ADXBAND16EBZ Prototyping Platform User Guide

FEATURES

- Transmit and receive frequency X band: 8GHz to 12GHz
- Direct sampling in L, S, and C bands with capacitor rotation
- Multichannel, wideband system development platform for the AD9084
- Mates with Xilinx VCU118 evaluation board (must be purchased separately)
- 16× RF receive channels (32× digital receive channels)
  - Total 16× 4GSPS to 20GSPS ADC channels
  - 48× digital downconverters (DDCs) and complex NCOs
  - 16× programmable finite impulse response filters (pFIRs)
  - 16× complex finite impulse response filters (CFIRs)
- Complete power management solution for all domains
- Single 12V supply voltage for easy power management
- Board size: 221mm × 158mm (8.9” × 6.2”)

APPLICATIONS

- Phased-array, radar, electronic warfare, and satellite communications (SATCOM)
- Communications infrastructure (multiband and mmWave 5G)
- Electronic test and measurement

EVALUATION KIT CONTENTS

- 12V, 30A+ wall supply and power cable
- Installed fans and heatsinks
- 2× SMA to SMPM cables, used to test with equipment
- 3× SMA to SMPM cables (2× included with ADXBAND16EBZ kit and 1× included with ADXBAND16EBZCAL kit)

EQUIPMENT NEEDED

- 400MHz reference oscillator or waveform generator
- Xilinx VCU118 evaluation board
- Ethernet cable
- 2× microUSB cables (included with VCU118 kit)
- 32× SMPM to SMPM cables (not included)

DOCUMENTS NEEDED

- AD9084 data sheet
- ADF4030 and ADF4382A data sheets
- LTC2977 data sheet

SOFTWARE NEEDED

- MATLAB
- Xilinx Vivado
- Putty
- Python

EVALUATION BOARD PHOTOGRAPH

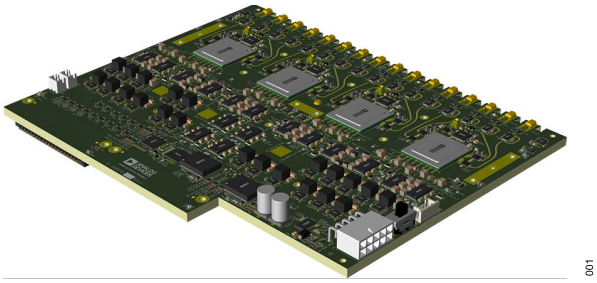


Figure 1. ADXBAND16EBZ Evaluation Board Photograph

GENERAL DESCRIPTION

This user guide serves as the main source of information for system engineers and software developers using the ADXBAND16EBZ. This platform contains four AD9084 software-defined, direct sampling mixed-signal front end (MxFE), including an RF front end, clocking, and power circuitry. Common target applications are phased array radars, electronic warfare, and ground-based SATCOM. The receive and transmit RF front ends have bypass configurations that allow customized frequency ranges of less than 8GHz up to 18GHz.

The ADXBAND16EBZ highlights a complete system solution. This platform is intended as a testbed for demonstrating multichip synchronization as well as the implementation of system-level calibrations and signal processing algorithms. The system is designed to mate with a Xilinx VCU118 evaluation board, which features the Virtex UltraScale+ XCVU9P field-programmable gate array (FPGA), with provided reference software, hardware design language (HDL) code, and MATLAB system-level interfacing.

A 16 transmit and 16 receive calibration board (ADXBAND16EBZ-CAL) can be used to develop system-level calibration algorithms to demonstrate power-up phase determinism. The calibration board demonstrates combined-channel dynamic range, spurious, and phase noise improvements that can also be controlled via a MATLAB add-on when connected to the peripheral module (PMOD) interface of the VCU118.

TABLE OF CONTENTS

TABLE OF CONTENTS

REVISION HISTORY

6/2026—Revision 0: Initial Version

HIGH-LEVEL BLOCK DIAGRAM

Figure 2 shows a high-level block diagram.

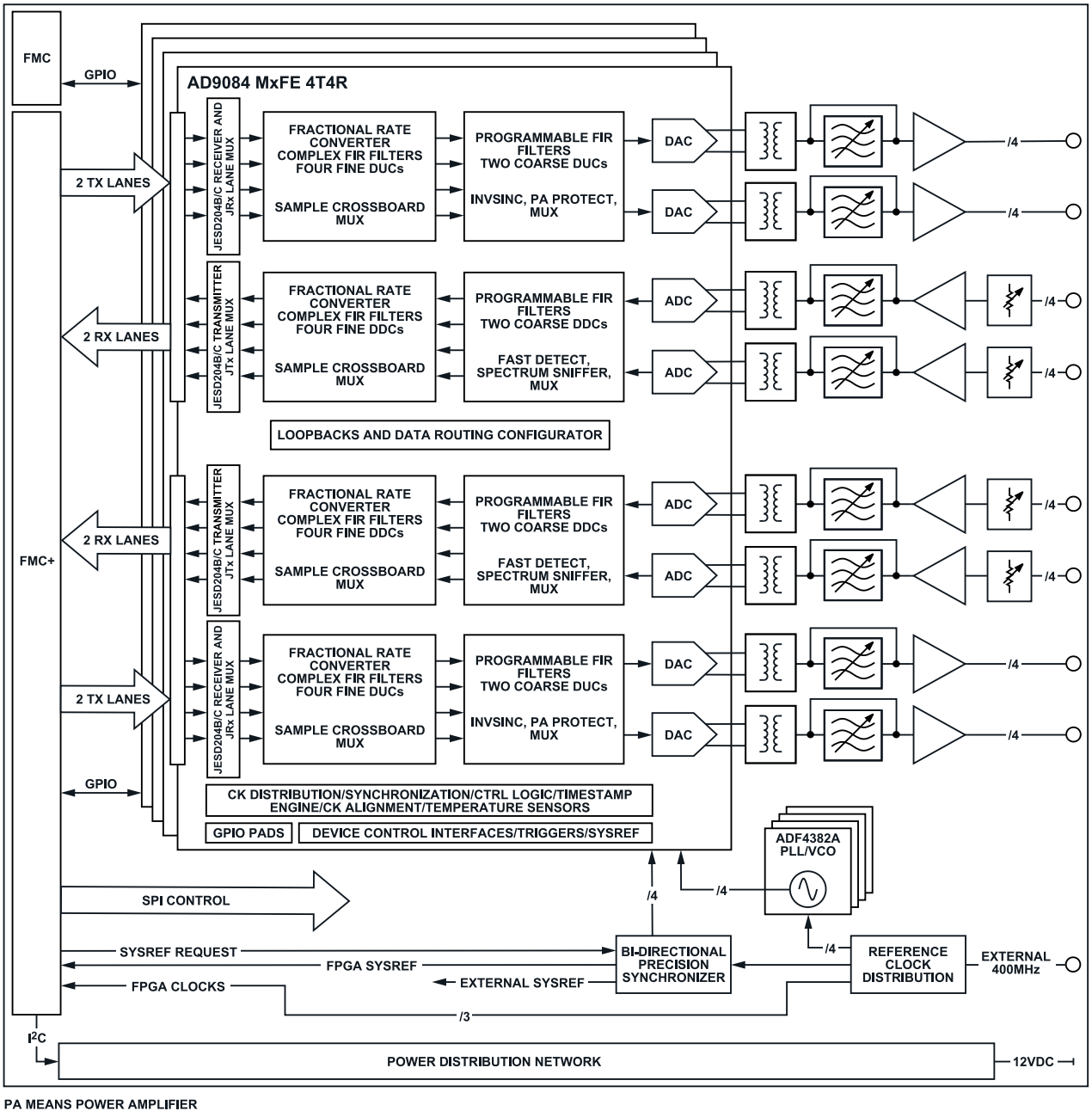


Figure 2. High-Level Block Diagram

RECEIVE SIGNAL CHAIN

Figure 3 shows the receive signal chain.

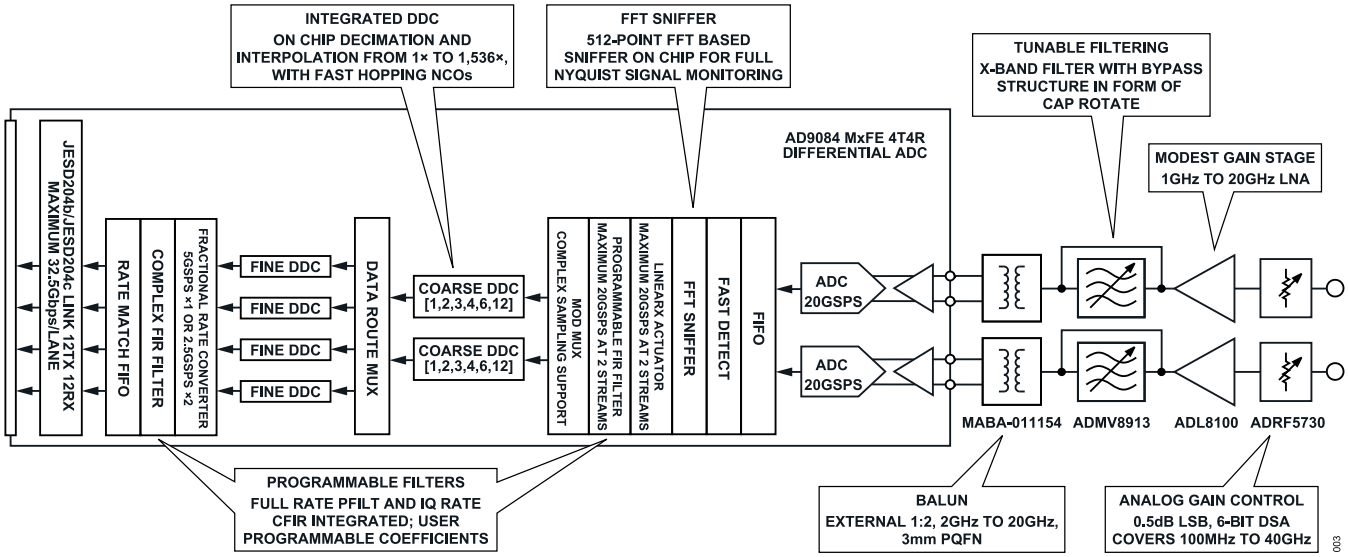


Figure 3. Receive Signal Chain

TRANSMIT SIGNAL CHAIN

Figure 4 shows the transmit signal chain.

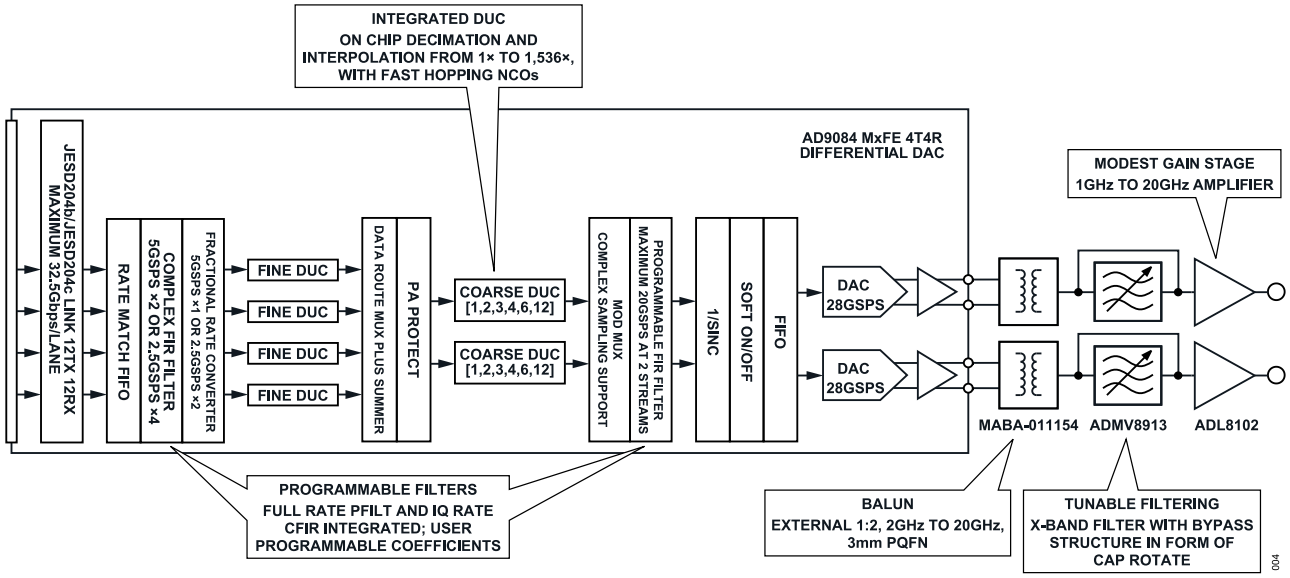


Figure 4. Transmit Signal Chain

SPECIFICATIONS

All measurement were taken using the ADXBAND16EBZ development kit using a 400MHz REF CLOCK input for 12.8GSPS and 25.6GSPS sampling with a 16× decimation rate at 25°C and a DC input voltage of 12V.

Table 1. Specifications

Parameter	Typical	Unit	Test Conditions/Comments
TRANSMIT			
Operating Frequency Range	8 to 12	GHz	Native x-band filtered configuration
RF Channels	16		
Spurious-Free Dynamic Range (SFDR) vs. Frequency			
Single	70 to 78	dB	
Combined	85	dB	
Single-Tone Output Power vs. Frequency (Full Scale)			
8GHz	0	dBm	
10GHz	+2	dBm	
12GHz	−2.5	dBm	
RECEIVE			
Operating Frequency Range	8 to 12	GHz	Native x-band filtered configuration
RF Channels	16		
Input Attenuation	0 to 31.5	dB	ADRF5730
SFDR vs. Frequency			
Single	68 to 74	dB	
Combined	78 to 84	dB	
SAMPLE RATE	25.6/12.8	GSPS	Digital-to-analog converter (DAC)/analog-to-digital converter (ADC) sample rates for standard 2:1 sampling ratio configuration
Maximum DAC	28	GSPS	In 2:1 ratio, ADC sample rate must be ½ of DAC rate or 14GSPS
Maximum ADC	20	GSPS	In 1:1 ratio, DAC sample rate must be equal to ADC rate or 20GSPS
JESD204B AND JESD204C LANE RATE	26.4/13.2	Gbps	800MHz/400MHz instantaneous bandwidth (IBW) use cases
INPUT THIRD-ORDER INTERCEPT POINT (IIP3)			
Typical	>5	dBm	
Worst	>2	dBm	
REF CLOCK (ADF4382A INPUT)	400/500	MHz	Standard configurations for 8 decimation and 16 decimation
BANDWIDTH			
IBW	400/800	MHz	Decimation by 16 or decimation by 8
Filtered Effective	320/640	MHz	80% of IBW due to decimation filter
CONNECTOR SPACING	12.5	mm	Channel to channel
	6.5	mm	Connector to connector on opposite sides of the printed circuit board (PCB)
DC			
Input Voltage	12	V	
Power Consumption	≤317	W	The provided AC and DC converter is rated to 400W
DIMENSIONS	221 × 158	mm	Equivalent to 8.9” × 6.2”

ADXBAND16EBZ REAL USE CASE FREQUENCY PLANS

Two ADXBAND16EBZ builds are available for 400MHz and 800MHz decimated real bandwidths, and both bandwidths operate with a 12.8GSPS rate for the ADCs and 25.6GSPS for the DACs (see Table 2).

Table 2. ADXBAND16EBZ Real Use Case Frequency Plans

Status	Sample Clock (GSPS)	ADC Sample Rate (GSPS)	DAC Sample Rate (GSPS)	JESD204C (Gbps)	Decimation Rate	IBW (MHz)	REF CLK (MHz)	Frequency Band	
								First Nyquist	Second Nyquist
Current	12.8	12.8	25.6	13.2	16	400	400	L, S, and half of C	Half of C and X
Current	12.8	12.8	25.6	26.4	8	800	400	L, S, and half of C	Half of Cand X

LAYOUT AND GENERAL PARTITION

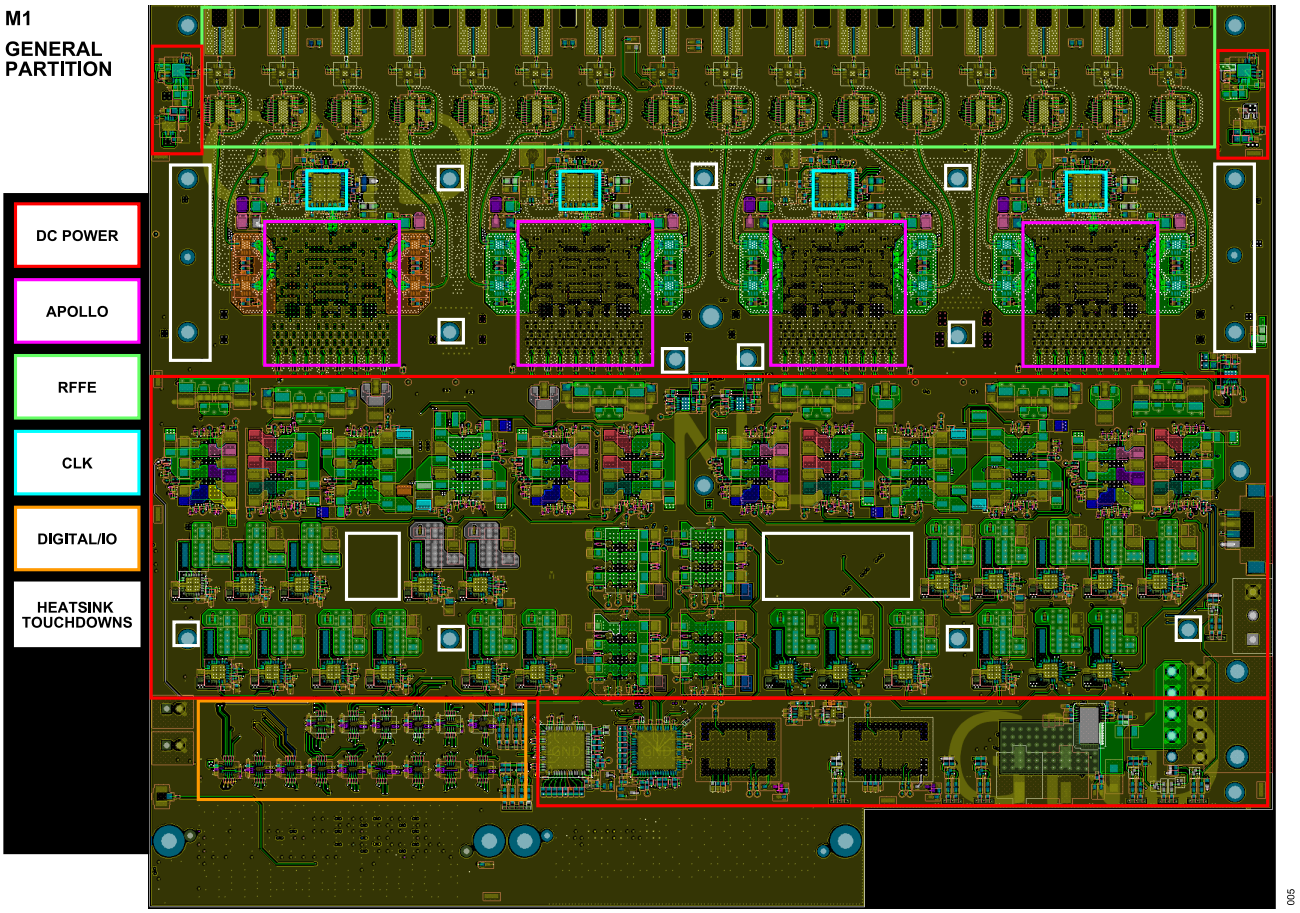


Figure 5. Metal Layer 1 (M1) Layout and General Partition

LAYOUT AND GENERAL PARTITION

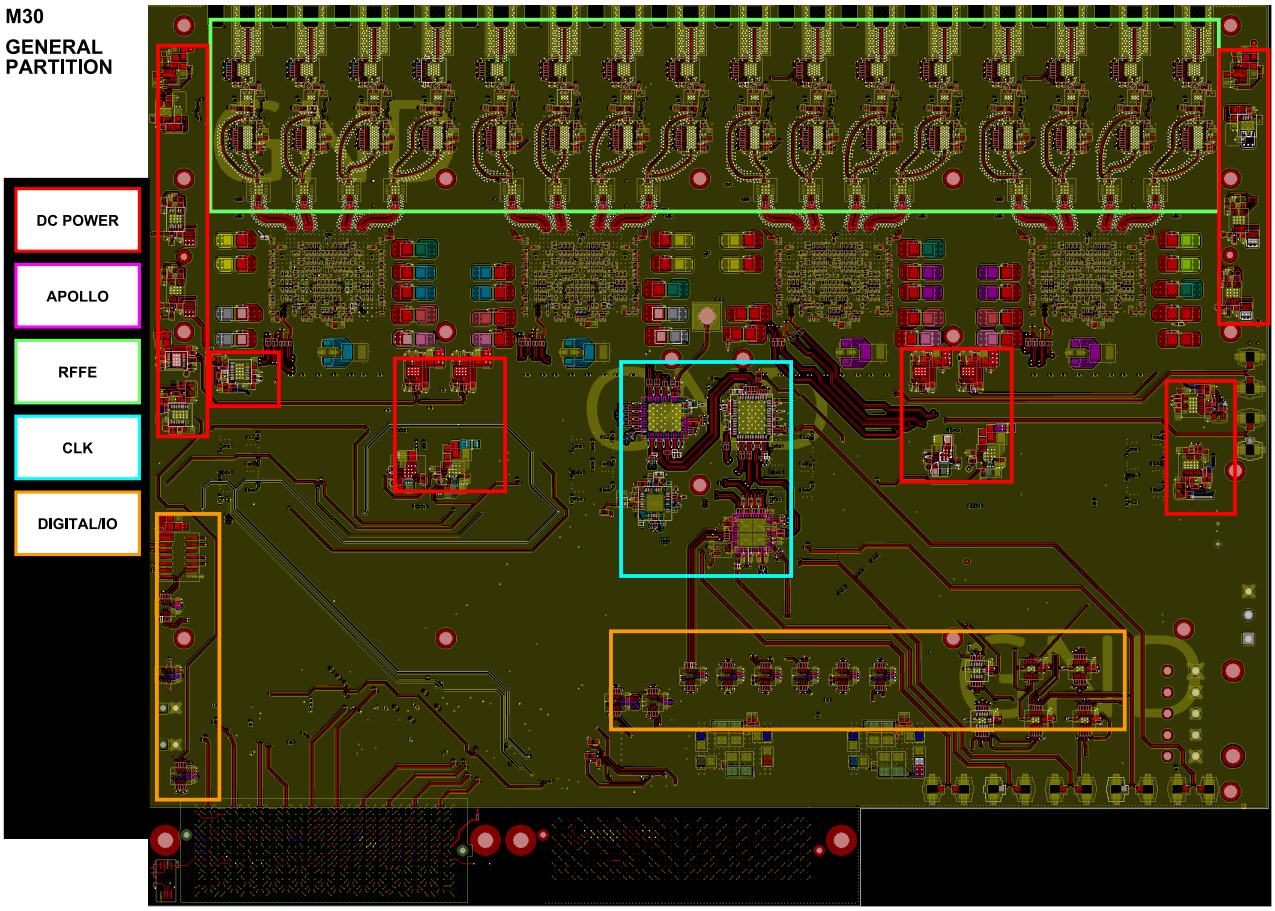


Figure 6. Metal Layer 30 (M30) Layout and General Partition

Table 3. Transmit and Receive Channel Mapping

Tx/ Rx	Channel	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Tx	Label	TX0	TX1	TX2	TX3	TX4	TX5	TX6	TX7	TX8	TX9	TX10	TX11	TX12	TX13	TX14	TX15
	Reference Designator	J10	J2	J18	J26	J11	J3	J19	J27	J12	J4	J20	J28	J13	J5	J21	J29
Rx	Label	RX0	RX1	RX2	RX3	RX4	RX5	RX6	RX7	RX8	RX9	RX10	RX11	RX12	RX13	RX14	RX15
	Reference Designator	J6	J14	J30	J22	J7	J15	J31	J23	J8	J16	J32	J24	J9	J17	J33	J25

Table 4. Pin Function Description

Reference Designator	Type	Description
J1	Input	System Reference Clock.
J2, J3, J4, J5, J10, J11, J12, J13, J18, J19, J20, J21, J26, J27, J28, J29	Output	Direct Sample Transmit Channels.
J6, J7, J8, J9, J14, J15, J16, J17, J22, J23, J24, J25, J30, J31, J32, J33	Input	Direct Sample Receive Channels.
J34, J35, J36, J37	Input	External Sample Clock for Apollo Mixed Signal front End (MxFE).
J40, J41	Input/output	ADF4030 BYSYNC Channel 4. Requires surface-mount technology (SMT) rework to connect. Default connect to FPGA for SYSREF.
J42, J43	Input/output	ADF4030 BYSYNC Channel 5. Requires SMT rework to connect. Default connect to FPGA for SYSREF.
J44, J45	Input/output	ADF4030 BYSYNC Channel 8. Requires SMT rework to connect. Default connect to LTC6952 for reference BSYNC.

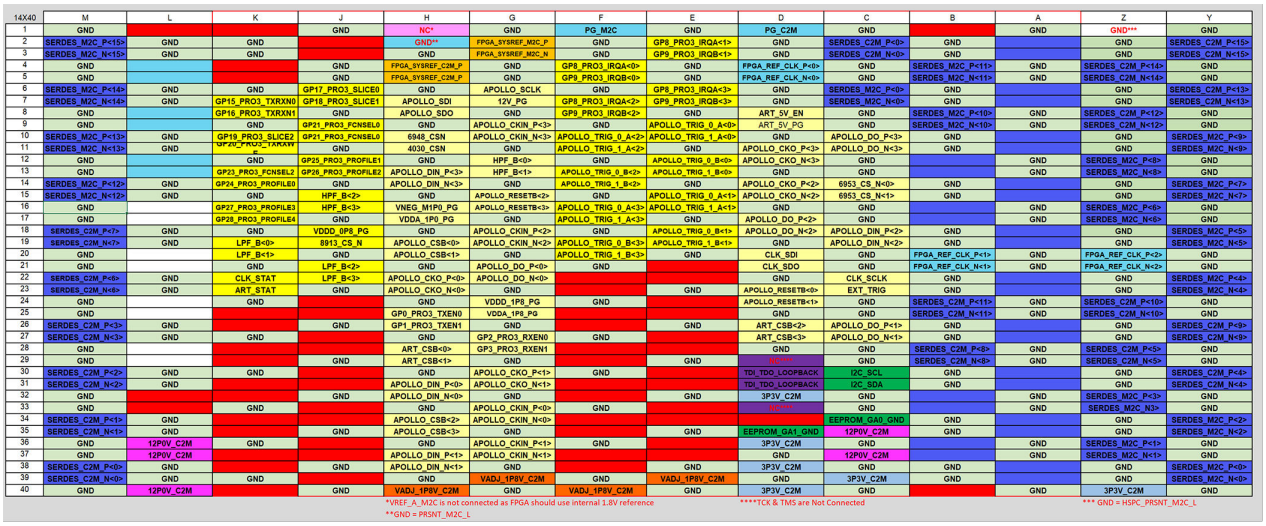
LAYOUT AND GENERAL PARTITION

Table 4. Pin Function Description (Continued)

Reference Designator	Type	Description
J46, J47	Input/output	ADF4030 BYSYNC Channel 9.
J48, J49	Output	LTC6952 Clock Output Test Point.
J50, J51	Output	LTC6953 Clock Output Test Point.
J52	Input/output	FPGA Trigger to Support Memory Configuration Storage (MCS).
P1	Input	12V DC Power Supply Header.
P2	Input/output	FMC+ Mezzanine Header, VITA 57.4 Standard.
P3	Input/output	FMC Mezzanine Header, VITA 57.4 Standard.
P4	Input/output	I^2C Header for LTPowerPlay Electrically Erasable Programmable Read-Only Memory (EEPROM).
P5, P6	Input	12V DC Header for Heatsink Fans.

DIGITAL INTERFACE

The pinout of the FMC+ mezzanine connector and FMC mezzanine connector follow the VITA 57.4 standard to be compatible with the VCU118 evaluation board. For more information on the VITA 57.4 standard, follow this link: https://fmchub.github.io/appendix/VITA57_FMC_HPC_LPC_SIGNALS_AND_PINOUT.html.

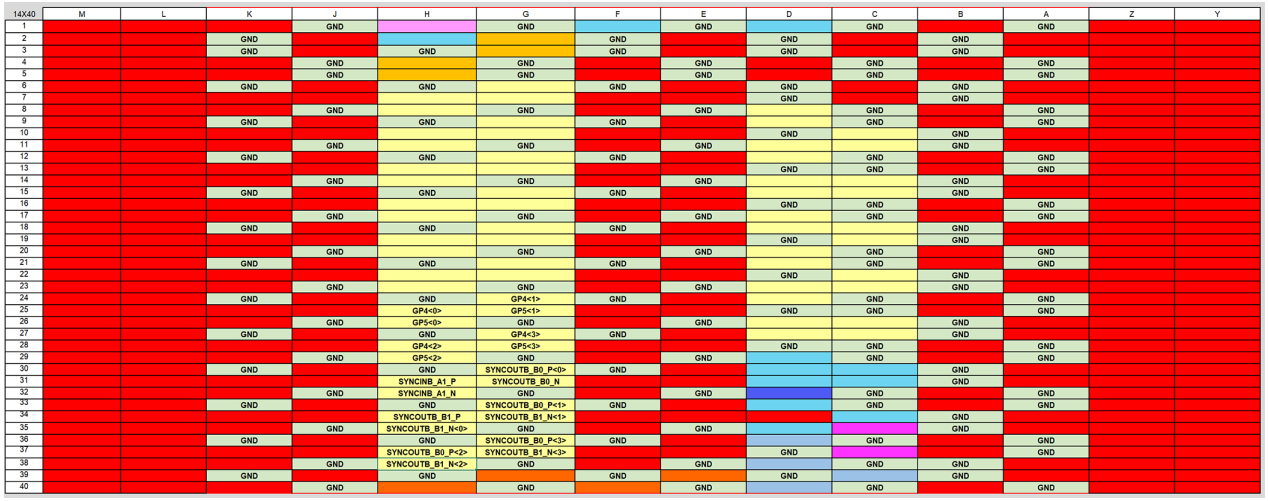


14X40	M	L	K	J	H	G	F	E	D	C	B	A	Z	Y
1	GND			GND	NC*	GND	PG_M2C	GND	PG_C2M	GND		GND	GND***	GND
2	SERDES_M2C_P<15>	GND	GND		GND**	FPGA_SYSREF_M2C_P	GND	GP8_PRO3_IRQA<1>	GND	SERDES_C2M_P<0>	GND		GND	SERDES_C2M_P<15>
3	SERDES_M2C_N<15>	GND	GND		GND	FPGA_SYSREF_M2C_N	GND	GP9_PRO3_IRQB<1>	GND	SERDES_C2M_N<0>	GND		GND	SERDES_C2M_N<15>
4	GND			GND	FPGA_SYSREF_C2M_P	GND	GP8_PRO3_IRQA<0>	GND	FPGA_REF_CLK_P<0>	GND	SERDES_M2C_P<11>	GND	SERDES_C2M_P<14>	GND
5	GND			GND	FPGA_SYSREF_C2M_P	GND	GP9_PRO3_IRQB<0>	GND	FPGA_REF_CLK_N<0>	GND	SERDES_M2C_N<11>	GND	SERDES_C2M_N<14>	GND
6	SERDES_M2C_P<14>	GND	GND	GP17_PRO3_SLICE0	GND	APOLLO_SCLK	GND	GP8_PRO3_IRQA<3>	GND	SERDES_M2C_P<0>	GND		GND	SERDES_C2M_P<13>
7	SERDES_M2C_N<14>	GND	GP15_PRO3_TXRXN0	GP18_PRO3_SLICE1	APOLLO_SDI	12V_PG	GP8_PRO3_IRQA<2>	GP9_PRO3_IRQB<3>	GND	SERDES_M2C_N<0>	GND		GND	SERDES_C2M_N<13>
8	GND		GP16_PRO3_TXRXN1	GND	APOLLO_SDO	GND	GP9_PRO3_IRQB<2>	GND	ART_5V_EN	GND	SERDES_M2C_P<10>	GND	SERDES_C2M_P<12>	GND
9	GND		GND	GP21_PRO3_FCNSEL0	GND	APOLLO_CKIN_P<3>	GND	APOLLO_TRIG_0_A<0>	ART_5V_PG	GND	SERDES_M2C_N<10>	GND	SERDES_C2M_N<12>	GND
10	SERDES_M2C_P<13>	GND	GP19_PRO3_SLICE2	GP21_PRO3_FCNSEL0	6948_CSN	APOLLO_CKIN_N<3>	APOLLO_TRIG_0_A<2>	APOLLO_TRIG_1_A<0>	GND	APOLLO_DO_P<3>	GND		GND	SERDES_M2C_P<9>
11	SERDES_M2C_N<13>	GND	GP20_PRO3_TXRXP0	GND	4030_CSN	GND	APOLLO_TRIG_1_A<2>	GND	APOLLO_CKO_P<3>	APOLLO_DO_N<3>	GND		GND	SERDES_M2C_N<9>
12	GND		GND	GP25_PRO3_PROFILE1	GND	HPF_B<0>	GND	APOLLO_TRIG_0_B<0>	APOLLO_CKO_N<3>	GND		GND	SERDES_M2C_P<8>	GND
13	GND		GP23_PRO3_FCNSEL2	GP26_PRO3_PROFILE2	APOLLO_DIN_P<3>	HPF_B<1>	APOLLO_TRIG_0_B<2>	APOLLO_TRIG_1_B<0>	GND	GND		GND	SERDES_M2C_N<8>	GND
14	SERDES_M2C_P<12>	GND	GP24_PRO3_PROFILE0	GND	APOLLO_DIN_N<3>	GND	APOLLO_TRIG_1_B<2>	GND	APOLLO_CKO_P<2>	6953_CS_N<0>	GND		GND	SERDES_M2C_P<7>
15	SERDES_M2C_N<12>	GND	GND	HPF_B<2>	GND	APOLLO_RESETB<2>	GND	APOLLO_TRIG_0_A<1>	APOLLO_CKO_N<2>	6953_CS_N<1>	GND		GND	SERDES_M2C_N<7>
16	GND		GP27_PRO3_PROFILE3	HPF_B<3>	VNEG_M1P0_PG	APOLLO_RESETB<3>	APOLLO_TRIG_0_A<3>	APOLLO_TRIG_1_A<1>	GND	GND		GND	SERDES_M2C_P<6>	GND
17	GND		GP28_PRO3_PROFILE4	GND	VDDA_1P0_PG	GND	APOLLO_TRIG_1_A<3>	GND	APOLLO_DO_P<2>	GND		GND	SERDES_M2C_N<6>	GND
18	SERDES_C2M_P<7>	GND	GND	VDDD_0P8_PG	GND	APOLLO_CKIN_P<2>	GND	APOLLO_TRIG_0_B<1>	APOLLO_DO_N<2>	APOLLO_DIN_P<2>	GND		GND	SERDES_M2C_P<5>
19	SERDES_C2M_N<7>	GND	LPF_B<0>	8913_CS_N	APOLLO_CSB<0>	APOLLO_CKIN_N<2>	APOLLO_TRIG_0_B<3>	APOLLO_TRIG_1_B<1>	GND	APOLLO_DIN_N<2>	GND		GND	SERDES_M2C_N<5>
20	GND		LPF_B<1>	GND	APOLLO_CSB<1>	GND	APOLLO_TRIG_1_B<3>	GND	CLK_SDI	GND	FPGA_REF_CLK_P<1>	GND	FPGA_REF_CLK_P<2>	GND
21	GND		GND	LPF_B<2>	GND	APOLLO_DO_P<0>	GND		CLK_SDO	GND	FPGA_REF_CLK_N<1>	GND	FPGA_REF_CLK_N<2>	GND
22	SERDES_C2M_P<6>	GND	CLK_STAT	LPF_B<3>	APOLLO_CKO_P<0>	APOLLO_DO_N<0>			GND	CLK_SCLK	GND		GND	SERDES_M2C_P<4>
23	SERDES_C2M_N<6>	GND	ART_STAT	GND	APOLLO_CKO_N<0>	GND		GND	APOLLO_RESETB<0>	EXT_TRIG	GND		GND	SERDES_M2C_N<4>
24	GND		GND		GND	VDDD_1P8_PG	GND		APOLLO_RESETB<1>	GND	SERDES_C2M_P<11>	GND	SERDES_C2M_P<10>	GND
25	GND				GP0_PRO3_TXEN0	VDDA_1P8_PG			GND	GND	SERDES_C2M_N<11>	GND	SERDES_C2M_N<10>	GND
26	SERDES_C2M_P<3>	GND		GND	GP1_PRO3_TXEN1	GND		GND	ART_CSB<2>	APOLLO_DO_P<1>	GND		GND	SERDES_C2M_P<9>
27	SERDES_C2M_N<3>	GND	GND		GND	GP2_PRO3_RXEN0	GND		ART_CSB<3>	APOLLO_DO_N<1>	GND		GND	SERDES_C2M_N<9>
28	GND				ART_CSB<0>	GP3_PRO3_RXEN1			GND	GND	SERDES_C2M_P<8>	GND	SERDES_C2M_P<5>	GND
29	GND			GND	ART_CSB<1>	GND		GND	NC***	GND	SERDES_C2M_N<8>	GND	SERDES_C2M_N<5>	GND
30	SERDES_C2M_P<2>	GND	GND		GND	APOLLO_CKO_P<1>	GND		TDI_TDO_LOOPBACK	I2C_SCL	GND		GND	SERDES_C2M_P<4>
31	SERDES_C2M_N<2>	GND			APOLLO_DIN_P<0>	APOLLO_CKO_N<1>			TDI_TDO_LOOPBACK	I2C_SDA	GND		GND	SERDES_C2M_N<4>
32	GND			GND	APOLLO_DIN_N<0>	GND		GND	3P3V_C2M	GND		GND	SERDES_M2C_P<3>	GND
33	GND		GND		GND	APOLLO_CKIN_P<0>	GND		NC***	GND		GND	SERDES_M2C_N3>	GND
34	SERDES_C2M_P<1>	GND			APOLLO_CSB<2>	APOLLO_CKIN_N<0>				EEPROM_GA0_GND	GND		GND	SERDES_M2C_P<2>
35	SERDES_C2M_N<1>	GND		GND	APOLLO_CSB<3>	GND		GND	EEPROM_GA1_GND	12P0V_C2M	GND		GND	SERDES_M2C_N<2>
36	GND	12P0V_C2M	GND		GND	APOLLO_CKIN_P<1>	GND		3P3V_C2M	GND		GND	SERDES_M2C_P<1>	GND
37	GND	12P0V_C2M			APOLLO_DIN_P<1>	APOLLO_CKIN_N<1>			GND	12P0V_C2M		GND	SERDES_M2C_N<1>	GND
38	SERDES_C2M_P<0>	GND		GND	APOLLO_DIN_N<1>	GND		GND	3P3V_C2M	GND	GND		GND	SERDES_M2C_P<0>
39	SERDES_C2M_N<0>	GND	GND		GND	VADJ_1P8V_C2M	GND	VADJ_1P8V_C2M	GND	3P3V_C2M	GND		GND	SERDES_M2C_N<0>
40	GND	12P0V_C2M		GND	VADJ_1P8V_C2M	GND	VADJ_1P8V_C2M	GND	3P3V_C2M	GND		GND	3P3V_C2M	GND

*VREF_A_M2C is not connected as FPGA should use internal 1.8V reference
**GND = PRSNT_M2C_L
***TCK & TMS are Not Connected
*** GND = HSPC_PRSNT_M2C_L

007

Figure 7. P2 FMC+ Pinout



14X40	M	L	K	J	H	G	F	E	D	C	B	A	Z	Y
1				GND		GND		GND		GND		GND		
2			GND				GND		GND		GND			
3			GND		GND		GND		GND		GND			
4				GND		GND		GND		GND		GND		
5				GND		GND		GND		GND		GND		
6			GND		GND		GND		GND		GND			
7									GND		GND			
8				GND		GND		GND		GND		GND		
9			GND		GND		GND			GND		GND		
10									GND		GND			
11				GND		GND		GND				GND		
12			GND		GND		GND			GND		GND		
13									GND	GND		GND		
14				GND		GND		GND			GND			
15			GND		GND		GND				GND			
16									GND	GND		GND		
17				GND		GND		GND		GND		GND		
18			GND		GND		GND				GND			
19									GND		GND			
20				GND		GND		GND		GND		GND		
21			GND		GND		GND			GND		GND		
22									GND		GND			
23				GND		GND		GND			GND			
24			GND		GND	GP4<1>	GND			GND		GND		
25					GP4<0>	GP5<1>			GND	GND		GND		
26				GND	GP5<0>	GND		GND			GND			
27			GND		GND	GP4<3>	GND				GND			
28					GP4<2>	GP5<3>			GND	GND		GND		
29				GND	GP5<2>	GND		GND		GND		GND		
30			GND		GND	SYNCOUTB_B0_P<0>	GND				GND			
31					SYNCINB_A1_P	SYNCOUTB_B0_N					GND			
32				GND	SYNCINB_A1_N	GND		GND		GND		GND		
33			GND		GND	SYNCOUTB_B0_P<1>	GND			GND		GND		
34					SYNCOUTB_B1_P	SYNCOUTB_B1_N<1>					GND			
35				GND	SYNCOUTB_B1_N<0>	GND		GND			GND			
36			GND		GND	SYNCOUTB_B0_P<3>	GND			GND		GND		
37					SYNCOUTB_B0_P<2>	SYNCOUTB_B1_N<3>			GND			GND		
38				GND	SYNCOUTB_B1_N<2>	GND		GND		GND	GND			
39			GND		GND		GND		GND		GND			
40				GND		GND		GND		GND		GND		

008

Figure 8. P3 FMC Pinout

DIGITAL INTERFACE

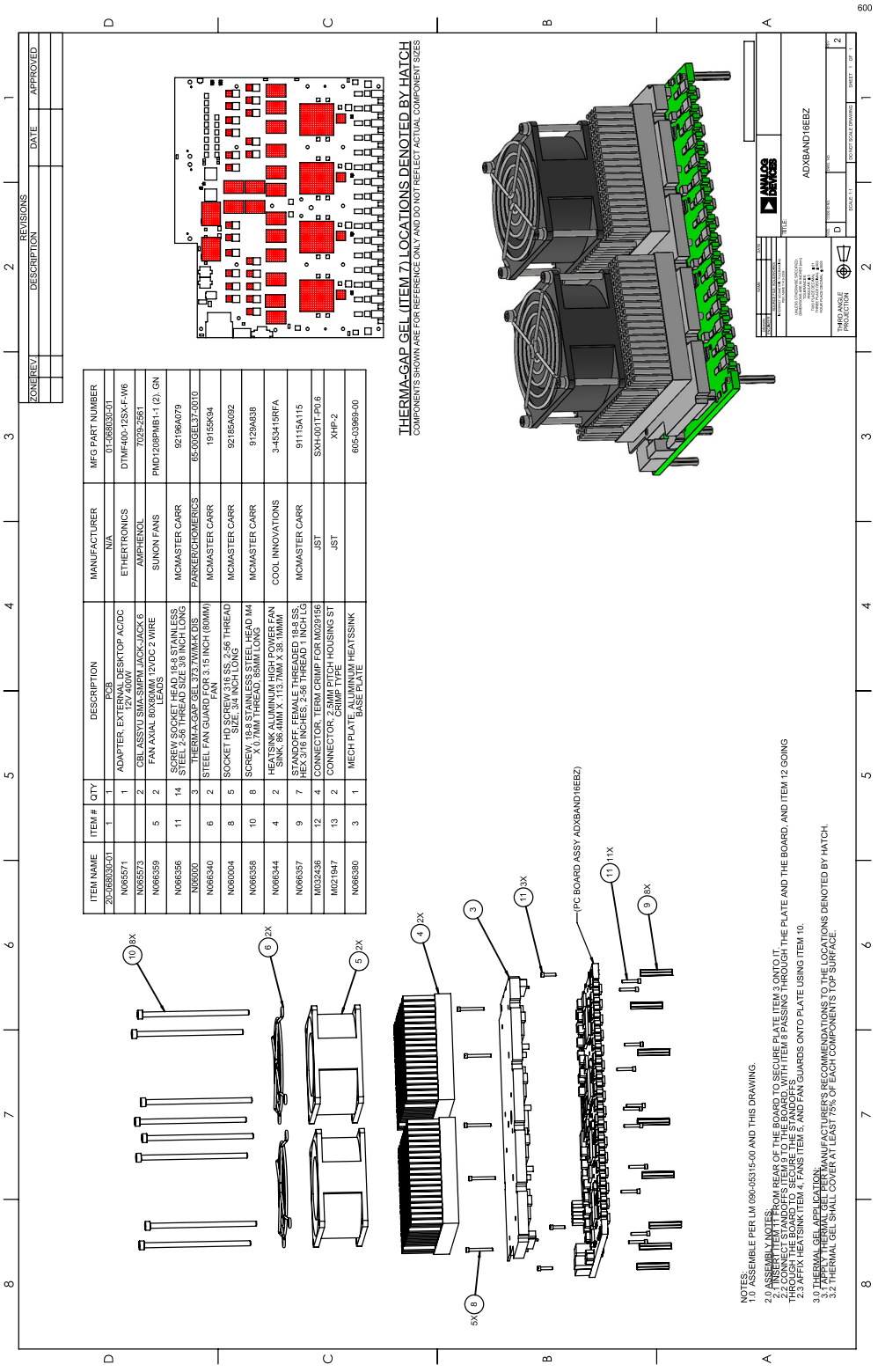


Figure 9. Thermal and Mechanical Housing

POWER DISTRIBUTION NETWORK

Note that the power numbers listed in Table 5 assume maximum load for worst-case analysis and estimates 78% regulator efficiency.

Table 5. Power Distribution Allocation

Part Number	Description	Quantity on Board	Voltage Rail Label	Voltage (V)	Maximum Current Consumption (mA)	Estimated Power per IC (mW)	Estimated Power per Chip Family (mW)	Estimated Powered per Chip (W)	Estimated Power per Chip Family (W)
AD9084	Apollo MxFE	4	VDDD_0P8	+0.8	+21200.0	16960.0	67840.0	49.6	198.6
			VDD_1P0_ADC	+0.8	+6800.0	5440.0	21760.0		
			VDD_1P0	+1.0	+1930.0	1930.0	7720.0		
			VDD_1P0_SERDES	+1.0	+2210.0	2210.0	8840.0		
			VDDD_1P8	+1.8	+250.0	450.0	1800.0		
			VDDD_1P8_MCS	+1.8	+100.0	180.0	720.0		
			VDDD_1P0_DAC_DIG	+1.0	+3570.0	3570.0	14280.0		
			VDDA_1P0_ADC	+1.0	+1890.0	1890.0	7560.0		
			VDDA_1P0_ADCBK	+1.0	+5470.0	5470.0	21880.0		
			VDDA_1P0_ADC_SCLK	+1.0	+240.0	240.0	960.0		
			VDDA_1P0_DAC	+1.0	+4790.0	4790.0	19160.0		
			VDDA_1P0_CK	+1.0	+2490.0	2490.0	9960.0		
			VDDA_1P8	+1.8	+1670.0	3006.0	12024.0		
			VDDA_1P8_DAC	+1.8	+340.0	612.0	2448.0		
			VREF_1P2V	+1.2	+1.3	1.6	6.2		
			VNEG_M1P0	−1.0	−400.0	400.0	1600.0		
LTC6953	Clock buffer	1	6953_3P3V_0	+3.3	+850	2805	2805	2.805	2.805
LTC6952	Clock buffer	1	6953_3P3V_1	+3.3	+850	2805	2805	2.805	2.805
ADF4351	Voltage-controlled oscillator (VCO)	1	6953_3P3V_0	+3.3	+160	528	528	0.528	0.528
			4382_5V_0	+5.0	+60	300	300	0.3	0.3
ADF4382A	Fractional-N PLL/VCO	4	3P3V_ISO	+3.3	+637	2102.1	8408.4	3.4171	13.6684
			5P0V_ISO	+5.0	+263	1315	5260		
ADF4030	10-channel precision synchronizer	1	4030_1P8V_LDO	+1.8	+28	50.4	50.4	2.8554	2.8554
			4030_3P3V_LDO	+3.3	+850	2805	2805		
ADMV8913	Tunable filter	32	3P3V_8913	+3.3	+0.125	0.4125	13.2	0.0004175	0.01336
			N2P5V_8913	−2.5	−0.002	0.005	0.16		
ADL8102	Low noise amplifier (LNA)	16	5P0V_810x	+5.0	+110	550	8800	0.55	8.8
ADL8100	LNA	16	5P0V_810x	+5.0	+200	1000	160000	1	16
ADRF5730	Digital step attenuator (DSA)	16	3P3V_5730	+3.3	+0.117	0.3861	6.1776	0.0007722	0.0123552
			N3P3V_5730	−3.3	−0.117	0.3861	6.1776		
							Total		246.3
							Total With PDN Efficiency		316.5

All RF and digital rails are biased from 12V through the 10-terminal power connector, a compatible AC adapter is included (rated to 400W), and the power connector is a Molex 39301100 dual-row, right-angle header (P1). The pins are detailed in Table 6 and shown in Figure 10.

Table 6. P1 12V DC Molex Power Connector Pin Descriptions

P1 Pin Number	1	2	3	4	5	6	7	8	9	10
Signal	GND	GND	GND	GND	GND	+12V	+12V	+12V	+12V	NC

POWER DISTRIBUTION NETWORK

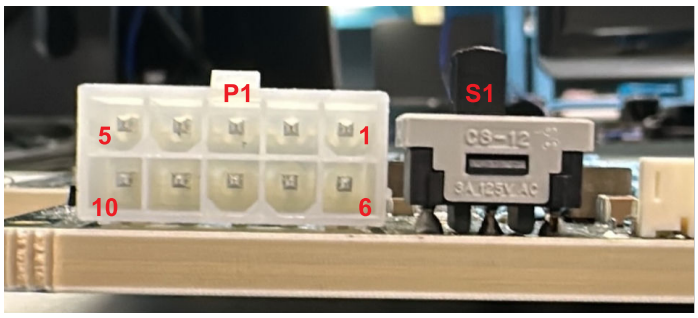


Figure 10. P1 12V DC Molex Power Connector Pin Locations

POWER TREE

Figure 11 shows the power distribution network for the ADXBAND16EBZ.

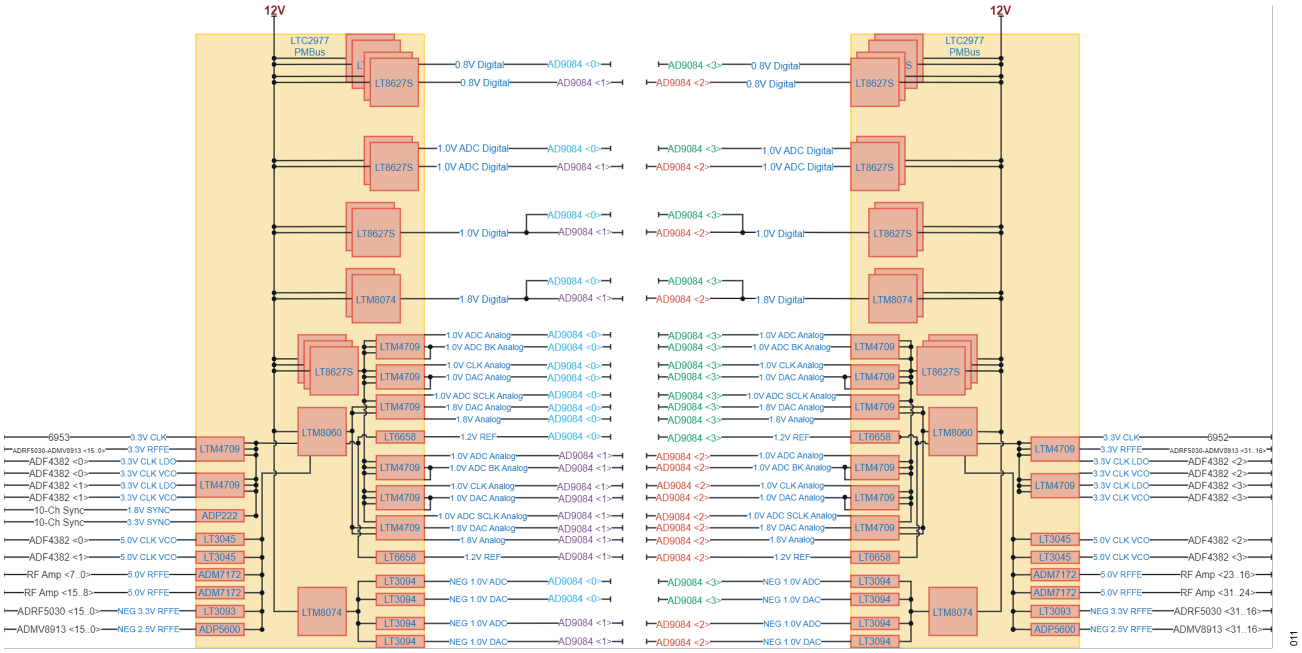


Figure 11. ADXBAND16EBZ Power Distribution Network

CLOCKING BLOCK DIAGRAM

Figure 12 shows the clocking block diagram.

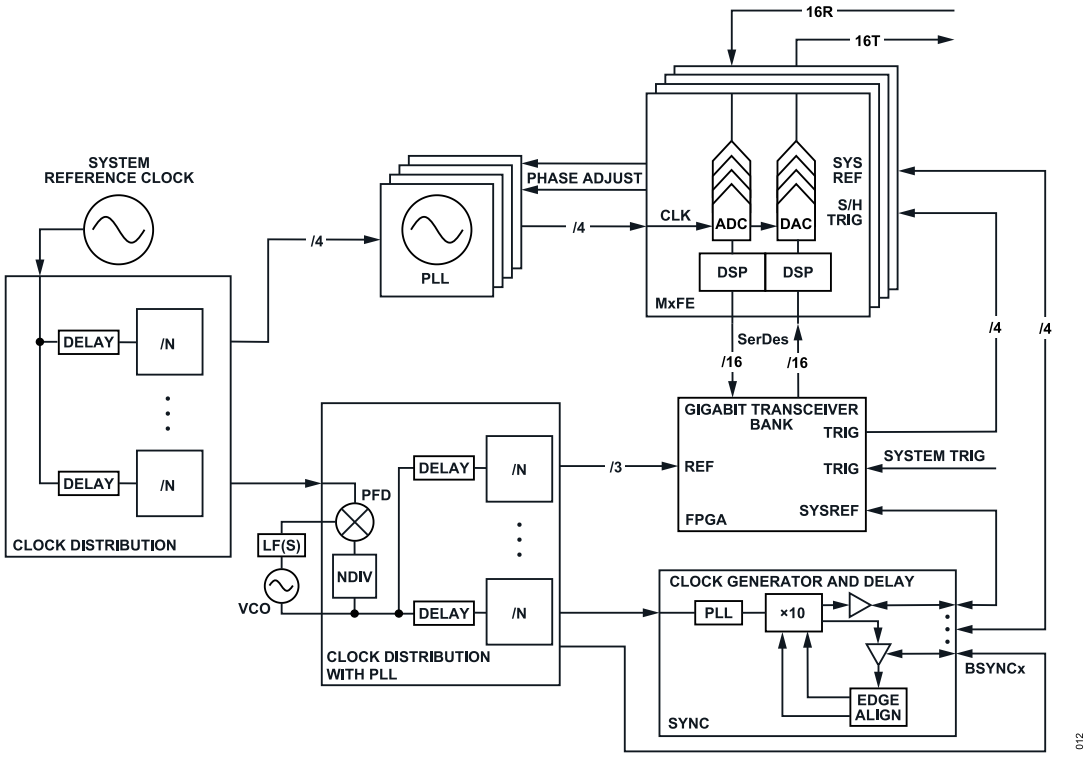


Figure 12. Clocking Block Diagram

CLOCK DEVICE SERIAL PERIPHERAL INTERFACE (SPI)

Figure 13 shows the SPI multidrop configuration for the clock tree.

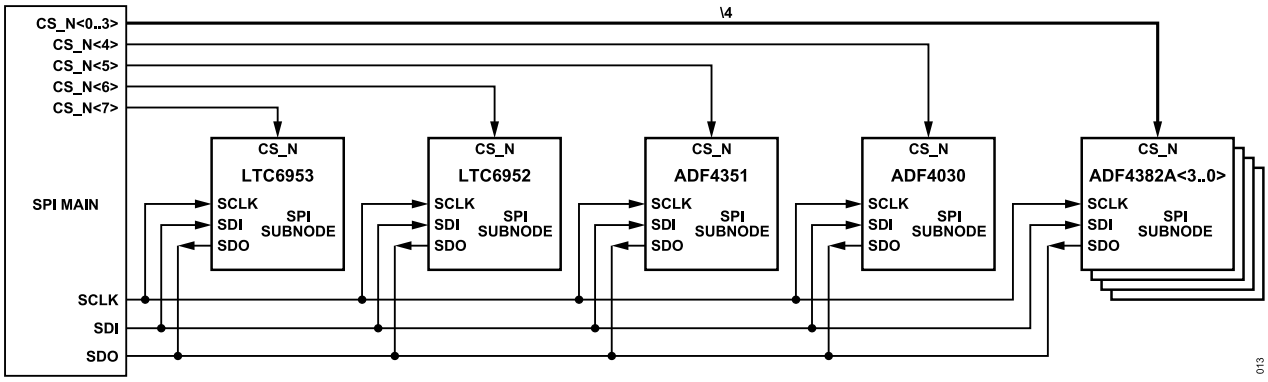


Figure 13. SPI Multidrop Configuration for the Clock Tree

CLOCKING BLOCK DIAGRAM

DIRECT MXFE CLOCKING

Provisions are available for directly driving the sampling clock of each MxFE data converter. An subminiature push-on micro (SMPM) plug is available on each clock input for this purpose, which connects to an AC-coupling capacitor that is not populated by default. Reference the schematic for more information. Table 7 lists the modifications required for direct clocking each channel.

Descriptions follow:

- External reference clock (Wenzel crystal oscillators recommended)
- External sync clock, which can synchronize multiple subarrays
- Phase-locked loop (PLL) synthesizer
  - ADF4382A
    - Generates Apollo frequency
    - Delay adjust (DELADJ)/delay strobe (DELSTR)
  - Apollo (AD9084)
    - Sample clock from the ADF4382A
    - Low frequency clock from LTC6953 (optional)
  - Clock buffer (LTC6953)
    - Generates SYSREFs
    - Generates FPGA reference
    - Generates JESD core clocks
  - ADF4030
    - Bidirectional SYNC signals
    - SYSREF replacement
    - Measure time to digital converter (TDC)

Table 7. Direct Clocking Modification

MxFE No.	SMPM Reference Designator	Modifications
0	J34	Depopulate C1595/C1596 and populate C1599/C1600
1	J35	Depopulate C1608/C1609 and populate C1612/C1613
2	J36	Depopulate C1621/C1622 and populate C1625/C1626
3	J37	Depopulate C1634/C1635 and populate C1638/C1639

CLOCKING USE CASE TABLE

See Table 8 for the Use Case 3 and Use Case 4 clocking details.

Table 8. Use Case Clocking Details for Use Case 1

		ADF4030		FGPA Core CLK			VCO	ADF4382A	
Use Case	REF CLK (MHz)	REF CLK (MHz)	REF_BSYNC (MHz)	REF CLK (MHz)	JTX (MHz)	JRX (MHz)	Frequency (MHz)	Reference Clock (MHz)	Frequency (GHz)
1	400	200	1.5625	200	200	200	3200	400	12.8

CLOCKING BLOCK DIAGRAM

SYNCHRONIZATION

Take the following four steps to synchronize (see Figure 14):

1. Align REF_BSYNC to BSYNCx within the precision synchronizer.
2. Align the internal SYSREF to the external BSYNC within the AD9084.
3. Null the external BYSYNC path delay deltas, which is important for extrapolation to large arrays.
4. Send TRIG to meet set up and hold to BSYNC.

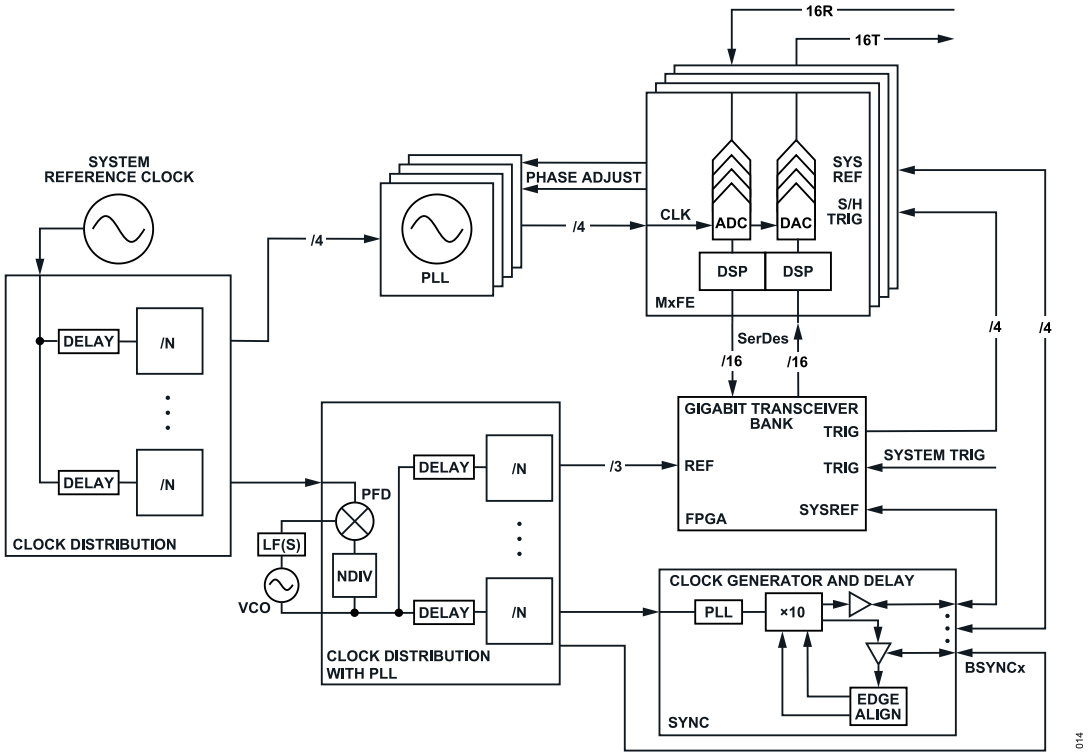


Figure 14. Synchronization

2-STAGE DAISY-CHAIN SYNCHRONIZATION EXAMPLE

See Figure 15 for a detailed example of a 2-stage daisy-chain synchronization.

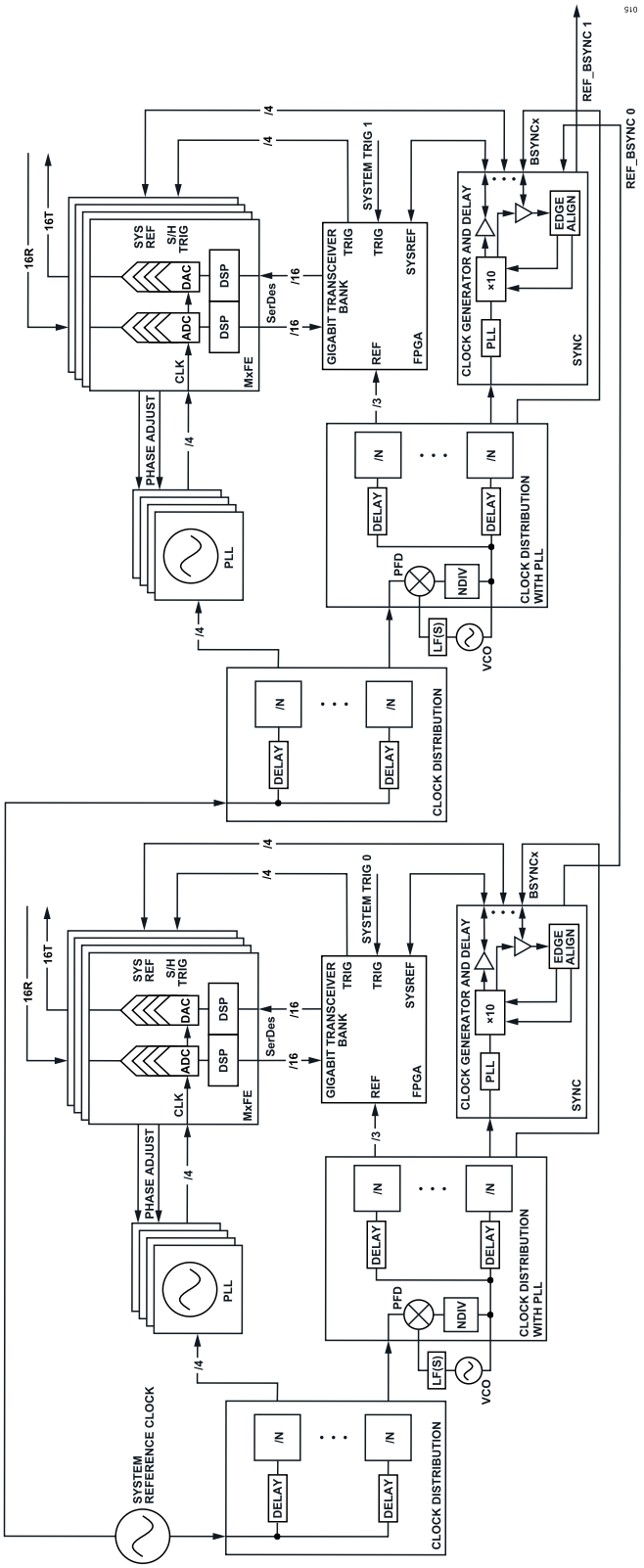


Figure 15. 2-Stage Daisy-Chain Synchronization Example

2-STAGE FANOUT SYNCHRONIZATION EXAMPLE

See Figure 16 for a detailed example of a 2-stage fanout synchronization, Figure 17 for an example of the performance improvement after phase and amplitude calibration across 16 channels, and Figure 18 for the complete system with heatsink, fans, VCU118 FPGA, and 16 transmit and 16 receive channels connected to the calibration board.

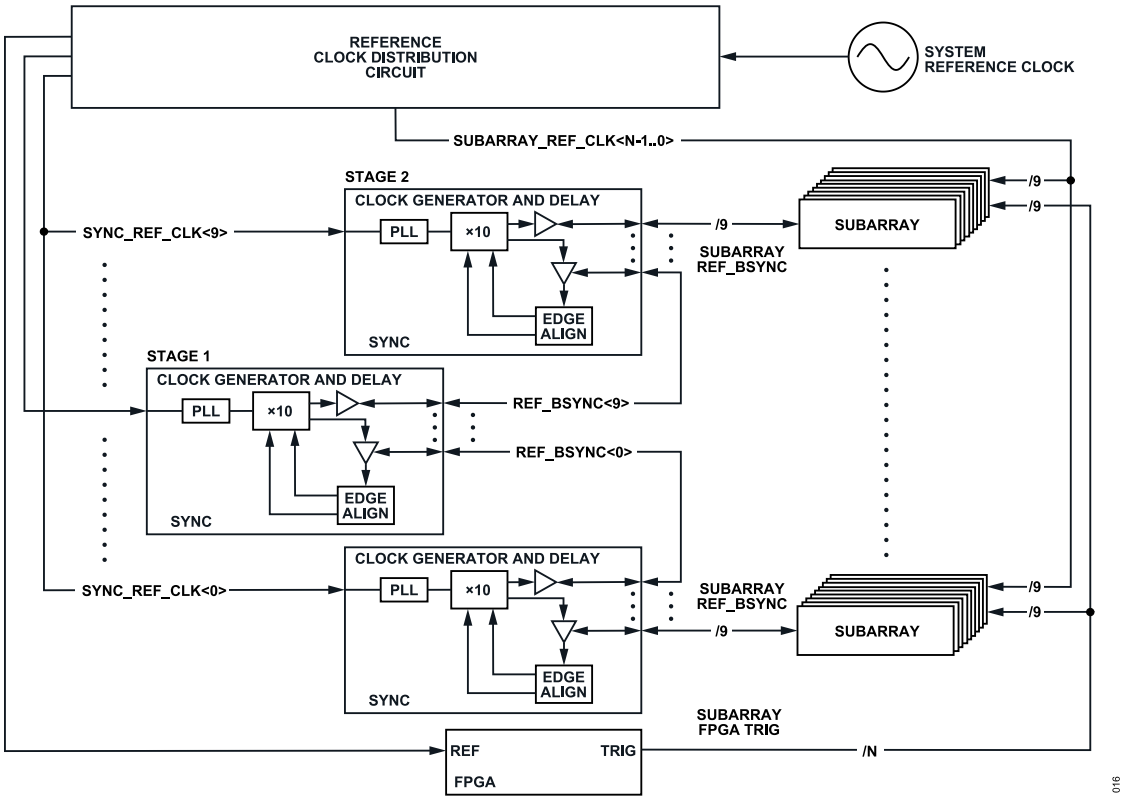


Figure 16. 2-Stage Fanout Synchronization Example

2-STAGE FANOUT SYNCHRONIZATION EXAMPLE

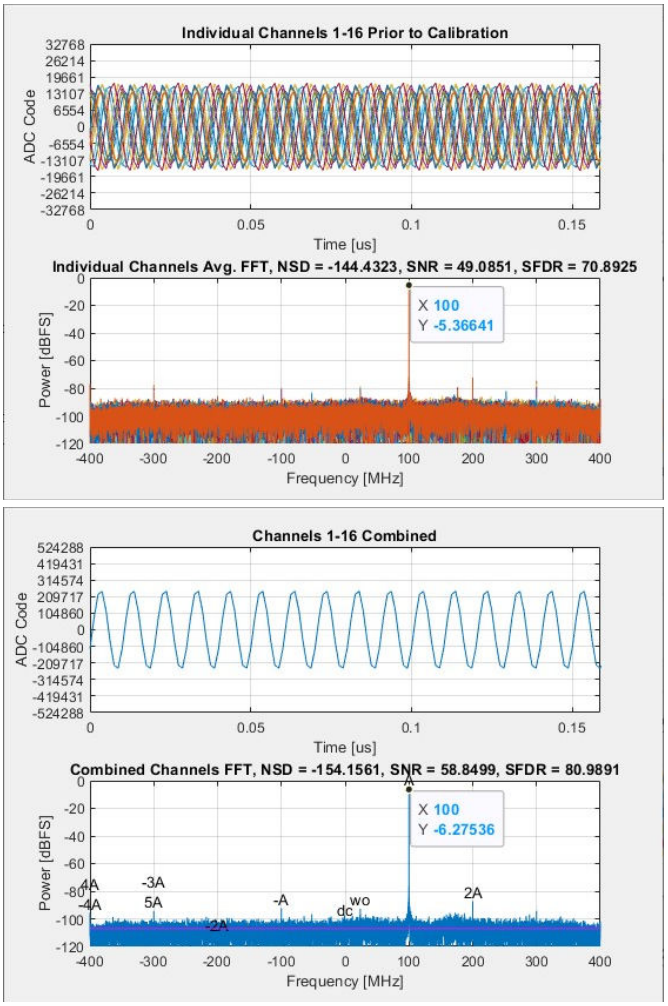


Figure 17. Example Performance Improvement after Phase and Amplitude Calibration Across 16 Channels

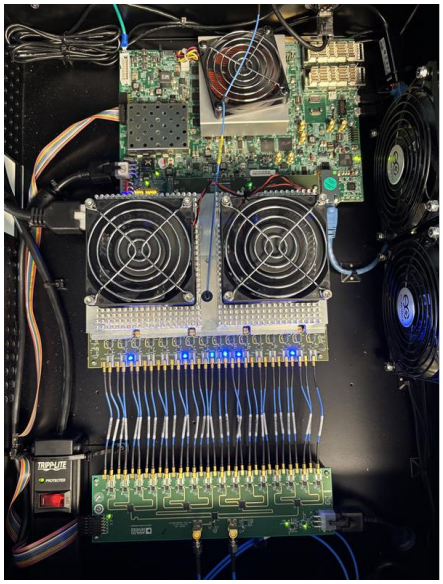


Figure 18. Complete System with Heatsink, Fans, VCU118 FPGA, and 16Tx and 16Rx Channels Connected to the Calibration Board

THERMAL MODEL

Figure 19 and Figure 20 show the typical thermal gradient based on modeling with a heatsink and fan assembly.

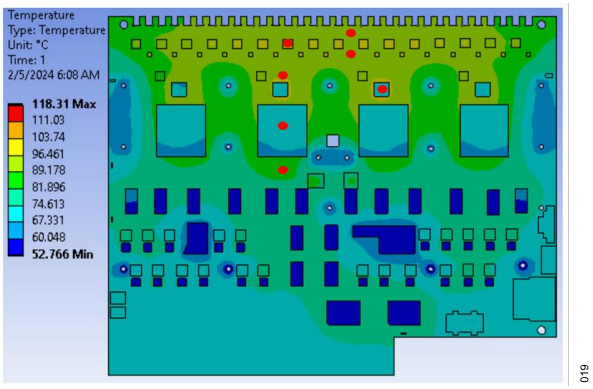


Figure 19. Top Layer

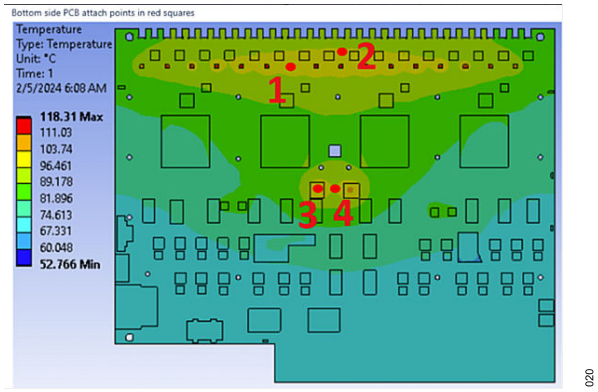


Figure 20. Bottom Layer

XILINX VIVADO LAB SOFTWARE DOWNLOAD

Go to the AMD website to download the Xilinx Vivado Lab software. Note that, it can only be downloaded after an AMD account is created. Once downloaded, take the following steps:

1. Extract the files using 7-zip to a known folder.
2. Run the xsetup application (note that you need an administrator role to complete this step).

Most versions of this software can be used. 2023.1 was the version used for this document.

HIGH-LEVEL ARCHITECTURE SUMMARY—VCU118 QUAD AD9084 REV. B DEVICE TREE

The design architecture describes a high performance RF data acquisition and generation platform built around the following:

- Xilinx VCU118 FPGA platform
- Four AD9084 multimode MxFE transceivers
- High-speed JESD204B and JESD204C data paths
- Extensive clock generation and clock distribution tree (ADF4030, ADF4382A, ADF4351)
- Multiple advanced eXtensible interface (AXI) peripherals (SPI, general-purpose input and output (GPIO), direct memory access (DMA), JESD, transceivers, and clock generators)

The device tree reflects a deeply synchronized, multidevice JESD204 fabric, precision timed clocking, and an FPGA fabric configured to support four MxFE lanes of both transmit and receive data.

CORE SYSTEM STRUCTURE

The base system includes the following:

- A top level model for **Analog Devices QuadAD9084 Rev.B FMC @Xilinx/vcu118**
- Inheritance from the base **vcu118.dtsi** file
- Global definitions for clock frequencies, JESD timings, dividers, and system reference constraints

These definitions act as the foundation that parameterizes the entire clocking and JESD structure.

CLOCKING ARCHITECTURE OVERVIEW

Clocking is the most complex and critical subsystem in this design.

Primary clock sources include the following:

- clkin_j1, which is a fixed clock at FREQ_J1_MHz (default 400MHz)
- Multistage clock synthesizers include the following:
  - LTC6953, which is a multioutput clock tree that feeds multiple ADF4382A transmit VCO references
  - LTC6952, which is a synchronized high-speed clock generation for the FPGA gigatransfer lanes
  - ADF4382A (×4), which is a high frequency synthesizers that delivers 12.8GHz VCO outputs
  - ADF4351, which is an auxiliary PLL/VCO
  - ADF4030, which handles SYSREF and BSYNC generation (system synchronization)

Clock Distribution Roles

The clock distribution roles provide the following:

- FPGA reference clocks
- Gigatbit transceiver clocks (receive and transmit)
- System reference (SYSREF) for JESD204 Subclass 1 alignment
- Per device VCO clocks for the four AD9084s

These nodes define clock names, clock output names, dividers, and delays, forming the large synchronized tree necessary for deterministic latency JESD operation.

JESD204 FABRIC ARCHITECTURE

The design implements an advanced multilink JESD204 topology integrating the following:

- JESD204 receive cores (axi_ad9084_rx_jesd)
- JESD204 transmit cores (axi_ad9084_tx_jesd)
- AD9084 transceivers (4×), each with multiple JESD links
- AXI-ADxCVR transceiver interfaces (gigabit transceivers) for both receive and transmit
- Clock and SYSREF injection paths

Each AD9084 instance declares the following:

- JESD top-level device role
- JESD link mappings (framer/deframer A0/B0 channels)
- Logical and physical lane mapping
- JESD clock connections (device clocks, lane clocks)

The fabric consists of the following:

- Four independent MxFE devices, each having
  - Bidirectional JESD lanes (transmit DAC and receiver ADC)
  - Connections to upstream and downstream framer and deframer blocks via JESD links

This entire structure is synchronized via LTC6952 → ADF4030 SYSREF paths.

HIGH-LEVEL ARCHITECTURE SUMMARY—VCU118 QUAD AD9084 REV. B DEVICE TREE

AD9084 MULTICHIP SETUP (FOUR DEVICE MULTIDEVICE SYNC)

The following four transceivers are instantiated:

- trx0_ad9084
- trx1_ad9084
- trx2_ad9084
- trx3_ad9084

Each device includes the following:

- SPI configuration
- GPIO exports for DSA control
- JESD204 multidevice synchronization
- High-speed inter-chip (HSIC) interfacing (via axi_hsci_*)
- Shared device profile configuration (firmware .bin profile)

Each AD9084 device declares the following:

- JESD lane mappings
- Nyquist zone configuration
- Transmit drive settings
- PLL and clock source connections
- Gigabit transceiver lane routing

The device tree defines all four as part of a coherent multichip JESD204 chain.

FPGA-SIDE DATAPATH

The FPGA includes the following:

- DMA engines as follows:
  - Receive DMA (rx_dma)
  - Transmit DMA (tx_dma)

  Each of these provides AXI-DMAC support, stream data in-to/out of the AD9084 cores, and interrupt signaling through an AXI interrupt controller.
- Data offload blocks as follows:
  - axi_data_offload_rx
  - axi_data_offload_tx

  These blocks support buffered or burst-mode streaming and lower latency or high throughput modes.

SPI, I²C, AND GPIO INFRASTRUCTURE

The SPI buses are used to configure the following:

- ADF4382A synthesizers
- ADF4030 SYSREF engine
- ADF4351 PLL
- LTC6952 and LTC6953 clock trees
- AD9084 chips (4× MxFE devices)

The I²C bus is used to access the following:

- Electrically erasable programmable read-only memory (EEPROM)
- LTC2977 power monitors (east and west rails)

The GPIO controllers (axi_gpio_2) provide the following:

- Input and output lines for board power-good indicators
- Reset lines for AD9084 chips
- Control for DSA attenuators
- Control lines for filters and multiplexers

The MUX controllers provide the following:

- Low-pass filter (LPF) and high-pass filter (HPF) selectors implemented through gpio-mux
- Control via the GPIO bits mapped to the filter banks

ANALOG FRONT-END HARDWARE (DSA, LPF, AND HPF)

The design contains DSAs (4× ADRF5730 devices). Each corresponds to a channel (0 to 3) and is connected to per device AD9084 GPIO controllers.

It also contains filter multiplexers: LPF MUX and HPF MUX. These multiplexors allow flexible analog front-end signal conditioning paths controlled via the GPIO.

SYSTEM-LEVEL SYNCHRONIZATION AND TRIGGERING

The **axi_adf4030_trig** block provides the following:

- JESD system trigger distribution
- Input and output BSYNC routing
- Connections to the ADF4030 SYSREF generator
- IO channels used for timing alignment

This block ensures coherent multichip alignment across all AD9084 devices.

HIGH-LEVEL ARCHITECTURE SUMMARY—VCU118 QUAD AD9084 REV. B DEVICE TREE

SUPPORTING PERIPHERALS

The design also incorporates the following:

- An Ethernet subsystem, axi_ethernet with fixed MAC address used for control plane or test data streaming.
- System identification, axi_sysid_0 used for versioning, building of metadata, and system identification.
- A high-speed chip interface, HSCI, (4×) that includes the following:
  - axi_hsci_0
  - axi_hsci_1
  - axi_hsci_2
  - axi_hsci_3

These peripherals provide extremely high-bandwidth communication between the FPGA logic and the AD9084 devices.

OVERALL SYSTEM SUMMARY

This distributed transmission system (DTS) describes a fully synchronous, tightly coupled multichip RF transceiver system that features the following:

- Four AD9084 MxFE devices, each with full JESD204B and JESD204C receive and transmit links
- Highly complex clock generation and distribution tree supporting deterministic latency
- Multiple PLLs and synthesizers orchestrating multiGHz timing domains
- Extensive FPGA infrastructure for JESD, DMA, clocking, and data offload
- Advanced SYSREF and BSYNC synchronization fabric enabling coherent multichip operation
- Robust GPIO, SPI, and I^2C control for board management and mixed-signal tuning
- Filter and attenuator control subsystems for analog signal conditioning
- Coordinated timing, triggers, and JESD synchronization across the entire platform

This system is essentially a blueprint for a state-of-the-art RF data converter system capable of multigigabit ADC and DAC streaming and precise, deterministic phase alignment across four high-performance transceivers.

NODE BY NODE LINUX DEVICE TREE SUMMARY

The node by node Linux device tree summary follows:

Top level/(root)

- **model**: "Analog Devices Quad-AD9084 Rev.B FMC @Xilinx/vcu118" — human-readable platform ID.
- **Includes**: vcu118.dtsi, several dt-bindings headers; numerous #define constants for clock frequencies, SYSREF rates, JESD helpers. (Used by subnodes such as LTC6952, LTC6953, ADF4030, AD9084 blocks.)

/clocks

- clkin_j1: clock@0
  - **compatible**: fixed-clock
  - **#clock-cells**: <0>
  - **clock-frequency**: <FREQ_J1_MHz> (default 400 MHz)
  - **clock-output-names**: "clkin_j1"
  - **Role**: Primary fixed input reference used upstream by LTC6953.

/vref: regulator-vref

- **compatible**: regulator-fixed
- **regulator-name**: "vref-ad5592"
- **regulator-min-microvolt / max**: 2500000µV
- regulator-always-on
- **Role**: Reference for the AD5592R mixed-signal device.

/calibration-board-control@0 (GPIO ADC/DAC aggregator)

- **compatible**: adi, one-bit-adc-dac
- **out-gpios**: 4 lines from &axi_gpio_2 (Bit 33, Bit 32, Bit 34, Bit 35)
- **#address-cells / #size-cells**: <1> / <0>
- **Channels**:
  - channel@0 label "5045_V1"
  - channel@1 label "5045_V2"
  - channel@2 label "CTRL_IND"
  - channel@3 label "CTRL_RX_COMBINED"
- **Role**: Board-level control and monitor pins abstracted as simple ADC and DAC like channels.

JESD status fake IIO devices (for sysfs visibility) follows:

- /axi-jesd204-rx@0
  - **compatible**: adi, iio-fake-platform-device
  - **adi,faked-dev**: &axi_ad9084_rx_jesd
  - **adi,attribute-names**: "status","encoder","lane0_info"... "lane15_info"
  - **label**: "axi-jesd204-rx"
  - **Role**: Exposes JESD RX status channels to userspace for debug.
- /axi-jesd204-rx@1
  - **compatible**: adi, iio-fake-platform-device
  - **adi,faked-dev**: &axi_ad9084_tx_jesd
  - **adi,attribute-names**: "status","encoder"
  - **label**: "axi-jesd204-tx"
  - **Role**: Exposes JESD TX status.

/axi-adxcvr-rx@2, /axi-adxcvr-tx@4 (fake platform)

- **compatible**: adi, iio-fake-platform-device
- **adi,faked-dev**: pointers to &axi_ad9084_adxcvr_rx/ &axi_ad9084_adxcvr_tx
- **adi,attribute-names**: eyescan/PRBS, reg access, etc.
- **label**: "axi-adxcvr-rx"/"axi-adxcvr-tx"
- **Role**: User-visible knobs for gigabit transceivers.

/gpio-board-control@0

- **compatible**: adi, one-bit-adc-dac
- **out-gpios**: &axi_gpio_2 Bit 30 (active-high)
- **in-gpios**: &axi_gpio_2 Bits[36:44] (9 inputs)
- **label**: "gpio-ctrl"
- **channels**: 12v_PG, Vdd_0.8v_PG, Vdda_1.0v_PG, Vddd_1.8v_PG, Vdda_1.8v_PG, Vneg_1.0v_PG, adf4382A_5.0v_PG, Artemis_Status, Clk_Status, admv8913_cs_n
- **Role**: Consolidated power-good and status GPIOs plus one output control.

4× DSAs, ADRF5730-compatible (modeled as HMC425A) follows:

- /adrf5730_0 … _3
  - **compatible**: adi, hmc425a
  - **ctrl-gpios**: six pins from each AD9084 node's exported GPIO controller (&trxX_ad9084 indices, 0 to 5) per device DSA control.
  - **label**: "dsa0" … "dsa3"
- **Role**: Per channel RF attenuation, controlled by MxFE GPIO exports.

NODE BY NODE LINUX DEVICE TREE SUMMARY

Filter mux controllers and logical muxes as follows:

- mux_lpf0: mux-controller0
  - **compatible**: gpio-mux
  - **#mux-control-cells**: <0>
  - **mux-gpios**: &axi_gpio_2, Bit 26 to Bit 29
- mux_hpf0: mux-controller1
  - **compatible**: gpio-mux
  - **mux-gpios**: &axi_gpio_2, Bit 22 to Bit 25
- lpf_mux: mux-lpf
  - **compatible**: adi,gen_mux
  - **mux-controls**: <&mux_lpf0>
  - **mux-state-names**: "0" … "15"; **mux-default-state**: "11"; **label**: "lpf-ctrl"
- hpf_mux: mux-hpf
  - **compatible**: adi,gen_mux
  - **mux-controls**: <&mux_hpf0>
  - **mux-state-names**: "0" … "15"; **mux-default-state**: "13"; **label**: "hpf-ctrl"
- **Role**: Software-selectable LPF and HPF banks via GPIO address lines.

INTERRUPT CONTROLLER AND ON-BOARD PERIPHERALS FROM THE REFERENCED NODES

The interrupt controller and on-board peripherals from the referenced nodes follows:

&axi_intc

- **xlnx, kind-of-intr**: <0xffff04f0>
- **xlnx, num-intr-inputs**: <17>
- **Role**: central interrupt controller for AXI peripherals (DMAs, SPI, JESD)

&axi_ethernet

- **local-mac-address**: 00:0a:35:00:90:84 (static)
- **Role**: Ethernet datapath and control access

&axi_iic_main

- **compatible**: xlnx, axi-iic-2.1, xlnx, xps-iic-2.00.a
- **Note**: ensures HWMON devices probe correctly

&fmc_i2c (FMC I²C bus)

- **eeprom@50** — compatible: at24, 24c02, Reg: 0x50
- **ltc2977_w@5f** — West power controller and monitor
- **ltc2977_e@60** — East power controller and monitor
- **Role**: board EEPROM and two LTC2977s for power rails

&amba_pl (AXI programmabe-logic (PL) region)

DMA engines follow:

- **rx_dma: dma@7c420000/tx_dma: dma@7c430000**
  - **compatible**: adi, axi-dmac-1.00.a
  - **reg**: base + 64KB window
  - **#dma-cells**: <1>; **#clock-cells**: <0>
  - **interrupt-parent**: &axi_intc
  - **interrupts**: RX <12 2>, TX <13 2>
  - **clocks**: &clk_bus_0
  - **Role**: stream receive and transmit to and from memory and AD9084 cores

AD9084 AXI datapath endpoints follow:

- axi_ad9084_core_rx: axi-ad9084-rx-hpc@44a10000
  - **compatible**: adi, axi-ad9081-rx-1.0
  - **reg**: <0x44a10000 0x8000>
  - **dmas**: <&rx_dma 0>; **dma-names**: "rx"
  - **spibus-connected**: <&trx0_ad9084>
  - **jesd204-device**; **#jesd204-cells**: <2>
  - **jesd204-inputs**: <&axi_ad9084_rx_jesd 0 FRAMER_LINK_B0_RX>
  - **Role**: receive data interface for first AD9084 chain into the DMA via the JESD receiver IP
- axi_ad9084_core_tx: axi-ad9084-tx-hpc@44b10000
  - **compatible**: adi, axi-ad9081-tx-1.0
  - **reg**: <0x44b10000 0x4000>
  - **dmas**: <&tx_dma 0>; **dma-names**: "tx"
  - **clocks**: <&trx0_ad9084 1>; **clock-names**: "sampl_clk"
  - **spibus-connected**: <&trx0_ad9084>
  - adi,axi-pl-fifo-enable
  - **adi,axi-data-offload-connected**: <&axi_data_offload_tx>
  - **jesd204-device**; **jesd204-inputs**: <&axi_ad9084_tx_jesd 0 DEFRAMER_LINK_B0_TX>
  - **Role**: transmit datapath for first AD9084, clocked by device sample clock, with optional PL first in, first out (FIFO) and data offload

JESD cores (FPGA) follow:

- axi_ad9084_rx_jesd: axi-jesd204-rx@44a90000
  - **compatible**: adi, axi-jesd204-rx-1.0
  - **reg**: <0x44a90000 0x4000>
  - **interrupts**: <14 2> via &axi_intc
  - **clocks**: s_axi_aclk = &clk_bus_0, link_clk = &axi_adxcvr_rx 1, device_clk = &ltc6952 4, lane_clk=&axi_adxcvr_rx 0
  - **#clock-cells**: <0>; **clock-output-names**: "jesd_rx_lane_clk"
  - **jesd204-device**; **jesd204-inputs**: <&axi_adxcvr_rx 0 FRAMER_LINK_B0_RX>
  - **adi, lanes-disable-mask**: <0x0F00> (disable lanes 8 to 11)
  - **Role**: JESD204 receive endpoint tying gigatransfer lane clocks to device clock
- axi_ad9084_tx_jesd: axi-jesd204-tx@44b90000
  - **compatible**: adi, axi-jesd204-tx-1.0
  - **reg**: <0x44b90000 0x4000>
  - **interrupts**: <15 2>
  - **clocks**: s_axi_aclk = &clk_bus_0, link_clk = &axi_adxcvr_tx 1, device_clk = &ltc6952 3, lane_clk = &axi_adxcvr_tx 0
  - **#clock-cells**: <0>; **clock-output-names**: "jesd_tx_lane_clk"
  - **jesd204-device**; **jesd204-inputs**: <&axi_adxcvr_tx 0 DEFRAMER_LINK_B0_TX>
  - **Role**: JESD204 transmit endpoint

INTERRUPT CONTROLLER AND ON-BOARD PERIPHERALS FROM THE REFERENCED NODES

AXI ADXCVR (gigatransfer transceivers) follow:

- axi_ad9084_adxcvr_rx: axi-adxcvr-rx@44a60000
  - **compatible**: adi, axi-adxcvr-1.0
  - **reg**: <0x44a60000 0x1000>
  - **clocks**: conv = &ltc6952 2
  - **#clock-cells**: <1>; **clock-output-names**: "rx_gt_clk","rx_out_clk"
  - **adi,sys-clk-select**: <XCVR_QPLL>; **adi,out-clk-select**: <XCVR_REFCLK>; **adi,use-lpm-enable**
  - **jesd204-device**; **jesd204-inputs**: <&ltc6953 0 FRAMER_LINK_B0_RX>
  - **Role**: receive gigatransfer transceiver with reference and converter clocks, feeding JESD receiver
- axi_ad9084_adxcvr_tx: axi-adxcvr-tx@44b60000
  - **compatible**: adi, axi-adxcvr-1.0
  - **reg**: <0x44b60000 0x1000>
  - **clocks**: conv = &ltc6952 2
  - **#clock-cells**: <1>; **clock-output-names**: "tx_gt_clk","tx_out_clk"
  - **adi,sys-clk-select**: <XCVR_QPLL>; **adi,out-clk-select**: <XCVR_REFCLK>
  - **jesd204-device**; **jesd204-inputs**: <&ltc6953 0 DEFRAMER_LINK_B0_TX>
  - **Role**: transmit gigatransfer transceiver

System ID follows:

- axi_sysid_0: axi-sysid-0@45000000
  - **compatible**: adi, axi-sysid-1.00.a
  - **reg**: 64 KB
  - **Role**: build and version metadata readable in software

Data offload follows:

- **axi_data_offload_rx: data_offload_rx@7c450000/axi_data_offload_tx: data_offload_tx@7c440000**
  - **compatible**: adi, axi-data-offload-1.0.a
  - **reg**: 64 KB each
  - **Role**: optional packetization and latency control blocks for receive and transmit streams

SPI controllers follow:

- axi_spi_2: spi@44A80000
  - **compatible**: xlnx, axi-quad-spi-3.2, xlnx, xps-spi-2.00.a
  - **reg**: <0x44A80000 0x1000>; **interrupts**: <7 0>
  - **num-cs**: 0x8; **fifo-size**: <16>; **xlnx,num-ss-bits**: <8>
  - **Role**: main SPI bus for clock devices (LTC6952, LTC6953, ADF4030, ADF4351, ADF4382A ×4).
- axi_spi3: spi@44B80000
  - **compatible**: xlnx,xps-spi-2.00.a
  - **reg**: <0x44B80000 0x10000>; **interrupts**: <16 0>
  - **num-cs**: 0x8; **fifo-size**: <16>
  - **Role**: SPI bus for AD5592R mixed-signal device

GPIO follows:

- axi_gpio_2: gpio@7c470000
  - **compatible**: xlnx, axi-gpio-2.0, xlnx,xps-gpio-1.00.a
  - **reg**: <0x7c470000 0x1000>; **interrupts**: <8 2>; **interrupt-controller**
  - **#gpio-cells**: <2>; **#interrupt-cells**: <2>
  - **xlnx,is-dual**: 1 (two 32-bit banks)
  - **gpio-hog**: adf4382A-en-hog → sets Bit 31 high (ADF4382A_5V_EN) at boot
  - **Role**: board GPIOs (power-good, mux selects, resets, and control)

Clock generator (fabric) follows:

- hsci_clkgen: axi-clkgen@44ad0000
  - **compatible**: adi, axi-clkgen-2.00.a
  - **reg**: <0x44ad0000 0x10000>
  - **clocks**: s_axi_aclk = &clk_bus_0, clkin1 = &clk_bus_0
  - **#clock-cells**: <0>; **clock-output-names**: "hsci_clkgen"
  - **Role**: provides HSPI and high-speed internal (HIS) fabric clock for HSCI blocks

HSCI blocks (HSCI to AD9084) follow:

- **axi_hsci_0 … _3** at 0x7c500000, 0x7c600000, 0x7c700000, 0x7c800000
  - **compatible**: adi, axi-hsci-1.0.a
  - **clocks**: <&hsci_clkgen> named "pclk"
  - **adi,hsci-interface-speed-mhz**: <HSCI_SPEED> (800MHz)
  - **Role**: per AD9084 high-speed serial control and data sideband; each AD9084 node references one HSCI

INTERRUPT CONTROLLER AND ON-BOARD PERIPHERALS FROM THE REFERENCED NODES

The ADF4030 trigger block follows:

- axi_adf4030_trig: axi-adf4030-trig@7c900000
  - **compatible**: adi, axi-adf4030-trig-1.0.a
  - **reg**: <0x7c900000 0x1000>
  - **clocks**: <&ltc6952 4> as "device_clk"
  - **jesd204-device**; **jesd204-inputs** from <&adf4030> FRAMER/DEFRAMER roles
  - **io-channels**: <&adf4030 4> "bsync"
  - **Role**: system-wide JESD alignment trigger and BSYNC routing hub for the ADF4030

&axi_spi3 children (AD5592R) follow:

- adc_ad5592r: ad5592r@0
  - **compatible**: adi, ad5592r
  - **reg**: <0>; **spi-max-frequency**: 390000 Hz; **spi-cpol**
  - **vref-supply**: <&vref>
  - **Channels**:
    - channel@0 — adi, mode=<CH_MODE_ADC>
    - channel@1 — adi, mode=<CH_MODE_ADC>
    - channel@5 — adi, mode=<CH_MODE_ADC>
    - channel@7 — adi, mode=<CH_MODE_ADC>
  - **Role**: auxiliary ADC inputs (and potentially DAC/IO if reconfigured)

&axi_spi_2 children (clocking and sync tree) follow:

- ltc6953@7
  - **compatible**: adi, ltc6953
  - **reg**: <7>; **spi-max-frequency**: 10MHz
  - **label**: "ltc6953"
  - **clocks**: clkin = &clkin_j1
  - **jesd204-device** with inputs from &ltc6952 (both FRAMER/DEFRAMER links)
  - **#clock-cells**: <1>; **clock-output-names**: 11 outputs (ltc6953_out0 … out10)
  - **adi,pulse-generator-mode**: <2> (four pulses); **adi, controller-srqmd-en-during-sync-en**
  - **channel@0..3**: Four ADF4382A REFCLKs; divider/delay per output
  - **channel@6**: "ADF4030_BSYNC_6" (BSYNC fanout)
  - **channel@7**: "J50_J51_UFL_Output" (SYSREF out)
  - **channel@8**: "LTC6952_REF_IN"
  - **channel@9**: "LTC6852_EZS_SRQ_IN" with adi, sysref-mode = <2> (request pass-through)
  - **channel@10**: "ADF4351_REF_IN"
  - **Role**: root fanout clock tree for VCOs, FPGA references, and SYSREF distribution into LTC6952 or ADF4030

adf4382@0 … @3 follows:

- **compatible**: adi, adf4382
- **reg**: <0|1|2|3>; **spi-max-frequency**: 1MHz
- **adi, output-power-value**: 4/5; **adi, power-up-frequency**: 12.8GHz
- adi, cmos-3v3
- **clocks**: ref_clk from &ltc6953 outputs (0→ch0, 1→ch3, 2→ch2, 3→ch1 mapping)
- **#clock-cells**: <1>; **clock-output-names**: "adf4382_X_out_clk"
- **#io-channel-cells**: <1>
- **Role**: high-frequency VCOs feeding individual AD9084 device clocks (via each MxFE DEC_CLK)

adf4351@5 follows:

- **compatible**: adi, adf4351; **reg**: <5>; **spi-max-frequency**: 10MHz
- **#clock-cells**: <1>; **clock-names**: "clkin" from &ltc6953 10
- **adi, channel-spacing**: 1MHz; **adi, power-up-frequency**: 2GHz; multiple PLL configuration props (charge pump current, polarity, mute-till-lock, and muxout)
- **Role**: external VCO input for the LTC6952, reference generation

ltc6952@6 follows:

- **compatible**: adi, ltc6952; **reg**: <6>; **spi-max-frequency**: 10 MHz
- **label**: "ltc6952"
- **clocks**: clkin = &ltc6953 8, vcoin = &adf4351 0
- **jesd204-device**; **jesd204-inputs** from &axi_adf4030_trig (FRAMER/DEFRAMER)
- **clock-output-names**: 11 outputs (ltc6952_out0 … out10); **#clock-cells**: <1>
- **adi, vco-frequency-hz**: 2GHz; **adi, pulse-generator-mode=<2>; **adi,sync-via-ezs-srq-enable
- **channel@2**: "FPGA_REFCLK" (divider + delays)
- **channel@3**: "FPGA_CORE_CLK_TX"
- **channel@4**: "FPGA_CORE_CLK_RX"
- **channel@5**: "ADF4030_REF_IN"
- **channel@6**: "ADF4030_BSYNC_8"
- **channel@7**: "J48_J49_UFL_Output"
- **Role**: core JESD and gigatransfer reference distribution to FPGA and the ADF4030 (SYSREF)

INTERRUPT CONTROLLER AND ON-BOARD PERIPHERALS FROM THE REFERENCED NODES

adf4030@4 (SYSREF and BSYNC generator) as follows:

- **compatible**: adi,adf4030; **reg**: <4>; **spi-max-frequency**: 1MHz
- **#clock-cells**: <1>; **#io-channel-cells**: <1>
- **clocks**: refin = &ltc6952 5
- **clock-output-names**: adf4030_bsync_0 … _9
- **jesd204-device**; **jesd204-sysref-provider**
- **adi, vco-frequency-hz**: 2.5 GHz; **adi, bsync-frequency-hz**: SYSREF_CLK_MHz; **adi, bsync-secondary-frequency-hz**: 12.5MHz
- **adi, bsync-autoalign-reference-channel**: <8> (LTC6952_OUT_6)
- **Channels**:
  - channel@0..3: "APOLLO_SYSREF_0/1/2/3" with adi, output-en, auto-align-on-sync-en, adi, rcm = <1>, adi, link-rx-en, adi, float-rx-en
  - channel@4: "FPGA_SYSREF_0" (RCM = 62, output enabled)
  - channel@6: "LTC6953_OUT_6" (input link/float-rx)
  - channel@8: "LTC6952_OUT_8" (input link/float-rx)
  - channel@9: "J46_J47_UFL" (output enabled, RCM = 62)
- **Role**: main SYSREF or BSYNC distribution and alignment engine

&fmc_spi children (4× AD9084 MxFE devices) as follows:

- **Pattern**: four nearly identical AD9084 nodes trx0_ad9084 … trx3_ad9084, each bound to one HSCI, one ADF4382A clock, with unique resets and JESD mesh connections; the following documents the **trx0_ad9084** fully and list the **differences** for trx1..trx3:

trx0_ad9084: ad9084@0

- **compatible**: adi, ad9084
- **reg**: <0>; **spi-max-frequency**: 5 MHz
- **label**: "axi-ad9084-rx0"
- **gpio-controller**; **#gpio-cells**: <2>
- **adi,gpio-exports**: <7 10 11 12 13 14> (6 exported GPIOs → used by DSA0)
- **adi,axi-hsci-connected**: <&axi_hsci_0>
- **Optional HSCI timing** (ifdef block commented by default)
- **adi,nyquist-zone**: <2>
- **adi,hsci-auto-linkup-mode-en**; **adi,adf4030-background-serial-alignment-en**
- **Lane mappings**
  - **adi, jtx0-logical-lane-mapping**: <11 0 11 11 11 11 11 1 11 11 11 11>
  - **adi, jtx1-logical-lane-mapping**: same as above
  - **adi, jrx0-physical-lane-mapping**: <11 7 0 0 0 0 0 0 0 0 0 0>
  - **adi, jrx1-physical-lane-mapping**: same as above
- **Serdes tuning**
  - **adi ,jtx-ser-amplitude**: <0>
  - **adi, jtx-ser-pre-emphasis**: <2>
  - **adi, jtx-ser-post-emphasis**: <1>
- **Clocks**
  - **clocks**: <&adf4382_0 0>; **clock-names**: "dev_clk"
  - **dev_clk-clock-scales**: <1 10>
  - **clock-output-names**: "rx0_sampl_clk","tx0_sampl_clk"; **#clock-cells**: <1>
- **JESD topology**
  - **jesd204-device**; **#jesd204-cells**: <2>; **jesd204-top-device**: <0> (TOP)
  - jesd204-ignore-errors
  - **jesd204-link-ids**: DEFRAMER_LINK_A0_TX, DEFRAMER_LINK_B0_TX, FRAMER_LINK_A0_RX, FRAMER_LINK_B0_RX
  - **jesd204-inputs**: chained to **&trx1_ad9084** (both framer and deframer), that is, multidevice link daisy-chain
- **Reset** and **Trigger**
  - **reset-gpios**: <&axi_gpio_2 0 0>
  - **trig-req-gpios**: <&axi_gpio 32 0>
  - adi, trigger-sync-en
- **Firmware profile**
  - **adi, device-profile-fw-name**: DEVICE_PROFILE_NAME (default "id01_uc42.bin")
  - **adi, multidevice-instance-count**: <4>
- **IIO channels**:
  - **io-channels**: <&adf4030 0>, <&adf4382_0 0>
  - **io-channel-names**: "bsync","clk"
- **Role**: first MxFE device, JESD “TOP device,” clocked from ADF4382_0, HSCI_0; provides sample clocks to FPGA transmit endpoint; exposes GPIO for DSA0

Differences for others follow:

- trx1_ad9084: axi-ad9084-rx1@1
  - **label**: "axi-ad9084-rx1"; **reg**: <1>
  - **adi,axi-hsci-connected**: &axi_hsci_1
  - **clocks**: <&adf4382_1 0>; **clock-output-names**: "rx1_sampl_clk","tx1_sampl_clk"
  - **reset-gpios**: <&axi_gpio_2 1 0>
  - **jesd204-inputs**: chained to **&trx2_ad9084** (next device)
  - **io-channels**: <&adf4030 1>, <&adf4382_1 0>
  - Role: second MxFE, midchain
- trx2_ad9084: axi-ad9084-rx2@2
  - **label**: "axi-ad9084-rx2"; **reg**: <2>
  - **adi,axi-hsci-connected**: &axi_hsci_2
  - **clocks**: <&adf4382_2 0>; **clock-output-names**: "rx2_sampl_clk","tx2_sampl_clk"
  - **reset-gpios**: <&axi_gpio_2 2 0>
  - **jesd204-ignore-errors** (set here explicitly)
  - **jesd204-inputs**: chained to **&trx3_ad9084**

INTERRUPT CONTROLLER AND ON-BOARD PERIPHERALS FROM THE REFERENCED NODES

- **io-channels**: <&adf4030 2>, <&adf4382_2 0>
- **Role**: third MxFE, midchain

- trx3_ad9084: axi-ad9084-rx3@3
  - **label**: "axi-ad9084-rx3"; **reg**: <3>
  - **adi, axi-hsci-connected**: &axi_hsci_3
  - **clocks**: <&adf4382_3 0>; **clock-output-names**: "rx3_sampl_clk","tx3_sampl_clk"
  - **reset-gpios**: <&axi_gpio_2 3 0>
  - **jesd204-inputs**: connects upstream to **FPGA cores** &axi_ad9084_core_rx and &axi_ad9084_core_tx rather than another MxFE (end of chain)
  - **io-channels**: <&adf4030 3>, <&adf4382_3 0>
  - **Role**: fourth MxFE, **chain terminus**, bridges to FPGA JESD cores

Other referenced nodes follow:

- **&axi_iic_main sub-devices** (previously covered under &fmc_i2c)
- **&axi_ethernet, &axi_intc, &axi_gpio_2** (which were previously detailed)

QUICK CROSSREFERENCE MAPS

The quick crossreference maps follow:

- Clock providers → consumers (selected highlights)
  - **clkin_j1** → **ltc6953** (root reference)
  - **ltc6953 outputs** → adf4382_[0..3], ltc6952 (REF IN), adf4351 (REF), SYSREF/BSYNC fanouts (J50/J51, EZS_SRQ, ADF4030)
  - **adf4351** → ltc6952 VCO in
  - **ltc6952 outputs** → FPGA gigabit clocks (axi_adxcvr_rx/tx conv), device clocks for JESD (axi_ad9084_rx_jesd/tx_jesd), adf4030 ref/SYSREF inputs
  - **adf4030 (SYSREF)** → SYSREFs to Apollo/FPGA (Channel 0 to Channel 4 and Channel 9), inputs from ADF6952/ADF6953 (Channel 6 and Channel 8)
  - **adf4382_[0..3]** → trx[0..3]_ad9084 DEV_CLK inputs
- JESD204 chain (end-to-end)
  - **MxFE 0** ⇄ **MxFE 1** ⇄ **MxFE 2** ⇄ **MxFE 3** ⇄ **FPGA JESD** (axi_ad9084_rx_jesd, axi_ad9084_tx_jesd)
  - **Gigatransfer transceivers** (axi_adxcvr_rx/axi_adxcvr_tx) sit between FPGA lane clocks and JESD cores; all synchronized via **LTC6952/LTC6953 + ADF4030 SYSREF**
- Control and monitoring
  - **GPIOs**: axi_gpio_2 → power-good inputs, mux selects, device resets, board enables (ADF4382A 5V)
  - **I^2C**: EEPROM + dual LTC2977 PMIC/monitors on FMC I^2C bus
  - **SPI**: Two controllers—clock tree and MxFE devices mapped to buses as previously described

An exhaustive documentation set is available for the VCU118 Quad-AD9084 RevB Linux device tree.

CHANNEL CALIBRATION USING THE ADXBAND16EBZ-CAL

OVERVIEW

The ADXBAND16EBZ employs multichip synchronized DACs and ADCs to enable phase-coherent RF signal generation across 16 channels. To achieve coherent combining (~24dB gain), per-channel phase alignment is required at every operating frequency. Calibration ensures deterministic phase alignment despite hardware variations.

Calibration of the ADXBAND16EBZ using the methods described herein requires the use of the ADXBAND16EBZ-CAL calibration board. This calibration board must be purchased separately and is not included with the ADXBAND16EBZ.

The ADXBAND16EBZ-CAL is a separately powered (12V) board with connectors for 16 ADC and 16 DAC channels. This calibration board mates directly to the ADXBAND16EBZ digitizing the card via RF connectors and the VCU118 PMOD interface. (Note that the cable is included with the ADXBAND16EBZ-CAL purchase).

The ADXBAND16EBZ-CAL provides both individual adjacent channel loopback and combined channel loopback options. This calibration board provides both a combined transmit channels out option and a combined receive channels in option through the Subminiature Version A (SMA) port.

PHASE DETERMINISM

The platform supports multichip synchronization, enabling deterministic power-up phase states, which allows a baseline lookup table (LUT) for repeatable start-up conditions. However, additional system-level RF calibration is still required to compensate for frequency-, amplitude-, and temperature-dependent effects.

CALIBRATION METHODS

The two following techniques are implemented in the ADXBAND16EBZ:

- The null power method
  - Rotates the phase of each channel relative to a reference.
  - Captures data and locates the minimum received amplitude (null).
  - Applies a 180° correction.
  - Is robust, but slow, and repeated for each frequency.
- The cross correlation method
  - Has 16 DACs each transmit one pulse, forming an 8192-sample sequence.
  - Combined waveform loops back to ADCs
  - 10 captures are taken at arbitrary times.
  - Cross-correlation is applied to time-align captures and coherently sums them.
  - Has a second correlation that aligns the summed waveform to the transmitted reference.
  - Phase of highest-correlation samples yields per-channel ADC and DAC offsets.
  - Is significantly faster than the null power method.

CROSS-CORRELATION FLOW

The cross-correlation flow follows:

1. Transmit interleaved pulses at RF.
2. Captures 10 arbitrary ADC data sets.
3. Cross-correlates to identify alignment points.
4. Time-align captures using circular shifts.
5. Sum captures coherently.
6. Cross-correlates with a reference waveform.
7. Extracts per-channel phase offsets.
8. Stores resulting calibration matrix.

FINAL CALIBRATION OUTPUT

After applying the derived offsets, the following results:

- All DAC and ADC channels align to sample zero of the reference.
- Phase alignment is consistent across all channels.
- Amplitude normalization can be applied, if required.

RF CALIBRATION METHODS

The RF calibration methods follow:

- Calibrate to take advantage of the coherent combining gain as follows:

  $20 \times \log(16) = \sim 24\text{dB}$ *Coherent Combining Gain*
- The two main methods of calibration to achieve phase alignment are already implemented within the system class.

NULL POWER METHOD

The null power method steps follow:

1. Rotate the phase, one channel at a time, relative to a reference channel and capture data.
2. Locate the lowest magnitude (null).
3. Apply a 180° phase shift
4. Store the data.
5. Repeat for every frequency.

The null power method is robust; however, it takes a significant amount of time.

CHANNEL CALIBRATION USING THE ADXBAND16EBZ-CAL

CROSS-CORRELATION

Cross-correlation include the following:

1. 16 DACs transmit 1 pulse each for a 8192 sample size. This sample size is combined and looped back to the ADCs.
2. Cross-correlate the transmitted and received data to determine pulse one of 16.
3. Align pulse one to sample one.
4. Extract phase data and repeat for every frequency.

Extremely fast method when compared to the null power method.

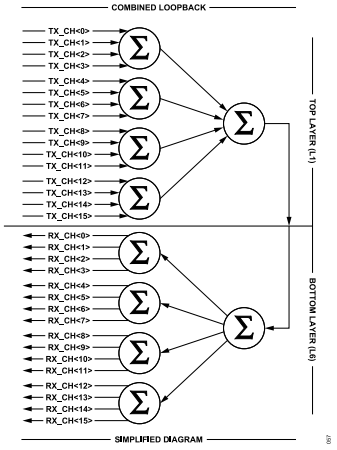


Figure 21. Cross-Correlation Simplified Diagram

CROSS-CORRELATION METHOD

The cross-correlation method includes the following:

- Time interleave DACs with pulsed baseband waveform and transmits at operating frequency. It includes 16 DACs in total and 16 pulses total (8192 × 16 waveform).
- 10 data captures, *taken at arbitrary points in time*, with the calibration board in combined loopback mode with numerically-controlled oscillators (NCOs) at zero phase. The ADCs see 16 pulses *each* (8192 × 16 × 10).
- Using a cross-correlation function, the 10 data captures can be time aligned and coherently summed. Note that the ADC code increase as each capture is aligned and summed (8192 × 16).
- Points of high correlation are identified for each data capture relative to an arbitrary reference data capture, and a circular sample shift is applied to the waveform to time align to the selected data capture.
- Perform another cross-correlation between the transmitted pulsed waveform and the coherently summed data capture for all 16 ADCs (8192 × 16).
- Samples with the highest correlation and their corresponding phase are determined for each channel relative to the reference waveform and are stored.
- The coherently summed data is shifted to sample zero of the reference waveform.
- The phase extracted at this point is used to calculate the ADC phase offset for calibration (16 × 16).

$$\theta_{CHANNEL\ REF} - \theta_{CHANNEL\ x} = \theta_{OFFSET\ CHANNEL\ x}$$

- Apply the known pulse separation from the transmitted waveform to the newly aligned ADC data. If alignment is correct, the sample values should fall onto each pulse.
- The phase values at the points can be used to calculate the phase offsets for all 16 DACs.

$$\theta_{PULSE\ REF} - \theta_{PULSE\ x} = \theta_{OFFSET\ CHANNEL\ x}$$

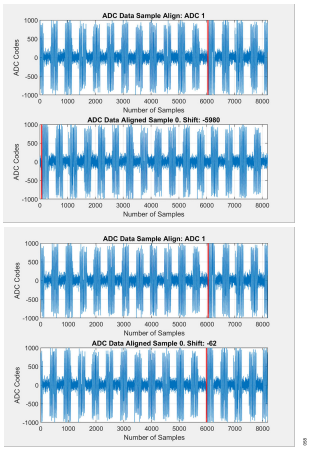


Figure 22. Example Plot of Calibration Using ADC Sample Shifting

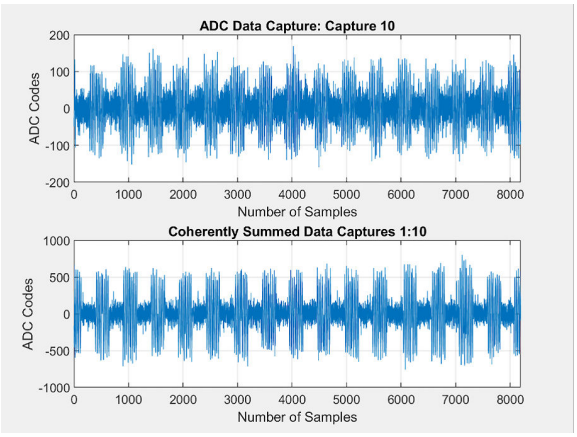


Figure 23. Coherent Summation of Multiple Captures of Data

CHANNEL CALIBRATION USING THE ADXBAND16EBZ-CAL

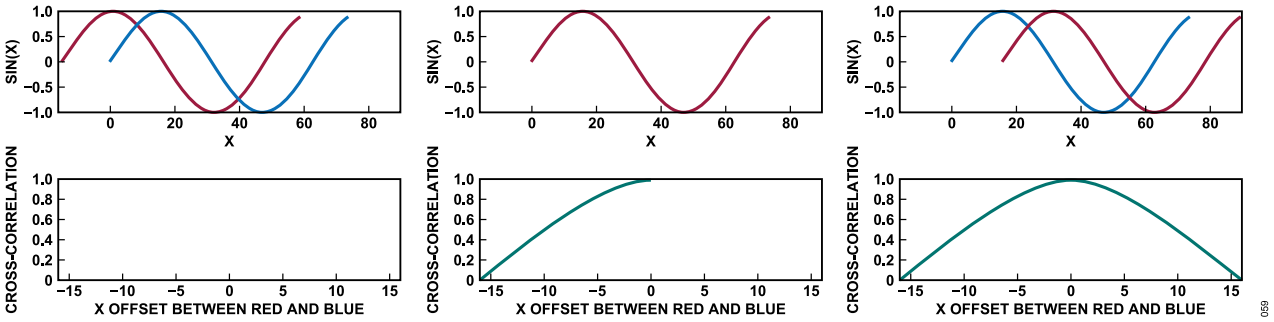


Figure 24. Example Plot Showing Cross-Correlation Using an Offset Between Two Channels

CHANNEL CALIBRATION USING THE ADXBAND16EBZ-CAL

VISUAL DEPICTION OF CROSS-CORRELATION CALIBRATION

Figure 25 through Figure 28 show the cross-correlation calibration plots.

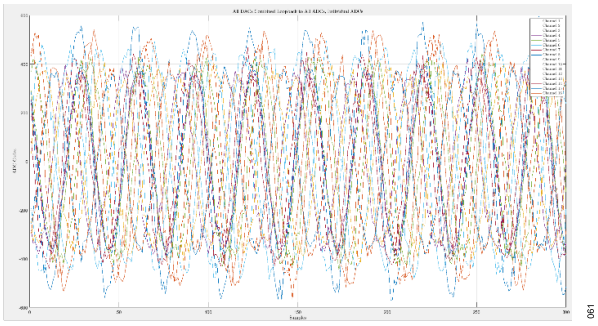


Figure 25. Apply XCORR-Based systemCal

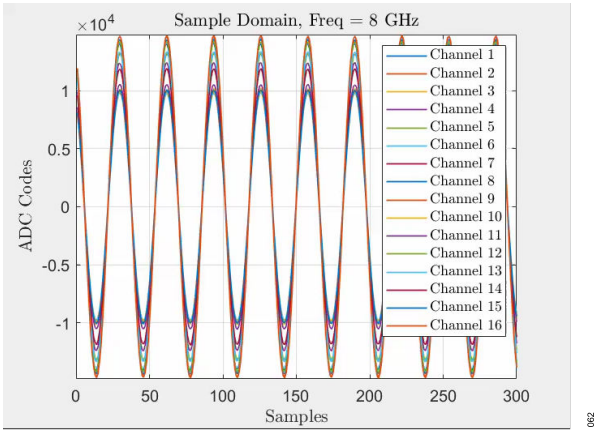


Figure 26. Normalize Amplitudes to Lowest Amplitude Waveform Using a Scaling Factor

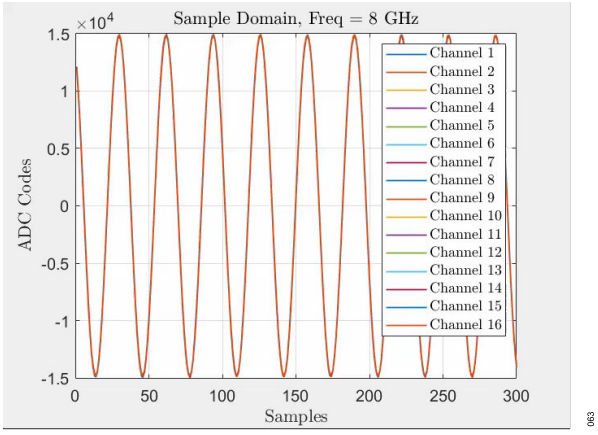


Figure 27. Sample Data Aligned Across Channels

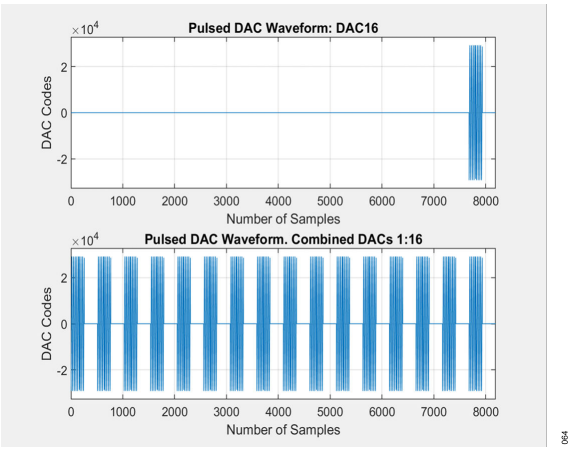


Figure 28. Pulsed DAC Waveform Combined Data

MULTICHIP SYNCHRONIZATION (MCS) THEORY

OVERVIEW OF THE MCS

MCS enables multiple AD9084/AD9088 devices to achieve ±10ps alignment accuracy, which is critical for phased array radar, electronic warfare, and other applications that require precise timing across multiple data converters.

The MCS system consists of three main components:

- AD9084/AD9088 (Apollo): the mixed-signal front-end with integrated ADCs and DACs
- ADF4030: a precision SYSREF/BSYNC generator with a TDC for path delay measurement
- ADF4382A: a low-noise PLL that provides the sampling clock with automatic phase alignment capabilities

SYSTEM ARCHITECTURE

Figure 29 shows the hardware topology of the system architecture.

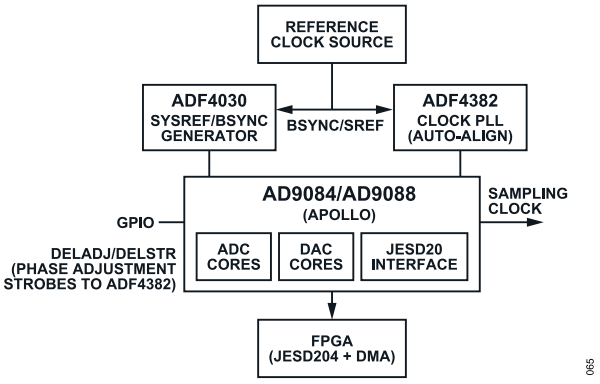


Figure 29. Hardware Topology

MCS DEFINITIONS

The following details the typical MCS definitions used:

- BSYNC is synonymous with SYSREF, which is a bidirectional alignment signal.

  1 MHz to 100s of MHz; note that, slower is better for more alignment margins, and faster is better for shorter convergence.

- TRIG is an alignment to specific BSYNC and aligns JESD204 and NCO across multiple Apollo devices.

  It is a one-shot event that must meet the setup (SU) and hold (HLD) to a specific BSYNC. It is an externally applied signal captured relative to the AD9084 internal SYSREF and the FPGA reference clock for system synchronization and high-speed event timing.

- The AD9084 is the Apollo MxFE.
- The ADF4382A is the Apollo MxFE clock.
- The ADF4030 is a 10 channel precision synchronizer.
- An external SYSREF/BSYNC that is a board-level SYSREF used to align internal timing. This signal is applied to AD9084.
- An internal SYSREF/BSYNC that is an internal clock divided down from the sample clock and used as a synchronization timing reference for AD9084 device synchronization. This signal can be aligned to an external SYSREF for Subclass 1 operation.
- The TDC is used for SYSREF averaging of the two signals to accurately measure the phase delta.

Note that the MCS on the ADXBAND16EBZ aligns ADC and DAC samples across all channels and can also align multiple ADXBAND16EBZ boards. In addition, the NCO and sampling frames can be aligned to the same phase.

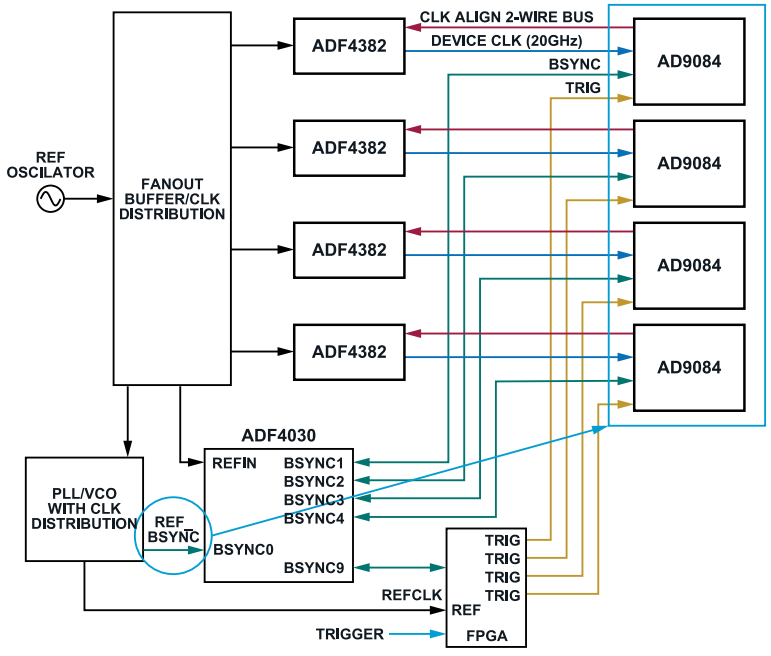


Figure 30. MCS Alignment on ADXBAND16EBZ , Which Aligns REF_BSYNC Across All AD9084 Devices, Independent of Spatial Separation

MULTICHIP SYNCHRONIZATION (MCS) THEORY

A brief description of the four MCS steps follows that automatically complete in the IIO code:

1. Aligning of REF_BSYNC to BSYNCx within the ADF4030.
2. Aligning of an internal to external BSYNC within the AD9084.
3. Nulling the external BSYNC path delay deltas.

   Repeat Step 2 because the external BSYNC has been changed.
4. Sending TRIG to meet the SU/HLD to BSYNC.

MULTICHIP SYNCHRONIZATION (MCS) THEORY

DETAILED DISCUSSION OF THE MCS STEPS

The following goes into further details on the MCS steps:

1. Aligning REF_BSYNC to BSYNCx within the ADF4030, which includes the following (see Figure 31):
   - The ADF4030 TDC measures time skew between BSYNCx and REF_BSYNC.
   - The ADF4030 adjusts the BSYNCx programmable delay (BSYNCx alignment < ±5ps).
   - Repeats the procedure for all BSYNCx.
   - The ADF4030 transmits aligned BSYNCx to the AD9084 devices. (Note that BSYNCx to TDC all have equal delays).

     Note that REF_BSYNC is now aligned <5ps on all ADF4030 outputs.

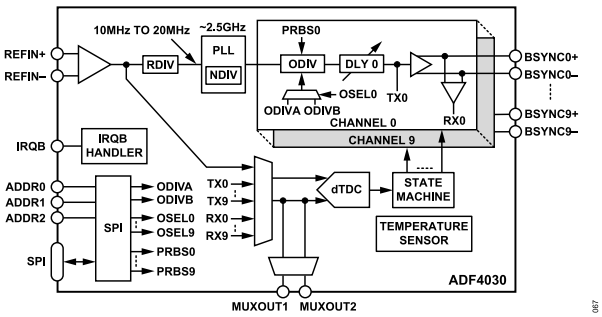


Figure 31. ADF4030 Block Diagram

2. Aligning an internal to external BSYNC within the AD9084, which includes the following:
   - The ADF4030 transmits BSYNC to Apollo.
   - The AD9084 internal BSYNC is based on the CLK/X device.
   - The internal and external BSYNC have the same frequency.
   - The AD9084 TDC measures Δt1 (see Figure 32).

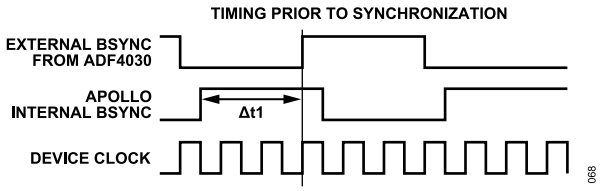


Figure 32. Step 2—TDC Measurement

   - The controller aligns an internal BSYNC to an external BSYNC within half the device clock using the digital cycle delays.
   - The alignment resolution is ±25ps for a 20GHz device clock.
   - The internal BSYNC coarse aligns to the external BSYNC (see Figure 33).

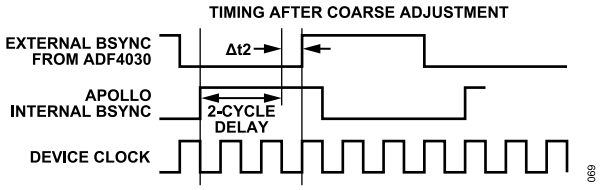


Figure 33. Step 2—Coarse Alignment

   - The AD9084 TDC measures the residual Δt1 or Δt2 .
   - The BSYNCs are fine aligned using delay adjust (the ADF4382A, see Figure 34).

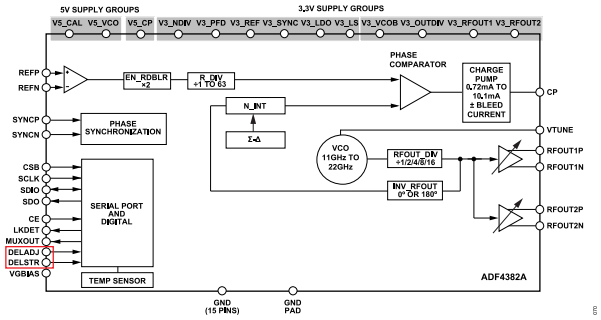


Figure 34. ADF4382A Block Diagram

   - The AD9084 → ADF4382A fine delay adjusts the 2-wire bus (see Figure 35).

     The delay device clock → delays to the internal BSYNC.

     Note that the internal and external BSYNC are now aligned within ~1ps.

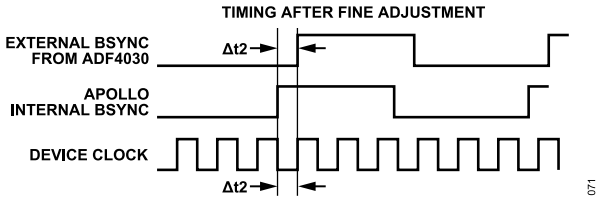


Figure 35. Step 2—Fine Alignment

3. Nulling the external BSYNC path delay deltas; aligns an external BSYNC to all AD9084 as follows:
   - The AD9084 and ADF4030 each change BSYNC direction.
   - The AD9084 sends a copy of its internal BSYNC (from Step 2) back to the ADF4030 on the same trace.
   - The ADF4030 receives BSYNCx.
   - The ADF4030 TDC measures skew: REF_BSYNC and BSYNCx.
   - The controller measures this time delta for each BSYNCx.
   - The controller writes −TRT/2 into every ADF4030 TDC.

     Note that the AD9084 locks onto an offset equal to −TRT/2.
   - The ADF4030 changes BSYNCs back to the outputs and to the inputs of the AD9084
   - The AD9084 internal BSYNC is now aligned to the external BSYNC using the −TRT/2 offset.

MULTICHIP SYNCHRONIZATION (MCS) THEORY

- All AD9084/AD9088 devices are aligned as follows:
  - An internal BSYNC is aligned to the ADF4030 REF_BSYNC
  - The AD9084 device clocks are aligned between themselves
- Repeat Step 2 because the external BSYNC has been changed.

4. Send TRIG from the FPGA to the AD9084 (must meet SU and HLD to BSYNC) as follows:
   - TRIG signals trigger AD9084 events, such as the following:
     - Aligns the JESD204C SERDES as a SYSREF.
     - Resets the NCO, fractional sample rate converter (FSRC), and dynamic reconfiguration.

     Note that aligning TRIG to BSYNC is a setup and hold timing issue for tens of MHz.

     The system controller sends TRIGGER (one pulse) to the host controller that is aligned to REF_BSYNC.

     The host controller generates the TRIG signals aligned to TRIGGER.
   - All BSYNCs and TRIGs are aligned.

     Unidirectional BSYNC connections between the ADF4030 and host controller are used to calculate and then correct how much the TRIG outputs drift from the REF_BSYNC.
   - The AD9084 has a feedback monitor to guide the alignment of the TRIG signal to ensure reliable capture.

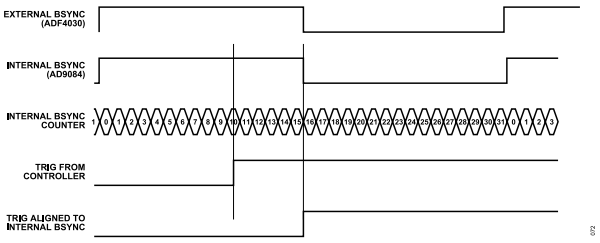


Figure 36. TRIG Timing Alignment to BSYNC

BSYNC CONNECTIONS

The external BSYNC signal is connected through the U.FL connectors on the bottom side of the ADXBAND16EBZ. J46 is the BSYNC_P, and J47 is the BSYNC_N. Note that a differential signal can be create to an ADF4030 evaluation board or through a balun to convert to a single-ended signal.

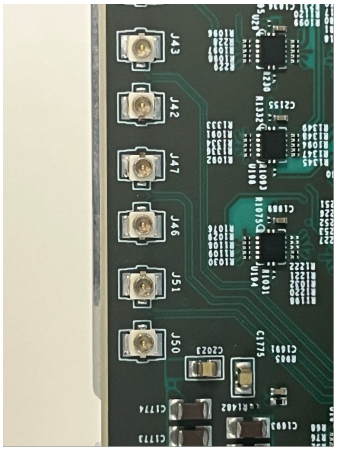


Figure 37. Bottom Side of ADXBAND16EBZ —BSYNC on U.FL J46 and J47

MULTICHIP SYNCHRONIZATION (MCS) THEORY

EXAMPLE IIO COMMANDS FOR MCS

The example IIO commands for MCS follow:

```
iio_attr -c adf4030 J46_J47_UFL output_enable 0 #This is the off-board BSYNC_REF from exter▶
nal ADF4030, so it must be an Rx
iio_attr -c adf4030 APOLLO_SYSREF_0 output_enable 1 #For Tx to Apollo 0
iio_attr -c adf4030 APOLLO_SYSREF_1 output_enable 1 #For Tx to Apollo 1
iio_attr -c adf4030 APOLLO_SYSREF_2 output_enable 1 #For Tx to Apollo 2
iio_attr -c adf4030 APOLLO_SYSREF_3 output_enable 1 #For Tx to Apollo 3
iio_attr -c adf4030 FPGA_SYSREF_0 output_enable 1 #For Tx to FPGA
iio_attr -c adf4030 APOLLO_SYSREF_3 reference_channel 9 #Use Channel 9 (J46_J47_UFL) as reference_chan▶
nel
iio_attr -c adf4030 APOLLO_SYSREF_2 reference_channel 9 #Use Channel 9 (J46_J47_UFL) as reference_chan▶
nel
iio_attr -c adf4030 APOLLO_SYSREF_1 reference_channel 9 #Use Channel 9 (J46_J47_UFL) as reference_chan▶
nel
iio_attr -c adf4030 APOLLO_SYSREF_0 reference_channel 9 #Use Channel 9 (J46_J47_UFL) as reference_chan▶
nel
iio_attr -c adf4030 FPGA_SYSREF_0 reference_channel 9 #Use Channel 9 (J46_J47_UFL) as reference_channel
##Off-Board ADF4030 makes BSYNC measurements vs. REF_BSYNC and adjusts phase
iio_attr -D axi-ad9084-rx-hpc trig_sync 1 #Trigger the main Apollo
iio_attr -D axi-ad9084-rx2 trig_sync 1 #Trigger Apollo 2
iio_attr -D axi-ad9084-rx3 trig_sync 1 #Trigger Apollo 3
iio_attr -D axi-ad9084-rx1 trig_sync 1 #Trigger Apollo 1
axi_adf4030_trig #applies external TRIG to VCU118 boards to align NCO
```

MULTICHIP SYNCHRONIZATION (MCS) THEORY

MULTIDEVICE CONFIGURATION

In a multidevice configuration, each Apollo device has its own dedicated ADF4382A and receives BSYNC from the ADF4030 through separate output channels. Each Apollo controls its ADF4382A via GPIO-driven DELADJ and DELSTR signals for phase correction (see Figure 38).

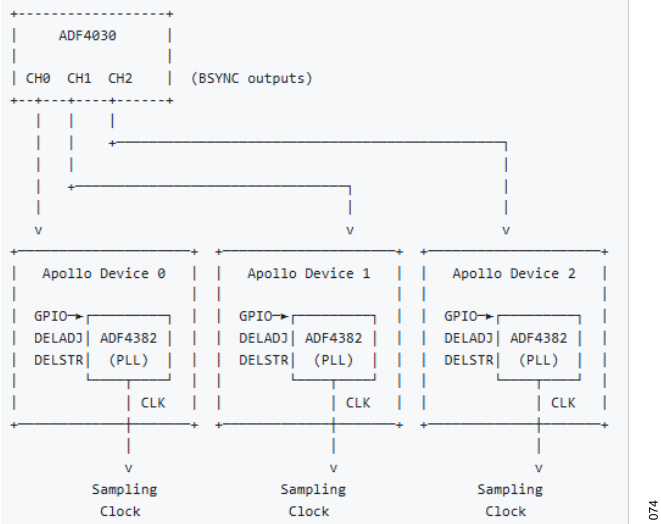


Figure 38. Multidevice Configuration

ADF4382A PHASE CONTROL VIA THE AD9084 GPIO

Each Apollo device controls its own dedicated ADF4382A's phase adjustment through the following two GPIO-driven strobe signals:

- DELADJ sets the direction and magnitude of the phase correction. Note that the Apollo firmware writes the bleed current values (polarity, coarse, and fine) and pulses this strobe.
- DELSTR triggers the ADF4382A to apply the phase adjustment. Note that each strobe pulse causes the PLL to adjust its output phase.

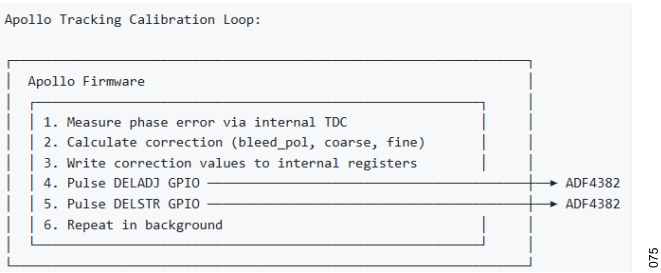


Figure 39. Apollo Tracking Calibration Loop

This architecture allows each Apollo to independently maintain phase alignment with its clock source, compensating for temperature drift and other environmental factors without software intervention.

LINUX DRIVER ARCHITECTURE

See Figure 40 for the Linux driver architecture driver stack.

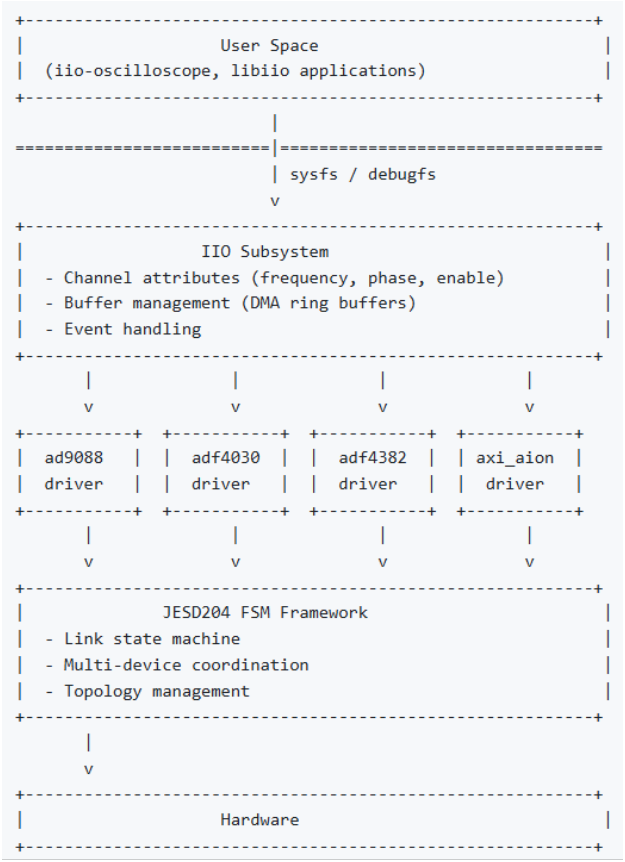


Figure 40. Driver Stack

MULTICHIP SYNCHRONIZATION (MCS) THEORY

DRIVER FILE ORGANIZATION

AD9088 (Apollo) Driver

The following code details the Apollo driver contents:

```
drivers/iio/adc/apollo/
├── ad9088.c                    # Main driver, IIO interface
├── ad9088.h                    # Driver header, structures
├── ad9088_dt.c                 # Device tree parsing
├── ad9088_jesd204_fsm.c        # JESD204 state machine callbacks
├── ad9088_mcs.c                # MCS calibration functions
├── ad9088_debugfs.c            # Debugfs interface
├── ad9088_cal.c                # Calibration save/restore
└── public/                     # Apollo API (vendor library)
```

Related Drivers

Related drivers to the Apollo drivers include:

```
drivers/iio/frequency/
├── adf4030.c             # ADF4030 SYSREF generator
└── adf4382.c             # ADF4382 clock PLL
```

IIO CHANNEL RELATIONSHIPS

The AD9088 driver connects to the ADF4030 and ADF4382A via the IIO consumer channels defined in the device tree shown in Figure 41.

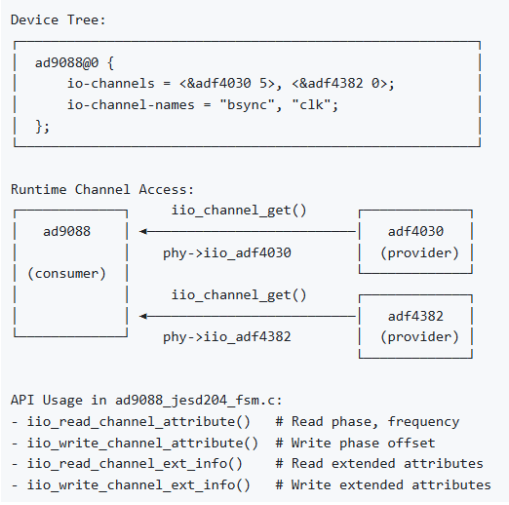


Figure 41. AD9088, ADF4030, and ADF4382A IIO Consumer Channels Device Tree

MULTICHIP SYNCHRONIZATION (MCS) THEORY

JESD204 STATE MACHINE INTEGRATION

The MCS calibration is integrated into the JESD204 finite state machine (FSM) framework, which coordinates initialization across all devices in the JESD204 link. See Figure 42 and Figure 43 for additional information.

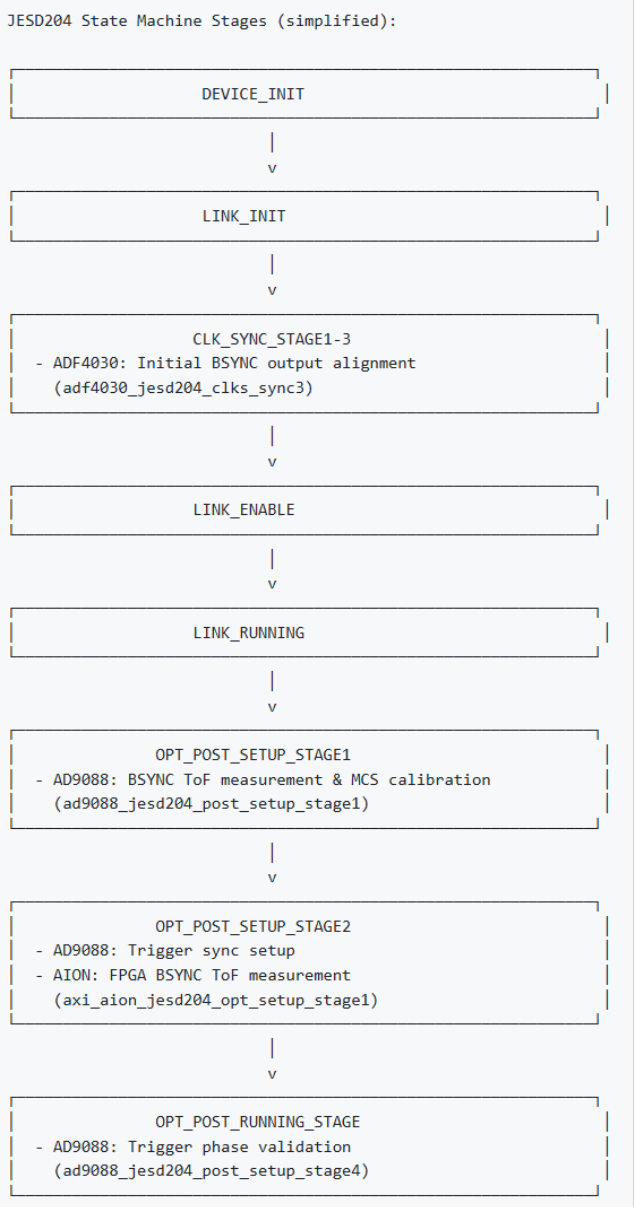


Figure 42. FSM Stage Execution Order

```
static const struct jesd204_dev_data ad9088_jesd204_data = {
    .state_ops = {
        [JESD204_OP_OPT_POST_SETUP_STAGE1] = {
            .per_device = ad9088_jesd204_post_setup_stage1,  // MCS ToF + Init Cal
        },
        [JESD204_OP_OPT_POST_SETUP_STAGE2] = {
            .per_device = ad9088_jesd204_post_setup_stage2,  // Trigger setup
        },
        [JESD204_OP_OPT_POST_SETUP_STAGE3] = {
            .per_device = ad9088_jesd204_post_setup_stage3,  // Trigger pulse
        },
        [JESD204_OP_OPT_POST_RUNNING_STAGE] = {
            .per_device = ad9088_jesd204_post_setup_stage4,  // Phase validation
        },
    },
};
```

079

Figure 43. AD9084 JESD204 Callbacks

BSYNC TIME OF FLIGHT (TOF) MEASUREMENT

The ToF measurement determines the propagation delay between the ADF4030 and Apollo, allowing precise phase compensation.

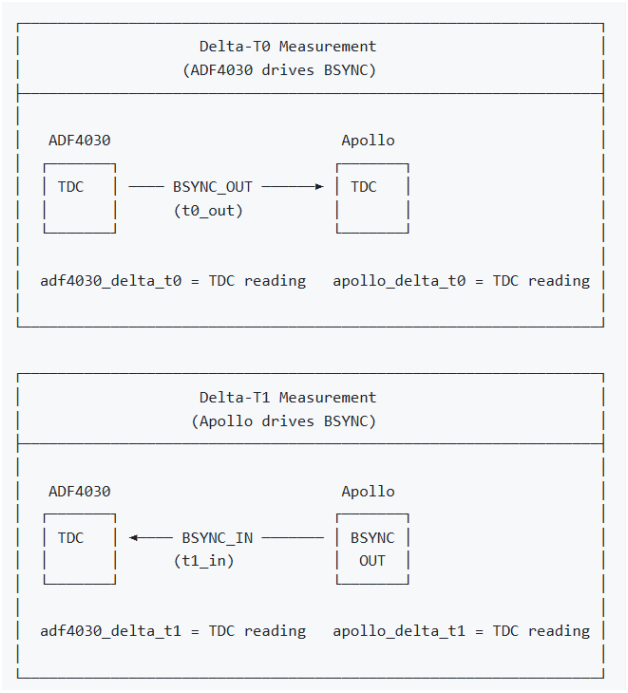


Figure 44. Measurement Principle

MULTICHIP SYNCHRONIZATION (MCS) THEORY

CALCULATION OF THE PATH DELAY

Calculate the path delay by using the four TDC measurements shown in Figure 45.

```
// From ad9088_jesd204_fsm.c
calc_delay = (adf4030_delta_t0 - adf4030_delta_t1)
           - (apollo_delta_t1 - apollo_delta_t0);

// Handle wraparound with modulo arithmetic
round_trip_delay = (calc_delay + bsync_period) % bsync_period;

// One-way path delay is half the round trip
path_delay = round_trip_delay / 2;

// Apply negative offset to compensate
iio_write_channel_attribute(adf4030, -path_delay, IIO_CHAN_INFO_PHASE);
```

081

Figure 45. Calculate the Path Delay With These Four TDC Measurements

MATHEMATICAL DERIVATION

Figure 46 shows the mathematical derivation.

```
Let:
  T_fwd = Forward path delay (ADF4030 → Apollo)
  T_rev = Reverse path delay (Apollo → ADF4030)

Delta-T0 measurement (ADF4030 transmits):
  adf4030_delta_t0 = reference time at ADF4030
  apollo_delta_t0  = adf4030_delta_t0 + T_fwd

Delta-T1 measurement (Apollo transmits):
  apollo_delta_t1  = reference time at Apollo
  adf4030_delta_t1 = apollo_delta_t1 + T_rev

Solving for round-trip delay:
  (adf4030_delta_t0 - adf4030_delta_t1) = forward contribution
  (apollo_delta_t1 - apollo_delta_t0)   = reverse contribution

  calc_delay = T_fwd + T_rev = round_trip_delay
  path_delay = round_trip_delay / 2  (assuming symmetric paths)
```

082

Figure 46. Mathematical Derivation

MULTICHIP SYNCHRONIZATION (MCS) THEORY

COMPLETE MCS FLOW SEQUENCE

The complete MCS flow executes in ad9088_jesd204_post_set-up_stage1() (see Figure 47).

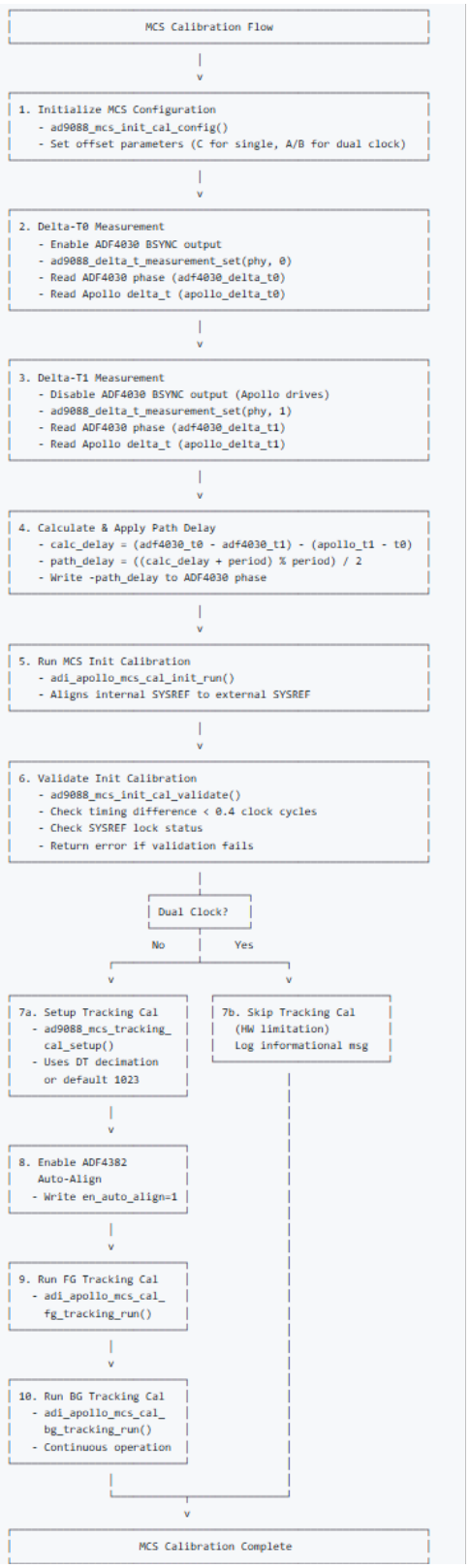


Figure 47. MCS Calibration Flow

MULTICHIP SYNCHRONIZATION (MCS) THEORY

TRACKING CALIBRATION

Tracking calibration maintains synchronization over time as temperature and other environmental factors cause phase drift.

ADF4382A BLEED CURRENT DAC FOR PHASE ADJUSTMENT

The ADF4382A PLL uses a bleed current DAC to make fine phase adjustments. Each Apollo controls its dedicated ADF4382A via GPIO-driven DELADJ and DELSTR strobe signals (see Figure 48).

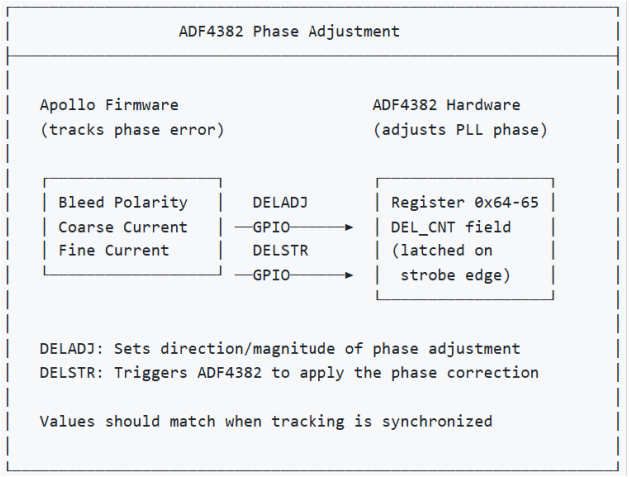


Figure 48. ADF4382A Phase Adjustment

TRACKING CALIBRATION VALIDATION

The AD9088_mcs_tracking_cal_validate debugfs attribute compares Apollo firmware values with the hardware values of the ADF4382A (see Figure 49).

```c
// From ad9088_mcs.c
int ad9088_mcs_tracking_cal_validate(struct ad9088_phy *phy, ...)
{
    // Freeze tracking to get stable reading
    adi_apollo_mcs_cal_bg_tracking_freeze(device);

    // Read Apollo firmware values
    adi_apollo_mcs_cal_tracking_status_get(device, &tracking_cal_status);
    fw_bleed_pol = tracking_cal_status.data.adf4382_status->Bleed_Pol;
    fw_coarse = tracking_cal_status.data.adf4382_status->Coarse;
    fw_fine = tracking_cal_status.data.adf4382_status->Fine;

    // Read ADF4382 hardware values via IIO
    iio_read_channel_ext_info(adf4382, "bleed_pol", &hw_bleed_pol);
    iio_read_channel_ext_info(adf4382, "coarse_current", &hw_coarse);
    iio_read_channel_ext_info(adf4382, "fine_current", &hw_fine);

    // Compare and report
    if (fw_bleed_pol == hw_bleed_pol &&
        fw_coarse == hw_coarse &&
        fw_fine == hw_fine) {
        status = "SYNCHRONIZED";
    } else {
        status = "MISMATCH - check hardware";
    }

    adi_apollo_mcs_cal_bg_tracking_unfreeze(device);
}
```

085

Figure 49. AD9088_mcs_tracking_cal_validate

SINGLE CLOCK MODE

The ADXBAND16EBZ only supports a single input clock mode configuration to the digitizer (see Figure 50).

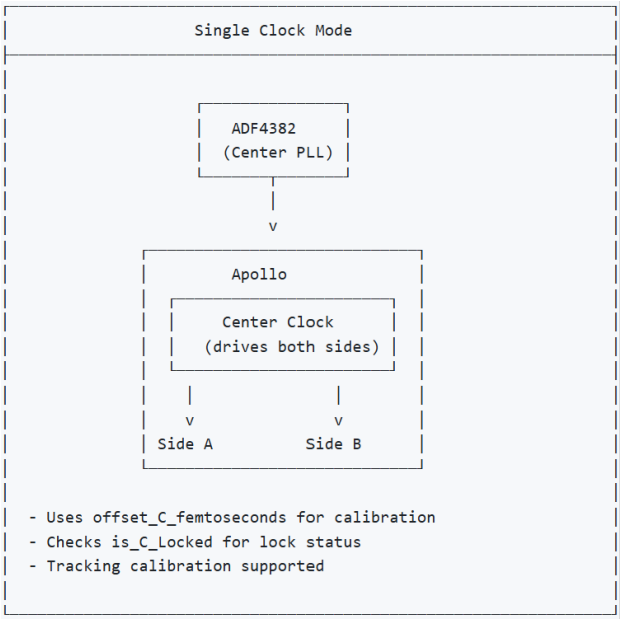


Figure 50. Single Clock Mode

MULTICHIP SYNCHRONIZATION (MCS) THEORY

DEVICE TREE CONFIGURATION

See Figure 51 and Table 9 for the complete MCS device tree configuration.

```
/* ADF4030 - SYSREF/BSYNC Generator */
adf4030: adf4030@0 {
    compatible = "adi,adf4030";
    reg = <0>;
    spi-max-frequency = <1000000>;

    /* Channel configuration */
    #io-channel-cells = <1>;

    /* BSYNC output channels */
    adi,channel@0 {
        reg = <0>;
        adi,output-enable;
        adi,auto-align-on-sync-en;
    };
    adi,channel@5 {
        reg = <5>;
        adi,output-enable;
        adi,auto-align-on-sync-en;
        adi,reference-channel = <0>;  /* Align to channel 0 */
    };
};

/* ADF4382 - Clock PLL */
adf4382: adf4382@1 {
    compatible = "adi,adf4382";
    reg = <1>;
    spi-max-frequency = <1000000>;

    #io-channel-cells = <1>;

    clocks = <&ref_clk>;
    clock-names = "ref_clk";

    adi,ref-frequency-hz = <125000000>;
    adi,output-frequency-hz = <12000000000>;
};

/* AD9088 - Apollo Device */
trx0_ad9084: ad9088@0 {
    compatible = "adi,ad9088";
    reg = <0>;
    spi-max-frequency = <1000000>;

    /* MCS IIO channel connections */
    io-channels = <&adf4030 5>, <&adf4382 0>;
    io-channel-names = "bsync", "clk";

    /* MCS Configuration */
    adi,mcs-track-decimation = /bits/ 16 <1023>;
    adi,trigger-sync-en;

    /* Profile and calibration */
    adi,device-profile-fw-name = "ad9088_profile.bin";
    adi,device-calibration-data-name = "ad9088_cal.bin";

    /* ... other properties ... */
};
```

087

Figure 51. Complete Example

Table 9. Key Device Tree Properties for MCS

Property	Type	Description
io-channels	phandle array	References to the ADF4030 (bsync) and ADF4382A (CLK) channels
io-channel-names	string array	Must be "bsync", "clk"
adi,mcs-track-decimation	u16	TDC decimation rate (1 to 32,767); default: 1023
adi,trigger-sync-en	bool	Enables trigger synchronization

MULTICHIP SYNCHRONIZATION (MCS) THEORY

DEBUG INTERFACE

MCS Attributes

The following code shows the MCS attributes:

```
/sys/kernel/debug/iio/iio:deviceX/
├── mcs_init                  # Initialize MCS
config (write 1)
├── mcs_dt0_measurement       # Delta-T0 meas▶
urement (write 1, read value)
├── mcs_dt1_measurement       # Delta-T1 meas▶
urement (write 1, read value)
├── mcs_dt_restore            # Restore delta-
T state (write 1)
├── mcs_cal_run               # Run init cali▶
bration (write 1, read status)
├── mcs_init_cal_status       # Read init cal
detailed status
├── mcs_track_cal_setup       # Setup tracking
cal (write 1)
├── mcs_fg_track_cal_run      # Run foreground
tracking (write 1)
├── mcs_bg_track_cal_run      # Control back▶
ground tracking (write 1/0)
├── mcs_bg_track_cal_freeze   # Freeze/un▶
freeze tracking (write 1/0)
├── mcs_track_status          # Read tracking
cal status
└── mcs_track_cal_validate    # Validate track▶
ing synchronicity (read)
```

```
# Check MCS init calibration status
cat /sys/kernel/debug/iio/iio:device0/mcs_init_cal_status

# Validate tracking calibration is synchronized
cat /sys/kernel/debug/iio/iio:device0/mcs_track_cal_validate

# Temporarily freeze background tracking
echo 1 > /sys/kernel/debug/iio/iio:device0/mcs_bg_track_cal_freeze
# ... perform measurements ...
echo 0 > /sys/kernel/debug/iio/iio:device0/mcs_bg_track_cal_freeze
```

088

Figure 52. MCS Example Usage

ERROR HANDLING AND DIAGNOSTICS

Driver Validation Checks

The driver performs the following validation checks during MCS calibration:

1. MCS initialization calibration failure, which includes the following:
   - Condition: SYSREF is not locked or timing difference > 0.4 cycles.
   - Action: returns error and stops JESD204 FSM.
   - Log: MCS Initcal Status: Failed.
2. Trigger phase margin, which includes the following:
   - Condition: phase outside 25% to 75% of the SYSREF period.
   - Action: warning logged and continues operation.
   - Log: triggers Phase X outside of the safe margin [Y, Z].
3. Path delay sanity, which includes the following:
   - Condition: calc_delay > the 2× period or path_delay at the extremes.
   - Action: warning logged and continues operation.
   - Log: the frequency sample (f_S) of Path Delay x is small.
4. Tracking calibration mismatch, which includes the following:
   - Condition: Apollo FW values != ADF4382A hardware values.
   - Action: reported via mcs_track_cal_validate.
   - Status: MISMATCH—check hardware.

MULTICHIP SYNCHRONIZATION (MCS) THEORY

KERNEL LOG MESSAGES

For normal operation, the kernal log messages follow:

- ad9088 spi0.0: total BSYNC path delay 12345678 f_S
- ad9088 spi0.0: MCS Initcal Status: Passed
- ad9088 spi0.0: triggers Phase 64 (Ideal 64) period 500000 f_S

The warning conditions for the kernal log messages follow:

- ad9088 spi0.0: calc_delay − 500000000 f_S exceeds 2× the BSYNC period
- ad9088 spi0.0: triggers Phase 12 outside of the safe margin [16, 48]
- ad9088 spi0.0: path delay of 1000 f_S is small (<5% of the period)

Dual-ADXBAND16EBZ MCS

See Figure 53 to Figure 55 for the MCS sync simplified block diagram, full block diagram, and the PMOD and scope connections.

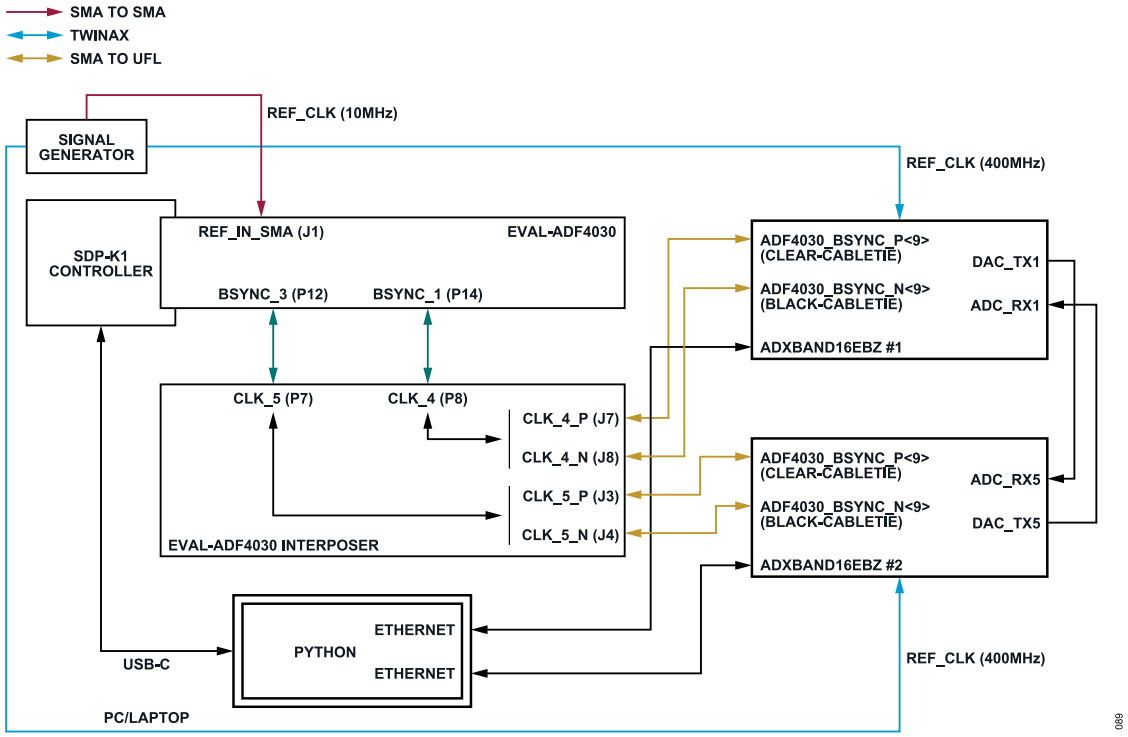


Figure 53. Dual-ADXBAND16EBZ—MCS Sync Simplified Block Diagram

MULTICHIP SYNCHRONIZATION (MCS) THEORY

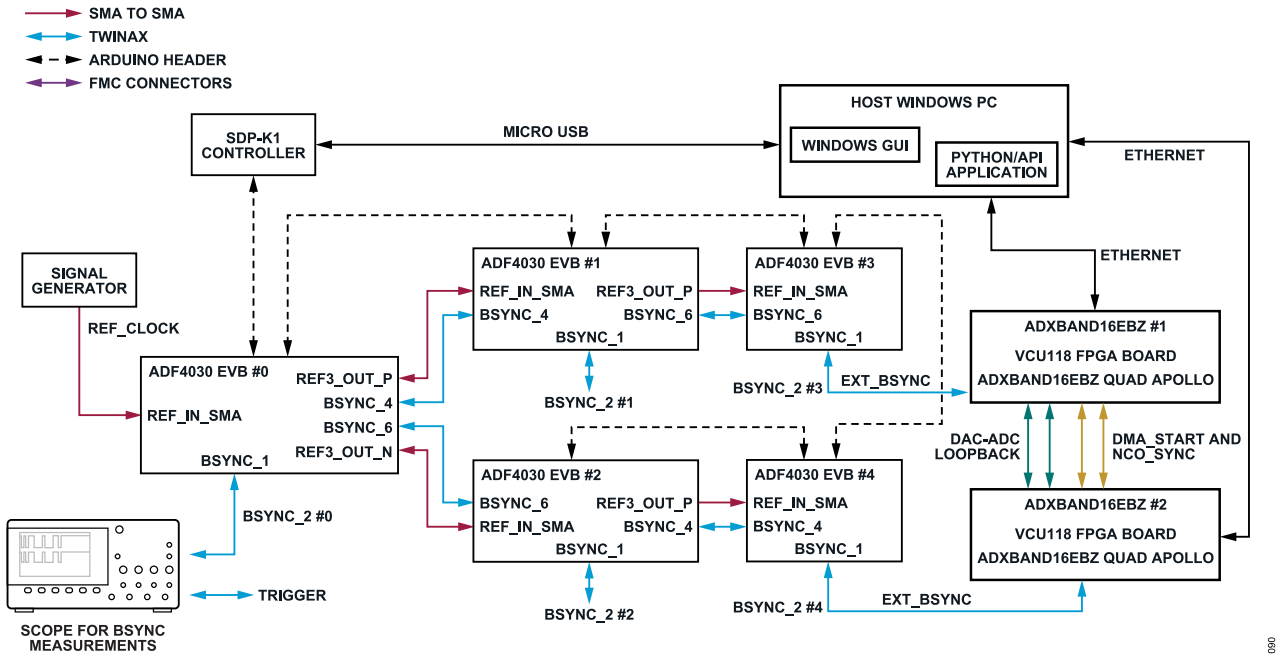


Figure 54. Full Block Diagram

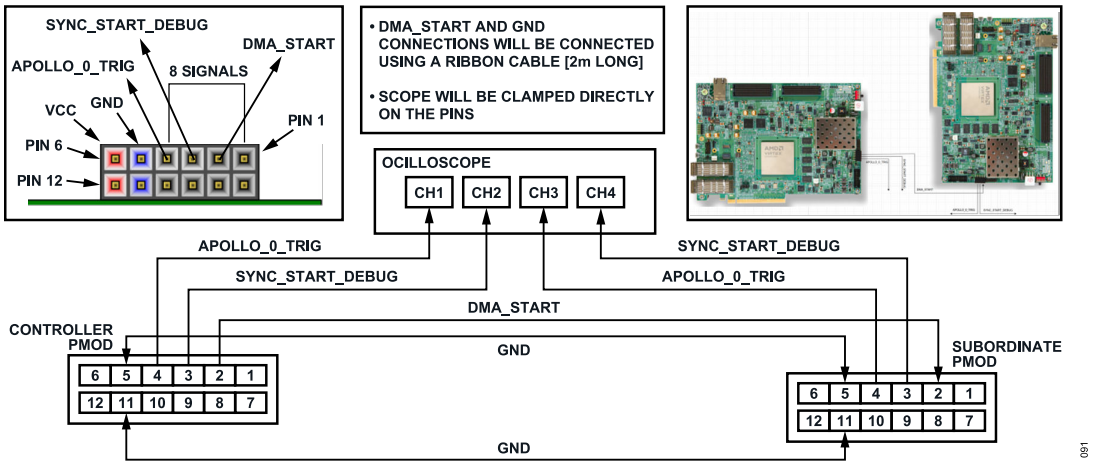


Figure 55. PMOD and Scope Connections

MULTICHIP SYNCHRONIZATION (MCS) THEORY

TRIG

Dual-ADXBAND16EBZ Synchronization and Phase Measurement

During this process, two ADXBAND16EBZ (quad AD9084 MxFE) boards are synchronized by using the axi_adf4030_trig of the primary board to issue a coordinated, synchronized start trigger to both boards.

The primary board transmits a DMA waveform, and the phase and sample offset between the primary receive and secondary receive captures is measured across multiple runs.

The dual-ADXBAND16EBZ synchronization and phase measurement flow follows:

1. Initialize the primary and secondary ADXBAND16EBZ devices.
2. Configure the axi_adf4030_trig and the ADF4030 PLL on the primary.
3. Resume JESD204 FSM on both boards, if paused.
4. Configure the NCOs and enabled the channels and buffer sizes on both boards.
5. The phase measurement loop follows:
    a. Arm sync start on both boards (transmit and receive).
    b. Load transmit waveform on the primary.
    c. Fire trigger from the axi_adf4030_trig of the primary.
    d. Verify sync disarm on both boards.
    e. Capture receive from both boards.
    f. Measure phase and sample delay.
6. Print statistics and, optionally, plot results.

Example GitHub scripts in analogdevices/pyadi-iio follow:

```
examples/dual_adxband16ebz_sync_start.py
adi/ad9084_mc.py
```

RELATED DOCUMENTS

Related Analog Devices technical articles, a design note, and a webcast follow:

- Das, Siddhartha, *Quad-Apollo MxFE: A 16 Receive/16 Transmit X-Band Direct Sampling Digital Beamforming Subsystem Reference Design*, Technical Article, Analog Devices, 2026
- Das, Siddhartha, *Reducing Clock Spur Interference in Phased Array Subsystems*, Technical Article, Analog Devices, 2026
- Delos, Peter; Frick, Charles; and Jones, Mike, *Multichannel RF to Bits Development Platform*, Design Note, Analog Devices, 2020
- Jones, Michael; Hennerich, Michael; and Delos, Peter, *Power-Up Phase Determinism Using Multichip Synchronization Features in Integrated Wideband DACs and ADCs*, Technical Article, Analog Devices, 2020
- Jones, Michael; Collins, Travis; and Frick, Charles, *Integrated Hardened DSP on DAC/ADC ICs Improves Wideband Multichannel Systems*, Technical Article, Analog Devices, 2021
- Jones, Michael, *Multi-Channel System Improvements Using Hardened DSP in Digitizer ICs*, Webcast, Analog Devices, 2021
- Delos, Peter and Jones, Michael, *Empirically Based Multichannel Phase Noise Model Validated in a 16-Channel Demonstrator*, Technical Article, Analog Devices, 2021

ORDERING INFORMATION

ORDERING GUIDE

Table 10. Evaluation Boards

Model[1]	Description
ADXBAND16EBZ	Evaluation Board
ADXBAND16EBZ-CAL	Calibration Board

[1] Z = RoHS-Compliant Part.

ESD Caution
ESD (electrostatic discharge) sensitive device. Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.