UltraController Lab1
Time-Of-Day Timer

Version 1.0
February 2004

1 Introduction

A simple design is used to demonstrate how easily the UltraController reference design can be modified and ported to perform a variety of applications. This lab will use the UltraController **uc_4i_4d_1ppc_vhdl** reference design obtained from the UltraController web page to implement a simple Time-Of-Day timer. This design will be implemented and tested on the Memec Design Virtex-II Pro 2VP4LC development board. The following figure shows a high-level block diagram of the Time-Of-Day Timer reference design.

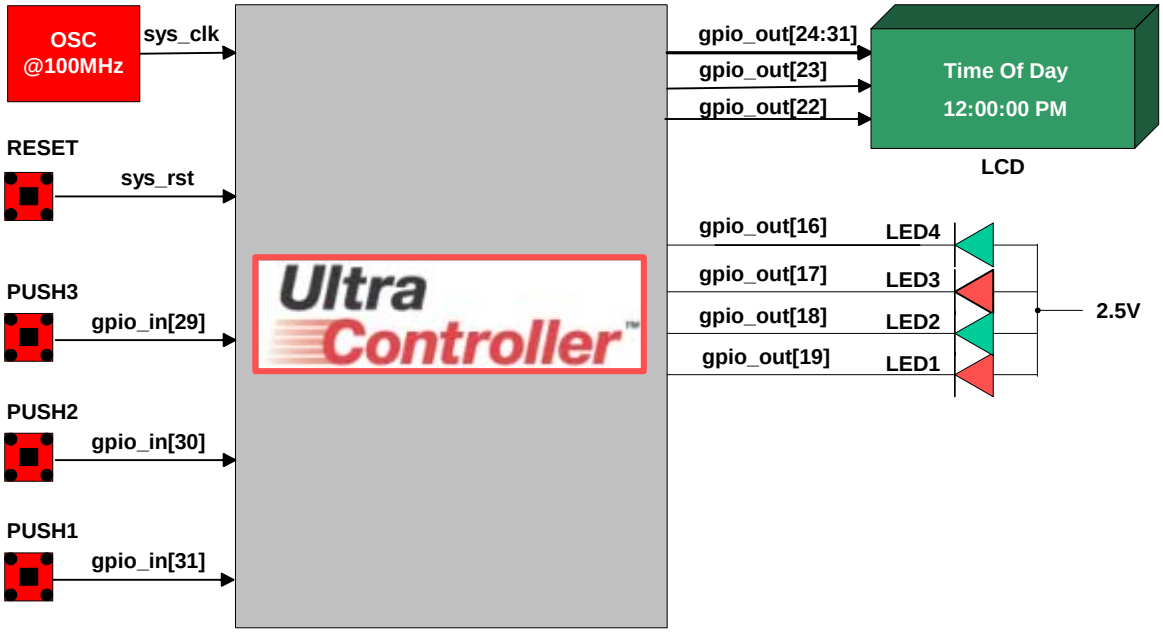

Figure 1 – Time-Of-Day Timer Block Diagram

2 Implementing the Design

The UltraController **uc_4i_4d_1ppc_vhdl** reference design has been downloaded from the Xilinx web site and placed in the **C:\UltraController_Workshop\UltraController_Lab1** folder. This reference design will be used to implement the **Time-Of-Day** Timer design. The **C:\UltraController_Workshop\UltraController_Lab1** folder contains the following files and sub-folders:

- **UC_4i_4d_1ppc_vhdl** folder – The UltraController reference design folder obtained from the UltraController web page (the **UC_4i_4d_1ppc_vhdl.zip** file is obtained from the web site and unzipped to the **UC_4i_4d_1ppc_vhdl** folder**).**
- **time_of_day.c** – The C source file for the Time-Of-Day timer design.
- **time_of_day.ucf** – This file is the top-level UCF for the Memec Design Virtex-II Pro **2VP4LC** development board. The UltraController **uc_4i_4d_1ppc_vhdl** reference design targets the Memec Design Virtex-II Pro **FG456** board. However, we will be using the Memec Design Virtex-II Pro **2VP4LC** board during this lab, so this modified UCF is needed.

2.1 Starting EDK and Opening the XPS project

Start EDK and open the UltraController reference design project by performing the following steps:

- Select **Start>Programs>Xilinx Embedded Development Kit 6.2> Xilinx Platform Studio** to start XPS.
- Select **File>Open Project** from the XPS GUI
- Browse to the C:\UltraController_Workshop\UltraController_Lab1\UC_4i_4d_1ppc_vhdl folder, select **system.xmp**, and click **Open**.

The following figure shows the XPS GUI after opening the project in EDK.

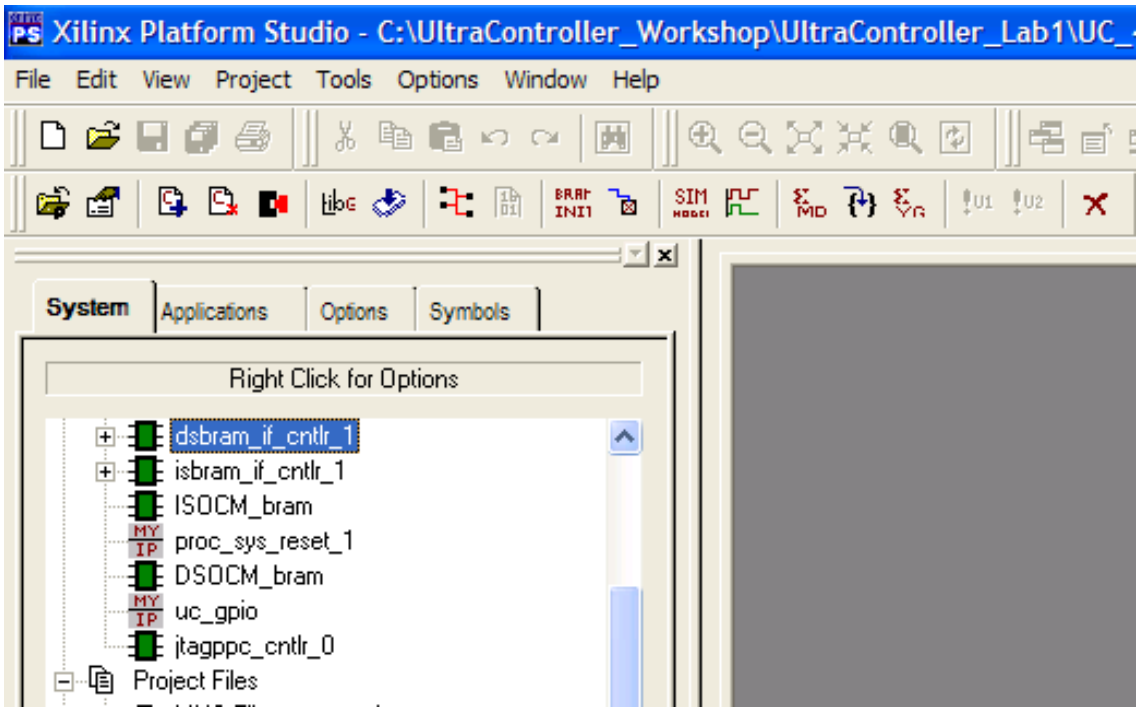

Figure 2 - UltraController Reference Design in the XPS GUI

Click on the **Applications** tab and expand the **Sources** to view the C source files used in the UltraController **uc_4i_4d_1ppc_vhdl** reference design as shown in the following figure.

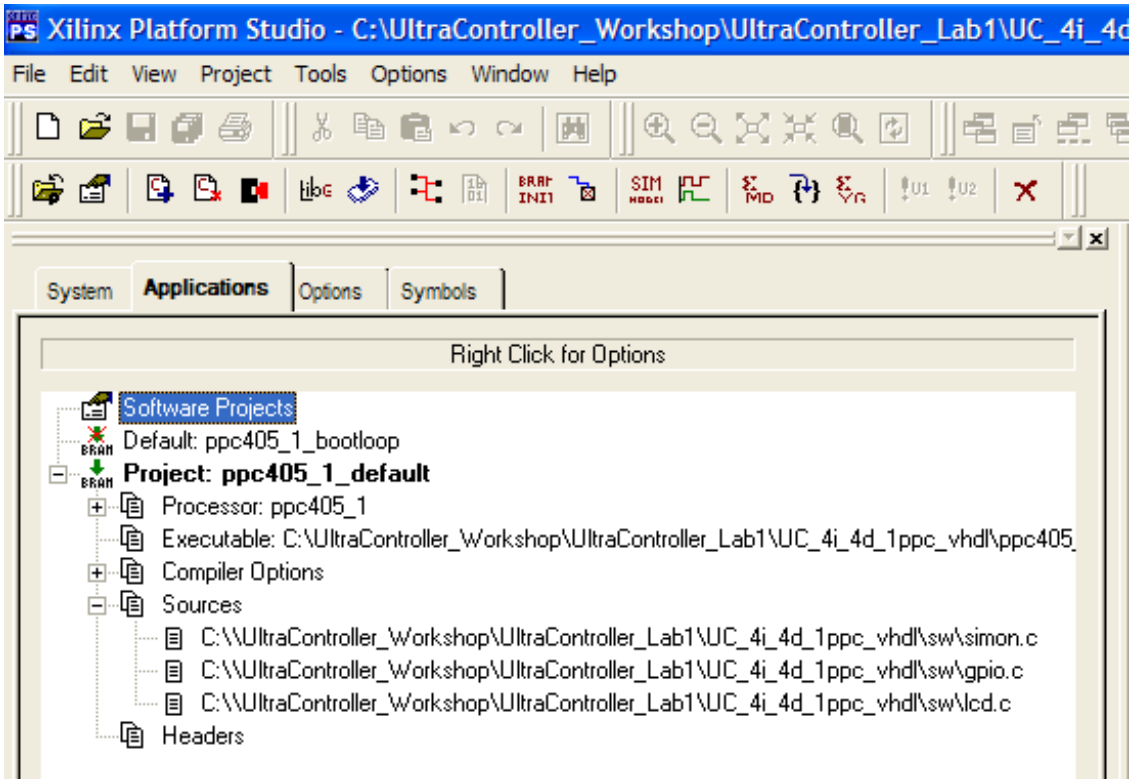

Figure 3 – UltraController Reference Design Software Source Files

2.2 Opening the gpio.c Source File in the XPS GUI

As shown in the above figure, the UltraController **uc_4i_4d_1ppc_vhdl** reference design uses three C source files. The **simon.c** is the main program, the **gpio.c** is the UltraController GPIO software driver, and the **lcd.c** is the driver for the LCD. Double-click on the **gpio.c** file to open it in the XPS GUI and go to the section that starts with the following line of code.

```
054
055     Xuint32 OUT_PORT[1] __attribute__ ((section(".io_reg"))) = { 0 }; // Address 0
056     Xuint32 IN_PORT[1]  __attribute__ ((section(".io_reg"))) = { 0 }; // Address 4
057
058 // GPIO Pin Assignments
059 /*
060     Description     Code Bit    Output Bit
061     LCD D0          0           31
062     LCD D1          1           30
063     LCD D2          2           29
064     LCD D3          3           28
065     LCD D4          4           27
066     LCD D5          5           26
067     LCD D6          6           25
068     LCD D7          7           24
069     LCD RS          8           23
070     LCD EN          9           22
071     --              10          21
072     Sound           11          20
073     LED 1           12          19
074     LED 2           13          18
075     LED 3           14          17
076     LED 4           15          16
077
078     Description     Code Bit    Input Bit
079     SW 1            0           31
080     SW 2            1           30
081     SW 3            2           29
082 */
```

Figure 4 – UltraController GPIO Software driver (gpio.c)

By looking at the above comment section in the **gpio.c** source file of the UltraController **uc_4i_4d_1ppc_vhdl** reference design and also by looking at the **Time-Of-Day** timer design block diagram, you will see that the UltraController **uc_4i_4d_1ppc_vhdl** reference design uses the same external I/O devices and GPIO input/output port connections as the **Time-Of-Day** Timer design with the exception of the GPIO bit 20 (Sound output).

So, if the Sound output is not used, the GPIO port connections for the **uc_4i_4d_1ppc_vhdl** reference design can be used for the **Time-Of-Day** Timer design. Given this, the software is the only component of the UltraController **uc_4i_4d_1ppc_vhdl** reference design that needs to be modified in order to port it to the Time-Of-Day Timer design. Please close the **gpio.c** source file in the XPS GUI.

2.3 Adding the Time-Of-Day Timer C Source File to the Design

The application software for the UltraController **uc_4i_4d_1ppc_vhdl** reference design implements the Simon game. So, this code is the only part of the reference design that needs to be replaced to port the design to the **Time-Of-Day** Timer design. Perform the following steps to replace the simon.c file with the time_of_day.c source file.

- Right-click on **simon.c** source file to select it and then select "**Delete File**" to delete this source file from the project.

- Right-click on "**Sources**" and then select "**Add Files**". Browse to the **C:\UltraController_Workshop\UltraController_Lab1** folder, select the **time_of_day.c** file and then click **Open**.

The following figure shows the XPS GUI after the simon.c file deletion and the time_of_day.c file addition to the project.

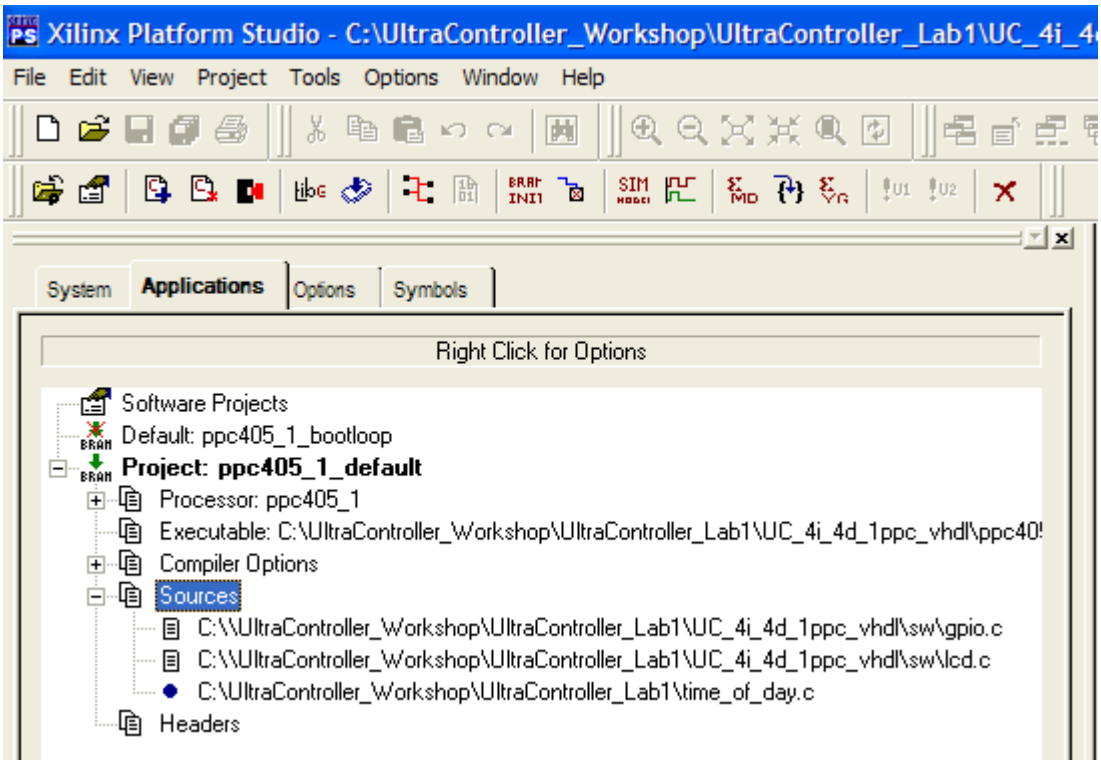

Figure 5 – XPS GUI Showing Addition of the time_of_day.c File

2.4 The Time-Of-Day Timer Application Software (time_of_day.c)

The following figure shows how the push button switches on the development board will be used to control the Time-Of-Day timer. On system power-on, the PUSH3 switch will be used to initiate the setting of the timer to the current time via PUSH1 and PUSH2 switches. The PUSH1 and PUSH2 are used to increment the Hours and Minutes respectively. After the initial setup, the PUSH3 switch serves no function until the board is reset.

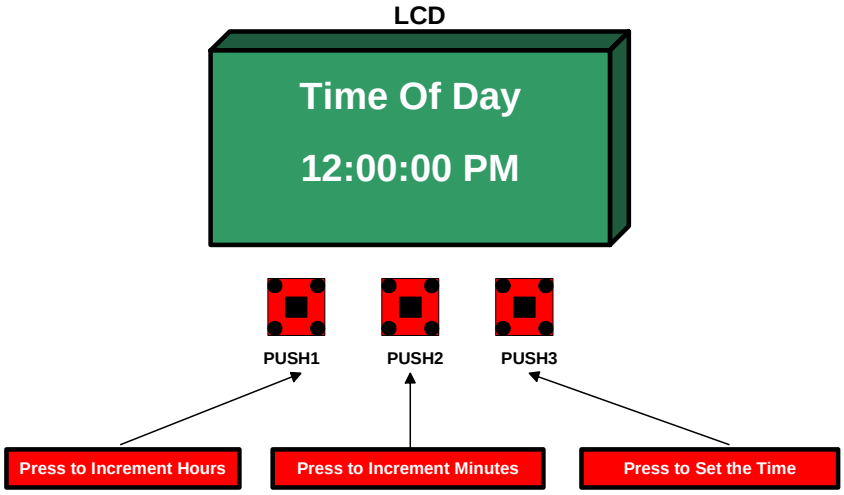

Figure 6 - Push Switch functions for the Timer Design

The application software for the Time-Of-Day timer design consists of two sections:

- Initialization On Reset
- Timer Function

The following sections describe how each part of the Time-Of-Day timer application software is implemented.

2.4.1 "Initialization On Reset" Flow Chart

The following figure shows the flow chart for the "Initialization On Reset" section of the design. On system reset, this function initializes the LCD panel and the global variables used in the program.

After this initialization step, this function enters a loop where messages are displayed on the LCD and the PUSH3 switch is read. The function remains in this loop until the PUSH3 switch is pressed. Upon activation of the PUSH3 switch, this function transfers the control of the program to the Timer Function, described in the following section.

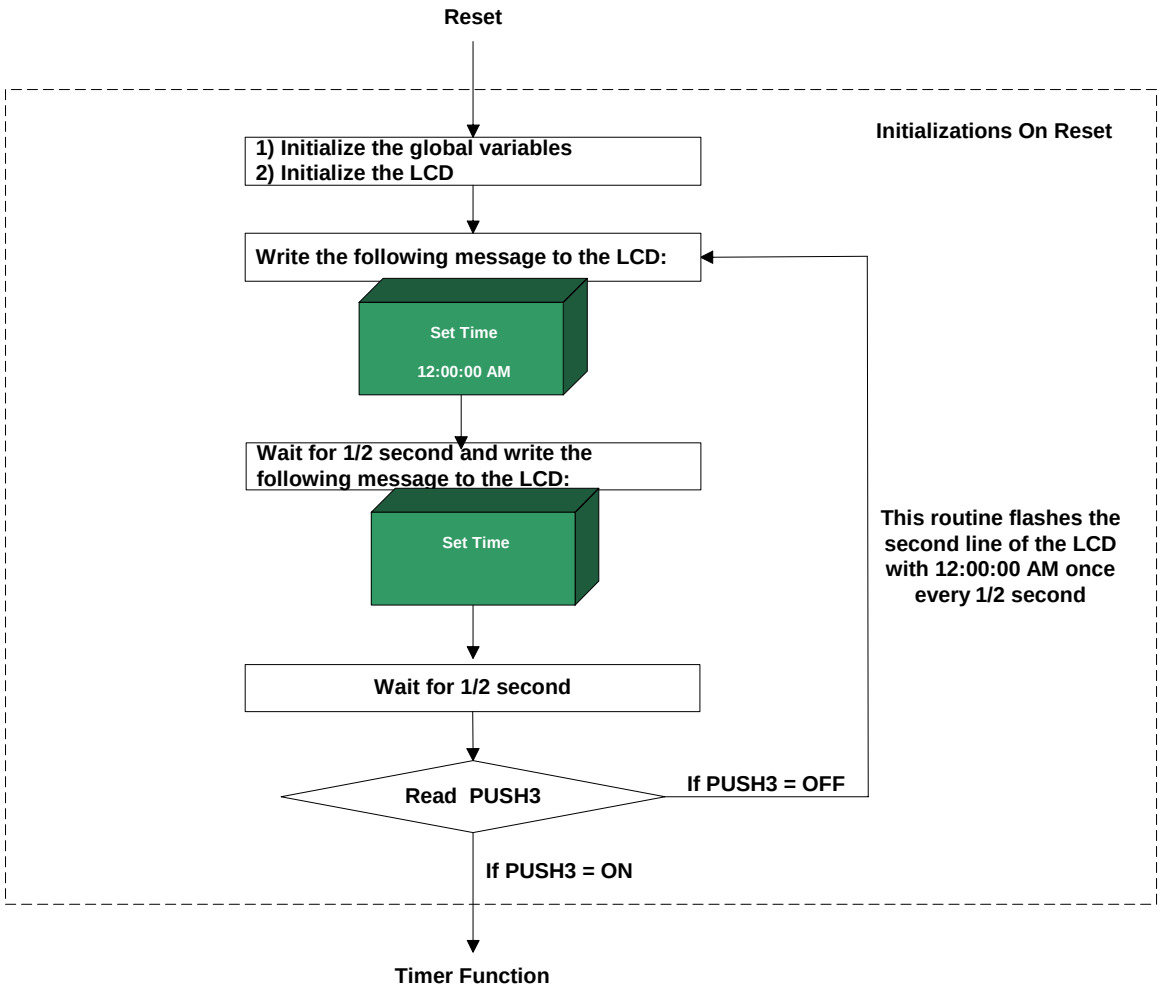

Figure 7 – Initialization On Reset Flow Chart

2.4.2 "Timer Function" Flow Chart

The following figure shows the flow chart for the "Timer Function" section of the design. After the initialization, the Timer Function turns ON an LED on the board (one at a time), reads PUSH1 and PUSH2 switches to see if the Hours and/or Minutes need to be incremented, displays the current time on the LCD, and it increments the Seconds every time the Timer Function is called.

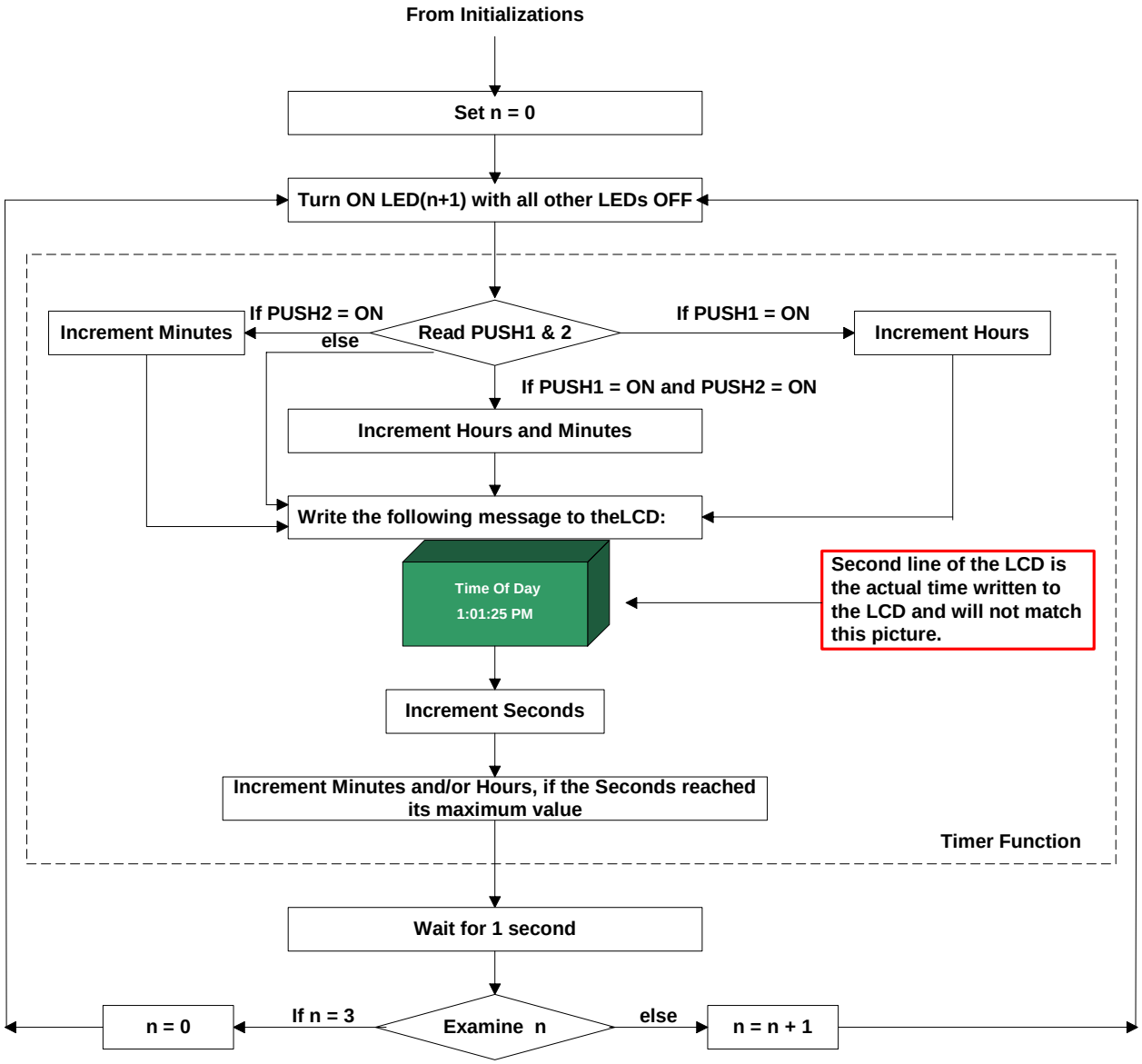

Figure 8 – Timer Function Flow Chart

2.4.3 Displaying the Current Time on the LCD

The following figure shows how the current time is displayed on the LCD panel. The first line of the LCD shows the " Time of Day " message and the second line shows the current time.

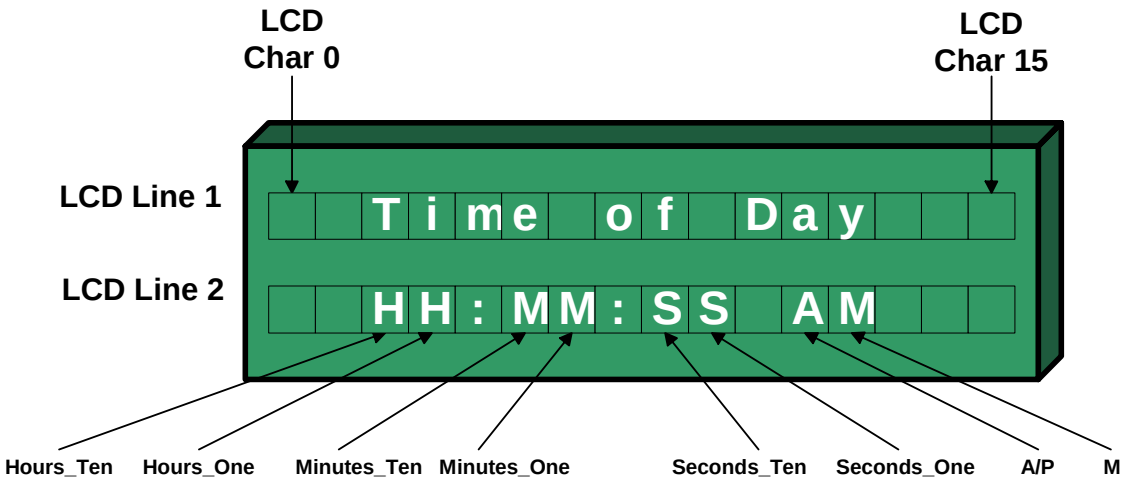

Figure 9 – LCD Panel Showing the Time

2.5 Compiling the C Source Files in EDK

From the XPS GUI select **Tools>Build All User Applications** to compile the C source files. The C source files are compiled and the executable.elf output file is generated.

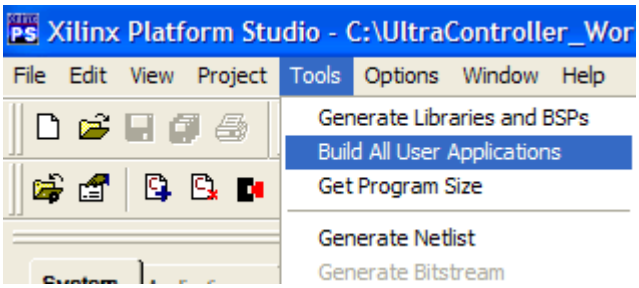

Figure 10 - Compiling Source Files

2.6 Generating a Netlist

From the XPS GUI, select **Tools> Generate Netlist** to generate netlist for the design (**DO NOT** export the project to the Project Navigator).

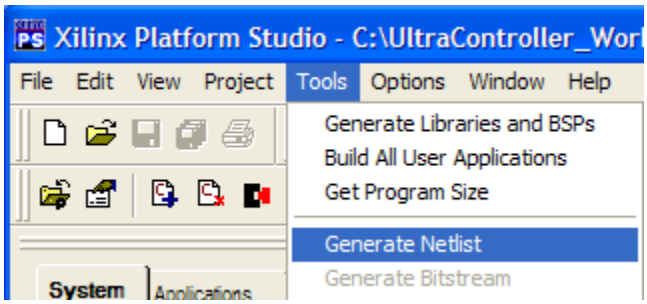

Figure 11 – Generating Netlist

2.7 Starting ISE and Opening the project in the Project Navigator

- Start Project Navigator

- Select **File>Open Project…**, browse to the **C:\UltraController_Workshop\UltraController_Lab1\UC_4i_4d_1ppc_vhdl\projnav** folder, select the **PPC_Lite.npl** file and then click **Open**. The project will open and you should see the following in the Project Navigator Sources Window.

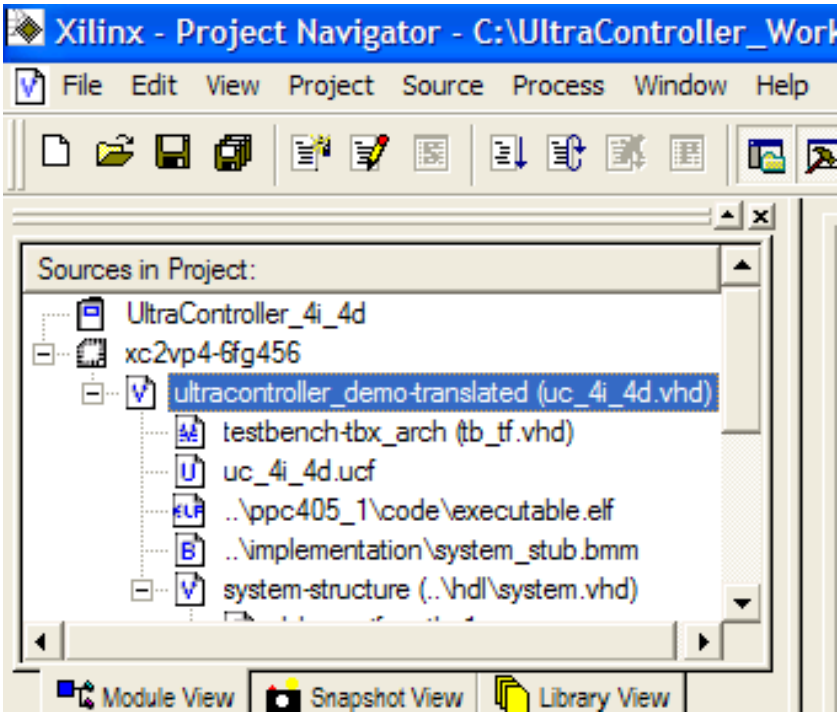

Figure 12 – Project Navigator Source Window

2.8 Adding UCF to the Project Navigator

In order to port the UltraController **uc_4i_4d_1ppc_vhdl** reference design to the Memec Design Virtex-II Pro 2VP4LC development board, the **uc_4i_4d.ucf** shown in the above figure must be replaced with the **time_of_day.ucf**. Please perform the following steps to replace the UCF.

- Remove the **uc_4i_4d.ucf** from the Project Navigator by right clicking on the file name and then selecting **Remove**.

- Add the **time_of_day.ucf** file to the project by selecting **Project>Add Source…** from the Project Navigator, browsing to the **C:\UltraController_Workshop\UltraController_Lab1** folder, selecting the **time_of_day.ucf** file, and then clicking **Open**. When asked for file association, select **UltraController_Demo**. Upon adding the new ucf to the project, the Project Navigator Sources Window should look as shown below.

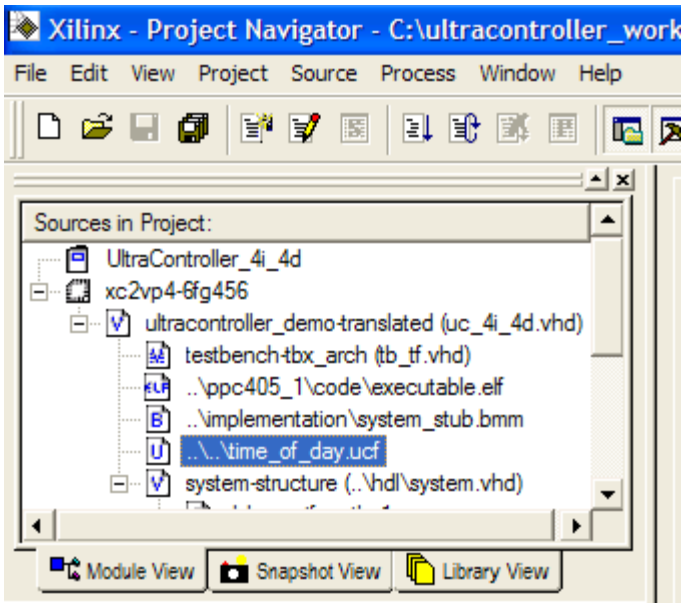

Figure 13 – Project Navigator Source Window

2.9 Generating a Bit File

In the Project Navigator, click on the top-level HDL in the Sources Window, right-click on the "**Generate Programming File**" in the Processes Window, and then select "**Rerun All**".

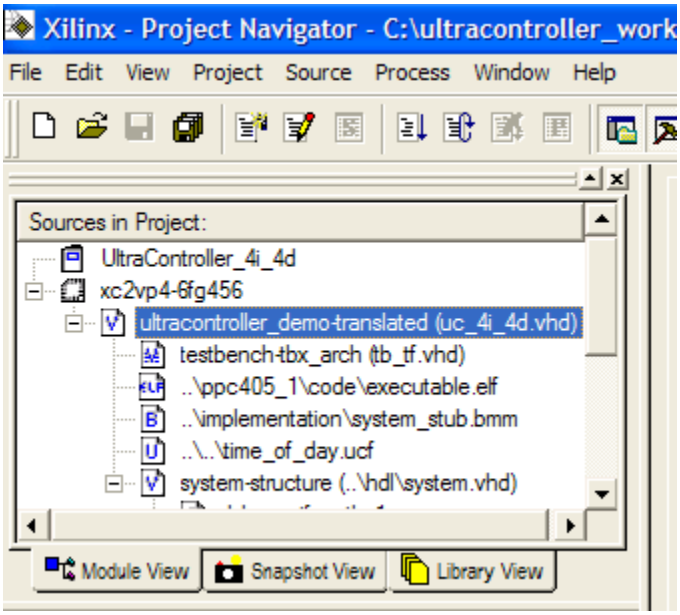

Figure 14 – Selecting the Top-level Design

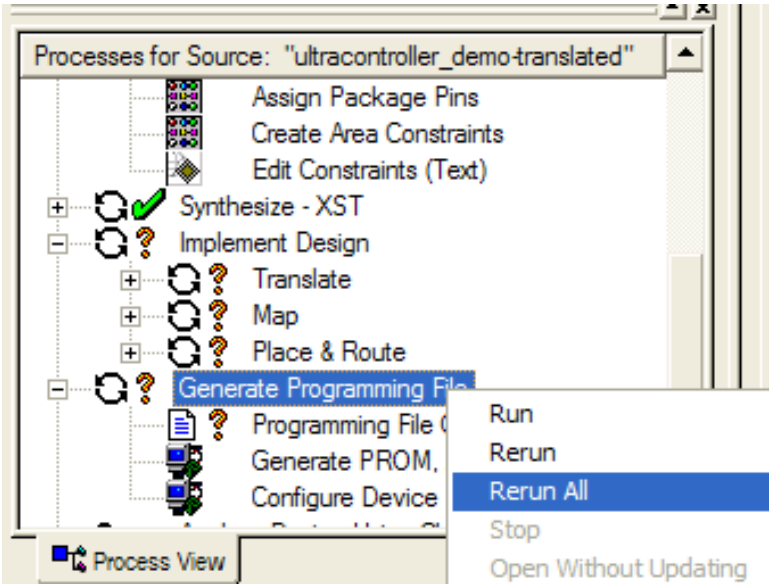

Figure 15 – Generating a Bit File

3 Setting Up the Development Board

Please install the following jumpers on the 2VP4LC development board:

1. Install jumpers on JP19 pins 1-2 and 3-4
2. Install jumpers on JP12 pins 3-4, and 5-6
3. Install a jumper on JP26 pins 1-2
4. Install a jumper on JP15 pins 2-3
5. Install a jumper on JP30 pins 1-2
6. Install a jumper on JP31 pins 1-2
7. Connect the PC3 JTAG cable to JP20 and the parallel port of the PC
8. **No other jumpers should be installed**

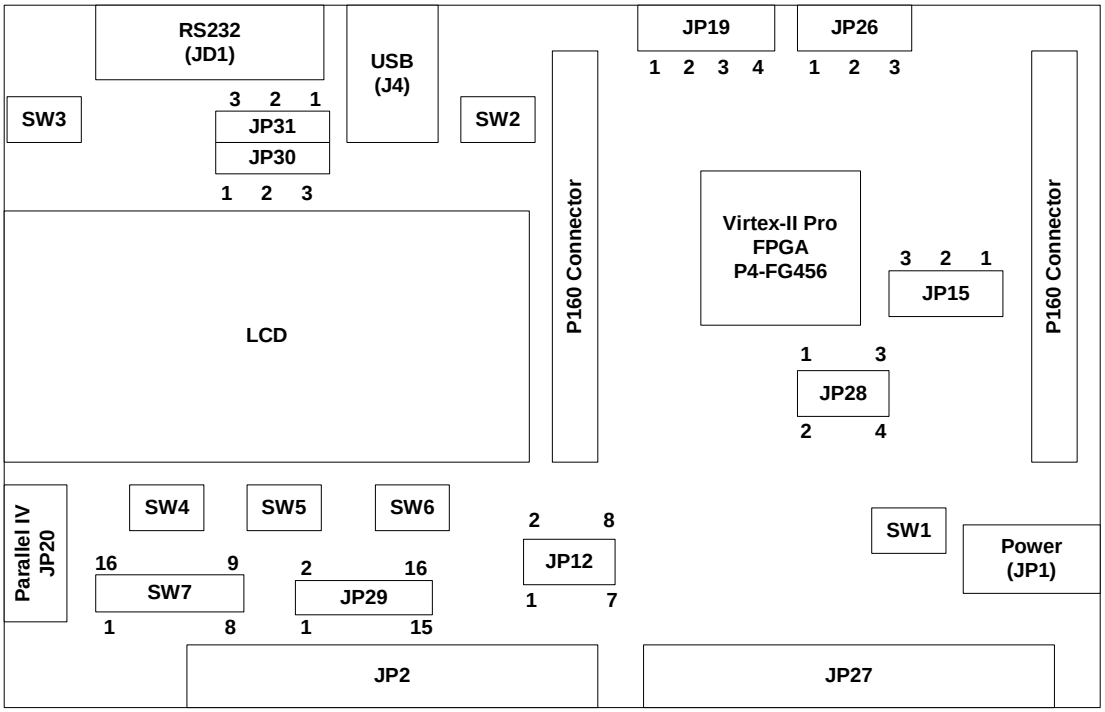

Figure 16 – 2VP4LC Development Board Component Side

4 Downloading the Bit File and Running the Design

In the Project Navigator, click on the top-level HDL in the Sources Window, right-click on the "**Configure Device (iMPACT)**" in the Processes Window, and then select "**Run**". You may have to expand the "Generate Programming File" to see the Configure Device (iMPACT) selection.

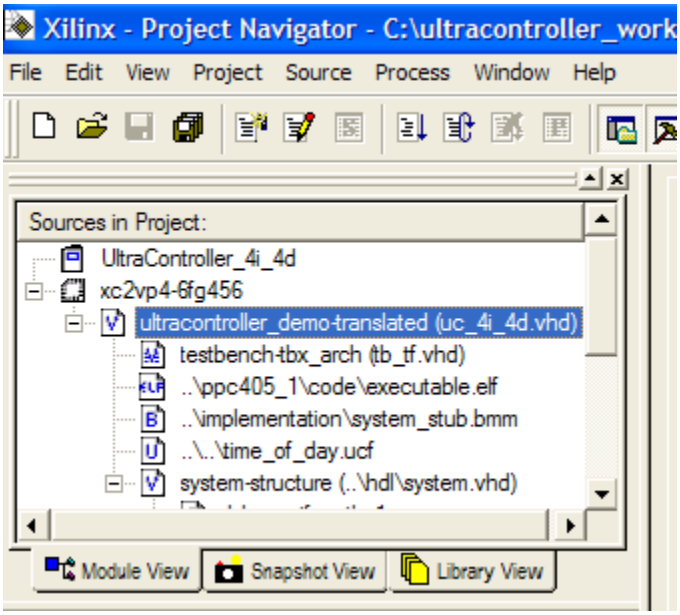

Figure 17 – Selecting the Top-level Design

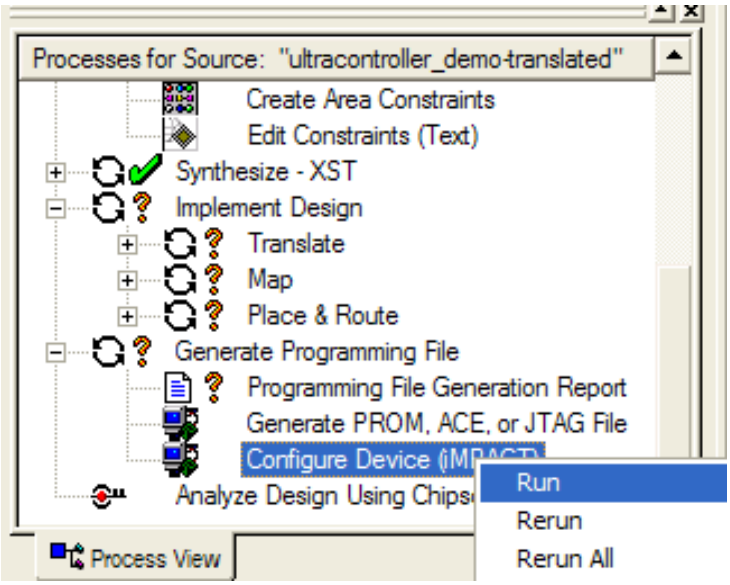

Figure 18 – Downloading the Bit File

In the iMPACT programming window skip the first device (PROM), browse to the **C:\UltraController_Workshop\UltraController_Lab1\UC_4i_4d_1ppc_vhdl\projnav** folder, select **ultracontroller_demo.bit**, and click **Open**. Click **OK** when the following dialog box appears.

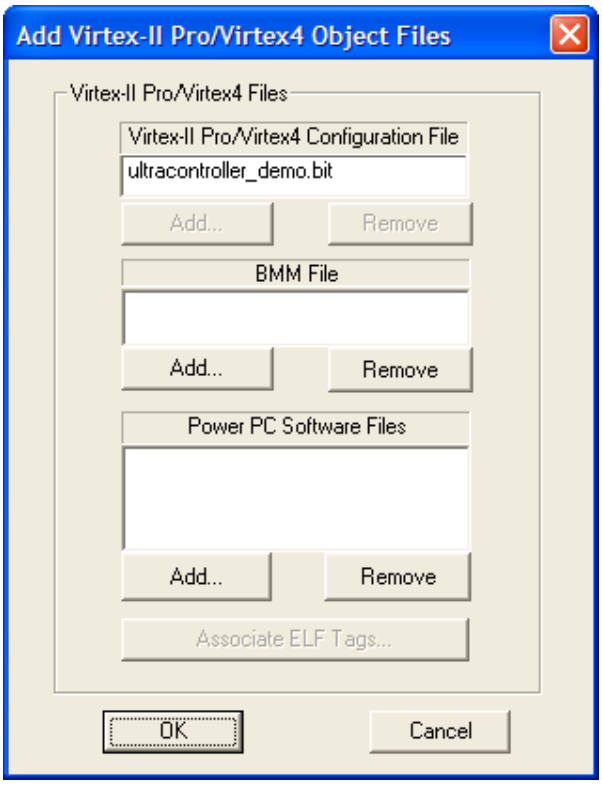

Figure 19 – Adding Software Dialog Box

Right-click on second device and select **Program…**.

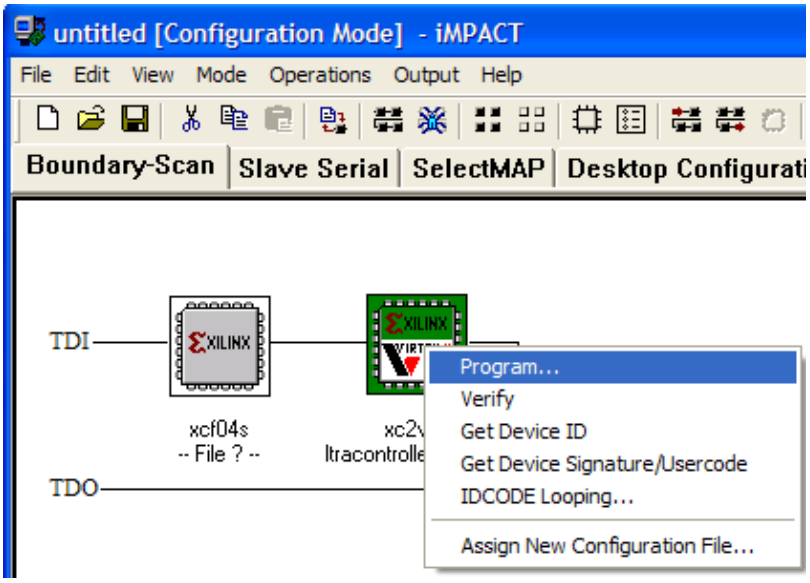

Figure 20 – iMPACT Programming Window

After downloading the bit file, use the **PUSH1**, **PUSH2**, and **PUSH3** push switches to set the time and verify the operation of the Time-Of-Day Timer by performing the following steps:

- After seeing the following message on the LCD panel

  Set Time
  12:00:00 AM

  press and hold **PUSH3** until you see the following message on the LCD panel

  Time of Day
  12:00:00 AM

  Release PUSH3.

- Use **PUSH1** to increment the Hours and **PUSH2** to increment the Minutes to set the time to the current time.

Are the 4 LEDs on the board flashing one at a time at a rate of one second? If not, proceed to the next section to modify the **time_of_day.c** source file and enable the LED flashing function.

4.1 Modifying the time_of_day.c Source File

Open the time_of_day.c source file in the XPS GUI by double-clicking on the file name and go to the code segment as shown in the following figure.

```
191         initial_value = 0x8;
192
193         while (1) {
194             for (loop_count = 0; loop_count < 4; loop_count++) {
195                 temp_data = (initial_value >> loop_count);
196 //              GPIO_writeLED(temp_data);
197
198                 timer_update();
199                 sleep(1);
200             }
201             initial_value = 0x8;
202         }
203 }
```

Figure 21 – Time_of_day.c Source File

As shown in the above figure, the write to the GPIO output port to flash the LEDs is commented out. Remove the comment from line 196, save and close the **time_of_day.c** source file.

4.1.1 Compiling the Modified time_of_day.c Source File

From the XPS GUI compile the source files with the change made to the time_of_day.c source file as shown in the following figure. This will update the executable.elf file in the project navigator.

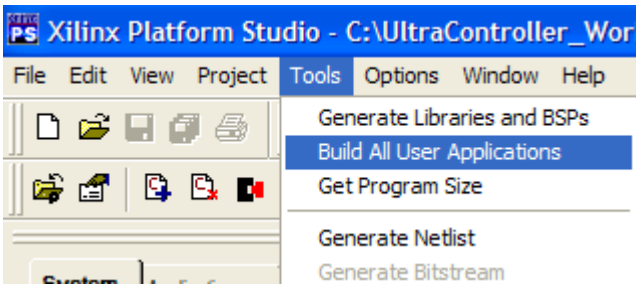

Figure 22 - Compiling Source Files

4.1.2 Generating a Bit File

Since the executable.elf has been updated, a bit file needs to be generated that reflects the changes. So, generate a bit file in the project navigator by double-clicking on **Generate Programming File** and download it to the board. Since, only software portion of the design has been modified, project navigator will not run the place and route tool for generating the new bit file and it will only run the bitgen utility. This is extremely useful for software debugging and saving place and route time.

Since iMPACT is already open, right-click on the FPGA and select **Assign New Configuration File…** to assign the updated bit file to the FPGA. Once the bit file is assigned to the FPGA, configure the FPGA with the new bit file.

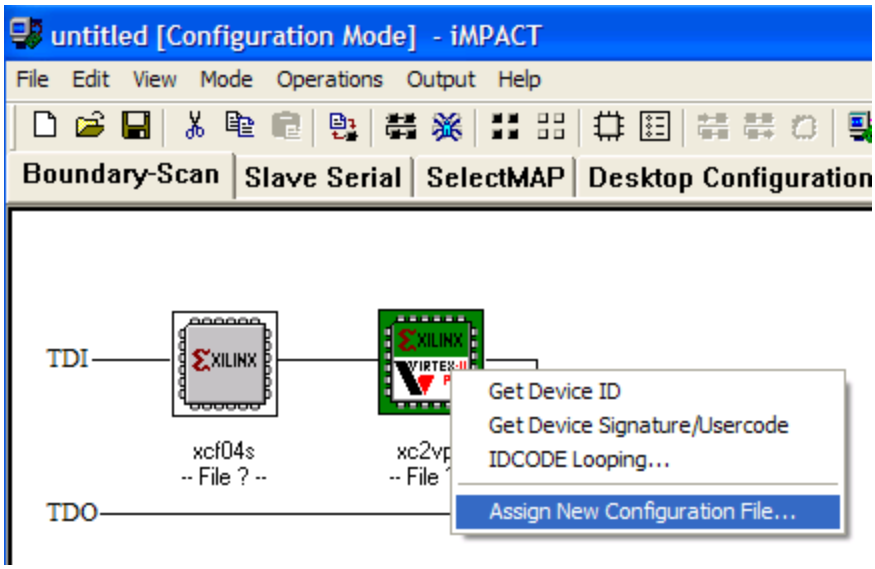

Figure 23 – iMPACT Programming Window

After downloading the bit file and pressing the PUSH3 switch, you should see the LEDs flashing.

5 Simulating the Time-Of-Day Timer Design

In this section of the lab, the ModelSim simulator will be used to perform HDL simulation of the Time-Of-Day timer design.

5.1 Adding the sim_sleep.c Source File to the Project

The **sim_sleep.c** file needs to be added to the EDK project for simulation purposes only. Adding this source file removes the delays in the software and allows the simulation to take less time to complete (delays are needed in the software for LCD interface and the timer function). Perform the following steps to add the sim_sleep.c source file to the project:

- Right-click on "**Sources**" in the XPS GUI and then select "**Add Files**". Browse to the **C:\UltraController_Workshop\UltraController_Lab1\UC_4i_4d_1ppc_vhdl\sw** folder, select **sim_sleep.c** file and then click **Open**.

Once this file is added to the project, the XPS GUI will look as shown in the following figure.

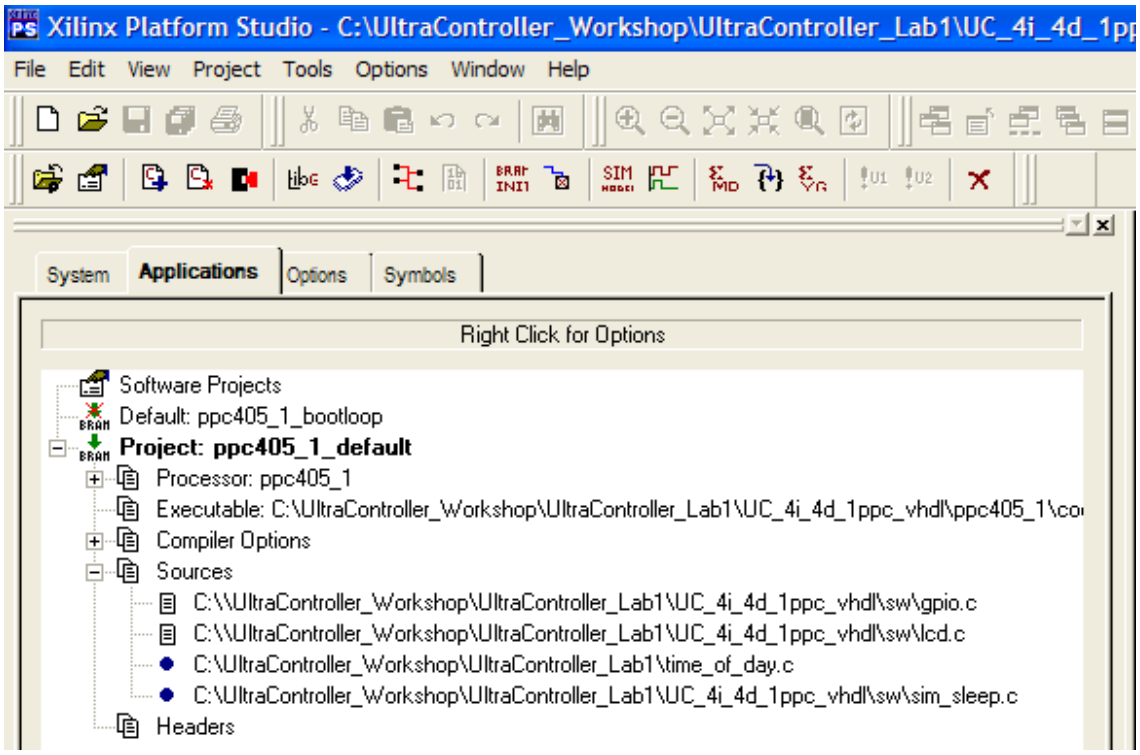

Figure 24 – Adding Sources to the Project

5.2 Compiling the Source Files in EDK

From the XPS GUI select **Tools>Build All User Applications** to compile the C source files. The C source files are compiled and the executable.elf output file is generated.

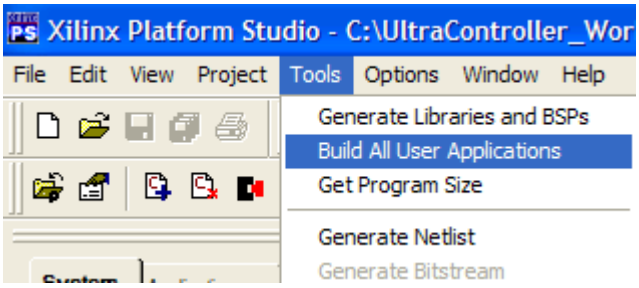

Figure 25 – Compiling Source Files

5.3 Modifying the GPIO Input/Output Port Connections for Simulation

In the Project Navigator, double-click on the **ultracontroller_demo** to open the **uc_4i_4d.vhd** file. Go to the **BEGIN** section of this HDL file and edit the gpio_in/gpio_out ports to match the code segment shown below. This change is needed for simulation only, and it forces all the GPIO output bits to a known state. With the un-modified code, since the upper 16 bits of the GPIO (bits 0:15) are not used for this reference design, they will become un-defined in simulation. Save and close the **uc_4i_4d.vhd** file.

```
109
110   BEGIN
111
112      -- reset logic
113      sys_rst <= (NOT nsys_rst);
114
115      -- strip off required number of gpio_out_s nets
116      gpio_out(0 TC 31) <= gpio_out_s(0 TC 31);
117
118      --strip off required number of gpio_in nets
119      gpio_in_s(0 TC 31) <= gpio_in(0 TC 31);
120
```

Figure 26 – UltraController Top-level HDL

5.4 Changing the Maximum Simulation Time in the Testbench File

In the Project Navigator, double-click on the **testbench** to open the **tb_tf.vhd** file. Go to the following section and change the MAX_SIM_TIME to **1000us** (the initial value is set to **100us**). Save and close the **tb_tf.vhd** file.

```
10  entity testbench is
11  end testbench;
12
13  architecture TBX_ARCH of testbench is
14
15  -- ********** CONSTANT DECLARATIONS *************
16       constant HALFCLKPERIOD   :    time := 2.857 nS;
17       constant MAX_SIM_TIME    :    time := 1000 uS;
18  -- *********** COMPONENT DECLARATIONS ***********
19
```

Figure 27 – UltraController Testbench

5.5 Running the ModelSim Simulator from the Project Navigator

Click on the **testbench** in the Project Navigator Sources Window, right-click on the "**Simulate Post-Place & Route VHDL Model**" in the Processes Window and then select **Run** to invoke the ModelSim simulator as shown in the following figures.

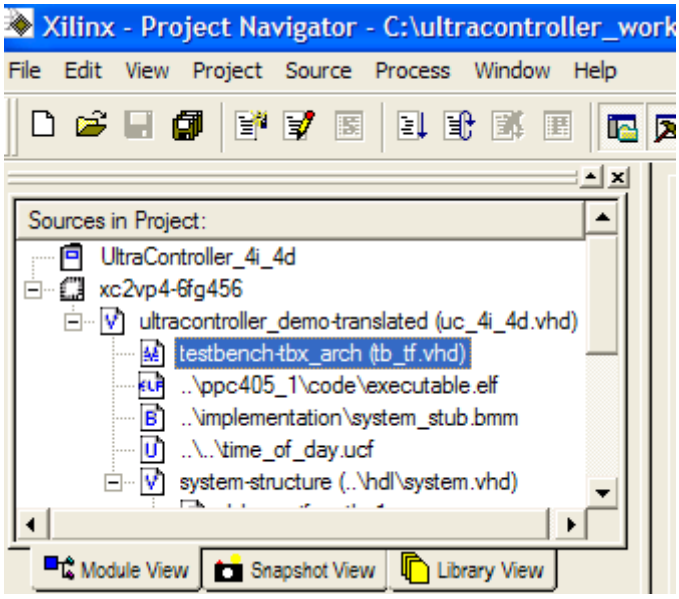

Figure 28 – Selecting the Testbench

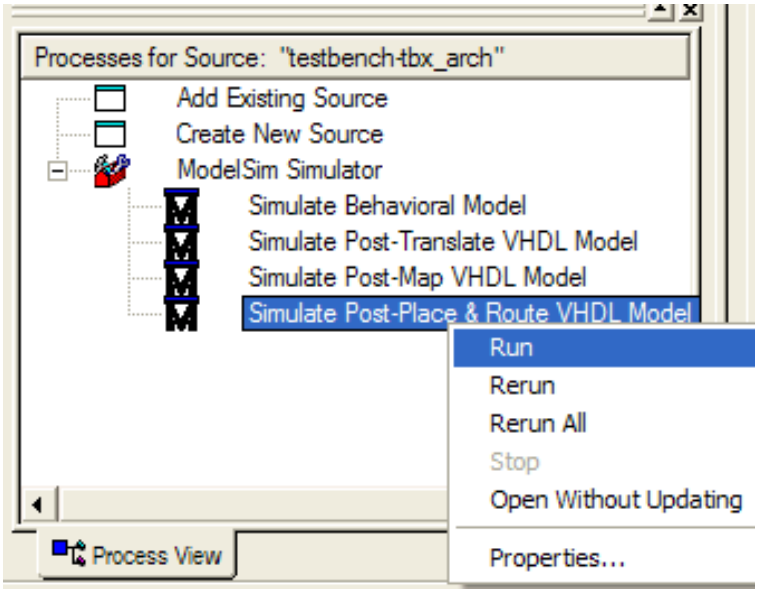

Figure 29 – Invoking the ModelSim Simulator

In the ModelSim Command Window, enter "**do wave.do**" and then run the simulation for **600us** as shown in the following figure. On the workshop laptops, it will take about **10** minutes to run the simulation for **600us**.

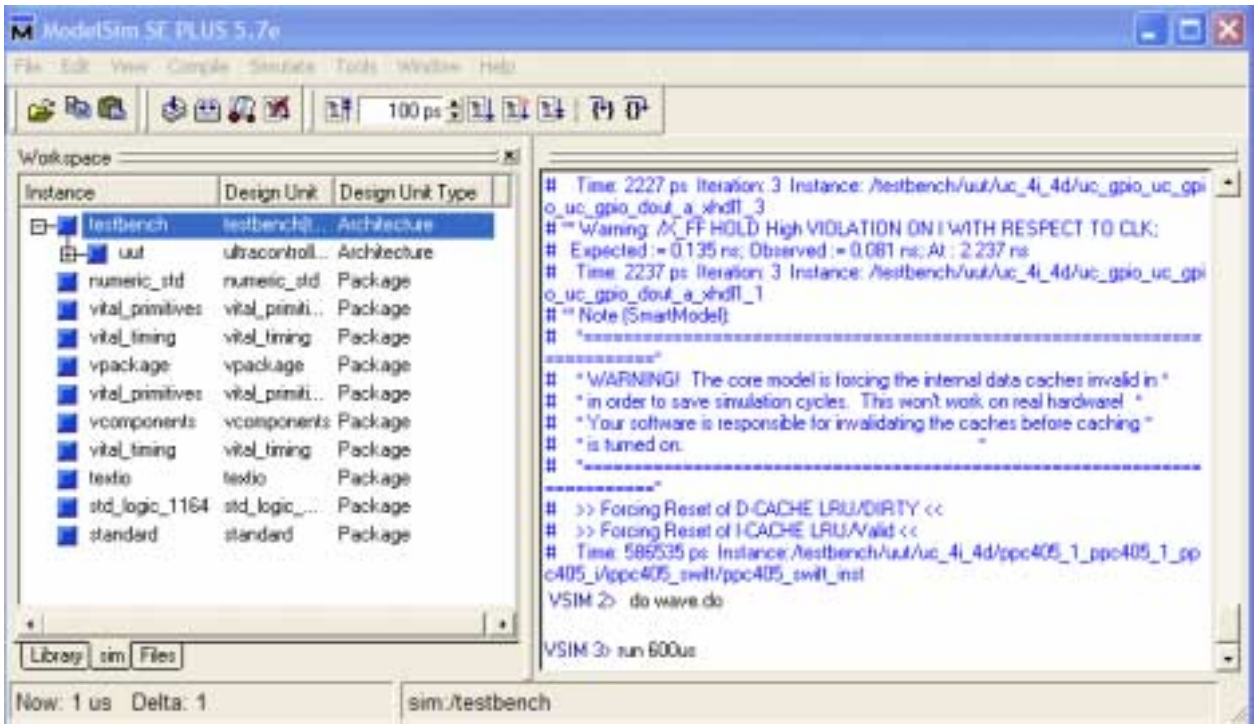

Figure 30 – ModelSim Command Window

5.6 Viewing Waveforms

The following figure shows the LCD initializations on system reset. Upon deactivation of the reset signal, a sequence of 8-bit data is written to the LCD panel to initialize it. The **LCD_Init()** function call in the time_of_day.c source file performs the LCD initialization.

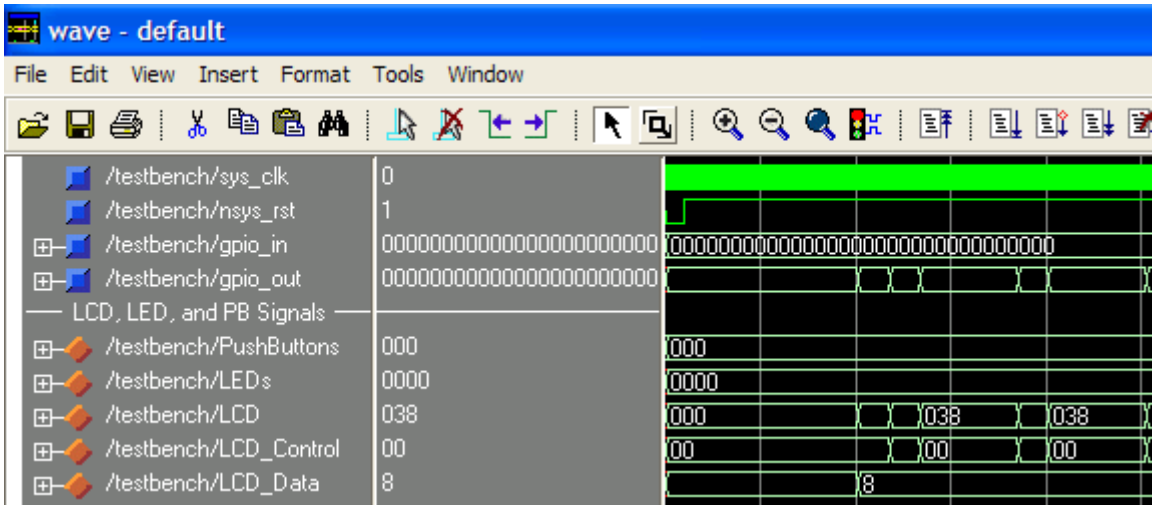

Figure 31 – LCD initializations on Reset

The following figure shows the “Set Time” message being displayed on the LCD panel. Move forward (to about **70us**) in the waveform window to view this message.

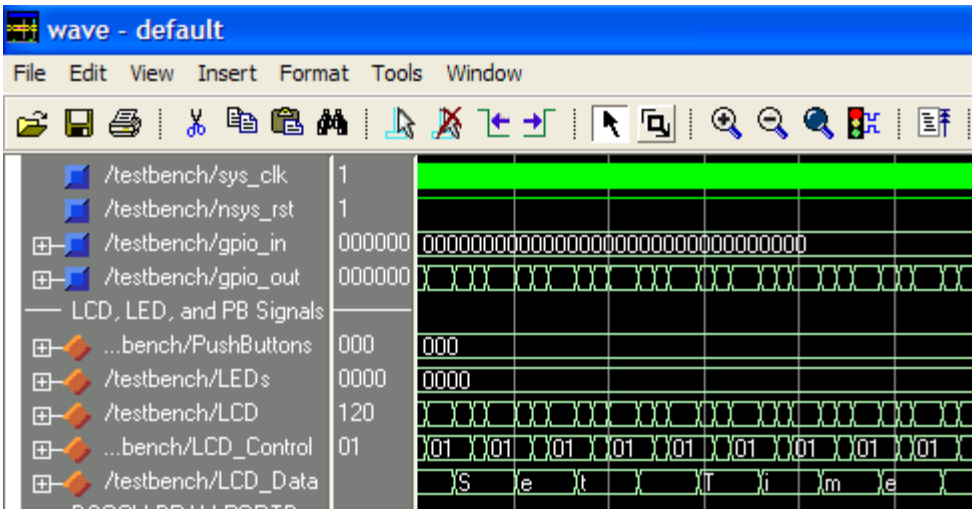

Figure 32 – Writing "Set Time" to line 1 of the LCD

The following figure shows the "12:00:00 AM" message being displayed on the LCD panel. Move forward (to about **90us**) in the waveform window to view this message.

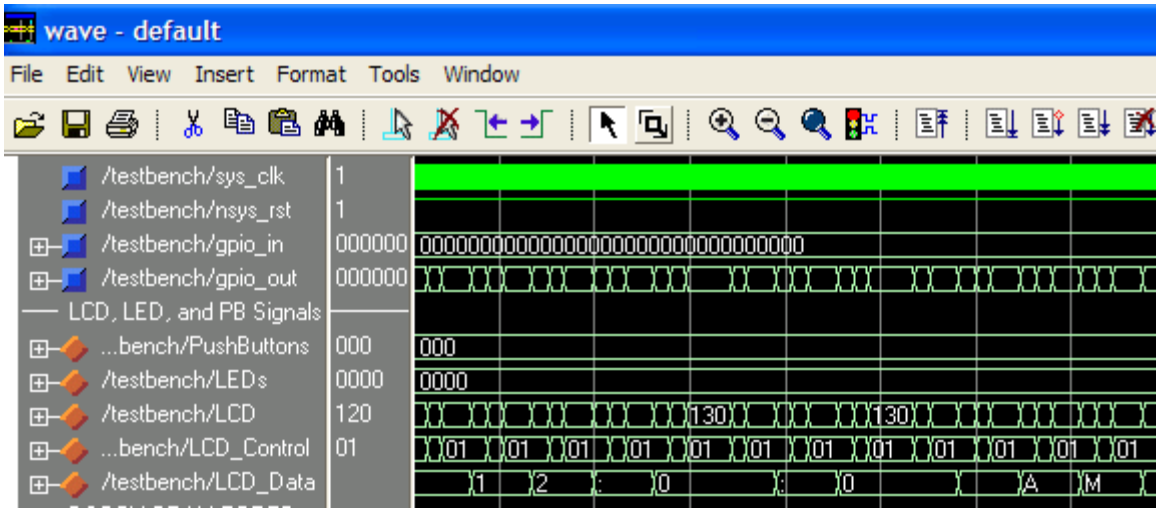

Figure 33 - Writing "12:00:00 AM" to line 2 of the LCD

The following figure shows LED1 being turned ON with all the other LEDs OFF (LEDs = 0111). Move forward (to about **330us**) in the waveform window to view this message.

When testing the design on the board, PUSH3 had to be pressed in order to advance in the code and see the LEDs flashing. The testbench used for simulation sets all GPIO input bits to zero activating all push switches; hence the code will advance automatically.

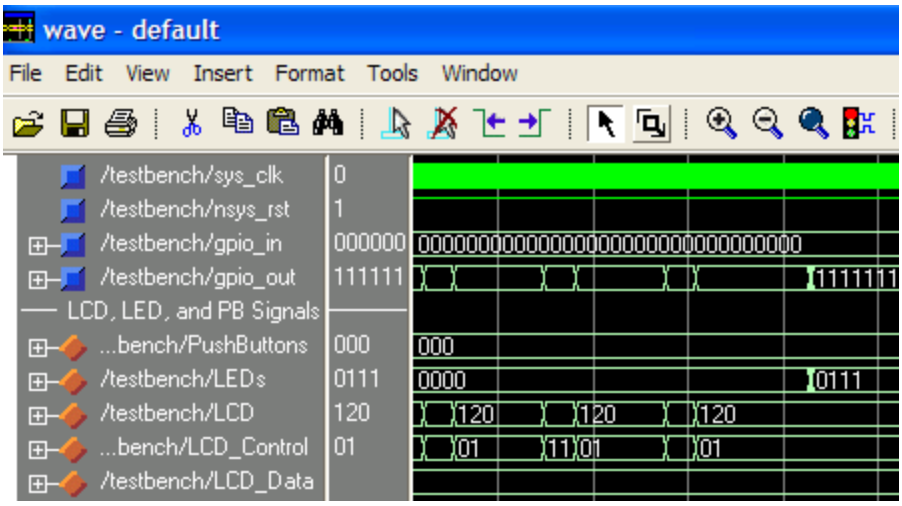

Figure 34 – The LED1 is turned ON With All Others OFF

The following figure shows the "Time of Day" message being displayed on the LCD panel. Move forward (to about **385us**) in the waveform window to view this message.

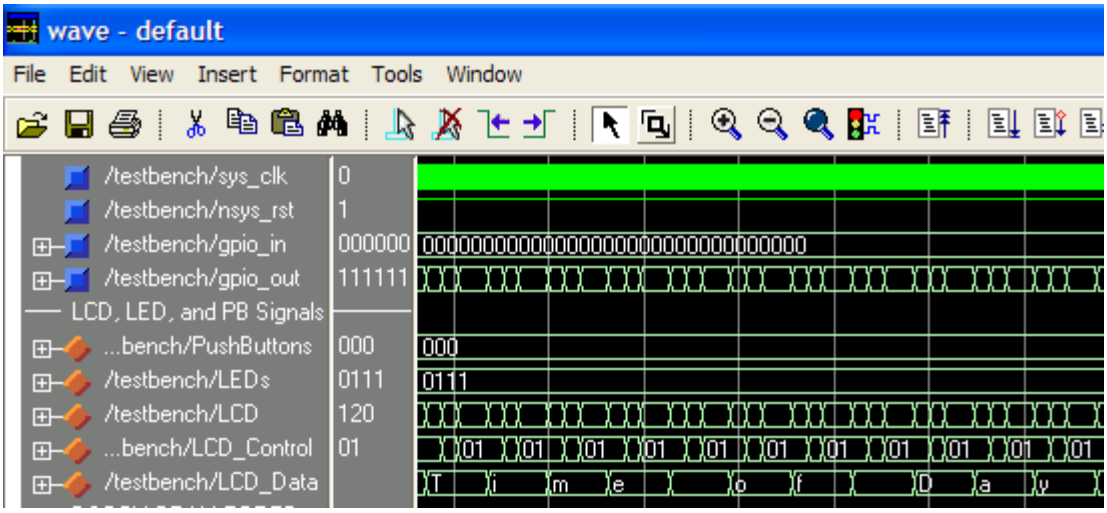

Figure 35 - Writing "Time of Day" to line 1 of the LCD

The following figure shows the "01:01:00 PM" message being displayed on the LCD panel. This is after the timer Hours and Minutes has been incremented by one. Move forward (to about **407us**) in the waveform window to view this message.

The timer increments the Hours and Minutes, because the GPIO input port is set to all zeros in the testbench. Hence, PUSH1, PUSH2, and PUSH3 are active allowing the simulator to step through the code and increment the Hours and Minutes.

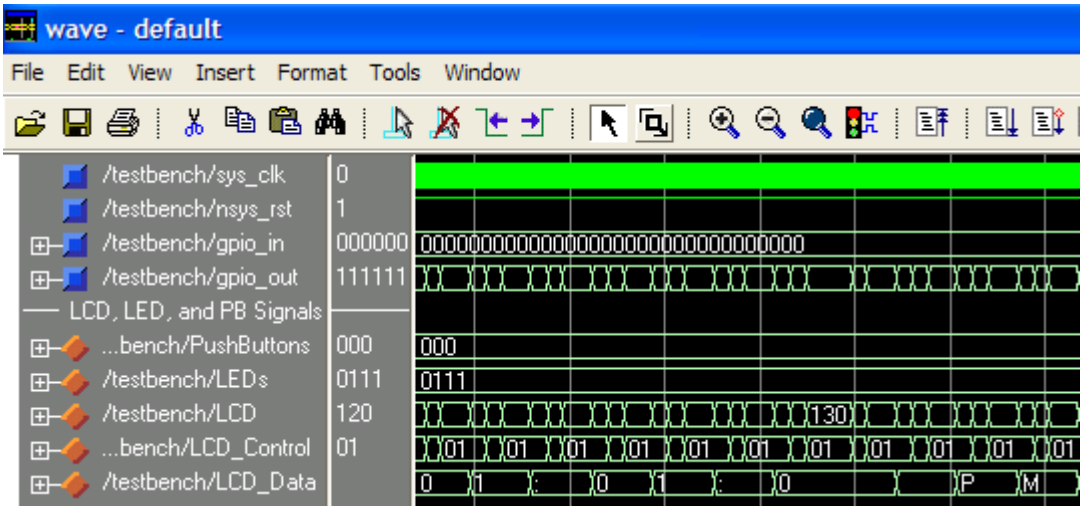

Figure 36 – Writing the Current Time to line 2 of the LCD

The following figure shows the LEDs being turned ON one at a time (you will see this, if you let the simulator run for about **900us**).

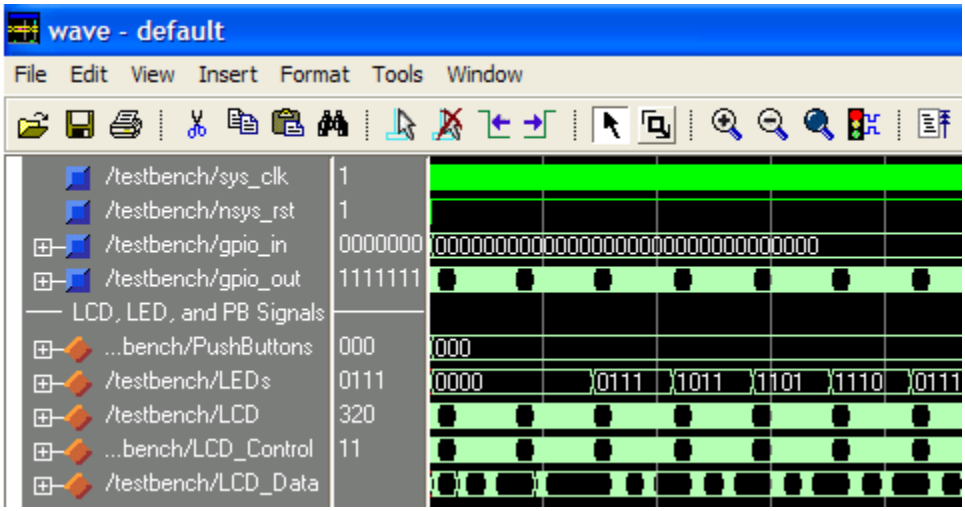

Figure 37 – LEDs Being Turned ON in Sequence

Congratulations, you have successfully simulated a processor-based design running software using an HDL simulator.