Contents

Appendix C UP2 Programming—Tutorial 3 2
C.1 Getting Started 2
C.1.1 Preparing a Folder for the Project 2
C.1.2 Creating a Project 3
C.1.3 Adding Files to the Project 4
C.1.4 Selecting the Target Device 5
C.1.5 Completing Project Creation 5
C.1.6 Viewing the Design Source Files 6
C.2 Analysis and Synthesis 6
C.3 Circuit Simulation 6
C.4 Mapping the I/O Signals 7
C.5 Fitting the Netlist and Pins to the PLD 8
C.6 Hardware Setup 9
C.6.1 Installing the ByteBlaster Driver 9
C.6.2 Selecting the Hardware Device 9
C.6.3 JTAG Jumper Settings 11
C.6.4 Hardware Connections 12
C.7 Programming the PLD Chip 12
C.8 Testing the Hardware 12
C.9 MAX7000S EPM7128SLC84-7 Summary 13
C.9.1 JTAG Jumper Settings 14
C.9.2 Prototyping Resources for Use 14
C.9.3 General Pin Assignments 14
C.9.4 Two Push-Button Switches 15
C.9.5 16 DIP Switches 15
C.9.6 16 LEDs 15
C.9.7 7-Segment LEDs 16
C.9.8 Clock 16
C.10 FLEX10K EPF10K70RC240-4 Summary 16
C.10.1 JTAG Jumper Settings 16
C.10.2 Prototyping Resources for Use 16
C.10.3 Two Push-Button Switches 17
C.10.4 8 DIP Switches 17
C.10.5 7-Segment LEDs 17
C.10.6 Clock 18
C.10.7 PS/2 Port 18
C.10.8 VGA Port 18

Appendix C UP2 Programming—Tutorial 3

Regardless of whether your design file is a HDL (VHDL or Verilog) source file or a schematic drawing, the procedures for synthesis, simulation, and PLD programming are the same. In fact, a project can contain design files of both HDL codes and schematic drawings. In this tutorial, we will program and test a 4-bit up-counter circuit on the UP2 development board. You can start with either the HDL source code or the schematic drawing for this counter circuit. In either case, the procedure and the final result will be the same.

In Tutorial 2 (Appendix B), you saw how a HDL description of a 4-bit counter circuit is synthesized and simulated in Quartus II. Test values for the input signals *Clock* and *Clear* were setup manually in the simulator. In order for the synthesized circuit to operate in the hardware, these input signals must be provided for by the hardware. For example, the *Clear* signal must be connected to an input switch, and a clock generator is needed for the *Clock* signal. Furthermore, the counter output signal *Q* must be connected to LEDs in order for you to see that the counter really works.

In this tutorial, we will expand on the 4-bit up-counter circuit by adding a clock divider, and a 7-segment decoder. The UP2 development board already has a built-in clock source running at a frequency of 25 MHz. The clock-divider circuit simply divides this clock speed down to approximately 1 Hz so that you can see the counting. Instead of viewing the 4-bit count as a binary number on discrete LEDs, a 7-segment decoder is used to convert the 4-bit counter output to drive a 7-segment LED display so that the count can be seen as a hexadecimal number. A top-level module called `up2flex` is used to connect these three components (`clockdiv`, `counter`, and `decoder`) together to form one complete circuit. This circuit is then downloaded to the FLEX chip on the UP2 development board, and after applying power, you actually can see the count being displayed on the 7-segment LED. Alternatively, you can use the top-level module `up2max` for programming the MAX chip instead.

The schematic for this complete counter circuit is shown in Figure C.1. The switches and LEDs on the UP2 board are all active-low. This is why the inverters are needed for all of the inputs and outputs in order to make them active-high, and the labels for these inputs and outputs have a *N* appended to it to denote that the signal is active-low. The eight *Vcc* lines are used optionally to turn off the second 7-segment LED on the UP2 board.

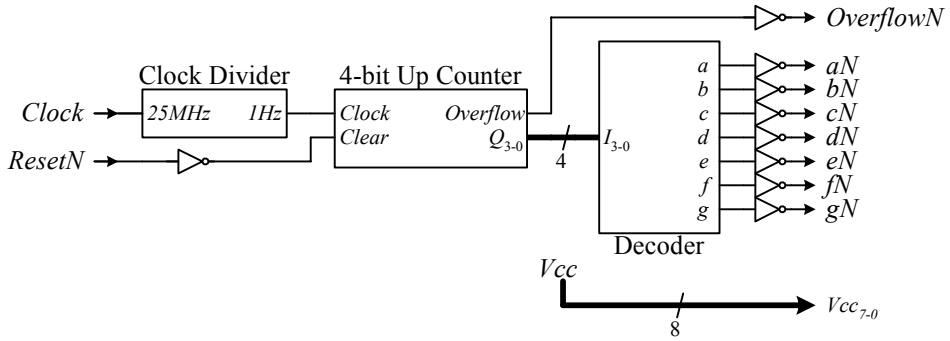


Figure C.1 Complete counter circuit for implementing on the UP2 board.

C.1 Getting Started

C.1.1 Preparing a Folder for the Project

1. Use Windows' File Manager to create a new folder for this project called `up2` in the root directory on the C drive.

   If you want to work with the schematic drawings for the counter, go to Step 2 to copy the schematic drawing source files. If you want to work with the VHDL codes for the counter, go to Step 3 to copy the VHDL source files. And if you want to work with the Verilog codes for the counter, go to Step 4 to copy the Verilog source files. After copying the respective files, continue on with the next Section C.1.2.

2. The schematic drawings for the four components, clockdiv, counter, decoder, and up2flex, are located on the accompanying CD-ROM in the four files `clockdiv.gdf`, `counter.gdf`, `decoder.gdf`, and `up2flex.gdf`, in the directory

   `<CD-ROM drive>:\Schematic Examples\Appendix C UP2 Tutorial 3\Source`.

   In addition to these four files, there is a fifth file, `ha.gdf` that is used by the counter. Using Windows File Manager, copy all of the files in this folder to the new folder `c:\up2` that you have created in Step 1.

   - All of the schematic drawing files have the extension `.gdf`, which stands for graphic design file. The Graphic Editor is used to view and edit the graphic design files.
   - The file `up2flex.gdf` is for programming the FLEX chip. The file `up2max.gdf` is for programming the MAX chip.

3. The VHDL source code for the four entities, clockdiv, counter, decoder, and up2flex, are located on the accompanying CD-ROM in the four files `clockdiv.vhd`, `counter.vhd`, `decoder.vhd`, and `up2flex.vhd`, in the directory

   `<CD-ROM drive>:\VHDL Examples\Appendix C UP2 Tutorial 3\Source`

   Using Windows File Manager, copy all of the files in this folder to the new folder `c:\up2` that you have created in Step 1.

   - All of the VHDL code files have the extension `.vhd`. The Text Editor is used to view and edit the VHDL source files.
   - The file `up2flex.vhd` is for programming the FLEX chip. The file `up2max.vhd` is for programming the MAX chip.

4. The Verilog source code for the four entities, clockdiv, counter, decoder, and up2flex, are located on the accompanying CD-ROM in the four files `clockdiv.v`, `counter.v`, `decoder.v`, and `up2flex.v`, in the directory

   `<CD-ROM drive>:\Verilog Examples\Appendix C UP2 Tutorial 3\Source`

   Using Windows File Manager, copy all of the files in this folder to the new folder `c:\up2` that you have created in Step 1.

   - All of the Verilog code files have the extension `.v`. The Text Editor is used to view and edit the Verilog source files.
   - The file `up2flex.v` is for programming the FLEX chip. The file `up2max.v` is for programming the MAX chip.

C.1.2 Creating a Project

1. Start Quartus II if it is not already started. If there are windows in Quartus II that are opened from a previous session, you can close them.
2. From the Quartus II menu, select **File | New Project Wizard**, and then click **Next** to skip the new project wizard introduction message if it appears. You should see the New Project Wizard: Directory, Name, Top-Level Entity [page 1 of 5] window shown in Figure C.2.

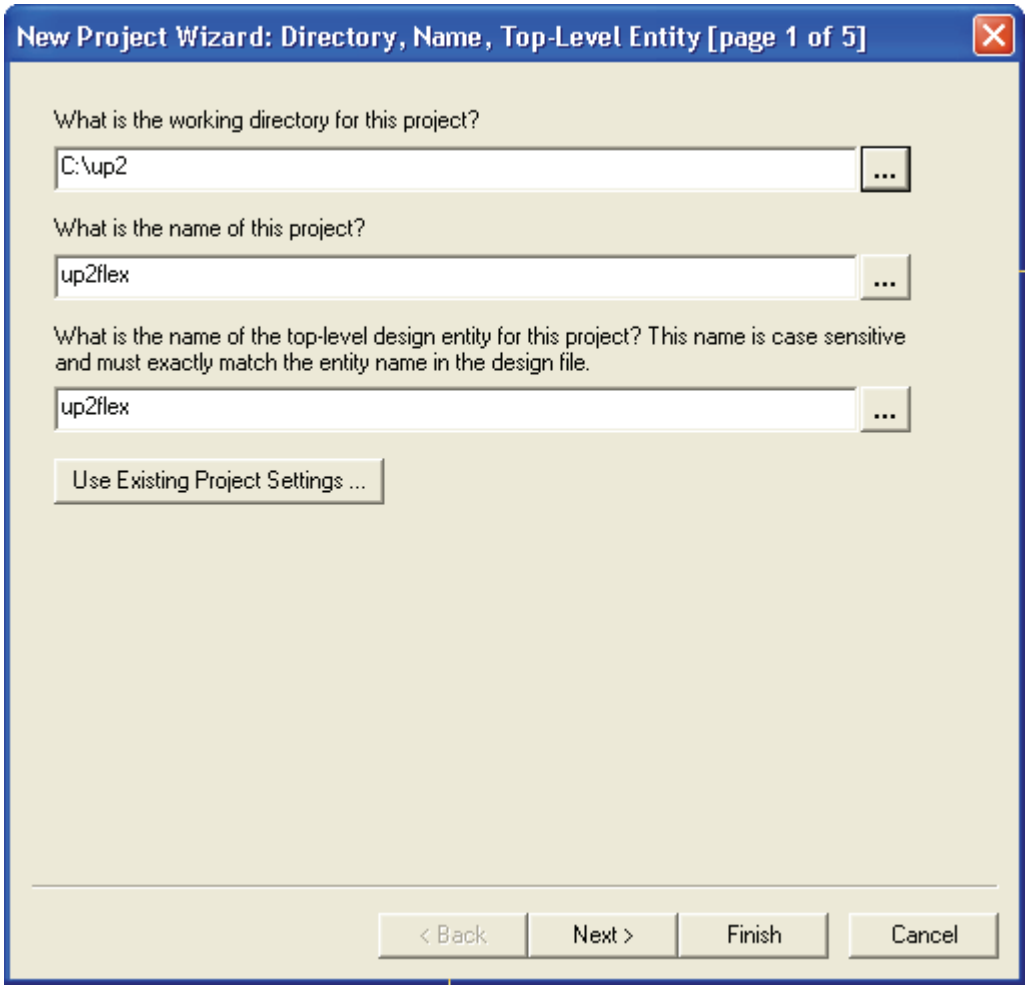


Figure C.2 The New Project Wizard: Directory, Name, Top-Level Entity window with the working directory, the project name, and the top-level entity name filled in.

3. To locate the working directory for this project, click on the ... icon next to it to browse to the directory `c:\up2` that you have created in Section C.1.1.
4. In the `up2` directory, select the file `up2flex`, and then click **OK**.
   - By selecting the file `up2flex` in the `up2` directory, the program not only fills in the directory name as `up2`, but also fills in the name of the project and the name of the top-level design file as `up2flex`.
5. Click **Next** to continue to the next window.

C.1.3 Adding Files to the Project

1. In the New Project Wizard: Add Files [page 2 of 5] window, click the **Add All** button to add all of the design files found in the `up2` directory into the project. There should be the four design files, `clockdiv`, `counter`, `decoder`, and `up2flex`.
2. Click **Next** to continue to the next window.

C.1.4 Selecting the Target Device

1. In the New Project Wizard: Family & Device Settings [page 3 of 5] window shown in Figure C.3, we select the target PLD device that we will be implementing the circuit on. The UP2 board has two different PLD devices for programming: the FLEX 10K, which is the larger PLD, and the MAX 7000S.
2. For this tutorial, we will use the FLEX 10K device. In the Family drop-down box, select **FLEX10K**.
3. In the Available devices list, select the device **EPF10K70RC240-4**. If this device is not listed, then you need to reinstall the Quartus II program, and make sure that all of the FLEX 10K devices are installed.
   - If you want to use the MAX chip, then select the MAX 7000S family, and select the EPM7128SLC84-7 device.
4. Click **Next** to continue to the next window.

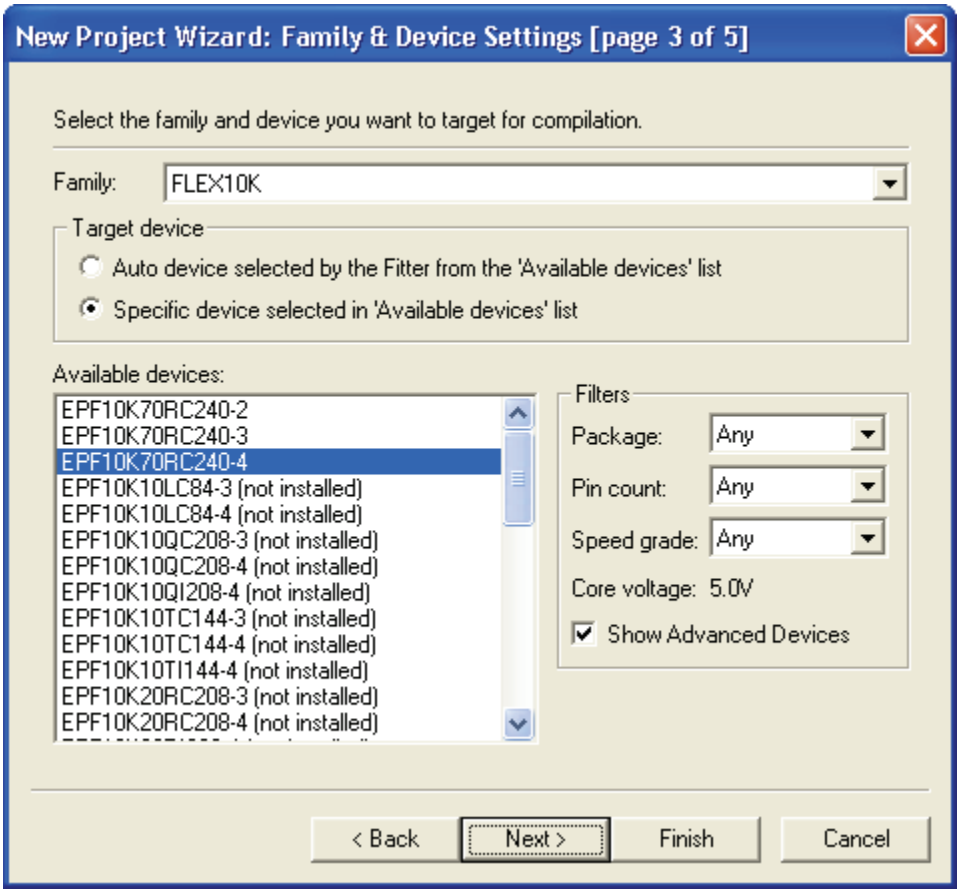


Figure C.3 The New Project Wizard: Family & Device Settings window with the device EPF10K70RC240-4 selected.

C.1.5 Completing Project Creation

1. In the New Project Wizard: EDA Tool Settings [page 4 of 5] window, we do not have any EDA tools to use for this project, so click **Next** to continue to the next window.
2. The final window is a summary of the choices that you have just made. Click **Finish** to create your new project.

C.1.6 Viewing the Design Source Files

1. To see the files that are currently associated with the project, click on the **Files** tab  in the Project Navigator window as shown in Figure C.4. If the Project Navigator window is not displayed, then click on the Project Navigator button  to display it. The design files currently added to the project are listed under the **Device Design Files** folder. If this folder is closed, you can expand it by clicking on the plus sign icon next to it.
2. Figure C.4 shows that this project has the four files, `clockdiv.v`, `counter.v`, `decoder.v`, and `up2flex.v`, added.

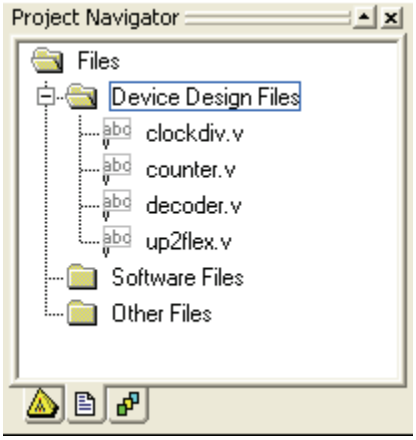


Figure C.4 Files associated with a project as shown in the Project Navigator window.

- To open a design file, simply double-click on the file that is listed in the Project Navigator window. Depending on the type of file, the associated editor will be used. The Graphic Editor is used to edit a Block Diagram/Schematic File, and a Text Editor is used to edit a VHDL or Verilog text file.

3. Double-click on the `counter.v` file to open it. The text editor is used to edit this source file. You can scroll through this file to see the Verilog code for describing the 4-bit binary counter. For this tutorial, we will not make any modifications, so you can close this file after viewing the code.

C.2 Analysis and Synthesis

After describing your circuit with the Text Editor or drawing your circuit with the Graphic Editor, the next step is to analyze and synthesize it. During this step, Quartus II collects all of the necessary information about your circuit, and produces a netlist for it.

1. From the Quartus II menu, select **Processing | Start | Start Analysis & Synthesis** to synthesize the circuit. Alternatively, you can click on the icon  .
2. If there are no errors in your circuit description, you should see the message "Quartus II Analysis & Synthesis was successful" in the Message window at the bottom.
3. If there are errors then they will be reported in the Message window and highlighted in red. You can double-click on the error message to see where the error is in the source file.

C.3 Circuit Simulation

Circuit simulation allows you to observe the behavior of the circuit before actually implementing the circuit in hardware. In practice, it is always advisable to simulate the circuit for correctness first before implementation. For this tutorial, we will skip this step since we have already simulated the circuit in Tutorial 2 (Appendix B).

C.4 Mapping the I/O Signals

Since we want to implement the circuit on a PLD, we need to assign all of the I/O signals from our circuit to the actual pins on the PLD. From Section C.1.4, we have already chosen the FLEX 10K PLD to implement our circuit on. We will now use the Assignment Editor to map each of the I/O signals from our circuit to the pins on the FLEX 10K chip.

1. From the Quartus II menu, select **Assignments | Assignment Editor** to bring up the Assignment Editor similar to Figure C.5. Alternatively, you can click on the **Assignment Editor** icon .

2. Click on the **Pin** button Pin in the Category Bar at the top of the Assignment Editor window to list the available I/O signals from the circuit. If the Category Bar is not shown, then click on the **Display the Category Bar** icon on the left side of the window. All of the available I/O signals from the circuit will be listed under the **To** column. If the I/O signals are not listed, then you need to go back and do the Analysis and Synthesis step in Section C.2.

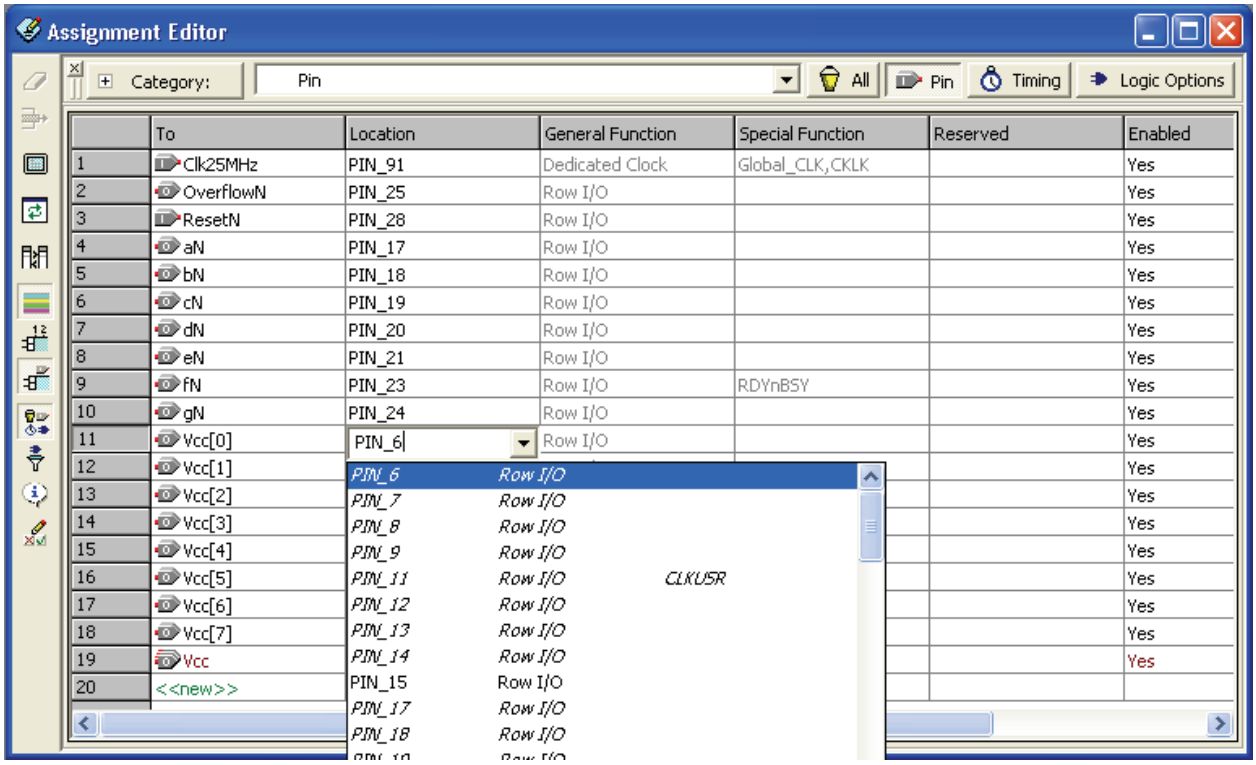


Figure C.5 The Assignment Editor showing the pin assignments of the FLEX 10K70RC240-4 chip to the I/O signals for the circuit. Pin 6 from the pop-up list is currently being assigned to the signal Vcc[0].

3. For each I/O signal name, double-click on the cell next to the signal name under the **Location** column to bring up a pop-up list of all the assignable pins from the PLD. Select the pin number that you want to assign to that I/O signal. Figure C.5 shows that Pin 6 is currently being assigned to the signal Vcc[0].

   - Alternatively, instead of using the pop-up list, you can type in the pin number.
   - Pins that are already assigned to a signal will be listed in italic.
   - Perform the following signal-to-pin assignments for the FLEX chip.

Signal	Pin Number	Comment
Clk25MHz	91	Pin 91 is connected to the built-in 25 MHz clock source
OverflowN	25	Pin 25 is connected to the decimal point on digit 2 of the FLEX 7-segment LED
ResetN	28	Pin 28 is connected to push-button switch FLEX_PB1
aN	17	Pin 17 is connected to segment a on digit 2 of the FLEX 7-segment LED
bN	18	Pin 18 is connected to segment b on digit 2 of the FLEX 7-segment LED
cN	19	Pin 19 is connected to segment c on digit 2 of the FLEX 7-segment LED
dN	20	Pin 20 is connected to segment d on digit 2 of the FLEX 7-segment LED
eN	21	Pin 21 is connected to segment e on digit 2 of the FLEX 7-segment LED
fN	23	Pin 23 is connected to segment f on digit 2 of the FLEX 7-segment LED
gN	24	Pin 24 is connected to segment g on digit 2 of the FLEX 7-segment LED
Vcc[0]	6	Optional assignment to turn off segment a on digit 1 of the FLEX 7-segment LED
Vcc[1]	7	Optional assignment to turn off segment b on digit 1 of the FLEX 7-segment LED
Vcc[2]	8	Optional assignment to turn off segment c on digit 1 of the FLEX 7-segment LED
Vcc[3]	9	Optional assignment to turn off segment d on digit 1 of the FLEX 7-segment LED
Vcc[4]	11	Optional assignment to turn off segment e on digit 1 of the FLEX 7-segment LED
Vcc[5]	12	Optional assignment to turn off segment f on digit 1 of the FLEX 7-segment LED
Vcc[6]	13	Optional assignment to turn off segment g on digit 1 of the FLEX 7-segment LED
Vcc[7]	14	Optional assignment to turn off the decimal point on digit 1 of the FLEX 7-segment LED

- Perform the following signal-to-pin assignments if you have selected to use the MAX chip from Section C.1.4.

Signal	Pin Number	Comment
Clk25MHz	83	Pin 83 is connected to the built-in 25 MHz clock source
OverflowN	79	Pin 79 is connected to the decimal point on digit 2 of the MAX 7-segment LED
ResetN	1	Connect a hook-up wire between Pin 1 and the pushbutton switch MAX_PB1
aN	69	Pin 69 is connected to segment a on digit 2 of the MAX 7-segment LED
bN	70	Pin 70 is connected to segment b on digit 2 of the MAX 7-segment LED
cN	73	Pin 73 is connected to segment c on digit 2 of the MAX 7-segment LED
dN	74	Pin 74 is connected to segment d on digit 2 of the MAX 7-segment LED
eN	76	Pin 76 is connected to segment e on digit 2 of the MAX 7-segment LED
fN	75	Pin 75 is connected to segment f on digit 2 of the MAX 7-segment LED
gN	77	Pin 77 is connected to segment g on digit 2 of the MAX 7-segment LED
Vcc[0]	58	Optional assignment to turn off segment a on digit 1 of the MAX 7-segment LED
Vcc[1]	60	Optional assignment to turn off segment b on digit 1 of the MAX 7-segment LED
Vcc[2]	61	Optional assignment to turn off segment c on digit 1 of the MAX 7-segment LED
Vcc[3]	63	Optional assignment to turn off segment d on digit 1 of the MAX 7-segment LED
Vcc[4]	64	Optional assignment to turn off segment e on digit 1 of the MAX 7-segment LED
Vcc[5]	65	Optional assignment to turn off segment f on digit 1 of the MAX 7-segment LED
Vcc[6]	67	Optional assignment to turn off segment g on digit 1 of the MAX 7-segment LED
Vcc[7]	68	Optional assignment to turn off the decimal point on digit 1 of the MAX 7-segment LED

4. Repeat Step 3 until all of the I/O signals have been assigned to the correct pins.

C.5 Fitting the Netlist and Pins to the PLD

Now that we have created the netlist for the circuit (from Section C.2), and have mapped all of the I/O signals to the actual pins on the PLD (from Section C.4), the next step is to fit the netlist and the pin assignments to the given PLD. This requires a full compilation of the circuit, which involves four individual steps: Analysis & Synthesis, Fitter, Assembler, and Timing Analyzer.

1. From the Quartus II menu, select **Processing | Start Compilation** to start the full compilation of the circuit.

Alternatively, you can click on the **Start Compilation** icon  . The full compilation will automatically go through the four steps.

C.6 Hardware Setup

Before the circuit can be downloaded onto the PLD, we need to setup the UP2 board and the software driver for communicating with the board.

C.6.1 Installing the ByteBlaster Driver

Windows XP, 2000, and NT users must install the ByteBlaster device driver in order to program the PLD chips on the UP2 board. This step only needs to be done once after the initial installation of the Quartus II program. Perform the following steps, shown in Figure C.6, to install the driver:

1. Get a command prompt window from **Windows Start | Run**, and type in `cmd <Enter>`.
2. Use the **cd** command to change to the directory where you have installed the Quartus II program. The following example changes to the default installation directory `c:\altera\quartus51` for version 5.1 of the program.

```
c:
cd \altera\quartus51
```

3. From the installation directory, change to the directory `drivers\i386`.

```
cd drivers\i386
```

4. Enter the command `bblpt /i <Enter>` at the command prompt to install the driver. You should see a message saying that the driver has been installed successfully.

```
bblpt /i
```

Refer to the ByteBlaster installation instructions in the UP2 User Guide on the accompanying CD-ROM for more information.

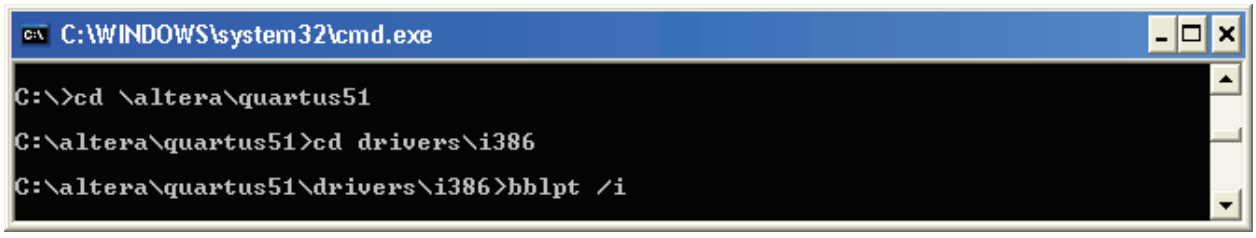


Figure C.6 Installation of the ByteBlaster device driver.

C.6.2 Selecting the Hardware Device

1. From the Quartus II menu, select **Tools | Programmer** to bring up the Programmer window as shown in Figure C.7. Alternatively, you can click on the **Programmer** icon  .

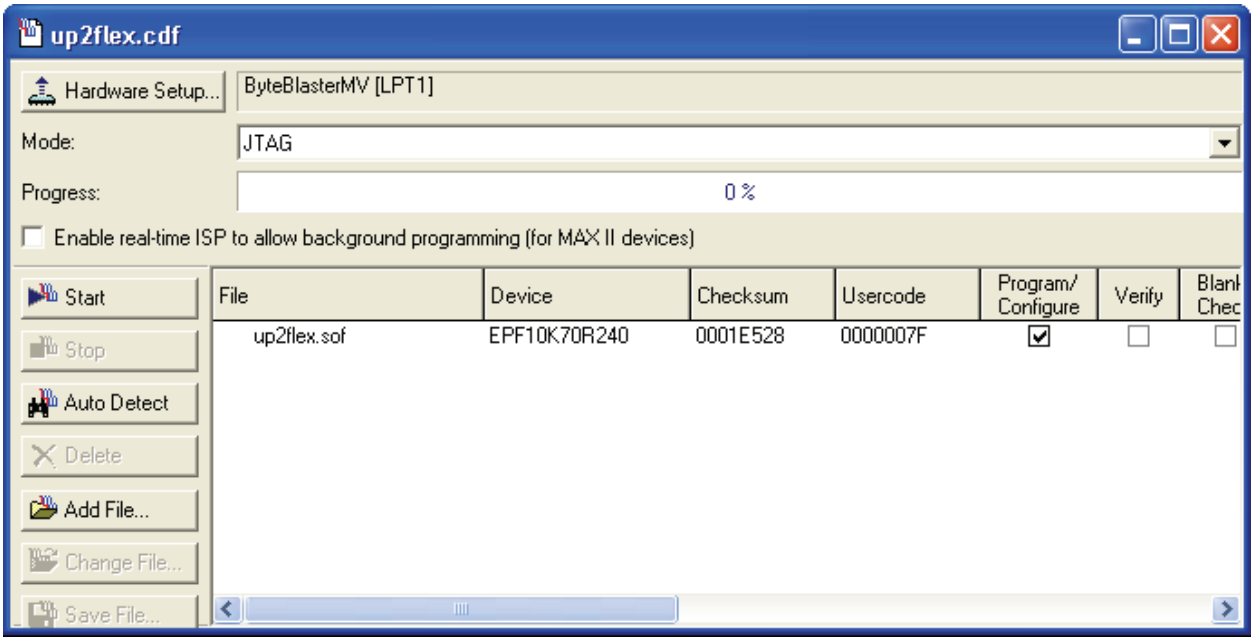


Figure C.7 Programmer window.

2. If the device listed next to the **Hardware Setup** button is `ByteBlasterMV [LPT1]`, then skip the remaining steps in this section and go to Section C.6.3. Otherwise, continue with Step 3 to install the hardware device.
3. In the Programmer window, click on the **Hardware Setup** button. This will bring up the Hardware Setup window as shown in Figure C.8.

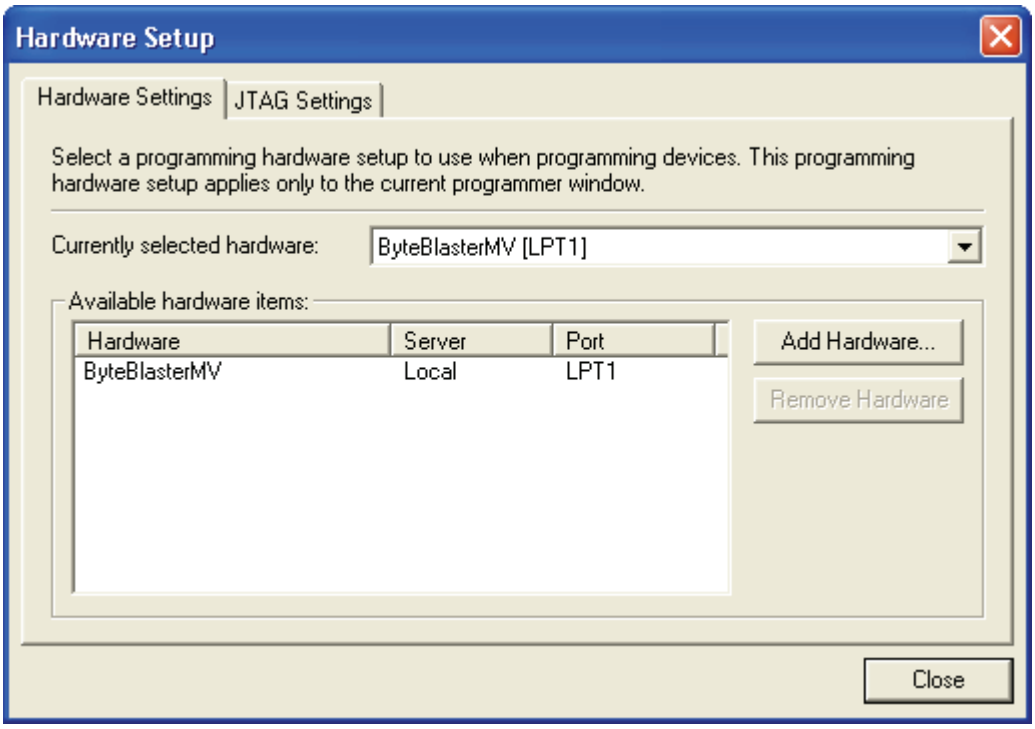


Figure C.8 Hardware Setup window for selecting the correct hardware device.

4. Under the **Hardware Setup** tab, make sure that the ByteBlasterMV on port LPT1 is listed in the available

hardware items, and that the **Currently selected hardware** is `ByteBlasterMV [LPT1]`. If the ByteBlasterMV is listed in the available hardware items, but it is not the currently selected hardware, then click on the **Currently selected hardware** drop-down list to select the ByteBlasterMV [LPT1] option. If this is correct, then skip Step 3 and go to Step 4.

5. If `No Hardware` is listed in the **Currently selected hardware**, then click on the **Add Hardware** button. Under **Hardware type**, select `ByteBlasterMV`. The **Port** should be `LPT1`. Click on **OK**. If the **Port** list says that the "Kernal mode driver not installed," then the ByteBlaster driver was not installed, or installed incorrectly. Go back to Section C.6.1 to reinstall the driver.
6. Click **Close**.
7. The device listed next to the **Hardware Setup** button in the Programmer window should be `ByteBlasterMV [LPT1]`, as shown in Figure C.7. If not, then you need to correct the problem by going back to Section C.6.1 to reinstall the driver.

C.6.3 JTAG Jumper Settings

The four JTAG jumpers, TDI, TDO, DEVICE, and BOARD, on the UP2 board must be set correctly depending on which PLD chip you want to use. Refer to Figure C.9 for the location of these four jumpers.

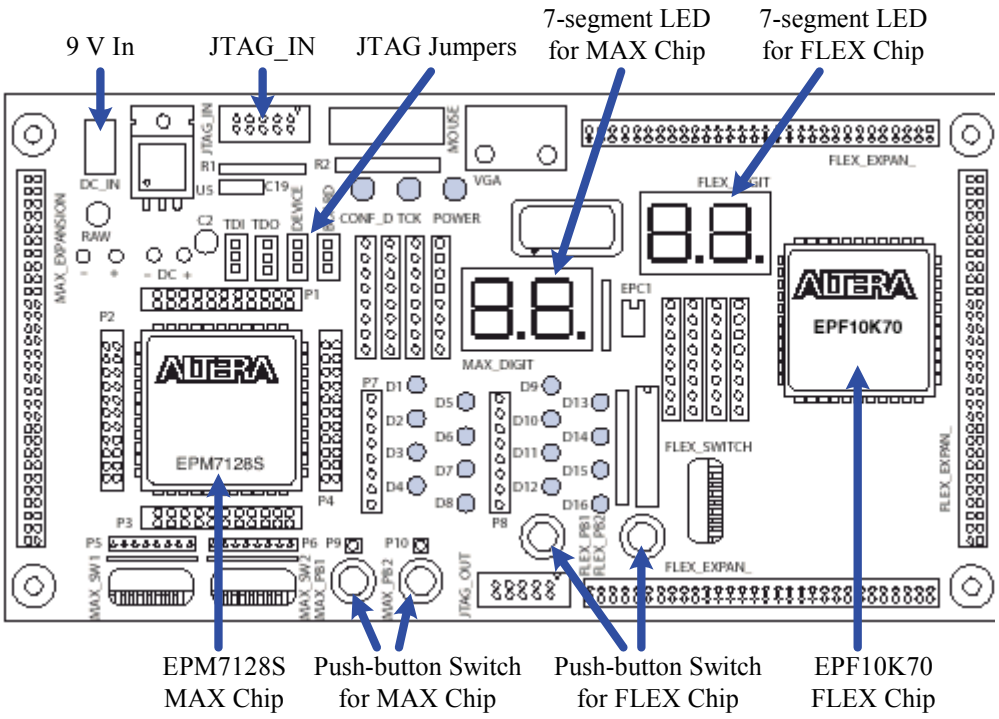
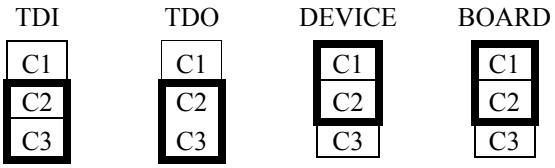
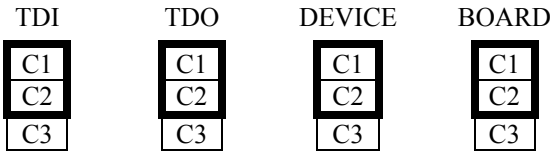


Figure C.9 UP2 development board layout.

1. In Section C.1.4, we selected the FLEX 10K device for implementing the circuit, so we need to set up the jumpers for the FLEX chip as follows.



- Use the following settings if you have selected to use the MAX chip from Section C.1.4.



C.6.4 Hardware Connections

1. Attach the ByteBlasterMV parallel cable directly to the PC's parallel port and the JTAG_IN connector on the UP2 development board. Refer to Figure C.9 for the location of the JTAG_IN connector.
2. Plug in the 9 V power supply.

C.7 Programming the PLD Chip

1. We are now ready to program the PLD chip on the UP2 board with our circuit. If the Programmer window is not up, then click on the **Programmer** icon  to bring it up.
2. Make sure that the project file name listed is **up2flex.sof**, and the device listed is **EPF10K70R240**.
3. Place a check mark in the **Program/Configure** check box.
4. Click on the **Start** button 

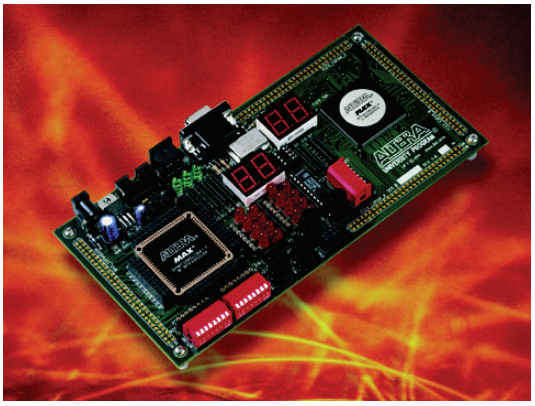
to start the programming of your circuit, and watch the progress bar.
5. When the progress bar reaches 100%, the programming is completed, and there should be a message in the Message window saying that the operation was performed successfully.
    - If you get an error message in the Message window saying that the programming is unsuccessful, the most likely reason is because the JTAG jumper settings on the board do not match the programming device that you have selected in Section C.1.4. First, go back to Section C.6.3 to check the jumper settings. If the jumper settings are correct, then check the device selection under **Assignments | Device**. If you need to change the device, then you will have to re-map the I/O pins, and re-synthesize the circuit.

C.8 Testing the Hardware

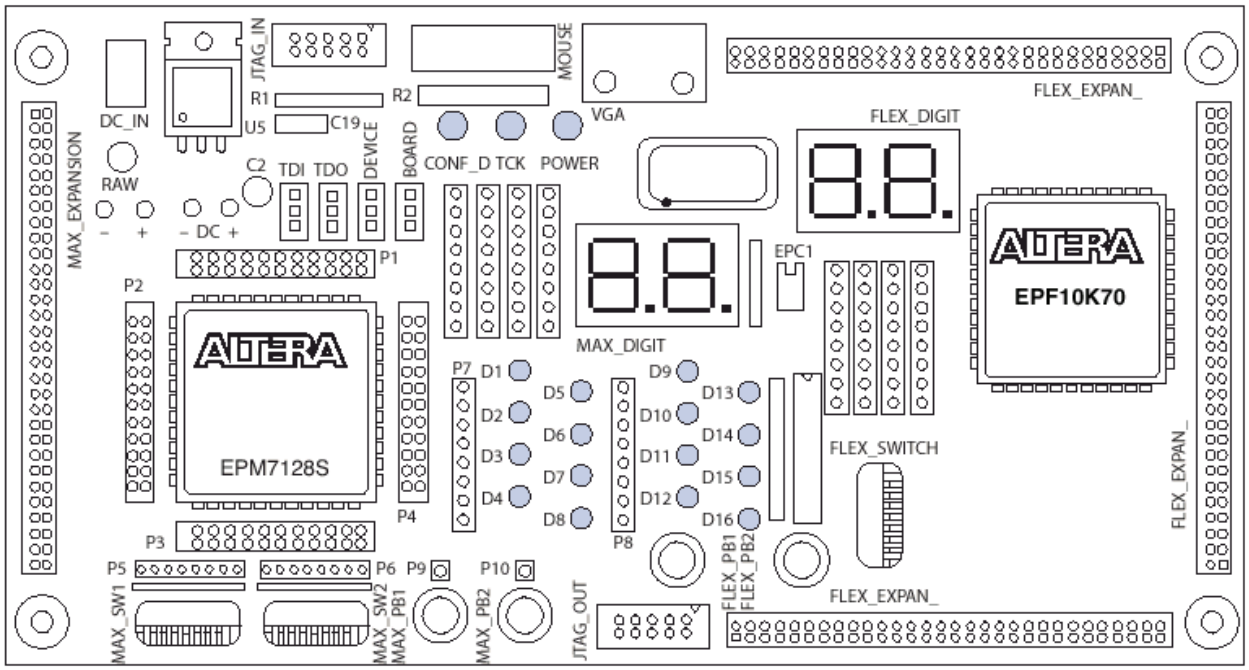
1. Immediately after programming the FLEX chip, you should see the counter counts on the 7-segment LED. The 7-segment LED will repeatedly display the 16 hexadecimals in sequence from 0 to F. When the count reaches F, the Overflow signal will turn on the decimal point light. The push-button FLEX_PB1 is connected to the Clear signal, so when you press this button, the counter will reset to zero.
    - If you have programmed the MAX chip, you need to first connect a hook-up wire between Pin 1 on the MAX dual-row female header strip P1 and the push-button MAX_PB1. This push-button is now the Reset button. Refer to Section C.9.3 for the location of Pin 1 on the header strip, and Section C.9.4 for the MAX_PB1 push-button switch. You may want to make all of the jumper connections described in Section C.9 now, since all of the circuits implemented on the MAX chip require these connections.

C.9 MAX7000S EPM7128SLC84-7 Summary

The UP2 development board contains two PLDs: a MAX EPM7128SLC84-7 and a FLEX EPF10K70RC240-4. The MAX chip has a capacity of 2,500 gates, and the FLEX chip has a capacity of 70,000 gates. The board provides switches, LEDs, and connectors for prototyping digital circuits. Figure C.10 shows a picture and physical layout of the UP2 development board.



(a)



(b)

Figure C.10 UP2 development board: (a) picture; (b) layout.

This section provides a summary of the resources available for the MAX EPM7128SLC84-7 PLD and the jumper connections for using them. All of the LEDs and switches are active-low. In other words, a 0 turns on a LED, and a 0 is generated when a switch is pressed or in the on position. Refer to the UP2 User Guide on the accompanying CD-ROM for a detailed description and usage of the board.

C.9.1 JTAG Jumper Settings

The four JTAG jumpers, TDI, TDO, DEVICE, and BOARD, on the UP2 development board must be set according to Figure C.11 in order to program the MAX EPM7128SLC84-7 PLD.

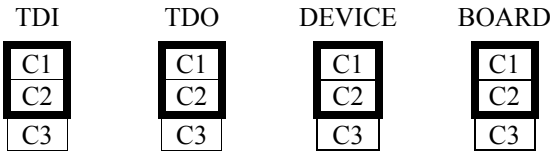


Figure C.11 JTAG jumper settings for the MAX EPM7128SLC84-7 device.

C.9.2 Prototyping Resources for Use

The following resources are available for use with the MAX EPM7128SLC84-7 PLD.

- Female headers for signal pins
- Two momentary push-button switches
- Two octal, dual in-line package (DIP) switches
- 16 LEDs
- Dual-digit 7-segment LED display
- On-board oscillator (25.175 MHz)
- Expansion port with 42 I/O pins and the dedicated global CLR, OE1, and OE2/GCLK2 pins

C.9.3 General Pin Assignments

The signal pins on the MAX EPM7128SLC84-7 chip are connected to four dual-row female header strips surrounding the chip. The pin numbers for the chip are printed on the board and summarized in Figure C.12. An "×" indicates an unconnected pin, and a crossed-out pin number indicates that the pin is non-assignable.

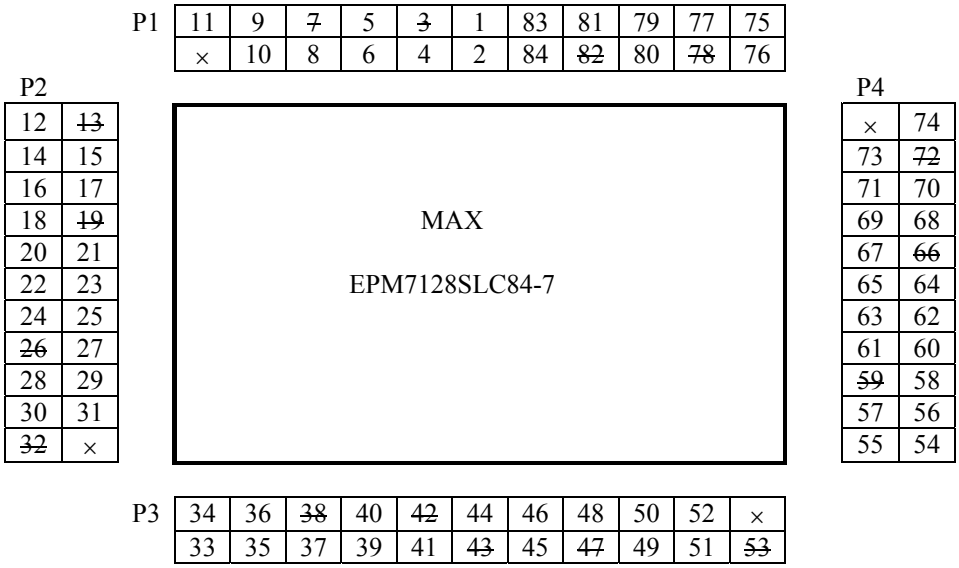


Figure C.12 Connection pins for the MAX EPM7128SLC84-7 device.

C.9.4 Two Push-Button Switches

- The MAX_PB1 and MAX_PB2 are two push-button switches that provide active-low signals (i.e., they output a 0 when pressed).
- Connections to these two push-buttons are made by inserting one end of the hook-up wire into the female header associated with it.
- All of the circuits for programming the MAX chip on the accompanying CD-ROM map the two push-button switches to the header pins on the MAX chip as shown Figure C.13. You may want to make these connections now using hook-up wires and leave them connected this way for all of the exercises.

MAX_PB Switch	Pin
MAX_PB1	1
MAX_PB2	2

Figure C.13 Hook-up wire connections for the two MAX push-button switches.

C.9.5 16 DIP Switches

- The MAX_SW1 and MAX_SW2 are two dual in-line package (DIP) switches that provide active-low signals (i.e., they output a 0 when set to on).
- Connections to these DIP switches are made by inserting one end of the hook-up wire into the female header associated with it.
- All of the circuits for programming the MAX chip on the accompanying CD-ROM map the 16 DIP switches to the header pins on the MAX chip, as shown in Figure C.14. You may want to make these connections now using hook-up wires and leave them connected this way for all of the exercises.

MAX_SW1 Switch	Pin
MAX_SW1 Switch 1	33
MAX_SW1 Switch 2	34
MAX_SW1 Switch 3	35
MAX_SW1 Switch 4	36
MAX_SW1 Switch 5	37
MAX_SW1 Switch 6	39
MAX_SW1 Switch 7	40
MAX_SW1 Switch 8	41

MAX_SW2 Switch	Pin
MAX_SW2 Switch 1	44
MAX_SW2 Switch 2	45
MAX_SW2 Switch 3	46
MAX_SW2 Switch 4	48
MAX_SW2 Switch 5	49
MAX_SW2 Switch 6	50
MAX_SW2 Switch 7	51
MAX_SW2 Switch 8	52

Figure C.14 Hook-up wire connections for the 16 MAX DIP switches.

C.9.6 16 LEDs

- Each LED is turned on with a logic 0.
- Connections to the 16 LEDs are made by inserting one end of the hook-up wire into the female header associated with that LED.
- All of the circuits for programming the MAX chip on the accompanying CD-ROM map the 16 LEDs to the header pins on the MAX chip, as shown in Figure C.15. You may want to make these connections now using hook-up wires and leave them connected this way for all of the exercises.

LEDs	Pin
D1	11
D2	12
D3	15
D4	16

LEDs	Pin
D9	22
D10	24
D11	25
D12	27

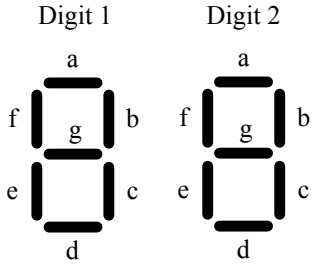
D5	17
D6	18
D7	20
D8	21

D13	28
D14	29
D15	30
D16	31

Figure C.15 Hook-up wire connections for the 16 MAX LEDs.

C.9.7 7-Segment LEDs

- Each LED segment is turned on with a logic 0.
- The dual-digit 7-segment display LEDs are connected permanently to the pins on the MAX chip, as shown in Figure C.16.



Segment	Digit 1 Pins	Digit 2 Pins
a	58	69
b	60	70
c	61	73
d	63	74
e	64	76
f	65	75
g	67	77
Decimal Point	68	79

Figure C.16 Connections for the two MAX 7-segment LEDs.

C.9.8 Clock

A 25.175 MHz clock signal is connected permanently to Pin 83 on the MAX chip.

C.10 FLEX10K EPF10K70RC240-4 Summary

This section provides a summary of the resources available for the FLEX EPF10K70RC240-4 PLD and the pin connections for using them. All of the LEDs and switches are active-low. In other words, a 0 turns on a LED, and a 0 is generated when a switch is pressed or in the on position. Refer to the UP2 User Guide on the accompanying CD-ROM for a detailed description and usage of the board.

C.10.1 JTAG Jumper Settings

The four JTAG jumpers, TDI, TDO, DEVICE, and BOARD, on the UP2 development board must be set according to Figure C.17 in order to program the FLEX EPF10K70RC240-4 PLD.

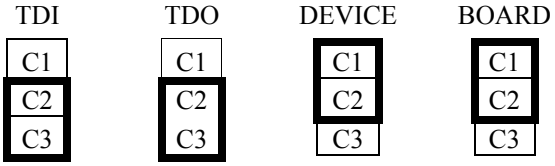


Figure C.17 JTAG jumper settings for the FLEX EPF10K70RC240-4 device.

C.10.2 Prototyping Resources for Use

The following resources are available for use with the FLEX EPF10K70RC240-4 PLD.

- Two momentary push-button switches

- One octal DIP switch
- Dual-digit 7-segment display
- On-board oscillator (25.175 MHz)
- PS/2 mouse or keyboard port
- VGA port
- Three expansion ports, each with 42 I/O pins and seven global pins

C.10.3Two Push-Button Switches

- The FLEX_PB1 and FLEX_PB2 are two push-button switches that provide active-low signals (i.e., they output a 0 when pressed).
- They are connected permanently to the pins on the FLEX chip, as shown in Figure C.18:

Name	Pin
FLEX_PB1	28
FLEX_PB2	29

Figure C.18 Connections for the two FLEX push-button switches.

C.10.48 DIP Switches

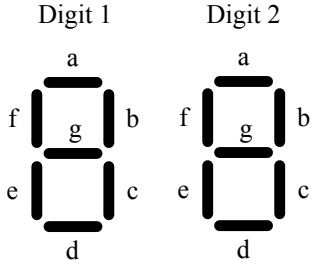
- The FLEX_SW1 is a dual in-line package (DIP) switch that provides active-low signals (i.e., it outputs a 0 when set to on).
- They are connected permanently to the pins on the FLEX chip, as shown in Figure C.19.

Name	Pin
FLEX_SWITCH-1	41
FLEX_SWITCH-2	40
FLEX_SWITCH-3	39
FLEX_SWITCH-4	38
FLEX_SWITCH-5	36
FLEX_SWITCH-6	35
FLEX_SWITCH-7	34
FLEX_SWITCH-8	33

Figure C.19 Connections for the 8 FLEX DIP switches.

C.10.57-Segment LEDs

- Each LED segment is turned on with a logic 0.
- The dual-digit 7-segment display LEDs are connected permanently to the pins on the FLEX chip, as shown in Figure C.20.



Segment	Digit 1 Pins	Digit 2 Pins
a	6	17
b	7	18
c	8	19
d	9	20
e	11	21
f	12	23
g	13	24
Decimal Point	14	25

Figure C.20 Connections for the two FLEX 7-segment LEDs.

C.10.6 Clock

- A 25.175 MHz clock signal is connected permanently to Pin 91 on the FLEX chip.

C.10.7 PS/2 Port

- A 6-pin mini-DIN connector is available to receive data from either a PS/2 mouse or a PS/2 keyboard.
- The connector is connected permanently to the FLEX chip, as shown in Figure C.21.

Mouse/Keyboard Signal	Mini-DIN pin	FLEX pin
MOUSE_CLK, KEYBOARD_CLK	1	30
MOUSE_DATA, KEYBOARD_DATA	3	31
VCC	5	–
GND	2	–

Figure C.21 Connections for the PS/2 port.

C.10.8 VGA Port

- The VGA interface allows the FLEX device to control an external monitor.
- The D-sub VGA connector is connected permanently to the FLEX chip, as shown in Figure C.22.

VGA Signal	Mini-DIN Pin	FLEX Pin
RED	1	236
GREEN	2	237
BLUE	3	238
GND	6, 7, 8, 10, 11	–
HORIZ_SYNC	13	240
VERT_SYNC	14	239
No connect	4, 5, 9, 15	–

Figure C.22 Connections for the VGA interface.