COMM_CTRL IP SPEC

Table of Contents

Introduction

The COMM_CTRL module is to analysis received data, and generate tx_data for coping to next device or response back.

Feature

Key features of the COMM_CTRL module are:

- propagate rx_data to next device.
- reset when CLK_32M_OK low
- support writing register bit
- support reading register bit
- CRC check
- support for both bridge and AFE application

Register Definition

Register Map

Table 1 COMM_CTRL1 Register Map

Name	Add	D7	D6	D5	D4	D3	D2	D1	D0	Default
COMM_CONF2	0x0003	COMN_TX_DIS	COMS_TX_DIS	STACK_RESPONSE<5:0>						00
CTRL1	0x2002	SRSTB	DIR_SEL		WAKE_TONE_GEN	STA_TONE_GEN	SD_TONE_GEN	TO_SD	TO_SLEEP	80
CTRL2	0x2003						CMP_BIST_GO	ADD_W_EN	SPI_DIR	00

Functional Details

Block Diagram

The following diagram shows the COMM_CTRL architecture and internal modules and connections.

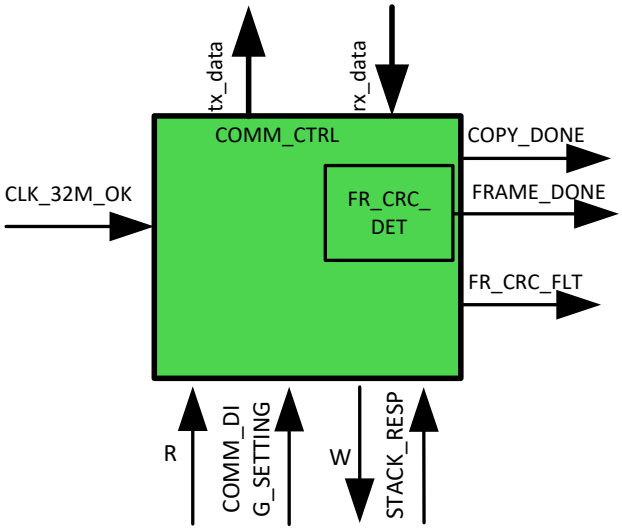


Figure1 COMM_CTRL diagram

Module input/output list

Name	Dir	Width	Description	duration
reg_addr	O	16	Register address	Level(CLK_REG domain)
ini_addr	O	16	Initial register address	Level(CLK_REG domain)
lsb_bit	O	1		
byte_cnt	O	7		
state	O	3	Receive frame state	
bytes	O	4	Frame operation bytes numer	Level(CLK_REG domain)
wr_update	O	1	Write update pulse	1 CLK_REG
rx_done	O	1		
wr_data	O	128	Received data buffer	Level(CLK_REG domain)
dev_addr_dlv	O	1	Device address identify delivery	Level(CLK_REG domain)
dev_addr_dlv_spi	O	1	Device address identify delivery when SPI_EN high	Level(CLK_REG domain)
dev_addr0	O	8	Device address	Level(CLK_REG domain)
tx_data	O	9	Data to be transmitted	Level(CLK_REG domain)
state_tx_init	O	1	tx_state is STATE_INIT	Level(CLK_REG domain)
state_tx_bps	O	1	tx_state is STATE_BYPASS	Level(CLK_REG domain)
state_tx_pec	O	1	tx_state is STATE_PEC	Level(CLK_REG domain)
state_rx_init	O	1	state is STATE_INIT	Level(CLK_REG domain)
state_rx_bps	O	1	state is STATE_BYPASS	Level(CLK_REG domain)
state_rx_cur_addr	O	1	state is STATE_CUR_ADR	Level(CLK_REG domain)

response	O	1	Response to Address Identify/Read command	Level(CLK_REG domain)
pos_response	O	1	Positive edge of response	1 CLK_REG
neg_response	O	1	Negative edge of response	1 CLK_REG
pos_next_rps	O	1	Current device is the next to response	1 CLK_32M
bypass_end	O	1	Mark the ending time of a bypass byte	1 CLK_REG
rx_dev_addr	O	1	Receive 9'h1C0 when state is STATE_INT or STATE_BYPASS	4 CLK_32M
cnt_rx_byte_num	O	8	Rx byte numer	Level(8M domain)
rd	O	1	Current device in read station	Level(8M domain)
tx_add_reg_addr	O	1	tx register address adds bytes end pulse	1 CLK_32M
tx_state_addr	O	1	tx_state is STATE_ADDR	Level(CLK_REG domain)
stack	O	1	Stack operation	Level(CLK_REG domain)
rd_clr_CV_CNT	O	1		
neg_rx_en	O	1	Negedge of rx_en	1 CLK_REG
next_rps	O	1	Current device is the next to response	Level(8M domain)
SOFB	O	1		
IERR	O	1		
TXDIS	O	1		
SOF	O	1		
UNEXP_C	O	1		
CRC	O	1		
CONFL	O	1		
RR	O	1		
neg_tx_init	O	1	Pulse after tx_state jump to STATE_INIT from STATE_PEC	1 CLK_REG
tx_phase2_flag	O	1		
FRAME_DONE	O	1	A complete frame is received.	Level(CLK_REG domain)
FR_CRC_FLT	O	1	Frame CRC fault	Level(CLK_REG domain)
adr_idty_done	O	1	Address identify done	Level(8M domain)
wait_re_clocking	O	14	Wait time before transmitting	CLK_REG domain
tx_start	O	1	Transmitting start	1 CLK_REG
tx_capture	O	1	Delayed 2 CLK_REG signal of tx_start	1 CLK_REG
COPY_NXT	O	1	tell SPI_BASIC to give next rx_data	Level(CLK_REG domain)
RD_DET	O	1	tell SPI_BASIC an Address Identify or Read Command initial byte is received	Level(CLK_REG domain)
RESP	O	1	Response by bridge device, tell SPI_BASIC an Address Identify or Read Command with right CRC is received	Level(CLK_REG domain)
SPI_DIR	O	1	Direction configured by i2c_master	Level(8M domain)
tail_blanking	O	1	Tail blanking time	Level(CLK_REG domain)
FCOMM_FLT_IN	O	1	Communication fault received from last device	Level(CLK_REG domain)
CLK_32M_SC	I	1	CLK_32M after scan mux	
resetb_CLK	I	1	Asynchronous reset signal(synchronously released)	
rstb_32M_ok_and_sr	I	1	CLK_32M_OK low or soft reset	

SOFT_RSTB_REG	I	1	Soft reset from COMM_REG directly	Level(CLK_REG domain)
CLK_REG_SC	I	1	Scan-mux result of 8MHz clock from CLK_32M	8MHz
read_data	I	8	Read data from COMM_REG	
SPI_EN	I	1	SPI enable	async
neg_rx_en_dsy	I	1	negedge of rx_en_dsy	4 CLK_32M
neg_rx_en_dsy_8M	I	1	negedge of rx_en_s_dsy or rx_en_n_dsy	1 CLK_REG
neg_rx_en_s_dsy	I	1	negedge of rx_en_s_dsy	4 CLK_32M
neg_rx_en_n_dsy	I	1	negedge of rx_en_n_dsy	4 CLK_32M
SPI_RX_EN	I	1	A byte is received by SPI interface	4 CLK_32M
SPI_DIR_REG	I	1	SPI_DIR setting from COMM_REG	Level(CLK_REG domain)
rst_spi	I	1	When SPI_EN, reset spi	4 CLK_32M
TX_DONE	I	1	All TX FIFOis empty and timeout	1 CLK_REG
DIR_SEL	I	1	Direction selection from COMM_REG	Level(CLK_REG domain)
DEV_ADD	I	7	Device address from COMM_REG	Level(CLK_REG domain)
rx_en_n	I	1	daisy chai signal is being received on N port	
rx_en_s	I	1	daisy chai signal is being received on S port	
TX_EN_N	I	1	enable daisy chain transmitting on N port	
TX_EN_S	I	1	enable daisy chain transmitting on S port	
STACK_RESPONSE	I	6	Internal time between response bytes	Level(8M domain)
send_char_end_pos	I	1	mark byte transmitting end time	4 CLK_32M
tx_crc	I	16	crc16 result of tx_one	
reg0000	I	8	Reg0000 from COMM_REG	Level(CLK_REG domain)
D2A_RX_EN_S	I	1	enable daisy chain receiving on S port	
D2A_RX_EN_N	I	1	enable daisy chain receiving on N port	
D2A_TOP_DEV	I	1	Current device is fastest from bridge	Level(8M domain)
neg_TX_EN_S	I	1	negedge of TX_EN_S	1 CLK_32M
neg_TX_EN_N	I	1	negedge of TX_EN_N	1 CLK_32M
clr_crc_dsy	I	1	crc clear	3~4 CLK_32M
clr_crc_spi	I	1	At SPI_CSB negedge, clr_crc_spi generate one pulse to set CRC result to default FFFF when SPI_EN high.	4 CLK_32M
TX_timeout	I	1	no data to tranmit for a timeout time when TX_EN_X high	Level(CLK_32M domain)
FLT_WAKE	I	1	Any unmasked fault happens	4 CLK_32M

Clock Domain

The clock for COMM_CTRL is CLK_REG_SC.

COMM_CTRL function description

1 Frame requirements

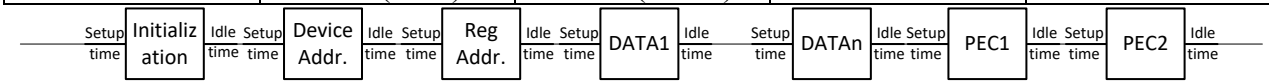
1.1 frame Packet

There are two kinds of frames: command and response.

All frame packets are framed by characters: Initialization Character, Data Character, PEC character.

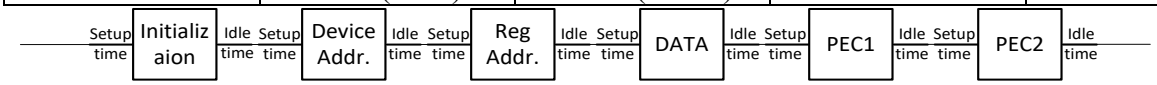
For Single Device Write Command, the frame is:

1Character	1Character	2Character	nCharacter	2Character
Initialization	Device Addr.(8bits)	Reg. Addr.(16bits)	Data (8n bits)	PEC(16bits)



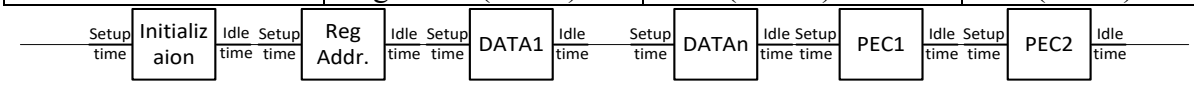
For Single Device Read Command, the frame is:

1Character	1Character	2Character	1Character	2Character
Initialization	Device Addr.(8bits)	Reg. Addr.(16bits)	Data (7 bits)	PEC(16bits)



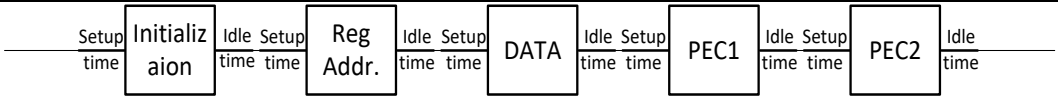
For Stack Devices Write Command, the frame is:

1Character	2Character	nCharacter	2Character
Initialization	Reg. Addr.(16bits)	Data (8n bits)	PEC(16bits)



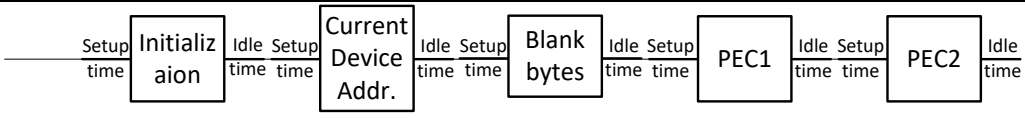
For Stack Devices Read Command, the frame is:

1Character	2Character	1Character	2Character
Initialization	Reg. Addr.(16bits)	Data (7 bits)	PEC(16bits)



For Address Identification Command, the frame is:

1Character	1Character	2Character	2Character
Initialization	Current Device Addr.(8bits)	Blank bytes(16bits)	PEC(16bits)



For Single Device Read Response, the frame is:

1Character	1Character	2Character	nCharacter	2Character
Initialization	Device Addr.(8bits)	Reg. Addr.(16bits)	Data (8n bits)	PEC(16bits)

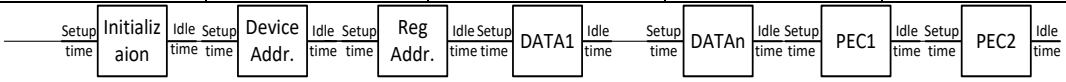


Table1 frame packet

For Stack Devices Read Response , the frame is to connect Single Device Read Response one by one.

1.2 Data Character

For frame initialization character: (HWR015_COMM_CTRL)

Data7	Data6	Data5	Data4	Data3	Data2	Data1	Data0
1 (command)	000: Single Device Read			The total only data bytes 0000-1111: 1 byte to 16 bytes. For Address Identification Command, the data are 0000.			
	001: Single Device Write						
	010: Stack Devices Read						
	011: Stack Devices Write						
	100: Address Identification						
	101: reserved						
	110: reserved						
	111: reserved						
0 (response)	The all bytes 0000000-1111111: 1 byte to 128 bytes.						

Table2 INIT byte definition

2 receiving state machine

8 states are realized in receiving state machine in Figure2.(HWR007_COMM_CTRL, HWR010_COMM_CTRL) State name is corresponding to Table1. Note that cnt_3_byt is used to merge 2 byte “blank bytes” into STATE_CUR_ADR.

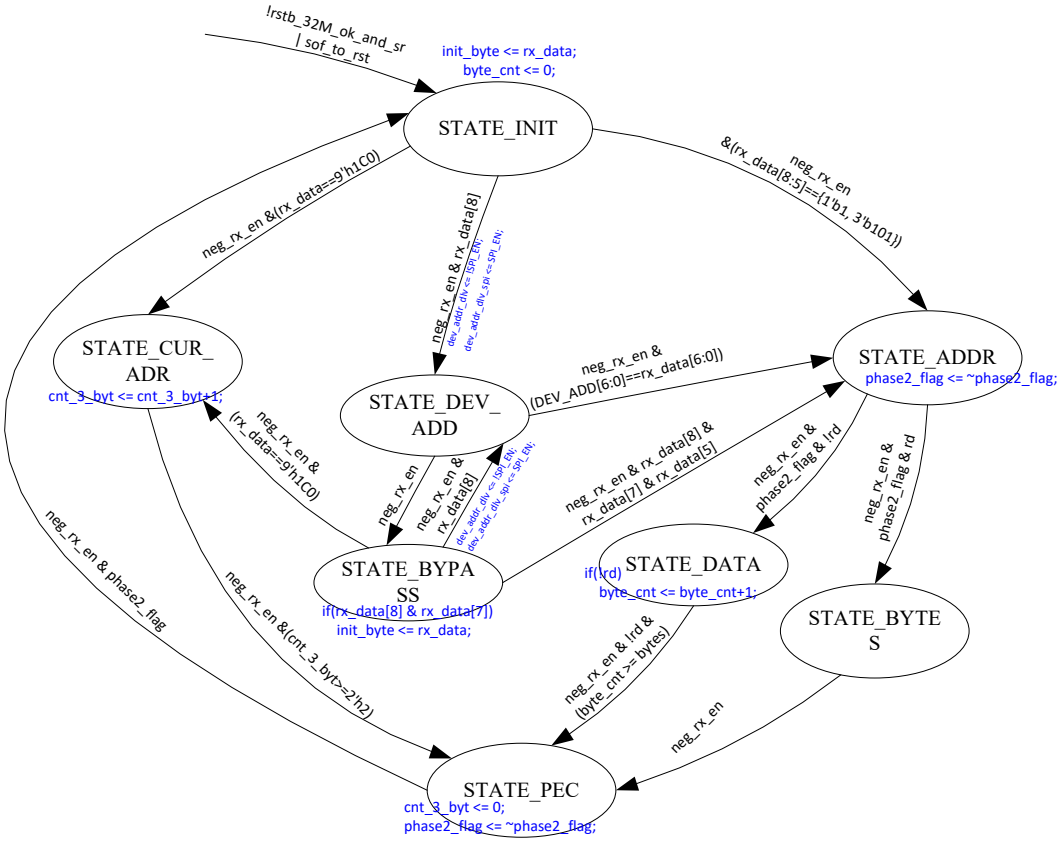


Figure2 receiving state machine

2.1 CLK_32M_OK low reset:

State can be reset to STATE_INIT when CLK_32M_OK is low by rstb_32M_ok_and_sr low. (HWR002_COMM_CTRL)

2.2 Writing and reading register bit:

In state STATE_DEV_ADD, device address is recorded with rx_data[8:0]. When device address matches input data, state jumps to STATE_ADDR. In state STATE_ADDR, initial register address is recorded with rx_data[8:0]. In state STATE_BYTES, bytes number is recorded with rx_data[8:0]. In write command, after STATE_PEC done, if CRC is right(CRC result is 16'h0), corresponding register in COMM_REG can be written. (HWR003_COMM_CTRL) In read command, after STATE_PEC done, if CRC is right(CRC result is 16'h0), tx_state starts to response data. (HWR004_COMM_CTRL)

2.3 STATE_DATA for writing and STATE_BYTES for reading:

In state STATE_DATA, received data rx_data[7:0] is shifted to buffer wr_data[127:0] for writing registers in COMM_REG. In state STATE_BYTES, the byte number to be read for read command is recorded in rd_bytes[6:0].(HWR008_COMM_CTRL)

2.4 receiving CRC caulculation:

Sub module FR_CRC_DET calculates 16bit IBM CRC result of rx_data[7:0] every byte in a frame. The polynomial is 8005(x^16+x^15+x^2+1) with 0xFFFF initialization. As daisy chain data are LSB-first and spi data are MSB-first, parallel algorithm is used. When a frame ends, if the result of CRC is 0, the frame is rightly received. If the result of CRC is not 0, the frame is wrong.(HWR009_COMM_CTRL)

2.5 SOF(Start Of Frame):

When rx_en and rx_data[8] high, sof_to_rst is high, SOF bit is recognized high. Whatever state is, it jumps to STATE_INIT. (HWR011_COMM_CTRL)

3 Transmitting state machine

7 states are realized in receiving state machine in Figure3. (HWR007_COMM_CTRL, HWR010_COMM_CTRL)

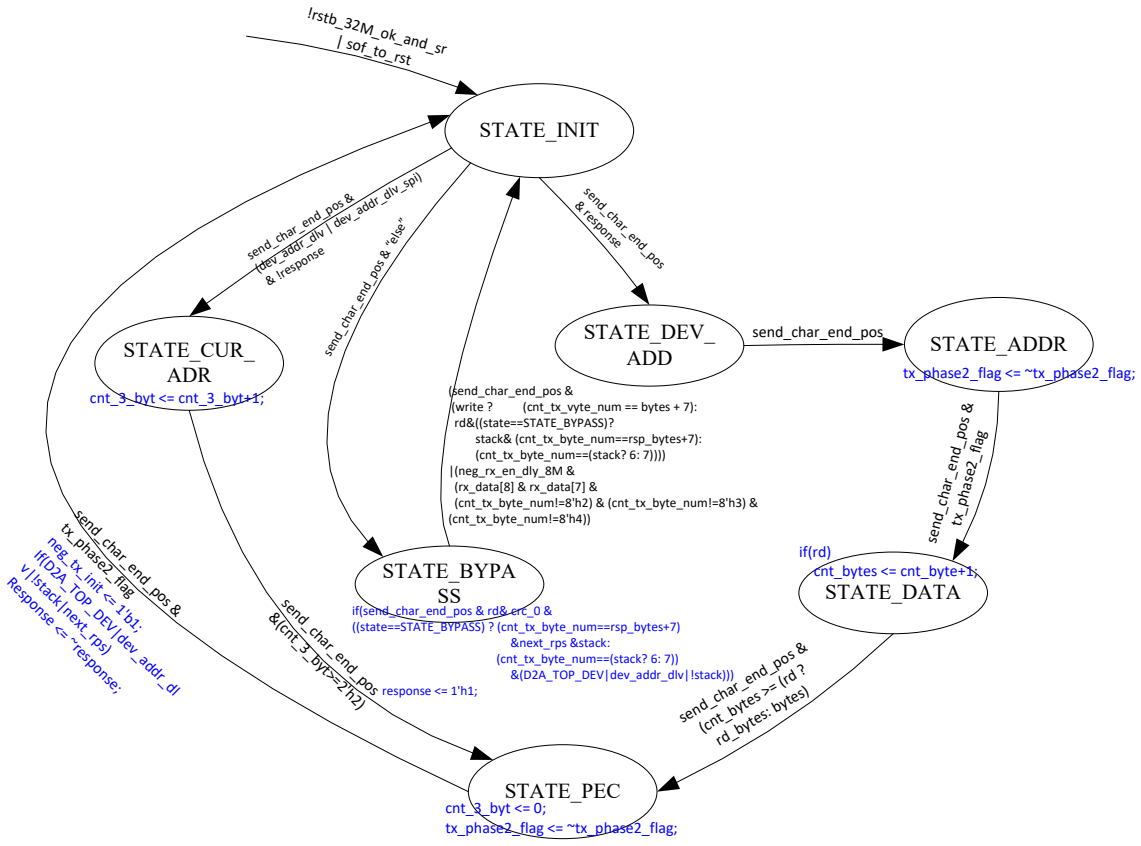


Figure3 transmitting state machine

3.1 Frame propagation:

No matter what rx_data[8:0] is, tx_data[8:0] delivers the data to next device.(HWR001_COMM_CTRL)

For Address Identify Command, the propagation starts after 72us. For other command, the propagation starts after a byte is completely received. (HWR016_COMM_CTRL)

3.2 CLK_32M_OK low reset:

Tx_state can be reset to STATE_INIT when CLK_32M_OK is low by rstb_32M_ok_and_sr low. (HWR002_COMM_CTRL)

3.3 Response:

Response is a signal to mark current device response time. For Address Identify Command and Single read Command, response is high when coping frame ends. For Stack Read Command, if D2A_TOP_DEV is high, response still is high when coping frame ends. If D2A_TOP_DEV is low, response is high only when the last device's response frame ends (next_rps high, which means the current device is the next to response). (HWR012_COMM_CTRL)

COMM_CTRL start responsing when response is high. When responsing, when tx_state is STATE_DATA, transmit data tx_data[8:0] are grabbed from COMM_REG via read_data[7:0]. (HWR005_COMM_CTRL)

3.4 transmitting CRC caulculation:

Tx_crc[15:0] is the 16bit IBM CRC result of previous transmitted data calculated in DS_BASIC. The polynomial is 8005(x^16+x^15+x^2+1) with 0xFFFF initialization. Tx_crc[15:0] is realized in serial algorithm. When tx_state is STATE_PEC, tx_data[7:0] is tx_crc[15:8] for the 1st byte, and is tx_crc[7:0] for the 2nd byte. (HWR009_COMM_CTRL)

3.5 wait_re_clocking:

Wait_re_clocking[13:0] is a CLK_REG domain counter defined for wait re-clocking time. When responding to read commands, interval time between response bytes is adjustable. Wait_re_clocking[13:0] counts up to (14+(STACK_RESPONSE*2)). STACK_RESPOSNE[5:0] is register bits set in COMM_REG. When STACK_RESPONSE[5:0] is 6'h0, the interval time between response bytes is 0.25us. When STACK_RESPONSE[5:0] is 6'h3F, the interval time between response bytes is 15.75us. (HWR013_COMM_CTRL)

3.6 FRAME_DONE:

When state goes back to STATE_INIT, or state keeps in STATE_BYPASS and received bytes number equals to respected number, FRAME_DONE is updated with crc_0. FRAME_DONE clear to 0 at the next CLK_REG to ensure it is a pulse.(HWR013_COMM_CTRL)

3.7 adr_idty_done:

Adr_idty_done means address identify done, when it is high, device get an address in the whole daisy chain. Adr_idty_done is initially low.

For AFE application(SPI_EN low), after device responded to Address Identify Command, adr_idty_done is high. Adr_idty_done can only be cleared by SOFT_RSTB_REG, cannot be cleared by CLK_32M_OK low or SOF bit.

For bridge application(SPI_EN high), as bridge is connected to MCU directly, its device address is always 0. So it doesn't respond to Address Identify Command. When SPI_EN high, adr_idty_done is high as if its device address has been updated, received Address Identify Command is only propagated to next device.(HWR019_COMM_CTRL)

3.8 SPI related outputs:

For bridge application(SPI_EN high), RESP, RD_DET and COPY_NXT are output for SPI_BASIC.

RESP equals to RD_DET when valid(with crc_0 information) read frame is received, and reset to 0 when CLR_DET or TX_DONE. (HWR020/021_COMM_CTRL)

RD_DET is high when read command is recognized from receiving INIT byte, and reset to 0 when CLR_DET or TX_DONE. (HWR020/022_COMM_CTRL)

COPY_NXT is high when send_char_end_pos(a byte has been transmitted by TX ports) is high. COPY_NXT is low when neg_rx_en from SPI port (a new byte is received from SPI port) is high. (HWR023_COMM_CTRL)