ESA R&D n° 20167/06/NL/FM

Further Development of the Spacecraft Controller on a Chip

Specification and Architecture of the SPW-RMAP module

	Name and Function	Date	Signature
Prepared by	**Tam LE NGOC** *Digital Design Engineer*	22/02/08	
Verified by	**Marc SOUYRI** *ASIC & FPGA expert*	22/02/2008	
Authorized by	**Franck KOEBEL** *SCOC3's project manager*	22/02/08	

Document type	Nb WBS	Keywords

DOCUMENT CHANGE LOG

Issue/ Revision	Date	Modification Nb	Modified pages	Observations
00	2007/04/18			
01	2007/04/30	14	20,21	4590 and 4705 : TX_DESC_ADD and RX_DESC_ADD shall not be modified during a transfer.
			27	4830: only current memory area is invalidated.
			33	Details on R/W, R, W and Reserved fields behaviour added.
			7,35,64, 65	Introduction of the fact the SW reset shall also last at least 3 CLK_TX cycles. 4030 and 4040 modified. RST description modified in CTRLREG description. Reset management figure updated. Paragraph 8.3.1 added, with a new figure describing Reset module.
			21,37,38, 39	ROLLOVER IT added. Modification of 4705. Modification of ITPFREG, ITMREG and ITCREG descriptions.
			65	Host interface architecture figure updated.
			16	RMW characteristics modifications. Byte, half-word and word RMW are accepted if memory aligned.
02	2007/05/23	1	20	4580 : Read replies status is always OK
03	2007/05/23		All	The specifications modified are: 4240, 4245, 4410, 4730, 4780, 4830, 4840 and 4945. Adding specifications 4841 and 4856. The CUR_BUF_END description in table 33 is more accurate.
04	2007/10/02		All	Modification of the registers. Adding 4222 and 4223.
05	2007/11/06		All	Specification modified according to the new interface: 4240, 4245, 4580, 4710, 4730, 4765, 4770, 4775, 4780, 4820, 4830, 4835, 4865, 4890, 4895, 4900, 4905. The specification 4840 is deleted and the 4901 is created.

PAGE ISSUE RECORD

Issue of this document comprises the following pages at the issue shown

Page	Issue/ Rev.	Page	Issue/ Rev.	Page	Issue/ Rev.	Page	Issue/ Rev.	Page	Issue/ Rev.	Page	Issue/ Rev.
1-76	00/00										
All	00/01										
All	00/02										
All	00/03										
All	00/04										
All	00/05										

TABLE OF CONTENTS

1 INTRODUCTION

This document is the specification and architecture of the SPW-RMAP module. This module is part of the SCoC3 design. This document is written in the frame of the ESA R&D "Further development of the Spacecraft Controller on a Chip" reference 20167/06/NL/FM.

Requirements are specified throughout this document in table format as follows:

Id	Requirement Text	Verification Method	Upper Links

- The absolute requirement identifier (Id), defined on 4 digits. The first digit corresponds to the current chapter number.
- The requirement text. If tables are considered as part of requirement they are referenced clearly in the text and inserted after and separated from the requirement table and are managed as free text attached to the identifier requirement.
- The verification method (Verif method) as one of the following : R/review of design; I/inspection; A/analysis; T/test (the paragraph given is from the simulation plan document), S/similarity, D/definition (not be tracked)
- The trace to the upper level requirements (Upper Links), shall be managed with the following format:
  - **AAAANNNN** where AAAA is a label associated to the upper document and NNNN the requirement identifier of this upper level.
  - Or **CREATED** key word if the requirement has no link with upper level

All document elements, which are not presented in the table format explained above are not requirements and will not be verified or tracked.

The SPW-RMAP module performs the Spacewire RMAP function.

The Spacewire is a serial high speed link compliant with the ECSS-E-50-12A specification [AD-10].

The RMAP is a protocol built on a spacewire network, compliant with the ECSS-E-50-11 Draft F specification [AD-12].

For the SCOC3 project, the SPW-RMAP block also contains AHB and APB interfaces.

It represents an evolution of the previous ESA/ASTRIUM Spacewire IP Core [AD-11].

It implements improvements of this IP, and the integration of RMAP protocol [AD-12].

The document contains the following sections:

- Applicable and reference documents followed by the list of the acronyms
- A general description of the module in its environment, and of its content
- A detailed description of the module functionality. This section starts by the description of the operating modes of the module. Then all the functions are described
- A description of the registers
- A description of the interfaces
- A detailed description of the architecture

2 DOCUMENTS AND ACRONYMS

2.1 APPLICABLE DOCUMENTS

AD-1	ECSS Q60-02: Final Draft ASIC/FPGA Development Standard, http://www.estec.esa.nl/microelectronics/asic/Final-Draft-ECSS-Q-60-02.pdf
AD-2:	VHDL Modelling Guidelines, ftp://ftp.estec.esa.nl/pub/vhdl/doc/ModelGuide.pdf
AD-3	ESA R&D " Further Development of the Spacecraft Controller on a Chip ".Statement of Work
AD-4	Telemetry Channel Coding Standard, ESA PSS-04-103, Issue 1, September 1989
AD-5	Telemetry Channel Coding, CCSDS 101.0-B-3, May 1992
AD-6	Packet Telemetry Standard, ESA PSS-04-106, Issue 1, January 1988
AD-7	Packet Telemetry, CCSDS 102.0-B-4, November 1995
AD-8	Packet Telecommand Standard, ESA PSS-04-107, Issue 2, April 1992
AD-9	Advanced Microcontroller Bus Architecture (AMBA™) Specification, revision 2.0, ARM IHI 0011A
AD-10	ESA Spacewire Standard – ECSS-E-50-12A, January 2003
AD-11	Spacewire IP Core – Specification and Architecture, June 2003
AD-12	Remote Memory Access Protocol – ECSS-E-50-11 Draft F, December 2006

2.2 REFERENCE DOCUMENTS

RD-1	Technical Documentation from Call-Off Order #3 (Spacecraft Controller On-a Chip) of ESA contract #13345/99/NL/FM (Building Blocks for System On-a Chip), known to both parties
RD-2:	Contract #13345/99/NL/FM (Building Blocks for System On-a Chip)
RD-3	LEON3 and GRLIB Documentation
RD-4	Synthesisable IP cores available from ESA
RD-5	ASIC Design and Manufacturing Requirements, ESA document WDN/PS/700
RD-6	Minutes of informal meeting R&D.SOC.MN.00395.V.ASTR from 15. December 2005, known to both parties

2.3 ACRONYMS

AD	Applicable Document
ADR	Architectural Design Review
AIT	Assembly Integration and Test
ASIC	Application Specific Integrated Circuit
ASIM	Application Specific Integrated Microsystem
ASSP	Application Specific Standard Product
BLADE	Board for LEON and Avionics Development
CDR	Critical Design Review
CPU	Central Processor Unit
DDR	Detailed Design Review
DFF	D-Type Flip Flop
DRC	Design Rule Check
DSP	Digital Signal Processor
EDAC	Error Detection And Correction
EDA	Electronic Design Automation
EGSE	Electrical Ground Support Equipment
ESA	European Space Agency
ESTEC	European Space Research and Technology Centre
FDIR	Failure Detection Isolation and Recovery
FPGA	Field Programmable Gate Array
GEO	Geosynchronous Equatorial Orbit
GRLIB	Gaisler Research Library,
HDL	Hardware Description Language
I/O	Input/Output
ID	Identification
IDR	Initial Design Review
IEEE	Institute of Electrical and Electronics Engineers
IP, IPR	Intellectual Property, Intellectual Property Rights
IPMON	Performance Monitoring (IP block)
ITT	Invitation To Tender
JTAG	Joint Test Action Group (refer to IEEE std 1149.1)
LEO	Low Earth Orbit
LET	Linear Energy Transfer
OBDH	On Board Data Handling
OBMU	On Board Management Unit
PCB	Printed Circuit Board
PDF	Portable Document Format
PDR	Preliminary Design Review
PM	Performance Monitoring (in fact called IPMON)
PM	Processor Module
RD	Reference Document
RTEMS	
RTOS	Real Time Operating System (example: RTEMS)
SOC	System On a Chip
SCoC	Spacecraft Controller on a Chip
SEE	Single Event Effect (or SEP Single Event Phenomena)
SEL	Single Event Latch up
SEP	see SEE
SET	Single Event Transient
SEU	Single Event Upset
SRAM	Static Random Access Memory
SRR	Specification Requirement Review
TC	TeleCommand
TID	Total Integrated Dose
TM	TeleMetry
TRP	Technological Research Programme
VHDL	VHSIC Hardware Description Language,
VLSI	Very Large Scale Integration
WP	Work Package
WWW	World Wide Web

3 GENERAL DESCRIPTION OF THE MODULE

3000	The SPW-RMAP shall be a high-speed serial link to transmit and receive packets of data [AD-10].	D	CREATED

3010	The SPW-RMAP shall support RMAP protocol [AD-12].	4.13	CREATED

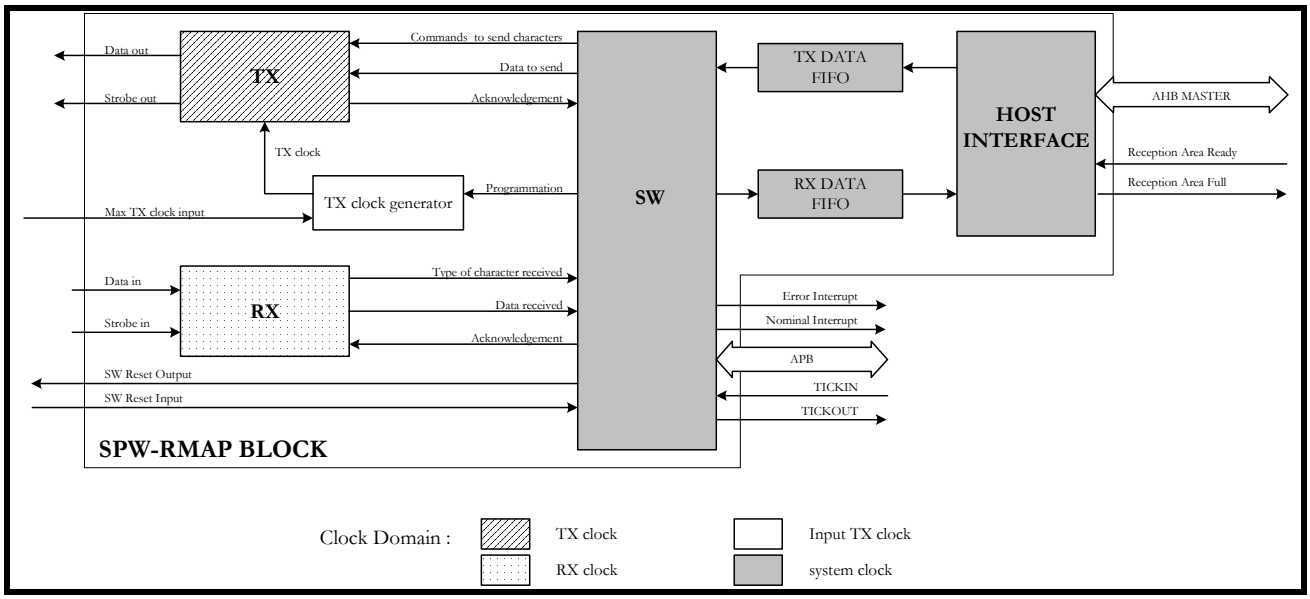


Figure 1 : SPW-RMAP General Architecture

3020	The SPW-RMAP shall implement general architecture given in Figure 1.	D	CREATED

3030	The Host Interface block shall be an interface with the AMBA AHB and APB buses. It shall contain the management of the data sent by the host application. It shall manage the storage of data into the host memory. All RMAP specificities shall be implemented in this block.	4.4 4.5 4.7 4.13	CREATED

3040	The TX Data FIFO block shall be a 9-bits FIFO containing the data to be transmitted.	D	CREATED

3050	The RX Data FIFO block shall be a 9-bits FIFO containing the data to be stored into the host memory.	D	CREATED

3060	The SW block shall manage the initialisation protocol. This block shall select the character to be transmitted and check any error occurrence.	4.12	CREATED

3070	The TX block shall send the character at the transmission frequency.	I	CREATED

3080	The RX block shall identify the received character type.	4.12 4.4.1	CREATED

3090	The TX clock generator block shall generate the clock transmission rate.	D	CREATED

4 DETAILED DESCRIPTION OF THE MODULE FUNCTIONALITY

4.1 OPERATING MODES

4000	The SPW-RMAP shall support the following operating modes:
	• RESET mode (resetn=0):
	  o TX and RX blocks shall be inactive
	  o Host interface shall be inactive
	• ACTIVE mode (resetn=1):
	  o TX block shall be inactive and RX block shall be active (when entering the ACTIVE mode)
	  o Host interface shall always be on
	The transition of the TX and RX states in the active mode shall be consistent of the link initialisation protocol described in [AD-10].
		D	CREATED

4.2 RESET MANAGEMENT

4010	The SPW-RMAP shall have a RESETN input, which shall place the whole IP in RESET mode, when set to 0.	D	CREATED

4020	The SPW-RMAP shall have a RST bit in CTRLREG register allowing the host application to perform a SW synchronous reset of the IP.	D	CREATED

4030	The activation of SW synchronous reset by the host application shall lead to a reset of the complete IP during at least 10 system cycles and 3 TX cycles, except for the following configuration registers which are only reset by the hardware reset : • CONFREG : all fields • CTRLREG : fields LINK_DISABLED, LINK_AUTOSTART. • NAKPIDREG : all fields • TXPKTSIZEREG: field TX_PKT_SIZE • RXPKTSIZEREG: field RX_PKT_SIZE • TIMOUTREG : all fields • START_AREA1	4.1	CREATED

	• START_AREA2 • AREA_SIZE		

Automatic connection attempts should be performed after a SW synchronous reset, if LINK_START, LINK_AUTOSTART or LINK_DISABLE were correctly configured before SW synchronous reset. If not a hardware reset should be necessary.

4040	During this period of reset, the SPW-RMAP shall activate its SPWRSTO output.	4.1	CREATED

4050	The activation of SPWRSTI input shall lead to a SW synchronous reset of the IP, without cycles count. The SPWRSTI activation shall not lead to a SPWRSTO activation nor RST bit activation.	4.1	CREATED

4.3 SPACEWIRE CORE SPECIFICATION

Only SpaceWire implementation specific characteristics are detailed hereafter.

Refer to Spacewire standard [AD-10] for detailed description.

4.3.1 Link initialization

4100	The link initialization protocol presented in [AD-10] chapter 8.5 shall be implemented.	4.12	CREATED

4.3.2 Transmission function

4105	The Spacewire Core transmission function shall receive 9-bit data from the TX data FIFO. The 9-bit data shall be composed of 8 bits of real data and 1 bit for particular character such as EOP and EEP (refer to [AD-10]).	D	CREATED

4110	The SPW-RMAP shall fetch 9-bit data in the TX data FIFO and send it to the TX module with the right command to transmit this data only when the credit counter is positive (refer to [AD-10]).	D	CREATED

4115	When the RX data FIFO free space allows the reception of 8 more bytes, the SW module shall generate an order to transmit a FCT.	D	CREATED

4120	To transmit a time code, the TICKIN signal shall be activated so that the SW module generates the right transfer.	4.11	CREATED

4125	When the TX module receives a command from the SW module, an acknowledgement shall be generated. Then the character corresponding to the command shall be transmitted through the LVDS link (Data and Strobe outputs).	D	CREATED

4130	The TX module shall automatically transmit NULL characters (refer to [AD-10]) when no other transmission is requested.	4.12	CREATED

4.3.2.1 TX clock programming

4135	The transmission frequency shall be programmable through the APB interface. The TX clock generator shall create the required TX frequency, which shall be **up to 4 times** the system clock frequency.	4.6 4.8	CREATED

4140	The FREQ_INIT field of CONFREG register shall configure the frequency in the initialisation state.	D	CREATED

4145	The FREQ_RUN field of CONFREG register shall configure the frequency in the run state.	D	CREATED

4150	For the gated TX clock configuration, when the TX_MAX_EN bit is asserted in CONFREG register, the FREQ_RUN value shall not be taken into account and the transmission frequency shall be equal to the input TX clock frequency.	D	CREATED

It is recommended not to change the FREQ_INIT and FREQ_RUN values when the spacewire link is in the run state.

4.3.3 Reception function

4155	The RX module shall perform the recognition of the received character type.	D	CREATED

4160	The RX clock shall be built from the data and strobe input signals (refer to [AD-10]).	D	CREATED

4165	The RX module shall indicate the received characters to the SW module. Each time that information of character type is received from the RX module, the SW module shall generate an acknowledgement. Then, following the received character, the SW module shall manage the credit counter and the outstanding counter.	D	CREATED

4170	A 9-bit word shall be stored into the RX data FIFO when a data is received.	D	CREATED

4175	The SW module shall activate the TICKOUT_CTM signal when a right time code is received.	4.11	CREATED

4.3.4 Format of the words stored in the TX and RX FIFOs:

4180	The TX and RX FIFOs shall contain 9-bit words compliant with Figure 2 and Table 1 descriptions. Refer to [AD-10] for details.	D	CREATED

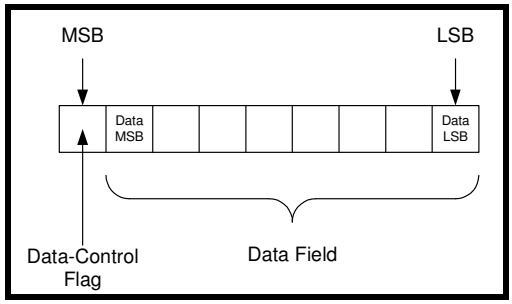


Figure 2 : FIFO word format

	Control flag	Data Bits(MSB…LSB)	Meaning
TX FIFO	0	XXXXXXXX	8-bit data
	1	XXXXXXX0	EOP
	1	XXXXXXX1	EEP
RX FIFO	0	XXXXXXXX	8-bit data
	1	00000000	EOP
	1	00000001	EEP

Table 1 : Word meaning

4.3.5 Link Timing specification

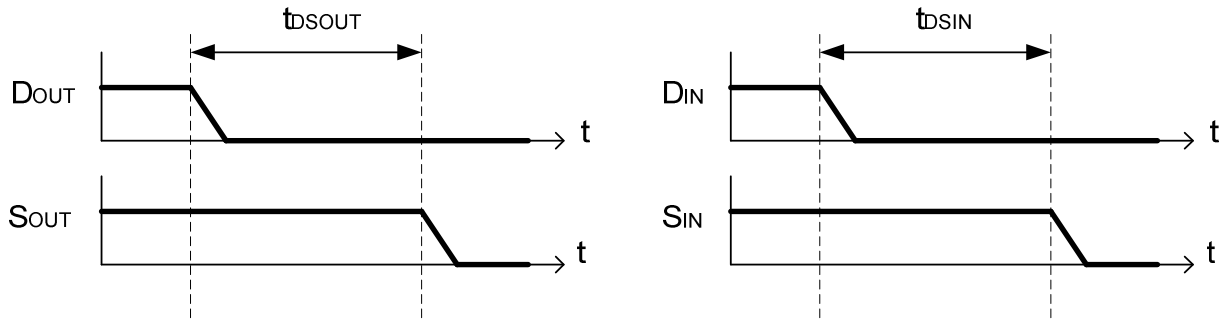


Figure 3: Timing characteristics

tDSOUT = 1 / Fop +/- 0,5 ns with Fop until 160 MHz. This constraint includes skew and jitter.

	Min	Max
tDSOUT	1 / Fop - 0,5 ns	1 / Fop + 0,5 ns
tDSIN	2 ns	-

tDSIN should be great or equal to 2 ns to have a correct generated reception clock.

4.4 HOST INTERFACE SPECIFICATION

4.4.1 Functional modes

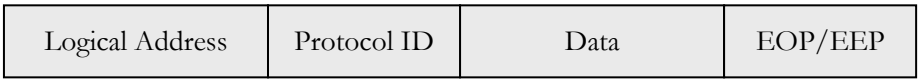
4200	The SPW-RMAP shall have 3 functional modes • Data-only packets mode [DATA MODE] • RMAP packet mode [RMAP MODE] • Data-only and RMAP packets accepted [MIXED MODE]	D	CREATED

4205	The choice of the SPW-RMAP mode shall depend on the value of PKT_MODE field of CONFREG register : • PKT_MODE = 1 -> RMAP_MODE • PKT_MODE = 2 -> MIXED_MODE • PKT_MODE = 0 or 3 -> DATA_MODE	D	CREATED

4210	When RMAP packets are handled and RM_SLV_DIS field of CONFREG register is set to 1, SPW-RMAP shall reject all received RMAP commands. If needed it shall acknowledge the command with an RMAP error reply.	4.13.1.8 4.13.2.5 4.13.3.6	CREATED

4.4.2 Packets structure

4215	In DATA MODE, the packets shall have the following structure :  No protocol shall be supported in this mode.	D	CREATED

4220	In RMAP MODE and MIXED MODE, the packets shall have the following structure :  All packets shall start by a Logical Address byte, followed by a Protocol ID byte.	D	CREATED

4222	The SPW-RMAP considers the byte before the EOP as the data CRC when it needs to check the data CRC of the received packet. So this byte is not included in the number of received data.	4.13.1.12	CREATED

4223	When the SPW-RMAP receives an EEP, the byte before the EEP is not considered as a CRC.	4.13.1.12	CREATED

4225	In RMAP and MIXED MODE, all received packets shall have a correct logical address byte, corresponding to the NODE_ADDRESS field of NAKPIDREG register, to be taken into account. Packets with another logical address shall be discarded and no interrupt shall be generated.	4.13.1.8 4.13.2.5 4.13.3.6	CREATED

4230	In RMAP MODE, all packets received shall have an RMAP protocol ID to be taken into account (protocol ID = 0x01). Packets with another protocol ID shall be discarded and a PID_ERR interrupt shall be sent to the host application.	4.13.1.8 4.13.2.5 4.13.3.6	CREATED

4235	In MIXED MODE, all RMAP received packets shall have a RMAP protocol ID to be taken into account (protocol ID = 0x01) and all data-only packets shall have a protocol ID corresponding to the DATA_PID field of NAKPIDREG register to be taken into account. Packets with another protocol ID shall be discarded and a PID_ERR interrupt shall be sent to the host application.	4.15.2	CREATED

4240	In all modes, all transmitted packets shall have a maximum length equal to TX_PKT_SIZE field of TXPKTSIZEREG register. TX_PKT_SIZE is the number of the bytes in the packet. The CRC bytes and the RMAP Header are not included in the TX_PKT_SIZE. Packets to transmit with a length exceeding this parameter shall not be sent : the segment causing the length to exceed TX_PKT_SIZE shall be truncated and the packet shall end with an EEP. TX_PKT_SIZE_ERR interrupt shall be sent to the host application.	4.4.4	CREATED

4245	The received packet in all modes shall have a maximum length equal to the RX_PKT_SIZE field of RXPKTSIZEREG register. The RMAP header are not included in the RX_PKT_SIZE count The RMAP packet shall have the data field length in the header equal to the data transmitted. The RX_PKT_SIZE_ERR interrupt is sent to the host application if not. If the RX host interface detects a packet exceeding this length, the storage shall be stopped and an error reply shall be sent if the packet was an RMAP command with acknowledge. RX_PKT_SIZE_ERR interrupt shall be sent to the host application.	4.7.7	CREATED

4.4.3 RMAP characteristics

Only RMAP implementation specific characteristics are detailed hereafter.

Refer to RMAP standard [AD-12] for detailed description.

4.4.3.1 General specification points

4300	Write, Read and Read-Modify-Write [RMW] commands shall be implemented.	4.13	CREATED

4305	Logical addressing shall be used and path addressing shall not be supported.	D	CREATED

4310	Extended protocol ID shall not be supported.	D	CREATED

4315	Destination key shall correspond to DESTINATION_KEY field of NAKPIDREG register. If the destination key has a wrong value, the command shall not be executed and an error reply shall be returned to the source node. No interrupt shall be raised.	4.13.1.8 4.13.2.5 4.13.3.6	CREATED

4320	Extended address shall not be supported.	D	CREATED

4325	The correlation between RMAP sent commands and received replies shall be performed by the host application. In particular, no hardware timeout mechanism shall be implemented.	D	CREATED

4.4.3.2 Write command conformance

Write Command			
Action	Supported / Not Supported	Maximum data length (bytes)	Non-aligned access accepted
8-bit write	S	1	-
16-bit write	S	2	Yes
32 bit write	S	16 M	Yes
64-bit write	NS	-	-
Verified write	S	4	No
Word or byte address	Unaligned accesses accepted		
Accepted logical address	0xFE at power-on Value of NODE_ADDRESS after initialisation		
Accepted destination keys	0x00 at power-on Value of DESTINATION_KEY after initialisation		
Accepted address ranges	0x00 0000 0000 -- 0x00 FFFF FFFF		
Address incrementation	Incrementing and not incrementing address are implemented		

Table 2 : Write command equipment characteristics

4330	Write commands shall be compliant to Table 2. Verified and non-verified writes shall be supported. Incrementing and non-incrementing writes shall be supported. Acknowledged and non-acknowledged writes shall be supported.	 4.13.1 4.13.1.9 4.13.1	CREATED

4335	Verified writes with data length from 1 up to 4 bytes shall be supported (3 bytes length shall not be supported). The address shall be aligned to the size : • 1B verified writes shall be done to any address • 2B verified writes shall be half-word aligned • 3B verified writes shall not be supported • 4B verified writes shall be word aligned.	4.13.1.1	CREATED

4340	Non-verified and non-incrementing writes shall have a data length multiple of 4B, and the address shall be word aligned. The data length shall range from 4B to 16 MB, if the resulting packet size doesn't exceed the TX_PKT_SIZE or RX_PKT_SIZE constraints.	4.13.1.9	CREATED

4345	Non-verified and incrementing writes shall have a data length from 1 to 16 MB, if it doesn't exceed the TX_PKT_SIZE or RX_PKT_SIZE constraints. Non-aligned addresses shall be supported.	4.13.1	CREATED

4.4.3.3 Read command conformance

Read Command			
Action	Supported / Not Supported	Maximum data length (bytes)	Non-aligned access accepted
8-bit read	S	1	-
16-bit read	S	2	Yes
32 bit read	S	16 M	Yes
64-bit read	NS	-	-
Word or byte address	Unaligned accesses accepted		
Accepted logical address	0xFE at power-on Value of NODE_ADDRESS after initialisation		
Accepted destination keys	0x00 at power-on Value of DESTINATION_KEY after initialisation		
Accepted address ranges	0x00 0000 0000 -- 0x00 FFFF FFFF		
Address incrementation	Incrementing and not incrementing address are implemented		

Table 3 : Read command equipment characteristics

4350	Read commands shall be compliant to Table 3. Incrementing and non-incrementing reads shall be supported.	D	CREATED

4355	Non-incrementing reads shall have a data length multiple of 4B, and the address shall be word aligned. The data length shall range from 4B to 16 MB, if it doesn't exceed the TX_PKT_SIZE or RX_PKT_SIZE constraints.	4.13.2.6	CREATED

4360	Incrementing reads shall have a data length from 1 to 16 MB, if it doesn't exceed the TX_PKT_SIZE or RX_PKT_SIZE constraints. Non-aligned addresses shall be supported.	4.13.2	CREATED

4.4.3.4 Read-modify-Write command conformance

Read-Modify-Write Command			
Action	Supported / Not Supported	Maximum data length (bytes)	Non-aligned access accepted
8-bit read-modify-write	S	1	-
16-bit read-modify-write	S	2	No
32 bit read-modify-write	S	4	No
64-bit read-modify-write	NS	-	-
Word or byte address	Shall be aligned to the size.		
Accepted logical address	0xFE at power-on Value of NODE_ADDRESS after initialisation		
Accepted destination keys	0x00 at power-on Value of DESTINATION_KEY after initialisation		
Accepted address ranges	0x00 0000 0000 -- 0x00 FFFF FFFF		

Table 4 : Read-Modify-Write command equipment characteristics

4365	Read-Modify-Write commands shall be compliant to Table 4.	D	CREATED

4370	Read-Modify-Writes with data length from 1 up to 4 bytes shall be supported (3 bytes length shall not be supported). The address shall be aligned to the size : • 1B Read-Modify-Writes shall be done to any address • 2B Read-Modify-Writes shall be half-word aligned • 3B Read-Modify-Writes shall not be supported • 4B Read-Modify-Writes shall be word aligned.	4.13.3.1	CREATED

4.4.4 AHB interface

4400	An AHB master interface shall be implemented, it shall be used for RX or TX DMA transfers. These transfers shall be exclusive.	D	CREATED
4405	Depending on address alignment and packet length, byte, half-word or word single accesses shall be performed on AHB bus.	4.10	CREATED
4410	If MEM_ALIGNEMENT field of CONFREG register is set to 1, packets written into host memory shall start on 4B-aligned addresses. Hence accesses performed shall only be word, half-word or byte accesses.	4.7.8	CREATED
4415	If enough data has to be transferred and alignment allows it, bursts of 4 words shall be performed on AHB bus. Bursts shall not cross 1KB boundaries.	4.10.1.5	CREATED
4420	Activation of BURST_DIS field of the BURSTDISREG register shall prevent AHB bursts to be performed.	4.16.5	CREATED
4425	When AHB responses are not OK, the transmission shall be aborted and an EEP added. The remaining packets of the linked list are not sent. The storage of received data is not stopped when an AHB error occurs. The AHB error is reported by the AMBA_ERR interrupt. If needed an RMAP error reply shall be sent with the status 10.	4.5 4.9	CREATED
4430	AHB bus shall be locked when performing read-modify-write accesses.	I	CREATED

4.4.5 APB interface

4435	An APB slave interface shall be implemented, it shall be used to access SPW-RMAP registers. Refer to paragraph 5 for a complete description of registers.	D	CREATED

4.4.6 TX host interface [THI]

4500	In all modes, the THI shall transmit a linked list of packets, each packet being a linked list of segments.	D	CREATED

4505	The segments of data shall be described by a linked list of segment descriptors, written in memory by the host application.	D	CREATED

4510	The host application shall be responsible for the consistency of the linked list described, and of its compliance with the mode chosen. The host application shall be responsible for the construction of headers needed by the SPW-RMAP IP, depending on the mode chosen.	D	CREATED

4.4.6.1 Segment descriptors

4515	A segment descriptor shall be made of 3 words, aligned on a 4 words boundary in memory : • Control word, refer to Table 5. • Segment address pointer, refer to Table 6. • Next linked segment descriptor pointer, refer to Table 7.	4.13.1.10	CREATED

31	30	29	28				24	23																						0
RMH	LS	IE	NCRC					SEGMENT LENGTH																						

RMH : RMAP Header

LS : Last Segment

IE : Interrupt Enable

NCRC : Non CRC bytes. It indicates the number of bytes not to take account in the CRC computation. It is used for a path address header segment. It is not needed for a logical address header segment.

Table 5 : Control word

31																												0
SEGMENT ADDRESS																												

Table 6 : Segment address pointer

31																												0
NEXT LINKED SEGMENT DESCRIPTOR ADDRESS																												

Table 7 : Next linked segment descriptor pointer

4520	The RMAP Header configuration bit shall be only set in the first segment descriptor of a RMAP packet. Non RMAP packets shall not have header segments.	D	CREATED

4525	The non-CRC bytes value shall be taken into account only in RMAP header segments. It shall represent the number of RMAP header bytes (at the beginning of the segment) not to be taken into account in the CRC computation. It shall be lesser than the RMAP header length.	4.13.1.10	CREATED

4530	The Last Segment configuration bit [LS] of the control word shall be set to 1 in the last segment of a packet.	D	CREATED

4535	Only segment length 8 LSB shall be taken into account in RMAP header segments. It means that RMAP header length shall be up to 255 bytes.	D	CREATED

4540	A null value in Next linked segment descriptor pointer shall indicate the end of the linked packets list.	D	CREATED

4.4.6.2 CRC computation

4545	RMAP header and data CRC bytes shall be computed by THI hardware. It shall be compliant to the CRC implementation provided in RMAP standard (refer to [AD-12]).	4.13	CREATED

4550	RMAP header CRC scope shall be all RMAP header bytes after the non-CRC bytes specified in the descriptor of the RMAP header segment.	4.13.1.10	CREATED

4555	Data CRC scope shall be all non RMAP header segments bytes of a packet.	D	CREATED

4560	The choice of appending or not a CRC byte at the end of a data-only packet shall depend of DATA_CRC field of CONFREG register : a CRC shall be used if DATA_CRC = 1.	4.4.5	CREATED

For RMAP packets, CRC insertion is mandatory.

4.4.6.3 RMAP replies management

In this situation, the SPW-RMAP TX host interface is seen as a RMAP slave.

4565	A RMAP reply shall be transmitted by THI when needed by RX host interface. It shall wait the end of running packet transmission. Then it shall have priority over new packets to be transmitted. THI shall be responsible for the RMAP reply header and the data CRCs computing and insertion.	4.13.1.4 4.13.1.5 4.13.2 4.13.3	CREATED

4570	THI shall transmit write and read-modify-write replies without performing any AHB access.	D	CREATED

4575	THI shall perform AHB read accesses needed to complete RMAP read commands.	4.13.2	CREATED

4580	Read data shall not be bufferized, hence the reply status and the data length provided in the reply header can differ from the contents of the data part of the packet, if an error occurred during the read. Reply status shall be considered as OK, if no other error occurred. Reply data length shall be the data length requested in the read command.	4.5.4	CREATED

4585	Once the reply is completely written in the TX FIFO, the THI shall signal to RX host interface the transmission of the requested reply.	D	CREATED

4.4.6.4 Transfer management

4590	The segments linked list transfer shall begin when the address of the first descriptor is written in TXDESCREG register.	4.4	CREATED

It is recommended not to modify TX_DESC_ADD when in run state.

4595	THI shall read segment descriptors and shall perform the transfer of the segments, with the CRC computation adapted.	4.4	CREATED

4600	THI shall append EOP at the end of the last segment of a packet.	4.4	CREATED

4605	An interrupt shall be generated by the THI when a segment with Interrupt Enable [IE] configuration bit set to 1 is transmitted.	4.14	CREATED

4610	As long as its credit counter allows it, the THI shall send data. It shall not wait for a RMAP reply after a command is transmitted.	D	CREATED

4615	The TXDESCREG register shall contain the address of the last read segment descriptor.	D	CREATED

4620	The transfer shall be aborted if the ABORT_PACKET bit of CTRLREG register is set. An EEP shall be added to the data in the TX FIFO. The linked list of segment descriptors shall be reset.	4.4.3	CREATED

4.4.7 RX host interface [RHI]

4.4.7.1 RX packet descriptors

4700	A RX packet descriptor shall be written for the following received packets : • DATA MODE : all received packets • RMAP MODE : all received replies and all write commands of more than 32 bits • MIXED MODE : all data-only packets, all RMAP received replies and all RMAP write commands of more than 32 bits.	4.7.1 4.13.1 4.13.2 4.13.3 4.15.1	CREATED

4705	RX packets descriptors shall be written in memory in an area starting at RX_DESC_ADD and ending at RX_DESC_END. The area shall be wrapped around without any host application intervention, and ROLLOVER IT raised. RX_DESC_ADD shall reflect the next RX packet descriptor address.	4.14.3 4.7.1	CREATED

It is recommended not to modify RX_DESC_ADD when in run state.

4710	A RX packet descriptor shall be made of 4 words, aligned on a 4 words boundary in memory : • Control word, refer to Table 8. • Packet address pointer, refer to Table 9. • Reply word 1, refer to Table 10. • Reply word 2, refer to Table 11.	D	CREATED

31	30	29	28	27	26	25	24	23																							0
	RM	R	AE		TR	EP	DC	PACKET LENGTH																							

RM : RMAP packet

R : Reply packet

AE : AMBA error

EP : EEP termination

TR : Truncated

DC : Data CRC error

Table 8 : Control word

31																												0
PACKET ADDRESS																												

Table 9 : Packet address pointer

31							24	23							16	15							8	7							0
SOURCE LOGICAL ADDRESS								PROTOCOL IDENTIFIER								PACKET TYPE, COMMAND…								STATUS							

Table 10 : Reply word 1

31							24	23							16	15							8	7							0
DEST LOGICAL ADDRESS								TRANSACTION ID (MS)								TRANSACTION ID (LS)								0	0	0	0	0	0	0	0

Table 11 : Reply word 2

4715	RMAP packet bit [RM] shall only be activated for received RMAP packets.	4.13	CREATED

4720	Reply packet bit [R] shall only be activated for received RMAP reply packets.	4.13.2.3	CREATED

4725	AMBA error bit [AE] shall be activated for received packets that had been partially written in memory and interrupted by an AHB error. Such packets shall be considered as incomplete, and CRC unchecked. It shall be the responsibility of the host application to find next packet containing the remaining data and to concatenate it.	4.9.1	CREATED

4730	The RX packet descriptor is generated once the entire packet is stored.	4.7.2	CREATED

4735	Truncated bit [TR] shall be activated for packets exceeding RX_PKT_SIZE, or RMAP data exceeding data length announced in RMAP header (applicable to received replies and more than 32 bits write commands).	4.7.7	CREATED

4740	EEP termination bit [EP] shall be activated for packets terminated by an EEP. They shall be considered as CRC unchecked.	4.4.3	CREATED

4745	Data CRC error bit [DC] shall be activated for packets untruncated, not exceeding Memory area, with data CRC error. Packets with no data CRC	4.13.1.3	CREATED

	verification shall not activate this configuration bit.		

4750	PACKET LENGTH and PACKET ADDRESS shall describe written packet. In case of a RMAP write reply, RX packet descriptor fields PACKET LENGTH and PACKET ADDRESS shall be null.	4.13.1.4	CREATED

4755	Descriptors of received RMAP replies shall copy the first 56 bits of the reply header into RMAP reply words 1 and 2, followed by a null byte. For other received packet types, those registers shall be null. However, the RX packet descriptors shall be aligned on 4 words addresses, whatever the mode, to allow burst4 accesses on AHB bus.	4.13.1.4	CREATED

4.4.7.2 DATA MODE

4760	In DATA MODE, RHI shall write all received packets in flip/flop storage area. Then it shall write received packet descriptors associated in memory, at current RX descriptor address, specified in RXDESCREG register.	4.7	CREATED

4.4.7.2.1 Flip/flop storage mechanism

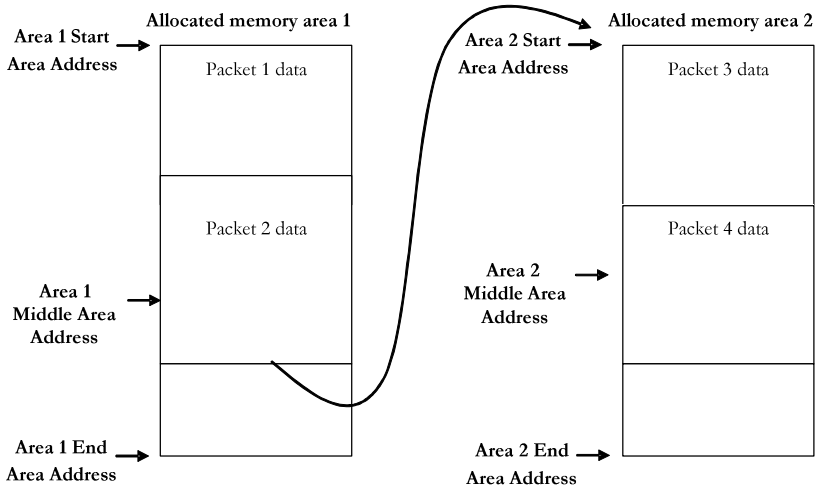


Figure 4 : Flip/flop storage format

4765	The host application shall allocate 2 memory areas (1 and 2) and configure 2 sets (one for each area) of three word aligned addresses in the SPW-RMAP. • The first address, called Start_Area Address, shall represent the beginning of (1 or 2) allocated area. It shall be set by the host application. The second address, called End_Area Address shall be the real end of the (1 or 2) allocated area. It is computed by the SPW-RMAP as the sum of Start_Area address and Area_Size length. • The last address, called Middle_Area Address is computed by the SPW-RMAP as the End_Area address minus RX_Packet_Size length.	4.7	CREATED

Thus the space between the Middle Area Address and the End Area Address is equal to the maximal size of a packet (expected to be received by the host application) in order to guarantee a safe protocol. Therefore the Area_Size length shall be bigger than the RX_Packet_Size length.

4770	The couple of start addresses shall be written in the SPW-RMAP configuration registers through the APB bus by the CPU. Likewise AREA_SIZE and RX_PACKET_SIZE shall be written in the SPW-RMAP configuration registers. In addition to these addresses, the host application shall provide a command indicating the validity of the areas.	4.7	CREATED

4775	Practically, the host application shall validate one memory area at least to enable the RX data transfer from the RX FIFO to the host memory. For this, the Start_Area, Area_Size and RX_Packet_Size Registers shall be programmed and the AREA1_VALID or/and AREA2_VALID bit(s) shall be set to validate the memory area(s).	4.7	CREATED

4780	The RHI shall indicate the current end of the currently used area (1 or 2), called Current Buffer End. • When the area is empty, the Current Buffer End shall be the Start Address of this area. • When an entire packet is stored into the used area, the Current Buffer End shall indicate the address of the next packet. • The Current Buffer End shall be updated while the storage of the RX packet is in progress.	4.7	CREATED

4785	The RHI shall indicate the used memory area (AREA1_USED or AREA2_USED bit in the management register) so that the host application can follow the storage progression by monitoring the Current Buffer End.	4.7	CREATED

4790	The RHI shall use the Start Address to start writing packets in host memory in a given area (1 or 2). The RHI shall first attempt to use area 1.	4.7	CREATED

4795	In case the RX receives data and no area is allocated, the SPW-RMAP shall generate the NO_AREA_VALID interrupt.	4.7	CREATED

4800	If the SPW-RMAP receives a NULL character and the link is not enabled, the LINK_NOT_ENABLED interrupt shall be generated to warn the host	4.14.2	CREATED

	application.		

4.4.7.2.2 Reaching end of packet

4805	The EOP, EEP and CRC shall not be written into the host memory.	4.7	CREATED
4810	When an EEP is detected in the RX FIFO, the RHI shall write into memory the descriptor of the RX packet it was writing, with a packet size corresponding to the number of bytes already written.	4.4.3	CREATED
4815	When an EOP is detected in the RX FIFO, the RHI shall check the CRC if DATA_CRC is set to 1. In this case, the byte received before the EOP shall be considered as a data CRC. Then it shall write into memory the descriptor of the RX packet that it had just finished to write.	4.4.5	CREATED

4.4.7.2.3 Reaching the Middle_Area Address

4820	If the RHI reaches the Middle_Area Address during a packet transfer, the current AREA1_USED or AREA2_USED bit shall be reset. The packet transfer shall be finished and EOP detection, CRC verification and RX descriptor storage shall be performed like specified above. The SPW-RMAP shall then invalidate the current memory area by resetting the AREA1_VALID or AREA2_VALID bit, and it shall assert the corresponding AREA1_FULL or AREA2_FULL output. The re-validation of the area or the definition of a new area shall be in charge of the host application, or the hardware flip/flop management mechanism.	4.7	CREATED
4825	If the other allocated area is valid, the RHI shall continue transferring the next packet into the other available allocated area. If the other area is not valid, the RHI shall generate the NO_AREA_VALID interrupt and shall not start the storage of the next packet until the host application provides another available area.	4.7.4 4.7.2	CREATED

4.4.7.2.4 Reaching the End_Area Address

4830	The RHI may reach the End_Area Address during a packet transfer but not exceed it because such a packet should have been truncated during its reception.	4.7.1	CREATED

4.4.7.2.5 AHB error occurrence

4835	When the RHI receives an AHB error response, the RHI shall finish writing the packet and the RX packet descriptor with the AE bit set. The AMBA_ERR interrupt shall be activated.	4.9	CREATED

4841	When an AMBA error occurs or when the size is bigger than the RX_PKT_SIZE value (max size defined in the RXPKTSIZEREG register), the RX FIFO data is cleared until next EOP or EEP. During this process, the SPILLING_RX_FIFO flag is asserted.	4.9 4.7.7	CREATED

4.4.7.2.6 Packet exceeding RX_PKT_SIZE

4845	If the length of the received packet exceeds RX_PKT_SIZE, the RHI shall write into memory the descriptor of the RX packet it was writing, with a packet size corresponding to the number of bytes already written. The remaining data shall be discarded.	4.7.7	CREATED

4.4.7.2.7 Hardware flip/flop management

4850	To allow hardware management of flip/flop area, RHI shall issue status signals AREA1_FULL and AREA2_FULL, as soon as these areas are full. RHI shall also manage input signals AREA1_RDY and AREA2_RDY, which shall have the same meaning as the configuration bits AREA1_VALID and AREA2_VALID.	4.7.4 4.7.1	CREATED

4855	Hardware management of flip/flop area shall be enabled only if HW_FF_MGT bit of CONFREG is set to 1. If set to 0, AREA1_RDY and AREA2_RDY inputs shall be ignored.	4.7.9	CREATED

4856	When the RX FIFO is being cleared, the SPILLING_RX_FIFO flag shall be asserted.	4.9.1	CREATED

4.4.7.3 RMAP MODE

4860	In RMAP MODE, RHI shall decode the RMAP header, and check logical address, protocol ID, destination key, the command or reply type and the header CRC.	4.13	CREATED

4865	In case of bad header CRC, the packet shall be discarded, and a CRC_ERR interrupt shall be raised.	4.13.1.2 4.13.2.2 4.13.3.2	CREATED

4.4.7.3.1 Reception of a RMAP command

In this situation, the SPW-RMAP RX host interface is seen as a RMAP slave.

4870	In case of bad logical address, unsupported command type, path addressing, bad destination key, the packet shall be discarded and an error reply shall be sent to the RMAP master via THI, when acknowledge is required.	4.13.1.8 4.13.2.5 4.13.3.6	CREATED

4875	All replies shall be prepared by RHI and sent to THI, in charge of transmitting it to the RMAP master. RHI shall wait for reply transmission acknowledge by THI to manage next packet.	4.13.1.4 4.13.2.1 4.13.3.1	CREATED

4880	Read replies shall be prepared and sent to THI, which shall perform the AHB read operation and send the reply to the RMAP master. Write and read-modify-write replies shall be prepared by RHI and sent to THI once the AHB transfer has been performed by RHI.	4.13.1 4.13.3	CREATED

4885	Read-modify-write and verified write commands data CRC shall be checked before writing anything into memory. If it happens to be bad, nothing shall be written into memory.	4.13.3.3 4.13.1.7	CREATED

4890	In case of bad data CRC, the reply shall specify this error case. CRC_ERR interrupt shall be raised.	4.13.1.3	CREATED

4895	In case of AHB error happening during a transfer, the operation shall be	4.9.5	CREATED

	stopped, the packet discarded, and the reply shall specify this error case. AMBA_ERR interrupt shall be raised.		

4900	In case of early EOP detection, the operation shall be stopped, the packet discarded, and the reply shall specify this error case. RX_PKT_SIZE_ERR interrupt shall be raised.	4.13.1.12	CREATED

4901	In case of EEP detection, the operation shall be stopped, the packet discarded, and the reply shall specify this error case. EEP_REC interrupt shall be raised.	4.13.1.12 4.16.1	CREATED

4905	In case of data exceeding specified length in the header or MAX_PKT_SIZE, the remaining data shall be discarded, and the reply shall specify this error case. RX_PKT_SIZE interrupt shall be raised.	4.7.7	CREATED

4910	In case of write command with data length exceeding 32 bits, RHI shall write a received packet descriptor associated in memory, at current RX descriptor address. If an error has been detected but data has already been written into memory, the RX packet descriptor shall specify the fact the written data is potentially damaged.	4.13.1.3	CREATED

4915	After data correct reception and storage (write or read-modify-write commands) and correct reply transmission, PACKET_REC interrupt shall be raised.	4.13.1.4 4.13.3.1	CREATED

4.4.7.3.2 Reception of a RMAP reply

In this situation, the SPW-RMAP RX host interface is seen as a RMAP master.

4920	If the received packet is a reply, RHI shall store the data reply (in case of a read or read-modify-write reply) in the flip/flop storage area, using the same mechanism as in DATA MODE.	4.13.2.3 4.13.3.1	CREATED

4925	RHI shall count the data in the reply and compare it to the length provided in the reply header. It shall provide a valid status and data length in the RX packet descriptor it shall build : • It shall assert TR bit if data turns out to be shorter than length specified in the reply header. • It shall assert EP if an EEP is detected. • It shall assert DC if a bad data CRC is detected.	4.13.2.5 4.13.2.4	CREATED

This requirement on the master side allows hiding to the host application the non-compliance with the RMAP standard as regards the read reply header status and data length effectively transmitted on the link.

4930	The RX packet descriptor shall be written into memory, at current RX descriptor address. It shall be followed by the first 7 bytes of the reply and a null byte, the whole representing a particular type of RX packet descriptor.	4.13.1.4 4.13.2.3 4.13.3.1	CREATED

4935	Reply status shall be checked. If OK, RMAP_REPLY_OK interrupt shall be raised, if NOK, RMAP_REPLY_NOK interrupt shall be raised.	4.13.1.4 4.14.5	CREATED

4940	If the reply doesn't provide data, RX packet descriptor fields PACKET LENGTH and PACKET ADDRESS shall be null.	4.13.1.4	CREATED

4.4.7.4 MIXED MODE

4945	In MIXED MODE, RHI shall decode the first 2 bytes to decide whether the received packet is a data-only packet or it is a RMAP packet. In case of a data packet, these two bytes shall not be stored.	4.15.1	CREATED

4950	In case of RMAP packet RHI shall apply the same process as in RMAP MODE.	4.15.3	CREATED

4955	In case of data-only packet RHI shall apply the same process as in DATA MODE. It shall be noted that if CRC has to be checked, Logical Address and Protocol ID bytes shall be within its scope, but these bytes shall not be written into memory.	4.15.2	CREATED

4.5 INTERRUPTS

4960	Refer to Registers Description paragraph for a detailed list of the interrupts. An interrupt shall remain active until its reset.	D	CREATED

4965	When an interrupt is asserted, the host application shall perform all the corresponding tasks before clearing this interrupt by writing into the corresponding bit of ITCREG register.	D	CREATED

4970	The ITMREG shall allow inhibiting the output interrupt signals (nominal interrupt and error interrupt) but shall not inhibit the ITPFREG register. ITPFREG register bits shall always be set.	4.14.2	CREATED

4975	The host application shall force interrupts needed by writing into the corresponding bit of ITPFREG register.	4.14.6	CREATED

4.6 TIME CODE TRANSMISSION AND RECEPTION

Refer to [AD-10] for detailed description of time interface mechanism.

4980	To send a time code, the host application shall either generate a pulse on the TICKIN_CTM input signal or assert the TICKIN bit of the CTRLREG register. A time code shall be sent when a rising edge is detected on TICKIN_CTM or TICKIN.	4.11	CREATED

4985	It shall be possible to initialise the time code value by writing in the TIMCODREG register (TIMESEND_REG byte).	4.11	CREATED

4990	When a correct time code is received, the SPW-RMAP shall generate a pulse on the TICKOUT_CTM output signal and the TICKOUT interrupt shall be asserted. The received time code value shall be written in the TIMCOD register (TIMEREC_REG byte).	4.11	CREATED

4.7 TEST MODE

4995	The test mode shall be activated when TEST_MODE_HARD is high. The test mode shall allow to invalidate an area by writing in the CTRLREG register. Otherwise, the only way to invalidate an area is to send enough data to fill that area. In simulation, that task can be long, so the TEST_MODE_HARD can make it faster.	D	CREATED

5 REGISTERS

5.1 REGISTERS LIST

APB Address Offset	Register Name	Description
0x00	CONFREG	Configuration register
0x04	CTRLREG	Control register
0x08	STATREG	Status register
0x0C	NAKPIDREG	Node Address / Destination Key / Data PID register
0x10	TXPKTSIZEREG	Maximum Transmitted packet Size register
0x14	RXPKTSIZEREG	Maximum Received Packet Size register
0x18	TIMOUTREG	Time Out register
0x1C	TIMCODREG	Time Code register
0x20	ITPFREG	IT Pending / Force register
0x24	ITCREG	IT Clear register
0x28	ITMREG	IT Mask register
0x2C	TXDESCREG	TX Segments Descriptor Address register
0x30	RXDESCREG	RX Packets Descriptor Address register
0x34	RXDESCENDREG	RX Packets Descriptor End Address register
0x38	START1REG	Start Address 1 register
0x3C	START2REG	Start Address 2 register
0x40	AREASIZEREG	RX areas size register
0x44	CBEREG	Current Buffer End register
0x48	BURSTDISREG	AHB Burst disable register
0x4C	NAKPIDMASKREG	Node Address / Destination Key / Data PID Mask register

Table 12 : Registers list

It shall be noticed that in next paragraph detailed description of SPW-RMAP registers is given and all register fields have the same behaviour as regards their Read/Write status :

Read/Write fields : they can be written and read with no restriction

Read only fields : they can be written, it won't lead to any effective write, and they have sense only when read.

Write only fields : they can be written with no restriction, and if read they will return 0, but such a read will no real sense.

Reserved fields : they are specified as Read only, so they behave like Read only fields, with the additional specificity that all read return a 0 value, and have no real sense.

5.2 REGISTERS DETAILED DESCRIPTION

Bits	Field Name	Description	R/W	Reset value
31..24	FREQ_INIT	Configuration of the TX clock frequency used during the link initialisation. Gated TX clock mode : TX freq = input TX clock freq / 2(FREQ_INIT+1). Not-gated TX clock mode : TX freq = input TX clock freq / (FREQ_INIT+1).	R/W	depending on strap_en and strap_freq
23..16	FREQ_RUN	Configuration of the TX clock frequency used after the link initialisation. Gated TX clock mode : TX freq = input TX clock freq / 2(FREQ_RUN+1) if TX_MAX_EN=0. Not-gated TX clock mode : TX freq = input TX clock freq / (FREQ_RUN+1).	R/W	depending on strap_en and strap_freq
15..9	RESERVED	Reserved	R	0
8	TX_MAX_EN	This bit is only used with the gated TX clock configuration. In run state, the TX frequency used will be the same as the input TX clock frequency if this bit is asserted. 0: max TX frequency Off 1: max TX frequency On	R/W	0
7	RESERVED	Reserved	R	0
6	RESERVED	Reserved	R	0
5	MEM_ALIGNMENT	This bit is only used for flip/flop area data storage. 0 : Received packets have to be concatenated. 1 : Received packets shall be aligned on word addresses in host memory	R/W	0
4	RESERVED	Reserved	R	0
3	DATA_CRC	This bit is only used for data-only packets. 0 : no CRC 1 : CRC used	R/W	0
2	RM_SLV_DIS	RMAP slave functionality disabled. 1 : no RMAP command accepted 0 : RMAP commands accepted if RMAP protocol supported.	R/W	0
1..0	PKT_MODE	Functional mode of the reception function : 3 : Same as 0 2 : MIXED MODE (RMAP and data-only packets supported) 1 : RMAP MODE (RMAP packets supported) 0 : DATA MODE (data-only packets supported)	R/W	1

Table 13 : [0x00] Configuration register [CONFREG]

Note : CONFREG can only be reset by the RESETN hardware reset.

Bits	Field Name	Description	R/W	Reset value
31..8	RESERVED	Reserved	R	0
7	RST	Synchronous reset of the SPW-RMAP, active high. This reset lasts at least 10 system cycles and 3 TX cycles and generates the activation of SPWRST_O output port, during this period.	R/W	0

6	ABORT_PACKET	Abortion of the data transfers. Writing a '1' launches the abortion process. Writing a '0' has no effect. This bit is automatically reset when the abortion process ends.	R/W	0
5	AREA2_VALID	This bit validates the area 2. So the SPW-RMAP can use the area 2 to store data. Writing a '1' validates the area 2. Writing a '0' has no effect. So the user shall write '0' when the write access has a purpose other than validating this area. This bit is automatically reset when the area 2 is full. In test mode, writing a '0' will reset this bit.	R/W	0
4	AREA1_VALID	This bit validates the area 1. So the SPW-RMAP can use the area 1 to store data. Writing a '1' validates the area 1. Writing a '0' has no effect. So the user shall write '0' when the write access has a purpose other than validating this area. This bit is automatically reset when the area 1 is full. In test mode, writing a '0' will reset this bit.	R/W	0
3	TICKIN	The host application can transmit a time code by asserting this bit. Writing a '1' launches the time code transmission Writing a '0' has no effect	W	0
2	LINK_DISABLED	The link is disabled. 0: link not disabled 1: link disabled	R/W	0
1	LINK_START	The link can start. 0: link not started 1: link started	R/W	0
0	LINK_AUTOSTART	The link automatically starts when a NULL character is received. 0: autostart off 1: autostart on	R/W	1

Table 14 : [0x04] Control register [CTRLREG]

Note : LINK_DISABLED, LINK_START and LINK_AUTOSTART can only be reset by the RESETN hardware reset.
When the link is disabled by asserting the LINK_DISABLED bit during a transfer, NO EEP should be added at the end of the packet.

Bits	Field Name	Description	R/W	Reset value
31..22	RESERVED	Reserved	R	0
21..16	OUTSTANDING_CNT	Outstanding counter value.	R	0
15..14	RESERVED	Reserved	R	0
13..8	CREDIT_CNT	Credit counter value.	R	0
7	SPILLING_RXFIFO	Indicates when the RX FIFO is being cleared : 0: Not being cleared 1: being cleared	R	0
6	GATED_TX_CLOCK	Indicates the TX clock configuration: 0: Not-gated TX clock configuration 1: Gated TX clock configuration	R	0
5	AREA2_USED	This bit is asserted when the host memory area 2 is used to store the data from the SPW-RMAP.	R	0
4	AREA1_USED	This bit is asserted when the host memory area 1 is used to store the data from the SPW-RMAP.	R	0

3	DMA_RUNNING	This bit indicates if the DMA is running or not. 0: DMA is not running 1: DMA is running	R	0
2..0	ST_TRANS	Status showing the link initialisation progression. 0 to 4: initialisation state (0=ErrorReset, 1=ErrorWait, 2=Ready, 3=Started, 4=Connecting) 5: run state	R	0

Table 15 : [0x08] Status register [STATREG]

Bits	Field Name	Description	R/W	Reset value
31..24	RESERVED	Reserved	R	0
23..16	DATA_PID	Data-only packets protocol ID (only useful in MIXED MODE)	R/W	0xF0
15..8	DEST_KEY	RMAP Destination Key	R/W	0x00
7..0	NODE_ADD	Node Logical Address	R/W	0xFE

Table 16 : [0x0C] Node Address / Destination Key / Data PID register [NAKPIDREG]

Note : NAKPIDREG can only be reset by the RESETN hardware reset.

Bits	Field Name	Description	R/W	Reset value
31..24	RESERVED	Reserved	R	0
23..0	TX_PKT_SIZE	Maximum number of bytes in a packet to be transferred.	R/W	0xFFFFFF

Table 17 : [0x10] Maximum TX Packet Size register [TXPKTSIZEREG]

Bits	Field Name	Description	R/W	Reset value
31..24	RESERVED	Reserved	R	0
23..0	RX_PKT_SIZE	Maximum number of data allowed to receive. For RMAP packet, the header is not included in this number.	R/W	0xFFFFFF

Table 18 : [0x14] Maximum RX Packet Size register [RXPKTSIZEREG]

Bits	Field Name	Description	R/W	Reset value
31..24	RESERVED	Reserved	R	0
23..16	DIS_CNT_LIM	This is the time out for the link disconnection detection. DIS_CNT_LIM = 1 means time out = 1 system clock period The DIS_CNT_LIM will take a definite value so that the time-out is set to 850 ns.	R/W	depending on strap_en and strap_freq
15..10	RESERVED	Reserved	R	0
9..0	DELAY_6_4	This is the 6.4 µs time out used in the link initialization protocol. The user will give a value so that this time out is about 6.4 µs. DELAY_6_4 = 1 means time out = 1 system clock period	R/W	depending on strap_en and strap_freq

Table 19 : [0x18] Time Out register [TIMOUTREG]

Note : TIMOUTREG can only be reset by the RESETN hardware reset.

Bits	Field Name	Description	R/W	Reset value
31..14	RESERVED	Reserved	R	0
13..8	TIMESEND_REG	The host application can initiate the time code value to send	R/W	0x00
6..7	RESERVED	Reserved	R	0
5..0	TIMEREC_REG	This register contains the received time code value	R	0x00

Table 20 : [0x1C] Time Code register [TIMCODREG]

Bits	Field Name	Description	R/W	Reset value
31..27	RESERVED	Reserved for future nominal interrupts	R	0
26	TICKOUT	This bit is asserted when a right time code has been received. Nominal Interrupt	R/W	0
25	RMAP_REPLY_OK	This bit is asserted when a right RMAP reply has been received. Nominal Interrupt	R/W	0
24	RMAP_REPLY_NOK	This bit is asserted when a wrong RMAP reply has been received. Nominal Interrupt	R/W	0
23	SEGMENT_SENT	This bit is asserted when a segment with Interrupt Enable activated has been transmitted. Nominal Interrupt	R/W	0
22	END_LIST	In TX process, when the SPW-RMAP reaches the end of the linked list of segments, this bit is asserted. Nominal Interrupt	R/W	0
21	ROLLOVER	This bit is asserted when the received packets descriptors area is wrapped around. Nominal Interrupt	R/W	0
20	PACKET_REC	This bit is asserted when a complete packet has been received. Nominal Interrupt	R/W	0
19	AREA_CHANGE	This bit is asserted when the SPW-RMAP reaches the end of a side of the flip-flop memory. Nominal Interrupt	R/W	0
18	CONNECTION_FAILED	This bit is asserted when the SPW-RMAP wants to initiate the link but fails to establish the connection. Nominal Interrupt	R/W	0
17	NO_AREA_VALID	When data have been received but any host memory area is available, this bit is asserted. Nominal Interrupt	R/W	0
16	LINK_NOT_ENABLED	This bit is asserted when the SPW-RMAP receives NULL character but is not configured to establish the connection (LINK_DISABLED On or (LINK_START Off and AUTOSTART Off)) Nominal Interrupt	R/W	0
15..11	RESERVED	Reserved for future error interrupts	R/W	0
10	CRC_ERR	This bit is asserted when a RMAP packet with bad CRC is received, or when a data packet with wrong CRC and the field DATA_CRC of the CONFREG register activated. Error Interrupt	R/W	0

9	PID_ERR	This bit is asserted when a RMAP header with unsupported protocol ID is received. Error Interrupt	R/W	0
8	AMBA_ERR	This bit is asserted when the SPW-RMAP receives an AHB error response. Error Interrupt	R/W	0
7	EEP_REC	This bit is asserted when an EEP has been received. Error Interrupt	R/W	0
6	TX_PKT_SIZE_ERR	This bit is asserted when the packet to send (without including RMAP Header in RMAP or mixed mode) has a size bigger than TX_PKT_SIZE. Error Interrupt	R/W	0
5	RX_PKT_SIZE_ERR	This bit is asserted when a received packet had been truncated due to MAX_PKT_SIZE violation, or the number of data received doesn't match the data length. Error Interrupt	R/W	0
4	CHAR_SEQ_ERR	This bit is asserted when a character sequence error is detected. Error Interrupt	R/W	0
3	CREDIT_ERR	This bit is asserted when the SPW-RMAP receives a FCT but the increment of the credit counter will exceed 56 or when the SPW-RMAP receives a data but is not waiting for any one. Error Interrupt	R/W	0
2	ESC_ERR	This bit is asserted when a received ESC character is followed by neither a FCT nor a data. Error Interrupt	R/W	0
1	PARITY_ERR	This bit is asserted when a parity error is detected. Error Interrupt	R/W	0
0	DISCONNECT_ERR	This bit is asserted when a link disconnection is detected. Error Interrupt	R/W	0

Table 21 : [0x20] IT Pending / Force register [ITPFREG]

Note : Writing '0' has no effect. Writing '1' forces the IT activation.

Bits	Field Name	Description	R/W	Reset value
31..23	RESERVED	Reserved	R	0
26..0	CLEAR	Write 1 to clear the corresponding IT	W	0

Table 22 : [0x24] IT Clear register [ITCREG]

Bits	Field Name	Description	R/W	Reset value
31..23	RESERVED	Reserved	R	0
26..0	MASK	0 = masked ; 1 = unmasked	R/W	1

Table 23 : [0x28] IT Mask register [ITMREG]

Bits	Field Name	Description	R/W	Reset value

31..4	TX_DESC_ADD	Writing here, the host application gives the address of the first segment of the linked list. The host application must not write again in this register before END_LIST activation. Reading here, the host monitors the progression in the linked list.	R/W	0
3..0	Reserved	TX_DESC_ADD shall be aligned on 4-words addresses.	R	0

Table 24 : [0x2C] TX Segments Descriptor Address register [TXDESCREG]

Bits	Field Name	Description	R/W	Reset value
31..4	RX_DESC_ADD	Writing here, the host application gives the address where the next received packet descriptor shall be written. The host application must not write in this register when reception is in progress. Reading here, the host application monitors the progression in the RX descriptors area.	R/W	0
3..0	Reserved	RX_DESC_ADD shall be aligned on 4-words addresses.	R	0

Table 25 : [0x30] RX Packets Descriptor Address register [RXDESCREG]

Bits	Field Name	Description	R/W	Reset value
31..4	RX_DESC_END	Writing here, the host application gives the address where the received packet descriptors area finishes. The descriptors area is wrapped around when RX_DESC_END is reached. The host application must not write in this register when reception is in progress.	R/W	0
3..0	Reserved	RX_DESC_END shall be aligned on 4-words addresses.	R	0

Table 26 : [0x34] RX Packets Descriptor End Address register [RXDESCENDREG]

Bits	Field Name	Description	R/W	Reset value
31..0	START_AREA1	This address pointer indicates the start address of the host memory area 1.	R/W	0

Table 27 : [0x38] Start Address 1 register [START1REG]

Bits	Field Name	Description	R/W	Reset value
31..0	START_AREA2	This address pointer indicates the start address of the host memory area 2.	R/W	0

Table 28 : [0x3C] Start Address 2 register [START2REG]

Bits	Field Name	Description	R/W	Reset value
31..0	AREA_SIZE	Number of bytes for storage area1 and area2.	R/W	0

Table 29 : [0x40] Area Size register [AREASIZEREG]

Note: For nominal behaviour, the following conditions shall be respected:

- START_AREAx + AREA_SIZE < FFFFFFFFH
- AREA_SIZE > RX_PKT_SIZE

Bits	Field Name	Description	R/W	Reset value
31..0	CUR_BUF_END	This address pointer indicates the current end of the used host memory area. All addresses between the start address value and the address pointer value (address pointer value not included) contain valid data. As the RX storage areas are only used to store data from data packet or RMAP read reply packet or RMAP RMW reply packet, the CUR_BUF_END is only updated in these cases. For example, the CUR_BUF_END shall not progress while receiving a RMAP write packet.	R/W	0

Table 30 : [0x44] Current Buffer End register [CBEREG]

Bits	Field Name	Description	R/W	Reset value
31..1	RESERVED	Reserved	R	0
0	BURST_DIS	0: AHB burst not disabled 1: AHB burst disabled	R/W	0

Table 31 : [0x48] Burst disable [BURSTDISREG]

Bits	Field Name	Description	R/W	Reset value
31..16	RESERVED	Reserved	R	0
23..16	DATA_PID_MASK	Data-only packets protocol ID Mask (only useful in MIXED MODE) Bit = 1 means masked	R/W	0
15..8	DEST_KEY_MASK	RMAP Destination Key Mask Bit = 1 means masked	R/W	0
7..0	NODE_ADD_MASK	Node Logical Address Mask Bit = 1 means masked	R/W	0

Table 32 : [0x4C] Node Address / Destination Key / Data PID Mask register [NAKPIDMASKREG]

6 MEMORY MAPPINGS

Below is an example of Transmitter and Receiver Memory mappings, and of the Segments/Packets linked lists structure.

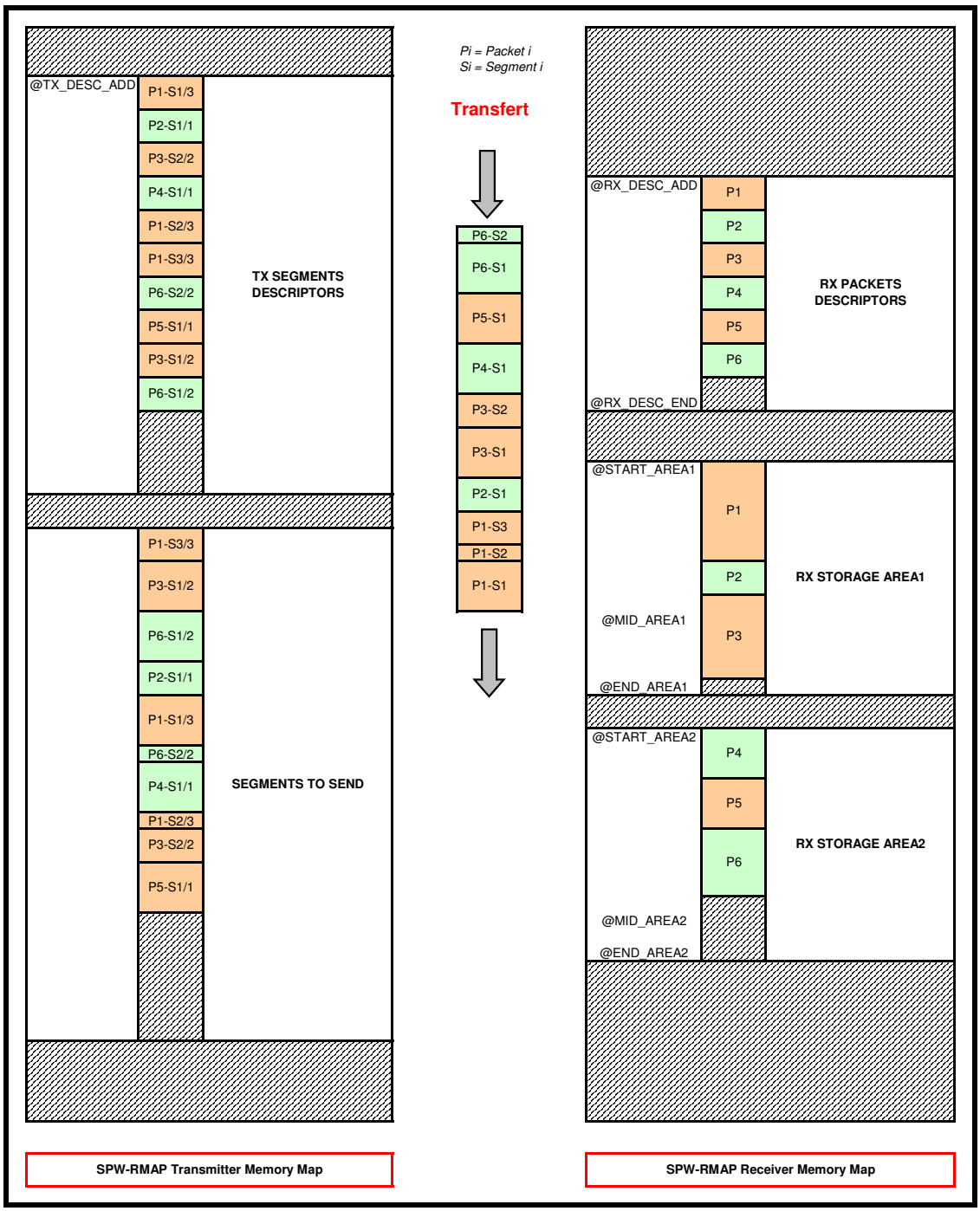


Figure 5 : Transmitter and Receiver memory mappings

7 INTERFACES

7.1 BASIC INTERFACE

Signal name	Direction	Description
CLK_SW	I	System clock
RESETN	I	Synchronous reset, active low. Asynchronous reset for IFAHB and IT register.
SPWRSTI	I	Synchronous reset input, active high
SPWRSTO	O	Synchronous reset output, active high
CLK_TXIN	I	TX clock
TEST_MODE_HARD	I	Test mode hardware configuration

Table 33 : Basic interface

7.2 APB SLAVE INTERFACE

The APB slave interface is used configure the SPW-RMAP and to retrieve statuses.

Signal name	Direction	Description
PSEL	I	slave select
PENABLE	I	strobe
PADDR (31..0)	I	address bus (byte)
PWRITE	I	write
PWDATA (31..0)	I	write data bus
PRDATA (31..0)	O	Read data bus

Table 34 : APB slave interface

7.3 AHB MASTER INTERFACE

The AHB master interface performs all read or write into the host memory.

Signal name	Direction	Description
HGRANT	I	Bus grant
HREADY	I	Transfer done
HRESP (1..0)	I	Response type
HRDATA (31..0)	I	Read data bus
HCACHE	I	Cacheable data
HBUSREQ	O	Bus request
HLOCK	O	Lock request
HTRANS (1..0)	O	Transfer type
HWRITE	O	Address bus (byte)
HADDR (31..0)	O	Read/write
HSIZE (2..0)	O	Transfer size
HBURST (2..0)	O	Burst type
HPROT (3..0)	O	Protection control
HWDATA (31..0)	O	Write data bus

Table 35 : AHB master interface

7.4 LINK INTERFACE

The Link interface brings together the data and strobe signals of the transmission and the reception.

Signal name	Direction	Description
D_IN	I	Data input
S_IN	I	Strobe input
D_OUT	O	Data output
S_OUT	O	Strobe output

Table 36 : Link interface

7.5 TIME INTERFACE

The Time interface manages the transmission and the reception of time code.

Signal name	Direction	Description
TICKIN_CTM	I	Time code to send
TICKOUT_CTM	O	Good time code received

Table 37 : Time interface

7.6 INTERRUPT INTERFACE

The Interrupt interface is used to warn the host application when a specific event appears.

Signal name	Direction	Description
ERR_INT	O	Error interrupt
NOM_INT	O	Nominal interrupt

Table 38 : Interrupt interface

7.7 HARDWARE FLIP/FLOP MANAGEMENT INTERFACE

The hardware flip/flop management interface is used to communicate with another IP in charge of the management of the flip/flop area without host application intervention.

Signal name	Direction	Description
AREA1_FULL	O	Area1 full, ready for DMA read
AREA2_FULL	O	Area2 full, ready for DMA read
AREA1_RDY	I	Area1 ready, ready for SPW-RMAP write
AREA2_RDY	I	Area2 ready, ready for SPW-RMAP write

Table 39 : Hardware flip/flop management interface

8 DETAILED ARCHITECTURE

Refer to [AD-11] for an overview of the baseline architecture.

8.1 MODIFICATIONS OVERVIEW

The main changes occur in host interface:

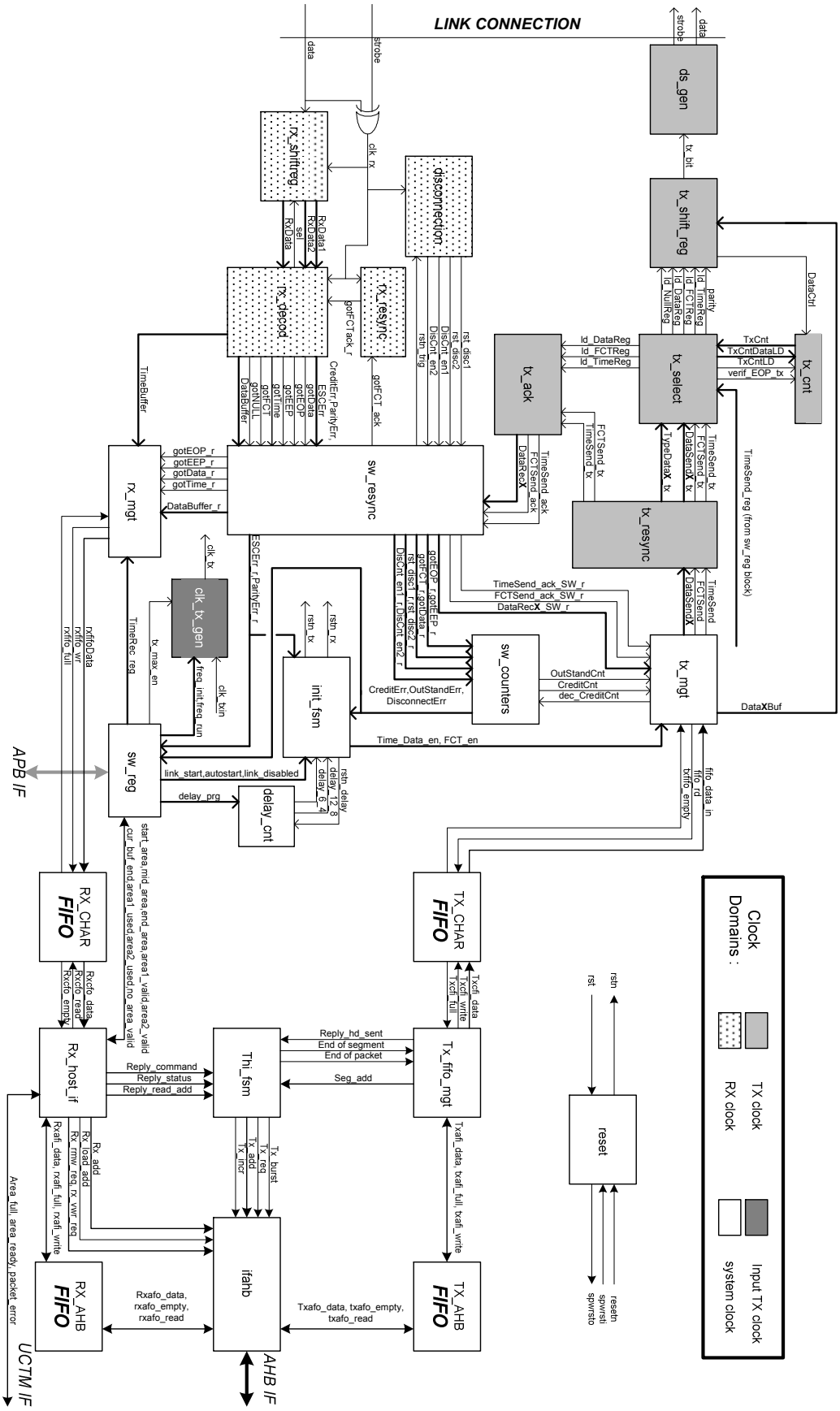
- AHB slave interface is suppressed.
- TX and RX AHB master interfaces are merged into only one AHB master interface.
- 2 36-bits wide AHB fifos are needed for TX and RX host interfaces.
- Use of 32-bits burst4 for RX and TX AHB operations.
- Handling of non-aligned memory accesses, using byte, half-word or word single accesses on AHB bus.
- Implementation of RMAP protocol.
- Limitation of the size of transmitted and received packets upgraded to 16 MB.
- Possibility to limit transmitted packet size to a value defined in a register.
- Possibility to limit received packet size to a value defined in a register.
- Hardware flip/flop area management mechanism.
- Use of segments for packet transmission.
- Storage in memory of a descriptor for received packets.
- Possible use of a CRC in data-only packets.

The changes that impact the whole IP are detailed hereafter:

- Modification of the reset tree: synchronous reset, possibility to reset the IP by host application intervention…
- Modification of the timecode size to be compliant of [AD-10].
- Reorganisation of configuration registers.

8.2 SPACEWIRE CORE ARCHITECTURE

The figure below gives an overview of the Spacewire RMAP module architecture.



NbCars 69942
NbWords 13939
FileName Spec_Archi_RMAP_rev05.doc

Figure 6 : Spacewire Core Architecture

The above schema also describes the clock trees. The TX clock used for the transmission is made from the input TX clock. The RX clock is made from the data and strobe input signals.

8.3 RESET MANAGEMENT

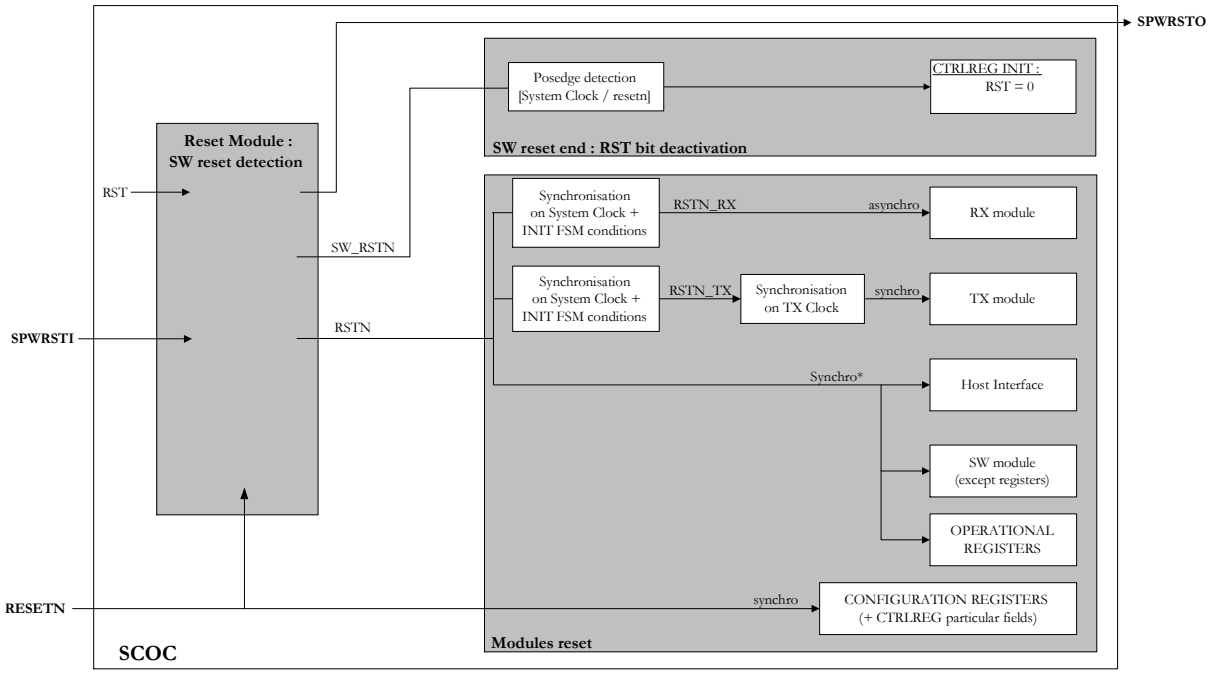


Figure 7 : Reset management

synchro* : All have synchronous reset except for the AHB interface and the IT registers which have asynchronous reset. For some configurations, the SPW has no input clock, so AHB interface and IT need an asynchronous reset in order not to deliver wrong request or message.

All the blocks working at RX clock have a asynchronous reset RSTN_RX, otherwise they cannot be reset if the link is disconnected. The RSTN_RX reset is not synchronously deactivated on the RX clock.

The RX module includes all blocks working at RX clock.

The TX module includes all blocks working at TX clock.

The SW module, host interface and registers represent all blocks working at system clock.

SW_RSTN occurrence leads to a synchronous reset, activated during 10 cycles, followed by the force of RST bit to 0, at the end of the reset.

LINK_DISABLE and LINK_AUTOSTART fields are not modified by a SW synchronous reset, they have to be correctly configured before this reset.

On Master SW side, it's a write to CTRLREG setting RST bit to 1 which starts a SW synchronous reset.

8.3.1 Reset Module

The Reset module is in charge of the detection of a SW reset, activated by the RST bit or by a rising edge of the SPWRSTI input.

In case of RST activation or SPWRSTI rising edge, it is responsible for forcing reset during at least 10 system cycle and 3 TX cycles, reflecting this on SPWRSTO output.

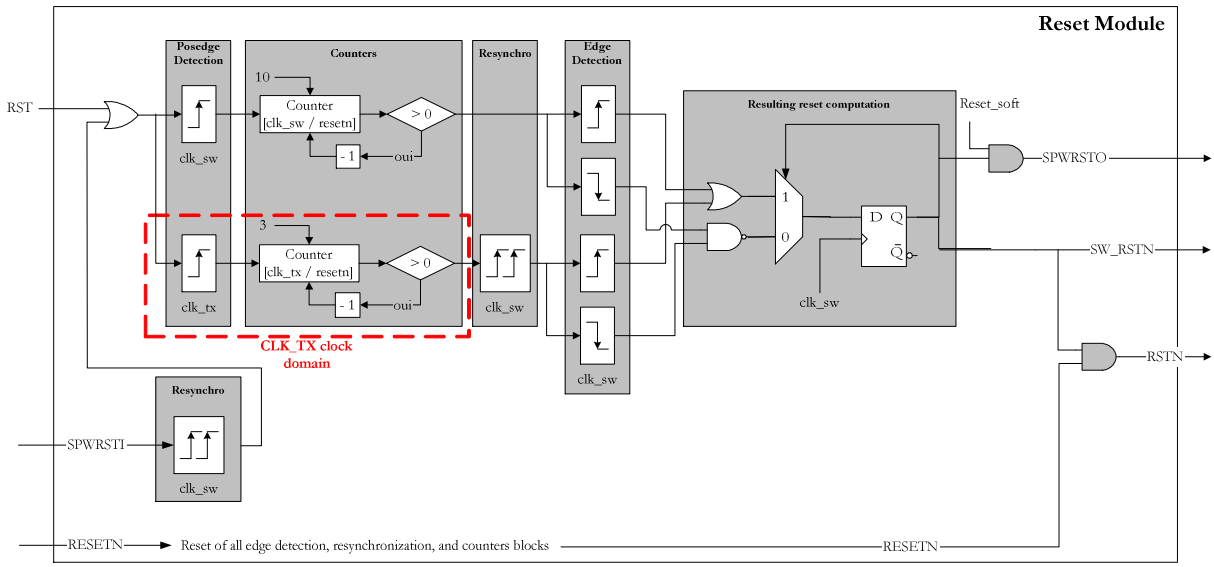


Figure 8 : Reset module architecture

SPWRSTO is only activated if the SW reset is generated by the RST bit. An activation of the SPWRSTI does not lead to an activation of SPWRSTO.

8.3.2 Blocks working at TX clock

8.3.2.1 CLK_TX_GEN block

There are two different architectures for the CLK_TX_GEN block :

- The first one is a gated TX clock. The generated TX clock has a various frequency following the 2(n+1) frequency divider.
- The second one is a not-gated TX clock. The generated TX clock has a constant and equal frequency to the input TX clock. The use of an enable signal (clk_tx_en) allows the TX frequency variation. A (n+1) frequency divider is used.

The GATED_TX_CLK parameter selects the CLK_TX_GEN block architecture:

- GATED_TX_CLK = True : gated TX clock is used. In this architecture, clk_tx is variable and clk_tx_en is stuck at 1.
- GATED_TX_CLK = False: not-gated TX clock is used. In this architecture, clk_tx is clk_txin, so it is constant and clk_tx_en is variable.

8.3.2.2 DS_GEN block

The goal is to generate the data and strobe signals according to the AD11 specification.

D and S cannot be reset at the same time because it can generate a glitch on the RX clock. So the reset of this block is synchronous, S is reset first then D.

8.3.2.3 TX_SHIFT_REG block

This block receives orders to load the shift registers then transmits the serial TX_BIT signal to the DS_GEN block.

As there are 4 data buffers (Data1LD(8:0), Data2LD(8:0), Data3LD(8:0) and Data4LD(8:0)), the block swaps from one to another each time a data is loaded. The DataSend_sel signal indicates which data is selected.

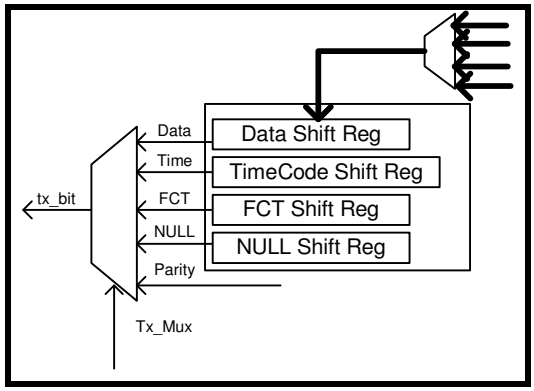


Figure 9 : TX shift registers

8.3.2.4 TX_SELECT block

This block manages the character transmission requests from the TX_MGT block. Following the priority order (time code > FCT > data > NULL), the TX_SELECT block generates the appropriate load signal to the TX_SHIFT_REG block.

The TX_SELECT block also activates the data or EOP/EEP check performed by the TX_CNT. Then this block manages the TX_CNT load.

The parity bit is computed in this block.

8.3.2.5 TX_CNT block

This 4-bit counter is used to count the characters length. Thus, the TX_SELECT block can generate the load signals in appropriate time.

This counter loads the TxCntDataLD value when the TxCntLD signal is high. The counter is corrected when an EOP/EEP is checked.

8.3.2.6 TX_ACK block

This block generates acknowledgement signals for the time code, FCT and data requests. The acknowledgement is activated when the corresponding shift register from the TX_SELECT block is loaded.

The DataRec1/DataRec2/DataRec3/DataRec4 signals are activated for 4 TX clock cycles, and then is automatically off after this time period.

After the activation of the TimeSend_ack/ FCTSend_ack signal, the deactivation is performed only when the TimeSend_tx/ FCTSend_tx signal is low.

8.3.2.7 TX_RESYNC block

This block performs the resynchronisation of the signals from blocks working at the system clock following the below architecture.

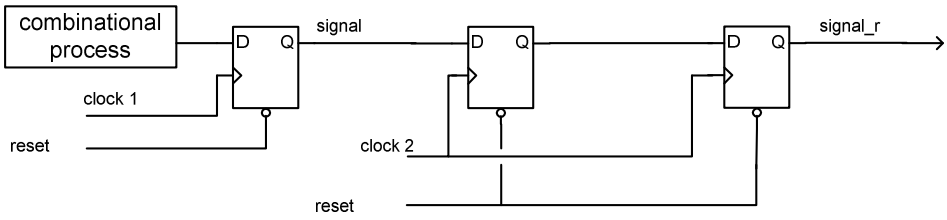


Figure 10 : TX Resynchronisation block

8.3.3 Blocks working at RX clock

The RX clock is built from the DATA and STROBE signals as shown hereafter:

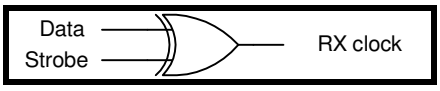


Figure 11 : RX clock generation

8.3.3.1 RX_SHIFTREG block

This block memorizes the input serial data on the rising and falling edge of the RX clock.

The RX_SHIFTREG block contains 2 shift registers. The one works on rising edge, the other on the falling edge.

The character can be received with its first bit sampled on falling or rising edge. So, RxData1(9:0) and RxData2(9:0) are used to determine on which edge the first bit is sampled.

This detection is only performed for the first NULL character. The first bit of the following characters is sampled on the same edge.

The RxData(9:0) word is either RxData1(9:0) or RxData2(9:0) following the SEL signal value which depends on the first bit detection on rising/falling edge.

The SEL signal is determined when the first NULL is detected and will remain unchanged as long as the link is running.

So, to detect the first NULL character, RxData1(9:0) and RxData2(9:0) are used. Then to detect the following characters, RxData(9:0) word is used.

The architecture of RX_SHIFTREG is shown hereafter:

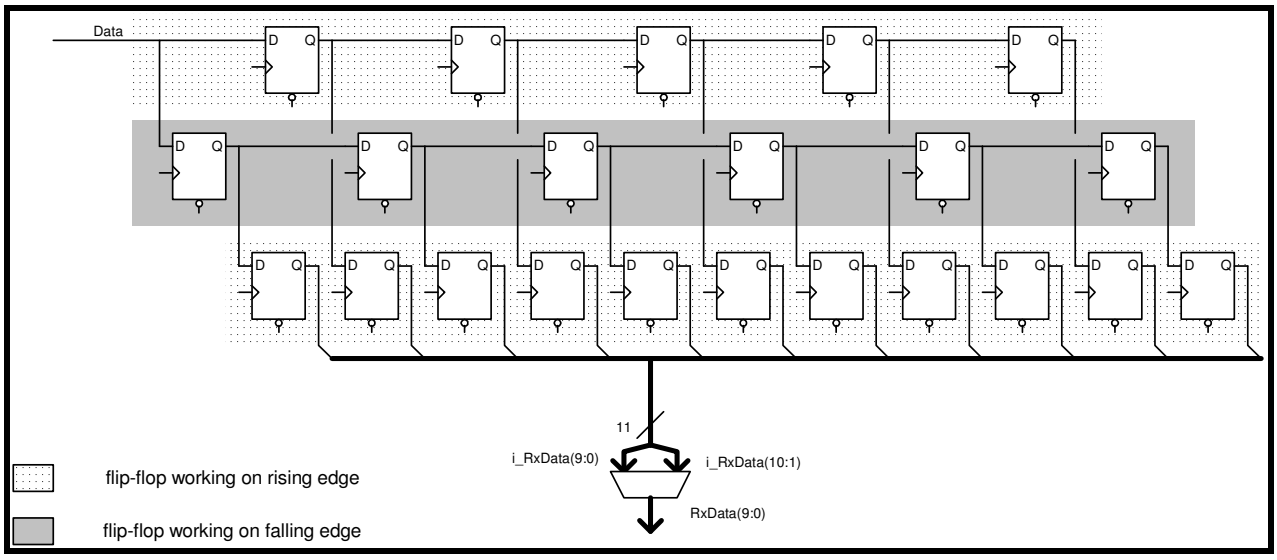


Figure 12 : RX shift registers

8.3.3.2 RX_DECOD block

The RX_DECOD block contains a 3-bit counter to note the number of FCT received.

Another 3-bit counter is used to determine the time that the character remains in the shift register (RX_SHIFTREG block).

The RX_DECOD block identifies the character type and verifies the parity.

When the parity is checked, the valid received character is flagged by a signal (gotData, gotEOP, gotEEP, gotTime, gotFCT or gotNULL). The gotData, gotEOP, gotEEP or gotTime is asserted for 2 RX clock cycles each time the corresponding character is received. The gotNULL is always asserted after the first NULL character reception.

As long as the 3-bit FCT counter value is not null, the gotFCT signal is generated. This signal uses a handshake protocol. Each time the gotFCT acknowledgement is received, the FCT counter is decremented and the gotFCT signal deasserted.

If the received character is a time code or a data, the value will be stored into the TimeBuffer or DataBuffer.

The block also generates 3 error signals. When the parity is false, the ParityErr signal is produced. The ESCErr indicates that a ESC is not followed by a FCT or a Data. Here, the CreditErr is asserted when the number of received FCTs is out of limit (>7). The SW_COUNTERS block also generates a CreditErr signal which depends on the number of data to be transmitted.

8.3.3.3 RX_RESYNC block

This block resynchronizes the signals from blocks working at system clock following the below architecture.

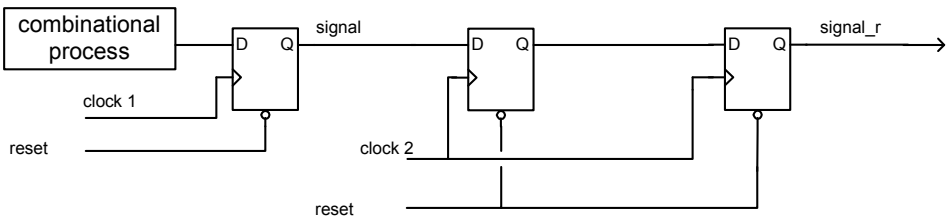


Figure 13 : RX Resynchronisation block

8.3.3.4 DISCONNECTION block

The DISCONNECTION block produces the reset and enables signals for the counter used to detect the link disconnection.

After the RX reset, the counter is enabled on the first edge of the RX clock. So, there are 2 enable signals (DisCnt_en1 and DisCnt_en2). The one is asserted on the rising edge of the RX clock, the other on the falling edge. Once they are asserted, the DisCnt_en1 and DisCnt_en2 signals remain activated.

Each time an edge of the RX clock occurs, the rst_disc1 or rst_disc2 signal is asserted to reset the counter. Then they are de-asserted once the counter is reset.

8.3.4 Blocks working at system clock

8.3.4.1 INIT_FSM block

This block contains the FSM described in the [AD-10]. This FSM manages the link initialisation protocol.

It also includes some additional outputs.

The ST_TRAN(2:0) is used to monitor the link initialisation progression:

ST_TRAN	STATE
000	ErrorReset
001	ErrorReset
010	Ready
011	Started
100	Connecting
101	Run

Table 40 : State signification

8.3.4.2 DELAY_CNT block

The DELAY_CNT block contains an 8-bit counter to compute the 6.4 µs and 12.8 µs delays used in the INIT_FSM block.

After reset (resetn or rstn_delay), the DELAY_6_4 signal goes high when the counter reaches the input CNTMAX(7:0) value once.

After reset (resetn or rstn_delay), the DELAY_12_8 signal goes high when the counter reaches the input CNTMAX(7:0) value twice.

8.3.4.3 RX_MGT block

The main purpose of this block is to store the RX data into the RX FIFO. The block receives an 8-bit data and stores it into the RX FIFO, adding a control flag bit. It also stores EOP/EEP when gotEOP/gotEEP is asserted. The format is described in Table 1.

If the block receives an EOP/EEP and another EOP/EEP later (without any data between the 2 EOP/EEP), only the first EOP/EEP will be written into the RX FIFO, the second one will not be taken into account.

The detection of time code, data, EOP or EEP is done on the rising edge of gotTime_r, gotData_r, gotEOP_r or gotEEP_r.

An interrupt (EEPRec signal) is generated when an EEP is received.

When a new time code is received, the block compares the new time code value (TimeBuffer) with the last stored time code value (TimeRec_reg). The TICKOUT signal is asserted when TimeBuffer=TimeRec_reg+1.

8.3.4.4 RX_FIFO block

The synchronous RX FIFO can contain 64 9-bit words. This FIFO stores the RX data.

8.3.4.5 TX_FIFO block

The synchronous TX FIFO can contain 8 9-bit words. This FIFO stores the TX data.

8.3.4.6 TX_MGT block

The TX_MGT block retrieves data from the TX FIFO and generates character transmission requests to the TX_SELECT block.

There are 4 data buffers (Data1Buf, Data2Buf, Data3Buf and Data4Buf) to keep the maximum data transfer rate. The DataSend1 request corresponds to the buffer 1, the DataSend2 request corresponds to the buffer2 and so on...

When more than one DataSend is asserted, the TX_SELECT block knows which one has priority because it takes it in turns.

When the TX reset (rstn_tx signal) rising edge is detected, the TX_MGT block flushes the TX FIFO until an EOP/EEP to delete the current data packet.

The following schema describes how the FCTSend signal is generated:

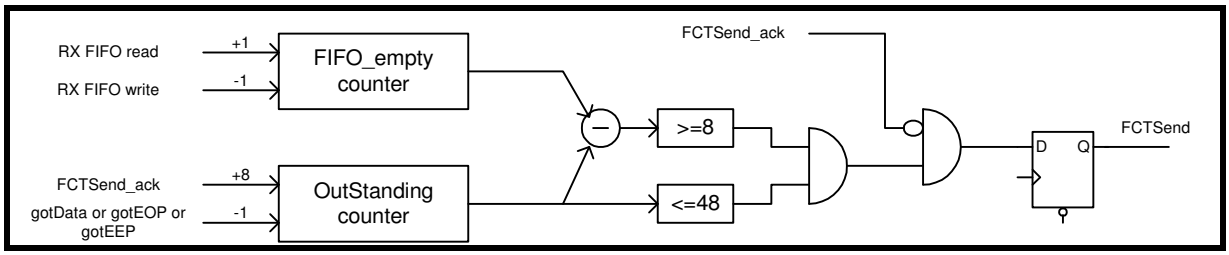


Figure 14 : FCT send function

The TimeSend request is activated after the TICKIN rising edge detection.

8.3.4.7 SW_COUNTERS block

This block contains 4 counters:

- The FifoECnt 7-bit counter is used to note the number of free space in the RX FIFO. This counter is incremented when a read is performed; it is decremented when a write is done. Its reset value is 64.
- The CreditCnt 6-bit counter is used to store the number of data that can be transmitted. Its reset value is 0. It is incremented by 8 when a FCT is received and is decremented when a data is transmitted.
- The OutStandCnt 6-bit counter is used to store the number of data that is expected to be received. Its reset value is 0. It is incremented by 8 when a FCT is transmitted and is decremented when a data is received.
- The DisCnt 8-bit counter is used to count the delay beyond which one the link disconnection error is activated. When the DISCONNECTION block enables this counter, it is incremented at each system clock period. Its reset value is 0. The reset is done at each edge of the RX clock.

The SW_COUNTERS block generates the Credit Error when its value is out of 56.

The OutStandErr signal is asserted when a data is received while no one is expected.

The DisConnectErr signal is asserted when the disconnection time out is reached.

8.3.4.8 SW_RESYNC block

This block resynchronises the signals from blocks working at RX or TX clock following the below architecture.

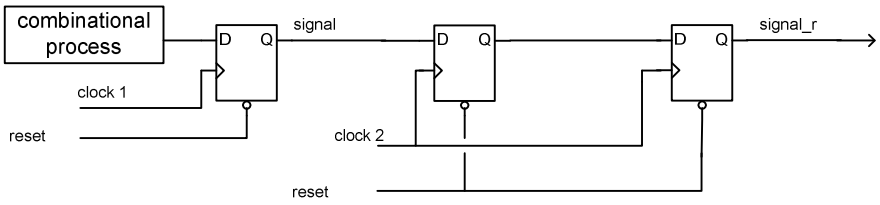


Figure 15 : SW Resynchronisation block

8.3.4.9 SW_REG block

This block contains all the registers described in the paragraph 5, except for the TXDESCREG register which is implemented in the TX Host Interface, and the RXDESCREG register which is implemented in the RX Host Interface. The read and write accesses to these registers through the APB interface are managed in the SW_REG block.

The management of the interrupts is also done here.

8.3.4.10 THI_FSM block

THI_FSM is responsible for the transmission of RMAP replies, the correct management and transmission of linked list of segments.

The following figure shows the FSM of the block.

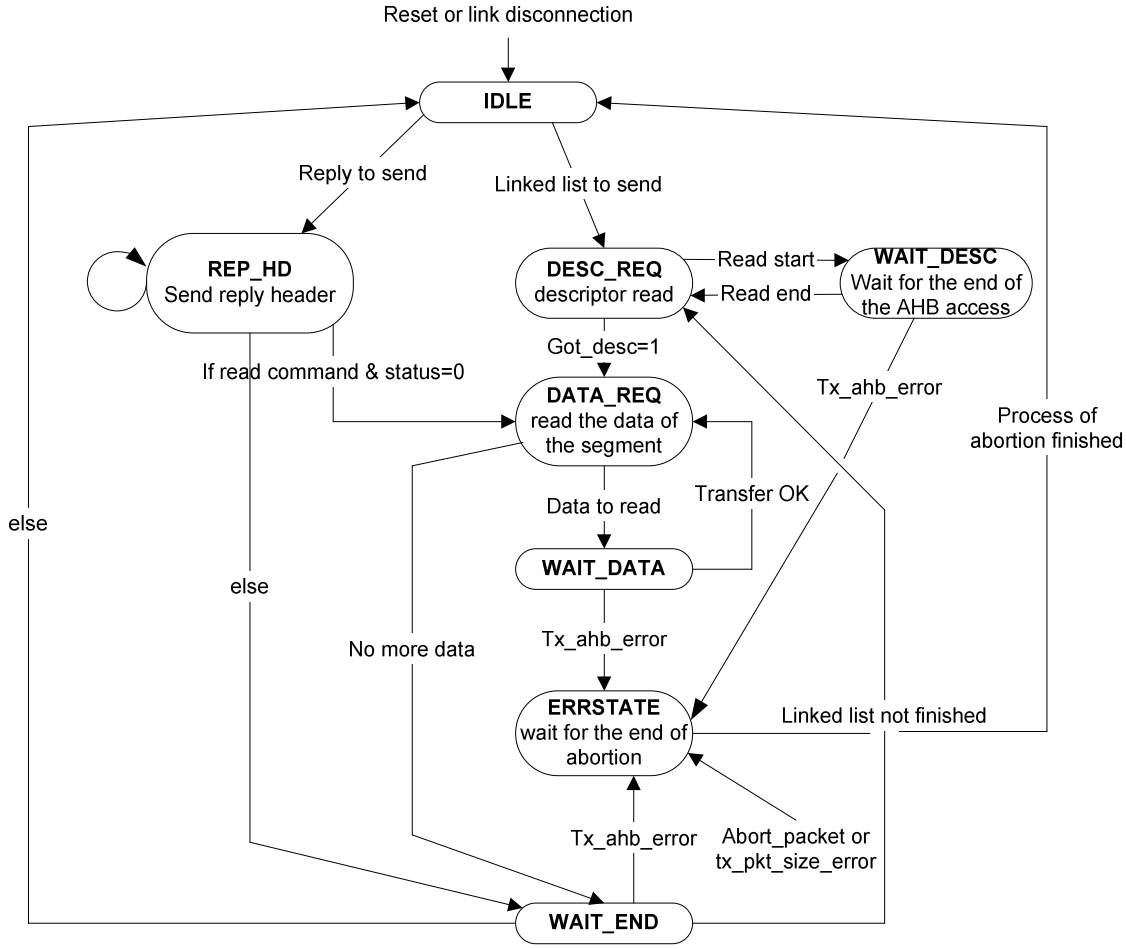


Figure 16 : THI_FSM diagram

In IDLE state : The FSM waits for a reply to send or a packet to send.

In REP_HD state : The FSM waits for the end of the header transmission. Then if the reply is a read reply without error status, the FSM goes to DATA_RED state to retrieve the data to send. In other cases, the FSM goes to WAIT_END state.

In DESC_REQ : If the TX descriptor is not read yet, the FSM starts a request and goes to WAIT_DESC. Once the TX descriptor read, the FSM goes to DATA_REQ.

In WAIT_DESC : The FSM waits for the end of the access (single or burst). If the response is not OK, the FSM goes to ERRSTATE, otherwise it returns to DESC_REQ.

In DATA_REQ : This state manages the different types of access (burst, single 32 bits, single 16 bits, single 8bits). After each request, the FSM goes to WAIT_DATA. When all the data of the segment are read, the FSM goes to WAIT_END.

In WAIT_DATA : The FSM waits for the end of the access. If the response is not OK, the FSM goes to ERRSTATE.

In WAIT_END : There is a priority management in this state. If it is not the end of a packet, TX packet has the priority and the FSM goes to DESC_REQ to read the next segment. If it is a reply or the end of a TX packet, the FSM goes to IDLE.

In ERRSTATE : The FSM waits for the end of the TX_FIFO_MGT error process then goes to IDLE.

8.3.4.11 TX_FIFO_MGT block

TX_FIFO_MGT performs the serialization of 32 bits words received from TX AHB FIFO into byte words transmitted into TX CHAR FIFO. It is responsible for the computing and the transmission of CRC bytes, of the addition of EOP or EEP bytes, the transmission of RMAP replies needed by RHI.

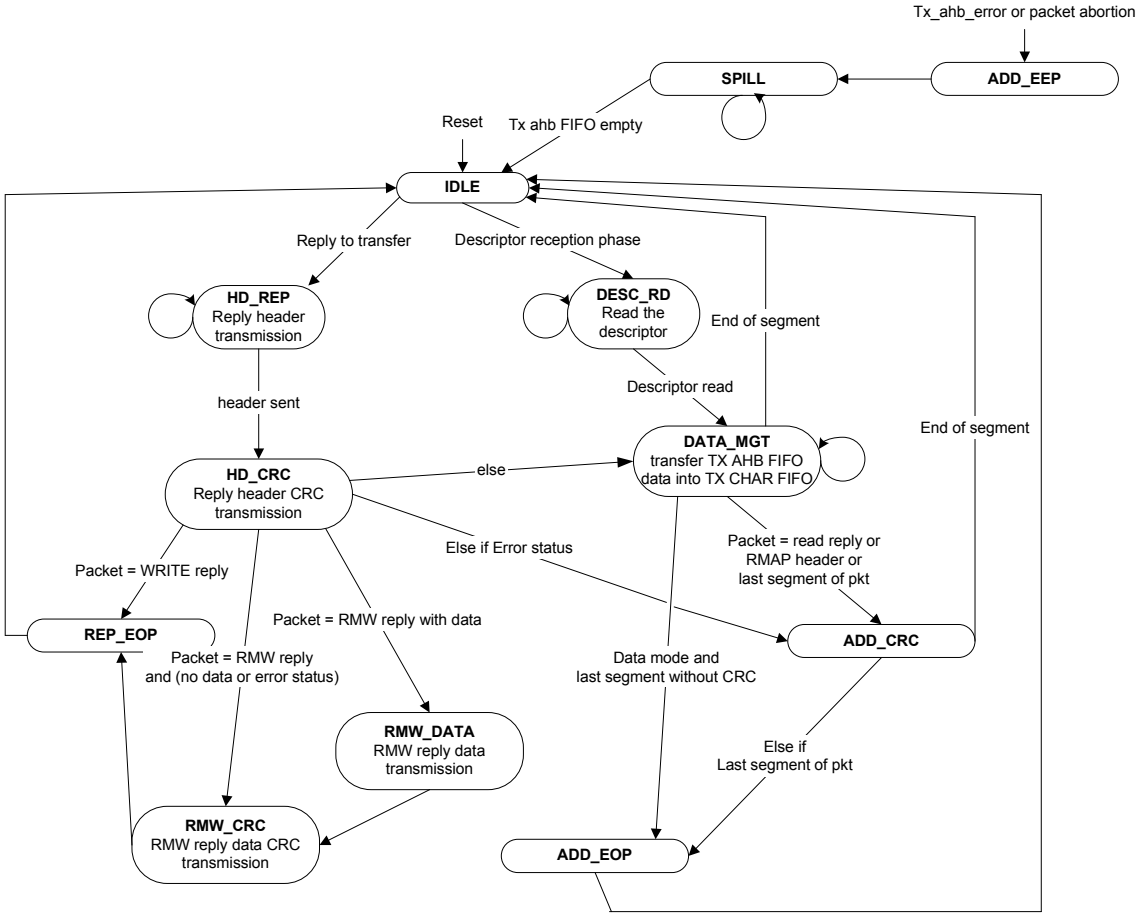


Figure 17 : TX_FIFO_MGT diagram

8.3.4.12 RX Host Interface [RHI]

RX Host Interface [RHI] performs the parallelization of byte words received from RX CHAR FIFO into 32 bits words transmitted into RX AHB FIFO. RHI is responsible for the decoding of received packets, and of their correct interpretation. It is able to manage RMAP commands, replies or data-only packets. RHI is responsible for the correct storage of data packets into host memory, and of the emission of a RMAP reply if needed.

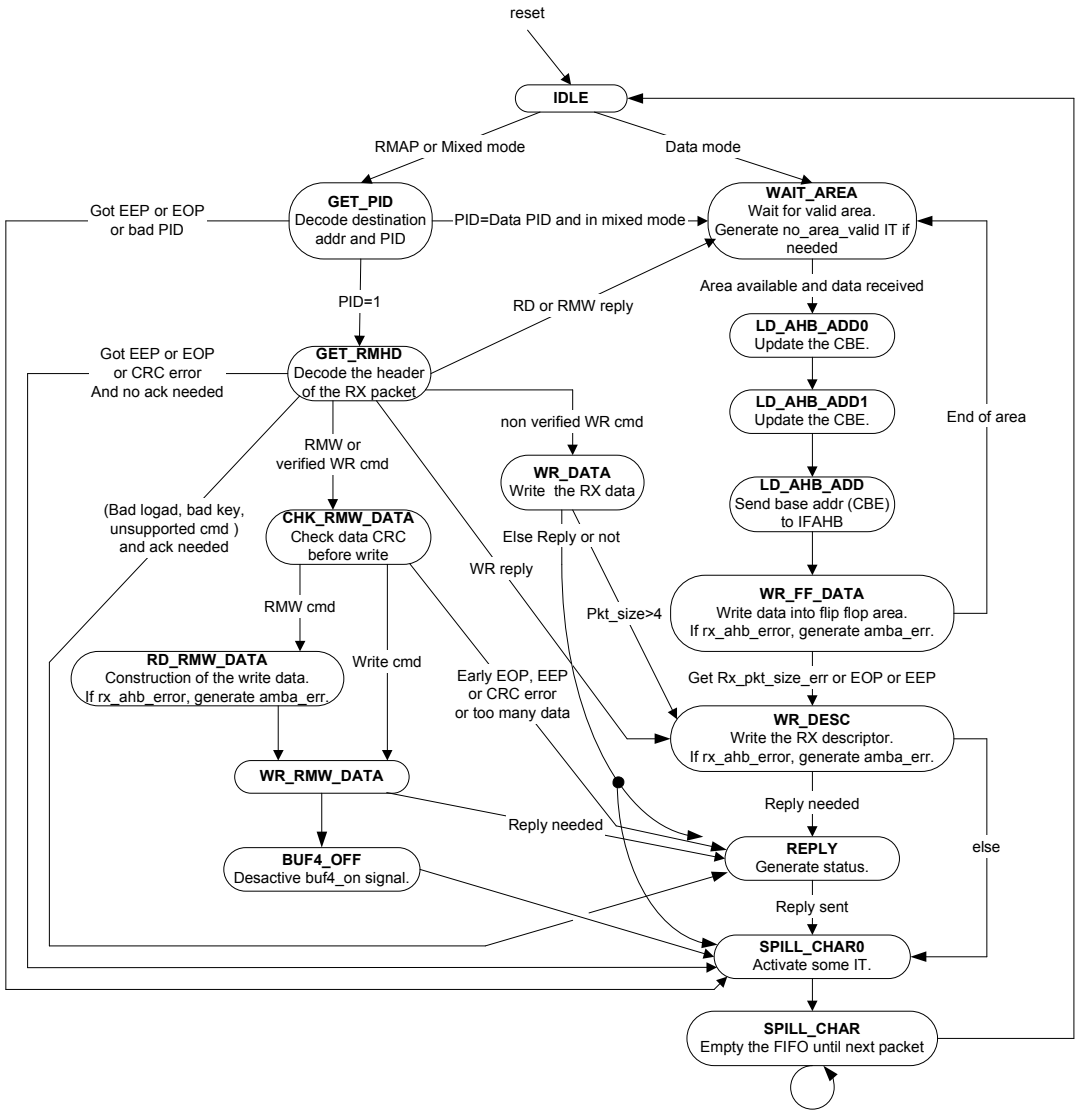


Figure 18 : RX Host Interface FSM

8.3.4.13 AHB Interface [IFAHB]

THI and RHI access to host memory through one AHB master interface, managed by the IFAHB module. IFAHB is responsible for the correct read or write operations into host memory, through AHB bus, using as much as possible burst4 transfers. It is also able to manage non-aligned accesses.

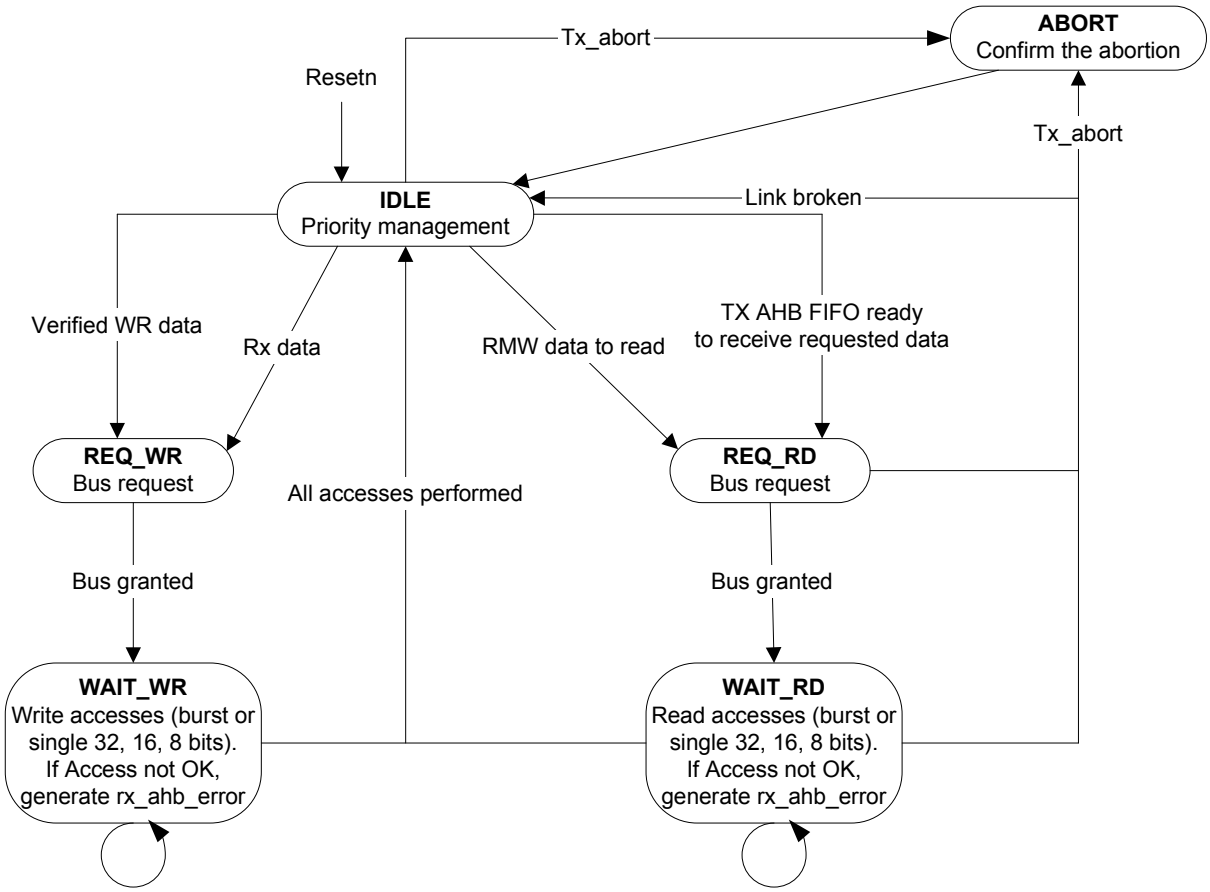


Figure 19 : AHB Interface FSM

8.3.4.14 FIFO AHB RX (ahb_fifo_rx) and FIFO_AHB_TX (ahb_fifo_tx)

The structure of the 2 FIFOs is shown hereafter. The TPRAM addrb signal is equal to adr_rd or adr_rd + 1, depending if a read is being proceed or not. This architecture allows a new data present on dataOut at each clock cycle. This is necessary during an AHB burst.

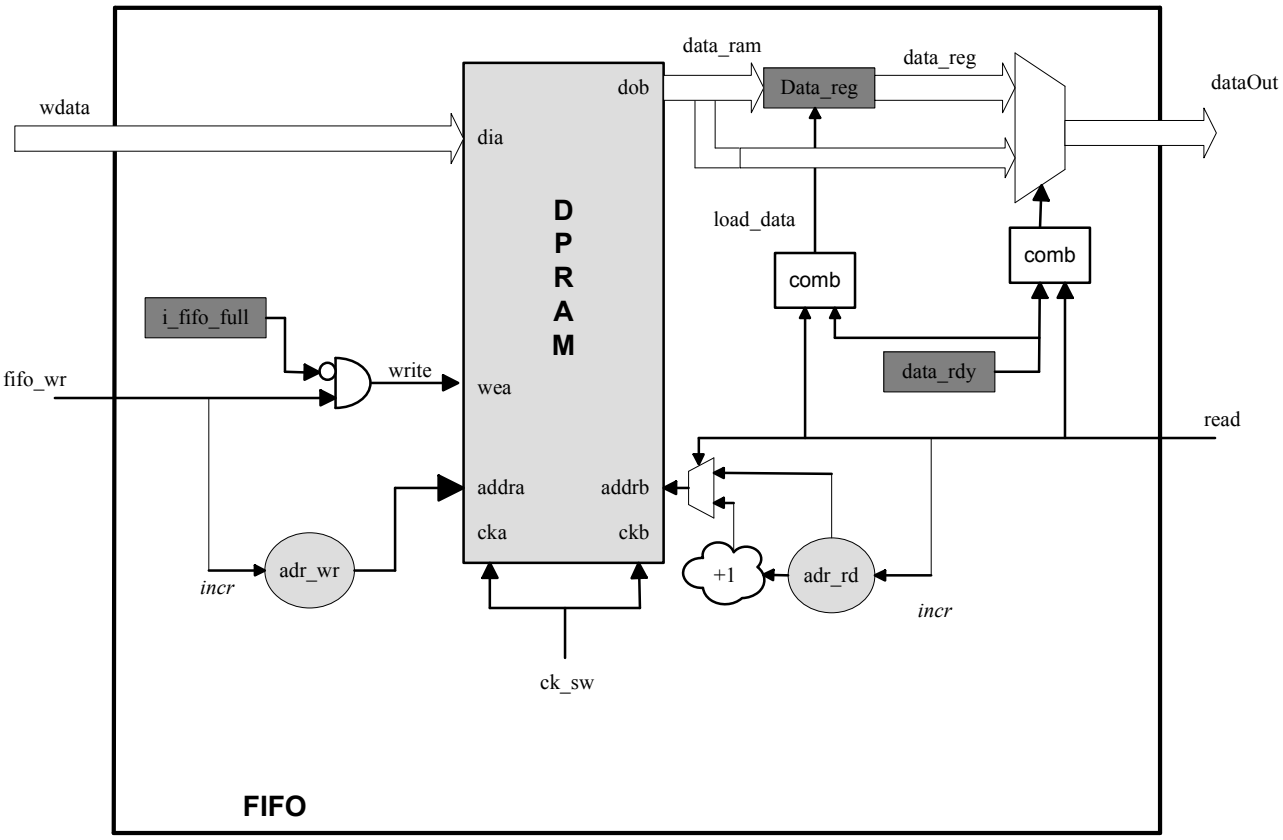


Figure 20: FIFO AHB structure

8.3.5 Block working at input TX clock

8.3.5.1 CLK_TX_GEN block

The architecture of the block is shown hereafter:

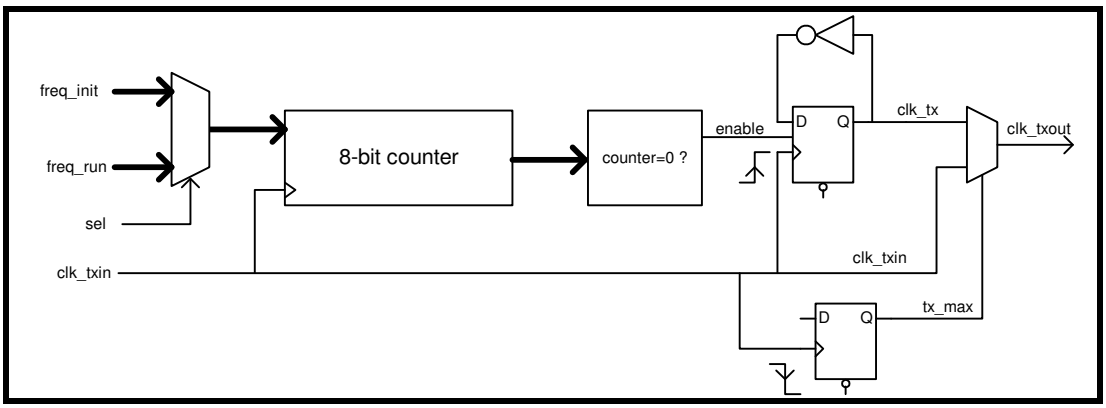


Figure 21 : TX clock generation

The internal clk_tx signal changes its value each time the counter value is 0.

The clock selection between clk_tx and clk_txin is done by the tx_max signal.

To avoid any glitch on the clk_txout signal, the tx_max signal is updated on clock falling edge while the clk_tx signal is updated on clock rising edge.

DISTRIBUTION LIST

	Overall document		Summary
	Action	Information	