

FSF-AD8200A

8-Channel, 185MSPS JESD204B ADC FMC



User Manual

January, 2014
Version 1.0

Fidus Systems Inc.
35 Fitzgerald Road, Suite 400
Ottawa, ON K2H 1E6
CANADA
Tel: (613) 828-0063
Fax: (613) 828-3113

Fidus Systems Inc. is at the forefront of technology innovation and for this reason, reserves the right to alter, without notice, the specification, design or conditions of supply of any product or service. Information provided by Fidus Systems is believed to be accurate and reliable. However, no responsibility is assumed by Fidus Systems Inc. for its use, nor any infringements of patents or other rights of third parties, which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Fidus Systems Inc.

© 2014 Fidus Systems Inc. All Rights Reserved. Fidus' name, Fidus logo, and "FMCs by Fidus" brand, are trademarks of Fidus Systems Inc. Other registered and unregistered trademarks are the property of their respective owners. Information is subject to change without notice.


Revision History

| Revision | Author | Release Date | Description of Change |
|----------|--------|--------------|--|
| 0.1 | JLM | 06-18-2013 | Draft |
| 0.2 | JLM | 12-12-2013 | Updates |
| 0.3 | EJD | 12.19.2013 | Updated section 4, to add info on Command Interface, Scripts, and refer to the Getting Started Guide, and Command Interface Guide. |
| 1.0 | ST | 01-03-2014 | Update. Release. |

Table of Contents

| | | |
|------------|---|-----------|
| 1. | SAFETY INFORMATION AND PRODUCT AND COMPLIANCE LIMITATIONS..... | 1 |
| 2. | GENERAL OVERVIEW | 2 |
| 3. | GLOSSARY..... | 4 |
| 4. | REFERENCE DOCUMENTS..... | 4 |
| 5. | DETAILED SPECIFICATIONS | 5 |
| 5.1 | PERFORMANCE | 5 |
| 5.1.1 | Input Frequency Response..... | 5 |
| 5.1.2 | SNR, SFDR, and THD Plots..... | 7 |
| 5.1.3 | Adjacent Channel Coupling..... | 8 |
| 5.1.4 | Intermodulation | 11 |
| 6. | INTERFACES..... | 13 |
| 6.1 | FMC INTERFACE: PIN SUPPORT AND OTHER REQUIREMENTS..... | 13 |
| 6.1.1 | FSF-AD8200A HPC FMC pin assignment and definition | 14 |
| 6.1.2 | FMC Voltage and Current requirements..... | 16 |
| 6.1.3 | Multi-Gigabit Serial Communications Lines | 16 |
| 6.2 | ADC ANALOG INPUTS..... | 17 |
| 6.3 | EXTERNAL CLOCK INPUT | 17 |
| 6.4 | EXTERNAL TRIGGER INPUT | 18 |
| 6.5 | COMMUNICATION AND CONTROL..... | 19 |
| 6.5.1 | SPI Interface..... | 19 |
| 6.5.1.1 | Clock Generator | 20 |
| 6.5.1.2 | Analog to Digital Converters | 20 |
| 6.5.2 | EEPROM | 21 |
| 7. | ENVIRONMENTAL AND MECHANICAL..... | 22 |
| 7.1 | ENVIRONMENT | 22 |
| 7.2 | THERMAL CONSIDERATIONS | 22 |
| 7.3 | MECHANICAL | 23 |
| 7.3.1 | Front Bezel (Faceplate) | 23 |
| 7.3.2 | Heat Sink..... | 24 |
| 8. | DEMONSTRATION | 25 |
| 8.1 | HARDWARE INSTALLATION..... | 25 |
| 8.2 | SOFTWARE INSTALLATION | 25 |
| 8.3 | PERFORMING AN 8-CHANNEL CAPTURE | 27 |
| 9. | ORDERING INFORMATION | 28 |
| 10. | WARRANTY | 28 |
| 11. | APPENDIX A: 'INIT' SCRIPT CONTENTS | 29 |

1. SAFETY INFORMATION AND PRODUCT AND COMPLIANCE LIMITATIONS

1. This product is designed for use and operation by an experienced electrical engineer or someone with similar experience, knowledge, and capabilities.
2. This product is not an apparatus in accordance with the definition in EMC directive (2004/108/EC), it is intended to be incorporated into an apparatus that is compliant to EMC directive.
3. To comply with the LVD directive (2006/95/EC), all the power sources for this card (12V, 3.3V, etc.) have to comply with LPS requirements as defined in IEC 60950-1.
4. This equipment must be disposed of and treated properly in compliance with WEEE directive 2012/19/EU, any product marked with this image  should not be disposed of in normal household waste as products with electrical or electronic components can be recycled and could also be harmful to the environment if sent to landfill.
5. This product is intended for incorporation into an apparatus that will carry the FCC 15 declaration.

Warning

Any unauthorized modification to this equipment may cause violation of the FCC or EMC rules resulting in the revocation of the authorization to operate the equipment

Notes



This equipment must be disposed of and treated properly in compliance with WEEE directive 2012/19/EU



This equipment is ESD sensitive and must be handled using industry accepted methods to help avoid damage from discharge events. This includes the wearing of grounding straps prior to and while handling the circuit board.



Sensitive electronic device

2. GENERAL OVERVIEW

The Fidus Systems FSF-AD8200A provides 8 channels of high performance analog capture in a VITA 57.1–2010 compatible, conduction cooled, single width FMC form factor.

Performance specifics include:

- Contains four IDT ADC1443D200 dual analog-to-digital converters
- JESD204B interface for low pin count, low noise, and deterministic latency
- Up to 185 MSPS conversion rate on each channel (default 180MSPS)
- 14-bit conversion on each channel
- Capable of sampling jitter below 100fsec RMS¹
- Extremely low channel-to-channel crosstalk
- Input -3 dB bandwidth of >500MHz

The product has planned hardware compatibility with the following Xilinx® evaluation boards:

- Virtex VC707
- Virtex VC709²
- Kintex KC705 (capable of 4 lanes due to the MGT assignment)²
- Virtex ML605 (capable of 2 JESD204B cores of 4 lanes each)²

The VC707 package includes all necessary FPGA code/files to:

- Form a single link, 8-lane JESD204B connection
- Either load a default setting to all configuration registers or experiment with custom settings
- Capture ADC samples from each converter into FPGA block memory resources
- Access the captured ADC samples through ILA (formerly known as ChipScope®), where they may then be exported for post processing
- Access all FSF-AD8200A registers through a command-line UART/terminal session

Figure 1 presents a detailed block diagram of the FSF-AD8200A functionality.

¹ Integrated phase noise in a 12kHz–20MHz offset from a 180MHz clock frequency.

² Although the existing hardware is theoretically compatible with these development boards, at the time of writing, Fidus has only verified operation with the VC707 development kit.

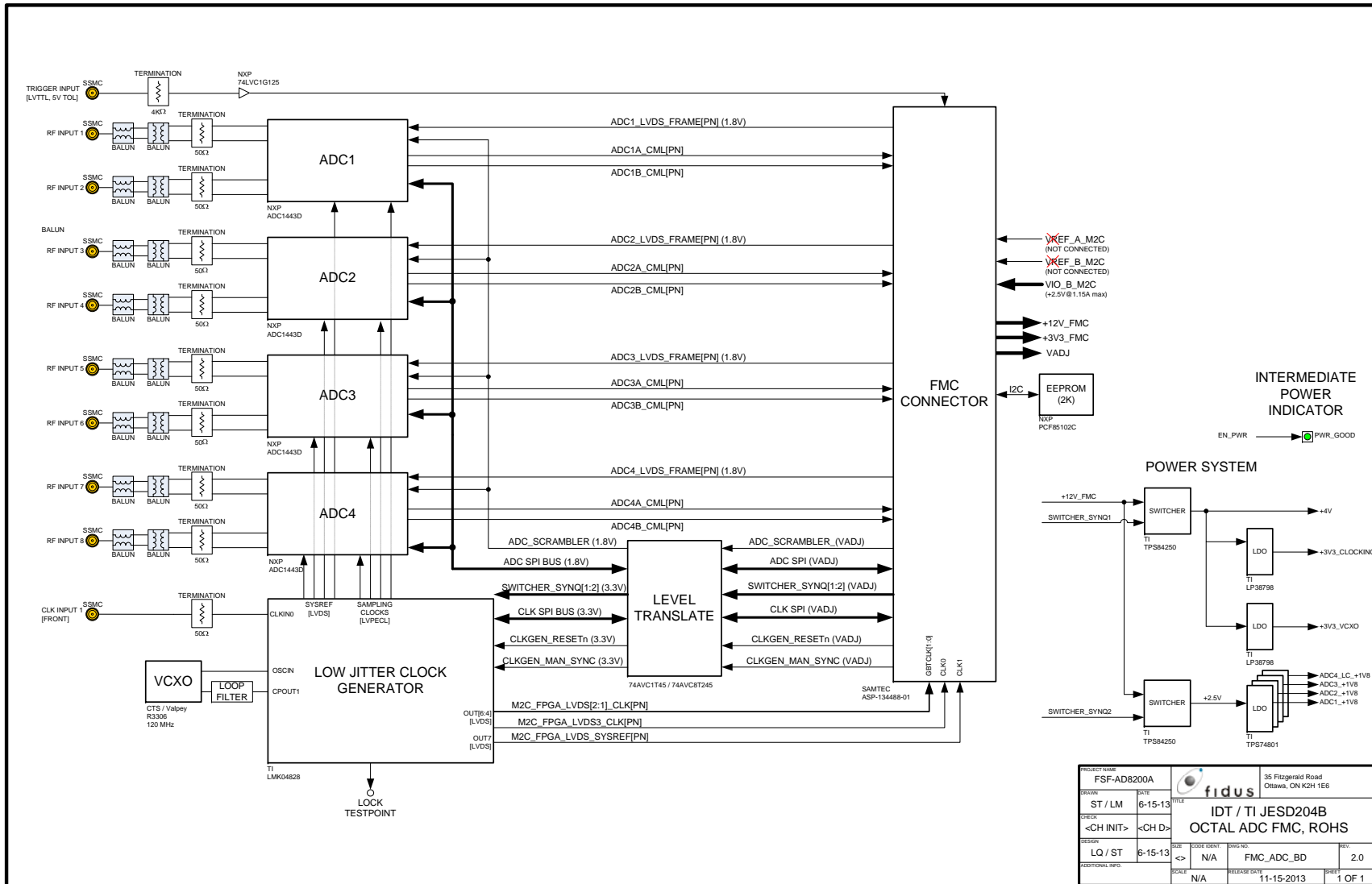


Figure 1: Detailed Block Diagram of the FSF-AD8200A

| | | | |
|-----------------------------|-----------------|---|-----------------------|
| PROJECT NAME FSF-AD8200A | | 35 Fitzgerald Road Ottawa, ON K2H 1E6 | |
| DRAWN ST / LM | DATE 6-15-13 | TITLE IDT / TI JESD204B OCTAL ADC FMC, ROHS | |
| CHECK <CH INIT> | <CH D> | SIZE LQ / ST | REV. 2.0 |
| DESIGN LQ / ST | DATE 6-15-13 | CODE IDENT N/A | DWG NO. FMC_ADC_BD |
| ADDITIONAL INFO NA | SCALE | RELEASE DATE 11-15-2013 | SHEET 1 OF 1 |

3. GLOSSARY

| | |
|------|--|
| ADC | Analog-to-Digital Converter |
| BW | Bandwidth |
| C2M | Carrier to Mezzanine (i.e. a signal output from carrier, input to FMC) |
| CML | Current Mode Logic |
| FMC | FPGA Mezzanine Card |
| HPC | High Pin Count |
| IMD | Intermodulation |
| M2C | Mezzanine to Carrier (i.e. a signal output from the FMC, input to the carrier) |
| MGT | Multi-Gigabit Transceiver |
| OFDM | Orthogonal Frequency Division Multiplexing |
| SFDR | Spurious Free Dynamic Range |
| SNR | Signal to Noise Ratio |

Xilinx®, Virtex®, and Vivado® are registered trademarks of Xilinx.
ChipScope™ is a trademark of Xilinx.

4. REFERENCE DOCUMENTS

| Reference Document | FSF-AD8200A Function | Hyperlink |
|--|----------------------|---|
| ANSI/VITA 57.1 FPGA Mezzanine Card (FMC) Standard – 2010. ISBN 1-885731-49-3 | | http://www.vita.com |
| IDT ADC1443D, datasheet | ADC | http://www.idt.com/document/dst/adc1443d-ser-datasheet |
| TI LMK04828B, datasheet | Clock Generator | http://www.ti.com/lit/gpn/lmk04828 |
| TI SN74AVC8T245 | Level shifter | http://www.ti.com/lit/ds/symlink/sn74avc8t245.pdf |
| TI SN74AVC1T245 | Level shifter | http://www.ti.com/lit/ds/symlink/sn74avc1t245.pdf |
| NXP PCF85102 | EEPROM | http://www.nxp.com/documents/data_sheet/PCF85102C_2.pdf |
| CTS/VF R3306, datasheet | 120MHz VCXO | http://www.ctsvalpey.com/Collateral/Documents/English-US/Products/FrequencyControl/Oscillators/VCXO/R3306.pdf |

Table 1: Reference Documents

5. DETAILED SPECIFICATIONS

5.1 Performance

The following sections present typical performance results, under the documented test conditions.

Test Conditions:

Ambient temperature: $25 \pm 5^\circ\text{C}$

Description: Controlled laboratory environment

Test Platform: Xilinx VC707 development kit

Sample Rate: $(178 + 4/7)$ MSPS

| Parameter | Min | Typ | Max | Unit | Conditions/Comments |
|-------------------------------------|-----|----------|-----|------|---|
| Usable analog input BW | | 1 to 700 | | MHz | S11 better than -7dB |
| Fin = 4.989MHz | | | | | |
| | | | | | @ -1.5dBFS Nyquist Zone 1 |
| SNR | | 69 | | dBc | |
| SFDR | | 83.2 | | dBc | |
| THD | | -79 | | dBc | |
| Coupling (adjacent) | | -88.2 | | dB | averaged based on both adjacent results |
| Coupling (2 nd adjacent) | | -100.3 | | dB | averaged based on both 2 nd adjacent results |
| Fin = 183.561MHz | | | | | |
| | | | | | @ -1.9dBFS Nyquist Zone 3 |
| SNR | | 61.7 | | dBc | |
| SFDR | | 70.2 | | dBc | |
| THD | | -74.8 | | dBc | |
| Coupling (adjacent) | | -77.5 | | dB | averaged based on both adjacent results |
| Coupling (2 nd adjacent) | | -100.4 | | dB | averaged based on both 2 nd adjacent results |

Table 2: Typical Performance at 4.989 and 183.561MHz

Notes:

- a) These results were captured without the heat sink and with additional applied air flow. The heat sink was added for customer ease of adoption, and is not expected to affect the results presented within this section.

5.1.1 Input Frequency Response

The ADC1443D converter claims a typical analog input bandwidth of 1GHz. On the FSF-AD8200A both the low end and high end frequency response is limited by front-end components/filters. The following plots describe the typical frequency performance of the inputs when swept.

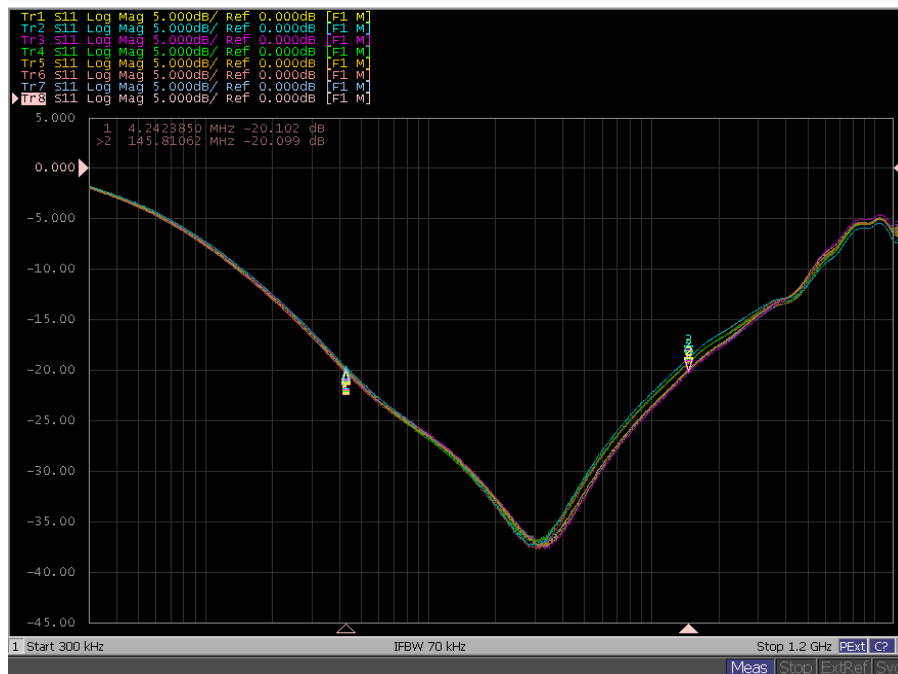


Figure 2: Overlay of typical Return Loss (S11) for all 8 Analog Input Channels (note log scale on x-axis)

The following plot describes the amplitude response and similarity of two analog inputs (Channel C and Channel D)

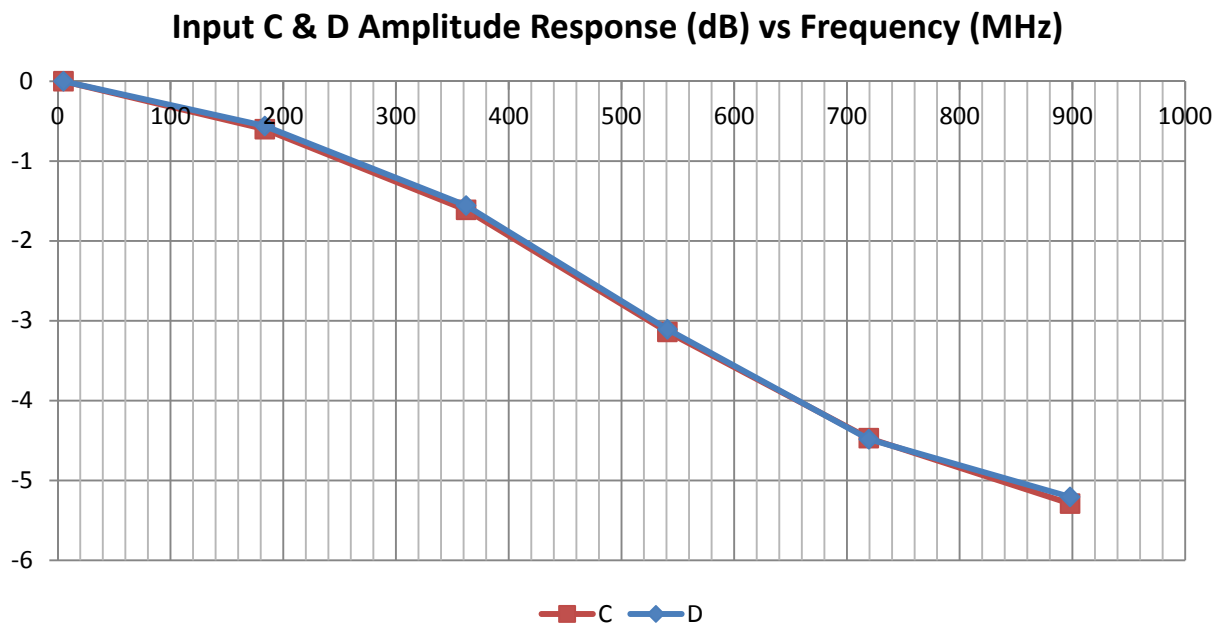


Figure 3: Typical Amplitude Response vs Frequency

5.1.2 SNR, SFDR, and THD Plots

The following plots provide the basis for the numbers in the typical performance table above.

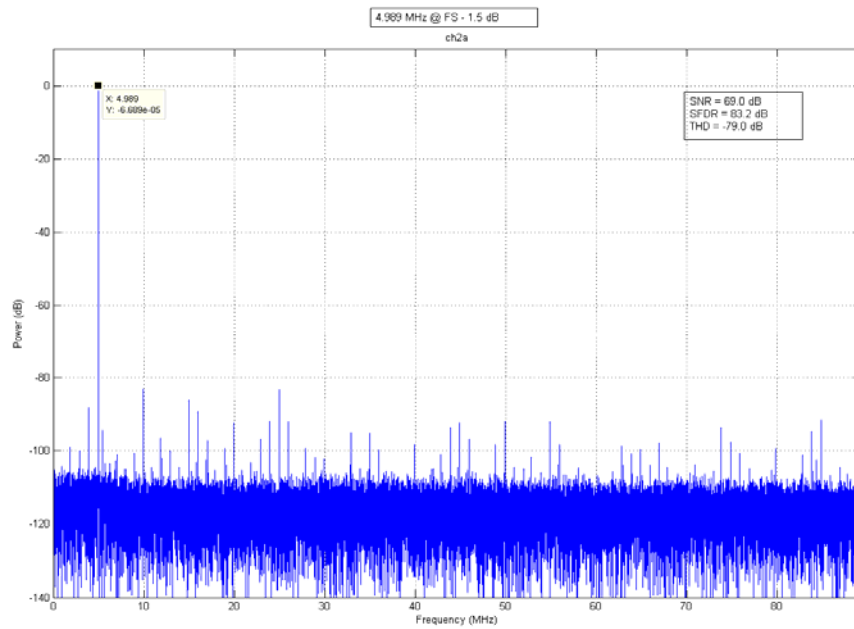


Figure 4: typical spectrum, Fin=4.989 MHz, Amp.=FS - 1.5dB (Nyquist Zone 1)

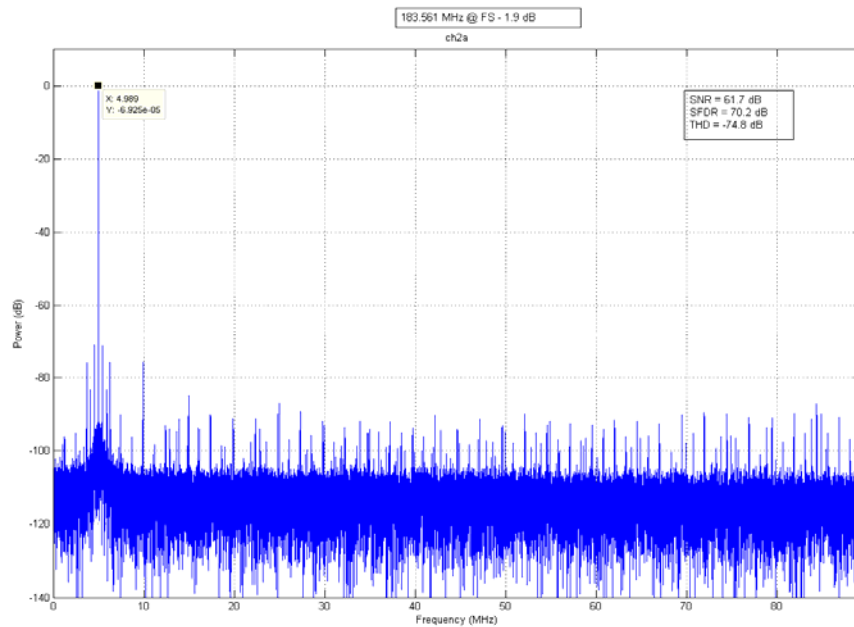


Figure 5: typical spectrum, Fin=183.561 MHz, Amp.=FS - 1.9dB (Nyquist Zone 3)

5.1.3 Adjacent Channel Coupling

To limit coupling, the FSF-AD8200A has strategic ground plane design as well as individual front-end shields. The following information describes the typical measured coupling between channels. When referred to, adjacent and second adjacent are defined as:

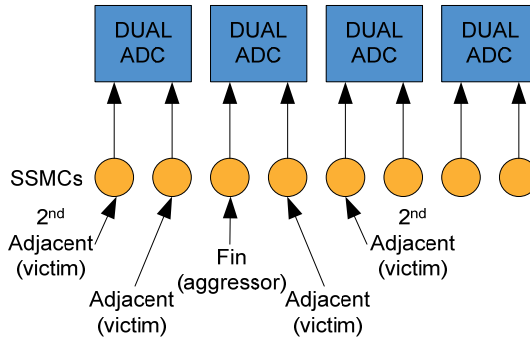


Figure 6: Coupling terminology explanation

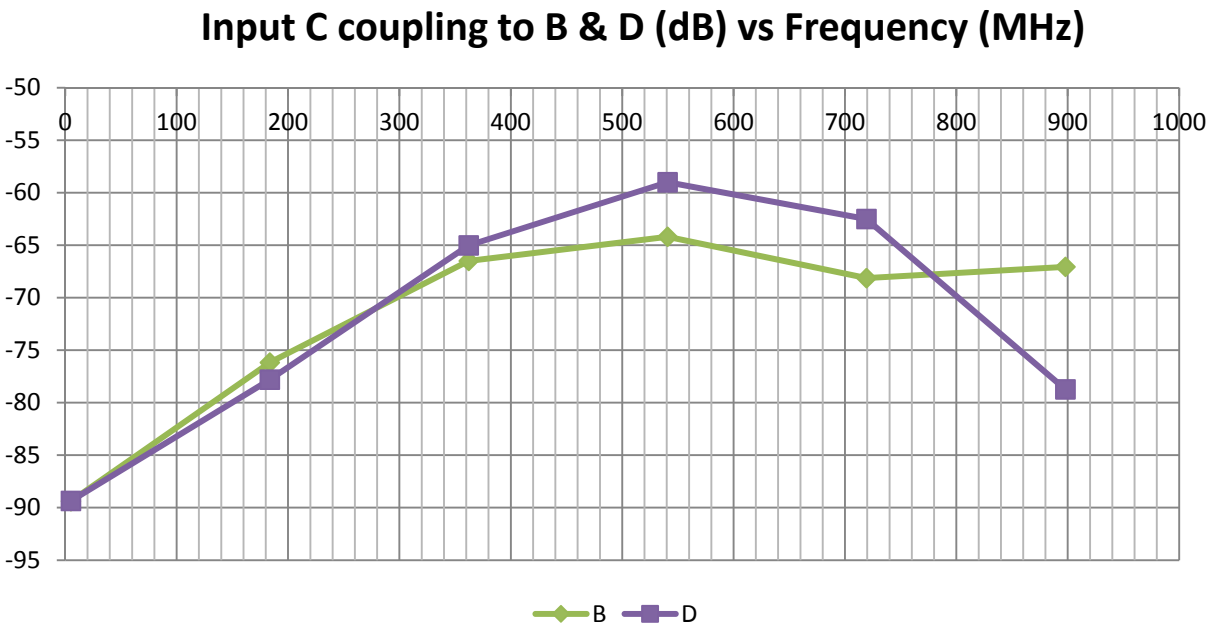


Figure 7: Typical Channel – Channel Crosstalk (input C to adjacent channels)

Input C coupling to A, E, F, G, H (dB) vs Frequency (MHz)

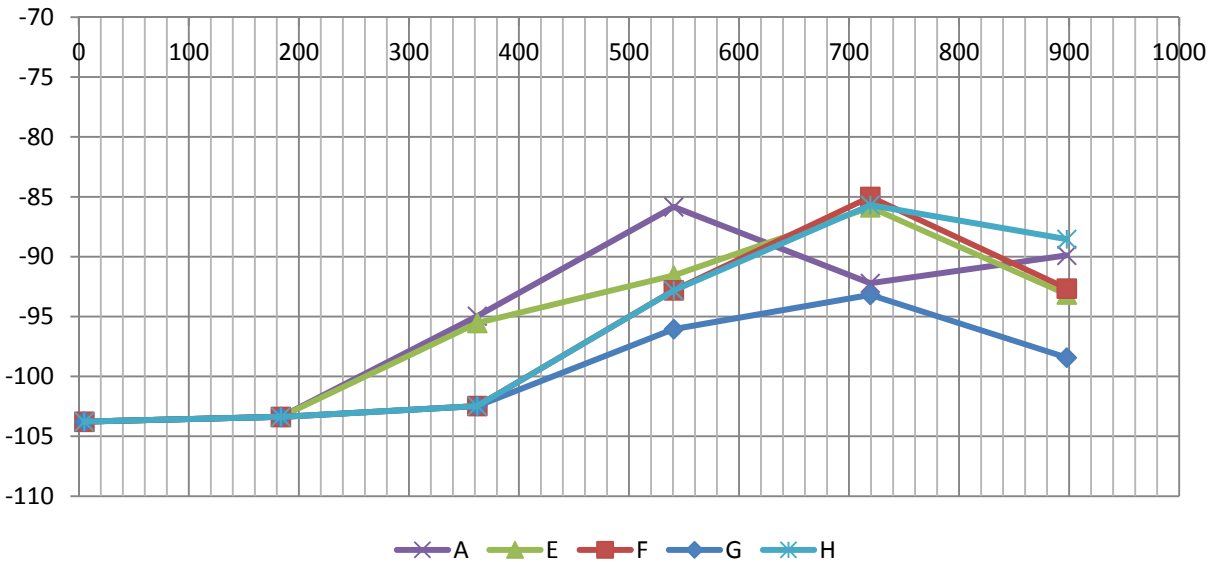


Figure 8: Typical Channel – Channel Crosstalk (input C to non-adjacent channels)

Input D coupling to C & E (dB) vs Frequency (MHz)

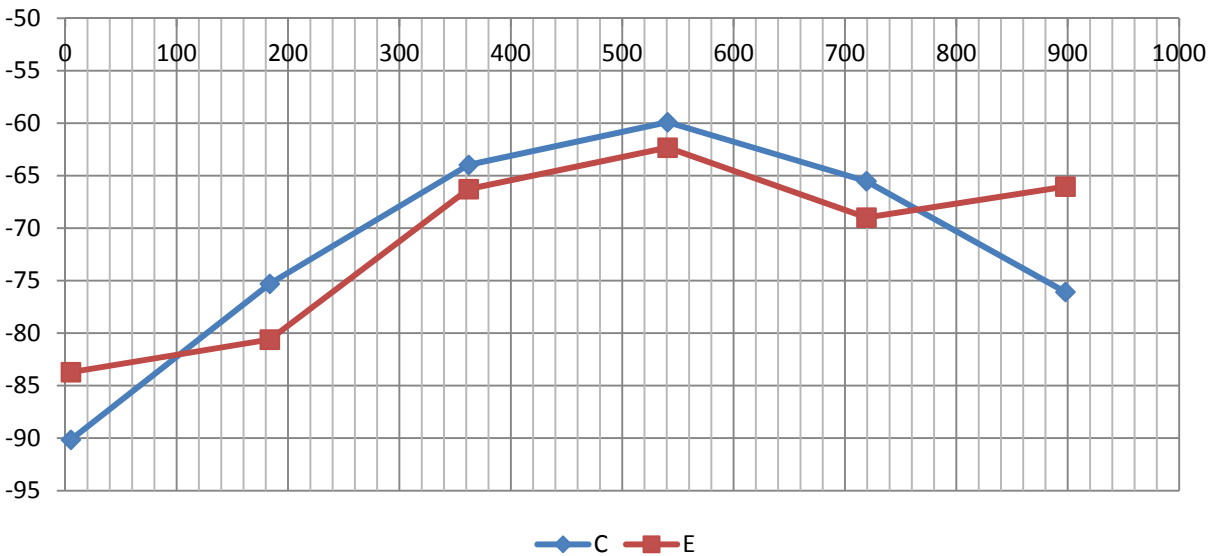


Figure 9: Typical Channel – Channel Crosstalk (input D to adjacent channels)

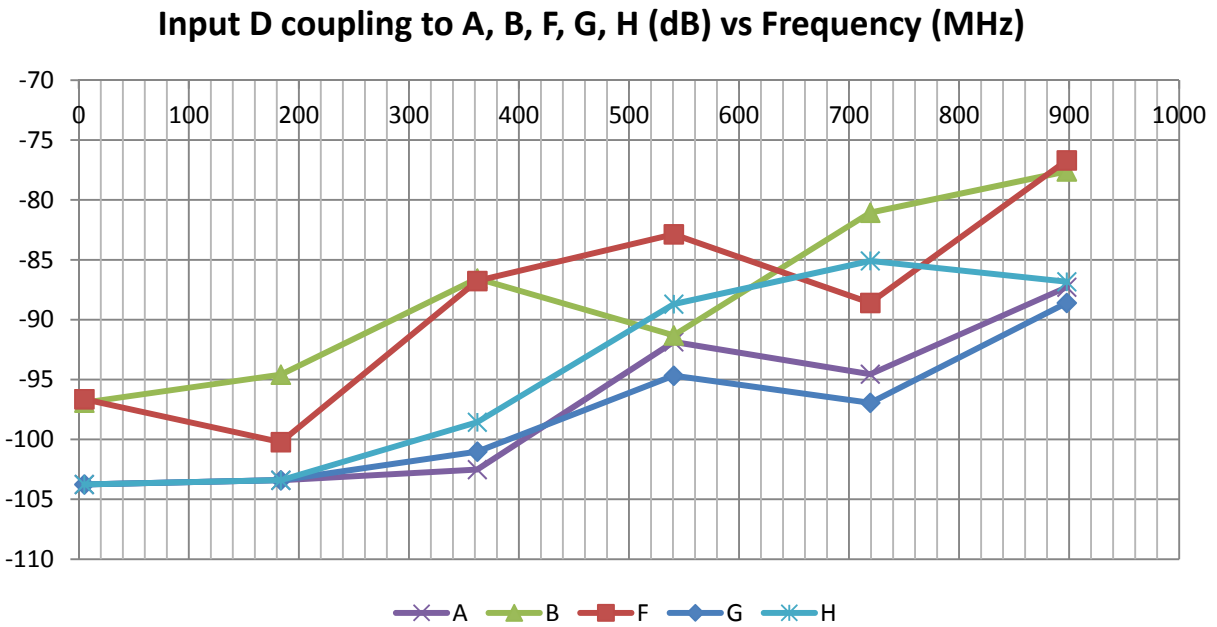


Figure 10: Typical Channel – Channel Crosstalk (input D to non-adjacent channels)

5.1.4 Intermodulation

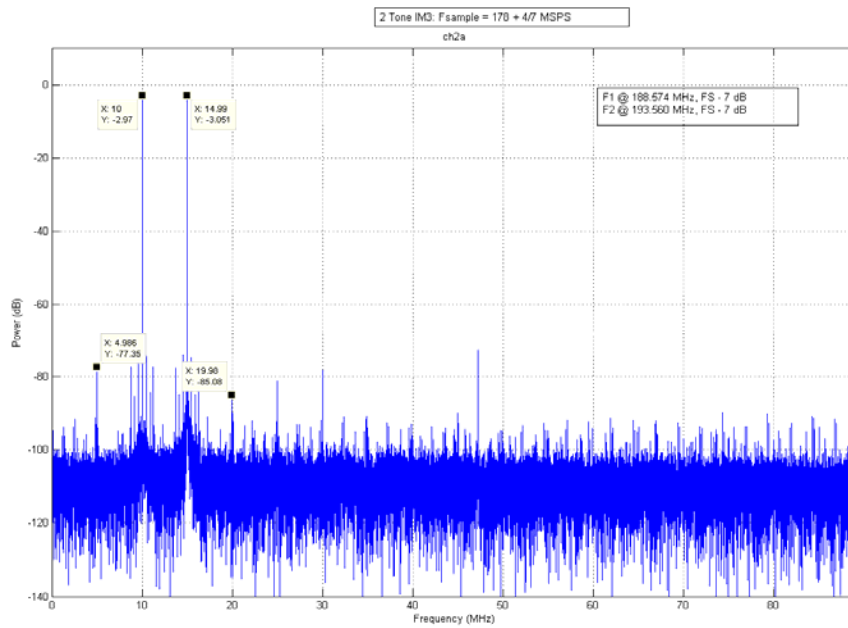


Figure 11: Two-tone IM3, F1=188.574MHz, F2=193.560MHz, amp=FS-7dB

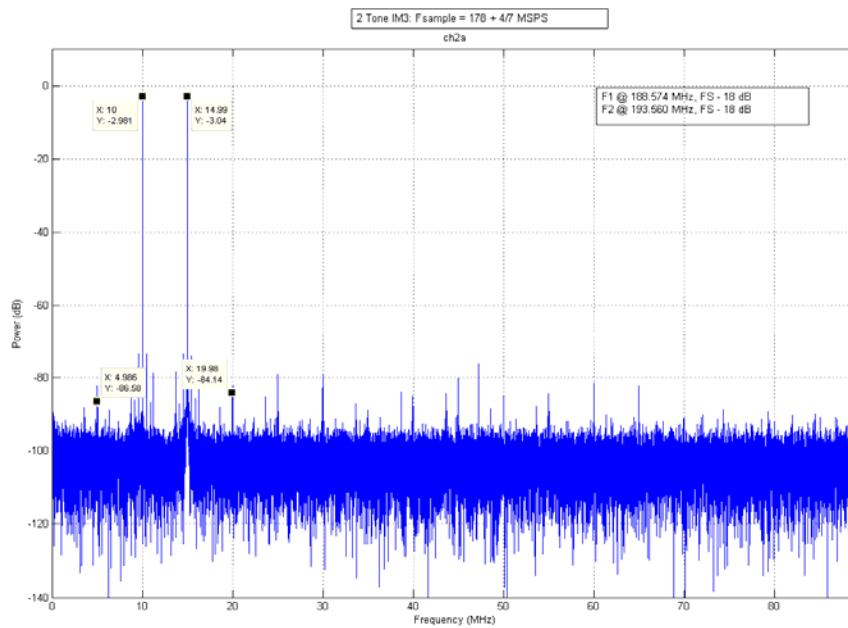


Figure 12: Two-tone IM3, F1=188.574MHz, F2=193.560MHz, amp=FS-18dB

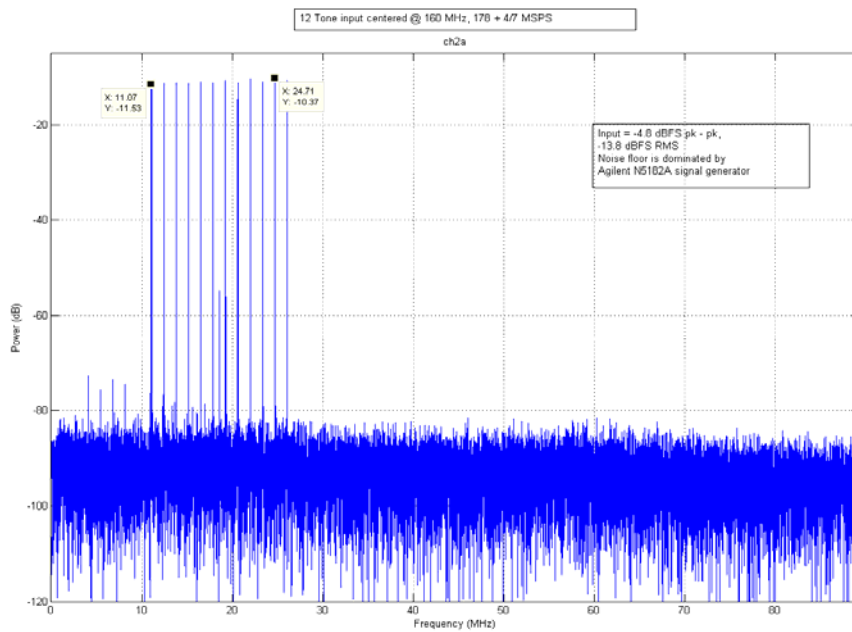


Figure 13: Multi-tone, centered @ 160MHz, amp= -13.8dBFS RMS (Nyquist Zone 2)

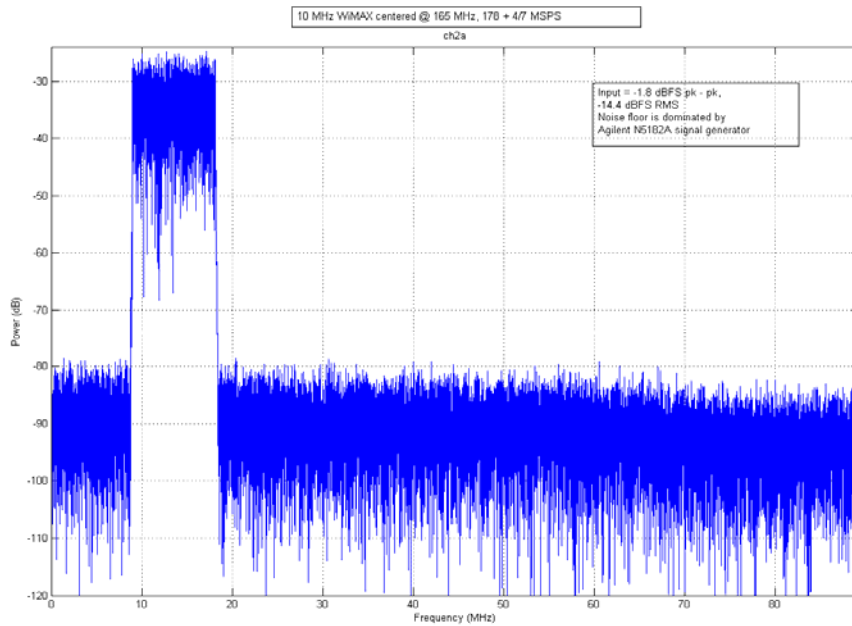


Figure 14: OFDM, center=165MHz, BW=10MHz, amp= -14.4dBFS RMS (Nyquist Zone 2)

6. INTERFACES

The FSF-AD8200A is comprised of both external and internal interfaces. The following interfaces are discussed in this section.

- FMC interface and pin-out
- ADC analog inputs
- External clock and trigger inputs
- SPI interface to the clock generator
- SPI interface to the ADC devices

6.1 FMC Interface: Pin support and other requirements

The FSF-AD8200A requires that the FMC carrier card contain a suitable HPC FMC connector. As the FMC standard allows designers a certain amount of flexibility in pin selection and voltage support, just because a carrier card as an HPC connector, it does not mean that it will be compatible with every FMC with an HPC connector. Compatibility must be carefully assessed on a case-by-case basis.

The following sub-sections describe the support that the FSF-AD8200A requires from the carrier card.

6.1.1 FSF-AD8200A HPC FMC pin assignment and definition

FMC HPC Connector Details: Unshaded Cells are “No Connect” (NC) and list the VITA pin name. Shaded Cells list the VITA pin name followed by FSF-AD8200A connection details.

| | K | J | H | G | F | E | D | C | B | A |
|----|------------------------------|------------------------------|-------------------------------------|--------------------------------------|-----------|-------------------|---------------------------------------|-----------------------------------|--------------------------------------|---------------------------|
| 1 | VREF_B_M2C | GND | VREF_A_M2C | GND | PG_M2C | GND | PG_C2M | GND | CLK_DIR: 3P3V_FMC_10K_PU | GND |
| 2 | GND | CLK3_BIDIR_P | PRSN_T_M2C_L: PRSN_T_M2C_L(GND) | CLK1_M2C_P: M2C_FPGA_LVDS_SYSREFP | GND | HA01_P_CC | GND | DP0_C2M_P | GND | DP1_M2C_P: ADC3B_CML_P |
| 3 | GND | CLK3_BIDIR_N | GND | CLK1_M2C_N: M2C_FPGA_LVDS_SYSREFN | GND | HA01_N_CC | GND | DP0_C2M_N | GND | DP1_M2C_N: ADC3B_CML_N |
| 4 | CLK2_BIDIR_P | GND | CLK0_M2C_P: M2C_FPGA_LVDS3_CLKp | GND | HA00_P_CC | GND | GBTCLK0_M2C_P: M2C_FPGA_LVDS1_CLKp | GND | DP9_M2C_P | GND |
| 5 | CLK2_BIDIR_N | GND | CLK0_M2C_N: M2C_FPGA_LVDS3_CLKn | GND | HA00_N_CC | GND | GBTCLK0_M2C_N: M2C_FPGA_LVDS1_CLKn | GND | DP9_M2C_N | GND |
| 6 | GND | HA03_P | GND | LA00_P_CC: ADC2_LVDS_FRAMEP | GND | HA05_P | GND | DP0_M2C_P:ADC3A_CML_P | GND | DP2_M2C_P: ADC4A_CML_P |
| 7 | HA02_P | HA03_N | LA02_P:C2M_VADJ_ADC4 1_SCRAMBLER | LA00_N_CC: ADC2_LVDS_FRAMEN | HA04_P | HA05_N | GND | DP0_M2C_N:ADC3A_CML_N | GND | DP2_M2C_N: ADC4A_CML_N |
| 8 | HA02_N | GND | LA02_N: M2C_VADJ_EXT_TRIGGER | GND | HA04_N | GND | LA01_P_CC: ADC3_LVDS_FRAMEP | GND | DP8_M2C_P | GND |
| 9 | GND | HA07_P | GND | LA03_P | GND | HA09_P | LA01_N_CC: ADC3_LVDS_FRAMEN | GND | DP8_M2C_N | GND |
| 10 | HA06_P | HA07_N | LA04_P | LA03_N | HA08_P | HA09_N | GND | LA06_P | GND | DP3_M2C_P: ADC4B_CML_P |
| 11 | HA06_N | GND | LA04_N | GND | HA08_N | GND | LA05_P | LA06_N | GND | DP3_M2C_N: ADC4B_CML_N |
| 12 | GND | HA11_P | GND | LA08_P: C2M_CLKGEN_VADJ_CSn | GND | HA13_P | LA05_N | GND | DP7_M2C_P: ADC1A_CML_P | GND |
| 13 | HA10_P | HA11_N | LA07_P:C2M_VADJ_SCLK | LA08_N: C2M_ADC1_VADJ_CSn | HA12_P | HA13_N | GND | DP7_M2C_N: ADC1A_CML_N | GND | GND |
| 14 | HA10_N | GND | LA07_N: C2M_VADJ_SDIO_DIR | GND | HA12_N | GND | LA09_P: C2M_ADC2_VADJ_CSn | LA10_P: C2M_ADC4_VADJ_CSn | GND | DP4_M2C_P: ADC2B_CML_P |
| 15 | GND | HA14_P | GND | LA12_P: CLKGEN_VADJ_MAN_SYNC | GND | HA16_P | LA09_N: C2M_ADC3_VADJ_CSn | LA10_N: C2M_VADJ_SDIO_ADC | GND | DP4_M2C_N: ADC2B_CML_N |
| 16 | HA17_P_CC | HA14_N | LA11_P: C2M_VADJ_SDIO_CLKGE N | LA12_N:nPOWER_FAULT | HA15_P | HA16_N | GND | GND | DP6_M2C_P: ADC1B_CML_P | GND |
| 17 | HA17_N_CC | GND | LA11_N: CLKGEN_VADJ_RESETn | GND | HA15_N | GND | LA13_P: C2M_VADJ_SWITCHER_SYNQ1 | GND | DP6_M2C_N: ADC1B_CML_N | GND |
| 18 | GND | HA18_P | GND | LA16_P | GND | HA20_P | LA13_N: CARD_ID_4K12_PD | LA14_P: C2M_VADJ_SWITCHER_SYNQ | GND | DP5_M2C_P: ADC2A_CML_P |
| 19 | HA21_P | HA18_N | LA15_P | LA16_N | HA19_P | HA20_N | GND | LA14_N | GND | DP5_M2C_N: ADC2A_CML_N |
| 20 | HA21_N | GND | LA15_N | GND | HA19_N | GND | LA17_P_CC: ADC4_LVDS_FRAMEP | GND | GBTCLK1_M2C_P: M2C_FPGA_LVDS2_CLK | GND |
| 21 | GND | HA22_P | GND | LA20_P | GND | HB03_P | LA17_N_CC: ADC4_LVDS_FRAMEN | GND | GBTCLK1_M2C_N: M2C_FPGA_LVDS2_CLK | GND |
| 22 | HA23_P | HA22_N | LA19_P | LA20_N | HB02_P | HB03_N | GND | LA18_P_CC: ADC1_LVDS_FRAMEP | GND | DP1_C2M_P |
| 23 | HA23_N | GND | LA19_N | GND | HB02_N | GND | LA23_P | LA18_N_CC: ADC1_LVDS_FRAMEN | GND | DP1_C2M_N |
| 24 | GND | HB01_P | GND | LA22_P | GND | HB05_P | LA23_N | GND | DP9_C2M_P | GND |
| 25 | HB00_P_CC | HB01_N | LA21_P | LA22_N | HB04_P | HB05_N | GND | GND | DP9_C2M_N | GND |
| 26 | HB00_N_CC | GND | LA21_N | GND | HB04_N | GND | LA26_P | LA27_P | GND | DP2_C2M_P |
| 27 | GND | HB07_P | GND | LA25_P | GND | HB09_P | LA26_N | LA27_N | GND | DP2_C2M_N |
| 28 | HB06_P_CC | HB07_N | LA24_P | LA25_N | HB08_P | HB09_N | GND | GND | DP8_C2M_P | GND |
| 29 | HB06_N_CC | GND | LA24_N | GND | HB08_N | GND | TCK | GND | DP8_C2M_N | GND |
| 30 | GND | HB11_P | GND | LA29_P | GND | HB13_P | TDI: FMC_TDI | SCL: FMC_SCL | GND | DP3_C2M_P |
| 31 | HB10_P | HB11_N | LA28_P | LA29_N | HB12_P | HB13_N | TDO: FMC_TDO | SDA: FMC_SDA | GND | DP3_C2M_N |
| 32 | HB10_N | GND | LA28_N | GND | HB12_N | GND | 3P3VAUX | GND | DP7_C2M_P | GND |
| 33 | GND | HB15_P | GND | LA31_P | GND | HB19_P | TMS | GND | DP7_C2M_N | GND |
| 34 | HB14_P | HB15_N | LA30_P | LA31_N | HB16_P | HB19_N | TRST_L | GA0: GA0_1K5_S | GND | DP4_C2M_P |
| 35 | HB14_N | GND | LA30_N | GND | HB16_N | GND | GA1: GA1_1K5_S | 12P0V: 12P0V_FMC | GND | DP4_C2M_N |
| 36 | GND | HB18_P | GND | LA33_P | GND | HB21_P | 3P3V: 3P3V_FMC | GND | DP6_C2M_P | GND |
| 37 | HB17_P_CC | HB18_N | LA32_P | LA33_N | HB20_P | HB21_N | GND | 12P0V: 12P0V_FMC | DP6_C2M_N | GND |
| 38 | HB17_N_CC | GND | LA32_N | GND | HB20_N | GND | 3P3V: 3P3V_FMC | GND | GND | DP5_C2M_P |
| 39 | GND | VIO_B_M2C: FMC_VADJ_OK0_S | GND | VADJ: FMC_VADJ | GND | VADJ: FMC_VADJ | GND | 3P3V: 3P3V_FMC | GND | DP5_C2M_N |
| 40 | VIO_B_M2C: FMC_VADJ_OK0_S | GND | VADJ: FMC_VADJ | GND | FMC_VADJ | GND | 3P3V: 3P3V_FMC | GND | RES0 | GND |

Table 3: FMC Pinout

The following table defines the signals, the I/O type, and their usage:

| Signal(s) | Type | Dir | Description |
|-----------------------------------|--------|-------|---|
| JESD204B Interface | | | |
| ADC[4:1][B:A]_CML[P/N] | CML | M2C | JESD204B data signals, AC-coupled |
| M2C_FPGA_LVDS_SYSREF[P/N] | LVDS | M2C | JESD204B SYSREF signal, AC-coupled |
| ADC[4:1]_LVDS_FRAME[P/N] | LVDS | C2M | JESD204B SYNC, DC-coupled |
| M2C_FPGA_LVDS1_CLK[P/N] | LVDS | M2C | Clock intended for JESD204B core reference clock (typically not required) |
| M2C_FPGA_LVDS2_CLK[P/N] | LVDS | M2C | Clock intended for JESD204B core reference clock (to be half of the sampling frequency, by default 90MHz) |
| M2C_FPGA_LVDS3_CLK[P/N] | LVDS | M2C | Clock intended for JESD204B core reference clock (typically not required) |
| SPI Bus | | | |
| C2M_ADC[4:1]_VADJ_CS _n | VADJ | C2M | Chip selects for ADCs |
| C2M_CLKGEN_VADJ_CS _n | VADJ | C2M | Chip select for clock generator |
| C2M_VADJ_SCLK | VADJ | C2M | SPI CLOCK |
| C2M_VADJ_SDIO_CLKGEN | VADJ | BIDIR | LMK04828B clock generator SDIO signal |
| C2M_VADJ_SDIO_ADC | VADJ | BIDIR | ADC[4:1] SDIO signal |
| C2M_VADJ_SDIO_DIR | VADJ | C2M | SDIO VADJ buffer direction control signal |
| I²C Bus | | | |
| | | | <i>The FMC specified EEPROM is the only device connected to the I²C bus</i> |
| FMC_SCL | LVTTTL | C2M | I ² C bus clock |
| FMC_SDA | LVTTTL | BIDIR | I ² C bidirectional data |
| GA[1:0] | LVTTTL | C2M | Geographic addresses used for onboard EEPROM only |
| JTAG | | | |
| | | | JTAG unused, TCK, TMS, TRST_L all unconnected |
| FMC_TDI | LVTTTL | C2M | Direct to TDO (unused on FMC) |
| FMC_TDO | LVTTTL | M2C | Direct to TDI (unused on FMC) |
| MISC CONTROL | | | |
| C2M_VADJ_SWITCHER_SYNQ[2:1] | VADJ | C2M | Clock signals used to synchronize the two onboard switchers. Default is 450kHz and each are expected to be 180° out of phase. |
| CLKGEN_VADJ_RESET _n | VADJ | C2M | 0 = Reset clock generator 1 = Clock generator active |
| CLKGEN_VADJ_MAN_SYNC _n | VADJ | C2M | Synchronization signal to LMK04828B clock generator (typically not required) |
| C2M_VADJ_ADC4-1_SCRAMBLER | VADJ | C2M | 0 = Disable ADC scramblers 1 = Enable ADC scramblers |
| M2C_VADJ_EXT_TRIGGER | VADJ | M2C | Trigger signal from front bezel. Either rising or falling edge active depending on FPGA code |
| CARD_ID | LVTTTL | M2C | 4.12kΩ to ground |

| | | | |
|---|------------|------------|---|
| CLK_DIR | LVTTTL | M2C | 10kΩ to 3P3V |
| PG_C2M | LVTTTL | C2M | 0 = Disable power on FMC 1 = Enable power on FMC |
| PG_M2C | LVTTTL | M2C | Unconnected |
| nPOWER_FAULT | VADJ | M2C | 0 = Power fault on FMC 1 = Power OK on FMC |
| Power Rails | | | |
| | <i>Min</i> | <i>Max</i> | |
| VADJ | 1.5 | 3.6 | Volts. Range provided by level-shifters on input of all single ended signals. |
| VIO_B_M2C | | | Connected to VADJ |
| VREF_A_M2C | | | Not connected |
| VREF_B_M2C | | | Not connected |
| 3P3V_AUX | 3.135 | 3.465 | Volts |
| 3P3V | 3.135 | 3.465 | Volts |
| 12P0V | 11.4 | 12.6 | Volts |
| Notes: | | | |
| a) Type VADJ indicates that the input and output thresholds are governed by the level shifters which are powered by VADJ. The level shifting system is comprised of two different level shifters from Texas Instruments: SN74AVC1T45 and SN74AVC8T245. As the thresholds depend on the value of VADJ, please refer to the respective Texas Instruments datasheets for complete threshold information. | | | |

Table 4: FMC Signal Definition

6.1.2 FMC Voltage and Current requirements

The following table lists the voltages and associated currents used by the FSF-AD8200A, including those required to be listed by VITA 57.1:

| FMC HPC Pins | Voltage (V) | | | Current (mA) | | |
|--------------|-------------|-----|-------|--------------|-----|-----|
| | Min | Nom | Max | Min | Nom | Max |
| 12P0V | 11.4 | 12 | 12.6 | | | tbd |
| 3P3V | 3.135 | 3.3 | 3.465 | | | tbd |
| VADJ | 1.5 | | 3.465 | | | tbd |
| 3P3VAUX | 3.135 | 3.3 | 3.465 | | | tbd |

Table 5: FSF-AD8200A Voltages and Currents

6.1.3 Multi-Gigabit Serial Communications Lines

The JESD204B defines a high-speed serial standard by which information is passed from a transmitter to a receiver. In the case of the FSF-AD8200A, the transmitters are the onboard ADCs, and then receiver is the FPGA.

Each ADC sends its conversion data over high-speed serial “current mode logic” lines CML_P / CML_N. These are received by multi-gigabit transceiver (MGT) IO in the FPGA and are compatible with JESD204B protocol requirements. The bit rate carried on each serial interface is 20-times³ the

³ Due to framing and other overhead, the serial line rate is greater than simply the sample rate multiplied by the converter resolution

ADC sample rate, yielding a 3.6Gbps rate on each of the 8 lanes, and providing an aggregate data bandwidth of 28.8Gbps.

It is important to note that FMC carrier cards are not required to support all 10 of the gigabit transceiver lanes specified in the FMC HPC specification. It is always important to validate the number of MGTs connected to the FMC connector.

6.2 ADC Analog Inputs

The 8 analog inputs of the FSF-AD8200A (labeled “A” through “H”) are each ESD protected and AC coupled to a 50Ω termination. The following table describes the expected operation of these inputs.

| Parameter | Min | Typ | Max | Unit | Conditions/Comments |
|-----------------------------|-----|-----------------|------|------|--|
| Connector Type | | 50Ω SSMC | | | |
| Input Impedance | | 50 | | Ω | |
| Input Return Loss | | 20 ⁴ | | dB | 5MHz to 140MHz |
| Absolute (do not exceed) | | | 4 | Vpp | Peak power of 19dBm |
| FS (Full-Scale) Input Range | | | 2.25 | Vpp | By default, but each ADC can be programmed to lower FS values in 1dB steps |

Table 6: Analog Input Requirements

Note that a small amount of sampling clock energy (including harmonics) leaks out from each ADC input and appears at the corresponding SSMC connector. This is normal for un-buffered high speed ADCs and is caused by the internal sampling switches. For the FSF-AD8200A with the ADC1443D devices, this level is typically -47dBm +/-3 dB at the second harmonic of the sampling frequency as measured at each of the 8 SSMC connectors.

6.3 External Clock Input

The External Clock SSMC connector input (labeled “CL”) is ESD protected and AC coupled to a 50Ω termination before proceeding to the clock generator chip. The external clock input supports both sine and square wave inputs as per the requirements below:

| Parameter | Min | Typ | Max | Unit | Conditions/Comments |
|--------------------------|-----|-----|------|------|---------------------|
| <i>Input Type</i> | | 50 | | Ω | |
| Absolute (do not exceed) | | | 2.4 | Vpp | |
| Sinewave | | | 11.5 | dBm | measured on 50Ω |
| Squarewave | | | 14.5 | dBm | measured on 50Ω |
| Slew Rate | 500 | | | V/μs | |
| Duty Cycle | 40 | | 60 | % | |

Table 7: External Clock Input Requirements

⁴ Refer to Input Return Loss plot: Figure 2

This input can be left open and the 8-channel capture application included with the FSF-AD8200A will operate correctly, but with a small frequency error in the ADC sampling clock (typically no more than ± 25 ppm). If synchronization of the ADCs' sampling clock frequency with an external reference is desired, the External Clock input can be driven with a 10 MHz source. This is commonly available at the back of many signal generators and is the reference to which output sine waves from the signal generator are locked. Supplying this 10 MHz reference to the External Clock Input will precisely lock the FSF-AD8200A ADC sampling clocks to the 10 MHz reference frequency. This capability is useful when post-capture FFT analysis is to be performed on sine wave inputs applied to the ADCs. If you would like to learn more about this capability, contact Fidus Systems for related documentation or assistance with other synchronization options.

6.4 External Trigger Input

The External Trigger SSMC connector input (labeled "TR") is terminated in a shunt 4.12k Ω resistor and ESD protection device. The signal is DC-coupled to a buffer gate that sends its output through a 22 Ω series termination directly to the FMC connector.

External Trigger Input Levels specification:

| Parameter | Min | Typ | Max | Unit | Conditions/Comments |
|--------------------------|------|--------------|-----|------|---------------------|
| <i>Input Type</i> | | LVCMOS/LVTTL | | | |
| Absolute (do not exceed) | -0.5 | | 6.5 | V | |
| Vih | 1.2 | | | V | |
| Vil | | | 0.6 | V | |
| Slew Rate | 50 | | | V/μs | |

Table 8: External Trigger Input Requirements

Note that the standard FPGA configuration supplied with the FSF-AD8200A does not make use of signals from the External Trigger input. Contact Fidus Systems if you need help applying this input in your own application.

6.5 Communication and Control

The carrier card’s FPGA is responsible for configuring and controlling the FSF-AD8200A FMC card. The majority of this supervisory activity is completed by SPI bus, however, there are also other signals that make up this host interface. These control signals are discussed in this section.

6.5.1 SPI Interface

The SPI bus provides the main control system for the onboard clock generator and each individual ADC on the FSF-AD8200A. It is important to note that this is a 3-wire SPI bus implementation, thus SDIO is bidirectional and directionality must be carefully considered. To avoid bus contention, it is critically important to control the SDIO direction at the appropriate time. The following schematic capture shows the SDIO level-shifting system (VADJ ↔ 1.8 or 3.3V) so that the FPGA designer understands how to avoid SDIO bus contention issues- C2M VADJ SDIO DIR must be switched at the appropriate time.

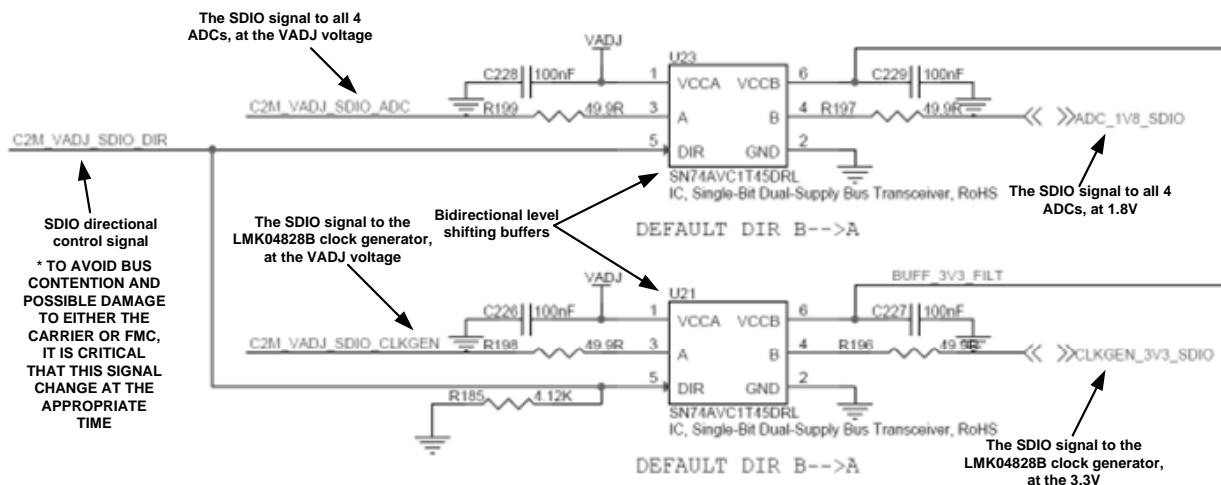


Figure 15: SDIO buffer direction control

6.5.1.1 Clock Generator

Figure 16 presents the SPI format details for communications with Texas Instrument’s LMK04828B clock generator/distribution device. The CS* line is active low and corresponds to the _CSn signal on pin G12 (LA08_P) of the FMC connector:

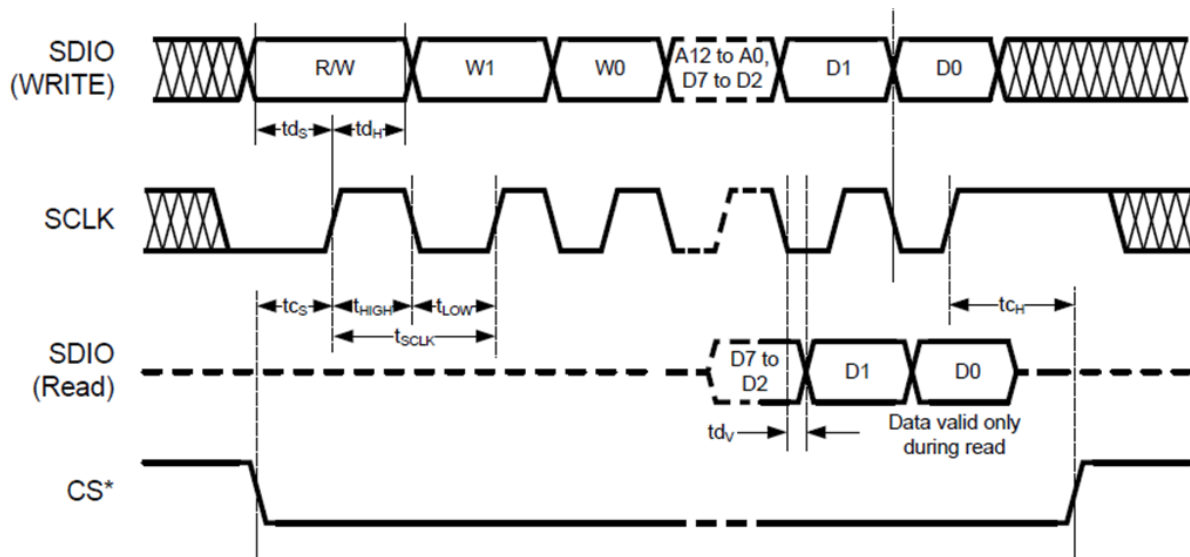


Figure 16: SPI Communications with the LMK04828B Clock Generator

Data applied to the SDIO pin is clocked into the device on the rising edges of SCK. A rising CS* line completes the SPI transaction.

Note that activity on any of the SPI lines while the LMK04828B is locked may degrade the phase noise of the output clocks. This can happen when device registers are being accessed, or to a lesser degree, when other devices sharing the same level translator IC have SPI transactions occurring. The design of the FSF-AD8200A isolates the FMC SPI lines from the device SPI lines in an effort to minimize SPI noise contamination. A slew rate of 30 volts/ μ sec or faster is recommended for all SPI signals.

The software code contained within the provided bitfile provides an example default configuration for SPI communications and register configuration for the LMK04828B. Other configuration settings are of course possible. Refer to the LMK04828B datasheet for more details on SPI communications and appropriate register settings, or contact Fidus Systems if you would like assistance in your application.

6.5.1.2 Analog to Digital Converters

Figure 17 presents the SPI format details for communications with each of the ADC1443D analog-to-digital converter devices. The SCS_N line is active low and corresponds to the four individual _CSn signals (one for each ADC) from the FMC connector:

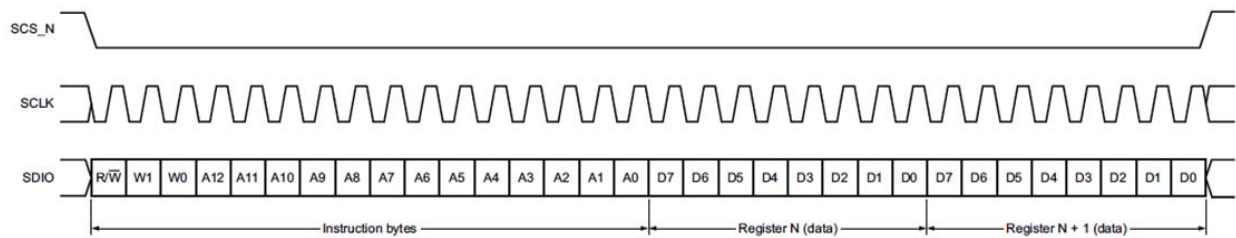


Figure 17: SPI Communications with the ADC1443D Converters

The software code contained within the provided bitfile provides an example default configuration for SPI communications and register configuration for the ADC1443D. Other configuration settings are of course possible. Refer to the ADC1443D datasheet for more details on SPI communications and appropriate register settings, or contact Fidus Systems if you would like assistance in your application.

6.5.2 EEPROM

The FSF-AD8200A contains non-volatile storage as defined by the FMC standard. The EEPROM is solely accessed via an I²C bus and is mastered by the carrier card’s FMC. The EEPROM contains critical information that describes the operational demands of the FMC. The end-user must ensure that their application first reads the contents of the EEPROM to ensure that the carrier card will be compatible prior to enabling power to the FMC connector site. If this step is ignored and compatibility is not validated prior to powering, either the carrier card or the FMC or both, could be permanently damaged or destroyed.

As per the VITA 57.1 standard, the EEPROM is powered by 3V3AUX, thus enabling communication with the EEPROM independent of any other onboard voltages.

| EEPROM Information | |
|--------------------|---|
| Manufacturer | NXP |
| Part Number | PCF85102C-2T/03 |
| Description | 2kbit, I ² C EEPROM |
| Address | 0b10100xx; where xx is specified by the GA[0:1] signals from the carrier card |

Table 9: EEPROM Information

[EEPROM CONTENTS AVAILABLE SOON]

7. ENVIRONMENTAL AND MECHANICAL

7.1 Environment

The FSF-AD8200A is primarily designed for laboratory experimentation and development. It is specified for operation as follows:

| Parameter | Min | Typ | Max | Unit | Conditions/Comments |
|--|-------------------------------|-----|-----|------|--|
| Ambient Operating Temperature ⁵ | 0 | | 45 | °C | This is the guaranteed operating range, and that which the FSF-AD8200A is verified to |
| FSF-AD8200A individual component ratings | -40 | | +85 | °C | This is the basic temperature rating of the components used in the design as per the component's datasheet |
| ESD | Not evaluated | | | | A level of ESD protection is present on each front-panel connector |
| Shock/Vibration | Not designed for or evaluated | | | | User responsibility |
| Salt Spray | Not designed for or evaluated | | | | User responsibility |
| Chemical | Not designed for or evaluated | | | | User responsibility |
| RoHS Compliant | YES | | | | |

Table 10: Environmental Operating Conditions

7.2 Thermal Considerations

The FSF-AD8200A is designed as a conduction cooled, single width, FMC. That being said, it is impossible to predict every conduction cooled situation. It is also very possible that conduction cooling will not be supported by the selected carrier card. Ultimately, it is the user's responsibility to ensure that their cooling solution ensures that they are not exceeding the temperature requirements of the FSF-AD8200A components (this would void the warranty).

To mitigate thermal concerns, each FSF-AD8200A comes with a heat sink (shown installed below). This heat sink was designed to maintain appropriate thermal margins in a natural convection environment with ambient temperatures reaching a maximum of 45 °C. Thermal margins can be maintained at higher ambient temperatures if the user employs forced air flow appropriately.

⁵ Operation over the industrial temperature range of -40 to +85 °C is possible but not warranted (all components are rated for this range, but thermal margins will depend on conduction and airflow).



Figure 18: FSF-AD8200A with heat sink installed

There may be scenarios where the end user must remove the heat sink (e.g. carrier card interference). In this situation the user must provide forced air, and is solely responsible for ensuring that the board maintains adequate thermal margins.

In summary, the FSF-AD8200A with its heat sink installed does not require conduction cooling and can operate reliably in a natural convection environment up to 45 °C. Removing the heat sink is not recommended, but if required ensure and monitor that sufficient air flow is applied for the given, ambient temperature conditions.

| Parameter | Min | Typ | Max | Unit | Conditions/Comments |
|---------------------------|-----|-----|-----|------|---|
| Onboard Power Dissipation | | TBD | | W | All converters active and sampling at 180MSPS |

Table 11: Power Dissipation

7.3 Mechanical

7.3.1 Front Bezel (Faceplate)

The following diagram illustrates the FSF-AD8200A’s front bezel construction. The front bezel is attached to the FMC by two M2.5 machine screws.

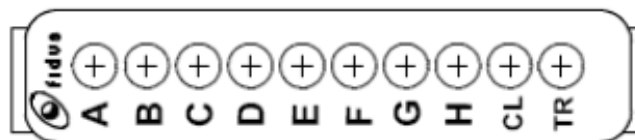


Figure 19: FSF-AD8200A Front Bezel

7.3.2 Heat Sink

The following diagram illustrates the FSF-AD8200A's custom heat sink. Compressible thermal pads provide the buffer and ensure a good thermal connection between the onboard components and the heat sink.

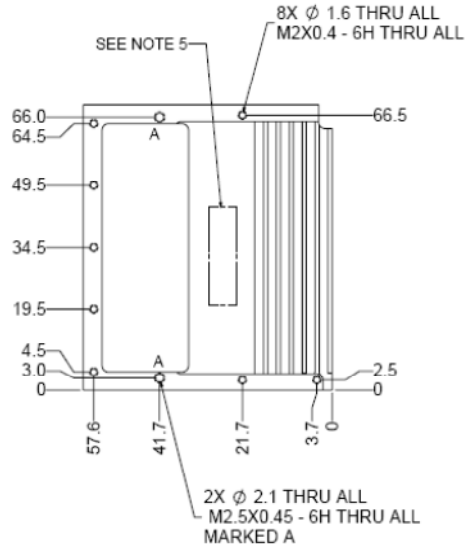


Figure 20: FSF-AD8200A Heat Sink

The heat sink is connected to the carrier card or conduction cooling solution using ten machine screws (8x M2, 2x M2.5). Loosely fit the heat sink on with all screws first, then proceed to tighten in a criss cross pattern to help ensure the equal pressure. Avoid over-tightening as this may result in damage to the heat sink, the circuit board and/or the carrier card.

8. DEMONSTRATION

This section describes how to experiment with the FSF-AD8200A paired with a VC707, using the Fidus provided bitfile. For detailed instructions refer to “FSF-AD8200A Quick Start Guide”.

8.1 Hardware Installation

Installation of the FSF-AD8200A mezzanine card is as simple as plugging the FMC connector onto a carrier card with a compatible HPC FMC connector⁶ – all power is supplied through the FMC connection. It is recommended that the carrier card not be powered when the FSF-AD8200A is installed or removed. Follow normal ESD prevention procedures, and avoid flexing both the mezzanine and carrier boards as much as possible. For the VC-707, use the FMC connector closest to the center of the board, labeled “FMC2 HPC”.

Once the HPC FMC connector is seated, power can be applied to the main (carrier) board. The green power LED on the FSF-AD8200A should come on. The mezzanine card is now ready for SPI configuration, followed by analog inputs on any or all of the SSMC input connectors. In no case should the input level applied to any SSMC analog input channel exceed 4 volts peak-to-peak.

8.2 Software Installation

The FSF-AD8200A software distribution includes an empty Vivado® Project called “FSF_AD8200A.zip”. Once unzipped, it creates the FSF_AD8200A directory holding an empty Vivado project. This project contains no source files, but it does contain a “customer_release” sub-directory. The necessary bitfiles and debug probe files, and other documentation to exercise the FSF-AD8200A card on a Xilinx VC707 Evaluation Card are all contained in this “customer_release” directory.

Please look in “customer_release/docs” directory and observe 3 important documents there. First there is the “Getting Started Guide” which is a step-by-step guide to installing the necessary software, configuring it, downloading a bitfile, and obtaining waveform traces. Another important document is the “Command Interface Guide” which explains all commands available through the command interface running on the VC707 Evaluation Card. Finally a copy of this document is also stored there.

Inside the empty project directory, find the customer_release directory, which contains the documents, and datafiles sub-directories. The datafiles directory holds the FPGA “download.bit” file which is downloaded into the FPGA on the VC707 carrier board. It configures the hardware of the FPGA to receive data from the FSF-AD8200A card using a Xilinx JESD204B core. Note the FPGA design includes an on-chip processor - the software for this processor is bundled into the “download.bit” file, and the probes file (“debug_netx.ltx”) which allows data to be displayed using the ILA. The “download.bit” file contains both the configuration for the FPGA and the software which runs on a soft processor in the FPGA to provide a simple command interface. This command line interface allows access to the key chips on the FSF-AD8200A and the supporting blocks in the FPGA.

The command line interface is controlled using any simple terminal program running on the host computer connected to the carrier board. For instance if the host computer is a PC, then Tera Term PRO can be used running in 8-N-1 mode, at 115200 baud, with Xon/Xoff flow control enabled.

⁶ and screwing the heat sink to the carrier card if available

When the command interface first runs, it prints some diagnostic information including the current FPGA hardware version and the Command Interface Software version, and then it presents the “Command>” prompt. The “help” command can be entered to get more information. Full information on the commands supported by this interface is available in the “FSF-AD8200A Finale Command Interface Guide.” This can be found in the “docs” sub-directory of the customer_release directory.

Typically the first command entered is “init” which sends a fixed sequence of commands to the FSF-AD8200A intended to configure the LMK04828B clock generator, the ADC1443D converters and the JESD204B core in the FPGA.⁷

Users can also create their own command sequences as text files on the host PC, and send them to the command interface using the “send file” feature of the Tera Term PRO program. The command interface running on the VC707 Evaluation Card will also accept typed commands one-at-a-time. It supports writes and reads from each device. It can also check reads against expected values and count any mismatches.

An example command line script file (“init.scr”) is supplied in the datafiles sub-directory. It does the same job as the built-in “init” command.

Once the LMK04828B clock generator, ADC1443D ADCs and JESD204B core in the FPGA are configured (using the init command, or a script), the system is ready to capture data simultaneously from all 8 channels. The captured data is stored in FPGA block RAM using Xilinx ILA cores (ILA stands for Integrated Logic Analyzer, formerly called Chipscope®). Using the Xilinx Vivado tool, the data in the ILA cores can be extracted and displayed graphically or saved to the host PC for subsequent numerical analysis.

A probe file (“debug_nets.ltx”) is provided in the “datafiles” sub-directory of the customer_release directory. Vivado requires the probes file when it tries to access the ILA cores to allow it to make sense of the data stored in the RAMs on the FPGA.

For a step-by-step description of the bitfile download procedure, including many screen captures, please see the document “FSF-AD8200A Getting Started Guide”.

⁷ See Appendix A for the ‘init’ script contents

8.3 Performing an 8-Channel Capture

The FPGA “download.bit” configuration file supplied with the FSF-AD8200A allows the VC707 Evaluation Card to use 8 large capture buffers accessible through ILA (formerly ChipScope®) to store data incoming from the FSF-AD8200A daughter card. Each buffer can hold 65,536 samples.

Simultaneous capture on all 8 channels begins when the Vivado ILA manual trigger button (labeled “>>”) is left-clicked and continues until the capture buffers are full. Once a capture is complete, the raw data can be exported for external analysis by entering the following command on the TCL command line at the bottom of the Vivado window:

```
write_hw_ila_data filename.zip [upload_ila hw_ila_1]
```

The command window will indicate where it has placed the captured data file once the file has been completely written.

9. ORDERING INFORMATION

| Part Number | Description |
|----------------|--|
| FSF-AD8200A-DS | FSF-AD8200A demonstration unit (early release) |
| FSF-AD8200A | FSF-AD8200A production unit |

10. WARRANTY

The FSF-AD8200A comes with a 6 month warranty. The warranty is governed by Fidus Systems' "Terms of Sale".

11. APPENDIX A: 'INIT' SCRIPT CONTENTS

NOTES:

ws = write

```
*****
// *****
```

```
void init_finale_FMC_card ()
{
```

```
    CommandType cmd ;
```

```
    decodeCommand("ws LMK04828 0000 90 ",&cmd);executeCommand(&cmd); // set RESET and SPI_3WIRE_DIS
    decodeCommand("ws LMK04828 0000 00 ",&cmd);executeCommand(&cmd); // clear RESET and SPI_3WIRE_DIS
    decodeCommand("ws LMK04828 0002 00 ",&cmd);executeCommand(&cmd); // clear POWERDOWN
    decodeCommand("ws LMK04828 0100 0E ",&cmd);executeCommand(&cmd); // set DCLKout0_DIV to 0x0e
    decodeCommand("ws LMK04828 0101 55 ",&cmd);executeCommand(&cmd); // set DCLKout0_DDLY_CNTH to 0x5, set DCLKout0_DDLY_CNTL to
0x5
    decodeCommand("ws LMK04828 0103 00 ",&cmd);executeCommand(&cmd); // clear DCLKout0_DDLY_ADLY to 0x0, clear DCLKout0_ADLY_MUX,
clear DCLKout0_MUX.
    decodeCommand("ws LMK04828 0104 22 ",&cmd);executeCommand(&cmd); // set SDCLKout1_MUX, set SDCLKout1_DDLY to 0x1;
    decodeCommand("ws LMK04828 0105 00 ",&cmd);executeCommand(&cmd); // clear SDCLKout1_ADLY_EN, clear SDCLKout1_ADLY to 0x0
    decodeCommand("ws LMK04828 0106 F0 ",&cmd);executeCommand(&cmd); // set DCLKout0_DDLY_PD, set DCLKout0_HSG_PD, set
DCLKout0_ADLYg_PD, set DCLKout0_ADLY_PD
    decodeCommand("ws LMK04828 0107 15 ",&cmd);executeCommand(&cmd); // set CLKout1_FMT to 0x1, set CLKout0_FMT to 0x5
    decodeCommand("ws LMK04828 0108 0E ",&cmd);executeCommand(&cmd); // set DCLKout2_DIV to 0xe
    decodeCommand("ws LMK04828 0109 55 ",&cmd);executeCommand(&cmd); // set DCLKout2_DDLY_CNTH to 0x5, set DCLKout2_DDLY_CNTL to
0x5
    decodeCommand("ws LMK04828 010B 00 ",&cmd);executeCommand(&cmd); // clear DCLKout2_DDLY_ADLY to 0x0, clear DCLKout2_ADLY_MUX,
clear DCLKout2_MUX.
    decodeCommand("ws LMK04828 010C 22 ",&cmd);executeCommand(&cmd); // set SDCLKout3_MUX, set SDCLKout3_DDLY to 0x1;
    decodeCommand("ws LMK04828 010D 00 ",&cmd);executeCommand(&cmd); // clear SDCLKout3_ADLY_EN, clear SDCLKout3_ADLY to 0x0
    decodeCommand("ws LMK04828 010E F9 ",&cmd);executeCommand(&cmd); // set DCLKout2_DDLY_PD, set DCLKout2_HSG_PD, set
DCLKout2_ADLYg_PD, set DCLKout2_ADLY_PD, set CLKout2_3_PD, set SDCLKout3_PD
    decodeCommand("ws LMK04828 010F 00 ",&cmd);executeCommand(&cmd); // clear CLKout3_FMT to 0x0, clear CLKout2_FMT to 0x0
    decodeCommand("ws LMK04828 0110 1C ",&cmd);executeCommand(&cmd); // set CLKout4_DIV to 0x1c
    decodeCommand("ws LMK04828 0111 55 ",&cmd);executeCommand(&cmd); // set DCLKout4_DDLY_CNTH to 0x5, set DCLKout4_DDLY_CNTL to
0x5
```



```

decodeCommand("ws LMK04828 0113 00 ",&cmd);executeCommand(&cmd); // clear DCLKout4_ADLY to 0x0, clear DCLKout4_MUX to 0x0
decodeCommand("ws LMK04828 0114 02 ",&cmd);executeCommand(&cmd); // set SDCLKout5_DDLY to 0x1
decodeCommand("ws LMK04828 0115 00 ",&cmd);executeCommand(&cmd); // clear SDCLKout5_ADLY_EN, clear SDCLKout5_ADLY to 0x0
decodeCommand("ws LMK04828 0116 F0 ",&cmd);executeCommand(&cmd); // set DCLKout4_DDLY_PD, set DCLKout4_HSG_PD, set
DCLKout4_ADLYg_PD, set DCLKout4_ADLY_PD,
decodeCommand("ws LMK04828 0117 11 ",&cmd);executeCommand(&cmd); // set CLKout5_FMT to 0x1, set CLKout4_FMT to 0x1
decodeCommand("ws LMK04828 0118 1C ",&cmd);executeCommand(&cmd); // DCLKout6_DIV to 0x1c
decodeCommand("ws LMK04828 0119 55 ",&cmd);executeCommand(&cmd); // set DCLKout6_DDLY_CNTH to 0x5, set DCLKout6_DDLY_CNTL to
0x5
decodeCommand("ws LMK04828 011B 00 ",&cmd);executeCommand(&cmd); // clear DCLKout6_ADLY to 0x0, clear DCLKout6_MUX, clear
DCLKout6_MUX to 0x0
decodeCommand("ws LMK04828 011C 22 ",&cmd);executeCommand(&cmd); // set SDCLKout7_MUX, set SDCLKout7_DDLY to 0x1
decodeCommand("ws LMK04828 011D 00 ",&cmd);executeCommand(&cmd); // clear SDCLKout7_ADLY_EN, clear SDCLKout7_ADLY to 0x0
decodeCommand("ws LMK04828 011E F0 ",&cmd);executeCommand(&cmd); // set DCLKout6_DDLY_PD, set DCLKout6_HSG_PD, set
DCLKout6_ADLYg_PD, set DCLKout6_ADLY_PD
decodeCommand("ws LMK04828 011F 11 ",&cmd);executeCommand(&cmd); // set CLKout7_FMT to 0x1, set CLKout6_FMT to 0x1
decodeCommand("ws LMK04828 0120 0E ",&cmd);executeCommand(&cmd); // set DCLKout8_DIV to 0xe
decodeCommand("ws LMK04828 0121 55 ",&cmd);executeCommand(&cmd); // set DCLKout8_DDLY_CNTH to 0x5, set DCLKout8_DDLY_CNTL to
0x5
decodeCommand("ws LMK04828 0123 00 ",&cmd);executeCommand(&cmd); // clear DCLKout8_ADLY to 0x0, clear DCLKout8_MUX, clear
DCLKout8_MUX to 0x0
decodeCommand("ws LMK04828 0124 22 ",&cmd);executeCommand(&cmd); // set SDCLKout9_MUX , set SDCLKout9_DDLY to 0x1
decodeCommand("ws LMK04828 0125 00 ",&cmd);executeCommand(&cmd); // clear SDCLKout9_ADLY_EN, clear SDCLKout9_ADLY to 0x0
decodeCommand("ws LMK04828 0126 F0 ",&cmd);executeCommand(&cmd); // set DCLKout8_DDLY_PD, set DCLKout8_HSG_PD, set
DCLKout8_ADLYg_PD, set DCLKout8_ADLY_PD
decodeCommand("ws LMK04828 0127 15 ",&cmd);executeCommand(&cmd); // set CLKout9_FMT to 0x1, set CLKout8_FMT to 0x5
decodeCommand("ws LMK04828 0128 0E ",&cmd);executeCommand(&cmd); // set DCLKout10_DIV to 0xe
decodeCommand("ws LMK04828 0129 55 ",&cmd);executeCommand(&cmd); // set DCLKout10_DDLY_CNTH to 0x5, set DCLKout10_DDLY_CNTL
to 0x5
decodeCommand("ws LMK04828 012B 00 ",&cmd);executeCommand(&cmd); // clear DCLKout10_ADLY to 0x0, clear DCLKout10_ADLY_MUX, clear
DCLKout10_MUX to 0x0
decodeCommand("ws LMK04828 012C 22 ",&cmd);executeCommand(&cmd); // set SDCLKout11_MUX, set SDCLKout11_DDLY to 0x1
decodeCommand("ws LMK04828 012D 00 ",&cmd);executeCommand(&cmd); // clear SDCLKout11_ADLY_EN, clear SDCLKout11_ADLY to 0x0
decodeCommand("ws LMK04828 012E F0 ",&cmd);executeCommand(&cmd); // set DCLKout10_DDLY_PD, set DCLKout10_HSG_PD, set
DCLKout10_ADLYg_PD, set DCLKout10_ADLY_PD
decodeCommand("ws LMK04828 012F 15 ",&cmd);executeCommand(&cmd); // set CLKout11_FMT to 0x1, set CLKout10_FMT to 0x5
decodeCommand("ws LMK04828 0130 0E ",&cmd);executeCommand(&cmd); // set DCLKout12_DIV to 0x0e
decodeCommand("ws LMK04828 0131 55 ",&cmd);executeCommand(&cmd); // set DCLKout12_DDLY_CNTH to 0x5, set DCLKout12_DDLY_CNTL
to 0x5
decodeCommand("ws LMK04828 0133 00 ",&cmd);executeCommand(&cmd); // clear DCLKout12_ADLY to 0x0, clear DCLKout12_ADLY_MUX, clear
DCLKout12_MUX to 0x0

```

```

decodeCommand("ws LMK04828 0134 22 ",&cmd);executeCommand(&cmd); // set SDCLKout13_MUX, set SDCLKout13_DDLY to 0x1
decodeCommand("ws LMK04828 0135 00 ",&cmd);executeCommand(&cmd); // clear SDCLKout13_ADLY_EN, clear SDCLKout13_ADLY to 0x0
decodeCommand("ws LMK04828 0136 F0 ",&cmd);executeCommand(&cmd); // set DCLKout12_DDLY_PD, set DCLKout12_HSG_PD, set
DCLKout12_ADLYg_PD, set DCLKout12_ADLY_PD
decodeCommand("ws LMK04828 0137 15 ",&cmd);executeCommand(&cmd); // set CLKout13_FMT to 0x1, set CLKout12_FMT to 0x5
decodeCommand("ws LMK04828 0138 00 ",&cmd);executeCommand(&cmd); // clear VCO_MUX to 0x0, clear OSCout_MUX, clear OSCout_FMT to
0x0
decodeCommand("ws LMK04828 0139 03 ",&cmd);executeCommand(&cmd); // set SYSREF_MUX to 0x3
decodeCommand("ws LMK04828 013A 07 ",&cmd);executeCommand(&cmd); // set SYSREF_DIV[12:8] to 0x07
decodeCommand("ws LMK04828 013B E0 ",&cmd);executeCommand(&cmd); // set SYSREF_DIV[7:0] to 0xe0
decodeCommand("ws LMK04828 013C 00 ",&cmd);executeCommand(&cmd); // set SYSREF_DDLY[12:8] to 0x00
decodeCommand("ws LMK04828 013D 08 ",&cmd);executeCommand(&cmd); // set SYSREF_DDLY[7:0] to 0x08
decodeCommand("ws LMK04828 013E 03 ",&cmd);executeCommand(&cmd); // set SYSREF_PULSE_CNT to 0x3
decodeCommand("ws LMK04828 013F 00 ",&cmd);executeCommand(&cmd); // clear PLL2_NCLK_MUX, clear PLL1_NCLK_MUX,clear FB_MUX to
0x0, clear FB_MUX_EN
decodeCommand("ws LMK04828 0140 01 ",&cmd);executeCommand(&cmd); // set SYSREF_DDLY_PD
decodeCommand("ws LMK04828 0141 00 ",&cmd);executeCommand(&cmd);
decodeCommand("ws LMK04828 0142 00 ",&cmd);executeCommand(&cmd);
decodeCommand("ws LMK04828 0143 10 ",&cmd);executeCommand(&cmd); // set SYNC_EN
decodeCommand("ws LMK04828 0144 00 ",&cmd);executeCommand(&cmd);
decodeCommand("ws LMK04828 0145 7F ",&cmd);executeCommand(&cmd); // this is a fixed value... not sure why we write it.
decodeCommand("ws LMK04828 0146 0F ",&cmd);executeCommand(&cmd); // set CLKin0_EN, set CLKin2_TYPE,set CLKin1_TYPE,set
CLKin0_TYPE
decodeCommand("ws LMK04828 0147 06 ",&cmd);executeCommand(&cmd); // set CLKin1_OUT_MUX to 0x1, set CLKin0_OUT_MUX to 0x2
decodeCommand("ws LMK04828 0148 02 ",&cmd);executeCommand(&cmd); // clear CLKin_SELO_MUX to 0x0, set CLKin_SELO_TYPE to 0x2
decodeCommand("ws LMK04828 0149 02 ",&cmd);executeCommand(&cmd); // clear CLKin_SEL1_MUX to 0x0, set CLKin_SEL1_TYPE to 0x2
decodeCommand("ws LMK04828 014A 02 ",&cmd);executeCommand(&cmd); // clear RESET_MUX to 0x0, set RESET_TYPE to 0x2
decodeCommand("ws LMK04828 014B 16 ",&cmd);executeCommand(&cmd); // set TRACK_EN, set MAN_DAC_EN, set MAN_DAC[9:8] to 0x2
decodeCommand("ws LMK04828 014C 00 ",&cmd);executeCommand(&cmd); // set MAN_DAC[7:0] to 0x00
decodeCommand("ws LMK04828 014D 00 ",&cmd);executeCommand(&cmd); // set DAC_TRIP_LOW to 0x00
decodeCommand("ws LMK04828 014E C0 ",&cmd);executeCommand(&cmd); // set DAC_CLK_MULT to 0x3, set DAC_TRIP_HIGH to 0x00
decodeCommand("ws LMK04828 014F 7F ",&cmd);executeCommand(&cmd); // set DAC_CLK_CNTR to 0x7f
decodeCommand("ws LMK04828 0150 1B ",&cmd);executeCommand(&cmd); // set HOLDOVER_PLL1_DET, set HOLDOVER_LOS_DET, set
HOLDOVER_HITLESS_SWITCH, set HOLDOVER_EN
decodeCommand("ws LMK04828 0151 02 ",&cmd);executeCommand(&cmd); // set HOLDOVER_DLD_CNT[13:8] to 0x02
decodeCommand("ws LMK04828 0152 00 ",&cmd);executeCommand(&cmd); // set HOLDOVER_DLD_CNT[7:0] to 0x00
decodeCommand("ws LMK04828 0153 00 ",&cmd);executeCommand(&cmd); // set CLKin0_R[13:8] to 0x00
decodeCommand("ws LMK04828 0154 02 ",&cmd);executeCommand(&cmd); // set CLKin0_R[7:0] to 0x02
decodeCommand("ws LMK04828 0155 00 ",&cmd);executeCommand(&cmd); // set CLKin1_R[13:8] to 0x00
decodeCommand("ws LMK04828 0156 02 ",&cmd);executeCommand(&cmd); // set CLKin1_R[7:0] to 0x02
decodeCommand("ws LMK04828 0157 00 ",&cmd);executeCommand(&cmd); // set CLKin2_R[13:8] to 0x00

```

```

decodeCommand("ws LMK04828 0158 96 ",&cmd);executeCommand(&cmd); // set CLKin2_R[7:0] to 0x96
decodeCommand("ws LMK04828 0159 00 ",&cmd);executeCommand(&cmd); // set PLL1_N[13:8] to 0x00
decodeCommand("ws LMK04828 015A 19 ",&cmd);executeCommand(&cmd); // set PLL1_N[7:0] to 0x19
decodeCommand("ws LMK04828 015B DF ",&cmd);executeCommand(&cmd); // set PLL1_WND_SIZE to 0x3, set PLL1_CP_POL, set PLL1_CP_GAIN
to 0xf
decodeCommand("ws LMK04828 015C 20 ",&cmd);executeCommand(&cmd); // set PLL1_DLD_CNT[13:8] to 0x20
decodeCommand("ws LMK04828 015D 00 ",&cmd);executeCommand(&cmd); // set PLL1_DLD_CNT[7:0] to 0x00
decodeCommand("ws LMK04828 015E 00 ",&cmd);executeCommand(&cmd); // set PLL1_R_DLY to 0x0, set 0x15E 0 0 PLL1_R_DLY to 0x0
decodeCommand("ws LMK04828 015F 0B ",&cmd);executeCommand(&cmd); // set PLL1_LD_MUX to 0x01, set PLL1_LD_TYPE to 0x3
decodeCommand("ws LMK04828 0160 00 ",&cmd);executeCommand(&cmd); // set PLL2_R[11:8] to 0x0
decodeCommand("ws LMK04828 0161 01 ",&cmd);executeCommand(&cmd); // set PLL2_R[7:0] to 0x00
decodeCommand("ws LMK04828 0162 44 ",&cmd);executeCommand(&cmd); // set PLL2_P to 0x2, set OSCin_FREQ 0x1, clear PLL2_XTAL_EN,
clear PLL2_REF_2X_EN
decodeCommand("ws LMK04828 0163 00 ",&cmd);executeCommand(&cmd); // clear PLL2_N_CAL[17:16] to 0x0
decodeCommand("ws LMK04828 0164 00 ",&cmd);executeCommand(&cmd); // clear PLL2_N_CAL[15:8] to 0x00
decodeCommand("ws LMK04828 0165 0C ",&cmd);executeCommand(&cmd); // set PLL2_N_CAL[7:0] to 0x0c
decodeCommand("ws LMK04828 017C 15 ",&cmd);executeCommand(&cmd); // set OPT_REG_1 to 0x15
decodeCommand("ws LMK04828 017D 33 ",&cmd);executeCommand(&cmd); // set OPT_REG_2 to 0x33
decodeCommand("ws LMK04828 0166 00 ",&cmd);executeCommand(&cmd); // clear PLL2_FCAL_DIS, clear PLL2_N[17:16] to 0x0
decodeCommand("ws LMK04828 0167 00 ",&cmd);executeCommand(&cmd); // clear PLL2_N[15:8]
decodeCommand("ws LMK04828 0168 0A ",&cmd);executeCommand(&cmd); // set PLL2_N[7:0] to 0x0a
decodeCommand("ws LMK04828 0169 59 ",&cmd);executeCommand(&cmd); // set PLL2_WND_SIZE to 0x2, set PLL2_CP_GAIN to 0x1, clear
_CP_POL, clear PLL2_CP_TRI
decodeCommand("ws LMK04828 016A 20 ",&cmd);executeCommand(&cmd); // set PLL2_DLD_CNT[15:8] to 0x20
decodeCommand("ws LMK04828 016B 00 ",&cmd);executeCommand(&cmd); // clear PLL2_DLD_CNT[7:0] to 0x00
decodeCommand("ws LMK04828 016C 00 ",&cmd);executeCommand(&cmd); // clear PLL2_LF_R4 to 0x0, clear PLL2_LF_R3 to 0x0
decodeCommand("ws LMK04828 016D 00 ",&cmd);executeCommand(&cmd); // clear PLL2_LF_C4 to 0x0, clear PLL2_LF_C3 to 0x0
decodeCommand("ws LMK04828 016E 13 ",&cmd);executeCommand(&cmd); // set PLL2_LD_MUX to 0x1, set PLL2_LD_TYPE to 0x3
decodeCommand("ws LMK04828 0173 00 ",&cmd);executeCommand(&cmd); // clear PLL2_PRE_PD, clear PLL2_PD
decodeCommand("ws LMK04828 1FFD 00 ",&cmd);executeCommand(&cmd); // clear SPI_LOCK[23:16] to 0x00
decodeCommand("ws LMK04828 1FFE 00 ",&cmd);executeCommand(&cmd); // clear SPI_LOCK[15:8] to 0x00
decodeCommand("ws LMK04828 1FFF 53 ",&cmd);executeCommand(&cmd); // set SPI_LOCK[7:0] to 0x53
decodeCommand("ws LMK04828 0144 FF ",&cmd);executeCommand(&cmd); // set SYNC_DISSYSREF, set SYNC_DIS12, set SYNC_DIS10, set
SYNC_DIS8, set SYNC_DIS6, set SYNC_DIS4, set SYNC_DIS2, set SYNC_DIS0
decodeCommand("ws AD1443D_1 0803 00 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_1 0802 08 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_1 0100 d1 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_1 0200 01 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_1 00ff 80 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_1 0102 07 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_1 0103 66 ",&cmd);executeCommand(&cmd);

```

```

decodeCommand("ws AD1443D_1 0012 10 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_1 0108 a3 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_1 010a c0 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_1 0154 01 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_1 0155 03 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_1 0156 d8 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_1 0160 ff ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_1 0161 17 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_1 0170 10 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_1 0171 10 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_1 0400 30 ",&cmd);executeCommand(&cmd);
delay_ms (400);
decodeCommand("ws AD1443D_1 0004 08 ",&cmd);executeCommand(&cmd);
delay_ms (400);
decodeCommand("ws AD1443D_1 0004 10 ",&cmd);executeCommand(&cmd);
delay_ms (400);
decodeCommand("ws AD1443D_1 0004 20 ",&cmd);executeCommand(&cmd);
delay_ms (400);
decodeCommand("ws AD1443D_1 0043 C7 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_1 081C 4A ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_1 0822 01 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_1 0810 C0 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_1 0811 40 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_1 0812 0A ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_1 081E 08 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_1 086B 02 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_1 086C 02 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_1 0872 04 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_2 0803 00 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_2 0802 08 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_2 0100 d1 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_2 0200 01 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_2 00ff 80 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_2 0102 07 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_2 0103 66 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_2 0012 10 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_2 0108 a3 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_2 010a c0 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_2 0154 01 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_2 0155 03 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_2 0156 d8 ",&cmd);executeCommand(&cmd);

```

```

decodeCommand("ws AD1443D_2 0160 ff ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_2 0161 17 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_2 0170 10 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_2 0171 10 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_2 0400 30 ",&cmd);executeCommand(&cmd);
delay_ms (400);
decodeCommand("ws AD1443D_2 0004 08 ",&cmd);executeCommand(&cmd);
delay_ms (400);
decodeCommand("ws AD1443D_2 0004 10 ",&cmd);executeCommand(&cmd);
delay_ms (400);
decodeCommand("ws AD1443D_2 0004 20 ",&cmd);executeCommand(&cmd);
delay_ms (400);
decodeCommand("ws AD1443D_2 0043 C7 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_2 081C 4A ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_2 0822 01 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_2 0810 C0 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_2 0811 40 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_2 0812 0A ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_2 081E 08 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_2 086B 02 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_2 086C 02 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_2 0872 04 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_3 0803 00 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_3 0802 08 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_3 0100 d1 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_3 0200 01 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_3 00ff 80 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_3 0102 07 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_3 0103 66 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_3 0012 10 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_3 0108 a3 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_3 010a c0 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_3 0154 01 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_3 0155 03 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_3 0156 d8 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_3 0160 ff ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_3 0161 17 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_3 0170 10 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_3 0171 10 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_3 0400 30 ",&cmd);executeCommand(&cmd);
delay_ms (400);

```

```

decodeCommand("ws AD1443D_3 0004 08 ",&cmd);executeCommand(&cmd);
delay_ms (400);
decodeCommand("ws AD1443D_3 0004 10 ",&cmd);executeCommand(&cmd);
delay_ms (400);
decodeCommand("ws AD1443D_3 0004 20 ",&cmd);executeCommand(&cmd);
delay_ms (400);
decodeCommand("ws AD1443D_3 0043 C7 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_3 081C 4A ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_3 0822 01 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_3 0810 C0 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_3 0811 40 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_3 0812 0A ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_3 081E 08 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_3 086B 02 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_3 086C 02 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_3 0872 04 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_4 0803 00 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_4 0802 08 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_4 0100 d1 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_4 0200 01 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_4 00ff 80 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_4 0102 07 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_4 0103 66 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_4 0012 10 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_4 0108 a3 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_4 010a c0 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_4 0154 01 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_4 0155 03 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_4 0156 d8 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_4 0160 ff ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_4 0161 17 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_4 0170 10 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_4 0171 10 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_4 0400 30 ",&cmd);executeCommand(&cmd);
delay_ms (400);
decodeCommand("ws AD1443D_4 0004 08 ",&cmd);executeCommand(&cmd);
delay_ms (400);
decodeCommand("ws AD1443D_4 0004 10 ",&cmd);executeCommand(&cmd);
delay_ms (400);
decodeCommand("ws AD1443D_4 0004 20 ",&cmd);executeCommand(&cmd);
delay_ms (400);

```

```
decodeCommand("ws AD1443D_4 0043 C7 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_4 081C 4A ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_4 0822 01 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_4 0810 C0 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_4 0811 40 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_4 0812 0A ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_4 081E 08 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_4 086B 02 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_4 086C 02 ",&cmd);executeCommand(&cmd);
decodeCommand("ws AD1443D_4 0872 04 ",&cmd);executeCommand(&cmd);
decodeCommand("ws JESD204B 04 01 ",&cmd);executeCommand(&cmd);
decodeCommand("ws JESD204B 0C 40008 ",&cmd);executeCommand(&cmd);
decodeCommand("ws JESD204B 00 790 ",&cmd);executeCommand(&cmd);
decodeCommand("ws JESD204B 00 712 ",&cmd);executeCommand(&cmd);

xil_printf("\n\rInfo: Done Initialization");

}
```