory `c:\2x8mux` does not yet exist, Quartus II will inform you of that and asks whether you want to create this new directory. Click ***Yes*** to create the directory.

In the ***New Project Wizard: Add Files [page 2 of 5]*** window, you can optionally add existing circuit source files associated with your project. For example, if you have a source file created in another project and want to use it in this project, you can specify that here.

- We are starting a new project and do not yet have any source files so click ***Next*** to continue to the next window.

In the ***New Project Wizard: Family & Device Settings [page 3 of 5]*** window as shown in Figure 39, we select the target FPGA device that we will be using to implement the circuit on. The Microprocessor Design Trainer development board uses the EP3C16F256C8 FPGA chip.

- In the ***Device Family*** drop-down box, select ***Cyclone III***.
- In the ***Available devices*** list, select the device ***EP3C16F256C8***. If this device is not listed, then you need to reinstall the Quartus II program with the Cyclone III device family option checked.
- Click ***Next*** to continue to the next window.