Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science

6.111 — Introductory Digital Systems Laboratory

FPGA Module

D. E. Troxel[1] (April 8, 2002)
(Revised: January 23, 2004)

1 WARNING - BURN OUT

A number of students have burned out FPGAs. The following describes the probable cause. This has not been confirmed by experiment. I really don't want to perform this experiment. The FPGAs that have been fried are the 10K70 (the right hand one). The 10K70 is surface mounted and it is expensive to replace. In addition, the pc board is often damaged in trying to replace the 10K70. We have a limited number of pc boards!

Altera states that unused pins MUST be unconnected. This is not possible for the pins connected to the AD bus (see below). In particular, you must not have the switches driving the AD bus unless the Flex devices are tristated for all of the AD bus pins that are driven by the switches. Remove the jumper used to enable the switches by grounding NUSW. If in doubt, read the handout that describes the kit wiring. If you can't find it then look for it on the web page. All of these AD bus pins must be used as inputs, outputs, or specifically tri-stated. In addition, all pins that are connected to a 50 pin connector (see below) should be used as inputs, outputs, or specifically tri-stated.

Unfortunately, there is no easy way to tristate unused pins. Altera literature says that unused pins are grounded and MUST be unconnected. Unused pins must be specifically listed in the top level entity and specifically tri-stated. See the example files tristate.acf and tristate.vhd in the directory /mit/6.111/altera/tristate/.

Please use the 10K10 (on the left) until you are familiar with the use and programming procedures. This device is in a socket and at least we can replace it if you burn it out.

2 FPGA Module

The FPGA module consists of two of Altera's FLEX 10K PLDs (one FLEX 10K10 and one FLEX 10K70) which can be accessed through the lab kit's NuBus interface and 50-pin ribbon cable connectors. The state of the NuBus interconnects can be continuously displayed on the lab kit's hex leds by connecting jumpers /LHEX and /HHEX to /CLK (or any other clock). Also, the Proto strip connection labeled NUHEX must be grounded.

[1]This document has been modified from the original version written by Brian Perrin April, 4, 2002.

Figure 1: FPGA Module Block Diagram

The 50-pin ribbon cable connectors can be fed directly into the inputs of the Logic Analyzer via 50-pin connectors on the kit. Note that each gate array has its own 50-pin connector which can be connected either to the K1 or K2 connectors on the kit. BEWARE, the signals which are grounded by the K1 and K2 connectors are different! It is never a good idea to use EDGEs of signals on the 50-pin connectors. Use only "levels" and wait for the ringing to settle down before sampling them.

Most of the NuBus connections to the FPGA are to I/O pins, but one is a clock pin. Both gate arrays are driven by the same clock. Note that AD31 is used for supplying the clock to the CPLD module if one is also used. Thus, this should not be used by either FPGA if a CPLD board is used. Because of a mistake, AD1 is not connected to either FPGA.

Information on the Flex devices can be found on Altera's web site.
The URL is http://altera.com/products/devices/flex10k/f10-index.html. From there, you can click on Data Sheets and find http://altera.com/literature/lit0f10.html and then click on the first data sheet. This tells you more than you want to know! Please don't print it out unless you really want to save the paper copy – it is 128 pages long! Pages 5 through 15 give you a good summary of capabilities.

This data sheet tells you that the 10K10 has 10,000 gates and 576 logic elements while the 10K70 has 70,000 gates and 3744 logic elements. That is a lot of capability. Even badly written HDL code is likely to fit!

These FPGAs have embedded RAM so moderate sized RAMs and ROMs can be realized by using LPM (Library of Parameterized Modules). Click File–>MegaWizard Plug-in Manager to access LPM. See the web page (click on Software Tools) for information on the formats used for embedded ROMs.

3 Programming Overview

The gate arrays are SRAM based which means they need to be configured each time power is applied. Happily, this is done automatically as the gate arrays are wired to a flash prom (EPC2). However, one does have to program the EPC2. This is done by the Max+plusII software using information in a `<project_name>`.pof file and communicating with the EPC2 via a JTAG interface.

You must generate the appropriate `<project_name>`.pof file by selecting the appropriate device (EPF10K10LC84-3 or EPF10K70RC240-2 and the configuration device option of EPC2LC20). See the beginner's guide for details on how to do this.

There is no simple (quick) way to erase the EPC2s. Instead one must "erase" them by programming them with an HDL file which tri-states all of the I/O pins connected to either the NuBus interface or the 50-pin connector (should you ever use it). This should be done whenever you use a new (to you) FPGA Module. Then all you need do is to program your FPGAs. Of course you want to make sure that the two FPGAs never drive the same pin on the NuBus interface, e.g., AD13. Appropriate files are /mit/6.111/altera/test/blank*. Also see /mit/6.111/altera/tristate/.

4 Programming

The following steps are the programming procedure:

Set up your lab kit next to a computer with a programmer attached to it. The PCs running MAX+plusII under Windows have ByteBlasters installed.

Turn on the lab kit. Insert the 10-pin ribbon connector from the programmer into the 10-pin connector socket corresponding to the FLEX 10K PLD you wish to program; the left connector programs the FLEX 10K10, the right connector programs the FLEX 10K70. Be sure the orientation of the cable in the socket is correct by lining up the notch on the socket with the groove on the cable. The red stripe should be on your left (asumming you are in front of the kit).

Using Altera's MAX+plusII software, load the `Programmer` module (from the `MAX+plusII` menu). If this is the first time you are using the programmer, a hardware configuration window will appear. On a PC a ByteBlaster should be selected along with the parallel port

LPT1. You can edit the hardware configuration any time by clicking on Options–>Hardware Setup.

To speed up the programming process turn off the automatic blank check and verify. Click on Options–>Programming Options to get a pop-up window. If any of these options are selected, then de-select them.

Be sure `Multi-Device JTAG Chain` is selected in the `JTAG` menu, then select `Multi-Device JTAG Chain Setup...` from the `JTAG` menu. In the new window, select `EPC2` as the Device Name, and select the programming file using the button. You want the `.pof` version of your compiled code. After choosing the file, click the `Add` button and then `OK`.

Click the `Program` button to program the file. x Programming will take a while. When the programmer is done, it will return a "Programming Successful" message. Your new program will not take effect until you turn the lab kit off and back on again.

5 NuBus Interface

There are 29 I/O pins from each FLEX 10K PLD connected to the NuBus interface, as well as to each other.

One clock pin is on AD0 of the NuBus proto strip. Each gate array has two clocks but you can only use one. Note that AD31 is used for supplying the clock to the CPLD module if one is also used. The clock pin is denoted in Table 1 as `CLK`.

Table 1: Correlation between NuBus Pins and FLEX 10K Pins

NuBus Address	**FLEX 10K10 Pin Number**	**FLEX 10K70 Pin Number**
AD0	1 (CLK)	91 (CLK)
AD1	NONE	NONE
AD2	17	7
AD3	18	8
AD4	19	9
AD5	21	12
AD6	22	13
AD7	23	14
AD8	24	15
AD9	25	17
AD10	27	18
AD11	28	19
AD12	29	20
AD13	30	21
AD14	35	24
AD15	36	25
AD16	37	28
AD17	38	29
AD18	39	30
AD19	47	31
AD20	48	33
AD21	49	34
AD22	50	35
AD23	51	36
AD24	52	38
AD25	53	39
AD26	54	40
AD27	58	41
AD28	59	43
AD29	60	44
AD30	61	45
AD31	62	46

6 50-pin Ribbon Cable Interface

There are two 50-pin ribbon cable connectors on the FPGA module. The one on the left accesses only pins from the FLEX 10K10 PLD, while the one on the right accesses only pins from the FLEX 10K70 PLD. The FLEX 10K10 50-pin connector provides access to an additional 21 pins, while the FLEX 10K70 50-pin connector provides access to an additional 25 pins. Some of the pins on the FLEX 10K10 are input only, designated by `IN` on Table 2.

The signals on the 50-pin connectors are fed directly into the three `L` interfaces for use with the Logic Analyzer, depending on which connector is used to connect with the FPGA module. Be certain that signals labeled as `GND` are either grounded or left floating in the FLEX 10K PLD programming, as these signals are set by the lab kit. Thus, you only have access to 20 more pins of the 10K10 and 24 more pins of the 10K70.

Table 2: Correlation between Ribbon Cable Pins and FLEX 10K Pins

FLEX 10K10 Pin Number	**FLEX 10K70 Pin Number**	**K1 Connector Interface**	**K2 Connector Interface**
-	80	L1-0	L2-8
-	79	L1-1	L2-9
-	78	L1-2	L2-10
-	76	L1-3	L2-11
81	75	L1-4	L2-12
80	74	L1-5	L2-13
79	73	L1-6	L2-14
78	72	L1-7	L2-15
72	71	L1-8	`GND`
71	70	L1-9	L3-0
70	68	L1-10	L3-1
67	67	L1-11	L3-2
66	66	L1-12	L3-3
65	65	L1-13	L3-4
11	64	L1-14	L3-5
10	63	L1-15	L3-6
9	62	`GND`	L3-7
8	61	L2-0	L3-8
7	56	L2-1	L3-9
6	55	L2-2	L3-10
5	54	L2-3	L3-11
84 (`IN`)	53	L2-4	L3-12
44 (`IN`)	51	L2-5	L3-13
42 (`IN`)	50	L2-6	L3-14
2 (`IN`)	49	L2-7	L3-15