Dictionary Code	D741
Edition	01
Revision	00
Number of pages	18
State	Approved

SATURN-SIL 2 Modules

FPGA Integration Manual

Document history

Ed	Rev	Date	Author	Comment
01	00	13/08/2015	D.RAMBAUD	First release

Table of content

Table of Figures

Table of Tables

I INTRODUCTION

This document is intended for developers of control electronics who want to design their own SATURN modules up to SIL 2. Such modules shall communicate on the SATURN ring network and thus, shall implement the full SATURN communication stack defined in the document [R2]. The real time part of the SATURN communication stack can be implemented in a FPGA. The purpose of this document is to explain how to implement this FPGA. It handles the following integration aspects:

- SATURN modules general architecture,
- FPGA cabling,
- FPGA SPI protocol,
- FPGA register set.

It is assumed that the user of this manual is familiar with:

- SATURN concept and architecture [R1],
- SATURN protocol [R2],
- FPGA implementation,
- SPI communication bus.

This manual addresses the integration of a SATURN FPGA that can be used for applications with a Safety Integrity up to level 2. Please refer to document xxx for implementation guidelines of a SATURN FPGA intended for SIL 4 applications.

II RELATED DOCUMENTS

Ref.	Title	Reference	Version
[R1]	SATURN – Spécifications fonctionnelles et techniques	D411	
[R2]	SATURN – Spécification du protocole	D411-A	
[R3]	SATURN – "Document chapeau pour donées en open source"		

III LEGENDS AND ABBREVIATIONS

FIFO	First In First Out
FPGA	Field Programmable Grip Array
IID	Internal Identification
JTAG	Joint Test Action Group
MSB	Most Significant Bit
RFU	Reserved for Future Use
SPI	Serial Peripheral Interface
TID	Transport Identification
VHDL	Very High Speed Hardware Description Language

IV SATURN MODULES GENERAL ARCHITECTURE

A SATURN module is a slave agent on the SATURN ring network. It acts as a gateway between the SATURN network and an operational application that can be Input / Output management, Data processing, Filed bus interface…

Whatever the final application of a SATURN module, it should have the following architecture.

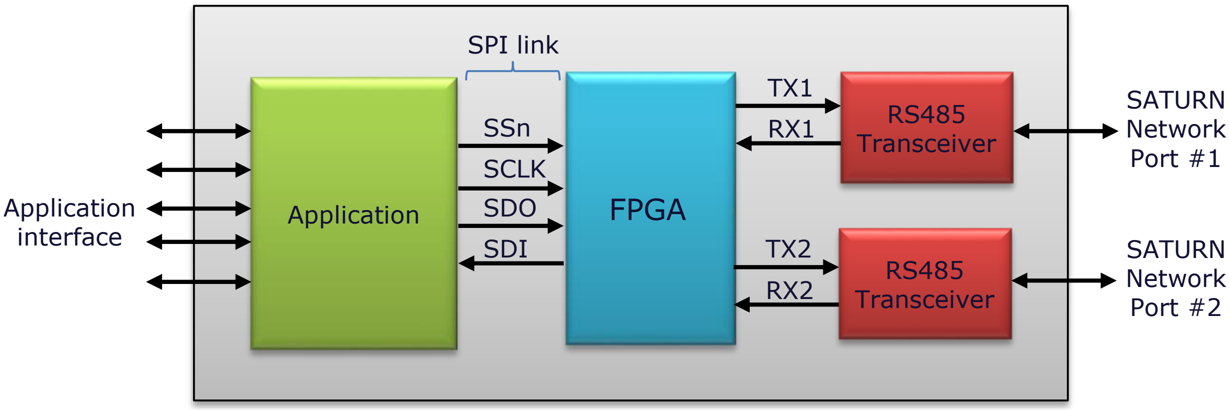

Figure 1 : SATURN Module general architecture

The FPGA ensures the low layers of the SATURN protocol stack.

The application part of the SATURN module can be ensured by a microprocessor, a microcontroller, another FPGA, or any processing unit able to handle:

- The communication stack application layer,
- The specific operations dedicated to the final application (I/O management, field bus controller, …).

FPGA and application part interface through a SPI link presented in §VI.

Note: The above architecture is suitable for applications up to SIL 2. The FPGA itself does not contribute in any safety functions and is design according to a SIL 0. The intended SIL shall be fully handled by the application part.

V FPGA Cabling

The SATURN starter kit comes with a predefined FPGA, available as binary configuration files. This section explains how to implement the FPGA in your own design.

Note: FPGA VHDL source code is available on demand. Please contact us for any special demand such as current FPGA adaptation to your own design or support for implementation in another FPGA family or brand.

The provided FPGA design is for a Xilinx Spartan 6 device referenced: XC6SLX4-2TQG144

The following table gives the FPGA pinning.

Name	Pin #	Direct ion	Definition
CLK_24	P88	IN	FPGA main clock. Shall be 24MHz +/- 50ppm
RST_n	P84	IN	FPGA main reset. Active low.
CONF_OKn	P79	OUT	Low when the FPGA is correctly configured after power up. Pull Up or Down according to Xilinx HSWAPEN configuration pin when configuration is failed.
RUNn	P78	OUT	Toggle alternatively High and Low when module is not in fallback position. Remains high while module is in fallback position.
FALLBACKn	P75	OUT	Active Low when module is in fallback position. High when not is fallback.
SDO_OFF	P50	IN	When High, SDO signal is in high impedance state. When Low, SDO behaves as described.
SL485_RX1	P140	IN	SATURN ring network port #1 receiver line.
SL485_TX1	P139	OUT	SATURN ring network port #1 transmitter line.
SL485_RX2	P132	IN	SATURN ring network port #2 receiver line.
SL485_TX2	P131	OUT	SATURN ring network port #2 transmitter line.
SSn	P11	IN	SPI slave select. Active Low.
SCLK	P14	IN	SPI Serial Clock.
SDI	P12	IN	SPI Serial Data In. Must be connected to SPI Master SDO.
SDO	P10	OUT	SPI Serial Data Out. Must be connected to SPI Master SDI. Inhibited by SDO_OFF.

Table 1 – FPGA Pinning

All other I/O pins shall be left unconnected. Please refer to the device datasheet for the other mandatory pins cabling, such as power supply, JTAG and configuration pins.

VI FPGA SPI PROTOCOL

SATURN FPGA shall be managed through various 8 bits registers (see §VII) accessible in read and write from a SPI interface.

The FPGA behaves as a slave on the SPI interface that shall be configured at master side as follow:

- SCLK Maximum frequency : 10MHz
- CPOL = 1
- CPHA = 0
- Framed mode (frames delimited by SSn signal).

All exchanges on SPI bus are initiated by the SPI master. Exchanged frames can be of any length, actual frame length being defined by SSn signal.

All frames on the SPI bus are composed of successive bytes transmitted MSB first, and shall be as follow:

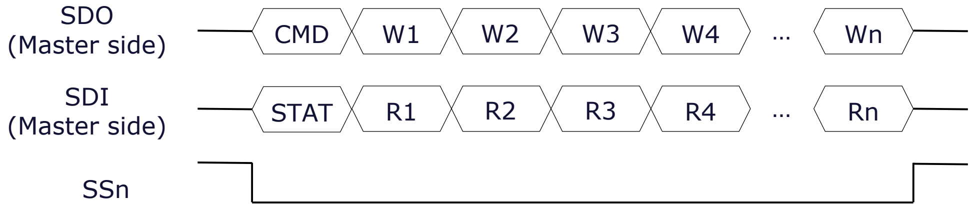

Figure 2 : SPI frame structure

The first byte emitted by SPI master on the SDO line (CMD) is a command byte, composed of the FPGA register address to be read or written (see §VII), coded on 7 MSB, followed by a '1' bit for a read operation, or a '0' bit for a write operation.
Thus CMD byte is built as follow:

A6	A5	A4	A3	A2	A1	A0	1	For a read operation at register address [A6:A0]
A6	A5	A4	A3	A2	A1	A0	0	For a write operation at register address [A6:A0]

Figure 3 : CMD byte structure

Following bytes on the SDO line can be either:

- Byte(s) to be written in FPGA register(s) pointed out by address [A6:A0] and subsequent addresses, in case of write operation.
- Stuffing bytes of any value in case of read operation.

In the same time, SPI master can monitor its SDI line. The first byte sent by the FPGA on the master SDI line is the STATUS byte as defined in §VII.4. Then, following bytes sent on the master SDI line are content of registers pointed out by address [A6:A0] and subsequent addresses.

Note: When SPI master accesses in read or write a FPGA register identified as I/O register, the FPGA initializes an internal pointer with the address coded in the CMD byte, then increases it automatically by 1 for each byte that follows the CMD byte in order to access subsequent registers. When the accessed register is of FIFO type, the internal pointer is not increased and all bytes are read out or written at the same address.

Note: In case of write operation, the register value read by the FPGA and sent on the master SDI line is the register value before the write.

Note: In case of write operation in a register identified as write only, the value returned by the FPGA on the master SDI line shall be discarded and not interpreted by the master.

Note: If SPI master just want to read the FPGA STATUS register, it can issue a single byte frame composed of any CMD byte. In this case, it will read back only the STATUS byte, and the frame will be of no action since not followed by data or stuffing bytes.

VII FPGA Register set

This section defines the overall register set implemented in the FPGA and accessible from SPI interface. All registers are 8 bits registers, mapped at the following addresses:

REGISTER Name	Type	Permission*	Address
IID[63:56]	I/O	RO	00h
IID[55:48]	I/O	RO	01h
IID[47:40]	I/O	RO	02h
IID[39:32]	I/O	RO	03h
IID[31:24]	I/O	RO	04h
IID[23:17]	I/O	RO	05h
IID[16:8]	I/O	RO	06h
IID[7:0]	I/O	RO	07h
TID	I/O	R/W	08h
CONTROL	I/O	R/W	09h
STATUS	I/O	R/W	0Ah
FRMRX_SIZE1	I/O	RO	0Bh
FRMRX_SIZE2	I/O	RO	0Ch
FRMTX_FREE	I/O	RO	0Dh
FIFO_RX1	FIFO	RO	0Eh
FIFO_RX2	FIFO	RO	0Fh
FIFO_TX	FIFO	WO	10h
VERSION	I/O	RO	11h
RFU			12h-15h
TRAFIC	I/O	RO	16h

Table 2 – FPGA registers mapping

** RO: Read Only, WO: Write Only, R/W: Read and Write supported*

In next sections, each register, or set of register is defined as follow:

Register Name								
7	6	5	4	3	2	1	0	Bit number
R/W	WO	WO	WO	R/W	R/W	R/W	R/W	Bit permission (RO, WO, R/W)
BN7	BN6	BN5	BN4	BN3	BN2	BN1	BN0	Bit name
1	1	0	0	0	1	0	0	Value after FPGA reset ('-' for undefined)

VII.1 IID

IID	
63	0
RO	
IID[63..0]	
xxxxxxxxh	

IID: Returns the IID that shall be used by the application for TID assignment. It corresponds to the FPGA internal unique serial number.

VII.2 TID

TID	
7	0
R/W	
TID[7..0]	
8Fh	

TID: TID value that will be used by the FPGA to manage the link layer of the SATURN protocol stack. This register shall be configured by the application part upon reception of a TID assignment SATURN frame.

VII.3 CONTROL

CONTROL							
7	6	5	4	3	2	1	0
R/W			WO	R/W	R/W	R/W	R/W
REPLI	RFU		TOPCY	STRTX	CLRTX	CPY2	CPY1
1			-	0	1	0	0

CPY1: Authorizes/Inhibits copy of an incoming frame on SATURN ring network port #1 to port #2

0b: Copy authorized

1b: Copy inhibited

CPY2: Authorizes/Inhibits copy of an incoming frame on SATURN ring network port #2 to port #1

0b: Copy authorized

1b: Copy inhibited

CLRTX: Flushes the content of the FIFO TX

0b: FIFO TX operational

1b: FIFO TX is empty, write is not available

STRTX: Tells to FPGA that it can transmit the SATURN frame previously stored in the FIFO TX. The bit is automatically cleared when the command has been taken into account.

0b: Write is without effect. Read indicates that the emitting command has been taken into account.

1b: Read indicates that the emitting command has not yet been taken into account. Write tells the FPGA to emit the SATURN frame in FIFO TX.

TOPCY: Synchronizes the FPGA with the SATURN communication cycle.

0b: Without effect

1b: Indicates the SATURN communication cycle start

REPLI: Tells the FPGA the application part mode (fallback or not – impacts only RUNn and FALLBACKn FPGA pins).

0b: Module in operational mode

1b: Module in fallback mode.

VII.4 STATUS

STATUS							
7	6	5	4	3	2	1	0
	R	R/W*	R/W*	R/W*	R/W*	R	R
RFU	TXNE	OVF2	OVF1	BFO2	BFO1	FRM2	FRM1
	0	0	0	0	0	0	0

* Status bits of type ‘Write One Clear’. Writing a ‘1’ in those bits clear the corresponding bit, writing à ‘0’ is without effect.

The following bits are all active high.

FRM : At least one frame is available in FIFO_RX1

FRM2: At least one frame is available in FIFO_RX2

BFO1: At least one corrupted frame has been received on port #1 (the reason can be a wrong CRC, a stuffing error, an unknown format). The corrupted frame is not recorded in FIFO_RX1.

BFO2: At least one corrupted frame has been received on port #2 (the reason can be a wrong CRC, a stuffing error, an unknown format). The corrupted frame is not recorded in FIFO_RX2.

OVF1: FIFO_RX1 has overflowed. At least one frame on port #1 has been lost.

OVF2: FIFO_RX2 has overflowed. At least one frame on port #2 has been lost.

TXNE: FIFO_TX is not empty.

VII.5 FRMRX_SIZE1

FRMRX_SIZE1	
7	0
R	
FRMRX_SIZE1[7..0]	
00h	

FRMRX_SIZE1: Gives the number of bytes of the frame being read in FIFO_RX1 remaining in the FIFO. If several frames are recorded in FIFO_RX1, the register gives the number of remaining byte of the first available frame. This register is valid and shall be interpreted only if bit FRM1 of STATUS register is set.

VII.6 FRMRX_SIZE2

FRMRX_SIZE2	
7	0
R	
FRMRX_SIZE2[7..0]	
00h	

FRMRX_SIZE2: Gives the number of bytes of the frame being read in FIFO_RX2 remaining in the FIFO. If several frames are recorded in FIFO_RX2, the register gives the number of remaining byte of the first available frame. This register is valid and shall be interpreted only if bit FRM2 of STATUS register is set.

VII.7 FRMTX_FREE

FRMTX_FREE	
7	0
R	
FRMTX_FREE[7..0]	
00h	

FRMTX_FREE: Gives the number of free bytes in the FIFO_TX (number of bytes that can be written without an overflow condition).

VII.8 FIFO_RX1

FIFO_RX1	
7	0
R	
FIFO_RX1[7..0]	
00h	

FIFO_RX1: Access to application data of frames received on port #1. The register is of type FIFO: reading N bytes from this register returns the N first bytes of the port #1 reception FIFO.

Note: Never read more than available data in the FIFO (given by FRMRX_SIZE1) while it may corrupt the following frame.

Note: The byte stream available at this register is exactly the application layer of the SATURN protocol as defined in [R2].

VII.9 FIFO_RX2

FIFO_RX2	
7	0
R	
FIFO_RX2[7..0]	
00h	

FIFO_RX2: Access to application data of frames received on port #2. The register is of type FIFO: reading N bytes from this register returns the N first bytes of the port #2 reception FIFO.

Note: Never read more than available data in the FIFO (given by FRMRX_SIZE2) while it may corrupt the following frame.

Note: The byte stream available at this register is exactly the application layer of the SATURN protocol as defined in [R2].

VII.10 FIFO_TX

FIFO_TX	
7	0
W	
FIFO_TX[7..0]	
00h	

FIFO_TX: Byte stream that compose the application layer of frames to be emitted by the module. The register is of type FIFO: writing N bytes in this register, write the N last bytes in the FIFO_TX.

Note 1: Never write more than available free space in the FIFO given by register FRMTX_FREE.

Note 2: Byte stream to be written in this register complies with the application layer of the SATURN protocol as defined in [R2], but precede by:

- The size in byte +1 of the frame to be emitted
- The TID of the frame target agent.

Once application layer of a frame has been written in the FIFO_TX, application software shall activate the effective emission of the frame by asserting the bit STRTX of CONTROL register. STRTX bit activation is possible only if it was previously '0'. If STRTX is not cleared, application software shall wait it goes low.

VII.11 VERSION

VERSION			
7	4	3	0
R		R	
MAJ[3..0]		MIN[3..0]	
-		-	

Gives the FPGA firmware version. Format is Majeur.Mineur.

MAJ: FPGA firmware major revision (from 1 to 15)

MIN: FPGA firmware minor revision (from 0 to 15)

VII.12 TRAFIC

TRAFIC							
7	6	5	4	3	2	1	0
						R	R
RFU						TRF2	TRF1
						0	0

TRF1: '1' indicates that at least one valid frame has been seen on the network port #1, during previous cycle, even if the module was not the target. '0' indicates that no valid frame has been seen on port #1 during previous cycle.

TRF2: '1' indicates that at least one valid frame has been seen on the network port #2, during previous cycle, even if the module was not the target. '0' indicates that no valid frame has been seen on port #2 during previous cycle.