Internal Specification

E2x-FCC2/PSIS011 model

(v1.2)

Summary:

This document describes the Detail Specification of E2x-FCC2/PSIS011 model

Relative Document

Reference Manuals				
No.	**Title name**	**Document number**	**Description**	**Path**
1	SC-HEAP_E3 common requirement (v1.0)	-	The common requirement (**File:** Common_Requirement_RVC.pdf)	**DMS:** Documents/010_ENG/140_FrontEnd/Project/01_SLD/2_SLD_Project/Model_Documents/02_MCS_Project/From_MCS
2	SC-HEAP_E3 Modeling guideline (Rev. 4.00) SC	IDF-14-010278-01	This document describes the Guideline for peripheral macro development which is connected to SC-HEAP_E3 simulator (**File**: SC-HEAP_E3_Modeling_Guideline.pdf)	
3	SC-HEAP_E3 BUS I/F outline	LLWEB-00010925 ZSG-F31-12-0029-01	The document describes the outline of bus I/F applied to SC-HEAP_E3 (**File**: scheap_e3_bus_if_outline_E.pdf)	

Reference Manuals				
4	SC-HEAP_E3 PYTHON I/F function specification (v2.0)	LLWEB-00105192 MSS-SG-12-0062-02	The document describes how to use python interface (**File**: SC-HEAP_E3 Python IF_t.pdf)	
5	RH850/E1x-FCC2 Hardware User's Manual. Section 25 Peripheral Sensor Interface 5 S (PSI5-S) (Rev.0.50)	R01UH0641EJ0050 (Rev.0.50 Dec 28, 2016)	Hardware user's manual of E2x-FCC2/PSI5S. (**File:** E25_PSI5S.docx)	-
6	M40PF uhiappsis011 (PSI5S) IP Specification (Ver.01.00)	uhiappsis011 (Ver.01.00 Nov 16, 2016)	IP target specification of E2x-FCC2/PSI5S. (**File:** uhiappsis011_IPSpec_v1.00.pdf)	-

***Note**: (*) Refer to TRA-MCS-17013_PSIS011 and DEV-MCS-17013_PSIS011 for version number.*

Contents

Index of Figures

Index of Tables

1.Model summary

(1) PSIS011 is a SystemC model of E2x-FCC2 platform, presents for PSI5S module. PSI5S stands for Peripheral Sensor Interface 5 S.

(2) Compare with HW specification, this model has some limitations. Refer to chapter 9 - Limitation for detail.

(3) In this design, the followings are supported in PSIS011 model:

(3.1) 32-bit width of APB bus.

(3.2) One target sockets for access registers inside. Refer chapter 3 for detail about registers.

(3.3) Both loosely time mode (LT) and approximately time (AT) mode.

(3.4) Little endian.

(3.5) Parameters/Commands of Python IF to control operation of PSIS011. Refer to chapter parameters/commands for more detail.

2.Supported features

Table 2.1 List of supported features in PSIS011 model

Feature	Description	
	Hardware	Model
Operating frequency	- 80 MHz (APB clock): For the operation of APB and peripheral functions - 80 MHz (communication clock): For UART communication and operation - 160 MHz (communication multiply clock): For UART clock generation and GTM interface operation - The frequency ratio of psis_clk to psis_mult_clk must always be 1:2.	- Unlimited frequency - The frequency ratio of psis_clk to psis_mult_clk must always be 1:2.
Synchronous reset (PRESETn)	Synchronous	Asynchronous
UART communication	Sampling clock output: 6.67 MHz to 26.67 MHz Baud rate: 1.333 Mbps to 5.333 Mbps Frame format (total of 10 or 11 bits) - Start bit: 1 bit - Data: 8 bits - Parity: None, even parity, or odd parity (can be specified separately for reception and transmission) - Stop bit: 1 bit	- Due to no limitation about clock frequency. So, sampling clock output, and Baud rate are not limited. - Frame format is same with HW.

Sensor-to-ECU communication	- Possible reception of eight channels of frame data - Possible reception of 8-bit to 28-bit payload - Automatic calculation of CRC and parity values from payload - Possible storage of the CRC and parity bits attached to the data in received frames - Monitoring of the number of packets in received frames - Timestamp function for received messages. - Monitoring of the received frames by WDT.	<-
ECU-to-sensor communication	- Automatic calculation of the CRC value to be added to frame data - Output of the formats pursuant to frame 1 to frame 4	<-
Other interface	APB interface (AMBA APB Protocol Version: 2.0)	- Use TLM socket for APB interface.
	Interrupt output - eight channels (Ch0 to CH7)	<-
	DMA request output - eight channels (Ch0 to CH7) for reception and seven channels (CH1 to CH7) for transmission	<-
Test Functions	- Counter test mode - Status test mode	Not support

Note: “<-”: same as hardware manual

3.Block diagram

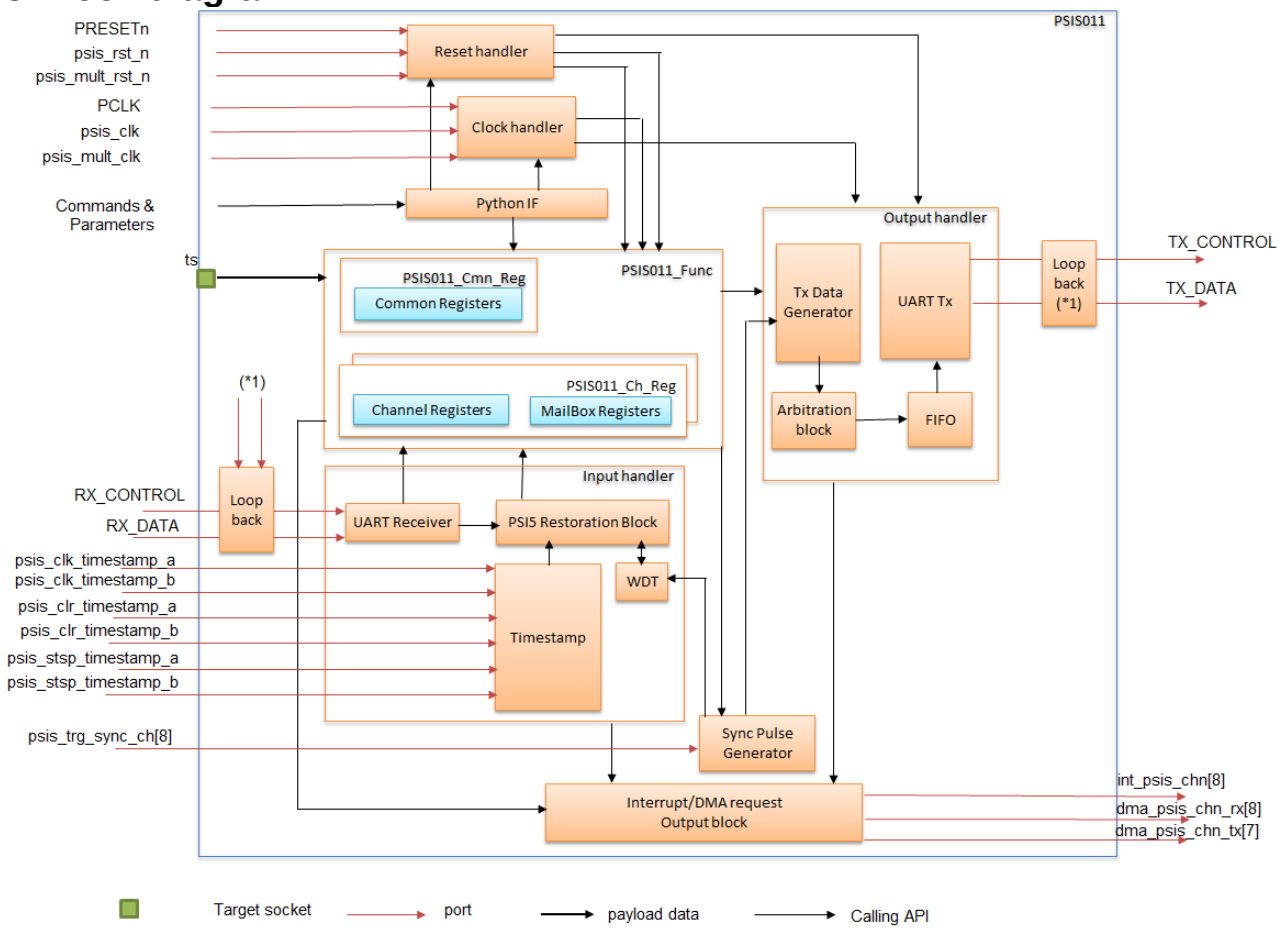


Figure 3.1: Block diagram of PSIS011 model

Explanation:

(1) The PSIS011 model has sub-blocks inside:

(1.1) Clock handler: checks the input clocks; calculate the period of clocks input; calculate clock for transmission/reception based on clock pre-scaler, clock divider.

(1.2) Reset handler: checks the reset ports, the reset command from Python IF to reset according parts.

(1.3) Target socket "ts" to access registers.

(1.4) Python IF: with parameters/commands supports controlling PSIS011.

(1.5) PSIS011_Func has sub-blocks inside to control operation related to registers:

(1.5.1) PSIS011_Cmn_Reg: There is 1 block to handles operations related to common registers.

(1.5.2) PSIS011_Ch_Reg: There are 8 blocks to handles operations related to channel registers, mail box registers of 8 channels.

(1.6) Sync Pulse Generator: generates Sync Pulse signal from input sync trigger or setting of register. This block supplies Sync Pulse to WDT and Tx Data Generator.

(1.7) Input handler has sub-blocks inside to receives input data:

(1.7.1) UART Receiver: receives UART frame, checks parity.

(1.7.2) PSI5 Restoration block: restores PSI5 packet from UART frames, checks CRC, XCRC, and other error.

(1.7.3) Timestamp: gives time stamp to received data.

(1.7.4) WDT: controls maximum period for receiving data.

(1.8) Output handler has sub-blocks inside to output data:

(1.8.1) Tx Data Generator: prepares data for transmission.

(1.8.2) UART Tx: adds parity bit, transmits UART frame via TX_DATA; calculates and sets baud rate to TX_CONTROL.

(1.8.3) Arbitration block: arbitrates requests from channels.

(1.8.4) FIFO: contains transmission requests from channels.

(1.9) Loopback block: When loopback is set, UART transmit data is wrapped.

Note: (*1): Output TX_CONTROL, TX_DATA are loopback to UART receiver.

(1.10) Interrupt/DMA request Output block: controls and issues interrupt/DMA requests.

4.List of registers

(1) Table 4.1 lists all registers in PSIS011 model.

Table 4.1 List of registers in PSIS011 model

Register name	Address Offset	Initial value	Size (Byte)	Write access size (bits)	Read access size (bits)	R/W	Bit position	Bit name	Explanation	Support
ICUS0ICRCMD (ICUS Command register)	Base + 0x0	0x0	4	32	8/16/32	RW	28	AIS31EN	Selects whether to conduct self-diagnosis when using the CMD_INIT_RNG. 0: Self-diagnosis is not conducted. 1: When the CMD_INIT_RNG command is used to generate random numbers, self-diagnosis proceeds simultaneously with random-number generation. Note: This bit exists in this register. But its operation is not supported.	Yes
							20	KEYMD	Specifies whether to select the extended keys when KEY_<n> is used and registered in command execution. This bit is valid for the CMD_ENC_ECB, CMD_ENC_CBC, CMD_DEC_ECB, CMD_DEC_CBC, CMD_GENERATE_MAC, CMD_VERIFY_MAC, and CMD_LOAD_KEY commands. 0: Specifies 1 to 10 for KEY_<n>. 1: Specifies extended keys 11 to 20 for KEY_<n>.	Yes
							19:16	KEYID	Select the key to be used with the command. These bits are valid with the CMD_ENC_ECB, CMD_ENC_CBC, CMD_DEC_ECB, CMD_DEC_CBC, CMD_GENERATE_MAC, and CMD_VERIFY_MAC commands. 2H, 4H to EH: KEY_ID of the key used with the command listed above Other than above: Invalid For the value of KEY_ID, see Section 4.2, KEY_ID. For KEY_ID selectable with respective commands.	Yes

							15:0	CMD	Values written to these bits lead to the execution of commands as listed below. 0000H: Reserved 0001H: CMD_ENC_ECB 0002H: CMD_ENC_CBC 0003H: CMD_DEC_ECB 0004H: CMD_DEC_CBC 0005H: CMD_GENERATE_MAC 0007H: CMD_VERIFY_MAC 0008H: CMD_LOAD_KEY 0009H: CMD_LOAD_PLAIN_KEY 000AH: CMD_EXPORT_RAM_KEY 000BH: CMD_INIT_RNG 000CH: CMD_EXTEND_SEED 000DH: CMD_RND 000EH: CMD_SECURE_BOOT 0020H: CMD_BOOT_FAILURE 0021H: CMD_BOOT_OK 0022H: CMD_GET_ID 0023H: CMD_DEBUG 003FH: CMD_CANCEL 7000H: CHK_VERIFY_MAC1 7100H: CHK_VERIFY_MAC2 Other than above: Setting prohibited Note: Support 7 commands: 0001H: CMD_ENC_ECB 0002H: CMD_ENC_CBC 0003H: CMD_DEC_ECB 0004H: CMD_DEC_CBC 0005H: CMD_GENERATE_MAC 0007H: CMD_VERIFY_MAC 003FH: CMD_CANCEL	Yes
ICUS0ICRIDAT (ICUS Input Data Register)	Base + 0x4	0x0	4	32	8/16/32	RW	31:0	IDAT	These bits are for the input of data such as text and messages for calculation to the PSI5S. Reading these bits returns the value written immediately before. Writing to ICUS0ICRIDAT four times allows the input of 128-bit data. PSI5S starts calculation after 128 bits of data are input. Be sure to write to this register four times. Data to be input to this register are byte-swapped in each word. To input the following 128 bits of data: [A15A14A13A12A11A10A9A8A7A6A5A4A3A2A1A0]H, (where An is the n-th byte of data), the order of writing will be as follows. 1st write: [A12A13A14A15]H 2nd write: [A 8A 9A10A11]H 3rd write: [A4A 5A 6A7]H 4th write: [A0 A1 A2 A3]H When inputting 256-bit data such as M2 of CMD_LOAD_KEY, the input order should be from the upper 128 bits to the lower 128 bits.	Yes

ICUS0ICRODAT (ICUS Output Data Register)	Base + 0x8	0x0	4	32	8/16/32	RW	31:0	ODAT	These bits are for the output of data such as results of calculations from the PSI5S. Reading ICUS0ICRODAT four times allows 128-bit data to be output. The PSI5S starts further processing after 128 bits of data are output. Be sure to read this register four times. Data output by this register are byte-swapped in each word. In response to output of the following 128 bits of data: [A15A14A13A12A11A10A9A8 A7A6A5A4 A3A2A1A0]H, (where An is the n-th byte of data), the order of reading will be as follows. 1st read: [A12A13A14A15]H 2nd read: [A8A9A10A11]H 3rd read: [A4A5A6A7]H 4th read: [A0A1A2A3]H In addition, when outputting 256-bit data such as M4 of CMD_LOAD_KEY, the output order should be from the upper 128 bits to the lower 128 bits.	Yes
ICUS0ICRSTS (ICUS Status Register)	Base + 0xC	0x0	4	-	8/16/32	R	7	INT_DEBUGGER	This bit is set to 1 when execution of CMD_DEBUG is successfully completed.	Yes
							6	EXT_DEBUGGER	This bit is set to 1 when the debugging interface is to be used. This flag still remains 1 even after an external debugger is not connected.	Yes
							5	RND_INIT	This bit is set to 1 when execution of CMD_INIT_RNG is successfully completed. This bit is cleared to 0 when execution of CMD_DEBUG is successfully completed.	Yes
							4	BOOT_OK	This bit is set to 1 when execution of CMD_SECURE_BOOT is successfully completed and comparison with the registered BOOT_MAC is successful. This bit is cleared to 0 when CMD_BOOT_FAILURE is executed.	Yes
							3	BOOT_FINISHED	While SECURE_BOOT = 1, BOOT_FINISHED = 0, and BOOT_OK = 1 and when CMD_BOOT_OK or CMD_BOOT_FAILURE is executed, this bit is set to 1. When CMD_SECURE_BOOT is executed and if its value disagrees with that of BOOT_MAC, this bit is set to 1. When CMD_SECURE_BOOT is executed without BOOT_MAC and if BOOT_MAC_KEY is successfully registered, this bit is set to 1. When CMD_SECURE_BOOT is executed and if ERC_KEY_NOT_AVAILABLE, ERC_MEMORY_FAILURE, or ERC_GENERAL_ERROR occurs during the execution of the command, this bit is set to 1. When CMD_CANCEL is issued during	Yes

									execution of CMD_SECURE_BOOT and processing of the command is cancelled, this bit is set to 1.	
							2	BOOT_INIT	When CMD_SECURE_BOOT is executed without BOOT_MAC and when registration of BOOT_MAC is completed, this bit is set to 1.	Yes
							1	SECURE_BOOT	This bit is set to 1 when verifying BOOT_MAC_KEY is effective after CMD_SECURE_BOOT is executed.	Yes
							0	BUSY	This bit is set to 1 while command execution is started. This bit is cleared to 0 when command execution is completed.	Yes
ICUS0ICRERR (ICUS Error Register)	Base + 0x10	0x0	4	-	8/16/32	R	11	ERR11	When ERC_GENERAL_ERROR is generated, this bit is set to 1.	Yes
							10	ERR10	When ERC_MEMORY_FAILURE is generated, this bit is set to 1.	Yes
							9	ERR9	When ERC_BUSY is generated, this bit is set to 1.	Yes
							8	ERR8	When ERC_NO_DEBUGGING is generated, this bit is set to 1.	Yes
							7	ERR7	When ERC_RNG_SEED is generated, this bit is set to 1.	Yes
							6	ERR6	When ERC_KEY_UPDATE_ERROR is generated, this bit is set to 1.	Yes
							5	ERR5	When ERC_KEY_WRITE_PROTECTED is generated, this bit is set to 1.	Yes
							4	ERR4	When ERC_NO_SECURE_BOOT is generated, this bit is set to 1.	Yes
							3	ERR3	When ERC_KEY_EMPTY is generated, this bit is set to 1.	Yes
							2	ERR2	When ERC_KEY_INVALID is generated, this bit is set to 1.	Yes
							1	ERR1	When ERC_KEY_NOT_AVAILABLE is generated, this bit is set to 1.	Yes
							0	ERR0	When ERC_SEQUENCE_ERROR is generated, this bit is set to 1.	Yes
ICUS0ICRSWINT (ICUS Data Transfer Request Flag Register)	Base + 0x14	0x0	4	-	8/16/32	R	1	RXREQ	This bit is set to 1 when calculation is completed and data is set to ICUS0ICRODAT, thus allowing data to be output. This bit is set to 1 when execution of a command is completed. This bit is cleared to 0 by writing 1 to bit 1 in the ICUS data transfer request flag clear register (ICUS0ICRSWINTCL).	Yes

							0	TXREQ	This bit is set to 1 to make a request for inputting data to ICUS0ICRIDAT for calculation. This bit is cleared to 0 by writing 1 to bit 0 in the ICUS data transfer request flag clear register (ICUS0ICRSWINTCL).	Yes
ICUS0ICRICUSTS (ICUS Enable Status Register)	Base + 0x18	-	4	-	8/16/32	R	-	-	-	No
ICUS0ICRIDATNUM (ICUS Data Transfer Number Register)	Base + 0x1C	0x0	4	32	8/16/32	RW	31:0	IDATNUM	Set the number of data blocks for calculation to be input to the PSI5S. When CMD_ENC_ECB or CMD_DEC_ECB is executed: 1 to 2^32–1: Number of data blocks for transfer When CMD_ENC_CBC or CMD_DEC_CBC is executed: 2 to 2^32–1: Number of data blocks for transfer (1 IV block + (setting value –1) TEXT blocks) Other than above: Setting prohibited	Yes
ICUS0ICRACC (ICUS Region Access Register)	Base + 0x20	-	4	-	8/16/32	R	-	-	-	No
ICUS0ICRSWINTCL (ICUS Data Transfer Request Flag Clear Register)	Base + 0x24	0x0	1	32	8/16/32(*1)	RW(*1)	1	RXREQCLR	Writing 1 to the RXREQCLR bit when bit 1 (RXREQ) in ICUS0ICRSWINT is 1 clears the RXREQ bit to 0. Writing 0 to the RXREQCLR bit does not clear the RXREQ bit.	Yes
							0	TXREQCLR	Writing 1 to the TXREQCLR bit when bit 0 (TXREQ) in ICUS0ICRSWINT is 1 clears the TXREQ bit to 0. Writing 0 to the TXREQCLR bit does not clears the TXREQ bit.	Yes
MALKEYi (ICUS Malfunction Test Key Setting Register i (i = 0 to 3))	Base + 0x90 + 0x4 *i	-	4	8/16/32	8/16/32	RW	-	-	-	No
MALDATi (ICUS Malfunction Test Data Setting Register i (i = 0 to 3))	Base + 0xA0 + 0x4 *i	-	4	8/16/32	8/16/32	RW	-	-	-	No

MALMACi (ICUS Malfunction Test MAC Expected Value Setting Register i (i = 0 to 3))	Base + 0xB0 + 0x4 *i	-	4	8/16/32	8/16/32	RW	-	-	-	No
ICUS0ICRERRCL (ICUS Error Flag Clearing Register)	Base + 0xE4	-	4	8/16/32	8/16/32	RW(*)	-	-	-	No

Register name	Address Offset	Initial value	Size (Byte)	Write access size (bits)	Read access size (bits)	R/W	Bit position	Bit name	Description	Support
Common register/Config										
PSI5SPUOEB (PSI5S/UART Operation Enable Register)	Base + 0x0	0x0	4	8/16/32	8/16/32	RW	0	OPEN	Operation enable 0: Disable 1: Enable This bit can be written 0b1 when PUOS.ACSTS is 0b0 (= configuration mode). This bit can be written 0b0 at any time. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
PSI5SPUOMD (PSI5S/UART Operation Mode Register)	Base + 0x4	0x0	4	8/16/32	8/16/32	RW	0	OPMD	Operation mode 0: Select UART mode 1: Select PSI5S mode This bit can be written when PSI5SPUOS.ACSTS is 1’b0 (= configuration mode). This bit is cleared when writing 1’b1 to PSI5SPUSWR.SWRST.	Yes

PSI5SPUOS (PSI5/UART Operation Status Register)	Base + 0x8	0x0	4	-	8/16/32	R	2	MSTS	Mode status 0: Mode is UART 1: Mode is PSI5S This bit shows mode status. This bit is read only. The write value is ignored. Writing 0b1 to PSI5SPUOEB.OPEN in a state of that PSI5SPUOMD.OPMD is 0b1, this bit is set to 0b1. Writing 0b1 to PSI5SPUOEB.OPEN in a state of that PSI5SPUOMD.OPMD is 0b0, this bit is reset to 0b0. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							1	ACSTS	Active status 0: Not Active (Configuration mode) 1: Active (UART mode or PSI5S mode) This bit shows configuration status. This bit is read only. The write value is ignored. Writing 0b1 to PSI5SPUOEB.OPEN, this bit will be 0b1. 1 shows that module is active. Writing 0b0 to PSI5SPUOEB.OPEN, this bit will be 0b0. 0 shows that module is during configuration mode. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							0	SWSTS	Software (SW) reset status 0: Not SW reset assert 1: During SW reset assert This bit shows software (SW) reset status. This bit is read only. The write value is ignored. Writing 0b1 to PSI5SPUSWR.SWRST, this bit will be 0b1. This bit is read as 0x1 while SW reset execution. After the SW reset execution, the bit is cleared to 0x0.	Yes

PSI5SPUNFST (PSI5S/UART Noise Filter Set Register)	Base + 0xC	0x0	4	8/16/32	8/16/32	RW	0	NFSET	Noise filter setting 0: Disable 1: Enable This bit defines the noise filter enable or disable. Writing 0b0 to this bit sets noise filter to disable. Writing 0b1 to this bit sets noise filter to enable. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
PSI5SPUSWR (PSI5/UART Software Reset Register)	Base + 0x10	0x0	4	8/16/32	8/16/32	RW	0	SWRST	Software (SW) reset 0: Is ignored 1: Software reset start for PSI5-S This bit is always read as 0.	Yes
PSI5SPRMBC (PSI5S Receive MailBox Data Clear Register)	Base + 0x14	0x0	4	8/16/32	8/16/32	RW	0	MBCLR	MailBox clear All Mailbox data clear 0: Is ignored 1: All Mailbox data cleared to 0 This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode).	Yes
PSI5SPUCLB (PSI5S/UART Communication Loop Back Register)	Base + 0x20	0	4	8/16/32	8/16/32	RW	7:1	TMKV	Test Mode Key Values Test mode key values for loopback test function Users are expected to write the test mode key sequence values to these bits. These bits are always read as 0.	Yes
							0	LBEN	Loopback enable Internal loopback enable 3rd write value of loopback test sequence is 0: Disable 3rd write value of loopback test sequence is 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

PSI5SPUPTS (PSI5S/UART Rx/Tx Parity Set Register)	Base + 0x24	0x0	4	8/16/32	8/16/32	RW	9:8	UTPRTY	UART Tx Parity 0: Parity disable 1: Even parity 2: 0 parity (Parity is always 0) 3: Odd parity When CPU sets parity to disable (=0b00), a packet (UART frame) is composed of 1 start bit, 8 data bits and 1 stop bit (10 bits in total). Else, a packet (UART frame) is composed of 1 start bit, 8 data bits, 1 parity bit and 1 stop bit (11 bits in total). These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							1:0	URPRTY	UART Rx parity 0: Parity disable 1: Even parity 2: Parity don’t care 3: Odd parity When CPU sets parity to disable (=0b00), a packet (UART frame) is composed of 1 start bit, 8 data bits and 1 stop bit (10 bits in total). Else, a packet (UART frame) is composed of 1 start bit, 8 data bits, 1 parity bit and 1 stop bit (11 bits in total). These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
PSI5SPUBCE (PSI5S/UART Baud Rate Clock Enable Register)	Base + 0x28	0x0	4	8/16/32	8/16/32	RW	0	SCKEN	Psis_tx_sclk output enable 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

PSI5SPUBPR (PSI5S/UART Baud Rate Parameter Register)	Base + 0x2C	0x4_0000	4	16/32	8/16/32	RW	19:16	RXOSMP	RX Over sample number 0 to 3: Writing prohibited 4: 5samples … x: x+1samples … 15: 16samples Writing 0 to 3 to these bits sets sampling number 5. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							15:8	SCKDIV	SCK clock divide value Clock divide value of UART SCLK(psis_tx_sclk) 0 = 1/1 1 = 1/2 … x = 1/(x+1) … 255 = 1/256 Writing to these bits are prohibited when PSI5SPUBCE.SCKEN is 0b1. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							6:0	SCKPRS	SCK pre-scaler Pre-scaler of UART SCLK(psis_tx_sclk) 0 = 1/1 1 = 1/2 … x = 1/(x+1) … 127= 1/128 Writing to these bits are prohibited when PSI5SPUBCE.SCKEN is 0b1. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
PSI5SPTPS (PSI5S Timestamp Pre-scaler Register)	Base + 0x30	0x0	4	32	8/16/32	RW	25:16	TSPRSU	Timestamp pre-scaler(Upper) Timestamp pre-scaler (Upper) for maximum 1ms enable generate. These bits define the load data of timestamp pre-scale counter (Upper 10 bits). These setting make max 1ms enable from 1us enable. Timestamp pre-scale counter (Upper) loads this setting and starts down count until it'll be 1. Load and down count repeated all the while. When timestamp pre-scale counter (Upper) becomes 1, timestamp tick outputs. If CPU sets 0 to these bits, this pulse isn't output. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							6:0	TSPRSL	Timestamp pre-scaler (Lower) Timestamp pre-scaler (Lower) for 1us enable generate. These bits define the Load data of Timestamp pre-scale counter (Lower 7 bits). Timestamp pre-scale counter (Lower) loads this setting and starts down count until it'll be 1. Load and down count repeated all the while. These setting make 1us clock form PCLK. When PCLK is 80MHz, these bits set 80. When timestamp pre-scale counter (Lower) becomes 1, 1us enable outputs. When CPU sets 0 to these bits, 1us enable isn't output. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
PSI5SPTCAS (PSI5S Timestemp Counter A Select Register)	Base + 0x34	0x0	4	8/16/32	8/16/32	RW	16	TSCACLS	Timestamp counter A clear Select 0: Generate signal of the PSI5-S is selected 1: GTM output is selected Writing 0b0 selects generate signal of the PSI5S module (= PSI5SPTCAC.TSCACLR || PSI5SPATCC.ATSCCLR). Writing 0b1 selects GTM. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							8	TSCAEBS	Timestamp counter A enable select 0: Generate signal of the PSI5-S is selected 1: GTM output is selected Writing 0b0 selects generate signal of the PSI5S module (= PTCAE.TSCAEB || PATCE.ATSCEB). Writing 0b1 selects GTM. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							0	TSCACKS	Timestamp counter A clock select 0: Generate signal of the PSI5-S is selected 1: GTM output is selected Writing 0b0 selects generate signal of the PSI5S. Writing 0b1 selects GTM output select. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
PSI5SPTCBS (PSI5S Timestamp Counter B Select Register)	Base + 0x38	0x0	4	8/16/32	8/16/32	RW	16	TSCBCLS	Timestamp counter B clear Select 0: Generate signal of the PSI5-S is selected 1: GTM output is selected Writing 0b0 selects generate signal of the PSI5S module (= PTCBC.TSCBCLR || PATCC.ATSCCLR). Writing 0b1 selects GTM module output. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							8	TSCBEBS	Timestamp counter B enable select 0: Generate signal of the PSI5-S is selected 1: GTM output is selected Writing 0b0 selects generate signal of the PSI5S module (= PTCBE.TSCABB || PATCE.ATSCEB). Writing 0b1 selects GTM module output. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							0	TSCBCKS	Timestamp counter B clock select 0: Generate signal of the PSI5-S is selected 1: GTM output is selected Writing 0b0 selects generate signal of the PSI5S module. Writing 0b1 selects GTM module output select. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
PSI5SPTCAE (PSI5S Timestamp Counter A Enable Register)	Base + 0x40	0x0	4	8/16/32	8/16/32	RW	0	TSCAEB	Timestamp counter A enable 0: Disable 1: Enable When PSI5SPTCAS.TSCAEBS (=select signal) is 0b0, this signal is selected and is used for timestamp counterA. When PSI5SPTCAS.TSCAEBS (=select signal) is 0b1, this signal isn't used. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

PSI5SPTCAC (PSI5S Timestamp Counter A Clear Register)	Base + 0x44	0x0	4	8/16/32	8/16/32	RW	0	TSCACLR	Timestamp counter A clear 0: Is ignored 1: Clears timestamp counterA When PSI5SPTCAS.TSCACLS (=select signal) is 0b0, this setting selects and uses timestamp counterA clear. When PSI5SPTCAS.TSCACLS (=select signal) is 0b1, this signal isn't used. This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes
PSI5SPTCBE (PSI5S Timestamp Counter B Enable Register)	Base + 0x48	0x0	4	8/16/32	8/16/32	RW	0	TSCBEB	Timestamp counter B enable 0: Disable 1: Enable When PSI5SPTCBS.TSCBEBS (=select signal) is 0b0, this signal is selected and is used for timestamp counterB. When PSI5SPTCBS.TSCBEBS (=select signal) is 0b1, this signal isn't used. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
PSI5SPTCBC (PSI5S Timestamp Counter B Clear Register)	Base + 0x4C	0x0	4	8/16/32	8/16/32	RW	0	TSCBCLR	Timestamp counter B clear 0: Is ignored 1: Clears timestamp counter B When PSI5SPTCBS.TSCBCLS (=select signal) is 0b0, this setting selects and uses timestamp counterB clear. When PSI5SPTCBS.TSCBCLS (=select signal) is 0b1, this signal isn't used. This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes

PSI5SPATCE (PSI5S All Timestamp Counter Enable Register)	Base + 0x50	0x0	4	8/16/32	8/16/32	RW	0	ATSCEB	All timestamp counter enable 0: Is ignored Timestamp counterA'enable follows PSI5SPTCAE.TSCAEB Timestamp counterB'enable follows PSI5SPTCBE.TSCBEB 1: All timestamp counter enable This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
PSI5SPATCC (PSI5S All Timestamp Counter Clear Register)	Base + 0x54	0x0	4	8/16/32	8/16/32	RW	0	ATSCCLR	All Timestamp Counters Clear 0: Is ignored 1: Clears all Timestamp counter When PSI5SPTCAS.TSCACLS (=select signal) is 0b0, writing 0b1 to this bit clears timestamp counterA. When PSI5SPTCAS.TSCACLS (=select signal) is 0b1, this signal isn't used for timestamp counterA. When PSI5SPTCBS.TSCBCLS (=select signal) is 0b0, writing 0b1 to this bit clears timestamp counterB. When PSI5SPTCBS.TSCBCLS (=select signal) is 0b1, this signal isn't used for timestamp counterB. This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes

PSI5SUCRIE (UART Communication Rx Interrupt Enable)	Base + 0x58	0x0	4	8/16/32	8/16/32	RW	3	IERFIN	Interrupt enable Rx finish Interrupt enable of UART Rx finish flag 0: Disable 1: Enable This bit defines the enable of CPU interruption which occurs by “UART frame receiving normal finish in UART mode (PSI5SUCRS.UTRFIN)”. Writing 0b0 to this bit sets interruption by PSI5SUCRS.UTRFIN to disable. Writing 0b1 to this bit sets interruption by PSI5SUCRS.UTRFIN to enable. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b0 (= UART mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							2	IEROE	Interrupt enable Rx overrun error flag Interrupt enable of UART Rx overrun error flag 0: Disable 1: Enable This bit defines the enable of CPU interruption which occurs by “Rx overrun error” of the UART frame in UART mode (PSI5SUCRS.UTROE)”. Writing 0b0 to this bit sets interruption by PSI5SUCRS.UTROE to disable. Writing 0b1 to this bit sets interruption by PSI5SUCRS.UTROE to enable. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b0 (= UART mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							1	IERFE	Interrupt enable Rx framing error flag Interrupt enable of UART Rx framing error flag 0: Disable 1: Enable This bit defines the enable of CPU interruption which occurs by “Rx framing error of the UART frame in UART mode (PSI5SUCRS.UTRFE)”. Writing 0b0 to this bit sets interruption by PSI5SUCRS.UTRFE to disable. Writing 0b1 to this bit sets interruption by PSI5SUCRS.UTRFE to enable. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b0 (= UART mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							0	IERPE	Interrupt enable Rx parity error flag Interrupt enable of UART Rx parity error flag 0: Disable 1: Enable This bit defines the enable of CPU interruption which occurs by “Rx parity error of the UART frame in UART mode (PSI5SUCRS.UTRPE)”. Writing 0b0 to this bit sets interruption by PSI5SUCRS.UTRPE to disable. Writing 0b1 to this bit sets interruption by PSI5SUCRS.UTRPE to enable. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b0 (= UART mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

PSI5SUCTIE (UART Communication Tx Interrupt Enable)	Base + 0x5C	0x0	4	8/16/32	8/16/32	RW	1	IETFIN	Interrupt enable Tx finish flag Interrupt enable of UART Tx finish flag 0: Disable 1: Enable This bit defines the enable of CPU interruption which occurs by “UART frame transmission finish in UART mode (PSI5SUCRS.UTTFIN)”. Writing 0b0 to this bit sets interruption by PSI5SUCRS.UTTFIN to disable. Writing 0b1 to this bit sets interruption by PSI5SUCRS.UTTFIN to enable. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b0 (= UART mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							0	IETOWE	Interrupt enable Tx overwrite error flag Interrupt enable of UART Tx overwrite error flag 0: Disable 1: Enable This bit defines the enable of CPU interruption which occurs by “Tx overwrite error” of the UART frame in UART mode (PSI5SUCRS.UTTFOWE)”. Writing 0b0 to this bit sets interruption by PSI5SUCRS.UTTFOWE to disable. Writing 0b1 to this bit sets interruption by PSI5SUCRS.UTTFOWE to enable. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b0 (= UART mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

PSI5SUCDRE (UART Communication DMA Request Enable Register)	Base + 0x60	0x0	4	8/16/32	8/16/32	RW	1	DRQEUTFN	DMA request enable at UART Tx finish Enable of DMA request by the UART Tx finish 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b0 (= UART mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							0	DRQEURFN	DMA request enable at UART Rx finish Enable of DMA request by the UART Rx finish 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b0 (= UART mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
PSI5PSLT (PSI5S Self-Test Register)	'	'	'			'	'	-	-	No
Common Register/Rx										
PSI5SUCRD (UART Communication Rx Data Register)	Base + 0x70	0x0	4	'	8/16/32	R	7:0	UTRDT	UART read data These bits show Rx UART frame data. CPU can read a stored Rx UART frame data from this address. When Rx UART frame's stop bit is detected in UART mode, PSI5SUCRD.UTRDT is stored. These bits are read only. The write value is ignored. When next UART frame finish CPU has not read this address yet, PSI5SUCRS.UTROE is set to 0b1 and this register is written new data. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

PSI5SUCRS (UART Communication Rx Status Register)	Base + 0x74	0x0	4	-	8/16/32	R	3	UTRFIN	UART Rx finish flag 0: A frame is not received successfully 1: A frame is received successfully This bit shows for UART finish flag. When UART frame's stop bit is detected in UART mode without any errors, this bit is set to 0b1 and an interruption (int_psis_ch0) and DMA request (dma_psis_ch7_rx) occur. This bit is read only. The write value is ignored. This bit is cleared when writing 0b1 to PSI5SUCRSC.UTRFINCL. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes
							2	UTROE	UART Rx overrun error flag 0: No error 1: An overrun error detected This bit shows for UART Rx overrun error. When next UART frame's stop bit is detected in UART mode, CPU has not read PSI5SUCRD.UTRDT yet, this bit is set to 0b1 and interrupt (int_psis_ch0) happen. This bit is read only. The write value is ignored. This bit is cleared when writing 0b1 to PSI5SUCRSC.UTROECL. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes

							1	UTRFE	UART Rx framing error flag 0: No error 1: A framing error detected This bit shows for UART Rx framing error. When UART frame's stop bit is detected in UART mode detecting bit value is 0b0, this bit is set to 0b1 and an interruption (int_psis_ch0) occurs. This bit is read only. The write value is ignored. This bit is cleared when writing 0b1 to PSI5SUCRSC.UTRFECL. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes
							0	UTRPE	UART Rx parity error 0: No error 1: A parity error detected This bit shows for UART Rx parity error. When UART frame's stop bit is detected timing in UART mode parity error was detected, this bit is set to 0b1 and interrupt (int_psis_ch0) happen. This bit is cleared when writing 0b1 to PSI5SUCRSC.UTRPECL. This bit is read only. The write value is ignored. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes
PSI5SUCRSC (UART Communication Rx Status Clear Register)	Base + 0x78	0x0	4	8/16/32	8/16/32	RW	3	UTRFINCL	UART Rx finish flag clear 0: Is ignored 1: UART Rx finish flag (PSI5SUCRS.UTRFIN) clear This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b0 (= UART mode).	Yes

							2	UTROECL	UART Rx overrun error clear 0: Is ignored 1: Overrun error (PSI5SUCRS.UTROE) clear This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b0 (= UART mode).	Yes
							1	UTRFECL	UART Rx framing error clear 0: Is ignored 1: Framing error (PSI5SUCRS.UTRFE) clear This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b0 (= UART mode).	Yes
							0	UTRPECL	UART Rx parity error clear 0: Is ignored 1: Parity error (PSI5SUCRS.UTRPE) clear This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b0 (= UART mode).	Yes
Common Register/Tx										
PSI5SPTFST (PSI5S Tx Frame Start Register)	Base + 0x80	0x0	4	8/16/32	8/16/32	RW	0	TXST	Tx start PSI5S command data transmission start 0: Is ignored 1: Transmission start When CPU writes 0b1 to this bit, 1-8 command data (PSI5SPTFD1.TDT1 to PTFD2.TDT8) transmission starts. This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes

PSI5SPTFNM (PSI5S Tx Frame Number Register)	Base + 0x84	0x0	4	8/16/32	8/16/32	RW	2:0	TXNUM	Tx command data number 0: 1 packet (UART frame) 1: 2 packet (UART frame) 2: 3 packet (UART frame) 3: 4 packet (UART frame) 4: 5 packet (UART frame) 5: 6 packet (UART frame) 6: 7 packet (UART frame) 7: 8 packet (UART frame) These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
PSI5SPTFD1 (PSI5S Tx Frame Data1 Register)	Base + 0x88	0x0	4	8/16/32	8/16/32	RW	31:24	TDT4	Transmission data of 4th packet (UART frame) These bits define the transmission data of 4th packet (UART frame). When PSI5SPTFNM.TXNUM is less than 3, this setting is ignored. These bits cannot be written when PSI5SPTFS.TXSTS is 0b1. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							23:16	TDT3	Transmission data of 3rd packet (UART frame) These bits define the transmission data of 3rd packet (UART frame). When PSI5SPTFNM.TXNUM is less than 2, this setting is ignored. These bits cannot be written when PSI5SPTFS.TXSTS is 0b1. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							15:8	TDT2	Transmission data of 2nd packet (UART frame) These bits define the transmission data of 2nd packet (UART frame). When PSI5SPTFNM.TXNUM is less than 1, this setting is ignored. These bits cannot be written when PSI5SPTFS.TXSTS is 0b1. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							7:0	TDT1	Transmission data of 1st packet (UART frame) These bits define the transmission data of 1st packet (UART frame). These bits cannot be written when PSI5SPTFS.TXSTS is 0b1. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
PSI5SPTFD2 (PSI5S Tx Frame Data2 Register)	Base + 0x8C	0x0	4	8/16/32	8/16/32	RW	31:24	TDT8	Transmission data of 8th packet These bits define the transmission data of 8th packet. When PSI5SPTFNM.TXNUM is less than 7, these setting are ignored. These bits cannot be written when PSI5SPTFS.TXSTS is 0b1. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							23:16	TDT7	Transmission data of 7th packet These bits define the transmission data of 7th packet. When PSI5SPTFNM.TXNUM is less than 6, these setting are ignored. These bits cannot be written when PSI5SPTFS.TXSTS is 0b1. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							15:8	TDT6	Transmission data of 6th packet These bits define the transmission data of 6th packet. When PSI5SPTFNM.TXNUM is less than 5, these setting are ignored. These bits cannot be written when PSI5SPTFS.TXSTS is 0b1. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							7:0	TDT5	Transmission data of 5th packet These bits define the transmission data of 5th packet. When PSI5SPTFNM.TXNUM is less than 4, these setting are ignored. These bits cannot be written when PSI5SPTFS.TXSTS is 0b1. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

PSI5SPTFS (PSI5S Tx Frame Status Register)	Base + 0x90	0x0	4	-	8/16/32	R	0	TXSTS	Tx status 0: Transmission not busy 1: Transmission busy This bit shows Tx status. When CPU sets 0b1to PSI5SPTFST.TXST, this is set to 0b1. When all transmission data (PSI5SPTFD1, PSI5SPTFD2) send to Tx Shifter, this bit reset to 0b0. When this bit is 0b1, PSI5SPTFD1 and PSI5SPTFD2 cannot be written. This bit is read only. The write value is ignored. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes
PSI5SPTFIS (PSI5S Tx FIFO Status Register)	Base + 0x94	0x1	4	-	8/16/32	R	1	TXFFFL	Tx fifo full 0: Not full 1: Full This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes
							0	TXFFEP	Tx fifo empty 0: Not empty 1: Empty This bit is set to 0b1 when writing 0b1 to PSI5SPUSWR.SWRST. This bit is set to 0b1 when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes

PSI5SUCTD (UART Communication Tx Data Register)	Base + 0xA0	0x0	4	8/16/32	8/16/32	RW	7:0	UTTDT	UART transmission data These bits define the transmission data of UART frame. When CPU writes these bits in UART mode, transmission of UART frame will start. These bits cannot be written when PSI5SUCTM.UTTBBF is 0b1. When CPU writes PSI5SUCTD.UTTDT in PSI5SUCTM.UTTBBF is 0b1, PSI5SUCTS.UTTOWE sets to 0b1. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b0 (= UART mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
PSI5SUCTM (UART Communication Tx Monitoring Register)	Base + 0xA4	0x0	4	-	8/16/32	R	1	UTTF	UART transmission flag 0: Not transmitting 1: Transmitting This bit shows status of Tx transmission in UART mode. When UART start bit transmission starts, this bit is set to 0b1. When stop bit output end without next write data, this bit reset to 0b0. This bit is read only. The write value is ignored. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes

							0	UTTBBF	UART Tx shifter busy flag 0: Permit to write to PSI5SUCTD.UTTDT 1: Prohibited to write to PSI5SUCTD.UTTDT This bit shows status of Tx shifter busy in UART mode. When CPU writes PSI5SUCTD.UTTDT, this bit is set to 0b1. When PSI5SUCTD.UTTDT data is stored to Tx shifter, this bit reset to 0b0. When this bit is 0b1, PSI5SUCTD.UTTDT cannot be written. This bit is read only. The write value is ignored. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes
PSI5SUCTS (UART Communication Tx Status Register)	Base + 0xA8	0x0	4	-	8/16/32	R	1	UTTFIN	UART Tx finish UART transmission finish 0: Transmission not finish 1: A UART frame transmission finish This bit shows UART Tx frame finish status in UART mode. When PSI5SUCTD.UTTDT data send to Tx shifter, this bit is set to 0b1 and an interruption (int_psis_ch1) and DMA request (dma_psis_ch7_tx) occur. This bit is read only. The write value is ignored. This bit is cleared when writing 0b1 to PSI5SUCTSC.UTTFINCL. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes

							0	UTTOWE	UART Tx overwrite error UART transmission overwrite error 0: No error 1: Overwrite error This bit shows for overwrite error status of UART transmission. When CPU writes PSI5SUCTD.UTTDT in PSI5SUCTM.UTTBBF is 0b1, this bit sets to 0b1 and an interruption (int_psis_ch1) occurs. This bit is read only. The write value is ignored. This bit is cleared when writing 0b1 to PSI5SUCTSC.UTTOWECL. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes
PSI5SUCTSC (UART Communication Tx Status Clear Register)	Base + 0xAC	0x0	4	8/16/32	8/16/32	RW	1	UTTFINCL	UART Tx finish flag clear Clear for UART Tx finish flag 0: Is ignored 1: UART Tx finish flag (PSI5SUCTS.UTTFIN) clear This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b0 (= UART mode).	Yes
							0	UTTOWECL	UART Tx overwrite error clear Clear for UART Tx overwrite error 0: Is ignored 1: Overwrite error (PSI5SUCTS.UTTOWE) clear This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b0 (= UART mode).	Yes
Common Register/Test ==> Not support test mode.										
Ch0 Register/Config										

PSI5SPRCF10 (PSI5S Receive Config1 ch0 Register)	Base + 0x100	0x0	4	32	8/16/32	RW	27:24	PFRMIDLE	Packet frame idle Minimum packet frame gap (all channel) 0 : 1 gap is judged as the next packet frame 1: 2 gap is judged as the next packet frame ... n: n+1 gap is judged as the next packet frame ... 15: 16 gap is judged as the next packet frame This setting is effective all channels. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							23:21	F6PKT	Frame 6 packet Frame 6 packet number (Ch0) 0: Frame 6 data is ignored 3: Packet number is set 3 4: Packet number is set 4 5: Packet number is set 5 6: Packet number is set 6 1, 2, 7: Writing prohibited These bits define the packet number of packet frame 6 in Ch0. The packet number means UART frame number per packet frame. "Frame 6" is packet frame that FID is 0b101. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							20:18	F5PKT	Frame 5 packet Frame 5 packet number (Ch0) 0: Frame 5 data is ignored 3: Packet number is set 3 4: Packet number is set 4 5: Packet number is set 5 6: Packet number is set 6 1, 2, 7: Writing prohibited These bits define the packet number of packet frame 5 in Ch0. The packet number means UART frame number per packet frame. "Frame 5" is packet frame that FID is 0b100. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							17:15	F4PKT	Frame 4 packet Frame 4 packet number (Ch0) 0: Frame 4 data is ignored 3: Packet number is set 3 4: Packet number is set 4 5: Packet number is set 5 6: Packet number is set 6 1, 2, 7: Writing prohibited These bits define the packet number of packet frame 4 in Ch0. The packet number means UART frame number per packet frame. "Frame 4" is packet frame that FID is 0b011. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							14:12	F3PKT	Frame 3 packet Frame 3 packet number (Ch0) 0: Frame 3 data is ignored 3: Packet number is set 3 4: Packet number is set 4 5: Packet number is set 5 6: Packet number is set 6 1, 2, 7: Writing prohibited These bits define the packet number of packet frame 3 in Ch0. The packet number means UART frame number per packet frame. "Frame 3" is packet frame that FID is 0b010. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							11:9	F2PKT	Frame 2 packet Frame 2 packet number (Ch0) 0: Frame 2 data is ignored 3: Packet number is set 3 4: Packet number is set 4 5: Packet number is set 5 6: Packet number is set 6 1, 2, 7: Writing prohibited These bits define the packet number of packet frame2 in Ch0. The packet number means UART frame numbers per packet frame. "Frame 2" is packet frame that FID is 0b001. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							8:6	F1PKT	Frame 1 packet Frame 1 packet number (Ch0) 0: Frame 2 data is ignored 3: Packet number is set 3 4: Packet number is set 4 5: Packet number is set 5 6: Packet number is set 6 1, 2, 7: Writing prohibited These bits define the packet number of packet frame1 in Ch0. The packet number means UART frame numbers per packet frame. "Frame 1" is packet frame that FID is 0b000. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							3	TSCS	Timestamp counter select Timestamp counter select (Ch0) 0: Timestamp counter B select 1: Timestamp counter A select This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							2	TSEN	Timestamp enable Timestamp capture enable (Ch0) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							1	RFCPS	Rx Frame checksum crc/parity select Rx frame checksum CRC/parity select (Ch0) 0: Parity select 1: CRC select This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							0	CHEN	Channel enable Channel enable (Ch0) 0: Disable 1: Enable This bit is invalid to "Frame lack error" and "Frame excess error". (Even when this bit is 0b0, "Frame lack error" and "Frame excess error" are not disable.) This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
PSI5SPRCF20 (PSI5S Receive Config2 ch0 Register)	Base + 0x104	0x0	4	32	8/16/32	RW	29:25	F6PAYLD	Frame6 payload The number of packet frame6's payload (Ch0) 0 to 7: Writing prohibited 8: Sets 8 … x: Sets x … 28: Sets 28 29 to 31: Writing prohibited Initial value (0) to these bits are handled as 8. Writing less than 8 to these bits are handled as 8. Writing more than 28 to these bits are handled as 28. These bits are used to decide about the CRC/parity calculation scope. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							24:20	F5PAYLD	Frame5 payload The number of packet frame5's payload (Ch0) 0 to 7: Writing prohibited 8: Sets 8 … x: Sets x … 28: Sets 28 29 to 31: Writing prohibited Initial value (0) to these bits are handled as 8. Writing less than 8 to these bits are handled as 8. Writing more than 28 to these bits are handled as 28. These bits are used to decide about the CRC/parity calculation scope. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							19:15	F4PAYLD	Frame4 payload The number of packet frame4's payload (Ch0) 0 to 7: Writing prohibited 8: Sets 8 … x: Sets x … 28: Sets 28 29 to 31: Writing prohibited Initial value (0) to these bits are handled as 8. Writing less than 8 to these bits are handled as 8. Writing more than 28 to these bits are handled as 28. These bits are used to decide about the CRC/parity calculation scope. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							14:10	F3PAYLD	Frame3 payload The number of packet frame3's payload (Ch0) 0 to 7: Writing prohibited 8: Sets 8 … x: Sets x … 28: Sets 28 29 to 31: Writing prohibited Initial value (0) to these bits are handled as 8. Writing less than 8 to these bits are handled as 8. Writing more than 28 to these bits are handled as 28. These bits are used to decide about the CRC/parity calculation scope. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							9:5	F2PAYLD	Frame2 payload The number of packet frame2's payload (Ch0) 0 to 7: Writing prohibited 8: Sets 8 … x: Sets x … 28: Sets 28 29 to 31: Writing prohibited Initial value (0) to these bits are handled as 8. Writing less than 8 to these bits are handled as 8. Writing more than 28 to these bits are handled as 28. These bits are used to decide about the CRC/parity calculation scope. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							4:0	F1PAYLD	Frame1 payload The number of packet frame1's payload (Ch0) 0 to 7: Writing prohibited 8: Sets 8 … x: Sets x … 28: Sets 28 29 to 31: Writing prohibited Initial value (0) to these bits are handled as 8. Writing less than 8 to these bits are handled as 8. Writing more than 28 to these bits are handled as 28. These bits are used to decide about the CRC/parity calculation scope. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
PSI5SPWDE0 (PSI5S WDT Enable ch0 Register)	Base + 0x108	0x0	4	8/16/32	8/16/32	RW	0	WDTEB	Watch Dog Timer enable Watchdog timer of Rx frame enable (Ch0) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
PSI5SPWDP0 (PSI5S WDT Pre-scaler ch0 Register)	Base + 0x10C	0x0	4	16/32	8/16/32	RW	11:0	WDTPRS	Watch Dog Timer pre-scaler Watchdog timer pre-scaler(Ch0) 0 : Stop Watchdog timer 1 to 4095: Enabled at 1 clock/x clock (x:1 to 4095) These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

PSI5SPWDEV0 (PSI5S WDT Expiration Value ch0 Register)	Base + 0x110	0x0	4	32	8/16/32	RW	23:0	WDTEX	Watch Dog Timer expiration value Watchdog timer expiration value (Ch0) These bits define the expiration value of watchdog timer for Rx packet frame monitor in Ch0. When watchdog counter counts down to 0 from this setting value, watchdog timer is judged as expired. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
PSI5SPCIE0 (PSI5S CPU Interrupt Enable Ch0 Register)	Base + 0x118	0x0	4	16/32	8/16/32	RW	14	IEBCTFN	Interrupt enable command Tx finish CPU Interrupt enable of command tx finish (Ch0) 0: Disable 1: Enable This bit defines the enable of CPU interruption which occurs by “command Tx finish in PSI5S mode (PSI5SPCIS0.ISCTFN)“. Writing 0b0 to this bit sets interruption by PSI5SPCIS0.ISCTFN to disable. Writing 0b1 to this bit sets interruption by PSI5SPCIS0.ISCTFN to enable. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							11	IEBRFN	Interrupt enable Rx frame finish CPU Interrupt enable of Rx packet frame finish flag (Ch0) 0: Disable 1: Enable This bit defines the enable of CPU interruption which occurs by “packet frame receiving end in Ch0 in PSI5S mode (PSI5SPCIS0.ISTRFN)“. Writing 0b0 to this bit sets interruption by PSI5SPCIS0.ISTRFN to disable. Writing 0b1 to this bit sets interruption by PSI5SPCIS0.ISTRFN to enable. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							10	IEBRFEX	Interrupt enable Rx frame excess error CPU Interrupt enable of Rx frame excess error (Ch0) 0: Disable 1: Enable This bit defines the enable of CPU interruption which occurs by “Rx frame excess error in Ch0 in PSI5S mode (PSI5SPCIS0.ISTRFEX)”. Writing 0b0 to this bit sets interruption by PSI5SPCIS0.ISTRFEX to disable. Writing 0b1 to this bit sets interruption by PSI5SPCIS0.ISTRFEX to enable. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							9	IEBRFLK	Interrupt enable Rx frame lack error CPU Interrupt enable of Rx frame lack error (Ch0) 0: Disable 1: Enable This bit defines the enable of CPU interruption which occurs by “Rx frame lack error in Ch0 in PSI5S mode (PSI5SPCIS0.ISTRFLK)”. Writing 0b0 to this bit sets interruption by PSI5SPCIS0.ISTRFLK to disable. Writing 0b1 to this bit sets interruption by PSI5SPCIS0.ISTRFLK to enable. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							8	IEBROV	Interrupt enable Rx overrun error CPU interrupt enable of Rx overrun error (Ch0) 0: Disable 1: Enable This bit defines the enable of CPU interruption which occurs by “Rx overrun error in Ch0 in PSI5S mode (PSI5SPCIS0.ISTROV)”. Writing 0b0 to this bit sets interruption by PSI5SPCIS0.ISTROV to disable. Writing 0b1 to this bit sets interruption by PSI5SPCIS0.ISTROV to enable. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							7	IEBRWDT	Interrupt enable Rx WDT error CPU Interrupt enable of Rx WDT error (Ch0) 0: Disable 1: Enable This bit defines the enable of CPU interruption which occurs by “Rx WDT error in Ch0 in PSI5S mode (PSI5SPCIS0.ISTOV)”. Writing 0b0 to this bit sets interruption by PSI5SPCIS0.ISTRWDT to disable. Writing 0b1 to this bit sets interruption by PSI5SPCIS0.ISTRWDT to enable. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							5	IEBUTFR	Interrupt enable UART framing error CPU interrupt enable of Rx UART framing error (Ch0) 0: Disable 1: Enable This bit defines the enable of CPU interruption which occurs by “UART framing error in Ch0 in PSI5S mode (PSI5SPCIS0.ISTUTFR)”. Writing 0b0 to this bit sets interruption by PSI5SPCIS0.ISTUTFR to disable. Writing 0b1 to this bit sets interruption by PSI5SPCIS0.ISTUTFR to enable. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							4	IEBUTPT	Interrupt enable UART parity error CPU interrupt enable of Rx UART parity error (Ch0) 0: Disable 1: Enable This bit defines the enable of CPU interruption which occurs by “UART parity error in Ch0 in PSI5S mode (PSI5SPCIS0.ISTUTPR)”. Writing 0b0 to this bit sets interruption by PSI5SPCIS0.ISTUTPR to disable. Writing 0b1 to this bit sets interruption by PSI5SPCIS0.ISTUTPR to enable. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							3	IEBTRST	Interrupt enable transceiver status error CPU interrupt enable of Rx transceiver status error (Ch0) 0: Disable 1: Enable This bit defines the enable of CPU interruption which occurs by “Rx transceiver status error in Ch0 in PSI5S mode (PSI5SPCIS0.ISTTRST)”. Writing 0b0 to this bit sets interruption by PSI5SPCIS0.ISTTRST to disable. Writing 0b1 to this bit sets interruption by PSI5SPCIS0.ISTTRST to enable. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							2	IEBPT	Interrupt enable parity error CPU interrupt enable of Rx payload data parity error (Ch0) 0: Disable 1: Enable This bit defines the enable of CPU interruption which occurs by “Rx payload data parity error in Ch0 in PSI5S mode (PSI5SPCIS0.ISTPT)”. Writing 0b0 to this bit sets interruption by PSI5SPCIS0.ISTPT to disable. Writing 0b1 to this bit sets interruption by PSI5SPCIS0.ISTPT to enable. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							1	IEBCRC	Interrupt enable CRC error CPU interrupt enable of Rx payload data CRC error (Ch0) 0: Disable 1: Enable This bit defines the enable of CPU interruption which occurs by “Rx payload data CRC error in Ch0 in PSI5S mode (PSI5SPCIS0.ISTCRC)”. Writing 0b0 to this bit sets interruption by PSI5SPCIS0.ISTCRC to disable. Writing 0b1 to this bit sets interruption by PSI5SPCIS0.ISTCRC to enable. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							0	IEBXCRC	Interrupt enable XCRC error CPU interrupt enable of packet frame XCRC error (Ch0) 0: Disable 1: Enable This bit defines the enable of CPU interruption which occurs by “Rx packet frame XCRC error in Ch0 in PSI5S mode (PSI5SPCIS0.ISTXCRC)”. Writing 0b0 to this bit sets interruption by PSI5SPCIS0.ISTXCRC to disable. Writing 0b1 to this bit sets interruption by PSI5SPCIS0.ISTXCRC to enable. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
PSI5SPDRE0 (PSI5S DMA Transfer Request Enable ch0 Register)	Base + 0x11C	0x0	4	32	8/16/32	RW	1	DRQEWDT	DMA request enable at WDT Enable of DMA request by the WDT error (Ch0) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							0	DRQERFN	DMA request enable at Rx finish Enable of DMA request by the channel data Rx finish (Ch0) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
Ch0 Register/Rx										

PSI5SPRES0 (PSI5S Receive Error Status ch0 Register)	Base + 0x130	0x0	4	-	8/16/32	R	1	RERRF2	Rx error frame2 Rx error at packet frame 2 (Ch0) 0: No error 1: An error has occurred This bit shows PSI5-S Rx error status at packet frame2 of Ch0. This bit is set to 0b1 when any of “Rx overrun error”, “Rx WDT error”, “UART framing error”, “UART parity error”, “transceiver status error” , ”payload data parity error”, ”payload data CRC error” and “packet frame XCRC error” in packet frame2 of Ch0 have occurred. This bit is read only. The write value is ignored. This bit is cleared when writing 0b1 to PSI5SPRESC0.RERRCLF2. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes
							0	RERRF1	Rx error frame1 Rx error at packet frame 1 (Ch0) 0: No error 1: An error has occurred This bit shows PSI5-S Rx error status at packet frame1 in Ch0. This bit is set to 0b1 when any of “Rx overrun error”, “transceiver status error” , ”payload data parity error”, ”payload data CRC error” in packet frame1 of Ch0 have occurred. This bit is read only. The write value is ignored. This bit is cleared when writing 0b1 to PSI5SPRESC0.RERRCLF1. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes

PSI5SPRESC0 (PSI5S Receive Error Status Clear ch0 Register)	Base + 0x134	0x0	4	8/16/32	8/16/32	RW	1	RERRCLF2	Rx error clear frame2 Rx error clear for packet Frame2 (Ch0) 0: Is ignored 1: Clears PSI5SPRES0.RERRF2 This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes
							0	RERRCLF1	Rx error clear frame1 Rx error clear for packet Frame1 (Ch0) 0: Is ignored 1: Clears PSI5SPRES0.RERRF1 This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes
PSI5SPTCDT0 (PSI5S Timestamp Capture Data ch0 Register)	Base + 0x138	0x0	4	-	8/16/32	R	23:0	TSCD	Timestamp capture data Timestamp capture data (Ch0) These bits show the timestamp capture value in Ch0. These bits are cleared when writing 0b1 to PSI5SPTCDC0.TSCCLR. These bits are cleared when writing 0b0 to PSI5SPRCF10.TSEN. These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
PSI5SPTCDC0 (PSI5S Timestamp Capture Data Clear ch0 Register)	Base + 0x13C	0x0	4	8/16/32	8/16/32	RW	0	TSCCLR	Timestamp capture clear Timestamp capture clear (Ch0) 0: Is ignored 1: Clears timestamp capture This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes
Ch0 Register/Interrupt										

PSI5SPCIS0 (PSI5S CPU Interrupt Status ch0 Register)	Base + 0x150	0x0	4	-	8/16/32	R	14	ISTCTFN	Interrupt status command Tx finish CPU interrupt status of command Tx finish 0: PSI5S command transmission is not finish 1: PSI5S command transmission finish This bit shows the status of CPU interruption which occurs by command Tx finish in PSI5S mode. When the last command sent to Tx Shifter frame is stored MB PSI5S frame has no errors, this bit is set to 0b1 and an interruption (int_psis_ch0) occurs. This bit is read only. The write value is ignored. This bit is cleared when writing 0b1 to PSI5SPCISC0.ISTCCTFN. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes
							11	ISTRFN	Interrupt status Rx finish CPU interrupt status of Rx finish (Ch0) 0: PSI5S frame is not received successfully 1: PSI5S frame is received successfully This bit shows the status of CPU interruption which occurs by Rx frame finish in PSI5S mode. When the PSI5S frame is stored MB PSI5S frame has no errors (without mailbox overrun error), this bit is set to 0b1 and an interruption (int_psis_ch0) and DMA request (dma_psis_ch0_rx) occur. This bit is read only. The write value is ignored. This bit is cleared when writing 0b1 to PSI5SPCISC0.ISTCRFN. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes

							10	ISTRFEX	Interrupt status Rx frame excess error CPU interrupt status of Rx frame excess error (Ch0) 0: No error 1: Error detected This bit shows the status of CPU interruption which occurs by Rx frame (packet) excess error in PSI5S mode. When packet is received over PSI5SPRCF10.FmPKT (m = 1 to 6), this bit is set to 0b1 and an interruption (int_psis_ch0) occurs. This bit is read only. The write value is ignored. This bit is cleared when writing 0b1 to PSI5SPCISC0.ISTCRFEX. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes
							9	ISTRFLK	Interrupt status Rx frame lack error CPU interrupt status of Rx frame lack error (Ch0) 0: No error 1: Error detected This bit shows the status of CPU interruption which occurs by Rx frame (packet) lack error in PSI5S mode. When a packet frame gap is detected in the state for which packet is lack of PSI5SPRCF10.FmPKT (m=1 to 6), this bit is set to 0b1 and an interruption (int_psis_ch0) occurs. This bit is read only. The write value is ignored. This bit is cleared when writing 0b1 to PSI5SPCISC0.ISTCRFLK. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes

							8	ISTROV	Interrupt status Rx overrun error CPU interrupt status of Rx overrun error (Ch0) 0: No error 1: Error detected This bit shows the status of CPU interruption which occurs by overrun error in PSI5S mode. When next PSI5S frame is stored to same MB before CPU read MB or set PSI5SPRMBC.MBCLR, this bit is set to 0b1 and an interruption (int_psis_ch0) occurs. This bit is read only. The write value is ignored. This bit is cleared when writing 0b1 to PSI5SPCISC0.ISTCROV. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes
							7	ISTRWDT	Interrupt status Rx WDT error CPU interrupt status of Rx WDT error (Ch0) 0: No error 1: Error detected This bit shows the status of CPU interruption which occurs by WDT error in PSI5S mode. When WDT error occurs in PSI5S mode, this bit is set to 0b1 and an interruption (int_psis_ch0) and DMA request (dma_psis_ch0_rx) occur. This bit is read only. The write value is ignored. This bit is cleared when writing 0b1 to PSI5SPCISC0.ISTCRWDT. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes

							5	ISTUTFR	Interrupt status UART framing error CPU interrupt status of Rx UART framing error (Ch0) 0: No error 1: Error detected This bit shows the status of CPU interruption which occurs by UART framing error in PSI5S mode. When UART frame's stop bit is detected in PSI5S mode detecting bit value is 0b0, this bit is set to 0b1 and an interruption (int_psis_ch0) occurs. This bit is read only. The write value is ignored. This bit is cleared when writing 0b1 to PSI5SPCISC0.ISTCUTFR. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes
							4	ISTUTPT	Interrupt status UART parity error CPU interrupt status of Rx UART parity error (Ch0) 0: No error 1: Error detected This bit shows the status of CPU interruption which occurs by UART parity error in PSI5S mode. When UART frame's stop bit is detected timing in PSI5S mode UART parity error occurred, this bit is set to 0b1 and an interruption (int_psis_ch0) occurs. This bit is read only. The write value is ignored. This bit is cleared when writing 0b1 to PSI5SPCISC0.ISTCUTPT. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes

							3	ISTTRST	Interrupt status transceiver status error CPU interrupt status of Rx transceiver status error (Ch0) 0: No error 1: Error detected This bit shows the status of CPU interruption which occurs by transceiver status error in PSI5S mode. When the PSI5S frame is stored MB transceiver status error occurred, this bit is set to 0b1 and an interruption (int_psis_ch0) occurs. This bit is read only. The write value is ignored. This bit is cleared when writing 0b1 to PSI5SPCISC0.ISTCTRST. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes
							2	ISTPT	Interrupt status parity error CPU interrupt status of payload data parity error (Ch0) 0: No error 1: Error detected This bit shows the status of CPU interruption which occurs by payload data parity error in PSI5S mode. When the PSI5S frame is stored to MB and payload data parity error occurred, this bit is set to 0b1 and an interruption (int_psis_ch0) occurs. This bit is read only. The write value is ignored. This bit is cleared when writing 0b1 to PSI5SPCISC0.ISTCPT. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes

							1	ISTCRC	Interrupt status CRC error CPU interrupt status of payload data CRC error (Ch0) 0: No error 1: Error detected This bit shows the status of CPU interruption which occurs by payload data CRC error in PSI5S mode. When the PSI5S frame is stored MB and payload data CRC error occurred, this bit is set to 0b1 and an interruption (int_psis_ch0) occurs. This bit is read only. The write value is ignored. This bit is cleared when writing 0b1 to PSI5SPCISC0.ISTCCRC. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes
							0	ISTXCRC	Interrupt status XCRC error CPU interrupt status of packet frame XCRC error (Ch0) 0: No error 1: Error detected This bit shows the status of CPU interruption which occurs by packet frame XCRC error in PSI5S mode. When the PSI5S frame is stored MB XCRC error occurred, this bit is set to 0b1 and an interruption (int_psis_ch0) occurs. This bit is read only. The write value is ignored. This bit is cleared when writing 0b1 to PSI5SPCISC0.ISTCXCRC. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes

PSI5SPCISC0 (PSI5S CPU Interrupt Status Clear ch0 Register)	Base + 0x154	0x0	4	16/32	8/16/32	RW	14	ISTCCTFN	Interrupt status clear command Tx finish Clear at CPU interrupt status of command Tx finish 0: Is ignored 1: Clears PSI5SPCIS0.ISCTFN This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes
							11	ISTCRFN	Interrupt status clear Rx finish Clear at CPU interrupt status of Rx finish (Ch0) 0: Is ignored 1: Clears PSI5SPCIS0.ISTRFN This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes
							10	ISTCRFEX	Interrupt status clear Rx frame excess error Clear at CPU interrupt status of Rx frame excess error (Ch0) 0: Is ignored 1: Clears PSI5SPCIS0.ISTRFEX This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes
							9	ISTCRFLK	Interrupt status clear Rx frame lack error Clear at CPU interrupt status of Rx frame lack error (Ch0) 0: Is ignored 1: Clears PSI5SPCIS0.ISTRFLK This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes

							8	ISTCROV	Interrupt status clear Rx overrun error Clear at CPU interrupt status of Rx overrun error (Ch0) 0: Is ignored 1: Clears PSI5SPCIS0.ISTROV This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes
							7	ISTCRWDT	Interrupt status clear Rx WDT error Clear at CPU interrupt status of Rx WDT error (Ch0) 0: Is ignored 1: Clears PSI5SPCIS0.ISTRWDT This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes
							5	ISTCUTFR	Interrupt status clear UART framing error Clear at CPU interrupt status of Rx UART framing error (Ch0) 0: Is ignored 1: Clears PSI5SPCIS0.ISTUTFR This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes
							4	ISTCUTPT	Interrupt status clear UART parity error Clear at CPU interrupt status of Rx UART parity error (Ch0) 0: Is ignored 1: Clears PSI5SPCIS0.ISTUTPT This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes

							3	ISTCTRST	Interrupt status clear transceiver status error Clear at CPU interrupt status of Rx transceiver status error (Ch0) 0: Is ignored 1: Clears PSI5SPCIS0.ISTTRST This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes
							2	ISTCPT	Interrupt status clear parity error Clear at CPU interrupt status of payload data parity error (Ch0) 0: Is ignored 1: Clears PSI5SPCIS0.ISTPT This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes
							1	ISTCCRC	Interrupt status clear CRC error Clear at CPU interrupt status of payload data CRC error (Ch0) 0: Is ignored 1: Clears PSI5SPCIS0.ISTCRC This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes
							0	ISTCXCRC	Interrupt status clear XCRC error Clear at CPU interrupt status of packet frame XCRC error (Ch0) 0: Is ignored 1: Clears PSI5SPCIS0.ISTXCRC This bit is always read as 0. This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes
Ch0 Register/Test ==> Not support test mode										
Ch n Register/Config (n: 1 to 7)										

PSI5SPRCF1n (PSI5S Receive Config1 chn Register)	Base + 0x180 + (n-1)*0x80	0x0	4	32	8/16/32	RW	23:21	F6PKT	Frame 6 Packet Frame 6 packet number (Ch n) 0: Frame 6 data is ignored 3: Packet number is set 3 4: Packet number is set 4 5: Packet number is set 5 6: Packet number is set 6 1, 2, 7: Writing prohibited These bits define the packet number of packet frame 6 in Ch n. The packet number means UART frame number per packet frame. "Frame 6" is packet frame that FID is 0b101. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							20:18	F5PKT	Frame 5 Packet Frame 5 packet number (Ch n) 0: Frame 5 data is ignored 3: Packet number is set 3 4: Packet number is set 4 5: Packet number is set 5 6: Packet number is set 6 1, 2, 7: Writing prohibited These bits define the packet number of packet frame 5 in Ch n. The packet number means UART frame number per packet frame. "Frame 5" is packet frame that FID is 0b100. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							17:15	F4PKT	Frame 4 Packet Frame 4 packet number (Ch n) 0: Frame 4 data is ignored 3: Packet number is set 3 4: Packet number is set 4 5: Packet number is set 5 6: Packet number is set 6 1, 2, 7: Writing prohibited These bits define the packet number of packet frame 4 in Ch n. The packet number means UART frame number per packet frame. "Frame 4" is packet frame that FID is 0b011. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							14:12	F3PKT	Frame 3 Packet Frame 3 packet number (Ch n) 0: Frame 3 data is ignored 3: Packet number is set 3 4: Packet number is set 4 5: Packet number is set 5 6: Packet number is set 6 1, 2, 7: Writing prohibited These bits define the packet number of packet frame 3 in Ch n. The packet number means UART frame number per packet frame. "Frame 3" is packet frame that FID is 0b010. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							11:9	F2PKT	Frame 2 Packet Frame 2 packet number (Ch n) 0: Frame 2 data is ignored 3: Packet number is set 3 4: Packet number is set 4 5: Packet number is set 5 6: Packet number is set 6 1, 2, 7: Writing prohibited These bits define the packet number of packet frame 2 in Ch n. The packet number means UART frame number per packet frame. "Frame 2" is packet frame that FID is 0b001. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							8:6	F1PKT	Frame 1 Packet Frame 1 packet number (Ch n) 0: Frame 1 data is ignored 3: Packet number is set 3 4: Packet number is set 4 5: Packet number is set 5 6: Packet number is set 6 1, 2, 7: Writing prohibited These bits define the packet number of packet frame 1 in Ch n. The packet number means UART frame number per packet frame. "Frame 1" is packet frame that FID is 0b000. These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							5	SYSEL	Synchronize select Asynchronous mode/synchronous mode select 0: Synchronize mode 1: Asynchronous mode This bit defines the selection of synchronous mode or asynchronous mode in Ch n. (n: 1 to 7) This setting affects WDT spec. and Tx synchronous pulse spec. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							4	TSCTS	Timestamp capture trigger select Timestamp capture trigger select (Ch n) (n: 1 to 7) 0: Transmission synchronous pulse timing select 1: Header receive timing select (CH n) This bit defines the selection of timestamp capture trigger. This setting is ignored in asynchronous mode (PSI5SPRCF1n.SYSEL=0b1). (n: 1 to 7) This bit can be written when PSI5S`PUOS.ACSTS is 0b0 (= configuration mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							3	TSCS	Timestamp counter select Timestamp counter select (Ch n) (n: 1 to 7) 0: Timestamp counter B select 1: Timestamp counter A select This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							2	TSEN	Timestamp enable Timestamp capture enable (Ch n) (n: 1 to 7) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							1	RFCPS	Rx Frame checksum CRC/Parity Select Rx frame checksum CRC/parity select (Ch n) (n: 1 to 7) 0: Parity select 1: CRC select This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							0	CHEN	Channel enable Channel enable (Ch n) 0: Disable 1: Enable This bit is invalid to "Frame lack error" and "Frame excess error". (Even when this bit is 0b0, "Frame lack error" and "Frame excess error" are not disable.) This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

PSI5SPRCF2n (PSI5S Receive Config2 chn Register)	Base + 0x184 + (n-1)*0x80	0x0	4	32	8/16/32	RW	29:25	F6PAYLD	Frame6 payload The number of packet frame6's payload (Ch n) 0 to 7 : Writing prohibited 8 : Sets 8 … x : Sets x … 28 : Sets 28 29 to 31 : Writing prohibited Initial value (0) to these bits are handled as 8. Writing less than 8 to these bits are handled as 8. Writing more than 28 to these bits are handled as 28. These bits are used to decide about the CRC/parity calculation scope. (n:1 to 7) These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							24:20	F5PAYLD	Frame5 payload The number of packet frame5's payload (Ch n) 0 to 7 : Writing prohibited 8 : Sets 8 … x : Sets x … 28 : Sets 28 29 to 31 : Writing prohibited Initial value (0) to these bits are handled as 8. Writing less than 8 to these bits are handled as 8. Writing more than 28 to these bits are handled as 28. These bits are used to decide about the CRC/parity calculation scope. (n:1 to 7) These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							19:15	F4PAYLD	Frame4 payload The number of packet frame4's payload (Ch n) 0 to 7 : Writing prohibited 8 : Sets 8 … x : Sets x … 28 : Sets 28 29 to 31 : Writing prohibited Initial value (0) to these bits are handled as 8. Writing less than 8 to these bits are handled as 8. Writing more than 28 to these bits are handled as 28. These bits are used to decide about the CRC/parity calculation scope. (n:1 to 7) These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							14:10	F3PAYLD	Frame3 payload The number of packet frame3's payload (Ch n) 0 to 7 : Writing prohibited 8 : Sets 8 … x : Sets x … 28 : Sets 28 29 to 31 : Writing prohibited Initial value (0) to these bits are handled as 8. Writing less than 8 to these bits are handled as 8. Writing more than 28 to these bits are handled as 28. These bits are used to decide about the CRC/parity calculation scope. (n:1 to 7) These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							9:5	F2PAYLD	Frame2 payload The number of packet frame2's payload (Ch n) 0 to 7 : Writing prohibited 8 : Sets 8 … x : Sets x … 28 : Sets 28 29 to 31 : Writing prohibited Initial value (0) to these bits are handled as 8. Writing less than 8 to these bits are handled as 8. Writing more than 28 to these bits are handled as 28. These bits are used to decide about the CRC/parity calculation scope. (n:1 to 7) These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							4:0	F1PAYLD	Frame1 payload The number of packet frame1's payload (Ch n) 0 to 7 : Writing prohibited 8 : Sets 8 … x : Sets x … 28 : Sets 28 29 to 31 : Writing prohibited Initial value (0) to these bits are handled as 8. Writing less than 8 to these bits are handled as 8. Writing more than 28 to these bits are handled as 28. These bits are used to decide about the CRC/parity calculation scope. (n:1 to 7) These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

PSI5SPWDEn (PSI5S WDT Enable chn Register)	Base + 0x188 + (n-1)*0x80	0x0	4	8/16/32	8/16/32	RW	0	WDTEB	Watchdog Timer enable Watchdog Timer of Rx frame Enable (Ch n) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) and when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
PSI5SPWDPn (PSI5S WDT Pre-scaler chn Register)	Base + 0x18C + (n-1)*0x80	0x0	4	16/32	8/16/32	RW	11:0	WDTPRS	Watch Dog Timer pre-scaler Watchdog timer pre-scaler (Ch n) 0: Stop WDT timer 1 to 4095: Enabled at 1 clock/x clock (x:1 to 4095) These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
PSI5SPWDEVn (PSI5S WDT Expiration Value chn Register)	Base + 0x190 + (n-1)*0x80	0x0	4	32	8/16/32	RW	23:0	WDTEX	Watchdog Timer expiration value Watchdog timer expiration value (Ch n) These bits define the expiration value of watchdog timer for Rx frame gap in Ch n. When watchdog counter counts down to 0 from this setting value, watchdog timer is judged as expired. (n: 1 to 7) These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

PSI5SPTCDn (PSI5S Tx Command Data chn Register)	Base + 0x194 + (n-1)*0x80	0x0	4	16/32	8/16/32	RW	15:11	ATRSCMD	Alternate transport command Alternate transport command (Ch n) These bits define the alternate transport command in Ch n. When ECU to sensor data is 0b1, select this command. (n: 1 to 7) These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							10:8	ACHID	Alternate channel ID Alternate transport ChID (Ch n) These bits define the alternate ChID in Ch n. When ECU to sensor data is 0b1, select this ChID. (n: 1 to 7) These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							7:3	TRSCMD	Transport command Transport command (Ch n) These bits define the transport command in Ch n. When ECU to sensor data is 0b0, select this command. (n: 1 to 7) These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							2:0	CHID	Channel ID Transport ChID (Ch n) These bits define the ChID in Ch n. When ECU to sensor data is 0b0, select this ChID. (n: 1 to 7) These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
PSI5SPCIEn (PSI5S CPU Interrupt Enable chn Register)	Base + 0x198 + (n-1)*0x80	0x0	4	16/32	8/16/32	RW	13	IEBDDSFN	Interrupt enable DDSR finish Interrupt enable of DDSR finish flag (Ch n) 0: Disable 1: Enable This bit defines the enable of CPU interruption which occurs by “DDSR transfer end in Ch n in PSI5S mode (PSI5SPCISn.ISTDDSFN)”. Writing 0b0 to this bit sets interruption by PSI5SPCISn.ISTDDSFN to disable. Writing 0b1 to this bit sets interruption by PSI5SPCISn.ISTDDSFN to enable. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							12	IEBDDSOW	Interrupt enable DDSR overwrite Interrupt enable of DDSR overwrite (Ch n) 0: Disable 1: Enable This bit defines the enable of CPU interruption which occurs by “DDSR transmission data overwrite error in Ch n in PSI5S mode (PSI5SPCISn.ISTDDSOW)”. Writing 0b0 to this bit sets interruption by PSI5SPCISn.ISTDDSOW to disable. Writing 0b1 to this bit sets interruption by PSI5SPCISn.ISTDDSOW to enable. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							11	IEBRFN	Interrupt enable Rx frame finish Interrupt enable of Rx frame finish flag (Ch n) 0: Disable 1: Enable This bit defines the enable of CPU interruption which occurs by “packet frame receiving end in Ch n in PSI5S mode (PSI5SPCISn.ISTRFN)”. Writing 0b0 to this bit sets interruption by PSI5SPCISn.ISTRFN to disable. Writing 0b1 to this bit sets interruption by PSI5SPCISn.ISTRFN to enable. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							10	IEBRFEX	Interrupt enable Rx frame excess error Interrupt enable of Rx frame excess error (Ch n) 0: Disable 1: Enable This bit defines the enable of CPU interruption which occurs by “Rx frame excess error in Ch n in PSI5S mode (PSI5SPCISn.ISTRFEX)”. Writing 0b0 to this bit sets interruption by PSI5SPCISn.ISTRFEX to disable. Writing 0b1 to this bit sets interruption by PSI5SPCISn.ISTRFEX to enable. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							9	IEBRFLK	Interrupt enable Rx frame lack error Interrupt enable of Rx frame lack error (Ch n) 0: Disable 1: Enable This bit defines the enable of CPU interruption which occurs by “Rx frame lack error in Ch n in PSI5S mode (PSI5SPCISn.ISTRFLK)”. Writing 0b0 to this bit sets interruption by PSI5SPCISn.ISTRFLK to disable. Writing 0b1 to this bit sets interruption by PSI5SPCISn.ISTRFLK to enable. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							8	IEBROV	Interrupt enable Rx overrun error CPU interrupt enable of Rx overrun error (Ch n) 0: Disable 1: Enable This bit defines the enable of CPU interruption which occurs by “Rx overrun error in Ch n in PSI5S mode (PSI5SPCISn.ISTROV)”. Writing 0b0 to this bit sets interruption by PSI5SPCISn.ISTROV to disable. Writing 0b1 to this bit sets interruption by PSI5SPCISn.ISTROV to enable. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							7	IEBRWDT	Interrupt enable Rx WDT error Interrupt enable of Rx frame WDT error (Ch n) 0: Disable 1: Enable This bit defines the enable of CPU interruption which occurs by “Rx frame WDT error in Ch n in PSI5S mode (PSI5SPCISn.ISTRWDT)”. Writing 0b0 to this bit sets interruption by PSI5SPCISn.ISTRWDT to disable. Writing 0b1 to this bit sets interruption by PSI5SPCISn.ISTRWDT to enable. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							3	IEBTRST	Interrupt enable transceiver status error CPU interrupt enable of Rx transceiver status error (Ch n) 0: Disable 1: Enable This bit defines the enable of CPU interruption which occurs by “Rx transceiver status error in Ch n in PSI5S mode (PSI5SPCISn.ISTTRST)”. Writing 0b0 to this bit sets interruption by PSI5SPCISn.ISTTRST to disable. Writing 0b1 to this bit sets interruption by PSI5SPCISn.ISTTRST to enable. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							2	IEBPT	Interrupt enable parity error CPU interrupt enable of parity error (Ch n) 0: Disable 1: Enable This bit defines the enable of CPU interruption which occurs by “Rx payload data parity error in Ch n in PSI5S mode (PSI5SPCISn.ISTPT)”. Writing 0b0 to this bit sets interruption by PSI5SPCISn.ISTPT to disable. Writing 0b1 to this bit sets interruption by PSI5SPCISn.ISTPT to enable. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							1	IEBCRC	Interrupt enable CRC error CPU interrupt enable of CRC error (Ch n) 0: Disable 1: Enable This bit defines the enable of CPU interruption which occurs by “Rx payload data CRC error in Ch n in PSI5S mode (PSI5SPCISn.ISTCRC)”. Writing 0b0 to this bit sets interruption by PSI5SPCISn.ISTCRC to disable. Writing 0b1 to this bit sets interruption by PSI5SPCISn.ISTCRC to enable. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
PSI5SPDREn (PSI5S DMA Transfer Request Enable chn Register)	Base + 0x19C + (n-1)*0x80	0x0	4	32	8/16/32	RW	2	DRQETFN	DMA request enable at ddsr Tx finish Enable of DMA request by the Tx finish (Ch n) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							1	DRQEWDT	DMA request enable at WDT Enable of DMA request by the WDT error (Ch n) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							0	DRQERFN	DMA request enable at Rx finish Enable of DMA request by the Rx finish (Ch n) 0: Disable 1: Enable This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
PSI5SPSTPn (PSI5S Sync Trigger Pre-scaler chn Register)	Base + 0x1A4 + (n-1)*0x80	0x0	4	16/32	8/16/32	RW	11:0	STPRS	Sync trigger pre-scaler Synchronous trigger generation counter's pre-scaler (Ch n) 0: Enabled at 1clock/1clock 1: Enabled at 1clock/2clock … x: Enabled at 1clock/(x+1)clock … 4095: Enabled at 1clock/4096 clock When PSI5SPSTSn.STSEL (=select signal) is 0b0, this generation trigger is selected and is used. When PSI5SPSTSn.STSEL (=select signal) is 0b1, this generation trigger is not used. (n: 1 to 7) These bits can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

PSI5SPSTEVn (PSI5S Sync Trigger Expiration Value chn Register)	Base + 0x1A8 + (n-1)*0x80	0x0	4	32	8/16/32	RW	23:0	STEX	Sync trigger expiration value Synchronous trigger generation counter expiration value (Ch n) These bits define the expiration value of synchronous trigger generation counter (Ch n). When synchronous trigger generation counter counts down to 0 from this setting value, synchronous trigger is judged as expired. When PSI5SPSTSn.STSEL (=select signal) is 0b0, this signal is selected and is used for synchronous trigger generation. When PSI5SPSTSn.STSEL (=select signal) is 0b1, this signal is not used for synchronous trigger generation. (n: 1 to 7) These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
PSI5SPSTSn (PSI5S Sync Trigger Select chn Register)	Base + 0x1AC + (n-1)*0x80	0x0	4	8/16/32	8/16/32	RW	0	STSEL	Sync trigger select Synchronous trigger select (Ch n) 0: Generate signal of the PSI5-S is selected 1: GTM output is selected, This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
Ch n Register/Rx (n: 1 to 7)										

PSI5SPRESn (PSI5S Receive Error Status chn Register)	Base + 0x1B0 + (n-1)*0x80	0x0	4	-	8/16/32	R	5	RERRF6	Rx error Frame6 Rx error at packet frame 6 (Ch n) 0: No error 1: An error has occurred This bit shows PSI5S Rx error status at packet frame6 of Ch n. This bit is set to 0b1 when any of “Rx overrun error”, “transceiver status error”, ”payload data parity error”, ”payload data CRC error” in packet frame6 of Ch n have occurred. This bit is read only. The write value is ignored. This bit is cleared when writing 0b1 to PSI5SPRESCn.RERRCLF6. (n: 1 to 7) This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes
							4	RERRF5	Rx error Frame5 Rx error at packet frame 5 (Ch n) 0: No error 1: An error has occurred This bit shows PSI5S Rx error status at packet frame5 of Ch n. This bit is set to 0b1 when any of “Rx overrun error”, “transceiver status error”, ”payload data parity error”, ”payload data CRC error” in packet frame5 of Ch n have occurred. This bit is read only. The write value is ignored. This bit is cleared when writing 0b1 to PSI5SPRESCn.RERRCLF5. (n: 1 to 7) This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes

							3	RERRF4	Rx error Frame4 Rx error at packet frame 4 (Ch n) 0: No error 1: An error has occurred This bit shows PSI5S Rx error status at packet frame4 of Ch n. This bit is set to 0b1 when any of “Rx overrun error”, “transceiver status error”, ”payload data parity error”, ”payload data CRC error” in packet frame4 of Ch n have occurred. This bit is read only. The write value is ignored. This bit is cleared when writing 0b1 to PSI5SPRESCn.RERRCLF4. (n: 1 to 7) This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes
							2	RERRF3	Rx error Frame3 Rx error at packet frame 3 (Ch n) 0: No error 1: An error has occurred This bit shows PSI5S Rx error status at packet frame3 of Ch n. This bit is set to 0b1 when any of “Rx overrun error”, “transceiver status error”, ”payload data parity error”, ”payload data CRC error” in packet frame3 of Ch n have occurred. This bit is read only. The write value is ignored. This bit is cleared when writing 0b1 to PSI5SPRESCn.RERRCLF3. (n: 1 to 7) This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes

							1	RERRF2	Rx error Frame2 Rx error at packet frame 2 (Ch n) 0: No error 1: An error has occurred This bit shows PSI5S Rx error status at packet frame2 of Ch n. This bit is set to 0b1 when any of “Rx overrun error”, “transceiver status error”, ”payload data parity error”, ”payload data CRC error” in packet frame2 of Ch n have occurred. This bit is read only. The write value is ignored. This bit is cleared when writing 0b1 to PSI5SPRESCn.RERRCLF2. (n: 1 to 7) This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes
							0	RERRF1	Rx error Frame1 Rx error at packet frame 1 (Ch n) 0: No error 1: An error has occurred This bit shows PSI5S Rx error status at packet frame1 of Ch n. This bit is set to 0b1 when any of “Rx overrun error”, “transceiver status error”, ”payload data parity error”, ”payload data CRC error” in packet frame1 of Ch n have occurred. This bit is read only. The write value is ignored. This bit is cleared when writing 0b1 to PSI5SPRESCn.RERRCLF1. (n: 1 to 7) This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes

PSI5SPRESCn (PSI5S Receive Error Status Clear chn Register)	Base + 0x1B4 + (n-1)*0x80	0x0	4	8/16/32	8/16/32	RW	5	RERRCLF6	Rx error clear Frame6 Rx error clear packet Frame6 (Ch n) 0: Is ignored 1: Clears PSI5SPRESn.RERRF6 This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes
							4	RERRCLF5	Rx error clear Frame5 Rx error clear packet Frame5 (Ch n) 0: Is ignored 1: Clears PSI5SPRESn.RERRF5 This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes
							3	RERRCLF4	Rx error clear Frame4 Rx error clear packet Frame4 (Ch n) 0: Is ignored 1: Clears PSI5SPRESn.RERRF4 This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes
							2	RERRCLF3	Rx error clear Frame3 Rx error clear packet Frame3 (Ch n) 0: Is ignored 1: Clears PSI5SPRESn.RERRF3 This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes

							1	RERRCLF2	Rx error clear Frame2 Rx error clear packet Frame2 (Ch n) 0: Is ignored 1: Clears PSI5SPRESn.RERRF2 This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes
							0	RERRCLF1	Rx error clear Frame1 Rx error clear packet Frame1 (Ch n) 0: Is ignored 1: Clears PSI5SPRESn.RERRF1 This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes
PSI5SPTCDTn (PSI5S Timestamp Capture Data chn Register)	Base + 0x1B8 + (n-1)*0x80	0x0	4	-	8/16/32	R	23:0	TSCD	Timestamp capture data Timestamp capture data (Ch n) These bits show the timestamp capture value in Ch n. These bits are cleared when writing 0b1 to PSI5SPTCDCn.TSCCLR. These bits are cleared when writing 0b0 to PSI5SPRCF1n.TSEN. These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
PSI5SPTCDCn (PSI5S Timestamp Capture Data Clear chn Register)	Base + 0x1BC + (n-1)*0x80	0x0	4	8/16/32	8/16/32	RW	0	TSCCLR	Timestamp capture clear Timestamp capture clear (Ch n) 0: Is ignored 1: Clears timestamp capture This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes
Ch n Register/Tx (n: 1 to 7)										

PSI5SPDDTPn (PSI5S DDSR Type chn Register)	Base + 0x1C0 + (n-1)*0x80	0x0	4	8/16/32	8/16/32	RW	1:0	DDSRTYPE	DDSR transmission type DDSR transmission type (Ch n) (n: 1 to 7) 0: Frame1 (Short) 1: Frame2 (Long) 2: Frame3 (XLong) 3: Frame4 (XXLong) Writing these bits are prohibited when PSI5SPDDSn.DDSRSTS is 0b1. This bit can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
PSI5SPDDDn (PSI5S DDSR Data chn Register)	Base + 0x1C4 + (n-1)*0x80	0x00FFFFFF	4	32	8/16/32	RW	23:4	DDSRDT	DDSR transmission data DDSR transmission data (Ch n) These bits cannot be written when PSI5SPDDSn.DDSRSTS is 0b1. When PSI5SPDDTPn.DDSRTYPE is 0, DDSR transmission data use LSB 3bit. PSI5SPDDDn.DDSRDT [19:4] should be set all 1. When PSI5SPDDTPn.DDSRTYPE is 1, DDSR transmission data use LSB 13bit. PSI5SPDDDn.DDSRDT [19:13] should be set all 1. When PSI5SPDDTPn.DDSRTYPE is 2, DDSR transmission data use LSB 19bit. PSI5SPDDDn.DDSRDT [20] should be set all 1. When PSI5SPDDTPn.DDSRTYPE is 3, DDSR transmission data use 20bit. (n: 1 to 7) These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							3:0	DDSRADR	DDSR transmission address DDSR transmission address (Ch n) These bits define the DDSR transmission address in Ch n. These bits cannot be written when PSI5SPDDSn.DDSRSTS is 0b1. (n: 1 to 7) These bits can be written when PSI5SPUOS.ACSTS is 0b0 (= configuration mode) or when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode). These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
PSI5SPDDSn (PSI5S DDSR Status chn Register)	Base + 0x1C8 + (n-1)*0x80	0x0	4	-	8/16/32	R	0	DDSRSTS	DDSR status 0: DDSR transmission is not busy 1: DDSR transmission is busy This bit shows DDSR transmission status in Ch n. In PSI5S mode, when PSI5SPDDDn is written, this bit set to 0b1. When last data of PSI5SPDDDn is written to Tx shifter, this bit will be 0b0. PSI5SPDDDn is cannot written when PSI5SPDDSn.DDSRSTS is 0b1. This bit is cleared when writing 0b1 to PSI5SPDDSPn.DDSRSTP. This bit is read only. The write value is ignored. (n: 1 to 7) This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes

PSI5SPDDSPn (PSI5S DDSR Stop chn Register)	Base + 0x1CC + (n-1)*0x80	0x0	4	8/16/32	8/16/32	RW	0	DDSRSTP	DDSR Tx stop DDSR Tx stop (Ch n) 0: Is ignored 1: Stop transmission This bit defines the DDSR transmission stop in Ch n. Writing 0b0 to this bit is ignored. Writing 0b1 to this bit, transmission command is force to 1 by reason that DDSR-SHIFT register is set all 1 in Ch n. When this bit is written PSI5SPDDSn.DDSRSTS is reset to 0b0. This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes
Ch n Register/Interrupt (n: 1 to 7)										
PSI5SPCISn (PSI5S CPU Interrupt Status chn Register)	Base + 0x1D0 + (n-1)*0x80	0x0	4	-	8/16/32	R	13	ISTDDSFN	Interrupt status DDSR finish CPU interrupt status of DDSR finish (Ch n) 0: DDSR transmission is not finish 1: DDSR transmission finish This bit shows the status of CPU interruption (Ch n) which occurs by DDSR Tx end in PSI5S mode. When PSI5SPDDDn data is written to Tx shifter, this bit is set to 0b1 and an interruption (int_psis_chn) and DMA request (dma_psis_chn_tx) occur. This bit is read only. The write value is ignored. (n: 1 to 7) This bit is cleared when writing 0b1 to PSI5SPCISCn.ISTCDDSFN. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes

							12	ISTDDSOW	Interrupt status DDSR overwrite error CPU interrupt status of DDSR overwrite error (Ch n) 0: No error 1: Error detected This bit shows the status of CPU interruption (Ch n) which occurs by DDSR overwrite error in PSI5S mode. When CPU writes PSI5SPDDDn in PSI5SPDDSn.DDSRSTS is 0b1, this bit is set to 0b1 and an interruption (int_psis_chn) occurs. This bit is read only. The write value is ignored. (n: 1 to 7) This bit is cleared when writing 0b1 to PSI5SPCISCn.ISTCDDSOW. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes
							11	ISTRFN	Interrupt status Rx finish CPU interrupt status of Rx finish (Ch n) 0: PSI5S frame is not received successfully 1: PSI5S frame is received successfully This bit shows the status of CPU interruption (Ch n) which occurs by end of Rx frame in PSI5S mode. When the PSI5S frame is stored to MB PSI5S frame has no errors (without mailbox overrun error), this bit is set to 0b1 and an interruption (int_psis_chn) and DMA request (dma_psis_chn_rx) occur. This bit is read only. The write value is ignored. (n: 1 to 7) This bit is cleared when writing 0b1 to PSI5SPCISCn.ISTCRFN. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes

							10	ISTRFEX	Interrupt status Rx frame excess error CPU interrupt status of Rx frame excess error (Ch0) 0: No error 1: Error detected This bit shows the status of CPU interruption (Ch n) which occurs by Rx frame (packet) excess error in PSI5S mode. When packet is received over PSI5SPRCF1n.FmPKT (m = 1 to 6), this bit is set to 0b1 and an interruption (int_psis_chn) occurs. This bit is read only. The write value is ignored. (n: 1 to 7) This bit is cleared when writing 0b1 to PSI5SPCISCn.ISTCRFEX. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes
							9	ISTRFLK	Interrupt status Rx frame lack error CPU interrupt status of Rx frame lack error (Ch0) 0: No error 1: Error detected This bit shows the status of CPU interruption (Ch n) which occurs by Rx frame (packet) lack error in PSI5S mode. When a packet frame gap is detected in the state for which packet is lack of PSI5SPRCF1n.FmPKT (m = 1 to 6), this bit is set to 0b1 and an interruption (int_psis_chn) occurs. This bit is read only. The write value is ignored. (n: 1 to 7) This bit is cleared when writing 0b1 to PSI5SPCISCn.ISTCRFLK. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes

							8	ISTROV	Interrupt status Rx overrun error CPU interrupt status of Rx overrun error (Ch n) 0: No error 1: Error detected This bit shows the status of CPU interruption (Ch n) which occurs by overrun error in PSI5S mode. When next PSI5S frame is stored to same MB before CPU read MB or set PSI5SPRMBC.MBCLR, this bit is set to 0b1 and an interruption (int_psis_chn) occurs. This bit is read only. The write value is ignored. (n: 1 to 7) This bit is cleared when writing 0b1 to PSI5SPCISCn.ISTCROV. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes
							7	ISTRWDT	Interrupt status Rx WDT error CPU interrupt status of Rx WDT error (Ch n) 0: No error 1: Error detected This bit shows the status of CPU interruption (Ch n) which occurs by WDT error in PSI5S mode. When WDT error occurs in PSI5S mode, this bit is set to 0b1 and an interruption (int_psis_chn) and DMA request (dma_psis_chn_rx) occur. This bit is read only. The write value is ignored. (n: 1 to 7) This bit is cleared when writing 0b1 to PSI5SPCISCn.ISTCRWDT. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes

							3	ISTTRST	Interrupt status transceiver status error CPU interrupt status of Rx transceiver status error (Ch n) 0: No error 1: Error detected This bit shows the status of CPU interruption (Ch n) which occurs by transceiver status error in PSI5S mode. When the PSI5S frame is stored to a MB transceiver status error occurred, this bit is set to 0b1 and an interruption (int_psis_chn) occurs. This bit is read only. The write value is ignored. (n: 1 to 7) This bit is cleared when writing 0b1 to PSI5SPCISCn.ISTCTRST. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes
							2	ISTPT	Interrupt status parity error CPU interrupt status of payload data parity error (Ch n) 0: No error 1: Error detected This bit shows the status of CPU interruption (Ch n) which occurs by payload data parity error in PSI5S mode. When the PSI5S frame is stored MB payload data parity error occurred, this bit is set to 0b1and an interruption (int_psis_chn) occurs. This bit is read only. The write value is ignored. (n: 1 to 7) This bit is cleared when writing 0b1 to PSI5SPCISCn.ISTCTPT. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes

							1	ISTCRC	Interrupt status CRC error CPU interrupt status of payload data CRC error (Ch n) 0: No error 1: Error detected This bit shows the status of CPU interruption (Ch n) which occurs by payload data CRC error in PSI5S mode. When the PSI5S frame is stored MB payload data CRC error occurred, this bit is set to 0b1and an interruption (int_psis_chn) occurs. This bit is read only. The write value is ignored. (n: 1 to 7) This bit is cleared when writing 0b1 to PSI5SPCISCn.ISTCCRC. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. This bit is cleared when PSI5SPUOS.ACSTS is changed to 0b0 (= configuration mode).	Yes
PSI5SPCISCn (PSI5S CPU Interrupt Status Clear chn Register)	Base + 0x1D4 + (n-1)*0x80	0x0	4	16/32	8/16/32	RW	13	ISTCDDSFN	Interrupt status clear DDSR Tx finish Clear at CPU interrupt status of DDSR finish(Chn) 0: Is ignored 1: Clears PSI5SPCISn.ISTDDSFN This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes
							12	ISTCDDSOW	Interrupt status clear DDSr overwrite error Clear at CPU interrupt status of DDSR overwrite error (Chn) 0: Is ignored 1: Clears PSI5SPCISn.ISTDDSOW This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes

							11	ISTCRFN	Interrupt status clear Rx finish Clear at CPU interrupt status of Rx finish (Chn) 0: Is ignored 1: Clears PSI5SPCISn.ISTRFN This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes
							10	ISTCRFEX	Interrupt status clear Rx frame excess error Clear at CPU interrupt status of Rx frame excess error (Chn) 0: Is ignored 1: Clears PSI5SPCISn.ISTRFEX This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes
							9	ISTCRFLK	Interrupt status clear Rx frame lack error Clear at CPU interrupt status of Rx frame lack error (Chn) 0: Is ignored 1: Clears PSI5SPCISn.ISTRFLK This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes
							8	ISTCROV	Interrupt status clear Rx overrun error Clears at CPU interrupt status of Rx overrun error (Chn) 0: Is ignored 1: Clears PSI5SPCISn.ISTROV This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes

							7	ISTCRWDT	Interrupt status clear Rx WDT error Clear at CPU interrupt status of Rx WDT error (Chn) 0: Is ignored 1: Clears PSI5SPCISn.ISTRWDT This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes
							3	ISTCTRST	Interrupt status clear transceiver status error Clear at CPU interrupt status of Rx transceiver status error (Chn) 0: Is ignored 1: Clears PSI5SPCISn.ISTTRST This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes
							2	ISTCPT	Interrupt status clear parity error Clear at CPU interrupt status of payload data parity error (Chn) 0: Is ignored 1: Clears PSI5SPCISn.ISTPT This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes
							1	ISTCCRC	Interrupt status clear CRC error Clear at CPU interrupt status of payload data CRC error (Chn) 0: Is ignored 1: Clears PSI5SPCISn.ISTCRC This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SPUOS.ACSTS is 0b1 and PSI5SPUOS.MSTS is 0b1 (= PSI5S mode).	Yes
Ch n Register/Test ==> Not support test mode										

Ch 0 Frm m MB Data (m: 1, 2)										
PSI5SPMB0mS (PSI5S Receive MailBox ch0 Frmm Status Register)	Base + 0x500 + (m-1)*0xC	0x0	4	-	8/16/32	R	31:28	DCI	DCI value DCI value (Ch0, Frm m) These bits show the DCI value. (Ch 0, Frm m) DCI value is generated at 4 bits counter, and every time PSI5 frame data is restored, it is count +1. These bits are read only. The write value is ignored. (m: 1, 2) These bits are cleared when writing 0b1 to PSI5SPRMBC.MBCLR. These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							24:22	CHID	Rx channel ID Rx channel ID (Ch0, Frm m) These bits show the Rx Channel ID value. (Ch 0, Frm m) These bits are read only. The write value is ignored. (m: 1, 2) When WDT error occurs, replacing the PSI5SPMB02S.CHID to the channel ID of the occurrence channel(0-7). These bits are cleared when writing 0b1 to PSI5SPRMBC.MBCLR. These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							21:19	FID	Rx frame ID Rx frame ID (Ch0, Frm m) These bits show the Frame ID value (=m-1). (Ch 0, Frm m) These bits are read only. The write value is ignored. (m: 1, 2) When WDT error occurs in synchronous mode (PSI5SPRCF1n.SYSEL=0b0), replacing the PSI5SPMB02S.FID to the packet frame counter value(0-7) of the occurrence channel. These bits are cleared when writing 0b1 to PSI5SPRMBC.MBCLR. These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							18	MBORERR	Mailbox over run error Mailbox overrun error (Ch0, Frm m) 0: No error 1: Error detected This bit shows the Mailbox over run error. (Ch 0, Frm m) This bit is read only. The write value is ignored. (m: 1, 2) This bit is cleared when writing 0b1 to PSI5SPRMBC.MBCLR. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							17	WDTERR	WDT error Rx frame WDT error (Ch0, Frm m) 0: No error 1: Error detected This bit shows the WDT error. (Frm m) This bit is read only. The write value is ignored. (m: 1, 2) This bit is cleared when writing 0b1 to PSI5SPRMBC.MBCLR. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							15	UTFRERR	UART framing error UART framing error (Ch0, Frm2) *1 0: No error 1: Error detected This bit shows the UART framing error. (Ch 0, Frm2) This bit is read only. The write value is ignored. This bit is cleared when writing 0b1 to PSI5SPRMBC.MBCLR. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. When this error occurred, data store in channel0, frame2. So, this error exists only in channel0, frame2. *1 This bit is only in 50Ch(Frm2)	Yes

							14	UTPTERR	UART parity error UART parity error (Ch0, Frm2) *1 0: No error 1: Error detected This bit shows the UART parity error. (Ch 0, Frm2) This bit is read only. The write value is ignored. This bit is cleared when writing 0b1 to PSI5SPRMBC.MBCLR. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. When this error occurred, data store in channel0, frame2. So, this error exists only in channel0, frame2. *1 This bit is only in 50Ch (Frm2)	Yes
							13	HEADERR	Header error Header error (Ch0, Frm m) 0: No error 1: Error detected This bit shows the Header error. (Ch 0, Frm m) This bit is read only. The write value is ignored. (m: 1, 2) This bit is cleared when writing 0b1 to PSI5SPRMBC.MBCLR. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							12:11	HEADST	Header status Header status These bits show the Header status. (Ch 0, Frm m) These bits are read only. The write value is ignored. (m: 1, 2) These bits are cleared when writing 0b1 to PSI5SPRMBC.MBCLR. These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							10	CRCERR	CRC error Rx CRC/Parity error (Ch0, Frm m) 0: No error 1: Error detected This bit shows the Rx frame CRC/Parity error. (Ch 0, Frm m) When PSI5SPRCF10.RFCPS is 0b1, this bit shows CRC error. And when PSI5SPRCF10.RFCPS is 0b0, this bit shows parity error. This bit is read only. The write value is ignored. (m: 1, 2) This bit is cleared when writing 0b1 to PSI5SPRMBC.MBCLR. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							9:7	CRC	CRC Rx frame CRC/Parity (Ch0, Frm m) These bits show the Rx frame CRC/Parity. (Ch 0, Frm m) When PSI5SPRCF10.RFCPS is 0b1, these bits show CRC (3bits). And when PSI5SPRCF10.RFCPS is 0b0, bit [9] shows parity, and bit [8:7] is reserved. These bits are read only. The write value is ignored. (m: 1, 2) These bits are cleared when writing 0b1 to PSI5SPRMBC.MBCLR. These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							6	XCRCERR	XCRC error Rx XCRC error (Ch0, Frm2) *1 0: No error 1: Error detected This bit shows the Rx frame XCRC error. (Ch 0, Frm2) This bit is read only. The write value is ignored. This bit is cleared when writing 0b1 to PSI5SPRMBC.MBCLR. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST. When this error occurred, data store in channel0, frame2. So, this error exists only in channel0, frame2. *1 This bit is only in 50Ch (Frm2)	Yes
							5:0	XCRC	XCRC Rx frame XCRC (Ch0, Frm m) These bits show the Rx frame XCRC. (Ch 0, Frm m) These bits are read only. The write value is ignored. (m: 1, 2) These bits are cleared when writing 0b1 to PSI5SPRMBC.MBCLR. These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
PSI5SPMB0mD (PSI5S Receive MailBox ch0 Frmm Data Register)	Base + 0x504 + (m-1)*0xC	0x0	4	-	8/16/32	R	31:28	DCI	DCI value DCI value (Ch0, Frm m) These bits show the DCI value. (Ch 0, Frm m) These bits are read only. The write value is ignored. (m: 1, 2) These bits are cleared when writing 0b1 to PSI5SPRMBC.MBCLR. These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							27:0	DATA	Rx message data Message data (Ch0, Frm m) These bits show the Rx message (=payload) data. (Ch 0, Frm m) When the number of payload (PSI5SPRCF20.FmPAYLD) is less than 28, module stores payload at LSB and MSB is fill 0. These bits are read only. The write value is ignored. (m: 1, 2) These bits are cleared when writing 0b1 to PSI5SPRMBC.MBCLR. These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
PSI5SPMB0mT (PSI5S Receive MailBox ch0 Frmm Timestamp Register)	Base + 0x508 + (m-1)*0xC	0x0	4	-	8/16/32	R	31:28	DCI	DCI value DCI value (CH0, Frm m) These bits show the DCI value. (Ch 0, Frm m) These bits are read only. The write value is ignored. (m: 1, 2) These bits are cleared when writing 0b1 to PSI5SPRMBC.MBCLR. These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							23:0	TMST	Timestamp data Timestamp data (CH0, Frm m) These bits show the Rx Timestamp data. (Ch 0, Frm m) These bits are read only. The write value is ignored. (m: 1, 2) These bits are cleared when writing 0b1 to PSI5SPRMBC.MBCLR. These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
Ch n Frm m MB Data (n: 1 to 7) (m: 1 to 6)										

PSI5SPMBnmS (PSI5S Receive MailBox ch n Frm m Status Register)	Base + 0x548 + (m-1)*0xC + (n-1)*0x48	0x0	4	-	8/16/32	R	31:28	DCI	DCI value DCI value (Chn, Frm m) These bits show the DCI value. (Ch n, Frm m) DCI value is generated at 4 bits counter, and every time PSI5 frame data is restored, it is count +1. These bits are read only. The write value is ignored. (n: 1 to 7), (m: 1 to 6) These bits are cleared when writing 0b1 to PSI5SPRMBC.MBCLR. These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							24:22	CHID	Rx channel ID Rx channel ID (Chn, Frm m) These bits show the Rx Channel ID value. (Ch n, Frm m) These bits are read only. The write value is ignored. (n: 1 to 7), (m: 1 to 6) These bits are cleared when writing 0b1 to PSI5SPRMBC.MBCLR. These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							21:19	FID	Rx frame ID Rx frame ID (Chn, Frm m) These bits show the Frame ID value (=m). (Ch n, Frm m) These bits are read only. The write value is ignored. (n: 1 to 7), (m: 1 to 6) These bits are cleared when writing 0b1 to PSI5SPRMBC.MBCLR. These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							18	MBORERR	Mailbox overrun error Mailbox overrun error (Chn, Frm m) 0: No error 1: Error detected This bit shows the Mailbox overrun error. (Ch n, Frm m) This bit is read only. The write value is ignored. (n: 1 to 7), (m: 1 to 6) This bit is cleared when writing 0b1 to PSI5SPRMBC.MBCLR. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							13	HEADERR	Header error Header error (Chn, Frm m) 0: No error 1: Error detected This bit shows the Header error. (Ch n, Frm m) This bit is read only. The write value is ignored. (n: 1 to 7), (m: 1 to 6) This bit is cleared when writing 0b1 to PSI5SPRMBC.MBCLR. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							12:11	HEADST	Header status Header status (Chn, Frm m) These bits show the Header status. (Ch n, Frm m) These bits are read only. The write value is ignored. (n: 1 to 7), (m: 1 to 6) These bits are cleared when writing 0b1 to PSI5SPRMBC.MBCLR. These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							10	CRCERR	CRC error Rx CRC/Parity error (Chn, Frm m) 0: No error 1: Error detected This bit shows the Rx frame CRC/Parity error. (Ch n, Frm m) When PSI5SPRCF1m.RFCPS is 0b1, this bit shows CRC error. And when PSI5SPRCF1m.RFCPS is 0b0, this bit shows parity error. This bit is read only. The write value is ignored. (n: 1 to 7), (m: 1 to 6) This bit is cleared when writing 0b1 to PSI5SPRMBC.MBCLR. This bit is cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							9:7	CRC	CRC Rx frame CRC/Parity (Chn, Frm m) *1 These bits show the Rx frame CRC/Parity. (Ch n, Frm m) When PSI5SPRCF1m.RFCPS is 0b1, these bits show CRC (3bits). And when PSI5SPRCF1m.RFCPS is 0b0, bit [9] shows parity, and bit [8:7] is reserved. These bits are read only. The write value is ignored. (n: 1 to 7), (m: 1 to 6) These bits are cleared when writing 0b1 to PSI5SPRMBC.MBCLR. These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							5:0	XCRC	XCRC Rx frame XCRC (Chn, Frm m) These bits show the Rx frame XCRC. (Ch n, Frm m) These bits are read only. The write value is ignored. (n: 1 to 7), (m: 1 to 6) These bits are cleared when writing 0b1 to PSI5SPRMBC.MBCLR. These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

PSI5SPMBnmD (PSI5S Receive MailBox ch0 Frmm Data Register)	Base + 0x54C + (m-1)*0xC + (n-1)*0x48	0x0	4	-	8/16/32	R	31:28	DCI	DCI value DCI value (Ch0, Frm m) These bits show the DCI value. (Ch 0, Frm m) These bits are read only. The write value is ignored. (m: 1, 2) These bits are cleared when writing 0b1 to PSI5SPRMBC.MBCLR. These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
							27:0	DATA	Rx message data Message data (Ch0, Frm m) These bits show the Rx message (=payload) data. (Ch 0, Frm m) When the number of payload (PSI5SPRCF20.FmPAYLD) is less than 28, module stores payload at LSB and MSB is fill 0. These bits are read only. The write value is ignored. (m: 1, 2) These bits are cleared when writing 0b1 to PSI5SPRMBC.MBCLR. These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes
PSI5SPMBnmT (PSI5S Receive MailBox ch0 Frmm Timestamp Register)	Base + 0x550 + (m-1)*0xC + (n-	0x0	4	-	8/16/32	R	31:28	DCI	DCI value DCI value (CH0, Frm m) These bits show the DCI value. (Ch 0, Frm m) These bits are read only. The write value is ignored. (m: 1, 2) These bits are cleared when writing 0b1 to PSI5SPRMBC.MBCLR. These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

							23:0	TMST	Timestamp data Timestamp data (CH0, Frm m) These bits show the Rx Timestamp data. (Ch 0, Frm m) These bits are read only. The write value is ignored. (m: 1, 2) These bits are cleared when writing 0b1 to PSI5SPRMBC.MBCLR. These bits are cleared when writing 0b1 to PSI5SPUSWR.SWRST.	Yes

5.List of implemented ports

(1) Table 5.1 lists implemented ports in PSIS011 model.

Table 5.1: List of implemented ports in PSIS011 model

Signal name		I/O	Type	Initial	Active	Sync. clock	Description	Support
HWM	**Model**							
Clock/reset								
PCLK	<-	In	sc_in<sc_dt::uint64>	-	-	-	Peripheral (APB) clock	Yes
psis_clk	<-	In	sc_in<sc_dt::uint64>	-	-	-	Communication clock	Yes
psis_mult_clk	<-	In	sc_in<sc_dt::uint64>	-	-	-	Communication multiply clock	Yes
PRESETn	<-	In	sc_in<bool>	-	LOW	-	APB reset	Yes
psis_rst_n	<-	In	sc_in<bool>	-	LOW	-	Reset of clock domain psis_clk	Yes
psis_mult_rst_n	<-	In	sc_in<bool>	-	LOW	-	Reset of clock domain psis_mult_clk	Yes
Scan								
scan_mode	-	In	-	-	-	-	Scan mode	No
scan_enable	-	In	-	-	-	-	Scan enable	No
APB I/F								
PSEL	ts	In	TlmTargetSocket	-	-	-	Target socket of APB bus interface to access to registers of model	Yes
PWRITE		In						
PENABLE		In						
PSTRB		In						
PADDR		In						
PWDATA		In						
PRDATA		Out						
PREADY		Out						
UART								
psis_rx_data	RX_DATA	In	sc_in<unsigned int>	-	-	psis_clk	UART Rx data	Yes
-	RX_CONTROL	In	sc_in<unsigned int>	-	-	psis_clk	UART Rx control	Yes

psis_tx_data	TX_DATA	Out	sc_out<unsigned int>	0	-	psis_mult_clk	UART Tx data	Yes
psis_tx_sclk	TX_CONTROL	Out	sc_out<unsigned int>	0	-	psis_mult_clk	UART Tx control	Yes
Interrupt								
int_psis_ch0	int_psis_chn	Out	sc_out<bool> [8]	LOW	HIGH	PCLK	Interrupt of channel n (n: 0 to 7)	Yes
int_psis_ch1		Out						
int_psis_ch2		Out						
int_psis_ch3		Out						
int_psis_ch4		Out						
int_psis_ch5		Out						
int_psis_ch6		Out						
int_psis_ch7		Out						
DMA								
dma_psis_ch0_rx	dma_psis_chn_rx	Out	sc_out<bool> [8]	LOW	HIGH	PCLK	DMA transfer request RX	Yes
dma_psis_ch1_rx		Out						
dma_psis_ch2_rx		Out						
dma_psis_ch3_rx		Out						
dma_psis_ch4_rx		Out						
dma_psis_ch5_rx		Out						
dma_psis_ch6_rx		Out						
dma_psis_ch7_rx		Out						
dma_psis_ch1_tx	dma_psis_chn_tx	Out	sc_out<bool> [7]	LOW	HIGH	PCLK	DMA transfer request TX	Yes
dma_psis_ch2_tx		Out						
dma_psis_ch3_tx		Out						
dma_psis_ch4_tx		Out						
dma_psis_ch5_tx		Out						
dma_psis_ch6_tx		Out						
dma_psis_ch7_tx		Out						
GTM interface								

psis_trg_sync_ch0	psis_trg_sync_chn	In	sc_in<bool> [8]	-	HIGH	psis_mult_clk	Sync pulse of channel n (n: 0 to 7)	Yes
psis_trg_sync_ch1		In						
psis_trg_sync_ch2		In						
psis_trg_sync_ch3		In						
psis_trg_sync_ch4		In						
psis_trg_sync_ch5		In						
psis_trg_sync_ch6		In						
psis_trg_sync_ch7		In						
psis_clk_timestamp_a	<-	In	sc_in<bool>	-	HIGH	psis_mult_clk	Timestamp clock A	Yes
psis_clk_timestamp_b	<-	In	sc_in<bool>	-	HIGH	psis_mult_clk	Timestamp clock B	Yes
psis_clr_timestamp_a	<-	In	sc_in<bool>	-	HIGH	psis_mult_clk	Timestamp clear A	Yes
psis_clr_timestamp_b	<-	In	sc_in<bool>	-	HIGH	psis_mult_clk	Timestamp clear B	Yes
psis_stsp_timestamp_a	<-	In	sc_in<bool>	-	HIGH	psis_mult_clk	Timestamp start stop A	Yes
psis_stsp_timestamp_b	<-	In	sc_in<bool>	-	HIGH	psis_mult_clk	Timestamp start stop A	Yes

6.Direction for users

6.1.File structures

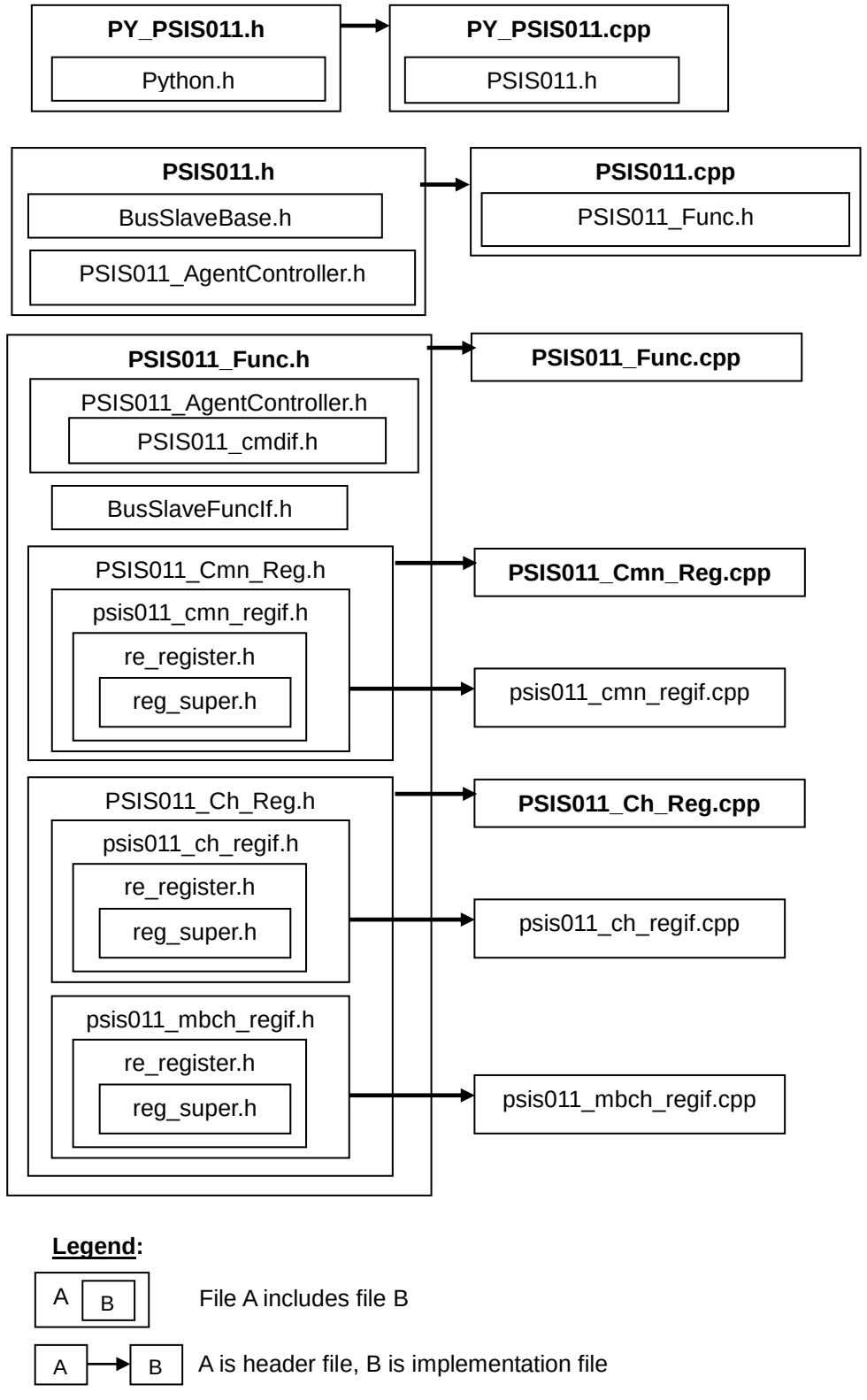


Figure 6.1: File structure of PSIS011 model

Table 6.1: File description for PSIS011

No.	File name	Version	Developed/ Reused/ Generated	Description
1	*PY_PSIS011.h*	-	Developed	Header file of Python Interface of PSIS011 model
2	*PY_PSIS011.cpp*		Developed	Implementation file of Python Interface of PSIS011 model.
3	*PSIS011.h*		Developed	Header file of PSIS011 model.
4	*PSIS011.cpp*		Developed	Implementation file of PSIS011 model.
5	*PSIS011_Func.h*		Developed	Header file of PSIS011 function block.
6	*PSIS011_Func.cpp*		Developed	Implementation file of PSIS011 function block.
7	*PSIS011_AgentController.h*		Developed	Header file includes virtual functions which are implemented in PSIS011 model.
8	*PSIS011_cmdif.h*		Generated*	Command interface of PSIS011 model.
9	*PSIS011_Cmn_Reg.h*		Developed	Header file of Common Registers block
10	*PSIS011_Cmn_Reg.cpp*		Developed	Implementation file of Common Registers block
11	*psis011_cmn_regif.h*		Generated*	Header file of Common registers' interface.
12	*psis011_cmn_regif.cpp*		Generated*	Implementation file of Common registers' interface.
13	*PSIS011_Ch_Reg.h*		Developed	Header file of Channel Registers block
14	*PSIS011_Ch_Reg.cpp*		Developed	Implementation file of Channel Registers block
15	*psis011_ch_regif.h*		Generated*	Header file of Channel registers' interface.
16	*psis011_ch_regif.cpp*		Generated*	Implementation file of Channel registers' interface.
17	*psis011_mbch_regif.h*		Generated*	Header file of Mail Box Frame registers' interface of each channel.
18	*psis011_mbch_regif.cpp*		Generated*	Implementation file of Mail Box Frame registers' interface of each channel.
19	Python.h	-	Reused	Header file of python library.
20	re_register.h	v2013_12_17 (**)	Reused	Header file of the re_register class.
21	re_register.cpp		Reused	Implement the attributes and the operations of common register class.
22	reg_super.h		Reused	General class for models to access to the memory array.
23	BusSlaveBase.h	-	Reused	Header file of BusSlaveBase class.
24	BusSlaveFuncIf.h		Reused	Header file of BusSlaveFuncIf class.
25	re_define.h	-	Reused	Define common define macro, enum and so on
26	OSCI2.h	-	Reused	Header file of TLM implementation.

(1) **(*)Note:** File *PSIS011_cmdif.h* is generated from Command I/F Generator v2015_02_12.

(2) File with format **_regif.h/cpp* in table above are generated from Register IF Generator tool v2015_04_06.

(3) (**) Note: re_register class is added UpdateBitInitValue() function. This function is used to update the initial value of each bit of register.

6.2.Input/Output file

(1) There is no input/output file.

6.3.How to connect Verification Environment

(1) The following steps should be done to connect PSIS011 into environment.

(1.1) Declare an instance of the PSIS011 class in environment.

(1.2) Connect the target socket of PSIS011 instance to according initiator socket.

(1.3) Connect reset/clock ports of PSIS011 to according ports.

(1.4) Connect interrupt output ports to CPU; connect DMA request output port to DMAC.

6.4.Commands and parameters

(1) Users set parameters/commands to the PSIS011 via Python IF to control operation. Table 6.2 and Table *6.3* list supported parameters/commands in PSIS011 model.

Table 6.2: List of supported parameters

Parameter	Type	Default	Description
PSIS011_MessageLevel	string	fatal|error	Select debug message level (“fatal”, “error”, “warning”, “info”). One or more than levels can be connected by vertical bar. Example “fatal|error”
PSIS011_DumpRegisterRW	bool	false	Dump register access information when registers are accessed. + false: Not dump register access information + true: Dump register access information
PSIS011_DumpInterrupt	bool	false	Dump interrupt information when interrupt is assert. + false: Not dump interrupt information + true: Dump interrupt information

Table 6.3: List of supported commands

Command	Type	Argument	Description
PSIS011_SetDNFDelay	void	delay	Set delay time for Noise Filter in PSIS011 model. Default delay time is 0. Unit is number of psis_clk clock. Example: PSIS011_SetDNFDelay(2). The delay time at Noise Filter is 2 psis_clk cycles. Note: When Noise Filter is set to disable (PSI5SPUNFST.NFSET bit = 0), this command is ignored.
PSIS011_DumpStatusInfo	void	-	Dump the status information of the PSIS011.
PSIS011_AssertReset	void	reset_name, start-time, period	Assert and deassert reset signal + std::string <reset_name>: name of reset signal ("PRESETn", "psis_rst_n", "psis_mult_rst_n") + double <start-time>: the time until asserting reset signal from current time. The unit is "ns" + double <period>: the time from asserting reset signal to de-assert it. The unit is "ns"

PSIS011_SetCLKFreq	void	clock_name, freq, unit	Set frequency value to these blocks + std::string <clock_name>: name of clock signal ("PCLK", "psis_clk", "psis_mult_clk") + sc_dt::uint64 <freq>: clock frequency + std::string <unit>: frequency unit ("Hz", "KHz", "MHz" or "GHz")
PSIS011_GetCLKFreq	void	clock_name	Get frequency value of these blocks + std::string <clock_name>: name of clock signal ("PCLK", "psis_clk", "psis_mult_clk")
PSIS011_ForceRegister	void	reg_name, chid, value	Force register with setting value + std::string <reg_name>: name of register. + unsigned int <chid>: channel index. Note, chid is used for register of channel from 0-7. For common registers, chid is ignored. + unsigned int <value>: value which is set to register
PSIS011_ReleaseRegister	void	reg_name, chid	Release register from force value + std::string <reg_name>: name of register. + unsigned int <chid>: channel index. Note, chid is used for register of channel from 0-7. For common registers, chid is ignored.
PSIS011_WriteRegister	void	reg_name, chid, value	Write a value to register + std::string <reg_name>: name of register. + unsigned int <chid>: channel index. Note, chid is used for register of channel from 0-7. For common registers, chid is ignored. + unsigned int <value>: value which is set to register
PSIS011_ReadRegister	void	reg_name, chid	Read a value from register + std::string <reg_name>: name of register. + unsigned int <chid>: channel index. Note, chid is used for register of channel from 0-7. For common registers, chid is ignored.
PSIS011_ListRegister	void	-	Dump register names of model
PSIS011_Help	void	type	Dump the direction how to use python interface parameters and commands + std::string <type>: "parameters" or "commands"

6.5.Message style

6.5.1.Register RW messages

Table 6.4: Dump Register RW message description

Condition	This message is output when registers are accessed and parameter PSIS011_DumpRegisterRW is set “true”.
Output	This message is printed to standard output (console).
Format: Info: <hier_instance_name>: [<time>ps] REG [<reg_name>] R Size = <size> Addr = <reg_address> Data = <reg_value> Info: <hier_instance_name>: [<time>ps] REG [<reg_name>] W Size = <size> Addr = <reg_address> Data = <reg_value> : <old_value> => <new_value> **Example**: Info: PSIS011: [2900000 ps] REG [PSI5SPDDTPn] R Size= 4 Addr= 0xFFF502C0 Data= 0x1 Info: PSIS011: [3270000 ps] REG [IPSI5SPDDTPn] W Size= 4 Addr= 0xFFF502C0 Data= 0x1 : 0x00 => 0x01	
Tag name	**Description**
hier_instance_name	Hierarchy instance name of PSIS011 model is being used.

time	Simulation time
reg_name	Name of accessed register.
size	Register size.
address	Register address.
value	Register value.
old_value	Register's value before writing.
new_value	Register's value after writing.

6.5.2.Interrupt message

Table 6.5: Dump Interrupt message description

Condition	This message is output when there is interrupt and parameter PSIS011_DumpInterrupt is set “true”.
Output	This message is printed to standard output (console).
Format: Info: <hier_instance_name>: [<time>ps] <interrupt_name> is changed to <value> **Example**: Info: PSIS011: [2900000 ps] INT [int_psis_ch0] Assert.	
Tag name	**Description**
hier_instance_name	Hierarchy instance name of PSIS011 model is being used.
time	Simulation time
interrupt_name	Name of interrupt
value	Value of interrupt

6.5.3.Help messages

Table 6.6: Dump parameter help message description

Condition	This message is dumped out when command PSIS011_Help is called with “parameters” argument.
Output	This message is printed to standard output (console). The help message is used for Python Interface.
--- parameters --- PSIS011_MessageLevel (“PSIS011 instance”, “fatal|error|warning|info”) Set debug message level (default: fatal|error). PSIS011_DumpRegisterRW (“PSIS011 instance”, “true/false”) Enable/disable dumping access register (default: false). PSIS011_DumpInterrupt (“PSIS011 instance”, “true/false”) Enable/disable dumping interrupt information (default: false).	

Table 6.7: Dump command help message description

Condition	This message is dumped out when command PSIS011_Help is called with “commands” argument.
Output	This message is printed to standard output (console). The help message is used for Python Interface.
--- commands --- PSIS011_SetDNFDelay (“PSIS011 instance”, delay) Set delay time for Noise Filter in PSIS011 model. Default delay time is 0. PSIS011_DumpStatusInfo (“PSIS011 instance”) Dump information of the error status register of PSIS011 model. PSIS011_AssertReset (“PSIS011 instance”, “rst_name”, start_time, period) Assert and de-assert reset signal to the PSIS011 model. PSIS011_SetCLKFreq (“PSIS011 instance”, “clk_name”, freq, “unit”) Set clock frequency to model. PSIS011_GetCLKFreq (“PSIS011 instance”, “clk_name”) Get clock frequency of model. PSIS011_ForceRegister (“PSIS011 instance”, “reg_name”, chid, value) Force a register with setting value. PSIS011_ReleaseRegister (“PSIS011 instance”, “reg_name”, chid) Release a register from force	

value.
PSIS011_WriteRegister (“PSIS011 instance”, “reg_name”,chid, value) Write a value to a register.
PSIS011_ReadRegister (“PSIS011 instance”, “reg_name”, chid) Read value from a register.
PSIS011_ListRegister (“PSIS011 instance”) Dump name of all registers.

6.5.4.List of error and debugging messages

Table 6.8: Error and debugging message in PSIS011 model

No.	Type	Severity	Message	Description
1	Users	error	Invalid access address 0x%08X with access size %d bytes	User access to model's register with wrong aligned address %08X: address %d: number of access bytes
2	Users	error	Invalid access address 0x%08X	Users access the model with invalid address %08X: address
3	Users	error	Invalid access size: %d bytes	This message is dumped when users access to register with invalid size. %d: number of bytes
4	Users	error	Cannot find the object of <model name> class	Users call PythonIF with wrong object of <model name> class
5	Users	error	<command name> has too much arguments	Dump this message when number of input arguments is incorrect.
6	Internal	error	<command name> command needs an argument [true/false]	Dump this message when input argument is missed.
7	Users	error	Reading access size to %s at address 0x%08X is wrong: %d byte(s).	Users read the value from registers with invalid size. %s: register name %8X: address %d: number of bytes
8	Users	error	Writing access size to %s at address 0x%08X is wrong: %d byte(s).	Users write the value to registers with invalid size. %8X: address %d: number of bytes
9	Users	warning	The <model name>_<command name> has not any arguments	Users call PythonIF of <model name>_<command name> with any argument. The argument should be not input.
10	Users	warning	The arguments of <command name> are wrong	Users call PythonIF of <command name> with wrong arguments
11	Users	warning	The name (%s) of <model name>_Help argument is wrong (commands or parameters).	Users call <model name>_Help command with invalid argument. It must be “commands” or “parameters”
12	Users	warning	Register name <register_name> is invalid	Dump this message when register name is invalid.

13	Users	warning	Should read all bit in a register	Dump this message when users read register with access size less than supported access size.
14	Users	warning	%s forbids to read	Dump this message when a write-only bit is read. %s: bit name
15	Users	warning	%s forbids to write %X	Dump this message when a read-only bit is written value. %s: bit name %X: written value.
16	Users	warning	Cannot write register when clock PCLK is 0.	Register is written when any clock PCLK is 0
17	Users	warning	Cannot write 1 to reserved bit.	Users write 1 to reserved area in a register.
18	Users	warning	Cannot launch call-back function during reset period	Users write the value to the registers during reset period
19	Users	warning	Cannot write during reset period of PRESETn.	Dump this message when users write to register during reset period of PRESETn.
20	Users	warning	Cannot write to reserved area (%08X.	Users write data to reserved area which has no register at this address. %08X: address access.
21	Users	warning	%s is blocked writing from Bus I/F.	Access write to register which it is locked by <model name>_ForceRegister.
22	Users	warning	Frequency unit (%s) is wrong, frequency unit is set as unit Hz default.	Users call <model name>_SetCLKFreq with frequency unit is wrong. The frequency unit must be Hz, Khz, MHz, GHz.
23	Users	warning	Clock name %s is invalid.	Dump this message when users call <model_name>_SetCLKFreq or <model_name>_GetCLKFreq with wrong clock name. %s: invalid clock name.
24	Users	warning	Invalid argument: <command name> <argument name>	Users call <command name> with invalid argument
25	Users	warning	The reset name (%s) is wrong. It should be PRESETn, psis_rst_n, psis_mult_rst_n	Users call AssertReset with wrong reset name. Refer Parameters_Commands sheet for detail.
26	Users	warning	The software reset of <reset name> is called in the reset operation of the model. So it is ignored	Users call AssertReset with name is PRESETn or psis_rst_n or psis_mult_rst_n in period of according hard reset.

27	Users	warning	Cannot write <register name> in <mode name>.	A register is prohibited to write in a mode. But users try to write it. <register name>: name of register <mode name>: configuration or PSI5 or UART.
28	Users	warning	Cannot write <bit name> in <mode name>.	A bit in register is prohibited to write in a mode. But users try to write it. <bit name>: bit name in a register <mode name>: configuration or PSI5 or UART.
29	Users	warning	Cannot write prohibited value (%d) to PSI5SPRCF1%d.%s bit. Value 3 is written to this bit	In each channel, the FPKT bit in PSI5SPRCF1n register is prohibited to set value less than 3. But users try to write it. %d: invalid value. %d: channel index %s: bit name of FPKT
30	Users	warning	Cannot write prohibited value (%d) to PSI5SPRCF1%d.%s bit. Value 6 is written to this bit	In each channel, the FPKT bit in PSI5SPRCF1n register is prohibited to set value larger than 6. But users try to write it. %d: invalid value. %d: channel index %s: bit name of FPKT
31	Users	warning	Cannot write prohibited value (%d) to PSI5SPRCF2%d.%s bit. Value 8 is written to this bit.	In each channel, the PAYLD bit in PSI5SPRCF2n register is prohibited to set value less than 8. But users try to write it. %d: invalid value. %d: channel index %s: bit name of PAYLD
32	Users	warning	Cannot write prohibited value (%d) to PSI5SPRCF2%d.%s bit. Value 28 is written to this bit.	In each channel, the PAYLD bit in PSI5SPRCF2n register is prohibited to set value larger than 28. But users try to write it. %d: invalid value. %d: channel index %s: bit name of PAYLD
33	Users	warning	Cannot write <register_name> when PSI5SPDDS%d.DDSRSTS is 1.	In each channel from 1-7, the PSI5SPDDTPn register and PSI5SPDDDn register are prohibited to write when PSI5SPDDSn.DDSRSTS is 1. But users try to write it in that case. <register name> register name of PSI5SPDDTPn register or PSI5SPDDDn register %d: channel index

34	Users	warning	Cannot write PSI5SPUBPR.SCKDIV bit and PSI5SPUBPR.SCKPRS bit when PSI5SPUBCE.SCKEN bit is 1	When PSI5SPUBCE.SCKEN bit is 1, it is prohibited to set PSI5SPUBPR.SCKDIV bit and PSI5SPUBPR.SCKPRS bit. But users try to write it in that case.
35	Users	warning	Cannot write PSI5SPTFD<index> register when PSI5SPTFS.TXSTS is 1.	When PSI5SPTFS.TXSTS is 1, it is prohibited to write to PSI5SPTFD1 register or PSI5SPTFD2 register. But users try to write it in that case. <index> is 1, 2 (for PSI5SPTFD1/2 register).
36	Users	warning	Wrong setting for PSI5SPTPS.TSPRSL bit (= %d) makes malfunction. It should be %d (because PCLK = %dMHz).	The PSI5SPTPS.TSPRSL bit must be set to make 1us from PCLK clock. Example: When PCLK = 80MHz, this bit is set 80. But users do not set like this. %d: invalue written value to PSI5SPTPS.TSPRSL bit. %d: value should be set to PSI5SPTPS.TSPRSL bit. %d: frequency of PCLK in MHz unit.
37	Users	warning	Wrong setting for PSI5SPTPS.TSPRSU bit (= %d > 1000). Value 1000 is written to this bit (due to max 1ms).	The PSI5SPTPS.TSPRSU bit must be set to make maximum 1ms from 1us (setting in PSI5SPTPS.TSPRSL bit). So the maximum value for this bit is 1000. But users write a value larger than 1000 to this bit. %d: invalid written value to PSI5SPTPS.TSPRSU bit.
38	Users	warning	Invalid received \"%s\" in Idle Reception State.	After receiving an IDLE strobe (strobe in RX_CONTROL port is IDLE), the next trobe of UART packet should be START. But in this case, model receive another strobe (not START). %s: Invalid strobe (STOP, ABORT).
39	Users	warning	Invalid received \"%s\" in Start Reception State. UART packet is aborted.	After receiving a START strobe, the model waits a certain period to receive start bit, data, parity bit (if any). But in this waiting time, the strobe in RX_CONTROL port is changed to another strobe (not START). In this case, the UART frame is not received successfully. %s: Invalid strobe (IDLE, STOP, ABORT).

40	Users	warning	Invalid received \"%s\" in Stop Reception State.	After finish waiting a period to receive start bit, data, parity bit (if any), the model waits to receive STOP strobe. But in this time, the strobe in RX_CONTROL port is changed to another strobe (not STOP). In this case, UART frame is received, but a framing error occurs. %s: Invalid strobe (IDLE, START, ABORT).
41	Users	warning	Cannot restore PSI5 frame in channel %d due to invalid setting PSI5SPRCF2<channel index>.F<frame index>PAYLD (=%d), PSI5SPRCF1<channel index>.RFCPS (=%d), PSI5SPRCF1<channel index>.F<frame index>PKT (=%d).	Users set invalid value to PSI5SPRCF1n and PSI5SPRCF2n register (not match with table 25.185. List of PRCF1n.FmPKT[2:0] settings, HWM). <channel index>: channel index from 0-7 <frame index>: frame index from 1-6 %d: value of PAYLD bit %d: value of RFCPS bit %d: value of FPKT bit.
42	Users	warning	Invalid channel index (%d). It must be from 0 to 7.	Dump this message when users call Python command ForceRegister, ReleaseRegister, WriteRegister, ReadRegister on register of channel with invalid channel index. %d: wrong channel index (out of range 0-7)
43	Users	info	(UART mode) Receive UART frame = %X.	Dump this message when receiving data in UART mode. %X: received data
44	Users	info	(PSI5 mode) Receive UART frame no.%d = %X.	Dump this message when receiving data in PSI5 mode. %d: index of UART frame in PSI5 packet. %X: received data
45	Users	info	(UART mode) Transmit <strobe name>, TX_DATA = %X, TX_CONTROL = %X	Dump this message when transmitting data in UART mode <strobe name>: strobe name in TX_CONTROL port. It is either IDLE, START, STOP, or ABORT. %X: transmitted data in TX_DATA port %X: Control value in TX_CONTROL port

46	Users	info	(PSI5 mode) (CPU Tx cmd) Transmit <strobe name>, TX_DATA = %X, TX_CONTROL = %X	Dump this message when transmitting CPU Tx command data in PSI5 mode. <strobe name>: strobe name in TX_CONTROL port. It is either IDLE, START, STOP, or ABORT. %X: transmitted data in TX_DATA port %X: Control value in TX_CONTROL port
47	Users	info	(PSI5 mode) (<channle index>) Transmit <strobe name>, TX_DATA = %X, TX_CONTROL = %X	Dump this message when transmitting ECU-to-sensor data in PSI5 mode. <channel index>: channel index is from 1-7 <strobe name>: strobe name in TX_CONTROL port. It is either IDLE, START, STOP, or ABORT. %X: transmitted data in TX_DATA port %X: Control value in TX_CONTROL port
48	Users	info	WDT error in channel %d	Dump this message when WDT counts down to 0 in a channel. %d: channel index from 0-7
49	Users	info	INT [%s] Assert.	Dump this message when any interrupt is changed value to 1, and the PSIS011_DumpInterrupt is TRUE. %s: name of interrupt
50	Users	info	INT [%s] De-assert.	Dump this message when any interrupt is changed value to 0 and the PSIS011_DumpInterrupt is TRUE. %s: name of interrupt
51	Users	info	dma_psis_ch%d_tx is %d	Dump this message when any DMA request for TX of channel 1-7 is change to 1.
52	Users	info	dma_psis_ch%d_rx is %d	Dump this message when any DMA request for RX of channel 0-7 is change to 1.
53	Users	info	The Sofware reset is asserted.	Dump this message when users write 1 to PSI5SPUSWR.SWRST bit to start software reset.
54	Users	info	The Sofware reset is de-asserted.	Dump this message when software reset is de-asserted. Software reset period is 8 cycles of PCLK. After 8 cycles PCLK from asserting time, software reset is automatically de-asserted.
55	Users	info	Move to <mode name> mode	Dump this message when model changes mode

				<mode name>: Configuration mode or UART mode or PSI5 mode.
56	Users	info	Issue request of CPU Tx Command to TX Request arbiter	Dump this message when users write data to PSI5SPTFST register to start transmit CPU Tx Command.
57	Users	info	Issue request of channel%d to TX Request arbiter	When users write data to PSI5SPDDD register, at next sync pulse, this message is dumped to indicate that the request to transmit ECU-to-sensor data of channel is sent to arbiter. %d: channel index.
58	Users	info	Initialize %s (%08x)	This message is dumped when initializing value of register during reset period. %s: register's name. %08X: register value.
59	Users	info	%s frequency is zero	Dump this message when any clock is set frequency 0.
60	Users	info	Reset period of %s is over.	Reset period of a reset name which is set by AssertReset is over.
61	Users	info	The model is reset by AssertReset command of %s.	Users call AssertReset command for a reset name and reset operation is accepted after specified time at first argument of <model name>_AssertReset.
62	Users	info	The model will be reset (%s) for %f ns after %f ns.	Users call AssertReset command for PRESETn, or psis_rst_n, or psis_mult_rst_n with start reset time and reset period.
63	Users	info	The reset port %s is asserted.	Users activate PRESETn, or psis_rst_n, or psis_mult_rst_n port
64	Users	info	The reset port %s is de-asserted.	Users deactivate PRESETn, or psis_rst_n, or psis_mult_rst_n port
65	Users	info	%s frequency is %0.0f %s	Users set frequency to PCLK, or psis_clk, or psis_mult_clk clock.
66	Users	info	<model name>_DumpInterrupt %s	This message is dumped when users call <model name>_DumpInterrupt without argument.
67	Users	info	(PSI5 mode) Ignore UART frame no.%d = %X.	Dump this message when ignoring data in PSI5 mode (because FPKT = 0). %d: index of UART frame in PSI5 packet. %X: received data

6.6.Define macro and template

(1) In this design, there is no macro.

(2) In this design, there is no template.

7.Flow diagrams

7.1.Sequence diagram of transmission in PSI5 mode

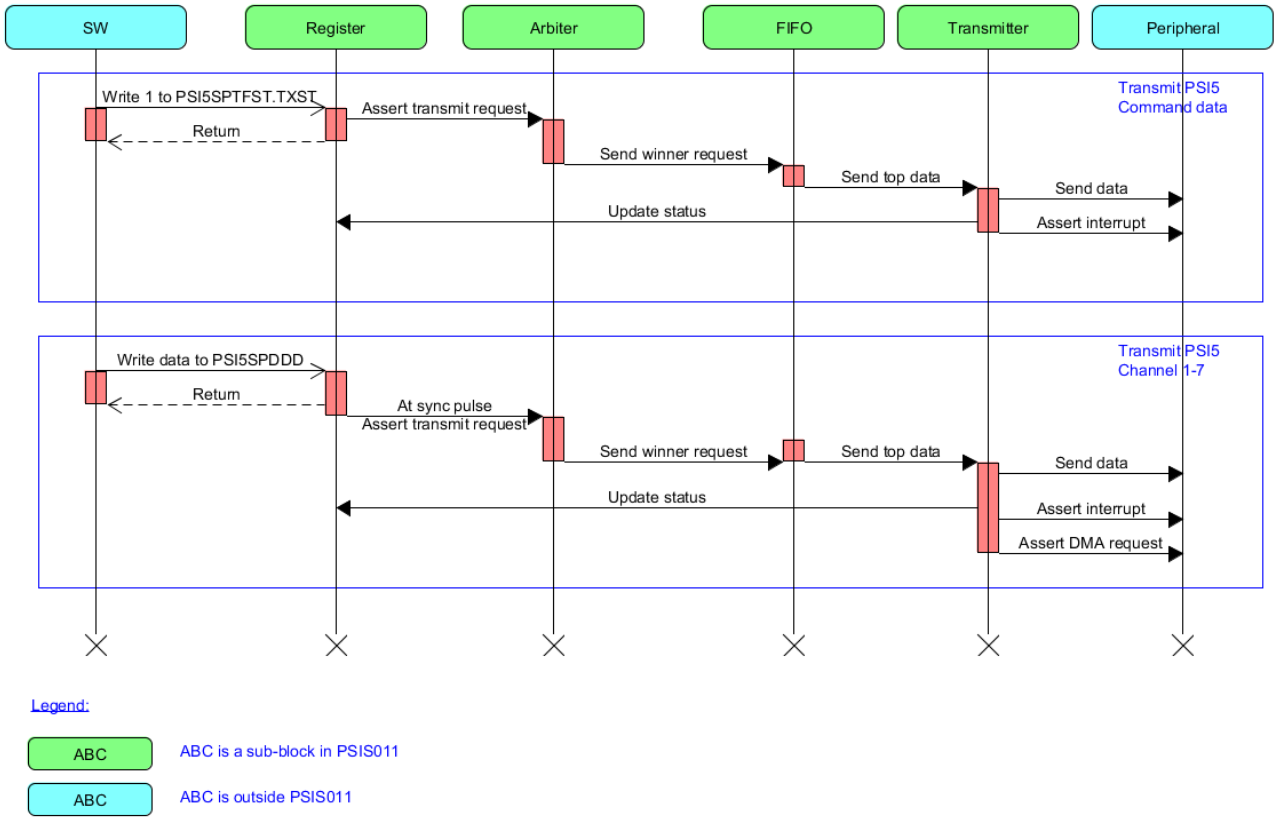


Figure 7.1: Sequence diagram of transmission in PSI5 mode

Explanation:

(1) Figure above describes sequence diagram of transmission in PSI5 mode for Command data, and ECU-to-Sensor data (channel 1-7).

(2) Transmit Command data:

(2.1) Command data is set in PSI5SPTFD1, and PSI5SPTFD2 register.

(2.2) The number of packet to send is set in PSI5SPTFNM register.

(2.3) The PSI5SPTFST.TXST is written 1 to start transmit command data. A request to start is sent to Arbiter.

(2.4) If request of command data is winner of arbitration, it is sent to Request-FIFO.

(2.5) The Transmitter transmits data of top request in Request-FIFO to outside via TX_CONTROL port and TX_DATA port.

(2.6) The interrupt, DMA request is asserted according status.

(3) Transmit ECU-to-Sensor data (channel 1-7):

(3.1) Frame type is set in PSI5SPDDTP register.

(3.2) ECU-to-Sensor data is set in PSI5SPDDD register. At the next Sync Pulse, a

transmit request is sent to Arbiter.

(3.3) If request of such ECU-to-Sensor data is winner of arbitration, it is sent to Request-FIFO.

(3.4) The Transmitter transmits data of top request in Request-FIFO to outside via TX_CONTROL port and TX_DATA port.

(3.5) The interrupt, DMA request is asserted according status.

(4) Refer to figure below for detail timing of transmission in PSI5 mode.

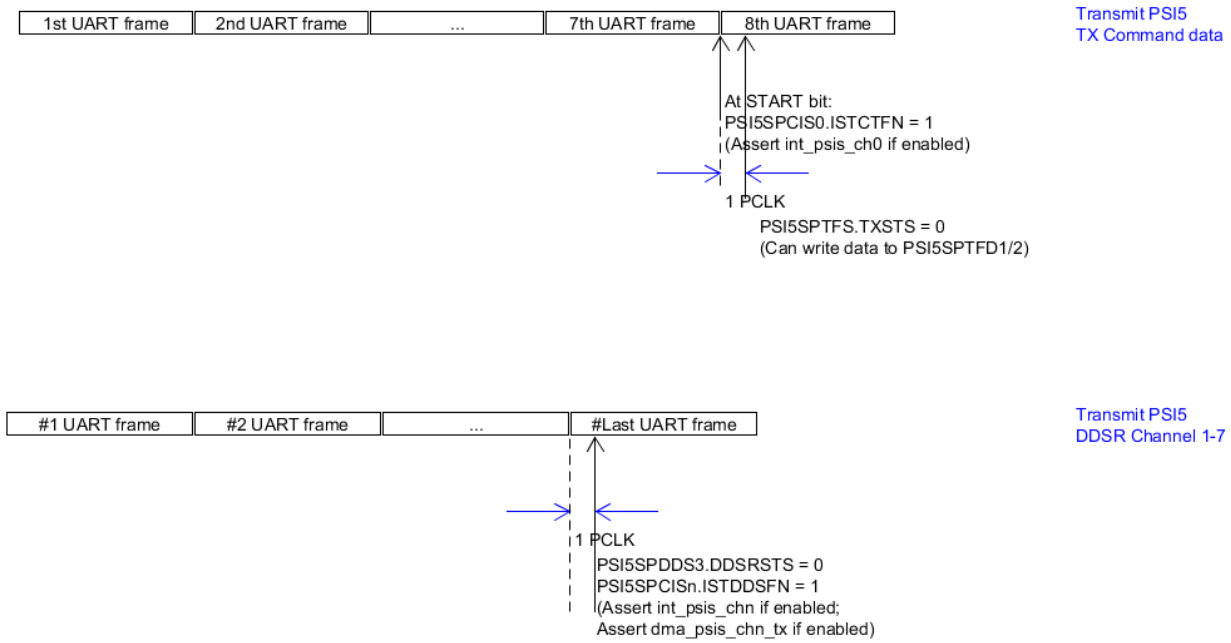


Figure 7.2: Timing chart of transmission in PSI5 mode

7.2.Sequence diagram of transmission in UART mode

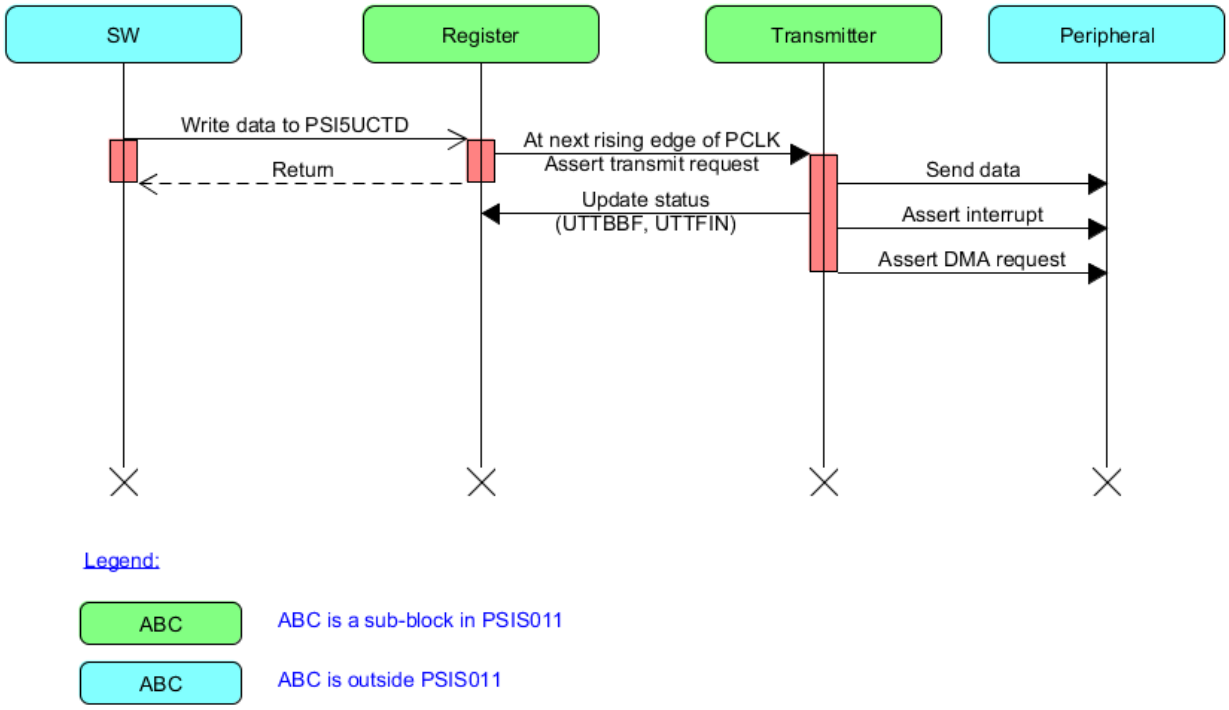


Figure 7.3: Sequence diagram of transmission in UART mode

Explanation:

(1) Figure above describes sequence diagram of transmission in UART mode.

(2) Data is set in PSI5UCTD register.

(3) At the next rising edge of PCLK, a transmit request is asserted.

(4) The Transmitter transmits the data to outside via TX_CONTROL port and TX_DATA port.
The status bits (UTTBBF, UTTFIN) are updated.

(5) The interrupt, DMA request is asserted according status.

(6) Refer to figure below for detail timing of transmission in UART mode.

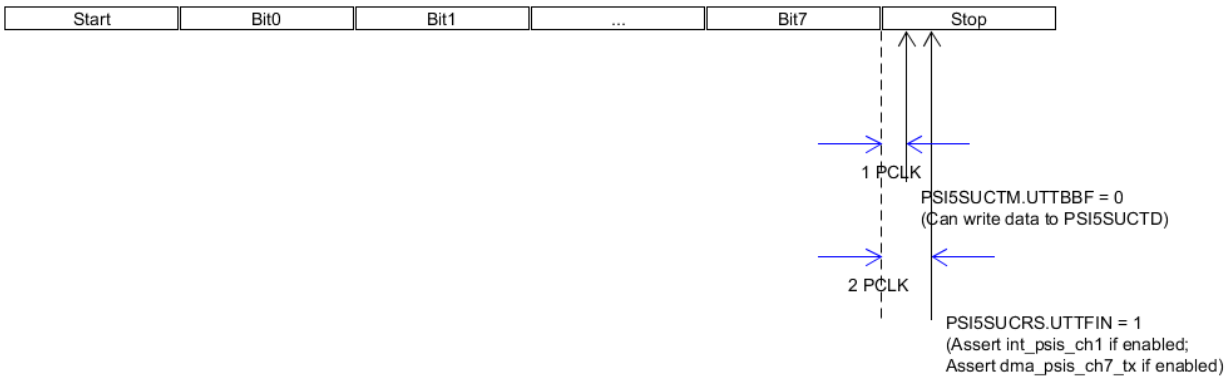


Figure 7.4: Timing chart of transmission in UART mode

7.3.Sequence diagram of reception in PSI5 mode

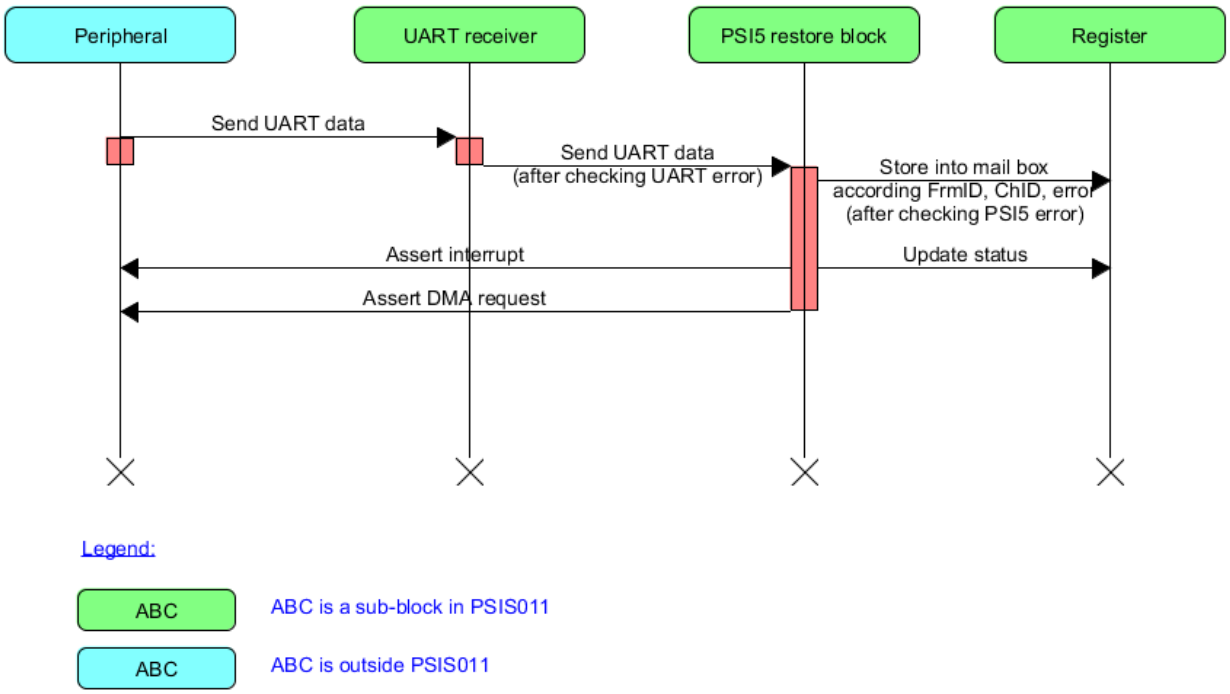


Figure 7.5: Sequence diagram of reception in PSI5 mode

Explanation:

(1) Figure above describes sequence diagram of reception in PSI5 mode.

(2) Models receives UART frame from outside via RX_CONTROL port, and RX_DATA port.

(3) UART Receiver block checks parity error, UART framing error; then sends UART data and such error to PSI5 Restore block.

(4) The PSI5 Restore block checks PSI5 errors: frame lack error, frame excess error, XCRC error, CRC/Parity error; stores data to register.

(5) The data is stored in Mail Box register based on the error occurs or not, and based on the Channel ID, the Frame ID.

(5.1) If any error occurs, data is stored in Channel0, Frame 2.

(5.2) If no error, data is stored in according Channel ID, Frame ID.

Note: In synchronous mode, Frame ID is got from header. In asynchronous mode, Frame ID is got from FrameID-counter.

(6) The status bits are updated according current status.

(7) The interrupt, DMA request is asserted according channels/status.

(8) Refer to figure below for detail timing of reception in PSI5 mode.

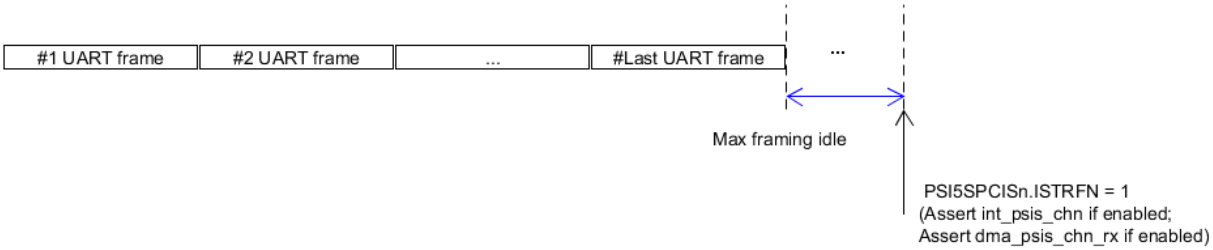


Figure 7.6: Timing chart of reception in PSI5 mode

7.4.Sequence diagram of reception in UART mode

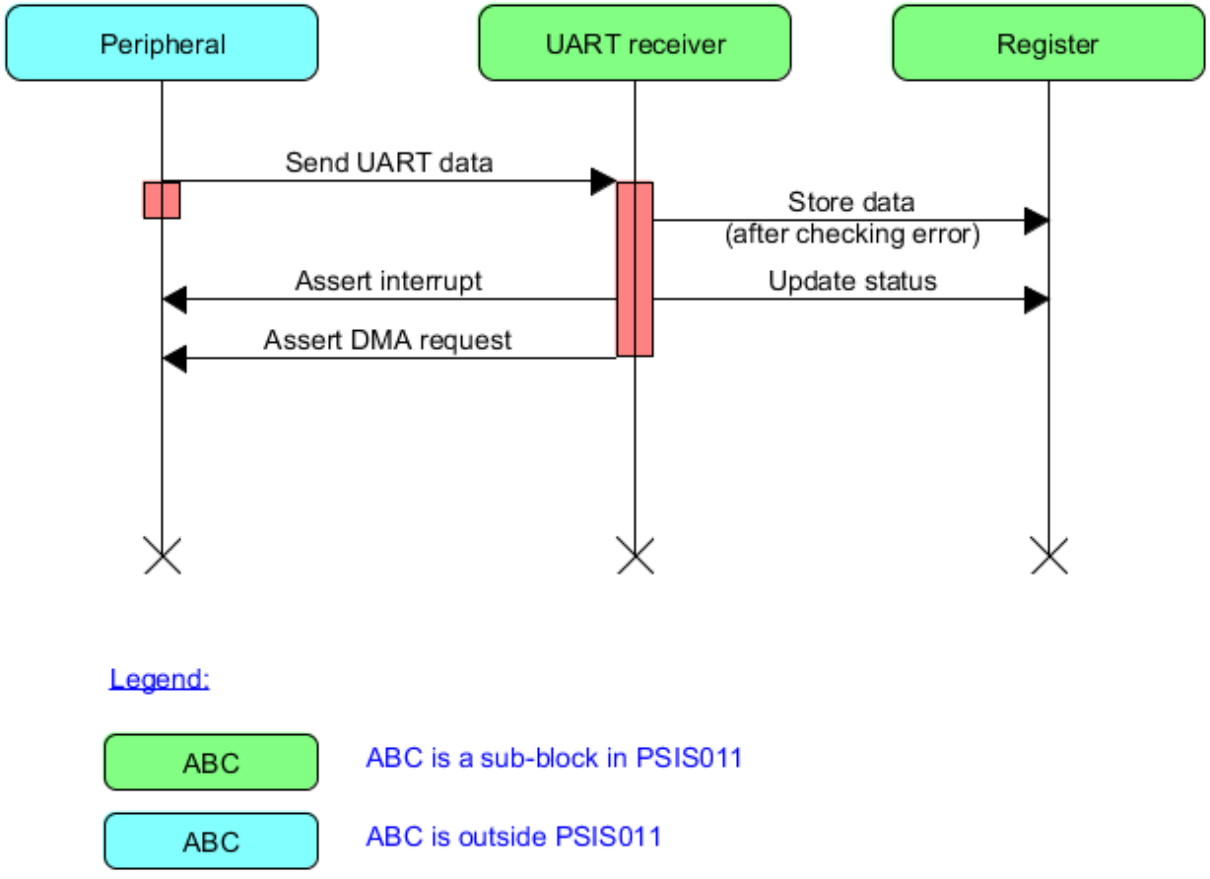


Figure 7.7: Sequence diagram of reception in UART mode

Explanation:

(1) Figure above describes sequence diagram of reception in UART mode.

(2) Models receives UART frame from outside via RX_CONTROL port, and RX_DATA port.

(3) UART Receiver block checks parity error, UART framing error, overrun error; then stores data to PSI5SUCRD register.

(4) The status bits are updated according current status.

(5) The interrupt, DMA request is asserted according channels/status.

(6) Refer to figure below for detail timing of reception in UART mode.

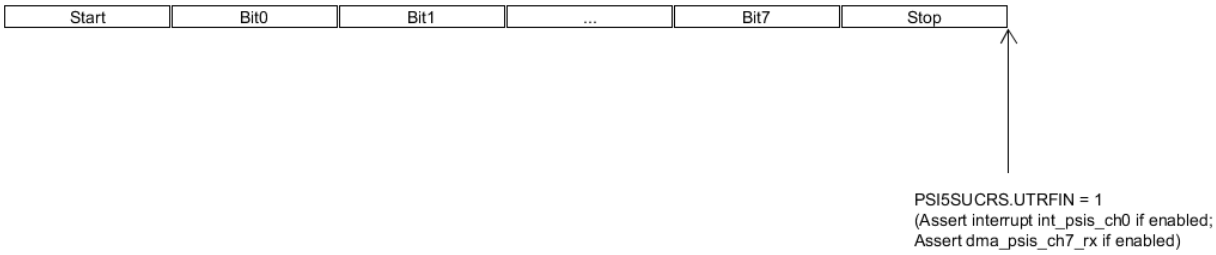


Figure 7.8: Timing chart of reception in UART mode

7.5.Flow of abnormal transmission in PSI5 mode

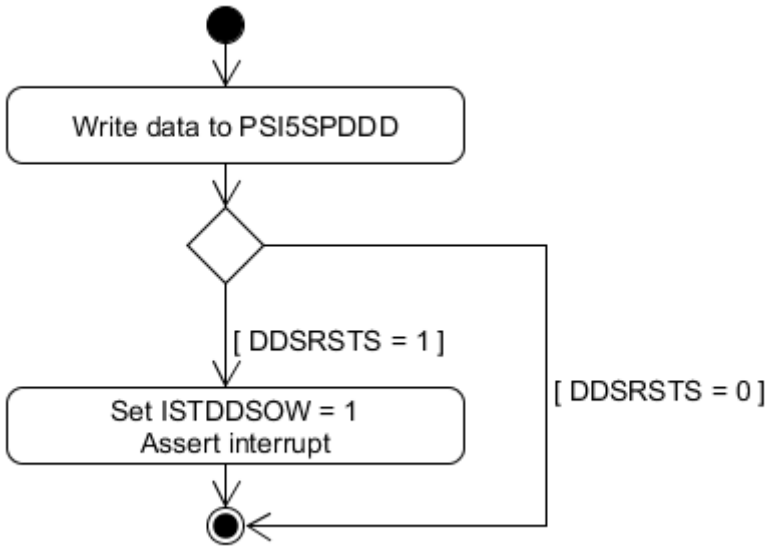


Figure 7.9: Flow of abnormal transmission in PSI5 mode

Explanation:

(1) Figure above describes operation of abnormal transmission in PSI5 mode.

(2) It occurs when writing into PSI5SPDDD register (in channel 1-7) and the DDSRSTS bit is 1.

(3) Status bit ISTDDSOW is updated to 1.

(4) Interrupt is asserted according channel index, and current status.

7.6.Flow of abnormal transmission in UART mode

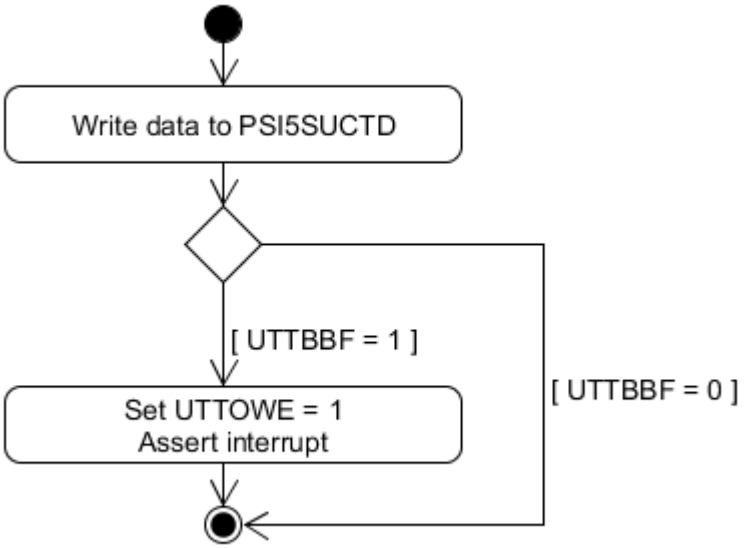


Figure 7.10: Flow of abnormal transmission in UART mode

Explanation:

(1) Figure above describes operation of abnormal transmission in UART mode.

(2) It occurs when writing into PSI5SUCTD register and the UTTBBF bit is 1.

(3) Status bit UTTOWE is updated to 1.

(4) Interrupt is asserted according current status.

7.7.Flow of abnormal reception in PSI5 mode

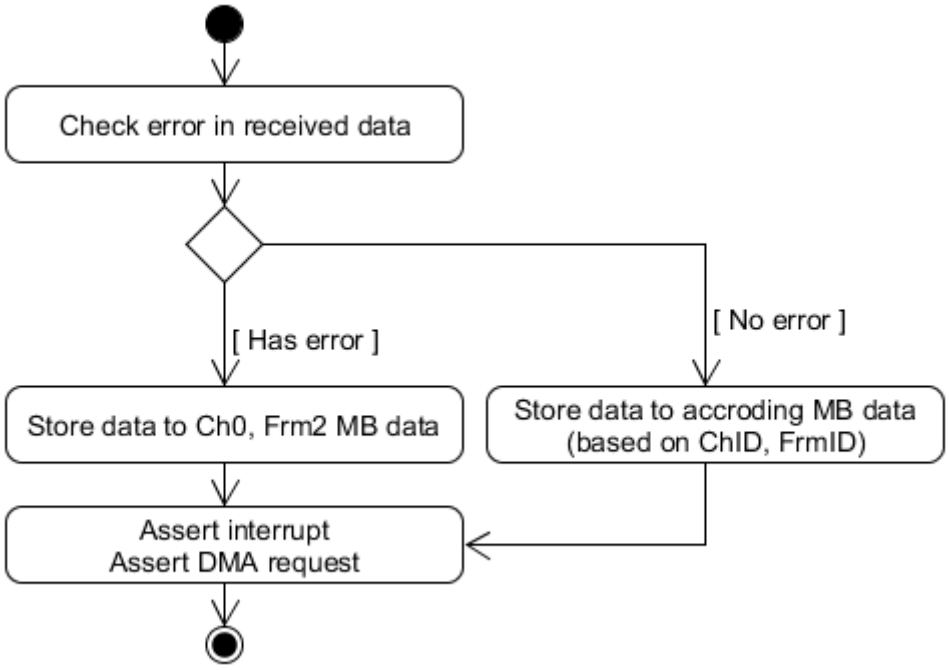


Figure 7.11: Flow of abnormal reception in PSI5 mode

Explanation:

(1) Figure above describes operation of abnormal reception in PSI5 mode.

(2) It occurs when there is any UART error (at UART Receiver), or PSI5 error (at PSI5 Restore block), WDT error occurs, or frame error (Frame ID = 6, 7).

When having error, the PSI5 restored data is stored in Mail Box of channel 0, frame 2.

When WDT error occurs, the frame after WDT error is not stored.

(3) If there is no error (normal case), the PSI5 restored data is stored in Mail Box of channel index, frame ID which defined in header.

(4) Interrupt and DMA request are asserted according channel index, and current status.

7.8.Flow of abnormal reception in UART mode

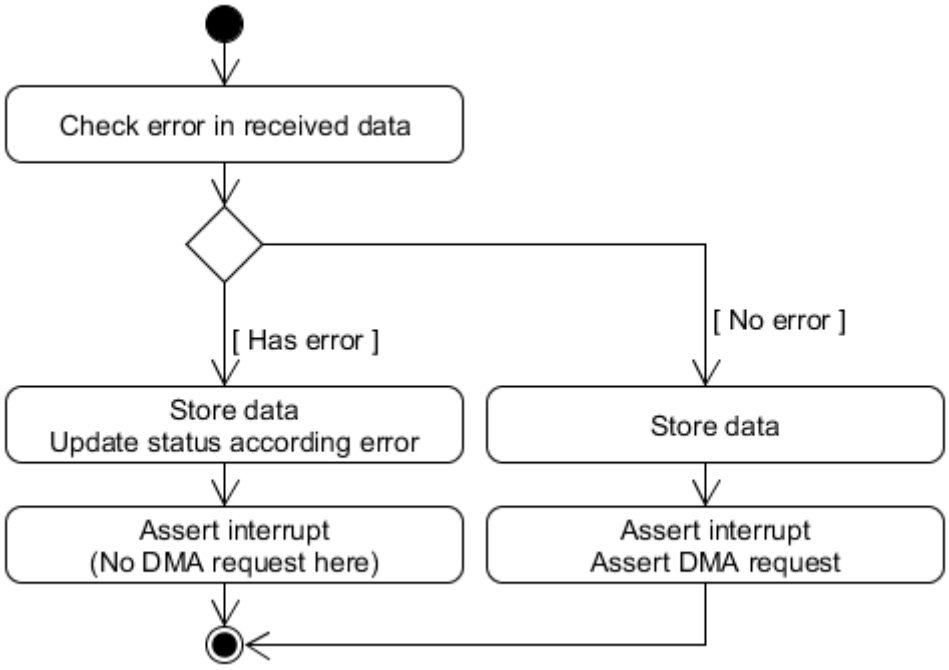


Figure 7.12: Flow of abnormal reception in UART mode

Explanation:

(1) Figure above describes operation of abnormal reception in UART mode.

(2) It occurs when there is any UART error (at UART Receiver). When having error:

(2.1) The UART frame is stored in PSI5SUCRD register.

(2.2) Status bit is updated according current error. The error can be parity error, framing error, overrun error. If both parity error and framing error occur, framing error is assumed.

(2.3) Interrupt is asserted according current status. Note that, there is no DMA request for this case.

(3) If there is no error (normal case):

(3.1) The UART frame is stored in PSI5SUCRD register.

(3.2) Interrupt and DMA request are asserted.

7.9.Flow of generating Sync Pulse

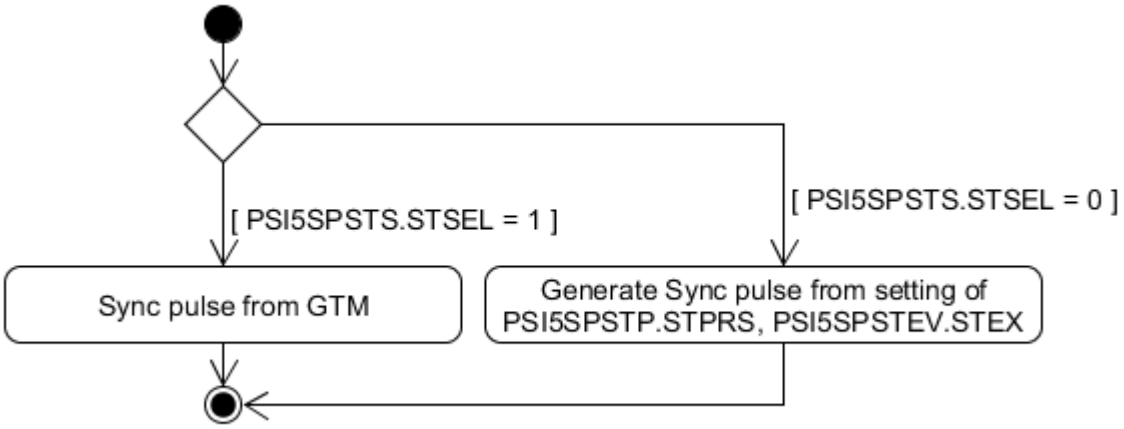


Figure 7.13: Flow of generating Sync Pulse

Explanation:

(1) Figure above describes operations about generating Sync Pulse in channel 1-7.

Note that, channel 0 always operates in asynchronous mode.

(2) The Sync Pulse can be generated from 2 sources.

(2.1) If the PSI5SPSTS.STSEL is 1, the Sync Pulse is the sync trigger from GTM.

(2.2) If the PSI5SPSTS.STSEL is 0, the Sync Pulse is generated from internal model.

A counter counts down from PSI5SPSTEV.STEX value to 0. The pre-scale of this counter is defined in PSI5SPSTP.STPRS bit. When this counter counts to 0, a Sync Pulse is generated.

This counter is started counting from the time enable operation (writing 1 to PSI5SPUOEB.OPEN bit).

7.10.Flow of PRESETn

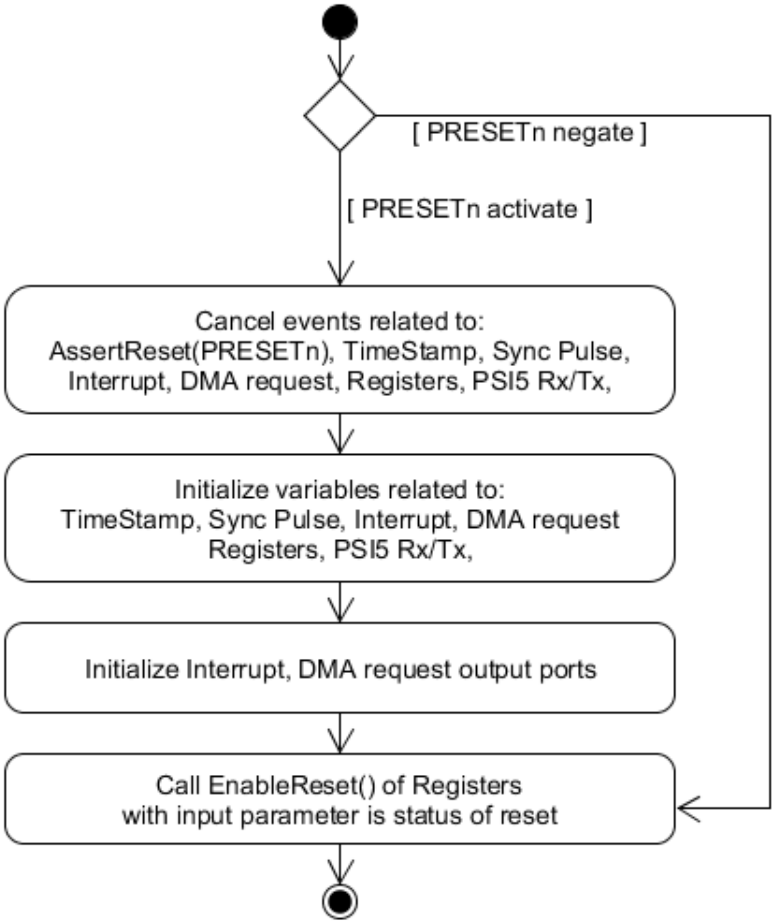


Figure 7.14: Flow of reset PRESETn

Explanation:

(1) Figure above describes operation of PSIS011 model when the PRESETn port is changed value.

(2) If PRESETn is activated:

(2.1) Events related to AssertReset command for PRESETn port are canceled.

(2.2) Events related to Time Stamp, Sync Pulse generator, Interrupt, DMA request, updating registers, PSI5 Rx/Tx operation, are canceled.

(2.3) Variable related to Time Stamp, Sync Pulse generator, Interrupt, DMA request, Registers, PSI5 Rx/Tx operation, are initialized.

(2.4) Interrupt ports, DMA request ports are initialzed.

(2.5) The EnableReset() function of sub-instance is called with input parameter is the status of reset.

(3) If PRESETn is negated, the EnableReset() function of sub-instance is called with input parameter is the status of reset.

7.11.Flow of psis_rst_n

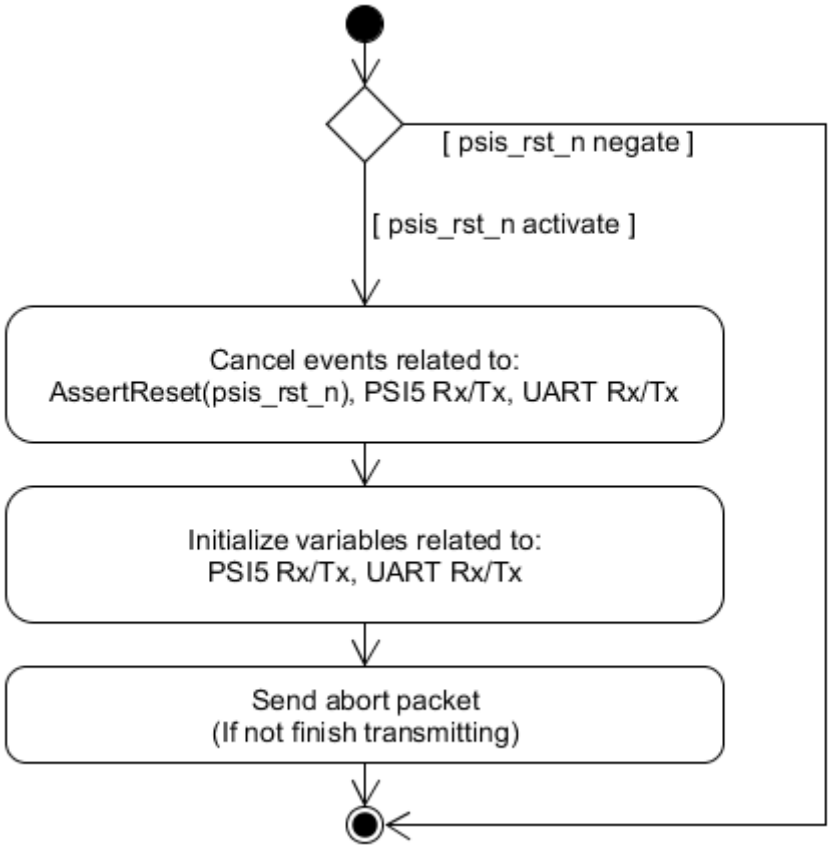


Figure 7.15: Flow of reset psis_rst_n

Explanation:

(1) Figure above describes operation of PSIS011 model when the psis_rst_n port is changed value.

(2) If psis_rst_n is activated:

(2.1) Events related to AssertReset command for psis_rst_n port are canceled.

(2.2) Events related to PSI5 Rx/Tx operation, UART Rx/Tx operation, are canceled.

(2.3) Variable related to PSI5 Rx/Tx operation, UART Rx/Tx operation, are initialized.

(2.4) An “abort” packet is sent if not finish transmitting. “Abort” packet means that UART packet having strobe bit is 2’b11 (in TX_CONTROL port).

(3) If psis_rst_n is negated, do nothing.

7.12.Flow of psis_mult_rst_n

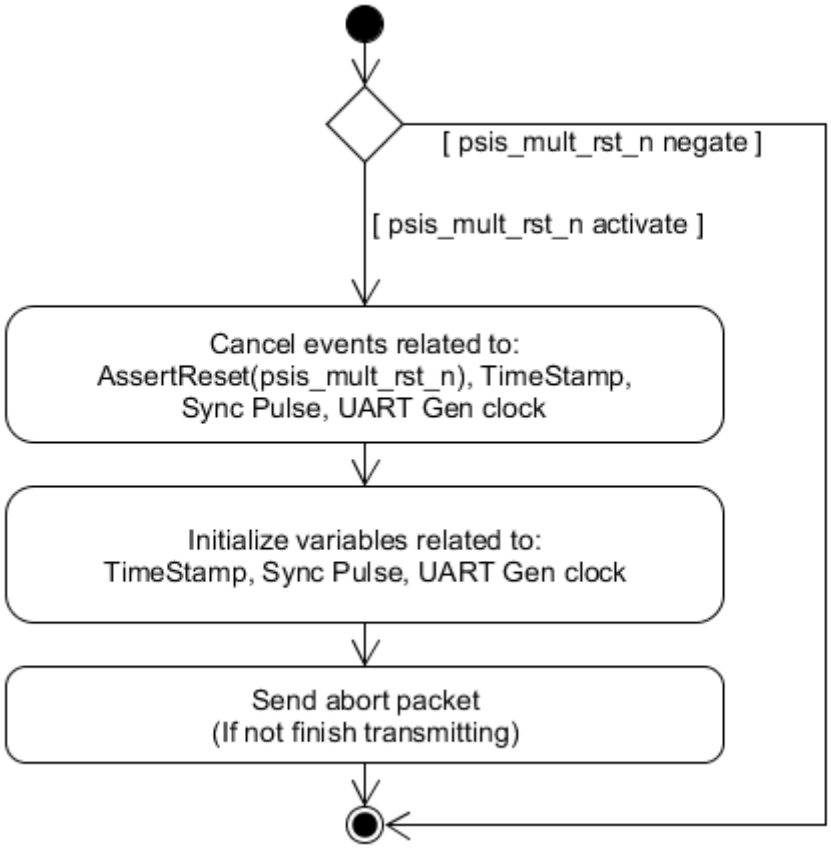


Figure 7.16: Flow of reset psis_mult_rst_n

Explanation:

(1) Figure above describes operation of PSIS011 model when the psis_mult_rst_n port is changed value.

(2) If psis_mult_rst_n is activated:

(2.1) Events related to AssertReset command for psis_mult_rst_n port are canceled.

(2.2) Events related to Time Stamp, Sync Pulse generator, UART clock generator, are canceled.

(2.3) Variable related to Time Stamp, Sync Pulse generator, UART clock generator, are initialized.

(2.4) An “abort” packet is sent if not finish transmitting. “Abort” packet means that UART packet having strobe bit is 2’b11 (in TX_CONTROL port).

(3) If psis_mult_rst_n is negated, do nothing.

7.13.Flow of SW reset

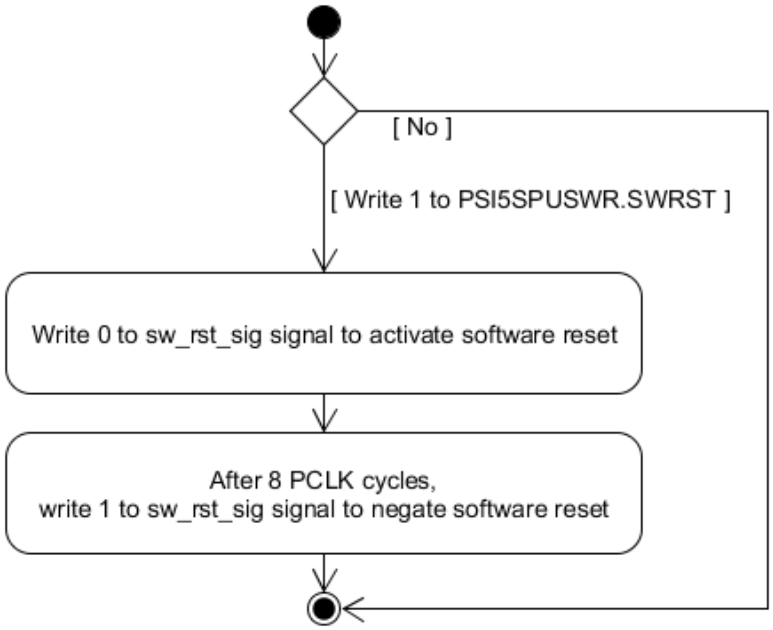


Figure 7.17: Flow of SW reset

Explanation:

(1) Figure above describes operation of PSIS011 model when users write 1 to PSI5SPUSWR.SWRST bit to trigger a SW reset.

(2) The sw_rst_sig signal is written 0 to activate software reset. This signal causes model process reset for 3 domains of PCLK, psis_clk, psis_mult_clk.

(3) After 8 PCLK cycles, the sw_rst_sig signal is written 1 to negate software reset. If other reset ports (PRESETn, psis_rst_n, psis_mult_rst_n) are not activated, this signal causes model negate reset for 3 domains of PCLK, psis_clk, psis_mult_clk.

7.14.Flow of stopping PCLK

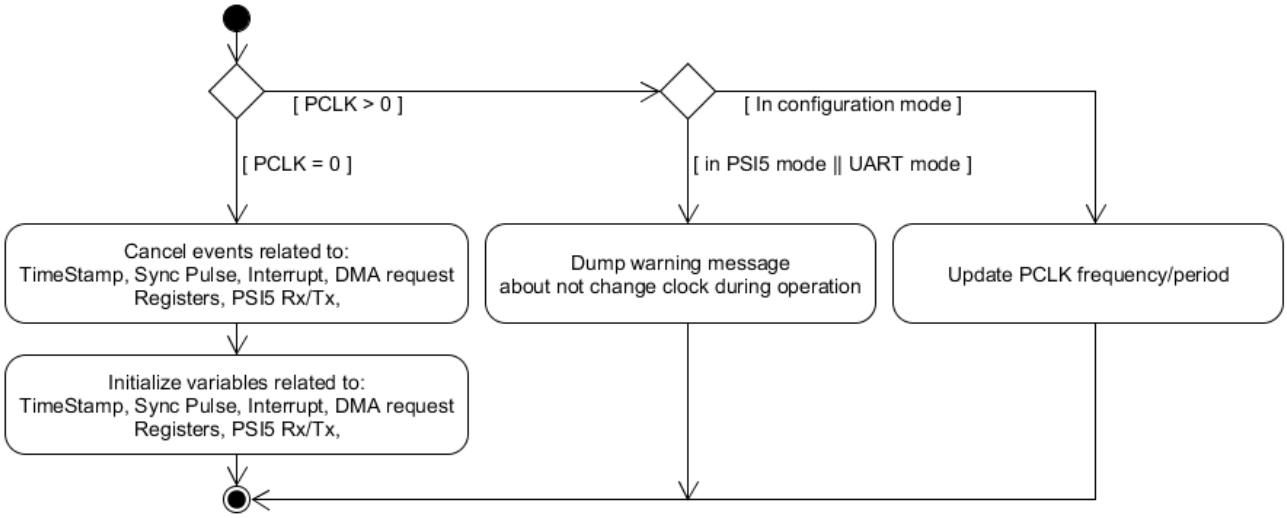


Figure 7.18: Flow of stop PCLK clock

Explanation:

(1) Figure above describes operation of PSIS011 model when stopping PCLK clock.

(2) When PCLK clock is stopped:

(2.1) Events related to Time Stamp, Sync Pulse generator, Interrupt, DMA request, updating registers, PSI5 Rx/Tx operation, are canceled.

(2.2) Variable related to Time Stamp, Sync Pulse generator, Interrupt, DMA request, Registers, PSI5 Rx/Tx operation, are initialized.

(3) Note: PCLK clock is not changed in PSI5 mode, UART mode. A warning message is dumped if change PCLK clock in such modes above.

(4) PCLK clock is only be changed in configuration mode. The frequency/period variable is updated according the change of PCLK clock.

7.15.Flow of stopping psis_clk

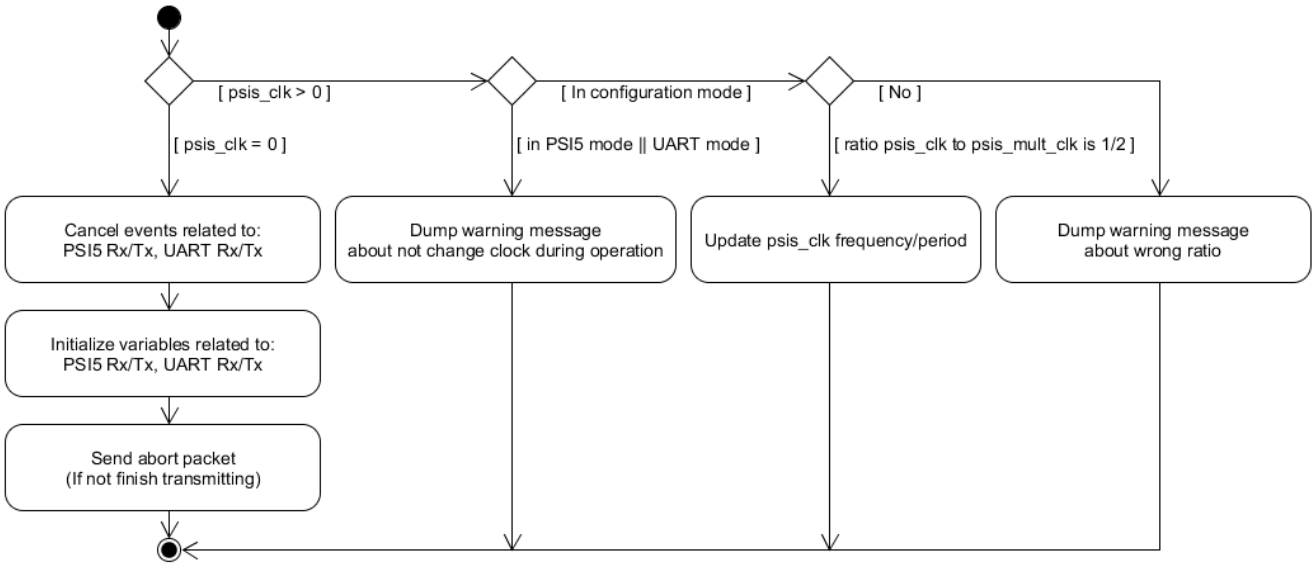


Figure 7.19: Flow of stop psis_clk clock

Explanation:

(1) Figure above describes operation of PSIS011 model when stopping psis_clk clock.

(2) When psis_clk clock is stopped:

(2.1) Events related to PSI5 Rx/Tx operation, UART Rx/Tx operation, are canceled.

(2.2) Variable related to PSI5 Rx/Tx operation, UART Rx/Tx operation, are initialized.

(2.3) An “abort” packet is sent if not finish transmitting. “Abort” packet means that UART packet having strobe bit is 2’b11 (in TX_CONTROL port).

(3) psis_clk clock is not changed in PSI5 mode, UART mode. A warning message is dumped if change psis_clk clock in such modes above.

(4) psis_clk clock is only be changed in configuration mode. The frequency/period variable is updated according the change of psis_clk clock. The frequency ratio of psis_clk to psis_mult_clk must always be 1/2. If not, a warning message is dumped.

7.16.Flow of stopping psis_mult_clk

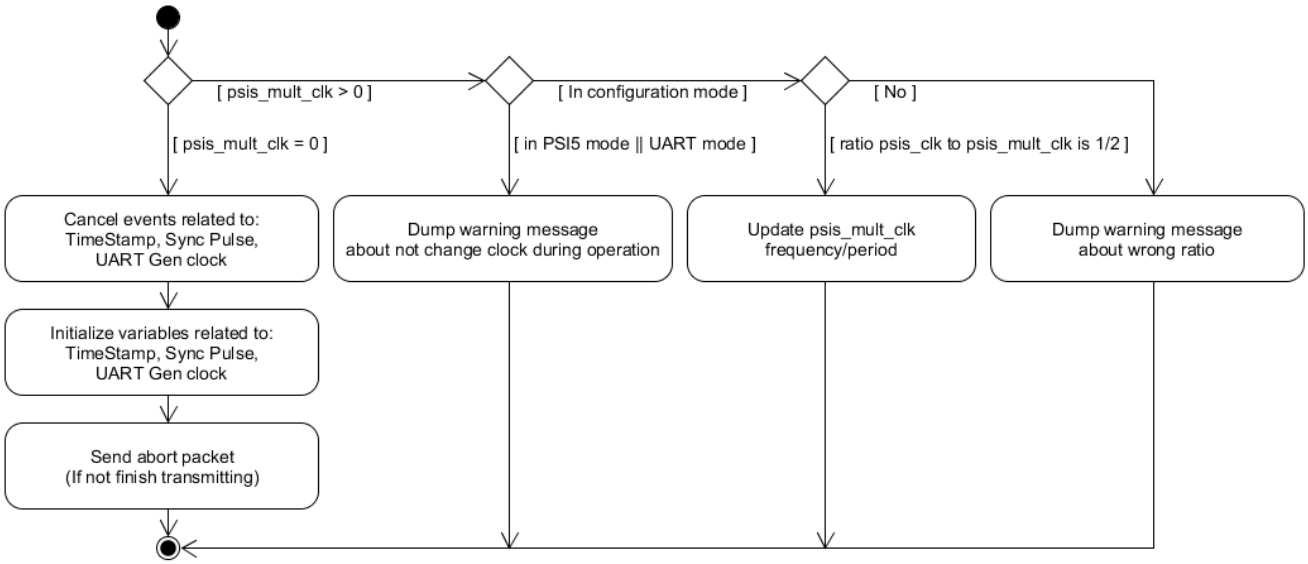


Figure 7.20: Flow of stop psis_mult_clk clock

Explanation:

(1) Figure above describes operation of PSIS011 model when stopping psis_mult_clk clock.

(2) When psis_mult_clk clock is stopped:

(2.1) Events related to Time Stamp, Sync Pulse generator, UART clock generator, are canceled.

(2.2) Variable related to Time Stamp, Sync Pulse generator, UART clock generator, are initialized.

(2.3) An “abort” packet is sent if not finish transmitting. “Abort” packet means that UART packet having strobe bit is 2’b11 (in TX_CONTROL port).

(3) psis_mult_clk clock is not changed in PSI5 mode, UART mode. A warning message is dumped if change psis_mult_clk clock in such modes above.

(4) psis_mult_clk clock is only be changed in configuration mode. The frequency/period variable is updated according the change of psis_clk clock. The frequency ratio of psis_clk to psis_mult_clk must always be 1/2. If not, a warning message is dumped.

7.17.Flow of arbitration for transmission in PSI5 mode

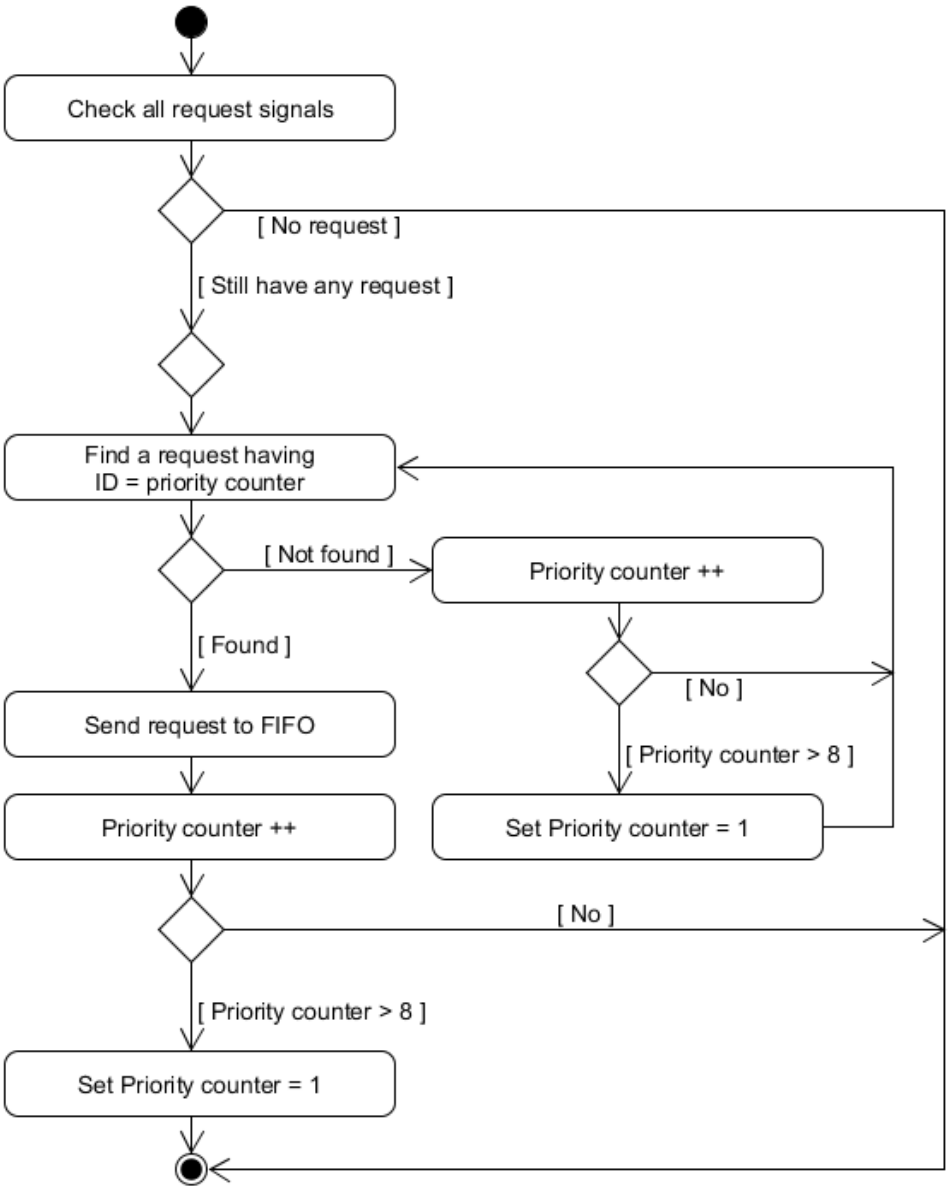


Figure 7.21: Flow of arbitration for transmission in PSI5 mode

Explanation:

(1) Figure above describes operations about arbitration for transmission in PSI5 mode. It occurs when there is a request (command data or ECU-to-Sensor data of channel 1-7) sent to Arbiter.

Note: There is no arbitration in UART mode, because there is only 1 request in UART mode.

(2) There is a priority counter. Its value is set 1 at constructor or reset PRESET or psis_rst_n.

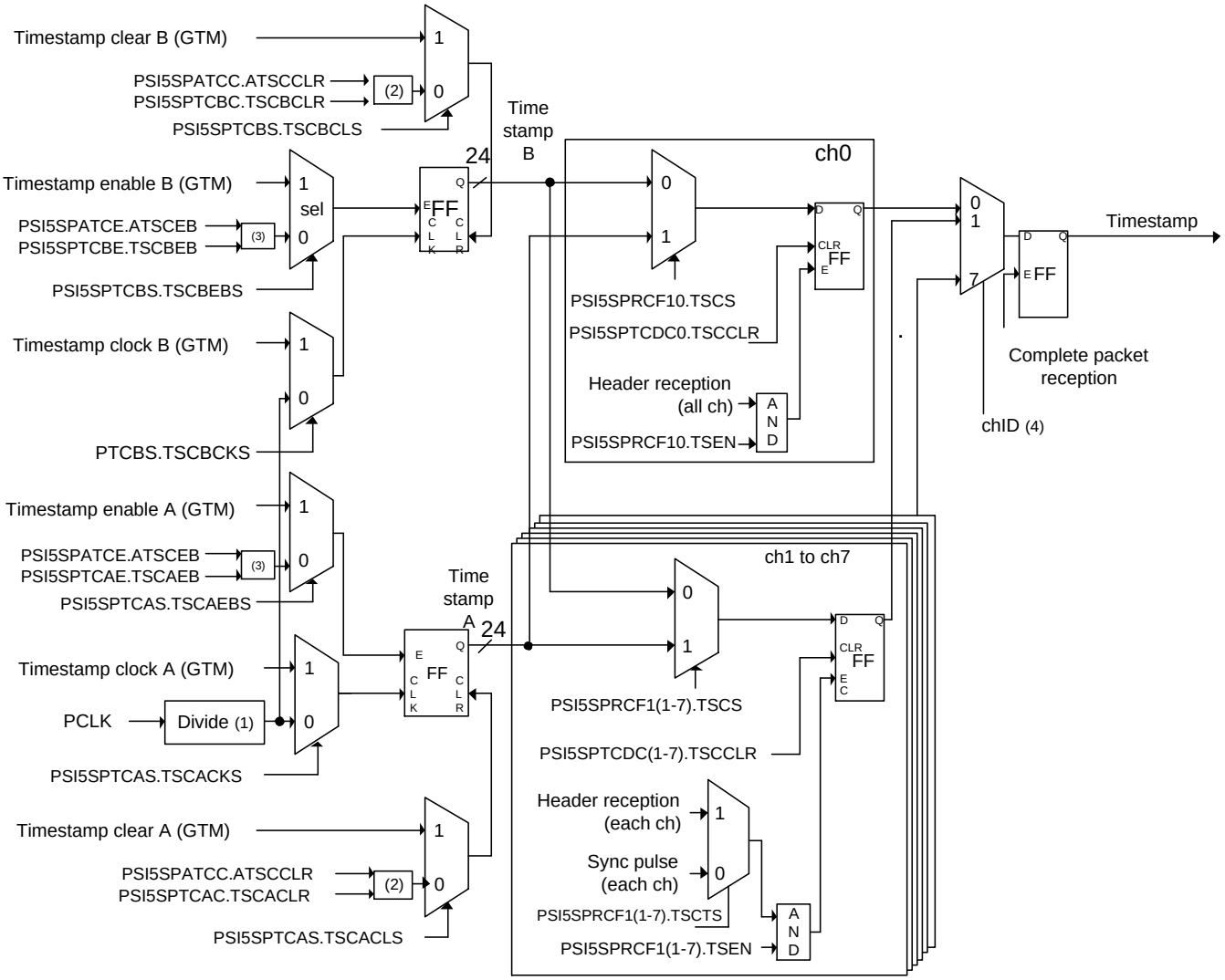
(3) Arbiter finds a request having ID equals to value of priority counter.

(3.1) If found, Arbiter sends such request above to Request-FIFO.

The priority counter is increased 1 for next arbitration. The priority counter is set to 1 if priority counter larger than 8.

(3.2) If not found, the priority counter is increased 1 and repeat finding a matched request.

7.18.Timestamp handler



(1) Generate the timestamp clock by the internal timing.
(2) Timestamp clear signals are specified individually or as a whole.
(3) Timestamp enable signals are specified individually or as a whole.
(4) Select Ch0 when an invalid frame is received; otherwise, select the received ChID.

Figure 7.22: Timestamp handler

Explanation:

(1) Figure above describes how timestamp is controlled by register and timestamp clock from GTM).

(2) There are 2 timestamps operating separately: timestamp A, and timestamp B.

(3) If timestamp clock from GTM is selected as counting clock, Timestamp counter is increased

1 at every rising edge of timestamp clock from GTM.

(4) If not, PCLK is divided into a pre-scaled clock based on setting in registers. Timestamp counter is increased 1 at every rising edge of such pre-scaled clock.

(5) In Synchronous mode:

(5.1) Based on setting in PSI5SPRCF1.TSCTS, channel 1-7 captures timestamp at Sync Pulse, or at time receiving frame header.

(5.2) Channel 0 does not operate in Synchronous mode.

(6) In Asynchronous mode:

(6.1) Channel 1-7 captures timestamp at time receiving frame header.

(6.2) Channel 0 captures timestamp at time receiving frame header, or at time WDT error occurring in any channel.

7.19.Flow of WDT counter in synchronous mode

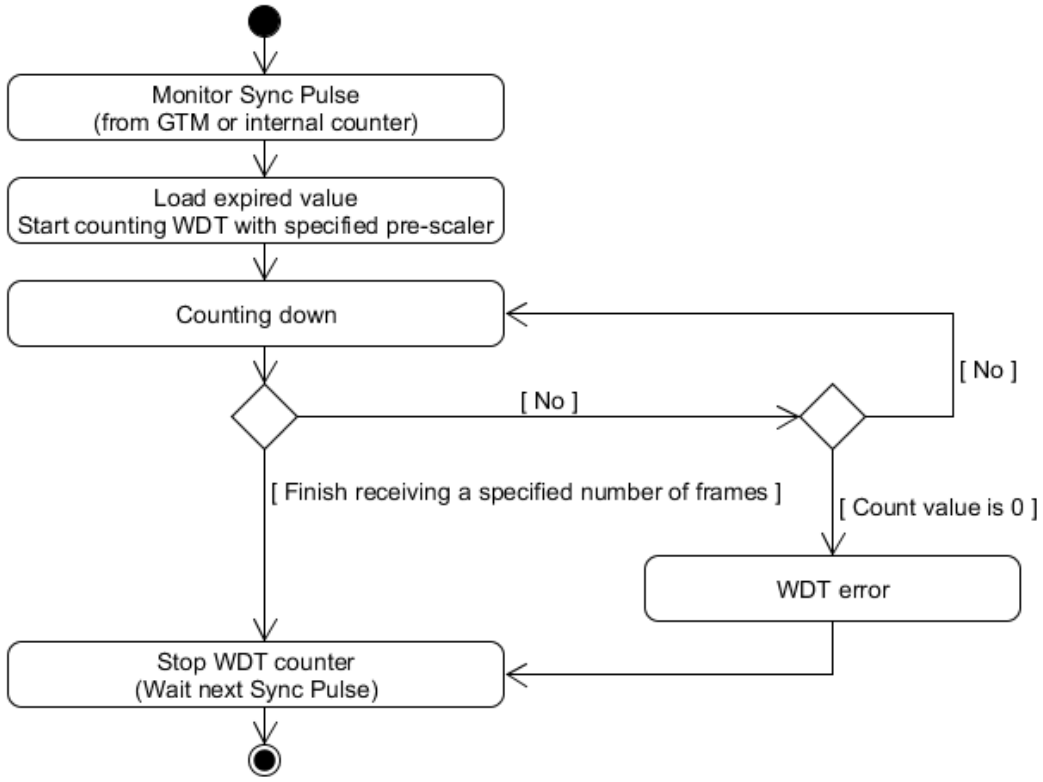


Figure 7.23: Flow of WDT counter in synchronous mode

Explanation:

(1) Figure above describes operation of WDT counter of each channel (0-7) when it is enabled (PSI5SPWDE.WDTEB = 1) in synchronous mode.

(2) At Sync Pulse (from GTM or internal counter), the WDT counter loads expired value (setting in PSI5SPWDEV register).

(3) The WDT starts counting down with specified pre-scaler (setting in PSI5SPWDP register).

At every count down value:

(3.1) If finish receiving a specified number of UART frames, the WDT is stopped. It will load expired value and start counting at next Sync Pulse.

(3.2) If the WDT count to 0, and not finish receiving a specified number of UART frames, WDT error occurs. The WDT is stopped. It will load expired value and start counting at next Sync Pulse.

(3.3) In other case, nothing is done. The WDT continues counting down.

7.20.Flow of WDT counter in asynchronous mode

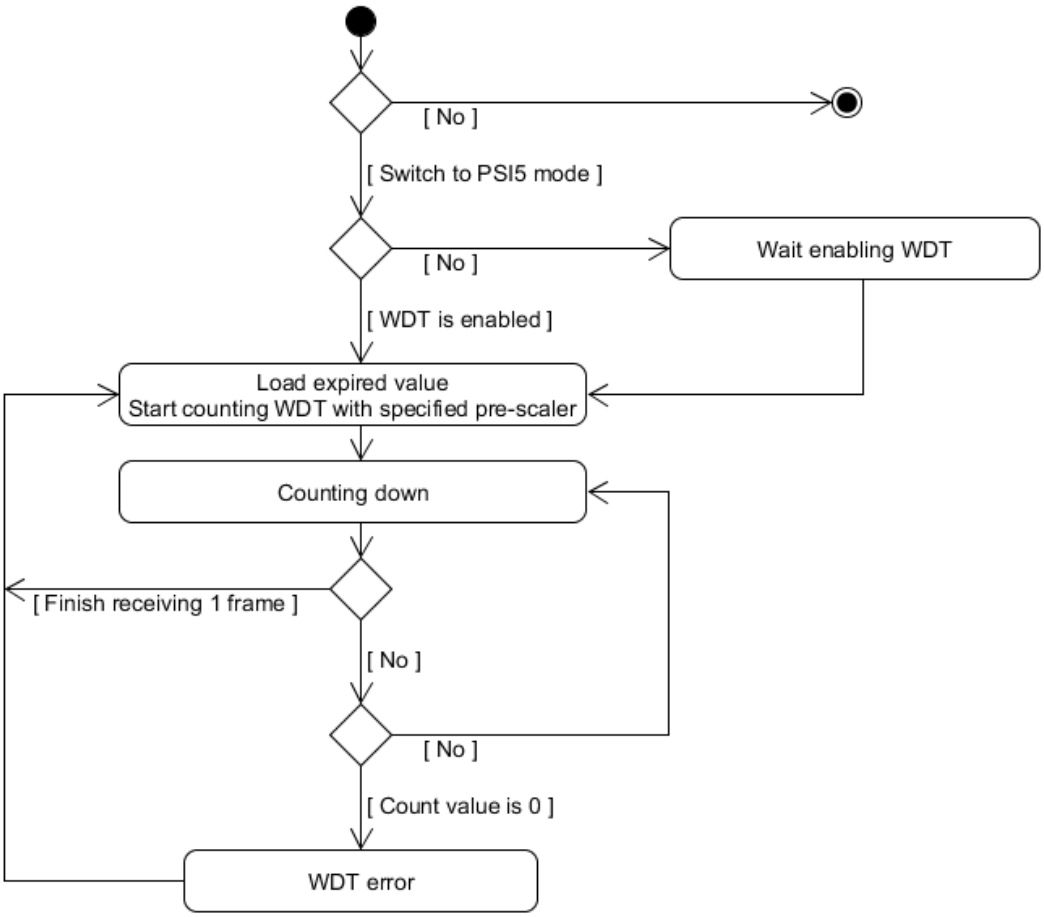


Figure 7.24: Flow of WDT counter in asynchronous mode

Explanation:

(1) Figure above describes operation of WDT counter of each channel (0-7) in asynchronous mode.

(2) Condition to WDT loads expired value and starts first running: PSIS011 model is switched to PSI5 mode and WDT is enabled.

(3) In PSI5 mode, when users write 1 to WDTEB bit, the WDT loads expired value and starts first running.

(4) The WDT reloads expired value to count again when finish receiving 1 frame.

(5) If the WDT counts down to 0 and current frame is not received yet, WDT error occurs and the WDT reloaded expired value to count again.

7.21.Flow of Python IF

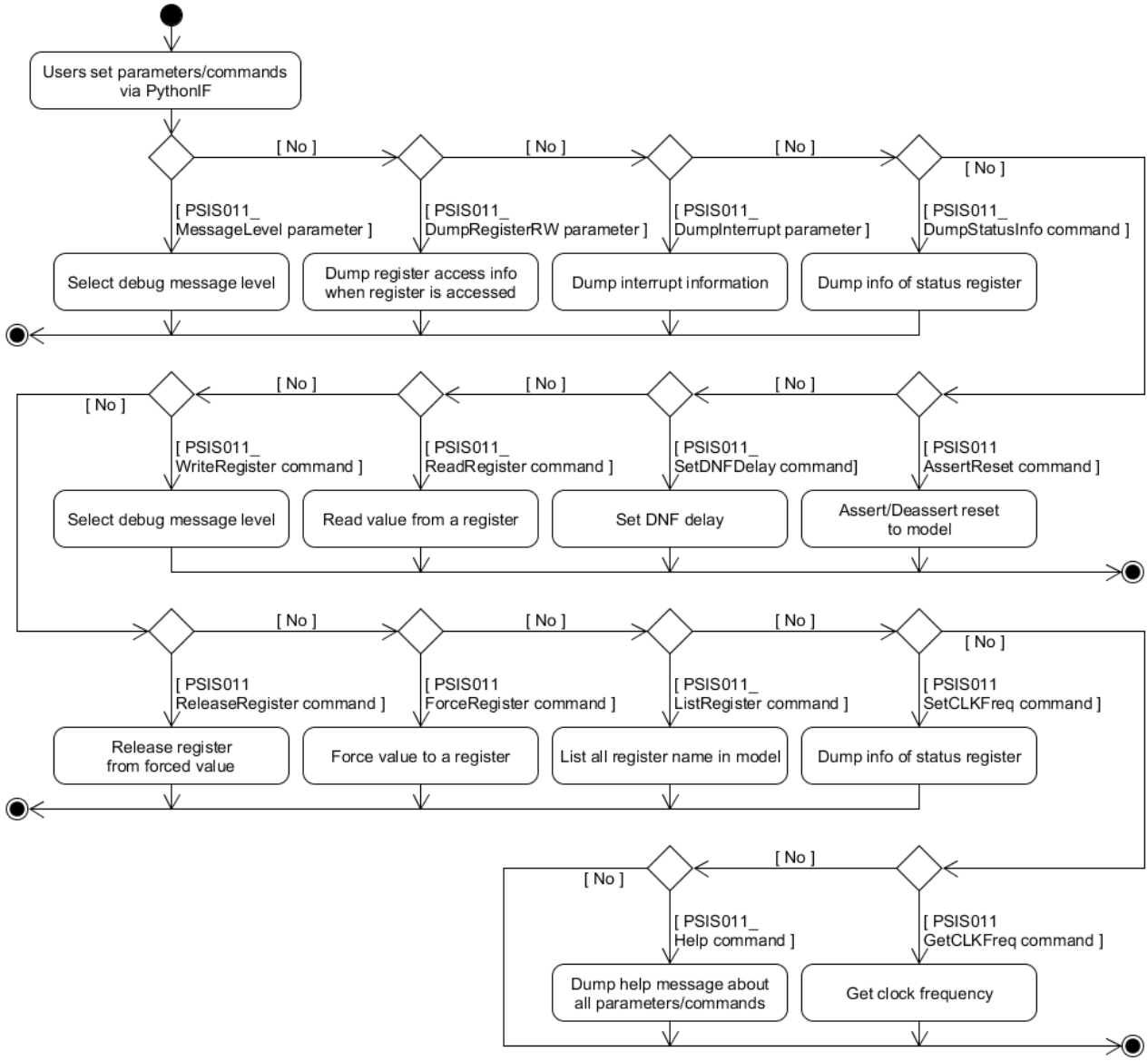


Figure 7.25: Flow of Python IF

Explanation:

(1) Figure above describes operation of Python IF when users set/call parameters/commands of PSIS011 model.

(2) Refer to chapter commands and parameters for more detail about functions of them.

7.22.Flow of loopback

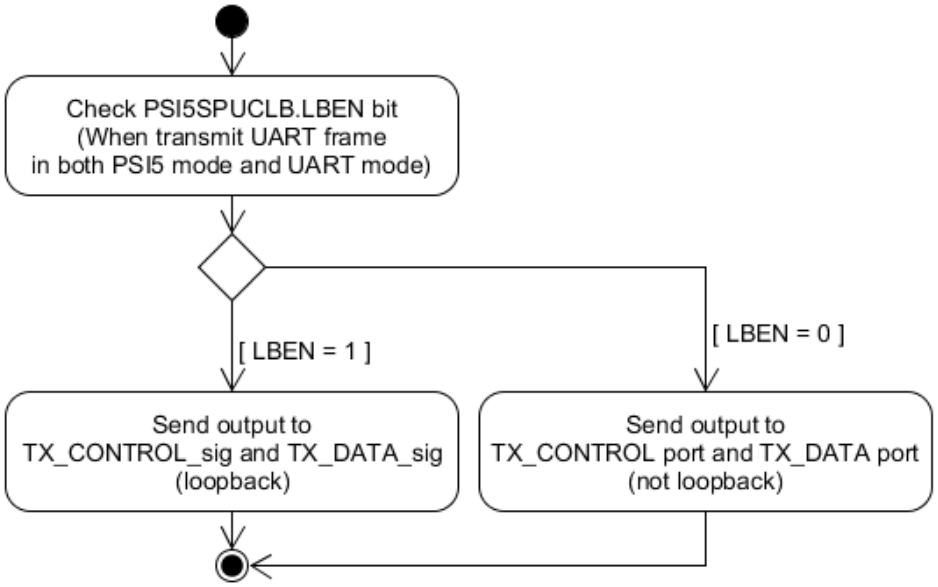


Figure 7.26: Flow of loopback

Explanation:

(1) Figure above describes about loopback operation in PSIS011 model.

(2) At every transmitting data in both PSI5 mode and UART mode, the PSI5SPUCLB.LBEN bit is checked to determine loopback or not.

(2.1) If LBEN bit is 1, model does loopback output data to input port. The output is sent to TX_CONTROL_sig and TX_DATA_sig. These signals are loopback to input RX_CONTROL and RX_DATA.

(2.2) If LBEN bit is 0, model sends output data to TX_CONTROL port and TX_DATA port. No loopback is done.

8.Function description

8.1.List of public/private function in Cwdt class

Table 8.1: List of public functions in Cwdt class

No.	Function name	Description
1	Cwdt (sc_module_name name);	Constructor of Cwdt class
2	~Cwdt ();	Destructor of Cwdt class
3	void EnableReset(const bool is_active);	Process when Cwdt is in reset state or idle state
4	void ConfigWDTClock(sc_dt::uint64 period, sc_time_unit time_unit);	Configure clock period, clock unit for WDT counter
5	void ConfigWDTCounter(unsigned int prescaler,	Configure prescaler, expired value for

	unsigned int expired);	WDT counter
6	void StartWDT();	Start WDT counter
7	void StopWDT();	Stop WDT counter

Table 8.2: List of private functions in Cwdt class

No.	Function name	Description
1	void WriteWDTErrorMethod();	Write wdt_error output port when the WDT count up to the expired value.
2	void NegateWDTErrorMethod();	Negate the wdt_error output port.
3	void Initialize();	Initialize internal variable when reset is activated.

8.2.List of public/private function in Csync_pulse class

Table 8.3: List of public functions in Csync_pulse class

No.	Function name	Description
1	Csync_pulse (sc_module_name name, PSIS011 *psis011);	Constructor of Csync_pulse class
2	~Csync_pulse ();	Destructor of Csync_pulse class
3	void EnableReset(const bool is_active);	Process when Csync_pulse is in reset state or idle state
4	void ConfigInnerSyncPulse(unsigned int select, unsigned int prescaler, unsigned int expired);	Configure kind of sync pulse (inner or from GTM, the prescaler, the expired value for inner counter.
5	void StartGenInnerSyncPulse();	Start generating inner sync pulse.
6	void StopGenInnerSyncPulse();	Stop generating innter sync pulse.
7	void ConfigSyncPulseClock(sc_dt::uint64 period, sc_time_unit time_unit);	Configure the clock period, clock unit for inner counter sync pulse.

Table 8.4: List of private functions in Csync_pulse class

No.	Function name	Description
1	void GenInnerSyncPulseThread();	A thread to generate the inner sync pulse after an expired time period.
2	void IssueSelectedSyncPulseMethod();	A method to issue the sync pulse output signal from the GTM or inner sync pulse according the sync pulse setting.
3	void Initialize();	Initialize internal variable when reset is activated.

8.3.List of public/private function in Ctimestamp class

Table 8.5: List of public functions in Ctimestamp class

No.	Function name	Description
1	Ctimestamp (sc_module_name name);	Constructor of Ctimestamp class
2	~Ctimestamp ();	Destructor of Ctimestamp class
3	unsigned int GetTimestamp();	Get current counter value (timestamp value).
4	void EnableReset(const bool is_active);	Process when Ctimestamp is in reset state or idle state
5	void NotifyClearTimestamp();	Clear counter value.
6	void ConfigTimestamp(bool clear_select_gtm, bool enable_select_gtm, bool clock_select_gtm);	Configure the setting related to select GTM or inner clock for couting timestamp.
7	void SetTSInnerEnable(bool ts_inner_enable);	Set enable for timestamp inner clock.

Table 8.6: List of private functions in Ctimestamp class

No.	Function name	Description
1	void ClearTimestampMethod();	Clear counter value according setting (select GTM input or inner input).
2	void MonitorGTMClockMethod();	Monitor the GTM clock for counting value if GTM input is selected (in configuration)
3	void MonitorInternalClockMethod();	Monitor the inner clock for counting value if inner input is selected (in configuration)
4	void Initialize();	Initialize internal variables when reset is activated.
5	bool CheckTimestampEnable();	Check timestamp is enabled to count or not.

8.4.List of public/private function in PSIS011 class

Table 8.7: List of public functions in PSIS011 class

No.	Function name	Description
1	PSIS011 (sc_module_name name, const unsigned int rLatency, const unsigned int wLatency);	Constructor of PSIS011 class
2	~PSIS011 ();	Destructor of PSIS011 class

3	double CalculateCLKPosEdge (const std::string clock_name, const bool add_period, const double time_point);	Calculate the rising edge of a clock.
4	bool CheckClockPeriod(const std::string clock_name);	Check a clock period is zero or not.
5	void SetMessageLevel (const std::string msg_lv);	Command function called from Python I/F. Set the message level (which will be enabled to dump) to PSIS011.
6	void DumpRegisterRW (const std::string is_enable);	Command function called from Python I/F. Enable/Disable dumping register access info.
7	void DumpInterrupt (const std::string is_enable);	Command function called from Python I/F. Enable/Disable dumping interrupt message.
8	void SetDNFDelay (const unsigned int delay);	Command function called from Python I/F. Set delay time for Noise Filter in PSIS011 model. Default delay time is 0. Unit is number of psis_clk clock.
9	void DumpStatusInfo ();	Dump the status information of the PSIS011.
10	void AssertReset (const std::string reset_name, const double start_time, const double period);	Command function called from Python I/F. Reset PSIS011 model according input arguments.
11	void SetCLKFreq (const std::string clock_name, const sc_dt::uint64 freq, const std::string unit);	Command function called from Python I/F. Set clock frequency to the PSIS011 model.
12	void GetCLKFreq (const std::string clock_name);	Command function called from Python I/F. Get clock frequency from PSIS011.
13	void ForceRegister (const std::string reg_name, const unsigned int chid, const unsigned int reg_value);	Command function called from Python I/F. Force a value to a register.
14	void ReleaseRegister (const std::string reg_name, const unsigned int chid);	Command function called from Python I/F. Release a register from a fixed value.

15	void WriteRegister (const std::string reg_name, const unsigned int chid, const unsigned int reg_value);	Command function called from Python I/F. Write a value to a register.
16	void ReadRegister (const std::string reg_name, const unsigned int chid);	Command function called from Python I/F. Read value from a register.
17	void ListRegister (void);	Command function called from Python I/F. List all supported register names.
18	void Help (const std::string type);	Command function called from Python I/F. Show the direction about using parameters/commands of the PSIS011.

Table 8.8: List of private functions in PSIS011 class

No.	Function name	Description
1	void MonitorPCLKMethod ();	Monitor PCLK input clock to update according clock period/frequency inside model.
2	void Monitorpsis_clkMethod ();	Monitor psis_clk input clock to update according clock period/frequency in model.
3	void Monitorpsis_mult_clkMethod ();	Monitor psis_mult_clk input clock to update according clock period/frequency in model.
4	void MonitorPRESETnMethod ();	Monitor PRESETn reset port to reset model.
5	void Monitorpsis_rst_nMethod ();	Monitor psis_rst_n reset port to reset model.
6	void Monitorpsis_mult_rst_nMethod ();	Monitor psis_mult_rst_n reset port to reset model.
7	void HandleResetHardMethod (const unsigned int reset_id);	Process reset function when reset port is active
8	void HandleResetCmdMethod (const unsigned int reset_id);	Process reset function when reset command is active
9	void CancelResetCmdMethod (const unsigned int reset_id);	Cancel reset function when reset command is active
10	void MonitorSyncPulseMethod(unsigned int chid);	Monitor selected sync pulse to issue DDSR request transmission, start WDT counter.

11	void MonitorWDTErrorMethod(unsigned int chid);	Monitor WDT error to stop reception, store imcomplete data in PSI5 mode
12	void MonitorDNFOutputMethod();	Monitor RX CONTROL, RX DATA after DNF delay.
13	void ArbitrateTxReqThread();	Arbitrate the transmission request from Tx Command data, DDSR request from channel 1-7.
14	void MoveToStopReceptionStateMethod();	Move to Stop Reception state after wait a period according 1 bit START, and 8/9 bit DATA.
15	void MoveToIdleReceptionStateMethod();	Move to Idle Reception state after wait a period for STOP bit.
16	void EndIdleReceptionStateMethod();	End the Idle Reception state after wait a maximum frame idle period (which specify in register).
17	void GenClockForTimestampThread();	Generate inner clock for couting timestamp.
18	void DNFDelayMethod();	Implement delay time for Noise Filter.
19	void ForwardDNFUARTRxMethod();	Forward received data/control to UART Rx block after a delay at DNF.
20	void WriteTxReqMethod(unsigned int reqid);	Write transmission request signal which is monitored by arbiter.
21	void NegateTxReqMethod(unsigned int reqid);	Negate the transmission request signal.
22	void TransPSI5Thread();	A thread to transmit the Tx command data, or ECU-to-sensor data (DDSR of channel 1-7)
23	void TransUARTThread();	A thread to transmit the UART data.
24	void WriteUARTOutputMethod();	Write TX_CONTROL and TX_DATA port.
25	void ClearTXSTSBitMethod();	Clear TXSTS bit, so that users can continue writing data to transmit Tx command data.
26	void ClearDDSRSTSBitMethod(unsigned int txreqid);	Clear DDSRSTS bit, so that users can continue writing data to transmit ECU-to-sensor data.
27	void WriteInterruptMethod(unsigned int chid);	Write value to int_psis_chn ports.
28	void NegateInterruptMethod(unsigned int chid);	Negate int_psis_chn ports.
29	void WriteDMARequestTXMethod(unsigned int	Write value to dma_psis_chn_tx ports.

	txreqid);	
30	void NegateDMARequestTXMethod(unsigned int txreqid);	Negate dma_psis_chn_tx ports.
31	void WriteDMARequestRXMethod(unsigned int chid);	Write value to dma_psis_chn_rx ports.
32	void NegateDMARequestRXMethod(unsigned int chid);	Negate dma_psis_chn_rx ports.
33	void ClearUTTBBFBitMethod();	Clear UTTBBF bit, so that users can continue writing data to transmit in UART mode.
34	void SetUTTFINBitMethod();	Set UTTFIN bit
35	void Writesw_rst_sigMethod();	Write value to software reset signal
36	void Negatesw_rst_sigMethod();	Negate software reset signal.
37	void InitializeOfPRESETn (void);	Initialize internal variable when PRESETn reset is activated.
38	void InitializeOfpsis_rst_n (void);	Initialize internal variable when psis_rst_n reset is activated.
39	void InitializeOfpsis_mult_rst_n (void);	Initialize internal variable when psis_mult_rst_n reset is activated.
40	void CancelEventsOfPRESETn();	Cancel events when PRESETn reset is activated.
41	void CancelEventsOfpsis_rst_n();	Cancel events when psis_rst_n reset is activated.
42	void CancelEventsOfpsis_mult_rst_n();	Cancel events when psis_mult_rst_n reset is activated.
43	void EnableReset(const std::string reset_name, const bool is_active);	Process when PSIS011 is in reset state or idle state of a reset port.
44	void EnablePRESETn(const bool is_active);	Process when PSIS011 is in reset state or not according to PRESETn port.
45	void Enablepsis_rst_n(const bool is_active);	Process when PSIS011 is in reset state or not according to psis_rst_n port.
46	void Enablepsis_mult_rst_n(const bool is_active);	Process when PSIS011 is in reset state or not according to psis_mult_rst_n port.
47	void ConvertClockFreq (sc_dt::uint64 &freq_out, std::string &unit_out, sc_dt::uint64 freq_in, std::string unit_in);	Convert clock frequency and frequency unit.
48	void GetTimeResolution (sc_dt::uint64 &resolution_value, sc_time_unit &resolution_unit);	Get time resolution.

49	void SeparateString (std::vector<std::string> &vtr, const std::string msg);	Separate some words from a string to store a vector.
50	std::string FindRegAreaFromName(const std::string reg_name);	Determine the register area (common, or channel) from the input register name.
51	void CreateValidFPKTvsFPAYLDTable();	Create a table describing valid setting for FPKT, FPAYLD.
52	void CreateStrobeStringMap();	Create a mapping between strobe value and its string name.
53	bool GenerateParity(unsigned int data, unsigned int length);	Generate parity from a data with a specified length
54	void ReceiveInIdleState();	Receive data in Idle state
55	void ReceiveInStartState();	Receive data in Start state
56	void ReceiveInStopState();	Receive data in Stop state
57	void RestorePSI5Frame(unsigned int data, bool parity_error, bool framing_error);	Restore PSI5 frame from a number of received UART frame
58	unsigned int PrepareUARTFrame(unsigned int data, unsigned int parity_option, unsigned int &numbit);	Prepare UART frame to transmit
59	void PrepareDDSRData(unsigned int chid, sc_dt::uint64 &ddsr, unsigned int &length);	Prepare DDSR from the setting in register.
60	unsigned int PrepareTXCONTROL(eStrobe strobe, unsigned int numbit, unsigned int tc);	Prepare value to write to TX_CONTROL port.
61	void UpdateTXFIFOStatus();	Update TX FIFO status (empty, or full, or normal) every put/pop element into/out of fifo.
62	bool GetResetStatus(const std::string reset_name);	Get reset status in PSIS011. This is overwritten function from PSIS011_AgentController.
63	void SoftwareReset();	Order PSIS011 to issue software reset. This is overwritten function from PSIS011_AgentController.
64	sc_dt::uint64 GetPCLKFreq();	Get PCLK frequency. This is overwritten function from PSIS011_AgentController.
65	void ClearTimestamp(std::string timestamp_name);	Order PSIS011 to clear timestamp when users set register to clear it. This is overwritten function from PSIS011_AgentController.

66	void ConfigTimestamp(std::string timestamp_name, bool clear_select_gtm, bool enable_select_gtm, bool clock_select_gtm);	Order PSIS011 to configure arguments for couting timestamp when users write value to register. This is overwritten function from PSIS011_AgentController.
67	void SetTSInnerEnable(std::string timestamp_name, bool ts_inner_enable);	Order PSIS011 to set timestamp enable when users write value to according register. This is overwritten function from PSIS011_AgentController.
68	void SetTSInnerCycle(unsigned int inner_cycle);	Order PSIS011 to set inner cycle for couting timestamp when users write to according register. This is overwritten function from PSIS011_AgentController.
69	void InformChangeMode(unsigned int open, unsigned int opmd);	Inform PSIS011 to change mode when users write to according register. This is overwritten function from PSIS011_AgentController.
70	void NotifySendReqTxToArbiter(unsigned int reqid);	Notify PSIS011 to write TX request to arbiter when users write to according register. This is overwritten function from PSIS011_AgentController.
71	void NotifyTransUART();	Notify PSIS011 to transmit UART frame when users write to according register. This is overwritten function from PSIS011_AgentController.
72	void AssertInterrupt(unsigned int intrid);	Order PSIS011 to assert an interrupt . This is overwritten function from PSIS011_AgentController.
73	void AssertDMARequestRX(unsigned int dmarxid);	Order PSIS011 to assert a DMA request RX. This is overwritten function from PSIS011_AgentController.
74	void AssertDMARequestTX(unsigned int dmatxid);	Order PSIS011 to assert a DMA request TX. This is overwritten function from PSIS011_AgentController.
75	void ChangeWDTEnableChannel(unsigned int chid, unsigned int enable);	Inform PSIS011 to enable/disable WDT of a channel when users write to accroding register. This is overwritten function from

		PSIS011_AgentController.
76	void InformChangeSCKENBit(unsigned int value);	Inform PSIS011 to enable/disable the UART output clock when users write to accroding register. This is overwritten function from PSIS011_AgentController.
77	void StopTransDDSR(unsigned int reqid);	Inform PSIS011 to stop transmit ECU-to-sensor data when users write to according register. This is overwritten function from PSIS011_AgentController.
78	void ConfigInnerSyncPulse(unsigned int chid, unsigned int select, unsigned int prescaler, unsigned int expired);	Configure argument for inner clock sync pulse.
79	unsigned int GetTimestamp(unsigned int chid);	Get timestamp of a specified channel.
80	void GenerateMBData(unsigned int &allocated_chid, unsigned int &allocated_frmid, unsigned int &status_reg_val, unsigned int &data_reg_val, unsigned int &timestamp_reg_val);	Generate the mailbox data (status, timestamp, data)
81	void GenerateMBDataForWDTError(unsigned int &allocated_chid, unsigned int &allocated_frmid, unsigned int &status_reg_val, unsigned int &data_reg_val, unsigned int &timestamp_reg_val);	Generate the mailbox data (status, timestamp, data) in case WDT error.
82	bool CheckValidFPKTvsFPAYLD(unsigned int fpayld, unsigned int rfcps, unsigned int fpkt);	Check validity between FPKT and FPAYLD bit.
83	unsigned int GenerateCRC(unsigned int data, unsigned int length);	Generate CRC value for a data with specified length.
84	unsigned int GenerateCRCExtra(sc_dt::uint64 data, unsigned int length);	Generate XCRC value for a data with specified length.
85	sc_dt::uint64 Reverse(sc_dt::uint64 input, unsigned int length);	Reverse data from LSB first to MSB first, or from MSB first to LSB first.

8.5.List of public/private function in PSIS011_Func class

Table 8.9: List of public functions in PSIS011_Func class

No.	Function name	Description
1	PSIS011_Func(std::string name, PSIS011_AgentController* PSIS011AgentController);	Constructor of PSIS011_Func class

2	~PSIS011_Func();	Destructor of PSIS011_Func class
3	void EnableReset(const bool is_active);	Process when PSIS011_Func is in reset state or idle state.
4	void RegisterHandler(const std::string reg_area, const unsigned int chid, const std::vector<std::string> &args);	Forward all command/parameter related to register interface.
5	void SoftwareReset();	Forward software reset to PSIS011.
6	void ClearAllMailBoxData();	Call all channels to clear all mail box data.
7	void StopGenOutputClock();	Order PSIS011 to stop generate output clock.
8	sc_dt::uint64 GetPCLKFreq();	Order PSIS011 to get PCLK frequency.
9	unsigned int GetACSTSBit();	Get ACSTS bit in common register.
10	unsigned int GetMSTSBit();	Get MSTS bit in common register
11	void ClearTimestamp(std::string timestamp_name);	Order PSIS011 to clear timestamp value.
12	void ConfigTimestamp(std::string timestamp_name, bool clear_select_gtm, bool enable_select_gtm, bool clock_select_gtm);	Order PSIS011 to configure argument related to timestamp.
13	void SetTSInnerCycle(unsigned int inner_cycle);	Order PSIS011 to set cycle for timestamp counting.
14	void InformChangeMode(unsigned int open, unsigned int opmd);	Order PSIS011 to change mode.
15	void GetConfigInnerSyncPulse(unsigned int chid, unsigned int &select, unsigned int &prescaler, unsigned int &expired);	Get configuration related to Sync Pulse setting in a specified channel.
16	void GetConfigWDT(unsigned int chid, unsigned int &syncmode, unsigned int &prescaler, unsigned int &expired, bool &wdt_enable);	Get configuration related to WDT in a specified channel register.
17	void GetConfigTimestamp(unsigned int chid, unsigned int &ch_en, unsigned int &ts_sel, unsigned int &ts_en, unsigned int &ts_trg_sel);	Get configuration related to timestamp in a specified channel register.
18	unsigned int GetPSI5SPUPTSReg();	Get PSI5SPUPTS register value in common register.
19	void GetClockDivider(unsigned int &rx_divider, unsigned int &tx_divider);	Get setting related to clock divider from common register.
20	unsigned int GetMaxIdle();	Get maximum framing idle value from common register.
21	void CaptureTimestamp(unsigned int chid, unsigned int	Capture timestamp into according

	timestamp);	register in a specified channel.
22	void StoreMBData(unsigned int chid, unsigned int frame, unsigned int status_reg_val, unsigned int data_reg_val, unsigned int timestamp_reg_val);	Store mail box data in a specified channel.
23	void StoreUARTData(unsigned int data, bool framing_error, bool parity_error);	Store UART data to according register.
24	unsigned int GetRFCPSBit(unsigned int chid);	Get RFCPS bit of a specified channel
25	void GetConfigPSI5RxFrame(unsigned int chid, unsigned int fid, unsigned int& fpkt, unsigned int& fpayload, unsigned int& rfcps);	Get configuration of a PSI5 reception frame of a specified channel.
26	bool CheckMBDataWasRead(unsigned int chid, unsigned int frmid);	Check a mailbox data was read before or not.
27	unsigned int GetCapturedTimestamp(unsigned int chid);	Get captured timestamp in a register from common register part.
28	void SetTSInnerEnable(std::string timestamp_name, bool ts_inner_enable);	Order PSIS011 to set enable/disable timestamp.
29	void GetDDSRInfo(unsigned int chid, unsigned int &frame_type, unsigned int &address, unsigned int &data);	Get information related to DDSR.
30	void GetConfigCPUTxCom(unsigned int &numfrm, sc_dt::uint64 &tx_cpu_com_data);	Get configuration related to CPU Tx command data.
31	unsigned int GetCommandData(unsigned int chid);	Get command data which written in register.
32	void NotifySendReqTxToArbiter(unsigned int reqid);	Order PSIS011 to write a transmission request to arbiter.
33	void NotifyTransUART();	Order PSIS011 to transmit UART frame.
34	unsigned int GetUTTDTBit();	Get UTTDT bit in common register.
35	void SetTXSTSBit(unsigned int value);	Set TXSTS bit in common register.
36	void ClearDDSRSTSBit(unsigned int chid);	Clear DDSRSTS bit in a specified channel.
37	void AssertInterrupt(unsigned int intrid);	Order PSIS011 to assert interrupt of a specified channel.
38	void AssertDMARequestRX(unsigned int dmarxid);	Order PSIS011 to assert DMA request RX for a specified channel.
39	void AssertDMARequestTX(unsigned int dmatxid);	Order PSIS011 to assert DMA request TX for a specified channel.
40	void ChangeWDTEnableChannel(unsigned int chid,	Order PSIS011 to enable/disable WDT of

	unsigned int enable);	a specified channel.
41	void InformChangeSCKENBit(unsigned int value);	Inform PSIS011 to enable/disable output UART clock.
42	unsigned int GetNFSETBit();	Get NFSET bit in common register.
43	void SetPSI5SPCISReg(unsigned int chid, std::string status);	Set PSI5SPCIS register in a specified channel.
44	void UpdatePSI5SUCTMReg(std::string bit_name, unsigned int value);	Update PSI5SUCTM register in common register.
45	void UpdateUTTFINBit(unsigned int value);	Update UTTFIN bit in common register.
46	void DumpStatusInfo();	Dump status info related to common register, and all channel register.
47	void SetPSI5SPTFISReg(unsigned int value);	Set PSI5SPTFIS register.
48	void SetPSI5SPRESReg(unsigned int chid, unsigned int frmid);	Set PSI5SPRES register.
49	void StopTransDDSR(unsigned int txreqid);	Order PSIS011 to stop transmit ECU-to-sensor data of a specified channel.
50	void SetSWSTSBit(unsigned int value);	Set SWSTS bit in common register.
51	bool GetLoopbackEnable();	Get loopback enable attribute from common register.
52	void read(unsigned int offsetAddress, TlmBasicPayload& trans, BusTime_t* t, bool debug);	Virtual function of Slave Interface to read common register or channels registers.
53	void write(unsigned int offsetAddress, TlmBasicPayload& trans, BusTime_t* t, bool debug);	Virtual function of Slave Interface to write common register or channels registers.

Table 8.10: List of private functions in PSIS011_Func class

No.	Function name	Description
1	std::string FindRegAreaFromAddr(const unsigned int address);	Determine register area (common, channel, or mailbox) from an address.

8.6.List of public/private function in PSIS011_Ch_Reg class

Table 8.11: List of public functions in PSIS011_Ch_Reg class

No.	Function name	Description
1	PSIS011_Ch_Reg(std::string name, unsigned int chid, PSIS011_Func* PSIS011Func);	Constructor of PSIS011_Ch_Reg class.

2	~PSIS011_Ch_Reg();	Destructor of PSIS011_Ch_Reg class.
3	void EnableReset (const bool is_active);	Process when PSIS011_Ch_Reg is in reset state or in idle state.
4	void Initialize();	Initialize internal variable when reset is activated.
5	void RegisterHandler (const std::string reg_area, const std::vector<std::string> &args);	Forward all command/parameter related to register interface.
6	void GetConfigInnerSyncPulse (unsigned int &select, unsigned int &prescaler, unsigned int &expired);	Get configuration related to Sync Pulse.
7	void GetConfigWDT (unsigned int &syncmode, unsigned int &prescaler, unsigned int &expired, bool &wdt_enable);	Get configuration related to WDT.
8	void GetConfigPSI5RxFrame (unsigned int fid, unsigned int& fpkt, unsigned int& fpayload, unsigned int& rfcps);	Get configuration of a PSI5 reception frame.
9	unsigned int GetRFCPSBit();	Get RFCPS bit.
10	unsigned int GetMaxIdle();	Get maximum framing idle value applying for reception PSI5 mode.
11	void GetConfigTimestamp(unsigned int &ch_en, unsigned int &ts_sel, unsigned int &ts_en, unsigned int &ts_trg_sel);	Get configuration related to timestamp.
12	void CaptureTimestamp(unsigned int timestamp);	Capture timestamp into according register.
13	unsigned int GetCapturedTimestamp();	Get captured timestamp.
14	bool CheckMBDataWasRead(unsigned int frmid);	Check a mailbox data was read before or not.
15	void StoreMBData(unsigned int frame, unsigned int status_reg_val, unsigned int data_reg_val, unsigned int timestamp_reg_val);	Store mail box data.
16	unsigned int GetCommandData();	Get command data which written in register.
17	void GetDDSRInfo(unsigned int &frame_type, unsigned int &address, unsigned int &data);	Get information related to DDSR.
18	void ClearDDSRSTSBit();	Clear DDSRSTS bit.
19	void ClearAllMailBoxData();	Call all channels to clear all mail box data.
20	void SetPSI5SPCISReg(std::string status);	Set PSI5SPCIS register.
21	void DumpStatusInfo();	Dump status info related channel

		register.
22	void ClearAllStatus();	Clear all status in channel
23	void SetPSI5SPRESReg(unsigned int frmid);	Set PSI5SPRES register.
24	void read (std::string area, unsigned int address, unsigned char *p_data, unsigned int size, bool debug);	Read channel register or mailbox according to the input address.
25	void write (std::string area, unsigned int address, unsigned char *p_data, unsigned int size, bool debug);	Write channel register or mailbox according to the input address

Table 8.12: List of private functions in PSIS011_Ch_Reg class

No.	Function name	Description
1	void cb_PSI5SPRCF1n_CHEN(Cpsis011_ch_regif::RegCBstr str);	Write-callback function of PSI5SPRCF1n register
2	void cb_PSI5SPRCF2n_F1PAYLD(Cpsis011_ch_regif::RegCBstr str);	Write-callback function of PSI5SPRCF2n register
3	void cb_PSI5SPWDEn_WDTEB(Cpsis011_ch_regif::RegCBstr str);	Write-callback function of PSI5SPWDEn register
4	void cb_PSI5SPWDPn_WDTPRS(Cpsis011_ch_regif::RegCBstr str);	Write-callback function of PSI5SPWDPn register
5	void cb_PSI5SPWDEVn_WDTEX(Cpsis011_ch_regif::RegCBstr str);	Write-callback function of PSI5SPWDEVn register
6	void cb_PSI5SPTCDn_CHID(Cpsis011_ch_regif::RegCBstr str);	Write-callback function of PSI5SPTCDn register
7	void cb_PSI5SPCIEn_IEBCRC(Cpsis011_ch_regif::RegCBstr str);	Write-callback function of PSI5SPCIEn register
8	void cb_PSI5SPDREn_DRQERFN(Cpsis011_ch_regif::RegCBstr str);	Write-callback function of PSI5SPDREn register
9	void cb_PSI5SPSTPn_STPRS(Cpsis011_ch_regif::RegCBstr str);	Write-callback function of PSI5SPSTPn register
10	void cb_PSI5SPSTEVn_STEX(Cpsis011_ch_regif::RegCBst	Write-callback function of PSI5SPSTEVn register

	r str);	
11	void cb_PSI5SPSTSn_STSEL(Cpsis011_ch_regif::RegCBstr str);	Write-callback function of PSI5SPSTSn register
12	void cb_PSI5SPRESCn_RERRCLF1(Cpsis011_ch_regif::RegCBstr str);	Write-callback function of PSI5SPRESCn register
13	void cb_PSI5SPTCDCn_TSCCLR(Cpsis011_ch_regif::RegCBstr str);	Write-callback function of PSI5SPTCDCn register
14	void cb_PSI5SPDDTPn_DDSRTYPE(Cpsis011_ch_regif::RegCBstr str);	Write-callback function of PSI5SPDDTPn register
15	void cb_PSI5SPDDDn_DDSRADR(Cpsis011_ch_regif::RegCBstr str);	Write-callback function of PSI5SPDDDn register
16	void cb_PSI5SPDDSPn_DDSRSTP(Cpsis011_ch_regif::RegCBstr str);	Write-callback function of PSI5SPDDSPn register
17	void cb_PSI5SPCISCn_ISTCCRC(Cpsis011_ch_regif::RegCBstr str);	Write-callback function of PSI5SPCISCn register

8.7.List of public/private function in PSIS011_Cmn_Reg class

Table 8.13: List of public functions in PSIS011_Cmn_Reg class

No.	Function name	Description
1	PSIS011_Cmn_Reg(std::string name, PSIS011_Func* PSIS011Func);	Constructor of PSIS011_Cmn_Reg class.
2	~PSIS011_Cmn_Reg();	Destructor of PSIS011_Cmn_Reg class.
3	void EnableReset(const bool is_active);	Process when PSIS011_Cmn_Reg is in reset state or in idle state.
4	void Initialize();	Initialize internal variable when reset is activated.
5	void RegisterHandler(const std::vector<std::string> &args);	Forward all command/parameter related to register interface.
6	void ClearPUCLBStepDone();	Clear recorded steps which used to enable loopback function.

7	unsigned int GetPUCLBStepDone();	Get recorded steps which is proceeded to enable loopback function
8	unsigned int GetACSTSBit();	Get ACSTS bit
9	unsigned int GetMSTSBit();	Get MSTS bit
10	void StoreUARTData(unsigned int data, bool framing_error, bool parity_error);	Store UART data to register
11	void GetClockDivider(unsigned int &rx_divider, unsigned int &tx_divider);	Get clock divider for transmit clock, receive clock.
12	unsigned int GetPSI5SPUPTSReg();	Get PSI5SPUPTS register value
13	void GetConfigCPUTxCom(unsigned int &numfrm, sc_dt::uint64 &tx_cpu_com_data);	Get configuration related to CPU Tx command data.
14	unsigned int GetUTTDTBit();	Get UTTDT bit
15	void SetTXSTSBit(unsigned int value);	Set value to TXSTS bit.
16	unsigned int GetNFSETBit();	Get NFSET bit
17	void UpdateUTTFINBit(unsigned int value);	Update value to UTTFIN bit.
18	void UpdatePSI5SUCTMReg(std::string bit_name, unsigned int value);	Update PSI5SUCTM register
19	void DumpStatusInfo();	Dump value of all status register in common register area.
20	void ClearAllStatus();	Clear all status in common register area.
21	void SetPSI5SPTFISReg(unsigned int value);	Set value to PSI5SPTFIS register
22	void SetSWSTSBit(unsigned int value);	Set value to SWSTS bit.
23	bool GetLoopbackEnable();	Check loopback function is enabled or not.
24	void read (unsigned int address, unsigned char *p_data, unsigned int size, bool debug);	Read register in common register area.
25	void write (unsigned int address, unsigned char *p_data, unsigned int size, bool debug);	Write register in common register area.

Table 8.14: List of private functions in PSIS011_Cmn_Reg class

No.	Function name	Description
1	void UpdatePUOSReg();	Update status in PSI5SPUOS register
2	void ConfigTimestamp(std::string timestamp_name);	Configure timestamp A/B when related registers are changed.
3	void cb_PSI5SPUOEB_OPEN(RegCBstr str);	Write-callback function of PSI5SPUOEB register

4	void cb_PSI5SPUOMD_OPMD(RegCBstr str);	Write-callback function of PSI5SPUOMD register
5	void cb_PSI5SPUNFST_NFSET(RegCBstr str);	Write-callback function of PSI5SPUNFST register
6	void cb_PSI5SPUSWR_SWRST(RegCBstr str);	Write-callback function of PSI5SPUSWR register
7	void cb_PSI5SPRMBC_MBCLR(RegCBstr str);	Write-callback function of PSI5SPRMBC register
8	void cb_PSI5SPUCLB_LBEN(RegCBstr str);	Write-callback function of PSI5SPUCLB register
9	void cb_PSI5SPUPTS_UTPRTY(RegCBstr str);	Write-callback function of PSI5SPUPTS.UTPRTY bit
10	void cb_PSI5SPUPTS_URPRTY(RegCBstr str);	Write-callback function of PSI5SPUPTS.URPRTY bit
11	void cb_PSI5SPUBCE_SCKEN(RegCBstr str);	Write-callback function of PSI5SPUBCE register
12	void cb_PSI5SPUBPR_RXOSMP(RegCBstr str);	Write-callback function of PSI5SPUBPR.RXOSMP bit
13	void cb_PSI5SPUBPR_SCKPRS(RegCBstr str);	Write-callback function of PSI5SPUBPR.SCKPRS bit
14	void cb_PSI5SPTPS_TSPRSL(RegCBstr str);	Write-callback function of PSI5SPTPS register
15	void cb_PSI5SPTCAS_TSCACLS(RegCBstr str);	Write-callback function of PSI5SPTCAS.TSCACLS bit
16	void cb_PSI5SPTCAS_TSCAEBS(RegCBstr str);	Write-callback function of PSI5SPTCAS.TSCAEBS bit
17	void cb_PSI5SPTCAS_TSCACKS(RegCBstr str);	Write-callback function of PSI5SPTCAS.TSCACKS bit
18	void cb_PSI5SPTCBS_TSCBCLS(RegCBstr str);	Write-callback function of PSI5SPTCBS.TSCBCLS bit
19	void cb_PSI5SPTCBS_TSCBEBS(RegCBstr str);	Write-callback function of PSI5SPTCBS.TSCBEBS bit
20	void cb_PSI5SPTCBS_TSCBCKS(RegCBstr str);	Write-callback function of PSI5SPTCBS.TSCBCKS bit
21	void cb_PSI5SPTCAE_TSCAEB(RegCBstr str);	Write-callback function of PSI5SPTCAE register
22	void cb_PSI5SPTCAC_TSCACLR(RegCBstr str);	Write-callback function of PSI5SPTCAC register

23	void cb_PSI5SPTCBE_TSCBEB(RegCBstr str);	Write-callback function of PSI5SPTCBE register
24	void cb_PSI5SPTCBC_TSCBCLR(RegCBstr str);	Write-callback function of PSI5SPTCBC register
25	void cb_PSI5SPATCE_ATSCEB(RegCBstr str);	Write-callback function of PSI5SPATCE register
26	void cb_PSI5SPATCC_ATSCCLR(RegCBstr str);	Write-callback function of PSI5SPATCC register
27	void cb_PSI5SUCRIE_IERPE(RegCBstr str);	Write-callback function of PSI5SUCRIE register
28	void cb_PSI5SUCTIE_IETOWE(RegCBstr str);	Write-callback function of PSI5SUCTIE register
29	void cb_PSI5SUCDRE_DRQEURFN(RegCBstr str);	Write-callback function of PSI5SUCDRE register
30	void cb_PSI5SUCRD_UTRDT(RegCBstr str);	Write-callback function of PSI5SUCRD register
31	void cb_PSI5SUCRSC_UTRPECL(RegCBstr str);	Write-callback function of PSI5SUCRSC register
32	void cb_PSI5SPTFST_TXST(RegCBstr str);	Write-callback function of PSI5SPTFST register
33	void cb_PSI5SPTFNM_TXNUM(RegCBstr str);	Write-callback function of PSI5SPTFNM register
34	void cb_PSI5SPTFD1_TDT4(RegCBstr str);	Write-callback function of PSI5SPTFD1.TDT4 bit.
35	void cb_PSI5SPTFD1_TDT3(RegCBstr str);	Write-callback function of PSI5SPTFD1.TDT3 bit.
36	void cb_PSI5SPTFD1_TDT2(RegCBstr str);	Write-callback function of PSI5SPTFD1.TDT2 bit.
37	void cb_PSI5SPTFD1_TDT1(RegCBstr str);	Write-callback function of PSI5SPTFD1.TDT1 bit.
38	void cb_PSI5SPTFD2_TDT8(RegCBstr str);	Write-callback function of PSI5SPTFD2.TDT8 bit.
39	void cb_PSI5SPTFD2_TDT7(RegCBstr str);	Write-callback function of PSI5SPTFD2.TDT7 bit.
40	void cb_PSI5SPTFD2_TDT6(RegCBstr str);	Write-callback function of PSI5SPTFD2.TDT6 bit.
41	void cb_PSI5SPTFD2_TDT5(RegCBstr str);	Write-callback function of PSI5SPTFD2.TDT5 bit.

42	void cb_PSI5SUCTD_UTTDT(RegCBstr str);	Write-callback function of PSI5SUCTD register
43	void cb_PSI5SUCTSC_UTTOWECL(RegCBstr str);	Write-callback function of PSI5SUCTSC register

9.Limitation

(1) Table below list all limitation of this model (comparing with HW).

Table 9.1: Limitation of model

No.	HWM	Model
1	Noise Filter to eliminate noise in input ports. The input to UART reception block is delayed a period.	Not support Noise Filter because input port is packet, not wire. Delay period at Noise Filter is set via Python IF command PSIS011_SetDNFDelay(). Refer "Command and Parameters" chapter for detail.
2	Test Function.	Not support test function.
3	When WDT error occurs in a channel, the incomplete data can be some bits in frame.	Just receive whole UART frame only, not store some bit of incomplete frame. So, if an UART frame is not finished receiving at time of WDT error, no bit in this frame is stored.

Revision History				
Rev.	**Modified Contents**	**Approved by RVC**	**Checked**	**Created**
1.0	Create new.	Binh Nguyen 10/17/2017	Hue Pham Chuong Le 10/17/2017	Chan Le 10/13/2017
1.1	- Update Table 6.8 about message. - Update Figure 7.2, Figure 7.4 about timing chart of transmission. - Add Figure 7.6, Figure 7.8 about timing chart of reception. - Update Figure 7.17 and its explanation about SW reset. - Full fill chapter 8. - Add limitation No.3 in chapter 9.	Binh Nguyen 12/19/2017	Chuong Le 12/19/2017	Chan Le 12/19/2017
1.2	- Add message No.67 to Table 6.8.	Chan Le 02/24/2018	Chuong Le 02/24/2018	Chan Le 02/23/2018