**Lime Microsystems Limited**

Surrey Technology Centre

Occam Road

The Surrey Research Park

Guildford, Surrey GU2 7YG

United Kingdom

Tel: +44 (0) 1483 685 063

e-mail: [enquiries@limemicro.com](mailto:enquiries@limemicro.com)

**LimeSDR-Mini**

***- FPGA Gateware Description****-*

REVISION HISTORY

The following table shows the revision history of this document:

|  |  |  |
| --- | --- | --- |
| **Date** | **Version** | **Description of Revisions** |
| 25/06/2018 | 1.0 | Initial version |
| 24/04/2022 | 2.0 | Update for LimeSDR-MINI v2 board |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Table of Contents**

[REVISION HISTORY 2](#_Toc102905520)

[1 Introduction 4](#_Toc102905521)

[2 FPGA gateware features 5](#_Toc102905522)

[3 Gateware description 6](#_Toc102905523)

[3.1 Main block diagram 6](#_Toc102905524)

[3.2 Clock network 7](#_Toc102905525)

[3.3 Softcore processor – cpu\_top 8](#_Toc102905526)

[3.3.1 Registers of fpgacfg module 10](#_Toc102905527)

[3.3.2 Registers of pllcfg module 14](#_Toc102905528)

[3.3.3 Registers of tstcfg module 16](#_Toc102905529)

[3.3.4 Registers of periphcfg module 19](#_Toc102905530)

[3.4 FT601 FIFO interface – FT601\_top 20](#_Toc102905531)

[3.5 LMS7002 Receive and transmit interface – rxtx\_top 22](#_Toc102905532)

[3.5.1 Receive interface – rx\_path\_top 24](#_Toc102905533)

[3.5.2 Transmit interface – tx\_path\_top 26](#_Toc102905534)

[3.6 General periphery – general\_periph\_top 27](#_Toc102905535)

[3.7 LMS7002 module – lms7002\_top 30](#_Toc102905536)

[3.8 Board test module – tst\_top 31](#_Toc102905537)

[4 Examples 33](#_Toc102905538)

[4.1 Accessing FPGA registers 33](#_Toc102905539)

[4.2 Accessing LMS7002M registers 34](#_Toc102905540)

[4.3 Periphery control 34](#_Toc102905541)

[4.4 Configuring FPGA PLL module 35](#_Toc102905542)

[4.4.1 PLL module – RX clock configuration (auto phase shift mode) 35](#_Toc102905543)

[4.4.2 PLL module – TX clock configuration (auto phase shift mode) 36](#_Toc102905544)

[4.5 Controlling TX and RX data stream 37](#_Toc102905545)

# Introduction

This document contains functional description of FPGA gateware project suited for LimeSDR-Mini board.

**FPGA project** - LimeSDR-Mini\_lms7\_trx project can be downloaded from GitHub repository

<https://gitlab.com/myriadrf/limesdr-mini_v2_gw>.

**Required hardware** – LimeSDR-Mini v2 board.

**Development software** – project is created with Lattice Diamond, Version 3.12.1.454. Mentioned software edition is free and can be downloaded from [https://www.latticesemi.com](https://www.latticesemi.com/en/Products/DesignSoftwareAndIP/FPGAandLDS/LatticeDiamond). It is recommended to use same version as project was created.

# FPGA gateware features

Gateware contains following features:

* Interface to LMS7002 LimeLightTM digital IQ interface in TRXIQ double data rate mode;
* Real time data transfer between PC and LMS7002 chip;
* Connection to FT601 FIFO interface for transferring data through USB3.0;
* TX samples synchronization with RX samples time stamp;
* SPI connection between LMS7002 chip and other on-board devices;
* Reconfigurable PLL block for LMS7002 clocking;
* Internal SPI registers for FPGA control.

# Gateware description

This chapter describes main modules of LimeSDR-Mini\_lms7\_trx project.

## Main block diagram

MAX 10 FPGAprovides FIFO interface with FT601 USB3.0 controller. There are two endpoints (EP02 and EP82) implemented for control data and two endpoints for stream data (EP03 and EP83). Control endpoints are connected to NIOS II softcore processor which provides SPI and I2C communication interfaces for LMS7002M chip, XO DAC, EEPROM, FLASH. NIOS also provides access to internal SPI configuration registers. Stream endpoints are dedicated for receiving and sending IQ data from/to LMS7002M. **Figure 1** contains top block diagram with main modules. Description of main FPGA instances can be found in **Table 1**.



Figure 1 Top block diagram

Table 1 Description of main instances

| **Instance** | **Description** |
| --- | --- |
| nios\_cpu | NIOS II softcore processor with memory registers. Provides periphery control. See **3.3 Softcore processor – cpu.** |
| FT601\_top | Provides data transfer between external FT601 USB 3.0 peripheral controller and FPGA. See **3.4 FT601 FIFO interface – FT601\_top.** |
| rxtx\_top | Receive and transmit logic between FPGA and external LMS7002 transceiver. See **3.5 LMS7002 Receive and transmit interface – rxtx\_top.** |
| general\_periph\_top | Control module for onboard periphery such as LEDs, GPIO, FAN. See **3.6 General periphery – general\_periph\_top.** |
| pll\_top | Module provides required clocks for rxtx\_top module. See **0**  **LMS7002 module –** |
| tst\_top | Board test logic to external clocks. See **3.8 Board test module – tst\_top.** |

## Clock network

**Figure 2** shows dataflow between main modules and clocking scheme. More details can be found in **Table 2**.



Figure 2 Gateware clock network

Table 2 Clock network description

| **Clock name** | **Frequency, MHz** |  | **Description** |
| --- | --- | --- | --- |
| LMS\_MCLK1 | Configurable | | TX clock from LMS7002M IC. |
| LMS\_MCLK2 | Configurable | | RX clock from LMS7002M IC. |
| LMS\_FCLK1 | Configurable | | Sample clock, LMS7002M IC latches LMS\_DIQ1 bus signals using this clock. |
| LMS\_FCLK2 | Configurable | | Not used |
| tx\_clk | Configurable | | FPGA launches LMS\_DIQ1 bus signals using this clock. Used for clocking FPGA TX modules. |
| rx\_clk | Configurable | | FPGA latches LMS\_DIQ2 bus signals using this clock. Used for clocking FPGA RX modules. |
| LMK\_CLK | 30.72 | | Reference clock from LMK00105 clock buffer. |
| FT\_CLK | 100 | | FT601 FIFO interface clock. |

## Softcore processor – cpu\_top

**Figure 3** shows block diagram of nios\_cpu module. This module contains softcore ALTERA NIOS II CPU and user accessible configuration registers for other modules. More detailed description can be found in **Table 3**. Module generic parameters are explained in **Table 4** and ports are described in **Table 5**.



Figure 3 nios\_cpu block diagram

Table 3 Description of nios\_cpu instances

| **Instance** | **Description** |
| --- | --- |
| lms\_ctr | NIOS II softcore processor instance. Processor constantly monitors input FIFO buffer connected to *exfifo\_if* ports and reads one packet containing 64 bytes. See **LMS64C control protocol** document for protocol description and command list. NIOS CPU executes received command and writes 64 bytes response packet to FIFO buffer connected to *exfifo\_of* ports. |
| cfg\_top | Wrapper module for SPI configuration registers. |
| fpgacfg | General configuration 32x16b addressable registers. Address range 0x0000 - 0x001F. See **Table 6** for register description. |
| pllcfg | PLL configuration registers. Address range 0x0020 - 0x003F. See **Table 7** for register description. |
| tstcfg | Test module configuration registers. Address range 0x0060 - 0x007F. see **Table 8** for register description. |
| periphcfg | Peripheral configuration registers. Address range 0x0020 - 0x003F. See **Table 9** for register description. |

Table 4 nios\_cpu module parameters

| **Parameter** | **Type** | **Default** | **Description** |
| --- | --- | --- | --- |
| Start address of SPI registers | | | |
| FPGACFG\_START\_ADDR | integer | 0 | Start address of SPI register modules. Has to be multiple of 32 |
| PLLCFG\_START\_ADDR | integer | 32 |
| TSTCFG\_START\_ADDR | integer | 64 |
| PERIPHCFG\_START\_ADDR | integer | 192 |

Table 5 nios\_cpu module ports

| **Port** | **Type** | **Width** | **Description** |
| --- | --- | --- | --- |
| clk | in | 1 | Free running clock. 30.72MHz |
| reset\_n | in | 1 | Asynchronous, active low reset |
| Control data FIFO | | | |
| exfifo\_if\_d | in | 32 | External control input FIFO data |
| exfifo\_if\_rd | out | 1 | External control input FIFO read request |
| exfifo\_if\_rdempty | in | 1 | External control input FIFO read empty |
| exfifo\_of\_d | out | 32 | External control output FIFO data |
| exfifo\_of\_wr | out | 1 | External control output FIFO write request |
| exfifo\_of\_wrfull | in | 1 | External control output FIFO write full |
| exfifo\_of\_rst | out | 1 | External control output FIFO reset request, active high |
| SPI 0 | | | |
| spi\_0\_MISO | in | 1 | SPI 0 master input |
| spi\_0\_MOSI | out | 1 | SPI 0 master output |
| spi\_0\_SCLK | out | 1 | SPI 0 clock |
| spi\_0\_SS\_n | out | 5 | SPI 0 slave select. spi\_0\_SS\_n[0] - connected to LMS7002, spi\_0\_SS\_n[1] - to internal SPI modules, spi\_0\_SS\_n[0] – connected to XO DAC |
| SPI 1 | | | |
| spi\_1\_MOSI | out | 1 | SPI 1 master output |
| spi\_1\_SCLK | out | 1 | SPI 1 clock |
| spi\_1\_SS\_n | out | 2 | SPI 1 slave select. Connected to SPI FLASH |
| I2C | | | |
| i2c\_scl | inout | 1 | I2C bus clock, connected to temperature sensor and EEPROM memory. |
| i2c\_sda | inout | 1 | I2C bus data, connected to temperature sensor and EEPROM memory. |
| General purpose I/O | | | |
| gpi | in | 8 | Not used |
| gpo | out | 8 | gpo[0] - indicates NIOS activity. 0 - Idle, 1 - Busy.  gpo[7-1] - not used |
| LMS7002 control | | | |
| lms\_ctr\_gpio | out | 4 | lms\_ctr\_gpio[0] - LMS7002 reset. lms\_ctr\_gpio[3-1] - not used |
| Configuration registers | | | |
| from\_fpgacfg | out | 512 | Input/output ports from/to SPI configuration registers |
| to\_fpgacfg | in | 512 |
| from\_pllcfg | out | 512 |
| to\_pllcfg | in | 512 |
| from\_tstcfg | out | 512 |
| to\_tstcfg | in | 512 |
| to\_tstcfg\_from\_rxtx | in | 512 |
| to\_periphcfg | in | 512 |
| from\_periphcfg | out | 512 |

### Registers of fpgacfg module

Table 6 Register description of fpgacfg module

| **Address** | **Def. value** | **Bits** | **Name** | **Description** |
| --- | --- | --- | --- | --- |
| 0x0000 |  |  | **Board identification number** | |
| 15-0 | **Board ID** | LimeSDR-Mini (**Default 0x0011**) |
| 0x0001 |  | **Gateware version control** | | |
| 15-0 | **GW\_VER** | Gatewate version number |
| 0x0002 |  | **Gateware revision control** | | |
| 15-0 | **GW\_REV** | Gateware revision number |
| 0x0003 |  | **Board version control** | | |
| 15-7 | Reserved |  |
| 6-4 | **BOM\_VER** | Bill of material version |
| 3-0 | **HW\_VER** | Hardware version. |
| 0x0004 | 0000 | 15-0 | Reserved |  |
| 0x0005 | 0000 | **Clock source selection for TX and RX interfaces** | | |
| 15-2 | Reserved |  |
| 1 | **DRCT\_CLK\_EN** | RX clk: |
| 0 - PLL source **(Default)** |
| 1 - Direct clock source |
| 0 | TX clk: |
| 0 - PLL source **(Default)** |
| 1 - Direct clock source |
| 0x0006 | 0000 | 15-0 | Reserved |  |
| 0x0007 | 0303 | **RX TX MIMO Channel control** | | |
| 15-10 | Reserved |  |
| 9 | **CH\_EN** | TX ch. 1: |
| 0 - Disabled |
| 1 - Enabled **(Default)** |
| 8 | TX ch. 0: |
| 0 - Disabled |
| 1 - Enabled **(Default)** |
| 7-2 | Reserved |  |
| 1 | **CH\_EN** | RX ch. 1: |
| 0 - Disabled |
| 1 - Enabled **(Default)** |
| 0 | RX ch. 0: |
| 0 - Disabled |
| 1 - Enabled **(Default)** |
| 0x0008 | 0102 | **DIQ interface control** | | |
| 15-11 | Reserved |  |
| 10 | **DLB\_EN** | Not used |
| 9 | **SYNCH\_DIS** | Packets synchronization using timestamps: |
| 0 - Enabled |
| 1 - Disabled **(Default)** |
| 8 | **MIMO\_INT\_EN** | MIMO mode: |
| 0 - Disabled |
| 1 - Enabled **(Default)** |
| 7 | **TRIQ\_PULSE** | TRXIQ\_pulse mode: |
| 0 - OFF **(Default)** |
| 1 - ON |
| 6 | **DDR\_EN** | DIQ interface mode: |
| 0 - SDR |
| 1 - DDR **(Default)** |
| 5 | **MODE** | Limelight port mode: |
| 0 - TRXIQ **(Default)** |
| 1 - JESD207 (Currently not implemented) |
| 4-2 | Reserved |  |
| 1-0 | **SMPL\_WIDTH** | Interface sample width selection: |
| "10" - 12bit **(Default)** |
| "01" - Do not use |
| "00" - 16bit |
| 0x0009 | 0003 | **Packet control** | | |
| 15-2 | Reserved |  |
| 1 | **TXPCT\_LOSS\_CLR** | TX packets dropping flag clear: |
| 0 - Normal operation **(Default)** |
| 1 - Rising edge clears flag |
| 0 | **SMPL\_NR\_CLR** | Reset timestamp: |
| 0 - Normal operation **(Default)** |
| 1 - Timestamp is cleared |
| 0x000A | 0000 | **RX and TX module control** | | |
| 15-10 | Reserved |  |
| 9 | **TX\_PTRN\_EN** | Test pattern on TX: |
| 0 - Disabled **(Default)** |
| 1 - Enabled |
| 8 | **RX\_PTRN\_EN** | Test pattern on RX: |
| 0 - Disabled **(Default)** |
| 1 - Enabled |
| 7-2 | Reserved |  |
| 1 | **TX\_EN** | TX chain: |
| 0 - Disabled **(Default)** |
| 1 - Enabled |
| 0 | **RX\_EN** | RX chain: |
| 0 - Disabled **(Default)** |
| 1 - Enabled |
| 0x000B | 0000 | 15-0 | Reserved |  |
| 0x000C | 0003 | **WFM player control 1** | | |
| 15-2 | Reserved |  |
| 1 | **WFM\_CH\_EN** | WFM ch.1: |
| 0 - Disabled |
| 1 - Enabled **(Default)** |
| 0 | WFM ch.0: |
| 0 - Disabled |
| 1 - Enabled **(Default)** |
| 0x000D | 0001 | **WFM player control 2** | | |
| 15-3 | Reserved |  |
| 2 | **WFM\_LOAD** | WFM player file load: |
| 0 to 1 transition starts WFM file loading |
| 0 - WFM file loading disabled **(Default)** |
| 1 | **WFM\_PLAY** | WFM player loaded file play enable: |
| 0 - Disabled |
| 1 - Enabled **(Default)** |
| 0 | Reserved |  |
| 0x000E | 0002 | **WFM player control 3** | | |
| 15-2 | Reserved |  |
| 1-0 | **WFM\_SMPL\_WIDTH** | WFM player sample width control: |
| "10" - 12bit, **(Default)** |
| "01" - Do not use |
| "00" - 16bit |
| 0x000F | 0000 | 15-0 | Reserved |  |
| 0x0010 | 0000 | 15-0 | Reserved |  |
| 0x0011 | 0000 | 15-0 | Reserved |  |
| 0x0012 | FFFF | **Controlled SPI enable** | | |
| 15-8 | Reserved | Not used |
| 7 | **SPI\_SS7** |
| 6 | **SPI\_SS6** |
| 5 | **SPI\_SS5** |
| 4 | **SPI\_SS4** |
| 3 | **SPI\_SS3** |
| 2 | **SPI\_SS2** |
| 1 | **SPI\_SS1** |
| 0 | **SPI\_SS0** |
| 0x0013 | 6F6F | **LMS7002 MISC pin control** | | |
| 15 | Reserved |  |
| 14 | **LMS2\_RXEN** | Not used |
| 13 | **LMS2\_TXEN** |
| 12 | **LMS2\_TXNRX2** |
| 11 | **LMS2\_TXNRX1** |
| 10 | **LMS2\_CORE\_LDO\_EN** |
| 9 | **LMS2\_RESET** |
| 8 | **LMS2\_SS** |
| 7 | Reserved |  |
| 6 | **LMS1\_RXEN** | RX hard enable: |
| 0 - Disabled |
| 1 - Enabled **(Default)** |
| 5 | **LMS1\_TXEN** | TX hard enable: |
| 0 - Disabled |
| 1 - Enabled **(Default)** |
| 4 | **LMS1\_TXNRX2** | Port 2 mode selection: |
| 0 - TXIQ **(Default)** |
| 1 - RXIQ |
| 3 | **LMS1\_TXNRX1** | Port 1 mode selection: |
| 0 - TXIQ |
| 1 - RXIQ **(Default)** |
| 2 | **LMS1\_CORE\_LDO\_EN** | Internal LDO control: |
| 0 - Disabled **(Default)** |
| 1 - Enabled |
| 1 | **LMS1\_RESET** | Hardware reset: |
| 0 - Reset activated |
| 1 - Reset inactive **(Default)** |
| 0 | **LMS1\_SS** | Not used |
| 0x0014 | 0000 | 15-0 | Reserved for lms3\_4 |  |
| 0x0015 | 0000 | 15-0 | Reserved for lms5-6 |  |
| 0x0016 | 0000 | 15-0 | Reserved for lms7-8 |  |
| 0x0017 | 0000 | **GPIO for external periphery** | | |
| 15-14 | Reserved |  |
| 13 | **GPIO13** | Not used |
| 12 | **GPIO12** |
| 11 | **GPIO11** |
| 10 | **GPIO10** |
| 9 | **GPIO9** |
| 8 | **GPIO8** |
| 7 | **GPIO7** |
| 6 | **GPIO6** | Ch. B shunt: |
| 0 - Disabled |
| 1 - Enabled **(Default)** |
| 5 | **GPIO5** | Ch. B attenuator |
| 0 - Disabled **(Default)** |
| 1 - Enabled |
| 4 | **GPIO4** | RF loopback ch. B: |
| 0 - Disabled **(Default)** |
| 1 - Enabled |
| 3 | **GPIO3** | Reserved |
| 2 | **GPIO2** | Ch. A shunt: |
| 0 - Disabled |
| 1 - Enabled **(Default)** |
| 1 | **GPIO1** | Ch. A attenuator: |
| 0 - Disabled **(Default)** |
| 1 - Enabled |
| 0 | **GPIO0** | RF loopback ch. A: |
| 0 - Disabled **(Default)** |
| 1 - Enabled |
| 0x0018 | 0001 | 15-1 | Reserved |  |
| 0 | **DEV\_CTRL0** | Not used |
| 0x0019 |  | 15-0 | Reserved |  |
| 0x001A | 0000 | **Onboard led control** | | |
| 15 | Reserved |  |
| 14 | Reserved |
| 13 | Reserved |
| 12 | Reserved |
| 11 | Reserved |
| 10 | Reserved |
| 9 | Reserved |
| 8 | Reserved |
| 7 | Reserved |
| 6 | **FPGA\_LED2\_G** | Green LED2 control, do not turn on while red LED2 is on: |
| 0 - OFF **(Default)** |
| 1 - ON |
| 5 | **FPGA\_LED2\_R** | Red LED2 control, do not turn on while green LED2 is on: |
| 0 - OFF **(Default)** |
| 1 - ON |
| 4 | **FPGA\_LED2\_OVRD** | LED2 control override: |
| 0 - OFF **(Default)** |
| 1 - ON |
| 3 | Reserved |  |
| 2 | **FPGA\_LED1\_G** | Green LED1 control, do not turn on while red LED1 is on: |
| 0 - OFF **(Default)** |
| 1 - ON |
| 1 | **FPGA\_LED1\_R** | Red LED1 control, do not turn on while green LED1 is on: |
| 0 - OFF **(Default)** |
| 1 - ON |
| 0 | **FPGA\_LED1\_OVRD** | LED1 control override: |
| 0 - OFF **(Default)** |
| 1 - ON |
| 0x001B | 0000 | 15-8 | Reserved |  |
| 7 | Reserved |
| 6 | Reserved |
| 5 | Reserved |
| 4 | Reserved |
| 3 | Reserved |
| 2 | Reserved |
| 1 | Reserved |
| 0 | Reserved |
| 0x001C | 0000 | 15-3 | Reserved | Onboard led control |
| 2 | **FX3\_LED\_G** | Green FX3 control, do not turn on while red FX3 is on: |
| 0 - OFF **(Default)** |
| 1 - ON |
| 1 | **FX3\_LED\_R** | Red FX3 control, do not turn on while green FX3 is on: |
| 0 - OFF **(Default)** |
| 1 - ON |
| 0 | **FX3\_LED\_OVRD** | FX3 control override: |
| 0 - OFF **(Default)** |
| 1 - ON |
| 0x001D | 0000 | 15-0 | Reserved |  |
| 0x001E | 0000 | 15-0 | Reserved |  |
| 0x001F | 0000 | 15-0 | Reserved |  |

### Registers of pllcfg module

Table 7 Register description of pllcfg module

| **Address** | **Def. value** | **Bits** | **Name** | **Description** |
| --- | --- | --- | --- | --- |
| 0x0020 | 0000 | 15-0 | Reserved |  |
| 0x0021 | 0001 |  | **PLL configuration status** | |
| 15-4 | Reserved |  |
| 3 | **AUTO\_PHCFG\_ERR** | Auto phase configuration error status: |
| 0 – no error |
| 1 – Error |
| 2 | **AUTO\_PHCFG\_DONE** | Auto phase configuration status: |
| 0 – Not done |
| 1 – Done |
| 1 | **BUSY** | PLL reconfiguration busy status: |
| 0 – Idle |
| 1 – Busy |
| 0 | **DONE** | PLL configuration status: |
| 0 – Not done |
| 1 – Done |
| 0x0022 | 0000 |  | **PLL lock status** | |
| 15-2 | Reserved |  |
| 1 | **PLL\_LOCK** | RX PLL: |
| 0 – No lock |
| 1 – Locked |
| 0 | TX PLL: |
| 0 – No lock |
| 1 – Locked |
| 0x0023 | 0000 |  | **PLL control** | |
| 15 | Reserved |  |
| 14 | **PHCFG\_MODE** | PLL phase configuration mode: |
| 0 - Manual |
| 1 - AUTO |
| 13 | **PHCFG\_UpDn** | Phase shift direction: |
| 0 - Down |
| 1 - Up |
| 12-8 | **CNT\_IND** | Counter index for phase shift: |
| 0000 - All output counters |
| 0001 - M counter |
| 0010 - C0 counter |
| 0011 - C1 counter |
| 7-3 | **PLL\_IND** | PLL index for reconfiguration: |
| 0000 - TX PLL |
| 0001 - RX PLL |
| Do not use other index values |
| 2 | **PLLRST\_START** | Reset bit for PLL: |
| 0 - Reset inactive |
| 0 to 1 transition triggers reset for PLL with selected index |
| 1 | **PHCFG\_START** | Phase shift start: |
| 0 - Phase shift process inactive |
| 0 to 1 - transition triggers phase shift process for PLL with selected indexes |
| 0 | **PLLCFG\_START** | PLL reconfiguration start: |
| 0 - Phase shift process inactive |
| 0 to 1 - transition triggers phase shift process for PLL with selected indexes |
| 0x0024 | 0000 |  | **PLL reconfiguration settings** | |
| 15-0 | **CNT\_PHASE** | Counter phase value |
| 0x0025 | 01F0 | 15 | Reserved |  |
| 14-11 | **PLLCFG\_BS** | Bandwidth setting (Not used) |
| 10-8 | **CHP\_CURR** | PLL charge Pump Current (1) |
| 7 | **PLLCFG\_VCODIV** | PLL VCO division value |
| 0 = 2 |
| 1 = 1 |
| 6-2 | **PLLCFG\_LF\_RES** | PLL Loop filter resistance (1) |
| 1-0 | **PLLCFG\_LF\_CAP** | PLL Loop filter capacitance (1) |
| 0x0026 | 0001 | 15-4 | Reserved | Counter bypass and odd division control bits (1) |
| 3 | **M\_ODDDIV** |
| 2 | **M\_BYP** |
| 1 | **N\_ODDDIV** |
| 0 | **N\_BYP** |
| 0x0027 | 555A | 15 | **C7\_ODDDIV** |
| 14 | **C7\_BYP** |
| 13 | **C6\_ODDDIV** |
| 12 | **C6\_BYP** |
| 11 | **C5\_ODDDIV** |
| 10 | **C5\_BYP** |
| 9 | **C4\_ODDDIV** |
| 8 | **C4\_BYP** |
| 7 | **C3\_ODDDIV** |
| 6 | **C3\_BYP** |
| 5 | **C2\_ODDDIV** |
| 4 | **C2\_BYP** |
| 3 | **C1\_ODDDIV** |
| 2 | **C1\_BYP** |
| 1 | **C0\_ODDDIV** |
| 0 | **C0\_BYP** |
| 0x0028 | 5555 | 15 | **C15\_ODDDIV** |
| 14 | **C15\_BYP** |
| 13 | **C14\_ODDDIV** |
| 12 | **C14\_BYP** |
| 11 | **C13\_ODDDIV** |
| 10 | **C13\_BYP** |
| 9 | **C12\_ODDDIV** |
| 8 | **C12\_BYP** |
| 7 | **C11\_ODDDIV** |
| 6 | **C11\_BYP** |
| 5 | **C10\_ODDDIV** |
| 4 | **C10\_BYP** |
| 3 | **C9\_ODDDIV** |
| 2 | **C9\_BYP** |
| 1 | **C8\_ODDDIV** |
| 0 | **C8\_BYP** |
| 0x0029 |  | 15-0 | Reserved |  |
| 0x002A | 0000 | 15-8 | **N\_HCNT[15:8]** | N counter values (1) |
| 7-0 | **N\_LCNT[7:0]** |
| 0x002B | 0000 | 15-8 | **M\_HCNT[15:8]** | M counter values (1) |
| 7-0 | **M\_LCNT[7:0]** |
| 0x002C | 0000 | 15-0 | **M\_FRAC[15:0]** | M fractional counter values (Only for fractional PLL) (1) |
| 0x002D | 0000 | 15-0 | **M\_FRAC[31:16]** |
| 0x002E | 0000 | 15-8 | **C0\_HCNT[15:8]** | C0 counter values (1) |
| 7-0 | **C0\_LCNT[7:0]** |
| 0x002F | 0000 | 15-8 | **C1\_HCNT[15:8]** | C1 counter values (1) |
| 7-0 | **C1\_LCNT[7:0]** |
| 0x0030 | 0000 | 15-8 | **C2\_HCNT[15:8]** | C2counter values (1) |
| 7-0 | **C2\_LCNT[7:0]** |
| 0x0031 | 0000 | 15-8 | **C3\_HCNT[15:8]** | C3 counter values (1) |
| 7-0 | **C3\_LCNT[7:0]** |
| 0x0032 | 0000 | 15-8 | **C4\_HCNT[15:8]** | C4 counter values (1) |
| 7-0 | **C4\_LCNT[7:0]** |
| 0x0033 | 0000 | 15-8 | **C5\_HCNT[15:8]** | C5 counter values (1) |
| 7-0 | **C5\_LCNT[7:0]** |
| 0x0034 | 0000 | 15-8 | **C6\_HCNT[15:8]** | C6 counter values (1) |
| 7-0 | **C6\_LCNT[7:0]** |
| 0x0035 | 0000 | 15-8 | **C7\_HCNT[15:8]** | C7 counter values (1) |
| 7-0 | **C7\_LCNT[7:0]** |
| 0x0036 | 0000 | 15-8 | **C8\_HCNT[15:8]** | C8 counter values (1) |
| 7-0 | **C8\_LCNT[7:0]** |
| 0x0037 | 0000 | 15-8 | **C9\_HCNT[15:8]** | C9 counter values (1) |
| 7-0 | **C9\_LCNT[7:0]** |
| 0x0038 |  | 15-0 | Reserved | Reserved for C10-C15 counter values |
| 0x0039 |  | 15-0 | Reserved |
| 0x003A |  | 15-0 | Reserved |
| 0x003B |  | 15-0 | Reserved |
| 0x003C |  | 15-0 | Reserved |
| 0x003D |  | 15-0 | Reserved |
| 0x003E | 0FFF |  | **Auto phase shift options** | |
|  | **AUTO\_PHCFG\_SMPLS** | Samples to compare in auto phase shift mode |
| 0x003F | 0002 |  | **AUTO\_PHCFG\_STEP** | Step size for auto phase |

Note 1: For detailed description see “Cyclone IV Device Handbook”, Chapter 5. Clock Networks and PLLs in Cyclone IV Devices.

### Registers of tstcfg module

Table 8 Register description of tstcfg module

| **Address** | **Def. value** | **Bits** | **Type** | **Name** | **Description** |
| --- | --- | --- | --- | --- | --- |
| 0x0060 | 00F0 |  | **SPI signature** | | |
| 15-8 |  | Reserved |  |
| 7-4 | R | **SPI\_SIGN\_REZULT** | Inverted bits from SPI\_SIGN register |
| 3-0 | R/W | **SPI\_SIGN** | SPI module test register. |
|  |  |  | **Test enable** | | |
| 0x0061 | 0000 | 15-6 |  | Reserved |  |
| 5 | R/W | **DDR2\_2\_TST\_EN** | DDR2\_2 memory test: |
| 0 - Disabled **(Default)** |
| 1 - Enabled |
| 4 | R/W | **DDR2\_1\_TST\_EN** | DDR2\_2 memory test: |
| 0 - Disabled **(Default)** |
| 1 - Enabled |
| 3 | R/W | **ADF\_TST\_EN** | Phase detector test: |
| 0 - Disabled **(Default)** |
| 1 - Enabled |
| 2 | R/W | **VCTCXO\_TST\_EN** | VCTCXO test: |
| 0 - Disabled **(Default)** |
| 1 - Enabled |
| 1 | R/W | **Si5351C\_TST\_EN** | Si5351C clock test: |
| 0 - Disabled **(Default)** |
| 1 - Enabled |
| 0 | R/W | **FX3\_PCLK\_TST\_EN** | FX3 PCLK clock test: |
| 0 - Disabled **(Default)** |
| 1 - Enabled |
| 0x0062 |  |  |  | Reserved |  |
| 0x0063 | 0000 |  | **Error insertion** | | |
| 15-6 |  | Reserved |  |
| 5 | R/W | **DDR2\_2\_TST\_FRC\_ERR** | DDR2\_2 insert error to memory test: |
| 0 - Disabled **(Default)** |
| 1 - Enabled |
| 4 | R/W | **DDR2\_1\_TST\_FRC\_ERR** | DDR2\_1 insert error to memory test: |
| 0 - Disabled **(Default)** |
| 1 - Enabled |
| 3 | R/W | **ADF\_TST\_FRC\_ERR** | Insert error to phase detector test: |
| 0 - Disabled **(Default)** |
| 1 - Enabled |
| 2 | R/W | **VCTCXO\_TST\_FRC\_ERR** | Insert error to VCTCXO test: |
| 0 - Disabled **(Default)** |
| 1 - Enabled |
| 1 | R/W | **Si5351C\_TST\_FRC\_ERR** | Insert error to Si5351C clock test: |
| 0 - Disabled **(Default)** |
| 1 - Enabled |
| 0 | R/W | **FX3\_PCLK\_TST\_FRC\_ERR** | Insert error to FX3 PCLK clock test: |
| 0 - Disabled **(Default)** |
| 1 - Enabled |
| 0x0064 |  |  |  | Reserved |  |
| 0x0065 | 0000 |  | **Test status** | | |
| 15-6 |  | Reserved |  |
| 5 | R | **DDR2\_2\_TST\_CMPLT** | DDR2\_2 test status: |
| 0 - Not completed |
| 1 - Completed |
| 4 | R | **DDR2\_1\_TST\_CMPLT** | DDR2\_1test status: |
| 0 - Not completed |
| 1 - Completed |
| 3 | R | **ADF\_TST\_CMPLT** | Phase detector test status: |
| 0 - Not completed |
| 1 - Completed |
| 2 | R | **VCTCXO\_TST\_CMPLT** | VCTCXO test status: |
| 0 - Not completed |
| 1 - Completed |
| 1 | R | **Si5351C\_TST\_CMPLT** | Si5351C clock test status: |
| 0 - Not completed |
| 1 - Completed |
| 0 | R | **FX3\_PCLK\_TST\_CMPLT** | FX3 PCLK clock test status: |
| 0 - Not completed |
| 1 - Completed |
| 0x0066 |  |  |  | Reserved |  |
|  |  |  | **Test results** | | |
| 0x0067 | 0000 | 15-6 |  | Reserved |  |
| 5 | R | **DDR2\_2\_TST\_REZ** | DDR2\_2 test result: |
| 0 - Fail |
| 1 - Pass |
| 4 | R | **DDR2\_1\_TST\_REZ** | DDR2\_1 test result: |
| 0 - Fail |
| 1 - Pass |
| 3 | R | **ADF\_TST\_REZ** | Not used |
| 2 | R | **VCTCXO\_TST\_REZ** | Not used |
| 1 | R | **Si5351C\_TST\_REZ** | Not used |
| 0 | R | **FX3\_PCLK\_TST\_REZ** | Not used |
|  |  |  | Clock test counter values | | |
| 0x0068 |  |  |  | Reserved |  |
| 0x0069 |  |  | R | **FX3\_CLK\_CNT** | FX3 PCLK clock counter value |
| 0x006A |  |  | R | **Si5351C\_CLK0\_CNT** | Si5351C CLK0 counter value |
| 0x006B |  |  | R | **Si5351C\_CLK1\_CNT** | Si5351C CLK1 counter value |
| 0x006C |  |  | R | **Si5351C\_CLK2\_CNT** | Si5351C CLK2 counter value |
| 0x006D |  |  | R | **Si5351C\_CLK3\_CNT** | Si5351C CLK3 counter value |
| 0x006E |  |  |  | Reserved |  |
| 0x006F |  |  | R | **Si5351C\_CLK5\_CNT** | Si5351C CLK5 counter value |
| 0x0070 |  |  | R | **Si5351C\_CLK6\_CNT** | Si5351C CLK6 counter value |
| 0x0071 |  |  | R | **Si5351C\_CLK7\_CNT** | Si5351C CLK7 counter value |
| 0x0072 |  |  | R | **LMK\_CLK\_CNT\_L** | LMK clock counter value |
| 0x0073 |  |  | R | **LMK\_CLK\_CNT\_H** |
| 0x0074 |  |  | R | **ADF\_CNT** | ADF transition count value |
| 0x0075 |  |  |  | Reserved |  |
|  |  |  | **DDR2\_1 detailed test results 1** | | |
| 0x0076 |  | 15-3 |  | Reserved |  |
| 2 | R | **DDR2\_1\_TST\_FAIL** | DDR2\_1 test result: |
| 0 - Test not completed |
| 1 - Fail |
| 1 | R | **DDR2\_1\_TST\_PASS** | DDR2\_1 test result: |
| 0 - Test not completed |
| 1 - Pass |
| 0 | R | **DDR2\_1\_TST\_CMPLT** | DDR2\_1 test result: |
| 0 - Test not completed |
| 1 - Test complete |
| 0x0077 |  |  | **DDR2\_1 detailed test results 2** | | |
| 15-0 | R | **DDR2\_1\_PNF\_PER\_BIT\_L** | DDR2\_1 data [15:0] bus pas not fail per bit: |
| 0 - Fail |
| 1 - Pass |
| 0x0078 |  |  | **DDR2\_1 detailed test results 3** | | |
| 15-0 | R | **DDR2\_1\_PNF\_PER\_BIT\_H** | DDR2\_1 data [31:16] bus pas not fail per bit: |
| 0 - Fail |
| 1 - Pass |
| 0x0079 |  | 15-0 |  | Reserved |  |
| 0x007A |  |  | **DDR2\_2 detailed test results 1** | | |
| 15-3 |  | Reserved |  |
| 2 | R | **DDR2\_2\_TST\_FAIL** | DDR2\_2 test result: |
| 0 - Test not completed |
| 1 - Fail |
| 1 | R | **DDR2\_2\_TST\_PASS** | DDR2\_2 test result: |
| 0 - Test not completed |
| 1 - Pass |
| 0 | R | **DDR2\_2\_TST\_CMPLT** | DDR2\_2 test result: |
| 0 - Test not completed |
| 1 - Test complete |
| 0x007B |  |  | **DDR2\_2 detailed test results 2** | | |
| 15-0 | R | **DDR2\_2\_PNF\_PER\_BIT\_L** | DDR2\_2 data [15:0] bus pas not fail per bit: |
| 0 - Fail |
| 1 - Pass |
| 0x007C |  |  | **DDR2\_2 detailed test results 3** | | |
| 15-0 | R | **DDR2\_2\_PNF\_PER\_BIT\_H** | DDR2\_2 data [31:16] bus pas not fail per bit: |
| 0 - Fail |
| 1 - Pass |
| 0x007D | AAAA |  | **TX test pattern 1** | | |
| 15-0 | R/W | **TX\_TST\_I** | TX test pattern I sample value |
| 0x007E | 5555 |  | **TX test pattern 2** | | |
| 15-0 | R/W | **TX\_TST\_Q** | TX test pattern Q sample value |
| 0x007F |  | 15-0 |  | Reserved |  |

### Registers of periphcfg module

Table 9 Register description of periphcfg module

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Address** | **Def. value** | **Bits** | **Type** | **Name** | **Description** |
| 0x00C0 | FFFF |  | **Board GPIO control 1** | | |
| 15-8 |  | Reserved |  |
| 7-0 | **R/W** | **BOARD\_GPIO\_OVRD** | GPIO control override (each bit controls corresponding GPIO): |
| 0 - Dedicated function |
| 1 - Overridden by user **(Default)** |
| 0x00C1 |  | 15-0 |  | Reserved for GPIO |  |
| 0x00C2 | 0000 |  | **Board GPIO control 2** | | |
| 15-8 |  | Reserved |  |
| 7-0 | **R** | **BOARD\_GPIO\_RD** | GPIO read value (each from corresponding GPIO): |
| 0 - Low level |
| 1 - High level |
| 0x00C3 |  | 15-0 |  | Reserved for GPIO |  |
| 0x00C4 | 0000 |  | **Board GPIO control 3** | | |
| 15-8 |  | Reserved |  |
| 7-0 | **R/W** | **BOARD\_GPIO\_DIR** | Onboard GPIO direction (each bit controls corresponding GPIO): |
| 0 - Input **(Default)** |
| 1 - Output |
| 0x00C5 |  | 15-0 |  | Reserved for GPIO |  |
| 0x00C6 | 0000 |  | **Board GPIO control 4** | | |
| 15-8 |  | Reserved |  |
| 7-0 | **R/W** | **BOARD\_GPIO\_VAL** | GPIO output value (each bit controls corresponding GPIO): |
| 0 - Low level |
| 1 - High level |
| 0x00C7 |  | 15-0 |  | Reserved for GPIO |  |
| 0x00C8 | 0000 | 15-0 |  | **PERIPH\_INPUT\_RD\_0** | Not used |
| 0x00C9 | 0000 | 15-0 |  | **PERIPH\_INPUT\_RD\_1** | Not used |
| 0x00CA |  | 15-0 |  | Reserved |  |
| 0x00CB |  | 15-0 |  | Reserved |  |
| 0x00CC | 0000 |  | **Board peripheral control 1** | | |
| 15-1 |  | **PERIPH\_OUTPUT\_OVRD\_0** | Not used |
| 0 | **R/W** | Fan control override: |
| 0 - Dedicated function **(Default)** |
| 1 - User controlled |
| 0x00CD | 0000 |  | **Board peripheral control 1** | | |
| 15-1 |  | **PERIPH\_OUTPUT\_VAL\_0** | Not used |
| 0 | **R/W** | Fan control pin: |
| 0 - OFF **(Default)** |
| 1- ON |
| 0x00CE | 0000 | 15-0 |  | **PERIPH\_OUTPUT\_OVRD\_1** | Not used |
| 0x00CF | 0000 | 15-0 |  | **PERIPH\_OUTPUT\_VAL\_1** | Not used |
| 0x00D0 |  | 15-0 |  | Reserved |  |
| 0x00D1 |  | 15-0 |  | Reserved |  |
| 0x00D2 |  | 15-0 |  | Reserved |  |
| 0x00D3 |  | 15-0 |  | Reserved |  |
| 0x00D4 |  | 15-0 |  | Reserved |  |
| 0x00D5 |  | 15-0 |  | Reserved |  |
| 0x00D6 |  | 15-0 |  | Reserved |  |
| 0x00D7 |  | 15-0 |  | Reserved |  |
| 0x00D8 |  | 15-0 |  | Reserved |  |
| 0x00D9 |  | 15-0 |  | Reserved |  |
| 0x00DA |  | 15-0 |  | Reserved |  |
| 0x00DB |  | 15-0 |  | Reserved |  |
| 0x00DC |  | 15-0 |  | Reserved |  |
| 0x00DD |  | 15-0 |  | Reserved |  |
| 0x00DE |  | 15-0 |  | Reserved |  |
| 0x00DF |  | 15-0 |  | Reserved |  |

## FT601 FIFO interface – FT601\_top

Provides data transfer between external FT601 USB 3.0 peripheral controller and FPGA trough FIFO interface (See <http://www.ftdichip.com/Products/ICs/FT600.html> for documentation).

All data exchange between FT601\_top module and other FPGA logic is done through FIFO buffers. Module FT601\_top constantly monitors FT601 FIFO status flags and all FIFO buffers. For example, internal logic writes IQ stream packets containing 4kB data to FIFO buffer through EP83 ports. Once FT601\_arb module detects that EP83 FIFO buffers contains 4kB data and FT601 FIFO flags indicate that FT601 controller is ready, all data is read from FIFO buffer and written to FT601 controller trough FT601 FIFO interface.



Figure 4 FT601\_top block diagram

Table 10 Description of FT601\_top instances

| **Instance** | **Description** |
| --- | --- |
| FT601 | Provides data transfer between FT601 FIFO interface and internal FIFO buffers. |
| FT601\_arb | Data transfer arbiter module. Decides when and what transfer should occur. |
| fifo\_inst (EP83) | Stream endpoint FIFO buffer of 16kB size. |
| two\_fifo\_inst (EP03) | Stream endpoint FIFO buffer of 6kB size. |
| fifo\_inst (EP82) | Control endpoint FIFO buffer of 1kB size. |
| fifo\_inst (EP02) | Control endpoint FIFO buffer of 0.5kB size. |

Table 11 FT601\_top module parameters

| **Parameter** | **Type** | **Default** | **Description** |
| --- | --- | --- | --- |
| FT601 FIFO Bus parameters | | | |
| FT\_data\_width | integer | 32 | FT601 data width |
| FT\_be\_width | integer | 4 | FT601 byte enable width |
| Internal FIFO buffers | | | |
| EP02\_rdusedw\_width | integer | 9 | EP02 FIFO read used words size (29-1= 256 words) |
| EP02\_rwidth | integer | 32 | EP02 FIFO read word size |
| EP82\_wrusedw\_width | integer | 9 | EP82 FIFO write used words size (29-1= 256 words) |
| EP82\_wwidth | integer | 32 | EP82 FIFO write word size |
| EP82\_wsize | integer | 64 | EP82 packet size in bytes, has to be multiple of 4 bytes |
| EP03\_rdusedw\_width | integer | 9 | EP03 FIFO read used words size (29-1= 256 words) |
| EP03\_rwidth | integer | 128 | EP03 FIFO read word size |
| EP83\_wrusedw\_width | integer | 12 | EP82 FIFO write used words size (212-1= 2048 words) |
| EP83\_wwidth | integer | 64 | EP82 FIFO write word size |
| EP83\_wsize | integer | 2048 | EP83 packet size in bytes, has to be multiple of 4 bytes |

Table 12 FT601\_top module ports

|  |  |  |  |
| --- | --- | --- | --- |
| **Port** | **Type** | **Width** | **Description** |
| clk | in | 1 | Clock 100 Mhz |
| reset\_n | in | 1 | Reset active low |
| FTDI external ports | | | |
| FT\_wr\_n | out | 1 | FT601 FIFO bus |
| FT\_rxf\_n | in | 1 |
| FT\_data | inout | FT\_data\_width |
| FT\_be | inout | FT\_be\_width |
| FT\_txe\_n | in | 1 |
| Control endpoint FIFO PC->FPGA | | | |
| EP02\_rdclk | in | 1 | Read clock |
| EP02\_rd | in | 1 | Read request |
| EP02\_rdata | out | EP02\_rwidth | Read data |
| EP02\_rempty | out | 1 | Read empty |
| Control endpoint FIFO FPGA->PC | | | |
| EP82\_wclk | in | 1 | Write clock |
| EP82\_aclrn | in | 1 | Asynchronous clear, active low |
| EP82\_wr | in | 1 | Write request |
| EP82\_wdata | in | EP82\_wwidth | Write data |
| EP82\_wfull | out | 1 | Write full |
| Stream endpoint FIFO PC->FPGA | | | |
| EP03\_aclrn\_0 | in | 1 | Asynchronous clear FT601 FIFO side, active low. |
| EP03\_aclrn\_1 | in | 1 | Asynchronous clear stream side, active low. |
| EP03\_rdclk | in | 1 | Read clock |
| EP03\_rd | in | 1 | Read request |
| EP03\_rdata | out | EP03\_rwidth | Read data |
| EP03\_rempty | out | 1 | Read empty |
| EP03\_rusedw | out | EP03\_rdusedw\_width | Red used words |
| Stream endpoint FIFO FPGA->PC | | | |
| EP83\_wclk | in | 1 | Write clock |
| EP83\_aclrn | in | 1 | Asynchronous clear, active low |
| EP83\_wr | in | 1 | Write request |
| EP83\_wdata | in | EP03\_rdusedw\_width | Write data |
| EP83\_wfull | out | 1 | Write full |
| EP83\_wrusedw | out | EP83\_wrusedw\_width | Write used words |

## LMS7002 Receive and transmit interface – rxtx\_top

Main function of rxtx\_top module is for receive and transmit IQ samples from/to LMS7002 chip and provide IQ sample synchronization. See **Figure 5** for block diagram and **Table 13** for instance description.



Figure 5 rxtx\_top block diagram

Table 13 Description of rxtx\_top instances

| **Parameter** | **Type** | **Default** | **Description** |
| --- | --- | --- | --- |
| DEV\_FAMILY | string | MAX 10 | Device family |
| TX parameters | | | |
| TX\_IQ\_WIDTH | integer | 12 | TX IQ sample width |
| TX\_N\_BUFF | integer | 4 | TX number of buffers, 2,4 valid values |
| TX\_IN\_PCT\_SIZE | integer | 4096 | TX packet size in bytes |
| TX\_IN\_PCT\_HDR\_SIZE | integer | 16 | TX packet header size in bytes |
| TX\_IN\_PCT\_DATA\_W | integer | 128 | TX packet read data width |
| TX\_IN\_PCT\_RDUSEDW\_W | integer | 11 | TX packet read used words width |
| TX\_OUT\_PCT\_DATA\_W | integer | 64 | TX output packet data width |
| RX parameters | | | |
| RX\_IQ\_WIDTH | integer | 12 | RX IQ sample width |
| RX\_INVERT\_INPUT\_CLOCKS | string | OFF | Clock invert option on LMS\_DIQ2 interface |
| RX\_SMPL\_BUFF\_RDUSEDW\_W | integer | 11 | RX sample buffer read used words width. Words=211-1 |
| RX\_PCT\_BUFF\_WRUSEDW\_W | integer | 12 | RX packet buffer read used words width. Words=212-1 |

Table 14 rxtx\_top parameters description

| **Parameter** | **Type** | **Default** | **Description** |
| --- | --- | --- | --- |
| DEV\_FAMILY | string | Cyclone IV E | Device family |
| TX parameters | | | |
| TX\_IQ\_WIDTH | integer | 12 | TX IQ sample width |
| TX\_N\_BUFF | integer | 4 | TX number of buffers, 2,4 valid values |
| TX\_IN\_PCT\_SIZE | integer | 4096 | TX packet size in bytes |
| TX\_IN\_PCT\_HDR\_SIZE | integer | 16 | TX packet header size in bytes |
| TX\_IN\_PCT\_DATA\_W | integer | 128 | TX packet read data width |
| TX\_IN\_PCT\_RDUSEDW\_W | integer | 11 | TX packet read used words width |
| TX\_OUT\_PCT\_DATA\_W | integer | 64 | TX output packet data width |
| RX parameters | | | |
| RX\_IQ\_WIDTH | integer | 12 | RX IQ sample width |
| RX\_INVERT\_INPUT\_CLOCKS | string | OFF | Clock invert option on LMS\_DIQ2 interface |
| RX\_SMPL\_BUFF\_RDUSEDW\_W | integer | 11 | RX sample buffer read used words width. Words=211-1 |
| RX\_PCT\_BUFF\_WRUSEDW\_W | integer | 12 | RX packet buffer rd used words width. Words=212-1 |

Table 15 rxtx\_top port description

|  |  |  |  |
| --- | --- | --- | --- |
| **Port** | **Type** | **Width** | **Description** |
| Configuration memory ports | | | |
| from\_fpgacfg | in | t\_FROM\_FPGACFG; | Configuration registers bus |
| to\_tstcfg\_from\_rxtx | out | t\_TO\_TSTCFG\_FROM\_RXTX; |
| from\_tstcfg | in | t\_FROM\_TSTCFG; |
| TX path | | | |
| tx\_clk | in | 1 | TX interface clock |
| tx\_clk\_reset\_n | in | 1 | TX interface reset, active low |
| tx\_pct\_loss\_flg | out | 1 | TX packet loss flag, 0 - No packet loss, 1 - Packet losst. |
| tx\_txant\_en | out | 1 | TX transmit flag. 0 - No transmission, 1 - TX is transmitting samples |
| TX interface data | | | |
| tx\_DIQ | out | TX\_IQ\_WIDTH | TX samples |
| tx\_fsync | out | 1 | TX sync signal |
| TX FIFO read ports | | | |
| tx\_in\_pct\_reset\_n\_req | out | 1 | TX packet buffer reset request, active low |
| tx\_in\_pct\_rdreq | out | 1 | TX packet buffer read request |
| tx\_in\_pct\_data | in | TX\_IN\_PCT\_DATA\_W | TX packet buffer read data |
| tx\_in\_pct\_rdempty | in | 1 | TX packet buffer read empty |
| tx\_in\_pct\_rdusedw | in | TX\_IN\_PCT\_RDUSEDW\_W | TX packet buffer read used words |
| RX path | | | |
| rx\_clk | in | 1 | RX interface clock |
| rx\_clk\_reset\_n | in | 1 | RX interface reset, active low |
| Rx interface data | | | |
| rx\_DIQ | in | RX\_IQ\_WIDTH | RX IQ samples |
| rx\_fsync | in | 1 | RX IQ sync signal |
| Packet FIFO ports | | | |
| rx\_pct\_fifo\_aclrn\_req | out | 1 | RX packet buffer reset request, active low |
| rx\_pct\_fifo\_wusedw | in | RX\_PCT\_BUFF\_WRUSEDW\_W | RX packet buffer write used words |
| rx\_pct\_fifo\_wrreq | out | 1 | RX packet buffer write request |
| rx\_pct\_fifo\_wdata | out | 64 | RX packet buffer write data |
| Sample compare | | | |
| rx\_smpl\_cmp\_start | in | 1 | RX interface sample compare. 0 - disabled, 1-enabled |
| rx\_smpl\_cmp\_length | in | 16 | RX interface number of samples to compare. |
| rx\_smpl\_cmp\_done | out | 1 | RX outterface sample compare done. 0 - not done, 1-done |
| rx\_smpl\_cmp\_err | out | 1 | RX outterface sample compare status. 0 - no error, 1 - error |

### Receive interface – rx\_path\_top

Once rx\_path\_top **Figure 6** is enabled diq2fifo and data2packets modules starts continuously packing IQ samples into 4kB packets. For packet structure see [Stream protocol](https://github.com/myriadrf/LimeSuite/blob/master/docs/StreamProtocol.pdf) document.

Packets are written to 16kB EP83 FIFO buffer to maintain continuous data flow in short periods when USB3.0 host cannot accept data. If USB3.0 host halts data transfer for longer time period and four packets are buffered into 16kB buffer, FIFO full condition arises and other packets are dropped. When host starts to receive data after FIFO full condition, host should expect to receive those four buffered packets.

Module rx\_path\_top provides two 64bit sample counters. One is for TX logic – tx\_path\_top. TX logic uses this counter to synchronize transmitted LMS\_DQ1 samples with received LMS\_DIQ2 samples. Other is used for LMS\_DI2 samples packing into 4kB packets.

When rx\_path\_top is enabled diq2fifo module starts to collect IQ samples from LMS\_DIQ2 bus, collected samples are written to FIFO buffer and each write enables smpl\_cnt:inst4 module to increase its counter value. This means that counter value increases in same continuous rate as IQ sample rate.

Module smpl\_cnt:inst3 is used for LMS\_DI2 samples packing into 4kB packets. Module data2packets reads IQ samples in bursts from FIFO buffer, each read enables smpl\_cnt:inst3 module to increase its counter value. One read burst fills one 4kB packet and there are some idle cycles between bursts.



Figure 6 rx\_path\_top block diagram

Table 16 rx\_path\_top inctance description

| **Instance** | **Description** |
| --- | --- |
| diq2fifo | Captures IQ samples and writes to FIFO buffer. |
| fifo\_inst | FIFO buffer for storing samples. |
| data2packets | Module for packing IQ samples to 4kB packets. |
| smpl\_cnt:inst3 | Sample counter for tx\_path\_top. |
| smpl\_cnt:inst4 | Sample counter for data2packets module. |

### Transmit interface – tx\_path\_top

Transmit module tx\_path\_top reads IQ samples from EP03 FIFO buffer packed in 4kB packets. Packet header (see [Stream protocol](https://github.com/myriadrf/LimeSuite/blob/master/docs/StreamProtocol.pdf) document) contains sample number (or so-called time stamp) at which packet should be transmitted.

By using sample numbers from rx\_path\_top and received sample numbers in packet header transmitted IQ samples can be synchronized with received IQ samples.

Module p2d\_wr\_fsm separates packet header and payload. Packet payload is written into one of four 4kB FIFO buffers located in packets2data module and packet header is stored in p2d\_rd module. This module can work in two modes:

* **Synchronization enabled** - module compares received sample number from packet header and sample number from rx\_path\_top. When sample number from received packet is equal to sample number of rx\_path\_top module (this means that it is time to send TX packet), read process begins and IQ samples are transmitted to LMS\_DIQ1 interface. When sample number from received packet is greater than sample number of rx\_path\_top module (this means that received packet should be sent after some time) p2d\_rd waits until those sample number will be equal. When sample number from received packet is less than sample number of rx\_path\_top module (this means that packet arrived too late) corresponding FIFO buffer is cleared.
* **Synchronization disabled** – module does not compare sample numbers and every received packet is transmitted to LMS\_DIQ1 interface.

Block diagram can be found in **Figure 7** and instance description in **Table 17**.



Figure 7 tx\_path\_top block diagram

Table 17 tx\_path\_top instance description

| **Instance** | **Description** |
| --- | --- |
| packets2data\_top | Wrapper file |
| packets2data | Wrapper file |
| p2d\_wr\_fsm | Module reads packets from EP03 buffer and places to one of the 4kB FIFO buffers in increasing order and stores corresponding sample number from packet header. |
| p2d\_rd | Module checks one of the FIFO buffers if it is filled with samples in increasing order. When buffer is ready depending on received sample number from packet header and sample number from rx\_path\_top module buffer can be cleared or IQ sample reading begins. |
| fifo\_inst | FIFO buffer |
| fifo2diq | Module reads samples from FIFO buffer and writes to LMS\_DIQ1 interface. |
| sync\_fifo\_rw | Dual clock FIFO buffer for clock domain crossing. |
| bit\_unpack\_64 | Depending on mode selection samples are unpacked (see [Stream protocol](https://github.com/myriadrf/LimeSuite/blob/master/docs/StreamProtocol.pdf) document). |

## General periphery – general\_periph\_top

General periphery - general\_periph\_top module is responsible for controlling on board periphery such as LED, GPIO and Fan, default functions can be found in **Table 18**. Also default function can be overridden by internal registers see chapter **3.3 Softcore processor – cpu**.

Table 18 Default functions of LEDS, GPIO and fan

| **Schematic name** | **Board label** | **Type** | **Description** |
| --- | --- | --- | --- |
| FPGA\_LED | FPGA1 | Clock status | Blinking indicates presence of TCXO clock.  Colour indicates status of FPGA PLLs that are used for LMS digital interface clocking: Green – both PLLs are locked; Red/Green – at least one PLL is not locked. |
| FPGA\_GPIO0 | FPGA\_GPIO |  | Indicates PLL lock status. 0 – no lock, 1 - locked |
| FPGA\_GPIO1 |  | - |
| FPGA\_GPIO2 |  | - |
| FPGA\_GPIO3 |  | - |
| FPGA\_GPIO4 |  | - |
| FPGA\_GPIO5 |  | - |
| FPGA\_GPIO6 |  | - |
| FPGA\_GPIO7 |  | - |
| FAN\_CTRL | FAN |  | Fan control pin. Connected to LM75\_OS temperature sensor pin. |

Block diagram can be found in **Figure 8**, instances are described in **Table 19**. See **Table 20** and **Table 21** for module parameters and port description.



Figure 8 Module general\_periph\_top block diagram

Table 19 Module instance description

| **Instance** | **Description** |
| --- | --- |
| alive | Basic counter to implement blinking on led1. |
| FPGA\_LED\_cntrl | Led1 control module, for showing clock status |
| gpio\_ctrl | GPIO control instance |

Table 20 Module general\_periph\_top parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Type** | **Default** | **Description** |
| DEV\_FAMILY | string | "MAX 10" | FPGA device family name |
| N\_GPIO | integer | 8 | Number of GPIO used |

Table 21 Module general\_periph\_top input and output port description

| **Port** | **Type** | **Width** | **Description** |
| --- | --- | --- | --- |
| clk | in | 1 | Free running clock |
| reset\_n | in | 1 | Asynchronous, active low reset |
| SPI registers (Default address range 0x00C0-0x00DF) | | | |
| periphcfg\_maddress | in | 10 | Address of SPI slave registers |
| periphcfg\_sdin | in | 1 | SPI slave datain |
| periphcfg\_sclk | in | 1 | SPI slave clock |
| periphcfg\_sen | in | 1 | SPI slave select |
| periphcfg\_sdout | out | 1 | SPI slave dataout |
| LED1(Clock and PLL lock status) | | | |
| led1\_pll1\_locked | in | 1 | Lock status from PLL1 |
| led1\_pll2\_locked | in | 1 | Lock status from PLL2 |
| led1\_ctrl | in | 3 | led1\_ctrl[0]–manual LED control enable; led1\_ctrl[1]–red LED enable in manual mode; led1\_ctrl[2]–green LED enable in manual mode; |
| led1\_g | out | 1 | Output to dual colour LED1 pin |
| led1\_r | out | 1 | Output to dual colour LED1 pin |
| LED2(TCXO control status) | | | |
| led2\_clk | in | 1 | Unused |
| led2\_adf\_muxout | in | 1 |
| led2\_dac\_ss | in | 1 |
| led2\_adf\_ss | in | 1 |
| led2\_ctrl | in | 3 |
| led2\_g | out | 1 |
| led2\_r | out | 1 |
| LED3(FX3 and NIOS CPU busy) | | | |
| led3\_g\_in | in | 1 | Unused |
| led3\_r\_in | in | 1 |
| led3\_ctrl | in | 3 |
| led3\_hw\_ver | in | 4 |
| led3\_g | out | 1 |
| led3\_r | out | 1 |
| GPIO | | | |
| gpio\_dir | in | N\_GPIO | GPIO direction control, 0 – input, 1 – output |
| gpio\_out\_val | in | N\_GPIO | GPIO output value when direction is set to output |
| gpio\_rd\_val | out | N\_GPIO | GPIO input value when direction is set to input |
| gpio | inout | N\_GPIO | Connected to GPIO pins |
| Fan control | | | |
| fan\_sens\_in | in | 1 | From temperature sensor |
| fan\_ctrl\_out | out | 1 | To Fan control output |

## 

## LMS7002 module – lms7002\_top

LMS7002 module – lms7002\_top (**Figure 9)** provides required clock sources for LM7002 RX and TX digital interfaces. Inside this module there is dynamically reconfigurable IO delay modules to change clock phase relationship on both LMS DIQ1 and DIQ2 interfaces while FPGA is in user mode. Instance description can be found in **Table 22**.



Figure 9 LMS7002 module – lms7002\_top

Table 22. pll\_top module instance description

| **Instance** | **Description** |
| --- | --- |
| lms7002\_clk | LMS7002 Clock control module with reconfigurable delay on feedback clocks fclk1/fclk2 |
| lms7002\_txiq | Transmit interface for LMS7002 with variable IO delay |
| lms7002\_rxiq | Receive interface for LMS7002 with variable IO delay |
| delay\_ctrl\_top | Delay control module which controls delay for feedback clocks fclk1/fclk2 and delay on DIQ1 bus |

Table 23 lms7002\_top port description

| **Port** | **Type** | **Width** | **Description** |
| --- | --- | --- | --- |
| Free running clock and reset | | | |
| clk | in | 1 | Free running clock |
| reset\_n | in | 1 | Low level reset for all logic inside |
| LMS7002 TX DIQ interface | | | |
| MCLK1 | in | 1 | Transmit master clock from LMS7002 |
| FLCK1 | out | 1 | Transmit feedback clock for LMS7002 |
| ENABLE\_IQSEL1 | out | 1 | Trnasmit IQ select |
| DIQ1\_D | out | 12 | Transmit DIQ bus |
| LMS7002 RX DIQ interface | | | |
| MCLK2 | in | 1 | Recevie master clock from LMS7002 |
| FLCK2 | out | 1 | Receive feedback clock for LMS7002 |
| ENABLE\_IQSEL2 | in | 1 | Receive IQ select |
| DIQ2\_D | in | 12 | Receive DIQ bus |
| Internal logic | | | |
| tx\_clk | out | 1 | Clock for internal transmit logic |
| tx\_diq1\_h | in | 13 | DIQ data from internal logic |
| tx\_diq1\_l | in | 13 | DIQ data from internal logic |
| rx\_clk | out | 1 | Clock for internal receive logic |
| rx\_diq2\_h | out | 13 | DIQ data to internal logic |
| rx\_diq2\_l | out | 13 | DIQ data to internal logic |
| Delay control | | | |
| delay\_en | in | 1 | Delay enable |
| delay\_sel | in | 2 | Delay select --0 FCLK1, 1 - TX\_DIQ(not supported), 2 - FLCK2(not supported), 3 - RX\_DIQ |
| delay\_dir | in | 1 | Delay dir 0 - Decrease delay, 1- increase delay |
| delay\_mode | in | 1 | Delay mode 0 - manual, 1- auto |
| delay\_done | out | 1 | Delay done |
| delay\_error | out | 1 | Delay error |
| Sample compare | | | |
| smpl\_cmp\_en | out | 1 | Sample compare enble, enables comparing of DIQ samples |
| smpl\_cmp\_done | in | 1 | Sample compare done, logic high is asserted when sample compare is done |
| smpl\_cmp\_error | in | 1 | Sample compare error, logic high is asserted when sample compare is done and error occurred |
| smpl\_cmp\_cnt | out | 16 | Number of samples to compare |

## Board test module – tst\_top

Board test module – tst\_top is used to test clock inputs. Separate tests can be enabled and results can be read from internal registers see **3.3.3 Registers of tstcfg module**. Module port description can be found in **Table 24**.

Table 24 tst\_top module port description

|  |  |  |  |
| --- | --- | --- | --- |
| **Port** | **Type** | **Width** | **Description** |
| FX3\_clk | in | 1 | 100MHz reference clock |
| reset\_n | in | 1 | Reset, active low |
| Clock inputs | | | |
| Si5351C\_clk\_0 | in | 1 | Not used |
| Si5351C\_clk\_1 | in | 1 |
| Si5351C\_clk\_2 | in | 1 |
| Si5351C\_clk\_3 | in | 1 |
| Si5351C\_clk\_5 | in | 1 |
| Si5351C\_clk\_6 | in | 1 |
| Si5351C\_clk\_7 | in | 1 |
| LMK\_CLK | in | 1 | Clock buffer |
| ADF\_MUXOUT | in | 1 | Not used |
| To configuration memory | | | |
| to\_tstcfg | out | t\_TO\_TSTCFG | Configuration bus |
| from\_tstcfg | in | t\_FROM\_TSTCFG |

# Examples

In this chapter various examples can be found on how to use gateware.

## Accessing FPGA registers

Internal FPGA registers can be accessed using USB3.0 host via EP02 and EP82 endpoints. See **LMS64C\_protocol** document for protocol structure and description of commands used in examples. See chapter **3.3 Softcore processor – cpu** for internal FPGA register description.

**Read** – 64byte packet containing request command “CMD\_BRDSPI16\_RD” has to be sent to EP02 endpoint and 64 bytes response packet has to be read from EP82 endpoint. Read example reads 0x0000 address Board\_ID register value, which is 0x0011 for LimeSDR-Mini board.

Request – USB3.0 host writes 64B to EP0F:

Address

0000 56 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00

0010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Response – USB3.0 host reads 64B from EP8F:

Address

0000 56 01 01 00 00 00 00 00 00 00 00 0E 00 00 00 00

0010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

**Write** – 64byte packet containing request command “CMD\_BRDSPI16\_WR” has to be sent to EP0F endpoint and 64 bytes response packet has to be read from EP8F endpoint. Write example writes 0x1234 value to 0x00DF address. This register is currently reserved and has no dedicated function.

Request – USB3.0 writes 64B to EP0F:

Address

0000 55 00 01 00 00 00 00 00 00 DF 12 34 00 00 00 00

0010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Response – USB3.0 host reads 64B from EP8F:

Address

0000 55 01 01 00 00 00 00 00 00 00 00 00 00 00 00 00

0010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

## Accessing LMS7002M registers

Configuration memory which is inside LMS7002M can be accessed using USB3.0 host via EP02 and EP82 endpoints. See **LMS64C\_protocol** document for protocol structure and description of commands used in examples. Registers map of LMS7002M can be found in [LMS7002M – Multi-Band, Multi-Standard MIMO, Programming and Calibration Guide](https://github.com/myriadrf/LMS7002M-docs/blob/master/LMS7002M_Programming_and_Calibration_Guide_v31r05.pdf).

**Read** – 64byte packet containing request command “CMD\_LMS7002\_RD” has to be sent to EP02 endpoint and 64 bytes response packet has to be read from EP82 endpoint. Read example reads 0x0020 address register value, which is 0xFFFF by default.

Request – USB3.0 writes 64B to EP0F:

Address

0000 22 00 01 00 00 00 00 00 00 20 00 00 00 00 00 00

0010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Response – USB3.0 host reads 64B from EP8F:

Address

0000 22 01 01 00 00 00 00 00 00 00 FF FF 00 00 00 00

0010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

**Write** – 64byte packet containing request command “CMD\_LMS7002\_WR” has to be sent to EP02 endpoint and 64 bytes response packet has to be read from EP8F endpoint. Write example writes 0xE4E4 value to 0x0024 address.

Request – USB3.0 writes 64B to EP0F:

Address

0000 21 00 01 00 00 00 00 00 00 24 E4 E4 00 00 00 00

0010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Response – USB3.0 host reads 64B from EP8F:

Address

0000 21 01 01 00 00 00 00 00 00 00 00 00 00 00 00 00

0010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

## Periphery control

**LED control -** modify FPGA register as showed in **Table 25** to turn on and change colour of FPGA\_LED.

Table 25 FPGA\_LED2 control example

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **N** | **CMD** | **Address (HEX)** | **Value (HEX)** | **Description** |
| 1 | WR | 001A | 0001 | Override FPGA\_LED control |
| 2 | WR | 001A | 0003 | Turn on FPGA\_LED\_R (red is on, green - off) |
| 3 | WR | 001A | 0005 | Turn on FPGA\_LED\_G (green is on, red - off) |

## Configuring FPGA PLL module

To configure PLL of pll\_top module LMS7002M chip has to be already configured and valid clock sources provided to LMS\_MCLK2 pins. For LMS7002M chip configuration see chapter 4.2 Accessing LMS7002M registers.

Configuration of pll\_top module can be done by accessing FPGA registers see chapter **4.1 Accessing FPGA registers**. For register description see chapter **3.3 Softcore processor – cpu**.

PLL output frequency Fout can be calculated using following equation:

(1); (2); (3);

where *Fref*- PLL reference frequency, *FVCO* – VCO frequency, *FOUT* – Output frequency. See MAX 10 [datasheet](https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/hb/max-10/m10_datasheet.pdf) for allowed frequency ranges.

### PLL module – RX clock configuration (auto phase shift mode)

This example assumes that LMS7002M chip is already configured, outputs 15.36 MHz clock on LMS\_MCLK2 pin and LMS\_DIQ2 interface outputs constant IQ values (I=0xAAA, Q=0x555). See **Table 26** for configuration sequence.

Table 26 rxpll\_top configuration sequence in auto phase shift mode

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **N** | **CMD** | **Address (HEX)** | **Value (HEX)** | **Description** |
| 1 | WR | 0005 | 0000 | Turn off direct clocking |
| 2 | WR | 0025 | 01F0 | Set PLL parameters |
| WR | 0023 | 0000 | Set PLL index to 0 and rest bits to zero |
| 3 | WR | 0023 | 0000 | Set PLL index to 0 and rest bits to zero |
| WR | 0026 | 0000 | N, M division bypass and odd division values. N, M division is not bypassed, odd division values disabled |
| WR | 002A | 0202 | N, count value = 0x02 + 0x02 = 0x04 (4 DEC) |
| WR | 002B | 5050 | M count value = 0x50 + 0x50 = 0xA0 (160 DEC) |
| WR | 002E | 1414 | C0 count value = 0x14 + 0x14 = 0x28 (40 DEC) |
| WR | 002F | 1414 | C1 count value = 0x14 + 0x14 = 0x28 (40 DEC) |
| WR | 0030 | 1414 | C2 count value = 0x14 + 0x14 = 0x28 (40 DEC) |
| WR | 0031 | 1414 | C3 count value = 0x14 + 0x14 = 0x28 (40 DEC) |
| WR | 0027 | 5500 | Counter C0-C7 bypass and odd division control bits. C0, C1, C2, C3 not bypassed, others bypassed. C0, C1, C2, C3 odd division values disabled, others not enabled. |
| WR | 0028 | 5555 | Counter C7-C15 bypass and odd division control bits. All counters are bypassed. |
| WR | 0023 | 0001 | Trigger reconfiguration for PLL index 0. |
| 4 | WR | 0023 | 6500 | Release PLL reconfiguration bit, set PLL index - 0, cnt index - 5, phase shift - up, phase shift mode - auto |
| 0024 | 013F | Phase shift value = 0x013F (319 DEC), represents 360 degrees (range in which auto phase shift is executed) |
| 0023 | 6502 | Trigger auto phase shift for PLL index 0, cnt index 5, phase shift - up, phase shift mode - auto |
|  | RD | 0021 |  | Read PLL configuration status register and wait for configuration done (0x0005) |
| 5 | WR | 0023 | 6500 | Release PLL phase shift bit, set PLL index - 0, cnt index - 5, phase shift - up, phase shift mode - auto |

### PLL module – TX clock configuration (auto phase shift mode)

This example assumes that LMS7002M chip is already configured, outputs 15.36MHz clock on LMS\_MCLK2 pin, LimeLight digital loopback is enabled and FPGA RX clock is already configured. See **Table 27** for configuration sequence.

Table 27 txpll\_top configuration sequence in auto phase shift mode

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **N** | **CMD** | **Address (HEX)** | **Value (HEX)** | **Description** |
| 1 | WR | 000A | 0000 | Stop RX TX stream, clear test pattern bits |
| 2 | WR | 000A | 0200 | Enable TX test pattern |
| 3 | WR | 0005 | 0000 | Turn off direct clocking |
| 4 | WR | 0025 | 01F0 | Set PLL parameters |
| WR | 0023 | 0000 | Set PLL index to 0 and rest bits to zero |
| 5 | WR | 0023 | 0000 | Set PLL index to 0 and rest bits to zero |
| WR | 0026 | 0000 | N, M division bypass and odd division values. N, M division is not bypassed, odd division values disabled |
| WR | 002A | 0202 | N, count value = 0x02 + 0x02 = 0x04 (4 DEC) |
| WR | 002B | 5050 | M count value = 0x50 + 0x50 = 0xA0 (160 DEC) |
| WR | 002E | 1414 | C0 count value = 0x14 + 0x14 = 0x28 (40 DEC) |
| WR | 002F | 1414 | C1 count value = 0x14 + 0x14 = 0x28 (40 DEC) |
| WR | 0030 | 1414 | C2 count value = 0x14 + 0x14 = 0x28 (40 DEC) |
| WR | 0031 | 1414 | C3 count value = 0x14 + 0x14 = 0x28 (40 DEC) |
| WR | 0027 | 5500 | Counter C0-C7 bypass and odd division control bits. C0, C1, C2, C3 not bypassed, others bypassed. C0, C1, C2, C3 odd division values disabled, others - disabled. |
| WR | 0028 | 5555 | Counter C7-C15 bypass and odd division control bits. All counters are bypassed. |
| 6 | WR | 0023 | 6300 | Release PLL reconfiguration bit, set PLL index - 0, cnt index - 3, phase shift - up, phase shift mode - auto |
| WR | 24 | 013F | Phase shift value = 0x013F (319 DEC), represents 360 degrees (range in which auto phase shift is executed) |
| WR | 23 | 6302 | Trigger auto phase shift for PLL index 0, cnt index 3, phase shift - up, phase shift mode - auto |
| RD | 21 |  | Read PLL configuration status register and wait for configuration done (0x0005) |
| 7 | WR | 23 | 6300 | Release PLL phase shift bit, set PLL index - 0, cnt index - 3, phase shift - up, phase shift mode - auto |

## Controlling TX and RX data stream

Data stream can be enabled when LMS7002M chip and FPGA PLL modules are configured. See chapters **4.2 Accessing LMS7002M registers** and **4.4 Configuring FPGA PLL module.**

**To enable TX and RX data stream –** follow FPGA register write sequence described in **Table 28**.

Table 28 enabling TX and RX data stream

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **N** | **CMD** | **Address (HEX)** | **Value (HEX)** | **Description** |
| 1 | WR | 000A | 0000 | Stop data stream |
| 2 | WR | 0009 | 0000 | Clear packet loss and reset timestamp bits. |
| 3 | WR | 0009 | 0003 | Clear packet loss flag and reset timestamp. |
| 4 | WR | 0009 | 0000 | Clear packet loss and reset timestamp bits. |
| 5 |  |  |  | Reset USB3.0 EP02 end EP82. endpoints (Use CMD\_STREAM\_RST command) |
| 6 | WR | 0008 | 102 | Set sample width -12, mode - TRXIQ, DDR - enabled, TRXIQ\_PULSE mode - disabled, packet synchronization - enabled |
| 7 | WR | 0007 | 0001 | Set active channels - 1 |
| 8 | WR | 000A | 0001 | Start stream |

**To disable** **TX and RX data stream –** follow FPGA register write sequence described in **Table 29.**

Table 29 disabling TX and RX data stream

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **N** | **CMD** | **Address (HEX)** | **Value (HEX)** | **Description** |
| 1 | WR | 000A | 0000 | Stop data stream |