

# FRS Reference Design Specification

This document could contain technical inaccuracies or typographical errors. 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](mailto:support@flexibilis.com).

© Copyright Flexibilis Oy 2014. All rights reserved.

Trademarks

All trademarks are the property of their respective owners.

## Contents

---

<b>1 About This Document .....</b>	<b>6</b>
1.1 Conventions Used in This Document .....	6
<b>2 General .....</b>	<b>7</b>
<b>3 FPGA Reference Design .....</b>	<b>8</b>
3.1 Top Level .....	8
3.2 QSYS Design .....	9
3.2.1 FRS Redbox .....	11
3.3 FRS Redbox Configuration .....	13
3.3.1 Generic Configuration .....	14
3.3.2 Interface Options .....	15
3.3.3 Interface Configuration .....	15
3.3.3.1 Port Interface Type .....	15
3.3.3.2 Port Address Configuration .....	17
3.3.4 Adapter Address Configuration .....	18
3.4 MDIO-to-Avalon Bridge Configuration .....	18
3.5 Interface Adapters .....	20
3.5.1 SGMII/1000Base-X .....	20
3.5.2 1000BASE-X .....	20
3.5.3 100BASE-FX .....	20
3.5.4 GMII .....	21
3.5.5 RGMII .....	21
3.6 Avalon Address Map .....	22
3.7 Compilation .....	24
3.7.1 Folder Structure .....	24
3.7.2 QSYS generation .....	24
3.7.3 Quartus Project .....	24
<b>4 SW Reference Design .....</b>	<b>26</b>
4.1 Modules .....	27
4.1.1 XR7 SoftSOC Control .....	27
4.1.2 XR7 PTP .....	27
4.1.3 Flexibilis Redundancy Supervision .....	27
4.1.4 FRS Management Protocol .....	27
4.1.5 XR7 SoftSOC NIF .....	27
4.1.6 flx_fes_lib .....	28
4.1.7 lwip .....	28
4.1.8 frs_ucosii_bsp .....	28
4.2 Compilation .....	28
4.2.1 Import Projects .....	28
4.2.2 Build Application .....	30
4.2.3 Create Flash Files .....	31
4.2.4 Program Flash .....	31
<b>5 Abbreviations .....</b>	<b>33</b>
<b>6 References .....</b>	<b>34</b>

## Figures

---

Figure 1. Bit and Byte Order .....	6
Figure 2. Reference Design Block Diagram .....	8
Figure 3. XR7_softsoc Design Block Diagram .....	10
Figure 4. FRS Redbox Component Block Diagram .....	12

Figure 5. Generics .....	14
Figure 6. Interface Options .....	15
Figure 7. Interface Configurations .....	16
Figure 8. Port Address Configurations .....	17
Figure 9. Adapter Address Configurations .....	18
Figure 10. MDIO-to-Avalon Bridge Configurations .....	19
Figure 11. Folder Structure .....	24
Figure 12. SW Reference Design .....	26

## Tables

---

Table 1. ALT_TSE Adapter Configuration Registers .....	20
Table 2. 1000BASE-X Adapter Configuration Registers .....	20
Table 3. 100BASE-FX Adapter Configuration Registers .....	21
Table 4. GMII Adapter Configuration Registers .....	21
Table 5. RGMII Adapter Configuration Registers .....	22
Table 6. Avalon Address Map .....	23
Table 7. SW Reference Design Modules .....	26
Table 8. XR7 SoftSOC Control Module Files .....	27
Table 9. XR7 SoftSOC NIF Module Files .....	28
Table 10. flx_fes_lib Module Files .....	28

## 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 About This Document

This document describes the reference design for Flexibilis Redundant Switch (FRS) [8]. FRS is an Ethernet switch Intellectual Property (IP) core targeted at programmable hardware platforms. 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](http://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.

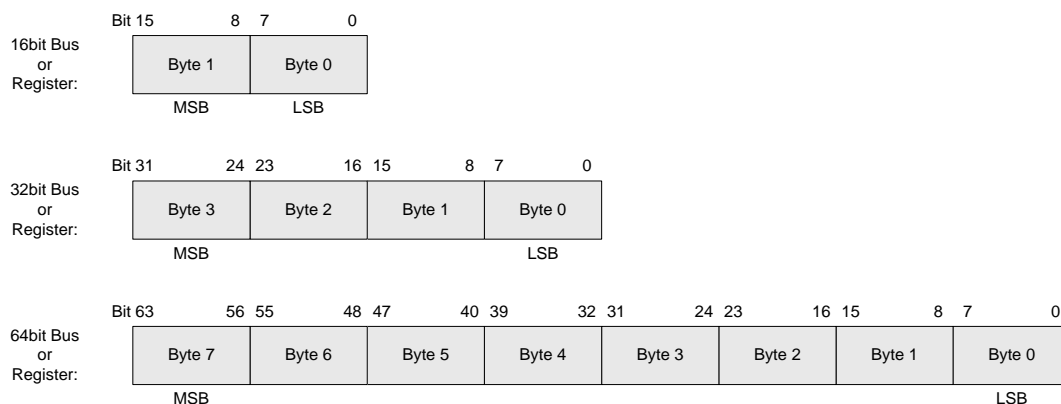
R/C = Read and Clear capable. The bits can be read and cleared. Writing 0 to the bit makes its value 0. Writing 1 does nothing.

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

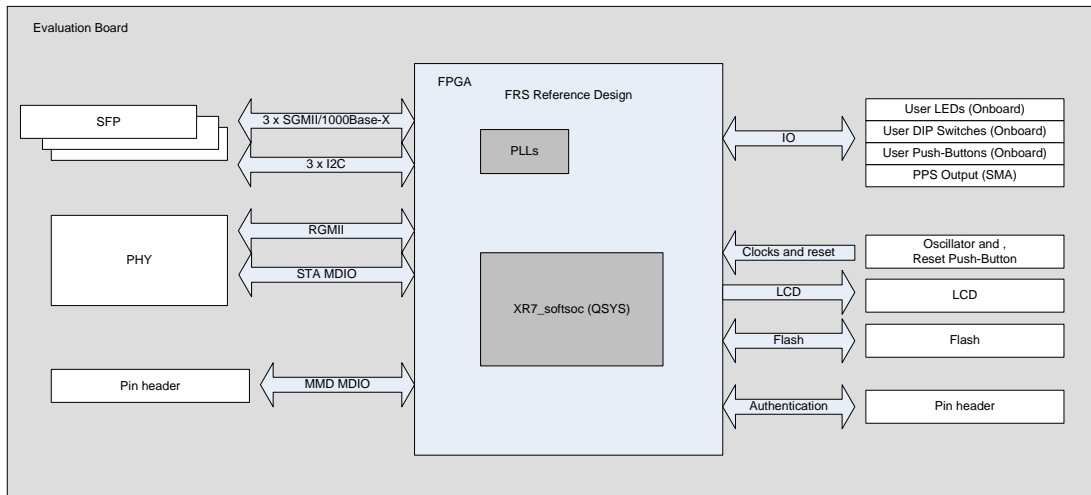
The FRS Reference Design consists of FPGA and SW design. The SW is run on NIOS2 softcore processor. The FPGA design is implemented using Altera QSYS tool that provides a graphical design tool for FPGA systems. FPGA design compilations are made using Altera Quartus II tool and the SW is compiled using 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 VHDL and QSYS 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.



**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)
  - o Redundant Port A
  - o Redundant Port B
  - o Interlink
- RGMII interface for onboard PHY
- 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
  - o C4GX and C5GX have identical LCD command interfaces. For C5GT the interface is I2C.
- Authentication, routed to external pin header (if available)
  - o Not available in C5GX. The design will function for 2 hours.
  - o Note that the Authentication chip [13] 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
  - o C4GX
    - 50 MHz, clk
      - Routed through PLL to LCD clock
      - Routed directly to QSYS design (xr7\_softsoc)
    - 125MHz, eth\_clk
      - Routed through PLL to 125MHz and 25 MHz Ethernet clocks for RGMII
      - Routed directly for SerDes
  - o C5GX



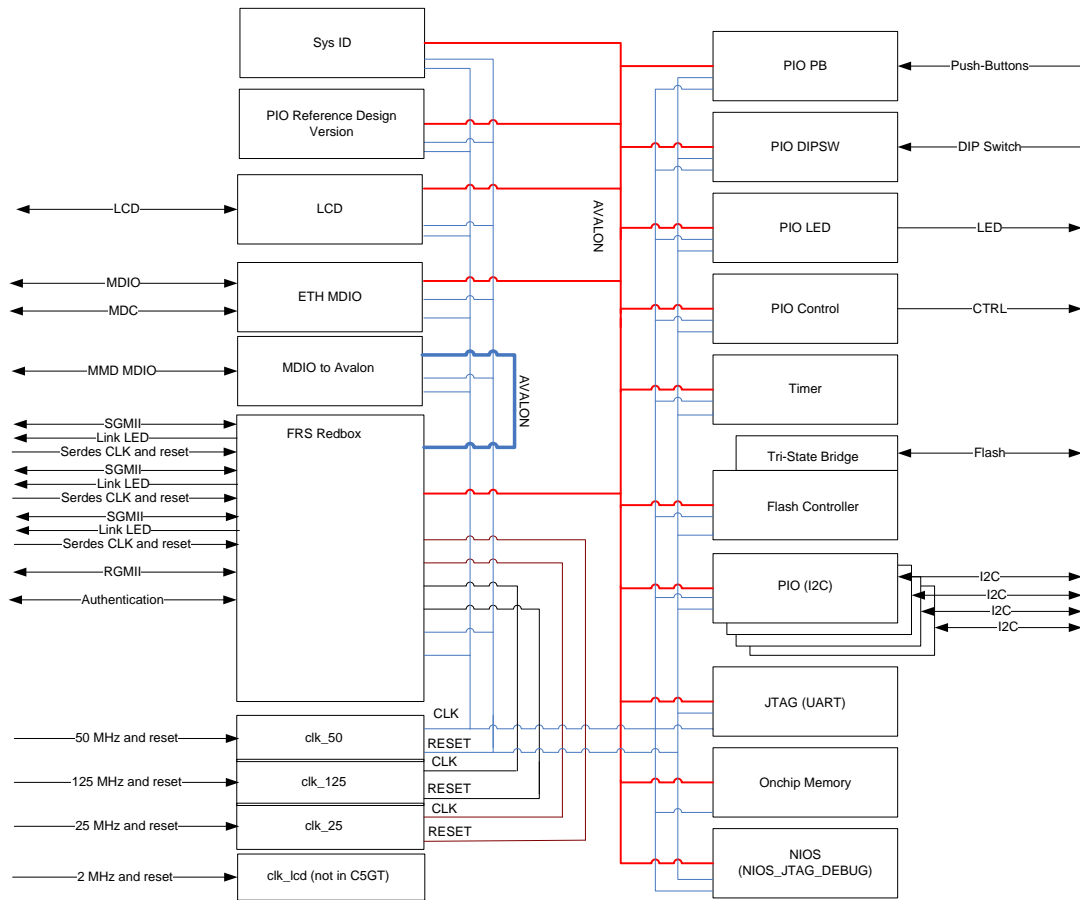
- 50MHz, clk
  - Routed through PLL to LCD clock
  - Routed directly to QSYS design (xr7\_softsoc)
- 125 MHz, eth\_clk
  - Routed through PLL to 125MHz and 25 MHz Ethernet clocks for RGMII
- 156.25 MHz, serdes\_clk
  - Routed through PLL to SerDes
- C5GT
  - 50MHz, clk
    - Routed directly to QSYS design (xr7\_softsoc)
  - 125 MHz, eth\_clk
    - Routed through PLL to 125MHz and 25 MHz Ethernet clocks for RGMII
  - 156.25 MHz, serdes\_clk
    - Routed through PLL to SerDes

The top level design as defined in fpga.vhd includes:

- PLL for LCD clock (in C4GX and C5GX)
- PLL for RGMII clocks
- PLL for Serdes Clock (C5GX and C5GT)
- Clock Control for RGMII clock selections
- QSYS design called xr7\_softsoc

### 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. Clock and reset for 2MHz (LCD), 25, 50 and 125MHz.
  - a. Clock source components are used in QSYS designs to feed clock and resets into the QSYS system/components
2. FRS RedBox
  - a. More detailed description in the next chapter
3. MDIO to Avalon Bridge
  - a. Provide method for the MMD MDIO to access Avalon. For more info on MDIO-to-Avalon Bridge see FRS User Manual Appendix “MDIO-to-avalon Bridge” [8].
  - b. Provide bridge between MMD MDIO and STA MDIO. However, in Reference Design the NIOS is controlling the STA MDIO and there is no access from MMD to STA MDIO.
  - c. Provide automatic PHY status polling functionality
4. ETH MDIO
  - a. Provides MDIO/MDC interface for NIOS. Used to configure external RGMII PHY.
5. LCD
  - a. Provides LCD interface
  - b. In C5GT the block is I2C
6. Reference Design Version
  - a. Includes Reference Design version number
7. SYS ID
  - a. Includes System ID
8. NIOS II Soft processor core
  - a. Includes also JTAG debug module

9. Onchip memory
  - a. Onchip memory is used for SW and for AFEC TX and RX buffers
10. JTAG Uart
  - a. Enables NIOS SW debugging
11. I2C
  - a. Actually a Parallel IO block (PIO), used to generate I2C interface
12. Flash Controller
  - a. Configuration Flash controller
13. Timer
14. PIO Control
  - a. Control signals, currently only PHY Reset
15. PIO LED
  - a. Control LEDs
16. PIO DIPSW
  - a. User DIP switch signals
17. PIO PB
  - a. User Push-Buttons

### 3.2.1 FRS RedBox

The FRS RedBox 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 FRS RedBox component folder. The FRS RedBox component block diagram with current configurations is presented in Figure 4.

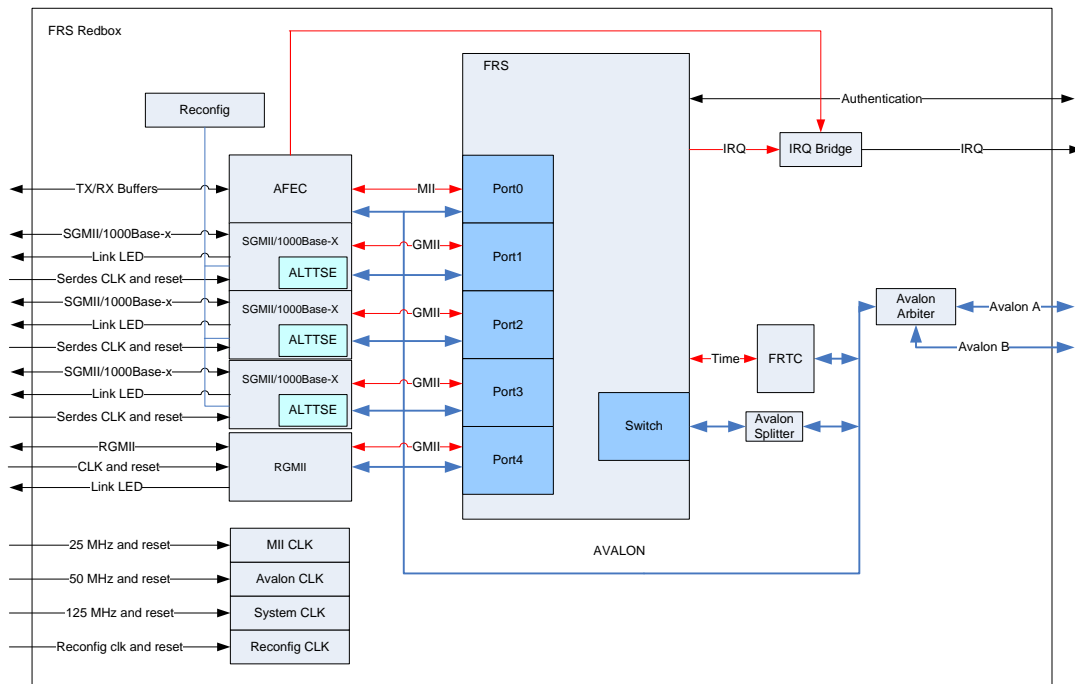


Figure 4. FRS RedBox Component Block Diagram

### FRS

FRS QSYS component is the Ethernet switch core, supporting 3-8 ports, HSR/PRP protocols and 1588 TC functionalities [8]. It can be instantiated also in QSYS as a separate component “Flexibilis Redundant switch” or in the VHDL code.

### 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 [7].

### SGMII/1000BASE-X

SGMII/1000BASE-X adapters provide interface conversion between GMII (FRS) and SGMII/1000BASE-X. This component includes Altera Triple Speed Ethernet (ALT\_TSE) IP, which is used in “PCS only” mode. For ALT\_TSE configuration refer to the ALT\_TSE user guide [1].

### 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

FRS RedBox 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-to-Avalon bridge
- NIOS

### AVALON SPLITTER

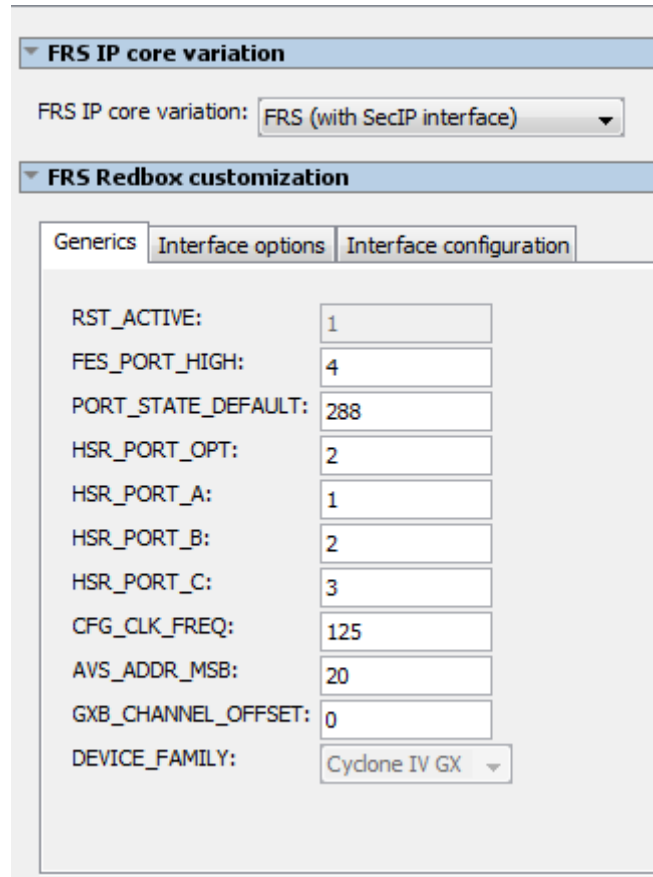
The Avalon Splitter component is generated automatically by the QSYS to support older FRS control mechanisms, even though it would not be required in this case.

## 3.3 FRS RedBox Configuration

This chapter describes how FRS RedBox 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



Parameter	Value
RST_ACTIVE:	1
FES_PORT_HIGH:	4
PORT_STATE_DEFAULT:	288
HSR_PORT_OPT:	2
HSR_PORT_A:	1
HSR_PORT_B:	2
HSR_PORT_C:	3
CFG_CLK_FREQ:	125
AVS_ADDR_MSB:	20
GXB_CHANNEL_OFFSET:	0
DEVICE_FAMILY:	Cyclone IV GX

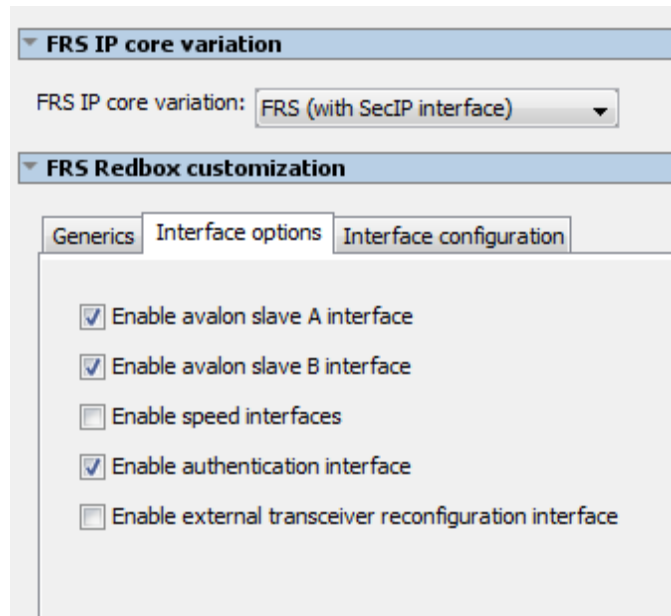
**Figure 5. Generics**

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

- FES\_PORT\_HIGH
  - o Sets the port count. Value 4 means that there are five ports.
- PORT\_STATE\_DEFAULT
  - o Sets the default value for all PORT\_STATE registers. Value 288 is 0x0120, which sets ports to forwarding mode, GMII and 1000Mbps
- HSR\_PORT\_OPT:
  - o Setting '0' would allow redundant functionality in any port.
  - o Setting '1', would allow redundant port capabilities only in two certain ports.
  - o Setting '2' allows redundant functionality in three ports
  - o Setting '3' would disable redundant features
- HSR\_PORT\_A
  - o If HSR\_PORT\_OPT = '1' then this setting determines the port for HSR\_A port. In this case HSR\_A is set to port 1.
- HSR\_PORT\_B
  - o If HSR\_PORT\_OPT = '1' then this setting determines the port for HSR\_B port. In this case HSR\_B is set to port 2.
- HSR\_PORT\_C
  - o If HSR\_PORT\_OPT = '2' then this setting determines the port for the HSR/PRP interlink port. In case HSR\_C is set to port 3
- CFG\_CLK\_FREQ
  - o This defines the system clock frequency for the FRS and it must be set to match the actual clock frequency. In this case it is 125 MHz
- AVR\_ADDR\_MSB

- Sets the Avalon address bus width. As a MSB definition it is actual width minus one.
- GXB\_CHANNEL\_OFFSET
  - Not applicable since only one FRS is instantiated inside one FPGA chip.
  - In case there would be multiple FRS inside one FPGA, the GXB\_CHANNEL\_OFFSET needs to be individual for each.

### 3.3.2 Interface Options



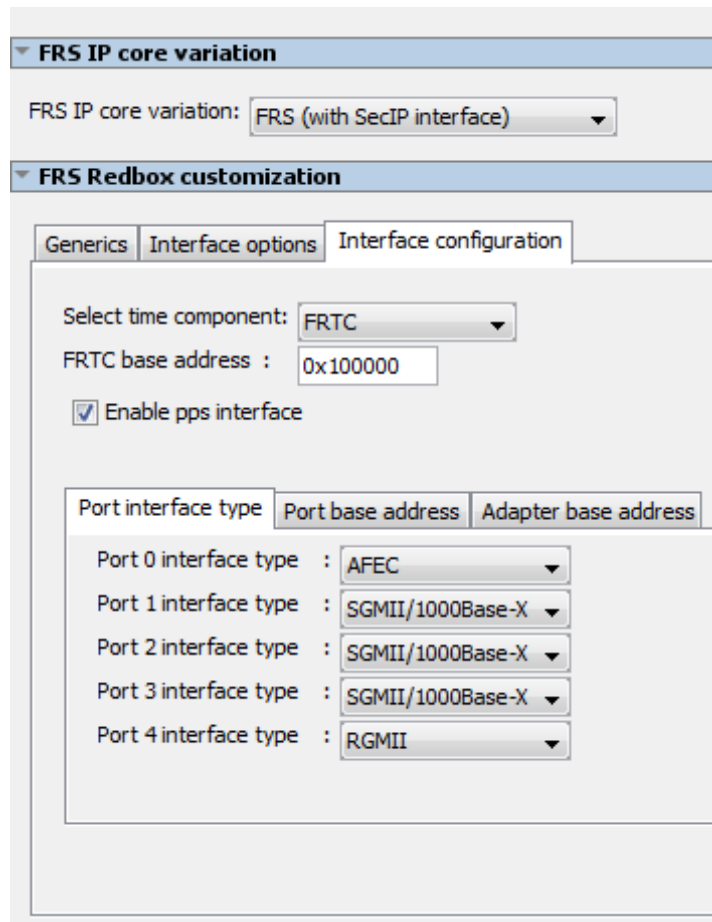
**Figure 6. Interface Options**

In the Interface Options page, Figure 6, it is possible to select what interfaces FRS RedBox provides. In this Reference Design Avalon A and B as well as Authentication interfaces are provided. Avalon A interface is used by the NIOS to access component registers on the Avalon bus. Avalon B is connected in QSYS to the MDIO-to-Avalon bridge. It is not recommended to change these settings without understanding what each setting actually does.

### 3.3.3 Interface Configuration

In the Interface configurations page there are three sub pages.

#### 3.3.3.1 Port Interface Type



The screenshot shows two configuration windows. The top window, titled 'FRS IP core variation', has a dropdown menu for 'FRS IP core variation' set to 'FRS (with SecIP interface)'. The bottom window, titled 'FRS Redbox customization', has three tabs: 'Generics', 'Interface options', and 'Interface configuration'. The 'Interface configuration' tab is active. It contains a 'Select time component' dropdown set to 'FRTC', an 'FRTC base address' text box with '0x100000', and a checked checkbox for 'Enable pps interface'. Below this is a table with three columns: 'Port interface type', 'Port base address', and 'Adapter base address'. The table lists configurations for Port 0 through Port 4.

Port interface type	Port base address	Adapter base address
Port 0 interface type : AFEC		
Port 1 interface type : SGMII/1000Base-X		
Port 2 interface type : SGMII/1000Base-X		
Port 3 interface type : SGMII/1000Base-X		
Port 4 interface type : RGMII		

**Figure 7. Interface Configurations**

The Port Interface type, Figure 7, is used to select interface type for each port. In this case:

- Port 0 is AFEC i.e. it used by the NIOS
- Port 1, 2 and 3 are SGMII/1000BASE-X
- Port 4 is RGMII

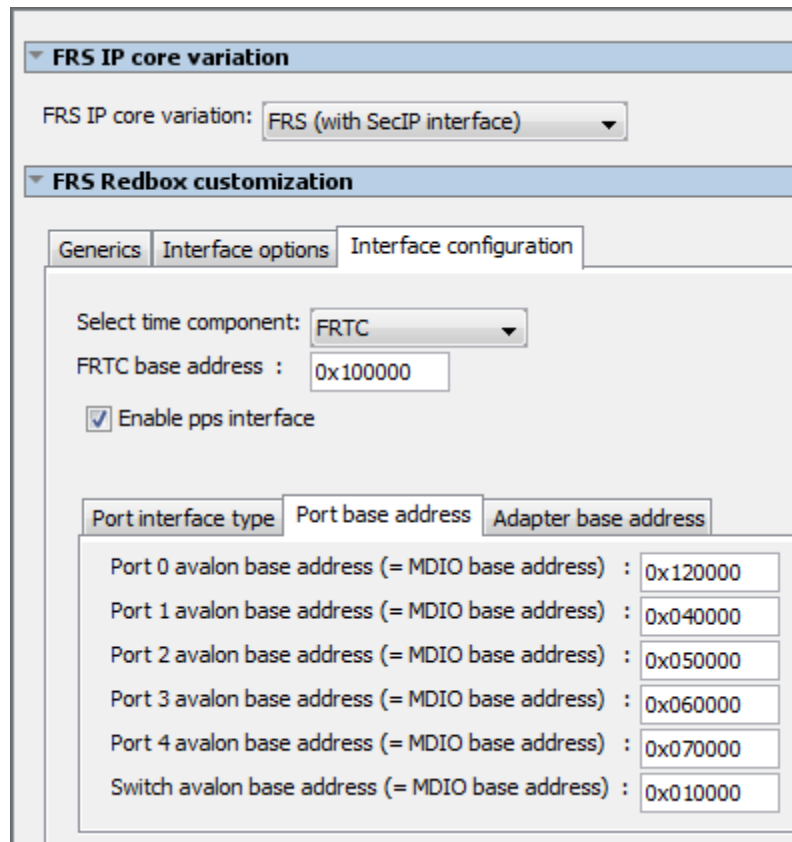
Other possible interface types include:

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

EMAC is used in Cyclone V SoC designs.



### 3.3.3.2 Port Address Configuration



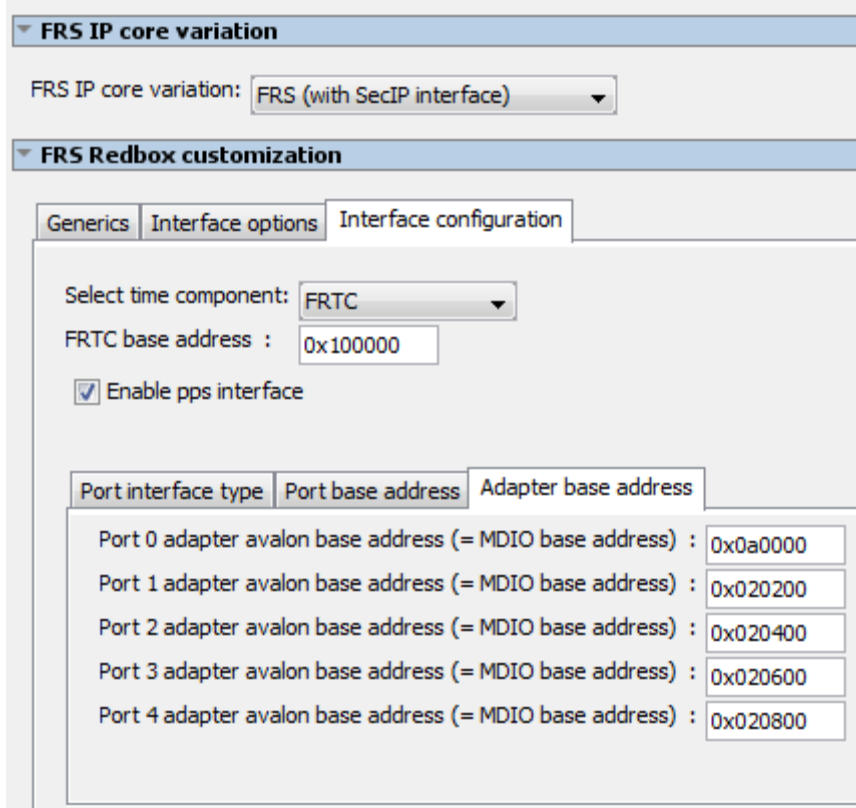
The screenshot shows the 'FRS Redbox customization' interface with the 'Interface configuration' tab selected. It includes a dropdown for 'FRS IP core variation' set to 'FRS (with SecIP interface)', a 'Select time component' dropdown set to 'FRTC', and an 'FRTC base address' field set to '0x100000'. A checked checkbox 'Enable pps interface' is also present. Below, a table lists 'Avalon base address (= MDIO base address)' values for Port 0 through Port 4 and the Switch.

Port interface type	Port base address	Adapter base address
Port 0 avalon base address (= MDIO base address)	0x120000	
Port 1 avalon base address (= MDIO base address)	0x040000	
Port 2 avalon base address (= MDIO base address)	0x050000	
Port 3 avalon base address (= MDIO base address)	0x060000	
Port 4 avalon base address (= MDIO base address)	0x070000	
Switch avalon base address (= MDIO base address)	0x010000	

**Figure 8. Port Address Configurations**

Port base address configuration, Figure 8, defines the Avalon address offset from FRS RedBox base address for FRS Port and Switch configuration registers. Avalon address map is defined in Chapter 3.6.

### 3.3.4 Adapter Address Configuration



The screenshot shows a configuration window with two main sections:

- FRS IP core variation:** A dropdown menu set to "FRS (with SecIP interface)".
- FRS Redbox customization:**
  - Sub-sections: "Generics", "Interface options", and "Interface configuration" (selected).
  - "Select time component:" dropdown set to "FRTC".
  - "FRTC base address:" text box containing "0x100000".
  - Checkbox "Enable pps interface" is checked.
  - Table for adapter base addresses:

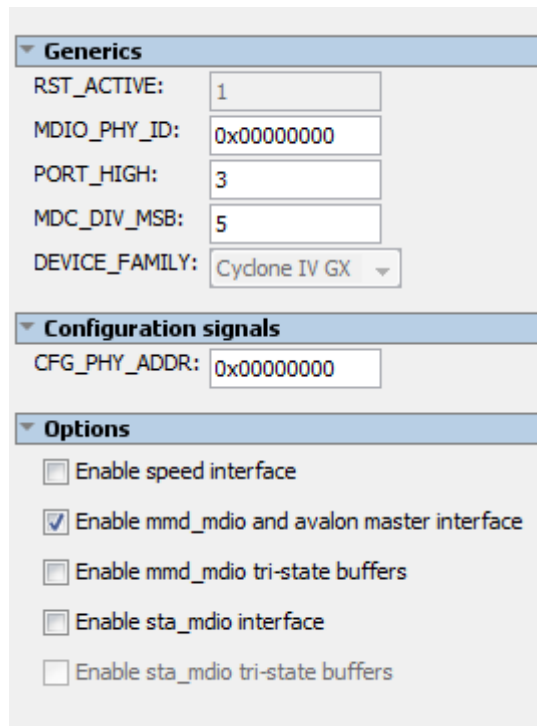
Port interface type	Port base address	Adapter base address
Port 0 adapter avalon base address (= MDIO base address) :		0x0a0000
Port 1 adapter avalon base address (= MDIO base address) :		0x020200
Port 2 adapter avalon base address (= MDIO base address) :		0x020400
Port 3 adapter avalon base address (= MDIO base address) :		0x020600
Port 4 adapter avalon base address (= MDIO base address) :		0x020800

**Figure 9. Adapter Address Configurations**

Figure 9 presents adapter base address configuration, which defines the Avalon address offset from the FRS RedBox base address for adapter registers. Avalon address map is defined in Chapter 3.6.

### 3.4 MDIO-to-Avalon Bridge Configuration

Also the MDIO-to-Avalon Bridge QSYS component (Figure 3) provides certain configurations for the user.



The screenshot shows a configuration window for MDIO-to-Avalon Bridge. It is divided into three sections: Generics, Configuration signals, and Options.

- Generics:**
  - RST\_ACTIVE: 1
  - MDIO\_PHY\_ID: 0x00000000
  - PORT\_HIGH: 3
  - MDC\_DIV\_MSB: 5
  - DEVICE\_FAMILY: Cyclone IV GX
- Configuration signals:**
  - CFG\_PHY\_ADDR: 0x00000000
- Options:**
  - Enable speed interface
  - Enable mmd\_mdio and avalon master interface
  - Enable mmd\_mdio tri-state buffers
  - Enable sta\_mdio interface
  - Enable sta\_mdio tri-state buffers

**Figure 10. MDIO-to-Avalon Bridge Configurations**

Figure 10 presents MDIO-to-Avalon Bridge configurations page with settings used in this case. Configurable generics are:

- MDIO\_PHY\_ID
  - o This is used in the MDIO-to-Avalon Bridge register PHYIDR1/2 [8]. This is set to 0x0 by default.
- PORT\_HIGH
  - o This setting defines the amount of FRS Ports accessed through MDIO-to-Avalon Bridge. Value 0 means one port, value 1 means two ports etc.
- MDC\_DIV\_HIGH
  - o Clock divider for the STA MDC. With this setting the MDC frequency is 1.56MHz since the basic clock is 50 MHz (Avalon) and it divided with  $2^5$  ( $50 \text{ MHz} / 2^5 = 1.56\text{MHz}$ )
- CFG\_PHY\_ADDR
  - o This defines which PHY Addresses are used by external PHY devices. Bit 0 refers to PHY address 1, bit 1 to PHY address 2 etc.. When bit is set to '0', it is not used by external PHY. In this case none of the PHY addresses are reserved for External PHYs since NIOS is used as the master on the STA MDIO via MDIO Controller block as illustrated in Figure 3.
  - o When polling functionality is enabled, addresses that have been set to '1' cannot be accessed from the MMD MDIO.
- Enable Speed interface
  - o Speed interface is used to notify speed polling status for the FRS. In this Reference Design the PHY polling is not used.
- Enable mmd\_mdio and Avalon master interface
  - o Enables possibility to access Avalon (FRS RedBox) from the mmd\_mdio
- Enable sta\_mdio
  - o Enables possibility to access sta\_mdio through mmd\_mdio. However, in this case NIOS is connected to the STA so there is no need for this interface.
- Enable tri-state buffers
  - o Enable tri-state buffers inside the FRS RedBox.

### 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 FRS RedBox provides also other interface adapters as mentioned in chapter 3.3.3.1.

Some of the adapters include registers for configuration and status. Chapters below define these registers.

#### 3.5.1 SGMII/1000BASE-X

ALT\_TSE is Altera's IP that implements 1000BASE-X/SGMII PCS. For more information, see ALT\_TSE user guide [12].

Address	Register	Description								
0x0000	ID	Reset: 0 x XX B2 Id and version are present in this register <table border="1"> <tr> <td>Bits</td> <td>7-0:</td> <td>RO</td> <td>Id (0xB2)</td> </tr> <tr> <td>Bits</td> <td>15-8:</td> <td>RO</td> <td>Version</td> </tr> </table>	Bits	7-0:	RO	Id (0xB2)	Bits	15-8:	RO	Version
Bits	7-0:	RO	Id (0xB2)							
Bits	15-8:	RO	Version							
0x0001	LINK_STATUS	Reset: 0 x 00 00 Link status register <table border="1"> <tr> <td>Bit</td> <td>0:</td> <td>R/W</td> <td>Link status. Must be updated by the user 0 = link down 1 = link up  This link status info is used only for link LED enable.</td> </tr> <tr> <td>Bits</td> <td>15-1:</td> <td>RO</td> <td>Reserved</td> </tr> </table>	Bit	0:	R/W	Link status. Must be updated by the user 0 = link down 1 = link up  This link status info is used only for link LED enable.	Bits	15-1:	RO	Reserved
Bit	0:	R/W	Link status. Must be updated by the user 0 = link down 1 = link up  This link status info is used only for link LED enable.							
Bits	15-1:	RO	Reserved							
0x0002-0x001f	RESERVED	Reset: 0 x 00 00 00 00 <table border="1"> <tr> <td>Bits</td> <td>15-0:</td> <td>RO</td> <td>Reserved</td> </tr> </table>	Bits	15-0:	RO	Reserved				
Bits	15-0:	RO	Reserved							
0x0020-0x003f	ALT_TSE_PCS	Altera's Triple Speed Ethernet IP PCS configuration address space. See Table 6–9 in [12].								

**Table 1. ALT\_TSE Adapter Configuration Registers**

#### 3.5.2 1000BASE-X

1000BASE-X adapter implements 1000BASE-X PCS. It can be used instead of ALT\_TSE adapter. 1000BASE-X adapter includes Altera's IP; ALTGX B for Cyclone IV and Custom PHY for Cyclone V.

Address	Register	Description								
0x0000	ID	Reset: 0 x XX B3 Id and version are present in this register <table border="1"> <tr> <td>Bits</td> <td>7-0:</td> <td>RO</td> <td>Id (0xB3)</td> </tr> <tr> <td>Bits</td> <td>15-8:</td> <td>RO</td> <td>Version</td> </tr> </table>	Bits	7-0:	RO	Id (0xB3)	Bits	15-8:	RO	Version
Bits	7-0:	RO	Id (0xB3)							
Bits	15-8:	RO	Version							
0x0001	LINK_STATUS	Reset: 0 x 00 00 Link status register <table border="1"> <tr> <td>Bit</td> <td>0:</td> <td>RO</td> <td>Link status. Informs 1000BASE-X link status. 0 = link down 1 = link up</td> </tr> <tr> <td>Bits</td> <td>15-1:</td> <td>RO</td> <td>Reserved</td> </tr> </table>	Bit	0:	RO	Link status. Informs 1000BASE-X link status. 0 = link down 1 = link up	Bits	15-1:	RO	Reserved
Bit	0:	RO	Link status. Informs 1000BASE-X link status. 0 = link down 1 = link up							
Bits	15-1:	RO	Reserved							

**Table 2. 1000BASE-X Adapter Configuration Registers**

#### 3.5.3 100BASE-FX

100BASE-FX adapter implements 100BASE-FX PCS and includes Altera's IP; ALTGX B for Cyclone IV and Native PHY for Cyclone V.

Address	Register	Description
0x0000	ID	Reset: 0 x XX B3 Id and version are present in this register Bits 7-0: RO Id (0xB4) Bits 15-8: RO Version
0x0001	LINK_STATUS	Reset: 0 x 00 00 Link status register Bit 0: RO Link status. Informs 100BASE-FX link status. 0 = link down 1 = link up Bits 14-1: RO Reserved

Table 3. 100BASE-FX Adapter Configuration Registers

### 3.5.4 GMII

GMII adapter implements GMII to GMII conversion i.e. makes GMII interface suitable for most PHYs.

Address	Register	Description
0x0000	ID	Reset: 0 x XX B3 Id and version are present in this register Bits 7-0: RO Id (0xB0) Bits 15-8: RO Version
0x0001	LINK_STATUS	Reset: 0 x 00 00 Link status register Bit 0: RO Link status. Informs GMII link status. 0 = link down 1 = link up Bits 14-1: RO Reserved Bits 15-1: RW Link Status PHY. Link status is provided for this block via link status signal. 0 = Not available. Link status is up by default. 1 = Available
0x0002	TX_CLK_MUX	Reset: 0 x 00 00 Link status register Bits 1-0: RW Transmit Clock selection. 0X = 125 MHz 10 = 25 MHz 11 = 2.5 MHz Bits 15-2: RO Version

Table 4. GMII Adapter Configuration Registers

### 3.5.5 RGMII

RGMII adapter implements GMII to RGMII conversion.

Address	Register	Description
0x0000	ID	Reset: 0 x XX B3 Id and version are present in this register Bits 7-0: RO Id (0xB1) Bits 15-8: RO Version
0x0001	LINK_STATUS	Reset: 0 x 00 00 Link status register

		Bit	0:	RO	Link status. Informs RGMII link status. 0 = link down 1 = link up
		Bits	2-1:	RO	Auto-negotiation result for speed 00 = Unknown 01 = 1000 Mbps 10 = 100 Mbps 11 = 10 Mbps
		Bits	15-3:	RO	Reserved

Table 5. RGMII Adapter Configuration Registers

### 3.6 Avalon Address Map

The Avalon Address Map is based on the settings in QSYS and its components. The Table 6 below defines the Address map for this Reference Design.

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		
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	0x0400_0300		
FRS RedBox	0x0600_0000		
Switch			
<i>Switch registers</i>	0x0601_0000		
<i>TS</i>	0x0601_1000		
<i>VLAN</i>	0x0601_2000		
FRS Port 0			
<i>GEN</i>	0x0612_0000		

Component	BUS and Base Address		
<i>HSR</i>	0x0612_1000		
<i>PTP</i>	0x0612_2000		
<i>CNT</i>	0x0612_3000		
<i>IPO</i>	0x0612_4000		
<b>FRS Port 1</b>			
<i>GEN</i>	0x0604_0000		
<i>HSR</i>	0x0604_1000		
<i>PTP</i>	0x0604_2000		
<i>CNT</i>	0x0604_3000		
<i>IPO</i>	0x0604_4000		
<b>FRS Port 2</b>			
<i>GEN</i>	0x0605_0000		
<i>HSR</i>	0x0605_1000		
<i>PTP</i>	0x0605_2000		
<i>CNT</i>	0x0605_3000		
<i>IPO</i>	0x0605_4000		
<b>FRS Port 3</b>			
<i>GEN</i>	0x0606_0000		
<i>HSR</i>	0x0606_1000		
<i>PTP</i>	0x0606_2000		
<i>CNT</i>	0x0606_3000		
<i>IPO</i>	0x0606_4000		
<b>FRS Port 4</b>			
<i>GEN</i>	0x0607_0000		
<i>HSR</i>	0x0607_1000		
<i>PTP</i>	0x0607_2000		
<i>CNT</i>	0x0607_3000		
<i>IPO</i>	0x0607_4000		
<b>Adapter P0</b>			
<i>AFEC</i>	0x060A_0000		
<b>Adapter P1</b>			
<i>SGMII/1000BASE-X</i>	0x0602_0200		
<b>Adapter P2</b>			
<i>SGMII/1000BASE-X</i>	0x0602_0400		
<b>Adapter P3</b>			
<i>SGMII/1000BASE-X</i>	0x0602_0600		
<b>Adapter P4</b>			
<i>RGMII</i>	0x0602_0800		
<b>FRTC</b>	0x0610_0000		

Table 6. 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 11. 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.

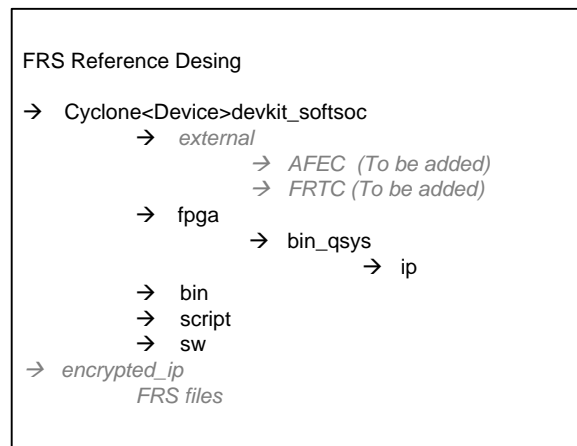



Figure 11. Folder Structure

### 3.7.2 QSYS Generation

The Reference Design includes *xr7\_softsoc.qsys* file, which includes information about the QSYS system used in this design. The QSYS system needs to be generated before the first compilation 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>devki.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:

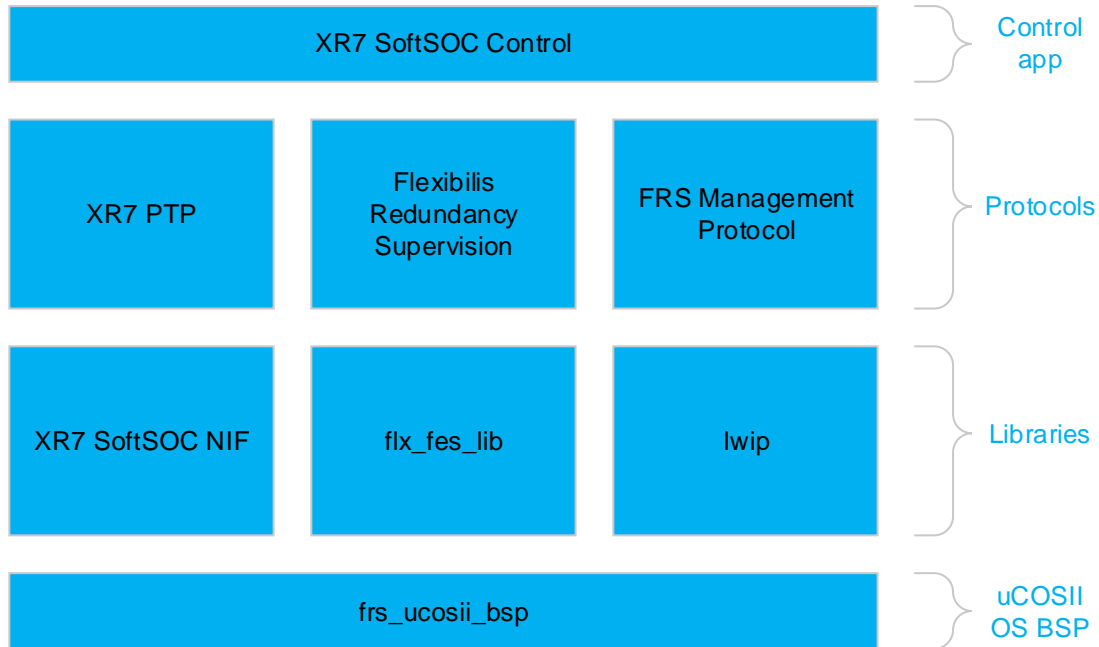
- FRS (Flexibilis)
- AFEC (Flexibilis)
- FRTC (Flexibilis)
- ALT\_TSE (Altera)
- NIOS (Altera)

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 12.



**Figure 12. SW Reference Design**

The modules depicted in Figure 12 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.
xr7_ptp	XR7 PTP – PTP protocol stack [3], [4]. Source code is delivered separately and requires a license from Flexibilis Oy.
flx_redundancy_supervision	HSR/PRP Supervision protocol [2], [5]. Source code is delivered separately and requires a license from Flexibilis Oy.
frs_management_protocol	FRS Management Protocol [6]. Source code is delivered separately and requires a license from Flexibilis Oy.
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, open source. Source code is located in Reference Design release in FESHA00E00-FBIT-V<ver>\cyclone4GXdevkit_softsoc\sw\lwip directory.
frs_ucosii_bsp	µCOS-II BSP for NIOS, generated by Altera NIOS II EDS according to delivered configuration. Configuration is available in FESHA00E00-FBIT-V<ver>\<board>\sw\frs_ucosii_bsp directory.

**Table 7. 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 8.

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 8. XR7 SoftSoC Control Module Files**

### 4.1.2 XR7 PTP

XR7 PTP implements Precision Time Protocol [3]. XR7 PTP is described in detail in XR7 PTP Design Specification [4].

### 4.1.3 Flexibilis Redundancy Supervision

Flexibilis Redundancy Supervision implements HSR/PRP Supervision protocol [2]. Flexibilis Redundancy Supervision is described in detail in Flexibilis Redundancy Supervision Design Specification [5].

### 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 [6].

### 4.1.5 XR7 SoftSoC NIF

XR7 SoftSoC NIF module contains drivers and other helper functions. The source code files are listed in Table 9.

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.c src/afec.c inc/flx_afec.h src/flx_afec.c	AFEC IP (Ethernet MAC) driver [7]
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/fes_if.h src/fes_ctrl.c	Device driver for FRS IP. [8]
inc/frtc.h src/frtc.c	Device driver for FRTC IP. [9]
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/sfp_ctrl.h src/sfp_ctrl.c	SFP module management and control driver. Uses I2C to access SFP modules.

Table 9. XR7 SoftSoC NIF Module Files

#### 4.1.6 flx\_fes\_lib

The library contains helper functions for managing FRS. The source code files are listed in Table 10.

File	Description
flx_fes.h flx_fes.c	Helper functions for configuring FRS IP [8]. Includes for example reading and writing of FRS registers and IPO settings.

Table 10. flx\_fes\_lib Module Files

#### 4.1.7 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 [10].

#### 4.1.8 frs\_ucosii\_bsp

Board Support Package (BSP) generated by Altera NIOS II EDS tool. Includes µCOS-II operating system. More information can be found in Altera and Micrium [11].

## 4.2 Compilation

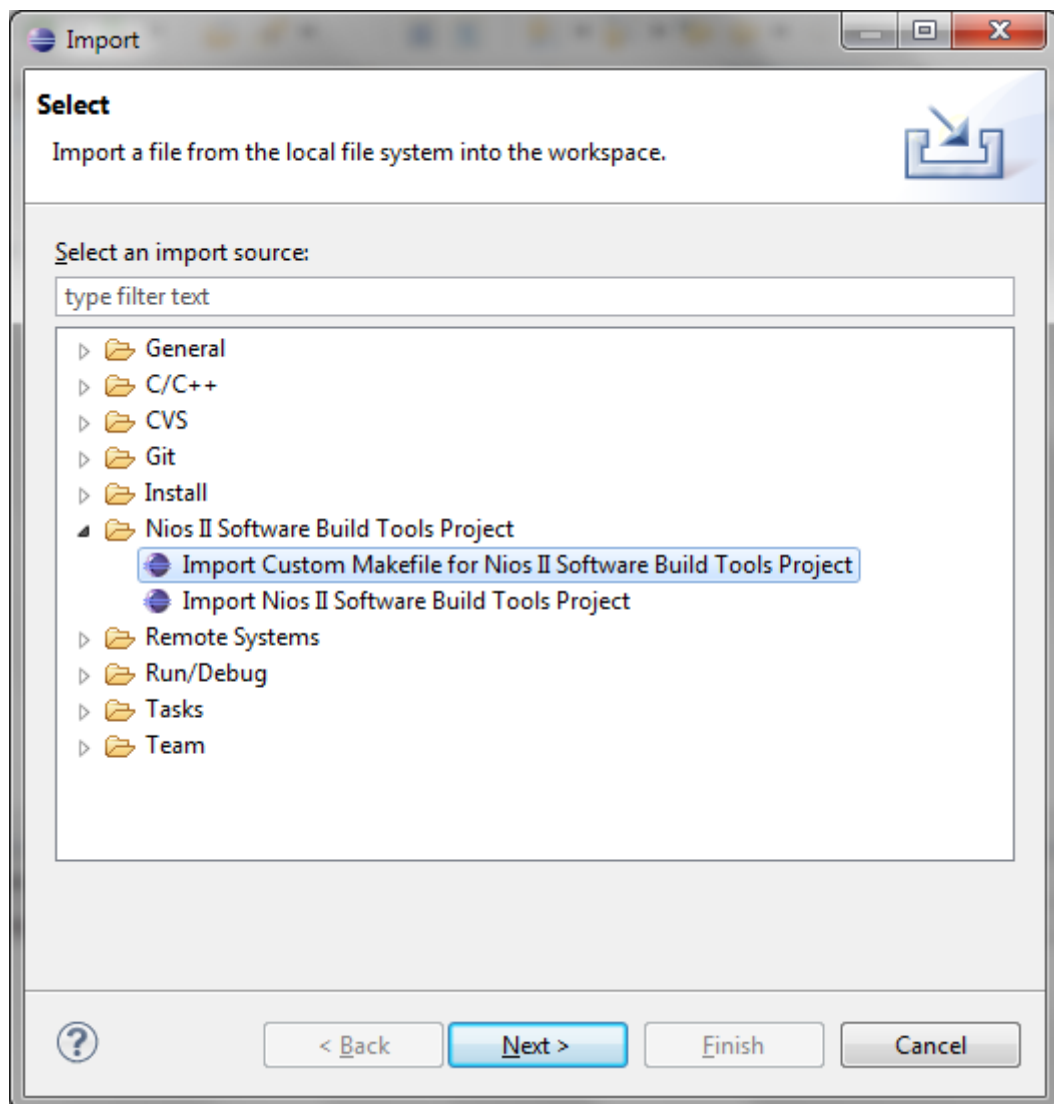
FRS Reference Design SW compilation is done in Altera NIOS II EDS. All the module sources listed in Table 7 need to be placed under FESHA00E00-FBIT-V<ver>\<board>\sw\ directory. SW Reference Designs for different boards use the same module sources. Only frs\_ucosii\_bsp, which is generated by the Altera NIOS II EDS tool is board dependent.

### 4.2.1 Import Projects

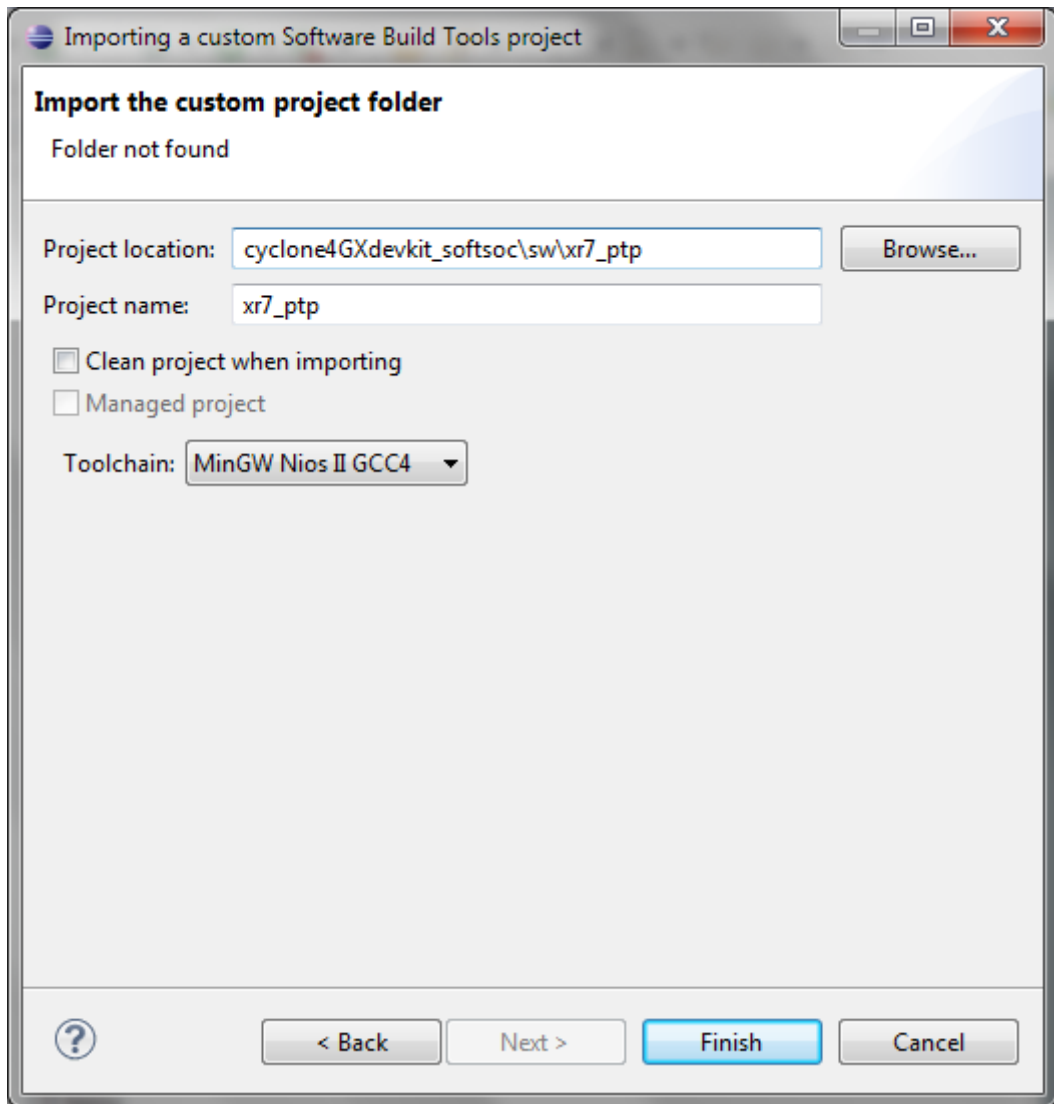
Every SW module is implemented as a separate project in NIOS II EDS. They have to be imported separately and after all the projects have been imported, the application can be compiled.

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).

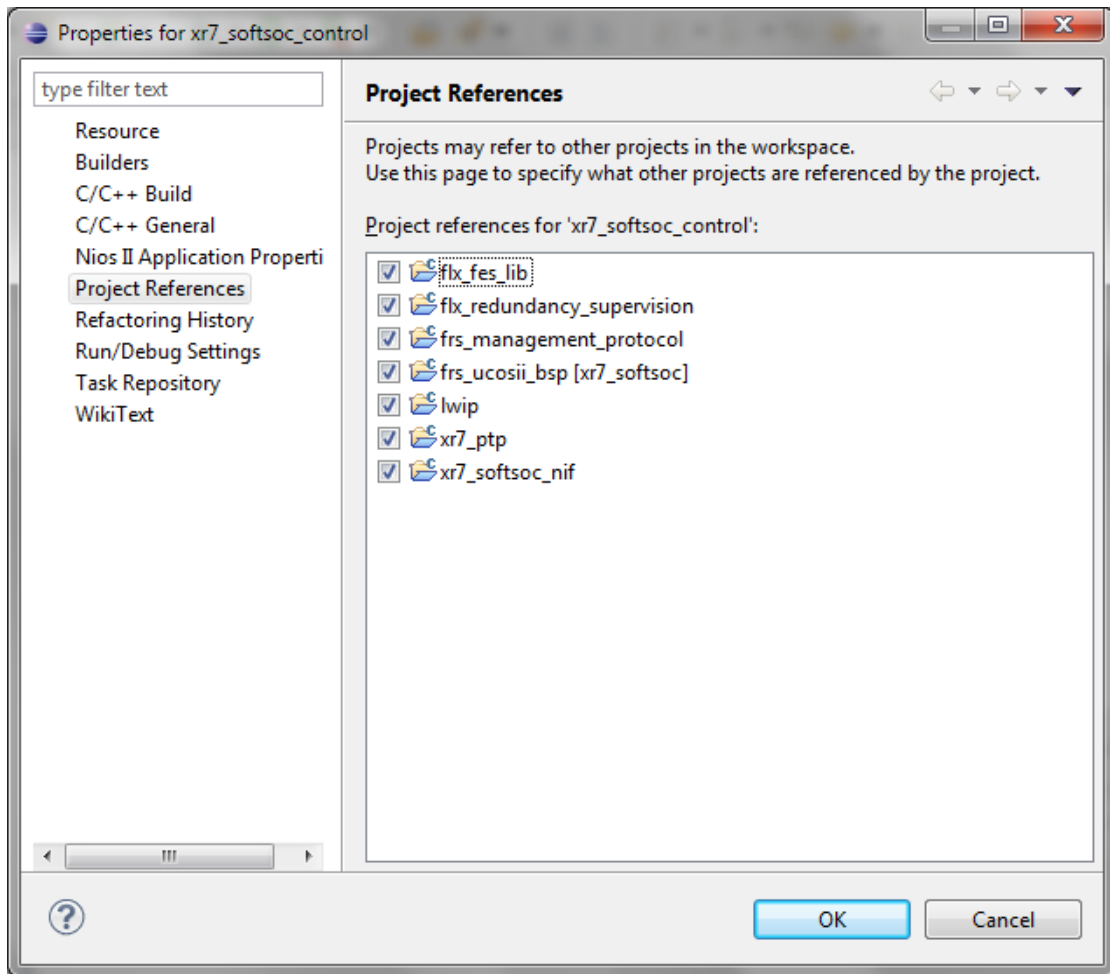


3. Click Browse and select SW module directory, for example xr7\_ptp. Set "Project name:" the module name, in this example xr7\_ptp, and click Finish.
4. Perform the steps 1-3 for every SW module.

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\_ucosii\_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

<b>Term</b>	<b>Description</b>
AFEC	Advanced Flexibilis Ethernet Controller
BSP	Board Support Package
FPGA	Field Programmable Gate Array
FRS	Flexibilis Redundant Switch
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] ALT\_TSE user guide, [http://www.altera.com/literature/ug/ug\\_ethernet.pdf](http://www.altera.com/literature/ug/ug_ethernet.pdf)
- [2] Standard IEC 62439-3:2011
- [3] IEEE standard 1588-2008
- [4] XR7 PTP design specification, xr5\_ptp\_design.pdf
- [5] Flexibilis Redundancy Supervision design specification, flx\_redundancy\_supervision\_design.pdf
- [6] FRS Management Protocol, FRS\_management\_protocol.pdf
- [7] Advanced Flexibilis Ethernet Controller (AFEC) User Manual, AFEC\_user\_manual.pdf
- [8] Flexibilis Redundant Switch (FRS) Manual, FRS\_Manual.pdf
- [9] FRTC User Manual, FRTC\_user\_manual.pdf
- [10] lwIP - A Lightweight TCP/IP stack, open source, <http://savannah.nongnu.org/projects/lwip/>
- [11] Micrium MicroC/OS-II Real-Time Operating System, <http://www.altera.com/devices/processor/nios2/tools/embed-partners/micrium/emb-micrium.html>
- [12] Triple-Speed Ethernet MegaCore Function User Guide, version 13.1, December 2013, [http://www.altera.com/literature/ug/ug\\_ethernet.pdf](http://www.altera.com/literature/ug/ug_ethernet.pdf)
- [13] IP License Authentication, Security Chip, Manual, version 1.2. [http://www.flexibilis.com/downloads/Security\\_Chip.pdf](http://www.flexibilis.com/downloads/Security_Chip.pdf)