Openpattern Modular Routing Platform – V.2 prototype

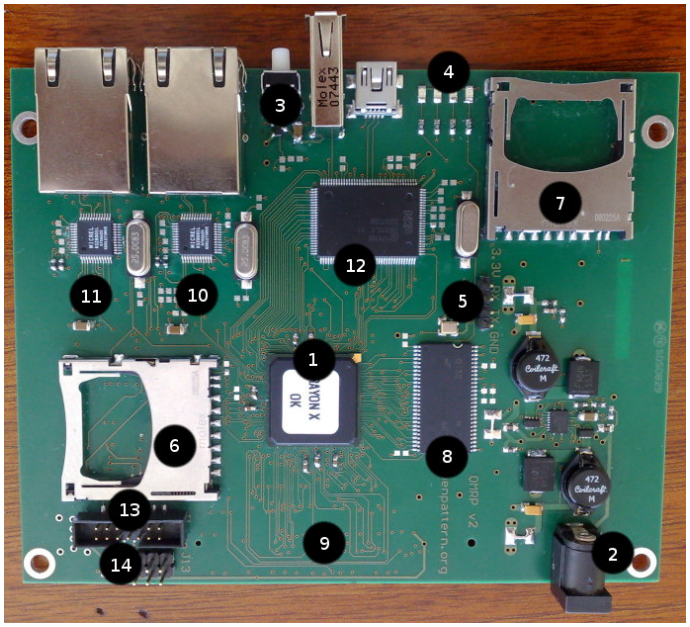
Preliminary board datasheet

http://openpattern.org

1 Introduction

The Openpattern Modular Routing Platform (OMRP) is a FPGA-based board targetting networking applications.
The goal is to create a new network router that can be used for mesh Wi-Fi networks, as an alternative of ISPs home-gateways, and as a development platform for open-hardware SoC projects.
This document describes the prototype version 2, which can be used as a FPGA development board. It contains information for FPGA developers to take advantage of the OMRP board.

2 Components



1. Xilinx XC3S1200E FPGA. This is the central component of the OMRP.
2. Power plug. The board must be powered from a regulated 5V source (included).
3. Reset button.

4. LEDs. One power indicator, one LED connected to the AVR MCU and two LEDs controllable by the FPGA.
5. UART connector.
6. First SDIO connector. This connector is designed to hold a SD (Secure Digital) card containing the FPGA bitstream and optional user data (OS, filesystem, ...).
7. Second SDIO connector.
8. Micron MT48LC16M16A2P-7E SDR SDRAM (32MB).
9. Mini PCI connector (under the board).
10. Micrel KSZ8001 10/100 Ethernet PHY. First Ethernet interface.
11. Micrel KSZ8001 10/100 Ethernet PHY. Second Ethernet interface.
12. ISP1761 3-port USB controller.
13. JTAG connector. This is connected to the FPGA and can be used for quick reconfiguration and debugging.
14. AVR ISP connector. Used to flash the ATtiny24 MCU that bootstraps the FPGA from the SD card.

3 Connections

3.1 Clock

The board is equipped with a 50MHz resonator that provides a clock to the FPGA. This clock can be fed to the on-chip DCMs to derive other frequencies inside the FPGA.

Description	**Signal name**	**FPGA pin**
50MHz system clock	SYS_CLK	B8

3.2 Reset button

The board provides a de-bounced pushbutton that can be used as a system reset. The reset is active low, ie. the logic level on the pin is 0 if the button is pressed, and 1 if it is not.

Description	**Signal name**	**FPGA pin**
Reset button signal	SYS_RST	U1

3.3 LEDs

Two LEDs are controllable by the FPGA. The two others are one AVR-controlled LED (see below) and one power indicator LED. LED signals are active high (i.e : logic level is 1 to power on the LED, 0 to power down).

Description	**Signal name**	**FPGA pin**
FPGA LED 1	LED1	F9
FPGA LED 2	LED2	F8

3.4 UART

The UART connector is a 4-pin HE10 3.3V connector. Pinout is :

1	GND
2	TX (data from FPGA to equipment)
3	RX (data from equipment to FPGA)
4	3.3V

This pinout is also engraved in the PCB copper as a reminder. To connect a PC's RS232 port, a level shifter is required, and can be powered from the 3.3V source of the UART connector.

Description	**Signal name**	**FPGA pin**
UART TX	UART_TX	H3
UART RX	UART_RX	H4

3.5 SDRAM

The board is equipped with a single Micron MT48LC16M16A2P-7E PC133 SDRAM chip, with a capacity of 32 megabytes. The chip provides a 16-bit data bus.

Description	**Signal name**	**FPGA pin**
Address line 12	SDRAM_A12	B10
Address line 11	SDRAM_A11	E10
Address line 10	SDRAM_A10	C11
Address line 9	SDRAM_A9	D11
Address line 8	SDRAM_A8	A11
Address line 7	SDRAM_A7	E11
Address line 6	SDRAM_A6	E12
Address line 5	SDRAM_A5	D13
Address line 4	SDRAM_A4	A13
Address line 3	SDRAM_A3	B13
Address line 2	SDRAM_A2	F12
Address line 1	SDRAM_A1	A12
Address line 0	SDRAM_A0	A16

Description	Signal name	FPGA pin
Bank Address 1	SDRAM_BA1	D10
Bank Address 0	SDRAM_BA0	B11
Column Address Strobe	SDRAM_nCAS	G9
Row Address Strobe	SDRAM_nRAS	E9
Write Enable	SDRAM_nWE	E8
Clock enable	SDRAM_CKE	C9
Chip select	SDRAM_nCS	A10
Clock	SDRAM_CLK	D9
Data mask line 1	SDRAM_DQM1	A14
Data mask line 0	SDRAM_DQM0	B14
Data line 15	SDRAM_DQ15	C3
Data line 14	SDRAM_DQ14	A4
Data line 13	SDRAM_DQ13	D5
Data line 12	SDRAM_DQ12	D4
Data line 11	SDRAM_DQ11	A6
Data line 10	SDRAM_DQ10	A7
Data line 9	SDRAM_DQ9	A8
Data line 8	SDRAM_DQ8	C7
Data line 7	SDRAM_DQ7	D7
Data line 6	SDRAM_DQ6	E7
Data line 5	SDRAM_DQ5	B6
Data line 4	SDRAM_DQ4	D6
Data line 3	SDRAM_DQ3	C5
Data line 2	SDRAM_DQ2	C4
Data line 1	SDRAM_DQ1	B4
Data line 0	SDRAM_DQ0	E6

3.6 SDIO connector 1

This first SDIO connector is shared between the AVR FPGA loader and the FPGA. At board power-up and during reconfigurations, the AVR takes control of this connector, reads the bitstream from the SD card, configures the FPGA and switches the SDIO interface to the FPGA. This is handled by on-board hardware and transparent to the FPGA.

The two SDIO interfaces are used in SPI mode and all SPI signals are shared between them, except for the chip-select lines.

Description	Signal name	FPGA pin
Clock (shared)	SDIO_CLK	C14
Master Out Slave In (shared)	SDIO_DI	K14
Master In Slave Out (shared)	SDIO_DO	M13
Chip select	SDIO0_CS	B16

3.7 SDIO connector 2

This connector shares its signals with the SDIO connector 1, except for the chip-select line.

Description	Signal name	FPGA pin
Clock (shared)	SDIO_CLK	C14
Master Out Slave In (shared)	SDIO_DI	K14
Master In Slave Out (shared)	SDIO_DO	M13
Chip select	SDIO1_CS	F7

3.8 Ethernet interface 1

The two Ethernet interfaces are driven by Micrel KSZ8001 10/100 Ethernet PHYs, which are wired in RMII (Reduced Media Independent Interface) mode.

The clock (ETH_RCLK), control interfaces (ETH_MDIO and ETH_MDC) and reset (ETH_nRST) are shared between the PHYs. Shared Ethernet signals are shown below :

Description	Signal name	FPGA pin
Management data input/output	ETH_MDIO	J1
Management data clock	ETH_MDC	R10
Reset (active low)	ETH_nRST	J2
RMII Clock	ETH_RCLK	V9

The first Ethernet interface has MII address 1. PHY address is wired on the board using resistors.

Description	Signal name	FPGA pin
Transmit data line 1	ETH0_TX1	R9
Transmit data line 0	ETH0_TX0	T5
Transmit enable	ETH0_TXEN	K5
Receive data line 1	ETH0_RX1	J5
Receive data line 0	ETH0_RX0	J4
Receive data valid	ETH0_RXDV	J2
Receive error	ETH0_RXER	K3
Interrupt (active low)	ETH0_nINT	L3

3.9 Ethernet interface 2

The second Ethernet interface is similar to the first, and has MII address 3.

Description	Signal name	FPGA pin
Transmit data line 1	ETH1_TX1<1>	T8
Transmit data line 0	ETH1_TX0	M9
Transmit enable	ETH1_TXEN	P10
Receive data line 1	ETH1_RX1	H2
Receive data line 0	ETH1_RX0	H1
Receive data valid	ETH1_RXDV	K4
Receive error	ETH1_RXER	K6
Interrupt (active low)	ETH1_nINT	L1

3.10 USB

The USB controller is a NXP ISP1761 device. It allows 1 device port and 2 host ports on the board.

The first two ports (following ISP1761's datasheet naming conventions) – one device and one host – are available on the front side of the board. The third port (host) is only available as a footprint, near the second SDIO connector. It will be replaced with a Mini-ExpressCard slot in the end product.

All host ports have power control with analog overcurrent detection.

The ISP1761's bus is 16-bit wide and DMA control signals are not used.

Description	**Signal name**	**FPGA pin**
Address line 17	USB_A17	C1
Address line 16	USB_A16	C2
Address line 15	USB_A15	D1
Address line 14	USB_A14	D2
Address line 13	USB_A13	V5
Address line 12	USB_A12	E2
Address line 11	USB_A11	E1
Address line 10	USB_A10	E4
Address line 9	USB_A9	E3
Address line 8	USB_A8	F1
Address line 7	USB_A7	F2
Address line 6	USB_A6	P7
Address line 5	USB_A5	N7
Address line 4	USB_A4	U6
Address line 3	USB_A3	V6
Address line 2	USB_A2	H6
Address line 1	USB_A1	H5
Data line 15	USB_D15	L5
Data line 14	USB_D14	L6
Data line 13	USB_D13	M5
Data line 12	USB_D12	M3
Data line 11	USB_D11	M4
Data line 10	USB_D10	P6
Data line 9	USB_D9	U3

Description	Signal name	FPGA pin
Data line 8	USB_D8	U4
Data line 7	USB_D7	P2
Data line 6	USB_D6	P1
Data line 5	USB_D5	U5
Data line 4	USB_D4	T4
Data line 3	USB_D3	R3
Data line 2	USB_D2	R2
Data line 1	USB_D1	T2
Data line 0	USB_D0	T1
Chip select (active low)	USB_CS_N	V7
Asynchronous read strobe (active low)	USB_RD_N	P8
Asynchronous write strobe (active low)	UWB_WR_N	N8
Reset (active low)	USB_RESET_N	L4
Peripheral controller IRQ	USB_DC_IRQ	P9
Host controller IRQ	USB_HC_IRQ	R8

3.11 Mini PCI

The Mini PCI connector, located on the bottom layer of the PCB, is connected directly to the FPGA I/O pads.

Description	Signal name	FPGA pin
Reset	PCI_RST	E16
Clock	PCI_CLK	U9
Clock enable for power management	PCI_CLKRUN	C17
Power Management Event	PCI_PME	K13
Interrupt A	PCI_INTA	G13
Interrupt B	PCI_INTB	R13
66MHz mode enable	PCI_M66EN	U13
Request	PCI_REQ	N12
Grant	PCI_GNT	D17
Transfer stop	PCI_STOP	R12
Indicates bulk transfer	PCI_FRAME	T16
Initiator ready	PCI_IRDY	P12
Target ready	PCI_TRDY	P13
Initialization device select	PCI_IDSEL	C18
System error	PCI)SERR	T14
Device selection	PCI_DEVSEL	T15
Parity	PCI_PAR	V15
Parity error	PCI_PERR	U15
Multiplexed address/data line 31	PCI_AD31	H16
Multiplexed address/data line 30	PCI_AD30	J13
Multiplexed address/data line 29	PCI_AD29	J12
Multiplexed address/data line 28	PCI_AD28	F18
Multiplexed address/data line 27	PCI_AD27	J15
Multiplexed address/data line 26	PCI_AD26	J16
Multiplexed address/data line 25	PCI_AD25	J17
Multiplexed address/data line 24	PCI_AD24	F17
Multiplexed address/data line 23	PCI_AD23	K15
Multiplexed address/data line 22	PCI_AD22	G15
Multiplexed address/data line 21	PCI_AD21	H15
Multiplexed address/data line 20	PCI_AD20	L17
Multiplexed address/data line 19	PCI_AD19	L18
Multiplexed address/data line 18	PCI_AD18	L15
Multiplexed address/data line 17	PCI_AD17	H14
Multiplexed address/data line 16	PCI_AD16	M18
Multiplexed address/data line 15	PCI_AD15	N18
Multiplexed address/data line 14	PCI_AD14	M16
Multiplexed address/data line 13	PCI_AD13	M15
Multiplexed address/data line 12	PCI_AD12	P18
Multiplexed address/data line 11	PCI_AD11	P17
Multiplexed address/data line 10	PCI_AD10	G14
Multiplexed address/data line 9	PCI_AD9	M14
Multiplexed address/data line 8	PCI_AD8	N14
Multiplexed address/data line 7	PCI_AD7	N15
Multiplexed address/data line 6	PCI_AD6	P16

Description	Signal name	FPGA pin
Multiplexed address/data line 5	PCI_AD5	R16
Multiplexed address/data line 4	PCI_AD4	R15
Multiplexed address/data line 3	PCI_AD3	T18
Multiplexed address/data line 2	PCI_AD2	R18
Multiplexed address/data line 1	PCI_AD1	T17
Multiplexed address/data line 0	PCI_AD0	U18
Byte enable 3	PCI_BE3	T12
Byte enable 2	PCI_BE2	V13
Byte enable 1	PCI_BE1	V12
Byte enable 0	PCI_BE0	N11

3.12 JTAG connector

The JTAG connector is a 2.5V 14-pin Xilinx JTAG header, compatible with Xilinx cables. It is connected directly to the FPGA, as there is no other device on the chain.

NC	14	13	GND
NC	12	11	GND
TDI	10	9	GND
TDO	8	7	GND
TCK	6	5	GND
TMS	4	3	GND
VREF (2.5V)	2	1	GND

3.13 AVR

The ATtiny24 AVR is connected on both the FPGA and the SDIO card. Only slot 1 is available to the AVR. The FPGA is put in slave serial mode to receive its configuration data from the AVR. The AVR also controls a multiplexer that switches the SDIO slot between the FPGA and itself.

The ATtiny24 pins are connected as follows :

Pin	Description
PA0	SD card chip-select signal.
PA1	Connected to the FPGA's PROG_B signal (B1). Warning: 2.5V signal (see below).
PA2	Connected to the FPGA's DONE signal (V17). Warning: 2.5V signal (see below).
PA3	Controls the switch of the SD card slot. 1 = SD card slot belongs to the AVR, 0 = SD card slot belongs to the FPGA.
PA4	SD card clock signal.
PA5	SD card data output (SD card to AVR).
PA6	SD card data input (AVR to SD card).
PA7	Connected to the DIN pin of the FPGA (N10).
PB0	Connected to a LED (active high).
PB1	Connected to the INIT_B signal of the FPGA (T3).
PB2	Connected to the CCLK pin of the FPGA (U16).
PB3	Reset pin.

PROG_B and DONE signals are 2.5V signals while the ATtiny24 operates from 3.3V, so care should be taken not to configure these pins as outputs driven high. These pins have on-board pull-up resistors to 2.5V, so you should configure them as inputs if you want to drive them high.

To program the ATtiny24, the board exposes an ISP connector using the following pinout :

GND	1	2	3.3V
RESET	3	4	MISO
MOSI	5	6	CLK

Standard AVR programming cables can be used to program the ATtiny24.

4 Atmel ATtiny24 MCU

The Atmel ATtiny24 MCU is a flexible replacement for the Xilinx Platform Flash, which allows FPGA reconfiguration using off-the-shelf tools and usually without the need to connect any cable to the OMRP. The FPGA is programmed in Slave Serial mode and the ATtiny24 reads the bitstream from the first FAT partition found on the on-board Secure Digital Card.

4.1 Xilinx boot protocol

Programming operations are described below :

- The MCU reads the on-board SD card to find the bitstream in a FAT partition.
- Once the file is found, the MCU starts reading the data blocks.
- It transfers them to the FPGA using the FPGA_DIN and FPGA_CCLK pins.
- The MCU waits for the FPGA_DONE signal to go high, indicating that the FPGA has been successfully configured.

4.2 Software requirements

You can use either AVR-studio, AVRdude or an avr toolchain packaged for your operating system. We recommend using gcc-avr, binutils-avr and libc-avr versions 4.3.0, 2.18 and 1.6.2 which are known to produce working binaries for the ATtiny24.

4.3 Programming the MCU

Several programming tools can be used that allow you to program the MCU over standard serial ports or USB. AVRstudio and uisp have been successfully tested with the MCU.

4.4 Error decoding

We use the built-in MCU Universal Serial Interface to do SPI transfers between the SD card and the the MCU, thus no UART output is available unless we multiplex pins. The MCU can control one LED in order to inform the user about any programming or read error that could happen. Error codes are shown below :

- LED is powered on for 500 ms, then powered off : an error occured while initializing the SD card.
- LED is powered on for 500 ms, powered off, then powered on for 500 ms and powered off again : an error occured while reading the SD card partition table.
- LED is powered on for 500 ms, powered off, then powered on for 1 s and powered off again : an error occured while initializing the FPGA (timeout).
- LED is powered on for 500 ms, powered off, then powered on for 2 s and powered off again : an error occured while programming the FPGA.

5 Software requirements

5.1 Synthesis software

We recommend using latest Xilinx ISE WebPack edition, superior or equal to version 9.2i w/ latest service packs applied. GNU make is also recommended in order to use the Makefiles shipped with the sources, however you can still generate an ISE project file and include the HDL sources to get the synthesis done.

5.2 Programming tools

If you plan to program the FPGA directly using the on-board 14-pin JTAG connector, we recommend using impact which is bundled with Xilinx ISE WebPack edition.

5.3 Simulation tools

Most of the simulations have been done using Cver GPL edition version 2.12a. Icarus Verilog is also known to be working.

6 Contact

OpenPattern welcomes feedback on this document and on the OMRP design. You can reach us :

- by email: `contact@openpattern.org`
- by phone: `+33(0)134933420`
- via our website: `http://www.openpattern.org`
- via IRC: `#openpattern` on FreeNode (`irc.freenode.net`)