TEXAS INSTRUMENTS

# THS1408/5691 EVM for the THS14xx and THS56xx DAC Families

## User's Guide

July 2000 AAP Data Conversion

SLAU045

# IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgment, including those pertaining to warranty, patent infringement, and limitation of liability.

TI warrants performance of its semiconductor products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Customers are responsible for their applications using TI components.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty or endorsement thereof.

Copyright © 2000, Texas Instruments Incorporated

## Preface

# Read This First

### About This Manual

The purpose of this user's guide is to serve as a reference manual for the THS1408 14-bit ADC, analog-to-digital converter module (EVM). This document provides information to assist hardware and software engineers in application development.

### How to Use This Manual

- ☐ Chapter 1 – Overview
- ☐ Chapter 2 – Physical Description
- ☐ Chapter 3 – Circuit Description

### Information About Cautions and Warnings

This book may contain cautions and warnings.

> **CAUTION**
>
> **This is an example of a caution statement.**
>
> **A caution statement describes a situation that could potentially damage your software or equipment.**

> **WARNING**
>
> **This is an example of a warning statement.**
>
> **A warning statement describes a situation that could potentially cause harm to you.**

The information in a caution or a warning is provided for your protection. Please read each caution and warning carefully.

# Contents

# Figures

# Tables

## Chapter 1

# Overview

This chapter presents a general overview of the THS1408_5691EVM evaluation module (EVM), and describes some of the factors that must be considered in using the module.

**Topic** **Page**

## 1.1 Purpose

The THS1408_5691EVM evaluation module (EVM) provides a platform for evaluating the THS14XX analog-to-digital converter (ADC) and the THS56XX digital-to-analog converter (DAC) under various signal, reference, and supply conditions.

**Note:**

The EVM in its current implementation supports only the THS14XX device family and associated circuitry. Support for the THS56XX family will become available at a later date. As a result of this, only the operation of the THS14XX and its associated circuitry will be described in this user's guide.

## 1.2 EVM Basic Functions

Analog input to the ADC is provided via two external SMB connectors. This input can be configured onboard to be true differential or single-ended transformer-coupled to the input of the device.

The EVM provides an external SMB connection for ADC clock input. This can be configured to be either ac- or dc-coupled. A site provided on the board for a crystal oscillator to perform this function can be populated if required. Further provision is made to run the ADC from a DSP timer if desired.

Output from the EVM takes place via two 80-pin daughtercard connectors. The digital lines from the ADC are buffered before going to the daughtercard connectors. More information on these connectors can be found in the TMS320C6000 daughtercard specification.

The EVM is powered via 4-mm banana sockets. Separate input connectors are provided for the analog and digital supplies. Provision is made to supply the EVM from the motherboard via the daughtercard connectors.

The EVM has onboard logic that controls the memory mapping of the ADC within the motherboard's peripheral memory space.

## 1.3 Power Requirements

The EVM can be powered directly through the daughtercard connector's +3.3-V supply. Provision has also been made to allow the EVM to be powered with independent +3.3-V analog and digital supplies where ultimate performance demands.

WARNING

**Voltage Limits**

**Exceeding the +3.3-V maximum can damage EVM components. Undervoltage may cause improper operation of some or all of the EVM components**

## 1.4 THS1408/5691EVM Operational Procedure

The THS1408_5691EVM provides a flexible means of evaluating the THS14XX in a number of modes of operation. A basic setup procedure that can be used as a board confidence check follows:

☐ Verify all jumper settings against the schematic jumper table:

| Device | Jumper Table (connection) |
|---|---|
| THS1408 | H3 pins 2–3, H5 pins 2–3, H6 pins 2–3, H7 pins 2–3, H8 pins 1–2, H9 pins 2–3, H12 pins 1–2, H13 pins 1–2 |

☐ Check that T2 is populated.

☐ Connect supplies to the EVM: +3.3 V on J7 and J8, and GND on J12 and J13.

☐ Switch power supplies on.

☐ Use a function generator with a 50-Ω output to apply a 5-MHz, 1.5-V offset, 3-Vp-p amplitude square-wave signal to J3.

☐ Use a function generator with a 50-Ω output to apply a 100-kHz, 0-V offset, 4-Vp-p amplitude sine-wave signal to J10.

☐ The digital pattern on the output daughtercard connector J1 should now represent a sine wave and can be monitored using a logic analyzer, or the EVM can be plugged into a motherboard and the data can be monitored using suitable software.

## Chapter 2

# Physical Description

This chapter describes the physical characteristics and PCB layout of the EVM and lists the components used on the module.

**Topic** **Page**

## 2.1 PCB Layout

The EVM is constructed on a four-layer, 100-mm (3.94 inch) × 86-mm (3.39 inch), 1.57-mm (0.062 inch) thick PCB using FR-4 material. Figures 2–1 through 2–6 show the individual layers.

*Figure 2–1. Silk Top*

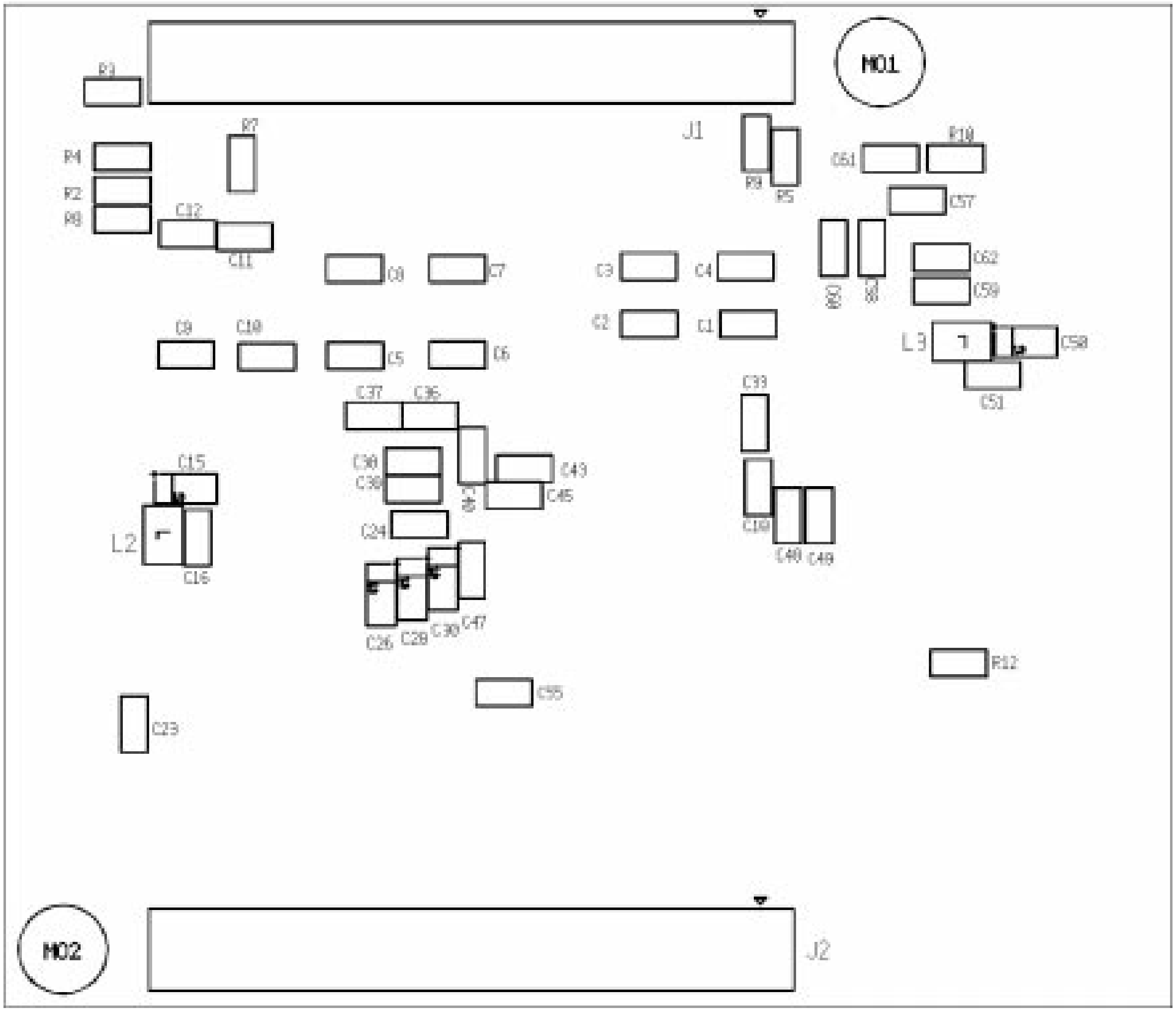


Figure 2–2. Silk Bottom

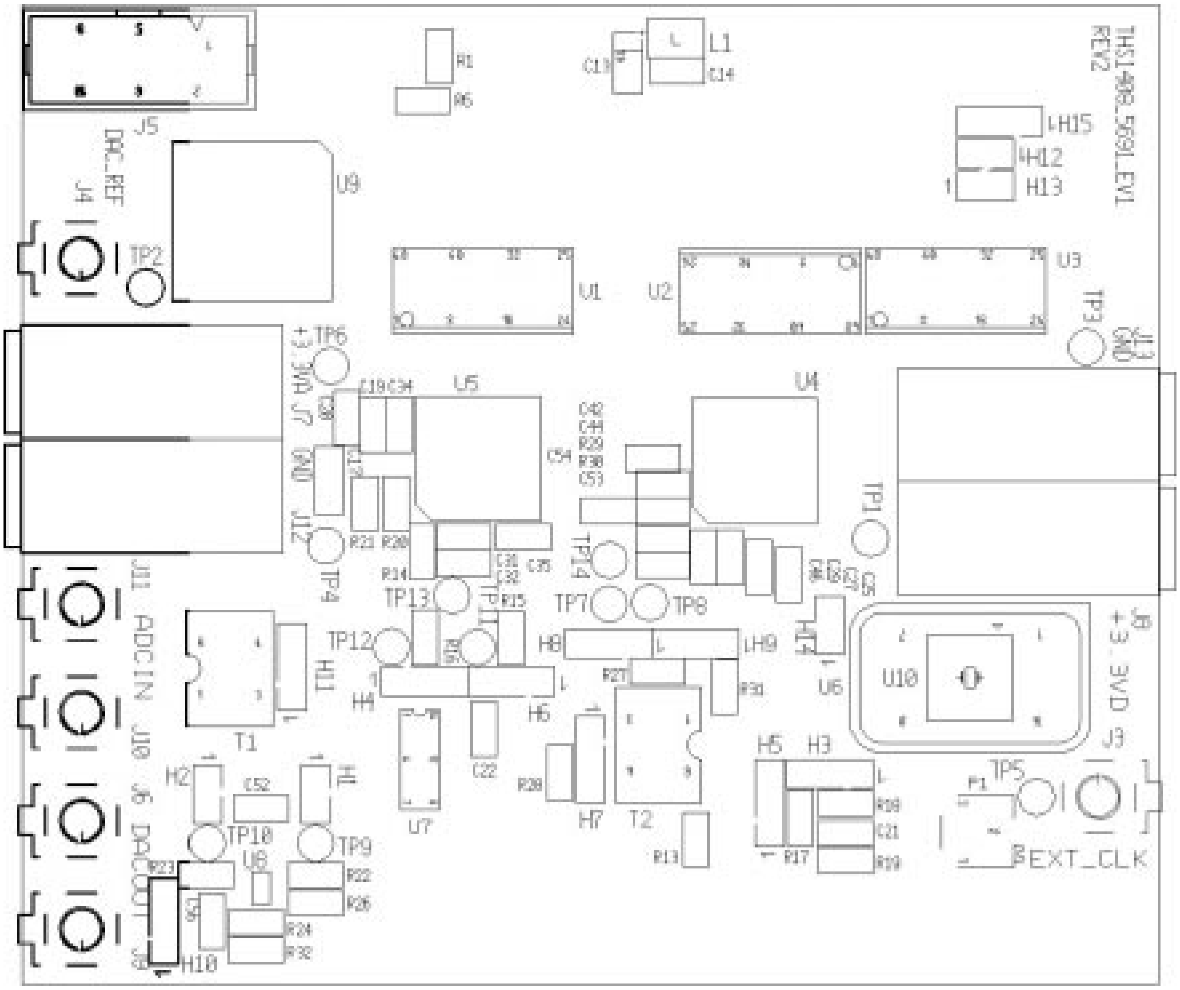


Figure 2–3. Top Layer

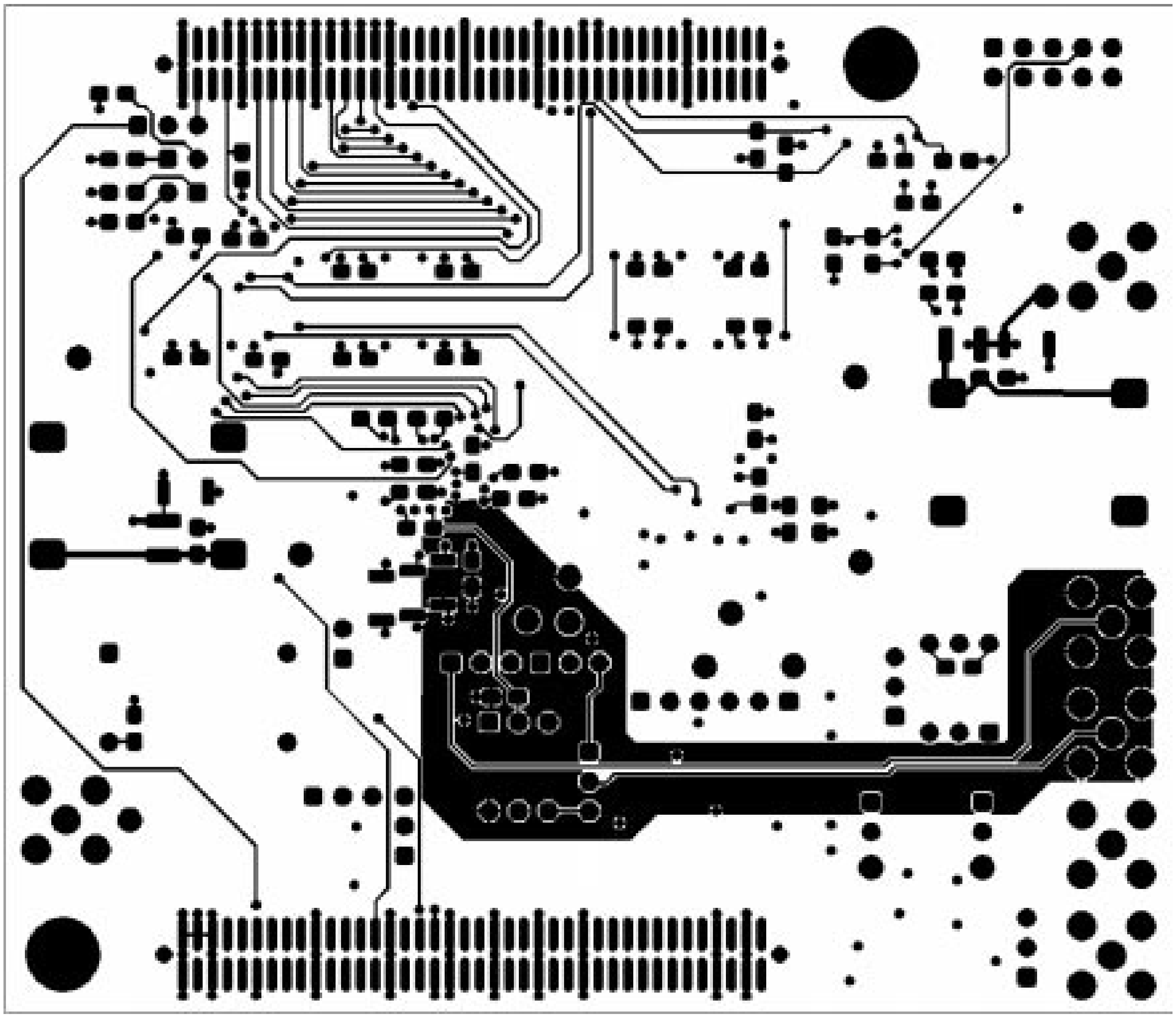


Figure 2–4. Inner Layer 1

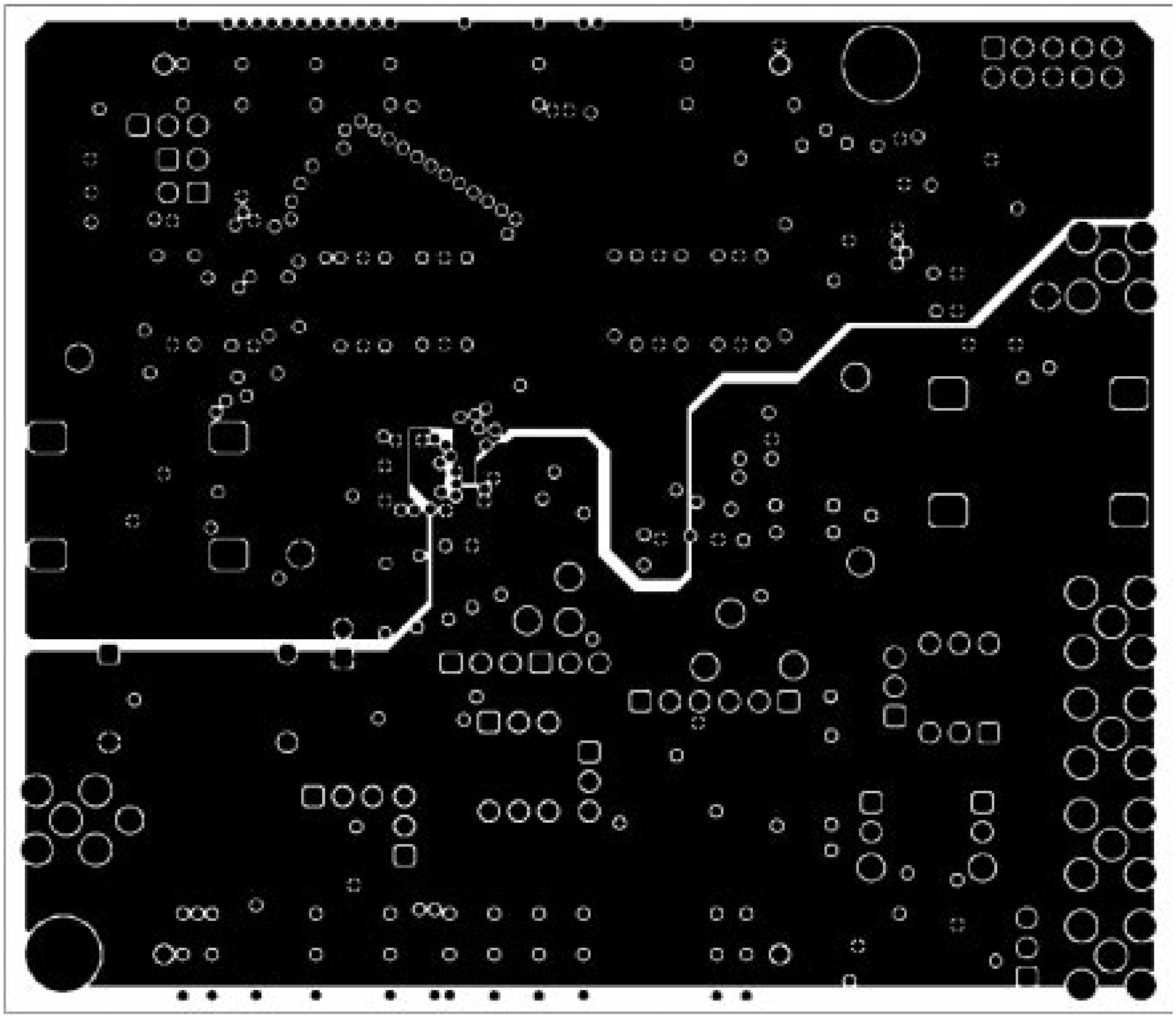


Figure 2–5. Inner Layer 2

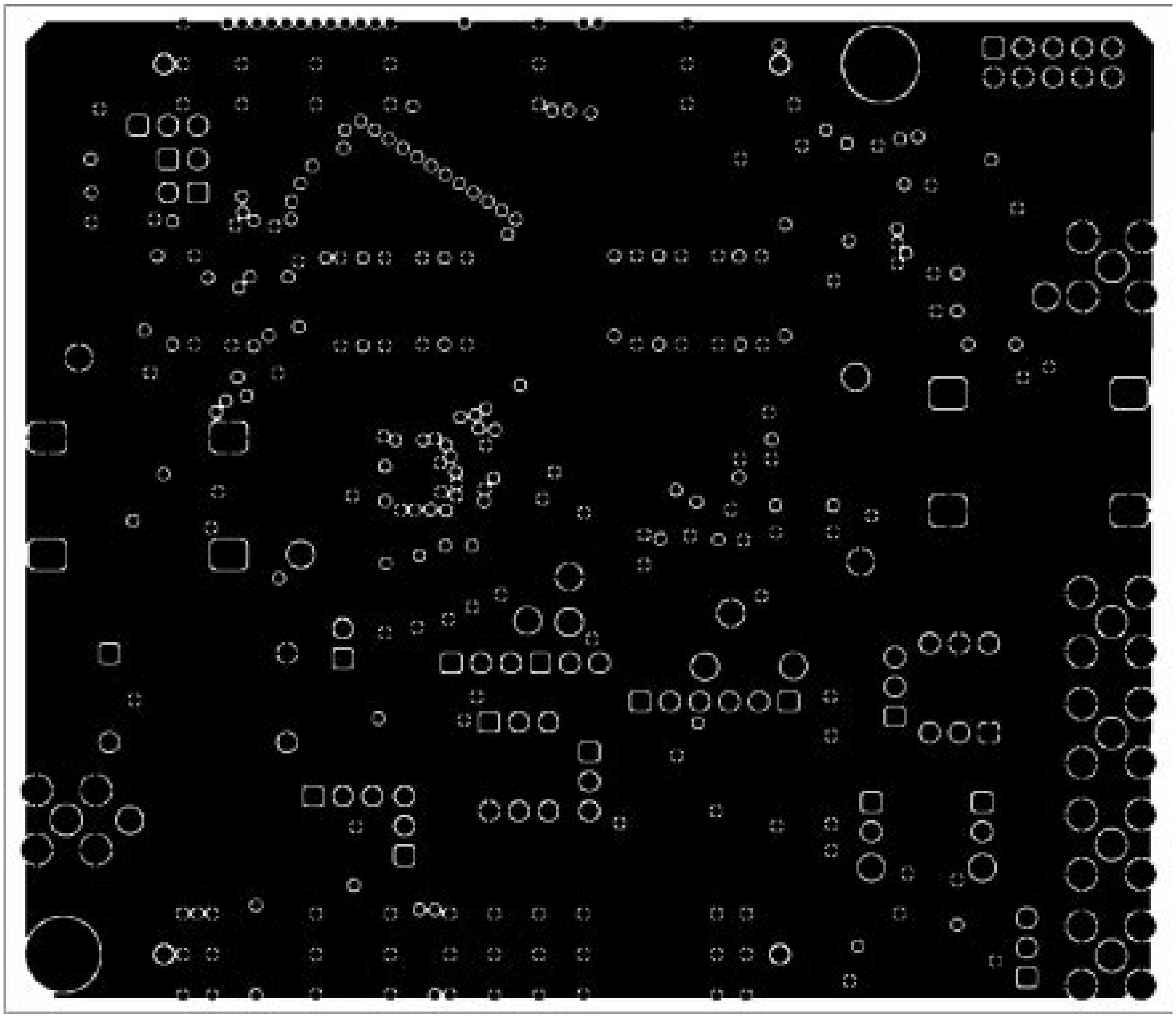
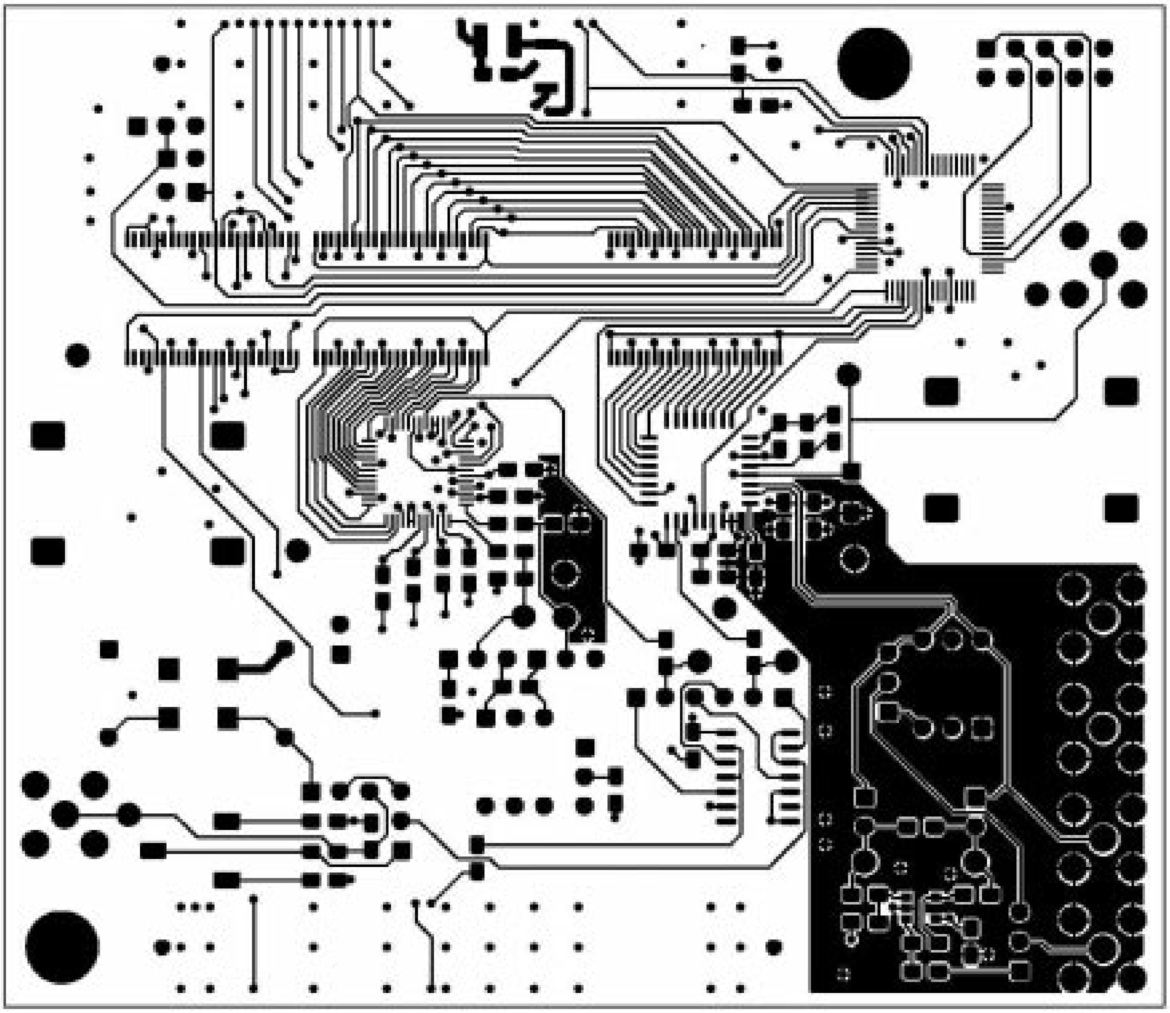


Figure 2–6. Bottom Layer



## 2.2 Parts List

Table 2–1 lists the parts used in constructing the EVM.

*Table 2–1. Parts List*

| Qty | Reference Description | Description | Manufacturer | Part Number |
|---|---|---|---|---|
| 2 | U3, U1 | SN74ALB16244DL bus driver | TI | SN74ALB16244DL |
| 1 | U2 | SN74ALB16245DL bus transceiver | TI | SN74ALB16245DL |
| 6 | C13, C15, C26, C28, C50, C30 | 10-µF 10-V case A SMD tantalum capacitor | AVX | TAJA106K010R |
| 1 | C17 | 1-µF 1206 SMD ceramic capacitor 16-V X7R | AVX | 1206YC105K2800J |
| 40 | C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C14, C16, C18, C19, C21, C22, C23, C24, C25, C27, C29, C32, C34, C35, C36, C37, C38, C39, C40, C43, C45, C47, C51, C55, C56, C57, C58, C59 | 0.1-µF 0805 SMD ceramic capacitor X7R | TDK | CC0805HX7R104KTR |
| 6 | C20, C31, C33, C42, C44, C46 | 470-pF 0805 SMD ceramic capacitor NPO | AVX | 08051A471JAT00J |
| 2 | C48, C49 | 22-pF 0805 SMD ceramic capacitor NPO | AVX | 08051A220JAT2A |
| 3 | C52, C53, C54 | UNPOP | | |
| 3 | C60, C61, C62 | 0.01-µF 0805 SMD ceramic capacitor X7R | TDK | CC0805HX7R103KTR |
| 6 | J3, J4, J6, J9, J10, J11 | SMB connector right angle PCB | MACOM | B65N10G999X99 |
| 1 | J5 | 0.1” spacing 2X5 header straight | Elco | 008380010000010 |
| 1 | U7 | SN74AHC14D hex inverter | TI | SN74AHC14D |
| 5 | H1, H2, H12, H13, H14 | 0.1” spacing 1X2 header straight | Harwin | M20–9990206 |
| 10 | H3, H4, H5, H6, H7, H8, H9, H10, H11, H15 | 0.1” spacing 1X3 header straight | Harwin | M20–9990306 |
| 3 | L1, L2, L3 | 0R 1206 thick-film resistor | Multicomp | RMC18W(1206)5%0R0 |
| 1 | U8 | MAX4212EUK 3.3-V operational amplifier | Maxim | MAX4212EUK |
| 1 | U6 | 8-MHz 14-pin DIL 3.3V oscillator 100 ppm | Golledge | GXO–U102F 8 MHz |
| 1 | U10 | 8.0-MHz SMD 3.3V oscillator 100 ppm | Golledge | GXO–U108L 8 MHz |
| 2 | J12, J13 | 4-mm PCB socket black | Deltron | 571–0100 |
| 2 | J7, J8 | 4-mm PCB socket red | Deltron | 571–0500 |
| 1 | P1 | 2-kΩ 3269P potentiometer | Bourns | 3269P1202 |

*Table 2–1. Parts List (Continued)*

| Qty | Reference Description | Description | Manufacturer | Part Number |
|---|---|---|---|---|
| 8 | R1, R2, R4, R7, R9, R10, R14 | 10-kΩ 0805 thick-film resistor 1% | Multicomp | RMC110W(O8O5)1%10K |
| 3 | R5, R6, R8 | 510R 0805 thick-film resistor 1% | Multicomp | RMC110W(O8O5)5%510R |
| 2 | R15, R16 | 39R 0805 thick-film resistor 1% | Multicomp | RMC110W(O8O5)1%39R |
| 2 | R22, R23 | 150R 0805 thick-film resistor 1% | Multicomp | RMC110W(O8O5)1%150R |
| 1 | R24 | 750R 0805 thick-film resistor 1% | Rohm | MCR10EZHF7500 |
| 3 | R13, R29, R30 | 0R 0805 thick-film resistor 1% | Multicomp | RMC110W(O8O5)5%0R0 |
| 1 | R3 | 1K 0805 thick-film resistor 1% | Multicomp | RMC110W(O8O5)1%1K |
| 1 | R17, R20, R21, R27, R28, R31, R32 | 49R9 0805 SMD chip resistor 1% | Rohm | MCR10EZHF49R9 |
| 1 | R12 | UNPOP | | |
| 2 | R18, R19 | 5K1 0805 thick-film resistor 1% | Multicomp | RMC110W(O8O5)5%5K1 |
| 1 | R26 | 2-kΩ 0805 thick-film resistor 1% | Rohm | MCR10EZHF2001 |
| 4 | TP3, TP4, TP13, TP14 | 1.32-mm test pin black | W Hughes | 100–103 |
| 10 | TP1, TP2, TP5, TP6, TP7, TP8, TP9, TP10, TP11, TP12 | 1.32-mm test pin red | W Hughes | 100–107 |
| 2 | J1, J2 | 0.05” TFM-series 80-pin TH connector | Samtec | TFM–140–31–S–D–LC |
| 1 | U4 | THS14XX ADC | TI | THS14XXPFB |
| 1 | U5 | THS56XX DAC | TI | THS56XXVF |
| 2 | T2, T1 | TT1–6–KK81 RF transformer | Mini Circuits | TT1–6–KK81 |
| 1 | U9 | XC9536XL–5VQ64C CPLD | Xilinx | XC9536XL–5VQ64C |

The following components have been left unpopulated in this implementation: C1, C2, C3, C4, C17, C18, C19, C20, C31, C32, C33, C34, C35, C48, C49, C52, C53, C54, C56, H1, H2, H4, H10, H11, J4, J6, J9, R12, R14, R16, R20, R21, R22, R23, R24, R26, R32, T1, TP6, TP9, TP10, TP12, U1, U5, U8, U10.

## Chapter 3

# Circuit Description

This chapter contains the EVM schematic diagram and discusses the various functions on the EVM.

**Topic** **Page**

## 3.1 Circuit Function

The following paragraphs describe the function of the individual circuits. Refer to the relevant data sheet for device operating characteristics.

### 3.1.1 Inputs and Outputs

The ADC has differential analog inputs. These are provided via SMB connectors J10 and J11 on the EVM, and can be configured in two ways:

- For true differential input, the positive differential signal is connected to J10 and the negative differential signal is connected to J11. The jumpers on the board are then configured as: H7 (1–2), H8 (2–3), and H9 (1–2). The inputs have 50-Ω terminators.
- For single-ended input, a positive signal is applied to connector J10. The jumpers on the board are then configured as: H7 (2–3), H8 (1–2), and H9 (2–3). Transformer T2 performs the single-ended to differential signal conversion. In this mode, the 50-Ω terminators R27 and R28 perform impedance matching to attain the best distortion performance, but the signal source must be able to drive the 25-Ω load. A 50-Ω source can be used if R27 is removed from the board, with a resulting marginal increase in distortion.

SMB connector J3 can be used to input a clock signal to the board from an external source. If the source is not at the correct dc level required to input to the 74AHC14 hex inverter IC (U7), then it can be ac-coupled through C21, with the dc level trimmed using potentiometer P1 if necessary.

The EVM is designed to comply with the TI TMS320C6000 daughtercard specification. The pinout used is listed in Tables 3–1 and 3–2. In order to make the EVM compatible with both the TMS320C6XXX and TMS320C5XXX motherboard design environments, the EVM chip enable is derived from the following sources:

- $\overline{\text{OUT\_CE}}$ (J1 P78) for the TMS320C6XXX, by inserting a jumper link in position 2–3 of H15.
- OUT_IS (J2 P70) for the TMS320C5XXX, by inserting a jumper link in position 1–2 of H15.

For further explanation of the daughtercard connector interface, refer to the relevant motherboard user’s guide or daughtercard specification.

Buffers U1, U2, and U3 are used to avoid bus contention and provide a degree of noise isolation from the DSP motherboard.

*Table 3–1. Daughtercard Connector J1*

| J1 Pin | Name | Function | J1 Pin | Name | Function |
|---|---|---|---|---|---|
| 1 | | NC | 41 | +3.3V | POWER |
| 2 | | NC | 42 | +3.3V | POWER |
| 3 | | NC | 43 | | NC |
| 4 | | NC | 44 | | NC |
| 5 | | NC | 45 | | NC |
| 6 | | NC | 46 | | NC |
| 7 | | NC | 47 | | NC |
| 8 | | NC | 48 | | NC |
| 9 | | NC | 49 | | NC |
| 10 | | NC | 50 | | NC |
| 11 | GND | GND | 51 | GND | GND |
| 12 | GND | GND | 52 | GND | GND |
| 13 | | NC | 53 | BIDIR_ED15 | DATA |
| 14 | | NC | 54 | BIDIR_ED14 | DATA |
| 15 | | NC | 55 | BIDIR_ED13 | DATA |
| 16 | | NC | 56 | BIDIR_ED12 | DATA |
| 17 | | NC | 57 | BIDIR_ED11 | DATA |
| 18 | | NC | 58 | BIDIR_ED10 | DATA |
| 19 | | NC | 59 | BIDIR_ED9 | DATA |
| 20 | OUT_EA6 | Address | 60 | BIDIR_ED8 | DATA |
| 21 | | NC | 61 | GND | GND |
| 22 | | NC | 62 | GND | GND |
| 23 | OUT_EA5 | Address | 63 | BIDIR_ED7 | DATA |
| 24 | OUT_EA4 | Address | 64 | BIDIR_ED6 | DATA |
| 25 | OUT_EA3 | Address | 65 | BIDIR_ED5 | DATA |
| 26 | OUT_EA2 | Address | 66 | BIDIR_ED4 | DATA |
| 27 | | NC | 67 | BIDIR_ED3 | DATA |
| 28 | | NC | 68 | BIDIR_ED2 | DATA |
| 29 | | NC | 69 | BIDIR_ED1 | DATA |
| 30 | | NC | 70 | BIDIR_ED0 | DATA |
| 31 | GND | GND | 71 | GND | GND |
| 32 | GND | GND | 72 | GND | GND |
| 33 | | NC | 73 | $\overline{\text{OUT\_ARE}}$ | READ STROBE |
| 34 | | NC | 74 | $\overline{\text{OUT\_AWE}}$ | WRITE STROBE |
| 35 | | NC | 75 | | NC |
| 36 | | NC | 76 | | NC |
| 37 | | NC | 77 | | NC |
| 38 | | NC | 78 | $\overline{\text{OUT\_CE}}\text{\_6}$ | CHIP ENABLE |
| 39 | | NC | 79 | GND | GND |
| 40 | | NC | 80 | GND | GND |

*Table 3–2. Daughtercard Connector J2*

| J2 Pin | Name | Function | J2 Pin | Name | Function |
|---|---|---|---|---|---|
| 1 | | NC | 41 | | NC |
| 2 | | NC | 42 | | NC |
| 3 | GND | GND | 43 | GND | GND |
| 4 | GND | GND | 44 | GND | GND |
| 5 | | NC | 45 | OUT_TOUT | TIMER INPUT |
| 6 | | NC | 46 | IN_TINP | TIMER OUTPUT |
| 7 | GND | GND | 47 | | NC |
| 8 | GND | GND | 48 | | NC |
| 9 | | NC | 49 | | NC |
| 10 | | NC | 50 | | NC |
| 11 | | NC | 51 | GND | GND |
| 12 | | NC | 52 | GND | GND |
| 13 | | NC | 53 | IN_EXT_INT | INTERRUPT |
| 14 | | NC | 54 | | NC |
| 15 | | NC | 55 | | NC |
| 16 | | NC | 56 | | NC |
| 17 | | NC | 57 | | NC |
| 18 | | NC | 58 | | NC |
| 19 | | NC | 59 | | NC |
| 20 | | NC | 60 | | NC |
| 21 | | NC | 61 | GND | GND |
| 22 | | NC | 62 | GND | GND |
| 23 | | NC | 63 | | NC |
| 24 | | NC | 64 | | NC |
| 25 | GND | GND | 65 | | NC |
| 26 | GND | GND | 66 | | NC |
| 27 | | NC | 67 | | NC |
| 28 | | NC | 68 | | NC |
| 29 | | NC | 69 | | NC |
| 30 | | NC | 70 | OUT_IS | CHIP ENABLE |
| 31 | GND | GND | 71 | | NC |
| 32 | GND | GND | 72 | | NC |
| 33 | | NC | 73 | | NC |
| 34 | | NC | 74 | | NC |
| 35 | | NC | 75 | GND | GND |
| 36 | | NC | 76 | GND | GND |
| 37 | GND | GND | 77 | | GND |
| 38 | GND | GND | 78 | | NC |
| 39 | | NC | 79 | GND | GND |
| 40 | | NC | 80 | GND | GND |

### 3.1.2 Clock Options

The EVM provides flexibility for the source of the ADC’s conversion clock. This clock can come from an external source as previously described, from a standard DIL14 HCMOS crystal oscillator module populating U6, or from a U108-style surface-mount oscillator populating U10.

**Note:**

Ensure that the crystal oscillator module selected can operate at the +3.3-V supply voltage.

Supplying the ADC clock using a timer output (OUT_TOUT J2 P45) is an option provided for applications that require synchronization of the ADC conversion to the DSP on the motherboard. This is selected using a jumper link on H6 (1–2).

A summary of the jumper link settings required for the various clock options is shown in Table 3–3.

*Table 3–3. Jumper Settings for Clock Options*

| Clock Options | Onboard Oscillator | External AC Coupled Via J3 | External DC Coupled Via J3 | DSP Timer Via OUT_TOUT |
|---|---|---|---|---|
| H 3 | 1–2 | – | 2–3 | – |
| H 5 | 2–3 | 1–2 | 2–3 | – |
| H 6 | 2–3 | 2–3 | 2–3 | 1–2 |

### 3.1.3 References

The EVM relies on the THS1408 ADC using its on-chip reference.

### 3.1.4 Power

Power is supplied to the EVM via 4-mm banana sockets and the daughtercard connectors. This provides flexibility and a trade-off between convenience and performance as follows:

- For best performance: Use a separate low-noise analog power supply connected to J7 (+3.3 V) and J12 (GND). Use the digital supply from the daughtercard connectors.
- For good performance: Use the daughtercard connectors for both analog and digital supplies. This is achieved by inserting a jumper link in H14 on the EVM.
- For stand-alone operation: Use a separate low-noise analog power supply connected to J7 (+3.3 V) and J12 (GND) and a separate low-noise digital power supply connected to J8 (+3.3 V) and J13 (GND).

### 3.1.5 Interrupts

The ADC produces two signals that can be used to interrupt a DSP:

- INT: This is routed via buffer U3 to the IN_EXT_INT pin on the daughtercard connector.

**Note:**

This function is only supported on certain members of the THS14XX family. Refer to the device datasheet.

- FOVL: This is routed via buffer U3 to both BIDIR_ED14 and IN_TINP. This gives the DSP code developer flexibility on using the signal in their system.

### 3.1.6 Hardware Loopback

The EVM can be configured via insertion of jumper links H12 and H13 to go into a hardware loopback mode. This is primarily intended so that the board can be put into an operational state when it is powered up stand-alone. The control logic for this is explained in section 3.2.8.

### 3.1.7 ADC Write/Read Cycle

The following daughtercard address and control lines are mapped directly to the ADC when performing a write to or a read from the device:

- OUT_EA2 is mapped to A0
- OUT_EA3 is mapped to A1
- $\overline{\text{OUT\_AWE}}$ is mapped to WRB
- $\overline{\text{OUT\_ARE}}$ is mapped to OEB

These, combined with the address decode logic described in section 3.2.8, allow direct control of the ADC operation via the normal write/read cycles of a DSP.

### 3.1.8 Address Decode Logic

The EVM uses a Xilinx XC9536 (U9) CPLD to perform address decode and control. The CPLD decodes the daughtercard address and strobe lines to control the ADC chip select and the buffer direction and output-enable states. The Abel code used to generate the logic is given in Listing 3–1**.**

## 3.2 Address Decode and Control Logic Listing

### *Module Address*

### *Address Decode for THS1408/5691*

This file performs the address decode and logic control for the THS1408_5691_EV1_Rev2 PCB.

The equivalent truth tables are:

**Chip Select Decode**

| OUT_EA6 | OUT_EA5 | OUT_EA4 | OUT_EA3 | OUT_EA2 | $\overline{\text{OUT\_CE}}$ | ADCCSB | DACCSB |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| All other combinations | | | | | | 1 | 1 |

**Buffer Control Decode**

| Operation | ADCCSB | DACCSB | OUT_ARE | OUT_AWE | DBUFOEB | ABUFDIR | ABUFDIRB | ABUFOEB | DACMODE |
|---|---|---|---|---|---|---|---|---|---|
| Write to DAC | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| Write to ADC | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| Read from ADC | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| Loop back | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| All other combinations | | | | | 1 | 1 | 1 | 1 | 1 |

**Note:** Bit 13 of the data bus to the DAC is inverted when in loopback mode to account for the DAC powering up in 2s-complement input mode while the ADC powers up in binary mode.

Inputs

| | |
|---|---|
| OUT_EA2 | pin; Address bit from DSP board |
| OUT_EA3 | pin; Address bit from DSP board |
| OUT_EA4 | pin; Address bit from DSP board |
| OUT_EA5 | pin; Address bit from DSP board |

OUT_EA6 pin; Address bit from DSP board

BIDIR_ED13 pin; Data bit 13

$\overline{\text{OUT\_AWE}}$ pin; DSP write enable

OUT_ARE pin; DSP read enable

OUT_CE pin; DSP chip enable

Outputs

DBUFOEB pin istype ’com’; DAC buffers 3-state control

ABUFDIR pin istype ’com’; ADC buffer direction control

ABUFDIRB pin istype ’com’; ADC FOVL flag buffer 3-state control

ABUFOEB pin istype ’com’; ADC data bus buffer 3-state control

DACMODE pin istype ’com’; DAC mode

Mod_ED13 pin istype ’com’; Selectively invert BIDIR_ED13 for loop-back

ADCCSB pin istype ’com’; Chip select for ADC

Equations

```
!ADCCSB = (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and !OUT_EA3 and OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and OUT_EA3 and !OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and OUT_EA3 and OUT_EA2 and !OUT_CE)
# (OUT_EA6 and OUT_EA5 and !OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE);
!DACCSB = (OUT_EA6 and !OUT_EA5 and OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and OUT_EA4 and !OUT_EA3 and OUT_EA2 and !OUT_CE)
# (OUT_EA6 and OUT_EA5 and !OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE);
!DBUFOEB = (ADCCSB and !DACCSB and OUT_ARE and !OUT_AWE)
# (!ADCCSB and !DACCSB and !OUT_ARE and OUT_AWE); Done in longhand so signal path is only once
  through device
!DBUFOEB = (!((OUT_EA6 and !OUT_EA5 and !OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and !OUT_EA3 and OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and OUT_EA3 and !OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and OUT_EA3 and OUT_EA2 and !OUT_CE)
# (OUT_EA6 and OUT_EA5 and !OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE)) and ((OUT_EA6
and !OUT_EA5 and OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and OUT_EA4 and !OUT_EA3 and OUT_EA2 and !OUT_CE)
# (OUT_EA6 and OUT_EA5 and !OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE)) and OUT_ARE and
  !OUT_AWE)
# (((OUT_EA6 and !OUT_EA5 and !OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and !OUT_EA3 and OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and OUT_EA3 and !OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and OUT_EA3 and OUT_EA2 and !OUT_CE)
# (OUT_EA6 and OUT_EA5 and !OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE)) and ((OUT_EA6
  and !OUT_EA5 and OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE)
```

```
# (OUT_EA6 and !OUT_EA5 and OUT_EA4 and !OUT_EA3 and OUT_EA2 and !OUT_CE)
# (OUT_EA6 and OUT_EA5 and !OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE)) and !OUT_ARE
    and OUT_AWE);
!ABUFDIR = (!ADCCSB and DACCSB and OUT_ARE and !OUT_AWE); Done in longhand so signal path is only
   once through device
!ABUFDIR = (((OUT_EA6 and !OUT_EA5 and !OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and !OUT_EA3 and OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and OUT_EA3 and !OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and OUT_EA3 and OUT_EA2 and !OUT_CE)
# (OUT_EA6 and OUT_EA5 and !OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE)) and !((OUT_EA6
   and !OUT_EA5 and OUT_EA4 and !OUT_EA3 & !OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and OUT_EA4 and !OUT_EA3 and OUT_EA2 and !OUT_CE)
# (OUT_EA6 and OUT_EA5 and !OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE)) and OUT_ARE and
   !OUT_AWE);
!ABUFDIRB = (!ADCCSB and DACCSB and !OUT_ARE and OUT_AWE)
# (!ADCCSB and !DACCSB and !OUT_ARE and OUT_AWE); Done in longhand so signal path is only once
   through device
!ABUFDIRB = (((OUT_EA6 and !OUT_EA5 and !OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and !OUT_EA3 and OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and OUT_EA3 and !OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and OUT_EA3 and OUT_EA2 and !OUT_CE)
# (OUT_EA6 and OUT_EA5 and !OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE)) and !((OUT_EA6
and !OUT_EA5 and OUT_EA4 and !OUT_EA3 & !OUT_EA2 and !OUT_CE)
# (OUT_EA6 and OUT_EA5 and !OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE)) and !OUT_ARE
    and OUT_AWE);
!ABUFOEB = (!ADCCSB and DACCSB and OUT_ARE and !OUT_AWE)
# (!ADCCSB and DACCSB and !OUT_ARE and OUT_AWE)
# (!ADCCSB and !DACCSB and !OUT_ARE and OUT_AWE); Done in longhand so signal path is only once
   through device
!ABUFOEB = (((OUT_EA6 and !OUT_EA5 and !OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and !OUT_EA3 and OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and OUT_EA3 and !OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and OUT_EA3 and OUT_EA2 and !OUT_CE)
# (OUT_EA6 and OUT_EA5 and !OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE)) and !((OUT_EA6
   and !OUT_EA5 and OUT_EA4 and !OUT_EA3 & !OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and OUT_EA4 and !OUT_EA3 and OUT_EA2 and !OUT_CE)
# (OUT_EA6 and OUT_EA5 and !OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE)) and OUT_ARE and
   !OUT_AWE)
# (((OUT_EA6 and !OUT_EA5 and !OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and !OUT_EA3 and OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and OUT_EA3 and !OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and OUT_EA3 and OUT_EA2 and !OUT_CE)
# (OUT_EA6 and OUT_EA5 and !OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE)) and !((OUT_EA6
   and !OUT_EA5 and OUT_EA4 and !OUT_EA3 & !OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and OUT_EA4 and !OUT_EA3 and OUT_EA2 and !OUT_CE)
# (OUT_EA6 and OUT_EA5 and !OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE)) and !OUT_ARE
    and OUT_AWE)
# (((OUT_EA6 and !OUT_EA5 and !OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and !OUT_EA3 and OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and OUT_EA3 and !OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and OUT_EA3 and OUT_EA2 and !OUT_CE)
# (OUT_EA6 and OUT_EA5 and !OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE)) and ((OUT_EA6
   and !OUT_EA5 and OUT_EA4 and !OUT_EA3 & !OUT_EA2 and !OUT_CE)
```

```
# (OUT_EA6 and !OUT_EA5 and OUT_EA4 and !OUT_EA3 and OUT_EA2 and !OUT_CE)
# (OUT_EA6 and OUT_EA5 and !OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE)) and !OUT_ARE
    and OUT_AWE);
!DACMODE = (!ADCCSB and !DACCSB and !OUT_ARE and OUT_AWE); Done in longhand so signal path is
   only once through device
!DACMODE = (((OUT_EA6 and !OUT_EA5 and !OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and !OUT_EA3 and OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and OUT_EA3 and !OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and OUT_EA3 and OUT_EA2 and !OUT_CE)# (OUT_EA6 and
  OUT_EA5 and !OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE)) and ((OUT_EA6 and !OUT_EA5
  and OUT_EA4 and !OUT_EA3 & !OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and OUT_EA4 and !OUT_EA3 and OUT_EA2 and !OUT_CE)
# (OUT_EA6 and OUT_EA5 and !OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE)) and !OUT_ARE
  and OUT_AWE);Mod_ED13 = (!BIDIR_ED13 and ((OUT_EA6 and !OUT_EA5 and !OUT_EA4 and !OUT_EA3
  and !OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and !OUT_EA3 and OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and OUT_EA3 and !OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and OUT_EA3 and OUT_EA2 and !OUT_CE)
# (OUT_EA6 and OUT_EA5 and !OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE)) and ((OUT_EA6
  and !OUT_EA5 and OUT_EA4 and !OUT_EA3 & !OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and OUT_EA4 and !OUT_EA3 and OUT_EA2 and !OUT_CE)
# (OUT_EA6 and OUT_EA5 and !OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE)) and !OUT_ARE
and OUT_AWE)
# (BIDIR_ED13 and !(((OUT_EA6 and !OUT_EA5 and !OUT_EA4 and !OUT_EA3 and !OUT_EA2 and
  !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and !OUT_EA3 and OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and OUT_EA3 and !OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and !OUT_EA4 and OUT_EA3 and OUT_EA2 and !OUT_CE)
# (OUT_EA6 and OUT_EA5 and !OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE)) and ((OUT_EA6
  and !OUT_EA5 and OUT_EA4 and !OUT_EA3 & !OUT_EA2 and !OUT_CE)
# (OUT_EA6 and !OUT_EA5 and OUT_EA4 and !OUT_EA3 and OUT_EA2 and !OUT_CE)
# (OUT_EA6 and OUT_EA5 and !OUT_EA4 and !OUT_EA3 and !OUT_EA2 and !OUT_CE)) and !OUT_ARE
  and OUT_AWE));

END
```

## 3.3 Schematic Diagrams

This section contains the schematic diagrams for the EVM.