FRS Reference Design

Specification

This document could contain technical inaccuracies or typographical errors. TTTech Flexibilis Oy may make changes in the product described in this document at any time.

Please, email comments about this document to support@flexibilis.com.

Trademarks

All trademarks are the property of their respective owners.

Contents

Figures

Tables

Revision History

Rev	Date	Comments
0.1	27.2.2014	Draft
0.2	7.5.2014	Updates from review
1.0	15.5.2014	First release
1.1	19.6.2014	Updates to SW BSP information
1.2	5.12.2014	Adapter / Reference Design updates for 2.9 release
1.3	25.6.2015	License info updated
1.4	1.3.2016	Added address format information
1.5	23.3.2016	Updated according to design changes
1.6	27.6.2016	FES system replaced FRS Redbox in QSYS
1.7	23.4.2021	Minor updates

1 About This Document

This document describes the reference design for Flexibilis Redundant Switch (FRS) [7]. FRS is an Ethernet switch Intellectual Property (IP) core targeted at programmable hardware platforms. The FRS is a variation of Flexibilis Ethernet Switch (FES).

The purpose of the Reference Design is to provide an FRS evaluation platform and it may be used as is or modified as required. The Reference Design implements a HSR/PRP RedBox with support for 1588 PTP ordinary and transparent clocks. The Reference Design is provided for Cyclone IV GX, Cyclone V GX and Cyclone V GT evaluation boards and it is downloadable from Flexibilis website www.flexibilis.com. There is also a Reference Design for Cyclone V SoC evaluation board but it is not covered by this document.

Chapter 2 describes the Reference Design in general. Chapter 3 describes the FPGA part of the Reference Design, i.e. VHDL, IP and QSYS issues. Chapter 4 describes the software included in the Reference Design. Chapter 5 contains abbreviations and chapter 6 references.

1.1 Conventions Used in This Document

Register descriptions in this document follow these rules: Unless otherwise stated, all the bits that activate or enable something are active when their value is 1 and inactive when their value is 0. The explanation of the bit types is the following:

RO = Read Capable Only. The bits marked with RO can be read. Writing to these bits is allowed if not otherwise stated. If writing is allowed, it does not affect the value of the bit.

R/W = Read and Write capable. The bits can be read and written. Writing 1 to the bit makes its value 1. Writing 0 to the bit makes its value 0.

R/C = Read and Clear capable. The bits can be read and cleared. Writing 0 to the bit makes its value 0. Writing 1 does nothing.

R/SC = Read and Self Clear. The bits can be read. After reading bits, the value automatically returns back to 0.

R/W/SC = Read, Write and Self Clear. The bits can be read and written. Writing 0 to the bit does nothing. Writing 1 to the bit makes its value 1 for a while, but after that the value automatically returns back to 0.

The bits marked as *Reserved* should not be written anything but 0, even if they are marked as read capable only, because their function may change in future versions.

Bit and byte order used for 16, 32 and 64 registers is depicted in Figure 1. Leftmost byte is in the lowest address.

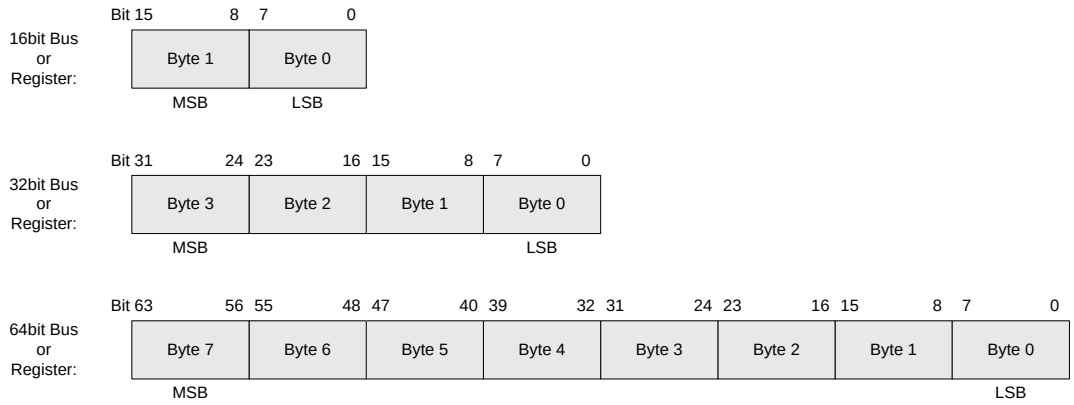


Figure 1. Bit and Byte Order

Signal names are written in document with `SignalName` style. Block names are written with Capital first letter. Pseudo code is written with `PseudoCode` style and command line commands are written with `CommandLine` style.

2 General

The FRS Reference Design consists of FPGA and SW design. The SW is run on NIOS2 softcore processor. The FPGA design is implemented using Altera QSYS tool that provides a graphical description for FPGA systems. FPGA design compilations are made using Altera Quartus II tool and the SW is compiled using Altera NIOS II EDS. Version information about the tools used, the IP blocks and the SW components are listed in the Reference Design release notes included in the release package.

3 FPGA Reference Design

The FPGA HW part of the design is described in this chapter.

3.1 Top Level

The Reference Design block diagram with external interfaces is described below. There are some differences between the Cyclone IV GX (C4GX), Cyclone V GX (C5GX) and Cyclone V GT (C5GT) designs, but mainly the designs are the same.

The main differences are:

1. C5GT reference design has fourth SFP module (SGMII/1000Base-X) instead of RGMII interface
2. C4GX reference design uses external 125 MHz clock source for transceiver reference clock. The C5GX/C5GT have fractional PLL to generate it from one common clock input of 50 MHz.

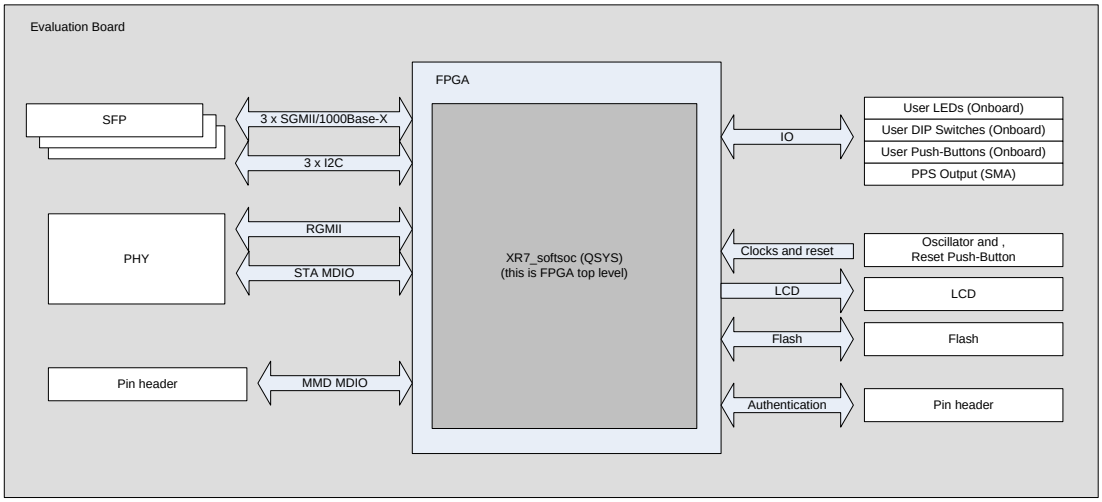


Figure 2. Reference Design Block Diagram

The Reference Design provides the following interfaces

- 3 x SGMII interface for fiber and copper SFP modules. (1000BASE-X and SGMII)
  - Redundant Port A
  - Redundant Port B
  - Interlink
  - In CVGT there is fourth SFP interlink
- RGMII interface for onboard PHY (only C4GX, CVGX)
- MMD MDIO/MDC for FRS management. Routed to external pin header (if available)
- STA MDIO(MDC for PHY management)
- I2C interfaces for SFP module management
- LCD interface
  - C4GX and C5GX have identical LCD command interfaces. For C5GT the interface is I2C.
- Authentication, routed to external pin header (if available)
  - Not available in C5GX. The design will function for 2 hours.
  - Note that the Authentication chip [11] is not on the evaluation board (C4GX, C5GT). Without external security chip the design will function for 2 hours
- Flash for FPGA configuration
- Push-buttons and DIP switches
- User and Link LEDs
- Clocks and reset
  - C4GX

- 50 MHz, clk
  - Main clock for QSYS components
  - QSYS design (xr7_softsoc) contains PLL's to generate all needed clock frequencies except :
- 125MHz, eth_clk
  - Used directly as transceiver reference clock

- C5GX
  - 50 MHz, clk
    - Main clock for QSYS components
    - QSYS design (xr7_softsoc) contains PLL's to generate all needed clock frequencies including transceiver reference clock
- C5GT
  - 50 MHz, clk
    - Main clock for QSYS components
    - QSYS design (xr7_softsoc) contains PLL's to generate all needed clock frequencies including transceiver reference clock

3.2 QSYS Design

The QSYS design, xr7_softsoc, is a combination of many QSYS components and their sub components. The block diagram below describes the QSYS system in the highest level and clock, reset and Avalon routing.

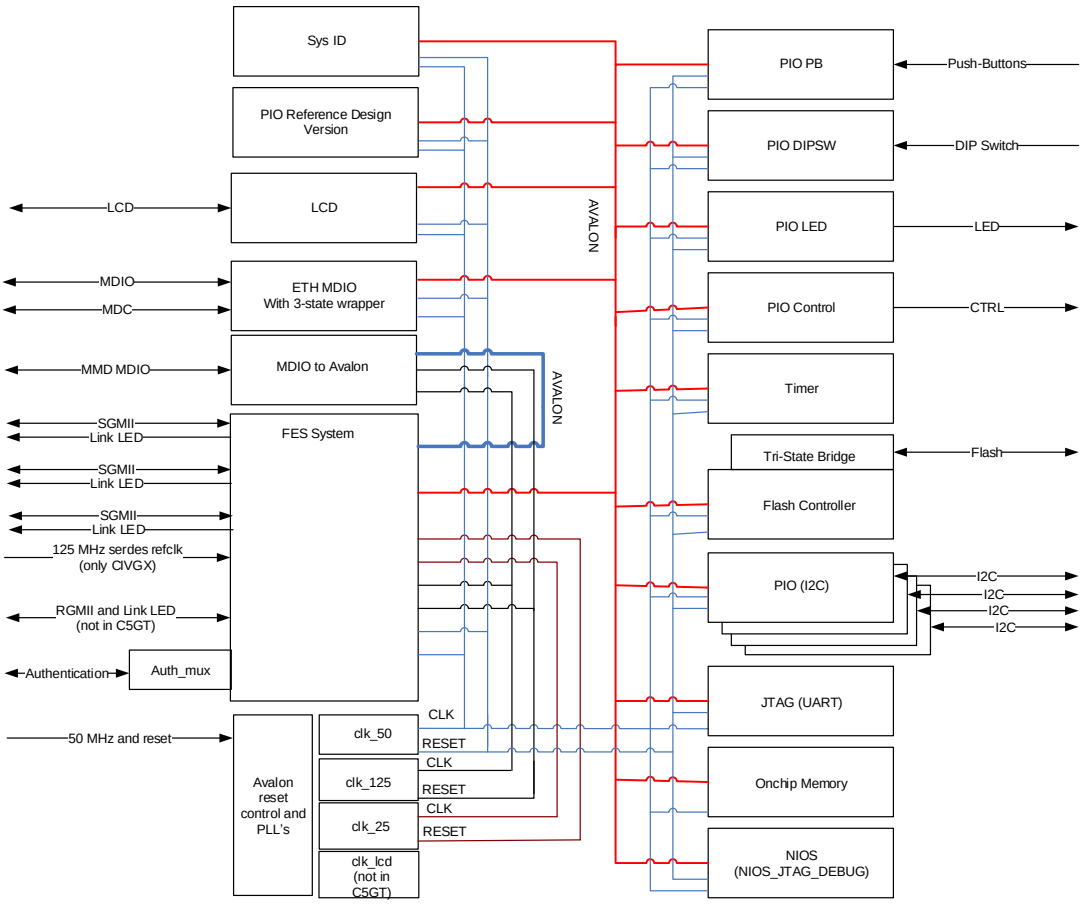


Figure 3. XR7_softsoc Design Block Diagram

XR7_softsoc includes the following components:

1. Avalon reset control and PLL's
    a. CPU_RESET push button de-bouncing function
    b. PLL's to generate needed clocks
    c. Clock bridge and clock source components are used in QSYS designs to import and distribute clock and resets into the QSYS system/components
2. FES System
    a. More detailed description in the next chapter
3. Authentication multiplexer
    a. Used to multiplex up to four FRS to use single SecIP. Used here even though there's only one FRS.
4. MDIO Slave
    a. Provide method for the MMD MDIO to access Avalon.
5. ETH MDIO
    a. Provides MDIO/MDC interface for NIOS. Used to configure external RGMII PHY.
    b. Use with 3-state wrapper makes it possible to export 3-state signals (without separate top level design file)
6. LCD
    a. Provides LCD interface
    b. In C5GT the block is I2C
7. Reference Design Version
    a. Includes Reference Design version number
8. SYS ID
    a. Includes System ID
9. NIOS II Soft processor core
    a. Includes also JTAG debug module
10. Onchip memory
    a. Onchip memory is used for SW and for AFEC TX and RX buffers
11. JTAG Uart
    a. Enables NIOS SW debugging
12. I2C
    a. Actually a Parallel IO block (PIO), used to generate I2C interface
13. Flash Controller
    a. Configuration Flash controller
14. Timer
15. PIO Control
    a. Control signals, currently only PHY Reset
16. PIO LED
    a. Control LEDs
17. PIO DIPSW
    a. User DIP switch signals
18. PIO PB
    a. User Push-Buttons

3.2.1 FES System

The FES System QSYS component is actually a subsystem i.e. it includes other QSYS components. It is generated based on configurations with tcl scripts, which can be found in the FES System component folder. The FES System component block diagram with current configuration is presented in Figure 4.

In previous Reference design versions, up to 2.9.4, similar block was called FRS Redbox. However, as the Flexibilis Ethernet Switch (FES) and Flexibilis Redundant Switch (FRS) were combined into same IP block, the FRS Redbox component was also replaced with a new QSYS component.

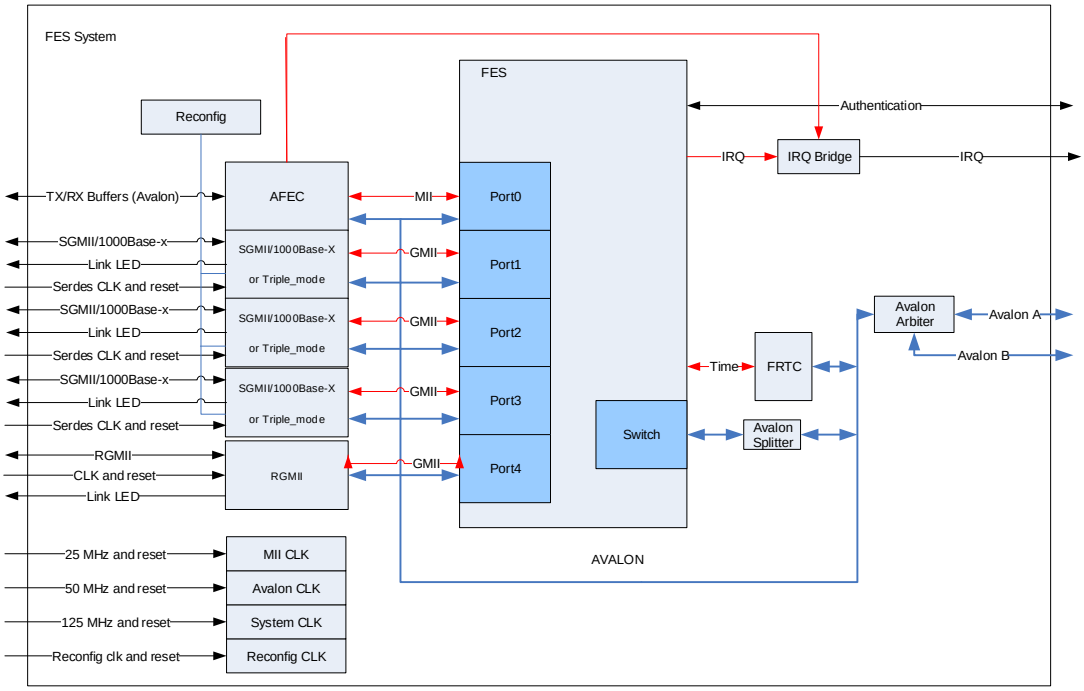


Figure 4. FES System Component Block Diagram

FES (FRS)

FES QSYS component is the Ethernet switch core which instantiated Flexibilis Ethernet Switch [7]. It can be instantiated also in QSYS as a separate component “Flexibilis Ethernet Switch” (FES_core) or in the VHDL code.

Flexibilis Ethernet Switch supports optional features which enable redundant communication (FRS variation). For more information see FES user manual [7].

AFEC

Advanced Flexibilis Ethernet Controller (AFEC) acts as an Ethernet Controller for the NIOS. More info on AFEC can be found in the AFEC user Manual [6].

SGMII/1000BASE-X

SGMII/1000BASE-X adapters, used in Cyclone IV reference design, provide interface conversion between GMII (FRS) and SGMII/1000BASE-X. This component includes Altera Transceiver IP, which is for Cyclone IV “ALTGXB” and for Cyclone V “Custom PHY”.

TRIPLE MODE

Triple Mode adapter, used in Cyclone V reference design, provides interface conversion between GMII (FRS) and SGMII/1000BASE-X/100BASE-FX.

RGMII

RGMII adapter provides interface conversion between GMII (FRS) and RGMII.

RECONFIG

Since the design implements Transceivers in the SGMII/1000BASE-X adapter, this block is required to be instantiated. Currently it does not include any functionality.

FRTC

Flexibilis Real Time Clock provides time information for FRS. FRTC can be controlled via Avalon.

IRQ Bridge

IRQ Bridge component provides a QSYS supported method for interrupt mapping.

Clock Sources

FES System component includes four clock source components (mii_clk, avalon_clk, system_clk, reconfig_clk), that are used to map clock signals into QSYS components.

Avalon Arbiter

The Avalon Arbiter is able to provide arbitration functionalities for two different Avalon masters. In this case the masters are

- MDIO Slave
- NIOS

Avalon Splitter

The Avalon Splitter component is generated automatically by the QSYS to support older FRS control mechanisms, even though it would not be required in this case.

3.3 FES System Configuration

This chapter describes how FES system component is configured and how configurations affect the actual design.

To ease the configuration, instantiation and mapping of VHDL designs, the QSYS component provides automatic component and signal mapping and instantiation based on the configuration made in QSYS GUI.

3.3.1 Generic Configuration

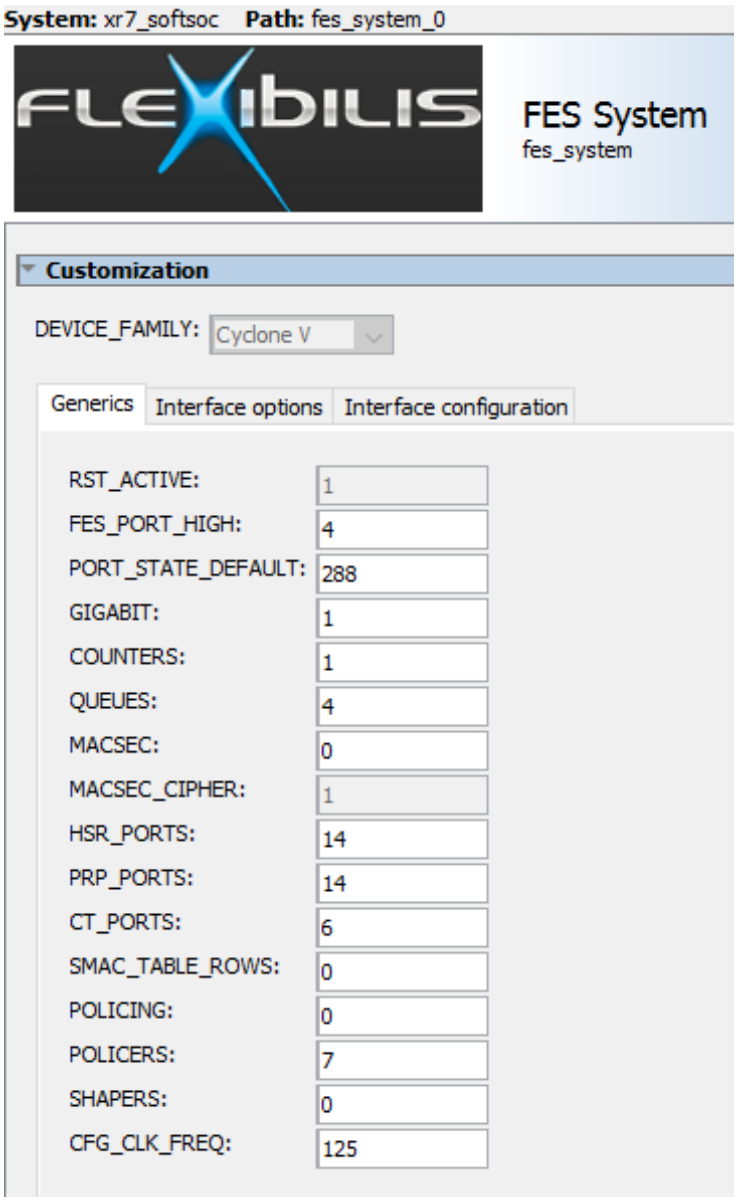


Figure 5. Generics

FES System Generics configurations page is shown in Figure 1, with settings used in this case. The configurable generics are:

- FES_PORT_HIGH
  - Sets the port count. Value 4 means that there are five ports.
- PORT_STATE_DEFAULT
  - Sets the default value for all PORT_STATE registers. Value 288 is 0x0120, which sets ports to forwarding mode, GMII and 1000Mbps
- GIGABIT
  - Enables Gigabit operation (optional feature of FES). For more information see FES user manual [7]
  - Gigabit operation is enabled in the reference design
- COUNTERS
  - If COUNTERS = ‘1’ then the COUNTERS block is synthesized (optional feature of FES). Use value ‘0’ when no COUNTERS are needed to save logic.

    - Counters are enabled in the reference design.
- QEUEUS
    - Number of priority queues (optional feature of FES). Possible values are 4 and 8. For more information see FES user manual [7].
    - Reference design uses 4 priority queues
- MACSEC
    - MACSEC port mask. Enables MACSEC functionality on corresponding port (optional feature of FES). For more information see FES user manual [7].
    - MACSEC is not supported in the reference design
- HSR_PORTS
    - Defines which ports support HSR. HSR support is optional features of FES.
    - Decimal value 14 means that ports 1,2 and 3 include HSR capabilities
- PRP_PORTS
    - Defines which ports support PRP. PRP support is optional features of FES.
    - Decimal value 14 means that ports 1,2 and 3 include PRP capabilities
- CT_PORTS
    - Enables Cut-through operation between two HSR ports. Cut-through is an optional feature of FES. For more information see FES user manual [7].
    - Value 6, means ports 1 and 2.
- SMAC_TABLE_ROWS
    - Defines how many rows are supported by the SMAC. SMAC is an optional feature of FES [7].
    - In this reference design the SMAC is not enabled.
- POLICING
    - Enables Policer functionalities (optional feature of FES). For more information see FES user manual [7].
        - Value 0: no policing functionalities (bandwidth limiting) supported
        - Value 1: Policing supported via IPO
        - Value 2: Policing supported via SMAC
        - Value 3: Policing supported via SMAC/IPO, selectable via registers configurations
    - In the reference design the policers are not supported
- POLICERS
    - Defines how many policers are supported per port. value 7 means 128 policers, value 8 means 256 policers etc.
    - Requires that POLICING generic has a value 1,2 or 3.
- SHAPERS
    - Enables Shaping functionalities (optional feature of FES). For more information see FES user manual [7].
    - Shaping is not supported by the reference design
- CFG_CLK_FREQ
    - This defines the system clock frequency for the FES and it must be set to match the actual clock frequency. In this case it is 125 MHz

3.3.2 Interface Options

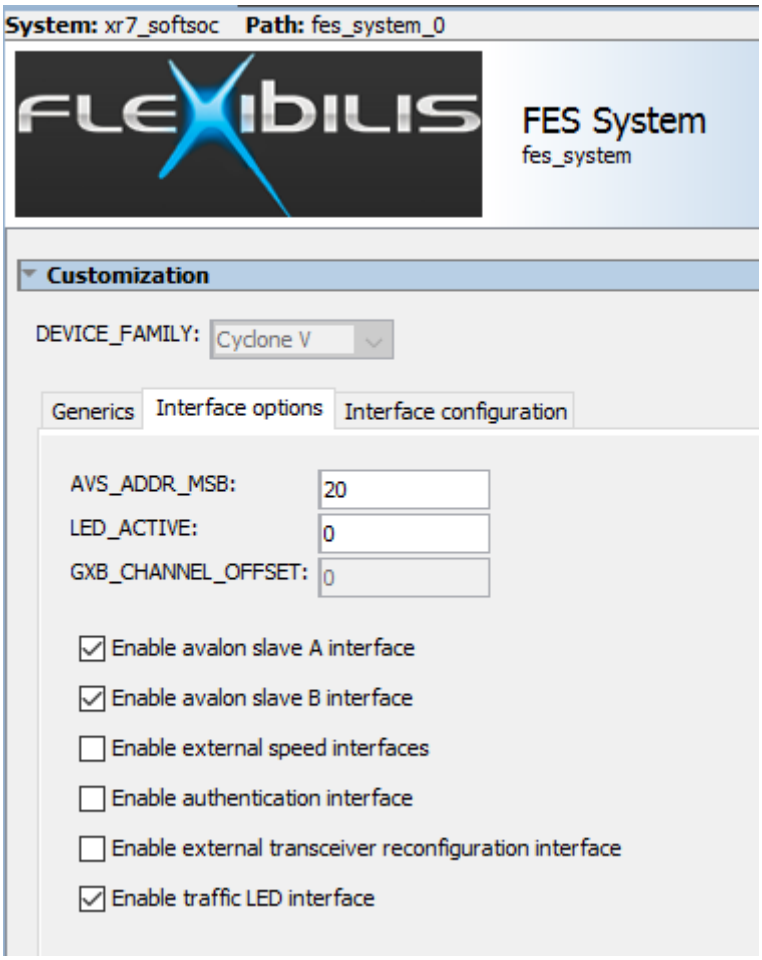


Figure 6. Interface Options

In the Interface Options page, Figure 6, it is possible select what interfaces FES system provides.

1. Avalon slave A interface
    a. Enabled in the reference design and connected to the NIOS in the QSYS
2. Avalon slave B interface
    a. Enabled in the reference design and connected to the MDIO Slave in the QSYS
3. External Speed interface
    a. Not enabled in the reference design. Could be used to set speed/interface mode for the FES ports.
4. Authentication Interface
    a. Not enabled in the reference design. Must be exported and used, if an external security chip is used for IP license validation
5. Transceiver reconfiguration interface
    a. Not enabled in the reference design. Could be used for providing reconfiguration interface from external controller. Usually used in designs with multiple FES cores using transceiver interfaces.
6. Traffic LED interface
    a. Enabled in the reference design. Provides signals for driving traffic LEDs (if provided by the interface adapters)

In addition this page defines the following interface related generics:

- AVR_ADDR_MSB

  - Sets the Avalon address bus width. As a MSB definition it is actual width minus one.
- LED_ACTIVE
  - Selects logic level when Link LEDs should lit.
- GXB_CHANNEL_OFFSET
  - Not applicable since only one FES is instantiated inside one FPGA chip.
  - In case there would be multiple FES inside one FPGA, the GXB_CHANNEL_OFFSET needs to be individual for each.

3.3.3 Interface Configuration

In the Interface configurations page there are three sub pages.

FRTC configurations are common for Interface configuration pages.

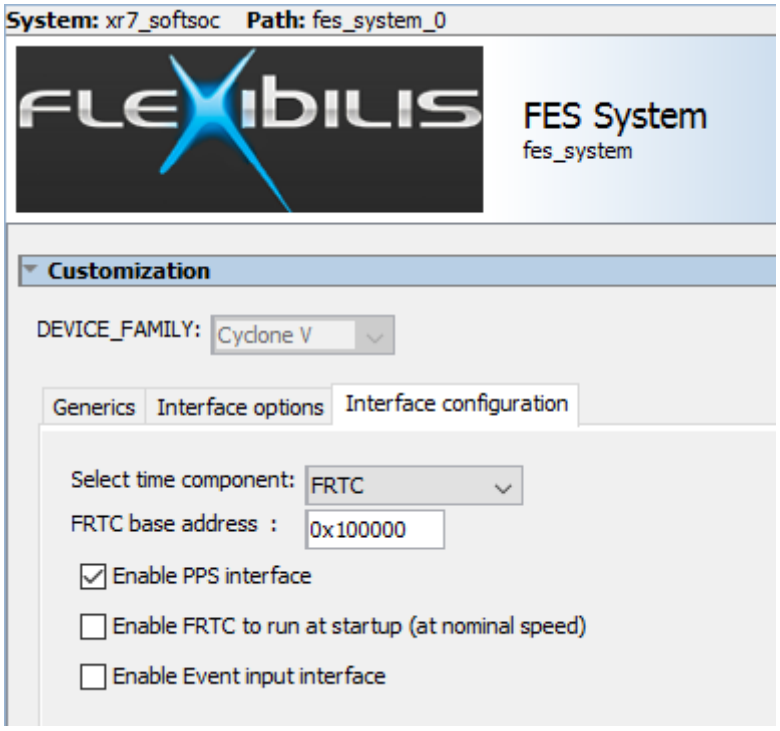


Figure 7. FRTC/FPTS Options

with these setting it is possible to enable the PPS signal output and start FRTC clock without register configurations. The Event input requires a separately licensable block called Flexibilis PPX Timestamper (FPTS).

3.3.3.1 Port Interface Type

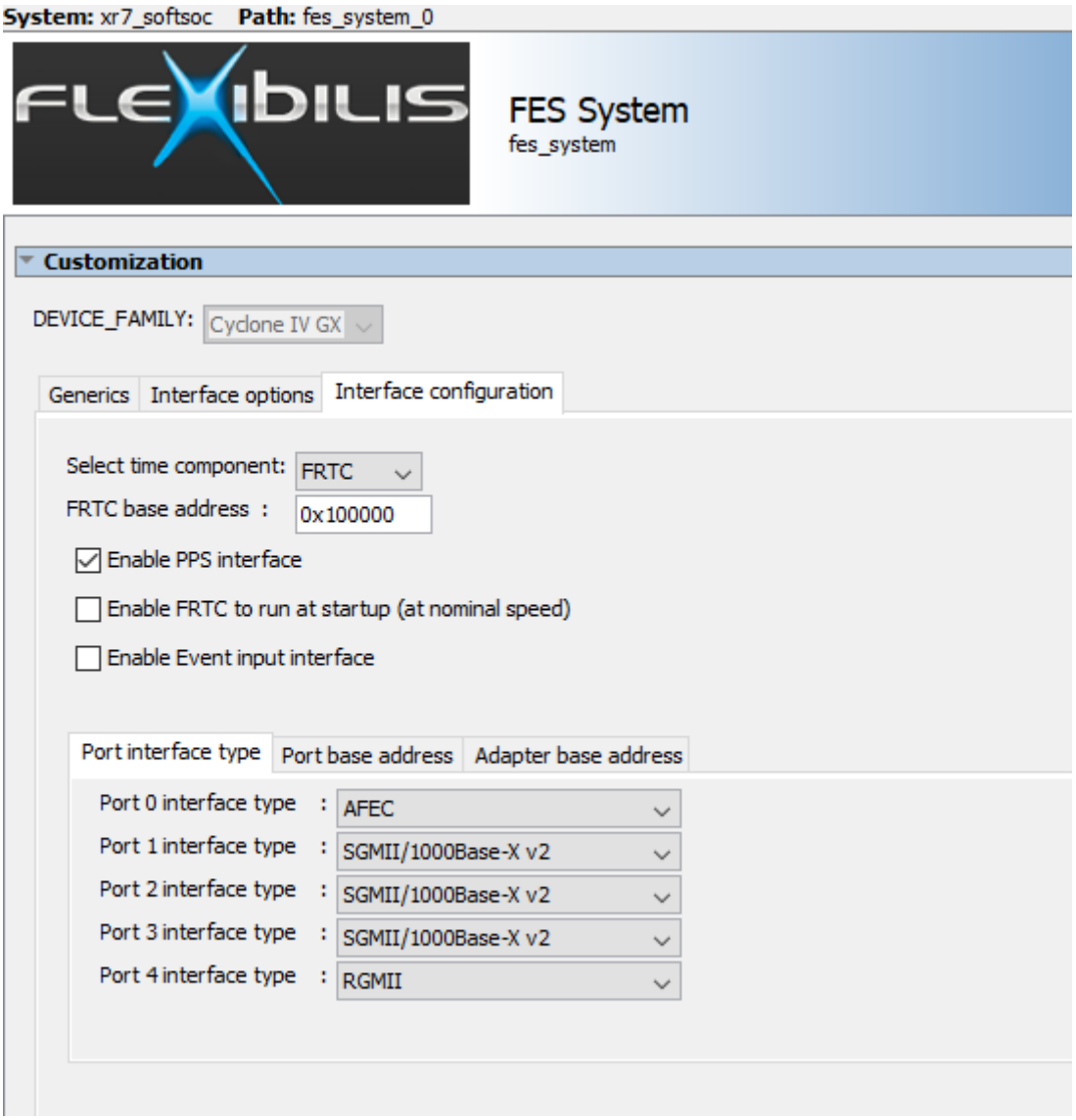


Figure 8. Interface Configurations for Cyclone IV

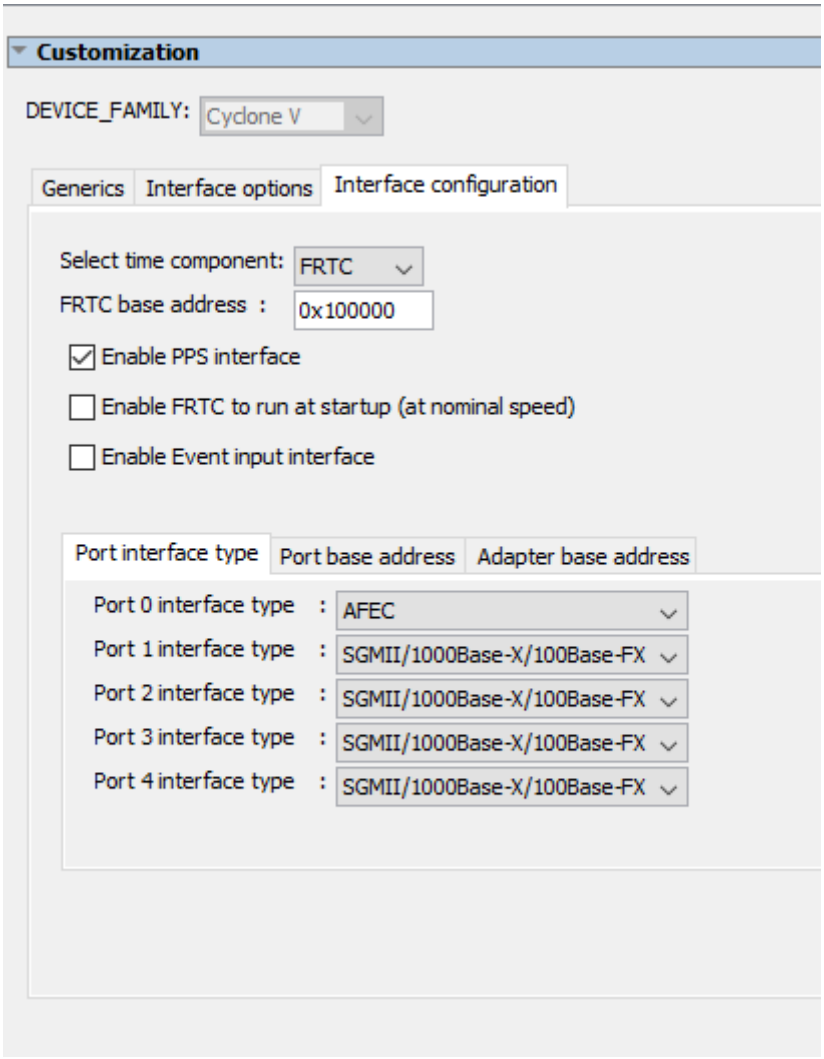


Figure 9. Interface Configurations for Cyclone V

The Port Interface type, Figure 8 and Figure 9, is used to select interface type for each port. In these case:

- Port 0 is AFEC i.e. it used by the NIOS
- Port 1, 2 and 3 are SGMII/1000BASE-X v2 [12] or Triple mode
- Port 4 is RGMII or Triple mode

Other possible interface types include:

- MII PHY mode
  - Provides interface, which is compatible with most PHY interfaces
- GMII/MII Native (None)
  - Provides FES GMII interface without modifications
  - Mainly for internal use i.e. QuadBox
- 1000BASE-X
- 100BASE-FX
- EMAC (for Altera's Ethernet Media Access Controller)
- RMII
- SGMII/1000base-X (using ALT_TSE)

EMAC is used in Cyclone V SoC designs.

3.3.3.2 Port Address Configuration

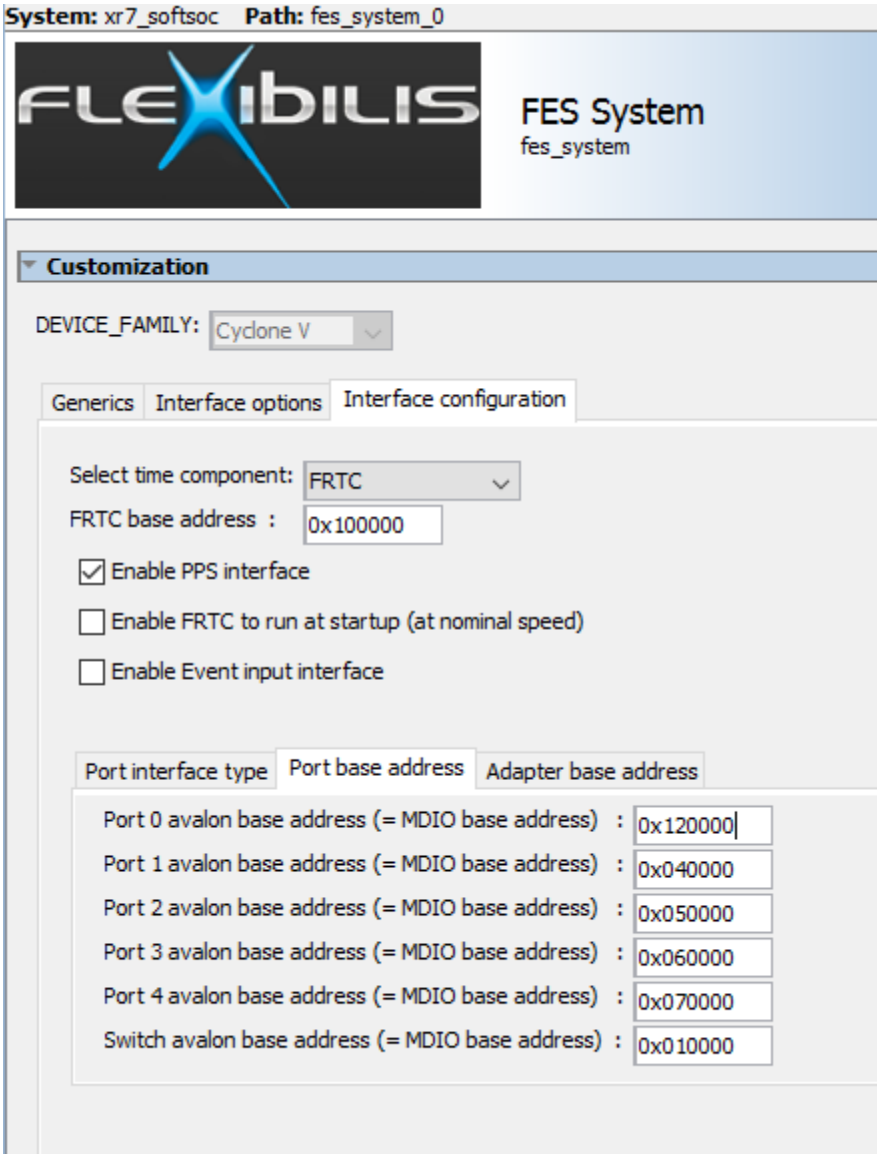


Figure 10. Port Address Configurations

Port base address configuration, Figure 10, defines the Avalon address offset from FES System base address for FRS Port and Switch configuration registers. Addresses are defined as word addresses. Avalon address map is defined in Chapter 3.6.

3.3.4 Adapter Address Configuration

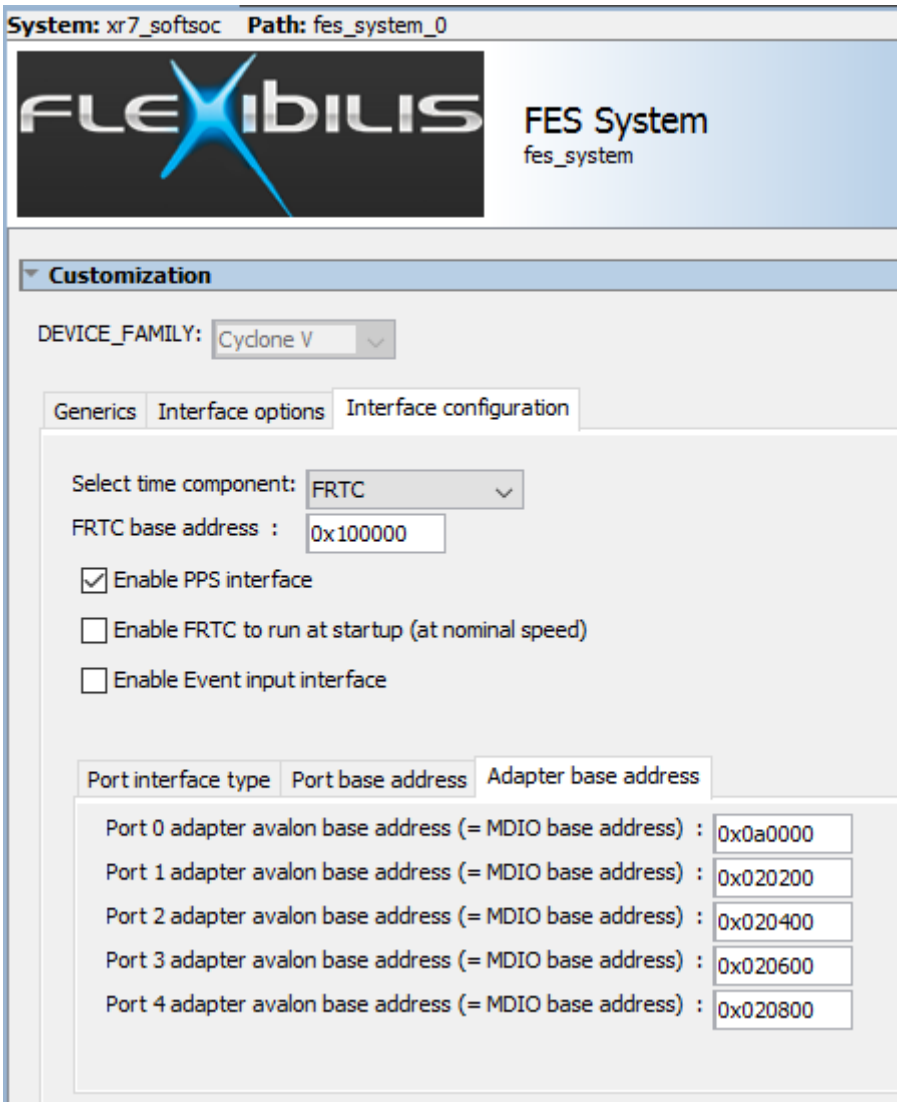


Figure 11. Adapter Address Configurations

Figure 11 presents adapter base address configuration, which defines the Avalon address offset from the FES System base address for adapter registers. Addresses are defined as word addresses. Avalon address map is defined in Chapter 3.6.

3.4 MDIO Slave Configuration

Also the MDIO slave QSYS component (Figure 3) provides certain configurations for the user.

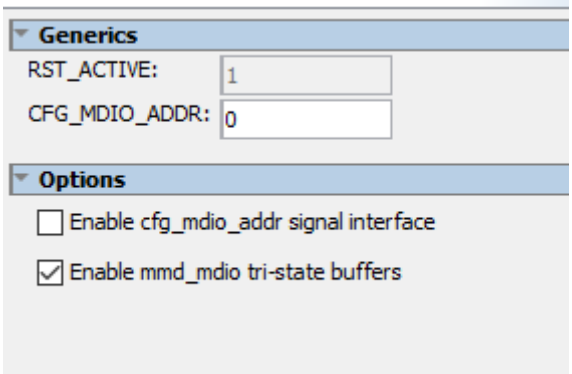


Figure 12. MDIO Slave Configurations

Figure 12 presents MDIO Slave configurations page with settings used in this case. Configurable generics are:

- CFG_MDIO_ADDR
  - Default value for MDIO address, if cfg_mdio_addr input is not used.
- Enable mmd_mdio tri-state buffers
  - Enable tri-state buffers
- Enable cfg_mdio_addr signal interface
  - Enabled the cfg_mdio_addr signals, which can be used to configure MDIO address for the MDIO Slave

3.5 Interface Adapters

Interface adapters are used with FRS to provide other Ethernet interface types than the FRS native MII/GMII. Figure 4 illustrates the Adapters used in this Reference Design. However, the FES System provides also other interface adapters as mentioned in chapter 3.3.3.1.

Some of the adapters include registers for configuration and status. For more information about the adapters refer to the SGMII/1000Base-X adapter specification [12] and Interface Adapters specification[13].

3.6 Avalon Address Map

The Avalon Address Map is based on the settings in QSYS and its components. The Table 1 below defines the Address map for this Reference Design. Addresses are defined as word addresses.

Component	BUS and Base Address		
	Data	Instruction	AFEC TX/RX Buffer
JTAG, debug on NIOS	0x0404_0800	0x0404_0800	
JTAG, Uart	0x0400_0010		
Onchip Memory	0x0500_0000	0x0500_0000	0x0500_0000
I2C_interface_0	0x0400_0100		
I2C_interface_1	0x0400_0140		
I2C_interface_2	0x0400_0180		
I2C_interface_3	0x0400_01c0		
Flash	0x0000_0000	0x0000_0000	
Sys ID	0x0400_0000		
LCD (Clock Crossing)	0x0700_0000		

Component	BUS and Base Address		
Timer	0x0400_0080		
PIO LED	0x0400_0040		
PIO DIPSW	0x0400_0420		
PIO PB	0x0400_0400		
PIO REF. Design version	0x0400_00c0		
PIO CTRL	0x0400_0440		
ETH MDIO tristate	0x0400_0300		
FES System	0x0600_0000		
Switch			
Switch registers	0x0601_0000		
TS	0x0601_1000		
VLAN	0x0601_2000		
FRS Port 0			
GEN	0x0612_0000		
HSR	0x0612_1000		
PTP	0x0612_2000		
CNT	0x0612_3000		
IPO	0x0612_4000		
FRS Port 1			
GEN	0x0604_0000		
HSR	0x0604_1000		
PTP	0x0604_2000		
CNT	0x0604_3000		
IPO	0x0604_4000		
FRS Port 2			
GEN	0x0605_0000		
HSR	0x0605_1000		
PTP	0x0605_2000		
CNT	0x0605_3000		
IPO	0x0605_4000		
FRS Port 3			
GEN	0x0606_0000		
HSR	0x0606_1000		
PTP	0x0606_2000		
CNT	0x0606_3000		
IPO	0x0606_4000		
FRS Port 4			
GEN	0x0607_0000		
HSR	0x0607_1000		
PTP	0x0607_2000		

Component	BUS and Base Address		
CNT	0x0607_3000		
IPO	0x0607_4000		
Adapter P0			
AFEC	0x060A_0000		
Adapter P1			
SGMII/1000BASE-X	0x0602_0200		
Adapter P2			
SGMII/1000BASE-X	0x0602_0400		
Adapter P3			
SGMII/1000BASE-X	0x0602_0600		
Adapter P4			
RGMII	0x0602_0800		
FRTC	0x0610_0000		

Table 1. Avalon Address Map

3.7 Compilation

This chapter describes shortly the Reference Design compilation steps. For more detailed information about Quartus, please refer to the documentation available in Altera's website.

The Reference Design package includes ready-to-use program files, so compilation is not a mandatory step.

3.7.1 Folder Structure

The Reference Design folder structure is illustrated in Figure 13. The *fpga* folder includes *bin_qsys* folder, which includes most of the quartus project files. In addition there is an *ip* folder, which includes QSYS component files. The FPGA project expects that *external* and *encrypted_ip* folders are in place and have the correct content. Therefore the user should place separate FRS, FRTC and AFEC IP cores into the correct folders or change the file path settings in Quartus project accordingly.

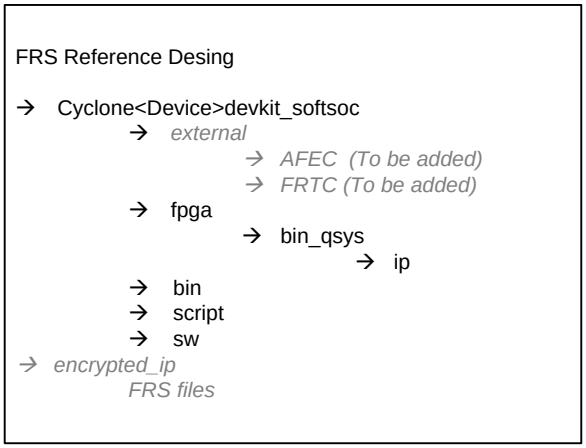


Figure 13. Folder Structure

3.7.2 QSYS Generation

The Reference Design is fully described by xr7_softsoc.qsys file, because it is directly the top level of the design. The QSYS system needs to be generated before the first synthesis/fitting is done, since the QSYS result design files are not included in the package. Also, the QSYS system needs a regeneration if something in the QSYS system is changed or the source codes of the QSYS components are modified.

QSYS is opened from Quartus. First, open the cyclone<Device>devkit.qpf (Quartus Project File) with quartus. Then on the toolbar open the QSYS  . Once QSYS opens, it will query the qsys system file to be opened. This is the xr7_softsoc.qsys file that is located in the bin_qsys folder. Once correct system has opened the user can either make changes or then directly generate a new qsys design. Generation is activated from the Generation page, pressing the Generate button. There should not be any errors or warnings during the generation. Once generation has finished, all necessary files have been generated and user can return to the Quartus project.

3.7.3 Quartus Project

The quartus project is configured with the correct assignments (pinout, IO Voltage, clock constrains etc.) and file references. The user should also make sure that the links to the files are correct, especially for the external IP cores.

To be able to compile the design, the user should make sure that the required licenses are available for the Quartus tool. The license setup can be found in the Tools menu. This Reference Design requires:

- FRS (Flexibilis)
- AFEC (Flexibilis)
    - OCP license provided
- FRTC (Flexibilis)
    - OCP license provided
- NIOS (Altera)
- SGMII/1000Base-X (Flexibilis)
- RGMII (Flexibilis)
- MDIO Slave (Flexibilis)

All of these licenses are available for evaluation.

Once the QSYS system is generated and the licenses are set, the design can be compiled in normal way. The design will generate warnings, and they should be looked thought. However, they all should be generated by Altera blocks and should not affect the operations. In case of suspicious warnings contact Flexibilis.

4 SW Reference Design

SW Reference Design is implemented as a modular design. The modules are depicted in Figure 14.

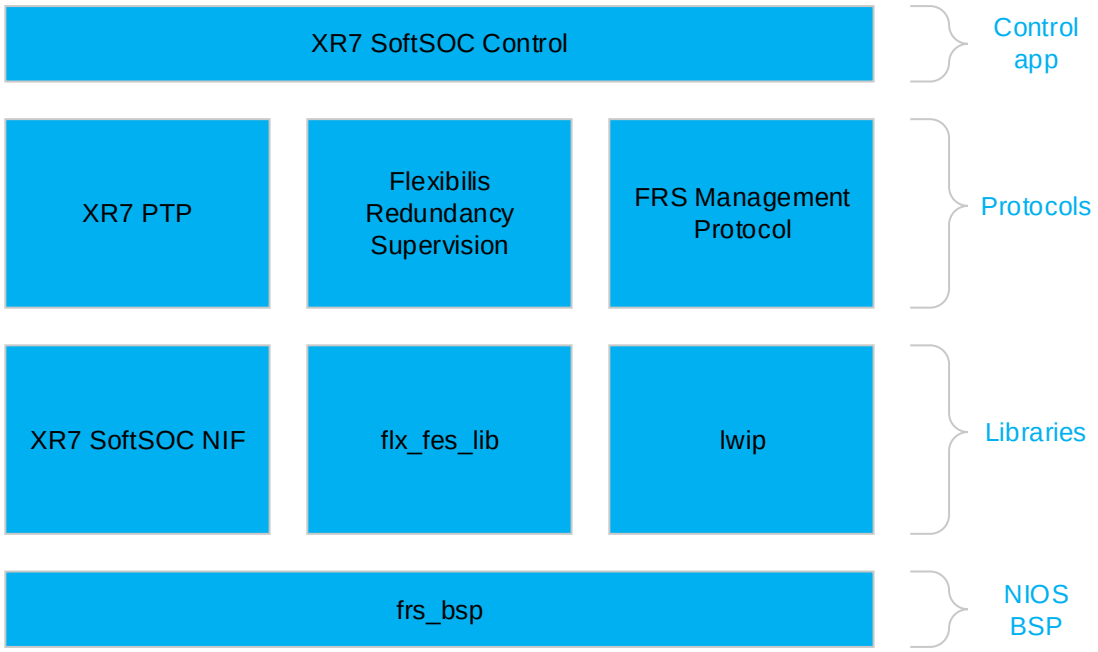


Figure 14. SW Reference Design

The modules depicted in Figure 14 are located in separate directories under sw directory in reference design directory.

Directory	Description
xr7_softsoc_control	XR7 SoftSoC Control - control application. Source code is located in Reference Design release in FESHA00E00-FBIT-V<ver>\cyclone4GXdevkit_softsoc\sw\xr7_softsoc_control\src directory.
xr7ptp	XR7 PTP – PTP protocol stack library [2], [3]. Source code is delivered separately and requires a license from Flexibilis Oy.
flx_redundancy_supervision	HSR/PRP Supervision protocol [1], [4]. Source code is delivered separately and requires a license from Flexibilis Oy.
frs_management_protocol	FRS Management Protocol [5]. Source code is delivered separately and requires a license from Flexibilis Oy.
time_daemon	Interface libraries for XR7PTP
xr7_softsoc_nif	Helper library, contains for example drivers for IP blocks and controlling of SFP modules and external PHY. Source code is located in Reference Design release in FESHA00E00-FBIT-V<ver>\cyclone4GXdevkit_softsoc\sw\xr7_softsoc_nif directory.
flx_fes_lib	Helper library, for accessing FRS configuration. Source code is located in Reference Design release in FESHA00E00-FBIT-V<ver>\cyclone4GXdevkit_softsoc\sw\flx_fes_lib directory.
lwip	lwIP - A Lightweight TCP/IP stack, licensed under BSD license (see Chapter 7). Source code is located in Reference Design release in FESHA00E00-FBIT-V<ver>\cyclone4GXdevkit_softsoc\sw\lwip directory.
frs_bsp	BSP for NIOS, generated by Altera NIOS II EDS according to delivered configuration. Configuration is available in FESHA00E00-FBIT-V<ver>\<board>\sw\frs_bsp directory.

Table 2. SW Reference Design Modules

Source codes for protocol stacks (XR7 PTP, Redundancy supervision, FRS management protocol) are delivered separately and they require a license with Flexibilis Oy. All protocol

stacks are included in the Reference Design release in a binary format and the functionality can be verified using the evaluation boards.

4.1 Modules

4.1.1 XR7 SoftSoC Control

XR7 SoftSoC Control is the main module of the system. It handles the initialization of all the other modules and threads and performs control tasks during the execution. The source code files are listed in Table 3.

File	Description
src/xr7_softsoc_ctrl.c	Main initialization and control functions. For example, initializes all threads, scans ports, DIP-switches, updates configurations and prints to LCD.
src/xr7_softsoc_ptp.h src/xr7_softsoc_ptp.c	XR7 PTP thread handling: configuration and control.
src/xr7_softsoc_supervision.h src/xr7_softsoc_supervision.c	Flexibilis Redundancy Supervision module thread handling: configuration and control.

Table 3. XR7 SoftSoC Control Module Files

4.1.2 XR7PTP

XR7PTP implements Precision Time Protocol [2]. XR7PTP is described in detail in XR7PTP User Manual [3].

4.1.3 Flexibilis Redundancy Supervision

Flexibilis Redundancy Supervision implements HSR/PRP Supervision protocol [1]. Flexibilis Redundancy Supervision is described in detail in Flexibilis Redundancy Supervision Design Specification [4].

4.1.4 FRS Management Protocol

FRS Management Protocol implements a proprietary management protocol for managing FRS Reference Design. FRS Management Protocol is described in detail in FRS Management Protocol specification [5].

4.1.5 Time Daemon

Interface libraries for XR7PTP are included in time daemon directory. These packet and clock libraries integrate XR7PTP to this NIOS reference design. More information about the role of these libraries can be found from XR7PTP User Manual [3].

4.1.6 XR7 SoftSoC NIF

XR7 SoffSoC NIF module contains drivers and other helper functions. The source code files are listed in Table 4.

File	Description
Inc/card_if.h	Definitions for Reference Design. For example GPIO mapping, config options, etc.
inc/88e1000_ctrl.h src/88e1000_ctrl.c	Marwell 88e1000 PHY driver
inc/afec.h src/afec.c inc/flx_afec.h src/flx_afec.c	AFEC IP (Ethernet MAC) driver [6]
inc/ethernet_input_thread.h src/ethernet_input_thread.c	Optional support for two level IRQ handling in Ethernet MAC. Enabled in card_if.h with ENABLE_ETHERNET_INPUT_THREAD define.
inc/flx_frs_if.h src/fes_ctrl.c	Device driver for FRS IP. [7]
inc/frtc.h src/frtc.c	Device driver for FRTC IP. [8]
inc/i2c_ctrl.h src/i2c_ctrl.c	I2C bit bang device driver.
inc/nif_ctrl.h src/nif_ctrl.c	Network stack (lwip) initialization and helper functions.
inc/adapter_ctrl.h src/adapter_ctrl.c	Adapter management
inc/interface_ctrl.h src/ interface _ctrl.c	Configuration and control for different interface types
inc/phy_ctrl.h src/phy _ctrl.c	Configuration and control for PHYs

Table 4. XR7 SoftSoC NIF Module Files

4.1.7 flx_fes_lib

The library contains helper functions for managing FRS. The source code files are listed in Table 5.

File	Description
flx_fes.h flx_fes.c	Helper functions for configuring FRS IP [7]. Includes for example reading and writing of FRS registers and IPO settings.

Table 5. flx_fes_lib Module Files

4.1.8 lwip

The library contains lwIP - A Lightweight TCP/IP stack. It is an open source TCP/IP stack. More information can be found in lwip documentation [9].

4.1.9 frs_bsp

Board Support Package (BSP) generated by Altera NIOS II EDS tool.

4.2 Compilation

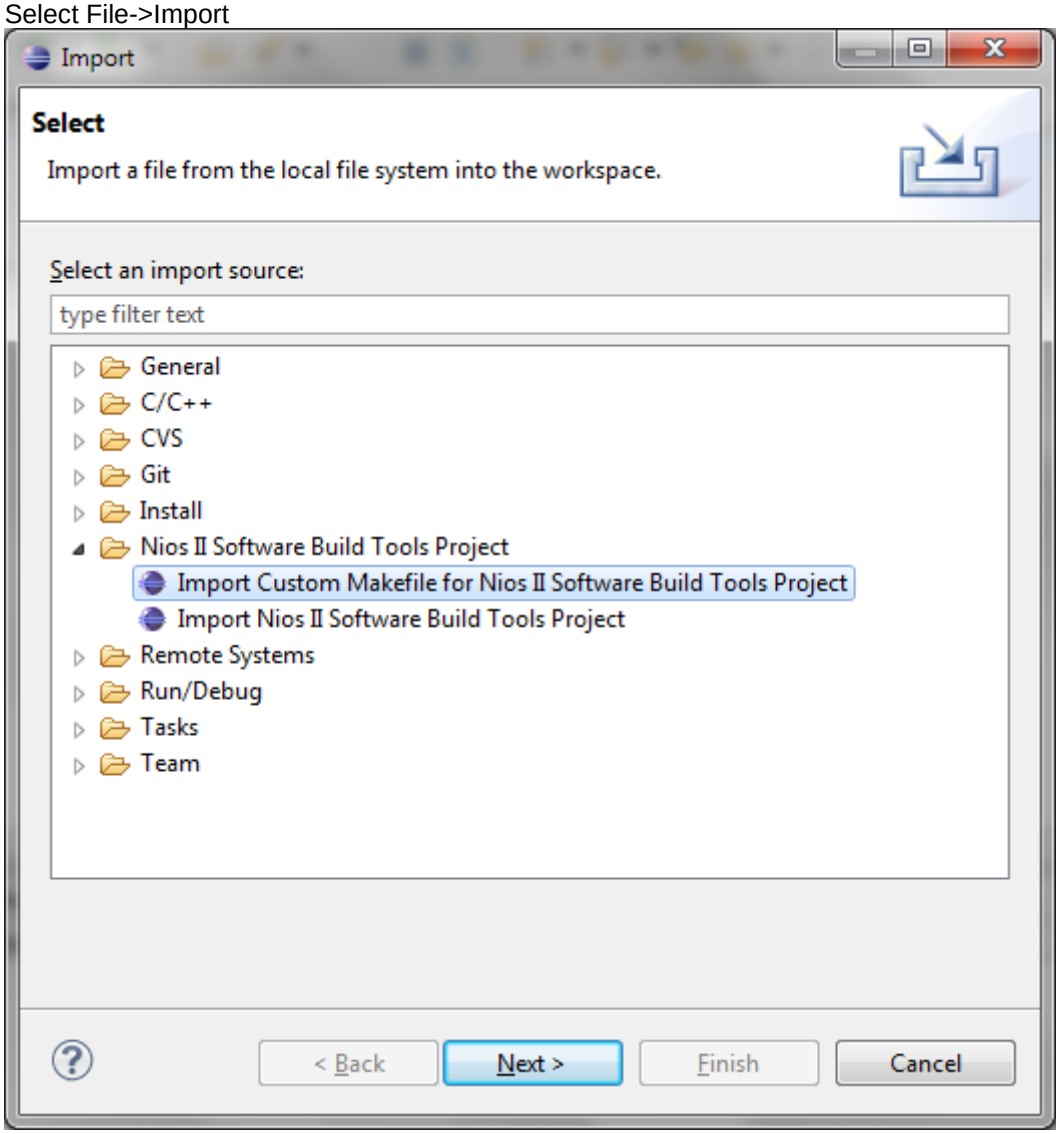
FRS Reference Design SW compilation is done in Altera NIOS II EDS. All the module sources listed in Table 2 need to be placed under FESHA00E00-FBIT-V<ver>\<board>\sw\ directory. SW Reference Designs for different boards use the same module sources. Only frs_bsp, which is generated by the Altera NIOS II EDS tool is board dependent.

4.2.1 Import Projects

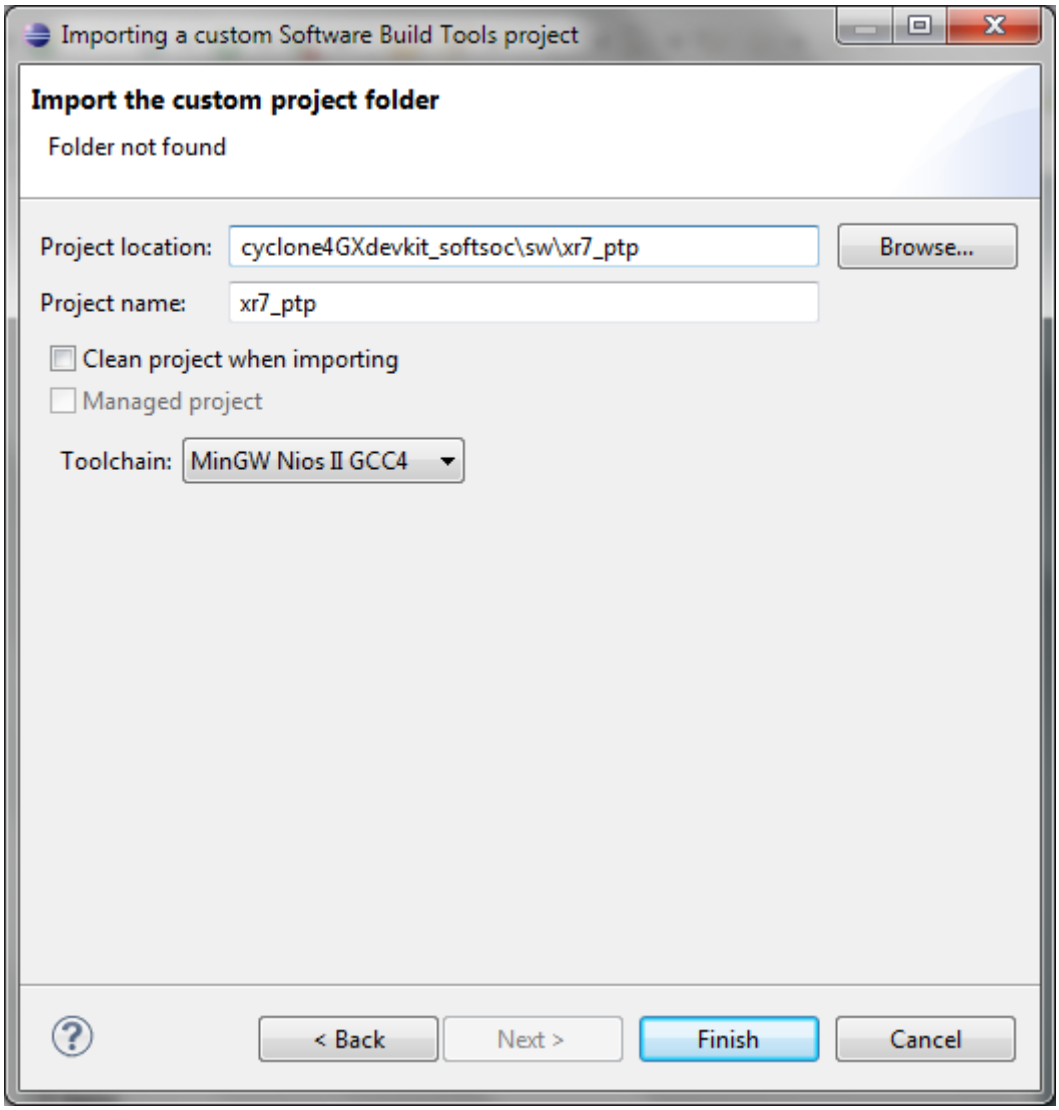
Every SW module is implemented as a separate project in NIOS II EDS. They have to be imported separately and after all the projects have been imported, the application can be compiled.

Importing steps are the following:

1. Select File->Import



2. From menu, select “Nios II Software Build Tools Project” and under that selection, select the correct import mode (discussed later).

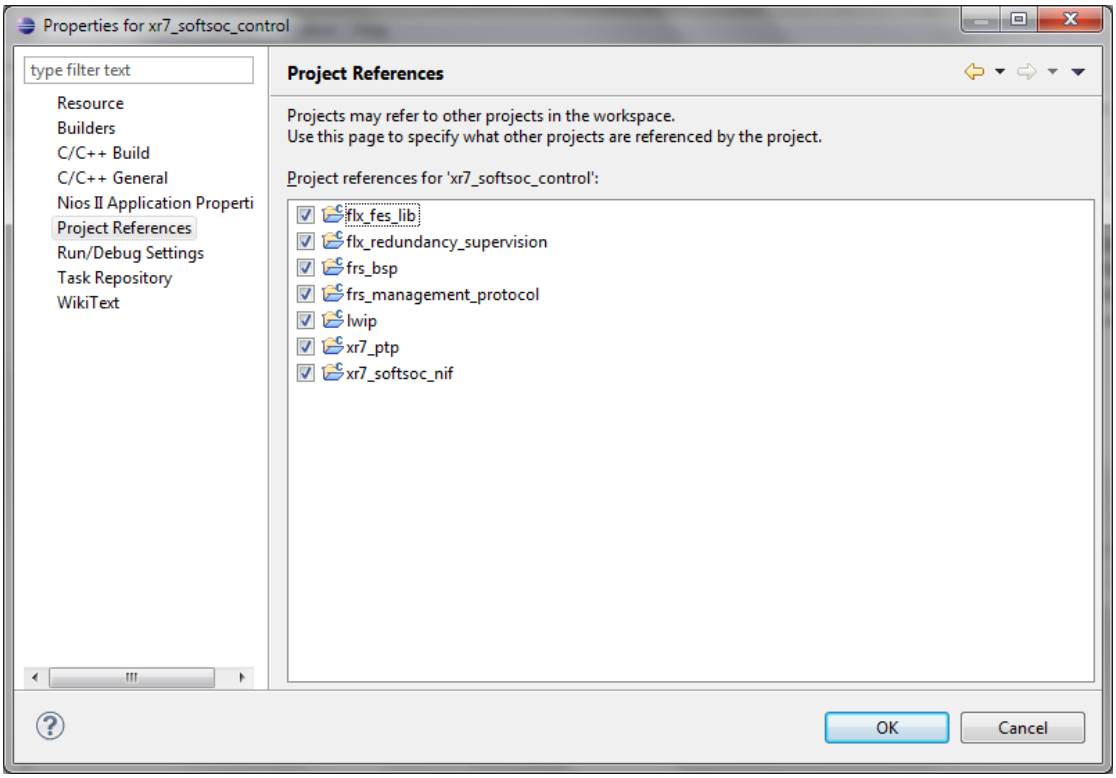


3. Click Browse and select SW module directory, for example xr7_ptp. Set “Project name:” the module name, in this example xr7_ptp, and click Finish.
4. Perform the steps 1-3 for every SW module.

In step 2, correct import type must be selected. All the protocol libraries and the lwip-library must be imported as “Import Custom Makefile for Nios II Software Build Tools Project”. All the other modules shall be imported as “Import Nios II Software Build Tools Project”.

4.2.2 Build Application

The first step in building the application is to include all other modules as referenced projects to xr7_softsoc_control. This can be done by first selecting xr7_softsoc_control project from Project Explorer in the left and then selecting File->Properties.



Then check all the SW modules referenced in the properties window.

The next step is to generate BSP. This can be done by right clicking the frs_bsp project and selecting Nios II -> Generate BSP.

After this, all that is needed is to build xr7_softsoc_control project by selecting the project and then selecting Project -> Build All. Now new xr7_softsoc_control.elf is generated.

4.2.3 Create Flash Files

New flash files can be generated with Nios II Command Shell in FESHA00E00-FBIT-V<ver>\<board>\script directory.

1. Open Nios II Command shell from Windows Start menu
2. Go to FESHA00E00-FBIT-V<ver>\<board>\script directory
3. Execute ./create_flash_bins.sh

Select correct options for the script according to the help. The SW option will generate only SW update flash file and the other options will generate also FPGA configuration flash file.

4.2.4 Program Flash

New flash files can be programmed to Reference Design board with Nios II Command Shell in FESHA00E00-FBIT-V<ver>\<board>\script directory.

1. Open Nios II Command shell from Windows Start menu
2. Go to FESHA00E00-FBIT-V<ver>\<board>\script directory
3. Execute ./flash_dev_board.sh

Select the correct options for the script according to the help. The SW option will program only the SW and the other options will program also the FPGA configuration to the flash. After programming the board, the board must be reset.

5 Abbreviations

Term	Description
AFEC	Advanced Flexibilis Ethernet Controller
BSP	Board Support Package
FES	Flexibilis Ethernet Switch
FPGA	Field Programmable Gate Array
FRS	Flexibilis Redundant Switch, Variation of FES
FRTC	Flexibilis Real Time Clock
HSR	High-availability Seamless Redundancy
PLL	Phase Locked Loop
PPS	Pulse Per Second
PTP	Precision Time Protocol
SFP	Small Form-factor Pluggable transceiver
SoC	System-on-Chip

6 References

[1] Standard IEC 62439-3:2011

[2] IEEE standard 1588-2008

[3] XR7PTP User Manual, xr7ptp_user_manual.pdf

[4] Flexibilis Redundancy Supervision design specification, flx_redundancy_supervision_design.pdf

[5] FRS Management Protocol, FRS_management_protocol.pdf

[6] Advanced Flexibilis Ethernet Controller (AFEC) User Manual, AFEC_user_manual.pdf

[7] Flexibilis Ethernet Switch (FES) Manual, FES_Manual.pdf

[8] FRTC User Manual, FRTC_user_manual.pdf

[9] lwIP - A Lightweight TCP/IP stack, open source, http://savannah.nongnu.org/projects/lwip/

[10] Triple-Speed Ethernet MegaCore Function User Guide, version 13.1, December 2013, http://www.altera.com/literature/ug/ug_ethernet.pdf

[11] IP License Authentication, Security Chip, Manual, version 1.2. http://www.flexibilis.com/downloads/Security_Chip.pdf

[12] SGMII/1000Base-X adapter Specification, FLXD816,version 1.0

[13] Interface Adapters specification, FLXD817, Version 1.0

7 BSD License

BSD License of the lwIP stack:

```
/*
 * Copyright (c) 2001, 2002 Swedish Institute of Computer Science.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without modification,
 * are permitted provided that the following conditions are met:
 *
 * 1. Redistributions of source code must retain the above copyright notice,
 *    this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright notice,
 *    this list of conditions and the following disclaimer in the documentation
 *    and/or other materials provided with the distribution.
 * 3. The name of the author may not be used to endorse or promote products
 *    derived from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED
 * WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
 * MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
 * SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
 * EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT
 * OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
 * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING
 * IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
 * OF SUCH DAMAGE.
 *
 * This file is part of the lwIP TCP/IP stack.
 *
 * Author: Adam Dunkels <adam@sics.se>
 *
 */
```