*Click **here** for production status of specific part numbers.*

MAX5855 Evaluation Kit

Evaluates: MAX5855

General Description

The MAX5855 evaluation kit (EV kit) contains a single MAX5855 wideband interpolating and modulating RF digital-to-analog converter (DAC) which can directly synthesize 1GHz of instantaneous bandwidth from DC to frequencies greater than 2.3GHz. The MAX5855 EV kit provides a complete system for evaluating MAX5855 performance as well as developing a FPGA plus DAC transmitter solution.

The MAX5855 accepts input data through a five-lane JESD204B serializer/deserializer (SerDes) interface at 9.8304Gbps that is Subclass-0 compliant. The MAX5855 EV kit connects to one FMC connector on the Xilinx VC707 evaluation kit, allowing the VC707 to communicate with the MAX5855's JESD204B serial link interface.

The evaluation kit includes Windows® 7/10-compatible software that provides a simple graphical user interface (GUI) for configuration of all the MAX5855 registers through the SPI interface, control of the VC707 FPGA and temperature monitoring.

Features

- Evaluates the MAX5855 RF DAC Performance, Capability, and Feature Set
- Single 3.3V Input Voltage Supply
- Direct Interface with Xilinx VC707 Data Source Board
- Windows 7/10-compatible Software
- Optional On-Board SPI Interface Control for the MAX5855
- On-Board SMBus™ Interface Control for the MAX6654 Temperature Sensor
- Integrated GUI Controls for VC707 Operation
- Proven 10-Layer PCB Design
- Fully Assembled and Tested

***Ordering Information** appears at end of data sheet.*

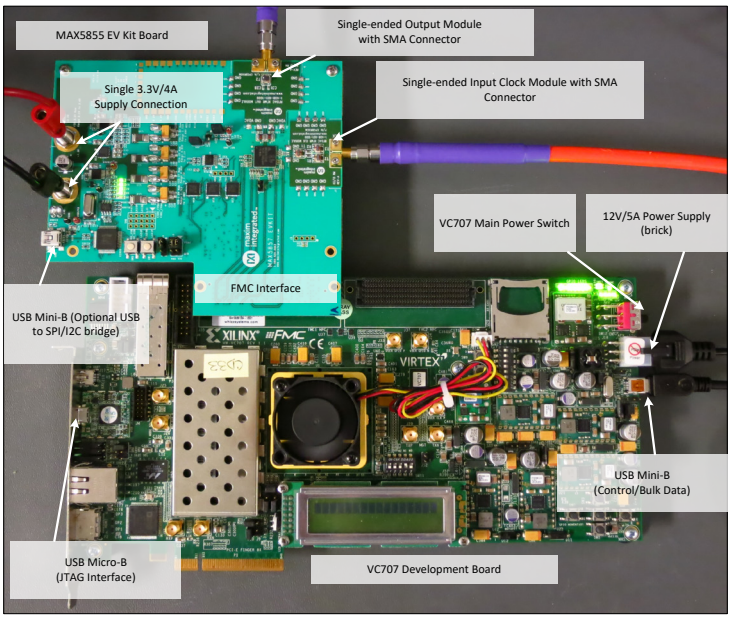


Figure 1. MAX5855 EV kit and VC707 System

Windows is a registered trademark and registered service mark of Microsoft Corporation.

MAX20069 EV Kit Files

FILE	DECRIPTION
MAX5855EVKITSoftwareController.exe	Application program
AppFiles Directory	Directory application support files including the USB_MS_Bulk_Transfer driver
DeviceScripts Directory	Directory with sample MAX5855 configuration scripts and PERL scripts for generating additional configurations
PatternFiles Directory	Directory with sample pattern files
VC707Files Directory	Directory with FPGA programming file and supporting documentation
Screenshots Directory	Directory with example spectrum analyzer screen captures
Miscellaneous DLLs to include FTD2XX_NET. dll, ftd2xx.dll, LibUsbDotNet.dll, libMPSSE.dll, StatusIndicatorTest.dll and MaximStyle.dll	Supporting DLL files for software operation

Maxim recommends using the default installation path (**c:\MaximIntegrated\MAX5855EVKIT**). If an alternate path is desired it must NOT contain any spaces or the Xilinx LabTools will not be accessed properly. This step should take ~ 2 minutes. The EV kit is fully assembled and tested. Follow the steps below to verify board operation:

Initial Setup

Required Equipment

- Window PC (Win-7, Win-10), with one or two available USB 2.0 ports
- Spectrum Analyzer (Agilent PXA or equivalent)
- RF signal generator (R & S® SMF100A or equivalent)
- 3.3V, 4A power supply for MAX5855 EV kit
- User-supplied Xilinx VC707 EV kit
  - VC707 board.
  - 12V/5A power cube.
  - 1 each USB-A to Mini-B cable for interfacing with the VC707 and MAX5855.
  - 1 each USB-A to Micro-B cable for programming the VC707.
- Low-loss SMA/SMA cables as needed for connections to the Spectrum Analyzer and Signal Generator
- **Included in the MAX5855 EV kit**
  - MAX5855 Evaluation Kit board.
  - Two 1” stand-offs with screws.
  - USB-A to Mini-B cable, for interfacing with the VC707/MAX5855 or directly to the EV kit board.

Required Software and Drivers

The MAX5855 EV kit software controller application requires the following third-party software components and drivers to be installed. *Refer to Appendix I of this document for additional information on this installation process*. It is highly recommended that the target PC be connected to a local area network and have access to the Internet, this allows for automatic download and updates of some drivers. This process may take **30 minutes** or more to complete.

- **Xilinx ISE 14.7 LabTools**
  This is a free tool set used for programming the VC707 evaluation board. No software license is required to use these tools. However, a Xilinx Login will be required for downloading.
- **Xilinx Drivers**
  After the LabTools have been installed on a PC, the VC707 USB interface drivers can be installed.
- **LIB USB Driver**
  Manual installation of this driver is required after the VC707 is programmed enabling the USB 2.0 bulk transfer port.
- **Microsoft .NET Framework 4**
  The Microsoft .NET Framework 4 is required by the **MAX5855EVKITsoftwareController** application. The Win-7 operating system requires manually installing the .NET Framework package using the installer copied to the *{installationpath}\MAX5855EVKIT\AppFiles\ThirdParty* during the EV kit software installation.

Install the MAX5855 EV kit Software

The MAX5855 EV kit Software Controller application can be obtained from the **www.maximintegrated.com** website. Access the **MAX5855RFDACEVKitSoftwareInstaller.exe** file to install the software.

Maxim recommends using the default installation path **(c:\MaximIntegrated\MAX5855EVKIT)**. If an alternate path is desired, it must NOT contain any spaces or the Xilinx LabTools will not be accessed properly. This step should take less than **2 minutes**.

Setup and Connect the MAX5855 EV Kit Hardware

1) Install the two 1" stand-offs included with the MAX5855 EV kit. Stand-offs should be installed on the RF DAC output side of the board.
2) Verify all jumpers on the MAX5855 EV kit board are in the default position; refer to Table 1.
3) Connect the MAX5855 EV kit board to the VC707 board using the FMC Interface connector shown, as shown in Figure 1.
4) Connect the 3.3V/4A supply to the MAX5855 EV kit board and enable the supply's output. Verify the four, green, LED board supply indicators are lit.
5) Connect the RF generator to the CLK Input with a low-loss SMA cable and set the frequency to 491.52MHz with output power at +2dBm, and enable the output.
6) Turn on the VC707 by sliding the VC707 Main Power Switch (SW12) to the ON position. Verify ALL the LEDs on the VC707 are momentarily lit; the GPIO LEDs should then begin sequencing.
7) Make the USB connections, see Figure 1 for locations
   a. Connect the USB A – micro B cable (JTAG) from Xilinx VC707 Eval board to the PC
   b. Connect the USB A – mini B cable (Ctrl/Bulk) from Xilinx VC707 Eval board to the PC

Please ensure that all the required third-party software and drivers are installed before proceeding to the next section.

Configure the MAX5855 EV kit Graphical User Interface

A few items need to be configured during the first execution of the GUI software.

1) Start the MAX5855 EV kit software. Double-click on the desktop icon or the **MAX5855EVKITsoftwareController.exe** executable located in the **C:\MaximIntegrated\MAX5855EVKIT** folder. A splash screen will be displayed while the USB connections are established (Figure 2) followed by the window shown in Figure 3.
2) Click the <SPI Communication to DAC from FPGA> button and the primary GUI software window will be loaded (see Figure 4).

Table 1. MAX5855 EV kit Jumper Settings

JUMPERS	POSITION	EV KIT FUNCTION
JU1	Installed*	Normal Operation
JU4	Installed*	Power for U4 – MAX6161 – external reference
	Not Installed	MAX6161 NOT powered
JU5	Installed*	MAX5855 external reference connected
	Not Installed	MAX5855 using internal reference
J4	1-2*	SPI control connected through FPGA
	2-3	SPI control connected to on-board USB
J10	1-2*, 4-5*	SCL, SDA connected through FPGA
	2-3, 5-6	SCL, SDA connected to on-board USB

**Default Position*



Figure 2. Splash Screen

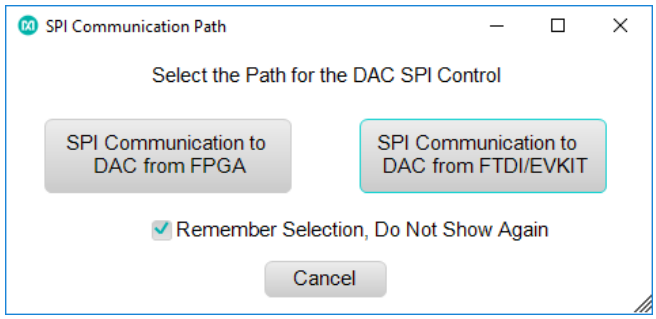


Figure 3. Communications Selection Window

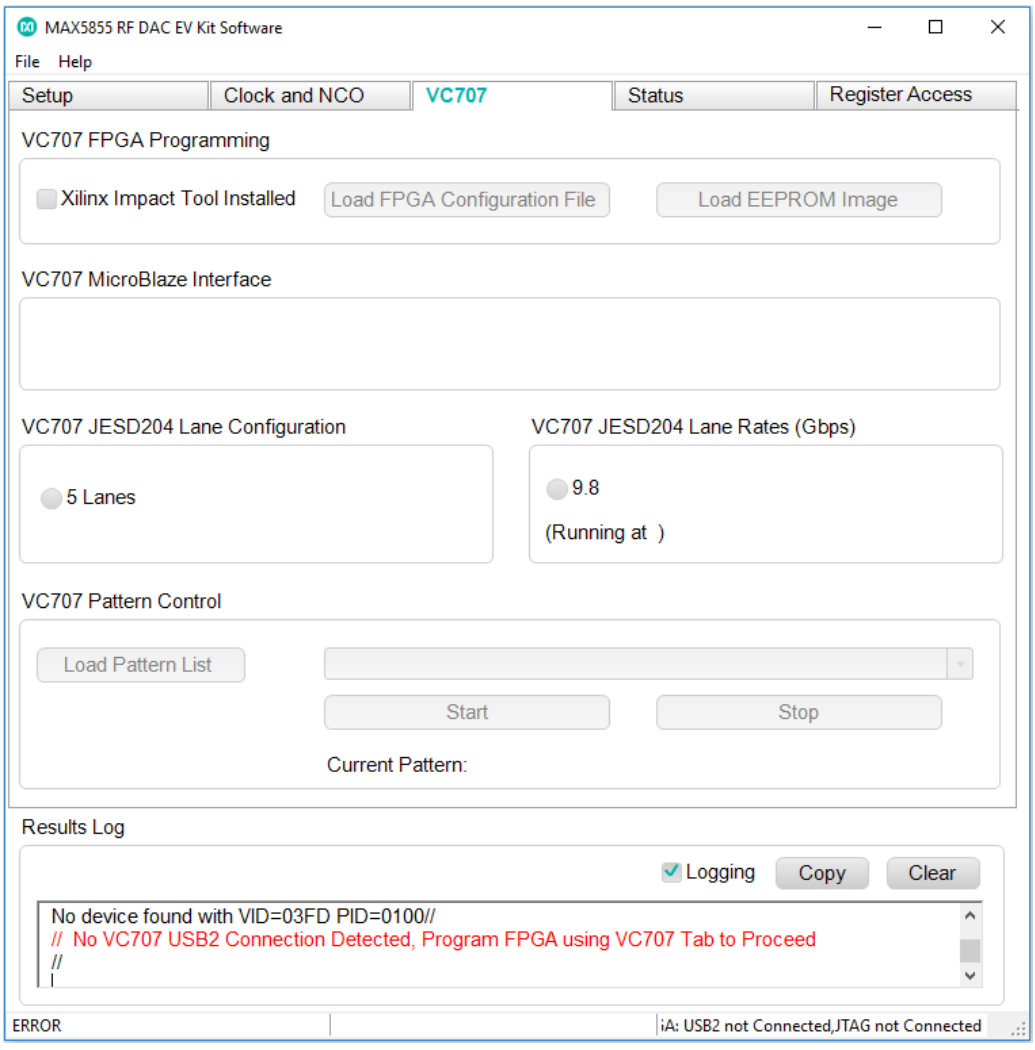


Figure 4. MAX5855 EV kit GUI—Initial View

3) Load the FPGA configuration
   a. Click on the <Xilinx Impact Tool Installed> check-box and a File Browser window will open
      i. Browse to the directory where the impact.exe program is located. If the default installation location was used for the Xilinx Lab Tools, the path will be:
         • For a 32-bit OS – **C:\Xilinx\14.7\LabTools\LabTools\bin\nt**
         • For a 64-bit OS – **C:\Xilinx\14.7\LabTools\LabTools\bin\nt64**
      ii. Select the **impact.exe** file
   b. Click the <Load FPGA Configuration File> button. A file browser will open in the **C:\maximintegrated\MAX5855EVKIT\VC707Files** folder.
   c. Select the **MAX585X_VC707Prog_15Jun2017.bit** file. A progress bar will display while the FPGA is configured, which should take < 2 minutes.

After completing the FPGA configuration, the PC will establish a connection to the new USB2.0 port on the FPGA. It should appear as a USB Mass Storage Device in Device Manager (Figure 6).

Note: *Before proceeding, ensure any USB flash drives have been ejected from the PC.*

4) Update the USB 2.0 port driver
   a. Open the Windows Device Manager
      i. Under Control Panel, click on Device Manager or search for "Device Manager"
   b. Select the USB Mass Storage Device and right click to select the "Update Driver Software…" option
   c. Select "Browse my computer for driver software"
   d. Select "Let me pick from a list of devices on my computer"
   e. Click the <Have Disk> button
   f. Click the <Browse> button in the "Load from Disk" pop-up window
   g. Browse to: C:\MaximIntegrated\MAX5855EVKIT\AppFiles\ThirdParty\USB_MS_Bulk_Transfer and select the USB_MS_Bulk_Transfer.inf file.
   h. A window will appear showing Maxim Integrated as the signature for the LIBUSB2 driver. Click on the <Install> button.
   j. The MAX5855 software may show a window indicating it has encountered a problem; click on the <Close> button to continue

5) Reboot the PC and power-cycle the FPGA system
   a. Turn off the VC707 by sliding switch VC707 Main Power Switch to the OFF position
   b. Disable the RF generator output (clock signal)
   c. Turn off the DC 3.3V/4A power supply to the EV kit
   d. Reboot the PC

After the PC has booted, the drivers will be properly configured for use with the MAX5855 software.

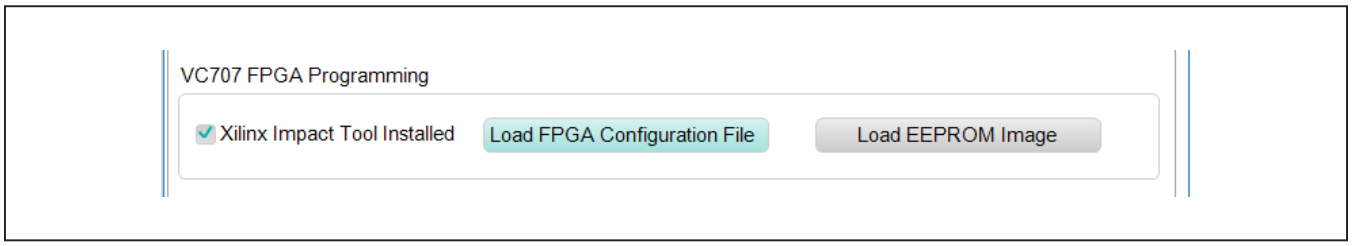


Figure 5. FPGA Configuration

Quick Start Procedure

Power-Up the MAX5855 EV Kit Hardware

1) Set the RF Generator to the desired baseband clock rate (491.52MHz) and the power to +2dBm, DO NOT ENABLE the RF output yet
2) Connect the DAC output to the spectrum analyzer
3) Enable the 3.3V power supply. Verify the four, green, LED board supply indicators are lit
4) Enable the RF generator output
5) Turn on the VC707 by sliding the VC707 Main Power Switch to the ON position (see Figure 1)
6) Verify all LEDs on the VC707 are lit, and the GPIO LEDs are sequencing (for an unprogrammed FPGA)

Run the MAX5855 EV kit Software

1) Start the MAX5855 EV kit software.
2) If the VC707 was powered cycled without a programmed EEPROM, reload the FPGA configuration
   a. Click the Xilinx Impact Tool Installed check box
   b. Click the <Load FPGA Configuration File> button
      i. A file browser opens in the VC707 folder
      ii. Select the **MAX5855_DataSource.bit** file and click the <Open> button
   c. Verify that the lower-left corner of the window states “FPGA Programmed”; this indicates the GUI is connected to the MAX5855 EV kit through the FPGA (Figure 6)
   d. Verify lower-right corner of the app states “FPGA: USB2 Connected, JTAG Connected” indicating these specific port connections have been established (Figure 6). The USB2 should not indicate a connection until the FPGA has been programmed.

Note: These status notes indicate when FPGA bulk USB 2.0 port and JTAG interfaces are connected and operating properly.

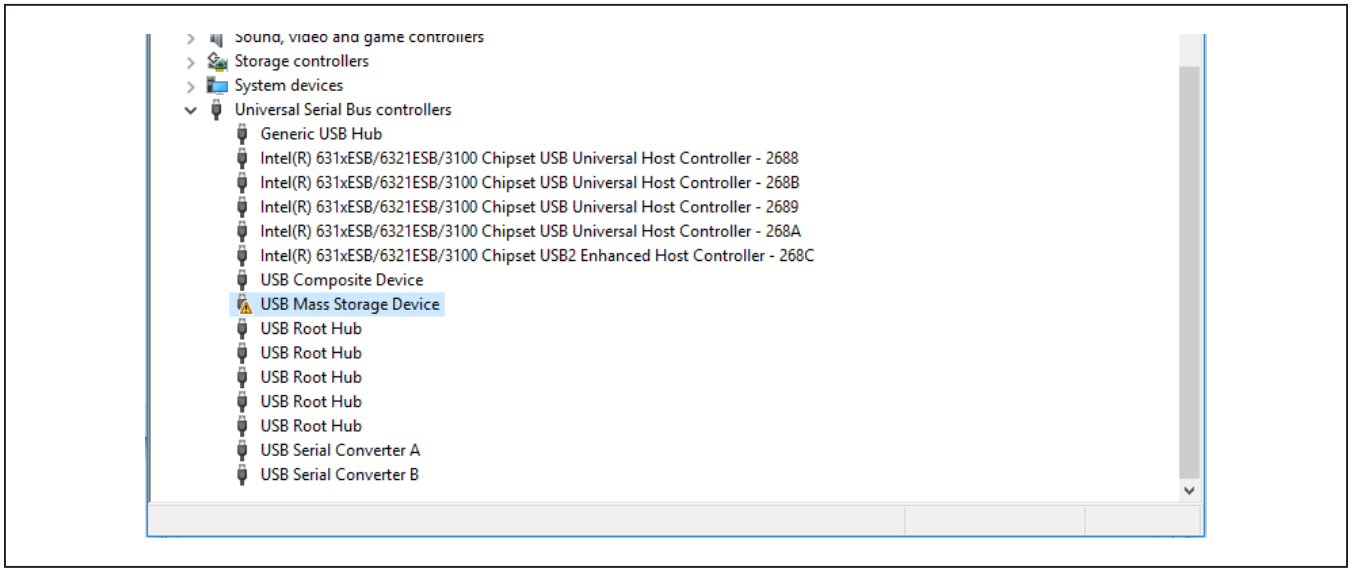


Figure 6. Device Manager Window

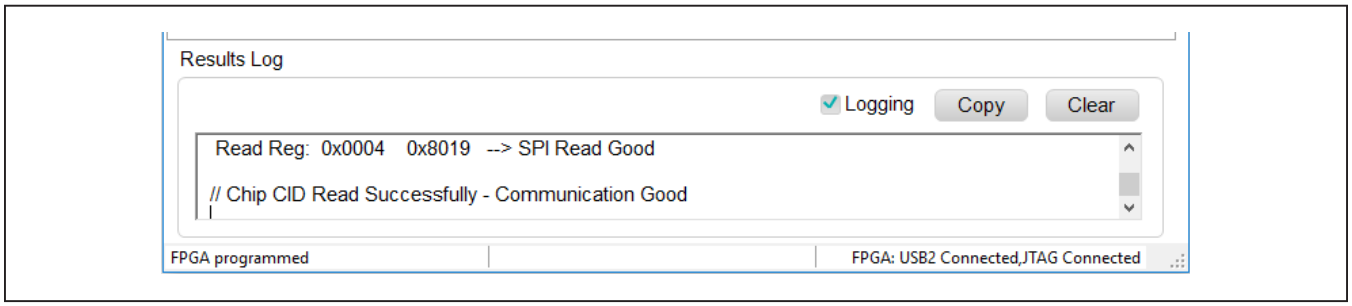


Figure 7. FPGA Connected

3) To quickly load a default pattern…
   a. Click on on the <Setup> tab of the GUI
   b. Click on one of the <Test Setup…> buttons
   c. Example:
      i. Click on the <Test Setup #2: 8 SCQAM, 256QAM, CF=800MHz> button
      ii. Set the center frequency of the spectrum analyzer to 800GHz
      iii. Click on the “Display Screenshot” selector switch
      iv. Confirm that the spectrum displayed on the analyzer matches the screenshot example (Figure 8)

Loading a Pattern Manually

1) Click on the <Setup> tab of the GUI
2) Click the <Load Settings> button
3) Select a configuration file: **MAX585x_DAC5898p24_CLK491p52_6L10G_RCLKDIV2_SRAMEN.cfg**
4) Click on the <VC707> tab of the GUI
5) Click on the <Load Pattern List> button
6) Select a pattern list file, for example: **TestLoadPatterns.txt**
   a. Wait for the patterns to load
7) Click on the “Select Pattern” drop-down list
8) Select a pattern, for example: “MAX585X_AnxB_8Ch_fs_1.2288E+009__-12dB.csv”
9) Click on the <Start Pattern> button
10) Click on the <Setup> tab
11) If the signal is not visible on the spectrum analyzer
    a. Check if the output is muted by clicking on the <Setup> tab and unmute the device, by clicking on the “Hardware Mute” switch
    b. Check that the center frequency of the spectrum analyzer matches the center frequency of the config file / NCO settings

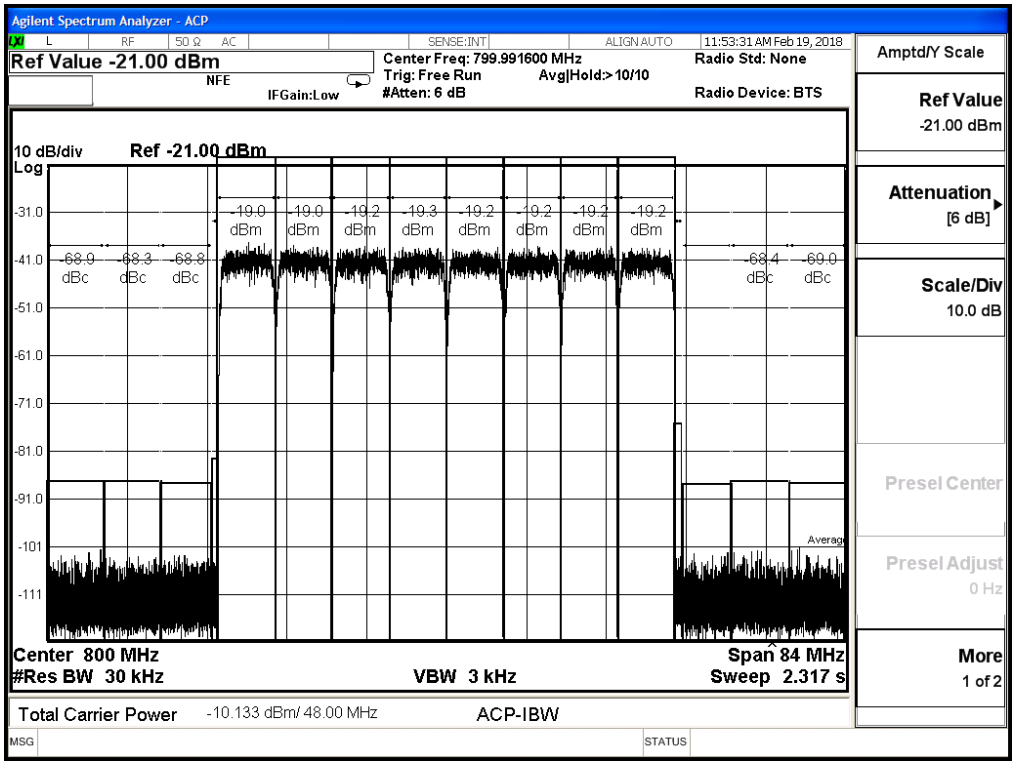


Figure 8. Spectral Output – 8 SCQAM at 800MHz

Detailed Description

Detailed Description of Hardware

MAX5855 EV Kit Printed Circuit Board

The MAX5855 EV kit board is manufactured on a 10-layer, 1oz copper, FR4 and Rogers 4350B dielectric stack-up PCB. Layers 2, 4, 6, and 9 are ground planes matched to controlled impedance, 50Ω differential, high-speed traces on the outer layers. All internal power planes (layers 5 and 7) and signal routing planes (layers 3 and 8) have copper ground pours in the unused areas to provide additional decoupling and to ease manufacturability.

Control Interface

The MAX5855 EV kit board provides two forms of communication and control interfacing to the RF DAC and the temperature monitor: a pass-through from the FPGA system and an on-board USB Interface. The FPGA pass-through provides a Serial Port Interface (SPI) to control the MAX5855 RF DAC, and a SMBus interface to control the MAX6654 (temperature monitor). The on-board USB interface uses an FTDI4232 device which provides the SPI and I2C bus signals, as well as GPIO controls for the hardwired MUTE, INTB and RESETB signals on the MAX5855. The FPGA pass-through and the on-board FTDI interface are selected with jumpers J4 and J10 which in-turn use CMOS switches and level translators to route the incoming control signals as needed.

The default settings of J4 and J10 are for using the FPGA pass-through interface. To use the on-board FTDI, switch the J4 and J10 jumpers as shown in Figure 10b.

Interface Modules

The evaluation kit employs two modules to allow for easy interfacing to Signal Generators and Spectrum Analyzers.

The Clock Input Module (RFDAC_XFMR_CLK_MODULE) converts a single ended Signal Generator output to a differential signal which in-turn, drives the CLKP/CLKN inputs of the MAX5855. The Clock Input Module can be removed from the MAX5855 EV kit board and a differential signal path can be substituted. The method for reconfiguring the clock input is the following:

1) De-solder and remove the Clock Input Module from the MAX5855 EV kit
2) Populate R61 and R62 with 0Ω resistors
3) Mechanically connect two Rosenberger, edge-launch SMAs to the MAX5855 EV kit at J7 and J8
4) Solder the center conductors of J7 and J8 to the differential traces

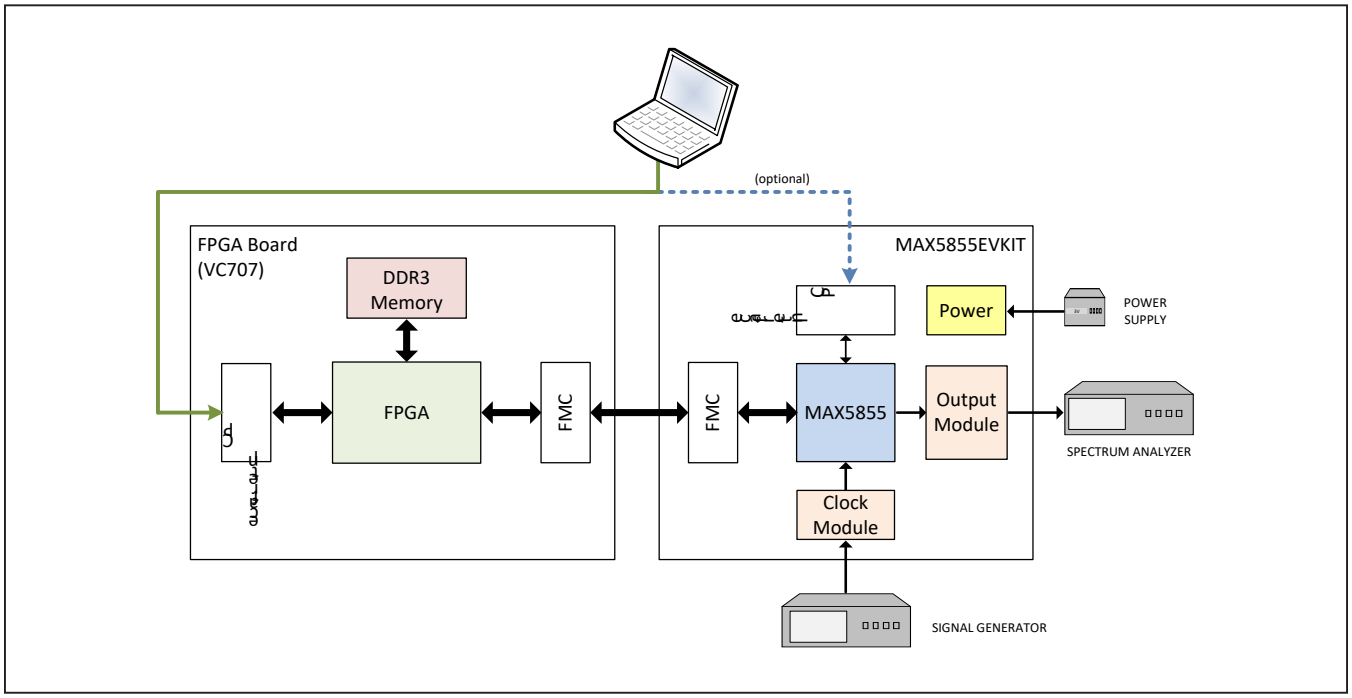


Figure 9. MAX5855 Evaluation System Block Diagram

The Output Module (RFDAC_XFMR_OUT_MODULE) converts the differential output of the MAX5855 RF DAC to a single ended 50Ω output suitable for driving the input of a 50Ω Spectrum Analyzer. The Output Module can be removed from the MAX5855 EV kit board and a differential signal path can be substituted. The method for reconfiguring the DAC output is the following:

1) De-solder and remove the Output Module from the MAX5855 EV kit
2) Populate C93 and C99 with 0.01µF capacitors
3) Mechanically connect two Rosenberger, edge-launch SMAs to the MAX5855 EV kit at J2 and J6
4) Solder the center conductors of J2 and J6 to the differential traces

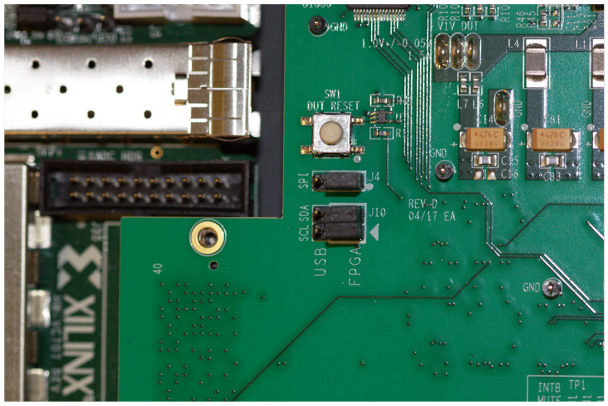


Figure 10. MAX5855 EV Kit Jumpers. 10a – default FPGA pass-through interface

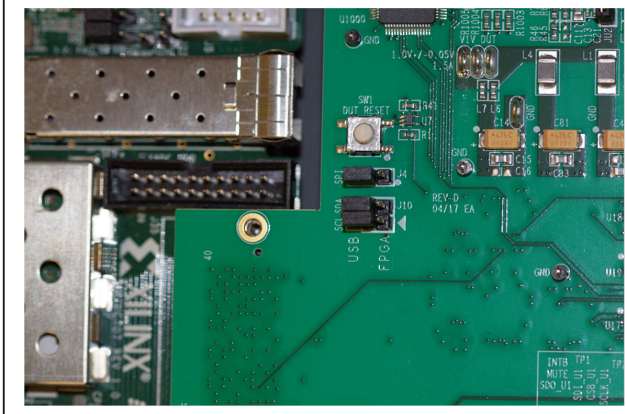


Figure 10. MAX5855 EV Kit Jumpers. 10b – on-board FTDI Interface

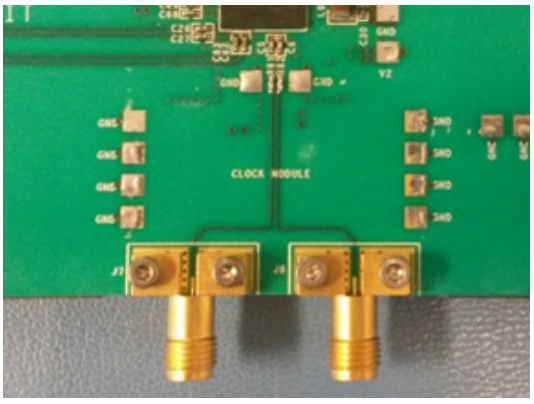


Figure 11. MAX5855 Differential Clock Input

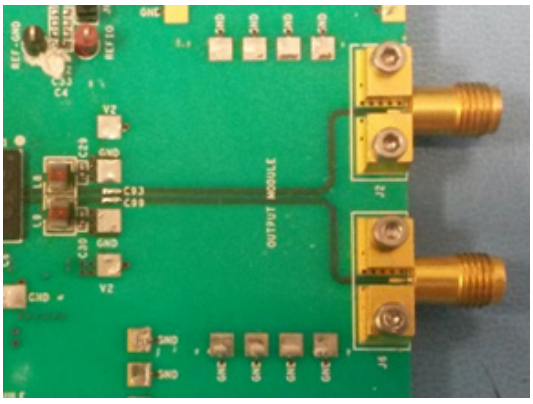


Figure 12. MAX5855 Differential DAC Output

Power

The MAX5855 EV kit board requires a single +3.3V, 4A power supply connected to the board through two "banana" jacks (marked +3.3V and GND) or a set of wire loops that can be used with EZ-Hooks (also marked +3.3V and GND).

The +3.3V supply is used by the various support circuits including two MAX8527 linear regulators (LDOs) which provide the 1.8V supply rails for the MAX5855 and various support circuitry plus a MAX8556 LDO which supplies the 1.0V level to the MAX5855. One LDO is used for each supply rail, however the LDO outputs are isolated between analog and digital domains by on board filter networks. The PLL supplies for the MAX5855 are isolated from the analog domain through additional filtering.

The operational status of each supply can be visually identified by LEDs on the board. When primary power is supplied to the 3.3V VIN on the board, D4 will light green immediately. When the 1.8V and 1.0V rails are within 10% of their nominal output voltages Power-OK lines will light their respective LED indicators.

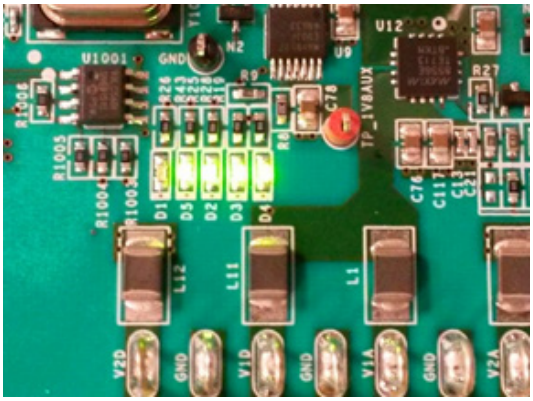


Figure 13. MAX5855 EV Kit LEDs

Temperature Monitoring

As described in the *Detailed Description of Software – Status Tab* section, an alarm threshold can be set for the MAX5855 device temperature. When this threshold temperature is exceeded, the ALERT output of the MAX6654 Temperature Monitor is asserted (active low) and D1 is lit as a visual warning. This is a latched output, so the alert needs to be cleared manually with the GUI software.

DAC Reference

The MAX5855 EV kit includes a MAX6161 precision reference for use as an external voltage level for the RF DAC. Power for the MAX6161 is supplied through jumper JU4, while JU5 connects the MAX6161 output to the MAX5855 VREF input.

Data Interface

The MAX5855 EV kit directly connects to the VC707 FPGA board through the HPC-1 FMC connector, providing a high-quality interconnect which can support up to 9.8304Gbps data rates for the JESD204B interface.

Schematic and layout files for the MAX5855 EV kit board are included with the software installation and can be found in the **MAX5855\EVKIT Info** folder

Xilinx VC707 FPGA Evaluation Board

The Xilinx VC707 board acts as the data source for the MAX5855, allowing for user defined signal generation. Test patterns, generated externally, are stored in the VC707's on-board DDR memory and subsequently transmitted through five lanes of JESD204B to the MAX5855. A total of 1GB of pattern(s) can be stored, allowing for the use of very long patterns, or multiple patterns consecutively. Multiple patterns allow the user to easily change patterns without repetitive upload commands. The USB2.0 (BULK) interface minimizes the time requirement for uploading the test patterns. Integrated commands allow the VC707 to properly drive all lane rate and speed combinations supported by the MAX5855.

Table 2. MAX5855 EV kit LED Description

LED	COLOR	DESCRIPTION
D1	Red	Normally Off; Temperature alert based on threshold setting in GUI
D5	Green	Normally On; Auxiliary 1.8V Power Indicator (U9 POK)
D2	Green	Normally On; DUT 1.0V Power Indicator (U12 POK)
D3	Green	Normally On; DUT 1.8V Power Indicator (U2 POK)
D4	Green	Normally On; Main MAX5855 EV kit 3.3V Power Indicator

The MAX5855 EV kit GUI software also provides a simple interface for controlling the VC707 board. Use the VC707 tab in the GUI to upload the firmware file which configures the on-board Virtex7 FPGA. The firmware design incorporates the MicroBlaze microcontroller function in the FPGA, which is used to manipulate the operation of the FPGA as well as pass-through commands for the MAX5855 EV kit. The supported set of MicroBlaze commands are listed in Appendix II for reference, however all required commands for normal operation are incorporated into specific controls in the GUI software.

When the VC707 board is first powered up, the INIT, DONE, and Supply LEDs will be solidly lit green while the GPIO LEDS will flash ON, cycling from 0 through 7 (see Figure 14a). Once the FPGA has been programmed using the .bit file as noted in the **Quick Start Procedure** for running the MAX5855 EV kit software, all of the LEDs will be solid-ON, including all eight of the GPIO LEDS (see Figure 14b). If the user has programmed the EEPROM on the VC707 with the **MAX5855EVKIT.mcs** file (see the *Programming the EEPROM* section), all of the LEDs will be ON except for GPIO LED number 6 when it first starts-up. As soon as the MAX5855 device is configured, the device will go on Mute until the system or user "unmutes" the output.

The GPIO LEDS can be used to identify various states of the FPGA-to-MAX5855 interface. Table 3 describes these states

All jumpers and switches on VC707 should be used in its default configuration for normal operation of the MAX5855 EV kit software. Occasionally jumpers may have been changed during use with other systems, so it is recommended the user confirm jumper J44 (near the USB 2.0 port) be connected 1-2 as in Figure 15a. Likewise, the user should confirm that Master BPI Programming switch bank, SW11 (near the FPGA and LCD display) be set to 00010 as in Figure 15b.

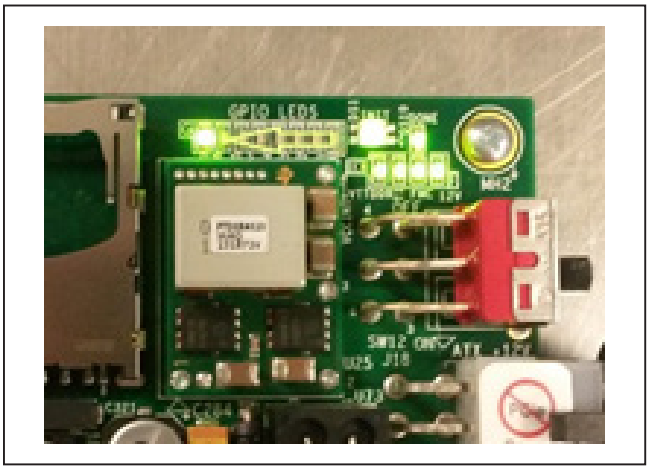


Figure 14a. VC707 LEDs—Before FPGA is Programmed

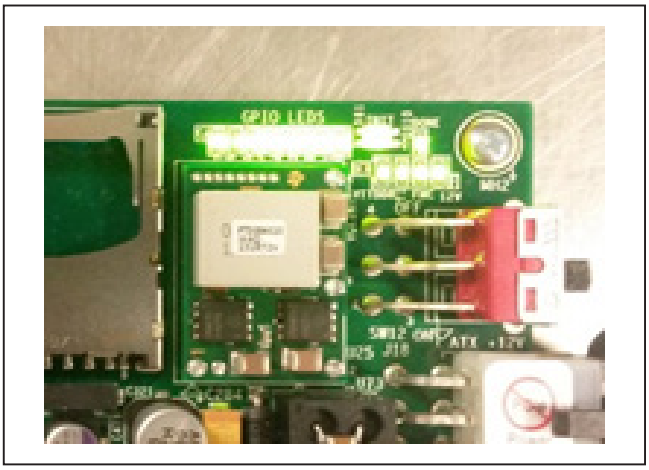


Figure 14b. VC707 LEDs 14b—After FPGA is Programmed

Table 3. VC707 LED Descriptions

LED	COLOR	STANDARD OPERATION	DESCRIPTION
0	Green	On	FPGA JESD PLL Locked
1	Green	On	DAC RX SYNCN Done
2	Green	On	FPGA MMCM Locked
3	Green	On	FPGA ResetN
4	Green	On	DAC Temp AlertB, active low
5	Green	On	DAC IntrB, active low
6	Green	Off	DAC Mute
7	Green	On	DAC ResetB, active low

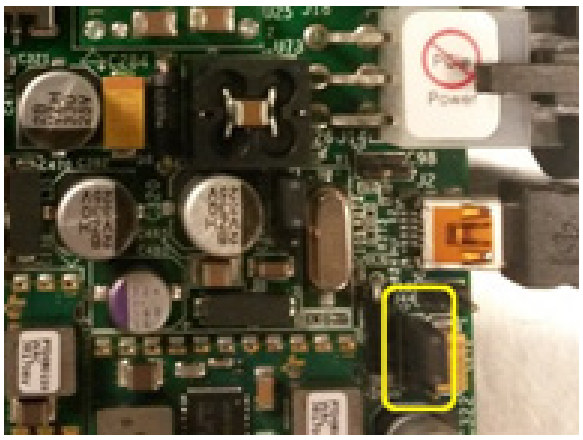


Figure 15a. VC707 Jumpers and Switches—J44;

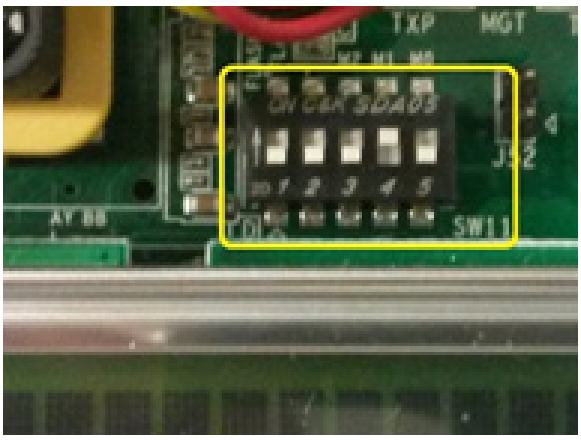


Figure 15b. VC707 Jumpers and Switches—SW11

Programming the EEPROM

Rather than programming the FPGA after each power cycle of the VC707, the on-board flash memory can be used to store the default RTL for the MAX5855 evaluation system. Once the EEPROM has been programmed, the USB cable connecting to the JTAG port (USB micro-B) will no longer be necessary.

For more information on programming the VC707 EEPROM, see the *VC707 FPGA Programming* section in the *Detailed Description of Software*.

Detailed Description of Software

The MAX5855 EV kit Software Controller GUI is designed to control the MAX5855 EV kit board and the VC707 board as shown in Figure 9. The software includes USB controls that provide SPI and SMBus communication to the MAX5855 and the MAX6654 interfaces. The GUI also controls the VC707 through the USB 2.0 control and bulk transfer port on the VC707 board.

The Communications Selection window (Figure 3) allows the user to select between the FPGA pass-through interface or the on-board FTDI interface for software control of the MAX5855 and the peripheral components. When selecting a communication path for the first time, the "Remember Selection, Do Not Show Again" check box will simplify the startup process by setting the selection as the default communication path.

If the user wishes to reset this selection, this can be done by deleting the file stored in the MAX5855 EV kit working directory. Browse to **C:\MaximIntegrated\MAX5855EVKIT** directory, find and delete the **FPGAPathSelection.txt** or **FTDIPathSelection.txt** file. This will cause the SPI Communication Path window to be displayed the next time the GUI is executed.

The MAX5855 **EV kit** Software Controller GUI features four window tabs for configuration and control of the MAX5855 and the VC707. The specific tabs are:

- Setup
  - Load and Reload MAX5855 Configuration
  - Hardware and Software MUTE Control
  - Various Reset Functions
- Clock and NCO
  - DAC Configuration Status
  - NCO Programming
- VC707
  - VC707 FGPA Programming
  - VC707 MicroBlaze Interface
  - VC707 Pattern Control
- Status
  - Temperature Readings and Control of the MAX6654 Temperature Sensor IC
  - Status of the MAX5855 EV kit Device Under Test
  - Automation Support Through TCP/IP Port
- Register Access
  - User Access to Read/Write MAX5855 Configuration and Status Registers

The *EV kit* GUI software begins on the <VC707> tab if the FPGA has not been programmed (Figure 4), otherwise the <Setup> tab will be active.

Logging Results

The MAX5855 EV kit GUI software automatically logs interactions between the GUI software, the MAX5855 EV kit board, the MAX5855 DAC, and the VC707 FPGA system. The Results Log block is displayed independently of the tab selection, so it remains visible within the window. Logging of most commands can be turned on or off by clicking on the <Logging> check box. The user can manually enter additional logging information into the text box and the whole log can be copied to the Windows clipboard or cleared by clicking on the <Copy> or <Clear> buttons respectively.

Setup Tab

The Setup tab, Figure 16, allows the user to load a MAX5855 device configuration file, provides basic operational controls, and has one-click “Quick Start” routines.

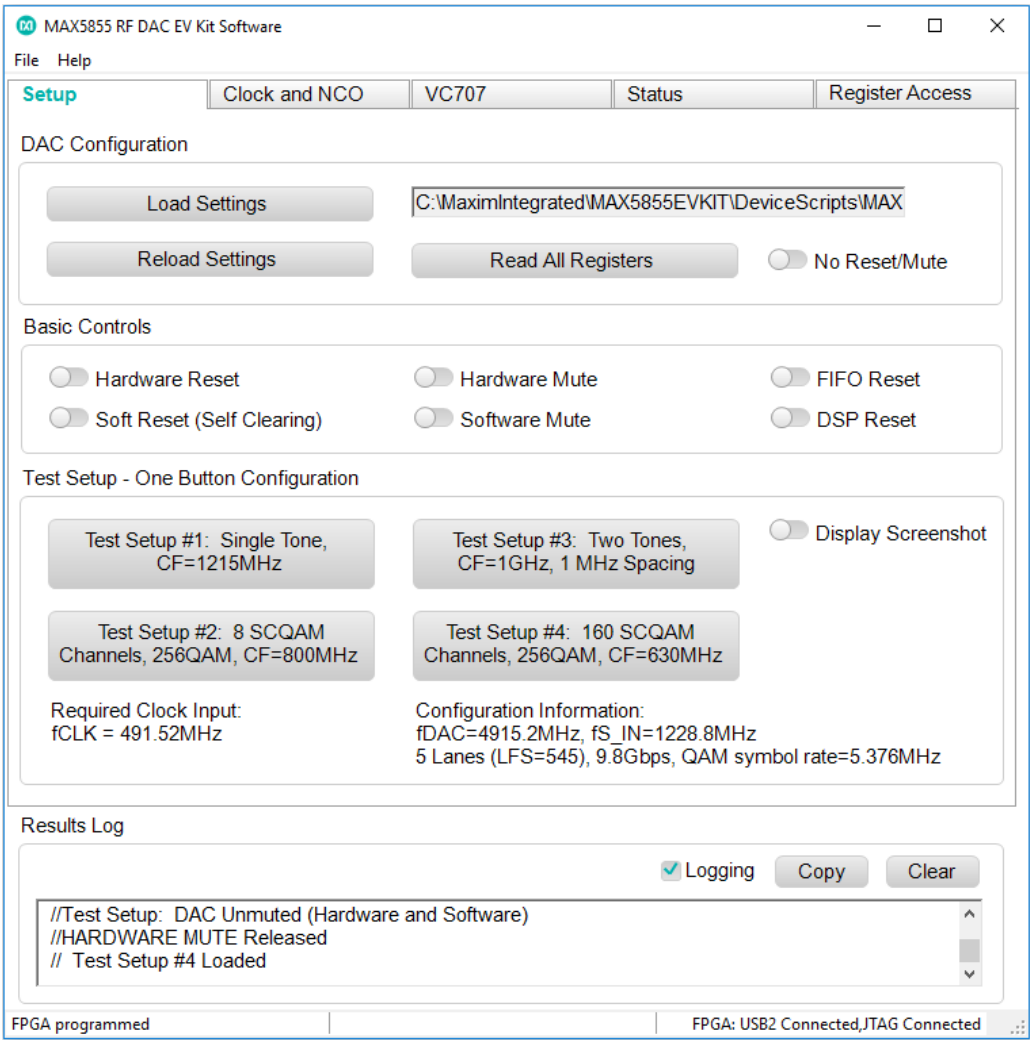


Figure 16. MAX5855 EV kit GUI—Setup

DAC Configuration

The DAC Configuration block allows for fast programming of the MAX5855 registers by using pre-sequenced, SPI register writes, consolidated into a text-based configuration file. Sample configuration files are included with the software installation and stored in the **C:\MaximIntegrated\MAX5855EVKIT\DeviceScripts** folder. Clicking the <Load Settings> button will cause a file selection window to open in the **\DeviceScripts** directory. The user then selects the **.cfg** file and clicks the <Open> button. This causes the software to assert, and then clear, a <Hardware Reset> prior to transferring the configuration (SPI writes) to the MAX5855. Once the configuration is complete, the <Hardware Mute> function is left ON to suppress the DAC output until the user is ready to observe the generated signal(s).

The configuration file contains a scripted and ordered set of commands that are sent to the MAX5855. These commands include register writes and "wait" directives. For additional information on the configuration file, reference the **Configuration Sequence** section of the MAX5855 Data Sheet.

Clicking on the <Reload Settings> button will cause the GUI software to use the same configuration file already selected in the text box and reload the MAX5855 registers with those initial configuration settings.

If the user wishes to load the configuration setting or reload the existing configuration without the software automatically resetting the device or muting the DAC output, select the "No Reset/Mute" switch. Asserting this switch will prevent any pre-cursor resets but will not prevent any soft reset commands that are within the configuration file itself.

To "dump" the MAX5855 registers to a file, click on the <Read All Registers> button. This will cycle through the DAC's internal registers and write them out to a file in numerical order. The **allregread.txt** file will be located in the main operating directory of the software.

Basic Controls

The Basic Controls block contains various switches including <Hardware Reset>, <Soft Reset>, and <FIFO Reset>. The user is NOT normally required to drive these controls.

Additional switches include <Hardware Mute>, a GPIO controlled pin on the DAC, is automatically asserted at startup. The <Software Mute>, a register-based control, is asserted every time a new configuration is loaded. The mute controls are used to protect downstream devices (PA) or equipment while the RF DAC is being configured and prior to the generation of valid test signals.

Test Setup – One Button Configuration

The quick-start <Test Setup …> buttons allow for a one-click configuration of the MAX5855, assuming the FPGA has been programmed. Four buttons are provided, each with a different default data pattern. Clicking on one of the Test Setup buttons will begin a sequence of commands both for the VC707 data source system and the MAX5855 RF DAC to properly configure the device through the SPI port and deliver data over the high-speed interface, resulting in a pre-defined output signal.

A basic example of the output signal can be displayed by clicking on the <Display Screenshot> switch. This will open another window which shows a Spectrum Analyzer screen shot that matches the "Quick-Start" pattern output selected with the <Test Setup …> buttons.

Clock and NCO Tab

The Clock and NCO Tab, Figure 17, displays the clock settings that have been loaded during configuration of the MAX5855. The configuration script also sets the NCO for a specific value, which is displayed in the Final fNCO text box.

Updating the NCO requires multiple steps. First, determine if the Extended NCO (fractional mode) is desired; if needed click the Extended NCO Enable toggle button. Select the NCO update mode from the drop-down list. Options are a) Immediate, b) Wait with Timeout, c) Increment/Decrement and d) Wait without Timeout. Please refer to the MAX5855 device data sheet for details on the various NCO operating modes. Next, enter the desired NCO frequency in the Target fNCO (MHz) text box. Click the Calculate Values button to determine the nearest possible programmed frequency for the selected NCO mode. The values written to the MAX5855 will be displayed in the CfgNCOF text box. The final step is to click the Apply Values button which writes the values to the MAX5855 registers and updates the Final fNCO text box.

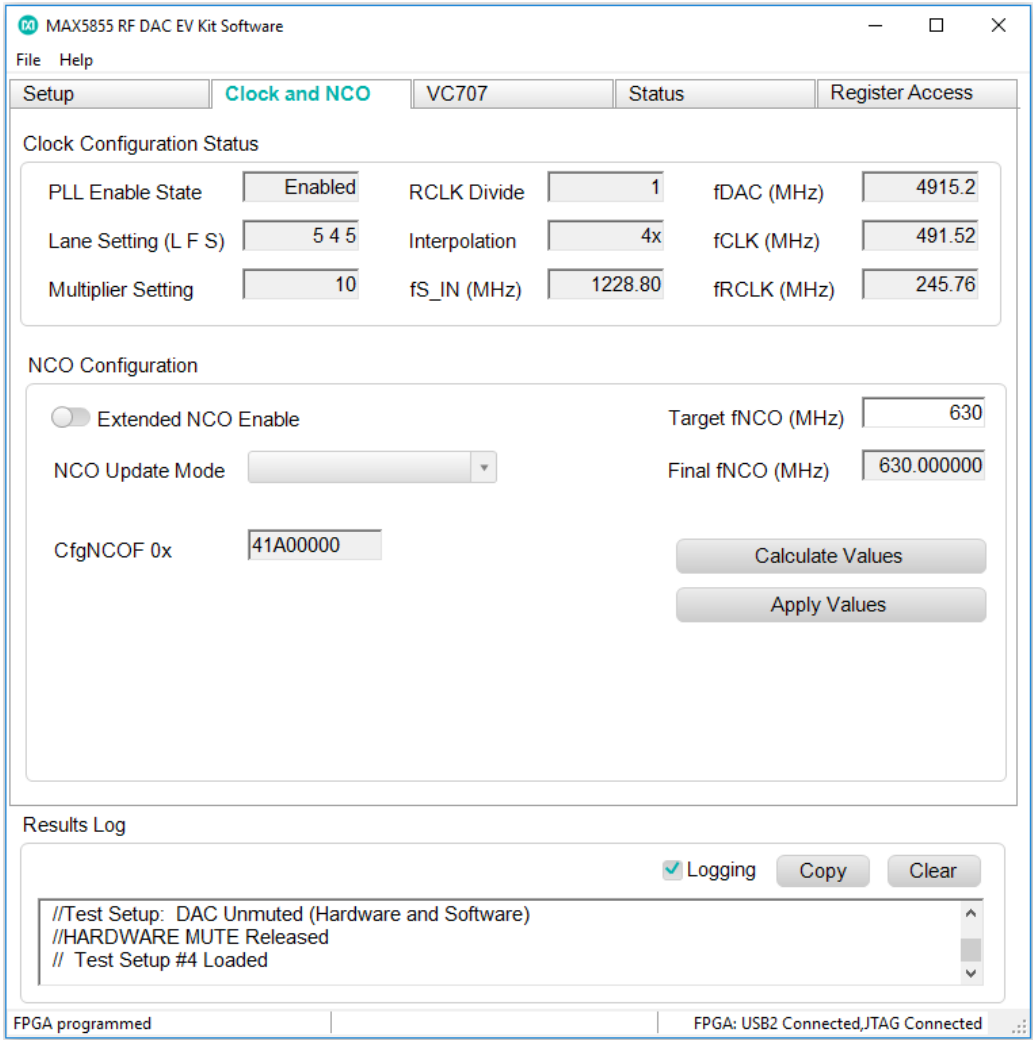


Figure 17. MAX5855 EV Kit GUI—Clock and NCO

VC707 Tab

The VC707 tab, Figure 18, allows the user to monitor the DAC configurations and interact with the VC707 FPGA system.

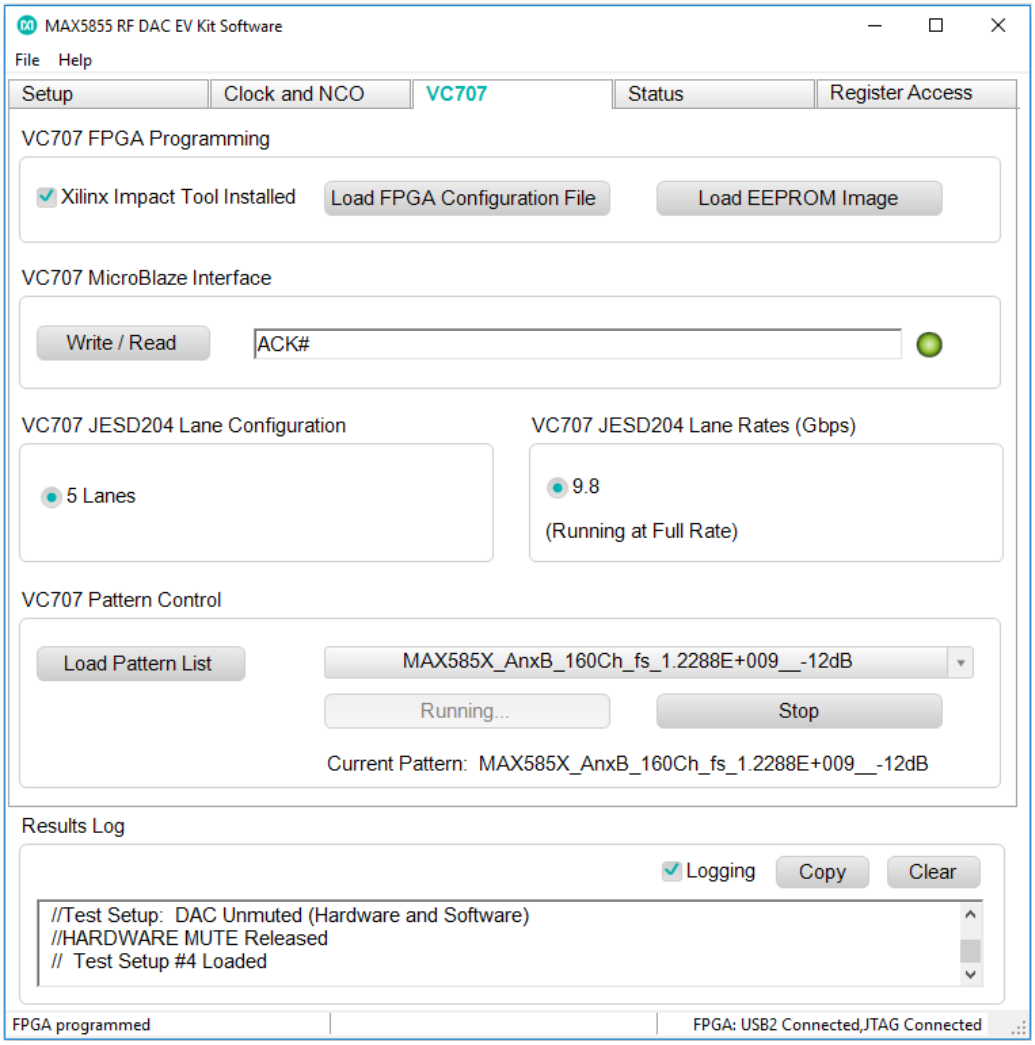


Figure 18. MAX5855 EV Kit GUI—VC707

VC707 FPGA Programming

The FPGA Programming block allows the user to interact with the Xilinx FPGA at a basic, bit-file level. As already discussed in the **Configure the MAX5855 EV kit Graphical User Interface** section, the VC707 is initially programed through this interface. The FPGA fabric provided with this evaluation system configures the FPGA to control and communicate with the MAX5855. The Virtex chip also acts as a data source system, loading pattern data into memory and transferring that data across the high-speed SerDes interface to the DAC.

By using the VC707's on-board flash, the user can permanently transfer the FPGA configuration file into memory allowing the FPGA to start in a pre-configured state, ready to interact with the MAX5855 EV kit.

To program the EEPROM:

1) Be sure the Xilinx Impact Tool box is checked
2) Click on the <Load EEPROM Image> button. This will open a browser window in the **C:\MaximIntegrated\MAX5855EVKIT\VC707Files** directory to load the **.mcs** file.
3) Select the **MAX585x_VC707Prog_xxJun2017.mcs** (or similar) file and click the <Open> button
4) If an "Exception" window opens (see Figure 19), simply click the <OK> button to continue
5) Loading the .mcs file into the EEPROM may take from 5 to 10 minutes

After the VC707 has been configured with the .mcs file, the MAX5855 EV kit GUI software no longer needs to use the Xilinx Impact Tools. Note the following improvements:

- Xilinx Impact Tools check box does not need to be clicked and the FPGA no longer needs to be loaded with a configuration file.
- The JTAG/USB connection is no longer needed. The USB Micro-B cable can be removed, requiring only one USB Mini-B cable (USB2 Control/Bulk port). for full communication and control of the MAX5855 evaluation system.

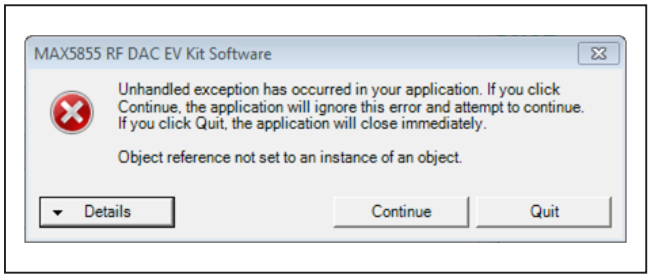


Figure 19. Exception Window

VC707 MicroBlaze Interface

The MicroBlaze Control block provides the user with a means to control and communicate with both the VC707 system and the MAX5855 using the passthrough SPI interface. The user is able to type commands into the text box, and can directly execute those commands by clicking on the <Write/Read> button. Any results that are returned, such as values or simply an "ACK#" response, will be displayed in the text box. The "LED" indicator to the right of the block will be green when an "ACK#" response is received and will turn red, when a "NAK#" response is received.

VC707 Pattern Control

The Pattern Control block allows the user to load different patterns by first opening a Pattern List file. The software utilizes these list files for loading test patterns into the VC707 memory, such as the example **TestLoadPatterns.txt** located in the **MAX5855\PatternFiles** folder. The list file simply contains an unordered list of names of test pattern files, including extensions. The format is simple ASCII text with one pattern file name on each line. Any line within the list that contains a '#' character will cause it to be skipped when the list is loaded by the GUI. The list can contain multiple patterns with up to 1MB in total pattern length, but only one pattern list can be loaded at a time; loading a new list will cause the previously loaded patterns to be overwritten. The MAX5855 EV kit software includes a single pattern list file: **TestLoadPatterns.txt**.

Pattern files contain the raw waveform data used by the RF DAC to generate an analog RF output. A collection of example patterns used to drive the MAX5855 RF DAC have been provided with the MAX5855 EV kit software. The user may have different types of patterns they wish to test with the evaluation system. To generate other pattern files, it is recommended the user have access to MathWorks MATLAB software.

For additional information on the Pattern File format used with the VC707 FPGA and the MAX5855 software, please refer to *Appendix III*.

Status Tab

The Status tab, Figure 20, allows the user to monitor DAC conditions such as temperature, internal lane states, and buffer status. It also provides the user with a means to control the GUI and the MAX5855 evaluation system remotely.

Temperature

The Temperature block displays the MAX5855 DAC temperature on request. To read the temperature from the MAX6654 via the SMBus serial interface, click on the <Read Temperature> button. If the user would like a visual indicator of an over-temperature fault (ALERT), enter the threshold temperature into the text box and click the <Set Threshold> button. This will cause the MAX6654 to set (active low) the ALERT output pin when the threshold temperature is exceeded. This will light the red, D1 LED on the MAX5855 EV kit board as a visual warning. This is a latched output so the alert needs to be cleared manually with the GUI software. To clear the temperature fault, click on the <Clear Temperature Alert> button.

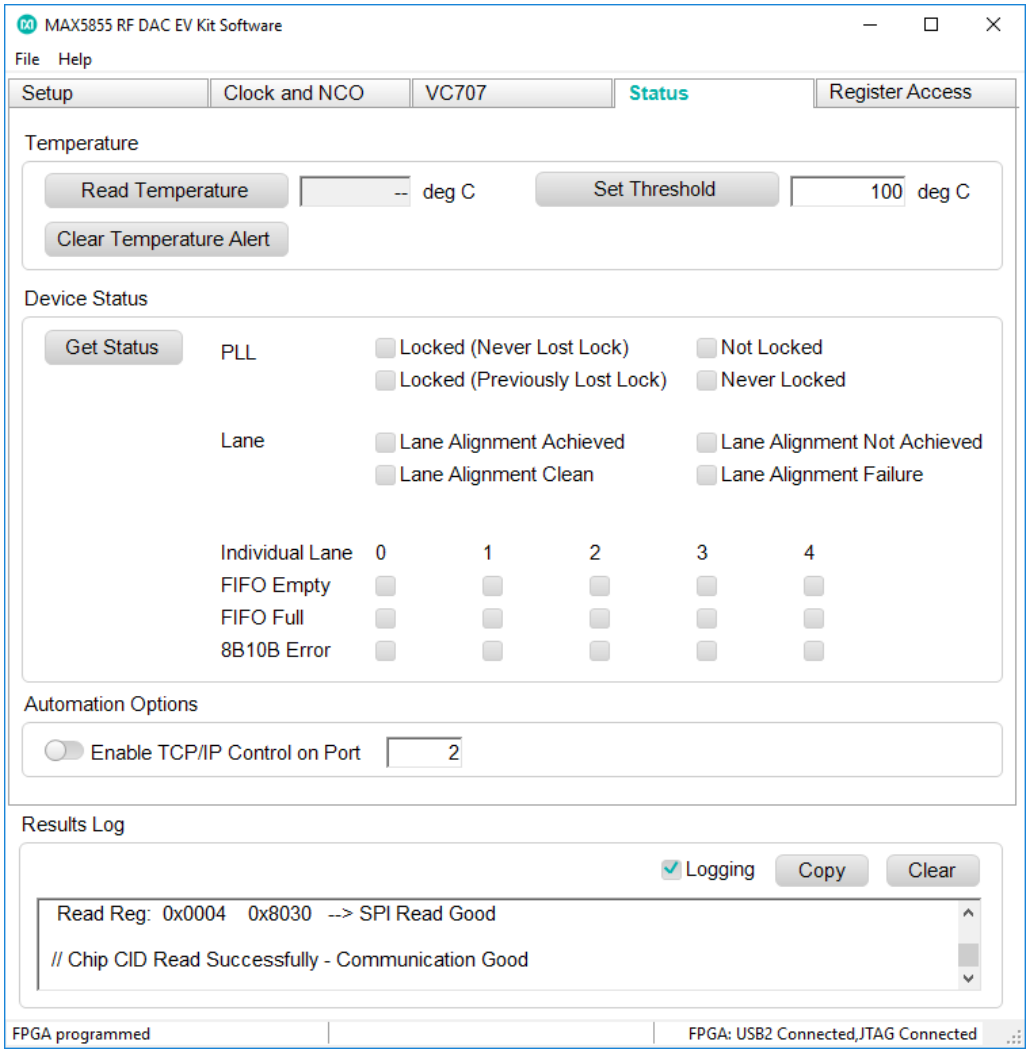


Figure 20. MAX5855 EV kit GUI—Status

Device Status

The Device Status block shows various operating states of the MAX5855. To update the device status, click on the <Get Status> button. The GUI software will read out the flag bits / registers for each of the five JESD lanes, including the FIFO overflow and underflow flags, the 8b/10b error indicator, and the lane alignment status bits. These are all displayed in the easy-to-read **Device Status** panel. If the user wishes to check the individual status bits, the *Register Access Tab* section describes how to interface directly with the MAX5855's internal registers.

Automation Options

The Automation block contains the "Enable TCP/IP Control" switch and a text box for entering a port number. This utility feature allows the user to run many of the GUI-based operations without the actual GUI. This TCP/IP command option turns the MAX5855 evaluation system into a valuable tool for automated evaluation or characterization of different DAC configurations typically used in automated bench testing.

See *Appendix II* for a list of supported TCP/IP, remote-control commands.

Register Access Tab

The Register Access tab, Figure 21, allows the user to interact directly with the MAX5855 through the SPI interface. The use of register read and write functions follows a simple, step-by-step procedure.

1) In the Select Device Register Blocks and then Select the Desired Register section
   a) Select a register block of interest by clicking on the "Block" drop-down and clicking on the block name
   b) Select the individual register of interest by clicking on the "Register" drop-down and clicking on the register name
2) In the Select Command section, choose a radio button to "Read" or "Write" to the previously selected register
3) In the Select Options section, if "Write" was selected then enter the intended value in this text box as a hexadecimal number ("0x" prefix is not necessary)
4) In the Execute Command section, after the desired "write" command has been assembled, the user simply clicks on the <Execute> button to write or read the selected register
5) In the Results section, the GUI will provide any feedback received from the MAX5855 through the FPGA interface. If only a "write" operation was executed, a read-back will be reflected in this text box. If a "read" operation was executed, then the value is displayed in this text box. Also note that the command that was written and the resultant read-back with be logged in the "Results Log" text in the lower portion of the GUI window.

To capture the state of all the internal registers ("dump the register map"), refer to the discussion of the <Read All Registers> button in the **Setup Tab** section above.

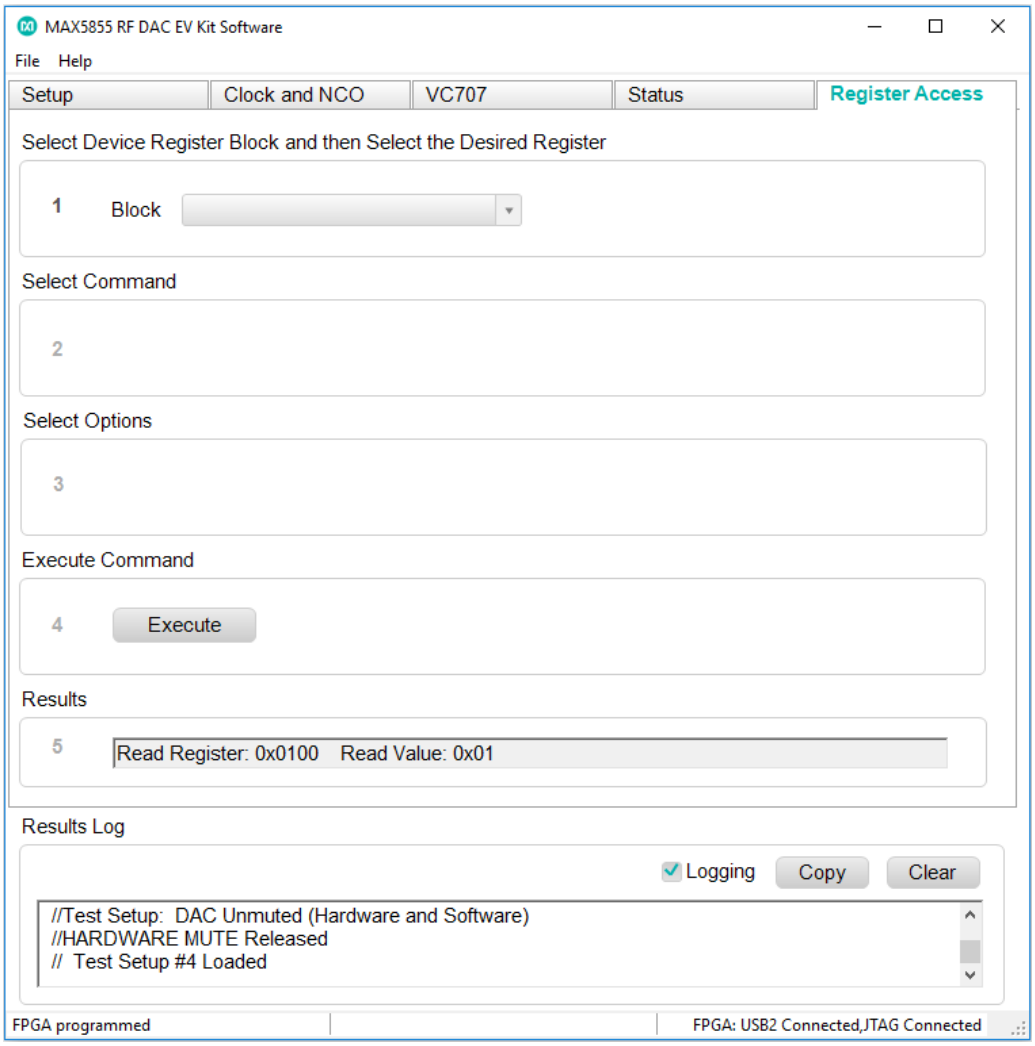


Figure 21. MAX5855 EV Kit GUI—Register Access

Component Suppliers

SUPPLIER	WEBSITE
Fairchild Semiconductor	www.fairchildsemi.com
Hong Kong Crystals Ltd.	www.hongkongcrystal.com
Murata Electronics North America, Inc.	www.murata-northamerica.com
Panasonic Corp.	www.panasonic.com
Taiyo Yuden	www.t-yuden.com
TDK Corp.	www.component.tdk.com

Note: *Indicate that you are using the MAX5855 when contacting these component suppliers.*

DAC Output Module Component List

PART	QTY	DESCRIPTION
C28, C31	2	CAPACITOR; SMT (0402); CERAMIC CHIP; 0.01UF; 16V; TOL=10%; MODEL=; TG=-55 DEGC TO +125 DEGC; TC=X7R
OUT	1	CONNECTOR; FEMALE; SMT; SMA JACK PCB; RIGHT ANGLE; 2PINS
T2	1	TRANSFORMER; SMT; 4.5-3000 MHZ; RF TRANSFORMER
PCB	1	PCB: MAXRFDACXFMROUTOPT1

Clock Input Module Component List

PART	QTY	DESCRIPTION
R1-R2	2	49.9 Ohm 1% resistor (0603)
T1-T3	3	TRANSFORMER; SMT; 4.5-3000 MHZ; RF TRANSFORMER
CLK	1	CONNECTOR; FEMALE; SMT; SMA JACK PCB; RIGHT ANGLE; 2PINS
PCB	1	PCB: EPCB_RXCM

Ordering Information

PART	TYPE
MAX5855EVKIT#	EV kit
EK-V7-V707-G	Xilinx Virtex 7 FPGA Board*

#Denotes RoHS compliant.

**Order from Xilinx or authorized distributor.*

MAX5855 EV Kit Bill of Materials

PART	QTY	DESCRIPTION
GND, +3.3V	2	CONNECTOR; MALE; PANELMOUNT; BANANA JACK; STRAIGHT; 1PIN
C1	1	CAPACITOR; SMT (CASE_D); ALUMINUM-ELECTROLYTIC; 150UF; 10V; TOL=20%; MODEL=FK SERIES
C3, C4, C15, C23, C29, C30, C38, C53, C62, C70, C73, C80, C173, C180	14	CAPACITOR; SMT (0402); CERAMIC CHIP; 1UF; 6.3V; TOL=10%; TG=-55 DEGC TO +85 DEGC; TC=X5R;
C5, C6, C18, C20-C22, C42, C57, C59, C60, C63, C65, C95, C97, C179	15	CAPACITOR; SMT (0402); CERAMIC CHIP; 100PF; 50V; TOL=5%; TG=-55 DEGC TO +125 DEGC; TC=C0G
C8, C13, C16, C17, C19, C26-C28, C35, C36, C39, C45, C46, C51, C54-C56, C58, C61, C64, C68, C69, C71, C74, C81, C94, C96, C174, C181	29	CAPACITOR; SMT (0402); CERAMIC CHIP; 0.01UF; 10V; TOL=10%; MODEL=GRM SERIES; TG=-55 DEGC TO +85 DEGC; TC=X5R
C9, C11, C14, C24, C37, C49, C50, C67, C72, C76-C79, C84, C116, C117, C172	17	CAPACITOR; SMT (0805); CERAMIC CHIP; 10UF; 6.3V; TOL=20%; TG=-55 DEGC TO +85 DEGC; TC=X5R
C10, C31, C47, C48	4	CAPACITOR; SMT (6032); TANTALUM CHIP; 47UF; 16V; TOL=20%; MODEL=TPS SERIES
C25	1	CAPACITOR; SMT (0402); CERAMIC CHIP; 430PF; 50V; TOL=5%; MODEL=; TG=-55 DEGC TO +125 DEGC; TC=C0G
C34, C40, C41, C43, C44, C86, C88, C90, C92, C1002-C1010	18	CAPACITOR; SMT (0402); CERAMIC CHIP; 0.1UF; 10V; TOL=10%; MODEL=GRM SERIES; TG=-55 DEGC TO +125 DEGC; TC=X7R; NOT RECOMMENDED FOR NEW DESIGN-USE 20-000u1-04A
C52	1	CAPACITOR; SMT (0402); CERAMIC CHIP; 2200PF; 50V; TOL=10%; TG=-55 DEGC TO +125 DEGC; TC=X7R
C1000, C1001	2	CAPACITOR; SMT (3528); TANTALUM CHIP; 4.7UF; 16V; TOL=20%
C1011, C1012	2	CAPACITOR; SMT (0402); CERAMIC CHIP; 8PF; 50V; TOL=+-0.25PF; MODEL=C0G; TG=-55 DEGC TO +125 DEGC; TC
C1013	1	CAPACITOR; SMT (0402); CERAMIC CHIP; 3.3UF; 6.3V; TOL=20%; MODEL=C SERIES; TG=-55 DEGC TO +85 DEGC; TC=X5R
D1	1	DIODE; LED; STANDARD; RED; SMT (0603); PIV=2V; IF=0.02A
D2-D5	4	DIODE; LED; WATER CLEAR GREEN; SMT (0603); VF=2.1V; IF=0.03A; -55 DEGC TO +85 DEGC
GND3, GND10, REF-GND	3	TEST POINT; PIN DIA=0.1IN; TOTAL LENGTH=0.3IN; BOARD HOLE=0.04IN; BLACK; PHOSPHOR BRONZE WIRE SILVER PLATE FINISH; RECOMMENDED FOR BOARD THICKNESS=0.062IN; NOT FOR COLD TEST

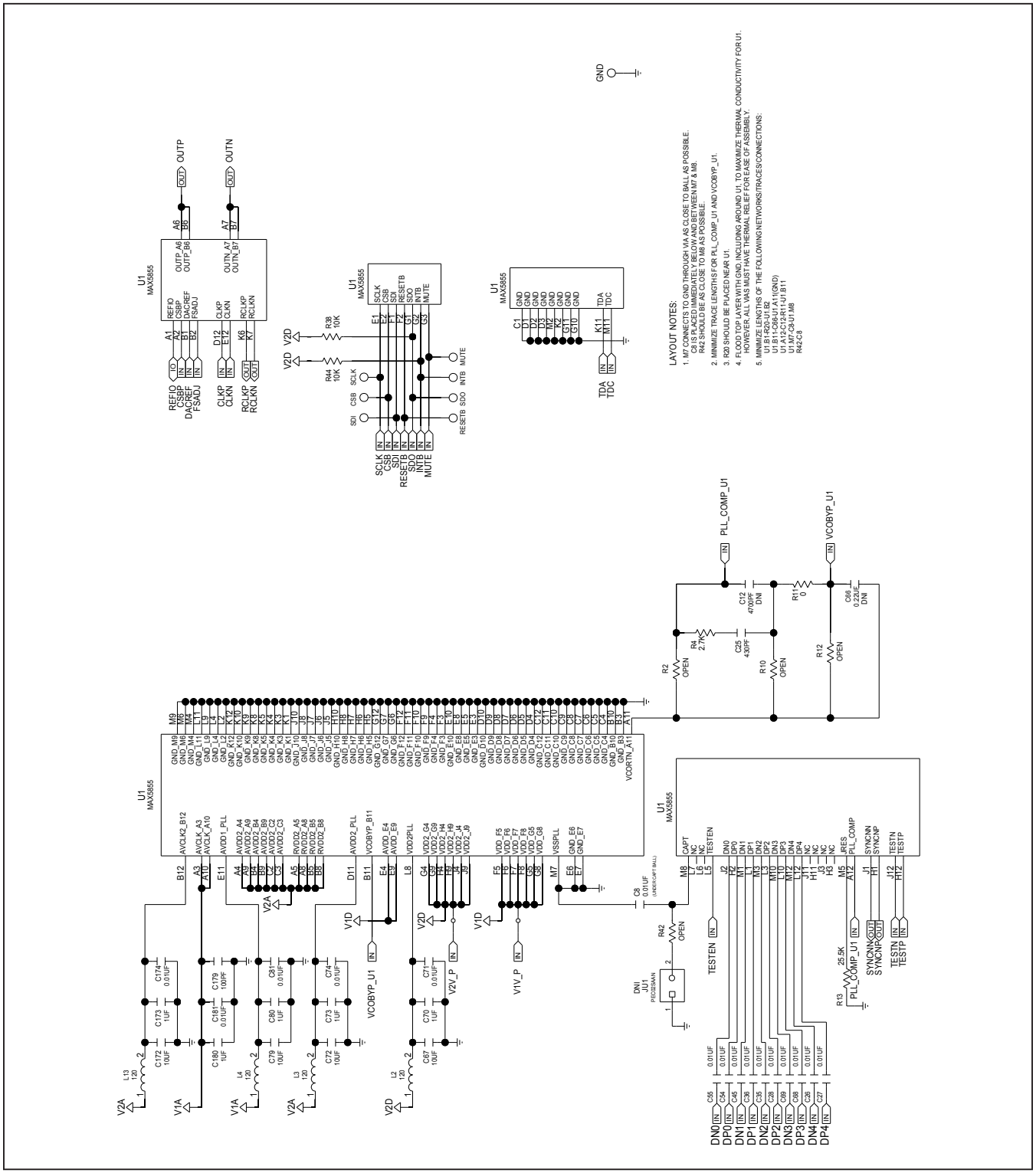
MAX5855 EV Kit Bill of Materials (continued)

PART	QTY	DESCRIPTION
V1A, V1D, V2A, V2D, GND6-GND9, VIN1, VIN2, GND11, GND12	12	EVK KIT PARTS; MAXIM PAD; WIRE; NATURAL; SOLID; WEICO WIRE; SOFT DRAWN BUS TYPE-S; 20AWG
J4	1	CONNECTOR; MALE; THROUGH HOLE; BREAKAWAY; STRAIGHT; 3PINS
J5	1	CONNECTOR; MALE; SMT; HIGH SPEED/HIGH DENSITY OPEN PIN FIELD TERMINAL ARRAY; STRAIGHT; 400PINS
J10	1	CONNECTOR; MALE; THROUGH HOLE; BREAKAWAY; STRAIGHT; 6PINS
JU4, JU5	2	CONNECTOR; MALE; THROUGH HOLE; BREAKAWAY; STRAIGHT; 2PINS
L1-L4, L10-L13	8	INDUCTOR; SMT (1812); FERRITE-BEAD; 120; TOL=+/-25%; 3A
L8, L9	2	INDUCTOR; SMT (1008); CERAMIC CHIP; 2.2UH; TOL=+/-5%; 0.28A; -40 DEGC TO +125 DEGC
L1000, L1001	2	INDUCTOR; SMT (0603); FERRITE-BEAD; 28; TOL=+/-25%; 4A
N1-N4	4	TRAN; ; NCH; SOT-23; PD-(0.33W); IC-(0.5A); VCEO-(60V); EVKIT PART FOR FORM FACTOR PROJECT
R1, R19, R25, R26, R28, R43	6	RESISTOR; 0603; 499 OHM; 1%; 100PPM; 0.10W; THICK FILM
R3, R5, R18, R21, R34-R36, R38, R41, R44, R1003-R1005	13	RESISTOR; 0603; 10K OHM; 5%; 200PPM; 0.10W; THICK FILM
R4	1	RESISTOR, 0402, 2.7K OHM, 1%, 100PPM, 0.0625W, THICK FILM
R6	1	RESISTOR; 0603; 4.22K; 1%; 100PPM; 0.10W; THICK FILM
R7, R9, R15	3	RESISTOR; 0603; 4.02K; 1%; 100PPM; 0.10W; THICK FILM
R8, R14	2	RESISTOR; 0603; 10.5K OHM; 1%; 100PPM; 0.063W; THICK FILM
R11	1	RESISTOR; 0402; 0 OHM; 0%; JUMPER; 0.10W; THICK FILM
R13	1	RESISTOR; 0603; 25.5K OHM; 1%; 100PPM; 0.10W; THICK FILM
R17, R33, R46, R56, R63	5	RESISTOR; 0603; 0 OHM; 5%; JUMPER; 0.10W; THICK FILM
R20	1	RESISTOR; 0603; 976 OHM; 1%; 100PPM; 0.10W; THICK FILM
R22, R23	2	RESISTOR; 0603; 4.7K; 1%; 100PPM; 0.10W; THICK FILM
R24	1	RESISTOR; 0603; 47 OHM; 5%; 200PPM; 0.10W; THICK FILM
R27, R29	2	RESISTOR; 0603; 100K; 1%; 100PPM; 0.10W; THICK FILM
R1001	1	RESISTOR; 0603; 1K OHM; 5%; 200PPM; 0.10W; THICK FILM
R1002	1	RESISTOR, 0603, 12K OHM, 1%, 100PPM, 0.10W, THICK FILM
R1006	1	RESISTOR; 0603; 2.2K OHM; 5%; 200PPM; 0.10W; THICK FILM
REFIO, USB3V3, TP_1V8AUX	3	TEST POINT; PIN DIA=0.1IN; TOTAL LENGTH=0.3IN; BOARD HOLE=0.04IN; RED; PHOSPHOR BRONZE WIRE SILVER PLATE FINISH; RECOMMENDED FOR BOARD THICKNESS=0.062IN; NOT FOR COLD TEST

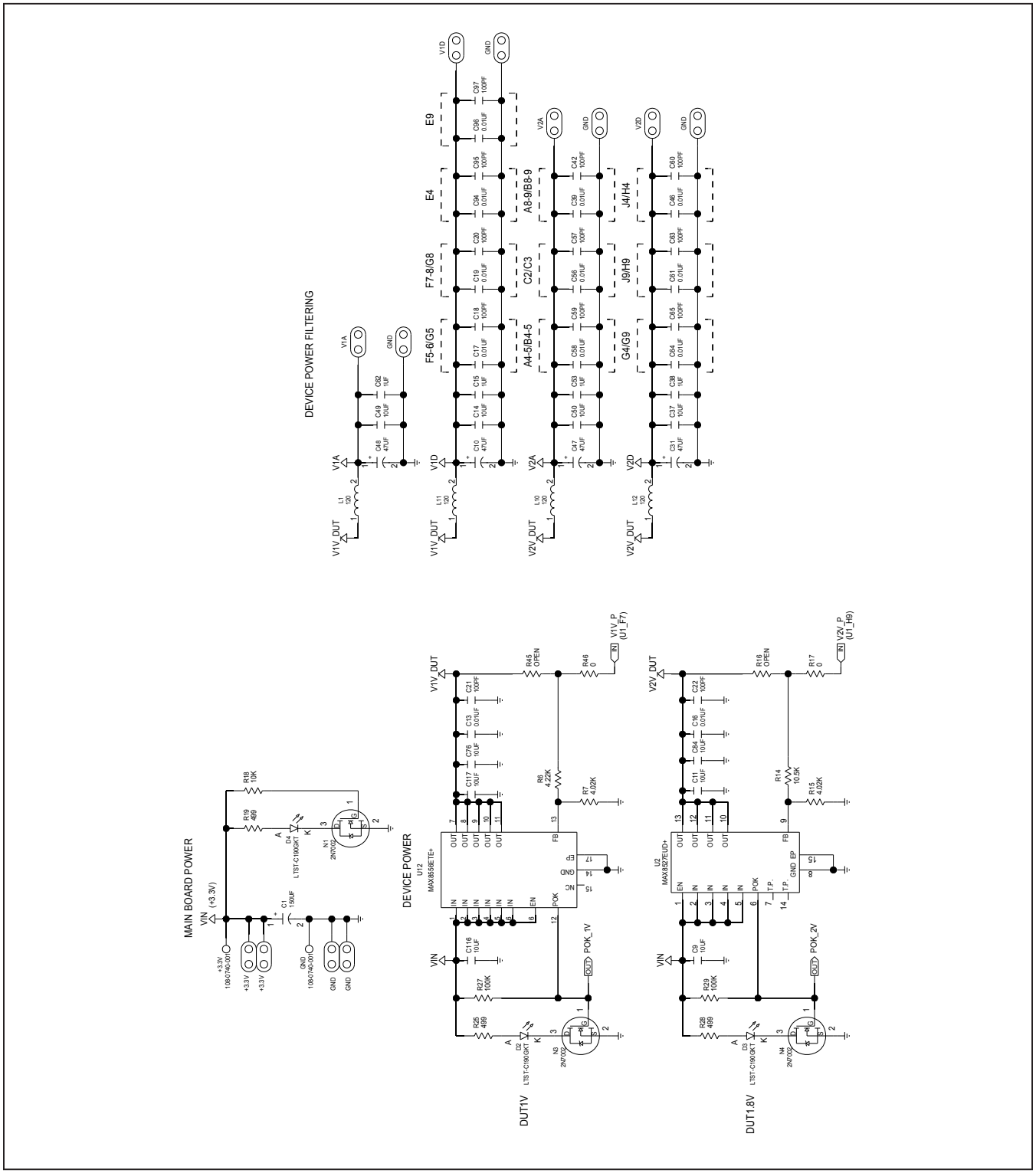
MAX5855 EV Kit Bill of Materials (continued)

PART	QTY	DESCRIPTION
SU2-SU6	5	TEST POINT; JUMPER; STR; TOTAL LENGTH=0.256IN; BLACK; INSULATION=PBT CONTACT=PHOSPHOR BRONZE; COPPER PLATED TIN OVERALL
SW1, SW2	2	SWITCH; SPST; SMT; 24V; 0.05A; NORMALLY OPEN-SURFACE MOUNT TACTILE SWITCH; RCOIL= OHM
U1	1	EVKIT PART- IC; BGA 10X10; 144 PINS; PKG. DWG. NO.: 21-0732; CD33
U2, U9	2	IC; VREG; 0.2V DROPOUT LDO REGULATOR; TSSOP14-EP
U3, U6, U14, U16	4	IC; TXRX; 4-BIT DUAL-SUPPLY BUS TRANSCEIVER WITH CONFIGURABLE VOLTAGE TRANSLATION AND 3-STATE OUTPUT; TSSOP16
U4	1	IC; VREF; LOW-COST; MICROPOWER; PRECISION; 3-TERMINA; 1.2V VOLTAGE-REFERENCE; SOT23-3
U5	1	IC; SNSR; ACCURATE TEMPERATURE SENSOR WITH SMBUS SERIAL INTERFACE; QSOP16
U7	1	IC; VSUP; LOW-POWER TRIPLE-VOLTAGE uP SUPERVISORY CIRCUIT; SC70-5
U10	1	IC; VREG; ULTRA-LOW-NOISE, HIGH PSRR, LOW-DROPOUT, LINEAR REGULATOR; SC70-5 ; -40 DEGC TO +85 DEGC
U12	1	IC; VREG; ULTRA-LOW-INPUT-VOLTAGE LDO REGULATOR; TQFN16-EP
U17, U18	2	IC; ASW; HIGH-BANDWIDTH; QUAD DPDT SWITCH; TQFN36-EP
U1000	1	IC; USB; QUAD HIGH SPEED USB TO MULTIPURPOSE UART/MPSSE IC; LQFP64 12X12
U1001	1	IC; EEPROM; 2K; 16-BIT MICROWIRE COMPATIBLE SERIAL EEPROM; NSOIC8 150MIL
USB	1	CONNECTOR; FEMALE; SMT; USB MINI B-TYPE SMT CONNECTOR WITH DOWEL PINS; RIGHT ANGLE; 9PINS
Y1000	1	CRYSTAL; SMT NO DATA; 18PF; 12MHZ; +/-30PPM; +/-50PPM
PCB	1	PCB:MAX5855
MODULE1	1	EVKIT PART-MODULE; MAXXFMROUT_OPT1#; RFDAC_XFMROUT_OPT1_EVKIT_A; PACKOUT PART
MODULE2	1	EV KIT MODULE, XFMR_CLK_MODULE, PACKOUT PART

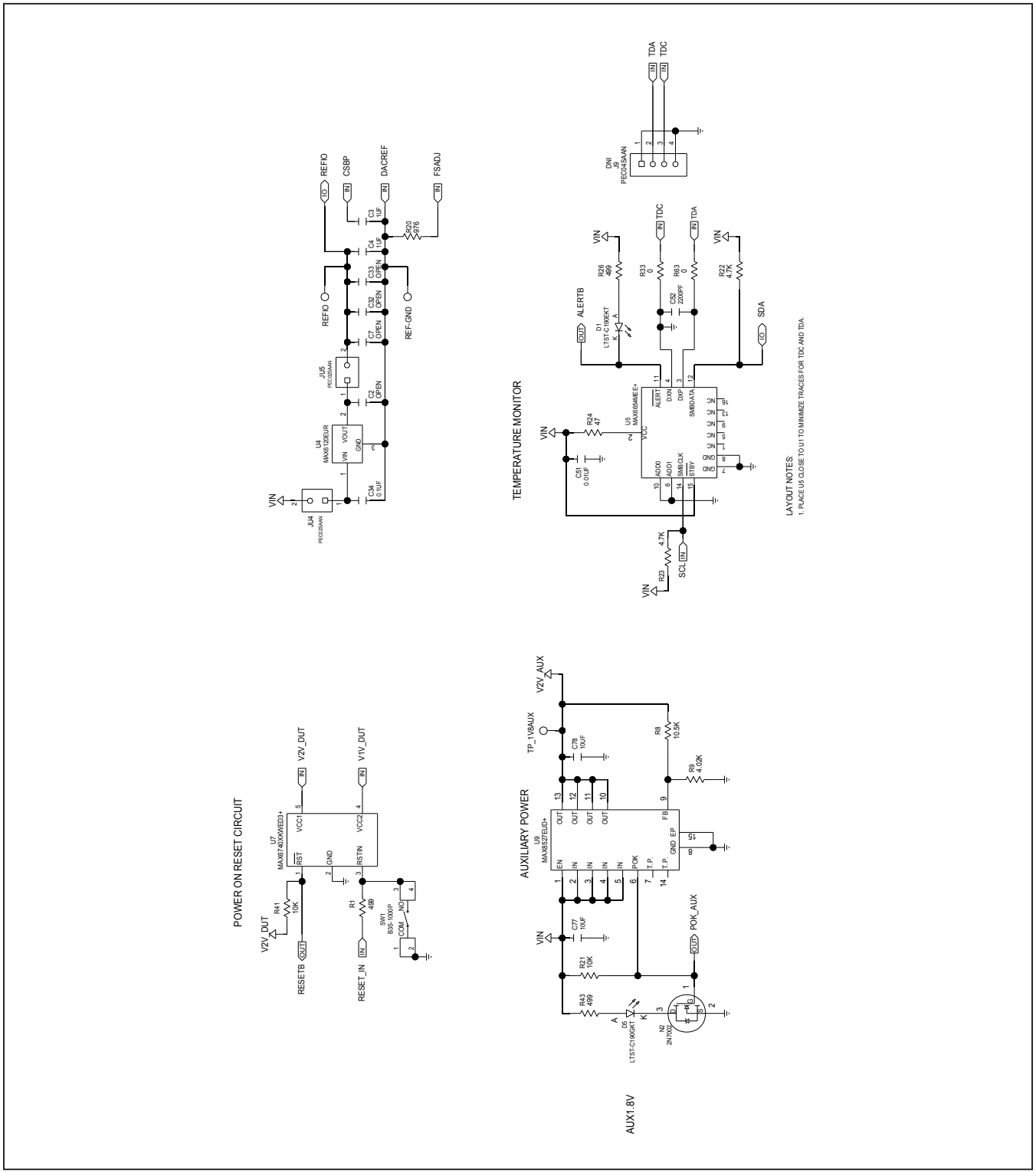
MAX5855 EV Kit Schematic



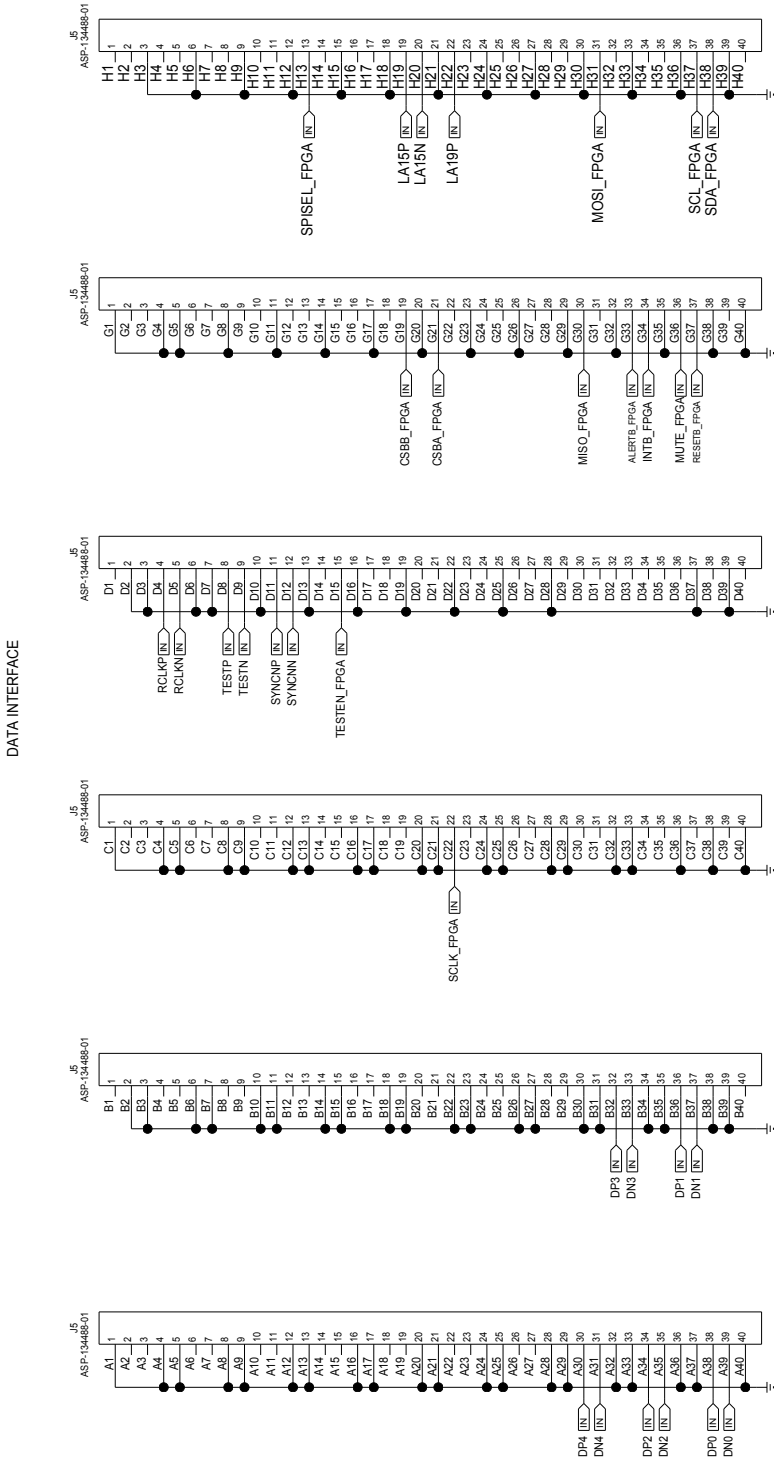
MAX5855 EV Kit Schematic (continued)



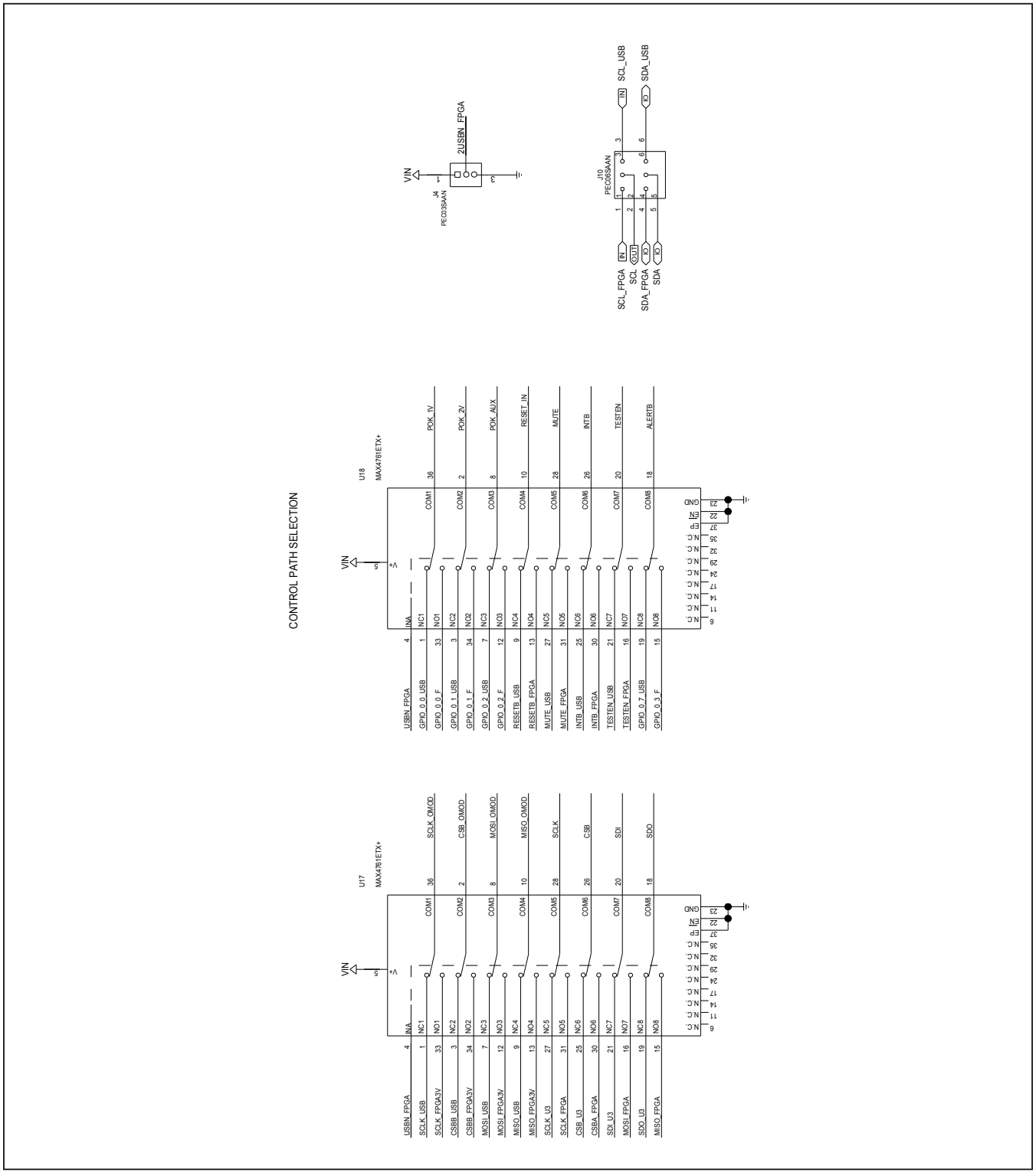
MAX5855 EV Kit Schematic (continued)



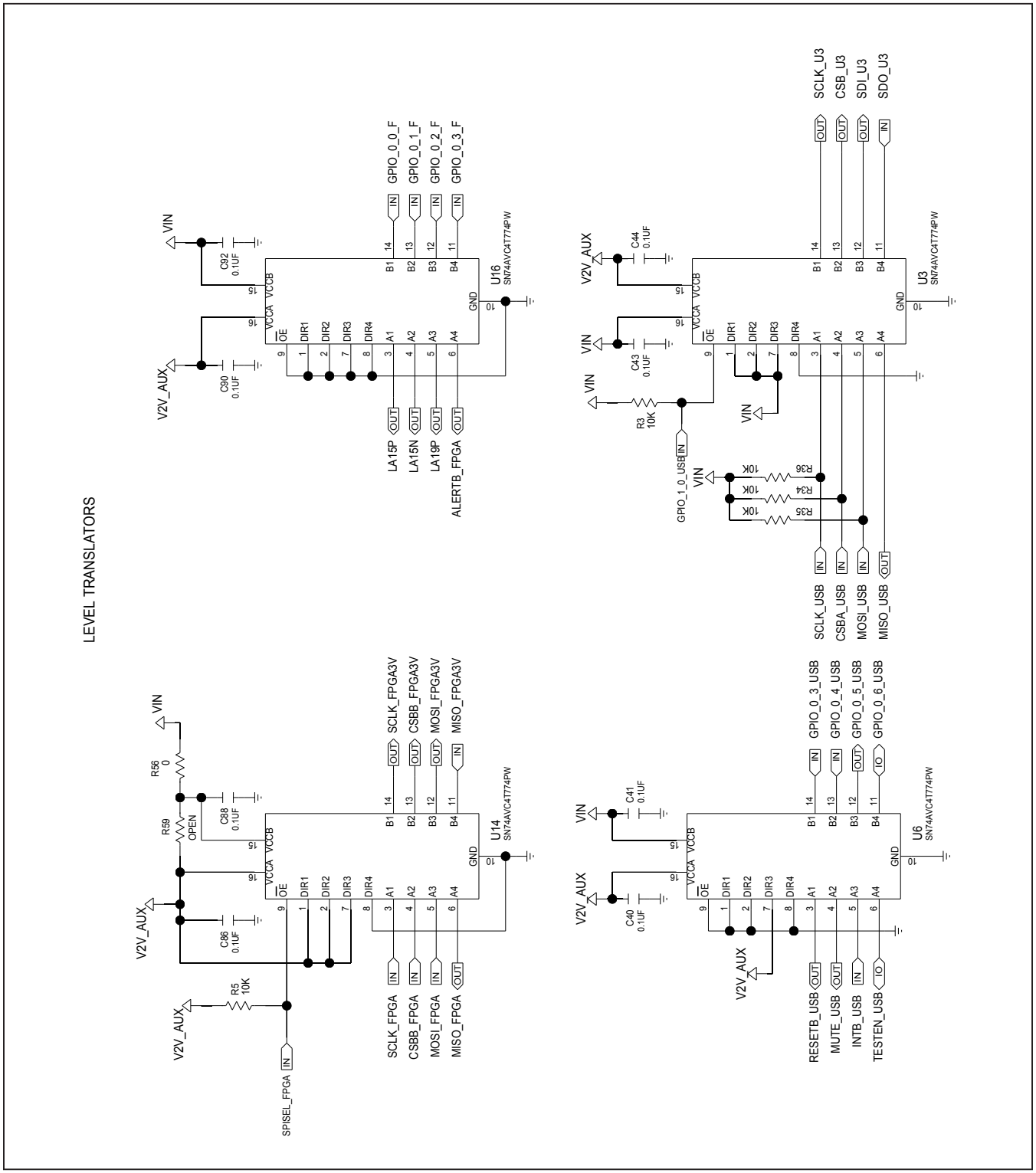
MAX5855 EV Kit Schematic (continued)



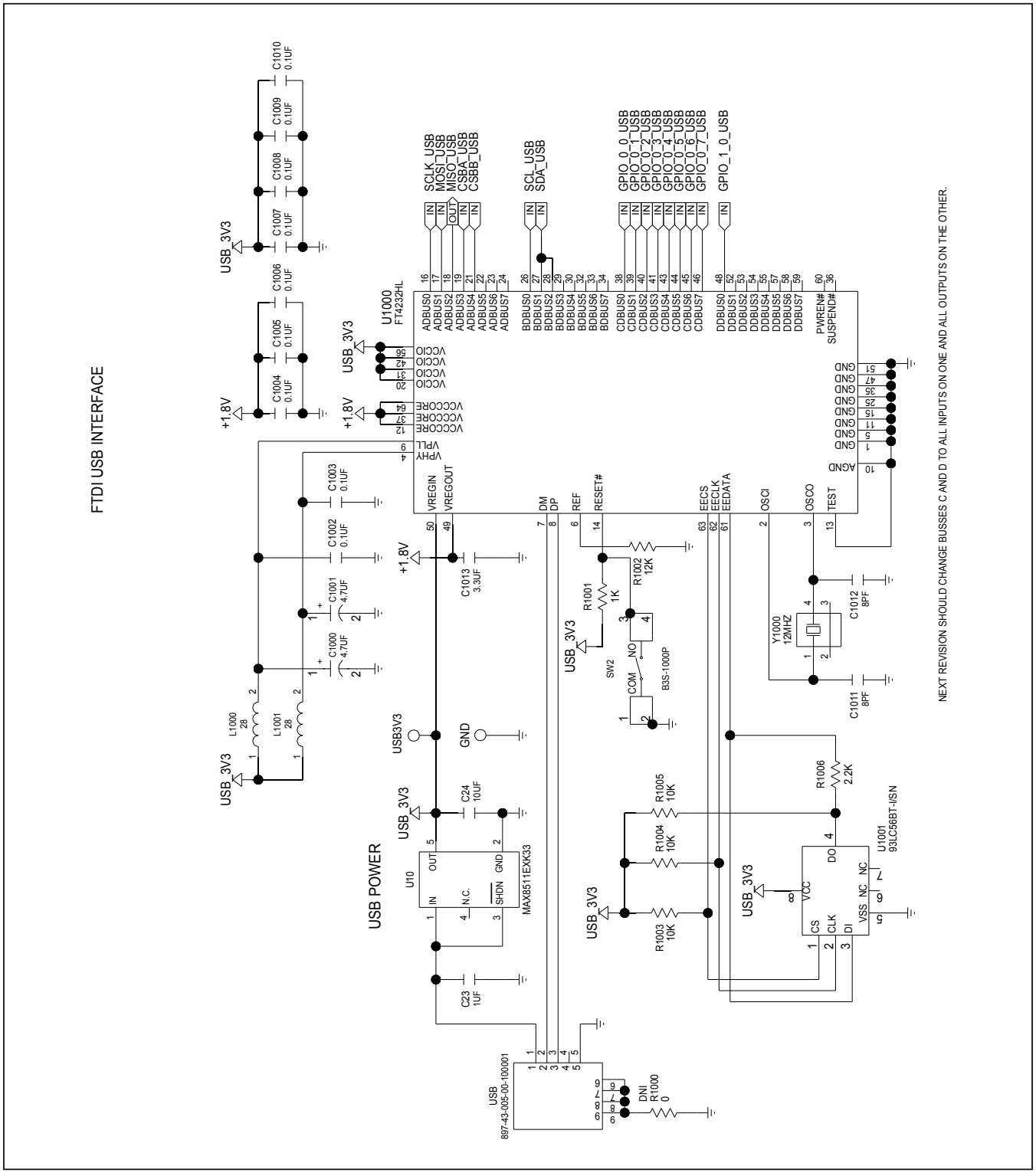
MAX5855 EV Kit Schematic (continued)



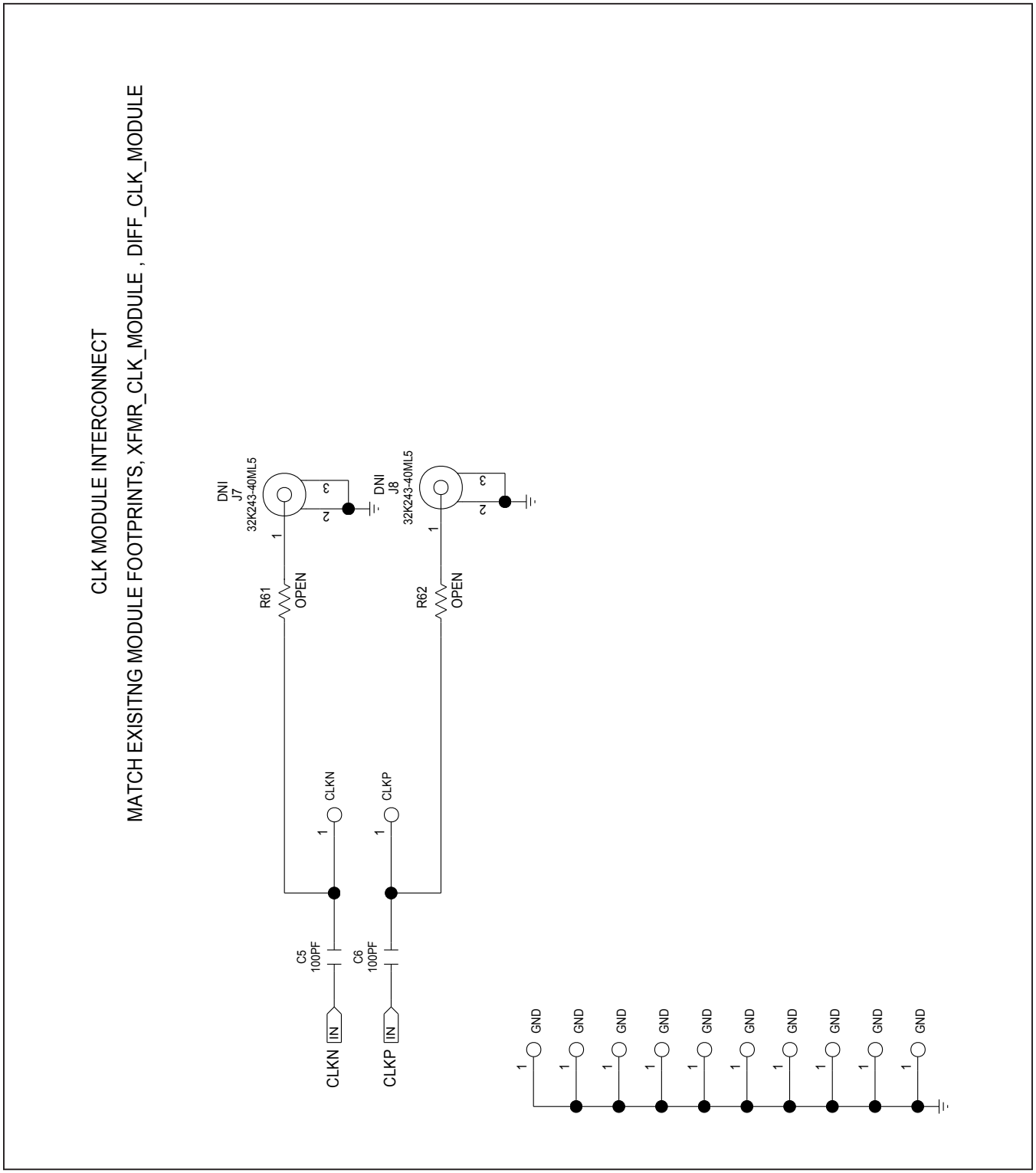
MAX5855 EV Kit Schematic (continued)



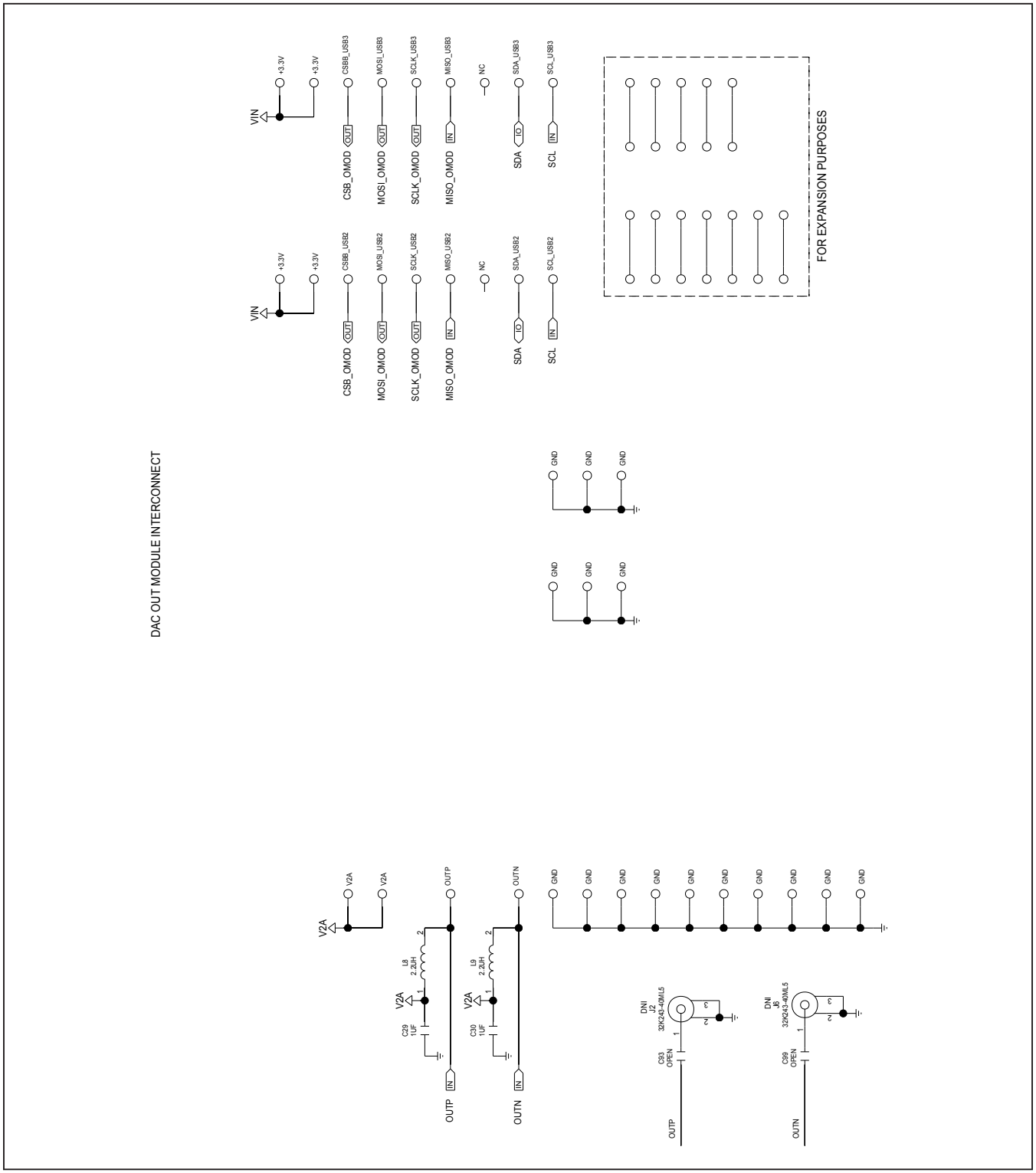
MAX5855 EV Kit Schematic (continued)



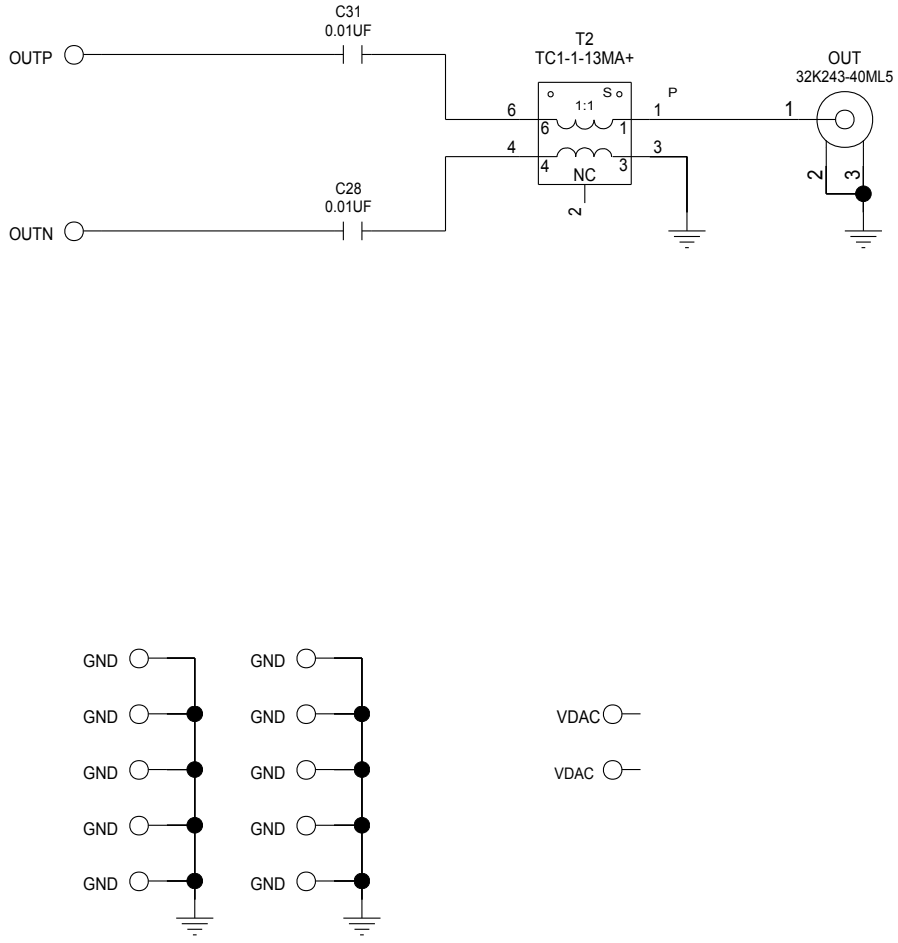
MAX5855 EV Kit Schematic (continued)



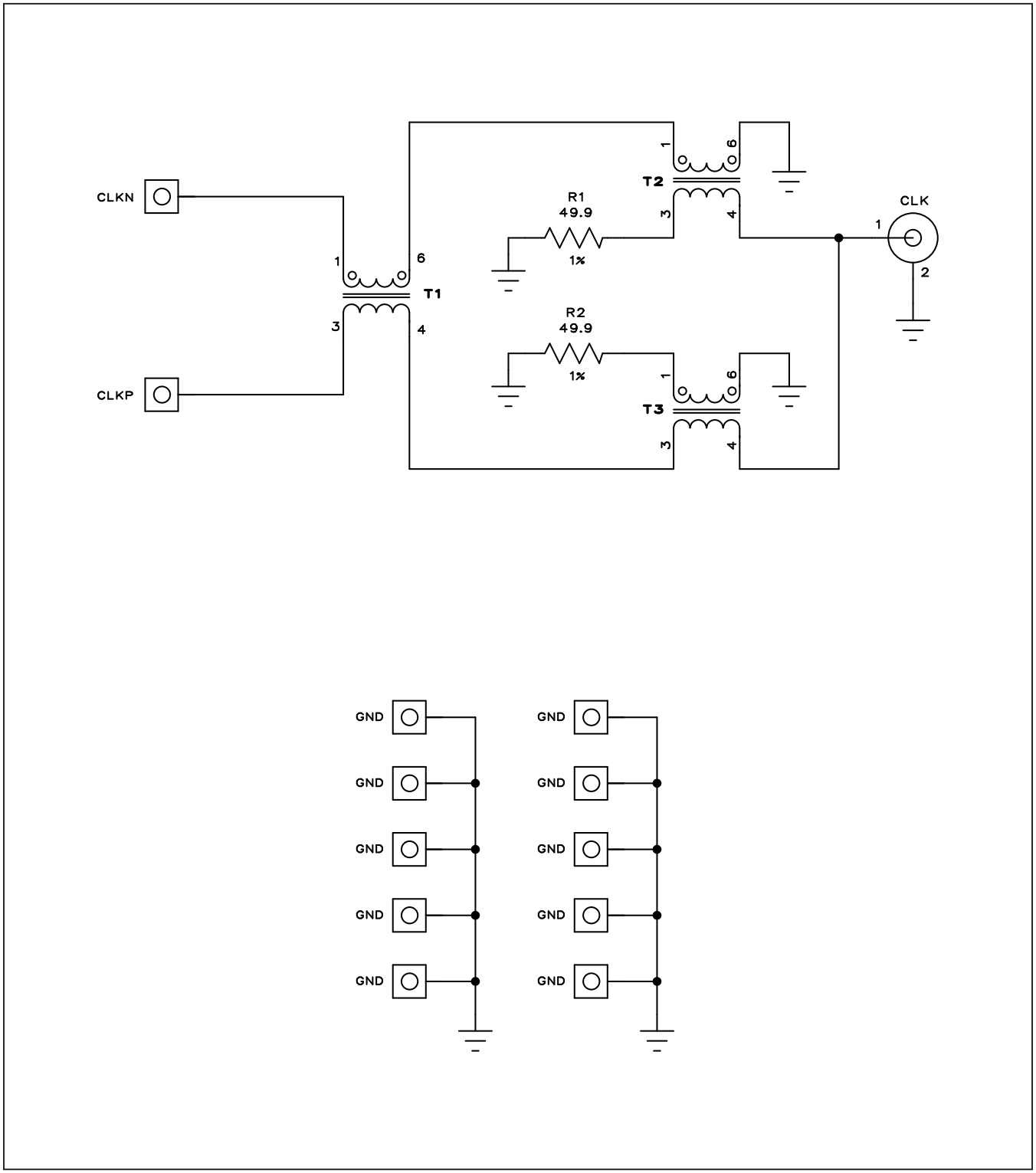
MAX5855 EV Kit Schematic (continued)



XFMROUT Module Schematic



XFMRCLK Module Schematic



Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	10/18	Initial release	—

For pricing, delivery, and ordering information, please visit Maxim Integrated's online storefront at https://www.maximintegrated.com/en/storefront/storefront.html.

Maxim Integrated cannot assume responsibility for use of any circuitry other than circuitry entirely embodied in a Maxim Integrated product. No circuit patent licenses are implied. Maxim Integrated reserves the right to change the circuitry and specifications without notice at any time. The parametric values (min and max limits) shown in the Electrical Characteristics table are guaranteed. Other parametric values quoted in this data sheet are provided for guidance.

Maxim Integrated and the Maxim Integrated logo are trademarks of Maxim Integrated Products, Inc. © 2018 Maxim Integrated Products, Inc.

Appendix I – Software and Driver References

Third-Party Software and Driver Installation

For the MAX5855 EV kit software to fully operate with the Xilinx VC707 FPGA platform, software and drivers need to be installed on the target PC. It is highly recommended that the PC be connected to a local area network and have access to the Internet, this will allow for automatic download and updates of some drivers. This process may take 30 minutes or more to complete.

Xilinx ISE 14.7 LabTools Installation

This is a free tool set used for programming the VC707 Evaluation Board, no software registration or license is required. Xilinx ISE LabTools can be downloaded directly from the web at **www.xilinx.com**; Type “Lab Tools 14.7 Download” in the site search bar. Open the download folder from the search result, scroll down page and download file “Windows 7/ XP/Server and Linux”. This tool is compatible for both Windows 7 and Windows 10 OS.

Note: *You may need to register or sign on to an account and verify company information to download files from Xilinx.*

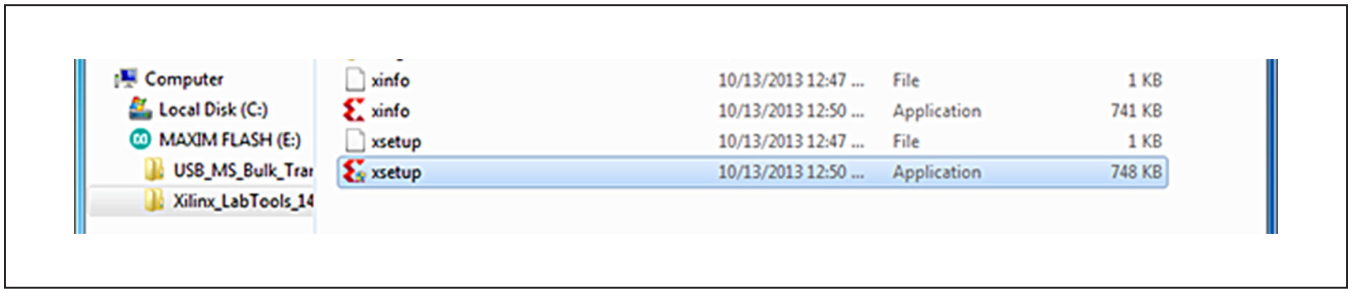


Figure A1-1. Xilinx ISE LabTools Installation

1) Double-click on the xsetup.exe program to start installation of the Xilinx ISE LabTools software.
2) Windows may prompt to allow the fallowing program to make changes, click to “Allow Changes.”
3) The ISE Design Suite splash screen will appear, then the Welcome Page, click Next > to continue.

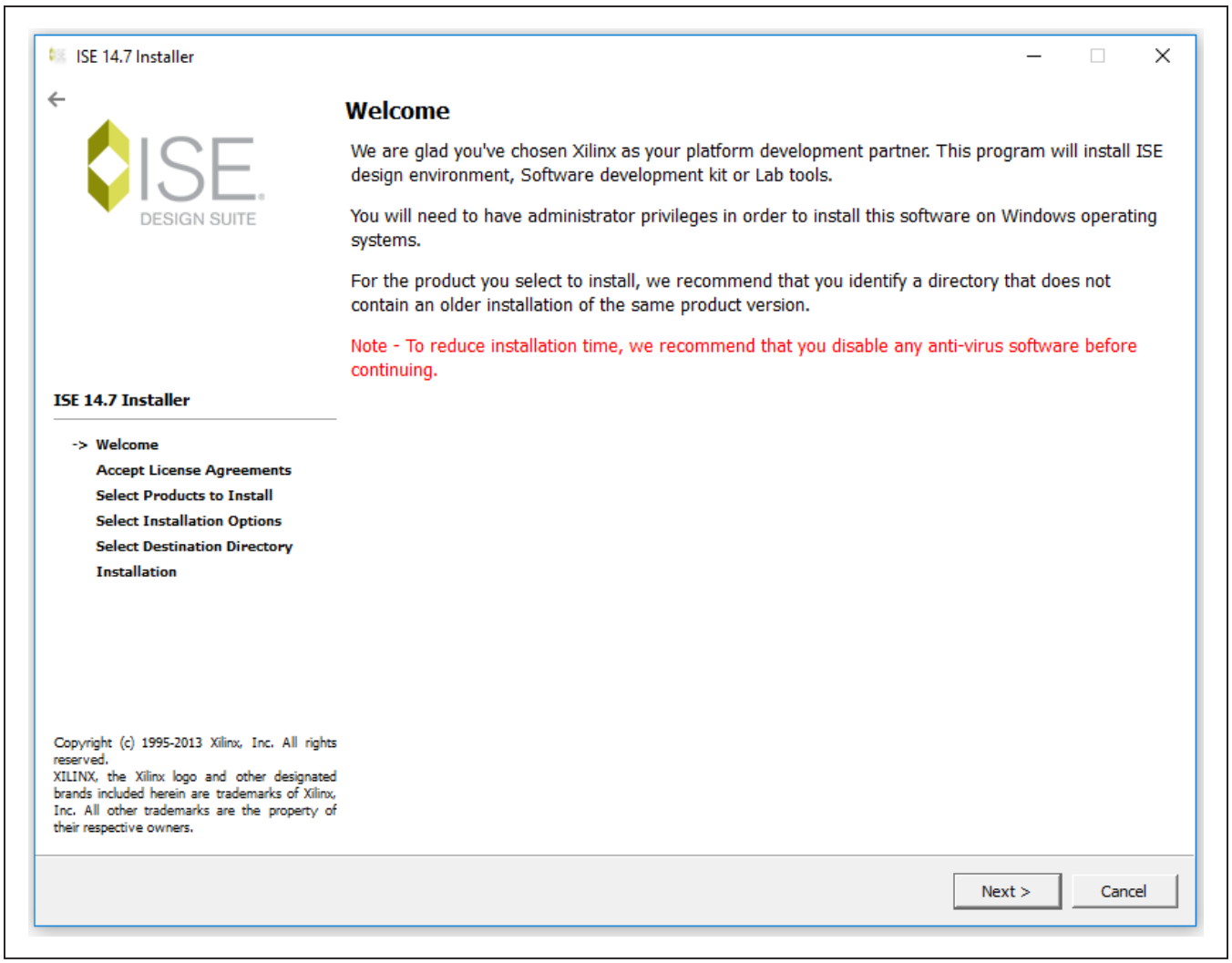


Figure A1-2. Various LabTools Installation Screens

4) Two “Accept License Agreement” pages appear, check the boxes and click Next > to continue.

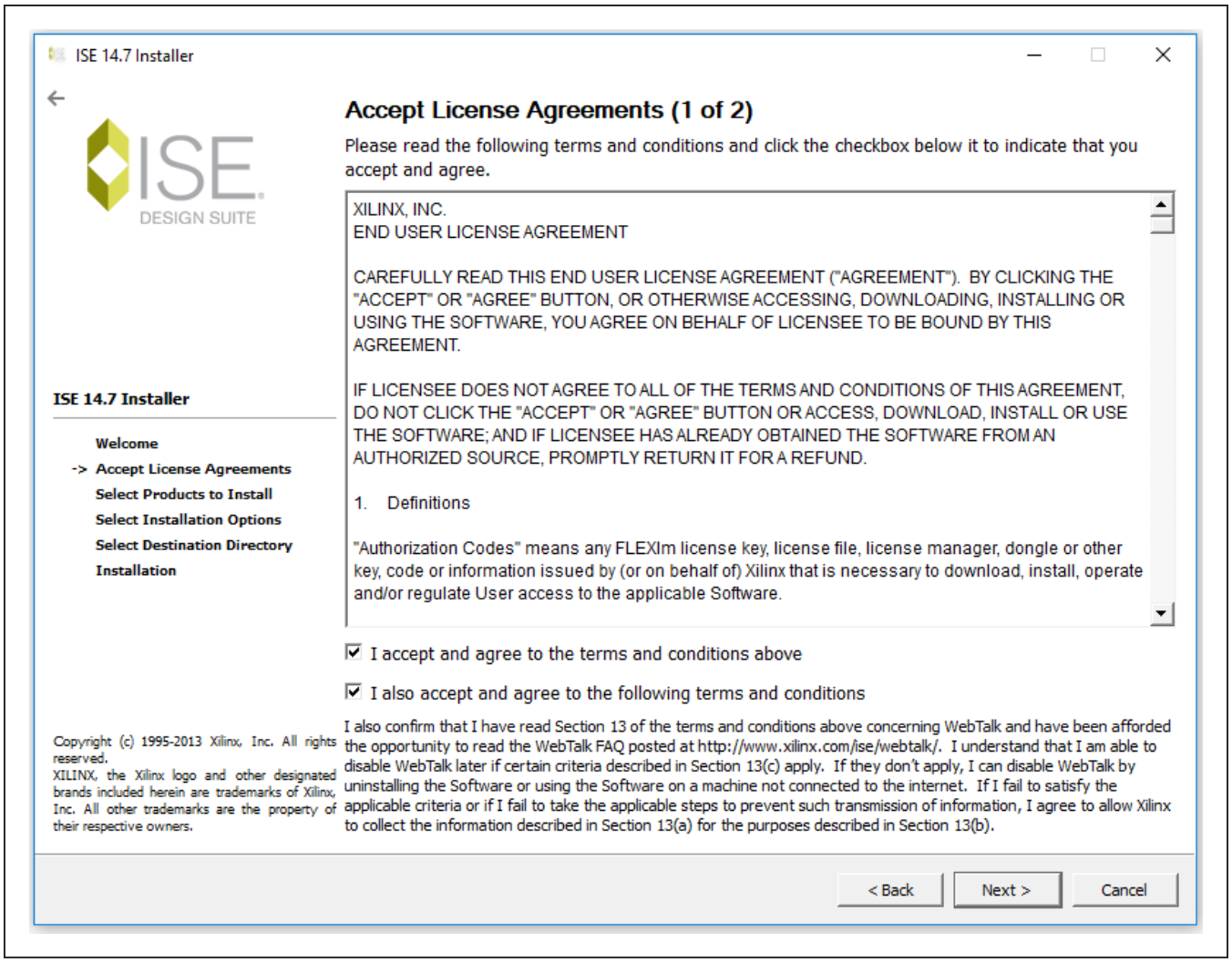


Figure A1-2. Various LabTools Installation Screens (continued)

5) The “Select Products to Install” page appears, the “Lab Tools – Standalone Installation” should be selected, click Next > to continue.

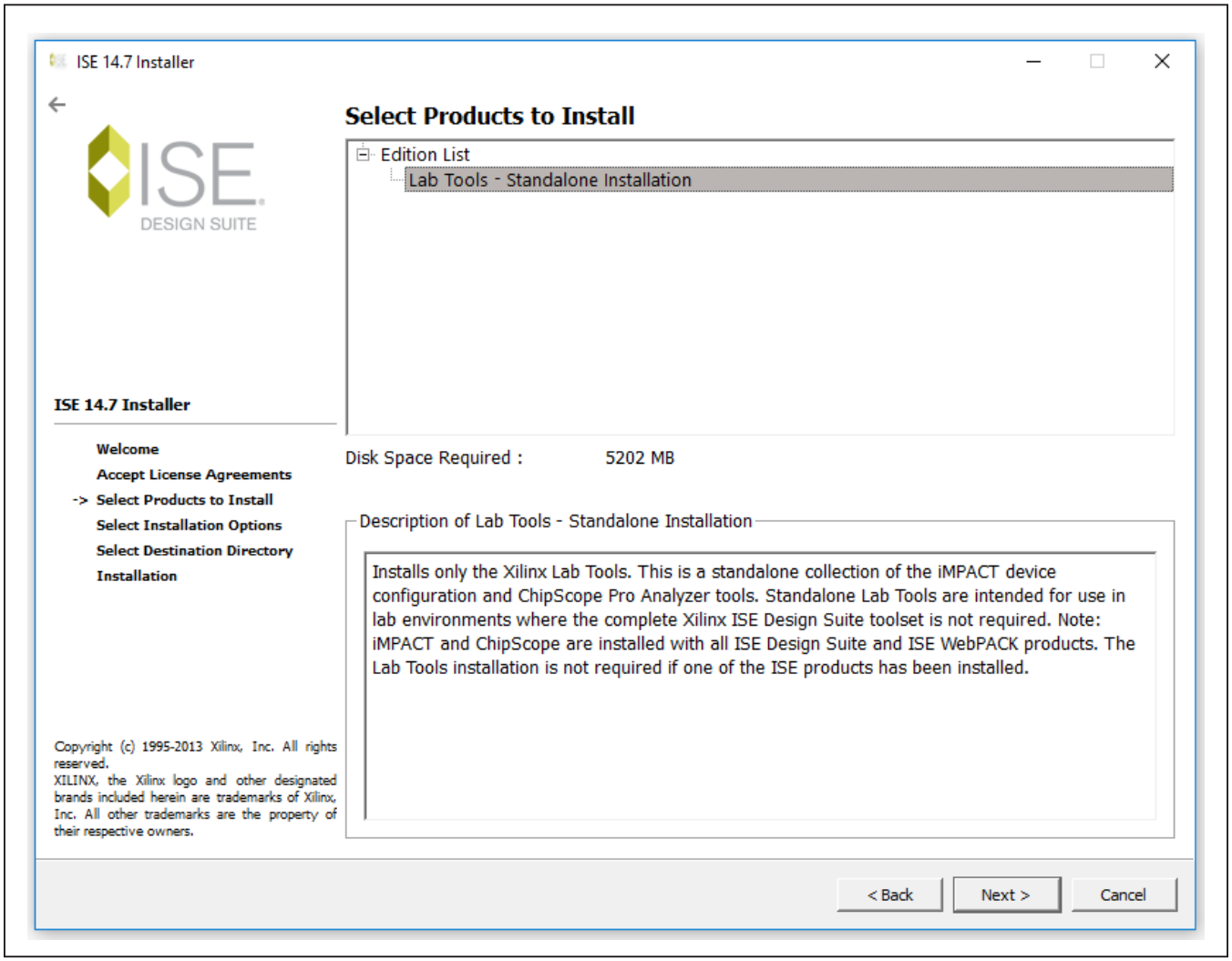


Figure A1-2. Various LabTools Installation Screens (continued)

6) The “Select Installation Options” page will appear (Figure A1-3), **uncheck** the “Acquire or Manage a License Key” option and click Next > to continue.

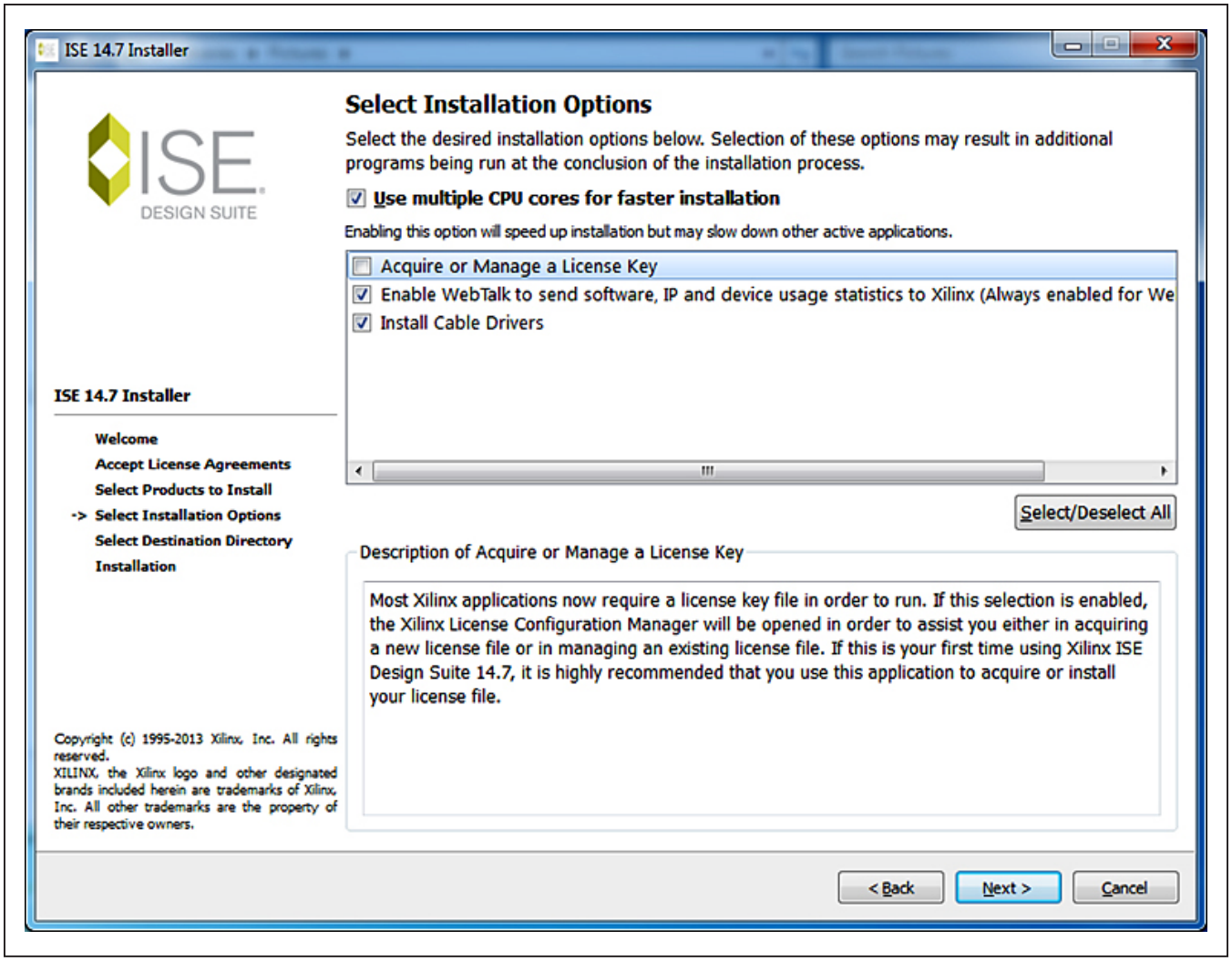


Figure A1-3. LabTools Installation—Uncheck “Acquire or Manage License Keys”

7) The “Select Destination Directory” page will appear. *It is highly recommended the default directory be used for the Xilinx LabTools*. Click Next > to continue.
8) The “Installation” summary page will appear, click on the Install button to continue.

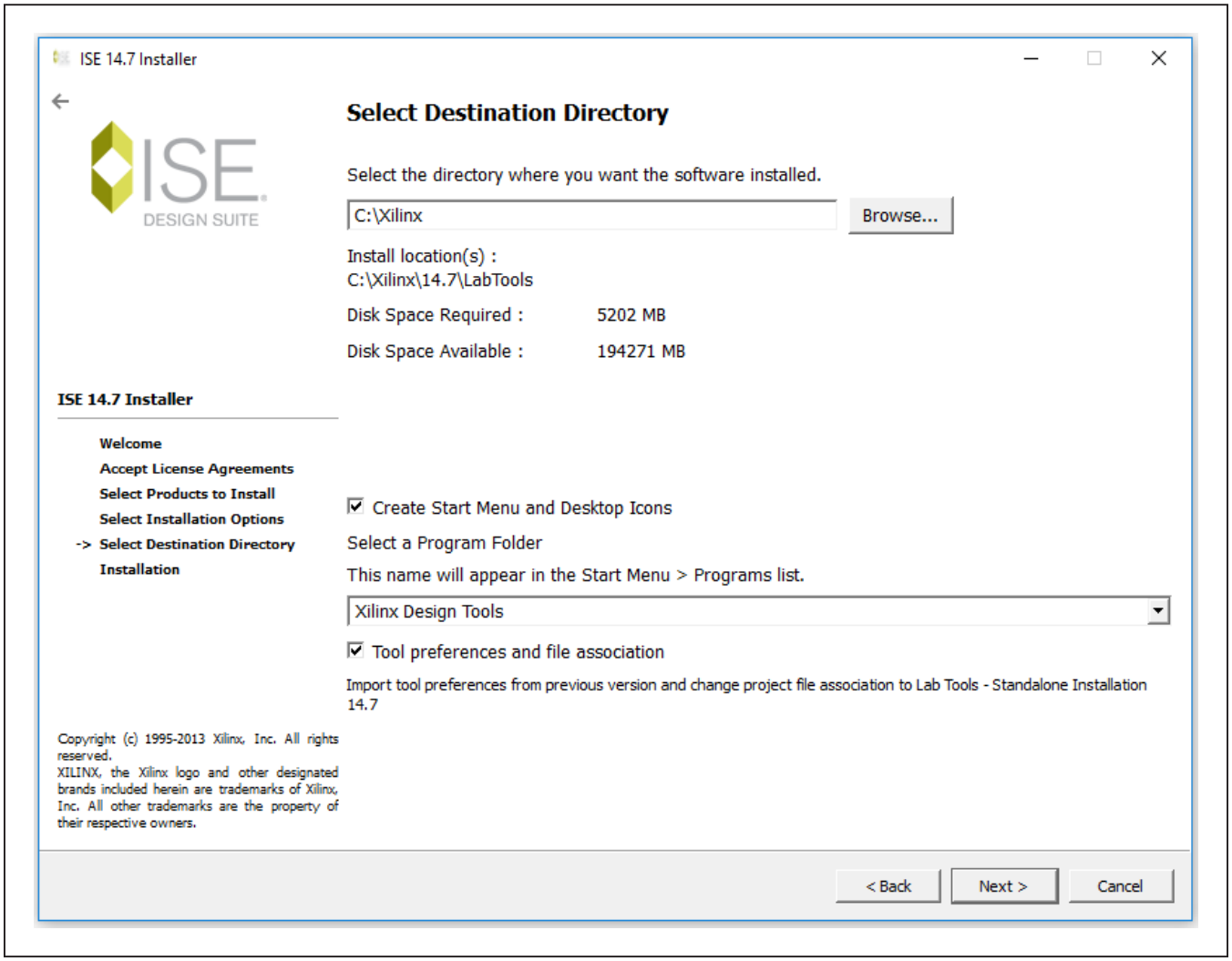


Figure A1-4. Various LabTools Installation Screens

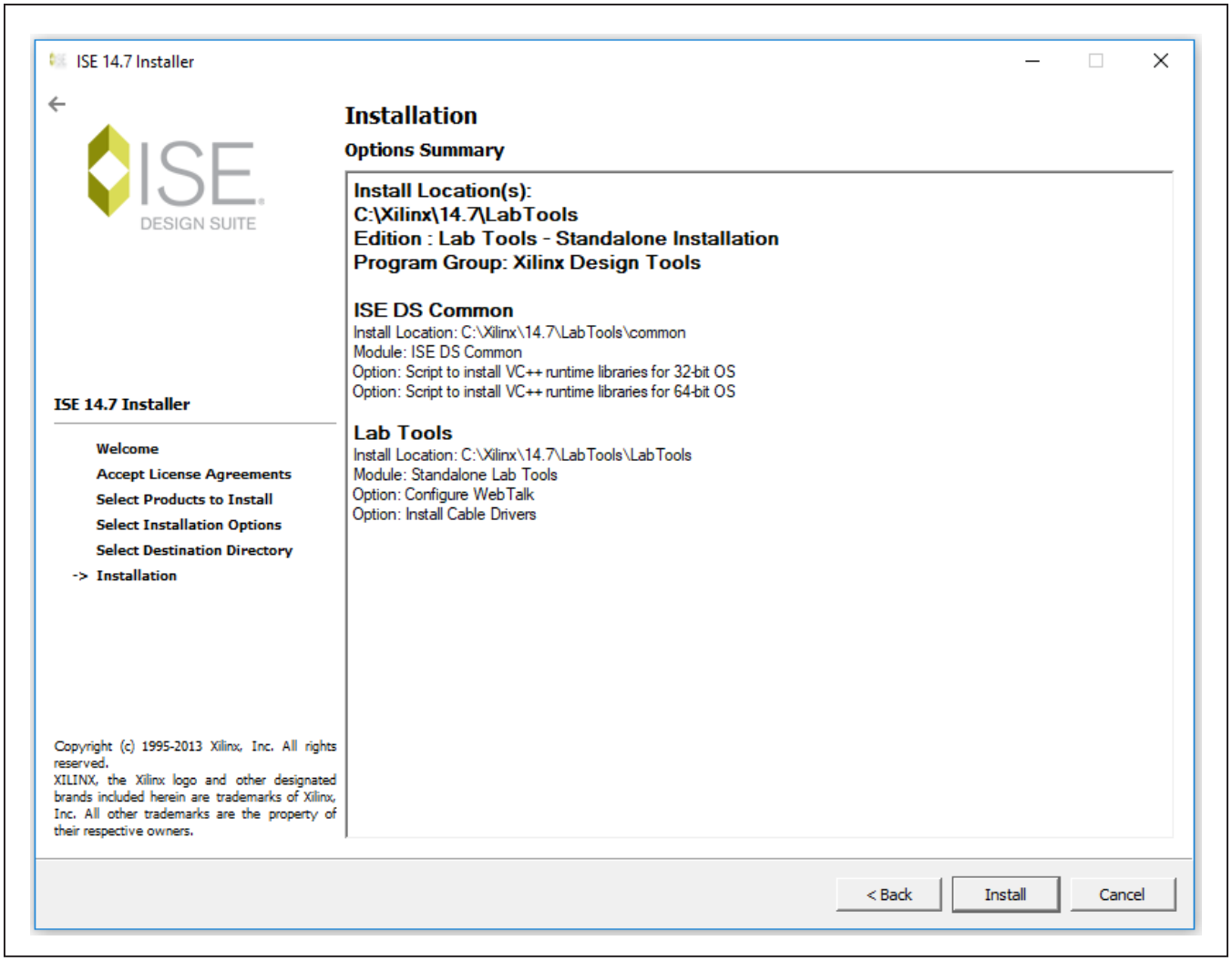
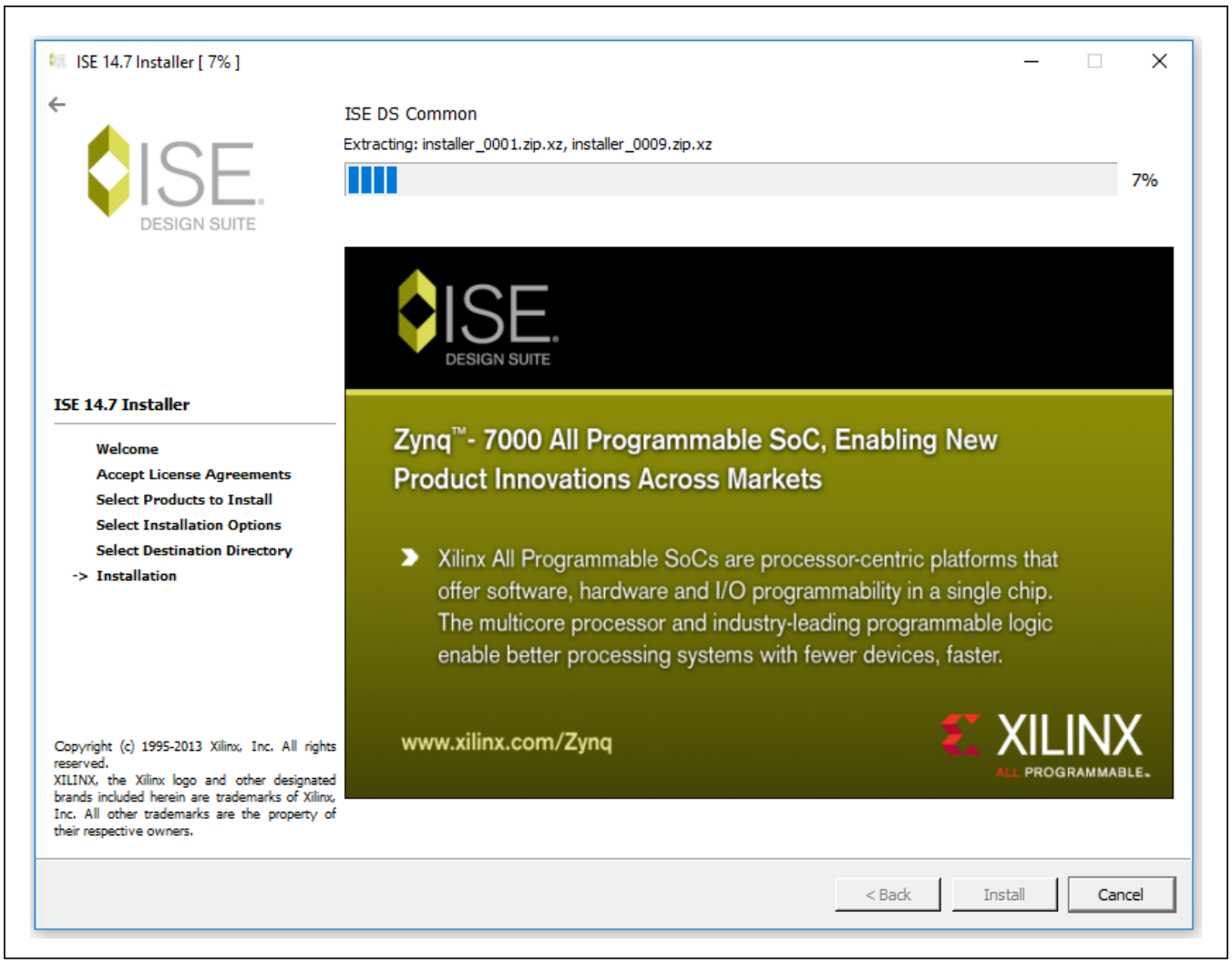
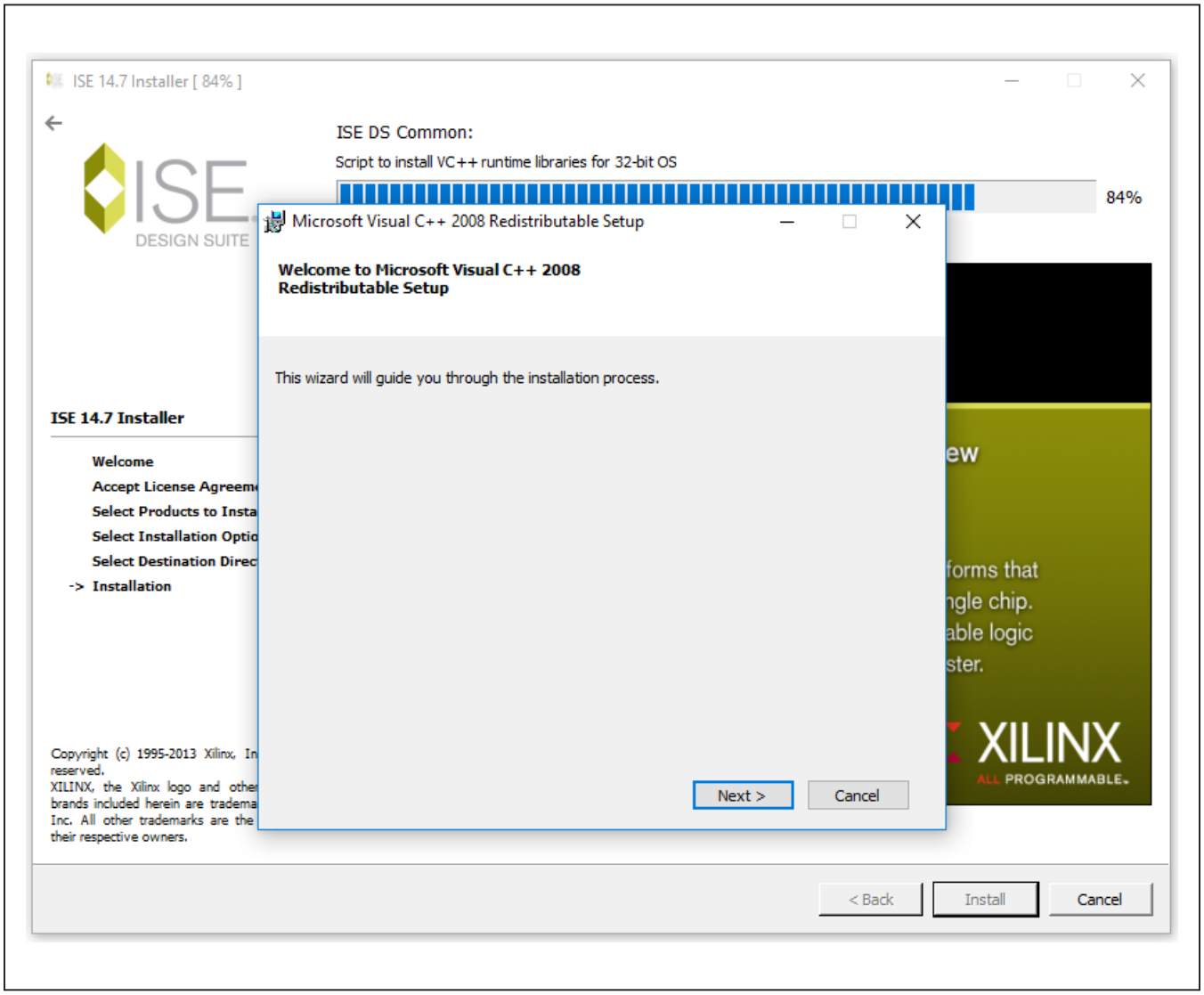


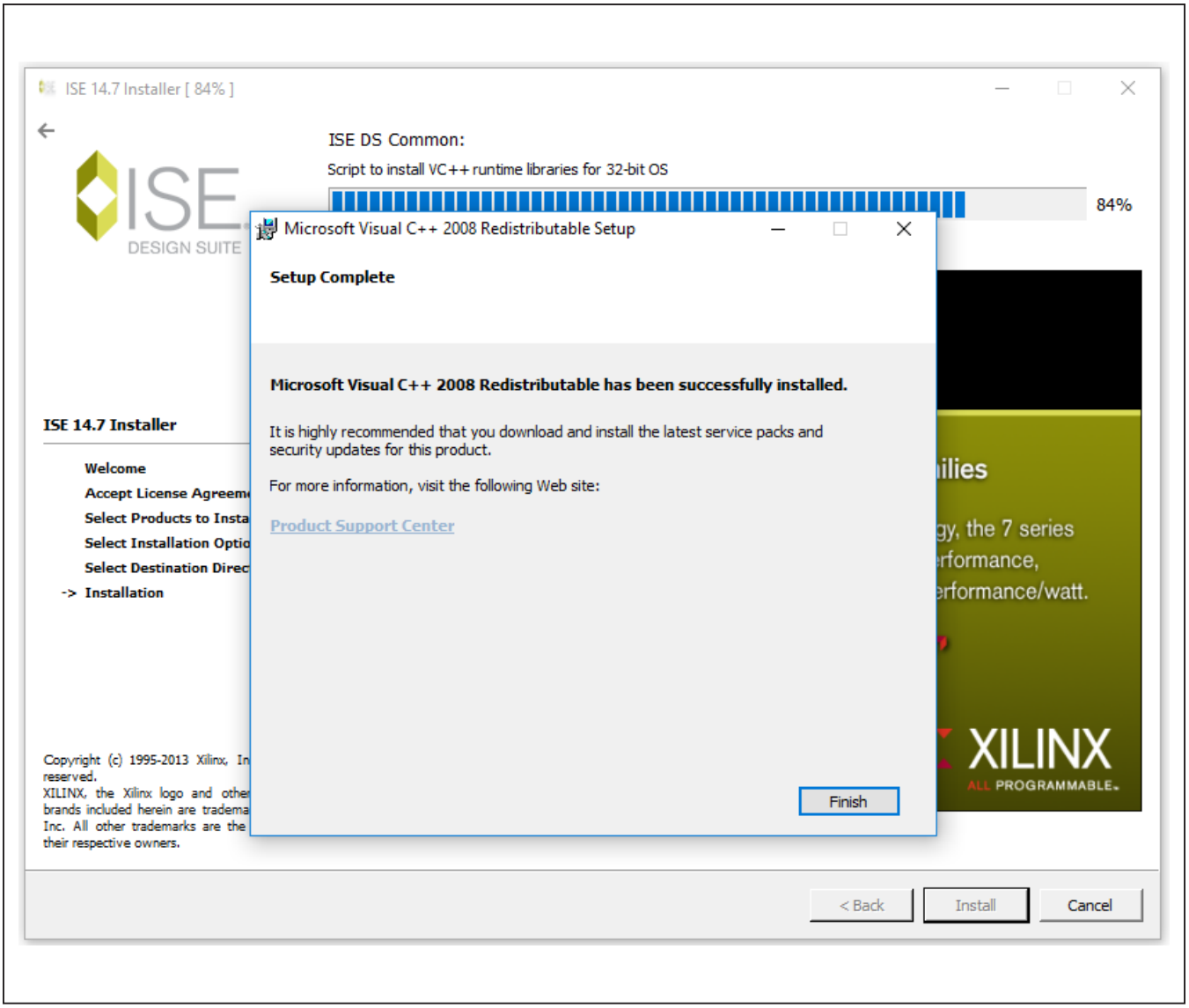
Figure A1-4b. Various LabTools Installation Screens

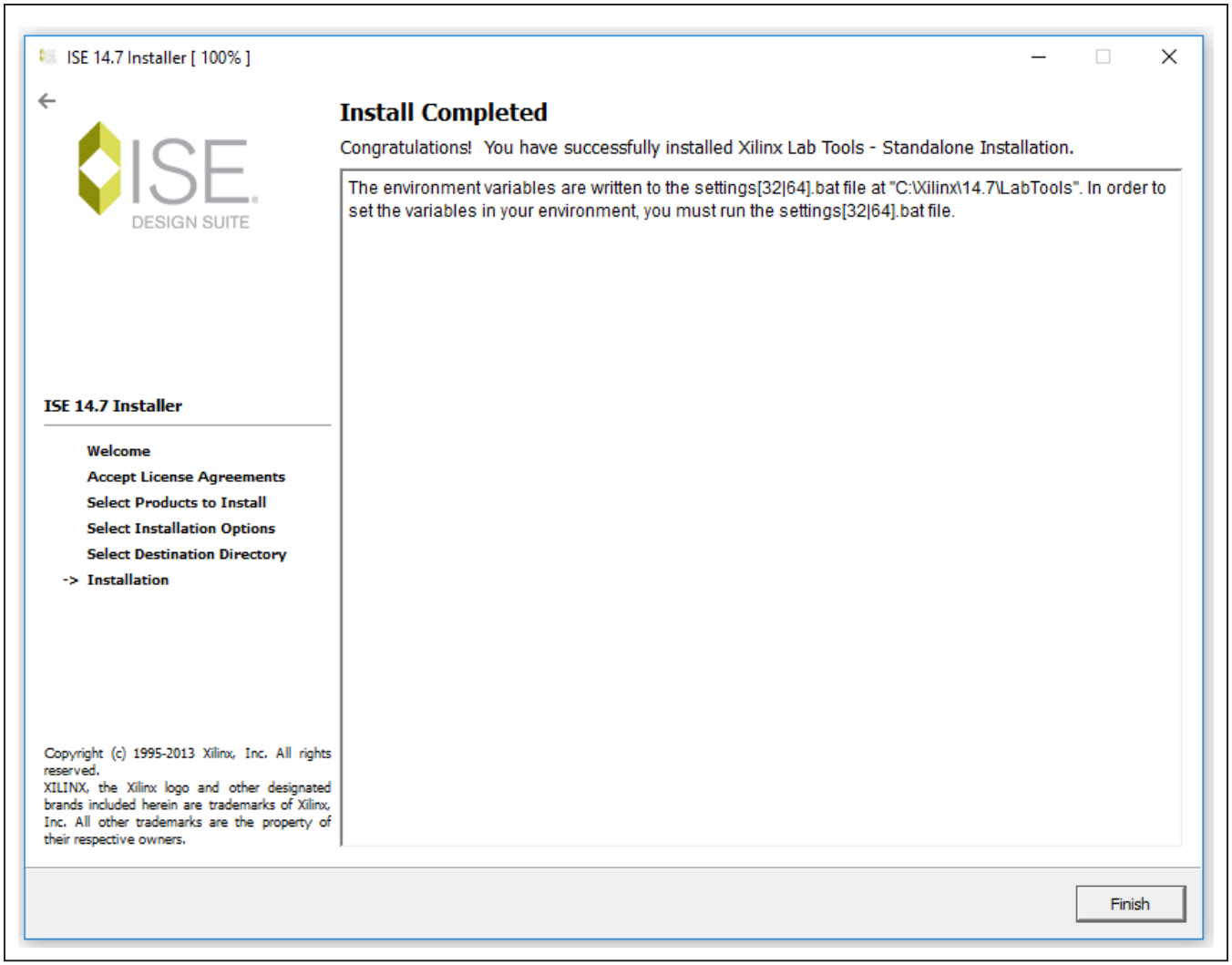

ISE 14.7 Installer [7%]
ISE DS Common
Extracting: installer_0001.zip.xz, installer_0009.zip.xz
7%
ISE DESIGN SUITE
ISE 14.7 Installer
Welcome
Accept License Agreements
Select Products to Install
Select Installation Options
Select Destination Directory
-> Installation
Zynq™- 7000 All Programmable SoC, Enabling New Product Innovations Across Markets
Xilinx All Programmable SoCs are processor-centric platforms that offer software, hardware and I/O programmability in a single chip. The multicore processor and industry-leading programmable logic enable better processing systems with fewer devices, faster.
www.xilinx.com/Zynq
XILINX
ALL PROGRAMMABLE.
Copyright (c) 1995-2013 Xilinx, Inc. All rights reserved.
XILINX, the Xilinx logo and other designated brands included herein are trademarks of Xilinx, Inc. All other trademarks are the property of their respective owners.
< Back
Install
Cancel

9) The ISE installation may require Microsoft Visual C++ 2008 Redistribution files, install these during the process if necessary.
   a. On the "…Redistribution Setup" page Click on Next > to continue.
   b. The "License Terms" page will appear, check the box and click on the <Install> button to continue.
   c. When the "Setup Complete" page appears, click on the <Finish> button to continue.
10) The installation may also require Jungo software to be installed separately, if prompted by Windows Security click on the <Install> button to continue.
11) The installation will also prompted Windows Security to install "Xilinx" software, click on the <Install> button to continue.
12) A final "Install Completed" page will appear, click on <Finish> to complete the ISE LabTools installation process

Note: *Notifications may vary slightly for Windows 10 OS.*



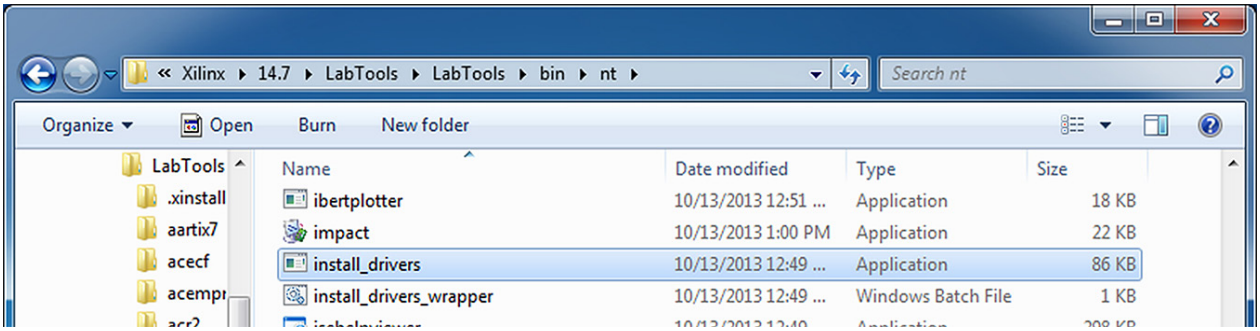

ISE 14.7 Installer [84%]
ISE DS Common:
Script to install VC++ runtime libraries for 32-bit OS
84%
Microsoft Visual C++ 2008 Redistributable Setup
Setup Complete
Microsoft Visual C++ 2008 Redistributable has been successfully installed.
It is highly recommended that you download and install the latest service packs and security updates for this product.
For more information, visit the following Web site:
Product Support Center
Finish
ISE 14.7 Installer
Welcome
Accept License Agreem
Select Products to Insta
Select Installation Optio
Select Destination Direc
-> Installation
Copyright (c) 1995-2013 Xilinx, In
reserved.
XILINX, the Xilinx logo and othe
brands included herein are tradema
Inc. All other trademarks are the
their respective owners.
ilies
gy, the 7 series
rformance,
erformance/watt.
XILINX
ALL PROGRAMMABLE.
< Back
Install
Cancel


ISE 14.7 Installer [100%]
Install Completed
Congratulations! You have successfully installed Xilinx Lab Tools - Standalone Installation.
The environment variables are written to the settings[32|64].bat file at "C:\Xilinx\14.7\LabTools". In order to set the variables in your environment, you must run the settings[32|64].bat file.
ISE DESIGN SUITE
ISE 14.7 Installer
Welcome
Accept License Agreements
Select Products to Install
Select Installation Options
Select Destination Directory
-> Installation
Copyright (c) 1995-2013 Xilinx, Inc. All rights reserved.
XILINX, the Xilinx logo and other designated brands included herein are trademarks of Xilinx, Inc. All other trademarks are the property of their respective owners.
Finish

Xilinx Drivers Installation

After the LabTools have been installed on a PC, the VC707 USB interface drivers need to be installed to access the JTAG port.

1) Browse to the Xilinx folder created during the installation of LabTools:
   for 32-bit OS - C:\Xilinx\14.7\LabTools\LabTools\bin\nt
   for 64-bit OS - C:\Xilinx\14.7\LabTools\LabTools\bin\nt64 (64-bit OS)
2) Execute the install_drivers.exe application.



3) A Command Prompt window may briefly flash, and a message may appear stating “This program might not have installed correctly”, click on “This program installed correctly” to continue. Note that messages might vary slightly for Windows 10 OS.

After these drivers are installed the JTAG port on the VC707 will be registered to the PC’s Device Manager.

LIBUSB Driver Installation

This driver is needed to properly interface to the USB 2.0 bulk port on the VC707. This port provides fast, bulk data transfers stored in the DDR and used to generate DAC data patterns. The USB 2.0 port is also used to control the FPGA and as a pass-through control port for the MAX5855 SPI.

A \USB_MS_Bulk_Transfer folder can be found on the flash drive and is also created during the MAX5855 EV kit software installation. This contains the “libusb-win32 devices” driver for the bulk port. Since the VC707 does not inherently use the USB 2.0 (ULPI) port, this driver can only be installed after the .bit file has been programmed into the Virtex FPGA.

Removal/Uninstallation of the MAX5855 EV kit Software

The MAX5855EVKIT Software Controller application can be removed from the system by running the unins000.exe executable located with the C:\MaximIntegrated\MAX5855EVKIT folder. This will remove any files placed on the system by the installation program. There may be residual files created after the installation process that can be removed by deleting the MAX5855EVKIT directory after running the uninstall program.

Upgrading/Updating the MAX5855 EV kit Software

Occasionally upgrades or updates may be available for the MAX5855 EV kit Controller Software. Be sure to check the **www.maximintegrated.com** website from time-to-time for information on the MAX5855 evaluation system. Applying upgraded software may require the removal of previous versions.

This document was written based on Revision 1.1.1 (FEB 2018) of the MAX5855 RF DAC EV kit Software.

Appendix II – Interface Commands

List of VC707 Control Commands

COMMAND	DESCRIPTION
help	Prints help. Use -a flag for all commands. Try “help -a” now!
mem	Memory operation
usb	USB operation
reg	Read / Write a register
capture	Manage capture DMA channel
play	Manage play DMA channel
ping	Does nothing but ack
baudrate	Set baud rate
spi	SPI commands to the MAX5855
i2c	I²C commands to the MAX6654 temperature sensor
gpio	GPIO commands which control or read pins on the MAX5855 EV kit board
init	Initialization
quit	Quits this program

Description and Syntax of VC707 Control Commands

Help

Online help is provided for all commands. Short and long version are available.

COMMAND	SYNTAX	EXAMPLES
help	<command1>,…,<commandN>,<-flags>	help mem help -all

ARGUMENTS/FLAGS	DESCRIPTION
<command>	Name of command, or command and subcommand to request help information about
-all **-a**	Print help for all the commands

Memory Operations

The **mem** commands are for reading and writing the DDR memory used for test data. Read a specific number of 32 bit words from a given address of DDR memory or write a specific number of bytes to a given address of DDR memory.

Flags allow different communication formats. The default mode is to send data in ASCII format. Use the -b flag to send/receive in binary mode. When the FPGA is sending in binary mode, the proper number of bytes will be sent, followed by an ACK. There will not be a CR/LF between the data and the ACK.

Follow the -b flag on a write request with a CR and LF. There will not be an ACK at this point. Then send the proper amount of binary data. Do not follow the binary data with a CR/LF. After the proper number of bytes has been received, an ACK will be sent.

COMMAND	SYNTAX	EXAMPLES
mem read	[adr] [num bytes] <-flags>	mem read 0x100 8 -c
mem write	[adr] [num bytes] <-flags> [word1] … [wordN]	mem write 0x100 8 -c 0x64636261 0x68676665

ARGUMENTS/FLAGS	DESCRIPTION
[adr]	Address to read from or write to, in any format that strtoul will parse
[num bytes]	Number of bytes to read or write in multiples of 4
-binary -b	Send data in binary mode
-chksum -c	Send 16 bit IPv4 checksum at the end of the data
[word]	32 bit values in any format that strtoul will parse

USB Operations

The **usb** commands are for reading and writing the DDR memory used for test data. Read a specific number of bytes from a given address of DDR memory or write a specific number of bytes to a given address of DDR memory. The data are transferred using a USB BULK OUT or USB BULK IN format.

After the proper number of bytes has been received or sent, an ACK will be returned.

COMMAND	SYNTAX	EXAMPLES
usb read	[adr] [num bytes]	usb read 0x100 8 -c
usb write	[adr] [num bytes] [word1] … [wordN]	usb write 0x100 8 -c 0x64636261 0x68676665

ARGUMENTS/FLAGS	DESCRIPTION
[adr]	Address to read from or write to, in any format that strtoul will parse
[num bytes]	Number of bytes to read or write in multiples of 4
[word]	32 bit values in any format that strtoul will parse

Register Operations

The **reg** commands are for reading and writing registers in the MAX5855. Read and write transactions use a 32-bit word. The **list** operation returns a list of the register or register spaces and when used with the **set** operation, it allows the user to write to control register fields by name. Only the bits of that field within the control register are modified.

After the proper number of bytes has been received or sent, an ACK will be returned.

COMMAND	SYNTAX	EXAMPLES
reg read	[adr]	reg read 0x800
reg write	[adr] [word]	reg write 0x800 0xA002007F
reg list	<reg space>	reg list
reg set	<reg space> <reg name> [value]	reg set

ARGUMENTS/FLAGS	DESCRIPTION
[adr]	Address to read from or write to, in any format that strtoul will parse
[word]	32 bit values in any format that strtoul will parse
<reg space>	List the hierarchical register spaces, or the registers in a register space. For the set command, <reg space> is optional
<reg name>	name of the control register field to set
[value]	value to write to the control register field

Valid register names are shown in the GUI software under the *Register Access Tab*.

Capture Operations

The **capture** commands are for configuring, starting, and stopping the capture DMA channel. *This set of operations is not commonly used with the RF DACs*. The **buffer** operation allows the user to configure the base address or starting address, and length of the capture buffer. The base address must start on a 64-byte alignment and be a multiple of 512 bytes. The RX DMA engine must be stopped before using this command. The **dumpregs** operation will print the capture channel registers, effectively dumping the contents of the RX DMA engine.

The **start** operation will begin the capture channel or start the RX DMA engine. The **capture buffer** command must be used first to define the buffer. The **stop** operation will close the capture channel or stop the RX DMA engine.

COMMAND	SYNTAX	EXAMPLES
capture buffer	[adr] [num bytes]	capture buffer 0x400 512
capture dumpregs		capture dumpregs
capture start		capture start
capture stop		capture stop

ARGUMENTS/FLAGS	DESCRIPTION
[adr]	Address to read from or write to, in any format that strtoul will parse
[num bytes]	Number of bytes to read or write in multiples of 512

Play Operations

The **play** commands are for configuring, starting, and stopping the play/TX DMA channel.

The **buffer** operation allows the user to configure the base address or starting address, and length of the play buffer. The base address must start on a 64-byte alignment and be a multiple of 512 bytes. The TX DMA engine must be stopped before using this command. The **dumpregs** operation will print the play channel registers, effectively dumping the contents of the TX DMA engine.

The **start** operation will begin the play channel or start the TX DMA engine. The **play buffer** command must be used first to define the buffer. The **stop** operation will close the play channel or stop the TX DMA engine.

COMMAND	SYNTAX	EXAMPLES
play buffer	[adr] [num bytes]	play buffer 0x400 512
play dumpregs		play dumpregs
play start		play start
play stop		play stop
play reset		play reset

ARGUMENTS/FLAGS	DESCRIPTION
[adr]	Address to read from or write to, in any format that strtoul will parse
[num bytes]	Number of bytes to read or write in multiples of 512

Ping Command

The **ping** command is a non-operation which simply returns an ACK from the system.

Baudrate Command

The **baudrate** command sets the communication rate for the UART interface.

SPI Operations

The **spi** commands are for reading and writing to the SPI bus which only interfaces to the MAX5855 RF DAC. The **read** operation will receive one byte from a given register address whereas the **read multiple bytes** will receive N bytes from the registers starting at the base address provided. The **write** operation will transmit a single byte to a given register address whereas the **write multiple bytes** operation will transmit N bytes to the registers starting at the address provided.

After the proper number of bytes has been received or sent, an ACK will be returned.

COMMAND	SYNTAX	EXAMPLES
spi read	[adr]	spi read 0x100
spi read multiple bytes	[adr] [N bytes]	spi read multiple bytes 0x100 4
spi write	[adr] [byte]	spi write 0x100 0x24
spi write multiple bytes	[adr] [N bytes] [word]	spi write multiple bytes 0x100 4 0x64636261

ARGUMENTS/FLAGS	DESCRIPTION
[adr]	Address to read from or write to, in any format that strtoul will parse
[N bytes]	Number of bytes to read or write in multiples of 4
[byte]	8 bit value in an format that strtoul will parse
[word]	32 bit values in any format that strtoul will parse

I2C Operations

The **i2c** commands are for reading and writing to the I2C bus which only interfaces to the MAX6654 temperature monitor. The **read** operation will receive one byte from a given register address whereas the **read multiple bytes** will receive N bytes from the registers starting at the base address provided. The **write** operation will transmit a single byte to a given register address whereas the **write multiple bytes** operation will transmit N bytes to the registers starting at the address provided.

After the proper number of bytes has been received or sent, an ACK will be returned.

COMMAND	SYNTAX	EXAMPLES
i2c read	[adr]	i2c read 0x08
i2c read multiple bytes	[adr] [N bytes]	i2c read multiple bytes 0x08 4
i2c write	[adr] [byte]	i2c write 0x08 0x24
i2c write multiple bytes	[adr] [N bytes] [word]	i2c write multiple bytes 0x08 4 0x22320010

ARGUMENTS/FLAGS	DESCRIPTION
[adr]	Address to read from or write to, in any format that strtoul will parse
[N bytes]	Number of bytes to read or write in multiples of 4
[byte]	8 bit value in an format that strtoul will parse
[word]	32 bit values in any format that strtoul will parse

GPIO Operations

The **gpio** commands are for controlling or reading the status of the general purpose I/O lines which are used to drive or are connected to pins on the MAX5855 EV kit board. The **resetb** operation is active-low signal, using the **-y** flag will reset the DAC, using the **-n** flag will disable reset. The **mute** operation is an active-high signal, using the **-y** flag will mute the DAC, using the **-n** flag will unmute the DAC. The **intb** line is connected to the active-low interrupt pin of the DAC, using the **intb** operation will read the status of the INTB pin. The **altb** line is connected to the alarm (alert) pin of the MAX6654, using the **altb** operation will read the status of the ALARMB pin of the temperature sensor.

COMMAND	SYNTAX	EXAMPLES
gpio resetb	[-y|-n]	gpio restb -y
gpio mute	[-y|-n]	gpio mute -n
gpio intb		gpio intb
gpio altb		gpio altb

ARGUMENTS/FLAGS	DESCRIPTION
[-y|-n]	Set (-y) or clear (-n) the appropriate line

Init Command

The **init** command is used to initiate communication with the FPGA.

Quit Command

The **quite** command closes communication with the FPGA.

List of JESD Commands

<table>
<tr><th>COMMAND</th><th colspan="2">DESCRIPTION</th></tr>
<tr><td>resync</td><td colspan="2">Resynchronize the JESD204 interface.</td></tr>
<tr><td>scramble</td><td colspan="2">Turn JESD204B scrambling on or off for RX and TX</td></tr>
<tr><td>loopback</td><td colspan="2">Set JESD204B internal loopback. Use -y to turn on loopback, and use -n to turn it off.</td></tr>
<tr><td>jesdreset</td><td colspan="2">Reset JESD204B PLL and resynchronize all lanes.</td></tr>
<tr><td>configure</td><td colspan="2">Set the JESD204B configuration to 623, 545, 422, or 343. This command will set the mapper mode, number of lanes, number of octets per frame, and number of frames per multi-frame. The parameter <abc> is a three digit number corresponding to the mapper mode desired. Each digit represents the following:
a -> Number of Lanes
b -> Octets per Frame
c -> Samples per Frame per Converter</td></tr>
<tr><td rowspan="4">jesd204 rate</td><td colspan="2">Sets the JESD204B lane rates. The parameters <f> and [d] are integers that represent the following:
f -> Full Lane Rate (1 => 6.144Gbps, 2 => 7.3728Gbps, 3 => 9.8304Gbps)
d-> Lane Rate Divider (Lane Rate will equal Full Lane Rate divided by ‘d’)
The [d] parameter is optional and will default to 1 for full lane rate if omitted.
Examples:</td></tr>
<tr><td>JESD204 COMMAND</td><td>LANE RATE SETTING ACHIEVED</td></tr>
<tr><td>jesd204 rate 1 2</td><td>6.144Gbps/2 = 3.072Gbps</td></tr>
<tr><td>jesd204 rate 2
or jesd204 rate 2 1</td><td>7.3728Gbps/1 = 7.3728Gbps</td></tr>
<tr><td>jesd204 rate 3 4</td><td>9.8304Gbps/4 = 2.4576Gbps</td></tr>
</table>

Appendix III – Pattern Files

Creating Pattern Files

The MAX5855 EV kit Software Controller is provided with a limited number of sample patterns. The provided patterns allow the user to generate a two-tone, CW signal with a 1MHz spacing between the tones. Typically, the user will want to generate signals with other properties, including modulated signals.

The pattern file can use any extension, such as csv, txt, etc., as long as the contents are simple text and conform to the format expected by the MAX5855EV kit Software Controller.

The specific format is as follows:

- The first line contains only the total number of I/Q data points, N, in the pattern.
- The total number of lines in the pattern file will be N + 1.
- The second through N lines contain four, comma separated integer values. These values represent, in order:
  - the I data Least Significant BYTE (LS BYTE)
  - the I data Most Significant BYTE (MS BYTE)
  - the Q data LS BYTE
  - the Q data MS BYTE.

Note: *I/Q data is in decimal format, offset binary with the 14-bits of data being LSB-justified. That is, the MSB values can range from 0 to 63 and the LSB values can range from 0 to 255.*

For example, the first few lines of the file could look something like this:

```
65536
19, 62, 253, 37
19, 62, 250, 37
17, 62, 248, 37
15, 62, 245, 37
13, 62, 242, 37
10, 62, 239, 37
7, 62, 236, 37
3, 62, 234, 37
```