COMM\_REG 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\_REG function description 6](#_Toc118214661)

## Introduction

The COMM\_REG module is to receive writing signals and change internal registers, and to generate setting bits for other modules, or to read out data for daisy chain or SPI interface. For fault register bits, excepting for writing from interface, inputs also come from fault inputs.

## Feature

Key features of the COMM\_REG module are:

•support writing registers

•support MTP writing and can load data from MTP

•read-only bits can be read out by communication interface

•fault registers can be set by fault inputs, and cleared by writing via interface

## Register Definition

### Register Map

|  |  |  |
| --- | --- | --- |
|  | Covered by MTP\_CRC? | Covered by CONF\_CRC? |
| SPF\_CONF |  | Y |
| SF\_CONF |  | Y |
| LF\_SM\_CONF |  | Y |
| SPF\_FUNC |  |  |
| SF\_FUNC |  |  |
| LF\_SM\_FUNC |  |  |
| TM |  | Y |
| TRIM | Y | Y |
| FLT\_SUM |  |  |
| FLT\_BIT |  |  |

Table1 block class

| **Name** | **Add** | **D7** | **D6** | **D5** | **D4** | **D3** | **D2** | **D1** | **D0** | **Default** | **Block** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| DEV\_ADD1 | 0x0000 | TOP\_DEV | ADD<6:0> | | | | | | | 80 | SPF\_CONF |
| DEV\_ADD2 | 0x0001 |  | NUM<6:0> | | | | | | | 00 | SPF\_CONF |
| COMM\_CONF1 | 0x0002 | CB\_STL<4:0> | | | | |  |  | I2C\_MAS\_EN | 08 | SPF\_CONF |
| COMM\_CONF2 | 0x0003 | COMN\_TX\_DIS | COMS\_TX\_DIS | STACK\_RESP\_CMD<5:0> | | | | | | 00 | SPF\_CONF |
| COMM\_TO | 0x0004 | LCTO\_SEL<1:0> | | LONG<2:0> | | | SHORT<2:0> | | | BB | SF\_CONF |
| COW\_DET | 0x0005 |  |  |  |  | C\_OW\_TDIS<3:0> | | | | 00 | LF\_SM\_CONF |
| GAP\_CMP1 | 0x0006 | GPIO\_GAP\_THR<2:0> | | | CELL\_GAP\_THR<4:0> | | | | | 00 | LF\_SM\_CONF |
| GAP\_CMP2 | 0x0007 |  | | | CELL\_GAP\_DEGL<4:0> | | | | | 07 | LF\_SM\_CONF |
| ADC\_CONF1 | 0x0008 | DLPF\_FC<2:0> | | |  | ADC\_CLK<1:0> | | ADC\_MODE<2:0> | | 09 | SPF\_CONF |
| ADC\_CONF2 | 0x0009 | CH\_TOP\_STL<3:0> | | | | CH\_STL<3:0> | | | | 55 | SPF\_CONF |
| ADC\_CONF3 | 0x000A | CH\_BOT\_STL<3:0> | | | |  | ADC\_CHP\_EN | CH\_DT<1:0> | | 55 | SPF\_CONF |
| ADC\_CONF4 | 0x000B | GPIO7\_REF\_SEL | GPIO6\_ REF\_SEL | GPIO5\_ REF\_SEL | GPIO4\_ REF\_SEL | GPIO3\_ REF\_SEL | GPIO2\_ REF\_SEL | GPIO1\_ REF\_SEL | GPIO0\_ REF\_SEL | 00 | SPF\_CONF |
| ADC\_CONF5 | 0x000C |  |  |  |  | GPIO11\_ REF\_SEL | GPIO10\_ REF\_SEL | GPIO9\_ REF\_SEL | GPIO8\_ REF\_SEL | 00 | SPF\_CONF |
| MON\_CONF | 0x000D | MON\_WAKE\_PERIOD<5:0> | | | | | |  | MON\_WAKE\_EN | 45 | SF\_CONF |
| OVUV\_OTUT \_CONF1 | 0x000E | OVUV\_OTUT\_EN |  |  | OVUV\_ DEGL<4:0> | | | | | 80 | LF\_SM\_CONF |
| OVUV\_OTUT \_CONF2 | 0x000F |  | OV\_THR<6:0> | | | | | | | 60 | LF\_SM\_CONF |
| OVUV\_OTUT \_CONF3 | 0x0010 |  | UV\_THR <6:0> | | | | | | | 60 | LF\_SM\_CONF |
| OVUV\_OTUT \_CONF4 | 0x0011 | OT\_ PACK\_THR<4:0> | | | | | UT\_ PACK\_THR<2:0> | | | 7E | LF\_SM\_CONF |
| OVUV\_OTUT \_CONF5 | 0x0012 | OT\_ PCB\_THR<4:0> | | | | | UT\_ PCB\_THR<2:0> | | | 7E | LF\_SM\_CONF |
| OVUV\_OTUT \_CONF6 | 0x0013 | GPIO7\_THR\_SEL | GPIO6\_THR\_SEL | GPIO5\_THR\_SEL | GPIO4\_THR\_SEL | GPIO3\_THR\_SEL | GPIO2\_THR\_SEL | GPIO1\_THR\_SEL | GPIO0\_THR\_SEL | 00 | LF\_SM\_CONF |
| OVUV\_OTUT \_CONF7 | 0x0014 |  |  |  |  | GPIO11\_THR\_SEL | GPIO10\_THR\_SEL | GPIO9\_THR\_SEL | GPIO8\_THR\_SEL | 00 | LF\_SM\_CONF |
| CB\_CONF1 | 0x0015 | CB\_ROT\_PACK\_THR<4:0> | | | | | CB\_DUTY<2:0> | | | 7F | LF\_SM\_CONF |
| CB\_CONF2 | 0x0016 | CB\_ROT\_PCB\_THR<4:0> | | | | | CB\_PERIOD<2:0> | | | 78 | LF\_SM\_CONF |
| CB\_CONF3 | 0x0017 | CB\_JOT\_THR<3:0> | | | | ROT\_PAUSE\_EN | ADC\_PAUSE\_EN | JOT\_PAUSE\_EN | FLT\_STOP\_EN | 00 | LF\_SM\_CONF |
| TWARN\_CONF | 0x0018 |  |  |  |  |  | TWARN\_THR<2:0> | | | 03 | LF\_SM\_CONF |
| GPIO\_PUPD\_EN1 | 0x0019 | GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 | 00 | SPF\_CONF |
| GPIO\_PUPD\_EN2 | 0x001A |  |  |  |  | GPIO11 | GPIO10 | GPIO9 | GPIO8 | 00 | SPF\_CONF |
| GPIO\_ASDIN\_EN1 | 0x001B | GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 | 00 | SPF\_CONF |
| GPIO\_ASDIN\_EN2 | 0x001C |  |  |  |  | GPIO11 | GPIO10 | GPIO9 | GPIO8 | 00 | SPF\_CONF |
| GPIO\_WEAK\_PUPD\_EN1 | 0x001D | GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 | 00 | SPF\_CONF |
| GPIO\_WEAK\_PUPD\_EN2 | 0x001E |  |  |  |  | GPIO11 | GPIO10 | GPIO9 | GPIO8 | 00 | SPF\_CONF |
| FLT\_SUM\_MSK | 0x0100 |  |  |  | COW\_FLT\_MSK | SYS\_FLT\_MSK | GAP\_FLT\_MSK | OTUT\_FLT\_MSK | OVUV\_FLT\_MSK | 00 | SF\_CONF |
| COW\_FLT\_MSK1 | 0x0101 | C7\_MSK | C6\_MSK | C5\_MSK | C4\_MSK | C3\_MSK | C2\_MSK | C1\_MSK | C0\_MSK | 00 | LF\_SM\_CONF |
| COW\_FLT\_MSK2 | 0x0102 | C15\_MSK | C14\_MSK | C13\_MSK | C12\_MSK | C11\_MSK | C10\_MSK | C9\_MSK | C8\_MSK | 00 | LF\_SM\_CONF |
| COW\_FLT\_MSK3 | 0x0103 |  |  |  |  |  | C18\_MSK | C17\_MSK | C16\_MSK | 00 | LF\_SM\_CONF |
| OV\_FLT\_MSK1 | 0x0104 | CH8\_MSK | CH7\_MSK | CH6\_MSK | CH5\_MSK | CH4\_MSK | CH3\_MSK | CH2\_MSK | CH1\_MSK | 00 | SF\_CONF |
| OV\_FLT\_MSK2 | 0x0105 | CH16\_MSK | CH15\_MSK | CH14\_MSK | CH13\_MSK | CH12\_MSK | CH11\_MSK | CH10\_MSK | CH9\_MSK | 00 | SF\_CONF |
| OV\_FLT\_MSK3 | 0x0106 |  |  |  |  |  |  | CH18\_MSK | CH17\_MSK | 00 | SF\_CONF |
| UV\_FLT\_MSK1 | 0x0107 | CH8\_MSK | CH7\_MSK | CH6\_MSK | CH5\_MSK | CH4\_MSK | CH3\_MSK | CH2\_MSK | CH1\_MSK | 00 | SF\_CONF |
| UV\_FLT\_MSK2 | 0x0108 | CH16\_MSK | CH15\_MSK | CH14\_MSK | CH13\_MSK | CH12\_MSK | CH11\_MSK | CH10\_MSK | CH9\_MSK | 00 | SF\_CONF |
| UV\_FLT\_MSK3 | 0x0109 |  |  |  |  |  |  | CH18\_MSK | CH17\_MSK | 00 | SF\_CONF |
| CELL\_GAP\_FLT\_MSK1 | 0x010A | CH8\_MSK | CH7\_MSK | CH6\_MSK | CH5\_MSK | CH4\_MSK | CH3\_MSK | CH2\_MSK | CH1\_MSK | 00 | LF\_SM\_CONF |
| CELL\_GAP\_FLT\_MSK2 | 0x010B | CH16\_MSK | CH15\_MSK | CH14\_MSK | CH13\_MSK | CH12\_MSK | CH11\_MSK | CH10\_MSK | CH9\_MSK | 00 | LF\_SM\_CONF |
| CELL\_GAP\_FLT\_MSK3 | 0x010C |  |  |  |  |  |  | CH18\_MSK | CH17\_MSK | 00 | LF\_SM\_CONF |
| GPIO\_GAP\_FLT\_MSK1 | 0x010D | GPIO7\_MSK | GPIO6\_MSK | GPIO5\_MSK | GPIO4\_MSK | GPIO3\_MSK | GPIO2\_MSK | GPIO1\_MSK | GPIO0\_MSK | 00 | LF\_SM\_CONF |
| GPIO\_GAP\_FLT\_MSK2 | 0x010E |  |  |  |  | GPIO11\_MSK | GPIO10\_MSK | GPIO9\_MSK | GPIO8\_MSK | 00 | LF\_SM\_CONF |
| OT\_FLT\_MSK1 | 0x010F | GPIO7\_MSK | GPIO6\_MSK | GPIO5\_MSK | GPIO4\_MSK | GPIO3\_MSK | GPIO2\_MSK | GPIO1\_MSK | GPIO0\_MSK | 00 | SF\_CONF |
| OT\_FLT\_MSK2 | 0x0110 |  |  |  |  | GPIO11\_MSK | GPIO10\_MSK | GPIO9\_MSK | GPIO8\_MSK | 00 | SF\_CONF |
| UT\_FLT\_MSK1 | 0x0111 | GPIO7\_MSK | GPIO6\_MSK | GPIO5\_MSK | GPIO4\_MSK | GPIO3\_MSK | GPIO2\_MSK | GPIO1\_MSK | GPIO0\_MSK | 00 | SF\_CONF |
| UT\_FLT\_MSK2 | 0x0112 |  |  |  |  | GPIO11\_MSK | GPIO10\_MSK | GPIO9\_MSK | GPIO8\_MSK | 00 | SF\_CONF |
| SYS\_FLT\_MSK1 | 0x0113 |  |  |  |  | TBYTE\_FAST\_MSK | TBYTE\_TO\_MSK | FCOMM\_FLT\_MSK | FR\_CRC\_MSK | 00 | LF\_SM\_CONF |
| SYS\_FLT\_MSK2 | 0x0114 | TWARN\_MSK | AGND\_OW\_MSK | CONF\_CRC\_MSK | MTP\_CRC\_MSK | CP\_OV\_MSK | CP\_UV\_MSK | VAA\_OV\_MSK | VAA\_UV\_MSK | 00 | LF\_SM\_CONF |
| SYS\_FLT\_MSK3 | 0x0115 |  |  | VDD\_OV\_MSK | VDD\_UV\_MSK | CLK\_256K\_OKB\_MSK | VDD\_OKB\_MSK | VAA\_OKB\_MSK | LCTO\_SD\_MSK | 00 | SF\_CONF |
| SYS\_FLT\_MSK4 | 0x0116 | CB\_CONF\_FLT\_MSK | SCTO\_MSK | LCTO\_MSK |  | RX\_FIFO\_OF\_MSK | TX\_FIFO\_OF\_MSK | TX\_FIFO\_UF\_MSK | CMP\_FLT\_MSK | 00 | SF\_CONF |
| SYS\_FLT\_MSK5 | 0x0117 |  |  |  |  |  | HBFAST\_MSK | HBTO\_MSK | FLT\_TONE\_DET\_MSK | 00 | SF\_CONF |
| MTP\_ADR\_MANU | 0x1FFB |  | ADR\_MANU<6:0> | | | | | | | X | SPF\_FUNC |
| MTP\_DIN\_MANU | 0x1FFC | DIN\_MANU<7:0> | | | | | | | | X | SPF\_FUNC |
| MTP\_DOUT | 0x1FFD | DOUT<7:0> | | | | | | | | X | SPF\_FUNC |
| MTP\_TEST | 0x1FFE | CLEN | CS\_MANU | MANU | SRL | MRGN | RD\_MANU | WR\_MANU | HVEN | X | SPF\_FUNC |
| MTP\_CTRL | 0x1FFF |  |  |  | ECED | WR\_NVM |  |  | BUSY | X | SPF\_FUNC |
| TM\_REGn  (n=1-10) | 0x0800-0x0809 |  |  |  |  |  |  |  |  | X | TM |
| TRIM\_ADC | 0x1000 |  | | | | | | | | X | TRIM |
| … |  | … | | | | | | | | X | TRIM |
| TRIM\_ANA |  |  | | | | | | | | X | TRIM |
| XY\_LOT |  |  | | | | | | | | X | TRIM |
| TRIM\_CRC | 0x107F |  | | | | | | | | X | TRIM |
| CONF\_ CRC\_H | 0x2000 | CRC<15:8> | | | | | | | | 00 | LF\_SM\_FUNC |
| CONF\_ CRC\_L | 0x2001 | CRC<7:0> | | | | | | | | 00 | LF\_SM\_FUNC |
| CTRL1 | 0x2002 | SRSTB | DIR\_SEL |  | WAKE\_TONE\_GEN | STA\_TONE\_GEN | SD\_TONE\_GEN | TO\_SD | TO\_SLEEP | 80 | SPF\_FUNC |
| CTRL2 | 0x2003 |  |  |  |  |  | CMP\_BIST\_GO | ADD\_W\_EN | SPI\_DIR | 00 | SF\_FUNC |
| ADC\_CTRL | 0x2004 | MON\_WAKE\_GO |  |  |  |  | FREEZE | ADC\_SGLE\_GO | ADC\_CNTI\_GO | 00 | SPF\_FUNC |
| CB\_CTRL | 0x2005 |  |  |  |  |  | CB\_MANU | CB\_PAUSE | CB\_GO | 00 | LF\_SM\_FUNC |
| CBFET\_EN1 | 0x2006 | CH8 | CH7 | CH6 | CH5 | CH4 | CH3 | CH2 | CH1 | 00 | LF\_SM\_FUNC |
| CBFET\_EN2 | 0x2007 | CH16 | CH15 | CH14 | CH13 | CH12 | CH11 | CH10 | CH9 | 00 | LF\_SM\_FUNC |
| CBFET\_EN3 | 0x2008 |  |  |  |  |  |  | CH18 | CH17 | 00 | LF\_SM\_FUNC |
| CB\_TO\_CHn\_H  (n=1-18) | 0x2009 | UNIT |  |  |  |  |  | TO<9:8> | | 83 | LF\_SM\_FUNC |
| CB\_TO\_CHn\_L  (n=1-18) | 0x200A | TO<7:0> | | | | | | | | FF | LF\_SM\_FUNC |
| GPIO\_PUPD1 | 0x202D | GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 | 00 | SF\_FUNC |
| GPIO\_ PUPD2 | 0x202E |  |  |  |  | GPIO11 | GPIO10 | GPIO9 | GPIO8 | 00 | SF\_FUNC |
| DIAG\_CTRL | 0x2100 |  |  | |  |  |  |  | C\_OW\_DET\_GO | 00 | LF\_SM\_FUNC |
| GPIO\_WEAK\_PUPD1 | 0x2101 | GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 | 00 | SF\_FUNC |
| GPIO\_WEAK\_PUPD2 | 0x2102 |  |  |  |  | GPIO11 | GPIO10 | GPIO9 | GPIO8 | 00 | SF\_FUNC |
| CS\_EN\_MANU1 | 0x2103 | CS7\_EN | CS6\_EN | CS5\_EN | CS4\_EN | CS3\_EN | CS2\_EN | CS1\_EN | CS0\_EN | 00 | SPF\_FUNC |
| CS\_EN\_MANU2 | 0x2104 | CS15\_EN | CS14\_EN | CS13\_EN | CS12\_EN | CS11\_EN | CS10\_EN | CS9\_EN | CS8\_EN | 00 | SPF\_FUNC |
| CS\_EN\_MANU3 | 0x2105 |  |  |  |  |  | CS18\_EN | CS17\_EN | CS16\_EN | 00 | SPF\_FUNC |
| I2C\_MAS\_CTRL | 0x2200 | STOP | RX | SR | ACK |  |  |  | TX | 00 | SF\_FUNC |
| I2C\_TR | 0x2201 | DATA<7:0> | | | | | | | | 00 | SF\_FUNC |
| I2C\_RD | 0x2202 | DATA<7:0> | | | | | | | | 00 | SF\_FUNC |
| RR\_CNT\_H | 0x3FFE | CNT<15:8> | | | | | | | | 00 | LF\_SM\_FUNC |
| RR\_CNT\_L | 0x3FFF | CNT<7:0> | | | | | | | | 00 | LF\_SM\_FUNC |
| CHn\_LFP\_H  (n=1-18) | 0x4000 |  | | | | | | | | 00 | SPF\_FUNC |
| CHn\_ LFP\_L  (n=1-18) | 0x4001 |  | | | | | | | | 00 | SPF\_FUNC |
| … |  |  | | | | | | | |  |  |
| AUX\_CHn\_ LFP\_H  (n=1-18) | 0x4024 |  | | | | | | | | 00 | LF\_SM\_FUNC |
| AUX\_CHn\_ LFP\_L  (n=1-18) | 0x4025 |  | | | | | | | | 00 | LF\_SM\_FUNC |
| … |  |  | | | | | | | |  |  |
| CHn\_H  (n=1-18) | 0x4048 |  | | | | | | | | 00 | LF\_SM\_FUNC |
| CHn\_L  (n=1-18) | 0x4049 |  | | | | | | | | 00 | LF\_SM\_FUNC |
| … |  |  | | | | | | | |  |  |
| AUX\_CHn\_H  (n=1-18) | 0x406C |  | | | | | | | | 00 | SF\_FUNC |
| AUX\_CHn\_L  (n=1-18) | 0x404D |  | | | | | | | | 00 | SF\_FUNC |
| … |  |  | | | | | | | |  |  |
| GPIOn\_H  (n=0-11) | 0x4090 |  | | | | | | | | 00 | SPF\_FUNC |
| GPIOn\_L  (n=0-11) | 0x4091 |  | | | | | | | | 00 | SPF\_FUNC |
| … |  |  | | | | | | | |  |  |
| AUX\_GPIOn\_H  (n=0-11) | 0x40A8 |  | | | | | | | | 00 | LF\_SM\_FUNC |
| AUX\_GPIOn\_L  (n=0-11) | 0x40A9 |  | | | | | | | | 00 | LF\_SM\_FUNC |
| … |  |  | | | | | | | |  |  |
| VPTAT\_H | 0x40C0 |  | | | | | | | | 00 | LF\_SM\_FUNC |
| VPTAT\_L | 0x40C1 |  | | | | | | | | 00 | LF\_SM\_FUNC |
| BG\_H | 0x40C2 |  | | | | | | | | 00 | LF\_SM\_FUNC |
| BG\_L | 0x40C3 |  | | | | | | | | 00 | LF\_SM\_FUNC |
| BG2\_H | 0x40C4 |  | | | | | | | | 00 | LF\_SM\_FUNC |
| BG2\_L | 0x40C5 |  | | | | | | | | 00 | LF\_SM\_FUNC |
| FR\_CNT\_H | 0x5000 | CNT<15:8> | | | | | | | | 00 | LF\_SM\_FUNC |
| FR\_CNT\_L | 0x5001 | CNT<7:0> | | | | | | | | 00 | LF\_SM\_FUNC |
| GPIO\_DIN1 | 0x5002 | GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 | 00 | SF\_FUNC |
| GPIO\_DIN2 | 0x5003 |  |  |  |  | GPIO11 | GPIO10 | GPIO9 | GPIO8 | 00 | SF\_FUNC |
| CB\_CH\_EN\_FULL\_DUTY1 | 0x5004 | CH8 | CH7 | CH6 | CH5 | CH4 | CH3 | CH2 | CH1 | 00 | SF\_FUNC |
| CB\_CH\_EN\_FULL\_DUTY2 | 0x5005 | CH16 | CH15 | CH14 | CH13 | CH12 | CH11 | CH10 | CH9 | 00 | SF\_FUNC |
| CB\_CH\_EN\_FULL\_DUTY3 | 0x5006 |  |  |  |  |  |  | CH18 | CH17 | 00 | SF\_FUNC |
| CB\_ODD\_CNT\_H | 0x5007 | CNT<15:8> | | | | | | | | 00 | SF\_FUNC |
| CB\_ODD\_CNT\_L | 0x5008 | CNT<7:0> | | | | | | | | 00 | SF\_FUNC |
| CB\_EVEN\_CNT\_H | 0x5009 | CNT<15:8> | | | | | | | | 00 | SF\_FUNC |
| CB\_EVEN\_CNT\_L | 0x500A | CNT<7:0> | | | | | | | | 00 | SF\_FUNC |
| FLT\_SUM | 0x5100 |  |  |  | COW\_FLT | SYS\_FLT | GAP\_FLT | OTUT\_FLT | OVUV\_FLT | 00 | FLT\_SUM |
| COW\_FLT1 | 0x5101 | C7 | C6 | C5 | C4 | C3 | C2 | C1 | C0 | 00 | FLT\_BIT |
| COW\_FLT2 | 0x5102 | C15 | C14 | C13 | C12 | C11 | C10 | C9 | C8 | 00 | FLT\_BIT |
| COW\_FLT3 | 0x5103 |  |  |  |  |  | C18 | C17 | C16 | 00 | FLT\_BIT |
| OV\_FLT1 | 0x5104 | CH8 | CH7 | CH6 | CH5 | CH4 | CH3 | CH2 | CH1 | 00 | FLT\_BIT |
| OV\_FLT2 | 0x5105 | CH16 | CH15 | CH14 | CH13 | CH12 | CH11 | CH10 | CH9 | 00 | FLT\_BIT |
| OV\_FLT3 | 0x5106 |  |  |  |  |  |  | CH18 | CH17 | 00 | FLT\_BIT |
| UV\_FLT1 | 0x5107 | CH8 | CH7 | CH6 | CH5 | CH4 | CH3 | CH2 | CH1 | 00 | FLT\_BIT |
| UV\_FLT2 | 0x5108 | CH16 | CH15 | CH14 | CH13 | CH12 | CH11 | CH10 | CH9 | 00 | FLT\_BIT |
| UV\_FLT3 | 0x5109 |  |  |  |  |  |  | CH18 | CH17 | 00 | FLT\_BIT |
| CELL\_GAP\_FLT1 | 0x510A | CH8 | CH7 | CH6 | CH5 | CH4 | CH3 | CH2 | CH1 | 00 | FLT\_BIT |
| CELL\_GAP\_FLT2 | 0x510B | CH16 | CH15 | CH14 | CH13 | CH12 | CH11 | CH10 | CH9 | 00 | FLT\_BIT |
| CELL\_GAP\_FLT3 | 0x510C |  |  |  |  |  |  | CH18 | CH17 | 00 | FLT\_BIT |
| GPIO\_GAP\_FLT1 | 0x510D | GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 | 00 | FLT\_BIT |
| GPIO\_GAP\_FLT2 | 0x510E |  |  |  |  | GPIO11 | GPIO10 | GPIO9 | GPIO8 | 00 | FLT\_BIT |
| OT\_FLT1 | 0x510F | GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 | 00 | FLT\_BIT |
| OT\_FLT2 | 0x5110 |  |  |  |  | GPIO11 | GPIO10 | GPIO9 | GPIO8 | 00 | FLT\_BIT |
| UT\_FLT1 | 0x5111 | GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 | 00 | FLT\_BIT |
| UT\_FLT2 | 0x5112 |  |  |  |  | GPIO11 | GPIO10 | GPIO9 | GPIO8 | 00 | FLT\_BIT |
| SYS\_FLT1 | 0x5113 |  |  |  |  | TBYTE\_FAST | TBYTE\_TO | FCOMM\_FLT | FR\_CRC | 00 | FLT\_BIT |
| SYS\_FLT2 | 0x5114 | TWARN | AGND\_OW | CONF\_CRC | MTP\_CRC | CP\_OV | CP\_UV | VAA\_OV | VAA\_UV | 00 | FLT\_BIT |
| SYS\_FLT3 | 0x5115 |  |  | VDD\_OV | VDD\_UV | CLK\_256K\_OKB | VDD\_OKB | VAA\_OKB | LCTO\_SD | 00 | FLT\_BIT |
| SYS\_FLT4 | 0x5116 | CB\_CONF\_FLT | SCTO | LCTO |  | RX\_FIFO\_OF | TX\_FIFO\_OF | TX\_FIFO\_UF | CMP\_FLT | 00 | FLT\_BIT |
| SYS\_FLT5 | 0x5117 |  |  |  |  |  | HBFAST | HBTO | FLT\_TONE\_DET | 00 | FLT\_BIT |
| TM\_KEY | 0x6000 | KEY<7:0> | | | | | | | | 00 | SF\_FUNC |

Table2 Register Map

## Functional Details

### Block Diagram

The following diagram shows the COMM\_REG architecture and internal modules and connections.



Figure1 COMM\_REG diagram

### Module input/output list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Dir | Width | Description | duration |
| DEV\_ADD | O | 7 | Device address | Level(CLK\_REG domain) |
| COMN\_TX\_DIS | O | 1 | N port transmission disable | Level(CLK\_REG domain) |
| COMS\_TX\_DIS | O | 1 | S port transmission disable | Level(CLK\_REG domain) |
| SPI\_DIR\_REG | O | 1 | 1: spi replace N port; 0: spi replace S port | Level(CLK\_REG domain) |
| WAKE\_TONE\_GEN | O | 1 | Wake tone generation | Level(CLK\_REG domain) |
| FREEZE | O | 1 | ADC results freeze for reading | Level(CLK\_REG domain) |
| DIR\_SEL | O | 1 | Indicate the receiving data direction  1: rx from N port; 0: rx from S port | Level(CLK\_REG domain) |
| STA\_TONE\_GEN | O | 1 | STA tone generation | Level(CLK\_REG domain) |
| TO\_SD | O | 1 | For analog | Level(CLK\_REG domain) |
| TO\_SLEEP | O | 1 | For analog | Level(CLK\_REG domain) |
| read\_data | O | 8 | Read data for daisy chain or SPI interface | Level(CLK\_REG domain) |
| reg0000~reg001E | O | 8\*31 | Register bits for CONF\_CRC | Level(CLK\_REG domain) |
| reg0100~reg0117 | O | 8\*24 | Register bits for CONF\_CRC | Level(CLK\_REG domain) |
| D2A\_TOP\_DEV | O | 1 | Top device in system | Level(CLK\_REG domain) |
| D2A\_LCTO\_SEL01 | O | 1 | For analog | Level(CLK\_REG domain) |
| D2A\_LCTO\_SEL10 | O | 1 | For analog | Level(CLK\_REG domain) |
| PROG\_LCTO | O | 3 | Program bits for Long timeout | Level(CLK\_REG domain) |
| PROG\_SCTO | O | 3 | Program bits for Long timeout | Level(CLK\_REG domain) |
| DLPF\_FC\_REG | O | 3 | Digital low pass filter stage setting | Level(CLK\_REG domain) |
| SD\_TONE\_GEN | O | 1 | Shut down tone generation | Level(CLK\_REG domain) |
| ADC\_SGLE\_GO | O | 1 | ADC single go | Level(CLK\_REG domain) |
| ADC\_CNTI\_GO | O | 1 | ADC continuous go | Level(CLK\_REG domain) |
| MON\_EN\_REG | O | 1 | Monitor enable bit | Level(CLK\_REG domain) |
| MON\_WAKE\_GO | O | 1 | Monitor wake go | Level(CLK\_REG domain) |
| ADR\_MANU | O | 7 | Manual address setting, for MTP\_TOP | Level(CLK\_REG domain) |
| CLEN | O | 1 | For MTP interface | Level(CLK\_REG domain) |
| CS\_MANU | O | 1 | Manual CS setting, for MTP\_TOP | Level(CLK\_REG domain) |
| MANU | O | 1 | MANU mode for ICELL test | Level(CLK\_REG domain) |
| SRL | O | 1 | For MTP\_TOP | Level(CLK\_REG domain) |
| MRGN | O | 1 | For MTP\_TOP | Level(CLK\_REG domain) |
| RD\_MANU | O | 1 | Manual RD setting, for MTP\_TOP | Level(CLK\_REG domain) |
| WR\_MANU | O | 1 | Manual WR setting, for MTP\_TOP | Level(CLK\_REG domain) |
| HVEN | O | 1 | For MTP\_TOP | Level(CLK\_REG domain) |
| wr\_nvm | O | 1 | Write MTP | Level(CLK\_REG domain) |
| DIN | O | 1 | DATA from MTP | Level(CLK\_REG domain) |
| GainErr\_code\_comm | O | 12 | TRIM\_ADC | Level(CLK\_REG domain) |
| ADC\_VCM\_GAIN\_EN | O | 1 | TRIM\_ADC | Level(CLK\_REG domain) |
| ADC\_TEMP\_COM\_EN | O | 1 | TRIM\_ADC | Level(CLK\_REG domain) |
| trim\_logic\_en | O | 1 | For TRIM\_LOGIC | Level(CLK\_REG domain) |
| GainErr\_code\_Cell1~18 | O | 7\*18 | TRIM\_ADC | Level(CLK\_REG domain) |
| GainErr\_code\_GPIO | O | 12 | TRIM\_ADC | Level(CLK\_REG domain) |
| GainErr\_code\_VPTAT | O | 12 | TRIM\_ADC | Level(CLK\_REG domain) |
| ADC\_TEMP\_code\_H | O | 8 | TRIM\_ADC | Level(CLK\_REG domain) |
| ADC\_TEMP\_code\_L | O | 8 | TRIM\_ADC | Level(CLK\_REG domain) |
| Vos\_code\_comm | O | 8 | TRIM\_ADC | Level(CLK\_REG domain) |
| Vos\_code\_cell1~18 | O | 6\*18 | TRIM\_ADC | Level(CLK\_REG domain) |
| Vos\_code\_GPIO | O | 8 | TRIM\_ADC | Level(CLK\_REG domain) |
| ALG\_GAINERR1 | O | 8 | TRIM\_ADC | Level(CLK\_REG domain) |
| Vcm\_code\_comm | O | 7 | TRIM\_ADC | Level(CLK\_REG domain) |
| GainErr\_code\_B\_comm | O | 12 | TRIM\_ADC | Level(CLK\_REG domain) |
| AUX\_VCM\_GAIN\_EN | O | 1 | TRIM\_ADC | Level(CLK\_REG domain) |
| AUX\_TEMP\_COM\_EN | O | 1 | TRIM\_ADC | Level(CLK\_REG domain) |
| GainErr\_code\_B\_Cell1~18 | O | 7\*18 | TRIM\_ADC | Level(CLK\_REG domain) |
| GainErr\_code\_B\_GPIO | O | 12 | TRIM\_ADC | Level(CLK\_REG domain) |
| GainErr\_code\_B\_VPTAT | O | 12 | TRIM\_ADC | Level(CLK\_REG domain) |
| AUX\_TEMP\_code\_B\_H | O | 8 | TRIM\_ADC | Level(CLK\_REG domain) |
| AUX\_TEMP\_code\_B\_L | O | 8 | TRIM\_ADC | Level(CLK\_REG domain) |
| Vos\_code\_B\_comm | O | 8 | TRIM\_ADC | Level(CLK\_REG domain) |
| Vos\_code\_B\_cell1~18 | O | 6\*18 | TRIM\_ADC | Level(CLK\_REG domain) |
| Vos\_code\_B\_GPIO | O | 8 | TRIM\_ADC | Level(CLK\_REG domain) |
| ALG\_GAINERR2 | O | 8 | TRIM\_ADC | Level(CLK\_REG domain) |
| Vcm\_code\_B\_comm | O | 7 | TRIM\_ADC | Level(CLK\_REG domain) |
| Reg1000~107F | O | 8\*128 | For MTP\_CRC | Level(CLK\_REG domain) |
| CONF\_CRC | O | 16 | For CONF\_CRC check | Level(CLK\_REG domain) |
| TM\_REG1~10 | O | 8\*10 | Regs for test mode | Level(CLK\_REG domain) |
| CELL\_OV\_THRESH\_REG | O | 7 | For OVUV\_OTUT\_CMP | Level(CLK\_REG domain) |
| CELL\_UV\_THRESH\_REG | O | 7 | For OVUV\_OTUT\_CMP | Level(CLK\_REG domain) |
| GPIO\_OTUT\_THRESH\_SEL\_REG | O | 12 | For OVUV\_OTUT\_CMP | Level(CLK\_REG domain) |
| GPIO\_OT\_PACK\_THRESH\_REG | O | 5 | For OVUV\_OTUT\_CMP | Level(CLK\_REG domain) |
| GPIO\_OT\_PCB\_THRESH\_REG | O | 5 | For OVUV\_OTUT\_CMP | Level(CLK\_REG domain) |
| GPIO\_UT\_PACK\_THRESH\_REG | O | 3 | For OVUV\_OTUT\_CMP | Level(CLK\_REG domain) |
| GPIO\_UT\_PCB\_THRESH\_REG | O | 3 | For OVUV\_OTUT\_CMP | Level(CLK\_REG domain) |
| CB\_OT\_PACK\_THRESH\_REG | O | 5 | For OVUV\_OTUT\_CMP | Level(CLK\_REG domain) |
| CB\_OT\_PCB\_THRESH\_REG | O | 5 | For OVUV\_OTUT\_CMP | Level(CLK\_REG domain) |
| TWARN\_THRESH\_REG | O | 3 | For OVUV\_OTUT\_CMP | Level(CLK\_REG domain) |
| GPIO\_PUPD | O | 12 | For analog | Level(CLK\_REG domain) |
| GPIO\_PUPD\_EN | O | 12 | For analog | Level(CLK\_REG domain) |
| GPIO\_AS\_IN\_EN | O | 12 | For analog | Level(CLK\_REG domain) |
| WEAK\_PUPD\_EN | O | 12 | For analog | Level(CLK\_REG domain) |
| WEAK\_PUPD | O | 12 | For analog | Level(CLK\_REG domain) |
| CS\_EN\_MANU | O | 19 | For analog | Level(CLK\_REG domain) |
| OVUV\_OTUT\_EN\_REG | O | 1 | For OVUV\_OTUT\_CMP | Level(CLK\_REG domain) |
| OVUV\_DEGL\_REG | O | 5 | For OVUV\_OTUT\_CMP | Level(CLK\_REG domain) |
| MON\_WAKE\_PERIOD\_REG | O | 6 | For CYC\_WAKE | Level(CLK\_REG domain) |
| CELL\_GAP\_THRESH\_REG | O | 5 | For GAP\_CMP | Level(CLK\_REG domain) |
| OTH\_GAP\_THRESH\_REG | O | 3 | For GAP\_CMP | Level(CLK\_REG domain) |
| CELL\_CAP\_DEGL\_REG | O | 5 | For GAP\_CMP | Level(CLK\_REG domain) |
| CH\_TOP\_STL\_REG | O | 4 | ADC\_SETTING | Level(CLK\_REG domain) |
| CH\_STL\_REG | O | 4 | ADC\_SETTING | Level(CLK\_REG domain) |
| CH\_BOT\_STL\_REG | O | 4 | ADC\_SETTING | Level(CLK\_REG domain) |
| ADC\_MODE\_REG | O | 2 | ADC\_SETTING | Level(CLK\_REG domain) |
| ADC\_CLK\_SET\_REG | O | 2 | ADC\_SETTING | Level(CLK\_REG domain) |
| ADC\_CHP\_EN\_REG | O | 1 | ADC\_SETTING | Level(CLK\_REG domain) |
| CH\_DT\_REG | O | 2 | ADC\_SETTING | Level(CLK\_REG domain) |
| STACK\_RESPONSE | O | 6 | Determine byte interval time in response frames | Level(CLK\_REG domain) |
| SOFT\_RSTB\_REG | O | 1 | 0:Soft reset 1:not reset | Level(CLK\_REG domain) |
| ADD\_W\_EN | O | 1 | Indicating address identify frame | Level(CLK\_REG domain) |
| DEV\_NUM | O | 7 | Device number from current device to the top device | Level(CLK\_REG domain) |
| I2C\_MAC\_EN | O | 1 | I2c master enable | Level(CLK\_REG domain) |
| i2c\_sr | O | 1 | For I2C\_MAS to generate a restart condition | Level(CLK\_REG domain) |
| i2c\_stop | O | 1 | For I2C\_MAS to generate a stop condition | Level(CLK\_REG domain) |
| i2c\_tx | O | 1 | For I2C\_MAS to send data | Level(CLK\_REG domain) |
| i2c\_tx | O | 1 | For I2C\_MAS to receive data | Level(CLK\_REG domain) |
| i2c\_tx\_data | O | 8 | For I2C\_MAS prepare data to send | Level(CLK\_REG domain) |
| C\_OW\_DET\_GO | O | 1 | For C\_OW\_CTRL | Cleared by clr\_C\_OW\_DET\_GO |
| C\_OW\_TDIS\_REG | O | 4 | For C\_OW\_CTRL | Level(CLK\_REG domain) |
| BIST\_GO | O | 1 | For CMP\_BIST\_CTRL | Cleared by clr\_BIST\_GO |
| CB\_GO | O | 1 | To start CB\_CTRL and load CB\_Setting\_REG | Cleared by clr\_CB\_GO |
| CB\_MANU\_PAUSE | O | 1 | pause CB\_CTRL | Level(CLK\_REG domain) |
| CB\_SETTLE | O | 5 | For RECLK\_COMP: CB settle time | Level(CLK\_REG domain) |
| CB\_GO | O | 1 | To start CB\_CTRL and load CB\_Setting\_REG | Cleared by clr\_CB\_GO |
| CB\_PERIOD\_REG | O | 3 | in automatic mode, indicate odd/even covert time, 5s-30min, 8steps refer to competitor spec | Level(CLK\_REG domain) |
| CB\_DUTY\_REG | O | 3 | duty of internal PWM, shall cover 12.5%-100%, 8steps  period is 200ms | Level(CLK\_REG domain) |
| CB\_MANUAL\_REG | O | 1 | CB mode select, 1: manual , 0:automatic | Level(CLK\_REG domain) |
| JOT\_EN\_REG | O | 1 | enable JOT to pause CB\_CTRL | Level(CLK\_REG domain) |
| ADC\_PAUSE\_EN\_REG | O | 1 | enable ADC\_EN to pause CB\_CTRL | Level(CLK\_REG domain) |
| GPIO\_CBOT\_EN\_REG | O | 1 | enable GPIO\_CBOT to pause CB\_CTRL | Level(CLK\_REG domain) |
| FLT\_STOP\_EN\_REG | O | N/A | enable FLT\_WAKE to stop CB\_CTRL, stop CB\_EN, and wait for another CB\_GO when FLT\_WAKE is "L" | Level(CLK\_REG domain) |
| CBFET\_EN\_REG | O | 18 | CB\_EN Input of 18 channels | Level(CLK\_REG domain) |
| CB\_TWARN\_THRESH\_REG | O | 4 | after CB\_GO,CB\_CTRL output CB\_TWARN\_THRESH to analog, don’t need other operation | Level(CLK\_REG domain) |
| CB\_UNIT\_REG | O | 18 | unit of CB\_TO\_THRESH\_REG | Level(CLK\_REG domain) |
| CB\_TO\_THRESH\_REG1-18 | O | 10\*18 | CB threshold time about each channel | Level(CLK\_REG domain) |
| X\_FLT\_SUM\_REG | O | 1\*5 | Fault sum regs for FLT\_LOGIC | Level(CLK\_REG domain) |
| X\_FLT\_REG | O | 1\*143 | Fault bit regs for FLT\_LOGIC | Level(CLK\_REG domain) |
| TM\_KEY | O | 8 | For TM\_KEY\_CHECK | Level(CLK\_REG domain) |
| CLK\_REG\_SC | I | 1 | Scan-mux result of 8MHz clock from CLK\_32M | 8MHz |
| resetb\_CLK | I | 1 | Asynchronous reset signal(synchronously released) |  |
| CLK\_8M\_256K\_SC | I | 1 | When CLK\_32M is OK, use CLK\_8M; otherwise use CLK\_256K; after scanmux | 8MHz or 256kHz |
| resetb\_CLK\_OUT | I | 1 | Asynchronous reset signal by CLK\_OUT(synchronously released) |  |
| CLK\_I2C\_SC | I | 1 | For I2C\_CTRL(change CLK\_REG to CLK\_I2C domain) |  |
| resetb\_SR\_CLK\_I2C | I | 1 | Asynchronous reset signal (synchronously released) with soft reset for I2C\_CTRL |  |
| SOFT\_RSTB\_32M | I | 1 | Soft reset from RSTGEN | Level(CLK\_REG domain) |
| SOFT\_RSTB | I | 1 | CLK\_SLOW synced SOFT\_RSTB\_REG from u\_SOFT\_RSTB\_sync | Level(CLK\_SLOW domain) |
| pulse\_2ms | I | 1 |  |  |
| ini\_addr | I | 16 | Initial address for writing | Level(CLK\_REG domain) |
| reg\_addr | I | 16 | Register address for reading | Level(CLK\_REG domain) |
| neg\_rx\_en | I | 1 |  |  |
| bytes | I | 4 | Bytes number for successive writing | Level(CLK\_REG domain) |
| wr\_data | I | 128 | Received data buffer | Level(CLK\_REG domain) |
| wr\_update | I | 1 | Write update time | Level(CLK\_REG domain) |
| SPI\_EN | I | 1 | 1:used as bridge 0:used as AFE | Async(constant) |
| rx\_en\_n | I | 1 | Receive data from N port | Level(CLK\_32M domain) |
| rx\_en\_s | I | 1 | Receive data from S port | Level(CLK\_32M domain) |
| dev\_addr\_dlv | I | 1 | Device address identify delivery | Level(CLK\_REG domain) |
| dev\_addr0 | I | 8 | Device address | Level(CLK\_REG domain) |
| state\_tx\_bps | I | 1 | tx\_state is STATE\_BYPASS | Level(CLK\_REG domain) |
| state\_rx\_bps | I | 1 | state is STATE\_BYPASS | Level(CLK\_REG domain) |
| state\_rx\_init | I | 1 | state is STATE\_INIT | Level(CLK\_REG domain) |
| state\_rx\_cur\_addr | I | 1 | state is STATE\_CUR\_ADR | Level(CLK\_REG domain) |
| 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) |
| TWARN | O | 1 | Fault input from u\_OVUV\_OTUT\_CMP | Level(CLK\_SLOW domain) |
| load\_done | O | 1 | MTP load done | Level(CLK\_REG domain) |
| DEV\_ADD\_LATCH | O | 7 | Latched device address | constant |
| DEV\_NUM\_LATCH | O | 7 | Latched device number in the rest of the daisy chain | constant |
| DIR\_SEL\_LATCH | O | 1 | Latched direction seletion  1:data from N to S 0:data from S to N | constant |
| TOP\_DEV\_LATCH | O | 1 | Latched top device information | constant |
| clr\_ADC\_GO | O | 1 | Signal to clear 4 kinds of ADC\_GO |  |
| clr\_MON\_WAKE\_GO | O | 1 | Signal to clear MON\_WAKE\_GO | 1 CLK\_SLOW |
| clr\_C\_OW\_DET\_GO | O | 1 | Signal to clear C\_OW\_DET\_GO | 1 CLK\_SLOW |
| RR\_COUNTER | I | 16 | Round-robin number, RR\_COUNTER is frozen when FREEZE\_DLY is detected, is cleared by ADC\_GO\_DLY is high . |  |
| pos\_HBFAST | I | 1 | HB too fast condition is detected | 1 CLK\_256K |
| pos\_HBTO | I | 1 | HB too slow(timeout) condition is detected | 1 CLK\_256K |
| clr\_TO\_SLEEP | I | 1 | Signal to clear TO\_SLEEP | 2~3 CLK\_SLOW |
| clr\_WAKE\_TONE\_GEN | I | 1 | Pulse for module u\_COMM\_REG to clear WAKE\_TONE\_GEN | 1 CLK\_256K |
| clr\_STA\_TONE\_GEN | I | 1 | Pulse for module u\_COMM\_REG to clear STA\_TONE\_GEN | 1 CLK\_256K |
| clr\_SD\_TONE\_GEN | I | 1 | Pulse for module u\_COMM\_REG to clear SD\_TONE\_GEN | 1 CLK\_256K |
| MISS | I | 1 |  |  |
| ORDER | I | 1 |  |  |
| SYNCT | I | 1 |  |  |
| SYNCD | I | 1 |  |  |
| BIT | I | 1 |  |  |
| RR | I | 1 |  |  |
| SOFB | I | 1 |  |  |
| IERR | I | 1 |  |  |
| TXDIS | I | 1 |  |  |
| SOF | I | 1 |  |  |
| UNEXP\_C | I | 1 |  |  |
| UNEXP\_R | I | 1 |  |  |
| CRC | I | 1 |  |  |
| CONFL | I | 1 |  |  |
| pos\_TBYTE\_FAST | I | 1 | fault flag: receiving data is too fast | 4 CLK\_32M |
| pos\_TBYTE\_TO | I | 1 | fault flag: receiving data is too slow | 4 CLK\_32M |
| FCOMM\_FLT\_IN | I | 1 | Communication fault received from last device | Level(CLK\_REG domain) |
| i2c\_rx\_data | I | 8 | I2c received data | Level(CLK\_I2C domain) |
| i2c\_ack\_out | I | 1 | I2c acknowledge bit | Level(CLK\_REG domain) |
| DOUT | I | 8 | Data loaded from u\_MTP | async |
| READ | I | 1 | READ pulse from u\_MTP\_TOP | Level(CLK\_MTP domain) |
| idle\_2h | I | 1 | Mark the time for shadow registers to update with DOUT | 8 CLK\_MTP |
| clrb\_wr\_mtp | I | 1 | Signal to clear WR\_MTP | Level(CLK\_MTP domain) |
| ADR | I | 7 | Address for MTP\_REG | Level(CLK\_MTP domain) |
| WR\_MTP | I | 1 | MTP writing is on-going |  |
| LCTO\_SD\_LATCH | I | 1 | Analog latched LCTO\_SD fault | Async(constant) |
| CLK\_256K\_OKB\_LATCH | I | 1 | Analog latched CLK\_256K not OK fault | Async(constant) |
| VDD\_OKB\_LATCH | I | 1 | Analog latched VDD not OK fault | Async(constant) |
| VAA\_OKB\_LATCH | I | 1 | Analog latched VAA not OK fault | Async(constant) |
| VDD\_OV\_LATCH | I | 1 | Analog latched VDD OV fault | Async(constant) |
| VDD\_UV\_LATCH | I | 1 | Analog latched VDD UV fault | Async(constant) |
| GPIO\_HL | I | 12 | From analog, read-only | Async |
| CELL\_UV | I | 18 | Cell UV fault | Level(CLK\_SLOW domain) |
| CELL\_OV | I | 18 | Cell OV fault | Level(CLK\_SLOW domain) |
| GPIO\_UT | I | 12 | GPIO UT fault | Level(CLK\_SLOW domain) |
| GPIO\_OT | I | 12 | GPIO OT fault | Level(CLK\_SLOW domain) |
| VAA\_OV | I | 1 | VAA OV fault from analog | Async |
| VAA\_UV | I | 1 | VAA UV fault from analog | Async |
| VDD\_OV | I | 1 | VDD OV fault from analog | Async |
| VDD\_UV | I | 1 | VDD UV fault from analog | Async |
| CP\_OV | I | 1 | CP OV fault from analog | Async |
| CP\_UV | I | 1 | CP UV fault from analog | Async |
| AGND\_OW | I | 1 | AGND OW fault from analog | Async |
| SET\_SCTO | I | 1 | Short timeout | Level(CLK\_REG domain) |
| SET\_LCTO | I | 1 | Long timeout | Level(CLK\_REG domain) |
| TRIM\_EN | I | 1 | Trim enable | Level(CLK\_REG domain) |
| TMREG\_EN | I | 1 | Test mode reg enable | Level(CLK\_REG domain) |
| tx\_add\_reg\_addr | I | 1 | tx register address adds bytes end pulse | 1 CLK\_32M |
| tx\_state\_addr | I | 1 | tx\_state is STATE\_ADDR | Level(CLK\_REG domain) |
| ADC\_CH1~18 | I | 16\*18 | Trimmed ADC\_CH data | Level(CLK\_REG domain) |
| ADC\_GPIO1~12 | I | 16\*12 | Trimmed ADC\_GPIO data | Level(CLK\_REG domain) |
| ADC\_VPTAT | I | 1 | Trimmed ADC\_VPTAT data | Level(CLK\_REG domain) |
| ADC\_VBG | I | 1 | Trimmed ADC\_VBG data | Level(CLK\_REG domain) |
| ADC\_VBG2 | I | 1 | Trimmed ADC\_VBG2 data | Level(CLK\_REG domain) |
| ADC\_LPF\_CH1~18 | I | 16\*18 | Trimmed ADC\_CH data after LPF | Level(CLK\_REG domain) |
| AUX\_CH1~18 | I | 16\*18 | Trimmed AUX\_CH data | Level(CLK\_REG domain) |
| AUX\_GPIO1~12 | I | 16\*12 | Trimmed AUX\_GPIO data | Level(CLK\_REG domain) |
| AUX\_VPTAT | I | 1 | Trimmed AUX\_VPTAT data | Level(CLK\_REG domain) |
| AUX\_VBG | I | 1 | Trimmed AUX\_VBG data | Level(CLK\_REG domain) |
| AUX\_VBG2 | I | 1 | Trimmed AUX\_VBG2 data | Level(CLK\_REG domain) |
| AUX\_LPF\_CH1~18 | I | 16\*18 | Trimmed AUX\_CH data after LPF | Level(CLK\_REG domain) |
| CB\_ODD\_CNT | I | 16 | odd/even is same in manual mode, is different in automatic mode, counter of odd group | Level(CLK\_CB domain) |
| CB\_EVEN\_CNT | I | 16 | counter of even group | Level(CLK\_CB domain) |
| CB\_EN\_FULL\_DUTY | I | 18 | output CB\_CH\_EN with full duty | Level(CLK\_CB domain) |
| FR\_CNT | I | 16 | Frame counter | Level(CLK\_REG domain) |
| CMP\_FLT | I | 16 | Compare fault from u\_CMP\_BIST\_CTRL | Level(CLK\_REG domain) |
| RX\_FIFO\_OF | I | 1 | TX FIFO overflow, when TX FIFO is full and daisy chain still write to TX FIFO | 1 CLK\_REG\_SC |
| TX\_FIFO\_OF | I | 1 | RXFIFO overflow, when RXFIFO is full and host still write to RXFIFO | 1 CLK\_REG\_SC |
| TX\_FIFO\_UF | I | 1 | TXFIFO underflow, when FIFO is empty and read from FIFO | 1 CLK\_REG\_SC |
| CONF\_REG\_CRC\_FLT | I | 1 | Configure registers CRC fault | Level(CLK\_SLOW domain) |
| MTP\_REG\_CRC\_FLT | I | 1 | MTP registers CRC fault | Level(CLK\_SLOW domain) |
| FLT\_TONE\_DET | I | 1 | Fault tone detected by analog | async |
| clr\_BIST\_GO | I | 1 | Signal to reset BIST\_GO |  |
| clr\_CB\_GO | I | 1 | output to clear CB\_GO | >1 CLK\_CB\_SC |
| CELL\_GAP\_FLT | I | 18 | Over gap threshold flag of CELL1-CELL18 | Level(CLK\_SLOW domain) |
| OTH\_GAP\_FLT | I | 12 | Over gap threshold flag of other channel | Level(CLK\_SLOW domain) |
| C\_OW\_FLT | I | 19 | Over range flags | Level(CLK\_REG domain) |
| ECED | I | 1 | MTP data corrected flag by ECC algorithm | aysnc |
| CB\_CONF\_FLT | I | 1 | >2 consecutive channels turn on in CBFET\_EN | 1 CLK\_CB\_SC |
| BUSY | I | 1 | MTP write on-going flag by u\_MTP | async |
| neg\_response | I | 1 | Negative edge of response | 1 CLK\_REG |
| SCAN\_CLK | I | 1 | CLK in scan mode | Up to 10MHz |
| SCAN\_MODE | I | 1 | SCAN\_MODE | Level(CLK\_REG domain) |
| SCAN\_RSTB | I | 1 | Async resetb in scan mode | Level(CLK\_32M domain) |
| Scan\_enable | I | 1 | 1:shift mode 0:capture mode or function mode | 4 CLK\_32M |

#### Clock Domain

The clock for COMM\_REG is CLK\_REG\_SC and CLK\_8M\_256K\_SC.

For u\_FLT\_BIT and u\_FLT\_SUM, both CLK\_REG\_SC and CLK\_8M\_256K\_SC are used.

For other sub-modules, only CLK\_REG\_SC is used.

### COMM\_REG function description

#### Control bits

TO\_SD bit is defined in reg2002[1]. TO\_SD is synchronized in module u\_TO\_SD\_sync by CLK\_SLOW, and module u\_TO\_SD\_sync output D2A\_TO\_SD for analog. Analog shutdown digital part, thus TO\_SD and D2A\_TO\_SD are both pulses.(HWR001\_COMM\_REG)

TO\_SLEEP bit is defined in reg2002[0]. TO\_SLEEP is synchronized in module u\_TO\_SLEEP\_sync by CLK\_SLOW, and module u\_TO\_SLEEP\_sync output D2A\_TO\_SLEEP for analog, and then generate clr\_TO\_SLEEP for u\_COMM\_REG to clear TO\_SLEEP, thus TO\_SLEEP is a pulse.(HWR002\_COMM\_REG)

PROG\_SCTO[2:0] is defined in reg0004[2:0]. (HWR003\_COMM\_REG)

PROG\_LCTO[2:0] is defined in reg0004[5:3]. (HWR004\_COMM\_REG)

LCTO\_SEL[1:0] is defined in reg0004[7:6]. (HWR005\_COMM\_REG)

ADC\_SGLE\_GO is defined in reg2004[1], ADC\_CNTI\_GO is defined in reg2004[0]. Both ADC\_SGLE\_GO and ADC\_CNTI\_GO are cleared by clr\_ADC\_GO. (HWR011\_COMM\_REG)

MON\_WAKE\_GO is defined in reg2004[7]. MON\_WAKE\_GO is cleared by clr\_MON\_WAKE\_GO. (HWR012\_COMM\_REG)

MON\_EN\_REG is defined in reg000D[0]. (HWR050\_COMM\_REG)

FREEZE is defined in reg2004[2]. (HWR013\_COMM\_REG)

DLPF\_FC[2:0] is defined in reg0008[7:5]. (HWR015\_COMM\_REG)

STACK\_RESPONSE[5:0] is defined in reg0003[5:0]. (HWR019\_COMM\_REG)

COMN\_TX\_DIS is defined in reg0003[7], COMN\_TX\_DIS is defined in reg0003[6]. (HWR054\_COMM\_REG)

SD\_TONE\_GEN is defined in reg2002[2], use SPI\_DIR as SD\_TONE\_DIR. SPI\_DIR is defined in reg2003[0]. (HWR051\_COMM\_REG) SD\_TONE\_GEN is cleared by clr\_SD\_TONE\_GEN. (HWR020\_COMM\_REG)

WAKE\_TONE\_GEN is defined in reg2002[4], WAKE\_TONE\_GEN reset to 0 when clr\_WAKE\_TONE\_GEN is high. (HWR052\_COMM\_REG)

STA\_TONE\_GEN is defined in reg2002[3], STA\_TONE\_GEN reset to 0 when clr\_STA\_TONE\_GEN is high. (HWR053\_COMM\_REG)

All fault bits have corresponding mask bits. Mask bits are defined in reg0100~0117. (HWR036\_COMM\_REG)

MTP\_CTRL\_BITS are defined in reg1FFB~1FFF. They are writable only when TRIM\_EN high. (HWR043\_COMM\_REG)

I2C\_CTRL bits include i2c\_stop, i2c\_rx, i2c\_sr, i2c\_tx. These 4 bits are defined in reg2200. I2C\_MAS\_EN is defined in reg0002[0]. I2C\_CTRL include i2c\_tx\_data[7:0], which is defined in reg2201. I2c\_rd\_data[7:0] is readable from address 0x2202. I2c\_ack\_out is readable from bit 4 of address 2200. (HWR047\_COMM\_REG)

SOFT\_RSTB\_REG is defined in reg2002[7]. SOFT\_RSTB\_REG is default high. When written 0 via COMM\_CTRL, SOFT\_RSTB\_REG is low. SOFT\_RSTB\_REG reset to high when clr\_SOFT\_RSTB high. (HWR049\_COMM\_REG)

#### 2 Trim bits

TRIM\_ADC bits are defined in reg1000~reg105F. (HWR008\_COMM\_REG)

TRIM\_ANA bits are defined in reg1060~reg107A. (HWR042\_COMM\_REG)

X-Y and lot information are readable from address 0x107C~107D. These bits are writable only when TRIM\_EN high. (HWR048\_COMM\_REG)

#### Setting bits

ADC\_SETTING\_REG bits are defined in reg0008~reg000C. (HWR010\_COMM\_REG)

MON\_WAKE\_PERIOD\_REG[5:0] is defined in reg000D[7:2]. (HWR031\_COMM\_REG)

CELL\_OV\_THRESH\_REG[6:0] is defined in reg000F[6:0], CELL\_UV\_THRESH\_REG[6:0] is defined in reg0010[6:0]. (HWR041\_COMM\_REG)

GPIO\_OT\_PACK\_THRESH\_REG[4:0] is defined in reg0011[7:3], GPIO\_UT\_PACK\_THRESH\_REG[2:0] is defined in reg0011[2:0], GPIO\_OT\_PCB\_THRESH\_REG[4:0] is defined in reg0012[7:3], GPIO\_UT\_PCB\_THRESH\_REG[2:0] is defined in reg0012[2:0]. (HWR044\_COMM\_REG)

GPIO\_PUPD[11:0] is defined in reg202D~202E, GPIO\_PUPD\_EN[11:0] is defined in reg0019~001A, GPIO\_AS\_IN\_EN[11:0] is defined in reg001B~001C. (HWR045\_COMM\_REG)

GPIO\_HL[11:0] is readable from address 0x5002~5003. (HWR046\_COMM\_REG)

OVUV\_DEGL\_REG[4:0] is defined in reg000E[4:0]. (HWR056\_COMM\_REG)

OVUV\_OTUT\_EN\_REG is defined in reg000E[7]. (HWR057\_COMM\_REG)

#### ADC results

ADC\_DATA\_LPF are readable from address 0x4000~0x4023. (HWR009\_COMM\_REG)

CELL\_ADC\_DATA are readable from address 0x4048~0x406B, OTH\_ADC\_DATA are readable from address 0x4090~0x40A7, 0x40C0~0x40C5. (HWR014\_COMM\_REG)

RR\_COUNTER is readable from address 0x3FFE~0x3FFF. (HWR016\_COMM\_REG, HWR018\_COMM\_REG)

#### COMM\_DIG\_SETTING

TOP\_DEV is defined in reg0000[7], DIR\_SEL is defined in reg2002[6], DEV\_ADD[6:0] is defined in reg0000[6:0], DEV\_NUM[6:0] is defined in reg0001[6:0]. (HWR021\_COMM\_REG, HWR024\_COMM\_REG)

Default values of COMM\_DIG\_SETTING registers are from XXX\_LATCH. DEV\_ADD cannot be changed when ADD\_W\_EN is low. (HWR023\_COMM\_REG)

ADD\_W\_EN is defined in reg2003[1]. ADD\_W\_EN is high when address identify starts. ADD\_W\_EN is cleared by writing reg2003[1] to 0.( HWR021\_COMM\_REG, HWR022\_COMM\_REG)

DEV\_NUM[6:0] indicates the device number from current device to the top device. DEV\_NUM[6:0] adds by step 1 when receiving device response when ADD\_W\_EN high(in address identify response stage). DEV\_NUM[6:0] reset to 0 when address identify starts. (HWR024\_COMM\_REG)

#### CB\_CTRL\_SETTING\_REG

CBFET\_EN\_REG1~18 are defined in reg2006~2008. CB\_GO is defined in reg2005[0]. CB\_GO is high when written high, is low when clr\_CB\_GO high. CB\_MANUAL\_REG is defined in reg2005[2]. (HWR025\_COMM\_REG)

CB\_TO\_THRESH\_REG1~18[9:0] and CB\_UNIT\_REG[17:0] are defined in reg2009~202C. (HWR026\_COMM\_REG)

CB\_TWARN\_THRESH\_REG[3:0] is defined in reg0017[7:4]. (HWR027\_COMM\_REG)

JOT\_EN\_REG is defined in reg0017[1]. (HWR028\_COMM\_REG)

GPIO\_CBOT\_EN\_REG is defined in reg0017[3]. (HWR029\_COMM\_REG)

CB\_OT\_PACK\_THRSH\_REG[4:0] is defined in reg0015[7:3], CB\_OT\_PCB\_THRSH\_REG[4:0] is defined in reg0016[7:3]. GPIO\_OTUT\_THRESH\_SEL\_REG[11:0] is defined in reg0013~0014. (HWR030\_COMM\_REG)

CB\_MANU\_PAUSE is defined in reg2005[1]. (HWR033\_COMM\_REG)

ADC\_PAUSE\_EN\_REG is defined in reg0017[2]. (HWR034\_COMM\_REG)

FLT\_STOP\_EN\_REG is defined in reg0017[0]. (HWR035\_COMM\_REG)

CB\_PERIOD\_REG[2:0] is defined in reg0016[2:0]. (HWR037\_COMM\_REG)

CB\_DUTY\_REG[2:0] is defined in reg0015[2:0]. (HWR038\_COMM\_REG)

CB\_ODD\_CNT[15:0] and CB\_EVEN\_CNT[15:0] are readable from address 0x5007~500A. (HWR039\_COMM\_REG)

CB\_CH\_EN\_FULL\_DUTY[17:0] are readable from address 0x5004~5006. (HWR040\_COMM\_REG)

CB\_SETTLE[4:0] is defined in reg0002[7:3]. (HWR055\_COMM\_REG)

#### Fault regs

Fault regs include FLT\_BIT registers and FLT\_SUM registers. FLT\_BIT registers are located in addresses 0x5101~5117, FLT\_SUM registers are located in address 0x5100.

SCTO flag bit is defined in reg5116[6], it is set by SET\_SCTO. (HWR001\_FLT\_REG)

LCTO flag bit is defined in reg5116[5], when it is set by SET\_LCTO or by LCTO\_LATCH. (HWR002\_FLT\_REG)

LCTO\_SD, VAA\_OKB, VDD\_OKB, CLK\_256K\_OKB, VDD\_OV, VDD\_UV flag bits are defined in address 0x5115. They support being only set by corresponding input XXX\_LATCH. FLT\_BIT output RST\_XXX\_LATCH when register bits is written to 1. (HWR003\_FLT\_REG)

FLT\_TONE\_DET\_REG is recorded in reg5117[0]. FLT\_TONE\_DET\_REG shall be high when FLT\_TONE\_DET is high. (HWR004\_FLT\_REG)

GPIO\_OTUT (including GPIO\_OT[11:0] and GPIO\_UT[11:0]) are recorded in FLT\_REG. OT\_FLT[11:0] recording GPIO\_OT faults, are in address 0x510F~5110, UT\_FLT[11:0] recording GPIO\_UT fault, are in address 0x5111~5112. (HWR005\_FLT\_REG)

CELL\_OVUV (including CELL\_OV[17:0] and CELL\_UV[17:0]) are recorded in FLT\_REG. OV\_FLT[17:0] recording CELL\_OV faults, are in address 0x5104~5106, UV\_FLT[17:0] recording CELL\_UV faults, are in address 0x5107~5109. (HWR006\_FLT\_REG)

Every fault bit has its corresponding mask bit. All fault bits can not be set high when corresponding mask bit(from reg0100~0117) is 1. (HWR009\_FLT\_REG)

COW[18:0] are fault flag from u\_C\_OW\_CTRL, are in address 0x5101~5103. (HWR010\_FLT\_REG)

All fault bits can not be written to 1 via COMM\_CTRL. When corresponding mask bit is 0, all fault bits canl be cleared to 0 when written 0 via COMM\_CTRL. (HWR011\_FLT\_REG)

CB\_CONF\_FLT is recorded in reg5116[7]. (HWR012\_FLT\_REG)

RX\_FIFO\_OF\_REG bit set by RX\_FIFO\_OF is recorded in reg5116[3], TX\_FIFO\_OF\_REG bit set by TX\_FIFO\_OF is recorded in reg5116[2]. (HWR013\_FLT\_REG)

TX\_FIFO\_UF\_REG bit set by TX\_FIFO\_UF is recorded in reg5116[1]. (HWR014\_FLT\_REG)