MAX5871 Evaluation Kit

Evaluates: MAX5871

General Description

The MAX5871 evaluation kit (EV kit) contains a single MAX5871 high-performance interpolating and modulating 16-bit 5.9Gsps 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 direct RF synthesis of single and multicarrier transmit signals in wireless communications applications. The MAX5871 meets spectral mask requirements of a wide range of communication standards, including MC GSM, UMTS, and LTE. The EV Kit provides a complete system for evaluating the performance of the MAX5871 and a platform for system development.

The EV kit connects to one FMC connector on the Xilinx® VC707 evaluation kit, allowing the VC707 to communicate with the MAX5871's JESD204B serial link interface.

The EV kit includes Windows® 7/8 and Windows XP®-compatible software that provides a simple graphical user interface (GUI) for configuration of all of the MAX5871 registers through the SPI interface, control of the VC707 FPGA, and temperature monitoring.

Features

- Evaluates MAX5871 RF DAC Performance, Capability and Feature Set
- Single 3.3V Input Voltage Supply
- Maximum 5.9Gsps Update Rate
- Direct Interface with Xilinx VC707 Data Source Board
- Windows 7/8 and Windows XP-Compatible Software
- On-Board SPI Interface Control for the MAX5871
- 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 XP is a registered trademark and registered service mark of Microsoft Corporation.

Xilinx is a registered trademark of Xilinx.

19-8534; Rev 0; 5/16

maxim integrated™

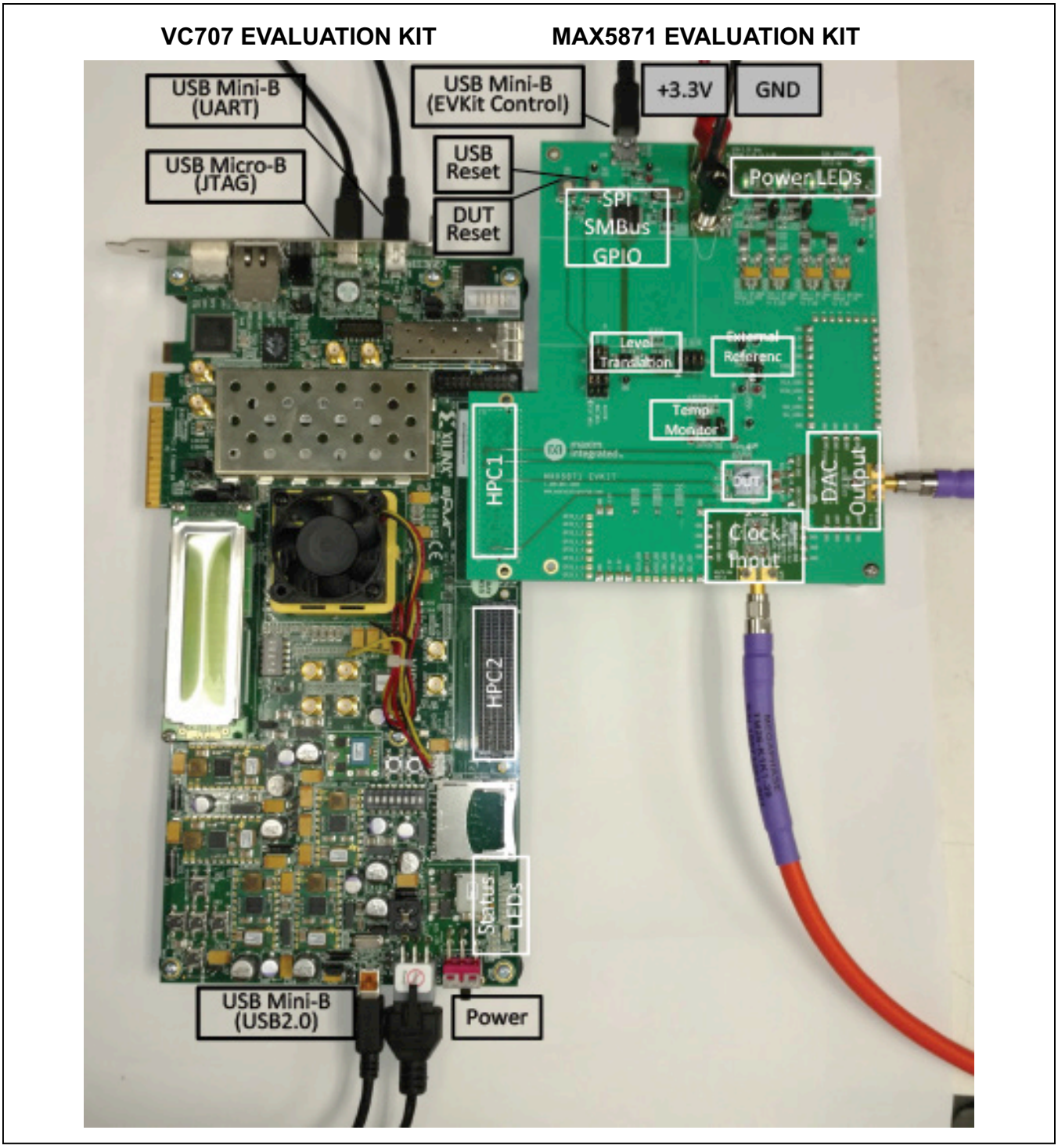


Figure 1. MAX5871/VC707 Evaluation Setup

Quick Start

Required Equipment

- Window PC (Win-7 Recommended, Win-XP, Win-8 optional), with one USB 2.0
- Spectrum Analyzer – Agilent PXA or equivalent
- RF signal generator – Rohde & Schwarz SMF100A or equivalent
- 3.3V, 3A power supply for MAX5871 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 MAX5871 EV kit
  - Two 1” stand-offs with screws
  - MAX5871 EV kit board
  - 4-port USB-powered hub
  - Two USB-A to Mini-B cables, additional one for the VC707 and one for MAX5871 EV kit board
  - USB thumb drive with documentation and all required software
  - Quick start guide

Required Installed Software and Drivers

The MAX5871EVKIT software controller application requires the following drivers and software components to be installed:

- **Xilinx ISE 14.7 LabTools**
  Installation: Browse the Xilinx folder on the thumb drive and run the LabTools Installation program. Lab Tools is a free tool set used for programming the VC707 Evaluation Board, no software registration or license is required. 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
- **Silicon Labs Driver**
  Installation: Browse to the SiLabs folder on the thumb drive and run the installation program for the CP210X Driver for the Windows Platform. Alternatively, the driver and instructions for installation can be accessed at the following location:
  https://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx

If the above files have not been installed, please allow up to 30 minutes for installation. These files are included on the USB thumb drive provided with the MAX5871 EV kit.

Procedure

1) Install the MAX5871 EV kit Software
   - The MAX5871 EV kit software controller application is included on the USB thumb drive delivered with the EV kit. Insert the thumb drive to a USB port on the target PC with a Win 7 or Win 8 OS. Access the *MAX5871 Setup* folder on the thumb drive and double-click the *MAX5871EVK Setup.exe* file to install the software. Several files and folders are created during installation; the general descriptions of these are provided in Table 1.
   - The installation routine requires the user to accept Maxim’s End-User License Agreement in order to proceed. Upon completion of the installation, a shortcut will be added to the Start Menu (Start -> MaximIntegrated -> MAX5871EVKITSoftwareController) as well as to the user’s desktop. Additionally, a short-cut to both the installation folder and the application will be placed on the user’s desktop. This step should take less than 5 minutes.
2) Setup and Connect the MAX5871 board (refer to Figure 1)
   a. Install the two 1” stand-offs included with the MAX5871 EV kit. Stand-offs should be installed on the DAC output side of the board.
   b. Verify all jumpers on the MAX5871 EV kit PCB are in the default position; refer to Table 2.
   c. Connect the MAX5871 EV kit board to the VC707 board HPC1 FMC connector.
   d. Connect the 3.3V/3A supply to the MAX5871 EV kit and enable the output. Verify the four LED board supply indicators are on and green.

e. 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.

f. 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.

g. Make the USB connections using the 4-port USB2.0 hub.

   i. Connect the USB A–Mini B cable from the MAX5871 mini USB to the hub.

   ii. Connect the USB A–micro B cable (JTAG) from Xilinx VC707 Eval board to the hub.

   iii. Connect the USB A–micro B cable (USB2.0) from Xilinx VC707 eval board to the hub.

   iv. Connect the USB A –mini B (UART) from Xilinx VC707 Eval board to the hub.

   v. Connect the power cube for the USB hub and plug into an appropriate outlet.

   vi. Connect the USB hub 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 MAX5871 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	DECRIPTION
MAX5871EVKITSoftwareController.exe	Application program
AppFiles	Directory with application support files including the USB_MS_Bulk_ Transfer driver
DeviceScripts	Directory with sample MAX5871 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 MAX5871
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
EULA.rtf	End-User License Agreement
Miscellaneous DLLs to include ftd2xx.dll, DTD2XX_NET.dll, libMPSSE.dll and MaximStyle.dll	Supporting DLL files for software operation

Table 2. MAX5871 EV Kit Jumper Settings

JUMPER	POSITION	EVKIT FUNCTION
JU1	Not Installed*	Normal Operation
JU2	Installed*	1.0V LDO drives MAX5871: AVCLK, AVDD1_PLL, AVDD
JU3	Installed*	1.8V LDO drives MAX5871: AVCLK2, AVDD2, RVDD2, AVDD2PLL, VDD2
JU4	Installed Not Installed*	Power for U4—MAX6161—external reference MAX6161 NOT powered
JU5	Installed Not Installed*	MAX5871 external reference connected MAX5871 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^2C) connected to USB SCL, SDA, ALERT (I^2C) connected to FPGA

**Default position.*

3) **Start the** *MAX5871EVKITSoftwareController.exe* located in the *C:\MaximIntegrated\MAX5871* folder. The Splash Screen will display while the USB connections are established, followed by the Main GUI Startup screen.
4) **Load the FPGA configuration**
   a. Select the <VC707> tab.
   b. 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
   c. Click the <Load FPGA Configuration File> button.
   d. A file browser will open in the *C:\maximintegrated\MAX5871\VC707Files* folder. Double-click the *MAX5871_DataSource.bit* file.
   e. A progress bar will display while the FPGA is configured (should take less than 2 minutes).
   f. 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*.
   g. 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\MAX5871\AppFiles\ThirdParty\USB_MS_Bulk_Transfer and select the USB_MS_Bulk_Transfer.inf* file.
   h. The arrival of the new device will appear in the Log window located at the bottom of the GUI as highlighted in Figure 2.

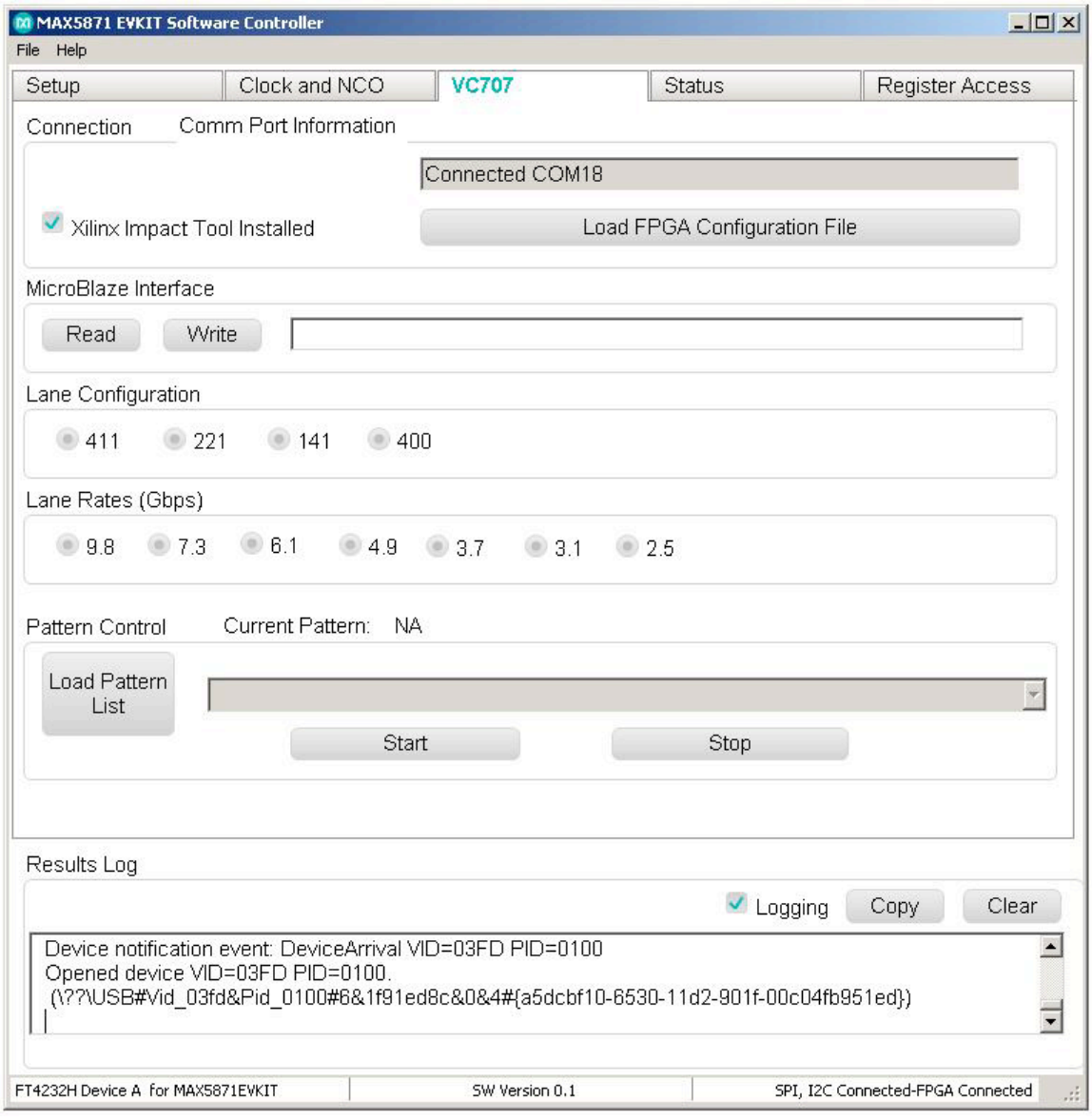


Figure 2. New Device Arrival

5) **Initial Setup is now complete**. It is recommended to reboot the PC at this time. Power off the EV kit, the VC707, the clock source and disconnect the USB hub (power and PC connections) before restarting the system.

6) **Setup the MAX5871 Evaluation System** after the PC has restarted.
   a. Set the RF Generator to the desired input clock frequency (e.g., 737.28MHz, refer to Table 3) and the power to +3dBm, DO NOT ENABLE the RF output yet.
   b. Connect the DAC output to the spectrum analyzer.
   c. Enable the 3.3V power supply. Verify the four LED board supply indicators are on and green.
   d. Enable the RF generator output.
   e. Turn on the VC707 by sliding switch SW12 to the on (left) position.
   f. Verify all LEDs on the VC707 are lit, and the GPIO LEDs are sequencing.
   g. Reconnect the USB hub (power and PC connections).

7) **Start** *MAX5871EVKITSoftwareController.exe* located in the *C:\MaximIntegrated\MAX5871* folder.
   a. Verify that the lower-left corner of the app states either “Connected” or “FT4232H Device A for MAX5871EVKIT”; Indicates the GUI is connected to the MAX5871 EV kit. (Figure 5)
   b. Verify lower-right corner of the app states “SPI, I2C connected-FPGA Connected” indicating these specific ports have been established. (Figure 5)

   NOTE: These notifications must be present before proceeding, and indicates that the JTAG, UART, and EV kit interfaces are connected and operating properly. With this release, these items do no show real time status, but only status at startup.

8) **Click the <Load Settings> button** on the <Setup> tab of the GUI. Select one of the sample configurations from Table 3 (e.g. fDo5898_fC737_I8x_fDi737_J4L7G_RCd4_SC1_DC0.cfg).

Table 3. Example (Preset) Device Configurations

FILE NAME	DAC					JESD204B			
	OUTPUT UPDATE RATE (Gsps)	INPUT CLOCK FREQUENCY (MHz)	PLL MULTIPLIER/ INTERPOLATION RATE	INPUT SAMPLE RATE (Msps)	RCLK DIVIDER	LANE RATE (Gbps)	NO. OF LANES	SUB-CLASS	RXLINK CLOCK
fDo4915_fC307_I16x_fDi307_J4L3p1G_RCd2_SC1_DC0.cfg	4.9152	307.2	16	307.2	2	3.072	4	1	DSP
fDo4915_fC614_I8x_fDi614_J4L6G_RCd4_SC1_DC0.cfg	4.9152	614.4	8	614.4	4	6.144	4	1	DSP
fDo5898_fC245_I24x_fDi245_J4L2p5G_RCd1_SC1_DC0.cfg	5.89824	245.76	24	245.76	1	2.4576	4	1	DSP
fDo5898_fC368_I16x_fDi368_J4L3p7G_RCd2_SC1_DC0.cfg	5.89824	368.64	16	368.64	2	3.6864	4	1	DSP
fDo5898_fC491_I12x_fDi491_J4L5G_RCd2_SC1_DC0.cfg	5.89824	491.52	12	491.52	2	4.9152	4	1	DSP
fDo5898_fC737_I8x_fDi737_J4L7G_RCd4_SC1_DC0.cfg	5.89824	737.28	8	737.28	4	7.3728	4	1	DSP

9) **Select the <Clock and NCO> tab**. Setup the GUI to match the requirements for the previously selected .cfg file. For this example:
   a. Click the <PLL ENABLE> button.
   b. Select **737.28** in the fCLK dropdown box.
   c. Select **8** in the multiplier dropdown box.
   d. Select **8** in the Interpolation Rate dropdown box.
   e. Select **4** in the RCLK Divide dropdown box.
   f. Click the <Apply Settings> button. The f_{DAC} text box should update with the value of 5.89824GHz. The f_{RCLK} text box should update with the value of 184.32MHz.
10) **Configure the NCO frequency** – center frequency for DAC output signals.
   a. Enter the desired center frequency in the Target fNCO text box.
   b. Click the <Calculate Values> button. The CfgFNCO box should update with an 8 character hex value.
   c. Click the <Apply Values> button. The fNCO text box should update with the programmed center frequency.
11) **Select the <VC707> tab** in the GUI
   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 *MAX5871_DataSource.bit* file and click the <Open> button.
      iii. After loading, the GUI should look like Figure 2, above.
   c. Select 411 in the Lane Configuration box.
   d. Select 7.3 in the Lane Rate (Gbps) box.
   e. Click the <Load Pattern List> button in the Pattern Control box.
      i. Select *patListFile.txt* in the file browser and click <Open>.
   f. Select the appropriate pattern (e.g., 2ToneSpc1M_737p28Msps_m1dBFS) in the drop-down box.
   g. Click the <Start Pattern> button.
12) **Enable the DAC Output**
   a. Select the <Setup> tab.
   b. Uncheck the <Hardware Mute> and <Software Mute> buttons.
   c. Observe the signals for the loaded pattern, centered at fNCO on the spectrum analyzer.
13) In order to change the center frequency select the <Clock and NCO> tab and repeat step 10).
14) In order to change the DAC configuration or load new patterns:
   a. Select the <VC707> tab and click the <Stop Pattern> button.
   b. Repeat steps 6) thru 9) for new configurations.
   c. Repeat steps 9e) thru 9g) for a new pattern list.

Refer to the *Test Pattern Lists and Files* section for details regarding the creation of custom patterns and lists. The *Custom Configurations* section contains a detailed procedure for creating configuration files.

Detailed Description of Software

The MAX5871 EV kit software controller GUI is designed to control the EV kit and the VC707 board, as shown in Figure 3. The MAX5871 EV kit software controller includes USB controls that provide SPI and SMBus communication to the MAX5871 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 3) is used for the transfer of patterns to the VC707 on board memory.

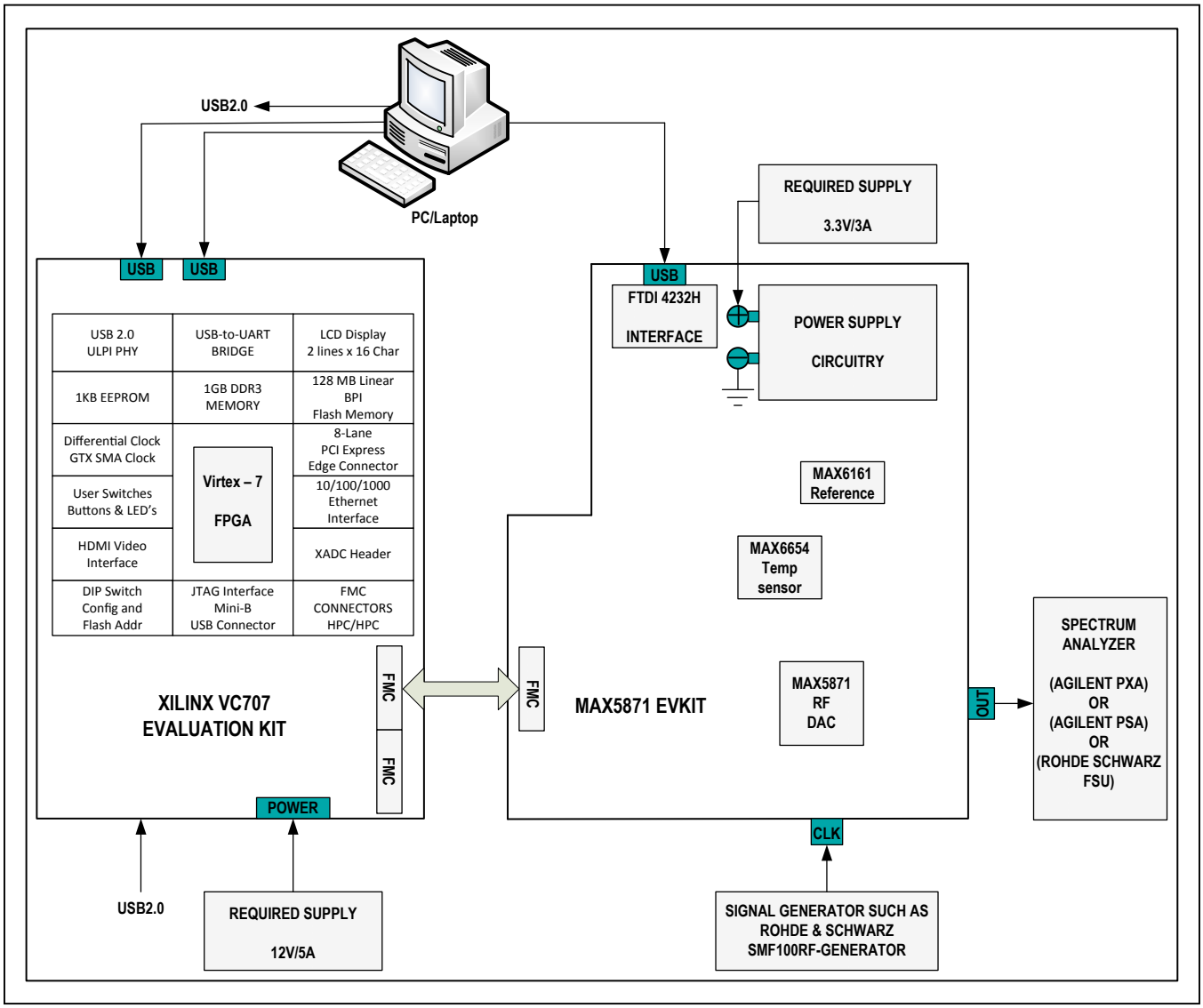


Figure 3. MAX5871 EV System Block Diagram

MAX5871 EV Kit Software Controller

The EV kit software controller GUI application displays a splash screen on initial startup, as shown in Figure 4. The splash screen is only visible while the application connects to the EV Kit USB interface and the UART on the VC707. The main application is displayed after these connections are made, as shown in Figure 5. 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 I2C/SMBus along with the FPGA connections.

The EV kit software controller features five window tabs for configuration and control of the MAX5871 and the VC707. The specific tabs are:

- Setup
  - Load and reload MAX5871 configurations
  - Hardware and software MUTE control
  - Various reset functions
- Clock and NCO
  - Clock configuration
  - NCO configuration
- VC707
  - Com port connection reporting
  - 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
  - Automation support through TCP/IP port
- Register Access
  - User access to read/write MAX5871 configuration and status registers

Figure 4. Splash Screen

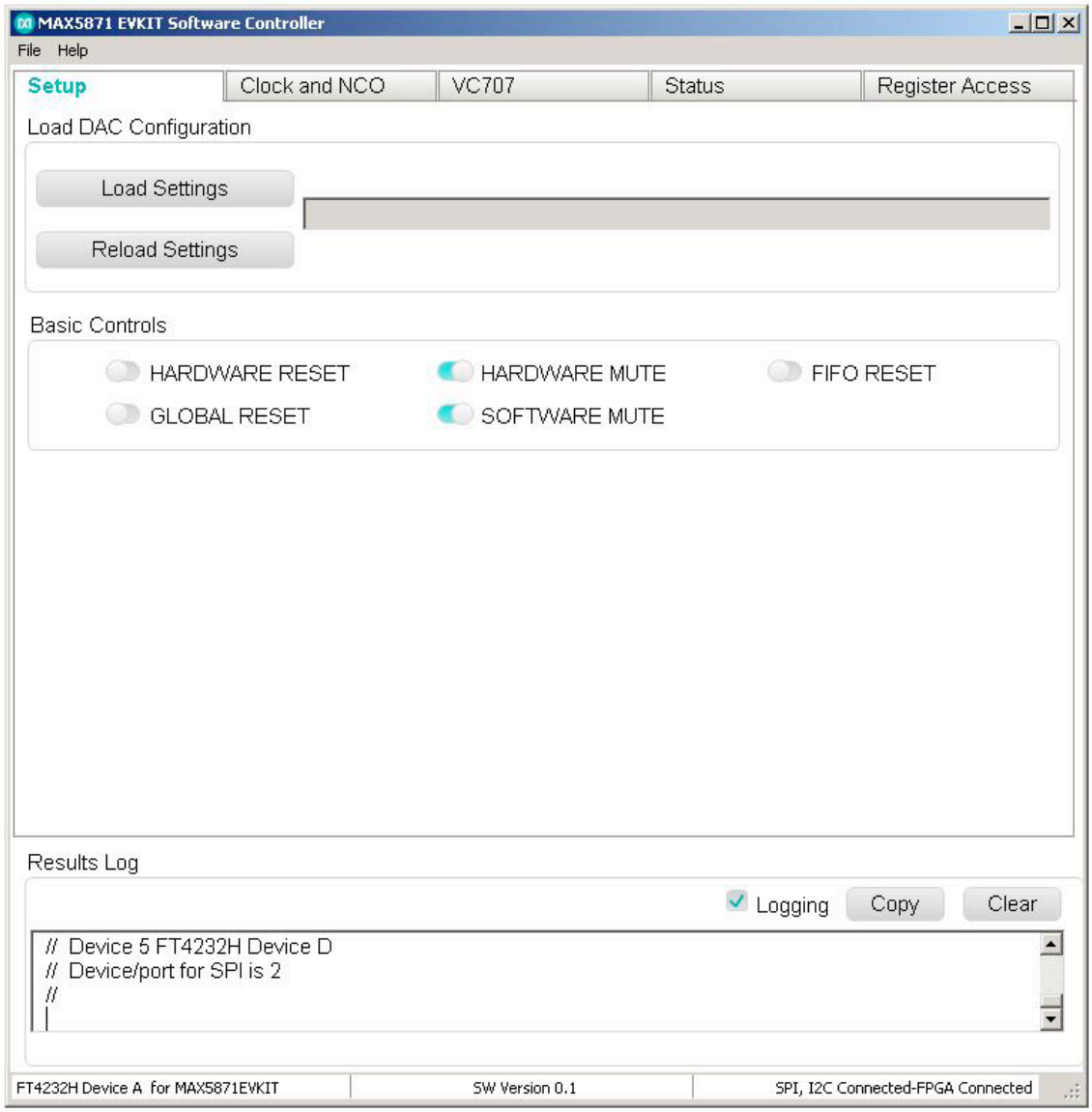


Figure 5. Main GUI Startup

The application will not display the main window if the required EV Kit connections are not complete. If the EV kit has not been properly identified, the pop-up window shown in Figure 6a will appear. The pop-up window in Figure 6b appears when the application does not detect a properly configured FTDI FT4232 device. The Device Manager application of the Windows OS can be used to monitor the connections if desired.

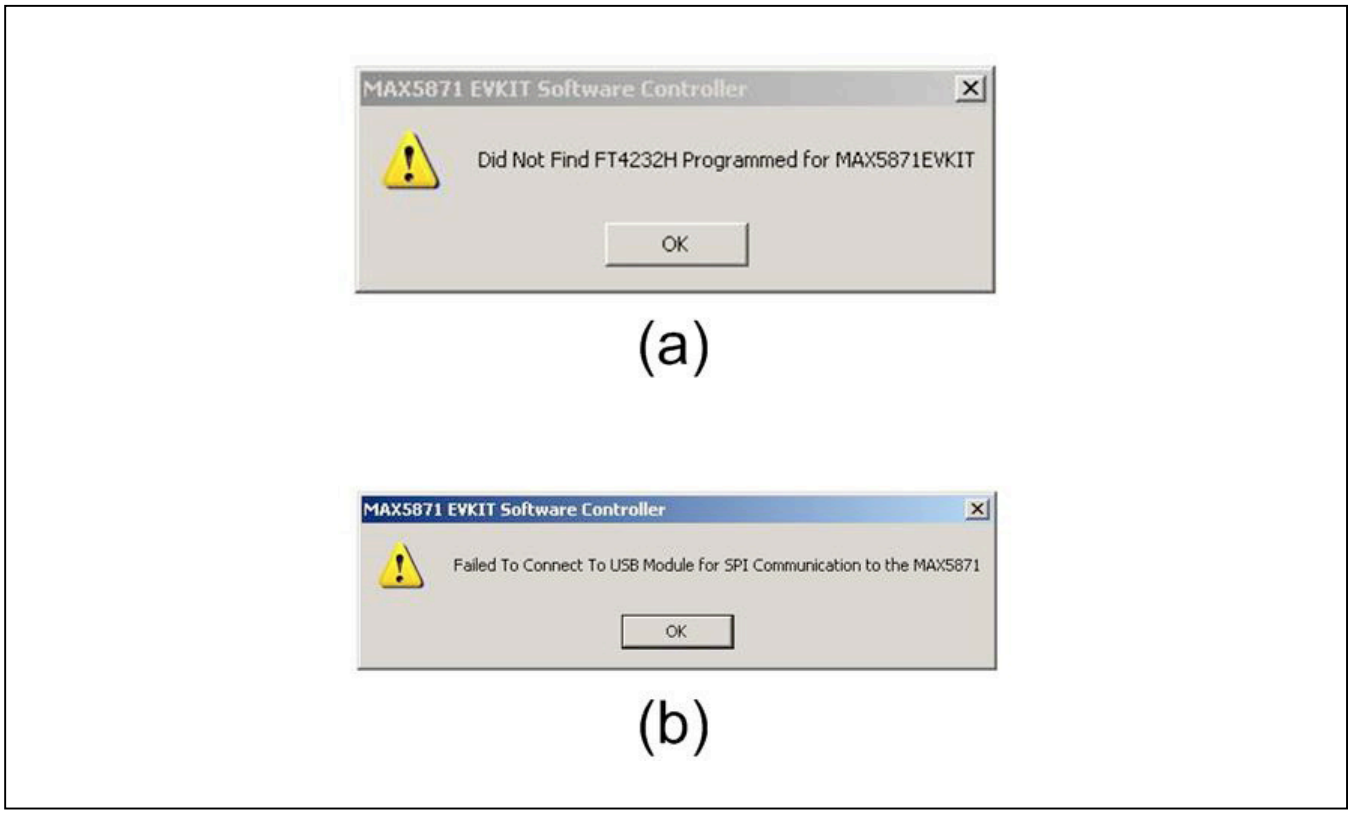


Figure 6. Error in Locating MAX5871EV Kit and in Locating FT4232 Device

Clicking the OK button for either or both of these pop-up windows will cause the MAX5871EVKIT Software Controller Setup window to appear as shown in Figure 7.

If desired, the <Demonstration Mode> button can be clicked to enter the SW GUI. Of course, there isn't a board to connect to and therefore no configuration can take place. If demonstration mode is used, the GUI will display with the appropriate "DEMONSTRATION MODE" title and "Not Connected" and "Adapter: None" messages, as shown in Figure 8.

Setup Tab

The Setup tab (Figure 5) allows the user to load a MAX5871 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 *MAX5871/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 MAX5871.

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.

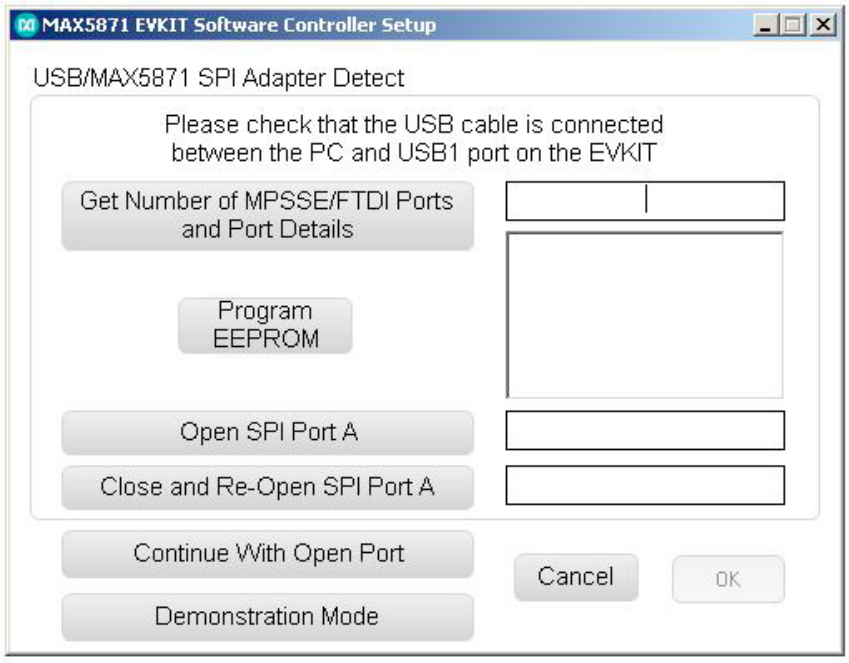


Figure 7. SPI Adapter Setup Window

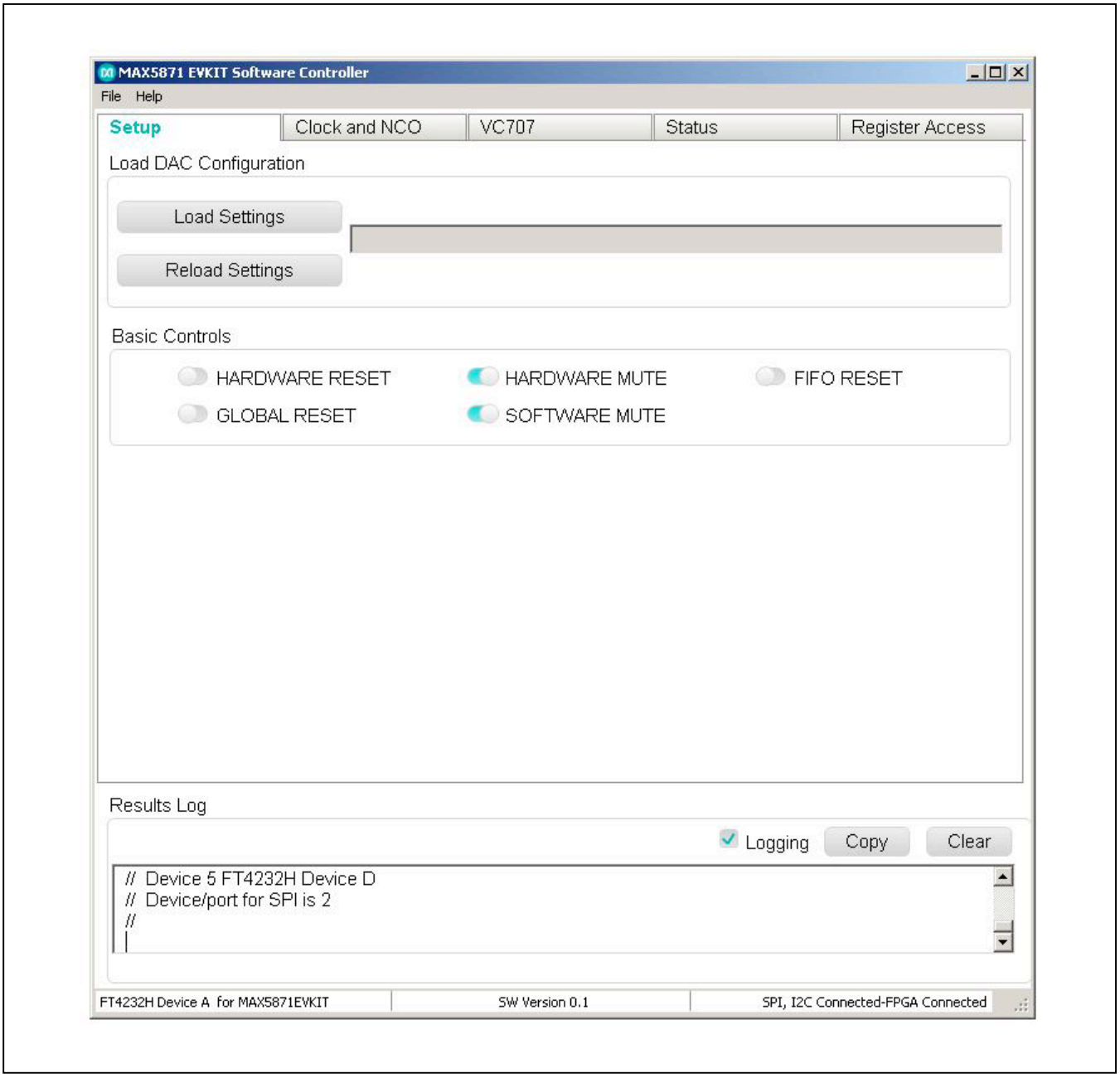


Figure 8. Demonstration Mode Startup

Clock and NCO Tab

The clock and NCO tab (Figure 9) provides the controls required to configure the DAC output update rate and the NCO frequency. The MAX5871 operates normally with a very limited number of discrete update rates. The GUI automatically populates the selections for configuring the clock based on the mode of operation. Should the user desire, the GUI provides an option to <Override Defaults>. Using this option is NOT recommended; contact the factory for additional support if this mode is desired.

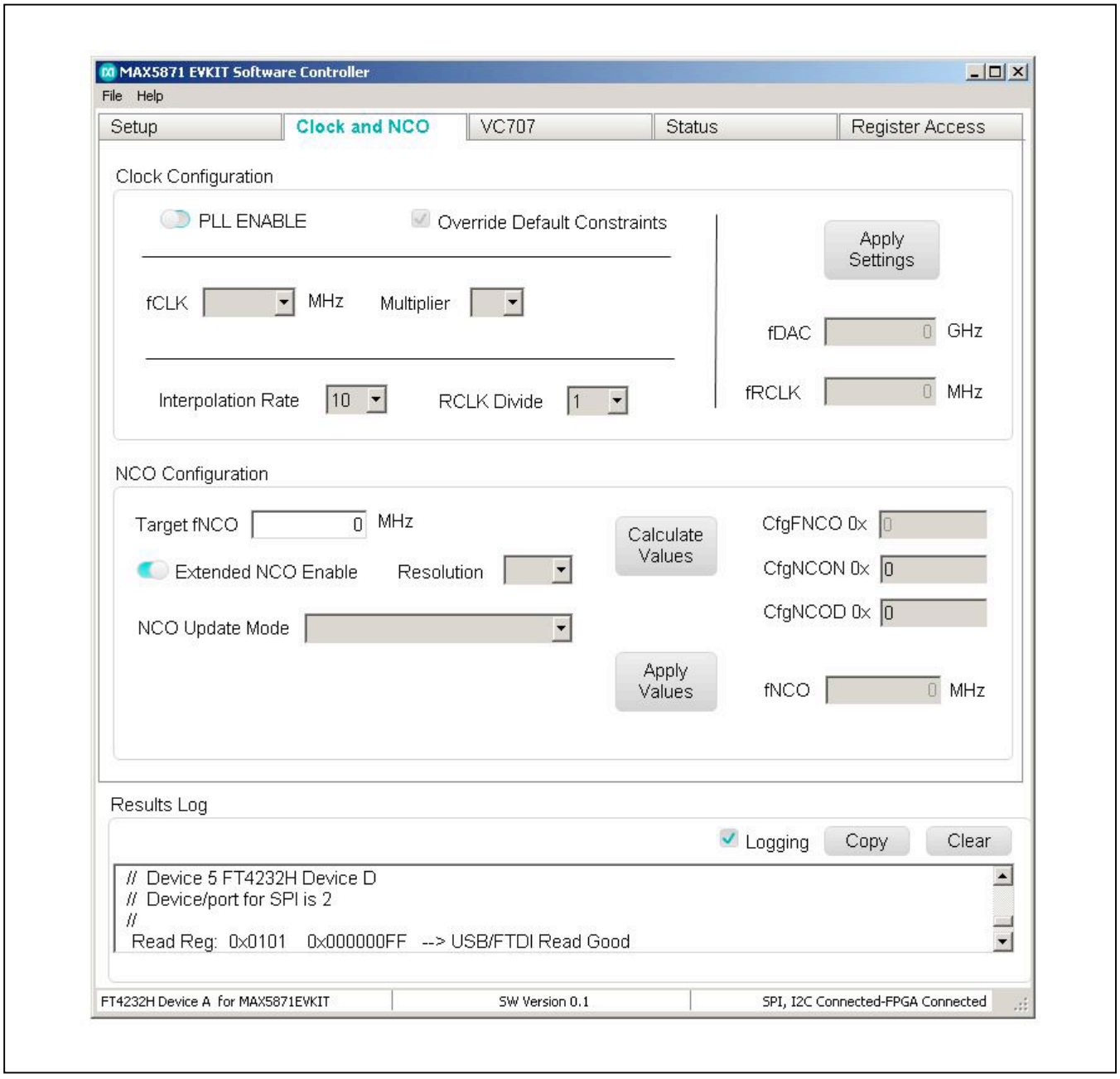


Figure 9. Clock and NCO Tab

The clock configuration section provides the controls needed to configure the DAC update rate. The clocking mode, PLL ENABLED or BYPASS, determines the frequency requirements for the clock signal applied to the EV Kit. Using the PLL ENABLE mode will populate the fCLK selection box with a list of the supported reference frequencies, and the BYPASS mode will populate the list with the allowed DAC update rates. The clock frequency applied to the EVKIT should match the selected value.

The PLL Multiplier selection box will populate with the allowable options based on the fCLK frequency selected. Options range from a minimum of 1 for use only in the BYPASS mode, to a maximum of 60 for use with the lowest reference frequency.

The Interpolation Rate selection box populates based on the fCLK and PLL Multiplier selections. The MAX5871 supports 5x, 6x, 6.67x, 8x, 10x, 12x, 13.33x, 16x, 20x, and 24x interpolation.

The RCLK Divider selection includes options for divide by 1, 2, or 4. The input sample rate is determined by dividing the DAC output update rate by the interpolation rate. The MAX5871 supports input sample rates of 245.76MHz, 307.2MHz, 368.64MHz, 491.52MHz, 614.4MHz, and 737.28MHz. The RCLK frequency is equal to the input sample rate when the divide by 1 option is selected. However, the VC707 firmware requires a specific RCLK frequency based on the input sample rate. Valid frequencies for the RCLK output are: 153.6MHz, 184.32MHz, or 245.76MHz, so the RCLK divide value should be chosen accordingly.

Clicking the <Apply Settings> button will update the MAX5871 registers to match the desired DAC update rate, PLL mode and RCLK output frequencies. The fDAC and fRCLK text boxes will update with the calculated frequencies based on the users selections.

The NCO configuration box contains all the controls required to configure the operation of the NCO used in the digital modulator block of the MAX5871. 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 x 8 = 16384 DAC clocks.

The GUI also provides controls to access the PLLs Fractional or Extended mode. Extended mode is enabled with the Extended NCO Enable toggle button. When this mode is enabled, the Resolution selection box will appear as shown in Figure 9. The available resolution settings are 10kHz, 1kHz, 100Hz, 10Hz, and 1Hz. Additonally, 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 MAX5871. 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 MAX5871.

VC707 Tab

The VC707 tab (Figure 10) provides all the VC707 controls required to: a) load the FPGA firmware, b) configure the VC707 JESD204B serial interface, and c) load, start and stop test patterns. 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).

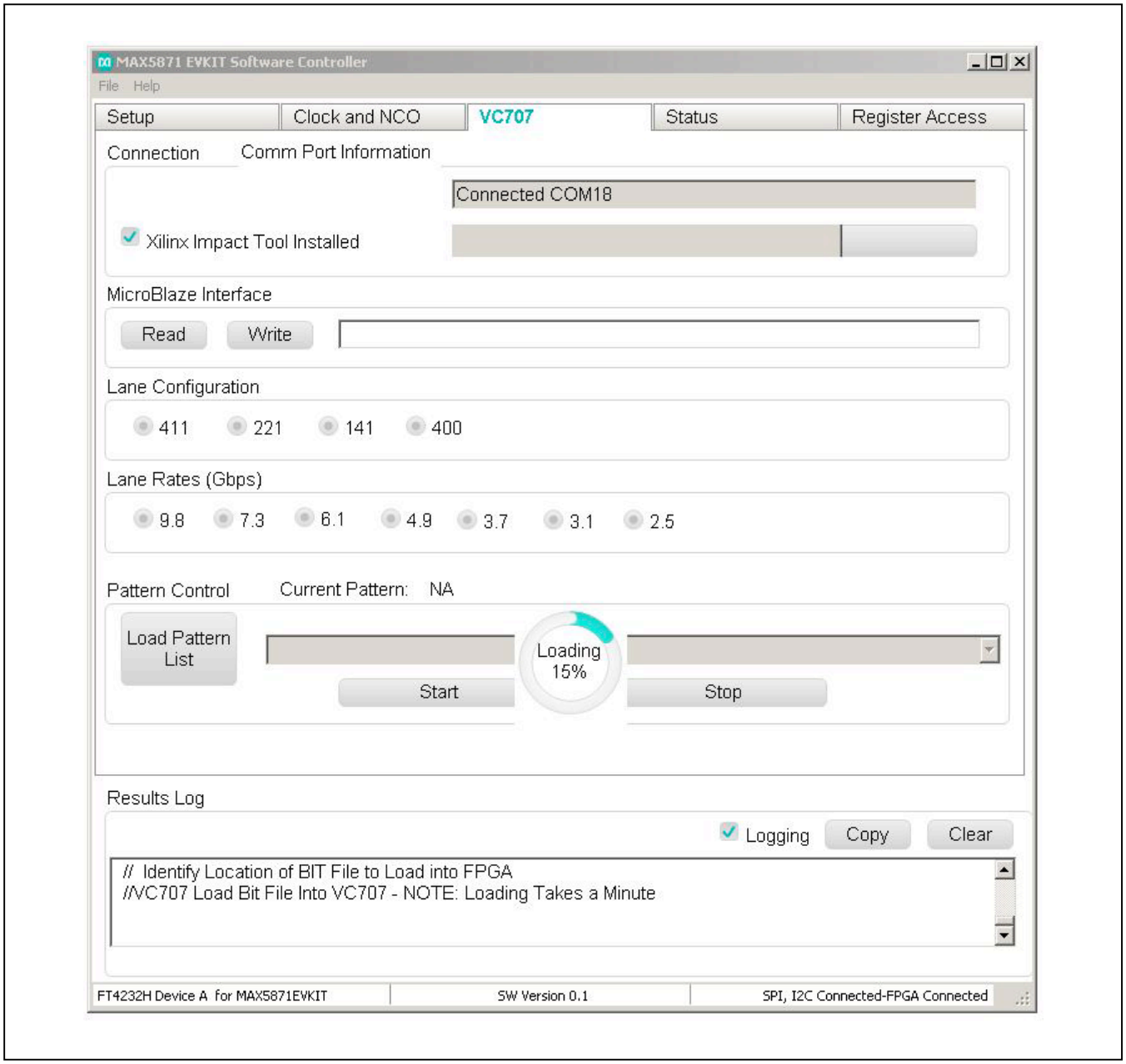


Figure 10. VC707 Tab

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 *MAX5871\VC707Files*. The user selects the MAX5871_DataSource.bit file and clicks open. The MAX5871 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 MAX5871 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.

Next, the Lane Configuration must be selected. The 411 option is used for 4-lane configurations. Similarly, 221 is for two lane and 141 is for single-lane modes. The numbers in the lane configuration selection (411) represent the LMF of the interface where:

L is the number of lanes

M is the number of samples per frame,

F is the number of frames per multiframe.

Refer to the JESD204B standard for more details on the LMF settings of a JESD link.

Next, the lane rate must be selected. This is determined by the register settings that were loaded into the MAX5871 when it was configured. Make sure the rate selected here matches the .cfg file that was loaded.

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 *MAX5871\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 MAX5871, the selected pattern is output to the JESD204B interface.

The <Stop> button must be clicked prior to selecting another pattern or changing the MAX5871 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 11, provides the user interface to the MAX6654 Temperature Monitor IC and the ability to enable control of the MAX5871EVKIT 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.

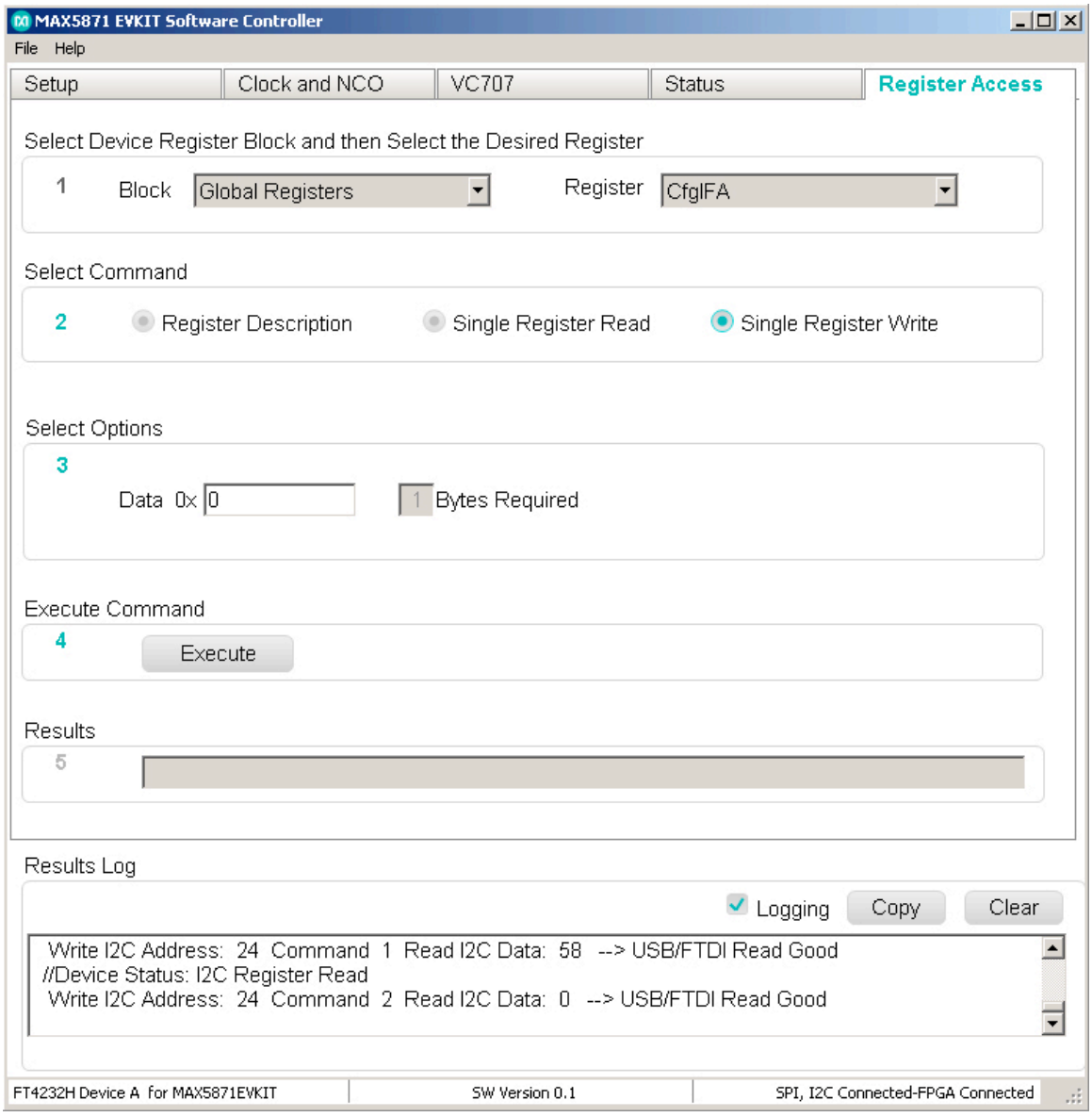


Figure 11. Status Tab

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 MAX5871 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 MAX5871EVKIT PCB. Activating the <Clear Temperature Alert> button will clear the alert. However, if the MAX5871 die temperature is still above the programmed threshold, the LED will immediately re-illuminate.

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 *MAX5871EVKit Automation Support rev#.pdf* document located in the *c:\MaximIntegrated\MAX5871\EVKIT* Info folder that was created during the software installation process.

Register Access Tab

The <Register Access> tab, Figure 12, allows the user to explore and modify the register contents of the MAX5871. 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 MAX5871 can be readily generated using the PERL script included with the software installation. The PERL script *MAX5871_gen_config.pl* is located in the *c:\MaximIntegrated\MAX5871\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 MAX5871 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_{S_IN}: _____
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_{S_IN}) 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\MAX5871\DeviceScripts* folder. As an example, a f_{S_IN} of 737.28Msps requires four lanes at 7.3728Gbps and the RCLK Divider set to 4, while a f_{S_IN} 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_{S_IN} 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\MAX5871\DeviceScripts* folder, the user locates the desired combination of these parameters that match the selected f_{S_IN}.

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 MAX5871 supports JESD204B Subclass-0 and Subclass-1 (required for deterministic latency). The Device Clock Source is a setting in the MAX5871 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.

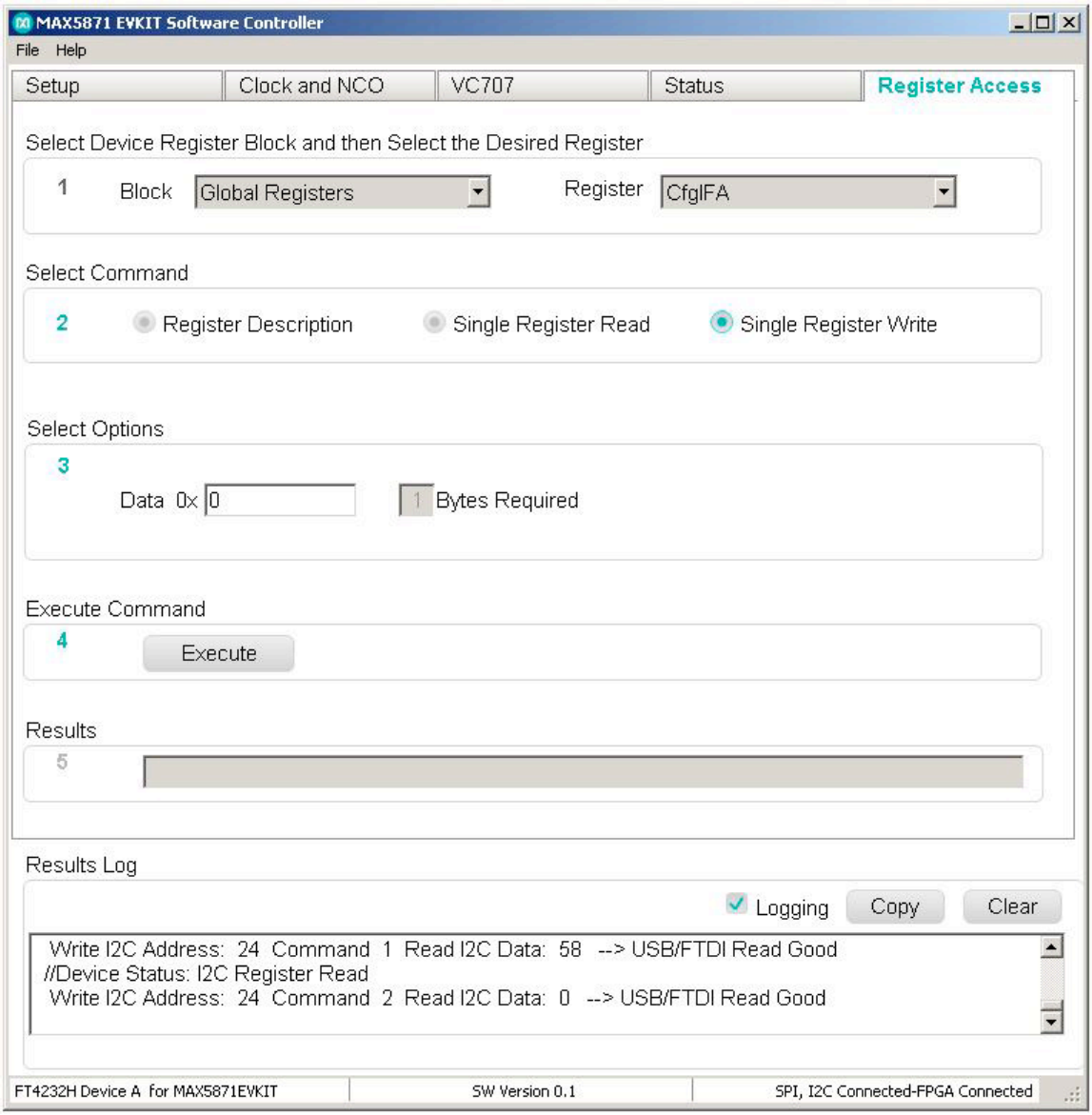


Figure 12. Register Access Tab

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.

Test Pattern Lists and Files

Pattern List Files

The MAX5871 EV kit software controller utilizes list files for loading test patterns into the VC707 memory, such as the example patListFile.txt located in the *MAX5871\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 MAX5871 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 *MAX5871\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 ±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 *MAX5871_PatFile.m* routine which formats the pattern appropriately for use with the VC707. Please refer to the comments, lines that begin with % following the copyright 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 or .txt), as long as the contents are simple text and conform to the format expected by the MAX5871 EV kit 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.*

MAX5871 Evaluation Board

The MAX5871 EV kit board incorporates a USB interface, that provides a serial port interface (SPI) to control the MAX5871 RF DAC, and an SMBus interface to control the MAX6654 (Temperature Monitor). The FTDI4232 USB interface device provides the SPI and I^2C interfaces, as well as GPIO controls for the hard-wired MUTE, INTB and RESETB signals on the MAX5871.

The EV kit employs two on-board modules to allow easy interfacing to signal generators and spectrum analyzers.

1) Clock Input Module (RFDAC_XFMR_CLK_MODULE): converts a single-ended signal generator output to a differential signal that drives the CLKP/CLKN inputs of the MAX5871.
2) Output Module (RFDAC_XFMR_OUT_MODULE): converts the differential output of the MAX5871 DAC to a single ended 50Ω output suitable for driving the input of a 50Ω spectrum analyzer.

The EV kit board requires a single +3.3V, 3A power supply connected to the board through BANANA jacks. The +3.3V supply is used by the various support circuits, and is also used as the input source for the MAX8527 linear regulators (LDO) that provide the 1.8V and 1.0V supplies for the MAX5871. One LDO is used for each level, however the LDO outputs are isolated for analog and digital domains by on board filter networks. Additionally, the PLL supplies for the MAX5871 are also isolated from the analog domain through more filtering.

The EV kit includes a MAX6161 precision reference for use as an external reference for the MAX5871. Power for the MAX6161 is supplied through jumper JU4, while JU5 connects the MAX6161 output to the MAX5871 V_{REF} input.

The MAX5871 EV kit provides direct connection the HPC-1 FMC connector on the VC707, providing a high-quality interconnect that supports the maximum 9.8304Gbps lane rate of the JESD204B interface.

Schematic and layout files for the MAX5871 EV Kit board are included with the software installation files and located in the MAX5871 EV kit info folder

Xilinx VC707 Evaluation Board

The Xilinx VC707 board acts as the data source for the MAX5871, allowing for user defined signal generation. Test patterns, generated externally, are stored in the VC707's on-board DDR memory and subsequently transmitted through the JESD204B to the MAX5871. 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 download commands. The USB2.0 (BULK) interface minimizes the time requirement for downloading the test patterns. Integrated commands allow the VC707 to properly drive all lane rate and speed combinations supported by the MAX5871.

The MAX5871 EV kit software controller also provides a simple interface for controlling the VC707 board. The user will utilize the VC707 tab in the GUI to download the firmware file that 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. The supported set of MicroBlaze commands are listed in Appendix 1 for reference. However, all required commands for normal operation are incorporated into specific controls in the application.

Appendix 1. 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
checksum	Checksum a region of DDR memory
fill	Fill a region of DDR memory with a pattern
ping	Does nothing but ack
memmap	Print a memory map of the system
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
lanes	Specify how many RX and TX lanes to use
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

COMMAND: help
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

Description/Syntax of VC707 Command Set (continued)

COMMAND: mem
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
mem read:
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
Example: ACK# mem read 0x80000000 8 -c 0x64636261 0x68676665 0x6A6E ACK# ACK# mem read 0x80000000 8 -b -c abcdefghnj ACK#

Description/Syntax of VC707 Command Set (continued)

mem write:
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.
Command: mem write [address] [number of bytes] <-flags> [word] [word] ...
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
flags: --binary -b send data in binary mode. --chksum -c check 16 bit IPv4 checksum at the end of the data
Example: ACK# mem write 0x80000000 8 -c 0x64636261 0x68676665 0x6A6E ACK# ACK# mem write 0x80000000 8 -b –c abcdefghnj ACK#
COMMAND: usb
The usb commands are for reading and writing the DDR memory used for test data.
usb [read|write] [address] [number of bytes]
Available usb commands: read Read a range of memory over USB write Write a range of memory over USB

Description/Syntax of VC707 Command Set (continued)

usb read:
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#
usb write:
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#
COMMAND: reg
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)

reg read:
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
reg write:
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
reg list:
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
reg set:
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)

COMMAND: capture
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
capture buffer:
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
capture dumpregs:
Print the contents of the RX DMA engine. Refer to Xilinx document PG021 for their meaning.
Command: capture dumpregs
capture start:
Start the RX DMA engine. The capture buffer command must be used first to define the buffer.
Command: capture start
capture stop:
Stop the RX DMA engine.
Command: capture stop

Description/Syntax of VC707 Command Set (continued)

COMMAND: play
The play commands are for configuring, starting, and stopping the play/TX DMA channel.
Available play commands: buffer: Configure the base address and length of play buffer dumpregs: Print the play channel registers start: Start the play channel stop: Stop the play channel reset: Reset both channels of the DMA engine.
play buffer:
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.
Command: play 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
play dumpregs:
Print the contents of the TX DMA engine. Refer to Xilinx document PG021 for their meaning.
Command: play dumpregs
play start:
Start the TX DMA engine. The play buffer command must be used first to define the buffer.
Command: play start
play stop:
Stop the TX DMA engine.
Command: play stop

Description/Syntax of VC707 Command Set (continued)

COMMAND: checksum
Caclulate a 16 bit IPv4/TCP/IP checksum on a range of memory.
Command: checksum [address] [number of bytes]
Arguments: address starting address in any format that strtoul will parse number of bytes how many bytes to checksum
COMMAND: fill
Fill a range of memory with a pattern.
Command: fill [address] [number of bytes] <word> <-flags>
Arguments: address starting address in any format that strtoul will parse number of bytes how many bytes to fill. Must be a multiple of 4 word 32 bit word to fill memory with
flags: --ramp -r Fill with a 32 bit ramp between start and stop values. --start -s [arg] Starting value for fill pattern. Defaults to 0. --stop -p [arg] Terminal value for fill pattern. Defaults to 0xFFFFFFFF
Example: fill 0x80000000 0x20000 0xA5A55A5A fill 0x80000000 0x20000 --ramp -s 0x1234 --stop 0x5678
COMMAND: scramble
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

Description/Syntax of VC707 Command Set (continued)

COMMAND: loopback
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
COMMAND: resync
The resync command is for forceing 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
COMMAND: jesdreset
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

Description/Syntax of VC707 Command Set (continued)

COMMAND: lanes
The lanes command is used to specify how many RX and TX lanes should be used.
Command: lanes <number> <-flags> number 1, 2, or 4
Flags: --tx or -t set number of TX lanes --rx or -r set number of RX lanes
COMMAND: configure
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
COMMAND: baudrate
Set the baud rate of the serial port.
Command: baudrate [rate] valid rates: 9600 14400 19200 38400 57600 115200 230400 460800

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 MAX5871 when contacting these component suppliers.*

PCB Layout and Schematics

See the links below for PCB layout diagrams and schematics.

- **MAX5871 EV PCB Layout**
- **MAX5871 EV Schematic**

Ordering Information

PART	TYPE
MAX5871EVKIT#	EV Kit
EK-V7-V707-G	Xilinx Virtex 7 FPGA Board*

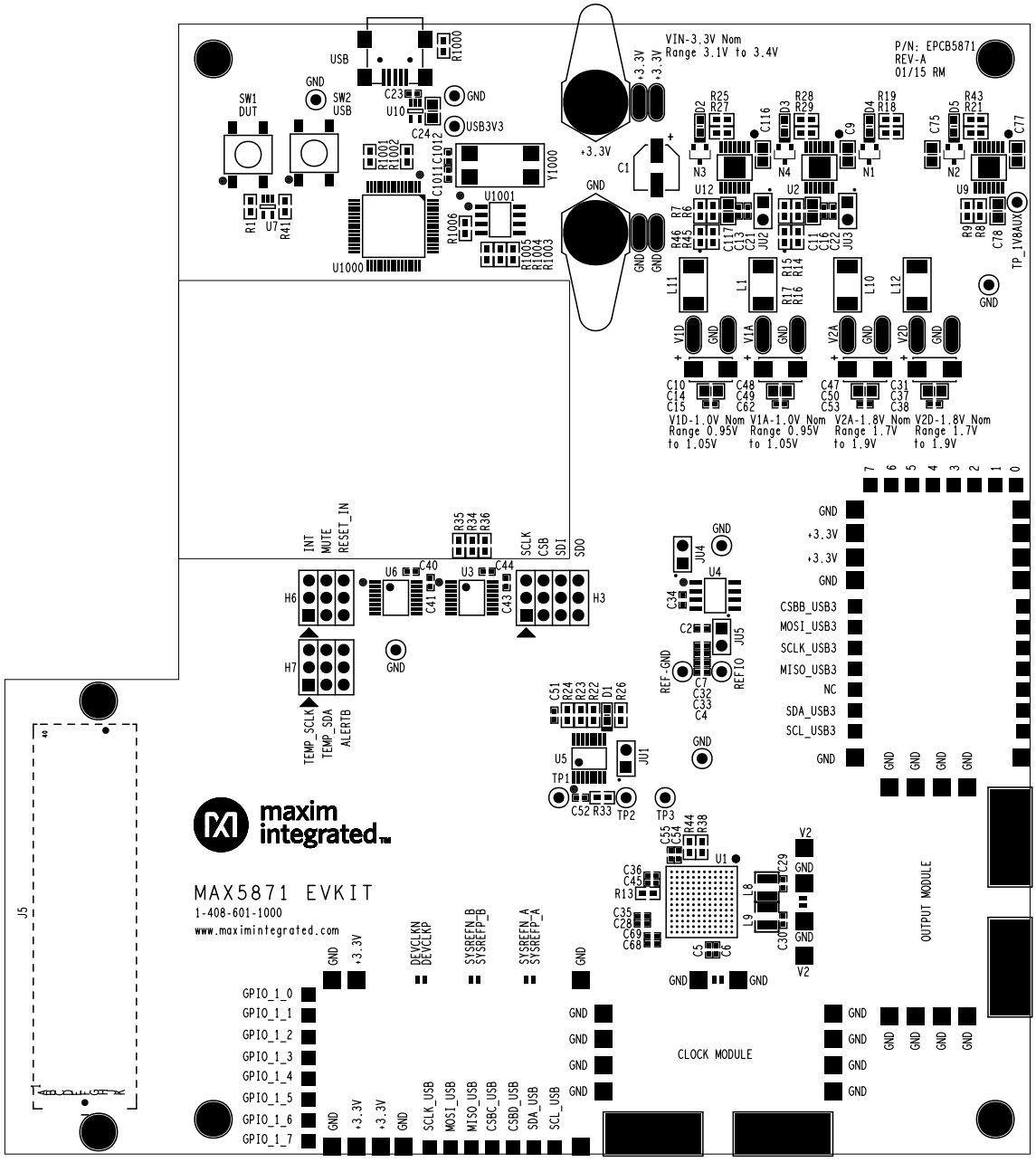
#Denotes RoHS compliant.
**Order from Xilinx or authorized distributor.*

Revision History

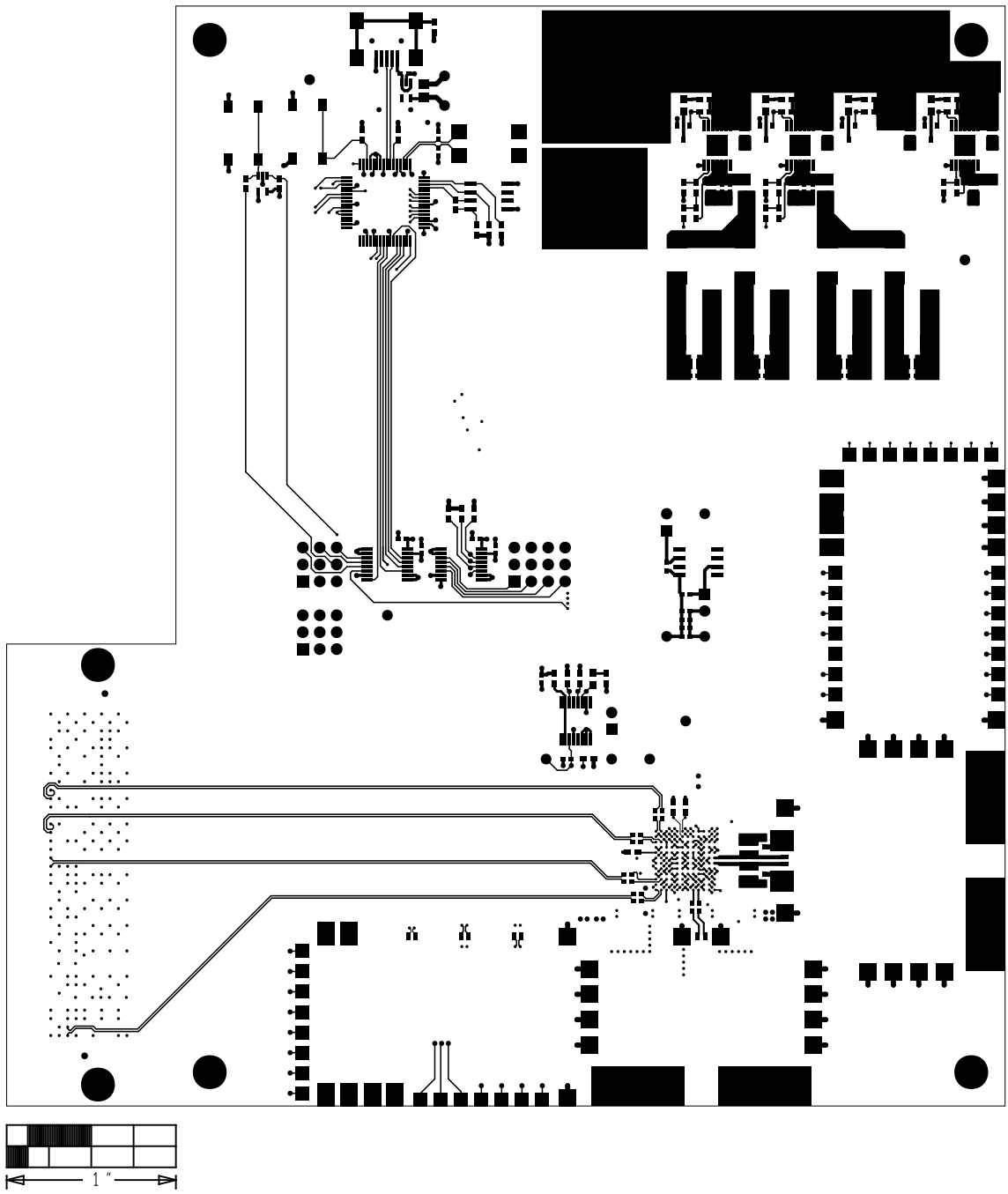
REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	5/16	Initial release	—

For pricing, delivery, and ordering information, please contact Maxim Direct at 1-888-629-4642, or visit Maxim Integrated's website at www.maximintegrated.com.

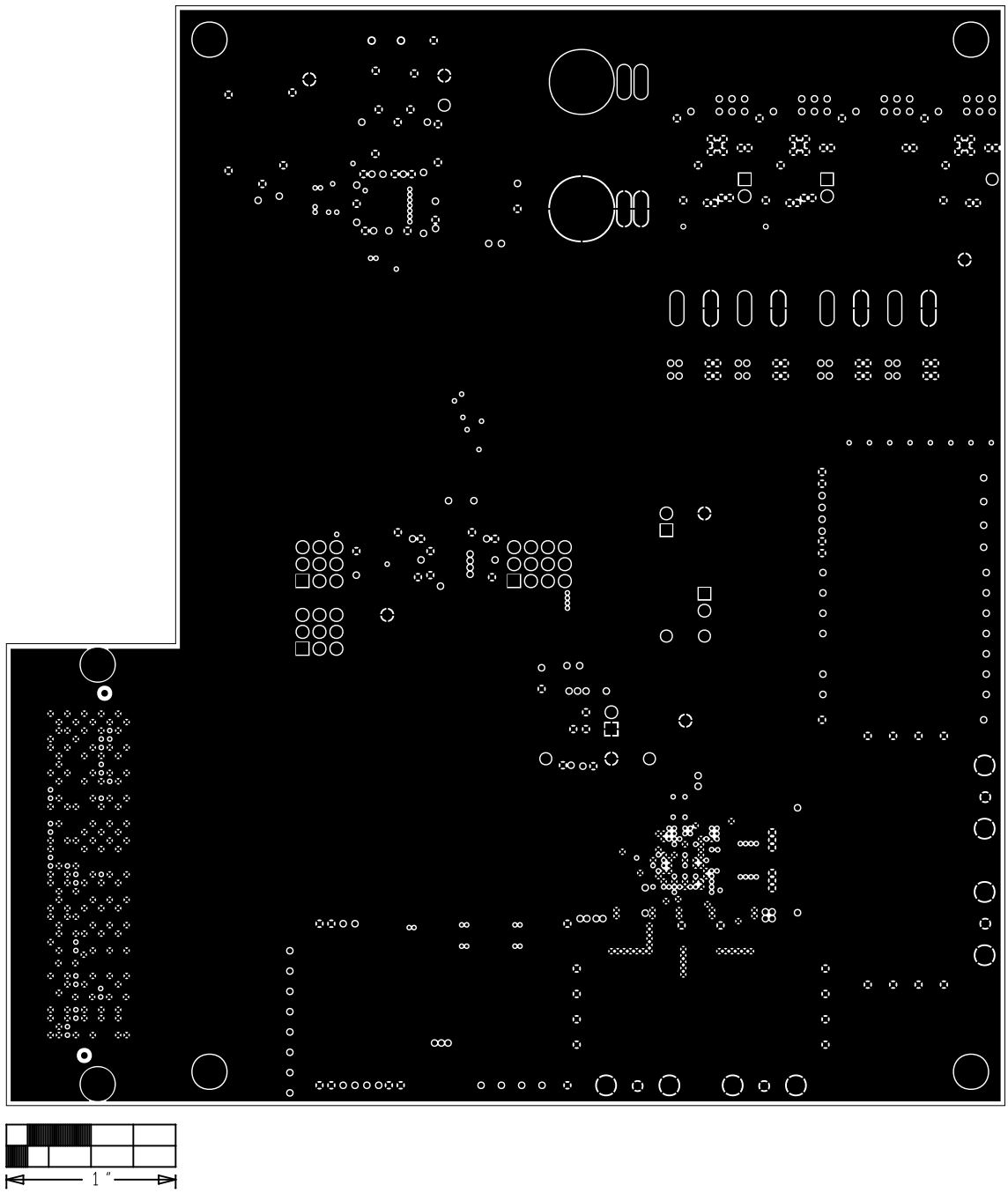
Maxim Integrated cannot assume responsibility for use of any circuitry other than circuitry entirely embodied in a Maxim Integrated product. No circuit patent licenses are implied. Maxim Integrated reserves the right to change the circuitry and specifications without notice at any time.



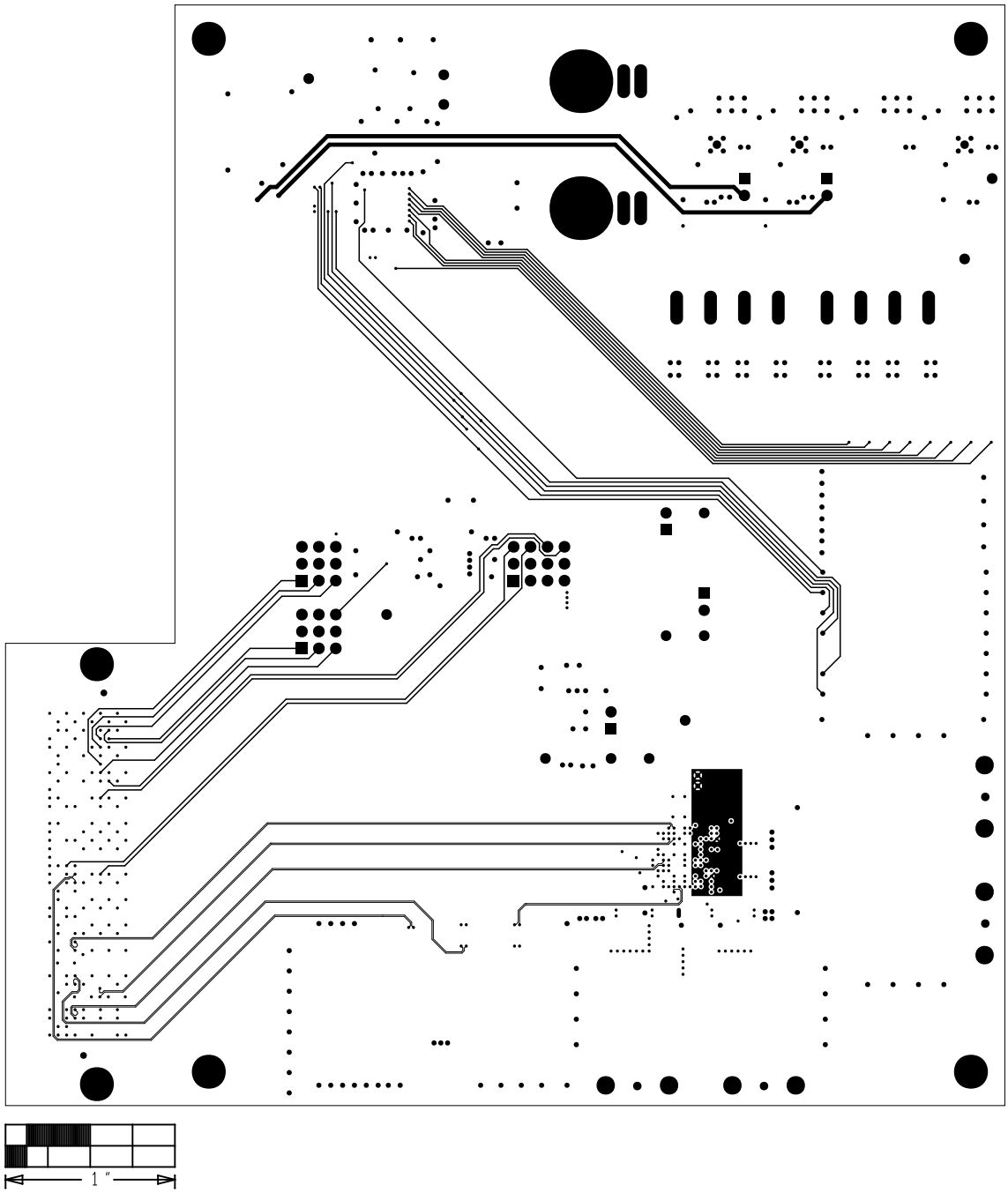
Silkscreen Top



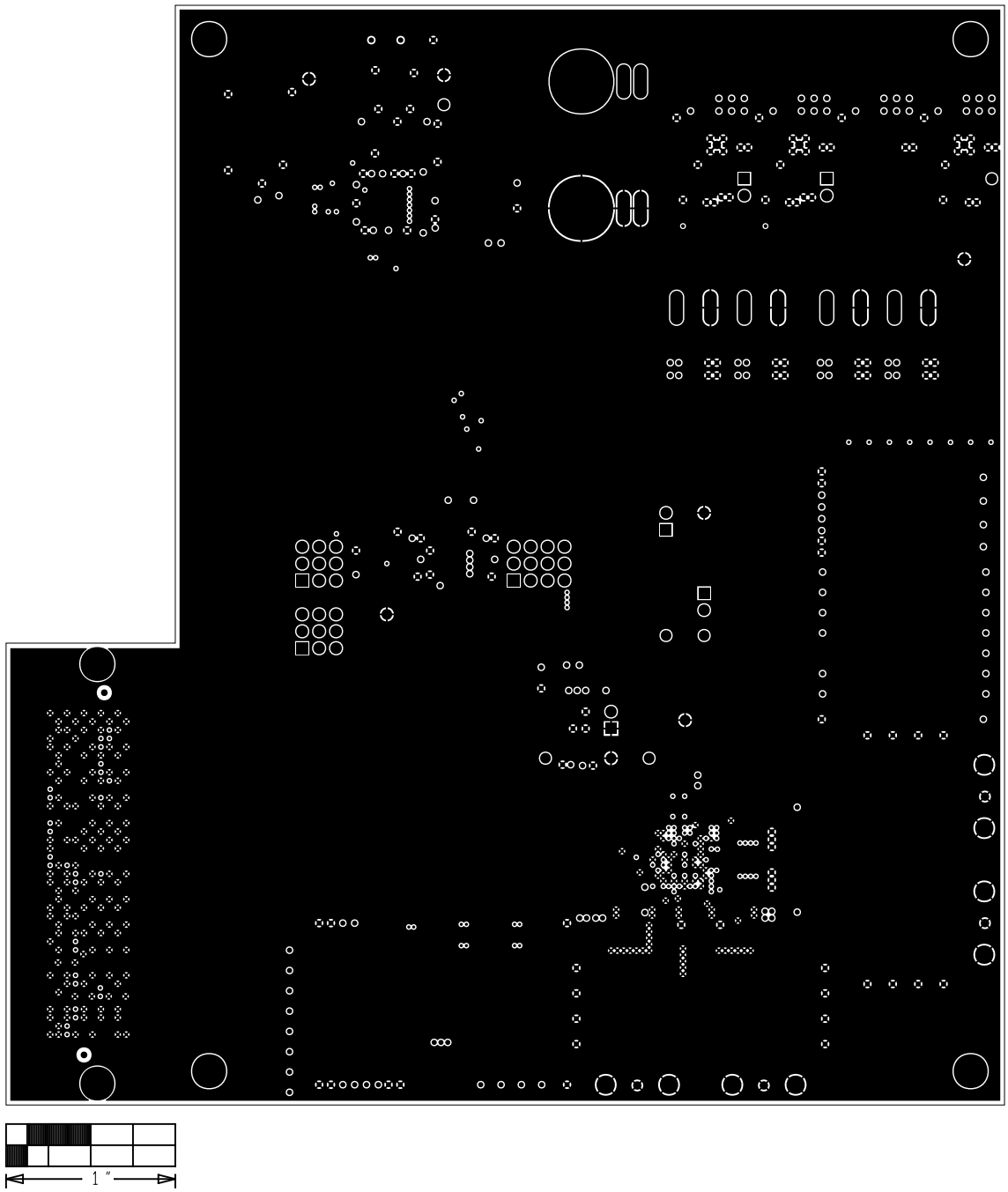
Top



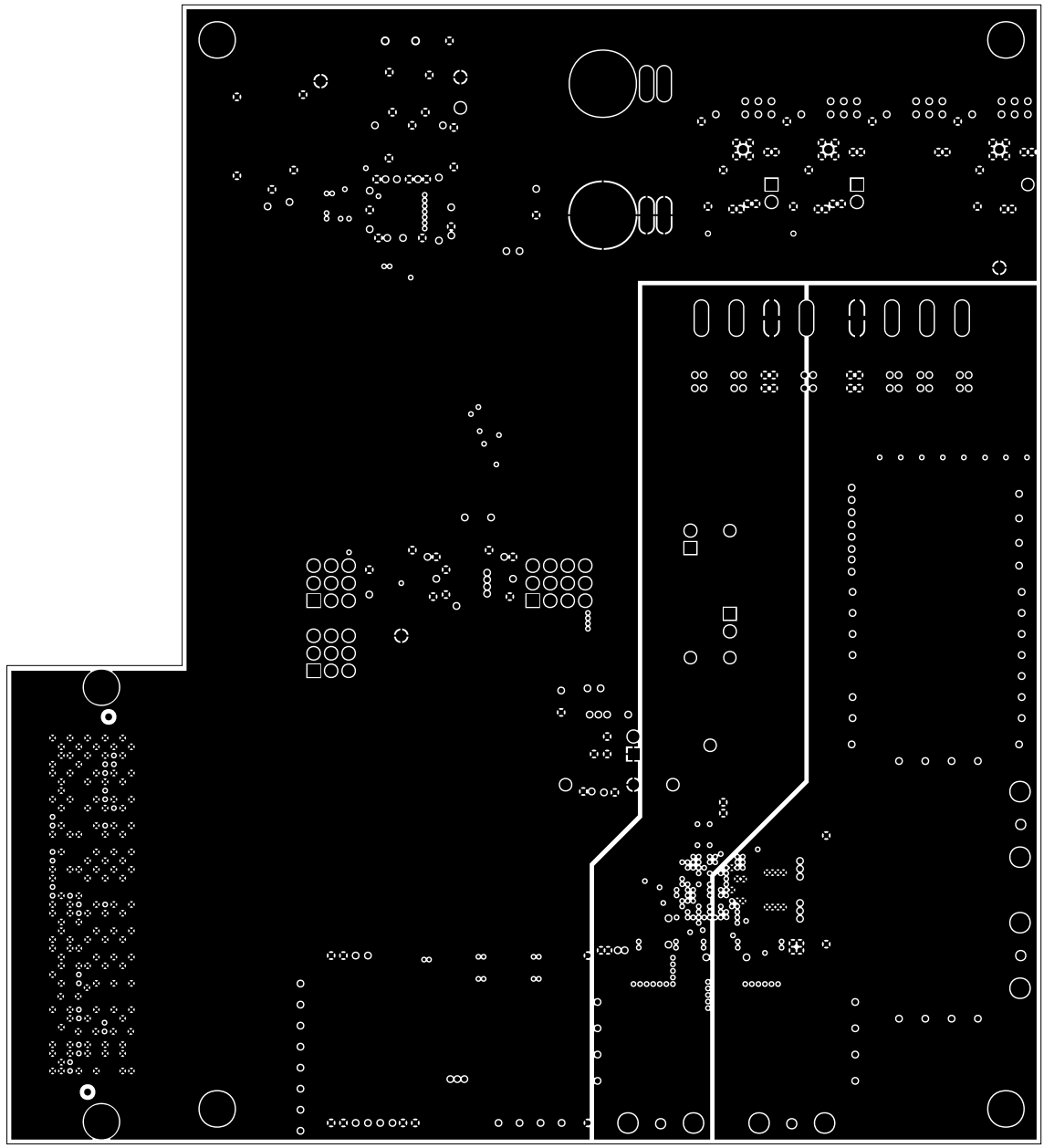
Level 2 GND



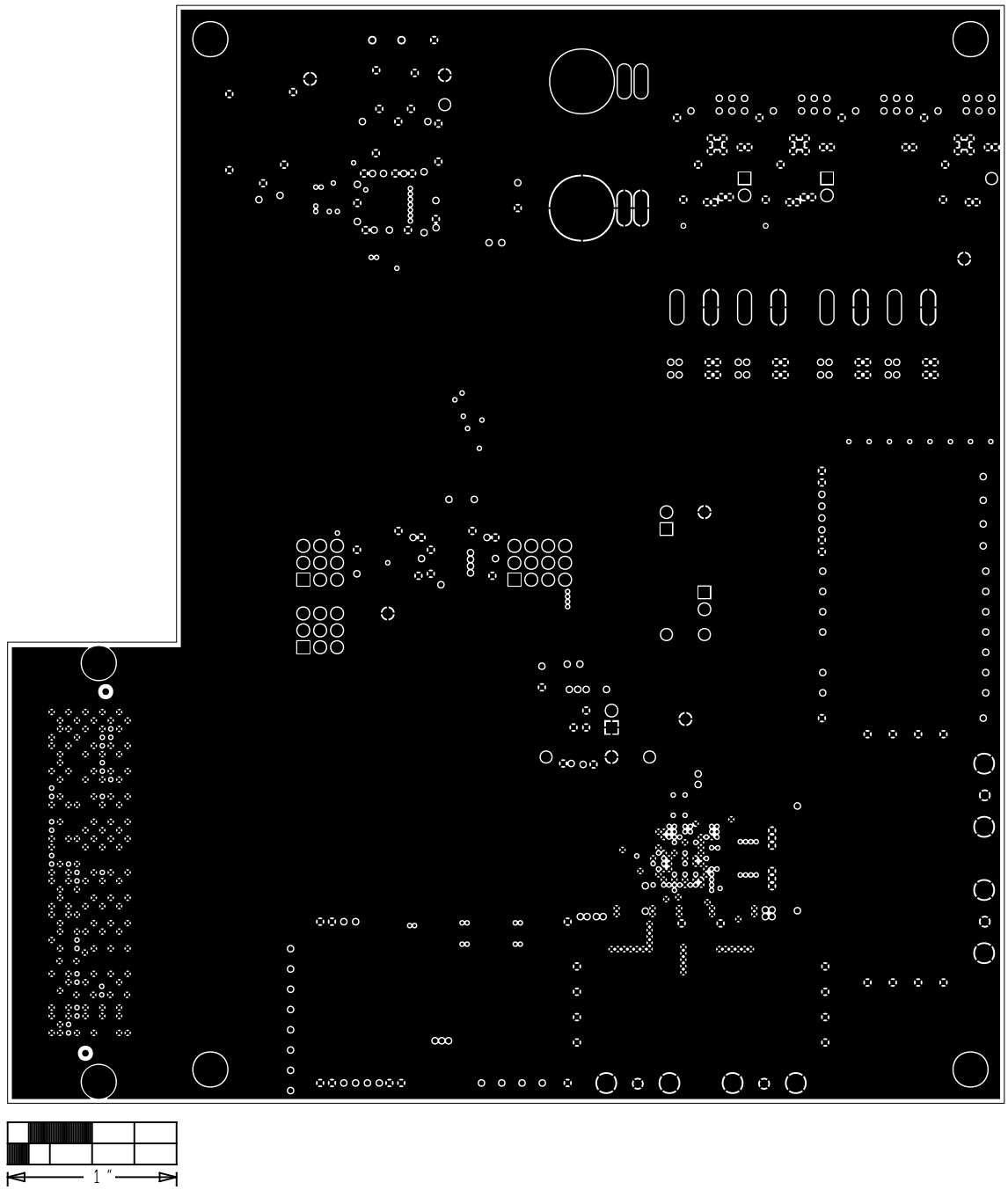
Level 3 Signal 1



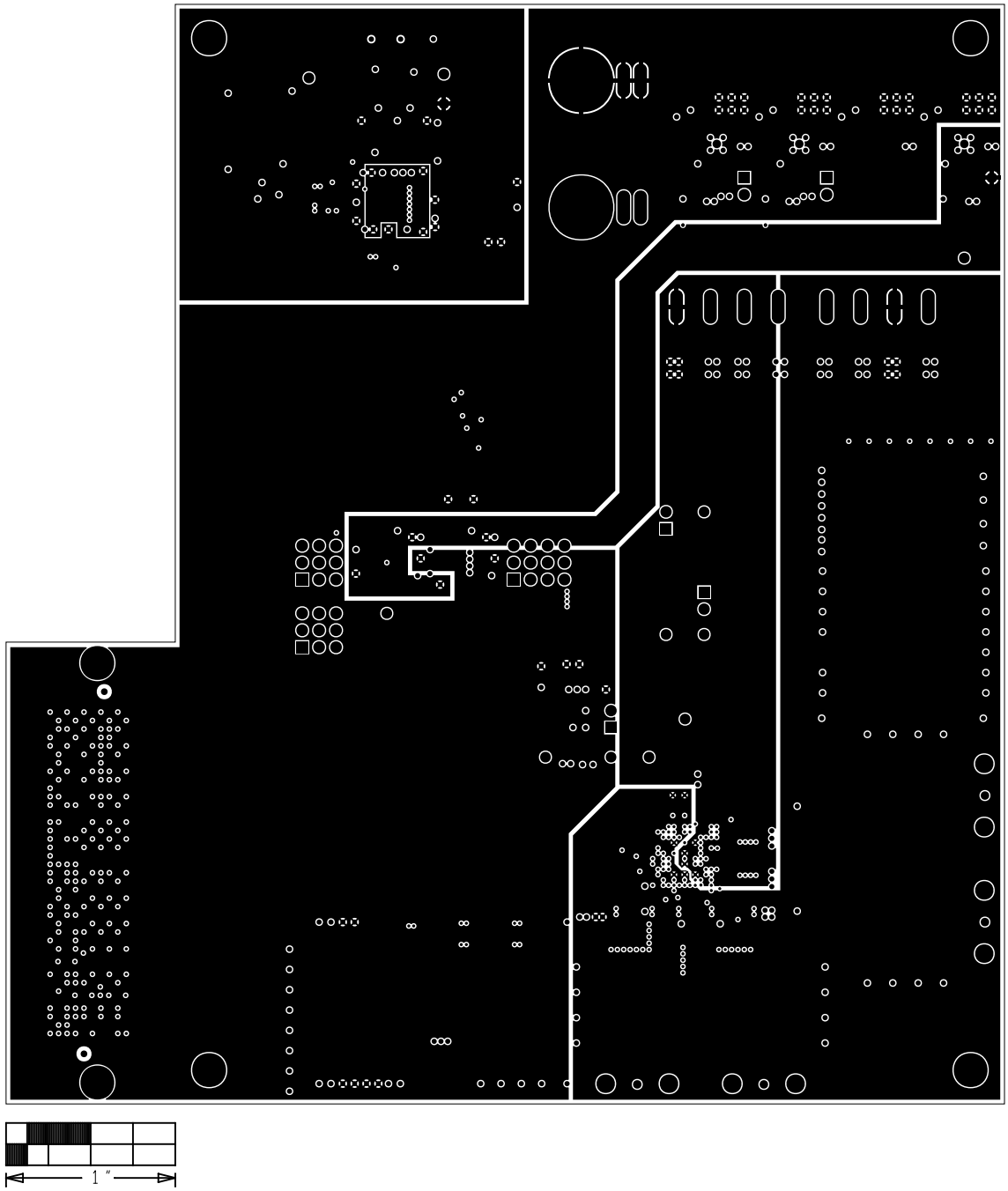
Level 4 GND



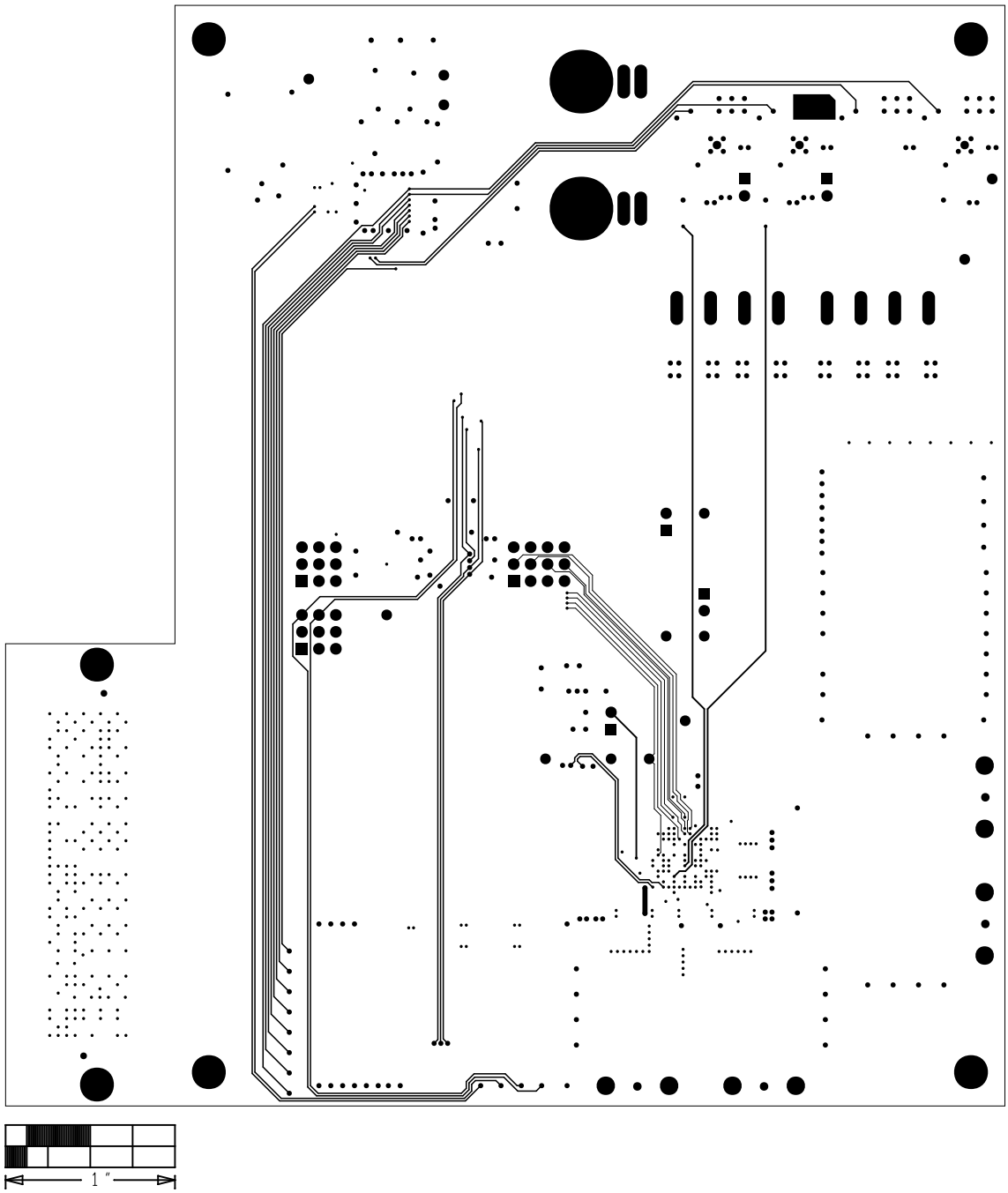
Level 5 Power



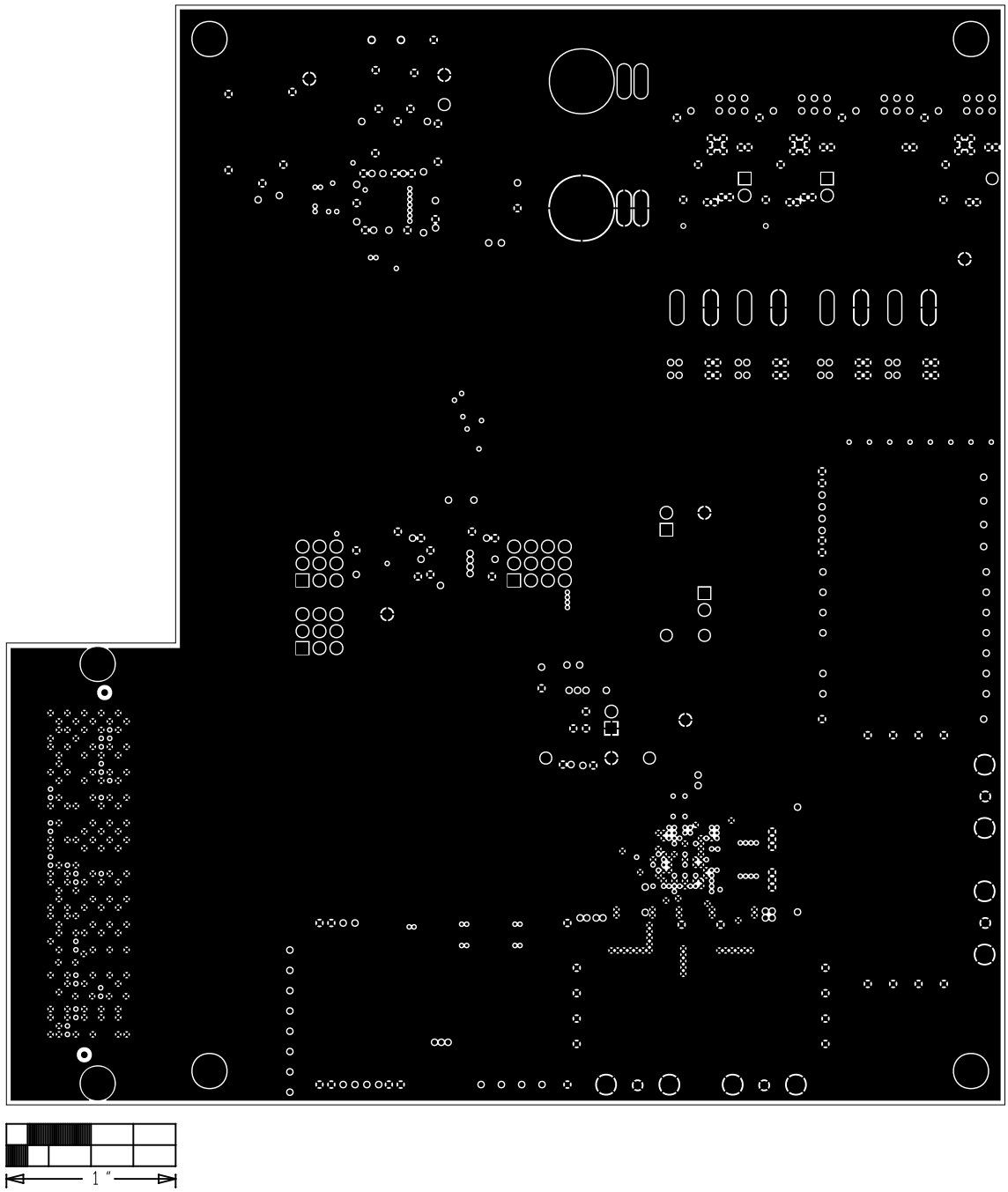
Level 6 GND



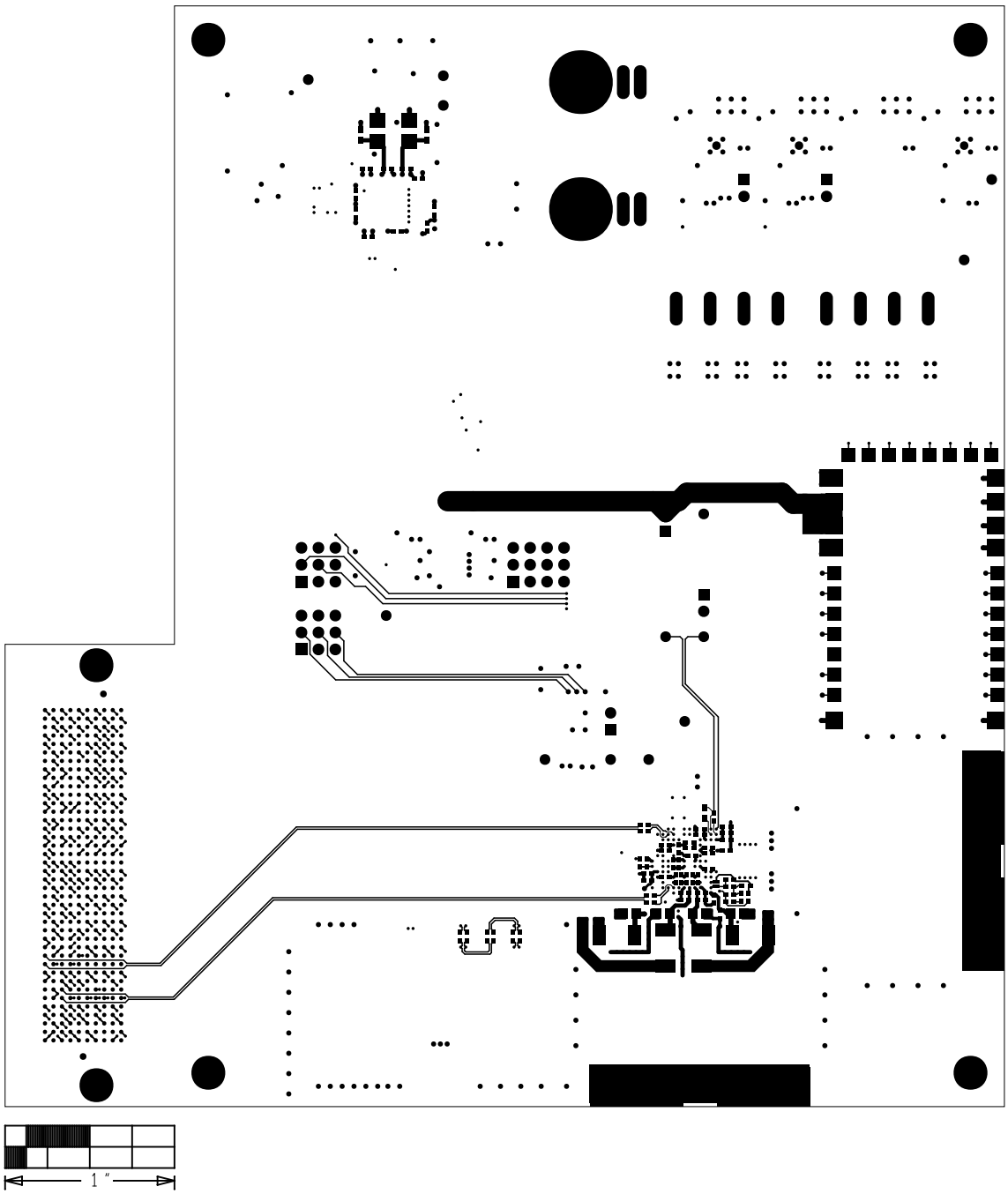
Level 7 Power



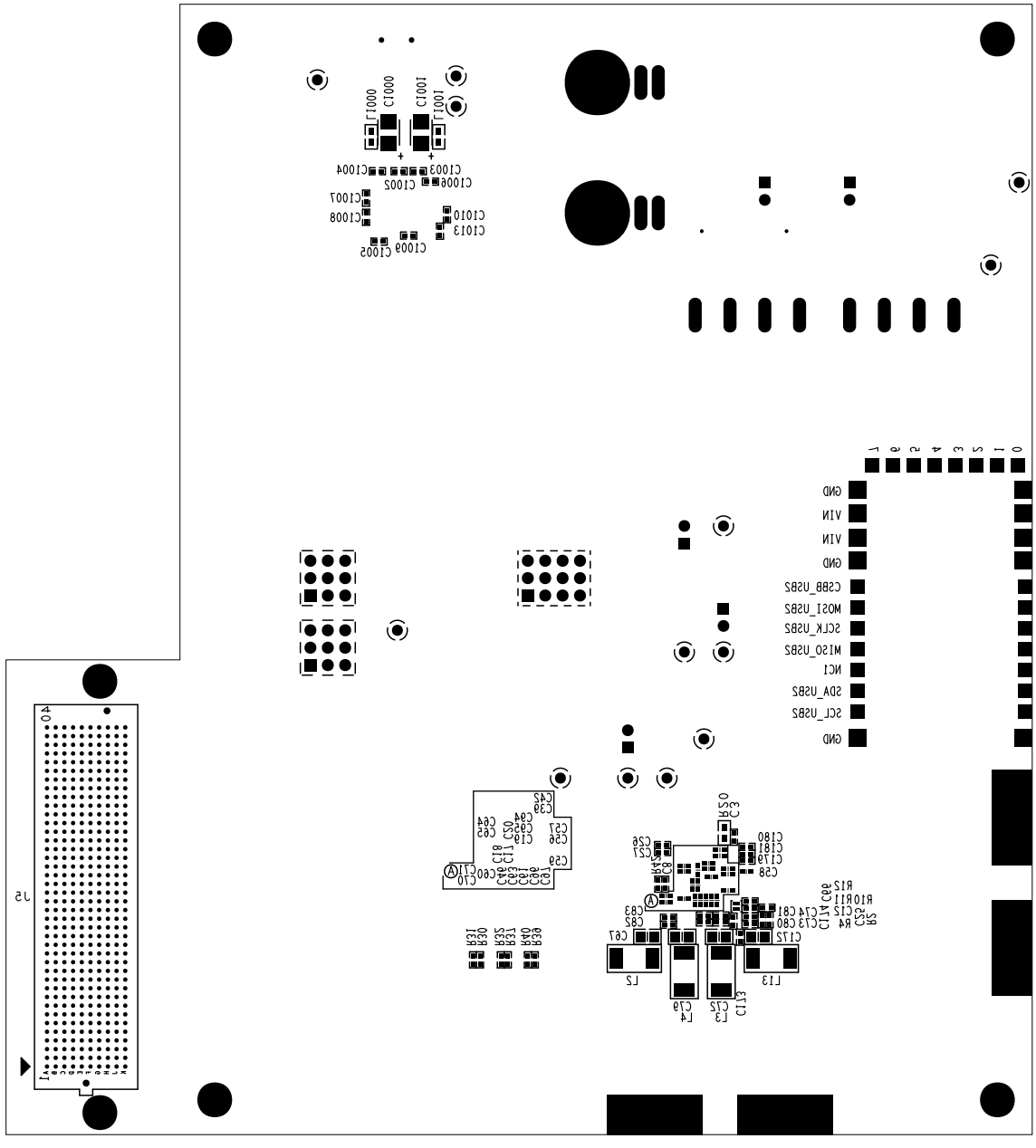
Level 8 Signal 2



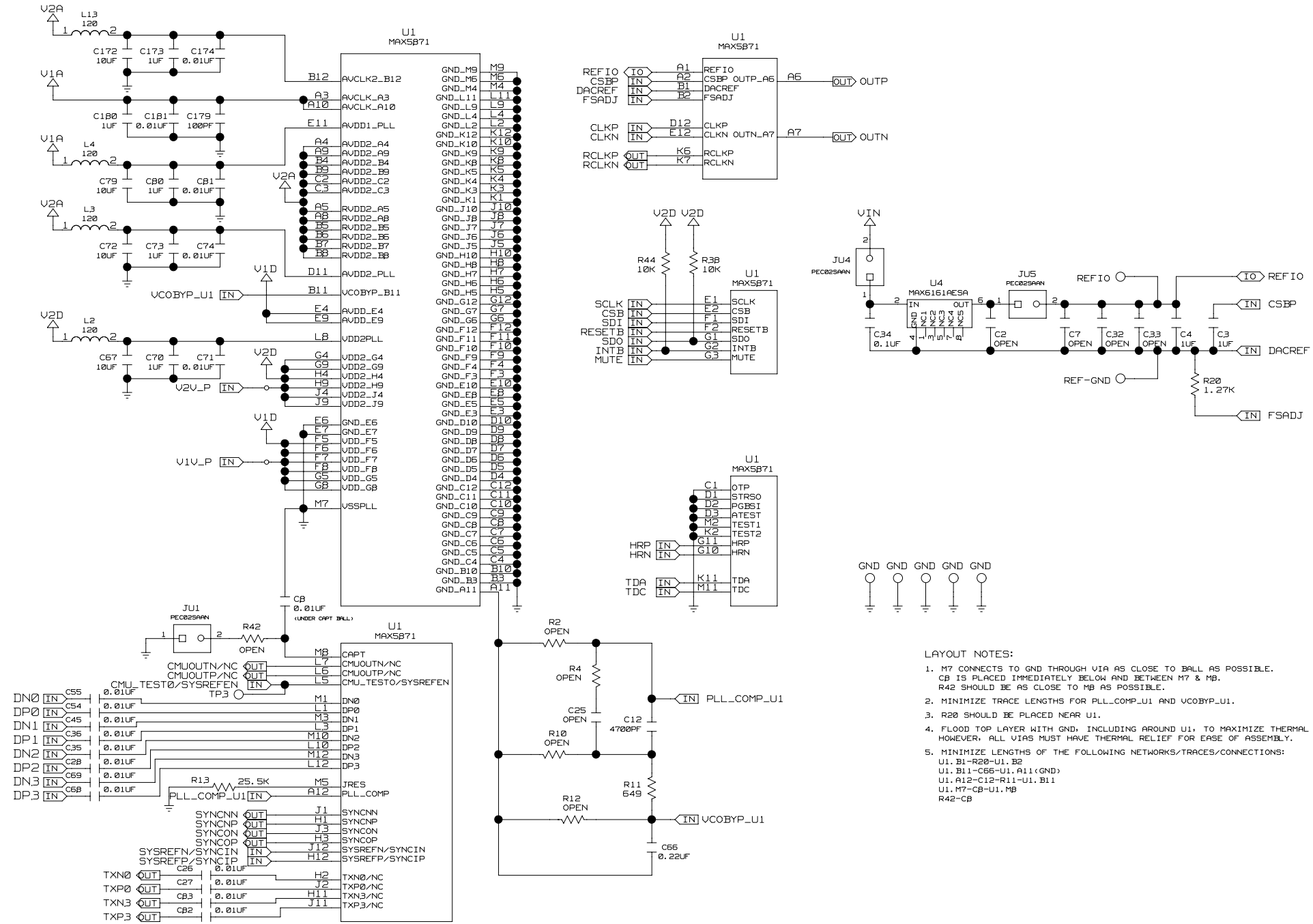
Level 9 GND



Bottom

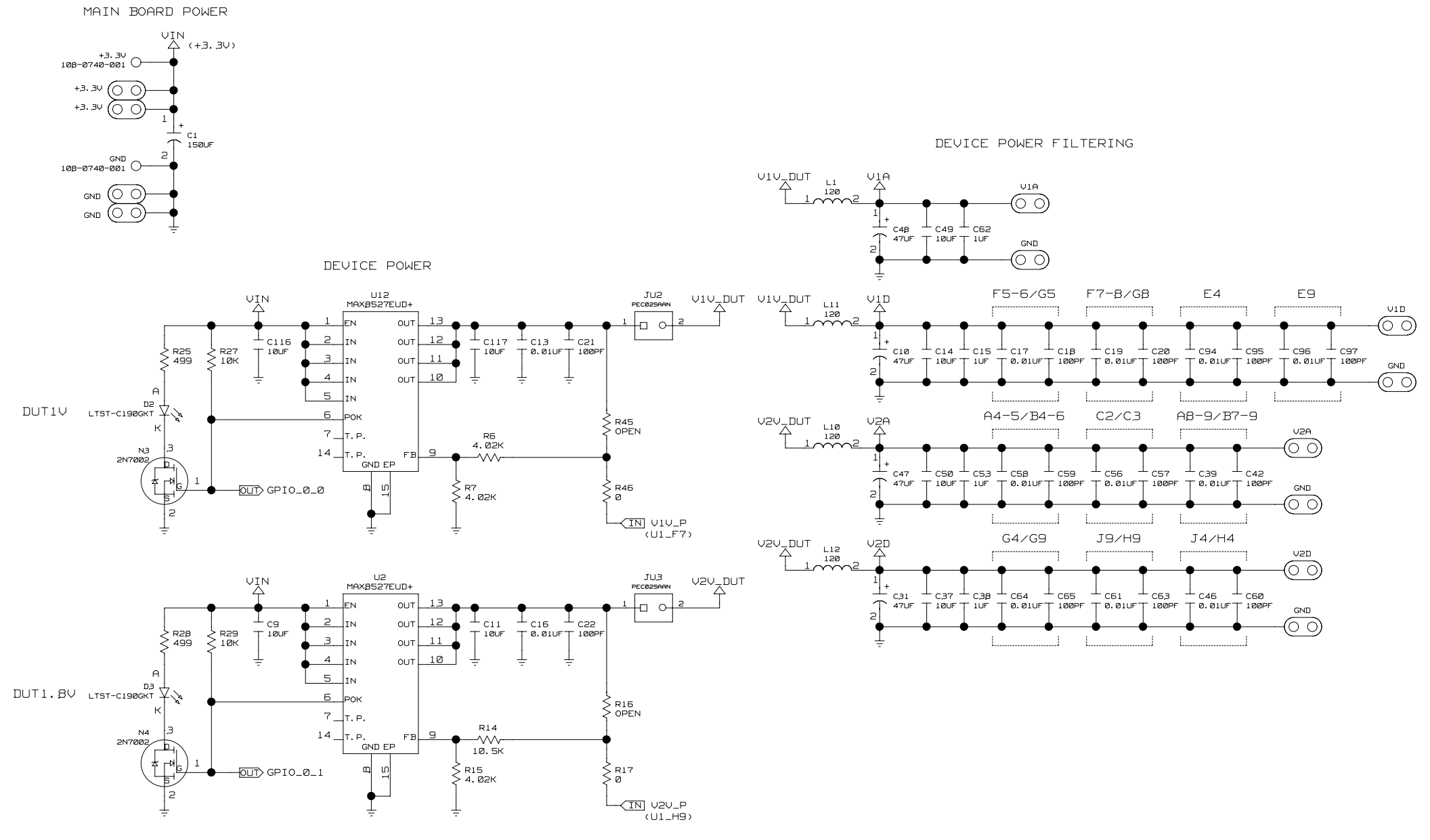


Silkscreen Bottom

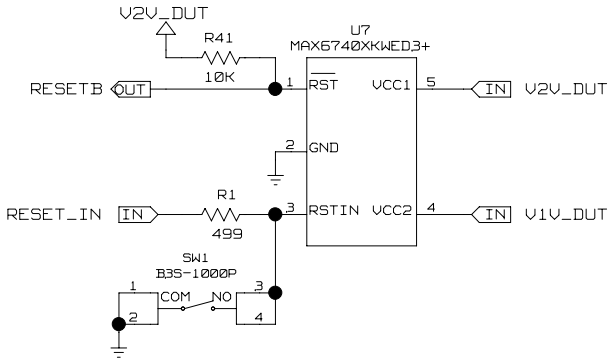


LAYOUT NOTES:

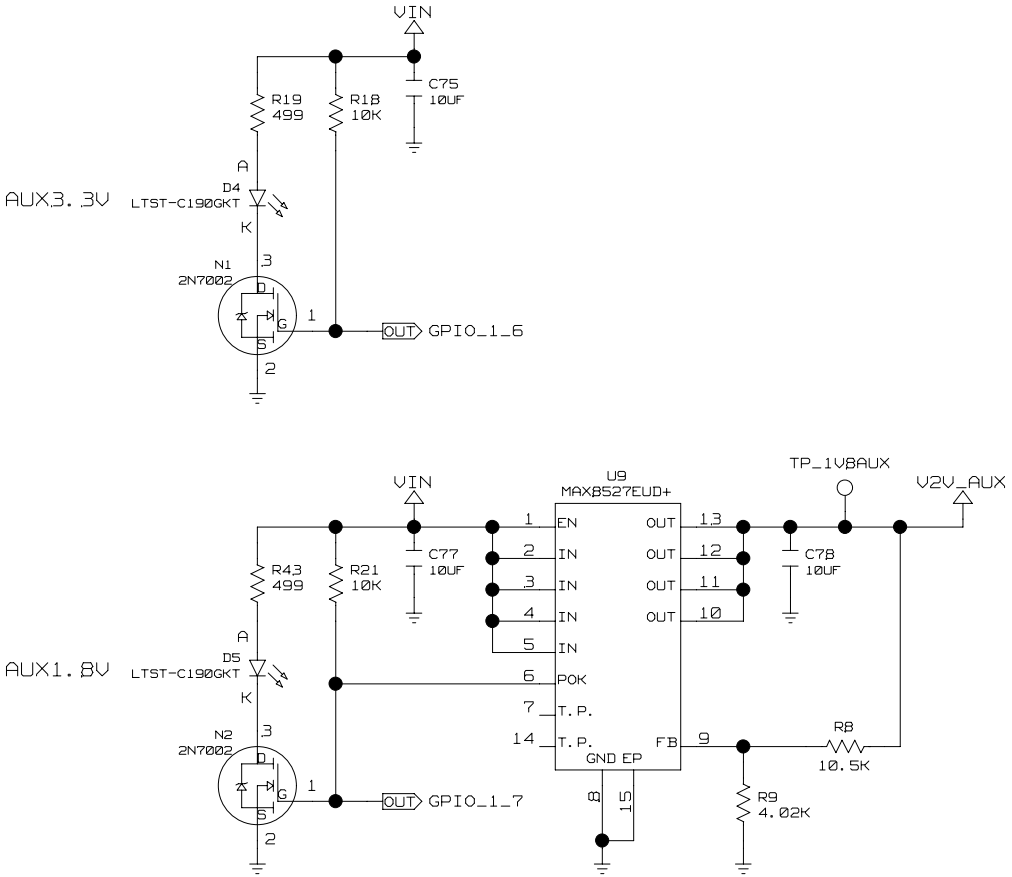
1. M7 CONNECTS TO GND THROUGH VIA AS CLOSE TO BALL AS POSSIBLE.
   C8 IS PLACED IMMEDIATELY BELOW AND BETWEEN M7 & M8.
   R42 SHOULD BE AS CLOSE TO M8 AS POSSIBLE.
2. MINIMIZE TRACE LENGTHS FOR PLL_COMP_U1 AND VCOBYP_U1.
3. R20 SHOULD BE PLACED NEAR U1.
4. FLOOD TOP LAYER WITH GND, INCLUDING AROUND U1, TO MAXIMIZE THERMAL CONDUCTIVITY FOR U1. HOWEVER, ALL VIAS MUST HAVE THERMAL RELIEF FOR EASE OF ASSEMBLY.
5. MINIMIZE LENGTHS OF THE FOLLOWING NETWORKS/TRACES/CONNECTIONS:
   U1.B1-R20-U1.B2
   U1.B11-C66-U1.A11(GND)
   U1.A12-C12-R11-U1.B11
   U1.M7-C8-U1.M8
   R42-C8


MAIN BOARD POWER
VIN
(+3.3V)
+3.3V
10B-0740-001
+3.3V
+3.3V
C1
150UF
GND
10B-0740-001
GND
GND
DEVICE POWER
VIN
U12
MAX8527EUD+
JU2
PEC02SAAN
V1V_DUT
C116
10UF
C117
10UF
C13
0.01UF
C21
100PF
R25
499
R27
10K
D2
LTST-C190GKT
DUT1V
N3
2N7002
OUT
GPIO_0_0
R6
4.02K
R7
4.02K
R45
OPEN
R46
0
IN
V1V_P
(U1_F7)
VIN
U2
MAX8527EUD+
JU3
PEC02SAAN
V2V_DUT
C9
10UF
C11
10UF
C16
0.01UF
C22
100PF
R28
499
R29
10K
D3
LTST-C190GKT
DUT1.8V
N4
2N7002
OUT
GPIO_0_1
R14
10.5K
R15
4.02K
R16
OPEN
R17
0
IN
V2V_P
(U1_H9)
DEVICE POWER FILTERING
V1V_DUT
L1
120
V1A
C48
47UF
C49
10UF
C62
1UF
GND
V1V_DUT
L11
120
V1D
F5-6/G5
F7-8/G8
E4
E9
C10
47UF
C14
10UF
C15
1UF
C17
0.01UF
C18
100PF
C19
0.01UF
C20
100PF
C94
0.01UF
C95
100PF
C96
0.01UF
C97
100PF
V1D
GND
V2V_DUT
L10
120
V2A
A4-5/B4-6
C2/C3
A8-9/B7-9
C47
47UF
C50
10UF
C53
1UF
C58
0.01UF
C59
100PF
C56
0.01UF
C57
100PF
C39
0.01UF
C42
100PF
V2A
GND
V2V_DUT
L12
120
V2D
G4/G9
J9/H9
J4/H4
C31
47UF
C37
10UF
C38
1UF
C64
0.01UF
C65
100PF
C61
0.01UF
C63
100PF
C46
0.01UF
C60
100PF
V2D
GND

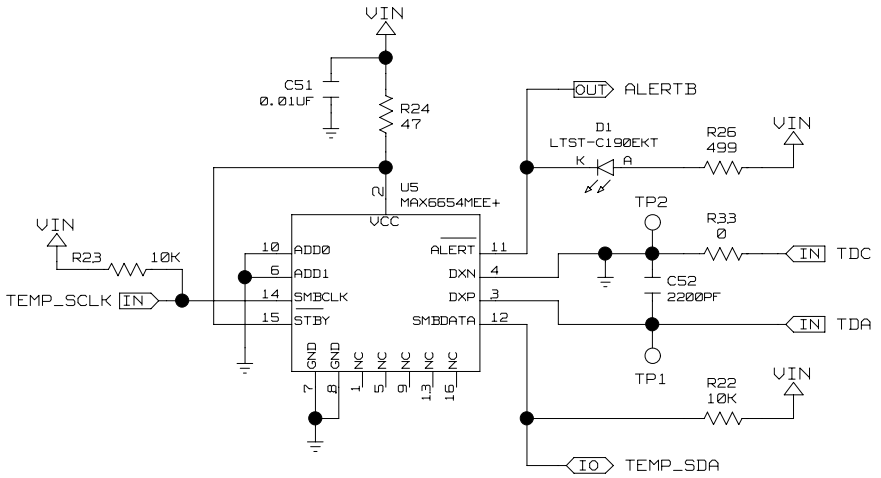
POWER ON RESET CIRCUIT



AUXILIARY POWER



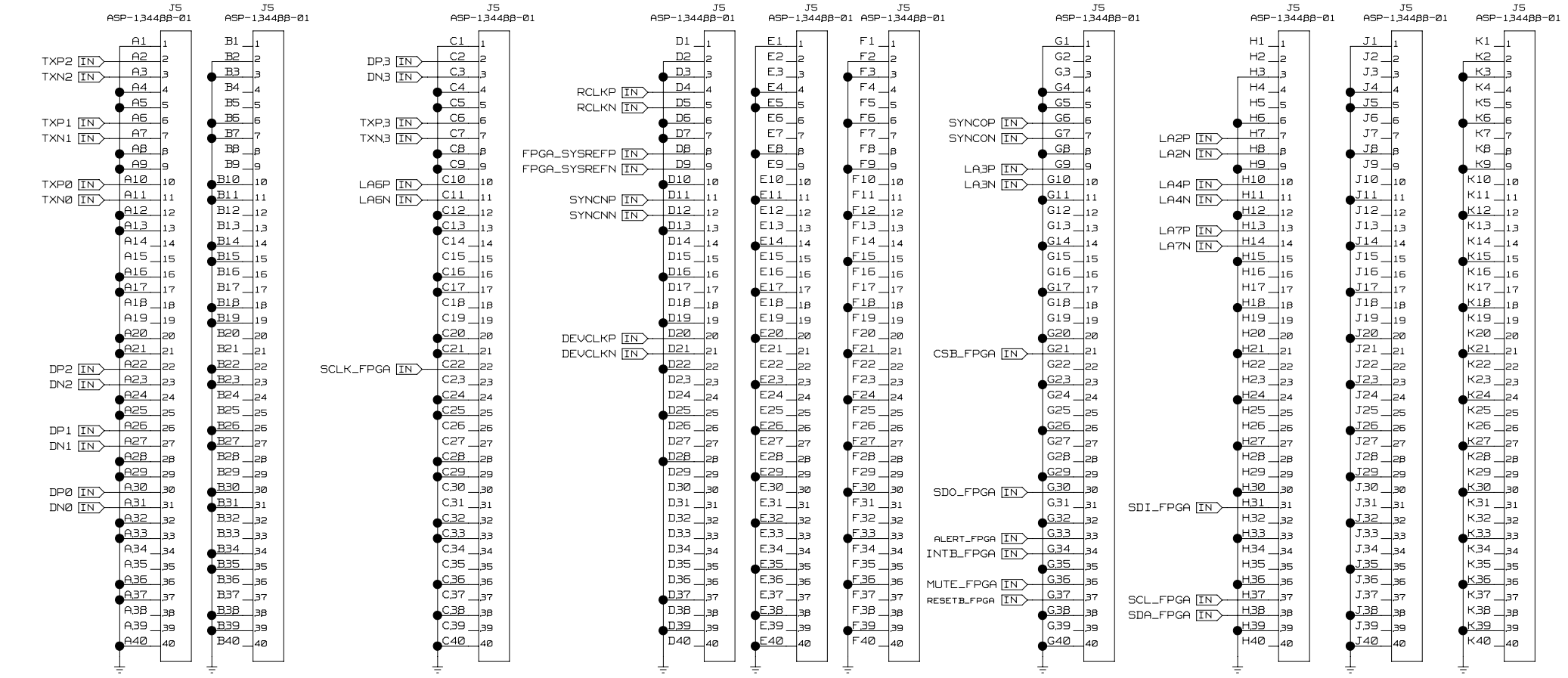
TEMPERATURE MONITOR



LAYOUT NOTES:

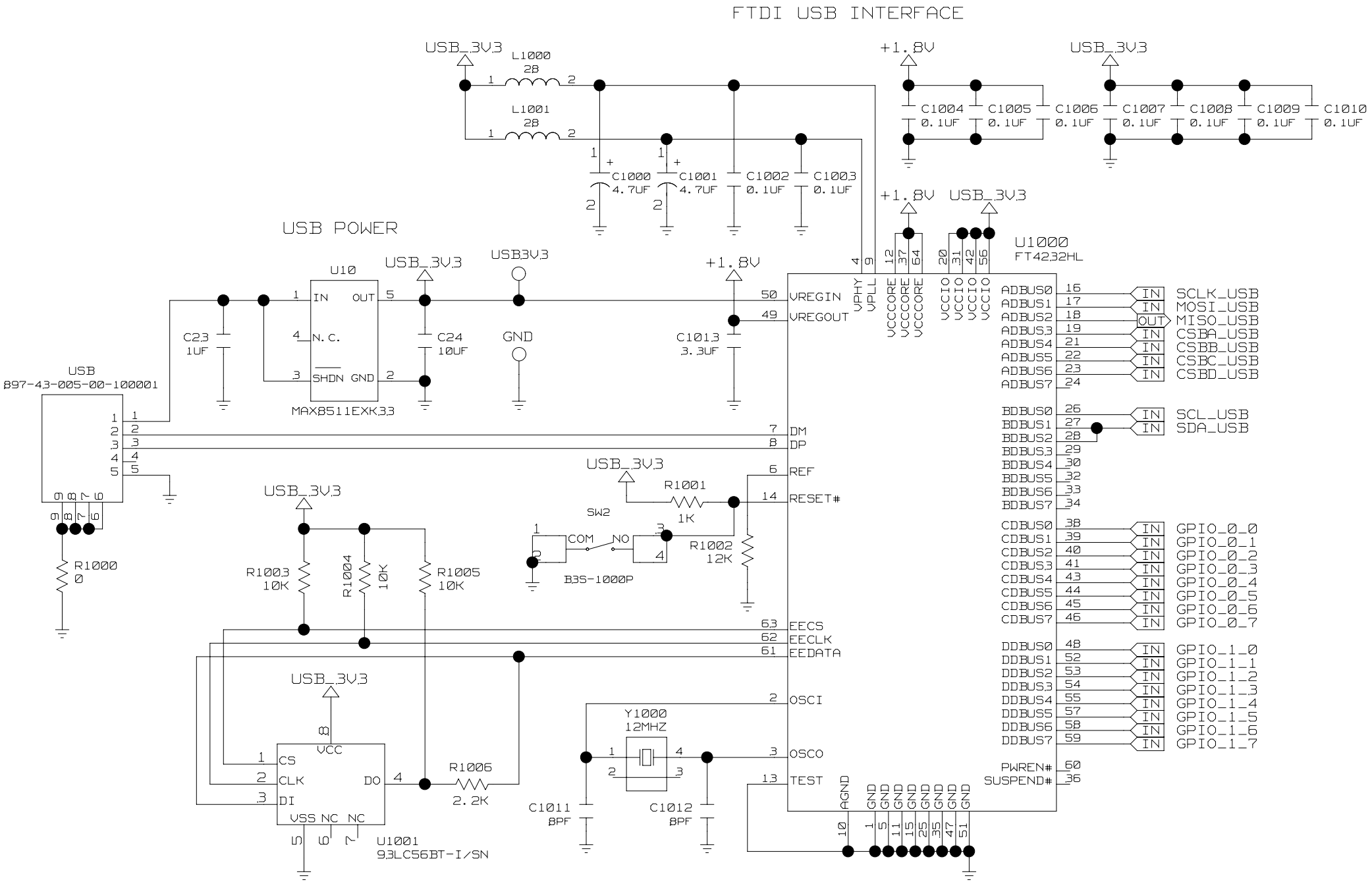
1. PLACE U5 CLOSE TO U1 TO MINIMIZE TRACES FOR TDC AND TDA.

DATA INTERFACE

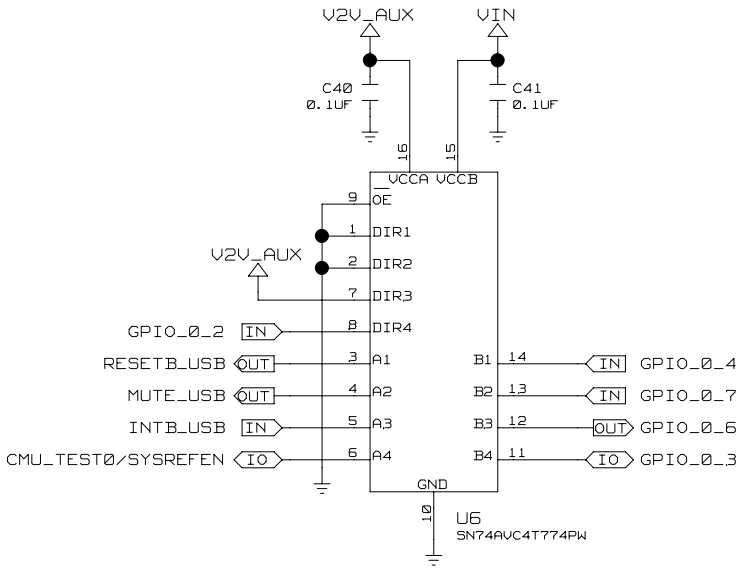


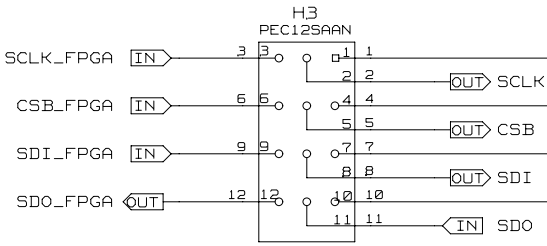
INSTALL 0 OHM RESISTORS TO CONNECT ONE OF THE FOLLOWING OPTIONS:

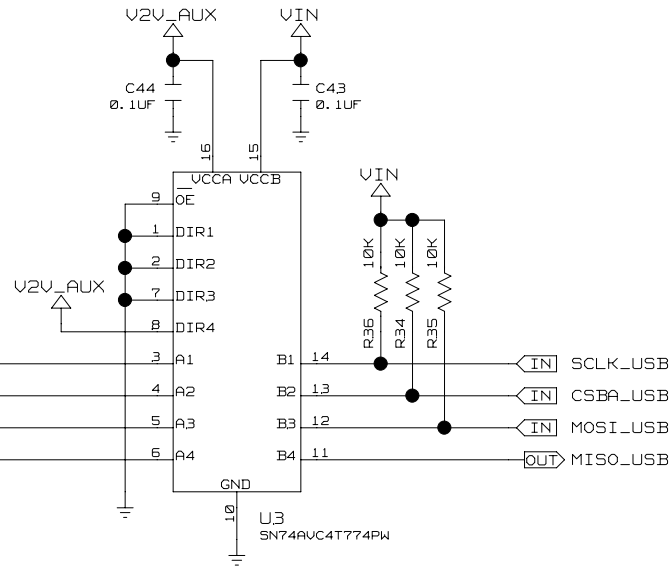
1) SYSREF__B CONNECTED TO FPGA_SYSREF AND SYSREF__A TO SYSREF_/SYNCI_
2) CONNECT SYSREF_/SYNCI_ TO FPGA_SYSREF, SYSREF__A AND B ARE LEFT FLOATING (DEFAULT)

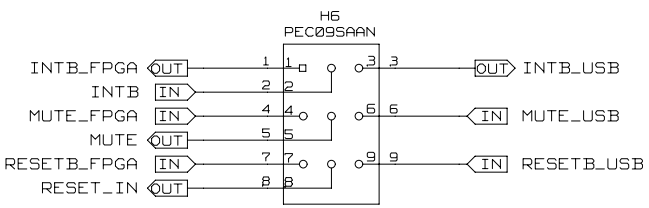

FTDI USB INTERFACE
USB_3V3
L1000
28
L1001
28
+1.8V
C1004
0.1UF
C1005
0.1UF
C1006
0.1UF
C1007
0.1UF
C1008
0.1UF
C1009
0.1UF
C1010
0.1UF
C1000
4.7UF
C1001
4.7UF
C1002
0.1UF
C1003
0.1UF
USB POWER
U10
C23
1UF
C24
10UF
GND
MAX8511EXK33
USB
897-43-005-00-100001
C1013
3.3UF
U1000
FT4232HL
VREGIN
VREGOUT
VPHY
VPLL
VCCCORE
VCCIO
DM
DP
REF
RESET#
SW2
B3S-1000P
R1000
0
R1001
1K
R1002
12K
R1003
10K
R1004
10K
R1005
10K
EECS
EECLK
EEDATA
OSCI
OSCO
TEST
Y1000
12MHZ
C1011
8PF
C1012
8PF
R1006
2.2K
U1001
93LC56BT-I/SN
ADBUS0
ADBUS1
ADBUS2
ADBUS3
ADBUS4
ADBUS5
ADBUS6
ADBUS7
SCLK_USB
MOSI_USB
MISO_USB
CSBA_USB
CSBB_USB
CSBC_USB
CSBD_USB
BDBUS0
BDBUS1
SCL_USB
SDA_USB
CDBUS0
CDBUS1
CDBUS2
CDBUS3
CDBUS4
CDBUS5
CDBUS6
CDBUS7
GPIO_0_0
GPIO_0_1
GPIO_0_2
GPIO_0_3
GPIO_0_4
GPIO_0_5
GPIO_0_6
GPIO_0_7
DDBUS0
DDBUS1
DDBUS2
DDBUS3
DDBUS4
DDBUS5
DDBUS6
DDBUS7
GPIO_1_0
GPIO_1_1
GPIO_1_2
GPIO_1_3
GPIO_1_4
GPIO_1_5
GPIO_1_6
GPIO_1_7
PWREN#
SUSPEND#
AGND
GND

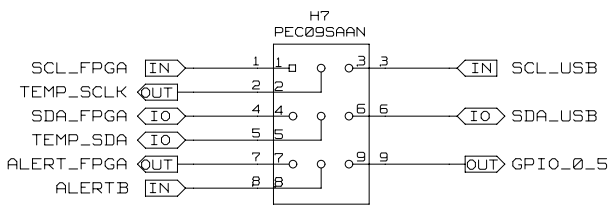
LEVEL TRANSLATORS



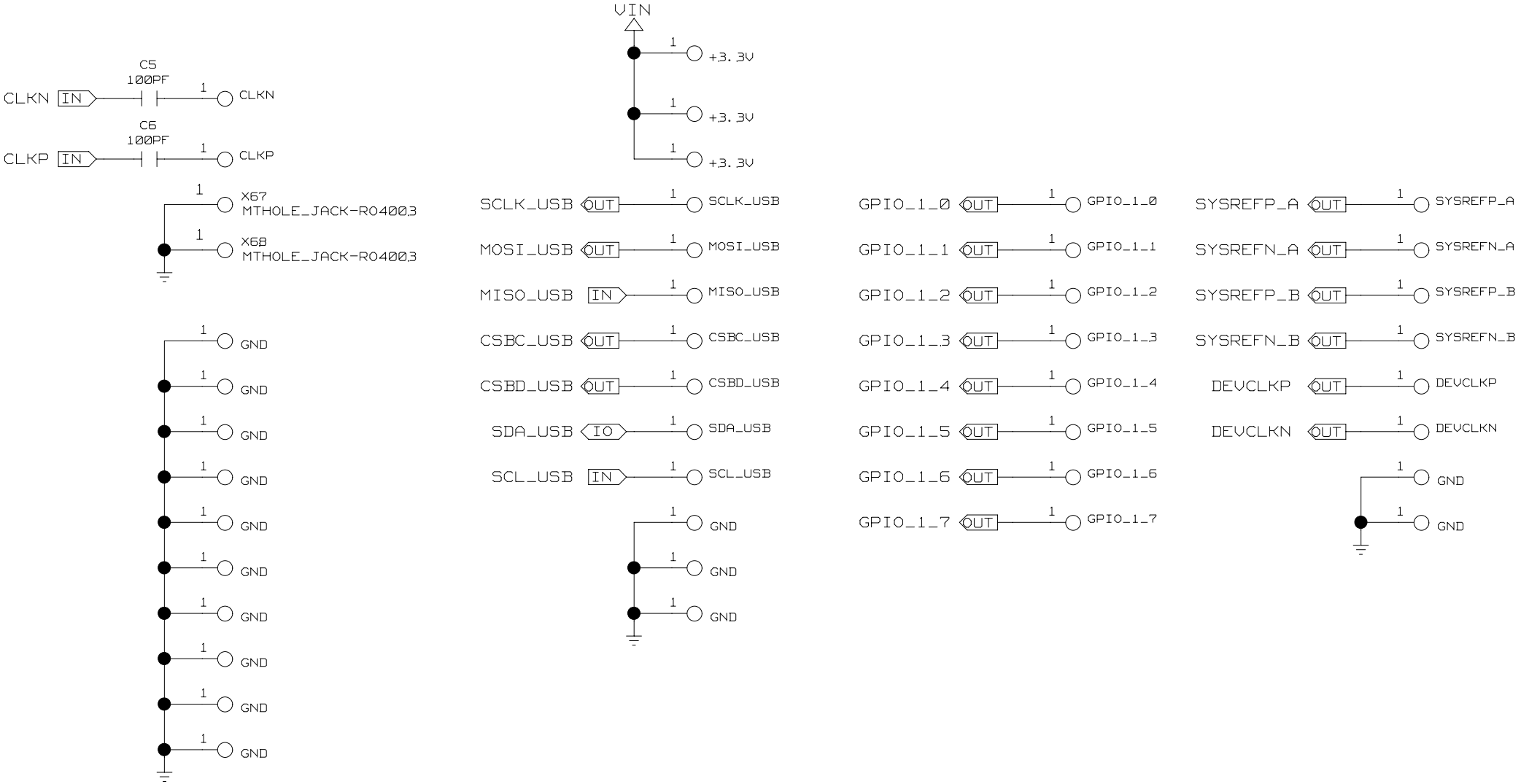




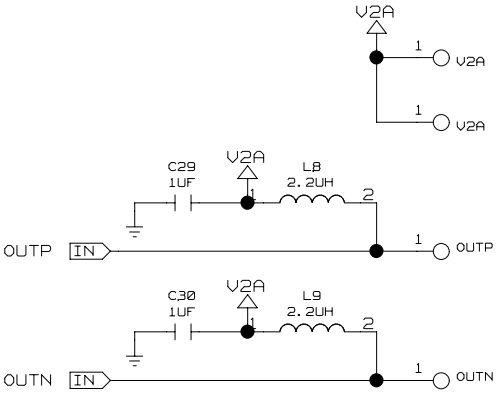


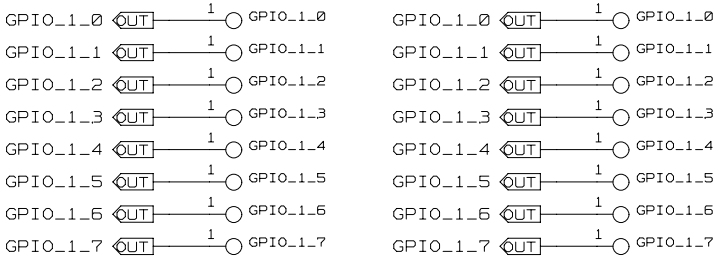


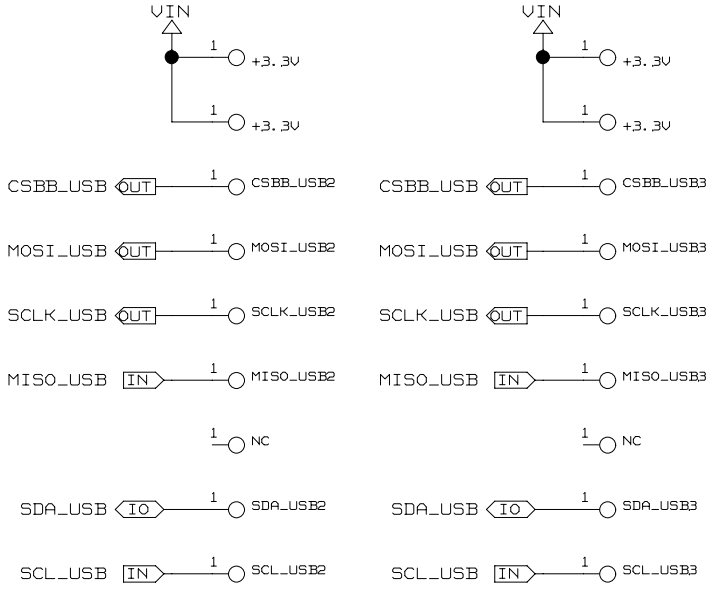
CLK MODULE INTERCONNECT

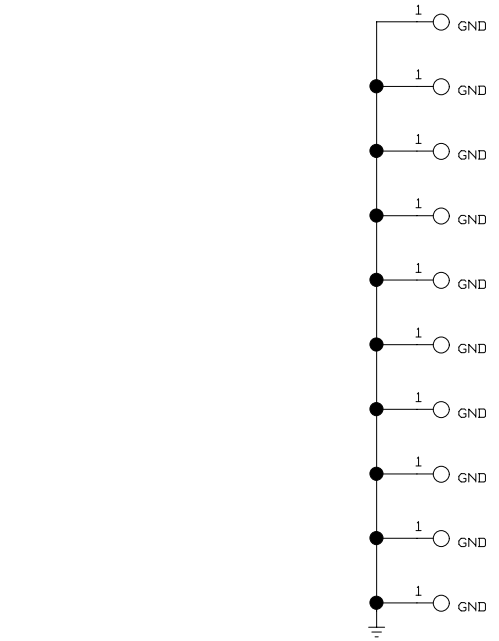


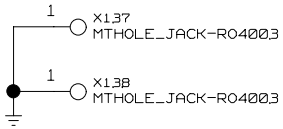
DAC OUT MODULE INTERCONNECT

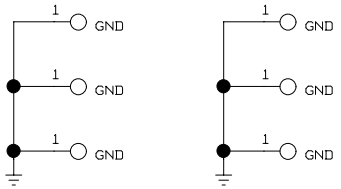


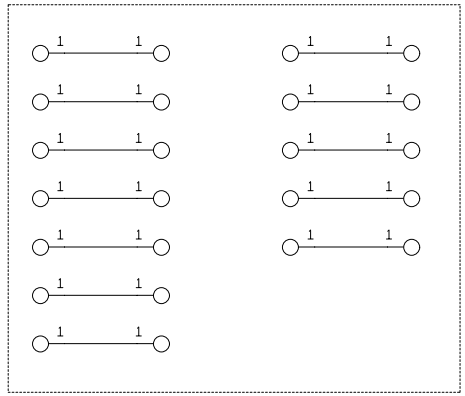












FOR EXPANSION PURPOSES