34.4 Registers

This section describes the internal registers of the I2C module. All register names described in this section are also referenced in other parts of the User's Manual by the module name prefix "I2Cm_". For an overview of all internal module registers, see **Section 34.4**.

In the following, the registers of the I2C module are listed. First of all, some explanation on the access conditions is given.

Special I2C Register Access Condition

Besides the general register protection, the I2C module has two main modes that must be considered when programming the peripheral:

- **Configuration Mode:** In this mode the peripheral can be prepared for transmission and reception via the configuration registers, which are only writable in this mode. The peripheral is in the configuration mode when bit RUN is set to 0.
- **Run Mode:** In this mode the peripheral is ready to transmit or receive data. Its configuration registers are locked for write access which will generate a bus error. The peripheral is in the run mode when bit **RUN** is set to 1.

I2C Registers Overview

There are the following blocks of registers (see **Figure 354**):

- Bus Peripheral Interface Registers
- Global Module Control Registers
- FIFO Registers
- Basic Interrupt Registers
- Error Interrupt Source Registers
- Protocol Interrupt Source Registers

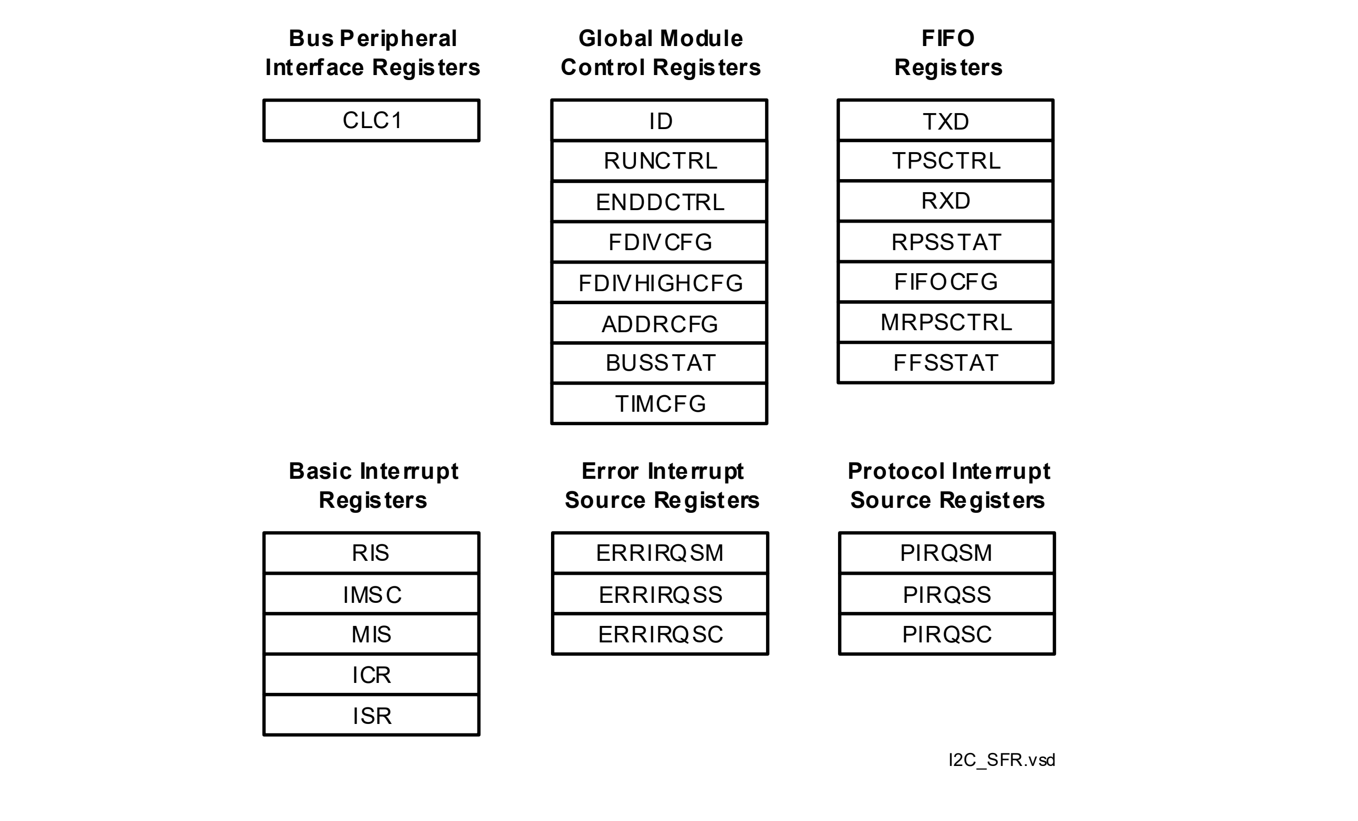

Figure 354 I2C Module Registers

The registers overview in **Table 303** shows the internal register names of the module instances, the offset addresses and the links to the names used in this specification.

Table 303 Register Overview - I2C (ascending Offset Address)

Short Name	Long Name	Offset Address	Access Mode		Reset	Page Number
			Read	Write		
CLC1	Clock Control 1 Register	00000_H	U,SV	U,SV,P	Application Reset	45
ID	Module Identification Register	00008_H	U,SV	BE	Application Reset	59
RUNCTRL	RUN Control Register	00010_H	U,SV	U,SV,P	Application Reset	59
ENDDCTRL	End Data Control Register	00014_H	U,SV	U,SV,P	Application Reset	60
FDIVCFG	Fractional Divider Configuration Register	00018_H	U,SV	U,SV,P	Application Reset	63
FDIVHIGHCFG	Fractional Divider High-speed Mode Configuration Register	$0001C_H$	U,SV	U,SV,P	Application Reset	64
ADDRCFG	Address Configuration Register	00020_H	U,SV	U,SV,P	Application Reset	61
BUSSTAT	Bus Status Register	00024_H	U,SV	BE	Application Reset	62
FIFOCFG	FIFO Configuration Register	00028_H	U,SV	U,SV,P	Application Reset	69
MRPSCTRL	Maximum Received Packet Size Control Register	$0002C_H$	U,SV	U,SV,P	Application Reset	70
RPSSTAT	Received Packet Size Status Register	00030_H	U,SV	BE	Application Reset	68
TPSCTRL	Transmit Packet Size Control Register	00034_H	U,SV	U,SV,P	Application Reset	67
FFSSTAT	Filled FIFO Stages Status Register	00038_H	U,SV	BE	Application Reset	71
TIMCFG	Timing Configuration Register	00040_H	U,SV	U,SV,P	Application Reset	64
ERRIRQSM	Error Interrupt Request Source Mask Register	00060_H	U,SV	U,SV,P	Application Reset	77
ERRIRQSS	Error Interrupt Request Source Status Register	00064_H	U,SV	BE	Application Reset	77
ERRIRQSC	Error Interrupt Request Source Clear Register	00068_H	U,SV	U,SV,P	Application Reset	78
PIRQSM	Protocol Interrupt Request Source Mask Register	00070_H	U,SV	U,SV,P	Application Reset	80
PIRQSS	Protocol Interrupt Request Source Status Register	00074_H	U,SV	BE	Application Reset	81

Table 303 Register Overview - I2C (ascending Offset Address) (cont'd)

Short Name	Long Name	Offset Address	Access Mode		Reset	Page Number
			Read	Write		
PIRQSC	Protocol Interrupt Request Source Clear Register	00078_H	U,SV	U,SV,P	Application Reset	82
RIS	Raw Interrupt Status Register	00080_H	U,SV	BE	Application Reset	72
IMSC	Interrupt Mask Control Register	00084_H	U,SV	U,SV,P	Application Reset	73
MIS	Masked Interrupt Status Register	00088_H	U,SV	BE	Application Reset	74
ICR	Interrupt Clear Register	$0008C_H$	U,SV	U,SV,P	Application Reset	75
ISR	Interrupt Set Register	00090_H	U,SV	U,SV,P	Application Reset	75
TXD	Transmission Data Register	08000_H	U,SV	U,SV,P	Application Reset	67
	Reserved ($03FFC_H$ Byte)	08004_H	BE	BE		
RXD	Reception Data Register	$0C000_H$	U,SV	BE	Application Reset	68
	Reserved ($03FFC_H$ Byte)	$0C004_H$	BE	BE		
CLC	Clock Control Register	10000_H	U,SV	SV,E,P	Application Reset	49
MODID	Module Identification Register	10004_H	SV	BE	Application Reset	50
GPCTL	General Purpose Control Register	10008_H	SV	SV,P	Application Reset	51
ACCEN0	Access Enable Register 0	$1000C_H$	SV	SV,SE	Application Reset	51
ACCEN1	Access Enable Register 1	10010_H	SV	SV,SE	Application Reset	52
KRST0	Kernel Reset Register 0	10014_H	SV	SV,E,P	Application Reset	52
KRST1	Kernel Reset Register 1	10018_H	SV	SV,E,P	Application Reset	53
KRSTCLR	Kernel Reset Status Clear Register	$1001C_H$	SV	SV,E,P	Application Reset	54

Notes

1. *All I2C registers are Application Reset registers.*

34.4.1 Global Module Control Registers

Module Identification Register

This register contains read-only information about the module and its revision.

ID

Module Identification Register **(00008$_H$)** **Application Reset Value: 0000 5705$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOD_NUMBER								**MOD_REV**							
r								r							

Field	Bits	Type	Description
MOD_REV	7:0	r	**Module Revision Number** This bit-field defines the revision number.
MOD_NUMBER	15:8	r	**Module Number** This bit-field defines the module identification number.
0	31:16	r	**Reserved** Read as 0; should be written with 0.

RUN Control Register

This register selects configuration mode or run mode.

RUNCTRL

RUN Control Register **(00010$_H$)** **Application Reset Value: 0000 0000$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0															**RUN**
r															rw

Field	Bits	Type	Description
RUN	0	rw	**Enable I2C-bus Interface** 0_B I2C-bus interface disabled; write access to configuration registers enabled 1_B Participation in I2C-bus communication enabled (if properly configured); write access to configuration registers disabled

Field	Bits	Type	Description
0	31:1	r	**Reserved** Read as 0; should be written with 0.

End Data Control Register

This register is used to either turn around the data transmission direction or address another slave without sending a stop condition. Also the software can stop the slave-transmitter by sending a not-acknowledge when working as master-receiver or even stop data transmission immediately when operating as master-transmitter. The writing to the bits of this control register is only effective in certain states.

ENDDCTRL
End Data Control Register **(00014_H)** **Application Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0														**SETEND**	**SETRSC**
r														w	w

Field	Bits	Type	Description
SETRSC	0	w	**Set Restart Condition** This bit is always read as 0. 0_B Has no effect. 1_B The master wants to restart a data transmission (changing slave/direction). The effect depends on the current state. MASTER RECEIVES BYTES: The master puts a not-acknowledge on the bus and switches to MASTER RESTART state. MASTER TRANSMITS BYTES: After the current byte has been sent, the master switches to MASTER RESTART state.

Field	Bits	Type	Description
SETEND	1	w	**Set End of Transmission** This bit is always read as 0. *Note: Do not write 1 to this bit when bus is free. This will cause an abort after the first byte when a new transfer is started.* 0_B Has no effect. 1_B The effect depends on the current state. MASTER RECEIVES BYTES: After receiving the current byte, the master puts a not-acknowledge on the bus to indicate the transmission end to the slave-transmitter. Next it produces a stop condition on the bus and changes its state to LISTENING. MASTER TRANSMITS BYTES: After sending the current byte and receiving an acknowledge or not-acknowledge from the slave-receiver, the master puts a stop condition on the bus to close the data transmission and changes its state to LISTENING. MASTER RESTART: The master puts a stop condition on the bus to close the data transmission and changes its state to LISTENING. SLAVE RECEIVES BYTES: The slave-receiver puts a not-acknowledge on the bus after the received byte and changes its state to TRANSMISSION FINISHED.
0	31:2	r	**Reserved** Read as 0; should be written with 0.

Address Configuration Register

This configuration register contains the I2C-address (when addressed as a slave) and some bits that control the basic operation of the peripheral.

ADDRCFG
Address Configuration Register **(00020_H)** **Application Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0										**SOPE**	**SONA**	**MnS**	**MCE**	**GCE**	**TBAM**
r										rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0						**ADR**									
r						rw									

Field	Bits	Type	Description
ADR	9:0	rw	**I2C-bus Device Address** This bit-field determines the address of the device when addressed as a slave. (Watch out for reserved addresses by referring to I2C-bus spec V2.1.) Depending on setting of TBAM, this is either a 7-bit address (bits [7:1]) or a 10-bit address (bits [9:0]).

Field	Bits	Type	Description
TBAM	16	rw	**Ten Bit Address Mode** *Note: When this bit is zero, only bits 7 down to 1 of the ADR field are valid.* 0_B 7-bit address mode enabled. 1_B 10-bit address mode enabled.
GCE	17	rw	**General Call Enable** 0_B Ignore general call occurrence. 1_B Enable general call detection; when detected, an acknowledge will be put on the bus
MCE	18	rw	**Master Code Enable** 0_B Device is not able to get along with high-speed mode 1_B Device is able to handle master code
MnS	19	rw	**Master / not Slave** 0_B Peripheral is configured as slave 1_B Peripheral is configured as master
SONA	20	rw	**Stop on Not-acknowledge** *Note: After successful transmission of a master code (during high-speed mode) SONA is not considered till a stop condition is manually generated by SETEND.* 0_B Device changes to MASTER RESTART state. 1_B Device puts a stop condition on the bus and changes to LISTENING state.
SOPE	21	rw	**Stop on Packet End** **Notes** *1. This bit-field should be used only in Master Mode. In slave modeshould always be 0.* *2. If device works as receiver a not-acknowledge is always generatedon package end.* *3. After successful transmission of a master code (during high-speedmode) SOPE is not considered till a stop condition is manually generated by SETEND.* 0_B Device enters MASTER RESTART state when the data packet end is indicated by the FIFO. 1_B Device puts a stop condition on the bus when the data packet end is indicated by the FIFO and changes to MASTER LISTENING state.
0	15:10, 31:22	r	**Reserved** Read as 0; should be written with 0.

Bus Status Register

This register contains status information of the I2C-bus. This additional information can be used by software to start appropriate actions.

BUSSTAT

Bus Status Register (00024$_H$) **Application Reset Value: 0000 0000$_H$**

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">0</td></tr>
<tr><td colspan="16">r</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="13">0</td><td>RnW</td><td colspan="2">BS</td></tr>
<tr><td colspan="13">r</td><td>rh</td><td colspan="2">rh</td></tr>
</table>

Field	Bits	Type	Description
BS	1:0	rh	**Bus Status** Shows the current status on the I2C-bus. 00_B I2C-bus is free (no start condition detected). 01_B A start condition has been detected on the bus (bus busy). 10_B The device is working as master and has claimed the control on the I2C-bus (busy master). 11_B A remote master has accessed this device as slave.
RnW	2	rh	**Read/not Write** Set by hardware automatically after address byte has been sent/received. 0_B Working as transmitter (Write to I2C-bus). 1_B Working as receiver (Read from I2C-bus).
0	31:3	r	**Reserved** Read as 0; should be written with 0.

Fractional Divider Configuration Register

This configuration register is used to program the fractional divider of the I2C-bus for standard and fast mode. Before the peripheral is switched on by setting the RUN bit, the register should be configured.

FDIVCFG

Fractional Divider Configuration Register (00018$_H$) **Application Reset Value: 0000 0000$_H$**

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">0</td><td colspan="8">INC</td></tr>
<tr><td colspan="8">r</td><td colspan="8">rw</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="5">0</td><td colspan="11">DEC</td></tr>
<tr><td colspan="5">r</td><td colspan="11">rw</td></tr>
</table>

Field	Bits	Type	Description
DEC	10:0	rw	**Decrement Value of Fractional Divider** For standard/fast mode, see **Clock and Timing Control**.

Field	Bits	Type	Description
INC	23:16	rw	**Increment Value of Fractional Divider** For standard/fast mode, see **Clock and Timing Control**.
0	15:11, 31:24	r	**Reserved** Read as 0; should be written with 0.

Fractional Divider High-speed Mode Configuration Register

This configuration register is used to program the fractional divider of the I2C-bus for high-speed mode. Before the peripheral is switched on by setting the RUN bit, the register should be configured if high-speed mode is used.

FDIVHIGHCFG

Fractional Divider High-speed Mode Configuration Register(0001C$_H$) Application Reset Value: 0000 0000$_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0				**0**								**INC**			
r				r								rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		0								**DEC**					
		r								rw					

Field	Bits	Type	Description
DEC	10:0	rw	**Decrement Value of Fractional Divider** For high-speed mode, see **Clock and Timing Control**.
INC	23:16	rw	**Increment Value of Fractional Divider** For high-speed mode, see **Clock and Timing Control**.
0	15:11, 30:24, 31	r	**Reserved** Read as 0; should be written with 0.

Timing Configuration Register

This configuration register adjusts some timings of the I2C-bus signals SCL and SCA. The delays are given in kernel_clk cycles (denoted as stages below).

The delayed stages may have +/- 1 stage deviation.

TIMCFG
Timing Configuration Register **(00040$_H$)** **Application Reset Value: 0000 0000$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SCL_LOW_LEN								**0**			**HS_SDA_DEL**				
rw								r			rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FS_SCL_LOW	**EN_SCL_LOW_LEN**	**0**		**SCL_DEL_HD_STA**			**HS_SDA_DEL_HD_DAT**			**SDA_DEL_HD_DAT**					
rw	rw	r		rw			rw			rw					

Field	Bits	Type	Description
SDA_DEL_HD_DAT	5:0	rw	**SDA Delay Stages for Data Hold Time in Standard and Fast modes** SDA delay stages for data hold time in standard and fast modes. *Note: SDA delay from SCL falling edge but will also affect SDA Setup time relative to next SCL rising edge* 00$_H$ 3 stages delay … 3F$_H$ 66 stages delay
HS_SDA_DEL_HD_DAT	8:6	rw	**SDA Delay Stages for Data Hold Time in High-speed Mode** SDA delay stages for data hold time in HS mode. *Note: SDA delay from SCL falling edge but will also affect SDA Setup time relative to next SCL rising edge* 000$_B$ 3 stages delay … 111$_B$ 10 stages delay
SCL_DEL_HD_STA	11:9	rw	**SCL Delay Stages for Hold Time Start (Restart) Bit** 000$_B$ 2 stages delay … 111$_B$ 9 stages delay
EN_SCL_LOW_LEN	14	rw	**Enable Direct Configuration of SCL Low Period Length in Fast Mode** 0$_B$ SCL low period is a fixed part of the whole period, as defined by FS_SCL_LOW 1$_B$ SCL low period is determined by the setting of SCL_LOW_LEN

Field	Bits	Type	Description
FS_SCL_LOW	15	rw	**Set Fast Mode SCL Low Period Timing** The internal duration of the SCL low time with respect to the period length as defined by the baudrate setting, can be enlarged for the Fast Speed Mode, in order to meet the asymmetric duty cycle requirements from the standard. The detailed formulas are given in the functional specification. 0_B Standard mode SCL low period timing. For INC = 1 it is 5/8 of period. 1_B Fast mode SCL low period timing. For INC = 1 it is 6/8 of period.
HS_SDA_DEL	20:16	rw	**SDA Delay Stages for Start/Stop bit in High-speed Mode** 00_H 3 stages delay … 07_H 10 stages delay
SCL_LOW_LEN	31:24	rw	**SCL Low Length in Fast Mode** If enabled by EN_SCL_LOW_LEN setting, this field determines the extension of the SCL low time. In case of INC = 1, the low time is extended by the number of kernel_clk cycles. In general, there is a more complex formula, as given in the functional specification. The total period time is not changed, i.e., the SCL high period is reduced accordingly. Setting SCL low time to period length or higher is not supported and would lead to unpredictable results.
0	13:12, 23:21	r	**Reserved** Read as 0; should be written with 0.

34.4.2 FIFO Registers

Transmission Data Register

The software has to write the characters to be transmitted into this register.

A larger address range (8000$_H$ to BFFC$_H$) is reserved for the FIFO. Accessing any address in the defined range has the same effect as accessing the first address.

A read access to TXD register is not possible, it will return 0 in all cases. Reading has no effect on the FIFO

When using byte or half word access from the bus, the TX FIFO pointer will only be increased, if one of the following conditions is fulfilled:

- The most significant byte or half word of the FIFO stage is written
- The packet end is reached

TXD

Transmission Data Register **(08000$_H$)** **Application Reset Value: 0000 0000$_H$**

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">**TXD**</td></tr>
<tr><td colspan="16">w</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">**TXD**</td></tr>
<tr><td colspan="16">w</td></tr>
</table>

Field	Bits	Type	Description
TXD	31:0	w	**Transmission Data** Characters to be transmitted

Transmit Packet Size Control Register

This register is used to indicate the peripheral the size of the packet to be transmitted. Writing the packet size to this register if the FIFO controller is configured for flow controller mode initiates the data requests (BREQ, SREQ, ...).Writing to this register in configuration state has no impact.

TPSCTRL

Transmit Packet Size Control Register **(00034$_H$)** **Application Reset Value: 0000 0000$_H$**

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">**0**</td></tr>
<tr><td colspan="16">r</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">**0**</td><td colspan="14">**TPS**</td></tr>
<tr><td colspan="2">r</td><td colspan="14">rwh</td></tr>
</table>

Field	Bits	Type	Description
TPS	13:0	rwh	**Transmit Packet Size** Length in characters of the transmit packet, write value range: 1 to 16383 Reading returns the written value as long as it is not loaded to an internal counter. After that, reading returns 0 and a new value can be written.
0	31:14	r	**Reserved** Read as 0; should be written with 0.

Reception Data Register

The software can read the received characters from this register.

A larger address range (C000$_H$ to FFFC$_H$) is reserved for the FIFO . Reading from any address in the defined range has the same effect as reading from the first address.

RXD

Reception Data Register **(0C000$_H$)** **Application Reset Value: 0000 0000$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXD															
rh															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXD															
rh															

Field	Bits	Type	Description
RXD	31:0	rh	**Reception Data** Received characters

Received Packet Size Status Register

This register indicates the size of the received data packet to the software. The software should read this register after the last request of a packet.

RPSSTAT

Received Packet Size Status Register **(00030$_H$)** **Application Reset Value: 0000 0000$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0		**RPS**													
r		rh													

Field	Bits	Type	Description
RPS	13:0	rh	**Received Packet Size** Length in characters of the received packet (0 to 16383)
0	31:14	r	**Reserved** Read as 0; should be written with 0.

FIFO Configuration Register

This configuration register is used to set up the FIFO before the peripheral is enabled and data is received or transmitted.

FIFOCFG
FIFO Configuration Register **(00028_H)** **Application Reset Value: 0000 0022_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						0							**CRBC**	**TXFC**	**RXFC**
						r							rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0		**TXFA**		**0**		**RXFA**		**0**		**TXBS**		**0**		**RXBS**	
r		rw		r		rw		r		rw		r		rw	

Field	Bits	Type	Description
RXBS	1:0	rw	**RX Burst Size** 00_B 1 word 01_B 2 words 10_B 4 words 11_B Do not use this combination
TXBS	5:4	rw	**TX Burst Size** 00_B 1 word 01_B 2 words 10_B 4 words 11_B Do not use this combination
RXFA	9:8	rw	**RX FIFO Alignment** Use byte alignment wherever it is possible. 00_B Byte aligned (character alignment) 01_B Half word aligned (character alignment of two characters) 10_B Word aligned (character alignment of four characters) 11_B Do not use this combination

Field	Bits	Type	Description
TXFA	13:12	rw	**TX FIFO Alignment** Use byte alignment wherever it is possible. 00_B Byte aligned (character alignment) 01_B Half word aligned (character alignment of two characters) 10_B Word aligned (character alignment of four characters) 11_B Do not use this combination
RXFC	16	rw	**RX FIFO Flow Control** 0_B RX FIFO not as flow controller 1_B RX FIFO as flow controller
TXFC	17	rw	**TX FIFO Flow Control** 0_B TX FIFO not as flow controller 1_B TX FIFO as flow controller
CRBC	18	rw	**Clear Request Behavior Configuration** Used to configure the clear request behavior for the FIFO data request. Can only be used for single request and must be set to “0” when burst accesses are used in the system (eg. when TX/RXBS > 0) 0_B Data request is cleared by Software. 1_B Data request is cleared automatically when Write/Read access to FIFO occurs.
0	3:2, 7:6, 11:10, 15:14, 31:19	r	**Reserved** Read as 0; should be written with 0.

Maximum Received Packet Size Control Register

This register is used to limit the received packet size. The register value may be changed in any state of the FIFO.

MRPSCTRL

Maximum Received Packet Size Control Register(0002C$_H$) Application Reset Value: 0000 0000$_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0		**MRPS**													
r		rwh													

Field	Bits	Type	Description
MRPS	13:0	rwh	**Maximum Received Packet Size** Length in characters of packet to be received; write value range: 0 (unlimited size) to 16383 Reading returns the written value as long as the previous packet has not been read completely from the FIFO. After that, MRPS is loaded to an internal register, reading returns 0 and a new value can be written.
0	31:14	r	**Reserved** Read as 0; should be written with 0.

Filled FIFO Stages Status Register

This register is used to indicate the number of filled FIFO stages.

FFSSTAT

Filled FIFO Stages Status Register **(00038_H)** **Application Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0										**FFS**					
r										rh					

Field	Bits	Type	Description
FFS	5:0	rh	**Filled FIFO Stages** Number of filled FIFO stages (0 to 8)
0	31:6	r	**Reserved** Read as 0; should be written with 0.

34.4.3 Basic Interrupt Registers

For an overview of the Service Request Block (SRB) see **Section 34.3.1.7**.

Raw Interrupt Status Register

This read-only register returns the current raw status value (without reflecting the mask) of the interrupt request sources. One status bit is provided for each request. A write to this register has no effect. The status bits are set by hardware or software (via register **ISR**) and can be cleared by software (via register **ICR**).

RIS

Raw Interrupt Status Register **(00080$_H$)** **Application Reset Value: 0000 0000$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0										**I2C_P_INT**	**I2C_ERR_INT**	**BREQ_INT**	**LBREQ_INT**	**SREQ_INT**	**LSREQ_INT**
r										rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
LSREQ_INT	0	rh	**Last Single Request Interrupt** 0_B No interrupt request 1_B Interrupt request pending
SREQ_INT	1	rh	**Single Request Interrupt** 0_B No interrupt request 1_B Interrupt request pending
LBREQ_INT	2	rh	**Last Burst Request Interrupt** 0_B No interrupt request 1_B Interrupt request pending
BREQ_INT	3	rh	**Burst Request Interrupt** 0_B No interrupt request 1_B Interrupt request pending
I2C_ERR_INT	4	rh	**I2C Error Interrupt** This is the combined bit for indication of FIFO errors due to overflow and underflow. 0_B No interrupt request 1_B Interrupt request pending

Field	Bits	Type	Description
I2C_P_INT	5	rh	**I2C Protocol Interrupt** This is the combined bit for indication of a protocol event in the I2C kernel. 0_B No interrupt request 1_B Interrupt request pending
0	31:6	r	**Reserved** Read as 0; should be written with 0.

Interrupt Mask Control Register

A write of 1 to a particular bit of this register enables the corresponding interrupt request; a write of 0 disables it. A read to this register returns the current mask bits. After reset all requests are disabled.

IMSC

Interrupt Mask Control Register **(00084_H)** **Application Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0										I2C_P_INT	I2C_ERR_INT	BREQ_INT	LBREQ_INT	SREQ_INT	LSREQ_INT
r										rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
LSREQ_INT	0	rw	**Last Single Request Interrupt** 0_B Interrupt request disabled 1_B Interrupt request enabled
SREQ_INT	1	rw	**Single Request Interrupt** 0_B Interrupt request disabled 1_B Interrupt request enabled
LBREQ_INT	2	rw	**Last Burst Request Interrupt** 0_B Interrupt request disabled 1_B Interrupt request enabled
BREQ_INT	3	rw	**Burst Request Interrupt** 0_B Interrupt request disabled 1_B Interrupt request enabled
I2C_ERR_INT	4	rw	**I2C Error Interrupt** 0_B Interrupt request disabled 1_B Interrupt request enabled
I2C_P_INT	5	rw	**I2C Protocol Interrupt** 0_B Interrupt request disabled 1_B Interrupt request enabled

Field	Bits	Type	Description
0	31:6	r	**Reserved** Read as 0; should be written with 0.

Masked Interrupt Status Register

This read-only register returns the masked status value (derived from registers **RIS** and **IMSC**) of the corresponding interrupt requests. A write to this register has no effect.

MIS

Masked Interrupt Status Register **(00088_H)** **Application Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0										**I2C_P_INT**	**I2C_ERR_INT**	**BREQ_INT**	**LBREQ_INT**	**SREQ_INT**	**LSREQ_INT**
r										rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
LSREQ_INT	0	rh	**Last Single Request Interrupt** 0_B No interrupt request 1_B Interrupt request pending
SREQ_INT	1	rh	**Single Request Interrupt** 0_B No interrupt request 1_B Interrupt request pending
LBREQ_INT	2	rh	**Last Burst Request Interrupt** 0_B No interrupt request 1_B Interrupt request pending
BREQ_INT	3	rh	**Burst Request Interrupt** 0_B No interrupt request 1_B Interrupt request pending
I2C_ERR_INT	4	rh	**I2C Error Interrupt** This is the combined bit for indication of FIFO errors due to overflow and underflow. 0_B No interrupt request 1_B Interrupt request pending
I2C_P_INT	5	rh	**I2C Protocol Interrupt** This is the combined bit for indication of a protocol event in the I2C kernel. 0_B No interrupt request 1_B Interrupt request pending
0	31:6	r	**Reserved** Read as 0; should be written with 0.

Interrupt Clear Register

On a write of 1 to a particular bit of this write-only register, the corresponding interrupt request is cleared; a write of 0 has no effect. Reading the register returns 0.

ICR

Interrupt Clear Register **($0008C_H$)** **Application Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0												**BREQ_INT**	**LBREQ_INT**	**SREQ_INT**	**LSREQ_INT**
r												w	w	w	w

Field	Bits	Type	Description
LSREQ_INT	0	w	**Last Single Request Interrupt** 0_B No change 1_B Clear interrupt request
SREQ_INT	1	w	**Single Request Interrupt** 0_B No change 1_B Clear interrupt request
LBREQ_INT	2	w	**Last Burst Request Interrupt** 0_B No change 1_B Clear interrupt request
BREQ_INT	3	w	**Burst Request Interrupt** 0_B No change 1_B Clear interrupt request
0	31:4	r	**Reserved** Read as 0; should be written with 0.

Interrupt Set Register

On a write of 1 to a particular bit of this write-only register, the corresponding interrupt request is set; a write of 0 has no effect. Reading the register returns 0.

ISR

Interrupt Set Register **(00090$_H$)** **Application Reset Value: 0000 0000$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0										**I2C_P_INT**	**I2C_ERR_INT**	**BREQ_INT**	**LBREQ_INT**	**SREQ_INT**	**LSREQ_INT**
r										w	w	w	w	w	w

Field	Bits	Type	Description
LSREQ_INT	0	w	**Last Single Request Interrupt** 0_B No change 1_B Set interrupt request
SREQ_INT	1	w	**Single Request Interrupt** 0_B No change 1_B Set interrupt request
LBREQ_INT	2	w	**Last Burst Request Interrupt** 0_B No change 1_B Set interrupt request
BREQ_INT	3	w	**Burst Request Interrupt** 0_B No change 1_B Set interrupt request
I2C_ERR_INT	4	w	**I2C Error Interrupt** 0_B No change 1_B Set interrupt request
I2C_P_INT	5	w	**I2C Protocol Interrupt** 0_B No change 1_B Set interrupt request
0	31:6	r	**Reserved** Read as 0; should be written with 0.

34.4.4 Error Interrupt Source Registers

For an overview of the source register operation see **Section 34.3.1.7.2**.

Error Interrupt Request Source Mask Register

A write of 1 to a particular bit of this register enables the corresponding error interrupt request source; a write of 0 disables it. A read to this register returns the current mask bits. After reset all sources are enabled.

The interrupts are explained in detail in description of register **ERRIRQSS**.

ERRIRQSM

Error Interrupt Request Source Mask Register (00060$_H$) **Application Reset Value: 0000 000F$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0												**TXF_OFL**	**TXF_UFL**	**RXF_OFL**	**RXF_UFL**
r												rw	rw	rw	rw

Field	Bits	Type	Description
RXF_UFL	0	rw	**RX FIFO Underflow** 0$_B$ Interrupt request source disabled 1$_B$ Interrupt request source enabled
RXF_OFL	1	rw	**RX FIFO Overflow** 0$_B$ Interrupt request source disabled 1$_B$ Interrupt request source enabled
TXF_UFL	2	rw	**TX FIFO Underflow** 0$_B$ Interrupt request source disabled 1$_B$ Interrupt request source enabled
TXF_OFL	3	rw	**TX FIFO Overflow** 0$_B$ Interrupt request source disabled 1$_B$ Interrupt request source enabled
0	31:4	r	**Reserved** Read as 0; should be written with 0.

Error Interrupt Request Source Status Register

This read-only register returns the current raw status value (without reflecting the mask) of the error interrupt request sources. A write to this register has no effect. The error status bits are set by hardware and can be cleared by software (via register **ERRIRQSC**).

ERRIRQSS

Error Interrupt Request Source Status Register (00064$_H$) **Application Reset Value: 0000 0000$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0												**TXF_OFL**	**TXF_UFL**	**RXF_OFL**	**RXF_UFL**
r												rh	rh	rh	rh

Field	Bits	Type	Description
RXF_UFL	0	rh	**RX FIFO Underflow** The FIFO has detected an RX FIFO underflow. 0_B No interrupt request 1_B Interrupt request pending
RXF_OFL	1	rh	**RX FIFO Overflow** The I2C kernel has detected a RX FIFO overflow 0_B No interrupt request 1_B Interrupt request pending
TXF_UFL	2	rh	**TX FIFO Underflow** The I2C kernel has detected a TX FIFO underflow. 0_B No interrupt request 1_B Interrupt request pending
TXF_OFL	3	rh	**TX FIFO Overflow** The FIFO has detected a TX FIFO overflow. 0_B No interrupt request 1_B Interrupt request pending
0	31:4	r	**Reserved** Read as 0; should be written with 0.

Error Interrupt Request Source Clear Register

On a write of 1 to a particular bit of this write-only register, the corresponding error interrupt request source is cleared and if no further error interrupt request sources are active, the whole error interrupt is cleared. If the corresponding bit is set by SW via the **ISR** register, then it can be cleared by setting any non-reserved bit of the interrupt request source clear register**.** A write of 0 has no effect. Reading the register returns 0.

The interrupts are explained in detail in description of register **ERRIRQSS**.

ERRIRQSC

Error Interrupt Request Source Clear Register (00068$_H$) **Application Reset Value: 0000 0000$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0												**TXF_OFL**	**TXF_UFL**	**RXF_OFL**	**RXF_UFL**
r												w	w	w	w

Field	Bits	Type	Description
RXF_UFL	0	w	**RX FIFO Underflow** 0_B No change 1_B Clear interrupt request source
RXF_OFL	1	w	**RX FIFO Overflow** 0_B No change 1_B Clear interrupt request source
TXF_UFL	2	w	**TX FIFO Underflow** 0_B No change 1_B Clear interrupt request source
TXF_OFL	3	w	**TX FIFO Overflow** 0_B No change 1_B Clear interrupt request source
0	31:4	r	**Reserved** Read as 0; should be written with 0.

34.4.5 Protocol Interrupt Source Registers

For an overview of the source register operation see **Section 34.3.1.7.2**.

Protocol Interrupt Request Source Mask Register

A write of 1 to a particular bit of this register enables the corresponding protocol interrupt request source; a write of 0 disables it. A read to this register returns the current mask bits. After reset all sources are enabled.

The interrupts are explained in detail in description of register **PIRQSS**.

PIRQSM
Protocol Interrupt Request Source Mask Register(00070_H) **Application Reset Value: 0000 $007F_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0									**RX**	**TX_END**	**NACK**	**AL**	**MC**	**GC**	**AM**
r									rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
AM	0	rw	**Address Match** 0_B Interrupt request source disabled 1_B Interrupt request source enabled
GC	1	rw	**General Call** 0_B Interrupt request source disabled 1_B Interrupt request source enabled
MC	2	rw	**Master Code** 0_B Interrupt request source disabled 1_B Interrupt request source enabled
AL	3	rw	**Arbitration Lost** 0_B Interrupt request source disabled 1_B Interrupt request source enabled
NACK	4	rw	**Not-acknowledge Received** 0_B Interrupt request source disabled 1_B Interrupt request source enabled
TX_END	5	rw	**Transmission End** 0_B Interrupt request source disabled 1_B Interrupt request source enabled
RX	6	rw	**Receive Mode** 0_B Interrupt request source disabled 1_B Interrupt request source enabled
0	31:7	r	**Reserved** Read as 0; should be written with 0.

Protocol Interrupt Request Source Status Register

This read-only register returns the current raw status value (without reflecting the mask) of the protocol interrupt request sources. A write to this register has no effect. The protocol interrupt status bits are set by hardware and can be cleared by software (via register **PIRQSC**).

PIRQSS

Protocol Interrupt Request Source Status Register(00074$_H$) **Application Reset Value: 0000 0000$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0									**RX**	**TX_END**	**NACK**	**AL**	**MC**	**GC**	**AM**
r									rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
AM	0	rh	**Address Match** Device (in master or slave mode) is addressed by remote master (matching device address). Accordingly, bit-field BS in register **BUSSTAT** is set to 11_B. 0_B No interrupt request 1_B Interrupt request pending
GC	1	rh	**General Call** Remote master has transmitted a general call. 0_B No interrupt request 1_B Interrupt request pending
MC	2	rh	**Master Code** Remote master has transmitted a master call. 0_B No interrupt request 1_B Interrupt request pending
AL	3	rh	**Arbitration Lost** Device (master mode) lost the control on the I2C-bus due to losing arbitration procedure. Accordingly, bit-field BS in register **BUSSTAT** is set to 01_B. 0_B No interrupt request 1_B Interrupt request pending
NACK	4	rh	**Not-acknowledge Received** When working as transmitter this interrupt indicates a not-acknowledge from the remote receiver. The SW has to decide what further steps have to be taken. 0_B No interrupt request 1_B Interrupt request pending

Field	Bits	Type	Description
TX_END	5	rh	**Transmission End** The device has ended the data transfer properly (after stop condition has been put on the bus or the MASTER RESTART state has been entered.) 0_B No interrupt request 1_B Interrupt request pending
RX	6	rh	**Receive Mode** I2C kernel indicates switching from transmitting data to receiving data. 0_B No interrupt request 1_B Interrupt request pending
0	31:7	r	**Reserved** Read as 0; should be written with 0.

Protocol Interrupt Request Source Clear Register

On a write of 1 to a particular bit of this write-only register, the corresponding protocol interrupt request source is cleared and if no further protocol interrupt request sources are active, the whole protocol interrupt is cleared. If the corresponding **RIS** bit is set by SW via the **ISR** register, then it can be cleared by setting any non-reserved bit of the interrupt request source clear register**.** A write of 0 has no effect. Reading the register returns 0.

The interrupts are explained in detail in description of register **PIRQSS**.

PIRQSC

Protocol Interrupt Request Source Clear Register(00078_H) **Application Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0									**RX**	**TX_END**	**NACK**	**AL**	**MC**	**GC**	**AM**
r									w	w	w	w	w	w	w

Field	Bits	Type	Description
AM	0	w	**Address Match** 0_B No change 1_B Clear Interrupt source
GC	1	w	**General Call** 0_B No change 1_B Clear Interrupt source
MC	2	w	**Master Code** 0_B No change 1_B Clear Interrupt source