

## MAX5869 Evaluation Kit

Evaluates: MAX5869

### General Description

The MAX5869 evaluation kit (EV kit) contains a single MAX5869 high-performance interpolating and modulating 16-bit 5.9Gbps RF DAC that can directly synthesize up to 600MHz of instantaneous bandwidth from DC to frequencies greater than 2.8GHz. The device is optimized for cable and digital video broadcast applications and meets spectral mask requirements for a broad set of communication standards, including EPoC, DVB-T, DVB-T2, DVB-C2, ISDB-T, and DOCSIS 3.0/3.1. The EV Kit provides a complete system for evaluating the performance of the MAX5869 device and a platform for developing a digital video solution.

The MAX5869 EV kit connects to one FMC connector on the Xilinx® VC707 evaluation kit, allowing the VC707 to communicate with the MAX5869's JESD204B serial link interface.

The EV kit includes Windows® 7/8/10 compatible software that provides a simple graphical user interface (GUI) for configuration of all of the MAX5869 registers through the SPI interface, control of the VC707 FPGA, and temperature monitoring.

### Features

- Evaluates MAX5869 RF DAC Performance, Capability and Feature Set
- Single 3.3V Input Voltage Supply
- Maximum 5.9Gbps Update Rate
- Direct Interface with Xilinx VC707 Data Source Board
- Windows 7/8/10 Compatible Software
- Optional On-Board SPI Interface Control for the MAX5869
- On-Board SMBus™ Interface Control for the MAX6654 Temperature Sensor
- GUI Controls for VC707 Operation
- Proven 10-Layer PCB Design
- Fully Assembled and Tested

Ordering Information appears at end of data sheet.

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

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

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

*Xilinx is a registered trademark of Xilinx.*

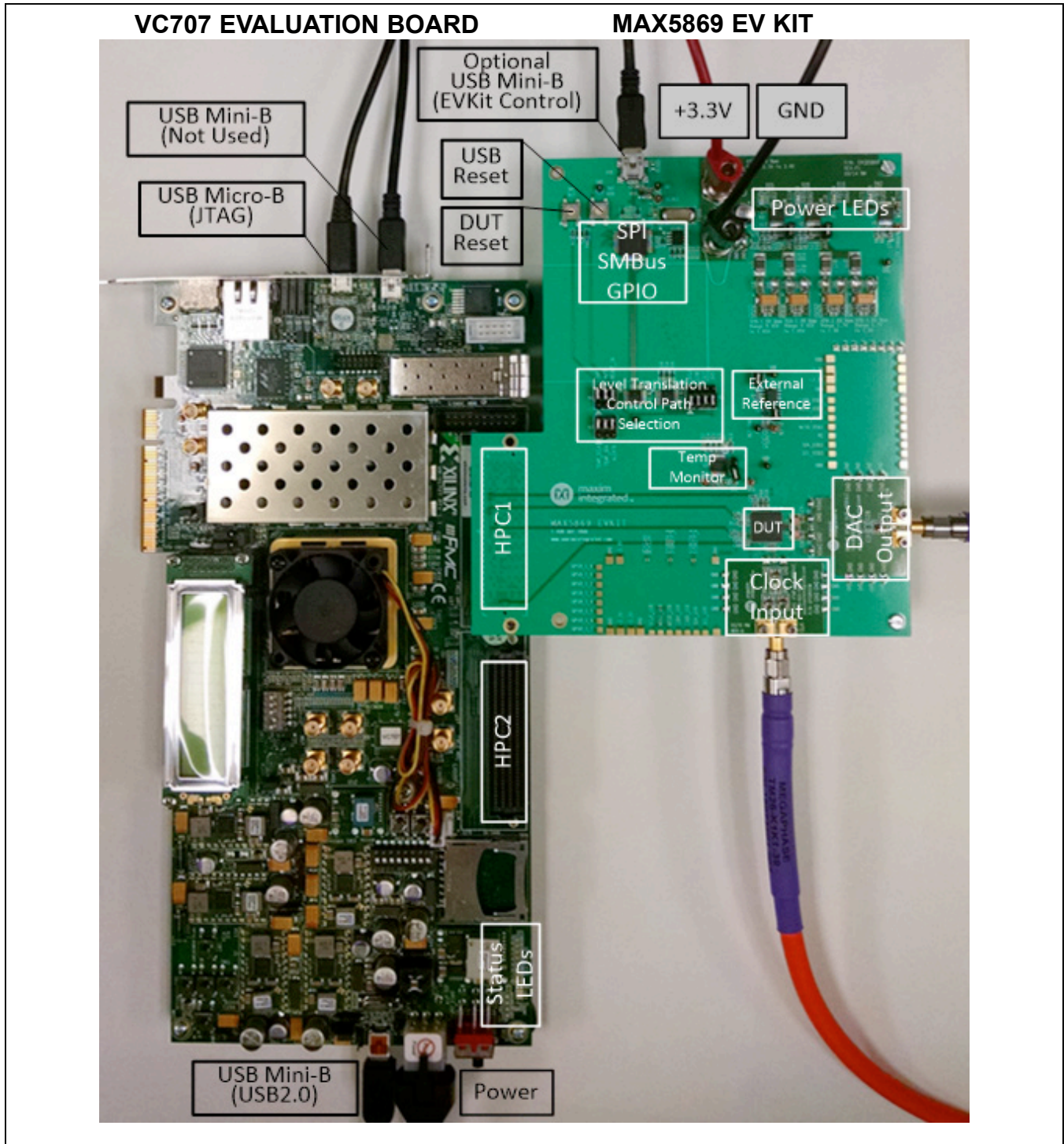


Figure 1. MAX5869/VC707 Evaluation Setup

## Initial Setup

### Required Equipment

- Window PC (Win-7/10 Recommended, Win-XP, Win-8 optional), with two USB 2.0 ports available
- Spectrum Analyzer – Agilent PXA or equivalent
- RF signal generator – Rohde & Schwarz SMF100A or equivalent
- 3.3V, 3A power supply for MAX5869 EV Kit
- Xilinx VC707 Evaluation Kit—user-supplied
  - VC707 board
  - 12V/5A power cube
  - 1 each USB-A to Mini-B cable for interfacing and programming
  - 1 each USB-A to Micro-B cable for interfacing and programming
- Low-loss SMA/SMA cables, as needed for connections to the spectrum analyzer and signal generator
- Included in the MAX5869 EV kit
  - Two 1" stand-offs with screws
  - 1" stand-offs with screws
  - MAX5869 EV kit board
  - One USB-A to Mini-B cables, for optional USB/FTDI control of the SPI and SMBus interfaces on the MAX5869 EV Kit.

### Required Software and Drivers

The MAX5869EVKIT software controller application requires the following drivers and software components to be installed:

- **Xilinx ISE 14.7 LabTools**  
Installation: Lab Tools is a free tool set used for programming the VC707 Evaluation Board, no software registration or license is required. However, a user account is required to download the ISE 14.7 installation package from the Xilinx web site. A DLL work-around for Windows 10 as well as additional, detailed information about the installation of LabTools can be found in Appendix I of this document. Alternatively, LabTools can be downloaded from the following location:  
<http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/design-tools.html>

- **Xilinx Drivers**

Installation: Browse to the Xilinx folder created during the installation of LabTools: C:\Xilinx\14.7\LabTools\LabTools\bin\nt (or nt64). Execute the install\_drivers.exe application

If the above files have not been installed, please allow up to 30 minutes for installation.

### Install the MAX5869EVKIT Software

The MAX5869EVKIT Software Controller application can be obtained from the [www.maximintegrated.com](http://www.maximintegrated.com) website. Select the **DESIGN**→**DESIGN TOOLS** menu option in the upper-left corner of the page. A new page will appear with several additional links. Under the **Applications Software** section, select the **EV Kit Software** option. The next page will contain a listing of all EV kit software available for download. The page provides a simple filter to ease the users search. Select the **Digital-to-Analog Converters** option if desired, then find and download the MAX5869 RF DAC EV Kit Software.

It is strongly suggested to use the default installation path (**c:\MaximIntegratedMAX5869EVKIT**). 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 **10 minutes**.

### Setup and Connect the MAX5869 EV Kit Hardware board (Figure 1)

- 1) Install the two 1" stand-offs included with the MAX5869 EV kit. Stand-offs should be installed on the DAC output side of the board.
- 2) Verify all jumpers on the MAX5869 EV kit PCB are in the default position; refer to [Table 2](#).
- 3) Connect the MAX5869 EV kit board to the VC707 board HPC1 FMC connector. Refer to [Figure 1](#).
- 4) Connect the 3.3V/3A supply to the MAX5869 EV kit and enable the output. Verify the four LED board supply indicators are on and green.
- 5) Connect the RF generator to the clock module with a low-loss SMA cable, set the frequency to 2.5GHz with output power at +3dBm.
- 6) Turn on the VC707 by sliding switch SW12 to the on (left) position. Verify all LEDs on the VC707 are on momentarily; the GPIO LEDs should then begin sequencing.

- 7) Make the USB connections using the 4-port USB2.0 hub.
  - a. Connect the USB A–micro B cable (JTAG) from Xilinx VC707 Eval board to the PC.
  - b. Connect the USB A–micro B cable (USB2.0) from Xilinx VC707 eval board to the PC.
  - c. Optionally connect the USB A–Mini B cable from the MAX5869 mini USB to the PC.

Please ensure that all the USB device drivers are installed and 'ready for use' (this may take several minutes, depending on the system) before proceeding to the next step. The drivers will typically install automatically when the USB connection is first made on a given port. The FTDI device on the MAX5869 EV kit is recognized as four separate USB devices and four separate COM ports, and the driver must be installed for each device and port. The Windows OS reports new device arrivals in the Notification Area of the Task Bar.

**Table 1. Installed Files and Folders**

FILE	DESCRIPTION
MAX5869EVKITSoftwareController.exe	Application program
AppFiles	Directory with application support files including the USB_MS_Bulk_Transfer driver
DeviceScripts	Directory with sample MAX5869 configuration scripts and Perl scripts for generating additional scripts
DeviceScripts\PERL	Directory with Perl scripts and supporting files to generate new configuration files to load into the MAX5869
PatternFiles	Directory with sample pattern files and Matlab routines for generating additional CW patterns
VC707Files	Directory with FPGA programming file and supporting documentation
EVKIT Info	Directory with PCB design details and automation support document
Screenshots	Directory with example spectrum analyzer screen captures
Miscellaneous DLLs to include ftd2xx.dll, DTD2XX_NET.dll, libMPSSSE.dll and MaximStyle.dll	Supporting DLL files for software operation

**Table 2. MAX5869 EV Kit Jumper Settings**

JUMPER	POSITION	EVKIT FUNCTION
JU1	Not Installed*	Normal Operation
JU2	Installed*	1.0V LDO drives MAX5869: AVCLK, AVDD1_PLL, AVDD
JU3	Installed*	1.8V LDO drives MAX5869: AVCLK2, AVDD2, RVDD2, AVDD2PLL, VDD2
JU4	Installed Not Installed*	Power for U4 – MAX6161 – external reference MAX6161 NOT powered
JU5	Installed Not Installed*	MAX5869 external reference connected MAX5869 using internal reference
H3	1-2, 4-5,7-8,10-11 2-3*,5-6*,8-9*,11-12*	SCLK, SDI, SDO, CSA pins connected to USB SCLK, SDI, SDO, CSA pins connected to FPGA
H6	2-3,5-6,8-9 1-2*,4-5*,7-8*	INTB, MUTE, RESETB connected to USB INTB, MUTE, RESETB connected to FPGA
H7	2-3,5-6,8-9 1-2*,4-5*,7-8*	SCL, SDA, ALERT (I <sup>2</sup> C) connected to USB SCL, SDA, ALERT (I <sup>2</sup> C) connected to FPGA

\*Default position.



- 8) **Start the** MAX5869EVKITSoftwareController.exe located in the C:\MaximIntegrated\MAX5869 folder. The Splash Screen will display while the USB connections are established, followed by the VC707 tab of the MAX5869 GUI.
  - a. After the splash screen, the application will display a window to select the SPI control path, either through the FPGA or through the FTDI/USB interface. Select the FPGA path for initial setup and operation using the examples in this guide.  
Future executions of the program can avoid this window by checking the box to remember the selection.
- 9) **Load the FPGA configuration**
  - a. Click on the Xilinx Impact Tool Installed checkbox. A file browser window will open. Locate the directory where the impact.exe program is located. If the default installation location is used for the Xilinx Lab Tools installation, the path will be:
    - i. For 32-bit operating system C:\Xilinx\14.7\LabTools\LabTools\bin\nt. Double click on the file impact.exe
    - ii. For 64-bit operating system C:\Xilinx\14.7\LabTools\LabTools\bin\nt64. Double-click on the file impact.exe
  - b. Click the <Load FPGA Configuration File> button.
  - c. A file browser will open in the C:\maximintegrated\MAX5869\VC707Files folder. Double click the MAX5869\_DataSource.bit file.
  - d. A progress bar will display while the FPGA is configured (should take less than 2 minutes).
  - e. 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 the *Device Manager*.
  - f. After allowing the connection to complete, select the *USB Mass Storage Device* in the Device Manager and right-click to select the Update Driver Option. **NOTE: Ensure the USB thumb drive has been ejected before proceeding.**
    - i. Select *Browse My Computer for driver software*.
    - ii. Select *Let Me Pick from a list of devices on my computer*.
    - iii. Select the *USB Mass Storage Device*, then click the <Have Disk> button.
    - iv. Click the <Browse> button in the Load from Disk pop up window.
    - v. Browse to C:\MaximIntegrated\MAX5869\AppFiles\ThirdParty\USB\_MS\_Bulk\_Transfer and select the USB\_MS\_Bulk\_Transfer.inf file.
    - vi. The MAX5869 software may show a window indicating it has encountered a problem; click on the Close button to continue.
- 10) **Reboot the PC and power-cycle the FPGA and EV kit system.**
  - a. Turn off the VC707 by sliding switch Main Power Switch to the OFF position
  - b. Turn of the power the MAX5869 EV Kit PCB
  - c. Reboot the PC  
After the PC has booted, the drivers will be properly configured for use with the MAX5869 software. Once the FPGA is programmed, the GUI Status Bar (bottom of window) should indicate the active connections as shown in [Figure 6](#).

## Quick Start

### Connect and Power-Up the MAX5869EVKIT Hardware

- 1) Connect the DAC output to the spectrum analyzer (see [Figure 1](#)). The settings below will match the test case used later in this quick start section:
  - a. Center frequency = 575MHz
  - b. Frequency span = 108MHz
  - c. Input attenuation = 6dB
  - d. Reference level = -25dBm
  - e. Detector mode = RMS
  - f. Resolution Bandwidth = 10kHz
- 2) Connect the RF Signal Generator to the CLOCK input on the MAX5869EV Kit
  - a. Amplitude = 0dBm
  - b. CW frequency = 491.52MHz
- 3) Connect the 3.3V power supply to the EV kit and mate the EV kit with the VC707
- 4) Enable the 3.3V power supply. Verify that the four green LED board supply indicators are lit.
- 5) Enable the RF Signal Generator output
- 6) Turn on the VC707 by sliding the Main Power Supply switch to the ON position
- 7) Verify all LEDs on the VC707 are lit, and the GPIO LEDs are sequencing (for an unprogrammed FPGA)

### Run the MAX5869 EV Kit Software

- 1) Start the MAX5869 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 **MAX5869\_DataSource.bit** file and click the **Open** button
- 3) To quickly load an example test case and data pattern
  - a. Click on the **Setup** tab of the GUI
  - b. Click the **Test Setup #1: 8 SCQAM Chan, -12dBFS, fs<sub>IN</sub> = 245Mpsps, 9.8G/1Lane, f<sub>OUT</sub> = 575MHz, f<sub>DAC</sub> = 4.9GHz** button
  - c. The software will perform the following sequence of events
    - i. Stop any pattern currently running
    - ii. Assert a device hardware Reset and hardware MUTE
    - iii. Load the appropriate DAC configuration
    - iv. Update the Clock and NCO tab of the GUI
    - v. Load the appropriate test pattern
    - vi. Synchronize the JESD204B Interface
    - vii. Start continuously looping the test pattern
    - viii. De-assert MUTE (hardware and/or software)
  - d. Observe the output on the spectrum analyzer
    - i. Click on the button to Display RF DAC output screenshot in the GUI
    - ii. A window will open displaying a screen captured from a spectrum analyzer ([Figure 2](#))
    - iii. The user's spectrum analyzer display should appear very similar

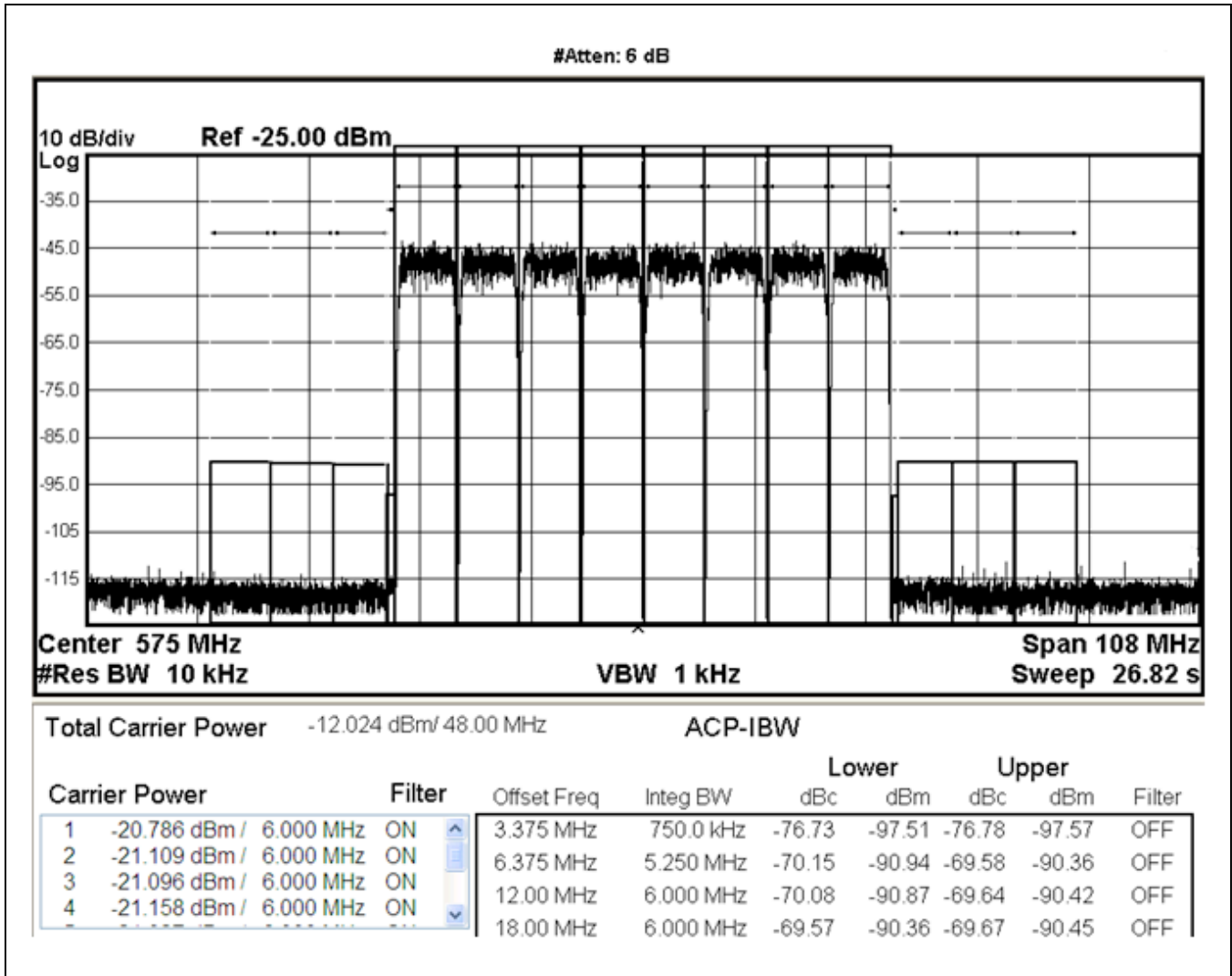


Figure 2. Spectral Output – Test Case #1, 8 SCQAM at 575MHz

**Detailed Description of Hardware**

**MAX5869 EV Kit Printed Circuit Board**

The MAX5869 EV kit PCB 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 MAX5869 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 MAX5869 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 I<sup>2</sup>C bus signals, as well as GPIO controls for the hardwired MUTE, INTB, and RESETB signals on the MAX5869. The FPGA pass-through or the on-board FTDI

interface are selected with jumpers installed on H3, H6, and H7 as shown in Figure 3a and Figure 3b.

**Interface Modules**

The MAX5869 EV kit employs two modules to allow for easy interfacing to Signal Generators and Spectrum Analyzers. The modular nature of these key interface circuits provides the user flexibility to adapt the EV Kit to custom interface requirements, allowing design and optimization at relatively low cost prior to committing the final product design.

The Clock Input Module (XFMR\_CLK\_MODULE) integrates a three-transformer single-ended to differential conversion circuit designed to be driven from a 50Ω RF signal source. The three transformers provide a symmetrical differential output that drives the CLKP/CLKN inputs of the MAX5869.

The output module (RFDAC\_XFMR\_OUT\_MODULE) employs a wideband RF transformer to convert the differential DAC output current of the MAX5869 RF DAC to a single ended 50Ω output suitable for driving the input of a 50Ω Spectrum Analyzer.

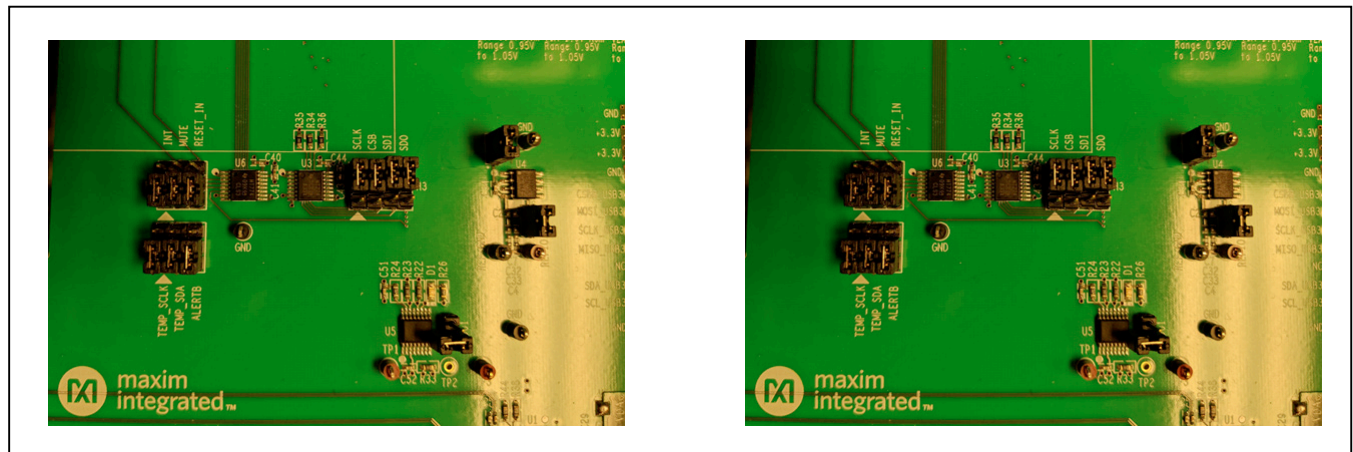


Figure 3. MAX5869EVKIT Jumpers – 3a. Default FPGA Pass-Through Interface; 3b. On-Board FTDI Interface



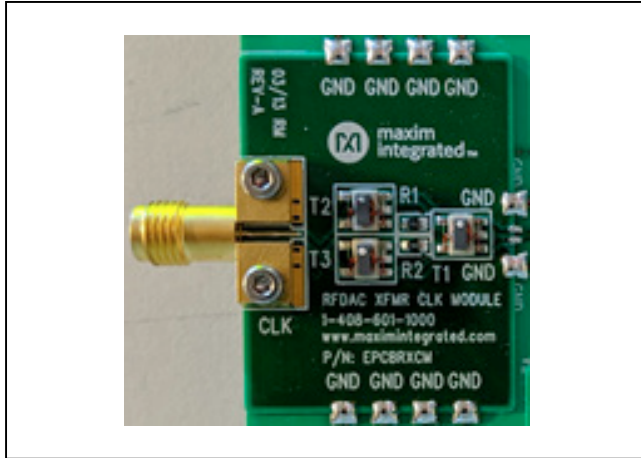


Figure 4. XFMR\_CLK\_MODULE



Figure 5. RFDAC\_XFMR\_OUT\_MODULE

**Power**

The MAX5869 EV kit board requires a single +3.3V, 3A 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 three MAX8527 linear regulators (LDOs), two which provide +1.8V rails and third providing +1.0V. The +1.0V and one of the +1.8V rails are used exclusively for the various supplies required by the MAX5869. The LDO outputs are isolated between the various analog and digital domains by on board filter networks. The PLL supplies for the MAX5869 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 V<sub>IN</sub> 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.

**Table 3. MAX5869 EV Kit LED Descriptions**

LED	COLOR	DESCRIPTION
D1	Red	Normally Off; Temperature alert based on DUT temperature and 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 MAX5869EVKIT +3.3V Power Indicator

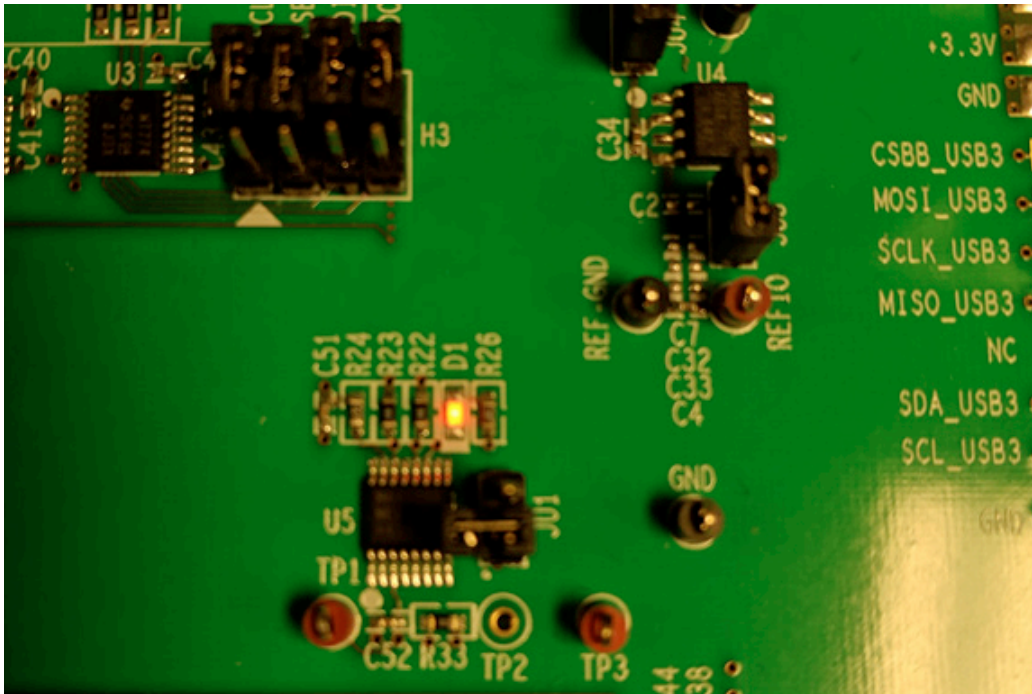
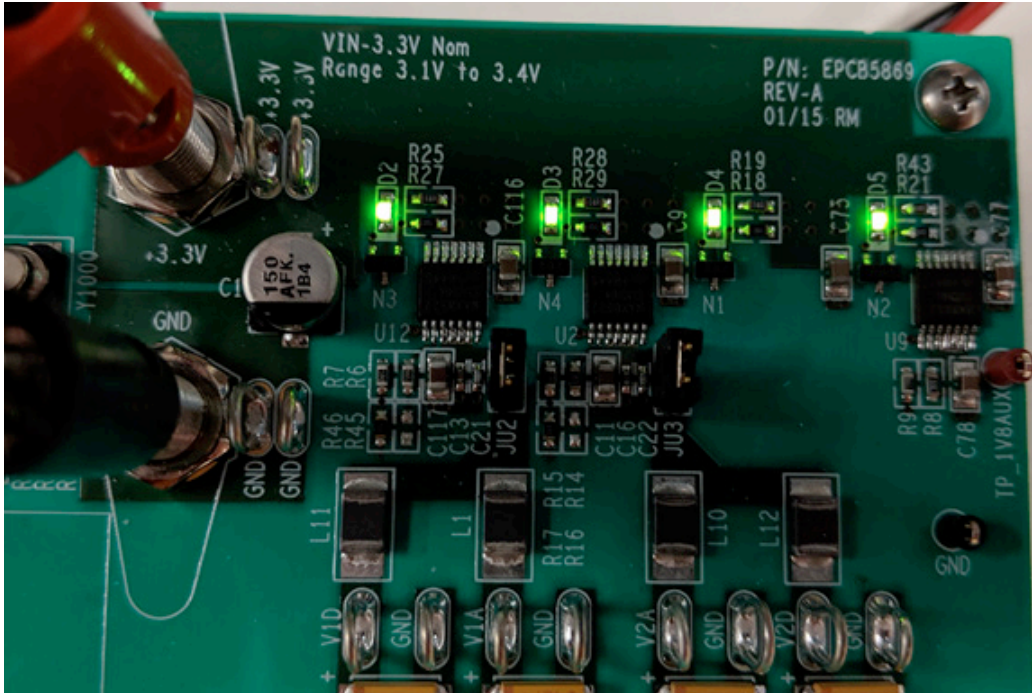


Figure 6. MAX5869EVKIT LEDs – 6a. Power Indicators; 6b. Over Temperature Alarm Indicator (Default is off)

**Temperature Monitoring**

As described in the [Detailed Description of Software–Status tab](#) section, an alarm threshold can be set for the MAX5869 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 MAX5869 EV kit includes a MAX6120 precision reference for use as an external voltage level for the RF DAC. Power for the MAX6120 is supplied through jumper JU4, while JU5 connects the MAX6120 output to the MAX5869 V<sub>REF</sub> input.

**Data Interface**

The MAX5869 EV kit directly connects to the VC707 FPGA board through the HPC-1 FMC connector, providing a high-quality interconnect for the JESD204B serial link, supporting lane rates up to 9.8304Gbps. The MAX5869 REFCLK output is used as the reference frequency for data rate synchronization with the FPGA.

Schematic and layout files for the MAX5869 EV kit board are included with the software installation and can be found in the MAX5869\EVKIT Info folder, later in this document, or at the Maxim website.

**Xilinx VC707 FPGA Evaluation Board**

The Xilinx VC707 board acts as the data source for the MAX5869, allowing for user-defined signal generation. Test patterns, generated externally, are stored in the VC707's on-board DDR memory and subsequently transmitted to the MAX5869. 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 interpolation rates and bus configurations supported by the MAX5869.

The MAX5869 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 MAX5869 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 7a](#)). The GPIO LEDS can be used to identify various states of the FPGA-to-MAX5869 interface. [Table 4](#) describes these states.

**Table 4. VC707 LED Descriptions**

LED	COLOR	STANDARD OPERATION	DESCRIPTION
0	Green	On	PLL at JESD Interface Locked
1	Green	On	JESD SYNC True
2	Green	On	MMCM at MIG Interface Locked
3	Green	On	MMCM at MIG Interface Active
4	Green	On	Temperature Alert Status
5	Green	On	MAX5869 - JESD204 Interrupt Status
6	Green	Off	MAX5869 - JESD204 Mute Status
7	Green	On	MAX5869 - JESD204 Reset Status

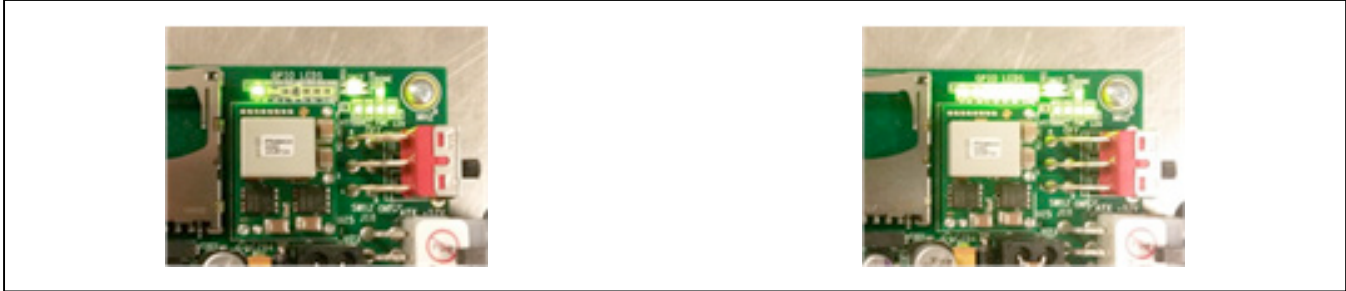


Figure 7. VC707 LEDs – 7a. Before FPGA is Programmed; 7b. After FPGA is Programmed

All jumpers and switches on VC707 should be used in its default configuration for normal operation of the MAX5869EVKIT 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 8a](#). 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 8b](#).

### Programming the EEPROM

Rather than using the GUI to program the FPGA after each power cycle of the VC707, the on-board flash memory can be used to store the default RTL for the MAX5869 data source. 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](#).

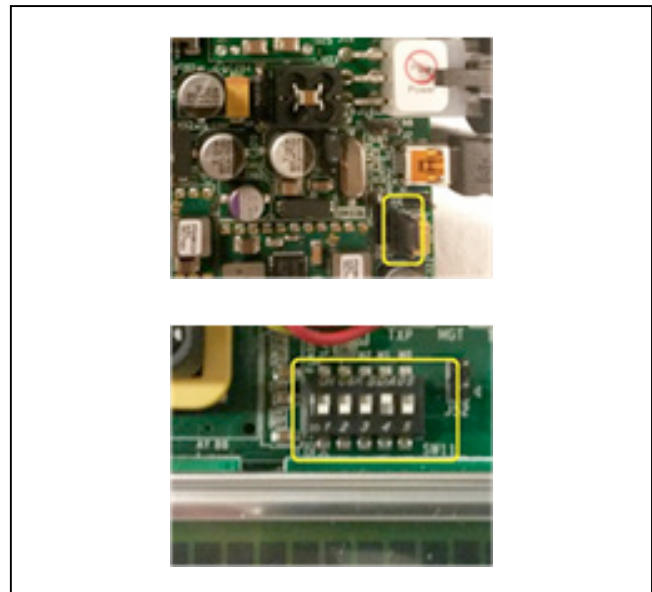


Figure 8. VC707 Jumpers and Switches – 8a. J44; 8b. SW11

### Detailed Description of Software

The MAX5869 EV kit software controller GUI is designed to control the EV kit and the VC707 board, as shown in [Figure 9](#). The MAX5869 EV kit software controller includes USB controls that provide SPI and SMBus communication to

the MAX5869 and the MAX6654 interfaces. The software also controls the VC707 through the Silicon Labs COM port on the VC707 board (UART connection on the board panel). The BULK port (USB2.0 in [Figure 9](#)) is used for the transfer of patterns to the VC707 on board memory.

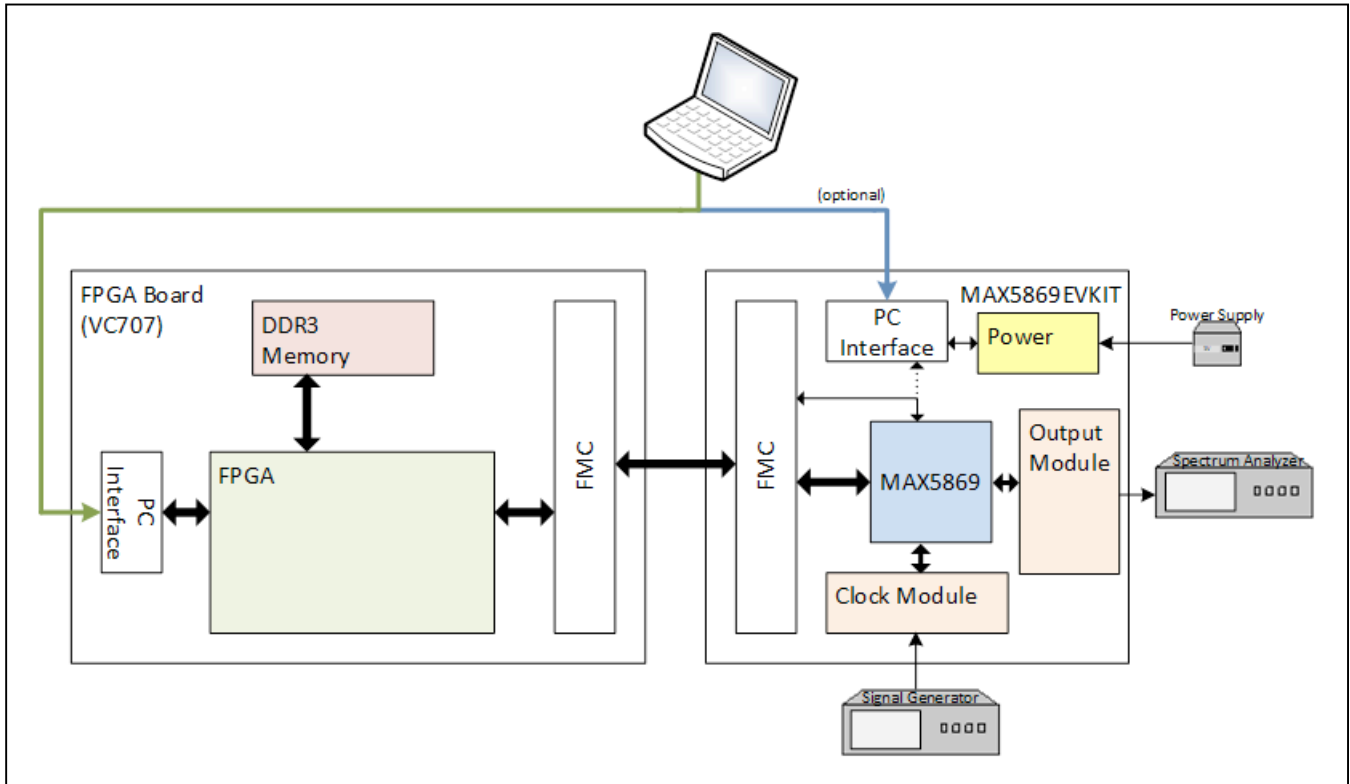


Figure 9. MAX5869 EV System Block Diagram



### MAX5869 EV Kit Software Controller

The EV kit software controller GUI application displays a splash screen on initial startup, as shown in [Figure 10](#). The splash screen is only visible while the application connects to the USB interfaces on the VC707. The main application is displayed after these connections are made. [Figure 11](#) shows the default window opening on the VC707 tab, which occurs when the software does not detect proper communication with the VC707. When proper communication with the VC707 is established during start-up of the GUI (i.e., a previously programmed FPGA), the window will open on the Setup tab as shown in [Figure 12](#). Note that the EV kit connection status is reported in the lower left corner of the GUI. The lower-right corner reports the SPI and I<sup>2</sup>C/SMBus along with the FPGA connections.

The EV kit software controller features five window tabs for configuration and control of the MAX5869 and the VC707. The specific tabs are:

- Setup
  - Load and reload MAX5869 configurations
  - Hardware and software MUTE control
  - Various reset functions
  - Example test configurations
- Clock and NCO
  - Clock configuration
  - NCO configuration
- VC707
  - FPGA programming
  - MicroBlaze command line execution
  - FPGA lane configuration and rate selection
  - Pattern loading and control
- Status
  - Temperature readings and control of the MAX6654 temperature sensor IC
  - Device status reporting
  - Automation support through TCP/IP port
- Register Access
  - User access to read/write MAX5869 configuration and status registers

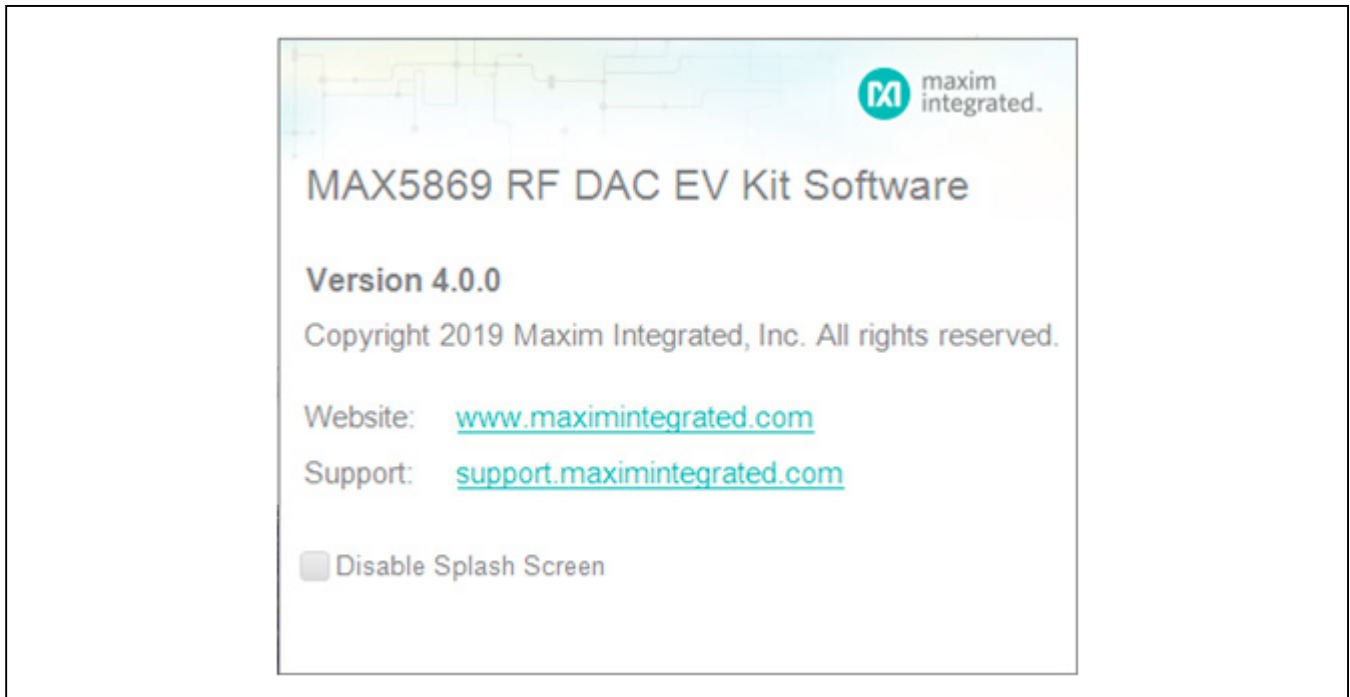


Figure 10. Splash Screen

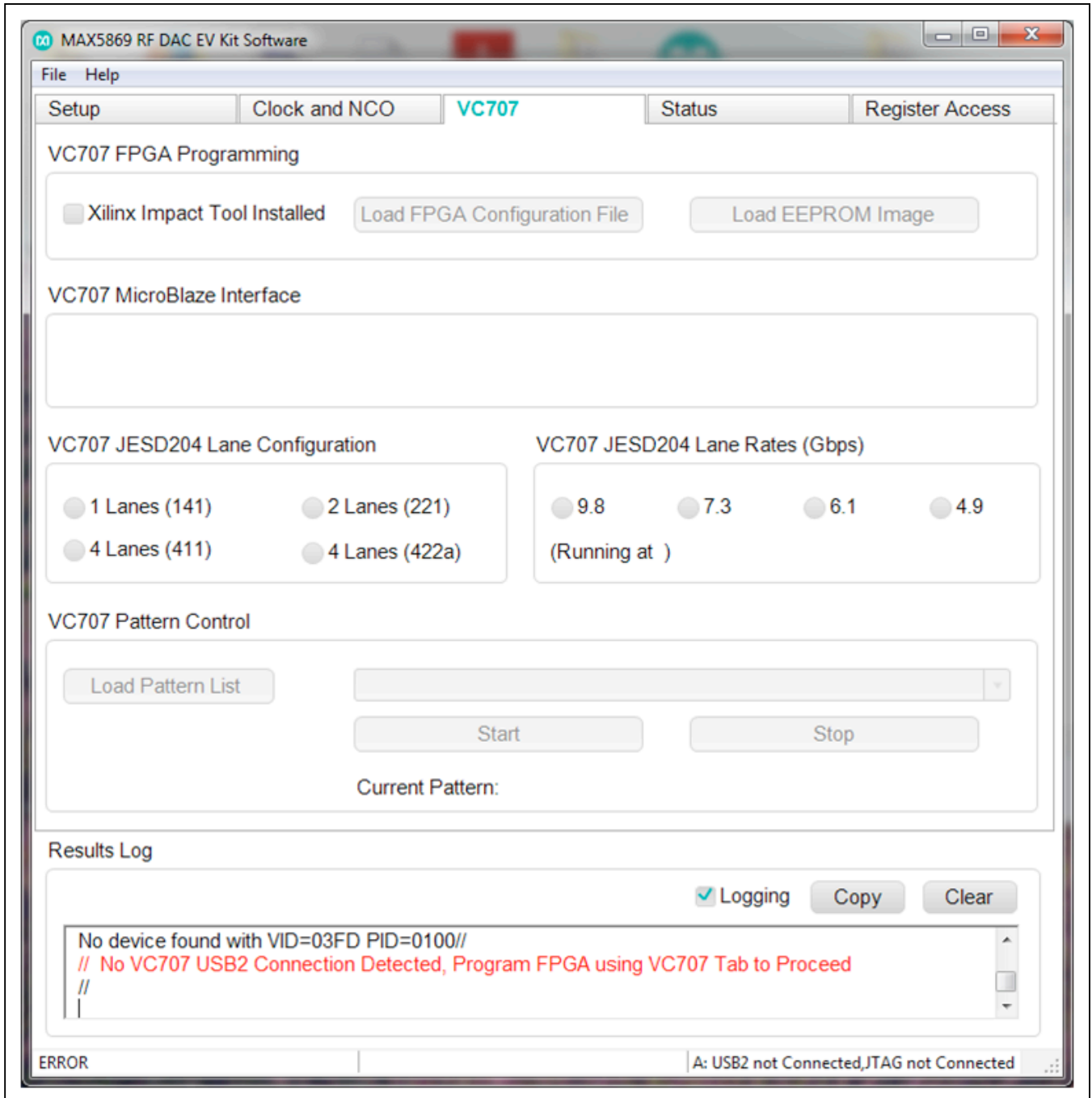


Figure 11. Startup window without FPGA communication

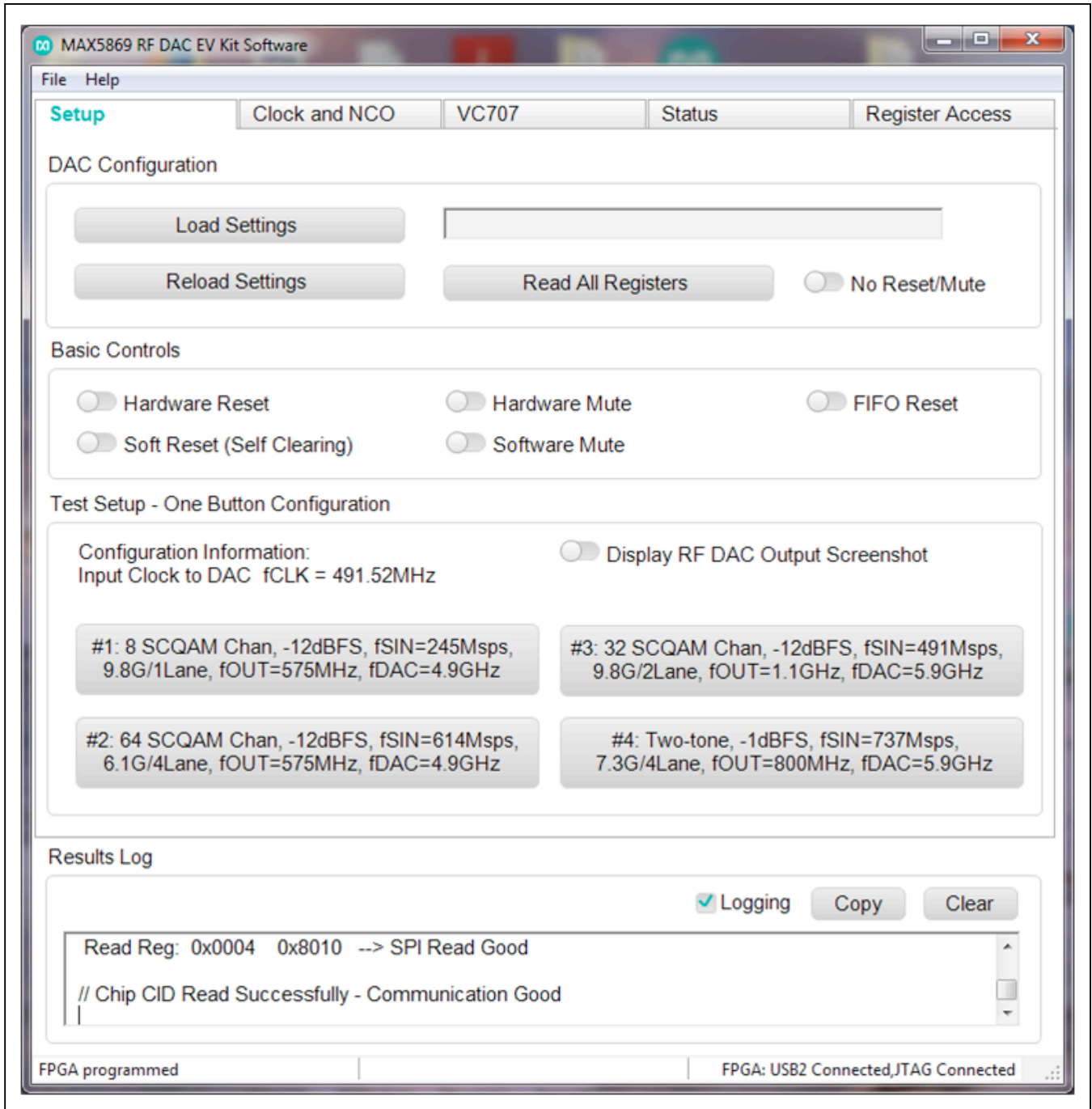


Figure 12. Startup window with FPGA communication

## Setup Tab

The Setup tab (Figure 12) allows the user to load a MAX5869 device configuration file. The configuration contains a specific sequence of SPI register writes required for proper operation of the desired operating mode. Several sample configuration files are included with the software installation and are stored in the 5869/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 of their choice. Clicking the <Open> button causes the software to assert, and then clear, a <HARDWARE RESET> prior to transferring the configuration to the MAX5869.

The various control buttons on the Setup tab include the <HARDWARE MUTE>, which is asserted at startup and controlled through the GPIO, as well as <SOFTWARE MUTE>, which is asserted through register control every time a new configuration is loaded. The <MUTE> controls are used to protect downstream equipment or devices while the device is configured and prior to the generation of valid test signals.

Additional controls include <GLOBAL RESET> and <FIFO RESET>. These controls are NOT normally required; however, the FIFO reset may be required after initially starting a test pattern in order to clear FIFO errors. The lower Test Setup section contains quick settings to configure the EV Kit system to generate example test signals. Four test settings are available including 3 SCQAM examples and a simple two-tone example. A slide switch is also provided to display actual spectrum captured using the active test condition.

## Clock and NCO Tab

The Clock and NCO tab (Figure 13) of the GUI displays the current PLL, RCLK and interpolation settings as defined by the register writes performed when loading the configuration file. The Clock Configuration Status box is populated automatically when the configuration sequence is performed, these settings are cannot be altered directly.

The NCO configuration box contains all the controls required to configure the operation of the NCO used in the digital modulator block of the MAX5869. First, the desired

center frequency for the output signal is entered in the text box labeled Target fNCO. Next, the user needs to determine how they would like the NCO frequency to update. Two options are available for the method of changing NCO frequencies: Immediate or Increment/Decrement.

The Immediate mode provides the fastest switching response time and requires no additional user inputs before applying the calculated values. However, it may create a glitch at the DAC output which may or may not be an issue when lab testing.

The Increment/Decrement option is glitch free and causes the NCO to change frequencies based on the step size programmed by the user. A text entry box will appear when selecting this option, which allows the user to enter the step size to be used when changing the NCO frequency. The NCO will step towards the new value, with one step for every 8 fDAC cycles. A specific example is shown here:

Example:

FCW-old = 0x80000000

FCW-new = 0x80008000

CfgNCOUS = 0x000010

Once the NCO update is issued, it takes  $(0x80008000 - 0x80000000) / 0x000010$  cycles of DAC Clock /8 which is a total of  $2048 \times 8 = 16384$  DAC clocks.

The GUI also reports the PLLs Fractional or Extended mode. Extended mode is enabled with the Extended NCO Enable toggle button. When this mode is enabled, the CfgNCON and CfgNCOD reporting boxes are added below the CfgFNCO text box.

Clicking the <Calculate Values> button will update the CfgFNCO and, when enabled, the extended mode numerator (CfgNCON) and denominator (CfgNCOD) text boxes. The CfgFNCO box which will display an 8 digit HEX value that is to be written to the CfgFNCO register in the MAX5869. The CfgNCON and CfgNCOD boxes will update with a 4-digit HEX value representing the fractional portion of the frequency calculation. The CfgNCON and CfgNCOD are reduced to the lowest possible denominator value. The Apply Values button is clicked in order to transfer the calculated register values to the MAX5869.

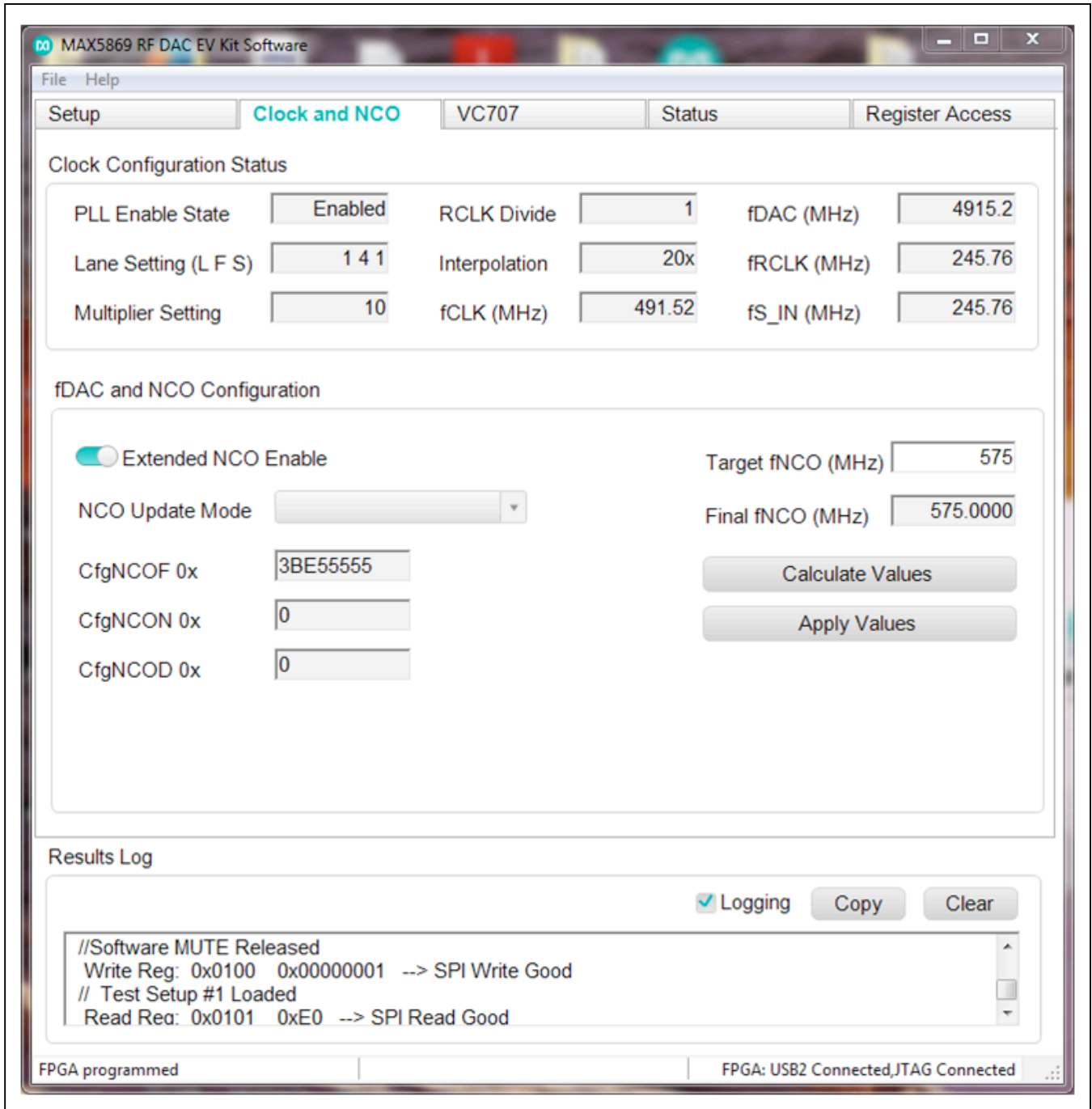


Figure 13. Clock and NCO Tab



**VC707 Tab**

The VC707 tab (Figure 14) provides controls to a) program the FPGA, b) access the Microblaze on the programmed FPGA and c) load, select, and start/stop test patterns. The page also reports the expected operating mode of the JESD204B serial link, showing number of active lanes and the rate of operation. The user must initially point to the

folder where the Impact tool resides, and then the location will be stored on disk for future executions. Clicking the check-box the first time the application is executed causes a file browser to open. The user navigates to the folder containing the Impact tool, by default this is installed in \Xilinx\14.7\LabTools\LabTools\bin\nt64 (for 64-bit PCs) or \Xilinx\14.7\LabTools\LabTools\bin\nt (for 32-bit PCs).

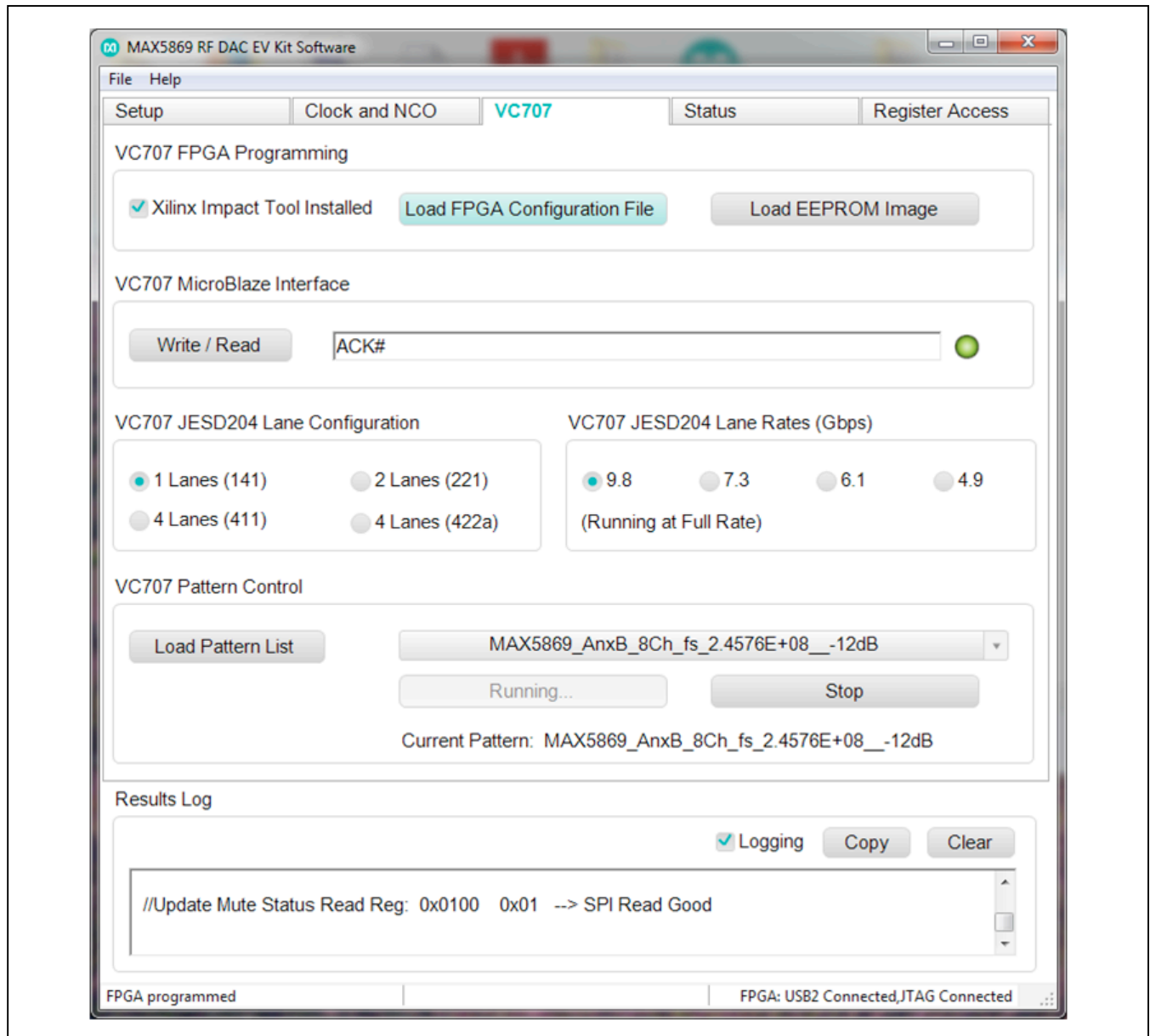


Figure 14. VC707 Tab

Once the Impact path has been captured, the FPGA Configuration (.bit) file can be downloaded. Clicking the Load FPGA Configuration button causes a file browser window to appear with the current folder set to MAX5869\VC707Files. The user selects the MAX5869\_DataSource.bit file and clicks open. The MAX5869 EV kit Software control generates and executes the command line call of the Impact tool to transfer the configuration to the FPGA. The GUI provides a Loading progress bar, which can take up to 60 seconds or more to complete. The fan on the VC707 will stop momentarily during the process. After the programming is completed, the fan will restart and the cycling LEDs on the VC707 will become static. If the progress completes within a few seconds, the fan never stops and the LEDs continue to cycle, then the firmware did not load properly. Ensure that the drive\_install.exe file, located in the same folder as the Impact tool, has been executed and try loading the firmware again.

The VC707 is ready to act as the data source for the MAX5869 once the firmware is load has completed. Additionally, the MicroBlaze Interface can be employed for direct communication with the FPGA. Clicking the Text Box allows the user type in a command that will either be a read or write command. Results of the command appear in the LOG window, and when successful the text box will update with ACK – Acknowledge. NAK is reported when the command fails, usually due to improper syntax. Writing a <PING> command to the MicroBlaze should at this point return and ACK, a useful step to verify configuration has completed.

The VC707 JESD204B Lane Configuration and the VC707 JESD204B Lane Rates are set when the device configuration file is loaded. These radio buttons cannot be directly altered, they are only provided to indicate the programmed operating mode for the serial link.

The VC707 requires that the test pattern(s) be stored in the on-board memory, which occurs when the <Load Patterns> button is clicked. The GUI opens a file browser

in the MAX5869\PatternFiles folder. The user then selects the desired list (patListFile.txt is included at installation for an example). The list of patterns is then read into the PC and transferred to the VC707 through the BULK USB connection. Multiple patterns can be loaded simultaneously with the total combined pattern size of up to 1GB.

Now that the FPGA is configured and patterns have been loaded, the user selects the desired pattern in the drop-down box. Clicking <Start> causes the JESD204B link in the FPGA to reset and, after synchronization with the MAX5869, the selected pattern is output to the JESD204B interface.

The <Stop> button must be clicked prior to selecting another pattern or changing the MAX5869 configuration. Only the NCO frequency can be altered on the fly allowing the user to dynamically move the center frequency of the output signal.

### Status tab

The Status tab, [Figure 15](#), provides the user interface to the MAX6654 Temperature Monitor IC, a device Status Report from the MAX5869, and the ability to enable control of the MAX5869EVKIT Software Controller through a TCP/IP port.

The Temperature section of this tab provides for controls for user access of the MAX6654 devices internal registers; Read Temperature, Read Sensor Register, Set Threshold and Clear Temperature Alert. Activating the <Read Temperature> button causes the MAX6654 to read the die temperature (TDA/TDC connections) and return the value. The GUI interprets the returned value, converting it to a Celsius value that is displayed in the adjacent text box.

Reading the sensor registers requires the user to enter the desired register to access in the text box. Activating the <Read Sensor Register> button updates the value text box with the current contents of the address specified; the value is reported in HEX format.

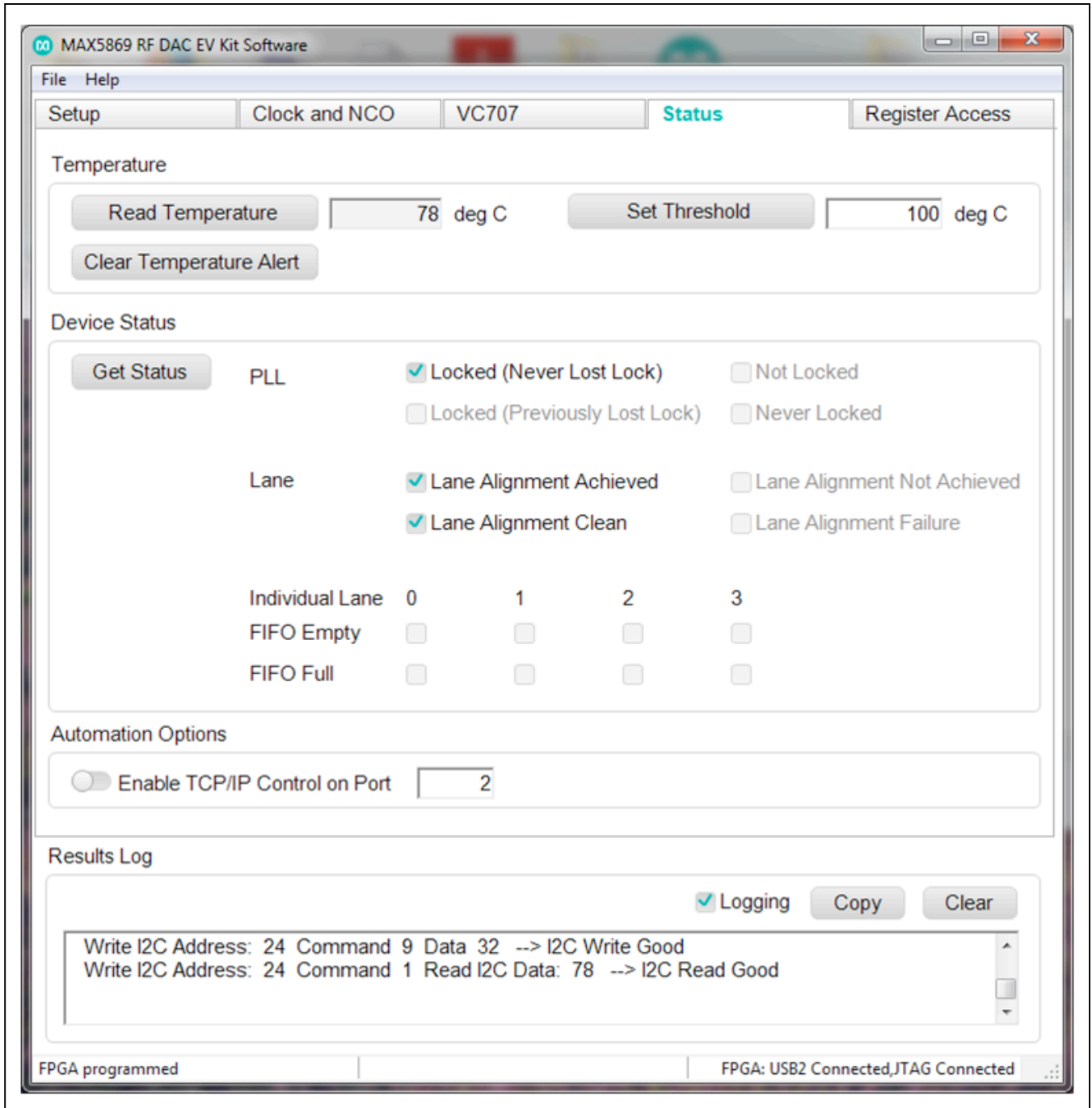


Figure 15. Status Tab

The <Set Threshold> button is used to configure the high temperature ALERT threshold for the MAX6654. The ALERT signal will be asserted when the temperature of the MAX5869 exceeds this value. The Celsius value entered in the adjacent text box will be converted into the appropriate register write for this function. When ALERT is asserted, a red LED (D1) will illuminate on the MAX5869EVKIT PCB. Activating the <Clear Temperature Alert> button will clear the alert. However, if the MAX5869 die temperature is still above the programmed threshold, the LED will immediately re-illuminate.

The Device Status section reports the status of the PLL and the serial lanes of the JESD204B link. The values are updated after each pattern start (JESD link synchronization) and whenever the user clicks the Get Status button.

The Automation Options section provides the user the control needed to enable remote operation through a locally opened TCP/IP port. The user enters a desired port number and then clicks the <Enable TCP/IP Control> toggle button. NOTE, the application window will not respond to direct control and will not update until it receives an 'RTL' command from the TCP/IP port. Refer to the MAX5869EVKIT Automation Support rev#.pdf document located in the c:\MaximIntegrated\MAX5869\EVKIT Info folder that was created during the software installation process.

**Register Access Tab**

The <Register Access> tab, [Figure 16](#), allows the user to explore and modify the register contents of the MAX5869. There are several hundred registers in the device, which are broken up into operational groups. The controls on this tab provide an intuitive, easily operated method for performing read and write operations as well as provide some detail on the specific functions of specific registers.

The tab is broken up into 5 sections; 1) Register Block and Register selection, 2) Command Selection, 3) Selection Options, 4) Execute button and 5) Results returned from command execution. Valid options either populate or enable based on previously selections and entries.

**Custom Configurations**

Custom configurations of the MAX5869 can be readily generated using the PERL script included with the software installation. The PERL script MAX5869\_gen\_config.pl is located in the c:\MaximIntegrated\MAX5869\DeviceScripts\PERL\_ScriptGen folder that was created during the software installation process. The user must have a PERL interpreter, such as the free application StrawberryPERL (<http://strawberryperl.com/>), installed on their system in order to execute the supplied script.

The PERL script reads a .setup file and creates the .cfg files containing the sequence of register writes required to configure the MAX5869 device for the intended operating mode. In order to create a new configuration file the user must identify the operating parameters for their application by completing the table below:

f <sub>S_IN</sub> :	_____
Link Rate:	_____
Lane Count:	_____
RCLK Divider:	_____
PLL Mode:	_____
fCLK:	_____
Interpolation Rate:	_____
SYSREF Mode:	_____
JESD204B Subclass:	_____
Device Clock source:	_____

First, determine the required baseband input sample rate (f<sub>S\_IN</sub>) based on the bandwidth of the signal to be generated: 737.28MHz, 614.4MHz, 491.52MHz, 368.64MHz, 307.2MHz, or 245.76MHz. The input sample rate determines the available options for the Link Rate and Lane Count, as illustrated in the LaneConfigOptions.pdf file located in the c:\MaximIntegrated\MAX5869\DeviceScripts folder. As an example, a f<sub>S\_IN</sub> of 737.28Msps requires four lanes at 7.3728Gbps and the RCLK Divider set to 4, while a f<sub>S\_IN</sub> of 245.76Msps has three possible lane count and rate conditions: four lanes at 2.4576Gbps, two lanes at 4.9152Gbps, or a single lane at 9.8304Gbps all with RCLK Divider setting of 1.

Once f<sub>S\_IN</sub> and the JESD204B configuration are determined, the DAC update rate, input clock frequency and interpolation rate can all be determined. Referring to the Valid\_Configurations.pdf file, also located in the c:\MaximIntegrated\MAX5869\DeviceScripts folder, the user locates the desired combination of these parameters that match the selected f<sub>S\_IN</sub>.

The remaining parameters depend on the system needs of the user. SYSREF Mode can be set to use either a continuous (0), clock-like signal or a single pulse (1). The MAX5869 supports JESD204B Subclass-0 and Subclass-1 (required for deterministic latency). The Device Clock Source is a setting in the MAX5869 that determines if the DSP section of the device (interpolators, complex modulator and NCO) uses a clock derived from fDAC or recovered from the JESD204B Link.

Now that all the required parameters are defined, they are entered into a .setup file. The user can open an existing .setup file in a text editor for modification. It is recommended that the file be saved with a new name that identifies the

properties of the entered configuration, as the name of the setup file will determine the name of the .cfg file generated by executing the PERL script.

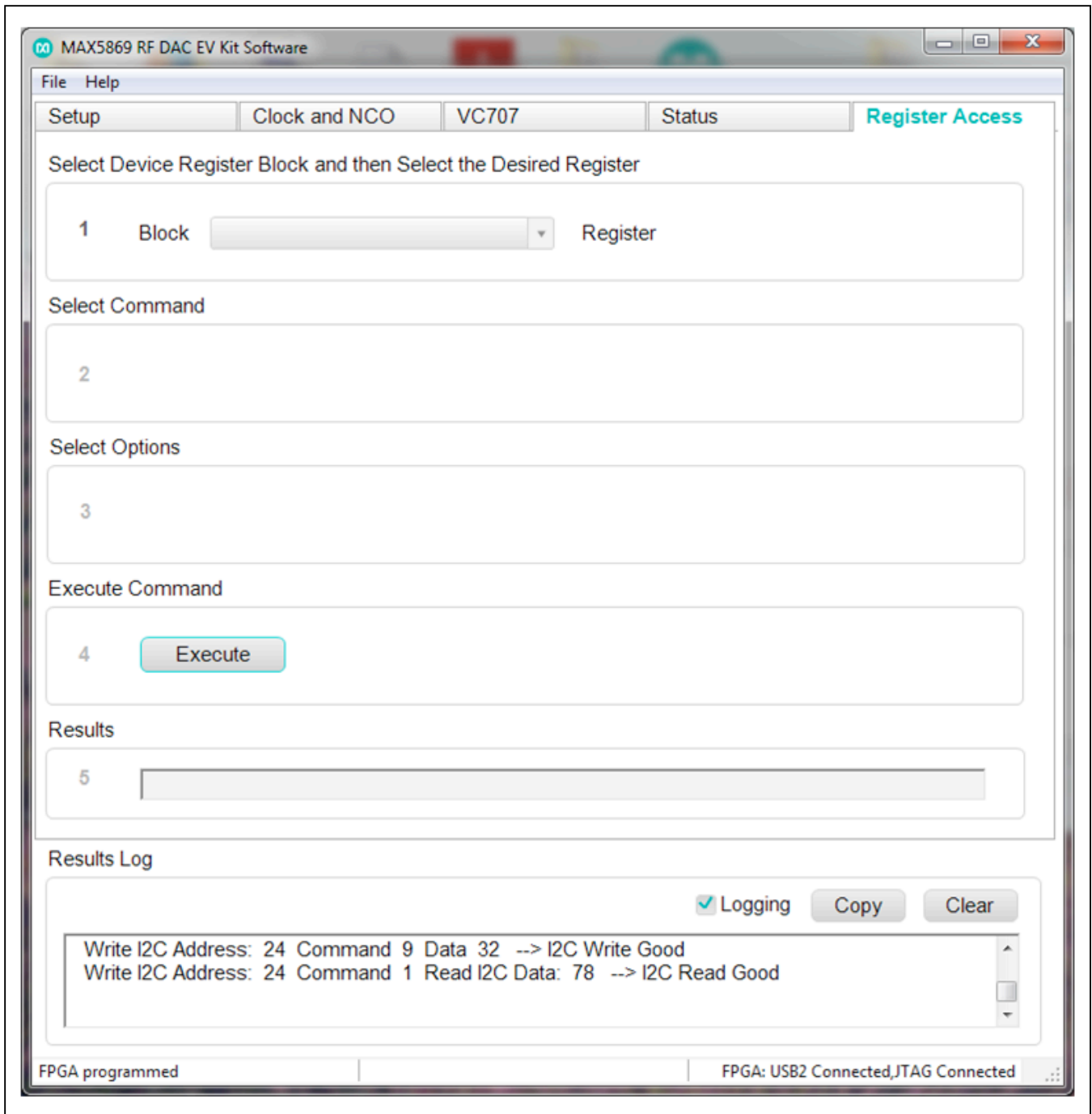


Figure 16. Register Access Tab



## Test Pattern Lists and Files

### Pattern List Files

The MAX5869 EV kit software controller utilizes list files for loading test patterns into the VC707 memory, such as the example patListFile.txt located in the MAX5869\PatternFiles folder. The list file simply contains the names, including extension, of the test pattern files. The format is simple text with one 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 1GB 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 total number of I/Q data points within a given pattern must be an integer multiple of 1024.

### Pattern Generation

The MAX5869 EV kit software controller is provided with a limited number of sample patterns, one for each of the supported input sample rates ( $f_{S\_IN}$ ). 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.

A Matlab folder is included in the MAX5869\PatternFiles folder, which contains sample Matlab routines (.m files) that allow a user with Matlab installed to create additional continuous wave, sinuous test patterns. The makesignal.m routine will performs the CW generation and outputs a complex vector with a range of  $\pm 1.0$ . The set\_datascale\_avg.m is used to convert the complex vector into integer representations of the offset binary values for the I and Q data paths. Finally, the scaled pattern is stored to a pattern file using the MAX5869\_PatFile.m routine which formats the pattern appropriately for use with the VC707. Please refer to the comments,

lines that begin with % following the Copy Right statement at the top of the file, in these files for information about the input arguments used by these routines.

The pattern file can use any extension, such as csv, txt, as long as the contents are simple text and conform to the format expected by the MAX5869 EVKIT Software Controller. The specific format is as follows:

The first line contains 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 and the Q data MS BYTE. That is: I LS BYTE, I MS BYTE, Q LS BYTE, Q MS BYTE

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

LINE NUMBER	CONTENTS
1	65536
2	19, 242, 253, 127
3	19, 242, 250, 127
4	17, 242, 248, 127
5	15, 242, 245, 127
...	
...	
...	
65534	13, 242, 242, 127
65535	10, 242, 239, 127
65536	7, 242, 236, 127
65537 (N+1)	3, 242, 234, 127

*N is the total number of I/Q data points in the file.*

## Component Suppliers

SUPPLIER	WEBSITE
Fairchild Semiconductor	www.fairchildsemi.com
Hong Kong X'tals Ltd.	www.hongkongcrystal.com
Murata Electronics	www.murata.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 MAX5869 when contacting these component suppliers.

**MAX5869 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-C83,C94,C96,C174,C181	31	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,C75,C77-C79,C116,C117,C172	16	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
C12	1	CAPACITOR; SMT; 0402; CERAMIC; 4700pF; 50V; 5%; X7R; -55degC to +125degC; 0 +/-15% degC MAX.
C34,C40,C41,C43,C44,C1002-C1010	14	CAPACITOR; SMT (0402); CERAMIC CHIP; 0.1UF; 10V; TOL=10%; MODEL=GRM SERIES; TG=-55 DEGC TO +125 DEGC; TC=X7R

PART	QTY	DESCRIPTION
C52	1	CAPACITOR; SMT (0402); CERAMIC CHIP; 2200PF; 50V; TOL=10%; TG=-55 DEGC TO +125 DEGC; TC=X7R
C66	1	CAPACITOR; SMT (0204); CERAMIC CHIP; 0.22UF; 4V; TOL=20%; MODEL=M SERIES; TG=-55 DEGC TO +105 DEGC; TC=X6S
C1000,C1001	2	CAPACITOR; SMT (3528); TANTALUM CHIP; 4.7UF; 16V; TOL=20%; MODEL=TPS SERIES
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
GND1-GND5,GND10,REF-GND	7	TEST POINT; BLACK; PHOSPHOR BRONZE WIRE SILVER PLATE FINISH
H3	1	CONNECTOR; MALE; THROUGH HOLE; BREAKAWAY; STRAIGHT; 12PINS
H6,H7	2	CONNECTOR; MALE; THROUGH HOLE; BREAKAWAY; STRAIGHT; 9PINS

## MAX5869 EV Kit Bill of Materials (continued)

PART	QTY	DESCRIPTION
J5	1	CONNECTOR; MALE; SMT; HIGH SPEED/HIGH DENSITY OPEN PIN FIELD TERMINAL ARRAY; STRAIGHT; 400PINS
JU1-JU5	5	CONNECTOR; MALE; THROUGH HOLE; BREAKAWAY; STRAIGHT; 2PINS; -65 DEGC TO +125 DEGC
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)
R1,R19,R25,R26,R28,R43	6	RESISTOR; 0603; 499 OHM; 1%; 100PPM; 0.10W; THICK FILM
R6,R7,R9,R15	4	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; 649 OHM; 0.1%; 25PPM; 0.063W; THICK FILM
R13	1	RESISTOR; 0603; 25.5K OHM; 1%; 100PPM; 0.10W; THICK FILM
R17,R33,R46	3	RESISTOR; 0603; 0 OHM; 5%; JUMPER; 0.10W; THICK FILM

PART	QTY	DESCRIPTION
R18,R21-R23,R27,R29,R34-R36,R38,R41,R44,R1003-R1005	15	RESISTOR; 0603; 10K OHM; 5%; 200PPM; 0.10W; THICK FILM
R20	1	RESISTOR; 0603; 1.27K OHM; 0.1%; 25PPM; 0.10W; THIN FILM
R24	1	RESISTOR; 0603; 47 OHM; 5%; 200PPM; 0.10W; THICK FILM
R32,R37	2	RESISTOR; 0402; 0 OHM; 5%; JUMPER; 0.063W; 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
TP1,TP3,REFIO,USB w3V3,TP_1V8AUX	5	TEST POINT; RED; PHOSPHOR BRONZE WIRE SILVER PLATE FINISH;
SU1-SU15	15	TEST POINT; JUMPER; BLACK; NSULATION=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
U2,U9,U12	3	IC; VREG; 0.2V DROPOUT LDO REGULATOR; TSSOP14-EP

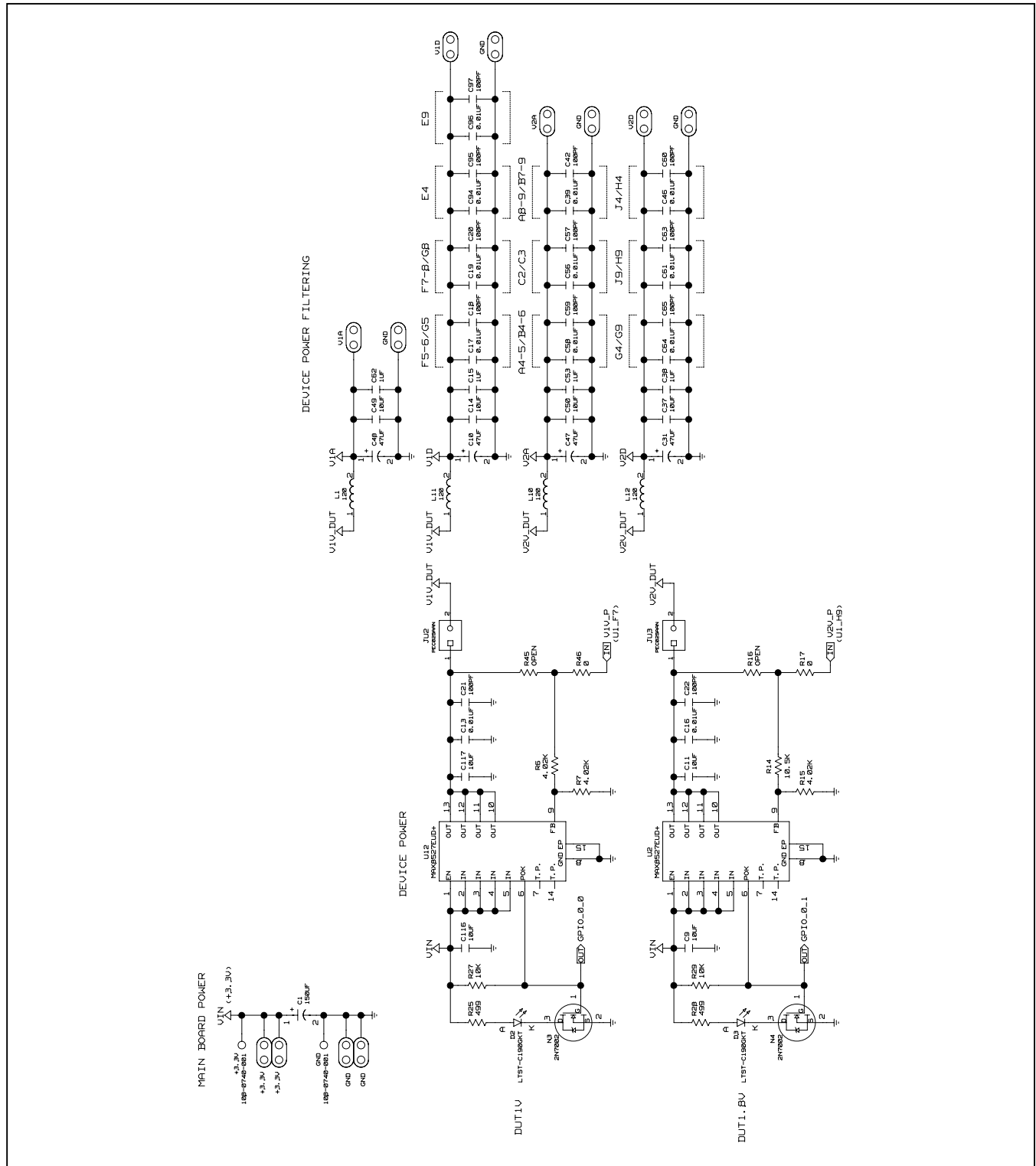
## MAX5869 EV Kit Bill of Materials (continued)

PART	QTY	DESCRIPTION
U3,U6	2	IC; TXRX; 4-BIT DUAL-SUPPLY BUS TRANSCEIVER WITH CONFIGURABLE VOLTAGE TRANSLATION AND 3-STATE OUTPUT; TSSOP16
U4	1	VOLTAGE REFERENCE
U5	1	IC; SNSR; ACCURATE TEMPERATURE SENSOR WITH SMBUS SERIAL INTERFACE; QSOP16
U7	1	IC; VSUP; LOW-POWER TRIPLE-VOLTAGE $\mu$ P SUPERVISORY CIRCUIT; SC70-5
U10	1	IC; VREG; ULTRA-LOW-NOISE, HIGH PSRR, LOW-DROPOUT, LINEAR REGULATOR; SC70-5 ; -40 DEGC TO +85 DEGC

PART	QTY	DESCRIPTION
U11	1	EV KIT MODULE, XFMR_CLK_MODULE
U13	1	EV KIT MODULE, XFMR_OUT_MODULE
U1000	1	IC; USB; QUAD HIGH SPEED USB TO MULTIPURPOSE UART/MPSSSE 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
Printed Circuit Board	1	PCB: EPCB5869

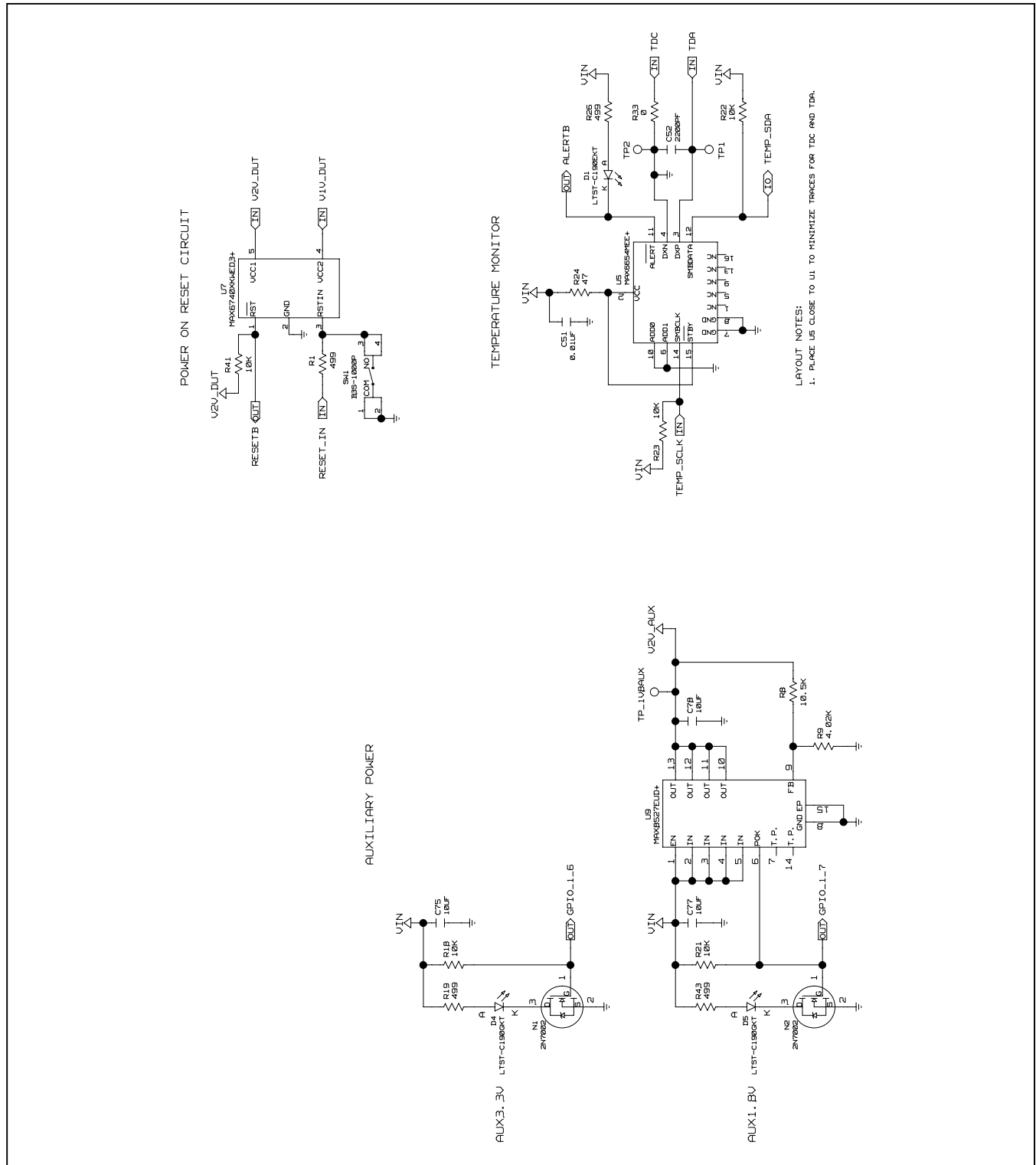


MAX5869 EV Kit Schematic (continued)



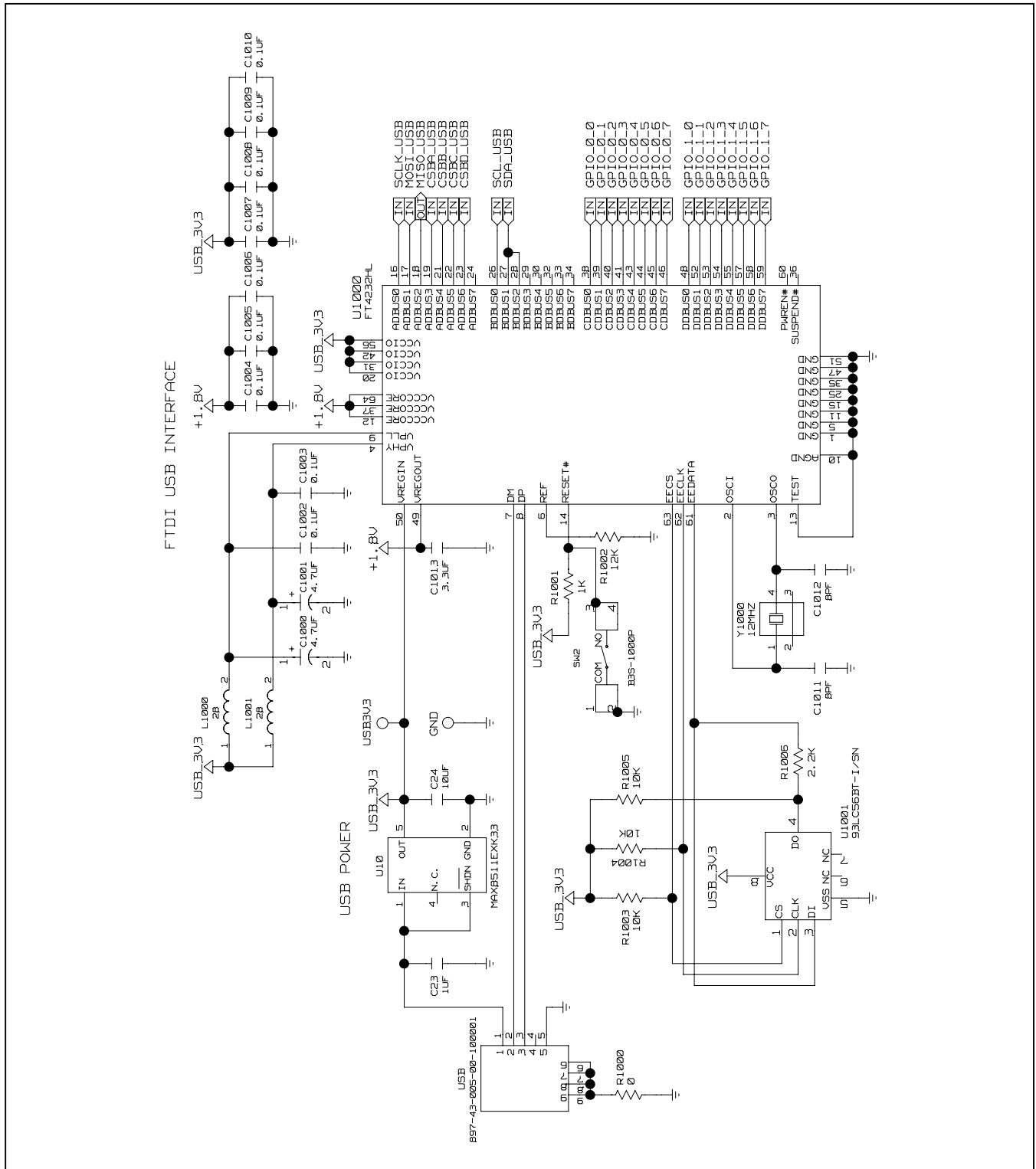


MAX5869 EV Kit Schematic (continued)

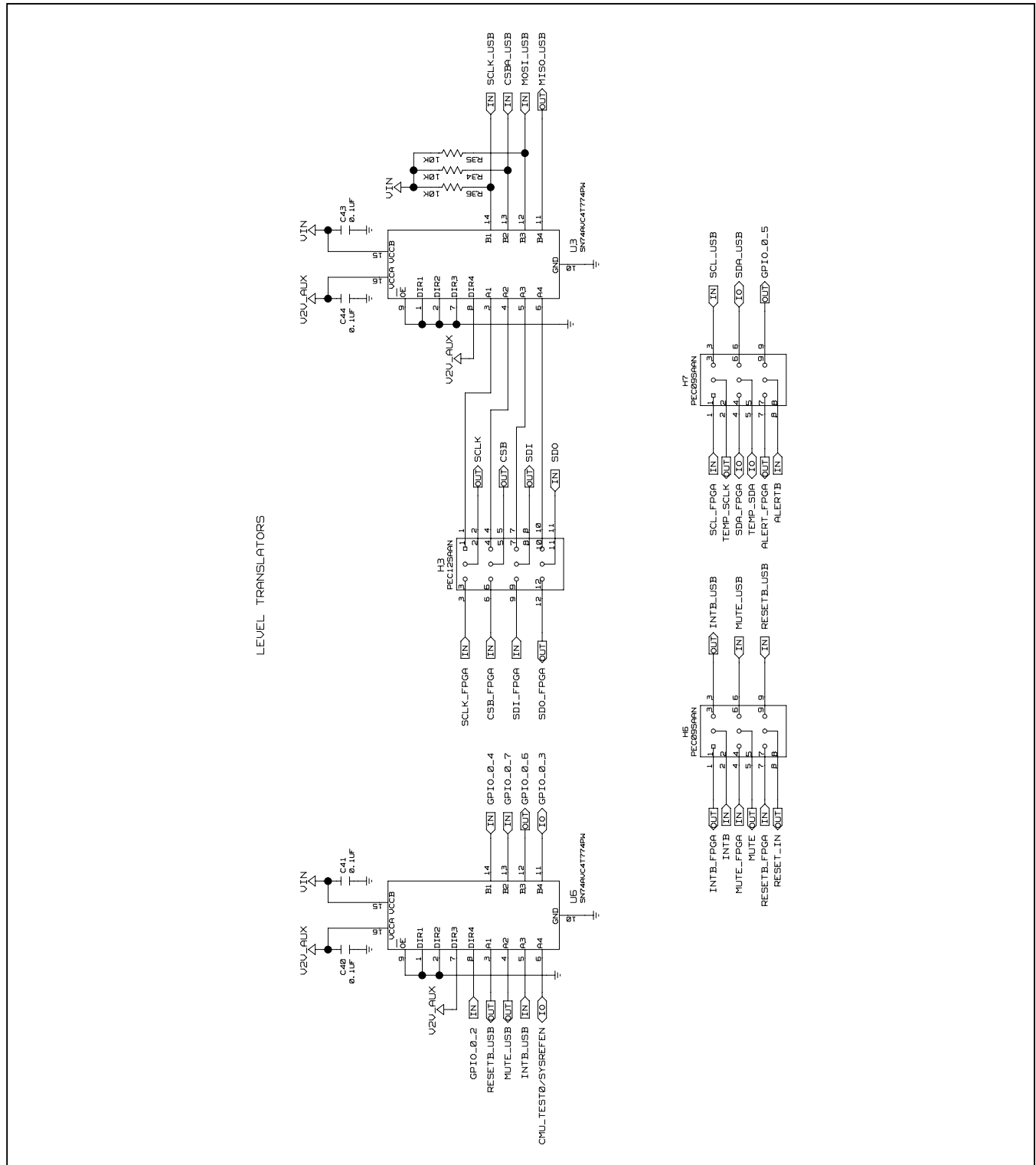




MAX5869 EV Kit Schematic (continued)

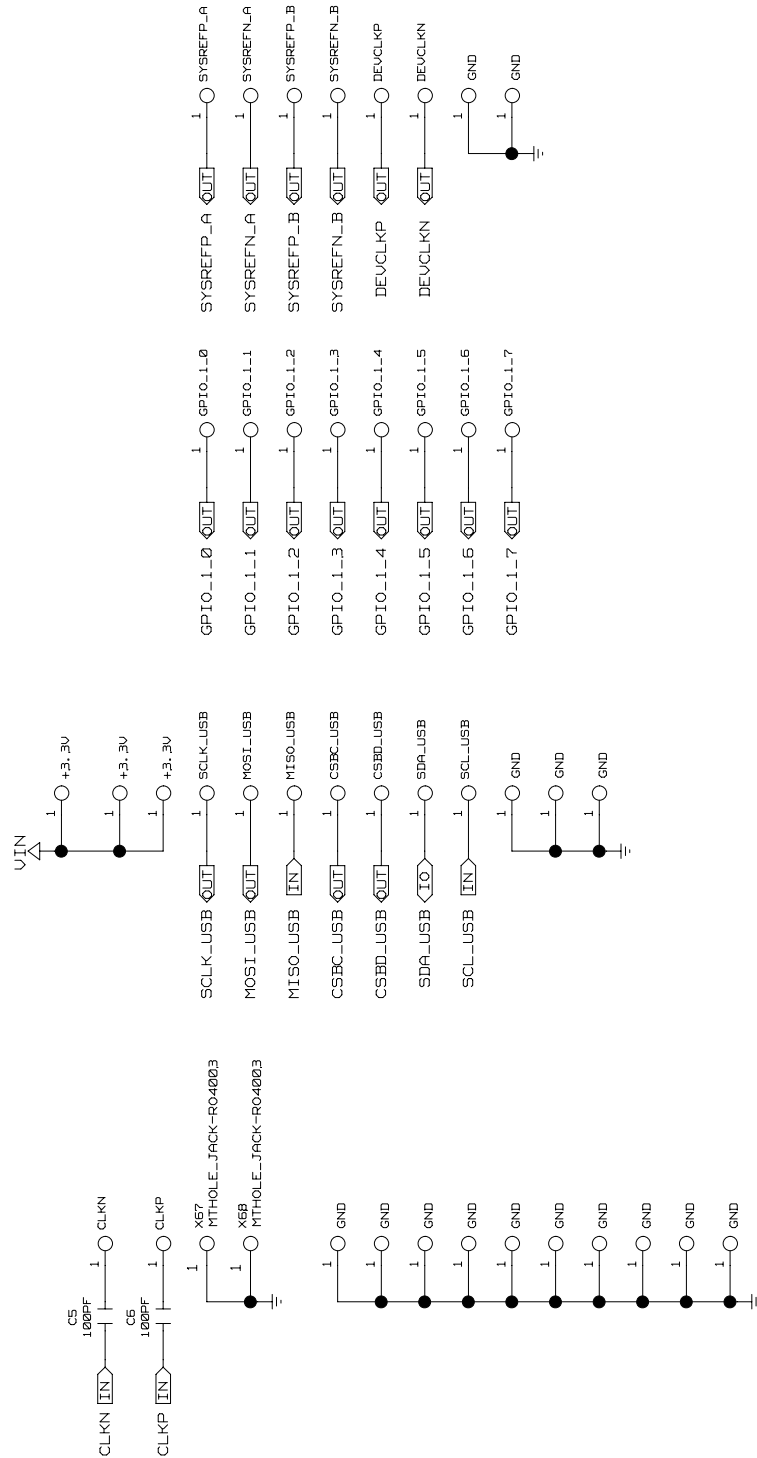


MAX5869 EV Kit Schematic (continued)

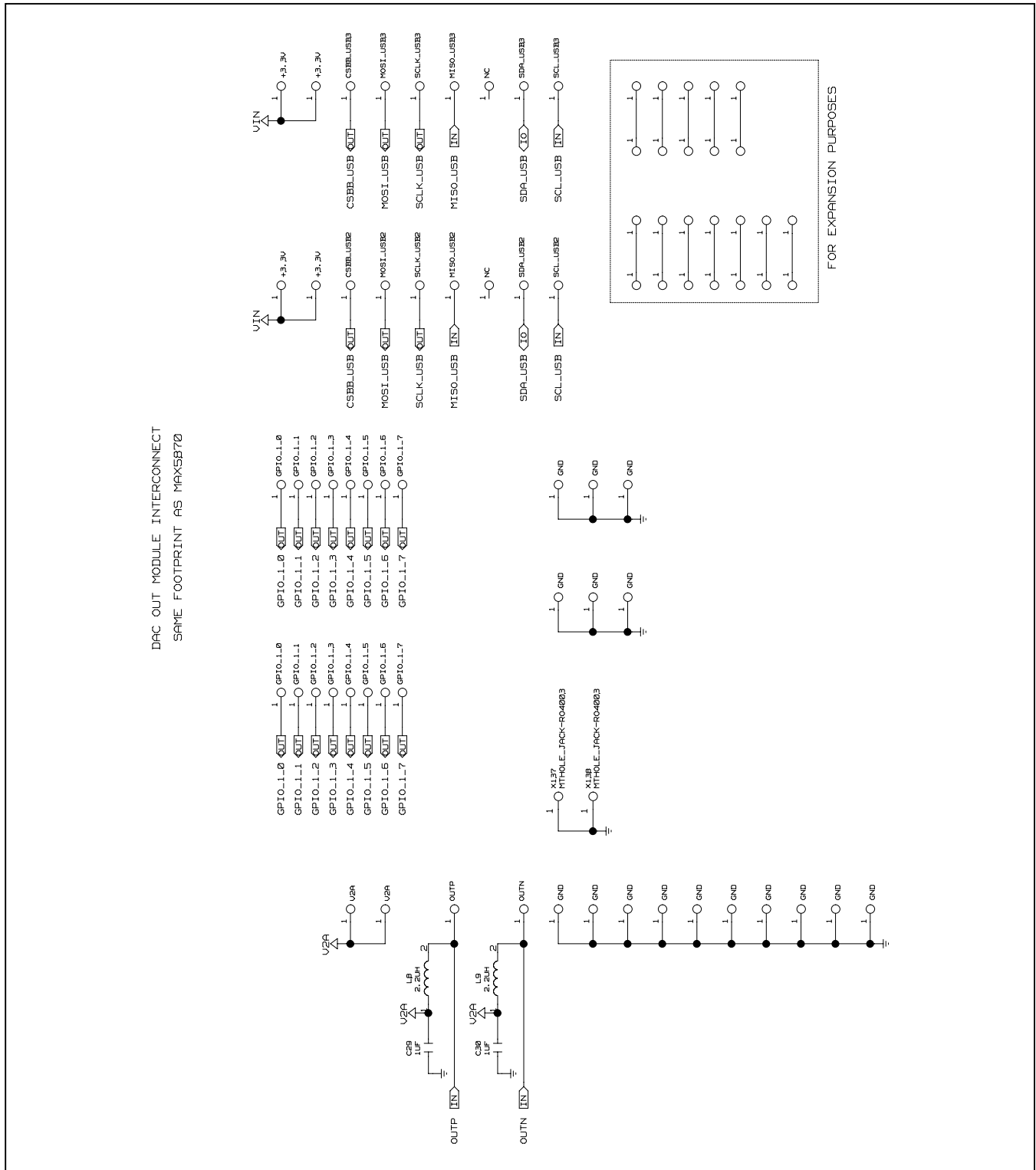


MAX5869 EV Kit Schematic (continued)

CLK MODULE INTERCONNECT  
MATCH EXISTING MODULE FOOTPRINTS, PLUS NEW ACTIVE MODULE

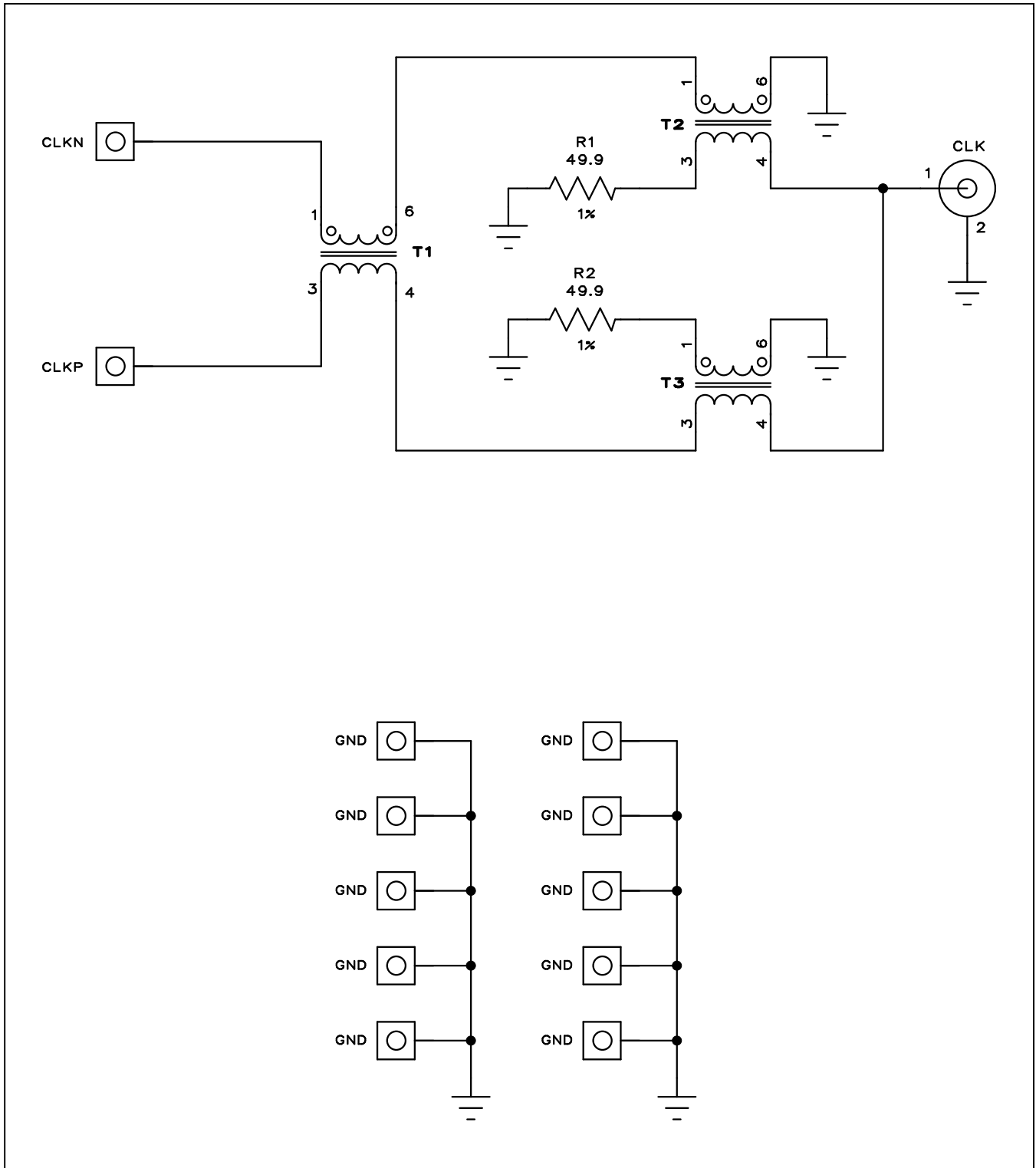


MAX5869 EV Kit Schematic (continued)

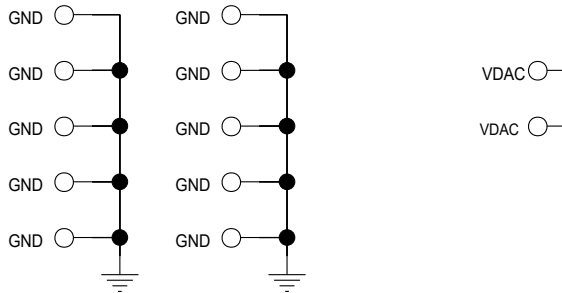
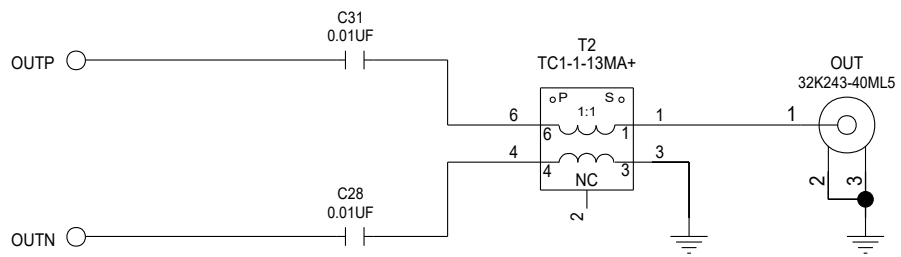




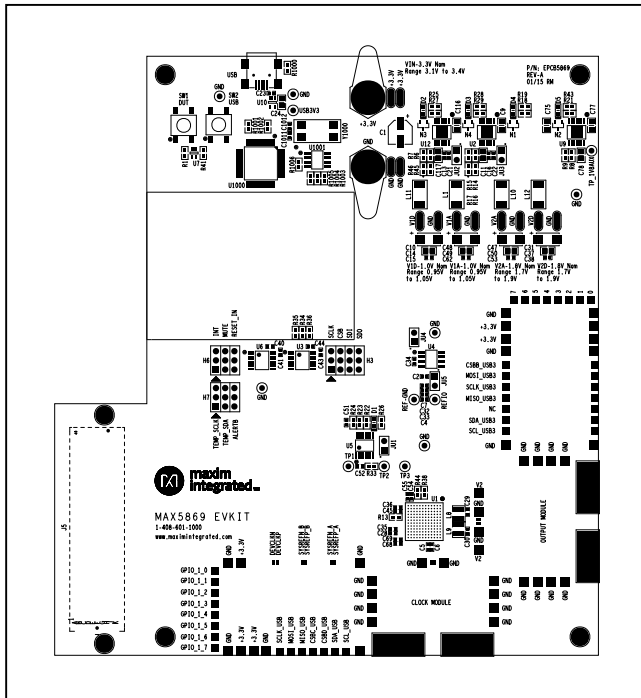
MAX5869 EV Kit Schematic—Clock Module



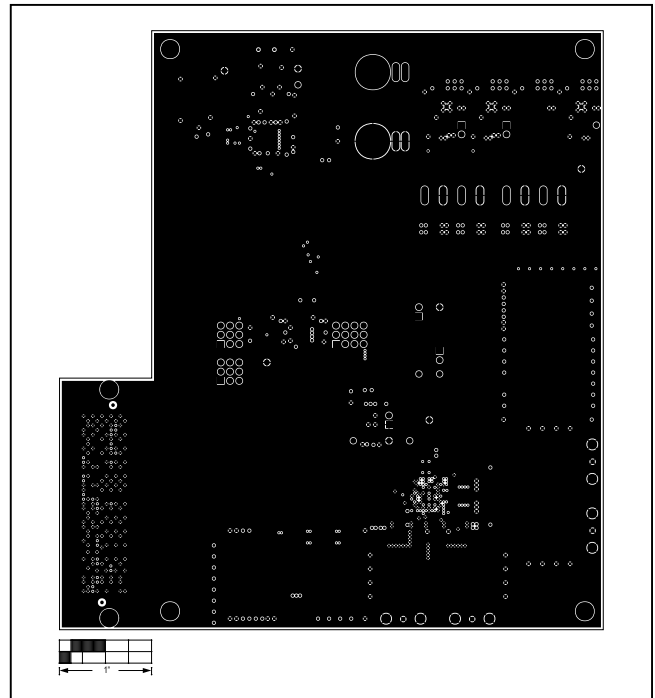
MAX5869 EV Kit Schematic—Output Module Schematic



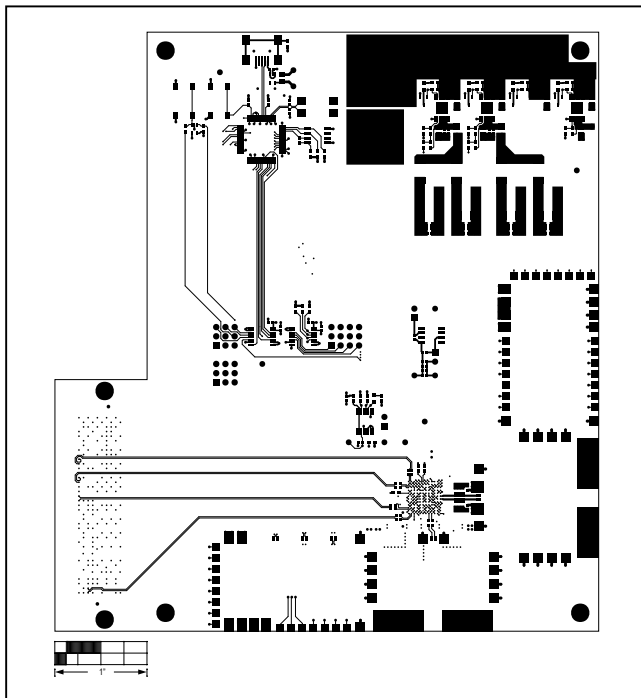
MAX5869 EV Kit PCB Layout Diagrams



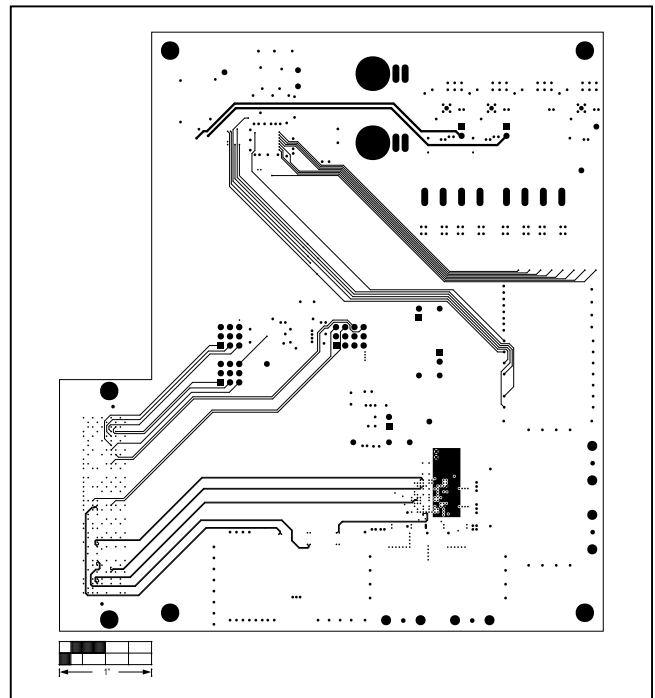
MAX5869 EV Kit—Top Silkscreen



MAX5869 EV Kit—Layer 2

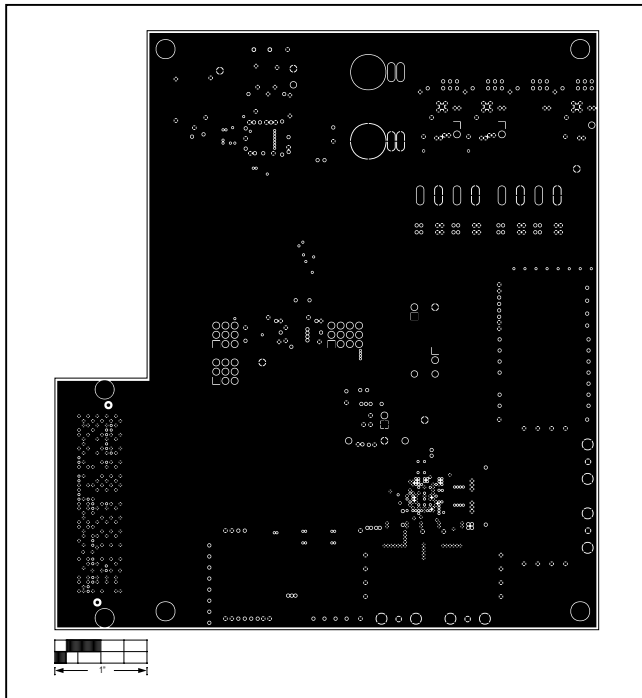


MAX5869 EV Kit—Top Copper

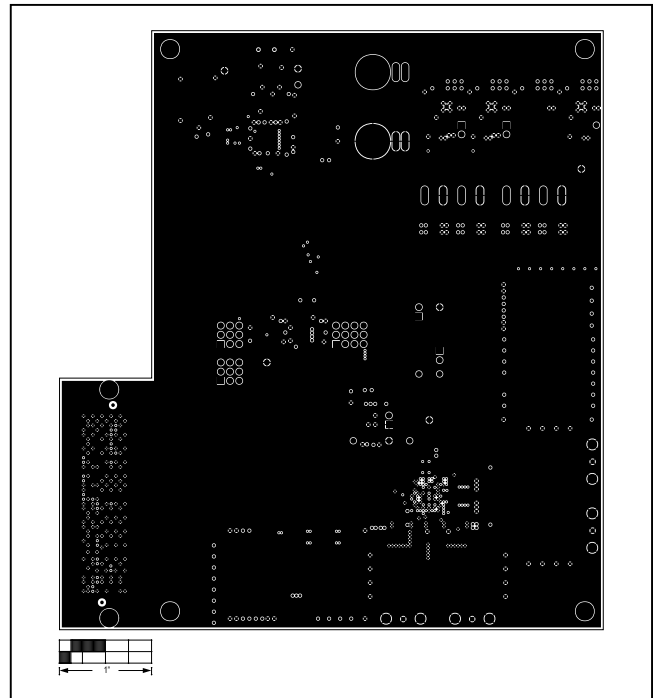


MAX5869 EV Kit—Layer 3

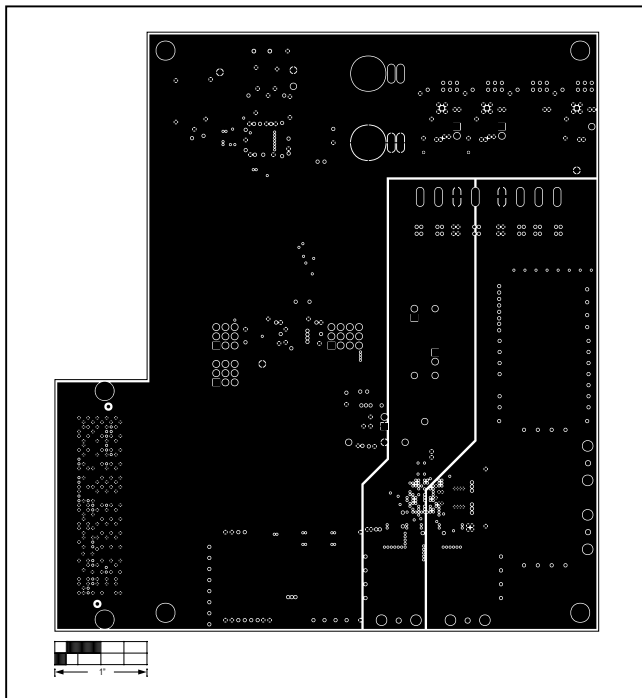
MAX5869 EV Kit PCB Layout Diagrams (continued)



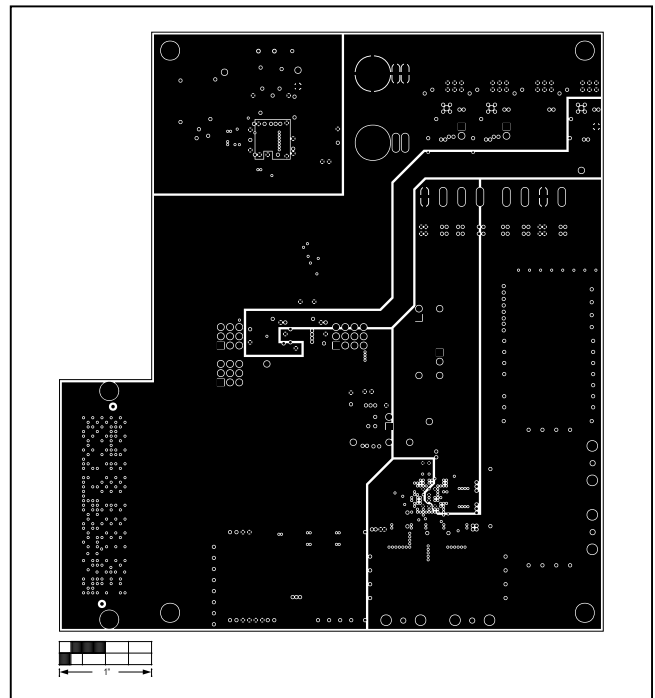
MAX5869 EV Kit—Layer 4



MAX5869 EV Kit—Layer 6

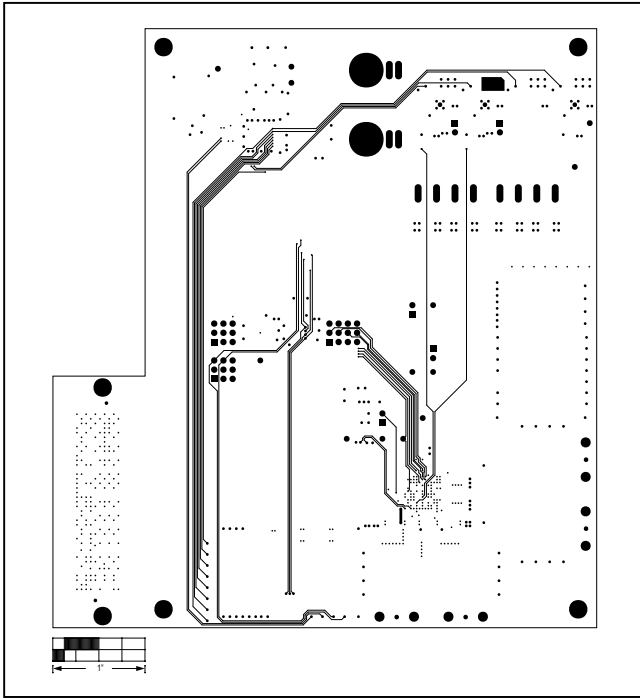


MAX5869 EV Kit—Layer 5

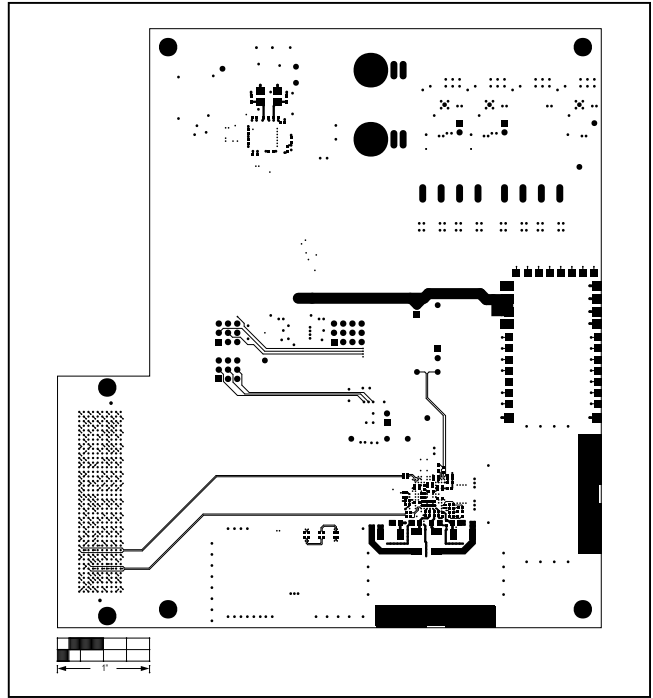


MAX5869 EV Kit—Layer 7

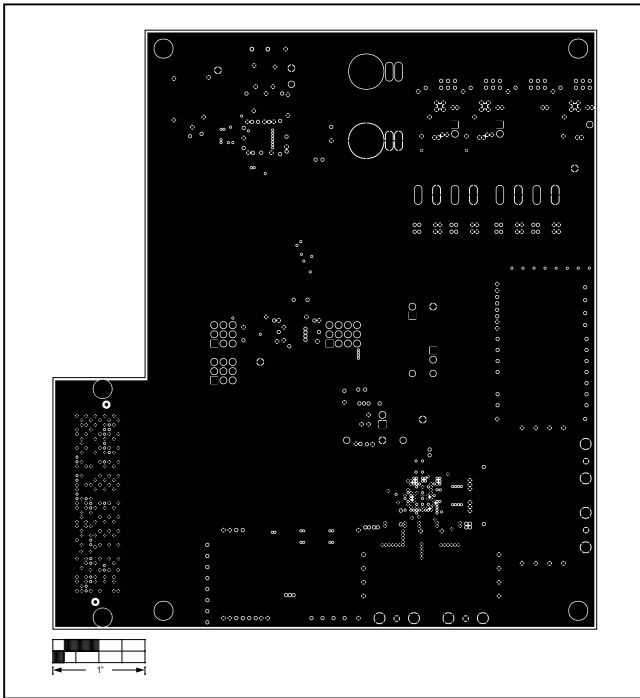
MAX5869 EV Kit PCB Layout Diagrams (continued)



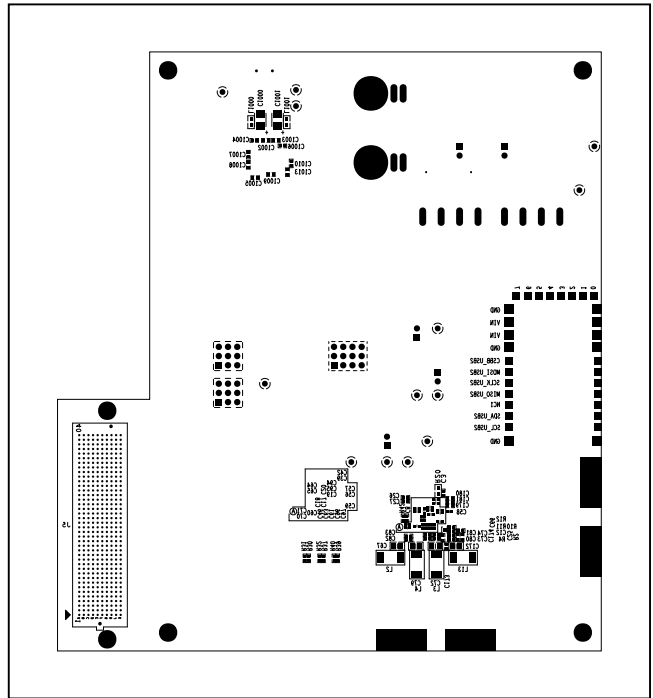
MAX5869 EV Kit—Layer 8



MAX5869 EV Kit—Bottom Copper

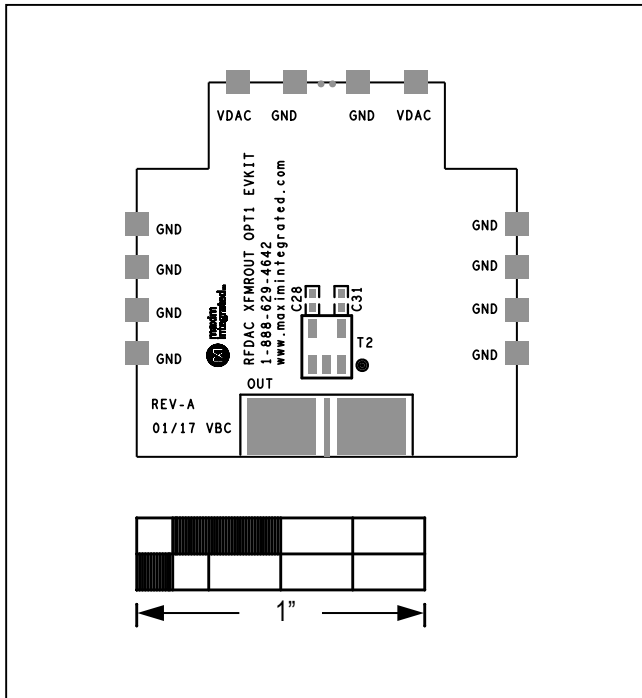


MAX5869 EV Kit—Layer 9

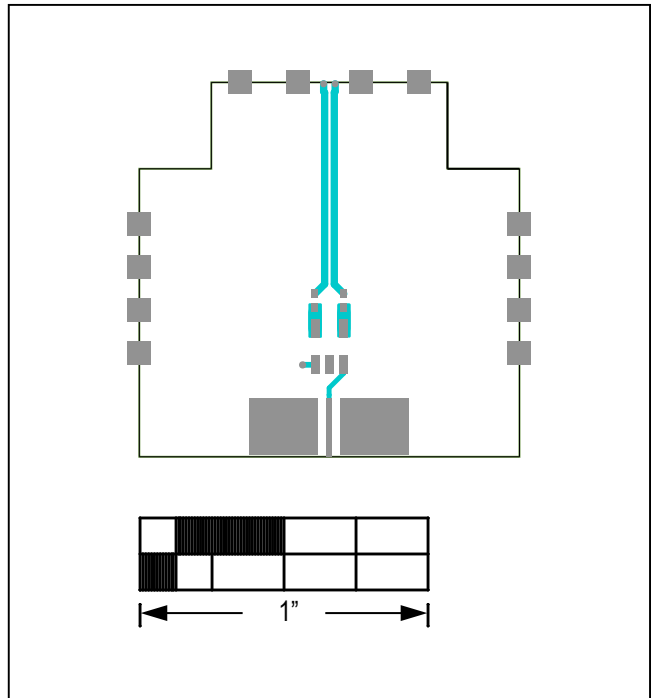


MAX5869 EV Kit—Bottom Silkscreen

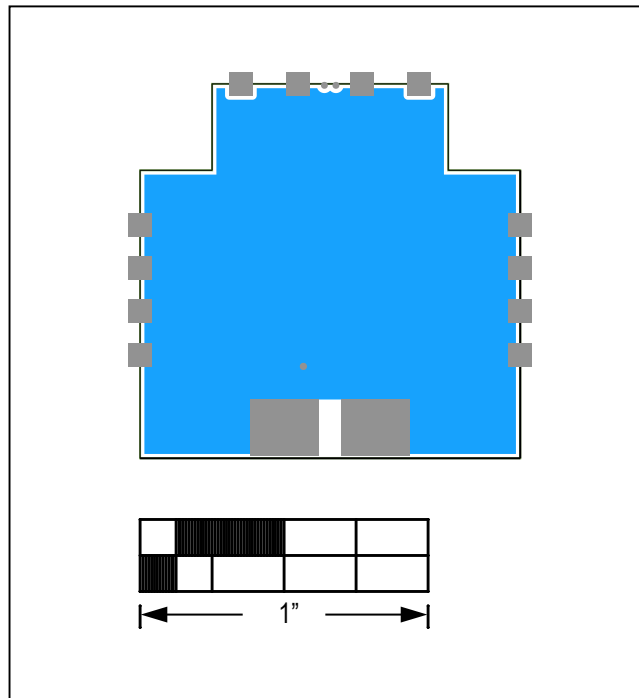
### Transformer Output Module Layout Diagrams



Transformer Output Module—Top Silkscreen



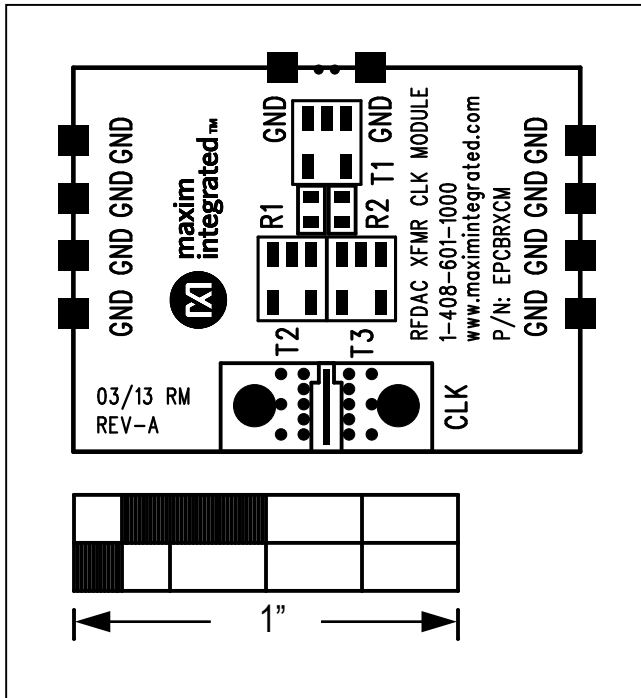
Transformer Output Module—Top



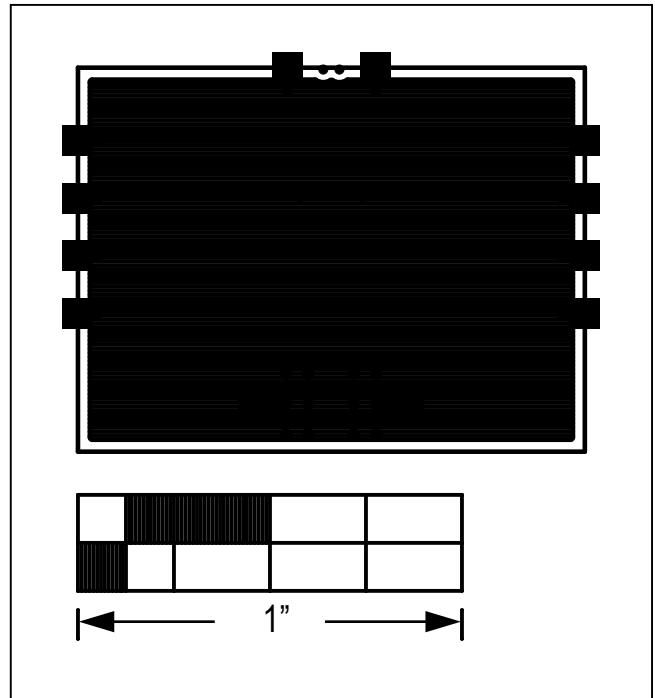
Transformer Output Module—Bottom



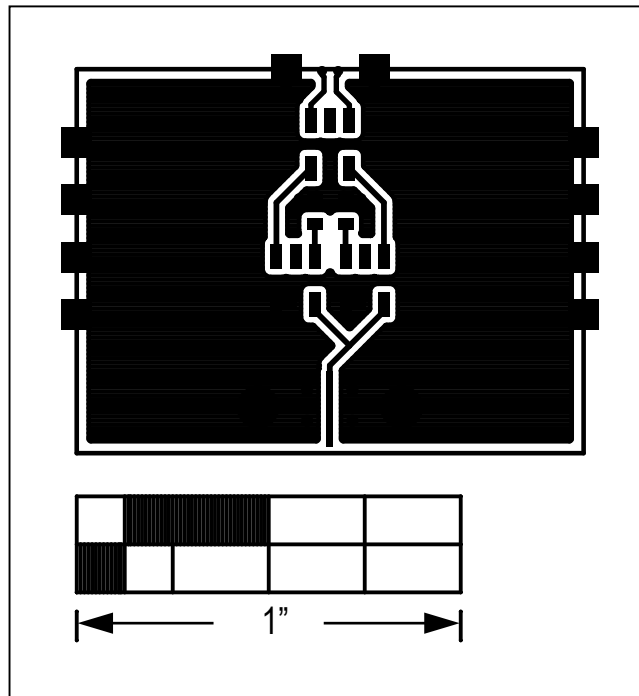
**Clock Module Layout Diagrams**



*Clock Module—Top Silkscreen*



*Clock Module—Solder Side*



*Clock Module—Component Side*

## Appendix I – Software and Driver References

### Third-Party Software and Driver Installation

For the MAX5869 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](http://www.xilinx.com); type “Lab Tools 14.7 Download” on 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.

- 1) Double-click on the **xsetup.exe** program to start installation of the Xilinx ISE LabTools software.
- 2) Windows may prompt to allow the following 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.
- 4) Two **Accept License Agreement** pages appear, check the boxes and click **Next >** to continue.
- 5) The **Select Products to Install** page appears, the **Lab Tools – Standalone Installation** should be selected, click **Next >** to continue.

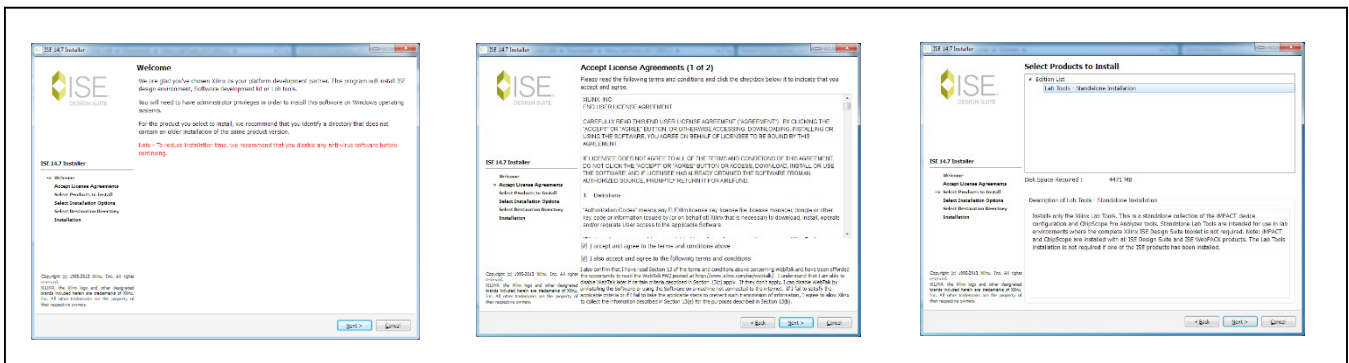


Figure A1-1. Various LabTools Installation Screens

- 6) The **Select Installation Options** page will appear (Figure A1-2), uncheck the **Acquire or Manage a License Key** option and click **Next >** to continue

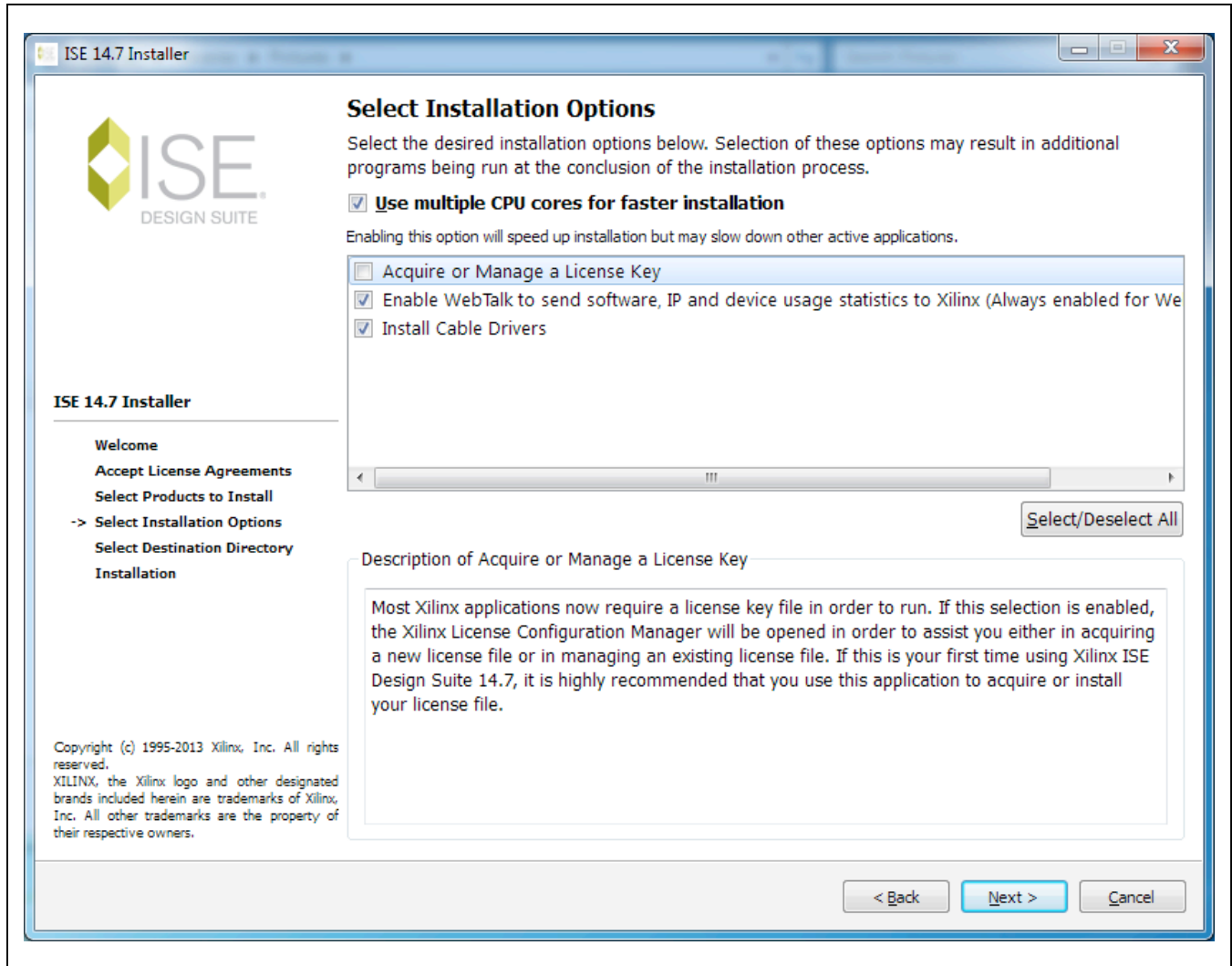


Figure A1-2. LabTools Installation – Uncheck “Acquire or Manage License Keys”

- 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.

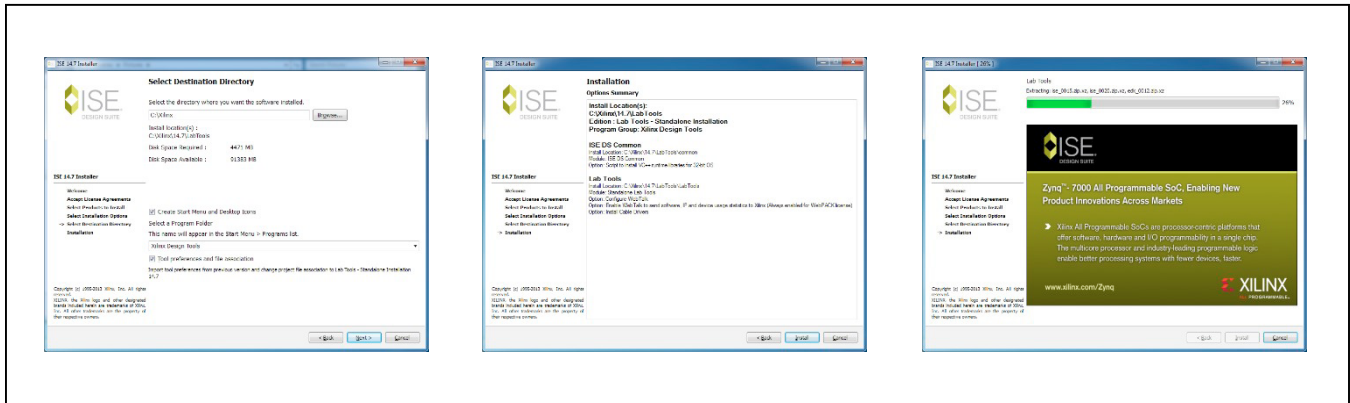


Figure A1-3. Various LabTools Installation screens

- 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 **Next >** to continue.
  - b. The **License Terms** page will appear, check the box and click the **Install** button to continue.
  - c. When the **Setup Complete** page appears, click 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 installer will also be prompted by 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.

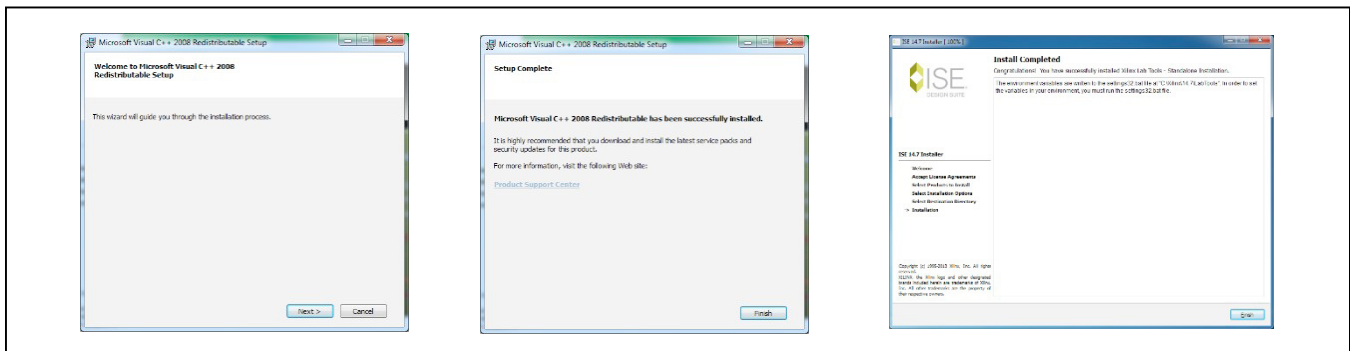


Figure A1-4. Microsoft Visual C++ Installation and ISE Completion

## 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:  
**C:\Xilinx\14.7\LabTools\LabTools\bin\nt** (or \nt64)
- 2) Execute the **install\_drivers.exe** application

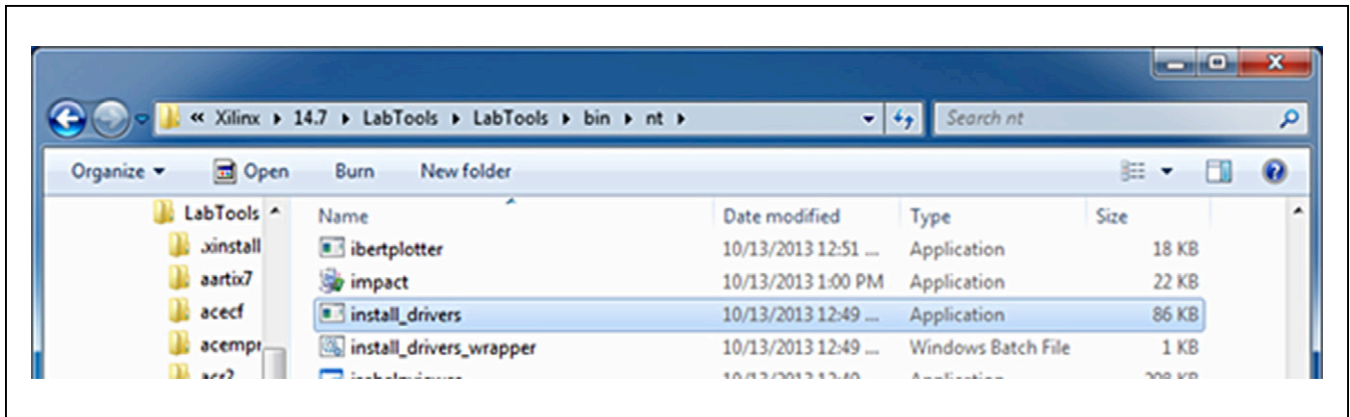


Figure A1-5. Driver Installation

- 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.

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

## VC707 USB2.0 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 MAX5869 SPI.

A **USB MS Bulk Transfer** folder can be found on the flash drive and is also created during the MAX5869 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.

It is recommended the user “extract all files” in order to properly install the driver required for operating the VC707.

## DLL Work-around

There is a workaround for the Xilinx Impact tools to work with Win 10 and the latest updates. A DLL must be swapped out in order for the EVK SW call to Impact to properly execute.

If using default Xilinx Directory structure on the installation of the LabTools:

In C:\Xilinx\14.7\LabTools\LabTools\lib\nt64 directory, rename the file “libPortability.dll” to “libPortability.dll.orig”

In that same directory, copy the “libPortabilityNOSH.dll” file to the same folder, and rename it to “libPortability.dll”

In C:\Xilinx\14.7\LabTools\common\lib\nt64 directory, rename the file “libPortability.dll” to “libPortability.dll.orig”

Copy the “libPortabilityNOSH.dll” from C:\Xilinx\14.7\LabTools\LabTools\lib\nt64 directory into the C:\Xilinx\14.7\LabTools\common\lib\nt64 directory and rename it to “libPortability.dll”

### Removal/Uninstallation of the MAX5869 EV Kit Software

For the MAX5869 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.

### Upgrading/Updating the MAX5869 EV Kit Software

Occasionally upgrades or updates may be available for the MAX5869EVKIT GUI software. Be sure to check the Maxim website from time-to-time for information on the MAX5869 evaluation system. Applying upgraded software may require the removal of previous versions.

### Installing Unsigned Drivers on Windows 10

- 1) Press the Win + C and click on PC settings.
- 2) Switch over to the Update & recovery section.
- 3) Click on the Recovery option on the left-hand side.
- 4) Once selected, an advanced startup section appears on the right-hand side. Click on the Restart now button.
- 5) Once the computer has rebooted, choose the Troubleshoot option.
- 6) Go to Advanced options.
- 7) Then Startup Settings.
- 8) Since boot time configuration settings are being modified, restart the computer.
- 9) Here a list of modifiable startup settings is provided. The applicable option is Disable driver signature enforcement. To choose the setting, press the F7 key.
- 10) The Windows operating system will now boot and the user will be able to install the unsigned driver.  
**NOTE:** Driver Signature Enforcement will be reactivated upon the next boot cycle.

## Appendix II. MicroBlaze Command Set

### List of Commands

COMMAND	DESCRIPTION
help	Prints help. Use -a flag for all commands. Example: "help -a"
mem	Read or write DDR memory
usb	Read or write DDR memory with USB bulk transfer
reg	Read or write a register
capture	manage capture DMA channel
play	manage play DMA channel
ping	Does nothing but ack
scramble	Turn JESD204B scrambling on or off for RX or TX
loopback	Turn the loop back on or off
resync	Force the data source to resync
jesdreset	Force a reset of the JESD subsystem
SPI	Read or write to SPI
configure	Specify how many RX and TX lanes to use, and how to map data
baudrate	Set the baud rate of the serial port
quit	Quits this program



**Description/Syntax of VC707 Command Set**

<b>COMMAND: help</b>	
Online help is provided for all commands. Short and long version are available. The help command with no options will print a list of top level commands.	
Command: help <arg> ... <arg> <-flags>	
Args:        command, or command and subcommand to get help about.	
Flags: --all or -a    Print help for all the commands	
<b>COMMAND: mem</b>	
The mem commands are for reading and writing the DDR memory used for test data.	
mem [read write] [address] [number of bytes] [word] ... [word]	
Available mem commands: read        Read a range of memory write       Write a range of memory	
<b>mem read:</b>	
Read a specific number of 32 bit words from a given address of DDR memory.	
Flags allow different output formats. Default mode is to send data in ASCII format. Use the -b flag to send in binary mode. When 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.	
Command:	
mem    read    [address] [number of bytes] <-flags>	
Arguments: address:        address to read from in any format that strtoul will parse number of bytes: how many bytes to read in multiples of 4 flags: --binary    -b    send data in binary mode --chksum   -c    send 16 bit IPv4 checksum at the end of the data	

**Description/Syntax of VC707 Command Set (continued)**

<p>Example:</p> <pre>ACK# mem read 0x80000000 8 -c 0x64636261 0x68676665 0x6A6E ACK#  ACK# mem read 0x80000000 8 -b -c abcdefghnj ACK#</pre>
<p><b>mem write:</b></p> <p>Write a specific number of bytes to a given address of DDR memory. Default mode is to send data in ASCII format. Use the -b flag to send in binary mode. When sending in binary mode, follow the -b flag 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.</p>
<p>Command:</p> <pre>mem write [address] [number of bytes] &lt;-flags&gt; [word] [word] ...</pre>
<p>Arguments:</p> <pre>address:      address to write to in any format that strtoul will parse number of bytes:  how many bytes to write in multiples of 4 word:         32 bit values in any format that strtoul will parse</pre>
<p>flags:</p> <pre>--binary  -b  send data in binary mode. --chksum  -c  check 16 bit IPv4 checksum at the end of the data</pre>
<p>Example:</p> <pre>ACK# mem write 0x80000000 8 -c 0x64636261 0x68676665 0x6A6E ACK#  ACK# mem write 0x80000000 8 -b -c abcdefghnj ACK#</pre>
<p style="text-align: center;"><b>COMMAND: usb</b></p>
<p>The usb commands are for reading and writing the DDR memory used for test data.</p>
<pre>usb [read write] [address] [number of bytes]</pre>
<p>Available usb commands:</p> <pre>read      Read a range of memory over USB write     Write a range of memory over USB</pre>

**Description/Syntax of VC707 Command Set (continued)**

<b>usb read:</b>
Read a specific number of bytes from a given address of DDR memory. The data is returned using a USB BULK IN transfer.
Command:
usb read [address] [number of bytes]
Arguments:
address: address to read from in any format that strtoul will parse
number of bytes: how many bytes to read in multiples of 4
Example:
ACK# usb read 0x80000000 8 ACK#
<b>usb write:</b>
Write a specific number of bytes to a given address of DDR memory. The data is sent using a USB BULK OUT transfer
Command:
usb write [address] [number of bytes]
Arguments:
address: address to write to in any format that strtoul will parse
number of bytes: how many bytes to write in multiples of 4
word: 32 bit values in any format that strtoul will parse
Example:
ACK# usb write 0x80000000 8 ACK#
<b>COMMAND: reg</b>
The reg commands are for reading and writing registers.
reg [read write] [address] <word>
Available reg commands:
read: read a register
write: write a register
list: list registers or reg spaces
set: set register fields by name

**Description/Syntax of VC707 Command Set (continued)**

<b>reg read:</b>
Read a specific 32 bit register at a given address.
Command: reg read [address]
Arguments: Address: address to read from in any format that strtoul will parse
<b>reg write:</b>
Write a specific 32 bit register at a given address of DDR memory.
Command: reg write [address] [word]
Arguments: address: address to write to in any format that strtoul will parse word: 32 bit values in any format that strtoul will parse
<b>reg list:</b>
List the available register spaces, or the registers in a register space.
Command: reg list <register space>
Arguments: register space: optional space name to show the registers in that space
<b>reg set:</b>
Write to control register fields by name. Only the bits of that field within the control register are modified.
Command: reg set <register space > ... <register space> [register name] [value]
Arguments: register space zero or more hierarchical register space names register name name of the control register field to set value value to write to the control register field

**Description/Syntax of VC707 Command Set (continued)**

<b>COMMAND: capture</b>
The capture commands are for configuring, starting, and stopping the capture DMA channel.
Available capture commands: buffer:       Configure the base address and length of capture buffer dumpregs:     Print the capture channel registers start:        Start the capture channel stop:         Stop the capture channel
<b>capture buffer:</b>
Configure the starting address and size of the capture buffer. Must start on an 64 byte alignment, and be a multiple of 512 bytes. The RX DMA engine must be stopped before using this command.
Command: capture buffer [address] [number of bytes]
Arguments: address        address to read from in any format that strtoul will parse number of bytes   how many bytes to read in multiples of 512
<b>capture dumpregs:</b>
Print the contents of the RX DMA engine. Refer to Xilinx document PG021 for their meaning.
Command: capture dumpregs
<b>capture start:</b>
Start the RX DMA engine. The capture buffer command must be used first to define the buffer.
Command: capture start
<b>capture stop:</b>
Stop the RX DMA engine.
Command: capture stop

**Description/Syntax of VC707 Command Set (continued)**

<b>COMMAND: play</b>
The play commands are for configuring, starting, and stopping the play/TX DMA channel.
<p>Available play commands:</p> <ul style="list-style-type: none"> <li>buffer: Configure the base address and length of play buffer</li> <li>dumpregs: Print the play channel registers</li> <li>start: Start the play channel</li> <li>stop: Stop the play channel</li> <li>reset: Reset both channels of the DMA engine.</li> </ul>
<b>play buffer:</b>
<p>Configure the starting address and size of the play buffer.                  Must start on an 64 byte alignment, and be a multiple of 512 bytes.                  The TX DMA engine must be stopped before using this command.</p>
<p>Command:                  play buffer [address] [number of bytes]</p>
<p>Arguments:</p> <ul style="list-style-type: none"> <li>address address to read from in any format that strtoul will parse</li> <li>number of bytes how many bytes to read in multiples of 512</li> </ul>
<b>play dumpregs:</b>
<p>Print the contents of the TX DMA engine.                  Refer to Xilinx document PG021 for their meaning.</p>
<p>Command:                  play dumpregs</p>
<b>play start:</b>
<p>Start the TX DMA engine.                  The play buffer command must be used first to define the buffer.</p>
<p>Command:                  play start</p>
<b>play stop:</b>
<p>Stop the TX DMA engine.</p>
<p>Command:                  play stop</p>

**Description/Syntax of VC707 Command Set (continued)**

<b>COMMAND: scramble</b>
The scramble command is for turning the JESD204B scrambling on/off in the RX and TX paths.
Command: scramble <-flags>
Flags: --yes or -y     enable scrambling --no or -n     disable scrambling --tx or -t     apply to TX --rx or -r     apply to RX
<b>COMMAND: loopback</b>
The loopback command is for turning the loopback on/off in the GTX transceivers for the TX to RX paths.
Command: loopback <-flags>
Flags: --yes or -y     enable loopback --no or -n     disable loopback
<b>COMMAND: resync</b>
The resync command is for forcing the data source transmitter to go through the resync process.
Command: resync <-flags>
Flags: --yes or -y     force sending the sync characters until turned off --no or -n     normal operation no flag means pulse the sync characters on, then back off
Example: resync   This will force the transmitter to send a pulse of sync characters



**Description/Syntax of VC707 Command Set (continued)**

<b>COMMAND: jesdreset</b>	
The jesdreset command is for forcing the data source JESD subsystem to go through a reset cycle. This will reset the JESD core and the DMA engines. It does not reset the processor, or the DDR memory.	
Command: jesdreset	
Example: jesdreset	
<b>COMMAND: spi</b>	
The spi command is used to perform read write operations to the MAX5869 configuration registers.	
Command: spi <write/read>	
<b>COMMAND: configure</b>	
The configure command is used to specify how many RX and TX lanes should be used, and how the data should be mapped. Currently, there is only a TX mapper, and not an RX demapper.	
Command: configure <number> <-flags>	
number	411,422,221,141,400
411	4 lanes, 1 octet per frame, 1 sample per frame
422	4 lanes, 2 octets per frame, 2 samples per frame
221	2 lanes, 2 octets per frame, 1 sample per frame
141	1 lane, 4 octets per frame, 1 sample per frame
400	legacy mode. 4 lanes, straight through mapping
Flags: --tx or -t   configure TX --rx or -r   configure RX	
Example:  configure 411 -t -r	
<b>COMMAND: baudrate</b>	
Set the baud rate of the serial port.	
Command: baudrate [rate]	
valid rates: 9600 14400 19200 38400 57600 115200 230400 460800	

### Ordering Information

PART	TYPE
MAX5869EVKIT#	EV Kit
EK-V7-V707-G	Xilinx Virtex 7 FPGA Board*

#Denotes RoHS compliant.

\*Order from Xilinx or authorized distributor.

## Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	12/15	Initial release	—
1	8/19	Updated content to reflect changes to user software	1–57

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.*