mikroprojekt

KONDOR AX

Advanced System Development Board

USER'S MANUAL

UM0026

Rev. 1.3

23.10.2015.

Table of contents

Revision History

Revision	Date	Author	Modification
1.0	02.09.2014.	BS	Initial
1.1	30.09.2015.	IP	Revamp
1.2	07.10.2015.	IP	Updates
1.3	23.10.2015	IP	Minor updates

Related Documents

ID	Code	Description
1	UM0027	KONDOR AX – Linux BSP Build Setup Guide
2	UM0028	KONDOR AX – Basic Demos Guide
3	UM0029	KONDOR AX – Basic Demos Reference Design Guide

1 Introduction

The KONDOR AX - Advanced System Development Board is targeted for demo, evaluation and development of RTL, hardware and software in wireless, wireline and computing applications. An ECP5UM FPGA and i.MX6 processor with numerous connectivity options can assist engineers with prototyping, developing and testing their specific designs. This user's guide describes revision B of the KONDOR AX - Advanced System Development Board featuring the ECP5 LFE5UM-85F-BG756 FPGA and Freescale i.MX6 Solo Processor.

1.1 Board Features

Table 1: Board features

Category	ECP5*	*i.MX6***
Description	**FPGA** • Lattice ECP5 • ECP5 LFE5UM-85F-BG756 • 84.000 LUTs • 207 Block RAMs • 156 18x18 Multipliers • 365 IO pins • 4 SERDES channels (In/out) • 400 MHz LPDDR3 Memory Support • 4 PLLs, 4 DLLs	**Freescale i.MX6 Solo ARM** • ARM® Cortex™-A9 1 GHz • 512 KB L2 cache • MIPI-HSI • PCIe 2.0 (1 lane) • Freescale PF100 power management unit • GPU 3D - Vivante GC880 • Video Decode | 1080p30 + D1 • Video Encode | 1080p30 H.264 BP/ Dual 720p encode
Programming options	• On-board USB JTAG interface • 10-pin JTAG header • Using i.MX6 processor	
Memory	• LPDDR3-1600 • EDF8132A1MC-GD-F • 800 MHz Clock Rate • **1x** 32-bit channel • 512 MB Memory	• 32-bit DDR3 memory • 64Mbit SPI Flash memory • eMMC memory • Micro SD Card
Connectivity options	• 2 SFP cages • FMC Connector • GPIO header • GPIO (differential) header • 10/100 Ethernet (RMII) • UART (using USB JTAG interface)	• HDMI output • Gigabit ethernet • OTG USB • USB serial interface • Camera interface
Other peripherals	• **4x** Single color LEDs • **4x** Dual color LEDs	• **6x** Status LEDs

Category	ECP5* *i.MX6***
Communication with on-board i.MX6 processor	• PCIe x1 interface • EIM interface • I2C, SPI, UART
Power Supply	• **12V** Input
Boot Options	• From 64 Mbit on-board SPI Flash memory • SPI Slave mode
Clocking	• Onboard 100MHz Differential Oscillator • Si5338 Clock Generator
Manufacturing	• RoHS Compliant

1.2 Board Layout

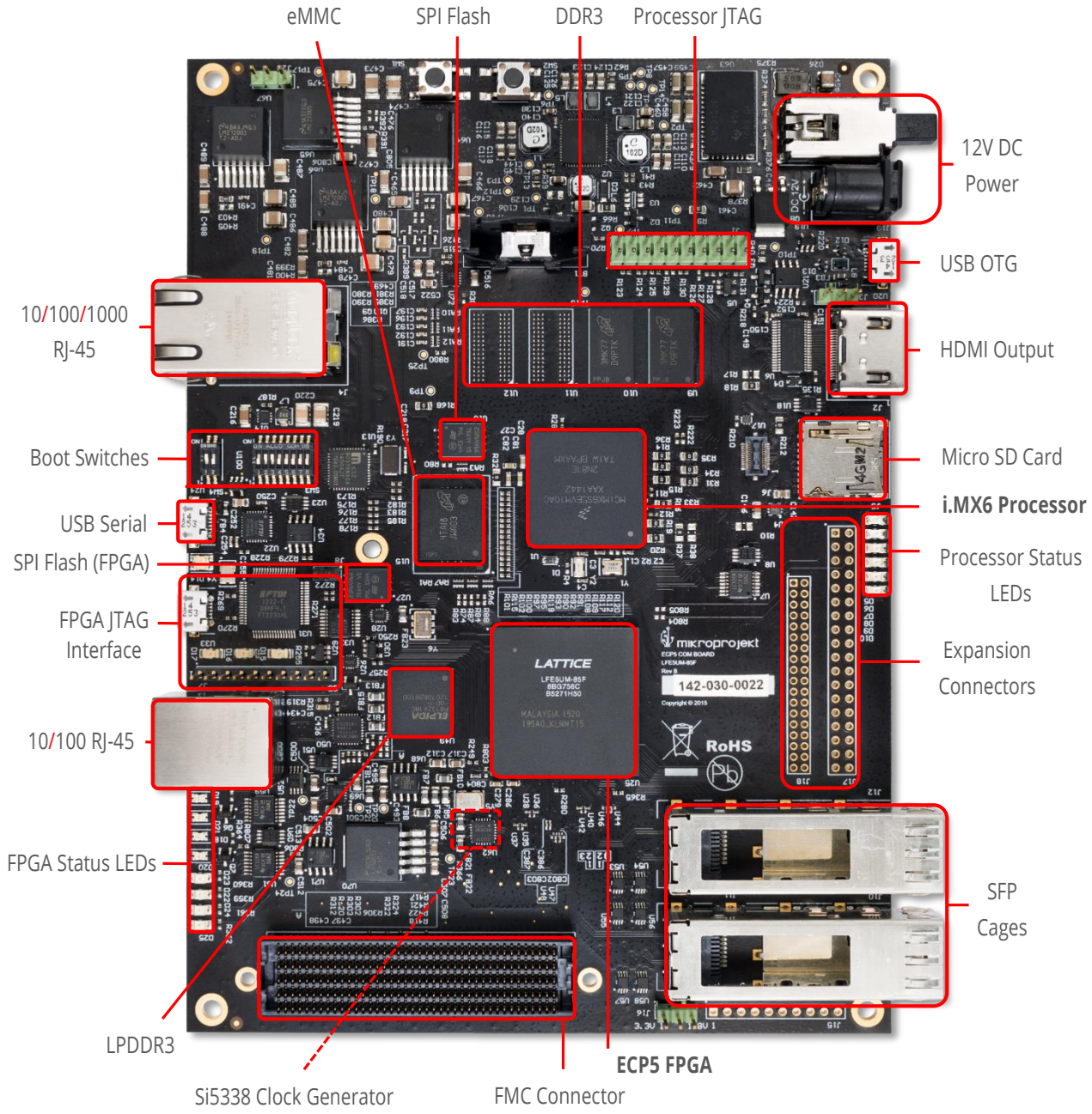

Figure 1. KONDOR AX - Advanced System Development Board

ATTENTION
OBSERVE PRECAUTIONS
ELECTOSTATIC
SENSITIVOS DEVICES

***Caution**: The KONDOR AX - Advanced System Development Board contains ESD-sensitive components. ESD safe practices should be followed while handling and using the evaluation board.*

1.3 Block schematic

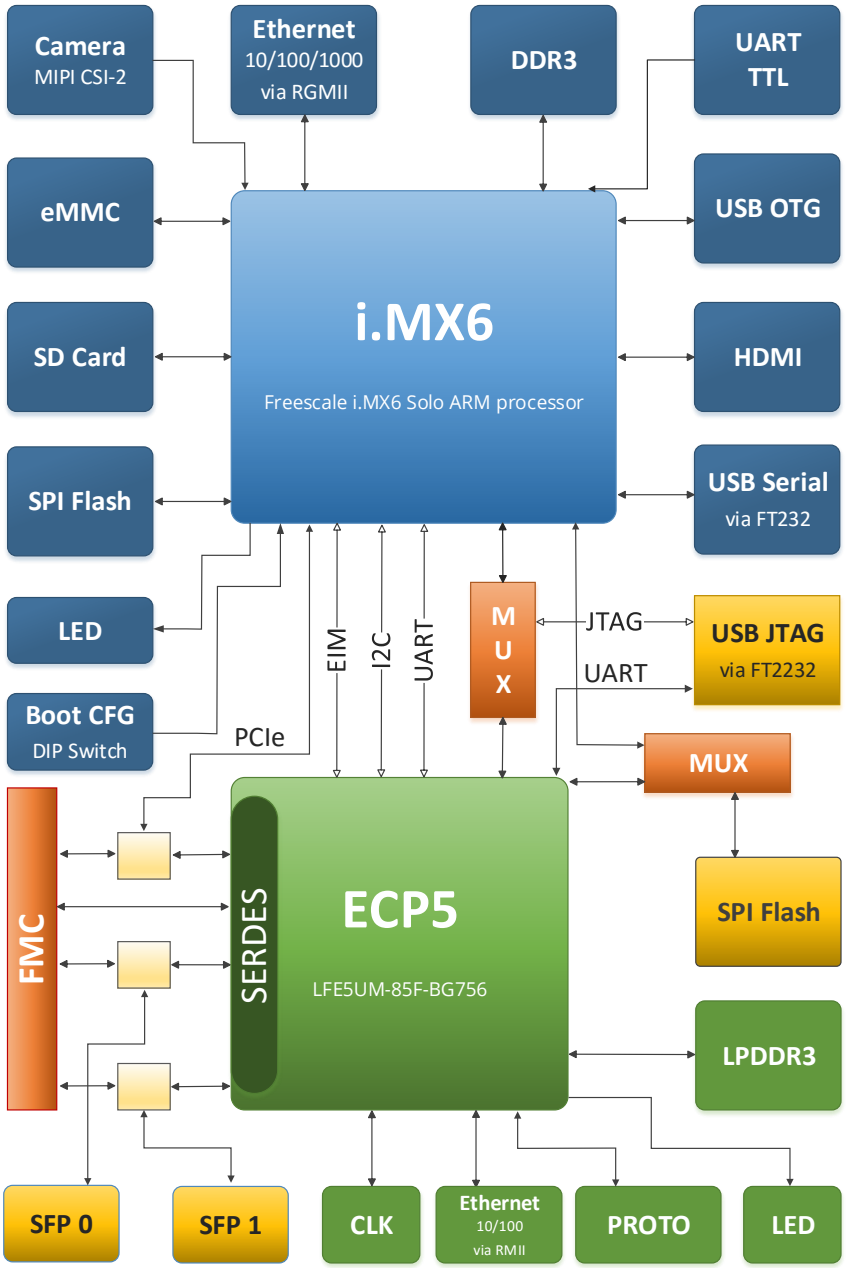

Figure 2. ECP5 COM Board Block Schematic

1.4 Applications

- HetNet & Small Cells
- Industrial IoT
- Cameras & Displays

2 Powering up the board

The KONDOR AX - Advanced System Development Board is ready to power on. To power on the board connect 12 V DC (1.5 A) power supply and then press S1 switch to apply power. A blue LED in the switch S1 indicate if the power is applied to the board. Table 2. shows all nominal voltages and the corresponding test points (please take a note that i.MX6 processor can dynamically adjust voltages depending on CPU load).

Table 2. Nominal voltages

Power	Voltage Range	Test Point
i.MX6		
MMPF0100 OTP		TP1
i.MX6 Core	1.35 V (Dynamic)	TP2
i.MX6 DDR3	1.5 V	TP3
i.MX6 SOC	1.35 V (Dynamic)	TP4
i.MX6 AUX	1.35 V ±5%	TP5
i.MX6 3.3V	3.3 V ±5%	TP6
i.MX6 5V	5 V ±5%	TP7
i.MX6 LDO2	1.5 V ±5%	TP8
i.MX6 LDO3	2.5 V ±5%	TP9
i.MX6 LDO4	1.8 V ±5%	TP10
i.MX6 LDO5	2.8 V ±5%	TP11
i.MX6 LDO6	3.3 V ±5%	TP12
i.MX6 NVS	3.0 V ±5%	TP13
i.MX6 SYS	4.2 V ±5%	TP14
DDR3 VTT	0.75 V	TP25
ECP5		
ECP5 Core	1.1 V ±5%	TP16
ECP5 3.3 V	3.3 V ±5%	TP17
ECP5 1.2 V	1.2 V ±5%	TP18
ECP5 2.5 V	2.5 V ±5%	TP19
ECP5 1.8 V	1.8 V ±5%	TP20
LPDDR3 REF	0.6 V	TP21
LPDDR3 VTT	0.6 V	TP22
ECP5 Serdes Core	1.1 V	TP23
ECP5 Serdes IO LDO	1.2 V	TP24

Default power sequencing ensures that the ECP5UM FPGA will be powered after the i.MX6 processor.

3 Hardware Instructions

Freescale i.MX6 ARM CPU supports many boot options. Here we are going to show the most often used scenario, where:

- Boot is in SPI flash
- Boot environment variables are in SPI flash
- (Start of MMC is reserved for optional boot and variables)
- Kernel is on MMC partition 1 formatted with FAT file system
- Linux main file system is on MMC partition 2 formatted with EXT3 file system

We will also show how to use SD card instead of MMC for kernel and main file system.

3.1 Flashing binaries to the board

Board is constructed in such way that it supports being programmed even when all memories are completely empty/new. We provide all the necessary software.

Preparation:

1. Turn Off power supply
2. Set 2-pole switch SW4 to 0 1 (programming mode)
3. Set 8-pole switch SW3 to 0 1 1 0 0 0 0 0 (boot from SPI flash, other settings not relevant now)
4. Connect USB cable to U24 micro USB connector and a Windows PC, - *this is for serial terminal printouts and commands*
5. Connect USB cable to U20 micro USB connector and a Windows PC, - *this is for downloading files into processors memory chips*
6. Prepare software package on the PC:
    - open cfg.ini file,
    - verify [LIST] section has entry "name = ecp5com mmc" (without quotation marks)

Individual switches on SW3 and SW4 are counted as 1 if they are closer to printed "ON" marking on the board. Printed "1" on the board means pin 1, out of 2 or 8 total pins.

Turn on power supply. In ***Control Panel -> Devices and printers***, a new device with name **Rigel** should appear. This is the ARM CPU signaling it is ready for software download.

Optionally, open **Teraterm** or similar serial terminal monitor utility. It enables monitoring progress of flash operations and later sending commands to the board.

3.2 Internal MMC memory

- Start **MfgTool2.exe**. It should detect a "***HID compliant device***". Press "***Start***" button. Wait 5-10 minutes until progress bar in MfgTool turns green and there is message "***Update complete***".
- If Windows asks do you want to format a new disk drive, press Cancel.
- Press "***Stop***" and "***Exit***", close the application. The programming of the board is complete.
- Turn off power supply.
- Set 2-pole switch SW 4 to 1 0 (operating mode).
- Turn on power supply. Terminal application should display boot messages, then kernel loading and finally Linux command line prompt. Log in as "root".

3.3 SD card

If, for any reason, you don't want to use internal MMC but SD card for Linux file system, that scenario is supported too.

To use SD card instead of internal MMC memory:

- Turn off power supply, insert micro SD card (8 GB cards are verified to work; generally any one from 2..32 GB should work depending on size of main file system)
- Repeat all the steps for flashing binaries to MMC, except in cfg.ini change [LIST] section entry into "name = ecp5com sd"

The flashing should last 10-20 minutes, depending on the speed of SD card.

3.4 Boot environment variables

To edit boot environment variables, press ***Enter*** key while boot is displaying the countdown.

"***help***" lists all the available commands.

"***printenv***" shows current environment variables.

set varname 'new contents' // sets varname

set varname // destroys a variable

"***saveenv***" saves changes permanently, always do it twice because rarely it doesn't work (on other boards, not verified yet on ECP5COM)

setenv bootcmd 'run bootcmd_mmc' // ... activates MMC, bootcmd_sd activates SD

"***reset***" resets the board nicely

"***Bad CRC, using default environment***" is normal message after the board was just programmed. Do "saveenv" to remote the warning.

3.5 Ethernet

By default the board is set to boot without enabling network.

Type:

```
> ifconfig eth0 up 192.168.2.222
```

to enable networing. This is just an example, you might need specify a different address depending on the configuration of your local network.

To use DHCP instead of a fixed address, edit bootargs_mmc or bootargs_sd boot variable and append "ip=dhcp".

3.6 Transferring files

To transfer files to the board, activate networking first. Then use:

- wget http://ip_address:port/folder/file.ext // ... to transfer file.ext to current folder on the board
- wget http://ip_address:port/folder/file.ext -O /local_folder/local_file.ext // ... to save the file with different path/name

For executable files, after transfer also do:

```
> chmod 755 file.ext
```

... or:

```
> chmod +x file.ext
```

If you are transferring kernel:

- cd /mnt
- mkdir temp
- mount /dev/mmcblk0p1 /mnt/temp // exact disk index depends whether MMC or SD is used
- cd temp
- mv ./uImage ./uImage-old
- wget http://server_address/uImage
- reboot

Server addresses in examples above refer to a machine where a web server, needed to serve files, is running. Any web server shall be usable, in practice Apache and Mongoose were verified to be compatible and work reliably.

Some of other options for transferring files to the board are:

- using SD card
- using mfg-tool and board in programming mode to do nothing but append new files to Linux file system

3.7 Tests

All test files are in /ecp5com folder. Some other functions present in the system are:

- HDMI shall work immediately and display a basic Gnome windowing system
- UART works if serial terminal works
- USB works if flashing binaries works
- USB OTG can be tested with a USB OTG cable and USB memory stick or mouse; more extensive support is work in progress
- LEDs (and CPU GPIO) can be tested using test_led.lxe

3.8 Downloading FPGA designs

While following chapters describe four possible ways to transfer designs to FPGA, it suffices to use only one of them, most suitable for desired use.

Direct download to FPGA is useful when having multiple versions under development.

For verified and stable designs, best choice is downloading them into SPI flash.

For designs under development or when you want to be able to switch between several designs in short time, best choice is using slave SPI.

JTAG method (from i.MX6) is supported for completeness and seems to have no benefits over slave SPI.

3.9 Downloading FPGA designs from PC using JTAG

3.9.1 To ECP5 FPGA

1. Connect micro USB cable to U33 connector on the board, install necessary USB/serial port drivers on the PC if required.
2. Start ***Lattice Diamond Programmer*** tool and select matching communication port for transferring data to the board.
3. ***Device Family*** should be ***ECP5UM***; ***Device*** should be ***LFE5UM-85F***.

4. Double click on the cell in ***Operation*** column to open ***Device properties*** dialog box. Set ***Access mode*** to ***JTAG 1532 Mode***; set ***Operation*** to ***Fast Program***.
5. Under ***Programming file***, select the desired ***.bit*** file containing FPGA design.
6. Close the dialog box by clicking ***OK***.
7. ***Menu Design → Program***.

3.9.2 To SPI flash

1. Connect micro USB cable to U33 connector on the board, install necessary USB/serial port drivers on the PC if required.
2. Start ***Lattice Diamond Programmer*** tool and select matching communication port for transferring data to the board.
3. ***Device Family*** should be ***ECP5UM***; ***Device*** should be ***LFE5UM-85F***.
4. Double click on the cell in ***Operation*** column to open ***Device properties*** dialog box. Set ***Access mode*** to ***SPI Flash Background Programming***; set ***Operation*** to ***SPI Flash Erase, Program, Verify***.
5. Under ***Programming file***, select the desired ***.bit*** file containing FPGA design. ***Family*** should be ***SPI Serial Flash***. ***Vendor*** is ***STMicro***; ***Device*** is ***SPI-M25P64***; ***Package*** is ***8-lead VDFPN8***.
6. ***Load from File*** start and end addresses.
7. Close the dialog box by clicking ***OK***.
8. ***Menu Design → Program***.

Note: since the design is saved into flash memory, it will "survive" board resets and shutdowns.

3.10 Generating files for embedded design

3.10.1 Using Slave SPI

Start ***Lattice Diamond Programmer*** tool. The USB cable is not required.

1. Set ***Device Family*** to ***ECP5UM***; set ***Device*** to ***LFE5UM-85F***.
2. Double click on the cell in ***Operation*** column to open ***Device properties*** dialog box. Set ***Access mode*** to ***Slave SPI Interface Programming***; set ***Operation*** to ***Slave SPI Fast Program***.
3. Select desired ***.bit*** file as usual.
4. ***File → Save Untitled As***, save an ***.xcf*** file into a desired folder.

Start ***Lattice Diamond Deployment Tool***.

1. ***Create New Deployment***.
2. Set ***Function Type*** to ***Embedded System***; set ***Output File Type*** to ***Slave SPI Embedded***. Click ***OK***.
3. Turn on the check mark ***Input XCF File***. Select previously saved ***.xcf*** file. Click ***Next***.
4. ***Compress embedded files*** should be selected. ***Next***.
5. Optionally change folder, and give recognizable names to algorithm and data file. ***Next***.
6. ***Generate***.

3.10.2 Using JTAG

Start ***Lattice Diamond Programmer*** tool. The USB cable is not required.

1. Set ***Device Family*** to ***ECP5UM***; set ***Device*** to ***LFE5UM-85F***.
2. Double click on the cell in ***Operation*** column to open ***Device properties*** dialog box. Set ***Access mode*** to ***JTAG 1532 Mode***; set ***Operation*** to ***Fast Program***.
3. Select desired ***.bit*** file as usual.
4. ***Menu File → Save Untitled As***, save an ***.xcf*** into a desired folder.

Start ***Lattice Diamond Deployment Tool***.

1. ***Create New Deployment***.
2. Set ***Function Type*** to ***Embedded System***; set ***Output File Type*** to ***JTAG Full VME Embedded***. Click ***OK***.
3. Turn on the check mark ***Input XCF File***. Select previously saved ***.xcf*** file. Click ***Next***.
4. ***Compress VME File*** and ***Include Header*** should be selected. ***Next***.
5. Optionally change folder, and give recognizable name to output file. ***Next***.
6. ***Generate***.

3.11 Downloading FPGA designs from i.MX6

3.11.1 Using Slave SPI

Connect a micro USB cable to U24 connector. Board should be recognized as virtual port. Start a terminal application, choose appropriate virtual port and baud rate 115200 bps.

Transfer algorithm (.sea) and data (.sed) file to the board - see Kondor Hardware Instruction Manual document for description of download procedure. Let's assume files are named spi_algo.sea and spi_data.sed.

Navigate to /ecp5com/demos/demo_* (* indicates number of demo) folder:

```
> cd /ecp5com/demos/demo_*
```

Launch SPI programming application:

```
> ../../sspiem.lxe spi_algo.sea spi_data.sed
```

Programming should last a couple of seconds.

Note: since the design is saved directly into FPGA, it will not "survive" shutdowns, but algorithm and data file will remain stored in file system. ARM CPU can be reset while preserving FPGA design by pressing reset switch on the board or telling it to:

```
> reboot
```

3.11.2 Using JTAG

Connect a micro USB cable to U24 connector. Board should be recognized as virtual port. Start a terminal application, choose appropriate virtual port and baud rate 115200 bps.

Transfer design file (.vme) to the board - see Kondor Hardware Instruction Manual document for description of download procedure. Let's assume file is named jtag.vme.

Navigate to /ecp5com/demos/demo_* (* indicates number of demo) folder:

```
> cd /ecp5com/demos/demo_*
```

Launch JTAG programming application:

```
> ../../ispvme.lxe jtag.vme
```

Programming should last about 3.5 minutes.

Note: since the design is saved directly into FPGA, it will not "survive" shutdowns, but algorithm and data file will remain stored in file system. ARM CPU can be reset while preserving FPGA design by pressing reset switch on the board or telling it to:

```
> reboot
```

-

4 ECP5

4.1 Configuration/Programming Headers

The ECP5 FPGA can be programmed using Diamond Programmer software with build-in USB download controller (FT2232) or external Lattice download cable using 10-pin 100mil header. To use build-in USB download controller, simply connect standard USB cable from micro USB connector (U33) to your PC. The PC will detect dual UART device, making the build-in cable available for use in Diamond Programmer software.

To use external Lattice download cable, jumper J8 has to be closed. This will disconnect the build-in USB download controller. When the build-in USB download cable is disconnected, the external Lattice download cable can be connected to 10-pin J7 header. The J7 header pinout is given in Table 3.

Table 3. JTAG Header Pinout (J7)

Pin	Function	Pin	Function
1	VCC/PWR	6	TMS
2	TDO	7	GND
3	TDI	8	TCK
4	PROG_N	9	DONE
5	NC	10	INIT_N

After power-up the ECP5 FPGA is in MSPI mode and it will try to configure itself from a SPI Flash memory (Micron M25P64-VME6TG device). The SPI Flash memory can be programed using easily via JTAG port.

Additionally, the i.MX6 processor can program the ECP5 FPGA using either JTAG mode or SSPI mode. While the i.MX6 processor is using JTAG mode to program the ECP5 FPGA, build-in USB download controller and the J7 header are disconnected from the FPGA. When the ECP5 FPGA is in SSPI mode, SPI flash memory (U27) is disconnected from the FPGA. Details how to configure and program FPGA using the i.MX6 processor can be found in **software manual**.

There are several LEDs on the board to indicate the ECP5 FPGA programming status. They are listed in Table 4.

Table 4. Programming status LEDs

LED	Color	Function
D15	Red	INITn
D16	Green	PROGRAMn
D17	Red	DONE

4.2 Clocking capabilities

The KONDOR AX - Advanced System Development Board provides several clock sources for the ECP5 device. A 100 MHz oscillator (Y6) with differential output provides clock for LPDDR3 memory interface. A Si5338 clock generator (U62) is used to provide two reference clocks for SERDES (for DCU0 and DCU1). The Si5338 device has 27 MHz clock input and it is capable of synthesizing any frequency on each of outputs and it is programmable via an I2C serial interface.

In case of DCU 0 dual, reference clock source can be either the Si5338 clock generator or FMC module. Switch between these two clock sources are done with U47 and U48 passive capacitor mux. When capacitors are mounted between pads 1 and 2 Si5338 clock generator is used as SERDES reference clock source. When capacitors are mounted between pads 2 and 3 FMC module is used as SERDES reference clock source. The I2C bus used for the Si5338 clock generator control is connected to the SYS_I2C_SCL and SYS_I2C_SDA signals. Connection of SYS_I2C bus is given in table below.

Table 5. SYS_I2C bus

Signal Name	ECP5 pin #
SYS_I2C_SCL	AK1
SYS_I2C_SDA	AH1

***Note**: PCIe reference clock is not connected to the ECP5 FPGA. When using PCIe interface the Si5338 clock generator has to be programmed to provide 100 MHz reference clock per PCIe specification. As consequence of this, clock spread should not be enabled on PCIe interface.*

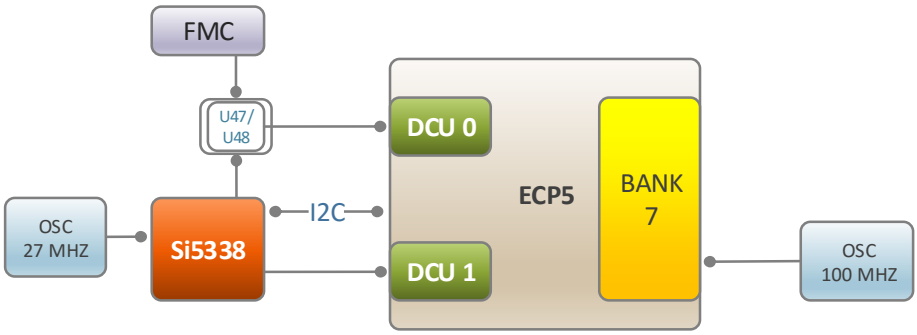

Figure 3. FPGA Clocking Scheme

4.3 SERDES

The ECP5 LFE5UM-85 features four SERDES channels arranged into two Duals. On the KONDOR AX - Advanced System Development Board following interfaces can be implemented using SERDES channels:

- *Single line PCIe*
- *Dual port CPRI*
- *Dual Port HiGig/SGMII/GigE*
- *Dual port JESD204B*

Due to limited number of available SerDes lanes not all interfaces can be implemented at same time. SerDes lanes used by each interface is given in Table 6.

Table 6. SerDes lanes used by interface

Interface	Used channels
PCIe	L0
CPRI	L2, L3
HiGig/SGMII/GigE	L2, L3
JESD204B	L0, L1

Selection of each interface to SerDes channels is done using passive capacitor (or resistor, if DC coupling is required) network. Each differential lane has two passive networks. List of passive networks for each differential lane, capacitors positions in the passive network, and selected interfaces for that position are given in Table 7.

Table 7. List of selected interfaces defined by capacitor position in passive network

Lane	Passive Network Designator	Capacitor position	Selected Interface
L0_TX	U38/U36	1-2	PCIe
L0_RX	U35/U37	1-2	PCIe
L0_TX	U38/U36	3-2	FMC connector
L0_RX	U35/U37	3-2	FMC connector
L1_TX	-	-	FMC connector
L1_RX	-	-	FMC Connector
L2_TX	U40/U42	1-2	SFP cage 0
L2_RX	U39/U41	1-2	SFP cage 0
L2_TX	U40/U42	3-2	FMC connector
L2_RX	U39/U41	3-2	FMC connector
L3_TX	U44/U46	1-2	SFP cage 1
L3_RX	U43/U45	1-2	SFP cage 1
L3_TX	U44/U46	3-2	FMC connector
L3_RX	U43/U45	3-2	FMC connector

Summary of possible interfaces usage combinations is given in Table 8.

Table 8. Summary of possible interfaces usage

Lane	Interface			
L0	PCIe	JESD204B	PCIe	JESD204B
L1	-	JESD204B	-	JESD204B
L2	CPRI	CPRI	HiGig/SGMII/GigE	HiGig/SGMII/GigE
L3	CPRI	CPRI	HiGig/SGMII/GigE	HiGig/SGMII/GigE

4.4 Ethernet interface

A KSZ8081RNB physical-layer transceiver, supporting 10Base-T/100Base-TX for transmission and reception of data over CAT-5 UTP cable, is included on the board. The RJ45 connector (J13) includes network magnetics providing the proper signal conditioning, electro-magnetic interference suppression and signal isolation. This connector includes two status LEDs (status LEDs configuration can be changed via MDC/MDIO interface). Reduced Media Independent Interface (RMII) is used for interface with the ECP5 FPGA. The ECP5 FPGA has to provide 50 MHz clock to the KSZ8081RNB transceiver. Table 9. Shows the pin connections of the interface between the ECP5 and the KSZ8081RNB.

Table 9. RMII interface pinout

Signal	ECP5 Pin #
RMII_RXD_0	B7
RMII_RXD_1	F8
RMII_RXDV	D7
RMII_RXER	A3
RMII_RXC	A4
RMII_TXD_0	A7
RMII_TXD_1	E8
RMII_TXEN	D8
RMII_CLK50	C17
RMII_MDC	A5
RMII_MDIO	C7
RMII_RST	A2
RMII_INTRP	C10

4.5 LPDDR3 Memory

On the KONDOR AX - Advanced System Development Board the ECP5 FPGA interfaces with a 32-bit wide LPDDR3 memory device (U49). Voltage reference for HSUL12 interfaces is generated with a TPS51200 Termination Regulator.

Table 10. List of signals associated with LPDDR3 memory

Signal	ECP5 Pin #	Signal	ECP5 Pin #
Address and Control			
LPDDR3_CK_P	C5	**LPDDR_CA2**	J6
LPDDR3_CK_N	D5	**LPDDR_CA3**	J7
LPDDR_CE0	K3	**LPDDR_CA4**	K6
LPDDR_CE1	L2	**LPDDR_CA5**	L4
LPDDR_CS0	J4	**LPDDR_CA6**	K7
LPDDR_CS1	H2	**LPDDR_CA7**	L7
LPDDR_ODT	K4	**LPDDR_CA8**	N4
LPDDR_CA0	H6	**LPDDR_CA9**	N7
LPDDR_CA1	H5		
DATA			
LPDDR_DQ0	R3	**LPDDR_DQ16**	Y1
LPDDR_DQ1	N1	**LPDDR_DQ17**	AD1
LPDDR_DQ2	W3	**LPDDR_DQ18**	AE3
LPDDR_DQ3	U1	**LPDDR_DQ19**	AE1
LPDDR_DQ4	P3	**LPDDR_DQ20**	AB4
LPDDR_DQ5	U3	**LPDDR_DQ21**	AB3
LPDDR_DQ6	T3	**LPDDR_DQ22**	AC1
LPDDR_DQ7	T1	**LPDDR_DQ23**	AD4

Signal	ECP5 Pin #	Signal	ECP5 Pin #
DATA			
LPDDR_DQS0_P	R1	LPDDR_DQS2_P	AC3
LPDDR_DQQ0_N	T2	LPDDR_DQQ2_N	AB2
LPDDR_DM0	P1	LPDDR_DM2	AB1
LPDDR_DQ8	AE4	LPDDR_DQ24	V7
LPDDR_DQ9	AD6	LPDDR_DQ25	U6
LPDDR_DQ10	Y4	LPDDR_DQ26	P5
LPDDR_DQ11	Y6	LPDDR_DQ27	R6
LPDDR_DQ12	AE5	LPDDR_DQ28	V6
LPDDR_DQ13	AC6	LPDDR_DQ29	U4
LPDDR_DQ14	W4	LPDDR_DQ30	T4
LPDDR_DQ15	AC7	LPDDR_DQ31	P4
LPDDR_DQS1_P	AB5	LPDDR_DQS3_P	R4
LPDDR_DQQ1_N	AB6	LPDDR_DQQ3_N	T5
LPDDR_DM1	Y5	LPDDR_DM3	T7

In order to improve immunity to SSO induced noise several IO pins in banks 6 and 7 are used as virtual VCC. In Table 11. are listed all pins that are used as virtual VCC pins.

Table 11. List of virtual VCC pins

ECP5 Pin #	Voltage	ECP5 Pin #	Voltage
T6	1.2 V	W5	1.2 V
U5	1.2 V	Y7	1.2 V
U2	1.2 V	F4	1.2 V
Y3	1.2 V	D2	1.2 V
V1	1.2 V	N6	1.2 V
W1	1.2 V	L6	1.2 V
AE2	1.2 V	K5	1.2 V
AC2	1.2 V	L3	1.2 V
AD3	1.2 V	L1	1.2 V
W2	1.2 V	H3	1.2 V
AD7	1.2 V	J3	1.2 V
AB7	1.2 V	E1	1.2 V

4.6 LEDs

The LEDS provided on the KONDOR AX - Advanced System Development Board are connected to general purpose FPGA I/Os. These LEDs provide status for user designs. The LEDs illuminate when the FPGA output is driven HIGH. Table 12. shows the LED and associated FPGA pins.

Table 12. LED Definitions

LED Number	ECP5 Pin #	PCB Designator	LED Color
LED0	J26	D18 (Dual LED)	Red
LED1	J27	D18 (Dual LED)	Green
LED2	K27	D19 (Dual LED)	Red
LED3	K26	D19 (Dual LED)	Green
LED4	A9	D20 (Dual LED)	Blue
LED5	F10	D20 (Dual LED)	Green
LED6	D10	D21 (Dual LED)	Blue
LED7	E10	D21 (Dual LED)	Green
LED8	AH3	D24	Amber
LED9	AK3	D23	Green
LED10	AL13	D22	Green
LED11	K30	D25	Amber

4.7 Prototyping area

The prototyping area provided on the KONDOR AX - Advanced System Development Board is consisted of two expansion connectors (not mounted). The first connector (J17) is two row 36 pin 100 mils (2.54 mm) header. It is intended to interface with 3.3V LVCMOS or LVTTL logic. On the header J17 I2C bus, 3.3 V power rail and ground signals are available, resulting with up to 28 signals that can be interfaced. Pinout of the J17 header is given in Table 13.

Table 13. Header J17 Pinout

Pin	Signal	ECP5 Pin #	Pin	Signal	ECP5 Pin #
1	GND	-	**19**	EXP_D12	H30
2	GND	-	**20**	EXP_D13	D29
3	3.3 V	-	**21**	GND	-
4	3.3 V	-	**22**	GND	-
5	SYS_I2C_SCL	AK1	**23**	EXP_D14	H31
6	SYS_I2C_SDA	AH1	**24**	EXP_D15	D32
7	EXP_D0	D30	**25**	EXP_D16	H32
8	EXP_D1	F28	**26**	EXP_D17	E29

Pin	Signal	ECP5 Pin #	Pin	Signal	ECP5 Pin #
9	EXP_D2	D31	27	EXP_D18	J32
10	EXP_D3	C29	28	EXP_D19	E32
11	EXP_D4	E30	29	EXP_D20	K31
12	EXP_D5	C30	30	EXP_D21	F29
13	EXP_D6	H27	31	EXP_D22	N29
14	EXP_D7	B32	32	EXP_D23	F32
15	EXP_D8	H28	33	EXP_D24	N30
16	EXP_D9	C31	34	EXP_D25	F31
17	EXP_D10	J29	35	EXP_D26	P27
18	EXP_D11	C32	36	EXP_D27	F30

Second expansion header (J18) is two row 36-pin 2 mm pitch header. It is intended to interface with high speed differential LVDS signals. Pinout of the J18 header is given in Table 14.

Table 14. Header J18 Pinout

Pin	Signal	ECP5 Pin #	Pin	Signal	ECP5 Pin #
1	GND	-	19	EXP_LVDS4_N	T28
2	GND	-	20	EXP_LVDS4_P	R29
3	EXP_LVDS0_N	P30	21	GND	-
4	EXP_LVDS0_P	P31	22	GND	-
5	GND	-	23	EXP_LVDS5_N	V29
6	GND	-	24	EXP_LVDS5_P	U29
7	EXP_LVDS1_N	P32	25	GND	-
8	EXP_LVDS1_P	N32	26	GND	-
9	GND	-	27	EXP_LVDS6_N	U28
10	GND	-	28	EXP_LVDS6_P	T29
11	EXP_LVDS2_N	T31	29	GND	-
12	EXP_LVDS2_P	R32	30	GND	-
13	GND	-	31	EXP_LVDS7_N	V26
14	GND	-	32	EXP_LVDS7_P	V27
15	EXP_LVDS3_N	U32	33	GND	-
16	EXP_LVDS3_P	T32	34	GND	-
17	GND	-	35	EXP_LVDS_CL_N	T26
18	GND	-	36	EXP_LVDS_CL_P	R26

4.8 FMC

A FPGA Mezzanine Card (FMC) connector is available on the KONDOR AX - Advanced System Development Board. Details and comments on connected signals are given in Table 15. Details how to set passive mux selectors in order to connect SERDES lanes to the FMC connector are given in section 0. VCC_FMC can be selected to 1.8 V or 3.3 V. To set VCC_FMC to 1.8 V pins 3 and 2 of a J16 header should be shorted. To set VCC_FMC to 3.3 V pins 1 and 2 of a J16 header should be shorted.

Table 15. FMC Connector Pinout

FMC Pin	FMC Port	Signal Name	ECP5 Pin #	Comment
A1	GND			
A2	DP1_M2C_P	FMC_RXCH1_P	AM11	
A3	DP1_M2C_N	FMC_RXCH1_N	AM12	
A4	GND			
A5	GND			
A6	DP2_M2C_P	FMC_RXCH2_P	AM17	Set U41 to select FMC
A7	DP2_M2C_N	FMC_RXCH2_N	AM18	Set U39 to select FMC
A8	GND			
A9	GND			
A10	DP3_M2C_P	FMC_RXCH3_P	AM20	Set U45 to select FMC
A11	DP3_M2C_N	FMC_RXCH3_N	AM21	Set U43 to select FMC
A12	GND			
A13	GND			
A14	DP4_M2C_P	-		
A15	DP4_M2C_N	-		
A16	GND			
A17	GND			
A18	DP5_M2C_P	-		
A19	DP5_M2C_N	-		
A20	GND			
A21	GND			
A22	DP1_C2M_P	FMC_TXCH1_P	AK12	
A23	DP1_C2M_N	FMC_TXCH1_N	AK13	
A24	GND			
A25	GND			
A26	DP2_C2M_P	FMC_TXCH2_P	AK18	Set U42 to select FMC
A27	DP2_C2M_N	FMC_TXCH2_N	AK19	Set U40 to select FMC
A28	GND			
A29	GND			
A30	DP3_C2M_P	FMC_TXCH3_P	AK21	Set U46 to select FMC
A31	DP3_C2M_N	FMC_TXCH3_N	AK22	Set U44 to select FMC
A32	GND			

FMC Pin	FMC Port	Signal Name	ECP5 Pin #	Comment
A33	GND			
A34	DP4_C2M_P	-		
A35	DP4_C2M_N	-		
A36	GND			
A37	GND			
A38	DP5_C2M_P	-		
A39	DP5_C2M_N	-		
A40	GND			
B1	RES1/CLK_DIR	-		
B2	GND			
B3	GND			
B4	DP9_M2C_P	-		
B5	DP9_M2C_N	-		
B6	GND			
B7	GND			
B8	DP8_M2C_P	-		
B9	DP8_M2C_N	-		
B10	GND			
B11	GND			
B12	DP7_M2C_P	-		
B13	DP7_M2C_N	-		
B14	GND			
B15	GND			
B16	DP6_M2C_P	-		
B17	DP6_M2C_N	-		
B18	GND			
B19	GND			
B20	GBTCLK1_M2C_P	-		
B21	GBTCLK1_M2C_N	-		
B22	GND			
B23	GND			
B24	DP9_C2M_P	-		
B25	DP9_C2M_N	-		
B26	GND			
B27	GND			
B28	DP8_C2M_P	-		
B29	DP8_C2M_N	-		
B30	GND			
B31	GND			
B32	DP7_C2M_P	-		
B33	DP7_C2M_N	-		
B34	GND			
B35	GND			
B36	DP6_C2M_P	-		
B37	DP6_C2M_N	-		
B38	GND			

FMC Pin	FMC Port	Signal Name	ECP5 Pin #	Comment
B39	GND			
B40	RES0	-		
C1	GND			
C2	DP0_C2M_P	FMC_TXCH0_P	AK9	Set U38 to select FMC
C3	DP0_C2M_N	FMC_TXCH0_N	AK10	Set U36 to select FMC
C4	GND			
C5	GND			
C6	DP0_M2C_P	FMC_RXCH0_P	AM8	Set U37 to select FMC
C7	DP0_M2C_N	FMC_RXCH0_N	AM9	Set U35 to select FMC
C8	GND			
C9	GND			
C10	LA06_P	-		
C11	LA06_N	-		
C12	GND			
C13	GND			
C14	LA10_P	FMC_LA10_P	K28	Routed as single ended lines via U53 voltage translator
C15	LA10_N	FMC_LA10_N	L27	
C16	GND			
C17	GND			
C18	LA14_P	FMC_LA14_P	N27	
C19	LA14_N	-		
C20	GND			
C21	GND			
C22	LA18_P_CC	-		
C23	LA18_N_CC	-		
C24	GND			
C25	GND			
C26	LA27_P	-		
C27	LA27_N	-		
C28	GND			
C29	GND			
C30	SCL	FMC_I2C_SCL	AJ30	
C31	SDA	FMC_I2C_SDA	AK29	
C32	GND			
C33	GND			
C34	GA0	FMC_GA0	-	Pull-up to 3.3V
C35	12P0V	VCC_12V		
C36	GND			
C37	12P0V	VCC_12V		
C38	GND			
C39	3P3V	VCC_3V3		
C40	GND			
D1	PG_C2M	FMC_PG_C2M	B8	
D2	GND			
D3	GND			
D4	GBTCLK0_M2C_P	FMC_GBTCLK0_M2C_P	AM14	Set U47 to select FMC

FMC Pin	FMC Port	Signal Name	ECP5 Pin #	Comment
D5	GBTCLK0_M2C_N	FMC_GBTCLK0_M2C_N	AM15	Set U48 to select FMC
D6	GND			
D7	GND			
D8	LA01_P_CC	FMC_LA1_P	AB30	Routed as differential LVDS pair
D9	LA01_N_CC	FMC_LA1_N	AB29	
D10	GND			
D11	LA05_P	FMC_LA5_P	AD27	Routed as differential LVDS pair
D12	LA05_N	FMC_LA5_N	AE27	
D13	GND			
D14	LA09_P	FMC_LA9_P	AG30	Routed as single ended lines via U54 voltage translator
D15	LA09_N	FMC_LA9_N	AK30	
D16	GND			
D17	LA13_P	FMC_LA13_P	P26	Routed as single ended
D18	LA13_N	FMC_LA13_N	N26	Routed as single ended
D19	GND			
D20	LA17_P_CC	-		
D21	LA17_N_CC	-		
D22	GND			
D23	LA23_P	-		
D24	LA23_N	-		
D25	GND			
D26	LA26_P	-		
D27	LA26_N	-		
D28	GND			
D29	TCK	FMC_TCK	C8	
D30	TDI	FMC_TDI	A8	
D31	TDO	FMC_TDO	F9	
D32	3P3VAUX	VCC_3V3		
D33	TMS	FMC_TMS	C9	
D34	TRST_L	FMC_TRSTn	D9	
D35	GA1	FMC_GA1		Pull-down resistor
D36	3P3V	VCC_3V3		
D37	GND			
D38	3P3V	VCC_3V3		
D39	GND			
D40	3P3V	VCC_3V3		
E1	GND			
E2	HA01_P_CC	-		
E3	HA01_N_CC	-		
E4	GND			
E5	GND			
E6	HA05_P	-		
E7	HA05_N	-		
E8	GND			
E9	HA09_P	-		
E10	HA09_N	-		

FMC Pin	FMC Port	Signal Name	ECP5 Pin #	Comment
E11	GND			
E12	HA13_P	-		
E13	HA13_N	-		
E14	GND			
E15	HA16_P	-		
E16	HA16_N	-		
E17	GND			
E18	HA20_P	FMC_HA20_P	K32	
E19	HA20_N	-		
E20	GND			
E21	HB03_P	-		
E22	HB03_N	-		
E23	GND			
E24	HB05_P	-		
E25	HB05_N	-		
E26	GND			
E27	HB09_P	-		
E28	HB09_N	-		
E29	GND			
E30	HB13_P	-		
E31	HB13_N	-		
E32	GND			
E33	HB19_P	-		
E34	HB19_N	-		
E35	GND			
E36	HB21_P	-		
E37	HB21_N	-		
E38	GND			
E39	VADJ	VCC_FMC		
E40	GND			
F1	PG_M2C	FMC_PG_M2C	AM2	
F2	GND			
F3	GND			
F4	HA00_P_CC	-		
F5	HA00_N_CC	-		
F6	GND			
F7	HA04_P	-		
F8	HA04_N	-		
F9	GND			
F10	HA08_P	FMC_HA08_P	W31	Routed as differential LVDS pair
F11	HA08_N	FMC_HA08_N	Y32	
F12	GND			
F13	HA12_P	-		
F14	HA12_N	-		
F15	GND			
F16	HA15_P	-		

FMC Pin	FMC Port	Signal Name	ECP5 Pin #	Comment
F17	HA15_N	-		
F18	GND			
F19	HA19_P	FMC_HA19_P	AJ29	Routed as single ended lines via U58 voltage translator
F20	HA19_N	FMC_HA19_N	AK28	
F21	GND			
F22	HB02_P	-		
F23	HB02_N	-		
F24	GND			
F25	HB04_P	-		
F26	HB04_N	-		
F27	GND			
F28	HB08_P	-		
F29	HB08_N	-		
F30	GND			
F31	HB12_P	-		
F32	HB12_N	-		
F33	GND			
F34	HB16_P	-		
F35	HB16_N	-		
F36	GND			
F37	HB20_P	-		
F38	HB20_N	-		
F39	GND			
F40	VADJ	VCC_FMC		
G1	GND			
G2	CLK1_C2M_P	-		
G3	CLK1_C2M_N	-		
G4	GND			
G5	GND			
G6	LA00_P_CC	FMC_CLK_LA0_P	AD32	Routed as differential LVDS pair
G7	LA00_N_CC	FMC_CLK_LA0_N	AE32	
G8	GND			
G9	LA03_P	FMC_SYSREF_LA3_P	AC30	Routed as differential LVDS pair
G10	LA03_N	FMC_SYSREF_LA3_N	AB31	
G11	GND			
G12	LA08_P	FMC_SYNC_LA8_P	AB28	Routed as differential LVDS pair
G13	LA08_N	FMC_SYNC_LA8_N	AB27	
G14	GND			
G15	LA12_P	FMC_LA12_P	L29	Routed as single ended lines via U55 voltage translator
G16	LA12_N	FMC_LA12_N	L26	
G17	GND			
G18	LA16_P	-		
G19	LA16_N	-		
G20	GND			
G21	LA20_P	-		
G22	LA20_N	-		

FMC Pin	FMC Port	Signal Name	ECP5 Pin #	Comment
G23	GND			
G24	LA22_P	-		
G25	LA22_N	-		
G26	GND			
G27	LA25_P	-		
G28	LA25_N	-		
G29	GND			
G30	LA29_P	-		
G31	LA29_N	-		
G32	GND			
G33	LA31_P	-		
G34	LA31_N	-		
G35	GND			
G36	LA33_P	-		
G37	LA33_N	-		
G38	GND			
G39	VADJ	VCC_FMC		
G40	GND			
H1	VREF_A_M2C	-		
H2	PRSNT_M2C_L	FMC_PRSTN_M2C	AL1	
H3	GND			
H4	CLK0_M2C_P	-		
H5	CLK0_M2C_N	-		
H6	GND			
H7	LA02_P	FMC_SFP2_SCL	F5	I2C Voltage Translator used (U52)
H8	LA02_N	FMC_SFP2_SDA	B1	
H9	GND			
H10	LA04_P	FMC_SFP3_SCL	D3	I2C Voltage Translator used (U51)
H11	LA04_N	FMC_SFP3_SDA	C2	
H12	GND			
H13	LA07_P	-		
H14	LA07_N	-		
H15	GND			
H16	LA11_P	FMC_LA11_P	L30	Routed as single ended lines via U57 voltage translator
H17	LA11_N	FMC_LA11_N	K29	
H18	GND			
H19	LA15_P	-		
H20	LA15_N	-		
H21	GND			
H22	LA19_P	-		
H23	LA19_N	-		
H24	GND			
H25	LA21_P	-		
H26	LA21_N	-		
H27	GND			
H28	LA24_P	-		

FMC Pin	FMC Port	Signal Name	ECP5 Pin #	Comment
H29	LA24_N	-		
H30	GND			
H31	LA28_P	-		
H32	LA28_N	-		
H33	GND			
H34	LA30_P	-		
H35	LA30_N	-		
H36	GND			
H37	LA32_P	-		
H38	LA32_N	-		
H39	GND			
H40	VADJ	VCC_FMC		
J1	GND			
J2	CLK3_BIDIR_P	FMC_CLK3_P	R27	Routed as differential LVDS pair
J3	CLK3_BIDIR_N	FMC_CLK3_N	T27	
J4	GND			
J5	GND			
J6	HA03_P	-		
J7	HA03_N	-		
J8	GND			
J9	HA07_P	-		
J10	HA07_N	-		
J11	GND			
J12	HA11_P	-		
J13	HA11_N	-		
J14	GND			
J15	HA14_P	-		
J16	HA14_N	-		
J17	GND			
J18	HA18_P	-		
J19	HA18_N	-		
J20	GND			
J21	HA22_P	-		
J22	HA22_N	-		
J23	GND			
J24	HB01_P	-		
J25	HB01_N	-		
J26	GND			
J27	HB07_P	-		
J28	HB07_N	-		
J29	GND			
J30	HB11_P	-		
J31	HB11_N	-		
J32	GND			
J33	HB15_P	-		
J34	HB15_N	-		

FMC Pin	FMC Port	Signal Name	ECP5 Pin #	Comment
J35	GND			
J36	HB18_P	-		
J37	HB18_N	-		
J38	GND			
J39	VIO_B_M2C	-		
J40	GND			
K1	VREF_B_M2C	-		
K2	GND			
K3	GND			
K4	CLK1_M2C_P	FMC_CLK1_P	P28	Routed as differential LVDS pair
K5	CLK1_M2C_N	FMC_CLK1_N	P29	
K6	GND			
K7	HA02_P	-		
K8	HA02_N	-		
K9	GND			
K10	HA06_P	-		
K11	HA06_N	-		
K12	GND			
K13	HA10_P	-		
K14	HA10_N	-		
K15	GND			
K16	HA17_P_CC	-		
K17	HA17_N_CC	-		
K18	GND			
K19	HA21_P	FMC_HA21_P	L32	Routed as single ended line via U56 voltage translator
K20	HA21_N	-		
K21	GND			
K22	HA23_P	FMC_HA23_P	L31	Routed as single ended line via U56 voltage translator
K23	HA23_N	-		
K24	GND			
K25	HB00_P_CC	-		
K26	HB00_N_CC	-		
K27	GND			
K28	HB06_P_CC	FMC_HB06_P	Y26	Routed as differential LVDS pair
K29	HB06_N_CC	FMC_HB06_N	Y27	
K30	GND			
K31	HB10_P	-		
K32	HB10_N	-		
K33	GND			
K34	HB14_P	-		
K35	HB14_N	-		
K36	GND			
K37	HB17_P_CC	FMC_HB17_P	Y29	Routed as differential LVDS pair
K38	HB17_N_CC	FMC_HB17_N	W29	
K39	GND			
K40	VIO_B_M2C	-		

4.9 SFP Connectors

The KONDOR AX - Advanced System Development Board features two SFP Connectors. First SFP connector (J9) is connected to the channel 0 of DCU 1, while second SFP connector (J11) is connected to the channel 1 of DCU 1. Details how to set passive mux selectors in order to connect SERDES lanes to the SFP connectors are given in section 0. Complete pinout of J9 SFP connector is given in Table 16.

Table 16. J8 SFP connector pinout

SFP Pin	Signal Name	ECP5 Pin #	Comment
1	GND		
2	SFP2_TX_FAULT	AK31	Pull-up resistor
3	SFP2_TX_DISABLE	AG32	
4	SFP2_MODE_DEF2	AK32	Pull-up resistor
5	SFP2_MODE_DEF1	AJ31	Pull-up resistor
6	SFP2_MODE_DEF0	AJ32	Pull-up resistor
7	SFP2_RATE_SEL	AH32	Pull-up resistor
8	SFP2_LOS	AH30	Pull-up resistor
9	GND		
10	GND		
11	GND		
12	SFP2_RX_N	AM18	Set U39 to select J9 SFP
13	SFP2_RX_P	AM17	Set U41 to select J9 SFP
14	GND		
15	VCC 3.3V		
16	VCC 3.3V		
17	GND		
18	SFP2_TX_P	AK18	Set U42 to select J9 SFP
19	SFP2_TX_N	AK19	Set U40 to select J9 SFP
20	GND		

Complete pinout of J11 SFP connector is given in Table 17.

Table 17. J11 SFP connector pinout

SFP Pin	Signal Name	ECP5 Pin #	Comment
1	GND		
2	SFP3_TX_FAULT	AM28	Pull-up resistor
3	SFP3_TX_DISABLE	AL32	
4	SFP3_MODE_DEF2	AL28	Pull-up resistor
5	SFP3_MODE_DEF1	AM29	Pull-up resistor
6	SFP3_MODE_DEF0	AM30	Pull-up resistor
7	SFP3_RATE_SEL	AL30	Pull-up resistor

SFP Pin	Signal Name	ECP5 Pin #	Comment
8	SFP3_LOS	AM31	Pull-up resistor
9	GND		
10	GND		
11	GND		
12	SFP3_RX_N	AM21	Set U33 to select J11 SFP
13	SFP3_RX_P	AM20	Set U45 to select J11 SFP
14	GND		
15	VCC 3.3V		
16	VCC 3.3V		
17	GND		
18	SFP3_TX_P	AK21	Set U46 to select J11 SFP
19	SFP3_TX_N	AK22	Set U44 to select J11 SFP
20	GND		

4.10 Interfaces with i.MX6 processor

The ECP5UM FPGA can interface with the i.MX6 processor using one of several links available on the board. Available links between the FPGA and the i.MX6 processor are: PCIe, EIM, SPI, UART and I2C.

4.10.1 PCIe

PCIe x1 link is implemented between the FPGA and the i.MX6 processor, as the i.MX6 processor supports only PCIe x1 link. For PCIe link implementation channel 0 of DCU 0 is used. Details how to set passive mux selectors in order use PCIe link are given in section 0. As only RX and TX lanes are connected between the FPGA and i.MX6 processor, the Si5338 clock generator has to provide 100 MHz reference clock to the FPGA.

Table 18. PCIe pinout

Signal Name	ECP5 Pin #	Comment
iMX6_PCIE_RX_N	AK10	Set U25 to select PCIe
iMX6_PCIE_RX_P	AK9	Set U27 to select PCIe
iMX6_PCIE_TX_N	AM9	Set U36 to select PCIe
iMX6_PCIE_TX_P	AM8	Set U38 to select PCIe

4.10.2 EIM

When PCIe link is not available, the EIM interface can be used to provide link between the FPGA and the processor. On the KONDOR AX - Advanced System Development Board 32-bit wide EIM interface is implemented. Complete pinout of EIM interface is given in Table 19.

Table 19. EIM pinout

Signal Name	ECP5 Pin #	Signal Name	ECP5 Pin #
EIM_D0	F24	**EIM_WAIT**	A15
EIM_D1	A28	**EIM_BCLK**	A18
EIM_D2	F25	**EIM_DA0**	E19
EIM_D3	B26	**EIM_DA1**	B19
EIM_D4	E25	**EIM_DA2**	C20
EIM_D5	A26	**EIM_DA3**	D20
EIM_D6	C23	**EIM_DA4**	E11
EIM_D7	C26	**EIM_DA5**	C13
EIM_D8	F23	**EIM_DA6**	A10
EIM_D9	D25	**EIM_DA7**	B16
EIM_D10	B25	**EIM_DA8**	A14
EIM_D11	C25	**EIM_DA9**	D13
EIM_D12	C24	**EIM_DA10**	F16
EIM_D13	E23	**EIM_DA11**	C14
EIM_D14	A25	**EIM_DA12**	A16
EIM_D15	A24	**EIM_DA13**	B14
EIM_D16	D22	**EIM_DA14**	B11
EIM_D17	C18	**EIM_DA15**	A13
EIM_D18	C16	**EIM_A16**	D11
EIM_D19	B22	**EIM_A17**	A11
EIM_D20	F18	**EIM_A18**	C19
EIM_D21	D19	**EIM_CS0**	D23
EIM_D22	D16	**EIM_CS1**	D24
EIM_D23	A20	**EIM_LBA**	B10
EIM_D24	C22	**EIM_OE**	A17
EIM_D25	A23	**EIM_RW**	C11
EIM_D26	A22		
EIM_D27	A19		
EIM_D28	B23		
EIM_D29	F19		
EIM_D30	E22		
EIM_D31	F22		
EIM_EB0	D15		
EIM_EB1	F11		
EIM_EB2	F14		
EIM_EB3	F13		
EIM_A19	D18		
EIM_A20	D14		

Signal Name	ECP5 Pin #	Signal Name	ECP5 Pin #
EIM_A21	E14		
EIM_A22	E16		
EIM_A23	C15		
EIM_A24	F15		
EIM_A25	F20		

4.10.3 SPI, UART, I2C

Several low bandwidth interfaces between the ECP5UM FPGA and the i.MX6 processor are provided on the KONDOR AX - Advanced System Development Board. Following interfaces are implemented: SPI, UART and I2C. Pinout of low bandwidth interfaces is given in Table 20.

Table 20. SPI, UART and I2C pinout

Signal name	ECP5 Pin #	Interface
ECSPI1_MOSI	A31	SPI
ECSPI1_SCLK	A30	SPI
ECSPI1_CS0	A29	SPI
ECSPI1_MISO	D26	SPI
I2C4_SDA	AH28	I2C
I2C4_SCL	AJ28	I2C
UART4_RX	AG29	UART
UART4_TX	AG28	UART

5 i.MX6

The KONDOR AX - Advanced System Development Board features an i.MX6Solo processor.

5.1 Boot

A MMPF0100 Power Manager ensures proper power sequencing for the i.MX6 processor. The MMPF0100 will power up board after power is applied to the board. During boot process the i.MX6 processor execute ROM that will check boot configuration. User can change boot configuration using SW3 and SW4 DIP switches. When DIP switches are in ON position, corresponding signal is in logic 1, otherwise it is in logic 0.

The SW4 DIP switch is used to set MX6_BOOT_MODE0 (position 2) and MX6_BOOT_MODE1 (position 1) signals. Possible combinations are given in Table 21.

Table 21. SW4 DIP switches combinations

Signal Name		Selected Boot Configuration
MX6_BOOT_MODE1	MX6_BOOT_MODE0	Boot from fuses
0	0	Boot from fuses
0	1	Serial Downloader
1	0	Internal Boot
1	1	Reserved

When boot from fuses is selected, the i.MX6 processor will boot from one of available boot sources: SPI Flash memory, SD card or eMMC memory. Selection and configuration of boot sources are done with SW3 DIP switches. List of signals set with SW3 DIP switch and corresponding positions are given in Table 22. In Table 23. are given SW4 DIP switch positions for proper configuration of supported boot sources. For more details on boot configuration one can consult an i.MX6 processor Reference Manual. When booting from fuses, power sequencing must ensure that the ECP5UM is powered after boot signals has been latched by the i.MX6 processor. This is to avoid possibility that the FPGA could change boot configuration signals that are shared with EIM interface.

Table 22. SW3 DIP switch positions and corresponding signal names

DIP position	Signal Name
1	MX6_BOOT_CFG1_6
2	MX6_BOOT_CFG1_5
3	MX6_BOOT_CFG1_4
4	MX6_BOOT_CFG2_3
5	MX6_BOOT_CFG2_4
6	MX6_BOOT_CFG2_5
7	MX6_BOOT_CFG2_6
8	MX6_BOOT_CFG2_7

Table 23. Selection and configuration of boot sources

DIP Switch Position Settings								Selected Interface
1	2	3	4	5	6	7	8	
0	1	1	x	x	x	x	x	SPI Flash
1	0	x	1	0	0	0	0	SD Card
1	1	x	1	1	0	1	0	eMMC

If serial downloader is selected as boot configuration, the i.MX6 processor will try to boot from USB OTG.

5.2 Debug

For debug purposes a 20-pin 100mil header J1 is connected to JTAG interface of the i.MX6 processor. Pinout of the J1 header is listed in Table 24.

Table 24. i.MX6 JTAG header pinout

Pin	Function	Pin	Function
1	VCC/PWR	11	VCC
2	TRSTn	12	GND
3	TDI	13	GND
4	TMS	14	GND
5	TCK	15	GND
6	RTCK	16	GND
7	TDO	17	GND
8	nSRST	18	GND
9	DE	19	GND
10	DACK	20	GND

5.3 SPI Flash

A M25P64 (U16) 64Mbit SPI Flash memory is connected to the i.MX6 processor. The M25P64 memory is connected to an ECSPI-3 controller. The SPI Flash memory can be used as boot source.

5.4 eMMC

For software storage an eMMC memory (U15) is provided on the KONDOR AX - Advanced System Development Board. The eMMC memory is connected to a USDHC-4 controller, and it can be used as boot source.

5.5 SD Card

If additional storage is required, a SD Card connector (J5) is provided on the KONDOR AX - Advanced System Development Board. If SD Card is inserted it can also be used as boot source for the i.MX6 processor. The SD Card is connected to a USDHC-2 controller.

5.6 HDMI

A HDMI output (J2) is provided on the KONDOR AX - Advanced System Development Board. As DDC_CLK and DDC_DAT lines are connected to an I2C2 controller, HDCP encryption is not supported on the KONDOR AX - Advanced System Development Board.

5.7 DDR3

As the KONDOR AX - Advanced System Development Board features the i.MX6 Solo processor, two (U9 and U10) 16-bit DDR3 memory chips (MT41K128M16JT) are mounted on the board. Result is 32-bit wide memory interface, with total size of 512 MB. Memory is routed with the fly-by topology.

5.8 Ethernet interface

The i.MX6 processor features a Gigabit Ethernet connection. The Gigabit Ethernet connection is implemented using RGMII interface on the i.MX6 processor and a KSZ9031RNX (U13) physical-layer transceiver. The KSZ9031RNX transceiver provides RGMII reference clock to the i.MX6 processor. The RJ45 connector (J4) includes network magnetics providing the proper signal conditioning, electro-magnetic interference suppression and signal isolation. This connector includes two status LEDs (status LEDs configuration can be changed via MDC/MDIO interface).

5.9 Camera Interface

Camera supporting dual lane MIPI CSI-2 interface can be connected to the i.MX6 processor via a J6 connector. The J6 connector is 24 pin 0.4 mm pitch board-to-FPC connector. Pinout of the J6 connector is compatible with LI-OV5640-MIPI-AF camera (available from Leopard Imaging Inc.). Camera can be controlled using I2C2 bus, while the i.MX6 processor has to supply clock signal to camera using GPIO_0 pin. Using GPIO pins the i.MX6 processor can control reset and power enable signal.

5.10 USB OTG

On the KONDOR AX - Advanced System Development Board microUSB connector (U20) is connected to an USB OTG controller of the i.MX6 processor. When OTG pin on microUSB connector is floating, USB OTG controller will be in device mode. In this mode the USB OTG controller can be used for serial download boot mode. Using USB cable that shorts OTG pin to ground USB OTG controller will switch to host mode and the i.MX6 processor will apply 5 V power to the microUSB connector. While in device mode power supply to the microUSB connector is disabled.

5.11 UART

On the KONDOR AX - Advanced System Development Board two UART interfaces, UART1 and UART2, are available for interface with the i.MX6 processor. The UART1

interface is connected with FT232RQ (U22), USB to serial UART chip. For UART activity indication one dual LED (with green and red indications) is provided on the board. A microUSB (U24) connector is used to interface with FT232RQ and UART1. The UART2 interface is available at J3 header. This signals are connected directly to pins of the i.MX6 processor, so only 3.3 V signal levels are supported on this header.

5.12 LEDs

For status indication 6 LEDs, which are connected to the i.MX6 processor, are provided on the KONDOR AX - Advanced System Development Board.

Table 25. LEDs definition

LED Number	i.MX6 Pin #	PCB Designator	LED Color
LED0	A18	D10	Amber
LED1	C17	D9	Amber
LED2	F16	D8	Green
LED3	D17	D7	Green
LED4	A19	D6	Red
LED5	B18	D5	Red

5.13 Interfaces with the ECP5UM FPGA

The i.MX6 processor can interface with the ECP5UM FPGA using one of several interfaces. High bandwidth interfaces available on the KONDOR AX - Advanced System Development Board are PCIe and EIM. Additionally, SPI, UART and I2C low bandwidth interfaces are also available on the board. The SPI interface is connected to the ECSPI1 controller, UART interface is connected to the UART4 controller, while I2C bus is connected to the I2C4 controller.

The i.MX6 processor can program and configure the ECP5UM FPGA. The FPGA programing and configuration can be done using JTAG or SSPI mode. SPI interface used for the FPGA programming is connected to the ECSPI2 controller.

When programming the FPGA in JTAG mode, select signal towards U26 analog mux has to be asserted to high state. This is required to switch JTAG signals from the FT2232 on board programming cable to the i.MX6 processor. Signals associated

with JTAG mode programming are listed Table 26. Signals associated with JTAG interface Table 26.

Table 26. Signals associated with JTAG interface

Signal Name	i.MX6 Pin #	Comment
FPGA_TMS	R1	
FPGA_TDI	P6	
FPGA_TDO	R7	
FPGA_TCK	R4	
ECP4_JTAG_SEL	T4	Set to 1 to connect JTAG interface to the i.MX6
FPGA_INITn	T2	
FPGA_PROGRAMn	R5	
FPGA_DONE	R2	

When programming the FPGA in SSPI mode, the FPGA has to be switched to the SSPI mode. The FPGA switches to the SSPI mode when value 001 is latched during power up on CFG[2:0] pins. Thus, to program the FPGA in SSPI mode, the i.MX6 processor has to set proper values on CFG[2:0] signals and then reset the FPGA. Details on how to reset the FPGA using FPGA_INITn and FPGA_PROGRAMNn can be found in ECP5UM documentation. The FPGA SPI signals are connected with the i.MX6 ECSPI controller using U28 analog mux. Signals associated with SSPI programming mode are listed in Table 27.

If the i.MX6 processor is not performing any ECP5UM FPGA configuration it is recommended to tri-state all signals associated with the ECP5UM FPGA configuration.

Table 27. Signals associated with SSPI programming mode

Signal Name	i.MX6 Pin #	Comment
ECSPI2_SCLK	U23	
ECSPI2_MISO	U24	
ECSPI2_MOSI	T21	
ECSPI2_CS0	V25	
FPGA_CFG0	T3	CFG[1:0]=01 SSPI mode (CFG0 is also used as mux select signal); CFG[1:0]=10 MSPI mode
FPGA_CFG1	R3	
FPGA_INITn	T2	
FPGA_PROGRAMn	R5	
FPGA_DONE	R2	

Errata for PCB Revisions A & B

SERDES lanes are muxed between FMC connector and other devices on the board with passive networks. Passive networks placed are U35/U37, U36/U38, U40/U42 and U44/U46. Pinout of passive networks is depicted if legend on silkscreen, as shown in Figure 1. However, legend drawn on silkscreen is not applicable for U36/U38. Correct pinout of U36/U38 passive network is depicted in Table 27.

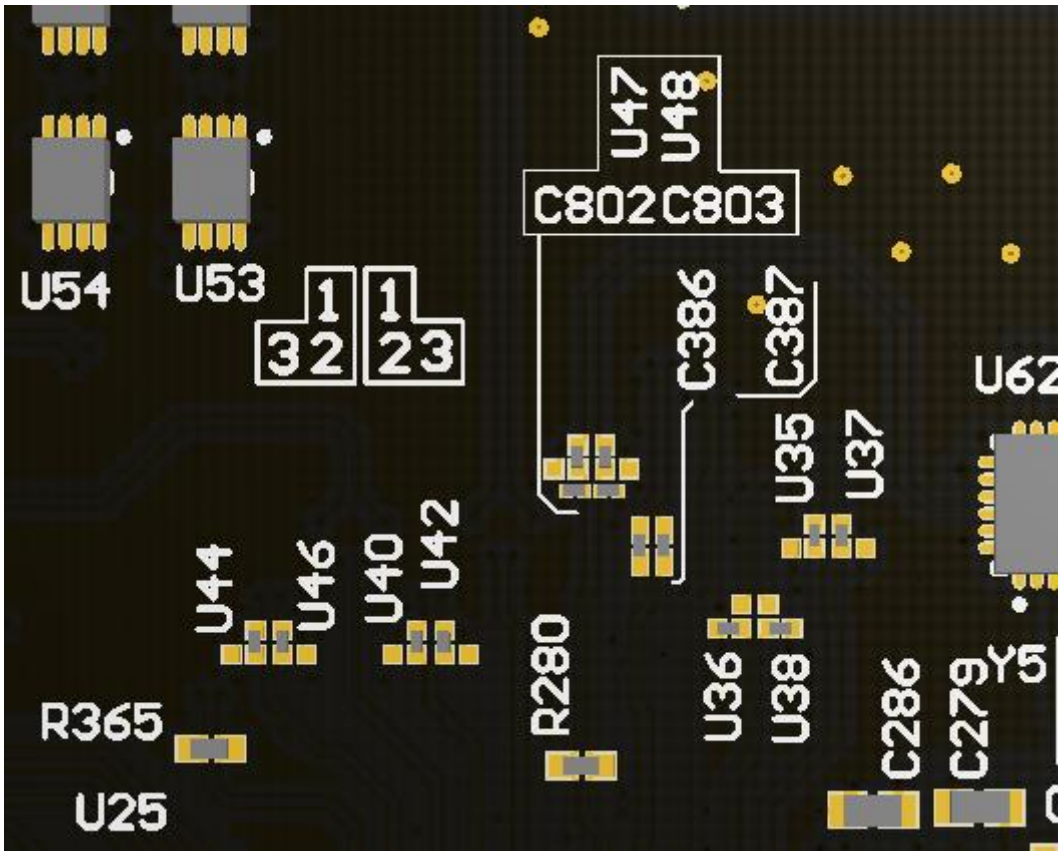

Figure 4: PCB Revision A & B

Table 28: Correct pinout

	3	3	
1	2	2	1

Index of Tables

Terms of use

The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders shall be liable for damages.

All rights, including rights created by patent grant or registration of a utility model or design, are reserved.

Technical data is subject to change at any time.

Copyright © 2015 Mikroprojekt d.o.o. All Rights Reserved.

Contact info