

Department of Electrical and Computer Engineering
University of Wisconsin-Madison

ECE 315
Introductory Microprocessor
Laboratory

ADuC7026-Cyclone
Board Manual

Michael G. Morrow

Last updated 1/14/2008 1:01 PM

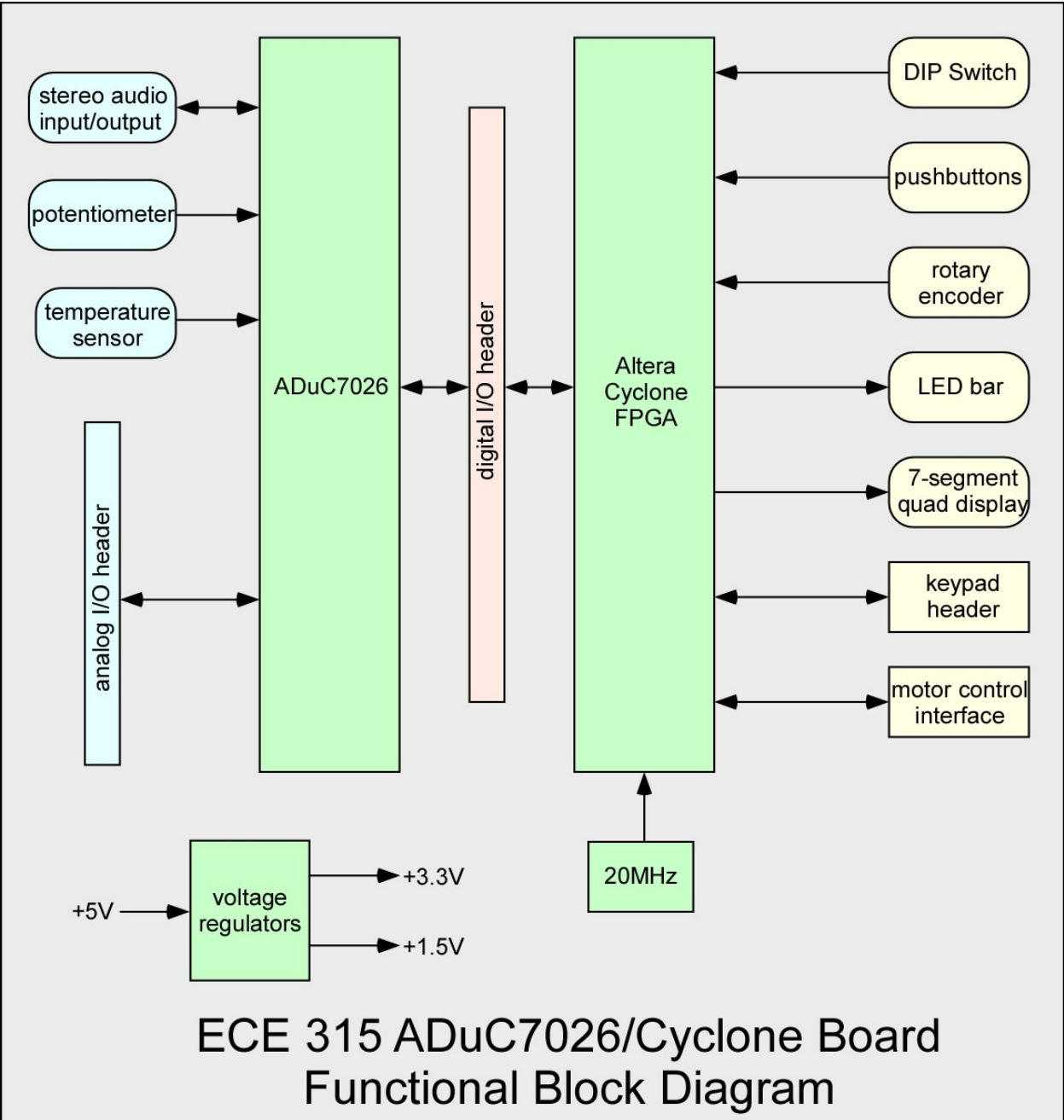
Table of Contents

Board Overview	1
1.1 Board Organization	1
1.2 Board Locations	2
1.3 Cautions	2
2 Using the ARM7-FPGA Board	3
2.1 Bench Set-Up	3
2.2 Using Altera Quartus	3
2.3 Using Keil μ Vision3	3
2.3.1 Keil μ Vision3 Project Files	3
2.3.2 Renaming Keil μ Vision3 Projects	4
3 Input/Output Devices	4
3.1 Analog Input/Output	4
3.1.1 Potentiometer Input	4
3.1.2 Temperature Sensor	4
3.1.3 Stereo Audio Interface	4
3.2 Digital Input/Output	4
3.2.1 Pushbuttons	5
3.2.2 DIP Switch	5
3.2.3 Rotary Encoder	5
3.2.4 Keypad Interface	5
3.2.5 LED Bar Display	5
3.2.6 Four-Digit LED 7-segment Display	6
3.2.7 Serial Communication Interfaces	6
3.2.8 Motor Control Interface	6
4 FPGA Pin Naming	7

Board Overview

1.1 Board Organization

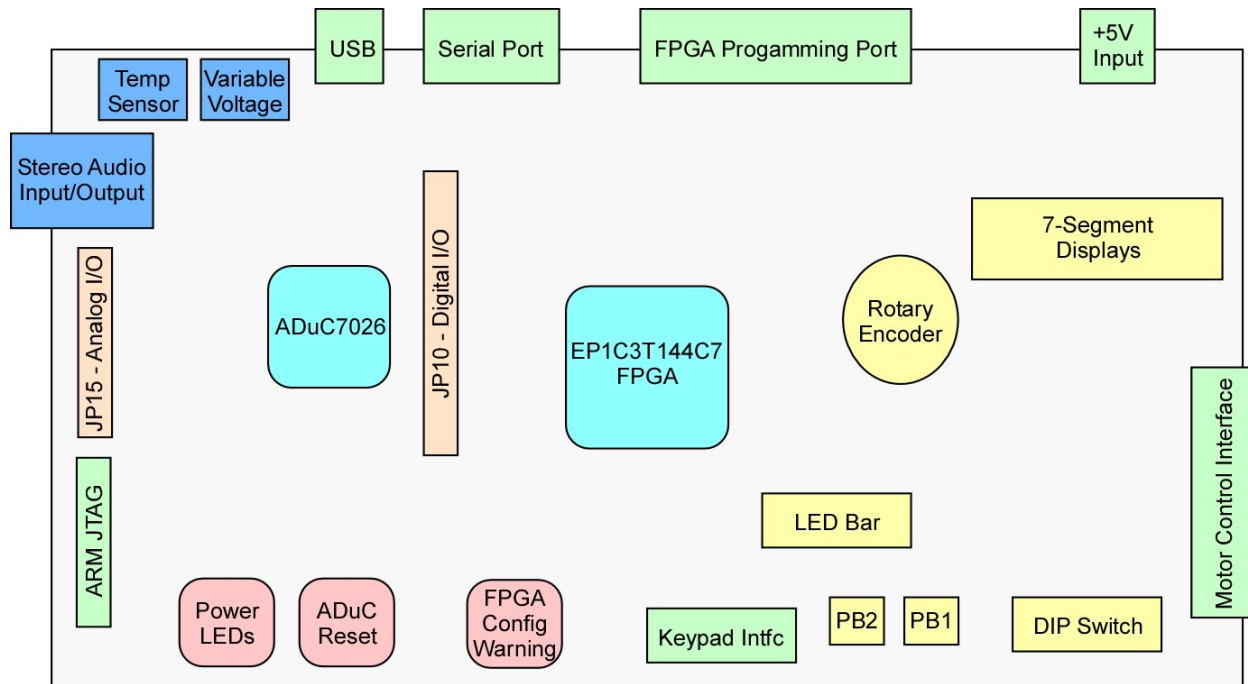
The ECE 315 ADuC7026-Cyclone board is organized around two key components, the Analog Devices ADuC7026 ARM7TDMI-based microcontroller and an Altera EP1C3T144C7 Cyclone FPGA. The FPGA serves as a connection fabric for a number of input/output devices that available on the board. A functional block diagram of the board is shown below.



Programming of the ADuC7026 is accomplished using a JTAG pod that is connected to JP3 on the board. The FPGA is programmed using onboard ByteBlaster-MV emulator logic that connects to the host computer parallel port. The board also includes provisions for an FPGA configuration device, although this is not normally installed on the board.

1.2 Board Locations

The locations of key elements of the ADuC7026-Cyclone board are shown in the figure below.



1.3 Cautions

There are a few cautions that need to be observed in using the ADuC7026-Cyclone board.

1. If the FPGA is configured such that it drives a signal onto the ADuC7026 P0.3 pin, the JTAG emulator pod will not be able to place the ADuC7026 into reset and hence will not be able to program it. For similar reasons, P0.0 (which controls the ADuC7026 boot mode) should never be driven by the FPGA.
2. When designing the FPGA logic, the Quartus project settings should be set so that all unused pins are configured as inputs with weak pull-ups. This will prevent contention between the FPGA and devices on the board. The sample Quartus project has this setting correctly configured.
3. Always use the FPGA pin naming as provided in the sample Quartus project.

2 Using the ARM7-FPGA Board

2.1 Bench Set-Up

The ARM7-FPGA board is to be powered from a +5V regulated power supply supplied in the lab kit. Connect the power cable to the power connector J1 on the ARM-Cyclone board, but do not connect to the AC line. Connect the parallel port cable to the FPGA programming connector J4. Connect the ADuC7026 JTAG pod to the host computer using the USB cable. Connect the 20-pin cable on the JTAG pod to the “ARM JTAG” header JP3 on the ARM-Cyclone board. Plug in the power supply, and verify that both green power LEDs and the red “FPGA not configured” LED on the board are lit.

There are sample projects supplied on the course web page. These projects create a system where a binary count driven out on P4 of the ADuC7026 appears on the LED bar segments 7-0.

2.2 Using Altera Quartus

All work done in the lab should be based on the sample Quartus project available on the course web page. This project has the proper pin assignments as discussed later in this document. Once a design is synthesized and ready for download, open the *Programmer* tool. The programmer should be set for “ByteBlaster-MV” on LPT1, and using JTAG mode. Once the design is downloaded, the red “FPGA not configured” LED on the board will go out. Note that if at any time power is interrupted to the board, the FPGA will need to be reconfigured.

Always verify that you have the correct pin assignment by opening the Pin report after you have synthesized a design. Failure to do so could result in board damage, and in a lot of your time spent debugging signals that are not where you think they are.

2.3 Using Keil μ Vision3

All work done in the lab should be based on the sample Keil μ Vision3 project available on the course web page. (Instructions on project renaming are shown below.) This project has the correct *aduc7026.s* file to perform basic initialization of the processor, including PLL set-up, and has the project settings configured to support a hardware target using JTAG emulation. Once a program has been assembled and is ready for download, click the Debug button. The status window should show the target being erased, then being programmed, and finally the debug window will display. At this point the code can be run and any of the debug functionality put to use. Be aware that the code will remain programmed into the target after the JTAG pod is removed, so the processor will run on its own when it is powered up.

2.3.1 Keil μ Vision3 Project Files

A Keil μ Vision3 project consists of a number of files. The purpose of the key files is listed below.

<i>project_name</i> .UV2	The main project file, containing information on the source code files that are part of the project and most of the project settings.
<i>project_name</i> .OPT	Contains the option settings for the project, including the debugger set-up.
<i>project_name</i> .AXF	Absolute executable file. Contains the binary program image.

<i>project_name</i> .HTM	Contains the call graph and stack depth analysis, although most of this information relies on information supplied by the C compiler that will not be present in our assembly code.
<i>project_name</i> .PLG	Project build log. (The file is in HTML format.)
<i>project_name</i> .MAP	Linker map file.

2.3.2 Renaming Keil μ Vision3 Projects

A sample Keil μ Vision3 project is available on the web, containing all necessary files to build a complete application for the ARM7-FPGA board. To rename the project, perform the following steps;

1. Rename *project_name*.UV2 and *project_name*.OPT to the new project name (i.e. *lab1*.UV2 and *lab1*.OPT).
2. Open the UV2 file using a text editor (i.e. Notepad), and change the line “**OutName** (*project_name*)” to replace *project_name* with your new project name (i.e. “OutName (*lab1*)”). This step can also be accomplished by opening the renamed project in μ Vision3, selecting **Project**→**Options for Target** ‘**Target 1**’ from the main menu, clicking the **Output** tab, and then changing the **Name of Executable**: field to the new project name.
3. The renamed project is now ready for use.

3 Input/Output Devices

3.1 Analog Input/Output

The ADuC7026 has an impressive analog input/output capability, with a 16-channel, 12-bit analog-to-digital converter (ADC) and four 12-bit voltage output digital-to-analog converter (DAC) channels. The board brings all of the available analog interface signals out to headers JP1 and JP15. Several ADC and DAC channels are dedicated to specific uses on the board, as listed below. ADC channels 4-12 and DAC channels 2-3 are available for general use.

3.1.1 Potentiometer Input

ADC channel 0 is driven by a voltage that can be varied with potentiometer R1. The input can be varied between 0V and 3.3V.

3.1.2 Temperature Sensor

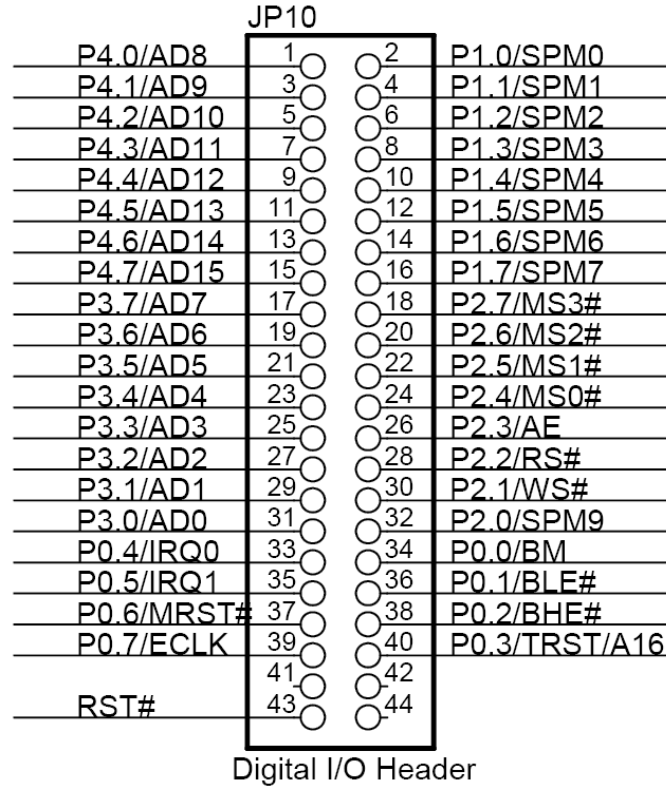
ADC channel 1 is driven by an LM34 temperature sensor. The LM34 provides an output voltage proportional to the temperature, where by $V_{OUT} = ^\circ F \cdot 10mV$.

3.1.3 Stereo Audio Interface

DAC channels 0/1 and ADC channels 2/3 are use to implement a stereo, line-level audio interface. Both inputs and outputs are AC-coupled. The audio interface should be used with the ADuC7026 internal 2.5V precision reference, to give an input/output range of $\pm 1.25V$.

3.2 Digital Input/Output

The board provides a number of devices for digital input/output. Connections to these devices in the FPGA should always be done using the pin names that are provided in the sample Quartus project. (See also the *FPGA Pin Naming* section later in this document.) All of the ADuC7026 GPIO pins connect to the FPGA and are made available at header JP10, as shown below.



3.2.1 Pushbuttons

Two pushbuttons with pull-up resistors are provided. Debouncing can be performed in the FPGA hardware or in the ADuC7026 software. The switches are active-low, i.e. read as 0 when pressed.

3.2.2 DIP Switch

An 8-position DIP switch with pull-up resistors is provided. The switches are active-low, i.e. read as 0 when ON.

3.2.3 Rotary Encoder

A quadrature rotary encoder using optical switches is provided. The encoder also includes a push-to-select switch. The select switch is active-low, i.e. read as 0 when ON. The encoder's A and B outputs do not need to be debounced, however, the push-to-select switch does.

3.2.4 Keypad Interface

A 9-pin header is used to interface to a matrix-connected keypad. The outputs used to drive the keypad must be configured as open-drain drivers. The connections to the keypad all have pull-up resistors on the board.

3.2.5 LED Bar Display

A 10-segment LED bar display with current limiting resistors is provided. The LED segments are configured as active-low, i.e. a 0 output turns on the segment.

3.2.6 Four-Digit LED 7-segment Display

A four-digit multiplexed common-cathode LED display is provided. The cathode drive is implemented with FET switches. The segment and digit drive are both active-high.

3.2.7 Serial Communication Interfaces

The board supports both RS-232 and USB communications. The RS-232 interface uses a Maxim MAX3231 level converter and a standard DB-9 connector. The USB interface is based on the Silicon Laboratories CP-2102 USB interface device. On the microcontroller interface, this device uses standard logic-level RS-232 formatted signals. The host-side interface is a standard USB-B connector. The proper drivers must be installed on the host, which will make the USB interface appear to be a COM port to the host computer software. Jumpers installed on JP4 determine which serial interface is in use. If serial communications are not used, the jumper should be removed to prevent contention on ADuC7026 pin P1.0.

3.2.8 Motor Control Interface

The motor control interface (J5) provides 5V, 3.3V, and 14 FPGA signals on the DB25 connector. A motor driver board is available that can be plugged in to provide control of a stepper motor or two DC motors. Documentation on the motor control board is available on the course web page. Other boards can plug in to this connector as well, to provide other external interfaces.

4 FPGA Pin Naming

A sample project is provided which includes all pin assignments. This project should be used as the starting point for all FPGA designs for the ARM7-FPGA board. The pin names, FPGA pin numbers, and corresponding board connections are shown below.

FPGA Pin Naming			
Pin Name	Pin	Board Connection	Remarks
CLK1	17	20MHz oscillator	FPGA clock input CLK1
CLK0_ARM	16	ADuC7026 P0.7/ECLK	FPGA clock input CLK0
nRST	11	RST#	Manual reset signal Do not configure this pin as an FPGA output
P0_0	2	ADuC7026 P0.0	Do not configure this pin as an FPGA output
P0_1	4	ADuC7026 P0.1	
P0_2	6	ADuC7026 P0.2	
P0_3	10	ADuC7026 P0.3	Do not configure this pin as an FPGA output
P0_4	3	ADuC7026 P0.4	
P0_5	5	ADuC7026 P0.5	
P0_6	7	ADuC7026 P0.6	
P1_0	104	ADuC7026 P1.0	
P1_1	106	ADuC7026 P1.1	
P1_2	108	ADuC7026 P1.2	
P1_3	110	ADuC7026 P1.3	
P1_4	112	ADuC7026 P1.4	
P1_5	114	ADuC7026 P1.5	
P1_6	120	ADuC7026 P1.6	
P1_7	122	ADuC7026 P1.7	
P2_0	144	ADuC7026 P2.0	
P2_1	142	ADuC7026 P2.1	
P2_2	140	ADuC7026 P2.2	
P2_3	134	ADuC7026 P2.3	
P2_4	132	ADuC7026 P2.4	
P2_5	130	ADuC7026 P2.5	
P2_6	128	ADuC7026 P2.6	
P2_7	126	ADuC7026 P2.7	
P3_0	1	ADuC7026 P3.0	
P3_1	143	ADuC7026 P3.1	
P3_2	141	ADuC7026 P3.2	
P3_3	139	ADuC7026 P3.3	
P3_4	133	ADuC7026 P3.4	

FPGA Pin Naming			
Pin Name	Pin	Board Connection	Remarks
P3_5	131	ADuC7026 P3.5	
P3_6	129	ADuC7026 P3.6	
P3_7	127	ADuC7026 P3.7	
P4_0	105	ADuC7026 P4.0	
P4_1	107	ADuC7026 P4.1	
P4_2	109	ADuC7026 P4.2	
P4_3	111	ADuC7026 P4.3	
P4_4	113	ADuC7026 P4.4	
P4_5	119	ADuC7026 P4.5	
P4_6	121	ADuC7026 P4.6	
P4_7	123	ADuC7026 P4.7	
ENC_A	75	Rotary encoder A	
ENC_B	74	Rotary encoder B	
ENC_SW	73	Rotary encoder select	Active-low switch
PB1	37	Pushbutton switch 1	Active-low switch (S3 on schematic)
PB2	36	Pushbutton switch 2	Active-low switch (S2 on schematic)
KEYPAD0	26	Keypad (JP8-1)	
KEYPAD1	27	Keypad (JP8-2)	
KEYPAD2	28	Keypad (JP8-3)	
KEYPAD3	31	Keypad (JP8-4)	
KEYPAD4	32	Keypad (JP8-5)	
KEYPAD5	33	Keypad (JP8-6)	
KEYPAD6	34	Keypad (JP8-7)	
KEYPAD7	35	Keypad (JP8-8)	
DIGIT4	82	Right digit	Active-high
DIGIT3	83	Middle right digit	Active-high
DIGIT2	84	Middle left digit	Active-high
DIGIT1	85	Left digit	Active-high
SEGA	98	7-segment A	Active-high
SEGB	103	7-segment B	Active-high
SEGC	97	7-segment C	Active-high
SEGD	94	7-segment D	Active-high
SEGE	91	7-segment E	Active-high
SEGF	99	7-segment F	Active-high
SEGG	100	7-segment G	Active-high
SEGP	96	7-segment decimal point	Active-high
LED_BAR0	51	LED bar segment 0	Active-low

FPGA Pin Naming			
Pin Name	Pin	Board Connection	Remarks
LED_BAR1	50	LED bar segment 1	Active-low
LED_BAR2	49	LED bar segment 2	Active-low
LED_BAR3	48	LED bar segment 3	Active-low
LED_BAR4	47	LED bar segment 4	Active-low
LED_BAR5	42	LED bar segment 5	Active-low
LED_BAR6	41	LED bar segment 6	Active-low
LED_BAR7	40	LED bar segment 7	Active-low
LED_BAR8	39	LED bar segment 8	Active-low
LED_BAR9	38	LED bar segment 9	Active-low
SW0	59	DIP switch 0	Active-low
SW1	58	DIP switch 1	Active-low
SW2	57	DIP switch 2	Active-low
SW3	56	DIP switch 3	Active-low
SW4	55	DIP switch 4	Active-low
SW5	54	DIP switch 5	Active-low
SW6	53	DIP switch 6	Active-low
SW7	52	DIP switch 7	Active-low
MC4	79	Motor control J1-4	
MC5	77	Motor control J1-5	
MC6	124	Motor control J1-6	
MC7	71	Motor control J1-7	
MC8	69	Motor control J1-8	
MC10	68	Motor control J1-10	
MC11	62	Motor control J1-11	
MC12	60	Motor control J1-12	
MC17	78	Motor control J1-17	
MC18	76	Motor control J1-18	
MC19	72	Motor control J1-19	
MC20	70	Motor control J1-20	
MC23	67	Motor control J1-23	
MC24	61	Motor control J1-24	