

SC14452

VoIP processor with integrated cordless telephony interface

General Description

The SC14452 digital CMOS IC is an optimized VoIP processor with an integrated 10/100 Mbit ethernet MAC and DECT interface targeted at VoIP desktop phones, ATAs and IP DECT basestations. The SC14452 is able to handle all VoIP applications and baseband processing within the DECT & DECT 6.0 (1.9 GHz) ISM band (2.4 GHz, 5.8 GHz).

The design combines low power and high performance with an 82.944 MHz CR16C+™ and dual 82.944 MHz Gen2DSPs capable of executing user defined DSP programs. A cryptography hardware accelerator block is integrated.

Static and dynamic memories are supported including Pseudo Static RAM (PSRAM). A SPI flash controller for SPI flash memories with four I/O pins is built-in. Programmable Pin Assignment allows flexible PCB layout.

Features

- Complies with DECT ETS 300 175-2,3 & 8 and DECT 6.0.
- 10.368 MHz or 12.288 MHz xtal with digital controlled oscillator with on-chip 165.888 MHz and 48/50 MHz PLLs.
- Processing power
 - 82.944 MHz 16 bit CompactRISC™ CR16C+ with instruction Cache and 4-ch DMA controller.
 - Dual 82.944 MHz Gen2DSPs for Acoustic echo cancelling, voice recognition, caller-id, Midi, G.711, G.726, G.722, G.729, iLBC, T.38 and user programmable algorithms like G723.1, MP3.
 - Dedicated Instruction Processor (DIP) with new 32 bits CRC and double slot prot.B-field for G722.
- Memories
 - Shared RAM1+2/ROM1+2: 16k+20k/ 2k+64k byte
 - Non shared RAM/iCache 24k+8k byte.

- Gen2DSP uCode RAM1+2: 20k+16k
uCode ROM1+2 :32k+112k
- External memory controller for SDRAM, SRAM, FLASH and Async/page PSRAM.
- Crypto engine capable of AES/DES/3DES
- Power management
 - 1.8Volt operating voltage with 1.8-3.45V I/O pads and separate 1.8-3.45V toward RF transceiver.
 - DC-DC converter used as buck (step-down) converter
 - On chip temperature sensor.
- Analog and Audio Interfaces
 - 8, 16, 32 kHz 16-bit linear audio CODEC.
 - Analog Front End to differential and single ended microphones and 28 ohm loudspeaker.
 - High efficiency 0.5W@2.5V, 1.152W@3.6V Li-Ion (4 ohm) switching amplifier.
 - 10bit ADC with 8/16 kHz sample rate.
 - Ringer input and **external** opamp for caller-id.
- Digital interfaces
 - General purpose I/O ports with Programmable Peripheral Assignment.
 - Microwire™ interface to RF transceiver.
 - Serial Debug interface, Nexus Class-1 compliant.
 - UART Full duplex 9600-230.4 kbaud.
 - Dual SPI™ interface 20.736 MHz (Master/Slave).
 - Quad SPI interface for FLASH up to 80MHz.
 - Dual ACCESS bus 100k, 400 kHz, 1.152 MHz
 - PCM Interface 4.608, 2.304, 1.152 MHz master, 2048 kHz slave
 - Ethernet MAC 10/100 Mbps supporting MII/RMII.
 - Two general purpose timers and watch dog timer.
 - e-LQFP-128 package with exposed pad.

Note 1: Microwire™, CompactRISC™ are trademarks of National Semiconductor. SPI™ is trademark of Motorola.

Note 2: End products using G.729 require license from Licensing Service Provider Sipro Lab Telecom (www.sipro.com)

System Diagram

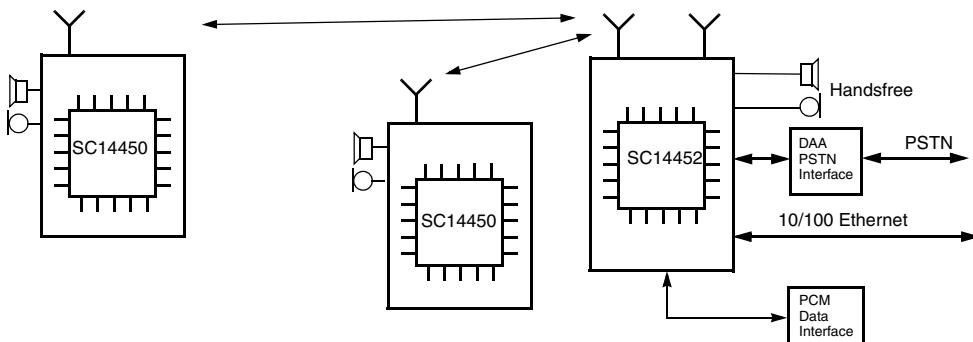


Table of Contents

1.0 Connection diagram.	5
1.1 PIN TYPE DEFINITIONS	13
1.2 PROGRAMMABLE PIN ASSIGNMENT	15
1.2.1 SDRAM control signal selection	15
2.0 Introduction	20
3.0 Functional overview	21
4.0 System Clock generation	22
4.1 CRYSTAL OSCILLATOR	22
4.2 OSCILLATOR FREQUENCY DEVIATION	22
4.3 FREQUENCY CONTROL	22
4.4 PHASE LOCKED LOOPS (PLL)	23
4.5 SYSTEM CLOCK GENERATION	24
4.6 LOW POWER MODE	30
4.6.1 Clock Gating	30
4.6.2 Extra Division.....	30
4.7 SWITCHING PROCEDURES	31
4.7.1 LDO_XTAL switching On/Off	31
4.7.2 PLL1/PLL2 switching On/Off	31
4.7.3 Switching between PLL1/XTAL	31
4.7.4 Programming different frequencies at PLL1	31
5.0 Power Management	32
5.1 DC-DC CONVERTER	32
5.2 LOW DROPOUT REGULATORS	32
6.0 ON / OFF control.	33
6.1 BANDGAP REFERENCE	33
7.0 Power-On and system Reset.	34
7.1 POWER ON RESET	34
7.2 SYSTEM RESET	34
7.3 SOFTWARE RESET	35
7.4 PROGRAM COUNTER AFTER RESET	35
8.0 Boot Program	36
8.1 BOOT FROM INTERNAL RAM	36
8.2 BOOT FROM JTAG	36
8.3 BOOT FROM UART	36
8.4 BOOT FROM SPI OR QSPI	37
8.5 BOOT FROM EXTERNAL MEMORY	38
9.0 JTAG-SDI	41
10.0 DIP	42
10.1 DIP ON/OFF CONTROL	43
10.2 DIP SEQUENCER RAM	43
10.3 DIP INTERRUPT GENERATION AND POWER CONTROL	43
10.4 INTERFACING TO GEN2DSPS	43
10.5 MINIMUM DELAY	43
10.6 MUTING	43
10.7 SLOT CONTROL INFORMATION	44
10.8 SLOT STATUS INFORMATION	44
10.9 SLOT SYNCHRONISATION	45
10.10 S-PATTERN CORRELATOR	45
10.11 DCS BLOCK ENCRYPTION	45
10.12 SCRAMBLER	45
10.13 R-CRC, X-CRC AND Z FIELD	45
10.14 ADPCM SUPPORT	46
10.15 DIP COMMANDS	46
11.0 The Radio Front-End Interface	49
11.1 MICROWIRE™ INTERFACE	49
11.2 REFERENCE CLOCK OUTPUTS	51
11.3 PROGRAMMABLE PD CONTROL PINS	51
11.4 PD1 SWITCHING	51
11.5 RSSI PEAKHOLD ADC	52
11.6 FAST ANTENNA DIVERSITY (FAD)	52
11.7 MODULATED OUTPUT	53
11.8 DEMODULATED DATA INPUT	53
12.0 Gen2DSP	55
12.1 START-UP AND INTERRUPTS	56
12.2 POWER SAVING	57
12.3 PCM SLAVE SYNCHRONISATION	57
12.4 INTERRUPTS	57
12.5 NESTING INTERRUPT	58
12.6 OVERFLOW	58
12.7 REGISTERS	59
12.8 RAMIO REGISTERS	59
12.9 SHARED ROM/ROM	59
12.10 MICRO CODE ROM/RAM	59
12.11 SUBROUTINE CONTROL PROGRAM (SCP)	60
12.12 ECZ, ECZ2 USAGE AND PORT MAPPING	63
13.0 PCM interface	64
13.1 INTERFACE SIGNALS	64
13.2 CHANNEL ACCESS	64
13.3 CLOCK STRETCHING/SHORTENING	65
13.4 DATA FORMATS	66

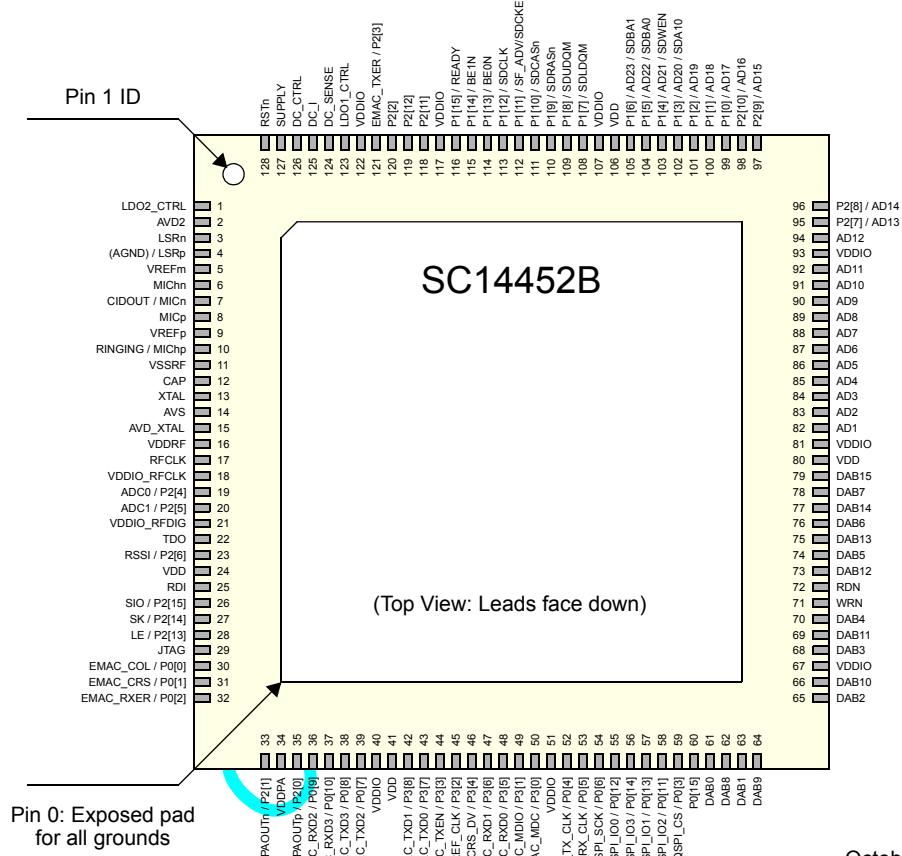
Table of Contents

13.5 EXTERNAL SYNCHRONISATION	67	20.3 FREEZING DMA CHANNELS	97
14.0 CODEC and Analog frontend	69	20.4 BURST MODE	97
14.1 LOUDSPEAKER CONNECTION	69	20.5 MEMORY INITIALIZATION MODE	97
14.2 MICROPHONE CONNECTION	69	20.6 DMA LENGTH REGISTERS	97
14.3 HEADSET CONNECTION	69	20.7 DMA INTERRUPT REGISTERS	97
14.4 CODEC OFFSET COMPENSATION	69	21.0 CCU	98
14.5 FILTERING	70	22.0 I/O ports P0, P1, P2, P3	99
14.6 PROGRAMMABLE CLOCKS	70	23.0 Keyboard Interface	100
14.7 CALLER-ID OPAMP	70	24.0 UART	102
14.8 INTEGRATED HEADSET DETECTION	72	24.1 IRDA MODE	103
15.0 CLASSD Amplifier	77	25.0 ACCESS BUS	104
15.1 DIGITAL AUDIO MODE PATH SELECTION	77	26.0 SPI+ Controller	107
15.2 ANTI-PLOP CIRCUIT	77	26.1 OPERATION WITHOUT FIFOs	107
15.3 CLIP DETECTION	77	26.2 FIFO OPERATION	109
15.3.1 Buzzer mode	77	27.0 ADC	112
15.4 DIGITAL OUTPUTS	77	28.0 Temperature Sensor	115
15.5 POWER SAVING	77	29.0 Timers	116
15.6 NOISE REDUCTION CIRCUIT	77	29.1 TIMER0	116
16.0 Cache Controller	79	29.2 TIMER1	118
16.1 CACHE INITIALIZATION	79	29.3 WATCHDOG TIMER	119
16.2 THE INSTRUCTION CACHE (IC)	80	29.4 CLOCK 100 TIMER	121
16.3 DATA CACHE (DC)	80	29.5 CAPTURE TIMERS/COUNTERS	122
16.4 CACHE LOCK	80	30.0 Ethernet MAC 10/100	124
16.4.1 Lock per cache	80	30.1 FEATURES	124
16.4.2 Lock per line	80	30.1.1 Ethernet MAC core	124
16.5 CACHE ORGANISATION	80	30.1.2 DMA block	124
16.6 RAM ARBITER	82	30.1.3 Transaction Layer - MTL	124
17.0 CR16C+	83	30.1.4 Monitoring, Test and Debugging Support	
18.0 APB Bridge	83	125	
19.0 EBI External Bus Interface	84	30.2 INTERNAL ARCHITECTURE	125
19.1 CHIP SELECTION	85	30.3 DMA CONTROLLER	125
19.2 IO EXPANDER WITH ACS3	86	30.3.1 Initialization	126
19.3 STATIC MEMORY CONTROLLER	86	30.3.2 Transmission	127
19.3.1 Memory types	86	30.3.3 Reception	130
19.3.2 Control registers	87	30.3.4 Interrupts	132
19.4 SDR-SDRAM CONTROLLER	89	30.4 MAC TRANSACTION LAYER (MTL)	132
19.5 SDRAM INITIALIZATION EXAMPLES	92	30.4.1 Transmit Path	132
19.6 BUS SHARING	93	30.4.2 Receive Path	134
20.0 DMA Controller	95	30.5 MAC CORE	135
20.1 DMA CHANNEL OPERATION	96	30.5.1 Transmission	135
20.2 DMA CHANNEL ARBITRATION	97		

Table of Contents

30.5.2 Reception.....	137	32.3.4 Received data	161
30.6 MAC MANAGEMENT COUNTERS (MMC).....	139	33.0 Generic Programming Registers	163
30.7 DESCRIPTORS	139	33.1 GENERAL DESCRIPTION	163
30.7.1 Descriptor Formats	139	34.0 Interrupt Control Unit (ICU).....	164
30.7.2 Receiver Descriptor.....	139	35.0 Memory map overview.....	166
30.7.3 Transmit Descriptor	144	35.1 GEN2DSP1, GEN2DSP2 ROM TABLES AND REGISTER FILES.....	168
30.8 STATION MANAGEMENT AGENT (SMA)	148	36.0 Memory map Detailed	171
30.8.1 Functions.....	148	37.0 Registers description detailed	186
30.8.2 MII Management Write Operation ...	148	38.0 Specifications.....	286
30.8.3 MII Management Read Operation ...	149	38.1 ELECTRICAL CHARACTERISTICS.....	287
30.9 REDUCED MEDIA INDEPENDENT INTER- FACE (RMII)	149	38.2 ANALOG FRONTEND SPECIFICATIONS .	299
30.9.1 Transmit/Receive Bit Ordering	149	38.3 CODEC SPECIFICATIONS	306
30.10 INTERRUPTS FROM THE MAC CORE ..	150	39.0 Timing Specifications	310
30.11 DESCRIPTION OF THE REGISTER FILES ..	151	39.1 MICROWIRE TIMING DIAGRAMS	312
30.11.1 MAC Register File	151	39.2 UART IRDA TIMING DIAGRAMS	314
30.11.2 MMC Register File.....	152	39.3 ACCESS BUS TIMING DIAGRAM	315
30.11.3 DMA Register File	153	39.4 PCM INTERFACE TIMING DIAGRAM ..	316
30.12 ACCESSING EMAC FIFOs THROUGH THE AHB BUS (FOR TESTING ONLY)	155	39.5 SPI BUS TIMING DIAGRAM.....	318
30.12.1 Write access	155	39.6 EBI STATIC MEMORY TIMING DIAGRAMS	319
30.12.2 Read access	155	39.7 EBI SDRAM TIMING DIAGRAMS.....	323
31.0 Crypto	156	40.0 Package information	325
31.1 GENERAL DESCRIPTION	156	40.1 MOISTURE SENSITIVITY LEVEL (MSL) ..	325
31.2 FEATURES	156	40.2 SOLDERING INFORMATION.....	325
31.3 DESCRIPTION	156	40.3 FLOOR AND SHELF LIFE ADVISORY....	325
31.4 ARCHITECTURE	156	40.4 PACKAGE OUTLINES	326
31.4.1 Crypto	156		
31.4.2 AES	157		
31.4.3 DES/Triple DES	158		
31.4.4 Modes	158		
31.5 PROGRAMMING.....	158		
32.0 Quad SPI Controller.....	159		
32.1 INTERFACE	159		
32.2 SPECIFICATION.....	160		
32.2.1 SPI modes	160		
32.2.2 Access modes	160		
32.2.3 Endianess	160		
32.2.4 Clock relations	160		
32.3 PROGRAMMING.....	160		
32.3.1 Auto Mode	160		
32.3.2 Manual Mode	161		
32.3.3 Clock selection.....	161		

1.0 Connection diagram



October 7, 2008

Order number: SC14452BNF128VPN (e-LQFP128 package)

Note 1: Refer to Order_Number_and_Package_Marking.pdf on www.sitelsemi.com for a definition of the order numbers and package marking. Lead free devices have always an 'N' in the last three digits of the order number.

Table 1: Pin Description (For multiplexing refer to Table 3)

PIN NAME	TYPE	Drive (mA)	Reset state (Note 2)	DESCRIPTION
Ports				
P0[15-0] or PPA	DIO	16	I-PUD	INPUT/OUTPUT General purpose I/O port bit or Programmable Peripheral Assignment nodes with selectable pull up/down resistor. Pull-up/pull-down enabled according P0_yy_MODE_REG during and after reset.
P1[15-0] or PPA	DIO	16	I-PUD	INPUT/OUTPUT General purpose I/O port bit or Programmable Peripheral Assignment nodes with selectable pull up/down resistor. Pull-up/pull-down enabled according P1_yy_MODE_REG during and after reset.
P2[12-0] or PPA	DIO	16	I-PD	INPUT/OUTPUT General purpose I/O port bit or Programmable Peripheral Assignment nodes with selectable pull up/down resistor. Pull-up/pull-down enabled according P2_yy_MODE_REG during and after reset. Pins P2[4] and P2[5] are supplied by VDDIO_RFDIG instead of VDDIO.
P3[8-0] or PPA	DIO	16	I-PD	INPUT/OUTPUT General purpose I/O port bit or Programmable Peripheral Assignment nodes with selectable pull up/down resistor. Pull-up/pull-down enabled according P3_yy_MODE_REG during and after reset.
Debug interface				
JTAG	DIOD	16	I-PU	INPUT/OUTPUT. SDI+ one wire interface with open drain and hot insertion. If pulled low after reset, the SDI with one wire interface is enabled and consumes power. Resetting is possible if this pin is high during reset or by writing to TEST_ENV_REG[ENV_SDI] = '1'. An external 1k pull-up resistor must be present on the board for debugging. (see also Figure 16)
System Bus (All device configurations)				
AD(23-13)	DO	16	I-PUD/O-0	OUTPUT. Address bus 23-13 for Static memory and SDRAM memory. Addresses point to byte address. Depending on the boot source one or more address lines are set to output. For more information see chapter "Boot Program" on page 36.
AD(12-4)	DO	16	O-0	OUTPUT. Address bus 12-4 for Static memory and SDRAM.
AD(3-1)	DIOPUS	16	I-PU (if RSTn= 0) O-0 (if chip is out of reset)	INPUT/ OUTPUT. Address bus 3-1 for Static memory and SDRAM. If RSTn =0 the input level of these pins determines the LDO1 and DC_VOUT reset values. '1': No external resistor connected. Internal pull-up resistor to VDD gives a '1', '0': External <100k pull-down resistor to VSS gives a '0'. The pin level is latched on the rising edge of the RSTn. Refer to SUPPLY_CTRL_REG for LDO1 and DC_VOUT voltages.
BEn(1-0)	DO	16	I-PUD/O-1	OUTPUT. Upper/Lower byte enable for Static memory. Also used for PSRAM PS_UDn, PS_LDn.
DAB(15-0)	DIO	16	O-0	INPUT/OUTPUT. Data bus 15-0 for static memory and SDRAM DQ(15-0).
RDn	DO	16	O-1	OUTPUT. Active low bus read for static memory.
WRn	DO	16	O-1	OUTPUT. Active low bus write for static memory.

Table 1: Pin Description (For multiplexing refer to Table 3)

PIN NAME	TYPE	Drive (mA)	Reset state (Note 2)	DESCRIPTION
READY	DIOPU	16	I-PD	INPUT with pull-up resistor. External static memory Peripheral Ready. If EBI_SMTMGR_SETx_REG[READY_MODE] is set to '1', the memory controller inserts wait cycles as long as this pin is low.
Interrupts				
INT8-INT6	DI		I-PUD	INPUT. 3 general purpose interrupts with rising/falling edge or high/low levels. All inputs generate a KEYB_INT.
INT5-INT0	DI		I-PUD	INPUT. 6 Keyboard input interrupts requests with or without debounce timer and programmable rising/falling edges. All inputs generate a KEYB_INT.
Synchronous FLASH				
SF_ADV	DO	16	I-PD/O-1	OUTPUT. Address Valid
SDCLK	DO	16	I-PD/O-0	OUTPUT. CLK 82.944 MHz Max. This clock can be switched off if only asynchronous mode us used.
SDRAM Interface				
SDCLK	DO	16	I-PD/O-0	OUTPUT. SDRAM System bus Clock.
SDCKE	DO	16	I-PD/O-1	OUTPUT. SDRAM Clock enable.
SDCASN	DO	16	I-PD/O-1	OUTPUT. SDRAM Column Access Strobe
SDRASN	DO	16	I-PD/O-1	OUTPUT. SDRAM Row Access Strobe
SDLDQM	DO	16	I-PD/O-1	OUTPUT. SDRAM Lower Byte input/output mask.
SDUDQM	DO	16	I-PD/O-1	OUTPUT. SDRAM Upper Byte input/output mask.
SDBA[1-0]	DO	16	I-PD/O-0	OUTPUT. SDRAM Bank 0-3 address selects.
SDWEN	DO	16	I-PD/O-1	OUTPUT. SDRAM write enable (not)
SDA10	DO	16	I-PD/O-0	OUTPUT. SDRAM Address 10
ACCESS bus Interface 1,2				
SDA	DIO/DIOD	16	I-PUD	INPUT/OUTPUT. ACCESS bus Data with programmable Push-pull or open drain
SCL	DIO/DIOD	16	I-PUD	INPUT/OUTPUT. ACCESS bus. Clock with programmable Push-pull or open drain. In open drain mode, SCL is monitored to support bit stretching by a slave.
SPI Interface 1,2				
SPI_DOUT	DO	16	I-PUD/O-0	OUTPUT. Data output
SPI_DIN	DI	-	I-PUD	INPUT. Data input
SPI_CLK	DIO	16	I-PUD/O-0	INPUT/OUTPUT. Clock (SPI1 master/slave, SPI2 Master only)
SPI_EN	DI		I-PUD	INPUT. Clock enable (SPI1 only)
Quad SPI Interface				
QSPIC_SCK	DO	16	I-PD/O-1	OUTPUT. Clock
QSPIC_CS	DO	16	I-PD/O-1	OUTPUT. Chip Select
QSPIC_IO0	DIO	16	I-PD	INPUT/OUTPUT. Single SPI mode: Data output (OUTPUT). Dual and Quad SPI modes: Data IO0 (INPUT/OUTPUT).
QSPIC_IO1	DIO	16	I-PD	INPUT/OUTPUT. Single SPI mode: Data input (INPUT) Dual and Quad SPI modes: Data IO1 (INPUT/OUTPUT).

Table 1: Pin Description (For multiplexing refer to Table 3)

PIN NAME	TYPE	Drive (mA)	Reset state (Note 2)	DESCRIPTION
QSPIC_IO2	DIO	16	I-PD	INPUT/OUTPUT. Single and Dual SPI mode: Write Protect. Quad SPI mode: Data IO2 (INPUT/OUTPUT).
QSPIC_IO3	DIO	16	I-PD	INPUT/OUTPUT. Single and Dual SPI mode: Hold. Quad SPI mode: Data IO3 (INPUT/OUTPUT).
EMAC (R)MII interface				
EMAC_MDC	DO	16	O	OUTPUT. Management Data Clock The EMAC provides timing reference for the EMAC_MDIO signal on the MII through this aperiodic clock. The maximum frequency of this clock is 2.5 MHz.
EMAC_MDIO	DIO	16	I	INPUT/OUTPUT. Management Data In/Out The EMAC uses this signal to transfer control and data information from/to the PHY.
EMAC_TXEN	DO	16	O-0	OUTPUT. PHY Transmit Data Enable. When high, indicates that valid data is being transmitted on the phy_tx bus. MII mode: Synchronous to EMAC_TX_CLK. RMII mode: Synchronous to EMAC_REF_CLK
EMAC_CRS_DV	DI	-	I	INPUT. PHY Receive Data Valid. MII: When high, indicates that data on the EMAC_RXD[3:0] bus is valid. It remains asserted continuously from the first recovered nibble of the frame through the final recovered nibble. Synchronous to EMAC_RX_CLK. RMII: Contains the crs and data valid information of the receive interface. Synchronous to EMAC_REF_CLK.
EMAC_COL	DI	-	I-PU	INPUT. PHY Collision This signal, valid only in MII mode, is asserted by the PHY when a collision is detected on the medium. This signal is not synchronous to any clock.
EMAC_CRS	DI	-	I-PU	INPUT. PHY CRS This signal, valid only in MII mode, is asserted by the PHY when either the transmit or receive medium is not idle. The PHY deasserts this signal when both transmit and receive medium are idle. This signal is not synchronous to any clock.
EMAC_RXER	DI	-	I-PD	INPUT. PHY Receive Error MII: When high, indicates an error or carrier extension in the received frame on the EMAC_RXD bus. Synchronous to EMAC_RX_CLK. RMII: Not used
EMAC_TXER	DO	16	I-PD/O-0	OUTPUT. PHY Transmit Error MII: When high, indicates a transmit error or carrier extension on the EMAC_TxD bus. Synchronous to EMAC_TX_CLK. RMII: Not used
EMAC_TxD[3:0]	DO	16	I-PD	OUTPUT. PHY Transmit Data MII: Bits [3:0] provide the MII transmit data nibble. The validity of the data is qualified with EMAC_TXEN and EMAC_TXER. Synchronous to EMAC_TX_CLK. RMII: Bits [1:0] provide the RMII transmit data. The validity of the data is qualified with EMAC_TXEN. Synchronous to EMAC_REF_CLK. Unused bits [3:2] in the RMII interface configuration are tied to low.

Table 1: Pin Description (For multiplexing refer to Table 3)

PIN NAME	TYPE	Drive (mA)	Reset state (Note 2)	DESCRIPTION
EMAC_RXD[3:0]	DI	-	I-PD	INPUT. PHY Receive Data MII: Bits [3:0] provide the MII receive data nibble. The validity of the data is qualified with EMAC_CRS_DV and EMAC_RXER. Synchronous to EMAC_RX_CLK. RMII: Bits [1:0] provide the RMII receive data. The validity of the data is qualified with EMAC_CRS_DV. Synchronous to EMAC_REF_CLK.
EMAC_TX_CLK	DI	-	I-PD	INPUT. MII Transmission Clock/ RMII Clock MII: This is the transmission clock. All the MII transmission signals generated by the EMAC are synchronous to this clock. RMII: not used.
EMAC_RX_CLK	DI	-	I-PD	INPUT. Reception Clock MII: This is the reception clock provided by the external PHY for MII. All the MII receive signals received by the EMAC are synchronous to this clock. RMII: not used
EMAC_REF_CLK	DIO	16	-	INPUT/OUTPUT. 50 MHz input/output reference clock for RMII PHY interface.
UART Interface				
UTX	DO	16	I-PUD/O-1	OUTPUT. UART data.
URX	DI	16	I-PUD	INPUT. UART data
PCM Interface				
PCM_DO	DO	16	I-PUD/O-1	OUTPUT. Data output (PCM)
PCM_DI	DI	-	I-PUD	INPUT. Data input (PCM)
PCM_CLK	DIO	16	I-PUD/O-1	INPUT/OUTPUT. Bus Clock. (PCM)
PCM_FSC	DIO	16	I-PUD	INPUT/OUTPUT. 8, 16, 32 kHz Frame sync (PCM)
RF interface				
RFCLK	DO	16	I-PD	OUTPUT. Digital single ended buffered inverted xtal clock output. Supplied from VDDIO_RFDIG or LDO_RFCLK output VDDIO_RFCLK.
LE	DIO	16	I-PD	INPUT/OUTPUT. Load Enable output. (input for test purposes) Supplied by VDDIO_RFDIG.
SK	DIO	16	O-0	INPUT/ OUTPUT. Microwire clock with programmable active edge. Input in test mode. Supplied by VDDIO_RFDIG.
SIO	DIOSPD	16	I-PD	INPUT/OUTPUT. Microwire bidirectional data I/O with weak pull down resistor. Pull-down is automatically disabled if output. Supplied by VDDIO_RFDIG.
RDI	DI	-	I (Note 3)	INPUT. Received Data from RF. Digital mode. Can be programmed to be inverted. Supplied by VDDIO_RFDIG.
TDO (ana)	AO	-		OUTPUT. Transmit BMC Data. Analog mode. Can be programmed to be inverted.
TDOD	DO	16	I_PUD/O-0	OUTPUT. Transmit BMC Data. Digital mode. Can be programmed to be inverted.

Table 1: Pin Description (For multiplexing refer to Table 3)

PIN NAME	TYPE	Drive (mA)	Reset state (Note 2)	DESCRIPTION
PD[5-1]	DIO-SPD	16	I-PUD	OUTPUT. DIP controlled Power Down pins. After PPA selection, the pins are set to output with the DIP <P_EN> instruction. If output, the pull down resistors are automatically disabled. PD1 can be programmed to automatically change level upon detection of the S field preamble. PD0 internally controls the RSSI peak hold ADC and is not available on any pin.
RSSI / P2[6] / ADC2	A1		I-PUD	INPUT. Received Signal Strength Indication to 6 bit peak-hold ADC
CLASSD power amplifier				
PAOUTp	A1	500	O	OUTPUT. CLASSD positive output to 4 ohm loudspeaker This pin can be configured as general purpose output, but this will result in an extra static DC current, so this is not recommended in portable applications. (see DC characteristics) (Note 5)
PAOUTn	A1	500	O	OUTPUT. CLASSD negative output to 4 ohm loudspeaker This pin can be configured as general purpose output, but this will result in an extra static DC current, so this is not recommended in portable applications. (see DC characteristics) (Note 5)
VDDPA	-		-	Supply voltage CLASSD audio amplifier
VSSPA		-	-	Ground for CLASSD power amplifier
Audio Codec				
LSRp or AGND	A1	-	O	OUTPUT. Positive loudspeaker output OUTPUT. Buffered analog ground if LSRP_MODE = 00.
LSRn or AGND	A1	-	O	OUTPUT. Negative loudspeaker output. OUTPUT. Buffered analog ground if LSRN_MODE = 00.
MIChp	A2	-	I	INPUT. Positive headset microphone input.
MIChn	A2	-	I	INPUT. Negative headset microphone input.
MICp	A2	-	I	INPUT. Positive microphone input
MICn	A2	-	I	INPUT. Negative microphone input. This pin is shared with CIDOUT input. If MICn is selected input to the microphone amplifier, CODEC_TONE_REG[CID_PD] must be set '1' to disable (default) the CIDOUT path to the ADC.
VREFp	A2	-	O	OUTPUT. Positive microphone reference
VREFm	A1	-	O	OUTPUT. Negative microphone reference. This pin must also be connected to VSS Ground, but make sure that the microphone ground is directly routed to VREFm (VREFm is the star point).
Caller-id opamps + RINGING				
RINGING / MIChp / ADC3	A1		I	ANALOG INPUT. ADC3 input to ADC input ANALOG INPUT. Ringer signal detection input to capture timers
CIDOUT	A1		I	ANALOG INPUT. An external Caller-id opamp is required. CODEC_TONE_REG[CID_PD] must be set '0' to enable the CIDOUT path to the ADC.
DC/DC converter				
DC_CTRL	A1	2	O-PD (fixed 100k pull-down)	OUTPUT. Switching clock for the DC/DC converter, this pad is supplied with SUPPLY.

Table 1: Pin Description (For multiplexing refer to Table 3)

PIN NAME	TYPE	Drive (mA)	Reset state (Note 2)	DESCRIPTION
DC_I	A1	-	I	ANALOG INPUT. Current sense input of DC/DC converter
DC_SENSE	A1	-	I	ANALOG INPUT. Voltage sense of DC/DC converter output
Miscellaneous				
CLK100	DO		I-PUD/O-0	OUTPUT. DIP 100 Hz output set by <CLK100> instruction
PWM[1-0]	DO	-	I-PUD/O-0	OUTPUT. Timer 0 PWM1 or PWM0 output
ECZ1	DO	-	I-PUD/O-0	OUTPUT. Gen2DSP1 or Gen2DSP2 output port set by DSP_ZCROSS1_OUT_REG[DSP_ZCROSS]
ECZ2	DO	-	I-PUD/O-0	OUTPUT. Gen2DSP1 or Gen2DSP 2output port set by DSP_ZCROSS2_OUT_REG[DSP_ZCROSS]
PLL2_CLK	DO	-	I-PUD/O-0	OUTPUT. 50/25 MHz PLL2 clock (1 ns jitter)
WTF_IN1	DO	-	I-PUD/O-0	OUTPUT. Gen2DSP1 enable signal used to monitor DSP load
WTF_IN2	DO	-	I-PUD/O-0	OUTPUT. Gen2DSP2 enable signal used to monitor DSP load
Power and supply				
LDO1_CTRL	A1	-	O-1	OUTPUT. IO voltage Regulator 1 control output. '1' at startup
LDO2_CTRL	A1	-	O-0	OUTPUT. Core voltage Regulator 2 control output.
RSTn	A5	8	I-PU (200k pull up)	INPUT/OUTPUT. Active low Reset input with Schmitt trigger input, open drain output and pull up resistor to VDD. Input may not exceed 2.0V
ADC0	A4		I-PUD/I	ANALOG INPUT. ADC0 input to ADC with programmable input protection (ADC0_PR_DIS) (Note 4)
ADC1	A4		I-PD/I	ANALOG INPUT. ADC1 input to ADC with programmable input protection (ADC1_PR_DIS) (Note 4)
RSSI / P2[6] / ADC2	A1		I-PD/I	ANALOG INPUT. ADC2 input to ADC. with programmable input protection (ADC2_PR_DIS) (Note 4)
AVD2		-	-	INPUT Analog supply voltage (1.8V) for CODEC AFE, DC/DC converter, ADC, Temperature sensor
AVS		-	-	Analog ground for AVD2.
SUPPLY	-	-	-	INPUT supply voltage.
VDD	-	-	-	INPUT Digital supply voltage 1.8V
VSS		-	-	Digital ground for VDD
VDDIO	-		-	INPUT Supply voltage (1.8 to 3.45V) for digital IOs
VDDIO_RFCLK	A6	-	-	OUTPUT of LDO_RFCLK Supply voltage for RFCLK pin. (1.6V, 2.3V, 3.1V). This pin must be decoupled with a 1uF ceramic capacitor if the LDO_RFCLK is used. (VDDIO_RFCLK has a separated VSS connected to the exposed pad.)
VDDIO_RFDIG	-	-	-	INPUT Supply voltage (1.8 to 3.45V) for RF digital IOs: LE, SIO, SK, RDI and LDO_RFCLK (See pad type A6)
VSSIO		-	-	Digital ground for IOs
VDDRF	A6	-	-	INPUT Analog supply voltage (1.8V) for LDO_XTAL, XTAL (XTAL_SUPPLY=1), PLL, RSSI, FAD, TDO gaussian shaper.
VSSRF		-	-	Ground for VDDRF
Xtal				
AVD_XTAL	A6		-	OUTPUT Supply voltage (1.6V) LDO for XTAL and ADC This pin must be decoupled with a 1uF ceramic capacitor.

Table 1: Pin Description (For multiplexing refer to Table 3)

PIN NAME	TYPE	Drive (mA)	Reset state (Note 2)	DESCRIPTION
AVS		-	-	Analog ground AVD_XTAL
CAP	A6	-	I	External capacitor for xtal tuning
XTAL	A6	-	I	INPUT. 9.216, 10.368, 12.288, 13.824 MHz crystal connection.
BXTAL	DO	-	I-PUD/O-0	OUTPUT. Digital buffer output of xtal oscillator for multiple baseband clocking or PLL clock divided by N for flexible clock output.

Note 2: Value before '/': Level and configuration during and after reset. When booting from external FLASH, all pins except AD23-AD13 are set to Pull-up by the booter, to avoid bus conflicts due to ACS1-4 mapped on pins which are active low.

Value after '/': Reset level of the peripheral block after PPA selection:

I-PUD = Input with pull-up or pull-down enabled depending on intended PPA pin selection.

I-PU = Pull-up resistor enabled, I-PD = Pull-down resistor enabled, I = input, Hi-Z= high impedance output

O-1 =output with 1= logic HIGH level, O-0= output with logic LOW level.

All digital inputs have Schmitt trigger inputs.

Note 3: Digital inputs without Pull-up or Pull-down resistors selected, may not be left unconnected in order to avoid static currents.

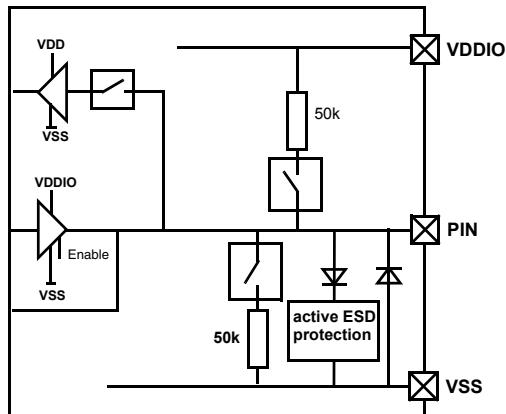
Note 4: For basestation applications, an input protection on all ADC inputs can be enabled with ADC_CTRL_REG[ADCx_PR_DIS] ='0'.

This affects the linearity of the ADC above 0.9V.

Note 5: In digital mode extra static VDDRF and VDDPA current will flow. So the digital mode is not recommended in portable applications.

1.1 PIN TYPE DEFINITIONS

Table 2: Pin type definitions



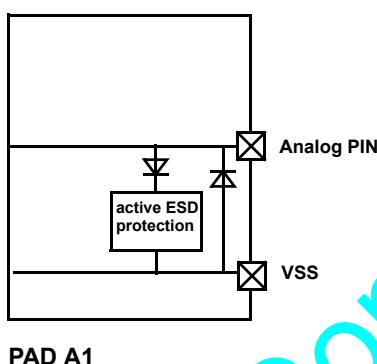
Digital / analog PADS

Digital/analog PAD IO Configurations:

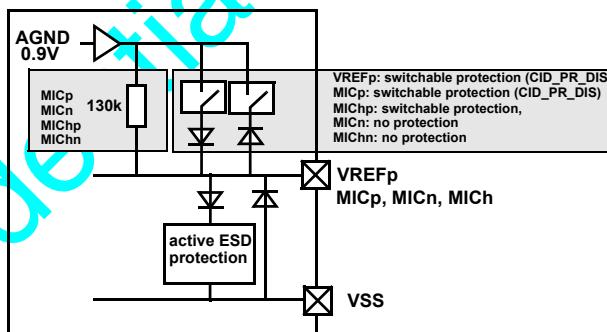
- DO: Digital Output
- DI: Digital Input
- DIO: Digital Input/Output
- DIOD: Digital Input/Output opendrain
- AI: Analog input
- AO: Analog Output
- AIO: Analog Input/Output

Pullup/pulldown extensions:

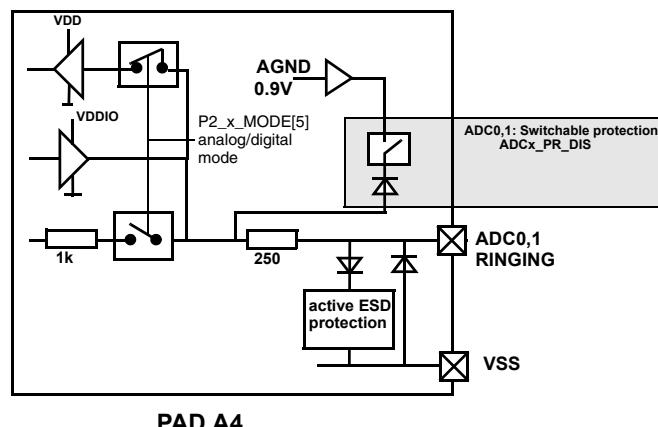
- PU: Fixed pull-up resistor
- PD: Fixed pull-down resistor
- SPU: Switchable pull-up resistor
- SPD: Switchable pull-down resistor



PAD A1



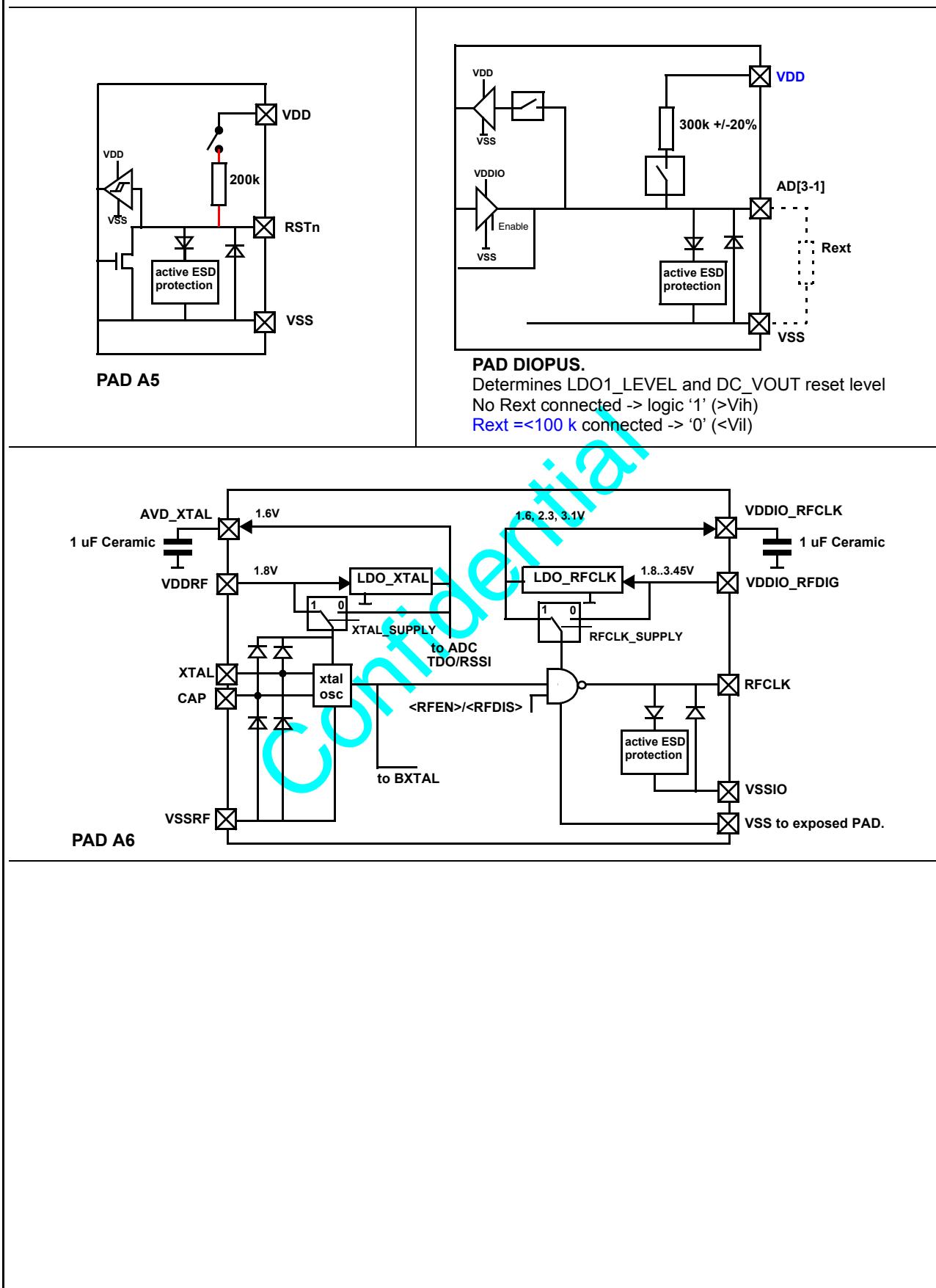
PAD A2



PAD A4

VoIP processor with integrated cordless telephony interface

Table 2: Pin type definitions



VoIP processor with integrated cordless telephony interface

1.2 PROGRAMMABLE PIN ASSIGNMENT

The SC14452 has a Programmable Pin Assignment (PPA) for secondary port functions. The peripheral I/Os can be freely mapped to a specific I/O port bit by writing Peripheral IO ID (PID) in Px_yy_MODE_REG. Refer to Detailed register Description for actual registers. Table 1 shows the implemented mapping matrix and PID values.

Analog functions have fixed pin assignment in order to limit interference with the digital domain. Fast bus signals also have a fixed location in order to deal with on-chip timing constraints.

General purpose Port mapping

Px_DATA_REG output has a fixed assignment to the corresponding Px[y] and is enabled with PID = 0x300

Reading Px_DATA_REG always return the value of Px[y] port pins, unless the analog input PID is selected.

Priority

The firmware has the possibility to assign the same peripheral output to more than one pin. It is the responsibility for the programmer to make a unique assignment.

In case more than one input signal is assigned to a peripheral input, the left most pin in Table 1 has priority.

Direction control

Direction is controlled by setting:

Px_yy_MODE_REG[9-8]

- 00 = Input, no resistors selected
- 01 = Input, pull-up selected
- 10 = Input, Pull-down selected
- 11 = Output, no resistors selected

In output mode and analog mode the pull-up/down resistors are automatically disabled.

If ACCESS bus or PCM interface selected, the input/output and pull-up/down control is controlled by the peripheral block.

Signals mapped on the CLASSD output are push-pull output only.

Examples:

P0_01_MODE_REG = 0x0200: General purpose input port mapped on P0[1].

P0_01_MODE_REG = 0x020C: URX input mapped on P0[1], pull-down enabled.

P0_12_MODE_REG = 0x0311: SPI1_DOUT output mapped on P0[12], pull-up/down automatically disabled if output.

P2_06_MODE_REG = 0x003F: ADC2 enabled on P2[6] pull-up/down automatically disabled in analog mode. Note that DIP < B_RC > RC2[7] = CM5 must be set to 1 to disable the RSSI discharge function from the RSSI / ADC2 pin.

1.2.1 SDRAM control signal selection

If SDCKE is selected, all other SDRAM signals SDWEN, SDBA0, SDBA1, SDLDQM, SDUDQM, SDRASn, SDCASn are automatically configured for SDRAM. This overrules any other PID. SDCLK and SF_ADV must be selected separately.

Refer to chapter "Boot Program" on page 36 for more information on sharing SDRAM and static memory busses.

Confidential!

Table 3: Programmable Peripheral Matrix

Table 3: Programmable Peripheral Matrix

VoIP processor with integrated cordless telephony interface

SC14452



Table 3: Programmable Peripheral Matrix



Table 3: Programmable Peripheral Matrix

VoIP processor with integrated cordless telephony interface

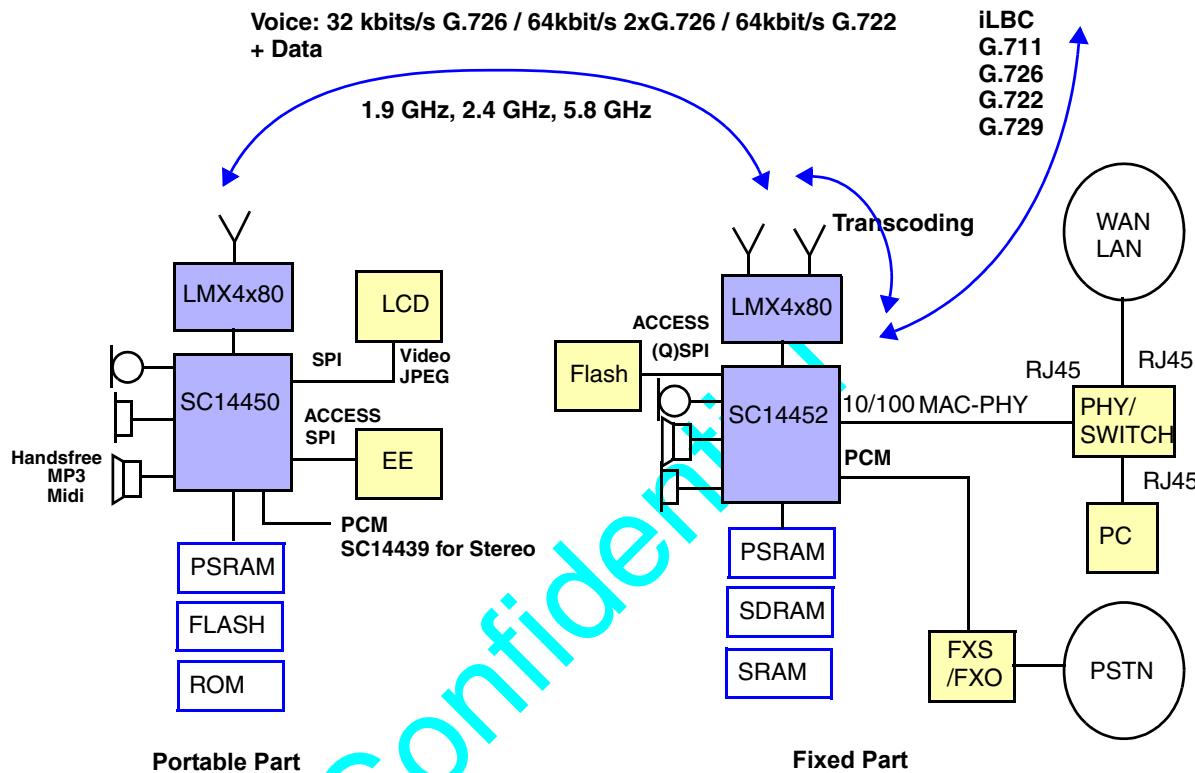
SC14452



2.0 Introduction

The SC14452 is a VoIP processor with an integrated integrated 10/100 Mbit ethernet MAC and a cordless telephony interface supporting DECT (6.0), CAT-iq,

WDCT and 5.8 GHz. It is targeted at VoIP desktop phones, ATAs and IP DECT basestations. Typical applications are shown in Figure 1.



September 12, 2008

Figure 1 SC14452 Portable Part and Fixed Part application example

3.0 Functional overview

SC14452

VoIP processor with integrated cordless telephony interface

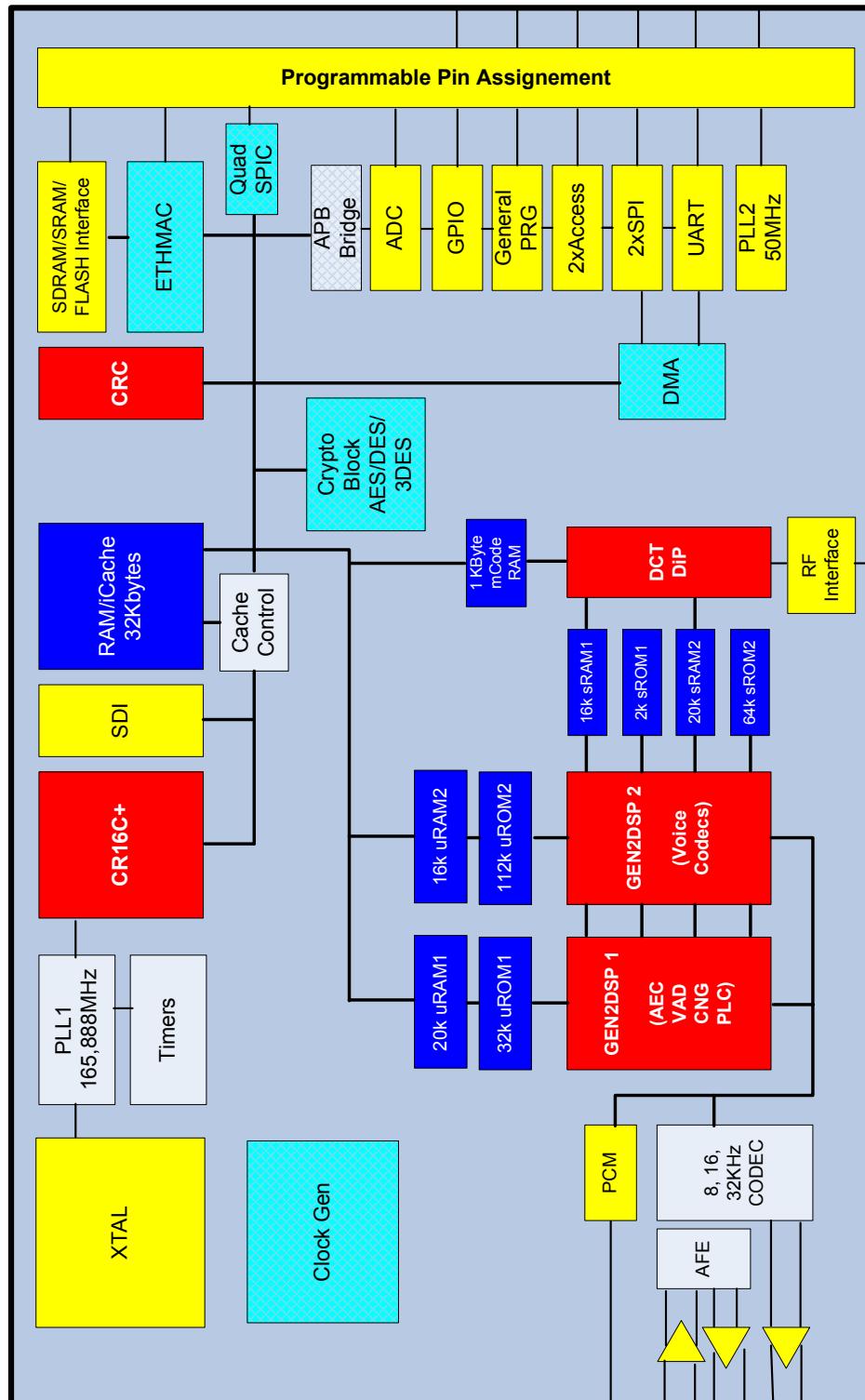


Figure 2 SC14452 Block diagram

4.0 System Clock generation

4.1 CRYSTAL OSCILLATOR

The Digital Controlled Xtal Oscillator (DXCO) is a Colpitts oscillator designed for low power consumption and high stability. The crystal oscillator frequency ranges from 9.216 MHz to 13.824 MHz depending on the application.

4.2 OSCILLATOR FREQUENCY DEVIATION

The Crystal frequency is always specified for a certain load capacitance CL. When the crystal is loaded with a different capacitance Cload, the frequency will deviate proportional to the pulling sensitivity S:

$$\frac{\Delta f}{f_c} = S \cdot (C_L - C_{LOAD})$$

The load capacitance is dependent on both internal and external capacitors. An equivalent circuit with all important capacitors is depicted in Figure 3. The values of the internal capacitors are listed in Table 4.

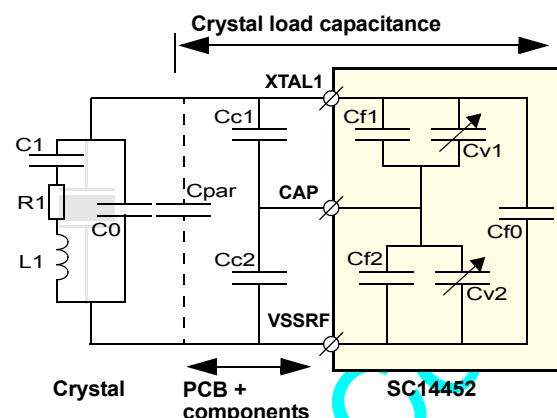


Figure 3 Xtal Equivalent circuit

Table 4: Capacitor values for eLQFP package

Capacitor	MIN	TYP	MAX
Cv1 (trim = 00)	0	0	0
Cv1 (trim = FF)	6.25 pF	9 pF	11.75 pF
Cv2 (trim = 00)	0	0	0
Cv2 (trim = FF)	8.75 pF	12.5 pF	16.25 pF
Cf0		2 pF	
Cf1		1 pF	
Cf2		2 pF	

For improved start-up, additional Cv1 and Cv2 of 25 pF each, are enabled. These capacitors must be disabled after start-up for reduced power consumption of the oscillator using CLK_XTAL_CTRL_REG[XTAL_EXTRA_CV]

With the equations below and the values from Table 4 the load capacitance can be calculated.

$$C_{x0} = C_{par} + C_{f0}$$

$$C_{x1} = C_{c1} + C_{f1} + C_{v1}$$

$$C_{x2} = C_{c2} + C_{f2} + C_{v2}$$

$$C_{LOAD} = C_{x0} + \frac{C_{x1} \cdot C_{x2}}{C_{x1} + C_{x2}}$$

Select the external capacitors Cc1 and Cc2 and the crystal parameters CL and S such that the required frequency range is obtained.

Refer to [AN-D-xxx for SC14450 Xtal oscillator](#) which includes a spread sheet for frequency deviation calculation based on Xtal and PCB parameters.

4.3 FREQUENCY CONTROL

Register CLK_FREQ_TRIM_REG controls the trimming of the crystal oscillator. The frequency is trimmed by two on-chip variable capacitors. Both capacitors are controlled through the same register.

Each of the two variable capacitors vary from minimum to maximum in 255 equal steps. With CLK_FREQ_TRIM_REG = 00 minimum capacitance and thus maximum frequency is selected. With CLK_FREQ_TRIM_REG = 0xFF maximum capacitance and thus minimum frequency is selected.

The five least significant bits of CLK_FREQ_TRIM_REG directly control five binary weighted capacitors, as depicted in Figure 4. The three most significant bits are decoded according to Table 5. Each of the seven outputs of the decoder controls a capacitor (capacitor value is 32 times the value of the smallest capacitor).

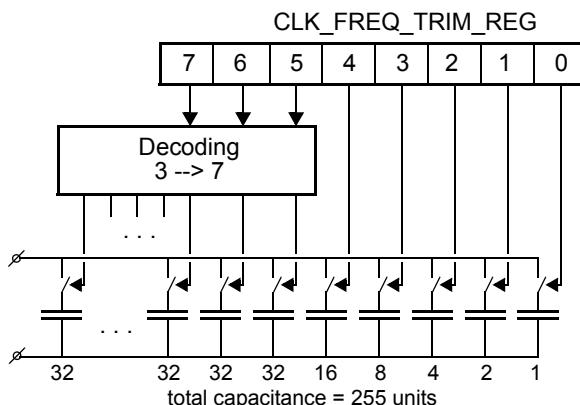


Figure 4 Frequency trimming

Trimming might cause phase jumps. In order to reduce these phase jumps change only one single switch at a time (this is especially true for the seven largest capacitors). Use bit 7...5 for coarse adjustment and always increment or decrement this value by 1. Wait approxi-

mately 10 msec to allow the adjustment to settle.

Table 5: CLK_FREQ_TRIM_REG Decoding

input[2:0]			output[6:0]						
2	1	0	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	1	1
0	1	1	0	0	0	0	1	1	1
1	0	0	0	0	1	1	1	1	1
1	0	1	0	0	1	1	1	1	1
1	1	0	0	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

Bits 4...0 are used for fine adjustment and preferably change only one bit per adjustment cycle.

As an example, the recommended way to change the frequency trim register from 0xFF to 00 is first to decrement the value of the three most significant bits by 1 at a time, and then change the least significant bits one-by-one:

0xFF --> 0xDF --> 0xBF --> 0x9F --> 0x7F --> 0x5F --> 0x3F --> 0x1F --> 0x0F --> 0x07 --> 0x03 --> 0x01 --> 0x00.

Adjusting the frequency will not affect the PLL behaviour. The small phase shifts will be followed by the PLL.

4.4 PHASE LOCKED LOOPS (PLL)

The SC14452 has two Phase Locked Loops (PLL) as shown in Figure 5:

- PLL1 for maximum CR16C+ and Gen2DSP: 165,888 or 82.944. For a basic DECT cordless telephone. The PLL is not required during a normal voice connection, but only needed if a higher cycle budget on CR16C+ or Gen2DSP is required like the handsfree application.
- PLL2 generates a 50MHz clock which is used as reference clock the internal QSPI Controller. It is possible to observe this clock signal to the external Fast Ethernet PHY through the EMAC_REF_CLK pin either as it is (50MHz) or divided by 2 (25MHz). **However, since the clock jitter is 1ns, it is not recommended to use this signal as a reference clock for external Ethernet PHYs.**

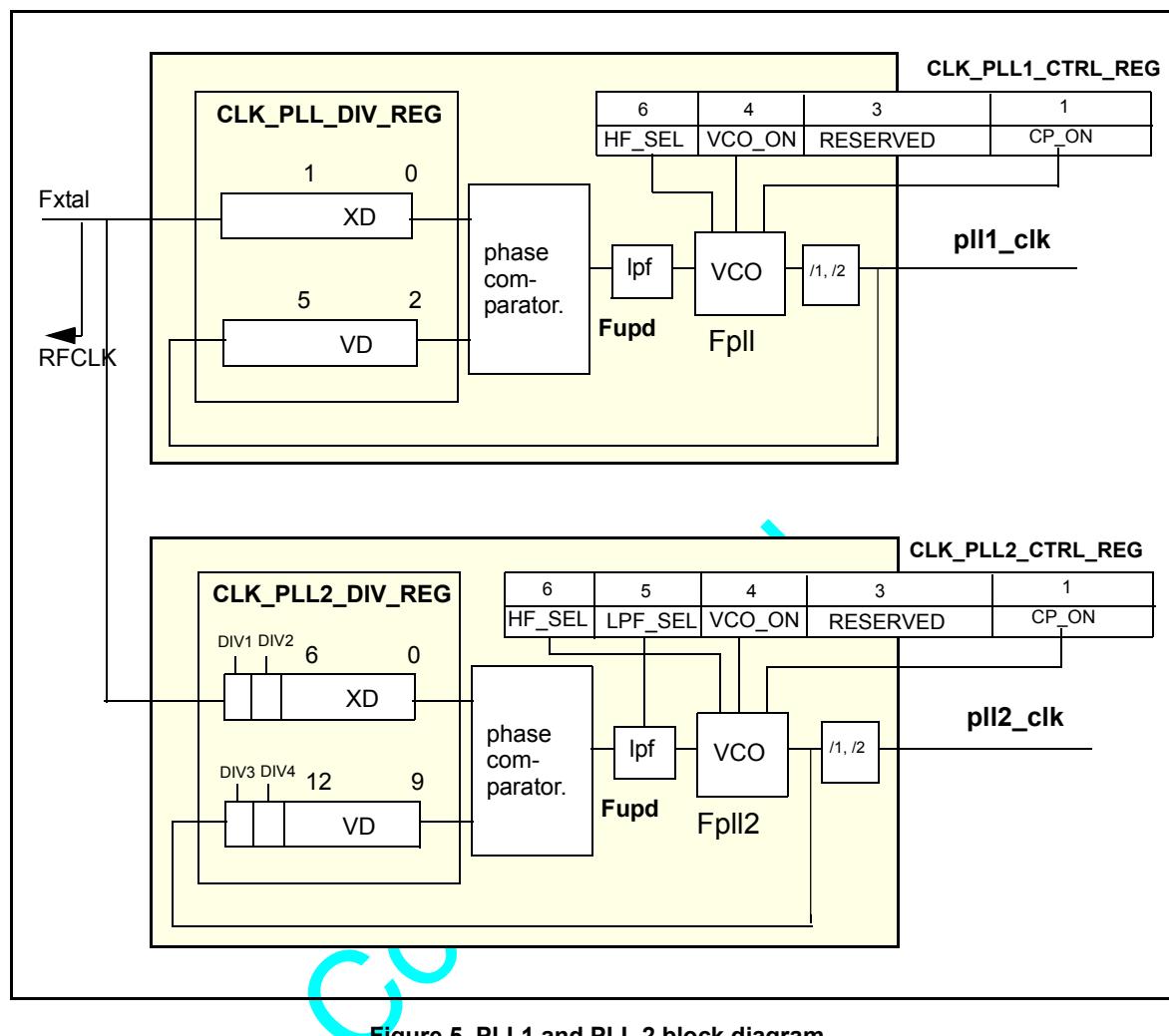


Figure 5 PLL1 and PLL 2 block diagram

4.5 SYSTEM CLOCK GENERATION

The clock generation is based on clock gating elements which produce the respective frequency by removing pulses from the PLL1/PLL2/XTAL clock waveform. The resulting clock is not 50% duty cycle. Whenever this is a requirement, an additional FF is used to further divide the clock gater's output by two, thus creating a 50% duty cycle waveform.

The following paragraphs describe the clocks, the minimum and maximum frequencies and the allowed division numbers when the system is clocked by the PLLs as well as by the XTAL. The division values for the PLL1 assume a PLL1 frequency of 165,888MHz.

Figure 6 depicts the generation of the various clocks of the system that start from the PLL1. SW_<name>_div denotes the programmable division factor while SW_<name>_en describes the start/stop enabling signals.

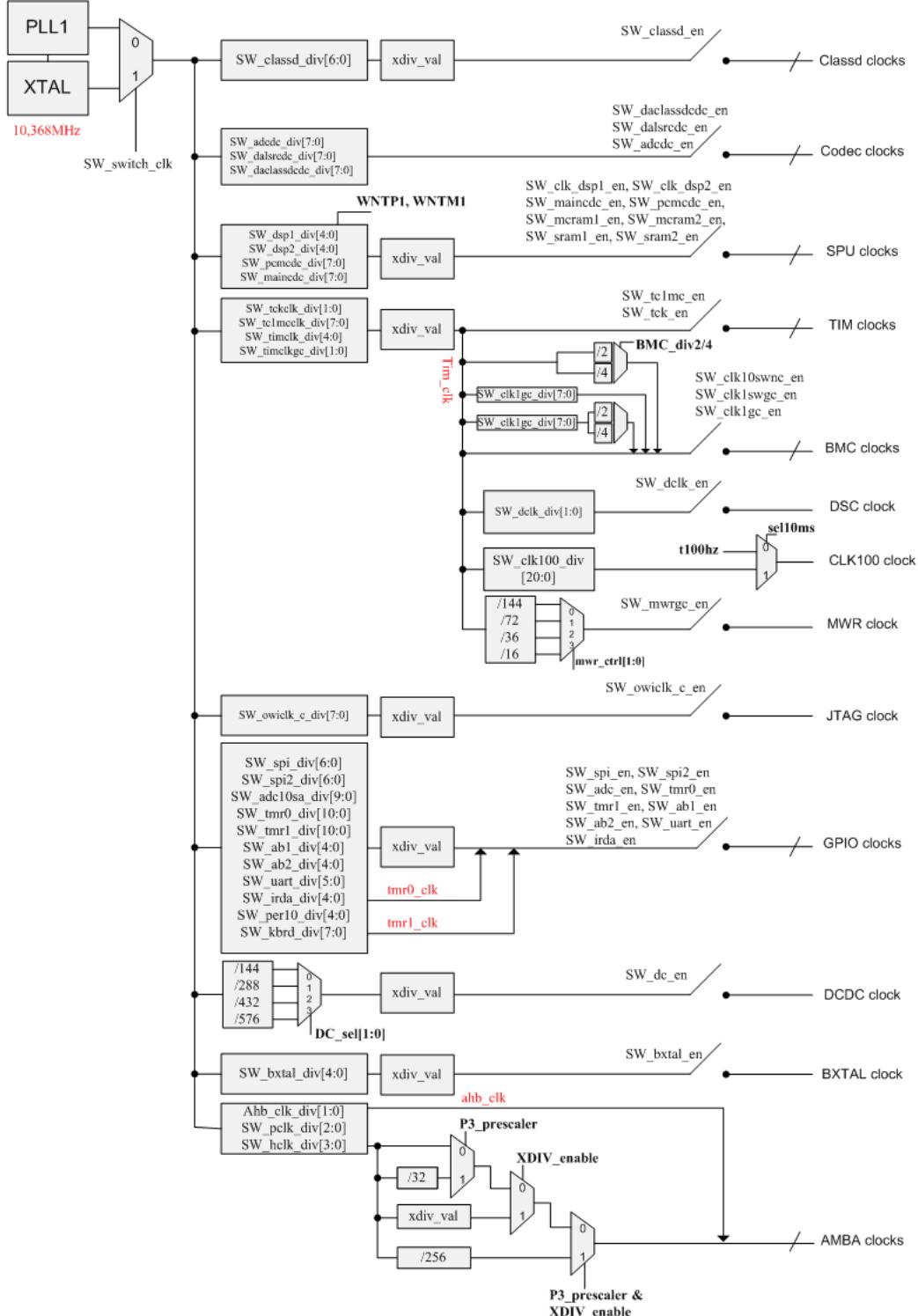


Figure 6 System clocks generation overview

ClassD Clocks

These clocks are generated for clocking the classd module. During switching from XTAL to PLL1 mode, these clocks must be switched off.

These clocks will retain their period while switching from XTAL to PLL1 mode. That means, programming their division with 32 is equal to programming with 2, both in XTAL mode. In the first case, the period will remain the same after switching without re-programming.

Table 6: ClassD clocks

Clock Name	Min/Max Freq (MHz)	PLL1 Div	XTAL Div
classd_clk	5.184	32	2

Codec clocks

These clocks are generated for clocking the codec module. During switching from XTAL to PLL1 mode and vice versa, these clocks must be switched off.

Table 7: Codec clocks

Clock Name	Min/Max Freq (MHz)	PLL1 Div	XTAL Div
ad_cdc_clk	1.152/ 2.304/ 4.608	144/72/36	9/4.5/2.25
da_lsr_cdc_clk	1.152/ 2.304/ 4.608	144/72/36	9/4.5/2.25
da_classd_cdc_clk	1.152/ 2.304/ 4.608	144/72/36	9/4.5/2.25

The fractional division is achieved using a pattern at the clock waveform.

SPU clocks

These clocks are generated for clocking the core of the Signal Processing Unit, the memory modules involved and the PCM interface.

Table 8: SPU clocks

Clock Name	Min/Max Freq (MHz)	PLL1 Div	XTAL Div
pcm_cdc_ck	1.152/ 2.304/ 4.608	144/72/36	9/4.5/2.25
sram1_clk	10.368..16 5.888	1	1
sram2_clk	10.368..16 5.888	1	1
mram1_ck	10.368..16 5.888	1	1

Table 8: SPU clocks

Clock Name	Min/Max Freq (MHz)	PLL1 Div	XTAL Div
mram2_ck	10.368 .. 165.888	1	1
dsp1_clk	1.296 .. 82.944	1 .. 8	1 .. 8
dsp2_clk	1.296 .. 82.944	1 .. 8	1 .. 8
main_cdc_clk	1.152/ 2.304/ 4.608	144/72/36	9/4.5/2.25

Before activating the dsp1_clk/dsp2_clk clocks only when in PLL mode, the dsp1/2 memory access (register bits CLK_SPU1_REG [SW_DSP1_MEM_ACC_FORCE] and CLK_SPU1_REG[SW_DSP2_MEM_ACC_FORCE]) has to be set to 1.

Setting the bits CLK_SPU1_REG [SW_DSP1_MEM_ACC_FORCE] and CLK_SPU1_REG[SW_DSP2_MEM_ACC_FORCE] is only allowed when the PLL frequency is 80 mhz or below. This is to save power PLL on 80 Mhz, RAM on 80 Mhz. But then you pay a penalty for CR16 access on the cycle budget from the DSP.

TIM clocks

These clocks are generated for clocking the TIM module. Switching from PLL1 clock to XTAL clock is only allowed in blinds slots.

Table 9: TIM clocks

Clock Name	Min/Max Freq (MHz)	PLL1 Div	XTAL Div
tim_clk	10.368/ 20.736	16/8	1
tim_clk_gc	10.368/ 20.736	16/8	1
tck_clk	10.368/ 20.736	16/8	1
tc1mc_clk	1.152/ 2.304	1.152/ 2.304	9

BMC clocks

These clocks are used for clocking the BMC module

Table 10: BMC clocks

Clock Name	Min/Max Freq (MHz)	PLL1 Div	XTAL Div
clk10sw_c	2.592 .. 20.736	64/32/16/8	4/2/1
clk1_gc	1.152/ 2.304	144/72	9

Table 10: BMC clocks

Clock Name	Min/Max Freq (MHz)	PLL1 Div	XTAL Div
clk1sw_gc	0.288 .. 2.304	576/288/144/72	36/18/9

MWR Clock

This clock is generated for clocking the microwire interface

Table 11: MWR clock

Clock Name	Min/Max Freq (MHz)	PLL1 Div	XTAL Div
mwr_gc	1.152 .. 10.368	144/72/36/16	9/4.5/2.25/1

The fractional division is achieved using a pattern at the clock waveform. Programming must be always done thinking at the PLL1 mode. When switching to XTAL mode, period will remain the same and the division factors will be automatically adjusted.

DSC clocks

These clocks are generated for clocking the ciphering module.

Table 12: DSC clocks

Clock Name	Min/Max Freq (MHz)	PLL1 Div	XTAL Div
dclk_g	10.368/20.736	16/8	1
dclk_gn	10.368/20.736	16/8	1

RISCutil clock

This clock is either 100Hz or 100KHz, depending on the DEBUG_REG[CLK100_SRC].

Table 13: RISCutil clock

Clock Name	Min/Max Freq (Hz)	PLL1 Div	XTAL Div
clk100	100Hz/100KHz	1658880	110524

JOWI clock

This is the clock generated for the JTAG interface

Table 14: JOWI clock

Clock Name	Min/Max Freq (MHz)	PLL1 Div	XTAL Div
owiclk	1.296 .. 82.944	1.2..4.8 .. 128	1.2..4.8

GPIO clocks

These clocks are generated for clocking the peripherals of the system namely, the ADC, the timer0 and timer1, the Access Bus Interfaces, the UART, the SPI and SPI2.

Table 15: GPIO clocks

Clock Name	Min/Max Freq (MHz)	PLL1 Div	XTAL Div
adc_clk	0.288	576	36
adc10sa_clk	0.288	576	36
tmr0_clk	0.1152/1.152	1440/144	90/9
tmr1_clk	0.1152/1.152	1440/144	90/9
ab1_clk	10.368	16	1
ab2_clk	10.368	16	1
uart_clk	10.368	16	1
irda_clk	10.368	16	1
kbrd_clk	1.152	144	9
spi_clk	1.296 .. 82.944	1 .. 63	1..4
spi2_clk	1.296 .. 82.944	1 .. 63	1..4
per10_clk	10.368	16	1

The SPI clocks (spi_clk and spi2_clk) can retain their period when switching from PLL1 mode to XTAL mode and vice versa, if programmed using the PLL1 division factors.

DCDC clock

This is the clock generated for the DC-DC converter

Table 16: DCDC clock

Clock Name	Min/Max Freq (MHz)	PLL1 Div	XTAL Div
dc_clk	0.288/0.384	576/432	36/27

The high time of this clock's pulse is always 192ns.

AMBA clocks

These clocks are generated for clocking the AHB/APB bus and the cache memory

Table 17: AMBA clocks

Clock Name	Min/Max Freq (MHz)	PLL1 Div	XTAL Div
ahb_clk	10.368/82.944	2	1

Table 17: AMBA clocks

Clock Name	Min/Max Freq (MHz)	PLL1 Div	XTAL Div
hclk	0.005 .. 82.944	0,1,2,4,8, 16	0,1,2,4,8, 16
pclk	0.005 .. 41.472	1,2,4,8,16	1,2,4,8,16

The hclk division must be set to greater than or equal to 4 before switching from XTAL to PLL1 mode. Division of 2 while switching might result to unwanted behaviour.

BXTAL clock

BXTAL can be used to clock external devices like TAM/Codec (e.g SC14439). BXTAL is available on many port pins. The valid divider values are 1 and even numbers up to 32.

Table 18: BXTAL clock

Clock Name	Min/Max Freq (MHz)	PLL1 Div	XTAL Div
bxtal_clk	1.296 .. 165.888	1,2,4,6,.. 32	1,2,4,6,.. 32

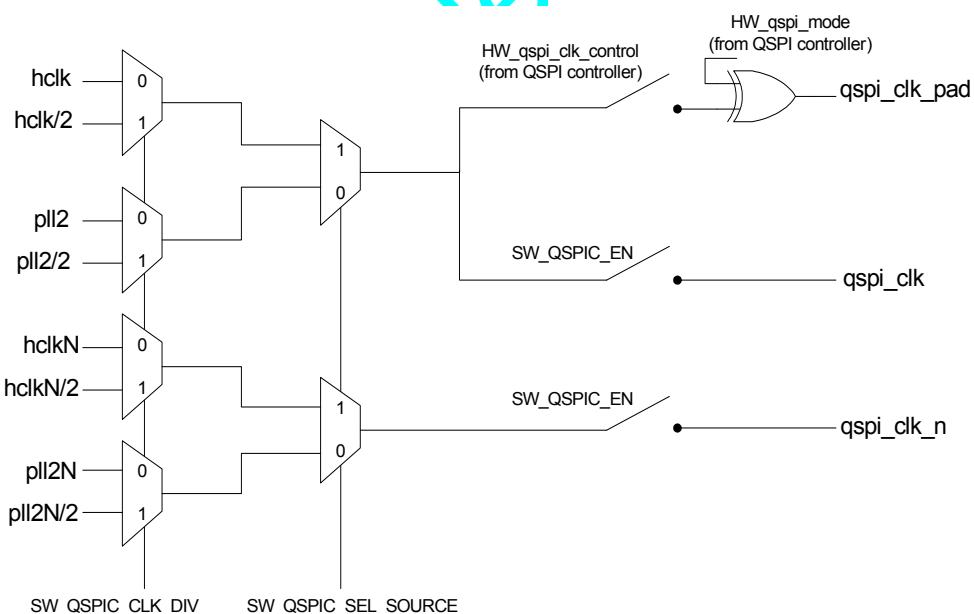
QSPI clocks

These clocks are used for clocking the Quad SPI Controller. Source of these clocks can be either the PLL1 or the PLL2. In the first case the clocking frequency is 50MHz while in the latter the clock frequency equals the AMBA system clock (hclk). An extra division by 2 is provided via programming of the SW_QSPIC_CLK_DIV.

Table 19: Qspi clocks

Clock Name	Min/Max Freq (MHz)	PLL2 Div	XTAL Div
qspi_clk	25, 50, hclk/2, hclk	1,2	1
qspi_clk_n	25, 50, hclk/2, hclk	1,2	1
qspi_clk_p ad	25, 50, hclk/2, hclk	1,2	1

The clock generation is depicted at Figure 6:


Figure 7 QuadSPI clocks generation

The qspi_clk_pad clock is used to clock the QSPI Flash module, while the qspi_clk/qspi_clk_n signals are clocking the QSPI controller circuits.

EMAC clocks

These clocks are generated for clocking the Ethernet MAC IP. Their source should be an external clock.

The following Figure 8 displays the different clocking capabilities of the EMAC module.

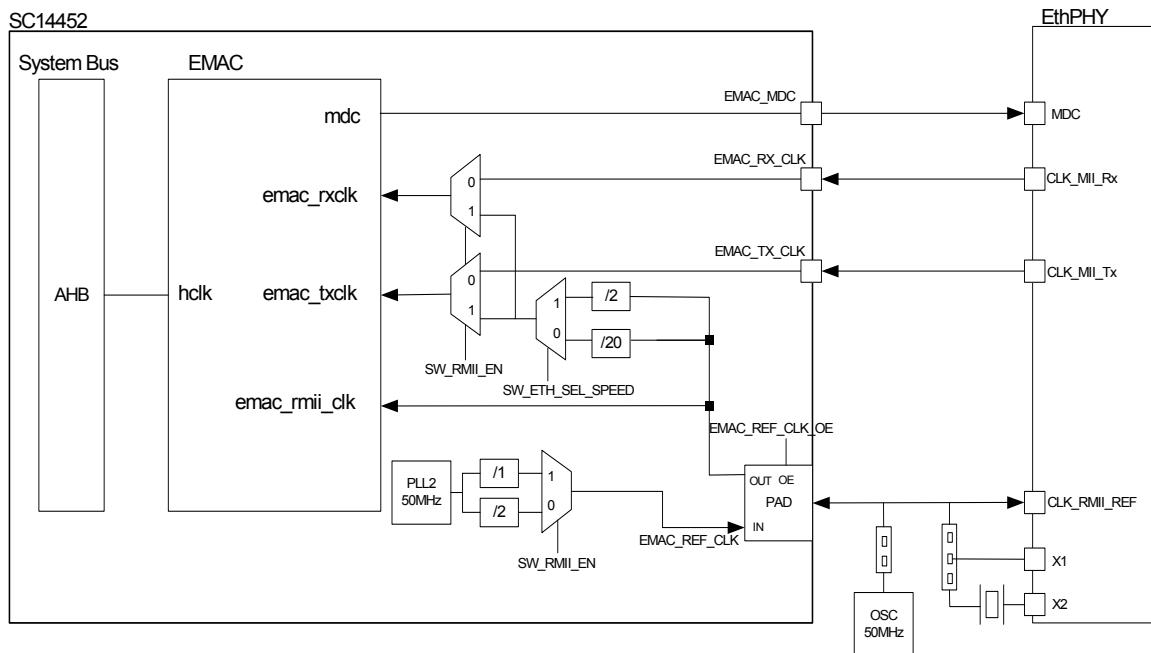


Figure 8 EMAC clocking options

The SC14452 software selects the Ethernet clocking with respect to the PHY by setting the following register bits:

"CLK_AUX_2_REG[SW_RMII_EN]", which selects for MII or Reduced MII mode.

"GPRG_R0_REG[EMAC_REF_CLK_OE]" which selects the direction of the reference clock (depicted as CLK_EMAC_Tx at the Figure 8)

The selection for the source of the RMII clock a.k.a from an external pin, is actually performed via the

selection of the direction of the EMAC_REF_CLK pad. It is not recommended to use the PLL2 as a source for the RMII clock since this signal comprises 1ns jitter and thus, cannot be applied to the external EthPHY devices.

The following table depicts the programming of the registers with respect to the mode of operation (MII/RMII, external clock):

Table 20: Programming for various EMAC operation modes

Mode	SW_RMII_EN	SW_ETH_SEL_SPEED	EMAC_REF_CLK_OE
RMII 100Mbps clock External	1	1	1
RMII 10Mbps clock External	1	0	1
MII 100Mbps clock External	0	1	*
MII 10Mbps clock External	0	0	*

The following table presents the EMAC clocks with respect to the operation modes:

Table 21: EMAC clocks for various operating modes

Mode	emac_txclk	emac_rxclk	emac_rmii_clk	EMAC_TX_CLK	EMAC_RX_CLK	EMAC_REF_CLK
RMII 100Mbps clock External	25 MHz source EMAC_REF_CLK div=2	25 MHz source EMAC_REF_CLK div=2	50 MHz source EMAC_REF_C_LK div=1	-	-	50 MHz Input
RMII 10Mbps clock External	2,5 MHz source EMAC_REF_CLK div=20	2,5 MHz source EMAC_REF_CLK div=20	50 MHz source EMAC_REF_C_LK div=1	-	-	50 MHz Input
MII 100Mbps clock External	25 MHz source EMAC_TX_CLK div=1	25 MHz source EMAC_RX_CLK div=1	*	25 MHz Input	25 MHz Input	*
MII 10Mbps clock External	2,5 MHz source EMAC_TX_CLK div=1	2,5 MHz source EMAC_RX_CLK div=1	*	2,5 MHz Input	2,5 MHz Input	*

4.6 LOW POWER MODE

The clock generation module contains provision for low power modes. There are two different ways of reducing power: disabling the clock gating cells that generate the divided clocks or using the extra division factor (xdiv).

4.6.1 Clock Gating

All signals that apply to the first low power technique are named after the convention:
SW_<clockname>_en.

For instance, the SW_mcram1_en bit found at the CLK_AMBA_REG will start/stop the clock for the MicroCode RAM1 module. Almost all generated clocks except for the hclk and pclk (AMBA clocks) can be started/stopped already in the Clock Generation module, even before the clock tree buffers, thus enhancing power saving.

4.6.2 Extra Division

The second way of reducing power is to further divide all clocks by the same division at the same time. This is a DIP controlled function and cannot be triggered by just programming the clock generation module parameters. It is triggered by the DIP controller via the P1 and P3 signals. Note that the DIP signal P1 is masked from the XDIV signal (inside the DIP block) and after that it is further being masked by the CLK_XDIV0_REG and CLK_XDIV1_REG bits. However, the extra division value can be programmed at the CLK_XDIV_VAL_REG. This register specifies that all

selected clocks (via CLK_XDIV0_REG and CLK_XDIV1_REG bits) will be further being divided by 2^{CLK_XDIV_VAL_REG}. In the case of AMBA clocks (refer to Figure 6, APB/AHB clocks) this 2^{CLK_XDIV_VAL_REG} factor will be applied if DIP.P3 = 0 and DIP.P1 = DIP.XDIV = CLK_XDIV1_REG[AHB_XDIV_EN] = 1 (note that DIP_CTRL_REG[PRESCALER] bit is a global enable bit for both DIP P1 and P3 signals). There are also two registers that control whether this extra division is to be applied at the respective generated clocks. The registers CLK_XDIV0_REG and CLK_XDIV1_REG contain enable bits for the extra division for groups of clocks. The following clocks are not affected by this function:

- ClassD clocks (Classd_clk, Classd_phase23_clk, Classd_phase34_clk)
- Codec clocks (ad_cdc_clk, da_lsr_cdc_clk, da_classd_cdc_clk)
- Timer clocks (tmr0_clk, tmr1_clk)
- Reset synchronization logic (ahb_clk)

Switching from PLL clock to XTAL clock and vice versa is not allowed while in the Extra Division (XDIV) mode.

4.7 SWITCHING PROCEDURES

4.7.1 LDO_XTAL switching On/Off

From startup, the Xtal oscillator is supplied from VDDRF. Before the Xtal supply can be supplied from LDO_XTAL, this LDO must first be switched on and be stabilized.

LDO switch on procedure:

- Set CLK_XTAL_CTRL_REG[LDO_XTAL_ON] = 1.
- Wait 200 us for LDO_XTAL to stabilize or CLK_XTAL_CTRL_REG[AVD_XTAL_OK] = 1.
- Set CLK_XTAL_CTRL_REG[XTAL_SUPPLY] = 0.

LDO switch off procedure:

- Set CLK_XTAL_CTRL_REG[XTAL_SUPPLY] = 1.
- Set CLK_XTAL_CTRL_REG[LDO_XTAL_ON] = 0.

Simultaneous setting of these bits is not possible, it must be in this order.

4.7.2 PLL1/PLL2 switching On/Off

The sequence to enable and disable the PLL1 is the following:

For **10.368 MHz XTAL**, Set PLL Xtal divider XD and VCO divider VD:

CLK_PLL1_DIV_REG = 0x1C for 165.888 MHz

CLK_PLL1_DIV_REG = 0x4 (recommended) or 0x1D for 82.944 MHz.

For **12.288 MHz XTAL**, set PLL Xtal divider XD and VCO divider VD CLK_PLL1_DIV_REG = 0x19 for 165.888 MHz

CLK_PLL1_CTRL_REG[PLL_OUT_DIV]=1 for 82.944 MHz.

VCO and charge pump can be switched on with

CLK_PLL1_CTRL_REG[4] = VCO_ON set to 1 and CLK_PLL1_CTRL_REG[1] = CP_ON set to 1.

The sequence to enable and disable the PLL2 is the following:

Set PLL2 Xtal divider XD and VCO divider. Refer to CLK_PLL2_CTRL_REG and CLK_PLL2_DIV_REG for settings.

CLK_PLL2_CTRL_REG[4] = VCO_ON set to 1 and

CLK_PLL2_CTRL_REG[1] = CP_ON set to 1.

The clock may be used after 500 us PLL2 start up time.

Set the corresponding bits in CLK_AUX2_REG in order to propagate the clock to the proper points of the system.

4.7.3 Switching between PLL1/XTAL

The system will be clocked either by the XTAL or the PLL1 clock. The switching between these two different domains is performed by the CLK_GLOBAL_REG[SW_SWITCH_CLK]. Setting this bit to '0' means that the PLL1 will clock the system.

Note that the actual switching of the clocks will be done a couple of clock cycles after the programming of the

aforementioned register. An important restriction of the switching procedure is that **the division factor of the hclk clock must be different than 2** before the switching is even programmed. Proposed value is 4:

```
//XTAL domain
// hclk = xtal clock / 4
CLK_AMBA_REG[SW_HCLK_DIV] = 4;
Delay(at least 4 clock cycles);
// switch to PLL1 domain
CLK_GLOBAL_REG[SW_SWITCH_CLK] = 0;
```

Switching from the PLL1 domain to XTAL can be performed with no restrictions.

4.7.4 Programming different frequencies at PLL1

The PLL1 can be programmed to give 165.888MHz, 82,944MHz or 41,472MHz. Seamless switching is only supported for the 165.888MHz and, 82,944MHz frequencies.

The CLK_GLOBAL_REG[SW_XTAL_PLL1_RATE] depicts the relation of the PLL1 and XTAL frequency. Initial value is 16 i.e. 165,888MHz/10,368MHz = 16.

When the PLL1 is programmed at a smaller frequency, this bit has also to be reprogrammed. If the PLL1 is set to 82,944MHz then SW_XTAL_PLL1_RATE should be set to 0 (meaning 8).

This is used for the automatic adjustment of clock division factors when switching from PLL1 to XTAL mode and vice versa. For example, the Timer0 clock division can be initially programmed to 144 which applies to the PLL1 mode. That would then a Timer0 clock frequency of $165,888 / 144 = 1,152\text{MHz}$.

When going to XTAL mode, this division factor will be automatically divided by SW_XTAL_PLL1_RATE and provide the adjusted division factor for the XTAL domain (in this case $144 / 16 = 9$). In this manner, the clock will retain the same period independently of the actual domain PLL1 or XTAL.

The clocks that are capable of retaining the same frequency without re-programming are the following:

- Codec Clocks
- MWR clock
- TMR0, TMR1 clocks
- AB1 AB2 clocks
- UART clock
- IrDA clock
- SPI, SPI2 clocks
- PCM clock
- MAINCDC clock

5.0 Power Management

The SC14452 has a power management function integrated.

Features:

- DC-DC convertor which can be configured as step-down (buck) converter
- Low Drop Out regulator (LDO2) for 1.8V core voltage.

- Low Drop Out regulators (LDO1) for programmable IO voltage (1.8 - 3.3V)
- On / off control
- Voltage measurement via scaler to 10 bits ADC
- Temperature measurement.

The following Figure 9 presents the connectivity of the Power Management.

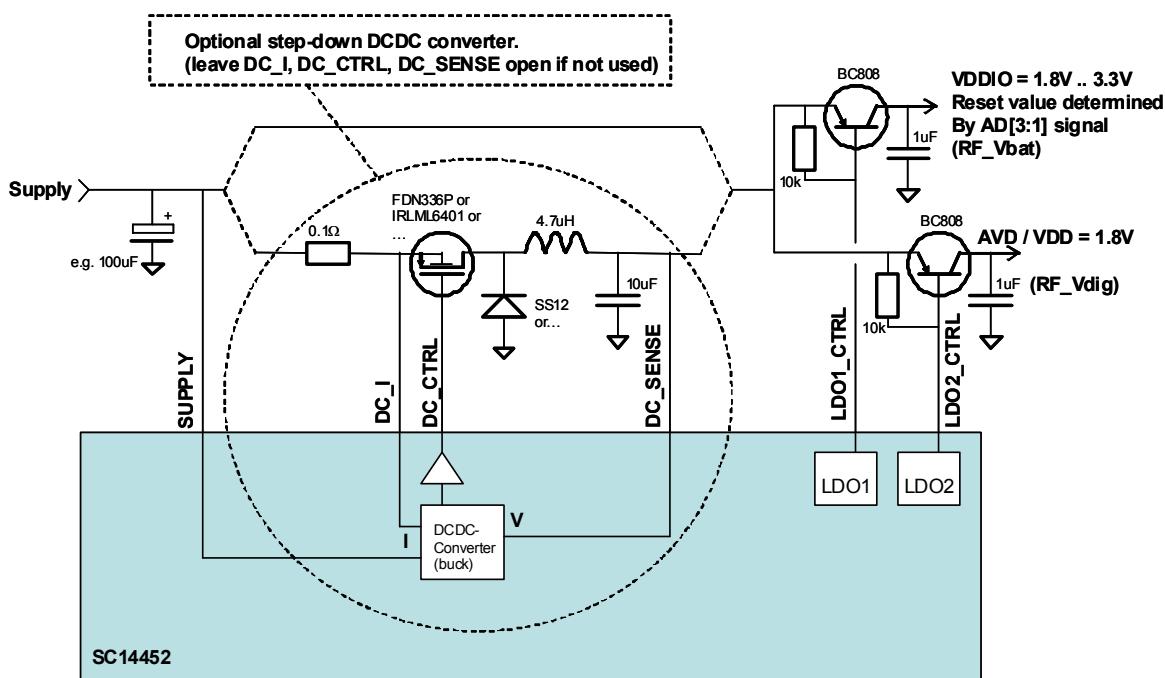


Figure 9 SC14452 Power management application

5.1 DC-DC CONVERTER

The SC14452 has an integrated control circuit for a step-down (buck) converter.

Included is a current limiting function (800mA over 100mili-ohm), and a soft-start function to reduce the maximum current during start-up (300mA). The switching frequency is related to the DECT clock in order to reduce interference with the other circuits in the device. The switching frequency can be set with SUPPLY_CTRL_REG[DC_FREQ].

From reset DCDC convertor is switched on. If the DC/DC is not used, in basestation, the **DC_ON** bit must set to 0.

5.2 LOW DROPOUT REGULATORS

The SC14452 has several control circuits for Low Dropout regulators (LDO).

- LDO2 has a fixed output voltage of 1.8V. LDO2 is activated automatically if start-up requirements are met. Refer to [Table 362 for LDO2 characteristics](#).
- LDO1 output voltage is programmable with

SUPPLY_CTRL_REG[LDO1_LEVEL] between 1.8V(*), 2.0, 2.5, 3.0, 3.3V(*). The reset value of DC_VOUT and LDO1_LEVEL is determined by the level of AD[3-1] when RSTn is low and this level is latched on the rising edge of RSTn. The latched values of AD[3-1] can be read in TEST_ENV_REG (0xFF4800) (table 325, page 281) bit 1.

LDO1 is enabled at startup if AVD2>1.8V. This means that LDO2 can not be connected to the output of LDO1. In case LDO1 is not used LDO1_sense must be connected to ground either or not via a resistor. The outputs of the LDOs can directly drive the base of external pnp transistors. Refer to [Table 361 for LDO1 characteristics](#).

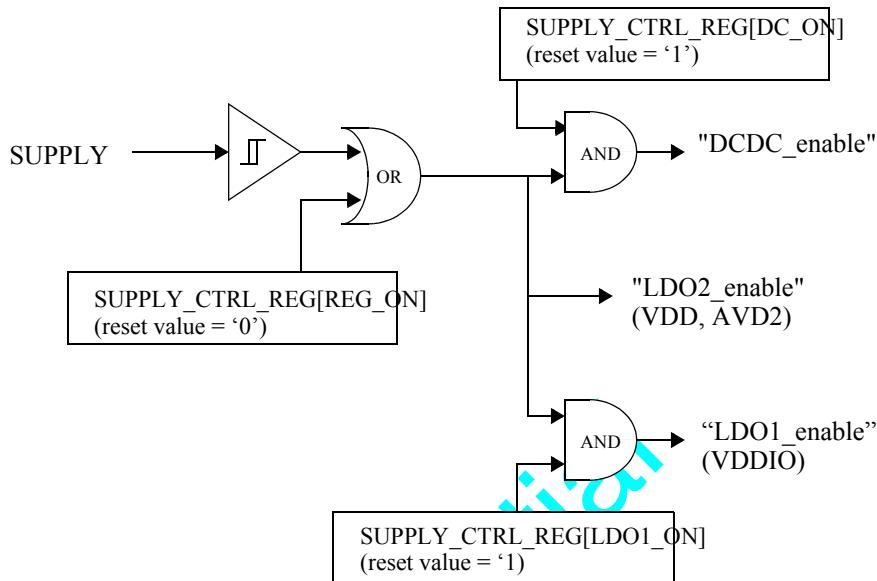
- LDO_XTAL provides 1.6V for Xtal oscillator, 6 bits TDO volume DAC and 10 bits DAC for ADC. For the two DACs, this LDO must be on. Refer to [Table 363 for LDO_XTAL characteristics](#).
- LDO_RFCLK provides a reference voltage for the RFCLK pad.

6.0 ON / OFF control

Switch on procedure

The SC14452 switches on the LDO's and DC-DC con-

verter as soon as the supply voltage is sufficient. After start-up, SUPPLY_CTRL_REG[REG_ON] can be set to 1 to keep the DC-DC converter and LDO's active, independent of the supply voltage.



June 12, 2008

Figure 10 Wake-up circuits

6.1 BANDGAP REFERENCE

The SC14452 has a bandgap register which is used by the analog codec front-end, Radio frontend and ADC as reference voltage.

The BANDGAP_REG has reset value 0x8, resulting in an initial VDD/AVD2/VDDRF voltage of 1.8V +/- 5%. This voltage can be tuned in production to 1.8V with BANDGAP_REG bits 3-0 to an accuracy better than 1%.

If VDD/AVD2/VDDRF is trimmed to 1%, VDDIO is 2% accurate.

The trimmed value written to BANDGAP_REG can be saved by the CR16C+ in the external EEPROM and restored in the BANDGAP_REG register at system startup.

See BANDGAP_REG (0xFF4810) (table 80, page 188) for a registers on how to trim the voltages.

The BANDGAP_REG is only reset by a hardware reset (RSTn)

7.0 Power-On and system Reset

7.1 POWER ON RESET

At $AVD2 = 1.7V\text{-}1.9V$ (BANDGAP_REG=0x8, reset value, see also Table 362 on page 295), or the LDO voltage has reached a stable voltage (LDO_BB_stable), the internal signal "VDDon" releases the RSTn pin via an open drain FET. With an external capacitor connected to the RSTn pin, the internal reset is released at Vih_{rst} and all blocks, including SDI+, the SW reset and HW reset will get inactive, port P0 will be latched in TEST_ENV_REG and the boot program will be started. (See "Boot Program" on page 36).

After power-on the RSTn must stay low for at least $Trst_low_pu$ (Table 386 on page 310)

The values of the external capacitor Crst is determined by the RC product with the internal pull-up on the RSTn (Ipu_rst_lo Table 353 on page 290).

A typical value for $Crst = Trst_low_pu \cdot R = Trst_low_pu / (U/Ipu_rst_lo(\text{MAX}) = 10E-3 / (1.8/20E-6) = 100nF$.

7.2 SYSTEM RESET.

After power on, the following conditions reset the whole SC14452 or parts of it as shown in Figure 12:

- If the RSTn pin is pulled down for at least $Trst_low$, then the "VDDon" goes LOW. A full device reset is issued. The RAM content is not intentionally reset, but might contain undefined data due to an early termination of a RAM write cycle. Spikes less than 100 ns are suppressed.
- If the AVD drops below $Vrst_on\text{-}Vhyst$ (typical 1.65V with trimmed bandgap), the RSTn pin is automatically pulled low.
- If the JTAG_SD+ issues the command sequence SYSRST_ON/SYSRST_OFF all blocks, excluding JTAG_SD+ and TEST_ENV_REG, are reset but the RSTn pin is not pulled low. (Note that a JTAG command that sets the internal TRSTn pin, only sets the SDI+ TAP controller in the reset state.)
- If the watchdog timer expires and $\text{TIMER_CTRL_REG}[WD0G_CTRL] = 1$, all blocks excluding JTAG_SD+ and TEST_ENV_REG, are reset but the RSTn pin is not pulled low. If $WD0G_CTRL=0$, an NMI will be executed and the user has full control over manual reset sequence (e.g setting SW_RESET)

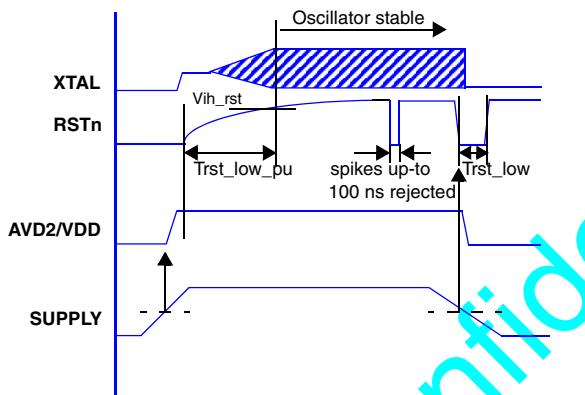


Figure 11 Reset diagrams

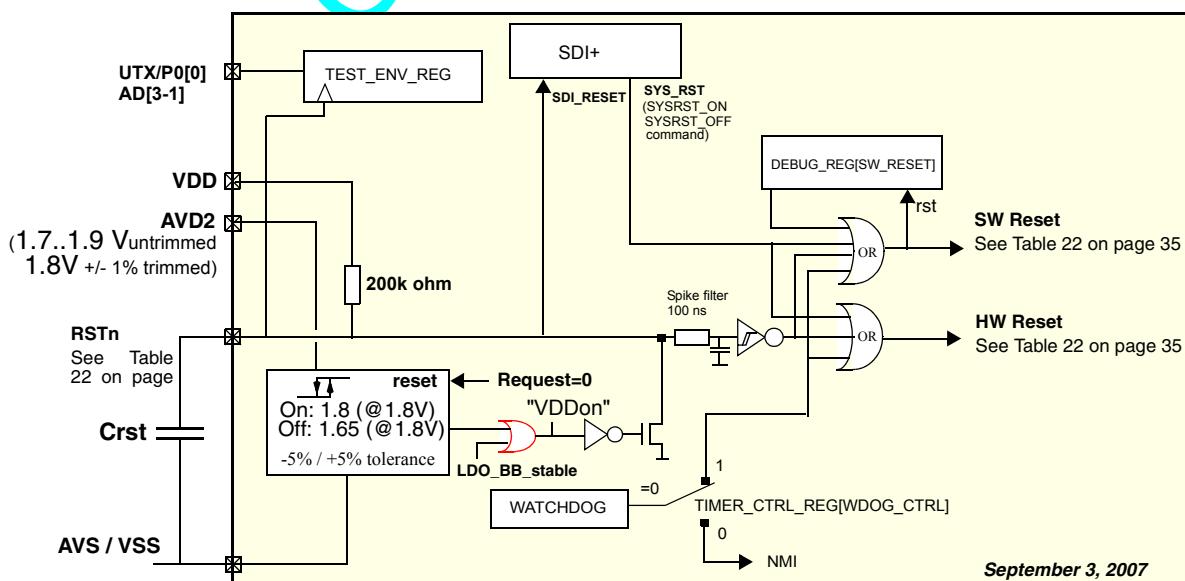


Figure 12 Reset circuit

7.3 SOFTWARE RESET

Purpose of a SW software reset is to reset the chip and jump to a specified start address defined by PC_START_REG. The software reset is a partial reset as shown in Table 22 and is invoked by setting DEBUG_REG[SW_RESET]=1.

The SW reset can be used in the following cases:

- Restart of the application upon a Watchdog time out. In case TIMER_CTRL_REG=0, an NMI will be executed and the ISR may give a SW reset.
- If a program is loaded via the UART boot protocol and SW_RESET is given, the UART boot protocol can be skipped if DEBUG_REG[ENV_B01] is set. See "Boot from internal RAM" on page 36

Special case: DC_VOUT and LDO1_LEVEL

Upon a hardware reset, the LDO1_LEVEL and DC_VOUT use the latched values of TEST_ENV_REG[AD3-1], upon a SW_RESET, the

LDO1_LEVEL and DC_VOUT value do not change

Table 22: Registers reset overview

Register	RSTn=0 or AVD< (Vrst_on-Vhyst)	SDI_SYS_RST or Watch-dog	HW reset	SW reset
- JTAG_SDI+	✓	-	-	-
- TEST_ENV_REG (+AD[3-1])	✓	-	-	-
- PC_START_REG	✓	✓	✓	-
- DEBUG_REG[CR16_DBGM]	✓	✓	✓	-
- DEBUG_REG[ENV_B01]	✓	✓	✓	-
- BANDGAP_REG	✓	✓	✓	-
- SUPPLY_CTRL_REG	✓	✓	✓	-
- Px_DATA_REG	✓	✓	✓	-
- Px_DIR_REG	✓	✓	✓	-
- Px_yy_MODE_REG	✓	✓	✓	-
- Other registers	✓	✓	✓	✓

7.4 PROGRAM COUNTER AFTER RESET

At a hardware reset caused by the RSTn pin, the PC_START_REG is reset to 0xFEF0 and the CR16C+ program counter PC bits 24, 9-0 are set to 0 and bits 23-10 are loaded with PC_START_REG[15-2] (See Figure 13)

At a SW_RESET, PC_START_REG is not reset but the CR16C+ PC is again loaded with PC_START_REG bit 15-2.

Note that the PC_START_REG can only be set to boundaries of 1k.

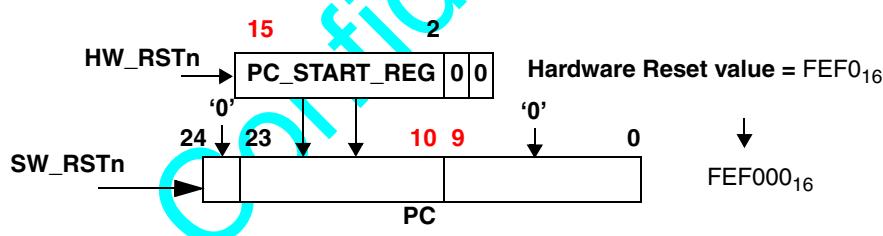


Figure 13 PC_START_REG loading in CR16C+ Program counter PC

8.0 Boot Program

The boot program is located at 0xFFE000 in ROM and is always executed after the RSTn pin is pulled low.

The boot program flow is shown in Figure 15. The boot program may boot from the following sources in this order:

1. Boot from internal RAM at 0x8080 see paragraph 8.1
2. Boot from JTAG pin see paragraph 8.2
3. Boot from UART see paragraph 8.3
4. Boot from Quad or Single SPI see paragraph 8.4
5. Boot from SDRAM. This is done by the program loaded via SPI or UART.
6. Boot from external FLASH/ROM see paragraph 8.5.

Note that the PC_START_REG is not modified in the boot code. Jumps to any address location are possible. A SW_RESET restarts using PC_START_REG which can only be set at boundaries of 1kByte.

8.1 BOOT FROM INTERNAL RAM

If DEBUG_REG[ENV_B01] = 1, the boot code will execute from on chip RAM at 0x8080. Bit ENV_B01 is mostly set in a loader program and provides a jump to the loaded program after a SW_RESET.

8.2 BOOT FROM JTAG

If pin **JTAG=0** on the rising edge of RSTn, the JTAG one wire interface is enabled and the boot code waits in an endless loop. After loading a program in on-chip or

off-chip memory, the PC_START_REG may be changed to the desired start address and a software reset (DEBUG_REG[SW_RESET]=1) can be executed to start the loaded program. The JTAG interface can be disabled and invoked again at any time during program execution. See "JTAG-SDI" on page 41.

8.3 BOOT FROM UART

The next step for the boot sequence is to check whether it will proceed with the "UART boot protocol" (refer to Figure 15)

UART Configuration

The UART is configured for 9600 baud, no parity and uses the I/O pins as presented in Table 23:

Table 23: UART interface assignment at start-up

Pin name	Port number
UTX	P0[0] output
URX	P0[1] input with pull-up

If **UTX=0** on the rising edge of RSTn, the UART boot protocol shown in Table 24 is executed. If parameters LEN and CRC are both ACKnowledged, the loaded program is executed by a jump to **0x8080**.

If the "UART boot protocol" fails, the UART interface state machine returns to the IDLE state, waiting for a new sequence.

The boot protocol still supports one wire UART mode.

Table 24: UART boot protocol

Port	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte5.N	ByteN+1	ByteN+2
UTX	STX=02				ACK=06 NACK=0x15		CRC (XOR over CODE)	-
URX		SOH=01	LEN_LSB	LEN_MSB		CODE		ACK=06

8.4 BOOT FROM SPI OR QSPI

The boot program tries to access an SPI (or QSPI) DataFlash through the following pins:

Table 25: SPI2 interface assignment for DataFlash

(Q)SPI DataFlash Pin	SC14452 Port Number and configuration
MOSI	P0[12] : output pin
MISO	P0[13] : input pin with pull-up
SCLK	P0[6] : output pin
CSn	P0[3] : output pin
WPn	P0[11] : output pin, drives "0", i.e. select Write Protect
HOLDn	P0[14] : output pin, drives "1", i.e. don't hold

- SPI MODE 3 timing is used.

- The selected speed is XTAL/4 = 2.592 MHz

- Byte mode is selected.

If no programmed (Q)SPI can be found, the SPI interface is switched back to the port reset values and the SPI is disabled and continues with the next boot source, which is the external memory.

If on-chip RAM above 0x100.0000 (16M) is selected, the response bytes 0-7 are copied to address 0 in on-chip memory for debugging purposes and the boot program will wait (for a hardware reset) in an endless loop.

SPI Configuration

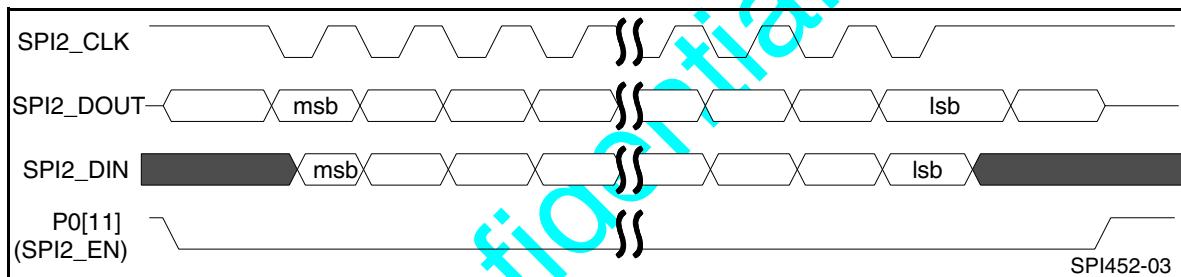


Figure 14 SPI master mode 3

Command format

The first step of the boot code is the “QSPI Reset Sequence” in order to reset any potential QSPI device into single SPI mode. The command sequence is presented in Table 26.

Table 26: QSPI Reset Sequence

Action	Description
-	Time between power up and the first (READ) command must be at least 15ms.
Command: 0xFF	Exit from Quad I/O mode
Command: 0xFFFF	Exit from Dual I/O mode
Command: 0xAB	Exit from the potential “Deep Power Down” mode
Delay for at least 30 usec	Wait the device to exit from the “Deep Power Down Mode”

After the “QSPI Reset Sequence” the boot code tries to

read the SPI memory by trying different possible READ commands, shown in Table 27. Three commonly used read command opcodes are used: 03h, 0Bh, E8h. With these commands the majority of DataFlash devices can be accessed. Each command has a corresponding number of dummy bytes.

If there will be no response on one READ command the boot code will proceed with the next one. If all three READ commands fail then the boot code will start again from the “QSPI Reset Sequence”. There will be three (3) trials in total. If all three trials of this command sequence fail, then the “booting from SPI” procedure fails and the boot code continues with the next boot source, which is the external memory.

Table 27: Three READ command formats to access various types of DataFlash

Signal	Byte 0	Byte1	Byte 2	Byte 3	ByteN...	ByteM...
Description	OPCODE	Address byte 0	Address byte 1	Address byte 2	Dummy bytes	SPI device Response (See Table 28)
MOSI	0x03	0	0	0	(none)	-
	0x0B	0	0	0	0	-
	0xE8	0	0	0	0,0,0,0	-
MISO	-	-	-	-	-	(Data)

DataFlash command response

See Table 28. The first two bytes in the DataFlash must consist of a unique string. If this string ("pP") is recognized, the following bytes are interpreted as start address and length of the loader. The loader contents

B0 to BN, starts from byte 8 and is copied for length L(1-0) to the specified location S(3-0) in on-chip RAM. The loader must be located between 0-16M, being the maximum code address space of the CR16C+.

Table 28: Response on SPI2_DI from SPI DataFlash

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte8	Byte N
Unique String	Loader Start Address Bits31-24	Loader Start Address Bits23-16	Loader Start Address Bits15-8	Loader Start Address Bits 7-0	Loader Length Bits15-8	Loader Length Bits 7-0	Loader Byte 0	Loader Byte N	
'p' (0x70)	'P' (0x50)	S3	S2	S1	S0	L1	L0	B0	BN

8.5 BOOT FROM EXTERNAL MEMORY

Boot from non volatile external memory (ROM, FLASH) is done via ACS0 (mapped on pin P0[11]) for normal asynchronous access.

The boot program initially assigns a minimum set of addresses AD[0-12] (while AD[13-23] are set as input with pull-down) and control signals according to Table 29 and reads the header from locations 2 to 8 in external memory at address 0x20000, being the first free external memory area.

The header format is shown in Table 30. The first two bytes is a branch to address to were the real program starts.

The second pair must consist of a unique string ("pP").

PROGRAM_SIZE[0-4]. If within correct range from 64k to 16M, EBI_ACS0_CTRL_REG is initialized as follows:

- MEM_SIZE = PROGRAM_SIZE.
- MEM_TYPE = FLASH memory type.
- REG_SELECT = Timing set 0.

Note that only those address lines are assigned that are required to access external memory. E.g. for 64k only AD[15-13] are assigned. If more address line are required, e.g for external address decoding, the external program must enable them.

PROGRAM_START_ADDRESS. If within 16M, bits 23-16 are copied to EBI_ACS0_LOW_REG[23-16], the other address pins remain input pull-down.

Note that the START_ADDRESS must be a multiple of the PROGRAM_SIZE. Refer to chapter "Chip selection" on page 85 for further information.

To avoid bus conflicts due to other ACS chip selects connected to port pins the boot code set all possible ACSx PPA assignments to **input pull-up**, except for AD13 to AD19. These pins are set to either Pull-down or driving address bus according Table 29 and Table 30. PAOUTn and PAOUTn stay in their reset state to preserve power, pull-down. **So when booting from external FLASH, ACS[1-4] shall not be mapped to AD23-AD13, P2[0] and P2[1], P0[7-10]**

Next the boot jumps to user code indicated by the PROGRAM_START_ADDRESS and branches to PROGRAM_START_ADDRESS +10 for user code execution (in start.asm). The user code must be linked to the PROGRAM_START_ADDRESS.

If the start-up header has illegal values, the whole header is copied to address 0 in on-chip memory for debugging purposes and the boot program will wait in an endless loop.

The PC_START_REG is not modified and stays 0xFEFO, pointing to the boot ROM. A software reset will therefore result in a clean boot from external memory.

PC_START_REG shall not be modified in the user program when executing from external memory because the memory controller is reset upon a SW reset and must be initialised by the boot code to enable the chip selects

Table 29: Initial External memory pin assignment

Pin name	Port number and comments
AD[23-13]	P1[6]/AD23 to P1[0]/AD17 and P2[10]/AD16 to P2[7]/AD13 are pull-down inputs. Initial addressing range using only AD1 to AD12 (8kByte).
ASC0	P0[11]/ASC0 is output. EBI_ACS0_CTRL_REG=0x41. Memory type is FLASH with block size 64KByte. EBI_ACS0_LOW_REG[23-16] = 0x2. Initial start address is 0x20000 (128k).
BE1N, BE0N	P1[14-13] are pull-up inputs. FLASH with Byte enables can not be used.
Others	All P0[] port pins (besides P0[11]) to P0[7] are pull-down inputs.

Header Example. For a 1Mbyte device to start at 0x100000, header at 0x20000 looks like this:

```

0x20000 = 0x10E5      // br *+10
0x20002 = 0x5070      // "pP" = 0x70, 0x50
0x20004 = 0x0500      // 5 = 1Mbyte
0x20006 = 0x0000      // 0x10.0000
0x20008 = 0x0010

```

Table 30: External memory header at 0x20000

Byte Address	Value	Comments
0	0xE5	CR16C+ instruction: br *+10
1	0x10	
2	'p' (0x70)	Unique String
3	'P' (0x50)	
4	0	
5	PROGRAM_SIZE: (value will be copied to EBI_ACS0_CTRL_REG[4-0]) 1 = 64kbyte. (AD[15-13] assigned) 2 = 128 kByte (AD[16-13] assigned) 3 = 256 kByte (AD[17-13] assigned) 4 = 512 kByte (AD[18-13] assigned) 5 = 1 Mbyte (AD[19-13] assigned) 6-17 = not allowed. Note: P1[3] / AD20 to P1[6] / AD23 and P0[7] / AD20 to P0[10] / AD23 kept to Input pull-down	
6	PROGRAM_START_ADDRESS[7-0]	
7	PROGRAM_START_ADDRESS[15-8]	
8	PROGRAM_START_ADDRESS[23-16] (value will be copied to EBI_ACS0_LOW_REG[23-16])	
9	0	
10 to N	User Program code	

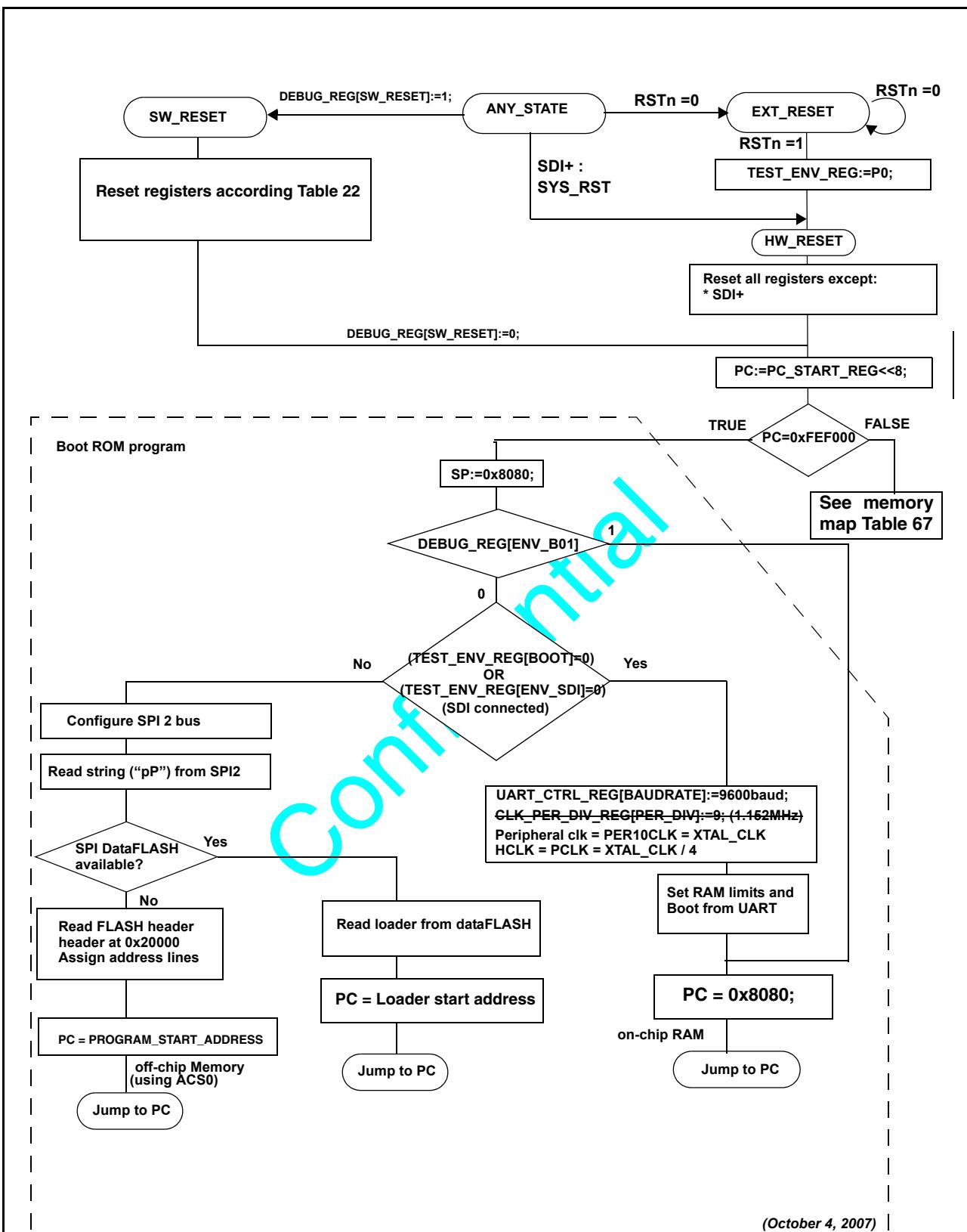


Figure 15 SC14452 Startup and boot code flow diagram

9.0 JTAG-SDI

The JTAG Serial Debug Interface (SDI+) is an on-chip debug interface, compliant to the Nexus 5001 Forum™ Standard for Global Embedded Processor Debug Interfaces (Revision 1.0, 15th December 1999), Class 1+. It implements all basic functions to allow application development and testing with the chip already installed in the final target application.

The SDI+ has the following features:

- Communication via One Wire Interface (OWI) through pin JTAG.

- Up to eight hardware breakpoints/watch points, triggered upon instruction execution or data access.
- Single-Stepping of instructions.
- Access to memory mapped resources by holding the CR16C+.
- Access to CR16C+ internal registers.
- Client break support for DIP and both Gen2DSPs

The SDI+ interface drivers software is available in [AN-D-073.zip](#)

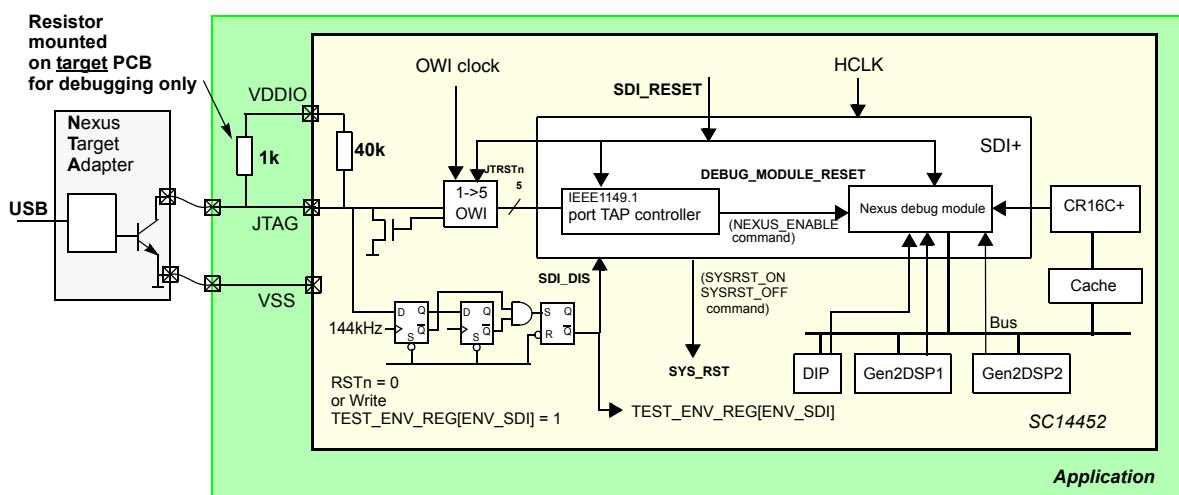


Figure 16 SDI debug module

On-chip communication to standard IEEE1149.1 (JTAG) TAP controller through signals TMS, TDI, TDO, TCK, TRSTn, TEVOn, RDYn

The SDI can take over the CR16C+ or DMA execution at any time. The SDI has the highest priority in bus arbitration over DMA and CR16C+ operation. See also "DMA CHANNEL arbitration" on page 97.

Enabling the SDI, Hot insertion

The SDI can be enabled at any time when the CR16C+ or DMA is running, by pulling the JTAG pin low for at least 15 us. This hot insertion feature is useful when debugging is required after the application has ran for a long time and software problems show up.

Enabling the SDI consumes power and must therefore be avoided in a normal application by keeping the JTAG pin either unconnected and/or pull-up to VDDIO

Disabling SDI

The SDI can be disabled in two ways:

- JTAG pin is disconnected or kept '1' during and after the RSTn pin is pulsed low.
- Writing a '1' to TEST_ENV_REG[ENV_SDID] followed by an inactive '1' state of the JTAG pin within 5 us. The JTAG SDI+ is enabled again if the JTAG pin is kept '0' for at least 15 us.

Start-up with no program loaded.

If no program is loaded in external FLASH or RAM or no EEPROM or no Serial FLASH is available to boot from during development, the JTAG must be kept low while RSTn is low and kept low for 20 us after the RSTn pin is set to '1'. This prevents uncontrolled program execution. By this way the boot code will enter the "right" branch of the boot ROM code flow diagram (See Figure 15) and starts "booting from UART" because TEST_ENV_REG[ENV_SDID] is 0. Because no external UART is connected, the CR16C+ waits in an endless loop which allows the SDI to become bus master by setting the DGCR[DBR] bit in the SDI+. Refer to [AN-D-073](#) for more details.

From this point CR16C+ program execution can be directed to execute the boot procedure from internal FLASH or ROM. (Refer to "Boot Program" on page 36 point 3) on how to proceed on this point.

Clock selection

The OWI clock can be enabled by CLK_JOWI_REG[SW_OWICLK_C_EN] and its frequency can be defined by programming the CLK_JOWI_REG[SW_OWICLK_C_DIV]. The default value (xtal selected divide by 1) shall only be modified by the DCTmon.exe tool. The SDI clock is equal to the CR16C+ clock HCLK and is application dependent.

10.0 DIP

The Dedicated Instruction Processor (DIP) in conjunction with the CR16C+ takes care of MAC-CSF functions as depicted in Figure 18. The S, A, B, and Z field order is controlled from the sequencer and 16 and 32 bits R-CRC generation is done in the BMC. The DIP handles slot and frame timing to the RF front end interface without CR16C+ interruption. The instructions for the DIP are stored in the Sequencer RAM. All necessary data and control information can be stored in advance to control a complete (10ms) frame. Consecutive instructions are performed at the symbol clock rate.

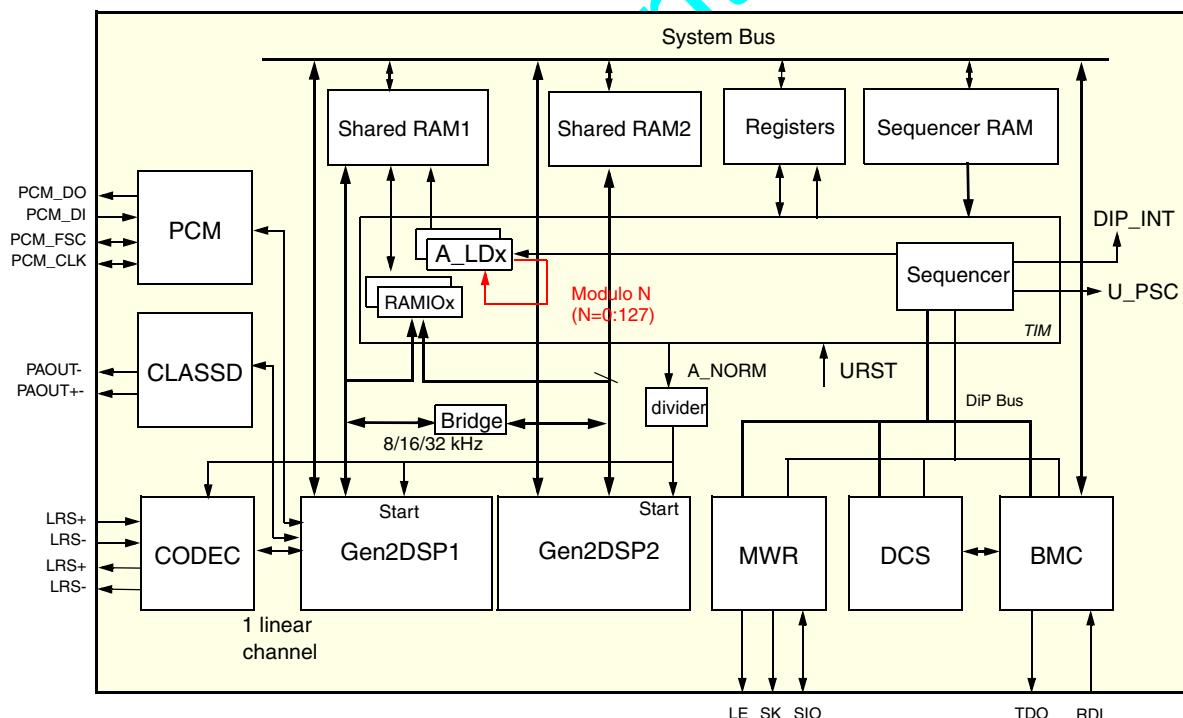
The DIP issues instructions to BMC, DCS (Cyphering) and Microwire interface and supports automatic Gen2DSP ADPCM and BMC pointer handling toward shared RAM.

For detailed information refer to "[AN-D-140 Dedicated Instruction Processor](#)"

Features

- Instructions for fix DECT full, long and double slot formats or user programmable Rx and Tx slot with one bit resolution.

- Instructions for fixed or programmable S-field.
- Instructions for fixed or programmable A-field with automatic 16 bit R-CRC generation and detection.
- Instructions for protected and non protected B-field formats with programmable 16 R-CRC and 32 bit B-CRC
- BMC Bit Clock divider factor 1, 2 and 4.
- Automatic ADPCM and BMC pointer handling (4xRx, 4xTx) to shared RAM for minimum delay of full slots, 640 bits long slots.
- Flexible slot format with Modulo 0:127 operation
- Instructions to synchronise the audio path from BMC to Gen2DSP and Codec. (A_NORM)
- Instructions to synchronize DIP frame to 8kHz PCM slave interface. (WSC)
- Bit accurate RF transceiver control over Micro-Wire interface.
- Full HW multiframe counter.



January 16, 2006

Figure 17 DIP blockdiagram

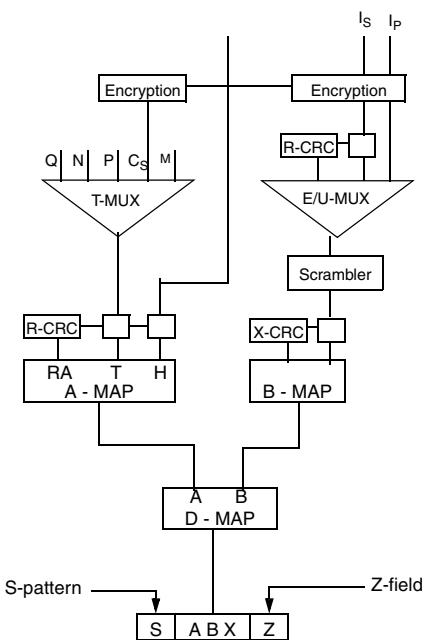


Figure 18 SC14452 MAC functionality

10.1 DIP ON/OFF CONTROL

The SC14452 CR16C+ controls the DIP with the DIP_CTRL_REG[URST] bit. With URST is set to 0 the DIP executes the instructions stored in sequencer from address 0x100.0002 at a clock rate of 1.152MHz in case of the DECT mode, 1.024MHz in case of 8/9 mode. (DIP_CTRL2_REG[EN_8DIV9])

10.2 DIP SEQUENCER RAM

The 512 16-bit words of internal DIP program RAM contain the DIP executable code. The executable code inside this RAM can be read and written by the CR16C+ at any time. An internal bus arbiter takes care of bus conflicts if the DIP and CR16C+ access the Data RAM at the same time.

10.3 DIP INTERRUPT GENERATION AND POWER CONTROL

DIP instructions <U_INT0>, <U_INT1>, <U_INT2>, <U_INT3> and <U_VINT> <X> generate a DIP_INT interrupt to the CR16C+. The corresponding interrupt vector can be read from register DIP_CTRL_REG[3-0]. <U_VINT> <X> set bits 3 to 0 to value <X>. <U_INT0> set bit 0, <U_INT1> sets bit 1 etc.

If the PRESCALER mode bit is set to 1 and the DIP executes the <U_PSC> command, the HCLK is divided depending on the XDIV bit and the CLK_XDIV0_REG, CLK_XDIV1_REG (refer also to Figure 6 and Section 4.6.2). This clock division remains active after the <U_PSC> command is executed until the DIP executes a <U_INTx> or <U_VINT> command or the prescaler mode bit is set to 0.

The DIP_CTRL_REG[DIP_BRK_INT] bit is set by the DIP if the BRK command is executed. At this time also a DIP interrupt is generated at the CR16C+. After execution of the break command the DIP program counter is not incremented until a 1 is written to DIP_CTRL_REG[DIP_BRK_INT]. The CR16C+ can read the program counter from the time the BRK command is executed until the break bit is cleared.

10.4 INTERFACING TO GEN2DSPS

See Figure 17.

Instruction <A_NORM> starts 8/16/32 kHz generation and both Gen2DSPs have full access to their own shared RAM.

10.5 MINIMUM DELAY

For minimum delay voice applications, the Gen2DSPs must use RAMIO registers DSP_RAMx_IN_REG and DSP_RAM_OUTx_REG so that the DIP handles the RAM access.

Both Gen2DSPs have access to the same RAMIO registers. It is the programmers responsibility to assign the registers to the right Gen2DSP.

The <A_LDRx>/<A_LDWx> instructions sets pointers to the buffers in shared RAM were the RAMIO registers will be written or read from. Every 4kHz (for 32kbit/s) or 8 kHz (64kbit/s) the DIP (TIM) reads from DSP_RAMx_IN_REG and writes one byte to DSP_RAM_OUTx_REG. With the same rate the Gen2DSP must access the RAMIO registers.

The TIM block takes care of copying these registers to circular buffer pointed to by <A_LDWx> pointer and auto increment these pointer with a modulo N (0:127). N= 39 for 32kbit/s ADPCM and N=79 for 64kbit/s ADPCM).

For minimum delay the BMC is able to read and write one location after the actual ADPCM pointer using instructions:

- <B_BRFU>, <B_BTFU> with parameter 0xFC for 32kbit/s ADPCM: 32S+64A+320B+4X+4Z (Buffer modulo 39 bytes).
- <B_BRLU>, <B_BTU> with parameter 0xFC for 64kbit/s ADPCM: 32S+64A+640B+(32BCRC)+4X, 4Z. (Buffer modulo 79 bytes). 32 bits CRC can be switched on/off with B_RC3[EN_32B].

10.6 MUTING

The SC14452 can sense the correctness of the A-field CRC by setting the SENSA bit to 1. If an erroneous frame is received the new B-field will not be written and the last received B-field will be repeated in order not to disturb the ADPCM sound.

If the SENSA bit is kept to 0 and an A-field error is detected, the CR16C+ can decide to initiate mute sequence to the Gen2DSP.

In case of sync loss the B-field is muted if the SENSS bit is set to 1.

Command <B_BTFM> transmits '0's or 1s in the B-field in order to mute the outgoing speech. If BMC control bit MUTELEVEL is 0 then '0's are transmitted, else 1s.

10.7 SLOT CONTROL INFORMATION

Receive and transmit burst timing on the RF interface is controlled by the DIP. With DIP command <B_RC>, the slot control information can be read from any Data RAM location (see Table 31). Control information is read before every received and transmitted slot. With DIP command <B_WRS> the MAC status information on received slots is written into the data RAM of the SC14452 (see Table 32). DIP command <B_WRS d_offset> updates all these parameters. At the end of every received time-slot an interrupt instruction <U_INTx>, <U_VINT> or <U_NMI> can be set in the sequencer RAM to enable the microprocessor to read out the appropriate status data.

Table 31: Slot control information in Data RAM

NAME	TYPE	FUNCTION
S_err[3..0]	Binary	Maximum number of errors allowed in non-masked bits of S-field pattern S[8..31]
Inv_RDI		Inversion of received data input
Inv_TDO		Inversion of transmit data output
SENSA		B field data is not written in the case when the A-field CRC is incorrect
SENSS		B field data is not written in the case when the BMC loses synchronisation.
PP/FPn		PP mode selected
Mask[3..0]	Binary	Mask S-field pattern[15..8] in error calculation over pattern [8..31]
Slide[3..0]	Binary	Mask S-field pattern[15..8] in Sliding error calculation over pattern[15..8]
ADP	Binary	Upon A-field CRC error, no phase adjustment will be made
WIN[3..0]	Binary	Defines maximum phase shift. If phase shift is between \pm WIN symbols, the IN_SYNC bit will be set
VOL[5..0]	Binary	TDO gaussian output volume control. Default '100000' after reset.

Table 31: Slot control information in Data RAM

NAME	TYPE	FUNCTION
M[1..0]	Code	TDO output mode: 00 = power down 01 = Gaussian output BT=0.5 10 = power down 11 = Gaussian output BT=1. Fixed mid level of 0.9V if idle.
FR_nr[3..0]	Binary	Frame number 0..15
MFR[23..0]	Binary	Multi frame number
IV[28..63]	Code	Encryption unit initialisation vector
CK[0..63]	Code	Encryption unit cypher key
ENC_OFF	Binary	Disable/enable Encryption
SCR_OFF	Binary	Disable/enable Scrambling
MUTELEV	Binary	Used with B_BTFM command. If 1 transmit 1's, 0 transmit 0 in B-field.
EN_32B	Binary	Enable CRC32 in B_BT LU and B_BRLU or Enable Padding in B_BT LP and B_BRLP

10.8 SLOT STATUS INFORMATION

At the end of every received slot the status is updated in the Data Memory.

Table 32: MAC Slot status information in Data RAM

NAME	TYPE	FUNCTION
ADC[5..0]	Binary	Maximum RSSI peak value measured during PD0.
ANT_SEL		0 = antenna 1 is selected. 1 = antenna 2 is selected
IN_SYNC		Set to 1 when the S-Field pattern falls within the defined window size
A_CRC		Set to 1 when the A-field CRC is correct
X_CRC		Set to 1 when the X-field CRC is correct
ZACK		Set to 1 when the Z-field = X-field
Bn-CRC		Set to 1 when protected B field R-CRC of block n is correctly received. n=0..9
SL_err[3..0]	Binary	Report number of sliding errors over unmasked S-field bits 8 to 15.

Table 32: MAC Slot status information in Data RAM

NAME	TYPE	FUNCTION
TAP[4..0]		S-field phase information, with 1/9 DECT bit accuracy
Phase[7..0]	2's Compl	S-field pattern phase error, with DECT bit accuracy
DC[5..0]	Binary	DC offset information
Offset[15..0]	Binary	Offset counter from <B_SR> command upto received S-field

10.9 SLOT SYNCHRONISATION

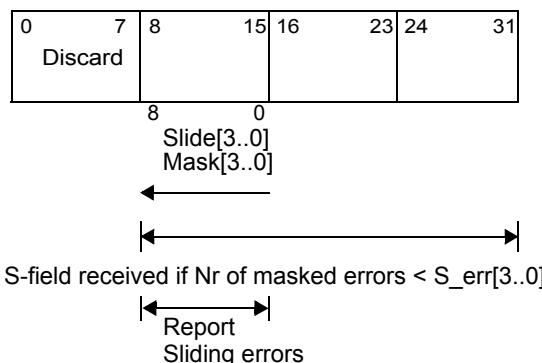
The SC14452 BMC is capable of detecting the S-field pattern. To be able to lock to the received frame in the PP mode the microprocessor enables the slot counter to be preset at S-field detection. If the preset of the slot counter occurred once, the preset is disabled. The microprocessor can enable this preset again. If disabled the SC14452 will only search for the S-field pattern in a predefined window repeated every 10ms after the first S-field pattern has been detected. In lock the SC14452 can accept a packet phase shift up to a programmed maximum window size. This window is programmable with control nibble WIN[3:0] from 0-15 symbol periods. This phase shift information is stored in PHASE[7..0] in two's complement notation in the MAC status bytes. If the phase shift is more than the programmed window the "in_sync" bit will not be set. Once enabled by the microprocessor the SC14452 will automatically adjust the frame timing. If the ADP bit is set 1 then the automatic phase adjustment will only be done if the A-field CRC is correct. If ADP is 0 the phase adjustment will be made regardless of the A-field result. Choosing to write the B-field data depending on the A-field CRC result is selectable using the control bit SENSA. If the phase shift between the internally generated frame and received frame is less than 2 symbol periods frame timing is adjusted with ± 1 symbol period at the end of the frame. The update will be maximum ± 2 symbols if this phase shift is 2 or more symbols. However the microprocessor can take over control and program phase jumps of ± 1 symbol period instructions for the next frame if appropriate and disable the automatic phase jumps.

If in an unlocked state an S-field is received, the slot counter is preset and the current WNT is terminated. If a <JMP1> instruction is then executed a jump will be executed.

10.10 S-PATTERN CORRELATOR

With the control bytes it is possible to define the criteria

for the S-field to be detected.


Figure 19 S-field pattern

The first eight bits of the S-field bit sync pattern S[0..7] are ignored. On the next eight bits a mask register Mask[3..0] masks bits 15 to 8. ("0011" unmasks bit 13,14,15.). If the total number of errors in bits 16 to 31 plus the unmasked bits 8 to 15 is less than a number programmed in the data RAM (S_err[3..0]), the S-field is received correctly and the phase shift can be calculated. S-field bits 15 through 8 are also used to check for sliding errors. These are reported in the MAC unit status register SL_err[3..0]. Mask register Slide[3..0] masks S-field bits 15..8. ("0011" unmasks bits 8,9,10). Only unmasked bits are taken in to account for the sliding error calculation. A sliding error occurs when a neighbouring un-synchronized station sends out a packet that slides into the received packet.

The received S-field is compared with the BMC_RX_SFIELD_REG. The transmitted S-field data can be transmitted as a fixed pattern with <B_ST>, from RAM with <B_ST2> 00, from BMC_TX_SFIELD_REG with <B_ST2> 01 to 04.

10.11 DCS BLOCK ENCRYPTION

The Key Stream Generator (KSG), used to encrypt A and B-field information, can be switched on or off by the microcontroller via instructions in the sequencer RAM. The KSG initial state is stored in the internal data memory. CK[63..0] is the cypher key information. The MAC unit recognizes Cs type messages in the A-field data and encrypts or decrypts only these messages. Encryption can be temporarily disabled with bit ENC_OFF set to 1. The encryption engine remains active but no data is encrypted.

10.12 SCRAMBLER

Scrambling and descrambling, using the frame counter bits FR_nr[3..0] to initialize the scrambler and de-scrambler, is performed on the B-field.

10.13 R-CRC, X-CRC AND Z FIELD

Generation and checking of the 16-bit CRC is performed on the A-field data (A_CRC will be set if CRC is correct). For the B-field, X-CRC and Z-field are generated and checked for transmission and reception

respectively. A, X and Z field results are stored in the internal data memory (A_CRC, X_CRC and ZACK bit). Full slot protected B-field format is supported and 16 bit CRC is calculated for every 64 bits of data.

Instruction <B_RON> and <B_RINV> are used to manually start and stop X-CRC.

<B_NIC> in conjunction with <B_BRFP>/<B_BTLP> supports 16bits R-CRC and <B_NIC> with <B_BRLP>/<B_BTLP> supports 32 bits B-CRC calculation. according ETS EN 300 175-3.

10.14 ADPCM SUPPORT

In order to support minimum delay for upto four voice links the following ADPCM commands are used: <A_NORM>, <A_LDWx>, and <A_LDRx>. The other <A_xxx> have no function any more, but can be kept in DIP sequencer code if software reuse requires this. See Table 33.

10.15 DIP COMMANDS

Table 33: DIP Commands overview (Continued)

Command	OPCODE	Description
BR	0x1	Branch Always
BRK	0x6E	Break for CR16C+
JMP	0x2	Jump to subroutine (4 deep)
JMP1	0x3	Jump conditionally to subroutine
RTN	0x4	Return from subroutine
WNT	0x8	Wait until start of time slot(s)
WT	0x9	Wait N time symbols (Immediate or indirect via LD_PTR2)
RFEN	0xB	Enable RF clock
RFDIS	0xA	Disable RF clock
BK_A	0x0E	Set memory bank for ADPCM channel
BK_A1	0x05	Set memory bank for ADPCM channel 1
BK_A2	0x1E	Set memory bank for ADPCM channel 2
BK_A3	0x15	Set memory bank for ADPCM channel 3
BK_MA	0x16	Set main memory bank for ADPCM channels
BK_C	0x0F	Set memory bank for burst mode controller, micro wire and encryption unit

Table 33: DIP Commands overview (Continued)

Command	OPCODE	Description
BK_MC	0x1F	Set main memory bank for burst mode controller, micro wire and encryption unit
BCPV	0x4B	Programs WNT-enter pre-set value
SLOTZERO	0x0D	Reset internal CLK100 timer
EN_SL_ADJ	0x2C	Enable DECT bitclock phase adjustment
WNTP1	0x7	Increment symbol counter
WNTM1	0x6	Decrement symbol counter
WSC	0x48	Wait for external synchronisation on PCM IF.
LD_PTR	0xC	Load pointer for indirect Data Ram access
LD_PTR2	0x1C	Load pointer for parameters of WT, P_LDH, P_LDL
UNLCK	0x28	Enter "PP unlocked" mode
A_NORM	0xC5	Start Gen2DSP main counter and determines start of ADPCM tick.
A_LDR	0xC6	Initialise start of ADPCM channel read buffer
A_LDW	0xC7	Load ADPCM channel write address
A_LDR1	0xCE	Initialise start address of ADPCM channel 1 read buffer
A_LDR2	0xD6	Initialise start address of ADPCM channel 2 read buffer
A_LDR3	0xDE	Initialise start address of ADPCM channel 2 read buffer
A_LDW1	0xCF	Initialise start address of ADPCM channel 1 write buffer
A_LDW2	0xD7	Initialise start address of ADPCM channel 2 write buffer
A_LDW3	0xDF	Initialise start address of ADPCM channel 3 write buffer
A_TX	0x4A	ADPCM channel assignment to BMC Tx slots
A_RX	0x49	ADPCM channel assignment to BMC Rx slots.

Table 33: DIP Commands overview (Continued)

Command	OPCODE	Description
B_ST (B_TX)	0x31	Transmit any data from Data Ram
B_ST2	0x21	Transmit fixed ETSI S-field in PP or FP mode
B_PPT	0x22	Transmit prolonged preamble
B_AT	0x32	Transmit A-field
B_AT2	0x37	Transmit A-field and A-field CRC (R_A)
B_BT	0x34	Transmit B-field
B_BTFU (B_BT2)	0x25	Transmit full slot unprotected B-field, B-field CRC (X) and Z-field (Z)
B_BTBM	0x23	Equal to B_BTFU but B-field contains either 0: (Mutelevel=0) or 1: (Mutelevel=1)
B_BTFP	0x35	Transmit full slot protected B-field, B-field CRC (X) and Z-field (Z)
B_BTDU	0x71	Transmit double slot unprotected B-field, B-field CRC (X) and Z-field (Z)
B_BTDP	0x72	Transmit double slot protected B-field, B-field CRC (X) and Z-field (Z)
B_BTBLU	0x95	Transmit 640 bits slot unprotected B-field: 32S, 64A, 640B, (32BCRC,) 4X, 4Z. (BCRC on/off with B_RC3[EN_32B])
B_BTLM	0x93	Equal to B_BTBLU, but B-field contains either 0: (Mutelevel=0) or 1: (Mutelevel=1)
B_BTLP	0x75	Transmit 640 bits slot protected B-field: 32S, 64A, 8x(64B/16CRC), (32bits padding), 4X, 4Z. (Padding bits on/off with B_RC3[EN_32B])
B_SR	0x29	Receive S-field
B_AR	0x3A	Receive A-field
B_AR2	0x3F	Receive A-field and A-field CRC (R_A)
B_RON	0x2F	Transmit A-field CRC or Receive and compare A-field
B_RINV	0x2E	Invert last bit A-field R_CRC

Table 33: DIP Commands overview (Continued)

Command	OPCODE	Description
B_BR	0x3C	Receive B-field
B_BRFU (B_BR2)	0x2D	Receive full slot unprotected B-field, B-field CRC (X) and Z-field (Z)
B_BRFP	0x3D	Receive full slot 320 bits protected B-field, B-field CRC (X) and Z-field (Z)
B_BRFD	0x2A	Equal to B_BRFP but no B-field data is stored
B_BRDU	0x79	Receive double slot unprotected B-field, B-field CRC (X) and Z-field (Z)
B_BRDP	0x7A	Receive double slot protected B-field, B-field CRC (X) and Z-field (Z)
B_BRLU	0x9D	Receive 640 bits slot unprotected B-field, 32S, 64A, 640B, (32BCRC,) 4X, 4Z. (BCRC on/off with B_RC3[EN_32B])
B_BRLD	0x9A	Equal to B_BRLU but no B-field data is stored
B_BRLP	0x7D	Receive 640 bits slot protected B-field: 32S, 64A, 8x(64B/16CRC), (32bits padding), 4X, 4Z (Padding bits on/off with B_RC3[EN_32B])
B_XR	0x2B	Receive X-field
B_WB_ON	0x65	Write zeroes in B-field receive buffer if A-CRC is 0
B_WB_OFF	0x64	Disable B_WB_ON command
B_WRS	0x39	Write BMC status information to Data Ram
B_WRS2	0x38	write synchronisation counter to RAM
B_RC	0x33	Load BMC control information from Data Ram in BMC
B_XOFF	0x26	X-CRC off
B_XON	0x27	X-CRC on
B_XT	0x24	Transmit X-CRC
B_RST	0x20	Set BMC in power down mode.
B_DIV1	0x4F	Set BMC clock to 1.152 MHz. (default)

Table 33: DIP Commands overview (Continued)

Command	OPCODE	Description
B_DIV2	0x4E	Set BMC clock to 1.152/2 MHz.
B_DIV4	0x4D	Set BMC clock to 1.152/4 MHz.
D_LDK	0x50	Load encryption data from Data Ram in DCS unit
D_PREP	0x44	Preprocess encryption data in DCS
D_PREP2	0x45	4 times faster D_PREP
D_WRS	0x5F	Write current encryption status to Data Ram
D_LDS	0x57	Load intermediate encryption status from Data RAM
D_RST	0x40	Reset keystream generator
M_WR	0xB9	MicroWire Write to SIO
M_RD	0xBC	MicroWire Read from SIO
M_RST	0xA9	Stop commands M_WR and M_RD
M_INI0	0xA0	Set SK,SO start/end levels to 0, SK falling edge
M_INI1	0xA1	Set SK,SO start/end levels to 1, SK falling edge
M_CON	0xAA	Set Microwire clock speed, edge and level
MEN1n	0xA4	Reset pin LE to 0
MEN1	0xA5	Set pin LE to 1
P_EN	0xE9	Enable PD pins from 3-state to active 1
P_LDH	0xED	Set marked PD pins to 1. (Immediate or indirect via LD_PTR2)
P_LDL	0xEC	Set marked PD pins to 0. (Immediate or indirect via LD_PTR2)
P_LD <n>	0xE8	Set PD pins to <n>
P_SC	0xEA	Program PIN control register
U_INT0 U_INT1 U_INT2 U_INT3	0x61, 0x6B, 0x6D, 0x6F	Generate interrupt to CR16C+
U_PSC	0x60	Set CR16C+ SCK prescaler (refer to Section 4.6.2)
U_VINT <x>	0x63	Generate Vectored interrupt x=1-0xF
U_VNMI <x>	0x4C	Generate Vectored NMI x=1-0xF

Table 33: DIP Commands overview (Continued)

Command	OPCODE	Description
PAG_ON	0x18	Disable Codec clocks and DIP RAM access
PAG_OFF	0x19	Enable Codec clocks and DIP RAM access
P_FAD	0xE6	Fast antenna diversity control
P_ANT	0xE7	Control of PD pins which control antenna
B_NIC	0x9F	Used in conjunction with full slot protected B field commands. This opcode disables the intermediate CRC calculations and transmit resp receive full slot protected B field with only 1X16 bit CRC at the end
JMPF	0x12	JuMP Far to bank 1 of DiP sequencer RAM
RTNF	0x14	ReTurN Far to bank 0 of DiP sequencer RAM
BR_B0	0x10	BRanch to bank 0 of DiP sequencer RAM
BR_B1	0x11	BRanch to bank 1 of DiP sequencer RAM

For more detailed information see [AN-D-140, "DIP Instruction Manual"](#).

11.0 The Radio Front-End Interface

The SC14452 can fully comply to any radio interface (including ARi1™*) providing all necessary timing signals and control information. An overview of the radio front end interface is shown in Figure 24 on page 54.

11.1 MICROWIRE™ INTERFACE

The MICROWIRE is a flexible bidirectional serial interface that allows unlimited data transfer between DIP and almost any RF synthesizer device at four different data rates. The flexibility is based on its instruction set that allows the user to choose the direction and the rate of desired data transfers.

* ARi1™ is a trademark of National Semiconductor Corporation

As shown in Figure 20 the MICROWIRE serial interface towards the DIP consists of a bidirectional data bus and a control bus. The interface towards the RF synthesizer device consists of:

- SIO, data in-output, reset value 3-state, with weak pull-down resistor. The pulldown resistor is automatically switched off if the output is driving (MWR direction). The pulldown resistor is automatically switched in off if the output is driving (MWR direction is output) or BMC_CTRL_REG[SIO_PD] =1. (The latter condition is only used for test purposes)
- SK, clock output, reset value 0.
- LE, output, reset value 0.

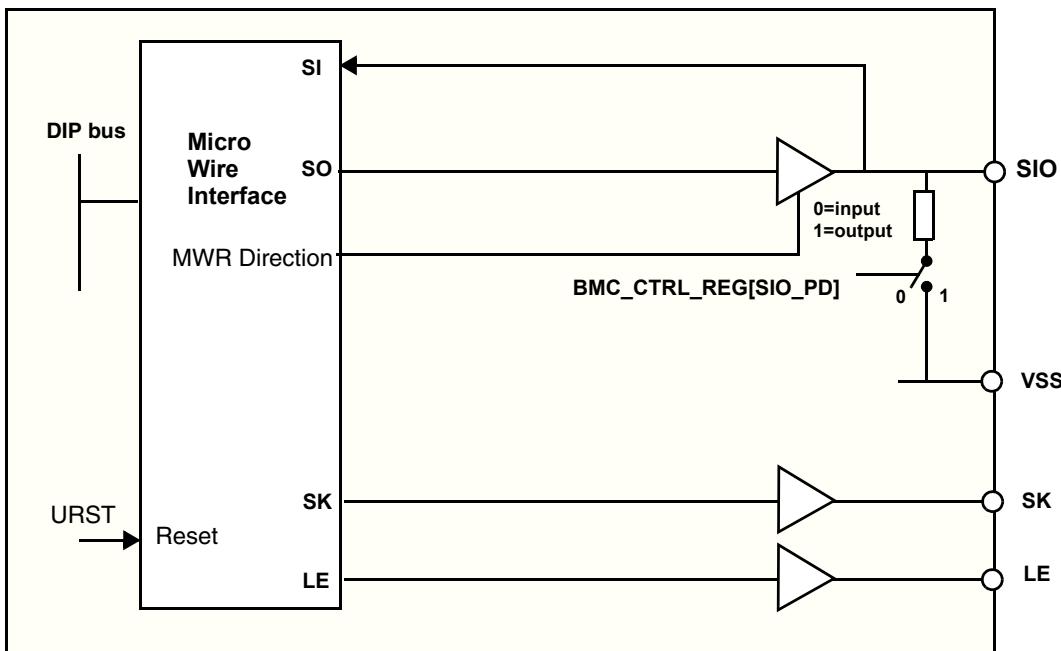


Figure 20 MICROWIRE interface

The MICROWIRE is completely reset if DIP_CTRL_REG[URST] =1. After reset, data on SIO is clocked in and out on the falling edge of SK. LE changes on the DIP bus clock TCK.

Table 34 shows the complete MICROWIRE instruction set. The instructions <M_INI0> and <M_INI1> can be used to set the active edge of SK different to the hardware reset value and the initial/final polarity of the SK and SIO outputs.

When a DIP instruction <M_WR> is used the SIO is switched to output to transfer data from data memory towards RF. In order to transfer data from the RF towards data memory, SIO can be switched to input using a <M_RD> instruction. The offset field within the <M_WR> and <M_RD> instructions denotes the data RAM address.

The data rate in both directions is programmable with the <M_CON> instruction. The parameter field of the <M_CON> instruction refers to the speed (data-rate) of the transfer. Note that the <M_CON> instruction may not be changed during the data transfer.

The data transfers are stopped when the <M_RST> instruction is used. Note that <M_RST> does not affect the end levels of SK and SIO as set by <M_INI0>/<M_INI1>.

The <MEN1> instruction sets the LE pin to 1. This value stays unchanged until the <MEN1n> instruction is executed, which sets the pin to 0. Note that after hardware reset the LE pin is set to 0.

Figure 21 shows a typical MICROWIRE timing with associated instructions. Transfer of data from RF towards data memory starts with the execution of the

<M_WR> instruction that instructs the RF to transmit data towards the MICROWIRE. As soon as a <M_RD> instruction is given, the SIO is switched to input. The moment at which the <M_RD> is given depends on the used RF device and can be timed by adding other instructions like the <WT> instruction in this example. In this case the data is clocked out by RF on rising edge of SK and will be clocked in by the MICROWIRE on falling edge of SK.

In case of a sequence with <M_INI0> and <M_INI1> instructions the data is clocked out by RF on falling edge of SK and will be clocked in by MICROWIRE on rising edge of SK.

Instead of using <M_INI0> and <M_INI1>, <M_CON> <PAR> bits 4,3,2 may also be used to set the active

edge of SK and start/end level of the SIO signal. If bit 4 is kept to '0', bits 2 and 3 have no effect.

<M_INI0> or <M_INI1> can be used in combination with <M_CON>. As soon as <M_CON> <PAR> bit 4 is set however, bits 3 and 2 determine active edge and level. Note that if bit 4 is reset again, the MICROWIRE reset state (SIO input, pull-down enabled) is active if no <M_INI0> or <M_INI1> were used.

The <M_CON> <PAR> bits may not be changed if the MICROWIRE is active, so changing parameters may only be changed before <M_WR> and <M_RD> instructions or after the <M_RST>.

Table 34: MICROWIRE instruction overview

Instruction	Parameter	Description	
M_INI0	00 ₁₆	Sets SIO, SK start/end level to 0. SIO data clocked out on rising edge of SK.	
M_INI1	00 ₁₆	Sets SIO, SK start/end level to 1. SIO data clocked out on rising edge of SK.	
M_WR	<offset> (00 ₁₆ -FF ₁₆)	SIO is switched to output and the serial clock SK is activated automatically. First the MSB is clocked out on rising edge of SK. (Active edge is programmable, see <M_CON> instruction)	
M_RD	<offset> (0016-FF16)	SIO is switched to input and the serial clock SK is activated automatically. First the MSB is clocked in on falling edge of SK. (Active edge is programmable, see <M_CON> instruction) <M_RD> may also be given before <M_WR>.	
M_RST	00 ₁₆	Stop <M_WR> and <M_RD> instructions. SIO returns to end level and SK active edge remains unchanged as set by <M_INI0>/<M_INI1>/<MCON> instructions or get default values if none of these instructions were given.	
MEN1	00 ₁₆	Sets LE pin to 1.	
MEN1n	00 ₁₆	Sets LE pin to 0.	
M_CON	Operand Bit	Description	Reset (Note 7)
	7-6		0
	5		
	4	0 = Bits 3 and 2 disabled 1 = Bits 3 and 2 enabled	0
	3	0 = Sets SIO Start/end level to '0' 1 = Sets SIO Start/end level to '1'	0
	2	0 = SIO data clocked in/out on falling edge of SK. 1 = SIO data clocked in/out on rising edge of SK.	0
	1-0	Average (Max) Frequency on SK (Note 6) 00 = 1.152 MHz (default) 01 = 2.3 MHz 10 = 4.6 MHz 11 = 10.368MHz	0

Note 6: SK = 10.368MHz / divider factor. Therefore SK duty cycle varies depending on the divider factor. If divider factor is a whole number then maximum frequency is valid (SK duty cycle is 50%). Otherwise the average frequency will decrease (SK duty cycle is unequal to 50%). For detailed information refer to the SK timing diagrams

Note 7: Reset if DIP_CTRL_REG[URST] = 1. Also <M_INI0> and <M_INI1> are reset: SIO returns to input, pull-down enabled, SIO changes on falling edge of SK.

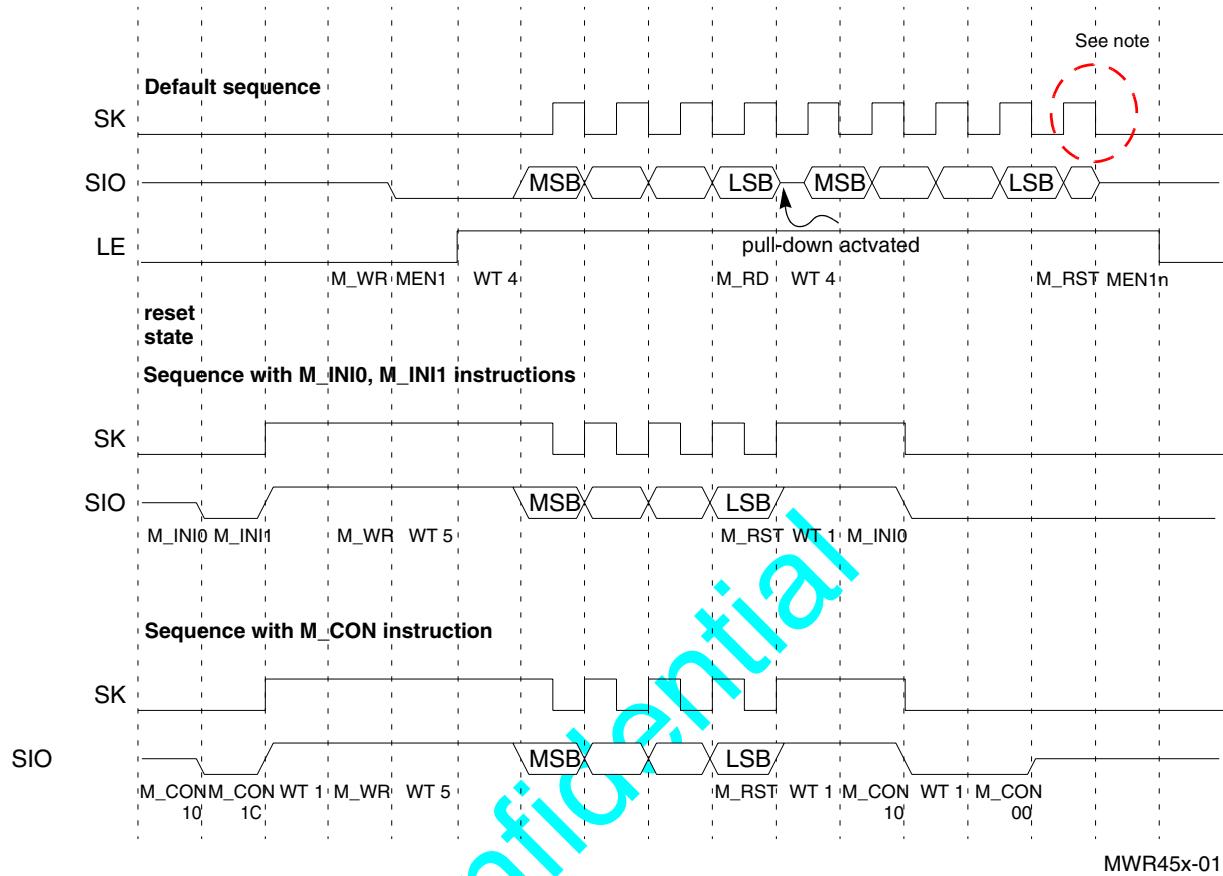


Figure 21 Typical MICROWIRE timing diagrams with 1.152 Mhz SK selection

Note 8: One or more clock pulse are active after the LSB is clocked in with <M_RD> instruction. The number of extra clock pulses depends on the selected SK frequency and the execution timeslot of the <M_RST> instruction. Refer to [AN-D-140 "DIP Instruction Manual"](#) for application examples and an overview of the extra number of cycles.

11.2 REFERENCE CLOCK OUTPUTS

The SC14452 has one single ended digital buffered xtal clock output RFCLK for the RF synthesizer. See Figure 24 on page 54 for a blockdiagram of the reference clock outputs.

For power saving, the outputs can be enabled, disabled with DIP commands <RFEN>, <RFDIS>.

11.3 PROGRAMMABLE PD CONTROL PINS

The SC14452 provides seven programmable timing signals PD5 to PD0. Each of them can be individually set or reset at any time during a frame. These timing signals can be used to activate or deactivate the front end circuitry. PD0 is only available internally to activate RSSI measurement.

After reset the PD outputs are input with a weak pull-down resistor enabled. If the PDx pins are enabled with <P_EN> command, the pins will get reset value 1 and the pulldown resistor is automatically disabled. The level can be controlled with commands <P_LD>,

<P_LDL> or <P_LDH>.

Pins PD5 to PD1 are multiplexed with general purpose ports. Refer to PPA table for reset values.

Refer to chapter "MICROWIRE™ interface" on page 49 for more details.

11.4 PD1 SWITCHING

PD1 can be used to control the sample and hold circuit in the RF module. PD1 can be programmed with instruction <P_SC> as shown in Table 35 to automatically change level and generate a DIP interrupts by setting the DIP_CTRL_REG[PD1_INT] if either the preamble of the whole S-field is detected.

If **preamble** detection is selected (RC5[DOFr]=1), PD1 changes after a 010100 or 101011 is detected at 6/9 of bit s17. If 010100 or 101011 is not found, PD1 changes at 6/9 of bit s32.

If **complete** S-field detection is selected (DOFr = 0) PD1 changes at 6/9 of bit s32.

The PD1 switching has an **automatic** reset mode which can be set with the P_SC[7]:

- 0 = <B_RST> resets the PD1 start value as programmed by the <P_SC> instruction.
- 1 = a symbol counter is started when PD1 is acti-

vated. When this symbol counter reaches the value 47 and a **complete** S-field is **not** detected the PD1 is reset to the start value as programmed by the <P_SC> instruction.

Table 35: P_SC command parameters

P_SC <par> bits 6,5	RC5 [DOFr]	Description
x0	-	PD1 is directly controlled with <P_LDL>, <P_LDH>, <P_LD> commands
11	1	PD1 goes high if S-field preamble received. Start level is 0. End level 1 must be set with <P_LDL>, <P_LDH> or <P_LD>
01	1	PD1 goes low if S-field preamble received. Start level is 1. End level 0 must be set with <P_LDL>, <P_LDH> or <P_LD>
11	0	PD1 goes high if complete S-field received. Start level is 0. End level 1 must be set with <P_LDL>, <P_LDH> or <P_LD>
01	0	PD1 goes low if complete S-field received. Start level is 1. End level 0 must be set with <P_LDL>, <P_LDH> or <P_LD>

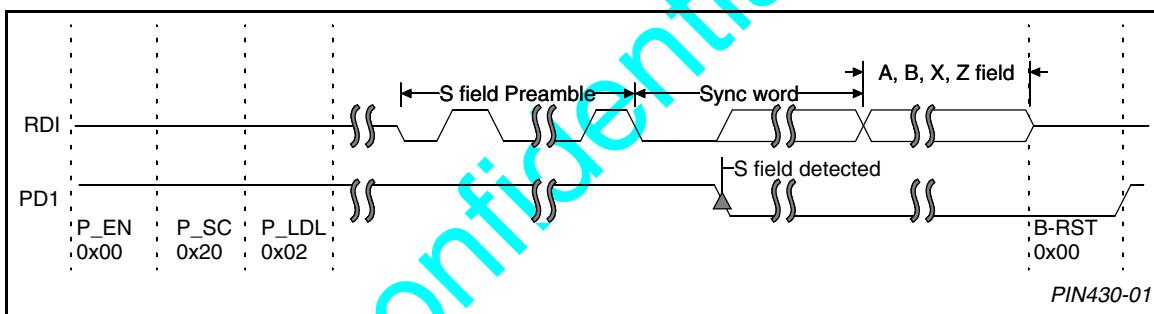


Figure 22 PD1 synchronisation with <P_SC> command

11.5 RSSI PEAKHOLD ADC

In order to select the best frequency channel with respect to signal to noise ratio, the so called Received Signal Strength Indication (RSSI) is measured for the different frequency bands on pin RSSI.

The RSSI value can be measured with 6 bits peak hold ADC. Because the RSSI pin is shared with ADC2 and the FAD circuit, the following conditions must be true for an RSSI measurement (see Figure 23):

- <B_RC> : RC2[7] = 0 to allow PD0 to discharge the RSSI capacitor.
- <P_FAD> (PC2 = 06), to set the switches 0,1,2 for RSSI (default after reset)
- Give any <B_> (except <B_DIV1>, <B_DIV2>, <B_DIV4>, <B_RST>) command to enable the analog part of the RSSI ADC.
- The RSSI ADC is started as soon as PD0 is set to 0 using DIP command <P_LDL> 01. Next the peak detector tracks the RSSI peaks from zero to maximum scale within 34 clock cycles of 1.152MHz

- The internal PD0 signal must be reset <P_LD> 01 to stop tracking and before <B_WRS> is given.
- The ADC data bits are written to the Data memory if the <B_WRS> command is executed.

The ADC consumes power as defined in Table 348 parameter lvddrf_rssi. Upon a <B_RST>, the ADC is in power down again.

Refer to Peak Hold ADC (continued) (table 367, page 297) for a full specification.

11.6 FAST ANTENNA DIVERSITY (FAD)

The SC14452 supports fast antenna switching during the (prolonged) preamble. To select between two signals coming from different antenna's first one antenna is selected, the RSSI value is sampled and then the other antenna is selected and the RSSI value is sampled. The RSSI value is sampled on two internal capacitors of maximum 2pF. Both sampled RSSI values are compared and the higher value determines the antenna to be selected. The analog implementation of the FAD algorithm consists of 3 switches and a compa-

rator. Since the RSSI and the FAD share the same hardware the RSSI and FAD algorithm may not operate during the same time.

To select the RSSI or the FAD 3 switches are implemented. The switches can be controlled by the new P_FAD instruction. Its parameter will directly control the switches 0, 1 and 2.

- Switch (0) is used to sample the RSSI value on the negative input of the ADC-comparator.
- Switch (1) is used to disconnect the ADC_DAC from the ADC-comparator input.
- Switch (2) is used to sample the RSSI value on the positive input of the ADC-comparator.

To use RSSI switch 1 and 2 must be closed and PD0 must be programmed low to initiate the RSSI measurement.

During FAD measurement switch 1 is opened and the

RSSI-pin connection to ground is disabled. When starting the measurement switches 0 and 2 will be closed and ANT_SEL will be '0' to select the first antenna. After some wait cycles switch 0 must be opened to freeze the RSSI value of the first antenna. ANT_SEL signal will become '1' and the second antenna is selected. Thereafter wait for another few cycles and open switch 2. This will activate the comparison and ANT_SEL will hold the result.

The PD pins will change at the same time as the internal switches (maximum 25 ns later). A DiP opcode <P_ANT> PAR specifies which PD pins are effective. The parameter holds two nibbles. The <PAR> will determine which PD pin will pass the ANT_SEL signal and the polarity. The value of ANT_SEL is saved in the shared RAM with the <B_WRS> instruction.

Only after a DiP reset the value of ANT_SEL is reset to '1'. Otherwise the previous value will be held.

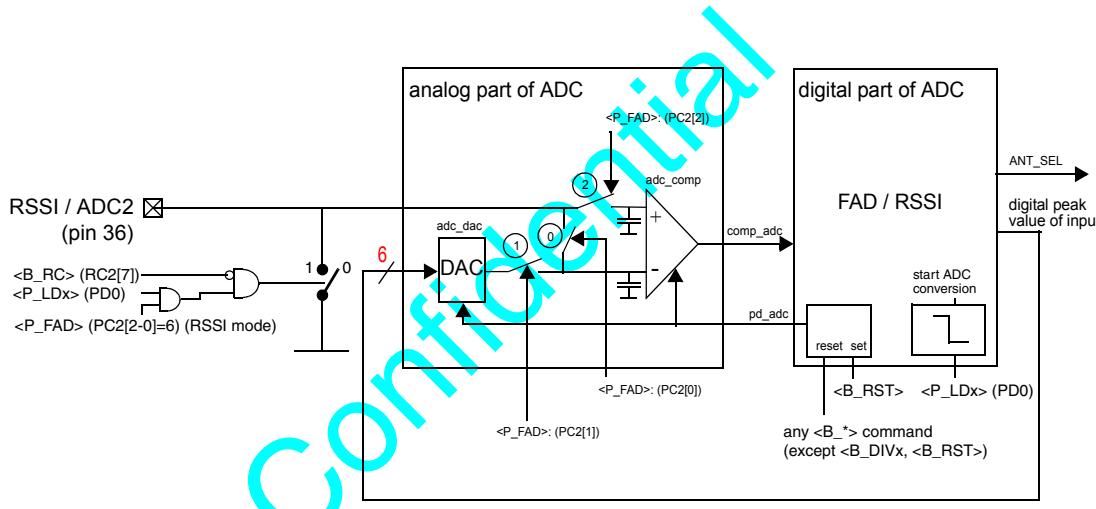


Figure 23 FAD/RSSI block diagram

11.7 MODULATED OUTPUT

The modulation method for RF transmissions is gaussian frequency shift keying with a bandwidth \times bit period product, BT=0.5 (amplitude controlled digital output). The SC14452 generates at the Transmit Data Output (pin TDO) the burst data at the baud rates 1.152 kbit/s, 576 kbit/s or 288 kbit/s. The TDO output can be programmed to output either a gaussian shaped symbol or a normal CMOS digital output (on TDOD only). The shape is spread over three symbols and coded with an eight bit DAC. The volume can be controlled with another 6 bits. The Gaussian output voltage is independent of the supply voltage level. With control bit "INV_TDO" the TDO output can be inverted. With M[1:0] set to: M[1:0]='00' or M[1:0] ='10' powers down. M[1:0]='01' activates the gaussian output shape (BT = 0.5) at a mid level. M[1:0]='11' is reserved at a mid level. With a <B_STx> command the actual data will be output. With VOL[5:0] the amplitude of the gaussian shape can be trimmed step wise.

The TDOD digital mode is available on a pin PPA pin

11.8 DEMODULATED DATA INPUT

The demodulator in the RF front end regenerates, the data bits. The Received Data In (RDI) is clocked into the SC14452 on the rising edge of the internal recovered clock signal (RCK). With control bit "INV_RDI" inversion of RDI signal can be selected. For debugging purposes, the RCK signal can be switched to an inputs.

The SC14452 is equipped with a very stable and accurate received symbol timing extraction unit using a nine times over sampled digital phase correlator on the S-field.

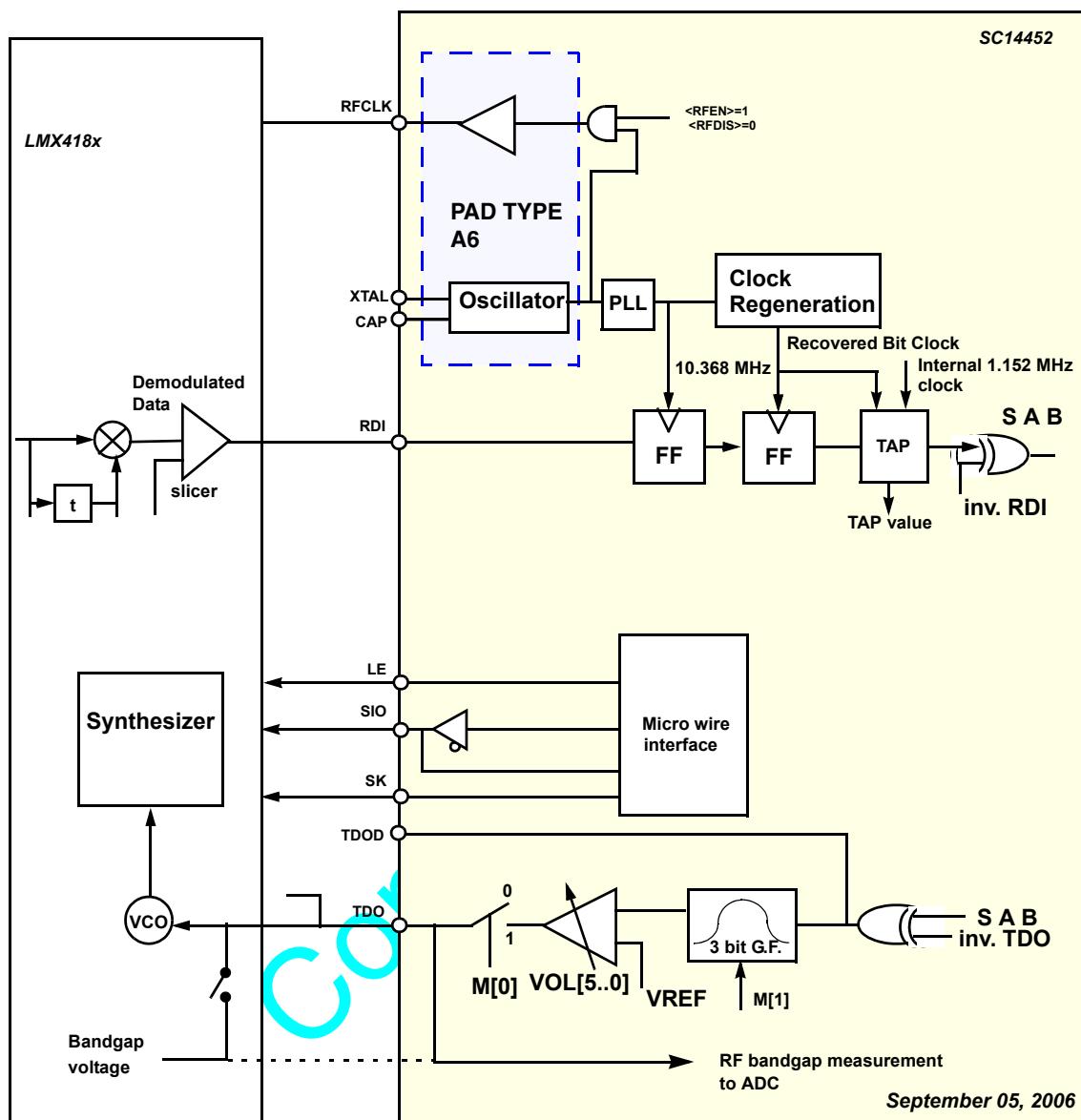


Figure 24 SC14452 Radio frontend interface

12.0 Gen2DSP

The SC14452 has two user programmable Gen2DSP (Generation 2, Generic DSP), for sample and block based algorithms. Both DSPs have fixed routines in microcode ROM and a micro code RAM which can execute user defined routines. The user defined algorithms can be compiled with C-compiler/assembler.

The input data as well as the algorithm parameters are stored in shared RAM1 or shared RAM2. Fixed parameters, like coefficients for iLBC or G.722 are placed in shared ROM.

Both Gen2DSPs have full zero wait state access to their own shared RAM for accessing algorithms parameters. The two Gen2DSPs can access the shared RAM of the neighbouring Gen2DSP giving extra wait states for both Gen2DSP. This bridged access eliminates the need to copy data between the shared RAMs.

Features

- 32k + 112k Micro code ROM with fixed audio, telecom and voice compression/decompression routines in ROM compatible with SC1443x. See Table 36 on page 60.
- 20k + 16k Micro code RAM for User defined algorithms developed with a C-compiler/assembler.
- 16k + 20k shared parameter RAM for zero wait state access of both DSPs, no wait states due to CR16C+/DMA memory access (PLL on), extra wait state with bridged Gen2DSP and DIP access.
- Parallel transcoding between sample processing (e.g. G.722/G.726) and block processing (e.g. iLBC) is possible.
- Interrupt capability e.g. for interrupting block based algorithms.
- Sample rates 8/16 and 32 kHz supporting standard, wide band Codec and Hi-Fi tone generation.
- Programmable clock frequency from up-to 82.944 MHz.

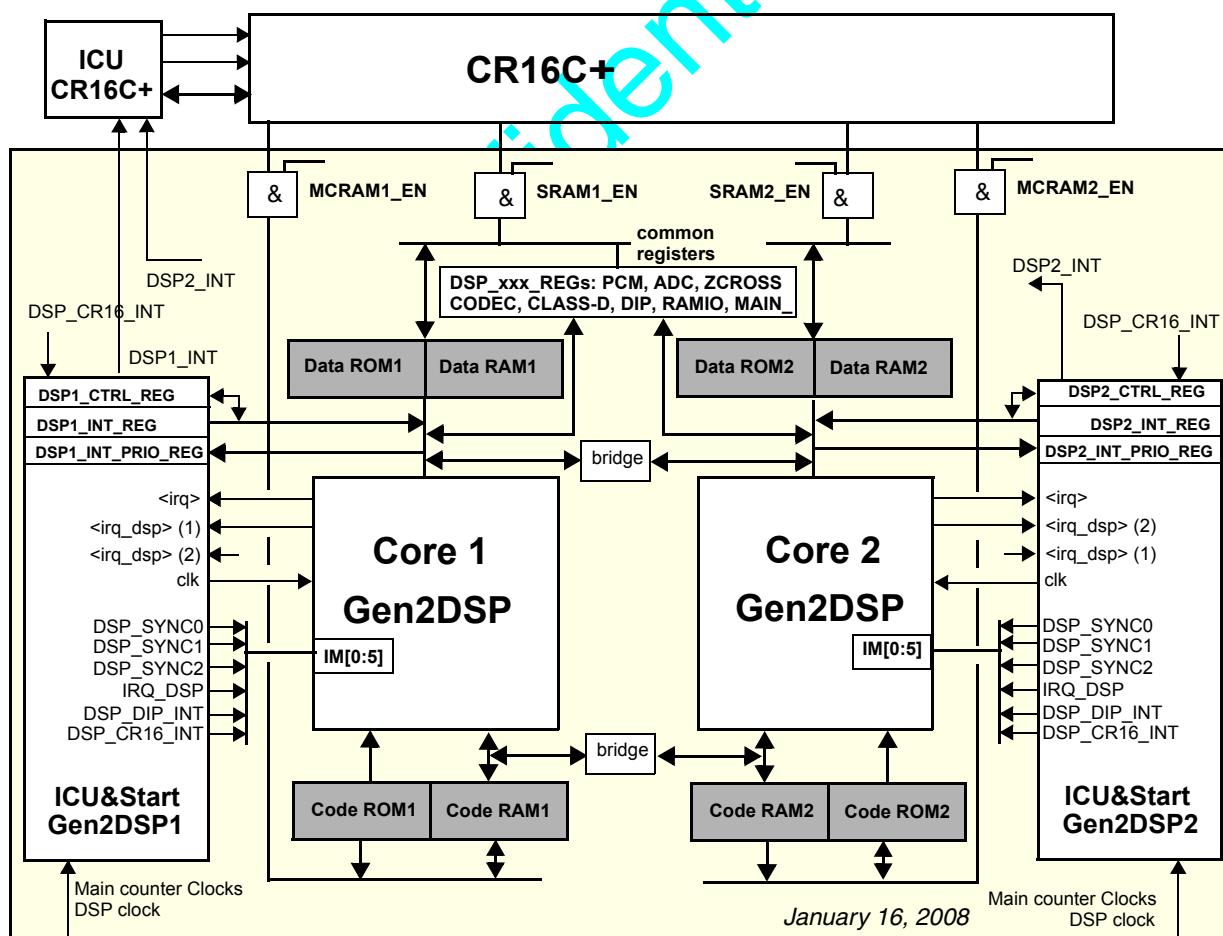


Figure 25 Gen2DSP1 and Gen2DSP2 block diagram

12.1 START-UP AND INTERRUPTS

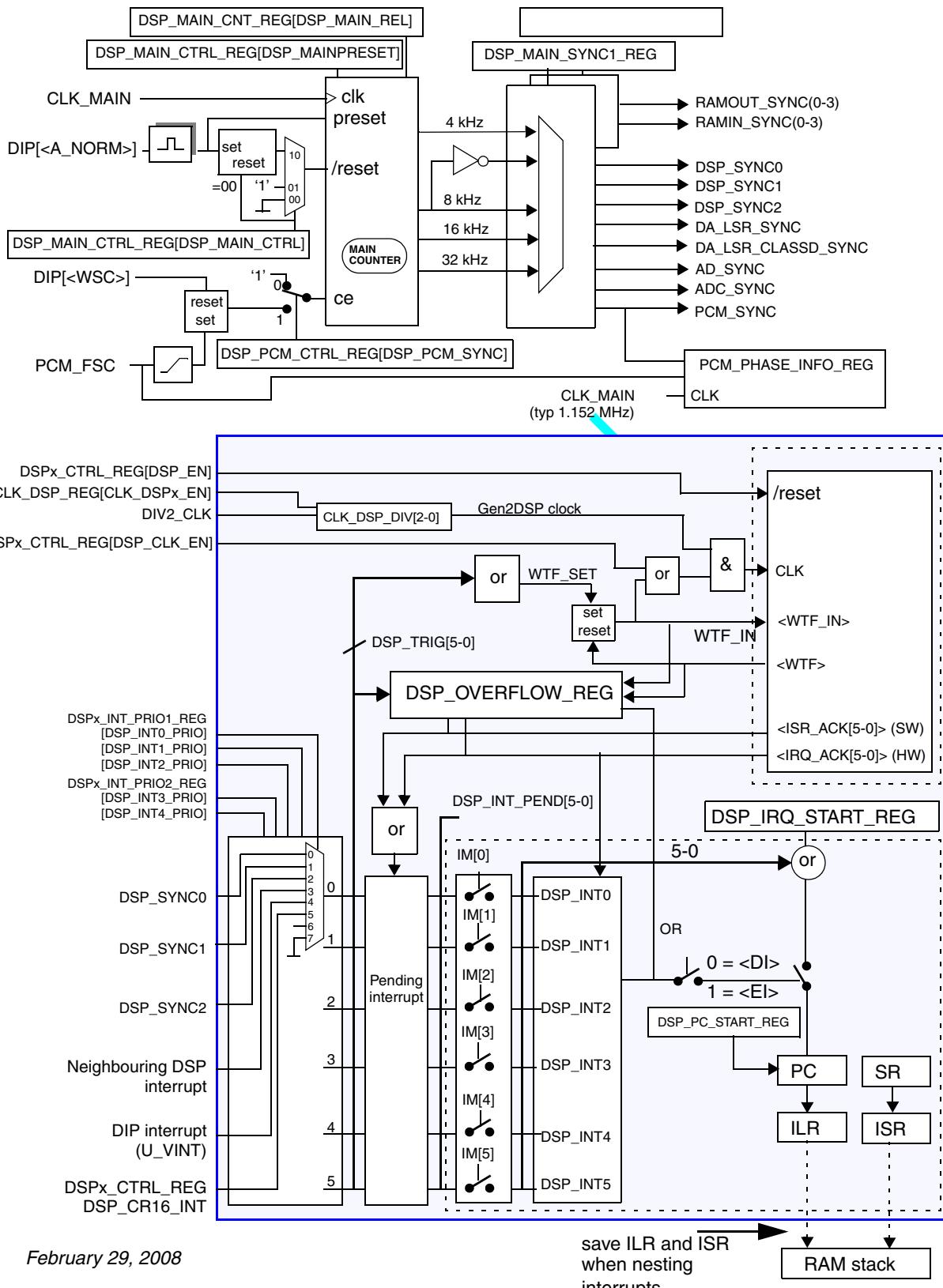


Figure 26 Gen2DSP 1 and Gen2DSP 2 main counter, start-up and Interrupts

Gen2DSP start-up

Both Gen2DSPx are started as follows:

- DSPx_PC_START_REG must be set
- CLK_DSP_REG[CLK_DSP1_DIV] must be set to required clock frequency
- DSPx_CTRL_REG[DSP_EN]=1. This bit sets Gen2DSP out of reset.
- Start triggers must be enabled in the DSPx_INT_PRIOX_REG and/or DSPx_INT_PRIOX_REG. The priority level is only important with interrupts (See next chapter)
- One of the start triggers must be given:
 1. DSP_MAIN_SYNC_REG[DSP_SYNC0] set for e.g. 8 kHz trigger/interrupts,
 2. DSP_MAIN_SYNC_REG[DSP_SYNC1] set for e.g. 16 kHz trigger/interrupts,
 3. DSP_MAIN_SYNC_REG[DSP_SYNC2] set for e.g. 32 kHz trigger/interrupts,
 4. Gen2DSP interrupt. Neighbouring Gen2DSP executes <IRQ_DSP>
 5. CR16C+ interrupt DSP_CR16_INT for e.g non-synchronized block processing (e.g JPEG)
 6. DIP interrupt. DIP executes <U_VINT> with operand=0001. 0000 to interrupt the Gen2DSP1 operand=0010. 0000 to interrupt the Gen2DSP2
- For triggers 1-3, the main counter must be enabled:
 - DSP_MAIN_CNT_REG[DSP_MAIN_REL] reload value to 0x8F (default for 8/16/32 kHz operation)
 - CLK_SPU2_REG[SW_MAINCDC_DIV] main counter clock to 1.152 MHz. (default for 8/16/32 kHz operation)
 - DSP_MAIN_CTRL_REG[DSP_MAIN_CTRL] set to
 - 01 = Main counter out of reset, free running. (Non DIP synchronised start-up for tone generation)
 - 10 = If Main counter was running single preset on <A_NORM>, else single start-up on <A_NORM> (DIP synchronised start-up for voice connection, SC14xxx compatible mode)
 - 11 = reserved
 - If DSP_MAIN_CTRL is started up with values 01 for e.g tone generation, the value may be changed to 10 to allow a resync by the DIP. For voice connections DSP_MAIN_CTRL_REG [DSP_MAIN_PRESET] must be set to 15 for SC14430 to get a compatible <A_NORM> ADPCM timing. (see AN-D-140)

If no interrupt are enabled (see next chapter), the Gen2DSP clock will be enabled by the WTF_SET condition (see will Figure 26) and starts executing.

For further details on synchronization refer to [AN-D-](#)

[140 "DIP Instruction Manual", chapter synchronisation.](#)

12.2 POWER SAVING

Gen2DSP clock disabling

The Gen2DSPs can be kept running e.g for block processing or can be put in power down after it has completed a certain processing task. To power down Gen2DSP, <WTF> instruction must be executed, which disables its clock.

If any 8/16/32 switching noise on VDD affects RF performance the Gen2DSP clock can be kept on all the time by setting DSPx_CTRL_REG[DSP_CLK_EN]=1. This has negative effect on power consumption however. In this case the <WTF> will only clear the "Start" synchronization bit and the <WTF_IN> instruction shall be used to poll the "Start" synchronization bit which will be set by one of the interrupt sources. In case the Gen2DSP starts up from power-down, the <WTF_IN> need not to be polled.

RAM/ROM interface clock disabling

The CR16C+ interface to shared RAM1 and RAM2, Gen2DSP1 and Gen2DSP2 registers and micro code RAM1 and RAM2 and Gen2DSP registers are disabled from start-up in order to save power. If a RAM is used, the interface must be enabled by setting appropriate CLK_AMBA_REG bits SRAM1_EN, SRAM2_EN, MCRAM1_EN or MCRAM2_EN to '1'. Access to disabled shared RAM and registers returns 0, while the microcode RAM and ROM return the NOP opcode (0xFFFF.FFFF).

RAM/ROM interface clock frequency reduction

For zero wait cycle shared RAM/ROM/Registers access and micro code RAM of CR16C+ and Gen2DSPs, CLK_PLL_REG[DIV2_CLK_SEL] must be set to 0 and the RAM/ROMs are clocked with the PLL clock (Max 165.888 MHz). Upon a simultaneous RAM/ROM read or write access of CR16C+/Gen2DSP1 or CR16C+/Gen2DSP2, the Gen2DSP is holded one cycle.

12.3 PCM SLAVE SYNCHRONISATION

In PCM slave mode all initialization is equal to the PCM master mode. DSP_PCM_CTRL_REG [DSP_PCM_SYNC] must be set to '1' to stop the main counter after DIP<WSC> instruction. At the rising edge of PCM_FSC, the main counter is resumed again. (See also 13.5 on page 67)

12.4 INTERRUPTS

Both Gen2DSPs can also use the six start-up triggers to interrupt other running processes as depicted in Figure 26. The interrupts can be used to interrupt:

- Block based algorithms
- Other interrupt routines

The Gen2DSP IM[5-0] (Interrupt Mask) register, masks the six interrupt sources. From start-up this mask is zero.

Interrupts are processed in order 0 to 5. However, the priority can also be changed in the DSPx_INT_PRIOx_REGS, which switches the interrupts triggers DSP_SYNC0 to DSP_CR16_INT to any of the interrupt inputs DSP_INT0 to DSP_INT5. If an interrupt occurs and the corresponding IM[x] mask bit is set, the Gen2DSPx jumps to address DSP_IRQ_START+ 2x of ISRn (Interrupt Source number) 0..5. The priority mechanism is only valid if more than one DSP_INTx is '1' at the same time. (see also "Nesting interrupts")

If an interrupt is serviced, the Status Register, SR (== CND, LF, IE, ISRnr[x]) bits are saved in ISR register, the global interrupt IE is disabled automatically, corresponding IM[x] is set '0' and the interrupt source is cleared. E.g. the DSPx_CTRL_REG [DSP_CR16_INT] is cleared. No other IM[x] are saved in the ISR.

At the end of a Interrupt Service Routine, the <RTI> (return from interrupt) restores register ISR in SR (CND, LF, IE, ISRnr). So IE and IM[x] are restored to '1', allowing interrupts again. The other IM[x] are unchanged. Next the PC is reloaded from the ILR register.

If a IM[x] is '0' or is set to '0', the corresponding interrupt stays pending. The pending interrupt is cleared if the ISR is serviced or if the <ISR_ACK[x]> instruction is given. Although the <ISR_ACK[x]> instruction is mainly used to detect an overflow (See also 12.6 on page 58), this instruction can also be given outside the Interrupt Service Routine to clear old pending interrupts.

Interrupt address "DSP_IRQ_START+ 2*ISRnr[0..5]" may contain either the interrupt routine itself (in case of a single interrupt) or it may contain a <jump> to the interrupt routine.

The interrupt routine is responsible for the whole context save, like registers

Note: Interrupt detection is done with the clock of the receiving Gen2DSP. Therefore the interrupt rate shall not exceed the frequency of the receiving Gen2DSP.

12.5 NESTING INTERRUPT

To allow nested interrupts, the ILR (interrupt link register) containing the saving PC, must be explicitly saved on stack and the <EI> (enable interrupt) must be executed. Recursive interrupts, interrupts from the same source, are not recommended, so the Interrupt mask IM[x] for the current interrupt shall be kept to '0' during an interrupt. To prevent interrupts with lower priority to interrupt a higher priority process who's IM mask bit is set to '0', IM mask bits of lower priority inputs shall also be set to '0' in the interrupt service routine.

When the Gen2DSPx is in power down after execution of <WTF>, any of the interrupt sources will power-up the Gen2DSP and the interrupt is handled. At the end of the interrupt routine, after the <RTI> is executed, the background program will resume. Note that the <WTF> shall be placed in the background process (main loop) and not in the interrupt routine in order to avoid inter-

rupt latency overhead and early clock disabling.

12.6 OVERFLOW

Both Gen2DSP have the following overflow status and mask bits in register DSP_OVERFLOW_REG. If both Status bit and corresponding mask register bits are set, a CR16C+ DSP_INT is generated.

- INT_OVERFLOW[x], M_INT_OVERFLOW[x].
This overflow indicates that a new DSP_TRIG[x] occurred before instruction <ISR_ACK[x]> was executed. DSP_TRIG[x] sets one of the internal signal bits TRIG_WIN[x]. (E.g. at 8kHz interrupt for sample processing). The <ISR_ACK[x]> resets internal signal TRIG_WIN[x] and must be executed after sample IO is done and before the next DSP_TRIG[x] occurs, else INT_OVERFLOW[x] is set. Note that only the sample IO must be done before the next DSP_TRIG[x]. The rest of the ISR may run longer but make sure that the next sample is copied before the next DSP_TRIG arrives. This is useful if an interrupt routine uses more cycles and temporarily goes over the DSP_TRIG[x] boundary. If interrupts are disabled in the Gen2DSP core with IM[5..0]=0, the INT_OVERFLOW bits are also set if the main SCP program does not execute the <ISR_ACK[x]>.
- WFT_OVERFLOW, M_WTF_OVERFLOW.
This bit is set if one of DSP_TRIG[x] start conditions occurs before <WTF> is executed. This is the 43x compatible overflow condition without interrupts.
- IRQ_OVERFLOW, M_IRQ_OVERFLOW
This overflow is set if a new DSP_TRIG[x] occurs while the previous interrupt request DSP_INT[x] was not acknowledged, so the ISR has not yet started. (IRQ_ACK still '1'). This overflow indicates that one of the enabled interrupts (With IM[x] =1) is not serviced.

Figure 27 shows the examples of the overflow cases.

Signal 1 to 4, Regions:

1. No INT_OVERFLOW.
2. INT_OVERFLOW. <ISR_ACK[x]> not yet given before new DSP_TRIG[x] occurs.

Signal 5 to 8, Regions:

1. No WFT_OVERFLOW
<WTF> set in time.
2. WFT_OVERFLOW. WFT too late, WFT_IN still '1' while new WFT_SET occurs.

Signal 9 to 11, Regions:

1. No IRQ_OVERFLOW.
2. IRQ_OVERFLOW. One of DIP_INT[x] not serviced.

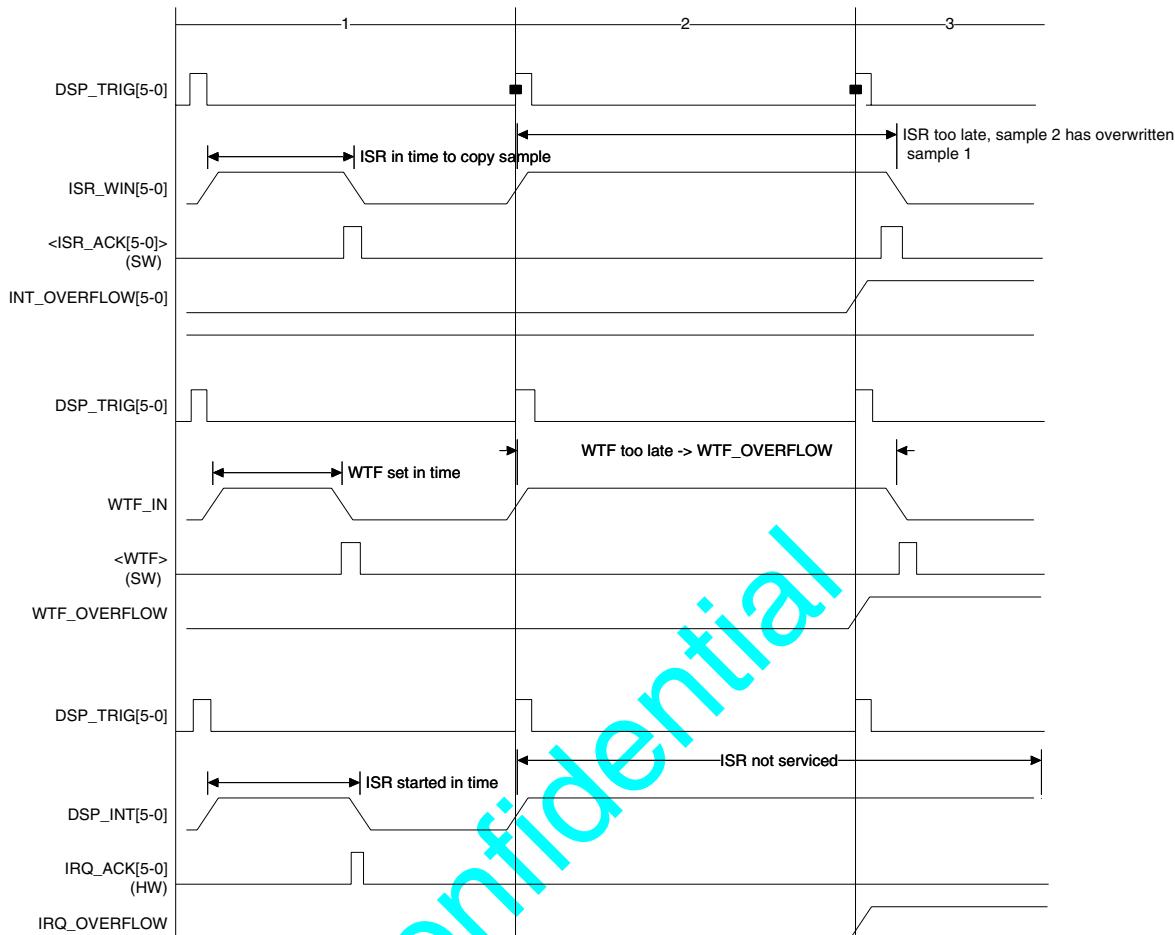


Figure 27 Gen2DSP overflow cases

12.7 REGISTERS

The registers are memory mapped Gen2DSP registers, therefore `DSP_CTRL_REG[SRAM1_EN]` and/or `[SRAM2_EN]` must be set to '1'. For common registers access only SRAM1_EN is set to '1'.

Both Gen2DSPx can access all the registers as shown in Table 70. It is the responsibility of the programmer that only one Gen2DSP updates the registers. In case both Gen2DSPs write to the same register at the same time, the second overwrites the values if the first. Both Gen2DSPs can read the same register at any time.

For debugging purposes the Gen2DSP program counter can be observed by the CR16C+ in the `DSPx_PC_REG`.

12.8 RAMIO REGISTERS

The RAMIO registers `DSP_RAM_OUTx_REG`, `DSP_RAM_INx_REG`, are special registers used for minimum delay I/O to the DIP. Refer to paragraph 10.5 for more information on minimum delay. The selection for shared RAM 1 or shared RAM 2 is determined by

DIP `<A_LDWx>`, `<A_LDRx>` instructions.

12.9 SHARED ROM/ROM

Gen2DSP1 can access the full range of Gen2DSP2 shared RAM region and vice versa as shown in Detailed shared Data RAM and ROM Memory map for DIP, CR16C+ and Gen2DSP1 and Gen2DSP2 (table 68, page 167). Both Gen2DSPs access their own memory at full speed, their neighbouring memory also at full speed with 32 bits instructions but with one wait cycle in case of 16 bits instructions. The DIP has always zero wait cycle access, in order to guarantee slot timing. The CR16C+ always has one wait cycle when accessing shared RAM1 and 2 and ROM 1 or 2.

Refer also to chapter "Power saving" on page 57.

12.10 MICRO CODE ROM/RAM

Gen2DSP1 can access the full range of Gen2DSP2 Micro RAM region (not ROM) and vice versa as shown in Detailed micro code RAM and ROM Memory map for DIP, CR16C+ and Gen2DSP1 and Gen2DSP2 (table 69, page 167). Both Gen2DSPs access their "own"

memory at full speed, their neighbouring with one wait cycle. For maintenance reasons, the ROM contents is only accessible via the jump index table. **The start address is programmable via registers DSPx_JTBL_REG, the reset values points to the micro code ROM.** The Micro Code RAMs are accessible by the CR16C+ can be updated while the Gen2DSP is executing without introducing wait states. Not used micro code RAM can therefore be used as scratch RAM for the CR16C+. One extra wait cycle will be introduced for the CR16C+, the Gen2DSPx has always full access. Both Gen2DSP can not access the micro code RAM or ROM as data RAM.

Refer also to chapter "Power saving" on page 57.

12.11 SUBROUTINE CONTROL PROGRAM (SCP)

The SCP has been redefined to be the main program loop from which the algorithms in micro code RAM or ROM are called. The SCP also runs in micro code RAM. The SCP can be also be placed in a interrupt routine. The algorithms are called via a jump table entry. The start of the jump table is the start of the ROM. Refer to Table 36 for address of the micro code ROM addresses.

For detailed information on Gen2DSP instructions refer to [AN-D-129 "SC14450/SC14480 Gen2DSP programming manual"](#).

Table 36: Gen2DSP algorithms in Micro Code ROM1 and Micro Code ROM2

Jump table index	Symbol	Algorithm Name	Gen2 DSP1	Gen2 DSP2
0	ABS	Absolute value	✓	✓
1	AEC	Acoustic Echo Canceller (428 compatible) See SC14450_bug47.	✓	✓
2	AGC	AGC	✓	✓
3	ANAGC	Analog AGC	✓	✓
4	AND	AND	✓	✓
5	BLKMINMAX	Block Min/max value	✓	✓
6	BLKNORM	Block normalisation	✓	✓
7	BUF_PACK	Bit, Nibble and Byte swap and data storage in a circular buffer	✓	✓
8	BUF_UNPACK	Bit, Nibble and Byte swap and retrieving data from a circular buffer	✓	✓
9	CASDET	CAS Detector	✓	✓
10	CASDET2	CAS Detector with separate inputs for low and high tone	✓	✓
11	CBUFFER	Versatile Buffer (delay line)	✓	✓
12	CFFT	FFT and IFFT	✓	✓
13	CONF	Conference	✓	✓
14	COPYB2B	Copy Block to Block	✓	✓
15	COPYP2B	Copy Pointer to Block	✓	✓
16	COPYP2P	Copy Pointer to Pointer	✓	✓
17	COUNTER	Programmable counter	✓	✓
18	DBLLAW2LIN	A/u law to linear for INTERP	✓	✓
19	DCT	Discrete Cosine Transform		✓
20	DEC	Decrement	✓	✓
21	DIVF	Fractional Division	✓	✓
22	DIVI	Integer Division	✓	✓
23	DSCRTGAIN	Discrete gain function	✓	✓
24	DTMFDET	DTMF detector (improved)	✓	✓
25	DTMFRU	DTMF for Russia	✓	✓
26	EDGEDET	Edge Detector	✓	✓
27	EXP	Gives number of significant bits	✓	✓

Table 36: Gen2DSP algorithms in Micro Code ROM1 and Micro Code ROM2

Jump table index	Symbol	Algorithm Name	Gen2 DSP1	Gen2 DSP2
28	FMOPERATOR	FM operator block	✓	✓
29	FSKDECOD	FSK decoder with data recovery	✓	✓
30	FSKGEN	FSK Generation	✓	✓
31	G722_RX	G722 ADPCM Decoder	✓	✓
32	G722_TX	G722 ADPCM Encoder	✓	✓
33	G726	G726 ADPCM Encoder and decoder	✓	✓
34	G726_VQI	G726 ADPCM decoder with VQI	✓	✓
35	GETREVISION	Gen2DSP HW and SW revision number	✓	✓
36	GRAY_SCALE	JPEG Gray scale pixel saturation between MIN and MAX	✓	✓
37	HAGC	Handsfree AGC	✓	✓
38	HFREE	Handsfree/ half duplex switch	✓	✓
39	IDCT	JPEG Inverse Discrete Cosine Transform		✓
40	INC	Increment	✓	✓
41	INTERP	Interpolator for pitch shift of wave tables	✓	✓
42	LAW2LIN	A/u law to linear	✓	✓
43	LEC	Line Echo Canceller	✓	✓
44	LIN2LAW	Linear to A/u law	✓	✓
45	LIN2LOG	Linear to logarithmic($2^{\log N}$)	✓	✓
46	LOG2LIN	Logarithmic to Linear (2^N)	✓	✓
47	MATRIX3X3	3x3 Matrix multiplication	✓	✓
48	MIXER	Sound mixer	✓	✓
49	MONITOR	Monitor	✓	✓
50	NC100HZ	100 Hz noise canceller	✓	✓
51	NSHL	Nibble Switch (shift left)	✓	✓
52	NSHR	Nibble Switch (shift right)	✓	✓
53	ORB	OR	✓	✓
54	PFILT	Prog. filter (biquad)	✓	✓
55	PFIRFLT	Prog. FIR filter	✓	✓
56	PLEVDET	Power Level Detector	✓	✓
57	PNLEVDET	Noise detection	✓	✓
58	POLY	Polynomial	✓	✓
59	RGB2YCC_420	RGB to YCC color conversion and downsampling	✓	✓
60	RNDGEN	Random generator	✓	✓
61	RS_PARSYN	R-S Parity and Syndrome calculation	✓	✓
62	SHIFT	Shift left/right	✓	✓
63	SINGLETONEDET	Single tone detector	✓	✓
64	SUB	Subtract	✓	✓
65	SUMM	Summator	✓	✓
66	SUPP	Suppressor	✓	✓
67	TONEDET	Tone detector	✓	✓
68	TONEGEN	Triple Tone Generation	✓	✓
69	UPSAMPLE420	Graphic Data upsampling 4-2-0	✓	✓

Table 36: Gen2DSP algorithms in Micro Code ROM1 and Micro Code ROM2

Jump table index	Symbol	Algorithm Name	Gen2 DSP1	Gen2 DSP2
70	UPSAMPLE422	Graphic Data upsampling 4-2-2	✓	✓
71	UPSAMPLE422V	Graphic Data upsampling 4-2-2V	✓	✓
72	WINDOWADD	FFT Window function	✓	✓
73	XORB	Exclusive or	✓	✓
74	YCC2RGB	YCC to RGB color conversion	✓	✓
75	ZCROSS	Zero cross detection	✓	✓
76	G729AB_ENC	G.729 AB Encoder		✓
77	G729AB_DEC	G.729 AB Decoder		✓
78	-	Reserved		
79	-	Reserved		
80	-	Reserved		
81	MIDI_SYNTH	Midi synthesizer is a configurable mix of FM melodic, FM percussion and wavetable voices - include wavetable 'chorus' option - includes Mixer, LFO's, timer - support 32 voices when DSP runs at 80MHz - generate configurable interrupt vector when the timer expires	✓	
82	PAEC	Perceptual Acoustic Echo canceller.	✓	✓
83	IRQ_CTX_SAVE	Interrupt Service Routine Context save	✓	✓
84	IRQ_CTX_REST	Interrupt Service Routine Context restore	✓	✓
85	USER0	Jump to user defined function in MC RAM address 0	✓	✓
86	USER1	Jump to user defined function in MC RAM address 2	✓	✓
87	USER2	Jump to user defined function in MC RAM address 4	✓	✓
88	USER3	Jump to user defined function in MC RAM address 6	✓	✓
89	USER4	Jump to user defined function in MC RAM address 8	✓	✓
90	USER5	Jump to user defined function in MC RAM address 10	✓	✓
91	USER6	Jump to user defined function in MC RAM address 12	✓	✓
92	USER7	Jump to user defined function in MC RAM address 14	✓	✓
93	RFFT	Real input FFT	✓	✓
94	RiFFT	Real output FFT	✓	✓
95	RFFTD	Dynamic Scaling real-to-complex FFT	✓	✓
96	RiFFTD	Dynamic Scaling complex-to-real iFFT	✓	✓
97	RESAMPLE	Audio buffer resampler (for USB/PCM resampling)	✓	✓
98	POLYPHASE	Sample rate resampler (for SVC)	✓	✓
99	CBUFIRQ	Rec/play buffer with programmable frame-interrupt generation	✓	✓
100	WIDEBAND	Artifical voice enhancement	✓	✓
101	AUDIOHD	Artifical voice enhancement	✓	✓
102	VAD_CNG_ENC	G711's and G726's Voice Activity Detection encoding function		✓
103	VAD_CNG_DEC	G711's and G726's Comfort Noise Generation decoding function		✓

Table 36: Gen2DSP algorithms in Micro Code ROM1 and Micro Code ROM2

Jump table index	Symbol	Algorithm Name	Gen2 DSP1	Gen2 DSP2
104	ILBC_GIPS_ENC	ILBC Encoder		✓
105	ILBC_GIPS_DEC	ILBC Decoder		✓
106	ILBC_GIPS_DECPLC	ILBC Decoder with PLC		✓
107	SVC_ENC	SVC Encoder		✓
108	SVC_DEC	SVC Decoder		✓
109	DOFE	G711's and G726's PLC main function		✓
110	ADDTOHISTORY	G711's and G726's PLC buffering function		✓
111	PACKETIZE_ENC_G729	G729 Buffer Pack		✓
112	DEPACKETIZE_ENC_G729	G729 Buf Buffer Unpack		✓
113	G722_PLIC	G722 decoder with Packet Loss Concealment	✓	✓

Note 9: End products using G.729 require license from Licensing Service Provider Sipro Lab Telecom (www.sipro.com)

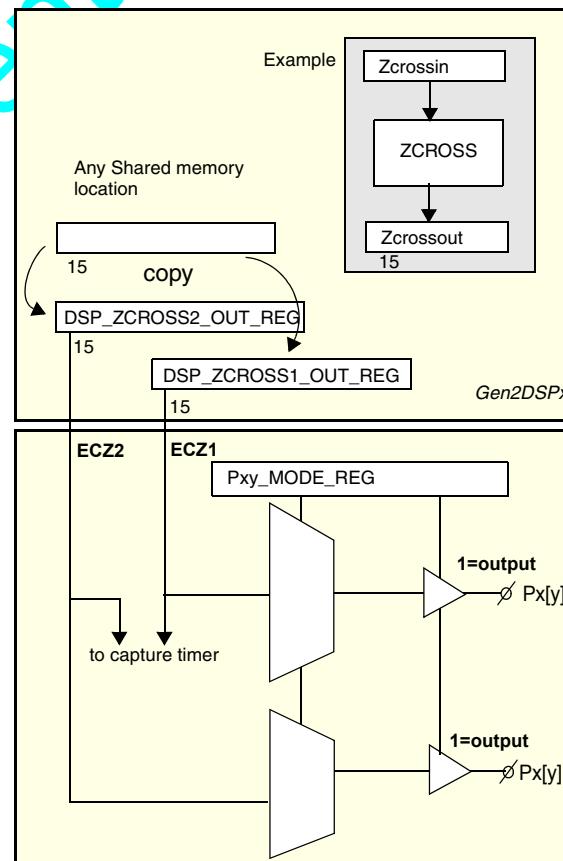
Note 10: Supply of this product does not convey a MP3 license under the relevant intellectual property of Thomson, Fraunhofer Gesellschaft and/or other third-parties nor imply any right to use this product in any finished end user or ready-to-use final product. An independent license for such use is required. For details, please visit <http://www.mp3licensing.com> and <http://en.wikipedia.org/wiki/MP3>.

12.12 ECZ, ECZ2 USAGE AND PORT MAPPING

For diagnostics or control purposes, the Gen2DSPs can directly set port pins using ECZ1 and ECZ2 signals as shown in Figure 28.

The Gen2DSP may copy any shared RAM location to internal registers `DSP_ZCROSS1_OUT_REG` and/or `DSP_ZCROSS2_OUT_REG`. Only bit 15 of the RAM is used. These register outputs are the ECZ1 resp ECZ2 signals which can be directly mapped on `Px[y]` resp `Px[y]` by setting `Px_yy_MODE_REG` multiplexer as shown above.

The ECZ1 and ECZ2 usage and names originate from the Gen2DSP ZCROSS function which requires a single bit hardware register to store output parameter "zcrossout" sign bit 15. These register outputs ECZ and ECZ2 are used by the "Capture timers/counters" on page 122 for frequency measurement of line interface signals.


Figure 28 EZCx port mapping

13.0 PCM interface

The PCM interfaces as shown in Figure 29 is a 8/16/32kHz synchronous interface to external audio devices, ISDN interface circuits and serial data interfaces.

Features

- PCM_CLK in Master mode Nx512, Nx576kHz (N=1,2,4,8) and Nx1.536 MHz
- Programmable strobe output 1,8,16,32 bits and Programmable strobe output before or on the first bit. (Master mode)
- PCM_CLK in Slave mode upto 4.608 MHz 1 bit strobe input on or before first bit.
- 4x16bits channels.
- Programmable inversion of PCM_CLK, PCM_FSC
- I2S mode (Left/Right channel selection)

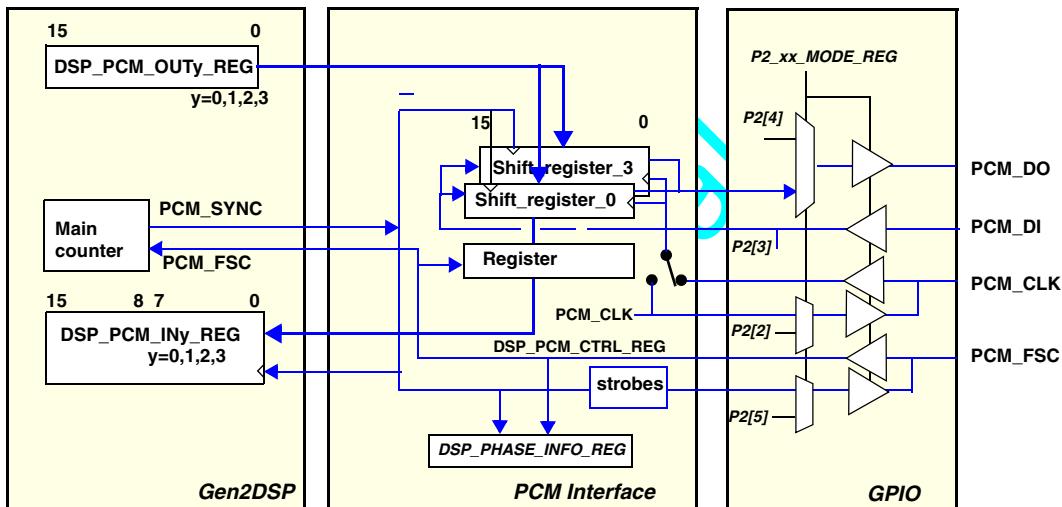


Figure 29 PCM interface

13.1 INTERFACE SIGNALS.

- PCM_FSC, strobe signal input, output Frequency is equal to Main counter/Gen2DSP first interrupt which can be 8/16/32 kHz
- PCM_CLK, PCM clock input, output, one clock/bit Frequency: Main counter clock 1.152, 2.304 or 4.608 MHz.
- PCM_DOUT, PCM Data output, push pull or open drain with external pull-up resistor.
- PCM_DIN, PCM Data input

Pin assignment can be done according Programmable Peripheral Matrix (table 3, page 16).

DSP_PCM_CTRL_REG bit PPOD = 1 sets PCM_DO to **open drain** configuration. This bit can be set if the bus is shared with other devices.

If PPOD is 0, PCM_DO is 0 after transmission of the last channel. With PPOD is 1, PCM_DO is Hi-Z if a '1' is transmitted and after the last channel.

The PCM interface is in power down if DSP_MAIN_SYNC1_REG[PCM_SYNC] is "11" and

CLK_SPU1_REG[SW_PCMCDC_EN] = '0'.

13.2 CHANNEL ACCESS

The PCM interface has 4 channels of 16 bits which are byte and word wise accessible by the CR16C+ and only word wise by the Gen2DSP. The channels are accessible through 16 bits registers DSP_PCM_OUT0_REG, DSP_PCM_OUT1_REG, DSP_PCM_OUT2_REG, DSP_PCM_OUT3_REG and DSP_PCM_IN0_REG, DSP_PCM_IN1_REG, DSP_PCM_IN2_REG, DSP_PCM_IN3_REG .

The first channel is mapped on bit 15-0, etc. Channels are shifted in and out at the same time with the MSB first.

Duty cycle jitter

The PCM clock divider generates three frequencies being the base frequency DIVx_CLK/CODEC_DIV and the base frequency x2 and x4.

Figure 30 shows an example of the PCM_CLK derived from DIVx_CLK = 10.368 MHz and CODEC_DIV=9.

- Base frequency $10.368/9 = 1.152$ MHz

- Base frequency $(10.368/9)*2= 2.304$ MHz
- Base frequency $(10.368/9)*4= 4.608$ MHz

If CODEC_DIV can not be divided by 2 or 4, then the derived clocks (x2 and x4) have a duty cycle jitter

determined by the DIVx_CLK of as shown below. If CODEC_DIV can be divided by 2, reps 4. x2 and x4 clocks have no jitter.

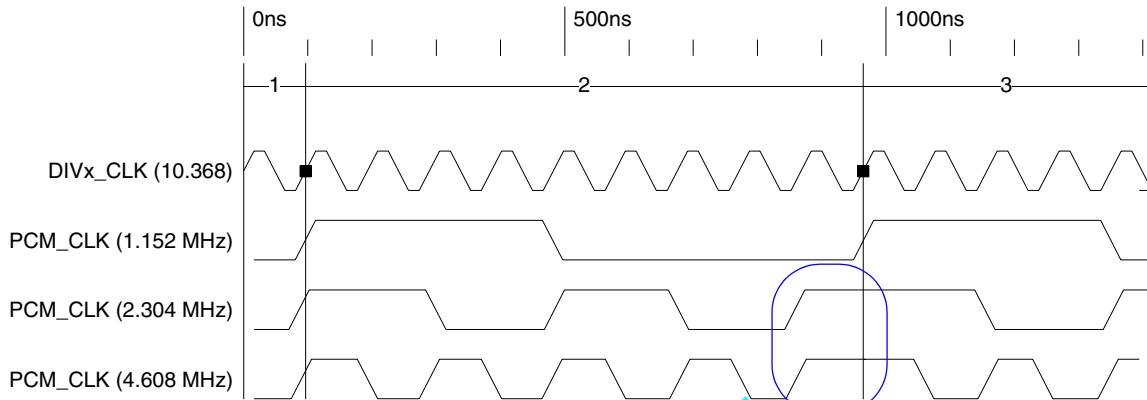


Figure 30 Duty cycle variation with CODEC_DIV

13.3 CLOCK STRETCHING/SHORTENING

In **portable** applications If PCM master, upon the execution of DIP <WNTP1>, <WNTM1> or <EN_SL_ADJ> the PCM_CLK and PCM_FSC are automatically stretched/shortened with 1 DECT bit of 1.152 MHz in order to maintain a constant phase to the received DIP frame. This stretching/shortening happens every 1.152 MHz with the selected clock DIVx_CLK (DIVN_CLK, DIV2_CLK or DIV1_CLK) until the 1.152 MHz bit is corrected.

13.4 DATA FORMATS

The various PCM data formats are shown on the next pages. Figure 31 show timing details of the PCM interface timing

Master mode

Master mode is selected if DSP_PCM_CTRL_REG bit [PCM_MASTER] = 1.

In master mode **PCM_FSC** is output and falls always over Channel 0. The duration of PCM_FSC is programmable with **PCM_FSCOLEN**: 1, 8, 16, 32 clock pulses high. The start position is programmable with **PCM_FSC0DEL** and can be placed before or on the first bit of channel 0. The repetition frequency of **PCM_FSC** is programmable in **DSP_MAIN_SYNC1_REG** bits **PCM_SYNC** between 8,16, 32 kHz.

If master mode selected, **PCM_CLK** is output. The polarity of the signal can be inverted with bit **PCM_CLKINV**.

The **PCM_CLK** frequency selection is described in paragraph 13.3.

Slave mode

In slave mode (bit **MASTER** = 0) **PCM_FSC** is input and determines the starting point of channel 0. The repetition rate of **PCM_FSC** must be equal to **PCM_SYNC** (8, 16 or 32 kHz) and must be high for at least one **PCM_CLK** cycle. Within one frame, **PCM_FSC** must be low for at least **PCM_CLK** cycle. Bit **PCM_FSCDEL** sets the start position of **PCM_FSC** **before or on** the first bit (MSB).

In slave mode **PCM_CLK** is input. The minimum received frequency is 256 kHz, the maximum is 4.608 MHz.

In slave mode **DSP_PCM_CTRL_REG** bit [DSP_PCM_SYNC] must be set to 1 to resume the main counter on a rising edge of **PCM_FSC**.

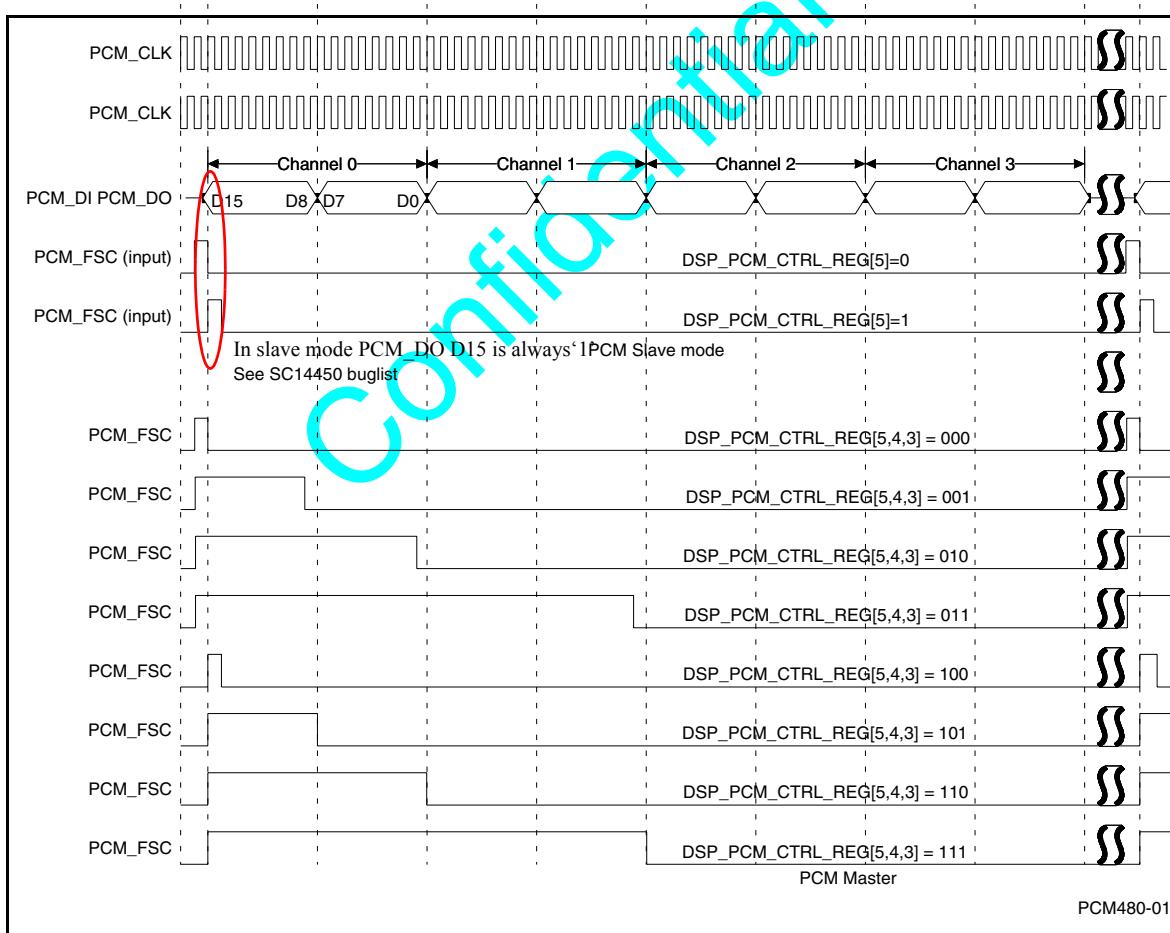


Figure 31 PCM interface formats

13.5 EXTERNAL SYNCHRONISATION

With the PCM interface in slave mode, the DIP can be started at an external PCM_FSC and the phase difference between the internal and external PCM_FSC can be monitored in DSP_PHASE_INFO_REG in units of main counter clock (typically 1.152 MHz).

To maintain a constant phase the following parameters may be changed by extending/shortening the DIP frame using <WT> instructions in combination with DIP instructions <WNTP1> and <WNTM1>, which stretches/shortens the following clocks:

- ad_cdc_clk,
- da_lsr_cdc_clk,
- da_classd_cdc_clk,
- pcm_cdc_clk,
- main_cdc_clk.

Initial synchronisation

Initial Phase synchronisation between PCM_FSC and the DIP frame can be done by executing DIP command <WSC> which halts the DIP processor and main counter. The next PCM_FSC rising edge resumes DIP operation and main counter [after 16x4 channel = 64x PCM_CLK cycles](#).

In order to synchronise the Gen2DSP to the PCM interface and DIP, the next steps must be followed:

- Set CLK_MAIN to 1.152 MHz (for 8/16 kHz sync)
- DSP_MAIN_CTRL_REG[9-8] = 2 // (wait for A_NORM)
- DSP_PCM_CTRL_REG bit [PCM_MASTER] = 0. // slave mode
- DSP_PCM_CTRL_REG bit [DSP_PCM_SYNC] = 1
- DSP_MAIN_SYN1_REG[PCM_SYNC] = 00, 01 or 10 equal to external PCM_SYNC frequency needed to measure internal and external phase difference in the DSP_PHASE_INFO_REG.
- DIP_CTRL2_REG[SLOTCNT_RES] = 1
- DIP <A_NORM> must be executed to trigger the main counter.
- DIP <WSC> must be executed to stop main counter and DIP processor.
- The Gen2DSP main counter (Gen2DSP start, RAMIO etc.) is resumed on the first rising edge of PCM_FSC.

Refer to [AN-D-140 "DIP Instruction Manual"](#) for detailed application information on synchronisation.

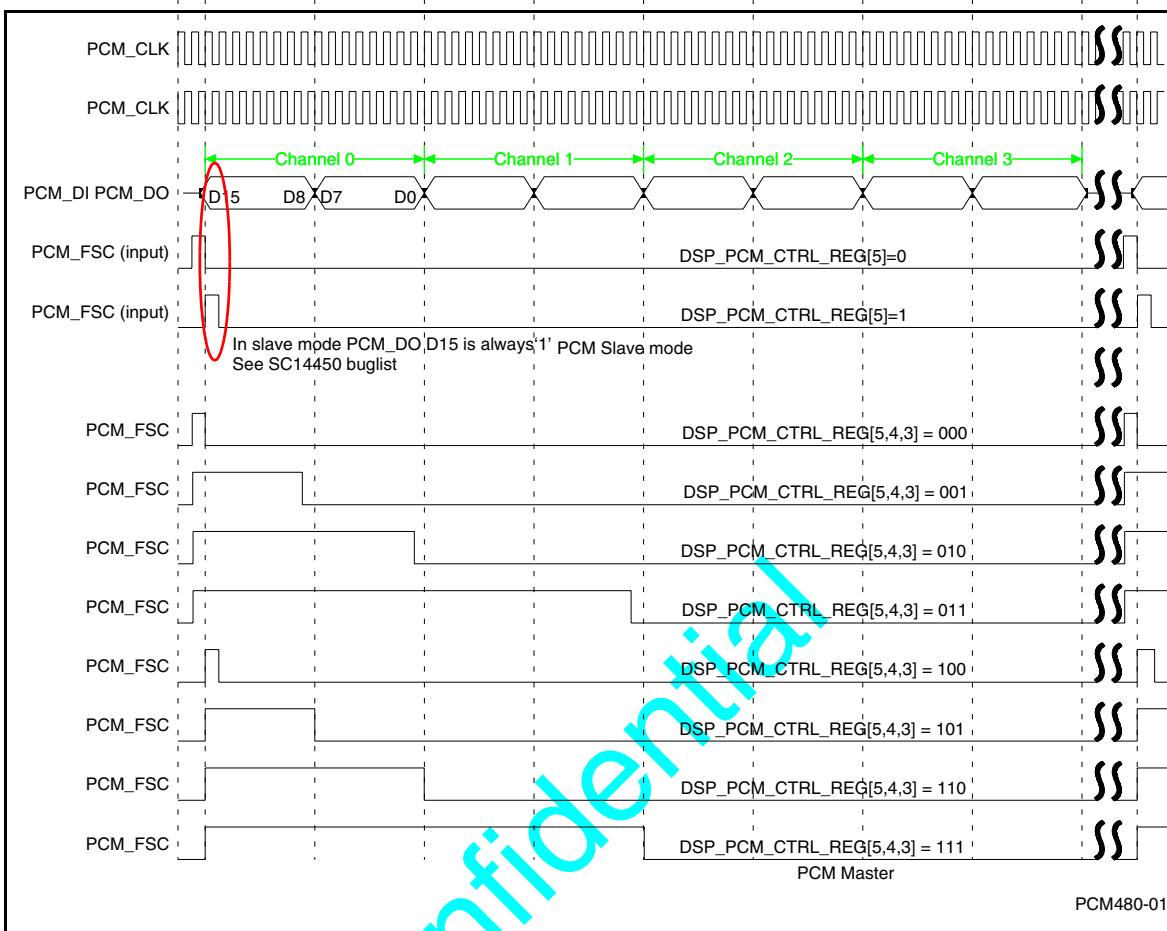


Figure 32 PCM interface formats

14.0 CODEC and Analog frontend

The SC14452 contains a 1.8 Volt CODEC and a full differential Analog Front End which can be configured for handset or basestation applications including on-caller-id, ring detection and voice activity detection for baby phone applications.

Features

- 8, 16 and 32 kHz operations
- Separate AD and DA clocks can be selected for improved out-of-band noise performance.
- Optional Low Pass Filter in DA path
- Very low microphone amplifier offset eliminates any digital offset compensation.
- Internal buffer circuit eliminates VBUF pin
- Integrated headset detection
- Single ended or differential microphone and loudspeaker connections
- Loudspeaker gain adjust in 8 steps of 2 dB
- Microphone gain 15 steps of 2 dB from 0 to +30 dB

14.1 LOUDSPEAKER CONNECTION

The loudspeaker stage can be fully configured through register **CODEC_LSR_REG**. The LSRN_MODE, and LSRP_MODE bits are used to select differential or single ended (see Figure 37 and Figure 38)

Dynamic loudspeakers with an impedance as low as $28\ \Omega$ can be connected as well as ceramic loudspeakers with an equivalent resistance of $600\ \Omega$ and $30\ \mu F$ capacitance can be connected without additional circuitry. The loudspeaker gain can be controlled with **CODEC_LSR_REG[LSRATT]** from plus 5 to minus -9 dB in 8 steps of ~ 2 dB/step.

In basestation applications (see Figure 37) the LSR+ or LSR- can be configured as single output to the line interface. The unused LSR+ or LSR- pin can be configured as a voltage reference pin (AGND) with 0.9V output voltage.

14.2 MICROPHONE CONNECTION

The microphone stage can be fully configured through register **CODEC_MIC_REG**. The MIC_MODE bits are used to select differential or single ended modes.

Several types of microphones can be used. The supply current for the microphone is fed through differential reference voltages independent of the supply voltage. The microphone gain can be adjusted in the **CODEC_MIC_REG[MIC_GAIN]** from 0 dB upto plus 30 dB in steps of 2dB.

For active microphones a voltage source with high supply voltage rejection ratio is provided on supply pins VREFp/VREFm. Filtering of internal and external reference voltages is provided with internal capacitor on the VBUF only. The time constants can be set with internal resistors. With VBUF_INIT set to '1' (reset value), 100k

ohm resistors are selected. If VBUF_INIT is '0' (recommended), 1M ohm series resistance is selected.

14.3 HEADSET CONNECTION

The AFE can be configured to handle two single ended microphones, one normal and one headset microphone. (see Figure 38) Both microphones are supplied from **VREFp**. The optional headset microphone, if connected to **MICn**, can be muted by setting **MICN_AGND_OUT** to '1'. The normal microphone signals is connected to **MICp**.

The headset loudspeaker is connected to the **LSRn** pin. By setting **LSRp** in power down, the normal loudspeaker can be muted.

In order to prevent a buzzer tone on the headset speaker the buzzer tone must only be put on **LSRp** (or **LSRn**) only (single-ended) when the headset speaker is plugged-in. The headset loudspeaker, if connected, can be muted by switching **LSRn** to AGND level.

This implies a -6 dB level reduction compared with the differential mode buzzer level (the case when the headset speaker is not plugged-in)

14.4 CODEC OFFSET COMPENSATION

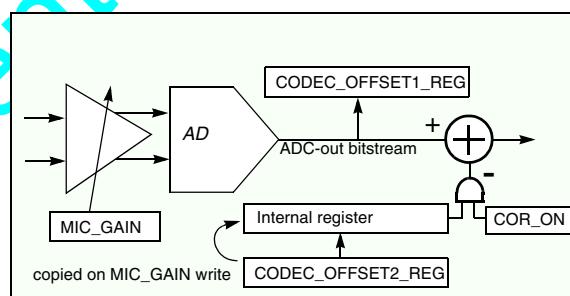


Figure 33 Codec offset compensation

The **MIC_GAIN** amplifier offset error can be compensated in the digital domain. Therefore the offset error for each gain setting must be measured at start-up by reading out **CODEC_OFFSET1_REG** while the **MIC_MUTE** is set to '1'.

For an applied **MIC_GAIN** setting, the corresponding offset error value must then be written to **CODEC_OFFSET2_REG** to compensate the offset error. The value in this register becomes active in a seamless way after the new **MIC_GAIN** value is written to the **CODEC_MIC_REG**.

This synchronisation process guarantees simultaneous activation of the new **MIC_GAIN** and corresponding **CODEC_OFFSET2_REG** value. During synchronisation, which takes about 156 us, **CODEC_TEST_CTRL_REG[COR_STAT]** becomes '1'. As soon as **COR_STAT** is '0', **CODEC_OFFSET2_REG** may be loaded with new value. The **COR_ON** bit must be switched on before the initial load of **CODEC_OFFSET2_REG** and **MIC_GAIN**.

14.5 FILTERING

In the DAC path of the codec has been extended with some extra features which can be used to reduce noise. An extra low-pass filter with a programmable cut-off frequency via CODEC_ADDA_REG[LPF_BW] in the range from 5.5 kHz to 30 kHz can be selected prior to final amplification. Depending on the pole frequency of this filter extra droop will occur in the pass-band. It is up to the application what is acceptable and desirable. The high-frequency poles for instance are useful to reduce out-band-noise (basestation application) without affecting the pass-band transfer function too much.

To reduce near-band noise, say from 4 kHz to 10 kHz when for instance loudspeakers are used with a larger bandwidth, extra filtering is normally necessary to meet the integrated idle channel noise specification. In this case the extra filter can be operated at a lower pole frequency.

As extra feature the DAC reconstruction filter double-pole (standard 2 x 8 kHz) can be halved by setting CODEC_ADDA_REG[DA_HBW] to '1'. This will give some droop in the pass-band (0 to 4 kHz). The HBW-mode can be used with or without the extra LPF. The passband-droop can be corrected by the DSP when necessary.

Also new in the codec is the fact that the sample clocks of the ADC (transmit path) and DAC (receive path) can be set to separate rates (1x, 2x, or 4x).

In order to increase DAC the audio quality it is possible for instance to operate the DAC at 2x or 4x the ADC clock (2x or 4x larger oversampling ratio) in combination with the HBW an extra LPF. The extra oversampling will reduce the (inband) noise density. This mode however will come with a cost: the DSP has to interpolate 2x or 4x at the DA-side in order to match the basic sample rate of 8 kHz.

Notice that the pole frequencies of the DAC and LPF are connected with the clock rate (standard 1.152 MHz). Doubling the clock doubles the pole frequencies and the bandwidth.

14.6 PROGRAMMABLE CLOCKS

The SC14452 has three separate clocks whose frequency can be set with bits:

- CLK_CODEC2_REG[SW_DALSRCDC_DIV] for DA path to loudspeaker.
- CLK_CODEC3_REG[SW_DACLASSDCDC_DIV] for DA path to CLASSD.
- CLK_CODEC1_REG[SW_ADCDC_DIV] for AD path from microphone.

Bit values unequal to 00, enables the digital parts of the codec paths.

For **narrow band 32kbit/s ADPCM (G.726)**, the AD and DA shall be set 1.152 MHz resulting in sample rate of 8 kHz. If improved out-of-band noise is required in DA direction one or more of following measures can be

taken:

- The sample Codec DA clock rate can be increased to 16 kHz by setting DA_LSR_SYNC to 1 while the AD sample rate can stay at 8 kHz. The Gen2DSP shall process at 8 kHz and interpolate the 8 kHz samples to 16 kHz, using an interpolation filter. The interpolated samples are stored by the SCP program in a two word software FIFO and the 16 kHz Gen2DSP interrupt routine copies these samples to the DSP_CODEC_OUT_REG. The Codec DA reads the FIFO at a 16 kHz rate.
- CODEC_ADDA_REG[DA_HBW] can be set to 1, divides the DA filter bandwidth by two.
- CODEC_ADDA_REG[LPF_BW] can be set to enable addition Low pass filters after the DAC.

For **wide band 64kbit/s ADPCM (G.722)**, the AD and DA shall be set 2.304 MHz resulting in sample rate of 16 kHz. No noise reduction is required in this mode, because the out-of-band noise contributes from 20 kHz which is not audible.

In case of **32 kHz MP3**, the out-of-band noise is hardly audible and the DA clock stays at 4.608 MHz for 32 kHz sample rate.

14.7 CALLER-ID OPAMP

The SC14452 has no on-chip Caller-id opamp. An external caller-id opamp can be connected to the CIDOUT input as shown in Figure 37 on page 75.

The opamp can be used in three configurations:

- On-hook Caller-id detection, CIDOUT is switched via MICn/CIDOUT to the Codec.
- On-hook ring detection, the CIDOUT goes via 10 bit ADC input and is sampled with 8 kHz. The digital output samples can be read by either Gen2DSP for further signal processing. See paragraph "ADC" on page 112.

To enable the CIDOUT path to the ADC, CODEC_TONE_REG[CID_PD] must be set to 0. The ring detection is also possible via the RINGING input and capture timer. Note that the capture timer accuracy has been enhanced by counting on both edges of the ringing signal compared to rising edge in the previous generations.

- Off-hook caller-id detection, the opamp can be disabled and the in-band caller-id information goes straight via MICh to the Codec.

Table 37: SC14452 Analog frontend supply, reference voltage and power down overview

AFE functions AFE register	MICp/n	VREFp	LSRp/n (AGND)	TONE (external CIDOUT)	CODEC ADC/DAC
CODEC_VREF_REG					
BIAS_PD	0	0	0	0	0
VREF_BG_PD	0	0	0	0	-
AMP1V5_PD	-	0	-	-	-
VREF_PD	-	0	-	-	-
CODEC_MIC_REG	Appl	-	-	-	-
CODEC_LSR_REG	-	-	Appl	-	-
LSR_GND_PD	-	-	-	1	-
CODEC_TONE_REG	-	-	-	Appl	-
CID_PD	1	-	-	0	-
CODEC_ADDA_REG	-	-	-	-	Appl
REFINT_PD	-	-	-	-	0

Appl = Application specific setting (See Note 48: on page 216)

'-' = don't care, preferably in power down to save power

14.8 INTEGRATED HEADSET DETECTION

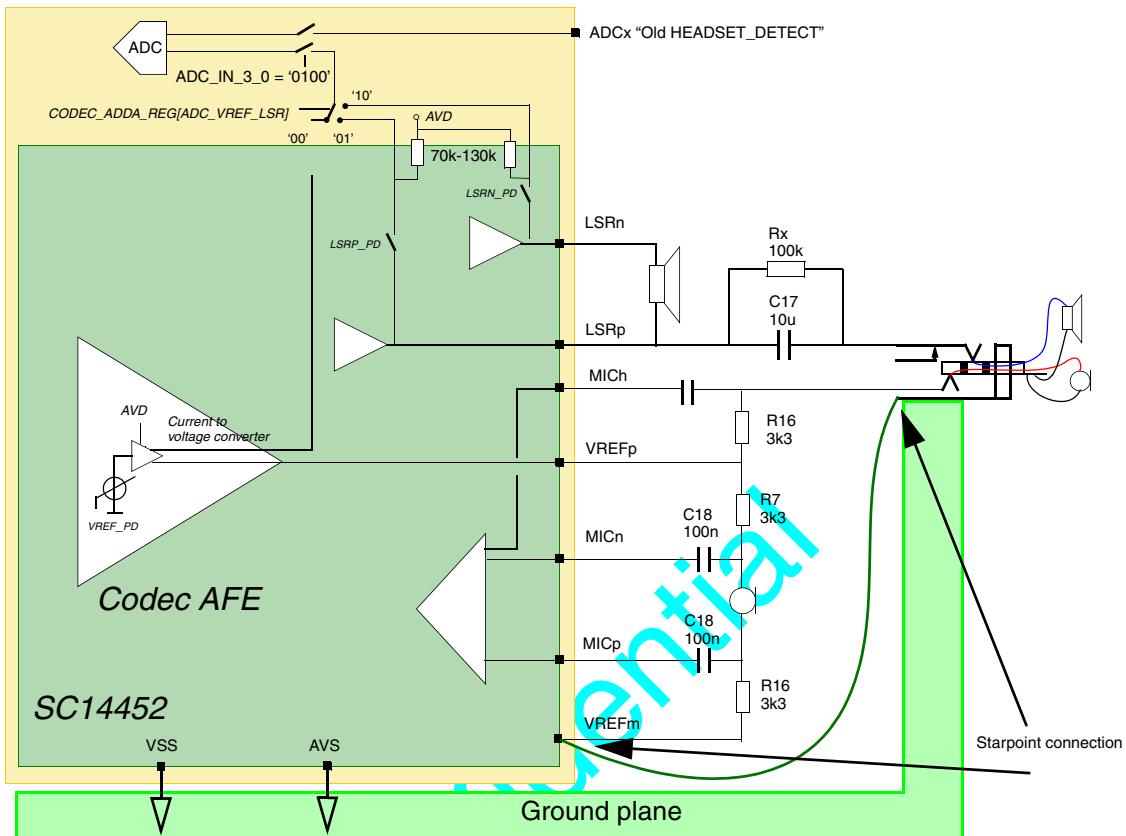


Figure 34 Integrated earphone and microphone detection

The LSRp/LSRn output driver and VREFp buffer circuits can be configured to detect presence/absence of an earphone and none, one or two microphones using the on-chip ADC. See [Figure 34 on page 72](#)

Earphone detection.

Earphone detection is done with the LSRx_PD=1, which configures the LSRp/LSRn driver as a voltage source with 100k output resistance. The output at LSRp/LSRn is measured full scale with the ADC if no loudspeaker connected and will drop to less than 1.2V if a low ohmic loudspeaker with 100 k series resistor over the AC capacitor is connected.

The voltage on the LSRp/LSRn pins can be measured if CODEC_ADDA_REG[ADC_VREF_LSR] =01/10 and ADC_CTRL_REG[ADC_3_0] = 0011

Table 38: ADC values for earphone detection

LSRx_PD	Earphone	ADC value
1	Not connected	V = 1.8V
1	Connected	V < 1.2V
0	Not connected	not defined
0	Connected	not defined

Microphone detection.

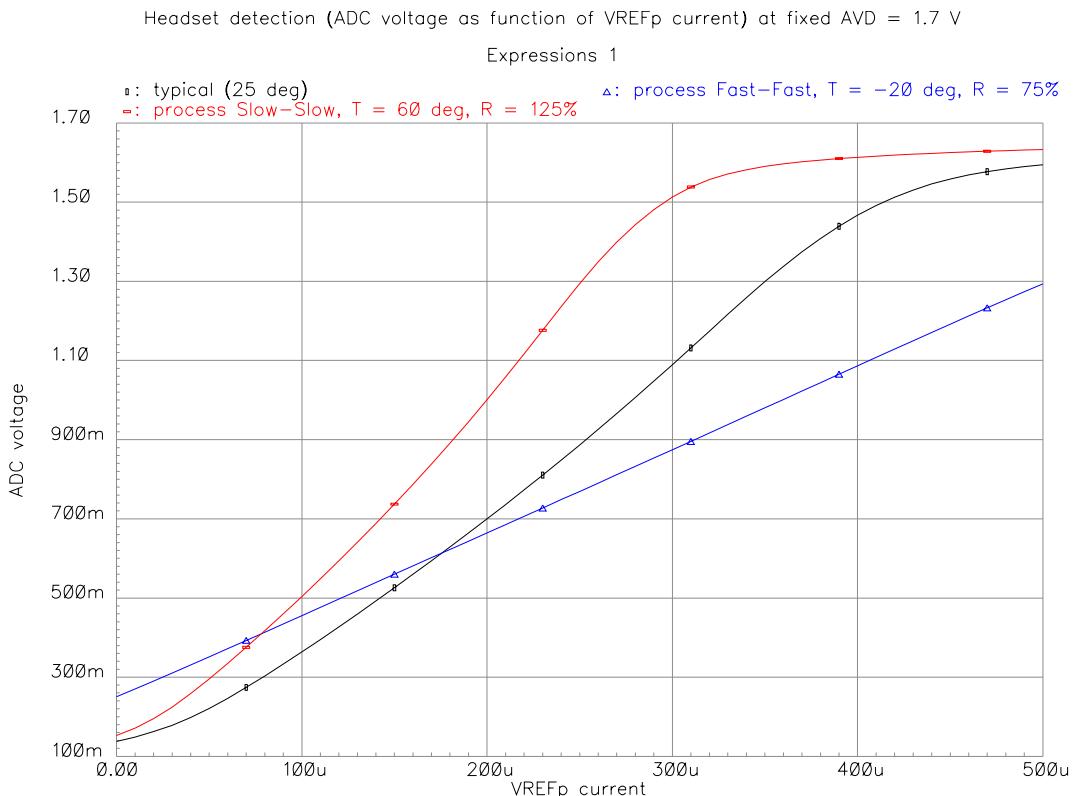
Detection of none, one or two microphones is done by measuring the difference in current driven by the VREFp buffer. For this purpose the VREFp driver has a current to voltage converter which can be switched to an ADC input. The characteristics of the converter is shown in [Figure 35 on page 73](#) with bandgap trimmed to 1.8V.

Due process tolerances and microphone bias current variations, it is obvious that calibration of the thresholds for none, one and two microphones is required.

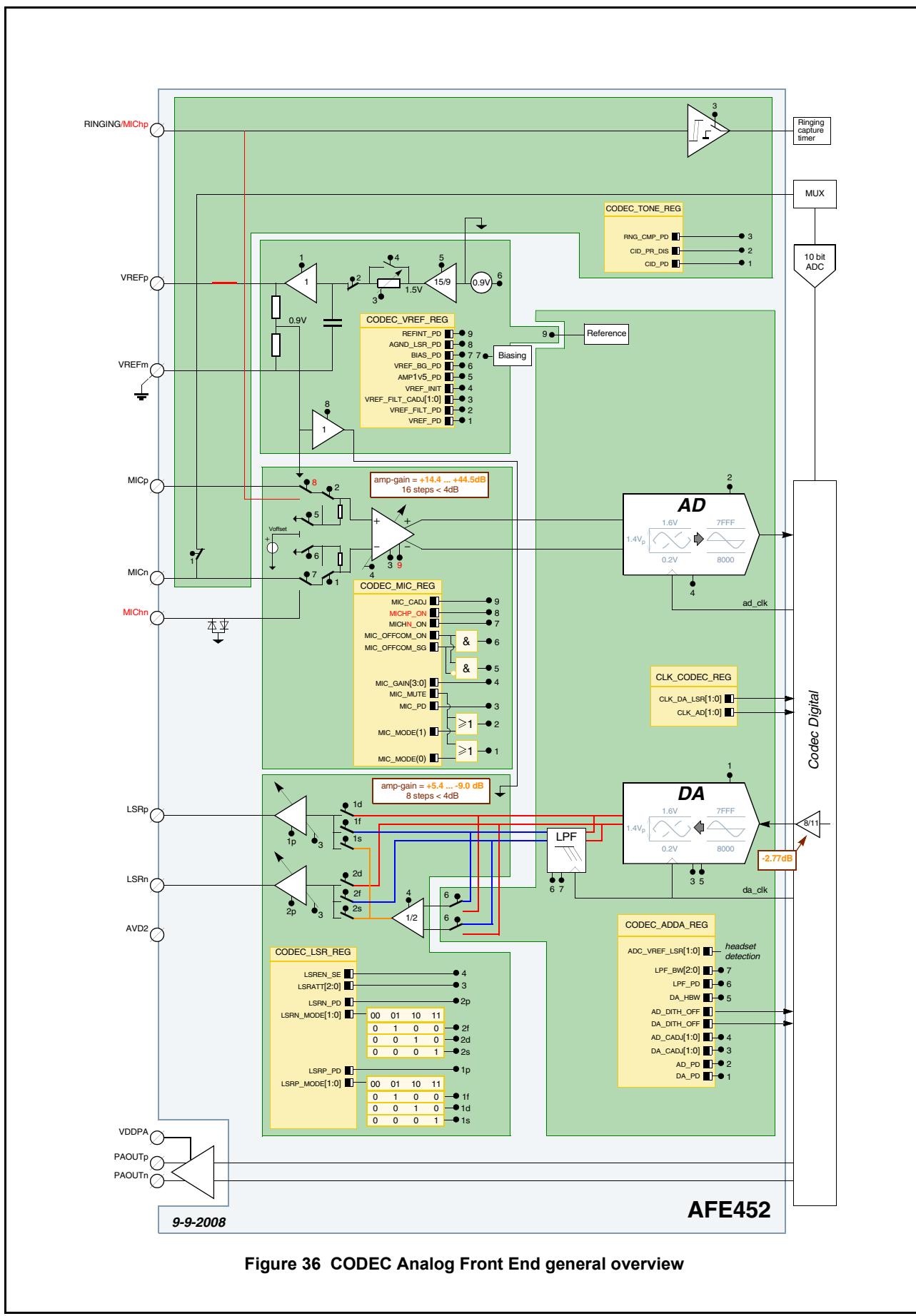
To enable the VREF buffer, set reference voltages according to Table 37: CODEC_VREF_PD[BIAS_PD], VREF_BG_PD, AMP_1V5_PD, VREF_PD] =0.

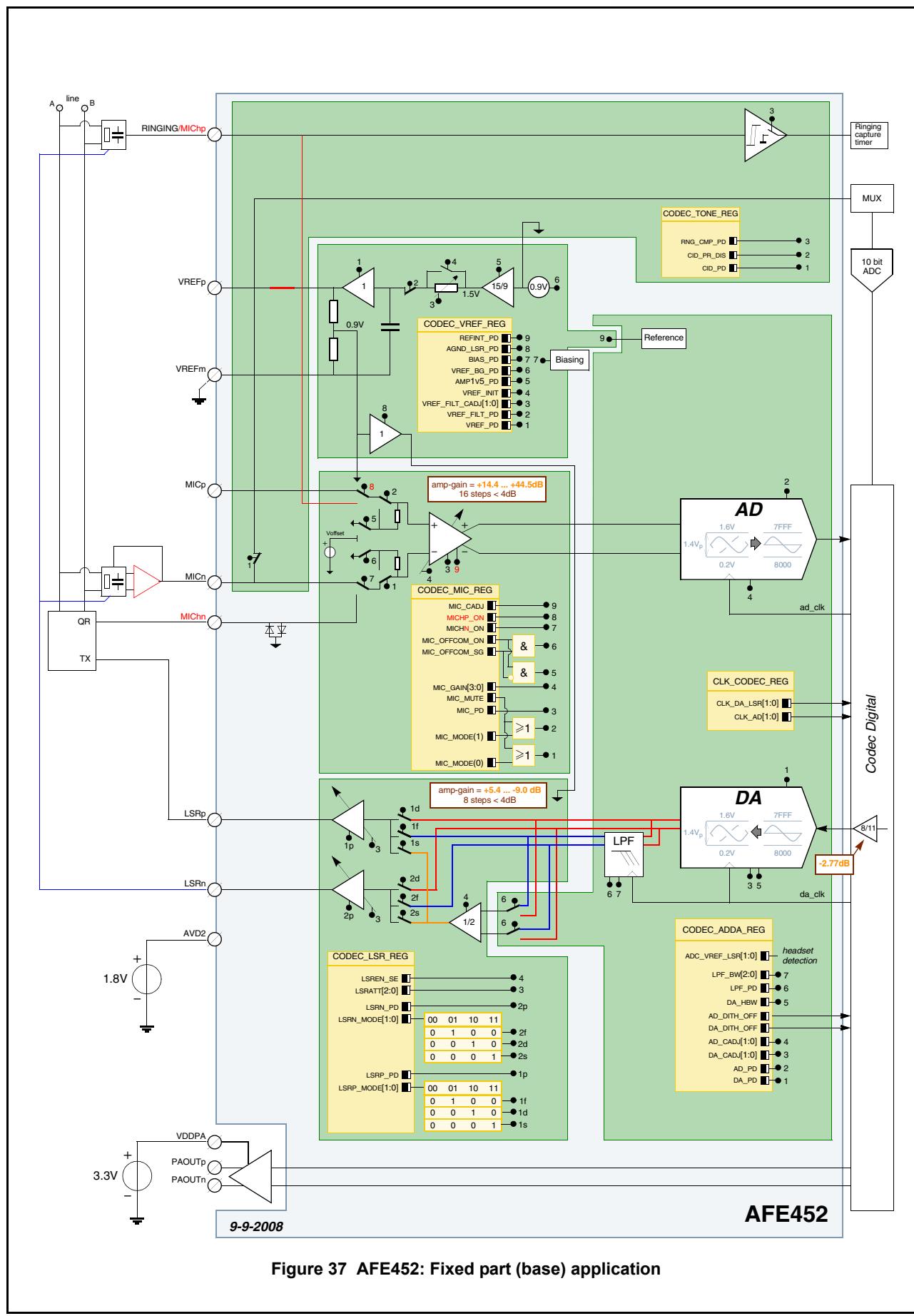
The voltage off the VREF current to voltage converter can be measured if CODEC_ADDA_REG[ADC_VREF_LSR] =00 and ADC_CTRL_REG[ADC_3_0] = 0011.

Note that microphone detection can not be done while VREF_PD =1.



**Figure 35 VREFp Current to Voltage converter characteristics.
Best (bottom), typical (middle) and worst case (top)**


Figure 36 CODEC Analog Front End general overview



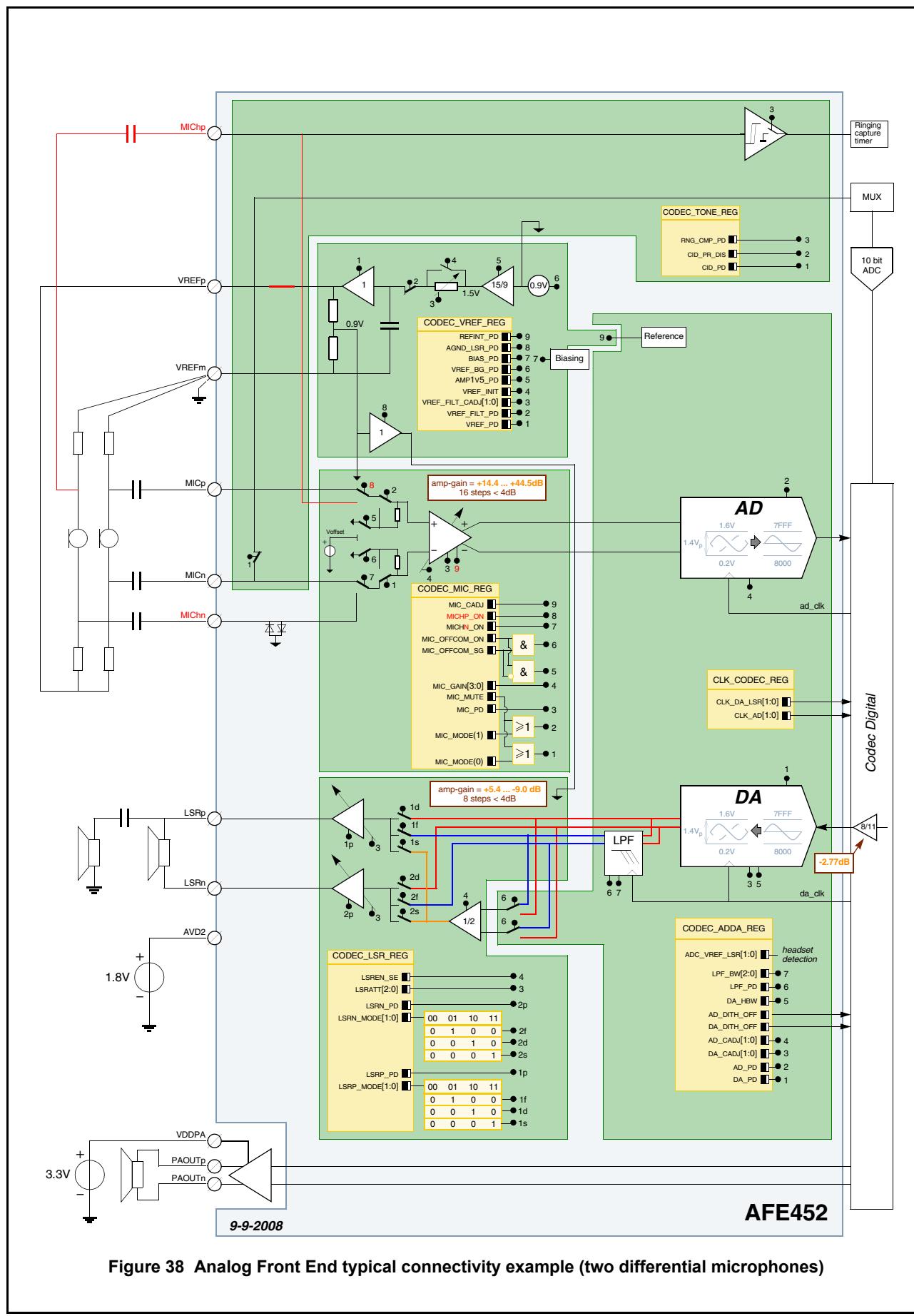


Figure 38 Analog Front End typical connectivity example (two differential microphones)

15.0 CLASSD Amplifier

The SC14452 has a fully integrated single-supply high efficiency switching "Class-D" type audio amplifier as shown in Figure 39.

Features

- Efficiency at 300mW@2V, 500mW@2.5V, 1.125W@3.6V (Li-Ion) into 4 ohm transducer 75%.
- Improved noise reduction.
- Buzzer mode with programmable attenuation
- Digital input mode.
- Separate supply pins for analog drivers
- The output gain and filtering characteristics of the audio signals is performed by the Gen2DSP.
- Anti-plop circuit and output disable mode.
- Independent audio path for CODEC and CLASS-D.
- Outputs can be used as general purpose digital outputs.

15.1 DIGITAL AUDIO MODE PATH SELECTION

Figure 39

For CLASS-D access via DSP_CLASSD_REG, set CLK_CODEC3_REG[SW_DACLASSDCDC_EN]=1.

For CLASS-D access via DSP_CODEC_OUT_REG, set CLK_CODEC3_REG[SW_DACLASSDCDC_EN]=1 and CLK_CODEC2_REG[SW_DALSRCDC_EN]=1.

CLK_CODEC3_REG[SW_DACLASSDCDC_DIV] must be set to 00 for 1.152MHz/8kHz, 10 for 2.304/16kHz, 11 for 4.608/32kHz operation

15.2 ANTI-PLOP CIRCUIT

In power-down state (CLASSD_PD=1), the PAOUTp and PAOUTm are connected to ground. This mid-level state suppresses a "plop" sound during power-up.

15.3 CLIP DETECTION

The CLASS-D amplifier has a clip detection by monitoring the output signal. When output is high (or low) for a number of periods, programmable between 32 and 2048 with bits CLASSD_CLIP, a CLASSD_INT interrupt is generated. This interrupt must be cleared with CLASSD_CLEAR_INT. Clipping can be stopped by adjusting the gain in Gen2DSP. (See **Note 103: on page 303**)

15.3.1 Buzzer mode

For CLASSD buzzer mode the following bits must be set:

- Gen2DSPx: DSP_CLASSD_BUZZOFF_REG[15] = '0' to enable the buzzer mode, or '1' to mute the buzzer. This register must be set to mute the buzzer output without audible side effects.
- CLASSD_BUZZER_REG[CLASSD_BUZ_MODE] =1 to switch from audio to buzzer mode.
- CLASSD_BUZZER_REG[CLASSD_BUZ_GAIN]

can be set as desired.

- Depending on the selected audio path, either DSP_CLASSD_REG or DSP_CODEC_OUT_REG is used as audio output register for the GenDSP.

15.4 DIGITAL OUTPUTS

The PAOUTp and PAOUTm outputs can also be used as GPO ports by setting P2_00_MODE resp P2_01_MODE to 00. The PUPD bits are don't care because the CLASSD is output only and has no pull-up/pull-down registers. The output level can be set with P2_DATA_OUT_REG[1-0], P2_RESET_DATA_REG[1-0], P2_SET_DATA_REG[1-0].

15.5 POWER SAVING

Enabling the CLASSD output stage will consume power, also in "digital output port mode". Refer to Supply currents (Indicative value) (table 349, page 289). For power saving, it is recommended to set CLASSD_PD to 1 if the CLASS-D amplifier is not used. If only the CLASS-D amplifier is used without the CODEC, the CLASS-D audio path selection is recommended to disable all not used analog block (MIC_PD, LSRN_PD, LSRP_PD, AD_PD, DA_PD all set to 1). In general, only blocks that are used, must be switched on.

15.6 NOISE REDUCTION CIRCUIT

The noise reduction circuit combines good PSSR at high signal levels with low noise at low signal levels.

When the level of the input signal is below a certain level (programmable with CLASSD_NR_LVL) for at least a certain time (programmable with CLASSD_NR_TON), the "noise reduction" is enabled.

When the input level comes above the threshold level plus an optional offset (programmable with CLASSD_NR_HYST), the "noise reduction" is disabled.

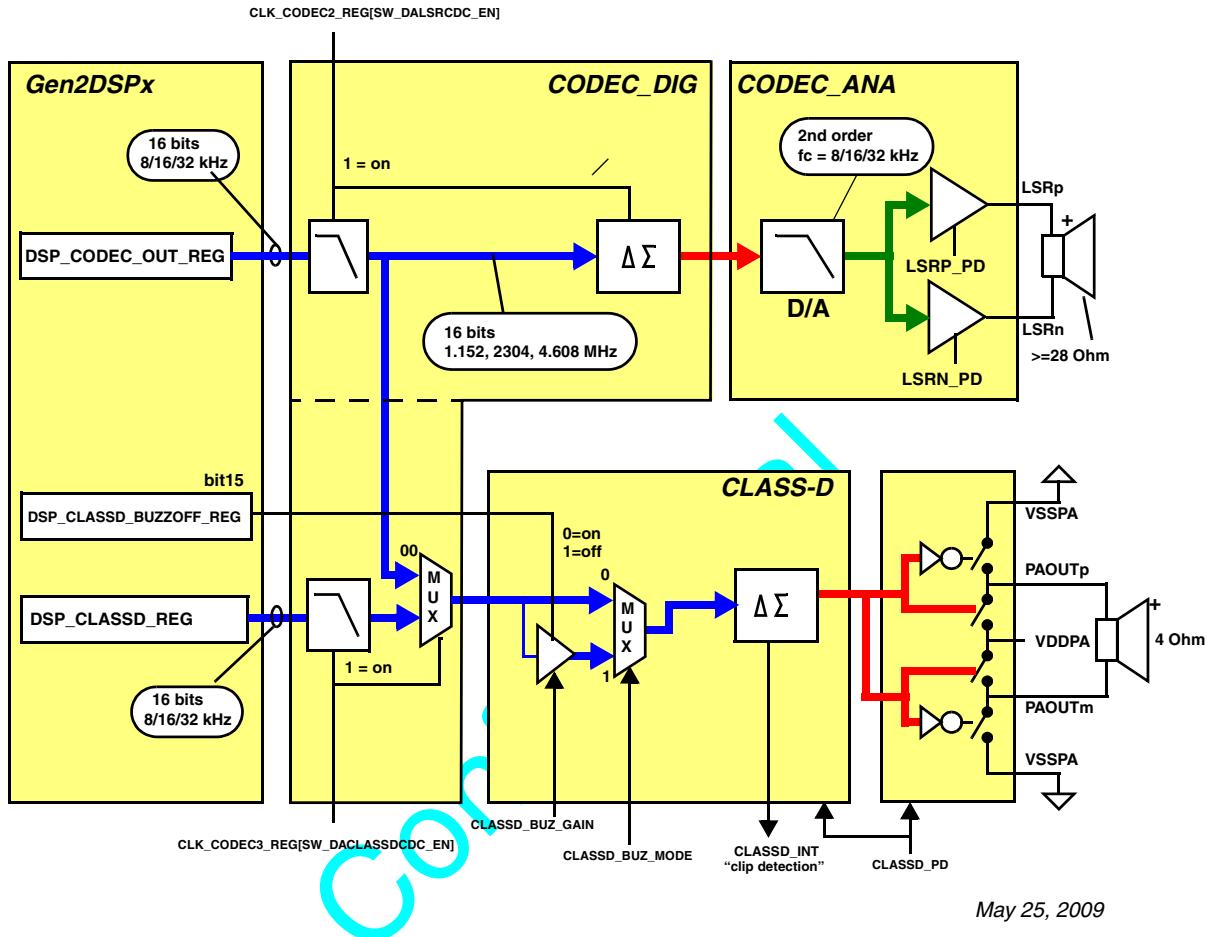


Figure 39 CLASSD amplifier

16.0 Cache Controller

The cache controller (Figure 40) uses non-shared RAM from 0 to 0x5FFF for either iCache (IC), dCache (DC) or non shared memory. See "SC14452 CR16C+ Memory Map" on page 166

The region from 0x8000 to 0x9FFF is used for cache administration buffer or non-shared RAM

Features:

- 1 way associative cache
- Cache size is programmable, 4, 8, 16 kByte.
- Line size is fixed, 4 words of 32 bits.
- Always complete cache line refill
- Memory read and bursts are selectable, 4 or 8 words of 32 bits.
- 16 bits administration record.
1 cache lock bit supported per line
1 valid bits per line
- 14 MSB address bits per line
- Organisation of Instruction cache and Data cache are identical. Data cache is write-through.
- No hardware protection for random access of cache.
- Status message in case of cache violation.
- Access up-to 82.944 MHz (zero wait states)

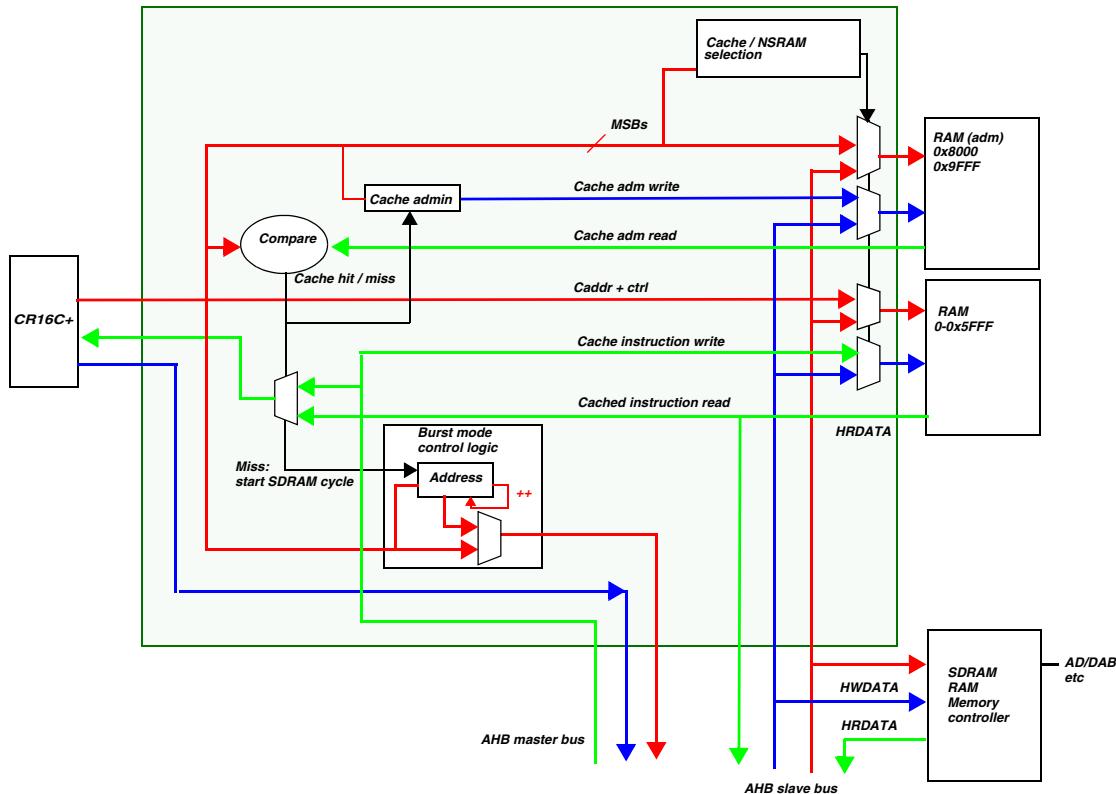


Figure 40 SC14452 Cache controller block diagram

The contents of the IC and DC are automatically loaded by the CR16C+ to maintain copies of recently used instructions and data values. The contents of the IC and DC can also be locked to hold copies of selected memory locations.

The CR16C+ configuration register CFG contains configuration bits to enable an IC and a DC. In addition, lines within the cache may be locked.

The Cache controller registers are:

"CACHE_CTRL_REG (0xFF5006)" on page 192
 "CACHE_STATUS_REG (0xFF5010)" on page 193
 "CACHE_START0_REG (0xFF500A)" on page 192
 "CACHE_LEN0_REG (0xFF5008)" on page 192
 "CACHE_START1_REG (0xFF500E)" on page 193
 "CACHE_LEN1_REG (0xFF500C)" on page 192

"Internal CR16C+ Configuration Register, CFG" on page 82

16.1 CACHE INITIALIZATION

Prior to enabling the cache:

- The administration area in the non-shared RAM must be cleared with zeros.

- The cache size field in the cache control register must be set to zero in order to reset internal registers.

Note 11: The cache size field must be set as part of initialization and must remain stable during operation)

Note 12: Cache initialization and invalidation must be done solely by SW. CR16Cplus instruction CINV is not supported to invalidate the IC or DC.

16.2 THE INSTRUCTION CACHE (IC).

If the CR16C+ fetches an instruction, the IC performs one of these following functions:

- If the IC is enabled (CFG.IC=1) and the fetch address is in the range as specified in CACHE_START0_REG (0xFF500A) + CACHE_LEN0_REG (0xFF5008) or, CACHE_START1_REG (0xFF500E) + CACHE_LEN1_REG (0xFF500C), the cache and the administration memory will be accessed to look up the instruction. CACHE_CTRL_REG[ICACHE_I_SIZE] bit 8 must be kept 0, the read burst length is always 16 bytes.
In case of a cache hit, the instruction is returned to the CR16C+ immediately. In case of a cache miss and the cache line is not locked, the cache burst controller will start a burst read of 16 bytes. The cache controller will write the read data into the IC and update the administration RAM.
- If IC is enabled or disabled but the fetch address is outside the specified range the cache is not used. The cache controller will be fully transparent for the CR16C+.

16.3 DATA CACHE (DC)

Data cache is used when the CR16C+ executes memory read or write access. For data read actions, the DC has the same functionality as the IC.

CACHE_CTRL_REG[DCACHE_B_SIZE] bit 10 must be kept 0, the read burst length is always 16 bytes.

In case the CR16C+ executes a memory write instructions, the DC performs one of the following functions:

- If DC is enabled (CFG.DC=1) and the write address is in the range as specified in CACHE_START0_REG (0xFF500A) + CACHE_LEN0_REG (0xFF5008) or, CACHE_START1_REG (0xFF500E) + CACHE_LEN1_REG (0xFF500C), the DC and the administration memory are accessed to access the cached copy of the data.
- If DC is enabled or disabled, but the memory write address is outside the DC address range, the cache is not used. The DC controller will be fully transparent for the CR16C+.

16.4 CACHE LOCK

16.4.1 Lock per cache

If CACHE_CTRL_REG[CACHE_LOCK]=0 and CFG.DCL=1 or CFG.LIC=1, the DC resp IC will be "frozen". The IC administration contents will not be changed any more, c.q. no new addresses will be added.

But In the DC it is still possible that a cache line is updated, e.g. when a variable, which also resides in cache, is changed by the CR16C+. In that case, the new value is, as usual, written in cache and in memory.

When the cache is locked, its contents will of course be used by the CR16C+ to speed up program execution.

If CACHE_CTRL_REG[CACHE_LOCK]=0, The CACHE_LOCK bits in the administration records don't have a function.

16.4.2 Lock per line

If CACHE_CTRL_REG[CACHE_LOCK]=1 and CFG.DCL=1 or CFG.LIC =1, the administration record CACHE_LOCK "lock per line" is enabled. The CACHE_LOCK and CACHE_VALID bit of the administration record will be set in case of a cache miss. In case of a cache hit, the administration record will NOT be updated. As long as the CACHE_LOCK bit is not set, the corresponding cache line may be updated.

As long as the CFG.DCL=1 or CFG.LIC=1 are active, CACHE_LOCK bits of records of IC or DC will not be reset. Gradually more and more cache lines will be locked as long as CFG.DCL=1 or CFG.LIC =1.

The administration RAM CACHE_LOCK bits can be reset as follows:

- Explicitly by software by clearing to the administration RAM (locations)
- Set CACHE_CTRL_REG[CACHE_LOCK]=0 and CFG.DCL=0 or CFG.LIC =0. With every cache update from now on, the CACHE_LOCK bit of the updated cache record will be reset.

16.5 CACHE ORGANISATION

The table below gives the address ranges that will be allocated for instruction & data cache and administration record as well as the CR16C+ CFG register.

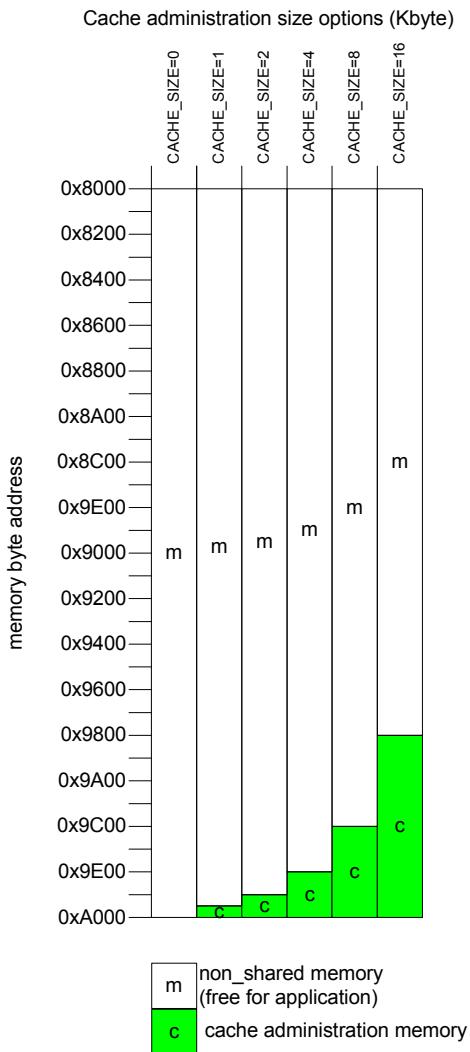


Figure 41 Start addresses for cache data and administration

Table 39: Administration record

Bit	Mode	Symbol	Description	Reset
15	R/W	CACHE_LOCK	Cache line locked	0
14	R/W	CACHE_VALID	Cache line valid	0
13-0	R/W	CACHE_LINE_ADDRESS	Most significant address bits 24-11	0

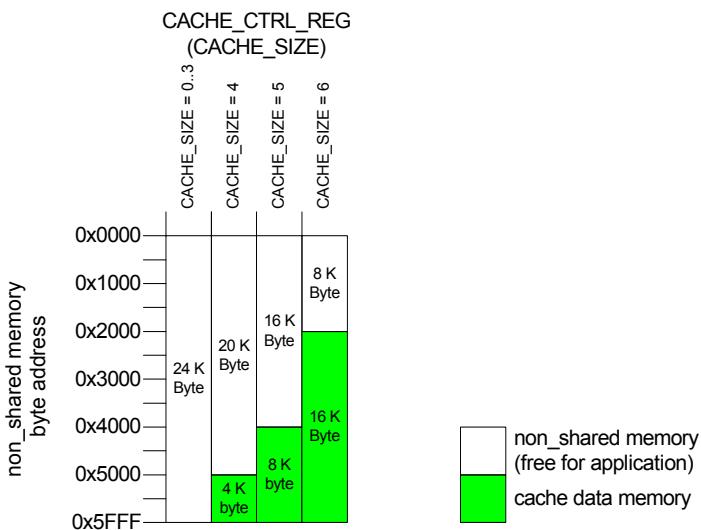


Figure 42 Non shared RAM Size allocation for Cache data buffer

Table 40: Internal CR16C+ Configuration Register, CFG

Bit	Mode	Symbol	Description	Reset
15-10	-			0
9	R/W	SR	Short register bit. Refer to programmers guide	0
8	R/W	ED	Extended dispatch bit. Refer to programmers guide	0
7-6	-			0
5	R/W	LIC	Lock Instruction Cache	0
4	R/W	IC	Instruction Cache Enable	0
3	R/W	LDC	Lock data Cache	0
2	R/W	DC	Data Cache Enable	0
1-0	-			0

16.6 RAM ARBITER

The ram arbiter schedules all read/write accesses from/to the non-shared ram (NS_RAM_a, NS_RAM_c). The following table indicates the priority of all resources that can access the non-shared RAM.

Table 41: RAM arbitration

Priority	non-shared RAM resource
1 (highest)	Cache burst-controller
2	Slave controller (CR16C+, DMA, SDI)
3 (Lowest)	Cache controller

17.0 CR16C+

Refer to the "CR16C Programmer's Reference Manual" (National document number 424521772-101), which may be downloaded from National Semiconductor's web site at <http://www.national.com>.

18.0 APB Bridge

The AMBA Peripheral Bus (APB) bridge interfaces the high speed system bus to low speed peripherals registers of:

- GPIO ports registers
- ADC registers
- ACCESS bus registers
- GPRG registers
- SPI registers
- UART registers
- CCU registers
- DMA registers

The peripheral cores run on a different clock as shown in Figure 6 on page 25.

The purpose of this bridge is to reduce the system bus load and therefore saving power. Because the CR16C+ or DMA access to the peripherals is infrequent, the peripherals are designed for a lower frequency in order to save power and area.

APB Bus speed

The maximum frequency of the APB interface is 41.472 MHz and can be set in register CLK_AMBA_REG[SW_PCLK_DIV]. The frequency depends on the system bus clock HCLK which is set with CLK_AMBA_REG[SW_HCLK_DIV] as shown in Figure 6 on page 25.

For lowest power the SW_PCLK_DIV should be set to its maximum value 4 which divides the HCLK to 4. If more performance during peripheral I/O is required, e.g. with faster DMA/SPI transfers, the SW_PCLK_DIV may be set to 2, giving a maximum APB frequency of 41.472 Mhz if the HCLK is 82.944 Mhz.

In order to avoid internal timing problems, **SW_HCLK_DIV and SW_PCLK_DIV values may never be modified at the same time**. Make sure that PCLK always stays below 41.472 MHz if SW_HCLK_DIV is changed first to get a higher HCLK frequency. (See **Note 44: on page 199**)

Register access

Because the APB bus is a 16 bits bus, **all APB peripheral register write access must be WORD wise**. Read access may be BYTE wise, but the not selected byte in the register will return 0. Writing BYTE wise to the registers will result undefined data in the not selected 8 bits in the 16 bits registers.

Accessing two successive 16 registers in a single 32 bits access gives incorrect results.

19.0 EBI External Bus Interface

The external bus interface (EBI) supports a wide variety memory type and devices as shown in Table 42. Depending on the system requirements, slow/fast low power/higher power devices or device modes can be selected.

Contact SiTel Semiconductor to check whether a specific device or combination of devices is supported by the memory controller.

Features

- Mobile and standard SDR-SDRAM up-to 82.944 MHz
- PSRAM RAM with page and asynchronous access.
- Three timing sets set for Static memories
- Timing parameters for FLASH, Sync FLASH
- Five chip selects including IO expander mode
- 8, 16 bit Static memory bus mode
- 16 bits SDRAM memory bus mode

Refer to [AN-D-179 "SC14450 guidelines for RF design due to EBI bus activity"](#).

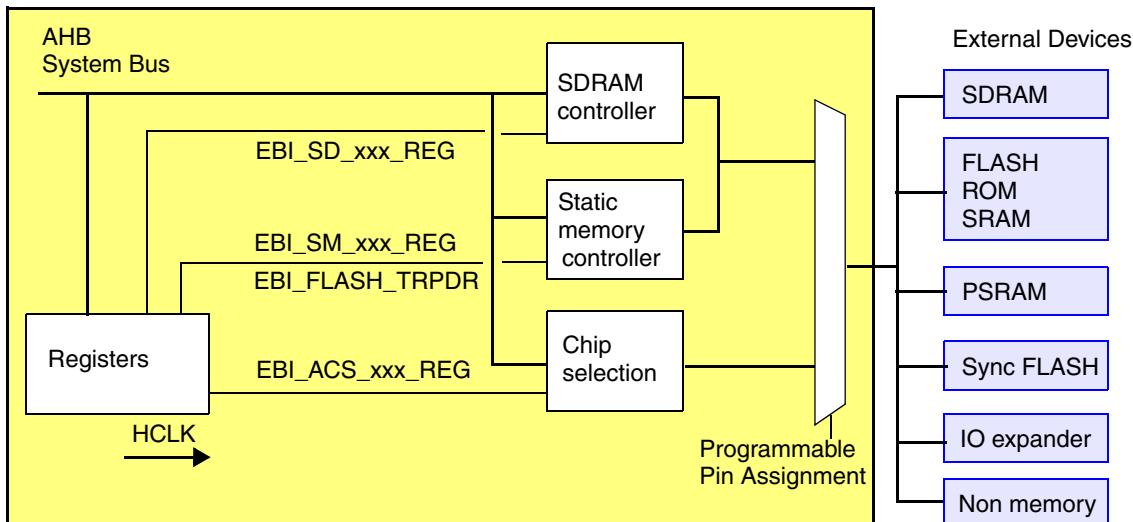


Figure 43: EBI blockdiagram with PPA interface

Table 42: Supported External memory devices

Device	Access type	Minimum Access time @CR16C+ frequency or Synchronous clock	Application	Remark
FLASH ROM SRAM	Asynchronous 16 bits access	96-21 = 75 ns @ 10.368 MHz 48-21 = 27ns @ 20.736 MHz	Handset (slow, low power or faster, high power)	
PSRAM (Pseudo Static)	Asynchronous Asynchronous Page mode	96-21 = 75 ns @ 10.368 MHz 48-21 = 27ns @ 20.736 MHz	Handset (Fast, low power)	Devices: e.g 1Mx16 bits Async: 70 ns Page: 20 ns
Synchronous FLASH	Asynchronous Asynchronous Page mode	96-21 = 75 ns @ 10.368 MHz 48-21 = 27ns @ 20.736 MHz	Handset (Fast, low power)	Devices: e.g 1Mx16 bits Async: 70 ns Page: 20 ns Sync: 104 MHz
	Synchronous Burst mode (read only)	HCLK = SDCLK = 82.944 MHz (12.05 ns)		

Table 42: Supported External memory devices

Device	Access type	Minimum Access time @CR16C+ frequency or Synchronous clock	Application	Remark
SDR- SDRAM (Including Mobile SDRAM)	Synchronous Burst mode	HCLK = SDCLK = 82.944 MHz (12.05ns)	Basestation (Fast, higher power)	

19.1 CHIP SELECTION

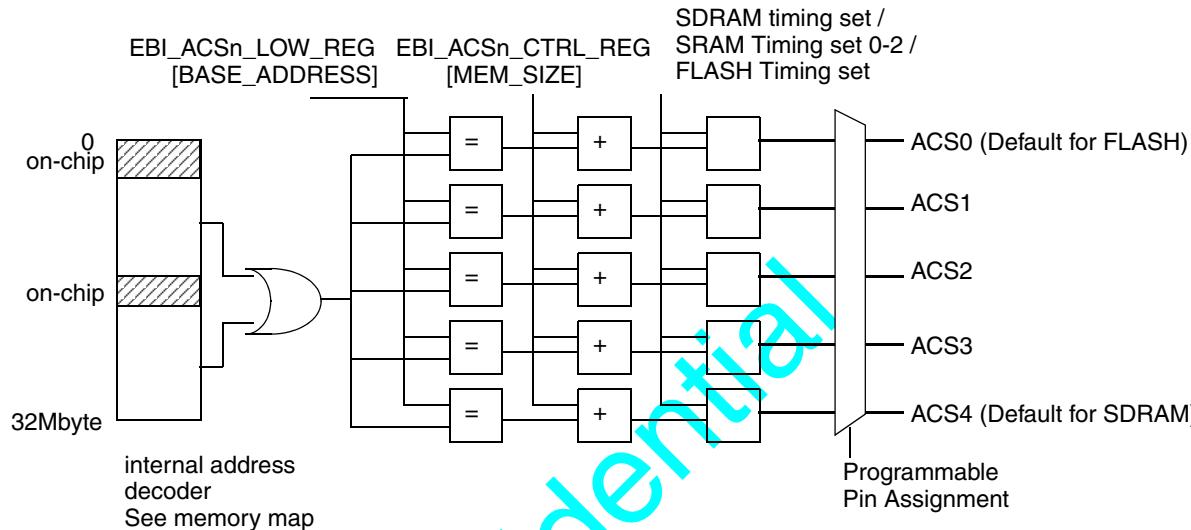


Figure 44 Programmable chip selects

The EBI supports up-to 5 programmable active low chip selects ACS0-ACS4 which are available through PPA port pins. The total addressing range is 32Mbyte minus the on-chip Refer to "Memory map overview" on page 166 for the available external ranges.

Due to the high speed requirements ACS4 has a fixed pin assignment and should be used for SDRAM. The other chip selects can be used freely for other external devices.

The following registers are used for chip selection and timing parameters

- **EBI_ACStn_LOW_REG** (n=0-4) Chip select n low base address registers in offset units of 64kbyte
- **EBI_ACStn_CTRL_REG** (n=0-4) Address Mask Registers specify memory size assigned to a particular chip select in multiples of 64kbyte. The EBI supports Static memory sizes from 64KB to 32MB. Bits 6:5 specify the type of memory type connected either SDRAM, SRAM, FLASH Bits 8:7 specify the Static Memory Timing Register set (0-2) with which this particular memory is associated.

Chip Select priorities.

Internal chip selects have priority over external chip selects. There are NO priorities in overlapping external

chip select regions. This means that more than one chip select may go low if base addresses in combination with memory size are overlapping. For read operation this will cause a bus conflict and must therefore be avoided in order to guarantee the life time of the external devices.

Base address boundaries

The base address specified for a particular chip select must be in terms of the block size chosen for that particular chip select. If the EBI_ACStn_LOW_REG base address value is not aligned with the block size, the base address will be rounded down to the nearest size boundary. For example 8Mbyte FLASH with base address set to 0x40.000 and size set to 4M will select a range 0x40.0000-0x7F.FFFF but selecting a size of 8M will result in a range of 0-0x7F.FFFF. So 8Mbyte FLASH must start at 0 or 0x80.0000. 16M SDRAM must start at 0 or 0x100.0000.

Note however that addresses 0-0x1.FFFF (128k), 0xFE.F000-0xFF.FFFF (68k) and 0x100.0000-107.FFFF (512k) are not accessible off chip. (see Table 67). External data stored at these locations can only be accessed after remapping the chip selects to other addresses. This must be done by a program running in on-chip RAM.

Note also that the CR16C+ address range is 32Mbyte.

but code can only be executed from 0-0xFFFF.

Table 43 show a typical configuration with 8M standard FLASH and 4 Mbyte PSRAM (with asynchronous and page mode), LCD and camera

Table 44 shows a configuration with 16 Mbyte FLASH or fast SDRAM and IO devices (LCD, camera) mapped above 16M. Accessing these devices is less efficient. See also SC14452 CR16C+ Memory Map (table 67, page 166).

Table 45 shows a configuration with 32 Mbyte SDRAM but without external devices.

Table 43: Chip select configuration 1

Base Address	Chip select	Device
0x002.0000	ACS3 (64k min)	IO expander EBI_SDREFR_REG [ACS3_IOEXP] =1
0x002.FFFF		
0x004.0000	ACS1	256kbyte LCD
0x007.FFFF		
0x008.0000	ACS4	256kbyte Camera
0x00F.FFFF		
0x040.0000	ACS2	4Mbyte PSRAM
0x07FFFF		
0x080.0000	ACS0	8Mbyte FLASH Code/data
0x0FF.FFFF		

Table 44: Chip select configuration 2

Base Address	Chip select	Device
0x000.0000	ACS0	16 Mbyte FLASH or 16Mbyte SDRAM Code/data
0x0FF.FFFF	ACS4	
0x108.0000	ACS3 (64k min)	IO expander EBI_SDREFR_REG [ACS3_IOEXP] =1
0x108.FFFF		
0x110.0000	ACS1	Device max 512kbyte if AD19 up is used for SDRAM
0x117.FFFF		
0x020.0000	ACS2	Device max 512kbyte if AD19 up is used for SDRAM
0x027.FFFF		

Table 45: Chip select configuration 3

Base Address	Chip select	Device
0	ACS4	32Mbyte SDRAM -16M Code/data -16M Data
0x0FF.FFFF		
0x100.0000		
0x1FF.FFFF		

19.2 IO EXPANDER WITH ACS3

Figure 45 shows ACS3 can be used a glueless clock for an IO expander connected to the DAB[15-0] data bus.

EBI_SDREFR_REG[ACS3_IOEXP] must be set to 1 to get the function (ASC3 OR WRn). On the rising edge of WRn within the ASC4 range, data on DAB[15-0] is clocked into the IO expander.

To read data from an IO expander, any chip select can be used.

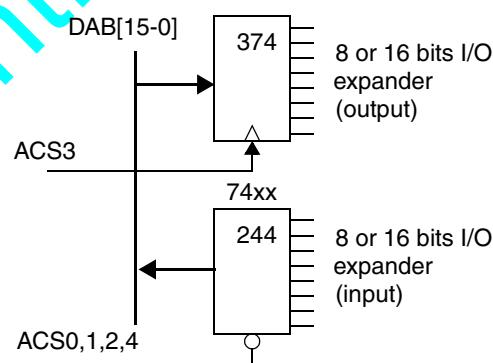


Figure 45 IO expander using ACS3

19.3 STATIC MEMORY CONTROLLER

19.3.1 Memory types

The Static memory controller supports the following devices:

Standard asynchronous SRAM, FLASH, ROM or non memory devices

- Refer to Figure 50 to Figure 52 for connection diagrams

PSRAMs with Page/asynchronous mode
(e.g Micron MT45W1MW16BDGB)

- See Figure 51
- Page mode read, page mode write
Pin SF_ADV is not used, don't care, ZZ# not supported, only SW access of Configuration register)

- Asynchronous read and write supported
Pin SF_ADV is not used

PSRAMs with burst/page/asynchronous mode (e.g Micron MT45W1MW16PA)

- Burst read, burst write is not supported, so this device is not recommended unless the price is cheaper than the page/asynchronous device.
- Page mode read, page mode write
Connect device pin ADV# pin to SF_ADV pin
- Asynchronous read and write
(pin SF_ADV is LOW)

Synchronous FLASH

- See Figure 51
- Synchronous read mode supported.
Connect device pin ADV# pin to SF_ADV pin.
- Synchronous write mode not supported
- Asynchronous write mode.
Connect device pin ADV# pin to VSS.
- Asynchronous page read operation is only supported if the device ADV# pin is not required to be kept low during the entire asynchronous page read process.

The reason is that the memory controller is designed for synchronous FLASH.

In burst mode the ADV# latches the address. In page mode the same latch is used and the ADV# rising edge also latches the address.

19.3.2 Control registers

The timing of the Static memories are controlled via registers:

- EBI_SMTMGR_SET0_REG,
EBI_SMTMGR_SET1_REG,
EBI_SMTMGR_SET2_REG (0xFF0094, 0xFF0098, 0xFF009C) (table 226, page 236)
- EBI_FLASH_TRPDR (0xFF00A0) (table 227, page 237)
- EBI_SMCTLR_REG (0xFF00A4) (table 228, page 237)

External databus width

The static memory control register **EBI_SMCTLR_REGx** sets the bus width for each of the three selected timing sets to either 8 or 16 bits wide databus access.

Timing sets

Up-to three timing sets can be selected in the Static Memory Timing Register **EBI_SMTMGR_SET0-2**. One or more chip selects can have the same timing set. Parameters in these registers determine read and write cycle duration, address setup and hold times, page mode selection and page sizes.

As an example a fast PSRAM can be selected with tim-

ing set 0 while a slower LCD controller is driven with set 1.

Asynchronous Page mode access

For PSRAM page mode timing, timing set 2 can be used with default values in EBI_SMTMGR_SET2_REG (Table 226 on page 236). But the parameters can be tuned to meet the timing requirements:

- EBI_PAGE_MODE must be set to '1'.
- EBI_PAGE_SIZE can be set to 4, 8, 16 or 32 word per page. Depending on the page size, a change of address line A2, A3, A4 or A5 initiates a new page access.
- The initial access T_RC and subsequent accesses T_PRC programmable from 0-15 CR16C+ clock cycles.

(e.g 70 ns interpage 20 ns intra page access CR16C+ = 82.944 MHz. T_RC = 8-1, T_PRC = 4-1)

Changing from asynchronous read mode to synchronous read mode (sync FLASH only)

- Change the FLASH device read mode to synchronous mode by writing to the FLASH internal configuration register.
- Set the "EBI_PAGE_MODE" bit in the Static Timing Register Set0/1/2 (SMTMGR_SET0/1/2) to enable the **burst read** (the power-on default mode should be non-page/burst mode).

You can specify synchronous FLASH timing parameters by programming the Static Memory Timing Register - Set 0 (SMTMGR_SET0), Static Memory Timing Register - Set 1 (SMTMGR_SET1), or Static Memory Timing Register - Set 2 (SMTMGR_SET2), depending on which of the three register sets should control the synchronous FLASH.

You can use the following synchronous FLASH timing parameters:

- Read access time (T_RC), Bits 5:0 of SMTMGR_SET0/1/2 specify the initial read data time in AHB clock cycles. The value of the T_RC should be multiples of the AHB clock frequency to FLASH clock frequency ratio.
- Burst read-cycle time (T_PRC), Bits 22:19 of the Static Memory Timing Register SMTMGR_SET0/1/2 specify the burst read-cycle time, which is equal to the AHB clock frequency to FLASH clock frequency ratio
- Write address setup time (T_AS), Bits 7:6 of the Static Memory Timing Register SMTMGR_SET0/1/2 specify the write address setup time in AHB clock cycles.
- Write pulse width (T_WP), Bits 15:10 of the Static Memory Timing Register SMTMGR_SET0/1/2 specify the write pulse width in AHB clock cycles.
- Write recovery time (T_WR) Bits 9:8 of the Static Memory Timing Register SMTMGR_SET0/1/2 spec-

ify the address hold time (in AHB clock cycles) when the write enable goes high.

- Bus turnaround time (T_BTA) "C Bits 18:16 of SMTMGR_SET0/1/2 force the EBI to insert T_BTA number of cycles between "read to write", or "write to read", or "read to read when chip-select changes". (see Figure 125).
- rp pin high to read/write delay (T_RPD) Bits 11:0 of the FLASH_TRPD register specify RP High-to-output delay in AHB clock cycles.

The Command Register Enable (CRE) pin is not supported. Instead, the software access sequence for accessing the command register shall be used. Consequently the PSRAM's Deep Power Down mode can not be set.

Confidential

19.4 SDR-SDRAM CONTROLLER

This section discusses the details of how the EBI supports Mobile/SDR-SDRAM devices. No DDR_SDRAM is supported.

The SDRAM controller has the following registers

- EBI_SDEXN_MODE_REG (0xFF00AC) (table 218, page 233), the Mobile SDRAM extended mode register
- EBI_SDCONR_REG (0xFF0000) (table 219, page 233), the SDRAM configuration registers
- EBI_SDTMG0R_REG (0xFF0004) (table 220, page 233), the SDRAM timing register 0
- EBI_SDTMG1R_REG (0xFF0008) (table 221, page 234), the SDRAM timing register 1
- EBI_SDCTLR_REG (0xFF000C) (table 222, page 234), the SDRAM control register
- EBI_SDREFR_REG (0xFF0010) (table 223, page 235), the SDRAM refresh and auxiliary control register

Power-On Initialization

The Mobile/SDR-SDRAM controller follows the JEDEC-recommended SDR-SDRAM power-on initialization sequence as follows:

1. Apply power and start clock; maintain a NOP condition at the inputs
2. Maintain stable power, stable clock, and NOP input conditions for a minimum of T_INIT clock cycles
3. Issue precharge commands for all banks of the device.
4. Issue auto-refresh commands, depending on the value NUM_INIT_REF in the programmable register
5. Issue a set-mode register command to initialize the mode register
6. Issue a set-extended-mode register command to initialize the extended mode register (to both SDRAM and Mobile SDRAM)

The EBI performs a power-on sequence of the SDRAM under these circumstances:

- Immediately after reset.
- When EBI_SDCTLR_REG (0xFF000C) bit 0 is set, the EBI resets the bit when it comes out of initialization.

All SDRAM read/write requests that occur during initialization are queued in the memory controller.

Figure 46 illustrates the commands issued to the SDRAM by the controller during the power-on initialization.

The T_INIT, T_RP, and T_RCAR are the default values which you can use for initialization. If you feel that the reset time of the system is long enough to take care of the T_INIT time, then you can assign a value of zero to the T_INIT parameter.

The EBI initializes the SDRAM after reset using the specified default timing parameters. After reset, if you feel that these timing parameters are not adequate, then you can program the timing parameters accordingly and then program the EBI_SDCTLR_REG (0xFF000C) bit INITIALIZE to 1 and wait until it becomes 0 again. This forces the EBI to initialize the SDRAM. The T_MRД is fixed at a value of 3 clock cycles, according to the JEDEC standard.

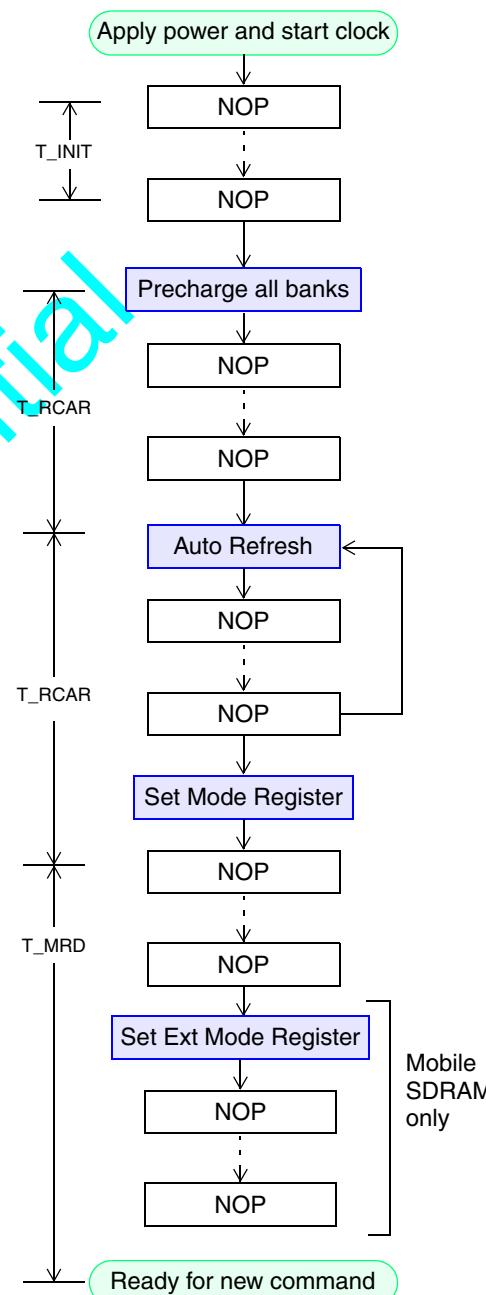


Figure 46 SDR-SDRAM Power-On Command Sequence

equation:

$$t_{REF} > 50 * (1/f)$$

Typically, the refresh cycle is 15.6us or 7.8us, depending on the refresh rate; Table 47 summarizes this:

Table 47: Minimum SDRAM clock frequency Requirement

Number of Rows	T_REF	Min Frequency
64K	(64ms-(50/f))/65536	51Mhz
32K	(64ms-(50/f))/32768	26Mhz
16K	(64ms-(50/f))/163904	13Mhz
8K	(64ms-(50/f))/8192	6Mhz
4K	(64ms-(50/f))/4096	3Mhz
2K	(64ms-(50/f))/4096	5Mhz

The refresh logic in the EBI is inactive when the EBI forces the SDRAM into self-refresh or power-down mode.

Figure 47 illustrates the command sequence issued by the SDRAM controller when performing an auto-refresh.

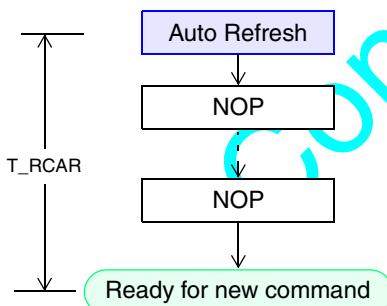


Figure 47 SDR/Mobile-SDRAM Auto-Refresh Command Sequence

Self-Refresh

You can put the SDRAM into self-refresh mode, at which point the SDRAM retains data without external clocking and auto-refresh. Figure 48 illustrates the command sequence issued by the SDRAM controller to initiate, maintain, and exit the self-refresh mode.

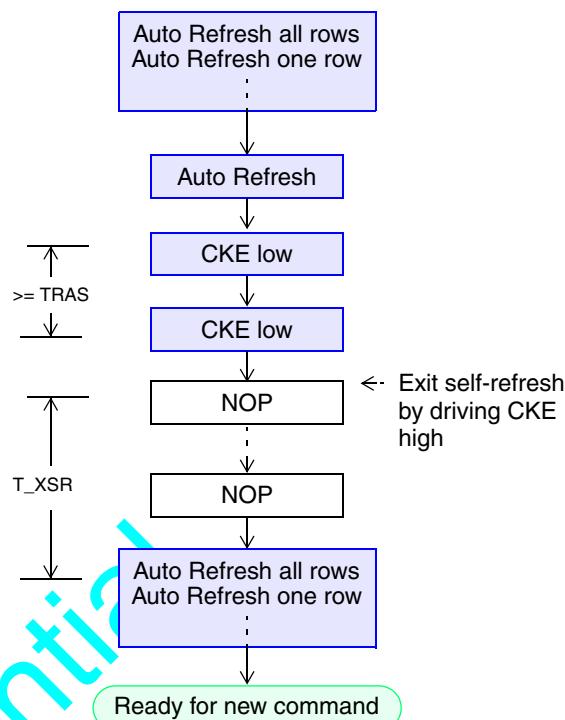


Figure 48 SDR/Mobile-SDRAM Self-Refresh Command Sequence

You can force the EBI to enter self-refresh mode by programming bit 1 of EBI_SDCTRL_REG (0xFF000C). The EBI forces the SDRAM to come out of self-refresh mode when bit 2 of the EBI_SDCTRL_REG (0xFF000C) is set to 0. You can set this bit to 0 by either programming the SDRAM control register.

Bits 4 and 5 of the EBI_SDCTRL_REG (0xFF000C) specify the type of refresh done by the EBI just prior to entering self-refresh mode and just after entering self-refresh mode. Programming bit 4 of the SCTRL to 0 forces the EBI to refresh only one row before putting the SDRAM into self-refresh mode. The default value of 1 forces the EBI to perform auto-refreshes for all rows. Bit 5 does the same, except that it controls the refresh pattern just after coming out of self-refresh mode.

Since it takes time between programming the control register bit to the SDRAM entering self-refresh mode, the EBI provides a read-only register bit, EBI_SDCTRL_REG (0xFF000C) bit 11 to indicate that the SDRAM is already in self-refresh mode. If you want to gate off the clock to the EBI when the SDRAM is in self-refresh mode, you should ensure this bit is set to 1 before you stop the clock.

The SDRAM must remain in self-refresh mode for a minimum period of T_RAS and can remain in self-refresh mode for an indefinite period of time. After the SDRAM exits self-refresh mode, the EBI issues NOP commands for T_XSR before it issues any other com-

mand. The T_RAS and T_XSR are programmable register values and have default values. These registers can be programmed only once after reset.

When an AHB read/write request to the SDRAM occurs while the SDRAM is in self-refresh mode, the EBI generates dummy ready signals to the AHB without accessing external memory.

Power-Down

The SDRAM can be put into power-down mode to save power by setting EBI_SDCTRLR_REG (0xFF000C) bit 2 should be 0 to bring the SDRAM out of power-down mode.

Figure 49 illustrates the command sequence issued by the SDRAM controller to initiate, maintain, and exit power-down mode.

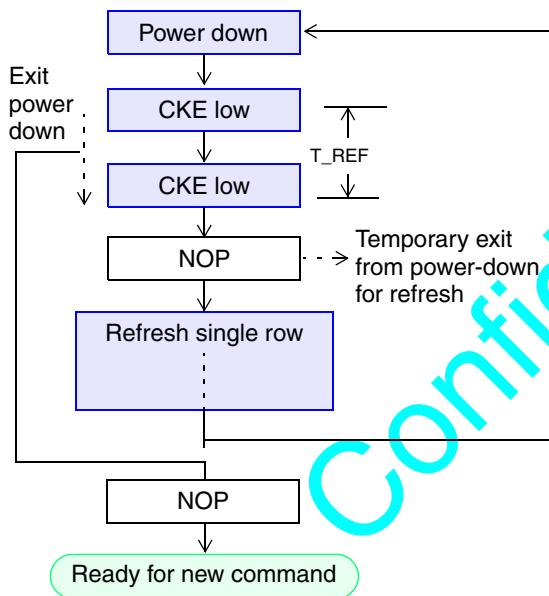


Figure 49 SDR/Mobile-SDRAM Power down Command Sequence

When in SDRAM power-down mode, the EBI keeps switching the device back and forth between power-down and refresh mode. It remains in power-down for a T_REF period of time, then comes out of power-down and does a single-row refresh; then it again goes into power-down mode.

The EBI keeps the SDRAM in this periodical power-down/refresh/power-down sequence until it is commanded to exit power-down mode (by programming bit 2 of EBI_SDCTRLR_REG (0xFF000C) to 0).

When an AHB read/write request to the SDRAM occurs while the SDRAM is in power-down mode, the EBI brings the SDRAM out of power-down mode and issues the read/write access to the SDRAM. The EBI

then puts the SDRAM back to power-down mode after the read/write access.

Address multiplexing

The EBI performs address multiplexing in order to convert the address generated by the host interface to the SDRAM row address, column address, and bank address.

Although the EBI support up-to 15 columns addresses and up-to 16 row addresses for SDRAM, the maximum number is limited by the addressable range of 32 MByte of the CR16C.

Selection of row and column sizes is done in EBI_SDCONR_REG bits S_ROW_ADDR_WIDTH EBI_SDCONR_REG bits S_COL_ADDR_WIDTH.

Table 48 shows 32Mbte SDRAM host address mapping with column address C0-C8, row address R0-R12, and bank address B0-B1.

The EBI supports a minimum page size of 256 words; thus, only bits C0-C7 are fixed. The position of all other address bits can change, depending on how many column, row, and bank address bits are used.

Table 48: Example 32Mbyte SDRAM with 4 banks, 16 bits bus (mt48LC16M16A2)

Physical Address	A24-A12	A11-A10	A9-A1	A0
SC14452 pins	AD13, AD12- AD1	SDBA1- SDBA0	AD9- AD1	
SDRAM pins	ROW A12-A0	BA1-BA0	COL A8-A0	

Note that pin AD13 is muxed with P2[7]/AD13.
Pin SDA10 is directly connected to SDRAM pin A10.

19.5 SDRAM INITIALIZATION EXAMPLES

```

#define EBI_ACS4_LOW_REG 0x1080000
SetPort(P0_15_MODE_REG, PORT_OUTPUT, PID_ACS4);
SetPort(P1_12_MODE_REG, PORT_OUTPUT, PID_SDCLK);

//enable all SD signals
SetPort(P1_11_MODE_REG, PORT_OUTPUT, PID_SDCKE);
SetDword(EBI_ACS4_LOW_REG, ACS4_START);
SetWord(EBI_ACS4_CTRL_REG, 0x08); // 8M
//=====================================================
// mt48lc16m16a2 is 32 Mbyte 4 banks of 8M (only 8M used)
// SET THE EXTENDED MODE REGISTER EQUAL TO
// THE NORMAL MODE REGISTER, WRITE TWICE
//=====================================================
SetDword(EBI_SDEXN_MODE_REG, 0x00000022);

SetDbits(EBI_SDCONR_REG, S_COL_ADDR_WIDTH, 8);
SetDbits(EBI_SDCONR_REG, S_ROW_ADDR_WIDTH, 12);
SetDbits(EBI_SDCONR_REG, S_DATA_WIDTH, 0); //16 bits
SetDbits(EBI_SDTRG0_REG, T_RCAR, 7);
SetDbits(EBI_SDTRG0_REG, T_RCSD, 7); // active to active 60
ns = 6
//=====================================================
// initialize the SDRAM
//=====================================================
SetWord(EBI_SDCTRLR_REG,
        (GetWord(EBI_SDCTRLR_REG) | INITIALIZE));
while ((GetByte(EBI_SDCTRLR_REG) & INITIALIZE) == INITIALIZE);

```

19.6 BUS SHARING

Address and data bus sharing between SDRAM and static memory devices or asynchronous peripherals is possible in a limited way.

Table 50 shows SDRAM and static memory bus sharing. No control signals can be shared because of SDRAM refresh cycles. Note that all SDxxxx signals are only connected to the SDRAM.

The address bus is shared. Note that SC14452 AD1 is connected to the address pins AD0 of both devices. SDRAM address A10 is directly connected to pin **SDA10**, needed for auto precharging

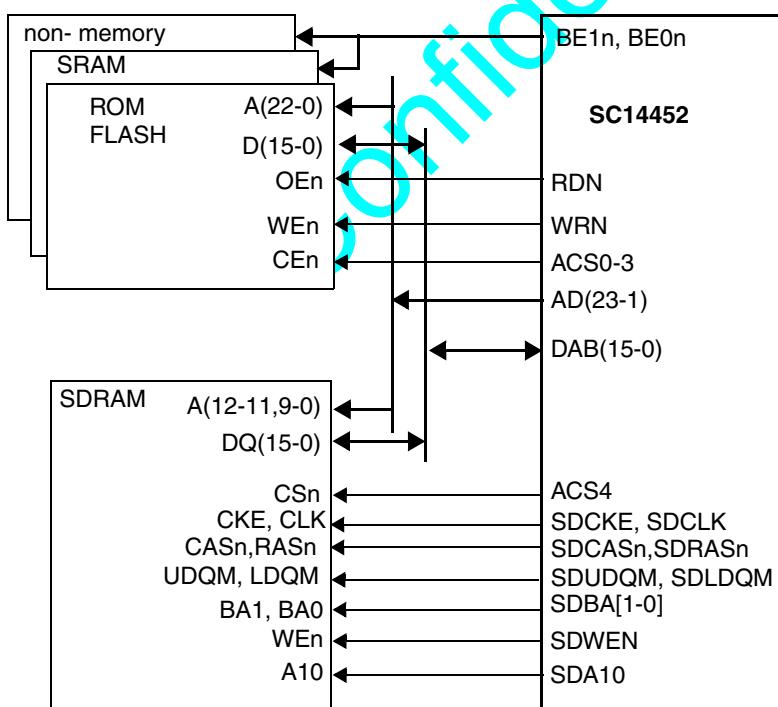
Table 49 shows how the address bus can be shared between Static memory and SDRAM.

Table 51 shows PSRAM and static memory bus sharing. ACS2 is used with a different timing set to support paging mode.

Table 52 shows Synchronous FLASH and static memory bus sharing.

Table 49: Static memory range selection

Address lines	Range	Purpose
AD12-AD1	8kbyte	Static memory and SDRAM address signals. SDA10 is directly connected to device A10.
AD13	16kbyte	
AD14	32kbyte	
AD15	64kbyte	
AD16	128kbyte	
AD17	256kbyte	
AD18	512kbyte	
AD19	1Mbyte	
AD20	2Mbyte	
AD21	4Mbyte	
AD22	8Mbyte	
AD23	16Mbyte	



January 23, 2008

Figure 50 External Bus sharing with SDRAM and up-to 16M total static memory or non memory devices

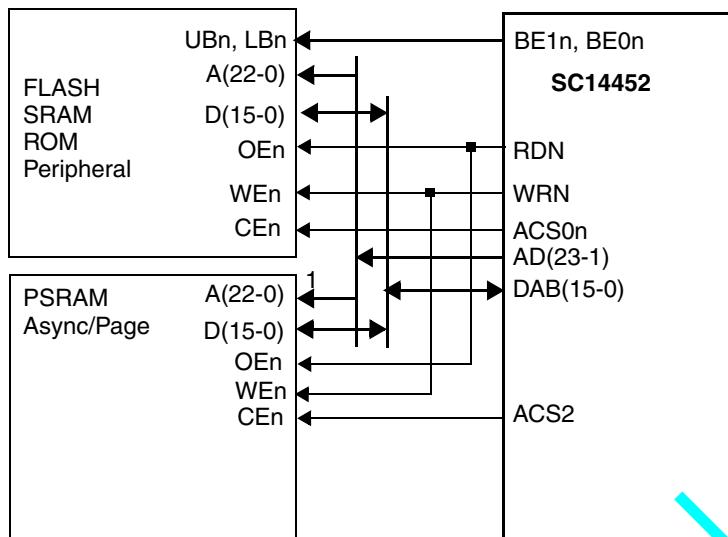

January 23, 2008

Figure 51 External Bus sharing Asynchronous memories and PSRAM with Page/Async mode

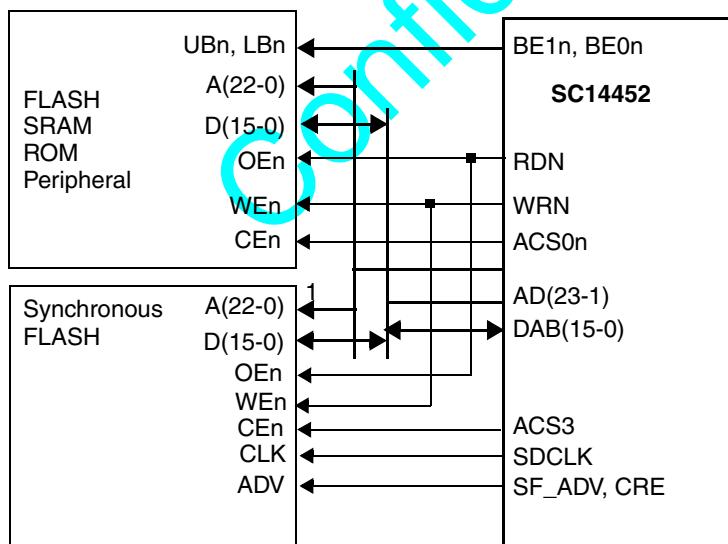

August 29, 2006

Figure 52 External Bus interface sharing Asynchronous memories and Synchronous FLASH with Burst / Page/asynchronous mode.

20.0 DMA Controller

The DMA Controller has four identical channels which can operate simultaneously. Its control registers are connected to APB to save current. One AHB master is connected to the AHB bus to be able to read and write to all memory-mapped registers and memories.

The AHB master is included to serve the demands of the DMA channels. The DMA channels request the use of the AHB master from an arbiter. The arbiter, based

on a priority schema, grants only one channel every time. Then waits the AHB master to make a transfer of maximum length 8 for the granted channel, and re-arbitrates. The performed actions of the AHB master is specified in the registers of the granted channel. With this way the DMA channel is only a specification of actions that the AHB master must execute.

Note 13: No DMA access should be performed at the non-shared RAM range, i.e. at the range of 0x0000.0000 up to 0x0000.FFFF

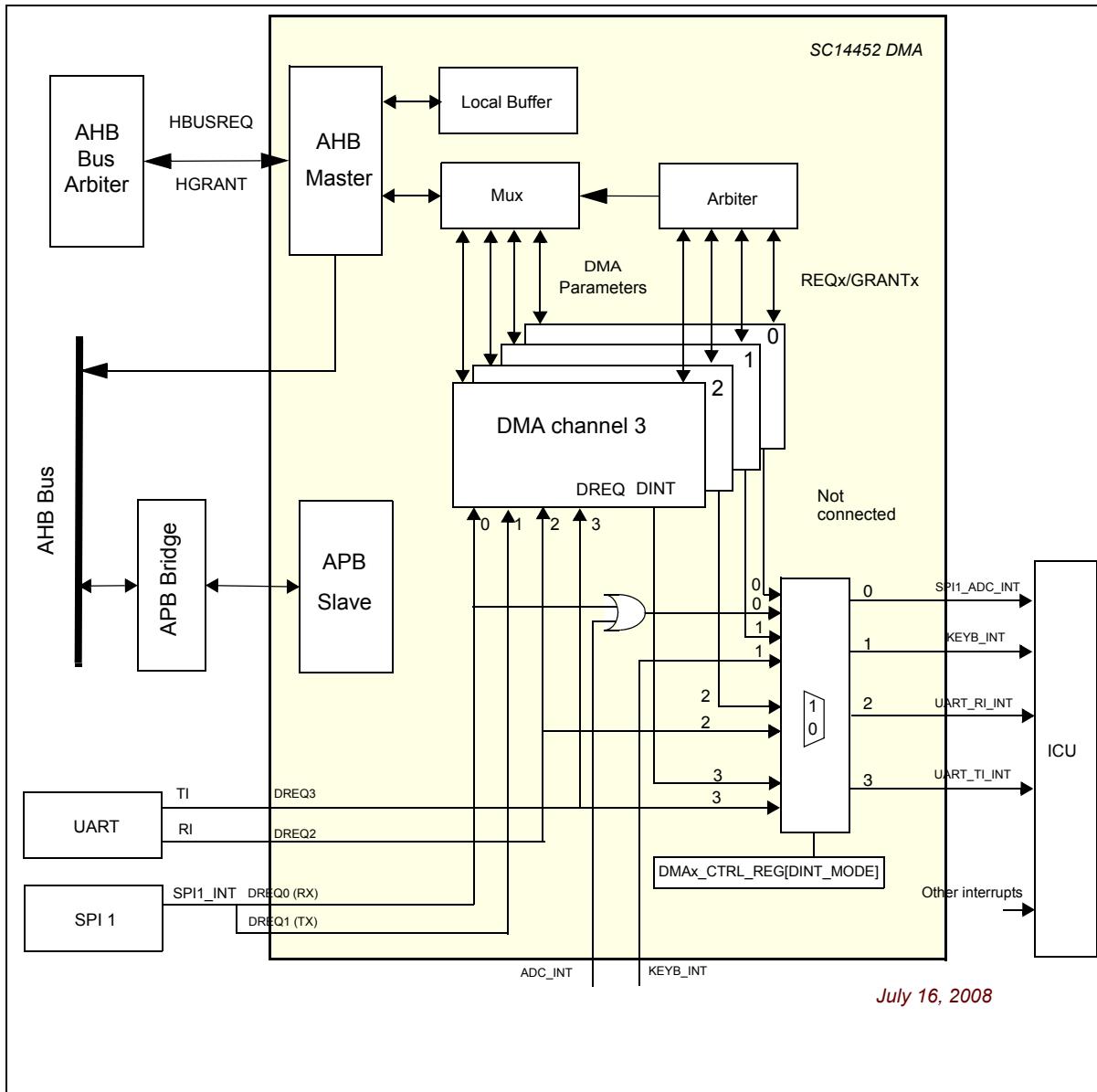


Figure 53 DMA controller block diagram

The four identical channels can describe fast data transfers between shared, non-shared internal or external memories or peripheral devices. All channels can copy up to 64K-1 bytes, half-words or words of data within the 32Mbyte address range.

The channels can describe the following data transfers:

- **Memory to memory.** A memory block is transferred from a source address to a destination address.
- **Memory to peripheral.** A memory block is transferred from a source address to a peripheral register pointed by the destination address.
- **Peripheral to memory.** A memory block is transferred from a peripheral register pointed by the source address to a destination address.
- **Memory initialization.** A region of memory that is clarified by the address of destination is written with zero "0" or one "1".

The following devices have DMA support for "**peripheral to memory**" or "**memory to peripheral**" transactions:

- DREQ0: SPI1 (Rx)
- DREQ1: SPI1 (Tx) or ADC_SYNC.
- DREQ2: UART Rx
- DREQ3: UART Tx

The AHB master can copy data from a predefined location A to another predefined location B. This is done in two stages: a) fetch data from A and store it in a storage buffer (Local Buffer), b) fetch data from the storage buffer and store it in B. Both A and B can be peripherals like UART and SPI or any type of memory. Because of the use of the storage buffer, data can be fetched and stored in bursts. The size of the storage buffer is 8 words x 32 bits, such that the maximum length of an incremental burst can be up to 8-beat (for 32 bits data words).

20.1 DMA CHANNEL OPERATION

A DMA channel is switched on with bit **DMA_ON**. This bit is automatically reset if the DMA transfer is finished. The DMA channels can either be triggered by software or by an hardware interrupt source. If **DREQ_MODE** is 0, then a DMA channel is immediately triggered. If **DREQ_MODE** is 1 the DMA channel can be triggered by an Hardware source.

When the DMA channel is triggered, it request the use of the AHB master. The arbiter grants the AHB master to the DMA channel when this is allowed by the priority of the channel. The AHB master request the AHB bus from the AHB Bus Arbiter

If AHB master is granted, data is transferred from the source address **{DMAx_A_STARTH, DMAx_A_STARTL}** to the destination address **{DMAx_B_STARTH, DMAx_B_STARTL}** for a length that is specified with the bits **MAX_BURST** and **DREQ_MODE** (see also "**Burst Mode**" on page 97). The data can be 8, 16 or 32 bits wide. The address

increment is realized with an internal 16 bits counter **DMA_IDX**, which is set to 0 if the DMA transfer starts and is compared with the **DMA_LEN** after each transfer. The register value is multiplied according to the **AINC** and **BINC** and **BW** values before it is added **DMA_B_STARTL/H_REG** and **DMA_B_STARTL/H_REG**. **AINC** or **BINC** must be 0 for register access.

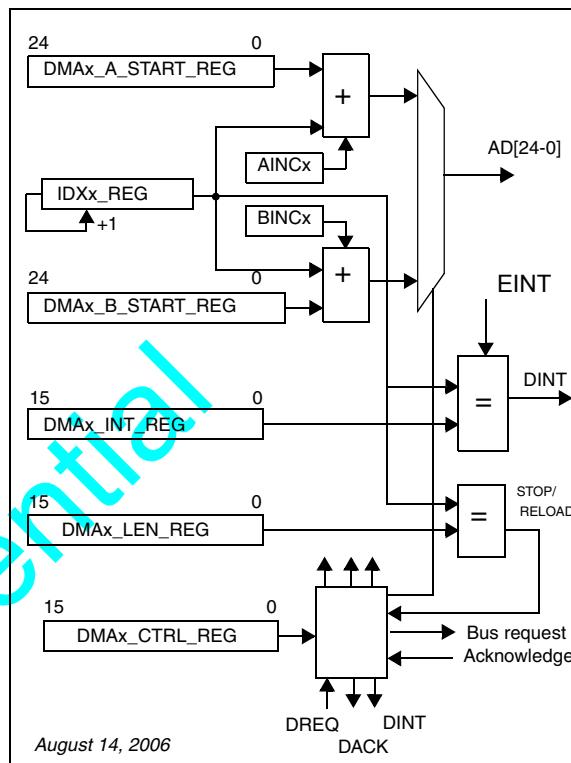


Figure 54 DMA channel diagram

Either the DMA channels or the interrupt sources can generate an interrupt to the ICU. This can be by bit **DINT_MODE**. (see also "Interrupt Control Unit (ICU)" on page 164)

If the DMA channel is started with **DREQ_MODE=1**, it always waits for the DMA start condition (DREQ=1 in Figure 53) before request the use of the AHB master. The DMA channel is not requesting the AHB master if the DREQ is low.

If the **DMA_LEN** is equal to the internal **DMA_IDX** the programmed DMA length is reached. In this case if the **CIRCULAR=0** or the **DREQ_MODE=0** the DMA channel stops and clears the **DMA_ON** bit. If bit **CIRCULAR=1** and the **DREQ_MODE=1** the DMA channel automatically reset the internal index registers and continues from its starting address without intervention of the processor. In this case the **DMA_ON** remain in active state.

Each DMA channel can generate an interrupt after a number of transfers determined by **DMAx_INT**. If **DMAx_INT** is zero, never an interrupt is generated.

20.2 DMA CHANNEL ARBITRATION

The priority level of a DMA channel can be set with bits **DMA_PRIO**. These bits determine which DMA channel will be activated in case more than one DMA channel requests DMA. Once the DMA channel is active, it can not be interrupted.

20.3 FREEZING DMA CHANNELS

Each channel of the DMA controller can be temporarily disabled by writing a 1 to bits FRZ_DMA_CH0,1,2,3 in the SET_FREEZE_REG.

To enable the channels again, a 1 to bits FRZ_DMA_CH0,1,2,3 in the RESET_FREEZE_REG must be written.

There is no hardware protection against miss programming the DMA registers.

DMA transfers will take place over the system bus, so they can be traced by a In Circuit Emulator.

The CR16C+ CLK divider register must be set such that the system clock is minimum 1.152 MHz.

20.4 BURST MODE

The AHB master supports 4-beat and 8-beat incremental bursts, which are the recommended lengths for an optimal AHB bus usage. However the controller can make single transfers when this is necessary, for example when the source or destination is a peripheral like UART or SPI. For DMA lengths different than 1, 4 or 8 the AHB master performs an unspecified length incremental access. In this case the length of a burst is always lower than 8 beat.

The AHB bus burst modes of the DMA can be controlled via the **MAX_BURST** bits of DMAx_CTRL_REG register. When **DREQ_MODE=1** the AHB master always performs single accesses independent from the value of the **MAX_BURST**.

20.5 MEMORY INITIALIZATION MODE

The DMA block can be used to initialize a memory area into all zeros "0" or into all ones "1" by using the **MEM_INIT** and **INIT_VAL** register bits of DMAx_CTRL_REG register.

20.6 DMA LENGTH REGISTERS

The DMAx_LEN_REG registers indicate the number of the data to be transferred. For example the value 5 means five words of 32-bit / 16bits / 8-bits width (according to the **BW** field).

20.7 DMA INTERRUPT REGISTERS

The DMAx_INT_REG registers specify the number of transfers until an interrupt is generated. For example a value "3" will cause an interrupt (if **DINT_MODE=1**) on the 3rd write access after the engagement of the DMA channel (**DMA_ON=1**). By setting those registers to zero "0" the appropriate DMA channel will not cause any interrupt.

21.0 CCU

The CRC Calculation Unit (CCU) supports single cycle 16 bits wide CRC calculations for three polynomials.

The **MODE** bits in the CCU_MODE_REG selects:

Table 50: CCU mode selection

MODE	Polynomial
00	CRC-16 calculation with polynomial $G(x) = 1+x^5+x^{12}+x^{16}$
01	CRC-32 calculation with polynomial: $G(x) = 1+x+x^2+x^4+x^5+x^7+x^8+x^{10}+x^{11}+x^{12}+x^{16}+x^{22}+x^{23}+x^{26}+x^{32}$
10	1's complement addition. (For IP Tx, TPC/UPD Rx/Tx frames)

The **CCU_IN_REG** is the CRC input register. If a WORD is written to this register, the CCU_CRC_REG will be updated according to selection of the MODE bit.

The **CCU_CRC_REG** contains the final calculated CRC or 1's complement sum. The reset value of this register is zero but the initial value can be set to any value by writing to the CCU_CRC_REG directly. In order to deal with little and big endian formats, the bits 15 to 8 and 7 to 0 of the input word can be swapped while they are written to the **CRC_IN_REG** by setting the bit **BIT_SWAP** to 1 in CCU_MODE_REG. If **BIT_SWAP** is 1, bits 15 to 8 are swapped to bit 8 to 15 and bits 7 to 0 are swapped to 0 to 7. The CCU_IN_REG is never swapped if read back. The CCU_CRC_REG is also swapped if written and read back if the **SWAP** bit is set to 1. Bits 31-16 can not be modified and return 0 if read. In addition bytes in a word can be swapped with **BYTE_SWAP**.

The CCU can be invoked from either CR16C+ or from the DMA controller working in indirect mode with a WORD block move to the CCU_IN_REG. The CCU_CRC_REG must be copied back to the buffer after the DMA transfer.

Note that the CCU_IN_REG is 16 bits so that an even number of bytes must be transferred.

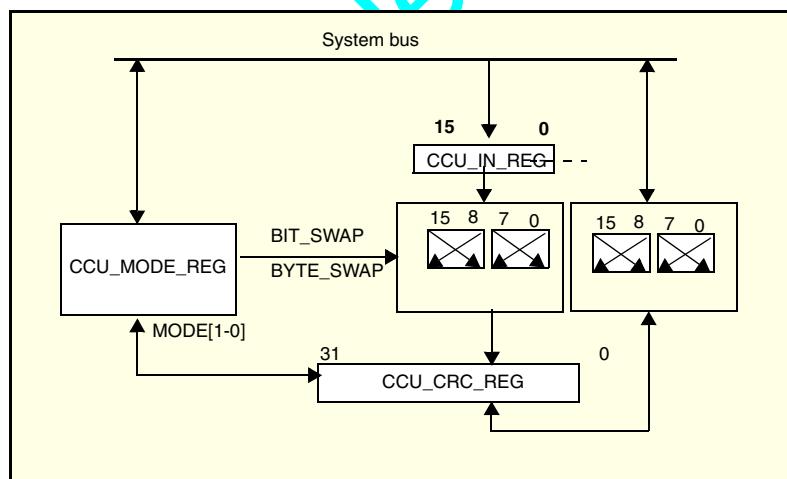


Figure 55 CRC Calculation Unit block diagram

22.0 I/O ports P0, P1, P2, P3

The SC14452 has general purpose I/O ports: P0 to P3 with peripheral assignment as explained in chapter "Programmable Pin Assignment" on page 15.

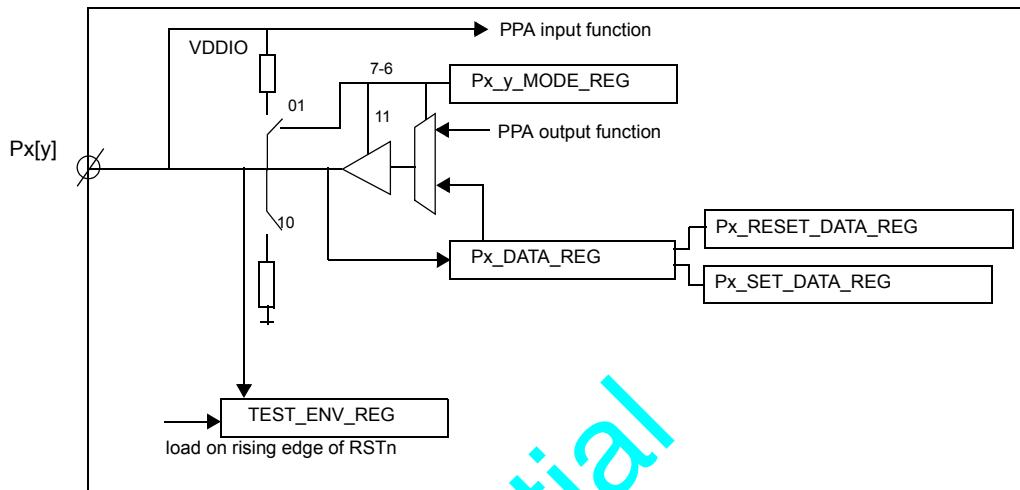


Figure 56 Port P0, P1, P2 with Programmable Peripheral Assignment

23.0 Keyboard Interface

A **keyboard** function can be made using Px as output and Py as input. Six port input pins INT[5-0] can be programmed to generate a KEYB_INT interrupt. Three pins INT[8-6] can be configured as general purpose interrupt input. PON and CHARGE pins can also be “ored” together with the P1 port to generate a keyboard interrupt. The PON and CHARGE pins can be programmed to bypass the debounce filter.

Features

- Up-to 6 keyboard and 3 general purpose maskable interrupt pins.
- Trigger on the positive or negative edges with or without debounce timer
- Separate PON, CHARGE inputs (rising edge)
- 3x edge/level interrupts shared with keyboard interrupt.
- Keyboard interrupt on key release.
- The debounce timer is programmable from N x 1ms N=0-63.
- Automatic keyboard interrupt on key pressed condition. The repeat time is programmable from N x 1ms N=1-63.
- Keyboard Status register with interrupt vector.

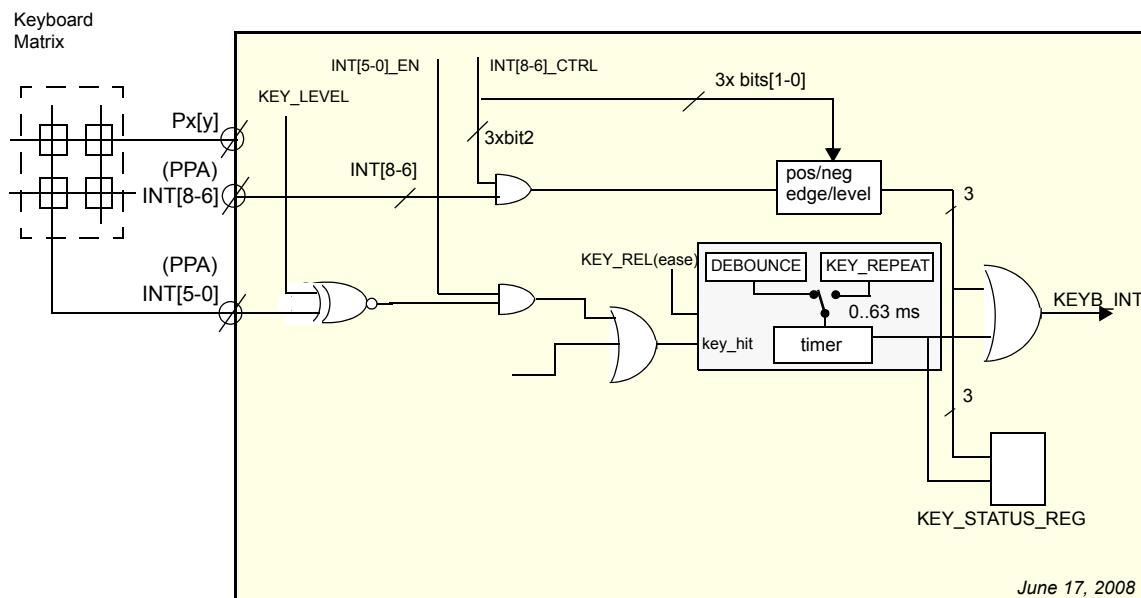


Figure 57 Interrupts block diagram

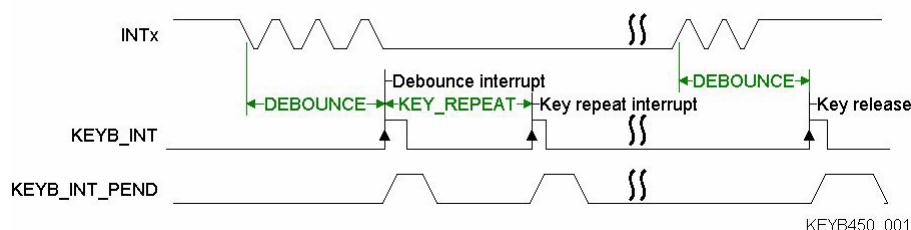


Figure 58 Basic Keyboard debounce and key repeat timing

Keyboard debounce timer and key repeat interrupt

(see Figure 58) A high to low level transition on one of the port inputs INT[5-0], while bits KEY_LEVEL =0 and INTx_EN =1, sets internal signal “key_hit” to 1. This

signal triggers the keyboard interface state machine as depicted in Figure 59. The debounce timer is loaded with value KEY_DEBOUNCE_REG[DEBOUNCE]. The timer counts down every 1ms. If the timer is 0 and the

"key_hit" signal is still '1', the timer is loaded with value KEY_DEBOUNCE_REG[KEY_REPEAT], generating a repeating sequence of interrupts every time the timer reaches 0.

If the key is released (key_hit = 0) and bit KEY_REL (key release) is set, a new debounce sequence is started and a KEYB_INT is generated after the debounce time.

The debounce timer can be disabled if DEBOUNCE is set to 0. Similarly, the key repeat function can be disabled if KEY_REPEAT is set to 0.

The low to high or high to low level change is programmable with bit KEY_BOARD_INT_REG[KEY_LEVEL], the key release function can be disabled with bit KEY_BOARD_INT_REG[KEY_REL] set to 0. One or more interrupts can be enabled by setting the corresponding KEY_BOARD_INT_REG[INTx_EN] to '1'.

The keyboard interrupt routine can distinguish which INTx has caused the interrupt by reading the P0_DATA_REG, P1_DATA_REG or P2_DATA_REG depending on the PPA configuration.

General purpose interrupts

INT[8-6] can be configured to generate an interrupt on a rising or falling edge or on a positive or negative level.

Refer to KEY_GP_INT_REG[INT8-6_CTRL] bits. The edge detection is done with a frequency of 1.152 MHz, so will introduce a latency of approximately 3us before the ICU is triggered.

Status registers

The status of the INT8, INT7, INT6 and keyboard timer can be found in KEY_STATUS_REG (0xFF49B6) (table 236, page 241). The status bits can only be cleared by writing a '1'. Bits not set to '1', shall be written with a '0'. This prevents unwanted interrupt clearing while monitoring this register via the JTAG interface.

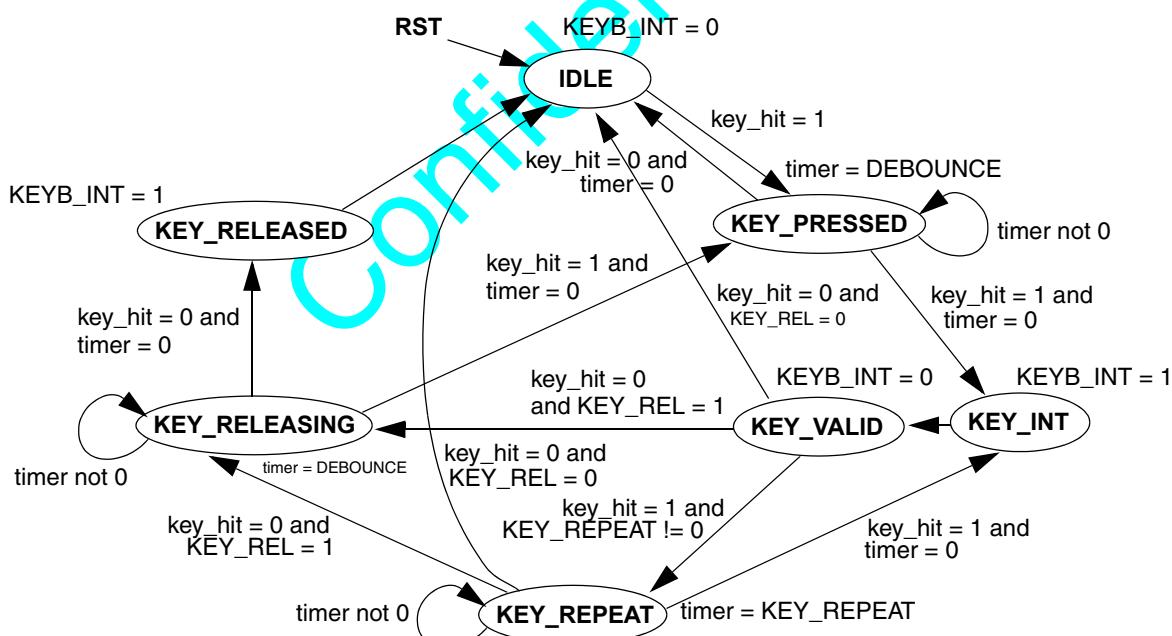
New interrupts on the same input will not be remembered until the interrupt is cleared.

Typical programming sequence to read and clear the interrupt vector bits:

```
key_vector = KEY_INT_STATUS_REG;
KEY_INT_STATUS_REG = key_vector;
```

Interface to interrupt control unit (ICU).

The KEYB_INT only generates an interrupt the ICU if the INT3_PRIORITY_REG[KEYB_PRIO] is set unequal to 0. If an interrupt is set, the interrupt must be cleared by writing RESET_INT_PENDING_REG[KEYB_INT_PEND]= '1'.



October 16, 2006

Figure 59 Keyboard interface state machine

24.0 UART

The SC14452 has one full duplex Universal Asynchronous Receiver Transmitter (UART) which interfaces through pins URX and UTX.

Features:

- Full/half duplex operation
- 1 start bit, 8 data bits (LSB first), 1 stop bit
- Baudrates 230.4k , 115.2 k, 57.6 k, 19200 or 9600 Baud with no parity. 10.125kBaud with even parity (Frequencies are based on PER10_DIV output frequency of 10.368 MHz.)
- Maskable receive and transmit interrupts
- DMA operation
- IrDA mode

The baudrates can be selected with **UART_CTRL_REG**. After reset the baudrate is 9600 baud.

Data loaded in the UART transmit register **UART_TX_REG** is transmitted on pin UTX. This register is loaded into the TX shift register only if the start bit, 8 data bits and stop bit are transmitted and the TI bit is 0 (no UART interrupt pending). After the shift register is reloaded the UART interrupt pending bit is set to 1. The RI/TI interrupt source can be read in **UART_CTRL_REG** bit 6 and 5. Data received on pin URX is copied from the RX shift register in **UART_RX_REG** after detection of the stop bit. After

transferring the data to the **UART_RX_REG** the UART interrupt pending bit is set to 1. After $(10 * \text{Rx/Tx Clock})$ the data received in the **UART_RX_REG** is overwritten by the next received data bit.

Note: **UART_RX_REG** and **UART_TX_REG** are two registers mapped on the same address.

Pull-up resistors are enabled as specified in the pin description.

Additionally 10.125 kbaud can be selected with even parity check on 8 data bits. In case the number of 1s in the received data including the parity bit is not even, the parity error **UART_ERROR_REG[PAR_STATUS]** is set. In transmit mode the parity bit makes the number of transmitted 1's even.

Transmitter and receiver operation are enabled with bits **UART_CTRL_REG[TEN]** and **REN**. If **TEN** is set to 0 during a transmission of the UART, the transfer is finished normally.

The UART RI and TI interrupts sources must be cleared by writing any value to **UART_CLEAR_RX_INT_REG**, resp **UART_CLEAR_TX_INT_REG**. During DMA operation these bits are cleared automatically.

The UART functional behaviour is not affected by the DMA.

For multiplexing UART pins to port pins, refer to **Py_xx_MODE_REGS**.

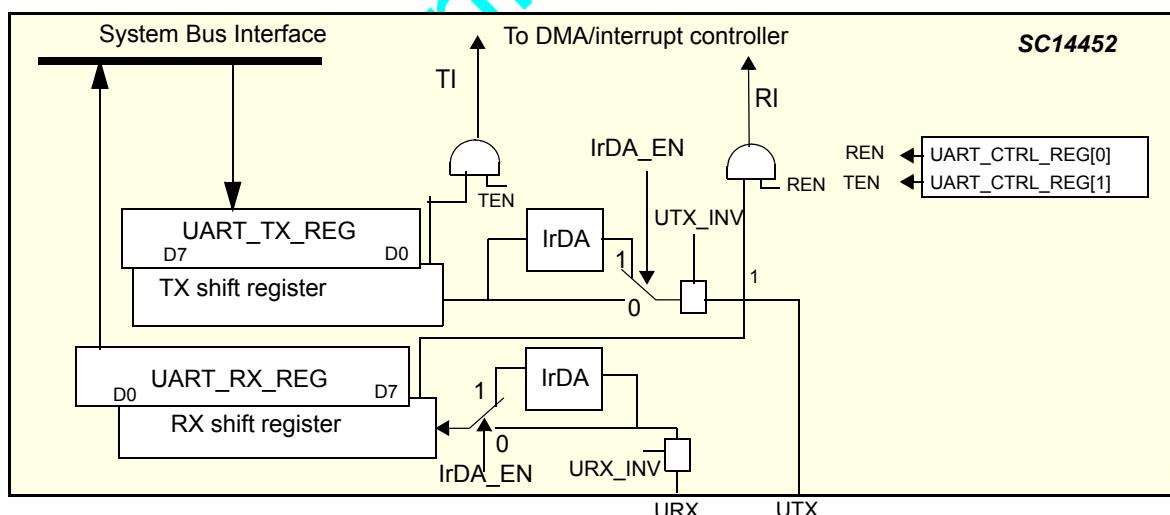


Figure 60 UART blockdiagram

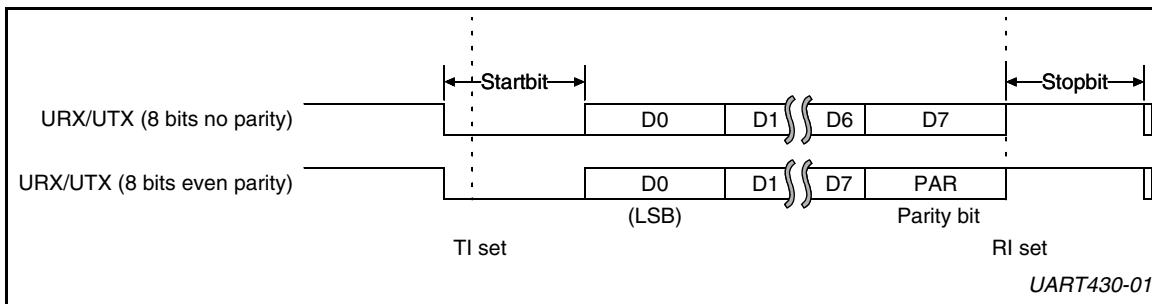


Figure 61 UART data transfer

24.1 IRDA MODE.

The UART (See also 24.0 on page 102) supports bidirectional IrDA data transfer with 1 start bit, 8 data bits and 1 stop bit (LSB first) and no parity. The supported bit rate are 9600, 19200, 57600 and 115200 baud with RZI modulation type.

The UART can be switched to IrDA mode with `UART_CTRL_REG` bit `IRDA_EN=1`. The timing signals of a IrDA Frame is shown in Figure 62.

Refer to Table 391 for accurate timing of bit time and pulse width.

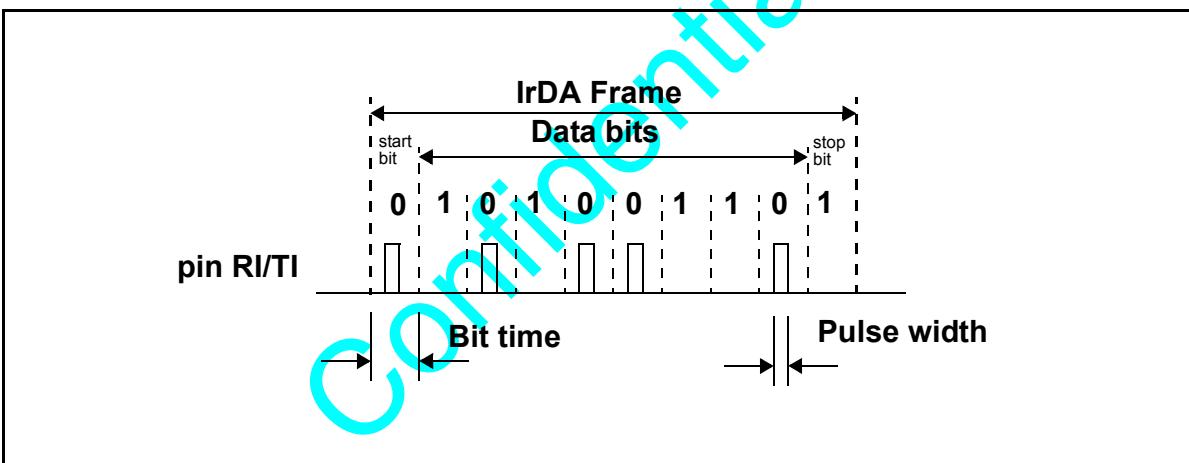


Figure 62 IrDA frame

25.0 ACCESS BUS

The SC14452 has two serial two wire ACCESS Bus interfaces (AB) for communication to a variety of peripheral devices.

Features:

- Master transmitter and master receiver mode
- 100kHz, 400 kHz and 1.152 MHz bus clocks.
(With PER10_DIV output = 10.368 MHz
100kHz: PER10_DIV output / 104,
400kHz: PER10_DIV output / 26
1.152 MHz: PER10_DIV output /9)
- 8 bits (master transmitter only) or 9 bits (8 + acknowledge) transfers

- Open drain or push pull outputs on clock and data
- Clock bit and byte stretching for flow control.
- Both blocks shared the same Interrupt

Interface signals

- SDA = Serial data in/out.
Selectable push-pull or open drain.
- SCL = Serial clock in/out
Selectable push-pull or open drain.
In open drain mode SCL is also input to support clock stretch

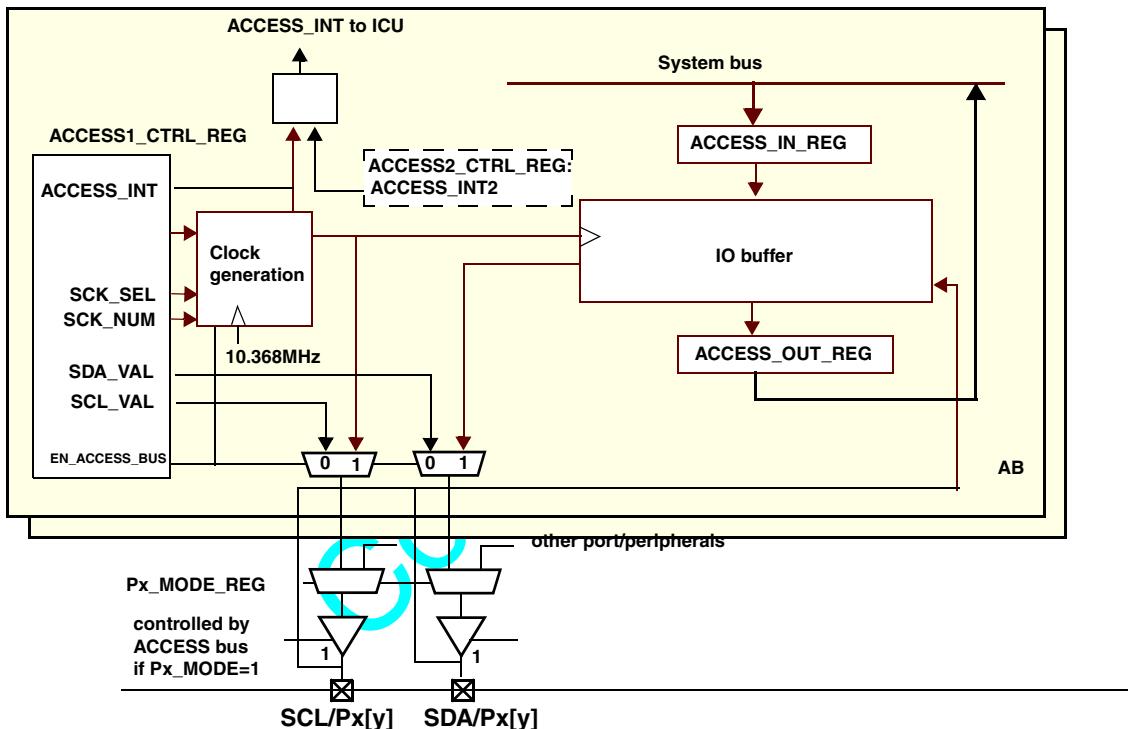


Figure 63 ACCESS bus 1 and 2 block diagram

Port configuration

To enable ACCESS bus operation, the port output multiplexer must be correctly set in Px_xx_MODE_REGS to switch SDA and SCL to the port pins.

If ACCESS bus mode is selected, the driving output state (Hi-Z, Push-pull or Open drain) is fully controlled by the ACCESS bus blocks depending on the state and SCK_NUM, SCL_OD bits. Note that bit SDA_OD and SCL_OD must be correctly set to push-pull or open drain.

Master transmitter mode

Prior to enabling the ACCESS bus interface (ACCESSx_CTRL_REG[EN_ACCESS_BUSx]= 0), the

Start condition must be programmed by software with a sequence of ACCESSx_CTRL_REG[SDA_VAL, SCL_VAL] bits: 11->01->00.

Next the ACCESS bus can be enabled with ACCESSx_CTRL_REG[EN_ACCESS_BUSx]= 1. Both SCLx and SDAx are set to 0. After a byte is written in the ACCESSx_IN_OUT_REG, 8 bits are shifted out on SDA (MSB first) and an interrupt is generated.

The interrupt is cleared by writing anything to ACCESSx_CLEAR_INT_REG. If a new byte is loaded the next transfer is started.

In 9 bits mode, (SCK_NUM=1), the received <ACKn> bit will be stored in the control register

`ACCESSx_CTRL_REG[ACKn]` and interrupt bit `ACCESSx_CTRL_REG[ACCESS_INT]` is set. If the new byte is loaded and the interrupt is cleared by writing anything to `CLEAR_ACCESS_INT_REG`, the next transfer is started.

Master receiver mode (9 bits mode only)

Like the master transmitter mode but with `<R/Wn>` bit D0 set to 1 the ACCESS bus is switched to receiver mode for the next byte(s). More bytes can be consecutively received. When in receive mode, the value of `ACCESSx_CTRL_REG[ACKn]` will be transmitted in the `<ACKn>` bit.

Once a complete byte has been received and the `<ACKn>` bit is transferred an interrupt bit is set in the `ACCESSx_CTRL_REG[ACCESS_INT]` (if enabled with `EN_ACCESS_INT`) and the SCK is halted until the interrupt bit is cleared by writing anything to `CLEAR_ACCESSx_INT_REG`. Note that bit `SDA_OD` must be set to 1 to configure SDA as open drain.

Stop condition

Once one or more bytes are transferred, the ACCESS bus can be disabled again with `ACCESSx_CTRL_REG[EN_ACCESS_BUSx]=0` and a Stop condition must be programmed by software with a sequence of `ACCESSx_CTRL_REG[SDA_VAL], SCL_VAL]` bits: 00->01->11.

Clock stretching

Clock stretching on `SCLx` for each bit clock cycle is supported. As soon as the `SCLx` is set to 1 by the master (transmit or receive mode), the `SCLx` level is read back. If the level is kept low or pulled low by a slave within 40 ns after the rising edge, the master waits forever until the `SCLx` line goes high again. As soon as `SCLx` has reached the value '1', clock stretching is ignored until the next rising edge of `SCLx`.

Note that `SCL` must be configured as open drain by (`SCL_OD` set to 1) in order to avoid conflicts on `SCL`. If a slave does not support clock stretching, the push-pull mode can be selected to save the pull-up resistor.

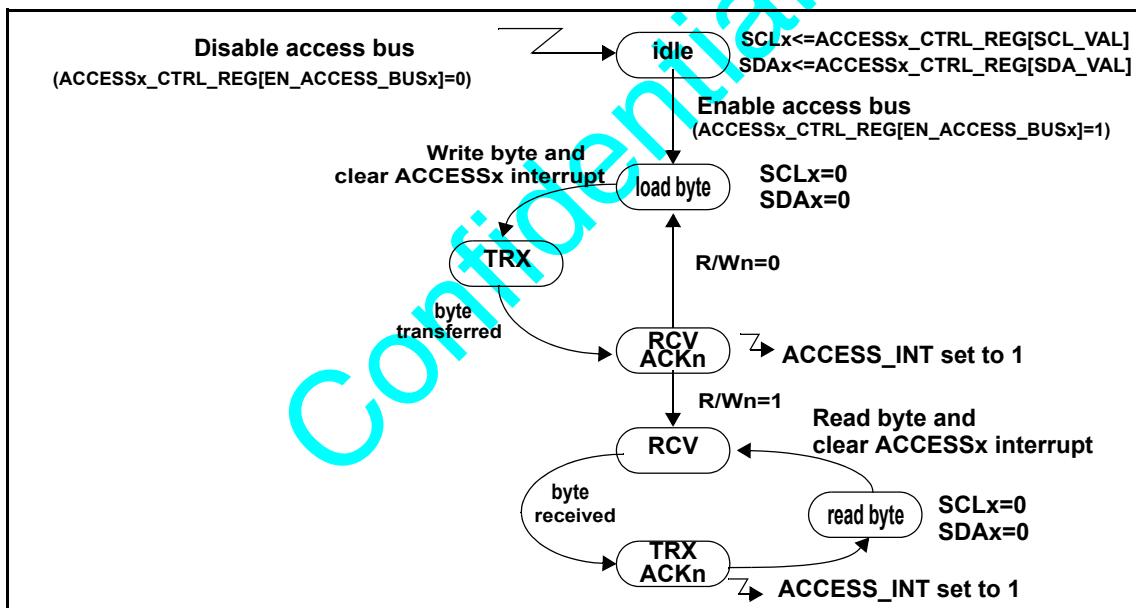


Figure 64 Access bus state machine

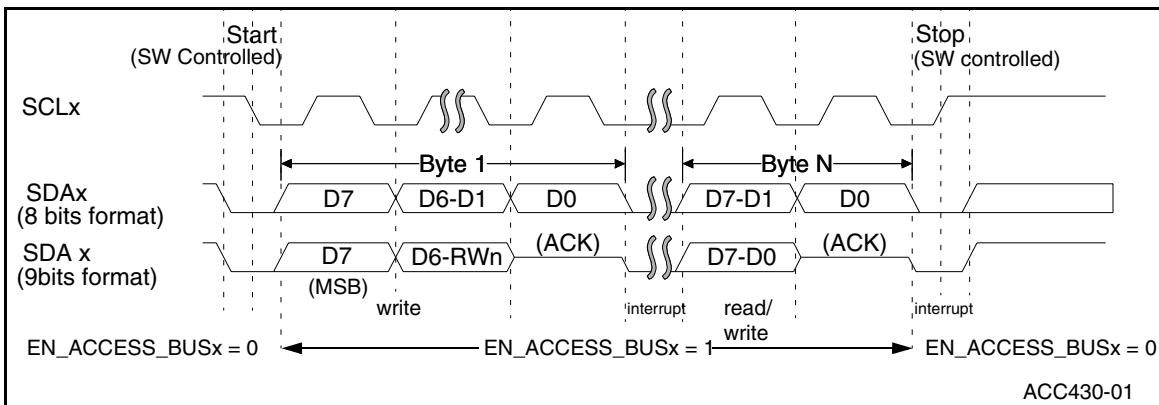


Figure 65 ACCESS bus master write/read access

26.0 SPI+ Controller

The SC14452 has two serial synchronous interfaces to support a subset of the Serial Peripheral Interface SPI™. The serial interface can transmit and receive 8, 16 or 32 bits in master/slave mode. The SPI+ controller has enhanced functionality with bidirectional FIFOs.

SPI™ is a trademark of Motorola, Inc.

Features

- Slave (SPI1 only) and Master mode
 - 8 bit, 16 bit or 32 bit operation
 - Clock speeds upto 20.736 MHz.
- Programmable output frequencies of SPI source

- clock divided by 2, 4, 8, 14
- SPI mode 0,1,2,3 support. (clock edge and phase)
- Programmable SPI_DOUT idle level
- Maskable Interrupt generation
- DMA support on SPI1, either on Rx and Tx data (8 and 16 bits mode only)
- Bus load reduction by unidirectional writes-only and reads-only modes.
- Built-in RX/TX FIFOs for continuous SPI bursts.
- Backwards compatible to SC1448x, SC1445x family members.

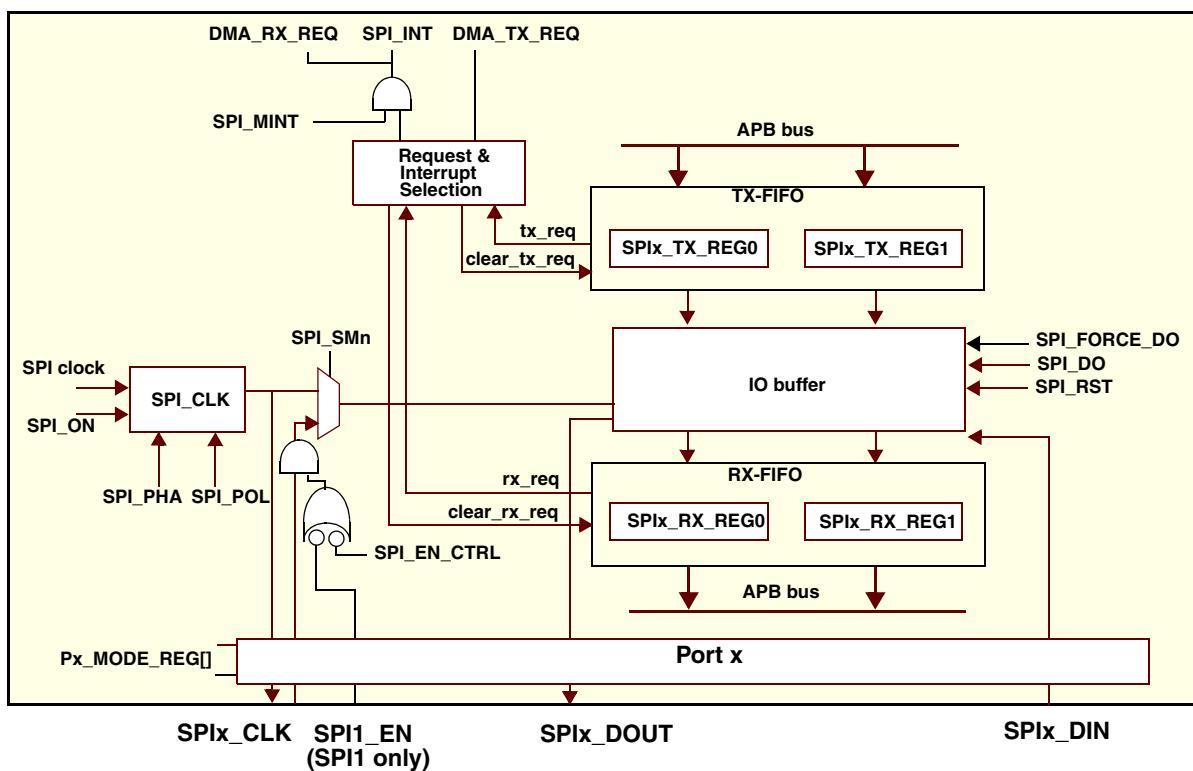


Figure 66 SPI blockdiagram

26.1 OPERATION WITHOUT FIFOs

This mode is the default mode, compatible SC144x, SC1448x device.

Master mode

To enable SPI™ operation, first the individual port signal must be enabled. Next the SPI must be configured in SPI1_CTRL_REG0, SPI2_CTRL_REG0 (0xFF4940,0xFF4950), for the desired mode. Finally the corresponding bit SPI_ON must be set to 1.

A SPI transfer cycle starts after writing to the SPIx_RX_TX_REG0. In case of 32 bits mode, the SPIx_RX_TX_REG1 must be written first. Writing to SPIx_RX_TX_REG0 also sets the SPI_TXH. As soon

as the holding register is copied to the IO buffer, the SPI_TXH is reset and a serial transfer cycle of 8/16/32 clock-cycles is started which causes 8/16/32 bits to be transmitted on SPIx_DOUT. Simultaneously, data is received on SPIx_DIN and shifted into the IO buffer. The transfer cycle finishes after the 8th/16th/32nd clock cycle and SPI_INT_BIT bit is set in the SPIx_CTRL_REG and SPI_INT_PEND bit in (RESET_INT_PENDING_REG is set. The received bits in the IO buffer are copied to the SPIx_RX_TX_REG0 (and SPIx_RX_TX_REG1 in case of 32 bits mode) were they can be read by the CR16C+.

Interrupts to the ICU can be disabled using the

SPI_MINT bit. To clear the SPI interrupt source, any value to SPIx_CLEAR_INT_REG must be written. Note however that SPI_INT will be set as long as the RX-FIFO contains unread data.

If data is read from SPIx_RX_TX_REG0/SPIx_RX_TX_REG1 before a SPI_INT is valid, the whole SPI interface might turn into an undefined state. To recover from this state, SPI_ON must be set to "0" or SPI_RST must be set to "1".

Slave mode

The slave mode is selected with **SPI_SMn** set to 1 and the Py_xx_MODE_REG must also select SPIx_CLK as input. The functionality of the IO buffer in slave and master mode is identical. The SPI module clocks data in on SPIx_DIN and out on SPIx_DOUT on every active edge of SPIx_CLK. As shown in Figure 67 to Figure 70. SPI1 has an active low clock enable SPI1_EN which can be enabled with bit SPI_EN_CTRL=1.

In slave mode the internal SPI clock must be more than four (4) times the SPIx_CLK.

In slave mode the SPI_EN serves as a clock enable if enabled with bit SPI_EN_CTRL. Data is shifted as long as SPI_EN=0 and a SPI_CLK occurs. To resynchronize with the sending master (if necessary), SPI_ON must be set to 0 or SPI_RST must be set to 1. This will reset the I/O buffers and also the I/O registers.

SPI_POL and SPI_PHA

The phase and polarity of the serial clock can be changed with bits SPI_POL and SPI_PHA in the SPIx_CTRL_REG.

SPI_DOUT idle levels

The idle level of signal SPI_DOUT depends on the master or slave mode and polarity and phase mode of the clock.

In master mode pin SPIx_DOUT gets the value of bit SPIx_CTRL_REG0[SPI_DO], if the SPI is idle in all modes. Also if slave in SPI modes 0 and 2, SPIx_CTRL_REG0[SPI_DO] bit value is the initial and final idle level.

In SPI modes 1 and 3 however there is no clock edge after the sampled LSB and pin SPIx_DOUT gets the LSB value of the IO buffer. If required, the SPIx_DOUT can be forced to the SPIx_CTRL_REG0[SPI_DO] bit level by resetting the SPI to the idle state by shortly setting bit **SPI_RST** to 1. (Optionally **SPI_FORCE_DO** can be set, but this does not reset the IO buffer). The following Figure 67 to Figure 70 show the timing of the SPITM interface.

DMA operation

Only SPI1 supports DMA, but only in 8 or 16 bit mode. In backward compatible mode both DMA0 and DMA1 channel must be used for either DMA supported transmit or transmit/receive. By triggering two DMA channels simultaneously upon a SPI1_INT interrupt, the SPI1 Rx data is handled with DMA Channel 0 and SPI

1 Tx with DMA channel 1.

To switch SPI1_INT to DMA0_REQ and DMA1_REQ, DMA0_CTRL_REG[DREQ_MODE] must be set to 1.

Example DMA initialization procedure for transmit/receive and both FIFOs enabled or not enabled:

- Make sure No SPI_INT interrupts are pending:
RESET_INT_PENDING_REG[SPI1_ADC_INT_PENDING] = 1
- DMA1_A_START_REG = TX_ARRAY+2
DMA1_B_START_REG = SPI1_RX_TX_REG
DMA1_LEN_REG = DMA_LENGTH-2
DMA1_INT_REG = DMA_LENGTH-2
- DMA0_A_START_REG = SPI1_RX_TX_REG
DMA0_B_START_REG = RX_ARRAY
DMA0_LEN_REG = DMA_LENGTH-1
DMA0_INT_REG = DMA_LENGTH-1
- Set DMA1_CTRL_REG:
DMA_PRIO = 2
DREQ_MODE = 1
AINC = 1
DINT_MODE = 1 (to optionally generate DMA1_INT after DMA burst)
BW = 16
DMA_ON = 1
- Set DMA0_CTRL_REG (after DMA1):
DMA_PRIO = 3 (RX must have higher priority than TX)
DREQ_MODE = 1
BINC = 1
DINT_MODE = 1 (to optionally generate DMA0_INT after DMA burst)
BW = 16
DMA_ON = 1
- To start DMA operation, refer to Table 51 to check whether an initial write is needed from the CR16C+:
SPI1_RX_TX_REG = TX_ARRAY[0]
- The next SPI1_INT triggers both DMA channels again and data is copied automatically.

For a transmit burst without receive, DMA0_CTRL_REG[AINC] must be kept to 0 (no increment) and the DMA0_A_START_REG, DMA0_B_START_REG must point to a dummy RX_ARRAY.

Terminating bursts

After the programmed number of bytes a DMA interrupt will be set. If after a DMA Tx interrupt the SPI need to be turned off, the SPI_BUSY must be polled to wait until the TX-FIFO is empty. After this, the SPI_EN pin may be deactivated if applicable.

Note that after the DMA Rx interrupt all bytes are read but the RX-FIFO may contain newly received data. Especially in reads only mode and in cases that the Tx DMA size is longer than the Rx DMA size this is most likely to happen. A SPI_RST (or SPI_ON) must be given in between DMA burst to clear all FIFO's.

DMA priority

If SPIx_CTRL_REG1[SPI_PRIORITY] is set to '0' then the requests to the DMAs will be reset at every acknowledge allowing other bus masters to use the AHB bus. If SPI_PRIORITY is '1', the requests will be kept high as long as RX/TX FIFO requests data. This setting is useful when high throughput is required. Disadvantage is that other bus masters could be holded longer.

26.2 FIFO OPERATION

In order to increase the overall throughput, the RX and/or TX FIFOs can be reconfigured as FIFO's. The transmitted bursts will become continuous as long as the FIFOs contain data. If a FIFO is enabled, the registers bits for 32-bit mode are used as FIFO. Depending on SPI_WORD size, the FIFO sizes are 4x 8-bit words, 2x 16-bit words or 1x 32-bit word. In addition the IO-buffer can hold one additional 8, 16 or 32 bit word. To enable the FIFO modes, SPIx_CTRL_REG1 [SPI_FIFO_MODE] must be set as shown in Table 51.

Table 51: DMA Request and interrupt selection (master and slave mode)

SPI_FIFO_MODE	FIFO usage	DMA_tx_req	DMA_rx_req/ SPI_INT*MINT=1	Write to SPIx_RX_TX_REG0 to start (No DMA)	Write to SPIx_RX_TX_REG0 to start (DMA)
0	Rx/Tx	TX-FIFO not full	RX-FIFO not empty	Yes	No
0	Rx (Reads only)	0	RX-FIFO not empty	No	No
1	Tx (writes only)	TX-FIFO not full	RX_REG not empty	Yes	No
1	No FIFO	TX_REG empty	RX_REG not empty	Yes	Yes

Writes only mode

In "writes only" mode (SPI_FIFO_MODE = "10") only the TX-FIFO is used. Received data will be copied to the SPIx_RX_TX_REGx, but if a new SPI transfer is finished before the old data is read from the memory, this register will be overwritten.

SPI_INT acts as a tx_request signal, indicating that there is still place in the FIFO. It will be '0' when the FIFO is full or else '1' when it's not full. This is also indicated in the SPIx_CTRL_REG[SPI_TXH], which is '1' if the TX-FIFO is full. Writing to the FIFO while this bit is still 1, will result in transmission of undefined data. The whole SPI interface might turn into an undefined state. To recover from this state, SPI_ON must be set to '0' or SPI_RST must be set to '1'.

If all data has been transferred, SPIx_CTRL_REG1 [SPI_BUSY] will become '0'.

For DMA operation only DMA1 must be configured. Starting transfers by manually writing to the SPIx_RX_REGx shall not be done because DMA_tx_req is already '1' when this mode is activated.

Reads only mode

In "reads only" mode (SPI_FIFO_MODE = "01") only the RX-FIFO is used. Transfers will start immediately when the SPI is turned on in this mode. In transmit direction the SPIx_DOUT pin will transmit the IO buffer contents being the actual value of the SPIx_RX_REGx (all 0's after reset). This means that no dummy writes are needed for read only transfers.

In **slave mode** transfers only take place if the external master initiates them, but in master mode this means that transfers will continue until the RX-FIFO is full. If this happens SPIx_CTRL_REG1[SPI_BUSY] will become '0'. If exactly N words need to be read from

SPI device, first read ($N - \text{fifosize} + 1$) words. Then wait until the SPI_BUSY becomes '0', set SPI_FIFO_MODE to "00" and finally read the remaining ($\text{fifosize} + 1$) words. Here *fifosize* is 4/2/1 words for 8/16/32 bits mode respectively.

If this is not done, more data will be read from the SPI device until the FIFO is completely filled, or the SPI is turned off.

For DMA operation only DMA0 must be configured. Manual transfers are not needed, as the SPI will start transferring immediately when turning on this mode.

Bidirectional transfers with FIFO.

If SPI_FIFO_MODE is "00", both registers are used as a FIFO. SPI_TXH indicates that TX-FIFO is full, SPI_INT indicates that there is data in the RX-FIFO.

DMA operation is recommended using both DMA0 and DMA1. No manual transfers are required because the requests will trigger the DMA automatically.

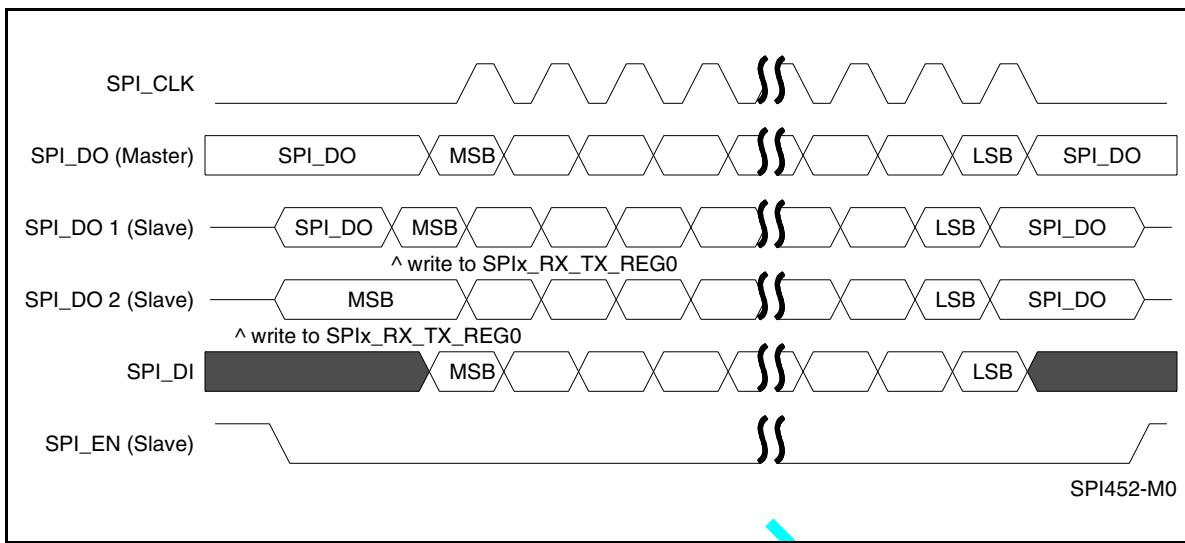


Figure 67 SPI Master/slave, mode 0: SPI_POL=0 and SPI_PHA=0

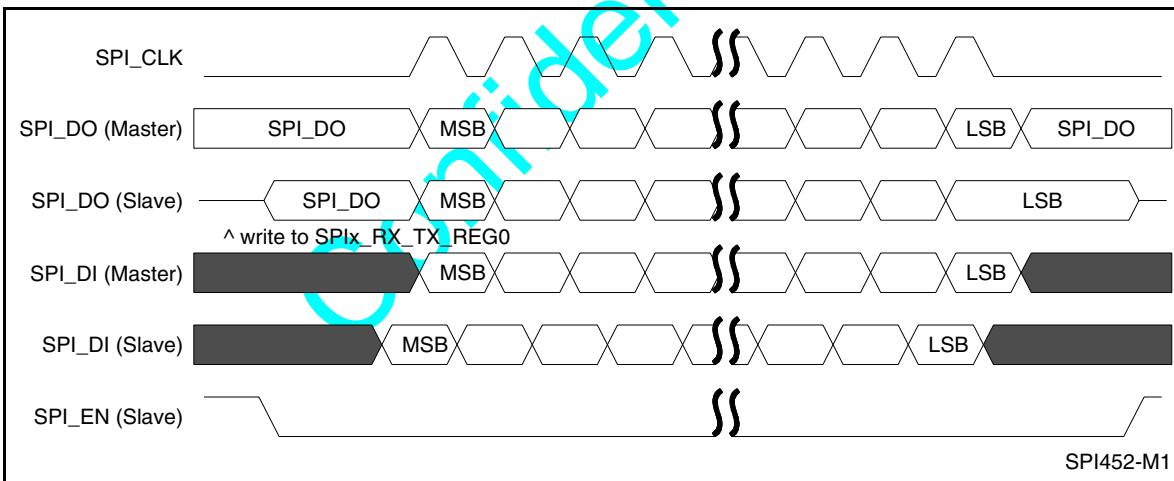


Figure 68 SPI Master/Slave, mode 1: SPI_POL=0 and SPI_PHA=1

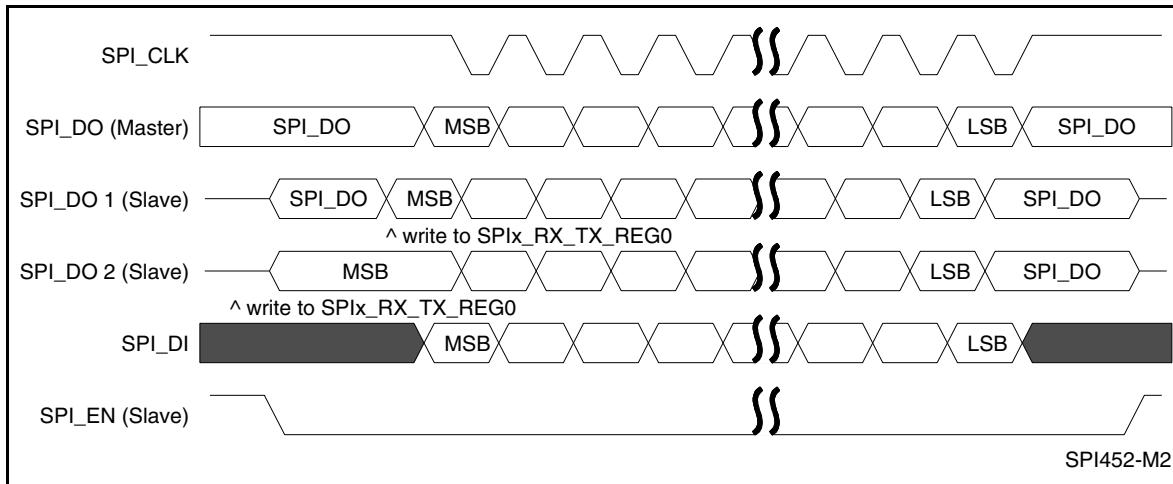


Figure 69 SPI Master/Slave, mode 2: SPI_POL=1 and SPI_PHA=0

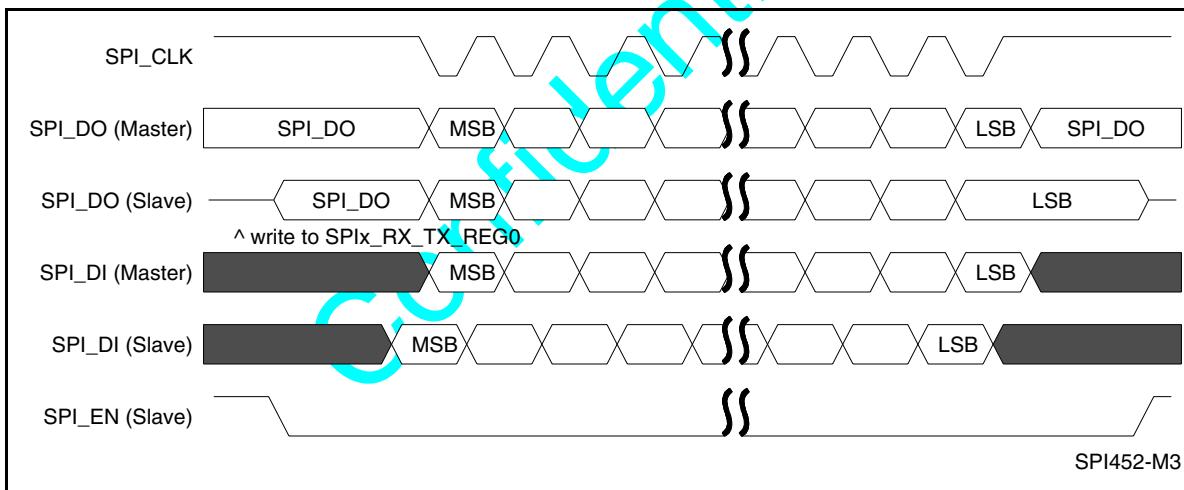


Figure 70 SPI Master/slave, mode 3: SPI_POL=1 and SPI_PHA=1

27.0 ADC

The SC14452 has a 10-bit successive approximation Analog Digital Converter. This ADC is optimized for portable applications for battery and battery temperature applications and fixed part applications to monitor line voltages and measure ring frequencies.

For line interface application protection circuits can be enabled on ADC0, ADC1, ADC2, ADC3, CIDOUT

The temperature sensor is described in "Temperature Sensor" on page 115.

Features

- 10 bits resolution
- Conversion time 55 μ s
- 1.6V LDO_XTAL reference range to suppress the effect of noise on VDDRF.
- External Inputs: ADC0, ADC1, ADC2, ADC3, CIDOUT, SUPPLY, VBUS monitoring.
- Switchable protection circuits for line application. (See Table 347 on page 286 for characteristics)
- Internal input: temperature sensor.
- Internal input: Codec headset detection
- Manual and automatic mode for real time 8/16 kHz signal processing on Gen2DSPs.

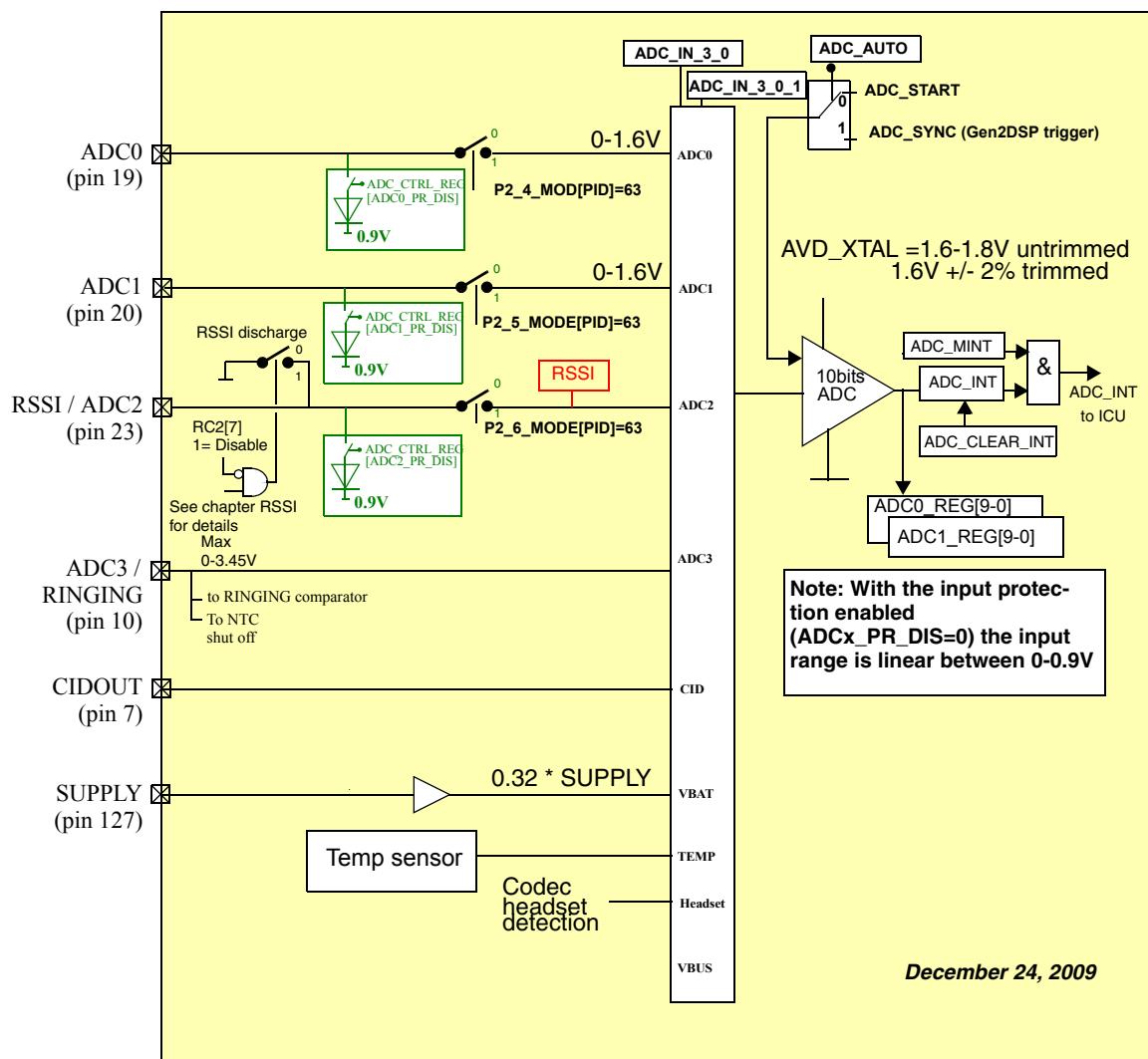


Figure 71 ADC Block Diagram

Manual AD conversion

The ADC_CTRL_REG[ADC_IN_3_0] determines which input is selected. An AD conversion is started by setting ADC_CTRL_REG[ADC_START] equal to 1. After 10 µs the input is sampled for 7µs. During this time an internal capacitive load of 2pF is charged or discharged from AVD/2 to the value of the input pin. The source impedance determines this time constant (See Table 359). After a total conversion time of 50 µs bit ADC_START is automatically cleared to 0 and bit ADC_INT is set. This must be cleared by SW by writing to register AD_CLEAR_INT. An interrupt to the ICU is only generated if ADC_MINT bit is also set to 1.

Automatic AD conversion

Automatic AD conversion can be used for:

- 8 kHz sampling of in band signals like ringing signal via CID opamp and signal processing by Gen2DSP.
- Relief of the CR16C+ from repetitive starting the AD converter and handling interrupts.

Automatic conversion is started by setting ADC_CTRL_REG[ADC_AUTO] to 1, while the automatic conversion is started on the rising edge of signal ADC_SYNC which can be set in DSP_MAIN_SYNC1_REG[ADC_SYNC] (See Figure 72). This signal can be 8/16 kHz.

Input selection

In automatic conversion mode, two separate ADC input can be selected for conversion, e.g ADC0 and Caller-id input by settings ADC_CTRL_REG[ADC_IN_3_0] resp ADC_CTRL1_REG[ADC_IN_3_0_1]. Both ADC_IN_3_0 and ADC_IN_3_0_1 may have the same value.

The output of the ADC is stored always ADC0_REG if ADC_CTRL_REG[ADC_ALT] = 0. If ADC_ALT = 1, samples selected by ADC_IN_3_0 are stored in ADC0_REG and samples selected by ADC_IN_3_0_1 are stored in ADC1_REG. Table 52 shows the maximum sample rate of ADC_SYNC if ADC_ALT =1.

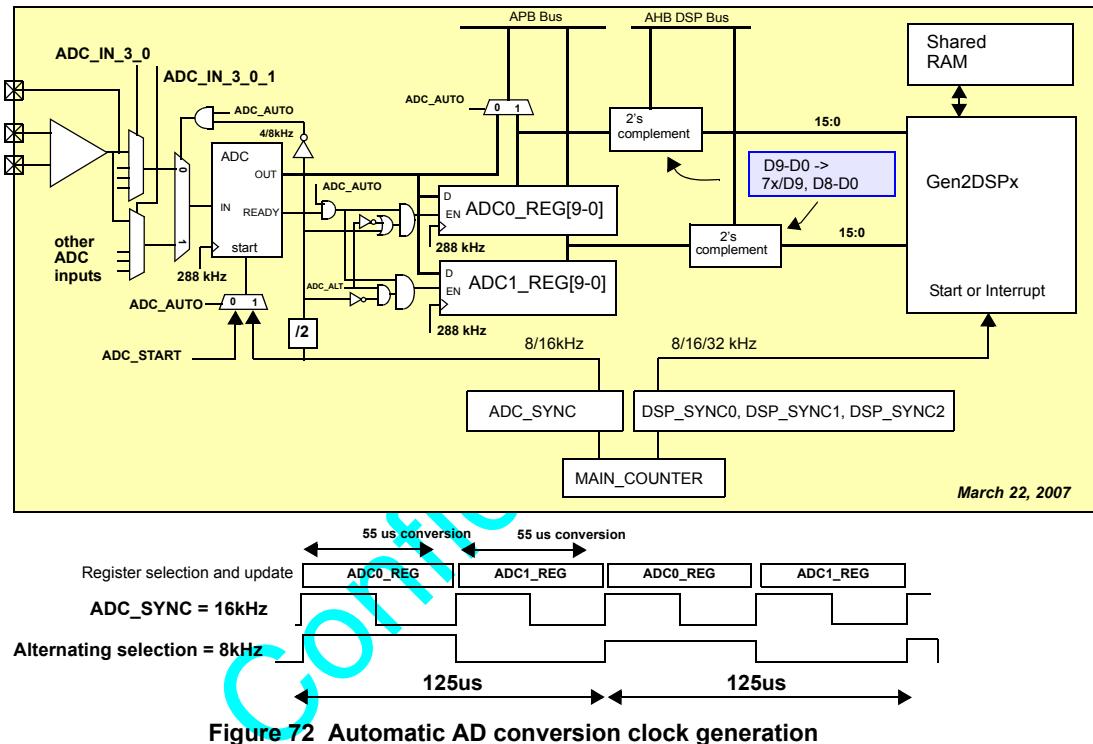
Output format

The converted values stored in ADC0_REG and ADC1_REG in 10 bits linear format. The Gen2DSP reads these value as 16 bits 2's complement in DSP_ADC0S_REG and DSP_ADC1S_REG for signal processing. In 2's complement notation, bits D9-D0 are converted to 7x(not)D9, **D8-D0** The CR16C+ can read both linear and 2's complement registers. The ADC0_REG and ADC1_REG can not be read by the Gen2DSP.

Table 52: ADC0_REG and ADC1_REG sample rate

ADC_AUTO	ADC_ALT	ADC0_REG	ADC1_REG
0	0	Manual	-
0	1	Manual	-
1	0	ADC_SYNC	
1	1	ADC_SYNC/2	ADC_SYNC/2

In automatic mode, no ADC_INT is generated. If switching from automatic to manual mode, the current conversion is correctly finished. However a new conversion may only be started after the ADC_START has become 0, otherwise the ADC0_REG may have an unpredictable result for one conversion. Switching from manual to automatic mode may be done at any time without data loss.



28.0 Temperature Sensor

The SC14452 contains an on-chip temperature sensor for measuring temperatures between 5 degrees and 55 degrees Celsius. The temperature sensor has a negative temperature coefficient α which must be determined during equipment production by measuring the ADCout value at room temperature and solving the

equation $ADCout = \alpha*T + 2960$. (T in Kelvin, ADCout is decimal value of ADC) Refer to Temperature sensor (table 357, page 293) for the minimal and maximal ranges of α .

Figure 73 shows transfer characteristic of the temperature sensor.

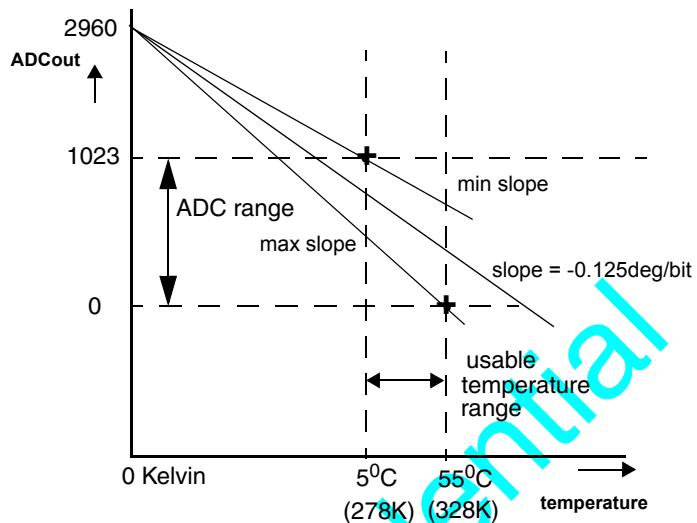


Figure 73 Temperature sensor

29.0 Timers

29.1 TIMER0

Timer 0 is used to generate Pulse Width Modulated signals PWM0 and PWM1 and to generate an interrupt to the CR16C+.

Features

- Programmable output Frequency = $(1.152 \text{ MHz or } 115.2 \text{ kHz}) / (M+1)+(N+1)$
($N=0..2^{**}16-1$, $M=0..2^{**}16-1$)
- Programmable duty cycle: $(N/M+1) * 100\%$
- Separate Programmable interrupt timer
 $(1.152\text{MHz or } 115.2\text{kHz}) / (ON+1)$

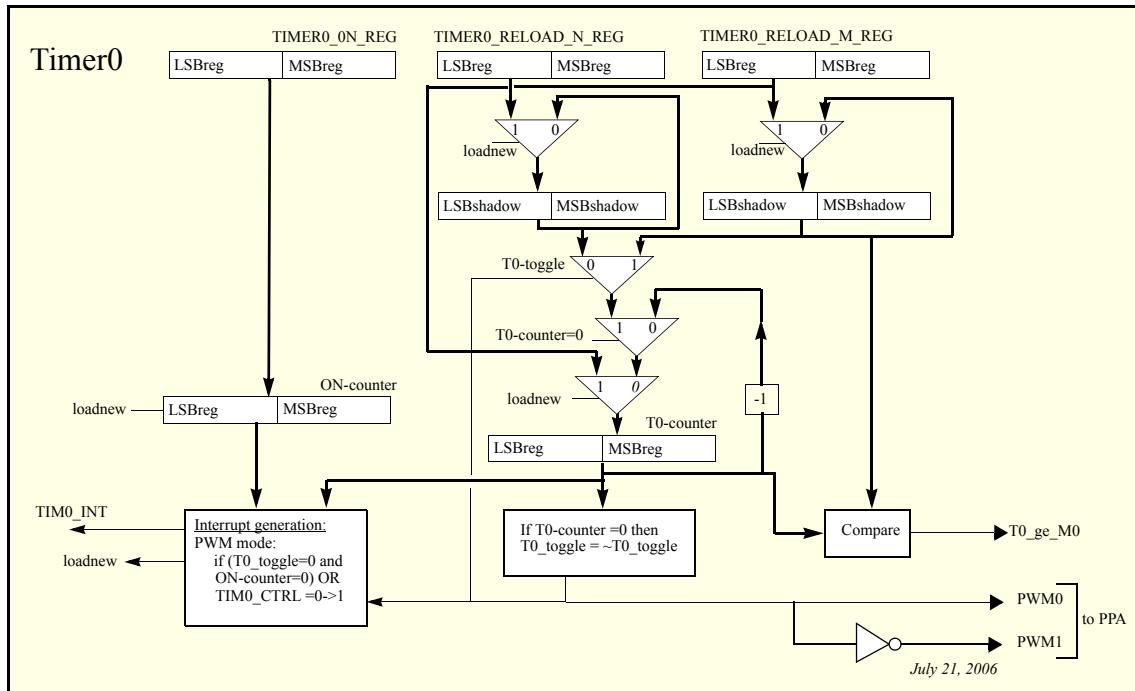


Figure 74 Timer0 block diagram

Figure 74 shows the block schematic of Timer0. The 16 bits timer consists of two counters: T0-counter and ON-counter, and three registers: TIMERO_RELOAD_M_REG, TIMERO_RELOAD_N_REG and TIMERO_ON_REG. On reset the counter and register values are 0000. Timer0 will generate a Pulse Width Modulated signal PWM0. The frequency and duty cycle of PWM0 are determined by the contents of the TIMERO_RELOAD_N_REG and the TIMERO_RELOAD_M_REG.

The timer can run at two different clocks: 115.2 kHz and 1.152 MHz. The clock speed can be selected by bits 0-10 of the CLK_GPIO3_REG2. The ON-counter only runs at 115.2 kHz.

Timer0 operates in PWM mode. Signals PWM0 and PWM1 can be switched to port pins.

Timer0 PWM mode

If TIM0_CTRL bit (bit 0) in the TIMER_CTRL_REG is

set, Timer0 will start running. TIM0_interrupt will be generated and the T0-counter will load its start value from the TIMERO_RELOAD_M_REG, and will decrement on each clock. Also the ON-counter loads its value from the TIMERO_ON_REG and decrements with a fixed clock of 115.2 kHz. When the T0-counter reaches zero the internal signal T0-toggle will be toggled and now selects the TIMERO_RELOAD_N_REG of which its value will be loaded in the T0-counter. Each time the T0-counter reaches zero it will alternately be reloaded with the values of the M0- and N0-shadow registers. If the ON-counter reaches zero it will remain zero until the T0-counter also reaches zero when it was decrementing the value loaded from the TIMERO_RELOAD_N_REG (PWM0 is low). The counter will then generate an interrupt (TIM0_interrupt). The ON-counter will be reloaded with the value of the ON-register. The T0-counter as well as the M0-shadow register will be loaded with the value of the TIMERO_RELOAD_M_REG. At the same time the N0-shadow register will be loaded by the

TIMER0_RELOAD_N_REG.

Both counters will be decremented on the next clock again and the sequence will be repeated.

During the time that the ON-counter is not zero, new values for the ON-register, M0-register and N0-register can be written, but they are not used by the T0-counter yet. The newly written values in the

TIMER0_RELOAD_M_REG and TIMER0_RELOAD_N_REG are only stored into the shadow registers when the ON-counter and the T0-counter have both reached zero and the T0-counter was decrementing the value loaded from the TIMER0_RELOAD_N_REG (see Figure 75).

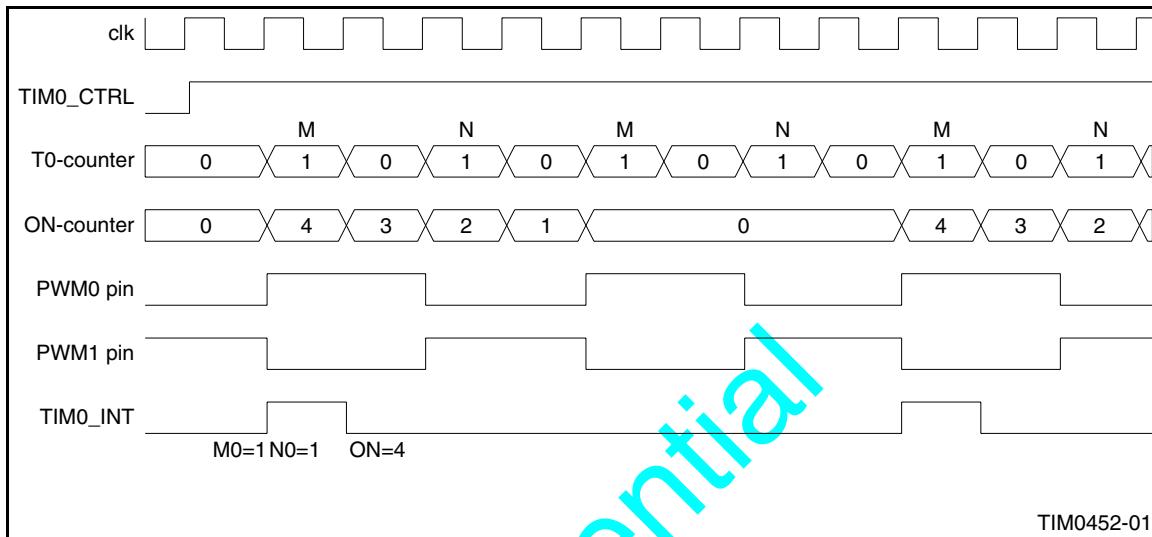


Figure 75 Timer 0 Pulse PWM mode

At start-up both counters and the PWM0 signal are low so also at start-up an interrupt is generated. If Timer0 is disabled all flip flops and outputs are in reset state except for the ON-register, TIMER0_RELOAD_N_REG and the TIMER0_RELOAD_M_REG. This way these registers can be programmed before the timer is activated.

The timer input registers ON-register, TIMER0_RELOAD_N_REG and TIMER0_RELOAD_M_REG can be written and the counter registers ON-counter and T0-counter can be read. If read from the address of the ON-register, the value of the ON-counter is returned. If read from the address of either the TIMER0_RELOAD_N_REG or the TIMER0_RELOAD_M_REG, the value of the T0-counter is returned.

It is possible to freeze the timer0 with bit 1 of the register SET_FREEZE_REG. When the timer is frozen the timer counters are not decremented. This will freeze all the timer registers at their last value. The timer will continue its operation again when FREEZE is cleared (bit 1 of register RESET_FREEZE_REG).

If the timer is switched off, all registers are switched into the reset state.

Interrupts

In 115.2 kHz mode, the interrupt routine must have a minimum duration of one 115.2 MHz cycle (8.6 us), else the interrupt will be executed again upon exiting

the ISR. So depending on CR16C+ clock frequency and available code, a small wait loop must be inserted. In 1.152 MHz mode, there is no problem:

29.2 TIMER1

Timer1 is used to generate an system clock tick interrupt to the CR16C+.

Features

- Programmable timer Frequency Mode 1
(1.152MHz or 115.2kHz) / (M+1)+(N+1)
(N=0..2**16-1, M=0-2**16-1)
- Programmable timer Frequency Mode 2
(1.152MHz or 115.2kHz) / N+1
(N=M shifting in time)
(N=0..2**16-1, M=0-2**16-1)

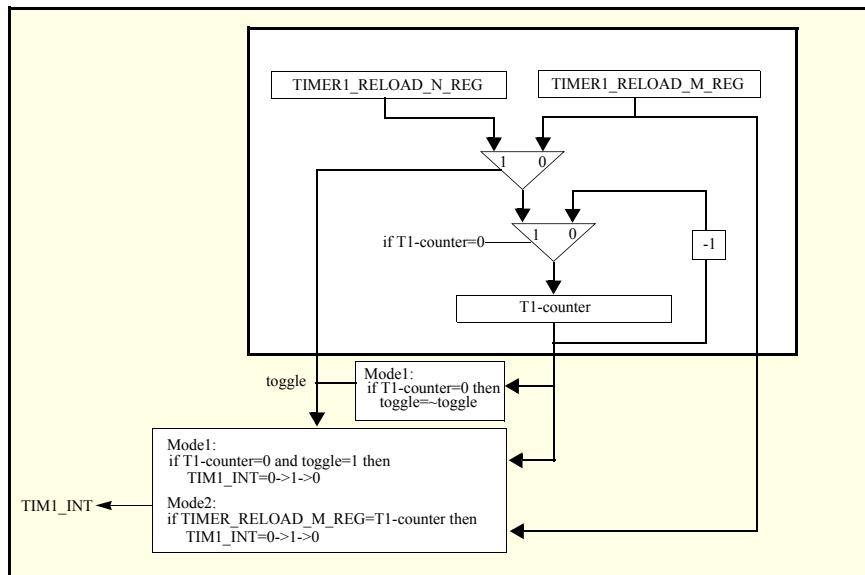


Figure 76 Timer1 block diagram

Figure 76 shows the timer 1 block diagram. The 16 bits timer consists of a counter, and two registers: **TIMER1_RELOAD_N_REG** and **TIMER1_RELOAD_M_REG**. On reset the counter and register values are 0000. Timer1 can be used as a general purpose timer.

The timer can be switched on and off with bit **TIM1_CTRL**. With bit **CLK_DIV8** of the **TIMER_CTRL_REG** a divider of 8 can be selected.

The timer input registers **TIMER1_RELOAD_N_REG** and **TIMER1_RELOAD_M_REG** can be written and the counter register **T1-counter** can be read. If read from the address of the **TIMER1_RELOAD_N_REG**, the value of the **T1-counter** is returned. If read from the address of the **TIMER1_RELOAD_M_REG**, the value of the **M1-register** is returned.

It is possible to freeze the timer with bit 2 of the register **SET_FREEZE_REG**. When the timer is frozen the timer counters are not decremented. This will freeze all the timer registers at their last value. The timer will continue its operation again when **FREEZE** is cleared (bit 2 of register **RESET_FREEZE_REG**).

If the timer is switched off, all registers are switched into the reset state.

If Timer1 is disabled all flipflops and outputs are in reset state except for the **TIMER1_RELOAD_N_REG** and the **TIMER1_RELOAD_M_REG**. This way these registers can be programmed before the timer is activated.

Timer1 mode1

After enabling this timer T1-counter is loaded with the value of the **TIMER1_RELOAD_M_REG**. The counter is then decremented until it reaches zero. Then the counter is loaded with the value of the **TIMER1_RELOAD_N_REG** and decremented again. Each time it reaches zero, the counter is alternately loaded with the values of the **TIMER1_RELOAD_M_REG** and the **TIMER1_RELOAD_N_REG**.

The **TIM1_INT** interrupt is activated when the **TIMER1_RELOAD_M_REG** is loaded into the counter, except for the first time when the timer is switched on.

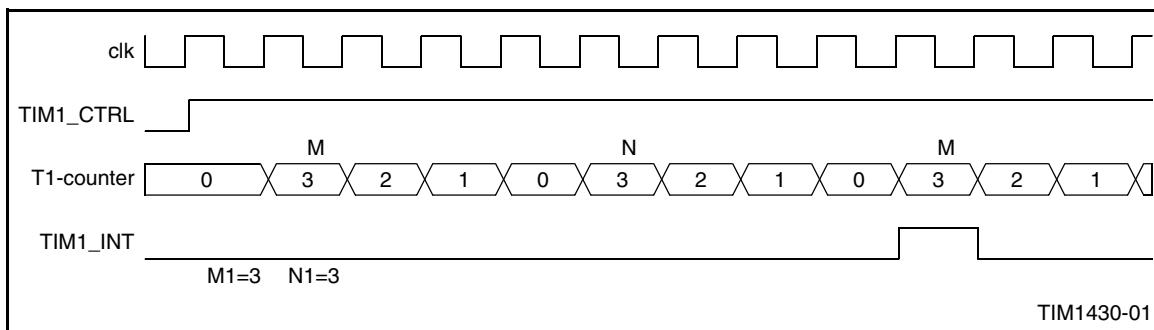


Figure 77 Timer 1 mode 1

Timer1 mode2

In this mode only the TIMER1_RELOAD_N_REG is repeatedly loaded into the counter when this reaches zero. A TIM1_INT interrupt will be generated each time the counter value is equal to the

TIMER1_RELOAD_M_REG value. This mechanism provides a constant interrupt repetition rate and the possibility to shift (fine tune) the interrupt trigger in time.

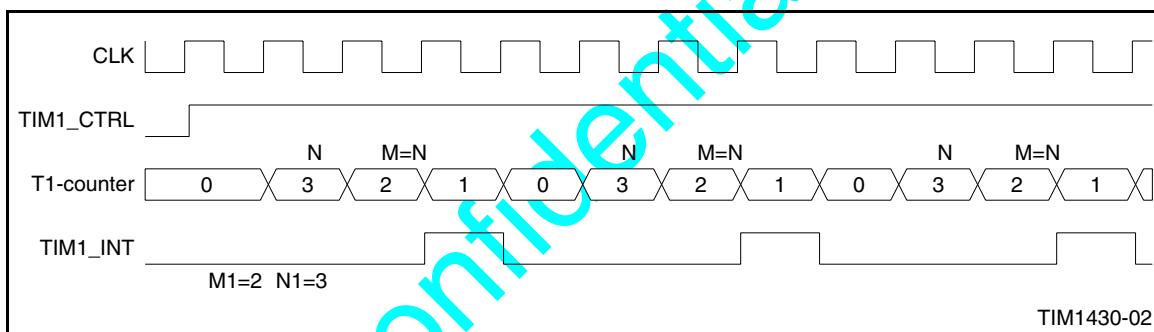


Figure 78 Timer 1 mode 2

29.3 WATCHDOG TIMER

The watchdog timer is an 8-bit timer that can be used to detect an unexpected execution sequence caused by a software run-away.

The watchdog timer consists of 8 bits, its contents is decremented by 1 every 10.66 msec. The WATCHDOG_REG is set to 0xFF at reset. This results in the maximum watchdog time of 2.56 seconds. If the watchdog reaches 0 the WATCHDOG_REG is set to 0xFF again. TIMER_CTRL_REG [WDOG_CTRL] can be set to either generate an NMI (default) or an internal HWreset if the watchdog timer reaches 0. In the latter case everything is reset but the RSTn pin is not pulled low and the TEST_ENV_REG is not reloaded. This control bit can only be set and will only be reset on a HWreset. See also Figure 12 on page 34 for an overview of the reset circuit.

Note that the NMI is shared with a DIP <U_VNMI> interrupt. Reading DIP1_STATUS_REG determines whether the DIP generated the interrupt or not.

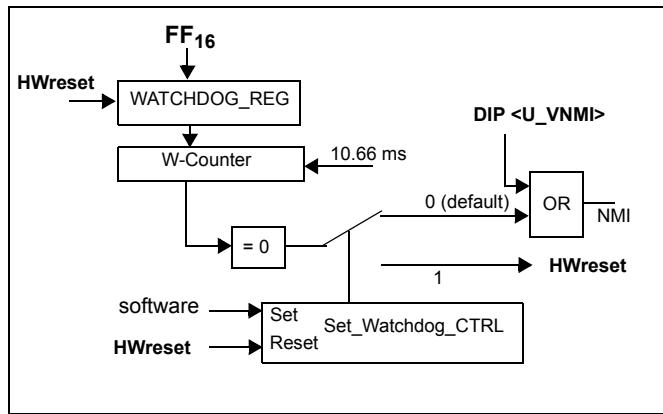


Figure 79 Watchdog timer

Confidential

29.4 CLOCK 100 TIMER

If CLK100_SRC bit in the DEBUG_REG is 0 then the clock 100 timer has a time period of 10 msec, synchronised to the execution of a DIP timer. If CLK100_SRC is 1 the clock 100 is a continuous signal with a time period of 10.66 msec. The clock 100 timer (10msec) is started if the sequencer program is started (URST = "0"). The clock 100 timer rising edge is synchronised to

the DIP SLOT_ZERO command. The clock 100 interrupt pending bit is set on every rising edge and/or falling edge of clock 100 depending on the DEBUG_REG[CLK100_POS, NEG] values. The active edge value can be read back with CLK100_EDGE. When the sequencer program is stopped (URST = "1") the clock timer is also stopped.

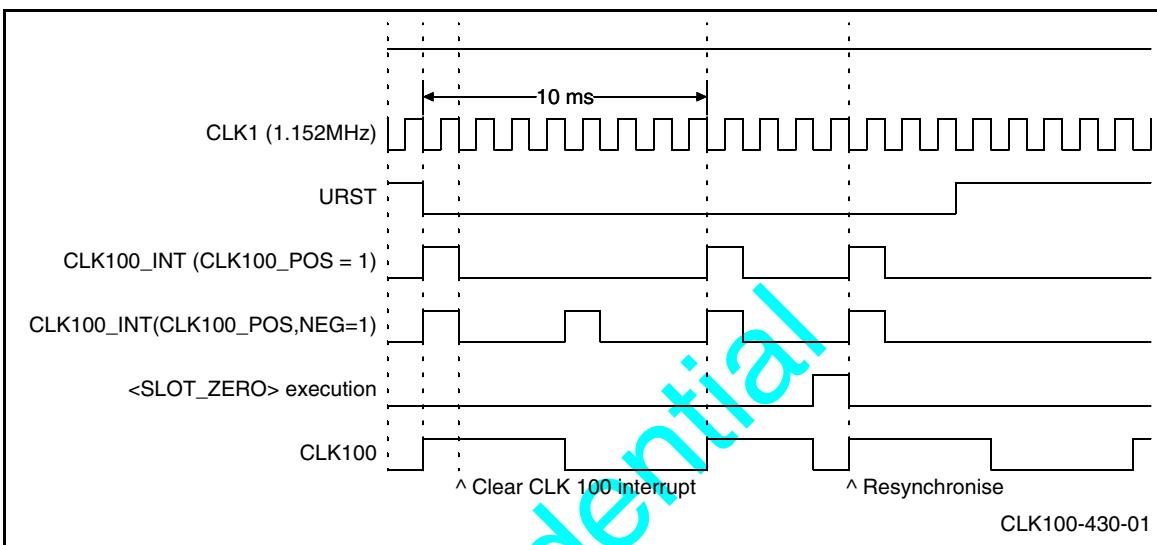


Figure 80 Clock 100 Timing

29.5 CAPTURE TIMERS/COUNTERS

The SC14452 has two programmable capture timer/counters (CT1 and CT2) to relief Gen2DSP measure frequencies of tones coming from the PSTN line or from the microphone. Figure 81 show the on-hook and off-hook basestation applications to detect ringing pulses with CT1 or CT2.

Table 53 shows the applications and how to select the multiplexers GATESRCx and CLKSRx as shown in Figure 82.

Table 53: Gate and clock selection

Application	GATESRC	CLKSRC
RINGING pin comparator	01	00
Call progress, ringing (ECZ via MICp/MICn pins and Gen2DSP)	10/11	00
Call progress, ring High Accuracy, high interrupt rate (ECZ via MICp/MICn pins and Gen2DSP)	00	10/11

The capture timer can either measure the time between two zero crossings (timer operation) or measure the number of zero crossing during a programmable time interval (counter operation).

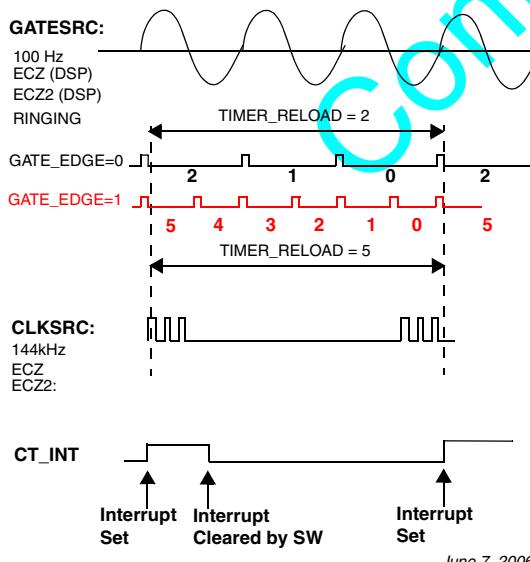


Figure 81 Timer/counter operation

The sources to measure the time between the zero crossings are selected with GATESRCx. The clocks that count the number of cycles are selected with CLKSRx in counter TONE_COUNTERx. (see Figure 82)

- GATESRC selection:
 - 00 = 100 Hz, the internal clock DECT frame clock
 - 01 = RINGING Comparator.
 - 10 = ECZ, from Gen2DSP Zero crossing detector 1 (optionally behind echo canceller)
 - 11 = ECZ2, from Gen2DSP Zero crossing det 2 (optionally behind echo canceller)

- CLKSR selection:

- 00 = 144 kHz, an internal clock
- 01 = Reserved
- 10 = ECZ1, from Gen2DSP Zero crossing det 1 (optionally behind echo canceller)
- 11 = ECZ2, from Gen2DSP Zero crossing det 2 (optionally behind echo canceller)

The ECZ and ECZ2 signal are outputs of the single bit Gen2DSP internal register DSP_ZCROSS1_OUT_REG and/or DSP_ZCROSS2_OUT_REG. Refer to chapter paragraph 12.12 on page 63 for more information.

The TONE_TIMERx determines the actual time interval for the measurement. It counts down from value determined by TIMER_RELOADx on every transition positive edge of GATESRCx. On value 0 the TONE_COUNTERx value is latched in TONE_LATCHx and the CTx_INT is set. If the mask interrupt bit MCT1x_INT is set to 1, an CT_INT interrupt in the ICU is set.

CTx_INT must be cleared by writing to TONE_CLEAR_INTx_REG. This resets also the TONE_LATCHx_REG automatically. CT_INT must be cleared using the RESET_INT_PENDING_REG.

With GATE_EDGE=0, only the rising edge of GATE_SRC is used to clock the TONE_TIMER. With GATE_EDGE=1, both edges are used. This feature gives higher resolution with e.g ringtone detection.

The formula below shows the value of the TONE_LATCHx register if a CT_INT interrupt is generated as function of frequencies of CLKSR, GATECLK input signals and TIMER_RELOADx value.

GATE_EDGE=0:
 $TONE_LATCHx = (TIMER_RELOADx+1) * F_{CLKSRCx} / F_{GATESRCx}$

GATE_EDGE=1:
 $TONE_LATCHx = (TIMER_RELOADx+1) * F_{CLKSRCx} / 2 * F_{GATESRCx}$

Example Figure 81
 $F_{CLKSRC}=144\text{kHz}$, $F_{GATESRC} = 4\text{ kHz}$

$$TONE_LATCH = (2+1)*144/4=108$$

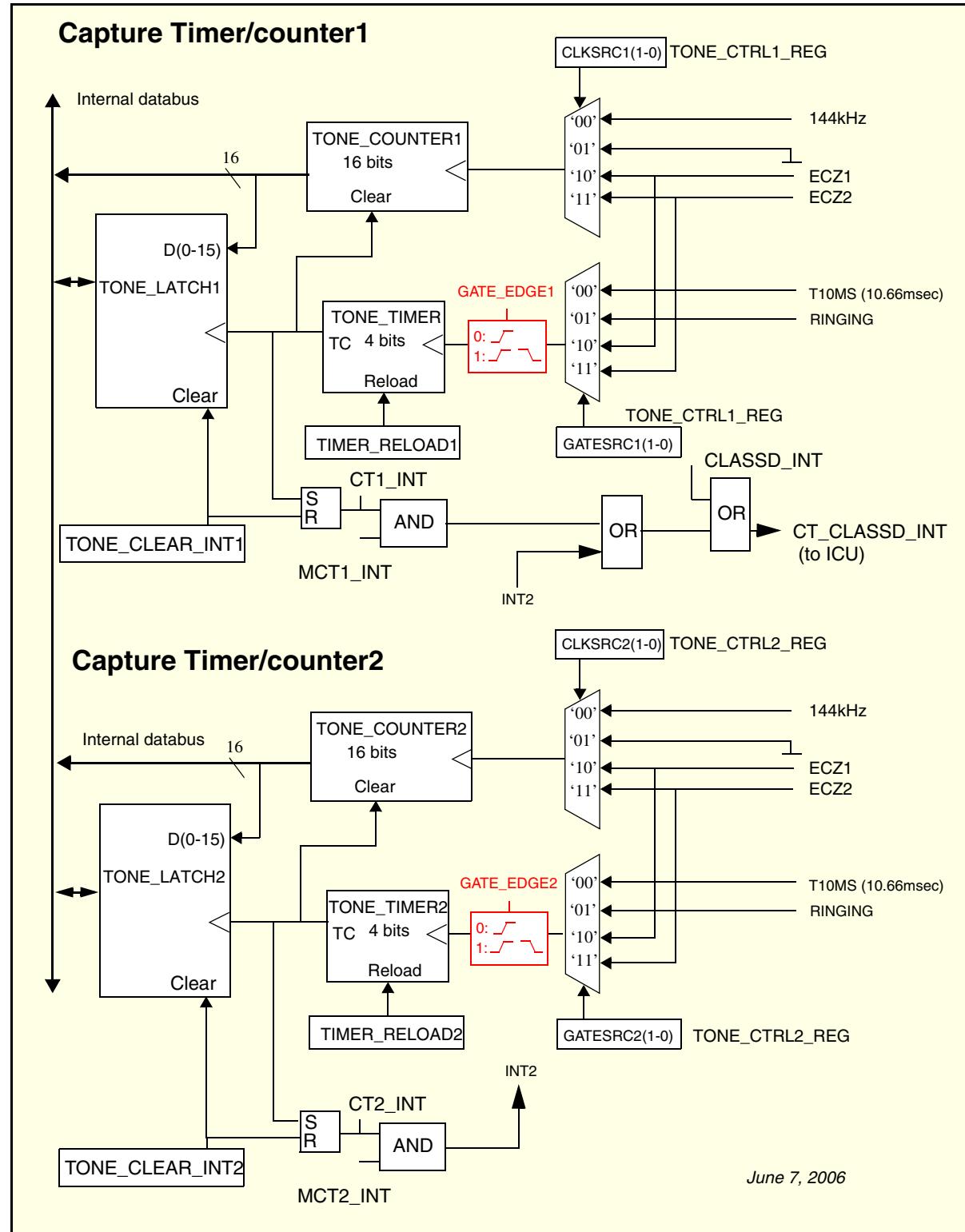


Figure 82 Capture timers block diagram

30.0 Ethernet MAC 10/100

The Ethernet MAC block* enables SC14452 to transmit and receive data over Ethernet in compliance with the IEEE 802.3 specification.

30.1 FEATURES

30.1.1 Ethernet MAC core

- Supports 10/100 Mbps data transfer rates
- Supports both IEEE 802.3-compliant MII and RMII interfaces to communicate with an external Fast Ethernet PHY
- Supports both full and half-duplex operation
 - Supports CSMA/CD Protocol for half-duplex operation
 - Supports IEEE 802.3x flow control for full-duplex operation
 - Optional forwarding of received pause control frames to the user application in full-duplex operation
 - Back-pressure support for half-duplex operation
 - Automatic transmission of zero-quanta pause frame on deassertion of flow control input in full duplex operation
- Preamble and start-of-frame data (SFD) insertion in Transmit, and deletion in Receive paths
- Automatic CRC and pad generation controllable on a per-frame basis
- Options for Automatic Pad/CRC Stripping on receive frames
- Programmable frame length to support Standard or Jumbo Ethernet frames with sizes up to 16 KB
- Programmable InterFrameGap (40-96 bit times in steps of 8)
- Implements 5 Station MAC Address register for 48-bit perfect (SA or DA) address filtering with masks for each byte.
- Promiscuous mode support to pass all frames without any filtering for network monitoring
- Separate 32-bit status returned for transmission and reception packets
- Supports IEEE 802.1Q VLAN tag detection for reception frames
- Separate transmission, reception, and control interfaces to the Application
- Network statistics with RMON/MIB (MMC) Counters (RFC2819/RFC2665)
- MDIO Master interface for PHY device configuration and management

* The Ethernet MAC block has been derived from the Synopsys® DesignWare® GMAC 10/100 v3.41a. Synopsys Inc. Proprietary. Used with permission.

30.1.2 DMA block

The DMA block exchanges data between the MTL block and host memory. A set of registers (DMA CSR) to control DMA operation is accessible by the host.

Note 14: No DMA access should be performed at the non-shared RAM range, i.e. at the range of 0x0000.0000 up to 0x0000.FFFF

- Single-channel independent Transmit and Receive engines
- Optimization for packet-oriented DMA transfers with frame delimiters
- Byte-aligned addressing for data buffer support
- Dual-buffer (ring) or linked-list (chained) descriptor chaining
- Descriptor architecture, allowing large blocks of data transfer with minimum CPU intervention; each descriptor can transfer up to 8 KB of data
- Comprehensive status reporting for normal operation and transfers with errors
- Individual programmable burst size for Transmit and Receive DMA Engines for optimal host bus utilization
- Programmable interrupt options for different operational conditions
- Per-frame Transmit/Receive complete interrupt control
- Round-robin or fixed-priority arbitration between Receive and Transmit engines
- Start/Stop modes
- Separate ports for host CSR access and host data interface
- Programmable fixed burst length (SINGLE, INCR4, INCR8, INCR16) or unspecified burst length (SINGLE, INCR) transfers.
- Takes care of AHB 1K boundary breaking

30.1.3 Transaction Layer - MTL

The MTL block consists of two sets of FIFOs: a Transmit FIFO with programmable threshold capability, and a Receive FIFO with a configurable threshold (default of 64 bytes).

- Single-channel Transmit and Receive engines
- Optimization for packet-oriented transfers with frame delimiters
- 128-Bytes Receive FIFO depth.
- 256-Bytes Transmit FIFO depth.
- Programmable burst-length support for starting a burst up to half the size of the MTL Rx and Tx FIFO
- Receive Status vectors inserted into the Receive FIFO after the EOF transfer enables multiple-frame storage in the Receive FIFO without requiring another FIFO to store those frames' Receive Status.

- Configurable Receive FIFO threshold (default fixed at 64 bytes) in Cut-Through mode
- Option to filter all error frames on reception and not forward them to the application in Store-and-Forward mode
- Option to forward under-sized good frames
- Supports Store and Forward mechanism for transmission to the Ethernet MAC core
- Supports threshold control for transmit buffer management
- Supports 2 frames to be stored in FIFO at any time.
- Automatic generation of PAUSE frame control or backpressure signal to the MAC core based on Receive FIFO-fill (threshold configurable) level.
- Handles automatic retransmission of Collision frames for transmission

- Discards frames on late collision, excessive collisions, excessive deferral and underrun conditions
- Software control to flush Tx FIFO
- Calculates and inserts IPv4 header checksum and TCP, UDP, or ICMP checksum in frames transmitted in Store-and-Forward mode.

30.1.4 Monitoring, Test and Debugging Support

- Supports internal loop-back on the MII for debugging
- DMA states (Tx and Rx) given as status bits
- Application Abort status bits
- MMC (RMON) module in the MAC core
- Current Tx/Rx Buffer pointer as status registers
- Current Tx/Rx Descriptor pointer as status registers

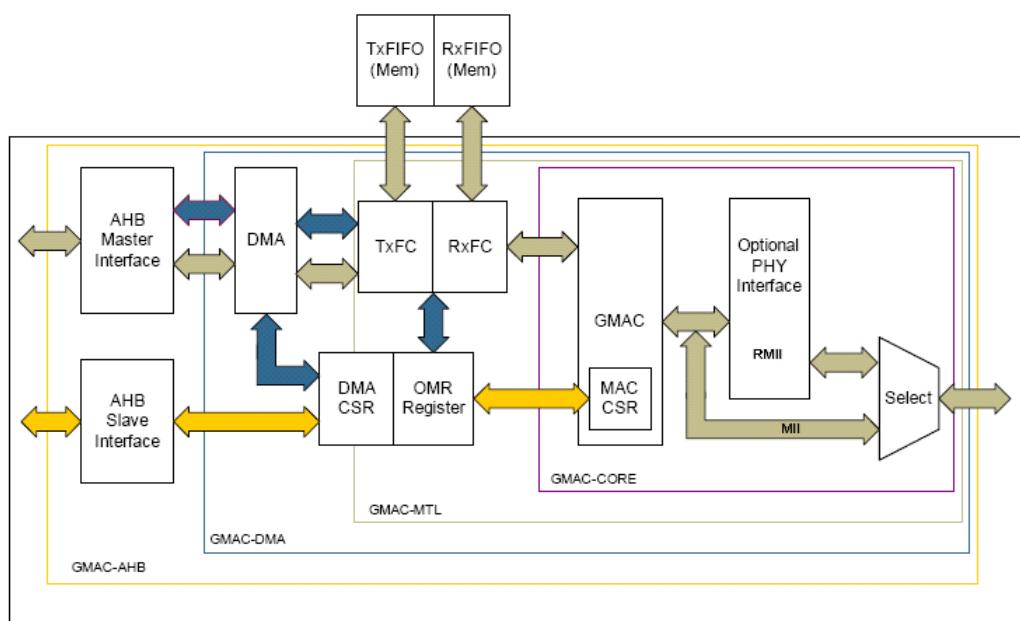


Figure 83 Ethernet MAC Block Diagram

30.2 INTERNAL ARCHITECTURE

The block diagram of the Ethernet MAC core and its four major sub-engines is presented in Figure 83. The following sections describe the internal engines.

30.3 DMA CONTROLLER

The DMA has independent Transmit and Receive engines, and a CSR space. The Transmit Engine transfers data from system memory to the device port (MTL), while the Receive Engine transfers data from the device port to system memory. The controller utilizes descriptors to efficiently move data from source to destination with minimal Host CPU intervention. The DMA is designed for packet-oriented data transfers such as frames in Ethernet. The controller can be programmed to interrupt the Host CPU for situations such

as Frame Transmit and Receive transfer completion, and other normal or error conditions.

The DMA and the Host driver communicate through two data structures:

- Control and Status registers (CSR)
- Descriptor lists and data buffers

The DMA transfers data frames received by the core to the Receive Buffer in the Host memory, and Transmit data frames from the Transmit Buffer in the Host memory. Descriptors that reside in the Host memory act as pointers to these buffers.

There are two descriptor lists; one for reception, and one for transmission. The base address of each list is written into DMA Register3 and Register4, respectively.

A descriptor list is forward linked (either implicitly or explicitly). The last descriptor may point back to the first entry to create a ring structure. Explicit chaining of descriptors is accomplished by setting the second address chained in both Receive and Transmit descriptors (RDES1[24] and TDES1[24]). The descriptor lists resides in the Host physical memory address space. Each descriptor can point to a maximum of two buffers. This enables two buffers to be used, physically addressed, rather than contiguous buffers in memory.

A data buffer resides in the Host physical memory

space, and consists of an entire frame or part of a frame, but cannot exceed a single frame. Buffers contain only data, buffer status is maintained in the descriptor. Data chaining refers to frames that span multiple data buffers. However, a single descriptor cannot span multiple frames. The DMA will skip to the next frame buffer when end-of-frame is detected. Data chaining can be enabled or disabled.

The descriptor ring and chain structure is shown in Figure 84.

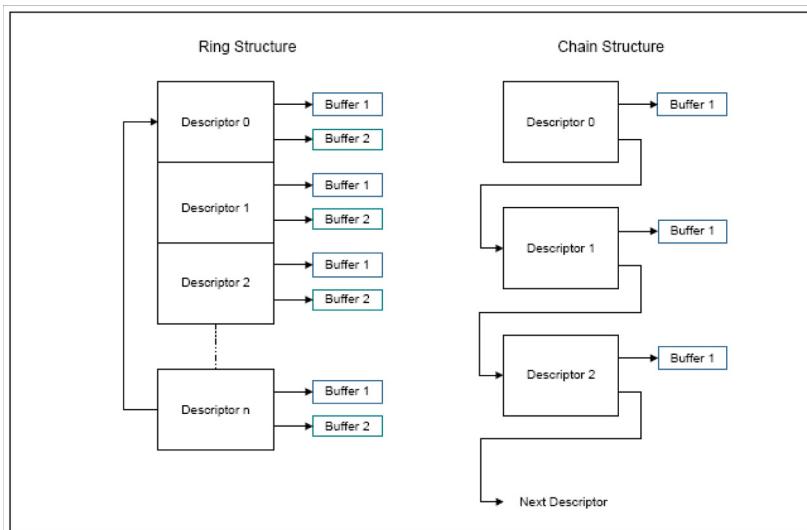


Figure 84. Descriptor Ring and Chain Structure

30.3.1 Initialization

Initialization for the MAC is as follows.

1. Write to DMA Register0 to set Host bus access parameters.
2. Write to DMA Register 7 to mask unnecessary interrupt causes.
3. The software driver creates the Transmit and Receive descriptor lists. Then it writes to both DMA Register 3 and DMA Register 4, providing the DMA with the starting address of each list.
4. Write to MAC Registers 1, 2 and 3 for desired filtering options.
5. Write to MAC Register 0 to configure and enable the Transmit and Receive operating modes. The PS and DM bits are set based on the auto-negotiation result (read from the PHY).
6. Write to DMA Register 6 to set bits 13 and 1 to start transmission and reception.
7. The Transmit and Receive engines enter the Running state and attempt to acquire descriptors from the respective descriptor lists. The Receive and Transmit engines then begin

processing Receive and Transmit operations. The Transmit and Receive processes are independent of each other and can be started or stopped separately.

30.3.1.1 Host Bus Burst Access

The DMA will attempt to execute fixed-length Burst transfers on the AHB Master interface if configured to do so (FB bit of DMA Register 0). The maximum Burst length is indicated and limited by the PBL field (DMA Register0[13:8]). The Receive and Transmit descriptors are always accessed in the maximum possible (limited by PBL or $16 * 8/\text{bus width}$) burst-size for the 16-bytes to be read.

The Transmit DMA will initiate a data transfer only when sufficient space to accommodate the configured burst is available in MTL Transmit FIFO or the number of bytes till the end of frame (when it is less than the configured burst-length). The DMA will indicate the start address and the number of transfers required to the AHB Master Interface. When the AHB Interface is configured for fixed-length burst, then it will transfer data using the best combination of INCR4/8/16 and SINGLE transactions. Otherwise (no fixed-length burst), it will transfer data using INCR (undefined

length) and SINGLE transactions.

The Receive DMA will initiate a data transfer only when sufficient data to accommodate the configured burst is available in MTL Receive FIFO or when the end of frame (when it is less than the configured burst-length) is detected in the Receive FIFO. The DMA will indicate the start address and the number of transfers required to the AHB Master Interface. When the AHB Interface is configured for fixed length burst, then it will transfer data using the best combination of INCR4/8/16 and SINGLE transactions. If the end-of frame is reached before the fixed-burst ends on the AHB interface, then dummy transfers are performed in-order to complete the fixed-burst. Otherwise (FB bit of DMA Register0 is reset), it will transfer data using INCR (undefined length) and SINGLE transactions.

When the AHB interface is configured for address-aligned beats, both DMA engines ensure that the first burst transfer the AHB initiates is less than or equal to the size of the configured PBL. Thus, all subsequent beats start at an address that is aligned to the configured PBL. The DMA can only align the address for beats up to size 16 (for PBL > 16), because the AHB interface does not support more than INCR16.

30.3.1.2 Host Data Buffer Alignment

The Transmit and Receive data buffers do not have any restrictions on data alignment (the start address for the buffers can be aligned to any of the four bytes). However, the DMA will always initiate transfers with address aligned to the bus width with dummy data for the byte lanes not required. This typically happens during the transfer of the beginning or end of an Ethernet frame.

30.3.1.3 Buffer Size Calculations

The DMA does not update the size fields in the Transmit and Receive descriptors. The DMA updates only the status fields (xDES0) of the descriptors. The driver has to perform the size calculations.

The transmit DMA transfers the exact number of bytes (indicated by buffer size field of TDES1) towards the MAC core. If a descriptor is marked as first (FS bit of TDES1 is set), then the DMA marks the first transfer from the buffer as the start of frame. If a descriptor is marked as last (LS bit of TDES1), then the DMA marks the last transfer from that data buffer as the end-of frame to the MTL.

The Receive DMA transfers data to a buffer until the buffer is full or the end-of frame is received from the MTL. If a descriptor is not marked as last (LS bit of RDES0), then the descriptor's corresponding buffer(s) are full and the amount of valid data in a buffer is accurately indicated by its buffer size field minus the data buffer pointer offset when the FS bit of that descriptor is set. The offset is zero when the data buffer pointer is aligned to the data bus width. If a descriptor is marked as last, then the buffer may not be full (as indicated by the buffer size in RDES1). To compute the amount of valid data in this final buffer, the driver must read the

frame length (FL bits of RDES0[29:16]) and subtract the sum of the buffer sizes of the preceding buffers in this frame. The Receive DMA always transfers the start of next frame with a new descriptor.

Note 15: Even when the start address of a receive buffer is not aligned to the system bus's data width, the system should allocate a receive buffer of a size aligned to the system bus width. For example, if the system allocates a 1,024-byte (1 KB) receive buffer starting from address 0x1000, the software can program the buffer start address in the Receive descriptor to have a 0x1002 offset. The Receive DMA writes the frame to this buffer with dummy data in the first two locations (0x1000 and 0x1001). The actual frame is written from location 0x1002. Thus, the actual useful space in this buffer is 1,022 bytes, even though the buffer size is programmed as 1,024 bytes, due to the start address offset.

30.3.1.4 DMA Arbiter

The arbiter inside the DMA module performs the arbitration between the Transmit and Receive channel accesses to the AHB Master interface. Two types of arbitrations are possible: round-robin, and fixed priority.

When round-robin arbitration is selected (DA bit of DMA Register0 is reset), the arbiter allocates the data bus in the ratio set by the PR bits of DMA Register0, when both Transmit and Receive DMAs are requesting for access simultaneously. When the DA bit is set, the Receive DMA always gets priority over the Transmit DMA for data access.

30.3.2 Transmission

30.3.2.1 TxDMA Operation: Default (Non-OSF) Mode

The Transmit DMA engine in default mode proceeds in the following sequence:

1. The Host sets up the transmit descriptor (TDES0-TDES3) and sets the Own bit (TDES0[31]) after setting up the corresponding data buffer(s) with Ethernet Frame data.
2. Once the ST bit (DMA Register6[13]) is set, the DMA enters the Run state.
3. While in the Run state, the DMA polls the Transmit Descriptor list for frames requiring transmission. After polling starts, it continues in either sequential descriptor ring order or chained order. If the DMA detects a descriptor flagged as owned by the Host, or if an error condition occurs, transmission is suspended and both the Transmit Buffer Unavailable (DMA Register5[2]) and Normal Interrupt Summary (DMA Register5[16]) bits are set. The Transmit Engine proceeds to Step 9.
4. If the acquired descriptor is flagged as owned by DMA (TDES0[31] = 1'b1), the DMA decodes the Transmit Data Buffer address from the acquired descriptor.
5. The DMA fetches the Transmit data from the

- Host memory and transfers the data to the MTL for transmission.
6. If an Ethernet frame is stored over data buffers in multiple descriptors, the DMA closes the intermediate descriptor and fetches the next descriptor. Steps 3, 4, and 5 are repeated until the end-of-Ethernet frame data is transferred to the MTL.
 7. When frame transmission is complete, status information is written into Transmit Descriptor 0 (TDES0) which has the end-of frame buffer.
 8. Transmit Interrupt (DMA Register5[0]) is set after completing transmission of a frame that has Interrupt on Completion (TDES1[31]) set in its Last Descriptor. The DMA engine then returns to Step 3.
 9. In Suspend state, the DMA tries to re-acquire the descriptor (jump to Step 3) when it receives a Transmit Poll demand and the Underflow Interrupt Status bit is cleared.

The default mode transmission flow is charted in Figure 85.

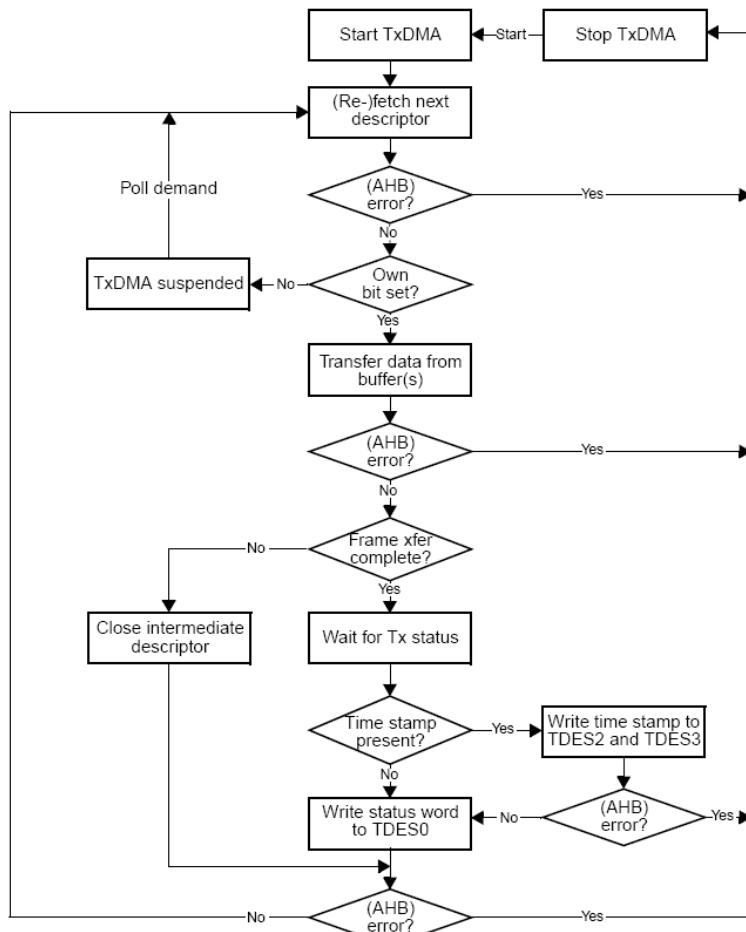


Figure 85 TxDMA Operation in Default Mode

30.3.2.2 TxDMA Operation: OSF Mode

While in the Run state, the transmit process can simultaneously acquire two frames without closing the Status descriptor of the first (if the OSF bit is set in DMA Register6[2]). As the transmit process finishes transferring the first frame, it immediately polls the Transmit Descriptor list for the second frame. If the second frame is valid, the transmit process transfers this frame before writing the status information of the first frame.

In OSF mode, the Run state Transmit DMA operates in the following sequence:

1. The DMA operates as described in steps 1–6 of the TxDMA (default mode).
2. Without closing the previous frame's descriptor, the DMA fetches the next descriptor.
3. If the DMA owns the acquired descriptor, the DMA decodes the transmit buffer address in this descriptor. If the DMA does not own the descriptor, the DMA goes into Suspend mode and skips to Step 7.
4. The DMA fetches the Transmit frame from the Host memory and transfers the frame to the MTL

- until the End-of-Frame data is transferred.
5. The DMA waits for the previous frame's frame transmission status and time stamp. Once the status is available, the DMA writes the time stamp to TDES2 and TDES3, if such time stamp was captured (as indicated by a status bit). The DMA then writes the status, with a cleared Own bit, to the corresponding TDES0, thus closing the descriptor. If time stamping was not enabled for the previous frame, the DMA does not alter the contents of TDES2 and TDES3.
 6. If enabled, the Transmit interrupt is set, the DMA fetches the next descriptor, then proceeds to Step 3 (when Status is normal). If the previous transmission status shows an underflow error,

- the DMA goes into Suspend mode (Step 7).
7. In Suspend mode, if a pending status and time stamp are received from the MTL, the DMA writes the time stamp (if enabled for the current frame) to TDES2 and TDES3, then writes the status to the corresponding TDES0. It then sets relevant interrupts and returns to Suspend mode.
 8. The DMA can exit Suspend mode and enter the Run state (go to Step 1 or Step 2 depending on pending status) only after receiving a Transmit Poll demand (DMA Register 1).

The basic flow is charted in Figure 86.

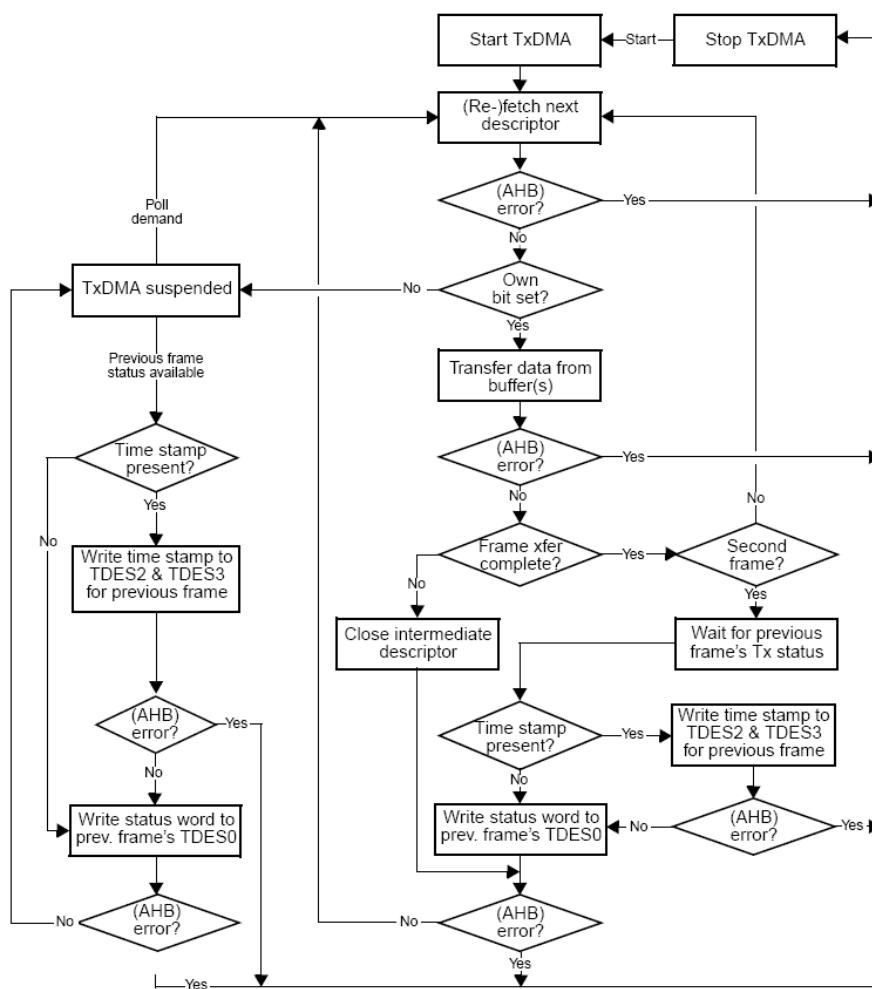


Figure 86 TxDMA Operation in OSF Mode

30.3.2.3 Transmit Frame Processing

The Transmit DMA expects that the data buffers contain complete Ethernet frames, excluding preamble, pad bytes, and FCS fields. The DA, SA, and Type/Len fields contain valid data. If the Transmit Descriptor indicates that the MAC core must disable CRC or PAD

insertion, the buffer must have complete Ethernet frames (excluding preamble), including the CRC bytes.

Frames can be data-chained and can span several buffers. Frames must be delimited by the First Descriptor (TDES1[29]) and the Last Descriptor (TDES1[30]), respectively.

As transmission starts, the First Descriptor must have TDES1[29] set. When this occurs, frame data transfers from the Host buffer to the MTL Transmit FIFO. Concurrently, if the current frame has the Last Descriptor (TDES1[30]) clear, the Transmit Process attempts to acquire the Next Descriptor. The Transmit Process expects this descriptor to have TDES1[29] clear. If TDES1[30] is clear, it indicates an intermediary buffer. If TDES1[30] is set, it indicates the last buffer of the frame.

After the last buffer of the frame has been transmitted, the DMA writes back the final status information to the Transmit Descriptor 0 (TDES0) word of the descriptor that has the last segment set in Transmit Descriptor 1 (TDES1[30]). At this time, if Interrupt on Completion (TDES1[31]) was set, Transmit Interrupt (DMA Register5[0]) is set, the Next Descriptor is fetched, and the process repeats.

Actual frame transmission begins after the MTL Transmit FIFO has reached either a programmable transmit threshold (DMA Register6[16:14]), or a full frame is contained in the FIFO. There is also an option for Store and Forward Mode (DMA Register6[21]). Descriptors are released (Own bit TDES0[31] clears) when the DMA finishes transferring the frame.

30.3.2.4 Transmit Polling Suspended

Transmit polling can be suspended by either of the following conditions:

- The DMA detects a descriptor owned by the Host (TDES0[31]=0). To resume, the driver must give descriptor ownership to the DMA and then issue a Poll Demand command.
- A frame transmission is aborted when a Transmit Error (like underflow) is detected. The appropriate Transmit Descriptor 0 (TDES0) bit is set.

If the second condition occur, both Abnormal Interrupt Summary (DMA Register5[15]) and Transmit Underflow bits (DMA Register5 [5]) are set, and the information is written to Transmit Descriptor 0, causing the suspension. If the DMA goes into SUSPEND state due to the first condition, then both Normal Interrupt Summary (DMA Register5 [16]) and Transmit Buffer Unavailable (DMA Register5 [2]) are set.

In both cases, the position in the Transmit List is retained. The retained position is that of the descriptor following the Last Descriptor closed by the DMA.

The driver must explicitly issue a Transmit Poll Demand command after rectifying the suspension cause.

30.3.3 Reception

The Receive DMA engine's reception sequence is depicted in Figure 87 and proceeds as follows:

1. The host sets up Receive descriptors (RDES0-RDES3) and sets the Own bit (RDES0[31]).
2. Once the SR (DMA Register6[1]) bit is set, the DMA enters the Run state. While in the Run

state, the DMA polls the Receive Descriptor list, attempting to acquire free descriptors. If the fetched descriptor is not free (is owned by the host), the DMA enters the Suspend state and jumps to Step 9.

3. The DMA decodes the receive data buffer address from the acquired descriptors.
4. Incoming frames are processed and placed in the acquired descriptor's data buffers.
5. When the buffer is full or the frame transfer is complete, the Receive engine fetches the next descriptor.
6. If the current frame transfer is complete, the DMA proceeds to Step 7. If the DMA does not own the next fetched descriptor and the frame transfer is not complete (EOF is not yet transferred), the DMA sets the Descriptor Error bit in the RDES0 (unless flushing is disabled). The DMA closes the current descriptor (clears the Own bit) and marks it as intermediate by clearing the Last Segment (LS) bit in the RDES0 value (marks it as Last Descriptor if flushing is not disabled), then proceeds to Step 8. If the DMA does own the next descriptor but the current frame transfer is not complete, the DMA closes the current descriptor as intermediate and reverts to Step 4.
7. The Receive engine checks the latest descriptor's Own bit. If the host owns the descriptor (Own bit is 0) the Receive Buffer Unavailable bit (Register5[7]) is set and the DMA Receive engine enters the Suspend state (Step 9). If the DMA owns the descriptor, the engine returns to Step 4 and awaits the next frame.
8. Before the Receive engine enters the Suspend state, partial frames are flushed from the Receive FIFO (You can control flushing using Bit 24 of DMA Register6).
9. The Receive DMA exits the Suspend state when a Receive Poll demand is given or the start of next frame is available from the MTL's Receive FIFO. The engine proceeds to Step 2 and refetches the next descriptor.

The DMA does not acknowledge accepting the status from the MTL until it has completed the time stamp write-back and is ready to perform status write-back to the descriptor.

If software has enabled time stamping through CSR, when a valid time stamp value is not available for the frame (for example, because the receive FIFO was full before the time stamp could be written to it), the DMA writes all-ones to RDES2 and RDES3. Otherwise (that is, if time stamping is not enabled), the RDES2 and RDES3 remain unchanged.

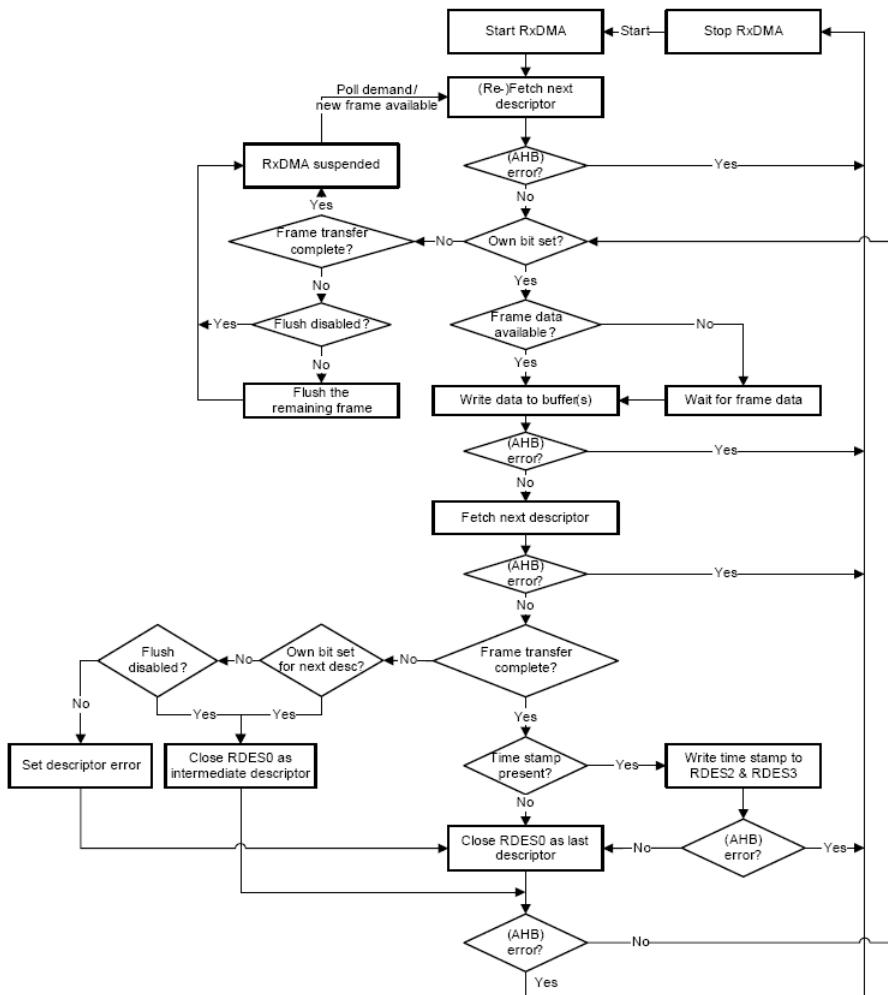


Figure 87 Receive DMA Operation

30.3.3.1 Receive Descriptor Acquisition

The Receive Engine always attempts to acquire an extra descriptor in anticipation of an incoming frame. Descriptor acquisition is attempted if any of the following conditions is satisfied:

- The receive Start/Stop bit (Register6[1]) has been set immediately after being placed in the Run state.
 - The data buffer of current descriptor is full before the frame ends for the current transfer.
 - The controller has completed frame reception, but the current Receive Descriptor is not yet closed.
 - The receive process has been suspended because of a host-owned buffer (RDES0[31] = 0) and a new frame is received.
 - A Receive Poll demand has been issued.

30.3.3.2 Receive Frame Processing

The MAC transfers the received frames to the Host

memory only when the frame passes the address filter and frame size is greater than or equal to configurable threshold bytes set for the Receive FIFO of MTL, or when the complete frame is written to the FIFO in Store-and-Forward mode.

If the frame fails the address filtering, it is dropped in the MAC block itself (unless Receive All MAC Register1[31] bit is set). Frames that are shorter than 64 bytes, because of collision or premature termination, can be purged from the MTL Receive FIFO.

After 64 (configurable threshold) bytes have been received, the MTL block requests the DMA block to begin transferring the frame data to the Receive Buffer pointed to by the current descriptor. The DMA sets First Descriptor (RDES0[9]) after the DMA Host Interface (AHB) becomes ready to receive a data transfer (if DMA is not fetching transmit data from the host), to delimit the frame. The descriptors are released when the Own (RDES[31]) bit is reset to 0, either as the Data buffer fills up or as the last segment of the frame is

transferred to the Receive buffer. If the frame is contained in a single descriptor, both Last Descriptor (RDES[8]) and First Descriptor (RDES[9]) are set.

The DMA fetches the next descriptor, sets the Last Descriptor (RDES[8]) bit, and releases the RDES0 status bits in the previous frame descriptor. Then the DMA sets Receive Interrupt (Register5[6]). The same process repeats unless the DMA encounters a descriptor flagged as being owned by the host. If this occurs, the Receive Process sets Receive Buffer Unavailable (Register5[7]) and then enters the Suspend state. The position in the receive list is retained.

30.3.3.3 Receive Process Suspended

If a new Receive frame arrives while the Receive Process is in Suspend state, the DMA refetches the current descriptor in the Host memory. If the descriptor is now owned by the DMA, the Receive Process re-enters the Run state and starts frame reception. If the descriptor is still owned by the host, by default, the DMA discards the current frame at the top of the MTL Rx FIFO and increments the missed frame counter. If more than one frame is stored in the MTL Rx FIFO, the process repeats.

The discarding or flushing of the frame at the top of the MTL Rx FIFO can be avoided by setting Operation Mode register bit 24 (DFF). In such conditions, the receive process sets the Receive Buffer Unavailable status and returns to the Suspend state.

30.3.4 Interrupts

Interrupts can be generated as a result of various events. DMA Register5 contains all the bits that might cause an interrupt. DMA Register7 contains an Enable bit for each of the events that can cause an interrupt.

There are two groups of interrupts, Normal and Abnormal, as described in DMA Register5. Interrupts are cleared by writing a "1" to the corresponding bit position. When all the enabled interrupts within a group are cleared, the corresponding summary bit is cleared. When both the summary bits are cleared, the interrupt signal to the CPU is deasserted. If the MAC core is the cause for assertion of the interrupt, then any of the GLI, GMI, or GPI bits of DMA Register5 will be set high.

Interrupts are not queued and if the interrupt event occurs before the driver has responded to it, no additional interrupts are generated. For example, Receive Interrupt (DMA Register5[6]) indicates that one or more frames was transferred to the Host buffer. The driver must scan all descriptors, from the last recorded position to the first one owned by the DMA.

An interrupt is generated only once for simultaneous, multiple events. The driver must scan DMA Register5 for the interrupt cause. The interrupt is not generated again, unless a new interrupting event occurs after the driver has cleared the appropriate DMA Register5 bit. For example, the controller generates a Receive Interrupt (DMA Register5[6]) and the driver begins reading DMA Register5. Next, Receive Buffer Unavailable

(DMA Register5[7]) occurs. The driver clears the Receive Interrupt. Even then, the CPU IRQ signal is not deasserted, due to the active or pending Receive Buffer Unavailable interrupt.

30.4 MAC TRANSACTION LAYER (MTL)

The MAC Transaction Layer provides FIFO memory to buffer and regulate the frames between the application system memory and the MAC core. The MTL layer has 2 data paths, namely the Transmit path and the Receive Path. The data path for both directions is 32-bit wide and operates with a simple FIFO protocol.

The MAC-MTL communicates with the application side with the Application Transmit Interface (ATI), Application Receive Interface (ARI), and the MAC Control Interface (MCI).

30.4.1 Transmit Path

The DMA controls all transactions for the transmit path through the ATI. Ethernet frames read from the system memory is pushed into the FIFO by the DMA. The frame is then popped out and transferred to the MAC core when triggered. When the end-of-frame is transferred, the status of the transmission is taken from the MAC core and transferred back to the DMA.

The Transmit FIFO depth is 256-bytes. FIFO-fill level is indicated to the DMA so that it can initiate a data fetch in required bursts from the system memory, using the AHB interface. The data from the AHB Master interface is pushed into the FIFO with the appropriate byte lanes qualified by the DMA. The DMA also indicates the start-of-frame (SOF) and end-of-frame (EOF) transfers along with a few sideband signals controlling the pad-insertion/CRC generation for that frame in the MAC Core.

Per-frame control bits, such as Automatic Pad/CRC Stripping disable, time stamp capture, and so forth are taken as sideband control inputs on the ATI, stored in a separate register FIFO, and passed on to the core transmitter when the corresponding frame data is read from the Transmit FIFO.

There are two modes of operation for popping data towards the MAC core. In Threshold mode, as soon as the number of bytes in the FIFO crosses the configured threshold level (or when the end-of-frame is written before the threshold is crossed), the data is ready to be popped out and forwarded to the MAC core. The threshold level is configured using the TTC bits of DMA Register0.

In Store-and-Forward mode, only after a complete frame is stored in the FIFO, the MTL pops the frame towards the MAC core. If the Tx FIFO size is smaller than the Ethernet frame to be transmitted (such as Jumbo frame), then the MTL pops the frame towards the MAC core when the Tx FIFO becomes almost full or when the ATI watermark becomes low. Therefore, the MTL never stalls in Store and Forward mode even if the Ethernet frame length is bigger than the Tx FIFO depth.

The application can flush the Transmit FIFO of all contents by setting the FTF (DMA Register6[20]) bit. This bit is self-clearing and initializes the FIFO pointers to the default state. If the FTF bit is set during a frame transfer from the MTL to the MAC core, then the MTL stops further transfer as the FIFO is considered to be empty. Hence an underflow event occurs at the MAC transmitter and the corresponding Status word is forwarded to the DMA.

30.4.1.1 Initialization

Upon reset, the MTL is ready to manage the flow of data to and from the DMA and the MAC.

There are no requirements for enabling the MTL. However, the MAC block and the DMA controller must be enabled individually through their respective CSRs.

30.4.1.2 Single-Packet Transmit Operation

During a transmit operation, the MTL block is slaved to the DMA controller. The general sequence of events for a transmit operation is as follows.

1. If the system has data to be transferred, the DMA controller, if enabled, fetches data from the Host through the AHB Master interface and starts forwarding it to the MTL. The MTL pushes the data received from the DMA into the FIFO. It continues to receive the data until the end-of-frame of the frame is transferred.
2. The data is taken out of the FIFO and sent to the MAC by the FIFO controller engine. When the threshold level is crossed or a full packet of data is received into the FIFO, the MTL pops out the frame data and drives them to the MAC core. The engine continues to transfer data from the FIFO until a complete packet has been transferred to the MAC. Upon completion of the frame, the MTL receives the Status from the MAC and then notifies the DMA controller

30.4.1.3 Transmit Operation — Two Packets in the Buffer

1. Because the DMA must update the descriptor status before releasing it to the Host, there can be at the most two frames inside a transmit FIFO. The second frame will be fetched by the DMA and put into the FIFO only if the OSF (Operate on Second Frame) bit is set. If this bit is not set, the next frame will be fetched from the memory only after the MAC has completely processed the frame and the DMA has released the descriptors.
2. If the OSF bit is set, the DMA starts fetching the second frame immediately after completing the transfer of the first frame to the FIFO. It does not wait for the status to be updated. The MTL, in the meantime, receives the second frame into the FIFO while transmitting the first frame. As soon as the first frame has been transferred and the status is received from the MAC, the MTL

pushes it to the DMA. If the DMA has already completed sending the second packet to the MTL, it must wait for the status of the first packet before proceeding to the next frame.

30.4.1.4 Retransmission During Collision

While a frame is being transferred from the MTL to the MAC, a collision event occurs on the MAC line interface in Half-Duplex mode. The MAC then indicates a retry attempt to the MTL by giving the status even before the end-of-frame is transferred from MTL. Then the MTL will enable the retransmission by popping out the frame again from the FIFO.

After more than 96 bytes (or 548 bytes in 1000-Mbps mode) are popped towards the MAC core, the FIFO controller frees up that space and makes it available to the DMA to push in more data. This means that the retransmission is not possible after this threshold is crossed or when the MAC core indicates a late-collision event.

30.4.1.5 Transmit FIFO Flush Operation

The MAC provides a control to the software to flush the Transmit FIFO in the MTL layer through the use of bit 20 of the Operation Mode register (see “Register6 (Operation Mode Register)”). The Flush operation is immediate and the MTL clears the Tx FIFO and the corresponding pointers to the initial state even if it is in the middle of transferring a frame to the MAC Core. This will result in an underflow event in the MAC Transmitter, and the frame transmission is aborted. The status of such a frame will be marked with both Underflow and Frame Flush events (TDES0, bits 13 and 1).

The MTL layer also stops accepting any data from the application (MDC module) during the Flush operation. It will generate and transfer Transmit Status Words to the application for the number of frames that is flushed inside the MTL (including partial frames). Frames that are completely flushed in the MTL will have the Frame Flush Status bit (TDES0 13) set. The MTL completes the Flush operation when the application (DMA) accepts all of the Status Words for the frames that were flushed, and then clears the Transmit FIFO Flush control register bit. At this point, the MTL starts accepting new frames from the application (DMA).

30.4.1.6 Transmit Checksum Offload Engine

Communication protocols such as TCP and UDP implement checksum fields, which help determine the integrity of data transmitted over a network. Because the most widespread use of Ethernet is to encapsulate TCP and UDP over IP datagrams, the MAC core has a Checksum Offload Engine (COE) to support checksum calculation and insertion in the transmit path, and error detection in the receive path.

Note 16: The checksum for TCP, UDP, or ICMP is calculated over a complete frame, then inserted into its corresponding header field. Due to this requirement, this function is enabled only when the Transmit FIFO is configured for Store-and-Forward

mode (that is, when the TSF bit is set in DMA Register6). If the core is configured for Threshold (cut-through) mode, the Transmit COE is bypassed.

Note 17: You must make sure the Transmit FIFO is deep enough to store a complete frame before that frame is transferred to the MAC Core transmitter. If the FIFO depth is less than the input Ethernet frame size, the payload (TCP/UDP/ICMP) checksum insertion function is bypassed and only the frame's IPv4 Header checksum is modified, even in Store-and-Forward mode.

This checksum engine can be controlled for each frame by setting the CIC bits (Bits 28:27 of TDES1, described in "Transmit Descriptor 1 (TDES1)").

30.4.1.6.1 IP Header Checksum Engine

In IPv4 datagrams, the integrity of the header fields is indicated by the 16-bit Header Checksum field (the eleventh and twelfth bytes of the IPv4 datagram). The COE detects an IPv4 datagram when the Ethernet frame's Type field has the value 0x0800 and the IP datagram's Version field has the value 0x4. The input frame's checksum field is ignored during calculation and replaced with the calculated value.

IPv6 headers do not have a checksum field; thus, the COE does not modify IPv6 header fields.

The result of this IP header checksum calculation is indicated by the IP Header Error status bit in the Transmit status (Transmit Descriptor 0, bit 16). This status bit is set whenever the values of the Ethernet Type field and the IP header's Version field are not consistent, or when the Ethernet frame does not have enough data, as indicated by the IP header Length field.

In other words, this bit is set when an IP header error is asserted under the following circumstances:

For IPv4 datagrams:

- The received Ethernet type is 0x0800, but the IP header's Version field does not equal 0x4
- The IPv4 Header Length field indicates a value less than 0x5 (20 bytes)
- The total frame length is less than the value given in the IPv4 Header Length field

For IPv6 datagrams:

- The Ethernet type is 0x86DD but the IP header Version field does not equal 0x6
- The frame ends before the IPv6 header (40 bytes) or extension header (as given in the corresponding Header Length field in an extension header) is completely received.

Even when the COE detects such an IP header error, it inserts an IPv4 header checksum if the Ethernet Type field indicates an IPv4 payload.

30.4.1.6.2 TCP/UDP/ICMP Checksum Engine

The TCP/UDP/ICMP Checksum Engine processes the IPv4 or IPv6 header (including extension headers) and determines whether the encapsulated payload is TCP,

UDP, or ICMP.

Note 18: For non-TCP, -UDP, or -ICMP/ICMPv6 payloads, this checksum engine is bypassed and nothing further is modified in the frame.

Note 19: Fragmented IP frames (IPv4 or IPv6), IP frames with security features (such as an authentication header or encapsulated security payload), and IPv6 frames with routing headers are bypassed and not processed by this engine.

The checksum is calculated for the TCP, UDP, or ICMP payload and inserted into its corresponding field in the header. This engine can work in the following two modes:

- In the first mode, the TCP, UDP, or ICMPv6 pseudo-header is not included in the checksum calculation and is assumed to be present in the input frame's Checksum field. This engine includes the Checksum field in the checksum calculation, then replaces the Checksum field with the final calculated checksum.
- In the second mode, the engine ignores the Checksum field, includes the TCP, UDP, or ICMPv6 pseudo-header data into the checksum calculation, and overwrites the checksum field with the final calculated value.

Note 20: For ICMP-over-IPv4 packets, the Checksum field in the ICMP packet must always be 16'h0000 in both modes, because pseudo-headers are not defined for such packets. If it does not equal 16'h0000, an incorrect checksum may be inserted into the packet.

The result of this operation is indicated by the Payload Checksum Error status bit in the Transmit Status vector (Bit 12 in Transmit Descriptor 0 - TDES0). This engine sets the Payload Checksum Error status bit when it detects that the frame has been forwarded to the MAC Transmitter engine in Store-and-Forward mode without the end-of-frame being written to the FIFO, or when the packet ends before the number of bytes indicated by the Payload Length field in the IP Header is received. When the packet is longer than the indicated payload length, the COE ignores them as stuff bytes, and no error is reported. When this engine detects the first type of error, it does not modify the TCP, UDP, or ICMP header. For the second error type, it still inserts the calculated checksum into the corresponding header field.

30.4.2 Receive Path

This module receives the frames given out by the MAC core and pushes them into the Rx FIFO. The status (fill level) of this FIFO is indicated to the DMA once it crosses the configured Receive threshold (RTC of DMA Register 6). The MTL also indicates the FIFO fill level so that the DMA can initiate pre-configured burst transfers towards the AHB interface.

30.4.2.1 Receive Operation

During an Rx operation, the MTL is slaved to the MAC. The general sequence of Receive operation events is as follows:

1. When the MAC receives a frame, it pushes in data along with byte enables. The MAC also indi-

cates the SOF and EOF. The MTL accepts the data and pushes it into the Rx FIFO. After the EOF is transferred, the MAC drives the status word, which is also pushed into the same Rx FIFO by the MTL.

2. The MTL_RX engine takes the data out of the FIFO and sends it to the DMA. In the default Cut-Through mode, when 64 bytes (configured with RTC bits of DMA Register 6) or a full packet of data are received into the FIFO, The MTL_RX engine pops out the data and indicates its availability to the DMA. Once the DMA initiates the transfer to the AHB interface, the MTL_RX engine continues to transfer data from the FIFO until a complete packet has been transferred. Upon completion of the EOF frame transfer, the MTL pops out the status word and sends it to the DMA controller.
3. In Rx FIFO Store-and-Forward mode (configured by the RSF bit of DMA Register6), a frame is read out only after being written completely into the Receive FIFO. In this mode, all error frames are dropped (if the core is configured to do so) such that only valid frames are read out and forwarded to the application. In Cut-Through mode, some error frames are not dropped, because the error status is received at the end-of-frame, by which time the start of that frame has already been read out of the FIFO.

30.4.2.2 Receive Operation Multiframe Handling

Since the status is available immediately following the data, the MTL is capable of storing any number of frames into the FIFO, as long as it is not full.

30.4.2.3 Error Handling

If the MTL Rx FIFO is full before it receives the EOF data from the MAC, an overflow is declared and the whole frame is dropped, and the overflow counter in the DMA (Register8) is incremented. If the start address of such a frame has already been transferred to the Read Controller, the rest of the frame is dropped and a dummy EOF is written to the FIFO along with the status word. The status will indicate a partial frame due to overflow.

The MTL Rx Control logic can filter error and undersized frames, if enabled (using the DMA Register6 FEF and FUF bits). If the start address of such a frame has already been transferred to the Rx FIFO Read Controller, that frame is not filtered. The start address of the frame is transferred to the Read Controller after the frame crosses the receive threshold (set by the DMA Register6 RTC bits).

If the MTL Receive FIFO is configured to operate in Store-and-Forward mode, all error frames can be filtered and dropped.

If you enable the Frame Length FIFO in MAC-MTL configuration (refer to "Frame Length Interface"), then error frames can be filtered even after the start address of

the frame is transferred to the Read Controller in the default Cut-Through mode. In this configuration, if a frame's status and length are available to the Read Controller using the Frame Length FIFO when that frame's SOF is read from the Rx FIFO, then the complete erroneous frame can be dropped indirectly in MTL.

30.4.2.4 Receive Status Word

At the end of the transfer of the Ethernet frame to the host, the MTL outputs the receive status. The detailed description of the receive status is the same as for Bits[31:0] of RDES0, except that Bits 31, 14, 9, and 8 are reserved and have a reset of "0" by default.

30.5 MAC CORE

The MAC core supports many interfaces towards the PHY chip. The PHY interface can be selected only once after reset. The MAC core communicates with the application side with the MAC Transmit Interface (MTI), MAC Receive Interface (MRI) and the MAC Control Interface (MCI).

30.5.1 Transmission

When the SOF signal is detected, the MAC accepts the data and begins transmitting to the MII. The time required to transmit the frame data to the MII after the Application initiates transmission is variable, depending on delay factors like IFG delay, time to transmit preamble/SFD, and any back-off delays for Half-Duplex mode.

After the EOF is transferred to the MAC Core, the core complete normal transmission and then gives the Status of Transmission back to the MTL. If a normal collision (in Half-duplex mode) occurs during transmission, the MAC core makes valid the Transmit Status to the MTL. It will then accept and drop all further data until the next SOF is received. The MTL block should retransmit the same frame from SOF on observing a Retry request (in the Status) from the MAC.

The MAC issues an underflow status if the MTL is not able to provide the data continuously during the transmission. During the normal transfer of a frame from MTL, if the MAC receives a SOF without getting an EOF for the previous frame, then it (the SOF) is ignored and the new frame is considered as continuation of the previous frame.

The following six modules constitute the transmission function of the MAC:

- Transmit Bus Interface Module (TBU)
- Transmit Frame Controller Module (TFC)
- Transmit Protocol Engine Module (TPE)
- Transmit Scheduler Module (STX)
- Transmit CRC Generator Module (CTX)
- Transmit Flow Control Module (FTX)

30.5.1.1 Transmit Frame Controller Module (TFC)

The Transmit Frame Controller (TFC) consists of two registers to hold data, byte enables, and the last data control received from the TBU. The register provides a buffer between the Application and the TPE to regulate data flow.

When the number of bytes received from the Application falls below 60 (DA+SA+LT+DATA), the state machine that interfaces with the TBU automatically appends zeros to the transmitting frame to make the data length exactly 46 bytes to meet the minimum data field requirement of IEEE 802.3. The MAC can be programmed not to append any padding through the MTI.

The TFC module receives the computed CRC and appends it to the data being transmitted to the TPE module. When the MAC is programmed to not append the CRC value to the end of Ethernet frames, the TFC module ignores the computed CRC and transmits only the data received from the TBU module to the TPE module. An exception to this rule is that when the MAC is programmed to append pads for frames (DA+SA+LT+DATA) less than 60 bytes sent by the TBU module, the TFC module will append the CRC at the end of padded frame.

30.5.1.2 Transmit Protocol Engine Module (TPE)

The Transmit Protocol Engine (TPE) module consists of a transmit state machine that controls the operation of Ethernet frame transmission. The module's transmit state machine performs the following functions to meet the IEEE 802.3/802.3z specifications.

- Generates preamble and SFD
- Generates jam pattern in Half-Duplex mode
- Jabber timeout
- Flow control for Half-Duplex mode (back pressure)
- Generates transmit frame status

When a new frame transmission from the TFC is requested, the transmit state machine sends out the preamble and SFD, followed by the data received. The preamble is defined as 7 bytes of 8'b10101010 pattern, and the SFD is defined as 1 byte of 8b'10101011 pattern.

The collision window is defined as 1 slot time (512 bit times for 10/100 Mbps Ethernet. The jam pattern generation is applicable only to Half-Duplex mode, not to Full-Duplex mode.

In MII mode, if a collision occurs any time from the beginning of the frame to the end of the CRC field, the transmit state machine sends a 32-bit jam pattern of 32'h55555555 on the MII to inform all other stations that a collision has occurred. If the collision is seen during the preamble transmission phase, the transmit state machine completes the transmission of preamble and SFD and then sends the jam pattern.

The TPE module maintains a jabber timer to cut off the transmission of Ethernet frames if the TFC module transfers more than 2,048 (default) bytes. The time-out

is changed to 10,240 bytes when the Jumbo frame is enabled.

The Transmit state machine uses the deferral mechanism for the flow control (Back Pressure) in Half-Duplex mode. When the Application requests to stop receiving frames, the Transmit state machine sends a JAM pattern of 32 bytes whenever it senses a reception of a frame, provided the transmit flow control is enabled. This will result in a collision and the remote station will back off. If the application requests a frame to be transmitted, then it will be scheduled and transmitted even when the backpressure is activated. Note that if the backpressure is kept activated for a long time (and more than 16 consecutive collision events occur) then the remote stations will abort their transmissions due to excessive collisions.

30.5.1.3 Transmit Scheduler Module (STX)

The Transmit Scheduler (STX) module is responsible for scheduling the frame transmission on the MII. The two major functions of this module are to maintain the inter-frame gap between two transmitted frames and to follow the Truncated Binary Exponential Back-off algorithm for Half-Duplex mode.

The STX module maintains an idle period of the configured inter-frame gap (IFG bits of Register0 between any two transmitted frames. If frames from the TFC arrive at the TPE module sooner than the configured IFG time, the TPE module waits for the enable signal from the STX module before starting the transmission on the MII. At the end of programmed IFG value, the module issues an enable signal to the TPE module in Full-Duplex mode. In Half-Duplex mode and when IFG is configured for 96 bit times, the STX module follows the rule of deference specified in Section 4.2.3.2.1 of the IEEE 802.3 specification. The module resets its IFG counter if a carrier is detected during the first two-thirds of the IFG interval. If the carrier is detected during the final one third of the IFG interval, the STX module continues the IFG count and enables the transmitter after the IFG interval.

The STX module implements the Truncated Binary Exponential Back-off algorithm when it operates in Half-Duplex mode.

30.5.1.4 Transmit CRC Generator Module (CTX)

The Transmit CRC Generator (CTX) module interfaces with the TFC module to generate CRC for the FCS field of the Ethernet frame.

This module calculates the 32-bit CRC for the FCS field of the Ethernet frame. The encoding is defined by the following generating polynomial.

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1 \quad (1)$$

The module gets the Ethernet frame's byte data from the TFC module (DA + SA + LT + DATA + PAD) qualified with a Data Valid signal.

30.5.1.5 Transmit Flow Control Module (FTX)

The Transmit Flow Control (FTX) module generates Pause frames and transmit them to the TFC module as necessary, in Full-Duplex mode. The Application can request the FTX module to send a Pause frame by setting the FCB bit in the Flow Control register Register6.

The Application requests the flow control by setting the FCB bit of Register6, the FTX module will generate and transmit a single Pause frame to the TFC module. The value of the Pause Time in the generated frame contains the programmed Pause Time value in the Register6. If the Application wants to extend the pause or end the pause prior to the time specified in the previously transmitted Pause frame, it must request another Pause frame transmission after programming the Pause Time register with appropriate value.

30.5.2 Reception

A receive operation is initiated when the MAC detects an SFD on the MII. The core strips the preamble and SFD before proceeding to process the frame. The header fields are checked for the filtering and the FCS field used to verify the CRC for the frame. The received frame is stored in a shallow buffer until the address filtering is performed. The frame is dropped in the core if it fails the address filter.

The following are the functional blocks in the Receive path of the MAC core.

- Receive Protocol Engine Module (RPE)
- Receive CRC Module (CRX)
- Receive IP Checksum checker (IPC)
- Receive Frame Controller Module (RFC)
- Receive Flow Control Module (FRX)
- Receive Bus Interface Unit Module (RBU)
- Address Filtering Module (AFM)

30.5.2.1 Receive Protocol Engine Module (RPE)

The RPE consists of the receive state machine which strips the preamble and SFD of the received Ethernet frame. Once the SFD is detected, the state machine begins sending the data of the Ethernet frame to the RFC module, beginning with the first byte following the SFD (destination address).

The receive state machine of the RPE module decodes the Length/Type field of the receiving Ethernet frame. If the Length/Type field is less than 600 (hex) and if the MAC is programmed for the auto crc/pad stripping option, the state machine sends the data of the frame up to the count specified in the Length/Type field, then starts dropping bytes (including the FCS field). The state machine of the RPE module decodes the Length/Type field and checks for the Length interpretation.

If the Length/Type field is greater than or equal to 600 (hex), the RPE module will send all received Ethernet frame data to the RFC module, irrespective of the value

on the programmed auto-CRC strip option.

As a default, the MAC is programmed for watchdog timer to be enabled, that is, frames above 2,048 (10,240 if Jumbo Frame is enabled) bytes (DA + SA + LT + DATA + PAD + FCS) are cut off at the RPE module. This feature can be disabled by programming the MAC Configuration register, Watchdog Disable. However even if the watchdog timer is disabled, frames greater than 16 KB in size are cut off and a watchdog time-out status is given.

The MAC supports loopback of transmitted frames onto its receiver. This feature can be enabled by programming the MAC Configuration register, Loopback bit. The transmit and receive clocks can have an asynchronous timing relationship, so an asynchronous FIFO is used to make the loopback path. The asynchronous FIFO is 6 bits wide to accommodate TXD, TXEN and TXER MII signals. The FIFO is nine deep and free-running to write on the write clock (Tx clock) and read on every read clock (Rx clock).

The write and read pointers gets re-initialized to have an offset of 4 at the start of each frame read out of the FIFO. This helps to avoid overflow/underflow during the transfer of a frame, and ensures that the overflow/underflow occurs only during the IFG period between the frames. Please note that the FIFO depth of nine is sufficient to prevent data corruption for frame sizes up to 9,022 bytes with a difference of 200 ppm between the MII Transmit and Receive clock frequencies. Hence, bigger frames should not be looped back, as they may get corrupted in this loopback FIFO.

30.5.2.2 Receive CRC Module (CRX)

The Receive CRC (CRX) interfaces to the RPE module to check for any CRC error in the receiving frame.

This module calculates the 32-bit CRC for the received frame that includes the Destination address field through the FCS field. The encoding is defined by the following generating polynomial.

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1 \quad (2)$$

The module gets the data from the RPE module (DA+SA+LT+DATA+PAD+FCS).

30.5.2.3 Receive IP Checksum Engine (IPC)

The application can enable IP header checksum checking and TCP/UDP checksum offload by setting the MAC Configuration register's IPC bit. This module calculates the 16-bit ones' complement of the Ethernet frame's payload data's (DATA field) ones' complement sum. The payload data is assumed to start from byte 15 (19 for a VLAN-tagged frame) of the received Ethernet frame. This module only processes IPv4 datagrams, bypassing and not processing all other types (such as IPv6).

This module also compares the calculated IP checksum with the received frame's IPv4 header checksum.

Bytes 25 and 26 of the received Ethernet frame (29 and 30 for a VLAN-tagged frame) are taken as the IP header checksum. The header checksum is calculated against the header length field (20 bytes minimum). The result of the comparison (pass or fail) is given to the RFC, which sets the appropriate bit in the receive status word. If the Header Length field value is less than 5 or if the IP Version field does not equal 4, an error is indicated for the IP header checksum.

The ones' complement sum of the IP datagram's 16-bit payload is also calculated. The start of the payload is considered to be the data after the IP header. If the data payload ends with a non-aligned halfword, then a pad byte is added for the sum calculation. The 16-bit ones' complement of the resultant sum is forwarded to the RFC module, which inserts it into the data stream (towards the application) right after the FCS bytes (MS byte first) of the Ethernet payload. This 16-bit sum helps the software check the TCP/UDP header checksums faster. Note that this 16-bit sum (which is always appended to the Ethernet frame in this mode) is invalid when the IP header checksum bit shows an Error status.

30.5.2.4 Receive Frame Controller Module (RFC)

The RFC module consists of a FIFO and two state machines for writing and reading the FIFO. The FIFO holds the received Ethernet frame data, along with a control bit to indicate the last data. The state machines manage the FIFO and provide a frame buffering for the receiving Ethernet frame from the RPE module. The main functions of the RFC module are:

- Data path conversion
- Frame filtering
- Attaching the calculated IP Checksum input from IPC.
- Update the Receive Status and forward to RBU.

If the RA bit of the MAC CSR Frame Filter register is set, the RFC module initiates the data transfer to the RBU module as soon as 4 bytes of Ethernet data are received from the RPE module. At the end of the data transfer, the RFC module sends out the received frame status that includes the frame filter bits (SA Filterfail and DAFilterfail) and status from the RFC module. These bits are generated based on the filterfail signals from the AFM module. This status bit indicates to the Application whether the received frame has passed the filter controls (both address filter and Frame Filter controls from CSR). The RFC module will not drop any frame on its own in this mode.

If the RA bit is reset, the RFC module performs frame filtering based on the destination/source address (the Application still needs to perform another level of filtering if it decides not to receive any bad frames like runt, CRC error frames, etc. The RFC module waits to receive the first 14 bytes of received data (type field) from the RPE module. Until then, the module will not initiate any transfers to the RBU module. After receiv-

ing the destination/source address bytes, the RFC checks the filter-fail signal from the AFM module for an address match. On detecting a filter-fail from AFB, the frame is dropped at the RFC module and not transferred to the Application.

30.5.2.5 Receive Flow Control Module (FRX)

The Receive Flow Controller (FRX) detects the receiving Pause frame and pauses the frame transmission for the delay specified within the received Pause frame. The FRX module is enabled only in Full-Duplex mode. The Pause frame detection function can be enabled or disabled with the RFE bit of MAC CSR Register6.

Once the receive flow control is enabled, the FRX module begins monitoring the received frame destination address for any match with the multicast address of the control frame (48'h0180C2000001). If a match is detected, the FRX module indicates to the RFC module, that the destination address of the received frame matches the reserved control frame destination address. The RFC module then decides whether or not to transfer the received control frame to the Application, based on the (PCF) bit setting of MAC CSR Register1 (Filter register).

The FRX module also decodes the Type, Op-code, and Pause Timer field of the receiving control frame. At the end of received frame, the FRX module gets the received frame status from RPE. If the byte count of the status indicates 64 bytes, and if there is no CRC error, the FRX module requests the MAC transmitter to pause the transmission of any data frame for the duration of the decoded Pause Time value, multiplied by the slot time (64 byte times). Meanwhile, if another Pause frame is detected with a zero Pause Time value, the FRX module resets the Pause Time and gives another pause request to the Transmitter. If the received control frame matches neither the Type field (16'h8808), Opcode (16'h00001), nor byte length (64 bytes), or if there is a CRC error, the FRX module does not generate a Pause request to Transmitter.

In the case of a pause frame with a multicast destination address, the RFC filters the frame based on the address match from the FRX module. For a pause frame with a unicast destination address, the filtering in the FRX module depends on whether the DA matched the contents of the MAC Address Register0 and the UP Bit of MAC Core Register6 is set (detecting a pause frame even with a unicast destination address). The PCF register bits (Bit [7:6] of MAC Register1) controls the filtering for control frames in addition to the Address filter module.

30.5.2.6 Address Filtering Module (AFM)

The Address Filtering (AFM) module performs the destination and source address checking function on all received frames. The address checking is based on different parameters (Frame Filter register) chosen by the Application.

The AFM module checks the destination and source

address field of each incoming packet.

30.5.2.6.1 Unicast Destination Address Filter

The AFM supports 1 MAC address (MacAddr0) for unicast perfect filtering. If perfect filtering is selected (HUC bit of Frame Filter register is reset), the AFM compares all 48 bits of the received unicast address with the programmed MAC address for any match.

30.5.2.6.2 Multicast Destination Address Filter

The MAC can be programmed to pass all multicast frames by setting the PM bit in the Frame Filter register.

30.5.2.6.3 Broadcast Address Filter

The AFM doesn't filter any broadcast frames in the default mode. However, if the MAC is programmed to reject all broadcast frames by setting the DBF bit in the Frame Filter register, the DAF module asserts the Filter fail signal to RFC, whenever a broadcast frame is received. This will tell the RFC module to drop the frame.

30.5.2.6.4 Inverse Filtering Operation

For both Destination and Source address filtering, there is an option to invert the filter-match result at the final output. These are controlled by the DAIF and SAIF bits of the Frame Filter register respectively. The DAIF bit is applicable for both Unicast and Multicast DA frames. The result of the unicast/multicast destination address filter is inverted in this mode. Similarly, when the SAIF bit is set, the result of unicast SA filter is reversed.

30.6 MAC MANAGEMENT COUNTERS (MMC)

The MMC module maintains a set of registers for gathering statistics on the received and transmitted frames. These include a control register for controlling the behaviour of the registers, two 32-bit registers containing interrupts generated (receive and transmit), and two 32-bit registers containing masks for the Interrupt register (receive and transmit).

The MMCs are accessed using transactions, in the same way the CSR address space is accessed. The following sections in the chapter describe the various counters and list the address for each of the statistics counters. This address will be used for Read/Write accesses to the desired transmit/receive counter.

The Receive MMC counters are updated for frames that are passed by the Address Filter (AFM) block. Statistics of frames that are dropped by the AFM module are not updated unless they are runt frames of less than 6 bytes (DA bytes are not received fully).

30.7 DESCRIPTORS

30.7.1 Descriptor Formats

The DMA in the Ethernet subsystem transfers data based on a linked list of descriptors, as explained in "DMA Controller" on page 125. The default descriptor formats (common for both Receive and Transmit Descriptors) are shown in Figure 88, and field descriptions are provided in Sections 30.7.2 and 30.7.3.

Each descriptor contains two buffers, two byte-count buffers, and two address pointers, which enable the adapter port to be compatible with various types of memory management schemes.

The descriptor addresses must be aligned to the bus width used (32-bit).

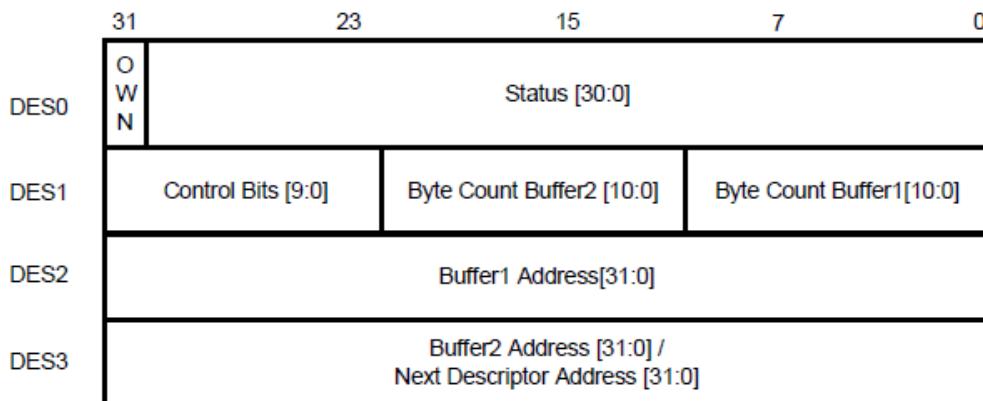


Figure 88 Rx/Tx Descriptors in Same-Endian Mode for 32-Bit, Little-Endian Format; Rx/Tx Descriptors Reverse-Endian Mode for 32-Bit, Big-Endian Format Data Bus

30.7.2 Receiver Descriptor

The EMAC Subsystem requires at least two descriptors when receiving a frame. The Receive state

machine of the DMA (in the EMAC Subsystem) always attempts to acquire an extra descriptor in anticipation of an incoming frame. (The size of the incoming frame is unknown). Before the RxDMA closes a descriptor, it

will attempt to acquire the next descriptor even if no frames are received.

In a single descriptor (receive) system, the subsystem will generate a descriptor error if the receive buffer is unable to accommodate the incoming frame and the

next descriptor is not owned by the DMA. Thus, the Host is forced to increase either its descriptor pool or the buffer size. Otherwise, the subsystem starts dropping all incoming frames.

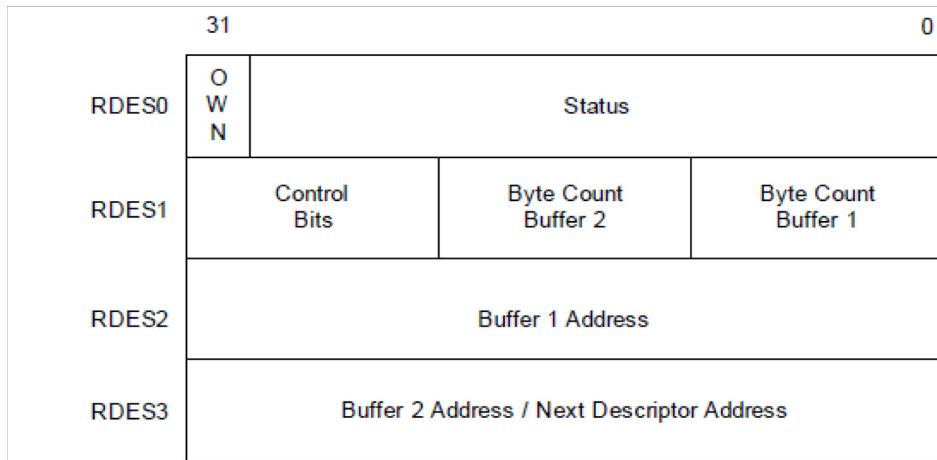


Figure 89 Receive Descriptor

30.7.2.1 Receive Descriptor 0 (RDES0)

RDES0 contains the received frame status, the frame length, and the descriptor ownership information.

Table 54: Receive Descriptor 0

Bit	Symbol	Description
31	OWN	Own Bit When set, this bit indicates that the descriptor is owned by the DMA of the EMAC Subsystem. When this bit is reset, this bit indicates that the descriptor is owned by the Host. The DMA clears this bit either when it completes the frame reception or when the buffers that are associated with this descriptor are full.
30	AFM	Destination Address Filter Fail When set, this bit indicates a frame that failed in the DA Filter in the GMAC Core.
29-16	FL	Frame Length These bits indicate the byte length of the received frame that was transferred to host memory (including CRC). This field is valid only when Last Descriptor (RDES0[8]) is set and Descriptor Error (RDES0[14]) is reset. The frame length also includes the two bytes appended to the Ethernet frame when IP checksum calculation (Type 1) is enabled and the received frame is not a MAC control frame. This field is valid when Last Descriptor (RDES0[8]) is set. When the Last Descriptor and Error Summary bits are not set, this field indicates the accumulated number of bytes that have been transferred for the current frame.
15	ES	Error Summary Indicates the logical OR of the following bits: <ul style="list-style-type: none"> • RDES0[1]: CRC Error • RDES0[3]: Receive Error • RDES0[4]: Watchdog Timeout • RDES0[6]: Late Collision • RDES0[7]: Giant Frame (This is not applicable when RDES0[7] indicates an IPV4 header Checksum error.) • RDES0[11]: Overflow Error • RDES0[14]: Descriptor Error This field is valid only when the Last Descriptor (RDES0[8]) is set.

Table 54: Receive Descriptor 0

Bit	Symbol	Description
14	DE	Descriptor Error When set, this bit indicates a frame truncation caused by a frame that does not fit within the current descriptor buffers, and that the DMA does not own the Next Descriptor. The frame is truncated. This field is valid only when the Last Descriptor (RDES0[8]) is set.
13	SAF	Source Address Filter Fail When set, this bit indicates that the SA field of frame failed the SA Filter in the EMAC Core.
12	LE	Length Error When set, this bit indicates that the actual length of the frame received and that the Length/ Type field does not match. This bit is valid only when the Frame Type (RDES0[5]) bit is reset.
11	OE	Overflow Error When set, this bit indicates that the received frame was damaged due to buffer overflow in MTL.
10	VLAN	VLAN Tag When set, this bit indicates that the frame pointed to by this descriptor is a VLAN frame tagged by the EMAC Core.
9	FS	First Descriptor When set, this bit indicates that this descriptor contains the first buffer of the frame. If the size of the first buffer is 0, the second buffer contains the beginning of the frame. If the size of the second buffer is also 0, the next Descriptor contains the beginning of the frame.
8	LS	Last Descriptor When set, this bit indicates that the buffers pointed to by this descriptor are the last buffers of the frame
7	IPC	IPC Checksum Error When set, this bit indicates that the 16-bit IPv4 Header checksum calculated by the core did not match the received checksum bytes. This error can be due to inconsistent Ethernet Type field and IP header Version field values, a header checksum mismatch in IPv4, or an Ethernet frame lacking the expected number of IP header bytes.
6	LC	Late Collision When set, this bit indicates that a late collision has occurred while receiving the frame in Half-Duplex mode.
5	FT	Frame Type When set, this bit indicates that the Receive Frame is an Ethernet-type frame (the LT field is greater than or equal to 16'h0600). When this bit is reset, it indicates that the received frame is an IEEE802.3 frame. This bit is not valid for Runt frames less than 14 bytes.
4	RWT	Receive Watchdog Timeout When set, this bit indicates that the Receive Watchdog Timer has expired while receiving the current frame and the current frame is truncated after the Watchdog Timeout.
3	RE	Receive Error When set, this bit indicates that the EMAC_RXER signal is asserted while EMAC_CRS_DV is asserted during frame reception. This error also includes carrier extension error in MII and Half-duplex mode. Error can be of less/no extension, or error (rxd != 0x0f) during extension.
2	DE	Dribble Bit Error When set, this bit indicates that the received frame has a non-integer multiple of bytes (odd nibbles). This bit is valid only in MII Mode.

Table 54: Receive Descriptor 0

Bit	Symbol	Description
1	CE	CRC Error When set, this bit indicates that a Cyclic Redundancy Check (CRC) Error occurred on the received frame. This field is valid only when the Last Descriptor (RDES0[8]) is set.
0	RMCE	Rx MAC Address/Payload Checksum Error When set, this bit indicates that the Rx MAC Address registers value (1 to 4) matched the frame's DA field. When reset, this bit indicates that the Rx MAC Address Register 0 value matched the DA field. When set, indicates the TCP, UDP, or ICMP checksum the core calculated does not match the received encapsulated TCP, UDP, or ICMP segment's Checksum field. This bit is also set when the received number of payload bytes does not match the value indicated in the Length field of the encapsulated IPv4 or IPv6 datagram in the received Ethernet frame.

The permutations of bits 5, 7, and 0 reflect the conditions discussed in Table 55.

Table 55: Receive Descriptor 0 with Checksum Offload Engine

Bit 5: Frame Type	Bit 7: IPC Checksum Error	Bit 0: Payload Checksum Error	Frame Status
0	0	0	IEEE 802.3 Type frame (Length field value is less than 0x0600.)
1	0	0	IPv4/IPv6 Type frame, no checksum error detected
1	0	1	IPv4/IPv6 Type frame with a payload checksum error (as described for PCE) detected
1	1	0	IPv4/IPv6 Type frame with an IP header checksum error (as described for IPC CE) detected
1	1	1	IPv4/IPv6 Type frame with both IP header and payload checksum errors detected
0	0	1	IPv4/IPv6 Type frame with no IP header checksum error and the payload check bypassed, due to an unsupported payload
0	1	1	A Type frame that is neither IPv4 or IPv6 (the Checksum Offload engine bypasses checksum completely.)
0	1	0	Reserved

30.7.2.2 Receive Descriptor 1 (RDES1)

RDES1 contains the buffer sizes and other bits that control the descriptor chain/ring.

Note 21: See "Buffer Size Calculations" on page 127 for further detail on calculating buffer sizes.

Table 56: Receive Descriptor 1

Bit	Symbol	Description
31	DIOC	Disable Interrupt on Completion When set, this bit will prevent the setting of the RI (CSR5[6]) bit of the Status Register for the received frame that ends in the buffer pointed to by this descriptor. This, in turn, will disable the assertion of the interrupt to Host due to RI for that frame.
30-26	-	Reserved

Table 56: Receive Descriptor 1

Bit	Symbol	Description
25	RER	Receive End of Ring When set, this bit indicates that the descriptor list reached its final descriptor. The DMA returns to the base address of the list, creating a Descriptor Ring.
24	RCH	Second Address Chained When set, this bit indicates that the second address in the descriptor is the Next Descriptor address rather than the second buffer address. When RDES1[24] is set, RBS2 (RDES1[21-11]) is a "don't care" value. RDES1[25] takes precedence over RDES1[24].
23-22	-	Reserved
21-11	RBS2	Receive Buffer 2 Size These bits indicate the second data buffer size in bytes. The buffer size must be a multiple of 4, even if the value of RDES3 (buffer2 address pointer) is not aligned to bus width. In the case where the buffer size is not a multiple of 4, the resulting behaviour is undefined. This field is not valid if RDES1[24] is set.
10-0	RBS1	Receive Buffer 1 Size Indicates the first data buffer size in bytes. The buffer size must be a multiple of 4, even if the value of RDES2 (buffer1 address pointer) is not aligned. In the case where the buffer size is not a multiple of 4, the resulting behaviour is undefined. If this field is 0, the DMA ignores this buffer and uses Buffer 2 or next descriptor depending on the value of RCH (Bit 24).

30.7.2.3 Receive Descriptor 2 (RDES2)

RDES2 contains the address pointer to the first data buffer in the descriptor.

Note 22: See "Host Data Buffer Alignment" on page 127 for further detail on buffer address alignment.

Table 57: Receive Descriptor 2

Bit	Symbol	Description
31-0	B1AP	Buffer 1 Address Pointer These bits indicate the physical address of Buffer 1. There are no limitations on the buffer address alignment except for the following condition: The DMA uses the configured value for its address generation when the RDES2 value is used to store the start of frame. Note that the DMA performs a write operation with the RDES2[1:0] bits as 0 during the transfer of the start of frame but the frame data is shifted as per the actual Buffer address pointer. The DMA ignores RDES2[1:0] if the address pointer is to a buffer where the middle or last part of the frame is stored.

30.7.2.4 Receive Descriptor 3 (RDES3)

RDES3 contains the address pointer either to the second data buffer in the descriptor or to the next descriptor.

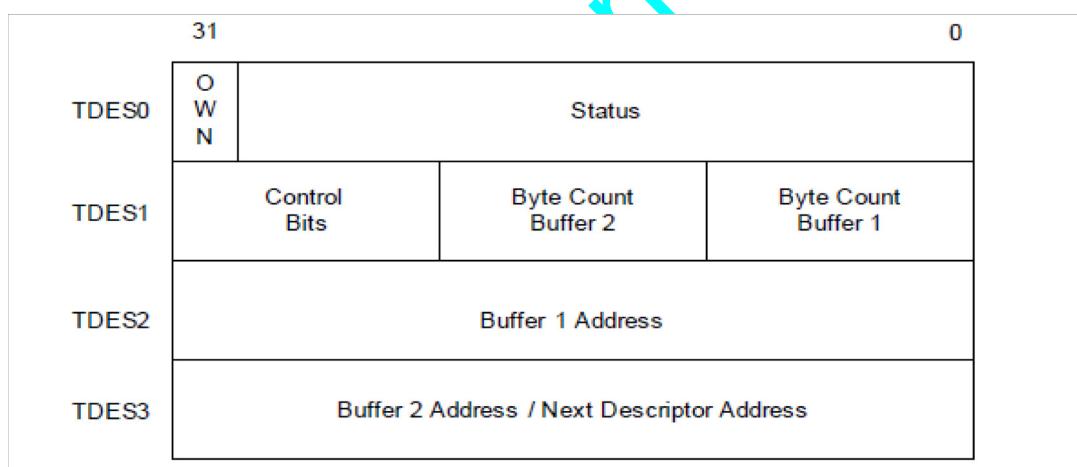
Table 58: Receive Descriptor 3

Bit	Symbol	Description
31-0	B2AP	<p>Buffer 2 Address Pointer (Next Descriptor Address)</p> <p>These bits indicate the physical address of Buffer 2 when a descriptor ring structure is used. If the Second Address Chained (RDES1[24]) bit is set, this address contains the pointer to the physical memory where the Next Descriptor is present.</p> <p>If RDES1[24] is set, the buffer (Next Descriptor) address pointer must be bus width-aligned (RDES3[1:0] = 0. LSBs are ignored internally.) However, when RDES1[24] is reset, there are no limitations on the RDES3 value, except for the following condition: The DMA uses the configured value for its buffer address generation when the RDES3 value is used to store the start of frame. The DMA ignores RDES3[1:0] if the address pointer is to a buffer where the middle or last part of the frame is stored.</p>

30.7.3 Transmit Descriptor

The descriptor addresses must be aligned to the bus width (32-bit). Figure 90 shows the transmit descriptor format.

Each descriptor is provided with two buffers, two byte-count buffers, and two address pointers, which enable the adapter port to be compatible with various types of memory-management schemes.


Figure 90 Transmit Descriptor Format

30.7.3.1 Transmit Descriptor 0 (TDES0)

TDES0 contains the transmitted frame status and the descriptor ownership information.

Table 59: Transmit Descriptor 0

Bit	Symbol	Description
31	OWN	Own Bit When set, this bit indicates that the descriptor is owned by the DMA. When this bit is reset, this bit indicates that the descriptor is owned by the Host. The DMA clears this bit either when it completes the frame transmission or when the buffers allocated in the descriptor are empty. The ownership bit of the First Descriptor of the frame should be set after all subsequent descriptors belonging to the same frame have been set. This avoids a possible race condition between fetching a descriptor and the driver setting an ownership bit.
30-18	-	Reserved
17	-	Reserved
16	IHE	IP Header Error When set, this bit indicates that the Checksum Offload engine detected an IP header error and consequently did not modify the transmitted frame for any checksum insertion. This bit provides the result of the Full Checksum Offload mechanism
15	ES	Error Summary Indicates the logical OR of the following bits: <ul style="list-style-type: none">• TDES0[14]: Jabber Timeout• TDES0[13]: Frame Flush• TDES0[11]: Loss of Carrier• TDES0[10]: No Carrier• TDES0[9]: Late Collision• TDES0[8]: Excessive Collision• TDES0[2]: Excessive Deferral• TDES0[1]: Underflow Error
14	JT	Jabber Timeout When set, this bit indicates the EMAC transmitter has experienced a jabber timeout. This bit is only set when the EMAC configuration register's JD bit is not set.
13	FF	Frame Flushed When set, this bit indicates that the DMA/MTL flushed the frame due to a SW flush command given by the CPU.
12	PCE	Payload Checksum Error This bit, when set, indicates that the Checksum Offload engine had a failure and did not insert any checksum into the encapsulated TCP, UDP, or ICMP payload. This failure can be either due to insufficient bytes, as indicated by the IP Header's Payload Length field, or the MTL starting to forward the frame to the MAC transmitter in Store-and-Forward mode without the checksum having been calculated yet. This second error condition only occurs when the Transmit FIFO depth is less than the length of the Ethernet frame being transmitted: to avoid deadlock, the MTL starts forwarding the frame when the FIFO is full, even in Store-and-Forward mode. When the Full Checksum Offload engine is not enabled during configuration, this bit is reserved.
11	LC	Loss of Carrier When set, this bit indicates that Loss of Carrier occurred during frame transmission (that is, the EMAC_CRS signal was inactive for one or more transmit clock periods during frame transmission). This is valid only for the frames transmitted without collision and when the EMAC operates in Half-Duplex Mode.
10	NC	No Carrier When set, this bit indicates that the carrier sense signal from the PHY was not asserted during transmission.
9	LC	Late Collision When set, this bit indicates that frame transmission was aborted due to a collision occurring after the collision window (64 byte times including Preamble in MII Mode and 512 byte times including Preamble and Carrier Extension in MII Mode). Not valid if Underflow Error is set.

Table 59: Transmit Descriptor 0

Bit	Symbol	Description
8	EC	Excessive Collision When set, this bit indicates that the transmission was aborted after 16 successive collisions while attempting to transmit the current frame. If the DR (Disable Retry) bit in the EMAC Configuration Register is set, this bit is set after the first collision and the transmission of the frame is aborted.
7	VF	VLAN Frame When set, this bit indicates that the transmitted frame was a VLAN-type frame.
6-3	CC	Collision Count This 4-bit counter value indicates the number of collisions occurring before the frame was transmitted. The count is not valid when the Excessive Collisions bit (TDES0[8]) is set.
2	ED	Excessive Deferral When set, this bit indicates that the transmission has ended because of excessive deferral of over 24,288 bit times (155,680 bits times in Jumbo Frame enabled mode) if the Deferral Check (DC) bit is set high in the EMAC Control Register.
1	UF	Underflow Error When set, this bit indicates that the EMAC aborted the frame because data arrived late from the Host memory. Underflow Error indicates that the DMA encountered an empty Transmit Buffer while transmitting the frame. The transmission process enters the suspended state and sets both Transmit Underflow (Register 5[5]) and Transmit Interrupt (Register 5[0]).
0	DB	Deferred Bit When set, this bit indicates that the EMAC defers before transmission because of the presence of carrier. This bit is valid only in Half-Duplex mode.

30.7.3.2 Transmit Descriptor 1(TDES1)

TDES1 contains the buffer sizes and other bits which control the descriptor chain/ring and the frame being transferred.

Note 23: See "Buffer Size Calculations" on page 127 for further detail on calculating buffer sizes.

Table 60: Transmit Descriptor 1

Bit	Symbol	Description
31	IC	Interrupt on Completion When set, this bit sets Transmit Interrupt (Register 5[0]) after the present frame has been transmitted.
30	LS	Last Segment When set, this bit indicates that the buffer contains the last segment of the frame.
29	FS	First Segment When set, this bit indicates that the buffer contains the first segment of a frame.

Table 60: Transmit Descriptor 1

Bit	Symbol	Description
28-27	CIC	<p>Checksum Insertion Control These bits control the insertion of checksums in Ethernet frames that encapsulate TCP, UDP, or ICMP over IPv4 or IPv6 as described below.</p> <ul style="list-style-type: none"> • 2'b00: Do nothing. Checksum Engine is bypassed • 2'b01: Insert IPv4 header checksum. Use this value to insert IPv4 header checksum when the frame encapsulates an IPv4 datagram. • 2'b10: Insert TCP/UDP/ICMP checksum. The checksum is calculated over the TCP, UDP, or ICMP segment only and the TCP, UDP, or ICMP pseudo-header checksum is assumed to be present in the corresponding input frame's Checksum field. An IPv4 header checksum is also inserted if the encapsulated datagram conforms to IPv4. • 2'b11: Insert a TCP/UDP/ICMP checksum that is fully calculated in this engine. In other words, the TCP, UDP, or ICMP pseudo-header is included in the checksum calculation, and the input frame's corresponding Checksum field has an all-zero value. An IPv4 Header checksum is also inserted if the encapsulated datagram conforms to IPv4. <p>The Checksum engine detects whether the TCP, UDP, or ICMP segment is encapsulated in IPv4 or IPv6 and processes its data accordingly.</p>
26	DC	<p>Disable CRC When set, the EMAC does not append the Cyclic Redundancy Check (CRC) to the end of the transmitted frame. This is valid only when the first segment (TDES1[29]).</p>
25	TER	<p>Transmit End of Ring When set, this bit indicates that the descriptor list reached its final descriptor. The returns to the base address of the list, creating a descriptor ring.</p>
24	TCH	<p>Second Address Chained When set, this bit indicates that the second address in the descriptor is the Next Descriptor address rather than the second buffer address. When TDES1[24] is set, TBS2 (TDES1[21–11]) are “don't care” values. TDES1[25] takes precedence over TDES1[24].</p>
23	DP	<p>Disable Padding When set, the EMAC does not automatically add padding to a frame shorter than 64 bytes. When this bit is reset, the DMA automatically adds padding and CRC to a frame shorter than 64 bytes and the CRC field is added despite the state of the DC (TDES1[26]) bit. This is valid only when the first segment (TDES1[29]) is set.</p>
22	-	Reserved
21-11	TBS2	<p>Transmit Buffer 2 Size These bits indicate the Second Data Buffer in bytes. This field is not valid if TDES1[24] is set.</p>
10:0	TBS1	<p>Transmit Buffer 1 Size These bits indicate the First Data Buffer byte size. If this field is 0, the DMA ignores this buffer and uses Buffer 2 or next descriptor depending on the value of TCH (Bit 24).</p>

30.7.3.3 Transmit Descriptor 2 (TDES2)

TDES2 contains the address pointer to the first buffer of the descriptor.

Table 61: Transmit Descriptor 2

Bit	Symbol	Description
31-0-	B1AP	<p>Buffer 1 Address Pointer These bits indicate the physical address of Buffer 1. There is no limitation on the buffer address alignment. See "Host Data Buffer Alignment" on page 127 for further detail on buffer address alignment.</p>

30.7.3.4 Transmit Descriptor 3 (TDES3)

TDES3 contains the address pointer either to the second buffer of the descriptor or the next descriptor.

Table 62: Transmit Descriptor 3

Bit	Symbol	Description
31-0-	B2AP	Buffer 2 Address Pointer (Next Descriptor Address) Indicates the physical address of Buffer 2 when a descriptor ring structure is used. If the Second Address Chained (TDES1[24]) bit is set, this address contains the pointer to the physical memory where the Next Descriptor is present. The buffer address pointer must be aligned to the bus width only when TDES1[24] is set. (LSBs are ignored internally.)

30.8 STATION MANAGEMENT AGENT (SMA)

The Station Management Agent (SMA) module allows the Application to access any PHY registers through a 2-wire Station Management interface (MIM). The interface supports accessing up to 32 PHYs.

The Application can select one of the 32 PHYs and one of the 32 registers within any PHY and send control data or receive status information. Only one register in one PHY can be addressed at any given time. For

more details on the communication from the Application to the PHYs, refer to the Reconciliation Sublayer and Media Independent Interface Specifications section of the IEEE 802.3z specification, 1000BASE Ethernet. The Application sends the control data to the PHY and receives status information from the PHY through the SMA module, as shown in Figure 91.

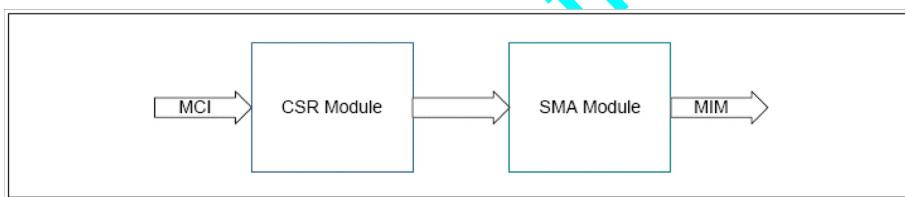


Figure 91 SMA Interface Block

30.8.1 Functions

The MAC initiates the Management Write/Read operation. The MDC clock pin is the System clock (AHB clock) divided by 42. The frame structure on the MDIO line is shown below.

- IDLE: The MDIO line is three-state; there is no clock on MDC pin
- PREAMBLE: 32 continuous bits of value 1
- START: Start-of-frame is 2'01
- OPCODE: 2'b10 for Read and 2'b01 for Write
- PHY ADDR: 5-bit address select for one of 32 PHYs
- REG ADDR: Register address in the selected PHY
- TA: Turnaround is 2'bZ0 for Read and 2'b10 for Write
- DATA: Any 16-bit value. In a Write operation, the MAC drives MDIO; in a Read operation, PHY drives it.

30.8.2 MII Management Write Operation

When the user sets the MII Write and Busy bits (see MII Address Register, "Register 4 (MII Address Register")¹), the MAC CSR module transfers the PHY

address, the register address in PHY, and the write data (MII Data Register) to the SMA to initiate a Write operation into the PHY registers. At this point, the SMA module starts a Write operation on the MII Management Interface using the Management Frame Format specified in the MII specifications (Section 22.2.4.5 of IEEE Standard). The application should not change the MII Address register contents or the MII Data register while the transaction is ongoing. Write operations to the MII Address register or the MII Data Register during this period are ignored (the Busy bit is high), and the transaction is completed without any error on the MCI interface.

After the Write operation has completed, the SMA indicates this to the CSR which then resets the Busy bit. The SMA module divides the CSR (Application) clock with the clock divider programmed (CR bits of MII Address Register) to generate the MDC clock for this interface. The MAC drives the MDIO line for the complete duration of the frame. The frame format for the

Write operation is described in Figure 92. .

IDLE	PREAMBLE	START	OPCODE	PHY ADDR	REG ADDR	TA	DATA	IDLE
Z	1111...11	01	01	AAAAAA	RRRRR	10	DDD ...DDD	Z

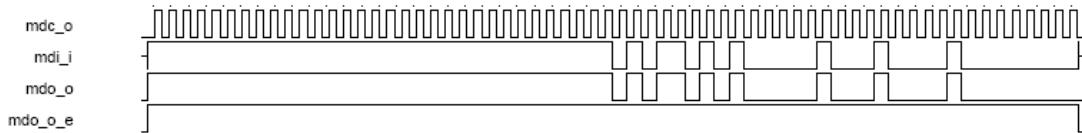


Figure 92 MII Write operation

30.8.3 MII Management Read Operation

When the user sets the MII Busy bit (see MII Address Register, “Register 4 (MII Address Register)”) with the MII Write bit as 0, the MAC CSR module transfers the PHY address and the register address in PHY to the SMA to initiate a Read operation in the PHY registers. At this point, the SMA module starts a Read operation on the MII Management Interface using the Management Frame Format specified in the MII specifications (Section 22.2.4.5 of IEEE Standard). The application should not change the MII Address register contents or the MII Data register while the transaction is ongoing. Write operations to the MII Address register or MII Data Register during this period are ignored (the Busy bit is

high) and the transaction completed without any error on the MCI interface.

After the Read operation has completed, the SMA indicates this to the CSR, which then resets the Busy bit and updates the MII Data register with the data read from the PHY. The SMA module divides the CSR (Application) clock with the clock divider programmed (CR bits of MII Address Register) to generate the MDC clock for this interface. The MAC drives the MDIO line for the complete duration of the frame except during the Data fields when the PHY is driving the MDIO line. The frame format for the Read operation is as described in Figure 93.

IDLE	PREAMBLE	START	OPCODE	PHY ADDR	REG ADDR	TA	DATA	IDLE
Z	1111...11	01	10	AAAAAA	RRRRR	Z0	DDD ...DDD	Z

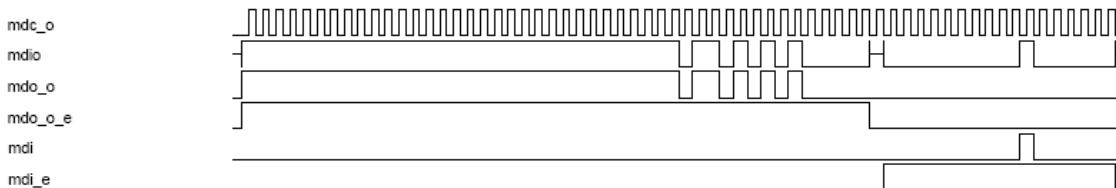


Figure 93 MII Read operation

30.9 REDUCED MEDIA INDEPENDENT INTERFACE (RMII)

The Reduced Media Independent Interface (RMII) specification reduces the pin count between Ethernet PHYs and Switch ASICs (only in 10/100 mode). According to the IEEE 802.3u standard, an MII contains 16 pins for data and control. In devices incorpo-

rating multiple MAC or PHY interfaces (such as switches), the number of pins adds significant cost with increase in port count. The RMII specification addresses this problem by reducing the pin count to 7 for each port — a 62.5% decrease in pin count.

The RMII block is placed in front of the Ethernet MAC to translate the MII signals to RMII signals.

30.9.1 Transmit/Receive Bit Ordering

Each nibble from the MII must be transmitted on the

RMII and vice versa a di-bit at a time with the order of di-bit transmission/reception shown in Figure 94 and Figure 95.

The lower order bits (D1 and D0) are transmitted first followed by higher order bits (D2 and D3). The lower order bits (D0 and D1) are received first, followed by the higher order bits (D2 and D3).

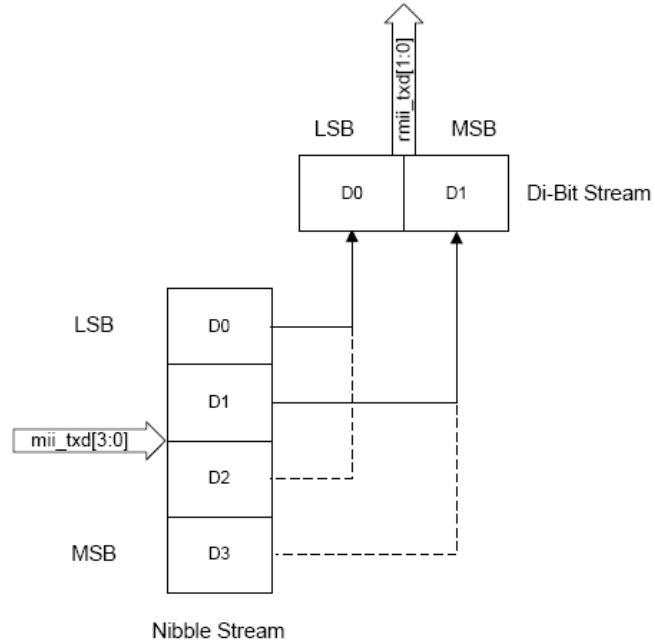


Figure 94 RMII Transmission Bit Ordering

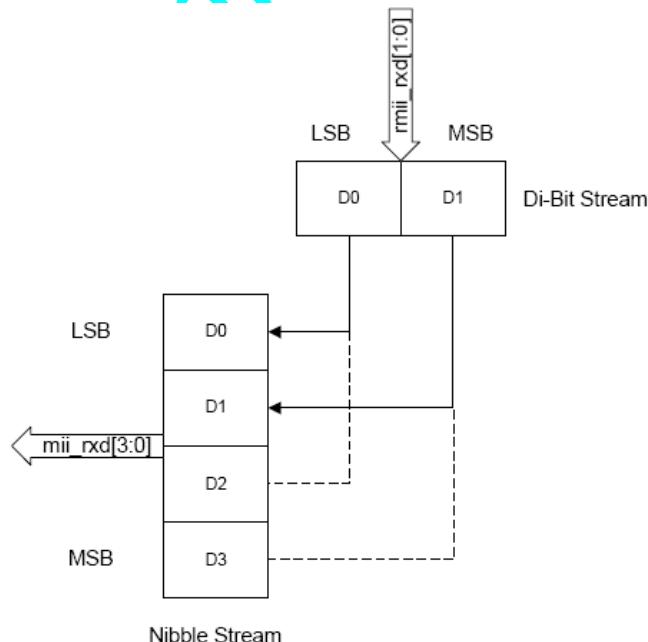


Figure 95 RMII Reception Bit Ordering

30.10 INTERRUPTS FROM THE MAC CORE

Interrupts can be generated from the MAC core as a

result of various events in the modules in it. The Interrupt Status register in the MAC Register Map describes the events that can cause an interrupt from the MAC

core. Each event can be prevented from asserting by setting the corresponding mask bits in the Interrupt Mask register.

The interrupt register bits only indicate the block from which the event is reported. You must read the corresponding status registers and other registers to clear the interrupt.

30.11 DESCRIPTION OF THE REGISTER FILES

30.11.1 MAC Register File

30.11.1.1 EMAC_MACR0_CONFIG_REG

The MAC Configuration register establishes receive and transmit operating modes.

Refer to Table 263 for the detailed bit descriptions.

30.11.1.2 EMAC_MACR1_FRAME_FILTER_REG

The MAC Frame Filter register contains the filter controls for receiving frames. Some of the controls from this register go to the address check block of the MAC, which performs the first level of address filtering. The second level of filtering is performed on the incoming frame, based on other controls such as Pass Bad Frames and Pass Control Frames.

Refer to Table 264 for the detailed bit descriptions.

30.11.1.3 EMAC_MACR4_MII_ADDR_REG

The MII Address register controls the management cycles to the external PHY through the management interface.

Refer to Table 265 for the detailed bit descriptions.

30.11.1.4 EMAC_MACR5_MII_DATA_REG

The MII Data register stores Write data to be written to the PHY register located at the address specified in Register 4. Register 5 also stores Read data from the PHY register located at the address specified by Register 4.

Refer to Table 266 for the detailed bit descriptions.

30.11.1.5 EMAC_MACR6_FLOW_CTRL_REG

The Flow Control register controls the generation and reception of the Control (Pause Command) frames by the MAC's Flow control module. A Write to a register with the Busy bit set to '1' triggers the Flow Control block to generate a Pause Control frame. The fields of the control frame are selected as specified in the 802.3x specification, and the Pause Time value from this register is used in the Pause Time field of the control frame. The Busy bit remains set until the control frame is transferred onto the cable. The Host must make sure that the Busy bit is cleared before writing to the register.

Refer to Table 267 for the detailed bit descriptions.

30.11.1.6 EMAC_MACR7_VLAN_TAG_REG

The VLAN Tag register contains the IEEE 802.1Q VLAN Tag to identify the VLAN frames. The MAC compares the 13th and 14th bytes of the receiving frame (Length/Type) with 16'h8100, and the following 2 bytes are compared with the VLAN tag; if a match occurs, it sets the received VLAN bit in the receive frame status. The legal length of the frame is increased from 1518 bytes to 1522 bytes.

If the VLAN Tag register is configured to be double-synchronized to the MII clock domain, then consecutive writes to these register should be performed only after at least 4 clock cycles in the destination clock domain.

Refer to Table 268 for the detailed bit descriptions.

30.11.1.7 EMAC_MACR8_CORE_VER_REG

The Version register's contents identify the version of the core. This register contains two bytes.

Refer to Table 269 for the detailed bit descriptions.

30.11.1.8 EMAC_MACR14_INT_REG

The Interrupt Status register contents identify the events in the MAC core that can generate interrupt.

Refer to Table 270 for the detailed bit descriptions.

30.11.1.9 EMAC_MACR15_INT_MSK_REG

The Interrupt Mask Register bits enables the user to mask the interrupt signal due to the corresponding event in the Interrupt Status Register.

Refer to Table 271 for the detailed bit descriptions.

30.11.1.10 EMAC_MACR16_MAC_ADDR0_HIGH_REG

The MAC Address0 High register holds the upper 16 bits of the 6-byte first MAC address of the station. Note that the first DA byte that is received on the MII interface corresponds to the LS Byte (Bits [7:0]) of the MAC Address Low register. For example, if 0x112233445566 is received (0x11 is the first byte) on the MII as the destination address, then the MacAddress0 Register [47:0] is compared with 0x665544332211.

Please note that consecutive writes to this Address Low Register should be performed only after at least 4 clock cycles in the destination clock domain for proper synchronization updates.

Refer to Table 272 for the detailed bit descriptions.

30.11.1.11 EMAC_MACR17_MAC_ADDR0_LOW_REG

The MAC Address0 Low register holds the lower 32 bits of the 6-byte first MAC address of the station.

Refer to Table 273 for the detailed bit descriptions.

30.11.1.12 EMAC_MACR18_MAC_ADDR1_HIGH_REG, EMAC_MACR20_MAC_ADDR2_HIGH_REG, EMAC_MACR22_MAC_ADDR3_HIGH_REG,

EMAC_MACR24_MAC_ADDR4_HIGH_REG

The MAC Address1 to Address4 High Registers hold the upper 16 bits of corresponding the 6-byte MAC address of the station.

Refer to Table 274 and Note 66, Note 68 for the detailed bit descriptions.

30.11.1.13 EMAC_MACR19_MAC_ADDR1_LOW_REG G, EMAC_MACR21_MAC_ADDR2_LOW_REG, EMAC_MACR23_MAC_ADDR3_LOW_REG, EMAC_MACR25_MAC_ADDR4_LOW_REG

The MAC Address1 to Address4 Low Registers hold the lower 32 bits of the corresponding 6-byte MAC address of the station.

Refer to Table 275 and Note 67, Note 68 for the detailed bit descriptions.

30.11.1.14 EMAC_MACR54_MII_STATUS_REG

The MII/RMII Status register indicates the status signals received by the MII/RMII (whichever is selected at HW or SW reset) from the PHY.

Refer to Table 276 for the detailed bit descriptions.

30.11.2 MMC Register File

30.11.2.1 EMAC_MMCCNTROL_REG

The MMC Control register establishes the operating mode of the management counters.

Refer to Table 277 for the detailed bit descriptions.

30.11.2.2 EMAC_MMCRXREG

The MMC Receive Interrupt register maintains the interrupts generated when receive statistic counters reach half their maximum values. (MSB of the counter is set.) It is a 32-bit wide register. An interrupt bit is cleared when the respective MMC counter that caused the interrupt is read. The least significant byte lane (bits[7:0]) of the respective counter must be read in order to clear the interrupt bit.

Refer to Table 278 for the detailed bit descriptions.

30.11.2.3 EMAC_MMCTRREG

The MMC Transmit Interrupt register maintains the interrupts generated when transmit statistic counters reach half their maximum values. (MSB of the counter is set.) It is a 32-bit wide register. An interrupt bit is cleared when the respective MMC counter that caused the interrupt is read. The least significant byte lane (bits[7:0]) of the respective counter must be read in order to clear the interrupt bit.

Refer to Table 279 for the detailed bit descriptions.

30.11.2.4 EMAC_MMCRXMASKREG

The MMC Receive Interrupt Mask register maintains the masks for the interrupts generated when receive

statistic counters reach half their maximum value. (MSB of the counter is set.) It is a 32-bit wide register.

Refer to Table 280 for the detailed bit descriptions.

30.11.2.5 EMAC_MMCTXMASKTXREG

The MMC Transmit Interrupt Mask register maintains the masks for the interrupts generated when transmit statistic counters reach half their maximum value. (MSB of the counter is set). It is a 32-bit wide register.

Refer to Table 281 for the detailed bit descriptions.

30.11.2.6 EMAC_MMCTXUNICASTFRAMESGBREG

Number of good and bad unicast frames transmitted.

Refer to Table 282 for the detailed bit descriptions.

30.11.2.7 EMAC_MMCTXUNDERFLOWERRORREG

Number of frames aborted due to frame underflow error.

Refer to Table 283 for the detailed bit descriptions.

30.11.2.8 EMAC_MMCTXSINGLECOLREG

Number of successfully transmitted frames after a single collision in Half-duplex mode.

Refer to Table 284 for the detailed bit descriptions.

30.11.2.9 EMAC_MMCTXMULTICOLREG

Number of successfully transmitted frames after more than a single collision in Half-duplex mode.

Refer to Table 285 for the detailed bit descriptions.

30.11.2.10 EMAC_MMCTXLATECOLREG

Number of frames aborted due to late collision error.

Refer to Table 286 for the detailed bit descriptions.

30.11.2.11 EMAC_MMTCARRIERERRORREG

Number of frames aborted due to carrier sense error (no carrier or loss of carrier).

Refer to Table 287 for the detailed bit descriptions.

30.11.2.12 EMAC_MMCTXOCTETCOUNTREG

Number of bytes transmitted, exclusive of preamble, in good frames only.

Refer to Table 288 for the detailed bit descriptions.

30.11.2.13 EMAC_MMCTXFRAMECOUNTREG

Number of good frames transmitted.

Refer to Table 289 for the detailed bit descriptions.

30.11.2.14 EMAC_MMCTXVLANFRAMESREG

Number of good VLAN frames transmitted, exclusive of

retried frames.

Refer to Table 290 for the detailed bit descriptions.

30.11.2.15 EMAC_MMCRXFRAMECOUNT_GB_REG

Number of good and bad frames received.

Refer to Table 291 for the detailed bit descriptions.

30.11.2.16 EMAC_MMCRXOCTETCOUNT_G_REG

Number of bytes received, exclusive of preamble, in good and bad frames.

Refer to Table 292 for the detailed bit descriptions.

30.11.2.17 EMAC_MMCRXBROADCASTFRAMES_G_REG

Number of good broadcast frames received.

Refer to Table 293 for the detailed bit descriptions.

30.11.2.18 EMAC_MMCRXMULTICASTFRAMES_G_REG

Number of good multicast frames received.

Refer to Table 294 for the detailed bit descriptions.

30.11.2.19 EMAC_MMCRXJABBERERROR_REG

Number of giant frames received with length (including CRC) greater than 1,518 bytes (1,522 bytes for VLAN tagged) and with CRC error. If Jumbo Frame mode is enabled, then frames of length greater than 9,018 bytes (9,022 for VLAN tagged) are considered as giant frames.

Refer to Table 295 for the detailed bit descriptions.

30.11.2.20 EMAC_MMCRXUNDERSIZE_G_REG

Number of frames received with length less than 64 bytes, without any errors.

Refer to Table 296 for the detailed bit descriptions.

30.11.2.21 EMAC_MMCRXOVERSIZE_G_REG

Number of frames received with length greater than the maxsize (1,518 or 1,522 for VLAN tagged frames), without errors.

Refer to Table 297 for the detailed bit descriptions.

30.11.2.22 EMAC_MMCRXUNICASTFRAMES_G_REG

Number of good unicast frames received.

Refer to Table 298 for the detailed bit descriptions.

30.11.2.23 EMAC_MMCRXFIFO_OVERFLOW_REG

Number of missed received frames due to FIFO overflow.

Refer to Table 299 for the detailed bit descriptions.

30.11.2.24 EMAC_MMCRXVLANFRAMES_GB_REG

Number of good and bad VLAN frames received.

Refer to Table 300 for the detailed bit descriptions.

30.11.3 DMA Register File

30.11.3.1 EMAC_DMAR0_BUS_MODE_REG

The Bus Mode register establishes the bus operating modes for the DMA.

Refer to Table 301 for the detailed bit descriptions.

30.11.3.2 EMAC_DMAR1_TX_POLL_DEMAND_REG

The Transmit Poll Demand register enables the Transmit DMA to check whether or not the current descriptor is owned by DMA. The Transmit Poll Demand command is given to wake up the TxDMA if it is in Suspend mode. The TxDMA can go into Suspend mode due to an Underflow error in a transmitted frame or due to the unavailability of descriptors owned by Transmit DMA. You can give this command anytime and the TxDMA will reset this command once it starts re-fetching the current descriptor from host memory.

Refer to Table 302 for the detailed bit descriptions.

30.11.3.3 EMAC_DMAR2_RX_POLL_DEMAND_REG

The Receive Poll Demand register enables the receive DMA to check for new descriptors. This command is given to wake up the RxDMA from SUSPEND state. The RxDMA can go into SUSPEND state only due to the unavailability of descriptors owned by it.

Refer to Table 303 for the detailed bit descriptions.

30.11.3.4 EMAC_DMAR3_RX_DESCRIPTOR_LIST_ADDR_REG

The Receive Descriptor List Address register points to the start of the Receive Descriptor List. The descriptor lists reside in the host's physical memory space and must be word aligned. The DMA internally converts it to bus width aligned address by making the corresponding LS bits low. Writing to Register 3 is permitted only when reception is stopped. When stopped, Register 3 must be written to before the receive Start command is given.

Refer to Table 304 for the detailed bit descriptions.

30.11.3.5 EMAC_DMAR4_TX_DESCRIPTOR_LIST_ADDR_REG

The Transmit Descriptor List Address register points to the start of the Transmit Descriptor List. The descriptor lists reside in the host's physical memory space and must be word aligned. The DMA internally converts it to bus width aligned address by making the corresponding LSB to low. Writing to Register 4 is permitted only when transmission has stopped. When stopped, Register 4 can be written before the transmission Start command is given.

Refer to Table 305 for the detailed bit descriptions.

30.11.3.6 EMAC_DMAR5_STATUS_REG

The Status register contains all the status bits that the DMA reports to the host and is usually read by the Software driver during an interrupt service routine or polling. Most of the fields in this register cause the host to be interrupted. Register 5 bits are not cleared when read. Writing 1'b1 to (unreserved) bits in Register5[16:0] clears them and writing 1'b0 has no effect. Each field (bits[16:0]) can be masked by masking the appropriate bit in Register 7.

Refer to Table 306 for the detailed bit descriptions.

30.11.3.7 EMAC_DMAR6_OPERATION_MODE_REG

The Operation Mode register establishes the Transmit and Receive operating modes and commands. Regis-

ter 6 should be the last CSR to be written as part of DMA initialization.

Refer to Table 307 for the detailed bit descriptions.

30.11.3.8 EMAC_DMAR7_INT_ENABLE_REG

The Interrupt Enable register enables the interrupts reported by Register 5. Setting a bit to 1'b1 enables a corresponding interrupt. After a hardware or software reset, all interrupts are disabled.

The interrupt is generated as shown in Figure 96. It is asserted only when the TTI, GPI, GMI, or GLI bits of the DMA Status register is asserted, or when the NIS/AIS Status bit is asserted and the corresponding Interrupt Enable bits (NIE/AIE) are enabled.

Refer to Table 308 for the detailed bit descriptions.

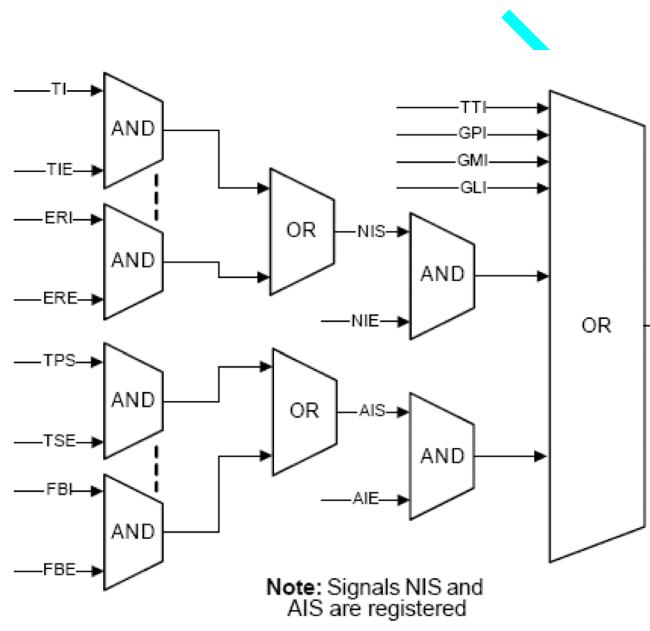


Figure 96 EMAC DMA Interrupt Generation

30.11.3.9 EMAC_DMAR8_MISSFRAME_BUFOVRCNT_REG

The DMA maintains two counters to track the number of missed frames during reception. This register reports the current value of the counter. The counter is used for diagnostic purposes.

Bits[15:0] indicate missed frames due to the host buffer being unavailable.

Bits[27:17] indicate missed frames due to buffer overflow conditions (MTL and MAC) and runt frames (good frames of less than 64 bytes) dropped by the MTL.

Refer to Table 309 for the detailed bit descriptions.

30.11.3.10 EMAC_DMAR18_CUR_HOST_TX_DESC

RIPTOR_REG

The Current Host Transmit Descriptor register points to the start address of the current Transmit Descriptor read by the DMA.

Refer to Table 310 for the detailed bit descriptions.

30.11.3.11 EMAC_DMAR19_CUR_HOST_RX_DESCRIPTOR_REG

The Current Host Receive Descriptor register points to the start address of the current Receive Descriptor read by the DMA.

Refer to Table 311 for the detailed bit descriptions.

30.11.3.12 EMAC_DMAR20_CUR_HOST_TX_BUF_

ADDR_REG

The Current Host Transmit Buffer Address register points to the current Transmit Buffer Address being read by the DMA.

Refer to Table 312 for the detailed bit descriptions.

30.11.3.13 EMAC_DMAR21_CUR_HOST_RX_BUF_ADDR_REG

The Current Host Receive Buffer Address register points to the current Receive Buffer address being read by the DMA.

Refer to Table 313 for the detailed bit descriptions.

30.12 ACCESSING EMAC FIFOS THROUGH THE AHB BUS (FOR TESTING ONLY)

In the Ethernet MAC there are two DPRAMs, one Rx FIFO 32x35 and one Tx FIFO 64x35. For testing purposes these memories can be accessed through the system bus.

The memories are mapped into the address ranges starting from EMAC_RXFIFO_LOW_START, EMAC_RXFIFO_HIGH_START, EMAC_TXFIFO_LOW_START and EMAC_TXFIFO_HIGH_START (refer to Table 71).

Before accessing those memories through the System Bus the memories must use only the System Bus clock. This can be configured by setting to "1" the "SW_DRAMODE" bit of CLK_GLOBAL_REG.

30.12.1 Write access

Since the memories are 35-bits wide the user must store first the 3 MSBs (bits 34 to 32) into a internal to EMAC temporary register by using the EMAC_(R/T)XFIFO_HIGH range. Then another write access should be performed to the EMAC_(R/T)XFIFO_LOW range in order to provide the 32 LSBs and finalize the write access.

The MSBs 35 to 32 are mapped into the bits 2 to 0 of the AHB data bus.

There is one temporary write-register for the Rx and one for the Tx FIFO. The temporary register is write only through the System Bus and ignores the address bits. Any read access from the EMAC_(R/T)XFIFO_HIGH ranges will access the corresponding FIFO and not the internal temporary register.

30.12.2 Read access

In order to fetch all 35 bits the user must perform two read access (with any order) to the proper address ranges (EMAC_(R/T)XFIFO_HIGH and EMAC_(R/T)XFIFO_LOW).

There is a limitation where there should not be any back to back (with no system clock space between them) of write and then read access at the same FIFO.

31.0 Crypto

31.1 GENERAL DESCRIPTION

The Crypto aims to accelerate the algorithms calculation that we need in order to implement the RFC4835.

The Crypto uses a DMA engine for transferring encrypted/decrypted data to a sheared memory in the AHB bus. The control registers of the IP are connected to the AHB bus.

Note 24: No DMA access should be performed at the non-shared RAM range, i.e. at the range of 0x0000.0000 up to 0x0000.FFFF

31.2 FEATURES

- AES (Advanced Encryption Standard) with 128, 192 or 256 bits key cryptographic algorithm.
- DES (Data Encryption Standard) cryptographic algorithm
- Triple DES cryptographic algorithm
- ECB (Electronic Code Book) mode of operation
- CBC (Cipher Block Chaining) mode of operation
- CTR (Counter) mode of operation.
- AHB Master DMA machine for data manipulation.
- AHB Slave register file for configuration.

31.3 DESCRIPTION

The Crypto IP aims to accelerate the algorithms calculation that we need in order to implement the RFC4835. The requirements, in terms of the various algorithms, of the RFC4835 are listed below:

- Encryption algorithms:

Table 63: RFC4835 Encryption requirements

Requirement	Encryption Algorithm
MUST	RFC3602 - AES-CBC 128-bits key
MUST	RFC2451- Triple DES-CBC
SHOULD	RFC3686 - AES-CTR

- Authentication algorithms:

Table 64: RFC4835 Authentication requirements

Requirement	Authentication Algorithm
MUST	RFC2404 - HMAC-SHA1-96
SHOULD+	RFC3566 [5] AES-XCBC-MAC-96
MAY	RFC2403 [1] HMAC-MD5-96

In order to achieve minimum die area, the RFC4835 is not implemented fully by the hardware IP, thus we need extra software in order to achieve full implementation of the RFC4835. The algorithms that the Crypto implements in hardware are:

- AES (128,192,256 key size)
- DES
- TripleDES.

The above algorithms can be used in the following modes of operation:

- ECB (Electronic Code Book)
- CBC (Cipher Block Chaining)
- CTR (Counter)

By choosing the above algorithms for hardware implementation within the Crypto, the covered requirements are :

- RFC3686 AES-CTR
- RFC3602 AES-CBC
- RFC2451 Triple DES-CBC

In terms of the authentication algorithms, the Crypto IP, by using the AES-CBC mode plus software support, would be able to accelerate the implementation of the RFC3566 AES-XCBC-MAC-96 algorithm.

Further more the capability of implementing the AES-CCM (interest already exists for the near future as RFC4835 describes) is feasible via the implementation of the AES-CTR algorithm, for cryptography, plus the AES-CBC for authentication production.

The algorithm RFC2404 HMAC-SHA1-96, that the RFC4835 is using for authentication, and the RFC 2403 HMAC-MD5-96 algorithm, are not implemented, by means of hardware, because of the large die area that consumes plus the lack of significant advantages. The fact that the authentication is need every time a session initialized, combine with the fact that the above two hash functions are using many variables (more than 26x32 bits registers that means extra die area), force as to choose a software solution for the implementation of the above algorithms.

By doing that we save die area and the same time we are not pushing the processor to its limits as the above functions does not taking place so often.

31.4 ARCHITECTURE

31.4.1 Crypto

The architectural view of the Crypto is the following:

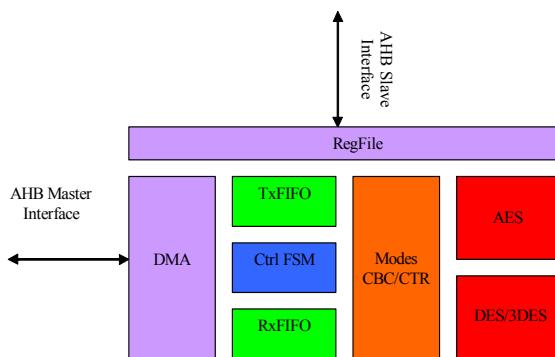


Figure 97 Crypto Architecture

The Crypto includes a DMA engine (AHB Master Interface) for transferring data between the IP and a sheared memory. The control registers of the Crypto are connected to AHB interface (AHB Slave interface).

The algorithms DES/3DES and AES are two independent building blocks but the “modes of operation” is implemented from a common module (Modes) for both algorithms. The “Modes” block is common so that the same registers can be shared for all modes of operation. The DES/3DES and AES is different building blocks because they implement completely different algorithms that are timing tight.

The “Modes” controls the DES/3DES and AES by implementing the relative mode that the selected encryption algorithms will operate each time. Also “Modes” communicates with DMA via two FIFO’s (RxFIFO and TxFIFO) which isolates the operation of the Crypto IP from the current status of the AHB-AMBA bus and the same time allow as the parallel transmission of data in bursts. By using bursts the bus is utilized better because we reduce the bus access requests, saving clock cycles that a normal bus access would needed. Further more by using bursts we achieve better access to SDRAM memory reducing the extra cycles from CAS/RAS latency that a SDRAM uses for random access.

The “Ctrl FSM” in the above block diagram checks the FIFO’s and DMA status continuously and decides for the amount of data traffic, plus which of the FIFO’s will be used. Also decides the “switch off” of the Crypto after transferring all results to the memory.

31.4.2 AES

This part of the architecture implements the AES algorithm describing in the AES-FIPS PUB 197. The capabilities that offer are the encryption and decryption of 128 bits data blocks by using 128, 192 or 256 bits encryption key.

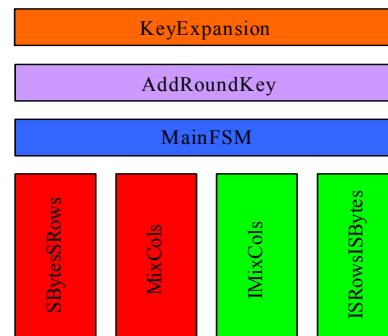


Figure 98 AES Architecture

The internal structure of the AES correlates with the logic function of the AES algorithm.

- KeyExpansion: The “KeyExpansion” is the process of generating the number of keys based on the initial key. More specific generates 11, 13 and 15 keys from an initial key of 128, 192 and 256 bits respectively. Each round of the algorithm uses each one of the above keys. For the encryption of each 128 bits input we need to use all generated, from this process, keys.
- AddRoundKey: Adds (modulo 2) the intermediate status that the input data already transformed (128 bits) with one of the generated (from “KeyExpansion”) keys. The output of this module contains the result that produced from the application of all the transformations that take place for the current round of the algorithm.
- SBytesSRows: When encryption is taking place the module applies the SubBytes transformation first and then the ShiftRows transformation on the input data.
- ISRowsISBytes: When decryption is taking place this module applies the Inverse ShiftRows transformation and then the Inverse SubBytes on the input data.
- MixCols: Being used for encryption and implements the MixColumns transformation.
- IMixCols: Being used for decryption and implements the Inverse MixColumns transformation.
- MainFSM: The basic FSM that controls all previous modules. In general controls the complete AES encryption/decryption.

All parts of AES is implemented using hardware except the “KeyExpansion” were we choose software and the reason is:

- The application doesn’t need many keys and additionally can be calculated and reused for a long time interval (The duration of a session)
- Hardware implementation means cost in die area

The support of 128, 192 and 256 bits key sizes over-heads only the “KeyExpansion” software implementation. By choosing software solution for the “KeyExpansion” algorithm the support of the 192 and 256 bits keys adds to the hardware only 4 extra memory locations (128 bits on memory cell), in which the generated keys is stored. The rest of the hardware does not overhead as the only change is the number of rounds.

31.4.3 DES/Triple DES

This block is responsible to implement the DES and Triple DES algorithms, as they describes in the DES-FIPS PUB 46-3. The capabilities that offer are the encryption/decryption of 64bits data blocks by using one key of 64 bits for DES and three keys of 64 bits for TripleDES.

The internal structure of DES/3DES looks like:

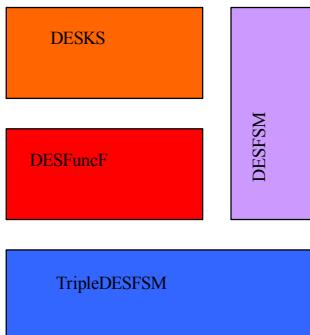


Figure 99 DES/Triple DES Architecture

- DESKS: Generates the appropriate number of keys that applies to each calculation round of DES, based on the initial key. More specific generates 16 keys of 48 bits, one key for each round of DES algorithm.
- DESFuncF: Includes all transformations that in each DES round applies on the data currently processing and the keys that the DESKS generates.
- DESFSM: Controls the DESKS and DESFuncF so that they can implement all DES rounds
- TripleDESFMSM: Uses the DESFSM to implement the Triple DES algorithm. Supplies the DESFSM with the counterpart keys in order to implement the three DES encryptions/decryptions that the Triple DES consist of.

The DES and Triple DES algorithms are implemented fully in hardware.

31.4.4 Modes

The block “Modes” uses the AES and DES/3DES in order to implement the following modes of operations:

- ECB (Electronic Code Book)
- CBC (Chiper Block Chaining)

- CTR (Counter)

By implementing all modes within one block, the reuse of the algorithm-variables registers is possible, saving also die area.

The padding requirements of the algorithms, in order for all data to be multiple of the 8 bytes (for DES/Triple DES) or 16 bytes (for AES), must be fulfil from the software. The padding simplicity allows as to not over-stressing the hardware part.

By applying successive programming of AES-CBC encryptions using software, the execution of the HMAC-XCBC-AES-96 algorithm is possible.

The implementation of the AES-CCM is feasible just like the implementation of AES-CTR algorithms for encryption, and AES-CBC for authentication generation. For both cases no extra hardware is needed but the right programming of the Crypto instead (software).

31.5 PROGRAMMING

The basic register for the programming of Crypto is the CRYPTO_CTRL_REG. Select the cryptographic algorithm by setting the CRYPTO_ALG register. The mode of operation can be programmed by choosing the suitable value for the CRYPTO_ALG_MD register. When only the final block of the resulting data must be stored to the memory set the CRYPTO_OUT_MD equal to one. Choose the encryption function or the decryption function by programming the CRYPTO_ENCDEC register. If the selected algorithm is the AES, use the CRYPTO_AES_KEY_SZ register to set the key size of the algorithm. To generate an interrupt request at the end of operation set CRYPTO_IRQ_EN equal to one.

Proportionally with the selected cryptographic algorithm and the selected mode of operation, read the CRYPTO_MREG's table and program the suitable registers with the parameters of the selected algorithms.

When the selected algorithm is the AES fill CRYPTO_KEYS_START memory with the results of the KeyExpansion function (Federal Information Processing Standards Publication 197, Advanced Encryption Standard).

Set the source address by writing the CRYPTO_FETCH_ADDR_REG register. The Crypto is reading the input data from the this memory position.

Set the destination address by writing the CRYPTO_DEST_ADDR_REG register. The Crypto is writing the output data to this memory position.

To start the calculation set CRYPTO_START_REG equal to one.

The end of calculation is indicated by the CRYPTO_INACTIVE flag. When this flag is equal to one, the calculation is finished and the resulting data are in the memory. In this case, if the CRYPTO_IRQ_EN is equal to one, an interrupt request is generated.

To clear an interrupt request from the Crypto write one to the CRYPTO_CLRIRQ register.

32.0 Quad SPI Controller

The Quad Serial Peripheral Interface Controller (QSPIC) provides a simple interface to serial FLASH memory devices. The QSPIC supports the high performance Single, Dual and Quad SPI Interfaces.

The QSPIC gives the ability to read data from a serial FLASH memory, which are memory mapped in to the SC14452 memory space, transparently through the SPI bus. In this case the QSPIC generates all the control signals for the SPI bus that are needed to read data from the serial FLASH memory. Additionally, the software can easily control the serial FLASH memory via a memory mapped register file which is contained in the QSPIC. All instructions supported by the FLASH memory, can be programmed using the above register file.

Features

- SPI Modes:
Single: Data transfer via two unidirectional pins
(Note 25).
Dual: Data transfer via two bidirectional pins.
Quad.: Data transfer via four bidirectional pins.
- Access Modes:
Auto Mode: up-to 4KByte transparent Code access for XIP (Execute In Place) and Data access.
Manual Mode: Direct SPI bus control by using the QSPIC register file.
- Up-to 82.944 MHz QSPI clock. Clock modes 0 and 3. Master mode only. The AHB clock and the SPI clock can be asynchronous or synchronous
- Vendor independent Instruction Sequencer.
- In Auto Mode the FLASH control signals are fully programmable.
- Support for single access and high performance burst mode in combination the with cache controller (in Auto Mode).
- Use of a special read instruction in the case of a specific (programmable) wrapping burst access.

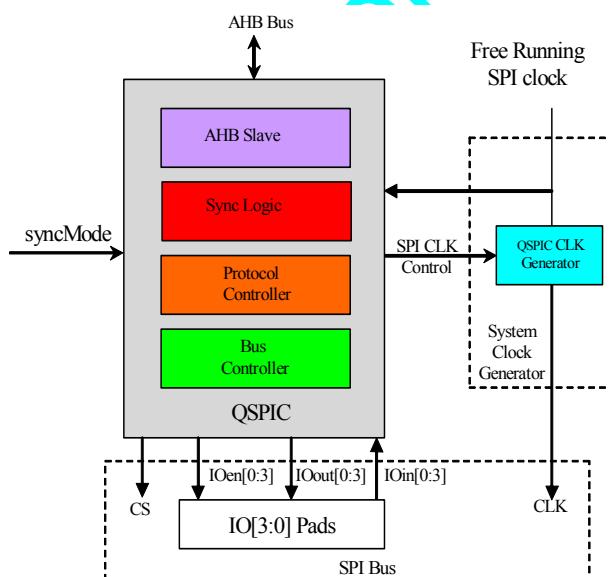


Figure 100 Quad SPI Controller architecture

32.1 INTERFACE

The Quad SPI Controller uses the following signals:

- QSPI_SCK: serial clock
- QSPI_CS: chip select
- QSPI_IO0:
 - DO (output) in Single SPI mode
 - IO0 (bidirectional) in Dual/Quad SPI mode.
- QSPI_IO1:
 - DI (input) in Single SPI mode
 - IO1 (bidirectional) in Dual/Quad SPI mode.
- QSPI_IO2:
 - General purpose (output) (e.g. WPn Write Protect) in Single SPI mode
 - IO2 (bidirectional) in Quad SPI mode.
- QSPI_IO3:
 - General purpose (output) (e.g. HOLDn) in Single SPI mode

- IO3 (bidirectional) at Quad SPI mode.

The above signals are multiplexed with the I/O port P0 signals. The multiplexing of these signals is explained in "Programmable Pin Assignment"

32.2 SPECIFICATION

32.2.1 SPI modes

The Quad SPI Controller (QSPIC) supports the following SPI standards:

- Single SPI: Data transfer via two unidirectional pins.
- Note 25:** The QSPIC supports communication to any single/dual or Quad SPI FLASH memory. In contradiction to the Standard SPI interface, the supported Single SPI interface does **not** support the bus modes 1 and 2, does **not** support full-duplex communications and does **not** support any SPI slave mode.
- Dual SPI (the data is transferred via two wires)
 - Quad SPI (the data is transferred via four wires)

32.2.2 Access modes

The access to a serial FLASH connected to the QSPI can be done in two modes:

- Auto mode
- Manual mode

These modes are **mutually exclusive**. The serial FLASH can be controlled only in one of the two modes. The registers which control the mode of operation can be used at any time.

The maximum size of the serial FLASH memory for the Auto Mode is 16 Mbytes (24 bits addressing). When the Manual Mode is selected, there is no restriction on the size of memory.

Auto mode

In auto mode the read access from the serial FLASH memory is full transparently through the SPI bus. A memory address range of 4KBytes is defined by the address pointers QSPIC_AUTO_START and QSPIC_AUTO_END. A read access to the 4KByte AHB bus range interface is translated by the QSPIC to the respective SPI bus control signals that are needed for the serial FLASH memory data reading access.

When the Auto Mode is disabled, any access (reading or writing) in the 4KBytes range will be ignored. In this case the response of the controller's slave AHB interface will be "OK". When the Auto Mode is enabled, only read access is supported. A write access causes an IAD interrupt to the CR16. A read access can be single access or incremental burst or wrapping burst access. The wrapping burst is supported even when the controlled serial FLASH don't support any special instruction for wrapping burst. A special read instruction it can be used in the case of a specific (programmable) wrapping burst access. When a serial FLASH supports a special instruction for wrapping burst access, this feature saves access time (less wait states). For maximizing the utilization of the bus and minimizing the number

of wait states, it is recommended to use burst accesses. However, non-sequential random accesses are supported with the cost of more wait states.

Manual Mode

In manual mode the serial FLASH memory is controlled via a register file. All instructions that are supported by a serial FLASH memory can be programmed using the register file. Moreover, the mode of interface (SPI, Dual SPI, Quad SPI) and the mode of operation (Auto or Manual Mode) can be configured via this register file. The register file supports the following data sizes for reading and writing accesses: 8-bits, 16-bits and 32 bits.

32.2.3 Endianess

The QSPIC IP operates in little-endian mode. For 32-bit or 16 bit access (for read and write operations) to a serial FLASH memory the least-significant byte comes first. For 32-bit access the byte ordering is: data [7:0], data [15:8], data [23:16], data [31:24] and for 16-bit access the byte ordering is: data [7:0], data [15:8].

32.2.4 Clock relations

The AHB clock and the SPI clock can be asynchronous or synchronous. The QSPIC IP contains synchronization logic to ensure correct data transfers from one clock domain to another. When the syncMode is selected no wait states (for synchronization reasons) are inserted in the AHB bus.

When SPI clock is the same as the AHB clock, the synchronization logic must be bypassed so that wait states on the AHB bus can be reduced. With the external input syncMode (see programming section) the relation between the two clocks can be declared. Setting syncMode to one (1), the AHB clock is supposed to be equal to the SPI clock thus bypassing the synchronization logic. Changing the state of this port, should be performed while the SPI is in reset state.

32.3 PROGRAMMING

32.3.1 Auto Mode

The 24bits address space is accessible through a 4096x4k paging scheme :

- GPRG_R1_REG[11:0] replaces the HADDR[23:12]
- The byte in a page is specified through the HADDR[11:0] address bits.

Read access to the memory space which starts from the QSPIC_AUTO_START and ends to the QSPIC_AUTO_END is translated by the QSPIC to the respective SPI bus signals, that is needed for the reading of the external SPI FLASH memory.

In the case of Auto Mode of operation the QSPIC generates a sequence of control signals in SPI BUS. This sequence of control signals is analysed to the following **phases**: instruction phase, address phase, extra byte phase, dummy clocks phase and read data phase. These phases can be programmed via registers

- QSPIC_BURSTCMDA_REG
- QSPIC_BURSTCMDB_REG.

Bit QSPIC_INST is used to set the selected instruction for the cases of incremental burst or single read access. If bit QSPIC_WRAP_MD is equal to 1, bitfield QSPIC_INST_WB can be used to set the used instruction for the case of a wrapping burst read access of length and size described by the bitfields QSPIC_WRAP_LEN and QSPIC_WRAP_SIZE respectively. In all other cases the QSPIC_INST is the selected instruction.

If the **instruction** must be transmitted only in the first access after the selection of Auto Mode then the QSPIC_INST_MD must be equal to 1.

To enable the **extra byte phase** set 1 to the QSPIC_EXT_BYTE_EN register. The transmitted byte during the extra byte phase is specified from the QSPIC_EXT_BYTE register. To disable (hi-z) the output pads during the transmission of bits [3:0] of extra byte, write 1 to the QSPIC_EXT_HF_DS register.

The number of **dummy bytes** during the dummy clocks phase is specified from the bitfield QSPIC_DMY_NUM.

The SPI BUS mode during each phase can be set with register bits:

- QSPIC_INST_TX_MD for the instruction phase
- QSPIC_ADR_TX_MD for the address phase
- QSPIC_EXT_TX_MD for the extra byte phase
- QSPIC_DMY_TX_MD for the dummy byte phase
- QSPIC_DAT_RX_MD for the read data phase.

If the Quad SPI mode is selected in any of the above phases, write 0 to the QSPIC_IO3_OEN and QSPIC_IO2_OEN.

If the serial FLASH Memory must be prepared for reading with the use of any instruction except the read instruction, then the **Manual Mode** must be used for the programming of the above instructions.

The final step to enable the use of Auto Mode of operation is to set the QSPIC_AUTO_MD equal to 1.

32.3.2 Manual Mode

For the Manual Mode of operation the QSPIC_AUTO_MD must be equal to zero.

Manual operation of the bus signal is done via QSPIC_CTRL_REG:

- The start and the finish of an access can be controlled using bits QSPIC_EN_CS and QSPIC_DIS_CS respectively.
- The SPI mode bus mode of operation set with bits QSPIC_SET_SINGLE, QSPIC_SET_DUAL and QSPIC_SET_QUAD.

Writing to QSPIC_WRITEDATA register is generating a data transfer from the QSPIC to the SPI bus.

A read access at QSPIC_READDATA register is generating a data transfer from the SPI bus.

Writing to QSPIC_DUMMYDATA register is generating a number of clock pulses to the SPI bus. During this activity in the SPI bus, the QSPI_IO data pads are in hi-z state.

When an access to the SPI bus via QSPIC_WRITEDATA, QSPIC_READDATA and QSPIC_DUMMYDATA is very slow, the delay in access to the AHB is very high. In this case set the QSPIC_HRDY_MD register equal to 1. With this feature the HREADY signal of the AHB slave interface is always equal to 1, when accessing the WriteData, ReadData and DummyData registers. All the masters can access the AHB bus without the waiting of the end of transmitting on SPI Bus. A reading of the QSPIC_BUSY register must be done to check the end of the activity at the SPI bus, before any more access to this. In this case the register QSPIC_RECVDATA contains the received data at the end of a read access.

The state and the value of the QSPI_IO[3:2] is specified with the registers bits:

- QSPIC_IO3_OEN, QSPIC_IO3_DAT
(Used for the WPn, Write Protect function).
- QSPIC_IO2_OEN, QSPIC_IO2_DAT respectively
(Used for the HOLDn function).

32.3.3 Clock selection

The SPI clock mode as set with bit QSPIC_CLK_MD

The supported modes for the generated SPI clock is:

- 0 = Mode 0. The QSPI_SCK is low, when the bus is idle (QSPI_CS is high).
- 1= Mode 3. The QSPI_SCK is high, when the bus is idle (QSPI_CS is high).

To program a new frequency for the SPI bus, it is recommended to follow this procedure:

- Set the SPI bus to the idle state (CS high).
- Disable the QSPIC clock (SW_QSPIC_EN).
- The selection of the required frequency can be set through register CLK_AUX2_REG to:

Table 65: QSPI_CLK selection

QCLK_CLK	SW_QSPIC_SEL_SOURCE	SW_QSPIC_CLK_DIV	SyncMode
hclk	1	0	TRUE
hclk/2	1	1	FALSE
PLL2	0	0	FALSE
PLL2/2	0	1	FALSE

- Enable the QSPIC clock (SW_QSPIC_EN).

32.3.4 Received data

The standard method to sample the received data is by

using the positive edge of the QSPI_SCK. However, when the output delay of the serial FLASH is high, a timing problem at the reading path is very likely. For this reason the QSPIC can be programmed to sample the received data with the negative edge of the QSPI_SCK. This is specified with the QSPIC_RXD_NEG register.

Confidential

33.0 Generic Programming Registers

33.1 GENERAL DESCRIPTION

The SC14452 contains peripherals that need some control signals which can not be driven by their internal register files. The purpose of this block is to implement and fully control by S/W any signal of this kind in SC14452. It has the following features:

- Contains only two 16-bit registers
- Both registers are accessible via APB memory segment.

Refer to

- GPRG_R0_REG (0xFF7000) (table 98, page 194)
- GPRG_R1_REG (0xFF7002) (table 99, page 194)

for their detailed description.

Confidential

34.0 Interrupt Control Unit (ICU)

The SC14452 has 8 non-maskable interrupt, maskable interrupts and 6 exception vectors. If an interrupt source becomes active the ICU generates an interrupt address. In case of a maskable interrupt source the ICU determines whether this interrupt source has the highest priority before the CR16C+ is forced to terminate the current instruction. In case of a non-maskable interrupt source the CR16C+ is always forced to terminate the current instruction. After saving the stack pointer and processor status register, the CR16C+ executes interrupt service procedure at

INTBASE + [vector * 2] if CFG.ED=0
INTBASE + [vector * 4] if CFG.ED=1

The INTBASE can be stored with CR16C+ instruction <LPR> or <LPRD>.

The vector addresses are shown in Table 66.

The NMI is shared between the watchdog timer and the DIP <U_VNMI> interrupt. Reading DIP1_STATUS_REG determines whether the DIP generated the interrupt or not.

Interrupt priorities

The CR16C+ has six fixed exception priorities as shown in Table 66. The DBG_TRAP has priority 1, IAD_TRAP has priority 2. The NMI is non maskable with priority 3. The ISE and exception vectors have the lowest priorities 5 and 6.

The non maskable interrupts with priority 4 have a programmable sub priority level which can be set in registers INTx_PRIORITY_REG (Table 230 to Table 232). Writing priority value '000' will disable these interrupts. If more than one interrupt request occurs the one with the highest priority is accepted, the others stay pending.

At the end of an interrupt service routine, the pending interrupt must be cleared by software by writing a 1 to register RESET_INT_PENDING_REG (see Table 315). This register also gives the status of the pending interrupt if read. If maskable interrupts have the same interrupt level the inherent priority scheme is used.

The maskable interrupts can also be set by software by writing a 1 to register SET_INT_PENDING_REG (see Table 317). This register also returns the status of the pending interrupt.

The watchdog timer will generate an NMI in case the TIMER_CTRL_REG[WDOG_CTRL] = 0, else a reset is generated.

Some interrupts are "ored" ; the interrupt routine must read the status registers to determine which interrupt has occurred. Next one or more service routines must be executed. Finally the active interrupt sources must be cleared and the RESET_INT_PENDING_REG can be set.

Four interrupts can also be generated by either DMA channels or other sources. DMAX_CTRL_REG[DINT]

can be set to make this selected. Table 66 column "Secondary function with DINT" indicates which interrupts are selectable. See also Figure 53 on page 95.

See also AN-D-027 "Interrupt handling with SC144xx"

Table 66: SC14452 Interrupt Vectors

Interrupt source	Vector	Exception priority	Inherent sub priority	Description	Secondary function with DINT
Reserved	00	-	-		
NMI_INT	01	3	-	Non maskable interrupt from DIP <VNMI> and Watchdog timer	
Reserved	02	-	-		
Reserved	03	-	-		
Reserved	04	-	-		
SVC_TRAP	05	6 (exceptions)	-	Supervisor Call Trap	
DVZ_TRAP	06			Divide by zero Trap	
FLG_TRAP	07			Flag Trap	
BPT_TRAP	08			Breakpoint Trap	
TRC_TRAP	09			Trace Trap	
UND_TRAP	10			Undefined Instruction Trap	
Reserved	11	-	-		
IAD_TRAP	12	2	-	Illegal address Trap	
Reserved	13	-	-		
DBG_TRAP	14	1	-	Debug Trap	
ISE_INT	15	5	-	In System Emulator Interrupt caused by ISE pin or SDI	
ACCESS12_INT	16	4	Lowest	Access bus 1 and 2 interrupt	
KEYB_INT	17	4		Keyboard Interrupt or vectored level/edge interrupt	DMA1_INT
EMAC_INT	18	4		EMAC interrupt	
CT_CLASSD_INT	19	4		Capture timer 1,2 interrupt or CLASSD clipping interrupt	
UART_RI_INT	20	4		UART RI interrupt	DMA2_INT
UART_TI_INT	21	4		UART TI interrupt	DMA3_INT
SPI1_ADC_INT	22	4		SPI 1 or ADC Interrupt	DMA0_INT
TIM0_INT	23	4		Timer 0 interrupt	
TIM1_INT	24	4		Timer 1 interrupt	
CLK100_INT	25	4		CLK100 interrupt	
DIP_INT	26	4		DIP Interrupt	
CRYPTO_INT	27	4		CRYPTO Interrupt	
SPI2_INT	28	4		SPI 2 interrupt	
DSP1_INT	29	4		Gen2DSP 1 Interrupt	
DSP2_INT	30	4	Highest	Gen2DSP 2 Interrupt	

Refer to Figure 53 on page 95 for an overview of interrupt multiplexers.

35.0 Memory map overview

Table 67: SC14452 CR16C+ Memory Map

Address	On-chip	Off-chip
0x000.0000 0x000.5FFF	Configurable non-shared RAM or iCache or dCache (24k*8)	Not accessible off-chip
0x000.6000 0x000.7FFF	Reserved (8k*8)	
0x0000.8000 0x0000.9FFF	Configurable non-shared RAM or iCache administration or dCache administration (8k*8)	
0x0000.A000 0x0000.FFFF	Reserved (24k*8)	
0x0001.0000 0x0001.FFFF	Shared RAMs for CR16C+, Gen2DSP1, 2, DIP 28k*8 (See Table 68 on page 167 for detailed map)	
0x0002.0000 0x000E.FFFF	External Program Memory (SDRAM / FLASH / ROM/ RAM) (832k*8)	15.5625 MBytes. ACSx: via external memory controller upto 32Mbyte.
0x000F.0000 0x00FE.EFFF	External Program Memory (SDRAM / FLASH / ROM/ RAM) 15.42M*8	
0x00FE.F000 0x00FE.F7FF	Boot ROM (2k*8)	Not accessible off-chip
0x00FE.F800 0x00FE.FFFF	Reserved (2k*8)	
0x00FF.0000 0x00FF.FBFF	On-chip Peripheral and Chip ID Registers (see Table 71 on page 171) (63k*8)	
0x00FF.FC00 0x00FF.FFFF	Internally used for ICU vectors and traps 1k*8	
0x0100.0000 0x0100.01FF	DIP Sequencer RAM (512*8)	
0x0100.0200 0x0100.03FF	DIP Sequencer RAM (512*8) addressed with <JMPF>,<BR_B1> commands	
0x0100.0400 0x0100.FFFF	Reserved (63k*8)	
0x0101.0000 0x0102.FFFF	Shared ROMs + Registers for Gen2DSP1, 2, CR16C+ 128k*8 (See Table 68 on page 167)	
0x0103.0000 0x0107.FFFF	Micro Program ROM1, RAM2, RAM1, ROM2 for Gen2DSP1, 2 (Note 26) 256k*8 (see Table 69 on page 167 for detailed map)	
0x0108.0000 0x01FF.FFFF	Not accessible on-chip (16M - 512k)*8.	Data only. (Note 28) (16M - 512k)Bytes
0x0200.0000 0xFFFF.FFFF	Reserved (4G - 32M)*8.	Not accessible off-chip

Note 26: The Gen2DSP Micro program ROMs are accessible by CR16C+ only in testmode.

Note 27: 1k*8 = 1 kbyte = 1024 bytes. 1Mbit = 1024*1024 bits. 1Gbyte = 1024*1024*1024 bytes.

Note 28: This region can only be accessed for data-manipulation and only using register pair relative and index addressing modes. Therefore use this address range for infrequently used data.

Note 29: No DMA access should be performed at the non-shared RAM range, i.e. at the range of 0x0000.0000 up to 0x0000.FFFF.

Table 68: Detailed shared Data RAM and ROM Memory map for DIP, CR16C+ and Gen2DSP1 and Gen2DSP2

CR16C+ Address	DIP Address	Gen2DSP1 Data Address	Gen2DSP2 Data Address	Device (Fast/slow for Gen2DSPx)
0x001.0000 0x001.3FFF (16k*8)	0x0000 0x3FFF (16k*8)	0x0000 0x1FFF (8k*16)	0x4000 0x5FFF (8k*16)	Shared RAM 1 (4k*32) (Fast for 1, Slow for 2)
0x001.4000 0x001.7FFF (16k*8)	0x4000 0x7FFF (16k*8)	0x2000 0x3FFF (8k*16)	0x6000 0x7FFF (8k*16)	Shared RAM 1 (Reserved)
0x001.8000 0x001.CFFF (20k*8)	0x8000 0xCFFF(20k*8)	0x4000 0x67FF (10k*16)	0x0000 0x27FF (10k*16)	Shared RAM 2 (5k*32) (Fast for 2, Slow for 1)
0x001.D000 0x001.FFFF (12k*8)	0xD000 0xFFFF (12k*8)	0x6800 0x7FFF (6K*16)	0x2800 0x3FFF (6k*16)	Shared RAM 2 (Reserved)
0x101.0000 0x101.07FF (2k*8)		0x8000 0x83FF (1k*16)		Shared ROM 1 (512x32) (Fast access for 1 only)
0x101.0800 0x101.FF4F ((62k-176)*8)		0x8400 0xFFA7 ((31k-88)*16)		Shared ROM 1 (Reserved)
0x101.FF50 0x101.FF7F (48*8)				Reserved
0x101.FF80 0x101.FFCF (80*8)		0xFFC0 0xFFE7 (40*16)	0xFFC0 0xFFE7 (40*16)	Gen2DSP1/2 common registers (Table 196 on page 226)
0x101.FFD0 0x101.FFFF (48*8)		0xFFA8 0xFFBF (24*16)	0xFFE8 0xFFFF (24*16)	Gen2DSP1 registers (Table 196 on page 226)
0x102.0000 0x102.FF4F ((64k-176)*8)			0x8000 0xFFA7 ((32k-88)*16)	Shared ROM 2 (16k*32) (Fast access for 2 only, last 44 DWords not accessible)
0x102.FF50 0x102.FFCF(128*8)				Reserved
0x102.FFD0 0x102.FFFF (48*8)		0FFE8 0xFFFF (24*16)	0FFA8 0xFFBF (24*16)	Gen2DSP2 registers (Table 202 on page 227)

Table 69: Detailed micro code RAM and ROM Memory map for DIP, CR16C+ and Gen2DSP1 and Gen2DSP2

CR16C+ Address	DIP Code Address	Gen2DSP1 Code Address	Gen2DSP2 Code Address	Micro Code Memory
0x100.0000 0x100.03FF (1k*8)	0x0000 0x1FF (512*16)			DIP Micro code RAM
0x100.0400 0x100.0FFF (3k*8)	0x200 0x7FF(1536*16)			DIP Micro code RAM (Reserved)
0x103.0000 0x103.4FFF (20k*8)		0x0000 0x27FF (10k*16)		Micro code RAM 1 (5k*32) (Fast access for 1 only) (Note 30)
0x103.5000 0x103.7FFF (12k*8)		0x2800 0x2FFF		Micro code RAM 1 (Reserved)
0x103.8000 0x103.BFFF (16K*8)		0x3000 0x4FFF (8k*16)	0x0000 0x1FFF (8k*16)	Micro code RAM 2 (4k*32) (Fast for 2, Slow for 1)
0x103.C000 0x103.FFFF (16k*8)				Micro code RAM 2 (Reserved)
0x104.0000 (Note 31) 0x104.7FFF (32k*8)		0x8000 0xFFFF (16k*16)		Micro code ROM 1 (8k*32) (Fast access for 1 only)

Table 69: Detailed micro code RAM and ROM Memory map for DIP, CR16C+ and Gen2DSP1 and Gen2DSP2

CR16C+ Address	DIP Code Address	Gen2DSP1 Code Address	Gen2DSP2 Code Address	Micro Code Memory
0x104.8000 0x105.FFFF (96k*8)				Micro code ROM 1 (Reserved)
0x106.0000 (Note 31) 0x107.BFFF (112k*8)			0x2000 0xFFFF (56k*16)	Micro code ROM 2 (28k*32) (Fast access for 2 only)
0x107.C000 0x107.FFFF (16k*8)				Micro code ROM 2 (Reserved)

Note 30: Gen2DSP2 access only the first 8K*16 out of 10K*16 bits of "Gen2DSP1 Micro code RAM 1".

Note 31: Microcode ROM is only accessible by CR16C+ in test mode. The Gen2DSP accesses the ROM code via the index.

Note 32: Note that each GEN2DSP has no access to the ROMs of the other GEN2DSP.

35.1 GEN2DSP1, GEN2DSP2 ROM TABLES AND REGISTER FILES

For analytical cross reference of the GEN2DSP1 and GEN2DSP2 ROM tables in contradiction to the CR16C+ memory map refer to AN-D-129 and Table 36.

Table 70: Gen2DSP1 and Gen2DSP2 internal + Common registers

CR16C+ Address	Gen2DSP1 Data Address	Gen2DSP2 Data Address	Port	Description
Gen2DSP1 and Gen2DSP2 common registers				
0x101FF80	0xFFC0	0xFFC0	DSP_MAIN_SYNC0_REG	DSP main counter outputs selection register 0
0x101FF82	0xFFC1	0xFFC1	DSP_MAIN_SYNC1_REG	DSP main counter outputs selection register 1
0x101FF84	0xFFC2	0xFFC2	DSP_MAIN_CNT_REG	DSP main counter register
0x101FF86	0xFFC3	0xFFC3	DSP_ADC0S_REG	ADC0 Input 2's Complement register
0x101FF88	0xFFC4	0xFFC4	DSP_ADC1S_REG	ADC1 Input 2's Complement register
0x101FF8A	0xFFC5	0xFFC5	DSP_CLASSD_REG	CLASSD Output output data register
0x101FF8C	0xFFC6	0xFFC6	DSP_CODEC_MIC_GAIN_REG	CODEC MIC GAIN register
0x101FF8E	0xFFC7	0xFFC7	DSP_CODEC_OUT_REG	CODEC DATA output register
0x101FF90	0xFFC8	0xFFC8	DSP_CODEC_IN_REG	CODEC DATA input register
0x101FF92	0xFFC9	0xFFC9	DSP_RAM_OUT0_REG	Shared RAM 1 or 2 output register 0
0x101FF94	0xFFCA	0xFFCA	DSP_RAM_OUT1_REG	Shared RAM 1 or 2 output register 1
0x101FF96	0xFFCB	0xFFCB	DSP_RAM_OUT2_REG	Shared RAM 1 or 2 output register 2
0x101FF98	0xFFCC	0xFFCC	DSP_RAM_OUT3_REG	Shared RAM 1 or 2 output register 3
0x101FF9A	0xFFCD	0xFFCD	DSP_RAM_IN0_REG	Shared RAM 1 or 2 input register 0
0x101FF9C	0xFFCE	0xFFCE	DSP_RAM_IN1_REG	Shared RAM 1 or 2 input register 1
0x101FF9E	0xFFCF	0xFFCF	DSP_RAM_IN2_REG	Shared RAM 1 or 2 input register 2
0x101FFA0	0xFFD0	0xFFD0	DSP_RAM_IN3_REG	Shared RAM 1 or 2 input register 3
0x101FFA2	0xFFD1	0xFFD1	DSP_ZCROSS1_OUT_REG	ZERO CROSSING 1 output register
0x101FFA4	0xFFD2	0xFFD2	DSP_ZCROSS2_OUT_REG	ZERO CROSSING 2 output register

Table 70: Gen2DSP1 and Gen2DSP2 internal + Common registers

CR16C+ Address	Gen2DSP1 Data Address	Gen2DSP2 Data Address	Port	Description
0x101FFA6	0xFFD3	0xFFD3	DSP_PCM_OUT0_REG	PCM channel 0 output register
0x101FFA8	0xFFD4	0xFFD4	DSP_PCM_OUT1_REG	PCM channel 1 output register
0x101FFAA	0xFFD5	0xFFD5	DSP_PCM_OUT2_REG	PCM channel 2 output register
0x101FFAC	0xFFD6	0xFFD6	DSP_PCM_OUT3_REG	PCM channel 3 output register
0x101FFAE	0xFFD7	0xFFD7	DSP_PCM_IN0_REG	PCM channel 0 input register
0x101FFB0	0xFFD8	0xFFD8	DSP_PCM_IN1_REG	PCM channel 1 input register
0x101FFB2	0xFFD9	0xFFD9	DSP_PCM_IN2_REG	PCM channel 2 input register
0x101FFB4	0xFFDA	0xFFDA	DSP_PCM_IN3_REG	PCM channel 3 input
0x101FFB6	0xFFDB	0xFFDB	DSP_PCM_CTRL_REG	PCM Control register
0x101FFB8	0xFFDC	0xFFDC	DSP_PHASE_INFO_REG	Phase information between PCM FSC 8/16/32 and main counter 8/16/32 kHz
0x101FFBA	0xFFDD	0xFFDD	DSP_VQI_REG	BMC VQI Register
0x101FFBC	0xFFDE	0xFFDE	DSP_MAIN_CTRL_REG	DSP Main counter control and preset value
0x101FFBE	0xFFDF	0xFFDF	DSP_CLASSD_BUZZOFF_REG	CLASSD buzzer on/off control
0x101FFC0	0FFE0	0FFE0	-	Reserved
0x101FFCF	0FFE7	0FFE7		
Gen2DSP1 registers (shared by Gen2DSP2)				
0x101FFD0	0xFFA8	0FFE8	DSP1_CTRL_REG	DSP1 control register
0x101FFD2	0xFFA9	0FFE9	DSP1_PC_REG	DSP1 Program counter
0x101FFD4	0xFFAA	0FFEA	DSP1_PC_START_REG	DSP1 Program counter start
0x101FFD6	0xFFAB	0FFEB	DSP1_IRQ_START_REG	DSP1 Interrupt vector start
0x101FFD8	0xFFAC	0FFEC	DSP1_INT_REG	DSP1 to CR16C+ interrupt vector
0x101FFDA	0xFFAD	0FFED	DSP1_INT_MASK_REG	DSP1 to CR16C+ interrupt vector mask
0x101FFDC	0FFAE	0FFEE	DSP1_INT_PRIO1_REG	DSP1 interrupt mux 1
0x101FFDE	0FFAF	0FFEF	DSP1_INT_PRIO2_REG	DSP1 interrupt mux 2
0x101FFE0	0FFB0	0FFF0	DSP1_OVERFLOW_REG	DSP1 to CR16C+ Interrupt overflow
0x101FFE2	0FFB1	0FFF1	DSP1_JTBL_START_REG	DSP1 jump table start address
0x101FFE4	0FFB2	0FFF2	-	Reserved
0x101FFEE	0FFB7	0FFF7		
0x101FFF0	0FFB8	0FFF8	DBG1_IREG	DSP1 JTAG instruction register
0x101FFF2	0FFB9	0FFF9	-	Reserved
0x101FFF4	0FFBA	0FFFA	DBG1_INOUT_REG_LSW	DSP1 DEBUG data register (32 bits)
0x101FFF6	0FFBB	0FFFB	DBG1_INOUT_REG_MSB	
0x101FFF8	0FFBC	0FFFC	-	Reserved
0x101FFFF	0FFBF	0FFFF		
Gen2DSP2 registers (shared by Gen2DSP1)				
0x102.FFD0	0FFE8	0FFA8	DSP2_CTRL_REG	DSP2 control register
0x102.FFD2	0FFE9	0FFA9	DSP2_PC_REG	DSP2 Program counter
0x102.FFD4	0FFEA	0FFAA	DSP2_PC_START_REG	DSP2 Program counter start
0x102.FFD6	0FFEB	0FFAB	DSP2_IRQ_START_REG	DSP2 Interrupt vector start
0x102.FFD8	0FFEC	0FFAC	DSP2_INT_REG	DSP2 to CR16C+ interrupt vector

Table 70: Gen2DSP1 and Gen2DSP2 internal + Common registers

CR16C+ Address	Gen2DSP1 Data Address	Gen2DSP2 Data Address	Port	Description
0x102.FFDA	0xFFED	0xFFAD	DSP2_INT_MASK_REG	DSP2 to CR16C+ interrupt vector mask
0x102.Ffdc	0xFFEE	0xFFAE	DSP2_INT_PRIO1_REG	DSP2 interrupt mux 1
0x102.Ffde	0xFFEF	0xFFAF	DSP2_INT_PRIO2_REG	DSP2 interrupt mux 2
0x102.Ffe0	0xFFFF0	0xFFB0	DSP2_OVERFLOW_REG	DSP2 to CR16C+ Interrupt overflow
0x102.Ffe2	0xFFFF1	0xFFB1	DSP2_JTBL_START_REG	DSP2 jump table start address
0x102.Ffe2	0xFFFF2	0xFFB2	-	Reserved
0x102.Ffee	0xFFFF7	0xFFB7		
0x102.Fff0	0xFFFF8	0xFFB8	DBG2_IREG	DSP2 JTAG instruction register
0x102.Fff2	0xFFFF9	0xFFB9	-	Reserved
0x102.Fff4	0xFFFFA	0xFFBA	DBG2_INOUT_REG (LSW)	
0x102.Fff6	0xFFFFB	0xFFBB	DBG2_INOUT_REG (MSW)	DSP2 DEBUG data register (32 bits)
0x102.Fff8	0xFFFFC	0xFFBC	-	Reserved
0x102.Ffff	0xFFFFF	0xFFBF		

Confidential

36.0 Memory map Detailed

Table 71: Detailed memory Map

Note 33: Test registers are not defined here. Write access to undefined addresses might result in undefined behaviour.

Address	Port	Description
0x000000	RAM_START	Non-Shared RAM/dCache/iCache (24k bytes)
0x005FFF	RAM_END	
0x006000		Reserved
0x007FFF		(8kbyte)
0x008000	RAM_ADMIN_START	Non-Shared RAM/dCache/iCache Admin
0x009FFF	RAM_ADMIN_END	(8k bytes)
0x00A000		Reserved
0x00FFFF		(24kbyte)
0x010000	SHARED_RAM1_START	Shared RAM1
0x013FFF	SHARED_RAM1_END	(16k bytes)
0x014000		Reserved 16k bytes
0x017FFF		
0x018000	SHARED_RAM2_START	Shared RAM2
0x01CFFF	SHARED_RAM2_END	(20k bytes)
0x01D000		Reserved 12k bytes
0x01FFFF		
0x020000		Free
0x0EFFFF		
0x0F0000	PROGRAM_START	External Program memory
0xFEEFFF	PROGRAM_END	14.75 Mbyte
0xFEFO00	BOOT_ROM_START	Boot ROM (2k bytes)
0xFEF7FF	BOOT_ROM_END	
0xFEF800		Reserved for boot ROM
0xFFFFFFF		
0xFF0000	EBI_SDCONR_REG	SDRAM Configuration register
0xFF0004	EBI_SDTMG0R_REG	SDRAM Timing register 0
0xFF0008	EBI_SDTMG1R_REG	SDRAM Timing register 1
0xFF000C	EBI_SDCTRL_REG	SDRAM Control register
0xFF0010	EBI_SDREFR_REG	SDRAM Refresh register
0xFF0014	EBI_ACS0_LOW_REG	Chip select 0 base address register
0xFF0018	EBI_ACS1_LOW_REG	Chip select 1 base address register
0xFF001C	EBI_ACS2_LOW_REG	Chip select 2 base address register
0xFF0020	EBI_ACS3_LOW_REG	Chip select 3 base address register
0xFF0024	EBI_ACS4_LOW_REG	Chip select 4 base address register
0xFF0054	EBI_ACS0_CTRL_REG	Chip select 0 control register
0xFF0058	EBI_ACS1_CTRL_REG	Chip select 1 control register
0xFF005C	EBI_ACS2_CTRL_REG	Chip select 2 control register
0xFF0060	EBI_ACS3_CTRL_REG	Chip select 3 control register
0xFF0064	EBI_ACS4_CTRL_REG	Chip select 4 control register

Table 71: Detailed memory Map

Note 33: Test registers are not defined here. Write access to undefined addresses might result in undefined behaviour.

Address	Port	Description
0xFF0094	EBI_SMTMGR_SET0_REG	Static memory Timing register Set 0
0xFF0098	EBI_SMTMGR_SET1_REG	Static memory Timing register Set 1
0xFF009C	EBI_SMTMGR_SET2_REG	Static memory Timing register Set 2
0xFF00A0	EBI_FLASH_TRPDR_REG	FLASH Timing Register
0xFF00A4	EBI_SMCTRLR_REG	Static memory Control register
0xFF00A8		Reserved
0xFF00AC	EBI_SDEXN_MODE_REG	Mobile SDRAM Extended mode register
0xFF0400		Reserved
0xFF07FF		
0xFF0800	CRYPTO_CTRL_REG	Crypto control register
0xFF0804	CRYPTO_START_REG	Start calculation
0xFF0808	CRYPTO_FETCH_ADDR_REG	DMA Fetch address
0xFF080C	CRYPTO_LEN_REG	The length of input block in bytes
0xFF0810	CRYPTO_DEST_ADDR_REG	DMA Destination address
0xFF0814	CRYPTO_STATUS_REG	Status register
0xFF0818	CRYPTO_CLRIrq_REG	Clear interrupt request
0xFF081C	CRYPTO_MREG0_REG	Mode depended register 0
0xFF0820	CRYPTO_MREG1_REG	Mode depended register 1
0xFF0824	CRYPTO_MREG2_REG	Mode depended register 2
0xFF0828	CRYPTO_MREG3_REG	Mode depended register 3
0xFF082C	CRYPTO_MREG4_REG	Mode depended register 4
0xFF0830	CRYPTO_MREG5_REG	Mode depended register 5
0xFF0834	CRYPTO_MREG6_REG	Mode depended register 6
0xFF0838	CRYPTO_MREG7_REG	Mode depended register 7
0xFF083C		Reserved
0xFF08FF		
0xFF0900	CRYPTO_KEYS_START	AES Keys storage memory
0xFF09FF	CRYPTO_KEYS_END	
0xFF0A00		Reserved
0xFF0BFF		
0xFF0E00	QSPIC_CTRLBUS_REG	SPI Bus Control register
0xFF0E04	QSPIC_CTRLMODE_REG	QSPIC Mode Control register
0xFF0E08	QSPIC_RECVDATA_REG	Received Data
0xFF0E0C	QSPIC_BURSTCMDA_REG	The way of reading in Auto Mode
0xFF0E10	QSPIC_BURSTCMDB_REG	The way of reading in Auto Mode
0xFF0E14	QSPIC_STATUS_REG	QSPIC Status register
0xFF0E18	QSPIC_WRITEDATA_REG	Write Data to SPI Bus
0xFF0E1C	QSPIC_READDATA_REG	Read Data from SPI Bus
0xFF0E20	QSPIC_DUMMYDATA_REG	Send Dummy clocks to SPI Bus
0xFF0E24		Reserved
0xFF0FFF		

Table 71: Detailed memory Map

Note 33: Test registers are not defined here. Write access to undefined addresses might result in undefined behaviour.

Address	Port	Description
0xFF1000	QSPIC_AUTO_START	QSPIC Auto Mode Memory space
0xFF1FFF	QSPIC_AUTO_END	
0xFF2000	EMAC_MACR0_CONFIG_REG	MAC Configuration Register
0xFF2004	EMAC_MACR1_FRAME_FILTER_REG	MAC Frame Filter
0xFF2010	EMAC_MACR4_MII_ADDR_REG	MII Address Register
0xFF2014	EMAC_MACR5_MII_DATA_REG	MII Data Register
0xFF2018	EMAC_MACR6_FLOW_CTRL_REG	Flow Control Register
0xFF201C	EMAC_MACR7_VLAN_TAG_REG	VLAN Tag Register
0xFF2020	EMAC_MACR8_CORE_VER_REG	Version Register
0xFF2038	EMAC_MACR14_INT_REG	Interrupt Register
0xFF203C	EMAC_MACR15_INT_MSK_REG	Interrupt Mask Register
0xFF2040	EMAC_MACR16_MAC_ADDR0_HIGH_REG	MAC Address0 High Register
0xFF2044	EMAC_MACR17_MAC_ADDR0_LOW_REG	MAC Address0 Low Register
0xFF2048	EMAC_MACR18_MAC_ADDR1_HIGH_REG	MAC Address1 High Register
0xFF204C	EMAC_MACR19_MAC_ADDR1_LOW_REG	MAC Address1 Low Register
0xFF2050	EMAC_MACR20_MAC_ADDR2_HIGH_REG	MAC Address2 High Register
0xFF2054	EMAC_MACR21_MAC_ADDR2_LOW_REG	MAC Address2 Low Register
0xFF2058	EMAC_MACR22_MAC_ADDR3_HIGH_REG	MAC Address3 High Register
0xFF205C	EMAC_MACR23_MAC_ADDR3_LOW_REG	MAC Address3 Low Register
0xFF2060	EMAC_MACR24_MAC_ADDR4_HIGH_REG	MAC Address4 High Register
0xFF2064	EMAC_MACR25_MAC_ADDR4_LOW_REG	MAC Address4 Low Register
0xFF20D8	EMAC_MACR54_MII_STATUS_REG	MII/RMII Status Register
0xFF2100	EMAC_MMCTRL_REG	MMC Control
0xFF2104	EMAC_MMCRX_INTR_REG	MMC Receive Interrupt
0xFF2108	EMAC_MMCTRANSFER_INTR_REG	MMC Transmit Interrupt
0xFF210C	EMAC_MMCRX_INTR_MASK_REG	MMC Receive Interrupt Mask
0xFF2110	EMAC_MMCTRANSFER_INTR_MASK_REG	MMC Transmit Interrupt Mask
0xFF213C	EMAC_MMCTRANSFERFRAMES_G_B_REG	Number of good and bad unicast frames transmitted.
0xFF2148	EMAC_MMCTRANSFERUNDERFLOW_ERROR_REG	Number of frames aborted due to frame underflow error
0xFF214C	EMAC_MMCTRANSFERCOLISIONS_G_REG	Number of successfully transmitted frames after a single collision in Half-duplex mode
0xFF2150	EMAC_MMCTRANSFERMULTICOLISIONS_G_REG	Number of successfully transmitted frames after more than a single collision in Half-duplex mode

Table 71: Detailed memory Map

Note 33: Test registers are not defined here. Write access to undefined addresses might result in undefined behaviour.

Address	Port	Description
0xFF2158	EMAC_MMCTXLATECOL_REG	Number of frames aborted due to late collision error
0xFF2160	EMAC_MMCTXCARRIERERROR_REG	Number of frames aborted due to carrier sense error (no carrier or loss of carrier).
0xFF2164	EMAC_MMCTXOCTETCOUNT_G_REG	Number of bytes transmitted, exclusive of preamble, in good frames only.
0xFF2168	EMAC_MMCTXFRAZECOUNT_G_REG	Number of good frames transmitted
0xFF2174	EMAC_MMCTXVLANFRAMES_G_REG	Number of good VLAN frames transmitted, exclusive of retried frames.
0xFF2180	EMAC_MMCRXFRAZECOUNT_GB_REG	Number of good and bad frames received
0xFF2188	EMAC_MMCRXOCTETCOUNT_G_REG	Number of bytes received, exclusive of preamble, only in good frames.
0xFF218C	EMAC_MMCRXBROADCASTFRAMES_G_REG	Number of good broadcast frames received
0xFF2190	EMAC_MMCRXMULTICASTFRAMES_G_REG	Number of good multicast frames received
0xFF21A0	EMAC_MMCRXJABBERERROR_REG	Number of giant frames received with length (including CRC) greater than 1,518 bytes (1,522 bytes for VLAN tagged) and with CRC error. If Jumbo Frame mode is enabled, then frames of length greater than 9,018 bytes (9,022 for VLAN tagged) are considered as giant frames.
0xFF21A4	EMAC_MMCRXUNDERSIZE_G_REG	Number of frames received with length less than 64 bytes, without any errors.
0xFF21A8	EMAC_MMCRXOVERSIZE_G_REG	Number of frames received with length greater than the maxsize (1,518 or 1,522 for VLAN tagged frames), without errors
0xFF21C4	EMAC_MMCRXUNICASTFRAMES_G_REG	Number of good unicast frames received
0xFF21D4	EMAC_MMCRXFIFO_OVERFLOW_REG	Number of missed received frames due to FIFO overflow.
0xFF21D8	EMAC_MMCRXVLANFRAMES_GB_REG	Number of good and bad VLAN frames received
0xFF3000	EMAC_DMAR0_BUS_MODE_REG	Bus Mode Register
0xFF3004	EMAC_DMAR1_TX_POLL_DEMAND_REG	Transmit Poll Demand Register
0xFF3008	EMAC_DMAR2_RX_POLL_DEMAND_REG	Receive Poll Demand Register
0xFF300C	EMAC_DMAR3_RX_DESCRIPTOR_LIST_ADDR_REG	Receive Descriptor List Address Register
0xFF3010	EMAC_DMAR4_TX_DESCRIPTOR_LIST_ADDR_REG	Transmit Descriptor List Address Register
0xFF3014	EMAC_DMAR5_STATUS_REG	Status Register
0xFF3018	EMAC_DMAR6_OPERATION_MODE_REG	Operation Mode Register
0xFF301C	EMAC_DMAR7_INT_ENABLE_REG	Interrupt Enable Register
0xFF3020	EMAC_DMAR8_MISSFRAME_BUFOVR_CNT_REG	Missed Frame and Buffer Overflow Counter Register

Table 71: Detailed memory Map

Note 33: Test registers are not defined here. Write access to undefined addresses might result in undefined behaviour.

Address	Port	Description
0xFF3048	EMAC_DMAR18_CUR_HOST_TX_DESCRIPTOR_REG	Current Host Transmit Descriptor Register
0xFF304C	EMAC_DMAR19_CUR_HOST_RX_DESCRIPTOR_REG	Current Host Receive Descriptor Register
0xFF3050	EMAC_DMAR20_CUR_HOST_TX_BU_F_ADDR_REG	Current Host Transmit Buffer Address Register
0xFF3054	EMAC_DMAR21_CUR_HOST_RX_BU_F_ADDR_REG	Current Host Receive Buffer Address Register
0xFF3058		Reserved
0xFF33FF		
0xFF3400	EMAC_RXFIFO_LOW_START	EMAC FIFOs (for testing only).
0xFF3500	EMAC_RXFIFO_HIGH_START	Note 34: Before accessing those addresses the EMAC FIFOs must use the system bus clock. This can be configured by setting the "SW_DRAMODE" bit of CLK_GLOBAL_REG.
0xFF3600	EMAC_TXFIFO_LOW_START	
0xFF3700	EMAC_TXFIFO_HIGH_START	
0xFF4000	CLK_GLOBAL_REG	Selection between PLL and XTAL mode
0xFF4002	CLK_AMBA_REG	HCLK, PCLK, divider and clock gates
0xFF4004	CLK_CODEC1_REG	Codec clock register 1
0xFF4006	CLK_CODEC2_REG	Codec clock register 2
0xFF4008	CLK_CODEC3_REG	Codec clock register 3
0xFF400A	CLK_SPU1_REG	SPU clock register 1
0xFF400C	CLK_SPU2_REG	SPU clock register 2
0xFF400E	CLK_CLASSD1_REG	CLASSD clock register 1
0xFF4010	CLK_CLASSD2_REG	CLASSD clock register 2
0xFF4012	CLK_TIM1_REG	TIM clock register 1
0xFF4014	CLK_TIM2_REG	TIM clock register 2
0xFF4016	CLK_BMC1_REG	BMC clock register 1
0xFF4018	CLK_BMC2_REG	BMC clock register 2
0xFF401A	CLK_BMC3_REG	BMC clock register 3
0xFF401C	CLK_GPIO1_REG	GPIO peripherals clock register 1
0xFF401E	CLK_GPIO2_REG	GPIO peripherals clock register 2
0xFF4020	CLK_GPIO3_REG	GPIO peripherals clock register 3
0xFF4022	CLK_GPIO4_REG	GPIO peripherals clock register 4
0xFF4024	CLK_GPIO5_REG	GPIO peripherals clock register 5
0xFF4026	CLK_GPIO6_REG	GPIO peripherals clock register 6
0xFF4028	CLK_GPIO7_REG	GPIO peripherals clock register 7
0xFF402A	CLK_GPIO8_REG	GPIO peripherals clock register 8
0xFF402C	CLK_AUX1_REG	AUX clock register 1
0xFF402E	CLK_AUX2_REG	AUX clock register 2
0xFF4030	CLK_JOWI_REG	JOWI clock register
0xFF4032	CLK_CLK100A_REG	CLK100 clock division register A
0xFF4034	CLK_CLK100B_REG	CLK100 clock division register B
0xFF4036	CLK_DCDC1_REG	DCDC clock register 1
0xFF4038	CLK_DCDC2_REG	DCDC clock register 2
0xFF403A	CLK_PLL1_CTRL_REG	PLL1 control register

Table 71: Detailed memory Map

Note 33: Test registers are not defined here. Write access to undefined addresses might result in undefined behaviour.

Address	Port	Description
0xFF403C	CLK_PLL1_DIV_REG	PLL1 divider register
0xFF403E	CLK_PLL2_CTRL_REG	PLL2 control register
0xFF4040	CLK_PLL2_DIV_REG	PLL2 divider register
0xFF4042	CLK_FREQ_TRIM_REG	XTAL frequency trimming register
0xFF4044	CLK_XTAL_CTRL_REG	XTAL control register
0xFF4046	CLK_XDIV0_REG	XDIV enable register 0
0xFF4048	CLK_XDIV1_REG	XDIV enable register 1
0xFF404A	CLK_XDIV_VAL_REG	XDIV divider register
0xFF404C	CLK_CDC_CORRECT_REG	CDC clock register
0xFF404E		Reserved
0xFF43FF		
0xFF4400	DMA0_A_STARTL_REG	Start address Low A of DMA channel 0
0xFF4402	DMA0_A_STARTH_REG	Start address High A of DMA channel 0
0xFF4404	DMA0_B_STARTL_REG	Start address Low B of DMA channel 0
0xFF4406	DMA0_B_STARTH_REG	Start address High B of DMA channel 0
0xFF4408	DMA0_INT_REG	DMA receive interrupt register channel 0
0xFF440A	DMA0_LEN_REG	DMA receive length register channel 0
0xFF440C	DMA0_CTRL_REG	Control register for the DMA channel 0
0xFF440E	DMA0_IDX_REG	Internal Index register for the DMA channel 0
0xFF4410	DMA1_A_STARTL_REG	Start address Low A of DMA channel 1
0xFF4412	DMA1_A_STARTH_REG	Start address High A of DMA channel 1
0xFF4414	DMA1_B_STARTL_REG	Start address Low B of DMA channel 1
0xFF4416	DMA1_B_STARTH_REG	Start address High B of DMA channel 1
0xFF4418	DMA1_INT_REG	DMA receive interrupt register channel 1
0xFF441A	DMA1_LEN_REG	DMA receive length register channel 1
0xFF441C	DMA1_CTRL_REG	Control register for the DMA channel 1
0xFF441E	DMA1_IDX_REG	Internal Index register for the DMA channel 1
0xFF4420	DMA2_A_STARTL_REG	Start address Low A of DMA channel 2
0xFF4422	DMA2_A_STARTH_REG	Start address High A of DMA channel 2
0xFF4424	DMA2_B_STARTL_REG	Start address Low B of DMA channel 2
0xFF4426	DMA2_B_STARTH_REG	Start address High B of DMA channel 2
0xFF4428	DMA2_INT_REG	DMA receive interrupt register channel 2
0xFF442A	DMA2_LEN_REG	DMA receive length register channel 2
0xFF442C	DMA2_CTRL_REG	Control register for the DMA channel 2
0xFF442E	DMA2_IDX_REG	Internal Index register for the DMA channel 2
0xFF4430	DMA3_A_STARTL_REG	Start address Low A of DMA channel 3
0xFF4432	DMA3_A_STARTH_REG	Start address High A of DMA channel 3
0xFF4434	DMA3_B_STARTL_REG	Start address Low B of DMA channel 3
0xFF4436	DMA3_B_STARTH_REG	Start address High B of DMA channel 3
0xFF4438	DMA3_INT_REG	DMA receive interrupt register channel 3

Table 71: Detailed memory Map

Note 33: Test registers are not defined here. Write access to undefined addresses might result in undefined behaviour.

Address	Port	Description
0xFF443A	DMA3_LEN_REG	DMA receive length register channel 3
0xFF443C	DMA3_CTRL_REG	Control register for the DMA channel 3
0xFF443E	DMA3_IDX_REG	Internal Index register for the DMA channel 3
0xFF4440		Reserved
0xFF47FF		
0xFF4800	TEST_ENV_REG	CR16C+ boot mode control register
0xFF4802	TEST_CTRL_REG	Test control register
0xFF4804	TEST_CTRL2_REG	Test control register 2
0xFF4810	BANDGAP_REG	Bandgap register
0xFF4812	SUPPLY_CTRL_REG	Power Management control register
0xFF4830	P0_DATA_REG	P0 Data input /out register
0xFF4832	P0_SET_DATA_REG	P0 Set port pins register
0xFF4834	P0_RESET_DATA_REG	P0 Reset port pins register
0xFF4840	P0_00_MODE_REG	P0y Mode Register
0xFF4842	P0_01_MODE_REG	P0y Mode Register
0xFF4844	P0_02_MODE_REG	P0y Mode Register
0xFF4846	P0_03_MODE_REG	P0y Mode Register
0xFF4848	P0_04_MODE_REG	P0y Mode Register
0xFF484A	P0_05_MODE_REG	P0y Mode Register
0xFF484C	P0_06_MODE_REG	P0y Mode Register
0xFF484E	P0_07_MODE_REG	P0y Mode Register
0xFF4850	P0_08_MODE_REG	P0y Mode Register
0xFF4852	P0_09_MODE_REG	P0y Mode Register
0xFF4854	P0_10_MODE_REG	P0y Mode Register
0xFF4856	P0_11_MODE_REG	P0y Mode Register
0xFF4858	P0_12_MODE_REG	P0y Mode Register
0xFF485A	P0_13_MODE_REG	P0y Mode Register
0xFF485C	P0_14_MODE_REG	P0y Mode Register
0xFF485E	P0_15_MODE_REG	P0y Mode Register
0xFF4860	P1_DATA_REG	P1 Data input /out register
0xFF4862	P1_SET_DATA_REG	P1 Set port pins register
0xFF4864	P1_RESET_DATA_REG	P1 Reset port pins register
0xFF4870	P1_00_MODE_REG	P1y Mode Register
0xFF4872	P1_01_MODE_REG	P1y Mode Register
0xFF4874	P1_02_MODE_REG	P1y Mode Register
0xFF4876	P1_03_MODE_REG	P1y Mode Register
0xFF4878	P1_04_MODE_REG	P1y Mode Register
0xFF487A	P1_05_MODE_REG	P1y Mode Register
0xFF487C	P1_06_MODE_REG	P1y Mode Register
0xFF487E	P1_07_MODE_REG	P1y Mode Register
0xFF4880	P1_08_MODE_REG	P1y Mode Register

Table 71: Detailed memory Map

Note 33: Test registers are not defined here. Write access to undefined addresses might result in undefined behaviour.

Address	Port	Description
0xFF4882	P1_09_MODE_REG	P1y Mode Register
0xFF4884	P1_10_MODE_REG	P1y Mode Register
0xFF4886	P1_11_MODE_REG	P1y Mode Register
0xFF4888	P1_12_MODE_REG	P1y Mode Register
0xFF488A	P1_13_MODE_REG	P1y Mode Register
0xFF488C	P1_14_MODE_REG	P1y Mode Register
0xFF488E	P1_15_MODE_REG	P1y Mode Register
0xFF4890	P2_DATA_REG	P2 Data input /out register
0xFF4892	P2_SET_DATA_REG	P2 Set port pins register
0xFF4894	P2_RESET_DATA_REG	P2 Reset port pins register
0xFF48A0	P2_00_MODE_REG	P2y Mode Register
0xFF48A2	P2_01_MODE_REG	P2y Mode Register
0xFF48A4	P2_02_MODE_REG	P2y Mode Register
0xFF48A6	P2_03_MODE_REG	P2y Mode Register
0xFF48A8	P2_04_MODE_REG	P2y Mode Register
0xFF48AA	P2_05_MODE_REG	P2y Mode Register
0xFF48AC	P2_06_MODE_REG	P2y Mode Register
0xFF48AE	P2_07_MODE_REG	P2y Mode Register
0xFF48B0	P2_08_MODE_REG	P2y Mode Register
0xFF48B2	P2_09_MODE_REG	P2y Mode Register
0xFF48B4	P2_10_MODE_REG	P2y Mode Register
0xFF48B6	P2_11_MODE_REG	P2y Mode Register
0xFF48B8	P2_12_MODE_REG	P2y Mode Register
0xFF48BA	P2_13_MODE_REG	P2y Mode Register
0xFF48BC	P2_14_MODE_REG	P2y Mode Register
0xFF48BE	P2_15_MODE_REG	P2y Mode Register
0xFF48C0	P3_DATA_REG	P3 Data input /out register
0xFF48C2	P3_SET_DATA_REG	P3 Set port pins register
0xFF48C4	P3_RESET_DATA_REG	P3 Reset port pins register
0xFF48D0	P3_00_MODE_REG	P3y Mode Register
0xFF48D2	P3_01_MODE_REG	P3y Mode Register
0xFF48D4	P3_02_MODE_REG	P3y Mode Register
0xFF48D6	P3_03_MODE_REG	P3y Mode Register
0xFF48D8	P3_04_MODE_REG	P3y Mode Register
0xFF48DA	P3_05_MODE_REG	P3y Mode Register
0xFF48DC	P3_06_MODE_REG	P3y Mode Register
0xFF48DE	P3_07_MODE_REG	P3y Mode Register
0xFF48E0	P3_08_MODE_REG	P3y Mode Register
0xFF4900	UART_CTRL_REG	UART control register
0xFF4902	UART_RX_TX_REG	UART data transmit/receive register
0xFF4904	UART_CLEAR_TX_INT_REG	UART clear transmit interrupt

Table 71: Detailed memory Map

Note 33: Test registers are not defined here. Write access to undefined addresses might result in undefined behaviour.

Address	Port	Description
0xFF4906	UART_CLEAR_RX_INT_REG	UART clear receive interrupt
0xFF4908	UART_ERROR_REG	UART Parity error register
0xFF4920	ACCESS1_IN_OUT_REG	ACCESS bus 1 receive/transmit register
0xFF4922	ACCESS1_CTRL_REG	ACCESS bus 1 Control register
0xFF4924	ACCESS1_CLEAR_INT_REG	Clear ACCESS bus 1 interrupt
0xFF4930	ACCESS2_IN_OUT_REG	ACCESS bus 2 receive/transmit register
0xFF4932	ACCESS2_CTRL_REG	ACCESS bus 2 Control register
0xFF4934	ACCESS2_CLEAR_INT_REG	Clear ACCESS bus 2 interrupt
0xFF4940	SPI1_CTRL_REG0	SPI 1 control register 0
0xFF4942	SPI1_RX_TX_REG0	SPI 1 RX/TX register 0
0xFF4944	SPI1_RX_TX_REG1	SPI 1 RX/TX register 1
0xFF4946	SPI1_CLEAR_INT_REG	SPI 1 clear interrupt register
0xFF4948	SPI1_CTRL_REG1	SPI 1 control register 1
0xFF4950	SPI2_CTRL_REG0	SPI 2 control register 0
0xFF4952	SPI2_RX_TX_REG0	SPI 2 RX/TX register 0
0xFF4954	SPI2_RX_TX_REG1	SPI 2 RX/TX register 1
0xFF4956	SPI2_CLEAR_INT_REG	SPI 2 clear interrupt register
0xFF4958	SPI2_CTRL_REG1	SPI 2 control register 1
0xFF4960	ADC_CTRL_REG	ADC control register
0xFF4962	ADC_CTRL1_REG	ADC control register 1
0xFF4964	ADC_CLEAR_INT_REG	Clears ADC interrupt if set in ADC_CTRL_REG
0xFF4966	ADC0_REG	ADC0 value
0xFF4968	ADC1_REG	ADC1 value
0xFF4970	TIMER_CTRL_REG	Timers control registers
0xFF4972	TIMER0_ON_REG	Timers 0 on control registers
0xFF4974	TIMER0_RELOAD_M_REG	2 x 16 bits reload value for Timer0
0xFF4976	TIMER0_RELOAD_N_REG	
0xFF4978	TIMER1_RELOAD_M_REG	2 x 16 bits reload value for Timer1
0xFF497A	TIMER1_RELOAD_N_REG	
0xFF4990	TONE_CTRL1_REG	Capture timer control register 1
0xFF4992	TONE_COUNTER1_REG	Capture timer Counter 1
0xFF4994	TONE_LATCH1_REG	Capture timer Latch 1
0xFF4996	TONE_CLEAR_INT1_REG	Clears CT1 interrupt and TONE_LATCH1_REG
0xFF4998	TONE_CTRL2_REG	Capture timer control register 2
0xFF499A	TONE_COUNTER2_REG	Capture timer Counter 2
0xFF499C	TONE_LATCH2_REG	Capture timer Latch 2
0xFF499E	TONE_CLEAR_INT2_REG	Clears CT2 interrupt and TONE_LATCH2_REG

Table 71: Detailed memory Map

Note 33: Test registers are not defined here. Write access to undefined addresses might result in undefined behaviour.

Address	Port	Description
0xFF49B0	KEY_GP_INT_REG	General purpose interrupts for KEYB_INT
0xFF49B2	KEY_BOARD_INT_REG	Keyboard interrupt enable register
0xFF49B4	KEY_DEBOUNCE_REG	Keyboard debounce and auto key generation timer
0xFF49B6	KEY_STATUS_REG	Keyboard interrupt status.
0xFF49D0		Reserved
0xFF4BFF		
0xFF4C00	WATCHDOG_REG	Watchdog preset value.
0xFF4C02		Reserved
0xFF4FFF		
0xFF5000	SET_FREEZE_REG	Freeze watchdog, timer1 and DIP during debugging
0xFF5002	RESET_FREEZE_REG	Release watchdog, timer1 and DIP during debugging after setting in freeze mode
0xFF5004	DEBUG_REG	DEBUG_REG for boot program control and debug control
0xFF5006	CACHE_CTRL_REG	Cache control register
0xFF5008	CACHE_LEN0_REG	Cache length register 0
0xFF500A	CACHE_START0_REG	Cache start register 0, can be used for iCache and dCache
0xFF500C	CACHE_LEN1_REG	Cache length register 1
0xFF500E	CACHE_START1_REG	Cache start register 1, can be used for iCache and dCache
0xFF5010	CACHE_STATUS_REG	Cache status register
0xFF501F		
0xFF5020		Reserved
0xFF53FF		
0xFF5400	SET_INT_PENDING_REG	Set interrupt pending register
0xFF5402	RESET_INT_PENDING_REG	Reset interrupt pending register
0xFF5404	INT0_PRIORITY_REG	Interrupt priority register 0
0xFF5406	INT1_PRIORITY_REG	Interrupt priority register 1
0xFF5408	INT2_PRIORITY_REG	Interrupt priority register 2
0xFF540A	INT3_PRIORITY_REG	Interrupt priority register 2
0xFF540C	PC_START_REG	CR16C+ startup address
0xFF540E		Reserved
0xFF57FF		
0xFF5800	CODEC_MIC_REG	Codec microphone control register
0xFF5802	CODEC_LSR_REG	Codec loudspeaker control register
0xFF5804	CODEC_VREF_REG	Codec vref control register
0xFF5806	CODEC_TONE_REG	Codec CID control register
0xFF5808	CODEC_ADDA_REG	Codec ad/da control register
0xFF580A	CODEC_OFFSET1_REG	Codec offset error and compensation register

Table 71: Detailed memory Map

Note 33: Test registers are not defined here. Write access to undefined addresses might result in undefined behaviour.

Address	Port	Description
0xFF580C	CODEC_TEST_CTRL_REG	Codec test control register codec
0xFF580E	CODEC_OFFSET2_REG	Codec offset compensation register
0xFF5810		Reserved
0xFF5BFF		
0xFF5C00	CLASSD_CTRL_REG	Class D control register
0xFF5C02	CLASSD_CLEAR_INT_REG	Class D Clear interrupt register
0xFF5C04	CLASSD_BUZZER_REG	Class D buzzer register
0xFF5C06	CLASSD_TEST_REG	Class D test register
0xFF5C08	CLASSD_NR_REG	Class D noise reduction register
0xFF5C0A		Reserved
0xFF5FFF		
0xFF6000	DIP_STACK_REG	DIP Stack pointer. (read only). The DIP stack is 4 deep
0xFF6002	DIP_PC_REG	DIP program counter
0xFF6004	DIP_STATUS_REG	DIP Status register,
0xFF6006	DIP_CTRL_REG	DIP Control register1
0xFF6008	DIP_STATUS1_REG	DIP Status register1,
0xFF600A	DIP_CTRL1_REG	DIP Control register, clears DIP_INT if read
0xFF600C	DIP_SLOT_NUMBER_REG	DIP slot number register, returns the current slot number
0xFF600E	DIP_CTRL2_REG	DIP Control register 2 (debug status information)
0xFF6010	DIP_USB_PHASE_REG	Phase between SLOTZERO and USB SOF
0xFF6012	DIP_MOD_SEL_REG	DIP Modulo counters selection
0xFF6014	DIP_MOD_VAL_REG	DIP Modulo values selection
0xFF6016	DIP_DC01_REG	DIP MultiFrame control
0xFF6018	DIP_DC23_REG	DIP MultiFrame control
0xFF601A		Reserved
0xFF63FF		
0xFF6400	BMC_CTRL_REG	BMC control register
0xFF6402	BMC_CTRL2_REG	BMC control register 2
0xFF6404	BMC_TX_SFIELDL_REG	BMC Tx S field register Low
0xFF6406	BMC_TX_SFIELDH_REG	BMC Tx S field register High
0xFF6408	BMC_RX_SFIELDL_REG	BMC Rx S field register Low
0xFF640A	BMC_RX_SFIELDH_REG	BMC Rx S field register High
0xFF640C		Reserved
0xFF67FF		
0xFF6800		Reserved
0xFF6BFF		
0xFF6C00	CCU_CRC_LOW_REG	CCU MSB result
0xFF6C02	CCU_CRC_HIGH_REG	CCU LSB result
0xFF6C04	CCU_IN_REG	CCU Input

Table 71: Detailed memory Map

Note 33: Test registers are not defined here. Write access to undefined addresses might result in undefined behaviour.

Address	Port	Description
0xFF6C06	CCU_MODE_REG	CCU mode
0xFF6C08		Reserved
0xFF6FFF		
0xFF7000	GPRG_R0_REG	General Programming Register 0
0xFF7002	GPRG_R1_REG	General Programming Register 1
0xFF7004		Reserved
0xFF73FF		
0xFF7400		Free
0xFFFFAFF		
0xFFFFB00		Reserved for Chip id registers
0xFFFFBF7		
0xFFFFBF8	CHIP_ID1_REG	Chip identification register 1
0xFFFFBF9	CHIP_ID2_REG	Chip identification register 2
0xFFFFBFA	CHIP_ID3_REG	Chip identification register 3
0xFFFFBFB	CHIP_MEM_SIZE_REG	Chip memory size register
0xFFFFBFC	CHIP_REVISION_REG	Chip revision registers (Corresponds to Chip Marking)
0xFFFFBFD	CHIP_TEST1_REG	Chip test register
0xFFFFBFE	CHIP_TEST2_REG	Chip test register
0xFFFFBFF		Reserved
0xFFFFC00	INT_ACK_CR16_START	CR16C+ interrupt acknowledge
0xFFFFFFF	INT_ACK_CR16_END	
DIP RAM		
0x1000000	DIP_RAM_START	DIP Sequencer RAM (256 words)
0x10001FF	DIP_RAM_END	
0x1000200	DIP_RAM_2_START	Sequencer RAM (256 words) to be addressed with new <JMPF>,<BR_B1> DiP commands
0x10003FF	DIP_RAM_2_END	
0x1000400		Reserved 63k bytes
0x100FFFF		
Gen2DSP1 Shared ROM		
0x1010000	SHARED_ROM1_START	Shared ROM 1 (2k bytes)
0x10107FF	SHARED_ROM1_END	
0x1010800		Reserved
0x101FF4F		
0x101FF50		Reserved
0x101FF7F		
Gen2DSP1 and Gen2DSP2 common registers		
0x101FF80	DSP_MAIN_SYNC0_REG	DSP main counter outputs selection register 0
0x101FF82	DSP_MAIN_SYNC1_REG	DSP main counter outputs selection register 1
0x101FF84	DSP_MAIN_CNT_REG	DSP main counter and reload register
0x101FF86	DSP_ADC1S_REG	ADC1 Input 2's Complement register
0x101FF88	DSP_ADC0S_REG	ADC0 Input 2's Complement register
0x101FF8A	DSP_CLASSD_REG	CLASSD Output output data register

Table 71: Detailed memory Map

Note 33: Test registers are not defined here. Write access to undefined addresses might result in undefined behaviour.

Address	Port	Description
0x101FF8C	DSP_CODEC_MIC_GAIN_REG	CODEC MIC GAIN register
0x101FF8E	DSP_CODEC_OUT_REG	CODEC DATA register
0x101FF90	DSP_CODEC_IN_REG	CODEC DATA register
0x101FF92	DSP_RAM_OUT0_REG	Shared RAM 1 or 2 output register 0
0x101FF94	DSP_RAM_OUT1_REG	Shared RAM 1 or 2 output register 1
0x101FF96	DSP_RAM_OUT2_REG	Shared RAM 1 or 2 output register 2
0x101FF98	DSP_RAM_OUT3_REG	Shared RAM 1 or 2 output register 3
0x101FF9A	DSP_RAM_IN0_REG	Shared RAM 1 or 2 input register 0
0x101FF9C	DSP_RAM_IN1_REG	Shared RAM 1 or 2 input register 1
0x101FF9E	DSP_RAM_IN2_REG	Shared RAM 1 or 2 input register 2
0x101FFA0	DSP_RAM_IN3_REG	Shared RAM 1 or 2 input register 3
0x101FFA2	DSP_ZCROSS1_OUT_REG	ZERO CROSSING 1 output register
0x101FFA4	DSP_ZCROSS2_OUT_REG	ZERO CROSSING 2 output register
0x101FFA6	DSP_PCM_OUT0_REG	PCM channel 0 output register
0x101FFA8	DSP_PCM_OUT1_REG	PCM channel 1 output register
0x101FFAA	DSP_PCM_OUT2_REG	PCM channel 2 output register
0x101FFAC	DSP_PCM_OUT3_REG	PCM channel 3 output register
0x101FFAE	DSP_PCM_IN0_REG	PCM channel 0 input register
0x101FFB0	DSP_PCM_IN1_REG	PCM channel 1 input register
0x101FFB2	DSP_PCM_IN2_REG	PCM channel 2 input register
0x101FFB4	DSP_PCM_IN3_REG	PCM channel 3 input register
0x101FFB6	DSP_PCM_CTRL_REG	PCM Control register
0x101FFB8	DSP_PHASE_INFO_REG	Phase information between PCM FSC 8/16/32 and main counter 8/16/32 kHz
0x101FFBA	DSP_VQI_REG	BMC VQI register
0x101FFBC	DSP_MAIN_CTRL_REG	DSP Main counter control and preset value
0x101FFBE	DSP_CLASSD_BUZZOFF_REG	CLASSD Buzzer on/off controller, bit 15
0x101FFC0		Reserved
0x101FFCF		
Gen2DSP1 registers (shared with Gen2DSP2)		
0x101FFD0	DSP1_CTRL_REG	DSP1 control register
0x101FFD2	DSP1_PC_REG	DSP1 Programma counter
0x101FFD4	DSP1_PC_START_REG	DSP1 Programma counter start
0x101FFD6	DSP1_IRQ_START_REG	DSP1 Interrupt vector start
0x101FFD8	DSP1_INT_REG	DSP1 to CR16C+ interrupt vector
0x101FFDA	DSP1_INT_MASK_REG	DSP1 to CR16C+ interrupt vector masks
0x101FFDC	DSP1_INT_PRIO1_REG	DSP1 Interrupt source mux 1 register
0x101FFDE	DSP1_INT_PRIO2_REG	DSP1 Interrupt source mux 2 register
0x101FFE0	DSP1_OVERFLOW_REG	DSP1 to CR16C+ overflow register
0x101FFE2	DSP1_JTBL_START_REG	DSP1 jump table start address
0x101FFE4		Reserved
0x101FFEE		
0x101FFF0	DBG1_IREG	
0x101FFF2		Reserved

Table 71: Detailed memory Map

Note 33: Test registers are not defined here. Write access to undefined addresses might result in undefined behaviour.

Address	Port	Description
0x101FFF4	DBG1_INOUT_REG_LSW	DSP1 DEBUG data register (32 bits)
0x101FFF6	DBG1_INOUT_REG_MSW	
0x101FFF8		Reserved
0x101FFFF		
Gen2DSP2 Shared ROM		
0x1020000	SHARED_ROM2_START	Shared ROM2 (64k-176 bytes)
0x102FF4F	SHARED_ROM2_END	
0x102FF50		Reserved
0x102FFCF		
Gen2DSP2 registers (shared with Gen2DSP1)		
0x102FFD0	DSP2_CTRL_REG	DSP2 control register
0x102FFD2	DSP2_PC_REG	DSP2 Programma counter
0x102FFD4	DSP2_PC_START_REG	DSP2 Programma counter start
0x102FFD6	DSP2_IRQ_START_REG	DSP2 Interrupt vector start
0x102FFD8	DSP2_INT_REG	DSP2 to CR16C+ interrupt vector
0x102FFDA	DSP2_INT_MASK_REG	DSP2 to CR16C+ interrupt vector masks
0x102FFDC	DSP2_INT_PRIO1_REG	DSP2 Interrupt source mux 1 register
0x102FFDE	DSP2_INT_PRIO2_REG	DSP2 Interrupt source mux 2 register
0x102FFE0	DSP2_OVERFLOW_REG	DSP2 to CR16C+ overflow register
0x102FFE2	DSP2_JTBL_START_REG	DSP2 jump table start address
0x102FFE4		Reserved
0x102FFEE		
0x102FFF0	DBG2_IREG	DSP2 JTAG instruction register
0x102FFF2		Reserved
0x102FFF4	DBG2_INOUT_REG_LSW	DSP2 DEBUG data register (32 bits)
0x102FFF6	DBG2_INOUT_REG_MSW	
0x102FFF8		Reserved
0x102FFFF		
Gen2DSP1 and Gen2DSP2 micro code RAMs		
0x1030000	DSP_MC_RAM1_START	Gen2DSP MicroCode RAM 1 20k bytes
0x1034FFF	DSP_MC_RAM1_END	
0x1035000		Reserved
0x1037FFF		
0x1038000	DSP_MC_RAM2_START	Gen2DSP MicroCode RAM 2 16k bytes
0x103BFFF	DSP_MC_RAM2_END	
0x103C000		Reserved
0x103FFFF		
Gen2DSP1 and Gen2DSP2 micro code ROMs		
0x1040000	DSP_MC_ROM1_START	Gen2DSP MicroCode ROM1 32k bytes
0x1047FFF	DSP_MC_ROM1_END	
0x1048000		96kbyte Reserved
0x105FFFF		
0x1060000	DSP_MC_ROM2_START	Gen2DSP MicroCode ROM 2 112k bytes
0x107BFFF	DSP_MC_ROM2_END	

Table 71: Detailed memory Map

Note 33: Test registers are not defined here. Write access to undefined addresses might result in undefined behaviour.

Address	Port	Description
0x107C000		Reserved
0x107FFFF		
0x1080000		Free
0xFFFFFFFFEF		

Confidential

37.0 Registers description detailed

Table 72: ACCESS1_IN_OUT_REG, ACCESS2_IN_OUT_REG (0xFF4920, 0xFF4930)

Bit	Mode	Symbol	Description	Reset
15-8	-	-	Reserved	0
7-0	R/W	ACCESS_DATA	Read: ACCESS bus x input register Write: ACCESS bus x output register	0

Table 73: ACCESS1_CTRL_REG, ACCESS2_CTRL_REG (0xFF4922, 0xFF4932)

Bit	Mode	Symbol	Description	Reset
15-11	-	-	Reserved	0
10-9	R/W	SCK_SEL	00 = 100 kHz mode selected 01 = 400 kHz mode selected. 1x = 1.152 MHz mode selected.	0
8	R/W	SCK_NUM	0 = 8 bits without ACK bits are generated, SDAx push pull 1 = 8 bits with ACK bit are generated. SDAx open drain, Pull up required.	0
7	R/W	SDA_OD	0 = SDAx push pull 1 = SDAx open drain, Pull up required	1
6	R/W	SCL_OD	0 = SCLx push pull 1 = SCLx open drain, Pull up required, allows clock stretching	0
5	R/W	EN_ACCESS_INT	0 = Disable ACCESS x bus interrupt 1 = Enable ACCESS x bus interrupt	0
4	R	ACCESS_INT	1 = ACCESS Bus x generated the interrupt. Must be reset by SW by writing to ACCESSx_CLEAR_INT_REG.	0
3	R/W	SDA_VAL	Output value of SDAx pin if EN_ACCESS_BUSx= 0	1
2	R/W	SCL_VAL	Output value of SCLx pin if EN_ACCESS_BUSx = 0	1
1	R/W	ACKn	In transmit mode: (Read only) 0 = ACK received 1 = No ACK received In receive mode: (Write only) 0 = transmit ACK on access bus 1 = transmit No ACK.	0
0	R/W	EN_ACCESS_BUS	0 = Disable ACCESS Bus x, power down mode 1 = Enable ACCESS Bus x	0

Table 74: ACCESS1_CLEAR_INT_REG, ACCESS2_CLEAR_INT_REG (0xFF4924, 0xFF4934)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	ACCESS_CLEAR_INT	Writing any value to this register will clear the ACCESS bus x interrupt. Reading returns 0	0

Table 75: ADC_CTRL_REG (0xFF4960)

Bit	Mode	Symbol	Description	Reset
15	R/W	ADC2_PR_DIS	0 = ADC2 pad protection enabled 1 = ADC2 pad protection disabled.	0
14	R/W	ADC1_PR_DIS	0 = ADC1 pad protection enabled 1 = ADC1 pad protection disabled.	0
13	R/W	ADC0_PR_DIS	0 = ADC0 pad protection enabled 1 = ADC0 pad protection disabled.	0
12	R/W	ADC_MINT	0 = Disable (mask) ADC_INT. 1 = Enable ADC_INT to ICU	0
11	R	ADC_INT	1 = AD conversion ready and has generated an interrupt. Must be cleared by writing any value to ADC_CLEAR_INT_REG Will not be set in automatic mode.	0
10-7	R/W	ADC_TEST	Test bits Must be set to 0	0
6-3	R/W	ADC_IN_3_0	Primary ADC input selection (Manual and automatic mode) 0000 = ADC0 input 0001 = ADC1 input 0010 = ADC2 input (DIP <B_RC> RC2[7] must be also set to '1') 0011 = Caller ID output as input 0100 = Codec headset detection. 0110 or 0101= SUPPLY 0111 =Temperature sensor 1000 = ADC3 input 1xxx = Reserved	0
2	R/W	ADC_ALT	0 = In automatic ADC mode ADC samples are always stored in ADC0_REG. 1 = In automatic ADC mode ADC samples are stored alternatively in ADC0_REG and ADC1_REG.	0
1	R/W	ADC_AUTO	0 = Manual ADC mode, by setting ADC_L_REG[ADC_START]. ADC result is stored in ADC0_REG 1 = Automatic ADC mode triggered by DSP_MAIN_SYNC1_REG[ADC_SYNC]. ADC_START read value must be 0 before auto mode is enabled. ADC_START write is ignored in auto mode ADC results is stored in ADC0_REG (ADC_ALT=0) or alternating ADC0_REG and ADC1_REG.	0
0	R/W	ADC_START	0: ADC conversion ready. 1: If a 1 is written, the ADC starts conversion. After conversion this bit will be set to 0 and the ADC_INT bit will be set.	0

Table 76: ADC_CTRL1_REG (0xFF4962)

Bit	Mode	Symbol	Description	Reset
15-4	-	-	Reserved	0
3-0	R/W	ADC_IN_3_0_1	Secondary ADC input selection in automatic ADC conversion Bit definitions are the same as ADC_IN_3_0	0

Table 77: ADC_CLEAR_INT_REG (0xFF4964)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	ADC_CLEAR_INT	Writing any value to this register will clear the ADC_INT interrupt. Reading returns 0.	0

Table 78: ADC0_REG (0xFF4966)

Bit	Mode	Symbol	Description	Reset
15-10	-	-	Reserved	0
9-0	R/W	ADC0_VAL	Read: Returns the 10 bits linear value of the last ADC conversion. (Note 35) . In automatic ADC mode returns the first automatic ADC conversion Write: ADC_DAC value in test mode. Output of ADC_DAC can be read on ADC0 if selected.	0

Note 35: ADC input values may never be more than AVD2, otherwise other ADC measurements will become inaccurate due to internal leakage.

Table 79: ADC1_REG (0xFF4968)

Bit	Mode	Symbol	Description	Reset
15-10	-	-	Reserved	0
9-0	R	ADC1_VAL	Read: Returns the 10 bits linear value of the second automatic ADC conversion	0

Table 80: BANDGAP_REG (0xFF4810)

Bit	Mode	Symbol	Description	Reset (Note 36)
15-9	-	-	Reserved	0
8-6	R/W	BANDGAP_I	Internal reference current adjust. Setting these bits will not affect the bandgap voltage. 100 = +40 % 101 = +30 % 110 = +20% 111 = +10% 000 = +0 % <- default value may not be changed. 001 = -10% 010 = -20% 011 = -30%	000
5-4	R/W	BANDGAP_VIT	00 = <- default value may not be changed 10 = Voltage and current -5.6 % <- for test purposes only ! 01 = Voltage and current +5.6 % <- for test purposes only!	00
3-0	R/W	BANDGAP_VI	Bandgap reference voltage and current adjust to trim AVD2. These bits must be set before the current bits 8-6 because it affects the reference currents. 0000 = -5.6 % 0001 = -4.9 % 0010 = -4.2 % 0011 = -3.5 % 0100 = -2.8 % 0101 = -2.1 % 0110 = -1.4 % 0111 = -0.7 % 1000 = midlevel <- default value. 1001 = +0.7 % 1010 = +1.4 % 1011 = +2.1 % 1100 = +2.8 % 1101 = +3.5 % 1110 = +4.2 % 1111 = +4.9 %	0x8

Note 36: The register is only reset with RSTn = 0, not on a SW_RESET

Note 37: The reset value of DC_VOUT and LDO1_LEVEL is determined by the level of AD[3-1] when RSTn is low and this level is latched on the rising edge of RSTn TEST_ENV_REG (0xFF4800) (table 325, page 281) bit 3-1. AD[3-1] are input when RSTn is low. An internal pull-up resistor to VDD gives a '1', and external 100k pull-down resistor to VSS gives a '0'. Upon a hardware reset, the LDO1_LEVEL and DC_VOUT use the latched values of TEST_ENV_REG[AD3-1], upon a SW_RESET, the LDO1_LEVEL and DC_VOUT value do not change. Examples:

- For 1.8V, AD3, AD2, AD1 no resistors.
- For 2.5V, AD3 pull-down to VSS, AD1, AD0 no resistors
- For 3.33V, AD2, AD1 no resistors, AD0 pull-down to VSS

Note 38: The register is only reset with RSTn = 0, not on a SW_RESET

Confidential

Table 81: SUPPLY_CTRL_REG (0xFF4812)

Bit	Mode	Symbol	Description	Reset (Note 38)
15	R/W	DC_CLK_SEL	DC/DC converter switching clock on pin DC_CTRL 0 = 600kHz free-running oscillator 1 = derived from XTAL, frequency set by DC_CLK_SEL	0
14-13	R/W	DC_FREQ	DC/DC converter switching clock on pin DC_CTRL. In all cases the high time of this signal is 192ns. 00 = 2.604 MHz 01 = 1.152 MHz 10 = 798 KHz 11 = 576 KHz	10
12	R/W	DC_IMAX	Current limit level (@ 0.1 Ohm sense resistor) 0 = ~800mA 1 = ~250mA	0
11	R/W	DC_HYST	0 = continuous mode. 1 = hysteresis mode	1
10-8	R/W	DC_VOUT	DC/DC output voltage DC_VOUT (sensed on DC_SENSE) 000 = 1.29V (not tested) 001 = 2.2V 010 = 2.5V 011 = 2.7V 100 = 3.0V 101 = 3.3V 110 = 3.6V 111 = 2.0V	(Note 37)
7	R/W	DC_MODE_1	DCDC converter. Extra RC pole in the feedback loop (to be used in combination with large ELCO / large inductor) 0 = suitable for 10uF ceramic capacitor + small inductor (3.3 or 4.7uH) 1 = Same as SC1443x (large ELCO + large inductor)	0
6	R/W	DC_MODE_0	DCDC converter loop gain 0 = default setting (needed for small inductor and ceramic capacitor), 1/4 of the SC1443x loop gain 1 = same as SC1443x	0
5-3	R/W	LDO1_LEVEL	LDO1 output voltage sensed on VDDIO AD[3-1] levels if RSTn=0: 000 = 1.8V 001 = 1.8V 010 = 2.0V 011 = 2.5V 100 = 2.5V 101 = 3.0V 110 = 3.33V 111 = 1.8V	(Note 37)
2	R/W	DC_ON	0 = DC/DC convertor disable 1 = DC/DC convertor enable and clock switched to DC_CTRL pin. Not reset by SW reset only HW reset	1
1	R/W	LDO1_ON	1 = LDO1 is enabled.	1
0	R/W	REG_ON	1 = LDO2 and DC/DC stay on after power-on procedure (See Figure 10 and Figure 12) Not reset by SW reset	0

Note 39: The reset value of DC_VOUT and LDO1_LEVEL is determined by the level of AD[3-1] when RSTn is low and this level is latched on the rising edge of RSTn in TEST_ENV_REG (0xFF4800) (table 325, page 281) bit 3-1. AD[3-1] are input when RSTn is low. An internal pull-up resistor to VDD gives a '1', and external 100k pull-down resistor to VSS gives a '0'. Upon a hardware reset, the LDO1_LEVEL and

DC_VOUT use the latched values of TEST_ENV_REG[AD3-1], upon a SW_RESET, the LDO1_LEVEL and DC_VOUT value do not change. Examples:

- For 1.8V, AD3, AD2, AD1 no resistors.
- For 2.5V, AD3 pull-down to VSS, AD1, AD0 no resistors
- For 3.27V, AD3, AD2 no resistors, AD1 pull-down to VSS

Note 40: The register is only reset with RSTn = 0, not on a SW_RESET

Table 82: BMC_CTRL_REG (0xFF6400)

Bit	Mode	Symbol	Description	Reset
15-14	-	-	Reserved	0
13	R/W	BCNT_INH	0 = Normal operation B_BR2, B_BT2, B_BRFD and B_BTFM instructions 1 = Disable automatic bitcounter in B_BR2, B_BT2, B_BRFD and B_BTFM instructions. (X-CRC not transmitted)	0
12-9	-	-	Reserved	0
8	R/W	SIO_PD	0 = SIO pull-down resistor is automatically enabled if SIO is input and automatically disabled if SIO output. 1 = SIO pull-down resistor disabled (also for test purposes)	0
7	-	-	Reserved	0
6	R/W	RSSI_TDO	Must be kept to 0. TDO(ana) may never be connected to RSSI pin	0
5	R/W	RSSI_RANGE	RSSI input range selection 0 = 70mV-1.16V 1 = 0-1.6V	0
4	R/W	GAUSS_REF	Gaussian output midlevel selection 0 = 900 mV midlevel 1 = reserved	0
3-0	-	-	Reserved	0

Table 83: BMC_CTRL2_REG (0xFF6402)

Bit	Mode	Symbol	Description	Reset
15-8	-	-	Reserved	0
7-3	R/W	RCV_CTL	For best clock recovery performance RCV_CTL bits shall be set to "11110" and B_RC[DPLL] shall be set to "000"	0
2-0	R/W	DAC_TEST	000 = Normal situation	0

Table 84: BMC_TX_SFIELDL_REG (0xFF6404)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	TX_SFIELDL	Transmitted S field register. If the <B_ST2> operand is unequal to 0 one or more bytes are transmitted from this register. For more information see AN-D-140	0xE98A

Table 85: BMC_TX_SFIELDH_REG (0xFF6406)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	TX_SFIELDH	Bits 31-16 of BMC_TX_SFIELDL_REG.	0xAAAA

Table 86: BMC_RX_SFIELDL_REG (0xFF6408)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	RX_SFIELDL	Received S field pattern bits 15-0.	0x9755

Table 87: BMC_RX_SFIELDH_REG (0xFF640A)

Bit	Mode	Symbol	Description	Reset
15-8	-	-	Reserved	0
7-0	R/W	RX_SFIELDH	Bits 23-16 of BMC_RX_SFIELDH_REG If <B_SR> operand bit 6 is 1, register RX_SFIELDH_REG[23-22] are overruled by <B_SR> operand[1-0].	0x51

Table 88: CACHE_CTRL_REG (0xFF5006)

Bit	Mode	Symbol	Description	Reset
15-11	-	-	Reserved	0
10	R/W	DCACHE_B_SIZE	0 = dCache burst read/write size = 16 bytes. Must be kept 0.	0
9	-	-	Reserved	0
8	R/W	ICACHE_B_SIZE	0 = iCache burst read/write size = 16 bytes. Must be kept 0.	0
7	-	-	Reserved	-
6	R/W	CACHE_LOCK	0 = lock per cache 1 = lock per line	0
5-4	R/W	CACHE_MODE	0 = unified, no predefined regions for iCache and dCache. 1 = half iCache, half dCache 2, 3 = undefined	0
3	-	-	Reserved	0
2-0	R/W	CACHE_SIZE	0,1,2,3 = Cache disabled 4 = 4kByte 5 = 8kByte 6 = 16kByte 7 = undefined Note: Dynamically changing the Cache size is not supported. Cache size must be programmed at start up.	0

Table 89: CACHE_LEN0_REG (0xFF5008)

Bit	Mode	Symbol	Description	Reset
15-9	-	-	Reserved	0
8-0	R/W	CACHE_LEN0	Length of iCachable or dCachable memory N*64kbyte. N= 0 to 512 (Max of 32Mbyte)	0

Table 90: CACHE_START0_REG (0xFF500A)

Bit	Mode	Symbol	Description	Reset
15-9	-	-	Reserved	0
8-0	R/W	CACHE_START0	Start address of dCachable or iCachable memory N*64kbyte. N= 0 to 512 (Max of 32Mbyte)	0

Table 91: CACHE_LEN1_REG (0xFF500C)

Bit	Mode	Symbol	Description	Reset
15-9	-	-	Reserved	0
8-0	R/W	CACHE_LEN1	Length of iCachable or dCachable memory N*64kbyte. N= 0 to 512 (Max of 32Mbyte)	0

Table 92: CACHE_START1_REG (0xFF500E)

Bit	Mode	Symbol	Description	Reset
15-9	-	-	Reserved	0
8-0	R/W	CACHE_START1	Start address of iCachable or dCachable memory N*64kbyte. N= 0 to 512 (Max of 32Mbyte)	0

Table 93: CACHE_STATUS_REG (0xFF5010)

Bit	Mode	Symbol	Description	Reset
15-14	-	-	Reserved	0
13	R	CACHE_TOUCH	0 = Normal operation 1 = Cache buffer written by source other than cache controller	0
12	R	CACHE_TOUCH_TOGGLE	This bit will change value each time the cache buffer is written by a source other than cache control.	0
11	R	DCACHE_HIT	0 = Data cache miss (Note 41) 1 = Data cache hit	0
10	R	ICACHE_HIT	0 = Instruction cache miss 1 = Instruction cache hit	0
9-0	-	-	Reserved	0

Note 41: The (data or instruction) cache hit bit will be set when the CR16C+ performs a data or instruction read access in cache area and the associated cache line is in cache memory. The (data or instruction) cache hit bit will be reset when the CR16C+ performs a data or instruction read access in cache area and the associated cache line is NOT in cache memory.

Table 94: CCU_CRC_LOW_REG (0xFF6C00)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	CCU_CRC_LOW	Calculated CRC register bits 15-0. Must be preset to the correct start value	0000

Table 95: CCU_CRC_HIGH_REG (0xFF6C02)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	CCU_CRC_HIGH	Calculated CRC register bits 31-0. Must be preset to the correct start value	0000

Table 96: CCU_IN_REG (0xFF6C04)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	CCU_IN	CRC input register	0000

Table 97: CCU_MODE_REG (0xFF6C06)

Bit	Mode	Symbol	Description	Reset
15-4	-	-	Reserved	0
3	R/W	CCU_BYTE_SWAP	Byte Swap in Word: 0 = no swap 1 = Swap: 15-8, 7-0 -> 7-0, 15-8	0
2	R/W	CCU_BIT_SWAP	Bit Swap in bytes: 0 = no swap 1 = Swap: 7-0 -> 0-7	0
1-0	R/W	CCU_MODE	00 = Calculated CRC-16 01 = Calculate CRC-32 10 = Calculate 1's complement.	00

Table 98: GPRG_R0_REG (0xFF7000)

Bit	Mode	Symbol	Description	Reset
15-14	R/W	PINDRV_AD23_AD13	Control the drive strength of the 11 pins: AD23-AD13 (aka P1[6:0] and P2[10:7]) (4,8,12,16 mA drive selected, refer to Table 355 for a detailed specification)	00
13-12	R/W	PINDRV_AD12_AD1	Control the drive strength of the 12 pins: AD12-AD1 (4,8,12,16 mA drive selected, refer to Table 355 for a detailed specification)	00
11-10	R/W	PINDRV_EBICTRL	Control the drive strength of the 8 pins: WRn, RDn, SDLDQM, SDUDQM, SDRASn, SDCASn, SDCKE, SDCLK (aka WRn, RDn and P1[12:7]) (4,8,12,16 mA drive selected, refer to Table 355 for a detailed specification)	00
9	R/W	PLL2_CLK_SEL_MATRIX	Clearing this bit to 0, the I/O Matrix forwards to the selected GPIOs the PLL2_CLK (refer to Table 3). Setting this bit to 1, the I/O Matrix forwards to the selected GPIOs the PLL2_CLK/2.	0
8	R/W	EMAC_REF_CLK_OE	EMAC REF_CLK pin output enable. If clear the REF_CLK pin is output. If set the REF_CLK pin is input.	0
7-6	R/W	PINDRV_QSPI	Control the drive strength of the 6 pins: QSPI_SCK, QSPI_IO0, QSPI_IO1, QSPI_IO2, QSPI_IO3, QSPI_CS (aka P0[14:11], P0[6], P0[3]) (4,8,12,16 mA drive selected, refer to Table 355 for a detailed specification)	00
5	-	-	Reserved	0
4-0	R/W	EBI_CS_POL	External Bus Memory Interface ACS Polarity Control. EBI_CS_POL[i] = 0 : ACS[i] normal polarity (active low) EBI_CS_POL[i] = 1 : ACS[i] reversed polarity (active high)	0

Table 99: GPRG_R1_REG (0xFF7002)

Bit	Mode	Symbol	Description	Reset
15-14	R/W	PINDRV_DAB15_DA8	Control the drive strength of the 8 pins: DAB15-DAB8 (4,8,12,16 mA drive selected, refer to Table 355 for a detailed specification)	00
13-12	R/W	PINDRV_DAB7_DA0	Control the drive strength of the 8 pins: DAB7-DAB0 (4,8,12,16 mA drive selected, refer to Table 355 for a detailed specification)	00
11-0	R/W	QSPI_A_HADDR	Select the QSPI Auto mode page number. It actually specifies/replaces the HADDR[23:12] bits.	0

Table 100: CHIP_ID1_REG (0xFFFFBF8)

Bit	Mode	Symbol	Description	Reset
7-0	R	CHIP_ID1	First character of device type in ASCII. CHIP_ID1_REG, CHIP_ID2_REG, CHIP_ID3_REG form the three character device type "452"	0x34

Table 101: CHIP_ID2_REG (0xFFFFBF9)

Bit	Mode	Symbol	Description	Reset
7-0	R	CHIP_ID2	Second character of device type in ASCII. CHIP_ID1_REG, CHIP_ID2_REG, CHIP_ID3_REG form the three character device type "452"	0x35

Table 102: CHIP_ID3_REG (0xFFFFBFA)

Bit	Mode	Symbol	Description	Reset
7-0	R	CHIP_ID3	Third character of device type in ASCII. CHIP_ID1_REG, CHIP_ID2_REG, CHIP_ID3_REG form the three character device type "452"	0x32

Table 103: CHIP_MEM_SIZE_REG (0xFFFFBFB)

Bit	Mode	Symbol	Description	Reset
7-0	R	MEM_SIZE_ID	Program ROM/FLASH size. 0= No ROM	0

Table 104: CHIP_REVISION_REG (0xFFFFBFC)

Bit	Mode	Symbol	Description	Reset
7-4	R	REVISION_ID	Chip version, corresponds with type number 0x1 = 'A', 0x2 = 'B', etc	1
3-0	-	-	Reserved	0

Table 105: CHIP_TEST1_REG (0xFFFFBFD)

Bit	Mode	Symbol	Description	Reset
7-4	R	CHIP_TEST1H		1
3-0	R	CHIP_TEST1L		0

Table 106: CHIP_TEST2_REG (0xFFFFBFE)

Bit	Mode	Symbol	Description	Reset
7-0	R	CHIP_TEST2		2

Table 107: CLASSD_BUZZER_REG (0xFF5C04)

Bit	Mode	Symbol	Description	Reset
15-5	-	-	Reserved	0
4	R/W	CLASSD_BUZ_MODE	0 = CLASSD normal audio path enable 1 = CLASSD Buzzer path enable. To enable the buzzer amplifier DSP_CLASSD_BUZZOFF_REG (0x101FFBE) bit 15 BUZZOFF must be set to '0'. To switch off the buzzer without audible side effects, bit 15 must be set to '1' at the end of a tone.	0
3-0	R/W	CLASSD_BUZ_GAIN	Amplitude control of the buzzer signal, relative to 1W into 4 Ohm: 1111 = +0dB 1110 = -2dB 1101 = -4dB 1100 = -6dB 1011 = -8dB 1010 = -10dB 1001 = -12dB 1000 = -18dB 0111 = -24dB 0110 = -30dB 0101 = -36dB 0100 = -42dB 0011 = -48dB 0010 = -54dB 0001 = -60dB 0000 = mute	0000

Table 108: CLASSD_CTRL_REG (0xFF5C00)

Bit	Mode	Symbol	Description	Reset
15-14	-	-	Reserved	00
13	R/W	CLASSD_MINT	0 = Disable (mask) CLASSD_INT_BIT to CT_CLASSD_INT 1 = Enable CLASSD_INT_BIT to CT_CLASSD_INT.	0
12	R/W	CLASSD_MODE	0 = Normal situation	0
11-10	R/W	CLASSD_DITH_A	Dithering in analog part 00 = Adaptive dithering 01 = Maximum dithering 10 = Small dithering 11 = No dithering	11
9-8	R/W	CLASSD_DITH_D	Dithering in digital part 00 = Adaptive dithering 01 = Maximum dithering 10 = Small dithering 11 = No dithering	00
7	R	CLASSD_INT_BIT	1 = the CLASSD amplifier has clipped (CLASSD_INT_BIT) Must be reset by SW by writing to CLASSD_CLEAR_INT_REG.	0
6-4	R/W	CLASSD_CLIP	000 = no clipping detection 001 = CLASSD_INT after 32 clipping occurrences 010 = CLASSD_INT after 64 clipping occurrences 011 = CLASSD_INT after 128 clipping occurrences 100 = CLASSD_INT after 256 clipping occurrences 101 = CLASSD_INT after 512 clipping occurrences 110 = CLASSD_INT after 1024 clipping occurrences 111 = CLASSD_INT after 2048 clipping occurrences	0

Table 108: CLASSD_CTRL_REG (0xFF5C00)

Bit	Mode	Symbol	Description	Reset
3-2	R/W	CLASSD_VOUT	Maximum differential output voltage selection between PAOUTp and PAOUTm. 00 = 5Vpp 01 = 6 Vpp 10 = 7.33 Vpp 11 = 10 Vpp	10
1	R	CLASSD_PROT	1 = Temperature and/or current has exceeded the safety limit (automatic protection has been activated) Must be reset by SW by writing to CLASSD_CLEAR_INT_REG.	0
0	R/W	CLASSD_PD	1 = CLASSD amplifier in powerdown, 0 = on (refer to (Note 42))	1

Note 42: When ClassD amplifier state is "on" then the P3_DATA_REG[1:0] should be kept to "00".

Table 109: CLASSD_CLEAR_INT_REG (0xFF5C02)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	CLASSD_CLEAR_INT	Writing any value to this register will clear the CLASSD_CTRL_REG[CLASSD_INT_BIT]. Reading returns 0.	0

Table 110: CLASSD_TEST_REG (0xFF5C06)

Bit	Mode	Symbol	Description	Reset
15-7	-	-	Reserved	0
6-4	R/W	CLASSD_ANA_TEST	0 = normal operation. 1 = map analog testpoints to the general purpose input of ADC	0
3	R/W	CLASSD_FORCE	0 = normal operation 1 = Force PAOUTp and PAOUTm to drive 1.	0
2	R/W	CLASSD_RST_A	0 = normal operation 1 = analog delta sigma modulator in reset	0
1	R/W	CLASSD_RST_D	0 = normal operation 1 = digital delta sigma modulator in reset	0
0	R/W	CLASSD_SWITCH	0 = normal operation 1 = alternative switching strategy (should result in less THD, but more risk of "plopping")	0

Note 43: The bits in CLASSD_TEST_REG must remain '0'

Table 111: CLASSD_NR_REG (0xFF5C08)

Bit	Mode	Symbol	Description	Reset
15-12	-	-	Reserved	0
11-10	-	CLASSD_NR_ZERO	Enable/disable noise reduction is delayed until 00 = no delay 01 = zero-crossing of input signal 10 = input level < -78dB 11 = input level < -60dB	10
9-8	R/W	CLASSD_NR_HYST	Noise reduction is disabled when level is ..dB higher than threshold: 00 = 0dB 01 = 6dB 10 = 12dB 11 = 18dB	00
7-4	R/W	CLASSD_NR_TON	Noise reduction is enabled when level is below threshold for: (based on 10.368/2 MHz) 0000 = direct 0001 = 0.8ms (2^12 periods) 0010 = 1.6ms (2^13 periods) 0011 = 3.2ms (2^14 periods) 0100 = 6.3ms (2^15 periods) 0101 = 12.6ms (2^16 periods) 0110 = 25ms (2^17 periods) 0111 = 50ms (2^18 periods) 1000 = 100ms (2^19 periods) 1001 = 200ms (2^20 periods) 1010 = 400ms (2^21 periods) 1011 = 800ms (2^22 periods)	0101
3-1	R/W	CLASSD_NR_LVL	Threshold for noise reduction, relative to full-scale: 111 = -30dB 110 = -36dB 101 = -42dB 100 = -48dB 011 = -54dB 010 = -60dB 001 = -66dB 000 = -72dB	100
0	R/W	CLASSD_NR_ACTIVE	0 = Noise reduction is disabled 1 = Noise reduction is enabled	1

Table 112: CLK_GLOBAL_REG (0xFF4000)

Bit	Mode	Symbol	Description	Reset
15	-	-	Reserved	0
14-7	R/W	SW_CODEC_CORRECTION_PRG	Controls the codec_div_s output of the CRG used for the codec clocks correction steps in PLL mode. If the system is at XTAL mode then codec_div_s=9 by default.	144
6	R/W	SW_DRAMODE	Controls the EMAC clocks. When '1' the rx/txclk are equal to hclk. This bit has to be programmed from DRA. It is used for the memory test of the EMAC.	0
5	R/W	SW_SWITCH_CLK	Selection between the PLL and XTAL mode 0 = Use the PLL1 clock as the main clock for the clocking of the system 1 = Use the XTAL clock as the main clock for the clocking of the system	1
4-1	-	-	Reserved	0
0	R/W	SW_XTAL_PLL1_RATE	Defines the XTAL-PLL1 clock rate. 1 = pll / xtal = 16 0 = pll /xtal = 8 Example : If PLL is reprogrammed to 80MHz then this value must be set to 0 (80MHz/10MHz=8)	1

Table 113: CLK_AMBA_REG (0xFF4002)

Bit	Mode	Symbol	Description	Reset
15-13	-	-	Reserved	0
12-11	R/W	AHB_CLK_DIV	Controls the ahb_clk, which is used for synchronizing the reset signal and at the Sequencer RAM interface.	1
10	R/W	SW_MCRAM2_EN	0 = Micro Code RAM2 Interface off, clock disabled. Upon RAM access, 0 is returned 1 = Micro Code RAM2 interface on	1
9	R/W	SW_MCRAM1_EN	0 = Micro Code RAM1 Interface off, clock disabled. Upon RAM access, 0 is returned 1 = Micro Code RAM1 interface on	1
8	R/W	SW_SRAM2_EN	0 = Shared RAM2, ROM2 and Gen2DSP2 Interface off, clock disabled. Upon interface access, 0 is returned. 1 = Shared RAM2, ROM2 and Gen2DSP2 register interface on.	1
7	R/W	SW_SRAM1_EN	0 = Shared RAM1, ROM1 and Gen2DSP1 register and common Gen2DSP register interface off, clock disabled. Upon interface access, 0 is returned. 1 = Shared RAM1, ROM1 and Gen2DSP1 register and common Gen2DSP register interface on.	1
6-4	R/W	SW_PCLK_DIV	Division for the pclk, based on the hclk. Maximum allowed clock frequency 41.472 MHz. (Note 44) . Programming of odd values (besides the value "1") will automatically turn to the nearest smaller even value. Divide by 0 means divide by 4.	2
3-0	R/W	SW_HCLK_DIV	Division for the hclk (Note 44). Divide by 0 means divide by 16 in PLL mode and by 8 in XTAL mode. Allowed values are 0,1, 2, 4, 8.	2

Note 44: SW_PCLK_DIV and SW_HCLK_DIV may not be changed at the same time. Make sure that PCLK (set by SW_PCLK_DIV) never exceeds 41.736 MHz as soon as the PLL is enabled at full speed. (See also 18.0 on page 83). During switching from XTAL clock to PLL1 clock always program SW_HCLK_DIV with values greater than or equal to 4.

Table 114: CLK_CODEC1_REG (0xFF4004)

Bit	Mode	Symbol	Description	Reset
15-9	-	-	Reserved	0
8	R/W	SW_ADCDC_EN	Enables the ad_cdc_clk clock 0 = ad_cdc_clk clock disabled. 1 = ad_cdc_clk clock enabled.	0
7-0	R/W	SW_ADCDC_DIV	Division for the ad_cdc_clk clock. CoDec AD from microphone clock frequency selection.	144

Table 115: CLK_CODEC2_REG (0xFF4006)

Bit	Mode	Symbol	Description	Reset
15-9	-	-	Reserved	0
8	R/W	SW_DALSRCDC_EN	Enables the da_lsr_cdc_clk clock 0 = da_lsr_cdc_clk clock disabled. 1 = da_lsr_cdc_clk clock enabled.	0
7-0	R/W	SW_DALSRCDC_DIV	Division for the da_lsr_cdc_clk clock.	144

Table 116: CLK_CODEC3_REG (0xFF4008)

Bit	Mode	Symbol	Description	Reset
15-9	-	-	Reserved	0
8	R/W	SW_DACCLASSCDC_EN	Enables the da_classd_cdc_clk clock 0 = da_classd_cdc_clk clock disabled. 1 = da_classd_cdc_clk clock enabled.	0
7-0	R/W	SW_DACCLASSCDC_DIV	Division for the da_classd_cdc_clk clock. CoDec DA to loudspeaker clock frequency selection.	144

Note 45: A change in frequency selection with CLK_XXX_SEL should only be done if the corresponding bit CLK_XXX_EN is 0.

Table 117: CLK_SPU1_REG (0xFF400A)

Bit	Mode	Symbol	Description	Reset
15	R/W	SW_DSP1_MEM_ACC_FORCE	Forces the dsp1_mem_access to 1	0
14	R/W	SW_DSP2_MEM_ACC_FORCE	Forces the dsp2_mem_access to 1	0
13	R/W	SW_PCMCDC_EN	Enables the pcm_cdc_clk clock 0 = pcm_cdc_clk clock disabled. 1 = pcm_cdc_clk clock enabled.	0
12	R/W	SW_CLK_DSP1_EN	Enables the dsp1_clk clock 0 = dsp1_clk clock disabled 1 = dsp1_clk clock enabled	0
11	R/W	SW_CLK_DSP2_EN	Enables the dsp2_clk clock 0 = dsp2_clk clock disabled 1 = dsp2_clk clock enabled	0
10	R/W	SW_MAINCDC_EN	Enables the main_cdc_clk clock 0 = main_cdc_clk clock disabled 1 = main_cdc_clk clock enabled	0
9-5	R/W	SW_DSP1_DIV	Division for the dsp1_clk. Divide by 0 means divide by 16 in PLL mode and by 8 in XTAL mode.	8
4-0	R/W	SW_DSP2_DIV	Division for the dsp2_clk. Divide by 0 means divide by 16 in PLL mode and by 8 in XTAL mode	8

Table 118: CLK_SPU2_REG (0xFF400C)

Bit	Mode	Symbol	Description	Reset
15-8	R/W	SW_PCMCDC_DIV	Division for the pcm_cdc_clk.	144
7-0	R/W	SW_MAINCDC_DIV	Division for the main_cdc_clk. DSP main counter clock frequency selection. 1.152MHz is the base frequency for generating 8, 16, 32 KHz.	144

Table 119: CLK_CLASSD1_REG (0xFF400E)

Bit	Mode	Symbol	Description	Reset
15-8	-	-	Reserved	0
7	R/W	SW_CLASSD_EN	Enables the classd_clk clock 0 = classd_clk clock disabled 1 = classd_clk clock enabled	0
6-0	R/W	SW_CLASSD_DIV	Division for the classd_clk	32

Table 120: CLK_CLASSD2_REG (0xFF4010)

Bit	Mode	Symbol	Description	Reset
15-12	-	-	Reserved	0
11	R/W	SW_CT_EN	Enables the ct_clk clock 0 = ct_clk clock disabled 1 = ct_clk clock enabled	1
10-0	R/W	SW_CT_DIV	Division for the ct_clk	72

Table 121: CLK_TIM1_REG (0xFF4012)

Bit	Mode	Symbol	Description	Reset
15-12	-	-	Reserved	0
11	R/W	SW_TCK_EN	Enables the tclk_clk clock 0 = tclk_clk clock disabled 1 = tclk_clk clock enabled	1
10	R/W	SW_TC1MC_EN	Enables the tc1_mc_clk clock 0 = tc1_mc_clk clock disabled 1 = tc1_mc_clk clock enabled	1
9-8	R/W	SW_TCKCLK_DIV	Division for the tck_clk	1
7-0	R/W	SW_TC1MCCLK_DIV	Division for the tc1mc_clk. This value will be multiplied with the "SW_timclk_div" to form the final division for the tc1mc_clk.	9

Table 122: CLK_TIM2_REG (0xFF4014)

Bit	Mode	Symbol	Description	Reset
15-7	-	-	Reserved	0
6-2	R/W	SW_TIMCLK_DIV	Division for the tim_clk	16
1-0	R/W	SW_TIMCLKGC_DIV	Division for the tim_clk_gc. This value will be multiplied with the "SW_timclk_div" to form the final division for tim_clk_gc.	1

Table 123: CLK_BMC1_REG (0xFF4016)

Bit	Mode	Symbol	Description	Reset
15-6	-	-	Reserved	0
5	R/W	SW_CLK10SWNC_EN	Enables the "clk10sw_gnc" clock. 0 = clk10sw_gnc clock disabled 1 = clk10sw_gnc clock enabled	1
4-1	-	-	Reserved	0
0	R/W	SW_CLK10SWC_EN	Enable the "clk10sw_c" clock.	1

Table 124: CLK_BMC2_REG (0xFF4018)

Bit	Mode	Symbol	Description	Reset
15-9	-	-	Reserved	
8	R/W	SW_CLK1GC_EN	Enables the clk1_gc clock 0 = clk1_gc clock disabled 1 = clk1_gc clock enabled	1
7-0	-		Reserved	9

Table 125: CLK_BMC3_REG (0xFF401A)

Bit	Mode	Symbol	Description	Reset
15-11	-	-	Reserved	
10	R/W	SW_CLK1SWGC_EN	Enables the clk1sw_gc clock 0 = clk1sw_gc clock disabled 1 = clk1sw_gc clock enabled	1
9-0	-	-	Reserved	0

Table 126: CLK_GPIO1_REG (0xFF401C)

Bit	Mode	Symbol	Description	Reset
15-10	R/W	SW_SPI_DIV	Division for the spi_clk	8
9-0	R/W	SW_ADC_DIV	Division for the adc_clk	36

Table 127: CLK_GPIO2_REG (0xFF401E)

Bit	Mode	Symbol	Description	Reset
15-10	R/W	SW_SPI2_DIV	Division for the spi2_clk	8
9-0	R/W	SW_ADC10SA_DIV	Division for the adc10sa_clk	36

Table 128: CLK_GPIO3_REG (0xFF4020)

Bit	Mode	Symbol	Description	Reset
15	-	-	Reserved	0
14	R/W	SW_SPI2_EN	Enables the spi2_clk clock	0
13	R/W	SW_ADC_EN	Enables the adc_clk clock	0
12	R/W	SW_SPI_EN	Enables the spi_clk clock	0
11	R/W	SW_TMR0_EN	Enables the tmr0_clk clock 0 = tmr0_clk clock disabled 1 = tmr0_clk clock enabled	0
10-0	R/W	SW_TMR0_DIV	Division for the tmr0_clk	144

Table 129: CLK_GPIO4_REG (0xFF4022)

Bit	Mode	Symbol	Description	Reset
15-12	-	-	Reserved	0
11	R/W	SW_TMR1_EN	Enables the tmr1_clk clock 0 = tmr1_clk clock disabled 1 = tmr1_clk clock enabled	0
10-0	R/W	SW_TMR1_DIV	Division for the tmr1_clk	144

Table 130: CLK_GPIO5_REG (0xFF4024)

Bit	Mode	Symbol	Description	Reset
15-12	-	-	Reserved	0
11	R/W	SW_AB1_EN	Enables the ab1_clk clock 0 = ab1_clk clock disabled 1 = ab1_clk clock enabled	0
10-6	R/W	SW_AB1_DIV	Division for the ab1_clk	1
5	R/W	SW_AB2_EN	Enables the ab2_clk clock 0 = ab2_clk clock disabled 1 = ab2_clk clock enabled	0
4-0	R/W	SW_AB2_DIV	Division for the ab2_clk	1

Table 131: CLK_GPIO6_REG (0xFF4026)

Bit	Mode	Symbol	Description	Reset
15-7	-	-	Reserved	0
6	R/W	SW_UART_EN	Enables the uart_clk clock 0 = uart_clk clock disabled 1 = uart_clk clock enabled	0
5-0	R/W	SW_UART_DIV	Division for the uart_clk	1

Table 132: CLK_GPIO7_REG (0xFF4028)

Bit	Mode	Symbol	Description	Reset
15-6	-	-	Reserved	0
5	R/W	SW_IRDACLK_EN	Enables the irda_clk clock 0 = irda_clk clock disabled 1 = irda_clk clock enabled	0
4-0	R/W	SW_IRDACLK_DIV	Division for the irda_clk	1

Table 133: CLK_GPIO8_REG (0xFF402A)

Bit	Mode	Symbol	Description	Reset
15-13	-	-	Reserved	0
12-8	R/W	SW_PER10_DIV	Division for the per10_clk	1
7-0	R/W	SW_KBRD_DIV	Division for the kbrd_clk	9

Table 134: CLK_AUX1_REG (0xFF402C)

Bit	Mode	Symbol	Description	Reset
15-7	-	-	Reserved	0
6	R	SWITCH_TO_PLL	This is a READ ONLY bit. If this signal is '0' then all clock gaters are switched to PLL mode.	1
5	R	SWITCH_TO_XTAL	This is a READ ONLY bit. If this signal is '1' then all clock gaters have switched to XTAL mode	1
4	R/W	SW_DCS_SEL	If '0' the dsc_clk1_data signal follows the clk1sw_gc clock. If '1' the dsc_clk1_data signal follows the tc1mc_clk clock.	1
3	R/W	SW_DCLK_EN	Enables the dclk_g clock 0 = dclk_g clock disabled 1 = dclk_g clock enabled	1
2-1	R/W	SW_DCLK_DIV	Division for the dclk_g. This value will be multiplied with the "SW_timclk_div" to form the final division for "tim_clk_gc".	1
0	R/W	SW_MWRGC_EN	Enables the mwr_gc clock 0 = mwr_gc clock disabled 1 = mwr_gc clock enabled	1

Table 135: CLK_AUX2_REG (0xFF402E)

Bit	Mode	Symbol	Description	Reset
15-14	-	-	Reserved	0
13	R/W	SW_QSPI_CK_CLK_DIV	Selects the division of the source clock for the generated qspi bus clock. 0 = the qspi bus clock is equal to the source clock 1 = the qspi bus clock is equal to the (source clock) /2 (see also SW_QSPI_SEL_SOURCE)	0
12	R/W	SW_EMAC_EN	Enables the clock of the EMAC ahb part (emac_hclk) 0 = emac_hclk clock disabled 1 = emac_hclk clock enabled	0
11	R/W	SW_CRYPTO_EN	Enables the clock of the CRYPTO (crypto_hclk) 0 = crypto_hclk clock disabled 1 = crypto_hclk clock enabled	0
10	R/W	SW_QSPI_EN	Enables the clocks (ahb part and qspi bus part) of the QSPI 0 = QSPI clocks disabled 1 = QSPI clocks enabled	0
9	R/W	SW_QSPI_SEL_SO URCE	Selects the qspi bus source clock 0 = use the PLL2 clock to generate the qspi bus clock 1 = use the HCLK clock to generate the qspi bus clock	0
8	R/W	SW_RMII_EN	Selects the RMII/MII mode 0 = for the MII mode 1 = for the RMII mode	1
7	-	-	Reserved	0
6	R/W	SW_ETH_SEL_SPEE D	Selects the speed of the interface 0 = for 10 Mbps 1 = for 100 Mbps	0
5	R/W	SW_BXTAL_EN	Enables the bxtal_clk clock 0 = bxtal_clk clock disabled 1 = bxtal_clk clock enabled	0
4-0	R/W	SW_BXTAL_DIV	Division for the bxtal_clk	8

Table 136: CLK_JOWI_REG (0xFF4030)

Bit	Mode	Symbol	Description	Reset
15-6	-	-	Reserved	0
8	R/W	SW_OWICLK_C_EN	Enables the owiclk clock 0 = owiclk clock disabled 1 = owiclk clock enabled	1
7-0	R/W	SW_OWICLK_C_DIV	Division for the owiclk. Divide by 0 means divide by 16 in PLL mode and by 8 in XTAL mode	1

Table 137: CLK_CLK100A_REG (0xFF4032)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	SW_CLK100_DI_VL	Division for the clk100 clock. The sixteen least significant bits.	44988

Table 138: CLK_CLK100B_REG (0xFF4034)

Bit	Mode	Symbol	Description	Reset
15-5	-	-	Reserved	0
4-0	R/W	SW_CLK100_DI_VH	Division for the clk100 clock. The five most significant bits.	1 (total reset value = 110523)

Table 139: CLK_DCDC1_REG (0xFF4036)

Bit	Mode	Symbol	Description	Reset
15-1	-	-	Reserved	0
0	R/W	SW_DC_EN	Enables the dc_clk clock	0

Table 140: CLK_PLL1_CTRL_REG (0xFF403A)

Bit	Mode	Symbol	Description	Reset																																								
15-7	-	-	Reserved	0																																								
6	R/W	PLL1_HF_SEL	PLL High frequency mode selection. Must always be set to 1 at the same time as VCO_ON is set. Examples: <table> <thead> <tr> <th>Fxtal</th> <th>Fvco</th> <th>VD/XD</th> <th>Fupd</th> <th>HF_SEL</th> </tr> </thead> <tbody> <tr> <td>10.368</td> <td>82.944</td> <td>2x4/1 or 4x4/2</td> <td>10.368 MHz</td> <td>1</td> </tr> <tr> <td>10.368</td> <td>124.416</td> <td>3x4/1</td> <td>10.368 MHz</td> <td>1</td> </tr> <tr> <td>10.368</td> <td>165.888</td> <td>4x4/1</td> <td>10.368 MHz</td> <td>1</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>12.288</td> <td>55.296</td> <td>1x9/2</td> <td>6.144 MHz</td> <td>1</td> </tr> <tr> <td>12.288</td> <td>110.582</td> <td>2x9/2</td> <td>6.144 MHz</td> <td>1</td> </tr> <tr> <td>12.288</td> <td>165.888</td> <td>3x9/2</td> <td>6.144 MHz</td> <td>1</td> </tr> </tbody> </table>	Fxtal	Fvco	VD/XD	Fupd	HF_SEL	10.368	82.944	2x4/1 or 4x4/2	10.368 MHz	1	10.368	124.416	3x4/1	10.368 MHz	1	10.368	165.888	4x4/1	10.368 MHz	1						12.288	55.296	1x9/2	6.144 MHz	1	12.288	110.582	2x9/2	6.144 MHz	1	12.288	165.888	3x9/2	6.144 MHz	1	0
Fxtal	Fvco	VD/XD	Fupd	HF_SEL																																								
10.368	82.944	2x4/1 or 4x4/2	10.368 MHz	1																																								
10.368	124.416	3x4/1	10.368 MHz	1																																								
10.368	165.888	4x4/1	10.368 MHz	1																																								
12.288	55.296	1x9/2	6.144 MHz	1																																								
12.288	110.582	2x9/2	6.144 MHz	1																																								
12.288	165.888	3x9/2	6.144 MHz	1																																								
5	-	-	Reserved	0																																								
4	R/W	PLL1_VCO_ON	0 = PLL1 VCO off, can only be set to 0 if PLL_CLK_SEL=0 1 = PLL1 VCO on, settling time to be timed with software (See "PLL1/PLL2 switching On/Off" on page 31)	0																																								
3	-	-	Reserved	0																																								
2	R/W	PLL1_OUT_DIV	0 = Divide PLL1 output by 1 1 = Divide PLL1 output by 2 (internal VCO frequency is divided by two)	0																																								
1	R/W	PLL1_CP_ON	0 = PLL VCO Charge pump off, can only be set to 0 if PLL_CLK_SEL=0 1 = PLL VCO Charge pump on (Normal operation)	0																																								
0	R/W	PLL1_TESTMODE_SEL	If PLL test mode enabled: 0 = VCO test mode 1 = PLL divider test mode.	0																																								

Table 141: CLK_PLL1_DIV_REG(0xFF403C)

Bit	Mode	Symbol	Description	Reset
15-5	-	-	Reserved	0
4-2	R/W	PLL1_VD	PLL1 VCO divider 000 = divide by 1x4 001 = divide by 2x4 010 = divide by 3x4 111 = divide by 4x4 100 = divide by 1x9 101 = divide by 2x9 110 = divide by 3x9	0
1-0	R/W	PLL1_XD	PLL1 xtal divider 00 = Divide by 1 01 = Divide by 2 1x = Reserved	0

Table 142: CLK_PLL2_CTRL_REG (0xFF403E)

Bit	Mode	Symbol	Description	Reset																														
15-7	-	-	Reserved	0																														
6	R/W	PLL2_HF_SEL		0																														
5	R/W	PLL2_LPF_SEL	0 = PLL slow loop filter; used if Fupd <=1 MHz 1 = PLL fast loop filter; used if Fupd > 1MHz Examples: <table border="1"><thead><tr><th>Fxtal</th><th>Fvco</th><th>XD/VD</th><th>Fupd</th><th>LPF SEL</th><th>PLL2_OUT</th></tr></thead><tbody><tr><td>10.368</td><td>48 MHz</td><td>27/125</td><td>384000 Hz</td><td>0</td><td>0</td></tr><tr><td>10.368</td><td>50 MHz</td><td>62/299</td><td>162226 Hz</td><td>0</td><td>1 (25MHz)</td></tr><tr><td>12.288</td><td>48 MHz</td><td>32/125</td><td>384000 Hz</td><td>0</td><td>0</td></tr><tr><td>12.288</td><td>25 MHz</td><td>29/59</td><td>423724 Hz</td><td>0</td><td>0</td></tr></tbody></table>	Fxtal	Fvco	XD/VD	Fupd	LPF SEL	PLL2_OUT	10.368	48 MHz	27/125	384000 Hz	0	0	10.368	50 MHz	62/299	162226 Hz	0	1 (25MHz)	12.288	48 MHz	32/125	384000 Hz	0	0	12.288	25 MHz	29/59	423724 Hz	0	0	0
Fxtal	Fvco	XD/VD	Fupd	LPF SEL	PLL2_OUT																													
10.368	48 MHz	27/125	384000 Hz	0	0																													
10.368	50 MHz	62/299	162226 Hz	0	1 (25MHz)																													
12.288	48 MHz	32/125	384000 Hz	0	0																													
12.288	25 MHz	29/59	423724 Hz	0	0																													
4	R/W	PLL2_VCO_ON	0 = PLL2 VCO off, can only be set to 0 if PLL_CLK_SEL=0 1 = PLL2 VCO on	0																														
3	-	-	Reserved	0																														
2	R/W	PLL2_PLL_OUT_DIV	0 = Divide PLL2 output by 1 1 = Divide PLL2 output by 2	0																														
1	R/W	PLL2_CP_ON	0 = PLL VCO Charge pump off, can only be set to 0 if PLL_CLK_SEL=0 1 = PLL VCO Charge pump on (Normal operation)	0																														
0	R/W	PLL2_TESTMODE_SEL	If PLL test mode enabled: 0 = VCO test mode 1 = PLL divider test mode.	0																														

Table 143: CLK_PLL2_DIV_REG (0xFF4040)

Bit	Mode	Symbol	Description	Reset
15	R/W	PLL2_DIV5	0 = VCO Divider according DIV3, DIV4, VD 1 = VCO Divider 13x23 = 299. (for 50 MHz out of 10.368 MHz)	0
14	R/W	PLL2_DIV3	PLL VCO divider	0
13	R/W	PLL2_DIV4	DIV3 DIV 4 VD[3-0] Divide by 0 0 0000 2x59 (For 50 MHz out of 12.288 MHz) 1 0 xxxx 1x8 0 0 0001 2x8=16 0 0 0010 3x8=24 0 0 1111 16x8=128	0
12-9	R/W	PLL2_VD	0 1 0000 5x25= 125 (for 48 MHz) 1 1 xxxx 1x12 0 1 0001 2x12=24 0 1 0010 3x12=36 0 1 1011 12x12=144 0 1 1111 16x12=192	0
8	R/W	PLL2_DIV1	PLL xtal divider	0
7	R/W	PLL2_DIV2	DIV1 DIV 2 XD[6-0] Divide by 0 0 000 0000 undefined	0
6-0	R/W	PLL2_XD	0 0 xxxx 4 1 1 xxxx xxxx 3 0 x 000 0111 7 0 x 000 1010 10 0 x 000 1011 11 0 x 000 1101 13 0 x 000 1110 14 0 x 100 0001 65 0 x 111 1111 127	0000000

Table 144: CLK_FREQ_TRIM_REG (0xFF4042)

Bit	Mode	Symbol	Description	Reset
15-10	-	-	Reserved	0
9	R	OSC_OK	0 = Oscillator amplitude too low due to too high values of quartz resistance and/or capacitors on XTAL and CAP. 1 = Oscillator amplitude ok	1
8	R/W	CL_SEL	0 = Xtal load capacitance CL >=20 pF 1 = Xtal load capacitance CL < 20 pF	0
7-5	R/W	COARSE_ADJ	Xtal frequency course trimming register. Increment or decrement the binary value with 1.	0x7
4-0	R/W	FINE_ADJ	Xtal frequency fine trimming register. Increment or decrement the binary value with 1. Change 1 bit at a time to prevent phase jumps	0x1F

Table 145: CLK_XTAL_CTRL_REG (0xFF4044)

Bit	Mode	Symbol	Description	Reset
15-8	-	-	Reserved	0
7	R/W	XTAL_EXTRA_CV	Additional XTAL oscillator Cv1 and Cv2 capacitors of 25 pF each for improved start-up. After start-up, this must be set to 0 for reduced power consumption in the xtal oscillator. This bit can only be cleared but not set by SW. 0 = Capacitors disabled. 1 = Capacitors enabled.	1
6	RO	LDO_RFCLK_OK	LDO_RFCLK regulator status 0 = LDO_RFCLK voltage not yet reached 1 = LDO_RFCLK voltage ok	-
5	R/W	RFCLK_SUPPLY	0 = RFCLK pad supplied by VDDIO_RFDIG. 1 = RFCLK pad supplied by LDO_RFCLK (Recommended) To use RFCLK supplied from LDO_RF, set LDO_RFCLK_ON to desired level, wait for LDO_RFCLK_OK=1 (or 200us), then set RFCLK_SUPPLY=1.	0
4-3	R/W	LDO_RFCLK_ON	00 = LDO_RFCLK is off. (If bit RFCLK_SUPPLY= 0, LDO_RFCLK can not be disabled) 01 = VDDIO_RFCLK = 1.6V, VDDIO_RFDIG >1.8V 10 = VDDIO_RFCLK = 2.3V, VDDIO_RFDIG >2.5V 11 = VDDIO_RFCLK = 3.1V, VDDIO_RFDIG >3.3V (not tested) VDDIO_RFCLK must be decoupled with 1uF ceramic capacitor if used.	0
2	RO	AVD_XTAL_OK	LDO_XTAL regulator (1.6V) status 0 = AVD_XTAL voltage not yet reached 1 = AVD_XTAL voltage ok	-
1	R/W	XTAL_SUPPLY	0 = XTAL supplied by LDO_XTAL. (Recommended) After LDO_XTAL_ON = 1, wait for AVD_XTAL_OK (or 200 us) before XTAL_SUPPLY is set to 0. (Refer to "switching procedureS" on page 31) 1 = XTAL supplied by VDDRF.	1
0	R/W	LDO_XTAL_ON	0 = LDO_XTAL is off. If bit XTAL_SUPPLY = 0, LDO_XTAL can not be disabled. 1 = LDO_XTAL is enabled. AVD_XTAL is 1.6V Supply voltage for Xtal oscillator, 6 bits TDO volume DAC and 10 bits ADC DAC. AVD_XTAL must be decoupled with 1uF ceramic capacitor.	0

Table 146: CLK_XDIV0_REG (0xFF4046)

Bit	Mode	Symbol	Description	Reset
15-11	-	-	Reserved	0
10	R/W	BXTAL_XDIV_EN	Divides all bxtal clocks by the XDIV_VAL	0
9	R/W	DCDC_XDIV_EN	Divides all dc当地 clocks by the XDIV_VAL	0
8	R/W	JOWI_XDIV_EN	Divides all jowi clocks by the XDIV_VAL	0
7	R/W	RISCUTIL_XDIV_EN	Divides all riscutil clocks by the XDIV_VAL	0
6	R/W	DSC_XDIV_EN	Divides all dsc clocks by the XDIV_VAL	0
5	-	-	Reserved	0
4	R/W	BMC_XDIV_EN	Divides all bmc clocks by the XDIV_VAL	0
3	R/W	TIM_XDIV_EN	Divides all tim clocks by the XDIV_VAL	0
2	R/W	SPU_XDIV_EN	Divides all spu clocks by the XDIV_VAL	0
1	R/W	CODEC_XDIV_EN	Divides all codec clocks by the XDIV_VAL	0
0	R/W	CLASSD_XDIV_EN	Divides all classd clocks by the XDIV_VAL	0

Table 147: CLK_XDIV1_REG (0xFF4048)

Bit	Mode	Symbol	Description	Reset
15-14	-	-	Reserved	0
13	R/W	AHB_XDIV_EN	Divides all hclk/pclk clocks by the XDIV_VAL	0
12	R/W	KBRD_XDIV_EN	Divides all kbrd clocks by the XDIV_VAL	0
11	R/W	PER10_XDIV_EN	Divides all per10 clocks by the XDIV_VAL	0
10	R/W	CT_XDIV_EN	Divides all ct clocks by the XDIV_VAL	0
9	R/W	SPI2_XDIV_EN	Divides all spi2 clocks by the XDIV_VAL	0
8	R/W	SPI_XDIV_EN	Divides all spi clocks by the XDIV_VAL	0
7	R/W	IRDA_XDIV_EN	Divides all irda clocks by the XDIV_VAL	0
6	R/W	UART_XDIV_EN	Divides all uart clocks by the XDIV_VAL	0
5	R/W	AB2_XDIV_EN	Divides all ab2 clocks by the XDIV_VAL	0
4	R/W	AB1_XDIV_EN	Divides all ab1 clocks by the XDIV_VAL	0
3	R/W	TMR1_XDIV_EN	Divides all tmr1 clocks by the XDIV_VAL	0
2	R/W	TMR0_XDIV_EN	Divides all tmr0 clocks by the XDIV_VAL	0
1	R/W	ADC10SA_XDIV_EN	Divides all adc10sa clocks by the XDIV_VAL	0
0	R/W	ADC_XDIV_EN	Divides all adc clocks by the XDIV_VAL	0

Table 148: CLK_XDIV_VAL_REG (0xFF404A)

Bit	Mode	Symbol	Description	Reset
15-5	-	-	Reserved	0
4-0	R/W	XDIV_VAL	Defines the value tha the XDIV flag wil further divide the output clocks with (shift actual divider by 4 meaning divide the clock by 16)	4

Table 149: CLK_CDC_CORRECT_REG (0xFF404C)

Bit	Mode	Symbol	Description	Reset
15-10	-	-	Reserved	0
9-8	R/W	SW_MAINCDC_COR	Enables the ccp/ccm commands at programmable period. 0 -> ccp/ccm applies at every cycle 1 -> ccp/ccm applies every second cycle 2 -> ccp/ccm applies every fourth cycle 3 -> ccp/ccm applies every eighth cycle	0
7-6	R/W	SW_PCMCDC_COR	Enables the ccp/ccm commands at programmable period. 0 -> ccp/ccm applies at every cycle 1 -> ccp/ccm applies every second cycle 2 -> ccp/ccm applies every fourth cycle 3 -> ccp/ccm applies every eighth cycle	0
5-4	R/W	SW_DACLASSD_CO_R	Enables the ccp/ccm commands at programmable period. 0 -> ccp/ccm applies at every cycle 1 -> ccp/ccm applies every second cycle 2 -> ccp/ccm applies every fourth cycle 3 -> ccp/ccm applies every eighth cycle	0
3-2	R/W	SW_DALSR_COR	Enables the ccp/ccm commands at programmable period. 0 -> ccp/ccm applies at every cycle 1 -> ccp/ccm applies every second cycle 2 -> ccp/ccm applies every fourth cycle 3 -> ccp/ccm applies every eighth cycle	0
1-0	R/W	SW_ADCDC_COR	Enables the ccp/ccm commands at programmable period. 0 -> ccp/ccm applies at every cycle 1 -> ccp/ccm applies every second cycle 2 -> ccp/ccm applies every fourth cycle 3 -> ccp/ccm applies every eighth cycle	0

Continued

Table 150: CODEC_ADDA_REG (0xFF5808)

Bit	Mode	Symbol	Description	Reset
15	R/W	AUTO_SYNC	0 = Auto sync to Gen2DSPx enabled. Codec resynchronizes automatically after the main counter resynchronisation. (e.g after DIP <A_NORM>) (recommended) 1 = Codec auto sync to Gen2DSPx disabled. Codec only automatically synchronizes at initial main counter start-up.	0
14-13	R/W	ADC_VREF_LSR	Headset detection selection for ADC. 00 = VREFp input selected for ADC input "0011" 01 = LSRp input selected for ADC input "0011" 10 = LSRn input selected for ADC input "0011" 11 = Reserved. (see Figure 34)	0
12-10	R/W	LPF_BW	Optional LPF filter after DAC Filter characteristics at CLK_DA_LSR_SEL = 1.152 MHz. (8kHz) All frequencies scale with CLK_DA_LSR_SEL value. 000 = 5.5 kHz -1.85 dB@4kHz 001 = 10.9 kHz -0.56 dB@4kHz 010 = 16 kHz -0.28 dB@4kHz 011 = 21 kHz -0.16 dB@4kHz 100 = 16 kHz -0.28 dB@4kHz 101 = 21 kHz -0.16 dB@4kHz 110 = 26 kHz -0.10 dB@4kHz 111 = 30.7 kHz -0.07dB@4kHz This optional noise improvement can be set if Gen2DSP processes at 8kHz and interpolates to 16 kHz while CLK_DA_LSR_SEL is set to 2.304 MHz/16 kHz.	0
9	R/W	LPF_PD	Optional LPF filter after DAC 0 = Enabled. CODEC_LSR_REG[LSRP_MODE] and [LSRN_MODE] must also be set to 01 or 11 1 = Disabled.	1
8	R/W	DA_HBW	DAC Half bandwidth selection. Filter characteristics at CLK_DA_LSR_SEL = 1.152 Mhz. (8kHz) All frequencies scale with CLK_DA_LSR value 0 = Full bandwidth: -3dB@8kHz, -0.45dB@4kHz 1 = Half Bandwidth: -3dB@4kHz, -0.45dB@2kHz This optional noise improvement can be set if Gen2DSP processes at 8kHz and interpolates to 16 kHz while CLK_DA_LSR_SEL is set to 2.304 MHz/16 kHz.	0
7	R/W	AD_DITH_OFF	0 = Codec AD dithering on (normal), 1 = off	0
6	R/W	DA_DITH_OFF	0 = Code DA dithering on (normal), 1 = off	0
5-4	R/W	AD_CADJ	AD opamps bias current adjustment. Shall not be modified.	10
3-2	R/W	DA_CADJ	DA opamps bias current adjustment. Shall not be modified.	10
1	R/W	AD_PD	1 = Codec AD powerdown, 0 = on	1
0	R/W	DA_PD	1 = Codec DA powerdown, 0 = on	1

Table 151: CODEC_LSR_REG (0xFF5802)

Bit	Mode	Symbol	Description	Reset
15-10	-	-	Reserved	0
9	R/W	LSREN_SE	1 = Differential to single ended conversion enabled. 0 = disabled	0
8-6	R/W	LSRATT	Loudspeaker amplifier analog gain in steps of 2 dB 000 = 2dB, 111 = -12 dB (See Table 375 for voltage level at 0 dBm0)	0
5	R/W	LSRN_PD	1 = LSR- amplifier powerdown, 0 = on.	1
4-3	R/W	LSRN_MODE	00 = LSR- output has AGND reference voltage (If LSRN_PD = 1 then LSR- has high impedance) 01 = DA- via extra LP-filter to LSR- amplifier 10 = Differential mode DA- to LSR- amplifier 11 = Single ended mode to LSR- amplifier (LSREN_SE must be set to 1)	00
2	R/W	LSRP_PD	1 = LSR+ amplifier powerdown, 0 = on.	1
1-0	R/W	LSRP_MODE	00 = LSR+ output has AGND reference voltage (If LSRP_PD = 1 then LSR+ has high impedance) 01 = DA+ via extra LP-filter to LSR+ amplifier 10 = Differential mode DA+ to LSR+ amplifier 11 = Single ended mode to LSR+ amplifier (LSREN_SE must be set to 1)	00

Table 152: CODEC_MIC_REG (0xFF5800)

Bit	Mode	Symbol	Description	Reset
15-14	-	-	Reserved	0
13	R/W	MIC_CADJ	Microphone amplifier current (bandwidth) adjustment 0 = 1 x current (default fs = 8 kHz mode) 1 = 1/2 x current	0
12	R/W	MICHP_ON	0 = MICHP input disabled. 1 = MICHP input enabled.	1 (Note 46)
11	R/W	MICHN_ON	0 = MICHn input disabled. 1 = MICHh input enabled.	0
10	R/W	DSP_CTRL	0 = Microphone gain controlled with MIC_GAIN 1 = Microphone gain controlled with GenDSP	0
9	R/W	MIC_OFFCOM_ON	Offset compensation control 0 = off (normal operation) 1 = on.	0
8	R/W	MIC_OFFCOM_SG	Offset compensation values 0 = added to MICp. 1 = added to MICn.	0
7-4	R/W	MIC_GAIN	Microphone amplifier gain in steps of 2 dB 0000 = 0 dB, 1111 = +30 dB (See Table 371 for voltage level at 0 dBm0)	0000
3	R/W	MIC_MUTE	0 = normal operation, 1 = mute MIC inputs for offset voltage measurement.	0
2	R/W	MIC_PD	1 = Microphone amplifier powerdown, 0 = on.	1
1-0	R/W	MIC_MODE	Microphone input configuration M[1], M[0] 00 = Normal Differential or single ended AC coupled. 01 = MICp single ended. MICn disabled; High impedance 10 = MICn single ended. MICp disabled; High impedance 11 = MICp/n disabled; High impedance. And microphone amplifier is muted. Disabled pins are internally connected to analog ground.	00

Note 46: Notice that MICHP_ON is default '1' and must be set to '0' when using MICp.

Table 153: CODEC_OFFSET1_REG (0xFF580A)

Bit	Mode	Symbol	Description	Reset
15-0	R	COR1	Offset error value (2's complement) measured with MIC_MUTE=1. Each MIC_GAIN setting has a different offset error value.	0000

Table 154: CODEC_OFFSET2_REG (0xFF580E)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	COR2	Offset compensation value 2 (2's complement). For an applied MIC_GAIN setting, the offset error value must be copied to this register.	0000

Table 155: CODEC_TEST_CTRL_REG (0xFF580C)

Bit	Mode	Symbol	Description	Reset
15-14	-	-	Reserved	0
13	R	COR_STAT	0 = CODEC_OFFSET2_REG may be loaded with a new value. 1 = The CODEC_OFFSET2_REG is being copied to the internal offset compensation register, caused by a write operation to the CODEC_MIC_REG. During this synchronisation process, the new MIC_GAIN and corresponding CODEC_OFFSET2_REG value become effective at the same time.	0
12	R/W	COR_ON	0 = Digital offset compensation disabled. 1 = Digital offset compensation enabled. Must be set to 1 before the CODEC_OFFSET2_REG and CODEC_MIC_REG[MIC_GAIN] are initially written.	0
11-0	R/W	TCR	Test control bit must be 0 for normal operation	0

Table 156: CODEC_TONE_REG (0xFF5806)

Bit	Mode	Symbol	Description	Reset
15-3	-	-	Reserved	
2	R/W	RNG_CMP_PD	0 = Ringing comparator on 1 = Ringing comparator powerdown,	1
1	R/W	CID_PR_DIS	0 = Enable protection diodes for MICp VREFp pin. 1 = Disable protection diodes for MICp VREFp pin.	0
0	R/W	CID_PD	0 = Enable external CIDOUT input to ADC 1 = Disable external CIDOUT input	1

Note 47: If CODEC_TONE_REG[CID_PR_DIS] = 1, the protection is disabled from MICp and VREFp (left over from old design). With protection enabled, a current will flow from VREFp. The device will not be damaged but the extra current will increase the power dissipation.

Table 157: CODEC_VREF_REG (0xFF5804)

Bit	Mode	Symbol	Description	Reset
15-10	-	-	Reserved	0
9	R/W	REFINT_PD	Internal reference opamps powerdown. Must be active if Codec AD or DA is used. 0 = on. 1 = powerdown	1
8	R/W	AGND_LSR_PD	1 = internal LSR analog ground buffer powerdown, 0 = on. Must be on if LSR is used.	1
7	R/W	BIAS_PD	1 = Bias current for all CODEC amplifiers powerdown. 0 = on.	1
6	R/W	VREF_BG_PD	1 = Bandgap current mirror powerdown. 0 = on. Must be active if VREFp is used.	1
5	R/W	AMP1V5_PD	1 = internal 0.9 V to 1.5 V convertor powerdown. 0 = on. Must be on if VREFp is used.	1
4	R/W	VREF_INIT	1 = bypass high-ohmic filter resistance with 200k ohm switch to precharge filter capacitor, 0 = high-ohmic resistance on (must be set after 100 us)	1
2-3	R/W	VREF_FILT_CADJ	Adjust noise filter bias current to adjust noise bandwidth. For Test purposes only. Recommended setting = 01.	01
1	R/W	VREF_FILT_PD	1 = bandgap noise filter powerdown. 0 = on	1
0	R/W	VREF_PD	1 = VREFp amplifier powerdown. 0 = on. Must be on to generate internal analog grounds.	1

Note 48: The CODEC AFE biasing (CODEC_VREF_REG) must be active before powering up the rest of the CODEC AFE blocks:

The following switch on order is recommended:

CODEC_VREF_REG -> bias on, all blocks on, VREF_INIT=1 to precharge filter capacitor,

wait at least 100 us

CODEC_VREF_REG[VREF_INIT] =0

CODEC_MIC_REG -> microphone amplifier on

CODEC_ADDA_REG -> AD and DA converters on

CODEC_LSR_REG -> loudspeaker amplifiers on.

Table 158: CRYPTO_CLRIRQ_REG (0xFF0818)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
31-1	-	-	Reserved	0
0	W	CRYPTO_CLRIRQ	Write 1 to clear a pending interrupt request.	0

Table 159: CRYPTO_CTRL_REG (0xFF0800)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
31-9	-	-	Reserved	0
8	R/W	CRYPTO_IRQ_EN	Interrupt Request Enable 0 – The interrupt generation ability is disabled. 1 – The interrupt generation ability is enabled. Generate an interrupt request at the end of operation.	0
7	R/W	CRYPTO_ENCDEC	Encryption/Decryption 0 - Encryption 1 - Decryption	0
6-5	R/W	CRYPTO_AES_KEY_SZ	The size of AES Key 00 - 128 bits AES Key 01 - 192 bits AES Key 10 - 256 bits AES Key 11 - 256 bits AES Key	0
4	R/W	CRYPTO_OUT_MD	Output Mode 0 - Write back to memory all the resulting data 1 - Write back to memory only the final block of the resulting data	0
3-2	R/W	CRYPTO_ALG_MD	Mode of operation 00 - ECB 01 - ECB 10 - CTR 11 - CBC	0
1-0	R/W	CRYPTO_ALG	Algorithm selection 00 - AES 01 - AES 10 - Triple DES 11 – DES	0

Table 160: CRYPTO_DEST_ADDR_REG (0xFF0810)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
31-0	R/W	CRYPTO_DEST_ADDR	Destination Address.The Crypto IP is writing the output data to this memory position.	0

Table 161: CRYPTO_FETCH_ADDR_REG (0xFF0808)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
31-0	R/W	CRYPTO_FETCH_ADDR	Fetch Address.The Crypto IP is reading the input data starting from this memory position.	0

Table 162: CRYPTO_KEYS_START (0xFF0900)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
31-0	R/W		CRYPTO_KEY{0-63} This is the AES keys storage memory (). Fill this memory with the results of the KeyExpansion function (Federal Information Processing Standards Publication 197, Advanced Encryption Standard). This memory is accessible via AHB slave interface, only when the Crypto IP is inactive (CRYPTO_INACTIVE = 1).	X

Note 49: The keys are stored in the memory range starting from the CRYPTO_KEYS_START up to CRYPTO_KEYS_END.

Table 163: CRYPTO_LEN_REG (0xFF080C)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
31-0	R/W	CRYPTO_LEN	It contains the number of bytes of input data. If this number is not a multiple of a block size, the data is automatically extended with zeros.	0

Table 164: CRYPTO_MREG0_REG, CRYPTO_MREG1_REG, CRYPTO_MREG2_REG, CRYPTO_MREG3_REG, CRYPTO_MREG4_REG, CRYPTO_MREG5_REG, CRYPTO_MREG6_REG, CRYPTO_MREG7_REG (0xFF081C, 0xFF0820, 0xFF0824, 0xFF0828, 0xFF082C, 0xFF0830, 0xFF0834, 0xFF0838)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
31-0	R/W	CRYPTO_MREG{0-7}	It contains mode of operation depended information. For more information look in "CRYPTO MREG's"	0

Table 165: CRYPTO MREG's

Register	CBC 64 bits	CBC 128 bits	CTR 64 bits	CTR 128 bits	DES	Triple DES
CRYPTO_MREG0_REG	IV[31:0]	IV[31:0]	CTRBLK[31:0]	CTRBLK[31:0]	-	
CRYPTO_MREG1_REG	IV[63:32]	IV[63:32]	CTRBLK[63:32]	CTRBLK[63:32]	-	
CRYPTO_MREG2_REG	-	IV[95:64]	-	CTRBLK[95:64]	DESKEY [31:0]	DESKEY1 [31:0]
CRYPTO_MREG3_REG	-	IV[127:96]	-	CTRBLK[127:96]	DESKEY [63:32]	DESKEY1 [63:32]
CRYPTO_MREG4_REG	-	-	-	-	-	DESKEY2 [31:0]
CRYPTO_MREG5_REG	-	-	-	-	-	DESKEY2 [63:32]
CRYPTO_MREG6_REG	-	-	-	-	-	DESKEY3 [31:0]
CRYPTO_MREG7_REG	-	-	-	-	-	DESKEY3 [63:32]

Note 50: IV: initialization vector

Note 51: CTRBLK: Counter Block. The Field CTRBLK [31:0] is the initial value of the 32 bits counter.

Note 52: The block size (64bits or 128 bits) of the selected mode of operation (CBC, CTR) is configured from the block size of the selected algorithm (AES, DES, Triple DES). For the AES algorithm the block size is 128 bits and for the DES/Triple DES algorithm the block size is 64 bits.

Table 166: CRYPTO_START_REG (0xFF0804)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
31-1	-	-	Reserved	0
0	W	CRYPTO_START	Write 1 to start an encryption/decryption operation. This register is auto-cleared.	0

Table 167: CRYPTO_STATUS_REG (0xFF0814)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
31-1	-	-	Reserved	0
0	R	CRYPTO_INACTIVE	0 – The Crypto IP is active. The encryption/decryption is in progress. 1 – The Crypto IP is inactive. The encryption/decryption has finished.	0

Table 168: DEBUG_REG (0xFF5004)

Bit	Mode	Symbol	Description	Reset
15-9	-	-	Reserved	0
7	R/W	SW_RESET	Software reset. 1 = the SC14452 puts all its on chip peripherals and registers in the reset state and CR16Cplus starts executing at address indicated by PC_START_REG. Registers that are not reset upon a S/W reset can be found in Registers reset overview (table 22, page 35)	0
6-5	-	-	Reserved	00
4	R/W	ENV_B01	0 = Normal boot code operation. 1 = Boot ROM always jumps to on-chip RAM address 0x8080. Used to return to an on-chip program after a SW_RESET without executing the whole boot sequence.	0 (Note 53)
3	R/W	CLK100_SRC	0 = CLK100 frequency is derived from 10 msec DIP clock. Note that if the DIP is off (URST=1) no CLK100_INT is generated. 1 = CLK100 frequency is derived from a continuous 10.66 msec clock.	0
2	R/W	CLK100_POS	0 = No CLK100_INT generated on rising edge. 1 = Generate CLK100_INT on the rising edge of CLK100	0
1	R/W	CLK100_NEG	0 = No CLK100_INT generated on falling edge. 1 = Generate CLK100_INT on the falling edge of CLK100	0
0	R	CLK100_EDGE	0 = CLK100 has generated an interrupt on falling edge. 1 = CLK100 has generated an interrupt on rising edge.	0

Note 53: ENV_B01 does not reset with SW_RESET, but only with a H/W reset

Table 169: DIP_STACK_REG (0xFF6000)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-8	-	-	Reserved	0
7-0	R	DIP_STACK	DIP Return address	0

Table 170: DIP_PC_REG (0xFF6002)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-9	-	-	Reserved	0
8	R	DIP_BANK	Read only DiP sequencer RAM bank bit	0
7-0	R	DIP_PC	DiP program counter Valid upon break or freeze only	0

Table 171: DIP_STATUS_REG (0xFF6004)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-8	-	-	Reserved	0
7	R	URST	Read only value of DIP_CTRL_REG[URST]	1
6	R	PRESALER	Read only value of DIP_CTRL_REG[PRESALER]	0
5	R	DIP_BRK_INT	Read only value of DIP_CTRL_REG[DIP_BRK_INT]	0
4	R	PD1_INT	Read only value of DIP_CTRL_REG[PD1_INT]	0
3-0	R	DIP_INT_VEC	Read only value of DIP_CTRL_REG[DIP_INT_VEC]	0

Table 172: DIP_CTRL_REG (0xFF6006)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-8	-	-	Reserved	0
7	R/W	URST	If this bit is set to 0 the DIP starts executing the DIP sequencer program. The first DIP instruction executed is located at address DIP_RAM+1. Writing a 1 to this bit stops the DIP sequencer program execution.	1
6	R/W	PRESALER	Global enable for P1 and P3 prescalers. If set to 1 and the DIP executes the <U_PSC> P3 divides by 16 and P1 divides by 16 if also XDIV =1 (leading to a total division of 256). Both P1 and P3 are switched off/on if this bit is set to 0/1. The execution of <U_INTx>, <U_VINT> or <U_VNMI> DIP commands switches P1 and P3 always back to divide to 1. (Note 54)	0
5	R/W	DIP_BRK_INT	If the DIP <BRK> command is executed the DIP stops executing the sequencer program and sets DIP_BRK_INT to 1. Also the DIP interrupt pending bit at the CR16C+ is set to 1. The DIP_BRK_INT bit is cleared on reading. Writing a 1 to it starts DIP program execution at the location where the BRK command was located.	0
4	R/W	PD1_INT	Generates a DIP interrupt if S-field preamble detected (PD 0->1). PD can be monitored on a GPIO pin. See also chapter 11.3 Reading this bit clears this bit and the DIP interrupt. Writing a 1 enables, a 0 disables this interrupt.	0
3-0	R	DIP_INT_VEC	If the DIP <U_INTx> (x = 0..3) or <U_VINT> command is executed this bit is set to 1 and the DIP interrupt pending is set to 1. Reading this register sets these bits to 0.	0000

Note 54: If CLK_AMBA_REG[CLK_DIV_PRE] = 1, P3 always divides by 16, regardless of PRESALER and <U_PSC>.

Note 55: The DIP_STATUS_REG is identical to DIP_CTRL_REG. If read, the DIP_INT the interrupt is not cleared.

Note 56: In case a DIP_BRK_INT has occurred, the DIP can only be reset by writing 0xA0, instead of 0x80.

Table 173: DIP_CTRL1_REG (0xFF600A)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-4	-	-	Reserved	0
3-0	R	DIP_VNMI_VEC	If the DIP <U_VNMI> command is executed the operand is ORed with these bits. Reading these bits sets the bits to 0.	0

Note 57: The DIP_STATUS_REG is identical to DIP_CTRL_REG. If read, the DIP_INT the interrupt is not cleared

Table 174: DIP_STATUS1_REG (0xFF6008)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-4	-	-	Reserved	0
3-0	R	DIP_VNMI_VEC	Read only value of DIP_CTRL1_REG[DIP_VNMI]	0

Table 175: DIP_CTRL2_REG (0xFF600E)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-11	-	-	Reserved	0
10	R/W	MFR_DSC	<D_LDK> parameters selection. 0 = IV[0-27] read from RAM (at DC0 to DC3[0-3]) 1 = IV[0-3] read from DIP_DC01_REG[0-3] IV[4-27] read from DIP_DC01_REG[4-15] and DIP_DC23_REG[0-11] IV[28-63] are hard coded '0' CK[0-63] is always read from RAM (at DC8 to DC15)	0
9	R/W	FR_BMC	<B_RC> MFR[2-0] parameters load selection 0 = MFR[0-2] read from RAM (at RC6[0-2]) 1 = MFR[0-2] read from DIP_DC01_REG[0-2]	0
8	R/W	SLOTCNT_RES	Slot counter resume enable after <WSC> and rising edge PCM_CLK 0 = Resume on "OR" of RAMOUT[3-0]_SYNC and RAMIN[3-0]_SYNC] (compatible with SC1443x, Sc1442x) 1 = Resume if main counter resumes (Recommended)	0
7	R/W	EN_8DIV9	0 = Normal mode 1 = 8/9 mode. Used with 9.216 MHz xtal. FDIP = 9.216 MHz, 20 Slots are used, bitrate is 1.024 Mbit/s. Select PER10_DIV=7 for correct UART frequencies and PER10_DIV=8 for ACCESS frequencies.	0
6	-	-	Reserved	0
5	R	DIP_BRK	DIP Break status 0 = No DIP <BRK> executed. 1 = The DIP has stopped due to the execution of a DIP <BRK>	0
4	R	PD1_INT	Reports the value written to DIP_CTRL_REG[PD1_INT] 0 = PD1 interrupt is disabled 1 = PD1 interrupt is enabled.	0
3	-	-	Reserved	0
2	R	BRK_PRE_OVR	0 = No DIP <BRK> issued while prescaler P3 was active. 1 = The DIP <BRK> is issued while prescaler P3 was active. This causes that prescaler P3 is automatically set to divide by 1. If DIP_CTRL_REG[DIP_BRK_INT] is set to 1, the prescaler is again set to divide by 16.	0
1	R	PRE_ACT	Prescaler status 0 = Prescaler P3 not active 1 = Prescaler P3 is active	0
0	R/W	DBUF	0 = Automatic double buffering of B-field data disabled 1 = Automatic double buffering of B-field data enabled. Refer to <A_TX>, <A_RX> for buffer selection. Modulo values of buffer pairs are the same and determined by the modulo value of the first pointer value: ARMOD0 for Rx buffer pair 0/2, ARMOD1 for Rx buffer pair 1/3 AWMOD0 for Tx buffer pair 0/2, AWMOD1 for Tx buffer pair 1/3	0

Table 176: DIP_MOD_SEL_REG (0xFF6012)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-8	-	-	Reserved	0
7	R/W	AWMOD3	ADPCM Modulo register selection	0
6	R/W	AWMOD2	0 = DIP_MODVAL_REG[MODULO0] value selected 32kbit/s@10ms = 320bits@10ms = 40 bytes@10ms (G726)	0
5	R/W	AWMOD1		0
4	R/W	AWMOD0	1 = DIP_MODVAL_REG[MODULO1] value selected 64kbit/s@10ms = 640bits@10ms = 80 bytes@10ms (G722)	0
3	R/W	ARMOD3		0
2	R/W	ARMOD2		0
1	R/W	ARMOD1		0
0	R/W	ARMOD0		0

Table 177: DIP_MOD_VAL_REG (0xFF6014)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-8	R/W	MODULO1	DIP ADPCM and B-field Modulo counter. (Nr of bits/slot/8)-1.	0x4F
7-0	R/W	MODUL00	Eg. 640 bits@10ms, (640/8) -1 = 79 = 0x4F Eg. 320 bits@10ms, (320/8) -1 = 39 = 0x27	0x27

Table 178: DIP_DC01_REG (0xFF6016)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-4	R/W	MFRAME_LOW	Multiframe number MFR[11-0] (IV[15-4]). Used by <D_LDK>	0
3-0	R/W	FRAME	Frame number FR[3-0] (IV[3-0]). Used by <D_LDK> Frame number FR[2-0]. Used by <B_RC>	0

Table 179: DIP_DC23_REG (0xFF6018)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-12	-	-	Reserved	0
11-0	R/W	MFRAME_HIGH	Multiframe number MFR[23-12] (IV[27-16]). Used by <D_LDK>	0

Table 180: DIP_SLOT_NUMBER_REG (0xFF600C)

Bit	Mode	Symbol	Description	Reset
15-5	-	-	Reserved	0
4-0	R	SLOT_CNTER	Read only actual value of DiP slot counter	00000

**Table 181: DMA0_A_STARTL_REG, DMA1_A_STARTL_REG, DMA2_A_STARTL_REG,
DMA3_A_STARTL_REG (0xFF4400, 0xFF4410, 0xFF4420, 0xFF4430)**

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-0	R/W	DMAx_A_STARTL	Source Start address, bits 15-0	0

**Table 182: DMA0_A_STARTH_REG, DMA1_A_STARTH_REG, DMA2_A_STARTH_REG,
DMA3_A_STARTH_REG (0xFF4402, 0xFF4412, 0xFF4422, 0xFF4432)**

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-9	-	-	Reserved	0
8-0	R/W	DMAx_A_STARTH	Source Start address, bits 24-16	0

Table 183: DMA0_B_STARTL_REG, DMA1_B_STARTL_REG, DMA2_B_STARTL_REG, DMA3_B_STARTL_REG (0xFF4404, 0xFF4414, 0xFF4424, 0xFF4434)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-0	R/W	DMAx_B_STARTL	Destination start address, bits 15-0	0

Table 184: DMA0_B_STARTH_REG, DMA1_B_STARTH_REG, DMA2_B_STARTH_REG, DMA3_B_STARTH_REG (0xFF4406, 0xFF4416, 0xFF4426, 0xFF4436)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-9	-	-	Reserved	0
8-0	R/W	DMAx_B_STARTH	Destination start address, bits 24-16.	0

Table 185: DMA0_IDX_REG, DMA1_IDX_REG, DMA2_IDX_REG, DMA3_IDX_REG (0xFF440E, 0xFF441E, 0xFF442E, 0xFF443E)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-0	R	DMAx_IDX	This internal register, added to DMA_A_STARTL/H_REG and DMA_B_STARTL/H_REG, determines the source/destination address of the next DMA cycle. The register value is multiplied according to the AINC and BINC and BW values before it is added DMA_A_STARTL/H_REG and DMA_B_STARTL/H_REG	0

Table 186: DMA0_INT_REG, DMA1_INT_REG, DMA2_INT_REG, DMA3_INT_REG (0xFF4408, 0xFF4418, 0xFF4428, 0xFF4438)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-0	R/W	DMAx_INT	Number of transfers until an interrupt is generated. if DINT_MODE must be set to 1.	0

Note 58: If an interrupt occurs this register can be used to distinguish between the DMA or basic interrupt source.

Table 187: DMA0_LEN_REG, DMA1_LEN_REG, DMA2_LEN_REG, DMA3_LEN_REG (0xFF440A, 0xFF441A, 0xFF442A, 0xFF443A)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-0	R/W	DMA_LEN	Transfer length in transfers (Byte (8 bits) , Halfword (16 bits), Word (32 bits).	0

Table 188: DMA0_CTRL_REG, DMA1_CTRL_REG, DMA2_CTRL_REG, DMA3_CTRL_REG (0xFF440C, 0xFF441C, 0xFF442C, 0xFF443C)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-14	-	-	Reserved	0
13	R/W	INIT_VAL	Determines the write value that is used when MEM_INIT=1. 0 = set all bit values to "0". 1= set all bit values to "1".	0
12	R/W	MEM_INIT	When set to "1" it desables the read accesses from the source address (DMAx_A_START) and performs only the write accesses (DMAx_B_START). The INIT_VAL register bit determines the value that is used during those write accesses. It is possible to use this feature only when DREQ_MODE=0.	0

**Table 188: DMA0_CTRL_REG, DMA1_CTRL_REG, DMA2_CTRL_REG, DMA3_CTRL_REG
(0xFF440C, 0xFF441C, 0xFF442C, 0xFF443C)**

BIT	MODE	SYMBOL	DESCRIPTION	RESET
11-10	R/W	MAX_BURST	<p>Specify the burst length. The DMA starts using the specified burst length as long as the number of the remaining data is larger than the burst length, otherwise it uses INCREMENTal burst.</p> <p>This value is ignored when DREQ_MODE=1, i.e. when the DMA channel is triggered from a hardware source. When DREQ_MODE=1 only SINGLE access is used.</p> <p>00 = Single access 01 = 4-beat access 10 = 8-beat access 11 = Reserved</p>	0
9-8	R/W	DMA_PRIO	<p>Set priority level of DMA channel to determine which DMA channel will be activated in case more than one DMA channel requests DMA.</p> <p>00 = lowest priority 01 = 10 = 11 = highest priority.</p> <p>If two or more priorities have the same level, an inherent priority is valid:</p> <p>Channel0: SPI Rx: 3 Channel1: SPI Tx: 1 Channel2: UART Rx: 2 Channel3: UART Tx: 0</p> <p>When two or more DMA channels have the same priority, then the DMA with the larger number has larger priority. For example, if both DMA0 and DMA1 have priority 0, then DMA1 has the highest priority.</p>	0
7	R/W	CIRCULAR	<p>0 = Normal mode. The DMA transfer stops the transfer after length DMAx_LEN_REG</p> <p>1 = Circular mode. The DMA channel repeats the transfer after length DMA_LEN_REGx with the initial register values DMAx_A_START_REG, DMAx_B_START_REG, DMAx_LEN_REG, DMAx_INT_REG. (Only works if DREQ_MODE=1)</p>	0
6	R/W	AINC	<p>Enable increment of source address.</p> <p>0 = do not increment 1 = increment according value of BW</p>	0
5	R/W	BINC	<p>Enable increment destination address</p> <p>0 = do not increment 1 = increment according value of BW</p>	0
4	R/W	DREQ_MODE	<p>0 = DMA channel starts immediately. 1 = DMA channel can be triggered by hardware interrupt source: Channel 0: SPI1_INT (BW must be set to 01: 16 bits) Rx Channel 1: SPI1_INT (BW must be set to 01: 16 bits) Tx KEY_INT_EN_REG[DREQ1_CTRL] selects either ADC_SYNC or SPI_INT. Channel 2: UART RI (BW must be set to 00: 8 bits) Channel 3: UART TI (BW must be set to 00: 8 bits)</p>	0
3	R/W	DINT_MODE	<p>0 = ICU gets interrupt from Hardware interrupt source 1 = ICU gets interrupt from DMA channel (0-3) See Figure 53</p>	0

**Table 188: DMA0_CTRL_REG, DMA1_CTRL_REG, DMA2_CTRL_REG, DMA3_CTRL_REG
(0xFF440C, 0xFF441C, 0xFF442C, 0xFF443C)**

BIT	MODE	SYMBOL	DESCRIPTION	RESET
2-1	R/W	BW	Bus transfer width: 00 = Byte (8 bits, used for UART) 01 = Halfword (16 bits, used for SPI) 10 = Word (32 bits, used e.g for SW transfer) 11 = reserved	0
0	R/W	DMA_ON	0 = DMA channel is off, clocks are disabled 1 = Enable DMA channel. This bit will be automatically cleared after a complete transfer. In circular mode this bit stays set.	0

Table 189: DSP_ADC1S_REG (0x101FF86)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-0	R	DSP_ADC1S	Gen2DSP Shared ADC1 register (2's complement value) ADC0_REG - 0x0200 = 7x(not)D9, D8-D0	0xFE00

Table 190: DSP_ADC0S_REG (0x101FF88)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-0	R	DSP_ADC0S	Gen2DSP Shared ADC0 register (2's complement value) ADC1_REG - 0x0200 = 7x(not)D9, D8-D0	0xFE00

Table 191: DSP_CLASSD_REG (0x101FF8A)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-0	R/W	DSP_CLASSD	CLASSD Output data	0

Table 192: DSP_CLASSD_BUZZOFF_REG (0x101FFBE)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15	R/W	BUZZOFF	0 = CLASSD buzzer gain amplifier on 1 = CLASSD buzzer gain amplifier off.	1
14-0	-	-	Reserved	0

Table 193: DSP_CODEC_MIC_GAIN_REG (0x101FF8C)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-12	R/W	DSP_MIC_GAIN	Gen2DSPx controlled Microphone amplifier gain in steps of 2 dB 0000 = 0 dB, 1111 = +30 dB To enable DSP controlled gain, set CODEC_MIC_REG [DSP_CTRL] =1	0
11-0	-	-	Reserved	0

Table 194: DSP_CODEC_OUT_REG (0x101FF8E)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-0	R/W	DSP_CODEC_OUT	Codec output data	0

Table 195: DSP_CODEC_IN_REG (0x101FF90)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-0	R	DSP_CODEC_IN	Codec input data	0

Table 196: DSP1_CTRL_REG, DSP2_CTRL_REG (0x101FFD0, 0x102FFD0)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-7	-	-	Reserved	0
8	R/W	DBG_EN	0 = Disable Gen2DSPx Debug module 1 = Enable Gen2DSPx Debug module (done by SDI)	0
7-5	-	-	Reserved	0
4	R/W	DSP_CR16_INT	CR16C+ interrupt request/start-up request to Gen2DSPx 0 = Idle 1 = Generate interrupt to the Gen2DSPx. As soon as the Gen2DSPx enters this interrupt routine, this bit is automatically cleared.	0
3	R/W	DSP_CLK_EN	Gen2DSPx clock enabled 0 = Gen2DSPx clock disabled if <WTF> executed (power saving) 1 = Gen2DSP Clock always on. (DSPx_CLK_SEL Consumes more power, but giving a static supply load on VDD which might reduce TDD noise.)	0
2	R/W	DSP_EN	0 = Gen2DSPx in power down 1 = Gen2DSPx out of reset.	0
1-0	-	-	Reserved	

Table 197: DSP1_INT_REG, DSP2_INT_REG (0x101FFD8, 0x102FFD8)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-0	R/W	VECTOR	The <int_vector> parameter of the Gen2DSP's <IRQ> instruction is "ored" with VECTOR bits. If one or more bits are set to 1 a DSPx_INT to CR16C+ is set. M_VECTOR bits only disables interrupts to ICU; they are always visible in the VECTOR bits if set. Register bits are cleared if a '1' is written to the corresponding bits. Writing a '1' to a bit that is already '1', clears the bit. Writing a '1' to a bit that changes from '0' to '1' does not clear the register bit.	0

Table 198: DSP1_INT_MASK_REG, DSP2_INT_MASK_REG (0x101FFDA, 0x102FFDA)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-0	R/W	M_VECTOR	Mask DSPx_INT_REG[VECTOR] bits 0 = Disable interrupt 1 = Enable interrupt.	0

Table 199: DSP1_INT_PRIO1_REG, DSP2_INT_PRIO1_REG (0x101FFDC, 0x102FFDC)

Bit	Mode	Symbol	Description	Reset
15-11	-		DSPx Interrupt priority mux 1.	0
10-8	R/W	DSP_INT0_PRIO	0 = DSP_SYNC0 1 = DSP_SYNC1 2 = DSP_SYNC2	7
7	-		3 = Neighbouring Gen2DSP trigger	0
6-4	R/W	DSP_INT1_PRIO	4 = DIP trigger	7
3	-		5 = CR16C+ trigger	0
2-0	R/W	DSP_INT2_PRIO	6 = Reserved 7 = Output disabled Example: DSP_INT0_PRIO = 5: Give CR16Cplus the highest priority DSP_INT4_PRIO = 0: Give DSP_SYNC0 the lowest priority	7

Table 200: DSP1_INT_PRIO2_REG, DSP2_INT_PRIO2_REG (0x101FFDE, 0x102FFDE)

Bit	Mode	Symbol	Description	Reset
15-11	-	-	DSPx Interrupt priority mux 2	0
10-8	R/W	DSP_INT3_PRIO	0 = DSP_SYNC0 1 = DSP_SYNC1 2 = DSP_SYNC2	7
7	-	-	3 = Neighbouring Gen2DSP trigger	0
6-4	R/W	DSP_INT4_PRIO	4 = DIP trigger	7
3	-	-	5 = CR16C+ trigger	0
2-0	R/W	DSP_INT5_PRIO	6 = Reserved 7 = Output disabled	7

Table 201: DSP1_IRQ_START_REG, DSP2_IRQ_START_REG (0x101FFD6, 0x102FFD6)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-4	R/W	DSPx_IRQ_START	Bits 15-4 of DSPx Interrupt Start address. Bits 3-0 are ORed with interrupt source: - Interrupt 0: jumps to DSPx_IRQ_START - Interrupt 1: jumps to DSPx_IRQ_START + 2 - Interrupt 2: jumps to DSPx_IRQ_START + 4 - Interrupt 3: jumps to DSPx_IRQ_START + 6 - Interrupt 4: jumps to DSPx_IRQ_START + 8 - Interrupt 5: jumps to DSPx_IRQ_START + 10	0
3-0	-	-	Reserved	0

Table 202: DSP_MAIN_SYNC0_REG (0x101FF80)

Bit	Mode	Symbol	Description	Reset
15-14	R/W	RAMOUT3_SYNC	RAMIO and ADPCM Write/Read Tick Frequency.	11
13-12	R/W	RAMOUT2_SYNC	RAMOUTx -> A_LDW -> B_TX -> TDO	11
11-10	R/W	RAMOUT1_SYNC	RDI -> B_RX -> A_LDR-> RAMINx	11
9-8	R/W	RAMOUT0_SYNC	00 = 4 kHz (E.g. G.726@8 kHz)	11
7-6	R/W	RAMIN3_SYNC	01 = 8 kHz (E.g. G.722@16kHz, G.726@16kHz)	11
5-4	R/W	RAMIN2_SYNC	10 = 16kHz	11
3-2	R/W	RAMIN1_SYNC	11 = RAMIO interface in Power-down	11
1-0	R/W	RAMIN0_SYNC		11

Table 203: DSP_MAIN_SYNC1_REG (0x101FF82)

Bit	Mode	Symbol	Description	Reset
15-14	R/W	PCM_SYNC	PCM FSC output frequency. In PCM slave mode, this register must be set equal to the PCM_FSC input frequency. The phase difference between the internal and external PCM_FSC is calculated in the DSP_PHASE_INFO_REG. 00 = 8 kHz 01 = 16kHz 10 = 32kHz 11 = PCM IO interface in Power-down	11
13-12	R/W	ADC_SYNC	ADC start frequency in automatic mode (May also be used the start a DMA transfer) 00 = 8 kHz 01 = 16kHz <- default for two conversions in 8 kHz frame 10 = 32kHz <- not allowed, test purpose only 11 = ADC IO interface in Power-down	11
11-10	R/W	DA_LSR_SYNC	Codec DA sample clock to loudspeaker 00 = 8 kHz 01 = 16kHz 10 = 32kHz 11 = Codec DA lsr interface in Power-down	11
9-8	R/W	DA_CLASSD_SYNC	Codec DA sample clock to CLASSD amplifier 00 = 8 kHz 01 = 16kHz 10 = 32kHz 11 = Codec DA CLASSD interface in Power-down	11
7-6	R/W	AD_SYNC	Codec AD sample clock from microphone 00 = 8 kHz 01 = 16kHz 10 = 32kHz 11 = Codec AD interface in Power-down	11
5-4	R/W	DSP_SYNC2	DSP 1 and 2 start/interrupt 2 00 = 8 kHz 01 = 16kHz 10 = 32kHz 11 = 8 kHz inverted	0
3-2	R/W	DSP_SYNC1	DSP 1 and 2 start/interrupt 1 00 = 8 kHz 01 = 16kHz 10 = 32kHz 11 = 8 kHz inverted	0
1-0	R/W	DSP_SYNC0	DSP 1 and 2 start/interrupt 0 00 = 8 kHz 01 = 16kHz 10 = 32kHz 11 = 8 kHz inverted	0

Note 59: Above frequencies are generated if DSP_MAIN_CNT_REG[DSP_MAIN_REL] = 0x8F and clocked with 1.152 MHz. The main counter then outputs 8kHz, 16kHz and 32kHz automatically.

Table 204: DSP_MAIN_CTRL_REG (0x101FFBC)

Bit	Mode	Symbol	Description	Reset
15-10	-	-	Reserved	
9-8	R/W	DSP_MAIN_CTRL	<p>Main counter on/off control.</p> <p>00 = Main counter in reset.</p> <p>01 = Main counter out of reset, free running.</p> <p>For DIP synchronisation later, toggle bits 01 -> 00 -> 10 and wait for new <A_NORM> instruction (Non DIP synchronised start-up for tone generation)</p> <p>10 = Main counter starts on <A_NORM> instruction at value DSP_MAIN_PRESET.</p> <p>For a new DIP synchronisation toggle bits. 00 -> 10 and wait for new <A_NORM> (Note 60) (DIP synchronised start-up for voice connection)</p> <p>11 = reserved</p>	10
7-0	R/W	DSP_MAIN_PRES ET	<p>Main counter reset and preset value. Determines time between <A_NORM> instruction and first ADPCM tick.</p> <p>Value 0: Main counter counts: 0..DSP_MAIN_REL, 0..DSP_MAIN_REL (e.g. 0..143, etc.)</p> <p>Values > 0: Main counter counts: DSP_MAIN_PRESET..255, 0..DSP_MAIN_REL, 0..DSP_MAIN_REL, etc.</p> <p>During the time between DSP_MAIN_PRESET...255 no 8, 16 or 32 kHz syncs are generated.</p> <p>Values for compatibility: 15 for SC14430. 1 for SC14434 (See AN-D-140)</p>	175

Note 60: For proper synchronisation to the RAMINx_REG and RAMOUTx_REG, it is required that the Gen2DSP BUF_PACK/BUF_UNPACK buffers are also reset by software. For more information See AN-D-140 "chapter Synchronisation"

Table 205: DSP_MAIN_CNT_REG (0x101FF84)

Bit	Mode	Symbol	Description	Reset
15-8	R/W	DSP_MAIN_REL	<p>Main counter reload value.</p> <p>For all sample rates, the main counter must always be clocked with CLK_SPU2_REG[SW_MAINCDC_DIV] = 1.152 MHz. With this frequency 4, 8, 16 and 32 kHz sync signals are always generated. and DSP_MAIN_SYNC1_REG are used to select the appropriate sync signal.</p> <p>The reload values is calculated as follows: DSP_MAIN_REL = (Main counter clock/8000)-1 = (1.152 MHz /8000)-1 = 144-1 = 0x8F</p>	0x8F
7-0	R	DSP_MAIN_CNT	Read only value of DSP main counter	0

Table 206: DSP1_OVERFLOW_REG, DSP2_OVERFLOW_REG (0x101FFE0, 0x102FFE0)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15	R/W	M_IRQ_OVERFLOW	Mask IRQ_OVERFLOW 0 = Disable 1 = Generate DSP_INT to CR16Cplus if IRQ_OVERFLOW bit is set	0
14	R/W	M_WTF_OVERFLOW	Mask WTF_OVERFLOW 0 = Disable 1 = Generate DSP_INT to CR16Cplus if WTF_OVERFLOW bit is set	0
13-8	R/W	M_INT_OVERFLOW	Mask INT_OVERFLOW[x] 0 = Disable 1 = Generate DSP_INT to CR16Cplus if INT_OVERFLOW[x] bit is set	0
7	R/W	IRQ_OVERFLOW	0 = No IRQ overflow 1 = Interrupt request overflow. New DSP_TRIG[5-0] occurred before corresponding internal DSP_INT[5-0] request was cleared. The DSP_INT[5-0] are cleared automatically as soon as the ISR is started. Register bit is cleared if a '1' is written. Writing a '1' to a bit that is already '1', clears the bit. Writing a '1' to a bit that changes from '0' to '1' does not clear the register bit.	0
6	R/W	WTF_OVERFLOW	0 = No WTF overflow 1 = WTF overflow. New DSP_TRIG[5-0] occurred before <WTF> was executed. This is the 43x compatible overflow condition without interrupts. Register bit is cleared if a '1' is written. Writing a '1' to a bit that is already '1', clears the bit. Writing a '1' to a bit that changes from '0' to '1' does not clear the register bit.	0
5-0	R/W	INT_OVERFLOW	0 = No Gen2DSP interrupt overflow 1 = Interrupt routine 5-0 overflow. New DSP_TRIG[5-0] occurred before <ISR_ACK[x]> executed. Register bits are cleared if a '1' is written to the corresponding bits. Writing a '1' to a bit that is already '1', clears the bit. Writing a '1' to a bit that changes from '0' to '1' does not clear the register bit. (Note 61)	0

Note 61: If Gen2DSP interrupts are disabled with (IM[5-0] = 0), the INT_OVERFLOW[x] will also be set when the <ISR_ACK[x]> instruction is not executed before the new DSP_TRIG[x] occurs. To disable DSP_INT generation, M_INT_OVERFLOW[x] can be set to 0.

Table 207: DSP1_JTBL_REG, DSP2_JTBL_REG (0x101FFE2, 0x102FFE2)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-9	R/W	DSPx_JTBL	Microcode jump table start. Reset value points to Microcode ROM	0x4000
8-0	-	-	Reserved	0

Table 208: DSP1_PC_REG, DSP2_PC_REG (0x101FFD2, 0x102FFD2)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-0	R	DSPx_PC	DSPx Program counter relative to the Micro Code RAM Address. Read only for debug purposes only	0

Table 209: DSP1_PC_START_REG, DSP2_PC_START_REG (0x101FFD4, 0x102FFD4)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-0	R/W	DSPx_PC_START	DSPx Start address. Value 0 points to Micro Code RAM/ROM address 0. This register is loaded into the Gen2DSP Program counter from start-up.	0

**Table 210: DSP_PCM_OUT0_REG, DSP_PCM_OUT1_REG, DSP_PCM_OUT2_REG, DSP_PCM_OUT3_REG
(0x101FFA6, 0x101FFA8, 0x101FFAA, 0x101FFAC)**

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-0	R/W	DSP_PCM_OUTx	PCM output channel x 16 bits	0xFFFF

**Table 211: DSP_PCM_IN0_REG, DSP_PCM_IN1_REG, DSP_PCM_IN2_REG, DSP_PCM_IN3_REG
(0x101FFAE, 0x101FFB0, 0x101FFB2, 0x101FFB4)**

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-0	R	DSP_PCM_INx	PCM input channel x 16 bits	0

Table 212: DSP_PCM_CTRL_REG (0x101FFB6)

Bit	Mode	Symbol	Description	Reset
15-8	-	-	Reserved	0
7	R/W	PCM_CLKINV	0 = PCM_CLK Rising edge 1 = PCM_CLK Falling edge	0
6	R/W	PCM_PPOD	0 = PCM_DOUT is push-pull 1 = PCM_DOUT is open drain, requires external pull-up	0
5	R/W	PCM_FSC0DEL	0 = PCM_FSC starts 1 data bit before MSB bit 1 = PCM_FSC starts at the same time as MSB bit	0
4-3	R/W	PCM_FSC0LEN	If MASTER = 1 00 = length of PCM_FSC equal to 1 data bit 01 = length of PCM_FSC equal to 8 data bits 10 = length of PCM_FSC equal to 16 data bits 11 = length of PCM_FSC equal to 32 data bits	00
2	R/W	DSP_PCM_SYNC	Main counter PCM synchronization if PCM slave 0 = Count enable always true. 1 = Count enable upon PCM_FSC rising edge	0
1	R/W	PCM_MASTER	PCM interface master/slave mode 0 = Slave mode 1 = Master mode	0
0	R/W	PCM_EN	0 = PCM interface down 1 = PCM interface enable.	0

Table 213: DSP_PHASE_INFO_REG (0x101FFB8)

Bit	Mode	Symbol	Description	Reset
15-8	-	-	Reserved	0
7-0	R	PHASE_INFO	Difference between rising edge of PCM_FSC and internal main-counter PCM_SYNC (in units of 1/(Main counter clock) determined by CLK_MAIN_SEL)	0

**Table 214: DSP_RAM_OUT0_REG, DSP_RAM_OUT1_REG, DSP_RAM_OUT2_REG, DSP_RAM_OUT3_REG
(0x101FF92, 0x101FF94, 0x101FF96, 0x101FF98)**

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-8	-	-	Reserved	0
7-0	R/W	DSP_RAM_OUTx	ADPCM output data to minimum delay buffers in shared RAM. The ADPCM pointers to shared RAM 1 or 2 are handled by the DIP instructions A_LDWx See AN-D-140 DIP commands manual.	0

**Table 215: DSP_RAM_IN0_REG, DSP_RAM_IN1_REG, DSP_RAM_IN2_REG, DSP_RAM_IN3_REG
(0x101FF9A, 0x101FF9C, 0x101FF9E, 0x101FFA0)**

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-8	-	-	Reserved	0
7-0	R	DSP_RAM_INx	ADPCM input data from minimum delay buffers in shared RAM. The ADPCM pointers to shared RAM 1 or 2 are handled by the DIP instructions A_LDRx. See AN-D-140 DIP commands manual.	0

Table 216: DSP_VQI_REG (0x101FFBA)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15-4	-	-	Reserved	0
3-0	R/W	BVQI_ON	BMC VQI bit "OR"ed with BVQI_ON(x) bit. The values of this bit can be used by the SCP to start VQI (voice quality improvement in the Gen2DSPx) Read: 0 = RDI data bit was '0' upon execution of <B_VQI> <i> and BVQI_ON(i) bit = '0' 1 = RDI data bit was '1' upon execution of <B_VQI> <i> or BVQI_ON(i) bit = '1' Write: Set BVQI_ON(i) bit.	0

Table 217: DSP_ZCROSS1_OUT_REG, DSP_ZCROSS2_OUT_REG (0x101FFA2, 0x101FFA4)

BIT	MODE	SYMBOL	DESCRIPTION	RESET
15	R/W	DSP_ZCROSSx	Common Gen2DSP1 Gen2DSP2 Output register ECZ1 and ECZ2 to Capture timer inputs. Signals ECZ1 and ECZ2 can also be used and general purpose GPIO port pins for diagnostics purpose.	0
14-0	-	-	Reserved	

Table 218: EBI_SDEXN_MODE_REG (0xFF00AC)

Bit	Mode	Symbol	Description	Reset
31-0	R/W	SDEXN	Mobile SDRAM Extended mode registers. Value Depends on used Mobile SDRAM. When using Normal SDRAM, this register must be the same as the SDRAM mode register 31-7 = 0 6-4 = EBI_SDCTRL_REG[CAS_LATENCY] value 3 = 0 2-0 = 010	0

Table 219: EBI_SDCONR_REG (0xFF0000)

Bit	Mode	Symbol	Description	Reset
31-21	-	-	Reserved	0
20-15	-	-	Reserved	40
14-13	R/W	S_DATA_WIDTH	Specifies SDRAM data width in bits 00 = 16 bits	0
12-9	R/W	S_COL_ADDR_WIDTH	Number of address bits for column address -1; 15 = reserved 14-7 = correspond to 15-8 bits 6-0 reserved	8
8-5	R/W	S_ROW_ADDR_WIDTH	Number of address bits for row address-1; 15-10= correspond to 16-11 bits 10-0 = reserved	12
4-3	R/W	S_BANK_ADDR_WIDTH	Number of bank address bits-1; 0-3 = correspond to 1-4 bits, and therefore select 2-16 banks	1
2-0	-	-	Reserved	0

Table 220: EBI_SDTMG0R_REG (0xFF0004)

Bit	Mode	Symbol	Description	Reset
31-27	R/W	T_XSR_H	Bits 8-4 of the T_XSR register field	0
26	R/W	CAS_LATENCY_H	Bit 2 is CAS_LATENCY_H register field Bit 1-0 is CAS_LATENCY register field	0
25-22	R/W	T_RCSD	Active-to-active command period; Values of 0-15 correspond to T_RCSD of 1-16 clocks.	7
21-18	R/W	T_XSR	Bits 3-0 of the T_XSR register field Exit self-refresh to active or auto-refresh command time; minimum time controller should wait after taking SDRAM out of self-refresh mode before issuing any active or auto-refresh commands; Values 0-511 correspond to t_xsr of 1-512 clocks	7
17-14	R/W	T_RCAR	Auto-refresh period; minimum time between two auto-refresh commands; Values 0-15 correspond to t_rcar of 1-16 clocks.	7
13-12	R/W	T_WRSD	For writes, delay from last data in to next precharge command; Values 0-3 correspond to T_WRSD of 1-4 clocks	1
11-9	R/W	T_RP	Precharge period; Values of 0-7 correspond to t_rp of 1-8 clocks	2
8-6	R/W	T_RCD	Minimum delay between active and read/write commands; Values 0-7 correspond to t_rcd values of 1-8 clocks	1

Table 220: EBI_SDTCMG0R_REG (0xFF0004)

Bit	Mode	Symbol	Description	Reset
5-2	R/W	T_RAS_MIN	Minimum delay between active and precharge commands; Values of 0-15 correspond to T_RAS_MIN of 1-16 clocks	4
1-0	R/W	CAS_LATENCY	Bit 2 is CAS_LATENCY_H register field Bit 1-0 is CAS_LATENCY register field Delay in clock cycles between read command and availability of first data: 0 = 1 clock 1 = 2 clocks 2 = 3 clocks 3 = 4 clocks 4, 5, 6, 7 = reserved	1

Table 221: EBI_SDTCMG1R_REG (0xFF0008)

Bit	Mode	Symbol	Description	Reset
31-22	-	-	Reserved	0
21-20	-	-	Reserved	0
19-16	R/W	NUM_INIT_REF	Number of auto-refreshes during initialization; Values 0-15 correspond to 1-16 auto-refreshes	7
15-0	R/W	T_INIT	Number of clock cycles to hold SDRAM inputs stable after power up, before issuing any commands.	8

Table 222: EBI_SDCTLR_REG (0xFF000C)

Bit	Mode	Symbol	Description	Reset
31-21	-	-	Reserved – must be set to all 0s at all times	0
20	-	-	Reserved	0
19	-	-	Reserved	0
18	-	-	Reserved	0
17	R/W	S_RD_READY_MODE	SDRAM read-data-ready mode; set to 1, indicates SDRAM read data is sampled after s_rd_ready goes active	0
16-12	R/W	NUM_OPEN_BANKS	Number of SDRAM internal banks to be open at any time; Values of 1-4 correspond to 0-3 banks open.	3
11	R	SELF_REFRESH_STATUS	0 = SDRAM not in self-refresh mode. 1 = SDRAM is in self-refresh mode. When “self_refresh/deep_power_mode” bit (bit 1 of SCTLR) is set, it may take some time before SDRAM is put into self-refresh mode, depending on whether all rows or one row are refreshed before entering self-refresh mode defined by full_refresh_before_sr bit Before gating clock in self-refresh mode, ensure this bit is set	0
10	-	-	Reserved	0
9	R/W	SET_MODE_REG	1 = Controller Forced to do update of SDRAM mode register; bit is cleared by controller once it has finished mode register update	0

Table 222: EBI_SDCTRL_REG (0xFF000C)

Bit	Mode	Symbol	Description	Reset
8-6	R/W	READ_PIPE	Indicates number of registers inserted in read data path for SDRAM in order to correctly latch data; 0 = no read pipe 1 = 1 read pipe register 2 = 2 read pipe registers, <- Recommended value 3,4,5,6,7 = not allowed. Note EBI_SDREFR_REG[READ_PIPE_CLK] must also be set.	1
5	R/W	FULL_REFRESH_AFTER_SR	Controls number of refreshes done by EBI after SDRAM is taken out of self-refresh mode: 1 = Refresh all rows before entering self-refresh mode 0 = Refresh just 1 row before entering self-refresh mode	0
4	R/W	FULL_REFRESH_BEFORE_SR	Controls number of refreshes done by EBI before putting SDRAM into self-refresh mode: 1 = Refresh all rows before entering self-refresh mode 0 = Refresh just one row before entering self-refresh mode	0
3	R/W	PRECHARGE_ALGO	Determines when row is precharged: 0 = Immediate precharge; row precharged at end of read/write operation 1 = Delayed precharge; row kept open after read/write operations	1
2	R/W	POWER_DOWN_MODE	Forces EBI to put SDRAM in power-down mode; bit 19 determines the type of power-down mode requested	0
1	R/W	SR_OR_DP_MODE	Forces EBI to put SDRAM in self-refresh mode. Bit can be cleared by writing to this bit or by clear_sr_dp pin, generated by external power management unit	0
0	R/W	INITIALIZE	Forces EBI to initialize SDRAM; bit reset to 0 by EBI once initialization sequence is complete	0

Table 223: EBI_SDREFR_REG (0xFF0010)

Bit	Mode	Symbol	Description	Reset
31-24	R	GPI	General purpose inputs. Reserved	0
23	R/W	GPO	General purpose output signals Reserved	0
22-20	R/W	READ_PIPE_MUX	Indicates number of registers inserted in read data Same value as EBSDCTRL_REG[READ_PIPE] Recommended value 2.	0
19	R/W	ACS3_IOEXP	0 = Normal ACS3 function 1 = ACS3 is ORed with WRn. This rising edge of ACS3 can now directly be used to clock in DAB[15-0] into an IO expander. See Figure 45 on page 86.	0
18-16	R/W	READ_PIPE_CLK	Read pipe clock delay. Represents the sample moment closest to 50% of the received SDRAM data. See Table 399 on page 324 0 = -1.22 ns 1 = 0.51 ns 2 = 2.60 ns <- recommended value 3 = 4.67 ns 4,5,6,7 = 6.93 ns	0
15-0	R/W	T_REF	Number of clock cycles between consecutive refresh cycles; for details on programming this register, refer to page 89	1040

Table 224: EBI_ACS0_LOW_REG, EBI_ACS1_LOW_REG, EBI_ACS2_LOW_REG, EBI_ACS3_LOW_REG, EBI_ACS4_LOW_REG (0xFF0014, 0xFF0018, 0xFF001C, 0xFF0020, 0xFF0024)

Bit	Mode	Symbol	Description	Reset
31-16	R/W	CS_BASE	Upper bits of chip select base. Determines chip select start address in block of 64kbyte. E.g. 0xF for ACS4 Startaddress 0xF0.000	0
15-11	R/W	EBI_RES	Reserved	0
10-0	-	-	Reserved	0

Table 225: EBI_ACS0_CTRL_REG, EBI_ACS1_CTRL_REG, EBI_ACS2_CTRL_REG, EBI_ACS3_CTRL_REG, EBI_ACS4_CTRL_REG (0xFF0054, 0xFF0058, 0xFF005C, 0xFF0060, 0xFF0064)

Bit	Mode	Symbol	Description	Reset
31-11	-	-	Reserved	0
10-8	R/W	REG_SELECT	Static memory Timing set. If MEM_TYPE = SDRAM, this field is don't care: 0 = Timing set 0 1 = Timing set 1 2 = Timing set 2	0,1,2,1,2
7-5	R/W	MEM_TYPE	Type of memory connected to corresponding Chip Select 0 = SDRAM 1 = SRAM or PSRAM 2 = FLASH	2,1,2,1,0
4-0	R/W	MEM_SIZE	Chip select range, added to CS_BASE 0 = no memory connected to chip select 1 = 64 kByte 2 = 128 kByte 3 = 256 kByte 4 = 512 kByte 5 = 1 MByte 6 = 2 Mbyte 7 = 4 Mbyte 8 = 8 Mbyte 9 = 16 Mbyte 10 = 32 Mbyte 11-17 = 64Mbyte to 4 GByte	0,0,0,0,0

Table 226: EBI_SMTMGR_SET0_REG, EBI_SMTMGR_SET1_REG, EBI_SMTMGR_SET2_REG (0xFF0094, 0xFF0098, 0xFF009C)

Bit	Mode	Symbol	Description	Reset
31-30	-	-	Reserved	0
29-28	R/W	SM_READ_PIPE	Number of inserted Read pipe registers for static memory devices. Value must be 0.	0, 0, 0
27	R/W	LOW_FREQ_SYNC		0, 0, 0
26	R/W	READY_MODE	Static memory Data bus ready timing. 0 = Static memory uses internal timing (T_RC) 1 = Static memory uses READY signal (P1[15]).	0, 0, 0

Table 226: EBI_SMTMGR_SET0_REG, EBI_SMTMGR_SET1_REG, EBI_SMTMGR_SET2_REG (0xFF0094, 0xFF0098, 0xFF009C)

Bit	Mode	Symbol	Description	Reset
25-24	R/W	EBI_PAGE_SIZE	PSRAM Page size 0 = 4 Words/page 1 = 8 Words/page 2 = 16 Words/page 3 = 32 Words/page	0, 0, 0
23	R/W	EBI_PAGE_MODE	0 = Device does not support page mode 1 = Device supports page mode.	0, 0, 1
22-19	R/W	T_PRC	Page mode read cycle time. Values of 0-15 correspond to a read cycle time of 1-16 HCLK cycles	0, 0, 3
18-16	R/W	T_BTA	Bus turn Around time. Static memory idle cycles between "read to write", or "write to read", or "read to read when chip-select changes" for memory data bus turn around time; Values of 0-7 correspond to 0-7 idle HCLK cycles.	1,3,1
15-10	R/W	T_WP	Write pulse width; Values of 0-63 correspond to write pulse width of 1-64 HCLK cycles	3,6,5
9-8	R/W	T_WR	Write address/data hold time; Values of 0-3 correspond to write address/data hold time of 0-3 HCLK cycles	1,1,1
7-6	R/W	T_AS	Write address setup time; values of 0-3 correspond to address setup time of 0-3 clock cycles; Value of 0 is only valid in case of Synchronous SRAM	1,1,1
5-0	R/W	T_RC	Number of HCLK wait states. Values of 0-63 wait states corresponds to a read cycle time of 1-64 clock cycles.	7,7,9

Table 227: EBI_FLASH_TRPDR (0xFF00A0)

Bit	Mode	Symbol	Description	Reset
31-12	-	-	Reserved	0
11-0	R/W	T_RPD	FLASH reset/power-down high to read/write delay; values correspond to sm_rp_n high to read/write delay minus one	200

Table 228: EBI_SMCTRLR_REG (0xFF00A4)

Bit	Mode	Symbol	Description	Reset
31-16	-	-	Reserved	0
15-13	R/W	SM_DATA_WIDTH_SET2	Width of Static memory data bus controlled by Static memory register SET2: 000 = 16 bits 100 = 8 bits 001, 010, 011 = Reserved	0
12-10	R/W	SM_DATA_WIDTH_SET1	Width of Static memory data bus controlled by Static memory register SET1: 000 = 16 bits 100 = 8 bits 001, 010, 011 = Reserved	0

Table 228: EBI_SMCTRLR_REG (0xFF00A4)

Bit	Mode	Symbol	Description	Reset
9-7	R/W	SM_DATA_WIDTH_SET0	Width of Static memory data bus controlled by Static memory register set 0: 000 = 16 bits 100 = 8 bits 001, 010, 011 = Reserved	0
6-4	-	-	Reserved	0
3-1	R/W	WP_N		0
0	R/W	sSM_RP_N	FLASH reset/power-down mode; after reset, controller internally performs a power-down for FLASH and then sets this bit to 1 To force FLASH to power-down mode during normal operation: 0 = Forces FLASH to power-down mode 1 = Takes FLASH out of power-down mode	1

Table 229: INT0_PRIORITY_REG (0xFF5404)

Bit	Mode	Symbol PR_LEVEL[2-0]	Description	Reset
15-11	-	-	The bits in INTx_PRIORITY_REG define the priority level of a maskable interrupt. The maskable interrupt with the interrupt pending bit set (1) and the highest priority, is serviced.	0
10-8	-	DSP2_INT_PRIO	After the interrupt is serviced the pending bit is NOT reset (0) by hardware. If the priority levels are equal, the inherent priority determines the priority (see Table 66)	0
7	-	-		0
6-4	R/W	DSP1_INT_PRIO		0
3	-	-		0
2-0	R/W	SPI2_INT_PRIO	PR_LEVEL[2] PR_LEVEL[1] PR_LEVEL[0]: Priority 0 0 0 0 Disabled 0 0 1 1 Lowest 0 1 0 2 0 1 1 3 1 0 0 4 1 0 1 5 1 1 0 6 1 1 1 7 Highest	0

Table 230: INT1_PRIORITY_REG (0xFF5406)

Bit	Mode	Symbol PR_LEVEL[2-0]	Description	Reset
15	-	-	The bits in INTx_PRIORITY_REG define the priority level of a maskable interrupt. The maskable interrupt with the interrupt pending bit set (1) and the highest priority, is serviced.	0
14-12	R/W	CRYPTO_INT_PRIO	After the interrupt is serviced the pending bit is NOT reset (0) by hardware. If the priority levels are equal, the inherent priority determines the priority (see Table 66)	0
11	-	-		0
10-8	R/W	DIP_INT_PRIO		0
7	-	-		0
6-4	R/W	CLK100_INT_PRIO	PR_LEVEL[2] PR_LEVEL[1] PR_LEVEL[0]: Priority 0 0 0 0 Disabled 0 0 1 1 Lowest	0
3	-	-		0
2-0	R/W	TIM1_INT_PRIO	0 1 0 2 0 1 1 3 1 0 0 4 1 0 1 5 1 1 0 6 1 1 1 7 Highest The CLK100 interrupt is also disabled if the DIP clock is selected and the DIP is stopped. See DEBUG_REG[3]	0

Table 231: INT2_PRIORITY_REG (0xFF5408)

Bit	Mode	Symbol PR_LEVEL[2-0]	Description	Reset																																				
15	-			0																																				
14-12	R/W	TIM0_INT_PRIO		0																																				
11	-			0																																				
10-8	R/W	SPI1_ADC_INT_PRIO		0																																				
7	-			0																																				
6-4	R/W	UART_TI_INT_PRIO		0																																				
3	-			0																																				
2-0	R/W	UART_RI_INT_PRIO	<p>The bits in INTx_PRIORITY_REG define the priority level of a maskable interrupt. The maskable interrupt with the interrupt pending bit set (1) and the highest priority, is serviced. After the interrupt is serviced the pending bit is NOT reset (0) by hardware. If the priority levels are equal, the inherent priority determines the priority (see Table 66)</p> <table> <thead> <tr> <th>PR_LEVEL[2]</th><th>PR_LEVEL[1]</th><th>PR_LEVEL[0]: Priority</th><th></th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>0</td><td>0 Disabled</td></tr> <tr> <td>0</td><td>0</td><td>1</td><td>1 Lowest</td></tr> <tr> <td>0</td><td>1</td><td>0</td><td>2</td></tr> <tr> <td>0</td><td>1</td><td>1</td><td>3</td></tr> <tr> <td>1</td><td>0</td><td>0</td><td>4</td></tr> <tr> <td>1</td><td>0</td><td>1</td><td>5</td></tr> <tr> <td>1</td><td>1</td><td>0</td><td>6</td></tr> <tr> <td>1</td><td>1</td><td>1</td><td>7 Highest</td></tr> </tbody> </table>	PR_LEVEL[2]	PR_LEVEL[1]	PR_LEVEL[0]: Priority		0	0	0	0 Disabled	0	0	1	1 Lowest	0	1	0	2	0	1	1	3	1	0	0	4	1	0	1	5	1	1	0	6	1	1	1	7 Highest	0
PR_LEVEL[2]	PR_LEVEL[1]	PR_LEVEL[0]: Priority																																						
0	0	0	0 Disabled																																					
0	0	1	1 Lowest																																					
0	1	0	2																																					
0	1	1	3																																					
1	0	0	4																																					
1	0	1	5																																					
1	1	0	6																																					
1	1	1	7 Highest																																					

Table 232: INT3_PRIORITY_REG (0xFF540A)

Bit	Mode	Symbol PR_LEVEL[2-0]	Description	Reset																																				
15	-			0																																				
14-12	R/W	CT_CLASSD_INT_PRIO	<p>The bits in INTx_PRIORITY_REG define the priority level of a maskable interrupt. The maskable interrupt with the interrupt pending bit set (1) and the highest priority, is serviced. After the interrupt is serviced the pending bit is NOT reset (0) by hardware. If the priority levels are equal, the inherent priority determines the priority (see Table 66)</p> <table> <thead> <tr> <th>PR_LEVEL[2]</th><th>PR_LEVEL[1]</th><th>PR_LEVEL[0]: Priority</th><th></th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>0</td><td>0 Disabled</td></tr> <tr> <td>0</td><td>0</td><td>1</td><td>1 Lowest</td></tr> <tr> <td>0</td><td>1</td><td>0</td><td>2</td></tr> <tr> <td>0</td><td>1</td><td>1</td><td>3</td></tr> <tr> <td>1</td><td>0</td><td>0</td><td>4</td></tr> <tr> <td>1</td><td>0</td><td>1</td><td>5</td></tr> <tr> <td>1</td><td>1</td><td>0</td><td>6</td></tr> <tr> <td>1</td><td>1</td><td>1</td><td>7 Highest</td></tr> </tbody> </table>	PR_LEVEL[2]	PR_LEVEL[1]	PR_LEVEL[0]: Priority		0	0	0	0 Disabled	0	0	1	1 Lowest	0	1	0	2	0	1	1	3	1	0	0	4	1	0	1	5	1	1	0	6	1	1	1	7 Highest	0
PR_LEVEL[2]	PR_LEVEL[1]	PR_LEVEL[0]: Priority																																						
0	0	0	0 Disabled																																					
0	0	1	1 Lowest																																					
0	1	0	2																																					
0	1	1	3																																					
1	0	0	4																																					
1	0	1	5																																					
1	1	0	6																																					
1	1	1	7 Highest																																					
11	-			0																																				
10-8	R/W	EMAC_INT_PRIO		0																																				
7	-			0																																				
6-4	R/W	KEYB_INT_PRIO		0																																				
3	-			0																																				
2-0	R/W	ACCESS_INT_PRIO		0																																				

Note 62: Reset value is device dependant according to CHIP_MEM_SIZE_REG (0xFFFFFBFB) (table 103, page 195).

Table 233: KEY_GP_INT_REG (0xFF49B0)

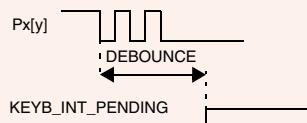
Bit	Mode	Symbol	Description	Reset
15-10	-	-	Reserved	0
9	R/W	DREQ1_CTRL	DMA request DREQ1 source selection: 0 = ADC_SYNC input. 1 = SPI1_INT source selection.	0
8-6	R/W	INT8_CTRL	0xx = Disable INT8 interrupt to KEYB_INT 100 = Pin INT8 generates a positive level interrupt on KEYB_INT 101 = Pin INT8 generates a negative level interrupt on KEYB_INT 110 = Pin INT8 generates a positive edge interrupt on KEYB_INT 111 = Pin INT8 generates a negative edge interrupt on KEYB_INT	0
5-3	R/W	INT7_CTRL	0xx = Disable INT7 interrupt to KEYB_INT 100 = Pin INT7 generates a positive level interrupt on KEYB_INT 101 = Pin INT7 generates a negative level interrupt on KEYB_INT 110 = Pin INT7 generates a positive edge interrupt on KEYB_INT 111 = Pin INT7 generates a negative edge interrupt on KEYB_INT	0
2-0	R/W	INT6_CTRL	0xx = Disable INT6 interrupt to KEYB_INT 100 = Pin INT6 generates a positive level interrupt on KEYB_INT 101 = Pin INT6 generates a negative level interrupt on KEYB_INT 110 = Pin INT6 generates a positive edge interrupt on KEYB_INT 111 = Pin INT6 generates a negative edge interrupt on KEYB_INT	0

Table 234: KEY_BOARD_INT_REG (0xFF49B2)

Bit	Mode	Symbol	Description	Reset
15-12	-	-	Reserved	
11	R/W	KEY_REL	0 = No interrupt on key release 1 = Interrupt on key release after DEBOUNCE time.	0
10	R/W	KEY_LEVEL	0 = INT0 to INT5 generate KEYB_INT if one or more pins are low 1 = INT0 to INT5 generate KEYB_INT if one or more pins are high	0
9-6	-	-	Reserved	0
5	R/W	INT5_EN	0 = Disable INT5 interrupt. 1 = Enable INT5 interrupt to KEYB_INT via debounce timer	0
4	R/W	INT4_EN	0 = Disable INT4 interrupt 1 = Enable INT4 interrupt to KEYB_INT via debounce timer	0
3	R/W	INT3_EN	0 = Disable INT3 interrupt 1 = Enable INT3 interrupt to KEYB_INT via debounce timer	0
2	R/W	INT2_EN	0 = Disable INT2 interrupt 1 = Enable INT2 interrupt to KEYB_INT via debounce timer	0
1	R/W	INT1_EN	0 = Disable INT1 interrupt 1 = Enable INT1 interrupt to KEYB_INT via debounce timer	0
0	R/W	INT0_EN	0 = Disable INT0 interrupt 1 = Enable INT0 interrupt to KEYB_INT via debounce timer	0

Table 235: KEY_DEBOUNCE_REG (0xFF49B4)

Bit	Mode	Symbol	Description	Reset
15-12	-	-	Reserved	0
11-6	R/W	KEY_REPEAT	While key is pressed, automatically generate repeating KEYB_INT after specified time unequal to 0. Repeat time: N*1 ms. N =1..63, N=0 disables timer. (Used clock is PER_CLK output /(8*144)) Refer to Figure 59 on page 101 for the keyboard interface statemachine.	0
5-0	R/W	DEBOUNCE	Keyboard debounce time. Generate KEYB_INT after specified time. Debounce time: N*1 ms. N =1..63 N=0 disables timer, gives immediate interrupt. (Used clock is PER_CLK output /(8*144)) Refer to Figure 59 for the keyboard interface statemachine.	0


Figure 101 Keyboard Debounce Time
Table 236: KEY_STATUS_REG (0xFF49B6)

Bit	Mode	Symbol	Description	Reset
15-6	-	-	Reserved	0
3-0	R/W	KEY_STATUS	Keyboard interrupt vector status. Register bits are cleared if a '1' is written to the corresponding bits. Writing a '1' to a bit that is '1', clears the bit. Writing a '1' to a bit that is '0' is not allowed. Writing a '0' is discarded. Bit 3= INT8 Bit 2= INT7 Bit 1= INT6 Bit 0 = Debounce or key repeat timer.	0

Table 237: PC_START_REG (0xFF540C)

Bit	Mode	Symbol	Description	Reset
15-2	R/W	PC_START	Bits 23-10 of the CR16C+ start address in blocks of Nx1kbyte. Bits 24 and 9-0 of the CR16C+ start address are always 0. If PC_START[15-2] = FFFF then PC = 0xFFFFC00, which is the maximum start address. Boot ROM start address: PC_START = FEF0, which corresponds to PC = 0xFEEF000	FEF0
1-0	R	PC_START10	Always 0	0

Table 238: P0_DATA_REG (0xFF4830)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	P0_DATA	If output, set P0[y], else returns the value of P0[y]	0

Table 239: P0_SET_DATA_REG (0xFF4832)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	P0_SET	If P0[y] output, writing a 1 sets P0[y] to 1. Writing 0 is discarded, Reading returns 0	0

Table 240: P0_RESET_DATA_REG (0xFF4834)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	P0_RESET	If P0[y] output, writing a 1 sets P0[y] to 0. Writing 0 is discarded, Reading returns 0	0

Table 241: P0_00_MODE_REG, P0_01_MODE_REG, P0_02_MODE_REG, P0_03_MODE_REG, P0_04_MODE_REG, P0_05_MODE_REG, P0_06_MODE_REG, P0_07_MODE_REG, P0_08_MODE_REG, P0_09_MODE_REG, P0_10_MODE_REG, P0_11_MODE_REG, P0_12_MODE_REG, P0_13_MODE_REG, P0_14_MODE_REG, P0_15_MODE_REG (0xFF4840, 0xFF4842, 0xFF4844, 0xFF4846, 0xFF4848, 0xFF484A, 0xFF484C, 0xFF484E, 0xFF4850, 0xFF4852, 0xFF4854, 0xFF4856, 0xFF4858, 0xFF485A, 0xFF485C, 0xFF485E)

Bit	Mode	Symbol	Description	Reset
15-10	-	-	Reserved	0,1,2,3 4,5,6,7 8,9,10,11 12,13,14,15
9-8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, Pull-down selected 11 = Output, no resistors selected In analog mode, these bits are don't care	1,1,2,2, 2,2,2,2, 2,2,2,1, 1,1,1,1
7-6	-	-	Reserved	0
5-0	R/W	PID	Peripheral ID (see Table 3) 0 = Port function, PUPD as set above X = Peripheral function according PPA table.	0,0,0,0, 0,0,0,0, 0,0,0,0, 0,0,0,0

Table 242: P1_DATA_REG (0xFF4860)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	P1_DATA	If output, set P1[y], else returns the value of P1[y]	

Table 243: P1_SET_DATA_REG (0xFF4862)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	P1_SET	If P1[y] output, writing a 1 sets P1[y] to 1. Writing 0 is discarded, Reading returns 0	0

Table 244: P1_RESET_DATA_REG (0xFF4864)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	P1_RESET	If P1[y] output, writing a 1 sets P1[y] to 0. Writing 0 is discarded, Reading returns 0	0

Table 245: P1_00_MODE_REG, P1_01_MODE_REG, P1_02_MODE_REG, P1_03_MODE_REG, P1_04_MODE_REG, P1_05_MODE_REG, P1_06_MODE_REG, P1_07_MODE_REG, P1_08_MODE_REG, P1_09_MODE_REG, P1_10_MODE_REG, P1_11_MODE_REG, P1_12_MODE_REG, P1_13_MODE_REG, P1_14_MODE_REG, P1_15_MODE_REG(0xFF4870, 0xFF4872, 0xFF4874, 0xFF4876, 0xFF4878, 0xFF487A, 0xFF487C, 0xFF487E, 0xFF4880, 0xFF4882, 0xFF4884, 0xFF4886, 0xFF4888, 0xFF488A, 0xFF488C, 0xFF488E)

Bit	Mode	Symbol	Description	Reset
15-10	-	-	Reserved	0,1,2,3 4,5,6,7 8,9,10,11 12,13,14,15
9-8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, Pull-down selected 11 = Output, no resistors selected In analog mode, these bits are don't care	2,2,2,2, 2,2,2,2, 2,2,2,2, 2,2,2,2
7-6	-	-	Reserved	0
5-0	R/W	PID	Peripheral ID (see Table 3) 0 = Port function, PUPD as set above X = Peripheral function according PPA table.	0,0,0,0, 0,0,0,0, 0,0,0,0, 0,0,0,0

Table 246: P2_DATA_REG (0xFF4890)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	P2_DATA	If output, set P2[y], else returns the value of P2[y] (Note 63)	0

Table 247: P2_SET_DATA_REG (0xFF4892)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	P2_SET	If P2[y] output, writing a 1 sets P2[y] to 1. Writing 0 is discarded, Reading returns 0 (Note 63)	0

Table 248: P2_RESET_DATA_REG,(0xFF4894)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	P2_RESET	If P2[y] output, writing a 1 sets P2[y] to 0. Writing 0 is discarded, Reading returns 0 (Note 63)	0

Table 249: P2_00_MODE_REG, P2_01_MODE_REG, P2_02_MODE_REG, P2_03_MODE_REG, P2_04_MODE_REG, P2_05_MODE_REG, P2_06_MODE_REG, P2_07_MODE_REG, P2_08_MODE_REG, P2_09_MODE_REG, P2_10_MODE_REG, P2_11_MODE_REG, P2_12_MODE_REG, P2_13_MODE_REG, P2_14_MODE_REG, P2_15_MODE_REG(0xFF48A0, 0xFF48A2, 0xFF48A4, 0xFF48A6, 0xFF48A8, 0xFF48AA, 0xFF48AC, 0xFF48AE , 0xFF48B0, 0xFF48B2, 0xFF48B4, 0xFF48B6, 0xFF48B8, 0xFF48BA, 0xFF48BC, 0xFF48BE)

Bit	Mode	Symbol	Description	Reset
15-10	-	-	Reserved	0,1,2,3, 4,5,6,7, 8,9,10,11 12,13,14, 15
9-8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, Pull-down selected 11 = Output, no resistors selected In analog mode, these bits are don't care (Note 63)	-, -, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2
7-6	-	-	Reserved	0
5-0	R/W	PID	Peripheral ID (see Table 3) 0 = Port function, PUPD as set above X = Peripheral function according PPA table.	63, 63, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

Note 63: PAOUTp/P2[0], PAOUTn/P2[1] are output only and have only internal pull-down resistor which can not be switched off.

Note 64: RSSI and ADC2 are combined. When using ADC2, setting DIP PD0=0 will set a logic 0 on this pin like when using the RSSI pin.

To disable the RSSI discharge function <B_RC> RC2[7] must be set to '1'

Note 65: If the analog function of P2[6:4] is selected (ADC0, ADC1, RSSI/ADC2) the digital input value is set to 0.

Table 250:P3_DATA_REG (0xFF48C0)

Bit	Mode	Symbol	Description	Reset
15-9	-	-	Reserved	0
8-0	R/W	P3_DATA	If output, set P3[y], else returns the value of P3[y]	0

Table 251: P3_SET_DATA_REG (0xFF48C2)

Bit	Mode	Symbol	Description	Reset
15-9	-	-	Reserved	0
8-0	R/W	P3_SET	If P3[y] output, writing a 1 sets P3[y] to 1. Writing 0 is discarded, Reading returns 0	0

Table 252: P3_RESET_DATA_REG (0xFF48C4)

Bit	Mode	Symbol	Description	Reset
15-9	-	-	Reserved	0
8-0	R/W	P3_RESET	If P3[y] output, writing a 1 sets P3[y] to 0. Writing 0 is discarded, Reading returns 0	0

**Table 253: P3_00_MODE_REG, P3_01_MODE_REG, P3_02_MODE_REG, P3_03_MODE_REG,
P3_04_MODE_REG, P3_05_MODE_REG, P3_06_MODE_REG, P3_07_MODE_REG,
P3_08_MODE_REG (0xFF48D0, 0xFF48D2, 0xFF48D4, 0xFF48D6, 0xFF48D8, 0xFF48DA,
0xFF48DC, 0xFF48DE , 0xFF48E0)**

Bit	Mode	Symbol	Description	Reset
15-10	-	-	Reserved	0,1,2,3, 4,5,6,7, 8
9-8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, Pull-down selected 11 = Output, no resistors selected In analog mode, these bits are do2n't care	2,2,2,2, 2,2,2,2, 2
7-6	-	-	Reserved	0
5-0	R/W	PID	Peripheral ID (see Table 3) 0 = Port function, PUPD as set above X = Peripheral function according PPA table.	0,0,0,0, 0,0,0,0, 0,

Table 254: QSPIC_BURSTCMDA_REG (0xFF0E0C)

Bit	Mode	Symbol	Description	Reset
31-30	R/W	QSPIC_DMY_TX_MD	It describes the mode of the SPI bus during the Dummy bytes phase. 00 – Single SPI 01 – Dual 10 – Quad 11 – Reserved	0
29-28	R/W	QSPIC_EXT_TX_MD	It describes the mode of the SPI bus during the Extra –Byte phase. 00 – Single SPI 01 – Dual 10 – Quad 11 – Reserved	0
27-26	R/W	QSPIC_ADR_TX_MD	It describes the mode of the SPI bus during the address phase. 00 – Signle SPI 01 – Dual 10 – Quad 11 – Reserved	0
25-24	R/W	QSPIC_INST_TX_MD	It describes the mode of the SPI bus during the instruction phase. 00 – Single SPI 01 – Dual 10 – Quad 11 – Reserved	0
23-16	R/W	QSPIC_EXT_BYTE	The value of an extra byte which will be transferred after address (only if QSPIC_EXT_BYTE_EN= 1). Usually this is the Mode Bits in Dual/Quad SPI I/O instructions. 0	0
15-8	R/W	QSPIC_INST_WB	Instruction Value for Wrapping Burst. This value is the selected instruction when QSPIC_WRAP_MD is equal to 1 and the access is a wrapping burst of length and size described by the bitfields QSPIC_WRAP_LEN and QSPIC_WRAP_SIZE respectively.	0
7-0	R/W	QSPIC_INST	Instruction Value for Incremental Burst or Single read access. This value is the selected instruction at the cases of incremental burst or single read access. Also this value is used when a wrapping burst is not supported (QSPIC_WRAP_MD)	0

Table 255: QSPIC_BURSTCMDB_REG (0xFF0E10)

Bit	Mode	Symbol	Description	Reset
31-15	-	-	Reserved	0
14-12	R/W	QSPIC_CS_HIGH_MIN	CS minimum high CS stays high for this minimum number of SPI CLKs.	0
11-10	R/W	QSPIC_WRAP_SIZE	It describes the selected data size of a wrapping burst (QSPIC_WRAP_MD). 00 – byte access (8-bits) 01 – half word access (16 –bits) 10 – word access (32-bits) 11 – Reserved	0
9-8	R/W	QSPIC_WRAP_LEN	It describes the selected length of a wrapping burst (QSPIC_WRAP_MD). 00 – 4 beat wrapping burst 01 – 8 beat wrapping burst 10 – 16 beat wrapping burst 11 – Reserved	0
7	R/W	QSPIC_WRAP_MD	Wrap mode 0 – The QSPIC_INST is the selected instruction at any access. 1 – The QSPIC_INST_WB is the selected instruction at any wrapping burst access of length and size described by the registers QSPIC_WRAP_LEN and QSPIC_WRAP_SIZE respectively. In all other cases the QSPIC_INST is the selected instruction. Use this feature only when the serial FLASH memory supports a special instruction for wrapping burst access.	0
6	R/W	QSPIC_INST_MD	Instruction mode 0 – Transmit instruction at any burst access. 1 – Transmit instruction only in the first access after the selection of Auto Mode.	0
5-4	R/W	QSPIC_DMY_NUM	Number of Dummy Bytes 00 - Zero Dummy Bytes (Don't Send Dummy Bytes) 01 - Send 1 Dummy Byte 10 - Send 2 Dummy Bytes 11 - Send 4 Dummy Bytes	0
3	R/w	QSPIC_EXT_HF_DS	Extra Half Disable Output 0 - if QSPIC_EXT_BYTE_EN=1 then transmit the complete QSPIC_EXT_BYTE 1 - if QSPIC_EXT_BYTE_EN=1 then disable (hi-z) output during the transmission of bits [3:0] of QSPIC_EXT_BYTE	0
2	R/W	QSPIC_EXT_BYTE_EN	Extra Byte Enable 0 - Don't Send QSPIC_EXT_BYTE 1 - Send QSPIC_EXT_BYTE	0
1-0	R/W	QSPIC_DAT_RX_MD	It describes the mode of the SPI bus during the data phase. 00 – Single SPI 01 – Dual 10 – Quad 11 – Reserved	0

Table 256: QSPIC_CTRLBUS_REG (0xFF0E00)

Bit	Mode	Symbol	Description	Reset
31-5	-	-	Reserved	0
4	W	QSPIC_DIS_CS	Write 1 to disable the chip select (active low).	0
3	W	QSPIC_EN_CS	Write 1 to enable the chip select (active low).	0
2	W	QSPIC_SET_QUAD	Write 1 to set the bus mode in Quad mode.	0
1	W	QSPIC_SET_DUAL	Write 1 to set the bus mode in Dual mode.	0
0	W	QSPIC_SET_SINGLE	Write 1 to set the bus mode in Single SPI mode.	0

Table 257: QSPIC_CTRLMODE_REG (0xFF0E04)

Bit	Mode	Symbol	Description	Reset
31-8	-	-	Reserved	0
7	R/W	QSPIC_RXD_NEG	0 – Sample the received data with the positive edge of the QSPIC_CLK. 1 – Sample the received data with the negative edge of the QSPIC_CLK	0
6	R/W	QSPIC_HRDY_MD	0 – Add wait states via hready signal when accessing the QSPIC_WRITEDATA, QSPIC_READDATA and QSPIC_DUMMYDATA registers. 1 – Don't use the hready signal when accessing the QSPIC_WRITEDATA, QSPIC_READDATA and QSPIC_DUMMYDATA registers. Read the QSPIC_BUSY bit from the QSPIC_STATUS register to check on the end of the activity at the SPI bus.	0
5	R/W	QSPIC_IO3_DAT	The value of QSPIC_IO3 pad if QSPIC_IO3_OEN is 1	0
4	R/W	QSPIC_IO2_DAT	The value of QSPIC_IO2 pad if QSPIC_IO2_OEN is 1	0
3	R/W	QSPIC_IO3_OEN	QSPIC_IO3 output enable. Use this only in SPI or Dual SPI mode to control /WP signal. When the Auto Mode is selected (QSPIC_AUTO_MD = 1) and the QUAD SPI is used, set this bit to zero. 0 – The QSPIC_IO3 pad is input. 1 – The QSPIC_IO3 pad is output.	0
2	R/W	QSPIC_IO2_OEN	QSPIC_IO2 output enable. Use this only in SPI or Dual SPI mode to control /WP signal. When the Auto Mode is selected (QSPIC_AUTO_MD = 1) and the QUAD SPI is used, set this bit to zero. 0 – The QSPIC_IO2 pad is input. 1 – The QSPIC_IO2 pad is output.	0
1	R/W	QSPIC_CLK_MD	Mode of the generated QSPIC_CLK clock 0 – Use Mode 0 for the QSPIC_CLK. The QSPIC_CLK is low when QSPIC_CS is high. 1 – Use Mode 3 for the QSPIC_CLK. The QSPIC_CLK is high when QSPIC_CS is high.	0
0	R/W	QSPIC_AUTO_MD	Mode of operation 0 – The Manual Mode is selected. 1 – The Auto Mode is selected.	0

Table 258: QSPIC_DUMMYDATA_REG (0xFF0E20)

Bit	Mode	Symbol	Description	Reset
31-0	W	QSPIC_DUMMYDATA	Writing to this register is generating a number of clock pulses to the SPI bus. During this activity in the SPI bus, the QSPIC_IO data pads are in hi-z state. The data size of the access to this register can be 32-bits / 16-bits/ 8-bits. The number of generated pulses is equal to: (size of AHB bus access) / (size of SPI bus). The size of SPI bus is equal to 1, 2 or 4 for Single, Dual or Quad SPI mode respectively.	0

Table 259: QSPIC_READDATA_REG (0xFF0E1C)

Bit	Mode	Symbol	Description	Reset
31-0	R	QSPIC_READDATA	A read access at this register is generating a data transfer from the serial FLASH memory to the QSFC controller. The data is transferred using the selected mode of the SPI bus (SPI, Dual SPI, Quad SPI). The data size of the access to this register can be 32-bits / 16-bits / 8-bits and is equal to the number of the transferred bits.	0

Table 260: QSPIC_RECVDATA_REG (0xFF0E08)

Bit	Mode	Symbol	Description	Reset
31-0	R	QSPIC_RECVDATA	When QSPIC_READDATA register is used and QSPIC_HRDY_MD=1 and QSPIC_BUSY=0, then this register contains the received data.	0

Table 261: QSPIC_STATUS_REG (0xFF0E14)

Bit	Mode	Symbol	Description	Reset
31-1	-	-	Reserved	0
0	R	QSPIC_BUSY	The status of the SPI Bus. 0 – The SPI Bus is idle 1 – The SPI Bus is active. ReadData, WriteData or DummyData activity is in progress.	0

Table 262: QSPIC_WRITEDATA_REG (0xFF0E18)

Bit	Mode	Symbol	Description	Reset
31-0	W	QSPIC_WRITEDATA	Writing to this register is generating a data transfer from the controller to the serial FLASH memory. The data written at this register, are transferred to the memory using the selected mode of the SPI bus (SPI, Dual SPI, Quad SPI). The data size of the access to this register can be 32-bits / 16-bits / 8-bits and is equal to the number of the transferred bits.	0

Table 263: EMAC_MACR0_CONFIG_REG (0xFF2000)

Bit	Mode	Symbol	Description	Reset
31-25	-	-	Reserved	0
24	R/W	TC	Transmit Configuration in RMII/MII When set, this bit enables the transmission of duplex mode, link speed, and link up/down information to the PHY in the RMII ports. The details of this feature are explained in "Reduced Media Independent Interface". When this bit is reset, no such information is driven to the PHY.	0
23	R/W	WD	Watchdog Disable When this bit is set, the MAC disables the watchdog timer on the receiver, and can receive frames of up to 16,384 bytes. When this bit is reset, the MAC allows no more than 2,048 bytes (10,240 if JE is set high) of the frame being received and cuts off any bytes received after that.	0
22	R/W	JD	Jabber Disable When this bit is set, the MAC disables the jabber timer on the transmitter, and can transfer frames of up to 16,384 bytes. When this bit is reset, the MAC cuts off the transmitter if the application sends out more than 2,048 bytes of data (10,240 if JE is set high) during transmission.	0
21	-	-	Reserved	0
20	R/W	JE	Jumbo Frame Enable When this bit is set, MAC allows Jumbo frames of 9,018 bytes (9,022 bytes for VLAN tagged frames) without reporting a giant frame error in the receive frame status.	0
19-17	R/W	IFG	Inter-Frame Gap These bits control the minimum IFG between frames during transmission. 000 = 96 bit times 001 = 88 bit times 010 = 80 bit times ... 111 = 40 bit times Note that in Half-Duplex mode, the minimum IFG can be configured for 64 bit times (IFG = 100) only. Lower values are not considered.	000
16	R/W	DCRS	Disable Carrier Sense During Transmission When set high, this bit makes the MAC transmitter ignore the MII CRS signal during frame transmission in Half-Duplex mode. This request results in no errors generated due to Loss of Carrier or No Carrier during such transmission. When this bit is low, the MAC transmitter generates such errors due to Carrier Sense and will even abort the transmissions.	0
15	-	-	Reserved	1
14	R/W	FES	Speed Indicates the speed in Fast Ethernet (MII) mode: 0 = 10 Mbps 1 = 100 Mbps This bit generates link speed encoding when TC (Bit 24) is set in RMII mode.	0

Table 263: EMAC_MACR0_CONFIG_REG (0xFF2000)

Bit	Mode	Symbol	Description	Reset
13	R/W	DO	Disable Receive Own When this bit is set, the MAC disables the reception of frames when the EMAC_TXEN is asserted in Half-Duplex mode. When this bit is reset, the MAC receives all packets that are given by the PHY while transmitting. This bit is not applicable if the MAC is operating in Full-Duplex mode.	0
12	R/W	LM	Loopback Mode When this bit is set, the MAC operates in loopback mode at MII. The MII Receive clock input (provided by the PHY) is required for the loopback to work properly, as the Transmit clock is not looped-back internally.	0
11	R/W	DM	Duplex Mode When this bit is set, the MAC operates in a Full-Duplex mode where it can transmit and receive simultaneously. This bit is RO with default value of 1'b1 in Full-Duplex-only configuration.	0
10	R/W	IPC	Checksum Offload When this bit is set, the MAC calculates the 16-bit one's complement of the one's complement sum of all received Ethernet frame payloads. It also checks whether the IPv4 Header checksum (assumed to be bytes 25–26 or 29–30 (VLAN-tagged) of the received Ethernet frame) is correct for the received frame and gives the status in the receive status word. The MAC core also appends the 16-bit checksum calculated for the IP header datagram payload (bytes after the IPv4 header) and appends it to the Ethernet frame transferred to the application (when Type 2 COE is deselected). When this bit is reset, this function is disabled. When Type 2 COE is selected, this bit, when set, enables IPv4 checksum checking for received frame payloads' TCP/UDP/ICMP headers. When this bit is reset, the COE function in the receiver is disabled and the corresponding PCE and IP HCE status bits are always cleared. If the IP Checksum Offload feature is not enabled during coreKit configuration, this bit is reserved (RO with default value).	0
9	R/W	DR	Disable Retry When this bit is set, the MAC will attempt only 1 transmission. When a collision occurs on the MII, the MAC will ignore the current frame transmission and report a Frame Abort with excessive collision error in the transmit frame status. When this bit is reset, the MAC will attempt retries based on the settings of BL.	0
8	R/W	LUD	Link Up/Down Indicates whether the link is up or down during the transmission of configuration in RMII interface: 0 = Link Down 1 = Link Up	0
7	R/W	ACS	Automatic Pad/CRC Stripping When this bit is set, the MAC strips the Pad/FCS field on incoming frames only if the length's field value is less than or equal to 1,500 bytes. All received frames with length field greater than or equal to 1,501 bytes are passed to the application without stripping the Pad/FCS field. When this bit is reset, the MAC will pass all incoming frames to the Host unmodified.	0

Table 263: EMAC_MACR0_CONFIG_REG (0xFF2000)

Bit	Mode	Symbol	Description	Reset
6-5	R/W	BL	<p>Back-Off Limit The Back-Off limit determines the random integer number (r) of slot time delays (4,096 bit times for 1000 Mbps and 512 bit times for 10/100 Mbps) the MAC waits before rescheduling a transmission attempt during retries after a collision. This bit is applicable only to Half-Duplex mode.</p> <p>00 = k := min (n, 10) 01 = k := min (n, 8) 10 = k := min (n, 4) 11 = k := min (n, 1), where n = retransmission attempt. The random integer r takes the value in the range $0 \leq r < 2^k$</p>	00
4	R/W	DC	<p>Deferral Check When this bit is set, the deferral check function is enabled in the MAC. The MAC will issue a Frame Abort status, along with the excessive deferral error bit set in the transmit frame status when the transmit state machine is deferred for more than 24,288 bit times in 10/100-Mbps mode. If the Core is configured for 1000 Mbps operation, or if the Jumbo frame mode is enabled in 10/100-Mbps mode, the threshold for deferral is 155,680 bits times. Deferral begins when the transmitter is ready to transmit, but is prevented because of an active CRS (carrier sense) signal on the MII. Defer time is not cumulative. If the transmitter defers for 10,000 bit times, then transmits, collides, backs off, and then has to defer again after completion of back-off, the deferral timer resets to 0 and restarts.</p> <p>When this bit is reset, the deferral check function is disabled and the MAC defers until the CRS signal goes inactive. This bit is applicable only in Half-Duplex mode.</p>	0
3	R/W	TE	<p>Transmitter Enable When this bit is set, the transmit state machine of the MAC is enabled for transmission on the MII. When this bit is reset, the MAC transmit state machine is disabled after the completion of the transmission of the current frame, and will not transmit any further frames.</p>	0
2	R/W	RE	<p>Receiver Enable When this bit is set, the receiver state machine of the MAC is enabled for receiving frames from the MII. When this bit is reset, the MAC receive state machine is disabled after the completion of the reception of the current frame, and will not receive any further frames from the MII.</p>	0
1-0	-	-	Reserved	00

Table 264: EMAC_MACR1_FRAME_FILTER_REG (0xFF2004)

Bit	Mode	Symbol	Description	Reset
31	R/W	RA	<p>Receive All When this bit is set, the MAC Receiver module passes to the Application all frames received irrespective of whether they pass the address filter. The result of the SA/DA filtering is updated (pass or fail) in the corresponding bits in the Receive Status Word. When this bit is reset, the Receiver module passes to the Application only those frames that pass the SA/DA address filter.</p>	0
30-20	-	-	Reserved	0
19-10	-	-	Reserved	0
9	R/W	SAF	<p>Source Address Filter Enable The MAC core compares the SA field of the received frames with the values programmed in the enabled SA registers. If the comparison matches, then the SAMatch bit of RxStatus Word is set high. When this bit is set high and the SA filter fails, the MAC drops the frame. When this bit is reset, then the MAC Core forwards the received frame to the application and with the updated SA Match bit of the RxStatus depending on the SA address comparison.</p>	0
8	R/W	SAIF	<p>SA Inverse Filtering When this bit is set, the Address Check block operates in inverse filtering mode for the SA address comparison. The frames whose SA matches the SA registers will be marked as failing the SA Address filter. When this bit is reset, frames whose SA does not match the SA registers will be marked as failing the SA Address filter.</p>	0
7-6	R/W	PCF	<p>Pass Control Frames These bits control the forwarding of all control frames (including unicast and multicast PAUSE frames). Note that the processing of PAUSE control frames depends only on RFE of Flow Control Register[2]. 0x = MAC filters all control frames from reaching the application 10 = MAC forwards all control frames to application even if they fail the Address Filter 11 = MAC forwards control frames that pass the Address Filter.</p>	0
5	R/W	DBF	<p>Disable Broadcast Frames When this bit is set, the AFM module filters all incoming broadcast frames. When this bit is reset, the AFM module passes all received broadcast frames.</p>	0
4	R/W	PM	<p>Pass All Multicast When set, this bit indicates that all received frames with a multicast destination address (first bit in the destination address field is '1') are passed. When reset, filtering of multicast frame depends on HMC bit.</p>	0
3	R/W	DAIF	<p>DA Inverse Filtering When this bit is set, the Address Check block operates in inverse filtering mode for the DA address comparison for both unicast and multicast frames. When reset, normal filtering of frames is performed</p>	0

Table 264: EMAC_MACR1_FRAME_FILTER_REG (0xFF2004)

Bit	Mode	Symbol	Description	Reset
2	-	-	Reserved	0
1	-	-	Reserved	0
0	R/W	PRM	Promiscuous Mode When this bit is set, the Address Filter module passes all incoming frames regardless of its destination or source address. The SA/DA Filter Fails status bits of the Receive Status Word will always be cleared when PR is set.	0

Table 265: EMAC_MACR4_MII_ADDR_REG (0xFF2010)

Bit	Mode	Symbol	Description	Reset
31-16	-	-	Reserved	0
15-11	R/W	PA	Physical Layer Address This field tells which of the 32 possible PHY devices are being accessed	0
10-6	R/W	GR	MII Register These bits select the desired MII register in the selected PHY device	0
5	-	-	Reserved	0
4-2	R/W	CR	CSR Clock Range The CSR Clock Range selection determines the system clock frequency and is used to decide the frequency of the MDC clock: Selection {system clock} MDC Clock 000 60-100 MHz {sys.clock}/42 001 100-150 MHz {sys.clock}/62 010 20-35 MHz {sys.clock}/16 011 35-60 MHz {sys.clock}/26 100 150-250 MHz {sys.clock}/102 101 250-300 MHz {sys.clock}/122 110, 111 Reserved	0
1	R/W	GW	MII Write When set, this bit tells the PHY that this will be a Write operation using the MII Data register. If this bit is not set, this will be a Read operation, placing the data in the MII Data register.	0
0	R/W	GB	MII Busy This bit should read a logic 0 before writing to Register 4 and Register 5. This bit must also be set to 0 during a Write to Register 4. During a PHY register access, this bit will be set to 1'b1 by the Application to indicate that a Read or Write access is in progress. Register 5 (MII Data) should be kept valid until this bit is cleared by the MAC during a PHY Write operation. The Register 5 is invalid until this bit is cleared by the MAC during a PHY Read operation. The Register 4 (MII Address) should not be written to until this bit is cleared.	0

Table 266: EMAC_MACR5_MII_DATA_REG (0xFF2014)

Bit	Mode	Symbol	Description	Reset
31-16	-	-	Reserved	0
15-0	R/W	GD	MII Data This contains the 16-bit data value read from the PHY after a Management Read operation or the 16-bit data value to be written to the PHY before a Management Write operation.	0

Table 267: EMAC_MACR6_FLOW_CTRL_REG (0xFF2018)

Bit	Mode	Symbol	Description	Reset
31-16	R/W	PT	Pause Time This field holds the value to be used in the Pause Time field in the transmit control frame. If the Pause Time bits is configured to be double-synchronized to the (G)MII clock domain, then consecutive writes to this register should be performed only after at least 4 clock cycles in the destination clock domain	0
15-8	-	-	Reserved	0
7	R/W	DZPQ	Disable Zero-Quanta Pause When set, this bit disables the automatic generation of Zero-Quanta Pause Control frames on the deassertion of the flow-control signal from the FIFO layer (MTL or external sideband flow control signal sbd_flowctrl_i/mti_flowctrl_i). When this bit is reset, normal operation with automatic Zero-Quanta Pause Control frame generation is enabled	0
6	-	-	Reserved	0
5-4	R/W	PLT	Pause Low Threshold This field configures the threshold of the PAUSE timer at which the input flow control signal is checked for automatic retransmission of PAUSE Frame. The threshold values should be always less than the Pause Time configured in Bits[31:16]. For example, if PT = 100H (256 slot-times), and PLT = 01, then a second PAUSE frame is automatically transmitted if the flow control signal is asserted at 228 (256 – 28) slot-times after the first PAUSE frame is transmitted. Selection Threshold 00 = Pause time minus 4 slot times 01 = Pause time minus 28 slot times 10 = Pause time minus 144 slot times 11 = Pause time minus 256 slot times Slot time is defined as time taken to transmit 512 bits (64 bytes) on the MII interface.	0
3	R/W	UP	Unicast Pause Frame Detect When this bit is set, the MAC will detect the Pause frames with the station's unicast address specified in MAC Address0 High Register and MAC Address0 Low Register, in addition to the detecting Pause frames with the unique multicast address. When this bit is reset, the MAC will detect only a Pause frame with the unique multicast address specified in the 802.3x standard.	0

Table 267: EMAC_MACR6_FLOW_CTRL_REG (0xFF2018)

Bit	Mode	Symbol	Description	Reset
2	R/W	RFE	Receive Flow Control Enable When this bit is set, the MAC will decode the received Pause frame and disable its transmitter for a specified (Pause Time) time. When this bit is reset, the decode function of the Pause frame is disabled	0
1	R/W	TFE	Transmit Flow Control Enable In Full-Duplex mode, when this bit is set, the MAC enables the flow control operation to transmit Pause frames. When this bit is reset, the flow control operation in the MAC is disabled, and the MAC will not transmit any Pause frames. In Half-Duplex mode, when this bit is set, the MAC enables the back-pressure operation. When this bit is reset, the back-pressure feature is disabled.	0
0	R/W /SC (FCB) R/W (BPA)	FCB_BPA	Flow Control Busy/Backpressure Activate This bit initiates a Pause Control frame in Full-Duplex mode and activates the backpressure function in Half-Duplex mode if TFE bit is set. In Full-Duplex mode, this bit should be read as 1'b0 before writing to the Flow Control register. To initiate a Pause control frame, the Application must set this bit to 1'b1. During a transfer of the Control Frame, this bit will continue to be set to signify that a frame transmission is in progress. After the completion of Pause control frame transmission, the MAC will reset this bit to 1'b0. The Flow Control register should not be written to until this bit is cleared. In Half-Duplex mode, when this bit is set (and TFE is set), then backpressure is asserted by the MAC Core. During backpressure, when the MAC receives a new frame, the transmitter starts sending a JAM pattern resulting in a collision. When the MAC is configured to Full-Duplex mode, the BPA is automatically disabled.	0

Table 268: EMAC_MACR7_VLAN_TAG_REG (0xFF201C)

Bit	Mode	Symbol	Description	Reset
31-17	-	-	Reserved	0
16	R/W	ETV	Enable 12-Bit VLAN Tag Comparison When this bit is set, a 12-bit VLAN identifier, rather than the complete 16-bit VLAN tag, is used for comparison and filtering. Bits[11:0] of the VLAN tag are compared with the corresponding field in the received VLAN-tagged frame. When this bit is reset, all 16 bits of the received VLAN frame's fifteenth and sixteenth bytes are used for comparison.	0
15-0	R/W	VL	VLAN Tag Identifier for Receive Frames This contains the 802.1Q LAN tag to identify VLAN frames, and is compared to the fifteenth and sixteenth bytes of the frames being received for VLAN frames. Bits[15:13] are the User Priority, Bit[12] is the Canonical Format Indicator (CFI) and bits[11:0] are the VLAN tag's VLAN Identifier (VID) field. When the ETV bit is set, only the VID (Bits[11:0]) is used for comparison. If VL (VL[11:0] if ETV is set) is all zeros, the MAC does not check the fifteenth and sixteenth bytes for VLAN tag comparison, and declares all frames with a Type field value of 0x8100 to be VLAN frames.	0

Table 269: EMAC_MACR8_CORE_VER_REG (0xFF2020)

Bit	Mode	Symbol	Description	Reset
31-16	-	-	Reserved	0
15-8	R	Version_high	Ethernet MAC core version	0x10
7-0	R	Version_low	Ethernet MAC core version	0x34

Table 270: EMAC_MACR14_INT_REG (0xFF2038)

Bit	Mode	Symbol	Description	Reset
31-8	-	-	Reserved	0
7	R	MRCOI	MMC Receive Checksum Offload Interrupt Status This bit is set high whenever an interrupt is generated in the MMC Receive Checksum Offload Interrupt Register. This bit is cleared when all the bits in this interrupt register are cleared. This bit is only valid when the optional MMC module and Checksum Offload Engine (Type 2) are selected during configuration.	0
6	R	MTI	MMC Transmit Interrupt Status This bit is set high whenever an interrupt is generated in the MMC Transmit Interrupt Register. This bit is cleared when all the bits in this interrupt register are cleared.	0
5	R	MRI	MMC Receive Interrupt Status This bit is set high whenever an interrupt is generated in the MMC Receive Interrupt Register. This bit is cleared when all the bits in this interrupt register are cleared.	0
4	R	MI	MMC Interrupt Status This bit is set high whenever any of bits 7:5 is set high and cleared only when all of these bits are low.	0
3-1	-	-	Reserved	0
0	R	RMI	RMII Interrupt Status This bit is set due to any change in value of the Link Status of RMII interface (Bit 3 of RMII Status Register in "Register 54 (RMII Status Register)"). This bit is cleared when the user makes a read operation the RMII Status register.	0

Table 271: EMAC_MACR15_INT_MSK_REG (0xFF203C)

Bit	Mode	Symbol	Description	Reset
31-3	-	-	Reserved	0
2	R/W	PANCI	PCS AN Completion Interrupt Mask This bit when set, will disable the assertion of the interrupt signal due to the setting of PCS Auto-negotiation complete bit in Register 14 caused due to the completion of Autonegotiation event.	0
1	R/W	PLSI	PCS Link Status Interrupt Mask This bit when set, will disable the assertion of the interrupt signal due to the setting of PCS Link-status changed bit in Register 14 caused due to change in link-status event	0
0	R/W	RMI	RMII Interrupt Mask This bit when set, will disable the assertion of the interrupt signal due to the setting of RMII Interrupt Status bit in Register 14	0

Table 272: EMAC_MACR16_MAC_ADDR0_HIGH_REG (0xFF2040)

Bit	Mode	Symbol	Description	Reset
31	-	-	Reserved	1
30-16	-	-	Reserved	0
15-0	R/W		MAC Address0 [47:32] This field contains the upper 16 bits (47:32) of the 6-byte first MAC address. This is used by the MAC for filtering for received frames and for inserting the MAC address in the Transmit Flow Control (PAUSE) Frames.	0xFFFF

Table 273: EMAC_MACR17_MAC_ADDR0_LOW_REG (0xFF2044)

Bit	Mode	Symbol	Description	Reset
31-0	R/W		MAC Address0 [31:0] This field contains the lower 32 bits of the 6-byte first MAC address. This is used by the MAC for filtering for received frames and for inserting the MAC address in the Transmit Flow Control (PAUSE) Frames	0xFFFF_FFFF

Table 274: EMAC_MACR18_MAC_ADDR1_HIGH_REG (0xFF2048)

Bit	Mode	Symbol	Description	Reset
31	R/W	AE	Address Enable When this bit is set, the Address filter module uses the second MAC address for perfect filtering. When reset, the address filter module will ignore the address for filtering.	0
30	R/W	SA	Source Address When this bit is set, the MAC Address1[47:0] is used to compare with the SA fields of the received frame. When this bit is reset, the MAC Address1[47:0] is used to compare with the DA fields of the received frame.	0
29-24	R/W	MBC	Mask Byte Control These bits are mask control bits for comparison of each of the MAC Address bytes. When set high, the EMAC core does not compare the corresponding byte of received DA/SA with the contents of MAC Address1 registers. Each bit controls the masking of the bytes as follows: <ul style="list-style-type: none"> • Bit 29: Register 18[15:8] • Bit 28: Register 18[7:0] • Bit 27: Register 19[31:24] ... • Bit 24: Register 19[7:0] 	0
23-16	-	-	Reserded	0
15-0	R/W		MAC Address1 [47:32] This field contains the upper 16 bits (47:32) of the 6-byte second MAC address.	0xFFFF

Table 275: EMAC_MACR19_MAC_ADDR1_LOW_REG (0xFF204C)

Bit	Mode	Symbol	Description	Reset
31-0	R/W		MAC Address1 [31:0] This field contains the lower 32 bits of the 6-byte second MAC address. The content of this field is undefined until loaded by the Application after the initialization process. (Note 68)	0xFFFF_FFFF

Note 66: The descriptions of registers **EMAC_MACR20_MAC_ADDR2_HIGH_REG** (0xFF2050), **EMAC_MACR22_MAC_ADDR3_HIGH_REG** (0xFF2058) and **EMAC_MACR24_MAC_ADDR4_HIGH_REG** (0xFF2060) are the same as for the **EMAC_MACR18_MAC_ADDR1_HIGH_REG**.

Note 67: The descriptions of registers **EMAC_MACR21_MAC_ADDR2_LOW_REG** (0xFF2054), **EMAC_MACR23_MAC_ADDR3_LOW_REG** (0xFF205C) and **EMAC_MACR25_MAC_ADDR4_LOW_REG** (0xFF2064) are the same as for the **EMAC_MACR19_MAC_ADDR1_LOW_REG**.

Note 68: The S/W must load first the **MAC_ADDRx_HIGH** register values and last the **MAC_ADDRx_LOW** register values.

Table 276: EMAC_MACR54_MII_STATUS_REG (0xFF20D8)

Bit	Mode	Symbol	Description	Reset
31-4	-	-	Reserved	0
3	R		Link Status Indicates whether the link is up (1'b1) or down (1'b0).	0
2-1	R		Link Speed Indicates the current speed of the link: 00 = 2.5 MHz 01 = 25 MHz 10 = 125 MHz	0
0	R		Link Mode Indicates the current mode of operation of the link: 0 = Half-Duplex mode 1 = Full-Duplex mode	0

Table 277: EMAC_MMICNTL_REG (0xFF2100)

Bit	Mode	Symbol	Description	Reset
31-4	-	-	Reserved	0
3	R/W		MMC Counter Freeze When set, this bit freezes all the MMC counters to their current value. (None of the MMC counters are updated due to any transmitted or received frame until this bit is reset to 0. If any MMC counter is read with the Reset on Read bit set, then that counter is also cleared in this mode.)	0
2	R/W		Reset on Read When set, the MMC counters will be reset to zero after Read (self-clearing after reset). The counters are cleared when the least significant byte lane (bits[7:0]) is read.	0
1	R/W		Counter Stop Rollover When set, counter after reaching maximum value will not roll over to zero	0
0	R/W		Counters Reset When set, all counters will be reset. This bit will be cleared automatically after 1 lock cycle	0

Table 278: EMAC_MMCI_NTR_RX_REG (0xFF2104)

Bit	Mode	Symbol	Description	Reset
31-24	-	-	Reserved	0
23	R/SS/ RC		The bit is set when the rxwatchdog error counter reaches half the maximum value	0
22	R/SS/ RC		The bit is set when the rxvlanframes_gb counter reaches half the maximum value	0
21	R/SS/ RC		The bit is set when the rxfifooverflow counter reaches half the maximum value	0
20	R/SS/ RC		The bit is set when the rxpauseframes counter reaches half the maximum value	0
19	R/SS/ RC		The bit is set when the rxoutofrange type counter reaches half the maximum value	0
18	R/SS/ RC		The bit is set when the rxlengtherror counter reaches half the maximum value	0
17	R/SS/ RC		The bit is set when the rxunicastframes_gb counter reaches half the maximum value.	0
16	R/SS/ RC		The bit is set when the rx1024tomaxoctets_gb counter reaches half the maximum value.	0
15	R/SS/ RC		The bit is set when the rx512to1023octets_gb counter reaches half the maximum value.	0
14	R/SS/ RC		The bit is set when the rx256to511octets_gb counter reaches half the maximum value.	0
13	R/SS/ RC		The bit is set when the rx128to255octets_gb counter reaches half the maximum value.	0
12	R/SS/ RC		The bit is set when the rx65to127octets_gb counter reaches half the maximum value.	0
11	R/SS/ RC		The bit is set when the rx64octets_gb counter reaches half the maximum value	0
10	R/SS/ RC		The bit is set when the rxoversize_g counter reaches half the maximum value	0
9	R/SS/ RC		The bit is set when the rxundersize_g counter reaches half the maximum value	0
8	R/SS/ RC		The bit is set when the rxjabbererror counter reaches half the maximum value	0
7	R/SS/ RC		The bit is set when the rxrunterror counter reaches half the maximum value	0
6	R/SS/ RC		The bit is set when the rxalignmenterror counter reaches half the maximum value	0
5	R/SS/ RC		The bit is set when the rxrcerror counter reaches half the maximum value	0
4	R/SS/ RC		The bit is set when the rxmulticastframes_g counter reaches half the maximum value.	0
3	R/SS/ RC		The bit is set when the rxbroadcastframes_g counter reaches half the maximum value.	0
2	R/SS/ RC		The bit is set when the rxoctetcount_g counter reaches half the maximum value	0
1	R/SS/ RC		The bit is set when the rxoctetcount_gb counter reaches half the maximum value	0
0	R/SS/ RC		The bit is set when the rxframecount_gb counter reaches half the maximum value.	0

Table 279: EMAC_MMCI_NTR_TX_REG (0xFF2108)

Bit	Mode	Symbol	Description	Reset
31-25	-	-	Reserved	0
24	R/SS/ RC		The bit is set when the txvlanframes_g counter reaches half the maximum value	0
23	R/SS/ RC		The bit is set when the txpauseframes error counter reaches half the maximum value	0
22	R/SS/ RC		The bit is set when the txoexcessdef counter reaches half the maximum value	0
21	R/SS/ RC		The bit is set when the txframecount_g counter reaches half the maximum value.	0
20	R/SS/ RC		The bit is set when the txoctetcount_g counter reaches half the maximum value	0
19	R/SS/ RC		The bit is set when the txcarriererror counter reaches half the maximum value	0
18	R/SS/ RC		The bit is set when the txexcesscol counter reaches half the maximum value	0
17	R/SS/ RC		The bit is set when the txlatecol counter reaches half the maximum value	0
16	R/SS/ RC		The bit is set when the txdeferred counter reaches half the maximum value	0
15	R/SS/ RC		The bit is set when the txmulticol_g counter reaches half the maximum value	0
14	R/SS/ RC		The bit is set when the txsinglecol_g counter reaches half the maximum value.	0
13	R/SS/ RC		The bit is set when the txunderflowerror counter reaches half the maximum value	0
12	R/SS/ RC		The bit is set when the txbroadcastframes_gb counter reaches half the maximum value	0
11	R/SS/ RC		The bit is set when the txmulticastframes_gb counter reaches half the maximum value	0
10	R/SS/ RC		The bit is set when the txunicastframes_gb counter reaches half the maximum value	0
9	R/SS/ RC		The bit is set when the tx1024tomaxoctets_gb counter reaches half the maximum value	0
8	R/SS/ RC		The bit is set when the tx512to1023octets_gb counter reaches half the maximum value	0
7	R/SS/ RC		The bit is set when the tx256to511octets_gb counter reaches half the maximum value	0
6	R/SS/ RC		The bit is set when the tx128to255octets_gb counter reaches half the maximum value	0
5	R/SS/ RC		The bit is set when the tx65to127octets_gb counter reaches half the maximum value	0
4	R/SS/ RC		The bit is set when the tx32to63octets_gb counter reaches half the maximum value	0
3	R/SS/ RC		The bit is set when the txmulticastframes_g counter reaches half the maximum value.	0

Table 279: EMAC_MMCI_NTR_TX_REG (0xFF2108)

Bit	Mode	Symbol	Description	Reset
2	R/SS/ RC		The bit is set when the txbroadcastframes_g counter reaches half the maximum value	0
1	R/SS/ RC		The bit is set when the txframecount_gb counter reaches half the maximum value	0
0	R/SS/ RC		The bit is set when the txoctetcount_gb counter reaches half the maximum value.	0

Table 280: EMAC_MMCI_NTR_MASK_RX_REG (0xFF210C)

Bit	Mode	Symbol	Description	Reset
31- 24	-	-	Reserved	0
23	R/W		Setting this bit masks the interrupt when the rxwatchdog counter reaches half the maximum value.	0
22	R/W		Setting this bit masks the interrupt when the rxvlanframes_gb counter reaches half the maximum value.	0
21	R/W		Setting this bit masks the interrupt when the rxfifooverflow counter reaches half the maximum value.	0
20	R/W		Setting this bit masks the interrupt when the rxpauseframes counter reaches half the maximum value.	0
...	R/W		... Same as above for corresponding counters in MMC Receive Interrupt Register	0
1	R/W		Setting this bit masks the interrupt when the rxoctetcount_gb counter reaches half the maximum value.	0
0	R/W		Setting this bit masks the interrupt when the rxframecount_gb counter reaches half the maximum value.	0

Table 281: EMAC_MMCI_NTR_MASK_TX_REG (0xFF2110)

Bit	Mode	Symbol	Description	Reset
31- 25	-	-	Reserved	0
24	R/W		Setting this bit masks the interrupt when the txvlanframes_g counter reaches half the maximum value.	
23	R/W		Setting this bit masks the interrupt when the txpauseframes counter reaches half the maximum value.	0
22	R/W		Setting this bit masks the interrupt when the txoexcessdef counter reaches half the maximum value.	0
21	R/W		Setting this bit masks the interrupt when the txframecount_g counter reaches half the maximum value.	0
...	R/W		... Same as above for corresponding counters in MMC Transmit Interrupt Register	0
1	R/W		Setting this bit masks the interrupt when the txframecount_gb counter reaches half the maximum value.	0
0	R/W		Setting this bit masks the interrupt when the txoctetcount_gb counter reaches half the maximum value.	0

Table 282: EMAC_MMCI_TXUNICASTFRAMES_GB_REG (0xFF213C)

Bit	Mode	Symbol	Description	Reset
31-0	R/W		Number of good and bad unicast frames transmitted	0

Table 283: EMAC_MMC_TXUNDERFLOWERROR_REG (0xFF2148)

Bit	Mode	Symbol	Description	Reset
31-0	R/W		Number of frames aborted due to frame underflow error	0

Table 284: EMAC_MMC_TXSINGLECOL_G_REG (0xFF214C)

Bit	Mode	Symbol	Description	Reset
31-0	R/W		Number of successfully transmitted frames after a single collision in Half-duplex mode	0

Table 285: EMAC_MMC_TXMULTICOL_G_REG (0xFF2150)

Bit	Mode	Symbol	Description	Reset
31-0	R/W		Number of successfully transmitted frames after more than a single collision in Half-duplex mode	0

Table 286: EMAC_MMC_TXLATECOL_REG (0xFF2158)

Bit	Mode	Symbol	Description	Reset
31-0	R/W		Number of frames aborted due to late collision error	0

Table 287: EMAC_MMC_TXCARRIERERROR_REG (0xFF2160)

Bit	Mode	Symbol	Description	Reset
31-0	R/W		Number of frames aborted due to carrier sense error (no carrier or loss of carrier).	0

Table 288: EMAC_MMC_TXOCTETCOUNT_G_REG (0xFF2164)

Bit	Mode	Symbol	Description	Reset
31-0	R/W		Number of bytes transmitted, exclusive of preamble, in good frames only.	0

Table 289: EMAC_MMC_TXFRAMECOUNT_G_REG (0xFF2168)

Bit	Mode	Symbol	Description	Reset
31-0	R/W		Number of good frames transmitted	0

Table 290: EMAC_MMC_TXVLANFRAMES_G_REG (0xFF2174)

Bit	Mode	Symbol	Description	Reset
31-0	R/W		Number of good VLAN frames transmitted, exclusive of retried frames.	0

Table 291: EMAC_MMC_RXFRAMECOUNT_GB_REG (0xFF2180)

Bit	Mode	Symbol	Description	Reset
31-0	R/W		Number of good and bad frames received	0

Table 292: EMAC_MMC_RXOCTETCOUNT_G_REG (0xFF2188)

Bit	Mode	Symbol	Description	Reset
31-0	R/W		Number of bytes received, exclusive of preamble, only in good frames.	0

Table 293: EMAC_MMC_RXBROADCASTFRAMES_G_REG (0xFF218C)

Bit	Mode	Symbol	Description	Reset
31-0	R/W		Number of good broadcast frames received	0

Table 294: EMAC_MMC_RXMULTICASTFRAMES_G_REG (0xFF2190)

Bit	Mode	Symbol	Description	Reset
31-0	R/W		Number of good multicast frames received	0

Table 295: EMAC_MMC_RXJABBERERROR_REG (0xFF21A0)

Bit	Mode	Symbol	Description	Reset
31-0	R/W		Number of giant frames received with length (including CRC) greater than 1,518 bytes (1,522 bytes for VLAN tagged) and with CRC error. If Jumbo Frame mode is enabled, then frames of length greater than 9,018 bytes (9,022 for VLAN tagged) are considered as giant frames.	0

Table 296: EMAC_MMC_RXUNDERSIZE_G_REG (0xFF21A4)

Bit	Mode	Symbol	Description	Reset
31-0	R/W		Number of frames received with length less than 64 bytes, without any errors	0

Table 297: EMAC_MMC_RXOVERSIZE_G_REG (0xFF21A8)

Bit	Mode	Symbol	Description	Reset
31-0	R/W		Number of frames received with length greater than the max-size (1,518 or 1,522 for VLAN tagged frames), without errors.	0

Table 298: EMAC_MMC_RXUNICASTFRAMES_G_REG (0xFF21C4)

Bit	Mode	Symbol	Description	Reset
31-0	R/W		Number of good unicast frames received	0

Table 299: EMAC_MMC_RXFIFO_OVERFLOW_REG (0xFF21D4)

Bit	Mode	Symbol	Description	Reset
31-0	R/W		Number of missed received frames due to FIFO overflow.	0

Table 300: EMAC_MMC_RXVLANFRAMES_GB_REG (0xFF21D8)

Bit	Mode	Symbol	Description	Reset
31-0	R/W		Number of good and bad VLAN frames received	0

Table 301: EMAC_DMAR0_BUS_MODE_REG (0xFF3000)

Bit	Mode	Symbol	Description	Reset
31-26	-	-	Reserved	0
25	R/W	AAL	Address-Aligned Beats When this bit is set high and the FB bit equals 1, the AHB interface generates all bursts aligned to the start address LS bits. If the FB bit equals 0, the first burst (accessing the data buffer's start address) is not aligned, but subsequent bursts are aligned to the address.	0
24	R/W		4xPBL Mode When set high, this bit multiplies the PBL value programmed (bits [22:17] and bits [13:8]) four times. Thus the DMA will transfer data in to a maximum of 4, 8, 16, 32, 64 and 128 beats depending on the PBL value.	0
23	R/W	USP	Use Separate PBL When set high, it configures the RxDMA to use the value configured in bits [22:17] as PBL while the PBL value in bits [13:8] is applicable to TxDMA operations only. When reset to low, the PBL value in bits [13:8] is applicable for both DMA engines	0
22-17	R/W	RPBL	RxDMA PBL These bits indicate the maximum number of beats to be transferred in one RxDMA transaction. This will be the maximum value that is used in a single block Read/Write. The RxDMA will always attempt to burst as specified in RPBL each time it starts a Burst transfer on the host bus. RPBL can be programmed with permissible values of 1, 2, 4, 8, 16, and 32. Any other value will result in undefined behavior. These bits are valid and applicable only when USP is set high.	0x01
16	R/W	FB	Fixed Burst This bit controls whether the AHB Master interface performs fixed burst transfers or not. When set, the AHB will use only SINGLE, INCR4, INCR8 or INCR16 during start of normal burst transfers. When reset, the AHB will use SINGLE and INCR burst transfer operations.	0
15-14	R/W	RXPR	Rx:Tx priority ratio. RxDMA requests given priority over TxDMA requests in the following ratio. This is valid only when the DA bit is reset. 00 = 1:1 01 = 2:1 10 = 3:1 11 = 4:1	0

Table 301: EMAC_DMAR0_BUS_MODE_REG (0xFF3000)

Bit	Mode	Symbol	Description	Reset												
13-8	R/W	PBL	<p>Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA transaction. This will be the maximum value that is used in a single block Read/Write. The DMA will always attempt to burst as specified in PBL each time it starts a Burst transfer on the host bus. PBL can be programmed with permissible values of 1, 2, 4, 8, 16, and 32. Any other value will result in undefined behavior. When USP is set high, this PBL value is applicable for TxDMA transactions only. The PBL values have the following limitations. The maximum number of beats (PBL) possible is limited by the size of the Tx FIFO and Rx FIFO in the MTL layer and the data bus width on the DMA. The FIFO has a constraint that the maximum beat supported is half the depth of the FIFO. For different data bus widths and FIFO sizes, the valid PBL range (including x4 mode) is provided in the following table. If the PBL is common for both transmit and receive DMA, the minimum Rx FIFO and Tx FIFO depths must be considered. Do not program out-of-range PBL values, because the system may not behave properly.</p> <table border="1"> <thead> <tr> <th>Data Bus Width</th> <th>FIFO Depth (Bytes)</th> <th>Valid PBL Range in Full Duplex Mode</th> <th>Valid PBL Range in Half Duplex Mode</th> </tr> </thead> <tbody> <tr> <td>32</td> <td>128</td> <td>16 or less</td> <td>8 or less</td> </tr> <tr> <td></td> <td>256</td> <td>32 or less</td> <td>32 or less</td> </tr> </tbody> </table>	Data Bus Width	FIFO Depth (Bytes)	Valid PBL Range in Full Duplex Mode	Valid PBL Range in Half Duplex Mode	32	128	16 or less	8 or less		256	32 or less	32 or less	0x01
Data Bus Width	FIFO Depth (Bytes)	Valid PBL Range in Full Duplex Mode	Valid PBL Range in Half Duplex Mode													
32	128	16 or less	8 or less													
	256	32 or less	32 or less													
7	-	-	Reserved	0												
6-2	R/W	DSL	<p>Descriptor Skip Length This bit specifies the number of Word/Dword/Lword (depending on 32/64/128-bit bus) to skip between two unchained descriptors. The address skipping starts from the end of current descriptor to the start of next descriptor. When DSL value equals zero, then the descriptor table is taken as contiguous by the DMA, in Ring mode.</p>	0												
1	R/W	DA	<p>DMA Arbitration scheme 0 = Round-robin with Rx:Tx priority given in bits [15:14] 1 = Rx has priority over Tx</p>	0												
0	R/WS/SC	SWR	<p>Software Reset When this bit is set, the MAC DMA Controller resets all MAC Subsystem internal registers and logic. It is cleared automatically after the reset operation has completed in all of the core clock domains. Read a 0 value in this bit before re-programming any register of the core.</p>	1												

Table 302: EMAC_DMAR1_TX_POLL_DEMAND_REG (0xFF3004)

Bit	Mode	Symbol	Description	Reset
31-0	R/WT	TPD	<p>Transmit Poll Demand When these bits are written with any value, the DMA reads the current descriptor pointed to by Register 18. If that descriptor is not available (owned by Host), transmission returns to the Suspend state and DMA Register 5[2] is asserted. If the descriptor is available, transmission resumes.</p>	0

Table 303: EMAC_DMAR2_RX_POLL_DEMAND_REG (0xFF3008)

Bit	Mode	Symbol	Description	Reset
31-0	R/WT	RPD	Receive Poll Demand When these bits are written with any value, the DMA reads the current descriptor pointed to by Register 19. If that descriptor is not available (owned by Host), reception returns to the Suspended state and Register 5[7] is not asserted. If the descriptor is available, the Receive DMA returns to active state.	0

Table 304: EMAC_DMAR3_RX_DESCRIPTOR_LIST_ADDR_REG (0xFF300C)

Bit	Mode	Symbol	Description	Reset
31-0	R/W		Start of Receive List This field contains the base address of the First Descriptor in the Receive Descriptor list. The LSB bits [1/2/3:0] for 32/64/128-bit bus width) will be ignored and taken as all-zero by the DMA internally. Hence these LSB bits are Read Only.	0

Table 305: EMAC_DMAR4_TX_DESCRIPTOR_LIST_ADDR_REG (0xFF3010)

Bit	Mode	Symbol	Description	Reset
31-0	R/W		Start of Transmit List This field contains the base address of the First Descriptor in the Transmit Descriptor list. The LSB bits [1/2/3:0] for 32/64/128-bit bus width) will be ignored and taken as all-zero by the DMA internally. Hence these LSB bits are Read Only	0

Table 306: EMAC_DMAR5_STATUS_REG (0xFF3014)

Bit	Mode	Symbol	Description	Reset
31-30	-	-	Reserved	0
29	R	TTI	Time-Stamp Trigger Interrupt This bit indicates an interrupt event in the MAC core's Time Stamp Generator block. The software must read the MAC core's Interrupt Status register, clearing its source (Bit 9), to reset this bit to 1'b0. When this bit is high, the interrupt signal from the MAC subsystem is high.	0
28	-	-	Reserved	0
27	R	GMI	MAC MMC Interrupt This bit reflects an interrupt event in the MMC module of the MAC core. The software must read the corresponding registers in the MAC core to get the exact cause of interrupt and clear the source of interrupt to make this bit as 1'b0. The interrupt signal from the MAC subsystem is high when this bit is high.	0
26	R	GLI	MAC Line interface Interrupt This bit reflects an interrupt event in the MAC Core's PCS or RMII interface block. The software must read the corresponding registers in the MAC core to get the exact cause of interrupt and clear the source of interrupt to make this bit as 1'b0. The interrupt signal from the MAC subsystem is high when this bit is high.	0

Table 306: EMAC_DMAR5_STATUS_REG (0xFF3014)

Bit	Mode	Symbol	Description	Reset																		
25-23	R	EB	<p>Error Bits These bits indicate the type of error that caused a Bus Error (error response on the AHB interface). Valid only with Fatal Bus Error bit (Register 5[13]) set. This field does not generate an interrupt.</p> <table> <tr> <td>Bit 23</td> <td>1'b1</td> <td>Error during data transfer by TxDMA</td> </tr> <tr> <td></td> <td>1'b0</td> <td>Error during data transfer by RxDMA</td> </tr> <tr> <td>Bit 24</td> <td>1'b1</td> <td>Error during read transfer</td> </tr> <tr> <td></td> <td>1'b0</td> <td>Error during write transfer</td> </tr> <tr> <td>Bit 25</td> <td>1'b1</td> <td>Error during descriptor access</td> </tr> <tr> <td></td> <td>1'b0</td> <td>Error during data buffer access</td> </tr> </table>	Bit 23	1'b1	Error during data transfer by TxDMA		1'b0	Error during data transfer by RxDMA	Bit 24	1'b1	Error during read transfer		1'b0	Error during write transfer	Bit 25	1'b1	Error during descriptor access		1'b0	Error during data buffer access	0
Bit 23	1'b1	Error during data transfer by TxDMA																				
	1'b0	Error during data transfer by RxDMA																				
Bit 24	1'b1	Error during read transfer																				
	1'b0	Error during write transfer																				
Bit 25	1'b1	Error during descriptor access																				
	1'b0	Error during data buffer access																				
22-20	R	TS	<p>Transmit Process State These bits indicate the Transmit DMA FSM state. This field does not generate an interrupt.</p> <ul style="list-style-type: none"> 3'b000 = Stopped; Reset or Stop Transmit Command issued. 3'b001 = Running; Fetching Transmit Transfer Descriptor. 3'b010 = Running; Waiting for status. 3'b011 = Running; Reading Data from host memory buffer and queuing it to transmit buffer (Tx FIFO). 3'b100, 3'b101 = Reserved for future use. 3'b110 = Suspended; Transmit Descriptor Unavailable or Transmit Buffer Underflow. 3'b111 = Running; Closing Transmit Descriptor. 	0																		
19-17	R	RS	<p>Receive Process State These bits indicate the Receive DMA FSM state. This field does not generate an interrupt.</p> <ul style="list-style-type: none"> 3'b000 = Stopped; Reset or Stop Receive Command issued. 3'b001 = Running; Fetching Receive Transfer Descriptor. 3'b010 = Reserved for future use. 3'b011 = Running; Waiting for receive packet. 3'b100 = Suspended; Receive Descriptor Unavailable. 3'b101 = Running; Closing Receive Descriptor. 3'b110 = Reserved for future use. 3'b111 = Running; Transferring the receive packet data from receive buffer to host memory. 	0																		
16	R/SS/WC	NIS	<p>Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in DMA Register 7:</p> <ul style="list-style-type: none"> Register 5[0] = Transmit Interrupt Register 5[2] = Transmit Buffer Unavailable Register 5[6] = Receive Interrupt Register 5[14] = Early Receive Interrupt <p>Only unmasked bits affect the Normal Interrupt Summary bit. This is a sticky bit and must be cleared (by writing a 1 to this bit) each time a corresponding bit that causes NIS to be set is cleared.</p>	0																		

Table 306: EMAC_DMAR5_STATUS_REG (0xFF3014)

Bit	Mode	Symbol	Description	Reset
15	R/SS/ WC	AIS	<p>Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in DMA Register 7:</p> <ul style="list-style-type: none"> Register 5[1] = Transmit Process Stopped Register 5[3] = Transmit Jabber Timeout Register 5[4] = Receive FIFO Overflow Register 5[5] = Transmit Underflow Register 5[7] = Receive Buffer Unavailable Register 5[8] = Receive Process Stopped Register 5[9] = Receive Watchdog Timeout Register 5[10] = Early Transmit Interrupt Register 5[13] = Fatal Bus Error <p>Only unmasked bits affect the Abnormal Interrupt Summary bit.</p> <p>This is a sticky bit and must be cleared each time a corresponding bit that causes AIS to be set is cleared.</p>	0
14	R/SS/ WC	ERI	<p>Early Receive Interrupt This bit indicates that the DMA had filled the first data buffer of the packet. Receive Interrupt Register 5[6] automatically clears this bit.</p>	0
13	R/SS/ WC	FBI	<p>Fatal Bus Error Interrupt This bit indicates that a bus error occurred, as detailed in [25:23]. When this bit is set, the corresponding DMA engine disables all its bus accesses.</p>	0
12-11	-	-	Reserved	0
10	R/SS/ WC	ETI	<p>Early Transmit Interrupt This bit indicates that the frame to be transmitted was fully transferred to the MTL Transmit FIFO.</p>	0
9	R/SS/ WC	RWT	<p>Receive Watchdog Timeout This bit is asserted when a frame with a length greater than 2,048 bytes is received (10,240 when Jumbo Frame mode is enabled).</p>	0
8	R/SS/ WC	RPS	<p>Receive Process Stopped This bit is asserted when the Receive Process enters the Stopped state</p>	0
7	R/SS/ WC	RU	<p>Receive Buffer Unavailable This bit indicates that the Next Descriptor in the Receive List is owned by the host and cannot be acquired by the DMA. Receive Process is suspended. To resume processing Receive descriptors, the host should change the ownership of the descriptor and issue a Receive Poll Demand command. If no Receive Poll Demand is issued, Receive Process resumes when the next recognized incoming frame is received. Register 5[7] is set only when the previous Receive Descriptor was owned by the DMA.</p>	0
6	R/SS/ WC	RXINT	<p>Receive Interrupt This bit indicates the completion of frame reception. Specific frame status information has been posted in the descriptor. Reception remains in the Running state.</p>	0
5	R/SS/ WC	UNF	<p>Transmit Underflow This bit indicates that the Transmit Buffer had an Underflow during frame transmission. Transmission is suspended and an Underflow Error TDES0[1] is set</p>	0

Table 306: EMAC_DMAR5_STATUS_REG (0xFF3014)

Bit	Mode	Symbol	Description	Reset
4	R/SS/ WC	OVF	Receive Overflow This bit indicates that the Receive Buffer had an Overflow during frame reception. If the partial frame is transferred to application, the overflow status is set in RDES0[11].	0
3	R/SS/ WC	TJT	Transmit Jabber Timeout This bit indicates that the Transmit Jabber Timer expired, meaning that the transmitter had been excessively active. The transmission process is aborted and placed in the Stopped state. This causes the Transmit Jabber Timeout TDES0[14] flag to assert.	0
2	R/SS/ WC	TU	Transmit Buffer Unavailable This bit indicates that the Next Descriptor in the Transmit List is owned by the host and cannot be acquired by the DMA. Transmission is suspended. Bits[22:20] explain the Transmit Process state transitions. To resume processing transmit descriptors, the host should change the ownership of the bit of the descriptor and then issue a Transmit Poll Demand command	0
1	R/SS/ WC	TPS	Transmit Process Stopped This bit is set when the transmission is stopped.	0
0	R/SS/ WC	TXINT	Transmit Interrupt This bit indicates that frame transmission is finished and TDES1[31] is set in the First Descriptor	0

Table 307: EMAC_DMAR6_OPERATION_MODE_REG (0xFF3018)

Bit	Mode	Symbol	Description	Reset
31- 27	-	-	Reserved	0
26	R/W	DT	Disable Dropping of TCP/IP Checksum Error Frames When this bit is set, the core does not drop frames that only have errors detected by the Receive Checksum Offload engine. Such frames do not have any errors (including FCS error) in the Ethernet frame received by the MAC but have errors in the encapsulated payload only. When this bit is reset, all error frames are dropped if the FEF bit is reset. If the Full Checksum Offload engine (Type 2) is disabled, this bit is reserved (RO with value 1'b0).	0
25	R/W	RSF	Receive Store and Forward When this bit is set, the MTL only reads a frame from the Rx FIFO after the complete frame has been written to it, ignoring RTC bits. When this bit is reset, the Rx FIFO operates in Cut-Through mode, subject to the threshold specified by the RTC bits.	0
24	R/W	DFF	Disable Flushing of Received Frames When this bit is set, the RxDMA does not flush any frames due to the unavailability of receive descriptors/buffers as it does normally when this bit is reset. (See "Receive Process Suspended")	0
23	-	-	Reserved	0
22	-	-	Reserved	0

Table 307: EMAC_DMAR6_OPERATION_MODE_REG (0xFF3018)

Bit	Mode	Symbol	Description	Reset
21	R/W	TSF	Transmit Store and Forward When this bit is set, transmission starts when a full frame resides in the MTL Transmit FIFO. When this bit is set, the TTC values specified in Register 6[16:14] are ignored. This bit should be changed only when transmission is stopped.	0
20	R/ WS/ SC	FTF	Flush Transmit FIFO When this bit is set, the transmit FIFO controller logic is reset to its default values and thus all data in the Tx FIFO is lost/flushed. This bit is cleared internally when the flushing operation is completed fully. The Operation Mode register should not be written to until this bit is cleared.	0
19- 17	-	-	Reserved	0
16- 14	R/W	TTC	Transmit Threshold Control These three bits control the threshold level of the MTL Transmit FIFO. Transmission starts when the frame size within the MTL Transmit FIFO is larger than the threshold. In addition, full frames with a length less than the threshold are also transmitted. These bits are used only when the TSF bit (Bit 21) is reset. 000 = 64 001 = 128 010 = 192 011 = 256 100 = 40 101 = 32 110 = 24 111 = 16	0
13	R/W	ST	Start/Stop Transmission Command When this bit is set, transmission is placed in the Running state, and the DMA checks the Transmit List at the current position for a frame to be transmitted. Descriptor acquisition is attempted either from the current position in the list, which is the Transmit List Base Address set by Register 4, or from the position retained when transmission was stopped previously. If the current descriptor is not owned by the DMA, transmission enters the Suspended state and Transmit Buffer Unavailable (Register 5[2]) is set. The Start Transmission command is effective only when transmission is stopped. If the command is issued before setting DMA Register 4, then the DMA behavior is unpredictable. When this bit is reset, the transmission process is placed in the Stopped state after completing the transmission of the current frame. The Next Descriptor position in the Transmit List is saved, and becomes the current position when transmission is restarted. The stop transmission command is effective only the transmission of the current frame is complete or when the transmission is in the Suspended state.	0

Table 307: EMAC_DMAR6_OPERATION_MODE_REG (0xFF3018)

Bit	Mode	Symbol	Description	Reset
12-11	R/W	RFD	<p>Threshold for deactivating flow control (in both HD and FD)</p> <p>These bits control the threshold (Fill-level of Rx FIFO) at which the flow-control is deasserted after activation.</p> <p>00 = Full minus 1 KB 01 = Full minus 2 KB 10 = Full minus 3 KB 11 = Full minus 4 KB</p> <p>Note that the deassertion is effective only after flow control is asserted. If the Rx FIFO is 8 KB or more, an additional bit (RFD[2]) is used for more threshold levels as described in bit [22].</p>	0
10-9	R/W	RFA	<p>Threshold for activating flow control (in both HD and FD)</p> <p>These bits control the threshold (Fill level of Rx FIFO) at which flow control is activated.</p> <p>00 = Full minus 1 KB 01 = Full minus 2 KB 10 = Full minus 3 KB 11 = Full minus 4 KB</p> <p>Note that the above only applies to Rx FIFOs of 4 KB or more when the EFC bit is set high. If the Rx FIFO is 8 KB or more, an additional bit (RFA[2]) is used for more threshold levels as described in bit [23].</p>	0
8	R/W	EFC	<p>Enable HW flow control</p> <p>When this bit is set, the flow control signal operation based on fill-level of Rx FIFO is enabled. When reset, the flow control operation is disabled. This bit is not used (reserved and always reset) when the Rx FIFO is less than 4 KB.</p>	0
7	R/W	FEF	<p>Forward Error Frames</p> <p>When this bit is reset, the Rx FIFO drops frames with error status (CRC error, collision error, GMII_ER, giant frame, watchdog timeout, overflow). However, if the frame's start byte (write) pointer is already transferred to the read controller side (in Threshold mode), then the frames are not dropped. Note that the Rx FIFO drops the error frames if that frame's start byte is not transferred (output) on the ARI bus. When FEF is set, all frames except runt error frames are forwarded to the DMA.</p>	0
6	R/W	FUF	<p>Forward Undersized Good Frames</p> <p>When set, the Rx FIFO will forward Undersized frames (frames with no Error and length less than 64 bytes) including pad-bytes and CRC).</p> <p>When reset, the Rx FIFO will drop all frames of less than 64 bytes, unless it is already transferred due to lower value of Receive Threshold (e.g., RTC = 01).</p>	0
5	-	-	Reserved	0

Table 307: EMAC_DMAR6_OPERATION_MODE_REG (0xFF3018)

Bit	Mode	Symbol	Description	Reset
4-3	R/W	RTC	<p>Receive Threshold Control</p> <p>These two bits control the threshold level of the MTL Receive FIFO. Transfer (request) to DMA starts when the frame size within the MTL Receive FIFO is larger than the threshold. In addition, full frames with a length less than the threshold are transferred automatically. Note that value of 11 is not applicable if the configured Receive FIFO size is 128 bytes. These bits are valid only when the RSF bit is zero, and are ignored when the RSF bit is set to 1.</p> <p>00 = 64 01 = 32 10 = 96 11 = 128</p>	0
2	R/W	OSF	<p>Operate on Second Frame</p> <p>When this bit is set, this bit instructs the DMA to process a second frame of Transmit data even before status for first frame is obtained</p>	0
1	R/W	SR	<p>Start/Stop Receive</p> <p>When this bit is set, the Receive process is placed in the Running state. The DMA attempts to acquire the descriptor from the Receive list and processes incoming frames. Descriptor acquisition is attempted from the current position in the list, which is the address set by DMA Register 3 or the position retained when the Receive process was previously stopped. If no descriptor is owned by the DMA, reception is suspended and Receive Buffer Unavailable (Register 5[7]) is set. The Start Receive command is effective only when reception has stopped. If the command was issued before setting DMA Register 3, DMA behavior is unpredictable.</p> <p>When this bit is cleared, RxDMA operation is stopped after the transfer of the current frame. The next descriptor position in the Receive list is saved and becomes the current position after the Receive process is restarted. The Stop Receive command is effective only when the Receive process is in either the Running (waiting for receive packet) or in the Suspended state.</p>	0
0	-	-	Reserved	0

Table 308: EMAC_DMAR7_INT_ENABLE_REG (0xFF301C)

Bit	Mode	Symbol	Description	Reset
31-17	-	-	Reserved	0
16	R/W	NIE	Normal Interrupt Summary Enable When this bit is set, a normal interrupt is enabled. When this bit is reset, a normal interrupt is disabled. This bit enables the following bits: Register 5[0] = Transmit Interrupt Register 5[2] = Transmit Buffer Unavailable Register 5[6] = Receive Interrupt Register 5[14] = Early Receive Interrupt	0
15	R/W	AIE	Abnormal Interrupt Summary Enable When this bit is set, an Abnormal Interrupt is enabled. When this bit is reset, an Abnormal Interrupt is disabled. This bit enables the following bits Register 5[1] = Transmit Process Stopped Register 5[3] = Transmit Jabber Timeout Register 5[4] = Receive Overflow Register 5[5] = Transmit Underflow Register 5[7] = Receive Buffer Unavailable Register 5[8] = Receive Process Stopped Register 5[9] = Receive Watchdog Timeout Register 5[10] = Early Transmit Interrupt Register 5[13] = Fatal Bus Error	0
14	R/W	ERE	Early Receive Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (Register 7[16]), Early Receive Interrupt is enabled. When this bit is reset, Early Receive Interrupt is disabled	0
13	R/W	FBE	Fatal Bus Error Enable When this bit is set with Abnormal Interrupt Summary Enable (Register 7[15]), the Fatal Bus Error Interrupt is enabled. When this bit is reset, Fatal Bus Error Enable Interrupt is disabled.	0
12-11	-	-	Reserved	0
10	R/W	ETE	Early Transmit Interrupt Enable When this bit is set with an Abnormal Interrupt Summary Enable (Register 7[15]), Early Transmit Interrupt is enabled. When this bit is reset, Early Transmit Interrupt is disabled.	0
9	R/W	RWE	Receive Watchdog Timeout Enable When this bit is set with Abnormal Interrupt Summary Enable (Register 7[15]), the Receive Watchdog Timeout Interrupt is enabled. When this bit is reset, Receive Watchdog Timeout Interrupt is disabled.	0
8	R/W	RSE	Receive Stopped Enable When this bit is set with Abnormal Interrupt Summary Enable (Register 7[15]), Receive Stopped Interrupt is enabled. When this bit is reset, Receive Stopped Interrupt is disabled.	0
7	R/W	RUE	Receive Buffer Unavailable Enable When this bit is set with Abnormal Interrupt Summary Enable (Register 7[15]), Receive Buffer Unavailable Interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable Interrupt is disabled.	0
6	R/W	RIE	Receive Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (Register 7[16]), Receive interrupt is enabled. When this bit is reset, Receive Interrupt is disabled.	0

Table 308: EMAC_DMAR7_INT_ENABLE_REG (0xFF301C)

Bit	Mode	Symbol	Description	Reset
5	R/W	UNE	Underflow Interrupt Enable When this bit is set with Abnormal Interrupt Summary Enable (Register 7[15]), Transmit Underflow Interrupt is enabled. When this bit is reset, Underflow Interrupt is disabled.	0
4	R/W	OVE	Overflow Interrupt Enable When this bit is set with Abnormal Interrupt Summary Enable (Register 7[15]), Receive Overflow Interrupt is enabled. When this bit is reset, Overflow Interrupt is disabled	0
3	R/W	TJE	Transmit Jabber Timeout Enable When this bit is set with Abnormal Interrupt Summary Enable (Register 7[15]), Transmit Jabber Timeout Interrupt is enabled. When this bit is reset, Transmit Jabber Timeout Interrupt is disabled.	0
2	R/W	TUE	Transmit Buffer Unavailable Enable When this bit is set with Normal Interrupt Summary Enable (Register 7[16]), Transmit Buffer Unavailable Interrupt is enabled. When this bit is reset, Transmit Buffer Unavailable Interrupt is disabled.	0
1	R/W	TSE	Transmit Stopped Enable When this bit is set with Abnormal Interrupt Summary Enable (Register 7[15]), Transmission Stopped Interrupt is enabled. When this bit is reset, Transmission Stopped Interrupt is disabled.	0
0	R/W	TIE	Transmit Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (Register 7[16]), Transmit Interrupt is enabled. When this bit is reset, Transmit Interrupt is disabled.	

Table 309: EMAC_DMAR8_MISSFRAME_BUFOVR_CNT_REG (0xFF3020)

Bit	Mode	Symbol	Description	Reset
31-29	-	-	Reserved	0
28	R/SS/RC		Overflow bit for FIFO Overflow Counter	0
27-17	R/SS/RC		Indicates the number of frames missed by the application. This counter is incremented each time the MTL asserts the sideband signal {MTL RxOverflow}. The counter is cleared when this register is read.	0
16	R/SS/RC		Overflow bit for Missed Frame Counter	0
15-0	R/SS/RC		Indicates the number of frames missed by the controller due to the Host Receive Buffer being unavailable. This counter is incremented each time the DMA discards an incoming frame. The counter is cleared when this register is read with mci_be_i[0] at 1'b1.	0

Table 310: EMAC_DMAR18_CUR_HOST_TX_DESCRIPTOR_REG (0xFF3048)

Bit	Mode	Symbol	Description	Reset
31-0	R		Host Transmit Descriptor Address Pointer Cleared on Reset. Pointer updated by DMA during operation	0

Table 311: EMAC_DMAR19_CUR_HOST_RX_DESCRIPTOR_REG (0xFF304C)

Bit	Mode	Symbol	Description	Reset
31-0	R		Host Receive Descriptor Address Pointer Cleared on Reset. Pointer updated by DMA during operation.	0

Table 312: EMAC_DMAR20_CUR_HOST_TX_BUF_ADDR_REG (0xFF3050)

Bit	Mode	Symbol	Description	Reset
31-0	R		Host Transmit Buffer Address Pointer Cleared on Reset. Pointer updated by DMA during operation	0

Table 313: EMAC_DMAR21_CUR_HOST_RX_BUF_ADDR_REG (0xFF3054)

Bit	Mode	Symbol	Description	Reset
31-0	R		Host Receive Buffer Address Pointer Cleared on Reset. Pointer updated by DMA during operation	0

Table 314: RESET_FREEZE_REG (0xFF5002)

Bit	Mode	Symbol	Description	Reset
15-8	-	-	Reserved	0
7	R/W	FRZ_DMA3	If 1 DMA channel 3 continues. 0 is discarded	0
6	R/W	FRZ_DMA2	If 1 DMA channel 2 continues. 0 is discarded	0
5	R/W	FRZ_DMA1	If 1 DMA channel 1 continues. 0 is discarded	0
4	R/W	FRZ_DMA0	If 1 DMA channel 0 continues. 0 is discarded	0
3	R/W	FRZ_WDOG	If 1 the watchdog timer continues. 0 is discarded	0
2	R/W	FRZ_TIM1	If 1 timer 1 continues. 0 is discarded	0
1	R/W	FRZ_TIM0	If 1 timer 0 continues. 0 is discarded	0
0	R/W	FRZ_DIP	If 1 the DIP continues. 0 is discarded	0

Table 315: RESET_INT_PENDING_REG (0xFF5402)

Bit	Mode	Symbol	Description	Reset
15	-	-	Reserved	0
14	R/W	DSP2_INT_PEND	Writing a 1 clears the pending DSP 2 interrupt. On reading the DSP2 interrupt status (pending/not pending) is returned.	0
13	R/W	DSP1_INT_PEND	Writing a 1 clears the pending DSP 1 interrupt. On reading the DSP 1 interrupt status (pending/not pending) is returned.	0
12	R/W	SPI2_INT_PEND	Writing a 1 clears the pending SPI2 interrupt. On reading the SPI2 interrupt status (pending/not pending) is returned.	0
11	R/W	CRYPTO_INT_PEND	Writing a 1 clears the pending CRYPTO interrupt. On reading the CRYPTO interrupt status (pending/not pending) is returned.	0
10	R/W	DIP_INT_PEND	Writing a 1 clears the pending DIP interrupt. On reading the DIP interrupt status (pending/not pending) is returned.	0
9	R/W	CLK100_INT_PEND	Writing a 1 clears the pending Clock 100 interrupt. On reading the Clock 100 interrupt status (pending/not pending) is returned. See also DEBUG_REG[3-0]	0
8	R/W	TIM1_INT_PEND	Writing a 1 clears the pending Timer1 interrupt. On reading the Timer1 interrupt status (pending/not pending) is returned.	0

Table 315: RESET_INT_PENDING_REG (0xFF5402)

Bit	Mode	Symbol	Description	Reset
7	R/W	TIM0_INT_PEND	Writing a 1 clears the pending Timer0 interrupt. On reading the Timer0 interrupt status (pending/not pending) is returned.	0
6	R/W	SPI1_ADC_INT_PEND	Writing a 1 clears the pending SPI interrupt. On reading the SPI interrupt status (pending/not pending) is returned. The SPI_INT interrupt is shared with the ADC_INT interrupt	0
5	R/W	UART_TI_INT_PEND	Writing a 1 clears the pending UART TI interrupt. On reading the UART TI interrupt status (pending/not pending) is returned.	0
4	R/W	UART_RI_INT_PEND	Writing a 1 clears the pending UART RI interrupt. On reading the UART RI interrupt status (pending/not pending) is returned.	0
3	R/W	CT_CLASSD_INT_PEND	Writing a 1 clears the pending CT interrupt. On reading the CT interrupt status (pending/not pending) is returned. The CT interrupt is shared with the CLASSD interrupt.	0
2	R/W	EMAC_INT_PEND	Writing a 1 clears the pending EMAC interrupt. On reading the EMAC interrupt status (pending/not pending) is returned.	0
1	R/W	KEYB_INT_PEND	Writing a 1 clears the pending keyboard interrupt. On reading the keyboard interrupt status (pending/not pending) is returned.	0
0	R/W	ACCESS_INT_PEND	Writing a 1 clears the pending Access bus interrupt. On reading the Access bus interrupt status (pending/not pending) is returned.	0

Note 69: Do not use CR16C+ bit instruction on this register (read-modify-write will clear ALL pending bits)

Table 316: SET_FREEZE_REG (0xFF5000)

Bit	Mode	Symbol	Description	Reset
15-8	-	-	Reserved	0
7	R/W	FRZ_DMA3	If 1 the DMA channel 3 is frozen. 0 is discarded	0
6	R/W	FRZ_DMA2	If 1 the DMA channel 2 is frozen. 0 is discarded	0
5	R/W	FRZ_DMA1	If 1 the DMA channel 1 is frozen. 0 is discarded	0
4	R/W	FRZ_DMA0	If 1 the DMA channel 0 is frozen. 0 is discarded	0
3	R/W	FRZ_WDOG	If 1 the watchdog timer is frozen. 0 is discarded	0
2	R/W	FRZ_TIM1	If 1 timer 1 is frozen. 0 is discarded	0
1	R/W	FRZ_TIM0	If 1 timer 0 is frozen. 0 is discarded	0
0	R/W	FRZ_DIP	If 1 the DIP is frozen. 0 is discarded	0

Table 317: SET_INT_PENDING_REG (0xFF5400)

Bit	Mode	Symbol	Description	Reset
15	-	-	Reserved	0
14	R/W	DSP2_INT_PEND	If a GenDSP 2 interrupt is generated this bit is set to 1. Writing a 1 will also generate a DSP 2 interrupt. A 0 insertion is discarded. On reading the DSP 2 interrupt status (pending/not pending) is returned.	0
13	R/W	DSP1_INT_PEND	If a GenDSP 1 interrupt is generated this bit is set to 1. Writing a 1 will also generate a DSP1 interrupt. A 0 insertion is discarded. On reading the DSP 1 interrupt status (pending/not pending) is returned.	0

Table 317: SET_INT_PEND_REG (0xFF5400)

Bit	Mode	Symbol	Description	Reset
12	R/W	SPI2_INT_PEND	Writing a 1 sets the pending SPI2interrupt. On reading the SPI2 interrupt status (pending/not pending) is returned.	0
11	R/W	CRYPTO_INT_PEND	If a CRYPTO interrupt is generated this bit is set to 1. Writing a 1 will also generate a CRYPTO interrupt. A 0 insertion is discarded. On reading the CRYPTO interrupt status (pending/not pending) is returned.	0
10	R/W	DIP_INT_PEND	If a DIP interrupt is generated this bit is set to 1. Writing a 1 will also generate a DIP interrupt. A 0 insertion is discarded. On reading the DIP interrupt status (pending/not pending) is returned.	0
9	R/W	CLK100_INT_PEND	If a Clock 100 interrupt is generated this bit is set to 1. Writing a 1 will also generate a Clock 100 interrupt. A 0 insertion is discarded. On reading the Clock 100 interrupt status (pending/not pending) is returned. See also DEBUG_REG[3-0]	0
8	R/W	TIM1_INT_PEND	If a Timer1interrupt is generated this bit is set to 1. Writing a 1 will also generate a Timer1interrupt. A 0 insertion is discarded. On reading the Timer1interrupt status (pending/not pending) is returned.	0
7	R/W	TIM0_INT_PEND	If a Timer0 interrupt is generated this bit is set to 1. Writing a 1 will also generate a Timer0interrupt. A 0 insertion is discarded. On reading the Timer0interrupt status (pending/not pending) is returned.	0
6	R/W	SPI1_ADC_INT_PEND	If a SPI interrupt is generated this bit is set to 1. Writing a 1 will also generate a SPI1 interrupt. A 0 insertion is discarded. On reading the SPI1 interrupt status (pending/not pending) is returned. The SPI_INT interrupt is shared with the ADC_INT interrupt.	0
5	R/W	UART_TI_INT_PEND	If a UART TI interrupt is generated this bit is set to 1. Writing a 1 will also generate a UART TI interrupt. A 0 insertion is discarded. On reading the UART TI interrupt status (pending/not pending) is returned.	0
4	R/W	UART_RI_INT_PEND	If a UART RI interrupt is generated this bit is set to 1. Writing a 1 will also generate a UART RI interrupt. A 0 insertion is discarded. On reading the UART RI interrupt status (pending/not pending) is returned.	0
3	R/W	CT_CLASSD_INT_PEND	If a CT interrupt is generated this bit is set to 1. Writing a 1 will also generate a CT interrupt. A 0 insertion is discarded. On reading the CT interrupt status (pending/not pending) is returned. The CT interrupt is shared with the CLASSD interrupt.	0
2	R/W	EMAC_INT_PEND	Writing a 1 sets the pending EMAC interrupt. On reading the EMAC interrupt status (pending/not pending) is returned.	0
1	R/W	KEYB_INT_PEND	If a keyboard interrupt is generated this bit is set to 1. Writing a 1 will also generate a keyboard interrupt. A 0 insertion is discarded. On reading the keyboard interrupt status (pending/not pending) is returned.	0
0	R/W	ACCESS_INT_PEND	Writing a 1 will generate a Access Bus interrupt. A 0 insertion is discarded. On reading the Access Bus interrupt status (pending/not pending) is returned.	0

Table 318: SPI1_CTRL_REG0, SPI2_CTRL_REG0 (0xFF4940,0xFF4950)

Bit	Mode	Symbol	Description	Reset
15	R/W	SPI_EN_CTRL	0 = SPI_EN pin disabled(SPI1 only). In slave mode pin SPI_CLK is not gated. 1 = SPI_EN pin enabled(SPI1 only). In slave mode SPI_CLK is active if SPI_EN=0 and inactive if SPI_EN=1.	0
14	R/W	SPI_MINT	0 = Disable SPI_INT_BIT to ICU 1 = Enable SPI_INT_BIT to ICU. Note that the SPI_INT interrupt is shared with ADC_INT interrupt	0
13	R	SPI_INT_BIT	0 = RX Register or FIFO is empty. 1 = SPI interrupt. Data has been transmitted and received Must be reset by SW by writing to SPI_CLEAR_INT_REG.	0
12	R	SPI_DI	Returns the actual value of pin SPIx_DIN (delayed with two internal SPI clock cycles)	0
11	R	SPI_TXH	0 = TX-FIFO is not full, data can be written. 1 = TX-FIFO is full, data can not be written.	0
10	R/W	SPI_FORCE_DO	0 = normal operation 1 = Force SPIDO output level to value of SPI_DO.	0
9	R/W	SPI_RST	0 = normal operation 1 = Reset SPI. Same function as SPI_ON except that internal clock remain active.	0
8-7	R/W	SPI_WORD	00 = 8 bits mode, only SPI_RX_TX_REG0 used 01 = 16 bit mode, only SPI_RX_TX_REG0 used 10 = 32 bits mode, SPI_RX_TX_REG0 & SPI_RX_TX_REG1 used 11 = reserved	0
6	R/W	SPI_SMN	Master/slave mode 0 = Master, 1 = Slave(SPI1 only)	0
5	R/W	SPI_DO	Pin SPIx_DOUT output level when SPI is idle or when SPI_FORCE_DO=1	0
4-3	R/W	SPI_CLK	Set the clock frequency of the SPI-bus in master mode, relative to the frequency of the SPI internal clock. (Note 70) 00 = Divide by 8 01 = Divide by 4 10 = Divide by 2 11 = Divide by 14	0
2	R/W	SPI_POL	Select SPIx_CLK polarity. 0 = SPIx_CLK is initially low. 1 = SPIx_CLK is initially high.	0
1	R/W	SPI_PHA	Select SPIx_CLK phase. See Figure 67 to Figure 70	0
0	R/W	SPI_ON	0 = SPI Module switched off (power saving). Everything is reset except the corresponding SPIx_CTRL_REG0 and SPIx_CTRL_REG1 1 = SPI Module switched on. Should only be set after all control bits have their desired values. So two writes are needed!	0

Note 70: SPI block input clock is configured with CLK_GPIO1_REG,CLK_GPIO2_REG,CLK_GPIO3_REG

Table 319: SPI1_CTRL_REG1, SPI2_CTRL_REG1 (0xFF4948, 0xFF4958)

Bit	Mode	Symbol	Description	Reset
15-4	R	-	These registers are unused, so writing has no effect. Reading returns 0.	0
3	R	SPI_BUSY	0 = The SPI is not busy with a transfer. This means that either no TX-data is available or that the transfers have been suspended due to a full RX-FIFO. The SPIx_CTRL_REG0[SPI_INT_BIT] can be used to distinguish between these situations. 1 = The SPI is busy with a transfer.	0
2	R/W	SPI_PRIORITY	0 = The SPI has low priority, the DMA request signals are reset after the corresponding acknowledge. 1 = The SPI has high priority, DMA request signals remain active until the FIFOs are filled/emptied.	0
1-0	R/W	SPI_FIFO_MODE	00 = TX-FIFO and RX-FIFO used (Bidirectional mode). 01 = RX-FIFO used (Read Only Mode) TX-FIFO single depth, no flow control 10 = TX-FIFO used (Write Only Mode), RX-FIFO single depth, no flow control 11 = No FIFOs used (backwards compatible mode)	11

Table 320: SPI1_RX_TX_REG0, SPI2_RX_TX_REG0 (0xFF4942, 0xFF4952)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	SPI_DATA0	Write: SPI_TX_REG0 output register 0 (TX-FIFO) Read: SPI_RX_REG0 input register 0 (RX-FIFO) In 8 bits mode bits 15 to 8 are not used, they contain old data.	0

Table 321: SPI1_RX_TX_REG1, SPI2_RX_TX_REG1 (0xFF4944, 0xFF4954)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	SPI_DATA1	Write: SPI_TX_REG1 output register 1 (MSB's of TX-FIFO) Read: SPI_RX_REG1 input register 1 (MSB's of RX-FIFO) In 8 or 16 bits mode this register is not used.	0

Table 322: SPI1_CLEAR_INT_REG, SPI2_CLEAR_INT_REG (0xFF4946, 0xFF4956)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	SPI_CLEAR_INT	Writing any value to this register will clear the corresponding SPIx_CTRL_REG[SPI_INT_BIT]. Reading returns 0.	0

Table 323: TEST_CTRL_REG (0xFF4802)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	TEST	For test purpose only. Must be all 0 for normal operation	0

Table 324: TEST_CTRL2_REG (0xFF4804)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	TEST2	For test purpose only. Must be all 0 for normal operation	0

Table 325: TEST_ENV_REG (0xFF4800)

Bit	Mode	Symbol	Description	Reset
15-8	-	-	Reserved	0
7-5	R	ENV_REG7_5	Value of port P0[7-5] on rising edge of RSTn pin. Free environment bits for application. Internally used in test mode.	(Note 71)
4	R	ENV_SDI	0 = JTAG SDI+ enabled. (See "JTAG-SDI" on page 41). This bit is set to '0' if the JTAG pin is set to '0' for more than 20 us if the RSTn pin = '1'. Writing a '0' is discarded. 1 = JTAG SDI+ disabled. Writing a '1' to this register disables the JTAG SDI interface; JTAG pin must stay '1' else JTAG SDI+ is enabled again after 20 us. If this register is written as last command from the NTA, the JTAG pin stays high. Any new command enables the JTAG SDI again. This bit can modified by writing to bit 2	1
3	R	AD3	Read: Value of port AD[3] on rising edge of RSTn pin. used as reset value for SUPPLY_CTRL_REG (0xFF4812) DC_VOUT and LDO1_LEVEL.	(Note 71)
2	R/W	AD2	Read: Value of port AD[2] on rising edge of RSTn pin used as reset value for SUPPLY_CTRL_REG (0xFF4812) DC_VOUT and LDO1_LEVEL. Write: ENV_SDI. Reading this bit is done at bit 4	(Note 71)
1	R	AD1	Value of port AD[1] on rising edge of RSTn pin used as reset value for SUPPLY_CTRL_REG (0xFF4812) DC_VOUT and LDO1_LEVEL.	(Note 71)
0	R	BOOT	Value of port P0[0] on rising edge of RSTn pin. Controls the on-chip boot program execution (See "Power-On and system Reset" on page 34) 0 = The boot program loads a program from the UART and executes it from 0x8080. If DEBUG_REG[ENV_B01] = 1 no program is loaded but execution is started from 0x8080 immediately. 1 = The boot program continues booting SPI and/or jumps to off-chip memory	(Note 71)

Note 71: Value on rising edge of RSTn

Table 326: TIMER_CTRL_REG (0xFF4970)

Bit	Mode	Symbol	Description	Reset
15-7	-	-	Reserved	0
6	R/W	CLK_DIV8	0 = Timer1 uses as time base the clock that the System Clock Generator provides. 1 = Timer1 uses as time base the clock that the System Clock Generator provides divided by 8.	0
5	R/W	TIM1_MODE	0 = Timer1 mode1 selected 1 = Timer1 mode2 selected.	0
4	R/W	WDOG_CTRL	0 = Watchdog Timer generates NMI. 1 = Watchdog Timer generates hardware Reset signal and watchdog cannot be frozen. Note that this bit can only be set to 1 by SW and only be reset with a hardware reset.	0
3-2	-	-	Reserved	0
1	R/W	TIM1_CTRL	0 = Timer1 is off and in reset state. 1 = Timer1 is running.	0
0	R/W	TIM0_CTRL	0 = Timer0 is off and in reset state. 1 = Timer0 is running.	0

Table 327: TIMER0_ON_REG (0xFF4972)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	TIM0_ON	Timer0 On reload value: If read the actual counter value ON_CNTer is returned	0

Table 328: TIMER0_RELOAD_M_REG (0xFF4974)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	TIM0_M	Timer0 'high' reload value If read the actual counter value T0_CNTer is returned	0

Table 329: TIMER0_RELOAD_N_REG (0xFF4976)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	TIM0_N	Timer0 'low' reload value: If read the actual counter value T0_CNTer is returned	0

Table 330: TIMER1_RELOAD_M_REG (0xFF4978)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	TIM1_M	Timer1 'high' reload value: If read TIM1_M value is returned	0

Table 331: TIMER1_RELOAD_N_REG (0xFF497A)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	TIM1_N	Timer1 'low' reload value: If read the actual counter value T1_CNTer is returned	0

Table 332: TONE_CTRL1_REG(0xFF4990)

Bit	Mode	Symbol	Description	Reset
15-11	-	-	Reserved	0
10	R/W	GATE_EDGE1	0 = Rising edge of GATE_CLK used to clock TONE_TIMER 1 = Rising and falling edge of GATE_CLK used	0
9	R/W	MCT1_INT	0 = Disable CT1_INT to CT_CLASSD_INT 1 = Enable CT1_INT to CT_CLASSD_INT, (Note 72)	0
8	R	CT1_INT	1 = Tone latch 1 interrupt. Must be cleared by writing to TONE_CLEAR_INT1_REG	0
7-4	R/W	TIMER_RELOAD1	Select clock divider reload value for clock to latch timer periodically 0 = divide by 1, 15 = divide by 16	0
3-2	R/W	CLKSRC1	00 = 144 kHz 01 = reserved 10 = ECZ1 from GenDSP 11 = ECZ2 from GenDSP	0
1-0	R/W	GATESRC1	00 = 93.8 Hz 01 = RINGING comparator 10 = ECZ1 from Gen2DSP 11 = ECZ2 from Gen2DSP	0

Note 72: CT1_INT and CT2_INT are ored to CT_CLASSD_INT

Table 333: TONE_COUNTER1_REG (0xFF4992)

Bit	Mode	Symbol	Description	Reset
15-0	R	TONE_COUNTER	The TONE_COUNTER counts continuously with a selectable clock source. Sources are selected with CLKSRC values in the TONE_CTRL1_REG. Periodically the counter is latched in TONE_LATCH1_REG	0

Table 334: TONE_LATCH1_REG (0xFF4994)

Bit	Mode	Symbol	Description	Reset
15-0	R	TONE_LATCH1	Contains the latched TONE_COUNTER1_REG value. Will be cleared when writing to TONE_CLEAR_INT1_REG	0

Table 335: TONE_CLEAR_INT1_REG (0xFF4996)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	TONE_CLEAR_INT1	Writing clears TONE_LATCH1_REG and bit CT1_INT. (Note: it takes maximum 7 us before register TONE_LATCH1_REG is cleared) Reading returns 0.	0

Table 336: TONE_CTRL2_REG (0xFF4998)

Bit	Mode	Symbol	Description	Reset
15-11	-	-	Reserved	0
10	R/W	GATE_EDGE2	0 = Rising edge of GATE_CLK used to clock TONE_TIMER 1 = Rising and falling edge of GATE_CLK used	0
9	R/W	MCT2_INT	0 = Disable CT2_INT to CT_CLASSD_INT 1 = Enable CT2_INT to CT_CLASSD_INT, (Note 72)	0
8	R	CT2_INT	1 = Tone latch 2 interrupt. Must be cleared by writing to TONE_CLEAR_INT2_REG	0
7-4	R/W	TIMER_RELOAD2	Select clock divider reload value for clock to latch timer periodically 0 = divide by 1, 15 = divide by 16	0
3-2	R/W	CLKSRC2	00 = 144 kHz 01 = reserved 10 = ECZ1 from Gen2DSP 11 = ECZ2 from Gen2DSP	0
1-0	R/W	GATESRC2	00 = 93.8 Hz 01 = RINGING comparator 10 = ECZ1 from Gen2DSP 11 = ECZ2 from Gen2DSP	0

Table 337: TONE_COUNTER2_REG (0xFF499A)

Bit	Mode	Symbol	Description	Reset
15-0	R	TONE_COUNTER2	The TONE_COUNTER2 counts continuously with a selectable clock source. Sources are selected with CLKSRC2 values in the TONE_CTRL2_REG. Periodically the counter is latched in TONE_LATCH2_REG	0

Table 338: TONE_LATCH2_REG (0xFF499C)

Bit	Mode	Symbol	Description	Reset
15-0	R	TONE_LATCH2	Contains the latched TONE_COUNTER2_REG value. Will be 0 when writing to TONE_LATCH2_REG.	0

Table 339: TONE_CLEAR_INT2_REG (0xFF499E)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	TONE_CLEAR_INT2	Writing clears TONE_LATCH2_REG and bit CT2_INT. (Note: it takes maximum 7 us before register TONE_LATCH2_REG is cleared) Reading returns 0.	0

Table 340: UART_CTRL_REG (0xFF4900)

Bit	Mode	Symbol	Description	Reset
15-11	-	-	Reserved	0
10	R/W	INV_UTX	0 = UTX is not inverted 1 = UTX is inverted	0
9	R/W	INV_URX	0 = URX is not inverted 1 = URX is inverted	0
8	R/W	IRDA_EN	0 = NRZ mode, normal UART mode 1 = RZI mode enabled for IrDA.	0
7	R/W	UART_MODE	0 = Baudrates according to bits 4,3,2 format: 8 bits no parity 1 = 10.125kbaud is selected. 8 bits even parity. (Note 74)	0
6	R	RI	If 1 UART receive interrupt. Must be cleared by SW by writing to UART_CLEAR_RX_INT_REG	0
5	R	TI	If 1 UART transmit interrupt. Must be cleared by SW by writing to UART_CLEAR_TX_INT_REG (Note 73)	0
4-2	R/W	BAUDRATE	UART baud rate selection: (Note 74) 000 = 9600 Baud 001 = 19200 Baud 010 = 57.6 kBaud 011 = 115.2 kBaud 100 = Fsys/(X*45) (230.4 kBaud @10.368 MHz0 (Note 75) 101 = reserved 11x = reserved.	000
1	R/W	UART_TEN	If 1 the UART transmitter is enabled.	0
0	R/W	UART_REN	If 1 the UART receiver is enabled.	0

Note 73: TI reset value is 0 and will be 1 as soon as UART_TEN is enabled.

Note 74: The indicated baudrates are valid with Fsys/CLK_PER_DIV_REG[PER_DIV]=1.152 MHz.

Note 75: If the DMA unit is used make sure that the SW_HCLK_DIV is set such that the system clock **Fsys** is higher than the baudrate.

Table 341: UART_CLEAR_TX_INT_REG (0xFF4904)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	CLEAR_TX	Writing any value to this register will clear the UART TI interrupt Reading returns 0.	0

Table 342: UART_CLEAR_RX_INT_REG (0xFF4906)

Bit	Mode	Symbol	Description	Reset
15-0	R/W	CLEAR_RX	Writing any value to this register will clear the UART RI interrupt Reading returns 0.	0

Table 343: UART_RX_TX_REG (0xFF4902)

Bit	Mode	Symbol	Description	Reset
15-8	-	-	Reserved	0
7-0	R/W	UART_DATA	UART x input/output register	0

Note 76: Same address for two registers one being for data input and the other for data output.

Table 344: UART_ERROR_REG (0xFF4908)

Bit	Mode	Symbol	Description	Reset
15-2	-	-	Reserved	0
1	R/W	DMA_PARITY_ERROR	Is set by PAR_STATUS and will be reset by writing any value to UART_ERROR_REG	0
0	R	PAR_STATUS	0: no parity error, 1: parity error. Updated every new byte.	0

Table 345: WATCHDOG_REG (0xFF4C00)

Bit	Mode	Symbol	Description	Reset
15-8	-	-	Reserved	0
7-0	R/W	WDOG_VAL	Watchdog preset value. Decremented by 1 every 10.66 msec. If 0 a NMI interrupt or internal reset is generated, depending on the TIMER_CTRL_REG[WDOG_CTRL] and value 0xFF is automatically reloaded into the watchdog timer. See also Figure 12	0xFF

38.0 Specifications

All MIN/MAX specification limits are guaranteed by design, or production test, or statistical methods unless note 77 is added to the parameter description. Typical values are informative.

Note 77: This parameter will not be tested in production. The MIN/MAX values are guaranteed by design and verified by characterization.

Table 346: ABSOLUTE MAXIMUM RATINGS (Note 78)

PARAMETER	DESCRIPTION	CONDITIONS	MIN	MAX	UNITS
SUPPLY_max	Max voltage on SUPPLY, DC_SENSE, LDO1_CTRL, LDO2_CTRL, VDDPA, DC_I			5.5V	V
Vdd_max	Max Core supply voltage (VDD-VSS / AVD2-AVS / VDDRF-VSSRF)			2.0	V
Vddio_max	Max Digital I/O supply voltage VDDIO-VSS			3.6	V
Vesd_HBM	ESD voltage according to Human Body model			2000	V
Vesd_MM	ESD voltage according to Man Machine model			175	V

Note 78: Absolute maximum ratings are those values that may be applied for maximum 50 hours.

Beyond these values, damage to the device may occur.

Table 347: OPERATING CONDITIONS (Note 79)

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Vsupply	Supply voltages SUPPLY-AVS		2.0		5.0	V
Vpa	CLASSD Supply voltage VDDPA-VSSPA		2.0	2.4	4.5	V
Vdd	Supply voltages VDD-VSS, AVD2-AVS, VDDRF-VSSRF.	(Note 82)	1.75	1.8	1.98	V
Vddio	Digital I/O supply voltage VDDIO-VSS		1.75	2.5	3.45	V
Vpower	Voltages on DC_SENSE, LDO1_CTRL, LDO2_CTRL, DC_I				5	V
Vdig	Voltage on digital pins Max voltage is 3.6V				VDDIO +0.3	V
Vana	Voltage on analog pins and RSTn pin and XTAL input with XTAL supply = VDDRF				AVD+ 0.3	V
Vxtal	Voltage on XTAL input with XTAL supply = LDO_XTAL (1.6V)				1.9	V
Vpin_neg	Minimum voltage on any pin		VSSIO- 0.3			V
Iprot_mic	Current through protection diode MICp to internal AGND	(Note 83)			2.4	mA
Iprot_adc	Current through protection diode of ADC0, ADC1 inputs to internal AGND.	(Note 83)			1	mA
Iprot_charge	Current through protection diode of CHARGE pin				10	mA
IVddio	Current through one I/O pin				20	mA
Ipa	Current through VDDPA, PAOUTx, VSSPA. The current is CLASSD_CTRL_REG[CLASSD_VOUT]/2*Zload_pa_xx (see Table 377)	(Note 80)			750	mA

Table 347: OPERATING CONDITIONS (Note 79)

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Ppackage	Package power dissipation @ 25 °C				1	W
Tstorage	Storage temperature, at relative humidity of <60%. Refer to SiTel Portal "Floor and shelf life advisory.pdf"		+20	+25	+30	°C
TA	Ambient temperature	(Note 81)	-20		60	°C

Note 79: Within the specified limits, a life time of 10 years is guaranteed.

Note 80: A life time of 10 years of the CLASSD amplifier is guaranteed if switched on for 10% of the time.

Note 81: Within this temperature range full operation is guaranteed.

Note 82: Full operating mode; the differences between AVD2, VDD may never be more than 300mV; during a short period of time e.g. during power up more than 300mV difference is allowed. Analog performance is only guaranteed from 1.75-1.98V

Note 83: For pads with the protection circuit enabled, the protection current **must** be limited with external source resistor Rext if the input voltage exceeds 0.4V resp 1.4V. Rext can be calculated as follows: $R_{ext} > (V_{ext}-1.4)/I_{protxx}$.

38.1 ELECTRICAL CHARACTERISTICS

VDD, VDDRF, AVD2 = 1.8 Volt all signals are related to VSS. Typical values are at +25 °C. MAX and MIN values are over the full temperature range TA. Crystal frequency = 10.368 MHz unless specified otherwise.

Confidential

**Table 348: Supply currents active mode (values derived from characterization, 4 sigma extrapolated values)
(Note 77)**

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
lvdd_static	Static VDD current	XTAL = VSS, T=25degC		200		µA
lvdd_static_t		XTAL = VSS, T=TA				µA
lvdd_rcur_ram108p	VDD supply current when CR16C+ is executing “while (1) from shared RAM , Not including lvdd_static	CR16@10.368/(8*16) MHz, PLL off. DIP executing U_PSC, PAGON, WT FF,		1.54		mA
lvdd_rcur_ram108		CR16@10.368/8 MHz, PLL off		1.88		mA
lvdd_rcur_ram101		CR16@10.368 MHz, PLL off		4.19		mA
lvdd_rcur_ram_80_1p25		CR16@10.368/8 MHz, PLL@82.944 MHz		12.03		mA
lvdd_rcur_ram_80_20		CR16@20.736 MHz, PLL@82.944 MHz		17.21		mA
lvdd_rcur_ram_165_1p25		CR16@10.368/8 MHz, PLL@165.888 MHz		22.76		mA
lvdd_rcur_ram_165_10		CR16@10.368 MHz, PLL@165.888 MHz		25.20		mA
lvdd_rcur_ram_165_20		CR16@20.736 MHz, PLL@165.888 MHz		28.00		mA
lvdd_rcur_ram_165_40		CR16@41.472 MHz, PLL@165.888 MHz		33.39		mA
lvdd_rcur_ram_165_80		CR16@82.944 MHz, PLL@165.888 MHz		44.23		mA
lvdd_bmctx	Delta VDD supply current of BMC Tx	BMC all slots encryption on, 1152 Mbit/s (B_DIV1), RFCLK disabled.				µA
lvdd_dsp1min10	Delta VDD Single DSP supply current (PLL1 currents are included in lvdd_rcur_xx)	Gen2DSP1,2 = 10.368 MHz. DSP_EN=1 waiting for trigger.				µA
lvdd_dsp1max10		Gen2DSP1,2 = 10.368 MHz. 1296 cycles. 10/90% SCP RAM/ROM load				mA
lvdd_dsp1min80		GenDSP1,2 = 81.944 MHz. DSP_EN=1 waiting for trigger.				mA
lvdd_dsp1max80		GenDSP1 = 82.944MHz. 10368 cycles 10/90% SCP RAM/ROM load				mA
lvdd_dsp2max80		GenDSP2 = 82.944MHz. 10368 cycles 10/90% SCP RAM/ROM load				mA
lvdd_codec	Delta VDD Codec supply current	COFF=0, MIC_PD=0.				µA
lavd2_codec	Delta AVD2 Codec supply current	Reference amplifiers on, microphone amp on, LSRp/n on, CLASSD off, Codec on.				mA
lavd2_power	AVD2 Power circuit current	Includes AVD2 static current, LDO1, LDO2 (1.8V) on, DCDC converter				µA
lvddrf_xtal	VDDRF xtal oscillator current including static current	Crystal type TQ710718-15 (10.368MHz). Cc1 = Cc2 = 10 pF, LDO_XTAL_ON=1, XTAL_SUPPLY = 0		0.35	0.46	mA
lvdd_pll2	Delta VDD PLL2 supply current	Fxtal =10.368 MHz Fpll = 50MHz				µA

Table 348: Supply currents active mode (values derived from characterization, 4 sigma extrapolated values)
(Note 77)

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
lvddrf_static	VDDRF static supply current.	CLASSD, rssi ADC in powerdown				uA
lvddrf_gaus	Delta VDDRF Gaussian supply current.	TDO in gaussian mode (max output level) On: <B_RC>: RC5[M1-M0]=x1 Off: <B_RC>: RC5[M1-M0]=00 E.g If TDO must keep midlevel, this current is always active, else the on/off ratio must be calculated.		0.7		mA
lvddrf_rssi	Delta VDDRF rssi supply current.	Peak-hold rssi ADC On: Any <B_*> command Off: <B_RST>. E.g. For 1 full slot this current must be divided by 24.		0.15		mA

Table 349: Supply currents (Indicative value)

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
lvddrf_pa	CLASSD normal mode supply current at VDDRF	CLASSD_PD=0		3.5		mA
lvddrf_dig	CLASSD digital mode supply current at VDDRF	(P5_1_MODE = 01 or P5_2_MODE = 01) and CLASSD_PD=1.		0.9		mA
lvddpa_pa	CLASSD supply current at VDDPA	(P5_1_MODE = 01 or P5_2_MODE = 01) and CLASSD_PD=1 or CLASSD_PD=0. No load		1.5		mA

Note 84: The analog currents (lavd2_xxx, lvddrf_xx) are mainly the bias currents with bandgap current bits set to their default value 000.

The bias currents can be trimmed by modifying e.g the LED current to the typical value in production with bits BANDGAP_REG[8-6].

Note 85: Total chip current active mode = (lvdd_bmcrx*rx_slot_on_ratio+ lvdd_bmxtx*tx_slot_on_ratio + (lvdd_rcur_ramxxx or (lvddio_rcur_extxxx (1 + (V-2.65V))*frequency_ratio +(lvdd_dspmin +(lvdd_DSPmax-lvdd_dspmin)*number_of_cycles/5184)) + lvddrf_pll*on + lvddrf_xtal + lvddrf_pa + lvddrf_gaus*tx_on_ratio + lvddrf_rssi*rx_on_ratio + lavd2_codec*on + lavd2_cid*on + lavd2_dc*on + lavd2_power + lvdd_static.

Changing values due to different frequencies:

- All AC currents: new_frequency/10.368

Additional currents due to external components **must be added** to the total chip current : E.g RFCLK pin switching currents (supplied from VDDIO), current through Loudspeaker and microphone + current through DC_CTRL due to the gate source capacitor of the switching transistor + (Add I = 1.8V * dc*dc_frequency * Cgs) + current through IO ports + p2[6] /stop charge

Note 86: Total chip current paging mode = lvdd_bmc/(16*24) + (lvdd_rcur_ramxxxp or lvdd_rcur_romxxxp or (lvdd_rcur_flashxxxp + lvddio_rcur_flashxxxp))*frequency_ratio + lvddrf_rssi* rx_on_ratio + lvddrf_xtal + lvddrf_pll*on + lavd2_dc*dc+ lavd2_power +lvdd_static

Changing values due to different frequencies:

- All AC currents: new_frequency/10.368

Additional currents and DCDC converter efficiency same as active mode.

Table 350: DCDC converter (buck-mode)

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
LR_dc/dc_buck	load regulation			0.012		%/mA
Psrr_dc/dc_buck	line regulation			1.2		%/V
Vbuck_ref0	Trimmed (+/-2%) internal reference voltages for DCDC converter	DC_VOUT = '000'		1.29		V
Vbuck_ref1	Output voltage drops with line regulation ratio.	DC_VOUT = '001'	2.10	2.20	2.30	V
Vbuck_ref2	Unloaded, the output voltage will be 5% higher due to minimum output current in combination with build-in output overvoltage protection.	DC_VOUT = '010'	2.40	2.50	2.60	V
Vbuck_ref3	(Note 87) (Note 77)	DC_VOUT = '011'	2.60	2.70	2.80	V
Vbuck_ref4		DC_VOUT = '100'	2.90	3.00	3.10	V
Vbuck_ref5		DC_VOUT = '101'	3.15	3.27	3.40	V
Vbuck_ref6		DC_VOUT = '110'	3.45	3.60	3.75	V
Vbuck_ref7		DC_VOUT = '111'	1.90	2.00	2.10	V

Note 87: Careful design is needed around the DC/DC converter to guarantee stable behaviour

Table 351: SUPPLY ON level

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Vsupply_on	SUPPLY Switch on level	Untrimmed BAND_GAP = 0x8		1.8		V
Vsupply_hyst	Vsupply_on- Vsupply_off			360		mV

Table 352: Reset circuit

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Vrst_on	Out of reset on-level	BANDGAP_REG = 8 (Note 88)	1.7		1.9	V
Vrst_hyst	Vrst_on-Vrst_off			160		mV

Note 88: The device leaves the reset state if either the AVD2 > Vrst_on or the LDO2 voltage regulator has stabilized. This guarantees that the device always goes out of reset even with the smallest offset in the reset comparator. As soon as the reset circuit is in the out of reset on- state, the reset level is automatically switched to the off value Vrst_off.

Table 353: Digital and analog switching inputs

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Vil_dig	Logic 0 input level all digital pads	VDDIO = 1.75-3V			0.3*VDDIO	V
Vil_dig_33	Logic 0 input level all digital pads	VDDIO = 3-3.45V			0.90	V
Vil_xtal	Logic 0 input level XTAL pin				0.50	V
Vih_dig	Logic 1 input level all digital pads	VDDIO=1.75-3.45V	0.7*VDDIO			V
Vih_xtal	Logic 1 input level XTAL pin with external clock.		1.20			V
Ileak_hi	Input current of all inputs with (programmable) pull-down resistors disabled, except ADC inputs (see ADC)	Vin=VDDIO = 3.45V			10	µA

Table 353: Digital and analog switching inputs

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Ileak_lo	Input current of all inputs with (programmable) pull-up resistors disabled, except ADC inputs (see ADC) and AD[3:1].	Vin = VSS			10	µA
Ipull_up_lo	Input current with internal pull up enabled.	Vin = VSS VDDIO=1.75-3.45V	60		115	µA
Ipull_down_hi	Input current with internal pull down enabled	Vin=VDDIO VDDIO=1.75-3.45V	60		115	µA
Ipu_RST_lo	Input current of RSTn from VDD through pull-up resistor.	Vin = AVS, VDD=1.8V	4		20	µA

Table 354: Digital outputs (except DC_CTRL, CLASSDp, CLASSDm digital mode, and pins with programmable drive strength (Table 355))

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Vol_100u	Logic 0 output level	Iout = 100 µA. VDDIO =3.45V			0.1	V
Vol_175	Logic 0 output level	Iout = 8 mA, VDDIO = 1.75V-1.98V			0.2*VDDIO	V
Vol_25	Logic 0 output level	Iout = 12 mA, VDDIO = 2.3V-2.75V			0.2*VDDIO	V
Vol_33	Logic 0 output level	Iout = 16mA, VDDIO = 3-3.45V			0.2*VDDIO	V
Vol_rfclk	Logic 0 output level RFCLK pin	Iout = 8 mA sink current, VDDIO_RFCLK =1.6, 2.3V			0.2* VDDIO_RF CLK	V
Voh_100u	Logic 1 output level	Iout = 100 µA	VDDIO - 0.1			V
Voh_175	Logic 1 output level	Iout = 8 mA, VDDIO = 1.75V-1.98V	0.8*VDDIO			V
Voh_25	Logic 1 output level	Iout = 12 mA, VDDIO = 2.3V-2.75V	0.8*VDDIO			V
Voh_33	Logic 1 output level	Iout = 16 mA, VDDIO = 3-3.45V	0.8*VDDIO			V
Voh_rfclk	Logic 1 output level RFCLK pin	Iout = 8 mA, VDDIO_RFCLK =1.6, 2.3V Source current buffered by external 1 uF capacitor	0.8* VDDIO_RF CLK			V

Note 89: For open collector configurations, only V_{OL} is applicable.

Note 90: Output must stay below V_{il} or above V_{ih} to avoid switching cross currents in the input stage.

Table 355: Digital outputs with controlable drive strengths (Note 77)

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Vol_18_2m	Logic 0 output level (Note 91)	Iout = 2mA, VDDIO = 1.75V, PINDRV_?[1-0] = 11b			0.2* VDDIO	V
Voh_18_2m	Logic 1 output level (Note 91)	Iout = 2mA, VDDIO = 1.75 V, PINDRV_?[1-0] = 11b	0.8* VDDIO			V
Vol_33_4m	Logic 0 output level (Note 91)	Iout = 4mA, VDDIO = 3.00 V, PINDRV_?[1-0] = 11b			0.2* VDDIO	V
Voh_33_4m	Logic 1 output level (Note 91)	Iout = 4mA, VDDIO = 3.00 V, PINDRV_?[1-0] = 11b	0.8* VDDIO			V
Vol_18_4m	Logic 0 output level (Note 91)	Iout = 4mA, VDDIO = 1.75 V, PINDRV_?[1-0] = 10b			0.2* VDDIO	V
Voh_18_4m	Logic 1 output level (Note 91)	Iout = 4mA, VDDIO = 1.75 V, PINDRV_?[1-0] = 10b	0.8* VDDIO			V
Vol_33_8m	Logic 0 output level (Note 91)	Iout = 8mA, VDDIO = 3.00 V, PINDRV_?[1-0] = 10b			0.2* VDDIO	V
Voh_33_8m	Logic 1 output level (Note 91)	Iout = 8mA, VDDIO = 3.00 V, PINDRV_?[1-0] = 10b	0.8* VDDIO			V
Vol_18_6m	Logic 0 output level (Note 91)	Iout = 6mA, VDDIO = 1.75 V, PINDRV_?[1-0] = 10b			0.2* VDDIO	V
Voh_18_6m	Logic 1 output level (Note 91)	Iout = 6mA, VDDIO = 1.75 V, PINDRV_?[1-0] = 01b	0.8* VDDIO			V
Vol_33_12m	Logic 0 output level (Note 91)	Iout = 12mA, VDDIO = 3.00 V, PINDRV_?[1-0] = 10b			0.2* VDDIO	V
Voh_33_12m	Logic 1 output level (Note 91)	Iout = 12 mA, VDDIO = 3.00 V, PINDRV_?[1-0] = 01b	0.8* VDDIO			V
Vol_18_8m	Logic 0 output level	Iout = 8mA, VDDIO = 1.75 V, PINDRV_?[1-0] = 00b			0.2* VDDIO	V
Voh_18_8m	Logic 1 output level	Iout = 8mA, VDDIO = 1.75 V, PINDRV_?[1-0] = 00b	0.8* VDDIO			V
Vol_33_16m	Logic 0 output level	Iout = 16mA, VDDIO = 3.00 V, PINDRV_?[1-0] = 00b			0.2* VDDIO	V
Voh_33_16m	Logic 1 output level	Iout = 16 mA, VDDIO = 3.00 V, PINDRV_?[1-0] = 00b	0.8* VDDIO			V

Note 91: Levels and Pad drive switching are tested during characterization.

Table 356: Xtal trimming capacitors

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Cmax_xtal	Internal trimming capacitor C1+C2	CLK_FREQ_TRIM_REG= "FF"	15	21.5	28	pF

Table 357: Temperature sensor

PARAMETER	DESCRIPTION	CONDITION	MIN	TYP	MAX	UNITS
Ttemp_sense	Temperature range inside the IC		5		55	deg. C
Rtemp_sense	Sensor resolution within temperature range including ADC tolerances		-	0.125	-	deg / bit
ADCtemp_sense	Decimal value of ADCout. K is temperature in Kelvin (0K = -273 deg C).		2960-9.02*K	2960-8*K	2960-6.97*K	

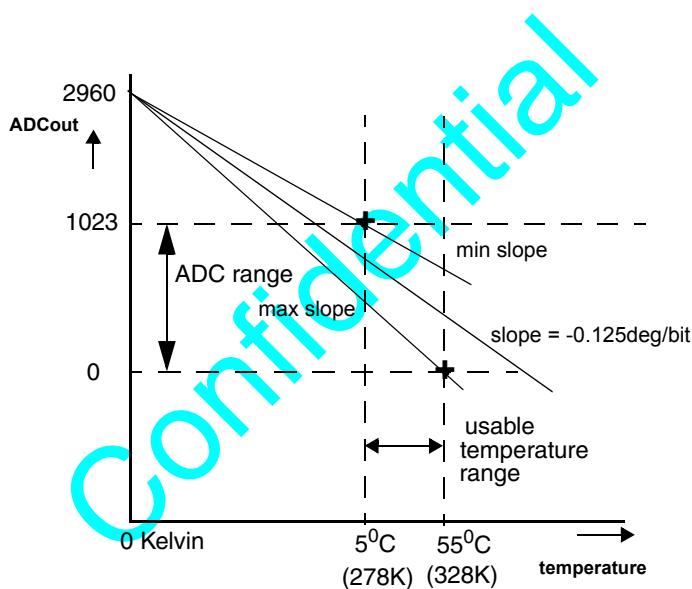


Figure 102 Temperature sensor accuracy and calibration references

Table 358: General purpose ADC. Inputs ADC0, ADC1, ADC2, ADC3 SUPPLY (Operating conditions)

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Nadc_bits	ADC Resolution			10		bits
Tadc_conv	ADC Conversion time			55		us
Vadc_range	ADC0,1,2,3 linear input range with protection disabled.	ADCx_PR_DIS=1 (Note 94)	0		AVD_XTAL	V
Vadc_range_adc3	ADC3 linear input range		0		1.5	V

Table 358: General purpose ADC. Inputs ADC0, ADC1, ADC2, ADC3 SUPPLY (Operating conditions)

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Vadc_range_p	ADC0,1 linear input range with protection enabled.	ADCx_PR_DIS=0 (Note 95)(Note 83)	0		0.9	V
Vadc_data0_r	Internal ADC DAC accuracy	ADC_DAC = 0 ADC_VREF = 0		0		V
Vadc_data1023_r		ADC_DAC = 0xFF ADC_VREF = 0		AVD_XTAL		V
Rsrc_adc	Output resistance of external circuit connected to ADC input				25	kΩ

Table 359: ADC0, ADC1, ADC2, ADC3, CIDOUT inputs (Note 92)

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Ladc_lsb_dnl	Differential non linearity (Note 93) (Note 77)		-0.5		+0.5	LSB
Ladc_lsb_inl	Integral non linearity (Note 93) (Note 77)		-8		+8	LSB
Ileak_adc0_lo	ADC0 Input leakage with pad protection disabled (Note 77)	ADC0 selected ADC0_PR_DIS=1 Vin = AVS	-0.90		0.90	μA
Ileak_adc0_hi	ADC0 Input leakage with pad protection disabled (Note 77)	ADC0 selected ADC0_PR_DIS=1 Vin = AVD	-0.90		0.90	μA
Ileak_adc0_pn	ADC0 Input leakage with pad protection enabled (Note 77)	ADC0 selected ADC0_PR_DIS=0 Vin = AVD/2	-0.90		0.90	μA
Ileak_adc1_lo	ADC1 Input leakage with pad protection disabled (Note 77)	ADC1 selected ADC1_PR_DIS=1 Vin = AVS	-0.90		0.90	μA
Ileak_adc1_hi	ADC1 Input leakage with pad protection disabled (Note 77)	ADC1 selected ADC1_PR_DIS=1 Vin = AVD	-0.90		0.90	μA
Ileak_adc1_pn	ADC1 Input leakage with pad protection enabled (Note 77)	ADC1 selected ADC1_PR_DIS=0 Vin = AVD/2	-0.90		0.90	μA
Ileak_adc2_lo	ADC2 Input leakage with pad protection disabled (Note 77)	ADC2 selected Vin = AVS	-0.90		0.90	μA
Ileak_adc2_hi	ADC2 Input leakage with pad protection disabled (Note 77)	ADC2 selected ADC2_PR_DIS=1 Vin = AVD	-0.90		0.90	μA
Ileak_adc2_pn	ADC2 Input leakage with pad protection enabled (Note 77)	ADC2 selected ADC2_PR_DIS=0 Vin = AVD/2	-0.90		0.90	μA
Ileak_adc3_lo	ADC3 Input leakage (Note 77)	ADC3 selected Vin = AVS	-0.90		0.90	μA
Ileak_adc3_hi	ADC3 Input leakage (Note 77)	ADC3 selected Vin = AVD	-0.90		0.90	μA
Ileak_cidout	Unbuffered ADC input via CIDOUT	CID selected Vin = AVD	-0.5		0.5	μA

Note 92: ADC specifications are valid for BANDGAP_REG trimmed such that AVD2=1.800V.

Note 93: 1 LSB equals to AVD_XTAL/1024.

Differential non-linearity (DNL) error is defined as the difference between an actual step width and the ideal value of 1 LSB
 Integral non-linearity (INL) is defined as being the maximum deviation from the ideal curve.

Note 94: ADC input may never be above the AVD2 supply voltage, since it will influence the ADC value on the other inputs.

Note 95: If the input protections are enabled the ADCs inputs have 9 bits linear range for Vin < internal AGND (0.9V)

Note 96: The switch in the path from ADC3 to ADC may cause limiting above 1.5V. Input voltages up to AVD2 are allowed however

Table 360: SUPPLY Scaler circuit (Note 92)

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Vsupply_scaler	Scaler output voltage to ADC	0 < SUPPLY < 5.11 V		0.313xSUPPLY		V

Table 361: LDO1 (with external transistor BC807-40 or equivalent)

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Vldo1_18	Trimmed (+/-2%) output voltage (Note 77)	LDO1_LEVEL = '000' AVD2 trimmed to 1.8V	1.75	1.8	1.84	V
Vldo1_20		LDO1_LEVEL = '001' AVD2 trimmed to 1.8V	1.95	2	2.04	V
Vldo1_25		LDO1_LEVEL = '010' AVD2 trimmed to 1.8V	2.44	2.5	2.55	V
Vldo1_30		LDO1_LEVEL = '011' AVD2 trimmed to 1.8V	2.93	3.0	3.06	V
Vldo1_33		LDO1_LEVEL = '100' AVD2 trimmed to 1.8V	3.19	3.27	3.34	V
Vldo1_18_8	Untrimmed (+/-7%) output voltage (Note 77)	LDO1_LEVEL = '000' BANDGAP_REG= xx8	1.67	1.8	1.93	V
Vldo1_20_8		LDO1_LEVEL = '001' BANDGAP_REG= xx8	1.86	2	2.14	V
Vldo1_25_8		LDO1_LEVEL = '010' BANDGAP_REG= xx8	2.32	2.5	2.68	V
Vldo1_30_8		LDO1_LEVEL = '011' BANDGAP_REG= xx8	2.79	3	3.21	V
Vldo1_33_8		LDO1_LEVEL = '100' BANDGAP_REG= xx8	3.04	3.27	3.50	V
psrr_ldo1	Line regulation	Dropout voltage > 200mV 1mA < Iload < 50mA		0.05		%/V
LR_ldo1	load regulation	Dropout voltage > 200mV, 1mA < Iload < 50mA		0.0005		%/mA
Imax_ldo1_ctrl	Maximum sink current in LDO1_CTRL pin (for maximum LDO1 output current multiply this value by current gain of external transistor)		4	7.5		mA
Cout_ldo1	Output capacitor (ceramic)		1.0			μF

Table 362: LDO2 (with external transistor BC807-40 or equivalent)

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Vldo2	Trimmed (+/-1%) output voltage (Note 77)	AVD2 trimmed to 1.8V	1.76	1.8	1.84	V
Vldo2_8	Untrimmed (+/-6%) output voltage (Note 77)	BANDGAP_REG= xx8	1.7	1.8	1.9	V
psrr_ldo2	Line regulation	Dropout voltage > 200mV, 1mA < Iload < 50mA		0.05		%/V

Table 362: LDO2 (with external transistor BC807-40 or equivalent)

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
LR_Ido2	load regulation	Dropout voltage > 200mV, 1mA < Iload < 50mA		0.0005		%/mA
Imax_Ido2_ctrl	Maximum sink current in LDO2_CTRL pin (for maximum LDO2 output current multiply this value by current gain of external transistor)		4	7.5		mA
Cout_Ido2	Output capacitor (ceramic)		1.0			μF

Table 363: LDO_XTAL

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Vldoxtal_out	Trimmed (+/-2%) output voltage (Note 77)	unloaded, AVD2 trimmed to 1.8V	1.56	1.6	1.64	V
Vldoxtal_out_8	Untrimmed (+/-7%) Output volt- age	unloaded, BANDGAP_REG= xx8	1.5	1.6	1.7	V
Vldoxtal_do	Dropout voltage	Iload = 10mA			200	mV
psrr_Idoxtal	Line regulation			0.12		%/V
LR_Idoxtal	load regulation	1mA < Iload < 10mA		0.07		%/mA
Imax_Idoxtal	Maximum output current		10			mA
Cout_Idoxtal	Output capacitor (ceramic)		1.0			μF

Table 364: LDO_RFCLK

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Vldorfclk_do	Dropout voltage	Iload = 2mA			200	mV
psrr_Idorfclk	Line regulation			0.05		%/V
LR_Idorfclk	load regulation	1mA < Iload < 2mA		0.03		%/mA
Imax_Idorfclk	Maximum output current		2			mA
Cout_Idorfclk	Output capacitor (ceramic)		1.0			μF

Table 365: FAD

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Vfad_offset	Detectable FAD Comparator input voltage difference including comparator offset (Note 77)	DC offset at RSSI pin. is 1V.	-40		40	mV

Table 366: Peak Hold ADC

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Nrss	RSSI ADC Resolution			6		bits
Trssi_conv	RSSI Conversion time	34 clock cycles of 1.152 (1.536) MHz		30		μs

Table 367: Peak Hold ADC (continued)

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Vrsssi_data0	Low level on RSSI pin	ADC[5:0]=00(hex) RSSI_RANGE=0	0.07	0.09		V
Vrsssi_dac_data63	High level on RSSI pin	ADC[5:0]=3F(hex) RSSI_RANGE=0		1.07	1.16	V
Vrsssi_data0_f	Low level on RSSI pin	ADC[5:0]=00(hex) RSSI_RANGE=1	0			V
Vrsssi_dac_data63_f	High level on RSSI pin.	ADC[5:0]=3F(hex) RSSI_RANGE=1 Higher voltage gives 3F value. (Input may be up-to 1.8V)			1.6	V
Lrsssi_lsb_dnl	Differential non linearity (Note 77)		-0.5		+0.5	LSB
Lrsssi_lsb_inl	Integral non linearity (Note 77)		-1		+1	LSB
Rrsssi_sw_open	Input impedance	PD0 = 0	1			MΩ
Rrsssi_sw_closed	Input impedance	PD0 = 1			5	kΩ

Note 97: The Peak hold RSSI ADC is supplied from AVD_XTAL, the indicated voltages are measured with AVD_XTAL=1.6V. Drop out voltages and LDO accuracy must be taken into account.

Table 368: TDO pin, Gaussian mode M[1:0] = '01' BT = 0.5

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Vgaus_amp	Peak to Peak output level (Note 98)	At 0dB level (VOL[5:0]=0x15), 0101010....-pattern	0.5	0.59	0.7	V
Vgaus_amp_max	Peak to Peak maximum output level	at 0dB level (VOL[5:0]=0x15), 111000111000....-pattern	0.6	0.67	0.75	V
Vgaus_mid_ut	Gaussian Mid level	BMC_CTRL_REG[GAUSS_REF]=0 BANDGAP_REG = 8 (10 % accuracy)	0.8	0.9	1.0	V
Vgaus_mid_t	Gaussian Mid level (Note 77)	BMC_CTRL_REG[GAUSS_REF]=0 AVD2 trimmed to 1.8V with BANDGAP_REG (3 % accuracy)	0.85	0.9	0.95	V
S_gaus_d2	SINAD. 2nd and 3rd harmonic relative to 1rst harmonic. Typical values are the ideal gaussian curve.	f = 600 kHz sinus, Vout=1.34Vpp Bandwidth = 1.2 MHz 101010.... pattern Vout = 1.34 Vpp, f = 1152 kHz			-36	dB
S_gaus_d3		f = 600 kHz sinus, Vout=1.34Vpp Bandwidth = 1.2 MHz 101010.... pattern Vout = 1.34 Vpp, f = 1728 kHz		-32		dB

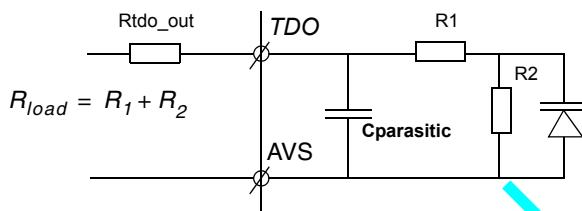
Note 98: V_{out} is independent of supply voltage

Table 369: TDO pin, Gaussian mode M[1:0] = '11' BT = 1

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Vgaus1_amp	Peak to Peak output level	at 0dB level (VOL[5:0]=0x15)	0.4	0.5	0.6	V

Table 370: TDO pin, Gaussian mode BT = 0.5 and BT = 1 load circuit

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Vgaus_adj	TDO Output level adjustment (Note 77)	6 bits tuning	-6		+6	dB
Rload	Load resistance	(see Figure 103) (R_1+R_2)	5			kΩ
Cparasitic	Load capacitance				20	pF
Rtdo_out	Output impedance				50	Ω


Figure 103 Typical application of TDO output in gaussian mode

38.2 ANALOG FRONTEND SPECIFICATIONS

Table 371: Microphone amplifier

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Vmic_0dB_unt	Untrimmed differential RMS input voltage between MICp and MICn (0dBm0 reference level) (Note 77)	0dBm0 on COUT (Note 99) MIC_GAIN[3:0] = 0, @ 1020 Hz; Tolerance: • 13% when untrimmed (BANDGAP_REG=8) • 6% when trimmed (Note 100)	114	131	149	mV
Amic_gain_0	Reference level at Vmic_0dB_unt	MIC_GAIN[3:0] = 0000		0		dB
Amic_gain_1	Microphone gain relative to Amic_gain_0	MIC_GAIN[3:0] = 0001		1.8		dB
Amic_gain_2		MIC_GAIN[3:0] = 0010		4.1		dB
Amic_gain_3		MIC_GAIN[3:0] = 0011		6.0		dB
Amic_gain_4		MIC_GAIN[3:0] = 0100		7.8		dB
Amic_gain_5		MIC_GAIN[3:0] = 0101		10.1		dB
Amic_gain_6		MIC_GAIN[3:0] = 0110		12.0		dB
Amic_gain_7		MIC_GAIN[3:0] = 0111		14.5		dB
Amic_gain_8		MIC_GAIN[3:0] = 1000		16.1		dB
Amic_gain_9		MIC_GAIN[3:0] = 1001		18.1		dB
Amic_gain_A		MIC_GAIN[3:0] = 1010		20.6		dB
Amic_gain_B		MIC_GAIN[3:0] = 1011		22.0		dB
Amic_gain_C		MIC_GAIN[3:0] = 1100		24.0		dB
Amic_gain_D		MIC_GAIN[3:0] = 1101		26.4		dB
Amic_gain_E		MIC_GAIN[3:0] = 1110		27.9		dB
Amic_gain_F		MIC_GAIN[3:0] = 1111		29.8		dB
Rin_mic	Resistance of activated microphone amplifier inputs (MICp, MICn and MICh) to internal AGND (Note 77)		75	150		kΩ
Vmic_offset	Input referred DC-offset (Note 77)	MIC_GAIN[3..0] = 1111 3 sigma deviation limits	-2.6		+2.6	mV

Note 99: 0 dBm0 on COUT = -3.14 dB of max PCM value. COUT is CODEC output in test mode

Note 100: Trimming possibility is foreseen. At system production the bandgap reference voltage can be controlled within 2% accuracy and data can be stored in Flash. Either AVD2 or VREF can be trimmed within 2% accuracy. If AVD2 is trimmed VREF will be within 2% accuracy related to either AVD2. Or vice versa VREF can be trimmed. For Vref trimming measure Δ (VREFp VREFm) and update BANDGAP_REG[3..0]

Table 372: Microphone amplifier (Operating Condition)

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Vmic_cm_level	MICp and MICn common mode voltage	MICp and MICn are set to AGND with internal resistors (Rin_mic). If DC coupled the input voltage must be equal to this voltage.		(0.9V/1.5)* VREFp		V

Table 373: Microphone supply voltages

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Vref_unt	VREFp-VREFm untrimmed	$I_{LOAD} = 0 \text{ mA}$ BANDGAP_REG = 8 (Note 100)	1.41	1.5	1.59	V
Rout_vrefp	VREFp output resistance	Figure 104		1		Ohm
Nvrefp_idle	Peak noise on VREFp-VREFm (Note 77)	CCITT weighted			-120	dBV
PSRRvrefp	Power supply rejection Vref output (Note 77)	See Figure 104, AVD2 to VREFp/m, f = 100 Hz to 4 kHz BANDGAP_REG[5:4] = 3	40			dB

Note 101: Vrefm is a clean ground input and is the 0V reference.

Table 374: VREFp load circuits

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Cload_vrefp	VREFp (parasitic) load capacitance				20	pF
Iout_vrefp	VREFp output current				1	mA

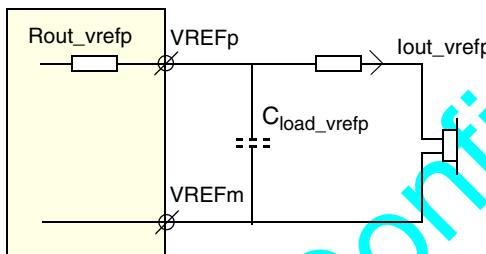


Figure 104 VREFp load circuit

Table 375: LSRp/LSRn outputs

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Vlsr_0dB_unt	Untrimmed differential RMS output voltage between LSRp and LSRn in audio mode (0dBm0 reference level)	0dBm0 on CIN (Note 102), LSRATT[2:0] = 001, @ 1020 Hz Load circuit A (see Figure 105, Table 376) with RL1= inf ohm, Cp1 or load circuit B (see Figure 106) with RL2, Cp2 and Cs2 <u>Tolerance:</u> <ul style="list-style-type: none"> • 13% when untrimmed (BANDGAP_REG=8) • 6% when trimmed (Note 100) 	621	714	807	mV
Rout_lsr	Resistance of activated loudspeaker amplifier outputs LSRp and LSRn			1		Ω

Table 375: LSRp/LSRn outputs

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Alsr_gain_0	Loudspeaker gain relative to Vlsr_0dB_unt	LSRATT[2:0] =000		2.3		dB
Alsr_gain_1		LSRATT[2:0] =001		0		dB
Alsr_gain_2		LSRATT[2:0] =010		-2.2		dB
Alsr_gain_3		LSRATT[2:0] =011		-4.0		dB
Alsr_gain_4		LSRATT[2:0] =100		-5.7		dB
Alsr_gain_5		LSRATT[2:0] =101		-8.0		dB
Alsr_gain_6		LSRATT[2:0] =110		-9.9		dB
Alsr_gain_7		LSRATT[2:0] =111		-12.1		dB
Vlsr_dc	DC offset between LSRp and LSRn (Note 77)	LSRATT[2:0] = 3 $R_{L1} = 28 \Omega$ 3 sigma deviation limits	-20		20	mV

Note 102:0 dBm0 on CIN = -3.14 dB of max PCM value

Table 376: LSRp/LSRn load circuits

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Cp1_RI1_inf	Load capacitance	see Figure 105, $R_{L1} = \infty$			30	pF
Cp1_RI1_1k	Load capacitance	see Figure 105, $R_{L1} \leq 1 \text{ k}\Omega$			100	pF
RI1	Load resistance		28			Ω
Cp2	Parallel load capacitance	see Figure 106			30	pF
Cs2	Serial load capacitance				30	μF
RI2	Load resistance		600			Ω

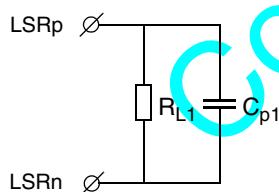
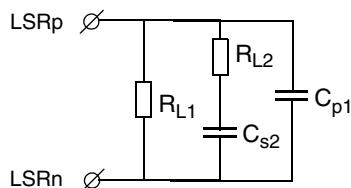

Figure 105 Load circuit A Dynamic loudspeaker

Figure 106 Load circuit B Piezo loudspeaker

Table 377: PAOUTp, PAOUTm outputs

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Vpa_00	Differential rms output voltage between PAOUTp and PAOUTm	Trimmed bandgap input = 0dBm0, 1kHz (Note 99) Output low-pass filtered CLASSD_VOUT = 00		1.23		Vrms
Vpa_01		As above, CLASSD_VOUT = 01		1.48		Vrms
Vpa_10		As above, CLASSD_VOUT = 10		1.81		Vrms
Vpa_11		As above, CLASSD_VOUT = 11		2.46		Vrms
Zload_pa_3v5	Speaker impedance, connected between PAOUTp and PAOUTm With these values, the peak currents stay below Ipa (see Table 347)	CLASSD_VOUT = 00	4			Ω
Zload_pa_5v		CLASSD_VOUT = 01	4			Ω
Zload_pa_7v		CLASSD_VOUT = 10	6			Ω
Zload_pa_10v		CLASSD_VOUT = 11	8			Ω

Confidential

Table 378: PAOUTp, PAOUTm outputs (Note 103)

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Rout_pa	Differential output resistance between PAOUTp and PAOUTm	See remark in (Note 103)		1		Ω

Note 103: Clipping of the outputs occurs when the VDDPA drops and the following conditions becomes true. If CLASSD_CTRL_REG[CLASSD_CLIP] is unequal to zero then upon a programmable number of clipping occurrences a CLASSD_INT is generated:
The software can stop clipping by reducing the gain by the GENDSP:

$$\text{Clipping occurs if } \frac{\text{peak}(\text{LowPassFiltered}(\text{PAOUTp} - \text{PAOUTm}))}{\text{VDDPA} - \text{VSSPA}} > \frac{Z_{load}}{Z_{load} + R_{out_pa}}$$

Table 379: CLASSD THD noise

PARAMETER	DESCRIPTION	CONDITIONS	VALUES
classd_THDN	Typical THD + noise versus output power (Note 77)	VDDPA-VSSPA = 2.5V CLASSD_CTRL_REG = 0C.00 Frequency = 1kHz, Output signal 22kHz filtered See Figure 108 for measurement setup	(see Figure 107)

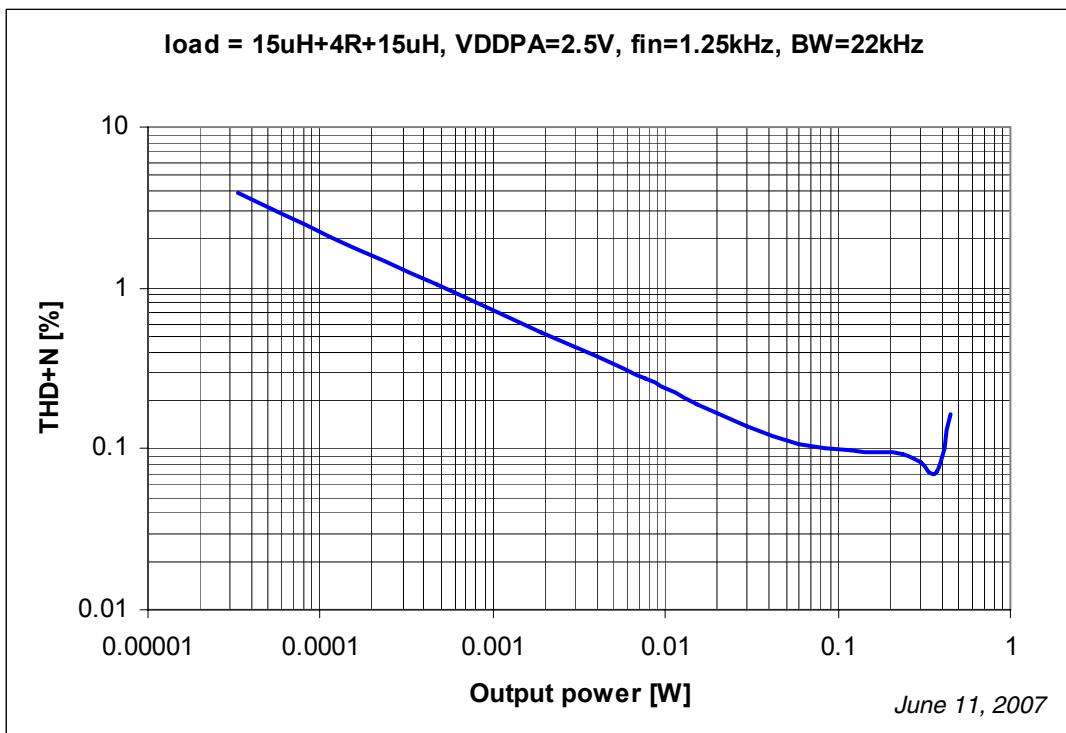

Figure 107 Class-D THD+N (total harmonic distortion + noise) versus output power

Table 380: CLASS-D output noise

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
N_classd_idle	CLASS-D amplifier Output noise. (Note 77)	CLASSD_CTRL_REG = 0x0308 CLASSD_BUZZER_REG = 0x0010 CODEC_ADDA_REG = 0x0002 DSP_CTRL_REG = 0xCF0E See Figure 108 for measurement setup		0.13	1.0	mV

Confidential

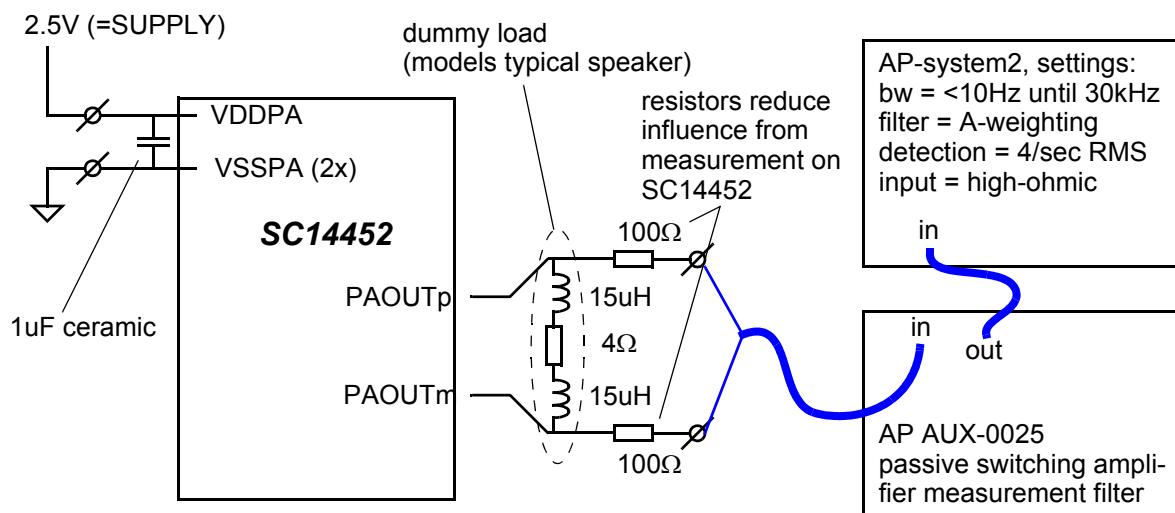


Figure 108 CLASSD amplifier measurement setup

38.3 CODEC SPECIFICATIONS

Table 381: CODEC characteristics

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
SDad_0dB	Signal to total distortion ratio Analog/Digital	differential input signal between MICp and MICn with f=(1020/8000)* Fs Hz ADPCM transcoder active MIC_GAIN[3:0] = 0. 0 dBm0 on COUT	40	60		dB
SDad_40dB		-40 dBm0 on COUT	35			dB
SDad_45dB		-45 dBm0 on COUT	30			dB
SDda_0dB	Signal to total distortion ratio Digital/Analog	Differential input signal between MICp and MICn with f= (1020/8000)* Fs Hz. ADPCM transcoder active LSRATT[2:0] = 3. 0 dBm0 on CIN	40	65		dB
SDda_40dB		-40 dBm0 on CIN	35			dB
SDda_45dB		-40 dBm0 on CIN	30			dB
Nad_idle	Idle channel noise Analog/ Digital	MICp short circuit to MICn Relative to 0 dBm0, CCITT weighted (Fs = 8kHz) MIC_GAIN[3:0] = 0x0B		-80		dBm0
Nda_idle	Idle channel noise Digital/ Analog	Relative to 0 dBm0, CCITT weighted (Fs = 8kHz) LSRATT[2:0] = 3		-83	-77	dBm0
PSRRad_0	Power supply rejection ratio Analog/Digital	AVD2 to COUT, f = 100 Hz to (Fs/2) kHz MIC_GAIN[3:0] = 0	40			dB
PSRRad_F		MIC_GAIN[3:0] = 0x0F	30			dB
PSRRda	Power supply rejection ratio Digital/Analog	AVD2 to LSRp/n, f = 100 Hz to (Fs/2) kHz LSRATT[2:0] = 3	40			dB

Table 382: CODEC frequency responses

PARAMETER	DESCRIPTION	CONDITIONS	RESULT
Fad_freq	Frequency response Analog/ Digital	Relative to (1020/8000)*Fs Hz	see frequency response diagrams Figure 109 on page 307, Figure 110 on page 307, Figure 111 on page 308
Fda_freq	Frequency response Digital/Analog	Relative to (1020/8000)*Fs Hz	see frequency response diagrams Figure 112 on page 308, Figure 113 on page 309, Figure 114 on page 309

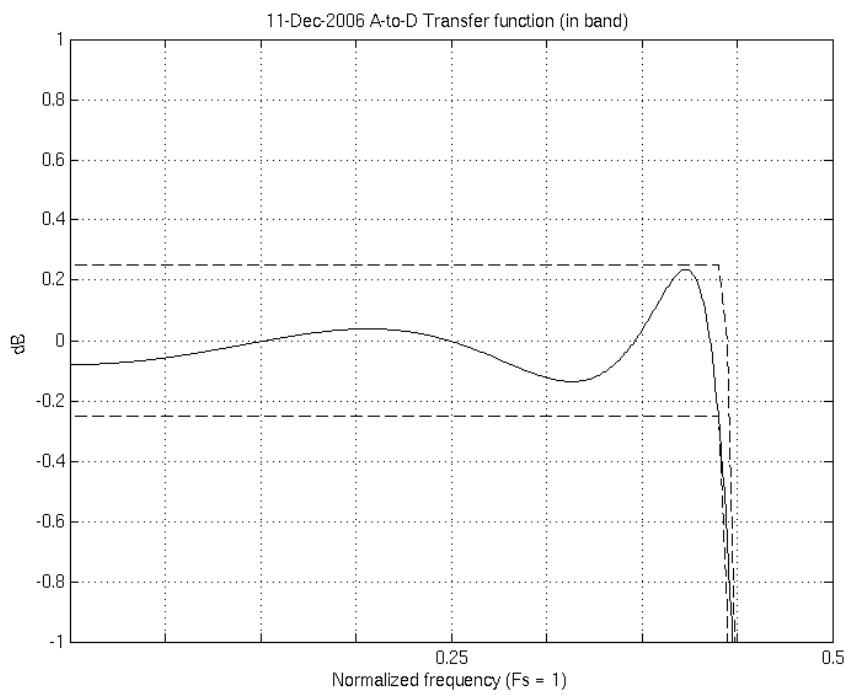


Figure 109 Codec Frequency Response, analog to digital, pass band.

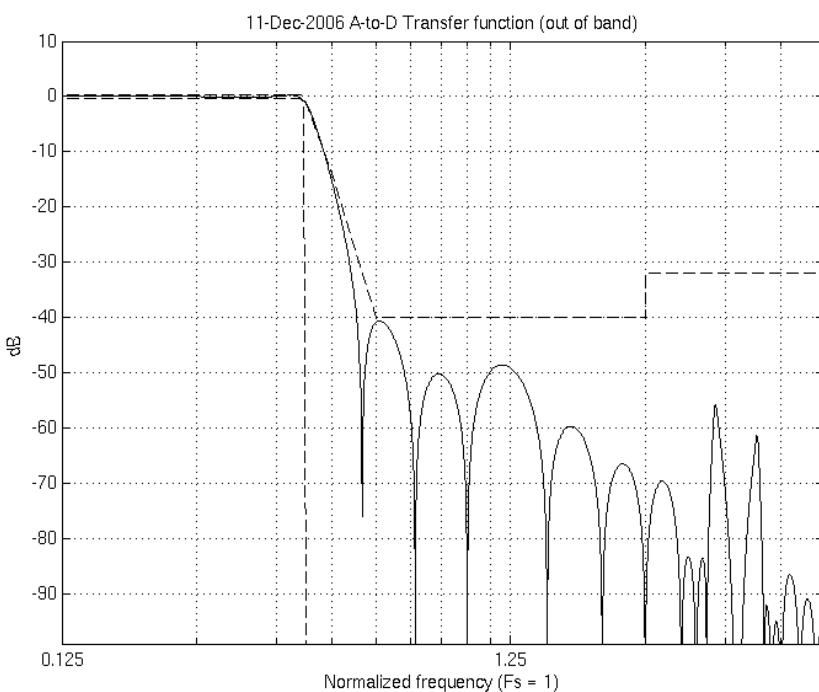


Figure 110 Codec Frequency Response analog to digital, Stop band

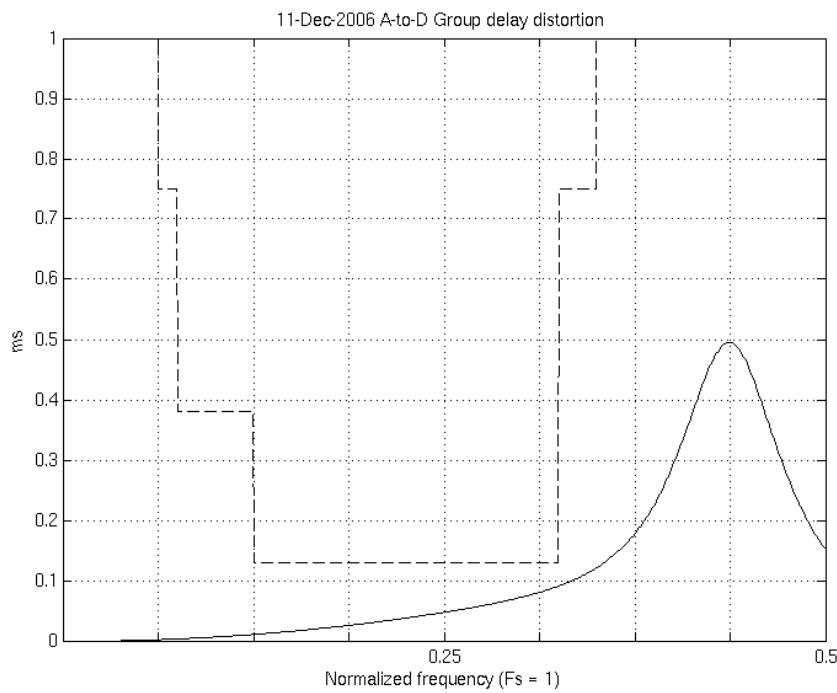


Figure 111 Codec analog to digital, group delay

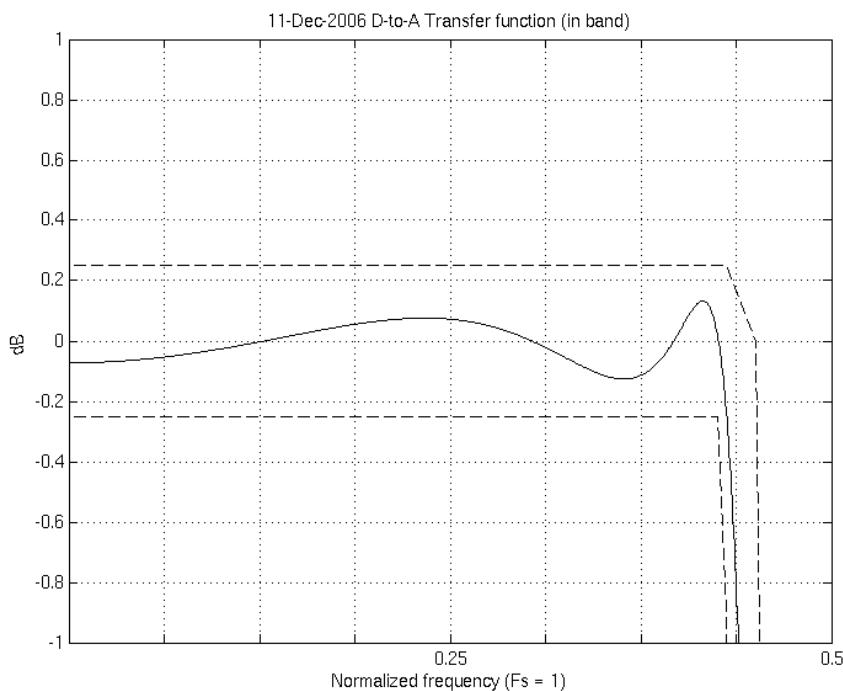


Figure 112 Frequency Response, digital to analog, pass band

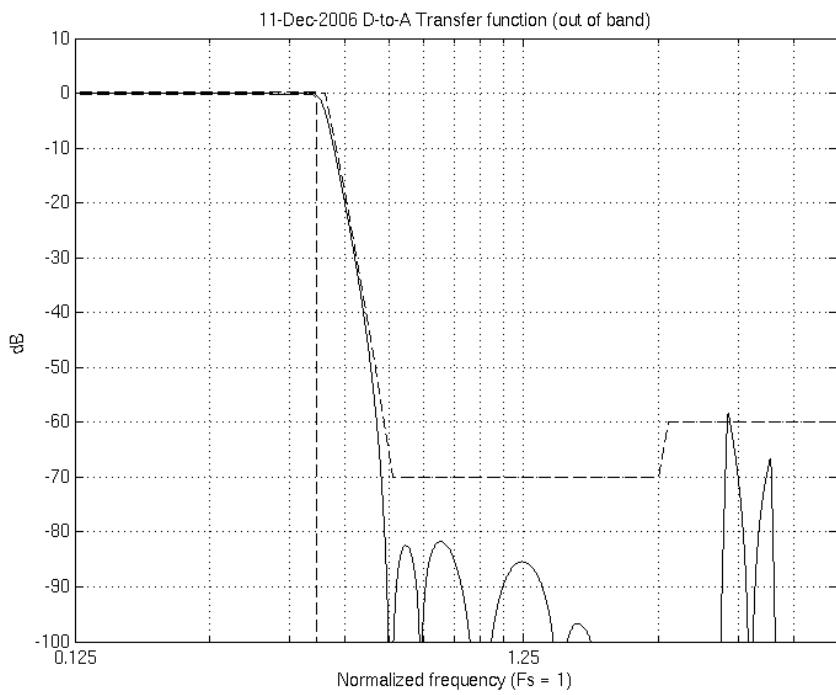


Figure 113 Codec Frequency Response, digital to analog, Stop band .

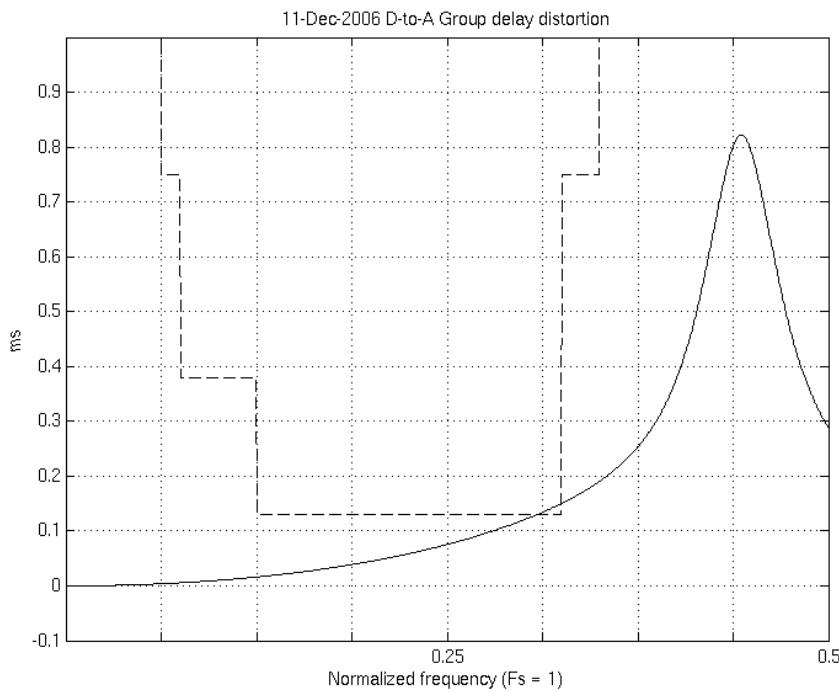


Figure 114 Codec digital to analog, Group delay distortion

39.0 Timing Specifications

Table 383: PLL lock time

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Tclock_pll1	PLL Lock time	Xtal = 10.368 MHz Fpll = 165.888. MHz			0.5	ms
Tclock_pll2	PLL Lock time	Xtal = 10.368 MHz Fpll = 48 MHz, LPF_EN = 0			0.5	ms

Table 384: PLL 1 characteristics

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Fpll1_max_vco	PLL maximum frequency	HF_SEL=1			170	MHz
Fpll1_min_vco	PLL minimum frequency	HF_SEL=0	55.296			MHz

Table 385: PLL2 characteristics

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Fpll2_max_vco	PLL maximum frequency	HF_SEL =1			56	MHz
Fpll2_min_vco	PLL minimum frequency		45			MHz

Table 386: RSTn PIN

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Tlow_RST_pu	Minimum low time to reset device	After power up.	10			ms
Tlow_RST_active		In active mode (Note 104)	15			μs

Note 104:spikes down to 100 ns may reset the device. Shorter spikes are filtered (See Figure 12 on page 34)

Table 387: XTAL with external clock

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Fxtal_ext	Input frequency external clock on XTAL for guaranteed PLL operation		9.216		13.824	MHz
Txtal_duty_ext	Duty cycle external clock on XTAL	Within Vil_xtal and Vih_xtal limits	40		60	%

Table 388: RFCLKd pin

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Frclk_9	RFCLKd Frequency	RFCLK enabled Xtal = 10.368		10.368		MHz
Txtal_duty_25	RFCLK duty cycle	VDDIO = 2.5V	40		60	%
Txtal_duty_18		VDDIO = 1.8V	30		70	%

Table 389: PD5-PD1, LE pins, PD0 internal signal

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Tpd_res	PD(6..0) Switching accuracy			1/1.152		us
Tle	LE Active high or low time	Fully programmable FDIP=10.368		N/1.152		us
Tle_acc	LE Switching accuracy			1/1.152		us

Confidential

39.1 MICROWIRE TIMING DIAGRAMS

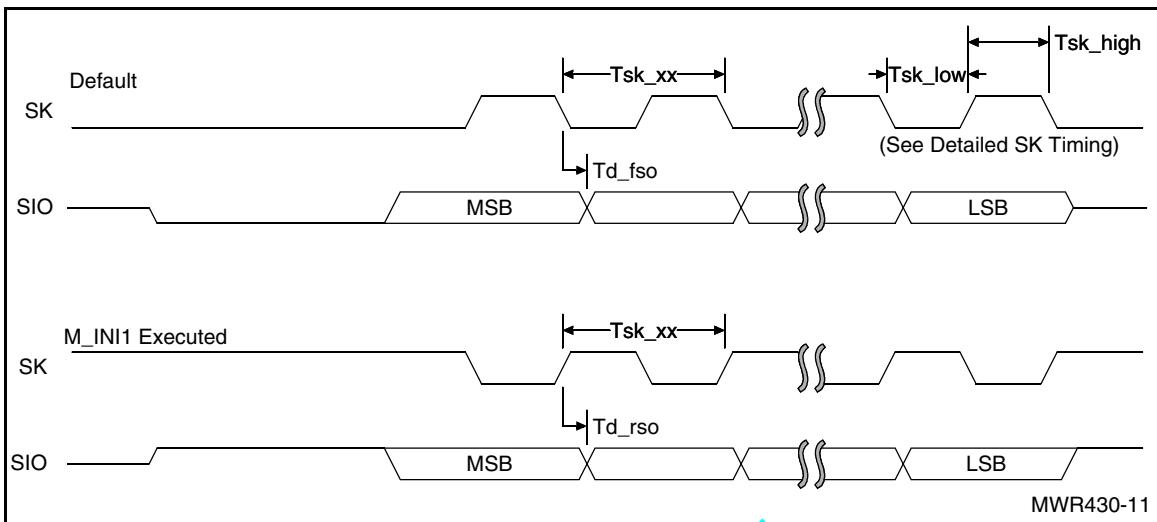


Figure 115 Microwire timing diagram

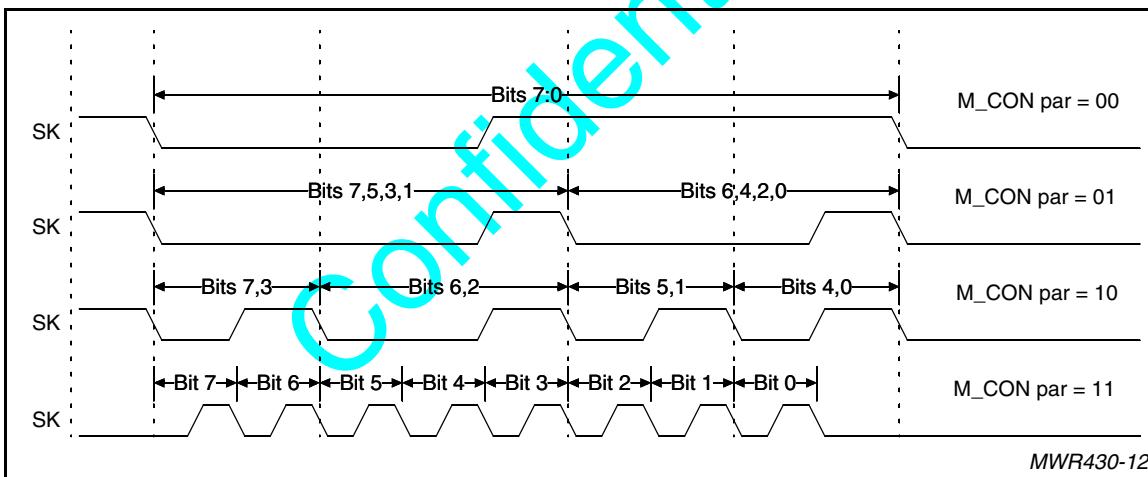


Figure 116 Detailed SK timing diagrams

Table 390: MICROWIRE interface (max 50 pF load on all outputs, FDIP=10.368 MHz)

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Tsk_00	SK period time	M_CON par = 00		9*96		ns
Tsk_01	SK period time	M_CON par = 01	4*96		5*96	ns
Tsk_02	SK period time	M_CON par = 02	2*96		3*96	ns
Tsk_04	SK period time	M_CON par = 03	1*96		2*96	ns
Tsk_low_00	SK low time (Note 105)	M_CON par = 00	4*96			ns
Tsk_low_01	SK low time (Note 105)	M_CON par = 01	3*96			ns
Tsk_low_02	SK low time (Note 105)	M_CON par = 02	96			ns
Tsk_low_03	SK low time (Note 105)	M_CON par = 03	48			ns
Td_fso	Delay time SK falling edge to SO				20	ns
Td_rso	Delay time SK rising edge to SO				20	ns

Note 105:If M_INI0 or M_INI1 are applied, the SK clock is inverted and the SK high times have the same values as the SK low times as shown above.

Confidential

39.2 UART IRDA TIMING DIAGRAMS

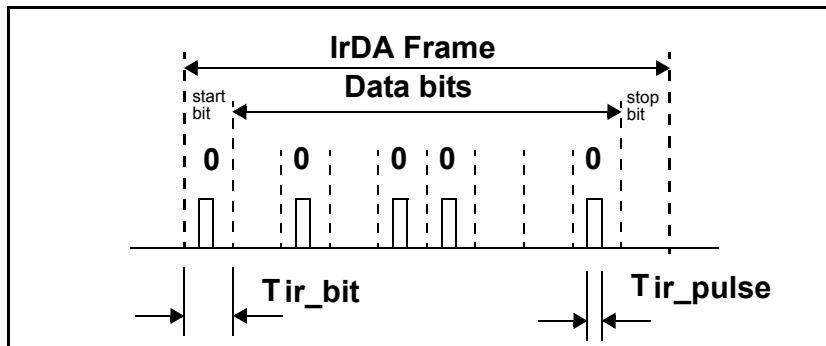


Figure 117 UART IrDA timing

Table 391: UART IrDA timing (capacitive load is 30 pF)

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Tir_pulse_96	Tx Pulse time	9600 baud (PER_DIV = 10.368 MHz)		24/1.152		us
Tir_pulse_192	Tx Pulse time	19.2 kbaud (PER_DIV = 10.368 MHz)		12/1.152		us
Tir_pulse_576	Tx Pulse time	57.6 kbaud (PER_DIV = 10.368 MHz)		4/1.152		us
Tir_pulse_115	Tx Pulse time	115.2 kbaud (PER_DIV = 10.368 MHz)		2/1.152		us
Tir_pulse	IrDA Receiver sensitivity	All baudrates (PER_DIV = 10.368 MHz)	1			us

39.3 ACCESS BUS TIMING DIAGRAM

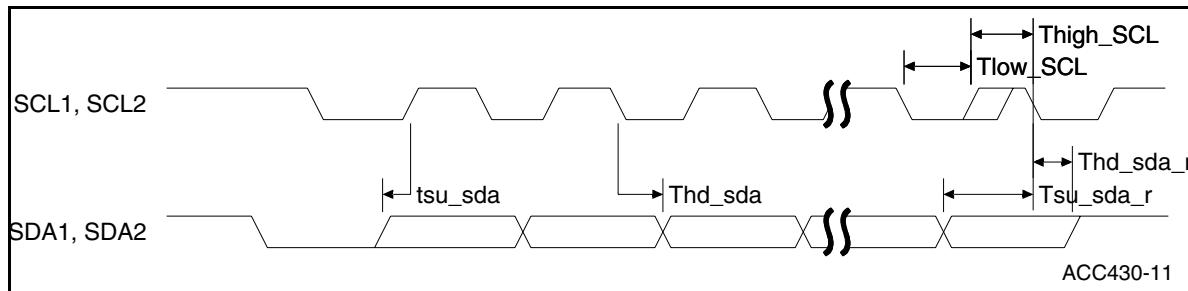


Figure 118 ACCESS bus timing

Table 392: ACCESS1, 2 bus timing (capacitive load is 30 pF)

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Tlow_scl	SCLx Low time (Note 106)	100 kHz mode		5208		ns
		400 kHz mode		1543		ns
		1.152 MHz mode		290		ns
Thigh_scl	SCLx high time (Note 106)	100 kHz mode		4823		ns
		400 kHz mode		965		ns
		1.152 MHz mode		578		ns
Tsu_sda	SDAx to SCLx setup time (transmission)	100 kHz mode	2595		2615	ns
		400 kHz mode	760		780	ns
		1.152 MHz mode	89		109	ns
Thd_sda	SCLx to SDAx hold time (transmission)	100 kHz mode	2595		2615	ns
		400 kHz mode	760		780	ns
		1.152 MHz mode	185		205	ns
Tsu_sda_r	SDAx to SCLx setup time (reception)	100 kHz mode	120			ns
		400 kHz mode	120			ns
		1.152 MHz mode	120			ns
Thd_sda_r	SCLx to SDAx hold time (reception)	100 kHz mode			0	ns
		400 kHz mode			0	ns
		1.152 MHz mode			0	ns

Note 106: The indicated values show the SCLx low and high time values if no clock stretching is applied.

If a slave device stretches the low time of SCL1 or SCL2 for less than 96 ns, the sum of Tlow_scl and Thigh_scl will stay the same. Else for every 96 ns longer stretch, the SCL clock period will increase with 96 ns.

The specified times are based on a PER_DIV output is 10.368. Refer to Figure 6 on page 25 for the clock divider overview)

39.4 PCM INTERFACE TIMING DIAGRAM

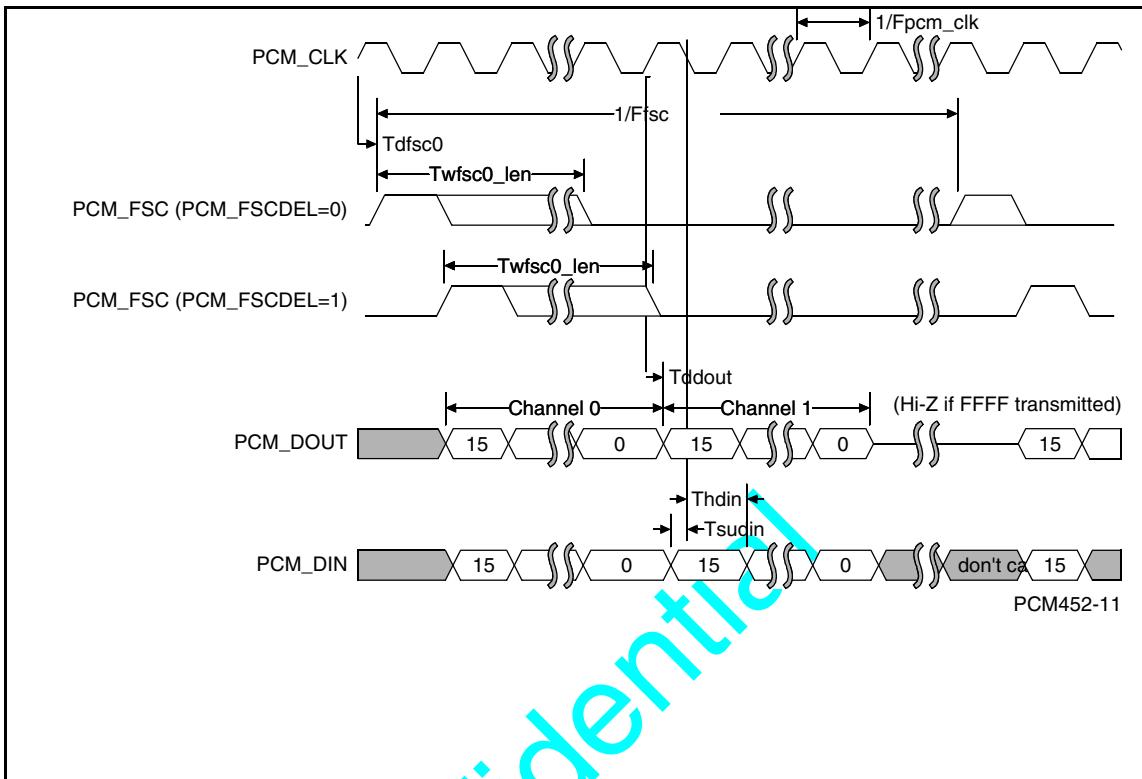


Figure 119 PCM interface timing master mode

Table 393: PCM bus timing master mode (Capacitive load is 30 pF)

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Ffsc	PCM_FSC Frequency	DSP_MAIN_SYNC1_REG [PCM_SYNC] determines frequency	8		32	kHz
Fpcm_clk	PCM_CLK frequency	CLK_CODE_REG [CLK_PCM_SEL] Determines frequency	1.152		4.608	MHz
Twfsc_len1	PCM_FSC duration	PCM_FSCLEN = 00		1		bits
Twfsc_len8	PCM_FSC duration	PCM_FSCLEN = 01		8		bits
Twfsc_len16	PCM_FSC duration	PCM_FSCLEN = 10		16		bits
Twfsc_len16	PCM_FSC duration	PCM_FSCLEN = 11		32		bits
Tdfsc	PCM_CLK to PCM_FSC delay				20	ns
Tddout	PCM_CLK to PCM_DOUT delay				20	ns
Tsudin	PCM_DIN to PCM_CLK setup time		20			ns
Thddin	PCM_DIN to PCM_CLK hold time		20			ns

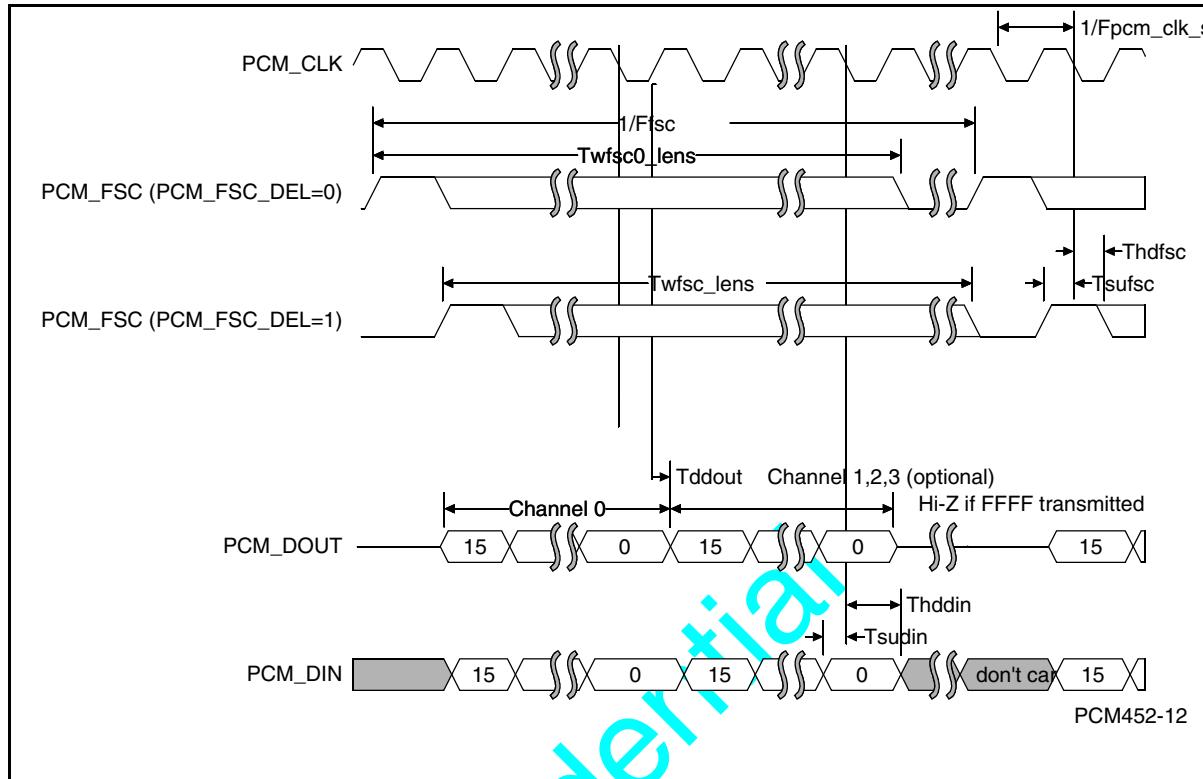


Figure 120 PCM interface timing slave mode

Table 394: PCM bus timing slave mode (capacitive load is 30 pF)

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Ffsc_s	PCM_FSC frequency			8		kHz
Fpcm_clk_s	PCM_CLK frequency		256		2048	kHz
Twfsc_lens	PMC_FSC duration	(Note 107)	1			bits
Tddout	PCM_CLK to PCM1_DOUT delay				20	ns
Tsufsc	PCM_FSC to PCM1_CLK setup time		20			ns
Thdfsc	PCM_FSC to PCM1_CLK hold time.		20			ns
Tsudin	PCM_DIN to PCM1_CLK setup time		20			ns
Thddin	PCM_DIN to PCM1_CLK hold time		20			ns

Note 107:PCM_FSC must be low for at least 1 PCM_CLK cycle

39.5 SPI BUS TIMING DIAGRAM

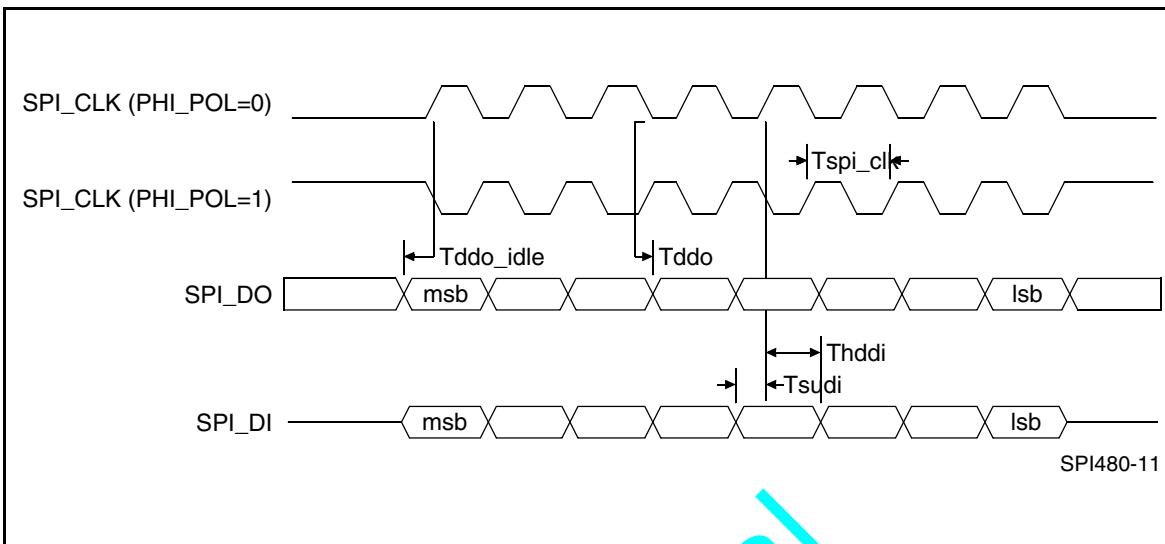


Figure 121 SPI timing

Table 395: SPI bus timing (capacitive load is 30 pF)

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Tspiclk	SPI_CLK cycle time	DIV2_CLK/ PER20_DIV = 41.476 MHz			20.736	MHz
Tddo_m	Delay time SPI_CLK master active edge to SPI_DO		-5		5	ns
Tddo_s	Delay time SPI_CLK slave active edge to SPI_DO		0		10	ns
Tddo_idle	Delay time SPI_CLK master active edge to first SPI_DO	SPI_PHA=0	(1/2 * Tspiclk) -10			ns
Tsudi_m	Setup time SPI_DI to SPI_CLK master		10			ns
Thddi_m	Hold time SPI_CLK master to SPI_DI		0			ns
Tsudi_s	Setup time SPI_DI to SPI_CLK slave		5			ns
Thddi_s	Hold time SPI_CLK slave to SPI_DI		5			ns

39.6 EBI STATIC MEMORY TIMING DIAGRAMS

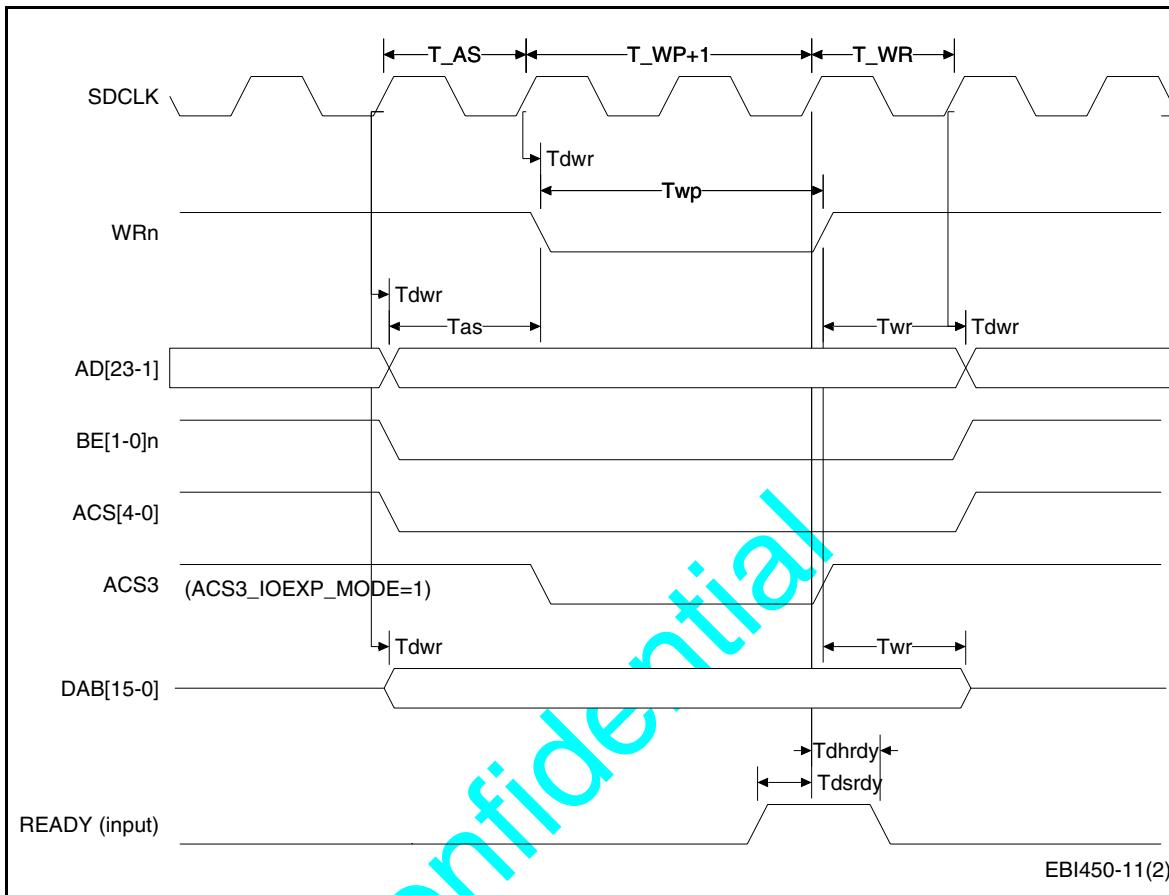


Figure 122 Static Memory Asynchronous Write cycle timing diagram

Table 396: Timing characteristics EBI Interface Static Memory write cycle (load = 30pF, VDDIO=1.8-3.3V)

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Tdwr	Delay time rising edge SDCLK to rising/falling edges of RDn, AD, ACS[4:0], BE[1:0]n, DAB[15:0]		2		5	ns
Twp	WRn low time			T_WP+1		SDCY
Tas	AD[23:1] time to WRn			T_AS		SDCY
Twr	WRn, DAB[15:0] time to AD[24:1]			T_WR		SDCY
Tdsrdy	READY setup time to SDCLK rising edge				5	ns
Tdhrdy	READY hold time from SDCLK rising edge		0			ns

SDCY = SDCLK (HCLK) cycles.

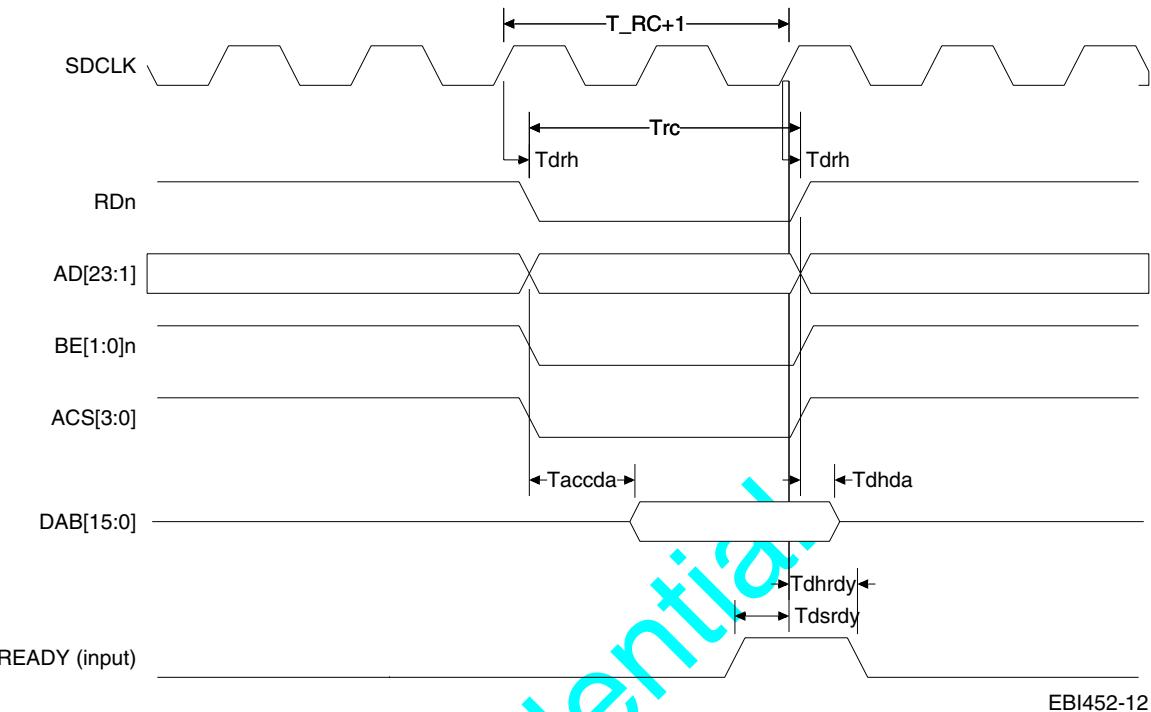


Figure 123 Static Memory Asynchronous Read cycle timing diagram

Table 397: Timing characteristics EBI Interface Asynchronous read cycle (load = 30pF, VDDIO=1.8-3.3V)

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Tdrh	Delay time rising edge SDCLK to rising/falling edges of RDn, AD, ACS[4-0], BE[1-0]n		2		5	ns
Trdc	RDn low time			T_RC+1		SDCY
Taccd	External access time + PCB delay				Trdc-21	ns
Tdhda	RDn hold time to DAB[15:0]		0			ns
Tdsrdy	Setup time READY to SDCLK rising edge				5	ns
Tdhrdy	Hold time READY to SDCLK rising edge		0			ns

Note 108: Refer to clock divider overview Figure 6 on page 25. for SDCLK cycle timing

SDCY = SDCLK (HCLK) cycles.

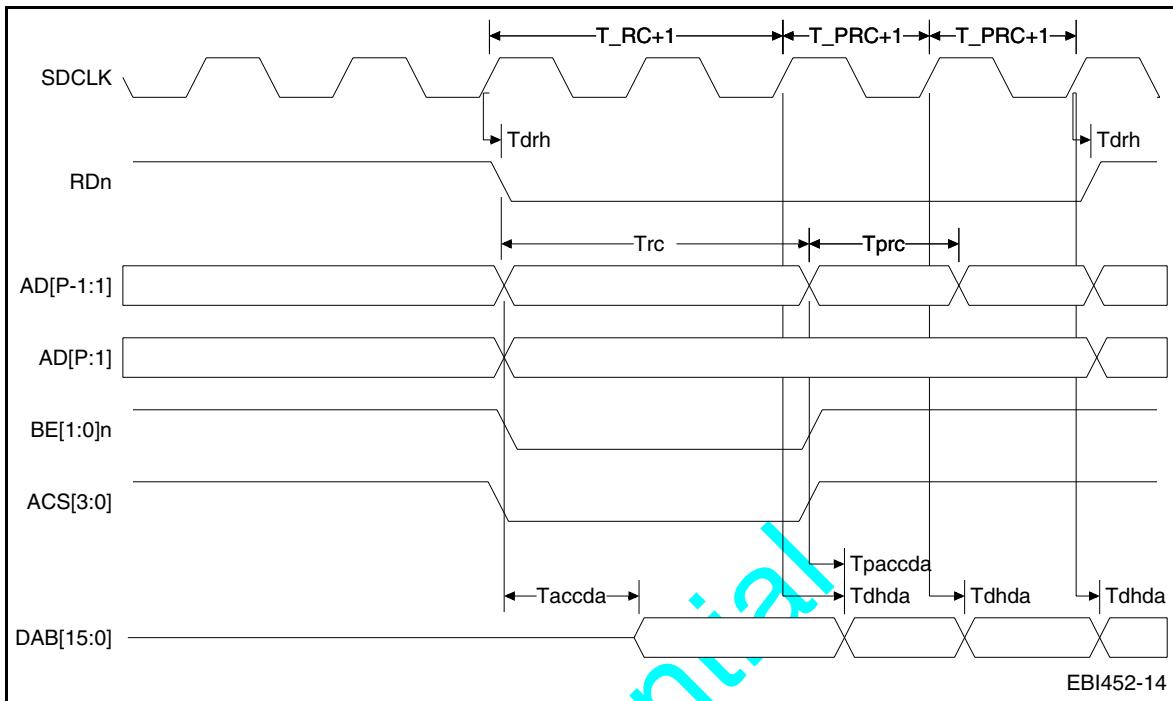


Figure 124 Static Memory Page Read cycle timing diagram

Table 398: Timing characteristics for the EBI Interface page read cycle (load = 30pF, VDDIO=1.8-3.3V)

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Tdrh	Delay time rising edge SDCLK to rising/falling edges RDn, AD, ACS[4-0], BE[1-0]n		2		5	ns
Trc	Initial read cycle			T_RC+1		SDCY
Tprc	Page read cycle time			T_PRC+1		SDCY
Taccda	Initial External access time, device + PCB delay				Trc-21	ns
Tpaccda	Page External access time, device + PCB delay				Tprc-21	ns
Tdhda	Hold time RDn to DAB[15:0]		0			ns

SDCY = SDCLK (HCLK) cycles.

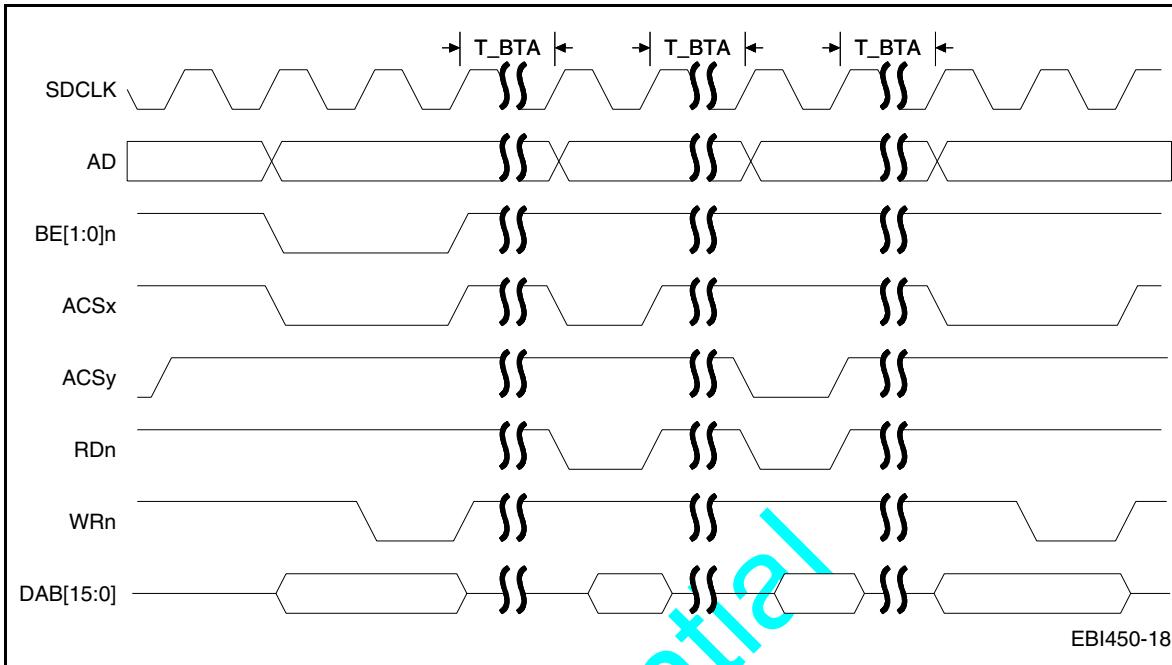


Figure 125 Static memory Bus Turn Around timing diagram

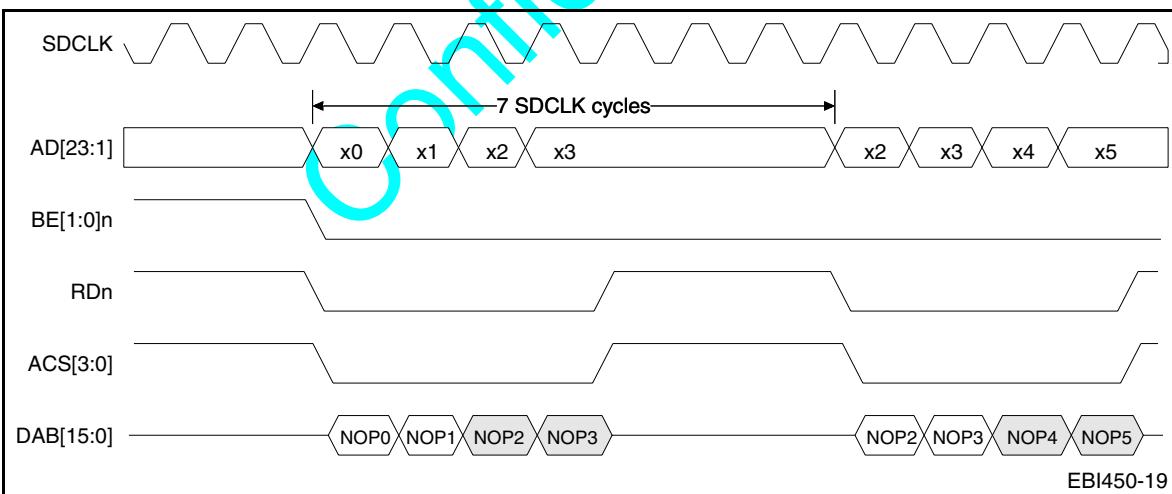


Figure 126 Static memory timing diagram CR16 executing consecutive NOPs (Cache disabled)

Figure 126 shows a basic timing diagram of the execution of consecutive NOP instructions from static memory. (Cache disabled, $T_{RC}=0$) Due to the conversion from the pipelined CR16 bus to the non-pipelined EBI static memory interface, a number of synchronization cycles are required:

- The CR16 fetches <NOP0, NOP1> as a 32 bits Word, requiring two 16 bits accesses from addresses x0 and x1 (2cycles).
- fills the pipeline (1cycle),

- decodes the instructions (2cycles)
- executes the instructions (2cycles, 1 cycle overlapping with decoding phase)
- puts the new address on the bus (1cycle, overlapping with execution phase, one clock delayed for synchronous ACS decoding).

In parallel, the EBI address generation continues (0x2, 0x3) but is restarted with the new CR16 bus address (0x2, 0x3, etc.) to fetch <NOP2, NOP3>

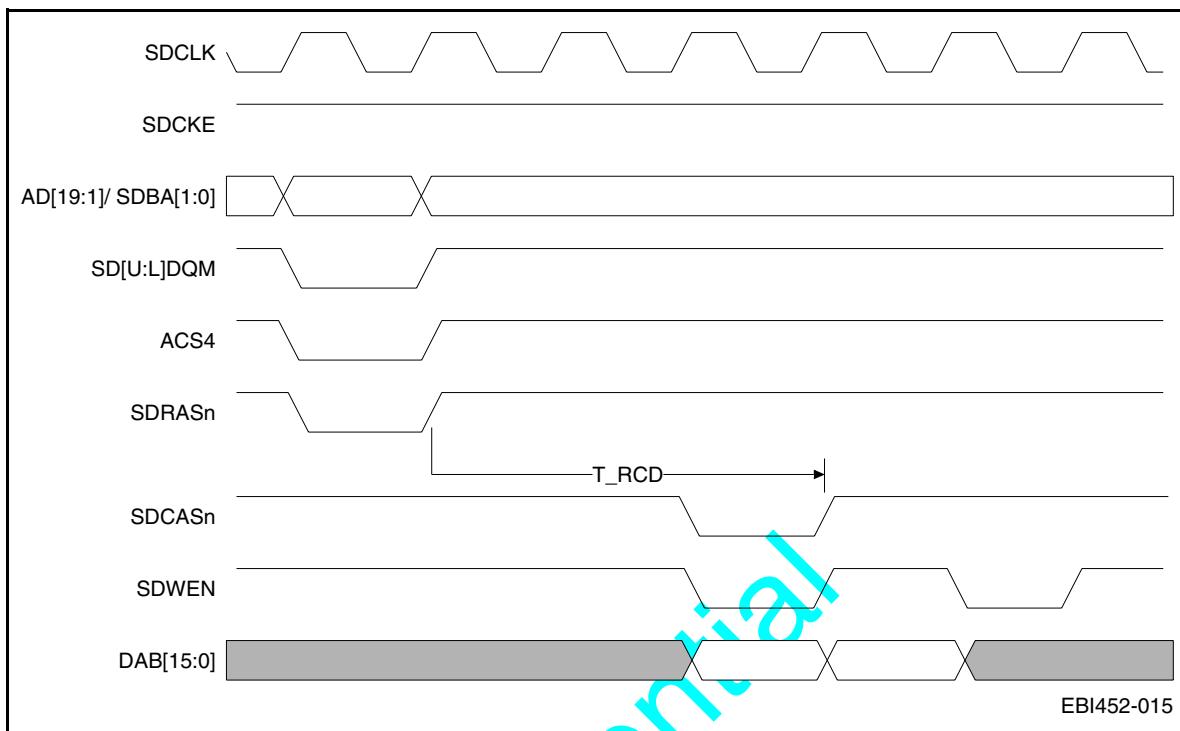
39.7 EBI SDRAM TIMING DIAGRAMS


Figure 127 SDRAM Page miss DWORD Write with burst termination timing diagram

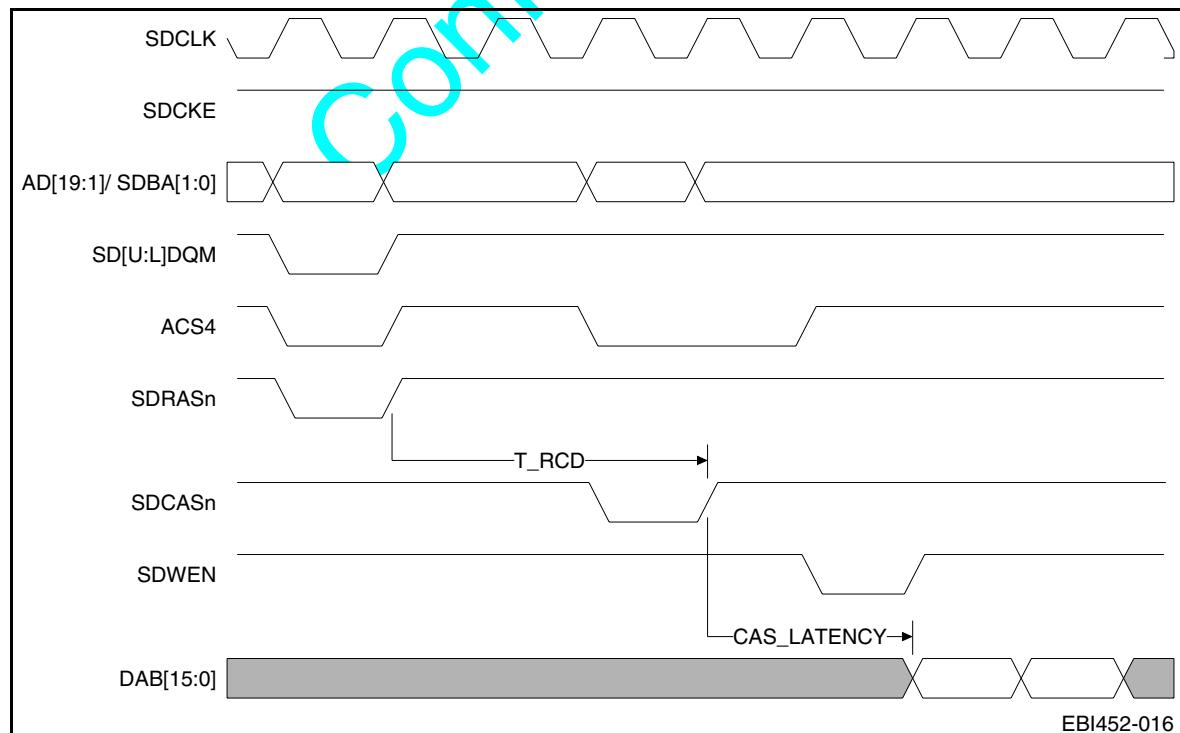


Figure 128 SDRAM Page miss DWORD Read with burst termination timing diagram

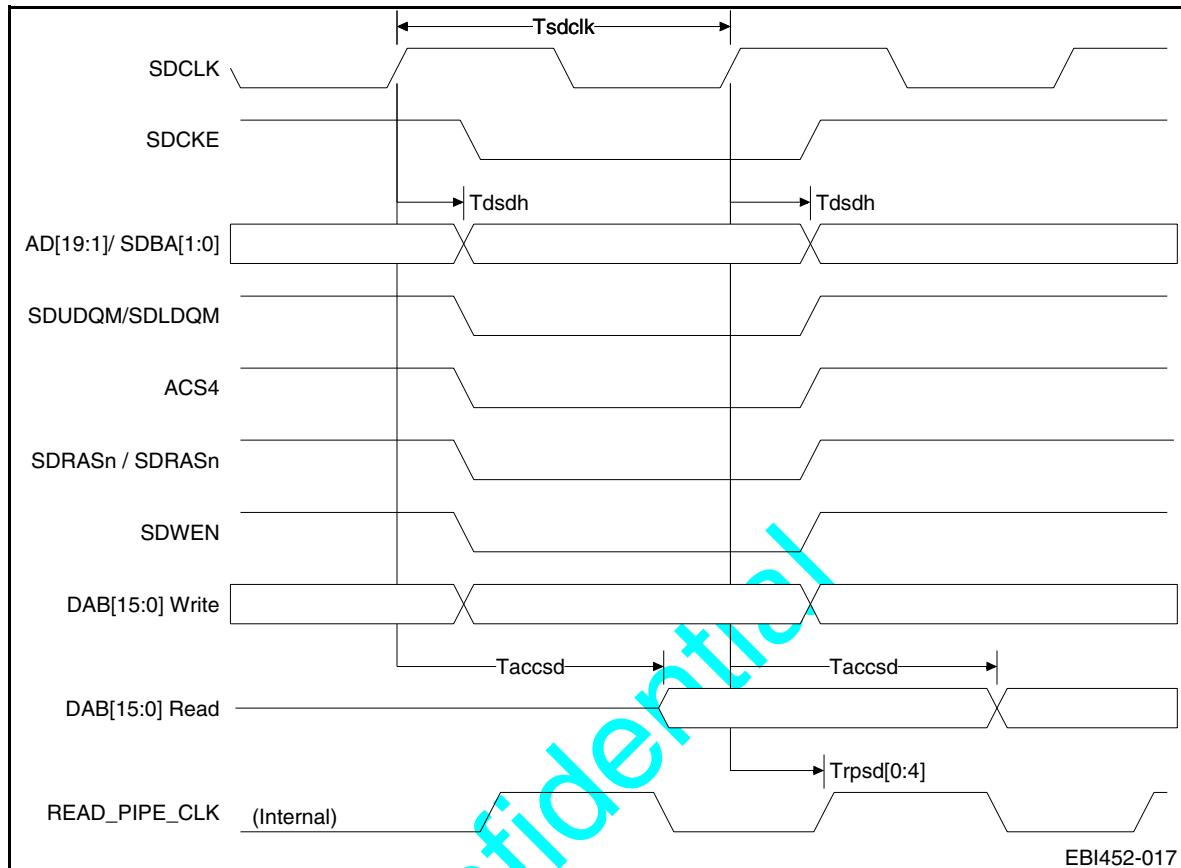


Figure 129 SDRAM Timing diagram (wave forms are for timing purposes, no CAS latency shown)

Table 399: Timing characteristics for the SBI Interface read cycle (load = 30pF, VDDIO=1.8-3.3V)

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNITS
Tdsdh	Delay time rising edge SDCLK to rising/falling edges SDCKE, AD[9:1], SDBA[1:0], SDA10, SDUDQM, SDLDQM, ACS4, SDRASn, SDCASn, SDWEN, DAB[15:0] write		1.5		9	ns
Trpsd0	EBI_SDREFR_REG [READ_PIPE_CLK] value represents the sample moment closest to 50% of the data cell. (Note 109)	READ_PIPE_CLK=0		-1.22		ns
Trpsd1		READ_PIPE_CLK=1		0.51		ns
Trpsd2		READ_PIPE_CLK=2		2.60		ns
Trpsd3		READ_PIPE_CLK=3		4.67		ns
Trpsd4		READ_PIPE_CLK=4-7		6.93		ns
Taccsd	External device access time including PCB delay		0		1/HCLK	ns
Tsdclk	Cycle time SDCLK		12			ns

Note 109:E.g Device = 7 ns, SDCLK to SDRAM PCB delay =0.3 ns, SDRAM DQ[15:0] to DAB[15:0] =0.3 ns. Total is 7.6ns. READ_PIPE_CLK shall be set closest to the middle of the received bit cell => 7.6/2= 3.6ns EBI_SDREFR[READ_PIPE_CLK] = 1. (recommended value)

EBI_SDREFR[READ_PIPE_MUX], EBSDCTRL_REG[READ_PIPE] must both be set to 2 as recommended value

The read pipe related registers must be set before SDRAM is accessed.

40.0 Package information

40.1 MOISTURE SENSITIVITY LEVEL (MSL)

The MSL is an indicator for the maximum allowable time period (floor life time) in which a moisture sensitive plastic device, once removed from the dry bag, can be exposed to an environment with a maximum temperature of 30°C and a maximum relative humidity of 60% RH. before the solder reflow process.

SC14452 128 pins package is qualified to MSL 3.

MSL Level	Floor Life Time
MSL 4	72 hours
MSL 3	168 hours
MSL 2A	4 weeks
MSL 2	1 year
MSL 1	Unlimited at 30°C/85%RH

40.2 SOLDERING INFORMATION

Refer to the JEDEC standard J-STD-020 for relevant soldering information.

This document can be downloaded from
<http://www.jedec.org>

40.3 FLOOR AND SHELF LIFE ADVISORY

Refer to SiTel Portal "Floor and shelf life advisory.pdf".

Confidential!

40.4 PACKAGE OUTLINES

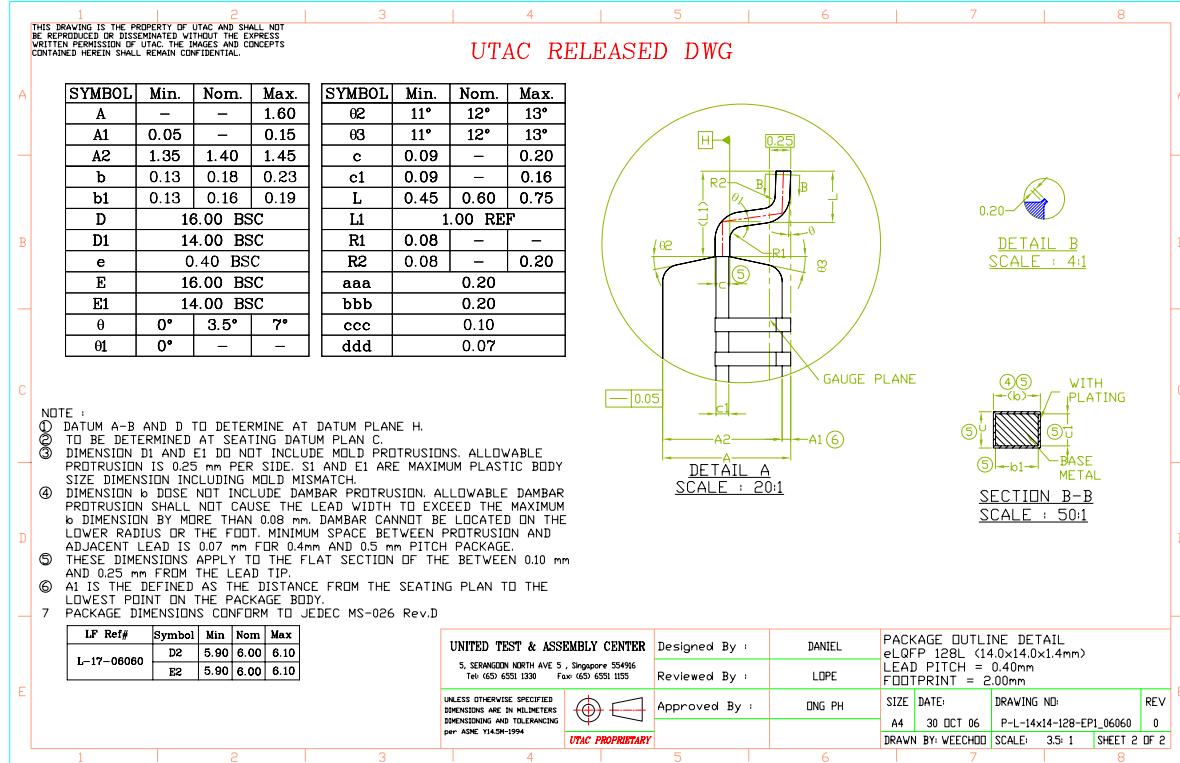
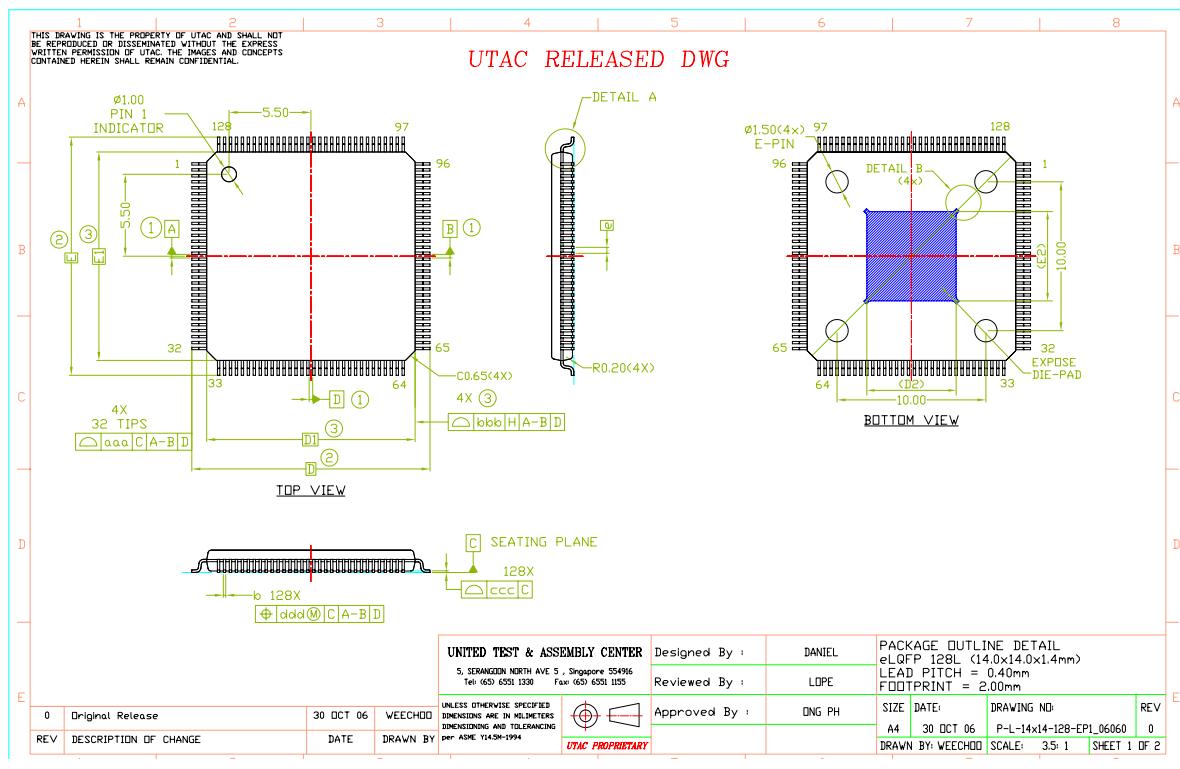


Figure 130 128 Pins e-LQFP Package Outline Drawings

VoIP processor with integrated cordless telephony interface

Product Status Definitions

Datasheet Status	Product Status	Definition
Advance Information	Formative or in Design	This data sheet contains the design specifications for product development. Specifications may change in any manner without notice.
Preliminary	First Production	This data sheet contains preliminary data. Supplementary data will be published at a later date. SiTel Semiconductor BV reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.
No Identification Noted	Full production	This data sheet contains final specifications. SiTel Semiconductor BV reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.
Obsolete	Not in Production	This data sheet contains specifications on a product that has been discontinued by SiTel Semiconductor BV. The data-sheet is printed for reference information only.

SiTel Semiconductor BV reserves the right to make changes without notice to any products herein to improve reliability, function or design. SiTel Semiconductor BV does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights, nor the right of others.

Life Support Policy

SITEL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNCIL OF SITEL SEMICONDUCTOR BV. As used herein:

1. Life support devices or systems are devices or systems which,
 - (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform, when properly used in accordance with instructions for use provided in the labelling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

SiTel Semiconductor BV does not assume any responsibility for use of any circuit described, no circuit patent licenses are implied, and SiTel reserves the right, at any time without notice, to change said circuitry or specifications.

RoHS Compliance

SiTel Semiconductor complies to DIRECTIVE 2002/95/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 27 January 2003 concerning Restriction of Hazardous Substances (RoHS).

SiTel Semiconductor RoHS statement on RoHS can be found on <http://www.sitelsemi.com/quality>. RoHS certificates from our suppliers are available on request.

Contacting SiTel Semiconductor



SiTel Semiconductor BV-HQ

Het Zuiderkruis 53
5215 MV 's-Hertogenbosch
The Netherlands
Tel: +31 (0) 73640 8822
Fax: +31 (0) 73640 8823

SiTel Semiconductor Hong Kong Ltd

SiTel Semiconductor Japan Ltd
SiTel Semiconductor Singapore
SiTel Semiconductor US
Refer to www.sitelsemi.com for addresses and telephone numbers

Customer support

dct.support@sitelsemi.com

cvm.support@sitelsemi.com

Customer service

customer.service.eu@sitelsemi.com

customer.service.us@sitelsemi.com

customer.service.jp@sitelsemi.com

customer.service.ap@sitelsemi.com