MimoKit Motherboard

REFERENCE MANUAL

Version 1.1 – June 09th 2008

Revision History

Revision	Date	Author	Comment
1.0	15-Feb-2008	Thierry Durand - Comsis	Document creation
1.1	09-Jun-2008	Thierry Durand – Comsis	Bugs in the ball assignments of the FPGA highway – fixed. CLK_1_P and CLK_1_N ball assignments crossed-over on FPGA2 – fixed. UART1 erroneaously connected to FPGA2 – fixed.

Table of Contents

1. About this manual

Scope of the Document

This document provides the complete functional description of the main board of the MimoKit hardware system. This system is based on a main PCI board carrying FPGAs, and an optional add-on radio front-end. The radio front-end is documented separately.

This document includes an architecture overview, a technical description of the major functional blocks, and an interface definition. The connectors and the FPGA ball assignments are covered in detail.

Intended audience

This document was written for design managers, system developers, ASIC designe engineers who are interested in evaluating MIMO technology and performing system prototyping.

Prior knowledge of electronics and RF systems is essential. Working knowledge of the Altera Quartus II environment is necessary to build a design and program the board.

Using this manual

This document covers the following sections:

- Section 1: this section
- Section 2: generic operation requirements, board handling information and intended use
- Section 3: high level board information, including block diagram
- Section 4: detailed on-board resource information, block by block

Contacting Comsis

Technical support:

support@comsis.fr

phone: +33(0)145891700

2. Introduction

System Requirements

Requirements for the installation and use of the MimoKit:

- a MimoKit board
- a recent version of Quartus II from Altera. Note that Quartus II Web Edition does not currently support the FPGA which are on the Mimokit. The commercial version is mandatory.
- a ByteBlaster, USB-Blaster or compatible FPGA programming device
- a PC with a parallel port (for ByteBlaster) or a USB port (for USB-Blaster)
- Optionally, a PC with a 3.3V PCI slot. This is essential to operate the PCI interface of the board, and is useful when the PCI interface is used as the debug port. It is however not required as the board can be operated standalone.

⚠ The Mimokit motherboard can only be inserted in a 3.3V PCI slot. Any attempt to insert the board in a 5V or universal (3.3V/5V) slot will result in permanent damage of the board and/or of the host.

- An ATX type of power supply with a Molex HDD connector. This is the supply source for the board, even if it's inserted into a PCI slot. No power is taken from the PCI connector.

Purpose of the Board

The MimoKit is designed for extensively networked embedded applications that require wireless LAN connectivity and Gigabit Ethernet. Based on two high-end field-programmable gate arrays (FPGAs), the MimoKit permits the implementation of extremely complex digital processing blocks, such as those found in modern multi-antenna WLAN interfaces. Moreover, it also has a sufficient gate capacity to host a complete system-on-programmable chip (SoPC), including microcontroller cores and advanced communication peripherals. Given these unique features, the MimoKit is the ideal platform for WLAN-capable SoC prototyping.

One major advantage of the MimoKit is the presence of the MIMO RF enabling the implementation of IEEE802.11n-compatible access points or stations. There are several variants of the RF daughter board. Please refer to the documentation of your RF daughter board for more information.

Handling Precautions

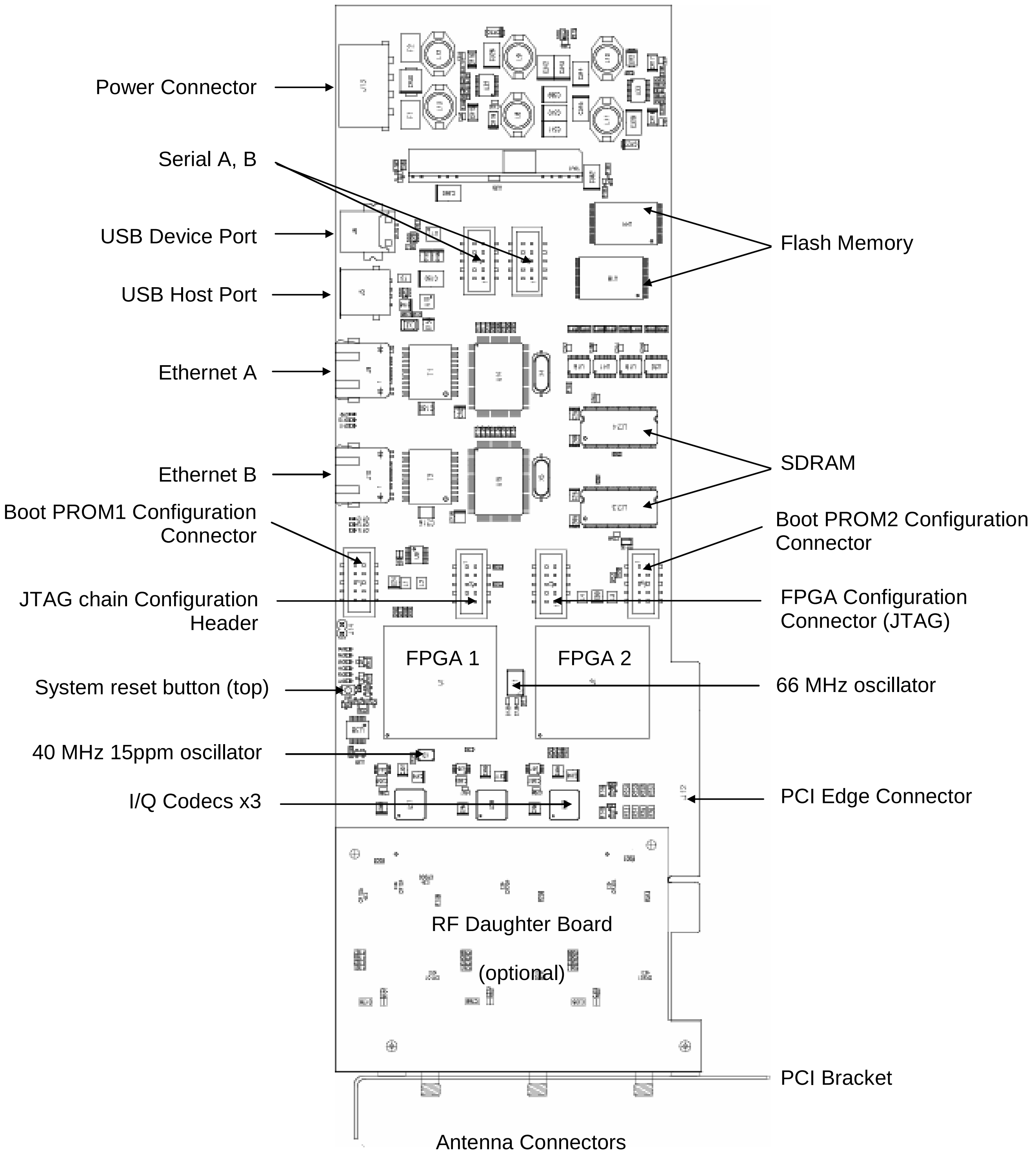
The MimoKit board contains components which are sensitive to electro-static discharge. The board can be permanently damaged if handled without following proper anti-static precautions.

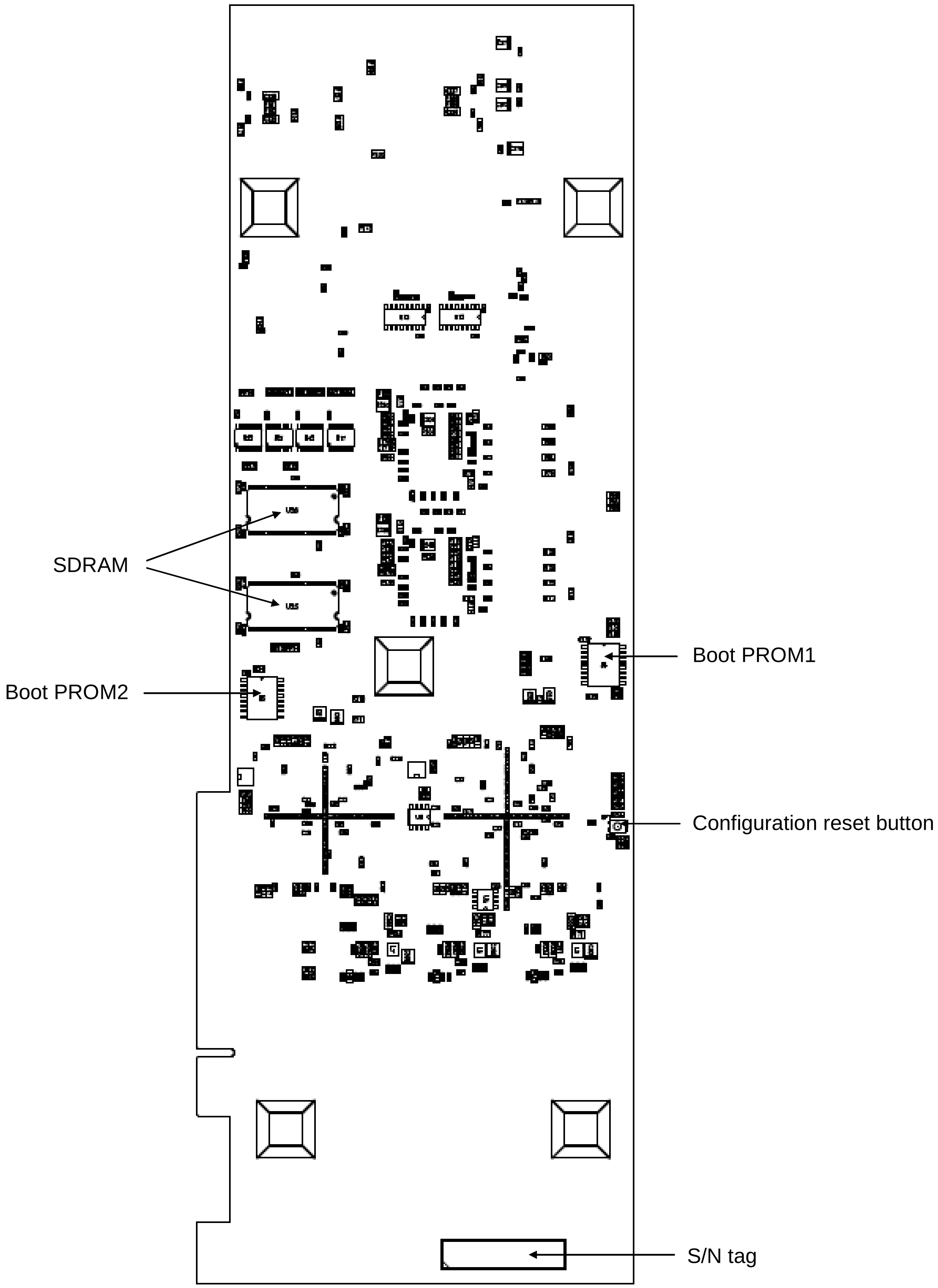
Please do use anti-static precautions when handling the board.

3. MimoKit Architecture

A Bird's Eye View of the MimoKit

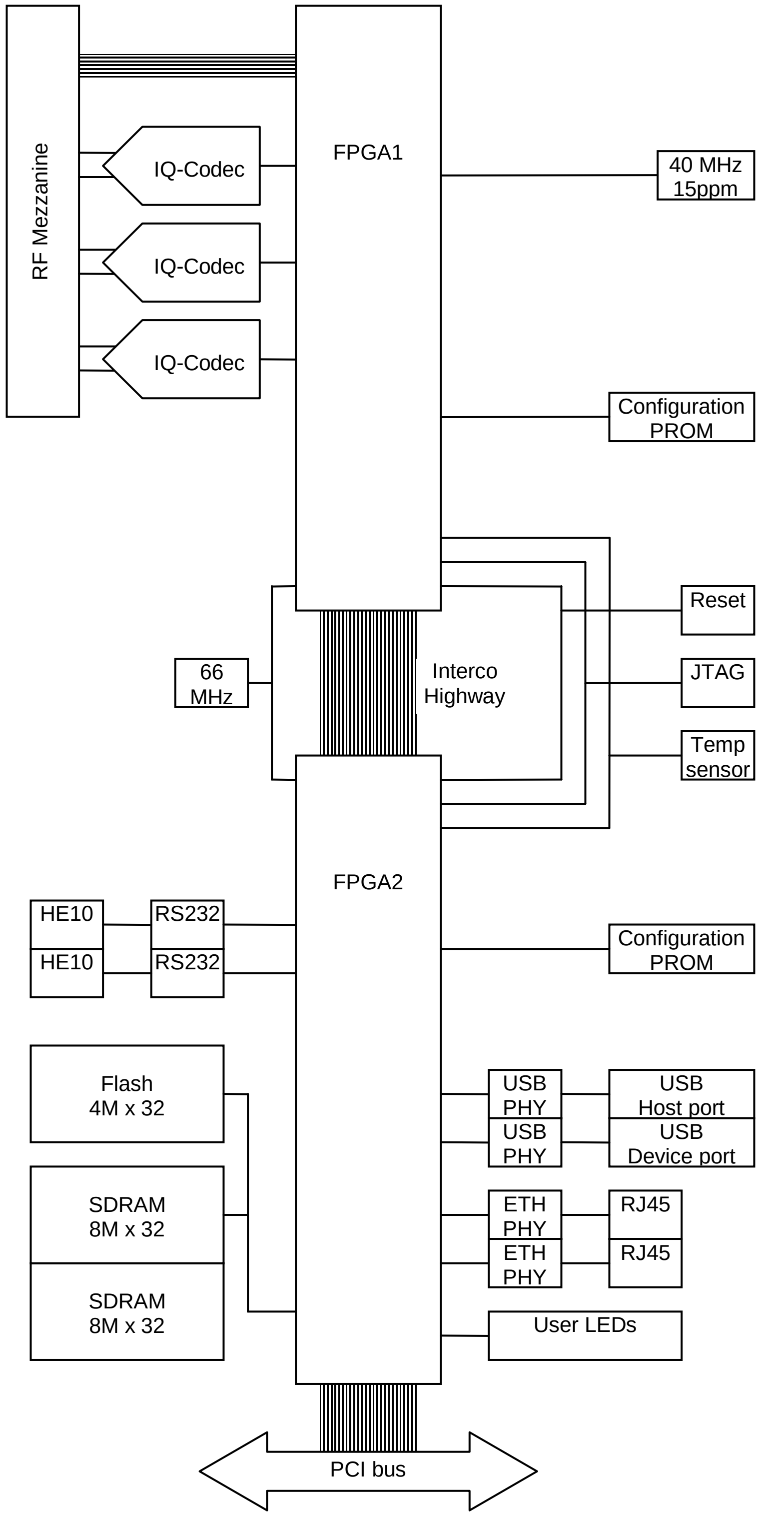
Illustration 1: Component placement overview




SDRAM
Boot PROM1
Boot PROM2
Configuration reset button
S/N tag

Block Diagram

Illustration 2: Block diagram of the MimoKit motherboard



4. Board Resources

This section details the hardware resources available on the MimoKit motherboard, and how they are connected to the central FPGA system.

Note that some resources are unique to a single FPGA, while some others are shared.

Power Supplies

The MimoKit motherboard is supplied through a standard HDD Molex connector. It can be easily connected to the HDD plug of a PC power supply box. The Molex connector provides a 5 V, a 12 V, and two ground wires.

The 12 V is not used on the motherboard itself. It is routed to the RF daughterboard connector in case the optional RF board needs it for its own purpose. The 12 V is also routed to a 2-pin header (J24) located close to the FPGA. This header can be used to run optional fans on top of the FPGA package if the complexity and operating frequency of their backend logic impose active cooling. However a fan is not necessary for most applications. The 12 V is protected by a replaceable fuse rated to 5 amps.

The 5 V input is used to derive all the power supplies necessary to the board operation:

- Digital 3.3V, 2.5V, 1.8V, 1.2V for the FPGA, memories, and all digital peripherals
- Three independent analog pairs of 3.3V and 2.85V, each pair supplying one IQ-codec chip

In some cases the optional RF daughter implements its own local power supply. There is a provision for these local supplies to send back status information and to drive 3 LEDs (typically one per MIMO stream) which are present on the motherboard.

The status of the various supplies is reflected by green LEDs listed in Table 1.

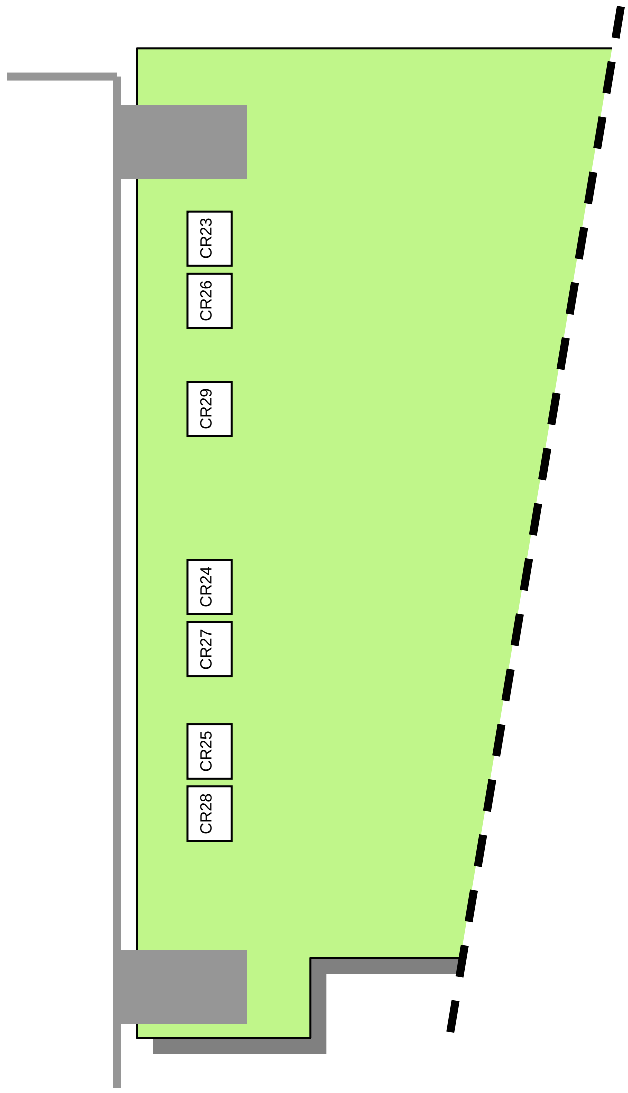
Table 1: Supply status LEDs

Supply	Voltage	LED ref	ON Condition
Digital	3.3 V	CR29 (green)	When the 4 voltages are within the acceptable range
	2.5 V		
	1.8 V		
	1.2 V		
Analog – Codec 1	3.3 V	CR23 (green)	When both voltages are within the acceptable range
	2.85 V		
Analog – Codec 2	3.3 V	CR24 (green)	When both voltages are within the acceptable range
	2.85 V		
Analog – Codec 3	3.3 V	CR25	When both voltages are within

Supply	Voltage	LED ref	ON Condition
	2.85 V	(green)	the acceptable range
RF – Supply 1	RF-dependent	CR26 (green)	Controlled by the RF daughterboard
RF – Supply 2	RF-dependent	CR27 (green)	Controlled by the RF daughterboard
RF – Supply 3	RF-dependent	CR28 (green)	Controlled by the RF daughterboard

The LEDs are located on the top side of the board, close to the PCI bracket, as shown in Illustration 3. When the PCI bracket is mounted, optical guides report the LED display on the bracket itself.

Illustration 3: Power status LEDs location



Reset

Two types of reset exist on the board:

- A configuration reset, which forces the FPGA to reconfigure by reading the contents of the boot PROM;
- A system reset, which is designed to be used as a master reset signal for the custom backend logic.

Each type of reset is controlled by its own push button. Pushing the button causes the corresponding reset to be forced low during a fixed period of time.

In order to avoid the confusion between the two push buttons, they have been located on different board sides. The configuration reset button is placed on the bottom side (back) of the board while the system reset button is placed on the component side.

Table 2 - Reset buttons

Reset Button	Function	FPGA Ball	Signal
BP1, bottom side	Configuration reset	U1-AL30	nCONFIG1
		U2-AL30	nCONFIG2
BP2, components side	System reset	U1-D17	RESET#
		U2-D17	

Note that pushing the configuration reset button while the board is inserted in a PCI slot of a PC will very likely crash the PC.

FPGA Devices

The MimoKit motherboard is equipped with two Altera EP2S180F1020 high-density FPGAs.

FPGA1 is connected to the mixed mode part of the board (the 3 IQ-codecs). In a typical MIMO WLAN application, FPGA1 will implement the physical layer of the WLAN interface.

FPGA2 is connected to memories and communication interfaces: USB device and host ports, Ethernet ports. In a typical application it will implement a microcontroller-based system.

FPGA1 and FPGA2 are connected to each other through a 460 wire highway. Numerous direction and electrical signal schemes are available addressing most interconnect requirements. See section FPGA Interconnect Highway for more details.

FPGA Clocks

The Altera EP2S180 devices handle the clock signals specifically. Special inputs and outputs are dedicated to clock signals, and are conveniently located close to the on-chip PLLs. The clock signals are also assigned to special global routing resources to minimize the clock skew. The ball assignment of the MimoKit motherboard takes advantage of this specific clock handling by using the dedicated clock inputs and outputs as much as possible. Note the clock-specific I/Os not only carry actual clock signals but also signals which have to be treated globally by the FPGA routing resources, e.g. resets and interrupts.

Table 3 summarize the ball assignments for the clock inputs of FPGA1.

Table 3 - Clock inputs on FPGA 1

FPGA CLK#	Signal Name	Function	FPGA1 Ball
CLK0p	CLK_RF2	Optional clock from RF daughter board. Please refer to the RF daughter board documentation for information on how this signal is driven.	T32
CLK0n		Tied to 0	T31
CLK1p	CLK_p_4	FPGA highway	T30
CLK1n	CLK_n_4	FPGA highway	T29
CLK2p		Tied to 0	U32
CLK2n	INT#	Temp monitor interrupt	U31
CLK3p		Tied to 0	U30
CLK3n	T_CRIT#	Critical temp signal	U29
CLK4p	CLK_p_5	FPGA highway	AM17
CLK4n	CLK_n_5	FPGA highway	AL17
CLK5p	CLK_RF1	Optional clock from RF daughter board. Please refer to the RF daughter board documentation for information on how this signal is driven.	AK17

FPGA CLK#	Signal Name	Function	FPGA1 Ball
CLK5n		Tied to 0	AJ17
CLK6p	CLK_40MHz	High accuracy 40 MHz clock from on-board TCXO	AM16
CLK6n		Tied to 0	AL16
CLK7p	FPGA1_CLK	66 MHz clock, buffered	AH16
CLK7n		Tied to 0	AG16
CLK8p	CLK_OUT_Codec1	Clock output from codec 1. Can be used to synchronize the data exchange between the codec and the FPGA	U1
CLK8n		Tied to 0	U2
CLK9p	Tx/Rx_CLK_Codec1	Tx/Rx direction control for codec 1	U3
CLK9n		Tied to 0	U4
CLK10p	CLK_OUT_Codec2	Clock output from codec 2. Can be used to synchronize the data exchange between the codec and the FPGA	T1
CLK10n		Tied to 0	T2
CLK11p	Tx/Rx_CLK_Codec2	Tx/Rx direction control for codec 2	T3
CLK11n		Tied to 0	T4
CLK12p	CLK_OUT_Codec3	Clock output from codec 3. Can be used to synchronize the data exchange between the codec and the FPGA	A16
CLK12n		Tied to 0	R16
CLK13p	Tx/Rx_CLK_Codec3	Tx/Rx direction control for codec 3	F16
CLK13n		Tied to 0	E16
CLK14p	FPGA2_CLK	Buffered 66 MHz clock source.	A17
CLK14n		Open	B17
CLK15p		Tied to 0	C17
CLK15n	RESET#	System reset	D17

Table 4 lists the assignment of the clock outputs on FPGA 1. Only the assigned balls are listed. Unlisted balls are left open.

Table 4 - Clock outputs on FPGA 1

FPGA CLK#	Signal Name	Function	FPGA1 Ball
PLL6_OUT0p	CLK_RF3	Clock signal to the RF daughter board. Please refer to the RF daughter board documentation for information on how this signal should be driven.	AK16
PLL6_OUT1p	CLK_80MHz	High accuracy clock to the analog front-end. This is typically a 80 MHz derived from the 40 MHz reference.	AJ15

FPGA CLK#	Signal Name	Function	FPGA1 Ball
PLL11_OUT0p	CLK_p_1	FPGA highway	B18
PLL11_OUT0n	CLK_n_1	FPGA highway	C18
PLL11_OUT1p	CLK_p_2	FPGA highway	D18
PLL11_OUT1n	CLK_n_2	FPGA highway	E18
PLL12_OUT0p	CLK_p_3	FPGA highway	AL18
PLL12_OUT0n	CLK_n_3	FPGA highway	AK18

Table 5 summarize the ball assignments for the clock inputs of FPGA2. Note that not all the signals connected on clock inputs are actually clock signals. There are either clocks or signals which may need special treatment in the FPGA, such as asynchronous reset or interrupt sources.

Table 5: Clock inputs on FPGA2

FPGA CLK#	Signal Name	Function	FPGA2 Ball
CLK0p	ETHERNET1_RX_CLK	MII/GMII Rx clock of Ethernet A	T32
CLK0n		Tied to 0	T31
CLK1p	ETHERNET1_TX_CLK	MII Tx clock of Ethernet A	T30
CLK1n		Tied to 0	T29
CLK2p	ETHERNET2_RX_CLK	MII/GMII Rx clock of Ethernet B	U32
CLK2n		Tied to 0	U31
CLK3p	ETHERNET2_TX_CLK	MII Tx clock of Ethernet B	U30
CLK3n		Tied to 0	U29
CLK4p		Tied to 0	AM17
CLK4n		Tied to 0	AL17
CLK5p	USB_HOST_CLK	USB host clock	AK17
CLK5n		Tied to 0	AJ17
CLK6p	PCI_CLK	PCI clock, directly from the PCI edge connector	AM16
CLK6n	PCI_RST	PCI reset, directly from the PCI edge connector	AL16
CLK7p	FPGA3_CLK	66 MHz clock, buffered	AH16
CLK7n		Tied to 0	AG16
CLK8p	CLK_p_1	FPGA highway	U1
CLK8n	CLK_n_1	FPGA highway	U2
CLK9p	CLK_p_2	FPGA highway	U3
CLK9n	CLK_n_2	FPGA highway	U4
CLK10p	CLK_p_3	FPGA highway	T1
CLK10n	CLK_n_3	FPGA highway	T2
CLK11p		Tied to 0	T3
CLK11n		Tied to 0	T4
CLK12p		Tied to 0	A16

FPGA CLK#	Signal Name	Function	FPGA2 Ball
CLK12n	INT#	Temp monitor interrupt	R16
CLK13p		Tied to 0	F16
CLK13n	T_CRIT#	Critical temp signal	E16
CLK14p	FPGA4_CLK	Buffered 66 MHz clock source.	A17
CLK14n		Open	B17
CLK15p		Tied to 0	C17
CLK15n	RESET#	System reset	D17

Table 6 lists the assignment of the clock outputs on FPGA 2. Only the assigned balls are listed. Unlisted balls are left open.

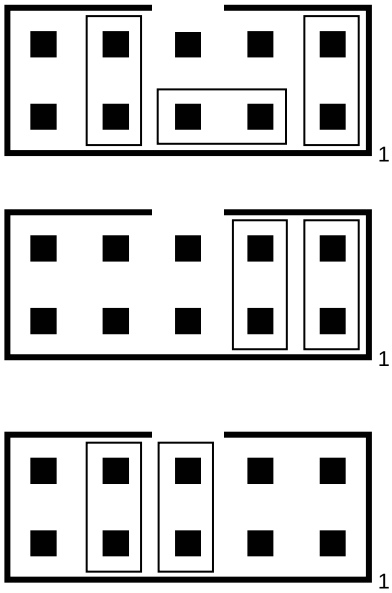
Table 6: Clock outputs of FPGA2

FPGA CLK#	Signal Name	Function	FPGA2 Ball
PLL5_OUT0p	CLK_p_4	FPGA highway	B15
PLL5_OUT0n	CLK_n_4	FPGA highway	C15
PLL5_OUT1p	CLK_p_5	FPGA highway	C16
PLL5_OUT1n	CLK_n_5	FPGA highway	D16
PLL6_OUT0p	RAM_CLK0	SDRAM clock	AK16
PLL6_OUT1p	RAM_CLK1	SDRAM clock (LSB device)	AJ15
PLL11_OUT0p	ETHERNET1_GTX_CLK	GMII Tx clock of Ethernet A	B18
PLL11_OUT1p	ETHERNET2_GTX_CLK	GMII Tx clock of Ethernet B	D18
PLL12_OUT0p	USB_DEVICE_CLOCK	USB device clock	AL18

JTAG Configuration

During the development phase, it is very convenient to program the FPGA through the JTAG connector (J3, see Illustration 1 for location). The jumpers on configuration header (J4, see Illustration 1 for location) define how the FPGA is connected to the JTAG chain. The possible configurations are listed in Illustration 4.

Illustration 4: J4 header description for JTAG configuration



FPGA1 and FPGA2 JTAG chains are connected in series

FPGA1 JTAG chain only

FPGA2 JTAG chain only

The left-most pins are unused and can be used to safely store a spare jumper.

The JTAG configuration can be made from within the Quartus II software using an Altera ByteBlaster, USB-Blaster, or any other compatible equipment.

Boot PROMs Configuration

When the debug phase is finished, the boot PROM can be programed with the resulting bitstreams, so that FPGA is configured automatically at start-up or when the configuration reset button is pushed. Two boot PROMs are present onboard, one for each FPGA. Each PROM has it own programmation connector:

- The boot PROM for FPGA1 is programed through J1
- The boot PROM for FPGA2 is programed through J2

Programing the boot PROM can be done via the Quartus II software using an Altera ByteBlaster, USB-Blaster, or any other compatible equipment.

In order to program the boot PROM, the following steps should be followed:

- Connect the ByteBlaster or USB-Blaster to the connector corresponding to the correct boot PROM.
- Launch the programing software. This can be done from within the Quartus development environment, or standalone.
- Select the programing mode as Active Serial. Selecting JTAG won’t work. Auto-detecting the boot PROM won’t work.
- Select your .pof bitstream file.

- Start programing which takes a few minutes to complete.
- Eventually push the configuration reset button to force a reconfiguration from the new boot PROM.

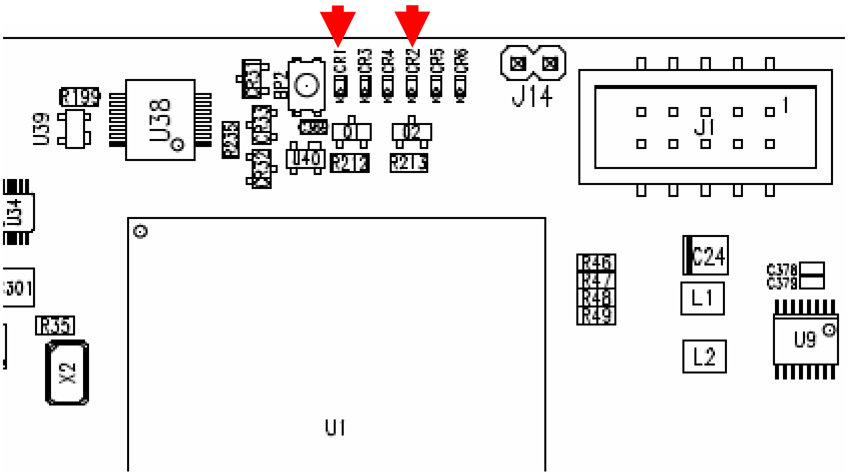
Configuration Status LEDs

Two LEDs reflect the configuration status of the FPGA, one for each device. When a LED is on, the configuration of the corresponding FPGA is complete.

- The configuration status of FPGA1 is reflected by CR1
- The configuration status of FPGA2 is reflected by CR2

CR1 and CR2 are located close to the reset buttons, as shown in Illustration 5

Illustration 5: Location of the FPGA configuration status LEDs



Temperature Sensors

The FPGA is a sensitive device, especially with regards to excessive operating temperature. When the back-end logic is complex and the operating frequency is high, the die temperature may become too hot and damage the device. The Altera EP2S180 has an internal temperature-sensitive diode which can be used to monitor the die temperature.

Both FPGAs are associated with an on-board temp sensor which measures the breakdown voltage of the diodes, compares it to user-programmable thresholds, and possibly triggers overtemp alert signals (interrupts). The temperature sensor is a National Semiconductor LM83, please refer to its datasheet for detailed information.

Table 7 details the connection of the temperature sensors to the FPGAs.

Table 7: Signal assignment of temperature sensor

Temp Sensor Signal	Function	FPGA1 ball	FPGA2 Ball
D1+	Positive terminal of temp measuring diode in FPGA1	G9	-
D2+	Positive terminal of temp measuring diode in FPGA2	-	G9

Temp Sensor Signal	Function	FPGA1 ball	FPGA2 Ball
D-	Negative terminal of both temp measuring diodes	B3	B3
/INT	Interrupt signal reporting a temperature over the programmed thresholds	U31	B16
/T_CRIT_A	Interrupt signal reporting a critical temperature condition on one FPGA. Immediate action is required.	U29	F16
SMBCLK	Clock signal of serial interface	N28	-
SMBDATA	Data signal of serial interface	N29	-

FPGA Interconnect Highway

FPGA1 and FPGA2 are interconnected through direct high speed ball-to-ball connections. Most of the interconnect signals are single-ended bidirectional and are multipurpose. Some signals carry clocks and are unidirectional while others can be used as high-speed LVDS pairs. The details of the interconnect highway are outlined in Table 8. The Net name column refers to the name associated with each interconnect net in the MimoKit motherboard schematic. All signals can be used in LVTTL signalling. LVDS capability is noted in the last column.

Table 8: Ball assignment of FPGA interconnect signals

Net name	FPGA1 ball	FPGA2 ball	Direction	LVDS capable
CLK_P_1	C18	U1	→	Yes
CLK_N_1	B18	U2	→	
CLK_P_2	D18	U3	→	Yes
CLK_N_2	E18	U4	→	
CLK_P_3	AL18	T1	→	Yes
CLK_N_3	AK18	T2	→	
CLK_P_4	T30	B15	←	Yes
CLK_N_4	T29	C15	←	
CLK_P_5	AM17	C16	←	Yes
CLK_N_5	AL17	D16	←	
CONNECT_1	AK20	B20	↔	
CONNECT_2	AJ20	C20	↔	
CONNECT_3	AJ19	D19	↔	
CONNECT_4	AD18	D20	↔	
CONNECT_5	AH19	E19	↔	
CONNECT_6	AH20	E20	↔	
CONNECT_7	-	-		
CONNECT_8	AB20	K18	↔	
CONNECT_9	AB21	L18	↔	
CONNECT_10	AM21	A21	↔	

Net name	FPGA1 ball	FPGA2 ball	Direction	LVDS capable
CONNECT_11	AM22	A22	↔	
CONNECT_12	AL20	B21	↔	
CONNECT_13	AL21	B22	↔	
CONNECT_14	AK21	C21	↔	
CONNECT_15	AK22	C22	↔	
CONNECT_16	AJ21	D21	↔	
CONNECT_17	AG22	D22	↔	
CONNECT_18	AJ22	D23	↔	
CONNECT_19	AF22	E22	↔	
CONNECT_20	AG20	F22	↔	
CONNECT_21	AE22	F23	↔	
CONNECT_22	AF19	H20	↔	
CONNECT_23	AB18	J19	↔	
CONNECT_24	AE20	J20	↔	
CONNECT_25	AC17	K19	↔	
CONNECT_26	AB17	L19	↔	
CONNECT_27	AM23	A23	↔	
CONNECT_28	AM24	A24	↔	
CONNECT_29	AM25	A25	↔	
CONNECT_30	AM26	A26	↔	
CONNECT_31	AL22	B23	↔	
CONNECT_32	AL23	B24	↔	
CONNECT_33	AL24	B25	↔	
CONNECT_34	AL25	B26	↔	
CONNECT_35	AK23	C23	↔	
CONNECT_36	AK24	C24	↔	
CONNECT_37	AK26	C26	↔	
CONNECT_38	AH24	D26	↔	
CONNECT_39	AF20	G21	↔	
CONNECT_40	AG23	G22	↔	
CONNECT_41	AE21	H21	↔	
CONNECT_42	AD20	J21	↔	
CONNECT_43	AE19	K20	↔	
CONNECT_44	AM27	A27	↔	
CONNECT_45	AM28	A28	↔	
CONNECT_46	AL26	B27	↔	
CONNECT_47	AL27	B28	↔	
CONNECT_48	AK25	C25	↔	
CONNECT_49	AK27	C27	↔	

Net name	FPGA1 ball	FPGA2 ball	Direction	LVDS capable
CONNECT_50	AH22	D25	↔	
CONNECT_51	AG24	D27	↔	
CONNECT_52	AJ23	E24	↔	
CONNECT_53	AD22	E25	↔	
CONNECT_54	AK29	E26	↔	
CONNECT_55	AL29	E27	↔	
CONNECT_56	AF21	H22	↔	
CONNECT_57	AC19	K21	↔	
CONNECT_58	AD19	L20	↔	
CONNECT_59	AM29	A29	↔	
CONNECT_60	AL28	B29	↔	
CONNECT_61	AJ27	C28	↔	
CONNECT_62	AJ28	C29	↔	
CONNECT_63	AH28	D28	↔	
CONNECT_64	AJ26	E28	↔	
CONNECT_65	AC20	J22	↔	
CONNECT_66	AC21	K22	↔	
CONNECT_67	AC18	L21	↔	
CONNECT_68	AM4	A4	↔	
CONNECT_69	AL4	B4	↔	
CONNECT_70	AK4	C4	↔	
CONNECT_71	AK5	C5	↔	
CONNECT_72	AH6	D5	↔	
CONNECT_73	AH5	E5	↔	
CONNECT_74	AE10	J11	↔	
CONNECT_75	AD12	J12	↔	
CONNECT_76	AD11	K12	↔	
CONNECT_77	AC11	L13	↔	
CONNECT_78	AM5	A5	↔	
CONNECT_79	AM6	A6	↔	
CONNECT_80	AM7	A7	↔	
CONNECT_81	AM8	A8	↔	
CONNECT_82	AL5	B5	↔	
CONNECT_83	AL6	B6	↔	
CONNECT_84	AL7	B7	↔	
CONNECT_85	AL8	B8	↔	
CONNECT_86	AK6	C6	↔	
CONNECT_87	AK7	C7	↔	
CONNECT_88	AK8	C8	↔	

Net name	FPGA1 ball	FPGA2 ball	Direction	LVDS capable
CONNECT_89	AK9	C9	↔	
CONNECT_90	AJ6	D6	↔	
CONNECT_91	AJ7	D7	↔	
CONNECT_92	AJ5	E6	↔	
CONNECT_93	AG8	E7	↔	
CONNECT_94	AF11	H11	↔	
CONNECT_95	AF12	H12	↔	
CONNECT_96	AD13	J13	↔	
CONNECT_97	AC12	K13	↔	
CONNECT_98	AC14	L14	↔	
CONNECT_99	AM9	A9	↔	
CONNECT_100	AM10	A10	↔	
CONNECT_101	AL9	B9	↔	
CONNECT_102	AL10	B10	↔	
CONNECT_103	AK10	C10	↔	
CONNECT_104	AK11	C11	↔	
CONNECT_105	AJ8	D8	↔	
CONNECT_106	AJ10	D10	↔	
CONNECT_107	AJ11	D11	↔	
CONNECT_108	AH7	E8	↔	
CONNECT_109	AH9	E9	↔	
CONNECT_110	AH8	F8	↔	
CONNECT_111	AG9	F9	↔	
CONNECT_112	AG11	F10	↔	
CONNECT_113	AE13	H13	↔	
CONNECT_114	AC13	J14	↔	
CONNECT_115	AC15	K14	↔	
CONNECT_116	AD14	K15	↔	
CONNECT_117	AB13	L15	↔	
CONNECT_118	AM11	A11	↔	
CONNECT_119	AM12	A12	↔	
CONNECT_120	AL11	B11	↔	
CONNECT_121	AL12	B12	↔	
CONNECT_122	AK12	C12	↔	
CONNECT_123	AJ12	D12	↔	
CONNECT_124	AH11	E11	↔	
CONNECT_125	AH13	E13	↔	
CONNECT_126	AE12	F11	↔	
CONNECT_127	AG12	F12	↔	

Net name	FPGA1 ball	FPGA2 ball	Direction	LVDS capable
CONNECT_128	AG13	F13	↔	
CONNECT_129	AG15	F14	↔	
CONNECT_130	AG10	G10	↔	
CONNECT_131	AF10	G11	↔	
CONNECT_132	AE11	G12	↔	
CONNECT_133	AF13	G13	↔	
CONNECT_134	AE14	H14	↔	
CONNECT_135	AC16	J15	↔	
CONNECT_136	AB14	L16	↔	
CONNECT_137	AM14	A14	↔	
CONNECT_138	AL13	B13	↔	
CONNECT_139	AL14	B14	↔	
CONNECT_140	AK13	C13	↔	
CONNECT_141	AJ13	D13	↔	
CONNECT_142	AJ14	D14	↔	
CONNECT_143	AH14	E14	↔	
CONNECT_144	AG14	F15	↔	
CONNECT_145	AB15	K16	↔	
CONNECT_146	AB12	K17	↔	
CONNECT_147	AB16	L17	↔	
CONNECT_148	AJ31	U22	↔	
CONNECT_149	AH31	U23	↔	
CONNECT_150	AE29	U27	↔	
CONNECT_151	AH30	V23	↔	
CONNECT_152	AG30	V24	↔	
CONNECT_153	AJ32	W22	↔	
CONNECT_154	AG31	W23	↔	
CONNECT_155	AF30	W24	↔	
CONNECT_156	AF31	W25	↔	
CONNECT_157	AE30	W26	↔	
CONNECT_158	AB31	W27	↔	
CONNECT_159	AG29	Y22	↔	
CONNECT_160	AD24	Y23	↔	
CONNECT_161	AA24	Y24	↔	
CONNECT_162	AE28	Y25	↔	
CONNECT_163	AE27	Y26	↔	
CONNECT_164	AB32	Y27	↔	
CONNECT_165	AB25	AA24	↔	
CONNECT_166	AB26	AA25	↔	

Net name	FPGA1 ball	FPGA2 ball	Direction	LVDS capable
CONNECT_167	AE26	AA26	↔	
CONNECT_168	AA31	AA27	↔	
CONNECT_169	Y24	AB25	↔	
CONNECT_170	AD25	AB26	↔	
CONNECT_171	AA32	AB27	↔	
CONNECT_172	AB30	AC26	↔	
CONNECT_173	AA30	AC27	↔	
CONNECT_174	W32	AC29	↔	
CONNECT_175	Y30	AC30	↔	
CONNECT_176	AB29	AD26	↔	
CONNECT_177	AA29	AD29	↔	
CONNECT_178	W31	AD30	↔	
CONNECT_179	W29	AF31	↔	
CONNECT_180	W28	AF32	↔	
CONNECT_181	AF29	AA22	↔	
CONNECT_182	Y22	AA23	↔	
CONNECT_183	AC24	AB23	↔	
CONNECT_184	AE25	AB24	↔	
CONNECT_185	AA25	AC24	↔	
CONNECT_186	Y23	AC25	↔	
CONNECT_187	AA26	AD24	↔	
CONNECT_188	Y26	AD25	↔	
CONNECT_189	AB27	AE26	↔	
CONNECT_190	Y29	AE27	↔	
CONNECT_191	AB28	AF30	↔	
CONNECT_192	Y28	AG29	↔	
CONNECT_193	AH29	AG30	↔	
CONNECT_194	AA27	AH29	↔	
CONNECT_195	W27	AH30	↔	
CONNECT_196	Y27	AH32	↔	
CONNECT_197	W26	AJ31	↔	
CONNECT_198	Y25	AJ32	↔	
CONNECT_199	M24	G32	↔	
CONNECT_200	R26	H29	↔	
CONNECT_201	T27	H30	↔	
CONNECT_202	T28	H31	↔	
CONNECT_203	R29	H32	↔	
CONNECT_204	K31	J29	↔	
CONNECT_205	M29	J30	↔	

Net name	FPGA1 ball	FPGA2 ball	Direction	LVDS capable
CONNECT_206	R31	J31	↔	
CONNECT_207	R30	J32	↔	
CONNECT_208	T22	K29	↔	
CONNECT_209	M30	K30	↔	
CONNECT_210	R25	K31	↔	
CONNECT_211	R27	K32	↔	
CONNECT_212	J29	L23	↔	
CONNECT_213	E32	L24	↔	
CONNECT_214	M23	M22	↔	
CONNECT_215	R28	M23	↔	
CONNECT_216	M22	M24	↔	
CONNECT_217	R24	M25	↔	
CONNECT_218	T23	M26	↔	
CONNECT_219	H29	M27	↔	
CONNECT_220	K29	P27	↔	
CONNECT_221	K32	P31	↔	
CONNECT_222	M26	P32	↔	
CONNECT_223	L24	R22	↔	
CONNECT_224	R23	R23	↔	
CONNECT_225	G31	R24	↔	
CONNECT_226	H31	R25	↔	
CONNECT_227	L23	R26	↔	
CONNECT_228	G32	R27	↔	
CONNECT_229	H32	R28	↔	
CONNECT_230	J31	R29	↔	
CONNECT_231	J32	R30	↔	
CONNECT_232	K30	R31	↔	
CONNECT_233	F31	T22	↔	
CONNECT_234	F32	T23	↔	
CONNECT_235	J30	T27	↔	
CONNECT_236	H30	T28	↔	
CONNECT_237	D11	AG15	↔	
CONNECT_238	D14	AH14	↔	
CONNECT_239	D13	AH13	↔	
CONNECT_240	B13	AJ11	↔	
CONNECT_241	H14	AD13	↔	
CONNECT_242	E9	AE12	↔	
CONNECT_243	E14	AE13	↔	
CONNECT_244	C8	AF10	↔	

Net name	FPGA1 ball	FPGA2 ball	Direction	LVDS capable
CONNECT_245	C13	AF11	↔	
CONNECT_246	C11	AK8	↔	
CONNECT_247	B14	AK9	↔	
CONNECT_248	K15	AB13	↔	
CONNECT_249	G13	AC12	↔	
CONNECT_250	K16	AC13	↔	
CONNECT_251	G12	AD12	↔	
CONNECT_252	E8	AE11	↔	
CONNECT_253	F15	AG8	↔	
CONNECT_254	A9	AG9	↔	
CONNECT_255	D10	AH6	↔	
CONNECT_256	C9	AH7	↔	
CONNECT_257	C10	AH8	↔	
CONNECT_258	A14	AH9	↔	
CONNECT_259	A10	AJ6	↔	
CONNECT_260	B10	AJ7	↔	
CONNECT_261	B9	AH5	↔	
CONNECT_262	A29	AJ27	↔	
CONNECT_263	A26	AK27	↔	
CONNECT_264	K17	AL28	↔	
CONNECT_265	B24	AE19	↔	
CONNECT_266	B26	AF19	↔	
CONNECT_267	B25	AF20	↔	
CONNECT_268	A25	AG20	↔	
CONNECT_269	A24	AG22	↔	
CONNECT_270	A23	AG23	↔	
CONNECT_271	J19	AH22	↔	
CONNECT_272	L17	AJ22	↔	
CONNECT_273	B23	AB17	↔	
LVDS_RX_1P	L1	AB7	→	Yes
LVDS_RX_1N	L2	AB8	→	
LVDS_RX_2P	E3	AB9	→	Yes
LVDS_RX_2N	E4	AB10	→	
LVDS_RX_3P	G3	AC6	→	Yes
LVDS_RX_3N	G4	AC7	→	
LVDS_RX_4P	F1	AC8	→	Yes
LVDS_RX_4N	F2	AC9	→	
LVDS_RX_5P	J3	AD10	→	Yes
LVDS_RX_5N	J4	AD7	→	

Net name	FPGA1 ball	FPGA2 ball	Direction	LVDS capable
LVDS_RX_6P	G5	AD8	→	Yes
LVDS_RX_6N	G6	AD9	→	
LVDS_RX_7P	H3	W4	→	Yes
LVDS_RX_7N	H4	W5	→	
LVDS_RX_8P	L3	W10	→	Yes
LVDS_RX_8N	L4	W11	→	
LVDS_RX_9P	N4	Y6	→	Yes
LVDS_RX_9N	N5	Y7	→	
LVDS_RX_10P	F3	Y8	→	Yes
LVDS_RX_10N	F4	Y9	→	
LVDS_RX_11P	N2	Y10	→	Yes
LVDS_RX_11N	N3	Y11	→	
LVDS_RX_12P	H1	AA6	→	Yes
LVDS_RX_12N	H2	AA7	→	
LVDS_RX_13P	E1	AA8	→	Yes
LVDS_RX_13N	E2	AA9	→	
LVDS_RX_14P	D1	AA10	→	Yes
LVDS_RX_14N	D2	AA11	→	
LVDS_RX_15P	G1	AB5	→	Yes
LVDS_RX_15N	G2	AB6	→	
LVDS_RX_16P	R2	U5	→	Yes
LVDS_RX_16N	R3	U6	→	
LVDS_RX_17P	M3	U10	→	Yes
LVDS_RX_17N	M4	U11	→	
LVDS_RX_18P	K3	V4	→	Yes
LVDS_RX_18N	K4	V5	→	
LVDS_RX_19P	K1	V6	→	Yes
LVDS_RX_19N	K2	V7	→	
LVDS_RX_20P	P1	V9	→	Yes
LVDS_RX_20N	P2	V10	→	
LVDS_RX_21P	J1	W6	→	Yes
LVDS_RX_21N	J2	W7	→	
LVDS_RX_22P	M1	W8	→	Yes
LVDS_RX_22N	M2	W9	→	
LVDS_RX_23P	AF1	P4	→	Yes
LVDS_RX_23N	AF2	P5	→	
LVDS_RX_24P	AB1	P6	→	Yes
LVDS_RX_24N	AB2	P7	→	
LVDS_RX_25P	AC3	P10	→	Yes

Net name	FPGA1 ball	FPGA2 ball	Direction	LVDS capable
LVDS_RX_25N	AC4	P11	→	
LVDS_RX_26P	AA3	R4	→	Yes
LVDS_RX_26N	AA4	R5	→	
LVDS_RX_27P	AE5	R6	→	Yes
LVDS_RX_27N	AE6	R7	→	
LVDS_RX_28P	V2	R10	→	Yes
LVDS_RX_28N	V3	R11	→	
LVDS_RX_29P	AE3	T5	→	Yes
LVDS_RX_29N	AE4	T6	→	
LVDS_RX_30P	Y4	T10	→	Yes
LVDS_RX_30N	Y5	T11	→	
LVDS_RX_31P	Y2	L5	→	Yes
LVDS_RX_31N	Y3	L6	→	
LVDS_RX_32P	AD1	L9	→	Yes
LVDS_RX_32N	AD2	L10	→	
LVDS_RX_33P	AG3	M6	→	Yes
LVDS_RX_33N	AG4	M7	→	
LVDS_RX_34P	AE1	M8	→	Yes
LVDS_RX_34N	AE2	M9	→	
LVDS_RX_35P	AH3	M10	→	Yes
LVDS_RX_35N	AH4	M11	→	
LVDS_RX_36P	W1	N6	→	Yes
LVDS_RX_36N	W2	N7	→	
LVDS_RX_37P	AF3	N8	→	Yes
LVDS_RX_37N	AF4	N9	→	
LVDS_RX_38P	AJ1	N10	→	Yes
LVDS_RX_38N	AJ2	N11	→	
LVDS_RX_39P	AD3	P8	→	Yes
LVDS_RX_39N	AD4	P9	→	
LVDS_TX_1P	M10	AE5	←	Yes
LVDS_TX_1N	M11	AE6	←	
LVDS_TX_2P	J6	AF3	←	Yes
LVDS_TX_2N	J7	AF4	←	
LVDS_TX_3P	M8	AG3	←	Yes
LVDS_TX_3N	M9	AG4	←	
LVDS_TX_4P	J8	AH1	←	Yes
LVDS_TX_4N	J9	AH2	←	
LVDS_TX_5P	K8	AH3	←	Yes
LVDS_TX_5N	K9	AH4	←	

Net name	FPGA1 ball	FPGA2 ball	Direction	LVDS capable
LVDS_TX_6P	L9	AJ1	←	Yes
LVDS_TX_6N	L10	AJ2	←	
LVDS_TX_7P	L5	AB3	←	Yes
LVDS_TX_7N	L6	AB4	←	
LVDS_TX_8P	R6	AC1	←	Yes
LVDS_TX_8N	R7	AC2	←	
LVDS_TX_9P	P10	AC3	←	Yes
LVDS_TX_9N	P11	AC4	←	
LVDS_TX_10P	R4	AD1	←	Yes
LVDS_TX_10N	R5	AD2	←	
LVDS_TX_11P	K6	AD3	←	Yes
LVDS_TX_11N	K7	AD4	←	
LVDS_TX_12P	P4	AE1	←	Yes
LVDS_TX_12N	P5	AE2	←	
LVDS_TX_13P	N8	AE3	←	Yes
LVDS_TX_13N	N9	AE4	←	
LVDS_TX_14P	N6	AF1	←	Yes
LVDS_TX_14N	N7	AF2	←	
LVDS_TX_15P	L7	AG1	←	Yes
LVDS_TX_15N	L8	AG2	←	
LVDS_TX_16P	N10	V2	←	Yes
LVDS_TX_16N	N11	V3	←	
LVDS_TX_17P	T5	W1	←	Yes
LVDS_TX_17N	T6	W2	←	
LVDS_TX_18P	M6	Y2	←	Yes
LVDS_TX_18N	M7	Y3	←	
LVDS_TX_19P	P8	Y4	←	Yes
LVDS_TX_19N	P9	Y5	←	
LVDS_TX_20P	T10	AA1	←	Yes
LVDS_TX_20N	T11	AA2	←	
LVDS_TX_21P	R10	AA3	←	Yes
LVDS_TX_21N	R11	AA4	←	
LVDS_TX_22P	P6	AB1	←	Yes
LVDS_TX_22N	P7	AB2	←	
LVDS_TX_23P	AB5	L1	←	Yes
LVDS_TX_23N	AB6	L2	←	
LVDS_TX_24P	V9	L3	←	Yes
LVDS_TX_24N	V10	L4	←	
LVDS_TX_25P	U10	M1	←	Yes

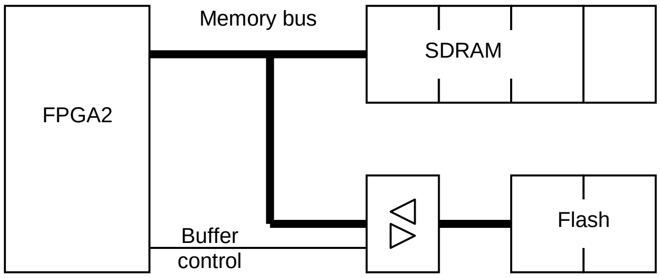
Net name	FPGA1 ball	FPGA2 ball	Direction	LVDS capable
LVDS_TX_25N	U11	M2	←	
LVDS_TX_26P	V6	M3	←	Yes
LVDS_TX_26N	V7	M4	←	
LVDS_TX_27P	Y6	N2	←	Yes
LVDS_TX_27N	Y7	N3	←	
LVDS_TX_28P	U5	N4	←	Yes
LVDS_TX_28N	U6	N5	←	
LVDS_TX_29P	W4	P1	←	Yes
LVDS_TX_29N	W5	P2	←	
LVDS_TX_30P	V4	R2	←	Yes
LVDS_TX_30N	V5	R3	←	
LVDS_TX_31P	AB7	F1	←	Yes
LVDS_TX_31N	AB8	F2	←	
LVDS_TX_32P	AD8	F3	←	Yes
LVDS_TX_32N	AD9	F4	←	
LVDS_TX_33P	AA6	G1	←	Yes
LVDS_TX_33N	AA7	G2	←	
LVDS_TX_34P	AB9	G3	←	Yes
LVDS_TX_34N	AB10	G4	←	
LVDS_TX_35P	W6	H1	←	Yes
LVDS_TX_35N	W7	H2	←	
LVDS_TX_36P	W8	J1	←	Yes
LVDS_TX_36N	W9	J2	←	
LVDS_TX_37P	Y10	J3	←	Yes
LVDS_TX_37N	Y11	J4	←	
LVDS_TX_38P	AC6	K1	←	Yes
LVDS_TX_38N	AC7	K2	←	
LVDS_TX_39P	W10	K3	←	Yes
LVDS_TX_39N	W11	K4	←	
LVDS_TX_40P	AA8	D1	←	Yes
LVDS_TX_40N	AA9	D2	←	
LVDS_TX_41P	AC8	E1	←	Yes
LVDS_TX_41N	AC9	E2	←	
LVDS_TX_42P	AD6	E3	←	Yes
LVDS_TX_42N	AD7	E4	←	
LVDS_TX_43P	AA10	G5	←	Yes
LVDS_TX_43N	AA11	G6	←	
LVDS_TX_44P	Y8	H3	←	Yes
LVDS_TX_44N	Y9	H4	←	

Memories

FPGA2 is connected to two types of memories: SDRAM and Flash.

The SDRAM and the Flash share the same Bus. The SDRAM is connected directly to the FPGA2 balls, while the Flash is connected through bidirectional buffers. The buffer direction and output controls are driven by the FPGA2. A simplified view of the memory topology is shown in Illustration 6. For a more detailed description, please refer to the MimoKit motherboard schematics.

Illustration 6: Simplified view of the memory circuitry



The SDRAM is composed of four 8M x 16 devices arranged to form two banks of 8M x 32 (i.e. two banks of 32 Mbytes).

The Flash is composed of two 4M x 16 devices arranged to form one bank of 4M x 32 (i.e. one bank of 16 Mbytes). Note that some boards may be equipped with pin compatible Flash devices with a different capacity (e.g. two 1M x 16).

Table 9 lists all the connections of the FPGA2 to the memory bus.

Note that the address signals of the SDRAM are offset by 10 bits. For example, the pin A0 of the SDRAM is connected the A10 signal of the memory bus. This offset should be taken into account by the controlling logic in the FPGA so that there is no change in the logical memory mapping.

Table 9: FPGA2 ball assignment to the memory bus

Bus Signal	Function	FPGA2 Ball
A0	A0 of the Flash (through buffer)	AB18
A1	A1 of the Flash (through buffer)	AD19
A2	A2 of the Flash (through buffer)	AM23
A3	A3 of the Flash (through buffer)	AL23
A4	A4 of the Flash (through buffer)	AK23
A5	A5 of the Flash (through buffer)	AJ23
A6	A6 of the Flash (through buffer)	AM24
A7	A7 of the Flash (through buffer)	AL24
A8	A8 of the Flash (through buffer)	AK24
A9	A9 of the Flash (through buffer)	AH24
A10	A0 of the SDRAM A10 of the Flash (through buffer)	AM25

Bus Signal	Function	FPGA2 Ball
A11	A1 of the SDRAM A11 of the Flash (through buffer)	AL25
A12	A2 of the SDRAM A12 of the Flash (through buffer)	AK25
A13	A3 of the SDRAM A13 of the Flash (through buffer)	AJ25
A14	A4 of the SDRAM A14 of the Flash (through buffer)	AH25
A15	A5 of the SDRAM A15 of the Flash (through buffer)	AM26
A16	A6 of the SDRAM A16 of the Flash (through buffer)	AL26
A17	A7 of the SDRAM A17 of the Flash (through buffer)	AK26
A18	A8 of the SDRAM A18 of the Flash (through buffer)	AJ26
A19	A9 of the SDRAM A19 of the Flash (through buffer)	AH26
A20	A10 of the SDRAM A20 of the Flash (through buffer)	AM27
A21	A11 of the SDRAM A21 of the Flash (through buffer)	AL27
D0	D0 of the SDRAM and the Flash	AF29
D1	D1 of the SDRAM and the Flash	AE32
D2	D2 of the SDRAM and the Flash	AE31
D3	D3 of the SDRAM and the Flash	AE30
D4	D4 of the SDRAM and the Flash	AE29
D5	D5 of the SDRAM and the Flash	AE28
D6	D6 of the SDRAM and the Flash	AD32
D7	D7 of the SDRAM and the Flash	AD31
D8	D8 of the SDRAM and the Flash	AC32
D9	D9 of the SDRAM and the Flash	AC31
D10	D10 of the SDRAM and the Flash	AB32
D11	D11 of the SDRAM and the Flash	AB31
D12	D12 of the SDRAM and the Flash	AB30
D13	D13 of the SDRAM and the Flash	AB29
D14	D14 of the SDRAM and the Flash	AB28
D15	D15 of the SDRAM and the Flash	AA32
D16	D16 of the SDRAM and the Flash	AA31
D17	D17 of the SDRAM and the Flash	AA30
D18	D18 of the SDRAM and the Flash	AA29
D19	D19 of the SDRAM and the Flash	Y31
D20	D20 of the SDRAM and the Flash	Y30

Bus Signal	Function	FPGA2 Ball
D21	D21 of the SDRAM and the Flash	Y29
D22	D22 of the SDRAM and the Flash	Y28
D23	D23 of the SDRAM and the Flash	W32
D24	D24 of the SDRAM and the Flash	W31
D25	D25 of the SDRAM and the Flash	W29
D26	D26 of the SDRAM and the Flash	W28
D27	D27 of the SDRAM and the Flash	V31
D28	D28 of the SDRAM and the Flash	V30
D29	D29 of the SDRAM and the Flash	V29
D30	D30 of the SDRAM and the Flash	V28
D31	D31 of the SDRAM and the Flash	U28
BA0	Bank address 0 of SDRAM	AH28
BA1	Bank address 1 of SDRAM	AM29
DQM0	Byte mask 0 of SDRAM	AJ21
DQM1	Byte mask 1 of SDRAM	AM22
DQM2	Byte mask 2 of SDRAM	AL22
DQM3	Byte mask 3 of SDRAM	AK22
/WE	Write enable of SDRAM	AL20
/CAS	Column address select of SDRAM	AM21
/RAS	Row address select of SDRAM	AH20
CLK0	Clock of SDRAM bank 0	AK16
CKE0	Clock enable of SDRAM bank 0	AK20
CLK1	Clock of SDRAM bank 1	AJ15
CKE1	Clock enable of SDRAM bank 1	AJ20
/CS0	Chip select of SDRAM bank 0	AL21
/CS1	Chip select of SDRAM bank 1	AK21
/WE	Write enable of Flash through buffer	AG31
/OE	Output enable of Flash through buffer	AG32
/CE	Chip select of Flash through buffer	AH31
/BUF_DIR	Buffer direction control	AE25
/BUF_OE	Buffer output enable	AD27

PCI Interface

The signals of the PCI edge connector are directly routed to balls of the FPGA2. Due to a limitation of the FPGA technology, only 3.3V signalling can be accepted.

Note that inserting the board into a PCI slot is not mandatory to use the board. No power is taken from the PCI connector.

⚠ The MimoKit motherboard can only be used with a 3.3V PCI slot. Any attempt to insert the board in a 5V or universal (3.3V/5V) slot will result in permanent damage of the board and/or of the host.

Note that the FPGA2 should already be configured with the PCI backend logic when the host (typ. a PC) boots. Otherwise, the MimoKit PCI device will not be detected by the host's operating system. This condition is fulfilled when the configuration is made out of the on-board Flash configuration device. If FPGA2 is to be programmed by JTAG, then the host must be restarted after the configuration to ensure the new MimoKit PCI device is detected.

Table 10 shows the PCI signal assignments on the FPGA2 balls.

Table 10: PCI signal assignments on FPGA2 balls

PCI signal	FPGA2 ball	PCI signal	FPGA2 ball	PCI signal	FPGA2 ball
/RST	AL16	AD13	AM10	AD30	AL4
CLK	AM16	AD14	AL10	AD31	AM4
/GNT	AC14	AD15	AK10	/CBE0	AC11
/REQ	AE14	AD16	AM9	/CBE1	AB12
AD0	AM14	AD17	AL9	/CBE2	AE10
AD1	AL14	AD18	AL8	/CBE3	AD11
AD2	AL13	AD19	AM8	/FRAME	AC16
AD3	AK13	AD20	AL7	/IRDY	AB15
AD4	AM12	AD21	AK6	/TRDY	AB16
AD5	AL12	AD22	AK7	/DEVSEL	AB14
AD6	AK12	AD23	AM7	/STOP	AC15
AD7	AJ14	AD24	AK5	/LOCK	AF13
AD8	AJ13	AD25	AL6	/PERR	AG14
AD9	AJ12	AD26	AK4	/SERR	AH11
AD10	AK11	AD27	AM6	PAR	AG13
AD11	AL11	AD28	AL5	/IDSEL	AD14
AD12	AM11	AD29	AM5	/INTA	AJ5

Ethernet Ports

The MimoKit motherboard features two independent Gigabit Ethernet ports. Each port consists of:

- An RJ45 connector with built-in link status LEDs
- A 10/100/1000 Mbps capable PHY
- The user back-end logic implementing the Ethernet MAC function in the FPGA2.

The two PHY ICs are connected to the FPGA2 through the MII/GMII interface as defined by the IEEE802.3-2005 standard. Table 11 shows the Ethernet-related signal assignments to the FPGA2 balls.

Table 11: Ethernet signal assignments to FPGA2

Port	MII/GMII Signal	Direction	FPGA2 Ball
Ethernet A	RxCLK	→	T32
	RxD0	→	M32
	RxD1	→	M31
	RxD2	→	M30
	RxD3	→	M29
	RxD4	→	L32
	RxD5	→	L31
	RxD6	→	L30
	RxD7	→	L29
	RxDV	→	N29
	RxER	→	N28
	RxCOL	→	P28
	RxCRS	→	P29
	GTxCLK	←	B18
	TxCLK	→	T30
	TxD0	←	N22
	TxD1	←	N23
	TxD2	←	N24
	TxD3	←	N25
	TxD4	←	N26
	TxD5	←	N27
	TxD6	←	P22
	TxD7	←	P23
	TxEN	←	N30
	TxER	←	N31
	MDC	←	P25

Port	MII/GMII Signal	Direction	FPGA2 Ball
	MDIO	↔	P24
	IRQ	→	P26
Ethernet B	RxCLK	→	U32
	RxD0	→	L26
	RxD1	→	L25
	RxD2	→	K27
	RxD3	→	K26
	RxD4	→	K25
	RxD5	→	K24
	RxD6	→	J27
	RxD7	→	J26
	RxDV	→	H28
	RxER	→	H27
	RxCOL	→	G29
	RxCRS	→	G30
	GTxCLK	←	D18
	TxCLK	→	U30
	TxD0	←	D31
	TxD1	←	D32
	TxD2	←	E29
	TxD3	←	E30
	TxD4	←	F29
	TxD5	←	F30
	TxD6	←	G27
	TxD7	←	G28
	TxEN	←	E31
	TxER	←	E32
	MDC	←	F32
	MDIO	↔	F31
	IRQ	→	G31

Please refer to Illustration 1 for an exact location of the Ethernet connectors on the board.

USB Host and Device

The MimoKit motherboard features two USB2 high speed (480 Mbps) ports, one host port and one device port. Both are built around the same USB transceiver and differ by the configuration of the transceiver and by the connector.

The selected USB transceiver is a Philips ISP1505. It is fully compliant with Universal Serial Bus Specification Rev. 2.0 and UTMI+ Low Pin Interface (ULPI) Specification Rev. 1.1. For more detailed information about the hardware configuration, please refer to the motherboard schematics.

The hardware of the motherboard is limited to the standard USB connectors, the transceivers (PHY layers), and a few passive components. The actual USB function is supposed to be implemented in the FPGA2 backend logic.

The connections between FPGA2 and the device and host transceivers are listed in Table 12.

Table 12: USB host and device signal assignments to FPGA2

Port	ULPI Signal	Function	Direction	FPGA2 Ball
Host port	CLK	ULPI clock	→	AK17
	D0	ULPI data bit 0	↔	AM28
	D1	ULPI data bit 1	↔	AC19
	D2	ULPI data bit 2	↔	AC20
	D3	ULPI data bit 3	↔	AD20
	D4	ULPI data bit 4	↔	AE20
	D5	ULPI data bit 5	↔	AE21
	D6	ULPI data bit 6	↔	AF21
	D7	ULPI data bit 7	↔	AG24
	NXT	ULPI Next signal	→	AB19
	STP	ULPI Stop signal	←	AD21
	DIR	ULPI Direction signal	→	AE22
	RST	Transceiver reset	←	AF22
Device port	CLK	ULPI clock	→	AL18
	D0	ULPI data bit 0	↔	AB20
	D1	ULPI data bit 1	↔	AB21
	D2	ULPI data bit 2	↔	AC21
	D3	ULPI data bit 3	↔	AD22
	D4	ULPI data bit 4	↔	AJ28
	D5	ULPI data bit 5	↔	AK28
	D6	ULPI data bit 6	↔	AK29
	D7	ULPI data bit 7	↔	AL29
	NXT	ULPI Next signal	→	AH19
	STP	ULPI Stop signal	←	AD18

Port	ULPI Signal	Function	Direction	FPGA2 Ball
	DIR	ULPI Direction signal	→	AC18
	RST	Transceiver reset	←	AC17

Please refer to Illustration 1 for an exact location of the USB host and device connectors on the board.

Serial Ports

The MimoKit motherboard features two RS-232-compatible ports with hardware flow control lines (RTS/CTS). Each serial port is connected to one FPGA through RS-232 level shifters. Serial A is connected to FPGA1, serial B is connected to FPGA2.

In order to save board area, the serial port connectors are 2.54mm pitch HE10 type, as is common on PC motherboards. HE10 to DB9 adapters are supplied with the board.

The assignment of the serial ports signals to the FPGA balls is described in Illustration 7. Table 13 shows the orientation of the connectors.

Illustration 7: Orientation of the HE10 serial ports connectors (J7 and J8)

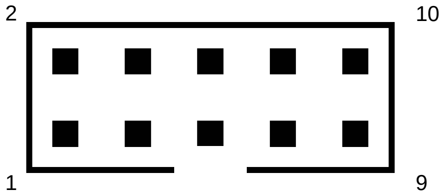


Table 13: Pin assignment of the serial ports connectors. Unlisted pins are unconnected.

Serial Port	RS232 Signal	HE10 Pin	Function	Direction	FPGA1 Ball	FPGA2 Ball
Serial A	TxD	3	Transmit data	←	D28	-
	RxD	5	Receive data	→	C28	-
	RTS	6	Request to send	←	E28	-
	CTS	4	Clear to send	→	H20	-
	GND	9	Signal ground	-	-	-
Serial B	TxD	3	Transmit data	←	-	AG10
	RxD	5	Receive data	→	-	AF12
	RTS	6	Request to send	←	-	AG11
	CTS	4	Clear to send	→	-	AG12
	GND	9	Signal ground	-	-	-

User LEDs

Four LEDs can be driven by the FPGA backend logic, two of them by FPGA1 (CR5 and CR6) and two by FPGA2 (CR3 and CR4). A LED is turned on by setting the driving signal at logic 1 (3.3 V typ.).

The location of the user LEDs is shown in Illustration 8. The signal assignments are listed in Table 14.

Illustration 8: Location of the user LEDs

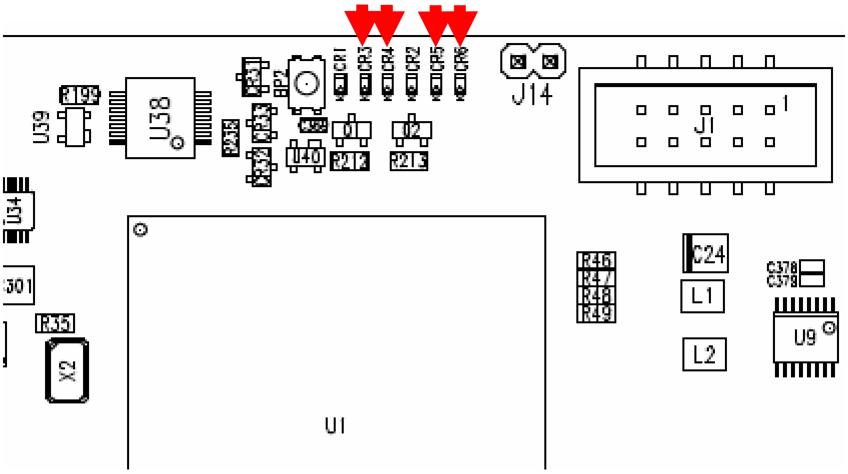


Table 14: User LEDs connection to the FPGAs

User LED	FPGA1 Ball	FPGA2 Ball
CR3	-	AJ8
CR4	-	AJ10
CR5	B29	-
CR6	C29	-

Analog Front-End

The MimoKit motherboard features three independent IQ-codec chips from Analog Devices (AD9861-80). All digital codec I/Os are connected to the FPGA1, while the analog I/Os are routed to the RF daughterboard connector.

Each IQ-codec provides the following features:

- Two matched 10-bits, 80 Msps analog to digital converters. In a typical MIMO WLAN application, these ADCs are used for the Baseband conversion of the Rx In-phase and Quadrature signals.

- Two matched 10-bits, 125 Msps digital to analog converters. In a typical MIMO WLAN application, these DACs are used for the Baseband conversion of the Tx In-phase and Quadrature signals

- A set of auxiliary ADC and DAC inputs. These converters can be conveniently used for RSSI measurement, Tx power measurement or to set the PA bias.

IQ-ADC

The IQ-ADC of the AD9861 converts the analog signals from the RF daughterboard into 10-bits digital data. On the MimoKit board the sampling rate is fixed at 80 Msps.

The IQ-ADC operates in differential voltage mode. For a detailed operation description of the IQ-ADC, please refer to the AD9861 datasheet, section Rx PATH BLOCK. The actual implementation with passive component values is available in the MimoKit motherboard schematic.

IQ-DAC

The IQ-DAC of the AD9861 converts 10-bits digital data coming from the FPGA1 into differential analog currents. On the MimoKit board, the sampling rate is fixed at 80 Msps.

It is important to note that the IQ-DAC operates in differential current mode. In a typical application the analog output of the IQ-DAC is connected to the Baseband input of an RF transceiver (on the RF daughterboard). As most common RF transceivers need a voltage mode input, a current-to-voltage circuitry (typically a simple network of matched resistors) is necessary. For performance reasons, the current-to-voltage conversion is not provided on the MimoKit motherboard. The best place to implement this passive network is close to the transceiver inputs, i.e. on the daughterboard.

Auxiliary Converters

The same analog pins of the auxiliary converters can be configured as ADC inputs or DAC outputs. The configuration is made through the AD9861 SPI port connected to the FPGA1. Please refer to the AD9861 datasheet for the exact usage of the AD9861 pins.

Codec to FPGA2 Connections

All the digital I/Os of the codec chips are routed to FPGA1 balls. Therefore, the detailed operation of the codec is controlled by the FPGA back-end logic. Although not mandatory, most MIMO WLAN applications will use the AD9861 in the so-called HD20 mode (i.e. half duplex, 10 bits for I plus 10 bits for Q). Other configurations are possible.

The Table 15 lists all the digital connections between the 3 IQ codec chips and the balls of the FPGA1.

Table 15: Connections between the IQ-codec chips and the FPGA1 balls

Codec #	Codec Pin	Function	Direction	FPGA2 Ball
Codec A	/RESET	Codec master reset	←	AE32
	L0	I signal bit 0 (LSB)	↔	AC30
	L1	I signal bit 1	↔	AC29
	L2	I signal bit 2	↔	AC26
	L3	I signal bit 3	↔	AC27
	L4	I signal bit 4	↔	AD26
	L5	I signal bit 5	↔	AD27
	L6	I signal bit 6	↔	AD29
	L7	I signal bit 7	↔	AD30
	L8	I signal bit 8	↔	AD31
	L9	I signal bit 9 (MSB)	↔	AE31
	U0	Q signal bit 0 (LSB)	↔	AG32
	U1	Qsignal bit 1	↔	AF32
	U2	Q signal bit 2	↔	AD32
	U3	Q signal bit 3	↔	AC31
	U4	Q signal bit 4	↔	AC32
	U5	Q signal bit 5	↔	U27
	U6	Q signal bit 6	↔	V28
	U7	Q signal bit 7	↔	V29
	U8	Q signal bit 8	↔	V30
	U9	Q signal bit 9 (MSB)	↔	V31
	IFACE1	Digital interface control	←	AH32
	IFACE2	Digital interface control	→	U1
	IFACE3	Digital interface control	→	U3
	RxPWRDWN	Rx path power down	←	AA22
	TxPWRDWN	Tx path power down	←	AA23
	SPI_CLK	SPI clock	←	U22
	SPI_CS	SPI chip select	←	W23
	SPI_DIO	SPI data	↔	V23
	AUX_SPI_CLK	Auxiliary SPI clock	←	Y31
	AUX_SPI_CS	Auxiliary SPI chip select	←	U23

Codec #	Codec Pin	Function	Direction	FPGA2 Ball
	AUX_SPI_SDO	Auxiliary SPI data out	→	W25
Codec B	/RESET	Codec master reset	←	D26
	L0	I signal bit 0 (LSB)	↔	M31
	L1	I signal bit 1	↔	L32
	L2	I signal bit 2	↔	H27
	L3	I signal bit 3	↔	J27
	L4	I signal bit 4	↔	K27
	L5	I signal bit 5	↔	J26
	L6	I signal bit 6	↔	K26
	L7	I signal bit 7	↔	H28
	L8	I signal bit 8	↔	L26
	L9	I signal bit 9 (MSB)	↔	L25
	U0	Q signal bit 0 (LSB)	↔	G27
	U1	Qsignal bit 1	↔	G28
	U2	Q signal bit 2	↔	G29
	U3	Q signal bit 3	↔	G30
	U4	Q signal bit 4	↔	F29
	U5	Q signal bit 5	↔	F30
	U6	Q signal bit 6	↔	E30
	U7	Q signal bit 7	↔	D32
	U8	Q signal bit 8	↔	K25
	U9	Q signal bit 9 (MSB)	↔	K24
	IFACE1	Digital interface control	←	K20
	IFACE2	Digital interface control	→	T1
	IFACE3	Digital interface control	→	T3
	RxPWRDWN	Rx path power down	←	H21
	TxPWRDWN	Tx path power down	←	G22
	SPI_CLK	SPI clock	←	E25
	SPI_CS	SPI chip select	←	E26
	SPI_DIO	SPI data	↔	G21
	AUX_SPI_CLK	Auxiliary SPI clock	←	J21
	AUX_SPI_CS	Auxiliary SPI chip select	←	D22
	AUX_SPI_SDO	Auxiliary SPI data out	→	E22
Codec C	/RESET	Codec master reset	←	B4
	L0	I signal bit 0 (LSB)	↔	C6
	L1	I signal bit 1	↔	C4
	L2	I signal bit 2	↔	C5
	L3	I signal bit 3	↔	E11
	L4	I signal bit 4	↔	D5

Codec #	Codec Pin	Function	Direction	FPGA2 Ball
	L5	I signal bit 5	↔	D6
	L6	I signal bit 6	↔	E5
	L7	I signal bit 7	↔	E6
	L8	I signal bit 8	↔	D7
	L9	I signal bit 9 (MSB)	↔	E7
	U0	Q signal bit 0 (LSB)	↔	G10
	U1	Qsignal bit 1	↔	F10
	U2	Q signal bit 2	↔	F12
	U3	Q signal bit 3	↔	G11
	U4	Q signal bit 4	↔	H11
	U5	Q signal bit 5	↔	J11
	U6	Q signal bit 6	↔	J12
	U7	Q signal bit 7	↔	K12
	U8	Q signal bit 8	↔	J13
	U9	Q signal bit 9 (MSB)	↔	J14
	IFACE1	Digital interface control	←	F9
	IFACE2	Digital interface control	→	A16
	IFACE3	Digital interface control	→	E16
	RxPWRDWN	Rx path power down	←	B7
	TxPWRDWN	Tx path power down	←	B8
	SPI_CLK	SPI clock	←	B11
	SPI_CS	SPI chip select	←	A7
	SPI_DIO	SPI data	↔	D12
	AUX_SPI_CLK	Auxiliary SPI clock	←	F8
	AUX_SPI_CS	Auxiliary SPI chip select	←	A6
	AUX_SPI_SDO	Auxiliary SPI data out	→	A8

RF Daughterboard Connector

The MimoKit motherboard will accept an RF daughterboard on the dedicated mezzanine connector J11. This connector transports the Baseband analog signals as well as the auxiliary analog signals, digital radio control lines and power supplies. The pin-out of the mezzanine connector is given in Table 16. The connector orientation is described in Illustration 9.

Illustration 9: RF daughterboard connector location and orientation

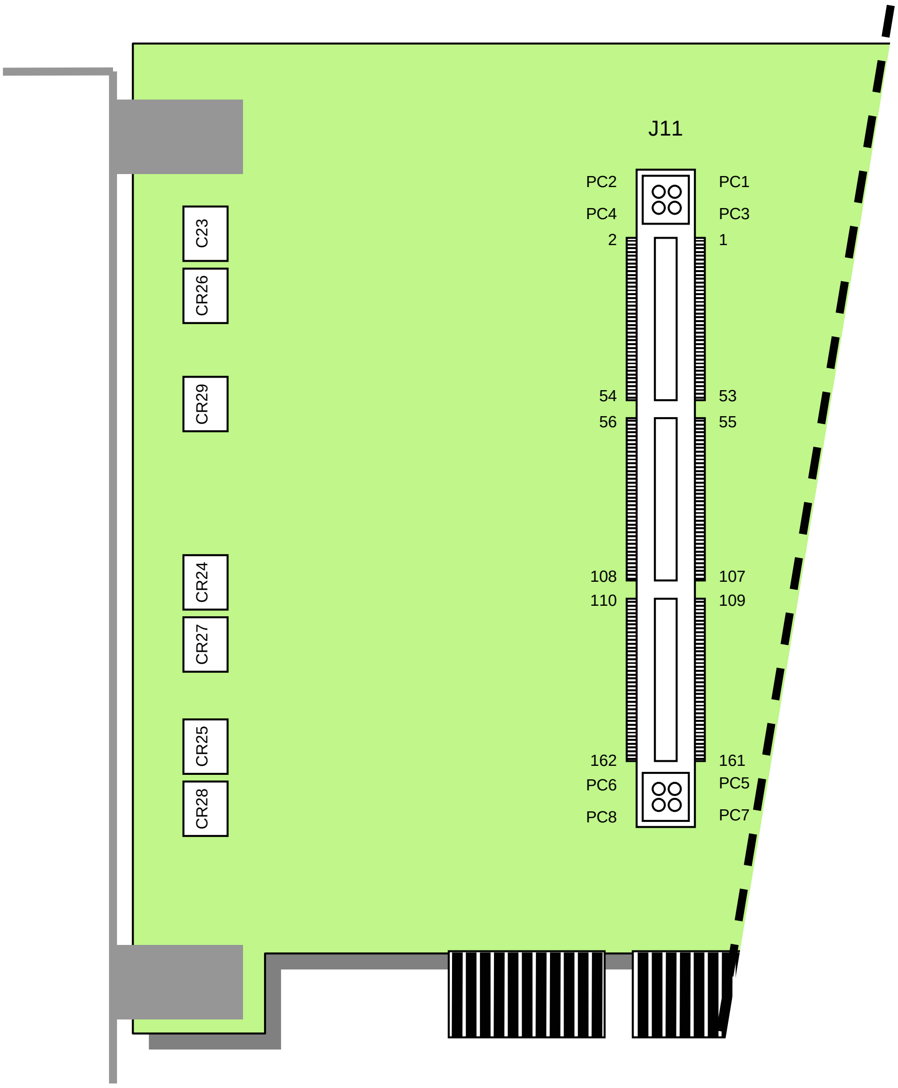


Table 16: RF daughterboard connector pinout

J11 Terminal	Function
PC1	5 V power supply to RF
PC2	12 V power supply to RF
PC3	Supply ground
PC4	Supply ground
1	Signal ground
2	Signal ground
3	IOUT+A of codec A (U27, pin 9). Positive part of In-phase Tx, current mode
4	IOUT+B of codec A (U27, pin 14). Positive part of Quadrature Tx, current mode
5	IOUT-A of codec A (U27, pin 8). Negative part of In-phase Tx, current mode
6	IOUT-B of codec A (U27, pin 15). Negative part of Quadrature Tx,

J11 Terminal	Function
	current mode
7	Signal ground
8	Signal ground
9	VIN+A of codec A (U27, pin 58). Positive part of In-phase Rx, voltage mode
10	VIN+B of codec A (U27, pin 54). Positive part of Quadrature Rx, voltage mode
11	VIN-A of codec A (U27, pin 57). Negative part of In-phase Rx, voltage mode
12	VIN-B of codec A (U27, pin 55). Negative part of Quadrature Rx, voltage mode
13	Signal ground
14	Signal ground
15	General purpose digital signal, connected to ball N22 of FPGA1
16	General purpose digital signal, connected to ball P31 of FPGA1
17	General purpose digital signal, connected to ball N23 of FPGA1
18	General purpose digital signal, connected to ball P32 of FPGA1
19	Signal ground
20	Signal ground
21	General purpose digital signal, connected to ball N24 of FPGA1
22	General purpose digital signal, connected to ball P27 of FPGA1
23	General purpose digital signal, connected to ball N25 of FPGA1
24	General purpose digital signal, connected to ball P25 of FPGA1
25	Signal ground
26	Signal ground
27	Signal ground
28	Signal ground
29	General purpose digital signal, connected to ball N26 of FPGA1
30	General purpose digital signal, connected to ball P24 of FPGA1
31	General purpose digital signal, connected to ball N27 of FPGA1
32	General purpose digital signal, connected to ball P23 of FPGA1
33	Signal ground
34	Signal ground
35	General purpose digital signal, connected to ball L29 of FPGA1
36	General purpose digital signal, connected to ball P22 of FPGA1
37	General purpose digital signal, connected to ball L30 of FPGA1
38	General purpose digital signal, connected to ball R22 of FPGA1
39	Signal ground
40	Signal ground
41	General purpose digital signal, connected to ball L31 of FPGA1
42	When set to 1, drives a green LED on the motherboard (CR26)

J11 Terminal	Function
43	General purpose digital signal, connected to ball P26 of FPGA1
44	General purpose digital signal, connected to ball U28 of FPGA1
45	Signal ground
46	Signal ground
47	General purpose digital signal, connected to ball P28 of FPGA1
48	Clock input from the RF board, connected to AK17 of FPGA1
49	General purpose digital signal, connected to ball P29 of FPGA1
50	Low frequency analog signal to/from the radio, e.g. an RSSI information. Connected to the auxiliary ADC/DAC inout of codec A (U27, pin 45) through a passive low-pass filter.
51	Signal ground
52	Signal ground
53	Shield (ground)
54	Shield (ground)
55	Signal ground
56	Signal ground
57	IOUT+A of codec B (U28, pin 9). Positive part of In-phase Tx, current mode
58	IOUT+B of codec B (U28, pin 14). Positive part of Quadrature Tx, current mode
59	IOUT-A of codec B (U28, pin 8). Negative part of In-phase Tx, current mode
60	IOUT-B of codec B (U28, pin 15). Negative part of Quadrature Tx, current mode
61	Signal ground
62	Signal ground
63	VIN+A of codec B (U28, pin 58). Positive part of In-phase Rx, voltage mode
64	VIN+B of codec B (U28, pin 54). Positive part of Quadrature Rx, voltage mode
65	VIN-A of codec B (U28, pin 57). Negative part of In-phase Rx, voltage mode
66	VIN-B of codec B (U28, pin 55). Negative part of Quadrature Rx, voltage mode
67	Signal ground
68	Signal ground
69	General purpose digital signal, connected to ball D21 of FPGA1
70	General purpose digital signal, connected to ball A27 of FPGA1
71	General purpose digital signal, connected to ball C21 of FPGA1
72	General purpose digital signal, connected to ball B27 of FPGA1
73	Signal ground
74	Signal ground
75	General purpose digital signal, connected to ball B21 of FPGA1

J11 Terminal	Function
76	General purpose digital signal, connected to ball A28 of FPGA1
77	General purpose digital signal, connected to ball D23 of FPGA1
78	General purpose digital signal, connected to ball C26 of FPGA1
79	Signal ground
80	Signal ground
81	Signal ground
82	Signal ground
83	General purpose digital signal, connected to ball A21 of FPGA1
84	General purpose digital signal, connected to ball B28 of FPGA1
85	General purpose digital signal, connected to ball B22 of FPGA1
86	General purpose digital signal, connected to ball C27 of FPGA1
87	Signal ground
88	Signal ground
89	General purpose digital signal, connected to ball A22 of FPGA1
90	General purpose digital signal, connected to ball D25 of FPGA1
91	General purpose digital signal, connected to ball C22 of FPGA1
92	General purpose digital signal, connected to ball D27 of FPGA1
93	Signal ground
94	Signal ground
95	General purpose digital signal, connected to ball C23 of FPGA1
96	When set to 1, drives a green LED on the motherboard (CR27)
97	General purpose digital signal, connected to ball C24 of FPGA1
98	General purpose digital signal, connected to ball D31 of FPGA1
99	Signal ground
100	Signal ground
101	General purpose digital signal, connected to ball C25 of FPGA1
102	Clock input from the RF board, connected to T32 of FPGA1
103	General purpose digital signal, connected to ball E24 of FPGA1
104	Low frequency analog signal to/from the radio, e.g. an RSSI information. Connected to the auxiliary ADC/DAC inout of codec B (U28, pin 45) through a passive low-pass filter.
105	Signal ground
106	Signal ground
107	Shield (ground)
108	Shield (ground)
109	Signal ground
110	Signal ground
111	IOUT+A of codec C (U30, pin 9). Positive part of In-phase Tx, current mode
112	IOUT+B of codec C (U30, pin 14). Positive part of Quadrature Tx, current mode

J11 Terminal	Function
113	IOUT-A of codec C (U30, pin 8). Negative part of In-phase Tx, current mode
114	IOUT-B of codec C (U30, pin 15). Negative part of Quadrature Tx, current mode
115	Signal ground
116	Signal ground
117	VIN+A of codec C (U30, pin 58). Positive part of In-phase Rx, voltage mode
118	VIN+B of codec C (U30, pin 54). Positive part of Quadrature Rx, voltage mode
119	VIN-A of codec C (U30, pin 57). Negative part of In-phase Rx, voltage mode
120	VIN-B of codec C (U30, pin 55). Negative part of Quadrature Rx, voltage mode
121	Signal ground
122	Signal ground
123	General purpose digital signal, connected to ball C12 of FPGA1
124	General purpose digital signal, connected to ball L15 of FPGA1
125	General purpose digital signal, connected to ball C7 of FPGA1
126	General purpose digital signal, connected to ball L13 of FPGA1
127	Signal ground
128	Signal ground
129	General purpose digital signal, connected to ball A5 of FPGA1
130	General purpose digital signal, connected to ball D8 of FPGA1
131	General purpose digital signal, connected to ball B6 of FPGA1
132	General purpose digital signal, connected to ball K14 of FPGA1
133	Signal ground
134	Signal ground
135	Signal ground
136	Signal ground
137	General purpose digital signal, connected to ball A4 of FPGA1
138	General purpose digital signal, connected to ball H12 of FPGA1
139	General purpose digital signal, connected to ball A11 of FPGA1
140	General purpose digital signal, connected to ball H13 of FPGA1
141	Signal ground
142	Signal ground
143	General purpose digital signal, connected to ball B12 of FPGA1
144	General purpose digital signal, connected to ball F14 of FPGA1
145	General purpose digital signal, connected to ball A12 of FPGA1
146	General purpose digital signal, connected to ball E19 of FPGA1
147	Signal ground

J11 Terminal	Function
148	Signal ground
149	General purpose digital signal, connected to ball E13 of FPGA1
150	When set to 1, drives a green LED on the motherboard (CR28)
151	General purpose digital signal, connected to ball F13 of FPGA1
152	General purpose digital signal, connected to ball D20 of FPGA1
153	Signal ground
154	Signal ground
155	General purpose digital signal, connected to ball F11 of FPGA1
156	Clock input from the RF board, connected to AK16 of FPGA1
157	General purpose digital signal, connected to ball B5 of FPGA1
158	Low frequency analog signal to/from the radio, e.g. an RSSI information. Connected to the auxiliary ADC/DAC inout of codec C (U30, pin 45) through a passive low-pass filter.
159	Signal ground
160	Signal ground
161	Shield (ground)
162	Shield (ground)