WONDERMEDIA

Data Sheet

WM8505
Application Processor

September 23, 2009
Revision 0.71

Revision History

Rev	Date	Initial	Note
0.7	Sep. 18, 09	TL	Initial internal release
0.71	Sep. 23, 09	TL	Added descriptions of NC balls to Ballout and Ball Lists

Table of Contents

List of Figures

List of Tables

Product Features

WM8505 Application Processor
Scalable Powerful RISC Core
Built-in Versatile Peripherals
Performance-enhanced JPEG Decoder
Simplified Design/Development with Low System BOM

- **300 MHz ARM926EJ-S RISC Processor**
  - ARM (Advanced RISC Microelectronics) 926EJ-S RISC Processor with MMU, 16k Bytes Instruction Cache and 16k Bytes Data Cache
  - Maximum Operating Frequency of 300 MHz @ 1.5V Core Power Voltage
  - 32/16-bit RISC Architecture (ARMv5TEJ)
  - 32-bit ARM Instruction Set and 16-bit Thumb Instruction Set
  - DSP Instruction Extensions and Single Cycle MAC
  - ARM Jazelle® Technology for Java® Acceleration

- **DDR2 SDRAM Memory Controller**
  - Supports DDR2 SDRAM Controllers, up to DDR2-667
  - Programmable Memory Timing
  - Supports 16-bit Memory Interface
  - Supports up to 512 MB

- **JPEG/Motion JPEG Decoder**
  - JPEG Baseline Profile Decoding Capability

- **Video Post-Processing Unit**
  - Built-in 2D Graphic Engine (65536/16.7M colors; 16 bpp/32 bpp) for Graphics-base Application
  - Support 1 Video and 2 Graphic Planes with Variety of Mixing Effects
  - 2D GE Supports Alpha Bitblt, ROP3, ROP4, Line Draw, and Rotate/Mirror Operations
  - Video Overlay Engine for Video, and Graphic Planes
  - Video H/V Scalar Capability (Arbitrary Scaling Limitation 0.125 ~ 16.0 with Bi-Linear Interpolation)

- **Video Interface**
  - Supports 24-bit Color VGA Output Mode for PC Monitor up to 1024x600 @ 60 Hz Refresh Rate
  - Supports 24-bit Color DVO Digital Video Output for External DVI/LVDS/HDMI Video Transmitter up to 1024x600 @ 60 Hz Refresh Rate
  - Supports Direct CCIR-656/601 Digital Video Input from External Video Decoder
  - Supports CMOS Sensor Digital Input from External CMOS Sensor Module

- **Audio Interface**
  - On-chip Audio Frequency Synthesizer to Support Sampling Rate up to 192 KHz
  - I^{2}S Audio Interface with 2-ch input and output
  - AC97 Audio Interface

- **Flash Memory Support**
  - Serial Flash Memory Controller
    - Supports Serial Flash (SPI) Memory Modules
    - Supports Two Chip Selects, up to 16 MB each
  - Strapping Option to Select whether NOR or Serial Flash Memory is used as Boot Device

- **NAND Flash Interface**
  - Supports 32 Mb – 32 Gb Flash
  - Supports up to 8-pcs Flash Chips
  - Allows Different Memory Type for Each Channel
  - Supports Both 8-bit and 16-bit Flash
  - Built-in Hardware 4-bit/8-bit ECC Correction for MLC Device and 1-bit ECC Correction for SLC Device (512-bytes ECC Calculation)
  - Configurable Timing Parameters
  - Supports Multi-Sector Read Operation
  - Supports High-Speed Programming (multi-block erase and multi-block program)

- **Secure Digital (SD) / SDIO / Multimedia Card (MMC) Interfaces**
  - Supports One SD/SDIO/MMC Interface
  - SD Memory Card Spec. 2.0 / SDIO Spec. 1.0 / MMC Spec. 4.2 Compatible
  - All SD Bus Modes Supported, Including SPI, 4-bit and 1-bit SD
  - SD and MMC Support Different Clock Rate from 390 kHz to 52 MHz
  - Supports CE-ATA Interface (compatible with MMC Version 4.2)
  - Supports High Capacity Cards

- **USB 2.0 Host / Device Controller**
  - Supports 2 USB Host Ports and One Device Port
  - USB2.0/EHCI 1.0/UHCI1.1d Specification Compliant
  - Integrated Root Hub with 2 Ports
  - Support High/Full/Low Speed Devices
  - USB2.0 UTMI Compliant

- **10/100 Ethernet MAC**
  - 802.3x Full Duplex Flow Control and Half Duplex Force Jam Capable
  - MII Interface for External Ethernet PHY Support
  - RevMII Interface for Switch Connectivity (Reverse MII)

- **Serial Peripheral Interface (SPI) Ports**
  - Three SPI Channels
  - Industry Standard Serial Peripheral Interface
  - Master and Slave SPI Modes Supported
  - All Channels Support Master Modes and Slave Mode 0 and 2
  - Four Signal Interfaces for each Port – Clock, Select, Transmit Data and Receive Data
  - Full Duplex Synchronous Serial Data Transfer
  - Programmable Clock Phase and Polarity
  - Programmable Asynchronous Transmission Speed up to 100 MHz

- **UART / IrDA Communication Ports**
  - Up to Six Half UARTs or Four Full UARTs Supported. One UART Port with Additional Support for IrDA Version 1.0 Short Range Infrared Communication
  - 16C550 Compliant UARTs with Baud Rate Support from 900 to 921.6k bps (2.4k to 115.2k bps for IrDA)
  - DMA Data Transfer Capability Using the on-chip System DMA Controller
  - RTS and CTS Modem Handshake Control Signals for Each UART when IrDA is not Used
  - Supports Bluetooth UART HCI Communication Break Signal Generation

- **I^2C Serial Port**
  - Two I^2C Ports Supported
  - 100k bps Standard Mode plus 400k bps Fast Mode
  - 7-bit Address
  - Signal-master (Multi-master not Supported)
  - Supports Slave Mode in one of the I^2C Ports
  - Software Programmable Clock Frequency and Acknowledge Bit
  - Interrupt Driven Data Transfer

- **Hardware Security Acceleration Engine**
  - Supports AES-128 Encryption / Decryption with ECB/CBC/CTR/OFB Mode
  - Supports SHA-1 (Secure Hash Algorithm) Hash Function
  - Supports RC4 Decryption

- **Keypad Interface**
  - Supports up to 8x8 Key Matrix if CCIR601/656 Digital Video Input Interface and One of the I^2C Serial Ports (I^2C1x) are not Used
- **PS/2 Keyboard Mouse Interface**
  - PS/2 & AT Keyboard and Mouse Supported
  - Intel 8042 Controller Compatible
  - Wakeup from Suspend Supported
  - Support Keyboard/Mouse Interface swapping
- **Built-in CIR Modulation Detection and Decode**
  - Supports NEC, Sony, Matsushita, and JVC Code Formats
- **Pulse Width Modulation System Timer**
  - 2 PWM Timers that may be Used as either PWM Timers or Simple System Timers
  - 2 PWM Timer Outputs to External Logic
- **General Purpose I/O**
  - 8 Dedicated General Purpose I/Os with Level/Edge Sensitive Interrupt Requests
  - Other Multiplexed GPIO Signals Allow One or More Peripherals to be Disabled for Multiplexed GPIO Signal Sections
- **System Interrupt Controller**
  - Support for up to 128 Interrupt Sources
  - Generates Interrupt and Fast-Interrupt to ARM
  - Interrupt Status Registers for Every Interrupt Output
  - Fixed and Rotating Priority Schemes
  - 16 External Interrupt Sources through GPIO Pins that are Edge or Level Sensitive
- **System DMA Controller**
  - 16 Channels
  - Supports Scatter-Gather DMA Operation
  - Memory-to-Peripheral Transfers, Peripheral-to-Memory Transfers and Memory-to-Memory Transfers Supported
  - 1-, 4-, and 8- Transfer Burst Sizes and 8 / 16 / 32-Bit Data Widths
  - Supports Dual DMA Buffers in Ping-Pong Fashion
- **Real Time Clock**
  - With on-chip Low Power 32.768 kHz Oscillator Requiring only a Crystal
  - Supports Time, Date and Day-of-the-week
  - Provides Time and Date in 24-hour or 12-hour Format with an AM/PM Flag
  - Encodes the Seconds, Minutes, Hours, Days, Months and Years in BCD Format
  - Calendar Function with Correction for Leap Year
  - Time Calibration (Providing a Time Correction Mechanism)
  - Programmable Alarm with Interrupt Request Generation
  - Second / Minute Updating Interrupt Request Generation
- **Reset and Power Management**
  - Supports Four Operating Modes: Normal, Idle, Sleep and Suspend
  - Gate-off Function of Peripheral Clocks to Put Unused Peripherals in Low Power Mode
- **PLLs and Clock Control**
  - Built-in PLLs to Provide Various Clock Frequencies
  - Internal Oscillator – Only One 27 MHz Crystal is Needed Externally
- **Operating System Timer**
  - Generates Periodic Interrupts for System Purposes
  - Generates Watchdog Reset
- **JTAG Debugging Interface**
  - In Compliance with IEEE 1149.1
  - All Internal Memory with BIST
- **Operating Temperature**
  - Commercial Grade: Ranges from 0°C to 70°C

- **479-ball PBGA Package (19 x 19 mm with 0.8 mm ball pitch)**

- **Operating Systems**
  - Embedded Linux
  - Windows CE

- **Supporting Software Package (based on selected OS)**
  - Device Drivers for Internal Peripherals
  - Supporting Binary Library, i.e. Linux Loadable Modules
  - Porting Guide
  - Sample Applications for Chip Validation, System H/W Evaluation, and Reference Design
  - Application Shell
  - Media Player
  - Image Viewer
  - System Setup Utilities
  - API, SDK, and Application Sample Codes

WM8505 System Overview

The WM8505 Application Processor is the SoC solution with superior networks and display capabilities that specially tailored for cost-effective embedded multimedia devices. The WM8505 supports display resolution up to 1024x600 and 10/100 Ethernet MAC. The WM8505 is a low power consumption SoC solution that demands 1.8V for DDR2 SDRAM, and 3.3 V for other interfaces as operating power. The WM8505 integrates multimedia features and functionalities, e.g. decoding popular audio/video streams, 2D graphics display, and peripheral I/Os that are aiming to meet upcoming market demands as well as reduce customer's total BOM cost. The WM8505 is designed to deliver the state-of-the-art performance for applications such as Networked Projector, Digital Signage, and Thin-Client Terminal.

The WM8505 Application Processor integrates all popular peripherals in an efficiently architected RISC (reduced instruction set computer) platform. Figure 1 illustrates the system block diagram of WonderMedia's WM8505 Application Processor.

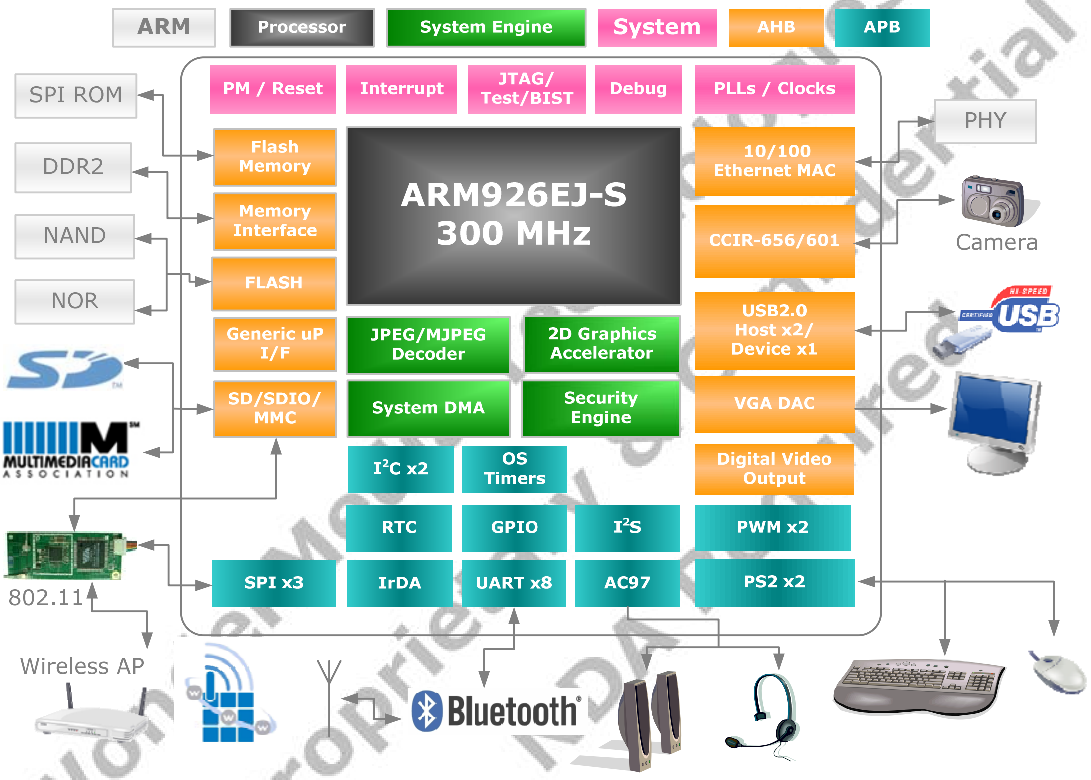


Figure 1 - WM8505 System Block Diagram

Figure 2 illustrates the software architecture of the WM8505 Application Processor. The WM8505 supports Linux and Windows CE for embedded systems. WonderMedia provides the software package that includes device drivers for multimedia, peripheral, and security-related devices as well as multimedia player with high audio, video, and image quality. The software architecture of WM8505 can easily collaborate with the 3rd party software packages to provide versatile applications.

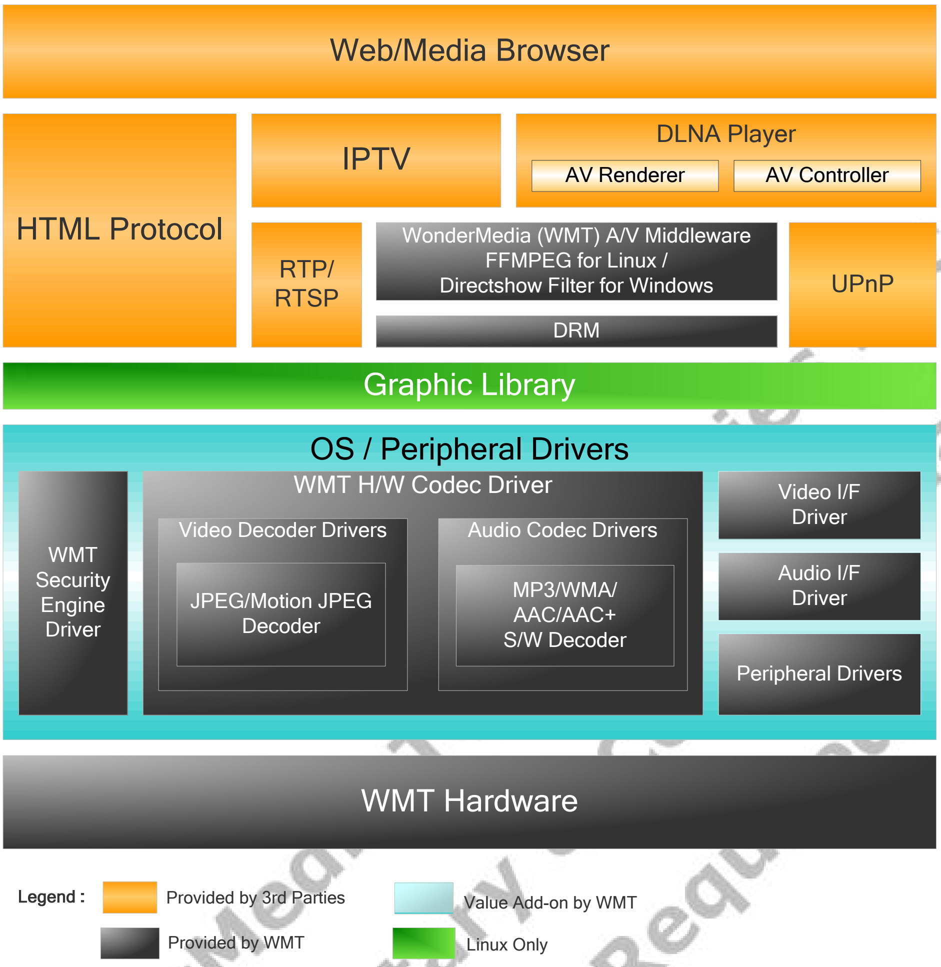


Figure 2 - WM8505 Software Architecture

Ballout: 479-ball PBGA Package (19 x 19 mm with 0.8 mm ball pitch)

	1	2	3	4	5	6	7	8	9	10	11
A	NC	MII0MDIO	MII0MDC	MII0RXD3	MII0RXD0	MII0TXC	MII0TXD1	PWREN33	PWREN18	PWREN15	PWRBTN#
B	NC	NC	MII0RXDV	GND	MII0RXC/GRXC	GND	MII0TXD0	MII0PHYPD#	WAKEUP3	RSMRST#	MSDT
C	NC	GND	NC	MII0RXER	MII0RXD1	MII0TXD3	MII0COL	MII0TXEN	WAKEUP0	WAKEUP2	KBDT
D	NC	NC	NC	NC	MII0RXD2	MII0TXD2	MII0CRS	PHY25MHZ	MII0PHYRST#	WAKEUP1/USBATTA	VSUS33
E	NC	NC	NC	GND	NC	GND	VSUS33	VSUS33	GND	VSUS15	VSUS33
F	NC	GND	NC	NC	NC	VSUS33	VCC33	VCC33	VCC33	VCC33	VCC33
G	NC	NC	NC	NC	MEMCOMP	VCCMEM	VDD	VDD	VDD	VDD	VDD
H	MEMA01	MEMA02	NC	GND	MEMCS0#	VCCMEM	VDD	GND	GND	GND	GND
J	MEMA07	MEMA04	MEMA00	MEMA03	MEMVREF0	VCCMEM	VDD	GND	GND	GND	GND
K	MEMA06	GND	MEMRAS#	MEMCAS#	MEMWE#	VCCMEM	VDD	GND	GND	GND	GND
L	MEMA05	MEMA08	MEMBA0	GND	MEMODT	VCCMEM	VDD	GND	GND	GND	GND
M	MEMA10	MEMA09	MEMA11	MEMA12	MEMBA1	VCCMEM	VDD	GND	GND	GND	GND
N	MEMD08	GND	MEMD13	MEMA13	MEMCKE	VCCMEM	VDD	GND	GND	GND	GND
P	MEMCLK-	MEMCLK+	MEMD14	GND	MEMBA2	VCCMEM	VDD	GND	GND	GND	GND
R	MEMDQM1#	MEMDQS1+	MEMDQS1-	MEMD10	MEMVREF1	VCCMEM	VDD	GND	GND	GND	GND
T	MEMD15	GND	MEMD11	MEMD12		NORA10	VDD	VDD	VDD	VDD	VDD
U	MEMD09	MEMD03	MEMD01	GND		NORA11	NORBAA#	VCC33	VCC33	VCC33	VCC33
V	MEMD07	MEMDQS0+	MEMDQS0-	MEMD05	NORCLK	NORA06/NANDCE6#	NORA22	NORA02/NANDCE2#	NORA08/NANDRB0#	NORD09/NANDIO09	NOROE#/NANDRE#
W	MEMDQM0#	GND	MEMD02	NORCE2#	NORA14	NORA13	NORA04/NANDCE4#	NORA03/NANDCE3#	NORADV#	NORRST#/NANDALE	NORWE#/NANDWE#
Y	MEMD00	MEMD04	NORCE3#	NORA12	NORA20	NORA17	NORD15/NANDIO15	NORD14/NANDIO14	GND	NORD08/NANDIO08	NORCE0#/NANDCLE
AA	MEMD06	NORA24/GUPBE1#	NORA15	NORA21	NORA18	NORA07/NANDCE7#	NORA01/NANDCE1#	NORD12/NANDIO12	NORD11/NANDIO11	NORD06/NANDIO06	NORD05/NANDIO05
AB	NORCE1#	NORA23/GUPBE0#	NORA09/NANDRB1#	NORA19	NORA16	NORA05/NANDCE5#	NORA00/NANDCE0#	NORD13/NANDIO13	NORD10/NANDIO10	NORD07/NANDIO07	NORD04/NANDIO04
	1	2	3	4	5	6	7	8	9	10	11

Figure 3 - WM8505 Ball Diagram – Left (Top View)

12	13	14	15	16	17	18	19	20	21	22	
PWRGD	USBP2-	GND	RTCXI	VBAT	MMCDATA4	SDCLK	SDDATA0	VDOUT23	VDOUT20	VDOUT19	A
MSCK	USBP2+	USBP1-	RTCXO	SDDATA2	SDCMD	MMCDATA6	GND	VDOUT22	GND	VDOUT18	B
KBCK	GND	USBP1+	USBREXT	SDDATA3	MMCDATA5	MMCDATA7	VDOUT21	VDOUT17	VDOUT16	VDOUT15	C
SUSGPIO	USBOC1#	USBP0+	VCCA33USBPLL	SDDATA1	VDOUT14	GND	VDOUT13	GND	VDOUT12	VDOUT11	D
VSUS33	USBOC0#	USBP0-	VCCA15USBPLL	SDWP	VDOUT10	VDOUT09	VDCLK	VDOUT08	VDOUT07	VDOUT06	E
VCC33USB	VCC33USB	VCC33USB	VCC33	VDOUT03	GND	SDCDT	GND	VDOUT05	GND	VDOUT04	F
VDD	VDD	VDD	VDD	VCC33	VDOUT00	VDHSYNC	VDDEN	VDVSYNC	VDOUT02	VDOUT01	G
GND	GND	GND	GND	VDD	VCC33	KPADROW5	VCCA33PLLAB	VGAHSYNC	VGAVSYNC	GND	H
GND	GND	GND	GND	VDD	VCC33	CLKTST/KPADROW3	VCCA33PLLCD	VGAREXT	VGAG	VGAR	J
GND	GND	GND	GND	VDD	VCC33	GNDAPLL1	24MHZXI	GND	VCCA33DAC	VGAB	K
GND	GND	GND	GND	VDD	VCC33	GNDAPLL0	VCCA33PLL1	CLKOUT/KPADROW4	GNDAXTAL	VCCA33DAC	L
GND	GND	GND	GND	VDD	VCC33	VDIN3/KPADCOL3	VCCA33PLL0	VCCA33XTAL	27MHZXO	27MHZXI	M
GND	GND	GND	GND	VDD	VCC33	VHSYNC/KPADROW0	GND	VDIN7/KPADCOL7	VDIN6/KPADCOL6	VDIN5/KPADCOL5	N
GND	GND	GND	GND	VDD	VCC33	VVSYNC/KPADROW1	VDIN2/KPADCOL2	VDIN4/KPADCOL4	VCLK/KPADROW2	I2C1SCL/KPADROW6	P
GND	GND	GND	GND	VDD	VCC33	VDIN1/KPADCOL1	VDIN0/KPADCOL0	I2C0SCL	I2C1SDA/KPADROW7	I2C0SDA	R
VDD	VDD	VDD	VDD	VCC33	PWMOUT1	UART1CTS	UART0RTS	UART0CTS	UART0TXD	UART0RXD	T
VCC33	VCC33	VCC33	VCC33	SPI2SS#	PWMOUT0	UART2RTS/UART4TXD	AC97RST#/I2SMCLK	UART1RTS	UART1TXD	UART1RXD	U
NORWAIT#/NANDWPD#	NORD00/NANDIO00	SPI0SS#	SPI1MOSI	SPI2CLK	UART2CTS/UART4RXD	UART2RXD	UART2TXD	AC97SDI/I2SDIN	AC97SDO/I2SDOUT	AC97SYNC/I2SWS	V
NORWP#/NANDWP#	SFCLK	SPI0MISO	SPI1CLK	SPI2MOSI	UART3RXD	JTAGTRST#	GND			AC97BCLK/I2SCLK	W
NORD03/NANDIO03	GND	SFDI	SPI1MISO	GND	UART3RTS/UART5TXD	JTAGTCK	RESETOUT#	GPIO2	GPIO6	GPIO3	Y
NORD02/NANDIO02	SFCS1#	SFDO	SPI0CLK	SPI2MISO	UART3TXD	JTAGTDI	TESTMODE	CIRIN	GPIO7	GPIO5	AA
NORD01/NANDIO01		SFCS0#	SPI0MOSI	SPI1SS#	UART3CTS/UART5RXD	JTAGTDO	JTAGTMS	GPIO1	GPIO0	GPIO4	AB
12	13	14	15	16	17	18	19	20	21	22	

Figure 4 - WM8505 Ball Diagram – Right (Top view)

Ball Lists: WM8505 479-ball PBGA Package

Signal Ball List: WM8505 479-ball PBGA Package – Sorted by Ball Number

Table 1 - WM8505 Ball and Signal List – Sorted by Ball Number

Ball	Signal Name	Ball	Signal Name	Ball	Signal Name	Ball	Signal Name
A01	NC	B22	VDOUT18	D22	VDOUT11	H20	VGAHSYNC
A02	MII0MDIO	C01	NC	E01	NC	H21	VGAVSYNC
A03	MII0MDC	C03	NC	E02	NC	J01	MEMA07
A04	MII0RXD3	C04	MII0RXER	E03	NC	J02	MEMA04
A05	MII0RXD0	C05	MII0RXD1	E05	NC	J03	MEMA00
A06	MII0TXC	C06	MII0TXD3	E13	USBOC0#	J04	MEMA03
A07	MII0TXD1	C07	MII0COL	E14	USBP0-	J05	MEMVREF0
A08	PWREN33	C08	MII0TXEN	E16	SDWP	J18	CLKTST/KPADROW3
A09	PWREN18	C09	WAKEUP0	E17	VDOUT10	J20	VGAREXT
A10	PWREN15	C10	WAKEUP2	E18	VDOUT09	J21	VGAG
A11	PWRBTN#	C11	KBDT	E19	VDCLK	J22	VGAR
A12	PWRGD	C12	KBCK	E20	VDOUT08	K01	MEMA06
A13	USBP2-	C14	USBP1+	E21	VDOUT07	K03	MEMRAS#
A15	RTCXI	C15	USBREXT	E22	VDOUT06	K04	MEMCAS#
A16	VBAT	C16	SDDATA3	F01	NC	K05	MEMWE#
A17	MMCDATA4	C17	MMCDATA5	F03	NC	K19	24MHZXI
A18	SDCLK	C18	MMCDATA7	F04	NC	K21	VCCA33DAC
A19	SDDATA0	C19	VDOUT21	F05	NC	K22	VGAB
A20	VDOUT23	C20	VDOUT17	F16	VDOUT03	L01	MEMA05
A21	VDOUT20	C21	VDOUT16	F18	SDCDT	L02	MEMA08
A22	VDOUT19	C22	VDOUT15	F20	VDOUT05	L03	MEMBA0
B01	NC	D01	NC	F22	VDOUT04	L05	MEMODT
B02	NC	D02	NC	G01	NC	L20	CLKOUT/KPADROW4
B03	MII0RXDV	D03	NC	G02	NC	M01	MEMA10
B05	MII0RXC/GRXC	D04	NC	G03	NC	M02	MEMA09
B07	MII0TXD0	D05	MII0RXD2	G04	NC	M03	MEMA11
B08	MII0PHYPD#	D06	MII0TXD2	G05	MEMCOMP	M04	MEMA12
B09	WAKEUP3	D07	MII0CRS	G17	VDOUT00	M05	MEMBA1
B10	RSMRST#	D08	PHY25MHZ	G18	VDHSYNC	M18	VDIN3/KPADCOL3
B11	MSDT	D09	MII0PHYRST#	G19	VDDEN	M21	27MHZXO
B12	MSCK	D10	WAKEUP1/USBATTA	G20	VDVSYNC	M22	27MHZXI
B13	USBP2+	D12	SUSGPIO	G21	VDOUT02	N01	MEMD08
B14	USBP1-	D13	USBOC1#	G22	VDOUT01	N03	MEMD13
B15	RTCXO	D14	USBP0+	H01	MEMA01	N04	MEMA13
B16	SDDATA2	D16	SDDATA1	H02	MEMA02	N05	MEMCKE
B17	SDCMD	D17	VDOUT14	H03	NC	N18	VHSYNC/KPADROW0
B18	MMCDATA6	D19	VDOUT13	H05	MEMCS0#	N20	VDIN7/KPADCOL7
B20	VDOUT22	D21	VDOUT12	H18	KPADROW5	N21	VDIN6/KPADCOL6

Signal Ball List: WM8505 479-ball PBGA Package – Sorted by Ball Number (Continued)

Ball	Signal Name	Ball	Signal Name	Ball	Signal Name	Ball	Signal Name
N22	VDIN5/KPADCOL5	U19	AC97RST#/I2SMCLK	W14	SPI0MISO	AA14	SFDO
P01	MEMCLK-	U20	UART1RTS	W15	SPI1CLK	AA15	SPI0CLK
P02	MEMCLK+	U21	UART1TXD	W16	SPI2MOSI	AA16	SPI2MISO
P03	MEMD14	U22	UART1RXD	W17	UART3RXD	AA17	UART3TXD
P05	MEMBA2	V01	MEMD07	W18	JTAGTRST#	AA18	JTAGTDI
P18	VVSYNC/KPADROW1	V02	MEMDQS0+	W22	AC97BCLK/I2SCLK	AA19	TESTMODE
P19	VDIN2/KPADCOL2	V03	MEMDQS0-	Y01	MEMD00	AA20	CIRIN
P20	VDIN4/KPADCOL4	V04	MEMD05	Y02	MEMD04	AA21	GPIO7
P21	VCLK/KPADROW2	V05	NORCLK	Y03	NORCE3#	AA22	GPIO5
P22	I2C1SCL/KPADROW6	V06	NORA06/NANDCE6#	Y04	NORA12	AB01	NORCE1#
R01	MEMDQM1#	V07	NORA22	Y05	NORA20	AB02	NORA23/GUPBE0#
R02	MEMDQS1+	V08	NORA02/NANDCE2#	Y06	NORA17	AB03	NORA09/NANDRB1#
R03	MEMDQS1-	V09	NORA08/NANDRB0#	Y07	NORD15/NANDIO15	AB04	NORA19
R04	MEMD10	V10	NORD09/NANDIO09	Y08	NORD14/NANDIO14	AB05	NORA16
R05	MEMVREF1	V11	NOROE#/NANDRE#	Y10	NORD08/NANDIO08	AB06	NORA05/NANDCE5#
R18	VDIN1/KPADCOL1	V12	NORWAIT#/NANDWPD#	Y11	NORCE0#/NANDCLE	AB07	NORA00/NANDCE0#
R19	VDIN0/KPADCOL0	V13	NORD00/NANDIO00	Y12	NORD03/NANDIO03	AB08	NORD13/NANDIO13
R20	I2C0SCL	V14	SPI0SS#	Y14	SFDI	AB09	NORD10/NANDIO10
R21	I2C1SDA/KPADROW7	V15	SPI1MOSI	Y15	SPI1MISO	AB10	NORD07/NANDIO07
R22	I2C0SDA	V16	SPI2CLK	Y17	UART3RTS/UART5TXD	AB11	NORD04/NANDIO04
T01	MEMD15	V17	UART2CTS/UART4RXD	Y18	JTAGTCK	AB12	NORD01/NANDIO01
T03	MEMD11	V18	UART2RXD	Y19	RESETOUT#	AB14	SFCS0#
T04	MEMD12	V19	UART2TXD	Y20	GPIO2	AB15	SPI0MOSI
T06	NORA10	V20	AC97SDI/I2SDIN	Y21	GPIO6	AB16	SPI1SS#
T17	PWMOUT1	V21	AC97SDO/I2SDOUT	Y22	GPIO3	AB17	UART3CTS/UART5RXD
T18	UART1CTS	V22	AC97SYNC/I2SWS	AA01	MEMD06	AB18	JTAGTDO
T19	UART0RTS	W01	MEMDQM0#	AA02	NORA24/GUPBE1#	AB19	JTAGTMS
T20	UART0CTS	W03	MEMD02	AA03	NORA15	AB20	GPIO1
T21	UART0TXD	W04	NORCE2#	AA04	NORA21	AB21	GPIO0
T22	UART0RXD	W05	NORA14	AA05	NORA18	AB22	GPIO4
U01	MEMD09	W06	NORA13	AA06	NORA07/NANDCE7#		
U02	MEMD03	W07	NORA04/NANDCE4#	AA07	NORA01/NANDCE1#		
U03	MEMD01	W08	NORA03/NANDCE3#	AA08	NORD12/NANDIO12		
U06	NORA11	W09	NORADV#	AA09	NORD11/NANDIO11		
U07	NORBAA#	W10	NORRST#/NANDALE	AA10	NORD06/NANDIO06		
U16	SPI2SS#	W11	NORWE#/NANDWE#	AA11	NORD05/NANDIO05		
U17	PWMOUT0	W12	NORWP#/NANDWP#	AA12	NORD02/NANDIO02		
U18	UART2RTS/UART4TXD	W13	SFCLK	AA13	SFCS1#		

Signal Ball List: WM8505 479-ball PBGA Package – Sorted by Signal Name

Table 2 – WM8505 Ball and Signal List – Sorted by Signal Name

Ball #	Signal Name	Ball #	Signal	Ball #	Signal Name	Ball #	Signal Name
K19	24MHZXI	J01	MEMA07	L05	MEMODT	D04	NC
M22	27MHZXI	L02	MEMA08	K03	MEMRAS#	E01	NC
M21	27MHZXO	M02	MEMA09	J05	MEMVREF0	E02	NC
W22	AC97BCLK/I2SCLK	M01	MEMA10	R05	MEMVREF1	E03	NC
U19	AC97RST#/I2SMCLK	M03	MEMA11	K05	MEMWE#	E05	NC
V20	AC97SDI/I2SDIN	M04	MEMA12	C07	MII0COL	F01	NC
V21	AC97SDO/I2SDOUT	N04	MEMA13	D07	MII0CRS	F03	NC
V22	AC97SYNC/I2SWS	L03	MEMBA0	A03	MII0MDC	F04	NC
AA20	CIRIN	M05	MEMBA1	A02	MII0MDIO	F05	NC
L20	CLKOUT/KPADROW4	P05	MEMBA2	B08	MII0PHYPD#	G01	NC
J18	CLKTST/KPADROW3	K04	MEMCAS#	D09	MII0PHYRST#	G02	NC
AB21	GPIO0	N05	MEMCKE	B05	MII0RXC/GRXC	G03	NC
AB20	GPIO1	P01	MEMCLK-	A05	MII0RXD0	G04	NC
Y20	GPIO2	P02	MEMCLK+	C05	MII0RXD1	H03	NC
Y22	GPIO3	G05	MEMCOMP	D05	MII0RXD2	AB07	NORA00/NANDCE0#
AB22	GPIO4	H05	MEMCS0#	A04	MII0RXD3	AA07	NORA01/NANDCE1#
AA22	GPIO5	Y01	MEMD00	B03	MII0RXDV	V08	NORA02/NANDCE2#
Y21	GPIO6	U03	MEMD01	C04	MII0RXER	W08	NORA03/NANDCE3#
AA21	GPIO7	W03	MEMD02	A06	MII0TXC	W07	NORA04/NANDCE4#
R20	I2C0SCL	U02	MEMD03	B07	MII0TXD0	AB06	NORA05/NANDCE5#
R22	I2C0SDA	Y02	MEMD04	A07	MII0TXD1	V06	NORA06/NANDCE6#
P22	I2C1SCL/KPADROW6	V04	MEMD05	D06	MII0TXD2	AA06	NORA07/NANDCE7#
R21	I2C1SDA/KPADROW7	AA01	MEMD06	C06	MII0TXD3	V09	NORA08/NANDRB0#
Y18	JTAGTCK	V01	MEMD07	C08	MII0TXEN	AB03	NORA09/NANDRB1#
AA18	JTAGTDI	N01	MEMD08	A17	MMCDATA4	T06	NORA10
AB18	JTAGTDO	U01	MEMD09	C17	MMCDATA5	U06	NORA11
AB19	JTAGTMS	R04	MEMD10	B18	MMCDATA6	Y04	NORA12
W18	JTAGTRST#	T03	MEMD11	C18	MMCDATA7	W06	NORA13
C12	KBCK	T04	MEMD12	B12	MSCK	W05	NORA14
C11	KBDT	N03	MEMD13	B11	MSDT	AA03	NORA15
H18	KPADROW5	P03	MEMD14	A01	NC	AB05	NORA16
J03	MEMA00	T01	MEMD15	B01	NC	Y06	NORA17
H01	MEMA01	W01	MEMDQM0#	B02	NC	AA05	NORA18
H02	MEMA02	R01	MEMDQM1#	C01	NC	AB04	NORA19
J04	MEMA03	V03	MEMDQS0-	C03	NC	Y05	NORA20
J02	MEMA04	V02	MEMDQS0+	D01	NC	AA04	NORA21
L01	MEMA05	R03	MEMDQS1-	D02	NC	V07	NORA22
K01	MEMA06	R02	MEMDQS1+	D03	NC	AB02	NORA23/GUPBE0#

Signal Ball List: WM8505 479-ball PBGA Package – Sorted by Signal Name (Continued)

Ball	Signal Name	Ball	Signal	Ball	Signal Name	Ball	Signal Name
AA02	NORA24/GUPBE1#	B10	RSMRST#	V17	UART2CTS/UART4RXD	E21	VDOUT07
W09	NORADV#	A15	RTCXI	U18	UART2RTS/UART4TXD	E20	VDOUT08
U07	NORBAA#	B15	RTCXO	V18	UART2RXD	E18	VDOUT09
Y11	NORCE0#/NANDCLE	F18	SDCDT	V19	UART2TXD	E17	VDOUT10
AB01	NORCE1#	A18	SDCLK	AB17	UART3CTS/UART5RXD	D22	VDOUT11
W04	NORCE2#	B17	SDCMD	Y17	UART3RTS/UART5TXD	D21	VDOUT12
Y03	NORCE3#	A19	SDDATA0	W17	UART3RXD	D19	VDOUT13
V05	NORCLK	D16	SDDATA1	AA17	UART3TXD	D17	VDOUT14
V13	NORD00/NANDIO00	B16	SDDATA2	E13	USBOC0#	C22	VDOUT15
AB12	NORD01/NANDIO01	C16	SDDATA3	D13	USBOC1#	C21	VDOUT16
AA12	NORD02/NANDIO02	E16	SDWP	E14	USBP0-	C20	VDOUT17
Y12	NORD03/NANDIO03	W13	SFCLK	D14	USBP0+	B22	VDOUT18
AB11	NORD04/NANDIO04	AB14	SFCS0#	B14	USBP1-	A22	VDOUT19
AA11	NORD05/NANDIO05	AA13	SFCS1#	C14	USBP1+	A21	VDOUT20
AA10	NORD06/NANDIO06	Y14	SFDI	A13	USBP2-	C19	VDOUT21
AB10	NORD07/NANDIO07	AA14	SFDO	B13	USBP2+	B20	VDOUT22
Y10	NORD08/NANDIO08	AA15	SPI0CLK	C15	USBREXT	A20	VDOUT23
V10	NORD09/NANDIO09	W14	SPI0MISO	A16	VBAT	G20	VDVSYNC
AB09	NORD10/NANDIO10	AB15	SPI0MOSI	K21	VCCA33DAC	K22	VGAB
AA09	NORD11/NANDIO11	V14	SPI0SS#	P21	VCLK/KPADROW2	J21	VGAG
AA08	NORD12/NANDIO12	W15	SPI1CLK	E19	VDCLK	H20	VGAHSYNC
AB08	NORD13/NANDIO13	Y15	SPI1MISO	G19	VDDEN	J22	VGAR
Y08	NORD14/NANDIO14	V15	SPI1MOSI	G18	VDHSYNC	J20	VGAREXT
Y07	NORD15/NANDIO15	AB16	SPI1SS#	R19	VDIN0/KPADCOL0	H21	VGAVSYNC
V11	NOROE#/NANDRE#	V16	SPI2CLK	R18	VDIN1/KPADCOL1	N18	VHSYNC/KPADROW0
W10	NORRST#/NANDALE	AA16	SPI2MISO	P19	VDIN2/KPADCOL2	P18	VVSYNC/KPADROW1
V12	NORWAIT#/NANDWPD#	W16	SPI2MOSI	M18	VDIN3/KPADCOL3	C09	WAKEUP0
W11	NORWE#/NANDWE#	U16	SPI2SS#	P20	VDIN4/KPADCOL4	D10	WAKEUP1/USBATTA
W12	NORWP#/NANDWP#	D12	SUSGPIO	N22	VDIN5/KPADCOL5	C10	WAKEUP2
D08	PHY25MHZ	AA19	TESTMODE	N21	VDIN6/KPADCOL6	B09	WAKEUP3
U17	PWMOUT0	T20	UART0CTS	N20	VDIN7/KPADCOL7		
T17	PWMOUT1	T19	UART0RTS	G17	VDOUT00		
A11	PWRBTN#	T22	UART0RXD	G22	VDOUT01		
A10	PWREN15	T21	UART0TXD	G21	VDOUT02		
A09	PWREN18	T18	UART1CTS	F16	VDOUT03		
A08	PWREN33	U20	UART1RTS	F22	VDOUT04		
A12	PWRGD	U22	UART1RXD	F20	VDOUT05		
Y19	RESETOUT#	U21	UART1TXD	E22	VDOUT06		

Signal Ball List: WM8505 Power and Ground Ball List

Table 3 – WM8505 Power and Ground Ball List

Ball #	Signal Name	Ball #	Signal Name	Ball #	Signal Name	Ball #	Signal Name
A14	GND	L10	GND	Y16	GND	R06	VCCMEM
B04	GND	L11	GND	L18	GNDAPLL0	G07	VDD
B06	GND	L12	GND	K18	GNDAPLL1	G08	VDD
B19	GND	L13	GND	L21	GNDAXTAL	G09	VDD
B21	GND	L14	GND	F07	VCC33	G10	VDD
C02	GND	L15	GND	F08	VCC33	G11	VDD
C13	GND	M08	GND	F09	VCC33	G12	VDD
D18	GND	M09	GND	F10	VCC33	G13	VDD
D20	GND	M10	GND	F11	VCC33	G14	VDD
E04	GND	M11	GND	F15	VCC33	G15	VDD
E6	GND	M12	GND	G16	VCC33	H07	VDD
E9	GND	M13	GND	H17	VCC33	H16	VDD
F02	GND	M14	GND	J17	VCC33	J07	VDD
F17	GND	M15	GND	K17	VCC33	J16	VDD
F19	GND	N02	GND	L17	VCC33	K07	VDD
F21	GND	N08	GND	M17	VCC33	K16	VDD
H04	GND	N09	GND	N17	VCC33	L07	VDD
H08	GND	N10	GND	P17	VCC33	L16	VDD
H09	GND	N11	GND	R17	VCC33	M07	VDD
H10	GND	N12	GND	T16	VCC33	M16	VDD
H11	GND	N13	GND	U08	VCC33	N07	VDD
H12	GND	N14	GND	U09	VCC33	N16	VDD
H13	GND	N15	GND	U10	VCC33	P07	VDD
H14	GND	N19	GND	U11	VCC33	P16	VDD
H15	GND	P04	GND	U12	VCC33	R07	VDD
H22	GND	P08	GND	U13	VCC33	R16	VDD
J08	GND	P09	GND	U14	VCC33	T07	VDD
J09	GND	P10	GND	U15	VCC33	T08	VDD
J10	GND	P11	GND	F12	VCC33USB	T09	VDD
J11	GND	P12	GND	F13	VCC33USB	T10	VDD
J12	GND	P13	GND	F14	VCC33USB	T11	VDD
J13	GND	P14	GND	E15	VCCA15USBPLL	T12	VDD
J14	GND	P15	GND	L22	VCCA33DAC	T13	VDD
J15	GND	R08	GND	M19	VCCA33PLL0	T14	VDD
K02	GND	R09	GND	L19	VCCA33PLL1	T15	VDD
K08	GND	R10	GND	H19	VCCA33PLLAB	E10	VSUS15
K09	GND	R11	GND	J19	VCCA33PLLCD	D11	VSUS33
K10	GND	R12	GND	D15	VCCA33USBPLL	E07	VSUS33
K11	GND	R13	GND	M20	VCCA33XTAL	E08	VSUS33
K12	GND	R14	GND	G06	VCCMEM	F06	VSUS33
K13	GND	R15	GND	H06	VCCMEM	E11	VSUS33
K14	GND	T02	GND	J06	VCCMEM	E12	VSUS33
K15	GND	U04	GND	K06	VCCMEM		
K20	GND	W02	GND	L06	VCCMEM		
L04	GND	W19	GND	M06	VCCMEM		
L08	GND	Y09	GND	N06	VCCMEM		
L09	GND	Y13	GND	P06	VCCMEM		

Signal Description

DRAM Interface

Signal Name	Ball #	I/O	Signal Description	Power Plane
MEMA[13:0]	N04, M04, M03, M01, M02, L02, J01, K01, L01, J02, J04, H02, H01, J03	O	Memory Address Bus	VCCMEM
MEMBA[2:0]	P05, M05, L03	O	Memory Bank Address Bus	VCCMEM
MEMDQM[1:0]#	R01, W01	O	DDR Data Mask for Byte 1-0; Active-low	VCCMEM
MEMCAS#	K04	O	Column Address – CAS; Active-low	VCCMEM
MEMCKE	N05	O	DDR Clock Enable	VCCMEM
MEMCLK+	P02	O	Differential Memory Clocks (+)	VCCMEM
MEMCLK-	P01	O	Differential Memory Clocks (-)	VCCMEM
MEMCS0#	H05	O	Memory Chip Selects 0; Active-low	VCCMEM
MEMRAS#	K03	O	Row Address – RAS; Active-low	VCCMEM
MEMWE#	K05	O	Write Enable; Active-low	VCCMEM
MEMVREF[1:0]	R05,J05	I	SSTL_2 I/O Reference Voltage	VCCMEM
MEMD[15:0]	T01, P03, N03, T04, T03, R04, U01, N01, V01, AA01, V04, Y02, U02, W03, U03, Y01	I/O	External Memory Data Bus	VCCMEM
MEMDQS[1:0]+	R02, V02	I/O	Differential DDR Data Strobe for Byte 1-0 (+)	VCCMEM
MEMDQS[1:0]-	R03, V03	I/O	Differential DDR Data Strobe for Byte 1-0 (+)	VCCMEM
MEMCOMP	G05	I	DDRCOMP Cell for PVT Adjustments	VCCMEM
MEMODT	L05	O	On-Die Termination	VCCMEM

Serial Flash Memory Controller Interface

Signal Name	Ball #	I/O	Signal Description	Power Plane
SFDI	Y14	I/O	Serial Flash Data In	VCC33
SFDO	AA14	O	Serial Flash Data Out	VCC33
SFCS[1:0]#	AA13, AB14	O	Serial Flash Chip Selects; Active-low	VCC33
SFCLK	W13	O	Serial Flash Clock	VCC33

Ethernet MAC Interface

Signal Name	Ball #	I/O	Signal Description	Power Plane
MII0RXC	B05	I/O	Receive Clock Input in MII Mode; Output in Reverse MII Mode	VCCMII
MII0RXD[3:0]	A04, D05, C05, A05	I	Receive Data	VCCMII
MII0RXDV	B03	I	Receive Data Valid in MII Mode	VCCMII
MII0RXER	C04	I	Receive Error	VCCMII
MII0TXC	A06	I/O	Transmit Clock Input in MII Mode	VCCMII
MII0TXD[3:0]	C06, D06, A07, B07	O	Transmit Data	VCCMII
MII0TXEN	C08	O	Transmit Enable	VCCMII
MII0COL	C07	I	Collision Detect	VCCMII
MII0CRS	D07	I	Carrier Sense	VCCMII
MII0MDIO	A02	I/O	Management Interface (MI) Data I/O	VCCMII
MII0MDC	A03	O	Management Interface (MI) Clock	VCCMII
MII0PHYRST#	D09	O	Ethernet PHY Reset; Active-low	VSUS33
MII0PHYPD#	B08	O	Ethernet PHY Power Down; Active-low	VSUS33
PHY25MHZ	D08	O	25 MHz Clock to Fast Ethernet PHY	VSUS33

USB 2.0 Host and Device Interface

Signal Name	Ball #	I/O	Signal Description	Power Plane
USBP[2:0]+	B13, C14, D14	I/O	USB D+; Internal Pull Down 15 kΩ Resistance	VCCU
USBP[2:0]-	A13, B14, E14	I/O	USB D-; Internal Pull Down 15 kΩ Resistance	VCCU
USBOC[1:0]#	D13, E13	I	Over Current Detection Input Internal Pull High 75 kΩ Resistance	VSUS33
USBATTA	D10	I	USB Device Attach Detect	VSUS33
USBREXT	C15	I/O	External 5.62 kΩ Resistance for BandGap Reference	VCCU
SUSGPIO	D12	I/O	Suspend Domain General Purpose IO, and can be Used to Control USB Bus Power.	VSUS33

PS/2 Keyboard Mouse Interface

Signal Name	Ball #	I/O	Signal Description	Power Plane
KBCK	C12	I/O	Keyboard Clock Input	VSUS33
KBDT	C11	I/O	Keyboard Serial Data Input/Output	VSUS33
MSCK	B12	I/O	Mouse Clock Input	VSUS33
MSDT	B11	I/O	Mouse Serial Data Input/Output	VSUS33

NAND Flash Interface

Signal Name	Ball #	I/O	Signal Description	Power Plane
NANDALE	W10	O	NAND Address Latch Enable	VCC33
NANDCLE	Y11	O	NAND Command Latch Enable	VCC33
NANDCE[7:0]#	AA06, V06, A B06, W07, W08, V08, AA07, AB07	O	NAND Chip Enable; Active-low	VCC33
NANDRE#	V11	O	NAND Read Enable	VCC33
NANDWE#	W11	O	NAND Write Enable	VCC33
NANDWP#	W12	O	NAND Write Protect	VCC33
NANDWPD#	V12	I	NAND Write Protect Detected	VCC33
NANDRB[1:0]#	AB03,V09	I/O	NAND Ready/Busy	VCC33
NANDIO[15:0]	Y07, Y08, AB08, AA08, AA09, AB09, V10, Y10, AB10, AA10, AA11, AB11, Y12, AA12, AB12, V13	I/O	NAND Command and Data I/O	VCC33

NOR Flash/Generic uP Interface

Signal Name	Ball #	I/O	Signal Description	Power Plane
NORRST#	W10	O	NOR Reset	VCC33
NORCE[3:0]#	Y03, W04, AB01, Y11	O	NOR Chip Select	VCC33
NORWE#	W11	O	NOR Write Enable	VCC33
NOROE#	V11	O	NOR Output Enable	VCC33
NORWP#	W12	O	NOR Write Protect	VCC33
NORWAIT#	V12	I	NOR Wait	
NORCLK	V05	O	NOR Synchronous Clock	VCC33
NORADV#	W09	O	NOR Address Valid	VCC33
NORBAA#	U07	O	NOR Burst Advance Acknowledge	VCC33
NORA[24:23]/ GUPBE[1:0]#	AA02,AB02	O	NOR Memory Address [24:23]; Generic uP Bus Extended Signals Byte Enable	VCC33
NORA[22:0]	V07, AA04, Y05, AB04, AA05, Y06, AB05, AA03, W05, W06, Y04, U06, T06, AB03, V09, AA06, V06, AB06, W07, W08, V08, AA07, AB07	O	NOR Memory Address [22:0]	VCC33
NORD[15:0]	Y07, Y08, AB08, AA08, AA09, AB09, V10, Y10, AB10, AA10, AA11, AB11, Y12, AA12, AB12, V13	I/O	NOR Data I/O [15:0]	VCC33

SD/MMC Interface

Signal Name	Ball #	I/O	Signal Description	Power Plane
SDCLK	A18	O	SD/MMC Bus Clock	VCC33
SDCMD	B17	I/O	SD/MMC Command	VCC33
SDWP	E16	I	SD/MMC Write Protect	VCC33
SDCDT	F18	I	SD Card Detect	VCC33
SDDATA[3:0]	C16, B16, D16, A19	I/O	SD/MMC Data Bus	VCC33
MMCDATA[7:4]	C18, B18, C17, A17	I/O	MMC Plus Data Bus	VCC33

DVO Interface

Signal Name	Ball #	I/O	Signal Description	Power Plane
VDOUT[23:0]	A20, B20, C19, A21, A22, B22, C20, C21, C22, D17, D19, D21, D22, E17, E18, E20, E21, E22, F20, F22, F16, G21, G22, G17	O	DVO Data Out	VCC33
VDCLK	E19	O	DVO Clock Out	VCC33
VDDEN	G19	O	DVO Data Enable	VCC33
VDHSYNC	G18	O	LCD Line (Horizontal Sync)	VCC33
VDVSYNC	G20	O	LCD Frame (Vertical Sync)	VCC33

Video Input Device (CCIR601/656) Interface

Signal Name	Ball #	I/O	Signal Description	Power Plane
VCLK	P21	I	Video Clock	VCC33
VDIN[7:0]	N20, N21, N22, P20, M18, P19, R18, R19	I	Video Digital Data Input	VCC33
VHSYNC	N18	I	Video Horizontal Sync	VCC33
VVSYNC	P18	I	Video Vertical Sync	VCC33

VGA Output Interface

Signal Name	Ball #	I/O	Signal Description	Power Plane
VGAR	J22	O	VGA Red Output	VCCA33DAC
VGAG	J21	O	VGA Green Output	VCCA33DAC
VGAB	K22	O	VGA Blue Output	VCCA33DAC
VGAREXT	J20	I	VGA REXT	VCCA33DAC
VGAHSYNC	H20	O	VGA Horizontal Sync	VCC33
VGAVSYNC	H21	O	VGA Vertical Sync	VCC33

Keypad Interface

Signal Name	Ball #	I/O	Signal Description	Power Plane
KPADROW[7:0]	R21, P22, H18, L20, J18, P21, P18, N18	I	Keypad Row Input	VCC33
KPADCOL[7:0]	N20, N21, N22, P20, M18, P19, R18, R19	I	Keypad Column Input	VCC33

Real Time Clock Interface

Signal Name	Ball #	I/O	Signal Description	Power Plane
RTCXI	A15	I	Real Time Clock Oscillator Crystal Input	VBAT
RTCXO	B15	O	Real Time Clock Oscillator Crystal Output	VBAT

Power Management Interface

Signal Name	Ball #	I/O	Signal Description	Power Plane
RSMRST#	B10	I	Resume Reset; Active-low This signal is a power management reset. It indicates that the SUS Power is good.	VSUS33
PWRGD	A12	I	Power Good This signal indicates that the Switch Power is good. It also serves as a reset to the normal logic.	VSUS33
PWREN33	A08	O	3.3V Power Enable	VSUS33
PWREN15	A10	O	1.5V Power Enable	VSUS33
PWREN18	A09	O	1.8V Power Enable	VSUS33
PWRBTN#	A11	I	Power Button; Active-low	VSUS33
WAKEUP[3:0]	B09, C10, D10, C09	I	Wakeup	VSUS33

UART Interface

Signal Name	Ball #	I/O	Signal Description	Power Plane
UART0RTS	T19	O	UART 0 Request to Send	VCC33
UART0TXD	T21	O	UART 0 Transmit Data	VCC33
UART0CTS	T20	I	UART 0 Clear to Send	VCC33
UART0RXD	T22	I	UART 0 Receive Data	VCC33
UART1RTS	U20	O	UART 1 Request to Send	VCC33
UART1TXD	U21	O	UART 1 Transmit Data	VCC33
UART1CTS	T18	I	UART 1 Clear to Send	VCC33
UART1RXD	U22	I	UART 1 Receive Data	VCC33
UART2RTS / UART4TXD	U18	O	UART 2 Request to Send; or UART 4 Transmit Data	VCC33
UART2TXD	V19	O	UART 2 Transmit Data	VCC33
UART2CTS / UART4RXD	V17	I	UART 2 Clear to Send; or UART 4 Receive Data	VCC33
UART2RXD	V18	I	UART 2 Receive Data	VCC33
UART3RTS / UART5TXD	Y17	O	UART 3 Request to Send; or UART 5 Transmit Data	VCC33
UART3TXD	AA17	O	UART 3 Transmit Data	VCC33
UART3CTS / UART5RXD	AB17	I	UART 3 Clear to Send; or UART 5 Receive Data	VCC33
UART3RXD	W17	I	UART 3 Receive Data	VCC33

Pulse Width Modulation Interface

Signal Name	Ball #	I/O	Signal Description	Power Plane
PWMOUT[1:0]	T17, U17	O	Pulse Width Modulation Timer Output	VCC33

SPI Interface

Signal Name	Ball #	I/O	Signal Description	Power Plane
SPI0CLK	AA15	I/O	SPI 0 Clock Output in Master Mode; Input in Slave Mode	VCC33
SPI0MISO	W14	I/O	SPI 0 Master IN; Slave OUT	VCC33
SPI0MOSI	AB15	I/O	SPI 0 Master OUT; Slave IN	VCC33
SPI0SS#	V14	I/O	SPI 0 Slave Select Output in Master Mode, Input in Slave Mode Active_low	VCC33
SPI1CLK	W15	I/O	SPI 1 Clock Output in Master Mode; Input in Slave Mode	VCC33
SPI1MISO	Y15	I/O	SPI 1 Master IN; Slave OUT	VCC33
SPI1MOSI	V15	I/O	SPI 1 Master OUT; Slave IN	VCC33
SPI1SS#	AB16	I/O	SPI 1 Slave Select Output in Master Mode; Input in Slave Mode Active_low	VCC33
SPI2CLK	V16	I/O	SPI 2 Clock Output in Master Mode; Input in Slave Mode	VCC33
SPI2MISO	AA16	I/O	SPI 2 Master IN, Slave OUT	VCC33
SPI2MOSI	W16	I/O	SPI 2 Master OUT, Slave IN	VCC33
SPI2SS#	U16	I/O	SPI 2 Slave Select Output in Master Mode; Input in Slave Mode Active-low	VCC33

I^{2}C Interface

Signal Name	Ball #	I/O	Signal Description	Power Plane
I2C0SCL	R20	I/O	I^{2}C 0 Serial Clock	VCC33
I2C0SDA	R22	I/O	I^{2}C 0 Serial Data	VCC33
I2C1SCL	P22	I/O	I^{2}C 1 Serial Clock	VCC33
I2C1SDA	R21	I/O	I^{2}C 1 Serial Data	VCC33

AC97 Interface

Signal Name	Ball #	I/O	Signal Description	Power Plane
AC97BCLK	W22	I	AC97 Bit Clock Input	VCC33
AC97SDO	V21	O	AC97 Serial Data Out	VCC33
AC97SDI	V20	I	AC97 Serial Data In	VCC33
AC97RST#	U19	O	AC97 Reset	VCC33
AC97SYNC	V22	O	AC97 Sync.	VCC33

Audio I^{2}S Interface

Signal Name	Ball #	I/O	Signal Description	Power Plane
I2SMCLK	U19	I/O	I^{2}S Main Clock Input/Output	VCC33
I2SWS	V22	O	I^{2}S Output Interface L/R Clock	VCC33
I2SCLK	W22	O	I^{2}S Output Interface Bit Clock	VCC33
I2SDIN	V20	I	I^{2}S Interface Serial Data Input	VCC33
I2SDOUT	V21	O	I^{2}S Output Interface Serial Data Output	VCC33

ARM JTAG Interface

Signal Name	Ball #	I/O	Signal Description	Power Plane
JTAGTDO	AB18	O	JTAG Test Data Output	VCC33
JTAGTCK	Y18	I	JTAG Test Clock	VCC33
JTAGTDI	AA18	I	JTAG Test Data Input	VCC33
JTAGTMS	AB19	I	JTAG Test Mode Select	VCC33
JTAGTRST#	W18	I	JTAG Test Reset Active-low	VCC33

Clock Support Interface

Signal Name	Ball #	I/O	Signal Description	Power Plane
27MHZXI	M22	I	Peripheral 27 MHz Crystal Input	VCCA33XTAL
27MHZXO	M21	O	Peripheral 27 MHz Crystal Output	VCCA33XTAL
24MHZXI	K19	I	Peripheral Clock Input and may be 24 MHz or 25 MHz	VCC33
CLKTST	J18	O	Clock Out from Internal CLKGEN	VCC33
CLKOUT	L20	O	Clock Out from Internal PLLs	VCC33

Miscellaneous Interface

Signal Name	Ball #	I/O	Signal Description	Power Plane
RESETOUT#	Y19	O	Reset Out, Active-low	VCC33
CIRIN	AA20	I	CIR Input	VCC33
TESTMODE	AA19	I	Test Mode Input	VCC33
GPIO[7:0]	AA21, Y21, AA22, AB22, Y22, Y20, AB20, AB21	I/O	Dedicated General Purpose Input/Output	VCC33

General Purpose Input/Output Interface

Signal Name	Ball #	I/O	Signal Description	Power Plane
GPIO[7:0]	AA21, Y21, AA22, AB22, Y22, Y20, AB20, AB21	I/O	Dedicated General Purpose Input/Output	VCC33
GPIO_VDOUT[23:0]	A20, B20, C19, A21, A22, B22, C20, C21, C22, D17, D19, D21, D22, E17, E18, E20, E21, E22, F20, F22, F16, G21, G22, G17	I/O	General Purpose Input/Output	VCC33
GPIO_VDHSYNC	G18	I/O	General Purpose Input/Output	VCC33
GPIO_VDVSYNC	G20	I/O	General Purpose Input/Output	VCC33
GPIO_VDIN[7:0]	N20, N21, N22, P20, M18, P19, R18, R19	I/O	General Purpose Input/Output	VCC33
GPIO_VHSYNC	N18	I/O	General Purpose Input/Output	VCC33
GPIO_VVSYNC	P18	I/O	General Purpose Input/Output	VCC33
GPIO_VGAHSYNC	H20	I/O	General Purpose Input/Output	VCC33
GPIO_VGAVSYNC	H21	I/O	General Purpose Input/Output	VCC33
GPIO_AC97BCLK	W22	I/O	General Purpose Input/Output	VCC33
GPIO_AC97SDO	V21	I/O	General Purpose Input/Output	VCC33
GPIO_AC97SDI	V20	I/O	General Purpose Input/Output	VCC33
GPIO_AC97RST#	U19	I/O	General Purpose Input/Output	VCC33
GPIO_AC97SYNC	V22	I/O	General Purpose Input/Output	VCC33
GPIO_NORA[22:0]	V07, AA04, Y05, AB04, AA05, Y06, AB05, AA03, W05, W06, Y04, U06, T06, AB03, V09, AA06, V06, AB06, W07, W08, V08, AA07, AB07	I/O	General Purpose Input/Output	VCC33
GPIO_NORD[15:0]	Y07, Y08, AB08, AA08, AA09, AB09, V10, Y10, AB10, AA10, AA11, AB11, Y12, AA12, AB12, V13	I/O	General Purpose Input/Output	VCC33
GPIO_SFDI	Y14	I/O	General Purpose Input/Output	VCC33
GPIO_SFDO	AA14	I/O	General Purpose Input/Output	VCC33
GPIO_SFCS[1:0]#	AA13, AB14	I/O	General Purpose Input/Output	VCC33
GPIO_SFCLK	W13	I/O	General Purpose Input/Output	VCC33

Signal Name	Ball #	I/O	Signal Description	Power Plane
GPIO_SDDATA[3:0]	C16, B16, D16, A19	I/O	General Purpose Input/Output	VCC33
GPIO_MMCDATA[7:4]	C18, B18, C17, A17	I/O	General Purpose Input/Output	VCC33
GPIO_UART0RTS	T19	I/O	General Purpose Input/Output	VCC33
GPIO_UART0TXD	T21	I/O	General Purpose Input/Output	VCC33
GPIO_UART0CTS	T20	I/O	General Purpose Input/Output	VCC33
GPIO_UART0RXD	T22	I/O	General Purpose Input/Output	VCC33
GPIO_UART1RTS	U20	I/O	General Purpose Input/Output	VCC33
GPIO_UART1TXD	U21	I/O	General Purpose Input/Output	VCC33
GPIO_UART1CTS	T18	I/O	General Purpose Input/Output	VCC33
GPIO_UART1RXD	U22	I/O	General Purpose Input/Output	VCC33
GPIO_UART2RTS	U18	I/O	General Purpose Input/Output	VCC33
GPIO_UART2TXD	V19	I/O	General Purpose Input/Output	VCC33
GPIO_UART2CTS	V17	I/O	General Purpose Input/Output	VCC33
GPIO_UART2RXD	V18	I/O	General Purpose Input/Output	VCC33
GPIO_UART3RTS	Y17	I/O	General Purpose Input/Output	VCC33
GPIO_UART3TXD	AA17	I/O	General Purpose Input/Output	VCC33
GPIO_UART3CTS	AB17	I/O	General Purpose Input/Output	VCC33
GPIO_UART3RXD	W17	I/O	General Purpose Input/Output	VCC33
GPIO_SPI0CLK	AA15	I/O	General Purpose Input/Output	VCC33
GPIO_SPI0MISO	W14	I/O	General Purpose Input/Output	VCC33
GPIO_SPI0MOSI	AB15	I/O	General Purpose Input/Output	VCC33
GPIO_SPI0SS#	V14	I/O	General Purpose Input/Output	VCC33
GPIO_SPI1CLK	W15	I/O	General Purpose Input/Output	VCC33
GPIO_SPI1MISO	Y15	I/O	General Purpose Input/Output	VCC33
GPIO_SPI1MOSI	V15	I/O	General Purpose Input/Output	VCC33
GPIO_SPI1SS#	AB16	I/O	General Purpose Input/Output	VCC33
GPIO_SPI2CLK	V16	I/O	General Purpose Input/Output	VCC33
GPIO_SPI2MISO	AA16	I/O	General Purpose Input/Output	VCC33

Signal Name	Ball #	I/O	Signal Description	Power Plane
GPIO_SPI2MOSI	W16	I/O	General Purpose Input/Output	VCC33
GPIO_SPI2SS#	U16	I/O	General Purpose Input/Output	VCC33
SUSGPIO	D12	I/O	General Purpose Input/Output	VSUS33
GPIO_WAKEUP[3:0]	B09, C10, D10, C09	I/O	General Purpose Input/Output	VSUS33

Power and Ground

Signal Name	Ball #	# of Balls	Signal Description
VCCA33XTAL	M20	1	Crystal/OSC Power (3.3V)
GNDAXTAL	L21	1	Crystal/OSC Ground
VCCA33PLL[1:0]	L19, M19	2	Primary PLL Analog Power (3.3V)
GNDAPLL[1:0]	K18, L18	2	Primary PLL Analog Ground
VCCA33PLLAB	H19	1	Secondary PLL Analog Power (3.3V)
VCCA33PLLCD	J19	1	Secondary PLL Analog Power (3.3V)
VCCA33DAC	L22	1	VGA DAC Analog Power (3.3V)
VBAT	A16	1	RTC Battery Power (3.3V)
VCCA33USBPLL	D15	1	USB PLL Analog Power (3.3V)
VCCA15USBPLL	E15	1	USB PLL Analog Power (1.5V)
VCC33USB	F12, F13, F14	3	USB Differential Output Power (3.3V)
VSUS33	D11, E11, E07, E08, F06	5	Suspend IO Power (3.3V)
VSUS15	E10	1	Suspend Core Power (1. 5V)
VCCMEM	G06, H06, J06, K06, L06, M06, N06, P06, R06	9	Memory I/O Power (1.8V)
VCC33	F07, F08, F09, F10, F11, F15, G16, H17, J17, K17, L17, M17, N17, P17, R17, T16, U08, U09, U10, U11, U12, U13, U14, U15	24	IO Power (3.3V)
VDD	G07, G08, G09, G10, G11, G12, G13, G14, G15, H07, H16, J07, J16, K07, K16, L07, L16, M07, M16, N07, N16, P07, P16, R07, R16, T07, T08, T09, T10, T11, T12, T13, T14, T15	34	Core Power (1.5V)
GND	A14, B04, B06, B19, B21, C02, C13, D18, D20, E04, E06, E09, F02, F17, F19, F21, H04, H08, H09, H10, H11, H12, H13, H14, H15, H22, J08, J09, J10, J11, J12, J13, J14, J15, K02, K08, K09, K10, K11, K12, K13, K14, K15, K20, L04, L08, L09, L10, L11, L12, L13, L14, L15, M08, M09, M10, M11, M12, M13, M14, M15, N02, N08, N09, N10, N11, N12, N13, N14, N15, N19, P04, P08, P09, P10, P11, P12, P13, P14, P15, R08, R09, R10, R11, R12, R13, R14, R15, T02, U04, W02, W19, Y09, Y13, Y16	95	Common Ground Plane

Register Overview

Memory Map

Table 4 summarizes the registers of all controllers of WM8505, listed in the order of address range in the memory map, lowest addresses first. Detailed descriptions of each register bit and field are provided following the summary table in the same order.

Table 4 - WM8505 Memory Map

Address	Register	Size in Bytes
0xD800:0000 – 0xD800:03FF	Reserved	1k
0xD800:0400 – 0xD800:07FF	DDR2 DRAM Controller Registers	1k
0xD800:0800 – 0xD800:0FFF	Reserved	2k
0xD800:1000 – 0xD800:17FF	Reserved	2k
0xD800:1800 – 0xD800:1BFF	System DMA Configuration Registers	1k
0xD800:1C00 – 0xD800:1FFF	VDMA Configuration Registers	1k
0xD800:1400 – 0xD800:1FFF	Reserved	3k
0xD800:2000 – 0xD800:23FF	Serial Flash Memory Controller Registers	1k
0xD800:2400 – 0xD800:2FFF	Reserved	3k
0xD800:3000 – 0xD800:33FF	Reserved	1k
0xD800:3400 – 0xD800:3FFF	Reserved	3k
0xD800:4000 – 0xD800:43FF	Ethernet MAC Configuration Registers	1k
0xD800:4400 – 0xD800:4FFF	Reserved	3k
0xD800:5000 – 0xD800:53FF	Reserved	1k
0xD800:5400 – 0xD800:5FFF	Reserved	3k
0xD800:6000 – 0xD800:6FFF	Cipher Control Register	4k
0xD800:7000 – 0xD800:7FFF	USB 2.0 Host Controller Registers	2k
0xD800:8000 – 0xD800:87FF	Reserved	2k
0xD800:8800 – 0xD800:8BFF	PS/2 Control Registers	1k
0xD800:8C00 – 0xD800:8FFF	Reserved	2k
0xD800:9000 – 0xD800:93FF	NAND Flash Controller Registers	1k
0xD800:9400 – 0xD800:97FF	NOR Flash / uP Bus Controller Registers	1k
0xD800:9800 – 0xD800:9FFF	USB 2.0 Device Controller Registers	2k
0xD800:A000 – 0xD800:A3FF	SD/SDIO/MMC Controller Registers	1k
0xD800:A400 – 0xD800:AFFF	Reserved	3k
0xD800:B000 – 0xD800:B3FF	Reserved	1k
0xD800:B400 – 0xD800:BFFF	Reserved	3k
0xD800:C000 – 0xD800:C3FF	Reserved	1k
0xD800:C400 – 0xD800:CFFF	Reserved	3k
0xD800:D000 – 0xD800:D7FF	Reserved	2k
0xD800:D800 – 0xD800:DFFF	Reserved	2k
0xD800:E000 – 0xD800:E3FF	Reserved	1k
0xD800:E400 – 0xD800:E7FF	Reserved	1k
0xD800:E800 – 0xD800:EFFF	Reserved	2k
0xD800:F000 – 0xD800:FFFF	Reserved	4k
0xD801:0000 – 0xD801:FFFF	Reserved	64k
0xD802:0000 – 0xD802:FBFF	Reserved	63k
0xD802:FC00 – 0xD802:FFFF	Async. APB Bridge Control Registers	1k
0xD803:0000 – 0xD803:FFFF	Reserved	64k
0xD804:0000 – 0xD804:FFFF	Reserved	64k
0xD805:0000 – 0xD805:00FF	Reserved	256
0xD805:0100 – 0xD805:02FF	Reserved	512
0xD805:0300 – 0xD805:03FF	GOVW Control Registers	256
0xD805:0400 – 0xD805:06FF	GE and GE Alpha Mixing Control Registers	768
0xD805:0700 – 0xD805:07FF	Reserved	256
0xD805:0800 – 0xD805:08FF	GOVR_HD1 / DV Control Registers	256
0xD805:0900 – 0xD805:09FF	GOVR_HD2 / DV Control Registers	256
0xD805:0A00 – 0xD805:0AFF	VID (CCIR-656 Interface) Control Registers	256

Address	Register	Size in Bytes
0xD805:0B00 – 0xD805:0BFF	Reserved	256
0xD805:0C00 – 0xD805:0CFF	Reserved	256
0xD805:0D00 – 0xD805:0DFF	SCL444 Control Registers	256
0xD805:0E00 – 0xD805:0EFF	Reserved	256
0xD805:0F00 – 0xD805:0FFF	VPP Global Control Registers	256
0xD805:1000 – 0xD805:FFFF	Reserved	60k
0xD806:0000 – 0xD806:FFFF	Reserved	64k
0xD807:0000 – 0xD807:FFFF	Reserved	64k
0xD808:0000 – 0xD808:FFFF	Reserved	64k
0xD809:0000 – 0xD809:FFFF	Reserved	64k
0xD80A:0000 – 0xD80A:FFFF	Reserved	64k
0xD80B:0000 – 0xD80B:FFFF	Reserved	64k
0xD80C:0000 – 0xD80C:FFFF	Reserved	64k
0xD80D:0000 – 0xD80D:FFFF	Reserved	64k
0xD80E:0000 – 0xD80E:FFFF	Reserved	64k
0xD80F:0000 – 0xD80F:DFFF	Reserved	56k
0xD80F:E000 – 0xD80F:EFFF	JPEG Decoder Control Registers	4k
0xD80F:F000 – 0xD80F:FFFF	Reserved	4k
0xD810:0000 – 0xD810:FFFF	Real-Time Clock (RTC) Control Registers	64k
0xD811:0000 – 0xD811:FFFF	GPIO Control Registers	64k
0xD812:0000 – 0xD812:FFFF	System Configuration Control Registers	64k
0xD813:0000 – 0xD813:FFFF	Power Management Control Registers	64k
0xD814:0000 – 0xD814:FFFF	Interrupt Controller 1 Registers	64k
0xD815:0000 – 0xD815:FFFF	Interrupt Controller 2 Registers	64k
0xD816:0000 – 0xD81F:FFFF	Reserved	640k
0xD820:0000 – 0xD820:FFFF	UART 0 Control Registers	64k
0xD821:0000 – 0xD821:FFFF	UART 2 Control Registers	64k
0xD822:0000 – 0xD822:FFFF	PWM Control Registers	64k
0xD823:0000 – 0xD823:FFFF	Reserved	64k
0xD824:0000 – 0xD824:FFFF	SPI 0 Control Registers	64k
0xD825:0000 – 0xD825:FFFF	SPI 1 Control Registers	64k
0xD826:0000 – 0xD826:FFFF	Keypad Control Registers	64k
0xD827:0000 – 0xD827:FFFF	CIR Receiver Control Registers	64k
0xD828:0000 – 0xD828:FFFF	I^{2}C 0 Control Registers	64k
0xD829:0000 – 0xD829:FFFF	AC97 Controller Registers	64k
0xD82A:0000 – 0xD82A:FFFF	SPI 2 Control Registers	64k
0xD82B:0000 – 0xD82B:FFFF	UART1 Control Registers	64k
0xD82C:0000 – 0xD82C:FFFF	UART3 Control Registers	64k
0xD82D:0000 – 0xD82D:FFFF	Reserved	64k
0xD82E:0000 – 0xD82F:FFFF	Reserved	128k
0xD830:0000 – 0xD830:FFFF	Reserved	64k
0xD831:0000 – 0xD831:FFFF	Reserved	64k
0xD832:0000 – 0xD832:FFFF	I^{2}C1 Control Registers	64k
0xD833:0000 – 0xD833:FFFF	I^{2}S Control Registers	64k
0xD834:0000 – 0xD834:FFFF	Reserved	64k
0xD835:0000 – 0xD835:FFFF	Reserved	64k
0xD836:0000 – 0xD836:FFFF	Reserved	64k
0xD837:0000 – 0xD837:FFFF	UART4 Control Registers	64k
0xD838:0000 – 0xD838:FFFF	UART5 Control Registers	64k

Abbreviation

Attribute Definitions

RO: Read Only.
WO: Write Only. (The register value can not be read by the software.)
RW: Read / Write.
RW1: Software can Read and "Write 1" to the register. "Write 0" has no effect.
RW1C: Read / Write of "1" clears bit to zero.
RWE: Software can Read and Write on it. It can also be updated from Virtual/Serial EEPROM
RWU: Software can Read and Write on it. It will be updated by hardware itself.
WR0: Software can write on it. But it is Read as 0's always.
RE: Software can Read on it. It can be updated from Virtual/Serial EEPROM.
R0: Read as 0 always.
SC: Software Driver Control.
HC: Hardware Control.
E: Updated by EEPROM Loading.
U: Updated by HC.
C/S: Clear or set by HC.
R: Can be Read by software.
W: Written by software.
W0: Software writes "1" and clear.
W1: Software writes "1" and set.

The attribute of a register is composed of the above primitive attributes.
RW0S: Software can read the register. Software can also "Write 1 to clear" it. "Write 0" has no effect.
RW1S: Software can read the register. Software can also "Write 1 to set" it. "Write 0" has no effect.
RU: Read Only. The "U" indicates that the register can be updated by hardware.
R0WC: Read as 0 always. Software can "Write 1" to trigger a specific event.

Register Descriptions

GPIO Control Registers

The GPIO module of WM8505 contains all of the control and status registers associated with General Purpose I/O functionality.
The associated registers will be accessed via the 64 k-byte address range assigned to the GPIO module. The apbb_gpio_sel signal will go active (a one) whenever this 64 k-byte address range is decoded, and the GPIO register will be accessed in accordance to the following address criteria:

Base address: 0xD811:0000

Offset 0x0040 – 0x0043

GPIO Enable Control Register for SD/MMC Signals

This register provides the software with the means to enable/disable the associated SD/MMC data signals to operate as General Purpose I/Os.

Bit	Attribute	Default	Description
31:8	RO	0	Reserved
7	RW	0	MMCDATA7 Signal GPIO Enable. 0: The MMCDATA7 signal will operate normally as the MMCDATA7 signal. 1: The MMCDATA7 signal will operate as a General Purpose I/O.
6	RW	0	MMCDATA6 Signal GPIO Enable. 0: The MMCDATA6 signal will operate normally as the MMCDATA6 signal. 1: The MMCDATA6 signal will operate as a General Purpose I/O.
5	RW	0	MMCDATA5 Signal GPIO Enable. 0: The MMCDATA5 signal will operate normally as the MMCDATA5 signal. 1: The MMCDATA5 signal will operate as a General Purpose I/O.
4	RW	0	MMCDATA4 Signal GPIO Enable. 0: The MMCDATA4 signal will operate normally as the MMCDATA4 signal. 1: The MMCDATA4 signal will operate as a General Purpose I/O.
3	RW	0	SDDATA3 Signal GPIO Enable. 0: The SDDATA3 signal will operate normally as the SDDATA3 signal. 1: The SDDATA3 signal will operate as a General Purpose I/O.
2	RW	0	SDDATA2 signal GPIO Enable. 0: The SDDATA2 signal will operate normally as the SDDATA2 signal. 1: The SDDATA2 signal will operate as a General Purpose I/O.
1	RW	0	SDDATA1 Signal GPIO Enable. 0: The SDDATA1 signal will operate normally as the SDDATA1 signal. 1: The SDDATA1 signal will operate as a General Purpose I/O.
0	RW	0	SDDATA0 Signal GPIO Enable. 0: The SDDATA0 signal will operate normally as the SDDATA0 signal. 1: The SDDATA0 signal will operate as a General Purpose I/O.

Offset 0x0044 – 0x0047

GPIO Enable Control Register for VDIN/VDOUT Signals

This register provides the software with the means to enable/disable the associated VDIN/VDOUT data signals to operate as General Purpose I/Os.

Bit	Attribute	Default	Description
31:8	RW	0	VDOUT[23:0] signal GPIO Enable. 0: The VDOUT[23:0] signal will operate normally as the VDOUT signal. 1: The VDOUT[23:0] signal will operate as a General Purpose I/O.
7:0	RW	0	VDIN[23:0] signal GPIO Enable. 0: The VDIN[7:0] signal will operate normally as the VDIN signal. 1: The VDIN[7:0] signal will operate as a General Purpose I/O.

Offset 0x0048 – 0x004B

GPIO Enable Control Register for SYNC Signals

The purpose of this register is to provide software with the means to enable/disable the associated SYNC signals to operate as General Purpose I/Os.

Bit	Attribute	Default	Description
31:6	RO	0	Reserved
5	RW	0	VDVSYNC Signal GPIO Enable. 0: The VDVSYNC signal will operate normally as the VDVSYNC signal. 1: The VDVSYNC signal will operate as a General Purpose I/O.
4	RW	0	VDHSYNC Signal GPIO Enable. 0: The VDHSYNC signal will operate normally as the VDHSYNC signal. 1: The VDHSYNC signal will operate as a General Purpose I/O.
3	RW	0	VGAVSYNC Signal GPIO Enable. 0: The VGAVSYNC signal will operate normally as the VGAVSYNC signal. 1: The VGAVSYNC signal will operate as a General Purpose I/O.
2	RW	0	VGAHSYNC Signal GPIO Enable. 0: The VGAHSYNC signal will operate normally as the VGAHSYNC signal. 1: The VGAHSYNC signal will operate as a General Purpose I/O.
1	RW	0	VVSYNC Signal GPIO Enable. 0: The VVSYNC signal will operate normally as the VVSYNC signal. 1: The VVSYNC signal will operate as a General Purpose I/O.
0	RW	0	VHSYNC Signal GPIO Enable. 0: The VHSYNC signal will operate normally as the VHSYNC signal. 1: The VHSYNC signal will operate as a General Purpose I/O.

Offset 0x004C – 0x004F

GPIO Enable Control Register for NORD Signals

This register provides the software with the means to enable/disable the associated NORD data signals to operate as General Purpose I/Os.

Bit	Attribute	Default	Description
31:16	RO	0	Reserved
15:0	RW	0	NORD Data Signal GPIO Enable. 0: The NORD[15:0] signal will operate normally as the NOR data signal. 1: The NORD[15:0] signal will operate as a General Purpose I/O.

Offset 0x0050 – 0x0053

GPIO Enable Control Register for NORA Signals

This register provides the software with the means to enable/disable the associated NORA signals to operate as General Purpose I/Os.

Bit	Attribute	Default	Description
31:25	RO	0	Reserved
24:0	RW	0	NORA Data Signal GPIO Enable. 0: The NORA[24:0] signal will operate normally as the NOR address signal. 1: The NORA[24:0] signal will operate as a General Purpose I/O.

Offset 0x0054 – 0x0057

GPIO Enable Control Register for AC97 Signals

This register provides the software with the means to enable/disable the associated AC97 signals to operate as General Purpose I/Os.

Bit	Attribute	Default	Description
31:5	RO	0	Reserved
4	RW	0	AC97RST Signal GPIO Enable. 0: The AC97RST signal will operate normally as the AC97RST signal. 1: The AC97RST signal will operate as a General Purpose I/O.
3	RW	0	AC97BCLK Signal GPIO Enable. 0: The AC97BCLK signal will operate normally as the AC97BCLK. 1: The AC97BCLK signal will operate as a General Purpose I/O.
2	RW	0	AC97SDO Signal GPIO Enable. 0: The AC97SDO signal will operate normally as the AC97SDO signal. 1: The AC97SDO signal will operate as a General Purpose I/O.
1	RW	0	AC97SYNC Signal GPIO Enable. 0: The AC97SYNC signal will operate normally as the AC97SYNC signal. 1: The AC97SYNC signal will operate as a General Purpose I/O.
0	RW	0	AC97SDI Signal GPIO Enable. 0: The AC97SDI signal will operate normally as the AC97SDI signal. 1: The AC97SDI signal will operate as a General Purpose I/O.

Offset 0x0058 – 0x005B

GPIO Enable Control Register for SPI Flash Signals

This register provides the software with the means to enable/disable the associated SF signals to operate as General Purpose I/Os.

Bit	Attribute	Default	Description
31:5	RO	0	Reserved
4	RW	0	SFDI Signal GPIO Enable. 0: The SFDI signal will operate normally as the SFDI signal. 1: The SFDI signal will operate as a General Purpose I/O.
3	RW	0	SFCLK Signal GPIO Enable. 0: The SFCLK signal will operate normally as the SFCLK signal. 1: The SFCLK signal will operate as a General Purpose I/O.
2	RW	0	SFCS1 Signal GPIO Enable. 0: The SFCS1 signal will operate normally as the SFCS1 signal. 1: The SFCS1 signal will operate as a General Purpose I/O.
1	RW	0	SFCS0 Signal GPIO Enable. 0: The SFCS0 signal will operate normally as the SFCS0 signal. 1: The SFCS0 signal will operate as a General Purpose I/O.
0	RW	0	SFDO Signal GPIO Enable. 0: The SFDO signal will operate normally as the SFDO signal. 1: The SFDO signal will operate as a General Purpose I/O.

Offset 0x005C – 0x005F

GPIO Enable Control Register for SPI Signals

This register provides the software with the means to enable/disable the associated SPI signals to operate as General Purpose I/Os.

Bit	Attribute	Default	Description
31:12	RO	0	Reserved
11	RW	0	SPI2SS Signal GPIO Enable. 0: The SPI2SS signal will operate normally as the SPI2SS signal. 1: The SPI2SS signal will operate as a General Purpose I/O.
10	RW	0	SPI2MOSI Signal GPIO Enable. 0: The SPI2MOSI signal will operate normally as the SPI2MOSI signal. 1: The SPI2MOSI signal will operate as a General Purpose I/O.
9	RW	0	SPI2MISO Signal GPIO Enable. 0: The SPI2MISO signal will operate normally as the SPI2MISO signal. 1: The SPI2MISO signal will operate as a General Purpose I/O.
8	RW	0	SPI2CLK Signal GPIO Enable. 0: The SPI2CLK signal will operate normally as the SPI2CLK signal. 1: The SPI2CLK signal will operate as a General Purpose I/O.
7	RW	0	SPI1SS Signal GPIO Enable. 0: The SPI1SS signal will operate normally as the SPI1SS signal. 1: The SPI1SS signal will operate as a General Purpose I/O.
6	RW	0	SPI1MOSI Signal GPIO Enable. 0: The SPI1MOSI signal will operate normally as the SPI1MOSI signal. 1: The SPI1MOSI signal will operate as a General Purpose I/O.

5	RW	0	SPI1MISO Signal GPIO Enable. 0: The SPI1MISO signal will operate normally as the SPI1MISO signal. 1: The SPI1MISO signal will operate as a General Purpose I/O.
4	RW	0	SPI1CLK Signal GPIO Enable. 0: The SPI1CLK signal will operate normally as the SPI1CLK signal. 1: The SPI1CLK signal will operate as a General Purpose I/O.
3	RW	0	SPI0SS Signal GPIO Enable 0: The SPI0SS signal will operate normally as the SPI0SS signal. 1: The SPI0SS signal will operate as a General Purpose I/O
2	RW	0	SPI0MOSI Signal GPIO Enable 0: The SPI0MOSI signal will operate normally as the SPI0MOSI signal. 1: The SPI0MOSI signal will operate as a General Purpose I/O.
1	RW	0	SPI0MISO Signal GPIO Enable 0: The SPI0MISO signal will operate normally as the SPI0MISO signal. 1: The SPI0MISO signal will operate as a General Purpose I/O.
0	RW	0	SPI0CLK Signal GPIO Enable 0: The SPI0CLK signal will operate normally as the SPI0CLK signal. 1: The SPI0CLK signal will operate as a General Purpose I/O.

Offset 0x0060 – 0x0063

GPIO Enable Control Register for UART Signals

This register provides the software with the means to enable/disable the associated UART signals to operate as General Purpose I/Os.

Bit	Attribute	Default	Description
31:16	RO	0	Reserved
15	RW	0	UART3RXD Signal GPIO Enable. 0: The UART3RXD signal will operate normally as the UART3RXD signal. 1: The UART3RXD signal will operate as a General Purpose I/O.
14	RW	0	UART3CTS Signal GPIO Enable. 0: The UART3RXD signal will operate normally as the UART3RXD signal. 1: The UART3RXD signal will operate as a General Purpose I/O.
13	RW	0	UART3TXD Signal GPIO Enable. 0: The UART3TXD signal will operate normally as the UART3TXD signal. 1: The UART3TXD signal will operate as a General Purpose I/O.
12	RW	0	UART3RTS Signal GPIO Enable. 0: The UART3RTS signal will operate normally as the UART3RTS signal. 1: The UART3RTS signal will operate as a General Purpose I/O.
11	RW	0	UART2RXD Signal GPIO Enable. 0: The UART2RXD signal will operate normally as the UART2RXD signal. 1: The UART2RXD signal will operate as a General Purpose I/O.
10	RW	0	UART2CTS Signal GPIO Enable. 0: The UART2CTS signal will operate normally as the UART2CTS signal. 1: The UART2CTS signal will operate as a General Purpose I/O.

9	RW	0	UART2TXD Signal GPIO Enable. 0: The UART2TXD signal will operate normally as the UART2TXD signal. 1: The UART2TXD signal will operate as a General Purpose I/O.
8	RW	0	UART2RTS Signal GPIO Enable. 0: The UART2RTS signal will operate normally as the UART2RTS signal. 1: The UART2RTS signal will operate as a General Purpose I/O.
7	RW	0	UART1RXD Signal GPIO Enable. 0: The UART1RXD signal will operate normally as the UART1RXD signal. 1: The UART1RXD signal will operate as a General Purpose I/O.
6	RW	0	UART1CTS Signal GPIO Enable. 0: The UART1CTS signal will operate normally as the UART1CTS signal. 1: The UART1CTS signal will operate as a General Purpose I/O.
5	RW	0	UART1TXD Signal GPIO Enable. 0: The UART1TXD signal will operate normally as the UART1TXD signal. 1: The UART1TXD signal will operate as a General Purpose I/O.
4	RW	0	UART1RTS Signal GPIO Enable. 0: The UART1RTS signal will operate normally as the UART1RTS signal. 1: The UART1RTS signal will operate as a General Purpose I/O.
3	RW	0	UART0RXD Signal GPIO Enable. 0: The UART0RXD signal will operate normally as the UART0RXD signal. 1: The UART0RXD signal will operate as a General Purpose I/O.
2	RW	0	UART0CTS Signal GPIO Enable. 0: The UART0CTS signal will operate normally as the UART0CTS signal. 1: The UART0CTS signal will operate as a General Purpose I/O.
1	RW	0	UART0TXD Signal GPIO Enable. 0: The UART0TXD signal will operate normally as the UART0TXD signal. 1: The UART0TXD signal will operate as a General Purpose I/O.
0	RW	0	UART0RTS Signal GPIO Enable. 0: The UART0RTS signal will operate normally as the UART0RTS signal. 1: The UART0RTS signal will operate as a General Purpose I/O.

Offset 0x0064 – 0x0067

GPIO Enable Control Register for Dedicated GPIO Signals

This register provides the software with the means to enable/disable the associated dedicated GPIO signals to operate as General Purpose I/Os.

Bit	Attribute	Default	Description
31:22	RO	0	Reserved
21	RW	0	SUSGPIO Signal GPIO Enable. 0: The SUSGPIO signal will operate normally as the SUSGPIO signal. 1: The SUSGPIO signal will operate as a General Purpose I/O.
20	RW	0	Reserved

19:16	RW	0	WAKEUP[3:0] Signal GPIO Enable. 0: The corresponding WAKEUP[3:0] signal will operate normally as the WAKEUP signal. 1: The corresponding WAKEUP[3:0] signal will operate as a General Purpose I/O.
15:8	RO	0	Reserved
7:0	RW	8'hDF	GPIO[7:0] Signal GPIO Enable. 0: The corresponding GPIO[7:0] signal will not operate as the GPIO signal. 1: The corresponding GPIO[7:0] signal will operate as a General Purpose I/O.

Offset 0x0068 – 0x006B

GPIO Output Enable Control Register for SD/MMC Signals

This register provides the software with the means to enable/disable the output of the associated SD/MMC signals to operate as General Purpose I/Os.

Bit	Attribute	Default	Description
31:8	RO	0	Reserved
7	RW	0	MMCDATA7 Signal GPIO Output Enable. 0: When this signal is enabled to be a GPIO, and this bit is inactive (a zero), the MMCDATA7 signal will operate as a General Purpose input. 1: When this signal is enabled to be a GPIO, and this bit is active (a one), the MMCDATA7 signal will operate as a General Purpose output. When this signal is not enabled to be a GPIO, the value of this bit has no effect.
6	RW	0	MMCDATA6 Signal GPIO Output Enable. 0: When this signal is enabled to be a GPIO, and this bit is inactive (a zero), the MMCDATA6 signal will operate as a General Purpose input. 1: When this signal is enabled to be a GPIO, and this bit is active (a one), the MMCDATA6 signal will operate as a General Purpose output. When this signal is not enabled to be a GPIO, the value of this bit has no effect.
5	RW	0	MMCDATA5 Signal GPIO Output Enable. 0: When this signal is enabled to be a GPIO, and this bit is inactive (a zero), the MMCDATA5 signal will operate as a General Purpose input. 1: When this signal is enabled to be a GPIO, and this bit is active (a one), the MMCDATA5 signal will operate as a General Purpose output. When this signal is not enabled to be a GPIO, the value of this bit has no effect.
4	RW	0	MMCDATA4 Signal GPIO Output Enable. 0: When this signal is enabled to be a GPIO, and this bit is inactive (a zero), the MMCDATA4 signal will operate as a General Purpose input. 1: When this signal is enabled to be a GPIO, and this bit is active (a one), the MMCDATA4 signal will operate as a General Purpose output. When this signal is not enabled to be a GPIO, the value of this bit has no effect.

3	RW	0	SDDATA3 Signal GPIO Output Enable. 0: When this signal is enabled to be a GPIO, and this bit is inactive (a zero), the SDDATA3 signal will operate as a General Purpose input. 1: When this signal is enabled to be a GPIO, and this bit is active (a one), the SDDATA3 signal will operate as a General Purpose output. When this signal is not enabled to be a GPIO, the value of this bit has no effect.
2	RW	0	SDDATA2 Signal GPIO Output Enable. 0: When this signal is enabled to be a GPIO, and this bit is inactive (a zero), the SDDATA2 signal will operate as a General Purpose input. 1: When this signal is enabled to be a GPIO, and this bit is active (a one), the SDDATA2 signal will operate as a General Purpose output. When this signal is not enabled to be a GPIO, the value of this bit has no effect.
1	RW	0	SDDATA1 Signal GPIO Output Enable. 0: When this signal is enabled to be a GPIO, and this bit is inactive (a zero), the SDDATA1 signal will operate as a General Purpose input. 1: When this signal is enabled to be a GPIO, and this bit is active (a one), the SDDATA1 signal will operate as a General Purpose output. When this signal is not enabled to be a GPIO, the value of this bit has no effect.
0	RW	0	SDDATA0 Signal GPIO Output Enable. 0: When this signal is enabled to be a GPIO, and this bit is inactive (a zero), the SDDATA0 signal will operate as a General Purpose input. 1: When this signal is enabled to be a GPIO, and this bit is active (a one), the SDDATA0 signal will operate as a General Purpose output. When this signal is not enabled to be a GPIO, the value of this bit has no effect.

Offset 0x006C – 0x006F

GPIO Output Enable Control Register for VDIN/VDOUT Signals

This register provides the software with the means to enable/disable the output of the associated VDIN/VDOUT signals to operate as General Purpose I/Os.

Bit	Attribute	Default	Description
31:8	RW	0	VDOUT[23:0] Signal GPIO Output Enable. 0: When the corresponding VDOUT[23:0] is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of VDOUT[23:0] signal will operate as a General Purpose input. 1: When the corresponding VDOUT[23:0] is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of VDOUT[23:0] signal will operate as a General Purpose output. When the corresponding VDOUT[23:0] is not enabled to be a GPIO, the value of this bit has no effect.
7:0	RW	0	VDIN[7:0] Signal GPIO Output Enable. 0: When the corresponding VDIN[7:0] is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of VDIN[7:0] signal will operate as a General Purpose input. 1: When the corresponding VDIN[7:0] is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of VDIN[7:0] signal will operate as a General Purpose output. When the corresponding VDIN[7:0] is not enabled to be a GPIO, the value of this bit has no effect.

Offset 0x0070 – 0x0073

GPIO Output Enable Control Register for SYNC Signals

This register provides the software with the means to enable/disable the output of the associated SYNC signals to operate as General Purpose I/Os.

Bit	Attribute	Default	Description
31:6	RO	0	Reserved
5	RW	0	VDVSYNC Signal GPIO Output Enable. 0: When VDVSYNC is enabled to be a GPIO, and this bit is inactive (a zero), the VDVSYNC signal will operate as a General Purpose input. 1: When VDVSYNC is enabled to be a GPIO, and this bit is active (a one), the VDVSYNC signal will operate as a General Purpose output. When the corresponding VDVSYNC is not enabled to be a GPIO, the value of this bit has no effect.
4	RW	0	VDHSYNC Signal GPIO Output Enable. 0: When VDHSYNC is enabled to be a GPIO, and this bit is inactive (a zero), the VDHSYNC signal will operate as a General Purpose input. 1: When VDHSYNC is enabled to be a GPIO, and this bit is active (a one), the VDHSYNC signal will operate as a General Purpose output. When the corresponding VDHSYNC is not enabled to be a GPIO, the value of this bit has no effect.
3	RW	0	VGAVSYNC Signal GPIO Output Enable. 0: When VGAVSYNC is enabled to be a GPIO, and this bit is inactive (a zero), the VGAVSYNC signal will operate as a General Purpose input. 1: When VGAVSYNC is enabled to be a GPIO, and this bit is active (a one), the VGAVSYNC signal will operate as a General Purpose output. When the corresponding VGAVSYNC is not enabled to be a GPIO, the value of this bit has no effect.
2	RW	0	VGAHSYNC Signal GPIO Output Enable. 0: When VGAHSYNC is enabled to be a GPIO, and this bit is inactive (a zero), the VGAHSYNC signal will operate as a General Purpose input. 1: When VGAHSYNC is enabled to be a GPIO, and this bit is active (a one), the VGAHSYNC signal will operate as a General Purpose output. When the corresponding VGAHSYNC is not enabled to be a GPIO, the value of this has no effect.
1	RW	0	VVSYNC Signal GPIO Output Enable. 0: When VVSYNC is enabled to be a GPIO, and this bit is inactive (a zero), the VVSYNC signal will operate as a General Purpose input. 1: When VVSYNC is enabled to be a GPIO, and this bit is active (a one), the VVSYNC signal will operate as a General Purpose output. When the corresponding VVSYNC is not enabled to be a GPIO, the value of this bit has no effect.
0	RW	0	VHSYNC Signal GPIO Output Enable. 0: When VHSYNC is enabled to be a GPIO, and this bit is inactive (a zero), the VHSYNC signal will operate as a General Purpose input. 1: When VHSYNC is enabled to be a GPIO, and this bit is active (a one), the VHSYNC signal will operate as a General Purpose output. When the corresponding VHSYNC is not enabled to be a GPIO, the value of this bit has no effect.

Offset 0x0074 – 0x0077

GPIO Output Enable Control Register for NOR Data Signals

This register provides the software with the means to enable/disable the output of the associated NOR data signals to operate as General Purpose I/Os.

Bit	Attribute	Default	Description
31:16	RO	0	Reserved
15:0	RW	0	NORD[15:0] Signal GPIO Output Enable. 0: When the corresponding NORD[15:0] is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of NORD[15:0] signal will operate as a General Purpose input. 1: When the corresponding NORD[15:0] is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of NORD[15:0] signal will operate as a General Purpose output. When the corresponding NORD[15:0] is not enabled to be a GPIO, the value of this bit has no effect.

Offset 0x0078 – 0x007B

GPIO Output Enable Control Register for NOR Address Signals

This register provides the software with the means to enable/disable the output of the associated NOR address signals to operate as General Purpose I/Os.

Bit	Attribute	Default	Description
31:25	RO	0	Reserved
24:0	RW	0	NORA[24:0] Signal GPIO Output Enable. 0: When the corresponding NORA[24:0] is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of NORA[24:0] signal will operate as a General Purpose input. 1: When the corresponding NORA[24:0] is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of NORA[24:0] signal will operate as a General Purpose output. When the corresponding NORA[24:0] is not enabled to be a GPIO, the value of this bit has no effect.

Offset 0x007C – 0x007F

GPIO Output Enable Control Register for AC97 Signals

This register provides the software with the means to enable/disable the output of the associated AC97 signals to operate as General Purpose I/Os.

Bit	Attribute	Default	Description
31:5	RO	0	Reserved
4	RW	0	AC97RST Signal GPIO Output Enable. 0: When the corresponding AC97RST is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of AC97RST signal will operate as a General Purpose input. 1: When the corresponding AC97RST is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of AC97RST signal will operate as a General Purpose output. When the corresponding AC97RST is not enabled to be a GPIO, the value of this bit has no effect.

3	RW	0	AC97BCLK signal GPIO Output Enable. 0: When the corresponding AC97BCLK is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of AC97BCLK signal will operate as a General Purpose input. 1: When the corresponding AC97BCLK is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of AC97BCLK signal will operate as a General Purpose output. When the corresponding AC97BCLK is not enabled to be a GPIO, the value of this bit has no effect.
2	RW	0	AC97SDO Signal GPIO Output Enable. 0: When the corresponding AC97SDO is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of AC97SDO signal will operate as a General Purpose input. 1: When the corresponding AC97SDO is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of AC97SDO signal will operate as a General Purpose output. When the corresponding AC97SDO is not enabled to be a GPIO, the value of this bit has no effect.
1	RW	0	AC97SYNC Signal GPIO Output Enable. 0: When the corresponding AC97SYNC is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of AC97SYNC signal will operate as a General Purpose input. 1: When the corresponding AC97SYNC is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of AC97SYNC signal will operate as a General Purpose output. When the corresponding AC97SYNC is not enabled to be a GPIO, the value of this bit has no effect.
0	RW	0	AC97SDI Signal GPIO Output Enable. 0: When the corresponding AC97SDI is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of AC97SDI signal will operate as a General Purpose input. 1: When the corresponding AC97SDI is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of AC97SDI signal will operate as a General Purpose output. When the corresponding AC97SDI is not enabled to be a GPIO, the value of this bit has no effect.

Offset 0x0080 – 0x0083

GPIO Output Enable Control Register for SPI Flash Signals

This register provides the software with the means to enable/disable the output of the associated SPI Flash signals to operate as General Purpose I/Os.

Bit	Attribute	Default	Description
31:5	RO	0	Reserved
4	WR0	0	SFDI Signal GPIO Output Enable. 0: When the corresponding SFDI is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of SFDI signal will operate as a General Purpose input. 1: When the corresponding SFDI is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of SFDI signal will operate as a General Purpose output. When the corresponding SFDI is not enabled to be a GPIO, the value of this bit has no effect.
3	RW	0	SFCLK Signal GPIO Output Enable. 0: When the corresponding SFCLK is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of SFCLK signal will operate as a General Purpose input. 1: When the corresponding SFCLK is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of SFCLK signal will operate as a General Purpose output. When the corresponding SFCLK is not enabled to be a GPIO, the value of this bit has no effect.

2	RW	0	SFCS1 Signal GPIO Output Enable. 0: When the corresponding SFCS1 is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of SFCS1 signal will operate as a General Purpose input. 1: When the corresponding SFCS1 is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of SFCS1 signal will operate as a General Purpose output. When the corresponding SFCS1 is not enabled to be a GPIO, the value of this bit has no effect.
1	RW	0	SFCS0 Signal GPIO Output Enable. 0: When the corresponding SFCS0 is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of SFCS0 signal will operate as a General Purpose input. 1: When the corresponding SFCS0 is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of SFCS0 signal will operate as a General Purpose output. When the corresponding SFCS0 is not enabled to be a GPIO, the value of this bit has no effect.
0	RW	0	SFDO Signal GPIO Output Enable. 0: When the corresponding SFDO is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of SFDO signal will operate as a General Purpose input. 1: When the corresponding SFDO is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of SFDO signal will operate as a General Purpose output. When the corresponding SFDO is not enabled to be a GPIO, the value of this bit has no effect.

Offset 0x0084 – 0x0087

GPIO Output Enable Control Register for SPI Signals

This register provides the software with the means to enable/disable the output of the associated SPI signals to operate as General Purpose I/Os.

Bit	Attribute	Default	Description
31:12	RO	0	Reserved
11	RW	0	SPI2SS Signal GPIO Output Enable. 0: When the corresponding SPI2SS is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of SPI2SS signal will operate as a General Purpose input. 1: When the corresponding SPI2SS is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of SPI2SS signal will operate as a General Purpose output. When the corresponding SPI2SS is not enabled to be a GPIO, the value of this bit has no effect.
10	RW	0	SPI2MOSI Signal GPIO Output Enable. 0: When the corresponding SPI2MOSI is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of SPI2MOSI signal will operate as a General Purpose input. 1: When the corresponding SPI2MOSI is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of SPI2MOSI signal will operate as a General Purpose output. When the corresponding SPI2MOSI is not enabled to be a GPIO, the value of this bit has no effect.

9	RW	0	SPI2MISO Signal GPIO Output Enable. 0: When the corresponding SPI2MISO is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of SPI2MISO signal will operate as a General Purpose input. 1: When the corresponding SPI2MISO is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of SPI2MISO signal will operate as a General Purpose output. When the corresponding SPI2MISO is not enabled to be a GPIO, the value of this bit has no effect.
8	RW	0	SPI2CLK Signal GPIO Output Enable. 0: When the corresponding SPI2CLK is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of SPI2CLK signal will operate as a General Purpose input. 1: When the corresponding SPI2CLK is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of SPI2CLK signal will operate as a General Purpose output. When the corresponding SPI2CLK is not enabled to be a GPIO, the value of this bit has no effect.
7	RW	0	SPI1SS Signal GPIO Output Enable. 0: When the corresponding SPI1SS is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of SPI1SS signal will operate as a General Purpose input. 1: When the corresponding SPI1SS is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of SPI1SS signal will operate as a General Purpose output. When the corresponding SPI1SS is not enabled to be a GPIO, the value of this bit has no effect.
6	RW	0	SPI1MOSI Signal GPIO Output Enable. 0: When the corresponding SPI1MOSI is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of SPI1MOSI signal will operate as a General Purpose input. 1: When the corresponding SPI1MOSI is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of SPI1MOSI signal will operate as a General Purpose output. When the corresponding SPI1MOSI is not enabled to be a GPIO, the value of this bit has no effect.
5	RW	0	SPI1MISO Signal GPIO Output Enable. 0: When the corresponding SPI1MISO is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of SPI1MISO signal will operate as a General Purpose input. 1: When the corresponding SPI1MISO is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of SPI1MISO signal will operate as a General Purpose output. When the corresponding SPI1MISO is not enabled to be a GPIO, the value of this bit has no effect.
4	RW	0	SPI1CLK Signal GPIO Output Enable. 0: When the corresponding SPI1CLK is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of SPI1CLK signal will operate as a General Purpose input. 1: When the corresponding SPI1CLK is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of SPI1CLK signal will operate as a General Purpose output. When the corresponding SPI1CLK is not enabled to be a GPIO, the value of this bit has no effect.
3	RW	0	SPI0SS Signal GPIO Output Enable. 0: When the corresponding SPI0SS is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of SPI0SS signal will operate as a General Purpose input. 1: When the corresponding SPI0SS is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of SPI0SS signal will operate as a General Purpose output. When the corresponding SPI0SS is not enabled to be a GPIO, the value of this bit has no effect.

2	RW	0	SPI0MOSI Signal GPIO Output Enable. 0: When the corresponding SPI0MOSI is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of SPI0MOSI signal will operate as a General Purpose input. 1: When the corresponding SPI10MOSI is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of SPI0MOSI signal will operate as a General Purpose output. When the corresponding SPI0MOSI is not enabled to be a GPIO, the value of this bit has no effect.
1	RW	0	SPI0MISO Signal GPIO Output Enable. 0: When the corresponding SPI0MISO is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of SPI0MISO signal will operate as a General Purpose input. 1: When the corresponding SPI0MISO is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of SPI0MISO signal will operate as a General Purpose output. When the corresponding SPI0MISO is not enabled to be a GPIO, the value of this bit has no effect.
0	RW	0	SPI0CLK Signal GPIO Output Enable. 0: When the corresponding SPI0CLK is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of SPI0CLK signal will operate as a General Purpose input. 1: When the corresponding SPI0CLK is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of SPI0CLK signal will operate as a General Purpose output. When the corresponding SPI0CLK is not enabled to be a GPIO, the value of this bit has no effect.

Offset 0x0088 – 0x008B

GPIO Output Enable Control Register for UART Signals

This register provides the software with the means to enable/disable the output of the associated UART signals to operate as General Purpose I/Os.

Bit	Attribute	Default	Description
31:16	RO	0	Reserved
15	RW	0	UART3RXD Signal GPIO Output Enable. 0: When the corresponding UART3RXD is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of UART3RXD signal will operate as a General Purpose input. 1: When the corresponding UART3RXD is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of UART3RXD signal will operate as a General Purpose output. When the corresponding UART3RXD is not enabled to be a GPIO, the value of this bit has no effect.
14	RW	0	UART3CTS Signal GPIO Output Enable. 0: When the corresponding UART3CTS is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of UART3CTS signal will operate as a General Purpose input. 1: When the corresponding UART3CTS is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of UART3CTS signal will operate as a General Purpose output. When the corresponding UART3CTS is not enabled to be a GPIO, the value of this bit has no effect.

13	RW	0	UART3TXD Signal GPIO Output Enable. 0: When the corresponding UART3TXD is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of UART3TXD signal will operate as a General Purpose input. 1: When the corresponding UART3TXD is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of UART3TXD signal will operate as a General Purpose output. When the corresponding UART3TXD is not enabled to be a GPIO, the value of this bit has no effect.
12	RW	0	UART3RTS Signal GPIO Output Enable. 0: When the corresponding UART3RTS is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of UART3RTS signal will operate as a General Purpose input. 1: When the corresponding UART3RTS is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of UART3RTS signal will operate as a General Purpose output. When the corresponding UART3RTS is not enabled to be a GPIO, the value of this bit has no effect.
11	RW	0	UART2RXD Signal GPIO Output Enable. 0: When the corresponding UART2RXD is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of UART2RXD signal will operate as a General Purpose input. 1: When the corresponding UART2RXD is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of UART2RXD signal will operate as a General Purpose output. When the corresponding UART2RXD is not enabled to be a GPIO, the value of this bit has no effect.
10	RW	0	UART2CTS Signal GPIO Output Enable. 0: When the corresponding UART2CTS is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of UART2CTS signal will operate as a General Purpose input. 1: When the corresponding UART2CTS is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of UART2CTS signal will operate as a General Purpose output. When the corresponding UART2CTS is not enabled to be a GPIO, the value of this bit has no effect.
9	RW	0	UART2TXD Signal GPIO Output Enable. 0: When the corresponding UART2TXD is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of UART2TXD signal will operate as a General Purpose input. 1: When the corresponding UART2TXD is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of UART2TXD signal will operate as a General Purpose output. When the corresponding UART2TXD is not enabled to be a GPIO, the value of this bit has no effect.
8	RW	0	UART2RTS Signal GPIO Output Enable. 0: When the corresponding UART2RTS is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of UART2RTS signal will operate as a General Purpose input. 1: When the corresponding UART2RTS is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of UART2RTS signal will operate as a General Purpose output. When the corresponding UART2RTS is not enabled to be a GPIO, the value of this bit has no effect.
7	RW	0	UART1RXD Signal GPIO Output Enable. 0: When the corresponding UART1RXD is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of UART1RXD signal will operate as a General Purpose input. 1: When the corresponding UART1RXD is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of UART1RXD signal will operate as a General Purpose output. When the corresponding UART1RXD is not enabled to be a GPIO, the value of this bit has no effect.

6	RW	0	UART1CTS Signal GPIO Output Enable. 0: When the corresponding UART1CTS is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of UART1CTS signal will operate as a General Purpose input. 1: When the corresponding UART1CTS is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of UART1CTS signal will operate as a General Purpose output. When the corresponding UART1CTS is not enabled to be a GPIO, the value of this bit has no effect.
5	RW	0	UART1TXD Signal GPIO Output Enable. 0: When the corresponding UART1TXD is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of UART1TXD signal will operate as a General Purpose input. 1: When the corresponding UART1TXD is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of UART1TXD signal will operate as a General Purpose output. When the corresponding UART1TXD is not enabled to be a GPIO, the value of this bit has no effect.
4	RW	0	UART1RTS Signal GPIO Output Enable. 0: When the corresponding UART1RTS is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of UART1RTS signal will operate as a General Purpose input. 1: When the corresponding UART1RTS is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of UART1RTS signal will operate as a General Purpose output. When the corresponding UART1RTS is not enabled to be a GPIO, the value of this bit has no effect.
3	RW	0	UART0RXD Signal GPIO Output Enable. 0: When the corresponding UART0RXD is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of UART0RXD signal will operate as a General Purpose input. 1: When the corresponding UART0RXD is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of UART0RXD signal will operate as a General Purpose output. When the corresponding UART0RXD is not enabled to be a GPIO, the value of this bit has no effect.
2	RW	0	UART0CTS Signal GPIO Output Enable. 0: When the corresponding UART0CTS is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of UART0CTS signal will operate as a General Purpose input. 1: When the corresponding UART0CTS is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of UART0CTS signal will operate as a General Purpose output. When the corresponding UART0CTS is not enabled to be a GPIO, the value of this bit has no effect.
1	RW	0	UART0TXD Signal GPIO Output Enable. 0: When the corresponding UART0TXD is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of UART0TXD signal will operate as a General Purpose input. 1: When the corresponding UART0TXD is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of UART0TXD signal will operate as a General Purpose output. When the corresponding UART0TXD is not enabled to be a GPIO, the value of this bit has no effect.
0	RW	0	UART0RTS Signal GPIO Output Enable. 0: When the corresponding UART0RTS is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of UART0RTS signal will operate as a General Purpose input. 1: When the corresponding UART0RTS is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of UART0RTS signal will operate as a General Purpose output. When the corresponding UART0RTS is not enabled to be a GPIO, the value of this bit has no effect.

Offset 0x008C – 0x008F

GPIO Output Enable Control Register for Dedicated GPIO Signals

This register provides the software with the means to enable/disable the output of the associated dedicated GPIO signals to operate as General Purpose I/Os.

Bit	Attribute	Default	Description
31:22	RO	0	Reserved
21	RW	0	SUSGPIO Signal GPIO Output Enable. 0: When SUSGPIO is enabled to be a GPIO, and this bit is inactive (a zero), the SUSGPIO signal will operate as a General Purpose input. 1: When SUSGPIO is enabled to be a GPIO, and this bit is active (a one), the SUSGPIO signal will operate as a General Purpose output. When the SUSGPIO is not enabled to be a GPIO, the value of this bit has no effect.
20	RW	0	Reserved
19:16	RW	0	WAKEUP[3:0] Signal GPIO Output Enable. 0: When the corresponding bit of WAKEUP[3:0] is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of WAKEUP[3:0] signal will operate as a General Purpose input. 1: When the corresponding bit of WAKEUP[3:0] is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of WAKEUP[3:0] signal will operate as a General Purpose output. When the corresponding bit of WAKEUP[3:0] is not enabled to be a GPIO, the value of this bit has no effect.
15:8	RO	0	Reserved
7:0	RW	0	GPIO[7:0] Signal GPIO Output Enable. 0: When the corresponding GPIO[7:0] is enabled to be a GPIO, and this bit is inactive (a zero), the corresponding bit of GPIO[7:0] signal will operate as a General Purpose input. 1: When the corresponding bit of GPIO[7:0] is enabled to be a GPIO, and this bit is active (a one), the corresponding bit of GPIO[7:0] signal will operate as a General Purpose output. When the corresponding bit of GPIO[7:0] is not enabled to be a GPIO, the value of this bit has no effect.

Offset 0x0090 – 0x0093

GPIO Output Data Register for SD/MMC Signals

This register provides the software with the means to control the output data values associated with the SD/MMC signals when they have been enabled as General Purpose outputs.

Bit	Attribute	Default	Description
31:8	RO	0	Reserved
7:4	RW	0	MMCDATA[7:4] Signal GPIO Output Data. When the corresponding bit of MMCDATA[7:4] signal is enabled to be a General Purpose output, the value of this bit is the value driven onto the signal.
3:0	RW	0	SDDATA[3:0] Signal GPIO Output Data. When the corresponding bit of SDDATA[3:0] signal is enabled to be a General Purpose output, the value of this bit is the value driven onto the signal.

Offset 0x0094 – 0x0097

GPIO Output Data Register for VDOUT/VDIN Signals

The purpose of this register is to provide software with the means to control the output data values associated with the VDOUT/VDIN signals when they have been enabled as General Purpose outputs.

Bit	Attribute	Default	Description
31:8	RW	0	VDOUT[23:0] Signal GPIO Output Data. When the corresponding bit of VDOUT[23:0] is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When the corresponding bit of VDOUT[23:0] is not enabled to be a GPIO output, the value of this bit has no effect.
7:0	RW	0	VDIN[7:0] Signal GPIO Output Data. When the corresponding bit of VDIN[7:0] is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When the corresponding bit of VDIN[7:0] is not enabled to be a GPIO output, the value of this bit has no effect.

Offset 0x0098 – 0x009B

GPIO Output Data Register for SYNC Signals

This register provides the software with the means to control the output data values associated with the SYNC signals when they have been enabled as General Purpose outputs.

Bit	Attribute	Default	Description
31:6	RO	0	Reserved
5	RW	0	VDVSYNC Signal GPIO Output Data. When VDVSYNC is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When VDVSYNC is not enabled to be a GPIO output, the value of this bit has no effect.
4	RW	0	VDHSYNC Signal GPIO Output Data. When VDHSYNC is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When VDHSYNC is not enabled to be a GPIO output, the value of this bit has no effect.
3	RW	0	VGAVSYNC Signal GPIO Output Data. When VGAVSYNC is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When VGAVSYNC is not enabled to be a GPIO output, the value of this bit has no effect.
2	RW	0	VGAHSYNC Signal GPIO Output Data. When VGAHSYNC is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When VGAHSYNC is not enabled to be a GPIO output, the value of this bit has no effect.
1	RW	0	VVSYNC Signal GPIO Output Data. When VVSYNC is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When VVSYNC is not enabled to be a GPIO output, the value of this bit has no effect.
0	RW	0	VHSYNC Signal GPIO Output Data. When VHSYNC is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When VHSYNC is not enabled to be a GPIO output, the value of this bit has no effect.

Offset 0x009C - 0x009F

GPIO Output Data Register for NOR Data Signals

This register provides the software with the means to control the output data values associated with the NORD signals when they have been enabled as General Purpose outputs.

Bit	Attribute	Default	Description
31:16	RO	0	Reserved
15:0	RW	0	NORD[15:0] Signal GPIO Output Data. When the corresponding bit of NORD[15:0] is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When the corresponding bit of NORD[15:0] is not enabled to be a GPIO output, the value of this bit has no effect.

Offset 0x00A0 – 0x00A3

GPIO Output Data Register for NOR Address Signals

This register provides the software with the means to control the output data values associated with the NORA signals when they have been enabled as General Purpose outputs.

Bit	Attribute	Default	Description
31:25	RO	0	Reserved
24:0	RW	0	NORA[24:0] Signal GPIO Output Data. When the corresponding bit of NORA[24:0] is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When the corresponding bit of NORA[24:0] is not enabled to be a GPIO output, the value of this bit has no effect.

Offset 0x00A4 – 0x00A7

GPIO Output Data Register for NOR Address Signals

This register provides the software with the means to control the output data values associated with the AC97 signals when they have been enabled as General Purpose outputs.

Bit	Attribute	Default	Description
31:5	RO	0	Reserved
4	RW	0	AC97RST Signal GPIO Output Data. When AC97RST is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When AC97RST is not enabled to be a GPIO output, the value of this bit has no effect.
3	RW	0	AC97BCLK Signal GPIO Output Data. When AC97BCLK is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When AC97BCLK is not enabled to be a GPIO output, the value of this bit has no effect.
2	RW	0	AC97SDO Signal GPIO Output Data. When AC97SDO is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When AC97SDO is not enabled to be a GPIO output, the value of this bit has no effect.
1	RW	0	AC97SYNC Signal GPIO Output Data. When AC97SYNC is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When AC97SYNC is not enabled to be a GPIO output, the value of this bit has no effect.

0	RW	0	AC97SDI Signal GPIO Output Data. When AC97SDI is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When AC97SDI is not enabled to be a GPIO output, the value of this bit has no effect.

Offset 0x00A8 – 0x00AB

GPIO Output Data Register for SPI Flash Signals

This register provides the software with the means to control the output data values associated with the SPI Flash signals when they have been enabled as General Purpose outputs.

Bit	Attribute	Default	Description
31:5	RO	0	Reserved
4	WR0	0	SFDI Signal GPIO Output Data. When SFDI is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When SFDI is not enabled to be a GPIO output, the value of this bit has no effect.
3	RW	0	SFCLK Signal GPIO Output Data. When SFCLK is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When SFCLK is not enabled to be a GPIO output, the value of this bit has no effect.
2:1	RW	0	SFCS[1:0] Signal GPIO Output Data. When the corresponding bit of SFCS[1:0] is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When the corresponding bit of SFCS[1:0] is not enabled to be a GPIO output, the value of this bit has no effect.
0	RW	0	SFDO Signal GPIO Output Data. When SFDO is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When SFDO is not enabled to be a GPIO output, the value of this bit has no effect.

Offset 0x00AC – 0x00AF

GPIO Output Data Register for SPI Signals

This register provides the software with the means to control the output data values associated with the SPI signals when they have been enabled as General Purpose outputs.

Bit	Attribute	Default	Description
31:12	RO	0	Reserved
11	RW	0	SPI2SS Signal GPIO Output Data. When SPI2SS is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When SPI2SS is not enabled to be a GPIO output, the value of this bit has no effect.
10	RW	0	SPI2MOSI Signal GPIO Output Data. When SPI2MOSI is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When SPI2MOSI is not enabled to be a GPIO output, the value of this bit has no effect.
9	RW	0	SPI2MISO Signal GPIO Output Data. When SPI2MISO is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When SPI2MISO is not enabled to be a GPIO output, the value of this bit has no effect.

8	RW	0	SPI2CLK Signal GPIO Output Data. When SPI2CLK is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When SPI2CLK is not enabled to be a GPIO output, the value of this bit has no effect.
7	RW	0	SPI1SS Signal GPIO Output Data. When SPI1SS is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When SPI1SS is not enabled to be a GPIO output, the value of this bit has no effect.
6	RW	0	SPI1MOSI Signal GPIO Output Data. When SPI1MOSI is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When SPI1MOSI is not enabled to be a GPIO output, the value of this bit has no effect.
5	RW	0	SPI1MISO Signal GPIO Output Data. When SPI1MISO is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When SPI1MISO is not enabled to be a GPIO output, the value of this bit has no effect.
4	RW	0	SPI1CLK Signal GPIO Output Data. When SPI1CLK is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When SPI1CLK is not enabled to be a GPIO output, the value of this bit has no effect.
3	RW	0	SPI0SS Signal GPIO Output Data. When SPI0SS is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When SPI0SS is not enabled to be a GPIO output, the value of this bit has no effect.
2	RW	0	SPI0MOSI Signal GPIO Output Data. When SPI0MOSI is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When SPI0MOSI is not enabled to be a GPIO output, the value of this bit has no effect.
1	RW	0	SPI0MISO Signal GPIO Output Data. When SPI0MISO is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When SPI0MISO is not enabled to be a GPIO output, the value of this bit has no effect.
0	RW	0	SPI0CLK Signal GPIO Output Data. When SPI0CLK is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When SPI0CLK is not enabled to be a GPIO output, the value of this bit has no effect.

Offset 0x00B0 – 0x00B3

GPIO Output Data Register for UART Signals

This register provides the software with the means to control the output data values associated with the UART signals when they have been enabled as General Purpose outputs.

Bit	Attribute	Default	Description
31:16	RO	0	Reserved
15	RW	0	UART3RXD Signal GPIO Output Data. When UART3RXD is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When UART3RXD is not enabled to be a GPIO output, the value of this bit has no effect.
14	RW	0	UART3CTS Signal GPIO Output Data. When UART3CTS is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When UART3CTS is not enabled to be a GPIO output, the value of this bit has no effect.
13	RW	0	UART3TXD Signal GPIO Output Data. When UART3TXD is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When UART3TXD is not enabled to be a GPIO output, the value of this bit has no effect.
12	RW	0	UART3RTS Signal GPIO Output Data. When UART3RTS is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When UART3RTS is not enabled to be a GPIO output, the value of this bit has no effect.
11	RW	0	UART2RXD Signal GPIO Output Data. When UART2RXD is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When UART2RXD is not enabled to be a GPIO output, the value of this bit has no effect.
10	RW	0	UART2CTS Signal GPIO Output Data. When UART2CTS is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When UART2CTS is not enabled to be a GPIO output, the value of this bit has no effect.
9	RW	0	UART2TXD Signal GPIO Output Data. When UART2TXD is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When UART2TXD is not enabled to be a GPIO output, the value of this bit has no effect.
8	RW	0	UART2RTS Signal GPIO Output Data. When UART2RTS is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When UART2RTS is not enabled to be a GPIO output, the value of this bit has no effect.
7	RW	0	UART1RXD Signal GPIO Output Data. When UART1RXD is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When UART1RXD is not enabled to be a GPIO output, the value of this bit has no effect.
6	RW	0	UART1CTS Signal GPIO Output Data. When UART1CTS is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When UART1CTS is not enabled to be a GPIO output, the value of this bit has no effect.

5	RW	0	UART1TXD Signal GPIO Output Data. When UART1TXD is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When UART1TXD is not enabled to be a GPIO output, the value of this bit has no effect.
4	RW	0	UART1RTS Signal GPIO Output Data. When UART1RTS is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When UART1RTS is not enabled to be a GPIO output, the value of this bit has no effect.
3	RW	0	UART0RXD Signal GPIO Output Data. When UART0RXD is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When UART0RXD is not enabled to be a GPIO output, the value of this bit has no effect.
2	RW	0	UART0CTS Signal GPIO Output Data. When UART0CTS is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When UART0CTS is not enabled to be a GPIO output, the value of this bit has no effect.
1	RW	0	UART0TXD Signal GPIO Output Data. When UART0TXD is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When UART0TXD is not enabled to be a GPIO output, the value of this bit has no effect.
0	RW	0	UART0RTS Signal GPIO Output Data. When UART0RTS is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When UART0RTS is not enabled to be a GPIO output, the value of this bit has no effect.

Offset 0x00B4 – 0x00B7

GPIO Output Data Register for Dedicated GPIO Signals

This register provides the software with the means to control the output data values associated with the Dedicated GPIO signals when they have been enabled as General Purpose outputs.

Bit	Attribute	Default	Description
31:22	RO	0	Reserved
21	RW	0	SUSGPIO Signal GPIO Output Data. When SUSGPIO is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When SUSGPIO is not enabled to be a GPIO output, the value of this bit has no effect.
20	RW	0	Reserved
19:16	RW	0	WAKEUP[3:0] Signal GPIO Output Data. When the corresponding bit of WAKEUP[3:0] is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When the corresponding bit of WAKEUP[3:0] is not enabled to be a GPIO output, the value of this bit has no effect.
15:8	RO	0	Reserved
7:0	RW	0	GPIO[7:0] Signal GPIO Output Data. When the corresponding bit of GPIO[7:0] is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When the corresponding bit of GPIO[7:0] is not enabled to be a GPIO output, the value of this bit has no effect.

Offset 0x00B8 – 0x00BB

GPIO Input Data Register for SD/MMC Signals

This register provides the software with the means to control the output data values associated with the SD/MMC signals when they have been enabled as General Purpose outputs.

Bit	Attribute	Default	Description
31:8	RO	0	Reserved
7:4	RO	-	MMCDATA[7:4] Signal GPIO Input Data. The value read will always be the value on the associated signal.
3:0	RO	-	SDDATA[3:0] Signal GPIO Input Data. The value read will always be the value on the associated signal.

Offset 0x00BC – 0x00BF

GPIO Input Data Register for VDOUT/VDIN Signals

This register provides the software with the means to control the output data values associated with the VDOUT/VDIN signals when they have been enabled as General Purpose outputs.

Bit	Attribute	Default	Description
31:8	RO	-	VDOUT[23:0] Signal GPIO Input Data. The value read will always be the value on the associated signal.
7:0	RO	-	VDIN[7:0] Signal GPIO Input Data. The value read will always be the value on the associated signal.

Offset 0x00C0 – 0x00C3

GPIO Input Data Register for SYNC Signals

This register provides the software with the means to control the output data values associated with the SYNC signals when they have been enabled as General Purpose outputs.

Bit	Attribute	Default	Description
31:6	RO	0	Reserved
5	RO	-	VDVSYNC Signal GPIO Input Data. The value read will always be the value on the associated signal.
4	RO	-	VDHSYNC Signal GPIO Input Data. The value read will always be the value on the associated signal.
3	RO	-	VGAVSYNC Signal GPIO Input Data. The value read will always be the value on the associated signal.
2	RO	-	VGAHSYNC Signal GPIO Input Data. The value read will always be the value on the associated signal.
1	RO	-	VVSYNC Signal GPIO Input Data. The value read will always be the value on the associated signal.
0	RO	-	VHSYNC Signal GPIO Input Data. The value read will always be the value on the associated signal.

Offset 0x00C4 – 0x00C7

GPIO Input Data Register for NOR Data Signals

This register provides the software with the means to control the output data values associated with the NORD signals when they have been enabled as General Purpose outputs.

Bit	Attribute	Default	Description
31:16	RO	0	Reserved
15:0	RO	-	NORD[15:0] Signal GPIO Input Data. The value read will always be the value on the associated signal.

Offset 0x00C8 - 0x00CB

GPIO Input Data Register for NOR Address Signals

This register provides the software with the means to control the output data values associated with the NORA signals when they have been enabled as General Purpose outputs.

Bit	Attribute	Default	Description
31:25	RO	0	Reserved
24:0	RO	-	NORA[24:0] Signal GPIO Input Data. The value read will always be the value on the associated signal.

Offset 0x00CC – 0x00CF

GPIO Input Data Register for AC97 Signals

This register provides the software with the means to control the output data values associated with the AC97 signals when they have been enabled as General Purpose outputs.

Bit	Attribute	Default	Description
31:5	RO	0	Reserved
4	RO	-	AC97RST Signal GPIO Input Data. The value read will always be the value on the associated signal.
3	RO	-	AC97BCLK Signal GPIO Input Data. The value read will always be the value on the associated signal.
2	RO	-	AC97SDO Signal GPIO Input Data. The value read will always be the value on the associated signal.
1	RO	-	AC97SYNC Signal GPIO Input Data. The value read will always be the value on the associated signal.
0	RO	-	AC97SDI Signal GPIO Input Data. The value read will always be the value on the associated signal.

Offset 0x00D0 – 0x00D3

GPIO Input Data Register for SPI Flash Signals

This register provides the software with the means to control the output data values associated with the SPI Flash signals when they have been enabled as General Purpose outputs.

Bit	Attribute	Default	Description
31:5	RO	0	Reserved
4	RO	-	SFDI Signal GPIO Input Data. The value read will always be the value on the associated signal.
3	RO	-	SFCLK Signal GPIO Input Data. The value read will always be the value on the associated signal.
2:1	RO	-	SFCS[1:0] Signal GPIO Input Data. The value read will always be the value on the associated signal.

0	RO	-	SFDO Signal GPIO Input Data. The value read will always be the value on the associated signal.

Offset 0x00D4 – 0x00D7

GPIO Input Data Register for SPI Signals

This register provides the software with the means to control the output data values associated with the SPI signals when they have been enabled as General Purpose outputs.

Bit	Attribute	Default	Description
31:12	RO	0	Reserved
11	RO	-	SPI2SS Signal GPIO Input Data. The value read will always be the value on the associated signal.
10	RO	-	SPI2MOSI Signal GPIO Input Data. The value read will always be the value on the associated signal.
9	RO	-	SPI2MISO Signal GPIO Input Data. The value read will always be the value on the associated signal.
8	RO	-	SPI2CLK Signal GPIO Input Data. The value read will always be the value on the associated signal.
7	RO	-	SPI1SS Signal GPIO Input Data. The value read will always be the value on the associated signal.
6	RO	-	SPI1MOSI Signal GPIO Input Data. The value read will always be the value on the associated signal.
5	RO	-	SPI1MISO Signal GPIO Input Data. The value read will always be the value on the associated signal.
4	RO	-	SPI1CLK Signal GPIO Input Data. The value read will always be the value on the associated signal.
3	RO	-	SPI0SS Signal GPIO Input Data. The value read will always be the value on the associated signal.
2	RO	-	SPI0MOSI Signal GPIO Input Data. The value read will always be the value on the associated signal.
1	RO	-	SPI0MISO Signal GPIO Input Data. The value read will always be the value on the associated signal.
0	RO	-	SPI0CLK Signal GPIO Input Data. The value read will always be the value on the associated signal.

Offset 0x00D8 – 0x00DB

GPIO Input Data Register for UART Signals

This register provides the software with the means to control the output data values associated with the UART signals when they have been enabled as General Purpose outputs.

Bit	Attribute	Default	Description
31:16	RO	0	Reserved
15	RO	-	UART3RXD Signal GPIO Input Data. The value read will always be the value on the associated signal.
14	RO	-	UART3CTS Signal GPIO Input Data. The value read will always be the value on the associated signal.
13	RO	-	UART3TXD Signal GPIO Input Data. The value read will always be the value on the associated signal.
12	RO	-	UART3RTS Signal GPIO Input Data. The value read will always be the value on the associated signal.
11	RO	-	UART2RXD Signal GPIO Input Data. The value read will always be the value on the associated signal.

10	RO	-	UART2CTS Signal GPIO Input Data. The value read will always be the value on the associated signal.
9	RO	-	UART2TXD Signal GPIO Input Data. The value read will always be the value on the associated signal.
8	RO	-	UART2RTS Signal GPIO Input Data. The value read will always be the value on the associated signal.
7	RO	-	UART1RXD Signal GPIO Input Data. The value read will always be the value on the associated signal.
6	RO	-	UART1CTS Signal GPIO Input Data. The value read will always be the value on the associated signal.
5	RO	-	UART1TXD Signal GPIO Input Data. The value read will always be the value on the associated signal.
4	RO	-	UART1RTS Signal GPIO Input Data. The value read will always be the value on the associated signal.
3	RO	-	UART0RXD Signal GPIO Input Data. The value read will always be the value on the associated signal.
2	RO	-	UART0CTS Signal GPIO Input Data. The value read will always be the value on the associated signal.
1	RO	-	UART0TXD Signal GPIO Input Data. The value read will always be the value on the associated signal.
0	RO	-	UART0RTS Signal GPIO Input Data. The value read will always be the value on the associated signal.

Offset 0x00DC – 0x00DF

GPIO Input Data Register for Dedicated GPIO Signals

This register provides the software with the means to control the output data values associated with the dedicated GPIO signals when they have been enabled as General Purpose outputs.

Bit	Attribute	Default	Description
31:22	RO	0	Reserved
21	RO	-	SUSGPIO Signal GPIO Input Data. The value read will always be the value on the associated signal.
20	RO	0	Reserved
19:16	RO	-	WAKEUP[3:0] Signal GPIO Input Data. The value read will always be the value on the associated signal.
15:8	RO	0	Reserved
7:0	RO	-	GPIO[7:0] Signal GPIO Input Data. The value read will always be the value on the associated signal.

Offset 0x0100 – 0x0103

Strapping Option Status Register

Bit	Attribute	Default	Description
31	RO	-	0: Auto Power On. 1: Need PWRBTN Pressed.
30	RO	-	0: Using ROSC clock. 1: Bypass Ring OSC Bypass.
29	RO	-	0: MII Mode. 1: RVMII Mode.
28	RO	-	0: Reserved. 1: MII/RVMII Mode.
27:25	RO	0	Reserved

24	RO	-	0: 4 UART mode. 1: 6 UART Mode.
23:21	RO	-	CLKTST/CLKOUT Output Select.
20	RO	-	0: CLK25 from internal PLL. 1: CLK25 from 24MHZXI pin.
19	RO	-	0: CLK24 from internal PLL. 1: CLK24 from 24MHZXI pin.
18	RO	-	0: USB cable attach detect by HW. 1: USB cable attach detect by SW.
17	RO	-	0: PLL source clock from OSC. 1: PLL source clock from 27MHZXI.
16:15	RO	0	Reserved
14	RO	-	0: SF negative edge sampling. 1: SF positive edge sampling.
13:12	RO	0	Reserved
11:10	RO	-	PLL and Clock Divisors/Mode.
9:8	RO	0	Reserved
7	RO	-	0: Disable KPAD. 1: Enable KPAD.
6:5	RO	-	NAND Flash Boot Address Cycle/NOR Flash Boot Timing Select.
4	RO	-	NAND Flash Boot Page Size/NOR Flash Boot Page Mode.
3	RO	-	Serial Flash Address Size(When Serial Flash Boot).
2:1	RO	-	NAND/NOR/Serial Flash Boot up Select.
0	RO	0	Reserved

Offset 0x0104 – 0x0107

PMC Suspend RTC Clock Exist Status Register

Bit	Attribute	Default	Description
31:1	RO	0	Reserved
0	RO	0	0: RTC clock does not exist. 1: RTC clock exists.

Offset 0x0108 – 0x010B: Reserved

Offset 0x010C – 0x010F

SF Negative Edge Sampling Control Register

This register is used to overwrite the configuration of STRAPPING_OPTION[25]. The software can use this register to control the internal SF interface to use negative edge or positive edge for sampling.

Bit	Attribute	Default	Description
31:2	RO	0	Reserved
1	RW	0	0: Use STRAPPING_OPTION[25]. 1: Use bit[0].
0	RW	0	0: SF positive edge sampling. 1: SF negative edge sampling.

Offset 0x0110 – 0x0113

Bonding Option Status Register

The Purpose of this register is for SW to read the values of CONFIG[2:0].

Bit	Attribute	Default	Description
31:3	RO	0	Reserved
2	RO	0	CONFIG2
1	RO	0	CONFIG1
0	RO	0	CONFIG0

Offset 0x0200 – 0x0203

Pin-Sharing Selection Register

This register provides the software with the means to select the sharing pin which may share with other IP.

Bit	Attribute	Default	Description
31	RW	0	DVO Output Enable.
30	RW	0	VGA Output Enable.
29:24	RW	0	Reserved
23	RW	0	DVI VCLK Inverting Option. 0: Use VCLK for internal use 1: Invert VCLK for internal use
22:5	RW	0	Reserved
4	RW	0	NOR/NAND Pin Sharing Select. 0:Select NAND 1:Select NOR
3	RW	0	Enable for CLKOUT Pin to Output 24 MHz Clock. 0: CLKOUT behaves as other functions. 1: CLKOUT behaves for CCIR function. 24 MHZ clock output via CLKOUT pin.
2	RW	0	I2SMCLK Output Enable Control. 0. Disable I2SMCLK output. 1. Enable I2SMCLK output.
1	RW	0	AC97/I^{2}S Pin Sharing Select. 0: I^{2}S 1: AC97
0	RW	0	Reserved

Offset 0x0300 – 0x0303

GPIO Interrupt Request Type Register

This register provides the software with the means to configure the edge/level sensitivity associated with generating interrupt requests via the GPIO[7:0] signal pins.

Bit	Attribute	Default	Description
31:16	RO	0	Reserved
15:14	RW	0	General Purpose IRQ7 Type Bits. These bits determine the type of IRQ event that will generate an active gpio_irq[7] signal, as follows: 00: The GPIO[7] input (io_gpio_in[7]) signal is a zero. 01: The GPIO[7] input (io_gpio_in[7]) signal is a one. 10: The GPIO[7] input (io_gpio_in[7]) signal generates a falling edge. 11: The GPIO[7] input (io_gpio_in[7]) signal generates a rising edge.

13:12	RW	0	General Purpose IRQ6 Type Bits. These bits determine the type of IRQ event that will generate an active gpio_irq[6] signal, as follows: 00: The GPIO[6] input (io_gpio_in[6]) signal is a zero. 01: The GPIO[6] input (io_gpio_in[6]) signal is a one. 10: The GPIO[6] input (io_gpio_in[6]) signal generates a falling edge. 11: The GPIO[6] input (io_gpio_in[6]) signal generates a rising edge.
11:10	RW	0	General Purpose IRQ5 Type Bits. These bits determine the type of IRQ event that will generate an active gpio_irq[5] signal, as follows: 00: The GPIO[5] input (io_gpio_in[5]) signal is a zero. 01: The GPIO[5] input (io_gpio_in[5]) signal is a one. 10: The GPIO[5] input (io_gpio_in[5]) signal generates a falling edge. 11: The GPIO[5] input (io_gpio_in[5]) signal generates a rising edge.
9:8	RW	0	General Purpose IRQ4 Type bits. These bits determine the type of IRQ event that will generate an active gpio_irq[4] signal, as follows: 00: The GPIO[4] input (io_gpio_in[4]) signal is a zero. 01: The GPIO[4] input (io_gpio_in[4]) signal is a one. 10: The GPIO[4] input (io_gpio_in[4]) signal generates a falling edge. 11: The GPIO[4] input (io_gpio_in[4]) signal generates a rising edge.
7:6	RW	0	General Purpose IRQ3 Type Bits. These bits determine the type of IRQ event that will generate an active gpio_irq[3] signal, as follows: 00: The GPIO[3] input (io_gpio_in[3]) signal is a zero. 01: The GPIO[3] input (io_gpio_in[3]) signal is a one. 10: The GPIO[3] input (io_gpio_in[3]) signal generates a falling edge. 11: The GPIO[3] input (io_gpio_in[3]) signal generates a rising edge.
5:4	RW	0	General Purpose IRQ2 Type Bits. These bits determine the type of IRQ event that will generate an active gpio_irq[2] signal, as follows: 00: The GPIO[2] input (io_gpio_in[2]) signal is a zero. 01: The GPIO[2] input (io_gpio_in[2]) signal is a one. 10: The GPIO[2] input (io_gpio_in[2]) signal generates a falling edge. 11: The GPIO[2] input (io_gpio_in[2]) signal generates a rising edge.
3:2	RW	0	General Purpose IRQ1 Type Bits. These bits determine the type of IRQ event that will generate an active gpio_irq[1] signal, as follows: 00: The GPIO[1] input (io_gpio_in[1]) signal is a zero. 01: The GPIO[1] input (io_gpio_in[1]) signal is a one. 10: The GPIO[1] input (io_gpio_in[1]) signal generates a falling edge. 11: The GPIO[1] input (io_gpio_in[1]) signal generates a rising edge.

1:0	RW	0	General Purpose IRQ0 Type Bits. These bits determine the type of IRQ event that will generate an active gpio_irq[0] signal, as follows: 00: The GPIO[0] input (io_gpio_in[0]) signal is a zero. 01: The GPIO[0] input (io_gpio_in[0]) signal is a one. 10: The GPIO[0] input (io_gpio_in[0]) signal generates a falling edge. 11: The GPIO[0] input (io_gpio_in[0]) signal generates a rising edge.

Offset 0x0304 – 0x0307

GPIO Interrupt Request Status Register

This register provides the software to read and clear the status of the associated GPIO Interrupt Request.

Bit	Attribute	Default	Description
31:8	RO	0	Reserved
7	RW1C	-	General Purpose IRQ7 Edge Request Active. 0: No edge sensitive interrupt has been configured and detected on the associated signal (io_gpio_in[7]). 1: An edge sensitive interrupt request has been configured and detected on the associated signal (io_gpio_in[7]). Once this bit is active (a one), the software must write a one to this bit to clear it to inactive (a zero); otherwise this bit and the associated GPIO Interrupt Request (gpio_irq[7]) will remain active (a one).
6	RW1C	-	General Purpose IRQ6 Edge Request Active. 0: No edge sensitive interrupt has been configured and detected on the associated signal (io_gpio_in[6]). 1: An edge sensitive interrupt request has been configured and detected on the associated signal (io_gpio_in[6]). Once this bit is active (a one), the software must write a one to this bit to clear it to inactive (a zero); otherwise this bit and the associated GPIO Interrupt Request (gpio_irq[6]) will remain active (a one).
5	RW1C	-	General Purpose IRQ5 Edge Request Active. 0: No edge sensitive interrupt has been configured and detected on the associated signal (io_gpio_in[5]). 1: An edge sensitive interrupt request has been configured and detected on the associated signal (io_gpio_in[5]). Once this bit is active (a one), the software must write a one to this bit to clear it to inactive (a zero); otherwise this bit and the associated GPIO Interrupt Request (gpio_irq[5]) will remain active (a one).
4	RW1C	-	General Purpose IRQ4 Edge Request Active. 0: No edge sensitive interrupt has been configured and detected on the associated signal (io_gpio_in[4]). 1: An edge sensitive interrupt request has been configured and detected on the associated signal (io_gpio_in[4]). Once this bit is active (a one), the software must write a one to this bit to clear it to inactive (a zero); otherwise this bit and the associated GPIO Interrupt Request (gpio_irq[4]) will remain active (a one).
3	RW1C	-	General Purpose IRQ3 Edge Request Active. 0: No edge sensitive interrupt has been configured and detected on the associated signal (io_gpio_in[3]). 1: An edge sensitive interrupt request has been configured and detected on the associated signal (io_gpio_in[3]). Once this bit is active (a one), the software must write a one to this bit to clear it to inactive (a zero); otherwise this bit and the associated GPIO Interrupt Request (gpio_irq[3]) will remain active (a one).

2	RW1C	-	General Purpose IRQ2 Edge Request Active. 0: No edge sensitive interrupt has been configured and detected on the associated signal (io_gpio_in[2]). 1: An edge sensitive interrupt request has been configured and detected on the associated signal (io_gpio_in[2]). Once this bit is active (a one), the software must write a one to this bit to clear it inactive (a zero); otherwise this bit and the associated GPIO Interrupt Request (gpio_irq[2]) will remain active (a one).
1	RW1C	-	General Purpose IRQ1 Edge Request Active. 0: No edge sensitive interrupt has been configured and detected on the associated signal (io_gpio_in[1]). 1: An edge sensitive interrupt request has been configured and detected on the associated signal (io_gpio_in[1]). Once this bit is active (a one), the software must write a one to this bit to clear it to inactive (a zero); otherwise this bit and the associated GPIO Interrupt Request (gpio_irq[1]) will remain active (a one).
0	RW1C	-	General Purpose IRQ0 Edge Request Active. 0: No edge sensitive interrupt has been configured and detected on the associated signal (io_gpio_in[0]). 1: An edge sensitive interrupt request has been configured and detected on the associated signal (io_gpio_in[0]). Once this bit is active (a one), the software must write a one to this bit to clear it to inactive (a zero); otherwise this bit and the associated GPIO Interrupt Request (gpio_irq[0]) will remain active (a one).

Offset 0x0400 – 0x0403

Secure Digital I/O Drive Strength and Slew Rate Register

This register provides the software with the means to select the drive strength and slew rate of the Secure Digital I/O.

Bit	Attribute	Default	Description
31:2	RO	0	Reserved
1	RW	0	Secure Digital I/O Slew Rate Bit. This bit determines the slew rate associated with the Secure Digital I/O.
0	RW	0	Secure Digital I/O Drive Strength Bit. This bit determines the drive strength associated with the Secure Digital I/O.

Offset 0x0404 – 0x0407

VID I/O Drive Strength and Slew Rate Register

This register provides the software with the means to select the drive strength and slew rate of the VID I/O.

Bit	Attribute	Default	Description
31:2	RO	0	Reserved
1	RW	0	VID I/O Slew Rate Bit. This bit determines the slew rate associated with the VID I/O.
0	RW	0	VID I/O Drive Strength Bit. This bit determines the drive strength associated with the VID I/O.

Offset 0x0408 – 0x040B

SPI I/O Drive Strength and Slew Rate Register

This register provides the software with the means to select the drive strength and slew rate of the SPI I/O.

Bit	Attribute	Default	Description
31:2	RO	0	Reserved
1	RW	0	SPI I/O Slew Rate Bit. This bit determines the slew rate associated with the SPI I/O.
0	RW	0	SPI I/O Drive Strength Bit. This bit determines the drive strength associated with the SPI I/O.

Offset 0x040C – 0x040F

NOR I/O Drive Strength and Slew Rate Register

This register provides the software with the means to select the drive strength and slew rate of the NOR I/O.

Bit	Attribute	Default	Description
31:2	RO	0	Reserved
1	RW	0	NOR I/O Slew Rate Bit. This bit determines the slew rate associated with the NOR I/O.
0	RW	0	NOR I/O Drive Strength Bit. This bit determines the drive strength associated with the NOR I/O.

Offset 0x0410 – 0x0413

MMC I/O Drive Strength and Slew Rate Register

This register provides the software with the means to select the drive strength and slew rate of the MMC I/O.

Bit	Attribute	Default	Description
31:2	RO	0	Reserved
1	RW	0	MMC I/O Slew Rate Bit. This bit determines the slew rate associated with the MMC I/O.
0	RW	0	MMC I/O Drive Strength Bit. This bit determines the drive strength associated with the MMC I/O.

Offset 0x0414 – 0x0417

CLKTST I/O Drive Strength and Slew Rate Register

This register provides the software with the means to select the drive strength and slew rate of the CLKTST I/O.

Bit	Attribute	Default	Description
31:2	RO	0	Reserved
1	RW	0	CLKTST I/O Slew Rate Bit. This bit determines the slew rate associated with the CLKTST I/O.
0	RW	0	CLKTST I/O Drive Strength Bit. This bit determines the drive strength associated with the CLKTST I/O.

Offset 0x0418 – 0x041B

AC97/I^{2}S I/O Drive Strength and Slew Rate Register

This register provides the software with the means to select the drive strength and slew rate of the AC97/I^{2}S I/O.

Bit	Attribute	Default	Description
31:2	RO	0	Reserved
1	RW	0	AC97/I^{2}S I/O Slew Rate Bit. This bit determines the slew rate associated with the AC97/I^{2}S I/O.
0	RW	0	AC97/I^{2}S I/O Drive Strength Bit. This bit determines the drive strength associated with the AC97/I^{2}S I/O.

Offset 0x0500 – 0x0503

GPIO Enable Control Register for I^{2}C/PWM Signals

This register provides the software with the means to enable/disable the associated I^{2}C/PWM data signals to operate as General Purpose I/Os.

Bit	Attribute	Default	Description
31:6	RO	0	Reserved
5	RW	0	PWMOUT1 Signal GPIO Enable. 0: The corresponding PWMOUT1 signal will operate as the normal PWMOUT1 signal. 1: The corresponding PWMOUT1 signal will operate as a General Purpose I/O.
4	RW	0	PWMOUT0 Signal GPIO Enable. 0: The corresponding PWMOUT0 signal will operate as the normal PWMOUT0 signal. 1: The corresponding PWMOUT0 signal will operate as a General Purpose I/O.
3	RW	0	I2C1SDA Signal GPIO Enable. 0: The corresponding I2C1SDA signal will operate as the normal I2C1SDA signal. 1: The corresponding I2C1SDA signal will operate as a General Purpose I/O.
2	RW	0	I2C1SCL Signal GPIO Enable. 0: The corresponding I2C1SCL signal will operate as the normal I2C1SCL signal. 1: The corresponding I2C1SCL signal will operate as a General Purpose I/O.
1	RW	0	I2C0SDA Signal GPIO Enable. 0: The corresponding I2C0SDA signal will operate as the normal I2C0SDA signal. 1: The corresponding I2C0SDA signal will operate as a General Purpose I/O.
0	RW	0	I2C0SCL Signal GPIO Enable. 0: The corresponding I2C0SCL signal will operate as the normal I2C0SCL signal. 1: The corresponding I2C0SCL signal will operate as a General Purpose I/O.

Offset 0x0504 – 0x0507

GPIO Output Enable Control Register for I^{2}C/PWM Signals

This register provides the software with the means to enable/disable the output enable of the associated I^{2}C/PWM signals to operate as General Purpose I/Os.

Bit	Attribute	Default	Description
31:6	RO	0	Reserved
5	RW	0	PWMOUT1 Signal GPIO Output Enable. 0: When PWMOUT1 is enabled to be a GPIO, and this bit is inactive (a zero), the PWMOUT1 signal will operate as a General Purpose input. 1: When PWMOUT1 is enabled to be a GPIO, and this bit is active (a one), the PWMOUT1 signal will operate as a General Purpose output. When the corresponding PWMOUT1 is not enabled to be a GPIO, the value of this bit has no effect.
4	RW	0	PWMOUT0 Signal GPIO Output Enable. 0: When PWMOUT0 is enabled to be a GPIO, and this bit is inactive (a zero), the PWMOUT0 signal will operate as a General Purpose input. 1: When PWMOUT0 is enabled to be a GPIO, and this bit is active (a one), the PWMOUT0 signal will operate as a General Purpose output. When the corresponding PWMOUT0 is not enabled to be a GPIO, the value of this bit has no effect.
3	RW	0	I2C1SDA Signal GPIO Output Enable. 0: When I2C1SDA is enabled to be a GPIO, and this bit is inactive (a zero), the I2C1SDA signal will operate as a General Purpose input. 1: When I2C1SDA is enabled to be a GPIO, and this bit is active (a one), the I2C1SDA signal will operate as a General Purpose output. When the corresponding I2C1SDA is not enabled to be a GPIO, the value of this bit has no effect.
2	RW	0	I2C1SCL Signal GPIO Output Enable. 0: When I2C1SCL is enabled to be a GPIO, and this bit is inactive (a zero), the I2C1SCL signal will operate as a General Purpose input. 1: When I2C1SCL is enabled to be a GPIO, and this bit is active (a one), the I2C1SCL signal will operate as a General Purpose output. When the corresponding I2C1SCL is not enabled to be a GPIO, the value of this bit has no effect.
1	RW	0	I2C0SDA Signal GPIO Output Enable. 0: When I2C0SDA is enabled to be a GPIO, and this bit is inactive (a zero), the I2C0SDA signal will operate as a General Purpose input. 1: When I2C0SDA is enabled to be a GPIO, and this bit is active (a one), the I2C0SDA signal will operate as a General Purpose output. When the corresponding I2C0SDA is not enabled to be a GPIO, the value of this bit has no effect.
0	RW	0	I2C0SCL Signal GPIO Output Enable. 0: When I2C0SCL is enabled to be a GPIO, and this bit is inactive (a zero), the I2C0SCL signal will operate as a General Purpose input. 1: When I2C0SCL is enabled to be a GPIO, and this bit is active (a one), the I2C0SCL signal will operate as a General Purpose output. When the corresponding I2C0SCL is not enabled to be a GPIO, the value of this bit has no effect.

Offset 0x0508 – 0x050B

GPIO Output Data Register for I^{2}C/PWM Signals

This register provides the software with the means to control the output data values associated with the I^{2}C/PWM signals when they have been enabled as General Purpose outputs.

Bit	Attribute	Default	Description
31:6	RO	0	Reserved
5	RW	0	PWMOUT1 Signal GPIO Output Data. When PWMOUT1 is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When PWMOUT1 is not enabled to be a GPIO output, the value of this bit has no effect.
4	RW	0	PWMOUT0 Signal GPIO Output Data. When PWMOUT0 is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When PWMOUT0 is not enabled to be a GPIO output, the value of this bit has no effect.
3	RW	0	I2C1SDA Signal GPIO Output Data. When I2C1SDA is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When I2C1SDA is not enabled to be a GPIO output, the value of this bit has no effect.
2	RW	0	I2C1SCL Signal GPIO Output Data. When I2C1SCL is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When I2C1SCL is not enabled to be a GPIO output, the value of this bit has no effect.
1	RW	0	I2C0SDA Signal GPIO Output Data. When I2C0SDA is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When I2C0SDA is not enabled to be a GPIO output, the value of this bit has no effect.
0	RW	0	I2C0SCL Signal GPIO Output Data. When I2C0SCL is enabled to be a GPIO output, the value of this bit indicates the value to be driven on this signal. When I2C0SCL is not enabled to be a GPIO output, the value of this bit has no effect.

Offset 0x050C – 0x050F

GPIO Input Data Register for I^{2}C/PWM Signals

This register provides the software with the means to read the input data values associated with the I^{2}C/PWM signals.

Bit	Attribute	Default	Description
31:6	RO	0	Reserved
5	RO	0	PWMOUT1 Signal GPIO Input Data. The value read will always be the value on the associated signal.
4	RO	0	PWMOUT0 Signal GPIO Input Data. The value read will always be the value on the associated signal.
3	RO	0	I2C1SDA Signal GPIO Input Data. The value read will always be the value on the associated signal.
2	RO	0	I2C1SCL Signal GPIO Input Data. The value read will always be the value on the associated signal.
1	RO	0	I2C0SDA Signal GPIO Input Data. The value read will always be the value on the associated signal.
0	RO	0	I2C0SCL Signal GPIO Input Data. The value read will always be the value on the associated signal.

Power Management Control Registers

The module contains all of the control and status registers associated with Power Management.
The associated registers will be accessed via the 64 k byte address range assigned to the module.

Base address: 0xD813:0000

Offset 0x0000 – 0x0003

Power Management Status Lower Register

This register provides the software with the lower status of various Power Management operations.

Bit	Attribute	Default	Description
31:29	RO	0	Reserved.
28	RO	0	Updating SPI 2Clock Divisor. 0: The PMC module is not presently trying to update the SPI2 clock divisor. 1: The PMC module is presently trying to update the SPI 2clock divisor, and any attempt to write another new SPI 2clock divisor value will be ignored. Also, any attempt to write a new PLL Multiplier Value will be ignored.
27	RO	0	Updating SPI 1Clock Divisor. 0: The PMC module is not presently trying to update the SPI1 clock divisor. 1: The PMC module is presently trying to update the SPI 1clock divisor, and any attempt to write another new SPI 1clock divisor value will be ignored. Also, any attempt to write a new PLL Multiplier Value will be ignored.
26	RO	0	Updating SPI 0Clock Divisor. 0: The PMC module is not presently trying to update the SPI0 clock divisor. 1: The PMC module is presently trying to update the SPI 0clock divisor, and any attempt to write another new SPI0 clock divisor value will be ignored. Also, any attempt to write a new PLL Multiplier Value will be ignored.
25	RO	0	Updating NAND Clock Divisor. 0: The PMC module is not presently trying to update the NAND clock divisor. 1: The PMC module is presently trying to update the NAND clock divisor, and any attempt to write another new NAND clock divisor value will be ignored. Also, any attempt to write a new PLL Multiplier Value will be ignored.
24:23	RO	0	Reserved
22	RO	0	Updating SD/MMC Clock Divisor. 0: The PMC module is not presently trying to update the SD/MMC clock divisor. 1: The PMC module is presently trying to update the SD/MMC clock divisor, and any attempt to write another new SD/MMC clock divisor value will be ignored. Also, any attempt to write a new PLL Multiplier Value will be ignored.
21	RO	0	Reserved
20	RO	0	Updating GENET Clock Divisor. 0: The PMC module is not presently trying to update the GENET clock divisor. 1: The PMC module is presently trying to update the GENET clock divisor, and any attempt to write another new GENET clock divisor value will be ignored. Also, any attempt to write a new PLL Multiplier Value will be ignored.

19	RO	0	Updating Serial Flash Memory Controller Divisor. 0: The PMC module is not presently trying to update the Serial Flash Memory Controller clock divisor. 1: The PMC module is presently trying to update the Serial Flash Memory Controller clock divisor, and any attempt to write another new Serial Flash Memory Controller clock divisor value will be ignored. Also, any attempt to write a new PLL Multiplier Value will be ignored.
18	RO	0	Reserved
17	RO	0	Updating DVO Clock Divisor. 0: The PMC module is not presently trying to update the DVO clock divisor. 1: The PMC module is presently trying to update the DVO clock divisor, and any attempt to write another new DVO clock divisor value will be ignored. Also, any attempt to write a new PLL Multiplier Value will be ignored.
16	RO	0	Updating NGC Clock Divisor. 0: The PMC module is not presently trying to update the NGC clock divisor. 1: The PMC module is presently trying to update the NGC clock divisor, and any attempt to write another new NGC clock divisor value will be ignored. Also, any attempt to write a new PLL Multiplier Value will be ignored.
15	RO	0	Updating APB Clock Divisor 0: The PMC module is not presently trying to update the APB clock divisor. 1: The PMC module is presently trying to update the APB clock divisor, and any attempt to write another new APB clock divisor value will be ignored. Also, any attempt to write a new PLL Multiplier Value will be ignored.
14	RO	0	Updating PWM Clock Divisor. 0: The PMC module is not presently trying to update the PWM clock divisor. 1: The PMC module is presently trying to update the PWM clock divisor, and any attempt to write another new PWM clock divisor value will be ignored. Also, any attempt to write a new PLL Multiplier Value will be ignored.
13	RO	0	Updating KBDC Pre Clock Divisor. 0: The PMC module is not presently trying to update the Key board clock divisor. 1: The PMC module is presently trying to update the Key board controller 1st stage divisor, and any attempt to write another new Key board controller clock divisor value will be ignored. Also, any attempt to write a new PLL Multiplier Value will be ignored.
12	RO	0	Updating KBDC Clock Divisor. 0: The PMC module is not presently trying to update the Key board clock divisor. 1: The PMC module is presently trying to update the Key board clock 2nd stage divisor, and any attempt to write another new Key board clock divisor value will be ignored. Also, any attempt to write a new PLL Multiplier Value will be ignored.
11	RO	0	Updating DDR Memory Controller Clock Divisor. 0: The PMC module is not presently trying to update the DDR Memory Controller clock divisor. 1: The PMC module is presently trying to update the DDR Memory Controller clock divisor, and any attempt to write another new DDR Memory Controller clock divisor value will be ignored. Also, any attempt to write a new PLL Multiplier Value will be ignored.

10	RO	0	Updating NA0 Clock Divisor. 0: The PMC module is not presently trying to update the NA0 clock divisor. 1: Active; the PMC module is presently trying to update the NA0 clock divisor, and any attempt to write another new NA0 clock divisor value will be ignored. Also, any attempt to write a new PLL Multiplier Value will be ignored.
9	RO	0	Updating NA12 Clock Divisor. 0: The PMC module is not presently trying to update the NA12 clock divisor. 1: The PMC module is presently trying to update the NA12 clock divisor, and any attempt to write another new NA12 clock divisor value will be ignored. Also, any attempt to write a new PLL Multiplier Value will be ignored.
8	RO	0	Updating AHB Clock Divisor. 0: The PMC module is not presently trying to update the AHB clock divisor. 1: The PMC module is presently trying to update the AHB clock divisor, and any attempt to write another new AHB clock divisor value will be ignored. Also, any attempt to write a new PLL Multiplier Value will be ignored.
7	RO	0	Updating ARM Clock Divisor. 0: The PMC module is not presently trying to update the ARM clock divisor. 1: The PMC module is presently trying to update the ARM clock divisor, and any attempt to write another new ARM clock divisor value will be ignored. Also, any attempt to write a new PLL Multiplier Value will be ignored.
6	RO	0	Updating I2C1 Clock Divisor. 0: The PMC module is not presently trying to update the I2C1 clock divisor. 1: The PMC module is presently trying to update the I2C1 clock divisor, and any attempt to write another new I2C1 clock divisor value will be ignored. Also, any attempt to write a new PLL Multiplier Value will be ignored.
5	RO	0	Updating I2C0 Clock Divisor. 0: The PMC module is not presently trying to update the I2C0 clock divisor. 1: The PMC module is presently trying to update the I2C0 clock divisor, and any attempt to write another new I2C0 clock divisor value will be ignored. Also, any attempt to write a new PLL Multiplier Value will be ignored.
4	RO	0	Updating ANY_PLL_MULTIPLIER. 0: The PMC module is not presently trying to update a PLL multiplier. 1: The PMC module is presently trying to update a PLL multiplier, and any attempt to write clock divisor values will be ignored. Also, any attempt to write a PLL Multiplier Value will be ignored.
3	RO	0	Updating ANY_CLOCK_DIVISOR. 0: The PMC module is not presently trying to update a clock divisor. 1: The PMC module is presently trying to update a clock divisor, and any attempt to write the applicable clock divisor value(s) will be ignored. Also, any attempt to write a PLL Multiplier Value will be ignored.
2	RO	0	Hibernation Operation Active. 0: The PMC module is not presently trying to enter Hibernation. 1: The PMC module is presently trying to enter Hibernation.
1	RO	0	IDLE Operation Active. 0: The PMC module is not presently trying to IDLE the ARM. 1: The PMC module is presently trying to IDLE the ARM.

0	RO	0	Real Time Clock (RTC) Clocked Logic Disabled. 0: The PMC module is fully functional. 1: The RTC clocked logic that supports Hibernation and the changing of the PLL Multiplier Values is disabled, and all such requests will simply be ignored. This condition occurs if the RTC oscillator is not running. Note: Following a reset (*pmc_sus_reset_x*), it may take up to 62 microseconds for the detection logic to recognize that the RTC clock is running. As a result, this bit may remain a one for up to 62 microseconds, until the logic can detect that the RTC clock is running. The initialization software should be written such that it only relies on the value of this bit after more than 62 microseconds have passed since the desertion of the *pmc_sus_reset_x* input signal.

Offset 0x0004

Power Management Status Higher Register

This register is to provide software with the higher status of various Power Management operations.

Bit	Attribute	Default	Description
31:4	RO	0	Reserved.
3	RO	0	Updating PLL_A Multiplier Value. 0: The PMC module is not presently trying to update the PLL_A Multiplier Value. 1: The PMC module is presently trying to update the PLL_A Multiplier Value, and any attempt to write another new PLL_A Multiplier Value will be ignored. Also, any attempt to write any new clock divisor value will be ignored.
2	RO	0	Updating PLL_B Multiplier Value. 0: The PMC module is not presently trying to update the PLL_B Multiplier Value. 1: The PMC module is presently trying to update the PLL_B Multiplier Value, and any attempt to write another new PLL_B Multiplier Value will be ignored. Also, any attempt to write any new clock divisor value will be ignored.
1	RO	0	Updating PLL_C Multiplier Value. 0: The PMC module is not presently trying to update the PLL_C Multiplier Value. 1: The PMC module is presently trying to update the PLL_C Multiplier Value, and any attempt to write another new PLL_C Multiplier Value will be ignored. Also, any attempt to write any new clock divisor value will be ignored.
0	RO	0	Updating PLL_D Multiplier Value. 0: The PMC module is not presently trying to update the PLL_D Multiplier Value. 1: The PMC module is presently trying to update the PLL_D Multiplier Value, and any attempt to write another new PLL_D Multiplier Value will be ignored. Also, any attempt to write any new clock divisor value will be ignored.

Offset 0x0008: Reserved

Offset 0x000C – 0x000D

Power-up / Power Enable Time Control Register

The Power-up Time Control Register allows the software to increase the power-up time associated with exiting the Suspend or Shutdown condition.

The power-up time is defined as the amount of time from the *pmc_sus_pwren* signal being driven active (a one) until the associated normal digital logic input voltage is stable. The PMC module will guarantee the normal digital logic appropriately held in reset throughout the power-up time. By default the power-up time will be approximately 1 ms, but with this register that time may be expanded to a maximum of approximately 0.5 sec.

Note:

After power-up, the PMC module will hold the normal digital logic in reset for an additional 1 ms, which is considered the PLL stabilization time.

The value of this register after a power-up reset (*pmc_sus_reset_x*) caused by a resume reset will be all zeros.

The value of this register after a power-up reset not caused by a resume reset will be to reload any previously programmed value.

The value of this register after a non power-up reset is not affected.

Bit	Attribute	Default	Description
9:8	RW	0	Power Enable Time Control Bits. The default reset value of these bits will be all zeros and that corresponds to a PWREN33/PWREN18 delay time of approximately 2 ms from PWREN15. The power enable delay time provided by the PMC module's logic may be calculated as follows: The approximate power-up time = (((the value of these bits)) x2) ms
7:0	RW	0	Power-up Time Control Bits. The default reset value of these bits will be all zeros and that corresponds to a power-up time of approximately 1 ms. The power-up time provided by the PMC module's logic may be calculated as follows: The approximate power-up time = ((2 x (the value of these bits)) + 1) ms

Offset 0x0010 – 0x0011

Hibernation Value Register

This register allows the software to save the 16 bits of data when entering a Power-off Hibernation Mode. The Power-off Hibernation Modes include the Suspend and Shutdown Modes.

The value of this register after a power-up reset (*pmc_sus_reset_x*) caused by a resume reset will be all zeros.

The value of this register after a power-up reset not caused by a resume reset will be to reload any previously programmed value.

The value of this register after a non power-up reset is not affected.

This register may be written in conjunction with the writing of the Hibernation Control Register by using a 4-byte access.

Bit	Attribute	Default	Description
15:0	RW	0	Read/Write Data.

Offset 0x0012 – 0x0013

Hibernation Control Register

This register allows the software to request the processor to enter a Hibernation Mode. The Hibernation Modes include the Sleep, Suspend, and Shutdown Modes.

The value of this register after a power-up reset (*pmc_sus_reset_x*) will be all zeros.

The value of this register after a non power-up reset is not affected.

Bit	Attribute	Default	Description
15:3	RO	0	Reserved.
2:0	RW	0	Hibernation Mode Control. 011: Sleep Request (A Power-on Hibernation Mode) 001: Suspend Request (A Power-off Hibernation Mode) 101: Shutdown Request (A Power-off Hibernation Mode) Note: Various wake-up events may be configured to wake the processor. The wake-up events must be configured before these bits are set.

Offset 0x0014 – 0x0015

Wake-up Status Register

This register allows the software the ability to see the wake-up event(s) that are active after exiting a Hibernation Mode.

Bit	Attribute	Default	Description
15	RW1C	0	Real Time Clock (RTC) Wake-up Status. This bit indicates that the RTC Alarm Interrupt (*rtc_intr*) wake-up event has occurred. To clear this bit to a zero, the software must write a one to this bit location.
8	RO	0	Reserved.
14	RW1C	0	Power Button Wake-up Status. This bit indicates that the Power Button wake-up event has occurred. To clear this bit to a zero, the software must write a one to this bit location.
13	RW1C	0	USB Device Attach Wakeup Status.
12	RW1C	0	USB Host Wakeup Status.
11	RW1C	0	Mouse Wakeup Status.
10	RW1C	0	Keyboard Controller Wakeup Status.
9	RW1C	0	Ethernet Wakeup Status.
8	R	0	Reserved.
7:0	RW1C	0	General Purpose Wake-up Status Bits 7-0. When one of these bits is active (a one), the associated general purpose wake-up event has occurred. To clear one of these bits to a zero, the software must write a one to that bit location.

Offset 0x001C – 0x001D

Wake-up Event Enable Register

This register allows the software the ability to enable wake-up event(s) for exiting Hibernation Modes.

The value of this register after a power-up reset (*pmc_sus_reset_x*) caused by a resume reset will be all zeros.

The value of this register after a power-up reset not caused by a resume reset will be to reload any previously programmed value.

The value of this register after a non power-up reset is not affected.

Bit	Attribute	Default	Description
15	RW	0	Real Time Clock (RTC) Wake-up Enable. 1: The RTC Interrupt (*rtc_intr*) is a wake-up event.
14	RO	0	Reserved.
13	RW	0	USB Device Wake-up Enable.
12	RW	0	USB Host Wake-up Enable.
11	RW	0	Mouse Controller Wake-up Enable.
10	RW	0	Keyboard Controller Wake-up Enable.
9	RW	0	Ethernet Wake-up Enable.
8	RO	0	Reserved.
7:0	RW	0	General Purpose Wake-up Enable Bits 7-0. When one of these bits is active (a one), it indicates that the associated general purpose wake-up event is enabled. The type of wake-up event is determined by the associated entries in the Wake-up Event Type register. Bit[7:4]: For internal IPs wakeup event in Sleep mode. Bit[2:0]: For external wakeup pins input.

Offset 0x0020 – 0x0023

Wake-up Event Type Register

This register allows the software the ability to configure wake-up event types for exiting Hibernation Modes.

The value of this register after a power-up reset (*pmc_sus_reset_x*) caused by a resume reset will be all zeros.

The value of this register after a power-up reset not caused by a resume reset will be to reload any previously programmed value.

The value of this register after a non power-up reset is not affected.

Bit	Attribute	Default	Description
31:28	RW	0	General Purpose Wake-up 7 Type Bits. When the corresponding General Purpose Wake-up Enable bit is active (a one), these bits determine the type of wake-up event as follows: 0000: The input wake-up signal is a zero. 0001: The input wake-up signal is a one. 0010: The input wake-up signal generates a falling edge. 0011: The input wake-up signal generates a rising edge. 0100: The input wake-up signal generates an edge. All Other Cases: Reserved.

27:24	RW	0	General Purpose Wake-up 6 Type Bits. When the corresponding General Purpose Wake-up Enable bit is active (a one), these bits determine the type of wake-up event as follows: 0000: The input wake-up signal is a zero. 0001: The input wake-up signal is a one. 0010: The input wake-up signal generates a falling edge. 0011: The input wake-up signal generates a rising edge. 0100: The input wake-up signal generates an edge. All Other Cases: Reserved.
23:20	RW	0	General Purpose Wake-up 5 Type Bits. When the corresponding General Purpose Wake-up Enable bit is active (a one), these bits determine the type of wake-up event as follows: 0000: The input wake-up signal is a zero. 0001: The input wake-up signal is a one. 0010: The input wake-up signal generates a falling edge. 0011: The input wake-up signal generates a rising edge. 0100: The input wake-up signal generates an edge. All Other Cases: Reserved.
19:16	RW	0	General Purpose Wake-up 4 Type Bits. When the corresponding General Purpose Wake-up Enable bit is active (a one), these bits determine the type of wake-up event as follows: 0000: The input wake-up signal is a zero. 0001: The input wake-up signal is a one. 0010: The input wake-up signal generates a falling edge. 0011: The input wake-up signal generates a rising edge. 0100: The input wake-up signal generates an edge. All Other Cases: Reserved.
15:12	RW	0	General Purpose Wake-up 3 Type Bits. When the corresponding General Purpose Wake-up Enable bit is active (a one), these bits determine the type of wake-up event as follows: 0000: The input wake-up signal is a zero. 0001: The input wake-up signal is a one. 0010: The input wake-up signal generates a falling edge. 0011: The input wake-up signal generates a rising edge. 0100: The input wake-up signal generates an edge. All Other Cases: Reserved.
11:8	RW	0	General Purpose Wake-up 2 Type Bits. When the corresponding General Purpose Wake-up Enable bit is active (a one), these bits determine the type of wake-up event as follows: 0000: The input wake-up signal is a zero. 0001: The input wake-up signal is a one. 0010: The input wake-up signal generates a falling edge. 0011: The input wake-up signal generates a rising edge. 0100: The input wake-up signal generates an edge. All Other Cases: Reserved.
7:4	RW	0	General Purpose Wake-up 1 Type Bits. When the corresponding General Purpose Wake-up Enable bit is active (a one), these bits determine the type of wake-up event as follows: 0000: The input wake-up signal is a zero. 0001: The input wake-up signal is a one. 0010: The input wake-up signal generates a falling edge. 0011: The input wake-up signal generates a rising edge. 0100: The input wake-up signal generates an edge. All Other Cases: Reserved.

3:0	RW	0	General Purpose Wake-up 0 Type Bits. When the corresponding General Purpose Wake-up Enable bit is active (a one), these bits determine the type of wake-up event as follows: 0000: The input wake-up signal is a zero. 0001: The input wake-up signal is a one. 0010: The input wake-up signal generates a falling edge. 0011: The input wake-up signal generates a rising edge. 0100: The input wake-up signal generates an edge. All Other Cases: Reserved.

Offset 0x0030 – 0x0033

Hibernation Scratch Pad Register 0

This register and the other Hibernation Scratch Pad registers provide the software with a scratch pad area to store off values desirable to maintain during Hibernation Modes.

The value of this register after a power-up reset (*pmc_sus_reset_x*) caused by a resume reset will be all zeros.

The value of this register after a power-up reset not caused by a resume reset will be to reload any previously programmed value.

The value of this register after a non power-up reset is not affected.

Bit	Attribute	Default	Description
31:0	RW	0	Scratch Pad 0 Data. These bits are general purpose scratch pad read write bits.

Offset 0x0034 – 0x0037

Hibernation Scratch Pad Register 1

This register and the other Hibernation Scratch Pad registers provide the software with a scratch pad area to store off values desirable to maintain during Hibernation.

The value of this register after a power-up reset (*pmc_sus_reset_x*) caused by a resume reset will be all zeros.

The value of this register after a power-up reset not caused by a resume reset will be to reload any previously programmed value.

The value of this register after a non power-up reset is not affected.

Bit	Attribute	Default	Description
31:0	RW	0	Scratch Pad 1 Data. These bits are general purpose scratch pad read write bits.

Offset 0x0038 – 0x003B

Hibernation Scratch Pad Register 2

This register and the other Hibernation Scratch Pad registers provide the software with a scratch pad area to store off values desirable to maintain during Hibernation.

The value of this register after a power-up reset (*pmc_sus_reset_x*) caused by a resume reset will be all zeros.

The value of this register after a power-up reset not caused by a resume reset will be to reload any previously programmed value.

The value of this register after a non power-up reset is not affected.

Bit	Attribute	Default	Description
31:0	RW	0	Scratch Pad 2 Data. These bits are general purpose scratch pad read write bits.

Offset 0x003C – 0x003F

Hibernation Scratch Pad Register 3

This register and the other Hibernation Scratch Pad registers is to provide the software with a scratch pad area to store off values desirable to maintain during Hibernation.

The value of this register after a power-up reset (*pmc_sus_reset_x*) caused by a resume reset will be all zeros.

The value of this register after a power-up reset not caused by a resume reset will be to reload any previously programmed value.

The value of this register after a non power-up reset is not affected.

Bit	Attribute	Default	Description
31:0	RW	0	Scratch Pad 3 Data. These bits are general purpose scratch pad read write bits.

Offset 0x0050 – 0x0053

Reset Status Register

This register provides the software with a means to determine the source of a reset.

Bit	Attribute	Default	Description
31:7	RO	0	Reserved.
6	RW1C	0	Power Good Reset Bit. 0: A Power Good Reset has not occurred since this bit last cleared by the software. 1: A Power Good Reset has occurred since this bit last cleared by the software. To clear this bit as inactive (a zero), the software must write a one to this bit. This bit may be active (a one) as other bits in this register are also active (ones). When Bit 0 of this register is read as a one (active), this bit will always be read inactive (a zero).
5	RW1C	0	Shutdown Reset Bit. 0: A Shutdown Reset has not occurred since this bit last cleared by the software. 1: A Shutdown Reset has occurred since this bit last cleared by the software. To clear this bit as inactive (a zero), the software must write a one to this bit. This bit may be active (a one) as other bits in this register are also active (ones). When Bit 0 of this register is read as a one (active), this bit will always be read inactive (a zero).
4	RW1C	0	Software Reset Bit. 0: A Software Reset has not occurred since this bit last cleared by the software. 1: A Software Reset has occurred since this bit last cleared by the software. To clear this bit as inactive (a zero), the software must write a one to this bit. This bit may be active (a one) as other bits in this register are also active (ones). When Bit 0 of this register is read as a one (active), this bit will always be read inactive (a zero).

3	RW1C	0	Watchdog Reset Bit. 0: A Watchdog Reset has not occurred since this bit last cleared by the software. 1: A Watchdog Reset has occurred since this bit was last cleared by software. To clear this bit as inactive (a zero), the software must write a one to this bit. This bit may be active (a one) as other bits in this register are also active (ones). When Bit 0 of this register is read as a one (active), this bit will always be read inactive (a zero).
2	RW1C	0	Suspend Reset Bit. 0: A Suspend Reset has not occurred since this bit last cleared by the software. 1: A Suspend Reset has occurred since this bit last cleared by the software. To clear this bit as inactive (a zero), the software must write a one to this bit. This bit may be active (a one) as other bits in this register are also active (ones). When Bit 0 of this register is read as a one (active), this bit will always be read inactive (a zero).
1	RO	0	Reserved This bit is read-only zero.
0	RW1C	1b	Power Management Reset (Resume Reset) Bit. 0: A Power Management Reset has not occurred since this bit last cleared by software. 1: A Power Management Reset has occurred since this bit last cleared by software. To clear this bit as inactive (a zero), the software must write a one to this bit.

Offset 0x0054 – 0x0057

Power Button Control Register

This register provides the software with a means to control the behavior of the Power Button in support of Hard and Soft Power capabilities.

The value of this register after a power-up reset (*pmc_sus_reset_x*) caused by a resume reset will be all zeros.

The value of this register after a power-up reset not caused by a resume reset will be to reload any previously programmed value.

The value of this register after a non power-up reset is not affected.

Bit	Attribute	Default	Description
31:1	RO	0	Reserved.
2	RW	0	Suspend to Dram Bit. 0: The hardware will turn-off DDR power while in power-off hibernation modes. 1: The hardware will support suspend to DRAM by keeping DDR power and DDR IO in proper state while in power-off hibernation modes (suspend or shutdown modes).
1	RW	0	MII IO ON Bit. 0: It will inhibit all MII IO outputs and gated their inputs to core. This will stop the connection to external Ethernet PHY. 1: The MII IO are all active to support MII connection
0	RW	0	Soft Power Enable Bit. 0: The Power Button will behave to support Hard Power features. 1: The Power Button will behave to support Soft Power features.

Offset 0x0060 – 0x0063

Software Request Reset Register

This register provides the software with a means to generate a reset.

Bit	Attribute	Default	Description
31:1	RO	0	Reserved.
0	RW	0	Software Reset Request Bit. 0: No software reset request is pending. 1: The software is making a software reset request. When the software request takes place, this bit will be self-cleared inactive (a zero).

Offset 0x0100 – 0x0103

OS Timer Match Register 0

This register allows the software the ability to set an OS Timer match value to generate an interrupt request or to generate a Watchdog reset.

The value of this register after any reset will be all zeros.

Bit	Attribute	Default	Description
31:0	RW	0	OS Timer Match Value 0. When these bits match the OS Timer count value and the interrupt is enabled (a one) for OS Timer channel 0, the OS Timer Channel 0 interrupt request (*pmc_os_timer_irq[0]*) signal will be active (a one). When these bits match the OS Timer count value and the Watchdog reset is enabled (a one) for OS Timer channel 0, the Watchdog reset will occur by driving the *pmc_reset_x* signal active (a zero) for four 12-MHz clock periods.

Offset 0x0104 – 0x0107

OS Timer Match Register 1

This register allows the software the ability to set an OS Timer match value to generate an interrupt request.

The value of this register after any reset will be all zeros.

Bit	Attribute	Default	Description
31:0	RW	0	OS Timer Match Value 1. When these bits match the OS Timer count value and the interrupt is enabled (a one) for OS Timer channel 1, the OS Timer Channel 1 interrupt request (*pmc_os_timer_irq[1]*) signal will be active (a one).

Offset 0x0108 – 0x010B

OS Timer Match Register 2

This register allows the software the ability to set an OS Timer match value to generate an interrupt request.

The value of this register after any reset will be all zeros.

Bit	Attribute	Default	Description
31:0	RW	0	OS Timer Match Value 2. When these bits match the OS Timer count value and the interrupt is enabled (a one) for OS Timer channel 2, the OS Timer Channel 2 interrupt request (*pmc_os_timer_irq[2]*) signal will be active (a one).

Offset 0x010C – 0x010F

OS Timer Match Register 3

This register allows the software the ability to set an OS Timer match value to generate an interrupt request.

The value of this register after any reset will be all zeros.

Bit	Attribute	Default	Description
31:0	RW	0	OS Timer Match Value 3. When these bits match the OS Timer count value and the interrupt is enabled (a one) for OS Timer channel 3, then the OS Timer Channel 3 interrupt request (*pmc_os_timer_irq[3]*) signal will be active (a one).

Offset 0x0110 – 0x0113

OS Timer Count Register

This register allows software to read and write the present OS Timer count value.

When writing this register, it may take a short amount of time before the value is loaded into the OS Timer which is running off of the input 12-MHz Clock.

To guarantee that a valid value is read from this register, the software should write a one to the OS Timer Read Count Request bit (Bit 1 of the OS Timer Control register at address offset 0x120), and then poll the OS Timer Access Status register until the OS Timer Read Count Active status bit (Bit 5 of the OS Timer Register Access Status register at address offset 0x0124) is inactive (a zero).

The value of this register after any reset will be all zeros.

Bit	Attribute	Default	Description
31:0	RW	0	OS Timer Count Value. Writing this register results in loading a new OS Timer count value. Reading from this register provides the present OS Timer count value. Follow the recommended read sequence that the value read via this register will be guaranteed to be a valid value. **IMPORTANT NOTE**: Reads from the OS Timer Count Register will always read the last value stored off resulting from the software writing a one to the OS Timer Read Count Request bit (Bit 1 of the OS Timer Control Register).

Offset 0x0114 – 0x0117

OS Timer Status Register

This register provides the software with the present match status of the OS Timer channels.

The value of this register after any reset will be all zeros.

Bit	Attribute	Default	Description
31:4	RO	0	Reserved.
3	RW1C	0	OS Timer Channel 3 Match Status Bit. 1: The OS Timer Match 3 value has matched the present OS Timer count value while the OS Timer Channel 3 Interrupt Enable bit (Bit 3 of the OS Timer Interrupt Enable Register) is a one. The software must write a one to this bit to clear it to its inactive value (a zero). Note: For the hardware to set this bit active (a one), the OS Timer Channel 3 Interrupt Enable bit must be active (a one). If the OS Timer Channel 3 Interrupt Enable bit is programmed to be inactive (a zero), while this bit is active (a one), this bit will also be cleared to a zero (inactive) as a result.

2	RW1C	0	OS Timer Channel 2 Match Status Bit. 1: The OS Timer Match 2 value has matched the present OS Timer count value while the OS Timer Channel 2 Interrupt Enable bit (Bit 2 of the OS Timer Interrupt Enable Register) is a one. The software must write a one to this bit to clear it to its inactive value (a zero). Note: For the hardware to set this bit active (a one), the OS Timer Channel 2 Interrupt Enable bit must be active (a one). If the OS Timer Channel 2 Interrupt Enable bit is programmed to be inactive (a zero), while this bit is active (a one), this bit will also be cleared to a zero (inactive) as a result.
1	RW1C	0	OS Timer Channel 1 Match Status Bit. 1: The OS Timer Match 1 value has matched the present OS Timer count value while the OS Timer Channel 1 Interrupt Enable bit (Bit 1 of the OS Timer Interrupt Enable Register) is a one. The software must write a one to this bit to clear it to its inactive value (a zero). Note: For the hardware to set this bit active (a one), the OS Timer Channel 1 Interrupt Enable bit must be active (a one). If the OS Timer Channel 1 Interrupt Enable bit is programmed to be inactive (a zero), while this bit is active (a one), this bit will also be cleared to a zero (inactive) as a result.
0	RW1C	0	OS Timer Channel 0 Match Status Bit. 1: The OS Timer Match 0 value has matched the present OS Timer count value while the OS Timer Channel 0 Interrupt Enable bit (Bit 0 of the OS Timer Interrupt Enable Register) is a one, and the OS Timer Channel 0 Watchdog Enable bit (Bit 0 of the OS Timer Interrupt Enable Register) is inactive (a zero). The software must write a one to this bit to clear it to its inactive value (a zero). Note: For the hardware to set this bit active (a one), the OS Timer Channel 0 Interrupt Enable bit must be active (a one) and the OS Timer Channel 0 Watchdog Enable bit must be inactive (a zero). If the OS Timer Channel 0 Interrupt Enable bit is programmed to be inactive (a zero), while this bit is active (a one), this bit will also be cleared to a zero (inactive) as a result.

Offset 0x0118 – 0x011B

OS Timer Watchdog Enable Register

This register provides the software with a means to enable/disable the Watchdog reset associated with the OS Timer Channel 0.

The value of this register after any reset will be all zeros.

Bit	Attribute	Default	Description
31:1	RO	0	Reserved.
0	RW	0	OS Timer Channel 0 Watchdog Enable Bit. 0: A Watchdog reset will not occur if the OS Timer Match 0 value has matched the present OS Timer count value. 1: A Watchdog reset will occur if the OS Timer Match 0 value has matched the present OS Timer count value. Note: When this bit is active (a one), the value of Bit 0 of the OS Timer Interrupt Enable Register has no effect because in such a case only a Watchdog reset will occur as the OS Timer Match 0 value has matched the present OS Timer count value, and no interrupt request will be made.

Offset 0x011C – 0x011F

OS Timer Interrupt Enable Register

This register provides the software with a means to enable/disable the interrupt requests associated with the OS Timer channels.

The value of this register after any reset will be all zeros. The value of this register after a non power-up reset is not affected.

Bit	Attribute	Default	Description
31:4	RO	0	Reserved.
3	RW	0	OS Timer Channel 3 Interrupt Enable Bit. 0: The OS Timer Channel 3 interrupt request will be disabled. It will also force the OS Timer Channel 3 Match Status bit (Bit 3 of the OS Timer Status Register) inactive (a zero). 1: An interrupt request will be made when OS Timer Match 3 value has matched the present OS Timer count value.
2	RW	0	OS Timer Channel 2 Interrupt Enable Bit. 0: The OS Timer Channel 2 interrupt request will be disabled. It will also force the OS Timer Channel 2 Match Status bit (Bit 2 of the OS Timer Status Register) inactive (a zero). 1: An interrupt request will be made when OS Timer Match 2 value has matched the present OS Timer count value.
1	RW	0	OS Timer Channel 1 Interrupt Enable Bit. 0: The OS Timer Channel 1 interrupt request will be disabled. It will also force the OS Timer Channel 1 Match Status bit (Bit 1 of the OS Timer Status Register) inactive (a zero). 1: An interrupt request will be made when OS Timer Match 1 value has matched the present OS Timer count value.
0	RW	0	OS Timer Channel 0 Interrupt Enable Bit. 0: The OS Timer Channel 0 interrupt request will be disabled. It will also force the OS Timer Channel 0 Match Status bit (Bit 0 of the OS Timer Status Register) inactive (a zero). 1: An interrupt request will be made when OS Timer Match 0 value has matched the present OS Timer count value. Note: When Bit 0 of the OS Timer Watchdog Enable Register is active (a one), the value of this bit has no effect because in such a case only a Watchdog reset will occur as the OS Timer Match 0 value has matched the present OS Timer count value, and no interrupt request will be made.

Offset 0x0120 – 0x0123

OS Timer Control Register

This register provides the software with a means to control the OS Timer functions.

The value of this register after any reset will be all zeros.

Bit	Attribute	Default	Description
31:2	RO	0	Reserved.
1	RW	0	OS Timer Read Count Request Bit. 0: The software is not making an OS Timer Read Count request. 1: The software is making an OS Timer Read Count request. This bit will remain active (a one) until the OS Timer Read Count Active status bit (Bit 5 of the OS Timer Register Access Status register) is cleared inactive (a zero) to indicate that the value that can be read via the OS Timer Count Register will be a valid value. When the OS Timer Read Count Active status bit is cleared inactive (a zero), this bit will be cleared inactive (a zero).

0	RW	0	OS Timer Enable Bit. 0: It will disable the OS Timer that will cause the present count value to be held. A new value may be written via the OS Timer Count Register while the OS Timer is disabled, and after which the newly written value will be held as the present count while the OS Timer is disabled. 1: The OS Timer will be enabled and incremented by 1'b1 every 3-MHz clock, except when it is loaded with a new value as the OS Timer Count Register is written.

Offset 0x0124 – 0x0127

OS Timer Access Status Register

This register is to provide the software with a means to determine if an access to the OS Timer registers has completed.

This is a read-only register. Writing to this register will be ignored.

Bit	Attribute	Default	Description
31:6	RO	0	Reserved.
5	RO	0	OS Timer Read Count Active Bit. 0: All previous read requests via the OS Timer Read Count Request bit have completed and the next value read from the OS Timer Count Register will be valid. 1: A previous read request initiated via the OS Timer Read Count Request bit (Bit 1 of the OS Timer Control Register) is still being performed, and any read of the OS Timer Count Register may not read a valid value. **IMPORTANT NOTE:** The value read from the OS Timer Count Register will always be the last value stored off as a result of the software writing a one to the OS Timer Read Count Request bit (Bit 1 of the OS Timer Control Register).
4	RO	0	OS Timer Count Write Active Bit. 0: All previous writes to the OS Timer Count Register have completed and software may write a new value to the OS Timer Count Register at any time. 1: A previous write to the OS Timer Count Register is still being performed, and any attempt to write the OS Timer Count Register will be ignored.
3	RO	0	OS Timer Match 3 Write Active Bit. 0: All previous writes to the OS Timer Match 3 Register have completed and software may write a new value to the OS Timer Match 3 Register at any time. 1: A previous write to the OS Timer Match 3 Register is still being performed, and any attempt to write the OS Timer Match 3 Register will be ignored.
2	RO	0	OS Timer Match 2 Write Active Bit. 0: All previous writes to the OS Timer Match 2 Register have completed and software may write a new value to the OS Timer Match 2 Register at any time. 1: A previous write to the OS Timer Match 2 Register is still being performed, and any attempt to write the OS Timer Match 2 Register will be ignored.
1	RO	0	OS Timer Match 1 Write Active Bit. 0: All previous writes to the OS Timer Match 1 Register have completed and software may write a new value to the OS Timer Match 1 Register at any time. 1: A previous write to the OS Timer Match 1 Register is still being performed, and any attempt to write the OS Timer Match 1 Register will be ignored.

0	RO	0	OS Timer Match 0 Write Active Bit. 0: All previous writes to the OS Timer Match 0 Register have completed and software may write a new value to the OS Timer Match 0 Register at any time. 1: A previous write to the OS Timer Match 0 Register is still being performed, and any attempt to write the OS Timer Match 0 Register will be ignored.

Offset 0x0200 – 0x0203

PLL_A Multiplier and Range Values Register

This register allows the software to change the PLL_A Multiplier value.

IMPORTANT NOTE:

Writing this register will have no effect on the PLL_A when the Real Time Clock (RTC) oscillator is not running.

The value of this register after a power-up reset (*pmc_sus_reset_x*) will be all zeros.

The value of this register after a non power-up reset is not affected.

Bit	Attribute	Default	Description
31:20	RO	0	Reserved.
19:18	RW	0	PLL_A VCO Range Value Bits. These bits provide extra range control for the PLL_A's VCO. Unless otherwise instructed the software should always write zeros to these bits.
17:16	RW	0	PLL_A Charge-pump Range Value Bits. These bits provide extra range control for the PLL_A's charge pump. Unless otherwise instructed software should always write zeros to these bits.
15:9	RO	0	Reserved.
8	RW	0	PLL_A Pre-Divisor Bypass. This bit indicates the PLL_A Pre-Divisor Bypass status. 0: The Pre-Divisor is enabled and will divide the source clock to the PLL_A by two before it is used in PLL_A operations. 1: The Pre-Divisor is bypassed and the source clock to the PLL_A will be passed through for use in PLL_A operations.
7:5	RO	0	Reserved.
4:0	RW	0	PLL_A Multiplier Value Bits 4-0. These bits indicate the PLL Multiplier value which will be used to be multiplied by the input oscillator clock. The resulting PLL multiplier selected by these signals are as below: 00000: Bypass PLL 00001: Bypass PLL 00010: Bypass PLL 00011: Bypass PLL 00100: 8 Times 00101: 10 Times 00110: 12 Times 00111: 14 Times . . . 11101: 58 Times 11110: 60 Times 11111: 62 Times Note: These options are not all guaranteed to work for the WM8505, and system designers should take into account the characterization of the PLL in determining the legal values.

Offset 0x0204 – 0x0207

PLL_B Multiplier and Range Values Register

This register allows the software to change the PLL_B Multiplier value.

IMPORTANT NOTE:

Writing this register will have no effect on the PLL_B when the Real Time Clock (RTC) oscillator is not running.

The value of this register after a power-up reset (*pmc_sus_reset_x*) will be all zeros.

The value of this register after a non power-up reset is not affected.

Bit	Attribute	Default	Description
31:20	RO	0	Reserved.
19:18	RW	0	PLL_B VCO Range Value Bits. These bits provide extra range control for the PLL_B's VCO. Unless otherwise instructed the software should always write zeros to these bits.
17:16	RW	0	PLL_B Charge-pump Range Value Bits. These bits provide extra range control for the PLL_B's charge pump. Unless otherwise instructed software should always write zeros to these bits.
15:9	RO	0	Reserved.
8	RW	0	PLL_B Pre-Divisor Bypass. This bit indicates the PLL_B Pre-Divisor Bypass status. 0: The Pre-Divisor is enabled and will divide the source clock to the PLL_B by two before it is used in PLL_B operations. 1: The Pre-Divisor is bypassed and the source clock to the PLL_B will be passed through for use in PLL_B operations.
7:5	RO	0	Reserved.
4:0	RW	0	PLL_B Multiplier Value Bits 4-0. These bits indicate the PLL Multiplier value which will be used to be multiplied by the input oscillator clock. The resulting PLL multiplier selected by these signals are as blow: 00000: Bypass PLL 00001: Bypass PLL 00010: Bypass PLL 00011: Bypass PLL 00100: 8 Times 00101: 10 Times 00110: 12 Times 00111: 14 Times . . . 11101: 58 Times 11110: 60 Times 11111: 62 Times Note: These options are not all guaranteed to work for the WM8505, and system designers should take into account the characterization of the PLL in determining the legal values.

Offset 0x0208 – 0x020B

PLL_C Multiplier and Range Values Register

This register allows the software to change the PLL_C Multiplier value.

IMPORTANT NOTE:

Writing this register will have no effect on the PLL_C when the Real Time Clock (RTC) oscillator is not running.

The value of this register after a power-up reset (*pmc_sus_reset_x*) will be all zeros.

The value of this register after a non power-up reset is not affected.

Bit	Attribute	Default	Description
31:20	RO	0	Reserved.
19:18	RW	0	PLL_C VCO Range Value Bits. These bits provide extra range control for the PLL_C's VCO. Unless otherwise instructed the software should always write zeros to these bits.
17:16	RW	0	PLL_C Charge-pump Range Value Bits. These bits provide extra range control for the PLL_C's charge pump. Unless otherwise instructed the software should always write zeros to these bits.
15:9	RO	0	Reserved.
8	RW	0	PLL_C Pre-Divisor Bypass. This bit indicates the PLL_C Pre-Divisor Bypass status. 0: The Pre-Divisor is enabled and will divide the source clock to the PLL_C by two before it is used in PLL_C operations. 1: The Pre-Divisor is bypassed and the source clock to the PLL_C will be passed through for use in PLL_C operations
7:5	RO	0	Reserved.
5	RW	0	PLL_C Clock Source 0: PLL_C reference clock from CLK27 MHz 1: PLL_C reference clock from CLK25 MHz
4:0	RW	0	PLL_C Multiplier Value Bits 4-0. These bits indicate the PLL Multiplier value which will be used to be multiplied by the input oscillator clock. The resulting PLL multiplier selected by these signals are as below: 00000: Bypass PLL 00001: Bypass PLL 00010: Bypass PLL 00011: Bypass PLL 00100: 8 Times 00101: 10 Times 00110: 12 Times 00111: 14 Times . . . 11101: 58 Times 11110: 60 Times 11111: 62 Times Note: These options are not all guaranteed to work for the WM8505, and system designers should take into account the characterization of the PLL in determining the legal value.

Offset 0x020C – 0x020E

PLL_D Multiplier and Range Values Register

This register allows the software to change the PLL_D Multiplier value.

IMPORTANT NOTE:

Writing this register will have no effect on the PLL_D when the Real Time Clock (RTC) oscillator is not running.

The value of this register after a power-up reset (*pmc_sus_reset_x*) will be all zeros.

The value of this register after a non power-up reset is not affected.

Bit	Attribute	Default	Description
31:20	RO	0	Reserved.
19:18	RW	0	PLL_D VCO Range Value Bits. These bits provide extra range control for the PLL_D’s VCO. Unless otherwise instructed the software should always write zeros to these bits.
17:16	RW	0	PLL_D Charge-pump Range Value Bits. These bits provide extra range control for the PLL_D’s charge pump. Unless otherwise instructed the software should always write zeros to these bits.
15:9	RO	0	Reserved.
8	RW	0	PLL_D Pre-Divisor Bypass. This bit indicates the PLL_D Pre-Divisor Bypass status. 0: The Pre-Divisor is enabled and will divide the source clock to the PLL_D by two before it is used in PLL_D operations. 1: The Pre-Divisor is bypassed and the source clock to the PLL_D will be passed through for use in PLL_D operations.
7:5	RO	0	Reserved.
4:0	RW	0	PLL_D Multiplier Value Bits 4-0. These bits indicate the PLL Multiplier value which will be used to multiply the input oscillator clock. The resulting PLL multiplier selected by these signals are as below: 00000: Bypass PLL 00001: Bypass PLL 00010: Bypass PLL 00011: Bypass PLL 00100: 8 Times 00101: 10 Times 00110: 12 Times 00111: 14 Times . . . 11101: 58 Times 11110: 60 Times 11111: 62 Times Note: These options are not all guaranteed to work for the WM8505, and system designers should take into account the characterization of the PLL in determining the legal values.

Offset 0x0210 – 0x0213

CLKGEN Control Register

This register allows the software to enable/disable the clocks generated by the CKLGEN model.

Bit	Attribute	Default	Description
31:21	RO	0	Reserved.
20:18	RW	0	CLKAUD_I, CLKAUD_O Frequency Select. 000: 16.3835 MHz 001: 24.5755 MHz 010: 22.5789 MHz 011: 33.8684 MHz 100: 36.8653 MHz

17:16	RW	0	CLKAUD_I, CLKAUD_O Frequency Divider. 0x: Divided by 1 10: Divided by 2 11: Divided by 4
15:13	RO	0	Reserved.
12	RW	0	CLKGEN.PLL0.Reset. 0: Normal operation 1: Assert CLKGEN. PLL0RST_
11	RW	0	CLKGEN.PLL1.Reset. 0: Normal operation 1: Assert CLKGEN. PLL1RST_
10	RW	0	CLKGEN.PLL2 Reset. 0: Normal operation 1: Assert CLKGEN. PLL2RST_
9	RW	0	CLKGEN.PLL3 Reset. 0: Normal operation 1: Assert CLKGEN. PLL3RST_
8	RW	0	Reserved.
7:4	RO	0	Reserved.
3	RO	0	PLL3 is Ready (from CLKGEN).
2	RO	0	PLL2 is Ready (from CLKGEN).
1	RO	0	PLL1 is Ready (from CLKGEN).
0	RO	0	PLL0 is Ready (from CLKGEN).

Offset 0x0250 – 0x0253

Clock Enables Lower Register

This register allows the software to enable/disable the clocks to various WM8505 peripherals. The value of this register after any reset will be all ones.

Bit	Attribute	Default	Description
31	RW	0	PLLC Program Alone 1: PLLC is allowed to be programmed independently with influencing the other PLLs.
30	RW	0	GOVRHD Clock Enable. 1: The clock(s) to the GE module will be enabled.
29	RW	0	GE Clock Enable. 1: The clock(s) to the GE module will be enabled.
28	RW	0	I^{2}S 6Pin Enable 1: The I^{2}S interface becomes 6 PIN mode. The RESETOUT ball will be used as the newly added I2SRWS function.
27	RW	0	JENC Clock Enable. 1: The clock(s) to the JENC module will be enabled.
26	RW	0	SPI0 Timeout Interrupt Disable
25	RW	0	SPI1 Timeout Interrupt Disable
24	RW	1b	AMP Clock Enable. 1: The clock(s) to the AMP module will be enabled.
23	RW	1b	UART 5 Clock Enable. 1: The clock(s) to the UART5 module will be enabled.
22	RW	0	UART 4 Clock Enable. 1: The clock(s) to the UART4 module will be enabled.
21	RW	1b	System Configuration Control (SCC) Clock Enable. 1: The clock(s) to the System Configuration Control module will be enabled.
20	RW	1b	Reserved

19	RW	0	AC97 Clock Enable. 1: The clock(s) to the AC97 module will be enabled.
18	RW	1b	DVO Clock Enable. 1: The clock(s) to the AHB1 module will be enabled.
17	RW	0	CIR Clock Enable. 1: The clock(s) to the AHB1 module will be enabled.
16	RW	0	I^{2}S Clock Enable. 1: The clock(s) to the I^{2}S module will be enabled.
15	RW	0	SPI2 Timeout Interrupt Disable.
14	RW	0	SPI2 Clock Enable. 1: The clock(s) to the SPI2 module will be enabled.
13	RW	0	SPI1 Clock Enable. 1: The clock(s) to the SPI1 module will be enabled.
12	RW	0	SPI0 Clock Enable. 1: The clock(s) to the SPI0 module will be enabled.
11	RW	1b	GPIO Clock Enable. 1: The AHB Clock to the GPIO module will be enabled.
10	RW	0	PWM Timers Clock Enable. 1: The clock(s) to the PWM module will be enabled.
9	RW	0	Keypad Clock Enable. 1: The clock(s) to the Keypad module will be enabled.
8	RW	0	Reserved
7	RW	1b	RTC Clock Enable. 1: The AHB Clock to the RTC module will be enabled. Note: This bit does not affect the RTC Clock.
6	RW	0	I^{2}C Slave Clock Enable. 1: The clock(s) to the I^{2}C Slave module will be enabled.
5	RW	0	I2C0 Clock Enable. 1: The clock(s) to the I2C0 module will be enabled.
4	RW	0	UART 3 Clock Enable. 1: The clock(s) to the UART3 module will be enabled.
3	RW	0	UART 2 Clock Enable. 1: When this bit is a one, it indicates that the clock(s) to the UART2 module will be enabled.
2	RW	0	UART1 Clock Enable. 1: The clock(s) to the UART1 module will be enabled.
1	RW	0	UART0 Clock Enable. 1: The clock(s) to the UART0 module will be enabled.
0	RW	0	I2C1 Clock Enable. 1: The clock(s) to the I2C1 module will be enabled.

Offset 0x0254 – 0x0257

Clock Enables Upper Register

This register allows the software to enable/disable the clocks to various WM8505 peripherals. The value of this register after any reset will be all ones.

Bit	Attribute	Default	Description
31	RW	0	VPP Clock Enable. 1: The AHB Clock to the VPP module will be enabled.
30	RW	0	VID Clock Enable. 1: The AHB Clock to the VID module will be enabled.

29	RW	0	GOVW Clock Enable. 1: The AHB Clock to the GOVW module will be enabled.
28	RW	0	SCL444U Clock Enable. 1: The AHB Clock to the scl444u module will be enabled.
27	RW	1b	Reserved
26	RW	0	Ethernet PHY Clock Enable. 1: The Peripheral Clock to the Ethernet PHY will be enabled.
25	RW	0	Reserved
24	RW	0	SAE Clock Enable. 1: The AHB Clock to the SAE module will be enabled.
23	RW	1b	Serial Flash Memory Controller Clock Enable. 1: The AHB and Peripheral Clocks to the Serial Flash Memory Controller module will be enabled.
22	RW	0	Reserved
21	RW	0	SYS Clock Enable. 1: The AHB and Peripheral Clocks to the SYS module will be enabled
20	RW	0	Ethernet MAC0 Clock Enable. 1: The AHB and Peripheral Clocks to the MAC0 module will be enabled.
19	RW	0	Reserved
18	RW	0	SD/MMC Clock Enable. 1: The AHB and Peripheral Clocks to the SD/MMC module will be enabled.
17	RW	0	Reserved
16	RW	1b	NAND Clock Enable. 1: The AHB and Peripheral Clocks to the NAND module will be enabled.
15	RW	1b	Reserved
14	RW	1b	SDTV Clock Enable. 1: The SDTV module will be enabled.
13	RW	1b	AHB Bridge AHB Clock Enable. 1: The AHB Clock to the AHB Bridge module will be enabled.
12	RW	1b	Reserved
11:10	RW	0	Reserved
9	RW	0	PDMA Clock Enable. 1: The AHB Clock to the PDMA module will be enabled.
8	RW	0	UDC Clock Enable. 1: The AHB and Peripheral Clocks to the UDC module will be enabled.
7	RW	1b	UHC Clock Enable. 1: The AHB and Peripheral Clocks to the UHC module will be enabled.
6	RW	0	Reserved
5	RW	0	DMA Clock Enable. 1: The DMA module will be enabled.
4	RW	0	PS2 Clock Enable. 1: The KBDC module will be enabled.
3	RW	1b	NOR/GUP Clock Enable. 1: The AHB and Peripheral Clocks to the NOR/GUP module will be enabled.
2	RW	0	NA12 Clock Enable. 1: The NA12 module will be enabled.
1	RW	0	NA0 Clock Enable. 1: The NA0 module will be enabled.

0	RW	1b	DDR Memory Controller Clock Enable. 1: The AHB and Peripheral Clocks to the DDR Memory Controller module will be enabled.

Offset 0x0300

ARM Clock Divisor Value Register

This register allows the software to change the ARM clock divisor value.

Write this register as the ARM Clock's High Pulse the Wide Pulse register is written by using a 2-byte write or 4-byte write.

The value of this register after every reset is all zeros.

Bit	Attribute	Default	Description
7:5	RO	0	Reserved
4:0	RW	0	ARM Clock Divisor Value Bits 4-0. These bits indicate the ARM clock divisor value which will divide the PLL output clock by the value programmed in these bits to generate the desired ARM output clock. When this register is written, the PMC module will work with the Clock Control and PLL (CCP) module to change the ARM clock to the appropriate frequency. 00000: Divided by 32. 00001: Divided by 1. 00010: Divided by 2. 00011: Divided by 3. 00100: Divided by 4. 00101: Divided by 5. . . 11100: Divided by 28. 11101: Divided by 29. 11110: Divided by 30. 11111: Divided by 31.

Offset 0x0301

ARM Clock High Pulse the Wide Pulse Register

This register allows the software to change the ARM clock high pulse or low pulse to be wider, i.e. if the ARM Clock's Divisor Value is an odd value other than one.

Write this register as the ARM Clock's Divisor Value register is written by using a 2-byte write or 4-byte write.

The value of this register after every reset is all zeros.

Bit	Attribute	Default	Description
7:1	RO	0	Reserved
0	RW	0	ARM Clock High Pulse is the Wide Pulse. When the ARM Clock Divisor value is an odd value greater than 5'h01: 0: The low pulse is the longer pulse. 1: The high pulse is the longer pulse. When the ARM Clock Divisor value is an even value, the low and high pulses will be of equal length. When the ARM Clock Divisor value is 5'h01, the clock will be the PLL output clock and will have the same duty cycle as that produced by the PLL.

Offset 0x0304

AHB Clock Divisor Value Register

This register allows the software to change the AHB clock divisor value.

The value of this register after every reset is 8’h01.

Bit	Attribute	Default	Description
7:3	RO	0	Reserved
2:0	RW	000b	AHB Clock Divisor Value. These bits indicate the AHB clock divisor value which will divide the ARM output clock by the value programmed in these bits to generate the desired AHB output clock. When this register is written, the PMC module will work with the Clock Control and PLL (CCP) module to change the AHB clock to the appropriate frequency. 000: Divided by 8. 001: Divided by 1. 010: Divided by 2. 011: Divided by 3. 100: Divided by 4. 101: Divided by 5. 110: Divided by 6. 111: Divided by 7.

Offset 0x0310

DDR Memory Controller Clock Divisor Value Register

This register allows the software to change the DDR memory controller clock divisor value.

The value of this register after every reset is 8’h00.

Bit	Attribute	Default	Description
7:5	RO	0	Reserved
4:0	RW	00000b	DDR Memory Controller Clock Divisor Value Bits 4-0. These bits indicate the DDR memory controller clock divisor value which will divide the PLL output clock by the value programmed in these bits to generate the desired DDR memory controller output clock. When this register is written, the PMC module will work with the Clock Control and PLL (CCP) module to change the DDR memory controller clock to the appropriate frequency. 00000: Divided by 32. 00001: Divided by 1. 00010: Divided by 2. 00011: Reserved 00100: Divided by 4. 00101: Reserved . . 11100: Divided by 28. 11101: Reserved 11110: Divided by 30. 11111: Reserved Note: Odd divisor values greater than 5’h01 are Reserved and should not be used.

Offset 0x0314

Serial Flash Memory Controller Clock Divisor Value Register

This register allows the software to change the Serial Flash Memory Controller clock divisor value.

The value of this register after every reset is 8'h00.

Bit	Attribute	Default	Description
7:5	RO	0	Reserved
4:0	RW	00000b	Serial Flash Memory Controller Clock Divisor Value Bits 4-0. These bits indicate the Serial Flash Memory Controller clock divisor value which will divide the PLL output clock by the value programmed in these bits to generate the desired Serial Flash Memory Controller output clock. When this register is written, the PMC module will work with the Clock Control and PLL (CCP) module to change the Serial Flash Memory Controller clock to the appropriate frequency. 00000: Divided by 32. 00001: Divided by 1. 00010: Divided by 2. 00011: Divided by 3. 00100: Divided by 4. 00101: Divided by 5. . . 11100: Divided by 28. 11101: Divided by 29. 11110: Divided by 30. 11111: Divided by 31.

Offset 0x0315

Serial Flash Memory Controller Clock High Pulse the Wide Pulse Register

This register allows the software to change the Serial Flash Memory Controller clock high pulse or low pulse to be wider, i.e. if the Serial Flash Memory Controller Clock's Divisor Value is an odd value other than one.

Write this register as the Serial Flash Memory Controller Clock's Divisor Value register is written by using a 2-byte write or 4-byte write.

The value of this register after every reset is all zeros.

Bit	Attribute	Default	Description
7:1	RO	0	Reserved
0	RW	0	Serial Flash Memory Controller Clock High Pulse the Wide Pulse. When the Serial Flash Memory Controller Clock Divisor value is an odd value greater than 5'h01: 0: The low pulse is the longer pulse. 1: The high pulse is the longer pulse. When the Serial Flash Memory Controller Clock Divisor value is an even value, the low and high pulses will be of equal length. When the Serial Flash Memory Controller Clock Divisor value is 5'h01, the clock will be the PLL output clock and will have the same duty cycle as that produced by the PLL.

Offset 0x0318

Keyboard Controller Pre Clock Divisor Value Register

This register allows the software to change the Keyboard Controller Pre clock divisor value.

The value of this register after every reset is 8’h00.

Bit	Attribute	Default	Description
7:5	RO	0	Reserved
4:0	RW	00000b	Keyboard Controller Pre Clock Divisor Value Bits 4-0. These bits indicate the Keyboard Controller Pre clock divisor value which will divide the PLL output clock by the value programmed in these bits to generate the desired Keyboard Controller Pre output clock. When this register is written, the PMC module will work with the Clock Control and PLL (CCP) module to change the Keyboard Controller clock to the appropriate frequency. 00000: Divided by 32. 00001: Divided by 1. 00010: Divided by 2. 00011: Divided by 3. 00100: Divided by 4. 00101: Divided by 5. . . 11100: Divided by 28. 11101: Divided by 29. 11110: Divided by 30. 11111: Divided by 31.

Offset 0x0319

Keyboard Controller Pre Clock High Pulse the Wide Pulse Register

The purpose of this register allows the software to change the Keyboard Controller Pre clock high pulse or low pulse to be wider, i.e. if the Keyboard Controller Pre Clock’s Divisor Value is an odd value other than one.

Write this register as the Keyboard Controller Pre Clock’s Divisor Value register is written by using a 2-byte write or 4-byte write.

The value of this register after every reset is all zeros.

Bit	Attribute	Default	Description
7:1	RO	0	Reserved
0	RW	0	Keyboard Pre Clock High Pulse the Wide Pulse. When the Keyboard Controller Pre Clock Divisor value is an odd value greater than 5’h01: 0: The low pulse is the longer pulse. 1: The high pulse is the longer pulse. When the Keyboard Controller Pre Clock Divisor value is an even value, the low and high pulses will be of equal length. When the Keyboard Controller Pre Clock Divisor value is 5’h01, the clock will be the PLL output clock and will have the same duty cycle as that produced by the PLL.

Offset 0x031C

Keyboard Controller Clock Divisor Value Register

This register allows the software to change the Keyboard Controller clock divisor value.

The value of this register after every reset is 8’h00.

Bit	Attribute	Default	Description
7:5	RO	0	Reserved
4:0	RW	00000b	Keyboard Controller Clock Divisor Value Bits 4-0. These bits indicate the Keyboard Controller clock divisor value which will divide the PLL output clock by the value programmed in these bits to generate the desired Keyboard Controller output clock. When this register is written, the PMC module will work with the Clock Control and PLL (CCP) module to change the Keyboard Controller clock to the appropriate frequency. 00000: Divided by 32. 00001: Divided by 1. 00010: Divided by 2. 00011: Divided by 3. 00100: Divided by 4. 00101: Divided by 5. . . 11100: Divided by 28. 11101: Divided by 29. 11110: Divided by 30. 11111: Divided by 31.

Offset 0x031D

Keyboard Controller Clock High Pulse the Wide Pulse Register

This register allows the software to change the Keyboard Controller clock high pulse or low pulse to be wider, i.e. if the Keyboard Controller Clock’s Divisor Value is an odd value other than one.

Write this register as the Keyboard Controller Clock’s Divisor Value register is written by using a 2-byte write or 4-byte write.

The value of this register after every reset is all zeros.

Bit	Attribute	Default	Description
7:1	RO	0	Reserved
0	RW	0	Keyboard Clock High Pulse the Wide Pulse. When the Keyboard Controller Clock Divisor value is an odd value greater than 5’h01: 0: The low pulse is the longer pulse. 1: The high pulse is the longer pulse (when this bit is a one). When the Keyboard Controller Clock Divisor value is an even value, the low and high pulses will be of equal length. When the Keyboard Controller Clock Divisor value is 5’h01, the clock will be the PLL output clock and will have the same duty cycle as that produced by the PLL.

Offset 0x0328

SD / MMC Clock Divisor Value Register

This register allows the software to change the SD/MMC clock divisor value.

The value of this register after every reset is all zeros.

Bit	Attribute	Default	Description
7:6	RO	0	Reserved
5	RW	0	SD/MMC Fixed Divisor Enable. 0: The additional Fixed Divisor stage of the SD/MMC clock divisor will be disabled. 1: The additional Fixed Divisor stage of the SD/MMC clock divisor will be enabled. Note: This Fixed Divisor stage provides a fixed division of the clock by a factor of 64.
4:0	RW	00000b	SD/MMC Clock Divisor Value Bits 4-0. These bits indicate the SD/MMC clock divisor value which will divide the PLL output clock by the value programmed in these bits to generate the desired SD/MMC output clock. When this register is written, the PMC module will work with the Clock Control and PLL (CCP) module to change the SD/MMC clock to the appropriate frequency. 00000: Divided by 32. 00001: Divided by 1. 00010: Divided by 2. 00011: Divided by 3. 00100: Divided by 4. 00101: Divided by 5. . . 11100: Divided by 28. 11101: Divided by 29. 11110: Divided by 30. 11111: Divided by 31.

Offset 0x0329

SD / MMC Clock High Pulse the Wide Pulse Register

This register allows the software to change the SD/MMC clock high pulse or low pulse to be wider, i.e. if the SD/MMC Clock Divisor Value is an odd value other than one.

Write this register as the SD/MMC Clock Divisor Value register is written by using a 2-byte write or 4-byte write.

The value of this register after every reset is all zeros.

Bit	Attribute	Default	Description
7:1	RO	0	Reserved
0	RW	0	SD/MMC Clock High Pulse the Wide Pulse. When the SD/MMC Clock Divisor value is an odd value greater than 5'h01: 0: The low pulse is the longer pulse. 1: The high pulse is the longer pulse. When the SD/MMC Clock Divisor value is an even value, the low and high pulses will be of equal length. When the SD/MMC Clock Divisor value is 5'h01, clock will be the PLL output clock and will have the same duty cycle as that produced by the PLL.

Offset 0x032C

GENET Clock Divisor Value Register

This register allows the software to change the GENET clock divisor value.

The value of this register after every reset is all zeros.

Bit	Attribute	Default	Description
7:5	RO	0	Reserved
4:0	RW	00000b	GENET Clock Divisor Value Bits 4-0. These bits indicate the GENET clock divisor value which will divide the PLL output clock by the value programmed in these bits to generate the desired GENET output clock. When this register is written, the PMC module will work with the Clock Control and PLL (CCP) module to change the GENET clock to the appropriate frequency. 00000: Divided by 32. 00001: Divided by 1. 00010: Divided by 2. 00011: Divided by 3. 00100: Divided by 4. 00101: Divided by 5. . . 11100: Divided by 28. 11101: Divided by 29. 11110: Divided by 30. 11111: Divided by 31.

Offset 0x032D

GENET Clock High Pulse the Wide Pulse Register

This register allows the software to change the GENET clock high pulse or low pulse to be wider, i.e. if the GENET Clock Divisor Value is an odd value other than one.

Write this register as the GENET Clock Divisor Value register is written by using a 2-byte write or 4-byte write.

The value of this register after every reset is all zeros.

Bit	Attribute	Default	Description
7:1	RO	0	Reserved
0	RW	0	GENET Clock High Pulse the Wide Pulse. When the GENET Clock Divisor value is an odd value greater than 5'h01: 0: The low pulse is the longer pulse. 1: The high pulse is the longer pulse. When the GENET Clock Divisor value is an even value, the low and high pulses will be of equal length. When the GENET Clock Divisor value is 5'h01, clock will be the PLL output clock and will have the same duty cycle as that produced by the PLL.

Offset 0x0330

NAND Clock Divisor Value Register

This register allows the software to change the NAND clock divisor value.

The value of this register after every reset is all zeros.

Bit	Attribute	Default	Description
7:5	RO	0	Reserved
4:0	RW	00000b	NAND Clock Divisor Value Bits 4-0. These bits indicate the NAND clock divisor value which will divide the PLL output clock by the value programmed in these bits to generate the desired NAND output clock. When this register is written, the PMC module will work with the Clock Control and PLL (CCP) module to change the NAND clock to the appropriate frequency. 00000: Divided by 32. 00001: Divided by 1. 00010: Divided by 2. 00011: Divided by 3. 00100: Divided by 4. 00101: Divided by 5. . . 11100: Divided by 28. 11101: Divided by 29. 11110: Divided by 30. 11111: Divided by 31.

Offset 0x0331

NAND Clock High Pulse the Wide Pulse Register

This register allows the software to change the NAND clock high pulse or low pulse to be wider, i.e. if the NAND Clock Divisor Value is an odd value other than one.

Write this register at the same time as the NAND Clock Divisor Value register is written by using a 2-byte write or 4-byte write.

The value of this register after every reset is all zeros.

Bit	Attribute	Default	Description
7:1	RO	0	Reserved
0	RW	0	NAND Clock High Pulse the Wide Pulse. When the NAND Clock Divisor Value is an odd value greater than 5’h01: 0: The low pulse is the longer pulse. 1: The high pulse is the longer pulse. When the NAND Clock Divisor Value is an even value, the low and high pulses will be of equal length. When the NAND Clock Divisor Value is 5’h01, clock will be the PLL output clock and will have the same duty cycle as that produced by the PLL.

Offset 0x0334

NOR Clock Divisor Value Register

This register allows the software to change the NOR clock divisor value.

The value of this register after every reset is all zeros.

Bit	Attribute	Default	Description
7:5	RO	0	Reserved
4:0	RW	00000b	NOR Clock Divisor Value Bits 4-0. These bits indicate the NOR clock divisor value which will divide the PLL output clock by the value programmed in these bits to generate the desired NOR output clock. When this register is written, the PMC module will work with the Clock Control and PLL (CCP) module to change the NOR clock to the appropriate frequency. 00000: Divided by 32. 00001: Divided by 1. 00010: Divided by 2. 00011: Divided by 3. 00100: Divided by 4. 00101: Divided by 5. . . 11100: Divided by 28. 11101: Divided by 29. 11110: Divided by 30. 11111: Divided by 31.

Offset 0x0335

NOR Clock High Pulse the Wide Pulse Register

This register allows the software to change the NOR clock high pulse or low pulse to be wider, i.e. if the NOR Clock Divisor Value is an odd value other than one.

Write this register as the NOR Clock Divisor Value register is written by using a 2-byte write or 4-byte write.

The value of this register after every reset is all zeros.

Bit	Attribute	Default	Description
7:1	RO	0	Reserved
0	RW	0	NOR Clock High Pulse the Wide Pulse. When the NOR Clock Divisor Value is an odd value greater than 5’h01: 0: The low pulse is the longer pulse. 1: The high pulse is the longer pulse. When the NOR Clock Divisor Value is an even value, the low and high pulses will be of equal length. When the NOR Clock Divisor Value is 5’h01, clock will be the PLL output clock and will have the same duty cycle as that produced by the PLL.

Offset 0x033C

SPI0 Clock Divisor Value Register

This register allows the software to change the SPI0 clock divisor value.

The value of this register after every reset is all zeros.

Bit	Attribute	Default	Description
7:5	RO	0	Reserved
4:0	RW	00000b	SPI0 Clock Divisor Value Bits 4-0. These bits indicate the SPI0 clock divisor value which will divide the PLL output clock by the value programmed in these bits to generate the desired SPI0 output clock. When this register is written, the PMC module will work with the Clock Control and PLL (CCP) module to change the SPI0 clock to the appropriate frequency. 00000: Divided by 32. 00001: Divided by 1. 00010: Divided by 2. 00011: Divided by 3. 00100: Divided by 4. 00101: Divided by 5. . . 11100: Divided by 28. 11101: Divided by 29. 11110: Divided by 30. 11111: Divided by 31.

Offset 0x033D

SPI0 Clock High Pulse the Wide Pulse Register

This register allows the software to change the SPI0 clock high pulse or low pulse to be wider, i.e. if the SPI0 Clock Divisor Value is an odd value other than one.

Write this register as the SPI0 Clock Divisor Value register is written by using a 2-byte write or 4-byte write.

The value of this register after every reset is all zeros.

Bit	Attribute	Default	Description
7:1	RO	0	Reserved
0	RW	0	SPI0 Clock High Pulse the Wide Pulse. When the SPI0 Clock Divisor Value is an odd value greater than 5'h01: 0: The low pulse is the longer pulse. 1: The high pulse is the longer pulse. When the SPI0 Clock Divisor Value is an even value, the low and high pulses will be of equal length. When the SPI0 Clock Divisor Value is 5'h01, clock will be the PLL output clock and will have the same duty cycle as that produced by the PLL.

Offset 0x0340

SPI1 Clock Divisor Value Register

This register allows the software to change the SPI1 clock divisor value.

The value of this register after every reset is all zeros.

Bit	Attribute	Default	Description
7:5	RO	0	Reserved
4:0	RW	00000b	SPI1 Clock Divisor Value Bits 4-0. These bits indicate the SPI1 clock divisor value which will divide the PLL output clock by the value programmed in these bits to generate the desired SPI1 output clock. When this register is written, the PMC module will work with the Clock Control and PLL (CCP) module to change the SPI1 clock to the appropriate frequency. 00000: Divided by 32. 00001: Divided by 1. 00010: Divided by 2. 00011: Divided by 3. 00100: Divided by 4. 00101: Divided by 5. . . 11100: Divided by 28. 11101: Divided by 29. 11110: Divided by 30. 11111: Divided by 31.

Offset 0x0341

SPI1 Clock High Pulse is the Wide Pulse Register

This register allows the software to change the SPI1 clock high pulse or low pulse to be wider, i.e. if the SPI1 Clock Divisor Value is an odd value other than one.

Write this register as the SPI1 Clock Divisor Value register is written by using a 2-byte write or 4-byte write.

The value of this register after every reset is all zeros.

Bit	Attribute	Default	Description
7:1	RO	0	Reserved
0	RW	0	SPI1 Clock High Pulse the Wide Pulse. When the SPI1 Clock Divisor Value is an odd value greater than 5'h01: 0: The low pulse is the longer pulse. 1: The high pulse is the longer pulse. When the SPI1 Clock Divisor Value is an even value, the low and high pulses will be of equal length. When the SPI1 Clock Divisor Value is 5'h01, clock will be the PLL output clock and will have the same duty cycle as that produced by the PLL.

Offset 0x0344

SPI2 Clock Divisor Value Register

This register allows the software to change the SPI2 clock divisor value.

The value of this register after every reset is all zeros.

Bit	Attribute	Default	Description
7:5	RO	0	Reserved
4:0	RW	00000b	SPI2 Clock Divisor Value Bits 4-0. These bits indicate the SPI2 clock divisor value which will divide the PLL output clock by the value programmed in these bits to generate the desired SPI2 output clock. When this register is written, the PMC module will work with the Clock Control and PLL (CCP) module to change the SPI2 clock to the appropriate frequency. 00000: Divided by 32. 00001: Divided by 1. 00010: Divided by 2. 00011: Divided by 3. 00100: Divided by 4. 00101: Divided by 5. . . 11100: Divided by 28. 11101: Divided by 29. 11110: Divided by 30. 11111: Divided by 31.

Offset 0x0345

SPI2 Clock High Pulse the Wide Pulse Register

This register allows the software to change the SPI2 clock high pulse or low pulse to be wider, i.e. if the SPI2 Clock Divisor Value is an odd value other than one.

Write this register as the SPI2 Clock Divisor Value register is written by using a 2-byte write or 4-byte write.

The value of this register after every reset is all zeros.

Bit	Attribute	Default	Description
7:1	RO	0	Reserved
0	RW	0	SPI2 Clock High Pulse the Wide Pulse. When the SPI2 Clock Divisor Value is an odd value greater than 5’h01: 0: The low pulse is the longer pulse. 1: The high pulse is the longer pulse. When the SPI2 Clock Divisor Value is an even value, the low and high pulses will be of equal length. When the SPI2 Clock Divisor Value is 5’h01, clock will be the PLL output clock and will have the same duty cycle as that produced by the PLL.

Offset 0x0348

PWM Clock Divisor Value Register

This register allows the software to change the PWM clock divisor value.

The value of this register after every reset is all zeros.

Bit	Attribute	Default	Description
7:5	RO	0	Reserved
4:0	RW	00000b	PWM Clock Divisor Value Bits 4-0. These bits indicate the PWM clock divisor value which will divide the PLL output clock by the value programmed in these bits to generate the desired PWM output clock. When this register is written, the PMC module will work with the Clock Control and PLL (CCP) module to change the PWM clock to the appropriate frequency. 00000: Divided by 32. 00001: Divided by 1. 00010: Divided by 2. 00011: Divided by 3. 00100: Divided by 4. 00101: Divided by 5. . . 11100: Divided by 28. 11101: Divided by 29. 11110: Divided by 30. 11111: Divided by 31.

Offset 0x0349

PWM Clock High Pulse the Wide Pulse Register

This register allows the software to change the PWM clock high pulse or low pulse to be wider, i.e. if the PWM Clock Divisor Value is an odd value other than one.

Write this register as the PWM Clock Divisor Value register is written by using a 2-byte write or 4-byte write.

The value of this register after every reset is all zeros.

Bit	Attribute	Default	Description
7:1	RO	0	Reserved
0	RW	0	PWM Clock High Pulse the Wide Pulse. When the PWM Clock Divisor Value is an odd value greater than 5'h01: 0: The low pulse is the longer pulse. 1: The high pulse is the longer pulse. When the PWM Clock Divisor Value is an even value, the low and high pulses will be of equal length. When the PWM Clock Divisor Value is 5'h01, clock will be the PLL output clock and will have the same duty cycle as that produced by the PLL.

Offset 0x0350

APB Clock Divisor Value Register

This register allows the software to change the APB clock divisor value.

The value of this register after every reset is all zeros.

Bit	Attribute	Default	Description
7:5	RO	0	Reserved
4:0	RW	00000b	APB Clock Divisor Value Bits 4-0. These bits indicate the APB clock divisor value which will divide the PLL output clock by the value programmed in these bits to generate the desired APB output clock. When this register is written, the PMC module will work with the Clock Control and PLL (CCP) module to change the APB clock to the appropriate frequency. 00000: Divided by 32. 00001: Divided by 1. 00010: Divided by 2. 00011: Divided by 3. 00100: Divided by 4. 00101: Divided by 5. . . 11100: Divided by 28. 11101: Divided by 29. 11110: Divided by 30. 11111: Divided by 31.

Offset 0x0358

NA0 Clock Divisor Value Register

This register allows the software to change the NA0 clock divisor value.

The value of this register after every reset is all zeros.

Bit	Attribute	Default	Description
7:5	RO	0	Reserved
4:0	RW	00000b	NA0 Clock Divisor Value Bits 4-0. These bits indicate the NA0 clock divisor value which will divide the PLL output clock by the value programmed in these bits to generate the desired NA0 output clock. When this register is written, the PMC module will work with the Clock Control and PLL (CCP) module to change the NA0 clock to the appropriate frequency. 00000: Divided by 32. 00001: Divided by 1. 00010: Divided by 2. 00011: Divided by 3. 00100: Divided by 4. 00101: Divided by 5. . . 11100: Divided by 28. 11101: Divided by 29. 11110: Divided by 30. 11111: Divided by 31.

Offset 0x035C

NA12 Clock Divisor Value Register

This register allows the software to change the NA12 clock divisor value.
The value of this register after every reset is all zeros.

Bit	Attribute	Default	Description
7:5	RO	0	Reserved
4:0	RW	00000b	NA12 Clock Divisor Value Bits 4-0. These bits indicate the NA12 clock divisor value which will divide the PLL output clock by the value programmed in these bits to generate the desired NA12 output clock. When this register is written, the PMC module will work with the Clock Control and PLL (CCP) module to change the NA12 clock to the appropriate frequency. 00000: Divided by 32. 00001: Divided by 1. 00010: Divided by 2. 00011: Divided by 3. 00100: Divided by 4. 00101: Divided by 5. . . 11100: Divided by 28. 11101: Divided by 29. 11110: Divided by 30. 11111: Divided by 31.

Offset 0x036C

I2C0 Clock Divisor Value Register

This register allows the software to change the I2C0 clock divisor value.
The value of this register after every reset is all zeros.

Bit	Attribute	Default	Description
7:5	RO	0	Reserved
4:0	RW	00000b	I2C0 Clock Divisor Value Bits 4-0. These bits indicate the I2C0 clock divisor value which will divide the PLL output clock by the value programmed in these bits to generate the desired I2C0 output clock. When this register is written, the PMC module will work with the Clock Control and PLL (CCP) module to change the I2C0 clock to the appropriate frequency. 00000: Divided by 32. 00001: Divided by 1. 00010: Divided by 2. 00011: Divided by 3. 00100: Divided by 4. 00101: Divided by 5. . . 11100: Divided by 28. 11101: Divided by 29. 11110: Divided by 30. 11111: Divided by 31.

Offset 0x0370

I2C1 Clock Divisor Value Register

This register allows the software to change the I2C1 clock divisor value.
The value of this register after every reset is all zeros.

Bit	Attribute	Default	Description
7:5	RO	0	Reserved
4:0	RW	00000b	I2C1 Clock Divisor Value Bits 4-0. These bits indicate the I2C1 clock divisor value which will divide the PLL output clock by the value programmed in these bits to generate the desired I2C1 output clock. When this register is written, the PMC module will work with the Clock Control and PLL (CCP) module to change the I2C1 clock to the appropriate frequency. 00000: Divided by 32. 00001: Divided by 1. 00010: Divided by 2. 00011: Divided by 3. 00100: Divided by 4. 00101: Divided by 5. . . 11100: Divided by 28. 11101: Divided by 29. 11110: Divided by 30. 11111: Divided by 31.

Offset 0x0374

DVO Clock Divisor Value Register

This register allows the software to change the DVO clock divisor value.
The value of this register after every reset is all zeros.

Bit	Attribute	Default	Description
7:5	RO	0	Reserved
4:0	RW	00000b	DVO Clock Divisor Value Bits 4-0. These bits indicate the DVO clock divisor value which will divide the PLL output clock by the value programmed in these bits to generate the desired DVO output clock. When this register is written, the PMC module will work with the Clock Control and PLL (CCP) module to change the DVO clock to the appropriate frequency. 00000: Divided by 32. 00001: Divided by 1. 00010: Divided by 2. 00011: Divided by 3. 00100: Divided by 4. 00101: Divided by 5. . . 11100: Divided by 28. 11101: Divided by 29. 11110: Divided by 30. 11111: Divided by 31.

Offset 0x0378

RingOsc Clock Divisor-1 Value Register

This register allows the software to change the RingOscillator clock 2nd stage divisor value.
The value of this register after every reset is all zeros.

Bit	Attribute	Default	Description
7:5	RO	0	Reserved
4:0	RW	00000b	RingOsc Clock Divisor-1 Value Bits 4-0. Bit[4:0] is used to set the "1st stage" divisor values to generate the reference clock for the "2nd stage" divider. Bit[4:0] indicates the RingOsc divisor value which will divide the RingOSC output clock by the value programmed in these bits. When this register is written, the PMC module will work with the Clock Control to change the reference clock of the 1st stage clock divider to the appropriate frequency. 00000: Divided by 32. 00001: Divided by 1. 00010: Divided by 2. 00011: Divided by 3. 00100: Divided by 4. 00101: Divided by 5. . . 11100: Divided by 28. 11101: Divided by 29. 11110: Divided by 30. 11111: Divided by 31.

Offset 0x037C

RingOsc Clock Divisor-2 Value Register

Bit[4:0] indicates the RingOsc divisor value used by the 2nd divider to generate the desired pseudo_rtc_clock.
The value of this register after every reset is all zeros.

Bit	Attribute	Default	Description
7:5	RO	0	Reserved
4:0	RW	00000b	RingOsc Clock Divisor-2 Value bits 4-0. Bit[4:0] indicates the RingOsc divisor value that divides the RingOSC output clock by the programmed value to generate the desired pseudo_rtc_clock. When this register is written, the PMC module will work with the Clock Control to change the pseudo_rtc clock to the appropriate frequency. 00000: Divided by 32. 00001: Divided by 1. 00010: Divided by 2. 00011: Divided by 3. 00100: Divided by 4. 00101: Divided by 5. . . 11100: Divided by 28. 11101: Divided by 29. 11110: Divided by 30. 11111: Divided by 31.

Interrupt Controller Registers

The Interrupt Controller Standard (ic_std) module contains 8 interrupt highest-priority registers, 8 interrupt priority control registers, 64 interrupt destination control registers and 16 interrupt status registers.

Base address of Interrupt Controller 0: 0xD814:0000
Base address of Interrupt Controller 1: 0xD815:0000

Interrupt Highest-priority Status Register

Offset 0x0000 / 0x0004 / 0x0008 / 0x000C / 0x0010 / 0x0014 / 0x0018 / 0x001C

IRQ0 / IRQ1 / IRQ2 / IRQ3 / IRQ4 / IRQ5 / IRQ6 / IRQ7 Interrupt Highest-priority Status Register

Each decimal value of the interrupt highest-priority registers indicates the one of the 64 interrupt sources that is currently active and is the request source with higher/highest priority over other currently active ones.

Bit	Attribute	Default	Description
31:6	RO	0	Reserved
5:0	RO	3Fh	Current Active Interrupt Input. Convert to decimal to indicates the current active IRQ number 0 ~ 63.

Note:
The default value for the Interrupt Highest Priority Register will cause it to indicate that interrupt source 63 is active, even if it is not. It is recommended that the software further check the Interrupt Status Register if interrupt source 63 is indicated.

Interrupt Priority Control Register

Offset 0x0020 / 0x0024 / 0x0028 / 0x002C / 0x0030 / 0x0034 / 0x0038 / 0x003C

IRQ0 / IRQ1 / IRQ2 / IRQ3 / IRQ4 / IRQ5 / IRQ6 / IRQ7 Interrupt Priority Control Register

The priority control registers indicate if the corresponding interrupt decoder is in rotating priority scheme (rp_en = 1'b1).

It can also be programmed to designate the highest priority to any one of the 64 interrupt inputs if it is not in rotating priority scheme (rp_en=1'b0).

Every interrupt output can have its own priority scheme and designated highest priority controlled by bit[6](rp_en) and bit[5:0] of this register, respectively.

Bit	Attribute	Default	Description
31:6	RW	0	Reserved
6	RW	0	Rotating Priority Enable Bit. When this bit is one, the corresponding Interrupt Decoder Block is in rotating priority mode.
5:0	RW	0	Fixed Priority Register. It contains the designated highest priority interrupt value in hexadecimal.

Note:
When rp_en is set to "1", the rotating priority scheme and the value stored/programmed in bit[5:0] will be ignored.

Destination Control Register

Offset 0x0040 / 0x0041 / 0x0042 / … / 0x007D / 0x007E / 0x007F

INT0 / INT1 / INT2 / … / INT61 / INT62 / INT63 Destination Control Register

The sixty-four interrupt destination control registers control the destinations and trigger mode of each interrupt source.

Each interrupt source can only have a single destination assigned to one of the eight possible interrupt decoders which subsequently generates the corresponding eight interrupt outputs.

Bit	Attribute	Default	Description
7:6	RW	0	Reserved
5:4	RW	00b	Interrupt Request Trigger Mode. 00: High level trigger 01: Posedge trigger 10: Negedge trigger 11: Both posedge and negedge trigger
3	RW	0	Interrupt Enable Bit. 1: The corresponding interrupt is enabled.
2:0	RW	000b	Route this Interrupt Input to Interrupt Decoder Block ic_irq_* 000: Route this interrupt input to Interrupt Decoder Block 0 (ic_irq[0]). 001: Route this interrupt input to Interrupt Decoder Block 1 (ic_irq[1]). 010: Route this interrupt input to Interrupt Decoder Block 2 (ic_irq[2]). 011: Route this interrupt input to Interrupt Decoder Block 3 (ic_irq[3]). 100: Route this interrupt input to Interrupt Decoder Block 4 (ic_irq[4]). 101: Route this interrupt input to Interrupt Decoder Block 5 (ic_irq[5]). 110: Route this interrupt input to Interrupt Decoder Block 6 (ic_irq[6]). 111: Route this interrupt input to Interrupt Decoder Block 7 (ic_irq[7]).

Note:
Using 16-bit/32-bit accesses to program these destination registers will simultaneously write 2/4 registers, so make sure it is the intention to program multiple registers at a time. Otherwise, 8-bit accesses must be used.

Interrupt Status Register

Offset 0x0080 / 0x0088 / 0x0090 / 0x0098 / 0x00A0 / 0x00A8 / 0x00B0 / 0x00B8

IRQ0 / IRQ1 / IRQ2 / IRQ3 / IRQ4 / IRQ5 / IRQ6 / IRQ7 Interrupt Status of INT31 to INT0 Register

Each interrupt output will have a pair of 32-bit registers that show all the 64 interrupt input status (active or inactive), excluding those not enabled.

This will give the programmer more flexibility/ability to control the priority.

Bit	Attribute	Default	Description
31:0	RW0S	0	Indicate Interrupt Input Status: INT31 to INT0

Note:
Interrupt Status Register is written one to clear. If interrupt trigger mode is edge mode (both posedge and negedge), its status can be cleared only by writing one. If the interrupt trigger mode is high level, it can be cleared only by disabling the interrupt input.

Offset 0x0084 / 0x008C / 0x0094 / 0x009C / 0x00A4 / 0x00AC / 0x00B4 / 0x00BC

IRQ0 / IRQ1 / IRQ2 / IRQ3 / IRQ4 / IRQ5 / IRQ6 / IRQ7 Interrupt Status of INT63 to INT32 Register

Each interrupt output will have a pair of 32-bit registers that show all the 64 interrupt input status (active or inactive), excluding those not enabled.

This will give the programmer more flexibility/ability to control the priority.

Bit	Attribute	Default	Description
31:0	RW0S	00h	Indicate Interrupt Input Status: INT63 to INT32

Note:
Interrupt Status Register is written one to clear. If interrupt trigger mode is edge mode (both posedge and negedge), its status can be cleared only by writing one. If the interrupt trigger mode is high level, it can be cleared only by disabling interrupt input.

UART Control Registers

UART registers are accessible to ARM CPU in a 64 kB block.
There are four identical UART instantiated in WM8505. Each of them has dedicated memory base address.

UART0 Base address: 0xD820:0000
UART1 Base address: 0xD82B:0000
UART2 Base address: 0xD821:0000
UART3 Base address: 0xD82C:0000
UART4 Base address: 0xD837:0000
UART5 Base address: 0xD838:0000

Offset 0x0000

URTDR

Transmit Data Register

Bit	Attribute	Default	Description
31:8	RO	0	Reserved.
7:0	RW	0	TXDATA Transmit Data.

Offset 0x0004

URRDR

Receive Data Register

Bit	Attribute	Default	Description
31:10	RO	0	Reserved
9	RO	0	FER Frame Error. This bit is the same as URISR[9].
8	RO	0	PER Parity Error. This bit is the same as URISR[8].
7:0	RO	0	RXDATA Receive Data.

Offset 0x0008

URDIV

UART Clock Divisor & Baud Rate Divisor Register

Bit	Attribute	Default	Description
31:20	RO	0	Reserved
19:16	RW	0Fh	URCLK_DIV UART Clock Divisor. These bits are used to divide the *uart_clk_source* signal to lower frequency.
15:10	RO	0	Reserved
9:0	RW	0	BRD Baud Rate Divisor. Before writing to this register, please make sure the status bit-BRDST is zero.

Offset 0x000C

URLCR

UART Line Control Register

Bit	Attribute	Default	Description
31:11	RO	0	Reserved
10	RW	0	PSLVERR Enable APB Interface Error Response. 0: Disabled; APB interface signal, *ur_pslverr*, will remain logic low at all time. 1: Enable this module to support AMBA3 APB Error response signal. When enabled, APB interface signal, *ur_pslverr*, will assert when APB interface attempts to read the empty RX FIFO.
9	RW	0	BKINIT BlueTooth Break Signal Initiation. 1: Trigger a sequence to generate the break signal character on *UR_TXDATA* output. Do not try to clear this bit by writing a "0" to it. This bit will be self-cleared upon the completion of break signal sequence. Poll URISR[12] or wait for an interrupt to detect completion of BREAK sequence. Setting this bit will reset the transmitter, the receiver, and the FIFOs even if the transmitter or receiver is active.
8	RW	0	DMAEN DMA Enable. 0: Disable DMA mode. 1: Enable UART DMA mode. The UART MUST be configured in FIFO mode (URFCR[0] == 1) if DMA is enabled.
7	RW	0	Reserved
6	RW	0	RTS Request to Send. A software controlled RTS modem signal; used when IrDA is disabled.
5	RW	0	PTYMODE Parity Mode. 0: Even number of 1's are transmitted or received. 1: Odd number of 1's are transmitted or received.
4	RW	0	PTYEN Parity Enable. 0: Do not transmit or receive parity bit in data stream 1: Enable parity bit transmission in data stream
3	RW	0	STBLEN Stop Bit Length. 0: 1-bit 1: 2-bit
2	RW	0	DLEN Data (Character) Length. 0: 7-bit 1: 8-bit
1	RW	0	RXEN Receive Enable. 0: Disables UART RX logic. 1: Enables UART RX logic.
0	RW	0	TXEN Transmit Enable. 0: Disables UART TX logic. 1: Enables UART TX logic.

Offset 0x0010

URICR

IrDA Control Register

Bit	Attribute	Default	Description
31:1	RO	0	Reserved
0	RW	0	IREN Infrared Enable. 0: Disable IrDA CODEC. 1: Enable IrDA CODEC.

Offset 0x0014

URIER

UART Interrupt Enable Register

Bit	Attribute	Default	Description
31:14	RO	0	Reserved
13	RW	0	EERR Enable for APB Error Response Interrupt.
12	RW	0	EBK Enable for Break Signal Done Interrupt.
11	RW	0	ERXTOUT Enable for Receive Time Out Interrupt.
10	RW	0	EMODM Enable for Modem Control Signal-CTS Interrupt.
9	RW	0	EFER Enable for Frame Error Interrupt.
8	RW	0	EPER Enable for Parity Error Interrupt.
7	RW	0	ERXDOVER Enable for RX Over Run Interrupt.
6	RW	0	ETXDUDR Enable for TX Under Run Interrupt.
5	RW	0	ERXFF Enable for RX FIFO Full Interrupt.
4	RW	0	ERXFAF Enable for RX FIFO Almost Full Interrupt.
3	RW	0	ETXFE Enable for TX FIFO Empty Interrupt.
2	RW	0	ETXFAE Enable for TX FIFO Almost Empty Interrupt.
1	RW	0	ERXDF Enable for RX Data Register Full Interrupt.
0	RW	0	ETXDE Enable for TX Data Register Empty Interrupt.

Offset 0x0018

URISR

UART Interrupt Status Register

Bit	Attribute	Default	Description
31:14	RO	0	Reserved
13	RW1C	0	ERR APB Error Response Asserted. 1: An APB Error Response asserted.
12	RW1C	0	BK_DONE Break Signal Done. 1: The break signal sequence has completed.
11	RO	0	RXTOUT Receive Time Out. 1: The most recent received data in RX FIFO was longer than N* continuous character, and the character has not been read by APB interface yet. The status will be cleared by read of RX FIFOs. N is a programmable value stored in TOD filed of URTOD register (Offset 0x0028). This bit is only used in FIFO mode.
10	RW1C	0	TCTS Toggle of CTS(Clear-To-Send) Modem Control Signal. This bit is used when IrDA is disabled.
9	RW1C	0	FER Frame Error. 1: The frame synchronization is lost in the receiver. Frame synchronization refers to correct detection of start bit after a 1-to-0 transition and correct detection of stop bit after a 0-1 transition at the end of transaction.
8	RW1C	0	PER Parity Error.
7	RW1C	0	RXDOVER RX Over Run.
6	RW1C	0	TXDUDR TX Under Run.
5	RO	0	RXFF RX FIFO Full. This bit is used in FIFO mode.
4	RW1C	0	RXFAF RX FIFO Almost Full. This bit is used in FIFO mode.
3	RW1C	0	TXFE TX FIFO Strikes Empty (Transition from 1 to 0). This bit is used in FIFO mode.
2	RW1C	0	TXFAE TX FIFO Almost Empty (Threshold). This bit is used in FIFO mode.
1	RO	0	RXDF RX Data Register Full. This bit is used in FIFO mode.
0	RW1C	0	TXDE TX Data Register Strikes Empty (transition from 1 to 0). This bit is used in Register mode.

Offset 0x001c

URUSR

UART Status Registers

Bit	Attribute	Default	Description
31:8	RO	0	Reserved
7	RO	1b	TXFE TX FIFO Empty. This bit is used in FIFO mode.
6	RO	1b	TXDE TX Data Register Empty. This bit is used in Register mode.
5	RO	0	BRDST Baud Rate Divisor Synchronization Status. 0: The synchronization process is complete. 1: The Baud Rate Register gets a new value, and the new value has not been synchronized to UART clock.
4	RO	0	CTS Clear to Send. Current Status of CTS Signal.
3	RO	0	RXDRDY Receiver Ready. 1: The received data is ready in either URRDR or RX FIFO.
2	RO	0	RXON Receiver On. 1: The receiver is active and is receiving data.
1	RO	0	TXDBSY Transmitter Busy. 1: The TX data is being loaded to TX port from either URTDR or TX FIFO.
0	RO	0	TXON Transmitter On. 1: The transmitter is active and is transmitting data.

Offset 0x0020

URFCR

UART FIFO Control Register

Bit	Attribute	Default	Description
31:12	RO	0	Reserved
11:8	RW	0000b	RXFLV RX FIFO Almost Full Level. When RX FIFO valid entries meet/exceed this value, it asserts RXFAF interrupt if ERXFAF=1. Values will be interpreted as follows: 0000: 16 0001: 1 0010: 2 .. 1000: 8 .. 1111: 15

7:4	RW	0000b	TXFLV TX FIFO Almost Empty Level. When TX FIFO invalid entries meet/exceed this value, it asserts TXFAE interrupt if ETXFAE=1. Values will be interpreted as follows: 0000: 16 0001: 1 0010: 2 .. 1000: 8 .. 1111: 15
3	RW	0	RXFRST RX FIFO Reset. 1: Reset RX FIFO. This bit will self-clear after RX FIFO index (counter) resets to 0. It will take a number of APB clock cycles to self-clear this bit. The software should read this bit or FIFO index register to confirm reset request.
2	RW	0	TXFRST TX FIFO Reset. 1: Reset TX FIFO. This bit will self-clear after TX FIFO index (counter) resets to 0. It will take a number of APB clock cycles to self-clear this bit. The software should read this bit or FIFO index register to confirm reset request.
1	RW	0	TRAIL RX FIFO Trailing Data Flush. 0: RX FIFO will NOT force ur_dma_rx_req signal to assert if Receive Time Out detected. 1: RX FIFO will force ur_dma_rx_req signal to assert if Receive Time Out detected.
0	RW	0	FIFOEN FIFO Enable. 0: Disable FIFO 1: Enable FIFO

Offset 0x0024

URFIDX

UART FIFO Index Register

Bit	Attribute	Default	Description
31:13	RO	0	Reserved
12:8	RO	0	RXFIDX Receive FIFO Index. It indicates the number of valid entry (how many available for Read) in RX FIFO whenever FIFO is enabled.
7:5	RO	0	Reserved
4:0	RO	0	TXFIDX Transmit FIFO Index. It indicates the number of valid entry (how many available for Write) in TX FIFO whenever FIFO is enabled.

Offset 0x0028

URTOD

UART Timeout Divisor Register

Bit	Attribute	Default	Description
31:8	RO	0	Reserved
7:0	RW	0Ah	TOD UART Time Out Divisor. Bit[7:0] controls the Receive Time Out period. The value programmed in this register indicates the number of characters times of idle before Receive Time Out interrupt is asserted.

PWM Control Registers

The Pulse Width Modulation Timer module provides four timers that may be configured to generate pulse width modulated outputs.

Base address: 0xD822:0000

Offset 0x0000 – 0x0003; 0x0010 – 0x0013; 0x0020 – 0x0023; 0x0030 – 0x0033

PWM Timer 0~3 Control Register

This register controls the bits associated with PWM Timer 0~3.

Bit	Attribute	Default	Description
31:6	RO	0	Reserved
5	RW	0	TM#_LPDI Timer # Load Period/Duty Immediately Bit. 1: The PWM Timer # Period Value Register and the PWM Timer # Duty Value Register will be immediately loaded into the PWM Timer #'s period and duty cycle logic, respectively. This bit will be self-clearing, but it may be read as a one until the associated load is completed.
4	RW	0	TM#_LPSI Timer # Load Pre-scalar Immediately Bit. 1: The PWM Timer # Pre-scalar Value Register will be immediately loaded into the PWM Timer #'s pre-scalar logic. This bit will be self-clearing, but it may be read as a one until the associated load is completed.
3	RW	0	TM#_SIM Timer # Stop Immediately Bit. 1: The PWM Timer # will stop immediately and hold its present state when the PWM Timer # Enable bit (Bit 0 of this register) is inactive (a zero).
2	RW	0	TM#_AUTO Timer # Auto-reload Bit. 1: When this bit and the PWM Timer # Enable bit (Bit 0 of this register) are both active (ones), the PWM Timer # will auto-reload when it counts its period value to zero. When the PWM Timer # Enable bit (Bit 0 of this register) is inactive (a zero), the value of this bit will have no effect upon the operation of PWM Timer #, i.e. the PWM Timer # will not auto-reload when/if it counts its period to zero.
1	RW	0	TM#_INVT Timer # Invert Output Bit. 1: The PWM Timer # output will be inverted with the low pulse being the first pulse associated with each period and the high pulse being the second pulse of the period. The effect of this bit is determined at the start of each period. As a result, if the value of this bit changes during an enabled period, it will have no effect on the PWM Timer # output during that period and will only take effect at the beginning of the next period.
0	RW	0	TM#_EN Timer # Enable Bit. 0: The PWM Timer # will be disabled. 1: The PWM Timer # will be enabled. This bit will be cleared to a zero if the PWM Timer # Auto-reload bit (Bit 2 of this register) is inactive (a zero) when the PWM Timer # counts its period value to zero.

Offset 0x0004 – 0x0007; 0x0014 – 0x0017; 0x0024 – 0x0027; 0x0034 – 0x0037

PWM Timer 0~3 Pre-Scalar Value Register

This register provides the pre-scalar value associated with PWM Timer 0~3.

Bit	Attribute	Default	Description
31:10	RO	0	Reserved
9:0	RW	0	TM#_PSV Timer # Pre-scalar Value Bits. This 10-bit value provides the pre-scalar value associated with PWM Timer #. The value loaded into these bits plus 1 will be the number of PWM clocks to be counted in association with decrementing the PWM Timer # period and duty cycle values by one. This value will only be loaded into the PWM Timer #'s pre-scalar logic as one of the following conditions is met: – The PWM clock after the PWM Timer #'s pre-scalar logic counts to zero, the PWM Timer #'s period logic count is not zero, and the PWM Timer # Stop Immediate bit (Bit 3 of the PWM Timer # Control Register) is inactive (a zero). – The PWM clock after the PWM Timer #'s pre-scalar logic counts to zero, the PWM Timer #'s period logic count is not zero, and the PWM Timer # Enable bit (Bit 0 of the PWM Timer # Control Register) is active (a one). – The PWM clock after the PWM Timer #'s pre-scalar logic counts to zero, the PWM Timer #'s period logic count is zero, the PWM Timer #'s Enable bit (Bit 0 of the PWM Timer # Control Register) is active (a one), and the PWM Timer #'s Auto-reload bit (Bit 2 of the PWM Timer # Control Register) is active (a one). – The PWM clock period after the PWM Timer # Load Pre-scalar Immediate bit (Bit 4 of the PWM Timer # Control Register) and the PWM Timer #'s Enable bit (Bit 0 of the PWM Timer # Control Register) are both written with ones. – The PWM clock after the PWM Timer #'s pre-scalar logic counts to zero, the PWM Timer # Load Period/Duty Immediate bit (Bit 5 of the PWM Timer # Control Register) and the PWM Timer #'s Enable bit (Bit 0 of the PWM Timer # Control Register) are active (ones). – The PWM clock after the PWM Timer #'s pre-scalar logic count is zero, the PWM Timer #'s period logic count is zero and the PWM Timer #'s Enable bit has just been written to one. Note: This case usually happens when the PWM Timer is first enabled. As a result, this register value may be changed without affecting the present PWM Timer #'s pre-scalar logic countdown and will be loaded by one of the events just described.

Offset 0x0008 – 0x000B; 0x0018 – 0x001B; 0x0028 – 0x002B; 0x0038 – 0x003B

PWM Timer 0~3 Period Value Register

This register provides the period value associated with PWM Timer 0~3.

Bit	Attribute	Default	Description
31:12	RO	0	Reserved
11:0	RW	0	TM#_PV Timer # Period Value Bits This 12-bit value provides the period value associated with PWM Timer #. The value loaded into these bits plus 1 will be the number of PWM Timer # Pre-scalar period's association with the PWM Timer # period. This value will only be loaded into the PWM Timer #'s period logic when the PWM Timer # is enabled (when Bit 0 of the PWM Timer # Control Register is a one) and when one of the following conditions is met: – The PWM clock after the PWM Timer #'s pre-scalar logic counts to zero, the PWM Timer #'s Auto-reload bit (Bit 0 of the PWM Timer # Control Register) is a one. – The PWM clock after the PWM Timer #'s pre-scalar logic counts to zero, and the PWM Timer # Load Period/Duty Immediate bit (Bit 5 of the PWM Timer # Control Register) is a one. – The PWM clock after the PWM Timer # Load Pre-scalar Immediate bit (Bit 4 of the PWM Timer # Control Register) and the PWM Timer # Load Period/Duty Immediate bit (Bit 5 of the PWM Timer # Control Register) are written to ones. – The PWM clock after the PWM Timer #'s pre-scalar logic count is zero, the PWM Timer #'s period logic count is zero, and the PWM Timer #'s Enable bit has just been written to one. Note: This case usually happens when the PWM Timer is first enabled. As a result, this register value may be changed without affecting the present PWM Timer #'s period logic countdown and will be loaded by one of the events just described.

Offset 0x000C – 0x000F; 0x001C – 0x001F; 0x002C – 0x002F; 0x003C – 0x003F

PWM Timer 0~3 Duty-Cycle Value Register

This register provides the duty cycle value associated with PWM Timer 0~3.

Bit	Attribute	Default	Description
31:12	RO	0	Reserved
11:0	RW	0	TM#_DV Timer # Duty Value Bits. This 12-bit value provides the duty cycle value associated with PWM Timer #. The value loaded into these bits plus 1 will be the number of PWM Timer # Pre-scalar period's association with the PWM Timer # first pulse in its period. This will be the high pulse if the PWM Timer # Invert Output bit (Bit 2 of the PWM Timer # Control Register) is a zero, and will be the low pulse if the PWM Timer # Invert Output bit is a one. The PWM Timer #'s duty cycle logic will count down to zero and stay at zero until loaded again. This value will only be loaded into the PWM Timer #'s duty cycle logic when the PWM Timer # is enabled (when Bit 0 of the PWM Timer # Control Register is a one) and when one of the following conditions is met: – The PWM clock after the PWM Timer #'s pre-scalar logic counts to zero, and the PWM Timer #'s Auto-reload bit (Bit 0 of the PWM Timer # Control Register) is a one. – The PWM clock after the PWM Timer #'s pre-scalar logic counts to zero, and the PWM Timer # Load Period/Duty Immediate bit (Bit 5 of the PWM Timer # Control Register) is a one. – The PWM clock after the PWM Timer # Load Pre-scalar Immediate bit (Bit 4 of the PWM Timer # Control Register) and the PWM Timer # Load Period/Duty Immediate bit (Bit 5 of the PWM Timer # Control Register) are written to ones. – The PWM clock after the PWM Timer #'s pre-scalar logic count is zero, the PWM Timer #'s period logic count is zero, and the PWM Timer #'s Enable bit has just been written to one. Note: This case usually happens when the PWM Timer is first enabled. As a result, this register value may be changed without affecting the present PWM Timer #'s duty cycle logic countdown and will be loaded by one of the events just described.

Offset 0x0040 – 0x0043

PWM Timer Status Register

This register provides the synchronization process associated with PWM Timer 0~3.

Bit	Attribute	Default	Description
31:28	RO	0	Reserved
27	RO	0	TM3_DVRSS Timer 3 Duty Value Register Synchronization Status. 0: The synchronization process is complete. 1: PWMDVR3 receives a new value, and the new value has not been synchronized to the PWM clock domain.
26	RO	0	TM3_PVRSS Timer 3 Period Value Register Synchronization Status. 0: The synchronization process is complete. 1: PWMPVR3 receives a new value, and the new value has not been synchronized to the PWM clock domain.
25	RO	0	TM3_PSRSS Timer 3 Pre-scalar Value Register Synchronization Status. 0: The synchronization process is complete. 1: PWMPSR3 receives a new value, and the new value has not been synchronized to the PWM clock domain.
24	RO	0	TM3_CRSS Timer 3 Control Register Synchronization Status. 0: The synchronization process is complete. 1: PWMTCR3 receives a new value, and the new value has not been synchronized to the PWM clock domain.
23:20	RO	0	These bits are Read-Only zeros.
19	RO	0	TM2_DVRSS Timer 2 Duty Value Register Synchronization Status. 0: The synchronization process is complete. 1: PWMDVR2 receives a new value, and the new value has not been synchronized to the PWM clock domain.
18	RO	0	TM2_PVRSS Timer 2 Period Value Register Synchronization Status. 0: The synchronization process is complete. 1: PWMPVR2 receives a new value, and the new value has not been synchronized to the PWM clock domain.
17	RO	0	TM2_PSRSS Timer 2 Pre-scalar Value Register Synchronization Status. 0: The synchronization process is complete. 1: PWMPSR2 receives a new value, and the new value has not been synchronized to the PWM clock domain.
16	RO	0	TM2_CRSS Timer 2 Control Register Synchronization Status. 0: The synchronization process is complete. 1: PWMTCR2 receives a new value, and the new value has not been synchronized to the PWM clock domain.
15:12	RO	0	These bits are Read-Only zeros.
11	RO	0	TM1_DVRSS Timer 1 Duty Value Register Synchronization Status. 0: The synchronization process is complete. 1: PWMDVR1 receives a new value, and the new value has not been synchronized to the PWM clock domain.
10	RO	0	TM1_PVRSS Timer 1 Period Value Register Synchronization Status. 0: The synchronization process is complete. 1: PWMPVR1 receives a new value, and the new value has not been synchronized to the PWM clock domain.

9	RO	0	TM1_PSRSS Timer 1 Pre-scalar Value Register Synchronization Status. 0: The synchronization process is complete. 1: PWMPSR1 receives a new value, and the new value has not been synchronized to the PWM clock domain.
8	RO	0	TM1_CRSS Timer 1 Control Register Synchronization Status. 0: The synchronization process is complete. 1: PWMTCR1 receives a new value, and the new value has not been synchronized to the PWM clock domain.
7:4	RO	0	These bits are Read-Only zeros.
3	RO	0	TM0_DVRSS Timer 0 Duty Value Register Synchronization Status. 0: The synchronization process is complete. 1: PWMDVR0 receives a new value, and the new value has not been synchronized to the PWM clock domain.
2	RO	0	TM0_PVRSS Timer 0 Period Value Register Synchronization Status. 0: The synchronization process is complete. 1: PWMPVR0 receives a new value, and the new value has not been synchronized to the PWM clock domain.
1	RO	0	TM0_PSRSS Timer 0 Pre-scalar Value Register Synchronization Status. 0: The synchronization process is complete. 1: PWMPSR0 receives a new value, and the new value has not been synchronized to the PWM clock domain.
0	RO	0	TM0_CRSS Timer 0 Control Register Synchronization Status. 0: The synchronization process is complete. 1: PWMTCR0 receives a new value, and the new value has not been synchronized to the PWM clock domain.

SPI Control Registers

Base address of SPI0: 0xD824:0000
Base address of SPI1: 0xD825:0000

Offset 0x0000

S2CR

SPI Control Register

Bit	Attribute	Default	Description
31:21	RW	0	Transmit Clock Divider. When in master mode, the SPI clock domain frequency is equal to the spi_clk input frequency divided by 2*"Transmit Clock Divider Bits". A value of "0" in the Transmit Clock Divider will cause no divisor. Hence, the divisor can be any even number in {1,2,4,6, ..., 4096). This is also the frequency of the spi_clk_out output when in master mode and transmitting. For Transmit Clock Divider value of "0", the spi_clk input frequency is by 50/50 duty cycle to achieve the maximum frequency. It can be 33/66 duty cycle if the setup and hold timing requirements are met. Example: To generate an output clock of 6.25 MHz with an input spi_clk clock of 100 MHz, Bit[31:21] should be set to 0x8. This will create a divisor of 16 (100 MHz / 16 = 6.25 MHz).
20:19	RW	00b	Slave Selection. When in master mode, the 2 bits indicate the spi_ssn_out_*, spi_ssn_en_* and spi_ssn_in_* signals that will be active for the coming transaction: 00: spi_ssn_out_a, spi_ssn_en_a, and spi_ssn_in_a. 01: spi_ssn_out_b, spi_ssn_en_b, and spi_ssn_in_b. 10: spi_ssn_out_c, spi_ssn_en_c, and spi_ssn_in_c. 11: spi_ssn_out_d, spi_ssn_en_d, and spi_ssn_in_d.
18	RW	0	Transmit FIFO Byte Write Method. Bit[18] causes only even numbered bytes from apb_wr_data to be stored in the transmit FIFO if apb_data_size is 01b; only Bit[7:0] are stored. If apb_data_size is 1xb, only Bit[23:16] and [7:0] are stored. For more information see Table 5 below.
17	RW	0	Receive FIFO Reset. Bit[17] forces a reset of the Receive FIFO pointers that results in the FIFO to change to empty. 1: Remain until it has been synchronized into the SPI clock domain, and after which it will be returned to 0b. This bit should be checked for a value of 0b before resuming normal Receive FIFO accesses.
16	RW	0	Transmit FIFO Reset. Bit[16] forces a reset of the Transmit FIFO pointers that results in the FIFO to change to empty. 1: Remain until it has been synchronized into the SPI clock domain, and after which it will be returned to 0b. This bit should be checked for a value of 0b before resuming normal Transmit FIFO accesses.

15	RW	0	DMA Request Control. Bit[15] allows requests for threshold passing to drive external apb_spi_tx_req or apb_spi_rx_reg. 0: apb_spi_tx_req and apb_spi_rx_req are disabled. 1: Threshold requests driven on appropriate output: Transmit FIFO threshold request on apb_spi_tx_req; Receive FIFO threshold request on apb_spi_rx_req.
14	RW	0	Receive FIFO Threshold Selection. Bit[14] selects the point at which a Receive threshold DMA request is asserted. The request is output through the apb_spi_rx_req output or the apb_spi_irq output (see Bit[15] and [7] of SPI Control Register). The Receive FIFO should hold at least 32 bytes of data. 0: Receive FIFO Requesting read of ½ of the number of bytes in the receive FIFO. 1: Receive FIFO Requesting read of ¼ of the number of bytes in the receive FIFO.
13	RW	0	Transmit FIFO Threshold Selection. Bit[13] selects the point at which a Transmit threshold DMA request is asserted. The request is output through the apb_spi_tx_req output or the apb_spi_irq output (see Bit[15] and [7] of SPI Control Register). The Transmit FIFO should hold at least 32 bytes of data. 0: Transmit FIFO Requesting write of ½ of the number of bytes in the transmit FIFO. 1: Transmit FIFO Requesting write of ¼ of the number of bytes in the transmit FIFO.
12	RW	0	Transmit FIFO Under-run Interrupt Enable. Bit[12] enables Transmit FIFO Under-run Interrupt Bit (SPI Status Register Bit[12]) to be passed to the apb_spi_irq output. 0: Transmit FIFO Under-run Interrupt Bit NOT passed to apb_spi_irq output. 1: Transmit FIFO Under-run Interrupt Bit passed to apb_spi_irq output.
11	RW	0	Transmit FIFO Empty Interrupt Enable. Bit[11] enables Transmit FIFO Empty Interrupt Bit (SPI Status Register Bit[11]) to be passed to the apb_spi_irq output. 0: Transmit FIFO Empty Interrupt Bit NOT passed to apb_spi_irq output. 1: Transmit FIFO Empty Interrupt Bit passed to apb_spi_irq output.
10	RW	0	Receive FIFO Over-run Interrupt Enable. Bit[10] enables Receive FIFO Over-run Interrupt Bit (SPI Status Register Bit[10]) to be passed to the apb_spi_irq output. 0: Receive FIFO Over-run Interrupt Bit NOT passed to apb_spi_irq output. 1: Receive FIFO Over-run Interrupt Bit passed to apb_spi_irq output.
9	RW	0	Receive FIFO Full Interrupt Enable. Bit[9] enables Receive FIFO Over-run Interrupt Bit (SPI Status Register Bit[10]) to be passed to the apb_spi_irq output. 0: Receive FIFO Over-run Interrupt Bit NOT passed to apb_spi_irq output. 1: Receive FIFO Over-run Interrupt Bit passed to apb_spi_irq output.
8	RW	0	Receive FIFO Empty Interrupt Enable. Bit[8] enables Receive FIFO Empty Interrupt Bit (SPI Status Register Bit[8]) to be passed to the apb_spi_irq output. 0: Receive FIFO Empty Interrupt Bit NOT passed to apb_spi_irq output. 1: Receive FIFO Empty Interrupt Bit passed to apb_spi_irq output.

7	RW	0	Threshold IRQ/DMA Selection. Bit[7] causes the Transmit threshold to request "Transmit FIFO Threshold Passed Interrupt (SPI Status Register Bit[13])" to be output through the apb_spi_irq output. The status bits are valid at all times. 0: TX threshold requests not passed to apb_spi_irq output. 1: TX threshold requests passed to apb_spi_irq output.
6	RW	0	Interrupt Enable. Bit[6] enables the apb_spi_irq output signal. 0: apb_spi_irq output disabled. 1: apb_spi_irq output enabled Polling of the SPI Status Register can be used to sense module status.
5	RW	0	Module Enable. Bit[5] turns the SPI module on (1b) and off (0b). 0: The SPI module will not perform any SPI transfer. Any transmission in progress will not complete. 1: Enables SPI operation according to register settings. Data transfer will start if the Master mode is enabled and transmit data available. Note: All SPI registers can be accessed through the APB interface independently of this bit value. However, when changing other control bits affects the function of this module, the value of this bit must be equal to 0 to avoid unexpected results.
4	RW	0	Mode Fault Error Interrupt Enable. Bit[4] enables Mode-Fault Error Interrupt Bit (Bit[4] of SPI Status Register) to be passed to the apb_spi_irq output. 0: Mode-Fault Error Interrupt Bit NOT passed to apb_spi_irq output. 1: Mode-Fault Error Interrupt Bit passed apb_spi_irq output.
3	RW	1b	Master/Slave Mode Select. Bit[3] controls the configuration of this SPI module related to the external device that it will be communicating with. 0: SPI configured as a master. 1: SPI configured as a slave.
2	RW	0	Clock Polarity Select. Bit[2] controls the edge of the SPI external interface clock that is active. 0: Data captured on falling edge; SPI_CLK idles low. 1: Data captured on rising edge; SPI_CLK idles high. SPI modes are defined using clock phase and clock polarity as : (polarity, phase) (0,0) = SPI mode 0 - data rising edge triggered. (0,1) = SPI mode 1 - data falling edge triggered. (1,0) = SPI mode 2 - data falling edge triggered. (1,1) = SPI mode 3 - data rising edge triggered.
1	RW	0	Clock Phase Select. 0: Transfers begin following the activation of SPI_SSN. 1: Transfers begin on the first SPI_CLK edge.
0	RW	0	Mode Fault Error Feature Enable. Bit[0] enables the mode fault error feature. 0: Mode Fault Error checking is disabled – for use when this SPI is the only master. 1: Mode Fault Error checking is enabled - for use when this SPI is part of a multi-master configuration.

Table 5 - Transmit FIFO Byte Write Method

Transmit FIFO Byte Write Method	**apb_data_size** **00b – 1 Byte** **01b – 2 Bytes** **1xb – 4 Bytes**	**TX FIFO Write Action** **apb_wr_data[7:0] - B0** **apb_wr_data[15:8] - B1** **apb_wr_data[23:16] - B2** **apb_wr_data[31:24] - B3**
0	1B, 2B, 4B	store all bytes (based on APB data size)
1	2B	store B0 byte
1	4B	store B0 and B2 bytes

Offset 0x0004

S2SR

SPI Status Register

Bit	**Attribute**	**Default**	**Description**
31:24	RO	0	RX FIFO Count. Real Time Count of Valid Data Remaining in RX FIFO. Supports FIFO depth up to 256.
23:16	RO	24h	TX FIFO Count. Real Time Count of Unoccupied Slots in TX FIFO Available for Writing to. Supports FIFO depth up to 256.
15	RO	0	TX FIFO Empty Status. Real Time TX FIFO Empty Status – Non-sticky for Use as Polling Bit.
14	RW1C	0	Receive FIFO Threshold Passed Interrupt. Bit[14] identifies when the space occupied in the receive FIFO matches or is greater than the programmed threshold level. 0: Receive FIFO available space is greater than threshold level. 1: Receive FIFO occupied space is ≥ programmed threshold.
13	RW1C	0	Transmit FIFO Threshold Passed Interrupt. Bit[13] identifies when the space available in the transmit FIFO matches or is greater than the programmed threshold level. 0: Transmit FIFO occupied greater than threshold level. 1: Transmit FIFO available space is ≥ programmed threshold
12	RW1C	0	Transmit FIFO Under-run Interrupt. Bit[12] identifies when an SPI transaction occurs after the Transmit FIFO has emptied and SPI is in slave mode. 0: Transmit FIFO is occupied. 1: Transmit FIFO is Under-run.
11	RW1C	0	Transmit FIFO Empty Interrupt Bit[11] is triggered when the transmit FIFO content changed from not empty to empty. Use SPI Status Register Bit[15] for polling. 0: Transmit FIFO has transitioned to a non-zero count or is not transitioning, including having a content count of zero. 1: Transmit FIFO has transitioned to Empty.
10	RW1C	0	Receive FIFO Over-run Interrupt Bit[10] identifies when a SPI external communication has attempted to write to a full Receive FIFO. When the Receive FIFO is full, the incoming data is dropped. 0: Not Receive FIFO over-run. 1: Write attempt to full Receive FIFO.

9	RW1C	0	Receive FIFO Full Interrupt. Bit[9] identifies when the Receive FIFO content changed from not full to full (36 bytes). When the Receive FIFO is full, the incoming data is dropped. 0: Receive FIFO has transitioned to some count less than full or is not transitioning when the FIFO is full. 1: Receive FIFO has transitioned to Full
8	RW1C	0	Receive FIFO Empty Interrupt. Bit[8] identifies when the Receive FIFO content changed from not empty to empty, not when the receive FIFO is empty. 0: Receive FIFO has transitioned to a non-zero count or is not transitioning that includes having at a content count of zero. 1: Receive FIFO has transitioned to Empty.
7	RO	0	spi_busy Bit[7] identifies when the SPI is working. 0: Disable SPI and issue new transaction.
6:5	RO	0	Reserved
4	RW1C	0	Mode-Fault Error Interrupt. Bit[4] is low if the spi_ssn_in_* input is active low while the SPI is configured as a master device. It identifies that an external device is also configured as a master. To prevent damage due to conflict between output drivers, the SPI module will change the Master/Slave Mode Select control bit to slave mode and generate an apb_spi_irq interrupt subject to the Interrupt Enable Bit (SPI Control Register Bit[7]). This bit is cleared one cycle later after you write 1 to this bit. 0: No Mode Fault Error. 1: Mode Fault Error occurred.
3:0	RO	0h	Reserved

Offset 0x0008

S2DFCR

SPI Data Format Control Register

Bit	Attribute	Default	Description
31:28	RW	0	SSN Preset Counter. This Counter will determine the time from SPI_SS# falling edge to SPI_CLK running, and also the time that SPI_SS# still needs to keep low after SPI_CLK stops. 0h: No time 1h: 1 half spi_clk_out clocks 2h: 1 spi_clk_out clocks Eh: 14 half spi_clk_out clocks Fh: 15 half spi_clk_out clocks … If the clock divider is zero, this delay can not be half cycle; it will be added extra half cycle if the delay clocks number is not integer. The register is only available when the phase is 0. If the phase is one, writing in this register will have no effect.
27	RO	0	Reserved
26	RW	0	SSN Hold EN. When SPI is working in master mode, and the phase is zero, this bit is used to control if the SSN signal deassert for every byte. 0: SSN will be asserted and deasserted for every byte transfer. 1: SSN will hold low until the TXFIFO empty.

25	RW	0	Microwire EN. When SPI working in master mode, this bit is set one to connect to the Microwire compatible bus. If SPI working in slave mode, this bit will be ignored. 0: Sample the data in the middle of the output cycle (SPI Bus protocol). 1: Sample the data at the end of the output cycle (Microwire compatible).
24	RW	0	RX Threshold Passed interrupt Enable. This bit causes the Receive threshold requests "Receive FIFO Threshold Passed Interrupt (SPI Status Register Bit[14])" to be output through the apb_spi_irq output. The status bits are valid at all times. 0: RX threshold requests not passed to apb_spi_irq output. 1: RX threshold requests passed to apb_spi_irq output.
23:16	RW	00h	Mode Fault Delay Count. Number of SPI Clock Domain Clocks Counted after a Master SPI Stops Driving spi_ssn_out_* and before Starting Mode Fault Checking on spi_ssn_in_*. Bit[23:16] is only valid when SPI Control Bit[5] and [0] are enabled, and SPI Control Bit[3] is in master mode. 00h: Delay 2 clocks 01h: Delay 2 clocks 02h: Delay 2 clocks … FFh: Delay 255 clocks Example: The SPI clock is 25 MHz; external pull-up on SPI_SSN causes 400 ns rise time after master SPI stops driving spi_ssn_out. Set Mode Fault Delay Count value greater than calculation value of 400 ns/(1/25 MHz) = 10 Note: The checking for mode fault error mechanism can wait never less than 2 clocks as need to sync the spi_ssn_in twice.
15:8	RW	0	TX Drive Count. Number of Bytes to be Transmitted Using TX No Data Value (SPI Data Format Control Register Bit[6]). 00h: Continuous transmission of TX No Data Value until SPI disabled or this feature disabled (SPI Data Format Control bit 7). 01h: One byte transmitted. … FFh: 255 bytes transmitted.
7	RW	0	TX Drive Enable. Enable the TX logic to transmit a fixed data value for a programmed (TX Drive Count) number of bytes when [1] the TXFIFO is empty and the SPI becomes enabled (SPI Control Register Bit[5]) or [2] the TXFIFO has just transitioned to empty. 0: Feature disabled 1: Enable this feature
6	RW	0	TX No Data Value. The Value Driven out of Transmits Logic when TX FIFO Empty and the SPI in Slave Mode or when TX Drive Enabled. 0: 0 driven to MOSI or MISO. 1: 1 driven to MOSI or MISO.
5	RW	1b	Direct SSN Enable. Value Driven to *spi_ssn_en_x* if SSN Control Enabled. Support SSN Port Mode as Set to 1. 0: 0 driven on spi_ssn_en_x - SSN active. 1: 1 driven on spi_ssn_en_x - SSN disable. See Table 6 Direct SSN Control below.

4	RW	1b	Direct SSN Value. Value Driven to spi_ssn_out if SSN Control Enabled. Support SSN Port Mode as Set to 0 and 1. 0: 0 driven on spi_ssn_out - SSN active. 1: 1 driven on spi_ssn_out - SSN disable. See Table 6 Direct SSN Control below.
3	RW	0	SSN Control. Set control of spi_ssn_out and spi_ssn_en when SPI is Master and phase is one. 0: Auto controlled by the hardware to be active for one byte at a time. 1: Controlled by programming "Direct SSN Value" and "Direct SSN Enable" control bits. See Table 6 Direct SSN Control below.
2	RW	0	SSN Port Mode. I/O Mode Selection for SPI_SSN Port while SPI is Master. 0: Open drain configuration to allow mode fault error checking (multi-master). 1: Regular drive configuration as slave select signal (point-to-point). See Table 6 Direct SSN Control below.
1	RW	0	Receive Significant Bit Order. Identify the bit of the incoming receive byte that was sent first. 0: MSB first - Receive byte is stored in Receive FIFO with MSB in bit 7 position. 1: LSB first - each byte received will be flipped so that when the byte is written into the Receive FIFO, the MSB of the byte is in bit 0 position.
0	RW	0	Transmit Significant Bit Order. Determine the bit of the outgoing transmit byte that is sent first. 0: MSB first 1: LSB first

Table 6 - Direct SSN Control

SPI Mode	SPI MODE (polarity, phase)	SSN Control	SSN Port Mode	Direct SSN Enable	Direct SSN Value	SPI_SSN I/O Port
Slave	X	X	X	X	X	not driven
Master	all	0	0	X	X	open drain auto-controlled
Master	all	0	1	X	X	regular driven auto-controlled
Master	(0,1), (1,1)	1	0	X	0	driven low
					1	external pull up controls
Master	(0,1), (1,1)	1	1	1	X	disabled with spi_ssn_en low
Master	(0,1), (1,1)	1	1	0	0	driven low
					1	driven high

Note:
"X" refers to "no effect".

Offset 0x000C

S2CRE

SPI Extended Control Register

Bit	Attribute	Default	Description
31:8	RO	0	Reserved
7:0	RW	00h	SSN Deassert Width Counter. This Counter will determine the time from SPI_SSN deassert to next SPI_SSN assert. 00h: 1 spi_clk_out clock cycle. 01h: 1 spi_clk_out clock cycle. 02h: 2 spi_clk_out clock cycles. … FEh: 254 spi_clk_out clock cycles. FFh: 255 spi_clk_out clock cycles.

Offset 0x0010 - 0x002F

S2TXFIFO

SPI Transmit FIFO (range for DMA)

Bit	Attribute	Default	Description
31:0	WO	0	SPI Transmit Register. Bit[31:0] is for writing SPI transmit data into the transmit FIFO. The transmit FIFO is 1 word (8 bits) in width by 32 words or deeper. APB bus writes based on apb_data_size are supported for writing to this register. Bytes from Write are loaded into FIFO Least Significant Byte first. See also SPI Control Register Bit[18] for further information about writing methods supported. New data will be dropped when attempting to write to a full FIFO. The address range to support the DMA burst access which requires contiguous address accesses is apb_addr[9:0] 10h thru 2Fh. This range does not map to the specific FIFO locations, but instead maps to a single access to the FIFO were the location is controlled by the FIFO in counter. The maximum DMA burst access would be 16 accesses of apb_data_size = 01b or 8 accesses of apb_data_size = 10b if the burst start address is 10b. If the burst address exceeds the address range, the written data will be lost.

Offset 0x0030 - 0x004F

S2RXFIFO

SPI Receive FIFO (range for DMA)

Bit	Attribute	Default	Description
31:0	RO	0	SPI Receive Register. Bit[31:0] is for reading data from the receive FIFO. The receive FIFO is 1 word (8 bits) in width and 32 words or more deeper. APB bus reads based on apb_data_size are supported for reading from this register. Zero's will be returned when attempting to read from an empty receive FIFO. Attempting to read a larger data_size than what is available in the FIFO will cause an undetermined read value and the FIFO will not empty. A smaller apbb_data_size is required to properly empty the FIFO. This is especially important for odd size fills of the Receive FIFO. SPI2 Status bits RX FIFO Count Bit[25:20] can be used to determine the correct read size. The address range to support the DMA burst access which requires contiguous address accesses is apb_addr[9:0] 30h thru 4Fh. This range does not map to the specific FIFO locations, but instead maps to a single access to the FIFO location controlled by the FIFO out counter. The maximum DMA burst access would be 16 accesses of apb_data_size = 01b or 8 accesses of apb_data_size = 10b if the burst start address is 30h. If the burst address exceeds the address range, the meaningless data will be read and should be all zero.

Keypad Controller Registers

The Keypad module contains a number of controls and status registers to support the keypad scan functionality. The associated registers will be accessed via the 64 k-byte address range assigned to the KPAD module. The apbb_kpad_psel signal will be active (a one) whenever this 64 k-byte address range is decoded and the appropriate PWM register will be accessed in accordance to the following address offsets.

Base address: 0xD826:0000

Offset 0x0000 – 0x0003

Keypad Matrix Control Register

This register supports the control bits associated with the Keypad Matrix.

Bit	Attribute	Default	Description
31:24	RO	0	Reserved
23:16	RW	0	mtx_ctl_reg_23_16[23:16] Manual Keypad Matrix Scan Output Signals. Bit[23:16] provides the Keypad Matrix column output signals when operating in manual Keypad Matrix scan mode (as Bit[3:2] of this register are both zeros and Bit 0 of this register is a one). When automatic Keypad Matrix scan mode is enabled (as either Bit[3:2] of this register is a one and Bit[0] of this register is a one), the Keypad Matrix column output signals selected by Bit[10:8] of this register will operate under the control of the automatic scan mode logic function. For the Keypad Matrix column output signals not selected by Bits[10:8] of this register, these bits will always drive the associated signals. As a result, such Keypad Matrix column output signals may then be used as General Purpose Output signals.
15	RO	0	Reserved
14:12	RW	000b	mtx_ctl_reg_14_12[14:12] Keypad Matrix Row Number Bits. These bits indicate the number of Keypad Matrix rows supported. The number of Keypad Matrix rows supported is always one greater than the value of these bits: 000: One Row Supported (gpio_kpad_row_in[0]). 001: Two Rows Supported (gpio_kpad_row_in[1:0]). 010: Three Rows Supported (gpio_kpad_row_in[2:0]). … 111: Eight Rows Supported (gpio_kpad_row_in[7:0]).
11	RO	0	Reserved
10:8	RW	0	mtx_ctl_reg_10_8[10:8] Keypad Matrix Column Number Bits. These bits indicate the number of Keypad Matrix columns supported. The number of Keypad Matrix columns supported is always one greater than the value of these bits: 000: One Column Supported (kpad_col[0]). 001: Two Columns Supported (kpad_col[1:0]). 010: Three Columns Supported (kpad_col[2:0]). … 111: Eight Columns Supported (kpad_col[7:0]).
7:5	RO	0	Reserved

4	RW	0	mtx_ctl_reg_4_0[4] Keypad Matrix Ignore Multiple Key-press Bit. 0: Following the completion of a scan sequence, all of the associated keys in the Keypad Matrix do not have to be released inactive before any new key press activity is detected 1: Following the completion of a scan sequence, all of the associated keys in the Keypad Matrix must be released inactive before any new key press activity is detected.
3	RW	0	mtx_ctl_reg_4_0[3] Keypad Matrix Automatic Scan on Activity Bit. 0: The Keypad Matrix interface will not perform an automatic scan. 1: The Keypad Matrix interface will perform an automatic scan upon detecting activity. Note: Bit 2 of this register takes precedence over this bit when they are both active (ones).
2	RW	0	mtx_ctl_reg_4_0[2] Keypad Matrix Automatic Scan Bit. 0: The Keypad Matrix interface will not perform an automatic scan. 1: The Keypad Matrix interface will perform an automatic scan and then this bit will self clear. Note: When active (a one), this bit takes precedence over Bit[3] of this register when it is active (a one), in other words, an automatic scan will be performed independent of any activity when this bit is active (a one). Additionally, if this bit goes active (goes from a zero to a one) at any time during or after an automatic scan performed as a result of Bit[3] of this register, the results of that automatic scan will be discarded and a new automatic scan based on this bit will begin.
1	RW	0	mtx_ctl_reg_4_0[1] Keypad Matrix Interrupt Request Enable Bit. 0: The Keypad Matrix interface interrupt request will be disabled. 1: The Keypad Matrix interface interrupt request is enabled, i.e. whenever the valid Keypad Matrix keystroke information is detected, an active interrupt request will be generated.
0	RW	0	mtx_ctl_reg_4_0[0] Keypad Matrix Enable Bit. 0: The Keypad Matrix interface will be disabled. 1: The Keypad Matrix interface will be enabled.

Offset 0x0004 – 0x0007

Keypad Direct Input Control Register

This register supports the control bits associated with the Keypad Direct Inputs.

The Keypad Row input signals, which are not configured to be associated with the Keypad Matrix support, may be used as Keypad Direct Inputs and will be controlled by the bits in this register.

The Keypad Row input signals which are configured to act as Keypad Matrix Row input signal can not be used as a Keypad Direct Input because the hardware will prevent them from being used in this manner.

Bit	Attribute	Default	Description
31:24	RO	0	Reserved
23:16	RW	0	dir_ctl_reg_23_16[23:16] Keypad Direct Input Enable Bit[7:0]. These bits provide individual enable bit for each of the possible Key- pad Direct Input signals. When these bits are inactive (zeros), then the corresponding Keypad Row signal will not be used as a Keypad Direct Input signal. To be used as a Keypad Direct Input signal, the corresponding bit must be active(a one) and the associated Keypad Row signal must not be selected to support the Keypad Matrix functionality.
15:5	RO	0	Reserved
4	RW	0	dir_ctl_reg_4_3[4] Keypad Direct Input Ignore Multiple Key-press Bit 0: Following the completion of a scan sequence, all of the associated Keypad Direct Input signals do not have to be released inactive before any new key press activity is detected. 1: Following the completion of a scan sequence, all of the associated Keypad Direct Input signals must be released inactive before any new key-press activity is detected.
3	RW	0	dir_ctl_reg_4_3[3] Keypad Direct Input Automatic Scan on Activity Bit. 0: The Keypad Direct Input interface will not perform an automatic scan. Read Bit[7:0] of the Keypad Direct Input Key Scan Register (at Offset address 0x0014). When this bit is inactive (a zero), it will provide the present Keypad Direct Inputs, and the way the software may immediately read the Keypad Direct Input values. 1: The Keypad Direct Input interface will perform an automatic scan upon detecting activity.
2	RO	0	Reserved
1	RW	0	dir_ctl_reg_1_0[1] Keypad Direct Input Interrupt Request Enable Bit. 0: The Keypad Direct Input interface interrupt request will be disabled. 1: The Keypad Direct Input interface interrupt request is enabled, i.e. whenever the valid Keypad Direct Input signal information is detected, an active interrupt request will be generated.
0	RW	0	dir_ctl_reg_1_0[0] Keypad Direct Input Enable Bit. 0: The Keypad Direct Input interface will be disabled. 1: The Keypad Direct Input interface will be enabled.

Offset 0x0008 – 0x000B

Keypad Invert Input Control Register

The purpose of this register is to indicate whether each of the Keypad Row input signals is active-high or active-low.

Bit	Attribute	Default	Description
31:24	RO	0	Reserved
23:16	RW	0	inv_inp_reg[7:0] Keypad Invert Input Bits 7-0. 0: The corresponding Keypad Row input signal is considered to be an active-high signal. 1: The corresponding Keypad Row input signal is considered to be an active-low signal.
15:0	RO	0	Reserved

Offset 0x000C – 0x000F

Keypad Status Register

This register is to indicate any active sources that may have generated an interrupt request.

Bit	Attribute	Default	Description
31:4	RO	0	Reserved
3	RW	0	dir_input_comp_flag Keypad Direct Input Active Bit. 0: No active direct Keypad input has been detected. 1: An active direct Keypad input has been detected. Once this bit is active (a one), to clear it to a zero, a one must be written to this bit.
2	RW	0	auto_scan_immed_comp_flag Keypad Matrix Automatic Scan Completed Bit. 0: No software initiated automatic Keypad Matrix scan has been completed. 1: A software initiated automatic Keypad Matrix scan has been completed after the software wrote a one to the Keypad Matrix Automatic Scan bit (Bit[2] of the Keypad Matrix Control Register). Once this bit is active (a one), to clear it to a zero, a one must be written to this bit.
1	RW	0	auto_scan_act_comp_flag Keypad Matrix Automatic Scan on Activity Bit. 0: No key-press activity has been detected and de- bounced. 1: The logic has detected and debounced one or more key-presses. Once this bit is active (a one), to clear it to a zero, a one must be written to this bit.
0	RW	0	man_scan_act_key_flag Keypad Matrix Manual Debounce Active Key Bit. 0: No active key-press has been detected during manual Keypad Matrix scan mode. 1: An active key-press has been detected during manual Keypad Matrix scan mode. Once this bit is active (a one), to clear it to a zero, a one must be written to this bit.

Offset 0x0010 – 0x0013

Keypad Matrix Primary Key Automatic Scan Register

This register provides the primary Keypad Matrix Automatic scan results.

Bit	Attribute	Default	Description
31	RO	0	auto_scan_pri_valid Valid Scan Bit. 0: The other bits in this register do not contain any valid information. 1: A valid Keypad Matrix automatic scan has completed and the other bits in this register contain information regarding the results of that scan. If this bit is active (a one), it will be cleared to a zero (inactive), when it is read. Note: While this bit is active (a one), no subsequent Keypad Matrix automatic scans due to activity will be performed. So once an active key-press has occurred and is scanned, this register will hold that key-press information until this register is read.
30:29	RO	00b	auto_scan_keys_num Keys Pressed Bits. These bits indicate the number of keys that have been detected as being pressed when Bit[31] of this register is active (a one): 00: No Keys have been pressed. 01: One key has been pressed. 1X: Two or more keys have been pressed and the Keypad Matrix Multiple Keys Scan Registers 3-0 should be read to determine the keys that have been pressed. When Bit[31] of this register is inactive (a zero), the value of these bits has no effect.
28:7	RO	0	Reserved
6:4	RO	any value	auto_scan_press_row Row Pressed Bits. When Bit[31] of this register is active (a one), and Bit[30:29] of this register indicates only one key has been pressed (when they are equal to 2’b01), these bits will indicate the row associated with the key that has been pressed. When Bit[31] of this register is inactive (a zero), or Bit[30:29] of this register are not equal to 2’b01, the value of these bits is a has no effect.
3	RO	0	Reserved
2:0	RO	any value	Column Pressed Bits (auto_scan_press_col) When Bit[31] of this register is active (a one), and Bit[30:29] of this register indicates only one key has been pressed (when they are equal to 2’b01), these bits will indicate the column associated with the key that has been pressed. When Bit 31 of this register is inactive (a zero), or Bits[30:29] of this register are not equal to 2’b01, then the value of these bits has no effect.

Offset 0x0014 – 0x0017

Keypad Direct Input Key Scan Register

This register indicates the Keypad Direct Input scan results.

Bit	Attribute	Default	Description
31	RO	0	dir_input_valid Valid Scan Bit. When this bit is inactive (a zero), and Bit[3] of the Keypad Direct Input Control Register is active (a one), the other bits in this register do not contain any valid information. 1: A valid Keypad Direct Input scan debounce has completed and the other bits in this register contain information regarding the results of that debounce scan. If this bit is active (a one), it will be cleared to a zero (inactive), when it is read. Note: While this bit is active (a one), no subsequent Keypad Direct Input debounce scans will be performed. So once an active key-press has occurred and is debounce scanned, this register will hold that key-press information until this register is read
30:8	RO	0	Reserved
7:0	RO	0	dir_input_key_data Direct Input Active Bits. When Bit[31] of this register is active (a one), these bits will indicate the active direct input signals after being debounced. When Bit[31] of this register is inactive (a zero) and Bit[3] of the Keypad Direct Input Control Register is active (a one), then the value of these bits has no effect. When Bit[3] of the Keypad Direct Input Control Register is inactive (a zero), the value of these bits provide the present direct input signals.

Offset 0x0018 – 0x001B

Keypad Manual Matrix Key Scan Register

This register provides the Keypad Manual Matrix scan results.

Bit	Attribute	Default	Description
31:8	RO	0	Reserved
7:0	RO	0	row_mtx_qual Keypad Matrix Row Input Active Bits. Bit[7:0] indicates the active Keypad Matrix Row input signals. The Keypad Row inputs that have not been configured to support the Keypad Matrix interface via Bit[14:12] of the Keypad Matrix Control Register will always be inactive (zeros) when reading these bits.

Offset 0x001C – 0x001F

Keypad Row Input Register

The purpose of this register is to provide the Keypad Row Input signal values.

Bit	Attribute	Default	Description
31:8	RO	undefined	Reserved
7:0	RO	undefined	r_row_in Keypad Row Input Active Bits. Bit[7:0] indicates the active Keypad Row input signals.

Offset 0x0020 – 0x0023

Keypad Matrix Multiple Keys Scan Register 0

This register provides the Keypad Row Input signal values associated with Keypad Matrix Columns 1-0 for cases where multiple keys in the Keypad Matrix have been scan detected as being pressed.

Bit	Attribute	Default	Description
31	RO	0	auto_scan_col_10_valid Valid Scan Bit. 0: The other bits in this register do not contain any valid information. 1: A valid Keypad Matrix automatic scan has completed and the other bits in this register contain information regarding the results of that scan. No subsequent Keypad Matrix automatic scans due to activity will be performed. So once an active key-press has occurred and is scanned, this register will hold that key-press information until this register is read. If neither of the associated Keypad Matrix Columns is configured as a Keypad Matrix Column, then this bit will never be set active (a one) If this bit is active (a one), it will be cleared to a zero (inactive), when it is read.
30:24	RO	0	Reserved
23:16	RO	0	auto_scan_set0_col_1 Keypad Matrix Column 1 Row Input Active Bits. Bit[23:16] indicates the active Keypad Row input signals associated with Keypad Matrix Column 1. Note: The Keypad Row inputs that have not been configured to support the Keypad Matrix interface via Bit[14:12] of the Keypad Matrix Control Register will always be inactive (zeros) when reading these bits.
15:8	RO	0	Reserved
7:0	RO	0	auto_scan_set0_col_0 Keypad Matrix Column 0 Row Input Active Bits. Bit[7:0] indicates the active Keypad Row input signals associated with Keypad Matrix Column 0. Note: The Keypad Row inputs that have not been configured to support the Keypad Matrix interface via Bit[14:12] of the Keypad Matrix Control Register will always be inactive (zeros) when reading these bits.

Offset 0x0024 -0x0027

Keypad Matrix Multiple Keys Scan Register 1

This register provides the Keypad Row Input signal values associated with Keypad Matrix Columns 3-2 for cases where multiple keys in the Keypad Matrix have been scan detected as being pressed.

Bit	Attribute	Default	Description
31	RO	0	auto_scan_col_32_valid Valid Scan Bit. 0: The other bits in this register do not contain any valid information. 1: A valid Keypad Matrix automatic scan has completed and the other bits in this register contain information regarding the results of that scan. No subsequent Keypad Matrix automatic scans due to activity will be performed. So once an active key-press has occurred and is scanned, this register will hold that key-press information until this register is read. If neither of the associated Keypad Matrix Columns is configured as a Keypad Matrix Column, then this bit will never be set active (a one). If this bit is active (a one), it will be cleared to a zero (inactive), when it is read.
30:24	RO	0	Reserved
23:16	RO	0	auto_scan_set0_col_3 Keypad Matrix Column 3 Row Input Active Bits. Bit[23:16] indicates the active Keypad Row input signals associated with Keypad Matrix Column 3. Note: The Keypad Row inputs that have not been configured to support the Keypad Matrix interface via Bits[14:12] of the Keypad Matrix Control Register will always be inactive(zeros) when reading these bits.
15:8	RO	0	Reserved
7:0	RO	0	auto_scan_set0_col_2 Keypad Matrix Column 2 Row Input Active Bits. These bits will indicate the active Keypad Row input signals associated with Keypad Matrix Column 2. Note: The Keypad Row inputs that have not been configured to support the Keypad Matrix interface via Bits[14:12] of the Keypad Matrix Control Register will always be inactive (zeros) when reading these bits

Offset 0x0028 – 0x002B

Keypad Matrix Multiple Keys Scan Register 2

This register provides the Keypad Row Input signal values associated with Keypad Matrix Columns 5-4 for cases where multiple keys in the Keypad Matrix have been scan detected as being pressed.

Bit	Attribute	Default	Description
31	RO	0	auto_scan_col_54_valid Valid Scan Bit. 0: The other bits in this register do not contain any valid information. 1: A valid Keypad Matrix automatic scan has completed and the other bits in this register contain information regarding the results of that scan. No subsequent Keypad Matrix automatic scans due to activity will be performed. So once an active key-press has occurred and is scanned, this register will hold that key-press information until this register is read. If neither of the associated Keypad Matrix Columns is configured as a Keypad Matrix Column, then this bit will never be set active (a one). If this bit is active (a one), it will be cleared to a zero (inactive), when it is read.
30:24	RO	0	Reserved
23:16	RO	0	auto_scan_set0_col_5 Keypad Matrix Column 5 Row Input Active Bits. Bit[23:16] indicates the active Keypad Row input signals associated with Keypad Matrix Column 5. Note: The Keypad Row inputs that have not been configured to support the Keypad Matrix interface via Bit[14:12] of the Keypad Matrix Control Register will always be inactive (zeros) when reading these bits
15:8	RO	0	Reserved
7:0	RO	0	auto_scan_set0_col_4 Keypad Matrix Column 4 Row Input Active Bits. These bits will indicate the active Keypad Row input signals associated with Keypad Matrix Column 4. Note: The Keypad Row inputs that have not been configured to support the Keypad Matrix interface via Bit[14:12] of the Keypad Matrix Control Register will always be inactive (zeros) when reading these bits

Offset 0x002C – 0x002F

Keypad Matrix Multiple Keys Scan Register 3

This register provides the Keypad Row Input signal values associated with Keypad Matrix Columns 7-6 for cases where multiple keys in the Keypad Matrix have been scan detected as being pressed.

Bit	Attribute	Default	Description
31	RO	0	auto_scan_col_76_valid Valid Scan Bit. 0: The other bits in this register do not contain any valid information. 1: A valid Keypad Matrix automatic scan has completed and the other bits in this register contain information regarding the results of that scan. No subsequent Keypad Matrix automatic scans due to activity will be performed. So once an active key-press has occurred and is scanned, this register will that key-press information until this register is read. If neither of the associated Keypad Matrix Columns is configured as a Keypad Matrix Column, then this bit will never be set active (a one). If this bit is active (a one), it will be cleared to a zero (inactive), when it is read.
30:24	RO	0	Reserved
23:16	RO	0	auto_scan_set0_col_7 Keypad Matrix Column 7 Row Input Active Bits. Bit[23:16] indicates the active Keypad Row input signals associated with Keypad Matrix Column 7. Note: The Keypad Row inputs that have not been configured to support the Keypad Matrix interface via Bit[14:12] of the Keypad Matrix Control Register will always be inactive (zeros) when reading these bits.
15:8	RO	0	Reserved
7:0	RO	0	auto_scan_set0_col_6 Keypad Matrix Column 6 Row Input Active Bits. These bits will indicate the active Keypad Row input signals associated with Keypad Matrix Column 6. Note: The Keypad Row inputs that have not been configured to support the Keypad Matrix interface via Bit[14:12] of the Keypad Matrix Control Register will always be inactive (zeros) when reading these bits.

Offset 0x0030 – 0x0033

Keypad Matrix Debounce and Scan Interval Register

This register provides the software with a means to configure the Keypad Matrix Debounce and Scan Interval times.

Bit	Attribute	Default	Description
31:24	RO	0	Reserved
23:16	RW	00b	mtx_scan_intv_reg[7:0] Keypad Scan Interval Bits. Bit[23:16] indicate the number of Advanced High-performance Bus (AHB) clocks associated with each Keypad Scan Interval. The following indicates the number of AHB clocks to be used for the various values of these bits: 00: (1 x 1024) = 1024 AHB clocks. 01: (2 x 1024) = 2048 AHB clocks. 02: (3 x 1024) = 3072 AHB clocks. 03: (4 x 1024) = 4096 AHB clocks. 04: (5 x 1024) = 5120 AHB clocks. . FF: (256 x 1024) = 262,144 AHB clocks.
15:12	RO	0	Reserved
11:0	RW	0	mtx_dbn_reg[11:0] Keypad Matrix Debounce Interval Bits. These bits indicate the number Keypad Scan Intervals between Keypad Matrix scans that must match to detect stable active key-press values. If these bits are all zeros, then there is no Keypad Matrix Debounce Interval time and the first Keypad Matrix scan will be the only one necessary.

Note:
The Keypad Scan Interval time defines how long each Keypad Matrix Column output signal will be driven active during the automatic scan sequences.

By default a minimum of 1024 Advance High-performance Bus (AHB) clocks will be the length of a Keypad Matrix scan interval, i.e. each Keypad Matrix Column output signal will be driven active (a one) for 128 AHB clock periods, with the corresponding Keypad Matrix Row input signals sampled at the end of each of these 128 AHB clock periods.

The Keypad Matrix Debounce Interval time defines the number of Keypad Scan Interval times between Keypad Matrix Scan Intervals that must match for an active key-press to be detected.

As an example, if the Keypad Matrix Debounce Interval value is configured to be 0x008, then once an active key-press scan has occurred, the logic will wait seven Key- pad Scan Interval periods and then perform a Keypad Matrix scan during the eighth Keypad Scan Interval.

If the two scan values are equal and detect at least one active key-press, the associated scan value(s) will be recorded off for reading.

If the two scan values are not equal, but at least one active key-press is detected, then the logic will wait seven Keypad Scan Interval periods, perform a Keypad Matrix scan during the eight Keypad Scan Interval, and continue repeating until either the associated scan value(s) are recorded off for reading or until no key-press activity is detected.

Offset 0x0034 – 0x0037

Keypad Direct Input Debounce Interval Register

This register provides the software with a means to configure the Keypad Direct Input Debounce times.

Bit	Attribute	Default	Description
31:12	RO	0	Reserved
11:0	RW	0	dir_dbn_reg[11:0] Keypad Direct Input Debounce Interval Bits. These bits indicate the number Keypad Scan Intervals between Keypad Direct Input scans that must match to detect stable active direct input values. If these bits are all zeros, then there is no Keypad Direct Input Debounce Interval time and the first Keypad Direct Input scan will be the only one necessary.

Note:
The Keypad Direct Input Debounce Interval time defines the number of Keypad Scan Interval times between Keypad Direct Input Scan Intervals that must match for an active key-press to be detected.

As an example, if the Keypad Direct Input Debounce Interval value is configured to be 0x005, once an active Keypad direct input scan has occurred, the logic will wait five Keypad Scan Interval periods, and then perform another Keypad direct input scan.

If the two scan values are equal and detect at least one active direct input, the associated scan value(s) will be recorded off for reading.

If the two scan values are not equal, but at least one active direct inputs is detected, the logic will wait five Key- pad Scan Interval periods, perform another Keypad direct input scan, and continue repeating until either the associated scan value(s) are recorded off for reading or until no direct input activity is detected.

Offset 0x0038 – 0xFFFF: Reserved

CIR Receiver Control Registers

WM8505 CIR function supports NEC/MATSUSHITA/SONY/JVC Infrared Code Detection, and the maximum received bit limitation is 32-bit.

Base address: 0xD827:0000

CIR Normal Mode Registers

Offset 0x000

CIR Software Reset

Bit	Attribute	Default	Description
31:0	RW	0	IR Software Reset. 0: No Reset. 1: Reset.

CIR Control Registers

Offset 0x004

CIR Control Register

Bit	Attribute	Default	Description
31:16	RO	0	Reserved
15:8	RW	0	Counter Mask of Irdata_stage_0 Update Threshold Control.
7:5	RO	0	Reserved
4	RW	0	To Detect NEC Subsequent Frame. 0: No detect NEC subsequent frame. 1: Detect NEC subsequent frame.
3	RW	0	CIR Input Polarity Control. 0: Normal, as input 1: Invert input signal
2:1	RW	0	CIR Remoter Vender Type 0h: NEC / JVC 1h: MATSUSHITA 2h: SONY 3h: Reserved
0	RW	0	CIR Enable. 0: Disable CIR 1: Enable CIR

CIR Status Registers

Offset 0x008

CIR INTRQ Status

Bit	Attribute	Default	Description
31:2	RO	0	Reserved
1	RO	0	NEC Repeat Flag (NEC Subsequent Frames). 0: New Code. 1: Repeat Code.
0	RO	0	CIR_INTRQ When CIR receives valid data, it will be asserted to 1. After CIR received data is read by CPU, this bit will auto clear to 0.

Offset 0x00C

CIR Received Data

Bit	Attribute	Default	Description
31:0	RO	0	CIR Received Data Once status of CIR_INTRQ is 1, CPU must read this register to clear CIR_INTRQ.

Offset 0x010: Reserved

Offset 0x014: Reserved

Offset 0x018: Reserved

CIR Receive Bit-Count Registers

Offset 0x01C

CIR Received Data Bit-Count

Bit	Attribute	Default	Description
31:21	RO	0	Reserved
20:16	RW	0	(SONY IR-Command Bit-Count) – 1 = 12 – 1 = 11
15:13	RO	0	Reserved
12:8	RW	0	(MATSUSHITA IR-Command Bit-Count) – 1 = 24 – 1 = 23
7:5	RO	0	Reserved
4:0	RW	0	(NEC IR-Command Bit-Count) – 1 = 32 – 1 = 31 (JVC IR-Command Bit-Count) – 1 = 16 – 1 = 15

CIR Parameter Registers

Offset 0x020

CIR Parameter Set 1

Bit	Attribute	Default	Description
31:26	RO	0	Reserved
25:0	RW	0	Sync-Pulse First Edge Cycle Count.

Offset 0x024

CIR Parameter Set 2

Bit	Attribute	Default	Description
31:26	RO	0	Reserved
25:0	RW	0	Sync-Pulse Second Edge Cycle Count.

Offset 0x028

CIR Parameter Set 3

Bit	Attribute	Default	Description
31:26	RO	0	Reserved
25:0	RW	0	Sync-Pulse Second Edge Cycle Count for NEC Subsequent Frame.

Offset 0x02C

CIR Parameter Set 4

Bit	Attribute	Default	Description
31:26	RO	0	Reserved
25:0	RW	0	Sync-Pulse Second Edge Cycle Count with Tolerance-cycles for NEC Subsequent Frame.

Offset 0x030

CIR Parameter Set 5

Bit	Attribute	Default	Description
31:26	RO	0	Reserved
25:0	RW	0	Measure-cycles of 1 T.

Offset 0x034

CIR Parameter Set 6

Bit	Attribute	Default	Description
31:26	RO	0	Reserved
25:0	RW	0	Measure-cycles of 2 T.

Offset 0x038

CIR Parameter Set 7

Bit	Attribute	Default	Description
31:26	RO	0	Reserved
25:0	RW	0	Tolerance-cycles of 2 T

CIR Continuous Command Registers

Offset 0x040

Continuous Command Control

Bit	Attribute	Default	Description
31:1	RO	0	Reserved
0	RW	0	Enable to Detect Continuous Command (with no Syn-Pulse, for JVC). 0: Disable Detect Continuous Command 1: Enable Detect Continuous Command

Offset 0x044

Continuous Command Period Count

Bit	Attribute	Default	Description
31:0	RW	0	Continuous Command Period (from data to data) Count.

Offset 0x048

Continuous Command Period Countdown Value

Bit	Attribute	Default	Description
31:0	RO	0	Continuous Command Period Countdown Value.

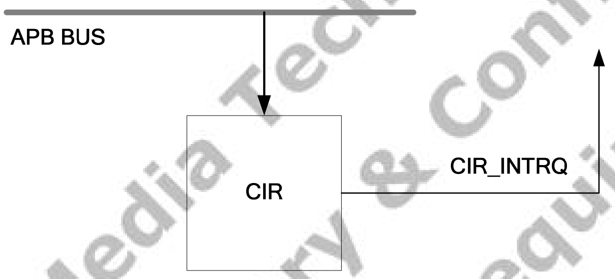


Figure 4 - WM8505 CIR Diagram

I^{2}C Control Registers

Base address of I^{2}C 0: 0xD828:0000
Base address of I^{2}C 1: 0xD832:0000

I^{2}C Master Register Group

Offset 0x0000

IICCR

IIC Controller Control Register

Bit	Attribute	Default	Description
15	RW	0	SLV_EN IIC Slave Enable. 0: The module default is act as an IIC master. 1: The module will act as an IIC slave.
14:4	RO	0	Reserved
3	RW	0	CPU_RDY After the Controller receives or transmits a data byte and Acknowledge bit, the SCL line is held low in the acknowledge-related SCL clock. After the CPU processes the data and acknowledge, this bit is set to release the SCL line HIGH.
2	RW	0	TX_END Bit[2] is used only when the Controller acts as a transmitter. In BYTE_END interrupt service routine, to end the current transfer, the software writes "1" to this bit. Then the hardware releases the SCL line, and generate a STOP condition automatically. Under this condition, the software need not set CPU_RDY bit.
1	RW	0	TX_ACK Bit[1] is used as programmable Acknowledge only when the Controller acts as a master-receiver. The software write this bit to set the Acknowledge bit sent in the Acknowledge cycle of the next data byte. To stop the current transfer, the software writes "1" to this bit, and then the transfer will be stopped after the next data byte transferred; otherwise, it writes "0" to this bit and the transfer will continue. 0: Acknowledge 1: Not acknowledge
0	RW	0	IICEN IIC Controller Enable. Set and cleared by software. When the module acts as slave, this bit will also works as a software reset bit. This bit can be set at the same time of SLV_EN (Bit[15]) selected. 0: Disabled 1: Enabled

Offset 0x0002

IICTCR

IIC Controller Transfer Control Register

Bit	Attribute	Default	Description
15	RW	0	MODE_SEL Fast Mode Select. This bit selects the transfer speed mode. 0: Standard mode (up to 100kbps). 1: Fast mode (up to 400 kbps). It can select the default frequency of SCL clock. In fast mode (MODE_SEL is 1), the default frequency is 400 kHz; in standard mode, it is 100 kHz.

14	RW	0	MASTER_RW Bit[14] indicates the direction of current transfer in which the Controller acts as a master. 0: Transmit 1: Receive
13	RW	0	HS_SEL High Speed Mode Select. This bit selects the high speed mode. 0: F/S mode (determined by MODE_SEL, bit 15) 1: High speed mode (3.4 MHz) When setting this bit, make sure that all the devices connected to the I^{2}C bus can run under high speed mode.
12:7	RO	0	Reserved
6:0	RW	00h	SSA6-SSA0 Seven Bits Slave Address. If it starts with 00001, which means high speed mode entered, change the clock frequency.

Note:
Each time the processor writes this register, a new transfer starts. To serve as a data-transmitter, write data register IICDR with the first data byte firstly.

Offset 0x0004

IICSR

IIC Controller Status Register

Bit	Attribute	Default	Description
15:2	RO	0	Reserved
1	RO	1b	READY IIC Controller Ready. 0: The IIC bus is busy. 1: The IIC bus is free. The controller clears this bit when detecting a START condition; it sets this bit when detecting a STOP condition or slave releases SCL line after SCL_TIMOUT interrupt is issued. If a repeated START condition is detected, this bit maintains 0. When SCL_TIMOUT interrupt is issued, this bit maintains 0 until the SCL line is released to HIGH. When READY is 0 in data-transmitting transfer, the software can also start a new transfer in BYTE_END interrupt service, and the Controller generates a repeated START to begin the new transfer after software writes 1 to CPU_RDY bit in IICCR to release the SCL line.
0	RO	0	RCV_ACK Received Acknowledge Bit. This bit is used when the Controller receives the Acknowledge from the external devices: 0: Acknowledgement was received. 1: Non-acknowledgement was received. This bit latches the bit on SDA line during the Acknowledge cycle. If the Controller receives a Not-acknowledge, the processor should not write new data byte in IICDR. Cleared by Software Read.

Offset 0x0006

IICISR

IIC Controller Interrupt Status Register

Bit	Attribute	Default	Description
15:3	RO	0	Reserved
2	RW1C	0	SCL_TIMOUT If the Slave has held the SCL low for too long a time (see SCLTP in IICTR), this bit will be set. It is an error condition.
1	RW1C	0	BYTE_END This bit is set during the Acknowledge cycle of data bytes. This interrupt lets software process the data byte and acknowledge bit after it serves the interrupt. This bit only works in the Data Transfer phase. When the processor serves the BYTE_END interrupt from the Controller, the hardware logic should hold the SCL line low to force the Slave addressed currently into Wait state; Data Transfer then continues after the CPU_RDY in IICCR is set, i.e. the Controller is ready for another byte of data and releases clock line SCL. One exception, when Acknowledge bit on the bus is 1 (Not-acknowledge), the hardware does not hold the SCL line to wait software processing and generates a STOP condition immediately.
0	RW1C	0	NACK_ADDR This bit is set when not Acknowledge is given for the Slave Address. At that time the Controller generates a STOP condition to end the transfer.

Offset 0x0008

IICIMR

IIC Controller Interrupt Mask Register

Bit	Attribute	Default	Description
15:3	RO	0	Reserved
2	RW	0	M_SCL_TIMOUT The Mask Bit of Interrupt Status Bit SCL_TIMOUT. 1: Enable SCL_TIMOUT interrupt.
1	RW	0	M_BYTE_END The Mask Bit of Interrupt Status Bit BYTE_END. 1: Enable BYTE_END interrupt.
0	RW	0	M_NACK_ADDR The Mask Bit of Interrupt Status Bit NACK. 1: Enable NACK_ADDR interrupt.

Offset 0x000A

IICDR

IIC Controller Data I/O Buffer Register

Bit	Attribute	Default	Description
15:8	RO	0	DATA2CPU When the Controller acts a master-receiver, the software reads data from the high byte of IICDR. In this version of design, data buffer is not supported.
7:0	RW	0	DATA2IIC When Controller acts as a master-transmitter, the software writes data to the low byte of IICDR. It can also be read for check. In this version of design, data buffer is not supported.

Offset 0x000C

IICTR

IIC Controller Time Parameters Register

Bit	Attribute	Default	Description
15:8	RW	80h	SCLTP SCL Line Holding Low Timeout Parameter. The SCLTP is in number of SCL clock half-cycle.
7:0	RW	19h	FSTP F/S Mode Time Parameter. Set the value of timing parameter CLK_HIGH. The FSTP is in number of system clock. Other parameters used in hardware logic can be calculated from CLK_HIGH. The default value is for the case of 100 kbps based on 20.4 MHz system clock. The hardware protects the setting value not beyond the reasonable scope. When read, it returns the real value used in hardware. FSTP = 64h when MODE_SEL is 0 (for standard mode). FSTP = 19h when MODE_SEL is 1 (for fast mode).

Offset 0x000E

IICMCR

Master Code in High Speed Mode.

Bit	Attribute	Default	Description
15:8	RW	00h	MST_CODE Master Code in High Speed Mode. It must be 00001XXX (b) if in high speed mode.
7:0	RO	0	Reserved

I^{2}C Slave Register Group

Offset 0x0010

IICSCR

I^{2}C Slave Control Register

Bit	Attribute	Default	Description
15	RO	0	Reserved
14	RW	0	HS_EN High-Speed Mode Enable. This bit enables the slave to work in high speed mode. 0: Only support F/S mode. 1: Support high speed mode (3.4 MHz). This bit specifies that the slave can work in High speed mode or not. When acting as slave, there is no difference in protocol between Fast mode and Standard mode.
13	RO	0	Reserved
12	RW	0	TX_ACK This bit is used as programmable Acknowledge only when the slave acts as a slave-receiver. The software writes this bit to set the Acknowledge bit sent in the next Acknowledge cycle. To stop the current transfer, the software writes "1" to this bit, then the transfer will be stopped after the next data byte transferred; otherwise, it writes "0" to this bit and the transfer will continue unless the master stop it. 0: Acknowledge. 1: Not acknowledge.
11:7	RO	0	Reserved

6:0	RW	00h	SLV_ADDR Bit[6:0] set the slave address of this module, which can be programmed by host CPU.

Offset 0x0012

IICSSR

I^{2}C Slave Status Register

Bit	Attribute	Default	Description
15:3	RO	0	Reserved
2	RO	0	SLV_ACT Slave Active Indication. If the address matches with the address set in this module, this bit will be set. It will be cleared when the Stop condition detected.
1	RO	0	R_W Read or Write Status in Slave Mode. 0: This module is written by external master. 1: This module is read by external master. It will be cleared when stop condition detected.
0	RO	0	RCV_ACK Received Acknowledge Bit. This bit is used when the Slave receives the Acknowledge from the external devices: 0: Acknowledge was received. 1: Not-acknowledge was received. This bit latches the bit on SDA line during the Acknowledge cycle. Since the data is requested by the DAT_REQ interrupt, the Not-acknowledge only shows the current transfer is going to stop. Bit[0] is cleared by the software Read.

Offset 0x0014

IICSISR

I^{2}C Slave Interrupt Status Register

Bit	Attribute	Default	Description
15:3	RO	0	Reserved
2	RW1C	0	SCL_TIMOUT When the data is not prepared or not read, I^{2}C slave will hold SCL low and wait Host CPU to response. If the time exceeds SCLTP value (see SCLTP in IICTR), this bit will be set. It is an error condition. The software writes 1 to clear this bit.
1	RW1C	0	BYTE_END This bit is set during the Acknowledge cycle of data bytes. This interrupt status is set when the I^{2}C slave acts as a slave receiver and a byte of data received. The software writes 1 to clear this bit.
0	RW1C	0	DAT_REQ This interrupt allows the software to write new data to be transferred in IICDR Bit[7:0] after it serves the interrupt. This bit only works in the slave-transmitter phase. Software writes 1 to clear this bit.

Offset 0x0016

IICSIMR

I²C Slave Interrupt Mask Register

Bit	Attribute	Default	Description
15:3	RO	0	Reserved
2	RW	0	M_SCL_TIMOUT The Mask Bit of Interrupt Status Bit SCL_TIMOUT. 1: Enable SCL_TIMOUT interrupt.
1	RW	0	M_BYTE_END The Mask Bit of Interrupt Status Bit BYTE_END. 1: Enable BYTE_END interrupt.
0	RW	0	M_DAT_REQ The Mask Bit of Interrupt Status Bit NACK. 1: Enable NACK_ADDR interrupt.

Offset 0x0018

IICSDR

I²C Slave Data I/O Buffer Register

Bit	Attribute	Default	Description
15:8	RO	0	DATA2CPU When I²C slave acts a slave-receiver, the software reads data from the high byte of IICDR. In this version of design, data buffer is not supported.
7:0	RW	0	DATA2IIC When I²C slave acts as a slave-transmitter, the software writes data to the low byte of IICDR. It can also be read for check. In this version of design, data buffer is not supported.

Offset 0x001A

IICSTR

I²C Slave Time Parameters Register

Bit	Attribute	Default	Description
15:0	RW	60h	SCLTP SCL Line Holding Low Timeout Parameter. SCLTP is in number of SCL clock cycle.

AC97 Control Registers

This section provides detailed definitions of the AC97 module APB accessible registers. Whenever necessary, a general rule of writing a one to clear a status bit is adopted for this module.

Base address: 0xD829:0000

Offset 0x00 – 0x03

ACCR

AC'97 Control Register

Bit	Attribute	Default1	Description
31:8	RO	0	Reserved.
7	RW	0	Double Rate Audio. Bit[7] enables Double-Rate Audio mode in which data from PCM L and PCM R in output slots 3 and 4 is used in conjunction with PCM L (n+1) and PCM R (n+1) data, to provide DAC streams at twice the sample rate designated by the PCM Front Sample Rate Control Register. DRA can be used without VRA; in that case the converter rates are forced to 96 kHz if DRA=1. The slots on which the (n+1) data is transmitted on is indicated by the DRSS[1:0] bits. In DRA mode, the programmed audio DAC sample rates are multiplied by 2x. For example in the 88.2-kHz DAC operation, the sample rate programmed would be 44.1 kHz, and the DRA bit will be programmed to 1. The DRA does not affect input ADC operation. 0: Double rate audio is disenabled. 1: Double rate audio is enabled.
6:5	RW	00b	Double Rate Slot Select. Bit[6:5] controls the slots that the n+1 data is present on for Double Rate Audio. 00: PCM L, R, C n+1 data is on Slots 10-12. 01: PCM L, R n+1 data is on slots 7, 8. 10: Reserved 11: Reserved
4	RW	0	CODEC Ready Interrupt Enable. Bit[4] indicates if an interrupt is asserted when a CODEC Ready indication is received by the AC'97 Controller. 0: Codec_ready_int_en is deasserted. 1: Codec_ready_int_en is asserted.
3	RW	0b	CODEC Read Done Interrupt Enable. Bit[3] indicates if an interrupt is asserted when a CODEC Read command is complete. 0: Codec_read_done_int_en is deasserted. 1: Codec_read_done_int_en is asserted.
2	RW	0b	CODEC Write Done Interrupt Enable. Bit[2] indicates if an interrupt is asserted when a CODEC Write command has been successfully sent to the CODEC. 0: Codec_write_done_int_en is deasserted. 1: Codec_write_done_int_en is asserted.

1	RW	0	AC97 Warm Reset. Setting this bit will assert ac97_sync, and clearing this bit will deassert ac97_sync during ACLINK power down mode2. This bit must remain asserted for a minimum of 1μs to guarantee a Warm Reset condition. The software routine should not set this bit within approximately 125 μs of initiating an ACLINK power down by writing to the CODEC Command Register (CCR).3. This bit does not accurately reflect the state of ac97_sync during ACLINK normal mode. 0: A "Warm Reset" is inactive or concluded (ac97_warm_reset is deasserted). 1: A "Warm Reset" is active (ac97_warm_reset is asserted).
0	RW	1b	AC97 Cold Reset. Setting this bit will assert ac97_rst_x4 . Clearing this bit will deassert ac97_rst_x4. Setting this bit will place all ac97_module logic in a reset state, including all register bits, with the exception of this bit. The software should guarantee the reset pulse width exceeds 1 μs. Asserting ac97_rst_x will reset both the CODEC and the ac97 module. This bit is reset on preset_x assertions only (the state of ac97_rst_x has no effect). 0: A "Cold Reset" is inactive or concluded (ac97_rst_x is deasserted). 1: A "Cold Reset" is active (ac97_rst_x is asserted).

Note:

1. All storage elements represented by the writable register bits (with the exception of ACCR[1:0]) are reset directly by ACR. ACCR[1:0]ACCR[0] is reset directly by preset_x.
2. Writing this bit while the ACLINK is in normal operational mode will almost certainly violate the ACLINK protocol and provide unpredictable operation of the AC97 module.
3. This bit setting has no effect on ac97_rst_x when test_mode is active (a one).

Offset 0x04 – 0x07

ACSR

AC97 Status Register

Bit	Attribute	Default	Description
31:8	RO	0	Reserved.
7	RO	0	CODEC Power Down. When CRDY is de-asserted (the CODEC is not ready), this bit indicates if the CODEC is in power down. If the CODEC is ready, this bit will always read zero. 0: The CODEC is not ready and is in reset, or the CODEC is ready (see CRDY). 1: The CODEC is not ready and the ACLINK is in power down mode.
6	RO	0	PCM Tx FIFO Interrupt Active. Bit[6] indicates if a PCM Tx FIFO interrupt is active. 0: pcm_tx_int is deasserted. 1: pcm_tx_int is asserted.
5	RO	0	PCM Rx FIFO Interrupt Active. Bit[5] indicates if a PCM Rx FIFO interrupt is active. 0: pcm_rx_int is deasserted. 1: pcm_rx_int is asserted.
4	RO	0	Mic FIFO Interrupt Active. Bit[4] indicates if a Mic FIFO interrupt is active. 0: mic_int is deasserted. 1: mic_int is asserted.

3	RW1C	0	CODEC Status Timeout. Bit[3] indicates if the most recent CODEC Read command completed due to a timeout error. If the Read command is complete due to a timeout error, the data in the CODEC Status Data Register (CSDR) is invalid. Writing a one will clear this bit. Writing a zero will have no effect. This bit is also cleared by writing to the CODEC Command Register (CCR). 0: The most recently completed CODEC read command completed normally. The CODEC Status Data Register contents are valid. 1: The most recently completed CODEC read command experienced a timeout error. The CODEC Status Data Register contents are invalid.
2	RW1C	0	CODEC Read Done. Bit[2] sets if the most recent CODEC read command is complete. The contents of the CODEC Status Valid bit (above) will indicate the validity of the CODEC Status Data Register (CSDR). Writing a one will clear this bit. Writing a zero will have no affect. This bit is also cleared by writing to the CODEC Command Register (CCR). 0: The most recent CODEC read command is not complete, or there is no pending CODEC read command. 1: The most recent CODEC read command is complete.
1	RW1C	0	CODEC Write Done. Bit[1] indicates if the most recent CODEC write command has been successfully sent over the AC97LINK. Writing a one will clear this bit. Writing a zero will have no effect. This bit is also cleared by writing to the CODEC Command Register (CCR). 0: The AC97 CODEC Write command is not complete, or there is no pending CODEC Write command. 1: The AC97 CODEC Write command is complete.
0	RO	0	CODEC Ready Bit[0] is set when Bit[16] of Input SLOT0 (TAG) is a one. This bit is cleared if Bit[16] of Input SLOT 0 (TAG) is a zero. This bit is also cleared if the ACLINK enters the power down state. 0: The CODEC is not in a ready state. 1: The CODEC is in a ready state.

Offset 0x08 – 0x0B

CSLR

CODEC Command Software Lock Bit

Bit	Attribute	Default	Description
31:1	R	0	Reserved.
0	RW1C	0	CODEC Software Lock Bit. Bit[0] will set on a Read. Writing a one will clear this bit. Writing a zero will have no effect. This bit is also cleared on the completion of a CODEC Read or Write command. This bit setting has no effect on the operation of the ACLINK or CODEC, and is intended for use as a software semaphore for access to CODEC registers. 0: No software routine is accessing the CODEC with a CODEC read or write command. 1: A software routine is currently executing a CODEC read or write command.

Offset 0x0C – 0x0F

CCR

CODEC Command Register

Bit	Attribute	Default	Description
31:16	RW	0	CODEC Write Command Data. This field contains the 16-bit data for the CODEC Command. Bit[31] is the MSB, and Bit[16] is the LSB.
15:8	RO	0	Reserved
7	RW	0	CODEC Command Read_Write. This bit indicates if the CODEC Command is a Read or Write. 0: The CODEC command is a write. 1: The CODEC command is a read.
6:0	RW	0	CODEC Command Address. This field contains the 7-bit register index for the CODEC Command. Bit[6] is the MSB, and Bit[0] is the LSB.

Offset 0x10 – 0x13

CSDR

CODEC Response Data Register

Bit	Attribute	Default	Description
31:16	RO	0h	Reserved
15:0	RO	0h	CODEC Status Data. Bit[15:0] is the 16-bit status data received from the CODEC in response to the most recent CODEC Read command.

Offset 0x20 – 0x23

PTCR

PCM Tx Control Register

Bit	Attribute	Default1	Description
31:14	RO	0	Reserved
13	RW	0	PCM Tx FIFO Mono Mode. Bit[13] indicates if the PCM Tx FIFO is in mono mode that the 16-bit data written to the FIFO will be duplicated for slots 3 & 4. 0: Stereo mode enabled. 1: Mono mode enabled.
12	RW	0	PCM Tx FIFO 8-bit Mode. Bit[12] indicates if the PCM Tx FIFO will operate in 8-bit mode that the data received from the ACLINK associated with the PCM Rx slots 3 & 4 will be zero padded to 16-bits and converted from unsigned to 2s complement when written to the PCM Tx FIFO AHB Write port. 0: 16-bit mode enabled. 1: 8-bit mode enabled.
11:8	RW	8h	PCM Tx FIFO Threshold. Bit[11:8] indicates the FIFO entry threshold for the PCM Tx FIFO. The “almost empty” condition occurs when the number of invalid (unoccupied) FIFO entries is equal to or greater than this threshold value. Any number from 4’h0 to 4’hF is valid. Bit[11] is the MSB, and Bit[8] is the LSB. 0000: Threshold = 16 0001: Threshold = 1 0010: Threshold = 2 ... 1111: Threshold = 15
7:6	RO	0	Reserved

5	RW	0	PCM Tx FIFO Overrun Interrupt Enable. Bit[5] indicates if an interrupt is asserted upon a PCM Tx FIFO overrun. 0: pcm_tx_over_int_en is deasserted. 1: pcm_tx_over_int_en is asserted.
4	RW	0	PCM Tx FIFO Underrun Interrupt Enable. Bit[4] indicates if an interrupt is asserted upon a PCM Tx FIFO underrun. 0: pcm_tx_under_int_en is deasserted. 1: pcm_tx_under_int_en is asserted.
3	RW	0	PCM Tx FIFO Almost Empty DMA Request Enable. Bit[3] indicates if a DMA request is asserted upon a PCM Tx FIFO almost empty condition. 0: pcm_tx_dma_en is deasserted. 1: pcm_tx_dma_en is asserted.
2	RW	0	PCM Tx FIFO Almost Empty Interrupt Enable Bit[2] indicates if an interrupt is asserted upon a PCM Tx FIFO almost empty condition. 0: pcm_tx_aempty_int_en is deasserted. 1: pcm_tx_aempty_int_en is asserted.
1	RW	0	PCM Tx FIFO Empty Interrupt Enable Bit[1] indicates if an interrupt is asserted upon a PCM Tx FIFO empty condition. 0: pcm_tx_empty_int_en is deasserted. 1: pcm_tx_empty_int_en is asserted.
0	RW	0	PCM Tx FIFO Enable Bit[0] indicates if the PCM Tx FIFO is enabled. 0: pcm_tx_en is deasserted. 1: pcm_tx_en is asserted.

Offset 0x24 – 0x27

PTSR

PCM Tx Status Register

Bit	Attribute	Default	Description
31:4	RO	0	Reserved
3	RW1C	0	PCM Tx FIFO Overrun Error. Bit[3] is set if the PCM Tx FIFO has experienced the overrun condition that occurs when the APB writes to the PCM Tx FIFO and the PCM Tx FIFO is full. Writing a one will clear this bit. Writing a zero has no effect. 0: No PCM Tx FIFO overrun condition. 1: PCM Tx FIFO overrun condition has occurred.
2	RW1C	0	PCM Tx FIFO Underrun Error. Bit[2] is set if the PCM Tx FIFO has experienced the under-run condition that occurs when the ACLINK requires data for transmitting slots 3 or 4 and the FIFO is empty. Writing a one will clear this bit. Writing a zero has no effect. 0: No PCM Tx FIFO under-run condition. 1: PCM Tx FIFO underrun condition has occurred.
1	RO	1b	PCM Tx FIFO Almost Empty. Bit[1] indicates if the number of PCM Tx FIFO invalid entries meets or exceeds the PCM Tx FIFO Threshold Value (PTFT). 0: pcm_tx_wr_aempty is deasserted. 1: pcm_tx_wr_aempty is asserted.

0	RO	1b	PCM Tx FIFO Empty. Bit[0] indicates if the PCM Tx FIFO is empty (contains no valid entries). 0: pcm_tx_wr_empty is deasserted. 1: pcm_tx_wr_empty is asserted.

Offset 0x28 – 0x2B

PRCR

PCM Rx Control Register

Bit	Attribute	Default	Description
31:14	RO	0	Reserved
13	RW	0	PCM Rx FIFO Mono Mode. Bit[13] indicates if the PCM Rx FIFO is in mono mode that only slot 3 data will be read from the FIFO. 0: Stereo mode enabled. 1: Mono mode enabled.
12	RW	0	PCM Rx FIFO 8-bit Mode. Bit[12] indicates if the PCM Rx FIFO will operate in 8-bit mode that the data received from the ACLINK associated with the PCM Rx slots 3 & 4 will be truncated to 8-bits and converted from 2s complement to unsigned when Read from the PCM Rx FIFO AHB Read port. 0: 16-bit mode enabled. 1: 8-bit mode enabled.
11:8	RW	8h	PCM Rx FIFO Threshold. Bit[11:8] indicates the FIFO entry threshold for the PCM Rx FIFO. The "almost full" condition occurs when the number of valid (occupied) PCM Rx FIFO entries is equal to or greater than this value. Any number from 4'h0 to 4'hF is valid. Bit[11] is the MSB, and Bit[8] is the LSB. 0000: Threshold = 16 0001: Threshold = 1 0010: Threshold = 2 ... 1111: Threshold = 15
7:6	RO	0	Reserved
5	RW	0	PCM Rx FIFO Overrun Interrupt Enable. Bit[5] indicates if an interrupt is asserted upon a PCM Rx FIFO overrun condition. 0: pcm_rx_over_int_en is deasserted. 1: pcm_rx_over_int_en is asserted.
4	RW	0	PCM Rx FIFO Underrun Interrupt Enable. Bit[4] indicates if an interrupt is asserted upon a PCM Rx FIFO underrun condition. 0: pcm_rx_under_int_en is deasserted. 1: pcm_rx_under_int_en is asserted.
3	RW	0	PCM Rx FIFO Almost Full DMA Request Enable. Bit[3] indicates if a DMA request is asserted upon a PCM Rx FIFO almost-full condition. 0: pcm_rx_dma_en is deasserted. 1: pcm_rx_dma_en is asserted.
2	RW	0	PCM Rx FIFO Almost Full Interrupt Enable. Bit[2] indicates if an interrupt is asserted upon a PCM Rx FIFO almost-full condition. 0: pcm_rx_afull_int_en is deasserted. 1: pcm_rx_afull_int_en is asserted.
1	RW	0	PCM Rx FIFO Full Interrupt Enable. Bit[1] indicates if an interrupt is asserted upon a PCM Rx FIFO full condition. 0: pcm_rx_full_int_en is deasserted. 1: pcm_rx_full_int_en is asserted.

0	RW	0	PCM Rx FIFO Enable. Bit[0] indicates if the PCM Rx FIFO is enabled. 0: pcm_rx_en is deasserted. 1: pcm_rx_en is asserted.

Offset 0x2C – 0x2F

PRSR

PCM Rx FIFO Status Register

Bit	Attribute	Default	Description
31:4	RO	0	Reserved
3	RW1C	0	PCM Rx FIFO Overrun Error. Bit[3] indicates the PCM Rx FIFO has experienced an overrun condition that occurs when the ACLINK receives data from Receive slots 3 or 4 and the FIFO is full. Writing a one will clear this bit. Writing a zero has no effect. 0: No PCM Rx FIFO overrun condition. 1: PCM Rx FIFO overrun error has occurred.
2	RW1C	0	PCM Rx FIFO Underrun Error. Bit[2] indicates the PCM Rx FIFO has experienced an underrun condition that occurs when the APB reads from the PCM RX FIFO Read port and the FIFO is empty. Writing a one will clear this bit. Writing a zero has no effect. 0: No PCM Rx FIFO under-run condition. 1: PCM Rx FIFO underrun error has occurred.
1	RO	0	PCM Rx FIFO Almost Full. Bit[1] indicates if the number of PCM Rx FIFO valid entries meets or exceeds the PCM Rx FIFO Threshold (PRFT). 0: pcm_rx_rd_afull is deasserted. 1: pcm_rx_rd_afull is asserted.
0	RO	0	PCM Rx FIFO Almost Full. Bit[0] indicates if the number of PCM Rx FIFO valid entries meets or exceeds the PCM Rx FIFO Threshold (PRFT). 0: pcm_rx_rd_afull is deasserted. 1: pcm_rx_rd_afull is asserted.

Offset 0x30 – 0x33

MCR

Microphone (Mic) FIFO Control Register

Bit	Attribute	Default	Description
31:13	RO	0	Reserved
12	RW	0	MIC FIFO 8-bit Mode. Bit[12] indicates if the Microphone FIFO will operate in 8-bit mode that the data received from the ACLINK associated with the Microphone input will be truncated to 8 bits and converted from 2s complement to unsigned when reading from the MIC FIFO AHB Read port. 0: 16-bit mode enabled. 1: 8-bit mode enabled
11:8	RW	8h	MIC FIFO Threshold. Bit[11:8] indicates the FIFO entry threshold for the Mic FIFO. An “almost full” condition occurs if the number of valid Mic FIFO entries is equal to or greater than this value. Any number from 4’h0 to 4’hF is valid. Bit[11] is the MSB, and Bit[8] is the LSB. 0000: Threshold = 16 0001: Threshold = 1 0010: Threshold = 2 ... 1111: Threshold = 15
7:6	RO	0	Reserved

5	RW	0	MIC FIFO Overrun Interrupt Enable. Bit[5] indicates if an interrupt is asserted upon a Mic FIFO overrun condition. 0: mic_over_int_en is deasserted. 1: mic_over_int_en is asserted.
4	RW	0	MIC FIFO Under-run Interrupt Enable. Bit[4] indicates if an interrupt is asserted upon a Mic FIFO underrun condition. 0: mic_under_int_en is deasserted. 1: mic_under_int_en is asserted.
3	RW	0	MIC Almost Full DMA Request Enable. Bit[3] indicates if a DMA request is asserted upon a Mic FIFO almost-full condition. 0: mic_dma_en is deasserted. 1: mic_dma_en is asserted.
2	RW	0	MIC FIFO Almost Full Interrupt Enable. Bit[2] indicates if an interrupt is asserted upon a Mic FIFO almost-full condition. 0: mic_afull_int_en is deasserted. 1: mic_afull_int_en is asserted.
1	RW	0b	Mic FIFO Full Interrupt Enable. Bit[1] indicates if an interrupt is asserted upon a Mic FIFO full condition. 0: mic_full_int_en is deasserted. 1: mic_full_int_en is asserted.
0	RW	0b	Mic FIFO Enable. This bit indicates if the Mic FIFO is enabled. 0: mic_fifo_en is deasserted. 1: mic_fifo_en is asserted.

Offset 0x34 – 0x37

MSR

Microphone (Mic) FIFO Status Register

Bit	Attribute	Default	Description
31:4	RO	0	Reserved
3	RW1C	0	MIC FIFO Overrun Error. Bit[3] indicates the MIC FIFO has experienced and overrun condition that occurs when the ACLINK receives data from Receive slot 6 and the FIFO is full. Writing a one will clear this bit. Writing a zero has no effect. 0: No MIC FIFO overrun condition. 1: MIC FIFO overrun error has occurred.
2	RW1C	0	MIC FIFO Under-run Error. Bit[2] indicates the MIC FIFO has experienced and under-run condition that occurs when the APB reads from the Mic FIFO Read port and the Mic FIFO is empty. Writing a one will clear this bit. Writing a zero has no effect. 0: No MIC FIFO under-run condition. 1: MIC FIFO Under-run Error has occurred.
1	RO	0b	Mic FIFO Almost Full. Bit[1] indicates if the number of PCM Rx FIFO valid entries meets or exceeds the Mic FIFO Threshold Value (MFT). 0: mic_rd_afull is deasserted. 1: mic_rd_afull is asserted.
0	RO	0b	Mic FIFO Full. Bit[0] indicates the MIC FIFO is full. 0: mic_rd_full is deasserted. 1: mic_rd_full is asserted.

Offset 0x80 – 0xBF **Attribute: WO**

PTFIFO

PCM Stereo Tx FIFO

The PTFIFO is the Write port of the PCM Tx FIFO.

The PTFIFO will capture data and increment the Write pointer as detailed in the Table "PCM Tx FIFO Write Port.

All APB reads will respond with zeros.

For reference, the following tables show a programmers model of the PCM Tx FIFO in the different modes.

Table 7 - PCM Tx FIFO Write Port (PTFIFO) Valid Access

PTF8M	PTFMM	Valid Access
0	0	32-bit Writes Only (STR, 32-bit DMA)
0	1	16-bit Writes Only (STRH, 16-bit DMA)
1	0	16-bit Writes Only (STRH, 16-bit DMA)
1	1	8-bit Writes Only (STRB, 8-bit DMA)

Table 8 - PCM Tx FIFO Write Port (PTFIFO)

PTF8M	PTFMM	apb_wr_en[3:0]	pcm_tx_wr_inc	pcm_tx_wr_data
0	0	1111b	1	pwdata[31:0]
0	0	others	0	0000 0000h
0	1	0011b	1	{pwdata[15:0], pwdata[15:0]}
0	1	1100b	1	{pwdata[31:16], pwdata[31:16]}
0	1	others	0	00000000h
1	0	0011b	1	{~pwdata[15], pwdata[14:8], 00h, ~pwdata[7], pwdata[6:0], 00h}
1	0	1100b	1	{~pwdata[31], pwdata[30:24], 00h, ~pwdata[23], pwdata[22:16], 00h}
1	0	others	0	00000000h
1	1	0001b	1	{2{~pwdata[7], pwdata[6:0], 00h}}
1	1	0010b	1	{2{~pwdata[15], pwdata[14:8], 00h}}
1	1	0100b	1	{2{~pwdata[23], pwdata[22:16], 00h}}
1	1	1000b	1	{2{~pwdata[31], pwdata[30:24], 00h}}
1	1	others	0	00000000h

PCM Tx FIFO Write Port (PTFIFO) 16-bit Stereo Mode

Bit	Attribute	Default	Description
31:16	WO	0	Tx Left Channel. 16-bit 2s-complement Audio Sample. MSB = 31, LSB = 16
15:0	WO	0	Tx Right Channel. 16-bit 2s-complement Audio Sample. MSB = 15, LSB = 0

PCM Tx FIFO Write Port (PTFIFO) 16-bit Mono Mode

Bit	Attribute	Default	Description
15:0	WO	0	Tx 16-bit Mono Channel. 16-bit 2s-complement Audio Sample. MSB = 15, LSB = 0

PCM Tx FIFO Write Port (PTFIFO) 8-bit Stereo Mode

Bit	Attribute	Default	Description
15:8	WO	0	Tx Left Channel. 8-bit 2s-complement Audio Sample. MSB = 15, LSB = 8
7:0	WO	0	Tx Right Channel. 8-bit 2s-complement Audio Sample. MSB = 7, LSB = 0

PCM Tx FIFO Write Port (PTFIFO) 8-bit Mono Mode

Bit	Attribute	Default	Description
7:0	WO	0	Tx 8-bit Mono Channel. 8-bit Unsigned Audio Sample. MSB = 7, LSB = 0

Offset 0xC0 – 0xFF **Attribute: RO**

PRFIFO

PCM Rx FIFO Read Port

The PRFIFO is the Read port of the PCM Rx FIFO.

The PRFIFO will drive data to the AHB bus and increment the Read pointer as detailed in the following "PCM Rx FIFO Read Port" Table.

All APB writes will be ignored.

For reference, following tables show a programmers model of the PCM Rx FIFO in the different modes.

Table 9 - PCM Rx FIFO Read Port (PRFIFO) Valid Access

PRF8M	PRFMM	Valid Access
0	0	32-bit Reads Only (LD, 32-bit DMA)
0	1	16-bit Reads Only (LDH, 16-bit DMA)
1	0	16-bit Reads Only (LDH, 16-bit DMA)
1	1	8-bit Reads Only (LDB, 8-bit DMA)

Table 10 - PCM Rx FIFO Read Port (PRFIFO)

PTF8M	PTFMM	psize[1:0]	pcm_rx_rd_inc	next_prdata
0	0	10b	1	pcm_rx_rd_data[31:0]
0	0	others	0	00000000h
0	1	01b	1	{2{pcm_rx_rd_data[31:15]}}
0	1	others	0	00000000h
1	0	01b	1	{2{~pcm_rx_rd_data[31], pcm_rx_rd_data[30:24], ~pcm_rx_rd_data[15], pcm_rx_rd_data[14:8]}}
1	0	others	0	00000000h
1	1	00b	1	{4{~pcm_rx_rd_data[31], pcm_rx_rd_data[30:24]}}
1	1	others	0	00000000h

PCM Rx FIFO Read Port (PRFIFO) 16-bit Stereo Mode

Bit	Attribute	Default	Description
31:16	RO	0h	Rx Left Channel. 16-bit 2s-complement Audio Sample. MSB = 31, LSB = 16
15:0	RO	0h	Rx Right Channel. 16-bit 2s-complement Audio Sample. MSB = 15, LSB = 0

PCM Tx FIFO Write Port (PTFIFO) 8-bit Mono Mode

Bit	Attribute	Default	Description
15:0	RO	0h	Rx 16-bit Mono Channel. 16-bit 2s-complement Audio Sample. MSB = 15, LSB = 0

PCM Rx FIFO Read Port (PRFIFO) 8-bit Stereo Mode

Bit	Attribute	Default	Description
15:8	RO	0h	Rx Left Channel. 8-bit Unsigned Audio Sample. MSB = 15, LSB = 8
7:0	RO	0h	Rx Right Channel. 8-bit Unsigned Audio Sample. MSB = 7, LSB = 0

PCM Rx FIFO Read Port (PRFIFO) 8-bit Mono Mode

Bit	Attribute	Default	Description
7:0	RO	0h	Rx 8-bit Mono Channel 8-bit Unsigned Audio Sample. MSB = 7, LSB = 0

Offset 0x100 – 0x13F **Attribute: RO**

MFIFO

Microphone (Mic) FIFO Read Port

The MFIFO is the Read port of the Mic FIFO.

The MFIFO will drive data on the AHB bus and increment the Read pointer as detailed in the following “Mic FIFO Read Port” Table.

All APB Writes will be ignored.

For reference, the following tables show a programmers model of the Mic FIFO in the different modes.

Table 11 - Mic FIFO Read Port (MFIFO) Valid Access

MF8M	Valid Access
0	16-bit Reads Only (LDH, 16-bit DMA)
1	8-bit Reads Only (LDB, 8-bit DMA)

Table 12 - Mic FIFO Read Port (MFIFO)

MF8M	psize[1:0]	mic_rd_inc	next_prdata
0	01b	1	{2{mic_rd_data[15:0]}}
0	others	0	0000 0000h
1	00b	1	{4{~mic_rd_data[15], mic_rd_data[14:8]}}
1	others	0	0000 0000h

MIC FIFO Read Port (MFIFO) 16-bit Mode

Bit	Attribute	Default	Description
15:0	RO	0h	16-bit Mono Channel. 16-bit 2s-complement Audio Sample. MSB = 15, LSB = 0

MIC FIFO Read Port (MFIFO) 8-bit Mode

Bit	Attribute	Default	Description
7:0	RO	0h	8-bit Mono Channel. 8-bit Unsigned Audio Sample. MSB = 7, LSB = 0

I^2S Control Registers

Base address: 0xD833:0000

Offset 0x00 - 0x03

AUDCTLCR

The AUDCTLCR provides control over the Audio Controller module.

Bit	Attribute	Default	Description
31:30	RO	0	Reserved
29	RW	1b	rx_rchn Record Right Channel Enable. 0: Right channel is disabled for record. 1: Audio data for right channel is enabled to receive.
28	RW	1b	rx_lchn Record Left Channel Enable. 0: Left channel is disabled for record. 1: Audio data for left channel is enabled to receive.
27	RW	0	rxsync_off RxSYNC_out Disable Bit (Only Used in Master Mode). 0: RxSYNC_out normal. 1: RxSYNC_out stopped and cannot go to high level while RxBCLK_out continues, causing external device to power down.
26:24	RW	000b	rxsample_bit Sample Bit for Received Data As Record. 000: 8 bit 001: 13 bit 010: 14 bit 011: 16 bit 100: 20 bit 101: 24 bit 110: 32 bit 111: Reserved
23	RW	0	rxbclk_inv Bit Clock Polarity Select for Received Data as Record. 0: Normal polarity. Output signals are on the falling edge of RxBCLK, and sample input data is on the rising edge of RxBCLK. 1: Inverse the normal polarity
22	RO	0	Reserved
21:20	RW	00b	rxdsp_mod Rx DSP Mode. It is only used when Bit[19:18] (rxaud_mode) is 2'b11. The detailed DSP mode of recording input data is defined here. 00: Short frame sync, and early mode. 01: Short frame sync, and late mode. 10: Long frame sync, and early mode. 11: Long frame sync, and late mode.
19:18	RW	00b	rxaud_mod Rx Audio Data Format. Select digital audio interface mode for received data as record. 00: I^2S mode 01: Right-justify mode 10: Left-justify mode 11: DSP mode
17	RW	0	rxslave Master/Slave Mode Bit for Received Data as Record. RxBCLK_in, RxSYNC_in are just active in the slave mode. RxBCLK_out, RxSYNC_out are just active in the master mode. 0: Rx in master mode 1: Rx in slave mode

16	RW	0	mclk_inv Select Master Clock Polarity for TXMCLK and RXMCLK. 0: The edge of Frame Sync Clock(txsync_out or rxsync_out) is synchronized with the falling edge of master clock(txmclk or rxmclk). 1: The edge of Frame Sync Clock(txsync_out or rxsync_out) is synchronized with the rising edge of master clock(txmclk or rxmclk).
15:14	RO	0	Reserved
13	RW	1b	tx_rchn Playback Right Channel Enable. 0: Right channel is disabled for playback. 1: Audio data for right channel is enabled to transmit.
12	RW	1b	tx_lchn Playback Left Channel Enable. 0: Left channel is disabled for playback. 1: Audio data for left channel is enabled to transmit.
11	RW	0	txsync_off TxSYNC_out Disable Bit (Only Used in Master Mode). 0: TxSYNC_out normal. 1: TxSYNC_out stopped and cannot go to high level while TxBCLK_out continues, causing external device to power down.
10:8	RW	000b	txsample_bit Sample Bit for Transmitted Data as Playback 000: 8 bit 100: 20 bit 001: 13 bit 101: 24 bit 010: 14 bit 110: 32 bit 011: 16 bit 111: Reserved
7	RW	0	txbclk_inv Select Bit Clock Polarity for the Transmitted Data as Playback. 0: Normal polarity. The output data is on the falling edge of TxBCLK, and the sample input data on the rising edge of TxBCLK. 1: Inverse the normal polarity.
6	RO	0	Reserved
5:4	RW	0	txdsp_mod Tx DSP Mode. It is only used when Bit[3:2](txaud_mode) is 2'b11. The detailed DSP mode of playback output data is defined here. 00: Short frame sync, and early mode. 01: Short frame sync, and late mode. 10: Long frame sync, and early mode. 11: Long frame sync, and late mode.
3:2	RW	0	txaud_mod Tx Audio Data Format. Select the digital audio interface mode for the transmitted data as playback. 00: I^{2}S mode 01: Right-justify mode (RJ) 10: Left-justify mode (LJ) 11: DSP mode
1	RW	0	txslave Master/Slave Mode Bit for Transmitted Data as Playback. TxBCLK_in and TxSYNC_in are just active in slave mode. TxBCLK_out and TxSYNC_out are just active in master mode. 0: Tx in master mode 1: Tx in slave mode

0	RW	0	aud_en I2S_v2 Enable Bit. Enable to transfer audio data between the Audio Controller and external audio device. This bit is also used to enable clock. divisors. 0: Disable 1: Enable

Offset 0x04 – 0x07

AUDDFCR

The AUDDFCR provides control over the audio data format

Bit	Attribute	Default	Description
31:15	RO	0	Reserved
14	RW	0	rx_msb RX FIFO First Bit Setup Bit. Bit[14] identifies the order in which the valid data is received from the ADCDIN port. This control bit is NOT affected by the programming of Bit[10]. 0: MSB bit first 1: LSB bit first
13	RW	0	rx_pad RX FIFO Padding Control Bit. Bit[13] controls the no-effect bits of right-aligned data written into the Receive FIFO from the valid data of ADCDIN. The left-aligned data is always zero padded. This bit is ignored if Bit[10] is zero. 0: No-effect bits are zero filled. 1: No-effect bits are sign-extension of MSB bit of valid data.
12	RW	0	rd_al RX FIFO Alignment Control Bit. Bit[12] controls the alignment of data written into the Receive FIFO from the data of the ADCDIN. This bit is ignored if Bit[10] is zero. 0: Right aligned within RX FIFO. 1: Left aligned within RX FIFO.
11	RW	0	rx_al ADCDIN Data Alignment Setup Bit. This bit identifies the left or right alignment of data being received in from the ADCDIN port so the data can be properly processed. This bit is ignored if Bit[10] is zero. 0: Right aligned within received data. 1: Left aligned within received data.
10	RW	0	rx_size Data Size in ADCDIN per Channel. 0: Same with sample bit(Register sample_bit). 1: When Rx sample bit is 8, 16 bits are received from ADCDIN. When Rx sample bit is 13, 16 bits are received from ADCDIN. When Rx sample bit is 14, 16 bits are received from ADCDIN. When Rx sample bit is 16, 32 bits are received from ADCDIN. When Rx sample bit is 20, 32 bits are received from ADCDIN. When Rx sample bit is 24, 32 bits are received from ADCDIN. When Rx sample bit is 32, 32 bits are received from ADCDIN.
9:7	RO	0	Reserved
6	RW	0	tx_msb Select DACDOUT First Bit. Bit[6] controls the order in which the data is transmitted. This control bit is NOT affected by the programming of Bit[2]. 0: MSB bit first 1: LSB bit first

5	RW	0	tx_pad DACDOUT First Bit Select Bit. Bit[5] controls the order in which data is transmitted. This control bit is NOT affected by the programming of Bit 2. 0: MSB bit first 1: LSB bit first
4	RW	0	tx_al DACDOUT First Bit Select Bit. Bit[4] controls the order in which the data is transmitted. This control bit is NOT affected by the programming of Bit 2. 0: MSB bit first 1: LSB bit first
3	RW	0	wr_al TX Write Data Alignment Setup Bit. Bit[3] identifies the left or right alignment of data on APB data bus that will be written into the TX FIFO so the data can be properly processed. This bit is ignored if Bit[2] is zero. 0: Right aligned on APB data bus bit[31:16] or bit[15:0] 1: Left aligned on APB data bus bit[31:16] or bit[15:0]
2	RW	0	tx_size Data Size in DACDOUT per Channel. 0: Same with sample bit (Register sample_bit) 1: When tx sample bit is 8, 16 bit transmitted to DACDOUT. When Tx sample bit is 13, 16 bit transmitted to DACDOUT. When Tx sample bit is 14, 16 bit transmitted to DACDOUT When Tx sample bit is 16, 32 bit transmitted to DACDOUT When Tx sample bit is 20, 32 bit transmitted to DACDOUT When Tx sample bit is 24, 32 bit transmitted to DACDOUT When Tx sample bit is 32, 32 bit transmitted to DACDOUT
1:0	RO	0	Reserved

Note:

Table 13 - Data Size of Playback Data Written from MCU or APB Interface for Each Channel

tx_size	**wr_al**	**Data Size of Playback**
0	x	defined by txsample_bit
1	0	defined by txsample_bit
1	1	data size transmitted to DACDOUT (as description in tx_size)

Table 14 - Data Size of Record Data Read by MCU or APB Interface for Each Channel

rx_size	**{rd_al,rx_pad}**	**Data Size of Record**
0	x	defined by rxsample_bit
1	2'b00	defined by rxsample_bit
1	others	data size received from ADCDIN (as description in rx_size)

Offset 0x08 – 0x0B

TXCLKDIV

The TXCLKDIV provides clock divisor control of the bit clock and master clock for the transmitted data.

Bit	Attribute	Default	Description
31:24	RO	0	Reserved
23:16	RW	0	txmclk_div TXMCLK Division Value. This value is used to divide SYSCLK[4] to the external master clock (TXMCLK[5]) output for playback according to the following equations. txmclk_div=Integer[freq(SYSCLK[4])/freq(TXMCLK)+1/2] freq(TXMCLK[5]) = freq(SYSCLK[4])/txmclk_div Note: (1) The value [0,1] is the same as the value 1, TXMCLK = SYSCLK[4]. (2) The software does not need to program this value if TXMCLK is not used. (3) TXMCLK will be output once the module is enabled, no matter it is in master mode or slave mode. (4) In MCU system, SYSCLK means sysclk_ply. (5) In SOC system, SYSCLK means audsysclk. In SOC system, TXMCLK means master clk.
15:11	RO	0	Reserved
10:0	RW	0	txbclk_div TXBCLK Division Value. This value is used to divide SYSCLK[2] to the bit clock (TXBCLK[3]) for playing according to the following equations. txbclk_div = Integer[freq(SYSCLK[2])/freq(TXBCLK) + 1/2] freq(TXBCLK[3])=freq(SYSCLK[2])/txbclk_div It has following 2 functions: – Generating TXBCLK in master mode. – Controlling the DACDOUT shift out time to meet its output delay timing. Note: (1) The value [0,1] is the same as the value 1, TXBCLK = ~SYSCLK[2]. (2) In MCU system, SYSCLK means sysclk_ply. In SOC system, SYSCLK means audsysclk. (3) In SOC system, TXBCLK means tx bit clk.

Offset 0x0C – 0x0F

TXFRAMECR

The TXFRAMECR provides control over the offset bit slot of the data beginning per tx frame, and also defines the TxSYNC frequency.

Bit	Attribute	Default	Description
31:24	RO	0	Reserved
23:16	RW	0	offset_txbit Offset Bit Slot of the Data Beginning each Tx Frame. It defines the offset bit slot number of the data beginning on the DACDOUT port. The default value is 0, which means the offset is 0 and the data beginning in the normal mode. Note: This bit is ignored if txaud_mod = 2’b01(Right_justify mode).
15:2	RO	0	Reserved

11:0	RW	040h	txsync_div TxSYNC Clock Division Value. This value is used to divide TXBCLK to the DAC sync clock (TxSYNC) according to following equations. txsync_div = Integer[freq(TXBCLK)/freq(TxSYNC) + 1/2] freq(TxSYNC)=freq(TXBCLK)/txsync_div It has the following two functions: – Generating TxSYNC_out in master mode. – Controlling the frequency of DAC sample rate. Note (1) The value [0,1] is the same with the value 2. (2) The default value is 64. (3) The software must correctly program this value no matter in master mode or slave mode. If in slave mode, the software should use the input TXBCLK and TxSYNC frequency to count this value. (4) The minimum value of this register is the tx_size*channel number. It is error if the configured value is smaller than the min. value. (5) In WM8505, the bit width of this register is only 8-bit.

Offset 0x10 – 0x13

RXCLKDIV

The RXCLKDIV provides clock divisor control over the bit clock and master clock for received data, just used for recording.

Bit	Attribute	Default	Description
31:24	RO	0	Reserved
23:16	RW	0	rxmclk_div RXMCLK Division Value. This value is used to divide SYSCLK[4] to the external master clock (RXMCLK) output for record according to following equations. txmclk_div=Integer[freq(SYSCLK[4])/freq(RXMCLK)+1/2] freq(RXMCLK) = freq(SYSCLK[4])/rxmclk_div Note: (1) The value 0,1 is the same as the value 1, RXMCLK = SYSCLK[4]. (2) The software does not need to program this value if RXMCLK is not used. (3) RXMCLK will be output once the module is enabled no matter it is in master mode or slave mode. (4) In MCU system, SYSCLK means sysclk_rec. In SOC system, SYSCLK means audsysclk.
15:11	RO	0	Reserved
10:0	RW	0	rxbclk_div RXBCLK Division Value. This value is used to divide SYSCLK[2]to the bit clock (RXBCLK) for recording according to following equations. rxbclk_div = Integer[freq(SYSCLK[2])/freq(RXBCLK) + 1/2] freq(RXBCLK)=freq(SYSCLK[2])/rxbclk_div It has the following two functions: – Generating RXBCLK in master mode. – Controlling the DACDOUT shift out time to meet its output delay timing. Note: (1) The value 0,1 is the same as the value 1, RXBCLK = ~ SYSCLK[2]. (2) In MCU system, SYSCLK means sysclk_rec. In SOC system, SYSCLK means audsysclk.

Offset 0x14 – 0x17

RxFRAMECR

The RxFRAMECR provides the control over the offset bit slot of the data beginning per rx frame, and also defines the RxSYNC frequency.

Bit	Attribute	Default	Description
31:24	RO	0	Reserved
23:16	RW	0	offset_rxbit Offset Bit Slot of the Data Beginning Each Rx Frame. It defines the offset bit slot number of the data beginning on the ADCDIN port. The default value is 0, which means the offset is 0 and the data beginning in normal mode. Note: This bit is ignored if rxaud_mod = 2'b01 (Right_justify mode).
15:12	RO	0	Reserved
11:0	RW	040h	rxsync_div RxSYNC Clock Division Value. This value is used to divide RXBCLK to the ADC sync clock (RxSYNC) according to the following equations. rxsync_div = Integer[freq(RXBCLK)/freq(RxSYNC) + 1/2] freq(RxSYNC)=freq(RXBCLK)/rxsync_div It has the following two functions: – Generating RxSYNC_out in master mode. – Controlling the frequency of ADC sample rate. Note: (1) The value [0, 1] is the same with the value 2. (2) The default value is 64. (3) The software must correctly program this value no matter in master mode or slave mode. If in slave mode, the software should use the input BCLK and RxSYNC frequency to count this value. (4) The minimum value of this register is the rx_size*channel number. It is error if the configured value is smaller than the min. value. (5) In WM8505, the bit width of this register is only 8-bit.

Offset 0x18 – 0x1B

TXEQURXCR

Bit	Attribute	Default	Description
31:8	RO	0	Reserved
7	RW	0	rxfifo_sw_flush RxFIFO Software Flush. When i2s_v2 stops receiving recording data from the digital audio interface (determined by the value of register rx_en and rx_stop_immed), RxFIFO will has a flush if this bit is 1. After this flush is completed, it can read 1 from rx_rd_empty (0x2C[4]). Then, it can write 0 to this bit to stop this flush. 0: End software flush for RxFIFO. 1: Start software flush for RxFIFO. Note: This bit is ignored if rxfifo_flush_en=0.
6:1	RW	0	Reserved

0	RW	0	txequrxcr Tx and Rx have the Same Configuration. Tx and Rx part of the I^{2}S bus have the same audio mode, clock divisor and offset. 1: When AUDCTLCR[15:0], TXCLKDIV and TXFRAMECR are configured, the same value is configured to AUDCTLCR[31:16], RXCLKDIV and RXFRAMECR, i.e. Tx part and Rx part has the same audio mode, clock divisor and offset.

Offset 0x20 – 0x23

TCR

The TCR controls the Tx FIFO.

Bit	Attribute	Default	Description
31:12	RO	0	Reserved
11:8	RW	8h	tx_wr_thresh Tx FIFO Threshold. Bit[11:8] indicate the FIFO entry threshold for the Tx FIFO. The "almost empty" condition occurs when the number of invalid (unoccupied) FIFO entries is equal to or greater than this value. Any number from 4'h00 to 4'hF is valid. Bit[11] is the MSB, and Bit[8] is the LSB. 0000: Threshold = 16*32-bit 0001: Threshold = 1*32-bit 0010: Threshold = 2*32-bit ... 1111: Threshold = 15*32-bit Note: This register only for SOC system. In MCU system, it is a reserved register and read only 4'b00000.
7:6	RO	0	Reserved
5	RW	0	tx_over_int_en Tx FIFO Overrun Interrupt Enable. Bit[5] indicates if an interrupt is generated upon a PCM Tx FIFO overrun. 0: i2s_tx_over_int_en is deasserted. 1: i2s_tx_over_int_en is asserted.
4	RW	0	tx_under_int_en Tx FIFO Under-run Interrupt Enable. Bit[4] indicates if an interrupt is generated upon a PCM Tx FIFO underrun. 0: i2s_tx_under_int_en is deasserted. 1: i2s_tx_under_int_en is asserted.
3	RW	0	tx_dma_en Tx FIFO Almost Empty DMA Request Enable. Bit[3] indicates if a DMA request is generated upon a PCM Tx FIFO almost empty condition. 0: i2s_tx_dma_en is deasserted. 1: i2s_tx_dma_en is asserted. Note: This register only for SOC system. In MCU system, it is a reserved register and read only 1'b0.
2	RW	0	tx_aempty_int_en Tx FIFO Almost Empty Interrupt Enable. Bit[2] indicates if an interrupt is generated upon a Tx FIFO almost empty condition. 0: i2s_tx_aempty_int_en is deasserted. 1: i2s_tx_aempty_int_en is asserted.

1	RW	0	tx_empty_int_en Tx FIFO Empty Interrupt Enable. Bit[1] indicates if an interrupt is generated upon a Tx FIFO empty condition. 0: i2s_tx_empty_int_en is deasserted. 1: i2s_tx_empty_int_en is asserted.
0	RW	0	tx_en Tx FIFO Enable. Bit[0] indicates if the Tx FIFO is enabled. To start the playback, please enable this bit after all other registers have been set. 0: i2s_tx_fifo_en is deasserted. 1: i2s_tx_fifo_en is asserted.

Offset 0x24 – 0x27

TSR

The TSR provides status for the Tx FIFO.

Bit	Attribute	Default	Description
31:4	RO	0	Reserved
3	RW1C	0	tx_overrun_error Tx FIFO Overrun Error. Bit[3] is set if the Tx FIFO has experienced an overrun condition that occurs when the APB writes to the Tx FIFO, and the Tx FIFO is enabled and full. Writing a one will clear this bit. Writing a zero has no effect. 0: No Tx FIFO overrun condition. 1: Tx FIFO overrun condition has occurred.
2	RW1C	0	tx_underrun_error Tx FIFO Under-run Error. Bit[2] is set if the Tx FIFO has experienced and underrun condition. This occurs when the I^{2}S bus requires data for i2s_sd_out and the Tx FIFO is enabled and empty. Writing a one will clear this bit. Writing a zero has no effect. 0: No Tx FIFO underrun condition. 1: Tx FIFO underrun condition has occurred.
1	RO	1b	tx_wr_aempty Tx FIFO Almost Empty. Bit[1] indicates if the number of Tx FIFO invalid entries has met or exceeded the Tx FIFO Threshold (TCR[11:8]). 0: tx_wr_aempty is deasserted (a zero). 1: tx_wr_aempty is asserted (a one).
0	RO	1b	tx_wr_empty Tx FIFO Empty. Bit[0] indicates the Tx FIFO is empty. 0: tx_wr_empty is deasserted (a zero). 1: tx_wr_empty is asserted (a one).

Offset 0x28 - 0x2B

RCR

The RCR Controls the Rx FIFO.

Bit	Attribute	Default	Description
31:12	RO	0	Reserved
11:8	RW	8h	rx_rd_thresh Rx FIFO Threshold. Bit[[11:8] indicate the FIFO entry threshold for the Rx FIFO. The "almost full" condition occurs when the number of valid (occupied) Rx FIFO entries is equal to or greater than this value. Any number from 4'h0 to 4'hF is valid. Bit[11] is the MSB, and Bit[8] is the LSB. 0000: Threshold = 16*32-bit 0001: Threshold = 1*32-bit 0010: Threshold = 2*32-bit ... 1111: Threshold = 15*32-bit Note: This register only for SOC system. In MCU system, it is a reserved register and read only 4'b00000.
7:6	RO	0	Reserved
5	RW	0	rx_over_int_en Rx FIFO Overrun Interrupt Enable. Bit[5] indicates if an interrupt is generated upon a Rx FIFO overrun condition. 0: i2s_rx_over_int_en is deasserted. 1: i2s_rx_over_int_en is asserted.
4	RW	0	rx_under_int_en Rx FIFO Under-run Interrupt Enable Bit[4] indicates if an interrupt is generated upon a PCM Rx FIFO underrun condition. 0: i2s_rx_under_int_en is deasserted. 1: i2s_rx_under_int_en is asserted.
3	RW	0	rx_dma_en Rx FIFO Almost Full DMA Request Enable. Bit[3] indicates if a DMA request is generated upon a Rx FIFO almost full condition. 0: i2s_rx_dma_en is deasserted. 1: i2s_rx_dma_en is asserted. Note: This register only for SOC system. In MCU system, it is a reserved register and read only 1'b0.
2	RW	0	rx_afull_int_en Rx FIFO Almost Full Interrupt Enable. Bit[2] indicates if an interrupt is generated upon the Rx FIFO almost full condition. 0: i2s_rx_afull_int_en is deasserted. 1: i2s_rx_afull_int_en is asserted.
1	RW	0	rx_full_int_en Rx FIFO Full Interrupt Enable. Bit[1] indicates if an interrupt is generated upon a Rx FIFO full condition. 0: i2s_rx_full_int_en is deasserted. 1: i2s_rx_full_int_en is asserted.

0	RW	0	rx_en Rx FIFO Enable. Bit[0] indicates if the Rx FIFO is enabled. To start recording, please enable this bit after all other registers have set. 0: i2s_rx_en is deasserted. 1: i2s_rx_en is asserted.

Offset 0x2C – 0x2F

RSR

The RSR provides status for the Rx FIFO.

Bit	Attribute	Default	Description
31:4	RO	0	Reserved
3	RW1C	0	rx_overrun_error Rx FIFO Overrun Error. Bit[3] indicates the Rx FIFO has experienced an overrun condition. This occurs when the i2s module receives an audio data frame from i2s_sd_in and the Rx FIFO is enabled and full. Writing a one will clear this bit. Writing a zero has no effect. 0: No Rx FIFO overrun condition. 1: Rx FIFO overrun error has occurred.
2	RW1C	0	rx_underrun_error Rx FIFO Under-run Error. Bit[2] indicates the Rx FIFO has experienced an underrun condition. This occurs when the APB Reads from the RX FIFO Read port and the FIFO is enabled and empty. Writing a one will clear this bit. Writing a zero has no effect. 0: No Rx FIFO underrun condition. 1: Rx FIFO underrun error has occurred.
1	RO	0	rx_rd_afull Rx FIFO Almost Full. Bit[1] indicates if the number of Rx FIFO valid entries has met or exceeded Rx FIFO Threshold (RCR[11:8]). 0: rx_rd_afull is deasserted. 1: rx_rd_afull is asserted.
0	RO	0	rx_rd_full Rx FIFO Full. Bit[0] indicates the Rx FIFO is full. 0: rx_rd_full is deasserted. 1: rx_rd_full is asserted.

Offset 0x80 – 0xBC

TFIFO

The TFIFO is the Write port of the Tx FIFO.

This address range will respond only to the 32-bit APB Writes, and will ignore all 8-, and 16-bit Writes. All APB Reads will return zeros.

This location will respond to all APB Reads with all zeros.

The Write pointer will only increment on a 32-bit Write APB ENABLE cycle.

This register is only for SOC system.

In MCU system, it is a reserved register and read only 32'h0.

Bit	Attribute	Default	Description
31:0	WO	0	txdata TxFIFO Write Port.

Note:

The Tx FIFO has the following features:

- Fully asynchronous Read and Write ports.
- 32 entries and 32-bit width.
- Configurable Threshold Indicator.
- Support for overrun, underrun, empty, and almost empty interrupt conditions.
- Support for almost empty DMA request.
- In the overrun condition, the Write port data will be discarded and the Write pointer will not increment.
- In the under-run condition, the last valid FIFO entry data will remain available on the Read port and the Read pointer will not increment.

Offset 0xC0 - 0xFC

RFIFO

The RFIFO is the Read port of the RxFIFO.

This address range will only respond to 32-bit Reads and will ignore all other APB Reads.

All APB Writes will be ignored.

The Read pointer will only increment on the 32-bit Read APB ENABLE cycles.

This register is only for SOC system.

In MCU system, it is a reserved register and read only 32'h0.

Bit	Attribute	Default	Description
31:0	RO	0	rxdata RxFIFO Read Port.

Note:

The Rx FIFO has the following features:

- Fully asynchronous Read and Write ports
- 32 entries and 32-bit width
- Configurable Threshold Indicator
- Support for overrun, underrun, full, and almost full conditions
- DMA support through use of the external DMA controller and module DMA request outputs.
- In the overrun condition, the Write port data will be discarded and the Write pointer will not increment.
- In the under-run condition, the last valid FIFO entry data will remain available on the Read port and the Read pointer will not increment.

Functional Descriptions

Power Management

Power Domains

The system power and reset architecture is structured to support a Battery Power domain, a Suspend Power domain and a Switch Power domain. The three power domain architecture lends itself to many potential applications.

The Battery Power domain is structured such that it will include the logic that is required to maintain the Real Time Clock function.

The Suspend Power domain is structured such that it will include the logic which is required to perform certain Power Management functions. These Power Management functions will include that logic required to perform suspend, wakeup, and reset generation tasks. The following logics are put in Suspend Power domain in WM8505 to support wakeup events from external keyboard and mouse:

- PS/2 keyboard/mouse controller
- Ethernet
- USB host/device, including PHYUSB20 macro

The Switch Power domain is made up of all additional logic within the system, including the IOs not associated with Battery or Suspend Power along with PHYs and all core logic.

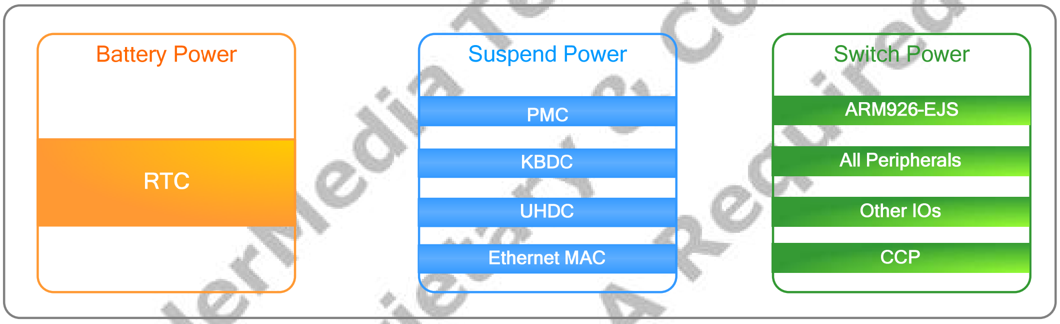


Figure 5 - WM8505 Power Domains

Battery Power

The Battery Power will be provided by the system in such a manner as to meet the needs of the application. The power interface provides the power to the RTC IOs and the RTC standard cell logic via the VBAT/VSST interface pin pairs, and sometimes refers to more specifically as the Permanent Power. It is important to note that it may take up to 1 second after the VBAT power is supplied and before the RTC oscillator circuit stabilizes.

Suspend Power

The Suspend Power provides power source for the PS/2 IOs, Fast Ethernet MII IOs, USB Host and Device IOs, and the logic for Power Management functions and wakeup logics from PS/2, Ethernet, and UHDC. This power is provided via the VSUS33/VSS (for I/O) and VSUS15/GND (for Core) interface pins and always refers to as the Suspend Power.

Switch Power

The Switch Power represents those power planes that may be powered down when the PWREN signal goes inactive (a zero). These include the remaining 3.3V IO Power, 1.5V Core Power and the power to PHYs. This represents the digital logic for the ARM Core, all peripherals, and their associated IOs.

Note:

1. In this architecture, the PWREN signal is defined to be inactive (a zero) when the RSMRST# input is active (a zero).
2. Just driving the PWRGD active (a one) is not sufficient to bring the part out of reset. The PWRGD input is effectively ignored when the PWREN is inactive (a zero).

Power Modes

The power and reset architecture supports multiple Power Modes. These modes represent different levels of operation and power usage by the system. While many of these modes are affected by software controls, some are in effect a hardware-only controlled mode. The modes affected only by hardware are RTC Maintenance Mode and OFF.

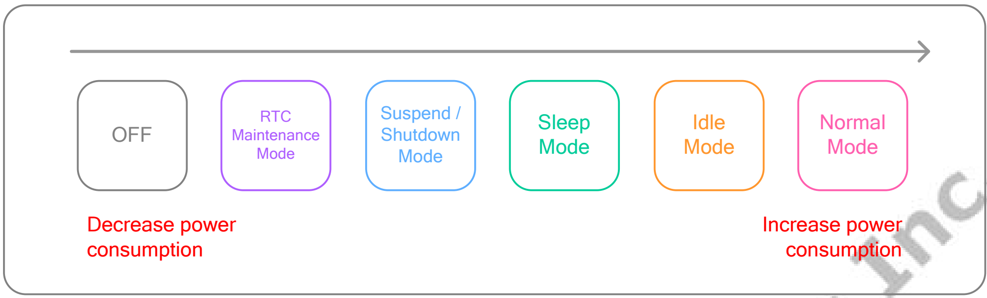


Figure 6 - Power Modes and Power Consumption

Normal Mode

The Normal Mode is the fully powered mode.

After any reset, condition all clocks are enabled. There are software accessible controls for the clocks associated with each peripheral in the system. They may be individually disabled by software control to reduce power usage and effectively disable the associated peripheral. However, the power to this logic is not affected by these controls and will remain powered.

Idle Mode

The IDLE Mode is a special case of the Normal Mode in which the clock to the ARM Core is disabled. The ARM Core clock is enabled upon a system interrupt occurring.

All other clocks are not affected by this mode. They are still controlled by the software accessible controls as described in Normal Mode operation.

Sleep Mode

The Sleep Mode is a power savings mode. This mode disables most clocks in the system. All logic remains powered and normal operation is resumed under various wake-up conditions. This wake-up does not generate a reset to the system, unless the wake-up was a reset generating condition. These wake-up conditions are, in general, software programmable. The exception is the PWRBTN# input which will always produce a wakeup.

Suspend Mode

The Suspend Mode is a deep power savings mode. This mode disables all clocks in the system, other than the RTC. An external signal, PWREN, allows for external power control logic to remove the power to the Switch Domain logic.

This mode supports many of the same wake-up conditions as the Sleep Mode. However, a wake-up from this mode does generate a reset to powered-down logic.

Note that this mode is unique from the RTC Maintenance mode by the distinction that both the VBAT and VSUS33/VSUS15 power pins associated with the Suspend Power domain will have power.

Shutdown Mode

The Shutdown Mode and the Suspend Mode are effectively the same. The only difference will be the status indication to software upon system startup.

RTC Maintenance Mode

The RTC Maintenance Mode is not a software controlled mode. This mode is for the condition of a loss of “wall-power”. In some applications, the RTC must be maintained even in the loss of wall power. This is done through a separate power supply and an indication that allows for isolation of the RTC from the rest of the system.

Note that this mode is unique from the Suspend mode by the distinction that only the VBAT power pin associated with the Battery Power domain will have power.

In systems where the RTC Maintenance mode is not required, the VBAT and VSUS may be connected together to the same source.

OFF Mode

The off mode is exactly the same as the name implies. No power is being supplied to the device.

Power Sequence

This section discusses the order in which the various power rails to the WM8505 must power-up to their appropriate levels.

There are two power-up sequences:

- **Initial Power-up**
  - This occurs when power is first provided to the WM8505 and the entire system is being initialized for the first time.
- **Suspend Power-up**
  - This occurs when the WM8505 has been in its Suspend mode and is returning to Normal Operational mode.

Initial Power-up Sequence

All of the power rails need to be powered up in the correct order to prevent damaging the parts. The following indicate the order in which each of the power rails are to be powered up. Power rails that are designated in the same grouping may power-up in any order relative to each other. .

Power-up Sequence

1) VBAT
2) VSUS15
3) VSUS33
4) VDD
5) VCC33, VCCA33, VCCMEM

Suspend/Shutdown Power-down Sequence

During suspend mode, the following power rails should remain powered:
VBAT, VSUS15 and VSUS33.

While the following power rails may be powered down, when the PWREN signal goes inactive (a zero):
VDD, VCC33, VCCA33, VCCMEM

Note:

1. These power rails may be powered off all at the same time.
2. VCCMEM should have power while suspend to DRAM.
3. Power for USB and MII are also powered in Suspend/Shutdown mode.

Suspend (Wake-up) Power-up Sequence

When the WM8505 returns to Normal Operation from Suspend, the powered down rails should be powered up in the following sequence. It is recommended that a minimum of 1-3 ms separate the powering up of each of these power rails:

Power-up Sequence

1) VDD
2) VCC33, VCCA33, VCCMEM

The following Figure 7, Figure 8, and Table 15 illustrate initial power-on sequence and timing of VT8500.

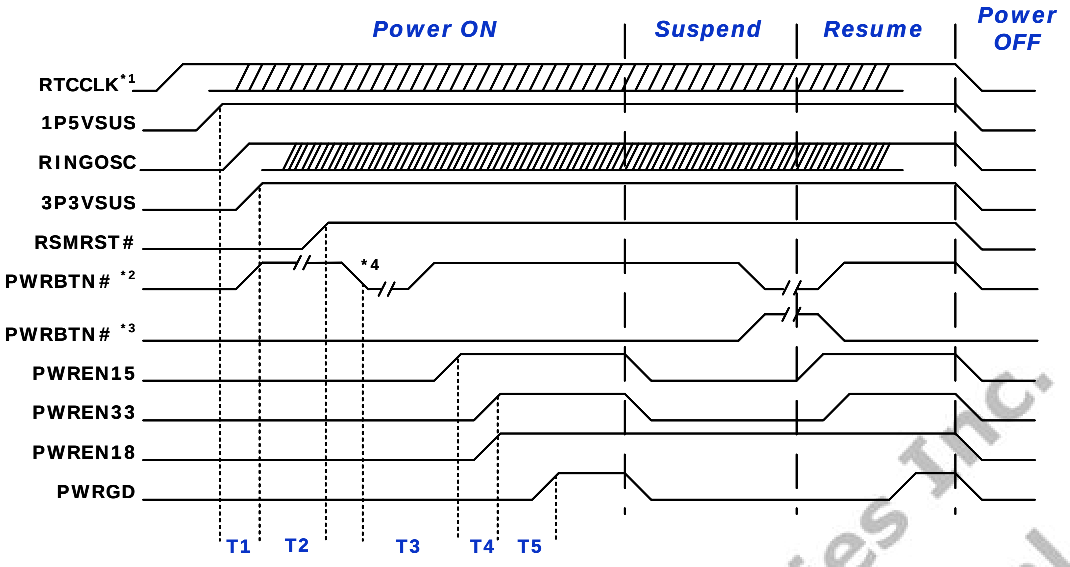


Figure 7 - Power On Sequence w/ Suspend Power

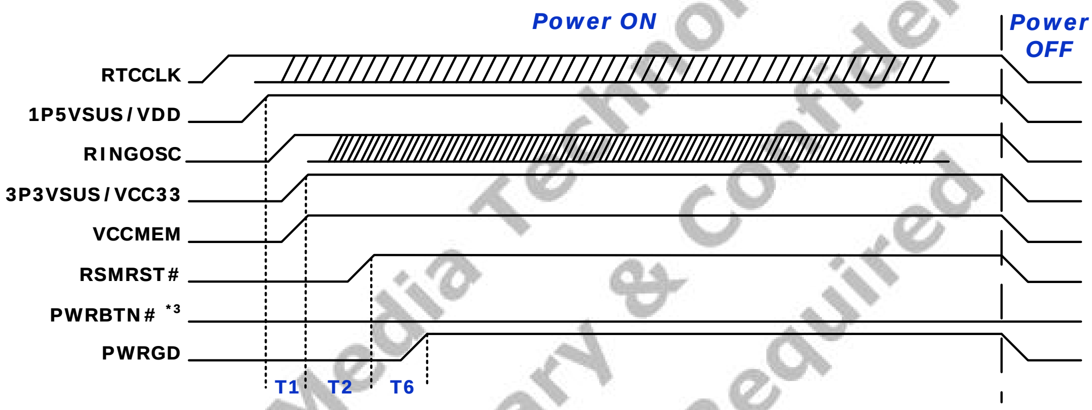


Figure 8 - Power On Sequence w/ Non-Suspend Power

*Note:

1. In some applications, the RTC must be maintained by separate power supply (VBAT) to gain the RTC function. While the RTC Maintenance mode is not required, the VBAT and 3P3VSUS may be connected together to the same source.
2. PWRBTN# will be active low while the system is strapped to Manual Power ON.
3. PWRBTN# will be active high while the system is strapped to Auto Power ON.
4. PWRBTN# shall be asserted low for a period no less than 180 μs and no more than 4 s. The assertion of 4 seconds on PWBTN# will be forced shutdown by the system.

Table 15 - Typical Timing Values of Initial Power-on Sequence

Symbol	Parameter	Min	Typ	Max	Unit
T1	1P5VSUS to 3P3VSUS	2	-	-	ms
T2	3P3VSUS to RSMRST#	20	-	-	ms
T3	PWRBTN# (Active Low) to PWREN15	180	-	220	μs
T4	PWREN15 to PWREN33/PWREN18	2	3	-	ms
T5	VCC33/VCCMEM to PWRGD	5	-	-	ms
T6	RSMRST# to PWRGD	5	-	-	ms

Strapping Options

To support more than one possible option for certain functions, the WM8505 supports a number of strapping options. Strapping options are values that are latched into the device during reset (for the WM8505 when the RSMRST# signal is active / low) and then used immediately as the part exits reset. When the RSMRST# signal is active (low), the signal pins in Table 16 will operate as Strapping-option inputs

These signals will act as inputs when the RSMRST# signal is active (low) and within each I/O Cell associated with these signals a weak pull-down resistor (10K) will turn on while the RSMRST# signal is active. This means that by default the strapping option value associated with each signal will be a zero (due to the enabled I/O Cell pull-down resistor). If the Strapping-option value associated with any given signal is desired to be a one, then a 1K pull- up resistor should be connected to this signal pin external to the WM8505.

When the RSMRST# signal transitions from active (low) to inactive (high), the WM8505 will close the strapping option registers and the associated strapping option values will be the values of these signal pins when the RSMRST# signal rises. While the RSMRST# signal is inactive (high), the strapping option values will be held.

Note that these signal pins associated with the strapping options are all output signals when operating in their normal peripheral mode. However, these signals may also be configured to operate as General Purpose I/O (GPIO) pins as well. If any of these signal pins are used as GPIOs, it is recommended that they should be used as outputs (not inputs). If any of these signal pins are used as GPIO inputs, then it is the system designer's responsibility to guarantee that they are either not driven when the RSMRST# signal is active (low) or they are driven with the correct Strapping-option values when the RSMRST# signal is active, otherwise the WM8505 may not operate correctly.

Table 16 - WM8505 Strapping Options

Pin	Default	Definition
PWRBTN#	1'b0	Strapping_Sus_Option[31] Auto PowerOn Switch Domain Power. 0: Auto enable switch powers after resume reset deasserted. PWRBTN# input is active HIGH. 1: Need PWRBTN# pressed to enable switch powers. PWRBTN# input is active LOW.
MIIPHYPD#	1'b0	Strapping option[30] Reserved
SUSGPIO	1'b0	Strapping_Sus_Option [29] Genet REVMII Mode. This strapping option determines whether Ethernet MAC operates in MII mode or Reverse MII (REVMII) mode. Take effect only Strapping_Sus_Option[0] = 1. 0: MII mode. 1: REVMII mode.
VSUS33	1'b0	Strapping_Sus_Option [28] Fast Ethernet MII Mode. This strapping option must be pulled-up to enable Fast Ethernet MAC. 1: MII/REVMII mode.
N/A	N/A	Strapping Option[27:26] Reserved.
NORBAA#	1'b0	Strapping Option[25] 0: The SF interface will use rising edge sample the input data. 1: The SF interface will use clock negedge to sample the input data. Used for high clock frequency and the flash output delay too large.

Pin	Default	Definition
NORA[24]	1'b0	Strapping option[24] UARTX6 Enable. This strapping option is used to enable 6 UART supports. 0: UART0 ~ UART3 in full UART mode. 1: UART0, UART1 in full UART mode. UART2 ~ UART5 in half UART mode.
NORA[23:21]	3'b111	Strapping option[23:21] Reserved
NORA[20:8]	0	Strapping option[20:8] Reserved
NORA[7]	0	Strapping_Options[7] Enable Keypad function
NORA[6:5]	2'b00	Strapping Option [6:5] NOR Flash Boot Timing Select. When NOR Flash boot enabled by strapping option [2:1], Bit [6] is used to specify NOR Flash reset timing. 0: Reset timing register get max value. 1: Reset timing register get typical value. When NOR Flash boot enabled by strapping option [2:1], Bit [5] is used to specify NOR Flash read timing. 0: Read timing register get max value. 1: Read timing register get typical value.
NORA[4]	0	Strapping Option [4] NOR Flash Boot Page Mode Support. When NOR Flash boot enabled by strapping option [2:1], this bit is used to enable / disable NOR Flash page mode support. 0: Non-page mode. 1: Page mode enabled.
NORA[3]	0	Strapping Option [3] Serial Flash Address Size (when Serial Flash boot). When Serial Flash boot enabled by strapping option [2:1], this bit is used to Determines the size of the Serial Flash Address. 0: 24-bit address. 1: 32-bit address. NOR Flash Bit Width (when NOR Flash boot). When NOR Flash boot enabled by strapping option [2:1], this bit is used to select the bit width of the external NOR Flash chips. 0: 8 bit. 1: 16 bit.
NORA[2:1]	2'b00	Strapping Option [2:1] NOR / Serial Flash Boot-up. This strapping option is used to select between NOR Flash and the Serial Flash interfaces, the enabled interface will then be responsible for boot code fetching. 00: NOR Flash interface. 01: Reserved. 1x: Serial Flash interface
NORA[0]	0	Strapping Option[0]. Reserved.

Clock Architecture

WM8505 uses two external clocks or crystal networks and one internal ring oscillator to generate all internal clocks needed for normal operations. The frequencies of the external clocks or crystal networks are:

27 MHz
To provide source clock for the internal clock synthesizer that generates all internal clocks needed for normal operations except Real Time Clock and power-up control signal generator in Power Management Interface.

32.768 kHz
To provide clock for Real Time Clock

27 MHz Clock Source

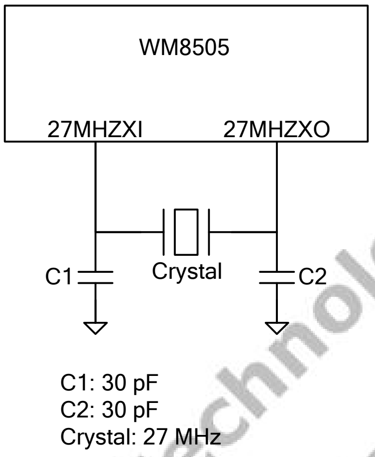


Figure 9 - 27 MHz Crystal Network

PLL 0, 1, 2, and 3 are grouped together as the primary PLL in WM8505. Their purposes are as following:

PLL 1
PLL 1 is used to generate 25MHz clock for ARM, AHB, A/V Engines, all peripherals, and Ethernet MAC. It is fixed to multiple by 25

PLL 2
PLL 2 is used to generate 24 MHz clock for USB, UART, OS Timer, and I^2C. It is fixed to multiple by 8

PLL 3
PLL 3 is used to generate audio input/output master clock. It can be programmed to be multiple by P.

- Fs is the sample frequency of audio data. It should be selected from 64, 88.2, 96 kHz.
- N is the over sampling rate. It should be selected from 256 or 384.
- R should be selected from 1, 2, or 4.
- This PLL will use P, Q to approximate the real frequency to reach the ideal frequency.
- The possible combinations of P, Q are in following table: (N*Fs = 27MHz * P / Q)
- CLKAUDSEL [2:0] in Table 17 are defined in register (Offset 034C).

Table 17 - Audio Master Clock Frequency Selection

CLKAUDSEL [2:0]	N*Fs (kHz)	Ideal Freq.(MHz)	P	Q	Real Freq.(MHz)
000	256*64	16.384	125	206	16.3835
001	384*64	24.576	223	245	24.5755
010	256*88.2	22.5792	143	171	22.5789
011	384*88.2	33.8688	143	114	33.8684
100	384*96	36.864	71	52	36.8653

Internal Clock Generation

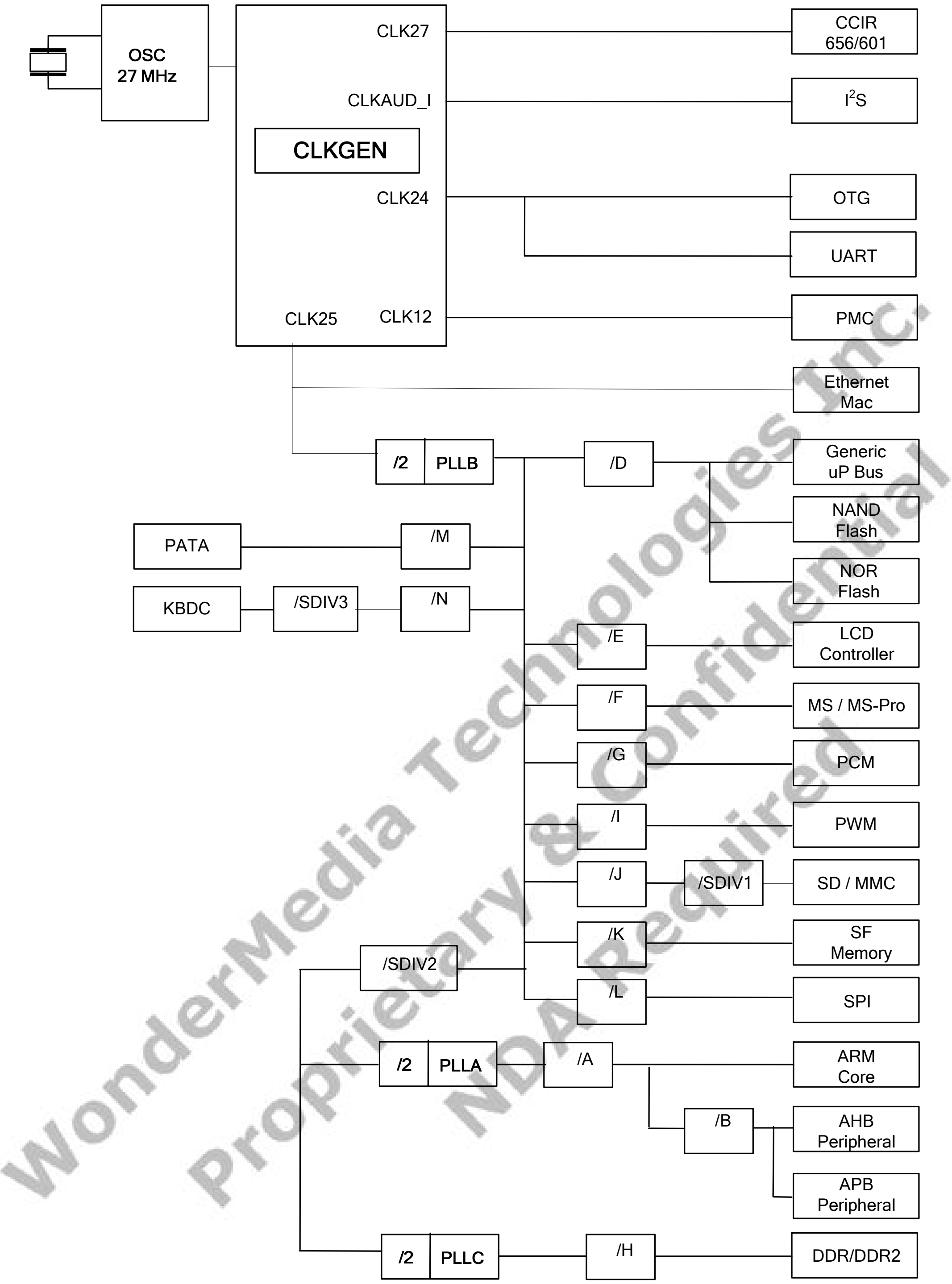
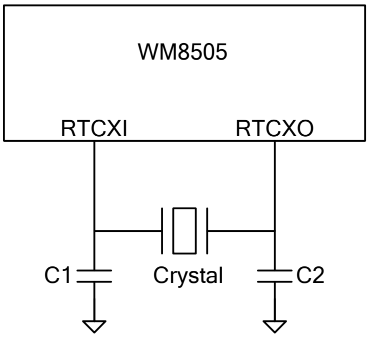


Figure 10 - Internal Clock Generation of 25 MHz Clock Network

The clock frequency divisor (A, D, E, … Z) in Figure 10 are defined in registers (Offset 200 – 345) of PMC (Power Management Control)

RTC Clock Source



C1: 15 pF
C2: 15 pF
Crystal: 32.768 kHz

Figure 11 - RTC Crystal Network

Unlike its predecessor, WM8505 does not rely on 32.768 kHz clock or crystal network to provide clocking for power-up control signal generator. It now uses the internal ring oscillator for this purpose. Hence for systems without the need of Real Time Clock, the external 32.768 kHz clock or crystal network is no longer needed. Figure 12 shows how the RTCXI and RTCXO should be connected when RTC is not required.

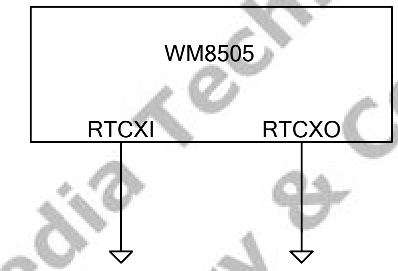


Figure 12 - RTCXI/RTCXO Connection as RTC not Needed

DMA Controller

DMA Controller Features

- Supports 16 channel memory-to-peripheral, peripheral-to-memory, memory-to-memory DMA transfers
- Support Scatter-Gather DMA
- Support appending descriptors
- Support both software and hardware request DMA transaction
- Support programmable priority scheme
- 1 AHB slave interface (S_IF) for configuration; 1 AHB master interface (M_IF_0) to access the memory; 2 AHB master interfaces (M_IF_1/2) to access the memory or the peripherals

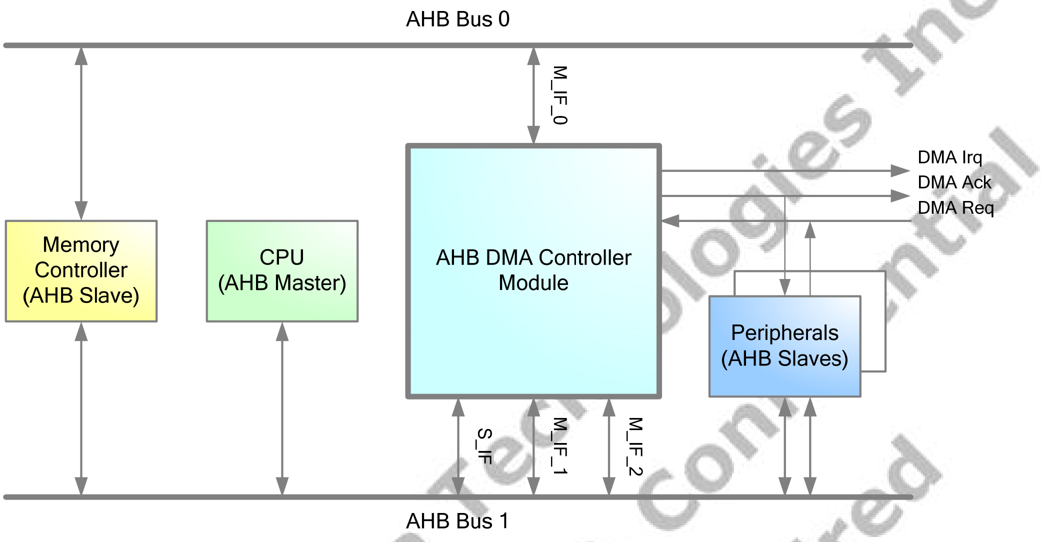


Figure 13 - AHB DMA Controller in System

Note: S_IF, M_IF_1 and M_IF_2 are not necessary on the same AHB bus.

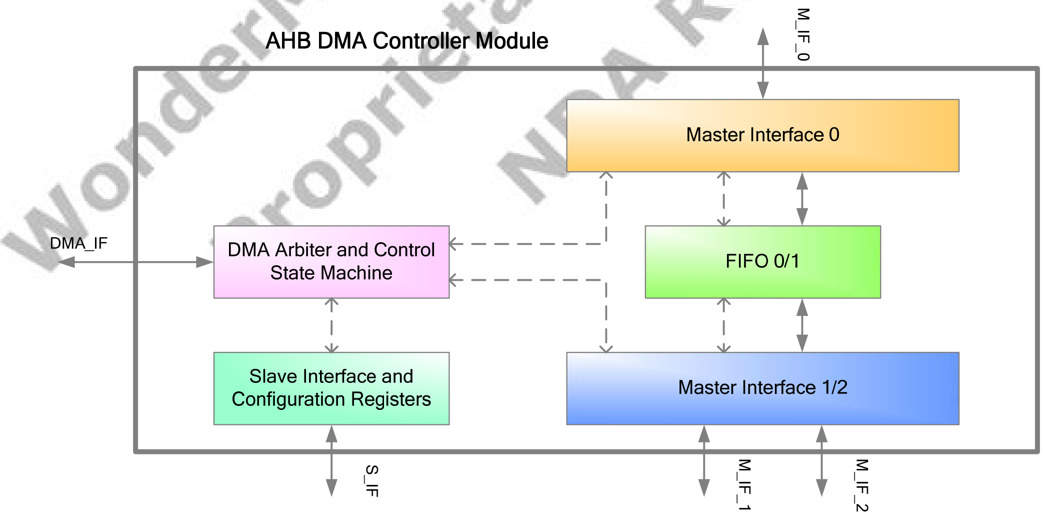

AHB DMA Controller Module
M_IF_0
Master Interface 0
DMA_IF
DMA Arbiter and Control State Machine
FIFO 0/1
Slave Interface and Configuration Registers
Master Interface 1/2
S_IF
M_IF_1
M_IF_2

Figure 14 - AHB DMA Controller Internal Block Diagram

The following sections provide additional detailed user information needed to utilize the system DMA controller for data transfer between system memory and various DMA peripherals.

DMA Bus Interface Feature

The DMA controller have three bus interfaces: one AHB slave interface for programming the DMA control registers; one AHB master interface to access the memory (with the name M_IF_0); two AHB interface to access the memory/ peripheral (with the name M_IF_1 and M_IF_2).

- **AHB Slave Interface**
  - AMBA revision 2.0 compliant 32-bit AHB slave interfaces
  - Supported burst operations: SINGLE, INCR, INCR4, INCR8, INCR16
  - Supported transfer size: 8bits, 16bits and 32 bits
  - Supported transfer type: IDLE, BUSY, NONSEQ, SEQ
  - Supported transfer response of OKAY
- **AHB Master Interface to Access the Memory/Peripheral**
  - AMBA revision 2.0 compliant 32-bit AHB master interface
  - Supported burst operations: SINGLE, INCR4, INCR8
  - Supported transfer size: 8bits, 16bits and 32 bits
  - Supported transfer response of OKAY, ERROR, RETRY and SPLIT
  - Supported transfer type: IDLE, NONSEQ and SEQ
  - Do NOT generate INCR16
  - Do NOT generate wrapping burst
  - Configurable whether or not generate INCR burst
  - Do NOT generate BUSY cycles.

Basic DMA Operation

The system DMA controller in the WM8505 can transfer a block of data from one device to another device without intervention by the ARM. For example, the system program can move a block of data from external memory into the transmit FIFO of one of the UARTs using the DMA controller. Before transferring the data, the DMA controller must be programmed with the essential parameters of a DMA operation, such as the starting address of the data and the address where the data should go to. The programming is done by a program executed by the ARM on the WM8505. Here is the basic set of parameters to program in order to perform a DMA transfer:

- Program the source address register with the starting address of the data,
- Program the destination address register with the starting address where the data should go to,
- Program the transfer count register with the amount of data to be transferred,
- Program the transfer direction bit in the channel configuration register to indicate the direction of data flow.

Once these registers are programmed, the DMA controller and the selected DMA channel need to be enabled by writing a “1” into the DMA_Enable bit of the Global Control register and the Channel_Enable bit of the Channel Control register. If the transfer is to a device which does not have a hardware DMA Request signal (e.g., system memory), the Software_Request bit in the Channel Control register also needs to be set to “1”.

The DMA controller will start the operation by reading data values from the source area and storing them into an internal FIFO. The source address register will increment to point to next piece of data. Then the DMA controller will forward the data to the destination area. The destination address register will increment and the transfer count register will decrement. The DMA controller will continue this operation until the transfer count register decrements to zero. At this point the DMA controller will disable the DMA channel by resetting the channel enable bit. It will also set the TC (Terminal Count) bit in the channel status register. Software is responsible for reading and resetting this status bit.

DMA Channel Feature

- Support memory-to-peripheral, peripheral-to-memory, memory-to-memory transfers as long as the source and the destination are connected to the different master interfaces of the DMAC.
- Support 16 DMA active channels at most.
- Support 32 DMA request sources.
- 16 individual channel interrupt requests on a per channel basis
- Programmable 2-bit address wrap length register on per channel basis for M_IF_1/2, support 1 burst, 2 burst, 4 burst and 8burst
- Programmable 2-bit burst length for each DMA request register on per channel basis for M_IF_1/2; support SINGLE, INCR4, INCR8. M_IF_0 always uses INCR8.
- Programmable 8-, 16- and 32-bit transfer size of M_IF_1/2 on per channel basis for M_IF_1/2. M_IF_0 always uses 32-bit transfer size.
- If the transfer size is 8/16/32 bits, the source address, destination address and transfer count are not necessary 8/16/32 bits alignment.
- M_IF_0 NOT supports Non-SG mode.
- Programmable SG mode or Non-SG mode for M_IF_1/2 on a per channel basis.
- Supports configurable software request DMA and hardware request DMA on a per channel basis.
- Programmable transfer directional on per channel basis.
- The transfer direction is fixed in one DMA operation but can change from one DMA operation to the next DMA operation.
- Supports flexible DMA request port signal to DMA channel mapping through dma_req_id field.
- Memory_register used.
- Programmable descriptor base address in memory_register for M_IF_0 and M_IF_1/2 on per channel basis.
- Programmable data address in memory_register for M_IF_1/2 when it is in Non_SG mode.
- DMA works in Little Endian that can support 32-bit data conversion from big endian to little endian for memory to memory transfer.
- DMA operation aborted when an AHB ERROR response is received.
- Support two kinds of descriptor format which enables the mix of descriptor_array and descriptor_chain.
- Configurable interrupt enable on a per descriptor basis.

Possible System DMA Transfer Sources and Destinations

The 16-channel System DMA Controller is capable of making transfers between the following:

- Internal DMA-requesting peripheral to / from System Memory.
- System Memory to System Memory.

The System DMA Controller will typically transfer to / from System Memory (i.e. external SRAM, DRAM or Flash) via the WM8505's Memory Controller.

Data Sizes and Data Alignment

DMA transfer data sizes refer to the size of the data to be transferred in each internal bus cycle. The AHB bus in the WM8505 is 32-bit and both the DMA controller and the memory controller support up to 32-bit data. They also support 16-bit and 8-bit data. The choice of data size is primarily based on the data size the peripherals can handle. Once the data size is determined, the programmer has to make sure the source and destination addresses programmed into the DMA controller are aligned to an address boundary equal to the data size. For example, if the data size is programmed to 32-bit, the lower 2 bits of the address must be 00 (A[1:0] = 00). If the data size is programmed to 16-bit, the lower 1 bit of the address must be zero. The 8-bit data size allows the address to be programmed at any location. The source and destination addresses do not need to align at the same boundary. For example, for 32-bit data size, the source address can be 1234_0008h and the destination address can be 1235_0000h. For 16-bit data size, the source address can be 1234_0000h and the destination address can be 1235_0002h. For 8-bit data size, the source and destination addresses can be at any location.

IMPORTANT NOTE: Although the following is not a requirement of the system DMA Controller, it is important for achieving optimum system performance. In deciding the starting address of DMA buffers, it is recommended to have the DMA buffers start at a cache line boundary, i.e. at 32- byte address boundary. If this is done, the memory controller will work at maximum efficiency.

Burst Transfers

During a DMA operation, the DMA controller divides the block of data into many small units. It transfers a unit of the data, then it releases the internal bus to allow other bus masters to use the bus. At the same time, it re-arbitrates DMA requests, i.e. it takes a snapshot of all DMA requests again and picks the highest priority channel to service.

This allows a fair use of the internal buses and other common resources by other AHB masters and different DMA channels (when the DMA controller is programmed to rotating priority). The size of the data unit can be programmed to a single piece of data, 4 pieces of data or 8 pieces of data. In the later two cases, 4 or 8 pieces of data are transferred across the bus without interruption. This is known as a burst transfer. Burst transfers are more efficient than single transfers because the DMA controller does not need to request the internal bus and re-arbitrate DMA requests as often. However, using burst transfers will lengthen the time the DMA controller occupies the bus (in a single transfer unit) and the time between re-arbitration of DMA requests, i.e. lengthening of both the internal bus latency as well as the DMA request latency.

IMPORTANT NOTE:

When a DMA channel is programmed to perform bursts of 4 or 8, the DMA controller will perform a burst of 4 or 8 (respectively) in response to each DMA request recognized by the DMA controller; so it is very important that when a peripheral makes a request on a DMA channel configured to perform a burst of 4 or 8, that the peripheral is capable of handling all 4 or 8 (respectively) transfers. Peripherals that are not capable of supporting this requirement must configure their DMA channels to only perform single DMA transfers.

Transfer Count

The DMA channel has a 24-bit transfer count register. Program the total amount of data to be transferred in this register before enabling the DMA channel. It can program 00_0001h, (i.e. transfer 1 piece of data, whether it is 8-bit, 16-bit or 32-bit), up to FF_FFFFh (i.e. transfer 2^24 – 1 pieces of data) into this register.

Once the DMA operation has started, the transfer count register will decrement as each data reaches the destination. When the last data transfer has finished (i.e., when the data transfer has terminated), the transfer count will be decremented to zero, so this condition is known as the Terminal Count (TC).

When terminal count occurs, the DMA operation has finished normally. If Dual Buffer mode is not enabled, the channel enable bit will be reset to disable the channel. The TC status bit in the channel status register and the global status register will be set.

Do not enable the DMA channel when the transfer count is zero as this may cause an unpredictable result. Do not write to the transfer count register before terminal count or after the DMA operation is terminated by an error. Reading the transfer count register during the DMA operation is allowed. However, this will give only an approximate snapshot of the transfer count because it is continuously updated on-the-fly.

Error Conditions

During a DMA operation, the DMA Controller may receive an error response from the memory controller or the bus arbiter. A typical reason for an error response is that the transfer address falls into a reserved address area, usually an area not decoded by any memory or peripheral devices.

When the DMA controller receives an error response, it will stop the DMA operation by resetting the channel enable bit in the channel control register and setting the appropriate error status bit in the channel status register and global status register.

When the error occurs on the source side, i.e. the DMA controller is reading the data, it will stop further reading and transfer the data read to the destination. When the error occurs on the destination side, i.e. the DMA controller is writing the data, it will stop further writing, and the data in the internal buffer of the DMA controller will be discarded. After the DMA controller stops the operation, the transfer count register will show exactly how many pieces of data are remaining. The software can then determine how many pieces of data have been transferred.

IMPORTANT NOTE:

Address and Transfer Count Registers Values after a Data Transfer Error:

When the error occurs at the source side, the Source Address Register (SAR) will stay at the address where the error occurred. The Destination Address Register (DAR) will stay at the address where the next piece of data should go.

The Transfer Count Register (TCR) will show how many pieces of data are remaining. Therefore it is safe to re-enable the DMA channel without re-programming these registers after the error condition is cleared. The DMA operation will pick up where it left off.

When the error occurs on the destination side, caution should be taken because the DMA controller could read extra data before it disables the channel. The TCR will show exactly how many pieces of data are remaining, so the amount of data that has been transferred can be calculated.

However, the TCR has to be re-written in this case with the read back value. This is to re-synchronize internal registers. The SAR should always be re-programmed to the original SAR value plus the number of pieces of data transferred.

The DAR will correctly point to the next address to be written, so it is optional to re-program the DAR. The channel can be re-enabled after these registers are re-programmed.

DMA Request Priority Schemes

When multiple DMA channels are programmed and enabled, multiple DMA requests may occur at the same time. The DMA controller will only select the channel with the highest priority to service. The way to assign different priorities to different channels is the priority scheme. There are two user selectable priority schemes: fixed priority and rotating priority.

The fixed priority scheme is straightforward. The priority of a channel is associated with its channel number and is fixed. Namely, channel 7 has the highest priority and channel 0 has the lowest priority.

The rotating priority scheme rotates, or re-assigns priorities to channels based on when they were last served.

Figure 15 below shows the concept of the rotating priority scheme. Any channel which requests DMA service and wins the arbitration will be served by the DMA controller. The channel will then be placed at the lowest priority. All channels with lower priority previously will have their priority increased by 1. All channels with higher priority before will have their priority unchanged. The priority of each channel after a system reset is shown in column 1 of Figure 15.

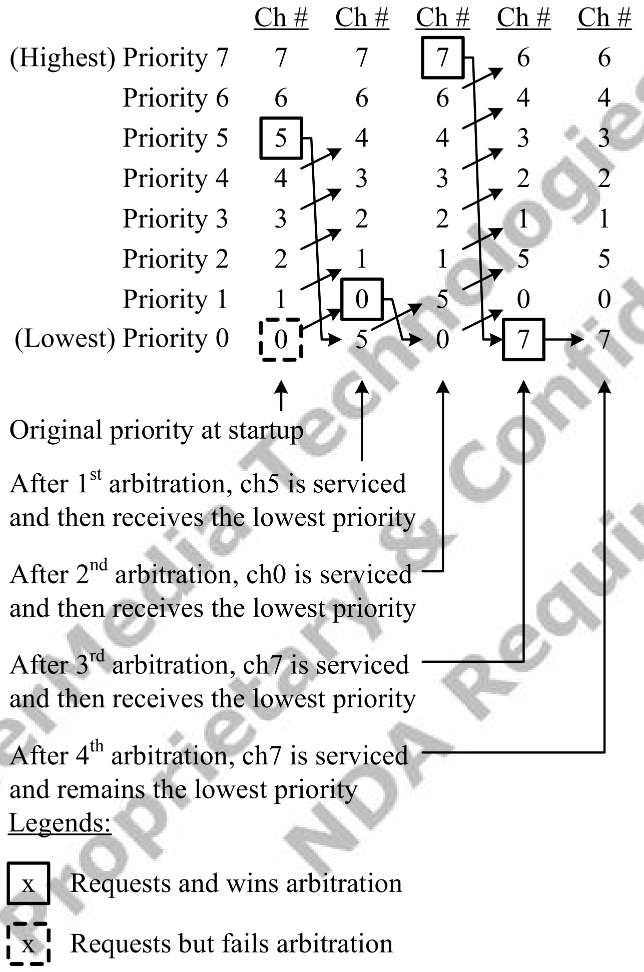

Ch # Ch # Ch # Ch # Ch #
(Highest) Priority 7 7 7 7 6 6
Priority 6 6 6 6 4 4
Priority 5 5 4 4 3 3
Priority 4 4 3 3 2 2
Priority 3 3 2 2 1 1
Priority 2 2 1 1 5 5
Priority 1 1 0 5 0 0
(Lowest) Priority 0 0 5 0 7 7
Original priority at startup
After 1st arbitration, ch5 is serviced and then receives the lowest priority
After 2nd arbitration, ch0 is serviced and then receives the lowest priority
After 3rd arbitration, ch7 is serviced and then receives the lowest priority
After 4th arbitration, ch7 is serviced and remains the lowest priority
Legends:
x Requests and wins arbitration
x Requests but fails arbitration

Figure 15 - DMA Rotating Priority Scheme

Software DMA Request

For peripherals which do not support DMA requests or for memory devices, the software may use the software request to perform DMA transfers. When using software requests, DMA channels are not tied to any particular peripherals, i.e. any free channel may be used to perform the DMA operation. Once a channel is properly programmed, DMA is started by writing the Software_Enable bit of the channel control register to “1”. While the DMA operation is going on, the program can suspend the DMA operation by resetting the software request bit. However, this is not a precise way to control the DMA operation. The software can read the transfer count or the address registers to approximately determine the progress of the DMA operation because these registers are being updated on-the-fly during DMA operation. Do not try to write to these registers while DMA is running as this may produce unpredictable results.

Interrupt Generation

In addition to setting the status bits in the status registers, the DMA controller can generate an interrupt for the events of terminal count and data transfer error. The Global_Int_En bit in the Global Control register and the individual Int_En bits in each of the channel control registers must be set in order to activate the interrupt output. The interrupt output can be reset by resetting all the status bits in the channel status registers which caused the interrupt.

Dual Buffer Operation

The system DMA controller supports dual buffer operation, allowing the DMA to automatically load in a secondary set of DMA parameters (i.e. source address, destination address and transfer count) when it reaches a terminal count (TC) condition. This allows the DMA controller to continue serving DMA requests from the same peripheral (until the next TC) without the need of immediate re-programming of the DMA parameters and re-enabling the DMA channel. This feature significantly relaxes the latency time requirement from receiving a DMA TC interrupt to reprogramming the DMA channel in data streaming applications.

The dual buffer function can be enabled / disabled on a per channel basis.

- **Dual_Buf_En = 0:**
  Dual buffer mode is disabled. When the DMA channel reaches TC condition, an interrupt is generated and the Ch_Enable bit will be reset. The transfer count register must be re-programmed and the Ch_Enable bit re-enabled in order to put the channel back to service.

- **Dual_Buf_En = 1:**
  Dual buffer mode is enabled.

- **Typical Programming Sequence and DMA Operation:**

  1. Before enabling a DMA channel, program source / destination address registers 0 and transfer count register 0 to point to buffer 0. Program the channel configuration register and global control register. The program can optionally program source / destination address registers 1 and transfer count register 1 to point to buffer 1. Then program the channel control register to enable the channel as well as dual buffer mode.
  2. The DMA channel will start serving DMA requests. Before it reaches the TC of buffer 0, buffer 1 parameters (source / destination address & transfer count) must be programmed. Note that the transfer count register must be the last one to be programmed among the 3 registers. Upon reaching the TC of buffer 0, the DMA controller will recognize buffer 1 parameters as valid and load them in for the next DMA request. The channel will stay enabled but the TC bit will be set and a TC interrupt will be generated. If the Software_Request bit is set, when DMA operation switches to the buffer 1, the Software_Request bit will stay set. If transfer count register 1 has not been written, the buffer 1 parameters are considered to be invalid and the DMA controller will disable the channel upon TC.
  3. Upon receiving the DMA controller TC interrupt, the software should service the interrupt by checking channel status to make sure no bus error or abort has occurred. If the Ch_Enable bit is still set and Buffer_Pointer is "1", the DMA channel has switched to buffer 1 and is still serving DMA requests. The software should re-program buffer 0 parameters. Transfer count register 0 must be the last one to be programmed. Then the program can process the data in buffer 0 and return to wait for the TC of buffer 1.
  4. Upon the TC of buffer 1, the DMA controller will recognize buffer 0 parameters as valid and load them in for the next DMA request. The channel will stay enabled but the TC bit will be set and a TC interrupt will be generated. If transfer count register 0 has not been written, buffer 0 parameters are considered to be invalid and the DMA controller will disable the channel upon TC.
  5. Upon receiving the TC interrupt of the DMA controller, the software should service the interrupt by checking channel status to make sure no bus error or abort has occurred. If the Ch_Enable bit is still set and Buffer_Pointer is "0", the DMA channel has switched to buffer 0 and is still serving DMA requests. The software should re-program buffer 1 parameters. Transfer count register 1 must be the last one to be programmed. Then the program can process the data in buffer 1 and return to wait for the TC of buffer 0.

6. Steps 2 to 5 will repeat indefinitely until either the software fails to re-program the transfer count of the next buffer in time so the channel will be disabled upon TC, or the DMA controller encounters an abort or bus error. When the DMA controller receives an abort or bus error, it will generate an interrupt and disable the Ch_Enable bit and the Software_Request bit. The source / destination address and transfer count of the current set in use will stay where they are when the abort / error occurred. The Buffer_Pointer will also stay at its current value. The software should re-initialize the DMA channel by:

   i. Clear the Error / Abort status bits.

   ii. Re-program source / destination address registers 0 and transfer count register 0.

   iii. Re-program source / destination address registers 1 and transfer count register 1.

   iv. Write a "1" to DMA_CSR_Chn bit 6 to reset Buffer_Pointer to 0.

   v. Re-enable the channel.

LCD Controller

LCD Controller DMA

The system DMA Controller is an AHB master in order to generate proper memory accesses for the LCD controller to read the external frame buffer and palette buffer. Data values coming in are stored in the 32-bit wide by 24 deep data input FIFO. In single scan mode, the DMA Controller only uses channel 0 for both palette buffer and frame buffer reads. When dual scan mode is selected, the DMA Controller uses channel 0 to fetch palette buffer and the upper half of the frame buffer. It uses channel 1 to fetch the lower half of the frame buffer. A separate FIFO is used to store data from DMA channel 1. The 2 DMA channels access memory in an interleaving way in dual scan mode.

When the LCD Controller is enabled, if the LCD mode settings require palette data (i.e. LCD_BPP = 1, 2 bpp in monochrome display mode, LCD_BPP = 1, 2, 4, 8 bpp in color display mode), the DMA Controller will load the LCD_Palette_Page_Addr (palette buffer start address) into the LCD DMA buffer 0 current address register and start DMA reads from the external palette buffer. The amount of palette data to load depends on the LCD_BPP setting and is calculated by the DMA controller hardware. When external palette data loading is finished, the frame buffer page address (LCD_FB_Page_Addr) and the frame buffer start address (LCD_FB0_Start_Addr) are loaded into the buffer 0 current address register in order to load data from the frame buffer. The amount of frame buffer data to load is calculated by the DMA Controller hardware. At the end of reading a complete frame buffer, the DMA Controller can be programmed to reload the palette or bypass palette loading and just re-fetch the frame buffer.

Frame Buffer Data Alignment

Each line of pixel data must start at a word (32-bit) boundary in the frame buffer. This requires the line size of the panel, in terms of pixels per line, to be an integral multiple of certain numbers. The following table shows the multiple numbers for various bit-per-pixel formats.

Table 18 - Line Size Multiples for Different Bits Per Pixel Formats

Bits Per Pixel	"Pixels-Per-Line" must be a multiple of
1 bpp	32
2 bpp	16
4 bpp	8
8 bpp	4
12 or 16 bpp	2
18 or 24 bpp	1

NOTE:

If the LCD panel's line size does not fall into the required multiple, the data values of each line still need to start at a word boundary. This leaves a few bit or byte locations unused at the end of each line in the frame buffer. In addition, the register LCD_PPL value must be adjusted to have the required multiple. The LCD controller will then send out a few dummy pixel data values at the end of each line. The panel must be able to ignore this dummy pixel data.

Frame Buffer Organization

The frame buffer is a block of memory located in the system memory area. It stores all the pixel data of a complete frame of the display. Its starting location is pointed to by the LCD_FB0_Start_Addr register. LCD Controller DMA will continuously read pixel data from the frame buffer and feed it to the LCD controller. The organization of the frame buffer data depends on the bit size of the pixels. Note that in the following tables, P0 denotes the first pixel at the upper left corner of the screen. P1 is the second pixel located on the same horizontal line to the right of P0 and so on.

Table 19 - Frame Buffer Format for 1 bpp Data

	Bit								
Offset Addr	31	30	29	28	…	3	2	1	0
00h	P31	P30	P29	P28	…	P3	P2	P1	P0
04h	P63	P62	P61	P60	…	P35	P34	P33	P32

Audio I^{2}S Interface

The Inter-Integrated Circuit Sound (I^{2}S) Controller for APB (i2s) Module is an APB peripheral supporting data transfer to and from an I^{2}S CODEC via an I^{2}S bus. The controller provides an interface for the host processor to transfer audio data to the CODEC from memory and conversely to memory from the CODEC.

The I^{2}S Controller for APB (i2s) Module has two primary interfaces, the Advanced Peripheral Bus (APB) and the I^{2}S Bus. For purposes of this specification, "receive" and "transmit" may be abbreviated as "rx" and "tx" respectively. The direction of data flow will always be in reference to the host controller (e.g. rx indicates data flow from the I^{2}S CODEC to the APB, and tx indicates data flow from the APB to the I2S CODEC).

Features Supported

The internal i2s codec supports 24 bit data format with slave mode, and stereo ADC and DAC with sample rates of 8 kHz, 16 kHz, 48 kHz, 96 kHz, 192 kHz, and 44.1 kHz.

- **5 / 6 / 8 pins of digital audio interface supported**
  - 8 pins mode (DCDIN, DACDOUT, TXMCLK, TXSYNC, TXBCLK, RXSYNC, RXBCLK, RXMCLK) supports independent sample rate for playback and recording.
  - 6 pins mode (ADCDIN, DACDOUT, TXMCLK ,TXSYNC, TXBCLK, RXSYNC) supports different sample rate for playback and recording, which are both synchronous with the master clock(txmclk).
  - 5 pins mode (ADCDIN, DACDOUT, TXMCLK, TXSYNC, TXBCLK) supports the same sample rate for playback and recording.
- **Sample bit: 8 / 13 / 14 / 16 / 20 / 24 / 32 bit**
- **Full Duplex Mono and Stereo supported**
- **Master and Slave modes supported**
- **Five popular interface formats are supported**
  - Right-justify mode
  - Left-justify mode
  - I2S mode
  - DSP Early mode
  - DSP Late mode
- **All transfer support MSB / LSB first**
- **Bit clock polarity can select**
- **Programmable offset where the data begins in each frame**
- **MCU Bus Supported in MCU system, or APB System Interface Supported in SOC system**
- **SPDIF TX OUT is supported in SOC system.**

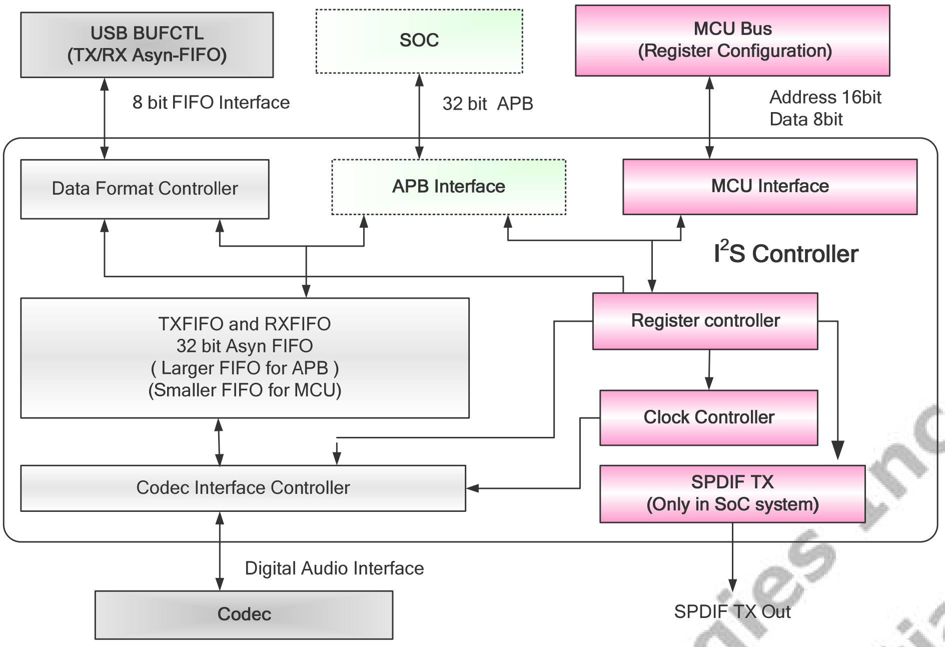


Figure 16 - Audio Controller Module Block Diagram

General Purpose Input/Output

Selection of GPIO or Peripheral

Following a power-up Resume Reset (RSMRST#), the signal pins that may operate as GPIOs will default to their General Purpose input functionality. The associated peripherals may be configured to use these signal pins via GPIO Enable Control registers 0000 to 0018. I/O Input Reset, Software Reset and Watchdog Reset will not effect the GPIO and Peripheral selection logic.

Interrupt Controller

Interrupt Requests

WM8505 has two levels of interrupt controller. The two interrupt controllers can support 11 level / edge-configurable external interrupt requests and 99 internal interrupt requests for a total of 110 interrupt requests.

External Interrupt Requests

The type of external interrupt can be configured in the GPIO Interrupt Request Type register (GPIO 0300) within the General Purpose I/O logic. The possible interrupt types are active-low, active-high, falling edge and rising edge. Sampling logic is used to detect falling and rising edges of these external interrupts when those types of interrupts are configured. This sampling logic runs off the internal Advanced High-performance Bus (AHB) clock and requires the external interrupt request signal to be sampled twice low then twice high to detect a rising edge (or twice high then twice low to detect a falling edge), which means that signals associated with edge sensitive interrupt requests must meet these pulse time requirements.

Interrupt Request Routing

The following tables indicates how the 64 interrupt requests are connected to the WM8505's internal Interrupt Controller:

Table 20 - Interrupt Controller 0 Request Sources

Req	Source	Req	Source	Req	Source	Req	Source
0	UHC FS	16	APB Bridge	32	UART 0	48	RTC Interrupt
1	UHC HS	17	DMA4 Channel 0	33	UART 1	49	RTC Second/Minute Update Interrupt
2	UDC DMA	18	I^2C 1	34	DMA4 Channel 2	50	UART 3
3	- Reserved -	19	I^2C 0	35	I^2S	51	DMA4 Channel 7
4	Mouse	20	SD/MMC Controller	36	PMC OS Timer 0	52	External Interrupt 5
5	UDC	21	SD/MMC Controller DMA	37	PMC OS Timer 1	53	External Interrupt 6
6	GPIO External Interrupt 0	22	PMC Wakeup	38	PMC OS Timer 2	54	External Interrupt 7
7	GPIO External Interrupt 1	23	Keyboard	39	PMC OS Timer 3	55	CIR
8	Keypad	24	SPI 0	40	DMA4 Channel 3	56	IRQ 0
9	VDMA	25	SPI 1	41	DMA4 Channel 4	57	IRQ 1
10	Ethernet MAC	26	SPI 2	42	AC97	58	IRQ 2
11	- Reserved -	27	DMA4 Channel 1	43	- Reserved -	59	IRQ 3
12	- Reserved -	28	NAND Flash Controller	44	NOR Flash Controller	60	IRQ 4
13	GPIO External Interrupt 2	29	NAND Flash Controller DMA	45	DMA4 Channel 5	61	IRQ 5
14	GPIO External Interrupt 3	30	UART 5	46	DMA4 Channel 6	62	IRQ 6
15	GPIO External Interrupt 4	31	UART 4	47	UART 2	63	IRQ 7

Table 21 - Interrupt Controller 1 Request Sources

Req	Source	Req	Source	Req	Source	Req	Source
0	- Reserved -	16	VPP	32	DMA4 Channel 12	48	GOVR SDSCD
1	JPEG Decoder	17	VID	33	DMA4 Channel 13	49	GOVR SDMIF
2	SAE	18	SPU	34	DMA4 Channel 14	50	GOVR HDMIF
3	- Reserved -	19	PIP Err	35	DMA4 Channel 15	51	GOVR HDSCD
4	- Reserved -	20	GOV	36	- Reserved -	52	- Reserved -
5	- Reserved -	21	GE	37	- Reserved -	53	- Reserved -
6	- Reserved -	22	DVO	38	- Reserved -	54	- Reserved -
7	- Reserved -	23	- Reserved -	39	- Reserved -	55	- Reserved -
8	- Reserved -	24	- Reserved -	40	- Reserved -	56	- Reserved -
9	- Reserved -	25	- Reserved -	41	- Reserved -	57	- Reserved -
10	- Reserved -	26	- Reserved -	42	- Reserved -	58	- Reserved -
11	- Reserved -	27	- Reserved -	43	- Reserved -	59	- Reserved -
12	- Reserved -	28	DMA4 Channel 8	44	- Reserved -	60	- Reserved -
13	- Reserved -	29	DMA4 Channel 9	45	- Reserved -	61	- Reserved -
14	- Reserved -	30	DMA4 Channel 10	46	- Reserved -	62	- Reserved -
15	VPU	31	DMA4 Channel 11	47	GOVW	63	- Reserved -

From the interrupt requests listed above, the WM8505’s Interrupt Controller may generate one of two interrupt requests directed to the ARM RISC processor: ARM Normal Interrupt Request or ARM Fast Interrupt Request.

From these interrupt requests, the WM8505’s Interrupt Controller may also generate one of four interrupt requests directed to the DSP: DSP Interrupt Request 0, DSP Interrupt Request 1, DSP Interrupt Request 2 or DSP Interrupt Request 3.

UART

UART Clock & Baud Rates

The UART uses the APB clock divided by URCLK_DIV as the main clock source, named UART_CLK. URCLK_DIV is a 4-bit value and should be programmed in the very beginning of the software flow. Inside the UART module, the UART clock should be always 12 MHz and must be at least 2 times slower than the APB clock (in other words, the APB clock can never be slower than 24 MHz (2x UART_CLK) and must be an integer multiple of 12 MHz.). The equation is as follows:

URCLK_DIV = [APB_CLK / UART_CLK (12 MHz)] – 1

The UART module also provides a baud rate divisor register that can be used to generate a wide range of baud rates. The main clock (UART_CLK) is further divided by the 10-bit baud rate divisor register (BRD). The 13x over sampling in the equation helps to avoid problems with mismatched clocks on each end of the serial channels.

Serial Data Transfer

The UARTs are designed to communicate data in a serial fashion over external RS-232A interfaces.

A serial data transfer cycle is shown in Figure 17 below. This diagram illustrates the relationship between the baud rate clock output from the Baud Rate Generator and the serial bits. Each bit in the transfer takes 13 "Baud Rate Clock" cycles. This 13x over-sampling helps avoid problems with mismatched clocks on each end of the serial channel. The Baud Rate Clock for each UART may be programmed using the Baud Rate Divisor register (offset 8 of each UART register block)

The transfer begins when the serial data line transitions from one to zero, followed by 7 or 8 data bits, an optional parity bit, and then one or two stop bits. The one-to-zero transition marks the reference starting point and thus the relative capture point for all received bits.

The data length can be programmed to either 7 or 8 bits, the number of stop bits can be programmed to 1 or 2 and parity can be even, odd or disabled (no parity bit).

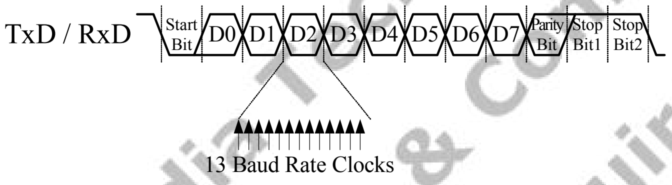


Figure 17 - UART Serial Data Transfer Format

Break Signal Generation

Each UART is capable of transmitting a break sequence on its transmit out pin. A break sequence is simply an unusually wide active-low pulse on the transmit serial data line that is longer than the transmission time for one data byte (including parity and stop bits) at the current baud rate setting.

The falling edge of the pulse is interpreted by the receiver as a start bit, all data and parity bits are then seen as 0, but when the stop bits are expected, the line will still be low resulting in a framing error on the receiving end.

This framing error condition is easily detected on the receiving end so may be used by external logic for special purposes, such as reset of an external Bluetooth module in the case where host software detects a disconnect from the Bluetooth module and wishes to re-establish communications.

Each UART allows the length of the break sequence to be programmed via the Break Count Value register. The setting of that register may be calculated by multiplying the baud rate by 0.004096.

Baud Rate Generation

The formula for calculating the baud rate of each UART is

Baud Rate = 12 M / (13 * (BRD + 1))

Where "12 M" is the assumed "UART Input Clock" frequency (M is MHz), "13" is the over-sampling rate, and "BRD" is the value programmed into the Baud Rate Divisor register. In words, the baud rate is equal to the input clock frequency divided by 13 times the BRD register value plus one. Using this formula, some example baud rate settings are shown in the table below:

Table 22 - UART Baud Rate Examples

UART Clock	Target Baud Rate	Divisor (BRD) Decimal	Actual Baud Rate	Error (%)
12 MHz	28,800	31	28,846	+0.16%
12 MHz	57,600	15	57,692	+0.16%
12 MHz	115,200	7	115,385	+0.16%
12 MHz	230,400	3	230,770	+0.16%
12 MHz	460,800	1	461,538	+0.16%
12 MHz	921,600	0	923,075	+0.16%

The input clock for the UART "n" logic may be selected by Power Management Controller "Miscellaneous Clock Select" register (PMC offset 0058) bit "n" as either 24 MHz (default) or as the ARM clock. This "UART Input Clock" is then divided by 2 to get the "UART Clock" which is used to determine the baud rate (resulting in the 12 MHz "UART Clock" frequency used in the equation above).

Using the 24 MHz default UART Input Clock, the max. baud rate supported is 921.6k in UART mode and 115.2k in IrDA mode. The UART logic supports higher baud rates by selecting the ARM clock as the UART Input Clock, however, the maximum UART Input Clock rate supported is 60 MHz, so in this case the max ARM clock rate programmed must be 60 MHz max. Also, one other factor is that the APB bus clock rate must be at least 2 times the UART input clock frequency (4 times the UART clock).

IrDA

The IrDA capability of UART4 is based on IrDA 1.0 specifications which supports baud rates from 2.4k to 115.2k bps. In this mode, optical output is active for a low level output and inactive for a high level output, resulting in short optical pulses. The length of the pulses is 3/13 of the baud rate selected. At a 115.2k baud rate, the minimum pulse width is 2.0 μs which is greater than the 1.6 μs requirement of the IrDA 1.0 specification.

FIFO Access & DMA

The UART module can function in two modes, Register mode or FIFO mode.

Register (Non-FIFO) Mode

The URTDR register (Offset 00h) is the transmission data write address and the data size is 8-bit (access can be 8/16/32). The URRDR register (Offset 04h) is the read address and the data size is 10-bit (access can be 16/32). In this mode, DMA should not be enabled. Only the CPU can communicate with the UART in this mode.

FIFO Mode

The address range from 30h to 3Fh is assigned to the Tx FIFO and the data size is 8-bit (access must be 8-bit write). The address range from 40h to 5Fh is allocated to the Rx FIFO and the data size can be 8-bit or 10-bit (access must be 8-bit read in DMA mode or 16-bit read in Non-DMA mode).

Non-FIFO Operation Mode

In this mode, the UART / IrDA operate without FIFOs. The following registers should be used:

1. **URTDR (Offset 00h).**

   Data to be transmitted should be written to this register. Access can be 8/16/32 and the lowest 8 bits are data.

2. **URRDR (Offset 04h).**

   Data received will be stored in this register. Software has the option to detect parity / frame errors in this register. Hence, Bit[8] and Bit[9] of URIER can be disabled. Access can be 16/32 and software has the choice to discard bits 9-8.

3. **URIER (Offset 14h).**

   Bit[1:0] must be enabled. Bit[5:2] must be disabled. Bit[9:8] are optional.

FIFO w/o DMA Operation Mode

This configuration requires the CPU to communicate with the UART. The Write data size is still 8-bit (byte), but the Read data size will be 10-bit (access must be 16-bit read). The extra two bits are frame error and parity error bits.

Read data presented on the system data bus will be 2 replicas of the same 16-bit data. In this mode, the following registers should be used.

1. TXFIFOs (30h~3Fh). Access must be 8-bit only. The burst type can be 4/8/16.
   - Burst of 4: start address can be 30h / 34h / 38h / 3Ch
   - Burst of 8: start address can be 30h / 38h
2. RXFIFOs (40h~5Fh). Access must be 16-bit only. The burst type can be 4/8/16. Data includes parity / frame errors.
   - Burst of 4: start address can be 40h / 48h / 50h / 58h
   - Burst of 8: start address can be 40h / 50h
3. URIER (offset 14h): Bits 1-0 must be disabled. Bits 5-2 are application dependent. Bits 9-8 are optional.

FIFO w/ DMA Operation Mode

This is the recommended mode for a high efficiency transmission. The DMA should be programmed to perform burst write of size byte(access must be 8-bit write) to TX FIFO and burst read of size byte(access must be 8-bit read) from RX FIFO.

In this mode, the following registers should be used.

1. TXFIFOs (30h~3Fh): Access must be 8-bit only. Burst type can be 4/8/16.
   - Burst of 4: start address can be 30h/34h/38h/3Ch
   - Burst of 8: start address can be 30h/38h
   - Burst of 16: start address can only be 30h (WM8505 DMAC does not support this type of burst.)
2. RXFIFOs (40h~5Fh): Access must be 8-bit only. Burst type can be 4/8/16. Data do NOT include parity / frame errors.
   - Burst of 4: start address can be 40h / 44h / 48h / 4Ch / 50h / 54h / 58h / 5Ch
   - Burst of 8: start address can be 40h / 48h / 50h / 58h
   - Burst of 16: start address can only be 40h / 50h (WM8505 DMA Controller does not support this type of burst.)
3. URIER(14h): Bit[1:0] and Bit[4] / Bit[2] must be disabled. Bit[9:8] must be enabled. Bit[5]/Bit[3] are optional.

FIFO Threshold & DMA Burst

To perform correctly and efficiently, the DMA burst length should be set to the FIFO threshold.

1. DMA Burst of 4. The threshold value for the Tx FIFO can be 12, 8, or 4. Any number above 12 is not recommended. The threshold value for the Rx FIFO should be 4 – the same as the burst length.
2. DMA Burst of 8. The threshold value for the Tx FIFO can be 8 or 4. Any number above 8 is not recommended. The threshold value for the Rx FIFO should be 8 – the same as the burst length.

Note:

The threshold definition will vary in different implementations - the Tx FIFO threshold defined in this module to be the number of valid data values in the Tx FIFO, the same definition applies for the Rx FIFO. In other implementations, the Tx FIFO threshold will represent the number of invalid data values in the Tx FIFO.

DMA Request and Threshold Interrupt

In non-DMA mode with the FIFO enabled, threshold interrupts are asserted only when the counter value passes across the threshold value in either decreasing or increasing directions. This type of interrupt source will be active for only one APB bus cycle and latched by the interrupt status register. Be aware that this behavior is different than the DMA request signals, DMA_Tx_Req and DMA_Rx_Req.

This module has two DMA request signals, DMA_Tx_Req and DMA_Rx_Req. In DMA mode, the entry counter in the Tx FIFO will assert the DMA_Tx_Req signal to the DMA engine for write requests whenever the counter value is under the threshold. The entry counter in the Rx FIFO will assert the DMA_Rx_Req signal to the DMA engine for Read requests whenever the counter value is over the threshold value. In this configuration, the threshold interrupt status bits, URISR[4] and URISR[2], will still be updated only when the counter value passes across the threshold value in either increasing or decreasing directions.

Trailing Data

Two scenarios that cause trailing data to be left in the Rx FIFO are discussed here:

Transfer data size is unknown:

DMA burst should be set to either 4 or 8 and the UART module receive-timeout interrupt (URIER it-11) enabled to assist the software to detect end of transfer. The following steps are taken to flush and complete the transfer:

- When the receive timeout interrupt is asserted, the software terminates the DMA channel associated with the Rx FIFO and records the DMA current accumulated destination address to where the trailing data should be written.
- Read the Rx FIFO index register to determine how many data values are left in the Rx FIFO and flush / read them out with proper read access.

Transfer data size is known:

DMA burst should be set to the proper length based on the transfer size (see note below) and the ART module receive-timeout interrupt (URIER bit-11) enabled to assist software to detect end of transfer. The following steps are taken to flush and complete the transfer:

- When the receive timeout interrupt is asserted, the software reads the DMA current accumulated destination address to where the trailing data should be written.
- Read the Rx FIFO index register to determine how many data values are left in the Rx FIFO and flush / read them out with proper read access.

Note:

For example, assuming 1027 bytes for transfer, 1024 bytes can be transferred by DMA with a burst of 4 and 256 counts or with a burst of 8 and 128 counts. The trailing 3 bytes may then be read by the CPU with 3 single accesses.

Overflow / Underflow

In FIFO receive mode, after the FIFO is full, the newly received characters written to it are ignored and the "Receive Data Overrun" status (Bit[7]) is set in the Interrupt Status register (ISR) (Offset 0018h). If the receive FIFO is read after it is empty, invalid data will be returned.

In FIFO transmit mode, if the FIFO is written when it is full, the additional characters written to it are ignored and the "Transmit Data Overrun" status (Bit[6]) is set in the ISR.

In Register (Non-FIFO) mode, after the Receive Data register (Offset 0004h) is full, the newly received characters written to it overwrite the previous data and the "Receive Data Overrun" status (Bit[7]) is set in the ISR. If the Receive Data register is read after it is empty, invalid data will be returned.

In Register (Non-FIFO) mode, if the Transmit Data register (Offset 0000h) is written when it is full, the additional characters written to it overwrite the previous data and the "Transmit Data Overrun" status (Bit[6]) is set in the ISR.

Interrupts

Two types of interrupt events are provided: pulse and level. All pulse type events are captured at their rising edge and converted to a level interrupt which will be cleared when the Interrupt Status register (ISR) is read. All level type interrupts will not be cleared by reading the ISR and instead will be cleared when the interrupt sources have been cleared.

Level interrupts include "Tx Data Register Empty" and "Rx Data Register Full" for Register mode plus "Tx FIFO Empty", "Rx FIFO Full" and "Rx FIFO Timeout" for FIFO mode.

Pulse interrupts include "Tx FIFO Almost Empty" and "Rx FIFO Almost Full" (FIFO mode), "Tx Data Under-run", "Rx Data Overrun", "Rx Parity Error", "Rx Framing Error", "CTS Change" (either 0-to-1 or 1-to-0) and "Break Done".

All events are qualified by their corresponding bit in the Interrupt Enable register. All events are OR'd together to create a single interrupt output for that UART module (see Table 20.).

OS Timers and Watchdog Timer

The WM8505 provides four OS Timers, one of which may be configured as a Watchdog Timer.

There is a single 32-bit counter that increments by 1 for each 3 MHz clock period. The output count value of this counter is then compared to the four OS Timer Match Register values when they are enabled. When a match occurs with an active OS Timer, its Match Status bit will be set in the OS Timer Status register (PMC 0114) which will generate an interrupt request if enabled via the OS Timer Interrupt Enable Register (PMC 011C).

OS Timer 0 may also be enabled as a Watchdog Timer via the OS Timer Watchdog Enable register (PMC 0118). When enabled as the Watchdog Timer, a reset will be generated instead of an interrupt request when OS Timer Match register 0 (PMC 0100) matches the counter value.

OS Timer Typical Usage

Typically the OS Timer Match Registers (PMC 0100 - PMC 010C) will be configured to generate interrupt requests at desired time intervals. A 32-bit counter running at 3 MHz provides a time range up to 23.85 minutes.

At some point, the software typically writes a new value to the 32-bit counter to restart the count value. Typically, this new count value will be all zeros. This is accomplished by writing the new 32-bit value to the OS Timer Count Register (PMC 0110). After this the 32-bit counter will again count up, generating the desired interrupt requests as the count value matches the match registers.

OS Timer Count Register Reads

To read the present 32-bit counter value, the logic must first synchronize that value to the bus interface. To correctly read the 32-bit counter value, the software should perform the following sequence:

1. Write a one to OS Timer Control register (PMC 0120) Bit[1].
2. Poll OS Timer Access Status register (PMC 0124) Bit[5] until it is zero.
3. Read the 32-bit counter value by reading the OS Timer Count Register (PMC 0110).

I^{2}C

The I^{2}C Controller is used to communicate with slow devices in a serial fashion over an I^{2}C bus. The I^{2}C Controller is the only master on the I^{2}C bus system in this design. The WM8505 can control and communicate with external LCD and EEPROM devices by way of the I^{2}C Controller.

A transfer on the I^{2}C bus begins with a START condition and ends with a STOP or a repeated START condition (a repeated START condition indicates a new transfer is beginning). Between the START (or repeated START) and STOP (or repeated START), it consists of one address byte and one or more data bytes with an acknowledge bit after each byte. See Figure 18 for details.

As the only master on the I^{2}C bus, the Controller is responsible for initiating and controlling all transfers. Slave devices can slow down transfers by driving and holding the SCL line low to force the controller to wait. All devices only drive the I^{2}C SCL and SDA lines low, never high, with the signal level only returning high as a result of external pull-up resistors when SCL or SDA is undriven (i.e., all devices drive SCL and SDA with "open-drain" outputs). As a result, either master or slave (or both) can control the SCL clock by driving it low to pause a transfer. The transfer continues only after both master and slave release the SCL line. This mechanism requires only 2 pins and is simple to use and to implement (therefore reducing cost of implementation) but the external pull-up resistor value limits the signal rise time and therefore the maximum data transfer rate. The I^{2}C interface is therefore used primarily for controlling low-speed devices (e.g., external on-board system power control chips, small EEPROMs, device identification, etc).

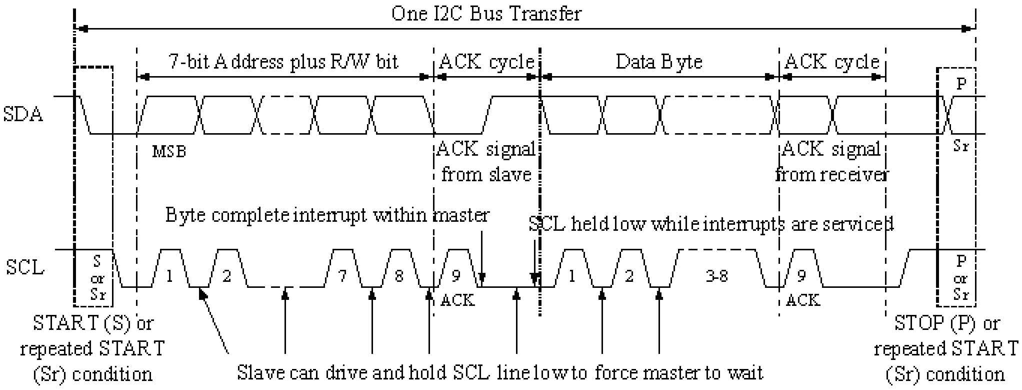

One I2C Bus Transfer
7-bit Address plus R/W bit
ACK cycle
Data Byte
ACK cycle
SDA
MSB
ACK signal from slave
ACK signal from receiver
P
Sr
Byte complete interrupt within master
SCL held low while interrupts are serviced
SCL
S or Sr
1
2
7
8
9
ACK
1
2
3-8
9
ACK
P or Sr
START (S) or repeated START (Sr) condition
Slave can drive and hold SCL line low to force master to wait
STOP (P) or repeated START (Sr) condition

Figure 18 - I^{2}C Bus Byte Transfer

Electrical Specification

Storage and Operation Temperature

Symbol	Parameter	Min	Max	Unit	Note
T_S	Storage / Ambient Temperature	–55	125	ºC	
T_C	Case Operating Temperature	0	85	ºC	

DC Characteristics

Voltage Requirements

Symbol	Parameter	Min	Max	Unit	Note
V_{DD}	Power Supply Voltage – Core	1.425	1.575	Volt	1.5V±5%
V_{CC33}	Power Supply Voltage – I/O	3.135	3.465	Volt	3.3V±5%
V_{CCMEM1}	Power Supply Voltage – Memory (DDR SDRAM)	2.375	2.625	Volt	2.5V±5%
V_{CCMEM2}	Power Supply Voltage – Memory (DDR2 SDRAM)	1.71	1.89	Volt	1.8V±5%
V_{MVREF1}	Power Supply Voltage – Memory Reference (DDR)	1.225	1.275	Volt	½ V_{CCMEM}±2%
V_{MVREF2}	Power Supply Voltage – Memory Reference (DDR2)	0.882	0.918	Volt	½ V_{CCMEM}±2%
V_{SUS15}	Power Supply Voltage – Suspend	1.425	1.575	Volt	1.5V±5%
V_{SUS33}	Power Supply Voltage – Suspend	3.135	3.465	Volt	3.3V±5%
V_{BAT}	Power Supply Voltage – RTC Battery	3.135	3.465	Volt	3.3V±5%
$V_{CC33USB}$	Power Supply Voltage – USB Differential Output	3.135	3.465	Volt	3.3V±5%
$V_{CCA33USBPLL}$	Power Supply Voltage – USB PLL Analog	3.135	3.465	Volt	3.3V±5%
$V_{CCA15USBPLL}$	Power Supply Voltage – USB PLL Analog	1.425	1.575	Volt	1.5V±5%
$V_{CCA33PLL}$	Power Supply Voltage – Primary PLL Analog	3.135	3.465	Volt	3.3V ±5%
$V_{CCA33PLLA}$	Power Supply Voltage – Secondary PLL Analog	3.135	3.465	Volt	3.3V ±5%
$V_{CCA33PLLB}$	Power Supply Voltage – Secondary PLL Analog	3.135	3.465	Volt	3.3V ±5%
$V_{CCA33PLLC}$	Power Supply Voltage – Secondary PLL Analog	3.135	3.465	Volt	3.3V ±5%
$V_{CCA33XTAL}$	Power Supply Voltage – Crystal/OSC	3.135	3.465	Volt	3.3V ±5%
$V_{CCA33PLL}$	Power Supply Voltage – Primary PLL Analog	3.135	3.465	Volt	3.3V ±5%

Current / Power Requirements

Power Consumption Summary Table

Mode	Test Condition	Power Consumption (mW)
Operation Mode	Linux Kernel Normal Mode	1147.30
	WiFi Projection Mode	1343.14
	Heavy Loading Mode	1652.96
Sleep Mode		441.17
Suspend Mode		7.27

Note:
The data above is measured in room temperature with random sampling.

Measurement Conditions

1. Linux Kernel Normal Mode: VGA DAC Enable
2. WiFi Projection Mode: DVI+ USB+ Ethernet+ I^2S
3. Heavy Loading Mode: DVI+ VGA+ USBx2+ I^2S+ JPEG Decoder
4. Sleep Mode
5. Suspend Mode

Power Consumption in Test Conditions

Linux Kernel Normal Mode

Power Pins	Source (V)	Current (mA/avg.)	Power (mW/avg.)
VDD	1.50	435.00	653.28
VCC33	3.30	21.14	69.84
VCCMEM	1.85	40.88	75.47
VCCA33DAC	3.30	67.27	222.07
VCC33USB	3.30	2.78	9.17
VCCA33USBPLL	3.30	5.41	17.84
VCCA15USBPLL	1.50	4.94	7.40
VCCA33XTAL	3.30	0.85	2.80
VCCA33PLL	3.30	13.96	46.12
1P5VSUS	1.50	4.55	6.82
3P3VSUS	3.29	10.70	35.27
Total Power (avg.)			**1147.30**

WiFi Projection Mode

Power Pins	Source (V)	Current (mA/avg.)	Power (mW/avg.)
VDD	1.50	580.84	870.41
VCC33	3.30	33.73	111.41
VCCMEM	1.83	66.84	122.29
VCCA33DAC	3.31	0.10	0.33
VCC33USB	3.29	24.69	81.27
VCCA33USBPLL	3.29	5.47	18.01
VCCA15USBPLL	1.50	6.42	9.63
VCCA33XTAL	3.31	0.76	2.51
VCCA33PLL	3.30	15.78	52.14
1P5VSUS	1.50	12.29	18.43
3P3VSUS	3.29	10.60	34.88
Total Power (avg)			**1343.14**

Heavy Loading Mode

Power Pins	Source (V)	Current (mA/avg)	Power (mW/avg)
VDD	1.50	594.83	891.37
VCC33	3.30	32.73	108.06
VCCMEM	1.83	70.24	128.44
VCCA33DAC	3.30	66.96	220.95
VCC33USB	3.28	45.27	148.71
VCCA33USBPLL	3.29	5.56	18.28
VCCA15USBPLL	1.50	4.99	7.49
VCCA33XTAL	3.30	0.73	2.42
VCCA33PLL	3.30	15.87	52.40
1P5VSUS	1.50	12.49	18.74
3P3VSUS	3.29	10.41	34.23
Total Power (avg.)			**1652.96**

Sleep Mode

Power Pins	Source (V)	Current (mA/avg.)	Power (mW/avg.)
VDD	1.51	17.60	26.63
VCC33	3.31	14.49	47.94
VCCMEM	1.87	11.20	20.97
VCCA33DAC	3.31	67.59	223.40
VCC33USB	3.29	4.40	14.48
VCCA33USBPLL	3.29	5.21	17.15
VCCA15USBPLL	1.50	3.52	5.29
VCCA33XTAL	3.31	0.91	3.01
VCCA33PLL	3.31	11.77	38.92
1P5VSUS	1.50	2.43	3.65
3P3VSUS	3.29	10.63	34.99
Total Power (avg.)			**441.17**

Suspend Mode

Power Pins	Source (V)	Current (mA/avg.)	Power (mW/avg.)
VDD	0.00	0.00	0.00
VCC33	0.00	0.00	0.00
VCCMEM	1.87	0.18	0.34
VCCA33DAC	0.00	0.00	0.00
VCC33USB	3.30	0.11	0.36
VCCA33USBPLL	3.30	0.03	0.11
VCCA15USBPLL	1.50	0.09	0.13
VCCA33XTAL	0.00	0.00	0.00
VCCA33PLL	0.00	0.00	0.00
1P5VSUS	1.50	0.44	0.66
3P3VSUS	3.30	1.57	5.17
Total Power (avg.)			**7.27**

AC Characteristics

VGA DAC Interface Timing

Operating Conditions

Parameter	Min	Typ	Max	Unit
Power Supply Voltage	3	3.3	3.6	V
Output Load Resistance		37.5		Ω
Output Load Capacitance		10		pF
Ambient Temperature	0	60	110	°C

Electrical Characteristics

Parameter	Min	Typ	Max	Unit
Power Dissipation (Power ON)		235		mW
Clock Input				
Input Capacitance (C_{CLK})		1		pF
Analog Outputs				
Output Current		18.67		mA
Output Compliance Range		1.3		V
Voltage Reference				
Band-gap Reference Voltage		1.124		V
Sense Reference Voltage		300		mV

DC Performance

Parameter	Min	Typ	Max	Unit
Resolution		8		Bits
Differential Non-Linearity (DNL)		±1		LSB
Integral Non-Linearity (INL)		±1		LSB
Offset Error		1		%FS
Gain Error		1		%FS
Full-scale Output Current		18.67		mA
Power Supply Rejection Ratio		3		%

AC Performance

Parameter	Min	Typ	Max	Unit
Analog Outputs				
Glitch Energy		75		pV-sec
DAC-to-DAC Cross-talk		-90		dB
Clock Feed-through		-56		dB

Switching Characteristics

Parameter	Min	Typ	Max	Unit
Clock Input				
Update Rate, t_3		200		MHz
Clock Pulse High Time, t_4		1.65		ns
Clock Pulse Low Time, t_5		1.65		ns
Data Input				
Setup Time, t_1		0.8		ns
Hold Time, t_2		0.2		ns
Analog Outputs				
Clock to Output Delay, t_6[2]		1		ns
Rise Time, t_7[3]		1.5		ns
Fall Time, t_8[3]		1.5		ns
Settling Time, t_9[4] (Rising Edge)		12		ns
Settling Time, t_{10}[4] (Falling Edge)		12		ns
DAC-to-DAC Matching		1		%
Monitor Detection				
SENSE Output Delay, t_{11}[5]		8		ns

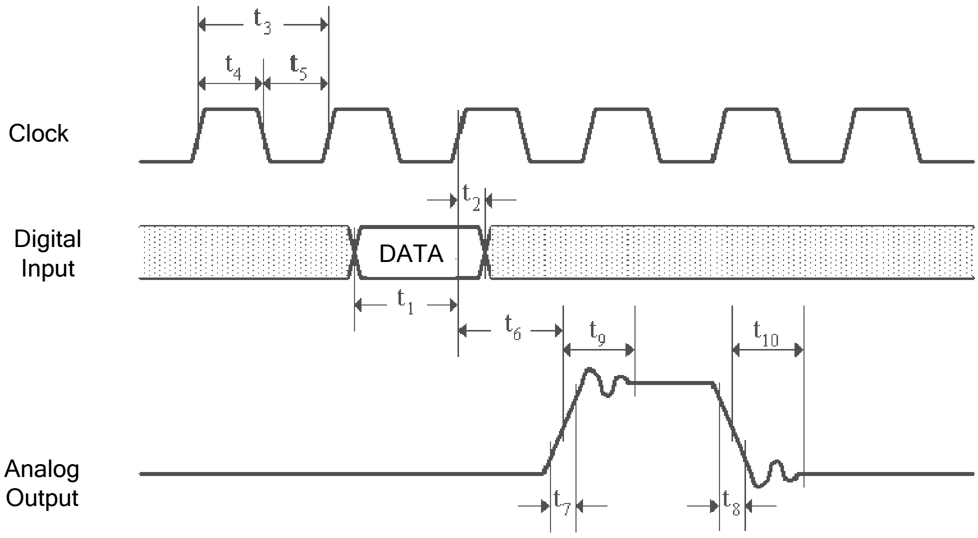


Figure 19 - DATA Timing Diagram

DRAM Interface Timing

Symbol	Parameter	Min	Typ	Max	Unit	Note
T_{Ck}	Memory Clock Period	3		8	ns	DDR2 667
T_{CL}	Memory Clock Low Pulse Width	0.48		0.52	T_{Ck}	
T_{CH}	Memory Clock High Pulse Width	0.48		0.52	T_{Ck}	
T_{IS}	Address and Control Input Setup Time	0.200			ns	
T_{IH}	Address and Control Input Hold Time	0.275			ns	
T_{DQSL}	DQS LOW Pulse Width	0.35			T_{Ck}	
T_{DQSH}	DQS HIGH Pulse Width	0.35			T_{Ck}	
T_{DQSQ} (read)	DQS-DQ Skew for DQS and Associated DQ Signals			0.240	ns	
T_{QH} (read)	DQ Output Hold Time from DQS	1.1			ns	
T_{DS} (write)	DQ and DM Input Setup Time (single-ended strobe)	0.100			ns	
T_{DH} (write)	DQ and DM Input Hold Time (single-ended strobe)	0.175			ns	

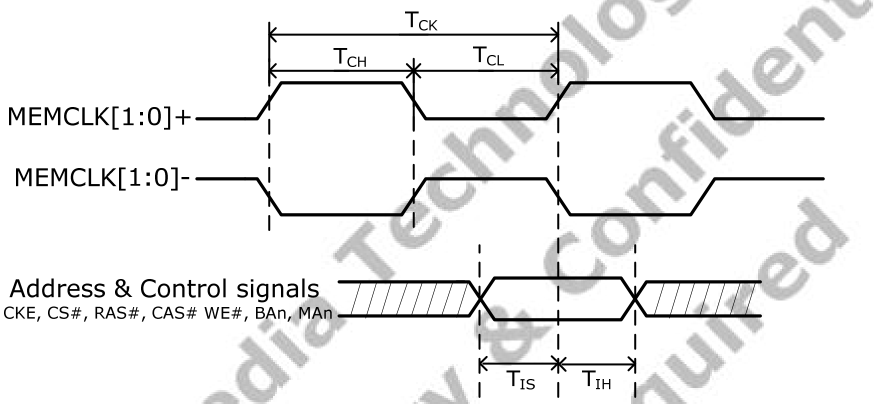


Figure 20 - Command Timing Dagram

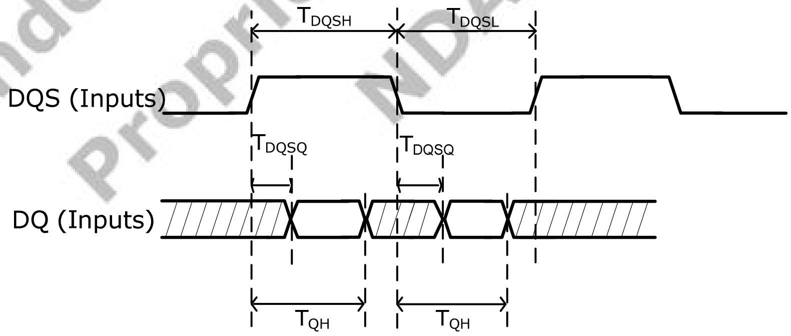


Figure 21 - DQ/DQS Read Timing Diagram

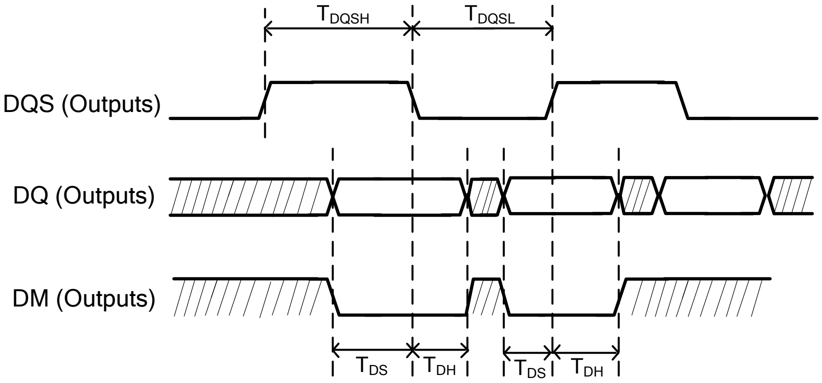


Figure 22 - DQ/DQS/DM Write Timing Diagram

Serial Flash Memory Controller Interface Timing

Symbol	Parameter	Min	Max	Unit	Note
1/ T_{CYC}	Clock Frequency – Data Transfer Mode		100	MHz	
T_{CL}	Clock Low Time	0.45*TCYC		ns	
T_{CH}	Clock High Time	0.45*TCYC		ns	
T_{CES}	SFCS# Active Setup Time	TCL -2		ns	
T_{CES1}	SFCS# Active Setup Time	1.5*TCL -2		ns	
T_{CEH}	SFCS# Active Hold Time	TCH		ns	
T_{CEH1}	SFCS# Active Hold Time	1.5*TCH		ns	
T_{DIS}	Data Input Setup Time	2		ns	
T_{DIS2}	Data Input Setup Time	2		ns	
T_{DIH}	Data Input Hold Time	1		ns	
T_{DIH2}	Data Input Hold Time	1		ns	
T_{DOD}	Data Output Delay Time		2	ns	

1. This timing is available when bit 8 (spi_cs_clk_dly) of SPI Interface Configuration Register (0x40) is one (default).
2. The timing parameter is used when using negedge of SFCLK to sample input data.

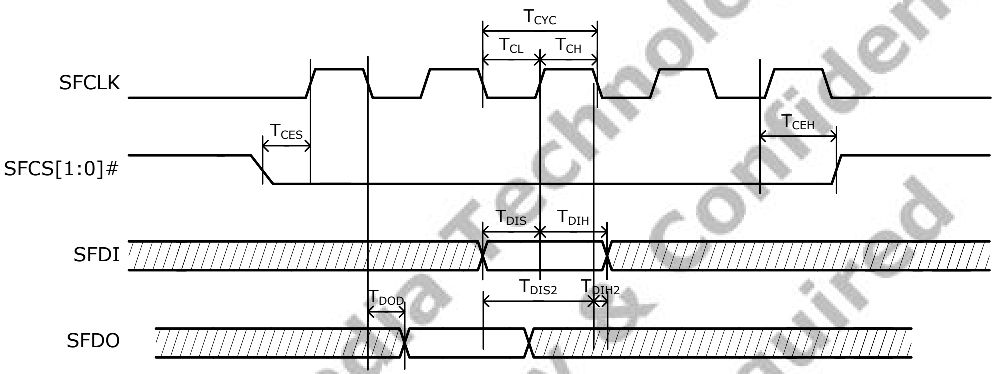


Figure 23 - Serial Flash Memory Controller Interface Timing

Ethernet MAC Interface Timing - MII Mode

Symbol	Parameter	Min	Max	Unit	Note
T_{CYC}	MIIRXCLK/MIITXCLK Cycle Time	399.9/ 39.99	400.1/ 40.01	ns	10BaseT/ 100BaseT
T_{CH}	MIIRXCLK/MIITXCLK Pulse Width High	0.45	0.55	T_{CYC}	
T_{CL}	MIIRXCLK/MIITXCLK Pulse Width Low	0.45	0.55	T_{CYC}	
T_{TVAL}	MIITXEN, MIITXD valid from MIITXCLK	0	25	ns	
T_{S_R}	MIIRXD, MIIRXDV, MIIRXERR Setup to MIIRXCLK (For Receive)	10		ns	
T_{H_R}	MIIRXD, MIIRXDV, MIIRXERR Hold from MIIRXCLK (For Receive)	10		ns	

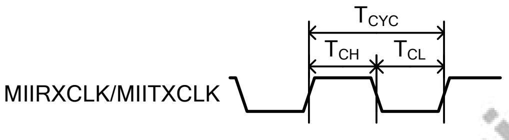


Figure 24 - Ethernet MAC Interface Timing – Clock

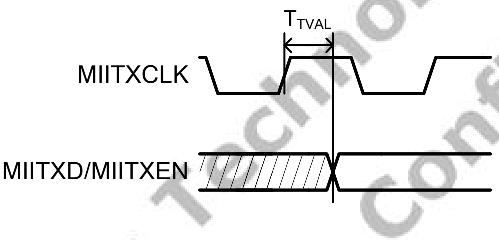


Figure 25 - Ethernet MAC Interface Timing –Transmit Timing

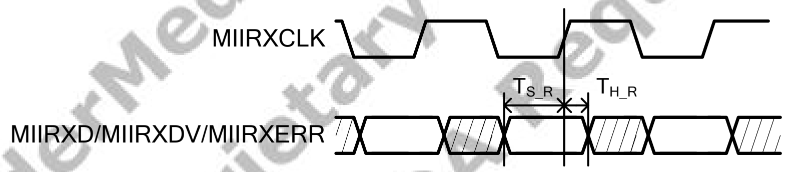


Figure 26 - Ethernet MAC Interface Timing –Setup & Hold Time

Ethernet MAC Interface Timing - RevMII Mode

Symbol	Parameter	Min	Max	Unit	Note
T_{CYC}	MIIRXCLK/MIITXCLK Cycle Time	395/ 39.5	405/ 40.5	ns	10BaseT/ 100BaseT
T_{CH}	MIIRXCLK/MIITXCLK Pulse Width High	0.40	0.60	T_{CYC}	
T_{CL}	MIIRXCLK/MIITXCLK Pulse Width Low	0.40	0.60	T_{CYC}	
T_{S_T}	MIITXD, MIITXEN Setup to MIIRXCLK (for Transmit)	10		ns	
T_{H_T}	MIITXD, MIITXEN Hold from MIIRXCLK (for Transmit)	15		ns	
T_{S_R}	MIIRXD, MIIRXDV, MIIRXERR Setup to MIITXCLK (For Receive)	10		ns	
T_{H_R}	MIIRXD, MIIRXDV, MIIRXERR Hold from MIITXCLK (For Receive)	10		ns	

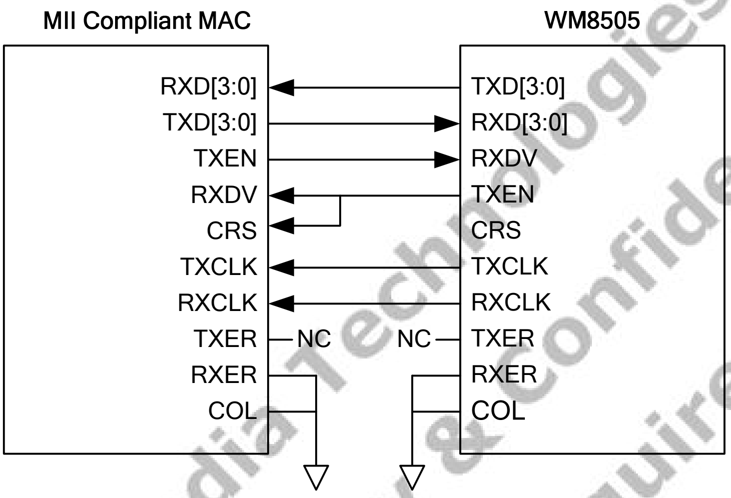


Figure 27 - RevMII Connection Diagram

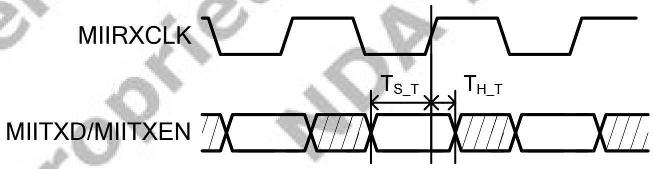


Figure 28 - Ethernet MAC Interface Timing –Setup & Hold Time

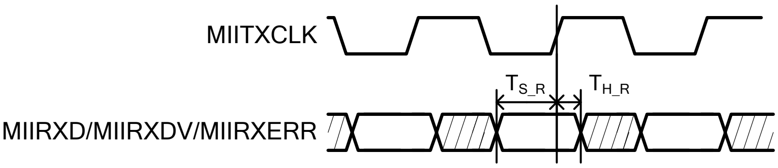


Figure 29 - Ethernet MAC Interface Timing –Setup & Hold Time

KBDC Interface Timing

Symbol	Parameter	Min	Max	Unit	Note
T1	Time for Inhibit Communication	100		μs	
T2	Time from Inhibition to Data Line Pull-down		90	μs	
T3	Time for Clock Generation		15	ms	
T4	High Level Duration of Clock	30	50	μs	
T5	Low Level Duration of Clock	30	50	μs	
T6	Data Valid after Clock Falling(SEND)		10	μs	
T7	Transmit or Receive Timeout		2	ms	
T8	KBDC Inhibit Communication for Data Processing		60	μs	
T9	Time for Waiting for Response		20	ms	
T10	Idle State Duration Time Detection (Clock stays at high)	50		μs	
T11	Time from Data Pull-down to First Clock Falling	3		μs	
T12	Data Valid before Clock Falling (RECEIVE)	3		μs	
T13	Time from Clock Rising to Data Transition	5	T4-5	μs	
T14	Time from Data Transition to Clock Falling	5	25	μs	
T15	Time after Transition until Clock Pull-down		8	μs	

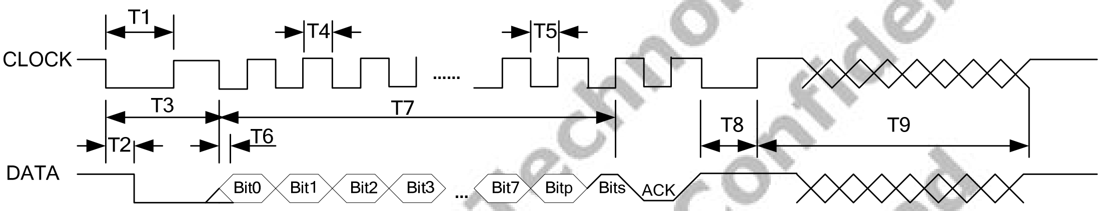


Figure 30 - KBDC Interface Timing – Sending Data to External Keyboard/mouse

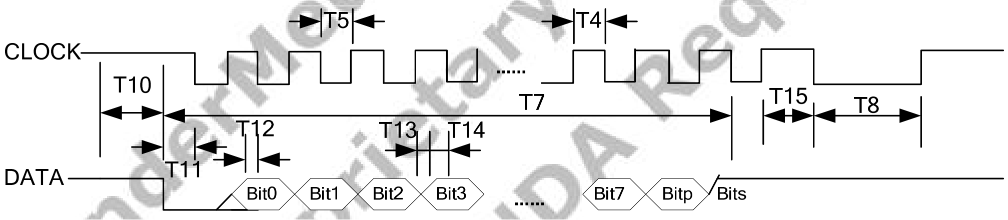


Figure 31 - KBDC Interface Timing – Reveiving Data from External Keyboard/mouse

NAND Flash Interface Timing - Command/Address Latch

Symbol	Parameter	Min	Max	Unit	Note
T_{CLS}	NANDCLE Setup Time	12	-	ns	
T_{CLH}	NANDCLE Hold Time	5	-	ns	
T_{CS}	NANDCE Setup Time	20	-	ns	
T_{CH}	NANDCE Hold Time	5	-	ns	
T_{WP}	NANDWE Pulse Width	17.5	-	ns	
T_{ALS}	NANDALE Setup Time	12	-	ns	
T_{ALH}	NANDALE Hold Time	5	-	ns	
T_{DS}	NANDIO Setup Time	12	-	ns	
T_{DH}	NANDIO Hold Time	5	-	ns	

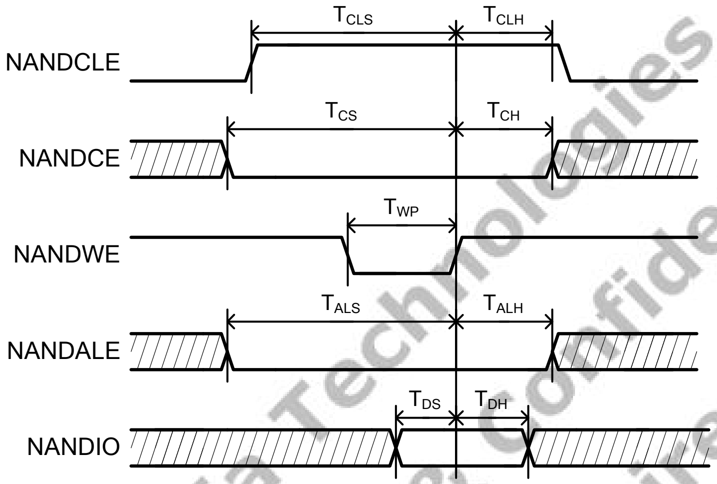


Figure 32 - NAND Flash Interface Timing – Command Latch

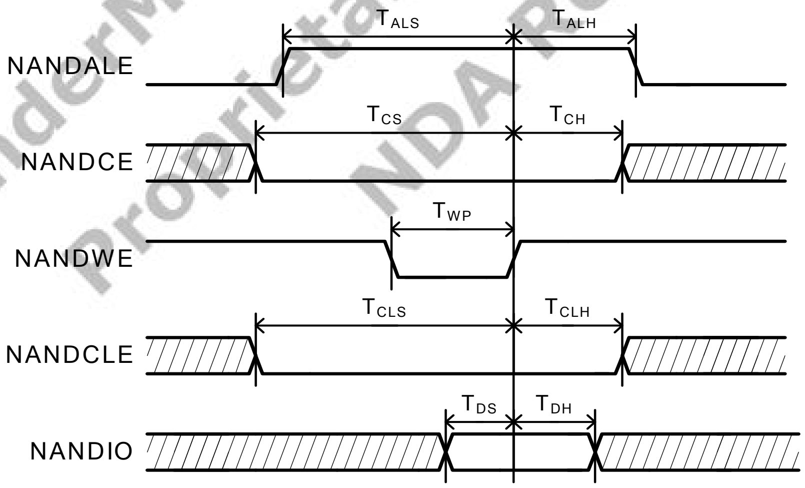


Figure 33 - NAND Flash Interface Timing – Address Latch

NAND Flash Interface Timing – Data Read

Symbol	Parameter	Min	Max	Unit	Note
T_{RC}	Read Cycle Time	34	-	ns	
T_{RP}	NANDRE Pulse Width	17	-	ns	
T_{REH}	NANDRE High Hold Time	17	-	ns	
T_{CH}	NANDCE Hold Time	5	-	ns	
T_{REA}	NANDRE Access Time	-	30	ns	
T_{OH}	NANDRE High to Output Hold	15	-	ns	
T_{CEA}	NANDCE Access Time	-	45	ns	
T_{RR}	NANDRB# to NANDRE Low	20	-	ns	

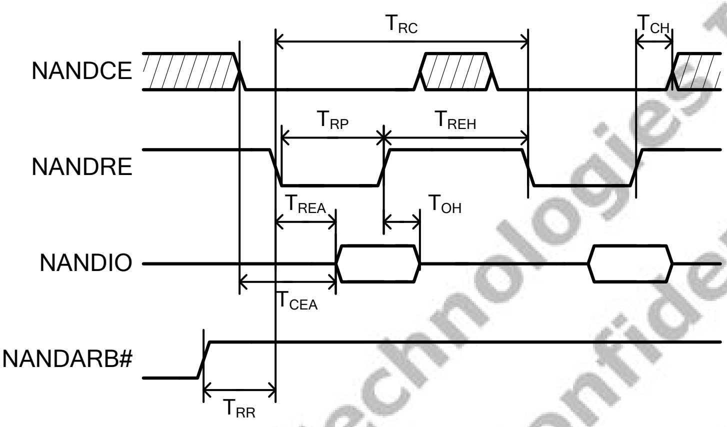


Figure 34 - NAND Flash Interface Timing – Data Read

NOR/GUP Interface Timing

Asynchronous READ Cycle Timing Requirements

Symbol	Parameter	Min	Max	Unit	Note
tCYC	Clock Period of the NOR/GuP Bus Timing Generator	16		ns	
tRITR	Read Initial Access Time	1	32	tCYC	
tPRTR	Page Access Time	1	8	tCYC	
tCETR	CE Change Turn Around Time	1	8	tCYC	
tRTWT	Read to Write Turn Around Time	1	8	tCYC	

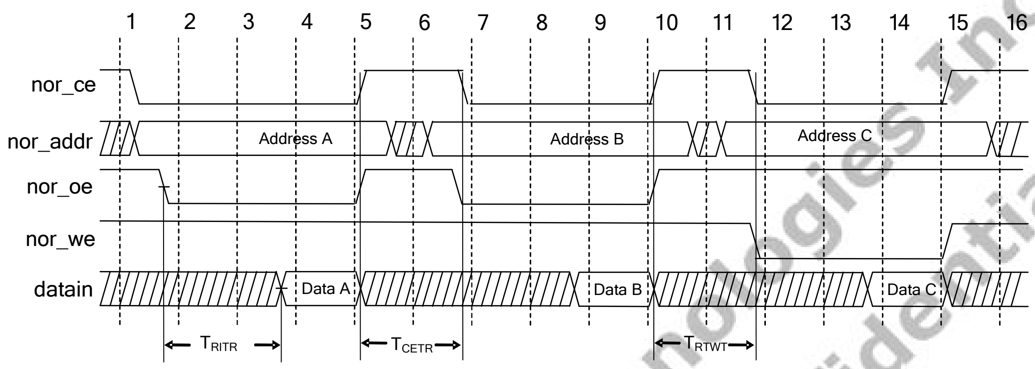


Figure 35 - Read Timing – Read Initial Cycle Timing in Asynchronous Read

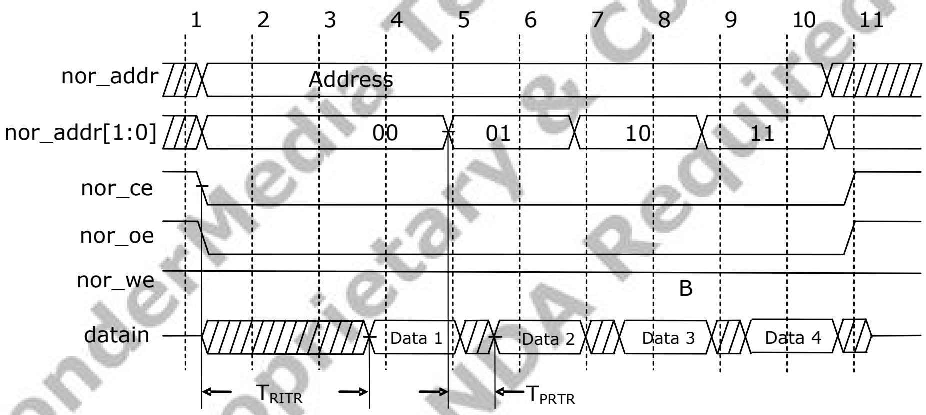


Figure 36 - Page Mode Read Timing in Asynchronous Mode

Asynchronous WRITE Cycle Timing Requirements

Symbol	Parameter	Min	Max	Unit	Note
tCYC	Clock Period of the NOR/GuP Bus Timing Generator	16		ns	Base clock
tWCTR	Write Cycle Time	1	32	tCYC	
tWPS	Write Pulse Start Time	0	16	tCYC	
tWPW	Write Pulse Width	0.5	16	tCYC	
tWRTR	Write to Other Cycle Turn-around Time	1	16	tCYC	

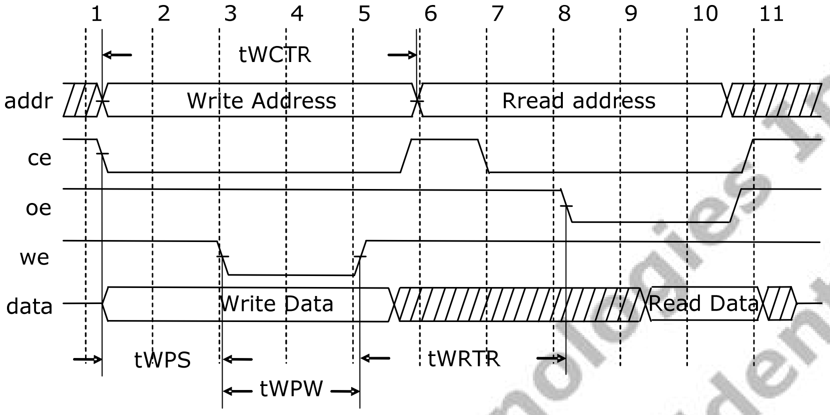


Figure 37 - Write Timing in Asynchronous Mode

Reset Timing Requirements

Symbol	Parameter	Min	Max	Unit	Note
tCYC	Clock Period of the NOR/GuP Bus Timing Generator	16		ns	Base clock
tRSTRT	The Recovery Time after Reset	1	65535	tCYC	
tRSTWP	Reset Pulse Width	1	256	tCYC	

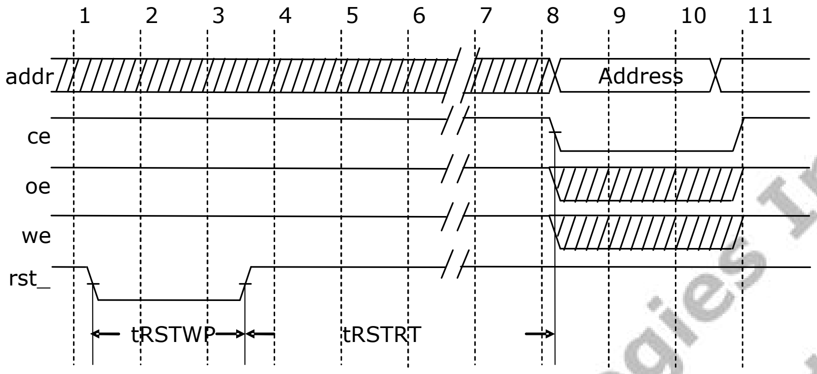


Figure 38 - Reset Timing

SD/MMC Interface Timing

Symbol	Parameter	Min	Max	Unit	Note
1/ T_{CYC}	Clock Frequency – Data Transfer Mode		25	MHz	CL ≤ 100pF (1 cards)
1/ T_{CYC}	Clock Frequency – Identification Mode†	0[1]/100	400	KHz	CL ≤ 250pF (1 cards)
T_{CL}	Clock Low Time	10		ns	CL ≤ 100pF (1 cards)
T_{CH}	Clock High Time	10		ns	CL ≤ 100pF (1 cards)
T_{CR}	Clock Rise Time		10	ns	CL ≤ 100pF (1 cards)
T_{CF}	Clock Fall Time		10	ns	CL ≤ 100pF (1 cards)
T_{DIS}	Input Setup Time (CMD, DAT)	5		ns	CL ≤ 25pF (1 card)
T_{DIH}	Input Hold Time (CMD, DAT)	5		ns	CL ≤ 25pF (1 card)
T_{DOD}	Output Delay Time during Data Transfer Mode (CMD, DAT)		14	ns	CL ≤ 25pF (1 card)
T_{DOD}	Output Delay Time during Identification Mode(CMD, DAT)		50	ns	CL ≤ 25pF (1 card)

Note:
1. 0 Hz means to stop the clock. The given minimum frequency range is for the cases where continuous clock is required.

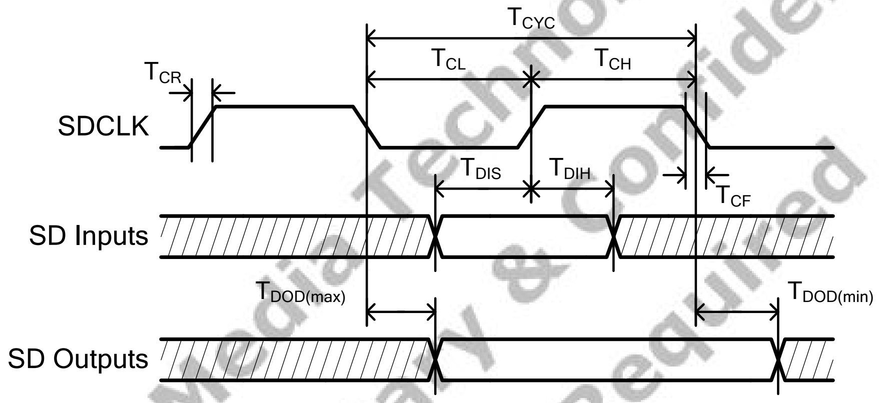


Figure 39 - SD Interface Timing – Data Read

DVO Interface Timing

Single-edge Mode

Symbol	Parameter	Min	Max	Unit	Note
T_{CYC}	DVO clock frequency	7.3		ns	Max frequency: 136.75 MHz
T_{SETUP}	Data / SYNC setup time to clock raising edge	2		ns	
T_{HOLD}	Data / SYNC hold time to clock raising edge	2		ns	

Dual-edge Mode

Symbol	Parameter	Min	Max	Unit	Note
T_{CYC}	DVO clock frequency	9.1		ns	Max frequency: 110 MHz
T_{SETUP}	Data / SYNC setup time to clock raising / falling edge	0.7		ns	
T_{HOLD}	Data / SYNC hold time to clock raising / falling edge	0.5		ns	

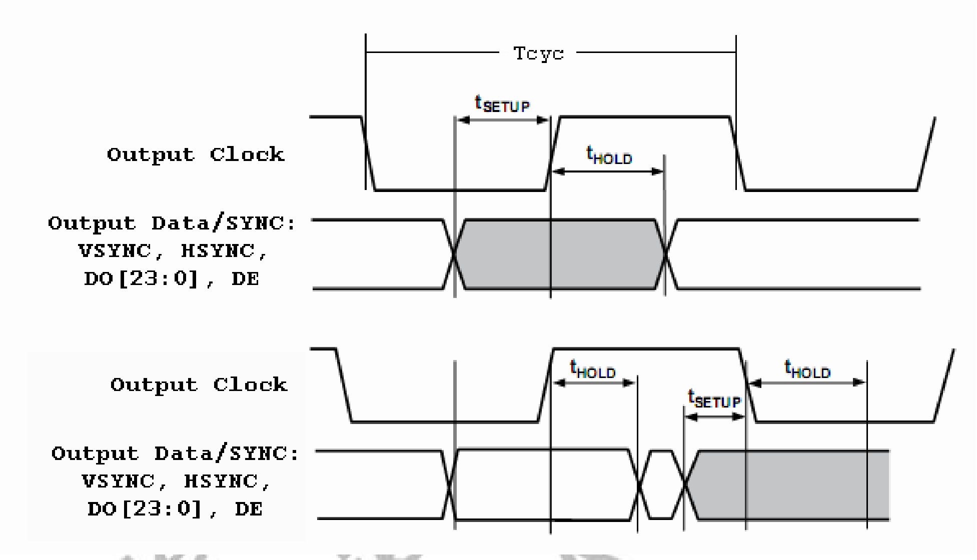


Figure 40 - DVO Interface Timing

SDTV Digital Video Interface Timing

Table 23 - CCIR-656/601 Input Pixel Data Timing

Symbol	Parameter	Min	Max	Unit	Note
T_{CYC}	VCLK Cycle Time	35	39	ns	Typical 37 ns
T_{CS}	Control Signal Setup Time	5	10	ns	
T_{CH}	Control Signal Hold Time	5		ns	
T_{DS}	Data Setup Time	5	10	ns	
T_{DH}	Data Hold Time	5		ns	

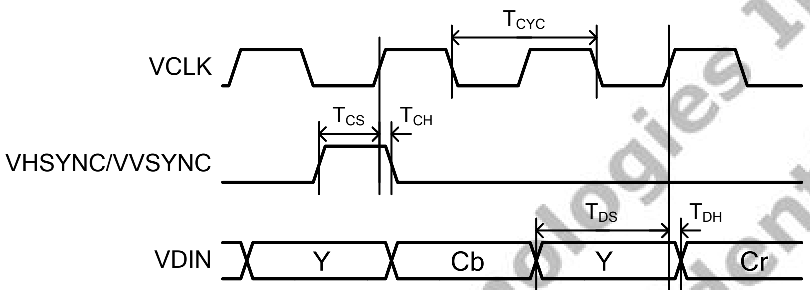


Figure 41 - CCIR-656/601 Input Pixel Data Timing

CMOS Sensor Interface Timing

Table 24 - CMOS Sensor Input Pixel Data Timing

Symbol	Parameter	Min	Max	Unit	Note
T_{CYC}	VCLK Cycle Time	13.5		ns	fmax=74 MHz
T_{CS}	Control Signal Setup Time	4		ns	
T_{CH}	Control Signal Hold Time	4		ns	
T_{DS}	Data Setup Time	4		ns	
T_{DH}	Data Hold Time	4		ns	

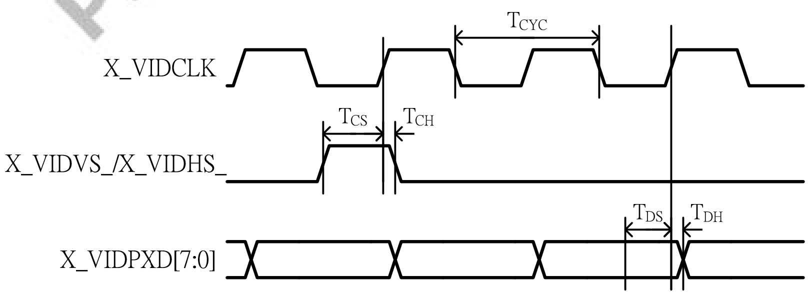


Figure 42 - CMOS Sensor Input Pixel Data Timing

Audio I^{2}S Interface Timing

Table 25 - Audio I^{2}S Interface Timing of Master Mode

Symbol	Parameter	Min	Max	Unit	Note
F_{MCKW}	Master CLK Frequency	24.576		MHz	
T_{MCKH}	Master CLK Cycle High	0.45*T_{MCKW}		ns	
T_{MCKL}	Master CLK Cycle Low	0.45*T_{MCKW}		ns	
F_{CLKW}	Bit CLK Frequency	24.576		MHz	
T_{CLKH}	Bit CLK Cycle High	0.45*T_{CLKW}		ns	
T_{CLKL}	Bit CLK Cycle Low	0.45*T_{CLKW}		ns	
T_{FDL}	Frame Sync Clock Output Delay Time		4	ns	
T_{DDL}	I2S Output Data Output Delay Time		4	ns	
T_{DST}	I2S Input Data Setup Time	7		ns	
T_{DHT}	I2S Input Data Hold Time	0		ns	

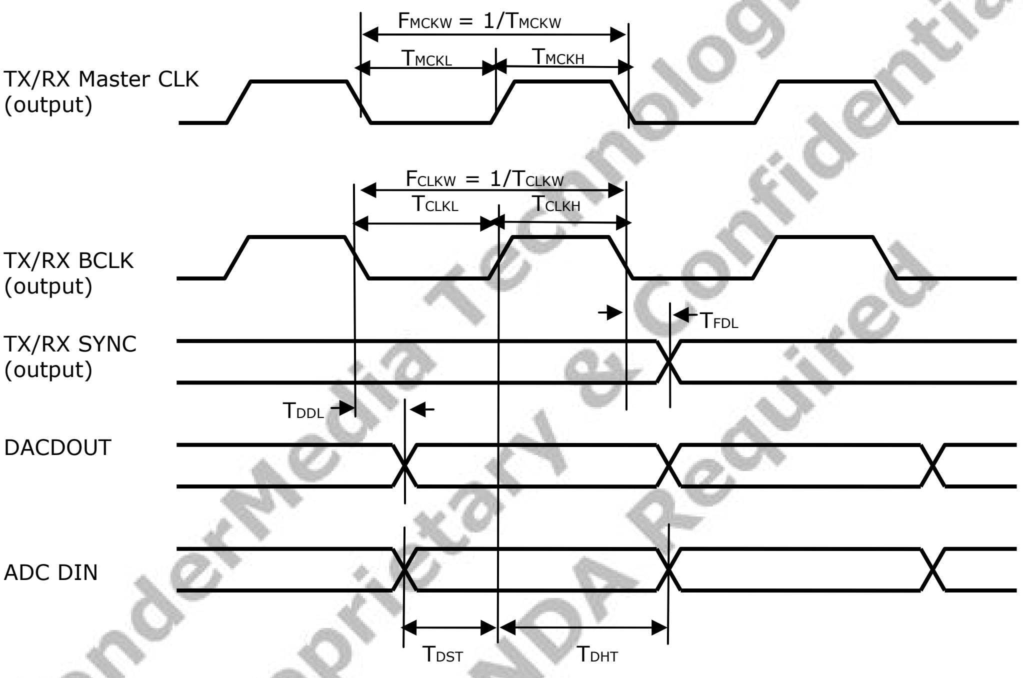


Figure 43 - Audio I^{2}S Interface Timing of Master Mode

Table 26 - Audio I^{2}S Interface Timing of Slave Mode

Symbol	Parameter	Min	Max	Unit	Note
F_{CLKW}	Bit CLK Frequency	24.576		MHz	
T_{CLKH}	Bit CLK Cycle High	0.45*T_{CLKW}		ns	
T_{CLKL}	Bit CLK Cycle Low	0.45*T_{CLKW}		ns	
T_{FST}	Frame Sync Clock Setup Time	3		ns	
T_{FHT}	Frame Sync Clock Hold Time	3		ns	
T_{DST}	I2S Input Data Setup Time	3		ns	
T_{DHT}	I2S Input Data Hold Time	3		ns	
T_{DDL}	I2S Output Data Output Delay Time		9	ns	

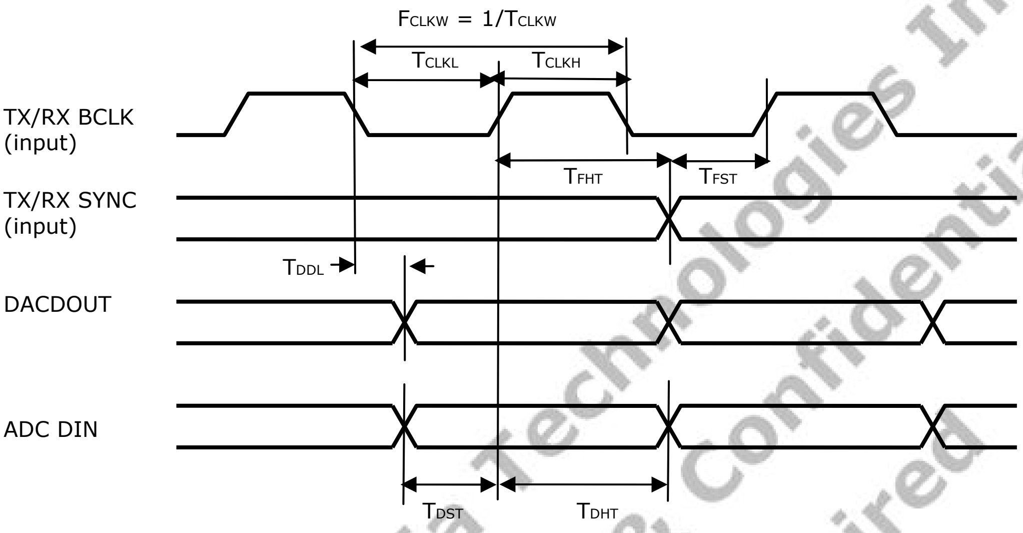


Figure 44 - Audio I^{2}S Interface Timing of Slave Mode

UART Interface Timing

Symbol	Parameter	Baud Rate	Value	Unit	Note
T_{1bit}	UART RX/TX 1 Bit Elapses Time	230.4 Kbps	4.34	μs	Typical
T_{1frame}	UART RX/TX 1 Frame Elapses Time	230.4 Kbps	47.74	μs	Typical

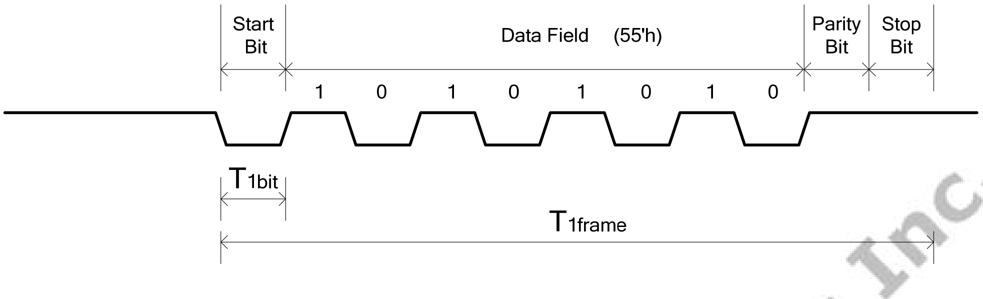


Figure 45 - UART Interface Timing

SPI Interface Timing

Table 27 - SPI Interface Timing – SPI Master, Divisor = 0

Symbol	Parameter	Min	Max	Unit	Note
T_{DIS}	Data Input Setup Time (Data Input Valid to SPICLK Edge)	2		ns	
T_{DIH}	Data Input Hold Time (SPICLK Last Sampling Edge to Data Input Not Valid)	1		ns	
T_{SSDW}[1]	SPISS# De-assertion Pulse Width	Tcyc		ns	
T_{CYC}	Clock Cycle	10		ns	
T_{CH}	Clock High Period	0.45*Tcyc		ns	
T_{CL}	Clock Low Period	0.45*Tcyc		ns	
T_{DOD}	Data Out Delay (SPICLK Edge to Data Out Valid)		2	ns	
T_{SS2C0}[1]	SPISS# Low to First SPICLK Edge for CPHASE = 0	T_{CL}-2		ns	
T_{SS2C1}[1]	SPISS# Low to First SPICLK Edge for CPHASE = 1	0	2	ns	
T_{CLK2SS}[1]	Last SPICLK Edge to SPISS# High	T_{CL}	T_{CL}+2	ns	

Table 28 - SPI Interface Timing – SPI Master, Divisor ≥ 1

Symbol	Parameter	Min	Max	Unit	Note
T_{DIS}	Data Input Setup Time (Data Input Valid to SPICLK Edge)	2		ns	
T_{DIH}	Data Input Hold Time (SPICLK Last Sampling Edge to Data Input Not Valid)	1		ns	
T_{SSDW}[1]	SPISS# De-assertion Pulse Width	Tcyc		ns	
T_{CYC}	Clock Cycle	20		ns	
T_{CH}	Clock High Period	0.45*Tcyc		ns	
T_{CL}	Clock Low Period	0.45*Tcyc		ns	
T_{DOD}	Data Out Delay (SPICLK Edge to Data Out Valid)		5	ns	
T_{SS2C0}[1]	SPISS# Low to First SPICLK Edge for CPHASE = 0	T_{CL}-5		ns	
T_{SS2C1}[1]	SPISS# Low to First SPICLK Edge for CPHASE = 1	0	5	ns	
T_{CLK2SS}[1]	Last SPICLK Edge to SPISS# High	0	5	ns	

Note:

1. These timing can be configured by using register.

T_{SPICLK} = CCP Clock to SPI Controller
T_{CYC} = Serial Required Clock = 2 * Divisor * T_{SPICLK}
If Divisor = 0, T_{CYC} = T_{SPICLK}
Divisor = "SPI Control Register [31:21]"

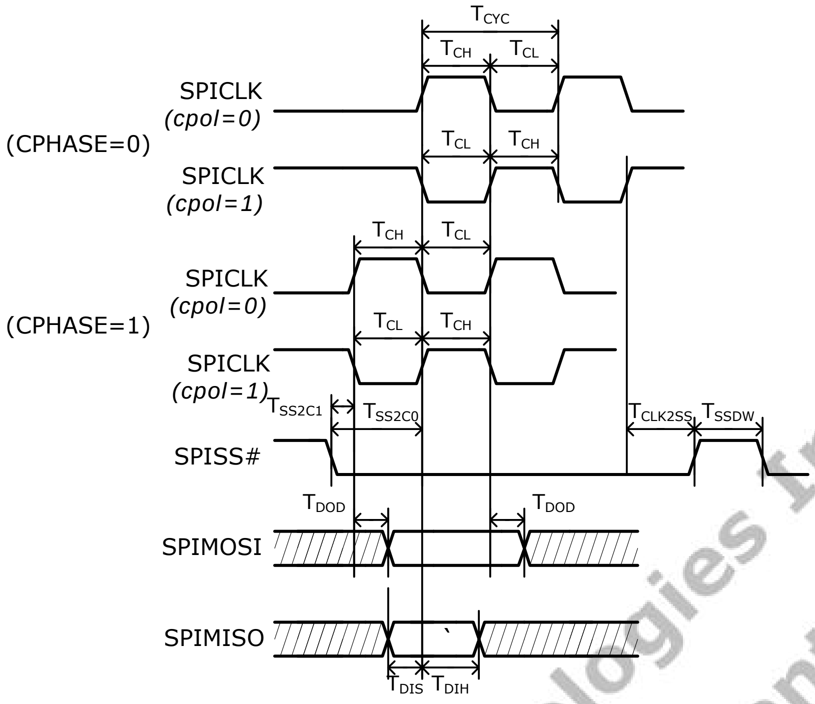


Figure 46 - SPI Interface Timing – SPI Master

Table 29 - SPI Interface Timing – SPI Slave

Symbol	Parameter	Min	Max	Unit	Note
T_{CYC}	Clock Cycle	10		ns	
T_{CH}	Clock High Period	0.45*Tcyc		ns	
T_{CL}	Clock Low Period	0.45*Tcyc		ns	
T_{SS2C0}	SPI_SS# Assertion to First SPI_CLK Edge for CPHASE = 0	2		ns	
T_{CLK2SS}	Last SPI_CLK Edge to SPI_SS# Not Asserted	0		ns	
T_{SSDW}	SPISS# De-assertion Pulse Width	Tcyc		ns	
T_{DIS}	Data Input Setup Time (Data Input Valid to SPI_CLK Edge)	2		ns	
T_{DIH}	Data Input Hold Time (SPI_CLK Last Sampling Edge to Data Input Not Valid)	1		ns	
T_{DOV}	SPI_SS# Assertion to Data Out Valid		5	ns	
T_{DOD}	Data Out Delay (SPI_CLK Edge to Data Out Valid)		5	ns	

Note:

T_{SPICLK} = CCP Clock to SPI Controller
$T_{CYC} > T_{SPICLK}$

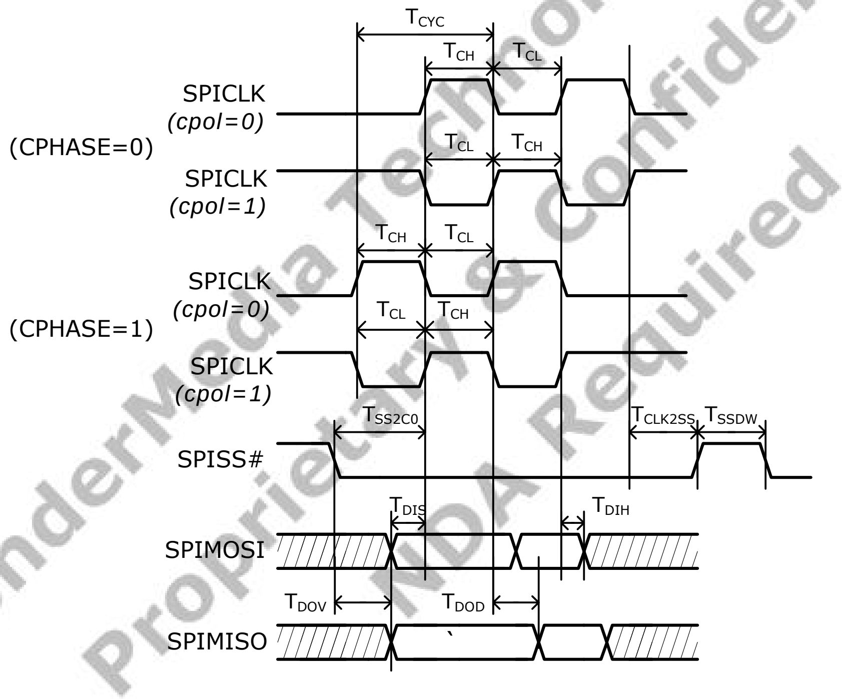


Figure 47 - SPI Interface Timing – SPI Slave

I^{2}C Interface Timing

Table 30 - I^{2}C Interface Standard and Fast Mode Timing

Symbol	Parameter	Standard Mode		Fast Mode		Unit	Note
		Min	Max	Min	Max		
$1/T_{CYC}$	I^{2}C Clock Frequency		100		400	kHz	
T_{CH}	I^{2}C Clock High Pulse	4.0		0.6		us	
T_{CL}	I^{2}C Clock Low Pulse	4.7		1.3		us	
T_{SS}	I^{2}C Stop Setup Time	4.0		0.6		us	
T_{FREE}	I^{2}C Bus Free Time Between Start and Stop	4.7		1.3		us	
T_{SH}	I^{2}C Start Hold Time	4.0		0.6		us	
T_{WDS}	I^{2}C Data Out Setup Time	250		100		ns	
T_{WDH}	I^{2}C Data Out Hold Time		3.45		0.9	ns	

Table 31 - I^{2}C Interface High Speed Mode Timing

Symbol	Parameter	C_b = 100 pF		C_b = 400 pF		Unit	Note
		Min	Max	Min	Max		
$1/T_{CYC}$	I^{2}C Clock Frequency		3.4		1.7	MHz	
T_{CH}	I^{2}C Clock High Pulse	60		120		ns	
T_{CL}	I^{2}C Clock Low Pulse	160		320		ns	
T_{SS}	I^{2}C Stop Setup Time	160		160		ns	
T_{SH}	I^{2}C Start Hold Time	160		160		ns	
T_{WDS}	I^{2}C Data Out Setup Time	10		10		ns	
T_{WDH}	I^{2}C Data Out Hold Time		70		150	ns	

Note:

For bus line loads Cb between 100 and 400 pF, the timing parameters must be linearly interpolated.

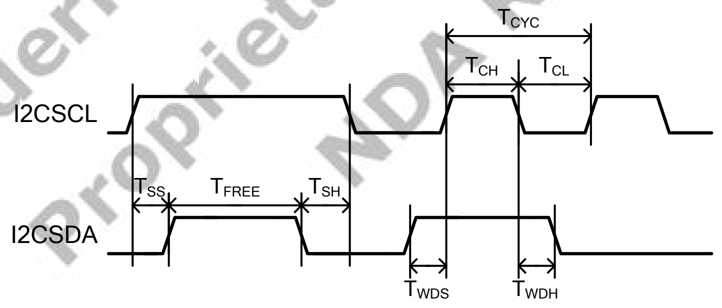


Figure 48 - I^{2}C Interface Timing

AC97 Interface Timing

Symbol	Parameter	Min	Max	Unit	Note
T_{ODS}	AC97 SYNC Output Valid Delay		15	ns	
T_{ODD}	AC97 Serial Data Output Valid Delay		15	ns	
T_{DS}	AC97 Serial Data Input Setup	10		ns	
T_{DH}	AC97 Serial Data Input Hold	10		ns	

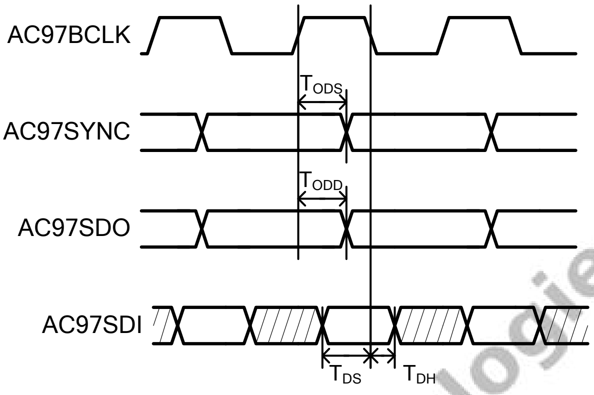


Figure 49 - AC97 Interface Timing

ARM JTAG Interface Timing

Table 32 - ARM JTAG Interface Timing – Test Clock

Symbol	Parameter	Min	Typ	Max	Unit	Note
T_{CYC}	JTAG Test Clock Period	200			ns	
T_{CH}	JTAG Test Clock Low Pulse Width	100			ns	
T_{CL}	JTAG Test Clock High Pulse Width	100			ns	

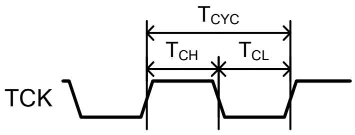


Figure 50 - ARM JTAG Interface Timing – Test Clock

Table 33 - ARM JTAG Interface Timing – Test Reset

Symbol	Parameter	Min	Typ	Max	Unit	Note
T_{RW}	JTAG Test Reset Pulse Width	50000	-	-	TCK cycles	

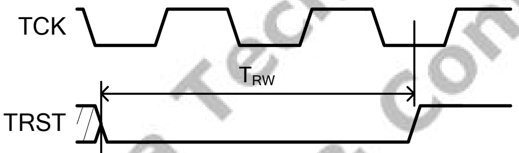


Figure 51 - ARM JTAG Interface Timing – Test Reset

Table 34 - ARM JTAG Interface Timing – Test Data/Select

Symbol	Parameter	Min	Max	Unit	Note
T_{SS}	JTAG Test Data Select (JTAG_TMS) Input Setup Time	10		ns	
T_{SH}	JTAG Test Data Select (JTAG_TMS) Input Hold Time	0		ns	
T_{DIS}	JTAG Test Data In (JTAG_TDI) Input Setup Time	10		ns	
T_{DIH}	JTAG Test Data In (JTAG_TDI) Input Hold Time	0		ns	
T_{DOD}	JTAG Test Data Out (JTAG_TDO) Output Valid Delay Time		10	ns	

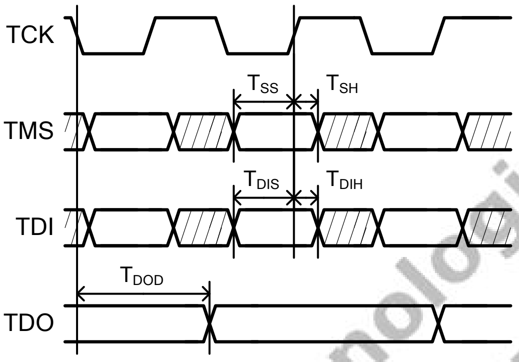
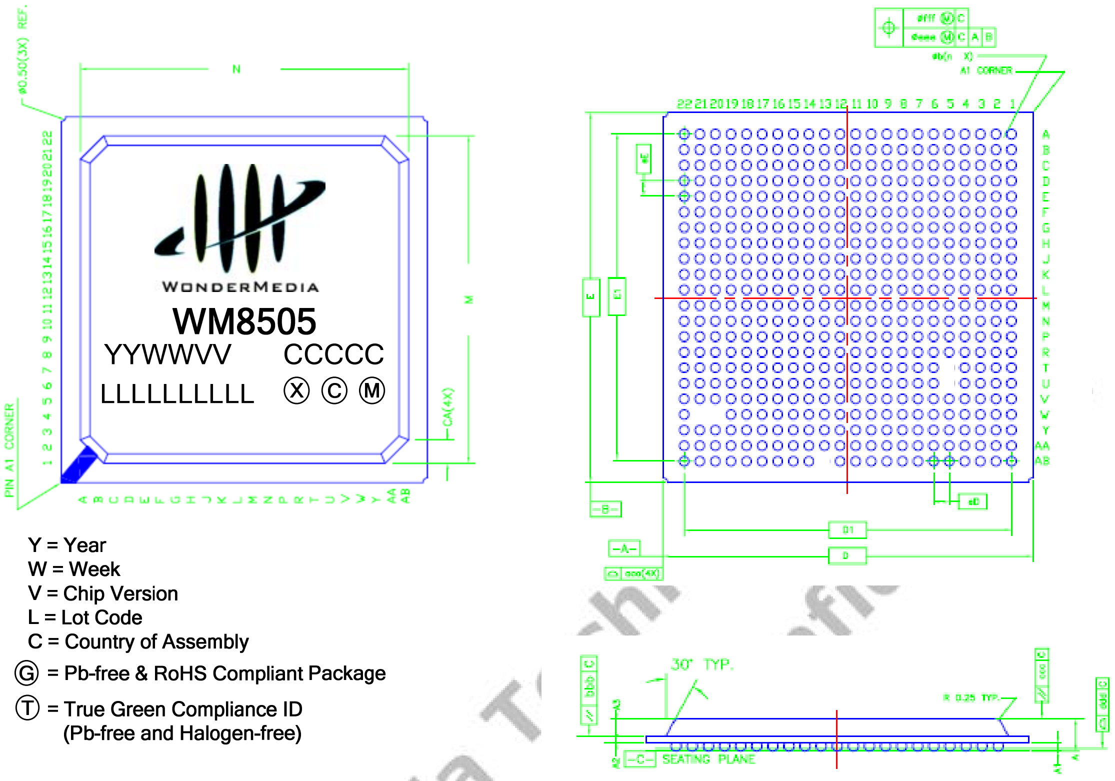


Figure 52 - ARM JTAG Interface Timing – Test Data/Select

Package Mechanical Specifications



Y = Year
W = Week
V = Chip Version
L = Lot Code
C = Country of Assembly
Ⓖ = Pb-free & RoHS Compliant Package
Ⓣ = True Green Compliance ID
(Pb-free and Halogen-free)

Figure 53 – WM8505 Mechanical Specification

Unit: mm

Name	Symbol	Dimension
Package		PBGA
Body Size	D / E	19.000
Ball Pitch	eD / eE	0.800
Total Thickness	A	1.61±0.190
Mold Thickness	A3	0.850 Ref.
Substrate Thickness	A2	0.360 Ref.
Ball Diameter		0.500
Stand Off	A1	0.300~0.500
Ball Width	b	0.400~0.600
Mold Area	M / N	17.000
Chamfer	CA	1.220 Ref.
Package Edge Tolerance	aaa	0.200
Substrate Flatness	bbb	0.250
Mold Flatness	ccc	0.350
Coplanarity	ddd	0.200
Ball Offset (Package)	eee	0.250
Ball Offset (Ball)	fff	0.100
Ball Count	N	479
Edge Ball Center to Center	D1 / E1	16.800

No license is granted, implied or otherwise, under any patent or patent rights of WonderMedia Technologies Inc. WonderMedia Technologies Inc. makes no warranties, implied or otherwise, in regard to this document and to the products described in this document. The information provided by this document is believed to be accurate and reliable as of the publication date of this document. However, WonderMedia Technologies Inc. assumes no responsibility for any errors in this document. Furthermore, WonderMedia Technologies Inc. assumes no responsibility for the use or misuse of the information in this document and for any patent infringements that may arise from the use of this document. The information and product specifications within this document are subject to change at any time, without notice and without obligation to notify any person of such change.

WonderMedia Technologies Inc.
1F, 531, Chung-Cheng Road,
Hsin-Tien, Taipei, 231 Taiwan
Tel (886-2) 2218-2250
Fax (886-2) 2218-5208
URL http://www.wondermedia.com.tw