DeVry eSOC III Board

User Guide

GLOSSARY

BINARY	base-2 number system: 0, 1
BNC CONNECTOR	connector for coaxial cables
CPLD	Complex Programmable Logic Device
eSOC	electronic system on a chip
DEBOUNCED	a de-bounced push button gives clean pulse transitions
DECIMAL	base-10 number system: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
FLASH MEMORY	a non-volatile memory technology
FPGA	Field Programmable Gate Array
HEXADECIMAL	base-16 number system: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f
HIGH	logic 1
I/O	Input/output
JTAG	Joint Test Action Group
LED	Light Emitting Diode
LOW	logic 0
LSB	Least Significant Bit
MSB	Most Significant Bit
NIOS II	an embedded processor for Altera FPGAs
PLL	Phase-Locked Loop
SPST	Single Pole, Single Throw (a type of switch or push button)
SRAM	Static Random Access Memory
USB	Universal Serial Bus
VERILOG HDL	Verilog hardware description language
VHDL	VHSIC Hardware Description Language
VOLATILE	a type of memory that loses its content when power is lost

TABLE OF CONTENTS

1. Package contents

- eSOC III Board
- 5V Power Supply
- USB Cable
- User Guide

2. Introduction

The purpose of the eSOC III board is to provide students with a means of learning and prototyping digital logic, computer architecture as well as coding HDL for FPGAs. The board provides access to a state-of-the-art Altera FPGA chip (Cyclone III EP3C16Q240C8 – abbreviated as 3C16), which is connected to various components, such as LEDs, 7-segment displays and switches. These features allow students to experiment with various electronic designs. The board can be connected to a PC (desktop or laptop) with a USB port, which allows design configuration files to be transferred from a computer to the board.

Students can use the web edition of Altera Quartus II to design logic for FPGAs using VHDL, Verilog HDL and other design entry methods. Students can also design with libraries of modules, such as the NIOS II processor, to study computer architecture.

WARNING:
The eSOC III kit is only for educational use within a laboratory environment and not for commercial/industrial applications.

3. FPGA

The board has an Altera Cyclone III FPGA chip (EP3C16Q240C8) and a configuration chip (EPCS4).

4. Features and I/O

- USB port for transferring FPGA configurations to the board
- 2-digit 7-segment display
- 8 green LEDs
- 8 red LEDs
- 2 sets of 8-position DIP switches
- 2 de-bounced push buttons
- 40-pin I/O connector for interfacing to other circuitry
- 24 MHz clock that can be multiplied or divided using a PLL
- optional vertical BNC connector that provides for an external 3.3V clock

Figure 1 shows a brief block diagram of the board and the locations of some of its I/O.

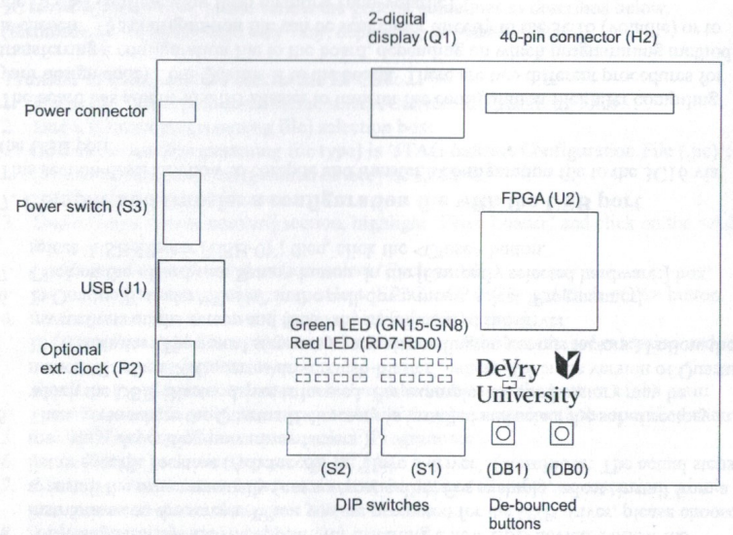


Figure 1 Brief block diagram of eSOC III board (Note: Actual appearance may vary.)

4.1 DIP switches

The sixteen DIP switches (15 to 8 (from S2) and 7 to 0 (from S1)) are SPST (single pole, single throw) switches. When a switch is set to position "0", a low (logic 0) is seen by the 3C16 input pin. When a switch is set to position "1", a high (logic 1) is seen by the 3C16 input pin. The pin assignments can be found in Table A-1 in Appendix A.

4.2 De-bounced push buttons

The two de-bounced push buttons (DB1 and DB0) are SPST push buttons. When a push button is not pressed, a low (logic 0) is seen by the 3C16 input pin. When a push button is pressed, a high (logic 1) is seen by the 3C16 input pin. The pin assignments can be found in Table A-2 in Appendix A.

4.3 Green LEDs, red LEDs and 7-segment display

The eight green and eight red LEDs (GN15 to GN8 and RD7 to RD0) are connected to the I/O lines of the 3C16 through resistors. The LED will not emit light when a low (logic 0) is sent by the 3C16 output pin. The LED emits light when a high (logic 1) is sent by the 3C16 output pin. The pin assignments can be found in Table A-3 in Appendix A.

The segments of the 2-digit 7-segment display behave in a similar way as the LEDs above, since the segments of each digit are just a collection of LEDs shaped into a digit. The pin assignments can be found in Table A-4 in Appendix A.

4.4 40-pin connector

The I/O pins of the 40-pin connector (H2), which conform to 3.3V logic levels, are connected to the 3C16 and are protected against ESD (electrostatic discharge) by ESD protection circuitry. The pin assignments can be found in Table A-5 in Appendix A.

4.5 Clocks

The 24 MHz crystal oscillator (U8), which is connected to the Global Clock 1 input (pin 32) of the 3C16, can be used as the main clock for a design. If a different clock rate is desired, the PLL (phase-locked loop) inside the 3C16 can be used to multiply or divide the 24 MHz reference clock.

The board has provisions for soldering an optional vertical BNC connector (P2) on the board to accept an external clock for designs that require an external clock reference. The BNC connector is connected to Global Clock 2 (pin 33) of the 3C16. The input clock has an input impedance of 50 ohms and the voltage level conforms to standard 3.3V logic levels. This input provides for a usable clock frequency of up to around 200 MHz. (The part number of the optional BNC connector is CP-BNCPC-001 (from Cambridge, www.emersonnetworkpower.com).)

5. Powering up the board

The package includes a 5V DC power supply.

1. Plug the jack of the power supply into the power connector on the board.
2. Plug the power supply in an AC wall outlet.
3. Power up the board using the power switch. The power LED (near the logo of the board) will light up. The board comes with a preloaded demonstration program, which will be described in Section 8.

CAUTION:

1. Please note that signal contention may occur if the board is interfaced to external circuits (i.e., through the 40-pin connector), since output signals from the board and output signals from external circuits may be connected together. Before the board is powered up or used, one should ensure that signal contention will not occur.
2. The fuse (F1) may become very hot if there is a short circuit (e.g. when VCC 3.3V is shorted to ground). If your board loses power, the fuse (F1) may have been activated. This indicates a short circuit has occurred. Remove power from the board and allow the fuse to cool. The source of the short circuit must also be removed before turning on the power again.
3. Some FPGA applications may cause the 3.3V regulator (U6) to get very hot when using an external, high frequency clock.
4. When the board is interfaced to external circuits (e.g. through the 40-pin connector), care must be taken not to overload the 3.3V supply. Excessive loading of the 3.3V regulator may cause the regulator to overheat and may activate the fuse.

6. Software tools

Quartus II Web Edition Software is required to compile and download FPGA designs to the board. Other optional tools, such as the evaluation version of NIOS II Embedded Design Suite (with limited features), can also be used.

In this user guide, the following notations are used:

"" is used to signify pull-down menus.
< > is used to signify buttons.
[ ] is used to signify selection boxes.

6.1 Configure the board for programming

To install the board as the default target programmer for Quartus II, please follow the general guidelines as described below. (i.e. The actual steps may vary, depending on various factors such as version of operating system and version of Altera software.)

Installation of Quartus II:

1. Install Quartus II Web Edition Software on a computer.

Installation of USB driver:

1. Run Quartus II.
2. Power up the board.
3. Plug one end of the USB cable into a USB port of your PC; then, plug the other end of the USB cable into the USB connector of the eSOC III board.
4. A message from Windows appears for installing a new USB device. Follow the instructions on the screen. When you are prompted for the USB driver, please choose to install the driver manually (not automatically). For example, select 'Install from a list or specific location (Advanced)' or 'Have a driver'. (**Reminder**: The actual steps may vary, depending on various factors.)
5. Then, go to where the Quartus II directory is installed and select the sub-directory in which the USB-Blaster driver is located. For example, the subdirectory may be named 'C:\altera\XX\quartus\drivers\usb-blaster', where XX is the version of Quartus II. (**Reminder**: The actual steps may vary, depending on various factors.) Follow the instructions on the screen and finish the installation of the driver.
6. In Quartus II, under **"Tools"** in the pull-down menu, select 'Programmer'.
7. Click on the <Hardware Setup> button. In the [Currently selected hardware:] box, select 'USB-Blaster (USB-0)'; then, click the <Close> button.

7. Compile and transfer a configuration file with the USB port

This section describes how to compile and transfer a configuration file to the 3C16 via the USB port.

The board has a built-in USB Blaster to transfer the configuration file (after compiling your design code) from Quartus II to the board. There are two different procedures for transferring a configuration file to the board, depending on which programming method is chosen. The configuration file can be sent either directly to the 3C16 (volatile) or to the EPCS4 configuration flash memory (non-volatile).

Method #1:
The 3C16 can be configured using the direct programming method. The 3C16 has built-in volatile memory to store the configuration data that is sent directly to the 3C16. Since the configuration data is stored in volatile memory, the configuration data will be lost when the board is powered down. This direct programming method will be useful during the development of a design, since it is faster.

Method #2:
The 3C16 can also be programmed using an alternative method. The configuration data can be sent to a non-volatile configuration flash memory (EPCS4) instead. Since the configuration data is stored in non-volatile memory, the configuration data is retained and will not be lost when the board is powered down. This programming method will be useful if one wants the configuration data to be automatically loaded when the board is powered on.

The next two sections describe how to load the configuration file either directly to the 3C16 (volatile) or to the EPCS4 configuration flash memory (non-volatile).

NOTE:
If the error 'Can't access JTAG chain' is observed during programming, try powering off the board and powering back on, and repeat the programming operation.

7.1 Transfer a configuration file to the 3C16 directly

For a design to be configured in the 3C16, it first needs to be compiled, simulated and debugged. Once the configuration file is generated, it is ready to be loaded into the FPGA. To load the configuration file directly to the 3C16, please follow the general guidelines as described below. (**Reminder**: The actual steps may vary, depending on various factors.)

Transfer of configuration files to the 3C16:

1. Under the **"Tools"** pull-down menu, select the item 'Programmer'.
2. In the [Mode:] selection box, select 'JTAG'.
3. Click <Add File…> button; select the desired file with .sof extension, which was generated during compile time; then, click the <Open> button.
4. In the main window, select the file and then check the [Program/Configure] box.
5. Click <Start> button to download the file to the 3C16 directly.
6. The code should now be running automatically after transferring to the 3C16.

(**Reminder:** Since the configuration data is stored in the volatile memory of the 3C16, the configuration data will be lost when the power is turned off.)

7.2 Transfer a configuration file to the EPCS4 Flash Memory

The non-volatile configuration flash memory (EPCS4) is used to configure the 3C16 automatically after every power up sequence. For a design configuration to be programmed, it first needs to be compiled, simulated and debugged. Once the configuration is generated, it is ready to be loaded into the EPCS4. To configure the

3C16 using the EPCS4, please follow the general guidelines as described below.
(**Reminder**: The actual steps may vary, depending on various factors.)

Transfer of configuration files to the EPCS4:

1. Under the **"File"** pull-down menu, select 'Convert Programming Files…'.
2. Under [Output programming file] selection box:
   (a) Ensure that [Programming file type] is 'JTAG Indirect Configuration File (.jic)';
   (b) Ensure that the [Configuration device] is 'EPCS4';
   (c) Specify the output filename.
3. Under [Input files to convert] section, highlight 'Flash Loader' and click on the <Add Device…> button.
   (a) Under [Device family] section, select 'Cyclone III'.
   (b) Under [Device name] section, select 'EP3C16Q240C8' and click <OK> button.
4. Under [Input files to convert] section, highlight 'SOF Data'.
   (a) Click the <Add file…> button; select the compiled file with .sof extension and click <Open> button.
   (b) Click the <Generate> button, which can be found near the bottom of the main window; a file with .jic extension should then be created with the specified name.
5. Under **"Tools"** pull-down menu, select 'Programmer'.
6. In the [Mode:] selection box, select 'JTAG'.
7. Click the <Add File…> button and select the desired file with .jic extension, which was generated above; then, click <Open> button.
8. In the main window:
   (a) Select the file and then check the [Program/Configure] box.
   (b) Select the 'EPCS4' and then check the [Program/Configure] box.
9. Click <Start> button. Wait until the [Progress:] box, which is near the top right-hand corner of main window, indicates that it is 100% done. Power down the board and then power back up; the new configuration should now be loaded automatically.

(**Reminder:** Since the configuration data is stored in non-volatile memory (EPCS4) , the configuration data is retained even after a power down sequence.)

8. Preloaded demonstration program

The board comes with a preloaded FPGA configuration that is designed to demonstrate the features available on the board. This configuration is preloaded in the EPCS4 configuration flash memory and is automatically loaded into the 3C16 when the board is powered on. The demonstration is designed to run without any external circuitry being connected to the board. Therefore, before powering up the board with the demonstration configuration, please make sure that the board is not connected to any external circuitry through the 40-pin connector. The features of the demonstration program, which are briefly described below, are summarized in Table 8-1.

In the demonstration, a counter is used to count through the two digits of the 7-segment display, i.e., each digit counts in a hexadecimal sequence (0 to F). A DIP switch or a push button can be used to turn on a LED. For example, setting DIP switch 15 to position "1" will light up the LED GN15. More details of the features of the demonstration program can be found in Table 8-1.

Table 8-1 Features of the preloaded demonstration program

Component	Label on the board	Function
7-segment display	Digit 1	It counts from 0 to F (hexadecimal).
	Digit 0	It counts from 0 to F (hexadecimal).
2 sets of 8-position DIP switches	15 to 8 (from S2)	Please ensure that no push buttons are pressed. If a DIP switch is set to position "1", a LED will light up: DIP switch 15 => GN15 DIP switch 14 => GN14 DIP switch 13 => GN13 DIP switch 12 => GN12 DIP switch 11 => GN11 DIP switch 10 => GN10 DIP switch 9 => GN9 DIP switch 8 => GN8
	7 to 0 (from S1)	Please ensure that no push buttons are pressed. If a DIP switch is set to position "1", a LED will light up: DIP switch 7 => RD7 DIP switch 6 => RD6 DIP switch 5 => RD5 DIP switch 4 => RD4 DIP switch 3 => RD3 DIP switch 2 => RD2 DIP switch 1 => RD1 DIP switch 0 => RD0
2 de-bounced push buttons	DB1 and DB0	Please set all DIP switches to position "0" first. If a push button is pressed, a LED will light up: DB1 => a green LED will light up DB0 => a red LED will light up

Appendix A – Pin assignments of FPGA

This section contains tables describing the connections to the pins of EP3C16Q240C8 FPGA (abbreviated as 3C16).

NOTE: The main clock (24 MHz) is connected to pin 32 of the 3C16.
(The optional external clock (P2) is connected to pin 33 of the 3C16.)

Table A-1 DIP switches

pin no. of 3C16	DIP switch	Description	Logic	
			at position 0:	at position 1:
99	0 (from S1)	DIP switch 0	low (logic 0)	high (logic 1)
88	1 (from S1)	DIP switch 1	low (logic 0)	high (logic 1)
87	2 (from S1)	DIP switch 2	low (logic 0)	high (logic 1)
86	3 (from S1)	DIP switch 3	low (logic 0)	high (logic 1)
85	4 (from S1)	DIP switch 4	low (logic 0)	high (logic 1)
84	5 (from S1)	DIP switch 5	low (logic 0)	high (logic 1)
83	6 (from S1)	DIP switch 6	low (logic 0)	high (logic 1)
82	7 (from S1)	DIP switch 7	low (logic 0)	high (logic 1)
81	8 (from S2)	DIP switch 8	low (logic 0)	high (logic 1)
80	9 (from S2)	DIP switch 9	low (logic 0)	high (logic 1)
78	10 (from S2)	DIP switch 10	low (logic 0)	high (logic 1)
76	11 (from S2)	DIP switch 11	low (logic 0)	high (logic 1)
73	12 (from S2)	DIP switch 12	low (logic 0)	high (logic 1)
72	13 (from S2)	DIP switch 13	low (logic 0)	high (logic 1)
71	14 (from S2)	DIP switch 14	low (logic 0)	high (logic 1)
70	15 (from S2)	DIP switch 15	low (logic 0)	high (logic 1)

Table A-2 De-bounced push buttons

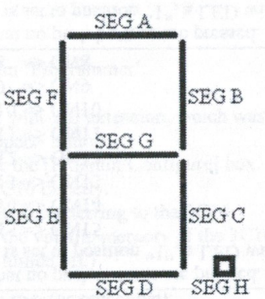
pin no. of 3C16	Push button	Description	Logic	
			if not pressed:	if pressed:
149	DB0	push button DB0	low (logic 0)	high (logic 1)
209	DB1	push button DB1	low (logic 0)	high (logic 1)

Table A-3 LEDs

pin no. of 3C16	LED	Description
69	RD0	red LED 0
68	RD1	red LED 1
65	RD2	red LED 2
64	RD3	red LED 3
63	RD4	red LED 4
57	RD5	red LED 5
56	RD6	red LED 6
55	RD7	red LED 7
52	GN8	green LED 8
51	GN9	green LED 9
50	GN10	green LED 10
49	GN11	green LED 11
46	GN12	green LED 12
45	GN13	green LED 13
44	GN14	green LED 14
43	GN15	green LED 15

Table A-4 2-digit 7-segment display

pin no. of 3C16	Description
20	segment A of Digit 0
21	segment B of Digit 0
22	segment C of Digit 0
37	segment D of Digit 0
38	segment E of Digit 0
39	segment F of Digit 0
41	segment G of Digit 0
4	segment A of Digit 1
5	segment B of Digit 1
6	segment C of Digit 1
9	segment D of Digit 1
13	segment E of Digit 1
18	segment F of Digit 1
19	segment G of Digit 1



NOTE:
SEG H is not connected on either digit.

Table A-5 40-pin I/O connector

pin no. of 3C16	I/O	Description
139	IO1	pin 1 of H2
142	IO2	pin 2 of H2
146	IO3	pin 3 of H2
147	IO4	pin 4 of H2
148	IO5	pin 5 of H2
161	IO6	pin 6 of H2
166	IO7	pin 7 of H2
169	IO8	pin 8 of H2
171	IO9	pin 9 of H2
173	IO10	pin 10 of H2
175	IO11	pin 11 of H2
177	IO12	pin 12 of H2
181	IO13	pin 13 of H2
182	IO14	pin 14 of H2
183	IO15	pin 15 of H2
184	IO16	pin 16 of H2
185	IO17	pin 17 of H2
186	IO18	pin 18 of H2
		pins 19, 21 and 23 of H2 = VCC 3.3V **pins 20, 22 and 24 of H2 = Ground**
187	IO19	pin 25 of H2
188	IO20	pin 26 of H2
189	IO21	pin 27 of H2
195	IO22	pin 28 of H2
197	IO23	pin 29 of H2
198	IO24	pin 30 of H2
199	IO25	pin 31 of H2
216	IO26	pin 32 of H2
217	IO27	pin 33 of H2
223	IO28	pin 34 of H2
230	IO29	pin 35 of H2
235	IO30	pin 36 of H2
237	IO31	pin 37 of H2
238	IO32	pin 38 of H2
239	IO33	pin 39 of H2
240	IO34	pin 40 of H2