COMM\_CTRL IP SPEC

Table of Contents

[Introduction 2](#_Toc118214654)

[Feature 2](#_Toc118214655)

[Register Definition 2](#_Toc118214656)

[Register Map 2](#_Toc118214657)

[Functional Details 3](#_Toc118214658)

[Block Diagram 3](#_Toc118214659)

[Module input/output list 3](#_Toc118214660)

[COMM\_CTRL function description 6](#_Toc118214661)

## 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\_CTRL 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

#### Frame requirements

* 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. 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

* 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)

* 1. 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)

* 1. 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)

* 1. 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)

* 1. 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)

* 1. 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)

* 1. 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)

* 1. 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)