<u>Phy 335, Unit 7</u>
<u>Using the Digital Logic Development Kit</u>

The work for Unit 7 uses a digital logic development board[1] which has a programmable logic chip and a variety of input and output (I/O) choices including general purpose digital I/O lines, switches, LED's, and advanced functionality including audio and video outputs and a secure digital (SD) memory card slot. A photograph of the development board is shown in Fig. 1. The digital functionality for this course has been programmed into the boards. *Students are not required to do the programming*, but use the existing functionality to test analog-to-digital interfacing.

<u>Setting up the system</u>

<u>Step 1:</u> The development kit is electrically attached to the lab breadboard by connecting the expansion header GPIO 0 (aka JP1, See Fig. 1) on the development kit to the standard lab breadboard using a 40-pin ribbon cable with the mass termination header connected to the breadboard (See Fig. 2). The connectors have a polarity, so make sure they are connected in the proper orientation.

The pin connections used in this lab are listed in Table 1. In particular note that on the connector

Pin 12 = GND (from dev kit)

This ground must be connected to the bread board (and therefore the power supply) ground. The corresponding position on the mass termination header is marked in black.

Signal Name	**Direction**	**Pin Connections (on GPIO0)**
Data In (to dev kit)	INPUT	27, 25, 23, 21, 19, 17, 15, 13 (LSB, pin 13)
ADC interrupt	INPUT	31
Data Out (from dev kit)	OUTPUT	28, 26, 24, 22, 20, 18, 16, 14 (LSB, pin 14)
Data Valid	OUTPUT	32
Store Data (to scope)	OUTPUT	34
Read Data (to scope)	OUTPUT	36

Table 1: General I/O connections for Phy335. The even pins are on one side of the connector and the odd pins are on the other side of the connector. The pin marked with a triangle on the connector is pin 1

<u>Step 2:</u> Once the kit and breadboard are connected, set all of the slider switches SW0-SW9 on the dev kit (See Fig. 1) down and plug in the development kit to the AC power from the silver power supply using the AC adapter in the dev kit box, and push the red

[1] The Altera Cyclone II FPGA Starter Development board, . See http://www.altera.com/products/devkits/altera/kit-cyc2-2C20N.html and the course web page.

button to turn the dev kit on. *When the device is plugged in with the switches down, the 7-segment LED displays (aka hex displays) should read "P335"*

Additional Information: Different parts of the unit require different logic functionality. The choice of a specific function is determined by setting a toggle switch. The switch to be turned on for the various parts of unit 7 is

Unit 7, part 2:	SW0 is on, all others are off
Unit 7, part 3:	SW0 and SW9 are on, all others are off
Unit 7, part 4:	SW1 and SW9 are on, all others are off

Using the development kit system for Unit 7, part 2

Set up the board as described above. Set SW0 to on (up), and all others to off. The LED hex display should read "00", and the red LED above the switch SW0 should turn on.

In this mode, the dev kit puts an *8-bit number on the DATA OUT pins* listed in Table 1. The 8-bit number appears on even numbered pins from 14-28. The least significant bit (DO0) is pin 14, and the most significant (DO7) is pin 28. Use these as the data inputs to the DAC0802 chip in the lab instructions. *The 8-bit value is simultaneously shown on two of the hex displays (in hexadecimal, base 16).*

The four push buttons (labeled KEY0 – KEY3 on the development kit board) have the following functions:

Button	**Function**
KEY0	Set output value to 0
KEY1	Add 1 to the output value (counter)
KEY2	Add 16 (= 10_{16}) to the output value
KEY3	Multiply the output value by 2.

Using the system for Unit 7, part 3

Set up the board as described above. Set SW0 and SW9 to on (up) and all others to off. The red LED above the switch SW0 should turn on. This mode is the same as the previous mode, except that in addition to displaying the 8-bit output number put on the rightmost two hex displays, the dev kit is programmed to read the 8-bit number on the DATA IN bus (Table 1) and display it on the left two hex LEDs.

We will use this to read back output from an ADC on the standard bread board

The ADC needs an analog voltage for an input. In general, this could come from any source, but in lab we can conveniently use the analog output from the DAC used in part 2. The switches KEY0 through KEY3 have the same functionality in this part as they did in part 2.

<u>Using the system for Unit 7, part 4</u>

Connect the dev kit and breadboard as described above. Set SW1 and SW9 to on (up) and all others to off. The red LED above the switch SW1 should turn on.

For part 4, we need to use the data store and data read functionality provided by a FIFO. This is programmed into the dev kit already. Leave the connections from part 2 and part 3 in place, and *add the ADC interrupt output to pin 31 of GPIO 0.* ***Be careful!*** *If you connect to the wrong pin, you may fry the ADC interrupt output*. This signal indicates to the dev kit that ADC data is ready to be stored.

Button	**Function**
KEY0	Reset, and then continuously show the current DAC input and ADC output on the hex LEDs
KEY1	<u>Store data</u>: 1024 consecutive conversions by the ADC are stored into the FIFO. The output of the FIFO does not change.
KEY2	<u>Read data:</u>The conversions stored after pushing KEY1 are driven onto the DATA OUT bus (connected to the DAC).
KEY3	Does nothing…

This set up allows one to play back (once) data stored from an earlier conversion. This method is a typical means to temporarily store results before reading to a computer. A scope can be triggered during “Read data” mode using pin 36.

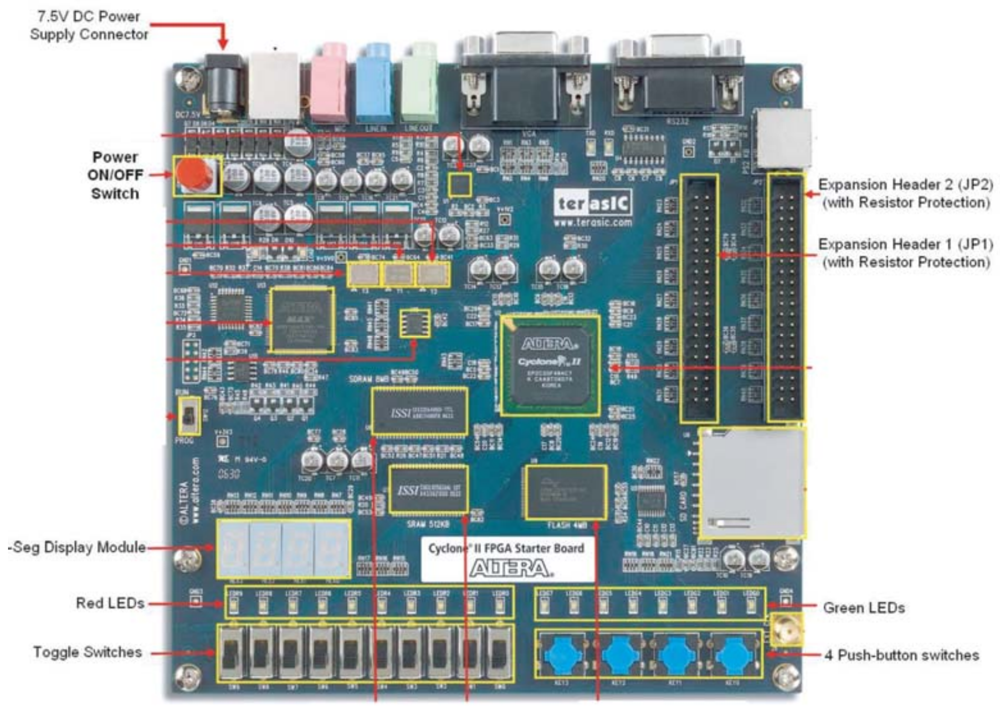
7.5V DC Power Supply Connector
Power ON/OFF Switch
Expansion Header 2 (JP2) (with Resistor Protection)
Expansion Header 1 (JP1) (with Resistor Protection)
-Seg Display Module
Red LEDs
Toggle Switches
Green LEDs
4 Push-button switches
Cyclone II FPGA Starter Board
ALTERA
terasIC

Fig. 1: The Programmable Logic Development Kit

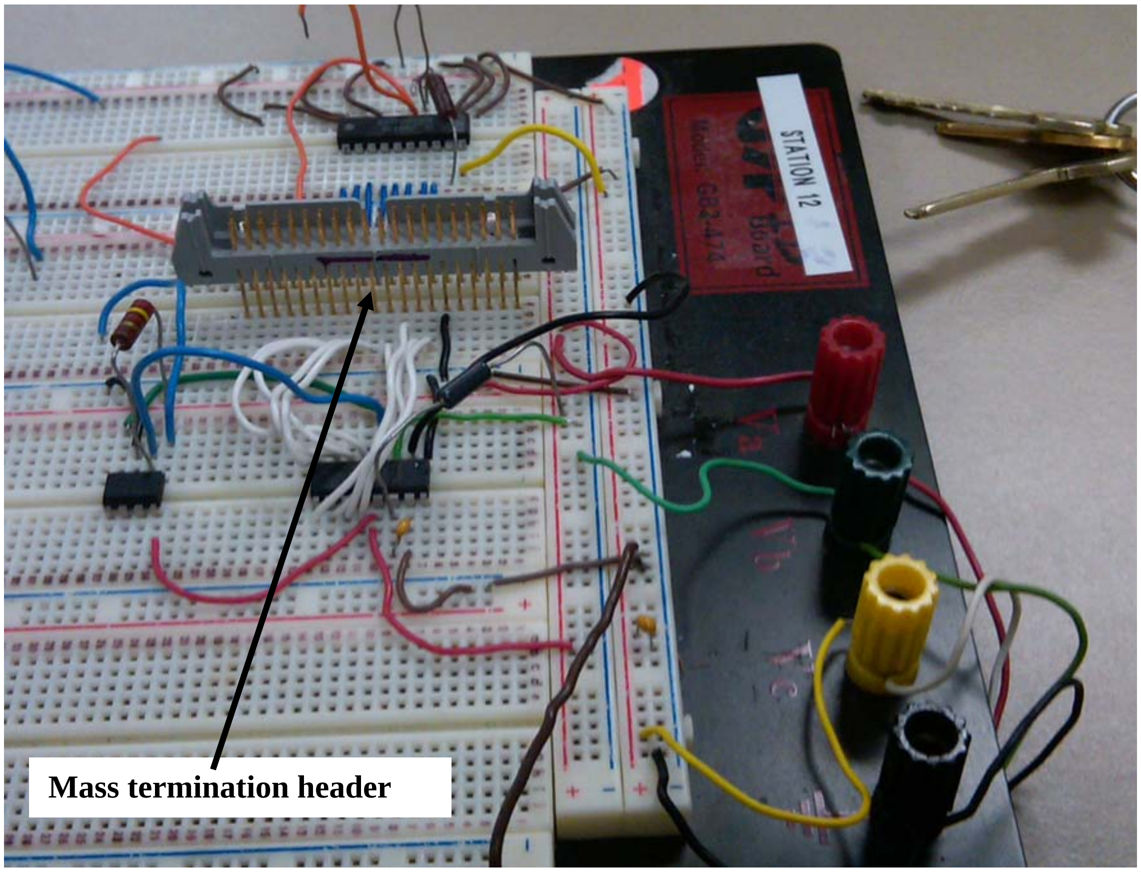

Fig. 2: The mass termination header connected to the lab breadboard