iCEman40™ iCE40HX8K Evaluation Kit User Guide

November 7, 2012 (1.0) User Guide

- **High-performance, low-power iCE40HX8K FPGA**
  - 7,680 logic cells, each with its own flip-flop
  - 32 RAM blocks, each containing 4Kbits; 128Kbits total RAM. Each has independent read and write ports.
  - 206 user I/O pins supporting multiple I/O standards
  - Two independent phase-locked loops (PLLs) for clock generation, phase alignment, etc.
- **USB programming, debugging, and virtual I/O functions**
- **27.0 MHz and 32.768 kHz clock sources**
- **8Mbit SPI serial configuration PROM supports multiple bitstream files**
- **Four user LEDs**
- **Two user slide switches and pushbutton switch**
- **International-compatible 5V DC wall adapter**
- **Supported by Lattice iCEcube2 design software**
- **Multi-voltage digital I/O connections with selectable 3.3V, 2.5V, or 1.8V I/O standards**
  - Four 40-pin male headers with 0.1" spacing (2x20)
  - Seven 12-pin Pmod-compatible sockets with 0.1" spacing (2x6)
- **Five differential input pairs and five differential output pairs**
- **Supports available, affordable third-party I/O expansion boards and modules**

Introduction

Thank you for choosing the Lattice Semiconductor iCEman40™ HX8K Evaluation Kit.

This guide describes how to begin using the iCEman40 Evaluation Kit, an easy-to-use platform for rapidly prototyping designs using iCE40 FPGAs.

Software Requirements

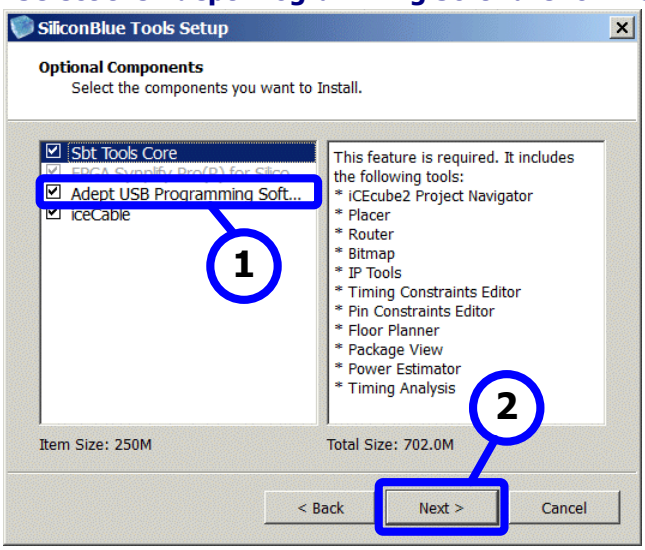
Before using the iCEman40 board, please be sure to download and install iCEcube2 Release 2012.06SP1 or later. This and later versions include the programming software for the iCEman40 board. Currently, the programming software is only available for the Windows operating system.

http://www.latticesemi.com/products/designsoftware/icecube2/downloads.cfm

During the installation process, be sure to install the Adept USB Programming Software, as shown in Figure 1.

1. Make sure that **`Adept USB Programming Software`** is checked. This is the default setting.
2. Click **`Next`**.

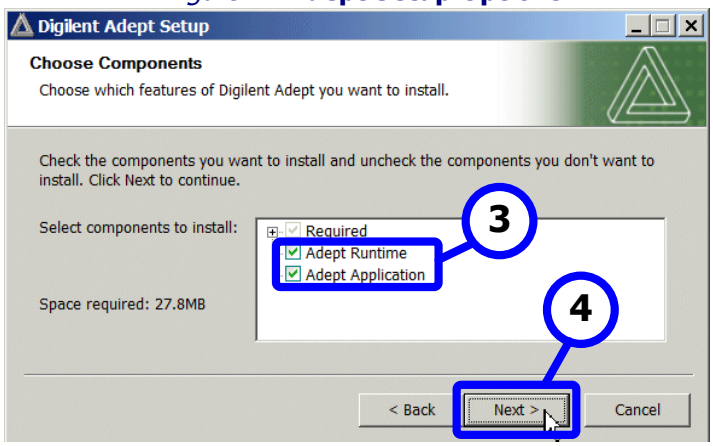
Figure 1: **Select the Adept Programming Software for Installation**



A few steps later, select the installation for the Adept programming software, as shown in Figure 2.

3. Make sure that both the **`Adept Runtime`** and **`Adept Application`** options are checked, which are the default settings.
4. Click **`Next`**.

Figure 2: **Adept Setup Options**



Connecting to the iCEman40 Evaluation Board for the First Time

Before connecting the iCEman40 board, be sure to download and install a supported version of the iCEcube2 software from the Lattice Semiconductor web site.

http://www.latticesemi.com/products/designsoftware/icecube2/downloads.cfm

Prepare the International-compatible AC Wall Adapter

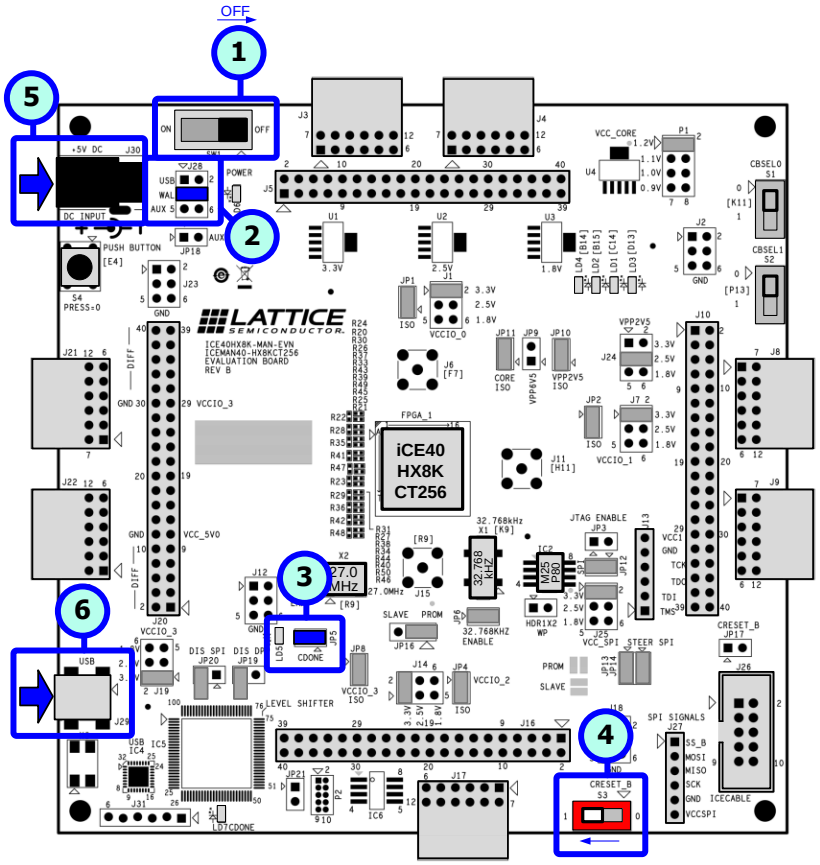
1 Remove the international-compatible AC wall adapter from the small cardboard box, including the plastic bag containing the plug ends.

2 From the plastic bag, select the plug end that matches the wall plugs used in your region.

3 Push the plug end into the AC wall adapter body.

Preparing the iCEman40 Board

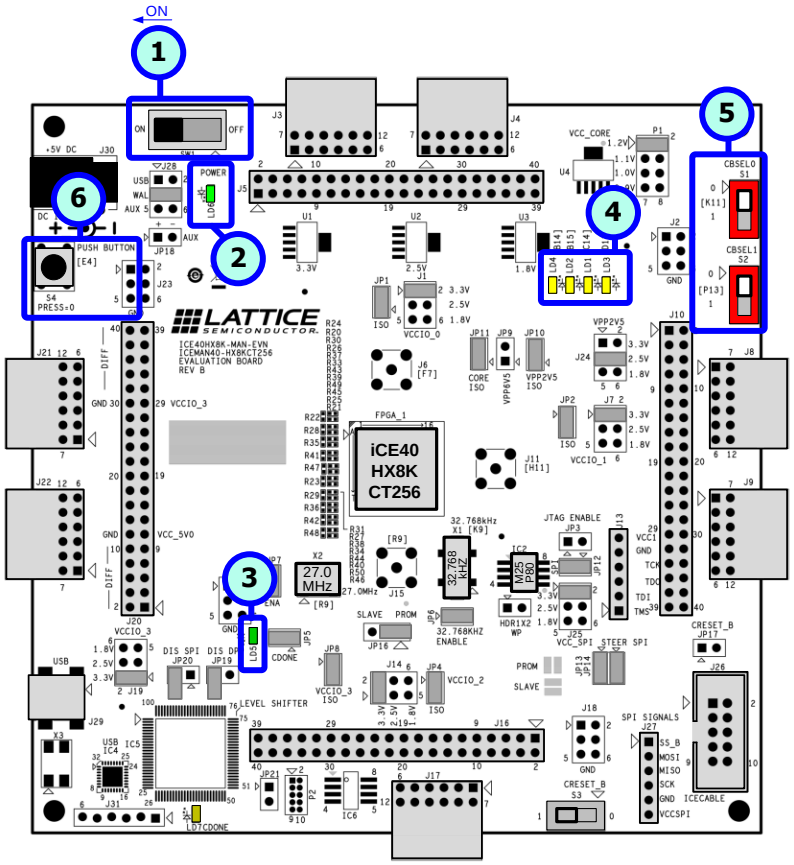
Figure 3: **Preparing the iCEman40 HX8K Board for First-time Use**



1. Turn off power to the board.
2. Set jumper header J28 to **WAL**, selecting the AC wall adapter as the primary power input.
3. Ensure that jumper JP5 must be installed. When inserted, this jumper allows the Configuration DONE (CDONE) LED, LD5, to light when the iCE40HX8K FPGA is properly configured.
4. Ensure that the Configuration Reset (CRESET_B) switch is in the 'I' position.
5. Connect the AC wall adapter to the power input.
6. Connect the included mini-USB cable to the computer where iCEcube2 is installed. This step is optional.

Powering On the iCEman40 Board, Configuring the FPGA

Figure 4: **Powering On the iCEman40 Board, Configuring the FPGA**



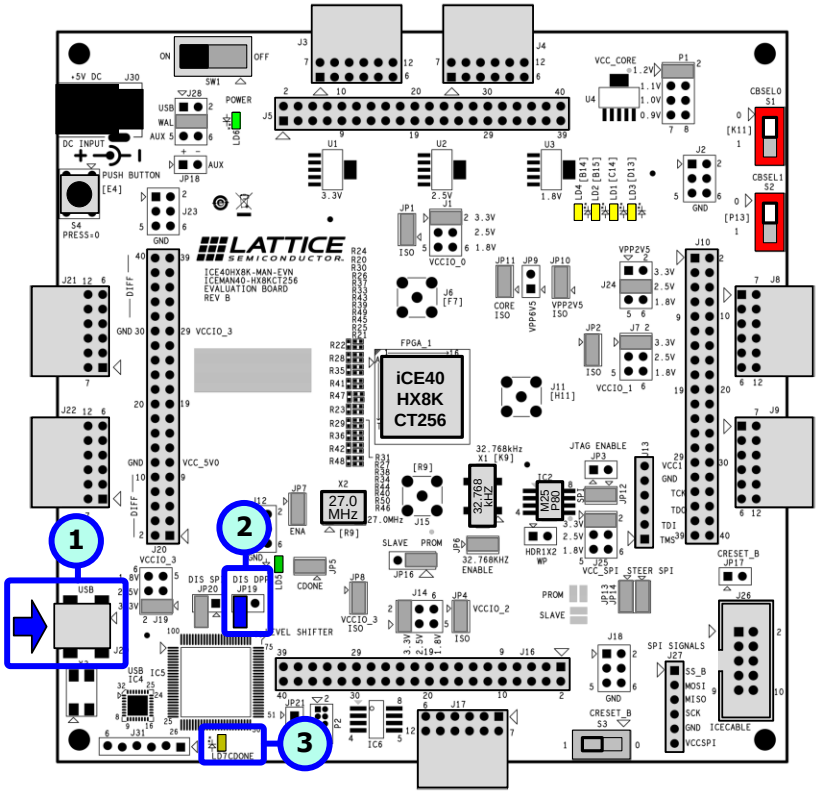
1. Re-apply power to the board.
2. The power-good LED, LD6 will light. If it does not, double-check that power is connected to the AC power input and that header J28 is set to **WAL**.
3. Check that the Configuration DONE (CDONE) LED is lit, indicating that the mobile FPGA is properly configured. The LED may be only dimly lit. If the LED is not lit, check that the adjacent jumper JP5 is installed. The iCEman40 board is shipped with a default application. If the CDONE LED is still not lit, then the FPGA's configuration PROM needs to be re-programmed.
4. The four user LEDs will light and will scroll to the left or to the right, depending on various switch settings.
5. Changing the settings on the two user switches, S1 and S2, will change the scrolling direction on the user LEDs.
6. Pressing the user push-button switch, S4, also changes the scrolling direction on the user LEDs.

The demonstration application is available for download from the Lattice Semiconductor web site.

Using the USB Virtual I/O and Debugging Interface

Preparing the USB Interface

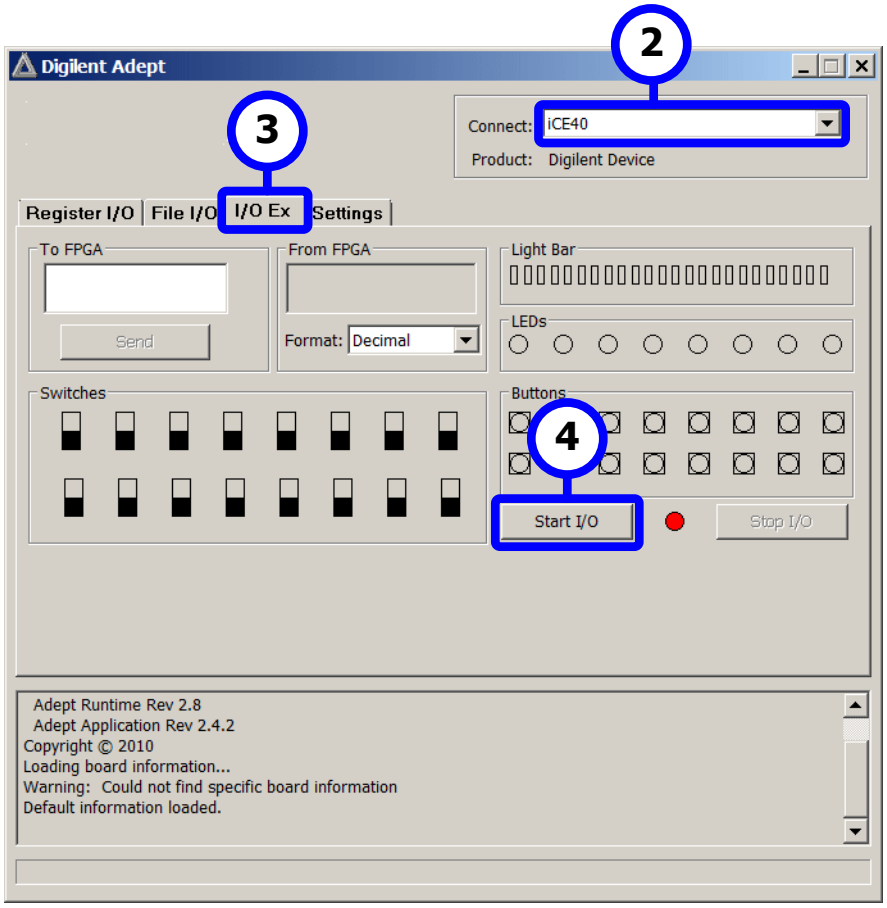
Figure 5: **Preparing the USB Virtual I/O and Debugging Interface**



1. To use the USB virtual I/O and debugging interface, the board must be connected to a computer running the iCEcube2 software via the included mini-USB cable.
2. Ensure that the Disable Digilent Parallel Interface (DPI) jumper, JP19, is disconnected or removed.
3. A Lattice Semiconductor iCE40HX1K FPGA provides voltage translation between the USB controller and the target iCE40HX8K FPGA. The iCE40HX1K FPGA must be properly configured for the USB interface to operate. Observe that the adjacent CDONE LED, LD7, is lit. The LED may be dimly lit.

Starting Digilent Adept

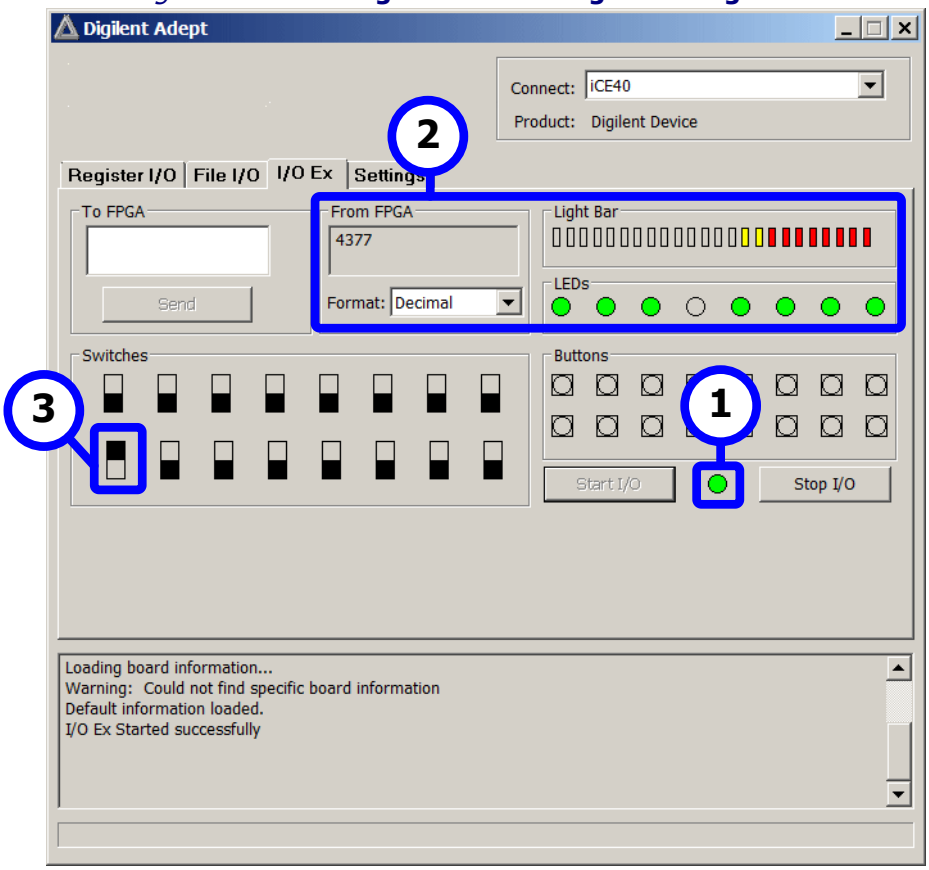
Figure 6: **Connecting the USB-based Digilent Adept Interface to the FPGA Application**



1. From the Windows Start menu, select **Start** → All Program → Digilent → Adept → Adept
2. Ensure that the Adept interface connects to the **iCE40**.
3. Click the **I/O Ex** tab.
4. Click **Start I/O**.

Observing and Controlling FPGA Logic via USB

Figure 7: **Observing and Controlling FPGA Logic via USB**



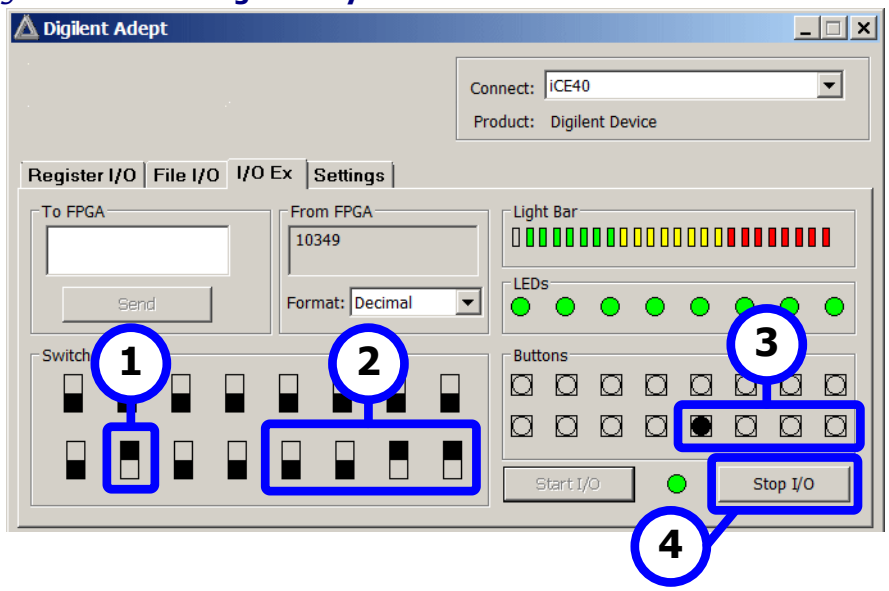
1 If the status indicated LED is green, the USB interface works. If it is yellow or red, then there is a failure in the USB-to-FPGA connection.

2 If the Light Bar is scrolling and the From FPGA text box is changing, then the 27.0 MHz oscillator (X2) is operating.

4 Change the position of bottom, left-most virtual slide switch. The scrolling direction of the physical LEDs on the board should change.

Controlling the Physical LEDs on the iCEman40 Board via USB

Figure 8: **Controlling the Physical LEDs on the iCEman40 Board via USB**



1. Switch the virtual slide switch up on the bottom rank, second from the left. This allows the virtual I/O interface to directly control the physical user LEDs on the iCEman40 board.
2. Change the four right-most virtual slide switches on the bottom rank to directly control the physical user LEDs on the iCEman40 board.
4. Press one of the four right-most virtual pushbutton switches on the bottom rank to directly invert the value of the associated physical user LEDs on the iCEman40 board.

Clocking Resources

The iCEman40 board includes two independent oscillators, as shown in Figure 9. Each has an independent enable control and supports all three voltages settings for I/O Bank 2.

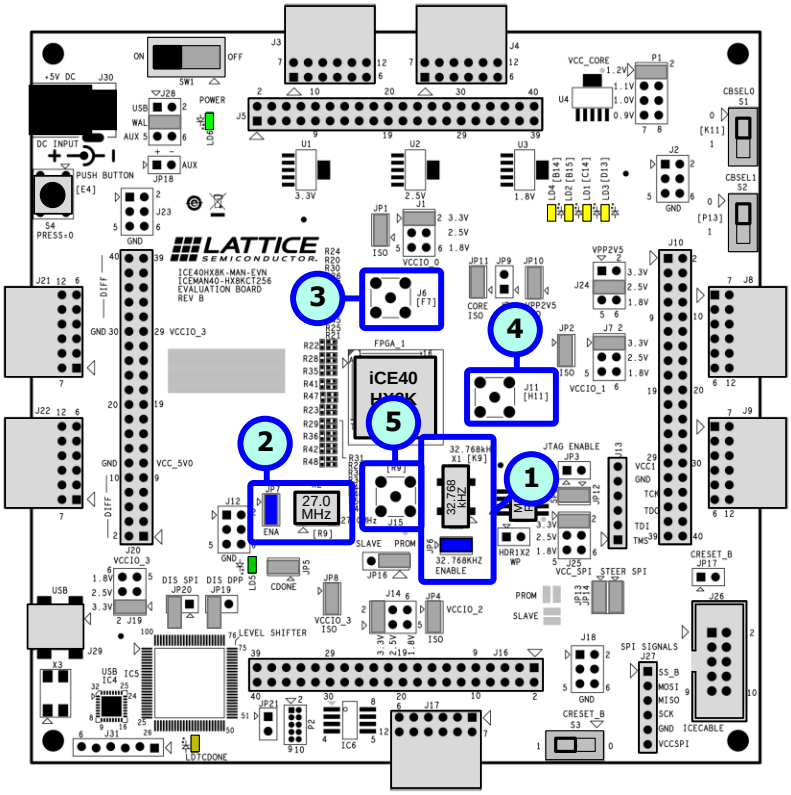
- 27.0 MHz oscillator
- 32.768 kHz oscillator

There are also three unstuffed SMA socket locations that connect directly to global buffer inputs on the FPGA.

Table 1: **Default Clock Sources and Pin Connections**

Clock Source	iCE40 Ball	Clock Supply Voltage	Enable Control	Comment
32.768 kHz Oscillator (X1)	K9	VCCIO_2	JP6	Connects to global buffer input GBIN4.
27.0 MHz Oscillator (X2)	R9	VCCIO_2	JP7	Connects to global buffer input GBIN5. In parallel with SMA connector J15.
SMA connector J6 (I/O Bank 0)	F7	VCCIO_0	None	Unstuffed socket position. Connects to global buffer input GBIN0.
SMA connector J11 (I/O Bank 1)	H11	VCCIO_1	None	Unstuffed socket position. Connects to global buffer input GBIN3
SMA connector J15 (I/O Bank 2)	R9	VCCIO_2	None	Unstuffed socket position. Connects to global buffer input GBIN5. If used as a clock input, disable the 27.0 oscillator (X2) by removing jumper JP7.

Figure 9: **iCEman40 HX8K Clock Sources and Controls**



1. 32.768 kHz oscillator (X1) and associated enable jumper control (JP6). Connects to FPGA ball K9.
2. 27.0 MHz oscillator (X2) and associated enable jumper control (JP7). Connects to FPGA ball R9.
3. Unstuffed SMA socket (J6). Connects to FPGA ball F7.
4. Unstuffed SMA socket (J11). Connects to FPGA ball H11.
5. Unstuffed SMA socket (J15). Connects to FPGA ball R5 in parallel with the 27.0 MHz oscillator (X2). Remove jumper JP7 to use J15 as a clock input.

32.768 kHz Oscillator

To demonstrate low-frequency operation, the iCEman40 evaluation board includes a 32.768 kHz oscillator (X1), sometimes also called a 32K oscillator. This oscillator is enabled when jumper JP6 is installed. The oscillator feeds ball K9 on the FPGA, which optionally feeds global buffer GBIN4, as described in Table 1. See also Figure 16.

To reclaim ball K9 as a user I/O, simply remove jumper JP6.

To generate a 1 second period, divide the 32.768 kHz oscillator output using a 15-bit binary counter. The output from bit Q15 generates a one second clock period with a 50% duty cycle.

27.0 MHz Oscillator

The iCEman40 evaluation board includes a 27.0 MHz oscillator (X2). The 27.0 MHz frequency was specifically chosen because it supports a wide range of graphic standards. The oscillator is enabled when jumper JP7 is installed. The oscillator feeds ball R9 on the FPGA, which optionally feeds global buffer GBIN5, as described in Table 1. See also Figure 16.

To reclaim ball R9 as a user I/O or to use the unpopulated SMA input J15, simply remove jumper JP7.

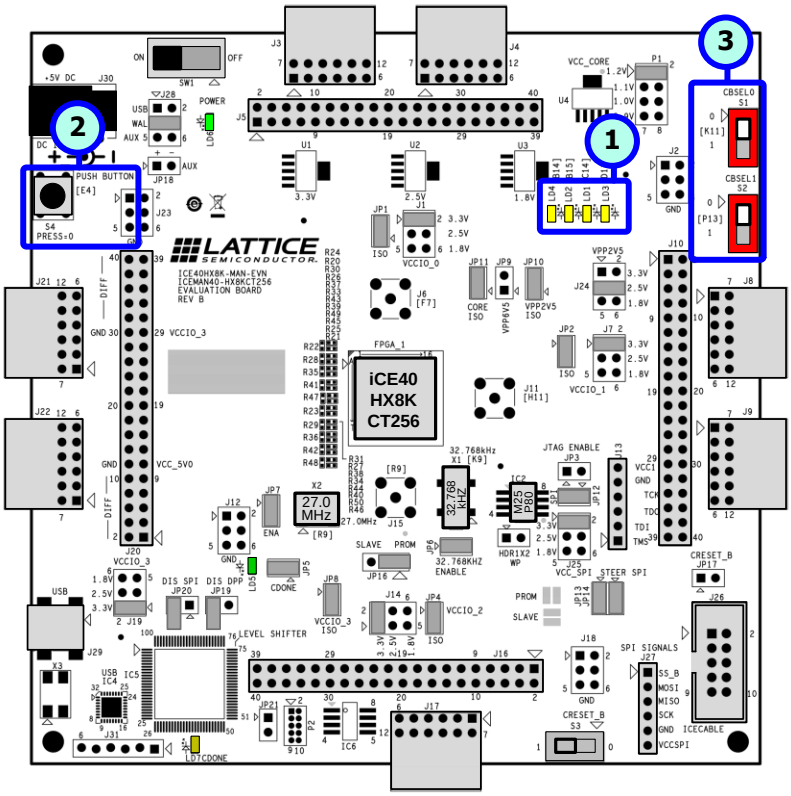
SMA Connector Locations

As shown in Figure 9, the iCEman40 has three unstuffed mounting locations for an SMA jack (straight-through, through-hole, 50 Ω). As shown in Table 1, these SMA locations potentially provide high-quality clock inputs directly to the global buffer inputs on the FPGA.

User LEDs

The iCEman40 evaluation kit board includes four user LEDs, located in the top-left corner of the board, as shown in Figure 10.

Figure 10: **Location of Physical User LEDs and User Switches**



1. Four user LEDs. Drive the associated FPGA pin High to light the LED.
2. Push-button switch. Normally High. Press to for FPGA ball E4 Low.
3. User slide switches. Optionally used as the CBSEL inputs for a FPGA application that supports ColdBoot.

Operation

To light a user LED, drive the associated FPGA pin High, as shown in Table 2. To darken the LED, drive the associated FPGA pin Low.

Table 2: **User LED Operation**

Operation	FPGA Action
Light LED	Drive High (1)
Turn Off LED	Drive Low (0)

The LEDs may appear to glow slightly before the FPGA is configured or if the FPGA pin is unused. This is because the FPGA I/Os have a soft pull-up resistor which may provide just enough current for the LED to glow dimly. To completely turn off an LED, drive it Low.

The LEDs are brightest when VCCO_0 is set to 3.3V via jumper block J1. The LEDs are dimmer at lower settings.

FPGA Connections

The FPGA drives the user LEDs using the FPGA pins listed in Table 3. See also Figure 14.

Table 3: **User LED Connections**

Designator	LD4	LD2	LD1	LD3
FPGA Pin	B14	B15	C14	D13

User Switches

The iCEman40-HX8K board includes a physical push-button momentary switch and two user slide switches, as shown in Figure 10.

Push-button Switch

The push-button switch, S4, is located along the left edge of the board, just below the AC power adapter input (item 2 in Figure 10). The switch connects to FPGA pin E4 found in I/O Bank 3. The operation of the push-button switch is described in Table 4. See also Figure 17.

Table 4: **User Push-button Switch Operation**

Push-button Switch Operation	FPGA Action
Unpressed	FPGA pin E4 pulled High (1) via pull-up resistor to VCCIO_2. A user application in the FPGA can drive pin E4 High or Low.
Pressed	Switch connects FPGA pin E4 to ground (0).

Slide Switches

The iCEman40-HX8k board includes two physical slide switches, S1 and S2, located along the right edge of the board (item 2 in Figure 10). Switch S1 is the top-most switch; switch S2 is the bottom-most switch. The operation of the each slide switch is described in Table 5. See also Figure 14.

Table 5: **User Slide Switch Operation**

Slide Switch Operation	FPGA Action
Up Position (↑)	Switch connects FPGA pin to ground (0).
Down Positon (↓)	FPGA pin pulled High (1) via pull-up resistor to VCCIO_2. A user application in the FPGA can drive pin High or Low.

Table 6 lists the pin connections between the slide switches and the FPGA. These slide switches can also be used to select one of four FPGA configuration images in the SPI PROM if the FPGA application is so design. The default design does not use the ColdBoot option.

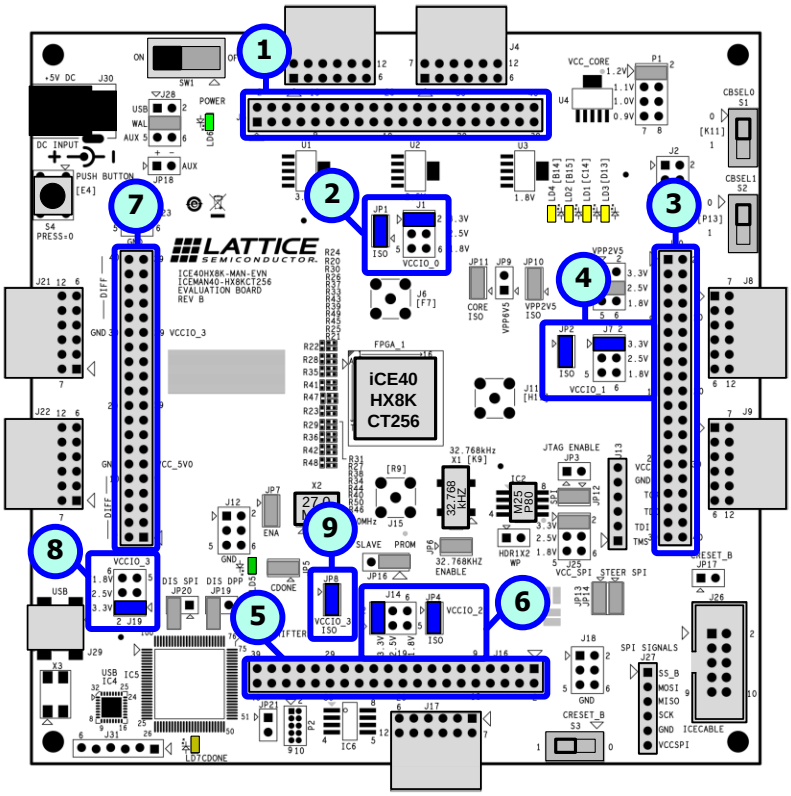
Table 6: **User Slide Switch Connections, ColdBoot Function**

Slide Switch Operation	FPGA Connection	Slide Switch Location	ColdBoot Select Function
S1	K11	Top-most	CBSEL0
S2	P13	Bottom-most	CBSEL1

40-Pin Headers

The iCEman40 board has four 40-pin ribbon cable headers, located on the top, right, bottom and left edges of the board. As shown in Table 7, each header has an independent voltage supply, controlled by a jumper selection.

Figure 11: **40-pin Headers and Voltage Controls on iCEman40 HX8K**



1. 40-pin header, J5, associated with I/O Bank 0. See also Figure 14.
2. Voltage control header for I/O Bank 0, J1 and voltage isolation jumper for I/O Bank 0, JP1.
3. 40-pin header, J10, associated with I/O Bank 1. See also Figure 15.
4. Voltage control header for I/O Bank 1, J7 and voltage isolation jumper for I/O Bank 1, JP2.
5. 40-pin header, J16, associated with I/O Bank 2. See also Figure 16.
6. Voltage control header for I/O Bank 2, J14 and voltage isolation jumper for I/O Bank 2, JP4.
7. 40-pin header, J20, associated with I/O Bank 3. See also Figure 17.
8. Voltage control header for I/O Bank 3, J19.
9. Voltage isolation jumper for I/O Bank 3, JP8.

Table 7: **40-pin Ribbon Cable Headers**

Header Designator	Header Designator	I/O Bank Voltage Source	Voltage Control Jumper	FPGA I/O Bank Voltage Isolation Jumper	Pinout Reference
J5	Top Edge	I/O Bank 0 (VCCIO_0)	Jumper block J1	Jumper JP1	Figure 14
J10	Right Edge	I/O Bank 1 (VCCIO_1)	Jumper block J7	Jumper JP2	Figure 15
J16	Bottom Edge	I/O Bank 2 (VCCIO_2)	Jumper block J14	Jumper JP4	Figure 16
J20	Right Edge	I/O Bank 3 (VCCIO_3)	Jumper block J19	Jumper JP8	Figure 17

Compatible Cabling

The 40-pin headers support IDC flat-ribbon assemblies with 0.1-inch pitch at least on socket connector types. The connectors are arranged as two rows of 20 connectors each (2x20). Both direct and twisted-pair versions of these cables are available from a variety of vendors.

- 2x20 Ribbon Cable, 0.1" Spacing
- IDE Hard Disk Drive (HDD) Cable

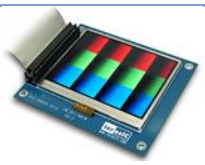
Compatible Third-Party Boards

The 40-pin header is designed to work with the following third-party products. These products require that the iCEman40 board be powered by an external AC adapter or from USB. The products listed in Table 8 connect to the iCEman40 board using a 40-wire ribbon cable included with the kits.

These products have not yet been verified to work with the iCEman40 board.

Lattice Semiconductor makes no warranty or guarantee of compatibility for these third-party products.

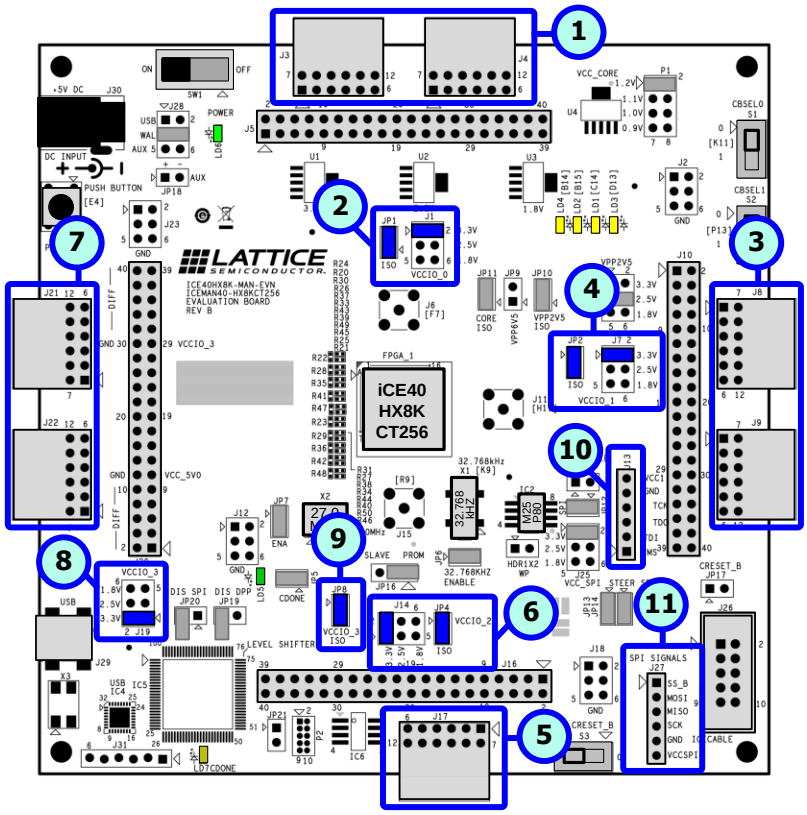
Table 8: **Available Expansion Boards for 40-Pin Header**

Board	Description, Link to More Information
	1.3 Megapixel Digital Camera Module www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=39&No=50
	5 Mega Pixel Digital Camera Package www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=68&No=281
	3.6-inch Digital Panel Kit www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=39&No=78
	4.3-inch Ultra-High Resolution LCD Touch Panel Kit www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=39&No=213

Pmod Peripheral Module Sockets

The iCEman40 board includes seven 12-pin, Pmod-compatible, peripheral module sockets located on all four sides, as shown in Figure 12. As shown in Table 9, each socket is associated with a particular I/O bank and has an associated voltage select jumper. As described on page 16, these sockets support a variety of affordable third-party add-on boards. These Pmod headers are also used on the iCEblink40 HX1K and iCEblink40LP1K boards.

Figure 12: **Pmod Peripheral Modules**



1. A double-wide Pmod socket or two Pmod12 sockets associated with I/O Bank 0, J3 and J4. See also Figure 14.
2. Voltage control header for I/O Bank 0, J1 and voltage isolation jumper for I/O Bank 0, JP1.
3. A double-wide Pmod socket or two Pmod12 sockets associated with I/O Bank 1, J8 and J9. See also Figure 15.
4. Voltage control header for I/O Bank 1, J7 and voltage isolation jumper for I/O Bank 1, JP2.
5. A Pmod12 socket associated with I/O Bank 2, J17. See also Figure 16.
6. Voltage control header for I/O Bank 2, J14 and voltage isolation jumper for I/O Bank 2, JP4.
7. A double-wide Pmod socket or two Pmod12 sockets associated with I/O Bank 3, J21 and J22. See also Figure 17.
8. Voltage control header for I/O Bank 3, J19.
9. Voltage isolation jumper for I/O Bank 3, JP8.
10. A Pmod male header associated with I/O Bank 1, J13. This header attaches to four user-I/O pins by default. This header can also be used to connect to the FPGA's JTAG interface.
11. A Pmod male header associated with the SPI Bank, J27. This header attaches to the four SPI signals on the mobile FPGA. The pins can be used to monitor SPI interface or can be used as user-I/O pins after configuration.

Table 9: **Digital I/O Headers and Their Functions**

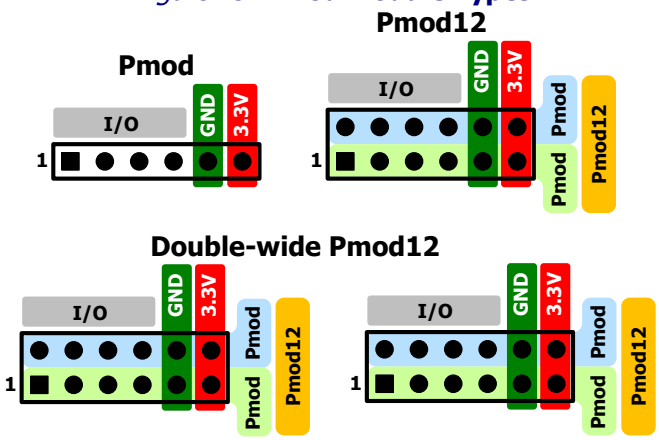
Header Designator	Header Type	Location	I/O Bank Voltage Source	Voltage Control Jumper	FPGA I/O Bank Voltage Isolation Jumper	Pinout Reference
J3	2x6 female Pmod12 socket, 0.1" pin centers	Top edge, left-most	I/O Bank 0 (VCCIO_0)	Jumper block J1	Jumper JP1	Figure 14
J4	2x6 female Pmod12 socket, 0.1" pin centers	Top edge, right-most	I/O Bank 0 (VCCIO_0)	Jumper block J1	Jumper JP1	Figure 14
J8	2x6 female Pmod12 socket, 0.1" pin centers	Right edge, top-most	I/O Bank 1 (VCCIO_1)	Jumper block J7	Jumper JP2	Figure 15
J9	2x6 female Pmod12 socket, 0.1" pin centers	Right edge, bottom-most	I/O Bank 1 (VCCIO_1)	Jumper block J7	Jumper JP2	Figure 15
J13	1x6 male Pmod header,0.1" pin centers	See Figure 12	I/O Bank 1 (VCCIO_1)	Jumper block J7	Jumper JP2	Figure 15
J17	2x6 female Pmod12 socket, 0.1" pin centers	Bottom edge	I/O Bank 2 (VCCIO_2)	Jumper block J14	Jumper JP4	Figure 16
J21	2x6 female Pmod12 socket, 0.1" pin centers	Left edge, top-most	I/O Bank 3 (VCCIO_3)	Jumper block J19	Jumper JP8	Figure 17
J22	2x6 female Pmod12 socket, 0.1" pin centers	Left edge, bottom-most	I/O Bank 3 (VCCIO_3)	Jumper block J19	Jumper JP8	Figure 17
J27	1x6 male Pmod header,0.1" pin centers	See Figure 12	SPI (VCC_SPI)	Jumper block J25	Jumper JP12	—

Supported Pmod Peripheral Modules

The iCEman40 board supports a variety of Pmod peripheral modules for easy I/O expansion. Pmod modules come in a few different form factors and each Pmod header includes power and ground supplies. Figure 13 shows the how the different Pmod form factors interrelate.

As shown in Figure 13, a Pmod module has six connections—four I/O plus power and ground. A Pmod12 module has 12 connections and the module is effectively two six-pin Pmod modules stacked together. Finally, a double-wide Pmod12 consists of two Pmod12 headers spaced apart. Most of the Pmod modules also include interface cables to allow easy connection to other header types.

Figure 13: **Pmod Module Types**



Pmod Add-on Boards

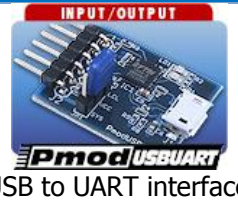
There are dozens of available, affordable, off-the-shelf Pmod peripheral boards that potentially operate with the iCEman40 board.

Digilent Inc.

Table 10 provides a sampling of the currently available Pmod modules from Digilent, Inc. Click the image to see more details on each module. For a complete list of Pmod peripheral modules, visit the Digilent web site.

http://www.digilentinc.com/Products/Catalog.cfm?NavPath=2,401&Cat=9

Table 10: **Sampling of Available Digilent Pmod Modules**

Displays				
 Large 256x64 OLED	 128x32 pixel OLED display	 16x2 character LCD, parallel interface	 16x2 character LCD, serial interface	 Two 7-segment LEDs
 Eight LEDs				
Wireless Communication				
 802.11b/g/n WiFi interface	 Bluetooth interface	 IEEE 802.15 radio transceiver	 Wireless radio transceiver	
Networking/Communication				
 10/100 Ethernet NIC	 USB to UART interface	 RS-232 interface	 PS/2 keyboard/mouse connector	

Sensors

 GPS Positioning Receiver	 3-axis digital gyroscope	 3-axis accelerometer	 Infrared light sensor	 Thermometer/ thermostat

Digital-to-Analog (D/A) and Analog-to-Digital (A/D) Conversion

 Resistor Ladder D/A Converter	 Four 8-bit D/A outputs	 Two 12-bit D/A outputs	 Two 12-bit A/D inputs	 Four-channel, 12-bit A/D inputs

Audio

 Mono, Class-D audio amplifier	 Microphone with digital interface	 I^2S stereo audio output	 Speaker/headphone amplifier	 RCA audio jacks

Keypads, buttons, switches, and joysticks

 16-button keypad	 Four pushbuttons	 Four slide switches	 Rotary encoder/switch	 Two-axis joystick

Connectors

 RJ45 connector pair	 BNC connectors	 Wire terminal connectors	 R/C servo control connectors	 Four digital inputs with diode protection, debounce filters
 H-bridge with feedback	 H-bridge with feedback	 Open-drain output	 Open-collector output	 I^2C I/O expansion module

Bread board and testpoint headers

 Wire wrap/bread board	 Pmod test header	 Pmod12 test header

Memory			Miscellaneous	
 SD card slot	 SPI Flash PROM	 x1, x2, x4 SPI Flash PROM	 Digital Potentiometer	 Real-time clock/calendar

Maxim Integrated

Table 11 provides a sampling of the currently available Pmod modules from Maximum Integrated. Click the image to see more details on each module. For a complete list of Pmod peripheral modules, visit the Maxim Integrated web site.

http://www.maximintegrated.com/products/evkits/fpga-modules/

Table 11: **Sampling of Available Maxim Integrated Pmod Modules**

Communication				
 RS-485 transceiver	 RS-232 transceiver	 Isolation module		
Sensors				
 Thermometer/ thermostat	 Thermocouple to digital converter	 Proximity Detector		
Digital-to-Analog (D/A) and Analog-to-Digital (A/D) Conversion				
 8-channel D/A Converter (DAC)	 Precision 16-bit D/A Converter (DAC)	 Precision 16-bit A/D Converter (ADC)		
Miscellaneous				
 Programmable Oscillator	 Real-time clock/calendar	 16-bit GPIO expander and LED driver	 Dual digital potentiometer	 8-channel relay driver
 Programmable current limit				

I/O Bank 0 Features and Pinout

Figure 14: **FPGA Pinout for Connections to I/O Bank 0**

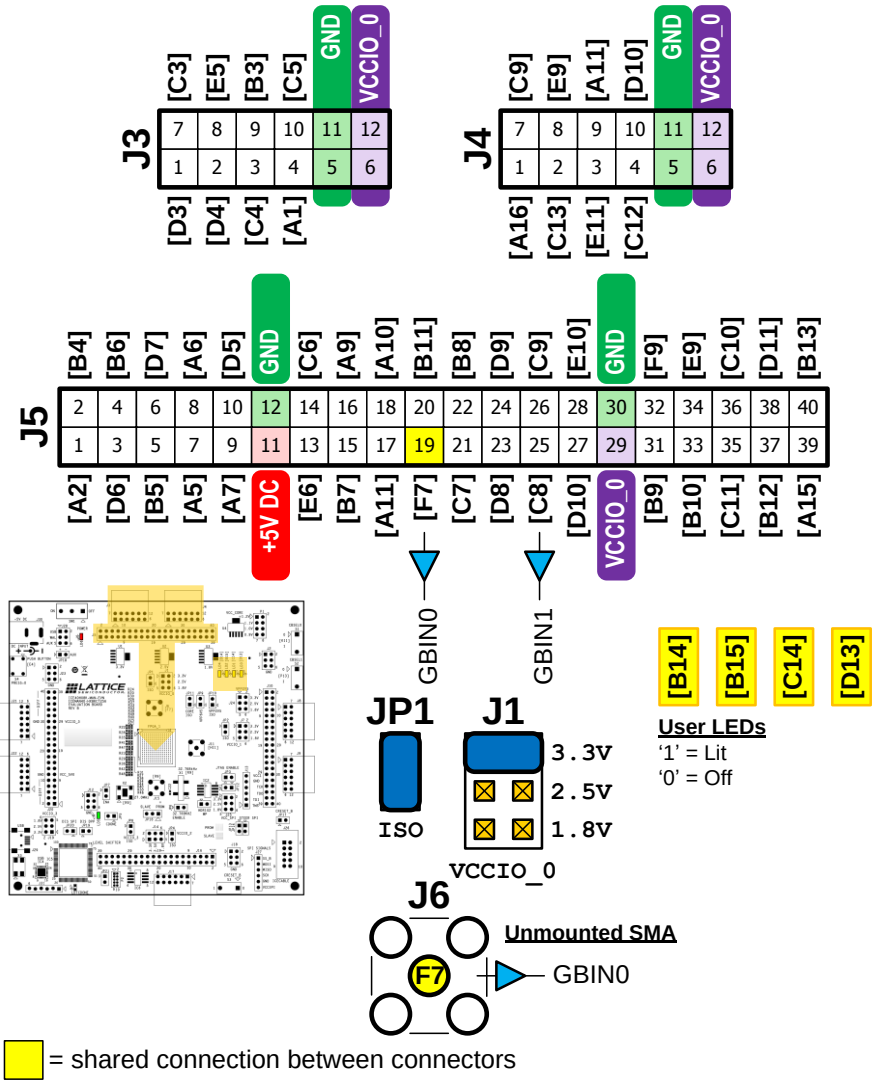


Table 12: **Major Features of I/O Bank 0 Connections**

Designation	Description
J3	Pmod12 socket. Also used with J4 as a double-wide Pmod socket.
J4	Pmod12 socket. Also used with J3 as a double-wide Pmod socket.
J5	40-pin header.
JP1	Power isolation jumper for measuring power in FPGA I/O Bank 0.
J1	Voltage select header for I/O Bank 0.
J6	Unstuffed SMA location.
LD1, LD2, LD3, LD4	User LEDs.

I/O Bank 1 Features and Pinout

Figure 15: **FPGA Pinout for Connections to I/O Bank 1**

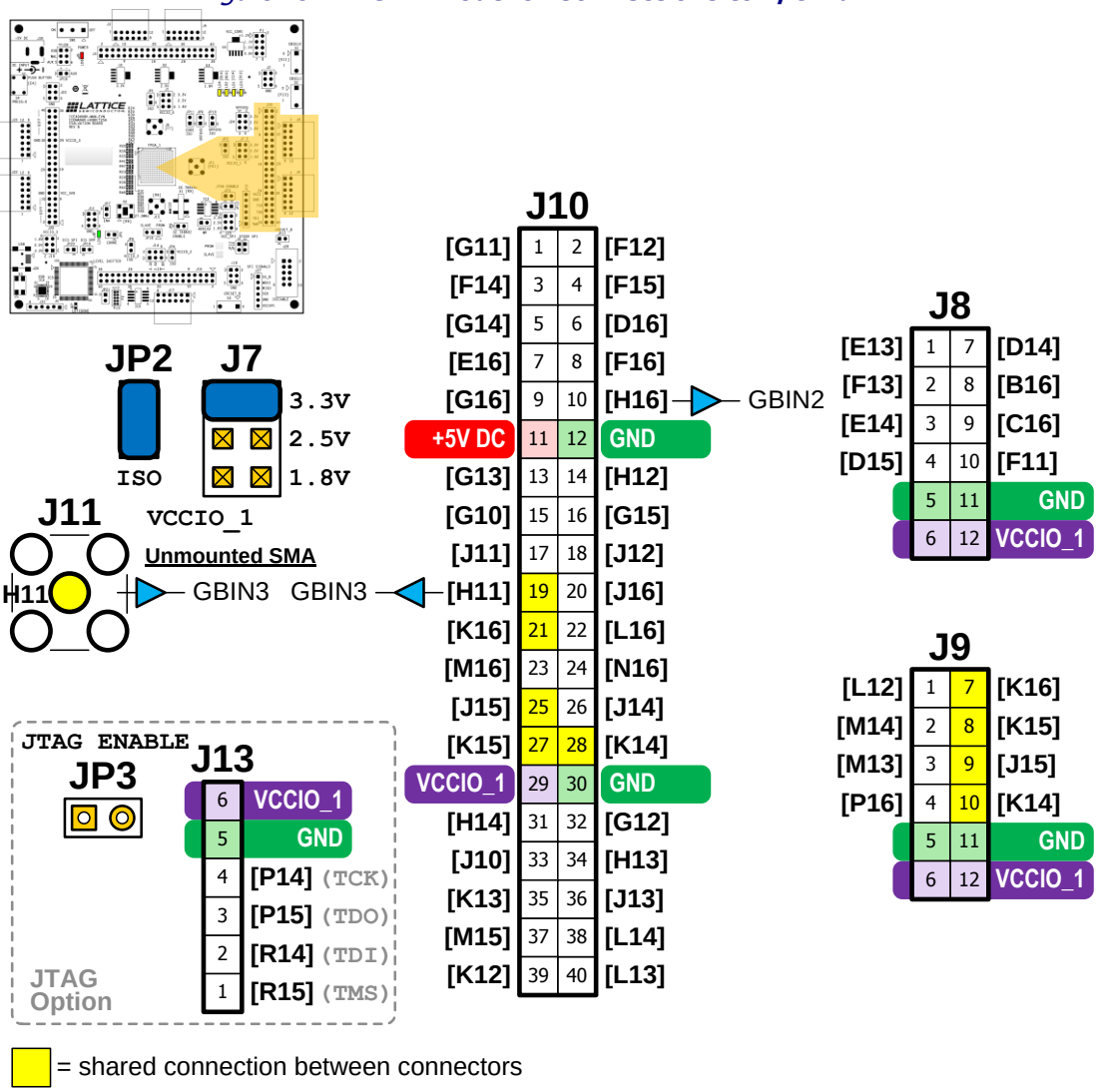


Table 13: **Major Features of I/O Bank 1 Connections**

Designation	Description
J8	Pmod12 socket. Also used with J9 as a double-wide Pmod socket.
J9	Pmod12 socket. Also used with J8 as a double-wide Pmod socket.
J10	40-pin header.
JP2	Power isolation jumper for measuring power in FPGA I/O Bank 1.
J7	Voltage select header for I/O Bank 1.
J11	Unstuffed SMA location.
JP3	Unstuffed 2-pin header for pulling TRST_N Low.
J13	6-pin Pmod male header. JTAG control signals by default. Short JP3 to reclaim these signals as user-I/O pins.

I/O Bank 2 Features and Pinout

Figure 16: **FPGA Pinout for Connections to I/O Bank 2**

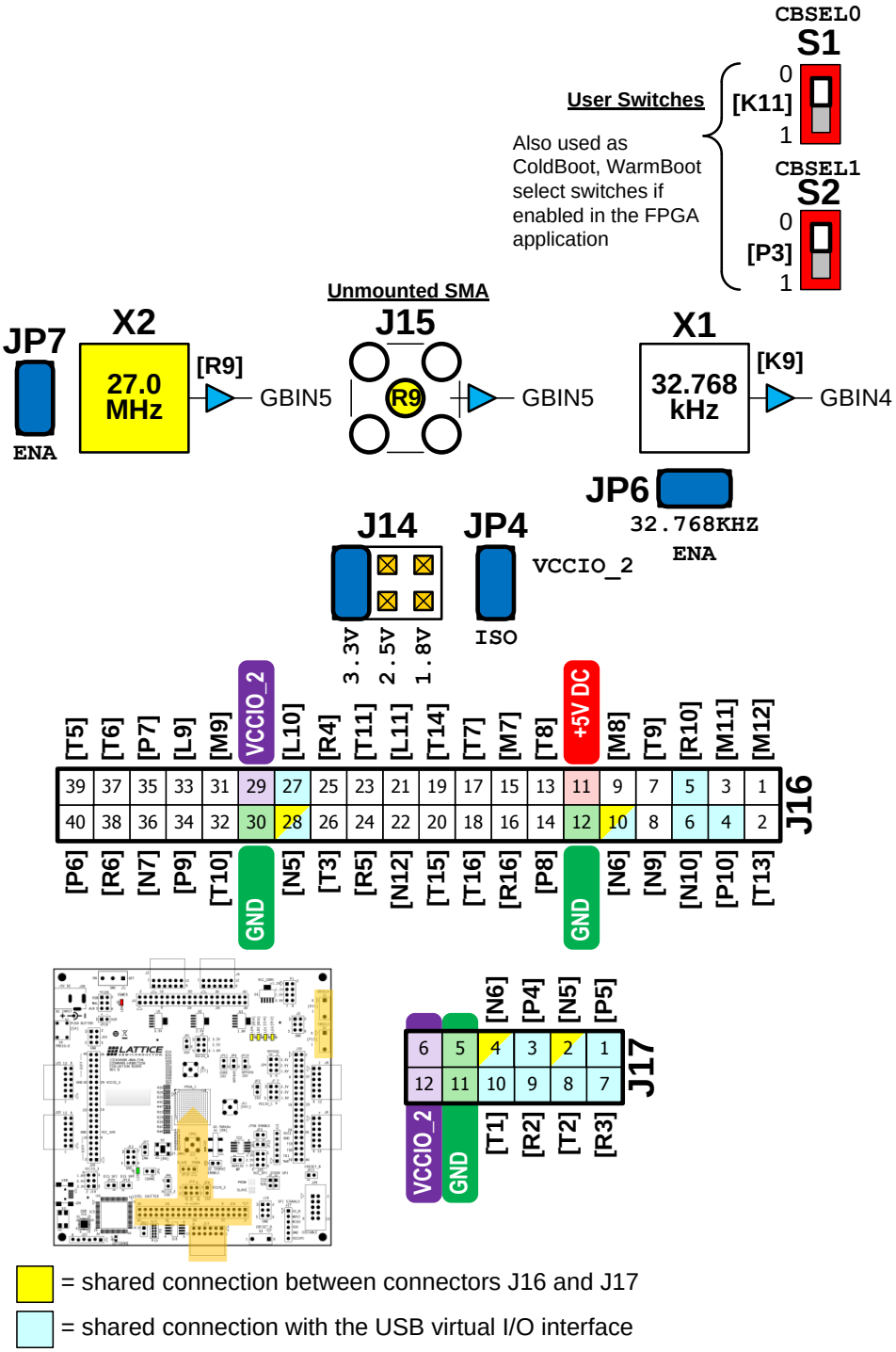


Table 14: **Major Features of I/O Bank 2 Connections**

Designation	Description
J17	Pmod12 socket. Shared with 40-pin header J16 and optional USB virtual I/O interface.
J16	40-pin header.
JP4	Power isolation jumper for measuring power in FPGA I/O Bank 2.
J14	Voltage select header for I/O Bank 2.
J15	Unstuffed SMA location.
X2	27.0 MHz oscillator.
JP7	Enables X2 oscillator when installed.
X1	32.768 kHz oscillator.
JP6	Enables X1 oscillator when installed.
S1	User slide switch. Optionally used as CBSEL0 during configuration in applications with ColdBoot enabled.
S2	User slide switch. Optionally used as CBSEL1 during configuration in applications with ColdBoot enabled.

I/O Bank 3 Features and Pinout

General Pinout

Figure 17: **FPGA Pinout for Connections to I/O Bank 3**

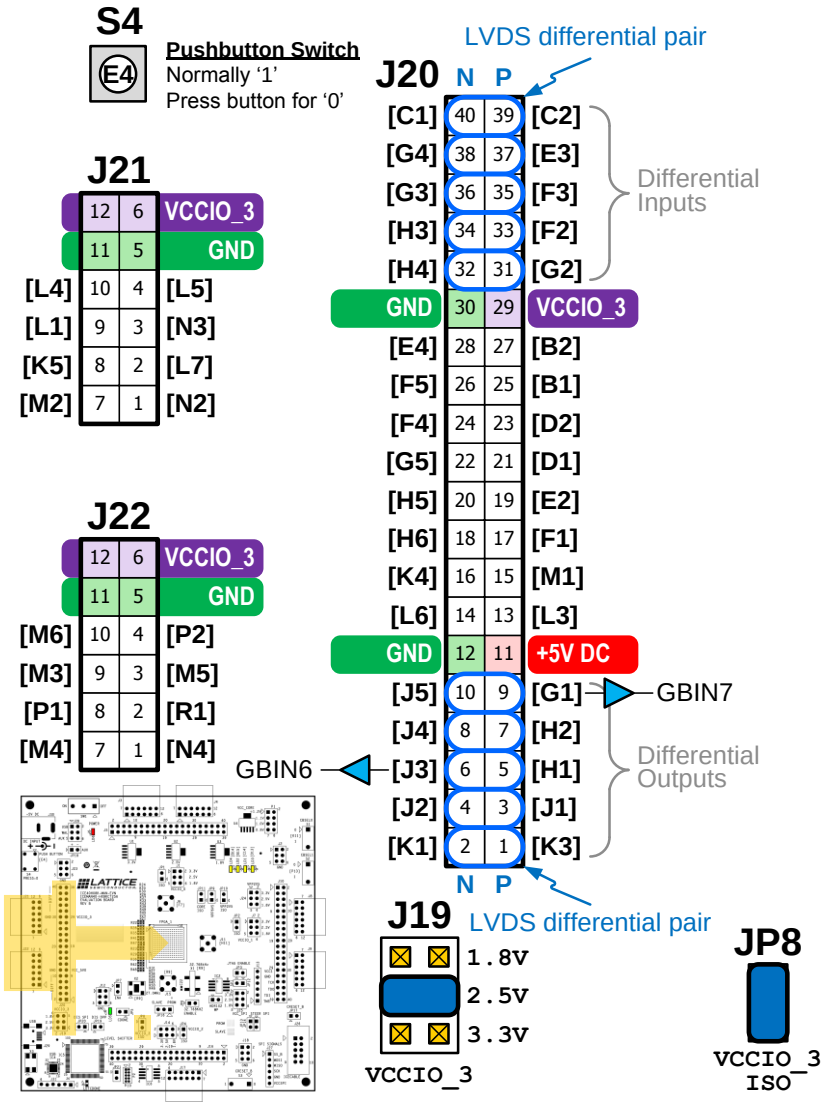


Table 15: **Major Features of I/O Bank 1 Connections**

Designation	Description
J21	Pmod12 socket. Also used with J22 as a double-wide Pmod socket.
J22	Pmod12 socket. Also used with J21 as a double-wide Pmod socket.
J20	40-pin header.
JP8	Power isolation jumper for measuring power in FPGA I/O Bank 2.
J19	Voltage select header for I/O Bank 2.
S4	User pushbutton switch.

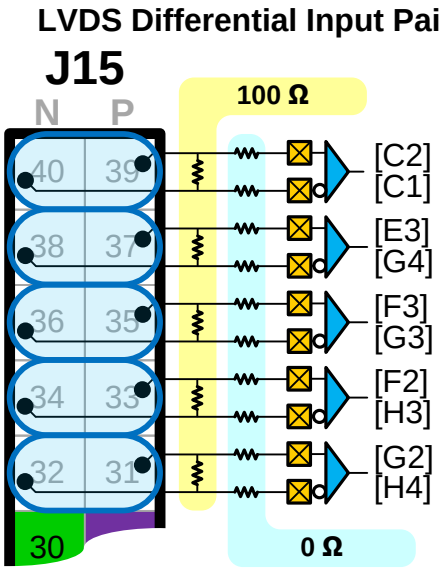
Differential Inputs

The connections on header J15, closest to the top of the board, are configured as five differential input pairs, as shown in Figure 18. The 100 Ω termination resistor supports LVPECL, LVDS, or SubLVDS inputs spending on the setting for header J19. The figure shows the FPGA balls for each differential pair.

Table 16: **Differential Input Standards and Voltage Settings**

Differential Input Standard	Header J19 Voltage Setting
LVDS	2.5V
SubLVDS	1.8V
LVPECL	3.3V

Figure 18: **LVDS Differential Input Pairs on 40-pin Header J15**



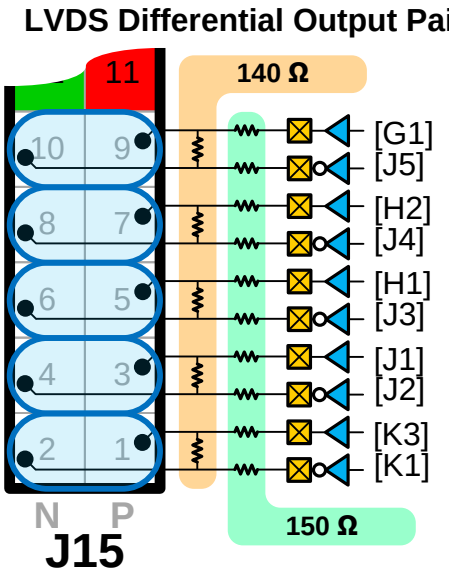
To learn more about using differential I/O in iCE40 FPGAs, see TN1253: "**Using Differential (LVDS, SubLVS) in iCE40 Devices**" available from the Lattice Semiconductor web site.
http://www.latticesemi.com/documents/TN1253.pdf

LVDS Outputs

The connections on header J15, closest to the top of the board, are configured as five differential input pairs, as shown in Figure 19. The external 150 Ω series and 140 Ω series output termination resistors general an LVDS signal level created using two single-ended outputs. The figure shows the FPGA balls for each differential pair.

Set jumper J19 to 2.5V to generate LVDS output signal levels.

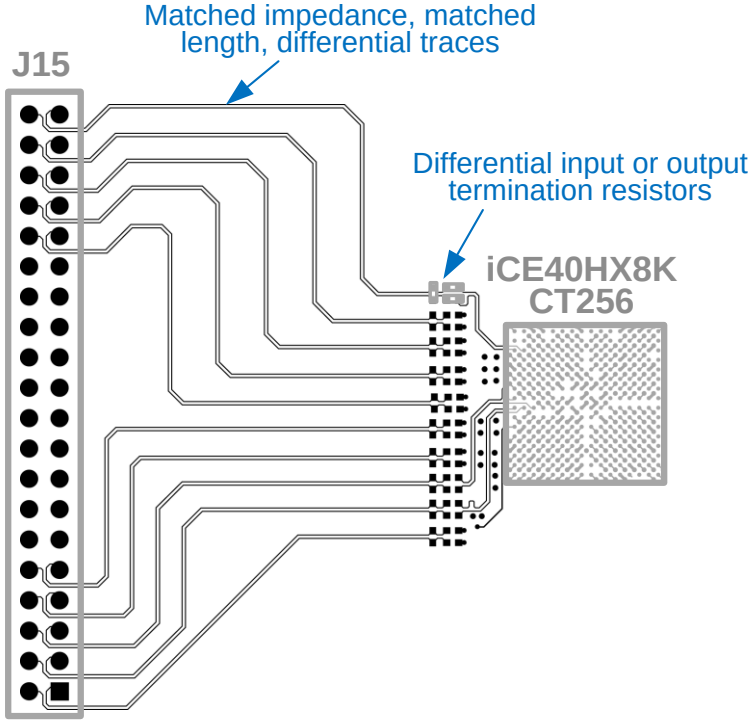
Figure 19: **LVDS Differential Output Pairs on 40-pin Header J15**



Differential Signal Routing

Figure 20 shows the differential signal routing used on the iCEman40 board. Each differential I/O pair is routed with matched impedance and matched trace length. Each differential input or output pair has an associated termination resistor set, which is located close to the iCE40 FPGA package.

Figure 20: **Differential Signal Routing on iCEman40 Board**



USB Interface

Connector

The USB connector to the board is located along the left edge of the board, toward the lower left corner, labeled J29. The board connects using a standard USB cable with a male mini-B connector.

SPI Flash Programming

The USB interface also provides Flash programming for the on-board SPI PROM, as described in "Program SPI Flash using On-board USB Programmer" on page 34.

Digilent Parallel Port (DPP)

The Digilent Parallel Port (DPP) interface is used for virtual I/O and debugging using a USB connection to the board from a Windows PC. See "Virtual I/O Expansion Debugging Interface" on page 28 for additional information.

Powering the iCEman40 Board via the USB Interface (WARNING!)

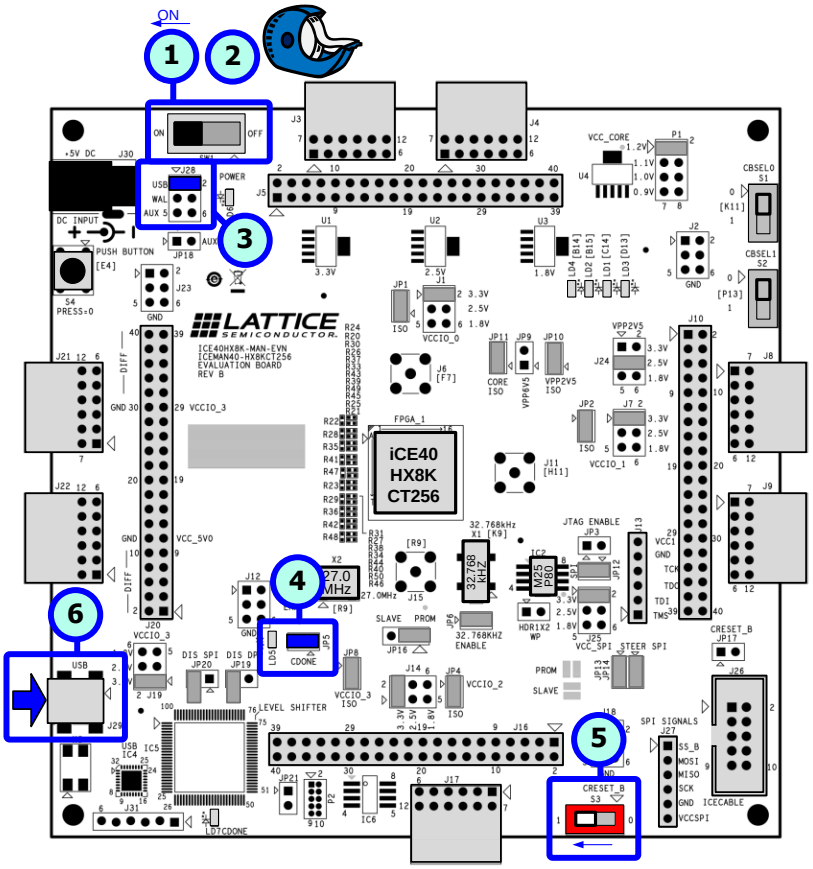
By default, the iCEman40 board is powered using the AC wall adapter included with the kit while the USB interface is primarily used for programming and debugging support.

However, with some limitations and care, the iCEm40 board can be powered directly from a computer USB port, a powered USB hub, or a USB-based AC adapter, commonly used in consumer electronics. A typical USB port provides up to 500 mA at 5V, providing up to 2.5W of total power. The iCE40HX8K consume SIGNIFICANTLY LESS than power, even when operating at full performance.

When powering the iCEman40 board from the USB connector, DO NOT use the power switch on the board.

ALWAYS CONNECT OR DISCCONECT POWER VIA THE USB CABLE!

Figure 21: **Powering the iCEman40 Board via the USB Connector**



1. Turn ON power to the board.
2. With the power switch in the ON position, place a piece of tape or a sticky label over the switch. This is to remind you to NOT to use the switch when powering the board from USB. Simply plug in or remove the USB cable to connect or remove power to the board.
3. Set jumper header J28 to **USB**, selecting the USB input as the primary power input.
4. Ensure that jumper JP5 must be installed. When inserted, this jumper allows the Configuration DONE (CDONE) LED, LD5, to light when the iCE40HX8K FPGA is properly configured.
5. Ensure that the Configuration Reset (CRESET_B) switch is in the 'I' position.
6. Connect the included mini-USB cable to the USB port on a computer, a powered USB hub, or a USB-based AC power adapter.

8Mbit SPI Configuration PROM

The configuration bitstream for the iCE40 FPGA is stored in an M25P80 8Mbit SPI serial Flash PROM. The PROM is large enough to hold four to five FPGA configuration images and supports the iCE40 WarmBoot feature, if so enabled within the FPGA application.

Virtual I/O Expansion Debugging Interface

The iCE40 USB interface provides a means to program the on-board SPI Flash for the FPGA. The USB connection also provides a convenient means to monitor and control logic inside the FPGA via a virtual I/O interface, as shown in Figure 22. The USB controller drives a byte-wide parallel port expander implemented within the FPGA, controlled by software running on the PC. The Digilent ADEPT2 I/O Expansion screen, shown in Figure 23, provides a mix of virtual switches, pushbuttons, LEDs, light bars, and 32-bit input and outputs.

The default FPGA shipped with the iCEman40 board includes a complete example. The virtual I/O interface connects to pins on the FPGA, as listed in Table 17. These connections are also shared with headers in I/O Bank 2.

Figure 22: **iCEman40 Board Supports Virtual I/O Connections over USB**

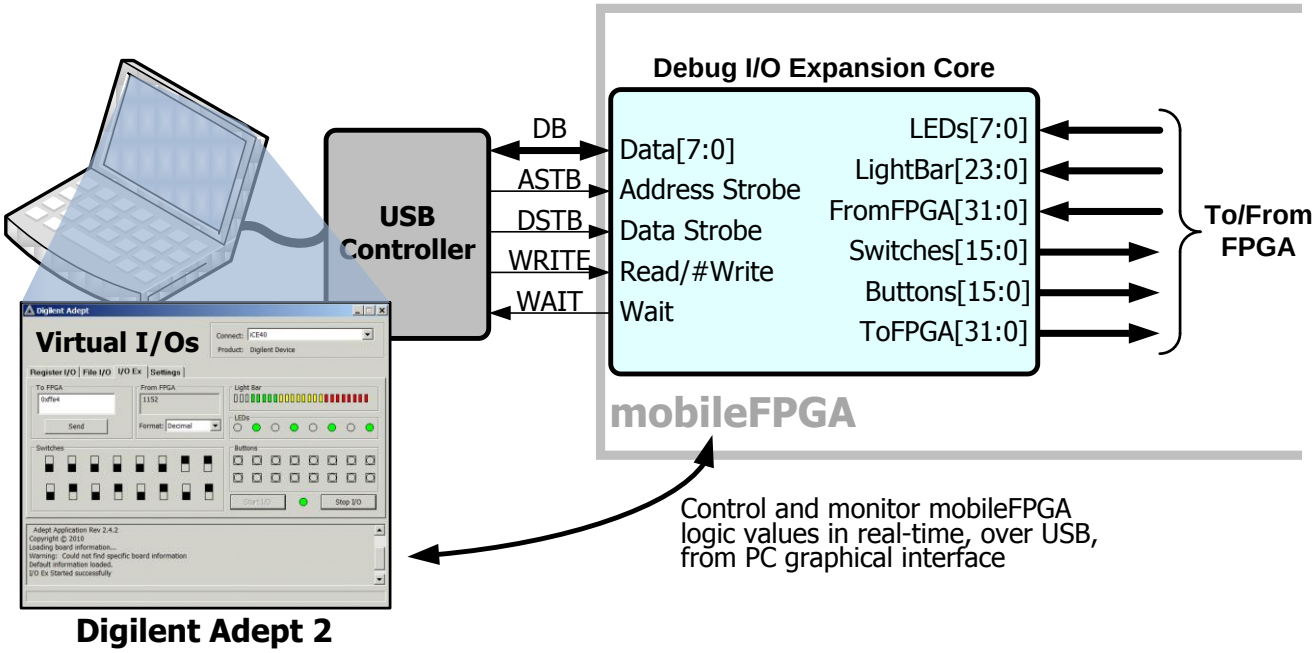


Figure 23: **Digilent Adept 2 I/O Expansion Interface and FPGA Connections**

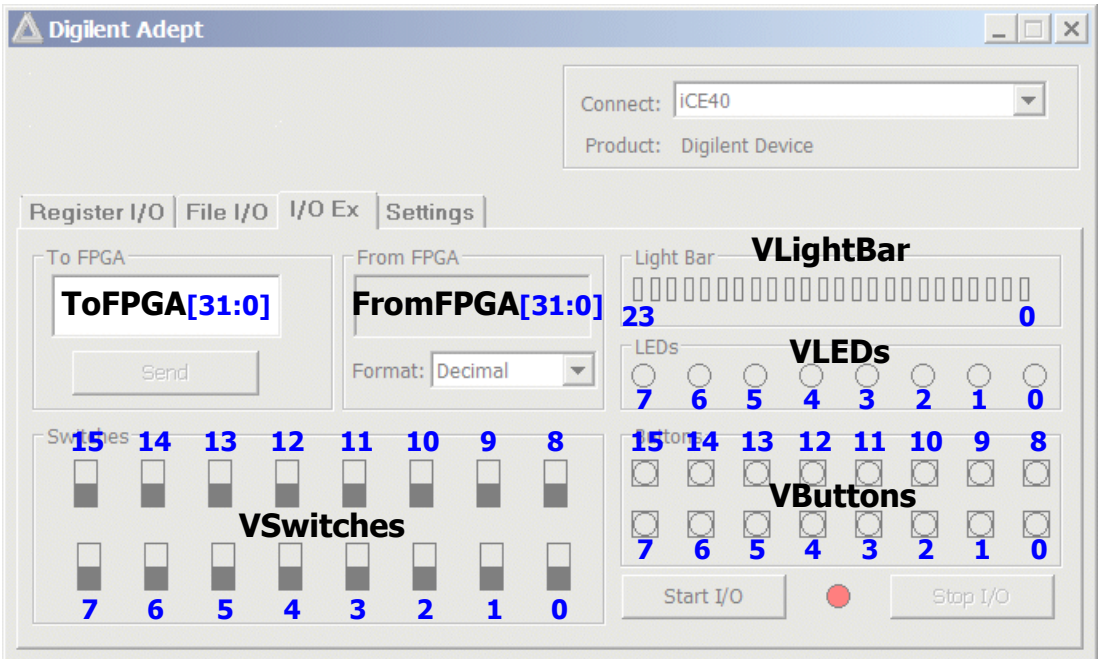


Table 17: **Virtual I/O Connections to FPGA, Shared Connections**

Virtual I/O Control Signal	FPGA Ball	Shared Socket Connection
ASTB	[P10]	J16.4
DB[0]	[T1]	J17.10
DB[1]	[R2]	J17.9
DB[2]	[T2]	J17.8
DB[3]	[R3]	J17.7
DB[4]	[N6]	J16.10, J17.4
DB[5]	[P4]	J17.3
DB[6]	[N5]	J16.28, J17.2
DB[7]	[P5]	J17.1
DSTB	[R10]	J16.5
WAIT	[L10]	J16.27
WRITE	[N10]	J16.6

Enabling the Virtual I/O Interface

To enabled the virtual I/O interface on the board, follow the steps described in "Preparing the USB Interface" on page 5.

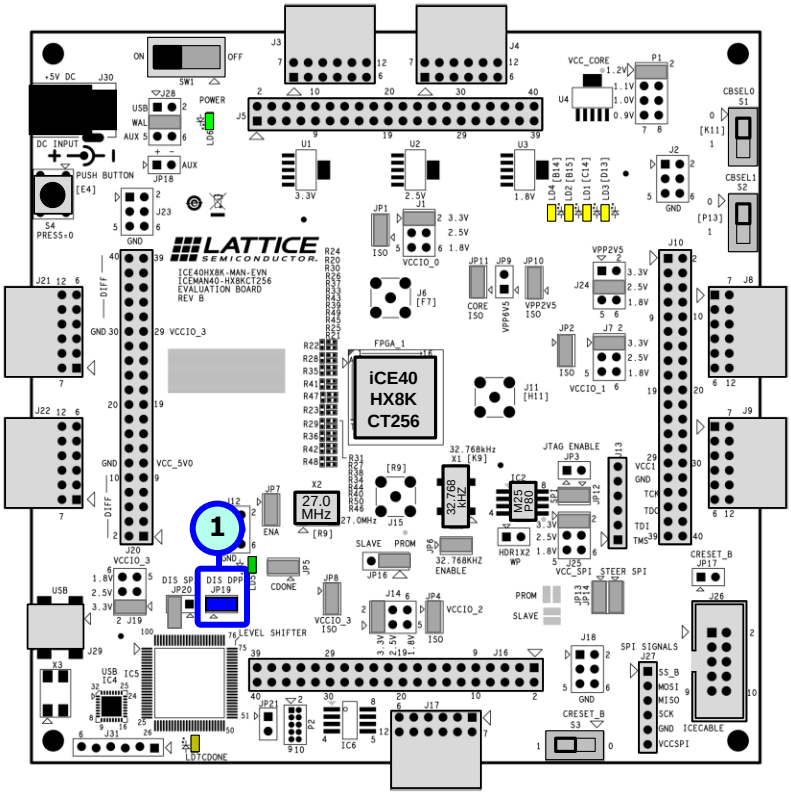
The virtual I/O interface also requires some FPGA logic, as shown in the source code for the default demonstration application.

The demonstration application is available for download from the Lattice Semiconductor web site at ...
`www.latticesemi.com/iCEman40-hx8k`

Disabling the Virtual I/O Interface

To disable the virtual I/O interface and reclaim all the I/O connections listed in Table 17, simply install jumper JP19, as shown in Figure 24. Disabling the virtual I/O interface will not affect the USB-based SPI Flash programming interface.

Figure 24: **Disabling the USB Virtual I/O and Debugging Interface**



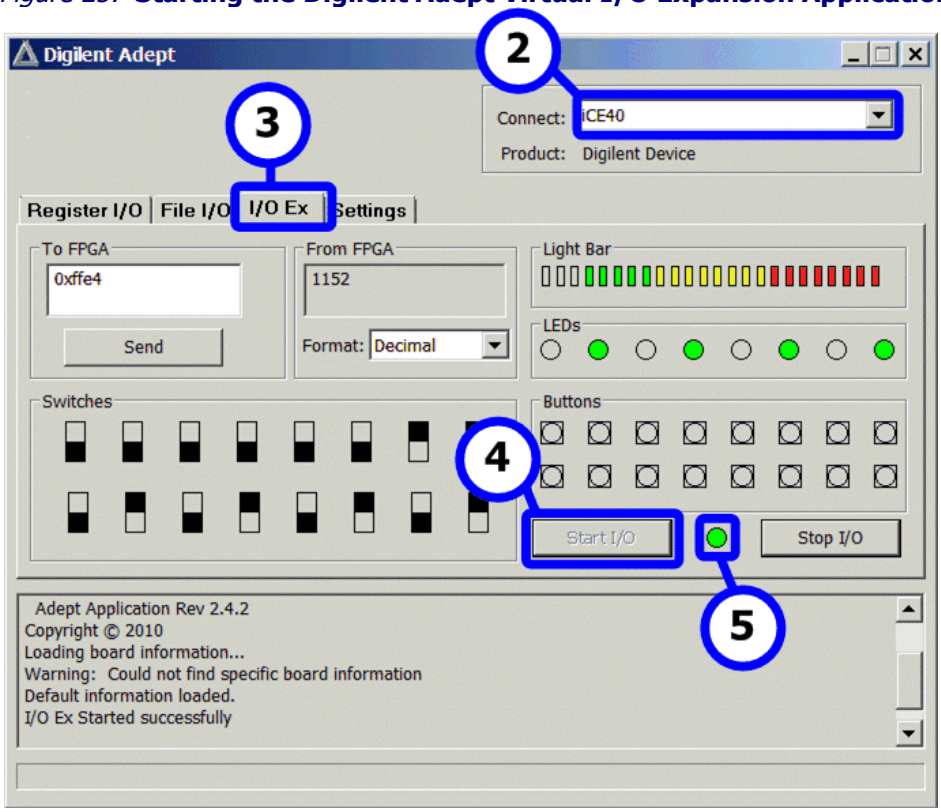
1 Insert jumper JP19 to disable the USB virtual I/O interface.

Using the Virtual I/O in the Default Demo Application

By default, the Virtual I/Os are disconnected and the USB controller's I/O connections to the FPGA are high-impedance (Hi-Z).

To connect the Virtual I/O, perform the following steps outlined in Figure 25.

Figure 25: **Starting the Digilent Adept Virtual I/O Expansion Application**



1. From the Windows Start menu, select **Start → All Program → Digilent → Adept → Adept**
2. Ensure that the Adept interface connects to the **iCE40**.
3. Click the **I/O Ex** tab.
4. Click **Start I/O**. Remember, the associated I/O Expander design must be part of the compiled FPGA design before the Virtual I/Os work.
5. If the virtual I/O expansion design is functioning correctly, the green virtual status LED will turn from red to green.

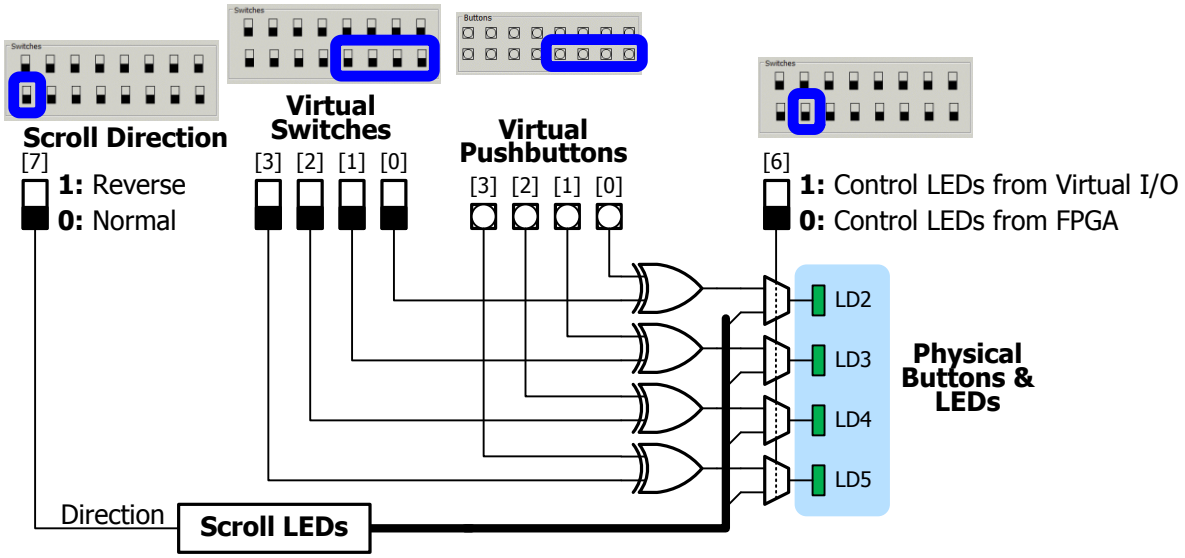
To disconnect the virtual I/O interface, simply click the **Stop I/O** button in the graphical interface.

Controlling the Physical LEDs from Virtual I/Os in the Demonstration Design

When active, the virtual I/Os optionally control the physical LEDs on the board, as shown in Figure 26. For example, with the virtual I/Os active, change the position of virtual switch [7] (the bottom left switch in the graphical interface). Note how the physical LEDs on the board change direction.

Change virtual switch [6] to the up position. Now, the physical LEDs are controlled by the virtual switches [3:0] and virtual pushbuttons [3:0]. The values of the virtual switches are XORed together inside the FPGA. The virtual slide switches set a specific value for the physical LEDs. The pushbutton momentarily inverts the value while the virtual pushbutton is pressed in the graphical interface.

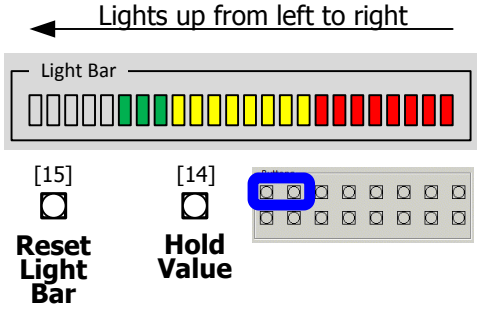
Figure 26: **Controlling the Physical LEDs from Virtual I/O**



Controlling the Virtual Light Bar in the Demonstration Design

When the virtual I/Os are active, the virtual light bar lights up from left to right, controlled by logic inside the FPGA. The virtual pushbuttons [15] and [14] control the light bar. Pushbutton [15] resets the light bar, clearing all the lights. Pushbutton [14] forces the light bar to hold its current value.

Figure 27: **Controlling the Virtual Light Bar**

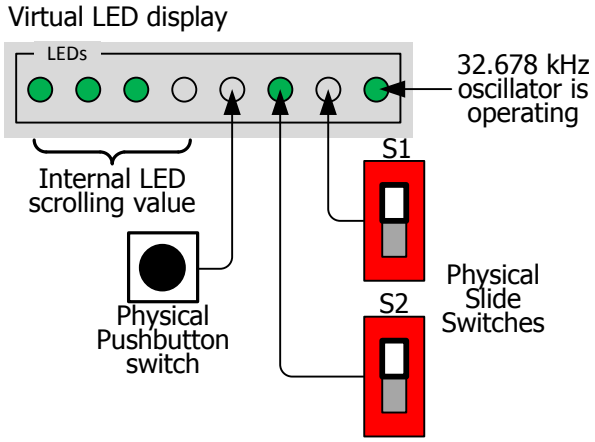


Controlling the Virtual LEDs in the Demonstration Design

The virtual I/O interface includes eight, round, green LEDs, as shown in Figure 28

The eight virtual LEDs are separated into left and right halves. The left-most LEDs indicate the current value of the internal scrolling pattern, regardless of the value indicated on the four physical LEDs on the board—which can be bypassed and controlled by virtual slide switches (see Figure 26).

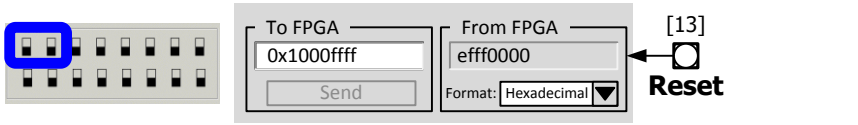
Figure 28: **Controlling the Virtual LEDs**



Virtual Values to and from FPGA in the Demonstration Design

The virtual I/O interface also includes a 32-bit value from the FPGA logic and a 32-bit value to the FPGA logic, as shown in Figure 29. Two virtual switches, [14] and [15], control the behavior in of the virtual 32-bit values in the demonstration design.

Figure 29: **Virtual Values to and from FPGA**



[15] [14]		
[15] [14]	No Function	Continuously increment
[15] [14]	No Function	Continuously decrement
[15] [14]	Input Value	One's complement of Input Value
[15] [14]	Input Value	Reverse bit order of Input Value

Programming the iCEman40 Board

The iCEman40 board provides two different support mechanisms for programming the on-board SPI configuration Flash associated with the FPGA. Both are supported by the iCEcube2 software.

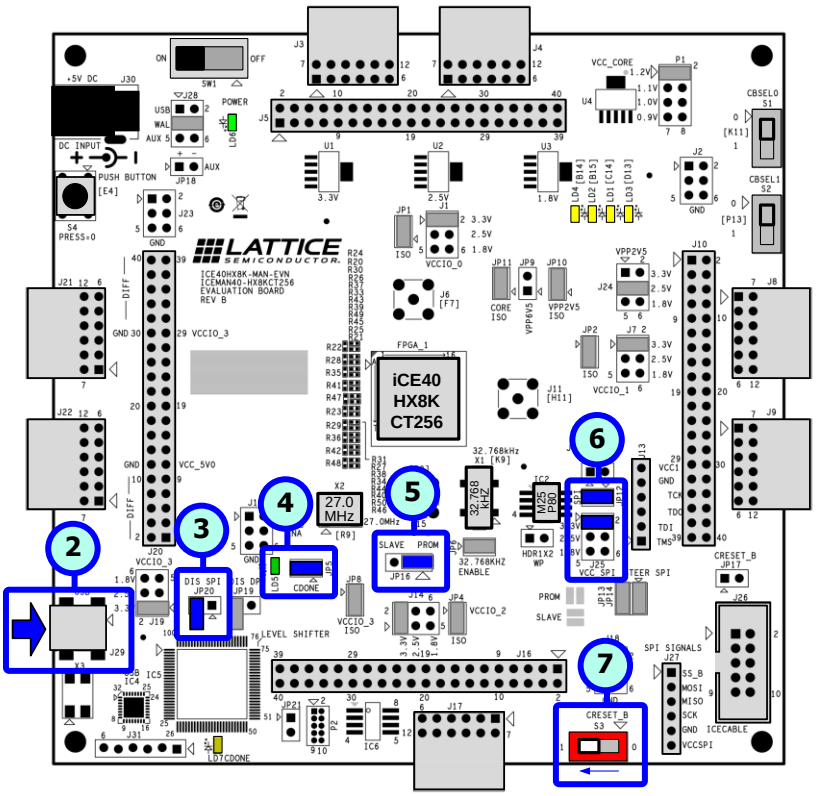
- On-board USB-based programmer
- External SPI programming adapter

Program SPI Flash using On-board USB Programmer

Setup Procedure

Figure 32 shows the steps required to set up the iCEman40 board for programming the on-board SPI configuration Flash PROM using the on-board USB programmer.

Figure 30: **Setting Up to Program SPI Flash using On-board USB Programmer**

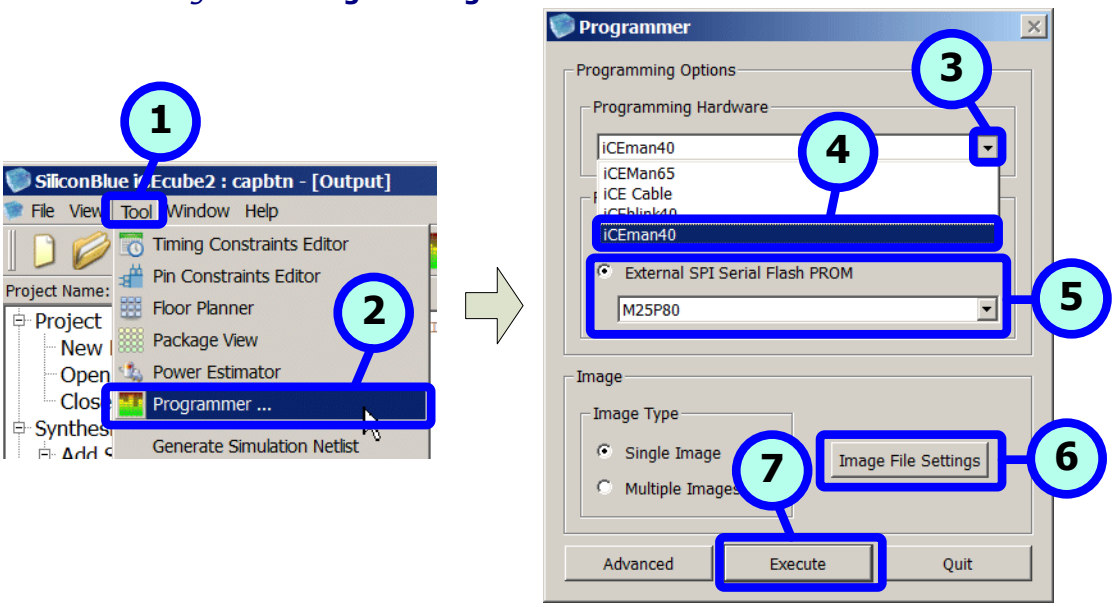


1. Disconnect power from the iCEman40 board.
2. Connect the USB cable between a computer running Lattice Semiconductor iCEcube2 and the mini-USB connector on the board, J29.
3. Ensure that jumper JP20 is removed or disconnected. This enables the on-board USB-based SPI programmer.
4. Ensure that jumper JP5 is installed. This allows the Configuration Done (CDONE) LED to light after the FPGA successfully configures from the newly-programmed SPI PROM.
5. Ensure that jumper JP16 is set to the **PROM** setting, as shown in the diagram.
6. Ensure that jumper JP12 is installed and that jumper J25 is set to **3.3V**.
7. Ensure that the Configuration Reset (CRESET_B) switch, S2, is set to **1**. This setting allows the FPGA to self-configure after the SPI Flash PROM is reprogrammed.
8. Re-apply power to the iCEman40 board.

Program with iCEcube2

Figure 33 shows the commend sequence for programming the SPI Flash PROM on the iCEman40 board using the iCEcube2 development software. These steps assume that a design project was created and processed using the iCEcube2 software.

Figure 31: **Programming the iCEman40 Board from iCEcube2**



1. Select **Tool** from the iCEcube2 menu bar ....
2. ... followed by **Programmer**.
3. Click the dropdown button ( ) under **Programming Hardware**.
4. Select **iCEman40** from the drop list.
5. The iCEcube2 software automatically selects **MP25P80** as the external SPI Flash PROM type.
6. The bitstream file should already be set appropriately based on the iCEcube2 project settings. If not, click **Image Files Settings** to select the configuration bitstream file.
7. Click **Execute** to program the iCEman40 board.
8. Continue to "Configure the FPGA from the Newly-programmed Flash PROM" on page 40.

Optionally using Command Line

The iCEman40 programming software can also be executed from a console window or DOS box. To open a console window or DOS box, click the Start button and type **cmd** in the textbox immediate above the Start button.

After installation, the programming software executable is called **iceutil.exe** and is located in the **\SbtTools\sbt_backend\bin\win32\opt** directory. The **iecutil.exe** executable can be copied into the same directory as the FPGA bitstream image or can be pointed to on the command line.

The required bitstream image is part of the iCEcube2 project. Multiple versions of the bitstream are stored in the **<projname>_Implmnt\sbt\outputs\bitmap** directory. The raw hexadecimal version of the bitstream is called **<projname>_bitmap.hex**. The alternate format of the same information is an Intel hexadecimal file called **<projname>_bitmap_int.hex**.

Raw Hexadecimal Command Example

```
<path>/iceutil -d iCE40 -res -cr -m M25P80 -fh -w <path/projname>_bitmap.hex
```

Intel Hexadecimal Command Example

```
<path>/iceutil -d iCE40 -res -cr -m M25P80 -fi -w <path/projname>_bitmap_int.hex
```

Help

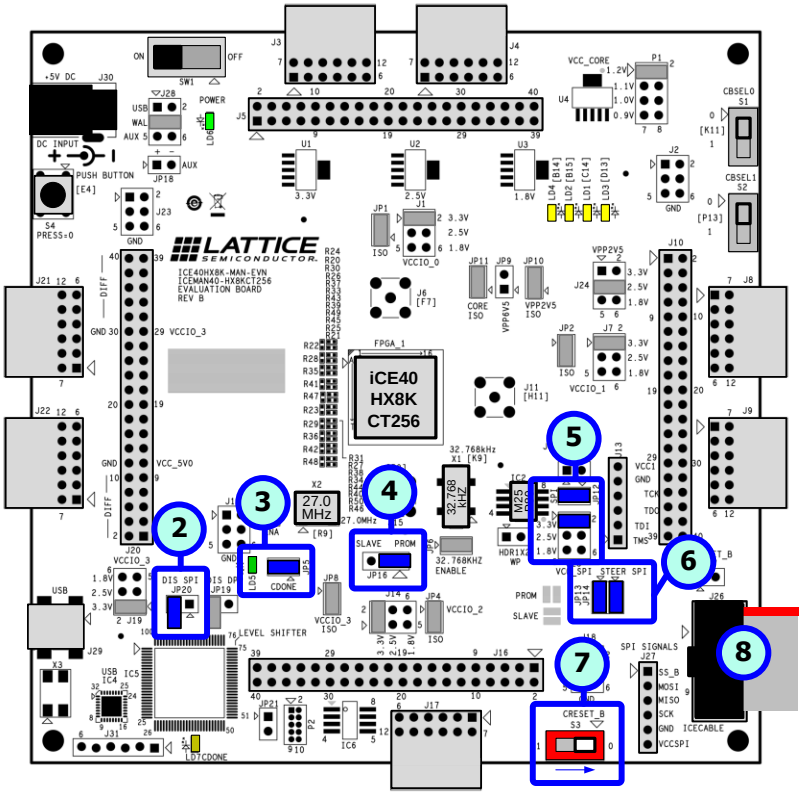
```
<path>/iceutil -help
```

Program SPI Flash using External SPI Adapter

Setup Procedure

Figure 32 shows the steps required to set up the iCEman40 board for programming the on-board SPI configuration Flash PROM using an external SPI programming adapter such as the Lattice Semiconductor iCEcable, the TotalPhase Aardvark or Cheetah SPI adapter, or the DediProg programming adapter.

Figure 32: **Setting Up to Program SPI Flash using iCEcable**

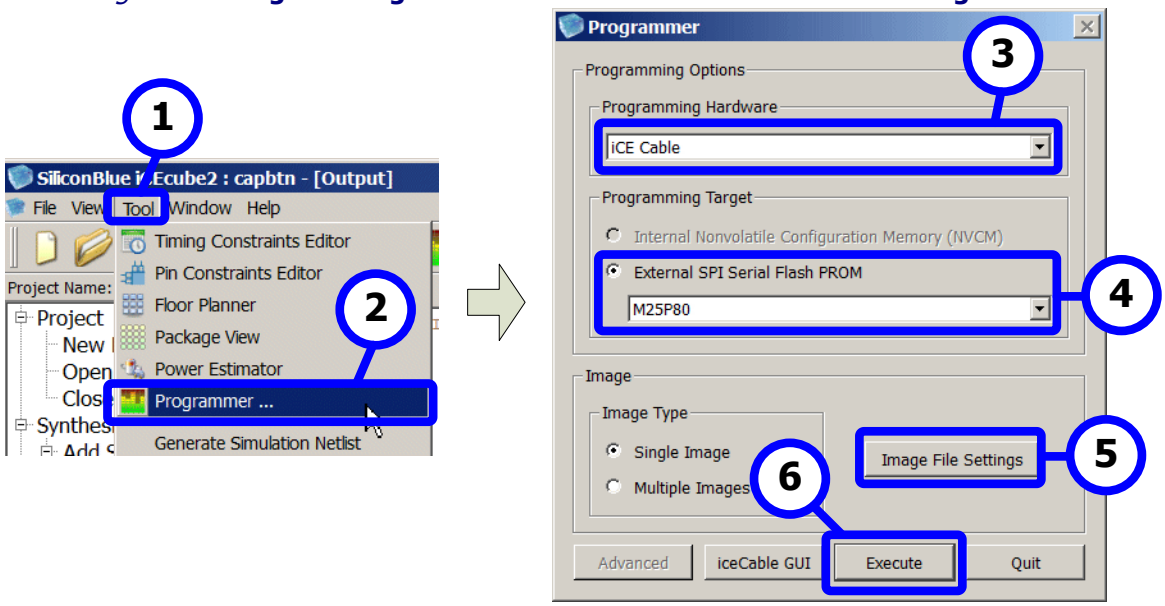


1. Disconnect power from the iCEman40 board.
2. Ensure that jumper JP20 is installed. This disables the USB-based SPI programmer.
3. Ensure that jumper JP5 is installed. This allows the Configuration Done (CDONE) LED to light after the FPGA successfully configures from the newly-programmed SPI PROM.
4. Ensure that jumper JP16 is set to the PROM setting, as shown in the diagram.
5. Ensure that jumper JP12 is installed and the jumper J25 is set to **3.3V**, as shown in Figure 32.
6. Ensure that jumpers JP13 and JP14 are installed and oriented as shown in the diagram. These jumpers steer the SPI data to the SPI PROM for the attached programming adapter cable.
7. Set switch S3 to **0**. This setting holds the FPGA's CRESET_B input Low and allows the iCEcable access to the SPI PROM.
8. Insert the programming header from the iCEcable as shown in the diagram. The socket is keyed to ensure proper alignment. The red edge of the ribbon cable faces the top edge of the iCEman40 board, as shown in Figure 32.
9. Re-apply power to the iCEman40 board using the power switch, SW1

Program SPI Flash using iCEcable and iCEcube2

Figure 33 shows the commend sequence for programming the SPI Flash PROM on the iCEman40 board using the iCEcube2 development software. These steps assume that a design project was created and processed using the iCEcube2 software.

Figure 33: **Programming the iCEman40 Board from iCEcube2 using iCEcable**

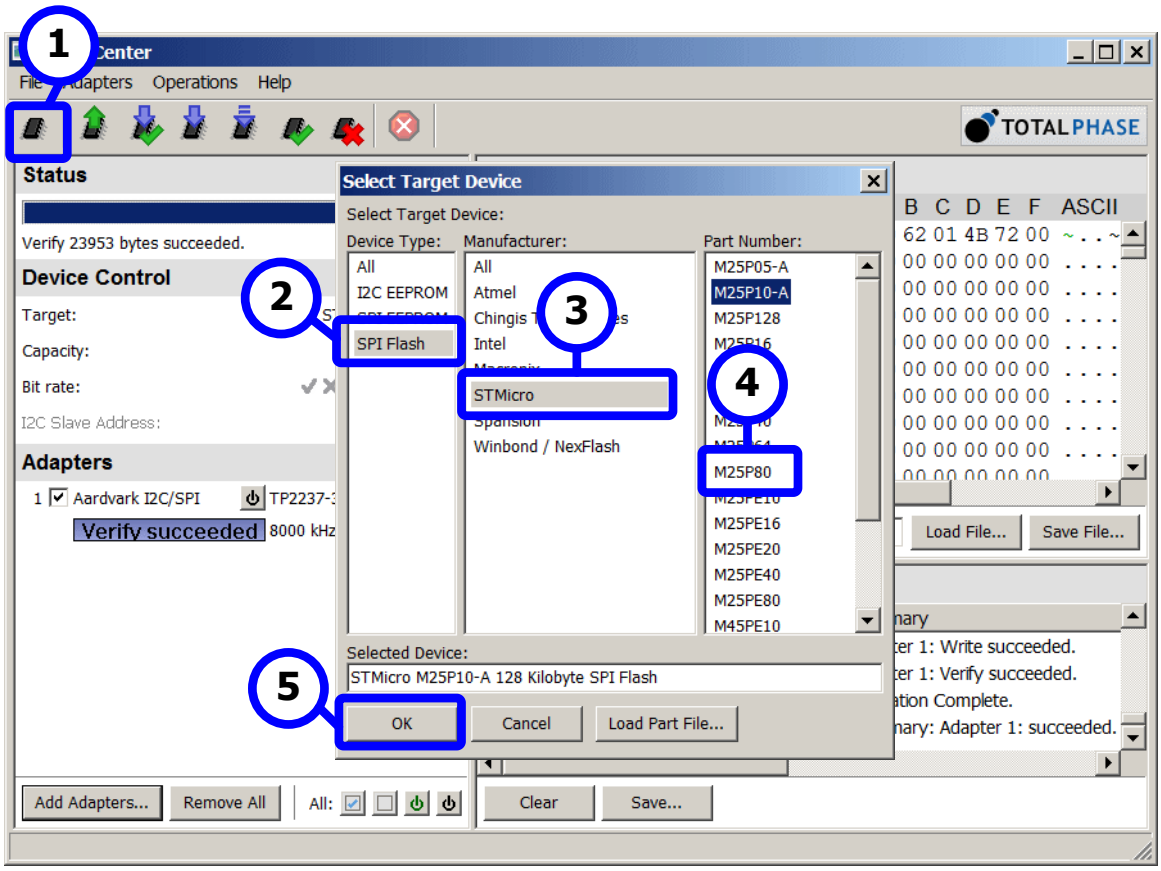


1. Select **Tool** from the iCEcube2 menu bar ....
2. ... followed by **Programmer**.
3. By default, **iCE Cable** is selected as the Programming Hardware.
4. By default, **MP25P80** is selected as the external SPI Flash PROM type.
6. The bitstream file should already be set appropriately based on the iCEcube2 project settings. If not, click **Image Files Settings** to select the configuration bitstream file.
7. Click **Execute** to program the iCEman40 board.
8. Continue to "Configure the FPGA from the Newly-programmed Flash PROM" on page 40.

Programming SPI Flash using Aardvark Adapter and Flash Center

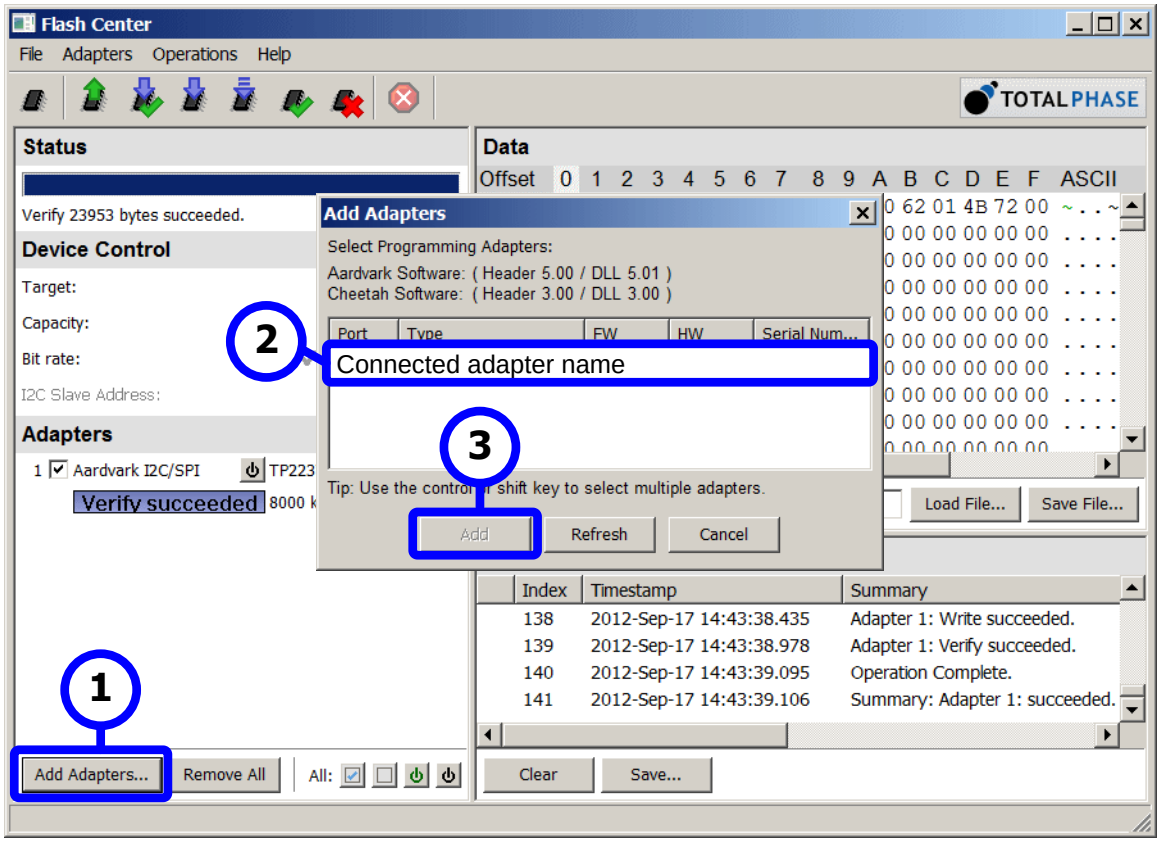
Invoke the TotalPhase Flash Center software.

Set the SPI PROM Type



1. Click the chip icon to select the programming target.
2. Choose **SPI Flash** for Device Type.
3. Choose **STMicro** for Manufacturer. The PROM may have been purchased from Micron or Numonyx but STMicro is the original designer.
4. Select **M25P80** as the Part Number.
5. Click **OK**.

Connect the Aardvark I2C/SPI Adapter

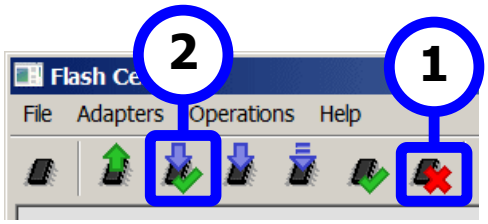


1 Click **Add Adapters …**.

2 Select the listed port adapter.

3 Click **Add**.

Programming the SPI Flash PROM



1 Click  to erase the SPI serial PROM.

If the erase operation is not successful, ensure that the shorted jumper header and the specialty header are properly connected and cycle the power on the board.

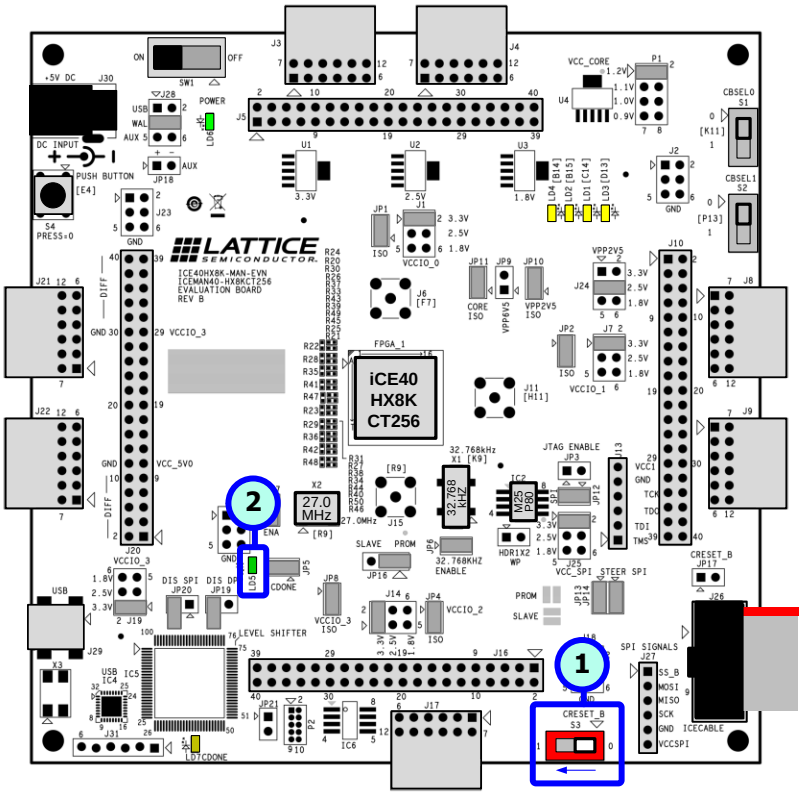
2 Click  to download and verify the SPI serial PROM.

3 Continue to "Configure the FPGA from the Newly-programmed Flash PROM" on page 40.

Configure the FPGA from the Newly-programmed Flash PROM

After successfully programming the SPI PROM, perform the steps outlined in Figure 34.

Figure 34: **Allowing FPGA to Configure from Newly-Programmed Flash PROM**



1. Set switch S3 to **1**. This setting releases the FPGA's CRESET_B input, allowing the FPGA to access and configure from the SPI PROM.
2. The Configure DONE LED (LD5) lights up when the FPGA successfully configures.

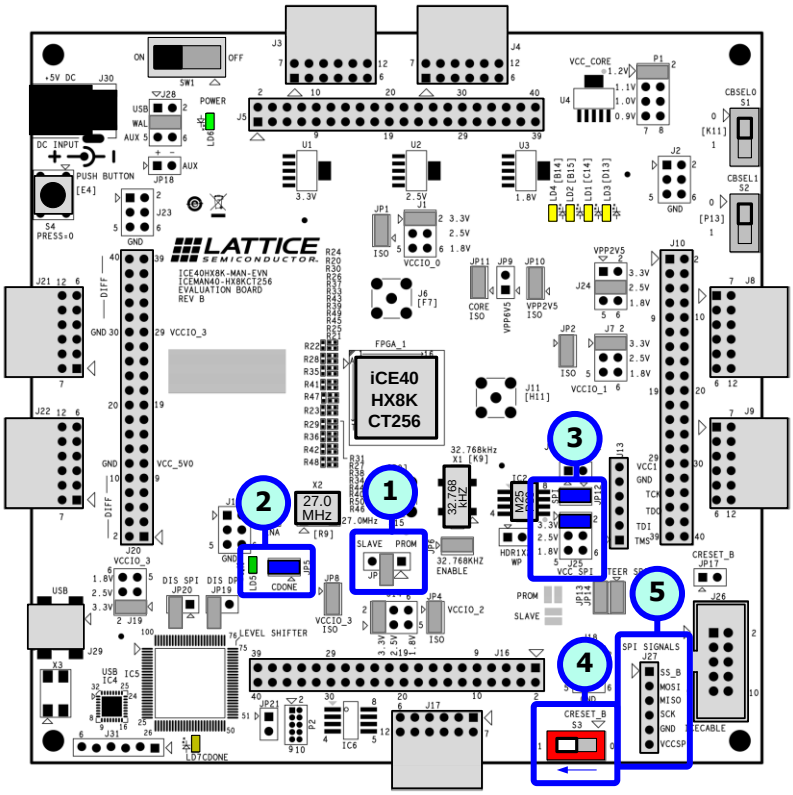
Configure the FPGA from an External SPI Flash PROM Module

Some iCEman40 board users prefer to configure the FPGA using an external SPI Flash PROM module such as the Digilent PmodSF module, shown in Figure 35. The SPI Flash module is programmed off board with the FPGA configuration image using iCEcable or other SPI programmer. This capability also provides a means to use SPI Flash vendors and components that are not supported by the iCEcube2 software.

Figure 35: **Digilent PmodSF Serial Flash Module and 6-pin Cable**

Once the module is programmed, plug the module into header J27, as described in Figure 36.

Figure 36: **Configuring the FPGA from an External SPI Flash Module**



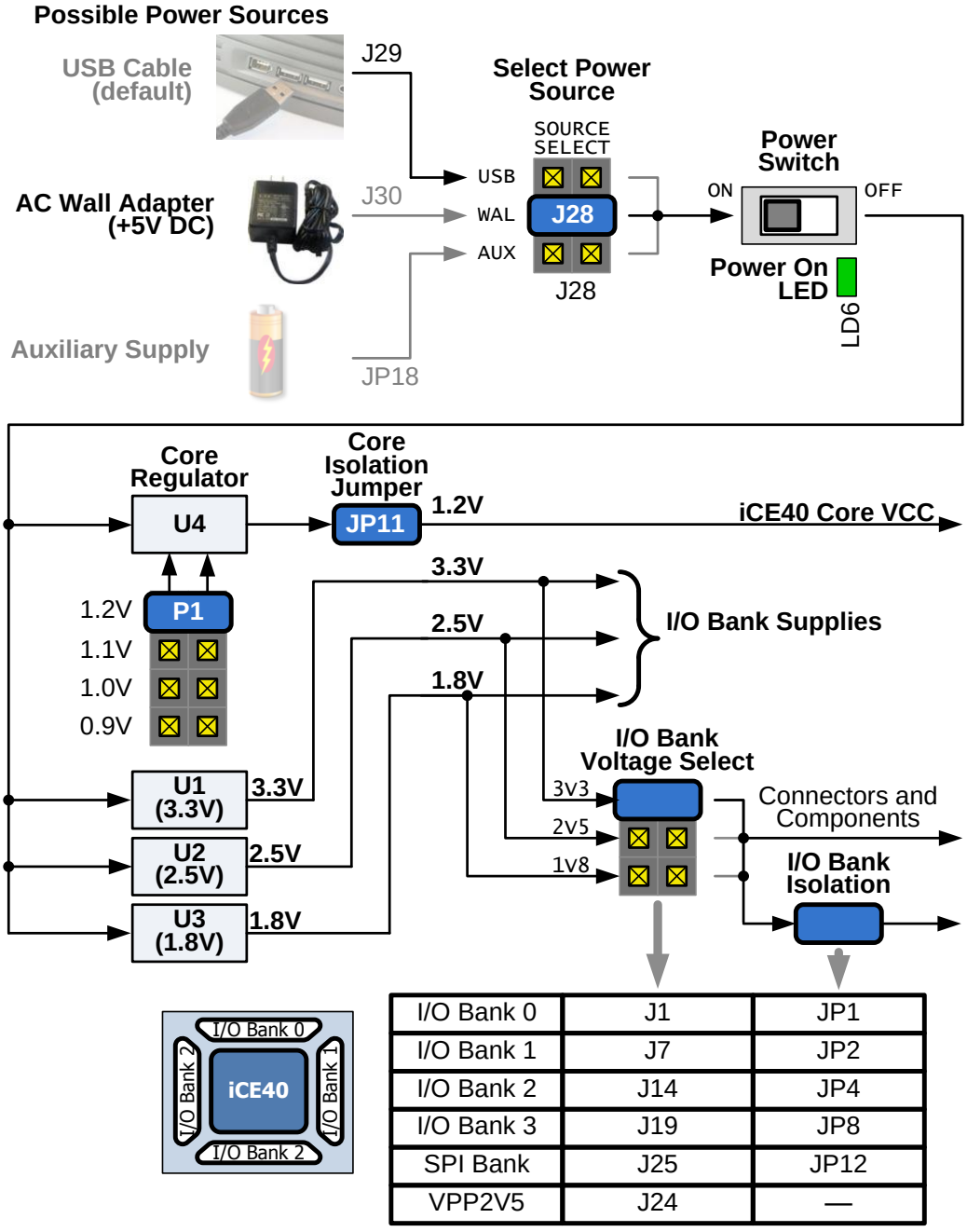
1. Disconnect jumper JP16, which disables the on-board SPI Flash PROM, IC2.
2. Ensure that jumper JP5 is installed. This allows the Configuration Done (CDONE) LED to light after the FPGA successfully configures from the newly-programmed SPI PROM.
3. Ensure that jumper JP12 is installed and that jumper J25 is set to the proper voltage level for the SPI Flash PROM module.
4. Ensure that the Configuration Reset (CRESET_B) switch, S2, is set to **1**. This setting allows the FPGA to self-configure after the SPI Flash PROM is reprogrammed.
5. Connect the external SPI Flash PROM module to header J27. Depending on the SPI PROM module, the connection may require a 6-pin female-to-female cable, often shipped with the module.

Power Supply Design Overview

Figure 37 depicts the power supply design for the iCEman40-HX8Kboard. There are three possible power inputs, although generally the board is powered from the AC wall adapter.

The J28 jumper selects between the three possible power sources. See "Powering the iCEman40 Board via the USB Interface (WARNING!)" on page 26 before using the USB option. The selected power source is then controlled by the power switch, SW1. If the board is connected to a power supply, then the red LED next to the switch lights up when the switch is turned ON.

Figure 37: **iCEman40 Board Power Supply Overview**



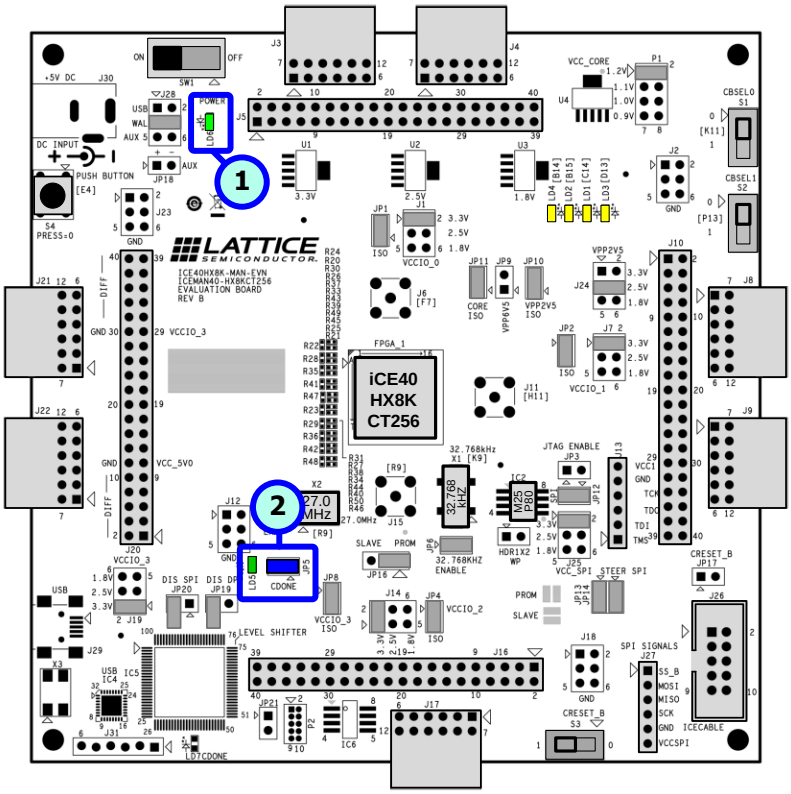
I/O Bank 0	J1	JP1
I/O Bank 1	J7	JP2
I/O Bank 2	J14	JP4
I/O Bank 3	J19	JP8
SPI Bank	J25	JP12
VPP2V5	J24	—

Power and Configuration Status LEDs

The iCEman40 evaluation board has two status LEDs, as shown in Figure 38. These two status LEDs indicate the current status of the iCEman40 board, as listed in Table 18. The green LED, LD6, located under the power switch indicates if power is applied to the board.

The dimly lit LED, LD5, located below the FPGA indicates whether the FPGA is configured properly. This LED lights up when the FPGA is correctly loaded with a valid bitstream.

Figure 38: **iCEman40 Status LEDs**



1 Power Good LED (LD6). Controlled by power switch SW1 and the power-source select jumper, J28.

2 Configuration DONE LED (LD5). Jumper JP5 must be installed for LD5 to light.

Table 18: **iCEman40 Status LEDs and Their Meaning**

Power-Good LED (LD6)	Configuration DONE LED (LD5)	Meaning
On	**On**	The board is powered, the FPGA successfully configured and the FPGA application is operating.
Off	**Off**	Board is unpowered. The FPGA is unconfigured.
On	**Off**	The board is powered but the FPGA is not yet configured.

SPI Configuration Interface

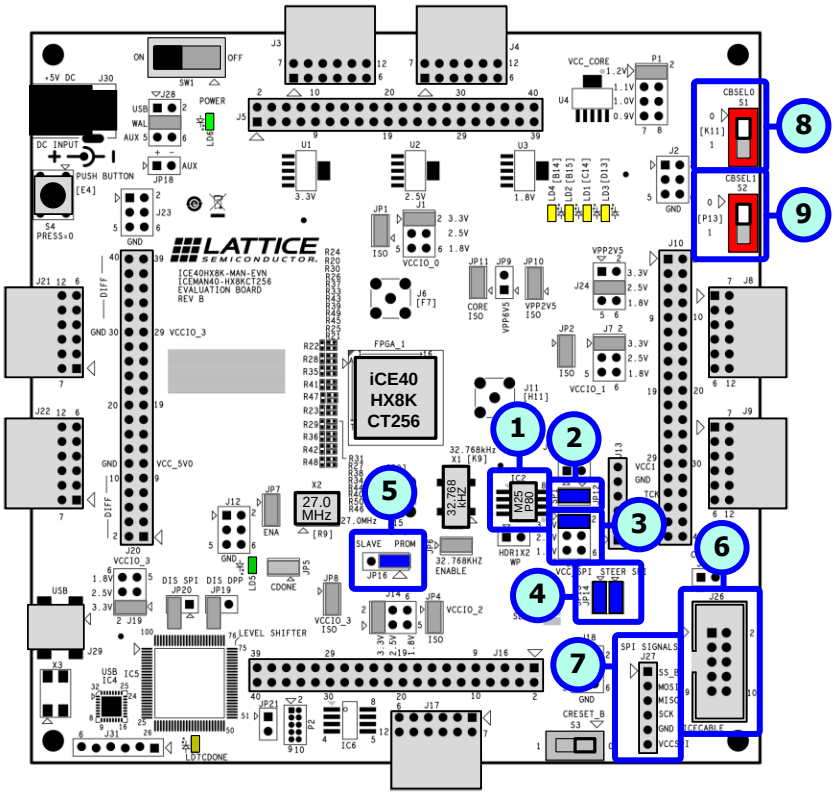
For development purposes, the iCE40 mobile FPGA on the iCEman40 board either configures from the on-board 8Mbit SPI PROM or is configured by an external SPI host. It does not configure from the internal programmable Nonvolatile Configuration Memory (NVCM).

Figure 39 shows the various options available on the SPI interface.

SPI Serial Flash

The iCEman40 board includes a 25-series SPI serial Flash PROM to hold one or more iCE40 configuration images.

Figure 39: **iCEman40-HX8K SPI Interface Controls**



1 Micron/Numonyx/STMicro M25P80 8Mbit SPI Serial Flash (IC2). The Flash-based PROM can be reprogrammed at least 100,000 times. Generally, the PROM holds a single configuration image. However, using the iCE40 ColdBoot or WarmBoot feature, the 8Mbit PROM is large enough to hold up to five FPGA configuration options.

2 SPI supply voltage isolation jumper, JP12.

3 SPI supply voltage select header, J25. The SPI PROM is specified to operate at 3.3V but will typically also operate at the 2.5V setting. The PROM on the board does not support 1.8V operation.

4 SPI Flash data steering jumpers. Align the jumpers vertically (up and down) when programming the SPI Flash or configuring from it.

5 The SPI interface is generally used in PROM mode. However, an external intelligent host, such as a microprocessor, can also download the configuration bitmap image to the FPGA via a slave SPI interface. This jumper, JP16, select between **PROM** mode and SPI **SLAVE** mode.

6 Header for an external SPI Flash programming adapter.

7 SPI control signals. Oriented like a six-pin Peripheral Module (Pmod).

8 If using the ColdBoot configuration option, slide switch S1 is the CBSEL0 control input during the FPGA configuration process. After the FPGA configures, the switch is general-purpose input switch.

9 If using the ColdBoot configuration option, slide switch S2 is the CBSEL1 control input during the FPGA configuration process. After the FPGA configures, the switch is general-purpose input switch.

Supply Voltage

Although the SPI interface can operate at different voltages, set jumper J25 (item 3 in Figure 39) to 3.3V. The SPI serial PROM mounted on the board is a 3.3V device. The SPI PROM likely functions and likely can be programmed at 2.5V, but this is not guaranteed.

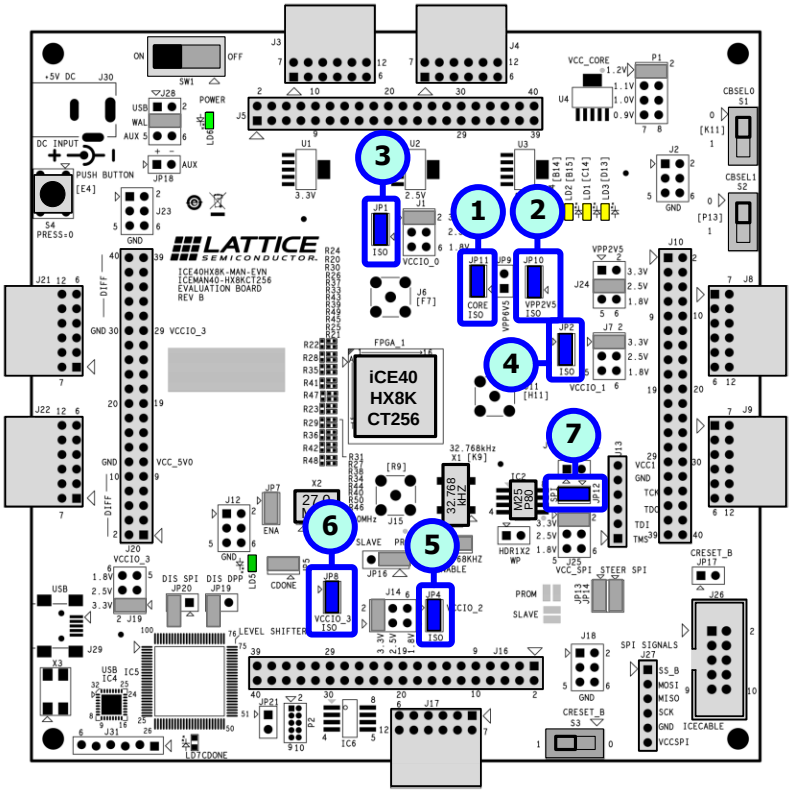
Write-Protect Jumper

The SPI PROM has a write-project option. The empty 2x1 header holes labeled "WP", located below the SPI PROM (item 1 in Figure 39) can be loaded with a 2x1 header or loaded with a wire jumper. Installing the associated jumper prevents the PROM from being erased, programmed, or overwritten. If installed, the jumper must be removed to program the SPI PROM.

Measuring FPGA Power

Various power isolation jumpers, shown in Figure 40, provide the ability to measure core power consumption by the FPGA.

Figure 40: **Power Isolation Jumpers for Measuring FPGA Power Consumption**



1. VCC: JP11 isolates the FPGA's core VCC voltage inputs for easy power measurement.
2. VPP_2V5: JP10 isolates the FPGA's VPP_2V5 voltage input for easy power measurement.
3. VCCIO_0: JP1 isolates the FPGA's VCCIO_0 voltage inputs for measuring power to the I/O pins in I/O Bank 0.
4. VCCIO_1: JP2 isolates the FPGA's VCCIO_1 voltage inputs for measuring power to the I/O pins in I/O Bank 1.
5. VCCIO_2: JP4 isolates the FPGA's VCCIO_2 voltage inputs for measuring power to the I/O pins in I/O Bank 2.
6. VCCIO_3: JP8 isolates the FPGA's VCCIO_3 voltage inputs for measuring power to the I/O pins in I/O Bank 3.
7. SPI_VCC: JP12 isolates the FPGA's SPI_VCC voltage input for measuring power to the I/O pins in the SPI interface.

Two power measurement methods are supported, as described below.

Easy Method using a Multimeter

Connect the iCEman40 board through your high-accuracy multimeter. Use a meter with a minimum of 10,000 counts; 50,000 counts or more is recommended for better accuracy.

To take a quick measurement, follow these steps.

1. Disconnect power to the iCEman40 board by removing the USB cable connection, either at the board or at the computer.
2. Remove the jumper JP1, which isolates the FPGA's core supply from the 1.2V supply on the board.
3. Connect your multimeter's alligator or test clips to the stake pins on header JP1.
4. Configure the multimeter to measure current using its highest mA or Amp range. This setting typically has the lowest voltage drop internally within the meter.
5. Re-connect the USB cable that supplies power to the iCEman40 board and configure the FPGA device if necessary.
6. Observe the power reading on the multimeter. At low clock rates, which resulst in lower power consumption, switch the meter to a lower amperage setting for better accuracy. However, this also may increase the resistance across the meter leads. Using too low of a meter setting causes a large voltage drop within the meter, potentially violating the minimum input voltage specification to the FPGA device.
7. The value measured by the multimeter is a current. Convert the measurement to power using Equation 1. The voltage is the operating voltage, the voltage across the jumper. This value can be accurately measured with a second multimeter to show the voltage drop across the first. However, just measuring the initial voltage, before taking any current readings, usually provides acceptable accuracy and the voltage drop across the meter is generally small.

Equation 1

$$\text{Power} = \text{Current} \times \text{Voltage}$$

Although this method is easy, here are a few caveats and pointers.

- Always start at the highest current setting for your meter. Using too small a setting may damage your meter! After determining the maximum current range for your measurement, then you can safely use the appropriate lower current setting.
- The voltage drop across the meter leads may violate the minimum supply voltage specification for the FPGA device. To determine the voltage drop, use a second multimeter to measure either the voltage across the first meter's leads during a test or the resistance between the first meter's leads.
- Using the highest current measurement setting typically results in the lowest voltage drop.

Using High-Precision, Small-Value Resistors

For more-accurate, time-sensitive measurements, place a low-value resistor across the jumper test point. According to Ohm's Law, the current passing through the resistor produces a voltage drop. Measure the voltage differential across the resistor during expected operation. Convert the measurement to power using Equation 2. The voltage is the measured voltage across the resistor; the resistance is the value of the resistor.

Equation 2

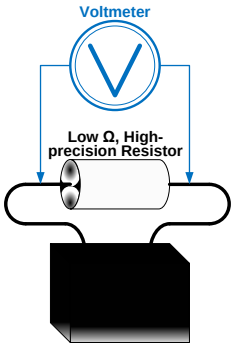
$$\text{Power} = \frac{(\text{Voltage})^2}{\text{Resistance}}$$

The following are a few guidelines on selecting a resistor.

- Use a high-precision resistor.
- The resistor must handle the power dissipated under the anticipated test conditions.
- Too small a resistor value may result in too small a voltage difference across the resistor to measure with your test equipment.
- Too large a resistor value may result in too large of a voltage difference across the resistor. Too large a voltage drop might violate the minimum voltage specifications for the FPGA device.

Figure 41 shows an example header block designed to fit over one of the jump locations. Measure the voltage drop across the low-value resistor, either with a voltmeter or with data acquisition equipment.

Figure 41: **Resistor Header Block**

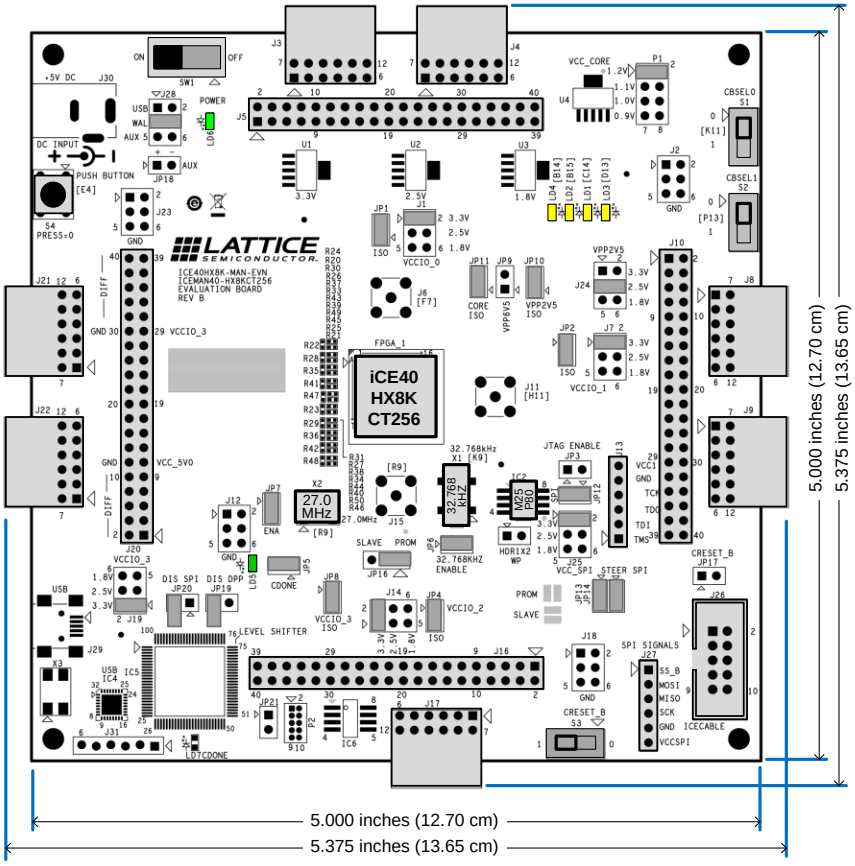


This method is recommended for taking power measurements over time.

Mechanical Specifications

Figure 42 shows the mechanical dimensions for the iCEman40 board. With a jumpers installed, the board height is approximately 0.700 inches high, including the four rubber feet mounted on the bottom side of the board.

Figure 42: **iCEman40 HX8K Board Mechanical Dimension**



Ordering Information

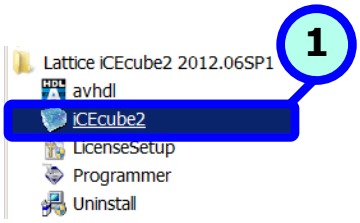
Table 19 lists the available or planned iCEman40 boards, the FPGA mounted on the board, and the ordering information for the boards.

Table 19: **Ordering Information for iCEman40 Evaluation Kits**

Product	FPGA	Part Number
iCEman40-HX1K Evaluation Kit	iCE40HX8KCT256	ICE40HX8K-MAN-EVN

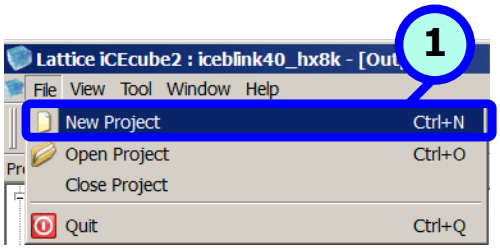
Re-creating the Default Design from Source Inputs

Invoke iCEcube2

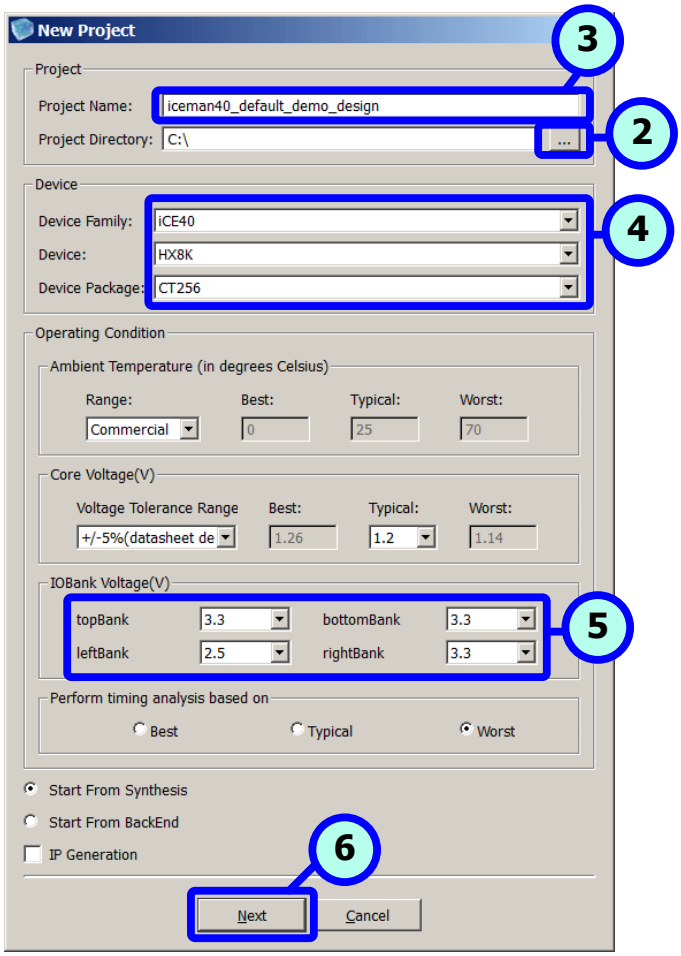


1 From Windows, select **Start → Lattice iCEcube2 → iCEcube2**.

Create New iCEcube2 Project



1 Once iCEcube2 starts, select **File → New Project**.



2 Choose the project directory.

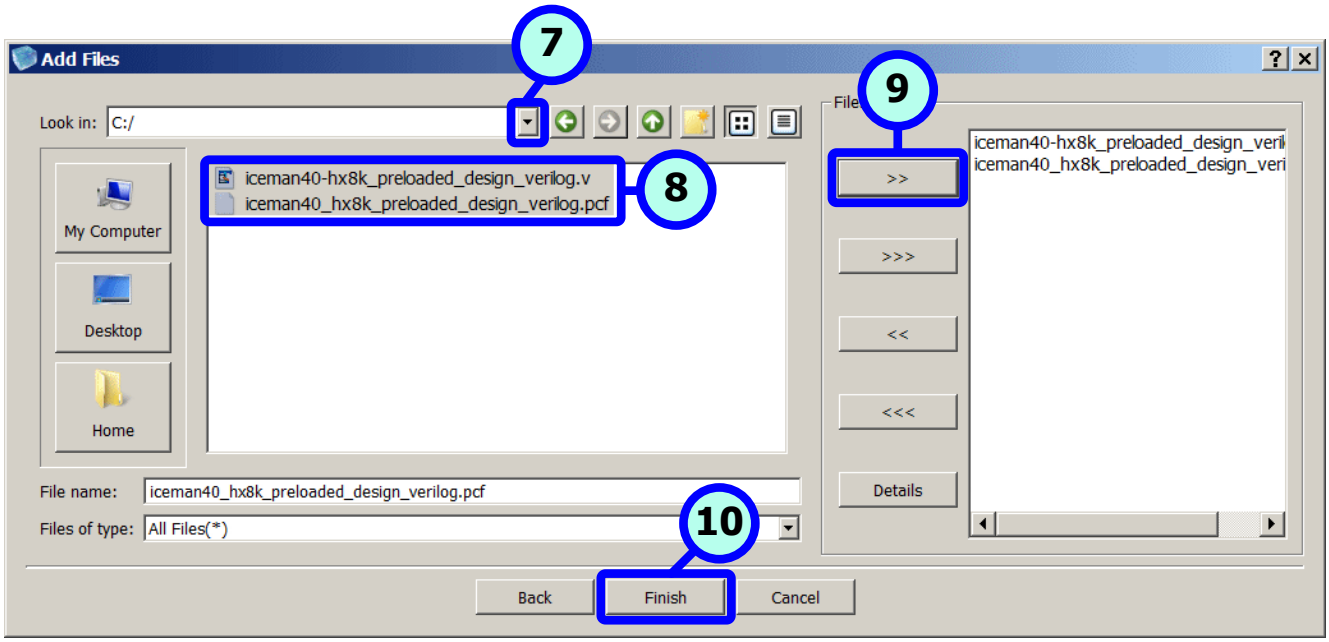
3 Enter a name for the iCEcube2 project.

4 Select the target device on the iCEman40 board.

- Device Family = **iCE40**
- Device = **HX8K**
- Device Package = **CT256**

5 For the delay calculator, set the voltage for each of the four I/O banks. The actual settings for the default design do not matter although the LEDs are brightest when the I/O Bank 0 voltage is set to 3.3V on the board.

6 Click **Next**.



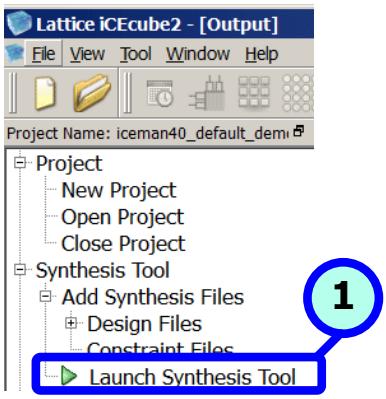
7 Select the directory where you saved the source project files.

8 Click on the Verilog hardware description and the pin constraints file (PCF) source files.

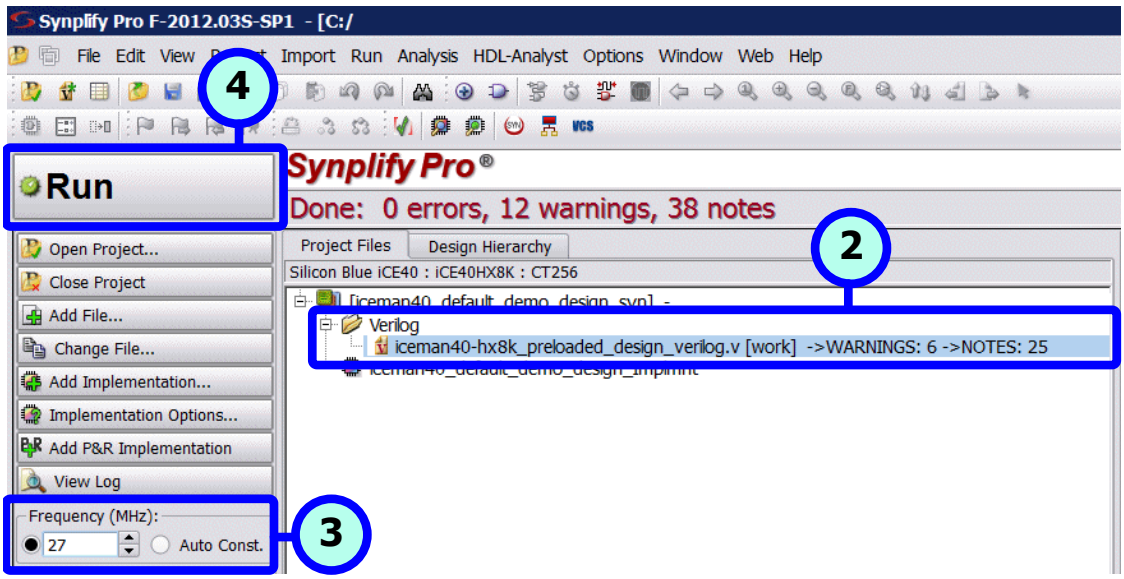
9 Click **>>** to add the source files to the project.

10 Click **Finish**.

Run Synopsys Synplicity to Synthesize Design



1 From iCEcube2, double-click **Launch Synthesis Tool**.



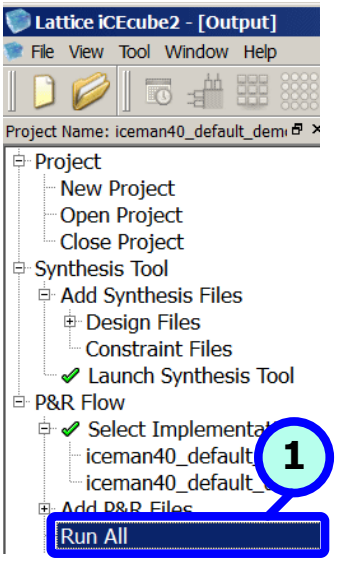
2 Click to expand Verilog in the build tree. Double-click on the Verilog file to view the source code in the syntax-aware editor.

3 Set the default operating Frequency (MHz) to **27**. The iCE40HX8K FPGA will operate well in excess of this speed.

4 Click **Run** to synthesize the design from the Verilog source file.

5 Select **File → Exit** to return to iCEcube2.

Compile the Design in iCEcube2

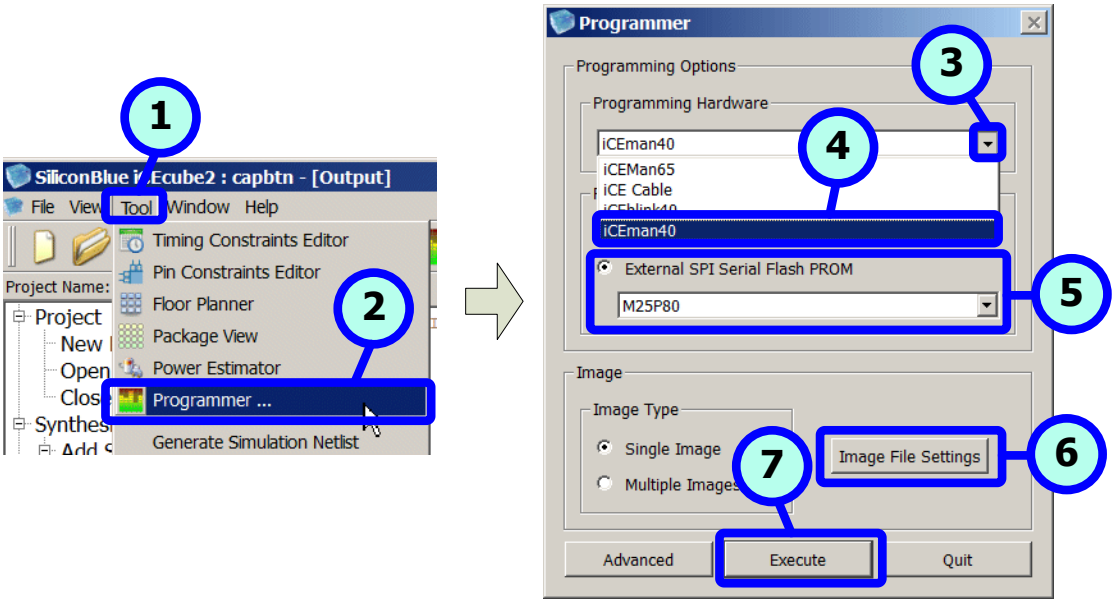


1 From iCEcube2, double-click **Run All**. This action imports the synthesized design, runs the place and router on the imported design, then generates the configuration bitmap for the completed design.

The place and route use the information provided in the pin constraints file (PCF) to assign the I/O locations.

Programming the FPGA's SPI PROM

This example assumes that the iCEman40 board is set up to use the on-board USB programmer.



1 Select **Tool** from the iCEcube2 menu bar ....

2 ... followed by **Programmer**.

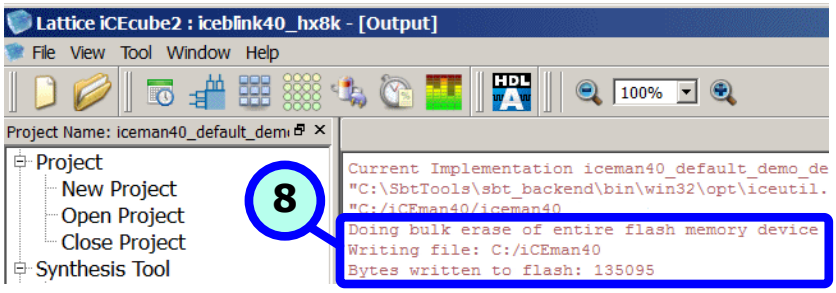
3 Click the dropdown button ( ) under **Programming Hardware**.

4 Select **iCEman40** from the drop list.

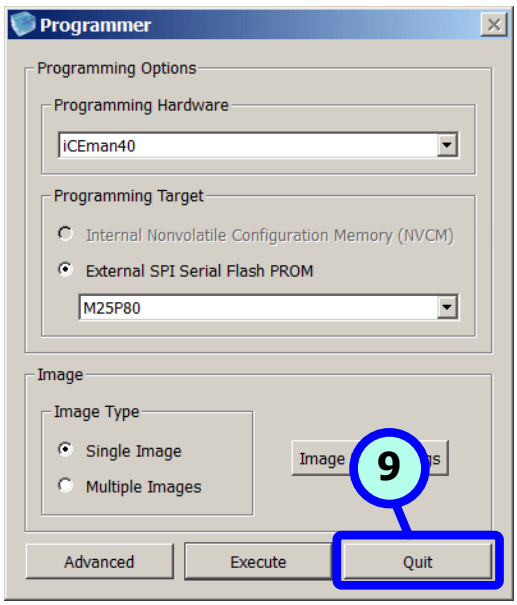
5 The iCEcube2 software automatically selects **MP25P80** as the external SPI Flash PROM type.

6 The bitstream file should already be set appropriately based on the iCEcube2 project settings. If not, click **Image Files Settings** to select the configuration bitstream file.

7 Click **Execute** to program the iCEman40 board.



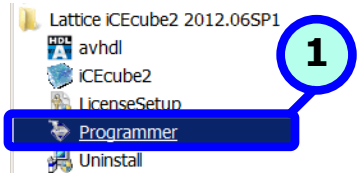
8 The Configuration Done (CDONE) LED will go out during programming. The iCEcube2 status window shows the progress and status of programming. After programming is complete, the CDONE LED should light, indicating that the SPI Flash is properly programmed and that the FPGA has properly configured.



9 Click **Quite** to return to iCEcube2.

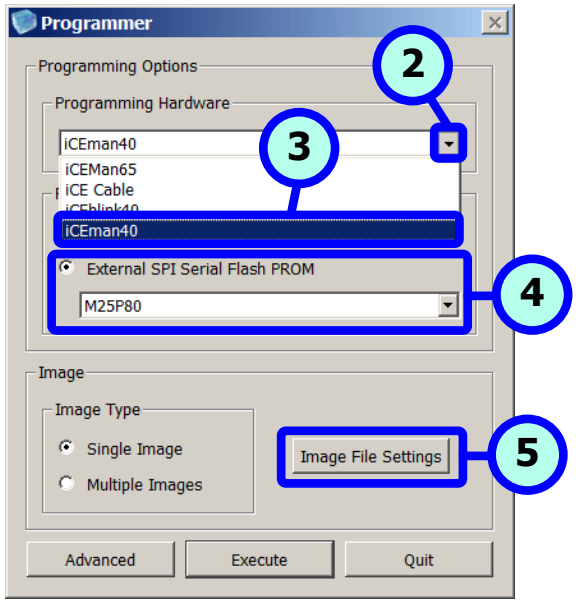
Restoring the Default Programming File from the FPGA Bitmap File

Invoke Lattice Programmer

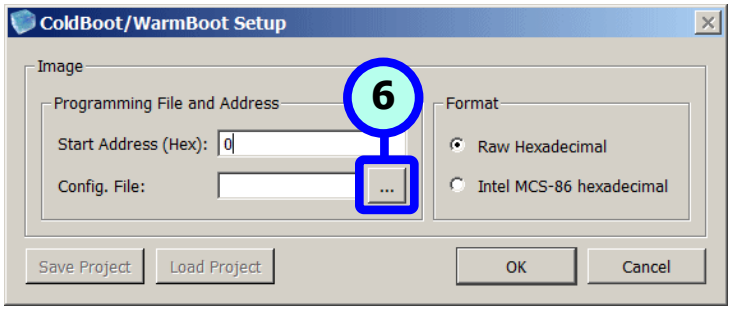


1 From Windows, select **Start** → **Lattice iCEcube2** → **Programmer**.

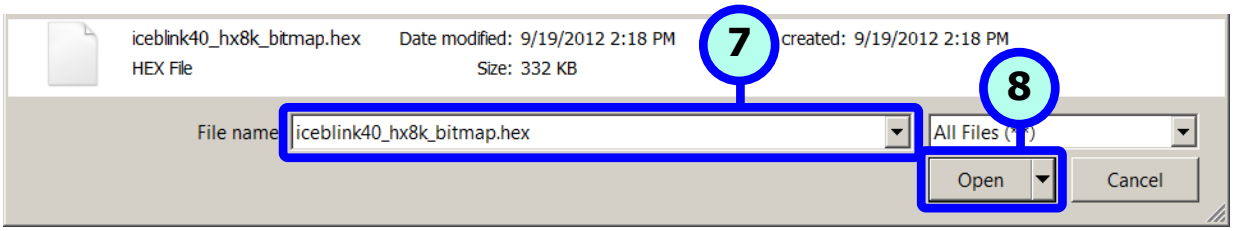
This example assumes that the iCEman40 board is set up to use the on-board USB programmer.



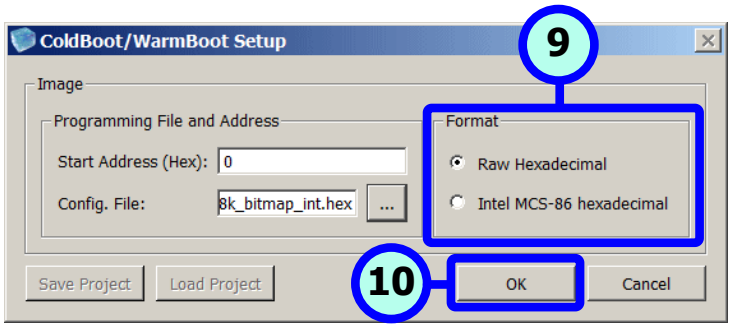
2. Click the dropdown button ( ) under **Programming Hardware**.
3. Select **iCEman40** from the drop list.
4. The iCEcube2 software automatically selects **MP25P80** as the external SPI Flash PROM type.
5. Click **Image Files Settings** to select the configuration bitstream file.



6. Click ... to select the configuration bitstream file.



7. Browse to select the bitmap image, **iceblink40_hx8k_bitmap.hex**.
8. Click **Open**.

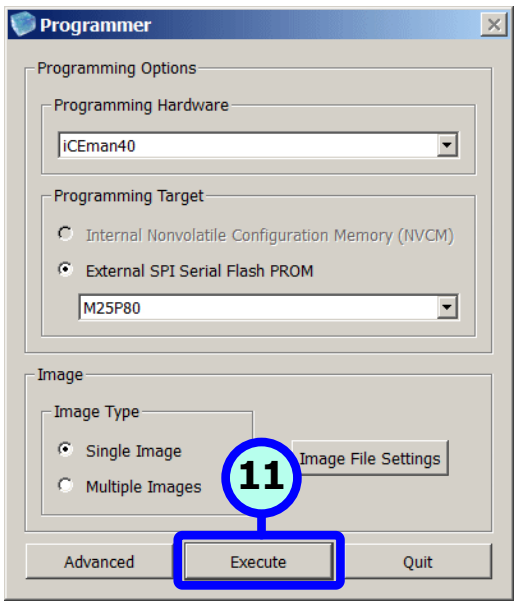


9. The selected file is in raw hexadecimal format, the default setting
8. Click **OK**.

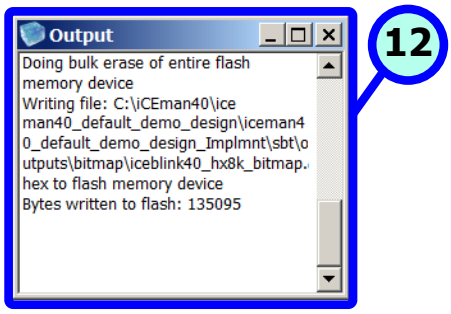
iCEcube2 produces two different types of hexadecimal formats:

- Raw hexadecimal format is saved as **<project_name.hex**
- Intel MCS-86 hexadecimal format is saved as **<project_name>_int.hex**

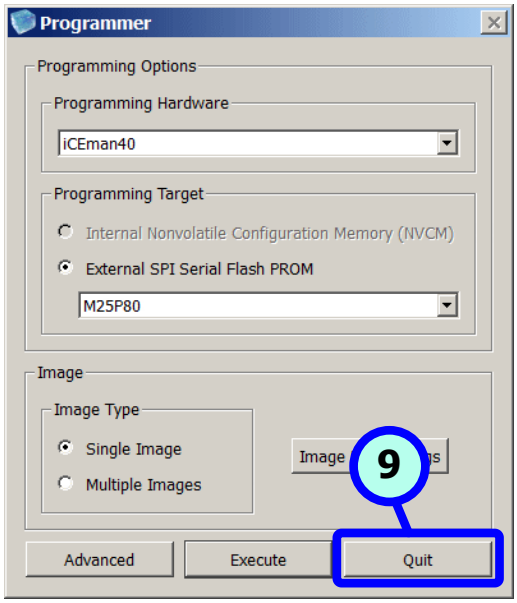
The Intel MCS-86 is widely supported by third-party programming adapters and software.



11 Click **Execute** to start the programming process.



12 A pop-up dialog box provides the status of the programming process. The FPGA will program and begin to operate on the iCEman40 board.

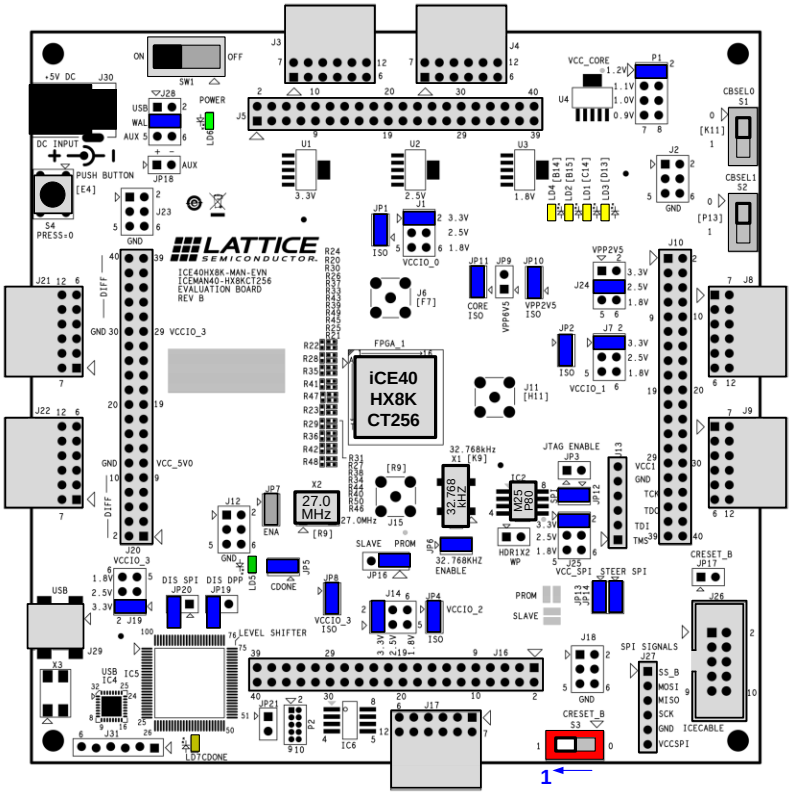


13 Click **Quit** when finished.

Default Jumper and Switch Settings

Figure 43 shows the default jumper and switch settings for the iCEman40 board as delivered.

Figure 43: **iCEman40 Default Jumper Settings (as delivered)**



Bill of Materials (Major Components)

Table 20 lists the major components on the iCEman40 board.

Table 20: **Bill of Material List for Major Components**

Reference	Manufacturer	Part Number	Description
FPGA_1	Lattice Semiconductor	iCE40HX8K-CT256	iCE40 HX-series FPGA in the 256-ball BGA package (0.8 mm ball pitch)
IC2	Micron Technology, Inc.	M25P80	8Mbit SPI serial configuration Flash PROM for FPGA_1. Holds multiple configuration images for ColdBoot and WarmBoot applications.
IC4	Atmel Corporation	AT90USB162-16MU	AVR-based USB programming and debugging interface
U4	Microchip Technology	MCP1825T-ADJE/DC	Adjustable voltage regulator for the FPGA core supply (1.2V, 1.1V, 1.0V, and 0.9V via a jumper setting).
U1, U2, U3	Texas Instruments	TPS78601DCQR	Voltage regulators for the FPGA I/O voltages. One each for 1.8V, 2.5V, and 3.3V.
X1	Epson Toyocom	SG-3030JC 32.7680KB3	32.768 kHz oscillator.
X2	ABRACON	ASVMB-27.000MHZ-LY-T	27.0 MHz oscillator.
IC5	Lattice Semiconductor	iCE40LX1K-VQ100	iCE40HX1K-VQ100 FPGA functions as a voltage level-shifter between the USB controller and FPGA_1.

Additional Information Available on LatticeSemi.com

Additional information about the Lattice Semiconductor iCEman40 board is available via the LatticeSemi.com web site, including ...

- The latest version of this user guide
- Schematics of the board (also see Appendix A: Schematic in this user guide)
- Layout files
- Verilog source files and pin constraint files for the FPGA application shipped with the board
- A compiled bitstream of the default application
- Pinout spreadsheet for the iCE40HX8K-CT256 FPGA

Appendix A: Schematic

Figure 44: **iCEman40 I/O Bank 0 Schematic**

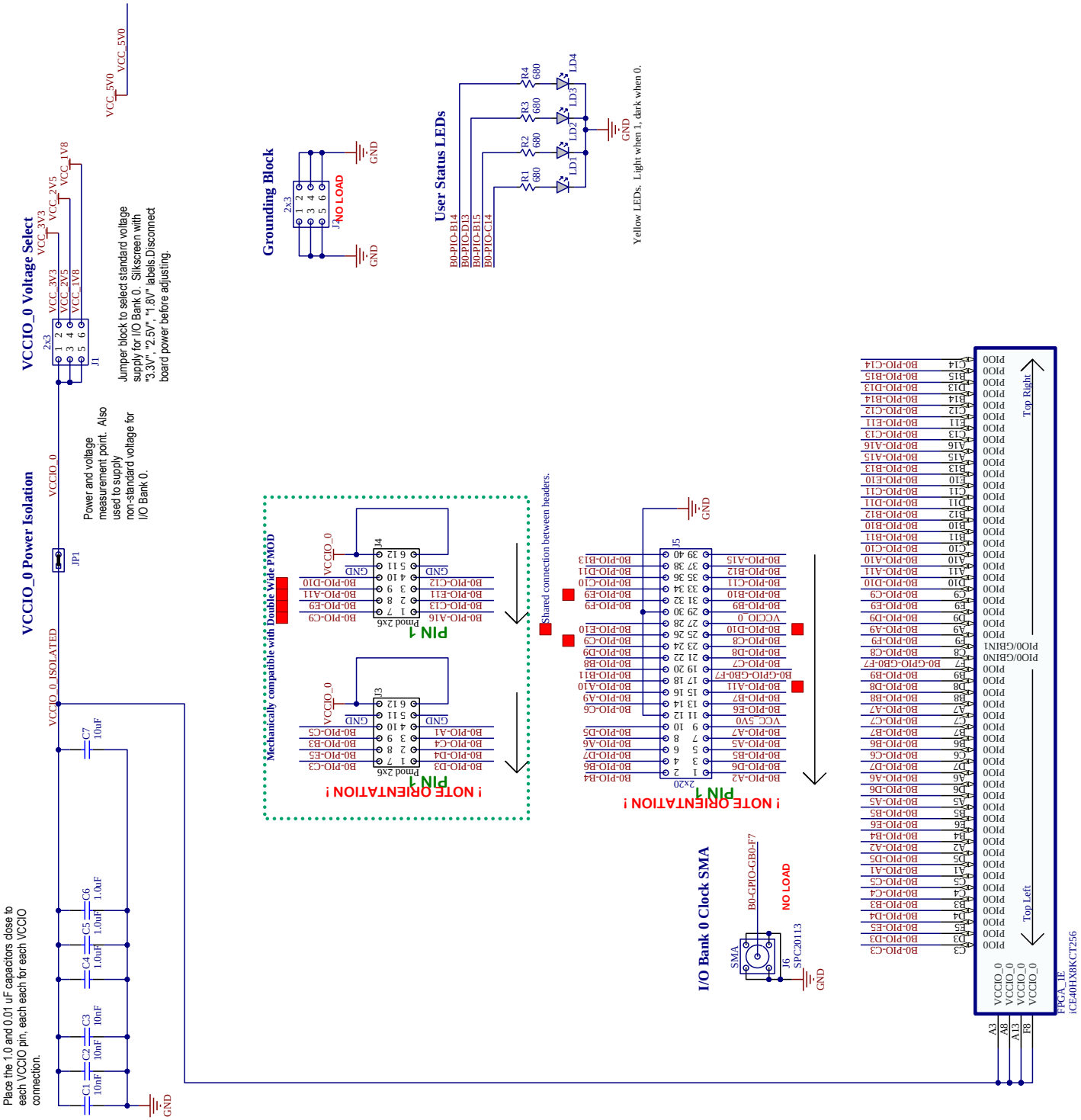


Figure 45: **iCEman40 I/O Bank 1 Schematic**

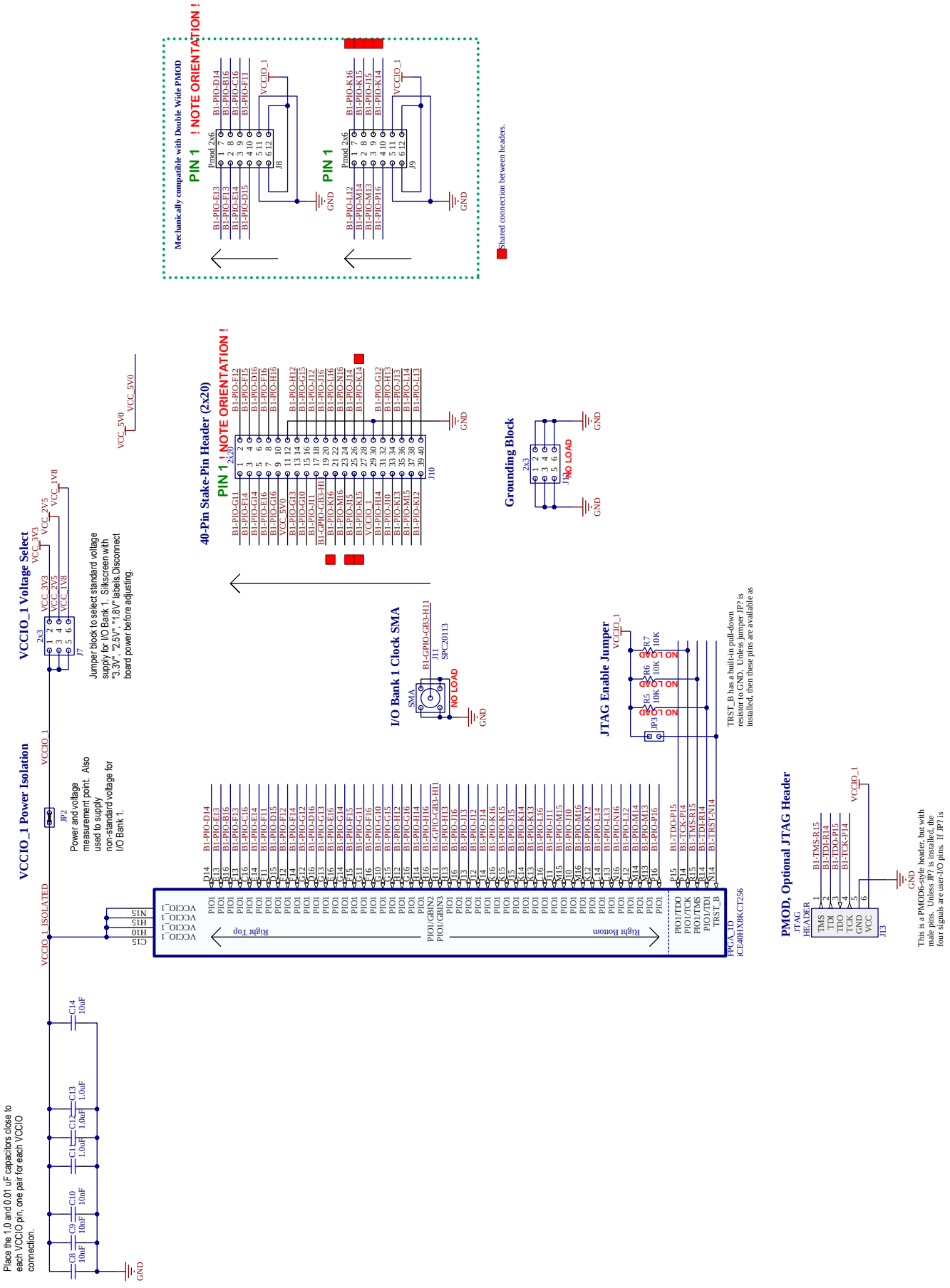


Figure 46: **iCEman40 I/O Bank 2 Schematic**

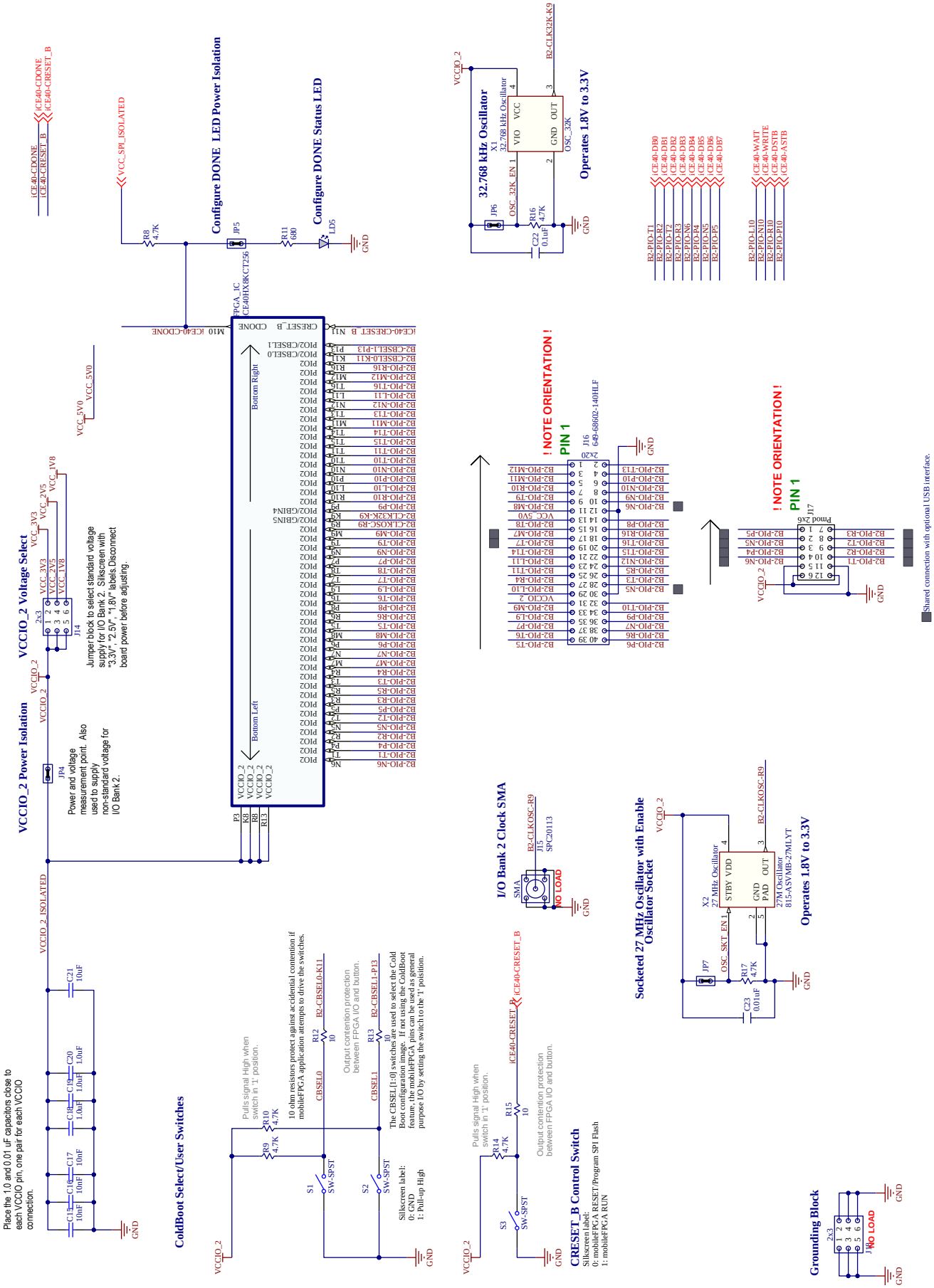


Figure 47: **iCEman40 I/O Bank 3 Schematic**

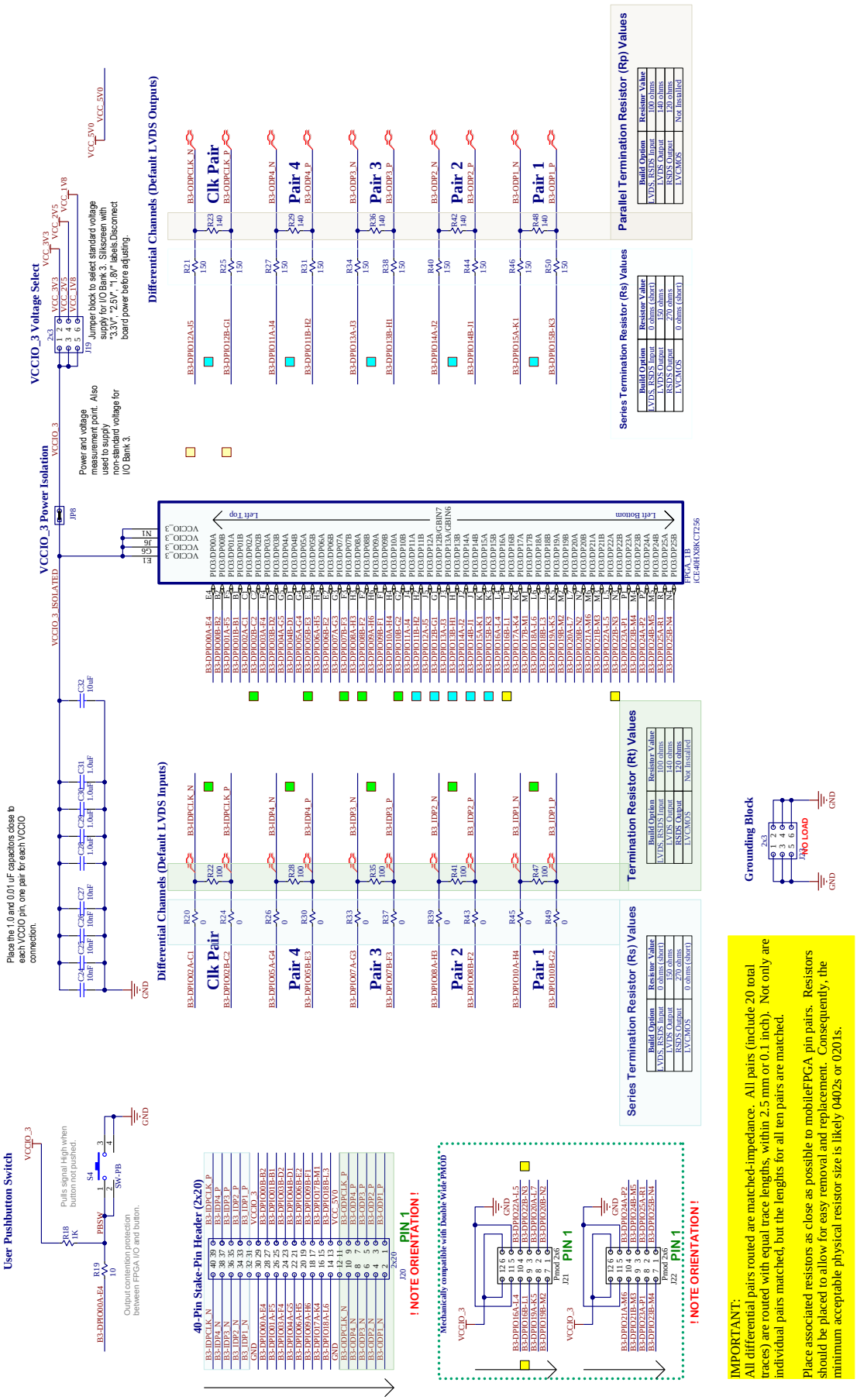


Figure 48: **iCEman40 SPI Bank Schematic**

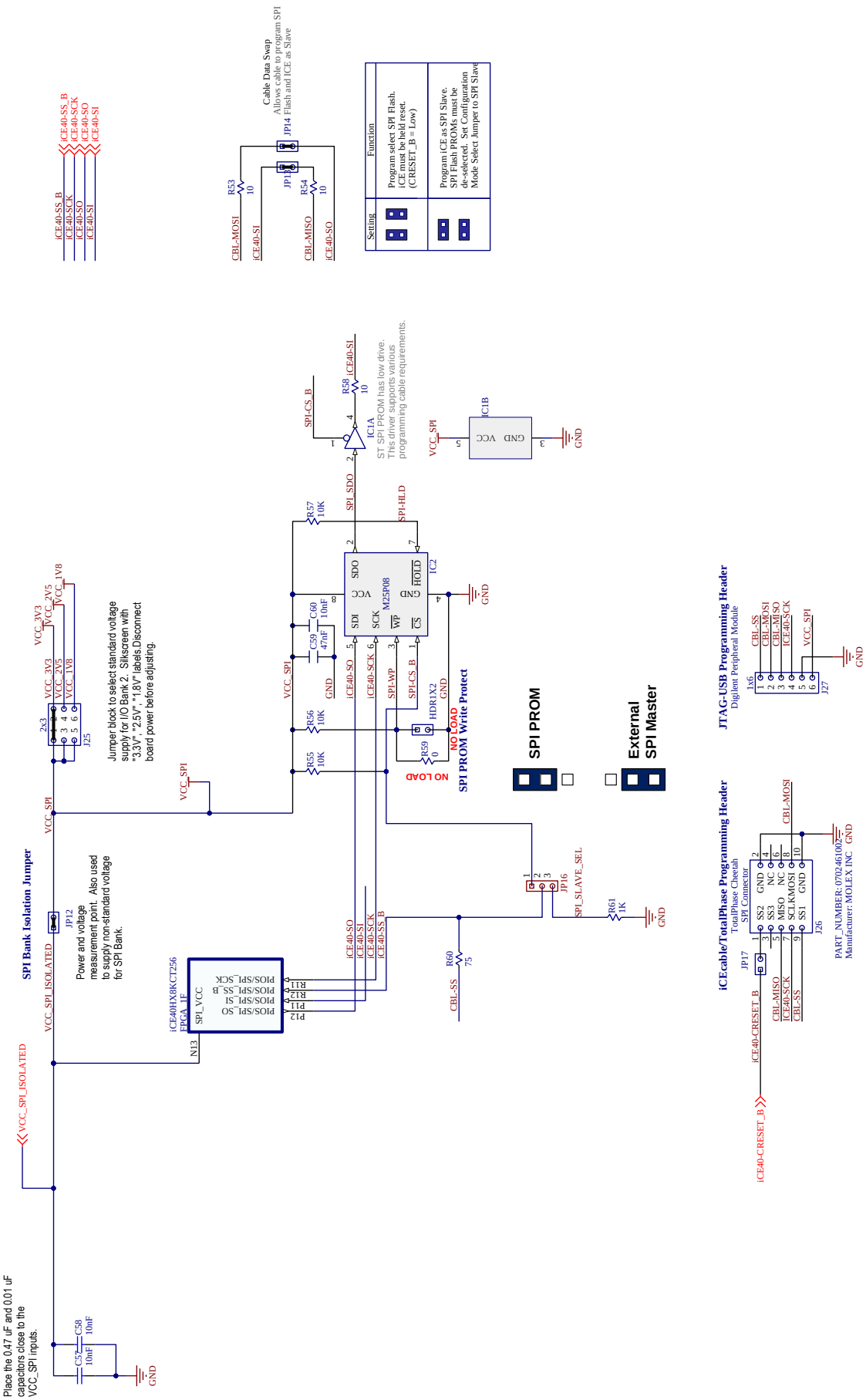


Figure 49: **iCEman40 Power Decoupling and PLL Schematic**

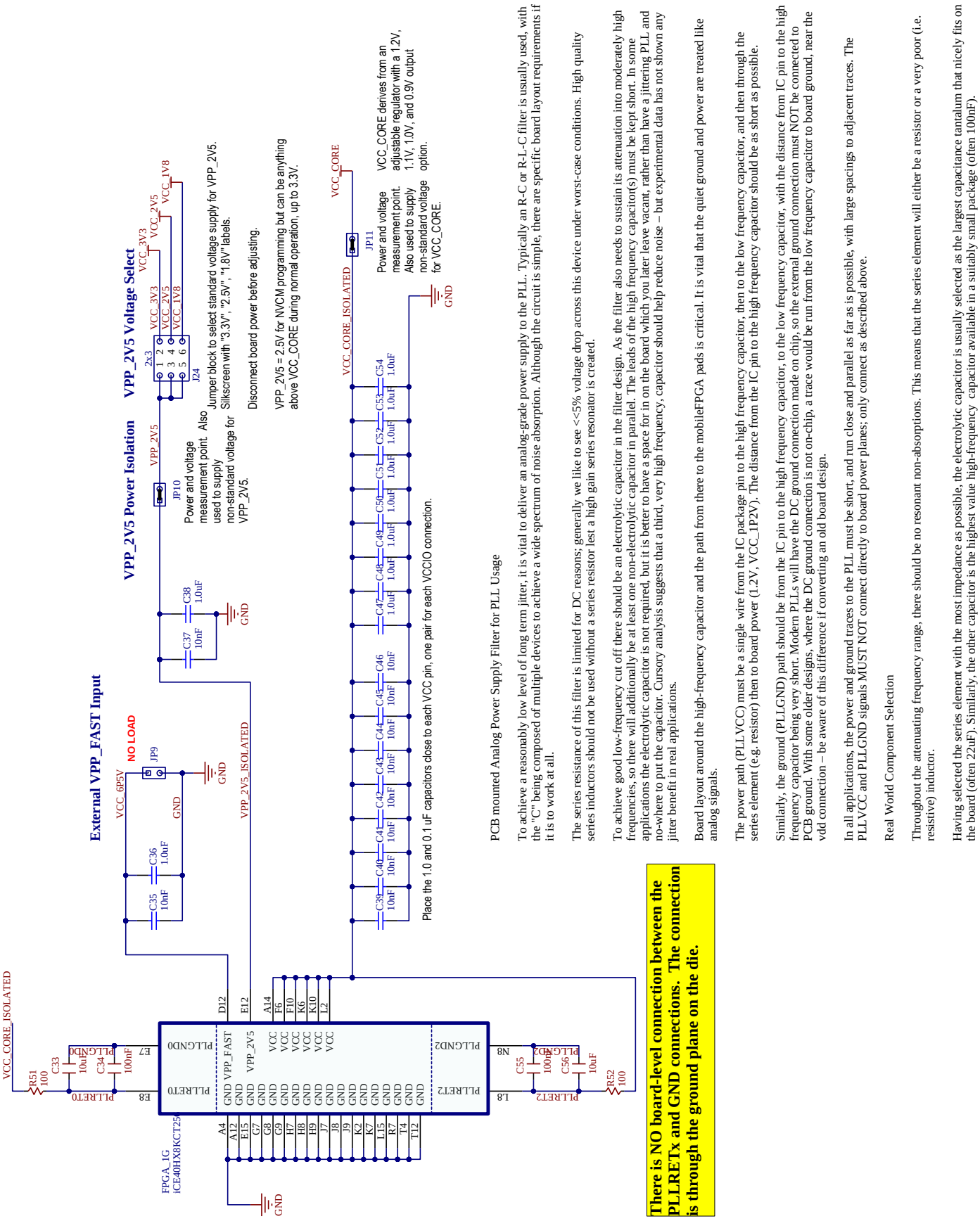


Figure 50: **iCEman40 Power Inputs and Voltage Regulator Schematic**

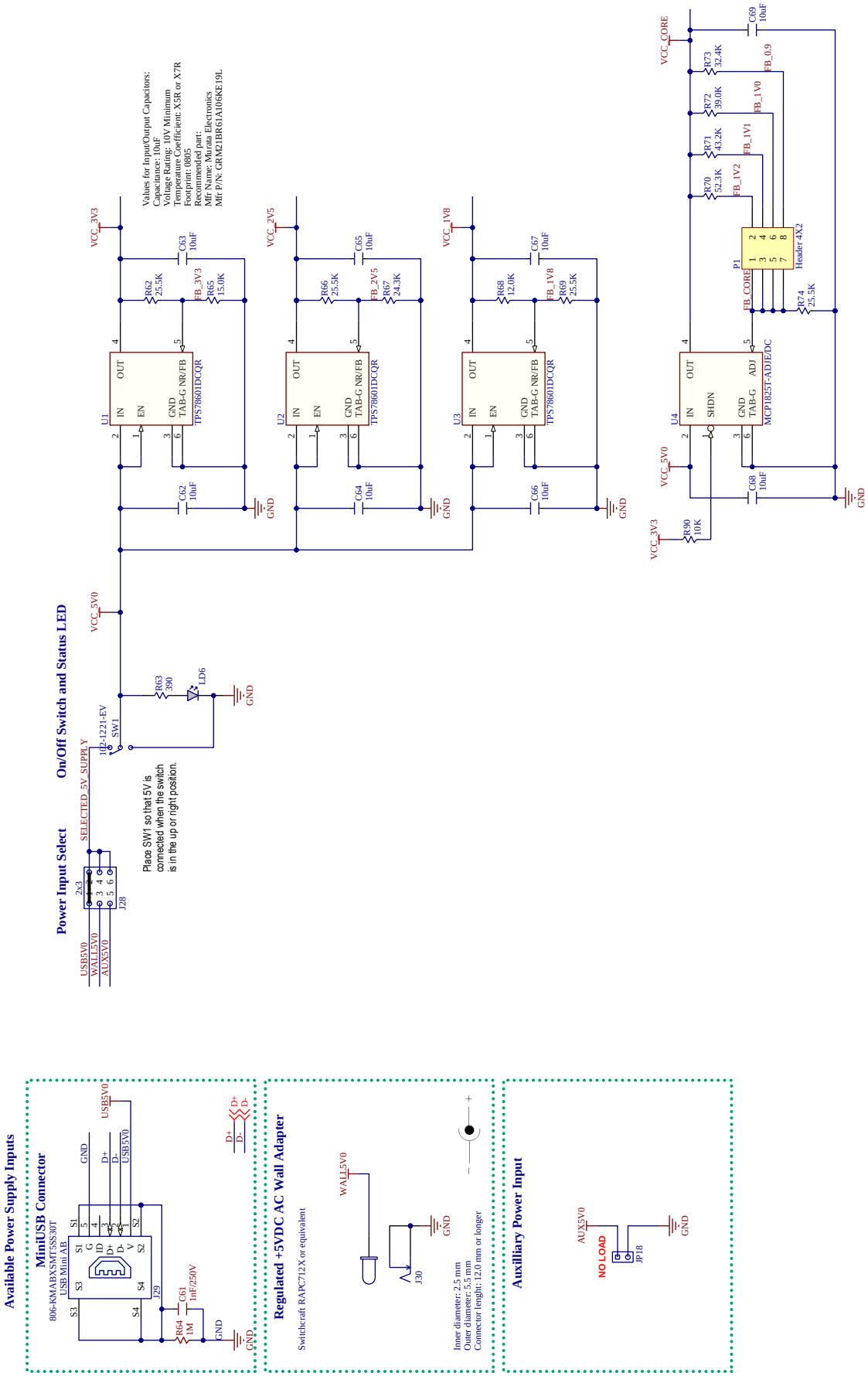


Figure 51: **iCEman40 Power Inputs and Voltage Regulator Schematic**

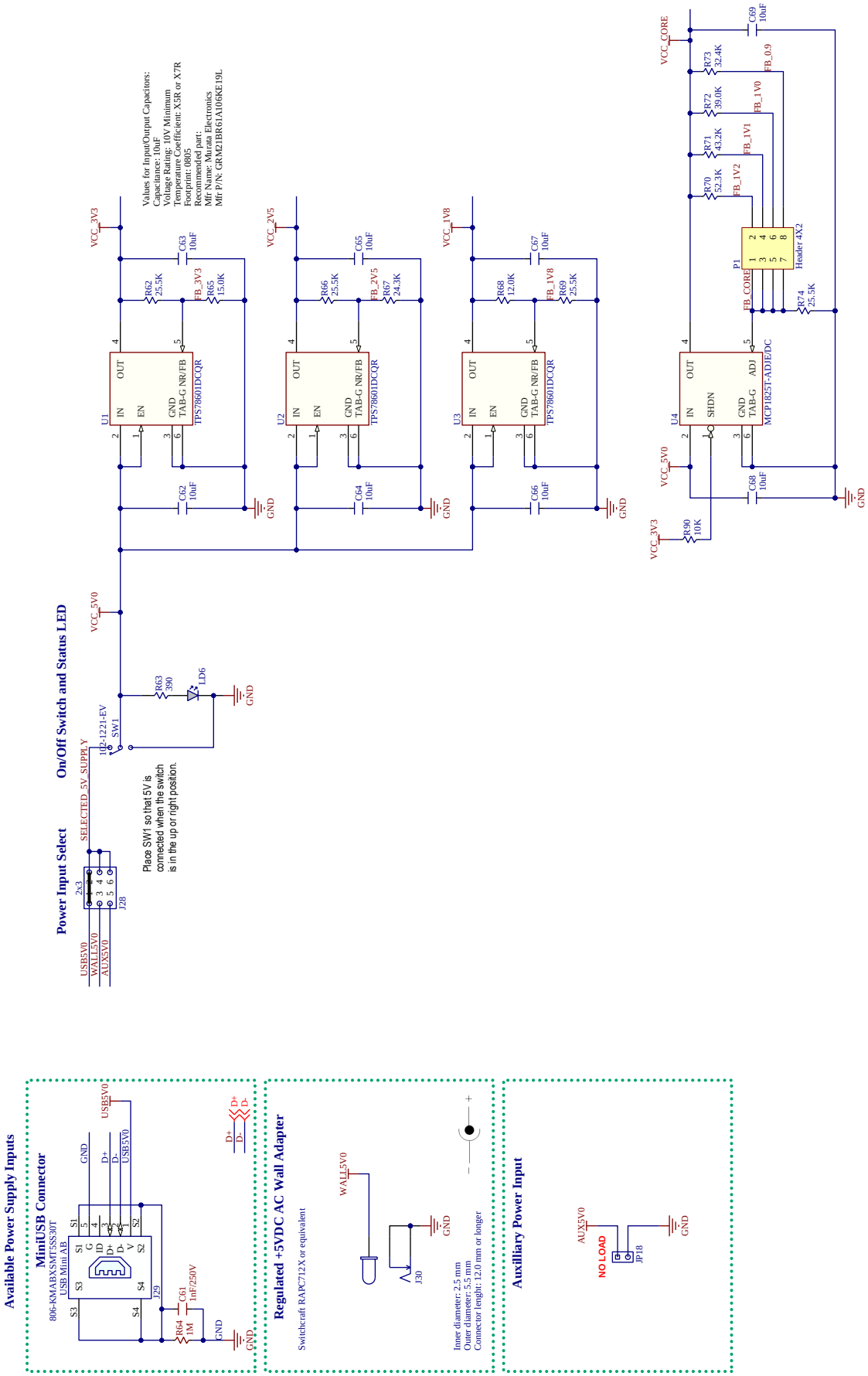
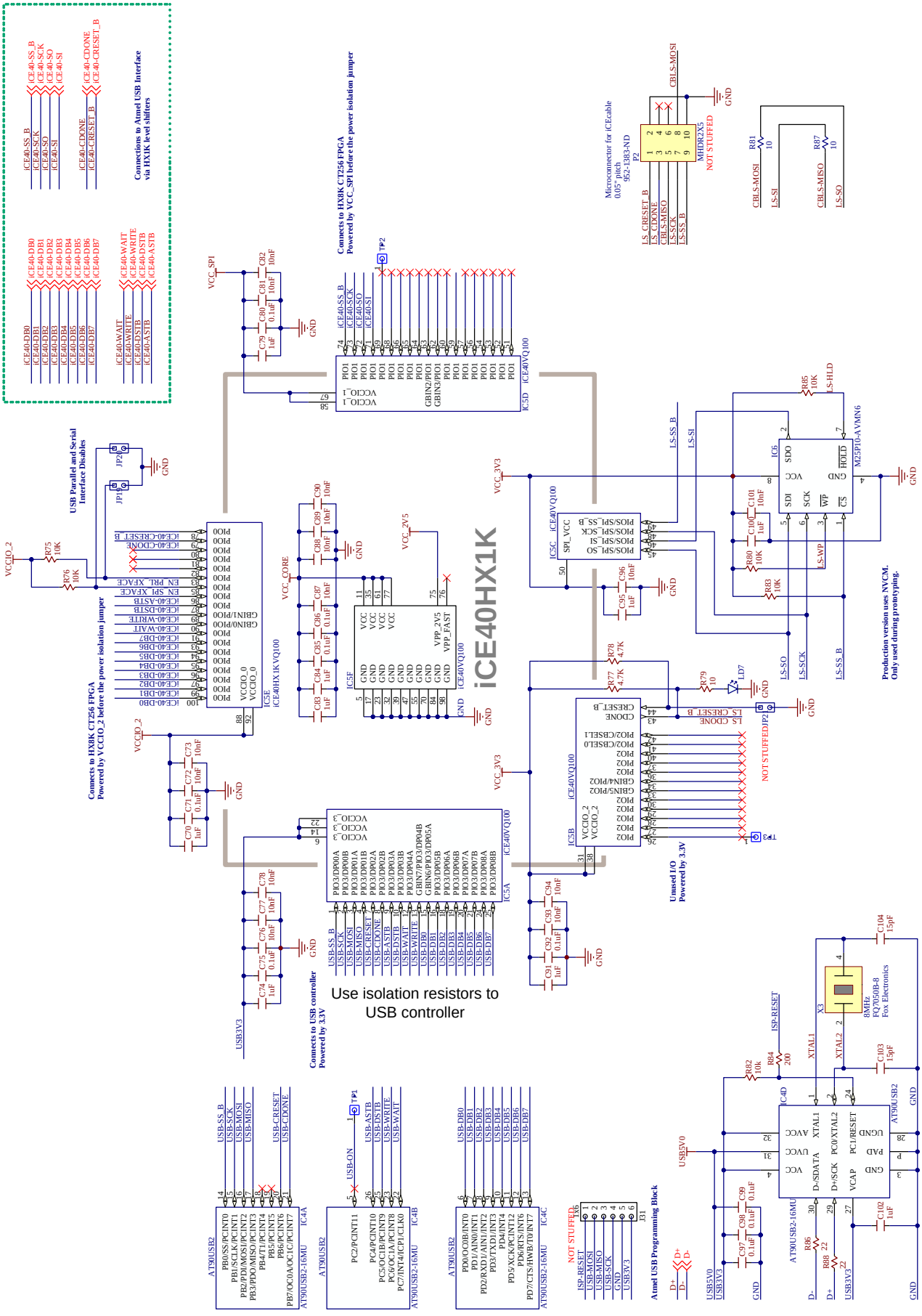


Figure 52: **iCEman40 USB Controller and Voltage Level-Shifter FPGA Schematic**



Revision History

Version	Date	Description
1.0	7-NOV-2012	Initial release.

Lattice Semiconductor Corporation
5555 N.E. Moore Court
Hillsboro, Oregon 97124-6421
United States of America

Tel: +1 503 268 8000
Fax: +1 503 268 8347