BASIC_CTRL IP SPEC

Table of Contents

Introduction

The BASIC_CTRL module is used to convert input dual daisy chain signals or input SPI signal to byte data rx_data[8:0], and convert byte data tx_data[8:0] to dual daisy chain outputs or SPI output.

The DS_BASIC module is used to convert input dual daisy chain signals to byte data rx_data[8:0], and convert byte data tx_data[8:0] to dual daisy chain outputs.

The SPI_BASIC module is used only for bridge application. It converts input SPI signal to rx_data[8:0], or converts tx_data[8:0] to SPI output.

Feature

Key features of the BASIC_CTRL module are:

- Adjust for both AFE and bridge application
  - Used as AFE: SPI_EN low
  - Used as bridge: SPI_EN high
- Direction control

Key features of the DS_BASIC module are:

- Analysis received daisy chain data
- Send daisy chain data
- FLT_WAKE receiving and indicating in daisy chain communication
- Direction control

Key features of the SPI_BASIC module are:

- Analysis received SPI data

- Send SPI data

Register Definition

Register Map

Table 1 1BASIC_CTRL Register Map

Name	Add	D7	D6	D5	D4	D3	D2	D1	D0	Default
COMM_CONF2	0x0003	COMN_TX_DIS	COMS_TX_DIS	STACK_RESP_CMD<5:0>						00
CTRL2	0x2003						CMP_BIST_GO	ADD_W_EN	SPI_DIR	

Functional Details

Block Diagram

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

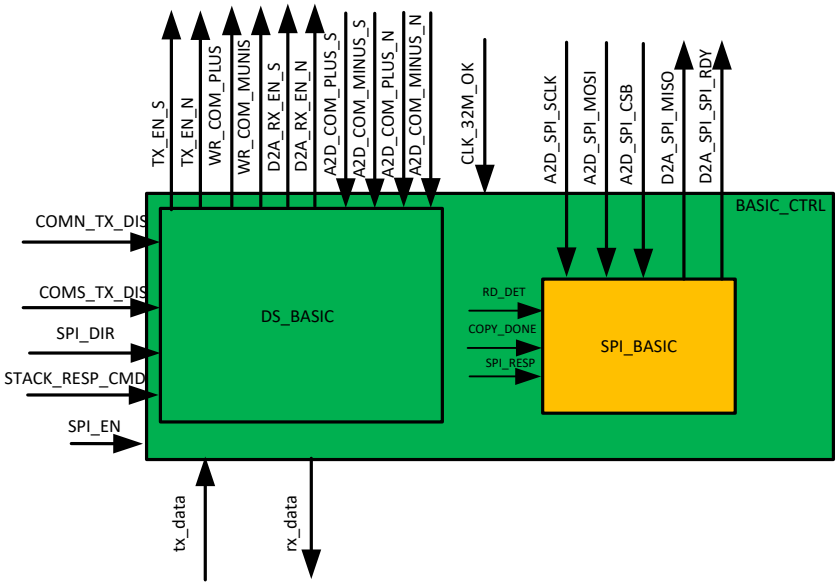

Figure 1 BASIC_CTRL diagram

Module input/output list

Name	Dir	Width	Discirption	duration
rev_dsy_data	O	9	received daisy chain data	Level(32M domain)
neg_rx_en_dsy	O	1	negedge of rx_en_dsy	4 CLK_32M
neg_rx_en_dsy_8M	O	1	negedge of rx_en_s_dsy or rx_en_n_dsy	1 CLK_REG
neg_rx_en_s_dsy	O	1	negedge of rx_en_s_dsy	4 CLK_32M
neg_rx_en_n_dsy	O	1	negedge of rx_en_n_dsy	4 CLK_32M
MISS	O	1		
ORDER	O	1		
SYNCT	O	1		
SYNCD	O	1		
BIT	O	1		
WR_COM_PLUS	O	1	daisy chain output data in positive phase	8 CLK_32M
WR_COM_MINUS	O	1	daisy chain output data in negtive phase	8 CLK_32M
TX_EN_S	O	1	enable daisy chain transmitting on S port	
TX_EN_N	O	1	enable daisy chain transmitting on N port	
D2A_RX_EN_S	O	1	enable daisy chain receiving on S port	
D2A_RX_EN_N	O	1	enable daisy chain receiving on N port	
send_char_end_pos	O	1	mark byte transmitting end time	4 CLK_32M

tx_crc	O	16	crc16 result of tx_one	
rx_en_n	O	1	daisy chai signal is being received on N port	
rx_en_s	O	1	daisy chai signal is being received on S port	
rx_en	O	1	daisy chai signal is being received	
tx_en_32M	O	1	sync send_start with CLK_32M	2 CLK_32M
neg_TX_EN_S	O	1	negedge of TX_EN_S	1 CLK_32M
neg_TX_EN_N	O	1	negedge of TX_EN_N	1 CLK_32M
clr_crc_dsy	O	1	Daisy chain crc clear	3~4 CLK_32M
clr_crc_spi	O	1	Spi crc clear, In RX state when the falling edge of A2D_SPI_CSB	1 CLK_REG
rx_data	O	9	Received data from daisy chain or SPI	Level(32M domain)
TX_timeout	O	1	no data to tranmit for a timeout time when TX_EN_X high	
pos_TBYTE_FAST	O	1	fault flag: receiving data is too fast	4 CLK_32M
pos_TBYTE_TO	O	1	fault flag: receiving data is too slow	4 CLK_32M
D2A_SPI_SPI_RDY	O	1	Indicate slave can be read or write by master	
D2A_SPI_MISO	O	1	Master input slave output, MSB first	
SPI_RX_EN	O	1	Indicate RX_DATA_SPI is update	1 CLK_REG
SPI_CLR_DET	O	1	CLR_DET module detect COMM_CLEAR command after detect falling edge and then receive 8'h00	1 CLK_REG
RX_FIFO_OF	O	1		1 CLK_REG
TX_FIFO_OF	O	1		1 CLK_REG
TX_FIFO_UF	O	1		1 CLK_REG
TX_DONE	O	1	All TX FIFOis empty and timeout	1 CLK_REG
rst_spi	O	1	Reset tx logics when SPI_EN high	1 CLK_REG
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	
CLK_REG_SC	I	1	Scan-mux result of 8MHz clock from CLK_32M	8MHz
CLK_REG	I	1	8MHz clock divided from CLK_32M	8MHz
SOFT_RSTB_32M	I	1	Soft reset	
SLEEP_MODE	I	1	Synchronous result of A2D_SLEEP_1P8 by CLK_256K_SC	Level
rx_en_256K	I	1	Daisy chain or spi rx_en	Level(CLK_256K domain)
A2D_COM_PLUS_S	I	1	Positive input comparator in S port	async
A2D_COM_MINUS_S	I	1	Negative input comparator in S port	async
A2D_COM_PLUS_N	I	1	Positive input comparator in N port	async
A2D_COM_MINUS_N	I	1	Negative input comparator in N port	async
TONE_TRANS_EN_N	I	1	N port tone transmission enable	Level(CLK_OUT domain)
TONE_TRANS_EN_S	I	1	S port tone transmission enable	Level(CLK_OUT domain)
state_tx_init	I	1	tx_state is STATE_INIT	1 CLK_REG

state_tx_pec	I	1	tx_state is STATE_PEC	1 CLK_REG
state_rx_init	I	1	state is STATE_INIT	1 CLK_REG
state_rx_bps	I	1	state is STATE_BYPASS	1 CLK_REG
response	I	1	Current device response	Level(8M domain)
pos_response	I	1	Positive edge of response	1 CLK_REG
neg_response	I	1	Negative edge of response	1 CLK_REG
pos_next_rps	I	1	Current device is the next to response	1 CLK_32M
bypass_end	I	1	Mark the ending time of a bypass byte	1 CLK_REG
rx_dev_addr	I	1	Receive 9'h1C0 when state is STATE_INT or STATE_BYPASS	4 CLK_32M
cnt_rx_byte_num	I	8	Rx byte numer	Level(8M domain)
rd	I	1	Current device in read station	Level(8M domain)
D2A_TOP_DEV	I	1	Current device is fastest from bridge	Level(8M domain)
stack	I	1	Stack operation	Level(8M domain)
COMN_TX_DIS	I	1	N port transmit disable	Level(8M domain)
COMS_TX_DIS	I	1	S port transmit disable	Level(8M domain)
wait_re_clocking	I	14	Wait time before transmitting	CLK_REG domain
adr_idty_done	I	1	Address identify done	Level(8M domain)
tail_blanking	I	1	Tail blanking time	Level(8M domain)
neg_rx_en	I	1	Negedge of rx_en	1 CLK_REG
next_rps	I	1	Current device is the next to response	Level(8M domain)
neg_tx_init	I	1	Pulse after tx_state jump to STATE_INIT from STATE_PEC	1 CLK_REG
STACK_RESPONSE	I	6	Internal time between response bytes	Level(8M domain)
FRAME_DONE	I	9	Received frame done	1 CLK_REG
FR_CRC_FLT	I	1	Frame CRC fault	1 CLK_REG
A2D_SPI_SCLK	I	1	Spi clock input	N/A
A2D_SPI_MOSI	I	1	Master output slave input, MSB first	N/A
A2D_SPI_CSB	I	1	Chip selection input	N/A
tx_data	I	9	Data to be transmit	CLK_REG domain
tx_start	I	1	Transmitting start	1 CLK_REG
tx_capture	I	1	Tx_data enable	1 CLK_REG
SPI_EN	I	1	Enable SPI	Async
SPI_DIR	I	1	"1" for north interface, "0" for south interface	1 CLK_REG
dev_addr_dlv	I	1	Device address identify delivery	Level(8M domain)
dev_addr_dlv_spi	I	1	Device address identify delivery when SPI_EN high	Level(8M domain)
RESP	I	1	RESP =1 , indicate is maser read state RESP =0 , indicate is master write state	Level(8M domain)
RD_DET	I	1	Device address identify delivery when SPI_EN high	Level(8M domain)
COPY_NXT	I	1	SPI to send next data to COMM_CTRL	Level(8M domain)

Clock Domain

The clock for BASIC_CTRL is CLK_32M_SC and A2D_SPI_SCLK.
For DS_BASIC, CLK_32M_SC is used.
For SPI_BASIC, both CLK_32M_SC and A2D_SPI_SCLK are used.

BASIC_CTRL function description

BASIC_CTRL module is the top module that instanced DS_BASIC and SPI_BASIC.

Only 2 signals are generated in BASIC_CTRL:
Rx_data(HWR001_BASIC_CTRL, HWR003_BASIC_CTRL):
When receiving data from SPI interface(SPI_EN high, and the corresponding D2A_RX_EN_x depended on SPI_DIR is high), choose rx_data_spi[8:0] as input data; else choose rev_dsy_data[8:0] as input data.
Rst_spi:
logic and result of SPI_EN and SPI_CLR_DET.

Outputs TX_EM_S, TX_EN_N, WR_COM_PLUS and WR_COM_MINUS are from DS_BASIC directly. These 4 signals can transmit data at right directions.(HWR001_BASIC_CTRL)

(Unrealized)
(HWR002_BASIC_CTRL) only sub-module DS_BASIC can be synchronously reset when CLK_32M_OK is low. Sun-module SPI_BASIC cannot.