

WDC reserves the right to make changes at any time without notice in order to improve design and supply the best possible product. Information contained herein is provided gratuitously and without liability, to any user. Reasonable efforts have been made to verify the accuracy of the information but no guarantee whatsoever is given as to the accuracy or as to its applicability to particular uses. In every instance, it must be the responsibility of the user to determine the suitability of the products for each application. WDC products are not authorized for use as critical components in life support devices or systems. Nothing contained herein shall be construed as a recommendation to use any product in violation of existing patents or other rights of third parties. The sale of any WDC product is subject to all WDC Terms and Conditions of Sales and Sales Policies, copies of which are available upon request.

Copyright (C) 1981-2009 by The Western Design Center, Inc. All rights reserved, including the right of reproduction in whole or in part in any form.



DOCUMENT REVISION HISTORY..... 3

Introduction 4

Features 4

Board Jumper Configuration..... 6

 JUMPER DESCRIPTIONS..... 7

Peripheral Interfaces..... 8

 Clock Sources 8

 Serial ATA (SATA) Interface..... 8

 GPIO Expansion Connector 8

 Santa Cruz Expansion Connector 9

 Mesa Expansion Connector 10

User Interface Signals 11

 Dual 7-Segment Display 12

 User LEDs 12

 Rotary Switch 13

 LCD Connector 13

 Pushbuttons 14

Components..... 15

Appendix A. Schematics..... 16

 Page 1/6 16

 Page 2/6 17

 Page 3/6 18

 Page 4/6 19

 Page 5/6 20

 Page 6/6 21



DOCUMENT REVISION HISTORY

Version	Date	Author	Description
1.0	12-02-09	David Gray	Initial Document Release
1.1	14-Dec-09	Cory Walton	Added Appendix B – Terbi Demo Code details and examples
1.2	17-Dec-09	David Gray	Revised 7 Segment display FPGA mapping



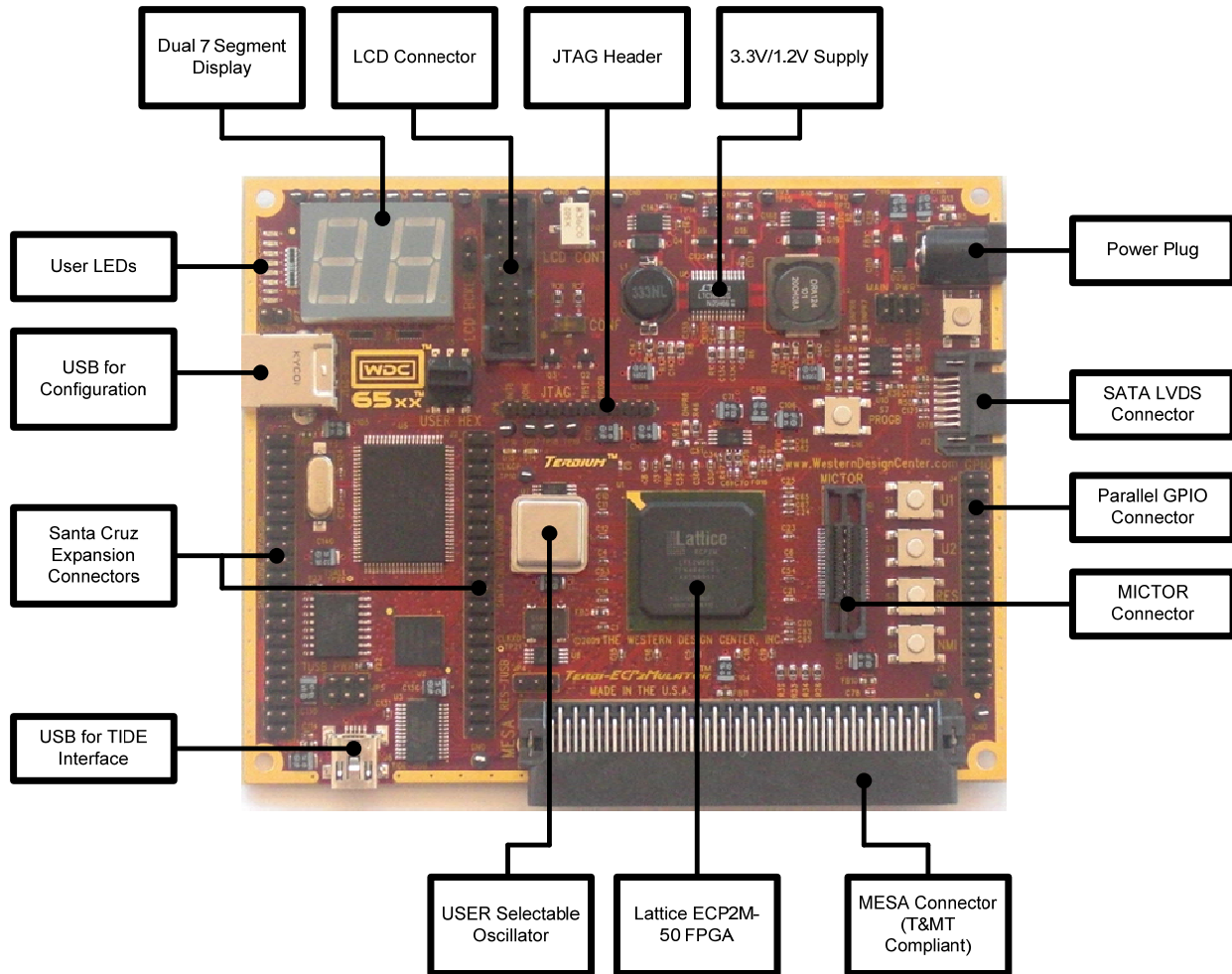
Introduction

This document describes the features and functionality of the WDC Terbi-ECP2Mulator Board. This board is designed as a hardware emulation platform for design and development with WDC's 65xx Microprocessor Technology.

Features

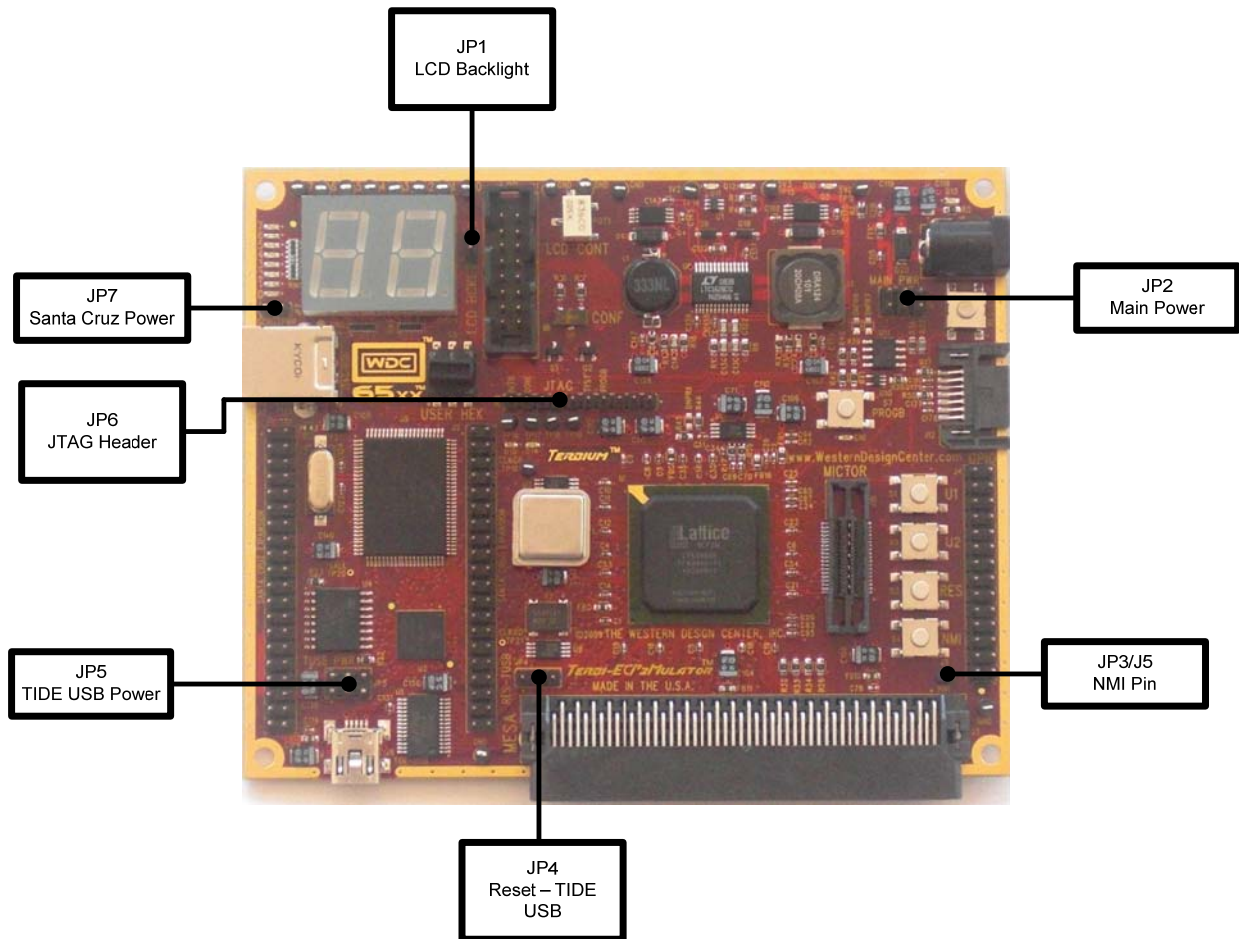
- Lattice ECP2M-50 FPGA with 48 kLUTs, 4147 kbit of Embedded Block RAM, 22 sysDSP™ blocks, 88 18x18 multipliers, 10 PLLs, and 270 user I/O pins
- Lattice MachXO™ with 640 LUTs and 6.1 Kbit of RAM
- USB 2.0 connector and integrated ispDOWNLOAD® cable for JTAG programming the FPGA
- Flywire connector for programming using an ispDOWNLOAD cable (available separately)
- USB 2.0 compatible host connectors for interfacing with WDC's ProSDK Development Tools
- SATA interface with two LVDS signal pairs for high-speed data transfer
- Altera Santa Cruz compliant connectors (2 x 40) providing 46 lines of I/O Expansion
- 100-pin "MESA" connector compliant with T&MT USB PHY Standards
- 28-pin expansion header for I/O. Compatible with WDC's Parallel IO Board
- LCD connector for character displays, with contrast potentiometer
- 38-pin MICTOR connector for high speed Logic Analyzer connection
- 25 MHz oscillator with clock distribution buffer
- 4-pin DIP half socket for user selected oscillator
- One HEX Rotary Switch
- Dual Character 7-segment Display
- 8 LEDs
- 8 Test Points
- 4 Pushbutton switches for debug and user input
- Green LED to indicate the proper operation of the 3.3V and 2.5 V power supplies
- Blue LED which shows the configuration status ("DONE")
- Red LED to signal that the FPGA can be configured ("INIT")
- Yellow LED indicating the FPGA PROGRAM# I/O is asserted ("PROGRAM#")
- Program key to initiate the configuration sequence of the FPGA from SPI Flash memory
- Reset key
- 5V power supply
- Switching regulator for the generation of the 3.3V I/O voltage and the 1.2Vcore voltage
- Robust power configuration with jumper selections to power board with either of the two USB ports or the DC input jack

Figure 1 – Terbi-ECP2Mulator



Board Jumper Configuration

Figure 2 - Terbi Jumper Layout





JUMPER DESCRIPTIONS

JP1 – LCD BACKLIGHT

1-2: Backlight ON
No Jumper: Backlight OFF

JP2 – Main Power Jumper

1-3: TIDE USB VBUS Selected. Recommend setting JP5 jumpers in 2-4 position. (DEFAULT)
5-3: JTAG USB VBUS Selected.
2-4: 5V DC Input from Wall Transformer is used –
6-4: MESA_CARDPWR Selected.

JP4 – TIDE USB RESET

No Jumper: FT245 TIDE USB Controller RESET# signal is pulled low
1-2: RESET# is de-asserted when USB is plugged in (DEFAULT)
2-3: RESET# is controlled by the FPGA

JP3/J5 – EXTERNAL NMI

This is a single pin for connecting to the NMI pin of the WDC IO board.

JP5 - TIDE USB POWER

2-4: Provides 5V supply from TIDE USB VBUS to Power the board. (DEFAULT)
3-4: Selects 3.3V supply for FT245RL TIDE USB Controller.
6-4: Selects 5V supply (non VBUS) for FT245RL TIDE USB Controller.
1-3: Invalid Jumper Setting – DO NOT USE
5-3: Invalid Jumper Setting – DO NOT USE

JP6 - JTAG Header Connector:

Table 1 - JTAG Header Connector

Pin	Signal Name
1	+3.3V
2	TDO
3	TDI
4	PROGB
5	PROGB
6	TRST
7	TMS
8	GND
9	DONE
10	INITB

JP7 - SantaCRUZ power jumper

1-2: SantaCRUZ power connected
No Jumper: SantaCRUZ power NOT Connected



Peripheral Interfaces

Clock Sources

A general purpose oscillator socket is provided. The default oscillator installed in the board is 24MHz. A on-board 25MHz oscillator supplies the FPGA (pin AD15), the CPLD (pin A8). The frequency can be measured via test point TP21 (CLKXO). To generate other clock frequencies use the PLLs of the FPGA. Detailed information on the usage of the PLLs can be found on the Lattice website and in the LatticeECP2/M Family Data Sheet.

The Cypress USB controller for ispDOWNLOAD configuration requires a 24MHz quartz crystal.

Serial ATA (SATA) Interface

The SATA interface provides a convenient method for evaluating or using LVDS or LVPECL signals with the FPGA. The SATA connector has differential nets with high-speed signals connected to them. See Table 2 for SATA connection information (J12). The positive signal is connected with a plus (+), the negative with a minus (-).

Table 2 - SATA Connector J12

Pin	Signal Name	FPGA Pin	Pin	Signal Name	FPGA Pin
1	GND	-	2	RXP	A18
3	RXN	B18	4	GND	-
5	TXN	B21	6	TXP	A21
7	GND	-	-	-	-

GPIO Expansion Connector

The GPIO Expansion connector provides 20 user I/Os connected to the FPGA. The remaining pins serve as Ground pins (4 pins are currently unconnected). The expansion connector is configured as one 2x14 100mil centered pin header. This header is compatible with WDC's Parallel IO board but can be used for any purpose.

Table 3 describes the connections to the FPGA.

Table 3 - GPIO Expansion Connector J4

Pin	Signal Name	FPGA Pin	Pin	Signal Name	FPGA Pin
1	UNCONNECTED	-	2	UNCONNECTED	-
3	GND	-	4	GND	-
5	GPIO_HSI	F18	6	GPIO_HSO	J20
7	GPIO_DATA0	F12	8	GPIO_DATA1	E12
9	GPIO_DATA2	A6	10	GPIO_DATA3	A7
11	GPIO_DATA4	A8	12	GPIO_DATA5	A9
13	GPIO_DATA6	D12	14	GPIO_DATA7	E13
15	GPIO_SW1	G10	16	GPIO_SW2	F17
17	GPIO_LED7	A5	18	GPIO_LED6	A4
19	GPIO_LED5	B7	20	GPIO_LED4	B8
21	GPIO_LED3	G11	22	GPIO_LED2	E11
23	GPIO_LED1	D11	24	GPIO_LED0	D10
25	GND	-	26	GND	-
27	UNCONNECTED	-	28	UNCONNECTED	-



Santa Cruz Expansion Connector

The Santa Cruz expansion connector provides 46 user I/Os connected to the FPGA. The remaining pins serve as power and clock supplies for expansion boards. The expansion connector is configured as two 2x20 100mil centered pin headers (J1 and J2). Tables 4 and 5 describe the connections to the FPGA.

Table 4 - Santa Cruz Expansion Connector J1

Pin	Signal Name	FPGA Pin	Pin	Signal Name	FPGA Pin
1	GND	-	2	VCC_50	-
3	SCRUZ_PIO[40]	R5	4	SCRUZ_PIO[29]	N6
5	SCRUZ_PIO[30]	N2	6	SCRUZ_PIO[31]	P6
7	SCRUZ_PIO[32]	N5	8	SCRUZ_PIO[33]	P1
9	SCRUZ_PIO[34]	P2	10	SCRUZ_PIO[35]	P3
11	SCRUZ_PIO[36]	P4	12	SCRUZ_PIO[37]	R2
13	SCRUZ_PIO[38]	R3	14	SCRUZ_PIO[39]	R4
15	SCRUZ_PIO[41]*	T1	16	SCRUZ_PIO[42]*	T2
17	SCRUZ_PIO[43]*	T6	18	SCRUZ_PIO[44]*	T7
19	SCRUZ_PIO[45]*	U1	20	SCRUZ_PIO[46]*	U2
21	Vunreg	-	22	GND	-
23	NC	-	24	GND	-
25	VCC_33	-	26	GND	-
27	VCC_33	-	28	GND	-
29	OSC	M5	30	GND	-
31	CLK1	N1	32	GND	-
33	CLK2	AB6, N7***	34	GND	-
35	VCC_33	-	36	GND	-
37	VCC_33	-	38	GND	-
39	VCC_33	-	40	GND	-

Table 5 - Santa Cruz Expansion Connector J2

Pin	Signal Name	FPGA Pin	Pin	Signal Name	FPGA Pin
1	RESETB	P5	2	GND	-
3	SCRUZ_PIO[0]	W5	4	SCRUZ_PIO[1]	Y4
5	SCRUZ_PIO[2]	W6	6	SCRUZ_PIO[3]	V6
7	SCRUZ_PIO[4]	AA3	8	SCRUZ_PIO[5]	AB2
9	SCRUZ_PIO[6]	T8	10	SCRUZ_PIO[7]	U7
11	SCRUZ_PIO[8]	U8	12	SCRUZ_PIO[9]	T9
13	SCRUZ_PIO[10]	V8	14	SCRUZ_PIO[11]	W8
15	SCRUZ_PIO[12]	Y6	16	SCRUZ_PIO[13]	Y5
17	SCRUZ_PIO[14]	AB3	18	SCRUZ_PIO[15]	AB4
19	GND	-	20	NC	-
21	SCRUZ_PIO[16]	AB5	22	GND	-
23	SCRUZ_PIO[17]	AA6	24	GND	-
25	SCRUZ_PIO[18]	V9	26	GND	-
27	SCRUZ_PIO[19]	U9	28	SCRUZ_PIO[20]	U10
29	SCRUZ_PIO[21]	T10	30	GND	-
31	SCRUZ_PIO[22]	W9	32	SCRUZ_PIO[23]	Y8



33	SCRUZ_PIO[24]	AA7	34	GND	-
35	SCRUZ_PIO[25]	Y7	36	SCRUZ_PIO[26]	AB7
37	SCRUZ_PIO[27]	M3	38	SCRUZ_CARDSELB	P7
39	SCRUZ_PIO[28]	M4	40	GND	-

Mesa Expansion Connector

The MESA Expansion Connector is a 100 pin connector providing 68 user I/Os connected to the FPGA. It is also compliant with T&MT USB PHY Standards. A T&MT PHY board can be added to Terbi for testing a USB PHY Macro cell. Table 6 describes the connections to the FPGA.

Table 6 - MESA Expansion Connector J3

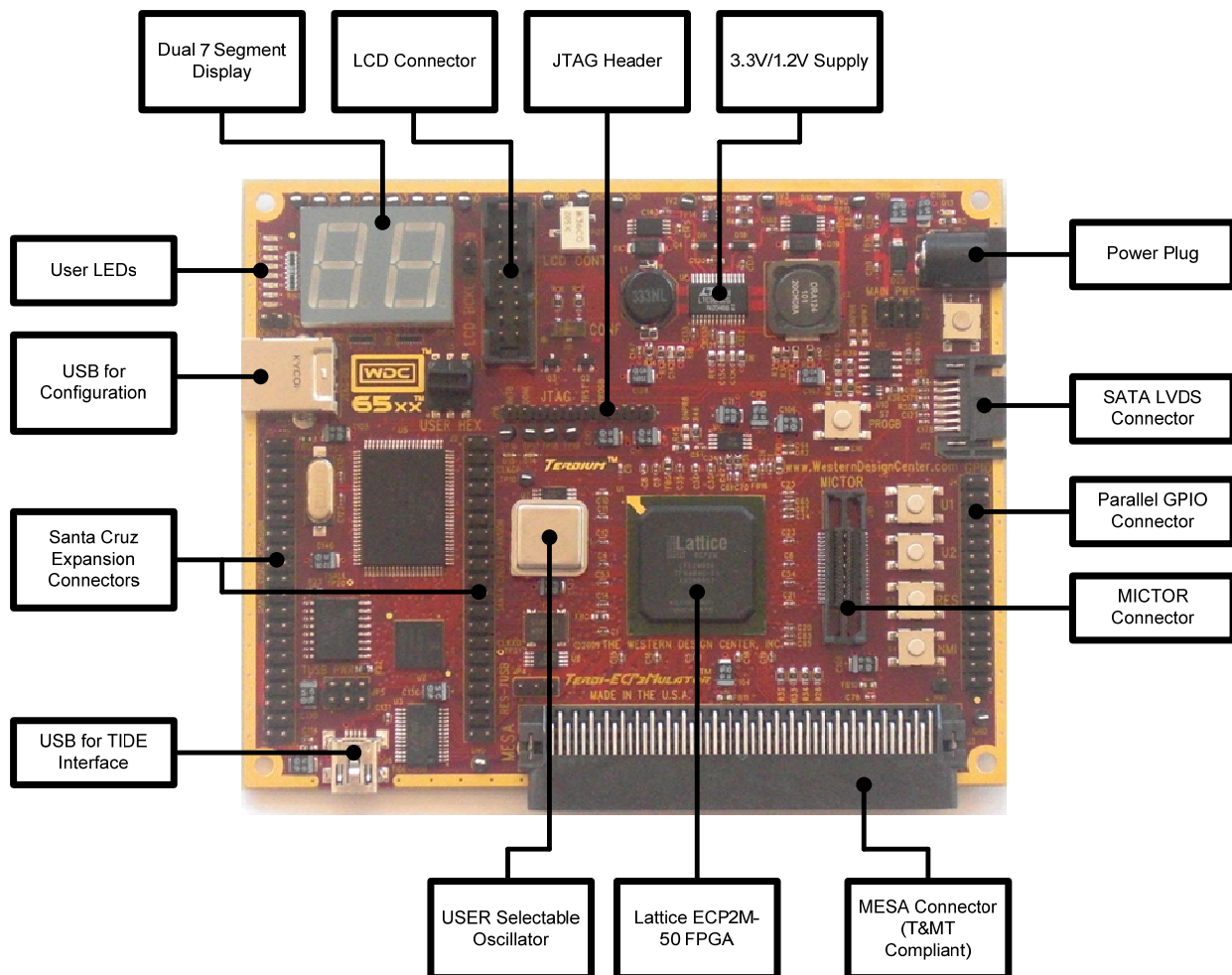
Pin	Signal Name	FPGA Pin	Pin	Signal Name	FPGA Pin
1	MESA_GPIO0	J22	2	GND	-
3	MESA_GPIO2	J16	4	GND	-
5	MESA_GPIO4	K16	6	MESA_GPIO5	L19
7	MESA_GPIO7	L17	8	+3.3V	-
9	GND	-	10	MESA_VCTRL3	P20
11	MESA_GPIO9	K22	12	MESA_GPIO11	L21
13	GND	-	14	MESA_VCTRL4	P21
15	MESA_VDATA4	AB17	16	+3.3V	-
17	MESA_RESET	V11	18	MESA_CTRL5	V15
19	MESA_CTRL6	V12	20	MESA_CTRL7	W12
21	GND	-	22	MESA_SUSPENDB	U11
23	MESA_CTRL2	T11	24	GND	-
25	MESA_CTRL3	U12	26	MESA_GPIO1	K18
27	GND	-	28	+5.0V	-
29	MESA_GPIO3	K17	30	MESA_VCTRL5	P22
31	MESA_DATA15	AB14	32	GND	-
33	MESA_DATA13	AA13	34	MESA_DATA11	U15
35	GND	-	36	MESA_DATA9	Y12
37	MESA_DATA7	V10	38	+3.3V	-
39	+3.3V	-	40	MESA_GPIO15	M20
41	MESA_DATA5	Y9	42	MESA_DATA3	T14
43	GND	-	44	MESA_DATA1	AA10
45	MESA_VDATA0	AB10	46	GND	-
47	MESA_CARDPWR	-	48	MESA_VDATA6	AB8
49	MESA_CARDPRESB	W11	50	MESA_GPIO13	M18
51	GND	-	52	MESA_SYSCLOCK	R19
53	GND	-	54	GND	-
55	MESA_VCTRL0	N16	56	MESA_GPIO6	K21
57	+3.3V	-	58	MESA_GPIO8	L20
59	MESA_VCTRL2	P19	60	MESA_CTRL12	V13
61	MESA_GPIO10	L22	62	GND	-
63	MESA_GPIO12	L16	64	MESA_CTRL0	AA17
65	GND	-	66	MESA_CTRL8	AB16
67	MESA_CTRL4	AA16	68	GND	-
69	+3.3V	-	70	MESA_VDATA1	W14
71	MESA_VDATA2	AA15	72	MESA_CTRL9	AB15
73	+3.3V	-	74	MESA_CTRL10	T12
75	MESA_CTRL13	T13	76	GND	-
77	MESA_CTRL11	U13	78	MESA_GPIO14	M19
79	MESA_VCTRL1	P18	80	GND	-



81	+3.3V	-	82	MESA_DATA14	Y14
83	MESA_DATA12	AA14	84	GND	-
85	MESA_DATA10	AB13	86	MESA_DATA8	AA12
87	+3.3V	-	88	MESA_DATA6	AB12
89	MESA_CTRL1	AB11	90	MESA_DEVCLOCK	P21, L18
91	MESA_DATA4	Y11	92	GND	-
93	MESA_DATA2	U14	94	MESA_DATA0	AB10
95	GND	-	96	MESA_VDATA3	AA9
97	MESA_VDATA5	Y9	98	MESA_VDATA7	V10
99	MESA_VCTRL6	V12	100	GND	-

User Interface Signals

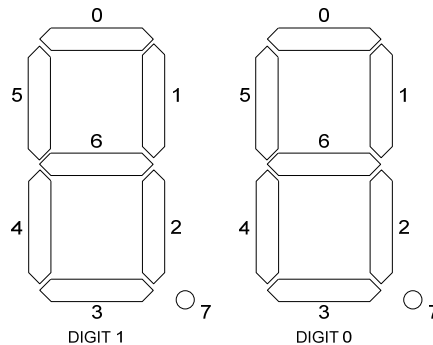
Figure 3 - User Interface Features



Dual 7-Segment Display

The Dual 7-Segment Display is wired as follows:

Table 7 - Dual 7 Segment DISP1 Pin Definition



Signal Name	Segment Bit – Digit 1	FPGA Pin	Signal Name	Segment Bit Digit 0	FPGA Pin
Dig1_0	0	L1	Dig0_0	0	K5
Dig1_1	1	L3	Dig0_1	1	M7
Dig1_2	2	L4	Dig0_2	2	L7
Dig1_3	3	L2	Dig0_3	3	J4
Dig1_4	4	K1	Dig0_4	4	J5
Dig1_5	5	K2	Dig0_5	5	K6
Dig1_6	6	L5	Dig0_6	6	J6
Dig1_7	7	L6	Dig0_7	7	K7

The signals of the 7-segment display are active-low. When the signal is logic '0', the segment is lit.

User LEDs

Eight LEDs can be used for custom status signaling. They signals of the LEDs are active-low. When the signal is logic '0', the LED is on. The LEDs are controlled using the signals below:

Table 8 - User LED – LED_7-LED_0 Connection

LED	Signal Name	FPGA Pin	LED	Signal Name	FPGA Pin
0	LED_0	F8	1	LED_1	C1
2	LED_2	B1	3	LED_3	F10
4	LED_4	E10	5	LED_5	D9
6	LED_6	C5	7	LED_7	A3

Rotary Switch

The Rotary Switch (CD16RK1) is a 16 position (no complement) hexadecimal switch with knob.

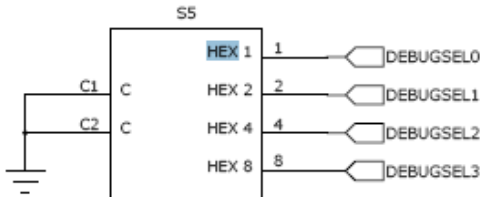


Table 9 - Rotary Switch S5 Connection

HEX	Signal Name	FPGA Pin	HEX	Signal Name	FPGA Pin
1	DEBUGSEL0	F7	4	DEBUGSEL2	G7
2	DEBUGSEL1	E8	8	DEBUGSEL3	G8

Below is a table showing the value of each HEX signal based on the position of the rotary switch.

POS.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
C	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
1		●		●		●		●		●		●		●		●
2			●	●			●	●			●	●			●	●
4					●	●	●	●					●	●	●	●
8									●	●	●	●	●	●	●	●

LCD Connector

The LCD connector is a 16-pin header with a standard pinning for LCD modules with back-light (e.g. Truly MTCC202DPRN-1N). In order to use an LCD module, attach it to the connector via a 16-pin ribbon cable or connector built on LCD. Put a jumper on connector JP1 to turn on the backlight of the LCD. The contrast of the LCD module is adjustable with the potentiometer POT1, because different LCD modules need different voltages for the best contrast.

Note: The LCD module is tied to a 5V supply. The LatticeECP2-50 to LCD interface is 3.3V.



Table 10 - LCD Connector J11 Pin Definition

Pin	Signal Name	FPGA Pin	Pin	Signal Name	FPGA Pin
1	GND	-	2	VCC5V	-
3	CONTRAST	-	4	LCD_REGSEL	
5	LCD_RWB	E3	6	LCD_EN	E4
7	LCD_DATA0	E5	8	LCD_DATA1	D3
9	LCD_DATA2	C2	10	LCD_DATA3	C3
11	LCD_DATA4	D4	12	LCD_DATA5	D7
13	LCD_DATA6	E6	14	LCD_DATA7	D6
15	BACKLIGHT	-	16	GND	-

Pushbuttons

The board is equipped with 4 pushbuttons, two of which are user selectable.

Table 11 - Pushbutton U1, U2, NMI, RES Connection

Button	Signal Name	FPGA Pin	Button	Signal Name	FPGA Pin
U1	USER_PB1	D8	NMI	NMIB_IN	B6
U2	USER_PB2	C8	RES	RESB_IN	C7



Components

CPLD

The Lattice MachXO LCMXO640 is a non-volatile, instant-on, reprogrammable logic device. It supports “background programming” called TransFR™ (i.e., the device can be programmed while in operation).

The MachXO comes preprogrammed from the factory. The factory program permits the CY7C68013A/MachXO combination to work as a built-in USB ispDOWNLOAD cable. Using ispVM software the built-in download cable permits the FPGA, and SPI PROM, to be programmed. It is not recommended for the MachXO to be reprogrammed.

For further information, please consult the [MachXO Family Data Sheet](#).

FPGA

The LatticeECP2M-50-484 FPGA represents the heart of the board. It has the following features:

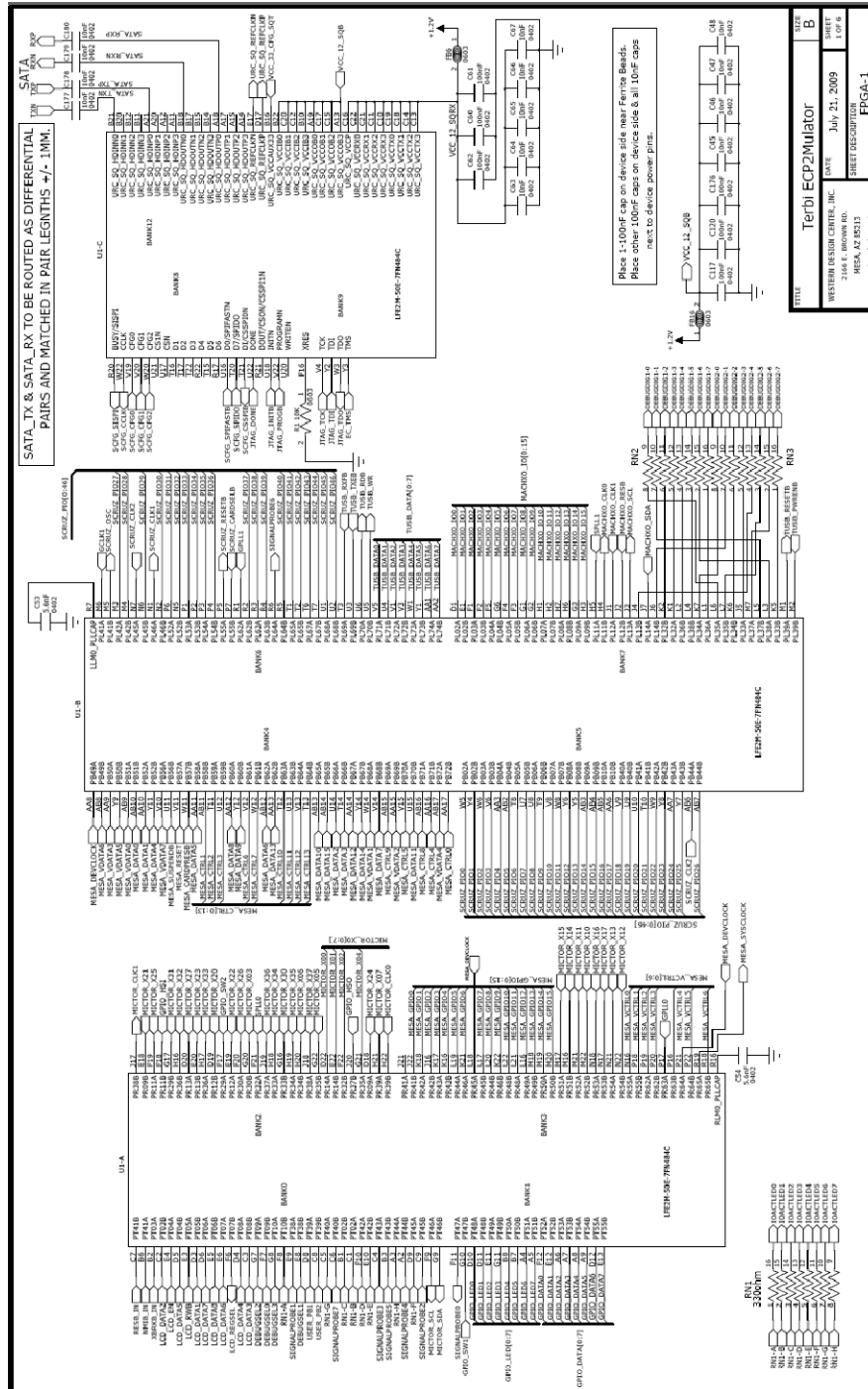
- 48 k Look-Up Tables (LUTs)
- 101 kbits of distributed RAM
- 4147 kbits of EBR SRAM
- 225 sysMEM blocks (18kb)
- 22 sysDSP blocks
- 88 18 x 18 multipliers
- 10 PLLs: 2 GPLLs, 6 SPLLs, 2 GDLLs
- 270 user I/Os
- DDR memory support (DDR1-400, DDR2-400)
- Supported I/O standards: LVCMOS, LVTTTL, SSTL, HSTL, LVDS, PCI, differential
- HSTL, differential SSTL, RSDS, Bus LVDS, MLVDS, LVPECL

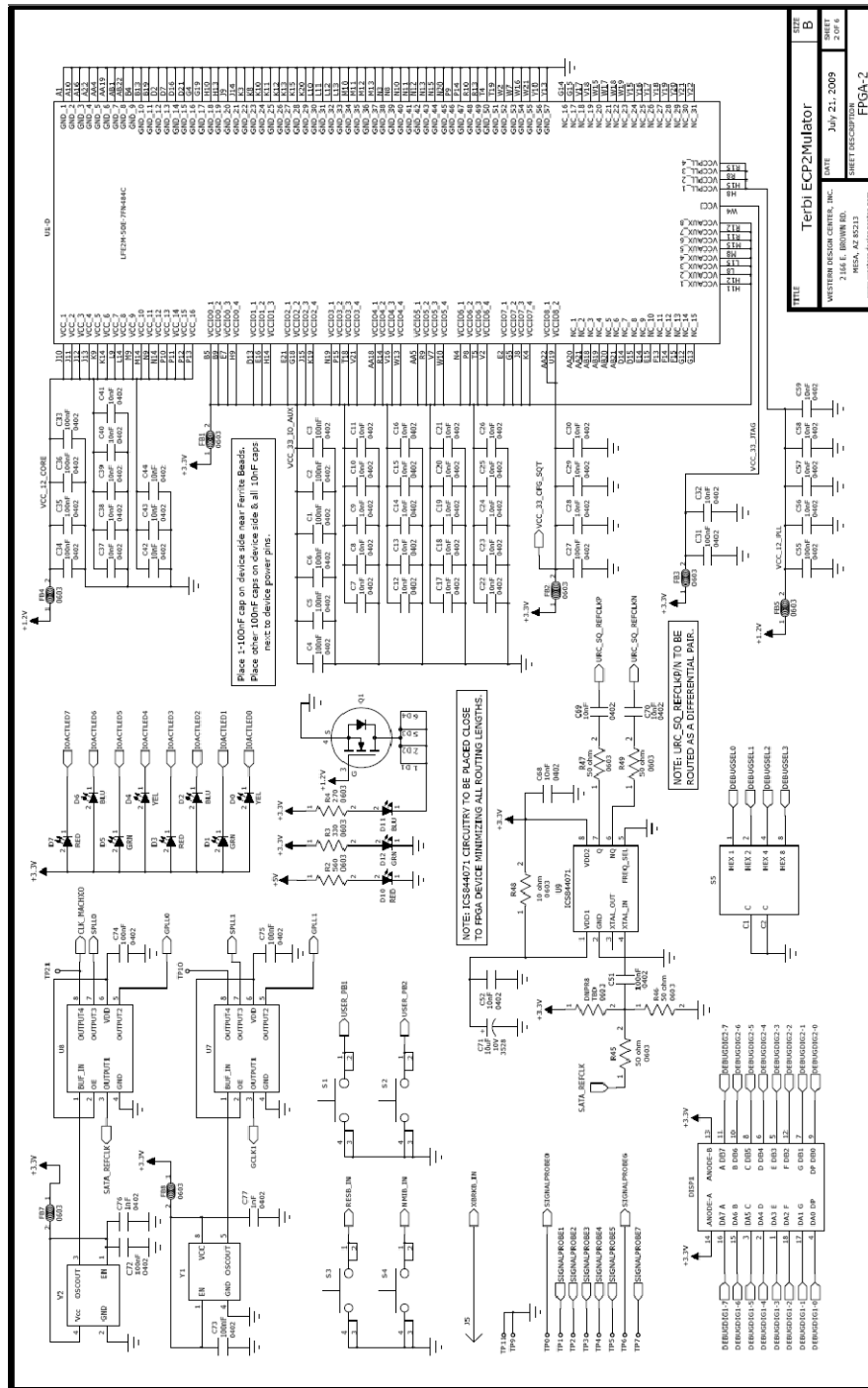
The ispLEVER design software can be used to develop/modify programs for the FPGA using Verilog or VHDL design entry methods. For more information on the ispLEVER software, see www.latticesemi.com/software.

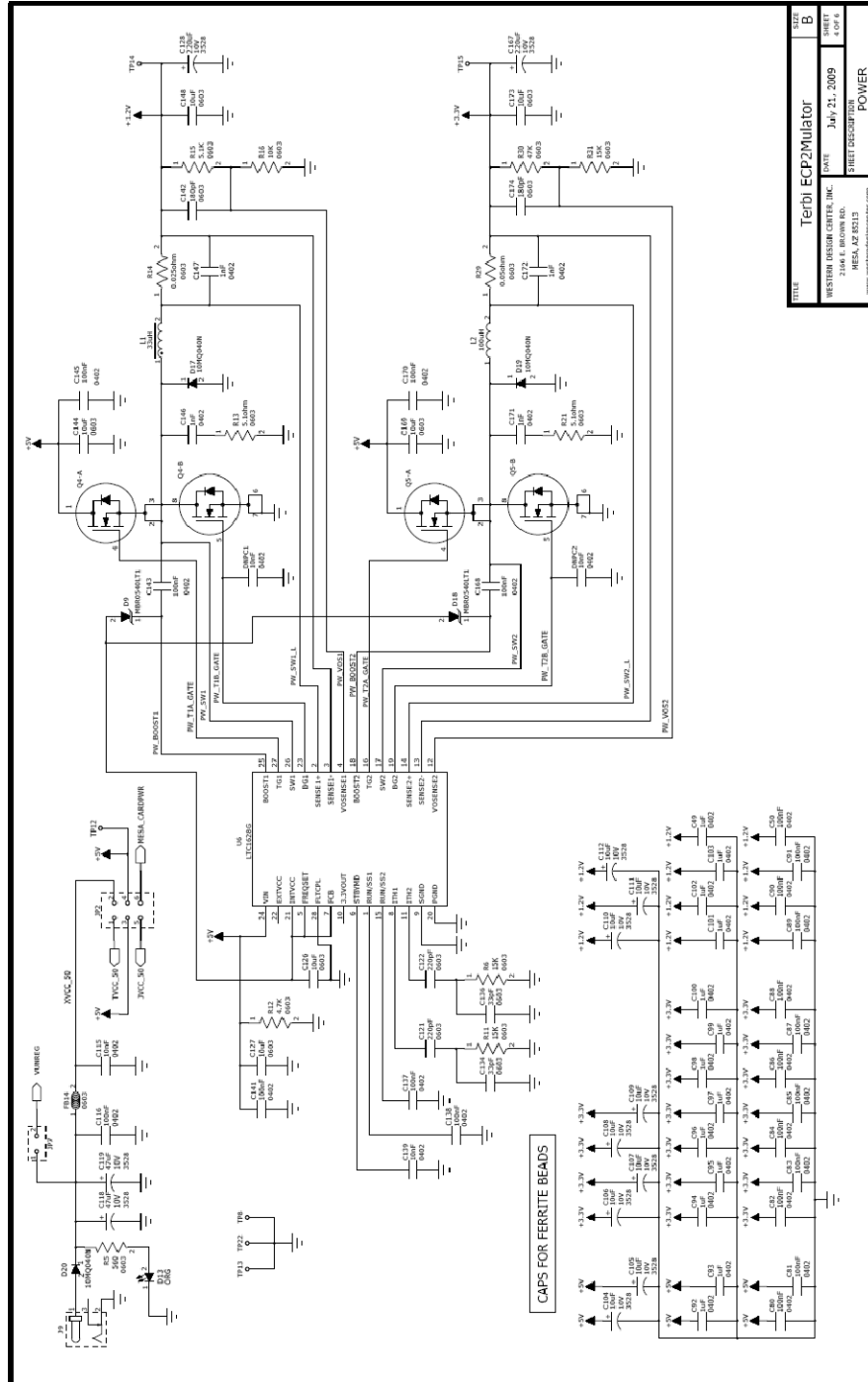
For further information please consult the LatticeECP2/M Family Data Sheet.

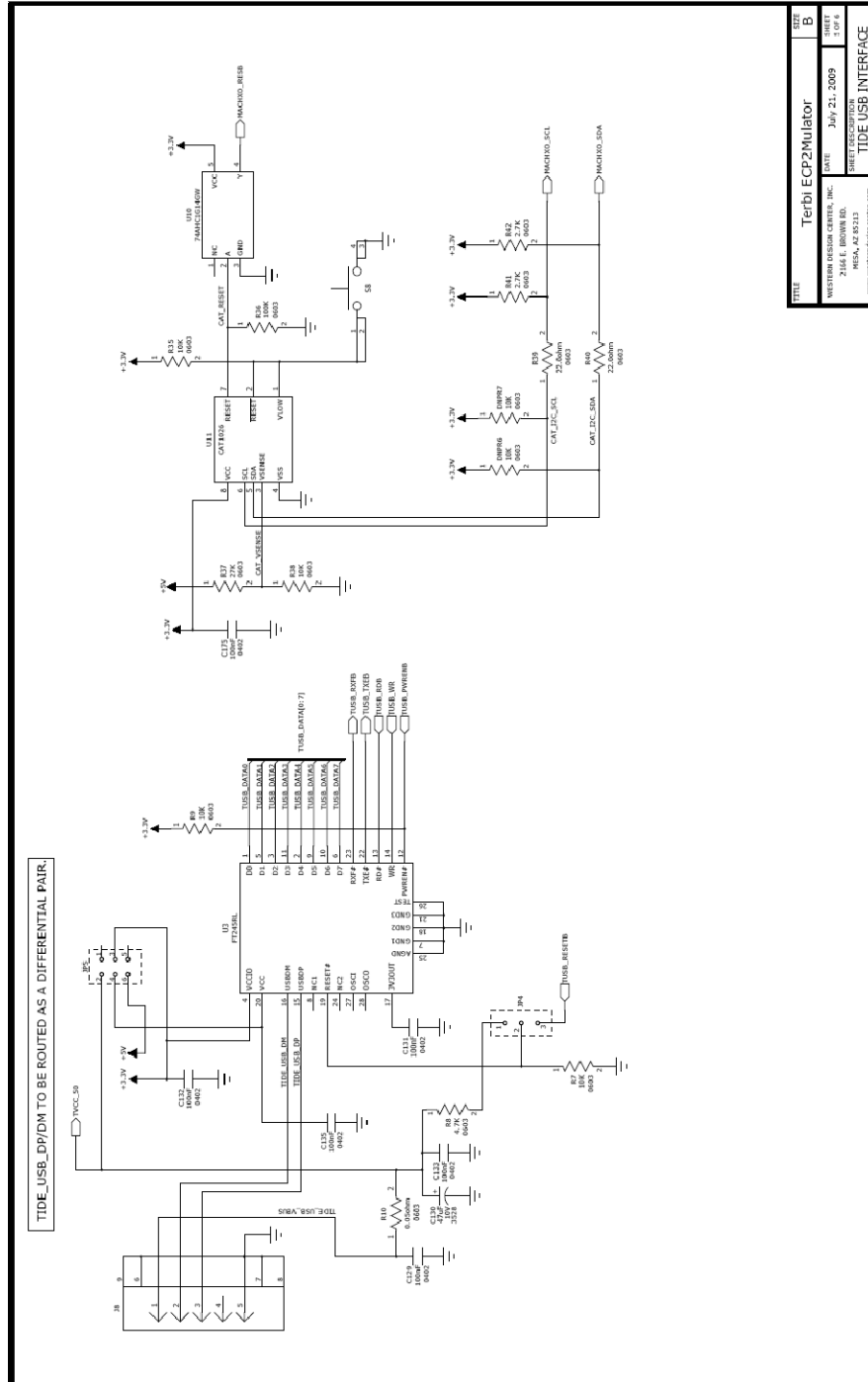
Appendix A. Schematics

Page 1/6

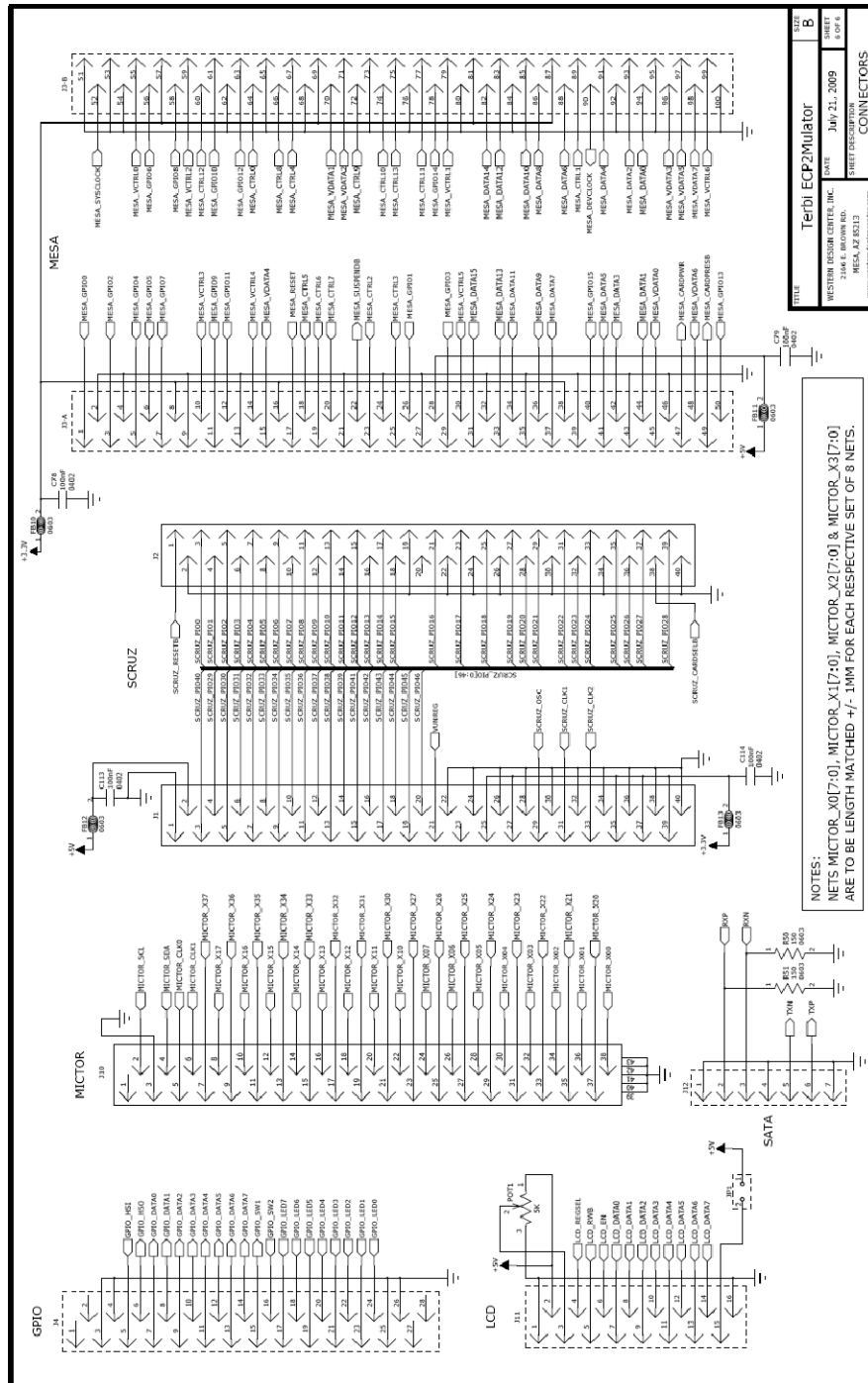








TITLE	Terbi ECP2Mulator	REV	B
WESTERN DESIGN CENTER, INC.	DATE	July 21, 2009	3 OF 6
2104 E. BROWN BLVD.	DRAWN DESCRIPTION		
www.westerndesigncenter.com	TIDE USB INTERFACE		



TITLE	Terbi-ECP2Mulator	REV	B
WESTERN DESIGN CENTER, INC.	DATE	July 21, 2009	SHEET
7000 WESTERN BLVD.	DESIGNED BY	WDC	9 OF 11
MESA, K2 E013	DRIFT DESCRIPTION		
www.westerndesigncenter.com	CONNECTORS		

NOTES:
NETS MICTOR_X01[7:0], MICTOR_X11[7:0], MICTOR_X21[7:0] & MICTOR_X31[7:0]
ARE TO BE LENGTH MATCHED +/- 1MM FOR EACH RESPECTIVE SET OF 8 NETS.



Appendix B. Example and Demo Code Projects

The Terbi Development Kit comes with 2 65xxProSDK projects of basic example code to interface to the W65C02SPMCU core. One project is in C code source, and the other is in Assembly code source.

To download this code into the Terbi-ECP2Mulator:

- Make sure that the W65C02SPMCU Lattice project has been programmed into the EPC2M FPGA (refer to the Lattice W65C02SPMCU build document)
- Make sure the Lattice build downloaded had the MON6502-USB.MEM file included (to allow USB interface with TIDE)
- That the TIDE USB cable is connected (not just the Lattice JTAG USB cable)
- Use TIDE from the 65xxProSDK tool suite to open and download the desired example/demo project. (Refer to the TIDE user manual for details)

Both example/Demo programs perform the same function. Below is a list of what this example code does:

- On reset:
 - Write a simple pattern to memory at 0x3000 starting with 0x52 (ASCII 'R')
 - Display "65" on 2 digit LED display
- On NMI:
 - Write 0x4E (ASCII 'N') to 0x3000
 - Display "E1" on 2 digit LED display
- On IRQ:
 - Write 0x49 (ASCII "I") to 0x3000
 - Display "E2" on 2 digit LED display
- Turn on/off LEDs in an animated "scan back/forth" pattern, repeating every 5 seconds
- If USER1 pushbutton is pressed:
 - Pause LED animation described above
 - Display current setting of Rotary switch as a single hex digit on the LED display, and the bitwise NOT as the other hex digit
 - Each time the USER1 pushbutton is pressed again, advance the LED scan animation to the next step
- If USER2 pushbutton is pressed:
 - Show "65" on LED display again
 - Continue LED scan animation at last rate
 - Each time the USER2 pushbutton is pressed again, speed up the LED animation rate from 5.0 seconds, to 2.5, to 1.0, to 0.5, then back to 5.0 (change speed with each press of USER2 pushbutton)
- Both USER1 and USER2 pushbuttons are de-bounced by 25ms in software

Listing 1 – C-TerbiDemo Source code

<will be added when code is complete and tested>

Listing 2 – A-TerbiDemo Source code

<will be added when code is complete and tested>