RealTek specification on

Set-Top Box Technology

Date: 2017-11-15

Version: 0.0.1

KYLIN-MISC-ARCH

Specification for Kylin Project

Specification for Kylin: MISC Architecture Specification

Warning

This document is not a STB Standard. It is distributed for review and comment. Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of

which they are aware and to provide supporting documentation. Distribution does not constitute publication.

1. MISC system

1.1 Register

1.1.1 Register summery

Physical Address	*Name*	*R/W*	*Description*
0x9801_B008	MIS_UMSK_ISR	R/W	MISC unmasked interrupt status Register
0x9801_B00C	MIS_ISR	R/W	MISC masked interrupt status Register. This register is the result of raw unmasked interrupt bits "and" with interrupt enable bits from each function (e.g. MIS_GPIE)
0x9801_B010	MIS_UMSK_ISR_SWC	R/W	MISC unmasked interrupt status Register in SWC
0x9801_B014	MIS_ISR_SWC	R/W	MISC masked interrupt status Register in SWC
0x9801_B018	MIS_SETTING_SWC	R/W	MISC register settings in SWC
0x9801_B01C	MIS_FAST_INT_EN_0	R/W	MISC fast interrupts enable
0x9801_B020	MIS_FAST_INT_EN_1	R/W	MISC fast interrupts enable
0x9801_B024	MIS_FAST_ISR	R/W	MISC fast interrupt status registers
0x9801_B028	Rvd	-	-
0x9801_B02C	MIS_DBG	R/W	MISC Debug Register.
0x9801_B030	MIS_DUMMY	R/W	MISC DUMMY Register. This is reserved for future used.
0x9801_B034	Rvd	R/W	
0x9801_B040	MIS_UMSK_ISR_GP0A	R/W	MISC unmasked GPIO 0 Assert interrupt status Register
0x9801_B044	MIS_UMSK_ISR_GP1A	R/W	MISC unmasked GPIO 1 Assert interrupt status Register
0x9801_B048 ~ 0x9801_B050	Rvd	-	-
0x9801_B054	MIS_UMSK_ISR_GP0DA	R/W	MISC unmasked GPIO 0 Dis-Assert interrupt status Register
0x9801_B058	MIS_UMSK_ISR_GP1DA	R/W	MISC unmasked GPIO 1 Dis-Assert interrupt status Register
0x9801_B05C	Rvd	-	-
0x9801_B060	MIS_UR_CTRL	R/W	MISC UR Control register
0x9801_B064	MIS_UR2_CTRL	R/W	MISC UR2 Control register
0x9801_B068	MIS_DUMMY1	R/W	MISC dummy register
0x9801_B06C	Rvd		
0x9801_B070	MIS_FAST_ISR_GPIO0_A	R/W	MISC fast masked GPIO assert interrupt status Register
0x9801_B074	MIS_FAST_ISR_GPIO1_A	R/W	MISC fast masked GPIO assert interrupt status Register
0x9801_B078	MIS_FAST_ISR_GPIO0_DA	R/W	MISC fast masked GPIO de-assert interrupt status Register

0x9801_B07C	MIS_FAST_ISR_GPIO1_DA	R/W	MISC fast masked GPIO de-assert interrupt status Register
0x9801_B080	MIS_SCPU_INT_EN	R/W	MISC SCPU Interrupt Enable register
0x9801_B084	Rvd	-	-
0x9801_B088	Rvd	-	-
0x9801_B08C	MIS_I2C2_SDA_DEL	R/W	MISC I2C2 SDA delay control register
0x9801_B090	MIS_I2C3_SDA_DEL	R/W	MISC I2C3 SDA delay control register
0x9801_B094	MIS_I2C4_SDA_DEL	R/W	MISC I2C4 SDA delay control register
0x9801_B098	MIS_I2C5_SDA_DEL	R/W	MISC I2C5 SDA delay control register
0x9801_B09C	MIS_RTC_SYS_SYNC	R/W	MISC RTC sys clock sync control register
0x9801_B0A0	MIS_FAST_INT_EN_2	R/W	MISC fast interrupts enable
0x9801_B0A4	MIS_UMSK_ISR_GP2A	R/W	MISC unmasked GPIO 2 Assert interrupt status Register
0x9801_B0A8	MIS_UMSK_ISR_GP2DA	R/W	MISC unmasked GPIO 2 Dis-Assert interrupt status Register
0x9801_B0AC	MIS_FAST_ISR_GPIO2_A	R/W	MISC fast masked GPIO assert interrupt status Register
0x9801_B0B0	MIS_FAST_ISR_GPIO2_DA	R/W	MISC fast masked GPIO de-assert interrupt status Register
0x9801_B0B4	MIS_FAST_INT_EN_3	R/W	MISC fast interrupts enable
0x9801_B0B8	MIS_UMSK_ISR_GP3A	R/W	MISC unmasked GPIO 3 Assert interrupt status Register
0x9801_B0BC	MIS_UMSK_ISR_GP3DA	R/W	MISC unmasked GPIO 3 Dis-Assert interrupt status Register
0x9801_B0C0	MIS_FAST_ISR_GPIO3_A	R/W	MISC fast masked GPIO assert interrupt status Register
0x9801_B0C4	MIS_FAST_ISR_GPIO3_DA	R/W	MISC fast masked GPIO de-assert interrupt status Register
0x9801_B0C8	Rvd		
0x9801_B0E0	MIS_GATING_EN	R/W	MIS clock gating enable control
0x9801_B0E4	MIS_DUMMY2	R/W	MISC dummy register
0x9801_B0E8	MIS_DUMMY3	R/W	MISC dummy register
0x9801_B0F0	MIS_UR_H5_CTRL	R/W	UR H5 basic control
0x9801_B0F4	MIS_UR_H5_ST	R	UR H5 status
0x9801_B0F8	MIS_UR2_H5_CTRL	R/W	UR H5 basic control
0x9801_B0FC	MIS_UR2_H5_ST	R	UR H5 status

1.1.2 Register Description

Module::MIS	Register::UMSK_ISR	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_B008

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:30	-	-	-
FAN_INT	29	R/W	‘b0	DC FAN interrupt flag.

Rvd	28	-	-	-
GSPI_INT	27	R/W	‘b0	GSPI interrupt flag.
Rvd	26:21	-	-	-
GPIODA_INT	20	R/W	‘b0	GPIO[98:0] dis-assert interrupt flag.
GPIOA_INT	19	R/W	‘b0	GPIO[98:0] assert interrupt flag. Reading MIS_ISR_GPDA to check which gpio assert interrupt.
Rvd	18:13	-	-	-
RTC_DATE_INT	12	R/W	‘b0	RTC date interrupt flag.
RTC_HOUR_INT	11	R/W	‘b0	RTC hour interrupt flag.
RTC_MIN_INT	10	R/W	‘b0	RTC minute interrupt flag.
RTC_HSEC_INT	9	R/W	‘b0	RTC half second interrupt flag.
Rvd	8	-	-	-
TC1_INT	7	R/W	‘b0	timer/counter 1 interrupt flag.
TC0_INT	6	R/W	‘b0	timer/counter 0 interrupt flag.
UR1_TO_INT	5	R/W	‘b0	uart1 timeout interrupt flag.
UR2_TO_INT	4	R/W	‘b0	uart2 timeout interrupt flag.
Rvd	3	-	-	-
WDOG_NMI_INT	2	R/W	‘b0	Watchdog Non-Mask interrupt.
Rvd	1	-	-	-
write_data	0	W	-	1 to set, 0 to clear bits with 1.

Note1: MIS_UMSK_ISR(MISC unmasked ISR) is raw interrupt flag and can’t be masked by interrupt enable. MIS_ISR is masked by interrupt enable and issue to SCPU interrupt controller.

Note2: I2C ,UART is synopsys IP. They only output masked interrupt.
We can polling I2C un_mask interupt flag from I2C register (I2C0: IC_RAW_INTR_STAT, I2C1: IC1_RAW_INTR_STAT).
But we can't polliing UART un_mask interrupt flag from UR register.

Note3: MIS_ISR_reg/MIS_UMSK_ISR_reg diagram.

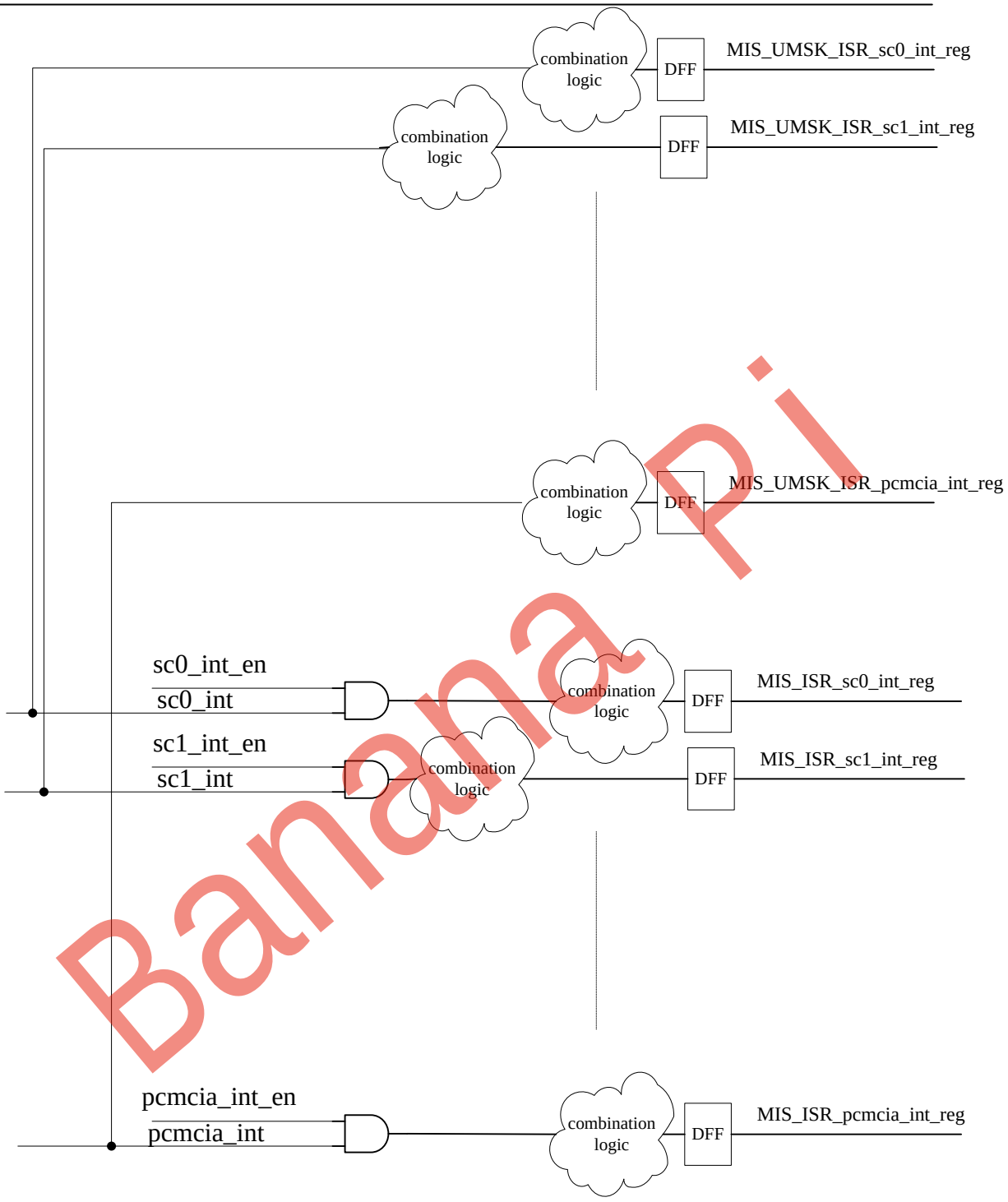


Figure 1 MIS_ISR_reg/MIS_UMSK_ISR_reg diagram.

Note4: Interrupt to SCPU/KCPU diagram.

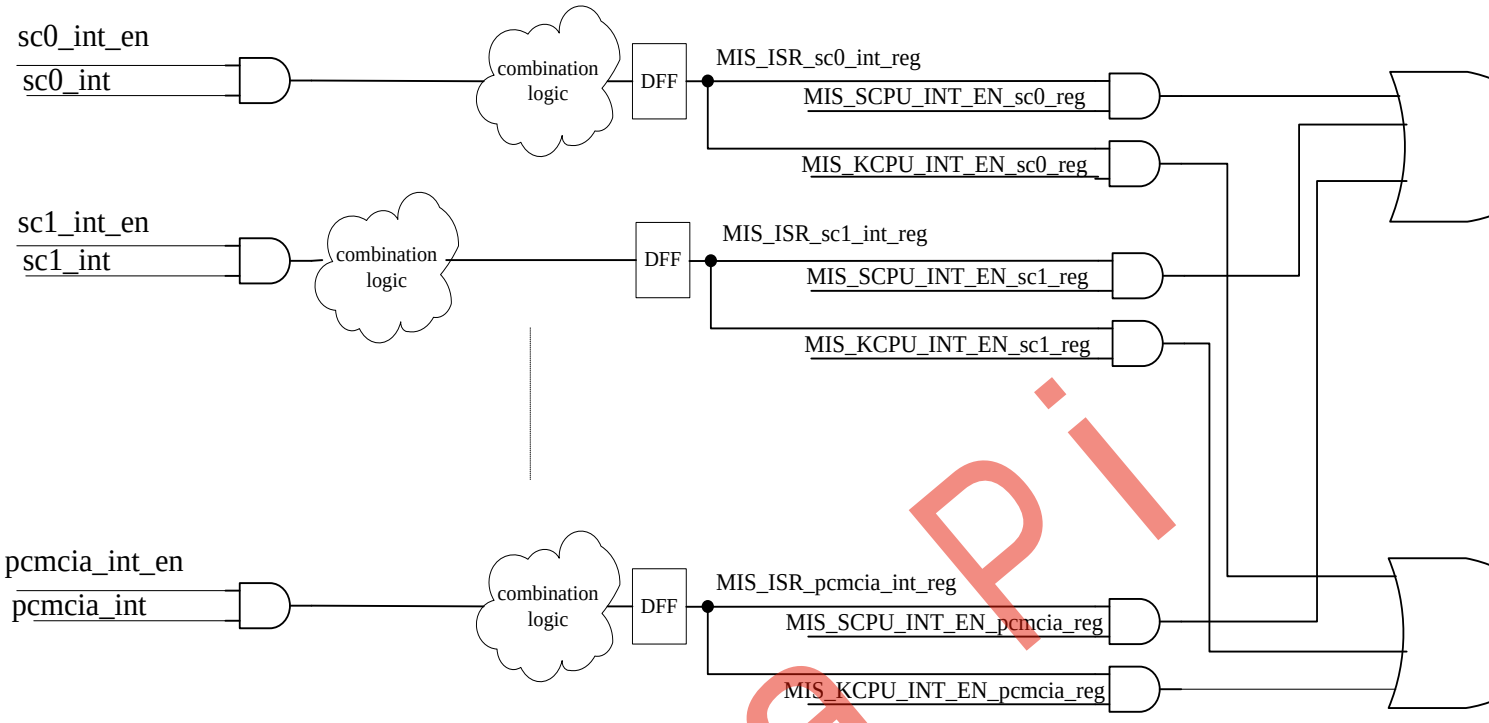


Figure 2 Interrupt to SCPU/KCPU diagram.

Module::MIS	Register::ISR	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_B00C

Name	**Bits**	**Read/Write**	**Reset State**	**Comments**
Rvd	31:30	-	-	-
FAN_INT	29	R/W	‘b0	DC FAN interrupt flag.
Rvd	28	-	-	-
GSPI_INT	27	R/W	‘b0	GSPI interrupt flag.
I2C2_INT	26	R/W	‘b0	Second i2c interrupt. If I2C2_INT be set to 1’b1, SW must handle i2c interrupt issue to let i2c exit interrupt condition first, then clear the I2C2_INT.
Rvd	25	-	-	-
SC0_INT	24	R/W	‘b0	Smart Card0 interrupt flag.
I2C3_INT	23	R/W	‘b0	Third i2c interrupt. If I2C3_INT be set to 1’b1, SW must handle i2c interrupt issue to let i2c exit interrupt condition first, then clear the I2C3_INT.

LSADC1_INT	22	R/W	‘b0	LSADC1 interrupt flag.
LSADC0_INT	21	R/W	‘b0	LSADC0 interrupt flag.
GPIODA_INT	20	R/W	‘b0	GPIO[100:0] dis-assert interrupt flag.
GPIOA_INT	19	R/W	‘b0	GPIO[100:0] assert interrupt flag. Reading MIS_ISR_GPDA to check which gpio assert interrupt.
Rvd	18:16	-	-	-
I2C4_INT	15	R/W	‘b0	Fourth i2c interrupt. If I2C4_INT be set to 1’b1, SW must handle i2c interrupt issue to let i2c exit interrupt condition first, then clear the I2C4_INT.
I2C5_INT	14	R/W	‘b0	Fivth i2c interrupt. If I2C5_INT be set to 1’b1, SW must handle i2c interrupt issue to let i2c exit interrupt condition first, then clear the I2C5_INT.
UR2_TO_INT	13	R/W	‘b0	uart2 timeout interrupt flag.
RTC_DATE_INT	12	R/W	‘b0	RTC date interrupt flag.
RTC_HOUR_INT	11	R/W	‘b0	RTC hour interrupt flag.
RTC_MIN_INT	10	R/W	‘b0	RTC minute interrupt flag.
RTC_HSEC_INT	9	R/W	‘b0	RTC half second interrupt flag. This regiter don’t issue isr to SCPU. Move function to ISO_ISR_RTC_HSEC_INT.This register will be remove at next project.. MIS_ISR_RTC_HSEC_INT don’t issue isr to SCPU. ISO_ISR_RTC_HSEC_INT issue isr to SCPU.
UR2_INT	8	R/W	‘b0	Uart2 interrupt. If UR2_INT be set to 1’b1, SW must handle uart2 interrupt issue to let uart2 exit interrupt condition first, then clear the UR2_INT.
TC1_INT	7	R/W	‘b0	timer/counter 1 interrupt flag.
TC0_INT	6	R/W	‘b0	timer/counter 0 interrupt flag.
UR1_TO_INT	5	R/W	‘b0	uart1 timeout interrupt flag.

Rvd	4	-	-	-
UR1_INT	3	R/W	‘b0	Uart1 interrupt. If UR1_INT be set to 1’b1, SW must handle uart1 interrupt issue to let uart1 exit interrupt condition first, then clear the UR1_INT.
WDOG_NMI_INT	2	R/W	‘b0	Watchdog interrupt.
Rvd	1	-	-	-
write_data	0	W	-	1 to set, 0 to clear bits with 1.

Note1: UART RX timeout condition: No characters in or out of the RX FIFO during the last 4 character times and there is at least 1 character in it during this time

Module::MIS	Register::UMSK_ISR_SWC	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_B010

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:4	-	-	-
WDOG_NMI_INT	3	R/W	‘b0	Watchdog Non-Mask interrupt.
Rvd	2	-	-	-
TC2_INT	1	R/W	‘b0	timer/counter 2 interrupt flag.
write_data	0	W	-	1 to set, 0 to clear bits with 1.

Module::MIS	Register::ISR_SWC	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_B014

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:4	-	-	-
WDOG_NMI_INT	3	R/W	‘b0	Watchdog interrupt.
I2C_2_INT	2	R/W	‘b0	I2C_2 (touch panel) interrupt flag
TC2_INT	1	R/W	‘b0	timer/counter 2 interrupt flag.
write_data	0	W	-	1 to set, 0 to clear bits with 1.

Module::MIS	Register::SETTING_SWC	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B018

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:2	-	-	-
I2C_2_SWC_EN	1	R/W	‘b0	Send to SB2 to control I2C_2 SWC mode enable or not.
I2C_2_EN	0	R/W	‘b0	Enable I2C_2 (touch panel) interrupt through mis_int_scpu or independent i2c_2_int_swc

				1: enable i2c_2 in SWC mode, and the interrupt output through i2c_2_int_swc port 0: i2c_2 in NWC mode, and the interrupt output through mis_int_scpu port

P.S. SWC int : Wdog_nmi, I2C_2 and TC2
FAST_int : GSPI, GPIO, I2C_2 and I2C_3
Enable Priority : SWC > FAST > ISR
GPIO extension from 60 to 103 is included in FAST ISR

signal	condition1	condition2	condition3	condition4
GSPI	MIS_FAST_INT_EN_1_gspi_int		MIS_SCPU_INT_EN_gspi	GSPI_SPI_INT_EN*
MIS_ISR_gspi_int	1'b0		1'b1	1'b1
MIS_FAST_ISR_gspi_int	1'b1		X, don't care	1'b1
I2C2	MIS_FAST_INT_EN_1_i2c2_int	MIS_SETTING_SWC_i2c_2_en	MIS_SCPU_INT_EN_i2c2	I2C
MIS_ISR_i2c2_int	1'b0	1'b0	1'b1	i2c2_int
MIS_ISR_SWC_i2c_2_int	X, don't care	1'b1	X, don't care	i2c2_int
MIS_FAST_ISR_i2c2_int	1'b1	1'b0	X,don't care	i2c2_int
I2C3	MIS_FAST_INT_EN_1_i2c3_int		MIS_SCPU_INT_EN_i2c3	I2C
MIS_ISR_i2c3_int	1'b0		1'b1	i2c3_int
MIS_FAST_ISR_i2c3_int	1'b1		X,don't care	i2c3_int
GPIO[59:0]	MIS_FAST_INT_EN_1_gp_int[N]		MIS_SCPU_INT_EN_gpio	MIS_GPIE[N]
MIS_ISR_gpioa_int	1'b0		1'b1	1'b1
MIS_FAST_ISR_gpioa_int	1'b1		X,don't care	1'b1

Module::MIS	Register::FAST_INT_EN_0	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B01C

Name	Bits	Read/Write	Reset State	Comments
GP_INT	31:1	R/W	'h0	GPIO30~0 through fast interrupt enable
Rvd	0	-	-	-

Module::MIS	Register:: FAST_INT_EN_1	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B020

Name	Bits	Read/Write	Reset State	Comments
GSPI_INT	31	R/W	'b0	GSPI through fast interrupt
I2C3_INT	30	R/W	'b0	I2C_3 through fast interrupt
GP_INT	29:1	R/W	'b0	GPIO59~31 through fast interrupt enable
I2C2_INT	0	R/W	'b0	I2C_2 through fast interrupt

Module::MIS	Register::FAST_INT_EN_2	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B0A0

Name	Bits	Read/Write	Reset State	Comments
GP_INT	31:1	R/W	'h0	GPIO90~60 through fast interrupt enable
Rvd	0	-	-	-

Module::MIS	Register::FAST_INT_EN_3	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B0B4

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:11	-	-	-
GP_INT	10:1	R/W	'h00	GPIO100~91 through fast interrupt enable
Rvd	0	-	-	-

Module::MIS	Register::FAST_ISR	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_B024

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:6	-	-	-
i2c2_int	5	R/W	'b0	I2C_2 (touch panel) interrupt flag.
gspi_int	4	R/W	'b0	GSPI interrupt flag
i2c3_int	3	R/W	'b0	I2C_3 (Mems) interrupt flag.
gpioda_int	2	R/W	'b0	GPIO dis-assert interrupt flag.
gpioa_int	1	R/W	'b0	GPIO assert interrupt flag.
write_data	0	W	-	1 to set, 0 to clear bits with 1.

Module::MIS	Register::DBG	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B02C

Name	Bits	Read/Write	Reset State	Comments
Rvd	31..12	-	-	-
write_enable3	11	W	-	-
sel1	10..7	R/W	'h0	Select control of dbg_sel1.
write_enable2	6	W	-	-
sel0	5..2	R/W	'h0	Select control of dbg_sel0.
write_enable1	1	W	-	-

enable	0	R/W	‘b0	Debug Enable. If set to 1, the debug port will be switched to the selected probed signals for observation. If clear to 0 (default), the mis_dbg_out0 and mis_dbg_out1 are both static at 16’h0.

sel[3:0]	debug port output
7	gspi_dbg[15:0]

Module::MIS	Register::DUMMY	Set::1	ATTR::nor	Type::SR	ADDR::0x9801_B030

Name	Bits	Read/Write	Reset State	Comments
write_enable4	31	W	-	-
Rvd4	30:24	R/W	‘h0	-
write_enable3	23	W	-	-
Rvd3	22:16	R/W	‘h0	-
write_enable2	15	W	-	-
Rvd2	14:8	R/W	‘h0	bit9: 1’b1, IR RC6_EN, enable IR RC6 mode bit8: IR RC6 Trailer bit length[7]
write_enable1	7	W	-	-
Rvd1	6:0	R/W	‘h0	bit[6:0]: IR RC6 Trailer bit length[6:0]

Module::MIS	Register::UMSK_ISR_GP0A	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_B040

Name	Bits	Read/Write	Reset State	Comments
INT30	31	R/W	‘b0	GPIO30 assert interrupt flag.
INT29	30	R/W	‘b0	GPIO29 assert interrupt flag.
INT28	29	R/W	‘b0	GPIO28 assert interrupt flag.
INT27	28	R/W	‘b0	GPIO27 assert interrupt flag.
INT26	27	R/W	‘b0	GPIO26 assert interrupt flag.
INT25	26	R/W	‘b0	GPIO25 assert interrupt flag.
INT24	25	R/W	‘b0	GPIO24 assert interrupt flag.
INT23	24	R/W	‘b0	GPIO23 assert interrupt flag.
INT22	23	R/W	‘b0	GPIO22 assert interrupt flag.
INT21	22	R/W	‘b0	GPIO21 assert interrupt flag.
INT20	21	R/W	‘b0	GPIO20 assert interrupt flag.
INT19	20	R/W	‘b0	GPIO19 assert interrupt flag.
INT18	19	R/W	‘b0	GPIO18 assert interrupt flag.

INT17	18	R/W	‘b0	GPIO17 assert interrupt flag.
INT16	17	R/W	‘b0	GPIO16 assert interrupt flag.
INT15	16	R/W	‘b0	GPIO15 assert interrupt flag.
INT14	15	R/W	‘b0	GPIO14 assert interrupt flag.
INT13	14	R/W	‘b0	GPIO13 assert interrupt flag.
INT12	13	R/W	‘b0	GPIO12 assert interrupt flag.
INT11	12	R/W	‘b0	GPIO11 assert interrupt flag.
INT10	11	R/W	‘b0	GPIO10 assert interrupt flag.
INT9	10	R/W	‘b0	GPIO9 assert interrupt flag.
INT8	9	R/W	‘b0	GPIO8 assert interrupt flag.
INT7	8	R/W	‘b0	GPIO7 assert interrupt flag.
INT6	7	R/W	‘b0	GPIO6 assert interrupt flag.
INT5	6	R/W	‘b0	GPIO5 assert interrupt flag.
INT4	5	R/W	‘b0	GPIO4 assert interrupt flag.
INT3	4	R/W	‘b0	GPIO3 assert interrupt flag.
INT2	3	R/W	‘b0	GPIO2 assert interrupt flag.
INT1	2	R/W	‘b0	GPIO1 assert interrupt flag.
INT0	1	R/W	‘b0	GPIO0 assert interrupt flag.
write_data	0	W	-	1 to set, 0 to clear bits with 1.

Module::MIS	Register::UMSK_ISR_GP1A	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_B044

Name	**Bits**	**Read/Write**	**Reset State**	**Comments**
INT61	31	R/W	‘b0	GPIO61 assert interrupt flag.
INT60	30	R/W	‘b0	GPIO60 assert interrupt flag.
INT59	29	R/W	‘b0	GPIO59 assert interrupt flag.
INT58	28	R/W	‘b0	GPIO58 assert interrupt flag.
INT57	27	R/W	‘b0	GPIO57 assert interrupt flag.
INT56	26	R/W	‘b0	GPIO56 assert interrupt flag.
INT55	25	R/W	‘b0	GPIO55 assert interrupt flag.
INT54	24	R/W	‘b0	GPIO54 assert interrupt flag.
INT53	23	R/W	‘b0	GPIO53 assert interrupt flag.
INT52	22	R/W	‘b0	GPIO52 assert interrupt flag.
INT51	21	R/W	‘b0	GPIO51 assert interrupt flag.
INT50	20	R/W	‘b0	GPIO50 assert interrupt flag.
INT49	19	R/W	‘b0	GPIO49 assert interrupt flag.
INT48	18	R/W	‘b0	GPIO48 assert interrupt flag.
INT47	17	R/W	‘b0	GPIO47 assert interrupt flag.
INT46	16	R/W	‘b0	GPIO46 assert interrupt flag.
INT45	15	R/W	‘b0	GPIO45 assert interrupt flag.
INT44	14	R/W	‘b0	GPIO44 assert interrupt flag.
INT43	13	R/W	‘b0	GPIO43 assert interrupt flag.

INT42	12	R/W	‘b0	GPIO42 assert interrupt flag.
INT41	11	R/W	‘b0	GPIO41 assert interrupt flag.
INT40	10	R/W	‘b0	GPIO40 assert interrupt flag.
INT39	9	R/W	‘b0	GPIO39 assert interrupt flag.
INT38	8	R/W	‘b0	GPIO38 assert interrupt flag.
INT37	7	R/W	‘b0	GPIO37 assert interrupt flag.
INT36	6	R/W	‘b0	GPIO36 assert interrupt flag.
INT35	5	R/W	‘b0	GPIO35 assert interrupt flag.
INT34	4	R/W	‘b0	GPIO34 assert interrupt flag.
INT33	3	R/W	‘b0	GPIO33 assert interrupt flag.
INT32	2	R/W	‘b0	GPIO32 assert interrupt flag.
INT31	1	R/W	‘b0	GPIO31 assert interrupt flag.
write_data	0	W	-	1 to set, 0 to clear bits with 1.

Module::MIS	Register::UMSK_ISR_GP2A	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_B0A4

Name	**Bits**	**Read/Write**	**Reset State**	**Comments**
INT92	31	R/W	‘b0	GPIO92 assert interrupt flag.
INT91	30	R/W	‘b0	GPIO91 assert interrupt flag.
INT90	29	R/W	‘b0	GPIO90 assert interrupt flag.
INT89	28	R/W	‘b0	GPIO89 assert interrupt flag.
INT88	27	R/W	‘b0	GPIO88 assert interrupt flag.
INT87	26	R/W	‘b0	GPIO87 assert interrupt flag.
INT86	25	R/W	‘b0	GPIO86 assert interrupt flag.
INT85	24	R/W	‘b0	GPIO85 assert interrupt flag.
INT84	23	R/W	‘b0	GPIO84 assert interrupt flag.
INT83	22	R/W	‘b0	GPIO83 assert interrupt flag.
INT82	21	R/W	‘b0	GPIO82 assert interrupt flag.
INT81	20	R/W	‘b0	GPIO81 assert interrupt flag.
INT80	19	R/W	‘b0	GPIO80 assert interrupt flag.
INT79	18	R/W	‘b0	GPIO79 assert interrupt flag.
INT78	17	R/W	‘b0	GPIO78 assert interrupt flag.
INT77	16	R/W	‘b0	GPIO77 assert interrupt flag.
INT76	15	R/W	‘b0	GPIO76 assert interrupt flag.
INT75	14	R/W	‘b0	GPIO75 assert interrupt flag.
INT74	13	R/W	‘b0	GPIO74 assert interrupt flag.
INT73	12	R/W	‘b0	GPIO73 assert interrupt flag.
INT72	11	R/W	‘b0	GPIO72 assert interrupt flag.
INT71	10	R/W	‘b0	GPIO71 assert interrupt flag.
INT70	9	R/W	‘b0	GPIO70 assert interrupt flag.
INT69	8	R/W	‘b0	GPIO69 assert interrupt flag.
INT68	7	R/W	‘b0	GPIO68 assert interrupt flag.
INT67	6	R/W	‘b0	GPIO67 assert interrupt flag.

INT66	5	R/W	‘b0	GPIO66 assert interrupt flag.
INT65	4	R/W	‘b0	GPIO65 assert interrupt flag.
INT64	3	R/W	‘b0	GPIO64 assert interrupt flag.
INT63	2	R/W	‘b0	GPIO63 assert interrupt flag.
INT62	1	R/W	‘b0	GPIO62 assert interrupt flag.
write_data	0	W	-	1 to set, 0 to clear bits with 1.

Module::MIS	Register::UMSK_ISR_GP3A	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_B0B8

Name	**Bits**	**Read/Write**	**Reset State**	**Comments**
Rvd	31:9	-	-	-
INT100	8	R/W	‘b0	GPIO100 assert interrupt flag.
INT99	7	R/W	‘b0	GPIO99 assert interrupt flag.
INT98	6	R/W	‘b0	GPIO98 assert interrupt flag.
INT97	5	R/W	‘b0	GPIO97 assert interrupt flag.
INT96	4	R/W	‘b0	GPIO96 assert interrupt flag.
INT95	3	R/W	‘b0	GPIO95 assert interrupt flag.
INT94	2	R/W	‘b0	GPIO94 assert interrupt flag.
INT93	1	R/W	‘b0	GPIO93 assert interrupt flag.
write_data	0	W	-	1 to set, 0 to clear bits with 1.

Module::MIS	Register::UMSK_ISR_GP0DA	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_B054

Name	**Bits**	**Read/Write**	**Reset State**	**Comments**
INT30	31	R/W	‘b0	GPIO30 dis-assert interrupt flag.
INT29	30	R/W	‘b0	GPIO29 dis-assert interrupt flag.
INT28	29	R/W	‘b0	GPIO28 dis-assert interrupt flag.
INT27	28	R/W	‘b0	GPIO27 dis-assert interrupt flag.
INT26	27	R/W	‘b0	GPIO26 dis-assert interrupt flag.
INT25	26	R/W	‘b0	GPIO25 dis-assert interrupt flag.
INT24	25	R/W	‘b0	GPIO24 dis-assert interrupt flag.
INT23	24	R/W	‘b0	GPIO23 dis-assert interrupt flag.
INT22	23	R/W	‘b0	GPIO22 dis-assert interrupt flag.
INT21	22	R/W	‘b0	GPIO21 dis-assert interrupt flag.
INT20	21	R/W	‘b0	GPIO20 dis-assert interrupt flag.

INT19	20	R/W	‘b0	GPIO19 dis-assert interrupt flag.
INT18	19	R/W	‘b0	GPIO18 dis-assert interrupt flag.
INT17	18	R/W	‘b0	GPIO17 dis-assert interrupt flag.
INT16	17	R/W	‘b0	GPIO16 dis-assert interrupt flag.
INT15	16	R/W	‘b0	GPIO15 dis-assert interrupt flag.
INT14	15	R/W	‘b0	GPIO14 dis-assert interrupt flag.
INT13	14	R/W	‘b0	GPIO13 dis-assert interrupt flag.
INT12	13	R/W	‘b0	GPIO12 dis-assert interrupt flag.
INT11	12	R/W	‘b0	GPIO11 dis-assert interrupt flag.
INT10	11	R/W	‘b0	GPIO10 dis-assert interrupt flag.
INT9	10	R/W	‘b0	GPIO9 dis-assert interrupt flag.
INT8	9	R/W	‘b0	GPIO8 dis-assert interrupt flag.
INT7	8	R/W	‘b0	GPIO7 dis-assert interrupt flag.
INT6	7	R/W	‘b0	GPIO6 dis-assert interrupt flag.
INT5	6	R/W	‘b0	GPIO5 dis-assert interrupt flag.
INT4	5	R/W	‘b0	GPIO4 dis-assert interrupt flag.
INT3	4	R/W	‘b0	GPIO3 dis-assert interrupt flag.
INT2	3	R/W	‘b0	GPIO2 dis-assert interrupt flag.
INT1	2	R/W	‘b0	GPIO1 dis-assert interrupt flag.
INT0	1	R/W	‘b0	GPIO0 dis-assert interrupt flag.
write_data	0	W	-	1 to set, 0 to clear bits with 1.

Module::MIS	Register::UMSK_ISR_GP1DA	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_B058

Name	**Bits**	**Read/Write**	**Reset State**	**Comments**
INT61	31	R/W	‘b0	GPIO61 dis-assert interrupt flag.
INT60	30	R/W	‘b0	GPIO60 dis-assert interrupt flag.
INT59	29	R/W	‘b0	GPIO59 dis-assert interrupt flag.
INT58	28	R/W	‘b0	GPIO58 dis-assert interrupt flag.
INT57	27	R/W	‘b0	GPIO57 dis-assert interrupt flag.
INT56	26	R/W	‘b0	GPIO56 dis-assert interrupt flag.
INT55	25	R/W	‘b0	GPIO55 dis-assert interrupt flag.
INT54	24	R/W	‘b0	GPIO54 dis-assert interrupt flag.
INT53	23	R/W	‘b0	GPIO53 dis-assert interrupt flag.
INT52	22	R/W	‘b0	GPIO52 dis-assert interrupt flag.
INT51	21	R/W	‘b0	GPIO51 dis-assert interrupt flag.
INT50	20	R/W	‘b0	GPIO50 dis-assert interrupt flag.
INT49	19	R/W	‘b0	GPIO49 dis-assert interrupt flag.
INT48	18	R/W	‘b0	GPIO48 dis-assert interrupt flag.
INT47	17	R/W	‘b0	GPIO47 dis-assert interrupt flag.
INT46	16	R/W	‘b0	GPIO46 dis-assert interrupt flag.
INT45	15	R/W	‘b0	GPIO45 dis-assert interrupt flag.

INT44	14	R/W	‘b0	GPIO44 dis-assert interrupt flag.
INT43	13	R/W	‘b0	GPIO43 dis-assert interrupt flag.
INT42	12	R/W	‘b0	GPIO42 dis-assert interrupt flag.
INT41	11	R/W	‘b0	GPIO41 dis-assert interrupt flag.
INT40	10	R/W	‘b0	GPIO40 dis-assert interrupt flag.
INT39	9	R/W	‘b0	GPIO39 dis-assert interrupt flag.
INT38	8	R/W	‘b0	GPIO38 dis-assert interrupt flag.
INT37	7	R/W	‘b0	GPIO37 dis-assert interrupt flag.
INT36	6	R/W	‘b0	GPIO36 dis-assert interrupt flag.
INT35	5	R/W	‘b0	GPIO35 dis-assert interrupt flag.
INT34	4	R/W	‘b0	GPIO34 dis-assert interrupt flag.
INT33	3	R/W	‘b0	GPIO33 dis-assert interrupt flag.
INT32	2	R/W	‘b0	GPIO32 dis-assert interrupt flag.
INT31	1	R/W	‘b0	GPIO31 dis-assert interrupt flag.
write_data	0	W	-	1 to set, 0 to clear bits with 1.

Module::MIS	Register::UMSK_ISR_GP2DA	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_B0A8

Name	**Bits**	**Read/Write**	**Reset State**	**Comments**
INT92	31	R/W	‘b0	GPIO92 dis-assert interrupt flag.
INT91	30	R/W	‘b0	GPIO91 dis-assert interrupt flag.
INT90	29	R/W	‘b0	GPIO90 dis-assert interrupt flag.
INT89	28	R/W	‘b0	GPIO89 dis-assert interrupt flag.
INT88	27	R/W	‘b0	GPIO88 dis-assert interrupt flag.
INT87	26	R/W	‘b0	GPIO87 dis-assert interrupt flag.
INT86	25	R/W	‘b0	GPIO86 dis-assert interrupt flag.
INT85	24	R/W	‘b0	GPIO85 dis-assert interrupt flag.
INT84	23	R/W	‘b0	GPIO84 dis-assert interrupt flag.
INT83	22	R/W	‘b0	GPIO83 dis-assert interrupt flag.
INT82	21	R/W	‘b0	GPIO82 dis-assert interrupt flag.
INT81	20	R/W	‘b0	GPIO81 dis-assert interrupt flag.
INT80	19	R/W	‘b0	GPIO80 dis-assert interrupt flag.
INT79	18	R/W	‘b0	GPIO79 dis-assert interrupt flag.
INT78	17	R/W	‘b0	GPIO78 dis-assert interrupt flag.
INT77	16	R/W	‘b0	GPIO77 dis-assert interrupt flag.
INT76	15	R/W	‘b0	GPIO76 dis-assert interrupt flag.
INT75	14	R/W	‘b0	GPIO75 dis-assert interrupt flag.
INT74	13	R/W	‘b0	GPIO74 dis-assert interrupt flag.
INT73	12	R/W	‘b0	GPIO73 dis-assert interrupt flag.
INT72	11	R/W	‘b0	GPIO72 dis-assert interrupt flag.
INT71	10	R/W	‘b0	GPIO71 dis-assert interrupt flag.
INT70	9	R/W	‘b0	GPIO70 dis-assert interrupt flag.

INT69	8	R/W	‘b0	GPIO69 dis-assert interrupt flag.
INT68	7	R/W	‘b0	GPIO68 dis-assert interrupt flag.
INT67	6	R/W	‘b0	GPIO67 dis-assert interrupt flag.
INT66	5	R/W	‘b0	GPIO66 dis-assert interrupt flag.
INT65	4	R/W	‘b0	GPIO65 dis-assert interrupt flag.
INT64	3	R/W	‘b0	GPIO64 dis-assert interrupt flag.
INT63	2	R/W	‘b0	GPIO63 dis-assert interrupt flag.
INT62	1	R/W	‘b0	GPIO62 dis-assert interrupt flag.
write_data	0	W	-	1 to set, 0 to clear bits with 1.

Module::MIS	Register::UMSK_ISR_GP3DA	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_B0BC

Name	**Bits**	**Read/Write**	**Reset State**	**Comments**
Rvd	31:9	-	-	-
INT100	8	R/W	‘b0	GPIO100 dis-assert interrupt flag.
INT99	7	R/W	‘b0	GPIO99 dis-assert interrupt flag.
INT98	6	R/W	‘b0	GPIO98 dis-assert interrupt flag.
INT97	5	R/W	‘b0	GPIO97 dis-assert interrupt flag.
INT96	4	R/W	‘b0	GPIO96 dis-assert interrupt flag.
INT95	3	R/W	‘b0	GPIO95 dis-assert interrupt flag.
INT94	2	R/W	‘b0	GPIO94 dis-assert interrupt flag.
INT93	1	R/W	‘b0	GPIO93 dis-assert interrupt flag.
write_data	0	W	-	1 to set, 0 to clear bits with 1.

Module::MIS	Register::DUMMY1	Set::1	ATTR::nor	Type::SR	ADDR::0x9801_B068

Name	**Bits**	**Read/Write**	**Reset State**	**Comments**
write_enable2	31	W	-	-
Rvd2	30:16	R/W	‘h3fff	
write_enable1	15	W	-	-
Rvd1	14:0	R/W	‘h0	

Module::MIS	Register:: UR_CTRL	Set::1	ATTR::nor	Type::SR	**ADDR::0x9801_B060**

Name	**Bits**	**Read/Write**	**Reset State**	**Comments**
Rvd3	31:29	R/W	‘h0	-
Rvd	28	-	-	-
to_len	27:20	R/W	‘h4	timeout length. UART RX timeout condition: No characters in or out of the RX FIFO during

				the last n character times and there is at least 1 character in it during this time ex: 8'h4: timeout will assert after 4 character time. 8'hF: timeout will assert after 15 character time.
to_int_en	19	R/W	'b0	1'b1: Timeout interrupt enable.
TOAUAREQ	18	R/W	'b0	Timeout Auto Assert Request to MD if rx_fifo_waterlevel > 0 Byte. 1'b1: Auto assert request to MD. 1'b0: Don't auto assert request to MD.
FLOW_POL	17	R/W	'b0	UR1 Flow control signal (CTS/RST) polarity. 1'b0 : low active. 1'b1 : high active.
MDIFEN	16	R/W	'b0	MD Interface Enable. Enable MD access UART TX/RX FIFO. 1'b1: UR-MD IF Enable. 1'b0: UR-MD IF Disable.
Rvd	15:14	R/W	-	-
TXEMPTHR	13:8	R/W	'h10	TX empty threshold for UR-MD IF. IF UR1 TX FIFO have TXEMPTHR bytes free space, issue read request to MD. 6'D8: 8 Bytes 6'D16: 16 Bytes 6'D24: 24 Bytes 6'D32: 32 Bytes
Rvd	7:6	R/W	-	-
RXFULTHR	5:0	R/W	'h10	RX full threshold for UR-MD IF. If UR1 RX receive RXFULTHR bytes data, UR1 will issue write request to MD. 6'D8: 8 Bytes 6'D16: 16 Bytes 6'D24: 24 Bytes 31 Bytes and 32 Bytes are unsupported.

p.s1: UR_CTRL is used for high speed UART (UART 1).

p.s2:

a. We can config RX FIFO threshold for rts_n (rts_n =1 : rx fifo is almost full，request "stop" to transfer data to Saturn.UR1.RX)

Threshold of UR DW IP, RX FIFO as following:（at 0x9801_BC08[7:6]）

00 = 1 character in the FIFO
01 = FIFO ¼ full
10 = FIFO ½ full
11 = FIFO 2 less than full

b. config UR DW RX FIFO threshold for issue request to MD (at 0x9801_B070[5:0])
support range: 1 bytes ~ 30 bytes.
Un_support range: 31 byte ~ 32 byte.

If we config 31 bytes or 32 bytes, DW UR IP will issue issue rts_n after receive 30 bytes(config 0x9801_BC08[7:6]=2'b11). Saturn UR will not receive rx data.
Then, time out assert,
If enable MD transfer at timeout, MD transfer 30 bytes data.
If dis-able MD transfer at timeout, timeout ISR will assert. Need CPU help to read RX data.
Then rts_n de-assert. Saturn UR will continur to receive rx data.

Module::MIS	Register:: UR2_CTRL	Set::1	ATTR::nor	Type::SR	**ADDR::0x9801_B064**

Name	**Bits**	**Read/Write**	**Reset State**	**Comments**
Rvd3	31:29	R/W	'h0	-
Rvd	28	-	-	-
to_len	27:20	R/W	'h4	timeout length. UART RX timeout condition: No characters in or out of the RX FIFO during the last n character times and there is at least 1 character in it during this time ex: 8'h4: timeout will assert after 4 character time. 8'hF: timeout will assert after 15 character time.
to_int_en	19	R/W	'b0	1'b1: Timeout interrupt enable.
TOAUAREQ	18	R/W	'b0	Timeout Auto Assert Request to MD if rx_fifo_waterlevel > 0 Byte. 1'b1: Auto assert request to MD. 1'b0: Don't auto assert request to MD.
FLOW_POL	17	R/W	'b0	UR2 Flow control signal (CTS/RST) polarity. 1'b0 : low active. 1'b1 : high active.
MDIFEN	16	R/W	'b0	MD Interface Enable. Enable MD access UART TX/RX FIFO. 1'b1: UR-MD IF Enable. 1'b0: UR-MD IF Disable.
Rvd	15:14	R/W	-	-

TXEMPTHR	13:8	R/W	‘h10	TX empty threshold for UR-MD IF. IF UR2 TX FIFO have TXEMPTHR bytes free space, issue read request to MD. 6’D8: 8 Bytes 6’D16: 16 Bytes 6’D24: 24 Bytes 6’D32: 32 Bytes
Rvd	7:6	R/W	-	-
RXFULTHR	5:0	R/W	‘h10	RX full threshold for UR-MD IF. If UR2 RX receive RXFULTHR bytes data, UR2 will issue write request to MD. 6’D8: 8 Bytes 6’D16: 16 Bytes 6’D24: 24 Bytes 31 Bytes and 32 Bytes are unsupported.

Module::MIS	Register::FAST_ISR_GPIO0_A	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_B070

Name	Bits	Read/Write	Reset State	Comments
int_30	31	R/W	‘b0	GPIO30 assert interrupt flag.
int_29	30	R/W	‘b0	GPIO29 assert interrupt flag.
int_28	29	R/W	‘b0	GPIO28 assert interrupt flag.
int_27	28	R/W	‘b0	GPIO27 assert interrupt flag.
int_26	27	R/W	‘b0	GPIO26 assert interrupt flag.
int_25	26	R/W	‘b0	GPIO25 assert interrupt flag.
int_24	25	R/W	‘b0	GPIO24 assert interrupt flag.
int_23	24	R/W	‘b0	GPIO23 assert interrupt flag.
int_22	23	R/W	‘b0	GPIO22 assert interrupt flag.
int_21	22	R/W	‘b0	GPIO21 assert interrupt flag.
int_20	21	R/W	‘b0	GPIO20 assert interrupt flag.
int_19	20	R/W	‘b0	GPIO19 assert interrupt flag.
int_18	19	R/W	‘b0	GPIO18 assert interrupt flag.
int_17	18	R/W	‘b0	GPIO17 assert interrupt flag.
int_16	17	R/W	‘b0	GPIO16 assert interrupt flag.
int_15	16	R/W	‘b0	GPIO15 assert interrupt flag.
int_14	15	R/W	‘b0	GPIO14 assert interrupt flag.
int_13	14	R/W	‘b0	GPIO13 assert interrupt flag.
int_12	13	R/W	‘b0	GPIO12 assert interrupt flag.
int_11	12	R/W	‘b0	GPIO11 assert interrupt flag.
int_10	11	R/W	‘b0	GPIO10 assert interrupt flag.
int_9	10	R/W	‘b0	GPIO9 assert interrupt flag.
int_8	9	R/W	‘b0	GPIO8 assert interrupt flag.
int_7	8	R/W	‘b0	GPIO7 assert interrupt flag.
int_6	7	R/W	‘b0	GPIO6 assert interrupt flag.
int_5	6	R/W	‘b0	GPIO5 assert interrupt flag.
int_4	5	R/W	‘b0	GPIO4 assert interrupt flag.
int_3	4	R/W	‘b0	GPIO3 assert interrupt flag.
int_2	3	R/W	‘b0	GPIO2 assert interrupt flag.
int_1	2	R/W	‘b0	GPIO1 assert interrupt flag.
int_0	1	R/W	‘b0	GPIO0 assert interrupt flag.
write_data	0	W	-	1 to set, 0 to clear bits with 1.

Module::MIS	Register::FAST_ISR_GPIO1_A	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_B074

Name	Bits	Read/Write	Reset State	Comments
int_61	31	R/W	‘b0	GPIO61 assert interrupt flag.
int_60	30	R/W	‘b0	GPIO60 assert interrupt flag.

int_59	29	R/W	‘b0	GPIO59 assert interrupt flag.
int_58	28	R/W	‘b0	GPIO58 assert interrupt flag.
int_57	27	R/W	‘b0	GPIO57 assert interrupt flag.
int_56	26	R/W	‘b0	GPIO56 assert interrupt flag.
int_55	25	R/W	‘b0	GPIO55 assert interrupt flag.
int_54	24	R/W	‘b0	GPIO54 assert interrupt flag.
int_53	23	R/W	‘b0	GPIO53 assert interrupt flag.
int_52	22	R/W	‘b0	GPIO52 assert interrupt flag.
int_51	21	R/W	‘b0	GPIO51 assert interrupt flag.
int_50	20	R/W	‘b0	GPIO50 assert interrupt flag.
int_49	19	R/W	‘b0	GPIO49 assert interrupt flag.
int_48	18	R/W	‘b0	GPIO48 assert interrupt flag.
int_47	17	R/W	‘b0	GPIO47 assert interrupt flag.
int_46	16	R/W	‘b0	GPIO46 assert interrupt flag.
int_45	15	R/W	‘b0	GPIO45 assert interrupt flag.
int_44	14	R/W	‘b0	GPIO44 assert interrupt flag.
int_43	13	R/W	‘b0	GPIO43 assert interrupt flag.
int_42	12	R/W	‘b0	GPIO42 assert interrupt flag.
int_41	11	R/W	‘b0	GPIO41 assert interrupt flag.
int_40	10	R/W	‘b0	GPIO40 assert interrupt flag.
int_39	9	R/W	‘b0	GPIO39 assert interrupt flag.
int_38	8	R/W	‘b0	GPIO38 assert interrupt flag.
int_37	7	R/W	‘b0	GPIO37 assert interrupt flag.
int_36	6	R/W	‘b0	GPIO36 assert interrupt flag.
int_35	5	R/W	‘b0	GPIO35 assert interrupt flag.
int_34	4	R/W	‘b0	GPIO34 assert interrupt flag.
int_33	3	R/W	‘b0	GPIO33 assert interrupt flag.
int_32	2	R/W	‘b0	GPIO32 assert interrupt flag.
int_31	1	R/W	‘b0	GPIO31 assert interrupt flag.
write_data	0	W	-	1 to set, 0 to clear bits with 1.

Module::MIS	Register::FAST_ISR_GPIO2_A	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_B0AC

Name	Bits	Read/Write	Reset State	Comments
int_92	31	R/W	'b0	GPIO92 assert interrupt flag.
int_91	30	R/W	'b0	GPIO91 assert interrupt flag.
int_90	29	R/W	'b0	GPIO90 assert interrupt flag.
int_89	28	R/W	'b0	GPIO89 assert interrupt flag.
int_88	27	R/W	'b0	GPIO88 assert interrupt flag.
int_87	26	R/W	'b0	GPIO87 assert interrupt flag.
int_86	25	R/W	'b0	GPIO86 assert interrupt flag.
int_85	24	R/W	'b0	GPIO85 assert interrupt flag.
int_84	23	R/W	'b0	GPIO84 assert interrupt flag.
int_83	22	R/W	'b0	GPIO83 assert interrupt flag.
int_82	21	R/W	'b0	GPIO82 assert interrupt flag.
int_81	20	R/W	'b0	GPIO81 assert interrupt flag.
int_80	19	R/W	'b0	GPIO80 assert interrupt flag.
int_79	18	R/W	'b0	GPIO79 assert interrupt flag.
int_78	17	R/W	'b0	GPIO78 assert interrupt flag.
int_77	16	R/W	'b0	GPIO77 assert interrupt flag.
int_76	15	R/W	'b0	GPIO76 assert interrupt flag.
int_75	14	R/W	'b0	GPIO75 assert interrupt flag.
int_74	13	R/W	'b0	GPIO74 assert interrupt flag.
int_73	12	R/W	'b0	GPIO73 assert interrupt flag.
int_72	11	R/W	'b0	GPIO72 assert interrupt flag.
int_71	10	R/W	'b0	GPIO71 assert interrupt flag.
int_70	9	R/W	'b0	GPIO70 assert interrupt flag.
int_69	8	R/W	'b0	GPIO69 assert interrupt flag.
int_68	7	R/W	'b0	GPIO68 assert interrupt flag.
int_67	6	R/W	'b0	GPIO67 assert interrupt flag.
int_66	5	R/W	'b0	GPIO66 assert interrupt flag.
int_65	4	R/W	'b0	GPIO65 assert interrupt flag.
int_64	3	R/W	'b0	GPIO64 assert interrupt flag.
int_63	2	R/W	'b0	GPIO63 assert interrupt flag.
int_62	1	R/W	'b0	GPIO62 assert interrupt flag.
write_data	0	W	-	1 to set, 0 to clear bits with 1.

Module::MIS	Register::FAST_ISR_GPIO3_A	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_B0C0

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:9	-	-	-
int_100	8	R/W	‘b0	GPIO100 assert interrupt flag.
int_99	7	R/W	‘b0	GPIO99 assert interrupt flag.
int_98	6	R/W	‘b0	GPIO98 assert interrupt flag.
int_97	5	R/W	‘b0	GPIO97 assert interrupt flag.
int_96	4	R/W	‘b0	GPIO96 assert interrupt flag.
int_95	3	R/W	‘b0	GPIO95 assert interrupt flag.
int_94	2	R/W	‘b0	GPIO94 assert interrupt flag.
int_93	1	R/W	‘b0	GPIO93 assert interrupt flag.
write_data	0	W	-	1 to set, 0 to clear bits with 1.

Module::MIS	Register::FAST_ISR_GPIO0_DA		Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_B078

Name	Bits	Read/Write	Reset State	Comments
int_30	31	R/W	‘b0	GPIO30 de-assert interrupt flag.
int_29	30	R/W	‘b0	GPIO29 de-assert interrupt flag.
int_28	29	R/W	‘b0	GPIO28 de-assert interrupt flag.
int_27	28	R/W	‘b0	GPIO27 de-assert interrupt flag.
int_26	27	R/W	‘b0	GPIO26 de-assert interrupt flag.
int_25	26	R/W	‘b0	GPIO25 de-assert interrupt flag.
int_24	25	R/W	‘b0	GPIO24 de-assert interrupt flag.
int_23	24	R/W	‘b0	GPIO23 de-assert interrupt flag.
int_22	23	R/W	‘b0	GPIO22 de-assert interrupt flag.
int_21	22	R/W	‘b0	GPIO21 de-assert interrupt flag.
int_20	21	R/W	‘b0	GPIO20 de-assert interrupt flag.
int_19	20	R/W	‘b0	GPIO19 de-assert interrupt flag.
int_18	19	R/W	‘b0	GPIO18 de-assert interrupt flag.
int_17	18	R/W	‘b0	GPIO17 de-assert interrupt flag.
int_16	17	R/W	‘b0	GPIO16 de-assert interrupt flag.
int_15	16	R/W	‘b0	GPIO15 de-assert interrupt flag.
int_14	15	R/W	‘b0	GPIO14 de-assert interrupt flag.
int_13	14	R/W	‘b0	GPIO13 de-assert interrupt flag.
int_12	13	R/W	‘b0	GPIO12 de-assert interrupt flag.
int_11	12	R/W	‘b0	GPIO11 de-assert interrupt flag.
int_10	11	R/W	‘b0	GPIO10 de-assert interrupt flag.
int_9	10	R/W	‘b0	GPIO9 de-assert interrupt flag.
int_8	9	R/W	‘b0	GPIO8 de-assert interrupt flag.
int_7	8	R/W	‘b0	GPIO7 de-assert interrupt flag.
int_6	7	R/W	‘b0	GPIO6 de-assert interrupt flag.
int_5	6	R/W	‘b0	GPIO5 de-assert interrupt flag.
int_4	5	R/W	‘b0	GPIO4 de-assert interrupt flag.
int_3	4	R/W	‘b0	GPIO3 de-assert interrupt flag.
int_2	3	R/W	‘b0	GPIO2 de-assert interrupt flag.
int_1	2	R/W	‘b0	GPIO1 de-assert interrupt flag.
int_0	1	R/W	‘b0	GPIO0 de-assert interrupt flag.
write_data	0	W	-	1 to set, 0 to clear bits with 1.

Module::MIS	Register::FAST_ISR_GPIO1_DA		Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_B07C

Name	Bits	Read/Write	Reset State	Comments
int_61	31	R/W	‘b0	GPIO61 de-assert interrupt flag.
int_60	30	R/W	‘b0	GPIO60 de-assert interrupt flag.

int_59	29	R/W	‘b0	GPIO59 de-assert interrupt flag.
int_58	28	R/W	‘b0	GPIO58 de-assert interrupt flag.
int_57	27	R/W	‘b0	GPIO57 de-assert interrupt flag.
int_56	26	R/W	‘b0	GPIO56 de-assert interrupt flag.
int_55	25	R/W	‘b0	GPIO55 de-assert interrupt flag.
int_54	24	R/W	‘b0	GPIO54 de-assert interrupt flag.
int_53	23	R/W	‘b0	GPIO53 de-assert interrupt flag.
int_52	22	R/W	‘b0	GPIO52 de-assert interrupt flag.
int_51	21	R/W	‘b0	GPIO51 de-assert interrupt flag.
int_50	20	R/W	‘b0	GPIO50 de-assert interrupt flag.
int_49	19	R/W	‘b0	GPIO49 de-assert interrupt flag.
int_48	18	R/W	‘b0	GPIO48 de-assert interrupt flag.
int_47	17	R/W	‘b0	GPIO47 de-assert interrupt flag.
int_46	16	R/W	‘b0	GPIO46 de-assert interrupt flag.
int_45	15	R/W	‘b0	GPIO45 de-assert interrupt flag.
int_44	14	R/W	‘b0	GPIO44 de-assert interrupt flag.
int_43	13	R/W	‘b0	GPIO43 de-assert interrupt flag.
int_42	12	R/W	‘b0	GPIO42 de-assert interrupt flag.
int_41	11	R/W	‘b0	GPIO41 de-assert interrupt flag.
int_40	10	R/W	‘b0	GPIO40 de-assert interrupt flag.
int_39	9	R/W	‘b0	GPIO39 de-assert interrupt flag.
int_38	8	R/W	‘b0	GPIO38 de-assert interrupt flag.
int_37	7	R/W	‘b0	GPIO37 de-assert interrupt flag.
int_36	6	R/W	‘b0	GPIO36 de-assert interrupt flag.
int_35	5	R/W	‘b0	GPIO35 de-assert interrupt flag.
int_34	4	R/W	‘b0	GPIO34 de-assert interrupt flag.
int_33	3	R/W	‘b0	GPIO33 de-assert interrupt flag.
int_32	2	R/W	‘b0	GPIO32 de-assert interrupt flag.
int_31	1	R/W	‘b0	GPIO31 de-assert interrupt flag.
write_data	0	W	-	1 to set, 0 to clear bits with 1.

Module::MIS	Register::FAST_ISR_GPIO2_DA		Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_B0B0

Name	Bits	Read/Write	Reset State	Comments
int_92	31	R/W	'b0	GPIO92 dis-assert interrupt flag.
int_91	30	R/W	'b0	GPIO91 dis-assert interrupt flag.
int_90	29	R/W	'b0	GPIO90 dis-assert interrupt flag.
int_89	28	R/W	'b0	GPIO89 dis-assert interrupt flag.
int_88	27	R/W	'b0	GPIO88 dis-assert interrupt flag.
int_87	26	R/W	'b0	GPIO87 dis-assert interrupt flag.
int_86	25	R/W	'b0	GPIO86 dis-assert interrupt flag.
int_85	24	R/W	'b0	GPIO85 dis-assert interrupt flag.
int_84	23	R/W	'b0	GPIO84 dis-assert interrupt flag.
int_83	22	R/W	'b0	GPIO83 dis-assert interrupt flag.
int_82	21	R/W	'b0	GPIO82 dis-assert interrupt flag.
int_81	20	R/W	'b0	GPIO81 dis-assert interrupt flag.
int_80	19	R/W	'b0	GPIO80 dis-assert interrupt flag.
int_79	18	R/W	'b0	GPIO79 dis-assert interrupt flag.
int_78	17	R/W	'b0	GPIO78 dis-assert interrupt flag.
int_77	16	R/W	'b0	GPIO77 dis-assert interrupt flag.
int_76	15	R/W	'b0	GPIO76 dis-assert interrupt flag.
int_75	14	R/W	'b0	GPIO75 dis-assert interrupt flag.
int_74	13	R/W	'b0	GPIO74 dis-assert interrupt flag.
int_73	12	R/W	'b0	GPIO73 dis-assert interrupt flag.
int_72	11	R/W	'b0	GPIO72 dis-assert interrupt flag.
int_71	10	R/W	'b0	GPIO71 dis-assert interrupt flag.
int_70	9	R/W	'b0	GPIO70 dis-assert interrupt flag.
int_69	8	R/W	'b0	GPIO69 dis-assert interrupt flag.
int_68	7	R/W	'b0	GPIO68 dis-assert interrupt flag.
int_67	6	R/W	'b0	GPIO67 dis-assert interrupt flag.
int_66	5	R/W	'b0	GPIO66 dis-assert interrupt flag.
int_65	4	R/W	'b0	GPIO65 dis-assert interrupt flag.
int_64	3	R/W	'b0	GPIO64 dis-assert interrupt flag.
int_63	2	R/W	'b0	GPIO63 dis-assert interrupt flag.
int_62	1	R/W	'b0	GPIO62 dis-assert interrupt flag.
write_data	0	W	-	1 to set, 0 to clear bits with 1.

Module::MIS	Register::FAST_ISR_GPIO3_DA	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_B0C4

Name	**Bits**	**Read/Write**	**Reset State**	**Comments**
Rvd	31:9	-	-	-
int_100	8	R/W	‘b0	GPIO100 dis-assert interrupt flag.
int_99	7	R/W	‘b0	GPIO99 dis-assert interrupt flag.
int_98	6	R/W	‘b0	GPIO98 dis-assert interrupt flag.
int_97	5	R/W	‘b0	GPIO97 dis-assert interrupt flag.
int_96	4	R/W	‘b0	GPIO96 dis-assert interrupt flag.
int_95	3	R/W	‘b0	GPIO95 dis-assert interrupt flag.
int_94	2	R/W	‘b0	GPIO94 dis-assert interrupt flag.
int_93	1	R/W	‘b0	GPIO93 dis-assert interrupt flag.
write_data	0	W	-	1 to set, 0 to clear bits with 1.

Module::MIS	Register::SCPU_INT_EN	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B080

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:30	-	-	-
FAN	29	R/W	‘b1	1’b1: Enable DC FAN interrupt to SCPU.
I2C3	28	R/W	‘b1	1’b1: Enable Third i2c interrupt to SCPU.
GSPI	27	R/W	‘b1	1’b1: Enable Second gspi interrupt to SCPU.
I2C2	26	R/W	‘b1	1’b1: Enable Second i2c interrupt to SCPU.
Rvd	25	-	-	-
SC0	24	R/W	‘b1	1’b1: Enable smart card0 interrupt to SCPU.
Rvd	23	-	-	-
LSADC1	22	R/W	‘b1	1’b1: Enable LSADC1 interrupt to SCPU
LSADC0	21	R/W	‘b1	1’b1: Enable LSADC0 interrupt to SCPU
GPIODA	20	R/W	‘b1	1’b1: Enable GPIO[59:0] dis-assert interrupt to SCPU.
GPIOA	19	R/W	‘b1	1’b1: Enable GPIO[59:0] assert interrupt to SCPU.
Rvd	18:16	-	-	-
I2C4	15	R/W	‘b1	1’b1: Enable Forth i2c interrupt to SCPU.
I2C5	14	R/W	‘b1	1’b1: Enable Fivth i2c interrupt to SCPU.
Rvd	13	-	-	-
RTC_DATE	12	R/W	‘b1	1’b1: Enable RTC date interrupt to SCPU.
RTC_HOUR	11	R/W	‘b1	1’b1: Enable RTC hour interrupt to SCPU.

RTC_MIN	10	R/W	‘b1	1’b1: Enable RTC minute interrupt to SCPU.
Rvd	9:8	-	-	-
UR2	7	R/W	‘b1	1’b1: Enable uart1 interrupt to SCPU.
UR2_TO	6	R/W	‘b1	1’b1: Enable uart2 timeout interrupt to SCPU.
UR1_TO	5	R/W	‘b1	1’b1: Enable uart1 timeout interrupt to SCPU.
Rvd	4	-	-	-
UR1	3	R/W	‘b1	1’b1: Enable uart1 interrupt to SCPU.
Rvd	2:0	-	-	-

Module::MIS	Register:: I2C2_SDA_DEL	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B08c

Name	**Bits**	**Read/Write**	**Reset State**	**Comments**
dmy1	31:12	R/W	‘hff000	-
Rvd	11:9	R/W	-	-
en	8	R/W	‘b0	SDA data phase delay enable.
Rvd	7:5	R/W	-	-
sel	4:0	R/W	‘h1	SDA data phase delay . sel[4:0] / SDA delay time (unit : ns) 1 / 518 2 / 1036 3 / 1554 4 / 2072 5 / 2590 6 / 3108 7 / 3626 8 / 4144 9 / 4662 10 / 5180 11 / 5698 12 / 6216 13 / 6734 14 / 7252 15 / 7770

				16 8288 17 8806 18 9324 19 9842 20 10360 21 10878 22 11396 23 11914 others: reserved

Module::MIS	Register:: I2C3_SDA_DEL	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B090

Name	**Bits**	**Read/Write**	**Reset State**	**Comments**
dmy1	31:12	R/W	‘hff000	-
Rvd	11:9	R/W	-	-
en	8	R/W	‘b0	SDA data phase delay enable.
Rvd	7:5	R/W	-	-
sel	4:0	R/W	‘h1	SDA data phase delay . sel[4:0] / SDA delay time (unit : ns) 1 518 2 1036 3 1554 4 2072 5 2590 6 3108 7 3626 8 4144 9 4662 10 5180 11 5698 12 6216 13 6734 14 7252 15 7770 16 8288

				17	8806
				18	9324
				19	9842
				20	10360
				21	10878
				22	11396
				23	11914
				others: reserved	

Module::MIS	Register:: I2C4_SDA_DEL	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B094

Name	**Bits**	**Read/Write**	**Reset State**	**Comments**
dmy1	31:12	R/W	‘hff000	-
Rvd	11:9	R/W	-	-
en	8	R/W	‘b0	SDA data phase delay enable.
Rvd	7:5	R/W	-	-
sel	4:0	R/W	‘h1	SDA data phase delay .

sel[4:0]	SDA delay time (unit : ns)
1	518
2	1036
3	1554
4	2072
5	2590
6	3108
7	3626
8	4144
9	4662
10	5180
11	5698
12	6216
13	6734
14	7252
15	7770
16	8288
17	8806

				18 9324 19 9842 20 10360 21 10878 22 11396 23 11914 others: reserved

Module::MIS	Register:: I2C5_SDA_DEL		Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B098

Name	**Bits**	**Read/Write**	**Reset State**	**Comments**
dmy1	31:12	R/W	'hff000	-
Rvd	11:9	R/W	-	-
en	8	R/W	'b0	SDA data phase delay enable.
Rvd	7:5	R/W	-	-
sel	4:0	R/W	'h1	SDA data phase delay . sel[4:0] / SDA delay time (unit : ns) 1 518 2 1036 3 1554 4 2072 5 2590 6 3108 7 3626 8 4144 9 4662 10 5180 11 5698 12 6216 13 6734 14 7252 15 7770 16 8288 17 8806 18 9324

				19 9842 20 10360 21 10878 22 11396 23 11914 others: reserved

Module::MIS	Register:: RTC_SYS_SYNC	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B09C

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:2	R/W	-	-
wdog_ov_xor_en	1	R/W	‘b0	Watch dog overflow reset XOR function enable 1’b0 : wdog_ov_swc OR wdog_ov_nwc 1’b1 : wdog_ov_swc XOR wdog_ov_nwc If 1’b1 and wdog_ov_swc,wdog_ov_nwc inverse at the same time the reset will be ignored.
en	0	R/W	‘b0	RTC interrupt asynchronous enable. 1’b1 : Hsec,Min,Hour and Date interrupt from RTC will be synchronous to sys domain and then osc domain 1’b0 : only 2 osc DFF and posedge latch 1’b1 : for RTC crystal at 27MHz or others 1’b0 : for RTC crystal at 32768Hz

Module::MIS	Register:: GATING_EN	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B0E0

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:6	R/W	-	-
fan	5	R/W	‘b1	Enable clock gating function.
timer	4	R/W	‘b1	Enable clock gating function.
ur2_h5	3	R/W	‘b1	Enable clock gating function.
ur1_h5	2	R/W	‘b1	Enable clock gating function.
dummy1	1	R/W	‘b1	Enable clock gating function.
dummy0	0	R/W	‘b1	Enable clock gating function.

Module::MIS	Register:: DUMMY2	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B0E4

Name	Bits	Read/Write	Reset State	Comments
dummy2	31..0	R/W	‘h0	Dummy

Module::MIS	Register:: DUMMY3	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B0E8

Name	Bits	Read/Write	Reset State	Comments
dummy3	31..0	R/W	‘hffffffff	Dummy

Module::MIS	Register:: UR_H5_CTRL	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B0F0

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:4	R/W	-	-
tx_clr	3	R/W	‘b0	
rx_int_mask	2	R/W	‘b0	
rx_err_clr	1	R/W	‘b0	
en	0	R/W	‘b0	

Module::MIS	Register:: UR_H5_ST	Set::1	ATTR::nor	Type::SR	ADDR::0x9801_B0F4

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:3	R/W	-	-
rx_err_st	2:0	R	‘h0	H5 err state 3’b001 : CRC error 3’b010 : Checksum error 3’b100 : payload length error

Module::MIS	Register:: UR2_H5_CTRL	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B0F8

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:4	R/W	-	-
tx_clr	3	R/W	‘b0	
rx_int_mask	2	R/W	‘b0	
rx_err_clr	1	R/W	‘b0	
en	0	R/W	‘b0	

Module::MIS	Register:: UR2_H5_ST	Set::1	ATTR::nor	Type::SR	ADDR::0x9801_B0FC

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:3	R/W	-	-
rx_err_st	2:0	R	‘h0	H5 err state 3’b001 : CRC error 3’b010 : Checksum error 3’b100 : payload length error

2 GPIO

2.1 Register

2.1.1 Register Summery

Physical Address	*Name*	*R/W*	*Description*
0x9801_B100	MIS_GP0DIR	R/W	GPIO 0 Direction Configuration Register
0x9801_B104	MIS_GP1DIR	R/W	GPIO 1 Direction Configuration Register
0x9801_B108	MIS_GP2DIR	R/W	GPIO 0 Direction Configuration Register
0x9801_B10C	MIS_GP3DIR	R/W	GPIO 1 Direction Configuration Register
0x9801_B110	MIS_GP0DATO	R/W	GPIO 0 Data Output Register
0x9801_B114	MIS_GP1DATO	R/W	GPIO 1 Data Output Register
0x9801_B118	MIS_GP2DATO	R/W	GPIO 0 Data Output Register
0x9801_B11C	MIS_GP3DATO	R/W	GPIO 1 Data Output Register
0x9801_B120	MIS_GP0DATI	R/W	GPIO 0 Data Input Register
0x9801_B124	MIS_GP1DATI	R/W	GPIO 1 Data Input Register
0x9801_B128	MIS_GP2DATI	R/W	GPIO 0 Data Input Register
0x9801_B12C	MIS_GP3DATI	R/W	GPIO 1 Data Input Register
0x9801_B130	MIS_GP0IE	R/W	GPIO 0 Interrupt Enable Register
0x9801_B134	MIS_GP1IE	R/W	GPIO 1 Interrupt Enable Register
0x9801_B138	MIS_GP2IE	R/W	GPIO 0 Interrupt Enable Register
0x9801_B13C	MIS_GP3IE	R/W	GPIO 1 Interrupt Enable Register
0x9801_B140	MIS_GP0DP	R/W	GPIO 0 Detection Polarity Register
0x9801_B144	MIS_GP1DP	R/W	GPIO 1 Detection Polarity Register
0x9801_B148	MIS_GP2DP	R/W	GPIO 0 Detection Polarity Register
0x9801_B14C	MIS_GP3DP	R/W	GPIO 1 Detection Polarity Register
0x9801_B150	MIS_GPDEB	R/W	GPIO De-bounce Length Register

2.1.2 Register Description

Module::MIS	Register::GP0DIR	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B100

Name	**Bits**	**Read/Write**	**Reset State**	**Comments**
GPDIR	31:0	R/W	‘h0	GPIO[31:0] direction configuration. 0: Configured as input pin 1: Configured as output pin

Module::MIS	Register::GP1DIR	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B104

Name	**Bits**	**Read/Write**	**Reset State**	**Comments**
GPDIR	31:0	R/W	‘h0	GPIO[63:32] direction configuration.

				0: Configured as input pin 1: Configured as output pin

Module::MIS	Register::GP2DIR	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B108

Name	Bits	Read/Write	Reset State	Comments
GPDIR	31:0	R/W	‘h0	GPIO[95:64] direction configuration. 0: Configured as input pin 1: Configured as output pin

Module::MIS	Register::GP3DIR	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B10C

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:5	-	-	-
GPDIR	4:0	R/W	‘h0	GPIO[100:96] direction configuration. 0: Configured as input pin 1: Configured as output pin

Module::MIS	Register::GP0DATO	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B110

Name	Bits	Read/Write	Reset State	Comments
GPDATO	31:0	R/W	‘h0	GPIO[31:0] data output. Write GPDATO to output data to gpio pad

Module::MIS	Register::GP1DATO	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B114

Name	Bits	Read/Write	Reset State	Comments
GPDATO	31:0	R/W	‘h0	GPIO[63:32] data output. Write GPDATO to output data to gpio pad

Module::MIS	Register::GP2DATO	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B118

Name	Bits	Read/Write	Reset State	Comments
GPDATO	31:0	R/W	‘h0	GPIO[95:64] data output. Write GPDATO to output data to gpio pad

Module::MIS	Register::GP3DATO	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B11C

Name	Bits	Read/Write	Reset	Comments

			State	
Rvd	31:5	-	-	-
GPDATO	4:0	R/W	‘h0	GPIO[100:96] data output. Write GPDATO to output data to gpio pad

Module::MIS	Register::GP0DATI		Set::1	ATTR::nor	Type::SR	ADDR::0x9801_B120
Name	**Bits**	**Read/Write**	**Reset State**	**Comments**		
GPDATI	31:0	R	‘h0	GPIO[31:0] data input. Read GPDATI to read data from gpio pad.		

Module::MIS	Register::GP1DATI		Set::1	ATTR::nor	Type::SR	ADDR::0x9801_B124
Name	**Bits**	**Read/Write**	**Reset State**	**Comments**		
GPDATI	31:0	R	‘h0	GPIO[63:32] data input. Read GPDATI to read data from gpio pad.		

Module::MIS	Register::GP2DATI		Set::1	ATTR::nor	Type::SR	ADDR::0x9801_B128
Name	**Bits**	**Read/Write**	**Reset State**	**Comments**		
GPDATI	31:0	R	‘h0	GPIO[95:64] data input. Read GPDATI to read data from gpio pad.		

Module::MIS	Register::GP3DATI		Set::1	ATTR::nor	Type::SR	ADDR::0x9801_B12C
Name	**Bits**	**Read/Write**	**Reset State**	**Comments**		
Rvd	31:5	-	-	-		
GPDATI	4:0	R	‘h0	GPIO[100:96] data input. Read GPDATI to read data from gpio pad.		

Module::MIS	Register::GP0IE		Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B130
Name	**Bits**	**Read/Write**	**Reset State**	**Comments**		
GP	31:0	R/W	‘h0	GPIO[31:0] Assert/Dis-assert Interrupt Enable Register.		

				0: disable interrupt. 1: enable interrupt.

Module::MIS	Register::GP1IE	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B134
Name	**Bits**	**Read/Write**	**Reset State**	**Comments**	
GP	31:0	R/W	‘h0	GPIO[63:32] Assert/Dis-assert Interrupt Enable Register. 0: disable interrupt. 1: enable interrupt.	

Module::MIS	Register::GP2IE	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B138
Name	**Bits**	**Read/Write**	**Reset State**	**Comments**	
GP	31:0	R/W	‘h0	GPIO[95:64] Assert/Dis-assert Interrupt Enable Register. 0: disable interrupt. 1: enable interrupt.	

Module::MIS	Register::GP3IE	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B13C
Name	**Bits**	**Read/Write**	**Reset State**	**Comments**	
Rvd	31:5	-	-	-	
GP	4:0	R/W	‘h0	GPIO[100:96] Assert/Dis-assert Interrupt Enable Register. 0: disable interrupt. 1: enable interrupt.	

Module::MIS	Register::GP0DP	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B140
Name	**Bits**	**Read/Write**	**Reset State**	**Comments**	
GPHA	31:0	R/W	‘hffffffff	GPIO[31:0] detection polarity 0: low active. 1: high active.	

Module::MIS	Register::GP1DP	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B144
Name	**Bits**	**Read/Write**	**Reset State**	**Comments**	
GPHA	31:0	R/W	‘hffffffff	GPIO[63:32] detection polarity 0: low active. 1: high active.	

Module::MIS	Register::GP2DP	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B148

Name	Bits	Read/Write	Reset State	Comments
GPHA	31:0	R/W	‘hffffffff	GPIO[95:64] detection polarity 0: low active. 1: high active.

Module::MIS	Register::GP3DP	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B14C

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:5	-	-	-
GPHA	4:0	R/W	‘h7	GPIO[100:96] detection polarity 0: low active. 1: high active.

Module::MIS	Register::GPDEB	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_B150

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:28	-	-	-
write_enable7	27	W	-	Write enable for bit[26:24]
CLK7	26:24	R/W	’b0	De-bounce clock base. MIS_GPDEB[26:24] control GPIO[100:96]. 3’h7: 30ms 3’h6: 20ms 3’h5: 10ms 3’h4: 1ms 3’h3: 100us 3’h2: 10us 3’h1: 1us 3’h0: 37ns (27MHz)
write_enable6	23	W	-	Write enable for bit[22:20]
CLK6	22:20	R/W	’b0	De-bounce clock base. MIS_GPDEB[22:20] control GPIO[95:80]. 3’h7: 30ms 3’h6: 20ms 3’h5: 10ms 3’h4: 1ms 3’h3: 100us 3’h2: 10us 3’h1: 1us 3’h0: 37ns (27MHz)
write_enable5	19	W	-	Write enable for bit[18:16]

CLK5	18:16	R/W	’b0	De-bounce clock base. MIS_GPDEB[18:16] control GPIO[79:64]. 3’h7: 30ms 3’h6: 20ms 3’h5: 10ms 3’h4: 1ms 3’h3: 100us 3’h2: 10us 3’h1: 1us 3’h0: 37ns (27MHz)
write_enable4	15	W	-	Write enable for bit[14:12]
CLK4	14:12	R/W	’b0	De-bounce clock base. MIS_GPDEB[14:12] control GPIO[63:48]. 3’h7: 30ms 3’h6: 20ms 3’h5: 10ms 3’h4: 1ms 3’h3: 100us 3’h2: 10us 3’h1: 1us 3’h0: 37ns (27MHz)
write_enable3	11	W	-	Write enable for bit[10:8]
CLK3	10:8	R/W	’b0	De-bounce clock base. MIS_GPDEB[10:8] control GPIO[47:32]. 3’h7: 30ms 3’h6: 20ms 3’h5: 10ms 3’h4: 1ms 3’h3: 100us 3’h2: 10us 3’h1: 1us 3’h0: 37ns (27MHz)
write_enable2	7	W	-	Write enable for bit[6:4]
CLK2	6:4	R/W	’b0	De-bounce clock base. MIS_GPDEB[6:4] control GPIO[31:16]. 3’h7: 30ms 3’h6: 20ms 3’h5: 10ms 3’h4: 1ms 3’h3: 100us 3’h2: 10us 3’h1: 1us 3’h0: 37ns (27MHz)

write_enable1	3	W	-	Write enable for bit[2:0]
CLK1	2:0	R/W	'b0	De-bounce clock base. MIS_GPDEB[2:0] control GPIO[15:0]. 3'h7: 30ms 3'h6: 20ms 3'h5: 10ms 3'h4: 1ms 3'h3: 100us 3'h2: 10us 3'h1: 1us 3'h0: 37ns (27MHz)

Notice: GPIO base on the de-bounce clock to read GPIO input data.

3 UART

3.1 Register

3.1.1 UR1 Register Description

Module::MIS	Register::U1RBR_THR_DLL	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B200

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:8	-	-	-
DLL	7:0	R/W	'h00	DLAB=1 Divisor Latch LSB
RBD	7:0	R	-	DLAB=0 Read: Receiver Buffer Data.
THD	7:0	W	-	DLAB=0 Write: Transmitter Holding Data.

Module::MIS	Register::U1IER_DLH	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B204

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:8	-	-	-
DLH	7:0	R/W	'h00	DLAB=1 Divisor Latch MSB
Rvd	31:8	-	-	-
PTIME	7	R/W	'h0	DLAB=0 Programmable THRE Interrupt Mode Enable. This is used to enable/disable the generation of THRE Interrupt. 0 = disabled 1 = enabled
Rvd	6:4	-	-	-
EDSSI	3	R/W	'h0	DLAB=0 Enable modem status register interrupt. This is used to enable/disable the generation of Modem Status Interrupt. This is the fourth highest priority
ELSI	2	R/W	'h0	DLAB=0 Enable receiver line status interrupt. This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt.

				0 = disabled 1 = enabled
ETBEI	1	R/W	‘h0	DLAB=0 Enable transmitter holding register empty interrupt. This is used to enable/disable the generation of Transmitter Holding Register Empty Interrupt. This is the third highest priority interrupt. 0 = disabled 1 = enabled
ERBFI	0	R/W	‘h0	DLAB=0 Enable received data available interrupt. This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFOs enabled). These are the second highest priority interrupts. 0 = disabled 1 = enabled

Module::MIS	Register::U1IIR_FCR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B208

Name	**Bits**	**Read/Write**	**Reset State**	**Comments**
Rvd	31:8	-	-	-
FIFO16	7:6	R	‘h 0	FIFOs Enabled. This is used to indicate whether the FIFOs are enabled or disabled. 00 = disabled 11 = enabled
Rvd	5:4	-	-	-
IID	3:0	R	‘h1	Interrupt ID. This indicates the highest priority pending interrupt which can be one of the following types: 0000 = modem status 0001 = no interrupt pending 0010 = THR empty 0100 = received data available 0110 = receiver line status 0111 = busy detect 1100 = character timeout

RTRG	7:6	W	‘h0	RCVR Trigger. This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt is generated. In auto flow control mode it is used to determine when the rts_n signal is de-asserted. It also determines when the dma_rx_req_n signal is asserted in certain modes of operation. The following trigger levels are supported: 00 = 1 character in the FIFO 01 = FIFO ¼ full 10 = FIFO ½ full 11 = FIFO 2 less than full
TET	5:4	-	‘h0	TX Empty Trigger. Writes have no effect when THRE_MODE_USER == Disabled. This is used to select the empty threshold level at which the THRE Interrupts are generated when the mode is active. It also determines when the dma_tx_req_n signal is asserted when in certain modes of operation. The following trigger levels are supported: 00 = FIFO empty 01 = 2 characters in the FIFO 10 = FIFO ¼ full 11 = FIFO ½ full
DMAM	3	W	‘b0	**DMA Mode. This determines the DMA signalling mode used for the dma_tx_req_n and dma_rx_req_n output signals. But dma_tx_req_n and dma_rx_req_n aren’t be used in hardware RTL design.** **0 = mode 0** **1 = mode 1**
TFRST	2	W	‘b0	Transmitter FIFO reset. Writes 1 to clear the transmitter FIFO.
RFRST	1	W	‘b0	Receiver FIFO reset. Writes 1 to clear the receiver FIFO.
EFIFO	0	W	‘b0	Enable FIFO. When this bit is set, enable the transmitter and receiver FIFOs. Changing this bit clears the FIFOs.

Module::MIS	Register::U1LCR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B20C

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:8	-	-	-
DLAB	7	R/W	‘b0	Divisor latch access bit
BRK	6	R/W	‘b0	Break Control Bit.This is used to cause a break condition to be transmitted to the receiving device. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared.
Rvd	5	-	-	-
EPS	4	R/W	‘b0	Even Parity Select. This is used to select between even and odd parity, when parity is enabled (PEN set to one). If set to one, an even number of logic 1s is transmitted or checked. If set to zero, an odd number of logic 1s is transmitted or checked.
PEN	3	R/W	‘b0	Parity enable. This bit is used to enable and disable parity generation and detection in transmitted and received serial character respectively. 0 = parity disabled 1 = parity enabled
STB	2	R/W	‘b0	Number of stop bits. This is used to select the number of stop bits per character that the peripheral transmits and receives. If set to zero, one stop bit is transmitted in the serial data. If set to one and the data bits are set to 5 (LCR[1:0] set to zero) one and a half stop bits is transmitted. Otherwise, two stop bits are transmitted. Note that regardless of the number of stop bits selected, the receiver checks only the first stop bit. 0 = 1 stop bit 1 = 1.5 stop bits when DLS (LCR[1:0]) is zero, else 2 stop bit
WLS	1:0	R/W	‘b00	Word length select or Data Length Select. This is used to select the number of data bits per character that the peripheral transmits and receives. The number of bit that may be selected areas follows: 00 = 5 bits 01 = 6 bits

				10 = 7 bits 11 = 8 bits

Module::MIS	Register::U1MCR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B210

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:6	-	-	-
AFCE	5	R/W	'b0	Auto Flow Control Enable. When FIFOs are enabled and the Auto Flow Control Enable (AFCE) bit is set, Auto Flow Control features are enabled 0 = Auto Flow Control Mode disabled 1 = Auto Flow Control Mode enabled
LOOP	4	R/W	'b0	Loopback
Rvd	3:2	-	-	-
RTS	1	R/W	'b0	Request to Send. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data. When Auto RTS Flow Control is not enabled (MCR[5] set to zero), the rts_n signal is set low by programming MCR[1] (RTS) to a high.In Auto Flow Control, AFCE_MODE == Enabled and active (MCR[5] set to one) and FIFOs enable (FCR[0] set to one), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold). The rts_n signal is de-asserted when MCR[1] is set low. Note that in Loopback mode (MCR[4] set to one), the rts_n output is held inactive high while the value of this location is internally looped back to an input.
DTR	0	R/W	'b0	Data Terminal Ready. This is used to directly control the Data Terminal Ready (dtr_n) output. The value written to this location is inverted and driven out on dtr_n, that is: 0 = dtr_n de-asserted (logic 1) 1 = dtr_n asserted (logic 0) The Data Terminal Ready output is used to inform the modem or data set that the UART is ready to establish communications. Note that in Loopback mode (MCR[4] set to one), the dtr_n output is held inactive high while the value of this location is internally looped back to an input.

Module::MIS	Register::U1LSR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B214

Name	Bits	Read/Write	Reset	Comments

			State	
Rvd	31:8	-	-	-
RFE	7	R	‘b0	Errors in receiver FIFO. At least one parity, framing and break error in the FIFO. Receiver FIFO Error bit. This bit is only relevant when FIFOs are enabled (FCR[0] set to one). This is used to indicate if there is at least one parity error, framing error, or break indication in the FIFO. 0 = no error in RX FIFO 1 = error in RX FIFO This bit is cleared when the LSR is read and the character with the error is at the top of the receiver FIFO and there are no subsequent errors in the FIFO.
TEMT	6	R	‘b0	Transmitter Empty bit. If FIFOs enabled (FCR[0] set to one), this bit is set whenever the Transmitter Shift Register and the FIFO are both empty. If in non-FIFO mode or FIFOs are disabled, this bit is set whenever the Transmitter Holding Register and the Transmitter Shift Register are both empty.
THRE	5	R	‘b0	Transmitter holding register empty. Character mode: THR is empty. FIFO mode: transmitter FIFO is empty Transmit Holding Register Empty bit. If THRE mode is disabled (IER[7] set to zero) and regardless of FIFO's being implemented/enabled or not, this bit indicates that the THR or TX FIFO is empty. This bit is set whenever data is transferred from the THR or TX FIFO to the transmitter shift register and no new data has been written to the THR or TX FIFO. This also causes a THRE Interrupt to occur, if the THRE Interrupt is enabled. If FIFO_MODE != NONE and both modes are active (IER[7] set to one and FCR[0] set to one respectively), the functionality is switched to indicate the transmitter FIFO is full, and no longer controls THRE interrupts, which are then controlled by the FCR[5:4] threshold setting.
BI	4	R	‘b0	Break interrupt indicator. Break Interrupt bit. This is used to indicate the detection of a break sequence on the serial input data. It is set whenever the serial input, sin, is held in a logic '0' state for longer than the sum of start time + data bits + parity + stop bits. In the FIFO mode, the character associated with the break condition is carried through the FIFO and is revealed when the character is at the top of the FIFO. Reading the LSR clears the BI bit. In the non-FIFO mode, the BI indication occurs immediately and persists until the LSR is read.
FE	3	R	‘b0	Framing error. This is used to indicate the occurrence of a framing error in the receiver. A framing error occurs when the receiver does not detect a valid STOP bit in the

				received data. In the FIFO mode, since the framing error is associated with a character received, it is revealed when the character with the framing error is at the top of the FIFO. When a framing error occurs, the UART tries to resynchronize. It does this by assuming that the error was due to the start bit of the next character and then continues receiving the other bit i.e. data, and/or parity and stop. It should be noted that the Framing Error (FE) bit (LSR[3]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). 0 = no framing error 1 = framing error Reading the LSR clears the FE bit.
PE	2	R	‘b0	Parity error. This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (LCR[3]) is set. In the FIFO mode, since the parity error is associated with a character received, it is revealed when the character with the parity error arrives at the top of the FIFO. It should be noted that the Parity Error (PE) bit (LSR[2]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). 0 = no parity error 1 = parity error Reading the LSR clears the PE bit.
OE	1	R	‘b0	Overrun error bit. This is used to indicate the occurrence of an overrun error. This occurs if a new data character was received before the previous data was read. In the non-FIFO mode, the OE bit is set when a new character arrives in the receiver before the previous character was read from the RBR. When this happens, the data in the RBR is overwritten. In the FIFO mode, an overrun error occurs when the FIFO is full and a new character arrives at the receiver. The data in the FIFO is retained and the data in the receive shift register is lost. 0 = no overrun error 1 = overrun error Reading the LSR clears the OE bit.
DR	0	R	‘b0	Data ready. Character mode: data ready in RBR FIFO mode: receiver FIFO is not empty. This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO. 0 = no data ready 1 = data ready This bit is cleared when the RBR is read in non-FIFO mode, or when the receiver FIFO is empty, in FIFO mode.

Module::MIS	Register::U1MSR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B218

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:8	-	-	-
DCD	7	R	'b0	In loopback mode, returns the bit 2 of MCR. In normal mode, returns 1.
RI	6	R	'b0	In loopback mode, returns the bit 3 of MCR. In normal mode, returns 0.
DSR	5	R	'b0	In loopback mode, returns the bit 0 of MCR In normal mode, returns 1.
CTS	4	R	'b0	Clear to send. 0 = CTS# detected high 1 = CTS# detected low
DDCD	3	R	'b0	Delta DCD. DCD change. The bit will be clear to zero after read the bit.
TERI	2	R	'b0	Delta RI. RI change. The bit will be clear to zero after read the bit.
DDSR	1	R	'b0	Delta DSR. DSR change. The bit will be clear to zero after read the bit.
DCTS	0	R	'b0	Delta clear to send. CTSchange . The bit will be clear to zero after read the bit.

p.s:

DCTS (bit 0), DDSR (bit 1) and DDCD (bit 3) bits record whether the modem control lines (`cts_n`, `dsr_n` and `dcd_n`) have changed since the last time the CPU read the MSR. TERI (bit 2) indicates `ri_n` has changed from an active low, to an inactive high state since the last time the MSR was read. In Loopback Mode, DCTS reflects changes on MCR bit 1 (RTS), DDSR reflects changes on MCR bit 0 (DTR) and DDCD reflects changes on MCR bit 3 (Out2), while TERI reflects when MCR bit 2 (Out1) has changed state from a high to a low. The CTS, DSR, RI and DCD Modem Status bits contain information on the current state of the modem control lines. CTS (bit 4) is the compliment of `cts_n`, DSR (bit 5) is the compliment of `dsr_n`, RI (bit 6) is the compliment of `ri_n` and DCD (bit 7) is the compliment of `dcd_n`. In Loopback Mode, CTS is the same as MCR bit 1 (RTS), DSR is the same as MCR bit 0 (DTR), RI is the same as MCR bit 2 (Out1) and DCD is the same as MCR bit 3 (Out2).

Module::MIS	Register::U1SCR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B21C

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:8	-	-	-
SCR	7:0	R/W	'h0	Scratch Register.

Module::MIS	Register::U1SRBR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B230

		6		

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:8	-	-	-
RBD	7:0	R	‘h0	Receiver buffer data. This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an overrun error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO are preserved, but any incoming data is lost. An overrun error also occurs.

Rvd	31:8	-	-	-
THD	7:0	W	-	This is a shadow register for the THR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If in non-FIFO mode or FIFOs are disabled (FCR[0] set to zero) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the

				THR before the THRE is set again causes the THR data to be overwritten. If in FIFO mode and FIFOs are enabled (FCR[0] set to one) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.

Module::MIS	Register::U1FAR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B270

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:1	-	-	-
FAR	0	R/W	'h0	This register is use to enable a FIFO access mode for testing, so that the receive FIFO can be written by the master and the transmit FIFO can be read by the master when FIFOs are implemented and enabled. When FIFOs are not implemented or not enabled it allows the RBR to be written by the master and the THR to be read by the master. 0 = FIFO access mode disabled 1 = FIFO access mode enabled Note, that when the FIFO access mode is enabled/disabled, the control portion of the receive FIFO and transmit FIFO is reset and the FIFOs are treated as empty.

Module::MIS	Register::U1TFR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B274

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:8	-	-	-
FAR	7:0	R	'h0	Transmit FIFO Read. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFOs are implemented and enabled, reading this register gives the data at the top of the transmit FIFO. Each consecutive read pops the transmit FIFO and gives the next data value that is currently at the top of the

				FIFO. When FIFOs are not implemented or not enabled, reading this register gives the data in the THR.

Module::MIS	Register::U1RFW	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B278

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:10	-	-	-
RFFE	9	W	'h0	Receive FIFO Framing Error. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFOs are implemented and enabled, this bit is used to write framing error detection information to the receive FIFO. When FIFOs are not implemented or not enabled, this bit is used to write framing error detection information to the RBR.
RFPF	8	W	'h0	Receive FIFO Parity Error. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFOs are implemented and enabled, this bit is used to write parity error detection information to the receive FIFO. When FIFOs are not implemented or not enabled, this bit is used to write parity error detection information to the RBR.
RFWD	7:0	W	'h0	Receive FIFO Write Data. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFOs are implemented and enabled, the data that is written to the RFWD is pushed into the receive FIFO. Each consecutive write pushes the new data to the next write location in the receive FIFO. When FIFOs are not implemented or not enabled, the data that is written to the RFWD is pushed into the RBR.

Module::MIS	Register::U1USR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B27C

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:5	-	-	-
RFF	4	R	'h0	Receive FIFO Full. This bit is only valid when FIFO_STAT == YES. This is used to

				indicate that the receive FIFO is completely full. 0 = Receive FIFO not full 1 = Receive FIFO Full This bit is cleared when the RX FIFO is no longer full.
RFNE	3	R	‘h0	Receive FIFO Not Empty. This bit is only valid when FIFO_STAT == YES. This is used to indicate that the receive FIFO contains one or more entries. 0 = Receive FIFO is empty 1 = Receive FIFO is not empty This bit is cleared when the RX FIFO is empty.
TFE	2	R	‘h0	Transmit FIFO Empty. This bit is only valid when FIFO_STAT == YES. This is used to indicate that the transmit FIFO is completely empty. 0 = Transmit FIFO is not empty 1 = Transmit FIFO is empty This bit is cleared when the TX FIFO is no longer empty.
TFNF	1	R	‘h0	Transmit FIFO Not Full. This bit is only valid when FIFO_STAT == YES. This is used to indicate that the transmit FIFO in not full. 0 = Transmit FIFO is full 1 = Transmit FIFO is not full This bit is cleared when the TX FIFO is full.
BUSY	0	R	‘h0	UART Busy. This bit is valid only when UART_16550_COMPATIBLE == NO and indicates that a serial transfer is in progress, ; when cleared, indicates that the DW_apb_uart is idle or inactive. 0 = DW_apb_uart is idle or inactive 1 = DW_apb_uart is busy (actively transferring data) NOTE: It is possible for the UART Busy bit to be cleared even though a new character may have been sent from another device. That is, if the DW_apb_uart has no data in THR and RBR and there is no transmission in progress and a start bit of a new character has just reached the DW_apb_uart. This is due to the fact that a

				valid start is not seen until the middle of the bit period and this duration is dependent on the baud divisor that has been programmed. If a second system clock has been implemented (CLOCK_MODE == Enabled), the assertion of this bit is also delayed by several cycles of the slower clock.

Module::MIS	Register::U1TFL	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B280
Name	**Bits**	**Read/Write**	**Reset State**	**Comments**	
Rvd	31:8	-	-	-	
TFL	7:0	R	'h0	Transmit FIFO Level. This is indicates the number of data entries in the transmit FIFO.	

Module::MIS	Register::U1RFL	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B284
Name	**Bits**	**Read/Write**	**Reset State**	**Comments**	
Rvd	31:8	-	-	-	
RFL	7:0	R	'h0	Receive FIFO Level. This is indicates the number of data entries in the receive FIFO.	

Module::MIS	Register::U1SRR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B288
Name	**Bits**	**Read/Write**	**Reset State**	**Comments**	
Rvd	31:3	-	-	-	
XFR	2	W	'h0	XMIT FIFO Reset. This is a shadow register for the XMIT FIFO Reset bit (FCR[2]). This can be used to remove the burden on software having to store previously written FCR values (which are pretty static) just to reset the transmit FIFO. This resets the control portion of the transmit FIFO and treats the FIFO as empty. This also de-asserts the DMA TX request and single signals when additional DMA handshaking signals are selected (DMA_EXTRA == YES). Note that this bit is 'self-clearing'. It is not necessary to clear this bit.	
RFR	1	W	'h0	RCVR FIFO Reset. This is a shadow register for the RCVR FIFO Reset bit	

				(FCR[1]). This can be used to remove the burden on software having to store previously written FCR values (which are pretty static) just to reset the receive FIFO This resets the control portion of the receive FIFO and treats the FIFO as empty. This also de-asserts the DMA RX request and single signals when additional DMA handshaking signals are selected (DMA_EXTRA == YES). Note that this bit is 'self-clearing'. It is not necessary to clear this bit.
UR	0	W	'h0	UART Reset. This asynchronously resets the DW_apb_uart and synchronously removes the reset assertion. For a two clock implementation both pclk and sclk domains are reset.

Module::MIS	Register::U1SBCR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B290

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:1	-	-	-
SBCR	0	R/W	'h0	

Module::MIS	Register::U1SDMAM	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B294

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:1	-	-	-
SDMAM	0	R/W	'h0	Shadow DMA Mode. This is a shadow register for the DMA mode bit (FCR[3]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the DMA Mode bit gets updated. This determines the DMA signalling mode used for the dma_tx_req_n and dma_rx_req_n output signals when additional DMA handshaking signals are not selected (DMA_EXTRA == NO). 0 = mode 0 1 = mode 1

Module::MIS	Register::U1SFE	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B298

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:1	-	-	-
SFE	0	R/W	'h0	Shadow FIFO Enable. This is a shadow register for the FIFO enable bit (FCR[0]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the FIFO enable bit gets updated.This enables/disables the transmit (XMIT) and receive (RCVR) FIFOs. If this bit is set to zero (disabled) after being enabled then both the XMIT and RCVR controller portion of FIFOs are reset.

Module::MIS	Register::U1SRT	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B29C

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:2	-	-	-
SRT	1:0	R/W	'h0	Shadow RCVR Trigger. This is a shadow register for the RCVR trigger bits (FCR[7:6]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the RCVR trigger bit gets updated. This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt is generated. It also determines when the dma_rx_req_n signal is asserted when DMA Mode (FCR[3]) = 1. The following trigger levels are supported: 00 = 1 character in the FIFO 01 = FIFO ¼ full 10 = FIFO ½ full 11 = FIFO 2 less than full

Module::MIS	Register::U1STET	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B2A0

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:2	-	-	-
STET	1:0	R/W	'h0	Shadow TX Empty Trigger. This is a shadow register for the TX empty trigger bits (FCR[5:4]). This can be used to remove

				the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the TX empty trigger bit gets updated. This is used to select the empty threshold level at which the THRE Interrupts are generated when the mode is active. The following trigger levels are supported: 00 = FIFO empty 01 = 2 characters in the FIFO 10 = FIFO ¼ full 11 = FIFO ½ full

Module::MIS	Register::U1HTX	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B2A4

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:1	-	-	-
HTX	0	R/W	'h0	This register is use to halt transmissions for testing, so that the transmit FIFO can be filled by the master when FIFOs are implemented and enabled. 0 = Halt TX disabled 1 = Halt TX enabled Note, if FIFOs are implemented and not enabled, the setting of the halt TX register has no effect on operation.

Module::MIS	Register::U1DMASA	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B2A8

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:1	-	-	-
DMASA	0	W	'h0	This register is use to perform a DMA software acknowledge if a transfer needs to be terminated due to an error condition. For example, if the DMA disables the channel, then the DW_apb_uart should clear its request. This causes the TX request, TX single, RX request and RX single signals to de-assert. Note that this bit is 'self-clearing'. It is not necessary to clear this bit.

Module::MIS	Register::U1CPR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B2F4

Name	Bits	Read/Writ	Reset	Comments

		e	State	
Rvd	31:24	-	-	
FIFO_MODE	23:16	R	-	0x00 = 0 0x01 = 16 0x02 = 32 to 0x80 = 2048 0x81- 0xff = reserved
Rvd	15:14	R	-	
DMA_EXTRA	13	R	-	0 = FALSE 1 = TRUE
UART_ADD_ENCODED_PARAMS	12	R	-	0 = FALSE 1 = TRUE
SHADOW	11	R	-	0 = FALSE 1 = TRUE
FIFO_STAT	10	R	-	0 = FALSE 1 = TRUE
FIFO_ACCESS	9	R	-	0 = FALSE 1 = TRUE
ADDITIONAL_FEAT	8	R	-	0 = FALSE 1 = TRUE
SIR_LP_MODE	7	R	-	0 = FALSE 1 = TRUE
SIR_MODE	6	R	-	0 = FALSE 1 = TRUE
THRE_MODE	5	R	-	0 = FALSE 1 = TRUE
AFCE_MODE	4	R	-	0 = FALSE 1 = TRUE
Rvd	3:2	R	-	
APB_DATA_WIDTH	1:0	R	-	00 = 8 bits 01 = 16 bits 10 = 32 bits 11 = reserved

Module::MIS	Register::U1UCV	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B2F8

Name	Bits	Read/Write	Reset State	Comments
UCV	31:0	R	-	ASCII value for each number in the version, followed by *. For example 32_30_31_2A represents the version 2.01*

Module::MIS	Register::U1CTR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B2FC

Name	Bits	Read/Write	Reset State	Comments
CTR	31:0	R	-	This register contains the peripherals identification code.

3.1.2 UR2 Register Description

3.1.2.1 UR2 Register Description

Module::MIS	Register::U2RBR_THR_DLL	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B400

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:8	-	-	-
DLL	7:0	R/W	'h00	DLAB=1 Divisor Latch LSB

RBD	7:0	R	-	DLAB=0 Read: Receiver Buffer Data.
THD	7:0	W	-	DLAB=0 Write: Transmitter Holding Data.

Module::MIS	Register::U2IER_DLH	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B404

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:8	-	-	-
DLH	7:0	R/W	'h00	DLAB=1 Divisor Latch MSB

Rvd	31:8	-	-	-
PTIME	7	R/W	'h0	DLAB=0 Programmable THRE Interrupt Mode Enable. This is used to enable/disable the generation of THRE Interrupt. 0 = disabled 1 = enabled
Rvd	6:4	-	-	-
EDSSI	3	R/W	'h0	DLAB=0 Enable modem status register interrupt. This is used to enable/disable the

				generation of Modem Status Interrupt. This is the fourth highest priority
ELSI	2	R/W	‘h0	DLAB=0 Enable receiver line status interrupt. This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt. 0 = disabled 1 = enabled
ETBEI	1	R/W	‘h0	DLAB=0 Enable transmitter holding register empty interrupt. This is used to enable/disable the generation of Transmitter Holding Register Empty Interrupt. This is the third highest priority interrupt. 0 = disabled 1 = enabled
ERBFI	0	R/W	‘h0	DLAB=0 Enable received data available interrupt. This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFOs enabled). These are the second highest priority interrupts. 0 = disabled 1 = enabled

Module::MIS	Register::U2IIR_FCR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B408

Name	**Bits**	**Read/Write**	**Reset State**	**Comments**
Rvd	31:8	-	-	-
FIFO16	7:6	R	‘h 0	FIFOs Enabled. This is used to indicate whether the FIFOs are enabled or disabled. 00 = disabled 11 = enabled
Rvd	5:4	-	-	-
IID	3:0	R	‘h1	Interrupt ID. This indicates the highest priority pending interrupt which can be one of the following types:

				0000 = modem status 0001 = no interrupt pending 0010 = THR empty 0100 = received data available 0110 = receiver line status 0111 = busy detect 1100 = character timeout

RTRG	7:6	W	‘h0	RCVR Trigger. This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt is generated. In auto flow control mode it is used to determine when the rts_n signal is de-asserted. It also determines when the dma_rx_req_n signal is asserted in certain modes of operation. The following trigger levels are supported: 00 = 1 character in the FIFO 01 = FIFO ¼ full 10 = FIFO ½ full 11 = FIFO 2 less than full
TET	5:4	-	‘h0	TX Empty Trigger. Writes have no effect when THRE_MODE_USER == Disabled. This is used to select the empty threshold level at which the THRE Interrupts are generated when the mode is active. It also determines when the dma_tx_req_n signal is asserted when in certain modes of operation. The following trigger levels are supported: 00 = FIFO empty 01 = 2 characters in the FIFO 10 = FIFO ¼ full 11 = FIFO ½ full
DMAM	3	W	‘b0	**DMA Mode. This determines the DMA signalling mode used for the dma_tx_req_n and dma_rx_req_n output signals. But dma_tx_req_n and dma_rx_req_n aren’t be used in hardware RTL design.** **0 = mode 0** **1 = mode 1**
TFRST	2	W	‘b0	Transmitter FIFO reset. Writes 1 to clear the transmitter FIFO.

RFRST	1	W	‘b0	Receiver FIFO reset. Writes 1 to clear the receiver FIFO.
EFIFO	0	W	‘b0	Enable FIFO. When this bit is set, enable the transmitter and receiver FIFOs. Changing this bit clears the FIFOs.

Module::MIS	Register::U2LCR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B40C

Name	**Bits**	**Read/Write**	**Reset State**	**Comments**
Rvd	31:8	-	-	-
DLAB	7	R/W	‘b0	Divisor latch access bit
BRK	6	R/W	‘b0	Break Control Bit.This is used to cause a break condition to be transmitted to the receiving device. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared.
Rvd	5	-	-	-
EPS	4	R/W	‘b0	Even Parity Select. This is used to select between even and odd parity, when parity is enabled (PEN set to one). If set to one, an even number of logic 1s is transmitted or checked. If set to zero, an odd number of logic 1s is transmitted or checked.
PEN	3	R/W	‘b0	Parity enable. This bit is used to enable and disable parity generation and detection in transmitted and received serial character respectively. 0 = parity disabled 1 = parity enabled
STB	2	R/W	‘b0	Number of stop bits. This is used to select the number of stop bits per character that the peripheral transmits and receives. If set to zero, one stop bit is transmitted in the serial data. If set to one and the data bits are set to 5 (LCR[1:0] set to zero) one and a half stop bits is transmitted. Otherwise, two stop bits are transmitted. Note that regardless of the number of stop bits selected, the receiver checks only the first stop bit. 0 = 1 stop bit 1 = 1.5 stop bits when DLS (LCR[1:0]) is

				zero, else 2 stop bit
WLS	1:0	R/W	‘b00	Word length select or Data Length Select. This is used to select the number of data bits per character that the peripheral transmits and receives. The number of bit that may be selected areas follows: 00 = 5 bits 01 = 6 bits 10 = 7 bits 11 = 8 bits

Module::MIS	Register::U2MCR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B410

Name	**Bits**	**Read/Write**	**Reset State**	**Comments**
Rvd	31:6	-	-	-
AFCE	5	R/W	‘b0	Auto Flow Control Enable. When FIFOs are enabled and the Auto Flow Control Enable (AFCE) bit is set, Auto Flow Control features are enabled 0 = Auto Flow Control Mode disabled 1 = Auto Flow Control Mode enabled
LOOP	4	R/W	‘b0	Loopback
Rvd	3:2	-	-	-
RTS	1	R/W	‘b0	Request to Send. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data. When Auto RTS Flow Control is not enabled (MCR[5] set to zero), the rts_n signal is set low by programming MCR[1] (RTS) to a high.In Auto Flow Control, AFCE_MODE == Enabled and active (MCR[5] set to one) and FIFOs enable (FCR[0] set to one), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold). The rts_n signal is de-asserted when MCR[1] is set low. Note that in Loopback mode (MCR[4] set to one), the rts_n output is held inactive high while the value of this location is internally looped back to an input.
DTR	0	R/W	‘b0	Data Terminal Ready. This is used to directly control the Data Terminal Ready (dtr_n) output. The value written to this location is inverted and driven out on dtr_n, that is: 0 = dtr_n de-asserted (logic 1) 1 = dtr_n asserted (logic 0) The Data Terminal Ready output is used to inform

				the modem or data set that the UART is ready to establish communications. Note that in Loopback mode (MCR[4] set to one), the dtr_n output is held inactive high while the value of this location is internally looped back to an input.

Module::MIS	Register::U2LSR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B414

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:8	-	-	-
RFE	7	R	'b0	Errors in receiver FIFO. At least one parity, framing and break error in the FIFO. Receiver FIFO Error bit. This bit is only relevant when FIFOs are enabled (FCR[0] set to one). This is used to indicate if there is at least one parity error, framing error, or break indication in the FIFO. 0 = no error in RX FIFO 1 = error in RX FIFO This bit is cleared when the LSR is read and the character with the error is at the top of the receiver FIFO and there are no subsequent errors in the FIFO.
TEMT	6	R	'b0	Transmitter Empty bit. If FIFOs enabled (FCR[0] set to one), this bit is set whenever the Transmitter Shift Register and the FIFO are both empty. If in non-FIFO mode or FIFOs are disabled, this bit is set whenever the Transmitter Holding Register and the Transmitter Shift Register are both empty.
THRE	5	R	'b0	Transmitter holding register empty. Character mode: THR is empty. FIFO mode: transmitter FIFO is empty Transmit Holding Register Empty bit. If THRE mode is disabled (IER[7] set to zero) and regardless of FIFO's being implemented/enabled or not, this bit indicates that the THR or TX FIFO is empty. This bit is set whenever data is transferred from the THR or TX FIFO to the transmitter shift register and no new data has been written to the THR or TX FIFO. This also causes a THRE Interrupt to occur, if the THRE Interrupt is enabled. If FIFO_MODE != NONE and both modes are active (IER[7] set to one and FCR[0] set to one respectively), the functionality is switched to indicate the transmitter FIFO is full, and no longer controls THRE interrupts, which are then controlled by the FCR[5:4] threshold setting.
BI	4	R	'b0	Break interrupt indicator. Break Interrupt bit. This is used to indicate the detection of a break sequence on the serial input data. It is set whenever the serial input, sin, is held in a logic '0' state for longer than the sum of start time + data bits + parity + stop bits.

				In the FIFO mode, the character associated with the break condition is carried through the FIFO and is revealed when the character is at the top of the FIFO. Reading the LSR clears the BI bit. In the non-FIFO mode, the BI indication occurs immediately and persists until the LSR is read.
FE	3	R	‘b0	Framing error. This is used to indicate the occurrence of a framing error in the receiver. A framing error occurs when the receiver does not detect a valid STOP bit in the received data. In the FIFO mode, since the framing error is associated with a character received, it is revealed when the character with the framing error is at the top of the FIFO. When a framing error occurs, the UART tries to resynchronize. It does this by assuming that the error was due to the start bit of the next character and then continues receiving the other bit i.e. data, and/or parity and stop. It should be noted that the Framing Error (FE) bit (LSR[3]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). 0 = no framing error 1 = framing error Reading the LSR clears the FE bit.
PE	2	R	‘b0	Parity error. This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (LCR[3]) is set. In the FIFO mode, since the parity error is associated with a character received, it is revealed when the character with the parity error arrives at the top of the FIFO. It should be noted that the Parity Error (PE) bit (LSR[2]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). 0 = no parity error 1 = parity error Reading the LSR clears the PE bit.
OE	1	R	‘b0	Overrun error bit. This is used to indicate the occurrence of an overrun error. This occurs if a new data character was received before the previous data was read. In the non-FIFO mode, the OE bit is set when a new character arrives in the receiver before the previous character was read from the RBR. When this happens, the data in the RBR is overwritten. In the FIFO mode, an overrun error occurs when the FIFO is full and a new character arrives at the receiver. The data in the FIFO is retained and the data in the receive shift register is lost. 0 = no overrun error 1 = overrun error Reading the LSR clears the OE bit.
DR	0	R	‘b0	Data ready.

				Character mode: data ready in RBR FIFO mode: receiver FIFO is not empty. This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO. 0 = no data ready 1 = data ready This bit is cleared when the RBR is read in non-FIFO mode, or when the receiver FIFO is empty, in FIFO mode.

Module::MIS	Register::U2MSR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B418

Name	**Bits**	**Read/Write**	**Reset State**	**Comments**
Rvd	31:8	-	-	-
DCD	7	R	‘b0	In loopback mode, returns the bit 2 of MCR. In normal mode, returns 1.
RI	6	R	‘b0	In loopback mode, returns the bit 3 of MCR. In normal mode, returns 0.
DSR	5	R	‘b0	In loopback mode, returns the bit 0 of MCR In normal mode, returns 1.
CTS	4	R	‘b0	Clear to send. 0 = CTS# detected high 1 = CTS# detected low
DDCD	3	R	‘b0	Delta DCD. DCD change. The bit will be clear to zero after read the bit.
TERI	2	R	‘b0	Delta RI. RI change. The bit will be clear to zero after read the bit.
DDSR	1	R	‘b0	Delta DSR. DSR change. The bit will be clear to zero after read the bit.
DCTS	0	R	‘b0	Delta clear to send. CTSchange . The bit will be clear to zero after read the bit.

p.s:

DCTS (bit 0), DDSR (bit 1) and DDCD (bit 3) bits record whether the modem control lines (`cts_n`, `dsr_n` and `dcd_n`) have changed since the last time the CPU read the MSR. TERI (bit 2) indicates `ri_n` has changed from an active low, to an inactive high state since the last time the MSR was read. In Loopback Mode, DCTS reflects changes on MCR bit 1 (RTS), DDSR reflects changes on MCR bit 0 (DTR) and DDCD reflects changes on MCR bit 3 (Out2), while TERI reflects when MCR bit 2 (Out1) has changed state from a high to a low. The CTS, DSR, RI and DCD Modem Status bits contain information on the current state of the modem control lines. CTS (bit 4) is the compliment of `cts_n`, DSR (bit 5) is the compliment of `dsr_n`, RI (bit 6) is the compliment of `ri_n` and DCD (bit 7) is the compliment of `dcd_n`. In Loopback Mode, CTS is the same as MCR bit 1 (RTS), DSR is the same as MCR bit 0 (DTR), RI is the same as MCR bit 2 (Out1) and DCD is the same as MCR bit 3 (Out2).

Module::MIS	Register::U2SCR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B41C

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:8	-	-	-
SCR	7:0	R/W	'h0	Scratch Register.

Module::MIS	Register::U2SRBR	Set::16	ATTR::sfdf	Type::SR	ADDR::0x9801_B430

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:8	-	-	-
RBD	7:0	R	'h0	Receiver buffer data. This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an overrun error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO are preserved, but any incoming data is lost. An overrun error also occurs.

Rvd	31:8	-	-	-
THD	7:0	W	-	This is a shadow register for the THR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains data to be

				transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If in non-FIFO mode or FIFOs are disabled (FCR[0] set to zero) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If in FIFO mode and FIFOs are enabled (FCR[0] set to one) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.

Module::MIS	Register::U2FAR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B470

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:1	-	-	-
FAR	0	R/W	'h0	This register is use to enable a FIFO access mode for testing, so that the receive FIFO can be written by the master and the transmit FIFO can be read by the master when FIFOs are implemented and enabled. When FIFOs are not implemented or not enabled it allows the RBR to be written by the master and the THR to be read by the master. 0 = FIFO access mode disabled 1 = FIFO access mode enabled Note, that when the FIFO access mode is enabled/disabled, the control portion of the receive FIFO and transmit FIFO is reset and the FIFOs are treated as empty.

Module::MIS	Register::U2TFR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B474

Name	Bits	Read/Write	Reset	Comments

			State	
Rvd	31:8	-	-	-
FAR	7:0	R	'h0	Transmit FIFO Read. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFOs are implemented and enabled, reading this register gives the data at the top of the transmit FIFO. Each consecutive read pops the transmit FIFO and gives the next data value that is currently at the top of the FIFO. When FIFOs are not implemented or not enabled, reading this register gives the data in the THR.

Module::MIS	Register::U2RFW	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B478

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:10	-	-	-
RFFE	9	W	'h0	Receive FIFO Framing Error. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFOs are implemented and enabled, this bit is used to write framing error detection information to the receive FIFO. When FIFOs are not implemented or not enabled, this bit is used to write framing error detection information to the RBR.
RFPF	8	W	'h0	Receive FIFO Parity Error. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFOs are implemented and enabled, this bit is used to write parity error detection information to the receive FIFO. When FIFOs are not implemented or not enabled, this bit is used to write parity error detection information to the RBR.
RFWD	7:0	W	'h0	Receive FIFO Write Data. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFOs are implemented and enabled, the data that is written to the RFWD is pushed into the receive FIFO. Each consecutive write pushes the new data to the next write location in the receive FIFO. When FIFOs

				are not implemented or not enabled, the data that is written to the RFWD is pushed into the RBR.

Module::MIS	Register::U2USR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B47C

Name	**Bits**	**Read/Write**	**Reset State**	**Comments**
Rvd	31:5	-	-	-
RFF	4	R	'h0	Receive FIFO Full. This bit is only valid when FIFO_STAT == YES. This is used to indicate that the receive FIFO is completely full. 0 = Receive FIFO not full 1 = Receive FIFO Full This bit is cleared when the RX FIFO is no longer full.
RFNE	3	R	'h0	Receive FIFO Not Empty. This bit is only valid when FIFO_STAT == YES. This is used to indicate that the receive FIFO contains one or more entries. 0 = Receive FIFO is empty 1 = Receive FIFO is not empty This bit is cleared when the RX FIFO is empty.
TFE	2	R	'h0	Transmit FIFO Empty. This bit is only valid when FIFO_STAT == YES. This is used to indicate that the transmit FIFO is completely empty. 0 = Transmit FIFO is not empty 1 = Transmit FIFO is empty This bit is cleared when the TX FIFO is no longer empty.
TFNF	1	R	'h0	Transmit FIFO Not Full. This bit is only valid when FIFO_STAT == YES. This is used to indicate that the transmit FIFO in not full. 0 = Transmit FIFO is full 1 = Transmit FIFO is not full This bit is cleared when the TX FIFO is full.
BUSY	0	R	'h0	UART Busy. This bit is valid only when UART_16550_COMPATIBLE == NO and indicates that a serial transfer is in progress, ; when cleared, indicates that the DW_apb_uart is idle or inactive. 0 = DW_apb_uart is idle or inactive

				1 = DW_apb_uart is busy (actively transferring data) NOTE: It is possible for the UART Busy bit to be cleared even though a new character may have been sent from another device. That is, if the DW_apb_uart has no data in THR and RBR and there is no transmission in progress and a start bit of a new character has just reached the DW_apb_uart. This is due to the fact that a valid start is not seen until the middle of the bit period and this duration is dependent on the baud divisor that has been programmed. If a second system clock has been implemented (CLOCK_MODE == Enabled), the assertion of this bit is also delayed by several cycles of the slower clock.

Module::MIS	Register::U2TFL	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B480

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:8	-	-	-
TFL	7:0	R	'h0	Transmit FIFO Level. This is indicates the number of data entries in the transmit FIFO.

Module::MIS	Register::U2RFL	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B484

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:8	-	-	-
RFL	7:0	R	'h0	Receive FIFO Level. This is indicates the number of data entries in the receive FIFO.

Module::MIS	Register::U2SRR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B488

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:3	-	-	-
XFR	2	W	'h0	XMIT FIFO Reset. This is a shadow register for the XMIT FIFO Reset bit (FCR[2]). This can be used to remove the burden on software having to store previously written FCR values (which are pretty static) just to reset the transmit FIFO.

				This resets the control portion of the transmit FIFO and treats the FIFO as empty. This also de-asserts the DMA TX request and single signals when additional DMA handshaking signals are selected (DMA_EXTRA == YES). Note that this bit is 'self-clearing'. It is not necessary to clear this bit.
RFR	1	W	‘h0	RCVR FIFO Reset. This is a shadow register for the RCVR FIFO Reset bit (FCR[1]). This can be used to remove the burden on software having to store previously written FCR values (which are pretty static) just to reset the receive FIFO This resets the control portion of the receive FIFO and treats the FIFO as empty. This also de-asserts the DMA RX request and single signals when additional DMA handshaking signals are selected (DMA_EXTRA == YES). Note that this bit is 'self-clearing'. It is not necessary to clear this bit.
UR	0	W	‘h0	UART Reset. This asynchronously resets the DW_apb_uart and synchronously removes the reset assertion. For a two clock implementation both pclk and sclk domains are reset.

Module::MIS	Register::U2SBCR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B490

Name	**Bits**	**Read/Write**	**Reset State**	**Comments**
Rvd	31:1	-	-	-
SBCR	0	R/W	‘h0	

Module::MIS	Register::U2SDMAM	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B494

Name	**Bits**	**Read/Write**	**Reset State**	**Comments**
Rvd	31:1	-	-	-
SDMAM	0	R/W	‘h0	Shadow DMA Mode. This is a shadow register for the DMA mode bit (FCR[3]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the DMA Mode bit

				gets updated. This determines the DMA signalling mode used for the dma_tx_req_n and dma_rx_req_n output signals when additional DMA handshaking signals are not selected (DMA_EXTRA == NO). 0 = mode 0 1 = mode 1

Module::MIS	Register::U2SFE	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B498
Name	**Bits**	**Read/Write**	**Reset State**	**Comments**	
Rvd	31:1	-	-	-	
SFE	0	R/W	'h0	Shadow FIFO Enable. This is a shadow register for the FIFO enable bit (FCR[0]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the FIFO enable bit gets updated.This enables/disables the transmit (XMIT) and receive (RCVR) FIFOs. If this bit is set to zero (disabled) after being enabled then both the XMIT and RCVR controller portion of FIFOs are reset.	

Module::MIS	Register::U2SRT	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B49C
Name	**Bits**	**Read/Write**	**Reset State**	**Comments**	
Rvd	31:2	-	-	-	
SRT	1:0	R/W	'h0	Shadow RCVR Trigger. This is a shadow register for the RCVR trigger bits (FCR[7:6]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the RCVR trigger bit gets updated. This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt is generated. It also determines when the dma_rx_req_n signal is asserted when DMA Mode (FCR[3]) = 1. The following trigger levels are supported: 00 = 1 character in the FIFO 01 = FIFO ¼ full 10 = FIFO ½ full	

				11 = FIFO 2 less than full

Module::MIS	Register::U2STET	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B4A0

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:2	-	-	-
STET	1:0	R/W	'h0	Shadow TX Empty Trigger. This is a shadow register for the TX empty trigger bits (FCR[5:4]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the TX empty trigger bit gets updated. This is used to select the empty threshold level at which the THRE Interrupts are generated when the mode is active. The following trigger levels are supported: 00 = FIFO empty 01 = 2 characters in the FIFO 10 = FIFO ¼ full 11 = FIFO ½ full

Module::MIS	Register::U2HTX	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B4A4

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:1	-	-	-
HTX	0	R/W	'h0	This register is use to halt transmissions for testing, so that the transmit FIFO can be filled by the master when FIFOs are implemented and enabled. 0 = Halt TX disabled 1 = Halt TX enabled Note, if FIFOs are implemented and not enabled, the setting of the halt TX register has no effect on operation.

Module::MIS	Register::U2DMASA	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B4A8

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:1	-	-	-
DMASA	0	W	'h0	This register is use to perform a DMA software acknowledge if a transfer needs to be terminated due to an error condition. For

				example, if the DMA disables the channel, then the DW_apb_uart should clear its request. This causes the TX request, TX single, RX request and RX single signals to de-assert. Note that this bit is 'self-clearing'. It is not necessary to clear this bit.

Module::MIS	Register::U2CPR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B4F4

Name	**Bits**	**Read/Write**	**Reset State**	**Comments**
Rvd	31:24	-	-	
FIFO_MODE	23:16	R	-	0x00 = 0 0x01 = 16 0x02 = 32 to 0x80 = 2048 0x81- 0xff = reserved
Rvd	15:14	R	-	
DMA_EXTRA	13	R	-	0 = FALSE 1 = TRUE
UART_ADD_ENCODED_PARAMS	12	R	-	0 = FALSE 1 = TRUE
SHADOW	11	R	-	0 = FALSE 1 = TRUE
FIFO_STAT	10	R	-	0 = FALSE 1 = TRUE
FIFO_ACCESS	9	R	-	0 = FALSE 1 = TRUE
ADDITIONAL_FEAT	8	R	-	0 = FALSE 1 = TRUE
SIR_LP_MODE	7	R	-	0 = FALSE 1 = TRUE
SIR_MODE	6	R	-	0 = FALSE 1 = TRUE
THRE_MODE	5	R	-	0 = FALSE 1 = TRUE
AFCE_MODE	4	R	-	0 = FALSE 1 = TRUE
Rvd	3:2	R	-	
APB_DATA_W	1:0	R	-	00 = 8 bits 01 = 16 bits

IDTH				10 = 32 bits 11 = reserved

Module::MIS	Register::U2UCV	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B4F8
Name	**Bits**	**Read/Write**	**Reset State**	**Comments**	
UCV	31:0	R	-	ASCII value for each number in the version, followed by *. For example 32_30_31_2A represents the version 2.01*	

Module::MIS	Register::U2CTR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B4FC
Name	**Bits**	**Read/Write**	**Reset State**	**Comments**	
CTR	31:0	R	-	This register contains the peripherals identification code.	

3.1 Register

3.1.1 Register summery

Physical Address	*Name*	*Width*	*R/W*	*Description*
0x9801_B700	IC2_CON	7 bits	R/W	I2C Control
0x9801_B704	IC2_TAR	12 bits	R/W	I2C Target Address
0x9801_B708	IC2_SAR	10 bits	R/W	I2C Slave Address
0x9801_B70C	IC2_HS_MADDR	3 bits	R/W	I2C HS Master Mode Code Address
0x9801_B710	IC2_DATA_CMD	9(W) 8 (R)	R/W	I2C RX/TX Data Buffer and Command
0x9801_B714	IC2_SS_SCL_HCNT	16 bits	R/W	Standard speed I2C Clock SCL High Count
0x9801_B718	IC2_SS_SCL_LCNT	16 bits	R/W	Standard speed I2C Clock SCL Low Count
0x9801_B71C	IC2_FS_SCL_HCNT	16 bits	R/W	Fast speed I2C Clock SCL High Count
0x9801_B720	IC2_FS_SCL_LCNT	16 bits	R/W	Fast speed I2C Clock SCL Low Count
0x9801_B724	Rvd			
0x9801_B728	Rvd			
0x9801_B72C	IC2_INTR_STAT	12 bits	R	I2C Interrupt Status
0x9801_B730	IC2_INTR_MASK	12 bits	R/W	I2C Interrupt Mask
0x9801_B734	IC2_RAW_INTR_STAT	12 bits	R	I2C Raw Interrupt Status
0x9801_B738	IC2_RX_TL	8 bits	R/W	I2C Receive FIFO Threshold
0x9801_B73C	IC2_TX_TL	8 bits	R/W	I2C Transmit FIFO Threshold
0x9801_B740	IC2_CLR_INTR	1 bit	R	Clear Combined and Individual Interrupts
0x9801_B744	IC2_CLR_RX_UNDER	1 bit	R	Clear RX_UNDER Interrupt
0x9801_B748	IC2_CLR_RX_OVER	1 bit	R	Clear RX_OVER Interrupt
0x9801_B74C	IC2_CLR_TX_OVER	1 bit	R	Clear TX_OVER Interrupt
0x9801_B750	IC2_CLR_RD_REQ	1 bit	R	Clear RD_REQ Interrupt
0x9801_B754	IC2_CLR_TX_ABRT	1 bit	R	Clear TX_ABRT Interrupt
0x9801_B758	IC2_CLR_RX_DONE	1 bit	R	Clear RX_DONE Interrupt
0x9801_B75c	IC2_CLR_ACTIVITY	1 bit	R	Clear ACTIVITY Interrupt
0x9801_B760	IC2_CLR_STOP_DET	1 bit	R	Clear STOP_DET Interrupt
0x9801_B764	IC2_CLR_START_DET	1 bit	R	Clear START_DET Interrupt
0x9801_B768	IC2_CLR_GEN_CALL	1 bit	R	Clear GEN_CALL Interrupt
0x9801_B76C	IC2_ENABLE	1 bit	R/W	I2C Enable
0x9801_B770	IC2_STATUS	5 bits	R	I2C Status register
0x9801_B774	IC2_TXFLR	4bits	R	Transmit FIFO Level Register
0x9801_B778	IC2_RXFLR	4 bits	R	Receive FIFO Level Register
0x9801_B77C	IC2_SDA_HOLD	16 bits	R/W	I2C SDA Hold Time Length Register
0x9801_B780	IC2_TX_ABRT_SOURCE	16 bits	R/W	I2C Transmit Abort Status Register
0x9801_B784	IC2_SLV_DATA_NACK_ONLY	1 bit	R/W	Generate Slave Data NACK Register
0x9801_B788	IC2_DMA_CR	2 bits	R/W	DMA Control Register for transmit and receive handshaking interface

0x9801_B78c	IC2_DMA_TDLR	3bits	R/W	DMA Transmit Data Level
0x9801_B790	IC2_DMA_RDLR	3bits	R/W	DMA Receive Data Level
0x9801_B794	IC2_SDA_SETUP	8 bits	R/W	I2C SDA Setup Register
0x9801_B798	IC2_ACK_GENERAL_CALL	1 bit	R/W	I2C ACK General Call Register
0x9801_B79C	IC2_ENABLE_STATUS	3 bits	R	I2C Enable Status Register
0x9801_B7F4	IC2_COMP_PARAM_1	32 bits	R	Component Parameter register
0x9801_B7F8	IC2_COMP_VERSION	32 bits	R	Component version ID
0x9801_B7FC	IC2_COMP_TYPE	32 bits	R	DW Component Type Register

Physical Address	*Name*	*Width*	*R/W*	*Description*
0x9801_B900	IC3_CON	7 bits	R/W	I2C Control
0x9801_B904	IC3_TAR	12 bits	R/W	I2C Target Address
0x9801_B908	IC3_SAR	10 bits	R/W	I2C Slave Address
0x9801_B90C	IC3_HS_MADDR	3 bits	R/W	I2C HS Master Mode Code Address
0x9801_B910	IC3_DATA_CMD	9(W) 8 (R)	R/W	I2C RX/TX Data Buffer and Command
0x9801_B914	IC3_SS_SCL_HCNT	16 bits	R/W	Standard speed I2C Clock SCL High Count
0x9801_B918	IC3_SS_SCL_LCNT	16 bits	R/W	Standard speed I2C Clock SCL Low Count
0x9801_B91C	IC3_FS_SCL_HCNT	16 bits	R/W	Fast speed I2C Clock SCL High Count
0x9801_B920	IC3_FS_SCL_LCNT	16 bits	R/W	Fast speed I2C Clock SCL Low Count
0x9801_B924	Rvd			
0x9801_B928	Rvd			
0x9801_B92C	IC3_INTR_STAT	12 bits	R	I2C Interrupt Status
0x9801_B930	IC3_INTR_MASK	12 bits	R/W	I2C Interrupt Mask
0x9801_B934	IC3_RAW_INTR_STAT	12 bits	R	I2C Raw Interrupt Status
0x9801_B938	IC3_RX_TL	8 bits	R/W	I2C Receive FIFO Threshold
0x9801_B93C	IC3_TX_TL	8 bits	R/W	I2C Transmit FIFO Threshold
0x9801_B940	IC3_CLR_INTR	1 bit	R	Clear Combined and Individual Interrupts
0x9801_B944	IC3_CLR_RX_UNDER	1 bit	R	Clear RX_UNDER Interrupt
0x9801_B948	IC3_CLR_RX_OVER	1 bit	R	Clear RX_OVER Interrupt
0x9801_B94C	IC3_CLR_TX_OVER	1 bit	R	Clear TX_OVER Interrupt
0x9801_B950	IC3_CLR_RD_REQ	1 bit	R	Clear RD_REQ Interrupt
0x9801_B954	IC3_CLR_TX_ABRT	1 bit	R	Clear TX_ABRT Interrupt
0x9801_B958	IC3_CLR_RX_DONE	1 bit	R	Clear RX_DONE Interrupt
0x9801_B95c	IC3_CLR_ACTIVITY	1 bit	R	Clear ACTIVITY Interrupt
0x9801_B960	IC3_CLR_STOP_DET	1 bit	R	Clear STOP_DET Interrupt
0x9801_B964	IC3_CLR_START_DET	1 bit	R	Clear START_DET Interrupt
0x9801_B968	IC3_CLR_GEN_CALL	1 bit	R	Clear GEN_CALL Interrupt
0x9801_B96C	IC3_ENABLE	1 bit	R/W	I2C Enable
0x9801_B970	IC3_STATUS	5 bits	R	I2C Status register
0x9801_B974	IC3_TXFLR	4bits	R	Transmit FIFO Level Register
0x9801_B978	IC3_RXFLR	4 bits	R	Receive FIFO Level Register
0x9801_B97C	IC3_SDA_HOLD	16 bits	R/W	I2C SDA Hold Time Length Register
0x9801_B980	IC3_TX_ABRT_SOURCE	16 bits	R/W	I2C Transmit Abort Status Register
0x9801_B984	IC3_SLV_DATA_NAC	1 bit	R/W	Generate Slave Data NACK Register

	K_ONLY			
0x9801_B988	IC3_DMA_CR	2 bits	R/W	DMA Control Register for transmit and receive handshaking interface
0x9801_B98c	IC3_DMA_TDLR	3bits	R/W	DMA Transmit Data Level
0x9801_B990	IC3_DMA_RDLR	3bits	R/W	DMA Receive Data Level
0x9801_B994	IC3_SDA_SETUP	8 bits	R/W	I2C SDA Setup Register
0x9801_B998	IC3_ACK_GENERAL_ CALL	1 bit	R/W	I2C ACK General Call Register
0x9801_B99C	IC3_ENABLE_STATUS	3 bits	R	I2C Enable Status Register
0x9801_B9F4	IC3_COMP_PARAM_1	32 bits	R	Component Parameter register
0x9801_B9F8	IC3_COMP_VERSION	32 bits	R	Component version ID
0x9801_B9FC	IC3_COMP_TYPE	32 bits	R	DW Component Type Register

Physical Address	***Name***	***Width***	***R/W***	***Description***
0x9801_BA00	IC4_CON	7 bits	R/W	I2C Control
0x9801_BA04	IC4_TAR	12 bits	R/W	I2C Target Address
0x9801_BA08	IC4_SAR	10 bits	R/W	I2C Slave Address
0x9801_BA0C	IC4_HS_MADDR	3 bits	R/W	I2C HS Master Mode Code Address
0x9801_BA10	IC4_DATA_CMD	9(W) 8 (R)	R/W	I2C RX/TX Data Buffer and Command
0x9801_BA14	IC4_SS_SCL_HCNT	16 bits	R/W	Standard speed I2C Clock SCL High Count
0x9801_BA18	IC4_SS_SCL_LCNT	16 bits	R/W	Standard speed I2C Clock SCL Low Count
0x9801_BA1C	IC4_FS_SCL_HCNT	16 bits	R/W	Fast speed I2C Clock SCL High Count
0x9801_BA20	IC4_FS_SCL_LCNT	16 bits	R/W	Fast speed I2C Clock SCL Low Count
0x9801_BA24	Rvd			
0x9801_BA28	Rvd			
0x9801_BA2C	IC4_INTR_STAT	12 bits	R	I2C Interrupt Status
0x9801_BA30	IC4_INTR_MASK	12 bits	R/W	I2C Interrupt Mask
0x9801_BA34	IC4_RAW_INTR_STAT	12 bits	R	I2C Raw Interrupt Status
0x9801_BA38	IC4_RX_TL	8 bits	R/W	I2C Receive FIFO Threshold
0x9801_BA3C	IC4_TX_TL	8 bits	R/W	I2C Transmit FIFO Threshold
0x9801_BA40	IC4_CLR_INTR	1 bit	R	Clear Combined and Individual Interrupts
0x9801_BA44	IC4_CLR_RX_UNDER	1 bit	R	Clear RX_UNDER Interrupt
0x9801_BA48	IC4_CLR_RX_OVER	1 bit	R	Clear RX_OVER Interrupt
0x9801_BA4C	IC4_CLR_TX_OVER	1 bit	R	Clear TX_OVER Interrupt
0x9801_BA50	IC4_CLR_RD_REQ	1 bit	R	Clear RD_REQ Interrupt
0x9801_BA54	IC4_CLR_TX_ABRT	1 bit	R	Clear TX_ABRT Interrupt
0x9801_BA58	IC4_CLR_RX_DONE	1 bit	R	Clear RX_DONE Interrupt
0x9801_BA5c	IC4_CLR_ACTIVITY	1 bit	R	Clear ACTIVITY Interrupt
0x9801_BA60	IC4_CLR_STOP_DET	1 bit	R	Clear STOP_DET Interrupt
0x9801_BA64	IC4_CLR_START_DET	1 bit	R	Clear START_DET Interrupt
0x9801_BA68	IC4_CLR_GEN_CALL	1 bit	R	Clear GEN_CALL Interrupt
0x9801_BA6C	IC4_ENABLE	1 bit	R/W	I2C Enable
0x9801_BA70	IC4_STATUS	5 bits	R	I2C Status register
0x9801_BA74	IC4_TXFLR	4bits	R	Transmit FIFO Level Register
0x9801_BA78	IC4_RXFLR	4 bits	R	Receive FIFO Level Register
0x9801_BA7C	IC4_SDA_HOLD	16 bits	R/W	I2C SDA Hold Time Length Register

0x9801_BA80	IC4_TX_ABRT_SOURCE	16 bits	R/W	I2C Transmit Abort Status Register
0x9801_BA84	IC4_SLV_DATA_NACK_ONLY	1 bit	R/W	Generate Slave Data NACK Register
0x9801_BA88	IC4_DMA_CR	2 bits	R/W	DMA Control Register for transmit and receive handshaking interface
0x9801_BA8c	IC4_DMA_TDLR	3bits	R/W	DMA Transmit Data Level
0x9801_BA90	IC4_DMA_RDLR	3bits	R/W	DMA Receive Data Level
0x9801_BA94	IC4_SDA_SETUP	8 bits	R/W	I2C SDA Setup Register
0x9801_BA98	IC4_ACK_GENERAL_CALL	1 bit	R/W	I2C ACK General Call Register
0x9801_BA9C	IC4_ENABLE_STATUS	3 bits	R	I2C Enable Status Register
0x9801_BAF4	IC4_COMP_PARAM_1	32 bits	R	Component Parameter register
0x9801_BAF8	IC4_COMP_VERSION	32 bits	R	Component version ID
0x9801_BAFC	IC4_COMP_TYPE	32 bits	R	DW Component Type Register

Physical Address	*Name*	*Width*	*R/W*	*Description*
0x9801_BB00	IC5_CON	7 bits	R/W	I2C Control
0x9801_BB04	IC5_TAR	12 bits	R/W	I2C Target Address
0x9801_BB08	IC5_SAR	10 bits	R/W	I2C Slave Address
0x9801_BB0C	IC5_HS_MADDR	3 bits	R/W	I2C HS Master Mode Code Address
0x9801_BB10	IC5_DATA_CMD	9(W) 8 (R)	R/W	I2C RX/TX Data Buffer and Command
0x9801_BB14	IC5_SS_SCL_HCNT	16 bits	R/W	Standard speed I2C Clock SCL High Count
0x9801_BB18	IC5_SS_SCL_LCNT	16 bits	R/W	Standard speed I2C Clock SCL Low Count
0x9801_BB1C	IC5_FS_SCL_HCNT	16 bits	R/W	Fast speed I2C Clock SCL High Count
0x9801_BB20	IC5_FS_SCL_LCNT	16 bits	R/W	Fast speed I2C Clock SCL Low Count
0x9801_BB24	Rvd			
0x9801_BB28	Rvd			
0x9801_BB2C	IC5_INTR_STAT	12 bits	R	I2C Interrupt Status
0x9801_BB30	IC5_INTR_MASK	12 bits	R/W	I2C Interrupt Mask
0x9801_BB34	IC5_RAW_INTR_STAT	12 bits	R	I2C Raw Interrupt Status
0x9801_BB38	IC5_RX_TL	8 bits	R/W	I2C Receive FIFO Threshold
0x9801_BB3C	IC5_TX_TL	8 bits	R/W	I2C Transmit FIFO Threshold
0x9801_BB40	IC5_CLR_INTR	1 bit	R	Clear Combined and Individual Interrupts
0x9801_BB44	IC5_CLR_RX_UNDER	1 bit	R	Clear RX_UNDER Interrupt
0x9801_BB48	IC5_CLR_RX_OVER	1 bit	R	Clear RX_OVER Interrupt
0x9801_BB4C	IC5_CLR_TX_OVER	1 bit	R	Clear TX_OVER Interrupt
0x9801_BB50	IC5_CLR_RD_REQ	1 bit	R	Clear RD_REQ Interrupt
0x9801_BB54	IC5_CLR_TX_ABRT	1 bit	R	Clear TX_ABRT Interrupt
0x9801_BB58	IC5_CLR_RX_DONE	1 bit	R	Clear RX_DONE Interrupt
0x9801_BB5c	IC5_CLR_ACTIVITY	1 bit	R	Clear ACTIVITY Interrupt
0x9801_BB60	IC5_CLR_STOP_DET	1 bit	R	Clear STOP_DET Interrupt
0x9801_BB64	IC5_CLR_START_DET	1 bit	R	Clear START_DET Interrupt
0x9801_BB68	IC5_CLR_GEN_CALL	1 bit	R	Clear GEN_CALL Interrupt
0x9801_BB6C	IC5_ENABLE	1 bit	R/W	I2C Enable
0x9801_BB70	IC5_STATUS	5 bits	R	I2C Status register

0x9801_BB74	IC5_TXFLR	4bits	R	Transmit FIFO Level Register
0x9801_BB78	IC5_RXFLR	4 bits	R	Receive FIFO Level Register
0x9801_BB7C	IC5_SDA_HOLD	16 bits	R/W	I2C SDA Hold Time Length Register
0x9801_BB80	IC5_TX_ABRT_SOURCE	16 bits	R/W	I2C Transmit Abort Status Register
0x9801_BB84	IC5_SLV_DATA_NACK_ONLY	1 bit	R/W	Generate Slave Data NACK Register
0x9801_BB88	IC5_DMA_CR	2 bits	R/W	DMA Control Register for transmit and receive handshaking interface
0x9801_BB8c	IC5_DMA_TDLR	3bits	R/W	DMA Transmit Data Level
0x9801_BB90	IC5_DMA_RDLR	3bits	R/W	DMA Receive Data Level
0x9801_BB94	IC5_SDA_SETUP	8 bits	R/W	I2C SDA Setup Register
0x9801_BB98	IC5_ACK_GENERAL_CALL	1 bit	R/W	I2C ACK General Call Register
0x9801_BB9C	IC5_ENABLE_STATUS	3 bits	R	I2C Enable Status Register
0x9801_BBF4	IC5_COMP_PARAM_1	32 bits	R	Component Parameter register
0x9801_BBF8	IC5_COMP_VERSION	32 bits	R	Component version ID
0x9801_BBFC	IC5_COMP_TYPE	32 bits	R	DW Component Type Register

3.1.2 Second I2C Register Description

IC2_CON

- **Name:** I2C Control Register 0
- **Size:** 7 bits
- **Address Offset:** 0x00
- **Read/Write Access:** Read/Write

This register can be written only when the DW_apb_i2c is disabled, which corresponds to the IC_ENABLE register being set to 0. Writes at other times have no effect.

Module::MIS	Register::IC2_CON	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B700

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:7	-	-	-
IC_SLAVE_DISABLE	6	R/W	'b1	This bit controls whether I2C has its slave disabled after reset. The slave can be disabled by programming a '1' into IC_CON[6]. B y default the slave is enabled. 0: slave is enabled 1: slave is disabled
IC_RESTART_EN	5	R/W	'b1	Determines whether RESTART conditions may be sent when acting as a master. Some older slaves do not support handling RESTART conditions; however, RESTART conditions are used in

				several DW_apb_i2c operations. 0: disable 1: enable When RESTART is disabled, the master is prohibited from performing the following functions: ● Change direction within a transfer (split) ● Send a START BYTE ● High-speed mode operation ● Combined format transfers in 7-bit addressing modes ● Read operation with a 10-bit address ● Send multiple bytes per transfer By replacing RESTART condition followed by a STOP and a subsequent START condition, split operations are broken down into multiple DW_apb_i2c transfers. If the above operations are performed, it will result in setting bit 6 (TX_ABRT) of the IC_RAW_INTR_STAT register.
IC_10BITADDR_MASTER	4	R/W	‘b1	This bit controls whether the DW_apb_i2c starts its transfers in 10-bit addressing mode when acting as a master. 0: 7-bit addressing 1: 10-bit addressing
IC_10BITADDR_SLAVE	3	R/W	‘b1	When acting as a slave, this bit controls whether the DW_apb_i2c responds to 7- or 10-bit addresses. 0: 7-bit addressing. The DW_apb_i2c ignores transactions which involve 10-bit addressing; for 7-bit addressing, only the lower 7 bits of the IC_SAR register are compared. 1: 10-bit addressing. The DW_apb_i2c responds to only 10-bit addressing transfers that match the full 10 bits of the IC_SAR register.
SPEED	2:1	R/W	’b10	Controls at which speed the DW_apb_i2c operates: 0: illegal; writing a 0 results in setting SPEED to IC_MAX_SPEED_MODE 1: standard mode (100 kbit/s) 2: fast mode (400 kbit/s) 3: high speed mode (3.4 Mbit/s) If the DW_apb_i2c is configured for fast or

				standard mode (1 or 2) and a value of 2 or 3 is written, then IC_MAX_SPEED_MODE is stored.
MASTER_MODE	0	R/W	'b1	This bit controls whether the DW_apb_i2c master is enabled or not. The slave is always enabled. 0: master disabled 1: master enabled

IC2_TAR

- **Name:** I2C Target Address Register
- **Size:** 12 bits
- **Address Offset:** 0x04
- **Read/Write Access:** Read/Write

All bits can be dynamically updated as long as any set of the following conditions are true:

- DW_apb_i2c is NOT enabled (IC_ENABLE is set to 0); or
- DW_apb_i2c is enabled (IC_ENABLE=1); AND

DW_apb_i2c is NOT engaged in any Master (tx, rx) operation (IC_STATUS[5]=0); AND
DW_apb_i2c is enabled to operate in Master mode (IC_CON[0]=1); AND
there are NO entries in the TX FIFO (IC_STATUS[2]=1)

Module::MIS	Register::IC2_TAR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B704

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:13	-	-	-
IC_10BITADDR_MASTER	12	R/W	'b0	This bit controls whether the DW_apb_i2c starts its transfers in 7-or 10-bit addressing mode when acting as a master. 0: 7-bit addressing 1: 10-bit addressing Dependencies: This bit exists in this register only if the I2C_DYNAMIC_TAR_UPDATE configuration parameter is set to "Yes" (1).
SPECIAL	11	R/W	'b0	This bit indicates whether software performs a General Call or START BYTE command. 0: ignore bit 10 GC_OR_START and use IC_TAR normally 1: perform special I2C command as specified in GC_OR_START bit
GC_OR_START	10	R/W	'b0	If bit 11 (SPECIAL) is set to 1, then this bit indicates whether a General Call or START byte command is to be performed by the DW_apb_i2c. 0: General Call Address – after issuing a General Call, only writes may be performed. Attempting to issue a read command results in setting bit 6 (TX_ABRT) of the IC_RAW_INTR_STAT register. The DW_apb_i2c remains in General Call mode until the

				SPECIAL bit value (bit 11) is cleared. 1: START BYTE
IC_TAR	9:0	R/W	'h055	This is the target address for any master transaction. When transmitting a General Call, these bits are ignored. To generate a START BYTE, the CPU needs to write only once into these bits. If the IC_TAR and IC_SAR are the same, loopback exists but the FIFOs are shared between master and slave, so full loopback is not feasible. Only one direction loopback mode is supported (simplex), not duplex. A master cannot transmit to itself; it can transmit to only a slave.

IC2_SAR

- **Name:** I2C Slave Address Register
- **Size:** 10 bits
- **Address Offset:** 0x08
- **Read/Write Access:** Read/Write

Module::MIS	Register::IC2_SAR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B708

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:10	-	-	-
IC_SAR	9:0	R/W	'h055	The IC_SAR holds the slave address when the I2C is operating as a slave. IC_SAR holds the slave address to which the DW_apb_i2c responds. For 7-bit addressing, only IC_SAR[6:0] is used. This register can be written only when the I2C interface is disabled, which corresponds to the *IC_ENABLE* register being set to 0. Writes at other times have no effect.

IC2_HS_MADDR

- **Name:** I2C HS Master Mode Code Address Register
- **Size:** 3 bits
- **Address Offset:** 0x0c
- **Read/Write Access:** Read/Write

Module::MIS	Register::IC2_HS_MADDR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B70C

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:3	-	-	-
IC_HS_MAR	2:0	R/W	'b0	This bit field holds the value of the I2C HS mode master code. HS-mode master codes are reserved 8-bit codes (00001xxx) that are not used for slave addressing or other purposes. Each master has its unique master code; up to eight high-speed mode masters can be present on the same I2C bus system. Valid values are from 0 to 7. This register goes away and

				becomes read-only returning 0's if the IC_MAX_SPEED_MODE configuration parameter is set to either Standard (1) or Fast (2). This register can be written only when the I2C interface is disabled, which corresponds to the IC_ENABLE register being set to 0. Writes at other times have no effect.

IC2_DATA_CMD

- **Name:** I2C Rx/Tx Data Buffer and Command Register; this is the register the CPU writes to when filling the TX FIFO and the CPU reads from when retrieving bytes from RX FIFO
- **Size:** 11 bits (writes)
- **Address Offset:** 0x10
- **Read/Write Access:** Read/Write

Module::MIS	Register::IC2_DATA_CMD	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B710

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:11	-	-	-
RESTART	10	W	'b0	This bit controls whether a RESTART is issued before the byte is sent or received. Thie bit is available only if IC_EMPTYFIFO_HOLD_MASTER_EN is configured to 1. 1 – If IC_RESTART_EN is 1, a RESTART is issued before the data is sent/received (according to the value of CMD), regardless of whether of not the transfer direction is changing from the previous command; if IC_RESTART_EN is 0, a STOP followed by a START is issued instead. 0 – If IC_RESTART_EN is 1, a RESTART is issued only if the transfer direction is changing from the previous command; if IC_RESTART_EN is 0, a STOP followed by a START is issued instead.
STOP	9	W	'b0	This bit controls whether a STOP is issued after the byte is sent or received. This bit is available only if IC_EMPTYFIFO_HOLD_MASTER_EN is configured to 1. 1 – STOP is issued after this byte, regardless of whether of not the Tx FIFO is empty. If the Tx FIFO is not empty, the master immediately tries to start a new transfer by issuing a START and arbitrating for the bus. 0 – STOP is not issued after this byte, regardless of whether or not the Tx FIFO is empty. If the Tx FIFO is not empty, the master continues the current transfer by sending/receiving data bytes according to the value of the CMD bit. If the Tx FIFO is empty, the master holds the SCL line low and stalls the bus until a new command is available in the Tx FIFO.
CMD	8	W	'b0	This bit controls whether a read or a write is performed. This bit does not control the direction when the DW_apb_i2c acts as a slave. It controls only the direction when it acts as a master. 1 = Read 0 = Write When a command is entered in the TX FIFO, this bit

				distinguishes the write and read commands. In slave-receiver mode, this bit is a “don’t care” because writes to this register are not required. In slave-transmitter mode, a “0” indicates that CPU data is to be transmitted and as DAT or IC_DATA_CMD[7:0]. When programming this bit, you should remember the following: attempting to perform a read operation after a General Call command has been sent results in a TX_ABRT interrupt (bit 6 of the IC_RAW_INTR_STAT register), unless bit 11 (SPECIAL) in the IC_TAR register has been cleared. If a “1” is written to this bit after receiving a RD_REQ interrupt, then a TX_ABRT interrupt occurs. NOTE: It is possible that while attempting a master I2C read transfer on DW_apb_i2c, a RD_REQ interrupt may have occurred simultaneously due to a remote I2C master addressing DW_apb_i2c. In this type of scenario, DW_apb_i2c ignores the IC_DATA_CMD write, generates a TX_ABRT interrupt, and waits to service the RD_REQ interrupt.
DAT	7:0	R/W	‘b0	This register contains the data to be transmitted or received on the I2C bus. If you are writing to this register and want to perform a read, bits 7:0 (DAT) are ignored by the DW_apb_i2c. However, when you read this register, these bits return the value of data received on the DW_apb_i2c interface.

IC2_SS_SCL_HCNT

- **Name:** Standard Speed I2C Clock SCL High Count Register
- **Size:** 16 bits
- **Address Offset:** 0x14
- **Read/Write Access:** Read/Write

Module::MIS	Register::IC2_SS_SCL_HCNT	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B714

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:16	-	-	-
IC_SS_SCL_HCNT	15:0	R/W	’h007a	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high-period count for standard speed. The table below shows some sample IC_SS_HCNT calculations. This register can be written only when the I2C interface is disabled which corresponds to the *IC_ENABLE*

				register being set to 0. Writes at other times have no effect. The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set. For designs with APB_DATA_WIDTH = 8 the order of programming is important to ensure the correct operation of the DW_apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed. When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only.

Notice: Read-only if IC_HC_COUNT_VALUES = 1.

I2C Data Rate (kbps)	ic_clk freq (MHz)	SCL High required min (us)	H_CNT (HEX/decimal)	SCL High Time (us)
100	2	4	0008/8	4.00
100	6.6	4	001B/27	4.09
100	10	4	0028/40	4.00
100	75	4	012C/300	4.00
100	100	4	0190/400	4.00
100	125	4	01F4/500	4.00
100	1000	4	0FA0/4000	4.00

IC2_SS_SCL_LCNT

- **Name:** Standard Speed I2C Clock SCL Low Count Register
- **Size:** 16 bits
- **Address Offset:** 0x98
- Read/Write Access: Read/Write

Module::MIS	Register::IC2_SS_SCL_LCNT	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B718

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:16	-	-	-
IC_SS_SCL_LCNT	15:0	R/W	'h008f	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for standard speed. The table below shows some sample IC_SS_LCNT calculations. This register can be written only when the I2C interface is disabled which corresponds to the *IC_ENABLE* register being set to 0. Writes at other times have no effect. The minimum valid value is 8; hardware prevents values

				less than this being written, and if attempted results in 8 being set. For designs with APB_DATA_WIDTH = 8 the order of programming is important to ensure the correct operation of the DW_apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed. When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only.

Notice : Read-only if IC_HC_COUNT_VALUES = 1.

I^2C Data Rate (kbps)	ic_clk_{freq} (MHz)	SCL Low required min (us)	L_CNT (HEX/Decimal)	SCL Low Time (us)
100	2	4.7	000A/10	5.00
100	6.6	4.7	0020/32	4.85
100	10	4.7	002F/47	4.70
100	75	4.7	0161/353	4.71
100	100	4.7	01D6/470	4.70
100	125	4.7	024C/588	4.70
100	1000	4	125C/4700	4.70

IC2_FS_SCL_HCNT

- **Name:** Fast Speed I2C Clock SCL High Count Register
- **Size:** 16 bits
- **Address Offset:** 0x1c
- **Read/Write Access:** Read/Write

Module::MIS	Register::IC2_FS_SCL_HCNT	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B71C

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:16	-	-	-
IC_FS_SCL_HCNT	15:0	R/W	'h0013	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high-period count for fast speed. It is used in high-speed mode to send the Master Code and START BYTE or General CALL. The table below shows some sample IC_FS_SCL_HCNT calculations. This register goes away and becomes read-only returning 0s if IC_MAX_SPEED_MODE = standard. This register can be written only when the I2C interface is disabled, which corresponds to the *IC_ENABLE* register being set to 0. Writes at other times have no effect.

				The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set. For designs with APB_DATA_WIDTH == 8 the order of programming is important to ensure the correct operation of the DW_apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed. When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only.

Notice: Read-only if IC_HC_COUNT_VALUES = 1.

I²C Data Rate (kbps)	ic_clk_{freq} (MHz)	SCL High required min (us)	H_CNT (HEX/Decimal)	SCL High Time (us)
400	10	0.6	0006/6	0.60
400	25	0.6	000F/15	0.60
400	50	0.6	001E/	0.60
400	75	0.6	002D/30	0.60
400	100	0.6	003C/60	0.60
400	125	0.6	004B/75	0.60
400	1000	0.6	0258/600	0.60
100 (through IC_MAX_SPEED_MODE = standard)	N/A	N/A	disabled	N/A

IC2_FS_SCL_LCNT

- **Name:** Fast Speed I2C Clock SCL Low Count Register
- **Size:** 16 bits
- **Address Offset:** 0x20
- **Read/Write Access:** Read/Write

Module::MIS	Register::IC2_FS_SCL_LCNT	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B720

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:16	-	-	-
IC_FS_SCL_LCNT	15:0	R/W	'h0028	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for fast speed. It is used in high-speed mode to send the Master Code and START BYTE or General CALL. The table below shows some sample IC_FS_SCL_LCNT calculations.

				This register goes away and becomes read-only returning 0s if IC_MAX_SPEED_MODE = standard. This register can be written only when the I2C interface is disabled, which corresponds to the *IC_ENABLE* register being set to 0. Writes at other times have no effect. The minimum valid value is 8; hardware prevents values less than this being written, and if attempted results in 8 being set. For designs with APB_DATA_WIDTH = 8 the order of programming is important to ensure the correct operation of the DW_apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed. If the value is less than 8 then the count value gets changed to 8. When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only.

Notice : Read-only if IC_HC_COUNT_VALUES = 1.

I2C Data Rate (kbps)	ic_clk_freq (MHz)	SCL Low required min (us)	L_CNT (HEX/Decimal)	SCL Low Time (us)
400	10	1.3	000D/13	1.30
400	25	1.3	0021/33	1.32
400	50	1.3	0041/65	1.30
400	75	1.3	0062/98	1.31
400	100	1.3	0082/130	1.30
400	125	1.3	00A3/163	1.30
400	1000	1.3	0514/1300	1.30
100 (through IC_MAX_SPEED_MODE = standard)	N/A	N/A	disabled	N/A

IC2_INTR_STAT

- **Name:** I2C Interrupt Status Register
- **Size:** 12 bits
- **Address Offset:** 0x2C
- **Read/Write Access:** Read

Each bit in this register has a corresponding mask bit in the IC_INTR_MASK register. These bits are cleared by reading the matching interrupt clear register. The unmasked raw versions of these bits are available in the IC_RAW_INTR_STAT register.

Module::MIS	Register::IC2_INTR_STAT		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B72C
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:12	-	-	-		
R_GEN_CALL	11	R	‘b0	See “IC_RAW_INTR_STAT” for a detailed description of these bits.		
R_START_DET	10	R	‘b0			
R_STOP_DET	9	R	‘b0			
R_ACTIVITY	8	R	‘b0			
R_RX_DONE	7	R	‘b0			
R_TX_ABRT	6	R	‘b0			
R_RD_REQ	5	R	‘b0			
R_TX_EMPTY	4	R	‘b0			
R_TX_OVER	3	R	‘b0			
R_RX_FULL	2	R	‘b0			
R_RX_OVER	1	R	‘b0			
R_RX_UNDER	0	R	‘b0			

IC2_INTR_MASK

- **Name:** I2C Interrupt Mask Register
- **Size:** 12 bits
- **Address Offset:** 0x30
- **Read/Write Access:** Read/Write

These bits mask their corresponding interrupt status bits. They are active high; a value of 0 prevents a bit from generating an interrupt.

Module::MIS	Register::IC2_INTR_MASK		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B730
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:12	-	-	-		
M_GEN_CALL	11	R/W	‘b1	Masks this bit in the IC_INTR_STAT register.		
M_START_DET	10	R/W	‘b0			
M_STOP_DET	9	R/W	‘b0			
M_ACTIVITY	8	R/W	‘b0			
M_RX_DONE	7	R/W	‘b1			
M_TX_ABRT	6	R/W	‘b1			
M_RD_REQ	5	R/W	‘b1			
M_TX_EMPTY	4	R/W	‘b1			
M_TX_OVER	3	R/W	‘b1			
M_RX_FULL	2	R/W	‘b1			
M_RX_OVER	1	R/W	‘b1			
M_RX_UNDER	0	R/W	‘b1			

IC2_RAW_INTR_STAT

- **Name:** I2C Raw Interpol Status Register
- **Size:** 12 bits
- **Address Offset:** 0x34
- **Read/Write Access:** Read/Write

Unlike the IC_INTR_STAT register, these bits are not masked so they always show the true status of the DW_apb_i2c.

Module::MIS	Register::IC2_RAW_INTR_STAT	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B734

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:12	-	-	-
GEN_CALL	11	R	‘b0	Set only when a General Call address is received and it is acknowledged. It stays set until it is cleared either by disabling DW_apb_i2c or when the CPU reads bit 0 of the IC_CLR_GEN_CALL register. DW_apb_i2c stores the received data in the Rx buffer.
START_DET	10	R	‘b0	Indicates whether a START or RESTART condition has occurred on the I2C interface regardless of whether DW_apb_i2c is operating in slave or master mode.
STOP_DET	9	R	‘b0	Indicates whether a STOP condition has occurred on the I2C interface regardless of whether DW_apb_i2c is operating in slave or master mode.
ACTIVITY	8	R	‘b0	This bit captures DW_apb_i2c activity and stays set until it is cleared. There are four ways to clear it: ● Disabling the DW_apb_i2c ● Reading the IC_CLR_ACTIVITY register ● Reading the IC_CLR_INTR register ● System reset Once this bit is set, it stays set unless one of the four methods is used to clear it. Even if the DW_apb_i2c module is idle, this bit remains set until cleared, indicating that there was activity on the bus.
RX_DONE	7	R	‘b0	When the DW_apb_i2c is acting as a slave-transmitter, this bit is set to 1 if the master does not acknowledge a transmitted byte. This occurs on the last byte of the transmission, indicating that the transmission is done.
TX_ABRT	6	R	‘b0	This bit indicates if DW_apb_i2c, as an I2C transmitter, is unable to complete the intended actions on the contents of the transmit FIFO. This situation can occur both as an I2C master or an I2C slave, and is referred to as a “transmit abort”. When this bit is set to 1, the IC_TX_ABRT_SOURCE register indicates the reason why the transmit abort takes places. NOTE: The DW_apb_i2c flushes/resets/empties the TX FIFO whenever this bit is set. The TX FIFO remains in this flushed

				state until the register IC_CLR_TX_ABRT is read. Once this read is performed, the TX FIFO is then ready to accept more data bytes from the APB interface.
RD_REQ	5	R	‘b0	This bit is set to 1 when DW_apb_i2c is acting as a slave and another I2C master is attempting to read data from DW_apb_i2c. The DW_apb_i2c holds the I2C bus in a wait state (SCL=0) until this interrupt is serviced, which means that the slave has been addressed by a remote master that is asking for data to be transferred. The processor must respond to this interrupt and then write the requested data to the IC_DATA_CMD register. This bit is set to 0 just after the processor reads the IC_CLR_RD_REQ register.
TX_EMPTY	4	R	‘b0	This bit is set to 1 when the transmit buffer is at or below the threshold value set in the IC_TX_TL register. It is automatically cleared by hardware when the buffer level goes above the threshold. When the IC_ENABLE bit 0 is 0, the TX FIFO is flushed and held in reset. There the TX FIFO looks like it has no data within it, so this bit is set to 1, provided there is activity in the master or slave state machines. When there is no longer activity, then with ic_en=0, this bit is set to 0.
TX_OVER	3	R	‘b0	Set during transmit if the transmit buffer is filled to IC_TX_BUFFER_DEPTH and the processor attempts to issue another I2C command by writing to the IC_DATA_CMD register. When the module is disabled, this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.
RX_FULL	2	R	‘b0	Set when the receive buffer reaches or goes above the RX_TL threshold in the IC_RX_TL register. It is automatically cleared by hardware when buffer level goes below the threshold. If the module is disabled (IC_ENABLE[0]=0), the RX FIFO is flushed and held in reset; therefore the RX FIFO is not full. So this bit is cleared once the IC_ENABLE bit 0 is programmed with a 0, regardless of the activity that continues.
RX_OVER	1	R	‘b0	Set if the receive buffer is completely filled to IC_RX_BUFFER_DEPTH and an additional byte is received from an external I2C device. The DW_apb_i2c acknowledges this, but any data bytes received after the FIFO is full are lost. If the module is disabled (IC_ENABLE[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.
RX_UNDER	0	R	‘b0	Set if the processor attempts to read the receive buffer when it is empty by reading from the IC_DATA_CMD register. If the module is disabled (IC_ENABLE[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.

IC2_RX_TL

•**Name:** I2C Receive FIFO Threshold Register
•**Size:** 8bits
•**Address Offset:** 0x38
•**Read/Write Access:** Read/Write

Module::MIS	Register::IC2_RX_TL	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B738

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:8	-	-	-
RX_TL	7:0	R/W	'h00	Receive FIFO Threshold Level Controls the level of entries (or above) that triggers the RX_FULL interrupt. The valid range is 0-255, with the additional restriction that hardware does not allow this value to be set to a value larger than the depth of the buffer. If an attempt is made to do that, the actual value set will be the maximum depth of the buffer. A value of 0 sets the threshold for 1 entry, and a value of 255 sets the threshold for 256 entries.**Depth of the rx_buffer is 128.**

IC2_TX_TL

•**Name:** I2C Transmit FIFO Threshold Register
•**Size:** 8 bits
•**Address Offset:** 0x3c
•Read/Write Access: Read/Write

Module::MIS	Register::IC2_TX_TL	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B73C

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:8	-	-	-
TX_TL	7:0	R/W	'h00	Transmit FIFO Threshold Level Controls the level of entries (or below) that trigger the TX_EMPTY interrupt. The valid range is 0-255, with the additional restriction that it may not be set to value larger than the depth of the buffer. If an attempt is made to do that, the actual value set will be the maximum depth of the buffer. A value of 0 sets the threshold for 0 entries, and a value of 255 sets the threshold for 255 entries.**Depth of the tx_buffer is 128.**

IC2_CLR_INTR

•**Name:** Clear Combined and Individual Interrupt Register

- **Size:** 1 bit
- **Address Offset:** 0x40
- **Read/Write Access:** Read

Module::MIS	Register::**IC2_CLR_INTR**	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B740

Name	**Bits**	**Read /Write**	**Reset State**	**Comments**
Rvd	31:1	-	-	-
CLR_INTR	0	R	'b0	Read this register to clear the combined interrupt, all individual interrupts, and the IC_TX_ABRT_SOURCE register. This bit does not clear hardware clearable interrupts but software clearable interrupts. Refer to Bit 9 of the IC_TX_ABRT_SOURCE register for an exception to clearing IC_TX_ABRT_SOURCE.

IC2_CLR_RX_UNDER

- **Name:** Clear RX_UNDER Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x44
- **Read/Write Access:** Read

Module::MIS	Register::IC2_CLR_RX_UNDER	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B744

Name	**Bits**	**Read /Write**	**Reset State**	**Comments**
Rvd	31:1	-	-	-
CLR_RX_UNDER	0	R	'b0	Read this register to clear the *RX_UNDER* interrupt.

IC2_CLR_RX_OVER

- **Name:** Clear RX_OVER Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x48
- **Read/Write Access:** Read

Module::MIS	Register::IC2_CLR_RX_OVER	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B748

Name	**Bits**	**Read /Write**	**Reset State**	**Comments**
Rvd	31:1	-	-	-
CLR_RX_OVER	0	R	'b0	Read this register to clear the *RX_OVER* interrupt.

IC2_CLR_TX_OVER

- **Name:** Clear TX_OVER Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x4c
- **Read/Write Access:** Read

Module::MIS	Register::IC2_CLR_TX_OVER	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B74C

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:1	-	-	-
CLR_TX_OVER	0	R	'b0	Read this register to clear the *TX_OVER* interrupt.

IC2_CLR_RD_REQ

- **Name:** Clear RD_REQ Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x50
- **Read/Write Access:** Read

Module::MIS	Register::IC2_CLR_RD_REQ	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B750

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:1	-	-	-
CLR_RD_REQ	0	R	'b0	Read this register to clear the *RD_REQ* interrupt.

IC2_CLR_TX_ABRT

- **Name:** Clear TX_ABRT Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x54
- **Read/Write Access:** Read

Module::MIS	Register::IC2_CLR_TX_ABRT	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B754

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:1	-	-	-
CLR_TX_ABRT	0	R	'b0	Read this register to clear the *TX_ABRT* interrupt, and the IC_TX_ABRT_SOURCE register.

IC2_CLR_RX_DONE

- **Name:** Clear RX_DONE Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x58

•**Read/Write Access:** Read

Module::MIS	Register::IC2_CLR_RX_DONE	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B758

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:1	-	-	-
CLR_RX_DONE	0	R	'b0	Read this register to clear the *RX_DONE* interrupt.

IC2_CLR_ACTIVITY

•**Name:** ACTIVITY Status Interrupt Register
•**Size:** 1 bit
•**Address Offset:** 0x5c
•**Read/Write Access:** Read

Module::MIS	Register::IC2_CLR_ACTIVITY	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B75C

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:1	-	-	-
CLR_ACTIVITY	0	R	'b0	Read this register to get status of the *ACTIVITY* interrupt. It is automatically cleared by hardware.

IC2_CLR_STOP_DET

•**Name:** Clear STOP_DET Interrupt Register
•**Size:** 1 bit
•**Address Offset:** 0x60
•**Read/Write Access:** Read

Module::MIS	Register::IC2_CLR_STOP_DET	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B760

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:1	-	-	-
CLR_STOP_DET	0	R	'b0	Read this register to clear the *STOP_DET* interrupt.

IC2_CLR_START_DET

•**Name:** Clear START_DET Interrupt Register
•**Size:** 1 bit
•**Address Offset:** 0x64
•**Read/Write Access:** Read

Module::MIS	Register::IC2_CLR_START_DET	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B764

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:1	-	-	-
CLR_START_DET	0	R	‘b0	Read this register to clear the *START_DET* interrupt.

IC2_CLR_GEN_CALL

- **Name:** Clear GEN_CALL Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x68
- **Read/Write Access:** Read

Module::MIS	Register::IC2_CLR_GEN_CALL	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B768

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:1	-	-	-
CLR_GEN_CALL	0	R	‘b0	Read this register to clear the *GEN_CALL* interrupt.

IC2_ENABLE

- Name: I2C Enable Register
- **Size:** 1 bit
- **Address Offset:** 0x6c
- **Read/Write Access:** Read/Write

Module::MIS	Register::IC2_ENABLE	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B76C

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:1	-	-	-
ENABLE	0	R/W	‘b0	Controls whether the DW_apb_i2c is enabled. 0: Disables DW_apb_i2c (TX and RX FIFOs are held in an erased state) 1: Enables DW_apb_i2c Software can disable DW_apb_i2c while it is active. However, it is important that care be taken to ensure that DW_apb_i2c is disabled properly. When DW_apb_i2c is disabled, the following occurs: • The TX FIFO and RX FIFO get flushed. • Status bits in the IC_INTR_STAT register are still active until DW_apb_i2c goes into IDLE state.If the module is transmitting, it stops as well as deletes the contents of the transmit buffer after the current transfer is complete. If the module is receiving, the DW_apb_i2c stops the current transfer at the end of the current byte and does not acknowledge the transfer.In systems with

				asynchronous pclk and ic_clk when IC_CLK_TYPE parameter set to asynchronous (1), there is a two ic_clk delay when enabling or disabling the DW_apb_i2c.

IC2_STATUS

- **Name:** I2C Status Register
- **Size:** 5 bits
- **Address Offset:** 0x70
- **Read/Write Access:** Read

This is a read-only register used to indicate the current transfer status and FIFO status. The status register may be read at any time. None of the bits in this register request an interrupt.

When the I2C is disabled by writing 0 in bit 0 of the IC_ENABLE register:

- Bits 1 and 2 are set to 1
- Bits 3 and 4 are set to 0

When the master or slave state machines goes to idle and ic_en=0:

- Bits 5 and 6 are set to 0

Module::MIS	Register::IC2_STATUS	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B770

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:7	-	-	-
SLV_ACTIVITY	6	R	'b0	Slave FSM Activity Status. When the Slave Finite State Machine (FSM) is not in the IDLE state, this bit is set. 0: Slave FSM is in IDLE state so the Slave part of DW_apb_i2c is not Active 1: Slave FSM is not in IDLE state so the Slave part of DW_apb_i2c is Active
MST_ACTIVITY	5	R	'b0	Master FSM Activity Status. When the Master Finite State Machine (FSM) is not in the IDLE state, this bit is set. 0: Master FSM is in IDLE state so the Master part of DW_apb_i2c is not Active 1: Master FSM is not in IDLE state so the Master part of DW_apb_i2c is Active
RFF	4	R	'b0	Receive FIFO Completely Full. When the receive FIFO is completely full, this bit is set. When the receive FIFO contains one or more empty location, this bit is cleared. 0 – Receive FIFO is not full 1 – Receive FIFO is full
RFNE	3	R	'b0	Receive FIFO Not Empty. Set when the receive FIFO contains one or more entries and is cleared when the receive FIFO is empty. This bit can be polled by software to completely empty the receive FIFO. 0 – Receive FIFO is empty 1 – Receive FIFO is not empty
TFE	2	R	'b1	Transmit FIFO Completely Empty. When the transmit FIFO is completely empty, this bit is set. When it contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt.

				0 – Transmit FIFO is not empty 1 – Transmit FIFO is empty
TFNF	1	R	'b1	Transmit FIFO Not Full. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full. 0 – Transmit FIFO is full 1 – Transmit FIFO is not full
ACTIVITY	0	R	'b0	I2C Activity Status.

IC2_TXFLR

- **Name:** I2C Transmit FIFO Level Register
- **Size:** 4
- **Address Offset:** 0x74
- **Read/Write Access:** Read

This register contains the number of valid data entries in the transmit FIFO buffer. It is cleared whenever:

- The I2C is disabled
- There is a transmit abort—that is, TX_ABRT bit is set in the IC_RAW_INTR_STAT register
- The slave bulk transmit mode is aborted

The register increments whenever data is placed into the transmit FIFO and decrements when data is taken from the transmit FIFO.

Module::MIS	Register::IC2_TXFLR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B774

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:4	-	-	-
TXFLR	3:0	R	'b0	**Transmit FIFO Level**. Contains the number of valid data entries in the transmit FIFO.

IC2_RXFLR

- **Name:** I2C Receive FIFO Level Register
- **Size:** 4
- **Address Offset:** 0x78
- **Read/Write Access:** Read

This register contains the number of valid data entries in the receive FIFO buffer. It is cleared whenever:

- The I2C is disabled
- Whenever there is a transmit abort caused by any of the events tracked in IC_TX_ABRT_SOURCE

The register increments whenever data is placed into the receive FIFO and decrements when data is

taken from the receive FIFO.

Module::MIS	Register::IC2_RXFLR		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B778
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:4	-	-	-		
RXFLR	3:0	R	'h0	**Receive FIFO Level**. Contains the number of valid data entries in the receive FIFO.		

IC2_SDA_HOLD

- **Name:** I2C SDA Hold Time Length Register
- **Size:** 16
- **Address Offset:** 0x7C
- **Read/Write Access:** Read / Write

This register controls the amount of hold time on the SDA signal after a negative edge of SCL in both master and slave mode, in units of ic_clk period. The value programmed must be greater than the minimum hold time in each mode for the value to be implemented – one cycle in master mode, seven cycles in slave mode. Writes to this register succeed only when IC_ENABLE=0.

The programmed SDA hold time cannot exceed at any time the duration of the low part of SCL. Therefore the programmed value cannot be larger then N_SCL_LOW-2, where N_SCL_LOW is the duration of the low part of the SCL period measured in ic_clk cycles.

Module::MIS	Register::IC2_SDA_HOLD		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B77C
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:16	-	-	-		
SDA_HOLD	15:0	R/W	'h0001	Sets the required SDA hold time in units of ic_clk period.		

IC2_TX_ABRT_SOURCE

- **Name:** I2C Transmit Abort Source Register
- **Size:** 16 bits
- **Address Offset:** 0x80
- **Read/Write Access:** Read

This register has 16 bits that indicate the source of the TX_ABRT bit. Except for Bit 9, this register is cleared whenever the IC_CLR_TX_ABRT register or the IC_CLR_INTR register is read. To clear Bit 9, the source of the ABRT_SBYTE_NORSTRT must be fixed first; RESTART must be enabled (IC_CON[5]=1), the SPECIAL bit must be cleared (IC_TAR[11]), or the GC_OR_START bit must be cleared (IC_TAR[10]). Once the source of the ABRT_SBYTE_NORSTRT is fixed, then this bit can be cleared in the same manner as other bits

in this register. If the source of the ABRT_SBYTE_NORSTRT is not fixed before attempting to clear this bit, Bit 9 clears for one cycle and is then re-asserted.

Module::MIS	Register::IC2_TX_ABRT_SOURCE	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B780

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:16	-	-	-
ABRT_SLVRD_INTX	15	R	'b0	1: When the processor side responds to a slave mode request for data to be transmitted to a remote master and user writes a 1 in CMD (bit 8) of IC_DATA_CMD register.
ABRT_SLV_ARBLOST	14	R	'b0	1: Slave lost the bus while transmitting data to a remote master. IC_TX_ABRT_SOURCE[12] is set at the same time. Note: Even though the slave never "owns" the bus, something could go wrong on the bus. This is a fail safe check. For instance, during a data transmission at the low-to-high transition of SCL, if what is on the data bus is not what is supposed to be transmitted, then DW_apb_i2c no longer own the bus.
ABRT_SLVFLUSH_TXFIFO	13	R	'b0	1: Slave has received a read command and some data exists in the TX FIFO so the slave issues a TX_ABRT interrupt to flush old data in TX FIFO.
ARB_LOST	12	R	'b0	1: Master has lost arbitration, or if IC_TX_ABRT_SOURCE[14] is also set, then the slave transmitter has lost arbitration. Note: I2C can be both master and slave at the same time.
ARB_MASTER_DIS	11	R	'b0	1: User tries to initiate a Master operation with the Master mode disabled.
ABRT_10B_RD_NORSTRT	10	R	'b0	1: The restart is disabled (IC_RESTART_EN bit (IC_CON[5]) = 0) and the master sends a read command in 10-bit addressing mode.
ABRT_SBYTE_NORSTRT	9	R	'b0	To clear Bit 9, the source of the ABRT_SBYTE_NORSTRT must be fixed first; restart must be enabled (IC_CON[5]=1), the SPECIAL bit must be cleared (IC_TAR[11]), or the GC_OR_START bit must be cleared (IC_TAR[10]). Once the source of the ABRT_SBYTE_NORSTRT is fixed, then this bit can be cleared in the same manner as other bits in this register. If the source of the ABRT_SBYTE_NORSTRT is not fixed before

				attempting to clear this bit, bit 9 clears for one cycle and then gets re-asserted. 1: The restart is disabled (IC_RESTART_EN bit (IC_CON[5]) = 0) and the user is trying to send a START Byte.
ABRT_HS_NORSTRT	8	R	'b0	1: The restart is disabled (IC_RESTART_EN bit (IC_CON[5]) = 0) and the user is trying to use the master to transfer data in High Speed mode.
ABRT_SBYTE_ACKDET	7	R	'b0	1: Master has sent a START Byte and the START Byte was acknowledged (wrong behavior).
ABRT_HS_ACKDET	6	R	'b0	1: Master is in High Speed mode and the High Speed Master code was acknowledged (wrong behavior).
ABRT_GCALL_READ	5	R	'b0	1: DW_apb_i2c in master mode sent a General Call but the user programmed the byte following the General Call to be a read from the bus (IC_DATA_CMD[9] is set to 1).
ABRT_GCALL_NOACK	4	R	'b0	1: DW_apb_i2c in master mode sent a General Call and no slave on the bus acknowledged the General Call.
ABRT_TXDATA_NOACK	3	R	'b0	1: This is a master-mode only bit. Master has received an acknowledgement for the address, but when it sent data byte(s) following the address, it did not receive an acknowledge from the remote slave(s).
ABRT_10ADDR2_NOACK	2	R	'b0	1: Master is in 10-bit address mode and the second address byte of the 10-bit address was not acknowledged by any slave.
ABRT_10ADDR1_NOACK	1	R	'b0	1: Master is in 10-bit address mode and the first 10-bit address byte was not acknowledged by any slave.
ABRT_7B_ADDR_NOACK	0	R	'b0	1: Master is in 7-bit addressing mode and the address sent was not acknowledged by any slave.

IC2_SLV_DATA_NACK_ONLY

- Name: Generate Slave Data NACK Register
- Size: 1 bit
- Address Offset: 0x84
- Read/Write Access: Read/Write

The register is used to generate a NACK for the data part of a transfer when DW_apb_i2c is acting as a slave-receiver. This register only exists when the IC_SLV_DATA_NACK_ONLY parameter is set to 1. When this parameter disabled, this register does not exist and writing to the register's address has no effect.

A write can occur on this register if either of the following conditions are met:

- DW_apb_i2c is disabled (IC_ENABLE[0] = 0)
- Slave part is inactive (IC_STATUS[6] = 0)

Module::MIS	Register::IC2_SLV_DATA	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B784

	_NACK_ONLY				
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**	
Rvd	31:1	-	-	-	
NACK	0	R/W	‘b0	Generate NACK. This NACK generation only occurs when DW_apb_i2c is a slave-receiver. If this register is set to a value of 1, it can only generate a NACK after a data byte is received; hence, the data transfer is aborted and the data received is not pushed to the receive buffer. When the register is set to a value of 0, it generates NACK/ACK, depending on normal criteria. 1 = generate NACK after data byte received 0 = generate NACK/ACK normally	

IC2_DMA_CR

- **Name:** DMA Control Register
- **Size:** 2 bits
- **Address Offset:** 0x88
- **Read/Write Access:** Read/Write

This register is only valid when DW_apb_i2c is configured with a set of DMA Controller interface signals (IC_HAS_DMA = 1). When DW_apb_i2c is not configured for DMA operation, this register does not exist and writing to the register’s address has no effect and reading from this register address will return zero. The register is used to enable the DMA Controller interface operation. There is a separate bit for transmit and receive. This can be programmed regardless of the state of IC_ENABLE.

Module::MIS	Register::IC2_DMA_CR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B788
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**	
Rvd	31:2	-	-	-	
TDMAE	1	R/W	‘b0	**Transmit DMA Enable**. This bit enables/disables the transmit FIFO DMA channel. 0 = Transmit DMA disabled 1 = Transmit DMA enabled	
RDMAE	0	R/W	‘b0	**Receive DMA Enable**. This bit enables/disables the receive FIFO DMA channel. 0 = Receive DMA disabled 1 = Receive DMA enabled	

IC2_DMA_TDLR

- **Name:** DMA Transmit Data Level Register
- **Size:** 2 bits

- **Address Offset:** 0x8c
- **Read/Write Access:** Read/Write

This register is only valid when the DW_apb_i2c is configured with a set of DMA interface signals (IC_HAS_DMA = 1). When DW_apb_i2c is not configured for DMA operation, this register does not exist; writing to its address has no effect; reading from its address returns zero.

Module::MIS	Register::IC2_DMA_TDLR		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B78C
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:3	-	-	-		
DMATDL	2:0	R/W	'h0	**Transmit Data Level**. This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and TDMAE = 1.		

IC2_DMA_RDLR

- **Name:** I2C Receive Data Level Register
- **Size:** 2 bits
- **Address Offset:** 0x90
- **Read/Write Access:** Read/Write

This register is only valid when DW_apb_i2c is configured with a set of DMA interface signals (IC_HAS_DMA = 1). When DW_apb_i2c is not configured for DMA operation, this register does not exist; writing to its address has no effect; reading from its address returns zero.

Module::MIS	Register::IC2_DMA_RDLR		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B790
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:4	-	-	-		
DMARDL	3:0	R/W	'h0	**Receive Data Level**. This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or more than this field value + 1, and RDMAE =1. For instance, when DMARDL is 0, then dma_rx_req is asserted when 1 or more data entries are present in the receive FIFO.		

IC2_SDA_SETUP

- **Name: I2C SDA Setup Register**

- **Size: 8 bits**
- **Address Offset: 0x94**
- **Read/Write Access: Read/Write**

This register controls the amount of time delay (in terms of number of ic_clk clock periods) introduced

in the rising edge of SCL, relative to SDA changing, when DW_apb_i2c services a read request in a

slave-transmitter operation. The relevant I2C requirement is tSU:DAT (note 4) as detailed in the I2C Bus

Specification.

Module::MIS	Register::IC2_SDA_SETUP	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B794
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**	
Rvd	31:8	-	-	-	
SDA_SETUP	7:0	R/W	'h0	SDA Setup. It is recommended that if the required delay is 1000ns, then for an ic_clk frequency of 10 MHz, IC_SDA_SETUP should be programmed to a value of 11.	

IC2_ACK_GENERAL_CALL

- Name: I2C ACK General Call Register
- Size: 1 bit
- Address Offset: 0x98
- Read/Write Access: Read/Write

The register controls whether DW_apb_i2c responds with a ACK or NACK when it receives an I2C

General Call address.

Module::MIS	Register::IC2_ACK_GENERAL_CALL	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B798
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**	
Rvd	31:1	-	-	-	
ACK_GEN_CALL	0	R/W	'h0	ACK General Call. When set to 1, DW_apb_i2c responds with a ACK (by asserting ic_data_oe) when it receives a General Call. Otherwise, DW_apb_i2c responds with a NACK (by negating ic_data_oe).	

IC2_ENABLE_STATUS

- Name: I2C Enable Status Register
- Size: 3 bits
- Address Offset: 0x9C
- Read/Write Access: Read

The register is used to report the DW_apb_i2c hardware status when the IC_ENABLE register is set from 1 to 0; that is, when DW_apb_i2c is disabled.

If IC_ENABLE has been set to 1, bits 2:1 are forced to 0, and bit 0 is forced to 1.

If IC_ENABLE has been set to 0, bits 2:1 is only be valid as soon as bit 0 is read as '0'.

Module::MIS	Register::IC2_ENABLE_STATUS	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B79C

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:3	-	-	-
SLV_RX_DATA_LOST	2	R/W	'h0	Slave Received Data Lost. This bit indicates if a Slave-Receiver operation has been aborted with at least one data byte received from an I2C transfer due to the setting of IC_ENABLE from 1 to 0.When read as 1, DW_apb_i2c is deemed to have been actively engaged in an aborted I2C transfer (with matching address) and the data phase of the I2C transfer has been entered, even though a data byte has been responded with a NACK. NOTE: If the remote I2C master terminates the transfer with a STOP condition before the DW_apb_i2c has a chance to NACK a transfer, and IC_ENABLE has been set to 0, then this bit is also set to 1.When read as 0, DW_apb_i2c is deemed to have been disabled without being actively involved in the data phase of a Slave-Receiver transfer. NOTE: The CPU can safely read this bit when IC_EN (bit 0) is read as 0.
SLV_DISABLED_WHILE_BUSY	1	R/W	'h0	Slave Disabled While Busy (Transmit, Receive). This bit indicates if a potential or active Slave operation has been aborted due to the setting of the IC_ENABLE register from 1 to 0. This bit is set when the CPU writes a 0 to the IC_ENABLE register while: (a) DW_apb_i2c is receiving the address byte of the Slave-Transmitter operation from a remote master; OR, (b) address and data bytes of the Slave-Receiver operation from a remote master. When read as 1, DW_apb_i2c is deemed to have forced a NACK during any part of an I2C transfer, irrespective of whether the I2C address matches the slave address set in

				DW_apb_i2c (IC_SAR register) OR if the transfer is completed before IC_ENABLE is set to 0 but has not taken effect. NOTE: If the remote I2C master terminates the transfer with a STOP condition before the DW_apb_i2c has a chance to NACK a transfer, and IC_ENABLE has been set to 0, then this bit will also be set to 1. When read as 0, DW_apb_i2c is deemed to have been disabled when there is master activity, or when the I2C bus is idle. NOTE: The CPU can safely read this bit when IC_EN (bit 0) is read as 0.
IC_EN	0	R/W	'h0	ic_en Status. This bit always reflects the value driven on the output port ic_en. When read as 1, DW_apb_i2c is deemed to be in an enabled state. When read as 0, DW_apb_i2c is deemed completely inactive. NOTE: The CPU can safely read this bit anytime. When this bit is read as 0, the CPU can safely read SLV_RX_DATA_LOST (bit 2) and SLV_DISABLED_WHILE_BUSY (bit 1).

IC2_COMP_PARAM_1

- **Name:** Component Parameter Register 1
- **Size:** 32 bits
- **Address Offset:** 0xf4
- **Read/Write Access:** Read

Module::MIS	Register::IC2_COMP_PARAM_1	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B7F4

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:24	-	-	-
TX_BUFFER_DEPTH	23:16	R	'h07	The value of this register is derived from the IC_TX_BUFFER_DEPTH coreConsultant parameter. 0x00 = Rvd 0x01 = 2 0x02 = 3 to 0xFF = 256
RX_BUFFER_DEPTH	15:8	R	'h07	The value of this register is derived from the IC_RX_BUFFER_DEPTH coreConsultant parameter. For a description of this parameter, see DesignWare DW_apb_i2c Databook, DesignWare DW_apb_i2c Databook, Table 4 on page 46. 0x00 = Rvd 0x01 = 2

				0x02 = 3 to 0xFF = 256
ADD_ENCODED_PARAMS	7	R	‘b1	The value of this register is derived from the IC_ADD_ENCODED_PARAMS coreConsultant parameter. For a description of this parameter, see DesignWare DW_apb_i2c Databook, Table 4 on page 46. 0 = False 1 = True
HAS_DMA	6	R	‘b0	The value of this register is derived from the IC_HAS_DMA coreConsultant parameter. For a description of this parameter, see DesignWare DW_apb_i2c Databook, Table 4 on page 46. 0 = False 1 = True
INTR_IO	5	R	‘b1	The value of this register is derived from the IC_INTR_IO coreConsultant parameter. For a description of this parameter, see DesignWare DW_apb_i2c Databook, Table 4 on page 46. 0 = Individual 1 = Combined
HC_COUNT_VALUES	4	R	‘b0	The value of this register is derived from the IC_HC_COUNT_VALUES coreConsultant parameter. For a description of this parameter, see DesignWare DW_apb_i2c Databook, Table 4 on page 46. 0 = False 1 = True
MAX_SPEED_MODE	3:2	*R*	’b10	The value of this register is derived from the IC_MAX_SPEED_MODE coreConsultant parameter. For a description of this parameter, see DesignWare DW_apb_i2c Databook, Table 4 on page 46. 0x0 = Rvd 0x1 = Standard 0x2 = Fast 0x3 = High
APB_DATA_WIDTH	1:0	R	’b10	The value of this register is derived from the IC_MAX_SPEED_MODE coreConsultant parameter. For a description of this parameter, see DesignWare DW_apb_i2c Databook, Table 4 on page 46. 0x0 = 8 bits 0x1 = 16 bits 0x2 = 32 bits 0x3 = Rvd

Notice: This is a constant read-only register that contains encoded information about the component's parameter settings. The reset value depends on coreConsultant parameter(s).

IC2_COMP_VERSION

- **Name:** I2C Component Version Register
- **Size:** 32 bits
- **Address Offset:** 0xf8
- **Read/Write Access:** Read

Module::MIS	Register::IC2_COMP_VERSION	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B7F8

Name	Bits	Read /Write	Reset State	Comments
IC_COMP_VERSION	31:0	R	'h3130332a	Lists the releases/versions of the DW_apb_i2c component and the I2C_COMP_VERSION value.

DesignWare AMBA Release	DW_apb_i2c Version	I2C_COMP_VERSION value	Databook Date
2004.06	1.03a	31_30_33_2A	June 21, 2004

IC2_COMP_TYPE

- **Name:** I2C Component Type Register
- **Size:** 32 bits
- **Address Offset:** 0xfc
- **Read/Write Access:** Read

Module::MIS	Register::IC2_COMP_TYPE	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B7FC

Name	Bits	Read /Write	Reset State	Comments
IC_COMP_TYPE	31:0	R	'h44570140	Designware Component Type number = 0x44_57_01_40. This assigned unique hex value is constant and is derived from the two ASCII letters "DW" followed by a 16-bit unsigned number.

3.1.3 Third I2C Register Description

IC3_CON

- **Name:** I2C Control Register 0

- **Size:** 7 bits
- **Address Offset:** 0x00
- **Read/Write Access:** Read/Write

This register can be written only when the DW_apb_i2c is disabled, which corresponds to the IC_ENABLE register being set to 0. Writes at other times have no effect.

Module::MIS	Register::IC3_CON	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B900

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:7	-	-	-
IC_SLAVE_DISABLE	6	R/W	‘b1	This bit controls whether I2C has its slave disabled after reset. The slave can be disabled by programming a ‘1’ into IC_CON[6]. B y default the slave is enabled. 0: slave is enabled 1: slave is disabled
IC_RESTART_EN	5	R/W	‘b1	Determines whether RESTART conditions may be sent when acting as a master. Some older slaves do not support handling RESTART conditions; however, RESTART conditions are used in several DW_apb_i2c operations. 0: disable 1: enable When RESTART is disabled, the master is prohibited from performing the following functions: ● Change direction within a transfer (split) ● Send a START BYTE ● High-speed mode operation ● Combined format transfers in 7-bit addressing modes ● Read operation with a 10-bit address ● Send multiple bytes per transfer By replacing RESTART condition followed by a STOP and a subsequent START condition, split operations are broken down into multiple DW_apb_i2c transfers. If the above operations are performed, it will result in setting bit 6 (TX_ABRT) of the IC_RAW_INTR_STAT register.
IC_10BITADDR_MASTER	4	R/W	‘b1	This bit controls whether the DW_apb_i2c starts its transfers in 10-bit addressing mode when acting as a master. 0: 7-bit addressing

				1: 10-bit addressing
IC_10BITADDR_SLAVE	3	R/W	‘b1	When acting as a slave, this bit controls whether the DW_apb_i2c responds to 7- or 10-bit addresses. 0: 7-bit addressing. The DW_apb_i2c ignores transactions which involve 10-bit addressing; for 7-bit addressing, only the lower 7 bits of the IC_SAR register are compared. 1: 10-bit addressing. The DW_apb_i2c responds to only 10-bit addressing transfers that match the full 10 bits of the IC_SAR register.
SPEED	2:1	R/W	’b10	Controls at which speed the DW_apb_i2c operates: 0: illegal; writing a 0 results in setting SPEED to IC_MAX_SPEED_MODE 1: standard mode (100 kbit/s) 2: fast mode (400 kbit/s) 3: high speed mode (3.4 Mbit/s) If the DW_apb_i2c is configured for fast or standard mode (1 or 2) and a value of 2 or 3 is written, then IC_MAX_SPEED_MODE is stored.
MASTER_MODE	0	R/W	‘b1	This bit controls whether the DW_apb_i2c master is enabled or not. The slave is always enabled. 0: master disabled 1: master enabled

IC3_TAR

- **Name:** I2C Target Address Register
- **Size:** 12 bits
- **Address Offset:** 0x04
- **Read/Write Access:** Read/Write

All bits can be dynamically updated as long as any set of the following conditions are true:

- DW_apb_i2c is NOT enabled (IC_ENABLE is set to 0); or
- DW_apb_i2c is enabled (IC_ENABLE=1); AND

DW_apb_i2c is NOT engaged in any Master (tx, rx) operation (IC_STATUS[5]=0); AND
DW_apb_i2c is enabled to operate in Master mode (IC_CON[0]=1); AND
there are NO entries in the TX FIFO (IC_STATUS[2]=1)

Module::MIS	Register::IC3_TAR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B904

Name	Bits	Read /Write	Reset State	Comments

Rvd	31:13	-	-	-
IC_10BITADDR_MASTER	12	R/W	'b0	This bit controls whether the DW_apb_i2c starts its transfers in 7-or 10-bit addressing mode when acting as a master. 0: 7-bit addressing 1: 10-bit addressing Dependencies: This bit exists in this register only if the I2C_DYNAMIC_TAR_UPDATE configuration parameter is set to "Yes" (1).
SPECIAL	11	R/W	'b0	This bit indicates whether software performs a General Call or START BYTE command. 0: ignore bit 10 GC_OR_START and use IC_TAR normally 1: perform special I2C command as specified in GC_OR_START bit
GC_OR_START	10	R/W	'b0	If bit 11 (SPECIAL) is set to 1, then this bit indicates whether a General Call or START byte command is to be performed by the DW_apb_i2c. 0: General Call Address – after issuing a General Call, only writes may be performed. Attempting to issue a read command results in setting bit 6 (TX_ABRT) of the IC_RAW_INTR_STAT register. The DW_apb_i2c remains in General Call mode until the SPECIAL bit value (bit 11) is cleared. 1: START BYTE
IC_TAR	9:0	R/W	'h055	This is the target address for any master transaction. When transmitting a General Call, these bits are ignored. To generate a START BYTE, the CPU needs to write only once into these bits. If the IC_TAR and IC_SAR are the same, loopback exists but the FIFOs are shared between master and slave, so full loopback is not feasible. Only one direction loopback mode is supported (simplex), not duplex. A master cannot transmit to itself; it can transmit to only a slave.

IC3_SAR

- **Name:** I2C Slave Address Register
- **Size:** 10 bits
- **Address Offset:** 0x08
- **Read/Write Access:** Read/Write

Module::MIS	Register::IC3_SAR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B908

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:10	-	-	-
IC_SAR	9:0	R/W	'h055	The IC_SAR holds the slave address when the I2C is operating as a slave. IC_SAR holds the slave address to which

				the DW_apb_i2c responds. For 7-bit addressing, only IC_SAR[6:0] is used. This register can be written only when the I2C interface is disabled, which corresponds to the *IC_ENABLE* register being set to 0. Writes at other times have no effect.

IC3_HS_MADDR

- **Name:** I2C HS Master Mode Code Address Register
- **Size:** 3 bits
- **Address Offset:** 0x0c
- **Read/Write Access:** Read/Write

Module::MIS	Register::IC3_HS_MADDR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B90C

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:3	-	-	-
IC_HS_MAR	2:0	R/W	'b0	This bit field holds the value of the I2C HS mode master code. HS-mode master codes are reserved 8-bit codes (00001xxx) that are not used for slave addressing or other purposes. Each master has its unique master code; up to eight high-speed mode masters can be present on the same I2C bus system. Valid values are from 0 to 7. This register goes away and becomes read-only returning 0's if the IC_MAX_SPEED_MODE configuration parameter is set to either Standard (1) or Fast (2). This register can be written only when the I2C interface is disabled, which corresponds to the IC_ENABLE register being set to 0. Writes at other times have no effect.

IC3_DATA_CMD

- **Name:** I2C Rx/Tx Data Buffer and Command Register; this is the register the CPU writes to when filling the TX FIFO and the CPU reads from when retrieving bytes from RX FIFO
- **Size:** 11 bits (writes)
- **Address Offset:** 0x10
- **Read/Write Access:** Read/Write

Module::MIS	Register::IC3_DATA_CMD	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B910

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:11	-	-	-
RESTART	10	W	'b0	This bit controls whether a RESTART is issued before the byte is sent or received. Thie bit is available only if IC_EMPTYFIFO_HOLD_MASTER_EN is configured to 1. 1 – If IC_RESTART_EN is 1, a RESTART is issued before the data is sent/received (according to the value of CMD), regardless of whether of not the transfer direction is changing from the

				previous command; if IC_RESTART_EN is 0, a STOP followed by a START is issued instead. 0 – If IC_RESTART_EN is 1, a RESTART is issued only if the transfer direction is changing from the previous command; if IC_RESTART_EN is 0, a STOP followed by a START is issued instead.
STOP	9	W	‘b0	This bit controls whether a STOP is issued after the byte is sent or received. This bit is available only if IC_EMPTYFIFO_HOLD_MASTER_EN is configured to 1. 1 – STOP is issued after this byte, regardless of whether of not the Tx FIFO is empty. If the Tx FIFO is not empty, the master immediately tries to start a new transfer by issuing a START and arbitrating for the bus. 0 – STOP is not issued after this byte, regardless of whether or not the Tx FIFO is empty. If the Tx FIFO is not empty, the master continues the current transfer by sending/receiving data bytes according to the value of the CMD bit. If the Tx FIFO is empty, the master holds the SCL line low and stalls the bus until a new command is available in the Tx FIFO.
CMD	8	W	‘b0	This bit controls whether a read or a write is performed. This bit does not control the direction when the DW_apb_i2c acts as a slave. It controls only the direction when it acts as a master. 1 = Read 0 = Write When a command is entered in the TX FIFO, this bit distinguishes the write and read commands. In slave-receiver mode, this bit is a “don’t care” because writes to this register are not required. In slave-transmitter mode, a “0” indicates that CPU data is to be transmitted and as DAT or IC_DATA_CMD[7:0]. When programming this bit, you should remember the following: attempting to perform a read operation after a General Call command has been sent results in a TX_ABRT interrupt (bit 6 of the IC_RAW_INTR_STAT register), unless bit 11 (SPECIAL) in the IC_TAR register has been cleared. If a “1” is written to this bit after receiving a RD_REQ interrupt, then a TX_ABRT interrupt occurs. NOTE: It is possible that while attempting a master I2C read transfer on DW_apb_i2c, a RD_REQ interrupt may have occurred simultaneously due to a remote I2C master addressing DW_apb_i2c. In this type of scenario, DW_apb_i2c ignores the IC_DATA_CMD write, generates a TX_ABRT interrupt, and waits to service the RD_REQ interrupt.
DAT	7:0	R/W	‘b0	This register contains the data to be transmitted or received on the I2C bus. If you are writing to this register and want to perform a read, bits 7:0 (DAT) are ignored by the DW_apb_i2c. However, when you read this register, these bits return the value of data received on the DW_apb_i2c interface.

IC3_SS_SCL_HCNT

- **Name:** Standard Speed I2C Clock SCL High Count Register
- **Size:** 16 bits
- **Address Offset:** 0x14
- **Read/Write Access:** Read/Write

Module::MIS	Register::IC3_SS_SCL_HCNT	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B914

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:16	-	-	-
IC_SS_SCL_HCNT	15:0	R/W	'h007a	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high-period count for standard speed. The table below shows some sample IC_SS_HCNT calculations. This register can be written only when the I2C interface is disabled which corresponds to the *IC_ENABLE* register being set to 0. Writes at other times have no effect. The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set. For designs with APB_DATA_WIDTH = 8 the order of programming is important to ensure the correct operation of the DW_apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed. When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only.

Notice: Read-only if IC_HC_COUNT_VALUES = 1.

I2C Data Rate (kbps)	ic_clk_{freq} (MHz)	SCL High required min (us)	H_CNT (HEX/decimal)	SCL High Time (us)
100	2	4	0008/8	4.00
100	6.6	4	001B/27	4.09
100	10	4	0028/40	4.00
100	75	4	012C/300	4.00
100	100	4	0190/400	4.00
100	125	4	01F4/500	4.00
100	1000	4	0FA0/4000	4.00

IC3_SS_SCL_LCNT

- **Name:** Standard Speed I2C Clock SCL Low Count Register
- **Size:** 16 bits
- **Address Offset:** 0x98
- Read/Write Access: Read/Write

Module::MIS	Register::IC3_SS_SCL_LCNT	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B918

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:16	-	-	-
IC_SS_SCL_LCNT	15:0	R/W	'h008f	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for standard speed. The table below shows some sample IC_SS_LCNT calculations. This register can be written only when the I2C interface is disabled which corresponds to the *IC_ENABLE* register being set to 0. Writes at other times have no effect. The minimum valid value is 8; hardware prevents values less than this being written, and if attempted results in 8 being set. For designs with APB_DATA_WIDTH = 8 the order of programming is important to ensure the correct operation of the DW_apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed. When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only.

Notice : Read-only if IC_HC_COUNT_VALUES = 1.

I2C Data Rate (kbps)	ic_clk_freq (MHz)	SCL Low required min (us)	L_CNT (HEX/Decimal)	SCL Low Time (us)
100	2	4.7	000A/10	5.00
100	6.6	4.7	0020/32	4.85
100	10	4.7	002F/47	4.70
100	75	4.7	0161/353	4.71
100	100	4.7	01D6/470	4.70
100	125	4.7	024C/588	4.70
100	1000	4	125C/4700	4.70

IC3_FS_SCL_HCNT

- **Name:** Fast Speed I2C Clock SCL High Count Register
- **Size:** 16 bits
- **Address Offset:** 0x1c
- **Read/Write Access:** Read/Write

Module::MIS	Register::IC3_FS_SCL_HCNT	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B91C

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:16	-	-	-
IC_FS_SCL_HCNT	15:0	R/W	'h0013	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high-period count for fast speed. It is used in high-speed mode to send the Master Code and START BYTE or General CALL. The table below shows some sample IC_FS_SCL_HCNT calculations. This register goes away and becomes read-only returning 0s if IC_MAX_SPEED_MODE = standard. This register can be written only when the I2C interface is disabled, which corresponds to the *IC_ENABLE* register being set to 0. Writes at other times have no effect. The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set. For designs with APB_DATA_WIDTH == 8 the order of programming is important to ensure the correct operation of the DW_apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed. When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only.

Notice: Read-only if IC_HC_COUNT_VALUES = 1.

I²C Data Rate (kbps)	ic_clk$_{freq}$ (MHz)	SCL High required min (us)	H_CNT (HEX/Decimal)	SCL High Time (us)
400	10	0.6	0006/6	0.60
400	25	0.6	000F/15	0.60
400	50	0.6	001E/	0.60
400	75	0.6	002D/30	0.60
400	100	0.6	003C/60	0.60
400	125	0.6	004B/75	0.60
400	1000	0.6	0258/600	0.60
100 (through IC_MAX_SPEED_MODE = standard)	N/A	N/A	disabled	N/A

IC3_FS_SCL_LCNT

- **Name:** Fast Speed I2C Clock SCL Low Count Register
- **Size:** 16 bits
- **Address Offset:** 0x20
- **Read/Write Access:** Read/Write

Module::MIS	Register::IC3_FS_SCL_LCNT		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B920
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:16	-	-	-		
IC_FS_SCL_LCNT	15:0	R/W	'h0028	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for fast speed. It is used in high-speed mode to send the Master Code and START BYTE or General CALL. The table below shows some sample IC_FS_SCL_LCNT calculations. This register goes away and becomes read-only returning 0s if IC_MAX_SPEED_MODE = standard. This register can be written only when the I2C interface is disabled, which corresponds to the *IC_ENABLE* register being set to 0. Writes at other times have no effect. The minimum valid value is 8; hardware prevents values less than this being written, and if attempted results in 8 being set. For designs with APB_DATA_WIDTH = 8 the order of programming is important to ensure the		

				correct operation of the DW_apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed. If the value is less than 8 then the count value gets changed to 8. When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only.

Notice : Read-only if IC_HC_COUNT_VALUES = 1.

I²C Data Rate (kbps)	ic_clk_{freq} (MHz)	SCL Low required min (us)	L_CNT (HEX/Decimal)	SCL Low Time (us)
400	10	1.3	000D/13	1.30
400	25	1.3	0021/33	1.32
400	50	1.3	0041/65	1.30
400	75	1.3	0062/98	1.31
400	100	1.3	0082/130	1.30
400	125	1.3	00A3/163	1.30
400	1000	1.3	0514/1300	1.30
100 (through IC_MAX_SPEED_MODE = standard)	N/A	N/A	disabled	N/A

IC3_INTR_STAT

- **Name:** I2C Interrupt Status Register
- **Size:** 12 bits
- **Address Offset:** 0x2C
- **Read/Write Access:** Read

Each bit in this register has a corresponding mask bit in the IC_INTR_MASK register. These bits are cleared by reading the matching interrupt clear register. The unmasked raw versions of these bits are available in the IC_RAW_INTR_STAT register.

Module::MIS	Register::IC3_INTR_STAT	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B92C

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:12	-	-	-
R_GEN_CALL	11	R	'b0	See "IC_RAW_INTR_STAT" for a detailed description of these bits.
R_START_DET	10	R	'b0	
R_STOP_DET	9	R	'b0	
R_ACTIVITY	8	R	'b0	
R_RX_DONE	7	R	'b0	
R_TX_ABRT	6	R	'b0	

R_RD_REQ	5	R	'b0	
R_TX_EMPTY	4	R	'b0	
R_TX_OVER	3	R	'b0	
R_RX_FULL	2	R	'b0	
R_RX_OVER	1	R	'b0	
R_RX_UNDER	0	R	'b0	

IC3_INTR_MASK

- **Name:** I2C Interrupt Mask Register
- **Size:** 12 bits
- **Address Offset:** 0x30
- **Read/Write Access:** Read/Write

These bits mask their corresponding interrupt status bits. They are active high; a value of 0 prevents a bit from generating an interrupt.

Module::MIS	Register::IC3_INTR_MASK	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B930

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:12	-	-	-
M_GEN_CALL	11	R/W	'b1	Masks this bit in the IC_INTR_STAT register.
M_START_DET	10	R/W	'b0	
M_STOP_DET	9	R/W	'b0	
M_ACTIVITY	8	R/W	'b0	
M_RX_DONE	7	R/W	'b1	
M_TX_ABRT	6	R/W	'b1	
M_RD_REQ	5	R/W	'b1	
M_TX_EMPTY	4	R/W	'b1	
M_TX_OVER	3	R/W	'b1	
M_RX_FULL	2	R/W	'b1	
M_RX_OVER	1	R/W	'b1	
M_RX_UNDER	0	R/W	'b1	

IC3_RAW_INTR_STAT

- **Name:** I2C Raw Interpol Status Register
- **Size:** 12 bits
- **Address Offset:** 0x34
- **Read/Write Access:** Read/Write

Unlike the IC_INTR_STAT register, these bits are not masked so they always show the true status of the DW_apb_i2c.

Module::MIS	Register::IC3_RAW_INTR_STAT	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B934

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:12	-	-	-
GEN_CALL	11	R	‘b0	Set only when a General Call address is received and it is acknowledged. It stays set until it is cleared either by disabling DW_apb_i2c or when the CPU reads bit 0 of the IC_CLR_GEN_CALL register. DW_apb_i2c stores the received data in the Rx buffer.
START_DET	10	R	‘b0	Indicates whether a START or RESTART condition has occurred on the I2C interface regardless of whether DW_apb_i2c is operating in slave or master mode.
STOP_DET	9	R	‘b0	Indicates whether a STOP condition has occurred on the I2C interface regardless of whether DW_apb_i2c is operating in slave or master mode.
ACTIVITY	8	R	‘b0	This bit captures DW_apb_i2c activity and stays set until it is cleared. There are four ways to clear it: ● Disabling the DW_apb_i2c ● Reading the IC_CLR_ACTIVITY register ● Reading the IC_CLR_INTR register ● System reset Once this bit is set, it stays set unless one of the four methods is used to clear it. Even if the DW_apb_i2c module is idle, this bit remains set until cleared, indicating that there was activity on the bus.
RX_DONE	7	R	‘b0	When the DW_apb_i2c is acting as a slave-transmitter, this bit is set to 1 if the master does not acknowledge a transmitted byte. This occurs on the last byte of the transmission, indicating that the transmission is done.
TX_ABRT	6	R	‘b0	This bit indicates if DW_apb_i2c, as an I2C transmitter, is unable to complete the intended actions on the contents of the transmit FIFO. This situation can occur both as an I2C master or an I2C slave, and is referred to as a “transmit abort”. When this bit is set to 1, the IC_TX_ABRT_SOURCE register indicates the reason why the transmit abort takes places. NOTE: The DW_apb_i2c flushes/resets/empties the TX FIFO whenever this bit is set. The TX FIFO remains in this flushed state until the register IC_CLR_TX_ABRT is read. Once this read is performed, the TX FIFO is then ready to accept more data bytes from the APB interface.
RD_REQ	5	R	‘b0	This bit is set to 1 when DW_apb_i2c is acting as a slave and another I2C master is attempting to read data from DW_apb_i2c. The DW_apb_i2c holds the I2C bus in a wait state (SCL=0) until this interrupt is serviced, which means that the slave has been addressed by a remote master that is asking for data to be transferred. The processor must respond to this interrupt and then write the requested data to the IC_DATA_CMD register. This bit is set to 0 just after the processor reads the IC_CLR_RD_REQ register.

TX_EMPTY	4	R	‘b0	This bit is set to 1 when the transmit buffer is at or below the threshold value set in the IC_TX_TL register. It is automatically cleared by hardware when the buffer level goes above the threshold. When the IC_ENABLE bit 0 is 0, the TX FIFO is flushed and held in reset. There the TX FIFO looks like it has no data within it, so this bit is set to 1, provided there is activity in the master or slave state machines. When there is no longer activity, then with ic_en=0, this bit is set to 0.
TX_OVER	3	R	‘b0	Set during transmit if the transmit buffer is filled to IC_TX_BUFFER_DEPTH and the processor attempts to issue another I2C command by writing to the IC_DATA_CMD register. When the module is disabled, this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.
RX_FULL	2	R	‘b0	Set when the receive buffer reaches or goes above the RX_TL threshold in the IC_RX_TL register. It is automatically cleared by hardware when buffer level goes below the threshold. If the module is disabled (IC_ENABLE[0]=0), the RX FIFO is flushed and held in reset; therefore the RX FIFO is not full. So this bit is cleared once the IC_ENABLE bit 0 is programmed with a 0, regardless of the activity that continues.
RX_OVER	1	R	‘b0	Set if the receive buffer is completely filled to IC_RX_BUFFER_DEPTH and an additional byte is received from an external I2C device. The DW_apb_i2c acknowledges this, but any data bytes received after the FIFO is full are lost. If the module is disabled (IC_ENABLE[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.
RX_UNDER	0	R	‘b0	Set if the processor attempts to read the receive buffer when it is empty by reading from the IC_DATA_CMD register. If the module is disabled (IC_ENABLE[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.

IC3_RX_TL

- **Name:** I2C Receive FIFO Threshold Register
- **Size:** 8bits
- **Address Offset:** 0x38
- **Read/Write Access:** Read/Write

Module::MIS	Register::IC3_RX_TL		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B938
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:8	-	-	-		
RX_TL	7:0	R/W	‘h00	Receive FIFO Threshold Level Controls the level of entries (or above) that triggers the		

				RX_FULL interrupt. The valid range is 0-255, with the additional restriction that hardware does not allow this value to be set to a value larger than the depth of the buffer. If an attempt is made to do that, the actual value set will be the maximum depth of the buffer. A value of 0 sets the threshold for 1 entry, and a value of 255 sets the threshold for 256 entries.**Depth of the rx_buffer is 128.**

IC3_TX_TL

- **Name:** I2C Transmit FIFO Threshold Register
- **Size:** 8 bits
- **Address Offset:** 0x3c
- Read/Write Access: Read/Write

Module::MIS	Register::IC3_TX_TL	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B93C

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:8	-	-	-
TX_TL	7:0	R/W	'h00	Transmit FIFO Threshold Level Controls the level of entries (or below) that trigger the TX_EMPTY interrupt. The valid range is 0-255, with the additional restriction that it may not be set to value larger than the depth of the buffer. If an attempt is made to do that, the actual value set will be the maximum depth of the buffer. A value of 0 sets the threshold for 0 entries, and a value of 255 sets the threshold for 255 entries.**Depth of the tx_buffer is 128.**

IC3_CLR_INTR

- **Name:** Clear Combined and Individual Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x40
- **Read/Write Access:** Read

Module::MIS	Register::**IC3_CLR_INTR**	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B940

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:1	-	-	-
CLR_INTR	0	R	'b0	Read this register to clear the combined interrupt, all individual interrupts, and the IC_TX_ABRT_SOURCE

				register. This bit does not clear hardware clearable interrupts but software clearable interrupts. Refer to Bit 9 of the IC_TX_ABRT_SOURCE register for an exception to clearing IC_TX_ABRT_SOURCE.

IC3_CLR_RX_UNDER

- **Name:** Clear RX_UNDER Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x44
- **Read/Write Access:** Read

Module::MIS	Register::IC3_CLR_RX_UNDER		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B944
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:1	-	-	-		
CLR_RX_UNDER	0	R	'b0	Read this register to clear the *RX_UNDER* interrupt.		

IC3_CLR_RX_OVER

- **Name:** Clear RX_OVER Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x48
- **Read/Write Access:** Read

Module::MIS	Register::IC3_CLR_RX_OVER		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B948
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:1	-	-	-		
CLR_RX_OVER	0	R	'b0	Read this register to clear the *RX_OVER* interrupt.		

IC3_CLR_TX_OVER

- **Name:** Clear TX_OVER Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x4c
- **Read/Write Access:** Read

Module::MIS	Register::IC3_CLR_TX_OVER		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B94C
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:1	-	-	-		
CLR_TX_OVER	0	R	'b0	Read this register to clear the *TX_OVER* interrupt.		

IC3_CLR_RD_REQ

- **Name:** Clear RD_REQ Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x50
- **Read/Write Access:** Read

Module::MIS	Register::IC3_CLR_RD_REQ	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B950

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:1	-	-	-
CLR_RD_REQ	0	R	'b0	Read this register to clear the *RD_REQ* interrupt.

IC3_CLR_TX_ABRT

- **Name:** Clear TX_ABRT Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x54
- **Read/Write Access:** Read

Module::MIS	Register::IC3_CLR_TX_ABRT	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B954

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:1	-	-	-
CLR_TX_ABRT	0	R	'b0	Read this register to clear the *TX_ABRT* interrupt, and the IC_TX_ABRT_SOURCE register.

IC3_CLR_RX_DONE

- **Name:** Clear RX_DONE Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x58
- **Read/Write Access:** Read

Module::MIS	Register::IC3_CLR_RX_DONE	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B958

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:1	-	-	-
CLR_RX_DONE	0	R	'b0	Read this register to clear the *RX_DONE* interrupt.

IC3_CLR_ACTIVITY

- **Name:** ACTIVITY Status Interrupt Register
- **Size:** 1 bit

- **Address Offset:** 0x5c
- **Read/Write Access:** Read

Module::MIS	Register::IC3_CLR_ACTIVITY	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B95C

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:1	-	-	-
CLR_ACTIVITY	0	R	'b0	Read this register to get status of the *ACTIVITY* interrupt. It is automatically cleared by hardware.

IC3_CLR_STOP_DET

- **Name:** Clear STOP_DET Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x60
- **Read/Write Access:** Read

Module::MIS	Register::IC3_CLR_STOP_DET	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B960

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:1	-	-	-
CLR_STOP_DET	0	R	'b0	Read this register to clear the *STOP_DET* interrupt.

IC3_CLR_START_DET

- **Name:** Clear START_DET Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x64
- **Read/Write Access:** Read

Module::MIS	Register::IC3_CLR_START_DET	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B964

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:1	-	-	-
CLR_START_DET	0	R	'b0	Read this register to clear the *START_DET* interrupt.

IC3_CLR_GEN_CALL

- **Name:** Clear GEN_CALL Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x68
- **Read/Write Access:** Read

Module::MIS	Register::IC3_CLR_GEN_CAL	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B968

	L				
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**	
Rvd	31:1	-	-	-	
CLR_GEN_CALL	0	R	‘b0	Read this register to clear the *GEN_CALL* interrupt.	

IC3_ENABLE

- Name: I2C Enable Register
- **Size:** 1 bit
- **Address Offset:** 0x6c
- **Read/Write Access:** Read/Write

Module::MIS	Register::IC3_ENABLE		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B96C
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:1	-	-	-		
ENABLE	0	R/W	‘b0	Controls whether the DW_apb_i2c is enabled. 0: Disables DW_apb_i2c (TX and RX FIFOs are held in an erased state) 1: Enables DW_apb_i2c Software can disable DW_apb_i2c while it is active. However, it is important that care be taken to ensure that DW_apb_i2c is disabled properly. When DW_apb_i2c is disabled, the following occurs: • The TX FIFO and RX FIFO get flushed. • Status bits in the IC_INTR_STAT register are still active until DW_apb_i2c goes into IDLE state.If the module is transmitting, it stops as well as deletes the contents of the transmit buffer after the current transfer is complete. If the module is receiving, the DW_apb_i2c stops the current transfer at the end of the current byte and does not acknowledge the transfer.In systems with asynchronous pclk and ic_clk when IC_CLK_TYPE parameter set to asynchronous (1), there is a two ic_clk delay when enabling or disabling the DW_apb_i2c.		

IC3_STATUS

- **Name:** I2C Status Register
- **Size:** 5 bits
- **Address Offset:** 0x70
- **Read/Write Access:** Read

This is a read-only register used to indicate the current transfer status and FIFO status. The status register may be read at any time. None of the bits in this register request an interrupt.
When the I2C is disabled by writing 0 in bit 0 of the IC_ENABLE register:

- Bits 1 and 2 are set to 1
- Bits 3 and 4 are set to 0

When the master or slave state machines goes to idle and ic_en=0:

- Bits 5 and 6 are set to 0

Module::MIS	Register::IC3_STATUS		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B970
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:7	-	-	-		
SLV_ACTIVITY	6	R	'b0	Slave FSM Activity Status. When the Slave Finite State Machine (FSM) is not in the IDLE state, this bit is set. 0: Slave FSM is in IDLE state so the Slave part of DW_apb_i2c is not Active 1: Slave FSM is not in IDLE state so the Slave part of DW_apb_i2c is Active		
MST_ACTIVITY	5	R	'b0	Master FSM Activity Status. When the Master Finite State Machine (FSM) is not in the IDLE state, this bit is set. 0: Master FSM is in IDLE state so the Master part of DW_apb_i2c is not Active 1: Master FSM is not in IDLE state so the Master part of DW_apb_i2c is Active		
RFF	4	R	'b0	Receive FIFO Completely Full. When the receive FIFO is completely full, this bit is set. When the receive FIFO contains one or more empty location, this bit is cleared. 0 – Receive FIFO is not full 1 – Receive FIFO is full		
RFNE	3	R	'b0	Receive FIFO Not Empty. Set when the receive FIFO contains one or more entries and is cleared when the receive FIFO is empty. This bit can be polled by software to completely empty the receive FIFO. 0 – Receive FIFO is empty 1 – Receive FIFO is not empty		
TFE	2	R	'b1	Transmit FIFO Completely Empty. When the transmit FIFO is completely empty, this bit is set. When it contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt. 0 – Transmit FIFO is not empty 1 – Transmit FIFO is empty		
TFNF	1	R	'b1	Transmit FIFO Not Full. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full. 0 – Transmit FIFO is full 1 – Transmit FIFO is not full		
ACTIVITY	0	R	'b0	I2C Activity Status.		

IC3_TXFLR

•**Name:** I2C Transmit FIFO Level Register

- **Size:** 4
- **Address Offset:** 0x74
- **Read/Write Access:** Read

This register contains the number of valid data entries in the transmit FIFO buffer. It is cleared whenever:

- The I2C is disabled
- There is a transmit abort—that is, TX_ABRT bit is set in the IC_RAW_INTR_STAT register
- The slave bulk transmit mode is aborted

The register increments whenever data is placed into the transmit FIFO and decrements when data is taken from the transmit FIFO.

Module::MIS	Register::IC3_TXFLR		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B974
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:4	-	-	-		
TXFLR	3:0	R	'b0	**Transmit FIFO Level**. Contains the number of valid data entries in the transmit FIFO.		

IC3_RXFLR

- **Name:** I2C Receive FIFO Level Register
- **Size:** 4
- **Address Offset:** 0x78
- **Read/Write Access:** Read

This register contains the number of valid data entries in the receive FIFO buffer. It is cleared whenever:

- The I2C is disabled
- Whenever there is a transmit abort caused by any of the events tracked in IC_TX_ABRT_SOURCE

The register increments whenever data is placed into the receive FIFO and decrements when data is taken from the receive FIFO.

Module::MIS	Register::IC3_RXFLR		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B978
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:4	-	-	-		
RXFLR	3:0	R	'h0	**Receive FIFO Level**. Contains the number of valid data entries in the receive FIFO.		

IC3_SDA_HOLD

- **Name:** I2C SDA Hold Time Length Register
- **Size:** 16

- **Address Offset:** 0x7C
- **Read/Write Access:** Read / Write

This register controls the amount of hold time on the SDA signal after a negative edge of SCL in both master and slave mode, in units of ic_clk period. The value programmed must be greater than the minimum hold time in each mode for the value to be implemented – one cycle in master mode, seven cycles in slave mode. Writes to this register succeed only when IC_ENABLE=0.
The programmed SDA hold time cannot exceed at any time the duration of the low part of SCL. Therefore the programmed value cannot be larger then N_SCL_LOW-2, where N_SCL_LOW is the duration of the low part of the SCL period measured in ic_clk cycles.

Module::MIS	Register::IC3_SDA_HOLD		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B97C
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:16	-	-	-		
SDA_HOLD	15:0	R/W	'h0001	Sets the required SDA hold time in units of ic_clk period.		

IC3_TX_ABRT_SOURCE

- **Name:** I2C Transmit Abort Source Register
- **Size:** 16 bits
- **Address Offset:** 0x80
- **Read/Write Access:** Read

This register has 16 bits that indicate the source of the TX_ABRT bit. Except for Bit 9, this register is cleared whenever the IC_CLR_TX_ABRT register or the IC_CLR_INTR register is read. To clear Bit 9, the source of the ABRT_SBYTE_NORSTRT must be fixed first; RESTART must be enabled (IC_CON[5]=1), the SPECIAL bit must be cleared (IC_TAR[11]), or the GC_OR_START bit must be cleared (IC_TAR[10]). Once the source of the ABRT_SBYTE_NORSTRT is fixed, then this bit can be cleared in the same manner as other bits in this register. If the source of the ABRT_SBYTE_NORSTRT is not fixed before attempting to clear this bit, Bit 9 clears for one cycle and is then re-asserted.

Module::MIS	Register::IC3_TX_ABRT_SOURCE		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B980
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:16	-	-	-		
ABRT_SLVRD_INTX	15	R	'b0	1: When the processor side responds to a slave mode request for data to be transmitted to a remote master and user writes a 1 in CMD (bit 8) of IC_DATA_CMD register.		
ABRT_SLV_ARBLOST	14	R	'b0	1: Slave lost the bus while transmitting		

				data to a remote master. IC_TX_ABRT_SOURCE[12] is set at the same time. Note: Even though the slave never "owns" the bus, something could go wrong on the bus. This is a fail safe check. For instance, during a data transmission at the low-to-high transition of SCL, if what is on the data bus is not what is supposed to be transmitted, then DW_apb_i2c no longer own the bus.
ABRT_SLVFLUSH_TXFIFO	13	R	'b0	1: Slave has received a read command and some data exists in the TX FIFO so the slave issues a TX_ABRT interrupt to flush old data in TX FIFO.
ARB_LOST	12	R	'b0	1: Master has lost arbitration, or if IC_TX_ABRT_SOURCE[14] is also set, then the slave transmitter has lost arbitration. Note: I2C can be both master and slave at the same time.
ARB_MASTER_DIS	11	R	'b0	1: User tries to initiate a Master operation with the Master mode disabled.
ABRT_10B_RD_NORSTRT	10	R	'b0	1: The restart is disabled (IC_RESTART_EN bit (IC_CON[5]) = 0) and the master sends a read command in 10-bit addressing mode.
ABRT_SBYTE_NORSTRT	9	R	'b0	To clear Bit 9, the source of the ABRT_SBYTE_NORSTRT must be fixed first; restart must be enabled (IC_CON[5]=1), the SPECIAL bit must be cleared (IC_TAR[11]), or the GC_OR_START bit must be cleared (IC_TAR[10]). Once the source of the ABRT_SBYTE_NORSTRT is fixed, then this bit can be cleared in the same manner as other bits in this register. If the source of the ABRT_SBYTE_NORSTRT is not fixed before attempting to clear this bit, bit 9 clears for one cycle and then gets re-asserted. 1: The restart is disabled (IC_RESTART_EN bit (IC_CON[5]) = 0) and the user is trying to send a START Byte.
ABRT_HS_NORSTRT	8	R	'b0	1: The restart is disabled (IC_RESTART_EN bit (IC_CON[5]) = 0) and the user is trying to use the master to transfer data in High Speed mode.
ABRT_SBYTE_ACKDET	7	R	'b0	1: Master has sent a START Byte and the START Byte was acknowledged (wrong behavior).
ABRT_HS_ACKDET	6	R	'b0	1: Master is in High Speed mode and the High Speed Master code was acknowledged (wrong behavior).
ABRT_GCALL_READ	5	R	'b0	1: DW_apb_i2c in master mode sent a General Call but the user programmed the byte following

				the General Call to be a read from the bus (IC_DATA_CMD[9] is set to 1).
ABRT_GCALL_NOACK	4	R	'b0	1: DW_apb_i2c in master mode sent a General Call and no slave on the bus acknowledged the General Call.
ABRT_TXDATA_NOACK	3	R	'b0	1: This is a master-mode only bit. Master has received an acknowledgement for the address, but when it sent data byte(s) following the address, it did not receive an acknowledge from the remote slave(s).
ABRT_10ADDR2_NOACK	2	R	'b0	1: Master is in 10-bit address mode and the second address byte of the 10-bit address was not acknowledged by any slave.
ABRT_10ADDR1_NOACK	1	R	'b0	1: Master is in 10-bit address mode and the first 10-bit address byte was not acknowledged by any slave.
ABRT_7B_ADDR_NOACK	0	R	'b0	1: Master is in 7-bit addressing mode and the address sent was not acknowledged by any slave.

IC3_SLV_DATA_NACK_ONLY

- Name: Generate Slave Data NACK Register
- Size: 1 bit
- Address Offset: 0x84
- Read/Write Access: Read/Write

The register is used to generate a NACK for the data part of a transfer when DW_apb_i2c is acting as a slave-receiver. This register only exists when the IC_SLV_DATA_NACK_ONLY parameter is set to 1. When this parameter disabled, this register does not exist and writing to the register's address has no effect.

A write can occur on this register if either of the following conditions are met:

- DW_apb_i2c is disabled (IC_ENABLE[0] = 0)
- Slave part is inactive (IC_STATUS[6] = 0)

Module::MIS	Register::IC3_SLV_DATA_NACK_ONLY	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B984

Name	**Bits**	**Read /Write**	**Reset State**	**Comments**
Rvd	31:1	-	-	-
NACK	0	R/W	'b0	Generate NACK. This NACK generation only occurs when DW_apb_i2c is a slave-receiver. If this register is set to a value of 1, it can only generate a NACK after a data byte is received; hence, the data transfer is aborted and the data received is not pushed to the receive buffer. When the register is set to a value of 0, it generates NACK/ACK, depending on normal criteria. 1 = generate NACK after data byte received

				0 = generate NACK/ACK normally

IC3_DMA_CR

- **Name:** DMA Control Register
- **Size:** 2 bits
- **Address Offset:** 0x88
- **Read/Write Access:** Read/Write

This register is only valid when DW_apb_i2c is configured with a set of DMA Controller interface signals (IC_HAS_DMA = 1). When DW_apb_i2c is not configured for DMA operation, this register does not exist and writing to the register's address has no effect and reading from this register address will return zero. The register is used to enable the DMA Controller interface operation. There is a separate bit for transmit and receive. This can be programmed regardless of the state of IC_ENABLE.

Module::MIS	Register::IC3_DMA_CR		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B988
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:2	-	-	-		
TDMAE	1	R/W	'b0	**Transmit DMA Enable**. This bit enables/disables the transmit FIFO DMA channel. 0 = Transmit DMA disabled 1 = Transmit DMA enabled		
RDMAE	0	R/W	'b0	**Receive DMA Enable**. This bit enables/disables the receive FIFO DMA channel. 0 = Receive DMA disabled 1 = Receive DMA enabled		

IC3_DMA_TDLR

- **Name:** DMA Transmit Data Level Register
- **Size:** 2 bits
- **Address Offset:** 0x8c
- **Read/Write Access:** Read/Write

This register is only valid when the DW_apb_i2c is configured with a set of DMA interface signals (IC_HAS_DMA = 1). When DW_apb_i2c is not configured for DMA operation, this register does not exist; writing to its address has no effect; reading from its address returns zero.

Module::MIS	Register::IC3_DMA_TDLR		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B98C
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:3	-	-	-		
DMATDL	2:0	R/W	'h0	**Transmit Data Level**. This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of		

				valid data entries in the transmit FIFO is equal to or below this field value, and TDMAE = 1.

IC3_DMA_RDLR

- **Name:** I2C Receive Data Level Register
- **Size:** 2 bits
- **Address Offset:** 0x90
- **Read/Write Access:** Read/Write

This register is only valid when DW_apb_i2c is configured with a set of DMA interface signals (IC_HAS_DMA = 1). When DW_apb_i2c is not configured for DMA operation, this register does not exist; writing to its address has no effect; reading from its address returns zero.

Module::MIS	Register::IC3_DMA_RDLR		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B990
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:4	-	-	-		
DMARDL	3:0	R/W	'h0	**Receive Data Level**. This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or more than this field value + 1, and RDMAE =1. For instance, when DMARDL is 0, then dma_rx_req is asserted when 1 or more data entries are present in the receive FIFO.		

IC3_SDA_SETUP

- **Name: I2C SDA Setup Register**
- **Size: 8 bits**
- **Address Offset: 0x94**
- **Read/Write Access: Read/Write**

This register controls the amount of time delay (in terms of number of ic_clk clock periods) introduced

in the rising edge of SCL, relative to SDA changing, when DW_apb_i2c services a read request in a

slave-transmitter operation. The relevant I2C requirement is tSU:DAT (note 4) as detailed in the I2C Bus

Specification.

Module::MIS	Register::IC3_SDA_SETUP		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B994
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:8	-	-	-		
SDA_SETUP	7:0	R/W	'h0	SDA Setup. It is recommended that if the required delay is 1000ns, then for an ic_clk frequency of 10 MHz, IC_SDA_SETUP should be programmed to a value of 11.		

IC3_ACK_GENERAL_CALL

- Name: I2C ACK General Call Register
- Size: 1 bit
- Address Offset: 0x98
- Read/Write Access: Read/Write

The register controls whether DW_apb_i2c responds with a ACK or NACK when it receives an I2C General Call address.

Module::MIS	Register::IC3_ACK_GENERAL_CALL		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B998
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:1	-	-	-		
ACK_GEN_CALL	0	R/W	'h0	ACK General Call. When set to 1, DW_apb_i2c responds with a ACK (by asserting ic_data_oe) when it receives a General Call. Otherwise, DW_apb_i2c responds with a NACK (by negating ic_data_oe).		

IC3_ENABLE_STATUS

- Name: I2C Enable Status Register
- Size: 3 bits
- Address Offset: 0x9C
- Read/Write Access: Read

The register is used to report the DW_apb_i2c hardware status when the IC_ENABLE register is set

from 1 to 0; that is, when DW_apb_i2c is disabled.

If IC_ENABLE has been set to 1, bits 2:1 are forced to 0, and bit 0 is forced to 1.

If IC_ENABLE has been set to 0, bits 2:1 is only be valid as soon as bit 0 is read as '0'.

Module::MIS	Register::IC3_ENABLE_STATUS	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B99C

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:3	-	-	-
SLV_RX_DATA_LOST	2	R/W	'h0	Slave Received Data Lost. This bit indicates if a Slave-Receiver operation has been aborted with at least one data byte received from an I2C transfer due to the setting of IC_ENABLE from 1 to 0.When read as 1, DW_apb_i2c is deemed to have been actively engaged in an aborted I2C transfer (with matching address) and the data phase of the I2C transfer has been entered, even though a data byte has been responded with a NACK. NOTE: If the remote I2C master terminates the transfer with a STOP condition before the DW_apb_i2c has a chance to NACK a transfer, and IC_ENABLE has been set to 0, then this bit is also set to 1.When read as 0, DW_apb_i2c is deemed to have been disabled without being actively involved in the data phase of a Slave-Receiver transfer. NOTE: The CPU can safely read this bit when IC_EN (bit 0) is read as 0.
SLV_DISABLED_WHILE_BUSY	1	R/W	'h0	Slave Disabled While Busy (Transmit, Receive). This bit indicates if a potential or active Slave operation has been aborted due to the setting of the IC_ENABLE register from 1 to 0. This bit is set when the CPU writes a 0 to the IC_ENABLE register while: (a) DW_apb_i2c is receiving the address byte of the Slave-Transmitter operation from a remote master; OR, (b) address and data bytes of the Slave-Receiver operation from a remote master. When read as 1, DW_apb_i2c is deemed to have forced a NACK during any part of an I2C transfer, irrespective of whether the I2C address matches the slave address set in DW_apb_i2c (IC_SAR register) OR if the transfer is completed before IC_ENABLE is set to 0 but has not taken effect. NOTE: If the remote I2C master terminates the transfer with a STOP condition before the DW_apb_i2c has a chance to NACK a transfer, and IC_ENABLE has been set to 0, then this bit will also be set to 1. When read as 0, DW_apb_i2c is deemed to have been disabled when there is master activity, or when the I2C bus is idle. NOTE: The CPU can safely read this bit when IC_EN (bit 0) is read as 0.
IC_EN	0	R/W	'h0	ic_en Status. This bit always reflects the value driven on the output port ic_en. When read as 1, DW_apb_i2c is deemed to be in an enabled state. When read as 0, DW_apb_i2c is deemed completely

				inactive. NOTE: The CPU can safely read this bit anytime. When this bit is read as 0, the CPU can safely read SLV_RX_DATA_LOST (bit 2) and SLV_DISABLED_WHILE_BUSY (bit 1).

IC3_COMP_PARAM_1

- **Name:** Component Parameter Register 1
- **Size:** 32 bits
- **Address Offset:** 0xf4
- **Read/Write Access:** Read

Module::MIS	Register::IC3_COMP_PARAM_1	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B9F4

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:24	-	-	-
TX_BUFFER_DEPTH	23:16	R	'h07	The value of this register is derived from the IC_TX_BUFFER_DEPTH coreConsultant parameter. 0x00 = Rvd 0x01 = 2 0x02 = 3 to 0xFF = 256
RX_BUFFER_DEPTH	15:8	R	'h07	The value of this register is derived from the IC_RX_BUFFER_DEPTH coreConsultant parameter. For a description of this parameter, see DesignWare DW_apb_i2c Databook, DesignWare DW_apb_i2c Databook, Table 4 on page 46. 0x00 = Rvd 0x01 = 2 0x02 = 3 to 0xFF = 256
ADD_ENCODED_PARAMS	7	R	'b1	The value of this register is derived from the IC_ADD_ENCODED_PARAMS coreConsultant parameter. For a description of this parameter, see DesignWare DW_apb_i2c Databook, Table 4 on page 46. 0 = False 1 = True
HAS_DMA	6	R	'b0	The value of this register is derived from the IC_HAS_DMA coreConsultant parameter. For a description of this parameter, see DesignWare DW_apb_i2c Databook, Table 4 on page 46. 0 = False

				1 = True
INTR_IO	5	R	‘b1	The value of this register is derived from the IC_INTR_IO coreConsultant parameter. For a description of this parameter, see DesignWare DW_apb_i2c Databook, Table 4 on page 46. 0 = Individual 1 = Combined
HC_COUNT_VALUES	4	R	‘b0	The value of this register is derived from the IC_HC_COUNT_VALUES coreConsultant parameter. For a description of this parameter, see DesignWare DW_apb_i2c Databook, Table 4 on page 46. 0 = False 1 = True
MAX_SPEED_MODE	3:2	*R*	’b10	The value of this register is derived from the IC_MAX_SPEED_MODE coreConsultant parameter. For a description of this parameter, see DesignWare DW_apb_i2c Databook, Table 4 on page 46. 0x0 = Rvd 0x1 = Standard 0x2 = Fast 0x3 = High
APB_DATA_WIDTH	1:0	R	’b10	The value of this register is derived from the IC_MAX_SPEED_MODE coreConsultant parameter. For a description of this parameter, see DesignWare DW_apb_i2c Databook, Table 4 on page 46. 0x0 = 8 bits 0x1 = 16 bits 0x2 = 32 bits 0x3 = Rvd

Notice: This is a constant read-only register that contains encoded information about the component's parameter settings. The reset value depends on coreConsultant parameter(s).

IC3_COMP_VERSION

- **Name:** I2C Component Version Register
- **Size:** 32 bits
- **Address Offset:** 0xf8
- **Read/Write Access:** Read

Module::MIS	Register::IC3_COMP_VERSION	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B9F8

Name	Bits	Read /Write	Reset State	Comments
IC_COMP_VERSION	31:0	R	‘h3130332a	Lists the releases/versions of the DW_apb_i2c component and the I2C_COMP_VERSION value.

DesignWare AMBA Release	DW_apb_i2c Version	I2C_COMP_VERSION value	Databook Date
2004.06	1.03a	31_30_33_2A	June 21, 2004

IC3_COMP_TYPE

- **Name:** I2C Component Type Register
- **Size:** 32 bits
- **Address Offset:** 0xfc
- **Read/Write Access:** Read

Module::MIS	Register::IC3_COMP_TYPE	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B9FC

Name	Bits	Read /Write	Reset State	Comments
IC_COMP_TYPE	31:0	R	'h44570140	Designware Component Type number = 0x44_57_01_40. This assigned unique hex value is constant and is derived from the two ASCII letters "DW" followed by a 16-bit unsigned number.

3.1.4 Fourth I2C Register Description

IC4_CON

- **Name:** I2C Control Register 0
- **Size:** 7 bits
- **Address Offset:** 0x00
- **Read/Write Access:** Read/Write

This register can be written only when the DW_apb_i2c is disabled, which corresponds to the IC_ENABLE register being set to 0. Writes at other times have no effect.

Module::MIS	Register::IC4_CON	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA00

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:7	-	-	-
IC_SLAVE_DISABLE	6	R/W	'b1	This bit controls whether I2C has its slave disabled after reset. The slave can be disabled by programming a '1' into IC_CON[6]. B y default the slave is

				enabled. 0: slave is enabled 1: slave is disabled
IC_RESTART_EN	5	R/W	‘b1	Determines whether RESTART conditions may be sent when acting as a master. Some older slaves do not support handling RESTART conditions; however, RESTART conditions are used in several DW_apb_i2c operations. 0: disable 1: enable When RESTART is disabled, the master is prohibited from performing the following functions: ● Change direction within a transfer (split) ● Send a START BYTE ● High-speed mode operation ● Combined format transfers in 7-bit addressing modes ● Read operation with a 10-bit address ● Send multiple bytes per transfer By replacing RESTART condition followed by a STOP and a subsequent START condition, split operations are broken down into multiple DW_apb_i2c transfers. If the above operations are performed, it will result in setting bit 6 (TX_ABRT) of the IC_RAW_INTR_STAT register.
IC_10BITADDR_MASTER	4	R/W	‘b1	This bit controls whether the DW_apb_i2c starts its transfers in 10-bit addressing mode when acting as a master. 0: 7-bit addressing 1: 10-bit addressing
IC_10BITADDR_SLAVE	3	R/W	‘b1	When acting as a slave, this bit controls whether the DW_apb_i2c responds to 7- or 10-bit addresses. 0: 7-bit addressing. The DW_apb_i2c ignores transactions which involve 10-bit addressing; for 7-bit addressing, only the lower 7 bits of the IC_SAR register are compared. 1: 10-bit addressing. The DW_apb_i2c responds to only 10-bit addressing transfers that match the full 10 bits of the IC_SAR

				register.
SPEED	2:1	R/W	'b10	Controls at which speed the DW_apb_i2c operates: 0: illegal; writing a 0 results in setting SPEED to IC_MAX_SPEED_MODE 1: standard mode (100 kbit/s) 2: fast mode (400 kbit/s) 3: high speed mode (3.4 Mbit/s) If the DW_apb_i2c is configured for fast or standard mode (1 or 2) and a value of 2 or 3 is written, then IC_MAX_SPEED_MODE is stored.
MASTER_MODE	0	R/W	'b1	This bit controls whether the DW_apb_i2c master is enabled or not. The slave is always enabled. 0: master disabled 1: master enabled

IC4_TAR

- **Name:** I2C Target Address Register
- **Size:** 12 bits
- **Address Offset:** 0x04
- **Read/Write Access:** Read/Write

All bits can be dynamically updated as long as any set of the following conditions are true:

- DW_apb_i2c is NOT enabled (IC_ENABLE is set to 0); or
- DW_apb_i2c is enabled (IC_ENABLE=1); AND

DW_apb_i2c is NOT engaged in any Master (tx, rx) operation (IC_STATUS[5]=0); AND
DW_apb_i2c is enabled to operate in Master mode (IC_CON[0]=1); AND
there are NO entries in the TX FIFO (IC_STATUS[2]=1)

Module::MIS	Register::IC4_TAR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA04

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:13	-	-	-
IC_10BITADDR_MASTER	12	R/W	'b0	This bit controls whether the DW_apb_i2c starts its transfers in 7-or 10-bit addressing mode when acting as a master. 0: 7-bit addressing 1: 10-bit addressing Dependencies: This bit exists in this register only if the I2C_DYNAMIC_TAR_UPDATE configuration parameter is set to "Yes" (1).
SPECIAL	11	R/W	'b0	This bit indicates whether software performs a General Call or START BYTE command. 0: ignore bit 10 GC_OR_START and use IC_TAR normally 1: perform special I2C command as specified in

				GC_OR_START bit
GC_OR_START	10	R/W	'b0	If bit 11 (SPECIAL) is set to 1, then this bit indicates whether a General Call or START byte command is to be performed by the DW_apb_i2c. 0: General Call Address – after issuing a General Call, only writes may be performed. Attempting to issue a read command results in setting bit 6 (TX_ABRT) of the IC_RAW_INTR_STAT register. The DW_apb_i2c remains in General Call mode until the SPECIAL bit value (bit 11) is cleared. 1: START BYTE
IC_TAR	9:0	R/W	'h055	This is the target address for any master transaction. When transmitting a General Call, these bits are ignored. To generate a START BYTE, the CPU needs to write only once into these bits. If the IC_TAR and IC_SAR are the same, loopback exists but the FIFOs are shared between master and slave, so full loopback is not feasible. Only one direction loopback mode is supported (simplex), not duplex. A master cannot transmit to itself; it can transmit to only a slave.

IC4_SAR

- **Name:** I2C Slave Address Register
- **Size:** 10 bits
- **Address Offset:** 0x08
- **Read/Write Access:** Read/Write

Module::MIS	Register::IC4_SAR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA08

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:10	-	-	-
IC_SAR	9:0	R/W	'h055	The IC_SAR holds the slave address when the I2C is operating as a slave. IC_SAR holds the slave address to which the DW_apb_i2c responds. For 7-bit addressing, only IC_SAR[6:0] is used. This register can be written only when the I2C interface is disabled, which corresponds to the *IC_ENABLE* register being set to 0. Writes at other times have no effect.

IC4_HS_MADDR

- **Name:** I2C HS Master Mode Code Address Register
- **Size:** 3 bits
- **Address Offset:** 0x0c
- **Read/Write Access:** Read/Write

Module::MIS	Register::IC4_HS_MADDR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA0C

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:3	-	-	-
IC_HS_MAR	2:0	R/W	‘b0	This bit field holds the value of the I2C HS mode master code. HS-mode master codes are reserved 8-bit codes (00001xxx) that are not used for slave addressing or other purposes. Each master has its unique master code; up to eight high-speed mode masters can be present on the same I2C bus system. Valid values are from 0 to 7. This register goes away and becomes read-only returning 0’s if the IC_MAX_SPEED_MODE configuration parameter is set to either Standard (1) or Fast (2). This register can be written only when the I2C interface is disabled, which corresponds to the IC_ENABLE register being set to 0. Writes at other times have no effect.

IC4_DATA_CMD

- **Name:** I2C Rx/Tx Data Buffer and Command Register; this is the register the CPU writes to when filling the TX FIFO and the CPU reads from when retrieving bytes from RX FIFO
- **Size:** 11 bits (writes)
- **Address Offset:** 0x10
- **Read/Write Access:** Read/Write

Module::MIS	Register::IC4_DATA_CMD	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA10

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:11	-	-	-
RESTART	10	W	‘b0	This bit controls whether a RESTART is issued before the byte is sent or received. Thie bit is available only if IC_EMPTYFIFO_HOLD_MASTER_EN is configured to 1. 1 – If IC_RESTART_EN is 1, a RESTART is issued before the data is sent/received (according to the value of CMD), regardless of whether of not the transfer direction is changing from the previous command; if IC_RESTART_EN is 0, a STOP followed by a START is issued instead. 0 – If IC_RESTART_EN is 1, a RESTART is issued only if the transfer direction is changing from the previous command; if IC_RESTART_EN is 0, a STOP followed by a START is issued instead.
STOP	9	W	‘b0	This bit controls whether a STOP is issued after the byte is sent or received. This bit is available only if IC_EMPTYFIFO_HOLD_MASTER_EN is configured to 1. 1 – STOP is issued after this byte, regardless of whether of not the Tx FIFO is empty. If the Tx FIFO is not empty, the master immediately tries to start a new transfer by issuing a START and arbitrating for the bus. 0 – STOP is not issued after this byte, regardless of whether or not the Tx FIFO is empty. If the Tx FIFO is not empty, the master continues the current transfer by sending/receiving data bytes

				according to the value of the CMD bit. If the Tx FIFO is empty, the master holds the SCL line low and stalls the bus until a new command is available in the Tx FIFO.
CMD	8	W	'b0	This bit controls whether a read or a write is performed. This bit does not control the direction when the DW_apb_i2c acts as a slave. It controls only the direction when it acts as a master. 1 = Read 0 = Write When a command is entered in the TX FIFO, this bit distinguishes the write and read commands. In slave-receiver mode, this bit is a "don't care" because writes to this register are not required. In slave-transmitter mode, a "0" indicates that CPU data is to be transmitted and as DAT or IC_DATA_CMD[7:0]. When programming this bit, you should remember the following: attempting to perform a read operation after a General Call command has been sent results in a TX_ABRT interrupt (bit 6 of the IC_RAW_INTR_STAT register), unless bit 11 (SPECIAL) in the IC_TAR register has been cleared. If a "1" is written to this bit after receiving a RD_REQ interrupt, then a TX_ABRT interrupt occurs. NOTE: It is possible that while attempting a master I2C read transfer on DW_apb_i2c, a RD_REQ interrupt may have occurred simultaneously due to a remote I2C master addressing DW_apb_i2c. In this type of scenario, DW_apb_i2c ignores the IC_DATA_CMD write, generates a TX_ABRT interrupt, and waits to service the RD_REQ interrupt.
DAT	7:0	R/W	'b0	This register contains the data to be transmitted or received on the I2C bus. If you are writing to this register and want to perform a read, bits 7:0 (DAT) are ignored by the DW_apb_i2c. However, when you read this register, these bits return the value of data received on the DW_apb_i2c interface.

IC4_SS_SCL_HCNT

- **Name:** Standard Speed I2C Clock SCL High Count Register
- **Size:** 16 bits
- **Address Offset:** 0x14
- **Read/Write Access:** Read/Write

Module::MIS	Register::IC4_SS_SCL_HCNT	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA14

Name	Bits	Read	Reset	Comments

		/Write	State	
Rvd	31:16	-	-	-
IC_SS_SCL_HCNT	15:0	R/W	'h007a	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high-period count for standard speed. The table below shows some sample IC_SS_HCNT calculations. This register can be written only when the I2C interface is disabled which corresponds to the *IC_ENABLE* register being set to 0. Writes at other times have no effect. The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set. For designs with APB_DATA_WIDTH = 8 the order of programming is important to ensure the correct operation of the DW_apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed. When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only.

Notice: Read-only if IC_HC_COUNT_VALUES = 1.

I^2C Data Rate (kbps)	ic_clk_{freq} (MHz)	SCL High required min (us)	H_CNT (HEX/decimal)	SCL High Time (us)
100	2	4	0008/8	4.00
100	6.6	4	001B/27	4.09
100	10	4	0028/40	4.00
100	75	4	012C/300	4.00
100	100	4	0190/400	4.00
100	125	4	01F4/500	4.00
100	1000	4	0FA0/4000	4.00

IC4_SS_SCL_LCNT

- **Name:** Standard Speed I2C Clock SCL Low Count Register
- **Size:** 16 bits
- **Address Offset:** 0x98
- Read/Write Access: Read/Write

Module::MIS	Register::IC4_SS_SCL_LCNT	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA18

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:1	-	-	-

	6			
IC_SS_SCL_LCNT	15:0	R/W	'h008f	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for standard speed. The table below shows some sample IC_SS_LCNT calculations. This register can be written only when the I2C interface is disabled which corresponds to the *IC_ENABLE* register being set to 0. Writes at other times have no effect. The minimum valid value is 8; hardware prevents values less than this being written, and if attempted results in 8 being set. For designs with APB_DATA_WIDTH = 8 the order of programming is important to ensure the correct operation of the DW_apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed. When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only.

Notice : Read-only if IC_HC_COUNT_VALUES = 1.

I²C Data Rate (kbps)	ic_clk_{freq} (MHz)	SCL Low required min (us)	L_CNT (HEX/Decimal)	SCL Low Time (us)
100	2	4.7	000A/10	5.00
100	6.6	4.7	0020/32	4.85
100	10	4.7	002F/47	4.70
100	75	4.7	0161/353	4.71
100	100	4.7	01D6/470	4.70
100	125	4.7	024C/588	4.70
100	1000	4	125C/4700	4.70

IC4_FS_SCL_HCNT

- **Name:** Fast Speed I2C Clock SCL High Count Register
- **Size:** 16 bits
- **Address Offset:** 0x1c
- **Read/Write Access:** Read/Write

Module::MIS	Register::IC4_FS_SCL_HCNT	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA1C

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:16	-	-	-
IC_FS_SCL_HCNT	15:0	R/W	'h0013	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register

				sets the SCL clock high-period count for fast speed. It is used in high-speed mode to send the Master Code and START BYTE or General CALL. The table below shows some sample IC_FS_SCL_HCNT calculations. This register goes away and becomes read-only returning 0s if IC_MAX_SPEED_MODE = standard. This register can be written only when the I2C interface is disabled, which corresponds to the *IC_ENABLE* register being set to 0. Writes at other times have no effect. The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set. For designs with APB_DATA_WIDTH == 8 the order of programming is important to ensure the correct operation of the DW_apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed. When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only.

Notice: Read-only if IC_HC_COUNT_VALUES = 1.

I2C Data Rate (kbps)	ic_clk$_{freq}$ (MHz)	SCL High required min (us)	H_CNT (HEX/Decimal)	SCL High Time (us)
400	10	0.6	0006/6	0.60
400	25	0.6	000F/15	0.60
400	50	0.6	001E/	0.60
400	75	0.6	002D/30	0.60
400	100	0.6	003C/60	0.60
400	125	0.6	004B/75	0.60
400	1000	0.6	0258/600	0.60
100 (through IC_MAX_SPEED_MODE = standard)	N/A	N/A	disabled	N/A

IC4_FS_SCL_LCNT

- **Name:** Fast Speed I2C Clock SCL Low Count Register
- **Size:** 16 bits
- **Address Offset:** 0x20
- **Read/Write Access:** Read/Write

Module::MIS	Register::IC4_FS_SCL_LCNT	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA20

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:16	-	-	-
IC_FS_SCL_LCNT	15:0	R/W	'h0028	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for fast speed. It is used in high-speed mode to send the Master Code and START BYTE or General CALL. The table below shows some sample IC_FS_SCL_LCNT calculations. This register goes away and becomes read-only returning 0s if IC_MAX_SPEED_MODE = standard. This register can be written only when the I2C interface is disabled, which corresponds to the *IC_ENABLE* register being set to 0. Writes at other times have no effect. The minimum valid value is 8; hardware prevents values less than this being written, and if attempted results in 8 being set. For designs with APB_DATA_WIDTH = 8 the order of programming is important to ensure the correct operation of the DW_apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed. If the value is less than 8 then the count value gets changed to 8. When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only.

Notice : Read-only if IC_HC_COUNT_VALUES = 1.

I^2C Data Rate (kbps)	ic_clk_{freq} (MHz)	SCL Low required min (us)	L_CNT (HEX/Decimal)	SCL Low Time (us)
400	10	1.3	000D/13	1.30
400	25	1.3	0021/33	1.32
400	50	1.3	0041/65	1.30
400	75	1.3	0062/98	1.31
400	100	1.3	0082/130	1.30
400	125	1.3	00A3/163	1.30
400	1000	1.3	0514/1300	1.30
100 (through IC_MAX_SPEED_MODE = standard)	N/A	N/A	disabled	N/A

IC4_INTR_STAT

- **Name:** I2C Interrupt Status Register
- **Size:** 12 bits
- **Address Offset:** 0x2C
- **Read/Write Access:** Read

Each bit in this register has a corresponding mask bit in the IC_INTR_MASK register. These bits are cleared by reading the matching interrupt clear register. The unmasked raw versions of these bits are available in the IC_RAW_INTR_STAT register.

Module::MIS	Register::IC4_INTR_STAT		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA2C
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:12	-	-	-		
R_GEN_CALL	11	R	'b0	See "IC_RAW_INTR_STAT" for a detailed description of these bits.		
R_START_DET	10	R	'b0			
R_STOP_DET	9	R	'b0			
R_ACTIVITY	8	R	'b0			
R_RX_DONE	7	R	'b0			
R_TX_ABRT	6	R	'b0			
R_RD_REQ	5	R	'b0			
R_TX_EMPTY	4	R	'b0			
R_TX_OVER	3	R	'b0			
R_RX_FULL	2	R	'b0			
R_RX_OVER	1	R	'b0			
R_RX_UNDER	0	R	'b0			

IC4_INTR_MASK

- **Name:** I2C Interrupt Mask Register
- **Size:** 12 bits
- **Address Offset:** 0x30
- **Read/Write Access:** Read/Write

These bits mask their corresponding interrupt status bits. They are active high; a value of 0 prevents a bit from generating an interrupt.

Module::MIS	Register::IC4_INTR_MASK		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA30
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:12	-	-	-		
M_GEN_CALL	11	R/W	'b1	Masks this bit in the IC_INTR_STAT register.		
M_START_DET	10	R/W	'b0			
M_STOP_DET	9	R/W	'b0			
M_ACTIVITY	8	R/W	'b0			
M_RX_DONE	7	R/W	'b1			
M_TX_ABRT	6	R/W	'b1			

M_RD_REQ	5	R/W	‘b1	
M_TX_EMPTY	4	R/W	‘b1	
M_TX_OVER	3	R/W	‘b1	
M_RX_FULL	2	R/W	‘b1	
M_RX_OVER	1	R/W	‘b1	
M_RX_UNDER	0	R/W	‘b1	

IC4_RAW_INTR_STAT

- **Name:** I2C Raw Interpol Status Register
- **Size:** 12 bits
- **Address Offset:** 0x34
- **Read/Write Access:** Read/Write

Unlike the IC_INTR_STAT register, these bits are not masked so they always show the true status of the DW_apb_i2c.

Module::MIS	Register::IC4_RAW_INTR_STAT	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA34

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:12	-	-	-
GEN_CALL	11	R	‘b0	Set only when a General Call address is received and it is acknowledged. It stays set until it is cleared either by disabling DW_apb_i2c or when the CPU reads bit 0 of the IC_CLR_GEN_CALL register. DW_apb_i2c stores the received data in the Rx buffer.
START_DET	10	R	‘b0	Indicates whether a START or RESTART condition has occurred on the I2C interface regardless of whether DW_apb_i2c is operating in slave or master mode.
STOP_DET	9	R	‘b0	Indicates whether a STOP condition has occurred on the I2C interface regardless of whether DW_apb_i2c is operating in slave or master mode.
ACTIVITY	8	R	‘b0	This bit captures DW_apb_i2c activity and stays set until it is cleared. There are four ways to clear it: ● Disabling the DW_apb_i2c ● Reading the IC_CLR_ACTIVITY register ● Reading the IC_CLR_INTR register ● System reset Once this bit is set, it stays set unless one of the four methods is used to clear it. Even if the DW_apb_i2c module is idle, this bit remains set until cleared, indicating that there was activity on the bus.
RX_DONE	7	R	‘b0	When the DW_apb_i2c is acting as a slave-transmitter, this bit is set to 1 if the master does not acknowledge a transmitted

				byte. This occurs on the last byte of the transmission, indicating that the transmission is done.
TX_ABRT	6	R	‘b0	This bit indicates if DW_apb_i2c, as an I2C transmitter, is unable to complete the intended actions on the contents of the transmit FIFO. This situation can occur both as an I2C master or an I2C slave, and is referred to as a “transmit abort”. When this bit is set to 1, the IC_TX_ABRT_SOURCE register indicates the reason why the transmit abort takes places. NOTE: The DW_apb_i2c flushes/resets/empties the TX FIFO whenever this bit is set. The TX FIFO remains in this flushed state until the register IC_CLR_TX_ABRT is read. Once this read is performed, the TX FIFO is then ready to accept more data bytes from the APB interface.
RD_REQ	5	R	‘b0	This bit is set to 1 when DW_apb_i2c is acting as a slave and another I2C master is attempting to read data from DW_apb_i2c. The DW_apb_i2c holds the I2C bus in a wait state (SCL=0) until this interrupt is serviced, which means that the slave has been addressed by a remote master that is asking for data to be transferred. The processor must respond to this interrupt and then write the requested data to the IC_DATA_CMD register. This bit is set to 0 just after the processor reads the IC_CLR_RD_REQ register.
TX_EMPTY	4	R	‘b0	This bit is set to 1 when the transmit buffer is at or below the threshold value set in the IC_TX_TL register. It is automatically cleared by hardware when the buffer level goes above the threshold. When the IC_ENABLE bit 0 is 0, the TX FIFO is flushed and held in reset. There the TX FIFO looks like it has no data within it, so this bit is set to 1, provided there is activity in the master or slave state machines. When there is no longer activity, then with ic_en=0, this bit is set to 0.
TX_OVER	3	R	‘b0	Set during transmit if the transmit buffer is filled to IC_TX_BUFFER_DEPTH and the processor attempts to issue another I2C command by writing to the IC_DATA_CMD register. When the module is disabled, this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.
RX_FULL	2	R	‘b0	Set when the receive buffer reaches or goes above the RX_TL threshold in the IC_RX_TL register. It is automatically cleared by hardware when buffer level goes below the threshold. If the module is disabled (IC_ENABLE[0]=0), the RX FIFO is flushed and held in reset; therefore the RX FIFO is not full. So this bit is cleared once the IC_ENABLE bit 0 is programmed with a 0, regardless of the activity that continues.
RX_OVER	1	R	‘b0	Set if the receive buffer is completely filled to IC_RX_BUFFER_DEPTH and an additional byte is received from an external I2C device. The DW_apb_i2c acknowledges this, but any data bytes received after the FIFO is full are lost. If the module is disabled (IC_ENABLE[0]=0), this bit keeps its level until the master or slave state machines go into idle,

				and when ic_en goes to 0, this interrupt is cleared.
RX_UNDER	0	R	'b0	Set if the processor attempts to read the receive buffer when it is empty by reading from the IC_DATA_CMD register. If the module is disabled (IC_ENABLE[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.

IC4_RX_TL

- **Name:** I2C Receive FIFO Threshold Register
- **Size:** 8bits
- **Address Offset:** 0x38
- **Read/Write Access:** Read/Write

Module::MIS	Register::IC4_RX_TL		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA38
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:8	-	-	-		
RX_TL	7:0	R/W	'h00	Receive FIFO Threshold Level Controls the level of entries (or above) that triggers the RX_FULL interrupt. The valid range is 0-255, with the additional restriction that hardware does not allow this value to be set to a value larger than the depth of the buffer. If an attempt is made to do that, the actual value set will be the maximum depth of the buffer. A value of 0 sets the threshold for 1 entry, and a value of 255 sets the threshold for 256 entries.**Depth of the rx_buffer is 32.**		

IC4_TX_TL

- **Name:** I2C Transmit FIFO Threshold Register
- **Size:** 8 bits
- **Address Offset:** 0x3c
- Read/Write Access: Read/Write

Module::MIS	Register::IC4_TX_TL		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA3C
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:8	-	-	-		
TX_TL	7:0	R/W	'h00	Transmit FIFO Threshold Level Controls the level of entries (or below) that trigger the TX_EMPTY interrupt. The valid range is 0-255, with the additional restriction that it may not be set to value larger than the depth of the buffer. If an attempt is made to do that, the actual value set will be the maximum		

				depth of the buffer. A value of 0 sets the threshold for 0 entries, and a value of 255 sets the threshold for 255 entries.**Depth of the tx_buffer is 32.**

IC4_CLR_INTR

- **Name:** Clear Combined and Individual Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x40
- **Read/Write Access:** Read

Module::MIS	Register::**IC4_CLR_INTR**	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA40

Name	**Bits**	**Read /Write**	**Reset State**	**Comments**
Rvd	31:1	-	-	-
CLR_INTR	0	R	‘b0	Read this register to clear the combined interrupt, all individual interrupts, and the IC_TX_ABRT_SOURCE register. This bit does not clear hardware clearable interrupts but software clearable interrupts. Refer to Bit 9 of the IC_TX_ABRT_SOURCE register for an exception to clearing IC_TX_ABRT_SOURCE.

IC4_CLR_RX_UNDER

- **Name:** Clear RX_UNDER Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x44
- **Read/Write Access:** Read

Module::MIS	Register::IC4_CLR_RX_UNDER	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA44

Name	**Bits**	**Read /Write**	**Reset State**	**Comments**
Rvd	31:1	-	-	-
CLR_RX_UNDER	0	R	‘b0	Read this register to clear the *RX_UNDER* interrupt.

IC4_CLR_RX_OVER

- **Name:** Clear RX_OVER Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x48
- **Read/Write Access:** Read

Module::MIS	Register::IC4_CLR_RX_OVE	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA48

	R					
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:1	-	-	-		
CLR_RX_OVER	0	R	'b0	Read this register to clear the *RX_OVER* interrupt.		

IC4_CLR_TX_OVER

- **Name:** Clear TX_OVER Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x4c
- **Read/Write Access:** Read

Module::MIS	Register::IC4_CLR_TX_OVER	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA4C
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**	
Rvd	31:1	-	-	-	
CLR_TX_OVER	0	R	'b0	Read this register to clear the *TX_OVER* interrupt.	

IC4_CLR_RD_REQ

- **Name:** Clear RD_REQ Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x50
- **Read/Write Access:** Read

Module::MIS	Register::IC4_CLR_RD_REQ	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA50
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**	
Rvd	31:1	-	-	-	
CLR_RD_REQ	0	R	'b0	Read this register to clear the *RD_REQ* interrupt.	

IC4_CLR_TX_ABRT

- **Name:** Clear TX_ABRT Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x54
- **Read/Write Access:** Read

Module::MIS	Register::IC4_CLR_TX_ABRT	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA54
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**	
Rvd	31:1	-	-	-	
CLR_TX_ABRT	0	R	'b0	Read this register to clear the *TX_ABRT* interrupt, and the	

				IC_TX_ABRT_SOURCE register.

IC4_CLR_RX_DONE

- **Name:** Clear RX_DONE Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x58
- **Read/Write Access:** Read

Module::MIS	Register::IC4_CLR_RX_DONE		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA58
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:1	-	-	-		
CLR_RX_DONE	0	R	'b0	Read this register to clear the *RX_DONE* interrupt.		

IC4_CLR_ACTIVITY

- **Name:** ACTIVITY Status Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x5c
- **Read/Write Access:** Read

Module::MIS	Register::IC4_CLR_ACTIVITY		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA5C
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:1	-	-	-		
CLR_ACTIVITY	0	R	'b0	Read this register to get status of the *ACTIVITY* interrupt. It is automatically cleared by hardware.		

IC4_CLR_STOP_DET

- **Name:** Clear STOP_DET Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x60
- **Read/Write Access:** Read

Module::MIS	Register::IC4_CLR_STOP_DET		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA60
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:1	-	-	-		
CLR_STOP_DET	0	R	'b0	Read this register to clear the *STOP_DET* interrupt.		

IC4_CLR_START_DET

- **Name:** Clear START_DET Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x64
- **Read/Write Access:** Read

Module::MIS	Register::IC4_CLR_START_DET	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA64

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:1	-	-	-
CLR_START_DET	0	R	'b0	Read this register to clear the *START_DET* interrupt.

IC4_CLR_GEN_CALL

- **Name:** Clear GEN_CALL Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x68
- **Read/Write Access:** Read

Module::MIS	Register::IC4_CLR_GEN_CALL	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA68

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:1	-	-	-
CLR_GEN_CALL	0	R	'b0	Read this register to clear the *GEN_CALL* interrupt.

IC4_ENABLE

- Name: I2C Enable Register
- **Size:** 1 bit
- **Address Offset:** 0x6c
- **Read/Write Access:** Read/Write

Module::MIS	Register::IC4_ENABLE	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA6C

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:1	-	-	-
ENABLE	0	R/W	'b0	Controls whether the DW_apb_i2c is enabled. 0: Disables DW_apb_i2c (TX and RX FIFOs are held in an erased state) 1: Enables DW_apb_i2c Software can disable DW_apb_i2c while it is active. However, it is important that care be taken to ensure that DW_apb_i2c is disabled properly. When DW_apb_i2c is disabled, the following occurs: • The TX FIFO and RX FIFO get flushed.

				• Status bits in the IC_INTR_STAT register are still active until DW_apb_i2c goes into IDLE state.If the module is transmitting, it stops as well as deletes the contents of the transmit buffer after the current transfer is complete. If the module is receiving, the DW_apb_i2c stops the current transfer at the end of the current byte and does not acknowledge the transfer.In systems with asynchronous pclk and ic_clk when IC_CLK_TYPE parameter set to asynchronous (1), there is a two ic_clk delay when enabling or disabling the DW_apb_i2c.

IC4_STATUS

- **Name:** I2C Status Register
- **Size:** 5 bits
- **Address Offset:** 0x70
- **Read/Write Access:** Read

This is a read-only register used to indicate the current transfer status and FIFO status. The status register may be read at any time. None of the bits in this register request an interrupt.
When the I2C is disabled by writing 0 in bit 0 of the IC_ENABLE register:

- Bits 1 and 2 are set to 1
- Bits 3 and 4 are set to 0

When the master or slave state machines goes to idle and ic_en=0:

- Bits 5 and 6 are set to 0

Module::MIS	Register::IC4_STATUS		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA70
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:7	-	-	-		
SLV_ACTIVITY	6	R	'b0	Slave FSM Activity Status. When the Slave Finite State Machine (FSM) is not in the IDLE state, this bit is set. 0: Slave FSM is in IDLE state so the Slave part of DW_apb_i2c is not Active 1: Slave FSM is not in IDLE state so the Slave part of DW_apb_i2c is Active		
MST_ACTIVITY	5	R	'b0	Master FSM Activity Status. When the Master Finite State Machine (FSM) is not in the IDLE state, this bit is set. 0: Master FSM is in IDLE state so the Master part of DW_apb_i2c is not Active 1: Master FSM is not in IDLE state so the Master part of DW_apb_i2c is Active		
RFF	4	R	'b0	Receive FIFO Completely Full. When the receive FIFO is completely full, this bit is set. When the receive FIFO contains one or more empty location, this bit is cleared. 0 – Receive FIFO is not full 1 – Receive FIFO is full		
RFNE	3	R	'b0	Receive FIFO Not Empty. Set when the receive FIFO contains one or more entries and is cleared when the receive FIFO is empty. This bit can be polled by software		

				to completely empty the receive FIFO. 0 – Receive FIFO is empty 1 – Receive FIFO is not empty
TFE	2	R	‘b1	Transmit FIFO Completely Empty. When the transmit FIFO is completely empty, this bit is set. When it contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt. 0 – Transmit FIFO is not empty 1 – Transmit FIFO is empty
TFNF	1	R	‘b1	Transmit FIFO Not Full. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full. 0 – Transmit FIFO is full 1 – Transmit FIFO is not full
ACTIVITY	0	R	‘b0	I2C Activity Status.

IC4_TXFLR

- **Name:** I2C Transmit FIFO Level Register
- **Size:** 4
- **Address Offset:** 0x74
- **Read/Write Access:** Read

This register contains the number of valid data entries in the transmit FIFO buffer. It is cleared whenever:

- The I2C is disabled
- There is a transmit abort—that is, TX_ABRT bit is set in the IC_RAW_INTR_STAT register
- The slave bulk transmit mode is aborted

The register increments whenever data is placed into the transmit FIFO and decrements when data is taken from the transmit FIFO.

Module::MIS	Register::IC4_TXFLR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA74

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:4	-	-	-
TXFLR	3:0	R	‘b0	**Transmit FIFO Level**. Contains the number of valid data entries in the transmit FIFO.

IC4_RXFLR

- **Name:** I2C Receive FIFO Level Register
- **Size:** 4
- **Address Offset:** 0x78
- **Read/Write Access:** Read

This register contains the number of valid data entries in the receive FIFO buffer. It is cleared whenever:

- The I2C is disabled
- Whenever there is a transmit abort caused by any of the events tracked in IC_TX_ABRT_SOURCE

The register increments whenever data is placed into the receive FIFO and decrements when data is taken from the receive FIFO.

Module::MIS	Register::IC4_RXFLR		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA78
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:4	-	-	-		
RXFLR	3:0	R	'h0	**Receive FIFO Level**. Contains the number of valid data entries in the receive FIFO.		

IC4_SDA_HOLD

- **Name:** I2C SDA Hold Time Length Register
- **Size:** 16
- **Address Offset:** 0x7C
- **Read/Write Access:** Read / Write

This register controls the amount of hold time on the SDA signal after a negative edge of SCL in both master and slave mode, in units of ic_clk period. The value programmed must be greater than the minimum hold time in each mode for the value to be implemented – one cycle in master mode, seven cycles in slave mode. Writes to this register succeed only when IC_ENABLE=0.

The programmed SDA hold time cannot exceed at any time the duration of the low part of SCL. Therefore the programmed value cannot be larger then N_SCL_LOW-2, where N_SCL_LOW is the duration of the low part of the SCL period measured in ic_clk cycles.

Module::MIS	Register::IC4_SDA_HOLD		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA7C
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:16	-	-	-		
SDA_HOLD	15:0	R/W	'h0001	Sets the required SDA hold time in units of ic_clk period.		

IC4_TX_ABRT_SOURCE

- **Name:** I2C Transmit Abort Source Register
- **Size:** 16 bits
- **Address Offset:** 0x80
- **Read/Write Access:** Read

This register has 16 bits that indicate the source of the TX_ABRT bit. Except for Bit 9, this register is cleared whenever the IC_CLR_TX_ABRT register or the IC_CLR_INTR register is read. To clear Bit 9, the source of the ABRT_SBYTE_NORSTRT must be fixed first; RESTART must be enabled (IC_CON[5]=1), the SPECIAL bit must be cleared (IC_TAR[11]), or the GC_OR_START bit must be cleared (IC_TAR[10]). Once the source of the ABRT_SBYTE_NORSTRT is fixed, then this bit can be cleared in the same manner as other bits in this register. If the source of the ABRT_SBYTE_NORSTRT is not fixed before attempting to clear this bit, Bit 9 clears for one cycle and is then re-asserted.

Module::MIS	Register::IC4_TX_ABRT_SOURCE	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA80

Name	**Bits**	**Read /Write**	**Reset State**	**Comments**
Rvd	31:16	-	-	-
ABRT_SLVRD_INTX	15	R	'b0	1: When the processor side responds to a slave mode request for data to be transmitted to a remote master and user writes a 1 in CMD (bit 8) of IC_DATA_CMD register.
ABRT_SLV_ARBLOST	14	R	'b0	1: Slave lost the bus while transmitting data to a remote master. IC_TX_ABRT_SOURCE[12] is set at the same time. Note: Even though the slave never "owns" the bus, something could go wrong on the bus. This is a fail safe check. For instance, during a data transmission at the low-to-high transition of SCL, if what is on the data bus is not what is supposed to be transmitted, then DW_apb_i2c no longer own the bus.
ABRT_SLVFLUSH_TXFIFO	13	R	'b0	1: Slave has received a read command and some data exists in the TX FIFO so the slave issues a TX_ABRT interrupt to flush old data in TX FIFO.
ARB_LOST	12	R	'b0	1: Master has lost arbitration, or if IC_TX_ABRT_SOURCE[14] is also set, then the slave transmitter has lost arbitration. Note: I2C can be both master and slave at the same time.
ARB_MASTER_DIS	11	R	'b0	1: User tries to initiate a Master operation with the Master mode disabled.
ABRT_10B_RD_NORSTRT	10	R	'b0	1: The restart is disabled (IC_RESTART_EN bit (IC_CON[5]) = 0) and the master sends a read command in 10-bit addressing mode.
ABRT_SBYTE_NORSTRT	9	R	'b0	To clear Bit 9, the source of the ABRT_SBYTE_NORSTRT must be fixed first; restart must be enabled (IC_CON[5]=1), the

				SPECIAL bit must be cleared (IC_TAR[11]), or the GC_OR_START bit must be cleared (IC_TAR[10]). Once the source of the ABRT_SBYTE_NORSTRT is fixed, then this bit can be cleared in the same manner as other bits in this register. If the source of the ABRT_SBYTE_NORSTRT is not fixed before attempting to clear this bit, bit 9 clears for one cycle and then gets re-asserted. 1: The restart is disabled (IC_RESTART_EN bit (IC_CON[5]) = 0) and the user is trying to send a START Byte.
ABRT_HS_NORSTRT	8	R	'b0	1: The restart is disabled (IC_RESTART_EN bit (IC_CON[5]) = 0) and the user is trying to use the master to transfer data in High Speed mode.
ABRT_SBYTE_ACKDET	7	R	'b0	1: Master has sent a START Byte and the START Byte was acknowledged (wrong behavior).
ABRT_HS_ACKDET	6	R	'b0	1: Master is in High Speed mode and the High Speed Master code was acknowledged (wrong behavior).
ABRT_GCALL_READ	5	R	'b0	1: DW_apb_i2c in master mode sent a General Call but the user programmed the byte following the General Call to be a read from the bus (IC_DATA_CMD[9] is set to 1).
ABRT_GCALL_NOACK	4	R	'b0	1: DW_apb_i2c in master mode sent a General Call and no slave on the bus acknowledged the General Call.
ABRT_TXDATA_NOACK	3	R	'b0	1: This is a master-mode only bit. Master has received an acknowledgement for the address, but when it sent data byte(s) following the address, it did not receive an acknowledge from the remote slave(s).
ABRT_10ADDR2_NOACK	2	R	'b0	1: Master is in 10-bit address mode and the second address byte of the 10-bit address was not acknowledged by any slave.
ABRT_10ADDR1_NOACK	1	R	'b0	1: Master is in 10-bit address mode and the first 10-bit address byte was not acknowledged by any slave.
ABRT_7B_ADDR_NOACK	0	R	'b0	1: Master is in 7-bit addressing mode and the address sent was not acknowledged by any slave.

IC4_SLV_DATA_NACK_ONLY

- Name: Generate Slave Data NACK Register
- Size: 1 bit
- Address Offset: 0x84
- Read/Write Access: Read/Write

The register is used to generate a NACK for the data part of a transfer when DW_apb_i2c is acting as a slave-receiver. This register only exists when the IC_SLV_DATA_NACK_ONLY parameter is set to 1. When this parameter disabled, this register does not exist and writing to the register's address has no

effect.
A write can occur on this register if either of the following conditions are met:
- DW_apb_i2c is disabled (IC_ENABLE[0] = 0)
- Slave part is inactive (IC_STATUS[6] = 0)

Module::MIS	Register::IC4_SLV_DATA_NACK_ONLY	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA84

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:1	-	-	-
NACK	0	R/W	'b0	Generate NACK. This NACK generation only occurs when DW_apb_i2c is a slave-receiver. If this register is set to a value of 1, it can only generate a NACK after a data byte is received; hence, the data transfer is aborted and the data received is not pushed to the receive buffer. When the register is set to a value of 0, it generates NACK/ACK, depending on normal criteria. 1 = generate NACK after data byte received 0 = generate NACK/ACK normally

IC4_DMA_CR

- **Name:** DMA Control Register
- **Size:** 2 bits
- **Address Offset:** 0x88
- **Read/Write Access:** Read/Write

This register is only valid when DW_apb_i2c is configured with a set of DMA Controller interface signals (IC_HAS_DMA = 1). When DW_apb_i2c is not configured for DMA operation, this register does not exist and writing to the register's address has no effect and reading from this register address will return zero. The register is used to enable the DMA Controller interface operation. There is a separate bit for transmit and receive. This can be programmed regardless of the state of IC_ENABLE.

Module::MIS	Register::IC4_DMA_CR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA88

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:2	-	-	-
TDMAE	1	R/W	'b0	**Transmit DMA Enable**. This bit enables/disables the transmit FIFO DMA channel. 0 = Transmit DMA disabled 1 = Transmit DMA enabled
RDMAE	0	R/W	'b0	**Receive DMA Enable**. This bit enables/disables the receive FIFO DMA channel. 0 = Receive DMA disabled 1 = Receive DMA enabled

IC4_DMA_TDLR

- **Name:** DMA Transmit Data Level Register
- **Size:** 2 bits
- **Address Offset:** 0x8c
- **Read/Write Access:** Read/Write

This register is only valid when the DW_apb_i2c is configured with a set of DMA interface signals (IC_HAS_DMA = 1). When DW_apb_i2c is not configured for DMA operation, this register does not exist; writing to its address has no effect; reading from its address returns zero.

Module::MIS	Register::IC4_DMA_TDLR		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA8C
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:3	-	-	-		
DMATDL	2:0	R/W	'h0	**Transmit Data Level**. This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and TDMAE = 1.		

IC4_DMA_RDLR

- **Name:** I2C Receive Data Level Register
- **Size:** 2 bits
- **Address Offset:** 0x90
- **Read/Write Access:** Read/Write

This register is only valid when DW_apb_i2c is configured with a set of DMA interface signals (IC_HAS_DMA = 1). When DW_apb_i2c is not configured for DMA operation, this register does not exist; writing to its address has no effect; reading from its address returns zero.

Module::MIS	Register::IC4_DMA_RDLR		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA90
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:4	-	-	-		
DMARDL	3:0	R/W	'h0	**Receive Data Level**. This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or more than this field value + 1, and RDMAE =1. For instance, when DMARDL is 0, then dma_rx_req is asserted when 1 or more data entries are present in the receive FIFO.		

IC4_SDA_SETUP

- **Name: I2C SDA Setup Register**
- **Size: 8 bits**
- **Address Offset: 0x94**
- **Read/Write Access: Read/Write**

This register controls the amount of time delay (in terms of number of ic_clk clock periods) introduced

in the rising edge of SCL, relative to SDA changing, when DW_apb_i2c services a read request in a

slave-transmitter operation. The relevant I2C requirement is tSU:DAT (note 4) as detailed in the I2C Bus

Specification.

Module::MIS	Register::IC4_SDA_SETUP		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA94
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:8	-	-	-		
SDA_SETUP	7:0	R/W	'h0	SDA Setup. It is recommended that if the required delay is 1000ns, then for an ic_clk frequency of 10 MHz, IC_SDA_SETUP should be programmed to a value of 11.		

IC4_ACK_GENERAL_CALL

- Name: I2C ACK General Call Register
- Size: 1 bit
- Address Offset: 0x98
- Read/Write Access: Read/Write

The register controls whether DW_apb_i2c responds with a ACK or NACK when it receives an I2C

General Call address.

Module::MIS	Register::IC4_ACK_GENERAL_CALL		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA98
Name	**Bits**	**Read**	**Reset**	**Comments**		

		/Write	State	
Rvd	31:1	-	-	-
ACK_GEN_CALL	0	R/W	‘h0	ACK General Call. When set to 1, DW_apb_i2c responds with a ACK (by asserting ic_data_oe) when it receives a General Call. Otherwise, DW_apb_i2c responds with a NACK (by negating ic_data_oe).

IC4_ENABLE_STATUS

- Name: I2C Enable Status Register
- Size: 3 bits
- Address Offset: 0x9C
- Read/Write Access: Read

The register is used to report the DW_apb_i2c hardware status when the IC_ENABLE register is set from 1 to 0; that is, when DW_apb_i2c is disabled.

If IC_ENABLE has been set to 1, bits 2:1 are forced to 0, and bit 0 is forced to 1.

If IC_ENABLE has been set to 0, bits 2:1 is only be valid as soon as bit 0 is read as ‘0’.

Module::MIS	Register::IC4_ENABLE_STATUS	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BA9C

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:3	-	-	-
SLV_RX_DATA_LOST	2	R/W	‘h0	Slave Received Data Lost. This bit indicates if a Slave-Receiver operation has been aborted with at least one data byte received from an I2C transfer due to the setting of IC_ENABLE from 1 to 0.When read as 1, DW_apb_i2c is deemed to have been actively engaged in an aborted I2C transfer (with matching address) and the data phase of the I2C transfer has been entered, even though a data byte has been responded with a NACK. NOTE: If the remote I2C master terminates the transfer with a STOP condition before the DW_apb_i2c has a chance to NACK a transfer, and IC_ENABLE has been set to 0, then this bit is also set to 1.When read as 0, DW_apb_i2c is deemed to have been disabled without being actively involved in the data phase of a Slave-Receiver transfer. NOTE: The CPU can safely read this bit when IC_EN (bit 0) is read as 0.
SLV_DISABLED_WHILE_BUSY	1	R/W	‘h0	Slave Disabled While Busy (Transmit, Receive). This bit indicates if a potential or active Slave operation has been aborted due to the setting of the IC_ENABLE register from 1 to 0. This bit is set when the CPU writes a 0 to the

				IC_ENABLE register while: (a) DW_apb_i2c is receiving the address byte of the Slave-Transmitter operation from a remote master; OR, (b) address and data bytes of the Slave-Receiver operation from a remote master. When read as 1, DW_apb_i2c is deemed to have forced a NACK during any part of an I2C transfer, irrespective of whether the I2C address matches the slave address set in DW_apb_i2c (IC_SAR register) OR if the transfer is completed before IC_ENABLE is set to 0 but has not taken effect. NOTE: If the remote I2C master terminates the transfer with a STOP condition before the DW_apb_i2c has a chance to NACK a transfer, and IC_ENABLE has been set to 0, then this bit will also be set to 1. When read as 0, DW_apb_i2c is deemed to have been disabled when there is master activity, or when the I2C bus is idle. NOTE: The CPU can safely read this bit when IC_EN (bit 0) is read as 0.
IC_EN	0	R/W	'h0	ic_en Status. This bit always reflects the value driven on the output port ic_en. When read as 1, DW_apb_i2c is deemed to be in an enabled state. When read as 0, DW_apb_i2c is deemed completely inactive. NOTE: The CPU can safely read this bit anytime. When this bit is read as 0, the CPU can safely read SLV_RX_DATA_LOST (bit 2) and SLV_DISABLED_WHILE_BUSY (bit 1).

IC4_COMP_PARAM_1

- **Name:** Component Parameter Register 1
- **Size:** 32 bits
- **Address Offset:** 0xf4
- **Read/Write Access:** Read

Module::MIS	Register::IC4_COMP_PARAM_1	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BAF4

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:24	-	-	-
TX_BUFFER_DEPTH	23:16	R	'h07	The value of this register is derived from the IC_TX_BUFFER_DEPTH coreConsultant parameter. 0x00 = Rvd 0x01 = 2 0x02 = 3 to 0xFF = 256
RX_BUFFER_DEPTH	15:8	R	'h07	The value of this register is derived from the IC_RX_BUFFER_DEPTH coreConsultant

				parameter. For a description of this parameter, see DesignWare DW_apb_i2c Databook, DesignWare DW_apb_i2c Databook, Table 4 on page 46. 0x00 = Rvd 0x01 = 2 0x02 = 3 to 0xFF = 256
ADD_ENCODED_PARAMS	7	R	‘b1	The value of this register is derived from the IC_ADD_ENCODED_PARAMS coreConsultant parameter. For a description of this parameter, see DesignWare DW_apb_i2c Databook, Table 4 on page 46. 0 = False 1 = True
HAS_DMA	6	R	‘b0	The value of this register is derived from the IC_HAS_DMA coreConsultant parameter. For a description of this parameter, see DesignWare DW_apb_i2c Databook, Table 4 on page 46. 0 = False 1 = True
INTR_IO	5	R	‘b1	The value of this register is derived from the IC_INTR_IO coreConsultant parameter. For a description of this parameter, see DesignWare DW_apb_i2c Databook, Table 4 on page 46. 0 = Individual 1 = Combined
HC_COUNT_VALUES	4	R	‘b0	The value of this register is derived from the IC_HC_COUNT_VALUES coreConsultant parameter. For a description of this parameter, see DesignWare DW_apb_i2c Databook, Table 4 on page 46. 0 = False 1 = True
MAX_SPEED_MODE	3:2	*R*	’b10	The value of this register is derived from the IC_MAX_SPEED_MODE coreConsultant parameter. For a description of this parameter, see DesignWare DW_apb_i2c Databook, Table 4 on page 46. 0x0 = Rvd 0x1 = Standard 0x2 = Fast 0x3 = High
APB_DATA_WIDTH	1:0	R	’b10	The value of this register is derived from the IC_MAX_SPEED_MODE coreConsultant parameter. For a description of this parameter, see DesignWare DW_apb_i2c Databook, Table 4 on page 46. 0x0 = 8 bits

				0x1 = 16 bits 0x2 = 32 bits 0x3 = Rvd

Notice: This is a constant read-only register that contains encoded information about the component's parameter settings. The reset value depends on coreConsultant parameter(s).

IC4_COMP_VERSION

- **Name:** I2C Component Version Register
- **Size:** 32 bits
- **Address Offset:** 0xf8
- **Read/Write Access:** Read

Module::MIS	Register::IC4_COMP_VERSION	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BAF8

Name	Bits	Read /Write	Reset State	Comments
IC_COMP_VERSION	31:0	R	'h3130332a	Lists the releases/versions of the DW_apb_i2c component and the I2C_COMP_VERSION value.

DesignWare AMBA Release	DW_apb_i2c Version	I2C_COMP_VERSION value	Databook Date
2004.06	1.03a	31_30_33_2A	June 21, 2004

IC4_COMP_TYPE

- **Name:** I2C Component Type Register
- **Size:** 32 bits
- **Address Offset:** 0xfc
- **Read/Write Access:** Read

Module::MIS	Register::IC4_COMP_TYPE	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BAFC

Name	Bits	Read /Write	Reset State	Comments
IC_COMP_TYPE	31:0	R	'h44570140	Designware Component Type number = 0x44_57_01_40. This assigned unique hex value is constant and is derived from the two ASCII letters "DW" followed by a 16-bit unsigned number.

IC5_CON

- **Name:** I2C Control Register 0
- **Size:** 7 bits
- **Address Offset:** 0x00
- **Read/Write Access:** Read/Write

This register can be written only when the DW_apb_i2c is disabled, which corresponds to the IC_ENABLE register being set to 0. Writes at other times have no effect.

Module::MIS	Register::IC5_CON	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB00

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:7	-	-	-
IC_SLAVE_DISABLE	6	R/W	'b1	This bit controls whether I2C has its slave disabled after reset. The slave can be disabled by programming a '1' into IC_CON[6]. B y default the slave is enabled. 0: slave is enabled 1: slave is disabled
IC_RESTART_EN	5	R/W	'b1	Determines whether RESTART conditions may be sent when acting as a master. Some older slaves do not support handling RESTART conditions; however, RESTART conditions are used in several DW_apb_i2c operations. 0: disable 1: enable When RESTART is disabled, the master is prohibited from performing the following functions: ● Change direction within a transfer (split) ● Send a START BYTE ● High-speed mode operation ● Combined format transfers in 7-bit addressing modes ● Read operation with a 10-bit address ● Send multiple bytes per transfer By replacing RESTART condition followed by a STOP and a subsequent START condition, split operations are broken down into multiple DW_apb_i2c transfers. If the above operations are performed, it will result in setting bit 6

				(TX_ABRT) of the IC_RAW_INTR_STAT register.
IC_10BITADDR_MASTER	4	R/W	‘b1	This bit controls whether the DW_apb_i2c starts its transfers in 10-bit addressing mode when acting as a master. 0: 7-bit addressing 1: 10-bit addressing
IC_10BITADDR_SLAVE	3	R/W	‘b1	When acting as a slave, this bit controls whether the DW_apb_i2c responds to 7- or 10-bit addresses. 0: 7-bit addressing. The DW_apb_i2c ignores transactions which involve 10-bit addressing; for 7-bit addressing, only the lower 7 bits of the IC_SAR register are compared. 1: 10-bit addressing. The DW_apb_i2c responds to only 10-bit addressing transfers that match the full 10 bits of the IC_SAR register.
SPEED	2:1	R/W	’b10	Controls at which speed the DW_apb_i2c operates: 0: illegal; writing a 0 results in setting SPEED to IC_MAX_SPEED_MODE 1: standard mode (100 kbit/s) 2: fast mode (400 kbit/s) 3: high speed mode (3.4 Mbit/s) If the DW_apb_i2c is configured for fast or standard mode (1 or 2) and a value of 2 or 3 is written, then IC_MAX_SPEED_MODE is stored.
MASTER_MODE	0	R/W	‘b1	This bit controls whether the DW_apb_i2c master is enabled or not. The slave is always enabled. 0: master disabled 1: master enabled

IC5_TAR

- **Name:** I2C Target Address Register
- **Size:** 12 bits
- **Address Offset:** 0x04
- **Read/Write Access:** Read/Write

All bits can be dynamically updated as long as any set of the following conditions are true:

- DW_apb_i2c is NOT enabled (IC_ENABLE is set to 0); or
- DW_apb_i2c is enabled (IC_ENABLE=1); AND

DW_apb_i2c is NOT engaged in any Master (tx, rx) operation (IC_STATUS[5]=0); AND
DW_apb_i2c is enabled to operate in Master mode (IC_CON[0]=1); AND

there are NO entries in the TX FIFO (IC_STATUS[2]=1)

Module::MIS	Register::IC5_TAR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB04

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:13	-	-	-
IC_10BITADDR_MASTER	12	R/W	'b0	This bit controls whether the DW_apb_i2c starts its transfers in 7-or 10-bit addressing mode when acting as a master. 0: 7-bit addressing 1: 10-bit addressing Dependencies: This bit exists in this register only if the I2C_DYNAMIC_TAR_UPDATE configuration parameter is set to "Yes" (1).
SPECIAL	11	R/W	'b0	This bit indicates whether software performs a General Call or START BYTE command. 0: ignore bit 10 GC_OR_START and use IC_TAR normally 1: perform special I2C command as specified in GC_OR_START bit
GC_OR_START	10	R/W	'b0	If bit 11 (SPECIAL) is set to 1, then this bit indicates whether a General Call or START byte command is to be performed by the DW_apb_i2c. 0: General Call Address – after issuing a General Call, only writes may be performed. Attempting to issue a read command results in setting bit 6 (TX_ABRT) of the IC_RAW_INTR_STAT register. The DW_apb_i2c remains in General Call mode until the SPECIAL bit value (bit 11) is cleared. 1: START BYTE
IC_TAR	9:0	R/W	'h055	This is the target address for any master transaction. When transmitting a General Call, these bits are ignored. To generate a START BYTE, the CPU needs to write only once into these bits. If the IC_TAR and IC_SAR are the same, loopback exists but the FIFOs are shared between master and slave, so full loopback is not feasible. Only one direction loopback mode is supported (simplex), not duplex. A master cannot transmit to itself; it can transmit to only a slave.

IC5_SAR

- **Name:** I2C Slave Address Register
- **Size:** 10 bits
- **Address Offset:** 0x08
- **Read/Write Access:** Read/Write

Module::MIS	Register::IC5_SAR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB08

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:10	-	-	-
IC_SAR	9:0	R/W	'h055	The IC_SAR holds the slave address when the I2C is operating as a slave. IC_SAR holds the slave address to which the DW_apb_i2c responds. For 7-bit addressing, only IC_SAR[6:0] is used. This register can be written only when the I2C interface is disabled, which corresponds to the *IC_ENABLE* register being set to 0. Writes at other times have no effect.

IC5_HS_MADDR

- **Name:** I2C HS Master Mode Code Address Register
- **Size:** 3 bits
- **Address Offset:** 0x0c
- **Read/Write Access:** Read/Write

Module::MIS	Register::IC5_HS_MADDR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB0C

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:3	-	-	-
IC_HS_MAR	2:0	R/W	'b0	This bit field holds the value of the I2C HS mode master code. HS-mode master codes are reserved 8-bit codes (00001xxx) that are not used for slave addressing or other purposes. Each master has its unique master code; up to eight high-speed mode masters can be present on the same I2C bus system. Valid values are from 0 to 7. This register goes away and becomes read-only returning 0's if the IC_MAX_SPEED_MODE configuration parameter is set to either Standard (1) or Fast (2). This register can be written only when the I2C interface is disabled, which corresponds to the IC_ENABLE register being set to 0. Writes at other times have no effect.

IC5_DATA_CMD

- **Name:** I2C Rx/Tx Data Buffer and Command Register; this is the register the CPU writes to when filling the TX FIFO and the CPU reads from when retrieving bytes from RX FIFO
- **Size:** 11 bits (writes)
- **Address Offset:** 0x10
- **Read/Write Access:** Read/Write

Module::MIS	Register::IC5_DATA_CMD	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB10

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:11	-	-	-

RESTART	10	W	‘b0	This bit controls whether a RESTART is issued before the byte is sent or received. Thie bit is available only if IC_EMPTYFIFO_HOLD_MASTER_EN is configured to 1. 1 – If IC_RESTART_EN is 1, a RESTART is issued before the data is sent/received (according to the value of CMD), regardless of whether of not the transfer direction is changing from the previous command; if IC_RESTART_EN is 0, a STOP followed by a START is issued instead. 0 – If IC_RESTART_EN is 1, a RESTART is issued only if the transfer direction is changing from the previous command; if IC_RESTART_EN is 0, a STOP followed by a START is issued instead.
STOP	9	W	‘b0	This bit controls whether a STOP is issued after the byte is sent or received. This bit is available only if IC_EMPTYFIFO_HOLD_MASTER_EN is configured to 1. 1 – STOP is issued after this byte, regardless of whether of not the Tx FIFO is empty. If the Tx FIFO is not empty, the master immediately tries to start a new transfer by issuing a START and arbitrating for the bus. 0 – STOP is not issued after this byte, regardless of whether or not the Tx FIFO is empty. If the Tx FIFO is not empty, the master continues the current transfer by sending/receiving data bytes according to the value of the CMD bit. If the Tx FIFO is empty, the master holds the SCL line low and stalls the bus until a new command is available in the Tx FIFO.
CMD	8	W	‘b0	This bit controls whether a read or a write is performed. This bit does not control the direction when the DW_apb_i2c acts as a slave. It controls only the direction when it acts as a master. 1 = Read 0 = Write When a command is entered in the TX FIFO, this bit distinguishes the write and read commands. In slave-receiver mode, this bit is a “don’t care” because writes to this register are not required. In slave-transmitter mode, a “0” indicates that CPU data is to be transmitted and as DAT or IC_DATA_CMD[7:0]. When programming this bit, you should remember the following: attempting to perform a read operation after a General Call command has been sent results in a TX_ABRT interrupt (bit 6 of the IC_RAW_INTR_STAT register), unless bit 11 (SPECIAL) in the IC_TAR register has been cleared. If a “1” is written to this bit after receiving a RD_REQ interrupt, then a TX_ABRT interrupt occurs. NOTE: It is possible that while attempting a master I2C read transfer on DW_apb_i2c, a RD_REQ interrupt may have occurred simultaneously due to a remote I2C master addressing DW_apb_i2c. In this type of scenario, DW_apb_i2c ignores the IC_DATA_CMD write, generates a TX_ABRT interrupt, and waits to service the RD_REQ interrupt.
DAT	7:0	R/W	‘b0	This register contains the data to be transmitted or received

				on the I2C bus. If you are writing to this register and want to perform a read, bits 7:0 (DAT) are ignored by the DW_apb_i2c. However, when you read this register, these bits return the value of data received on the DW_apb_i2c interface.

IC5_SS_SCL_HCNT

- **Name:** Standard Speed I2C Clock SCL High Count Register
- **Size:** 16 bits
- **Address Offset:** 0x14
- **Read/Write Access:** Read/Write

Module::MIS	Register::IC5_SS_SCL_HCNT	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB14

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:16	-	-	-
IC_SS_SCL_HCNT	15:0	R/W	'h007a	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high-period count for standard speed. The table below shows some sample IC_SS_HCNT calculations. This register can be written only when the I2C interface is disabled which corresponds to the *IC_ENABLE* register being set to 0. Writes at other times have no effect. The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set. For designs with APB_DATA_WIDTH = 8 the order of programming is important to ensure the correct operation of the DW_apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed. When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only.

Notice: Read-only if IC_HC_COUNT_VALUES = 1.

I2C Data Rate (kbps)	ic_clk freq (MHz)	SCL High required min (us)	H_CNT (HEX/decimal)	SCL High Time (us)
100	2	4	0008/8	4.00
100	6.6	4	001B/27	4.09
100	10	4	0028/40	4.00
100	75	4	012C/300	4.00
100	100	4	0190/400	4.00
100	125	4	01F4/500	4.00
100	1000	4	0FA0/4000	4.00

IC5_SS_SCL_LCNT

- **Name:** Standard Speed I2C Clock SCL Low Count Register
- **Size:** 16 bits
- **Address Offset:** 0x98
- Read/Write Access: Read/Write

Module::MIS	Register::IC5_SS_SCL_LCNT	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB18

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:16	-	-	-
IC_SS_SCL_LCNT	15:0	R/W	'h008f	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for standard speed. The table below shows some sample IC_SS_LCNT calculations. This register can be written only when the I2C interface is disabled which corresponds to the *IC_ENABLE* register being set to 0. Writes at other times have no effect. The minimum valid value is 8; hardware prevents values less than this being written, and if attempted results in 8 being set. For designs with APB_DATA_WIDTH = 8 the order of programming is important to ensure the correct operation of the DW_apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed. When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only.

Notice : Read-only if IC_HC_COUNT_VALUES = 1.

I²C Data Rate (kbps)	ic_clk$_{freq}$ (MHz)	SCL Low required min (us)	L_CNT (HEX/Decimal)	SCL Low Time (us)
100	2	4.7	000A/10	5.00
100	6.6	4.7	0020/32	4.85
100	10	4.7	002F/47	4.70
100	75	4.7	0161/353	4.71
100	100	4.7	01D6/470	4.70
100	125	4.7	024C/588	4.70
100	1000	4	125C/4700	4.70

IC5_FS_SCL_HCNT

- **Name:** Fast Speed I2C Clock SCL High Count Register
- **Size:** 16 bits
- **Address Offset:** 0x1c
- **Read/Write Access:** Read/Write

Module::MIS	Register::IC5_FS_SCL_HCNT	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB1C

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:16	-	-	-
IC_FS_SCL_HCNT	15:0	R/W	'h0013	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high-period count for fast speed. It is used in high-speed mode to send the Master Code and START BYTE or General CALL. The table below shows some sample IC_FS_SCL_HCNT calculations. This register goes away and becomes read-only returning 0s if IC_MAX_SPEED_MODE = standard. This register can be written only when the I2C interface is disabled, which corresponds to the *IC_ENABLE* register being set to 0. Writes at other times have no effect. The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set. For designs with APB_DATA_WIDTH == 8 the order of programming is important to ensure the correct operation of the DW_apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed. When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only.

Notice: Read-only if IC_HC_COUNT_VALUES = 1.

I²C Data Rate (kbps)	ic_clk$_{freq}$ (MHz)	SCL High required min (us)	H_CNT (HEX/Decimal)	SCL High Time (us)
400	10	0.6	0006/6	0.60
400	25	0.6	000F/15	0.60
400	50	0.6	001E/	0.60
400	75	0.6	002D/30	0.60
400	100	0.6	003C/60	0.60
400	125	0.6	004B/75	0.60
400	1000	0.6	0258/600	0.60
100 (through IC_MAX_ SPEED_MODE = standard)	N/A	N/A	disabled	N/A

IC5_FS_SCL_LCNT

- **Name:** Fast Speed I2C Clock SCL Low Count Register
- **Size:** 16 bits
- **Address Offset:** 0x20
- **Read/Write Access:** Read/Write

Module::MIS	Register::IC5_FS_SCL_LCNT	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB20

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:16	-	-	-
IC_FS_SCL_LCNT	15:0	R/W	'h0028	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for fast speed. It is used in high-speed mode to send the Master Code and START BYTE or General CALL. The table below shows some sample IC_FS_SCL_LCNT calculations. This register goes away and becomes read-only returning 0s if IC_MAX_SPEED_MODE = standard. This register can be written only when the I2C interface is disabled, which corresponds to the *IC_ENABLE* register being set to 0. Writes at other times have no effect. The minimum valid value is 8; hardware prevents values less than this being written, and if attempted results in 8 being set. For designs with APB_DATA_WIDTH = 8 the order of programming is important to ensure the

				correct operation of the DW_apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed. If the value is less than 8 then the count value gets changed to 8. When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only.

Notice : Read-only if IC_HC_COUNT_VALUES = 1.

I²C Data Rate (kbps)	ic_clk_{freq} (MHz)	SCL Low required min (us)	L_CNT (HEX/Decimal)	SCL Low Time (us)
400	10	1.3	000D/13	1.30
400	25	1.3	0021/33	1.32
400	50	1.3	0041/65	1.30
400	75	1.3	0062/98	1.31
400	100	1.3	0082/130	1.30
400	125	1.3	00A3/163	1.30
400	1000	1.3	0514/1300	1.30
100 (through IC_MAX_SPEED_MODE = standard)	N/A	N/A	disabled	N/A

IC5_INTR_STAT

- **Name:** I2C Interrupt Status Register
- **Size:** 12 bits
- **Address Offset:** 0x2C
- **Read/Write Access:** Read

Each bit in this register has a corresponding mask bit in the IC_INTR_MASK register. These bits are cleared by reading the matching interrupt clear register. The unmasked raw versions of these bits are available in the IC_RAW_INTR_STAT register.

Module::MIS	Register::IC5_INTR_STAT	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB2C

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:12	-	-	-
R_GEN_CALL	11	R	'b0	See "IC_RAW_INTR_STAT" for a detailed description of these bits.
R_START_DET	10	R	'b0	
R_STOP_DET	9	R	'b0	
R_ACTIVITY	8	R	'b0	
R_RX_DONE	7	R	'b0	
R_TX_ABRT	6	R	'b0	

R_RD_REQ	5	R	'b0	
R_TX_EMPTY	4	R	'b0	
R_TX_OVER	3	R	'b0	
R_RX_FULL	2	R	'b0	
R_RX_OVER	1	R	'b0	
R_RX_UNDER	0	R	'b0	

IC5_INTR_MASK

- **Name:** I2C Interrupt Mask Register
- **Size:** 12 bits
- **Address Offset:** 0x30
- **Read/Write Access:** Read/Write

These bits mask their corresponding interrupt status bits. They are active high; a value of 0 prevents a bit from generating an interrupt.

Module::MIS	Register::IC5_INTR_MASK	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB30

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:12	-	-	-
M_GEN_CALL	11	R/W	'b1	Masks this bit in the IC_INTR_STAT register.
M_START_DET	10	R/W	'b0	
M_STOP_DET	9	R/W	'b0	
M_ACTIVITY	8	R/W	'b0	
M_RX_DONE	7	R/W	'b1	
M_TX_ABRT	6	R/W	'b1	
M_RD_REQ	5	R/W	'b1	
M_TX_EMPTY	4	R/W	'b1	
M_TX_OVER	3	R/W	'b1	
M_RX_FULL	2	R/W	'b1	
M_RX_OVER	1	R/W	'b1	
M_RX_UNDER	0	R/W	'b1	

IC5_RAW_INTR_STAT

- **Name:** I2C Raw Interpol Status Register
- **Size:** 12 bits
- **Address Offset:** 0x34
- **Read/Write Access:** Read/Write

Unlike the IC_INTR_STAT register, these bits are not masked so they always show the true status of the DW_apb_i2c.

Module::MIS	Register::IC5_RAW_INTR_STAT	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB34

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:12	-	-	-
GEN_CALL	11	R	‘b0	Set only when a General Call address is received and it is acknowledged. It stays set until it is cleared either by disabling DW_apb_i2c or when the CPU reads bit 0 of the IC_CLR_GEN_CALL register. DW_apb_i2c stores the received data in the Rx buffer.
START_DET	10	R	‘b0	Indicates whether a START or RESTART condition has occurred on the I2C interface regardless of whether DW_apb_i2c is operating in slave or master mode.
STOP_DET	9	R	‘b0	Indicates whether a STOP condition has occurred on the I2C interface regardless of whether DW_apb_i2c is operating in slave or master mode.
ACTIVITY	8	R	‘b0	This bit captures DW_apb_i2c activity and stays set until it is cleared. There are four ways to clear it: ● Disabling the DW_apb_i2c ● Reading the IC_CLR_ACTIVITY register ● Reading the IC_CLR_INTR register ● System reset Once this bit is set, it stays set unless one of the four methods is used to clear it. Even if the DW_apb_i2c module is idle, this bit remains set until cleared, indicating that there was activity on the bus.
RX_DONE	7	R	‘b0	When the DW_apb_i2c is acting as a slave-transmitter, this bit is set to 1 if the master does not acknowledge a transmitted byte. This occurs on the last byte of the transmission, indicating that the transmission is done.
TX_ABRT	6	R	‘b0	This bit indicates if DW_apb_i2c, as an I2C transmitter, is unable to complete the intended actions on the contents of the transmit FIFO. This situation can occur both as an I2C master or an I2C slave, and is referred to as a “transmit abort”. When this bit is set to 1, the IC_TX_ABRT_SOURCE register indicates the reason why the transmit abort takes places. NOTE: The DW_apb_i2c flushes/resets/empties the TX FIFO whenever this bit is set. The TX FIFO remains in this flushed state until the register IC_CLR_TX_ABRT is read. Once this read is performed, the TX FIFO is then ready to accept more data bytes from the APB interface.
RD_REQ	5	R	‘b0	This bit is set to 1 when DW_apb_i2c is acting as a slave and another I2C master is attempting to read data from DW_apb_i2c. The DW_apb_i2c holds the I2C bus in a wait state (SCL=0) until this interrupt is serviced, which means that the slave has been addressed by a remote master that is asking for data to be transferred. The processor must respond to this interrupt and then write the requested data to the IC_DATA_CMD register. This bit is set to 0 just after the processor reads the IC_CLR_RD_REQ register.

TX_EMPTY	4	R	‘b0	This bit is set to 1 when the transmit buffer is at or below the threshold value set in the IC_TX_TL register. It is automatically cleared by hardware when the buffer level goes above the threshold. When the IC_ENABLE bit 0 is 0, the TX FIFO is flushed and held in reset. There the TX FIFO looks like it has no data within it, so this bit is set to 1, provided there is activity in the master or slave state machines. When there is no longer activity, then with ic_en=0, this bit is set to 0.
TX_OVER	3	R	‘b0	Set during transmit if the transmit buffer is filled to IC_TX_BUFFER_DEPTH and the processor attempts to issue another I2C command by writing to the IC_DATA_CMD register. When the module is disabled, this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.
RX_FULL	2	R	‘b0	Set when the receive buffer reaches or goes above the RX_TL threshold in the IC_RX_TL register. It is automatically cleared by hardware when buffer level goes below the threshold. If the module is disabled (IC_ENABLE[0]=0), the RX FIFO is flushed and held in reset; therefore the RX FIFO is not full. So this bit is cleared once the IC_ENABLE bit 0 is programmed with a 0, regardless of the activity that continues.
RX_OVER	1	R	‘b0	Set if the receive buffer is completely filled to IC_RX_BUFFER_DEPTH and an additional byte is received from an external I2C device. The DW_apb_i2c acknowledges this, but any data bytes received after the FIFO is full are lost. If the module is disabled (IC_ENABLE[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.
RX_UNDER	0	R	‘b0	Set if the processor attempts to read the receive buffer when it is empty by reading from the IC_DATA_CMD register. If the module is disabled (IC_ENABLE[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.

IC5_RX_TL

- **Name:** I2C Receive FIFO Threshold Register
- **Size:** 8bits
- **Address Offset:** 0x38
- **Read/Write Access:** Read/Write

Module::MIS	Register::IC5_RX_TL		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB38
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:8	-	-	-		
RX_TL	7:0	R/W	‘h00	Receive FIFO Threshold Level Controls the level of entries (or above) that triggers the		

				RX_FULL interrupt. The valid range is 0-255, with the additional restriction that hardware does not allow this value to be set to a value larger than the depth of the buffer. If an attempt is made to do that, the actual value set will be the maximum depth of the buffer. A value of 0 sets the threshold for 1 entry, and a value of 255 sets the threshold for 256 entries.**Depth of the rx_buffer is 32.**

IC5_TX_TL

- **Name:** I2C Transmit FIFO Threshold Register
- **Size:** 8 bits
- **Address Offset:** 0x3c
- Read/Write Access: Read/Write

Module::MIS	Register::IC5_TX_TL		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB3C
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:8	-	-	-		
TX_TL	7:0	R/W	'h00	Transmit FIFO Threshold Level Controls the level of entries (or below) that trigger the TX_EMPTY interrupt. The valid range is 0-255, with the additional restriction that it may not be set to value larger than the depth of the buffer. If an attempt is made to do that, the actual value set will be the maximum depth of the buffer. A value of 0 sets the threshold for 0 entries, and a value of 255 sets the threshold for 255 entries.**Depth of the tx_buffer is 32.**		

IC5_CLR_INTR

- **Name:** Clear Combined and Individual Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x40
- **Read/Write Access:** Read

Module::MIS	Register::**IC5_CLR_INTR**		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB40
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:1	-	-	-		
CLR_INTR	0	R	'b0	Read this register to clear the combined interrupt, all individual interrupts, and the IC_TX_ABRT_SOURCE register. This bit does not clear hardware clearable interrupts but software clearable interrupts. Refer to Bit 9 of the		

				IC_TX_ABRT_SOURCE register for an exception to clearing IC_TX_ABRT_SOURCE.

IC5_CLR_RX_UNDER

- **Name:** Clear RX_UNDER Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x44
- **Read/Write Access:** Read

Module::MIS	Register::IC5_CLR_RX_UNDER		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB44
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:1	-	-	-		
CLR_RX_UNDER	0	R	'b0	Read this register to clear the *RX_UNDER* interrupt.		

IC5_CLR_RX_OVER

- **Name:** Clear RX_OVER Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x48
- **Read/Write Access:** Read

Module::MIS	Register::IC5_CLR_RX_OVER		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB48
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:1	-	-	-		
CLR_RX_OVER	0	R	'b0	Read this register to clear the *RX_OVER* interrupt.		

IC5_CLR_TX_OVER

- **Name:** Clear TX_OVER Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x4c
- **Read/Write Access:** Read

Module::MIS	Register::IC5_CLR_TX_OVER		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB4C
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:1	-	-	-		
CLR_TX_OVER	0	R	'b0	Read this register to clear the *TX_OVER* interrupt.		

IC5_CLR_RD_REQ

- **Name:** Clear RD_REQ Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x50
- **Read/Write Access:** Read

Module::MIS	Register::IC5_CLR_RD_REQ	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB50

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:1	-	-	-
CLR_RD_REQ	0	R	'b0	Read this register to clear the *RD_REQ* interrupt.

IC5_CLR_TX_ABRT

- **Name:** Clear TX_ABRT Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x54
- **Read/Write Access:** Read

Module::MIS	Register::IC5_CLR_TX_ABRT	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB54

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:1	-	-	-
CLR_TX_ABRT	0	R	'b0	Read this register to clear the *TX_ABRT* interrupt, and the IC_TX_ABRT_SOURCE register.

IC5_CLR_RX_DONE

- **Name:** Clear RX_DONE Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x58
- **Read/Write Access:** Read

Module::MIS	Register::IC5_CLR_RX_DONE	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB58

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:1	-	-	-
CLR_RX_DONE	0	R	'b0	Read this register to clear the *RX_DONE* interrupt.

IC5_CLR_ACTIVITY

- **Name:** ACTIVITY Status Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x5c
- **Read/Write Access:** Read

Module::MIS	Register::IC5_CLR_ACTIVITY		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB5C
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:1	-	-	-		
CLR_ACTIVITY	0	R	'b0	Read this register to get status of the *ACTIVITY* interrupt. It is automatically cleared by hardware.		

IC5_CLR_STOP_DET

- **Name:** Clear STOP_DET Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x60
- **Read/Write Access:** Read

Module::MIS	Register::IC5_CLR_STOP_DET		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB60
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:1	-	-	-		
CLR_STOP_DET	0	R	'b0	Read this register to clear the *STOP_DET* interrupt.		

IC5_CLR_START_DET

- **Name:** Clear START_DET Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x64
- **Read/Write Access:** Read

Module::MIS	Register::IC5_CLR_START_DET		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB64
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:1	-	-	-		
CLR_START_DET	0	R	'b0	Read this register to clear the *START_DET* interrupt.		

IC5_CLR_GEN_CALL

- **Name:** Clear GEN_CALL Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x68
- **Read/Write Access:** Read

Module::MIS	Register::IC5_CLR_GEN_CALL		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB68
Name	**Bits**	**Read**	**Reset**	**Comments**		

		/Write	State	
Rvd	31:1	-	-	-
CLR_GEN_CALL	0	R	‘b0	Read this register to clear the *GEN_CALL* interrupt.

IC5_ENABLE

- Name: I2C Enable Register
- **Size:** 1 bit
- **Address Offset:** 0x6c
- **Read/Write Access:** Read/Write

Module::MIS	Register::IC5_ENABLE		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB6C
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:1	-	-	-		
ENABLE	0	R/W	‘b0	Controls whether the DW_apb_i2c is enabled. 0: Disables DW_apb_i2c (TX and RX FIFOs are held in an erased state) 1: Enables DW_apb_i2c Software can disable DW_apb_i2c while it is active. However, it is important that care be taken to ensure that DW_apb_i2c is disabled properly. When DW_apb_i2c is disabled, the following occurs: • The TX FIFO and RX FIFO get flushed. • Status bits in the IC_INTR_STAT register are still active until DW_apb_i2c goes into IDLE state.If the module is transmitting, it stops as well as deletes the contents of the transmit buffer after the current transfer is complete. If the module is receiving, the DW_apb_i2c stops the current transfer at the end of the current byte and does not acknowledge the transfer.In systems with asynchronous pclk and ic_clk when IC_CLK_TYPE parameter set to asynchronous (1), there is a two ic_clk delay when enabling or disabling the DW_apb_i2c.		

IC5_STATUS

- **Name:** I2C Status Register
- **Size:** 5 bits
- **Address Offset:** 0x70
- **Read/Write Access:** Read

This is a read-only register used to indicate the current transfer status and FIFO status. The status register may be read at any time. None of the bits in this register request an interrupt.
When the I2C is disabled by writing 0 in bit 0 of the IC_ENABLE register:

- Bits 1 and 2 are set to 1
- Bits 3 and 4 are set to 0

When the master or slave state machines goes to idle and ic_en=0:

- Bits 5 and 6 are set to 0

Module::MIS	Register::IC5_STATUS		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB70
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:7	-	-	-		
SLV_ACTIVITY	6	R	'b0	Slave FSM Activity Status. When the Slave Finite State Machine (FSM) is not in the IDLE state, this bit is set. 0: Slave FSM is in IDLE state so the Slave part of DW_apb_i2c is not Active 1: Slave FSM is not in IDLE state so the Slave part of DW_apb_i2c is Active		
MST_ACTIVITY	5	R	'b0	Master FSM Activity Status. When the Master Finite State Machine (FSM) is not in the IDLE state, this bit is set. 0: Master FSM is in IDLE state so the Master part of DW_apb_i2c is not Active 1: Master FSM is not in IDLE state so the Master part of DW_apb_i2c is Active		
RFF	4	R	'b0	Receive FIFO Completely Full. When the receive FIFO is completely full, this bit is set. When the receive FIFO contains one or more empty location, this bit is cleared. 0 – Receive FIFO is not full 1 – Receive FIFO is full		
RFNE	3	R	'b0	Receive FIFO Not Empty. Set when the receive FIFO contains one or more entries and is cleared when the receive FIFO is empty. This bit can be polled by software to completely empty the receive FIFO. 0 – Receive FIFO is empty 1 – Receive FIFO is not empty		
TFE	2	R	'b1	Transmit FIFO Completely Empty. When the transmit FIFO is completely empty, this bit is set. When it contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt. 0 – Transmit FIFO is not empty 1 – Transmit FIFO is empty		
TFNF	1	R	'b1	Transmit FIFO Not Full. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full. 0 – Transmit FIFO is full 1 – Transmit FIFO is not full		
ACTIVITY	0	R	'b0	I2C Activity Status.		

IC5_TXFLR

- **Name:** I2C Transmit FIFO Level Register
- **Size:** 4
- **Address Offset:** 0x74

•**Read/Write Access:** Read

This register contains the number of valid data entries in the transmit FIFO buffer. It is cleared whenever:

- The I2C is disabled
- There is a transmit abort—that is, TX_ABRT bit is set in the IC_RAW_INTR_STAT register
- The slave bulk transmit mode is aborted

The register increments whenever data is placed into the transmit FIFO and decrements when data is taken from the transmit FIFO.

Module::MIS	Register::IC5_TXFLR		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB74
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:4	-	-	-		
TXFLR	3:0	R	'b0	**Transmit FIFO Level**. Contains the number of valid data entries in the transmit FIFO.		

IC5_RXFLR

•**Name:** I2C Receive FIFO Level Register
•**Size:** 4
•**Address Offset:** 0x78
•**Read/Write Access:** Read

This register contains the number of valid data entries in the receive FIFO buffer. It is cleared whenever:

- The I2C is disabled
- Whenever there is a transmit abort caused by any of the events tracked in IC_TX_ABRT_SOURCE

The register increments whenever data is placed into the receive FIFO and decrements when data is taken from the receive FIFO.

Module::MIS	Register::IC5_RXFLR		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB78
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:4	-	-	-		
RXFLR	3:0	R	'h0	**Receive FIFO Level**. Contains the number of valid data entries in the receive FIFO.		

IC5_SDA_HOLD

•**Name:** I2C SDA Hold Time Length Register
•**Size:** 16
•**Address Offset:** 0x7C
•**Read/Write Access:** Read / Write

This register controls the amount of hold time on the SDA signal after a negative edge of SCL in both master and slave mode, in units of ic_clk period. The value programmed must be greater than the minimum hold time in each mode for the value to be implemented – one cycle in master mode, seven cycles in slave mode. Writes to this register succeed only when IC_ENABLE=0.
The programmed SDA hold time cannot exceed at any time the duration of the low part of SCL. Therefore the programmed value cannot be larger then N_SCL_LOW-2, where N_SCL_LOW is the duration of the low part of the SCL period measured in ic_clk cycles.

Module::MIS	Register::IC5_SDA_HOLD		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB7C
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:16	-	-	-		
SDA_HOLD	15:0	R/W	'h0001	Sets the required SDA hold time in units of ic_clk period.		

IC5_TX_ABRT_SOURCE

- **Name:** I2C Transmit Abort Source Register
- **Size:** 16 bits
- **Address Offset:** 0x80
- **Read/Write Access:** Read

This register has 16 bits that indicate the source of the TX_ABRT bit. Except for Bit 9, this register is cleared whenever the IC_CLR_TX_ABRT register or the IC_CLR_INTR register is read. To clear Bit 9, the source of the ABRT_SBYTE_NORSTRT must be fixed first; RESTART must be enabled (IC_CON[5]=1), the SPECIAL bit must be cleared (IC_TAR[11]), or the GC_OR_START bit must be cleared (IC_TAR[10]). Once the source of the ABRT_SBYTE_NORSTRT is fixed, then this bit can be cleared in the same manner as other bits in this register. If the source of the ABRT_SBYTE_NORSTRT is not fixed before attempting to clear this bit, Bit 9 clears for one cycle and is then re-asserted.

Module::MIS	Register::IC5_TX_ABRT_SOURCE		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB80
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:16	-	-	-		
ABRT_SLVRD_INTX	15	R	'b0	1: When the processor side responds to a slave mode request for data to be transmitted to a remote master and user writes a 1 in CMD (bit 8) of IC_DATA_CMD register.		
ABRT_SLV_ARBLOST	14	R	'b0	1: Slave lost the bus while transmitting data to a remote master. IC_TX_ABRT_SOURCE[12] is set at		

				the same time. Note: Even though the slave never "owns" the bus, something could go wrong on the bus. This is a fail safe check. For instance, during a data transmission at the low-to-high transition of SCL, if what is on the data bus is not what is supposed to be transmitted, then DW_apb_i2c no longer own the bus.
ABRT_SLVFLUSH_TXFIFO	13	R	'b0	1: Slave has received a read command and some data exists in the TX FIFO so the slave issues a TX_ABRT interrupt to flush old data in TX FIFO.
ARB_LOST	12	R	'b0	1: Master has lost arbitration, or if IC_TX_ABRT_SOURCE[14] is also set, then the slave transmitter has lost arbitration. Note: I2C can be both master and slave at the same time.
ARB_MASTER_DIS	11	R	'b0	1: User tries to initiate a Master operation with the Master mode disabled.
ABRT_10B_RD_NORSTRT	10	R	'b0	1: The restart is disabled (IC_RESTART_EN bit (IC_CON[5]) = 0) and the master sends a read command in 10-bit addressing mode.
ABRT_SBYTE_NORSTRT	9	R	'b0	To clear Bit 9, the source of the ABRT_SBYTE_NORSTRT must be fixed first; restart must be enabled (IC_CON[5]=1), the SPECIAL bit must be cleared (IC_TAR[11]), or the GC_OR_START bit must be cleared (IC_TAR[10]). Once the source of the ABRT_SBYTE_NORSTRT is fixed, then this bit can be cleared in the same manner as other bits in this register. If the source of the ABRT_SBYTE_NORSTRT is not fixed before attempting to clear this bit, bit 9 clears for one cycle and then gets re-asserted. 1: The restart is disabled (IC_RESTART_EN bit (IC_CON[5]) = 0) and the user is trying to send a START Byte.
ABRT_HS_NORSTRT	8	R	'b0	1: The restart is disabled (IC_RESTART_EN bit (IC_CON[5]) = 0) and the user is trying to use the master to transfer data in High Speed mode.
ABRT_SBYTE_ACKDET	7	R	'b0	1: Master has sent a START Byte and the START Byte was acknowledged (wrong behavior).
ABRT_HS_ACKDET	6	R	'b0	1: Master is in High Speed mode and the High Speed Master code was acknowledged (wrong behavior).
ABRT_GCALL_READ	5	R	'b0	1: DW_apb_i2c in master mode sent a General Call but the user programmed the byte following the General Call to be a read from the bus (IC_DATA_CMD[9] is set to 1).

ABRT_GCALL_NOACK	4	R	'b0	1: DW_apb_i2c in master mode sent a General Call and no slave on the bus acknowledged the General Call.
ABRT_TXDATA_NOACK	3	R	'b0	1: This is a master-mode only bit. Master has received an acknowledgement for the address, but when it sent data byte(s) following the address, it did not receive an acknowledge from the remote slave(s).
ABRT_10ADDR2_NOACK	2	R	'b0	1: Master is in 10-bit address mode and the second address byte of the 10-bit address was not acknowledged by any slave.
ABRT_10ADDR1_NOACK	1	R	'b0	1: Master is in 10-bit address mode and the first 10-bit address byte was not acknowledged by any slave.
ABRT_7B_ADDR_NOACK	0	R	'b0	1: Master is in 7-bit addressing mode and the address sent was not acknowledged by any slave.

IC5_SLV_DATA_NACK_ONLY

- Name: Generate Slave Data NACK Register
- Size: 1 bit
- Address Offset: 0x84
- Read/Write Access: Read/Write

The register is used to generate a NACK for the data part of a transfer when DW_apb_i2c is acting as a slave-receiver. This register only exists when the IC_SLV_DATA_NACK_ONLY parameter is set to 1. When this parameter disabled, this register does not exist and writing to the register's address has no effect.

A write can occur on this register if either of the following conditions are met:

- DW_apb_i2c is disabled (IC_ENABLE[0] = 0)
- Slave part is inactive (IC_STATUS[6] = 0)

Module::MIS	Register::IC5_SLV_DATA_NACK_ONLY	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB84

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:1	-	-	-
NACK	0	R/W	'b0	Generate NACK. This NACK generation only occurs when DW_apb_i2c is a slave-receiver. If this register is set to a value of 1, it can only generate a NACK after a data byte is received; hence, the data transfer is aborted and the data received is not pushed to the receive buffer. When the register is set to a value of 0, it generates NACK/ACK, depending on normal criteria. 1 = generate NACK after data byte received 0 = generate NACK/ACK normally

IC5_DMA_CR

- **Name:** DMA Control Register
- **Size:** 2 bits
- **Address Offset:** 0x88
- **Read/Write Access:** Read/Write

This register is only valid when DW_apb_i2c is configured with a set of DMA Controller interface signals (IC_HAS_DMA = 1). When DW_apb_i2c is not configured for DMA operation, this register does not exist and writing to the register's address has no effect and reading from this register address will return zero. The register is used to enable the DMA Controller interface operation. There is a separate bit for transmit and receive. This can be programmed regardless of the state of IC_ENABLE.

Module::MIS	Register::IC5_DMA_CR		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB88
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:2	-	-	-		
TDMAE	1	R/W	'b0	**Transmit DMA Enable**. This bit enables/disables the transmit FIFO DMA channel. 0 = Transmit DMA disabled 1 = Transmit DMA enabled		
RDMAE	0	R/W	'b0	**Receive DMA Enable**. This bit enables/disables the receive FIFO DMA channel. 0 = Receive DMA disabled 1 = Receive DMA enabled		

IC5_DMA_TDLR

- **Name:** DMA Transmit Data Level Register
- **Size:** 2 bits
- **Address Offset:** 0x8c
- **Read/Write Access:** Read/Write

This register is only valid when the DW_apb_i2c is configured with a set of DMA interface signals (IC_HAS_DMA = 1). When DW_apb_i2c is not configured for DMA operation, this register does not exist; writing to its address has no effect; reading from its address returns zero.

Module::MIS	Register::IC5_DMA_TDLR		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB8C
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:3	-	-	-		
DMATDL	2:0	R/W	'h0	**Transmit Data Level**. This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and TDMAE = 1.		

IC5_DMA_RDLR

- **Name:** I2C Receive Data Level Register
- **Size:** 2 bits
- **Address Offset:** 0x90
- **Read/Write Access:** Read/Write

This register is only valid when DW_apb_i2c is configured with a set of DMA interface signals (IC_HAS_DMA = 1). When DW_apb_i2c is not configured for DMA operation, this register does not exist; writing to its address has no effect; reading from its address returns zero.

Module::MIS	Register::IC5_DMA_RDLR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB90

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:4	-	-	-
DMARDL	3:0	R/W	'h0	**Receive Data Level**. This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or more than this field value + 1, and RDMAE =1. For instance, when DMARDL is 0, then dma_rx_req is asserted when 1 or more data entries are present in the receive FIFO.

IC5_SDA_SETUP

- **Name: I2C SDA Setup Register**
- **Size: 8 bits**
- **Address Offset: 0x94**
- **Read/Write Access: Read/Write**

This register controls the amount of time delay (in terms of number of ic_clk clock periods) introduced

in the rising edge of SCL, relative to SDA changing, when DW_apb_i2c services a read request in a

slave-transmitter operation. The relevant I2C requirement is tSU:DAT (note 4) as detailed in the I2C Bus

Specification.

Module::MIS	Register::IC5_SDA_SETUP		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB94
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:8	-	-	-		
SDA_SETUP	7:0	R/W	'h0	SDA Setup. It is recommended that if the required delay is 1000ns, then for an ic_clk frequency of 10 MHz, IC_SDA_SETUP should be programmed to a value of 11.		

IC5_ACK_GENERAL_CALL

- Name: I2C ACK General Call Register
- Size: 1 bit
- Address Offset: 0x98
- Read/Write Access: Read/Write

The register controls whether DW_apb_i2c responds with a ACK or NACK when it receives an I2C General Call address.

Module::MIS	Register::IC5_ACK_GENERAL_CALL		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB98
Name	**Bits**	**Read /Write**	**Reset State**	**Comments**		
Rvd	31:1	-	-	-		
ACK_GEN_CALL	0	R/W	'h0	ACK General Call. When set to 1, DW_apb_i2c responds with a ACK (by asserting ic_data_oe) when it receives a General Call. Otherwise, DW_apb_i2c responds with a NACK (by negating ic_data_oe).		

IC5_ENABLE_STATUS

- Name: I2C Enable Status Register
- Size: 3 bits
- Address Offset: 0x9C
- Read/Write Access: Read

The register is used to report the DW_apb_i2c hardware status when the IC_ENABLE register is set from 1 to 0; that is, when DW_apb_i2c is disabled.

If IC_ENABLE has been set to 1, bits 2:1 are forced to 0, and bit 0 is forced to 1.

If IC_ENABLE has been set to 0, bits 2:1 is only be valid as soon as bit 0 is read as '0'.

Module::MIS	Register::IC5_ENABLE_STATUS	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BB9C

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:3	-	-	-
SLV_RX_DATA_LOST	2	R/W	'h0	Slave Received Data Lost. This bit indicates if a Slave-Receiver operation has been aborted with at least one data byte received from an I2C transfer due to the setting of IC_ENABLE from 1 to 0.When read as 1, DW_apb_i2c is deemed to have been actively engaged in an aborted I2C transfer (with matching address) and the data phase of the I2C transfer has been entered, even though a data byte has been responded with a NACK. NOTE: If the remote I2C master terminates the transfer with a STOP condition before the DW_apb_i2c has a chance to NACK a transfer, and IC_ENABLE has been set to 0, then this bit is also set to 1.When read as 0, DW_apb_i2c is deemed to have been disabled without being actively involved in the data phase of a Slave-Receiver transfer. NOTE: The CPU can safely read this bit when IC_EN (bit 0) is read as 0.
SLV_DISABLED_WHILE_BUSY	1	R/W	'h0	Slave Disabled While Busy (Transmit, Receive). This bit indicates if a potential or active Slave operation has been aborted due to the setting of the IC_ENABLE register from 1 to 0. This bit is set when the CPU writes a 0 to the IC_ENABLE register while: (a) DW_apb_i2c is receiving the address byte of the Slave-Transmitter operation from a remote master; OR, (b) address and data bytes of the Slave-Receiver operation from a remote master. When read as 1, DW_apb_i2c is deemed to have forced a NACK during any part of an I2C transfer, irrespective of whether the I2C address matches the slave address set in DW_apb_i2c (IC_SAR register) OR if the transfer is completed before IC_ENABLE is set to 0 but has not taken effect. NOTE: If the remote I2C master terminates the transfer with a STOP condition before the DW_apb_i2c has a chance to NACK a transfer, and IC_ENABLE has been set to 0, then this bit will also be set to 1. When read as 0, DW_apb_i2c is deemed to have been disabled when there is master activity, or when the I2C bus is idle. NOTE: The CPU can safely read this bit when IC_EN (bit 0) is read as 0.
IC_EN	0	R/W	'h0	ic_en Status. This bit always reflects the value driven on the output port ic_en. When read as 1, DW_apb_i2c is deemed to be in an enabled state. When read as 0, DW_apb_i2c is deemed completely inactive. NOTE: The CPU can safely read this bit anytime. When

				this bit is read as 0, the CPU can safely read SLV_RX_DATA_LOST (bit 2) and SLV_DISABLED_WHILE_BUSY (bit 1).

IC5_COMP_PARAM_1

- **Name:** Component Parameter Register 1
- **Size:** 32 bits
- **Address Offset:** 0xf4
- **Read/Write Access:** Read

Module::MIS	Register::IC5_COMP_PARAM_1	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BBF4

Name	Bits	Read /Write	Reset State	Comments
Rvd	31:24	-	-	-
TX_BUFFER_DEPTH	23:16	R	'h07	The value of this register is derived from the IC_TX_BUFFER_DEPTH coreConsultant parameter. 0x00 = Rvd 0x01 = 2 0x02 = 3 to 0xFF = 256
RX_BUFFER_DEPTH	15:8	R	'h07	The value of this register is derived from the IC_RX_BUFFER_DEPTH coreConsultant parameter. For a description of this parameter, see DesignWare DW_apb_i2c Databook, DesignWare DW_apb_i2c Databook, Table 4 on page 46. 0x00 = Rvd 0x01 = 2 0x02 = 3 to 0xFF = 256
ADD_ENCODED_PARAMS	7	R	'b1	The value of this register is derived from the IC_ADD_ENCODED_PARAMS coreConsultant parameter. For a description of this parameter, see DesignWare DW_apb_i2c Databook, Table 4 on page 46. 0 = False 1 = True
HAS_DMA	6	R	'b0	The value of this register is derived from the IC_HAS_DMA coreConsultant parameter. For a description of this parameter, see DesignWare DW_apb_i2c Databook, Table 4 on page 46. 0 = False 1 = True
INTR_IO	5	R	'b1	The value of this register is derived from the

				IC_INTR_IO coreConsultant parameter. For a description of this parameter, see DesignWare DW_apb_i2c Databook, Table 4 on page 46. 0 = Individual 1 = Combined
HC_COUNT_VALUES	4	R	'b0	The value of this register is derived from the IC_HC_COUNT_VALUES coreConsultant parameter. For a description of this parameter, see DesignWare DW_apb_i2c Databook, Table 4 on page 46. 0 = False 1 = True
MAX_SPEED_MODE	3:2	*R*	'b10	The value of this register is derived from the IC_MAX_SPEED_MODE coreConsultant parameter. For a description of this parameter, see DesignWare DW_apb_i2c Databook, Table 4 on page 46. 0x0 = Rvd 0x1 = Standard 0x2 = Fast 0x3 = High
APB_DATA_WIDTH	1:0	R	'b10	The value of this register is derived from the IC_MAX_SPEED_MODE coreConsultant parameter. For a description of this parameter, see DesignWare DW_apb_i2c Databook, Table 4 on page 46. 0x0 = 8 bits 0x1 = 16 bits 0x2 = 32 bits 0x3 = Rvd

Notice: This is a constant read-only register that contains encoded information about the component's parameter settings. The reset value depends on coreConsultant parameter(s).

IC5_COMP_VERSION

- **Name:** I2C Component Version Register
- **Size:** 32 bits
- **Address Offset:** 0xf8
- **Read/Write Access:** Read

Module::MIS	Register::IC5_COMP_VERSION	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BBF8

Name	Bits	Read /Write	Reset State	Comments
IC_COMP_VERSION	31:0	R	'h3130332a	Lists the releases/versions of the DW_apb_i2c component and the I2C_COMP_VERSION value.

DesignWare AMBA Release	DW_apb_i2c Version	I2C_COMP_VERSION value	Databook Date
2004.06	1.03a	31_30_33_2A	June 21, 2004

IC5_COMP_TYPE

- **Name:** I2C Component Type Register
- **Size:** 32 bits
- **Address Offset:** 0xfc
- **Read/Write Access:** Read

Module::MIS	Register::IC5_COMP_TYPE	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BBFC

Name	Bits	Read /Write	Reset State	Comments
IC_COMP_TYPE	31:0	R	'h44570140	Designware Component Type number = 0x44_57_01_40. This assigned unique hex value is constant and is derived from the two ASCII letters "DW" followed by a 16-bit unsigned number.

4 Timer Control

4.1 Register

4.1.1 Register Summery

Physical Address	*Name*	*R/W*	*Description*
0x9801_B500	MIS_TC0TVR	R/W	Timer/Counter 0 Target Value Register
0x9801_B504	MIS_TC1TVR	R/W	Timer/Counter 1 Target Value Register
0x9801_B508	MIS_TC2TVR	R/W	Timer/Counter 2 Target Value Register
0x9801_B50C	MIS_TC0CVR	R/W	Timer/Counter 0 Current Value Register
0x9801_B510	MIS_TC1CVR	R/W	Timer/Counter 1 Current Value Register
0x9801_B514	MIS_TC2CVR	R/W	Timer/Counter 2 Current Value Register
0x9801_B518	MIS_TC0CR	R/W	Timer/Counter 0 Control Register
0x9801_B51C	MIS_TC1CR	R/W	Timer/Counter 1 Control Register
0x9801_B520	MIS_TC2CR	R/W	Timer/Counter 2 Control Register
0x9801_B524	MIS_TC0ICR	R/W	Timer/Counter 0 Interrupt Control Register
0x9801_B528	MIS_TC1ICR	R/W	Timer/Counter 1 Interrupt Control Register
0x9801_B52C	MIS_TC2ICR	R/W	Timer/Counter 2 Interrupt Control Register
0x9801_B530	Reserved	-	-
0x9801_B534	Reserved	-	-
0x9801_B538	MIS_CLK90K_CTRL	R/W	clk_90KHz counter Control Register.
0x9801_B53C	MIS_SCPU_CLK27M_90K	R	SCPU clk27MHz to clk90kHz divider
0x9801_B540	MIS_SCPU_CLK90K_LO	R	SCPU 90kHz timer current value low word
0x9801_B544	MIS_SCPU_CLK90K_HI	R	SCPU 90kHz timer current value high word
0x9801_B548	MIS_ACPU_CLK27M_90K	R	ACPU clk27MHz to clk90kHz divider
0x9801_B54C	MIS_ACPU_CLK90K_LO	R	ACPU 90kHz timer current value low word
0x9801_B550	MIS_ACPU_CLK90K_HI	R	ACPU 90kHz timer current value high word
0x9801_B554	Reserved	-	-
0x9801_B558	Reserved	-	-
0x9801_B55C	Reserved	-	-
0x9801_B560	Reserved	-	-
0x9801_B564	Reserved	-	-
0x9801_B568	Reserved	-	-
0x9801_B56C	-	-	-
0x9801_B570	MIS_PCR_CLK90K_CTRL	R/W	PCR clk_90KHz counter Control Register.

0x9801_B574	MIS_PCR_SCPU_CLK27M_90K	R	PCR SCPU clk27MHz to clk90kHz divider
0x9801_B578	MIS_PCR_SCPU_CLK90K_LO	R	PCR SCPU 90kHz timer current value low word
0x9801_B57C	MIS_PCR_SCPU_CLK90K_HI	R	PCR SCPU 90kHz timer current value high word
0x9801_B580	MIS_PCR_ACPU_CLK27M_90K	R	PCR ACPU clk27MHz to clk90kHz divider
0x9801_B584	MIS_PCR_ACPU_CLK90K_LO	R	PCR ACPU 90kHz timer current value low word
0x9801_B588	MIS_PCR_ACPU_CLK90K_HI	R	PCR ACPU 90kHz timer current value high word
0x9801_B58C	Reserved	-	-
0x9801_B590	Reserved	-	-
0x9801_B594	Reserved	-	-
0x9801_B598	Reserved	-	-
0x9801_B59C	Reserved	-	-
0x9801_B5A0	Reserved	-	-
0x9801_B5A4 ~ 0x9801_B5AC	Reserved	-	-
0x9801_B5B0	MIS_TCWCR	R/W	Timer/Counter Watchdog Control Register
0x9801_B5B4	MIS_TCWTR	R/W	Timer/Counter Watchdog Trigger Register
0x9801_B5B8	MIS_TCWNMI	R/W	Timer/Counter Watchdog NMI Register
0x9801_B5BC	MIS_TCWOV	R/W	Timer/Counter Watchdog overflow Register
0x9801_B5C0	MIS_TCWCR_SWC	R/W	Timer/Counter Watchdog Control Register in SWC
0x9801_B5C4	MIS_TCWTR_SWC	R/W	Timer/Counter Watchdog Trigger Register in SWC
0x9801_B5C8	MIS_TCWNMI_SWC	R/W	Timer/Counter Watchdog NMI Register in SWC
0x9801_B5CC	MIS_TCWOV_SWC	R/W	Timer/Counter Watchdog overflow Register in SWC

p.s: 0x9801_B570 ~ 0x9801_B59F reserved for clk_pcr domain

4.1.2 Register Description

Module::MIS	Register::TC0TVR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B500

Name	Bits	Read/Write	Reset State	Comments
TC0TVR	31:0	R/W	-	The timer/counter 0 target value 0 and 1 are not allowed. Counts from 0 to

				TC0TVR.

Module::MIS	Register::TC1TVR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B504

Name	Bits	Read/Write	Reset State	Comments
TC1TVR	31:0	R/W	-	The timer/counter 1 target value 0 and 1 are not allowed. Counts from 0 to TC1TVR.

Module::MIS	Register::TC2TVR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B508

Name	Bits	Read/Write	Reset State	Comments
TC2TVR	31:0	R/W	-	The timer/counter 2 target value 0 and 1 are not allowed. Counts from 0 to TC2TVR.

Notice: TC0,TC1,TC2 count at 27MHz

Module::MIS	Register::TC0CVR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B50C

Name	Bits	Read/Write	Reset State	Comments
TC0CVR	31:0	R	-	The timer/counter 0 current value.

Module::MIS	Register::TC1CVR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B510

Name	Bits	Read/Write	Reset State	Comments
TC1CVR	31:0	R	-	The timer/counter 1 current value.

Module::MIS	Register::TC2CVR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B514

Name	Bits	Read/Write	Reset State	Comments
TC2VR	31:0	R	-	The timer/counter 2 current value.

Module::MIS	Register::TC0CR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B518

Name	Bits	Read/Write	Reset State	Comments
TC0En	31	R/W	‘b0	Timer/Counter 0 enable 0: Disable TC0. Clear TC0CVR to 32’b0 1: Enable TC0. Timer/Counter 0 count at 27MHz
TC0Mode	30	R/W	‘b0	Timer/Counter 0 mode 0: counter mode 1: timer mode
TC0Pause	29	R/W	‘b0	Timer/Counter 0 pause.

				1: Pause
RvdA	28:24	R/W	‘h0	Rvd Register.
Rvd	23:0	-	-	-

Module::MIS	Register::TC1CR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B51C

Name	Bits	Read/Write	Reset State	Comments
TC1En	31	R/W	‘b0	Timer/Counter 1 enable 0: Disable TC1. Clear TC1CVR to 32’b0 1: Enable TC1. Timer/Counter 1 count at 27MHz
TC1Mode	30	R/W	‘b0	Timer/Counter 1 mode 0=counter mode 1=timer mode
TC1Pause	29	R/W	‘b0	Timer/Counter 1 pause. 1: Pause
RvdA	28:24	R/W	‘h0	Rvd Register.
Rvd	23:0	-	-	-

Module::MIS	Register::TC2CR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B520

Name	Bits	Read/Write	Reset State	Comments
TC2En	31	R/W	‘b0	Timer/Counter 2 enable 0: Disable TC2. Clear TC2CVR to 32’b0 1: Enable TC2. Timer/Counter 2 count at 27MHz
TC2Mode	30	R/W	‘b0	Timer/Counter 2 mode 0=counter mode 1=timer mode
TC2Pause	29	R/W	‘b0	Timer/Counter 2 pause. 1: Pause
RvdA	28:24	R/W	‘h0	Rvd Register.
Rvd	23:0	-	-	-

Module::MIS	Register::TC0ICR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B524

Name	Bits	Read/Write	Reset State	Comments
TC0IE	31	R/W	‘b0	Timer/Counter 0 interrupt enable.
Rvd	30:0	-	-	-

Module::MIS	Register::TC1ICR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B528

Name	Bits	Read/Write	Reset	Comments

			State	
TC1IE	31	R/W	‘b0	Timer/Counter 1 interrupt enable.
Rvd	30:0	-	-	-

Module::MIS	Register::TC2ICR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B52C

Name	Bits	Read/Write	Reset State	Comments
TC2IE	31	R/W	‘b0	Timer/Counter 2 interrupt enable.
Rvd	30:0	-	-	-

Module::MIS	Register::CLK90K_CTRL	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B538

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:1	-	-	-
EN	0	R/W	‘b1	1: 90KHz counter enable. 0: 90KHz counter disable. Set EN=1’b0 to clear CLK90KHz counter to 48’b0. EN can control CLK90KHz counter only, it can’t control clk27M_clk90K_cnt[8:0].

Notice1: clk27MHz to clk90KHz counter

Notice2:

1. If SCPU want to read 90KHz counter, it must read MIS_SCPU_CLK90K_LO first and it will generate a latch enable signal to latch MIS_SCPU_CLK90K_HI and MIS_SCPU_CLK27M_90K at the same time.

2. ACPU have the same function to SCPU. The only different are that they must read the counter register belong to ACPU.

Module::MIS	Register::SCPU_CLK27M_90K	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B53C

Name	Bits	Read/Write	Reset	Comments

			State	
Rvd	31:9	-	-	-
CNT	8:0	R	‘h0	SCPU clk_27MHz to clk_90KHz count. Read MIS_SCPU_CLK90K_LO first to ensure the correct CNT.

Module::MIS	Register::SCPU_CLK90K_LO	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B540

Name	Bits	Read/Write	Reset State	Comments
VAL	31:0	R	‘h0	90KHz timer current value bit31:0.

Module::MIS	Register::SCPU_CLK90K_HI	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B544

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:16	-	-	-
VAL	15:0	R	‘h0	90KHz timer current value bit47:32. Read MIS_SCPU_CLK90K _LO first to ensure the correct MIS_SCPU_CLK90K_HI.

Module::MIS	Register::ACPU_CLK27M_90K	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B548

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:9	-	-	-
CNT	8:0	R	‘h0	ACPU clk_27MHz to clk_90KHz count. Read MIS_ACPU_CLK90K_LO first to ensure the correct CNT.

Module::MIS	Register::ACPU_CLK90K_LO	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B54C

Name	Bits	Read/Write	Reset State	Comments
VAL	31:0	R	‘h0	90KHz timer current value bit31:0.

Module::MIS	Register::ACPU_CLK90K_HI	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B550

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:16	-	-	-

VAL	15:0	R	'h0	90KHz timer current value bit47:32. Read MIS_ACPU_CLK90K _LO first to ensure the correct MIS_ACPU_CLK90K_HI.

Module::MIS	Register::PCR_CLK90K_CTRL	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B570

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:1	-	-	-
EN	0	R/W	'b1	1: 90KHz counter enable. 0: 90KHz counter disable. Set EN=1'b0 to clear CLK90KHz counter to 48'b0. EN can control CLK90KHz counter only, it can't control clk27M_clk90K_cnt[8:0].

Notice1: clk27MHz to clk90KHz counter

Notice2:

3. If SCPU want to read 90KHz counter, it must read MIS_PCR_SCPU_CLK90K_LO first and it will generate a latch enable signal to latch MIS_PCR_SCPU_CLK90K_HI and MIS_PCR _SCPU_CLK27M_90K at the same time.

4. ACPU have the same function to SCPU. The only different are that they must read the counter register belong to ACPU.

Module::MIS	Register::PCR_SCPU_CLK27M_90K	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B574

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:9	-	-	-
CNT	8:0	R	'h0	SCPU clk_27MHz to clk_90KHz count. Read MIS_SCPU_CLK90K_LO first to ensure the correct CNT.

Module::MIS	Register::PCR_SCPU_CLK90K_LO	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B578

Name	Bits	Read/Write	Reset State	Comments
VAL	31:0	R	'h0	90KHz timer current value bit31:0.

Module::MIS	Register::PCR_SCPU_CLK90K_HI	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B57C

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:16	-	-	-
VAL	15:0	R	'h0	90KHz timer current value bit47:32. Read MIS_PCR _SCPU_CLK90K _LO first to ensure the correct MIS_PCR _SCPU_CLK90K_HI.

Module::MIS	Register::PCR_ACPU_CLK27M_90K	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B580

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:9	-	-	-
CNT	8:0	R	'h0	ACPU clk_27MHz to clk_90KHz count. Read MIS_ACPU_CLK90K_LO first to ensure the correct CNT.

Module::MIS	Register::PCR_ACPU_CLK90K_LO	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B584

Name	Bits	Read/Write	Reset State	Comments
VAL	31:0	R	'h0	90KHz timer current value bit31:0.

Module::MIS	Register::PCR_ACPU_CLK90K_HI	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B588

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:16	-	-	-
VAL	15:0	R	'h0	90KHz timer current value bit47:32. Read MIS_PCR _ACPU_CLK90K _LO first to ensure the correct MIS_PCR _ACPU_CLK90K_HI.

Module::MIS	Register::TCWCR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B5B0

Name	Bits	Read/Write	Reset	Comments

			State	
WD_INT_EN	31	R/W	‘h1	Watchdoe interrupt enable
Rvd	30:16	-	-	-
NMIC	15:12	R/W	‘h0	Watchdog NMI Occurrence Count Write “1111” to clear NMIC. Other value can’t effect NMIC. Reset to zero by power-on reset or bottom reset only. Watchdog reset and software reset can’t reset NMIC.
WDC	11:8	R/W	‘h0	Watchdog Occurrence Count Write “1111” to clear WDC. Other value can’t effect WDC. Reset to zero by power-on reset or bottom reset only. Watchdog reset and software reset can’t reset WDC.
WDEN	7:0	R/W	‘hA5	Watchdog Enable. When these bits are set to 0xA5, the watchdog timer stops. Other value can enable the watchdog timer and cause a system reset when an overflow signal occurs.

Note1: watchdog count at 27MHz
Note2: WDC and NMIC only can reset by power-on-reset and bottom reset(RESET_N).
Note3: WDEN can reset by power-on-reset, bottom reset(RESET_N) and wdog_reset.

MIS watchdog(NWC or SWC) overflow 時會 reset watchdog(NWC+SWC) enable, interrupt enable , ov_sel and nmi_sel
NWC overflow 發生時會 clear NWC 內部 counter , 不會 clear SWC 內部 counter
SWC overflow 發生時會 clear SWC 內部 counter , 不會 clear NWC 內部 counter
所以要注意使用前要 clear counter TCWTR_WDCLR

Module::MIS	Register::TCWTR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B5B4

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:1	-	-	-
WDCLR	0	W	‘b0	Watchdog clear. Write “1” to clear the watchdog counter. It is auto cleared after the write.

Module::MIS	Register::TCWNMI	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B5B8

Name	Bits	Read/Write	Reset State	Comments
SEL	31:0	R/W	‘hffffffff	Watchdog NMI select.

				NMI condition = TCWNMI_SEL * 1/27MHz ex: TCWNMI_SEL | NMI time (us) 32'h0070_0000 | 271852 32'h0080_0000 | 310688 32'h00f0_0000 | 582541 32'h0100_0000 | 621377

Note1: TCWNMI can reset by power-on-reset, bottom reset(RESET_N) and wdog_reset.

Module::MIS	Register::TCWOV	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B5BC

Name	Bits	Read/Write	Reset State	Comments
SEL	31:0	R/W	'hffffffff	Watchdog overflow select. overflow condition = TCWOV_SEL * 1/27MHz ex: TCWOV_SEL | overflow time (us) 32'h0070_0000 | 271852 32'h0080_0000 | 310688 32'h00f0_0000 | 582541 32'h0100_0000 | 621377

Note1: TCWOV can reset by power-on-reset, bottom reset(RESET_N) and wdog_reset.

Module::MIS	Register::TCWCR_SWC	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B5C0

Name	Bits	Read/Write	Reset State	Comments
WD_INT_EN	31	R/W	'h1	Watchdoe interrupt enable
Rvd	30:16	-	-	-
NMIC	15:12	R/W	'h0	Watchdog NMI Occurrence Count Write "1111" to clear NMIC. Other value can't effect NMIC. Reset to zero by power-on reset or bottom reset only. Watchdog reset and software reset can't reset NMIC.

WDC	11:8	R/W	‘h0	Watchdog Occurrence Count Write “1111” to clear WDC. Other value can’t effect WDC. Reset to zero by power-on reset or bottom reset only. Watchdog reset and software reset can’t reset WDC.
WDEN	7:0	R/W	‘hA5	Watchdog Enable. When these bits are set to 0xA5, the watchdog timer stops. Other value can enable the watchdog timer and cause a system reset when an overflow signal occurs.

Note1: watchdog count at 27MHz
Note2: WDC and NMIC only can reset by power-on-reset and bottom reset(RESET_N).
Note3: WDEN can reset by power-on-reset, bottom reset(RESET_N) and wdog_reset.

MIS watchdog(NWC or SWC) overflow 時會 reset watchdog(NWC+SWC) enable, interrupt enable , ov_sel and nmi_sel
NWC overflow 發生時會 clear NWC 內部 counter , 不會 clear SWC 內部 counter
SWC overflow 發生時會 clear SWC 內部 counter , 不會 clear NWC 內部 counter
所以要注意使用前要 clear counter TCWTR_WDCLR

Module::MIS	Register::TCWTR_SWC	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B5C4

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:1	-	-	-
WDCLR	0	W	‘b0	Watchdog clear. Write “1” to clear the watchdog counter. It is auto cleared after the write.

Module::MIS	Register::TCWNMI_SWC	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B5C8

Name	Bits	Read/Write	Reset State	Comments
SEL	31:0	R/W	‘hffffffff	Watchdog NMI select. NMI condition = TCWNMI_SWC_SEL * 1/27MHz ex:

TCWNMI_SEL	NMI time (us)
32'h0070_0000	271852
32'h0080_0000	310688
32'h00f0_0000	582541
32'h0100_0000	621377

Note1: TCWNMI can reset by power-on-reset, bottom reset(RESET_N) and wdog_reset.

Module::MIS	Register::TCWOV_SWC	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B5CC

Name	**Bits**	**Read/Write**	**Reset State**	**Comments**
SEL	31:0	R/W	‘hffffffff	Watchdog overflow select. overflow condition = TCWOV_SWC_SEL * 1/27MHz ex:

TCWOV_SEL	overflow time (us)
32'h0070_0000	271852
32'h0080_0000	310688
32'h00f0_0000	582541
32'h0100_0000	621377

Note1: TCWOV can reset by power-on-reset, bottom reset(RESET_N) and wdog_reset.

5 Real-time clock

5.1 Register

5.1.1 Register Summery

Physical Address	*Name*	*R/W*	*Description*
0x9801_B600	MIS_RTCSEC	R/W	current RTC second
0x9801_B604	MIS_RTCMIN	R/W	current RTC minute
0x9801_B608	MIS_RTCHR	R/W	current RTC hour
0x9801_B60C	MIS_RTCDATE1	R/W	current RTC date [7:0]
0x9801_B610	MIS_RTCDATE2	R/W	current RTC date [13:8]
0x9801_B614	MIS_ALMMIN	R/W	Alarm minute
0x9801_B618	MIS_ALMHR	R/W	Alarm hour
0x9801_B61C	MIS_ALMDATE1	R/W	Alarm date [7:0]
0x9801_B620	MIS_ALMDATE2	R/W	Alarm date [13:8]
0x9801_B624	MIS_RTCSTOP	R	RTC STOP
0x9801_B628	MIS_RTCACR	R/W	RTC analog circuit control register
0x9801_B62C	MIS_RTCEN	R/W	RTC enable register
0x9801_B630	MIS_RTCCR	R/W	RTC control register

5.1.2 Register Description

Module::MIS	Register::RTCSEC	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B600

Name	**Bits**	**Read/Write**	**Reset State**	**Comments**
Rvd	31:7	-	-	-
RTCSEC	6:0	R/W	-	Reading register gets the current RTC second, and writing this register updates new RTC second.

Module::MIS	Register::RTCMIN	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B604

Name	**Bits**	**Read/Write**	**Reset State**	**Comments**
Rvd	31:6	-	-	-
RTCMIN	5:0	R/W	-	Reading register gets the current RTC Minute, and writing this register updates new RTC Minute

Module::MIS	Register::RTCHR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B608

Name	**Bits**	**Read/Write**	**Reset State**	**Comments**
Rvd	31:5	-	-	-
RTCHR	4:0	R/W	-	RTC hour.

Module::MIS	Register::RTCDATE1	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B60C

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:8	-	-	-
RTCDATE1	7:0	R/W	-	RTC date LSB.

Module::MIS	Register::RTCDATE2	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B610

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:7	-	-	-
RTCDATE2	6:0	R/W	-	RTC date MSB.

Module::MIS	Register::ALMMIN	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B614

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:6	-	-	-
ALMMIN	5:0	R/W	-	Alarm minute

Module::MIS	Register::ALMHR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B618

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:5	-	-	-
ALMHR	4:0	R/W	-	Alarm hour.

Module::MIS	Register::ALMDATE1	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B61C

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:8	-	-	-
ALMDATE1	7:0	R/W	-	Alarm date LSB.

Module::MIS	Register::ALMDATE2	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B620

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:7	-	-	-
ALMDATE2	6:0	R/W	-	Alarm date MSB.

Module::MIS	Register::RTCSTOP	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B624

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:3	-	-	-
VREF	2:1	R/W	-	RTC power-on-cell Voltage Reference.
RTCSTOP	0	R	-	Reading this register to check RTC is STOP or not. Reading '1' indicates RTC is stopped and reading '0' indicates RTC is running

Module::MIS	Register::RTCACR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B628

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:8	-	-	-
RTCPWR	7	R/W	-	RTC power status. SW write this bit to 1'b1 when RTC power stable. This bit will be clear to zero if RTC power-on-cell issue reset.
CO	6:5	R/W	-	RTC Crystal RC control. Please SW write it to zero.
CI	4:3	R/W	-	RTC Crystal RC control. Please SW write it to zero.
CLKSEL	2	R/W	-	RTC_clk select. 0: RTC_clk= 32.768KHz for normal function. 1: RTC_clk=27MHz for RTC counter test mode.
BC	1:0	R/W	-	Oscillator bias current control.

p.s.1: SW should set CKSEL, BC to zero to initialize RTC.
p.s.2: CLKSEL have two function. One is select RTC_clk frequency. Second is select RTC_testcnt source for MP testing.
CLKSEL=1'b0: RTC_testcnt output RTC_clk for analog_test_mode.
CLKSEL=1'b1: RTC_testcnt output "RTC counter test mode output" for digital_test.
After take the pllrtc of Sirius. (add the function for 27MHz crystal, but iso_mis debounce the hsecint in phoenix. Need to fix in Unicorn)
Rtc_test == 1'b0 , normal mode RTC_CKSEL can choose the counter in 32768Hz or 27MHz
Rtc_test ==1'b1 , test mode
RTC_CKSEL can choose the RTC_testcnt output RTC_clk or RTC_testcnt

Nike

RTC_CKSEL	RTC_clk	Hsec_int	Application
0	32.768KHz	14'h3fff/32.768K = 0.5sec	hsec
1			Only for CP test

Sirius : in normal mode(rtc_test==1'b0)

RTC_CKSEL	RTC_clk	Hsec_int	Application
0	32.768KHz	14'h3fff/32.768K = 0.5sec	32.768KHz osc
1	27MHz	24'd13499999/27M=0.5sec	27MHz osc

Application :
RTC_CKSEL
1'b0:
Use XIRTC_APAD, XORTC_APAD to generate 32768Hz to RTC_CLK
1'b1: Connect XIRTC_APAD to RTC_CLK directly.

RTC_test : anlg_test_mode , shorter counter for CP

Module::MIS	Register::RTCEN	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B62C

Name	Bits	Read/Write	Reset State	Comments

Rvd	31:8	-	-	-
RTCEN	7:0	R/W	-	RTC enable. 8'h5A: enable RTC counter. 8'h00: disable RTC counter. Stop RTC counter. Firmware has to read RTCSTOP to ensure RTC is stop. Only use 8'h00 to disable RTC, other value maybe have glitch to hardware and can't disable RTC.

Note: If RTCEN isn't equal to 8'h5A, it means that user have not initialize RTC timer or RTC power have ever exhausted. The RTC timer value isn't available anymore and user to set the RTC timer. DVR should ignore the value of RTC timer register and display a default time to the VFD. GUI also can display a hint to request the user to setup the time of DVR.

Module::MIS	Register::RTCCR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_B630

Name	Bits	Read/Write	Reset State	Comments
Rvd	31:7	-	-	-
RTCRST	6	R/W	'b0	Writing '1' to reset RTC counter (MIS_RTCSEC, MIS_RTCMIN, MIS_RTCHR,MIS_RTCDATE1,MIS_RTCDATE2) asynchronously until re-writing this bit to '0'. Firmware has to disable RTC before using this reset function.
Rvd	5:4	-	-	-
DAINTE	3	R/W	'b0	Enable date interrupt.
HUINTE	2	R/W	'b0	Enable hour interrupt.
MUINTE	1	R/W	'b0	Enable minute interrupt.
HSUINTE	0	R/W	'b0	Enable half second interrupt.

6.2 Register Sets

GSPI module 0

Module::GSPI	Register::SCK_CTRL	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_BD00

Name	Bits	R/W	Default	Comments
Rvd	31..24	-	-	Reserved
sck_divider	23..16	R/W	‘h1	Frequency divider for SCK 8’h0: 1 : 247MHz (not support) 8’h1: 2 : 123.5MHz (not support) 8’h2: 4 : 61.75MHz 8’h3: 6 : 41.16MHz 8’h4: 8 : 30.1MHz 8’h5: 10 : 24.7MHz 8’h6:12 : 20.6MHz 8’h7:14 : 17.6MHz 8’h8:16 : 15.43MHz 8’h9:20 : 12.35MHz 8’ha:24 : 10.3MHz 8’hb:48: 5.14MHz 8’hc:256 : 0.96MHz 8’hd:512 : 0.48MHz 8’he:2048 : 0.12MHz 8’hf: 2560 : 0.096MHz 8’h10: 3 (not support) 8’h11: 5 8’h12: 7 Others : reserve
Rvd	15..9	-	-	Reserved
Phase_adjust	8	R/W	‘b0	Adjust the phase of address/write data changed at falling edge in mode0
Rvd	7..2	-	-	Reserved
phase	1	R/W	‘b0	Phase selection 1’b1: if polarity = 0, so align to positive edge of sck; sample si at negative edge of sck if polarity = 1, so align to negative edge of sck; sample si at postive edge of sck 1’b0: if polarity = 1, so align to positive edge of sck; sample si at negative edge of sck if polarity = 0, so align to negative edge of sck; sample si at positive edge of sck

polarity	0	R/W	‘b0	Polarity selection 1’b1: sck idle state = H 1’b0: sck idle state = L

Module::GSPI	Register::CS_TIMING	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_BD04

Name	Bits	R/W	Default	Comments
Rvd	31	-	-	
CS_O_VAL	30	R/W	‘b0	CS output value setting at CS_SEL==1’b1
CS_OE_VAL	29	R/W	‘b0	CS_OE setting at CS_SEL==1’b1
CS_SEL	28	R/W	‘b0	CS control select 1’b0: CS pin be controlled auto 1’b1: CS pin be controlled by CS_O_VAL and CS_OE_VAL
Rvd	27..24	-	-	
Tcs_high	23..16	R/W	‘b0	CS falling edge to previous CS rising edge time 8’h0 : 1 bus_clk 8’hff: 256 bus_clk
Tcs_end	15..8	R/W	‘b0	Last SCK active edge to CS rising time 8’h0 : 0 bus_clk 8’hff: 255 bus_clk
Tcs_start	7..0	R/W	‘b0	CS falling edge to first SCK active edge time 8’h0: 1 bus_clk 8’hff: 256 bus_clk

Module::GSPI	Register::AUX_CTRL	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_BD08

Name	Bits	R/W	Default	Comments
Rvd	31..9	-	-	
SLAVE_MODE_ENABLE	8	R/W	‘b0	Slave mode enable 1’b0: Master mode 1’b1: Slave mode
Rvd	7..2	-	-	
CS_1_ENABLE	1	R/W	‘b0	CS 1 output eanble 1’b0: disable CS_1 output 1’b1: enable CS_1 output
CS_0_ENABLE	0	R/W	‘b1	CS 0 output eanble 1’b0: disable CS_0 output

				1’b1: enable CS_0 output

Module::GSPI	Register::CMD_DUMMY	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_BD0C

Name	Bits	R/W	Default	Comments
Rvd	31..23	-	-	Reserved
CMD_LEN	22..16	R/W	‘h0	Command length: 7’h0: no command out 7’h1: command = 1 bit 7’h20: command = 32 bits
Rvd	15..6	-	-	Reserved
DMY_LEN	5: 0	R/W	‘h0	Dummy bit Length: 6’h0 : No Dummy phase 6’h1 : 1 bit dummy 6’h10 : 16 bits dummy 6’h1f : 31 bits dummy 6’h20 : 32 bits dummy

Module::GSPI	Register::MOSI_CTRL	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_BD10

Name	Bits	R/W	Default	Comments
RWCMD	31	R/W	‘b0	Read or Write Command. 0: read (default) 1: write
Master_Full_duplex	30	R/W	‘b0	Master Full duplex mode 1: SI is input mode when SO is output mode (RWCMD = 1’b1) 1’b0: master single mode(Receive only) PS. No implement.
ADDR_LEN	29..22	R/W	‘h0	ADDR bit Length. 8’h0: No address phase 8’h1: 1 bit addr 8’h20: 32 bit addr others : reserved
LSB first	21	R/W	‘b0	MSB first or LSB first selection 1’b1: LSB is the 1st bit on so 1’b0: MSB is the 1st bit on so
Dual_mode_sel	20..19	R/W	‘h0	dual mode selection 2’h0: no dual mode 2’h1: only data phase in dual mode 2’h2: both address phase & data phase in dual mode others : reserved

First_bit_sel	18	R/W	'b0	First bit selection (only in dual mode) 1'b0: first bit on SI 1'b1: first bit on SO
Rvd	17	-	-	Reserved
Slave_Full_duplex	16	R/W	'b0	Slave Full duplex mode 1'b1: SI is input mode when SO is output mode 1'b0: slave single mode (Transmit only) PS. No implement.
WR_LEN	15..0	R/W	'b0	Write Data bit Length: Number of write data bits (minimum number is 1 bit) 16'h0: 0 bit (can't use) 16'h1: 1 bits 16'h8: 8 bits 16'h7f: 127 bits 16'h3ff:1023 bits 16'h400: 1024 bits 16'h800: 2048 bits 16'h2000:8192 bits Support length as below sheet

Module::GSPI	Register::MISO_CTRL	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_BD14

Name	Bits	R/W	Default	Comments
Rvd	31..29	-	-	
LSB first	28	R/W	'b0	MSB first or LSB first selection 1'b1: LSB is the 1st bit on si 1'b0:MSB is the 1st bit on si
Rvd	27..16	-	-	
RD_LEN	15..0	R/W	'b0	Read Data bit Length: Number of read data bits 16'h0:0 bit 16'h1:1 bits 16'h8: 8 bits 16'h7f:127 bits 16'h3ff:1023 bits 16'h400:1024 bits 16'h800:2048 bits 16'h2000:8192 bits Support length as below sheet

P.S.
MISI_CTRL_LSB first and MISO_CTRL_LSB first function
Because of register FIFO depth is 8*32 and md path getting data order issue, there is a 128bits limit when MSB_first(LSB_first==1'b0)
Solution :
1. Repeat the 128bits read/write command

2. Check the swap function in MD application.
3. Use LSB+Swap function

GSPI Data Length Support		
	Write data length (Byte)	Read data length (Byte)
Rbus+MSB	1~32	1~32
Rbus+LSB	1~32	1~32
Rbus+LSB+Swap	1~32	1~32
Rbus+DMA+MSB	1~16	1~16
Rbus+DMA+LSB	1~16, 32, 64, 96, 128	1~16, 32, 64, 96, 128
Rbus+DMA+LSB+Swap	1~16, 32, 64, 96, 128	1~16, 32, 64, 96, 128
Descriptor +DMA+MSB	1~16	1~16
Descriptor +DMA+LSB	1~16, 32, 64, 96, 128	1~16, 32, 64, 96, 128
Descriptor +DMA+LSB+Swap	1~16, 32, 64, 96, 128	1~16, 32, 64, 96, 128

Module::GSPI	Register::W_FIFO	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_BD18

Name	Bits	R/W	Default	Comments
W_FIFO	31..0	R/W	'h0	32 Bytes Data FIFO. (Write)

Module::GSPI	Register::R_FIFO	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_BD1C

Name	Bits	R/W	Default	Comments
R_FIFO	31..0	R/W	'h0	32 Bytes Data FIFO. (Read)

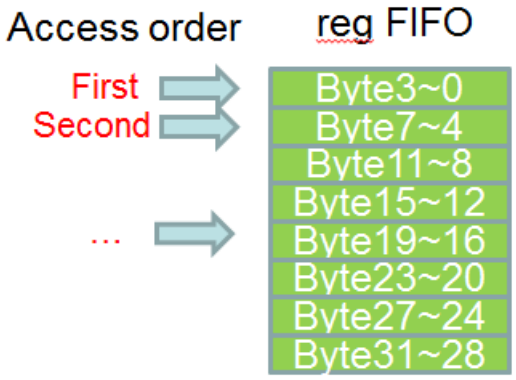
In the MSB/LSB mode the data sending order of Byte and bit of Byte.
MSB mode: byte from High to Low , bit from High to Low
LSB mode : byte from Low to High , bit from Low

P.S.
SW should to take care the Access order of W_FIFO and R_FIFO below in **MSB** mode
Write the W_FIFO oder before command enable or Read the R_FIFO order after command done.

The First time access the W/R_FIFO , will store/get the Byte 3~0 in MSB order
The Second time access the W/R_FIFO , will store/get the Byte 7~4 in BSB order.
The same order continued to 8th access.

For Example , If SW want to write 7 byte data in W_FIFO like 0x070605_04030201

First step : Write the W_FIFO 0x04030201
Second step : write the W_FIFO 0x070605 , and trig the command.
The data will send in GSPI waveform in the order 0x07, 0x06,…,0x02,0x01
Read is the same as the rule.
First step : access the R_FIFO , SW will get the 0x04030201
Second step : access the R_FIFO , SW will get the 0x070605



Module::GSPI	Register::INSTRUCTION		Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_BD20
Name	Bits	R/W	Default	Comments		
SPI_INS	31..0	R/W	'h0	SPI command (instruction)		

Module::GSPI	Register::ADDR		Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_BD24
Name	Bits	R/W	Default	Comments		
SPI_ADDR	31..0	R/W	'h0	SPI address		

Module::GSPI	Register::SPI_CTRL		Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_BD28
Name	Bits	R/W	Default	Comments		
Rvd	31..22	-	-			
CAL_SEL	21..20	R/W	'h1	Read date sample timing select 2'b00: original 2'b01: delay 1T sys_clk 2'b10: delay 2T sys_clk 2'b11: delay 3T sys_clk		
Rvd	19..18	-	-			
R_FIFO_ALMOST_	17	R/W	'b0	R_FIFO_almost_full_sel		

FULL_SEL				0: 1/2 FIFO valid (16 Bytes) 1: 1/4 FIFO valid (8 Bytes)
W_FIFO_ALMOST_FULL_SEL	16	R/W	‘b0	W_FIFO_almost_empty_sel 0: 1/2 FIFO valid (16 Bytes) 1: 1/4 FIFO valid (8 Bytes)
Rvd	15..9	-	-	
FIFO_PTR_RST	8	W	‘b0	Reset R_FIFO & W_FIFO pointer 1’b1 : reset R_FIFO & W_FIFO pointer, this bit is need clear by SW. If no clear it, W_FIFO can not use.
RDFIFO_BSWAP	7	R/W	‘b0	RDFIFO_BSWAP 0: no swap function 1: RD_DATA byte swap Ex : [7:0] -> [0:7], [15:8] -> [8:15]
CONTI_TRANS_EN	6	R/W	‘b0	Continuous Transfer mode enable 1’b0: CS will go de-assert after finishing single command 1’b1: CS will not go de-assert until last command finished. PS. No implement.
CONTI_TRANS_LAST	5	R/W	‘b0	Continuous Transfer mode Last command 1’b1: last command in Continuous Transfer mode, (valid only when CONTI_TRANS_EN = 1’b1)
DES_MODE_ENABLE	4	R/W	‘b0	Descriptor mode enable 1’b0: descriptor mode disable, SW updates GSPI control registers when CMD done 1’b1: descriptor mode enable, SW prepares commands in DRAM first, then sets Des_mode_enable =1, GSPI will issue read command to MD to get commands for control registers, this bit is auto clear when all commands for descriptor mode has been executed.
Rvd	3..2	-	-	
DMA_MODE_ENABLE	1	R/W	‘b0	DMA mode enable 1’b1: W/R data from/to MD interface 1’b0 W/R data from/to R-bus
CMD_ENABLE	0	R/W	‘b0	Command Enable

				1'b1 : issue spi cmd, this bit is auto clear when command finished

Note:
DES_MODE_ENABLE = 1, & DMA_MODE_ENABLE =1, means cmd/data from/to MD
DES_MODE_ENABLE = 1, & DMA_MODE_ENABLE =0, no function
DES_MODE_ENABLE = 0, & DMA_MODE_ENABLE =1, means data from/to MD
DES_MODE_ENABLE = 0, & DMA_MODE_ENABLE =0, means data from/to rbus interface

Module::GSPI	Register::SPI_STATUS	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_BD2C

Name	Bits	R/W	Default	Comments
Rvd	31..29	-	-	
R_FIFO_PTR	28..24	R	'h0	R_FIFO_PTR
Rvd	23..21	-	-	
W_FIFO_PTR	20..16	R	'h0	W_FIFO_PTR
Rvd	15..8	-	-	
Rvd	7..2	-	-	
MD_empty	1	R	'b0	MD empty status 1'b1: issue request to MD, but MD empty Write one to clear
SFC_BUSY	0	R	'b0	SFC busy now 1'b1 : SPI controller is still in action

Module::GSPI	Register::SPI_INT	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_BD30

Name	Bits	R/W	Default	Comments
Rvd	31..26	-	-	
R_FIFO_ALMOST_FULL	25	R	'b0	R_FIFO_ALMOST_FULL,, 1/2 or 1/4 FIFO valid for read data, base on 0x9801_BD20[17]
R_FIFO_FULL	24	R	'b0	R_FIFO_FULL, 16 bytes data stored in R_FIFO PS. Not correct when MSB first
Rvd	23..18	-	-	
W_FIFO_ALMOST_EMPTY	17	R	'b0	WR_FIFO_almost_empty, 1/2 or 1/4 FIFO valid for write data, base

				on 0x9801_BD20[16]
W_FIFO_EMPTY	16	R	‘b0	W_FIFO_EMPTY, no data stored in 16 Bytes FIFO PS. Not correct when MSB first
Rvd	15..8	-	-	
Rvd	7..3	-	-	
SYN_CODE_ERR	2	R	‘b0	Synchronization code error 1’b1 : write one to clear this bit
DES_MODE_DONE	1	R	‘b0	Descriptor mode Done 1’b1 : write one to clear this bit
CMD_DONE	0	R	‘b0	Command Done 1’b1 : write one to clear this bit

Module::GSPI	Register::SPI_INT_EN	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_BD34

Name	Bits	R/W	Default	Comments
Rvd	31..26	-	-	
R_FIFO_ALMOST_FULL	25	R/W	‘b0	R_FIFO_ALMOSTFULL interrupt enable
R_FIFO_FULL	24	R/W	‘b0	R_FIFO_FULL
Rvd	23..18	-	-	
W_FIFO_ALMOST_EMPTY	17	R/W	‘b0	WR_FIFO_ALMOST_EMPTY interrupt enable
W_FIFO_EMPTY	16	R/W	‘b0	W_FIFO_EMPTY interrupt enable
Rvd	15..8	-	-	
Rvd	7..3	-	-	
SYN_CODE_ERR_IE	2	R/W	‘b0	Synchronization code error interrupt enable
DES_MODE_DONE_IE	1	R/W	‘b0	Descriptor mode Command Done interrupt enable
CMD_DONE_IE	0	R/W	‘b0	Command Done interrupt enable

Module::GSPI	Register::SPI_SYN_CTRL	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_BD38

Name	Bits	R/W	Default	Comments
SYN_MODE_EN	31	R/W	‘b0	Synchronization mode enable 1’b0: disable 1’b1: enable
Rvd	30..14	-	-	
SYN_CODE_LENGTH	13..8	R/W	‘h0	Synchronization code bit Length: Number of synchronization code bits
Rvd	7..6	-	-	

RESPONSE_LENGTH	5..0	R/W	‘h0	Response bit Length: Number of response bits

Module::GSPI	Register::SYN_CODE	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_BD3C

Name	Bits	R/W	Default	Comments
SYN_CODE	31..0	R	‘h0	32 bits synchronization code

Module::GSPI	Register::SYN_CODE_MASK	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_BD40

Name	Bits	R/W	Default	Comments
SYN_CODE_MASK	31..0	R/W	‘h0	32 bits mask for synchronization code

Module::GSPI	Register::SYN_CODE_VAL	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_BD44

Name	Bits	R/W	Default	Comments
SYN_CODE_VAL	31..0	R/W	‘h0	32 bits check value for synchronization code

Module::GSPI	Register:: POWER	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_BD48

Name	Bits	R/W	Default	Comments
Rvd	31..1	-	-	-
gating_en	0	R/W	‘b1	Enable power gating function.

Module::GSPI	Register:: DUMMY1	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_BD50

Name	Bits	R/W	Default	Comments
dummy1	31..0	R/W	‘h0	Dummy

Module::GSPI	Register:: DUMMY2	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_BD54

Name	Bits	R/W	Default	Comments
dummy2	31..0	R/W	‘hffffffff	Dummy

7 DC-FAN CTRL

7.1 Register

7.1.1 Register Summery

Physical Address	*Name*	*R/W*	*Description*
0x9801_BC00	FAN_CTRL	R/W	DC FAN control
0x9801_BC04	FAN_DEBOUNCE	R/W	DC FAN de-bounce select
0x9801_BC08	FAN_TIMER_TAR_VAL	R/W	DC FAN timer target value
0x9801_BC0C	FAN_TIMER_CNT_VAL	R	DC FAN timer count value
0x9801_BC10	FAN_COUNTER_CNT_VAL	R/W	DC FAN counter count value

7.1.2 Register Description

Module::MIS	Register::FAN_CTRL		Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_BC00
Name	**Bits**	**Read/Write**	**Reset State**	**Comments**		
Rvd	31..2	-	-	-		
INT_EN	1	R/W	‘b0	1’b1: Eanble DC FAN function interrupt		
EN	0	R/W	‘b0	1’b1: Eanble DC FAN function		

Module::MIS	Register::FAN_DEBOUNCE		Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_BC04
Name	**Bits**	**Read/Write**	**Reset State**	**Comments**		
Rvd	31..4	-	-	-		
write_en1	3	W	-	Write enable for bit[2:0]		
CLK	2..0	R/W	‘b0	De-bounce clock base. 3’h7: 30ms 3’h6: 20ms 3’h5: 10ms 3’h4: 1ms 3’h3: 100us 3’h2: 10us 3’h1: 1us 3’h0: 37ns (27MHz)		

Module::MIS	Register::FAN_TIMER_TV	Set::1	ATTR::ctrl	Type::SR	ADDR::0x9801_BC08
Name	**Bits**	**Read/Write**	**Reset State**	**Comments**	
TAR_VAL	31..0	R/W	‘b0	DC FAN timer target value	

Module::MIS	Register::FAN_TIMER_CV	Set::1	ATTR::nor	Type::SR	ADDR::0x9801_BC0C
Name	**Bits**	**Read/Write**	**Reset State**	**Comments**	
CNT_VAL	31..0	R	‘b0	DC FAN timer count value	

Module::MIS	Register::FAN_COUNTER_CV	Set::1	ATTR::nor_up	Type::SR	ADDR::0x9801_BC10
Name	**Bits**	**Read/Write**	**Reset State**	**Comments**	
CNT_VAL	31..0	R/W	‘b0	DC FAN counter count value	

8 UART H5

8.1 Register

8.1.1 Register Summary

Physical Address	*Name*	*R/W*	*Description*
0x9801_BE00	MIS_SC0_FP	R/W	Smartcard0 Carrier Frequency Programmer
0x9801_BE04	MIS_SC0_CR	R/W	Smartcard0 Control Register
0x9801_BE08	MIS_SC0_PCR	R/W	Smartcard0 Protocol Control Register
0x9801_BE0C	MIS_SC0_TXFIFO	W	Smartcard0 Transmit FIFO Register
0x9801_BE10			
0x9801_BE14	MIS_SC0_RXFIFO	R	Smartcard0 Receive FIFO Register
0x9801_BE18	MIS_SC0_RXLENR	R	Smartcard0 Receive Length Register
0x9801_BE1C	MIS_SC0_FCR	R	Smartcard0 Flow Control Register
0x9801_BE20	MIS_SC0_IRSR	R/W	Smartcard0 Interrupt Status Register
0x9801_BE24	MIS_SC0_IRER	R/W	Smartcard0 Interrupt Enable Register

8.1.2 SC0 Register Description

Module::MIS	Register::SC0_FP	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BE00

Name	Bits	Read/ Write	Reset State	Comments
Rvd	31:25	-	-	-
CLK_EN	24	R/W	‘b1	Write “1” to enable clock. If 0, SC_CLK and io_clk are stopped. Remaining high or low is determined by MIS_SC0_CR[CLK_STOP] bit.
SC_CLKDIV	23:18	R/W	‘h00	Controls the frequency of the SC_CLK I/O pin. Its value is from 0~31. f_{sc_clk} = (system clock) / (PRE_CLKDIV + 1) / (SC_CLKDIV+1)
BAUDDIV2	17:16	R/W	‘h0	Controls the baud rate used by the transmitter. 00 – divide by 31 01 – divide by 32 10 – divide by 39 (For T=14 => F=624= BAUDDIV1x BAUDDIV2 =16 x 39)

BAUDDIV1	15:8	R/W	‘h0b	The internal clock is divided by the 8 bit (BAUDDIV2 + 1) and BAUDDIV2 to determine the baud clock.
PRE_CLKDIV	7:0	R/W	‘h07	Controls the frequency of the etu and the SC_CLK I/O pin. Divided by system clock rate (27 MHz).

Module::MIS	Register::SC0_CR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BE04

Name	Bits	Read/ Write	Reset State	Comments
FIFO_RST	31	W	‘b0	Smartcard FIFO Soft Reset, write “1” clears all RX FIFO
RST	30	W	‘b0	Smartcard Reset, write “1” to reset smartcard
SCEN	29	W	‘b0	Smartcard Enable bit, write “1” to enable smartcard function
TX_GO	28	W	‘b0	Write “1” to start sending the contents of the transmit buffer until the transmit buffer is empty.
AUTO_ATR	27	W	‘b1	When this bit is set, the next character received is interpreted as the initial character (TS) and the convention is automatically set. This bit is set before the first character of the Answer-to-Reset is received. This bit is ignored after the reception of the TS character unless RST bit is set.
CONV	26	W	‘b1	Determines the convention if AUTO_ATR is NOT used to automatically set the convention. 0=Direct Convention 1=Inverse Convention
CLK_STOP	25	W	‘b0	Indicates which electrical state is preferred on CLK when the clock is stopped, if the card supports clock stop mechanism. 0=state low 1=state high
PS	24	R/W	‘b1	Parity select 0: no parity 1: with parity (T=14 with no parity)
Rvd	23:0	-	-	Reserved

Module::MIS	Register::SC0_PCR	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BE08

Name	Bits	Read/ Write	Reset State	Comments
TXGRDT	31:24	R/W	‘h00	Transmit guard time. It determines the number of additional etus automatically inserted between transmit characters. A guard time of 2 etus is the default (TXGRDT=0). 0~254 : Extra Guard time = 12 + TXGRDT 255: when T =0, Extra Guard time = 12 etu when T=1, Extra Guard time = 11 etu
CWI	23:20	R/W	‘h0	Character Waiting Integer. This field determines the CWT which is the number of etus to allow between characters in a receive block. When the number of etus between received characters is greater than the CWT then IF_SCIRSR [CWT_INT] is set and the smart card interrupt is asserted if enabled. The character waiting time is encoded as following equation: CWT=(2^{CWI} + 11) etu
BWI	19:16	R/W	‘h0	Block Waiting Integer. This field determines the Block Waiting Time (BWT) that is the maximum delay between the leading

				edge of the last character of the block received by the ICC and the leading edge of the first character of the next block sent by the ICC. BWT is used to detect an unresponsive card. This condition generates an interrupt if IF_SCIRER[BWT_EN] is set. BWT = 11 etu + 2^{BWI} * 960 * 372* D/F (Actually, BWT = 11 + 2^{BWI} * 1024 etu)
WWI	15:12	R/W	'h0	Work Waiting Integer. This field determines the Work Waiting Time (WWT) that is the maximum delay between the leading edge of any character sent by the ICC and the leading edge of the precious character (sent either by the ICC or IFD). WWT = (960*WWI)*D etu (Actually, WWT = 1024*2^{WWI} etu)
BGT	11:7	R/W	'h16	Block Guard Time. This field determines the BGT that is the minimum delay between the leading edge of two consecutive characters sent in opposite directions. The value written to the field is the number of etus. IF_SCIRSR[BGT_INT] is set if this minimum delay is violated. In addition, the smart card interrupt is asserted if IF_SCRIRER[BGT_EN] is enabled. The value for BGT is 22 etu for most applications. BGT must be programmed to a value greater than 0x0C
EDC_EN	6	R/W	'b0	Enables EDC (Error Detection Code) method for T=1 protocol. When set to 1, an EDC (CRC16 or LRC) is appended to the end of each transmit block and the EDC received is checked against the received data. Note that the EDC bytes are always loaded into the receive buffer regardless of the state of EDC_EN.
CRC	5	R/W	'b0	Selects the EDC method for T=1 protocol. EDC_EN bit enables this feature. 1: for two bytes CRC16 0: for one byte LRC.
PROTOCOL_T	4	R/W	'b0	Determines the protocol. 0 for T=0 protocol, 1 for T=1 protocol
T0RTY	3	R/W	'b1	For T=0 protocol, when enabled, characters are automatically retransmitted when the ICC indicates a parity error. The number of times any one character is retransmitted is determined by T0RTY_CNT.
T0RTY_CNT	2:0	R/W	'b000	Specify the number of transmit parity retries per character. This is feature is enabled if T0RTY is set to 1. When ICC requests more retries than T0RTY_CNT for a single character, it will stop sending any characters until TX_GO is set again. The IF_SCIRSR[TXP_INT] will always be set if enabled to allow the application to detect if too many parity retries are requested. 000: No limit to the number 001-111: Set the number of retries to 1-7. (For EMV, the number of retries is set to 3.

Module::MIS	Register::SC0_TXFIFO	Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BE0C

Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**
Rvd	31:9	-	-	-
tx_fifo_full	8	R	-	TX FIFO FULL
DAT	7:0	W	-	TX FIFO, 32 bytes depth. When this register is written, the contents are loaded into the transmit buffer. The contents of the

				transmit buffer are sent when IF_SCCR[TX_GO] is set to 1. Characters are automatically transmitted using the convention established during ATR, therefore, characters written to this register are always written in direct convention.

Module::MIS	Register::SC0_RXFIFO		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BE14
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**		
Rvd	31:8	-	-	-		
DAT	7:0	R	-	RX FIFO, 32 bytes depth. Receive characters from the receive buffer. Characters are always read from RXFIFO in direct convention.		

Module::MIS	Register::SC0_RXLENR		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BE18
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**		
Rvd	31:8	-	-	-		
RXLEN	7:0	R	-	This is the number of bytes currently in the receive buffer. If RXLEN>32 then an overflow condition has occurred. Overflow is registered in IF_SCSR[RX_FOVER].		

Note: SC0_RXLENR is RX/TX length register.

Module::MIS	Register::SC0_FCR		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BE1C
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**		
Rvd	31:2	-	-	-		
RXFLOW	1	W	‘b0	If FLOW_EN is enabled and RXFLOW set to 1, SC_IO is pulled low at 11.5 etu into the next receive character, unless the next receive character is due to a retransmission request (In this case, SC_IO is pulled low in the receive character after the successful retransmission). SC_IO remains low until RXFLOW is written back to 0.		
FLOW_EN	0	W	‘b0	When set to 1, SC_IO is tested after the TGUARD time expires (at 12 etu or later) by the transmitter to determine if flow control is requested. If SC_IO is low when tested, then the transmitter waits until SC_IO is high again. In the receiver, the state of RXFLOW is tested at 11.5 etu. If high, then SC_IO is pulled low until RFLOW is set to 0..		

Module::MIS	Register::SC0_IRSR		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BE20
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**		
Rvd	31:17	-	-	-		
PRES	16	R	‘b0	Smartcard Present bit. 1: present 0: absent		
CPRES_INT	15	R/W	‘b0	Card Presence Interrupt Status. This bit is set when card is insert to or remove from the hardware interface.		
TX_FLOW_INT	14	R/W	‘b0	When set, this bit indicates that the transmitter has detected a flow control condition form the ICC’s receiver and is waiting		

				until it is released. When released (after SC_IO is pulled high by the ICC) this bit returns to zero and the transmitter will continue sending any characters in the transmit buffer.
TXP_INT	13	R/W	‘b0	Transmit Parity Interrupt Status. This bit is set when the ICC requests a retransmit due to parity error in the last transmitted character. This is applicable in T=0 when characters are resent when the ICC detects a parity error and requests a re-send. If IF_SCIRER[TXP_EN]=1, then this condition also causes a smart card interrupt.
TXDONE_INT	12	R/W	‘b0	Transmit Done Interrupt Status. Set high when the transmitter has finished sending one byte in SCTXFIFO.
TXEMPTY_INT	11	R/W	‘b0	Transmit Empty Interrupt Status. Set high when the transmitter has finished sending the number of bytes in SCTXFIFO.
EDCERR_INT	10	R/W	‘b0	Error Detection Code Error Interrupt Status. Indicates that the received block has bit errors because the calculated CRC16 or LRC failed.
RX_FOVER	9	R/W	‘b0	RX FIFO Overflow. This is an error condition and is cleared by resetting the buffer using IF_SCCR[FIFO_RST]. In addition, the SC should also be reset using IF_SCCR[RST].
RXP_INT	8	R/W	‘b0	Receive Parity Interrupt Status. This bit is set immediately when a parity error is detected on a received character.
ATRS_INT	7	R/W	‘b0	Answer To Reset Start Interrupt Status. This bit is set when receiving first byte (TS) of the ATR.
BGT_INT	6	R/W	‘b0	Block Guard Time Interrupt Status. This bit is set when the BGT is violated. The BGT is applicable for T=1 protocol and is used with the BWT to define a window for when the ICC can send a response to the IFD.
CWT_INT	5	R/W	‘b0	Character Waiting Time Out Interrupt Status. This is used in T=1 application can indicate the ICC is not functioning properly or that there is an error in the length of a lock.
RLEN_INT	4	R/W	‘b0	Receive Length Error Interrupt Status. This bit is set when one of two conditions occurs: An extra character is received in a T=1 block, or when the receive buffer overflows.
WWT_INT	3	R/W	‘b0	Work Waiting Time Interrupt Status.
BWT_INT	2	R/W	‘b0	Block Waiting Time Interrupt Status.
RCV_INT	1	R/W	‘b0	Receive Character Interrupt Status. This bit is set when the receive buffer goes not empty and has at least one character.
DRDY_INT	0	R/W	‘b0	Receive Ready Interrupt Status. Indicates a block has been received and is ready in the receive buffer. SCRXLENR indicates the number of bytes in the message. The hardware asserts IF_SCSR[RX_DR] when the calculated block size is equal to the number of bytes in the buffer. The hardware calculates the block size by extracting the LEN field from a T=1 message and compensating for the prologue and epilogue fields.

Module::MIS	Register::SC0_IRER		Set::1	ATTR::sfdf	Type::SR	ADDR::0x9801_BE24
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**		
Rvd	31:16	-	-			
CPRES_EN	15	R/W	‘b0	Card Presence Interrupt Enable.		
TXFLOW_INT_ EN	14	R/W	‘b0	Transmit Flow Detect Interrupt Enable.		
TXP_EN	13	R/W	‘b0	Transmit Parity Interrupt Enable.		

TXDONE_EN	12	R/W	‘b0	Transmit Done Interrupt Enable.
TXEMPTY_EN	11	R/W	‘b0	Transmit Empty Interrupt Enable.
EDCERR_EN	10	R/W	‘b0	Error Detection Code Error Interrupt Enable.
RX_FOVER_EN	9	R/W	‘b0	RX FIFO Overflow Interrupt Enable.
RXP_EN	8	R/W	‘b0	Receive Parity Interrupt Enable.
ATRS_EN	7	R/W	‘b0	Answer To Reset Start Interrupt Enable.
BGT_EN	6	R/W	‘b0	Block Guard Time Interrupt Enable.
CWT_EN	5	R/W	‘b0	Character Waiting Time Out Interrupt Enable.
RLEN_EN	4	R/W	‘b0	Receive Length Error Interrupt Enable.
WWT_EN	3	R/W	‘b0	Work Waiting Time Interrupt Enable.
BWT_EN	2	R/W	‘b0	Block Waiting Time Interrupt Enable.
RCV_EN	1	R/W	‘b0	Receive Character Interrupt Enable.
DRDY_EN	0	R/W	‘b0	Receive Ready Interrupt Enable.